HITACHI
Inspire the Next

uCosminexus Business Reporting Processor **開発者ガイド**

解説・文法書

3020-3-N39-20

マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，巻末の「ソフトウェアマニュアルのサービス ご案内」をご参照ください。

■対象製品

P-245Z-A154 uCosminexus Business Reporting Processor 05-00 （適用 OS：Windows 2000，Windows XP，Windows Server 2003）
P-9S5Z-A144 uCosminexus Business Reporting Processor 04-00 （適用 OS：Red Hat Enterprise Linux AS 4 update3（x86），Red Hat Enterprise Linux AS 4 update4（x86），Red Hat Enterprise Linux ES 4 update3（x86），Red Hat Enterprise Linux ES 4 update4（x86）

■輸出時の注意

本製品を輸出される場合には，外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上，必要な手続きをお取りください。
なお，ご不明な場合は，弊社担当営業にお問い合わせください。

■商標類

Java 及びすべての Java 関連の商標及びロゴは，米国及びその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。
Linux は，Linus Torvalds の米国およびその他の国における登録商標または商標です。
Microsoft は，米国およびその他の国における米国 Microsoft Corp. の登録商標です。
Red Hat は，米国およびその他の国で Red Hat, Inc. の登録商標若しくは商標です。
Sun Microsystems は，米国 Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。
UNIX は，X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。
Windows は，米国およびその他の国における米国 Microsoft Corp. の登録商標です。
Windows Server は，米国 Microsoft Corporation の米国及びその他の国における登録商標です。
XBRL は，American Institute of Certified Public Accountants（「AICPA」）の商標です。

■発行

2006 年 11 月（第 1 版）3020-3-N39
2008 年 11 月（第 3 版）3020-3-N39-20

■著作権

All Rights Reserved. Copyright (C) 2006, 2008, Hitachi, Ltd.

変更内容

変更内容（3020-3-N39-20）uCosminexus Business Reporting Processor 05-00

追加・変更内容	変更個所
XBRL Dimensions1.0 に対応した。 これに伴い，API を次のように変更した。 com.hitachi.xbrl パッケージ 次のクラスを追加した。 • XBRLDimensionException • XBRLDimensionUserException com.hitachi.xbrl.instance パッケージ 次のインターフェースを追加した。 • XBRLDimensionMember • XBRLExplicitMember • XBRLTypedMember 次のインターフェースにメソッドまたはフィールドを追加した。 • XBRLInstanceDocument • XBRLInstanceDocumentBuilder • XBRLInstanceDocumentBuilderFactory • XBRLInstanceItem • XBRLInstanceNode • XBRLScenario • XBRLSegment 次のインターフェースのメソッドの説明を変更した。 • XBRLInstanceDocument • XBRLInstanceDocumentBuilder • XBRLInstanceNode 次のインターフェースに Dimensions の場合の説明を追加した。 • XBRLScenario • XBRLSegment	1.1.1，1.2.3，2.1，2.2，3.4.3，4.1，4.10，4.11，4.12，4.15，4.16.1，4.16.2 (23) (24)，4.16.3 (4)，5.3.1，5.3.2 (3) (4)，5.3.5 (12)，5.4.1，5.4.9，6.1.1，6.1.3 (1) (2)，6.1.4 (2)，6.2.1，6.2.2，6.2.3，6.2.4，6.2.6，6.3，6.5，6.5.6，6.5.7，6.6，6.6.7 (81) (83) (84) (85)，6.6.8 (8) (23) (24) (25) (26) (27) (28) (29)，6.6.9 (17) (18)，6.6.12 (30)，6.6.13 (18) (19) (20)，6.6.14 (1) (4) (5) (6), 6.6.24 (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13)， 6.6.26 (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13)，6.6.29，6.6.30，6.6.31，6.7，6.7.7 (1) (5) (6)，6.7.8，6.7.9，6.7.10，6.7.11，6.7.12，6.7.13，6.7.14，6.7.15，6.7.16，6.7.18，6.8，6.8.13 (31) (32)，6.8.14 (13)，6.8.15 (19) (20)，6.8.16，6.8.17 (5)，6.8.18 (1) (3) (4)，6.8.23，6.8.24，6.10.4，6.10.10 (11)，6.11.9 (2)，付録 A.2 (1)，付録 A.3 (1)

追加・変更内容	変更個所
com.hitachi.xbrl.link パッケージ 次のインターフェースを追加した。 • XBRLDimensionalRelationship • XBRLHasHypercubeRelationship • XBRLHypercubeDimensionRelationship • XBRLDimensionDomainRelationship • XBRLDomainMemberRelationship • XBRLDimensionDefaultRelationship • XBRLDimensionalRelationshipSet • XBRLDRSDimension • XBRLDRSMember 次のクラスを追加した。 • XBRLDimensionIllegalCycleException 次のインターフェースにフィールドを追加した。 • XBRLRelationship 次のインターフェースのメソッドの説明を変更した。 • XBRLRelationship com.hitachi.xbrl.taxonomy パッケージ 次のインターフェースを追加した。 • XBRLTaxonomyHypercubeItem • XBRLTaxonomyDimensionItem 次のインターフェースにメソッドまたはフィールドを追加した。 • XBRLTaxonomyDocument • XBRLTaxonomyDocumentBuilder • XBRLTaxonomyDocumentBuilderFactory • XBRLTaxonomyElement • XBRLTaxonomyItem • XBRLTaxonomyNode 次のインターフェースのメソッドの説明を変更した。 • XBRLTaxonomyNode com.hitachi.xbrl.util パッケージ 次のクラスにメソッドまたはフィールドを追加した。 • XBRLConstants • XBRLEqualUtil com.hitachi.xbrl.simple.taxonomy パッケージ 次のインターフェースのメソッドの説明を変更した。 • SimpleTaxonomyBuilder	
タプルの検証内容を変更した。	2.3.3 (5)

追加・変更内容	変更個所
製品に Xlink Proccessor を取り込んだ。これに伴い，製品のバージョンが 05-00 の場合の，csmxlpr.jar ファイルのデフォルトの格納先を追加した。	4.2.1
Generics に対応した。 これに伴い，サンプルコードを変更した。 また，メソッドの形式に戻り値を追加した。	4.3.2 (2)，4.5.2 (2)，4.6.3 (2)，4.8.2 (2)，4.9.4 (2)，6.6.2 (10) (11)，6.6.5 (13) (14)，6.6.7 (7) (12) (20) (23) (24) (29) (33) (37) (43) (48) (49) (54) (55) (73) (74) (75) (76), 6.6.8 (5)，6.6.9 (9)，6.6.12 (8) (9) (10) (12) (13) (14) (15) (16) (20) (21) (22) (23) (28) (30), 6.6.15 (2) (3)，6.6.24 (4) (5) (6)，6.6.26 (4) (5) (6)，6.6.28 (4) (5) (6) (7) (8)，6.7.9 (2) (3) (4) (5) (6) (7) (9) (10) (11), 6.8.1 (4)，6.8.3 (1)，6.8.6 (5) (6)，6.8.8 (3)，6.8.10 (4) (5)，6.8.11 (4) (5) (7)，6.8.13 (4) (5) (7) (8) (9) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (24) (25) (26) (27) (28) (29) (31), 6.8.16 (10) (11) (12) (14) (15) (16) (17) (18) (19) (20) (21)，6.10.8 (2)
エンティティリゾルバのサンプルコードを，XBRL2.0，XBRL2.1，XBRL Dimensions1.0 の場合で共通にした。	4.14.1
インスタンス文書の妥当性検証を実施する場合に，<link:schemaRef> 要素をあらかじめ検索してからスキーマ検証する機能を追加した。これに伴い，プロパティを追加した。	4.16.2 (25)

追加・変更内容	変更個所
メッセージを追加した。 KEBR16000-E，KEBR16001-E， KEBR16002-E，KEBR16003-E， KEBR16004-E，KEBR16005-E， KEBR16006-E，KEBR16007-E， KEBR16008-E，KEBR16009-E， KEBR16010-E，KEBR16011-E， KEBR16012-E，KEBR16013-E， KEBR16014-E，KEBR16015-E， KEBR16016-E，KEBR16017-E， KEBR16018-E，KEBR16019-E， KEBR16020-E，KEBR16021-E， KEBR16022-E，KEBR16023-E， KEBR16200-E，KEBR16201-E， KEBR16202-E，KEBR16203-E， KEBR16204-E，KEBR16205-E， KEBR16206-E，KEBR16207-E， KEBR16208-E，KEBR16209-E， KEBR16210-E，KEBR16211-W， KEBR16212-I，KEBR22100-E， KEBR22101-E，KEBR22102-E， KEBR22103-E，KEBR22104-E， KEBR22105-E，KEBR22106-E	7.2，7.3，7.4，付録 A.2 (1)，付録 A.3 (1)

単なる誤字・脱字などはお断りなく訂正しました。

変更内容（3020-3-N39-10）uCosminexus Business Reporting Processor 04-00

追加・変更内容
前提 OS に，Red Hat Enterprise Linux AS 4 update3（x86），Red Hat Enterprise Linux AS 4 update4（x86），Red Hat Enterprise Linux ES 4 update3（x86）および Red Hat Enterprise Linux ES 4 update4（x86）を追加した。
メッセージを追加した。 KEXL00001-E，KEXL00002-E，KEXL00003-E，KEXL00004-E，KEXL00005-E， KEXL00006-E，KEXL00007-E，KEXL00008-E，KEXL00009-E，KEXL00010-E， KEXL00011-E，KEXL00012-E，KEXL00013-E，KEXL00014-E，KEXL10001-E， KEXL10002-E，KEXL10003-E，KEXL10004-E，KEXL10005-E，KEXL10006-E， KEXL10007-E，KEXL10008-E，KEXL10009-E，KEXL10010-E，KEXL10011-E， KEXL10012-E，KEXL10013-E，KEXL10014-E，KEXL10015-E，KEXL10016-E， KEXL10017-E，KEXL10018-E，KEXL10019-E，KEXL10020-E，KEXL10021-E， KEXL10022-E，KEXL10023-E，KEXL10024-E，KEXL10025-E，KEXL10026-E， KEXL10027-E，KEXL10028-E，KEXL10029-E，KEXL10030-E，KEXL10031-E， KEXL10034-E，KEXL10035-E，KEXL10036-E，KEXL10037-E，KEXL10038-E， KEXL10039-E，KEXL10040-E，KEXL10041-E，KEXL10042-E，KEXL10043-E， KEXL10044-E，KEXL10045-E，KEXL19998-E

はじめに

このマニュアルは，プログラムプロダクト　P-245Z-A154　uCosminexus Business Reporting Processor，および P-9S5Z-A144　uCosminexus Business Reporting Processor を使用した，eXtensible Business Reporting Language の規格に従った文書（XBRL 文書）を操作するアプリケーションを開発する方法について説明したものです。

■対象読者

uCosminexus Business Reporting Processor が提供する API を利用して，XBRL 文書を操作するアプリケーションを開発する方を対象としています。
このマニュアルを読む方は，Java 言語を使用したプログラミングに関する知識，および次の標準仕様についての知識を理解されていることを前提としています。

- eXtensible Business Reporting Language (XBRL) 2.0 Specification
- eXtensible Business Reporting Language (XBRL) 2.1 Specification
- XBRL Dimensions 1.0
- XBRL 2.1 Conformance Suite 1.0
- Extensible Markup Language (XML) 1.0 (Second Edition)
- XML Schema Part 0: Primer
- XML Schema Part 1: Structures
- XML Schema Part 2: Datatypes
- XML Base
- Namespaces in XML
- XML Path Language (XPath) Version 1.0
- XPointer Framework
- XPointer Element() scheme
- XML Linking Language (XLink) Version 1.0
- Document Object Model (DOM) Level 2 Core Specification Version 1.0

各仕様書は，XBRL International，または World Wide Web Consortium（W3C）で提供されています。

■マニュアルの構成

このマニュアルは，次に示す章と付録から構成されています。

第 1 章　uCosminexus Business Reporting Processor の概要

XBRL の概要，uCosminexus Business Reporting Processor の特長，および前提プログラムを含めたシステムでの位置づけについて説明しています。

第 2 章　XBRL 文書の参照と検証

XBRL 文書の参照や検証時に実施される XBRL 文書の読み込みやリンクの上書きなどについて説明しています。

第 3 章　XBRL 文書の作成と編集

XBRL 文書の作成や編集時に実施される ID の採番や名前空間の正規化について説明しています。

第 4 章　アプリケーションの開発

アプリケーションの開発方法とサンプルコードについて説明しています。

第 5 章　取り扱える XBRL 文書とデータ型

uCosminexus Business Reporting Processor が取り扱える XBRL 文書の形式，およびデータ型について説明しています。

第 6 章　API リファレンス

XBRL 文書を操作するための API について説明しています。

第 7 章　メッセージ

API 利用時に表示されるメッセージについて説明しています。

付録 A　XBRL 文書の検証内容と対応するメッセージ ID

XBRL 文書の詳細な検証内容と，表示されるメッセージ ID との対応について説明しています。

付録 B　脚注リンクの再作成

XBRL 文書編集時に再作成される脚注リンクについて説明しています。

付録 C　バージョン間の差異

uCosminexus Business Reporting Processor のバージョン間の差異について説明しています。

付録 D　用語解説

このマニュアルで使用する用語の意味について説明しています。

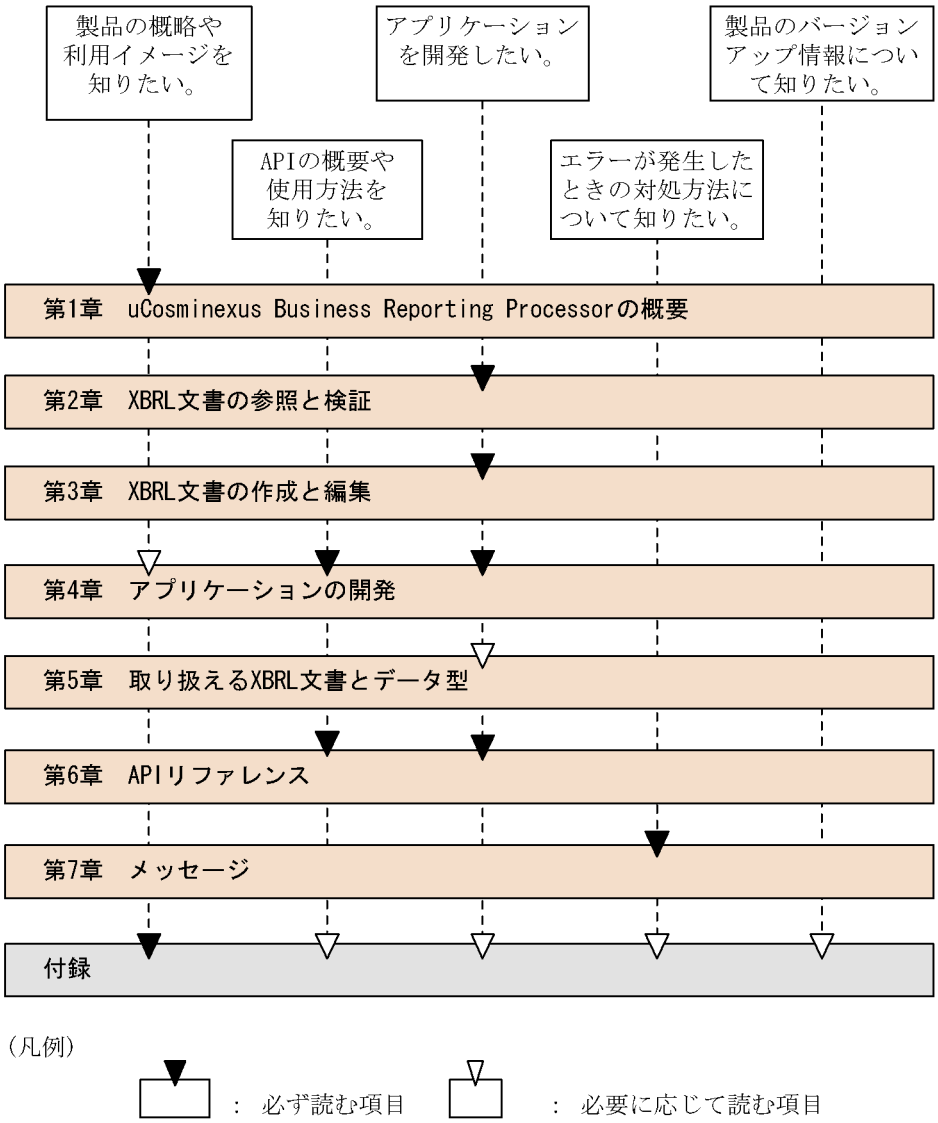
■関連マニュアル

このマニュアルは，次のマニュアルと関連しています。必要に応じてお読みください。

- Cosminexus XML Processor ユーザーズガイド（3020-3-M44）
- Processing Kit for XML ユーザーズガイド（3000-3-935）
- uCosminexus XML Link Processor ユーザーズガイド（3020-3-J40）

■読書手順

このマニュアルは，利用目的に合わせて章を選択して読むことができます。利用目的別に，次の流れに従ってお読みいただくことをお勧めします。



■このマニュアルでの表記

このマニュアルで使用する英略語を次に示します。

英略語	英字での表記
AP	Application Program
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
c-equal	context-equal

英略語	英字での表記
cwa	Closed World Assumption
DOM	Document Object Model
DRS	Dimensional Relationship Set
DTD	Document Type Definition
DTS	Discoverable Taxonomy Set
HTTP	Hyper Text Transfer Protocol
ISO	International Standardization Organization
J2SE	Java™2 Platform,Standard Edition
JAXP	Java™ API for XML Processing
p-equal	parent-equal
PTVI	Post-Taxonomy Validation Infoset
PTVLI	Post-Taxonomy Validation Linkbase Infoset
RFC	Request For Comments
s-equal	structure-equal
SDK	Software Development Kit
u-equal	unit-equal
UML	Unified Modeling Language
URI	Universal Resource Identifier
UTC	Coordinated Universal Time
UTF	UCS Transformation Format
v-equal	value-equal
VM	Virtual Machine
W3C	World Wide Web Consortium
x-equal	[XPATH]-equal
XBRL	eXtensible Business Reporting Language™
XHTML	eXtensible Hyper Text Markup Language
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformations

このマニュアルでは，製品名称を省略して表記しています。製品の正式名称と，このマニュアルでの表記を次に示します。

製品の正式名称	このマニュアルでの表記	
Microsoft(R) Windows(R) 2000 Professional Operating System	Windows 2000	Windows

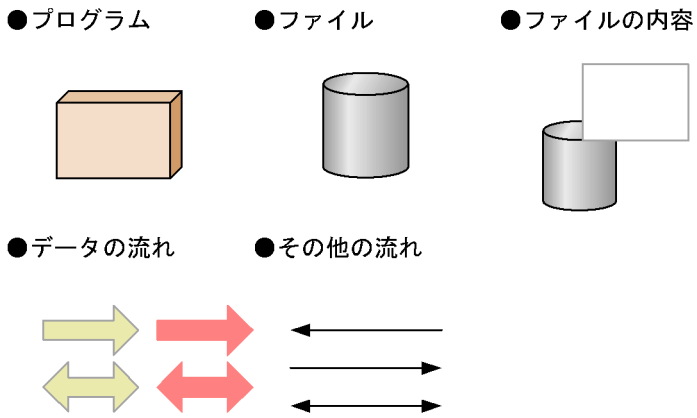
製品の正式名称	このマニュアルでの表記	
Microsoft(R) Windows(R) 2000 Server Operating System		
Microsoft(R) Windows(R) 2000 Advanced Server Operating System		
Microsoft(R) Windows(R) XP Professional	Windows XP	
Microsoft(R) Windows Server™ 2003, Standard Edition	Windows Server 2003	
Microsoft(R) Windows Server™ 2003, Enterprise Edition		
Red Hat Enterprise Linux AS 4 update3（x86）	Linux	
Red Hat Enterprise Linux AS 4 update4（x86）		
Red Hat Enterprise Linux ES 4 update3（x86）		
Red Hat Enterprise Linux ES 4 update4（x86）		
uCosminexus XML Link Processor	XLink プロセッサ	
uCosminexus Application Server	XML プロセッサ	
Processing Kit for XML		
日立ネットワークオブジェクトプラザトレース共通ライブラリ	日立トレース共通ライブラリ	

このマニュアルでは，XBRL 文書内に記述されている要素や属性名に，名前空間接頭辞を付けて表記しています。使用する名前空間接頭辞と対応する名前空間 URI を次に示します。

名前空間接頭辞	名前空間 URI
xsd	http://www.w3.org/2001/XMLSchema
xsi	http://www.w3.org/2001/XMLSchema-instance
xlink	http://www.w3.org/1999/xlink
xbrli	http://www.xbrl.org/2001/instance（XBRL2.0 の場合） http://www.xbrl.org/2003/instance（XBRL2.1 の場合）
link	http://www.xbrl.org/2001/XLink/xbrllinkbase（XBRL2.0 の場合） http://www.xbrl.org/2003/linkbase（XBRL2.1 の場合）
xl	http://www.xbrl.org/2001/XLink（XBRL2.0 の場合） http://www.xbrl.org/2003/XLink（XBRL2.1 の場合）
ptv	http://www.xbrl.org/2003/ptv（XBRL2.1 の場合）
xbrldi	http://xbrl.org/2006/xbrldi（XBRL Dimensions1.0 の場合）
xbrldie	http://xbrl.org/2005/xbrldi/errors（XBRL Dimensions1.0 の場合）
xbrldt	http://xbrl.org/2005/xbrldt（XBRL Dimensions1.0 の場合）
xbrldte	http://xbrl.org/2005/xbrldt/errors（XBRL Dimensions1.0 の場合）

■図中で使用する記号

このマニュアルの図中で使用する記号を，次のように定義します。



■常用漢字以外の漢字の使用について

このマニュアルでは，常用漢字を使用することを基本としていますが，次に示す用語については，常用漢字以外の漢字を使用しています。
個所（かしょ） 桁（けた） 伝播（でんぱ） 必須（ひっす）

■ KB（キロバイト）などの単位表記について

1KB（キロバイト），1MB（メガバイト），1GB（ギガバイト），1TB（テラバイト）はそれぞれ1,024 バイト，$1{,}024^2$ バイト，$1{,}024^3$ バイト，$1{,}024^4$ バイトです。

■適用 OS による表現の相違について

このマニュアルでは，Windows の表現を使用して説明しています。Windows 以外の OS をご利用の場合は，Windows の表現をご利用の OS の表現に読み替えてください。

目次

1 uCosminexus Business Reporting Processor の概要 1

1.1 XBRL の概要 2

1.1.1 XBRL 文書 2

1.1.2 タクソノミー文書とは 4

1.1.3 インスタンス文書とは 5

1.1.4 タクソノミー文書の拡張 5

1.2 uCosminexus Business Reporting Processor の特長 7

1.2.1 uCosminexus Business Reporting Processor でできること 7

1.2.2 XBRL DOM とは 8

1.2.3 準拠する規格 9

1.3 uCosminexus Business Reporting Processor の位置づけ 11

2 XBRL 文書の参照と検証 13

2.1 タクソノミー文書の参照 15

2.1.1 タクソノミー文書の XBRL DOM ツリーへの展開 15

2.1.2 拡張されたタクソノミー文書の参照 16

2.1.3 タクソノミー文書の定義情報の参照 17

2.2 インスタンス文書の参照 20

2.3 タクソノミー文書の検証 21

2.3.1 XML スキーマ検証 21

2.3.2 リンクの検証 24

2.3.3 タクソノミー文書の形式の検証 29

2.4 インスタンス文書の検証 33

2.4.1 XML スキーマ検証 33

2.4.2 リンクの検証 33

2.4.3 インスタンス文書の形式の検証 41

2.5 DTS の読み込み 44

2.5.1 DTS の読み込み範囲と指定方法（XBRL2.0 の場合） 44

2.5.2 DTS の読み込み範囲と指定方法（XBRL2.1 の場合） 48

2.5.3 DTS のメモリキャッシュ 53

2.5.4 リンクベースの遅延読み込み 54

2.5.5 DTS の拡張読み込み（XBRL2.1 の場合） 57

2.6 リンクの上書き（XBRL2.0 の場合） 59

2.6.1 関係リンクの上書き 59
2.6.2 ラベルリンクの上書き 63
2.6.3 リファレンスリンクの上書き 66
2.6.4 脚注リンクの上書き 67
2.7 リンクの上書き（XBRL2.1 の場合） 68
2.7.1 リンクが上書きされる条件 68
2.7.2 priority 属性値が同じ場合のリンクの上書き 70
2.8 リンク構造の参照（XBRL2.1 の場合） 72
2.9 PTVLI の生成（XBRL2.1 の場合） 74
2.9.1 PTVLI の構造 74
2.9.2 PTVLI の生成例 75
2.10 PTVI の生成（XBRL2.1 の場合） 77
2.10.1 PTVI の構造 77
2.10.2 PTVI のアイテムの作成方法 78
2.10.3 PTVI の生成例 81
2.11 XBRL 同等性の判定 83
2.11.1 コンテキストの s-equal 判定 83
2.11.2 タプルの duplicate 判定 83
2.11.3 アイテムの s-equal 判定 83

3 XBRL 文書の作成と編集 85
3.1 タクソノミー文書の新規作成 86
3.1.1 タクソノミー文書の出力フォーマット 86
3.1.2 タクソノミー文書を作成する際の注意事項 91
3.2 インスタンス文書の新規作成と編集 92
3.2.1 インスタンス文書の出力フォーマット 92
3.3 ID 管理と採番 95
3.3.1 ID の管理と API を使用した要素同士の関係づけ 95
3.3.2 ID の採番方法 97
3.4 名前空間の正規化 100
3.4.1 名前空間一覧および接頭辞の自動生成 100
3.4.2 名前空間の正規化の例 101
3.4.3 正規化されない要素 107

4 アプリケーションの開発 111
4.1 この章の読み方 113
4.2 アプリケーションの開発環境の設定 115
4.2.1 クラスパスの設定 115
4.3 タクソノミー文書の参照と検証（XBRL2.0 の場合） 117
4.3.1 タクソノミー文書の読み込み 117
4.3.2 タクソノミー本体とリンクベースの参照 118
4.4 インスタンス文書の新規作成と保存（XBRL2.0 の場合） 122
4.4.1 インスタンス文書の新規作成 122
4.4.2 非数値コンテキスト要素の追加 124
4.4.3 数値コンテキスト要素の追加 127
4.4.4 非数値コンテキストに対応したアイテムの追加 129
4.4.5 数値コンテキストに対応したアイテムの追加 130
4.4.6 インスタンス文書の保存 132
4.5 既存のインスタンス文書の編集（XBRL2.0 の場合） 134
4.5.1 インスタンス文書の読み込み 134
4.5.2 脚注リンクの追加 135
4.5.3 インスタンス文書の再保存 138
4.6 タクソノミー文書の参照と検証（XBRL2.1 の場合） 139
4.6.1 タクソノミー文書の読み込み 139
4.6.2 タクソノミー本体とリンクベースの参照（アイテム名またはタプル名を参照する場合） 141
4.6.3 タクソノミー本体とリンクベースの参照（タクソノミー文書の要素の階層を参照する場合） 143
4.7 インスタンス文書の新規作成と保存（XBRL2.1 の場合） 146
4.7.1 インスタンス文書の新規作成 146
4.7.2 コンテキスト要素の追加 148
4.7.3 単位要素の追加 150
4.7.4 非数値アイテムの追加 151
4.7.5 数値アイテムの追加 152
4.7.6 インスタンス文書の保存 154
4.7.7 ID の採番をするインスタンス文書の作成 155
4.8 既存のインスタンス文書の編集（XBRL2.1 の場合） 160
4.8.1 インスタンス文書の読み込み 160
4.8.2 脚注リンクの追加 161
4.8.3 インスタンス文書の再保存 164

4.9 タクソノミー文書の拡張（XBRL2.1 の場合） 165
4.9.1 タクソノミー文書の拡張 165
4.9.2 アイテム定義の追加 167
4.9.3 ラベルの追加 168
4.9.4 関係の禁止 169
4.9.5 タクソノミー文書の保存 172
4.10 タクソノミー文書の参照と検証（XBRL Dimensions1.0 の場合） 174
4.10.1 タクソノミー文書の読み込み 174
4.10.2 タクソノミー本体とリンクベースの参照（DRS の情報を参照する場合） 175
4.11 インスタンス文書の新規作成と保存（XBRL Dimensions1.0 の場合） 179
4.11.1 多次元データを含んだコンテキスト要素の追加 179
4.11.2 プライマリー・アイテムの追加 182
4.12 既存のインスタンス文書の参照と検証（XBRL Dimensions1.0 の場合） 186
4.12.1 インスタンス文書の読み込み 186
4.12.2 インスタンス文書とリンクベースの参照（プライマリー・アイテムのディメンジョンとアイテムの値を参照する場合） 187
4.13 エラーハンドラ 191
4.14 エンティティリゾルバ 192
4.14.1 サンプルコード 192
4.15 サンプルアプリケーションの実行 194
4.15.1 サンプルファイルの格納場所 194
4.15.2 実行までの流れ 197
4.15.3 必要なスキーマ定義 197
4.16 アプリケーションの実行環境の設定 199
4.16.1 プロパティの指定方法と優先順位 199
4.16.2 プロパティ一覧 199
4.16.3 プロパティファイルの記載形式 209
4.16.4 文字コードの設定 212
4.16.5 JavaVM のカスタマイズ 212
4.17 トレースログ 213

5 取り扱える XBRL 文書とデータ型 215
5.1 取り扱えるタクソノミー文書（XBRL2.0 の場合） 216
5.1.1 タクソノミー文書の基本構造 216
5.1.2 エレメント要素（element） 217
5.1.3 リンクベースへの参照定義 220

5.1.4 リンクベース要素（linkbase） 221
5.2 取り扱えるインスタンス文書（XBRL2.0 の場合） 229
5.2.1 インスタンス文書の基本構造 229
5.2.2 ルート要素（group） 230
5.2.3 アイテム型要素（item） 231
5.2.4 コンテキスト要素（context） 231
5.2.5 脚注リンク要素（footnoteLink） 232
5.3 取り扱えるタクソノミー文書（XBRL2.1 の場合） 235
5.3.1 タクソノミー文書の基本構造 235
5.3.2 エレメント要素（element） 237
5.3.3 カスタムロール，カスタムアークロール 241
5.3.4 リンクベースへの参照定義 243
5.3.5 リンクベース要素（linkbase） 244
5.4 取り扱えるインスタンス文書（XBRL2.1 の場合） 256
5.4.1 インスタンス文書の基本構造 256
5.4.2 ルート要素（xbrl） 259
5.4.3 タクソノミー文書への参照定義 259
5.4.4 カスタムロール，カスタムアークロールへの参照定義 260
5.4.5 コンテキスト要素（context）と単位要素（unit） 261
5.4.6 アイテム型要素（item） 263
5.4.7 脚注リンク要素（footnoteLink） 264
5.4.8 タプル型要素（tuple） 266
5.4.9 ディメンジョンの指定（XBRL Dimensions1.0 の場合） 267
5.5 取り扱えるデータ型 268
5.5.1 XBRL データ型 268
5.5.2 ユーザーデータ型（XBRL2.0 の場合） 270
5.5.3 ユーザーデータ型（XBRL2.1 の場合） 272

6 API リファレンス 275
6.1 パッケージの概要 276
6.1.1 名前空間接頭辞 276
6.1.2 オブジェクトモデル（XBRL2.0 の場合） 277
6.1.3 オブジェクトモデル（XBRL2.1 の場合） 280
6.1.4 ノード 287
6.2 API 一覧（パッケージ順） 291
6.2.1 com.hitachi.xbrl パッケージ一覧 291

6.2.2 com.hitachi.xbrl.instance パッケージ一覧 292
6.2.3 com.hitachi.xbrl.link パッケージ一覧 311
6.2.4 com.hitachi.xbrl.taxonomy パッケージ一覧 314
6.2.5 com.hitachi.xbrl.types パッケージ一覧 324
6.2.6 com.hitachi.xbrl.util パッケージ一覧 331
6.2.7 com.hitachi.xbrl.simple.taxonomy パッケージ一覧 333
6.3 メソッド一覧（アルファベット順） 338
6.4 API リファレンスの形式 388
6.4.1 パッケージの形式 388
6.4.2 インターフェースおよびクラスの形式 388
6.4.3 メソッドの形式 388
6.5 com.hitachi.xbrl パッケージ 390
6.5.1 XBRLParseErrorHandler インターフェース 391
6.5.2 XBRLDocumentManager クラス 393
6.5.3 XBRLDOMException クラス 398
6.5.4 XBRLDOMRuntimeException クラス 399
6.5.5 XBRLDOMUserException クラス 400
6.5.6 XBRLDimensionException クラス 402
6.5.7 XBRLDimensionUserException クラス 405
6.6 com.hitachi.xbrl.instance パッケージ 409
6.6.1 XBRLArcroleRef インターフェース 411
6.6.2 XBRLContext インターフェース 412
6.6.3 XBRLContext2 インターフェース 419
6.6.4 XBRLEntity インターフェース 420
6.6.5 XBRLFootnote インターフェース 422
6.6.6 XBRLIdentifier インターフェース 429
6.6.7 XBRLInstanceDocument インターフェース 431
6.6.8 XBRLInstanceDocumentBuilder インターフェース 507
6.6.9 XBRLInstanceDocumentBuilderFactory インターフェース 536
6.6.10 XBRLInstanceDocumentWriter インターフェース 550
6.6.11 XBRLInstanceDocumentWriterFactory インターフェース 553
6.6.12 XBRLInstanceElement インターフェース 554
6.6.13 XBRLInstanceItem インターフェース 580
6.6.14 XBRLInstanceNode インターフェース 602
6.6.15 XBRLInstanceTuple インターフェース 616
6.6.16 XBRLLinkbaseRef インターフェース 622
6.6.17 XBRLMeasure インターフェース 624

6.6.18 XBRLMeasureOrOperator インターフェース 625
6.6.19 XBRLNonNumericContext インターフェース 626
6.6.20 XBRLNumericContext インターフェース 628
6.6.21 XBRLOperator インターフェース 630
6.6.22 XBRLPeriod インターフェース 634
6.6.23 XBRLRoleRef インターフェース 637
6.6.24 XBRLScenario インターフェース 639
6.6.25 XBRLSchemaRef インターフェース 651
6.6.26 XBRLSegment インターフェース 652
6.6.27 XBRLUnit インターフェース 663
6.6.28 XBRLUnit2 インターフェース 665
6.6.29 XBRLDimensionMember インターフェース 670
6.6.30 XBRLExplicitMember インターフェース 672
6.6.31 XBRLTypedMember インターフェース 673
6.7 com.hitachi.xbrl.link パッケージ 675
6.7.1 XBRLCalculationRelationship インターフェース 677
6.7.2 XBRLDefinitionRelationship インターフェース 678
6.7.3 XBRLFootnoteRelationship インターフェース 678
6.7.4 XBRLLabelRelationship インターフェース 678
6.7.5 XBRLPresentationRelationship インターフェース 678
6.7.6 XBRLReferenceRelationship インターフェース 679
6.7.7 XBRLRelationship インターフェース 679
6.7.8 XBRLDimensionalRelationship インターフェース 685
6.7.9 XBRLDimensionalRelationshipSet インターフェース 685
6.7.10 XBRLDimensionDefaultRelationship インターフェース 697
6.7.11 XBRLDimensionDomainRelationship インターフェース 698
6.7.12 XBRLDomainMemberRelationship インターフェース 699
6.7.13 XBRLDRSDimension インターフェース 700
6.7.14 XBRLDRSMember インターフェース 702
6.7.15 XBRLHasHypercubeRelationship インターフェース 704
6.7.16 XBRLHypercubeDimensionRelationship インターフェース 707
6.7.17 XBRLIllegalCycleException クラス 708
6.7.18 XBRLDimensionIllegalCycleException クラス 711
6.8 com.hitachi.xbrl.taxonomy パッケージ 715
6.8.1 XBRLArcroleType インターフェース 716
6.8.2 XBRLLabel インターフェース 719
6.8.3 XBRLLinkbase インターフェース 723

6.8.4 XBRLTaxonomyLinkbaseRef インターフェース 725
6.8.5 XBRLPart インターフェース 727
6.8.6 XBRLReference インターフェース 728
6.8.7 XBRLTaxonomyRoleRef インターフェース 732
6.8.8 XBRLRoleType インターフェース 734
6.8.9 XBRLTaxonomyAttributeDeclaration インターフェース 736
6.8.10 XBRLTaxonomyBody インターフェース 737
6.8.11 XBRLTaxonomyComplexTypeDefinition インターフェース 741
6.8.12 XBRLTaxonomyComponent インターフェース 746
6.8.13 XBRLTaxonomyDocument インターフェース 748
6.8.14 XBRLTaxonomyDocumentBuilder インターフェース 771
6.8.15 XBRLTaxonomyDocumentBuilderFactory インターフェース 780
6.8.16 XBRLTaxonomyElement インターフェース 796
6.8.17 XBRLTaxonomyItem インターフェース 814
6.8.18 XBRLTaxonomyNode インターフェース 822
6.8.19 XBRLTaxonomyOtherElement インターフェース 828
6.8.20 XBRLTaxonomySimpleTypeDefinition インターフェース 828
6.8.21 XBRLTaxonomyTuple インターフェース 828
6.8.22 XBRLTaxonomyTypeDefinition インターフェース 829
6.8.23 XBRLTaxonomyDimensionItem インターフェース 832
6.8.24 XBRLTaxonomyHypercubeItem インターフェース 834
6.9 com.hitachi.xbrl.types パッケージ 835
6.9.1 XBRLAbstractBinary クラス 836
6.9.2 XBRLAbstractPrecision クラス 838
6.9.3 XBRLBase64Binary クラス 843
6.9.4 XBRLDate クラス 849
6.9.5 XBRLDateTime クラス 853
6.9.6 XBRLDateUnion クラス 858
6.9.7 XBRLDecimals クラス 863
6.9.8 XBRLDuration クラス 866
6.9.9 XBRLFraction クラス 869
6.9.10 XBRLGDay クラス 872
6.9.11 XBRLGMonth クラス 875
6.9.12 XBRLGMonthDay クラス 877
6.9.13 XBRLGYear クラス 880
6.9.14 XBRLGYearMonth クラス 882
6.9.15 XBRLHexBinary クラス 885

6.9.16 XBRLPrecision クラス 891
6.9.17 XBRLQName クラス 893
6.9.18 XBRLTime クラス 900
6.10 com.hitachi.xbrl.util パッケージ 904
6.10.1 XBRLIdCreator インターフェース 904
6.10.2 XBRLDocumentWriter インターフェース 907
6.10.3 XBRLDocumentWriterFactory インターフェース 911
6.10.4 XBRLConstants クラス 912
6.10.5 XBRLIdManager クラス 919
6.10.6 XBRLNamespaceDeclaration クラス 923
6.10.7 XBRLSchemaLocation クラス 926
6.10.8 PTVInstance クラス 928
6.10.9 PTVLinkbase クラス 929
6.10.10 XBRLEqualUtil クラス 930
6.11 com.hitachi.xbrl.simple.taxonomy パッケージ 937
6.11.1 SimpleConceptDefinition インターフェース 937
6.11.2 SimpleEdge インターフェース 938
6.11.3 SimpleElementDefinition インターフェース 941
6.11.4 SimpleItemDefinition インターフェース 945
6.11.5 SimpleLabel インターフェース 951
6.11.6 SimpleLabelEdge インターフェース 956
6.11.7 SimpleNode インターフェース 957
6.11.8 SimpleTaxonomy インターフェース 959
6.11.9 SimpleTaxonomyBuilder インターフェース 972
6.11.10 SimpleTaxonomyBuilderFactory インターフェース 975

7 メッセージ 977
7.1 メッセージの形式 978
7.2 メッセージ一覧 980
7.3 メッセージ KEBR10000-KEBR19999 1002
7.4 メッセージ KEBR20000-KEBR29999 1081
7.5 メッセージ KEBR30000-KEBR39999 1128
7.6 メッセージ KEXL00000-KEXL09999 1133
7.7 メッセージ KEXL10000-KEXL19999 1137

付録

1149

付録 A　XBRL 文書の検証内容と対応するメッセージ ID　1150

付録 A.1　XBRL 文書共通の検証　1150

付録 A.2　インスタンス文書の検証　1155

付録 A.3　タクソノミー文書の検証　1161

付録 A.4　リンクベースおよびリンクの検証　1166

付録 B　脚注リンクの再作成　1170

付録 B.1　再作成される脚注リンク（XBRL2.0 の場合）　1170

付録 B.2　再作成される脚注リンク（XBRL2.1 の場合）　1172

付録 C　バージョン間の差異　1175

付録 D　用語解説　1176

索引

1181

図目次

図 1-1　タクソノミー文書とインスタンス文書の関係　3
図 1-2　タクソノミー文書の階層構造　6
図 1-3　XBRL DOM を使用した場合と，使用しない場合のアプリケーション開発　9
図 1-4　前提プログラムを含めたシステムでの位置づけ　11
図 2-1　タクソノミー文書の参照とツリーへの展開　16
図 2-2　拡張されたタクソノミー文書の構成　17
図 2-3　タクソノミー文書（リンクベース）の XML スキーマ検証　23
図 2-4　XBRL2.0 の場合のコンテキスト検証　42
図 2-5　インスタンス文書を解析する場合の DTS の読み込み範囲（XBRL2.0 の場合）　46
図 2-6　タクソノミー文書を解析する場合の DTS の読み込み範囲（XBRL2.0 の場合）　47
図 2-7　DTS として読み込むタクソノミー文書の階層の数え方　48
図 2-8　インスタンス文書を解析する場合の DTS の読み込み範囲（XBRL2.1 の場合）　50
図 2-9　タクソノミー文書を解析する場合の DTS の読み込み範囲（XBRL2.1 の場合）　51
図 2-10　リンクベースの遅延読み込み時に読み込まれる範囲　56
図 2-11　ラベルリンクを参照するときに読み込まれる範囲　58
図 2-12　関係リンクの同一リンクの判断　60
図 2-13　アークロールの目的が異なる例　61
図 2-14　アークロールに指定されている値が異なるが，目的が同じである例　62
図 2-15　priority 属性の値が同じ場合の例　63
図 2-16　ラベルリンクの同一リンクの判断　65
図 3-1　ID のフォーマット　98
図 5-1　タクソノミー文書の基本構造　217
図 5-2　インスタンス文書の基本構造　230
図 5-3　タクソノミー文書の基本構造　236
図 5-4　インスタンス文書の基本構造　257
図 5-5　<xbrli:context> 要素および <xbrli:unit> 要素の構造　258
図 6-1　インスタンス文書のオブジェクトモデルを表現した UML のクラス図（全体）　278
図 6-2　インスタンス文書のオブジェクトモデルを表現した UML のクラス図（コンテキスト要素部分）　279
図 6-3　タクソノミー文書のオブジェクトモデルを表現した UML のクラス図　280
図 6-4　インスタンス文書のオブジェクトモデルを表現した UML のクラス図（全体）　281
図 6-5　インスタンス文書のオブジェクトモデルを表現した UML のクラス図（コンテキスト要素部分）　282

図 6-6　タクソノミー文書のオブジェクトモデルを表現した UML のクラス図　284
図 6-7　タクソノミー文書の XBRL 文書管理機能のオブジェクトモデルを表現した UML のクラス図　285
図 6-8　単純なタクソノミー文書のオブジェクトモデルを表現した UML のクラス図　286
図 6-9　単純なタクソノミー文書の XBRL 文書管理機能のオブジェクトモデルを表現した UML のクラス図　287
図 6-10　ノードの表す各インターフェースの継承関係（XBRL2.0 の場合）　288
図 6-11　ノードの表す各インターフェースの継承関係（XBRL2.1 の場合）　289
図 6-12　リンクの構造を表す XBRLRelationship インターフェースの継承関係　290
図 6-13　単純なタクソノミー文書での SimpleNode インターフェースの継承関係　290
図 6-14　アクティブになるノードの範囲　603
図 6-15　XBRLIllegalCycleException.getErrorRoutes() メソッドが返す配列の例　710
図 6-16　XBRLDimensionIllegalCycleException.getErrorRoutes() メソッドが返す配列の例　714
図 6-17　一つの要素に対して同じ言語とロールを持つラベルが複数関連づけられている場合の例　805

表目次

表 1-1	XBRL で使用できるリンクベース	4
表 2-1	拡張リンクの構文規則	31
表 2-2	脚注リンクの構文規則	43
表 2-3	DTS 読み込み時に参照される要素と属性の一覧（XBRL2.0 の場合）	45
表 2-4	DTS 読み込み時に参照される要素と属性の一覧（XBRL2.1 の場合）	49
表 2-5	リンクベースの遅延読み込み時に最初に参照される要素と属性の一覧（XBRL2.0 の場合）	56
表 2-6	リンクベースの遅延読み込み時に最初に参照される要素と属性の一覧（XBRL2.1 の場合）	57
表 2-7	アークロールの目的	60
表 2-8	アーク型要素の属性一覧とリンクの上書きに必要な条件	69
表 2-9	拡張リンクの種類と対応するインターフェース	72
表 2-10	PTVLI の <ptv:arc> 要素の属性	74
表 2-11	インスタンス文書から PTVI へ含まれる要素とその内容	77
表 3-1	一元管理される ID	95
表 3-2	自動採番される ID と XBRL のバージョン	97
表 3-3	ID のフォーマットの出力内容	98
表 4-1	アプリケーションの開発時に参照する内容（XBRL2.0，XBRL2.1 の場合）	113
表 4-2	アプリケーションの開発時に参照する内容（XBRL Dimensions1.0 の場合）	113
表 4-3	タクソノミー文書の読み込みで使用するクラスおよびインターフェース	117
表 4-4	タクソノミー本体とリンクベースの参照で使用するインターフェース	119
表 4-5	インスタンス文書の新規作成で使用するクラスおよびインターフェース	122
表 4-6	非数値コンテキストの値	125
表 4-7	非数値コンテキスト要素の追加で使用するクラスおよびインターフェース	126
表 4-8	数値コンテキストの値	127
表 4-9	数値コンテキスト要素の追加で使用するクラスおよびインターフェース	128
表 4-10	非数値コンテキストに対応したアイテムの値	130
表 4-11	非数値コンテキストに対応したアイテムの追加で使用するインターフェース	130
表 4-12	数値コンテキストに対応したアイテムの値	131
表 4-13	数値コンテキストに対応したアイテムの追加で使用するインターフェース	131
表 4-14	インスタンス文書の保存で使用するインターフェース	132
表 4-15	インスタンス文書の読み込みで使用するクラスおよびインターフェース	134
表 4-16	脚注リンクを追加する非数値アイテムの値	136

表 4-17　追加する <link:footnote> 要素　136
表 4-18　追加される <link:loc> 要素および <link:footnoteArc> 要素　136
表 4-19　脚注リンクの追加で使用するクラスおよびインターフェース　137
表 4-20　タクソノミー文書の読み込みで使用するクラスおよびインターフェース　139
表 4-21　タクソノミー本体とリンクベースの参照で使用するインターフェース　141
表 4-22　タクソノミー本体とリンクベースの参照で使用するインターフェース　143
表 4-23　インスタンス文書の新規作成で使用するクラスおよびインターフェース　146
表 4-24　コンテキスト要素　149
表 4-25　コンテキスト要素の追加で使用するインターフェース　149
表 4-26　単位要素　151
表 4-27　単位要素の追加で使用するクラスおよびインターフェース　151
表 4-28　追加する非数値アイテムの値　152
表 4-29　非数値アイテムの追加で使用するインターフェース　152
表 4-30　数値アイテムの値　153
表 4-31　数値コンテキストに対応したアイテムの追加で使用するクラスおよびインターフェース　153
表 4-32　インスタンス文書の保存で使用するインターフェース　155
表 4-33　ID の採番で使用するクラスおよびインターフェース　156
表 4-34　インスタンス文書の読み込みで使用するクラスおよびインターフェース　160
表 4-35　脚注リンクを追加する非数値アイテムの値　162
表 4-36　追加する <link:footnote> 要素　162
表 4-37　追加される <link:loc> 要素，および <link:footnoteArc> 要素　162
表 4-38　脚注リンクの追加で使用するインターフェース　163
表 4-39　タクソノミー文書の拡張で使用するクラスおよびインターフェース　165
表 4-40　アイテム定義を追加する際に使用するインターフェース　168
表 4-41　ラベルを追加する際に使用するクラスおよびインターフェース　168
表 4-42　既存の関係の禁止や新たな関係の追加で使用するクラスおよびインターフェース　170
表 4-43　タクソノミー文書の保存で使用するインターフェース　172
表 4-44　タクソノミー文書の読み込みで使用するクラスおよびインターフェース　174
表 4-45　タクソノミー本体とリンクベースの参照で使用するインターフェース　176
表 4-46　多次元データを含んだコンテキスト要素　179
表 4-47　多次元データを含んだコンテキスト要素の追加で使用するクラスおよびインターフェース　180
表 4-48　プライマリー・アイテムの追加で使用するクラスおよびインターフェース　183
表 4-49　インスタンス文書の読み込みで使用するクラスおよびインターフェース　186
表 4-50　インスタンス文書とリンクベースの参照で使用するインターフェース　187

表 4-51　サンプルアプリケーションの詳細　194
表 4-52　必要なスキーマ定義とファイル名　198
表 4-53　プロパティの指定方法と優先順位　199
表 4-54　メモリサイズの設定例　212
表 5-1　エレメント要素の属性（アイテム型要素）　218
表 5-2　エレメント要素の属性（タプル型要素）　219
表 5-3　<link:linkbaseRef> 要素の属性　220
表 5-4　xlink:role 属性のリンク種別ごとの推奨値　221
表 5-5　拡張リンク型要素の共通属性　222
表 5-6　<link:loc> 要素の属性　223
表 5-7　<link:label> 要素の属性　224
表 5-8　<link:reference> 要素の属性　224
表 5-9　<link:presentationArc> 要素の属性　225
表 5-10　<link:calculationArc> 要素の属性　225
表 5-11　<link:definitionArc> 要素の属性　226
表 5-12　<link:labelArc> 要素の属性　227
表 5-13　<link:referenceArc> 要素の属性　228
表 5-14　アイテム型要素の検証内容　231
表 5-15　<xbrli:period> 要素の子要素　231
表 5-16　<link:footnoteLink> 要素の属性　232
表 5-17　<link:loc> 要素の属性　233
表 5-18　<link:footnote> 要素の属性　233
表 5-19　<link:footnoteArc> 要素の属性　234
表 5-20　エレメント要素の属性（アイテム型要素）　237
表 5-21　エレメント要素の属性（タプル型要素）　238
表 5-22　エレメント要素の属性（ハイパーキューブ型要素）　239
表 5-23　エレメント要素の属性（ディメンジョン型要素）　240
表 5-24　<link:roleType> 要素の属性　241
表 5-25　<link:arcroleType> 要素の属性　242
表 5-26　<link:linkbaseRef> 要素の属性　243
表 5-27　xlink:role 属性のリンク種別ごとの標準値　243
表 5-28　<link:linkbase> 要素の属性　244
表 5-29　<link:roleRef> 要素の属性　245
表 5-30　<link:arcroleRef> 要素の属性　246
表 5-31　拡張リンク型要素の共通属性　247

表 5-32 <link:loc> 要素の属性 248
表 5-33 <link:label> 要素の属性 249
表 5-34 <link:reference> 要素の属性 250
表 5-35 <link:presentationArc> 要素の属性 251
表 5-36 <link:calculationArc> 要素の属性 251
表 5-37 <link:definitionArc> 要素の属性 252
表 5-38 xlink:arcrole 属性値と <link:definitionArc> 要素の属性の関係 253
表 5-39 <link:labelArc> 要素の属性 254
表 5-40 <link:referenceArc> 要素の属性 255
表 5-41 <link:schemaRef> 要素の属性 260
表 5-42 XBRL2.0 と XBRL2.1 のコンテキスト要素の変更点 261
表 5-43 廃止された <xbrli:numericContext> 要素の属性 261
表 5-44 <xbrli:period> 要素の子要素 262
表 5-45 XBRL2.0 と XBRL2.1 の単位要素の属性，および子要素の変更点 262
表 5-46 XBRL2.0 と XBRL2.1 のアイテム型要素の変更点 263
表 5-47 <link:footnoteLink> 要素の属性 264
表 5-48 <link:loc> 要素の属性 265
表 5-49 <link:footnote> 要素の属性 265
表 5-50 <link:footnoteArc> 要素の属性 266
表 5-51 <xbrldi:typedMember> 要素の属性 267
表 5-52 <xbrldi:explicitMember> 要素の属性 267
表 5-53 XBRL データ型と，対応する XBRL DOM で保持するデータ型クラス 268
表 6-1 パッケージの概要 276
表 6-2 名前空間接頭辞と名前空間 URI との対応（XBRL2.0 の場合） 276
表 6-3 名前空間接頭辞と名前空間 URI との対応（XBRL2.1 の場合） 277
表 6-4 com.hitachi.xbrl パッケージのインターフェースの概要 390
表 6-5 com.hitachi.xbrl パッケージのクラスの概要 390
表 6-6 com.hitachi.xbrl パッケージの例外の概要 390
表 6-7 XBRLParseErrorHandler インターフェースのメソッドの概要 391
表 6-8 XBRLDocumentManager クラスのメソッドの概要 393
表 6-9 XBRLDOMException クラスのメソッドの概要 398
表 6-10 XBRLDOMRuntimeException クラスのメソッドの概要 399
表 6-11 XBRLDOMUserException クラスのコンストラクタの概要 400
表 6-12 XBRLDimensionException クラスのフィールドの概要 402
表 6-13 XBRLDimensionException クラスのメソッドの概要 404

表 6-14 XBRLDimensionUserException クラスのコンストラクタの概要 406
表 6-15 com.hitachi.xbrl.instance パッケージのインターフェース 409
表 6-16 XBRLArcroleRef インターフェースのメソッドの概要 411
表 6-17 XBRLContext インターフェースのメソッドの概要 413
表 6-18 設定できる時期要素の内容に関する制限 416
表 6-19 XBRLEntity インターフェースのメソッドの概要 420
表 6-20 XBRLFootnote インターフェースのメソッドの概要 422
表 6-21 XBRLIdentifier インターフェースのメソッドの概要 429
表 6-22 インスタンス文書の新規作成時に初期状態で設定される名前空間宣言（XBRL2.0 の場合） 432
表 6-23 インスタンス文書の新規作成時に初期状態で設定される名前空間宣言（XBRL2.1 の場合） 432
表 6-24 XBRLInstanceDocument インターフェースのメソッドの概要 434
表 6-25 ノードのインポート 499
表 6-26 XBRLInstanceDocumentBuilder インターフェースのメソッドの概要 507
表 6-27 XBRLInstanceDocumentBuilderFactory インターフェースのメソッドの概要 536
表 6-28 XBRLInstanceDocumentWriter インターフェースのメソッドの概要 550
表 6-29 XBRLInstanceDocumentWriterFactory インターフェースのメソッドの概要 554
表 6-30 XBRLInstanceElement インターフェースのフィールドの概要 555
表 6-31 XBRLInstanceElement インターフェースのメソッドの概要 556
表 6-32 XBRLInstanceItem インターフェースのフィールドの概要 581
表 6-33 XBRLInstanceItem インターフェースのメソッドの概要 584
表 6-34 アイテムのベースデータ型による単位要素の制約 589
表 6-35 getValue() メソッドが返すオブジェクト 593
表 6-36 setValue() メソッドの引数に指定できるオブジェクト 595
表 6-37 XBRLInstanceNode インターフェースのフィールドの概要 604
表 6-38 XBRLInstanceNode インターフェースのメソッドの概要 605
表 6-39 利用できるノードの型と XBRL のバージョンの関係 606
表 6-40 getParentNode メソッドが返す親ノードの型 607
表 6-41 ノードの複製処理 609
表 6-42 getW3CDOMNode メソッドが返す org.w3c.dom.Node オブジェクト 613
表 6-43 org.w3c.dom.Node オブジェクトに対して許可される更新操作 614
表 6-44 XBRLInstanceTuple インターフェースのメソッドの概要 616
表 6-45 XBRLLinkbaseRef インターフェースのメソッドの概要 623
表 6-46 XBRLMeasure インターフェースのメソッドの概要 624
表 6-47 XBRLMeasureOrOperator インターフェースのフィールドの概要 626

表 6-48	XBRLMeasureOrOperator インターフェースのメソッドの概要	626
表 6-49	XBRLNonNumericContext インターフェースのメソッドの概要	627
表 6-50	XBRLNumericContext インターフェースのメソッドの概要	628
表 6-51	XBRLOperator インターフェースのフィールドの概要	631
表 6-52	XBRLOperator インターフェースのメソッドの概要	631
表 6-53	XBRLPeriod インターフェースのフィールドの概要	634
表 6-54	XBRLPeriod インターフェースのメソッドの概要	634
表 6-55	利用できる時期要素の内容の意味と XBRL のバージョンの関係	635
表 6-56	XBRLRoleRef インターフェースのメソッドの概要	638
表 6-57	XBRLScenario インターフェースのメソッドの概要	640
表 6-58	XBRLSchemaRef インターフェースのメソッドの概要	651
表 6-59	XBRLSegment インターフェースのメソッドの概要	652
表 6-60	XBRLUnit インターフェースのメソッドの概要	664
表 6-61	XBRLUnit2 インターフェースのフィールドの概要	665
表 6-62	XBRLUnit2 インターフェースのメソッドの概要	666
表 6-63	XBRLDimensionMember インターフェースのメソッドの概要	671
表 6-64	XBRLExplicitMember インターフェースのメソッドの概要	672
表 6-65	XBRLTypedMember インターフェースのメソッドの概要	674
表 6-66	com.hitachi.xbrl.link パッケージのインターフェース	675
表 6-67	com.hitachi.xbrl.link パッケージの例外の概要	677
表 6-68	XBRLCalculationRelationship インターフェースのメソッドの概要	677
表 6-69	XBRLPresentationRelationship インターフェースのメソッドの概要	679
表 6-70	XBRLRelationship インターフェースのフィールドの概要	680
表 6-71	XBRLRelationship インターフェースのメソッドの概要	681
表 6-72	アークの種類とインターフェースの関係	681
表 6-73	アークの種類と from 側のリソースの関係	683
表 6-74	アークの種類と to 側のリソースの関係	684
表 6-75	XBRLDimensionalRelationshipSet インターフェースのメソッドの概要	686
表 6-76	XBRLDimensionDomainRelationship インターフェースのメソッドの概要	698
表 6-77	XBRLDomainMemberRelationship インターフェースのメソッドの概要	699
表 6-78	XBRLDRSDimension インターフェースのメソッドの概要	701
表 6-79	XBRLDRSMember インターフェースのメソッドの概要	703
表 6-80	XBRLHasHypercubeRelationship インターフェースのフィールドの概要	704
表 6-81	XBRLHasHypercubeRelationship インターフェースのメソッドの概要	704
表 6-82	<link:definitionArc> 要素の xbrldt:contextElement 属性の種類	705

表 6-83 <link:definitionArc> 要素の xlink:arcrole 属性の種類 706
表 6-84 XBRLHypercubeDimensionRelationship インターフェースのメソッドの概要 707
表 6-85 XBRLIllegalCycleException クラスのメソッドの概要 708
表 6-86 XBRLDimensionIllegalCycleException クラスのメソッドの概要 711
表 6-87 com.hitachi.xbrl.taxonomy パッケージのインターフェース 715
表 6-88 XBRLArcroleType インターフェースのメソッドの概要 717
表 6-89 XBRLLabel インターフェースのフィールドの概要 719
表 6-90 XBRLLabel インターフェースのメソッドの概要 720
表 6-91 XBRLLinkbase インターフェースのフィールドの概要 724
表 6-92 XBRLLinkbase インターフェースのメソッドの概要 724
表 6-93 XBRLTaxonomyLinkbaseRef インターフェースのメソッドの概要 725
表 6-94 XBRLPart インターフェースのメソッドの概要 727
表 6-95 XBRLReference インターフェースのメソッドの概要 728
表 6-96 XBRLTaxonomyRoleRef インターフェースのメソッドの概要 733
表 6-97 XBRLRoleType インターフェースのメソッドの概要 734
表 6-98 XBRLTaxonomyAttributeDeclaration インターフェースのメソッドの概要 736
表 6-99 XBRLTaxonomyBody インターフェースのメソッドの概要 738
表 6-100 XBRL XBRLTaxonomyComplexTypeDefinition インターフェースのフィールドの概要 741
表 6-101 XBRLTaxonomyComplexTypeDefinition インターフェースのメソッドの概要 742
表 6-102 XBRLTaxonomyComponent インターフェースのフィールドの概要 746
表 6-103 XBRLTaxonomyComponent インターフェースのメソッドの概要 747
表 6-104 XBRLTaxonomyDocument インターフェースのメソッドの概要 748
表 6-105 XBRLTaxonomyDocumentBuilder インターフェースのメソッドの概要 771
表 6-106 XBRLTaxonomyDocumentBuilderFactory インターフェースのメソッドの概要 780
表 6-107 XBRLTaxonomyElement インターフェースのフィールドの概要 796
表 6-108 XBRLTaxonomyElement インターフェースのメソッドの概要 796
表 6-109 XBRLTaxonomyItem インターフェースのフィールドの概要 814
表 6-110 XBRLTaxonomyItem インターフェースのメソッドの概要 816
表 6-111 利用できるアイテムのデータ型と XBRL のバージョンの関係 817
表 6-112 アイテムのデータ型と isNumeric() メソッドの戻り値との対応関係 819
表 6-113 XBRLTaxonomyNode インターフェースのフィールドの概要 822
表 6-114 XBRLTaxonomyNode インターフェースのメソッドの概要 823
表 6-115 利用できるノードの型と XBRL のバージョンの関係 824
表 6-116 getParentNode メソッドが返す親ノードの型 825
表 6-117 getW3CDOMNode() メソッドが返す org.w3c.dom.Node オブジェクト 826

表 6-118 XBRL XBRLTaxonomyTypeDefinition インターフェースのフィールドの概要 829
表 6-119 XBRLTaxonomyTypeDefinition インターフェースのメソッドの概要 829
表 6-120 XBRLTaxonomyDimensionItem インターフェースのメソッドの概要 833
表 6-121 com.hitachi.xbrl.types パッケージのクラス一覧 835
表 6-122 XBRLAbstractBinary クラスのメソッドの概要 836
表 6-123 XBRLAbstractPrecision クラスのフィールドの概要 839
表 6-124 XBRLAbstractPrecision クラスのメソッドの概要 839
表 6-125 XBRLBase64Binary クラスのコンストラクタの概要 843
表 6-126 XBRLBase64Binary クラスのメソッドの概要 844
表 6-127 XBRLDate クラスのコンストラクタの概要 849
表 6-128 XBRLDate クラスのメソッドの概要 850
表 6-129 XBRLDateTime クラスのコンストラクタの概要 854
表 6-130 XBRLDateTime クラスのメソッドの概要 854
表 6-131 XBRLDateUnion クラスのフィールドの概要 859
表 6-132 XBRLDateUnion クラスのコンストラクタの概要 859
表 6-133 XBRLDateUnion クラスのメソッドの概要 859
表 6-134 XBRLDecimals クラスのコンストラクタの概要 864
表 6-135 XBRLDecimals クラスのメソッドの概要 864
表 6-136 XBRLDuration クラスのコンストラクタの概要 866
表 6-137 XBRLDuration クラスのメソッドの概要 866
表 6-138 XBRLFraction クラスのコンストラクタの概要 869
表 6-139 XBRLFraction クラスのメソッドの概要 869
表 6-140 XBRLGDay クラスのコンストラクタの概要 873
表 6-141 XBRLGDay クラスのメソッドの概要 873
表 6-142 XBRLGMonth クラスのコンストラクタの概要 875
表 6-143 XBRLGMonth クラスのメソッドの概要 875
表 6-144 XBRLGMonthDay クラスのコンストラクタの概要 878
表 6-145 XBRLGMonthDay クラスのメソッドの概要 878
表 6-146 XBRLGYear クラスのコンストラクタの概要 880
表 6-147 XBRLGYear クラスのメソッドの概要 880
表 6-148 XBRLGYearMonth クラスのコンストラクタの概要 883
表 6-149 XBRLGYearMonth クラスのメソッドの概要 883
表 6-150 XBRLHexBinary クラスのコンストラクタの概要 885
表 6-151 XBRLHexBinary クラスのメソッドの概要 886
表 6-152 XBRLPrecision クラスのコンストラクタの概要 891

表 6-153	XBRLPrecision クラスのメソッドの概要	891
表 6-154	XBRLQName クラスのコンストラクタの概要	894
表 6-155	XBRLQName クラスのメソッドの概要	894
表 6-156	XBRLTime クラスのコンストラクタの概要	901
表 6-157	XBRLTime クラスのメソッドの概要	901
表 6-158	com.hitachi.xbrl.util パッケージのインターフェース	904
表 6-159	com.hitachi.xbrl.util パッケージのクラス一覧	904
表 6-160	XBRLIdCreator インターフェースのフィールドの概要	905
表 6-161	XBRLIdCreator クラスのメソッドの概要	906
表 6-162	ID の自動生成が必要な要素と，type に指定する値	906
表 6-163	XBRLDocumentWriter インターフェースのメソッドの概要	907
表 6-164	XBRLDocumentWriterFactory インターフェースのメソッドの概要	911
表 6-165	XBRLConstants クラスのフィールドの概要	912
表 6-166	XBRLIdManager クラスのコンストラクタの概要	919
表 6-167	XBRLIdManager クラスのメソッドの概要	919
表 6-168	XBRLNamespaceDeclaration クラスのコンストラクタの概要	923
表 6-169	XBRLNamespaceDeclaration クラスのメソッドの概要	924
表 6-170	XBRLSchemaLocation クラスのコンストラクタの概要	926
表 6-171	XBRLSchemaLocation クラスのメソッドの概要	926
表 6-172	PTVInstance クラスのメソッドの概要	929
表 6-173	PTVLinkbase クラスのメソッドの概要	930
表 6-174	XBRLEqualUtil クラスのコンストラクタの概要	930
表 6-175	XBRLEqualUtil クラスのメソッドの概要	931
表 6-176	com.hitachi.xbrl.simple.taxonomy パッケージのインターフェースの概要	937
表 6-177	SimpleEdge インターフェースのフィールドの概要	938
表 6-178	SimpleEdge インターフェースのメソッドの概要	938
表 6-179	SimpleElementDefinition インターフェースのメソッドの概要	941
表 6-180	getSubstitutionGroup の要素定義と戻り値	945
表 6-181	SimpleItemDefinition インターフェースのフィールドの概要	946
表 6-182	SimpleItemDefinition インターフェースのメソッドの概要	949
表 6-183	SimpleLabel インターフェースのメソッドの概要	952
表 6-184	SimpleNode インターフェースのフィールドの概要	957
表 6-185	SimpleNode インターフェースのメソッドの概要	957
表 6-186	SimpleTaxonomy インターフェースのフィールドの概要	959
表 6-187	SimpleTaxonomy インターフェースのメソッドの概要	960

表 6-188	SimpleTaxonomyBuilder インターフェースのメソッドの概要	973
表 6-189	SimpleTaxonomyBuilderFactory インターフェースのメソッドの概要	975
表 A-1	XBRL 文書共通の検証	1151
表 A-2	別文書と関係する場合の検証	1152
表 A-3	インスタンス文書の検証	1155
表 A-4	アイテムの型と，対応する <xbrli:measure> 要素の値	1160
表 A-5	インスタンス文書の完全な妥当性検証	1160
表 A-6	タクソノミー文書の検証	1162
表 A-7	タクソノミー文書の完全な妥当性検証	1166
表 A-8	リンクベースおよびリンクの検証	1167
表 A-9	リンクベースの完全な妥当性検証	1168
表 B-1	新しく作成する <link:footnoteLink> 要素の各属性	1170
表 B-2	変更する <link:footnote> 要素の各属性	1171
表 B-3	新しく作成する <link:loc> 要素の各属性	1171
表 B-4	<link:footnote> 要素から <link:loc> 要素への <link:footnoteArc> 要素	1171
表 B-5	<link:loc> 要素から <link:footnote> 要素への <link:footnoteArc> 要素	1172
表 B-6	新しく作成する <link:footnoteLink> 要素の各属性	1173
表 B-7	変更する <link:footnote> 要素の各属性	1173
表 B-8	新しく作成する <link:loc> 要素の各属性	1174
表 B-9	新しく作成する <link:footnoteArc> 要素の各属性	1174
表 C-1	uCosminexus Business Reporting Processor の動作の差異（04-00 と 05-00 の差異）	1175
表 C-2	uCosminexus Business Reporting Processor の動作の差異（03-00 と 04-00 の差異）	1175

1 uCosminexus Business Reporting Processor の概要

uCosminexus Business Reporting Processor は，XBRL（eXtensible Business Reporting Language）に準拠した文書を扱うアプリケーションを開発するための製品です。
この章では，XBRL の概要，uCosminexus Business Reporting Processor の特長，および前提プログラムを含めたシステムでの位置づけについて説明します。

1.1　XBRL の概要

1.2　uCosminexus Business Reporting Processor の特長

1.3　uCosminexus Business Reporting Processor の位置づけ

1.1 XBRL の概要

XBRL（eXtensible Business Reporting Language）は，企業の財務情報を記述する標準言語です。XBRL は，XML の技術を使用しています。XBRL International が XBRL の仕様を策定し，規格として公開しています。XBRL は，すでに公的機関や企業で導入され始めていて，財務報告に欠かせない技術になると注目されています。

企業の財務情報は，日々の仕訳に始まり，会計，納税，決算公告など，多数の目的に使用されます。決算公告では，企業の経営状態を判断する材料となる財務諸表を，多くの投資家や金融機関，監査法人などに開示します。

XBRL のメリットには，財務情報を開示する企業にとって迅速な決算公告が実現できること，および開示された情報を利用する投資家や金融機関，監査法人などにとって，データの再利用ができることがあります。

迅速で正確な決算報告を実現

迅速で正確な決算報告は，企業経営にとって避けて通れない課題です。XBRL を使用すると，企業は，日々の仕訳から決算報告までを迅速に進められるようになります。会計システムに蓄積されている財務データを XBRL 形式に変換することで，企業の財務情報をタイムリーに開示できます。

また，XBRL 形式の財務データを利用して，公共機関への納税や金融機関への融資申請などに必要な書類を容易に作成できます。

データの再利用を実現

今までは，各企業が独自のシステムを使用して財務情報を開示していたため，開示される財務諸表のデータ形式が不統一でした。企業の財務情報を利用する金融機関などは，入手したデータの手入力が必要となり，データの再利用には，多大な労力と時間が必要でした。

XBRL を導入することで，入手した各企業の財務データを，さらに適切な形式に加工して，複数企業の比較分析や税務を扱うシステムで再利用できます。

ここでは，XBRL で記述された文書の概要について説明します。

1.1.1 XBRL 文書

XBRL で記述された文書を XBRL 文書と呼びます。ここでは，XBRL 文書の構成について説明します。

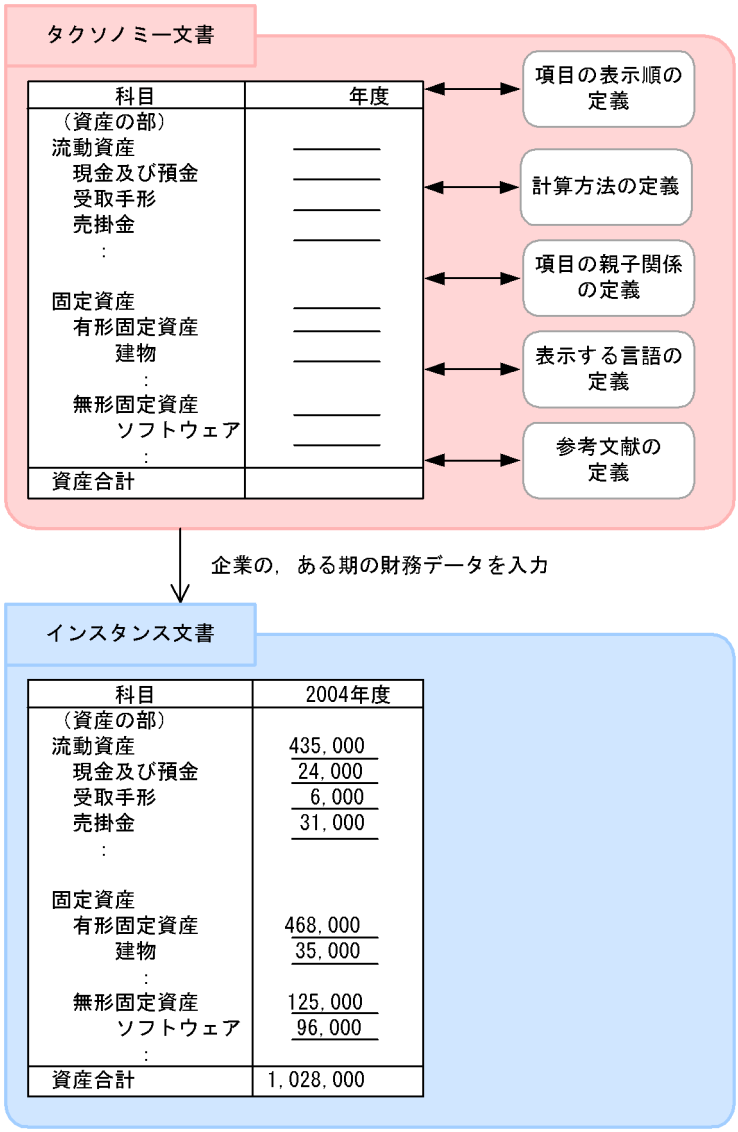
XBRL 文書は，タクソノミー文書とインスタンス文書から構成されます。タクソノミー文書は，文書の書式や記載する項目，計算方法などを定義した XML スキーマです。インスタンス文書は，定義されたスキーマ（タクソノミー文書）に従って実際の値を入力した XML 文書です。

XBRL の仕様には複数のバージョンがあり，バージョンによって作成される XBRL 文書

が異なります。uCosminexus Business Reporting Processor では，XBRL 2.0 および XBRL 2.1 に対応しています。また，XBRL2.1 の拡張規格である XBRL Dimensions1.0 にも対応しています。XBRL Dimensions1.0 では，財務報告の情報を多次元で表現できます。

タクソノミー文書とインスタンス文書の関係を次の図に示します。

図 1-1　タクソノミー文書とインスタンス文書の関係



財務諸表を例に挙げると，財務諸表の書式，記載する勘定科目やその計算式，根拠となる会計基準などの参考文献をまとめたひな型がタクソノミー文書に該当し，企業のある期の財務データを入力して作成した財務諸表がインスタンス文書に該当します。企業は，

財務諸表用のタクソノミー文書で定義した内容に従って，企業のある期のデータを基に財務諸表を作成します。

1.1.2 タクソノミー文書とは

タクソノミー文書は，勘定科目を表すアイテムおよび計算式を表す計算リンクなど，インスタンス文書で記載する内容を定義したひな型です。

タクソノミー文書は，**タクソノミー本体**と**リンクベース**から構成されます。財務諸表を例に挙げると，財務諸表で記載する勘定科目を定義するのがタクソノミー本体，勘定科目の表示順，計算式，会計基準などの参考文献を定義するのがリンクベースに該当します。

タクソノミー文書では，国や業種，会計基準によって異なる内容を別文書（リンクベース）として切り分け，それぞれをリンクさせています。別文書に分かれているため，会計基準などの変更に対応しやすく，各企業の事情に合わせた形式にカスタマイズできます。

タクソノミー本体

タクソノミー本体は，XML Schema 言語を使用して，インスタンス文書に記述するアイテムを定義した XML スキーマです。

アイテムとは報告内容の各勘定科目のことで，数値と非数値があります。また，複数の項目をまとめて解釈する必要がある場合には，アイテムのまとまりを示す**タプル**も定義できます。

タクソノミー本体は，XBRL2.1 仕様ではタクソノミースキーマ（Taxonomy Schema）と記載されていますが，このマニュアルでは，タクソノミー本体と記載しています。

リンクベース

リンクベースは，XLink や XPath の技術を使用して，タクソノミー本体に定義したアイテム間の関係などを記述した XML 文書です。XBRL では，次のリンクベースが使用できます。

表 1-1 XBRL で使用できるリンクベース

リンク種別		定義内容
関係リンク	プレゼンテーションリンク	アイテムまたはタプルの表示順序を定義します。
	計算リンク	アイテムまたはタプルの数値データの重みつき加減算式を定義します。
	定義リンク	アイテムの親子関係など，アイテムまたはタプル同士の関係を定義します。
ラベルリンク		アイテムまたはタプルの表示名称（ラベル）を定義します。日本語または英語などを指定して，さまざまな言語でラベルを記述したり，別名を定義したりできます。

リンク種別	定義内容
リファレンスリンク	参考文献を定義します。会計基準など，根拠となる文献への参照に使用します。

タクソノミー本体とリンクベースの参照関係

リンクベースは，タクソノミー本体に記述することもできますが，多くの場合は別文書に記述します。

タクソノミー本体からリンクベースへの参照は，XLink の単純リンクが使用され，リンクベースからタクソノミー本体への参照は XLink の拡張リンク（ロケータ型要素）が使用されます。

ロケータ型要素では，文書内の特定の項目を参照できるように，URI にフラグメントを付けます。フラグメントの指定には，XPointer のショートハンドポインタ（Shorthand Pointer）を使用できます。XBRL2.1 の場合は，スキームベースポインタ（Scheme-Based Pointer）の element() スキームも指定できます。

1.1.3 インスタンス文書とは

インスタンス文書は，タクソノミー文書に従って実際の財務データを入力した文書です。財務諸表を例に挙げると，インスタンス文書は，企業がある時期の財務状態を報告した財務諸表に該当します。インスタンス文書には各勘定科目に具体的な数値が記入されていて，企業の財務状態が一目でわかるようになっています。

インスタンス文書には，インスタンス文書に特化したリンクベースや脚注リンクを追加できます。脚注リンクは，通常の脚注と同様に，インスタンス文書に記述した事実間の不規則な関係を表現します。

インスタンス文書とタクソノミー本体の参照関係

XBRL2.0 の場合，インスタンス文書からタクソノミー本体への参照は，ルート要素の xsi:schemaLocation 属性を使用して参照します。

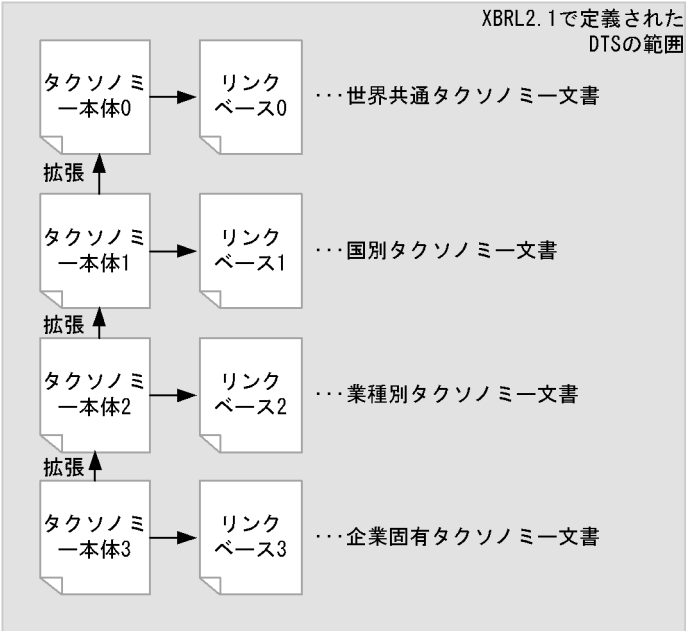
XBRL2.1 の場合，XLink の単純リンクである <link:schemaRef> 要素を使用して参照します。

1.1.4 タクソノミー文書の拡張

XBRL では，既存のタクソノミー文書に，勘定科目を追加したり変更したりできます。これをタクソノミー文書の拡張と呼びます。タクソノミー文書を拡張すると，一般に公開されているタクソノミー文書を流用して，独自の記載形式を定義できます。ここでは，タクソノミー文書の拡張と，タクソノミー文書の階層構造について説明します。

タクソノミー文書の階層構造を次の図に示します。

図 1-2 タクソノミー文書の階層構造



タクソノミー文書の階層構造は，上位に世界共通タクソノミー文書があり，その下に国別タクソノミー文書，そして用途や業種ごとのタクソノミー文書，企業固有のタクソノミー文書などで構成されます。そのため，XBRL 文書を取り扱うときには，複数のタクソノミー文書を組み合わせて一つの論理的なタクソノミー文書とみなす必要があります。

このように既存のタクソノミー文書を拡張して，新しいタクソノミー文書を定義した場合の，論理的に一つのタクソノミー文書であると判断する範囲を DTS（Discoverable Taxonomy Set）と呼びます。XBRL2.0 仕様では DTS という用語は使用されていませんが，このマニュアルでは複数のタクソノミー文書の集まりを論理的な一つのタクソノミー文書としてみなす場合，DTS と記載します。

タクソノミー文書の拡張には，次の方法があります。

- タクソノミー本体で <xsd:import> 要素を使用して，ほかのタクソノミー文書をインポートする。
- インスタンス文書で，ほかのタクソノミー文書を取り込む。
  次の方法で，つながりのない複数のタクソノミー文書を一つの DTS として定義できます。
  XBRL2.0 の場合，インスタンス文書のルート要素の xsi:schemaLocation 属性に，複数のタクソノミー文書を指定します。
  XBRL2.1 の場合，タクソノミー文書ごとに <link:schemaRef> 要素を記述します。

1.2　uCosminexus Business Reporting Processor の特長

uCosminexus Business Reporting Processor は，次の機能を提供します。

- XBRL 文書を扱うために体系化したオブジェクトモデル（XBRL DOM）
- XBRL 文書を読み込んで XBRL DOM の生成や操作ができる API（Java）

ここでは，これらの機能を利用してできること，XBRL DOM の詳細，および uCosminexus Business Reporting Processor が準拠する規格について説明します。

1.2.1　uCosminexus Business Reporting Processor でできること

uCosminexus Business Reporting Processor が提供する API（Application Programming Interface）を使えば，タクソノミー文書やインスタンス文書の読み込みおよび作成をするためのアプリケーションを開発できます。アプリケーションの開発方法については「4.　アプリケーションの開発」を参照してください。API の種類については「6.　API リファレンス」を参照してください。

複雑な構造の XBRL 文書を容易に操作できる

XBRL は，XML Schema，XLink，XPath など XML のさまざまな技術を利用しています。これらの技術の詳細を理解した上でアプリケーションを開発するのは大変です。uCosminexus Business Reporting Processor が提供する API を使用すると，XML 技術の詳細な仕様を意識しなくても XBRL 文書の読み込み，作成，編集などをするアプリケーションを開発できます。

例えば，タクソノミー本体からリンクベースを参照するには，リンク構造を解析し，参照先を検索する複雑な処理が必要になります。uCosminexus Business Reporting Processor が提供する API を使用すれば，リンク先の内容を読み込んだり，編集したりという操作が容易にできます。

複数の XBRL バージョンに準拠した文書を扱える

XBRL の仕様には，複数のバージョンがあります。

uCosminexus Business Reporting Processor は，XBRL2.0 および XBRL2.1 に準拠した XBRL 文書に対応しています。アプリケーションでどちらの文書を扱うかを指定することで，それぞれの規格に準拠した文書を取り扱えます。

XBRL 文書を効率良く処理できる

インスタンス文書を作成するときには，対象となるタクソノミー文書を読み込む必要があります。タクソノミー文書は，タクソノミー本体と複数のリンクベースから構成され，さらに複数のタクソノミー文書の集まり（DTS）を読み込む場合もあります。

uCosminexus Business Reporting Processor では，XBRL 文書を効率良く処理するために次の機能を提供しています。

- 一度読み込んだ DTS をメモリにキャッシュできる
- 最初に DTS に定義されているリンクベースをすべて読み込まなくても，必要なときに使用するリンクベースだけを読み込むことができる

タクソノミー文書やインスタンス文書の内容が正しいかを検証できる

タクソノミー文書やインスタンス文書の内容が正しいか，プロパティを設定することで検証できます。さらに，**完全な妥当性検証**を実行すれば，文書が XBRL 2.1 Conformance Suite 1.0 に完全に準拠しているかを検証できます。

検証方法については，「2.3　タクソノミー文書の検証」または「2.4　インスタンス文書の検証」を参照してください。また，使用できるプロパティについては，「4.16.2　プロパティ一覧」を参照してください。

1.2.2　XBRL DOM とは

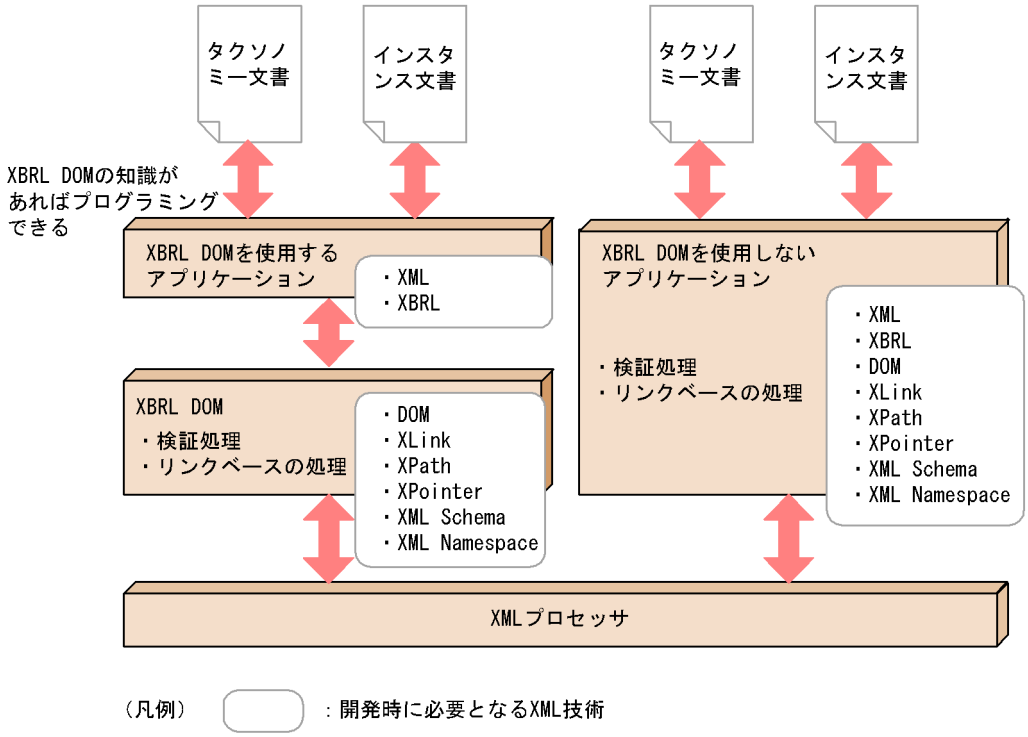
XBRL DOM とは，XBRL 文書を扱うために体系化したドキュメントオブジェクトモデルのことです。

ドキュメントオブジェクトモデルとは，文書の各要素をツリー構造で表現したもので，要素を移動させたり追加したりするための API を提供します。XML データのツリー構造を操作するための API として，W3C が規格化した DOM があります。このマニュアルでは，XBRL DOM と区別するため，W3C が規格化した DOM を W3C DOM と表記します。

XBRL DOM は，W3C DOM と比較すると，次の点で XBRL 文書の操作に適しています。

- 文書の複雑な構成を意識させない
  XBRL 文書を扱うときは，複数文書のまとまりを論理的に一つの文書とみなしたり，別文書に記述されている内容を参照させたりするなど，複雑な操作が必要になります。このような場合も，XBRL DOM を使用すれば，階層や文書の構成を意識することなく，文書を操作できます。
- XML 技術の難解さを意識させない
  XBRL 文書では，XML Schema，XLink，XPath などの XML 技術が使われています。XBRL DOM は XBRL 文書を操作するときに必要な API を用意しています。ユーザーは，アプリケーション開発時に XBRL DOM の API を使用するだけで，XML 技術の詳細を意識することなくアプリケーションを開発できます。
  XBRL DOM を使用した場合と，使用しない場合のアプリケーション開発を次の図に示します。

図 1-3　XBRL DOM を使用した場合と，使用しない場合のアプリケーション開発



XBRL DOM は，XML Schema の知識が必要となるスキーマ検証，XLink および XPath の知識が必要となるリンクベースの処理をすべて隠ぺいします。そのため，XBRL DOM を使用したアプリケーション開発では，ユーザーは XML，XBRL，および XBRL DOM の知識だけでアプリケーションを開発できます。

1.2.3　準拠する規格

uCosminexus Business Reporting Processor は，次の規格に準拠しています。

- eXtensible Business Reporting Language(XBRL) 2.0 Specification
- eXtensible Business Reporting Language(XBRL) 2.1 Specification
- XBRL Dimensions 1.0
- XBRL 2.1 Conformance Suite 1.0

また，これらの XBRL の規格は次の XML 関連技術を利用しています。

- Extensible Markup Language (XML) 1.0 (Second Edition)
- XML Schema Part 0: Primer
- XML Schema Part 1: Structures

- XML Schema Part 2: Datatypes
- XML Base
- Namespaces in XML
- XML Path Language (XPath) Version 1.0
- XPointer Framework
- XPointer Element() scheme
- XML Linking Language (XLink) Version 1.0
- Document Object Model (DOM) Level 2 Core Specification Version 1.0

XBRL の規格については，XBRL International のサイト（http://www.xbrl.org/）を参照してください。

1.3　uCosminexus Business Reporting Processor の位置づけ

ここでは，前提プログラムを含めた，システムでの uCosminexus Business Reporting Processor の位置づけについて説明します。

uCosminexus Business Reporting Processor を使用して開発したアプリケーションを実行する環境には，uCosminexus Business Reporting Processor の前提プログラムとして，XML プロセッサおよび XLink プロセッサが必要です。

XML プロセッサ

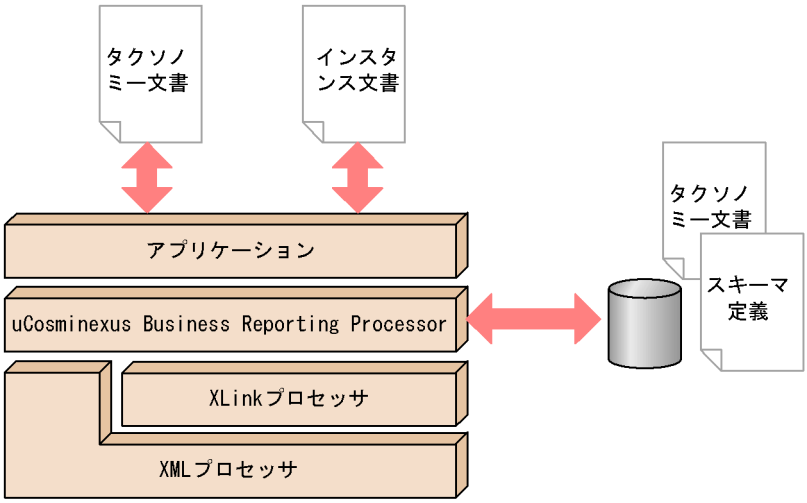
アプリケーションサーバを使用する場合は，uCosminexus Application Server が必要です。使用しない場合は，uCosminexus Developer が必要です。uCosminexus Developer は，XML プロセッサと Java 仮想マシンの機能を持つプログラムです。

XLink プロセッサ

uCosminexus XML Link Processor が必要です。

uCosminexus Business Reporting Processor の前提プログラムを含めたシステムでの位置づけを次の図に示します。

図 1-4　前提プログラムを含めたシステムでの位置づけ



アプリケーションでは，タクソノミー文書やインスタンス文書の読み込みおよび作成をするアプリケーションを開発できます。uCosminexus Business Reporting Processor は，タクソノミー文書やスキーマ定義を参照して，読み込んだ XBRL 文書を検証したり，タクソノミー文書やスキーマ定義に従った文書を作成したりします。

2 XBRL 文書の参照と検証

この章では，XBRL 文書の参照や検証時に実施される DTS の読み込み，リンクの上書き，およびリンク構造の参照について説明します。
DTS とは，複数のタクソノミー文書を一つの論理的なタクソノミー文書とみなす範囲のことです。DTS に読み込んだタクソノミー文書を，別のタクソノミー文書で拡張できます。これをリンクの上書きといいます。XBRL2.1 の場合は，リンクが上書きされたあと，DTS 内のタクソノミー本体やリンクベースとの関係を，ユーザーアプリケーションから容易に参照できる機能もあります。
また，この章では，完全な妥当性検証を実行した際に生成される，PTVLI と PTVI の構造と生成例についても説明します。

2.1 タクソノミー文書の参照

2.2 インスタンス文書の参照

2.3 タクソノミー文書の検証

2.4 インスタンス文書の検証

2.5 DTS の読み込み

2.6 リンクの上書き（XBRL2.0 の場合）

2.7 リンクの上書き（XBRL2.1 の場合）

2.8 リンク構造の参照（XBRL2.1 の場合）

2.9 PTVLI の生成（XBRL2.1 の場合）

2.10 PTVI の生成（XBRL2.1 の場合）

2.11 XBRL 同等性の判定

2.1 タクソノミー文書の参照

タクソノミー文書は，インスタンス文書に記載する項目を定義するためのタクソノミー本体と，定義した項目間の関係などを記述したリンクベースとで構成されています。

ここでは，タクソノミー文書の参照について，説明します。

タクソノミー文書を参照する場合，タクソノミー文書が対応する XBRL のバージョンを指定する必要があります。デフォルトでは 2.0 になるため，XBRL2.1 に対応する文書を操作する場合は，XBRL のバージョンを変更します。

また，タクソノミー文書が XBRL Dimensions に対応している場合，XBRL Dimensions のバージョンを指定する必要があります。その場合は，必ず XBRL のバージョンに 2.1 を指定してください。2.0 を指定したときは，タクソノミー文書は XBRL2.0 として処理されます。

指定方法

- XBRL のバージョンは，プロパティや API などで指定します。指定するプロパティを次に示します。
  cbrp.parse.XBRLVersion
- XBRL Dimensions のバージョンは，プロパティや API などで指定します。指定するプロパティを次に示します。
  cbrp.parse.DimensionsVersion

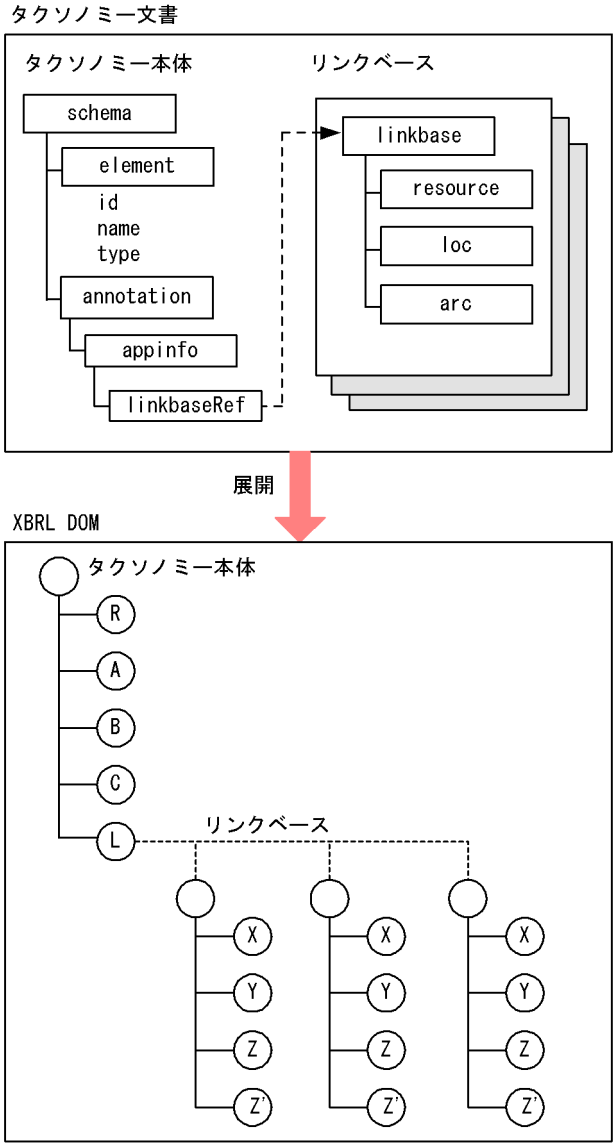
指定方法の詳細については，「4.16.1　プロパティの指定方法と優先順位」を参照してください。

2.1.1 タクソノミー文書の XBRL DOM ツリーへの展開

uCosminexus Business Reporting Processor は，タクソノミー文書を読み込み，XBRL DOM を使用したツリーに展開します。ユーザーアプリケーションから API を使用して XBRL DOM を操作することで，タクソノミー文書に定義された項目の一覧などを参照できます。

タクソノミー文書の参照とツリーへの展開を次の図に示します。

図 2-1　タクソノミー文書の参照とツリーへの展開



2.1.2　拡張されたタクソノミー文書の参照

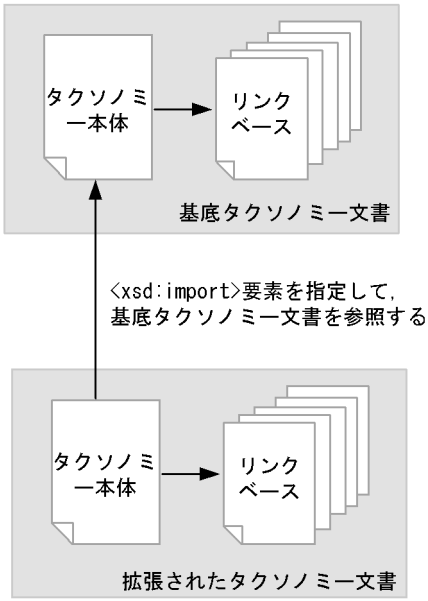
既存のタクソノミー文書を拡張し，新しいタクソノミー文書を定義することもできます。例えば，タクソノミー本体に <xsd:import> 要素を指定してほかのタクソノミー文書をインポートし，タクソノミー文書を拡張できます。新しいタクソノミー文書を定義するときに，基になったタクソノミー文書を**基底タクソノミー文書**と呼びます。

uCosminexus Business Reporting Processor では，タクソノミー文書が拡張されていた場合，基底タクソノミー文書も自動的に読み込み，拡張されたタクソノミー文書の集ま

り（DTS）として扱えます。物理的に複数のタクソノミー文書から構成されている場合でも，論理的な一つのタクソノミー文書として参照できます。

拡張されたタクソノミー文書の構成を，次の図に示します。

図 2-2　拡張されたタクソノミー文書の構成



この場合，タクソノミー文書で <xsd:import> 要素を使用して，基底タクソノミー文書を参照しています。

DTS を参照する場合，タクソノミー文書の読み込み範囲などをあらかじめ指定できます。DTS の読み込みに関する設定項目については，「2.5　DTS の読み込み」を参照してください。

2.1.3　タクソノミー文書の定義情報の参照

タクソノミー文書に記述された定義情報を参照できます。これは，タクソノミー文書のガイドラインとして 2005 年 4 月 25 日版の FRTA　1.0 （Financial Reporting Taxonomy Architecture）が勧告案として公開されており，この FRTA に準拠しているかどうかの検証を行うために必要な情報です（また，インスタンス文書のガイドラインについても，同様に仕様策定が進められており，2004 年 11 月 14 日版の FRIS　1.0（Financial Reporting Instance Standards）が公開作業ドラフトとして公開されています）。

このタクソノミー文書の定義情報を参照する場合，cbrp.parse.validate.frtafris のプロパティに true を指定してからタクソノミー文書を読み込ませて，FRTA/FRIS 用の解析を uCosminexus Business Reporting Processor にさせる必要があります。

指定方法

タクソノミー文書の定義情報を参照するかどうかは，プロパティまたは API で指定します。指定するプロパティを次に示します。

- FRTA/FRIS 用の解析
  cbrp.parse.validate.frtafris

指定方法の詳細については，「4.16.1　プロパティの指定方法と優先順位」を参照してください。

この指定により利用できるタクソノミー文書の定義情報とメソッドを下記に示します。メソッドの明記がないものは，そのインターフェースの全メソッドが対象となります。

● XBRLLinkbase インターフェース

getExtendedLinksW3CDOMNode(int extendedLinkType)

● XBRLTaxonomyLinkbaseRef インターフェース

● XBRLTaxonomyRoleRef インターフェース

● XBRLTaxonomyAttributeDeclaration インターフェース

● XBRLTaxonomyBody インターフェース

getElementFormDefault()

getAttributeFormDefault()

getImportBodies()

getIncludeBodies()

● XBRLTaxonomyComplexTypeDefinition インターフェース

● XBRLTaxonomyDocument インターフェース

getNamespaceDeclarations()

getLinkbaseRefs()

getRoleRefs()

getTaxonomyTypeDefinitions()

getNamespaceDeclarationsByPrefix(java.lang.String prefix)

getNamespaceDeclarationsByURI(java.lang.String uri)

getTaxonomyTypeDefinition(java.lang.String namespaceURI, java.lang.String name)

● XBRLTaxonomyElement インターフェース

getTaxonomyTypeDefinition()

- XBRLTaxonomySimpleTypeDefinition インターフェース
- XBRLTaxonomyTypeDefinition インターフェース

2.2 インスタンス文書の参照

インスタンス文書は，タクソノミー文書に従って，企業の財務情報の具体的な値が記述された文書です。uCosminexus Business Reporting Processor では，すでに作成されているインスタンス文書を読み込み，XBRL DOM を使用したツリーに展開します。ユーザーアプリケーションから API を使用して XBRL DOM を操作することで，既存のインスタンス文書を参照できます。

インスタンス文書を参照する場合，インスタンス文書中に指定されているタクソノミー文書の URI を基に，使用しているタクソノミー文書が自動的に読み込まれます。ユーザーアプリケーションでは，uCosminexus Business Reporting Processor が提供する API を使用して，インスタンス文書を参照できます。

また，インスタンス文書を参照する場合，インスタンス文書が対応する XBRL のバージョンを指定する必要があります。デフォルトでは 2.0 になるため，XBRL2.1 に対応する文書を操作する場合は，XBRL のバージョンを変更します。

また，インスタンス文書が XBRL Dimensions に対応している場合，XBRL Dimensions のバージョンを指定する必要があります。その場合は，必ず XBRL のバージョンに 2.1 を指定してください。2.0 を指定したときは，インスタンス文書は XBRL2.0 として処理されます。

指定方法

- XBRL のバージョンは，プロパティや API などで指定します。指定するプロパティを次に示します。
  cbrp.parse.XBRLVersion
- XBRL Dimensions のバージョンは，プロパティや API などで指定します。指定するプロパティを次に示します。
  cbrp.parse.DimensionsVersion

指定方法の詳細については，「4.16.1　プロパティの指定方法と優先順位」を参照してください。

2.3 タクソノミー文書の検証

uCosminexus Business Reporting Processor では，タクソノミー文書の読み込み時に，その内容が正しいかどうかを検証できます。

タクソノミー文書の検証では，次の内容を検証します。

- XML スキーマ検証
- リンクの検証（リンクの整合性チェック）
- タクソノミー文書の形式の検証

指定方法

タクソノミー文書を検証するかどうかは，プロパティまたは API で指定します。指定するプロパティを次に示します。

- 通常の検証
  cbrp.parse.validate
- 完全な妥当性検証
  cbrp.parse.validate.full-checking

指定方法の詳細については，「4.16.1　プロパティの指定方法と優先順位」を参照してください。
また，これらのプロパティで検証しない設定にした場合でも，XBRL DOM を作成できないエラーについては必ず検証します。検証内容については，「付録 A　XBRL 文書の検証内容と対応するメッセージ ID」を参照してください。

2.3.1 XML スキーマ検証

タクソノミー文書は，XBRL のスキーマ定義に従って記述された XML 文書であり，これを構成するタクソノミー本体は XML Schema 言語で記述されています。そのため，XML プロセッサを使用して，次のスキーマ定義を基にタクソノミー文書が XML 文書として妥当であるかを検証できます。

- XBRL のスキーマ定義
  XBRL のバージョンに合ったスキーマ定義が必要です。
- XML Schema のスキーマ定義

検証するには，公開されているスキーマ定義をダウンロードするか，公開先にアクセスできる状態にしてください。検証に必要なスキーマ定義の詳細については，「4.15.3　必要なスキーマ定義」を参照してください。

（1） XBRL のバージョンによる XML スキーマ検証の違い

XML スキーマ検証は，XBRL のバージョンによって次のように異なります。

● XBRL2.0 の場合

プロパティや API の指定に関係なく，読み込み時に常に XML スキーマ検証が実行されます。

● XBRL2.1 の場合

プロパティまたは API での指定に従って，XML スキーマ検証が実行されます。

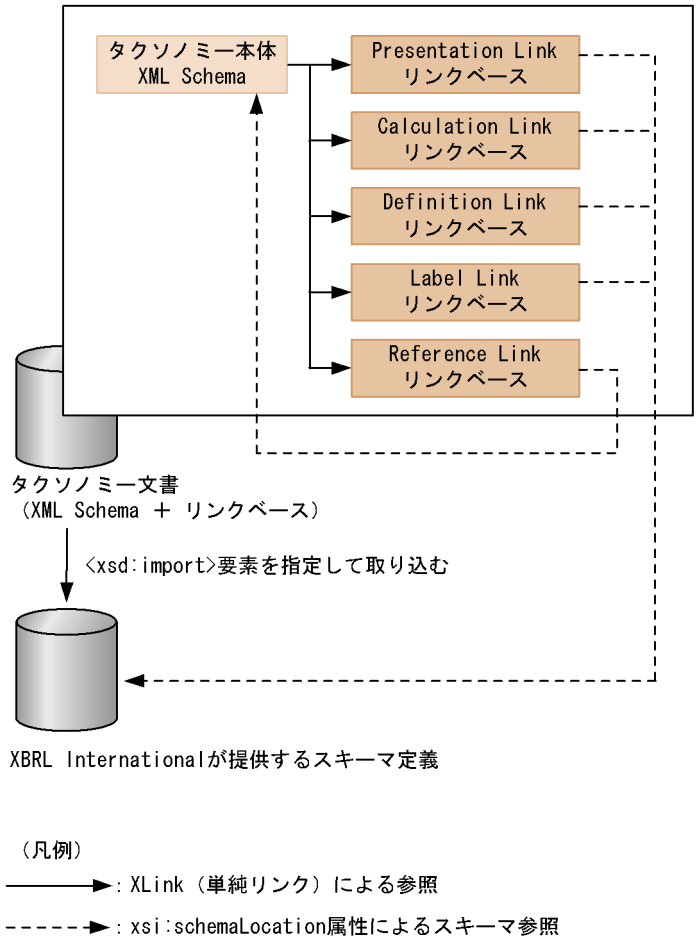
(2) リンクベースの XML スキーマ検証

リンクベースは，XBRL のスキーマ定義に従って記述された XML 文書です。例えばリンクベースの一つであるリファレンスリンクでは，タクソノミー本体で定義されたパート要素を記述することがありますが，このような場合，リンクベースを参照するタクソノミー本体もリンクベースのスキーマ定義として使用します。

リンクベースの xsi:schemaLocation 属性にタクソノミー本体を指定することで，タクソノミー本体をリンクベースのスキーマ定義として読み込めます。

タクソノミー文書（リンクベース）の XML スキーマ検証を次の図に示します。

図 2-3 タクソノミー文書（リンクベース）の XML スキーマ検証



(3) ユーザーデータ型の XML スキーマ検証【完全な妥当性検証の場合だけ有効】

完全な妥当性検証を実行すれば，ユーザーデータ型に XBRL2.1 で禁止されている定義内容がないかを検証できます。これによって，XML Schema として正しいデータ型でも，XBRL として不正ならばエラーを検出させることができます。

この検証方法を使うと，xbrli:fractionItemType 型以外から派生したユーザーデータ型が構造を持っているとエラーとなるため，次に示すユーザーデータ型定義（<xsd:complexType> 要素）の子要素に <xsd:simpleContent> 要素や <xsd:annotation> 要素以外が使用されているとエラーになります。

- あるアイテムとして認識された <xsd:element> 要素の子要素である <xsd:complexType> 要素（局所定義）
- あるアイテムとして認識された <xsd:element> 要素の type 属性値で参照される

<xsd:complexType> 要素（大域定義）

2.3.2 リンクの検証

タクソノミー文書の検証では，リンクベースに記述されたリンクの整合性のチェックとして，XLink プロセッサで次の内容を確認します。

- 単純リンクやロケータ型要素の検証
- ラベル定義の有無の検証
- 同じ弧を持つリンクの有無の検証
- xlink:role，xlink:arcrole 属性値の URI 表記の検証
- 関係ネットワーク上のサイクルの検証
- essence-alias アークの検証
- summation-item アークの検証

URI の指定方法に関する注意事項と，検証内容について次に説明します。

(1) URI の指定方法

URI の指定方法には，次の 2 種類があります。

- 絶対 URI
- 相対 URI

相対 URI は絶対 URI に変換され，リモートリソースを特定します（相対 URI の解決）。相対 URI の解決は xlink:href 属性だけでなく，<xsd:import> 要素などの，XBRL 文書でリモートリソースへの参照定義に使用される相対 URI に適用されます。ここでは，相対 URI の解決，エンティティリゾルバを使用した URI の解決，URI 表記に関する注意事項，および XPointer を使用した URI の指定方法について説明します。

(a) 相対 URI の解決

タクソノミー本体やリンクベースの要素など，リモートリソースへの参照定義に相対 URI を使用した場合，相対 URI は絶対 URI に変換されます。XBRL2.1 の場合は，基底となる URI を記述する xml:base 属性を指定できます。

uCosminexus Business Reporting Processor は，XML Base の仕様に従って相対 URI を解決します。

次の場合，解決後の URI は http://xxx/yyy/test.xsd となります。

```
<schemaRef xml:base="http://xxx/yyy/" xlink:href="test.xsd" ...(中略)>
```

xlink:href 属性値と xml:base 属性値を組み合わせても絶対 URI にならない場合，XML 文書読み込み元の URI が基底 URI として使用されます。XML 文書読み込み元の URI が定まっていない場合，アプリケーションの現在の作業フォルダが基底 URI として使用

され，絶対 URI に変換されます。

(b) エンティティリゾルバを使用した URI の解決

エンティティリゾルバは，アプリケーションで外部エンティティを解決するための機能です。uCosminexus Business Reporting Processor では，org.xml.sax.EntityResolver インターフェースを利用できます。

uCosminexus Business Reporting Processor は，エンティティリゾルバが登録されていた場合，タクソノミー文書などの外部エンティティを解決する前に，アプリケーションにコールバックします。例えば，エンティティリゾルバを使用して，絶対 URI をローカルファイル名に変換し，リモートリソースにアクセスしないようにできます。

エンティティリゾルバのコーディングの詳細については，「4.14　エンティティリゾルバ」を参照してください。

(c) URI 表記に関する注意

XBRL 文書の URI は，uCosminexus Business Reporting Processor によって正規化され，XBRL 文書を識別するためのキーとして管理されます。そのため，URI 表記について，次の点に注意してください。

- 大文字／小文字の表記を統一する
  XBRL 文書に，同じ文字列で大文字／小文字の表記だけが異なる URI がある場合，uCosminexus Business Reporting Processor では別文書として認識されます。同じ URI を記述する場合，大文字／小文字の表記を統一してください。
  例：次の URI は，異なる URI と判断されます。
  - file:/test/dmy.xsd
  - file:/test/DMY.xsd
  - file:/test/Dmy.Xsd
- URI の記述形式を統一する
  URI 中のホスト名およびポート番号を省略した場合，uCosminexus Business Reporting Processor では異なる URI として認識されます。同じ URI を記述する場合，URI の記述形式を統一してください。
  例：次の URI は，異なる URI と判断されます。
  - http://localhost:80/test/dmy.xsd
  - http:///test/dmy.xsd
- 先頭の文字に OS のドライブ名を指定しない
  例えば，「C:/xxx.xml」を指定した場合，URI が正しく変換されないため，使用しないでください。
- 先頭の文字に「/」を指定した場合
  先頭の文字に「/」を使用した場合，アプリケーションの現在の作業フォルダの影響を受けるため，注意してください。

例を次に示します。
次の記述がある文書は，file:///C:/instance/sample.xml だったと仮定します。

```
<schemaRef xml:base="/xxx/" xlink:href="test.xsd" ...(中略)>
```

この場合，相対 URI の解決後の絶対 URI は file:/xxx/test.xsd となります。
先頭の文字に「/」を指定した場合，相対 URI"test.xsd" は，xml:base 属性値に指定された値 "/xxx/" と組み合わされて，絶対パス "/xxx/test.xsd" となります。この記述のある文書のスキームとホスト名から正規化された絶対 URI は "file:/xxx/test.xsd" となります。
この絶対 URI でリソースを求めると，アプリケーションの現在の作業フォルダのドライブからのパスとして解決されます。現在の作業フォルダが D:¥sample であった場合には，求められるリソースは，file:/D:/xxx/test.xsd となります。

(d) XPointer を使用した URI の指定方法

xlink:href 属性に指定する URI のフラグメントに，XPointer を使用できます。ここでは，使用できる形式と指定例について説明します。

XBRL2.0 の場合，URI のフラグメントには XPointer のショートハンドポインタだけを指定できます。

XBRL2.1 の場合，次に示す形式を使用できます。

- ショートハンドポインタ
  XPointer のショートハンドポインタを使用してリンク先の要素の id 属性を指定します。記載形式は，#<id> です。
- element() スキーム
  XPointer のスキームベースポインタの element() スキームを使用します。複数の条件を指定した場合，条件に一致する要素が見つかった時点で，検索が終了します。
  記載形式は，element(< 値 >)[element(< 値 2>)] です。

指定例を次に示します。

- ショートハンドポインタを使用して，リンク先の id 属性を指定する例
  この例では，test.xsd 文書内の test1 という id を持つ要素を参照しています。

```
<link:roleRef xlink:type="simple"
 xlink:href="test.xsd#test1"
 roleURI="http://www.mycomp.com/role/endnote"
 />
```

- element() スキームを使用して，リンク先の id 属性を指定する例
  この例では，test.xsd 文書内の test1 という id を持つ要素を参照しています。

```
<link:roleRef xlink:type="simple"
 xlink:href="test.xsd#element(test1)"
 roleURI="http://www.mycomp.com/role/endnote"
 />
```

- ルート要素からの階層を指定する例
  この例では，test.xsd 文書のルート要素下にある 14 番目の子要素を参照しています。

```
<link:roleRef xlink:type="simple"
 xlink:href="test.xsd#element(/1/14)"
 roleURI="http://www.mycomp.com/role/endnote"
 />
```

- 特定の要素からの階層を指定する例
  この例では，test.xsd 文書内の test1 という id を持つ要素下にある 3 番目の子要素の 2 番目の子要素を参照しています。

```
<link:roleRef xlink:type="simple"
 xlink:href="test.xsd#element(test1/3/2)"
 roleURI="http://www.mycomp.com/role/endnote"
 />
```

- 複数の条件を指定した例
  element() スキームに二つの条件を指定した場合，順番に検索され，条件に一致する要素が見つかった時点で検索が終了します。

```
<link:roleRef xlink:type="simple"
 xlink:href="test.xsd#element(test1/3/2)  element(/1/2)"
 roleURI="http://www.mycomp.com/role/endnote"
 />
```

  この例では，次の順番で検索されます。
  1. test.xsd 文書内の test1 という id を持つ要素下にある 3 番目の子要素の 2 番目の子要素
  2. test.xsd 文書内のルート要素下にある 2 番目の子要素

(2) 単純リンクやロケータ型要素の検証

単純リンクやロケータ型要素が，参照先となるリモートリソースにトラバースできるかを検証します。これは，xlink:href 属性に記述されたリモートリソースへの URI から，参照先のタクソノミー本体の要素や，リンクベースの要素が特定できることを意味します。

(3) ラベル定義の有無の検証

アーク型要素が参照しているラベル（xlink:label 属性）が，その拡張リンク型要素内で定義されているかを検証します。

(4) 同じ弧を持つリンクの有無の検証

同一拡張リンク内に同じ弧を持つリンク（同じ xlink:from 属性値および同じ xlink:to 属性値を持つアーク型要素）がないかを検証します。

(5) xlink:role，xlink:arcrole 属性値の URI 表記の検証

xlink:role，xlink:arcrole 属性値が絶対 URI で指定されているかを検証します。

(6) 関係ネットワーク上のサイクルの検証【完全な妥当性検証の場合だけ有効】

関係ネットワークとは，拡張リンク型要素の xlink:role 属性値で分類され，さらにアーク型要素の xlink:arcrole で細分化される関係の集合です。完全な妥当性検証の場合，関係ネットワーク上で禁止されているサイクル（循環など）が存在しないか検証できます。

ラベルリンク，リファレンスリンクおよび脚注リンクは，検証の対象になりません。検証の対象となるリンクと，各リンクに許可されているサイクルのタイプを次に示します。

リンクの種類	要素名	xlink:arcrole 属性値	サイクルのタイプ
プレゼンテーションリンク	link:presentationArc	http://www.xbrl.org/2003/arcrole/parent-child	undirected
計算リンク	link:calculationArc	http://www.xbrl.org/2003/arcrole/summation-item	any
定義リンク	link:definitionArc	http://www.xbrl.org/2003/arcrole/general-special	undirected
		http://www.xbrl.org/2003/arcrole/essence-alias	undirected
		http://www.xbrl.org/2003/arcrole/similar-tuples	any
		http://www.xbrl.org/2003/arcrole/requires-element	any
―	―	カスタムアークロール（<link:arcroleType> 要素の arcroleURI 属性値）	cyclesAllowed 属性の値

（凡例）

―：要素名の指定は不要です。

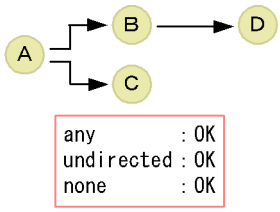
サイクルのタイプ：

指定できるサイクルのタイプ（cyclesAllowed 属性の指定値を含む）と，それぞれの意味は次のとおりです。

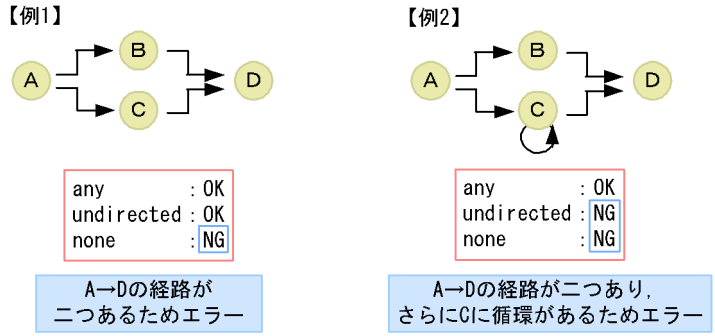
- any：関係ネットワーク上でのサイクルの制限はない。
- undirected：関係ネットワーク上での循環はエラーとするが，ノード間に複数の経路が存在してもエラーとはしない。
- none：関係ネットワークが循環することも，あるノード間に経路が複数存在することも許可しない。

関係ネットワーク上の経路とサイクルのタイプの組み合わせによって，検証結果がエラーになる場合とならない場合を次の例で示します。

■サイクルの検証結果がエラーにならない例



■サイクルの検証結果がエラーになる例



(7) essence-alias アークの検証【完全な妥当性検証の場合だけ有効】

essence-alias アークが，次の制約に従って記述されているかを検証します。

- 実名概念と別名概念のアークの両端がアイテムであること
- 実名概念と別名概念の要素宣言（アイテム）のデータ型は同じであること（派生元が同じでも，データ型が異なる場合は別のものと判断する）
- 実名概念と別名概念の要素宣言（アイテム）の periodType 属性値が同じであること
- 実名概念と別名概念の要素宣言（アイテム）の両方に balance 属性が記述されていた場合，balance 属性値が同じであること

(8) summation-item アークの検証【完全な妥当性検証の場合だけ有効】

summation-item アークが，次の制約に従って記述されているかを検証します。

- 計算リンクを構成するアークの両端は，アイテムであること
- 計算リンクを構成する要素宣言（アイテム）は，すべて数値型であること
- 計算リンクを構成する要素宣言（アイテム）のデータ型は，同じであること（派生元が同じならば，データ型が異なる場合でも同じものと判断する）

2.3.3　タクソノミー文書の形式の検証

XML スキーマ検証，リンクの検証に加え，uCosminexus Business Reporting Processor ではタクソノミー文書として正しいかをチェックします。ここでは代表的なチェック内容について説明します。より詳細なチェック内容については，「5.　取り扱える XBRL 文

書とデータ型」を参照してください。

(1) 要素の二重定義の有無の検証

XBRL では，タクソノミー本体に，インスタンス文書で使用するアイテム要素やタプル要素を記述します。このとき，同一名称の要素を記述できません。

同一名称の要素とは，同じ名前空間に属し，同じ name 属性値を持つ要素のことです。同じ name 属性値でも異なる名前空間に属する場合は，異なる名称の要素であると判断されます。

タクソノミー文書が拡張されていた場合，XML スキーマ検証では要素が同一名称を持つかどうかがチェックされないため，uCosminexus Business Reporting Processor では，同一名称を持った要素が記述されていないかを検証します。

(2) 単純リンクのリモートリソースの参照先の検証

リンクの検証では，単純リンク（<link:linkbaseRef> 要素など）のリモートリソースへトラバースできるかは確認できますが，その参照先が正しい内容であるかは確認できません。そのため，uCosminexus Business Reporting Processor では，リモートリソースの内容をチェックします。例えば，単純リンクの xlink:role 属性値が計算リンクを示す値であっても，実際のリモートリソースが定義リンクであることも考えられます。この場合，uCosminexus Business Reporting Processor は，実際のリモートリソースが定義された値と異なるというエラーを通知します。

XBRL2.1 では，単純リンク型要素として，<link:roleRef> 要素および <link:arcroleRef> 要素が追加されました。これらの要素は，リンクベースから，タクソノミー本体に定義されているカスタムロールおよびカスタムアークロールを参照するための要素です。uCosminexus Business Reporting Processor では，<link:roleRef> 要素および <link:arcroleRef> 要素の参照先がタクソノミー本体であるか，および <link:roleType> 要素および <link:arcroleType> 要素であるかをチェックします。

(3) 拡張リンクの構文の検証

XBRL2.0 の場合，uCosminexus Business Reporting Processor では，拡張リンクの構文が正しいかをチェックします。XBRL2.1 の場合，XBRL International によって提供されるスキーマ定義を使用して，XML スキーマ検証時にチェックします。

拡張リンクの構文チェックでは，各拡張リンク要素の構文（内容）について，拡張リンク要素内に含まれるはずのない要素がないかをチェックします。例えば，<calculationLink> 要素に，<labelArc> 要素が含まれていないかなどをチェックします。

XBRL では，リンクベースで使用する拡張リンクの構文規則が示されています。拡張リンクの構文規則を次の表に示します。

表 2-1　拡張リンクの構文規則

項番	要素名※	日本語名	規則	
			必要な要素	xlink:type 属性値
1	calculationLink	計算リンク	<link:loc> 要素	locator
			<link:calculationArc> 要素	arc
2	definitionLink	定義リンク	<link:loc> 要素	locator
			<link:definitionArc> 要素	arc
3	presentationLink	プレゼンテーションリンク	<link:loc> 要素	locator
			<link:presentationArc> 要素	arc
4	labelLink	ラベルリンク	<link:loc> 要素	locator
			<link:label> 要素	resource
			<link:labelArc> 要素	arc
5	referenceLink	リファレンスリンク	<link:loc> 要素	locator
			<link:reference> 要素	resource
			<link:referenceArc> 要素	arc

注※　拡張リンク要素の xlink:type 属性値には，extended を指定します。

(4) リンク構造の検証

XBRL では，タクソノミー文書を拡張するために，リンクを上書きできます。このリンクの上書きでは，アーク型要素の use 属性および priority 属性の値に基づいて，リンクの上書きやリンクの禁止を判断する必要があります。uCosminexus Business Reporting Processor では，このリンクの上書きを処理し，XBRL として有効なリンク構造であるかを検証できます。

(5) タプルの検証

タプル定義については，次の内容を検証します。

- タプルが xbrli:tuple から派生していること
- タプルを宣言する要素に，xbrli:periodType 属性と xbrli:balance 属性が含まれていないこと

さらに，完全な妥当性検証を実行すれば，次の内容も検証できます。

- タプルが混合内容または単純な内容を持たないこと
  例えば，次のような宣言はエラーとなる。

<xsd:complexContent mixed="true">

- タプル内要素の宣言が，xbrli:item または xbrli:tuple から派生した要素のグローバル要素であること
  例えば次の場合，element 要素が ref 属性で参照されていないのでエラーとなる。
  <xsd:element name="managementName" type="xbrli:tokenItemType"/>

2.4 インスタンス文書の検証

インスタンス文書は，タクソノミー文書に定義された項目に従って実際の値が入力されたドキュメントです。タクソノミー文書と同様にリンクベースを埋め込み，リンクを上書きできます。また，インスタンス文書では，項目と脚注を関係づける脚注リンクを記述することもできます。

このためインスタンス文書の検証では，タクソノミー文書と同様に，次の内容を検証します。

- XML スキーマ検証
- リンクの検証（リンクの整合性チェック）
- インスタンス文書の形式の検証

指定方法

インスタンス文書を検証するかどうかは，プロパティまたは API で指定します。指定するプロパティを次に示します。

- 通常の検証
  cbrp.parse.validate
- 完全な妥当性検証
  cbrp.parse.validate.full-checking

指定方法の詳細については，「4.16.1　プロパティの指定方法と優先順位」を参照してください。
また，これらのプロパティで検証しない設定にした場合でも，XBRL DOM を作成できないエラーについては必ず検証します。検証内容については，「付録 A　XBRL 文書の検証内容と対応するメッセージ ID」を参照してください。

2.4.1 XML スキーマ検証

インスタンス文書は，タクソノミー文書（タクソノミー本体）に従って記述された XML 文書です。

タクソノミー文書をインスタンス文書のスキーマ定義としてスキーマ検証をすることで，インスタンス文書が XML 文書として妥当であるかを検証します。

2.4.2 リンクの検証

インスタンス文書では単純リンクを記述できます。また，インスタンス文書には，拡張リンクの一つである脚注リンクを記述できます。このため，タクソノミー文書と同様にリンクの整合性をチェックします。

さらに，完全な妥当性検証を実行すれば，次の内容も検証できます。

(1) requires-element アークの検証【完全な妥当性検証の場合だけ有効】

定義リンク（link:definitionLink）の requires-element アークで，xlink:from 属性に対応するアイテムやタプルがインスタンス文書中に存在する場合，それと対になる xlink:to 属性に対応するアイテムやタプルもインスタンス文書中に存在するかを検証します。存在しなければエラーとします。ただし，xlink:to 属性が指すアイテムまたはタプルがあって，xlink:from 属性が指すアイテムまたはタプルがない場合は，エラーとしません。

なお，この検証では，インスタンス文書の任意の位置に from 属性および to 属性に対応するアイテムまたはタプルが存在すればよく，「それぞれのアイテムまたはタプルが p-equal，c-equal および u-equal でなければならない」という条件はありません。
p-equal，c-equal および u-equal の意味は次のとおりです。これらの用語の詳細については，XBRL の仕様書を参照してください。

- c-equal（context-equal）：コンテキストが同じか，同じコンテキストに属する関係
- p-equal（parent-equal）：親が同じか，同じ親を持つ関係
- u-equal（unit-equal）：ユニット（単位）が同じか，同じユニット（単位）に属する関係

(2) essence-alias アークの検証【完全な妥当性検証の場合だけ有効】

定義リンク（link:definitionLink）の essence-alias アークで，xlink:from 属性値のトラバース先であるアイテム（実名アイテム）と，それと対になる xlink:to 属性値のトラバース先であるアイテム（別名アイテム）とが同一かを検証します。ここでいう同一とは，p-equal および c-equal の関係にあるアイテム同士が v-equal（value-equal：値が同じ，または同じ値を持つ関係）であることをいいます。また，検証の対象となるのは，リンクベースの検証結果がエラーになっていない関係ネットワークの essence-alias アークだけです。

なお，実名アイテムが存在しても，別名アイテムが存在しない場合は検証されません。
このほかに検証が実施されないケースを次に示します。

● アイテムが重複する場合
複数の実名アイテム（または別名アイテム）の名前空間とローカル名が同一で，さらに p-equal かつ c-equal の関係にある場合は，二つ目以降のアイテムを警告とし，essence-alias の検証は実施しません。
同様に，実名アイテムを推測する機能を使用している際に別名アイテムが重複した場合も，一つ目のアイテムについてだけ実名アイテムを推測します。二つ目以降の別名アイテムについては警告とし，アイテムの推測は実施しません。

● アイテムに nil 属性が指定されている場合
実名アイテムや別名アイテムの xsi:nil 属性が true である場合は，その essence-alias アークの検証は実施しません。

また，アイテムの有効数字がない場合等の特殊なケースを次に示します。

● アイテムの有効数字がない場合
実名アイテム，および別名アイテムの両方または，片方のアイテムの precision 属性値が 0 であるか，または decimals 属性値から推測される precision 属性値が 0 である場合は，essence-alias アークの検証は常に不正（偽：false）となります。
ただし，プロパティ ” cbrp.parse.validate.full-checking.ignoringZeroPrecision ” が false の場合であり，true が設定されている場合には精度を考慮せず数値比較となります。
このプロパティの詳細については，「4.16.2(21)　cbrp.parse.validate.full-checking. ignoringZeroPrecision」を参照してください。

● 数値アイテムの値が NaN である場合
実名アイテム，および別名アイテムの両方または，片方のアイテムの値が NaN である場合，実名アイテム，および別名アイテムの precision 属性値，または decimals 属性値から推測される precision 属性値（精度）に関係なく，常に不正（偽：false）となります。NaN==NaN も不正（偽：false）となります。

● 数値アイテムの値が INF である場合
実名アイテム，および別名アイテムの両方が INF である場合，実名アイテム，および別名アイテムの precision 属性値，または decimals 属性値から推測される precision 属性値（精度）に関係なく，正常 (真：true) となります。

● 数値アイテムの値が -INF である場合
実名アイテム，および別名アイテムの両方が -INF である場合，実名アイテム，および別名アイテムの precision 属性値，または decimals 属性値から推測される precision 属性値（精度）に関係なく，正常 (真：true) となります。

● 数値アイテムの値が 0 である場合
実名アイテム，および別名アイテムの両方が数値として 0 である場合，実名アイテム，および別名アイテムの precision 属性値，または decimals 属性値から推測される precision 属性値（精度）に関係なく，正常 (真：true) となります。0.0 と 0 も正常となります。また，0.000 と 0.0 も正常となります。0.0 と 0.0001 は不正となります。

また，必要に応じて次の検証を実施することもできます。

(a) インスタンス文書に存在しないアイテムを推測し検証する

インスタンス文書中に，ある essence-alias アークの xlink:to 属性値に対応するアイテム B が存在し，かつ，このアイテムと p-equal および c-equal が成立するアイテム A（xlink:from 属性値に対応）が存在しない場合，アイテム A を推測することもできます。アイテムの推測にはプロパティ cbrp.parse.validate.full-checking.essence-alias.infer を使用します。

essence-alias アークによる実名アイテムと別名アイテムの関係が連続する場合には，インスタンス文書に存在しないアイテムをすべて再帰的に推測して検証します。この機能を使用すると，推測されたアイテムは PTVI（Post-Taxonomy Validation Infoset）に含

まれるため，推測したアイテムと既存アイテムとの間に検出された矛盾内容は PTVI で確認できます。

このプロパティの詳細については，「4.16.2(20) cbrp.parse.validate.full-checking.essence-alias.infer」を参照してください。PTVI の出力内容については，「2.10　PTVI の生成（XBRL2.1 の場合）」を参照してください。また，PTVI の詳しい意味については，XBRL の仕様書を参照してください。

インスタンス文書に存在しないアイテムを推測する例を次に示します。

例 1：アイテム A →アイテム B への essence-alias アーク

- アイテム B の内容

```
<B contextRef="c1" precision="3" unitRef="u1">123</B>
```

- アイテム B から推測されるアイテム A の内容

```
<A contextRef="c1" precision="3" unitRef="u1">123</A>
```

例 2：アイテム A →アイテム B →アイテム C への essence-alias アーク

- アイテム C の内容

```
<C contextRef="c1" precision="3" unitRef="u1">100</C>
```

- アイテム C から推測されるアイテム A，アイテム B の内容

```
<A contextRef="c1" precision="3" unitRef="u1">100</A>
<B contextRef="c1" precision="3" unitRef="u1">100</B>
```

例 3：アイテム A →アイテム B，アイテム A →アイテム C への essence-alias アーク

- アイテム B の内容

```
<B contextRef="c1" precision="3" unitRef="u1">123</B>
```

- アイテム C の内容

```
<C contextRef="c1" precision="2" unitRef="u1">120</C>
```

- アイテム B またはアイテム C から推測されるアイテム A の内容

```
<A contextRef="c1" precision="3" unitRef="u1">123</A>
```

この場合は，アイテム B・アイテム C のどちらからでもアイテム A を推測できます。アイテム B（またはアイテム C）から推測されたアイテム A は，アイテム C（またはアイテム B）と v-equal が成立するかを検証します。そのため，アイテム B とアイテム C の v-equal が成立しない場合はエラーになり，推測されたアイテム A はエラーの根拠として PTVI に出力されます。
一方，v-equal が成立し，かつ数値アイテムの場合は，推測したアイテム A とアイテム C（またはアイテム B）のうち，precision 属性値が大きい方のアイテムからアイテム A を推測し直します。そのため，アイテム A はアイテム B から推測した値となります。また，非数値アイテムの場合は，余分な空白を削除した値（XPath の normalize-space() 関数で得られる結果と同じ）が推測されるため，v-equal である場合はどちらも同じ値です。

(b) 検証結果の例

例：アイテム A →アイテム B →アイテム C への essence-alias アーク

これらの例の u1・u2・u3 および c1・c2・c3 は，s-equal（structure-equal）でないと仮定する。

- 次の場合は A → B の範囲で検証する。

```
<A unitRef="u1" contextRef="c1" precision="2" id="a1">10</A>
<B unitRef="u1" contextRef="c1" precision="2" id="b1">10</B>
```

- 次の場合は c-equal，p-equal の B または C がないため，検証しない。

```
<A unitRef="u1" contextRef="c4" precision="2" id="a2">10</A>
```

- 次の場合は A → B の範囲で検証する。

```
<tuple>
  <A unitRef="u1" contextRef="c2" precision="2" id="a3"></A>
  <B unitRef="u1" contextRef="c2" precision="2" id="b3"></B>
</tuple>
```

- 次の場合は xsi:nil 属性に true が指定されているため，検証しない。

```
<A unitRef="u1" contextRef="c2" precision="2" id="a4" xsi:nil='true'></A>
<B unitRef="u2" contextRef="c2" precision="2" id="b4" xsi:nil='true'></B>
```

- 次の場合は二つのアイテムが重複しているため，警告となる。

```
<A unitRef="u2" contextRef="c1" precision="2" id="a7">10</A>
<B unitRef="u2" contextRef="c1" precision="2" id="b8">10</B>
```

- 次の場合は，拡張機能を利用していれば A → B → C の範囲で検証する。
  その場合，C から B を推測して検証する。

```
<A unitRef="u3" contextRef="c3" precision="2" id="a9">20</A>
<C unitRef="u3" contextRef="c3" precision="2" id="b9">20</C>
```

(3) summation-item アークの検証【完全な妥当性検証の場合だけ有効】

計算リンク（link:calculationLink）の summation-item アークで，次の値が同じになっているかを検証します。

- summation-item アークの xlink:from 属性に対応するアイテム（合計値）
- summation-item アークの xlink:to 属性値に対応するアイテムの集合から算出した値（検証値）

ただし，リンクベースの完全な妥当性検証でエラーになっていない関係ネットワークにある summation-item アークだけが検証の対象になります。また，関係ネットワークに含まれても，インスタンス文書に存在しないアイテムについては検証しません。

また，検証が実施されないケースを次に示します。

● アイテムが重複する場合
p-equal，u-equal および c-equal で重複するアイテムが検出された場合には，重複したアイテムを含む summation-item アークによる合計値の検証をしません。また，検証を行わなかった合計値（アイテム），および重複したアイテムについて警告を通知します。ただし，同一スコープであっても，p-equal でないアイテムについては，ともに合計値の計算に含まれるものとします。

● アイテムに nil 属性が指定されている場合
summation-item アークの xlink:from 属性に対応するアイテム（合計値）の nil 属性値が true の場合には，その合計値の検証をしません。
また，summation-item アークの関係ネットワークにおける xlink:to 属性値に対応するアイテム（数値 n）の nil 属性値が true の場合には，検証値の算出には含まれません。

また，アイテムの有効数字がない場合の特殊なケースを次に示します。

● アイテムの有効数字がない場合
summation-item アークの xlink:from 属性に対応するアイテム（合計値）の precision 属性値が 0 であるか，または decimals 属性値から推測される precision 属性値が 0 である場合には，その合計値の検証は常に不正と判断します。ただし，合計値が 0 であった場合には，検証値との検証を行います。
また，precision 属性値が 0 である場合に常に不正と判断するのは，プロパティ ” cbrp.parse.validate.full-checking.ignoringZeroPrecision ” が false に設定されている場合であり，true が設定されている場合には精度を考慮せずに数値比較となります。このプロパティの詳細については，「4.16.2(21)　cbrp.parse.validate.full-checking. ignoringZeroPrecision」を参照してください。

(4) 精度の推測と概数【完全な妥当性検証の場合だけ有効】

検証時に数値アイテムの v-equal を比較するには，それぞれの値を精度の小さい方に丸めた値で比較します。数値アイテムの値の精度は，precision 属性（有効桁数）や decimals 属性（正確であると考えられる小数位の桁数）から決定されます。
uCosminexus Business Reporting Processor では，インスタンス文書に対して完全な妥当性検証を実行するために，次の処理をします。

- precision 属性が省略されている場合，推測された precision 属性値を基に概数を求める。
- 比較する場合，求めた概数同士をさらに precision 属性の小さい方で丸めた値で比較し検証する。

次に，precision 属性の推測と概数の求め方と比較の方法を示します。

(a) decimals 属性値を基に precision 属性値を推測する方法

数値アイテムの値の精度は precision 属性で示されます。precision 属性が省略されている場合は，decimals 属性から次の要領で precision 属性値を推測します。

● **fractionItemType 型（またはその派生されたデータ型）のアイテムの場合**

precision 属性値として「INF」が仮定されます。

● **fractionItemType 型（またはその派生されたデータ型）以外のアイテムの場合**

precision 属性値は次の計算式で算出されます。ただし，算出結果が 0 以下の場合は，precision 属性値は 0 になります。

precision属性値の計算式＝n1＋n2＋n3

計算式の n1，n2，n3 の意味を次に示します。

- n1：小数点（小数点が省略されている場合は，1 の位の右に小数点を仮定します）の左側に 0 以外の桁がある場合は，その数値アイテムのそのままの値の小数点より左側にある，前ゼロを除いた桁数になります。
  小数点の左側に 0 以外の桁がなく，小数点の右側に 0 以外の桁がある場合は，その数値アイテムのそのままの値の小数点とその右側にある，最初の 0 でない桁の間にある 0 の個数を負にした値になります。
  それ以外の場合，n1 は 0 になります。
- n2：数値アイテムの値の表現に指数表記がある場合は，その指数の値になります。ない場合は 0 になります。
- n3：数値アイテムに対して指定されている decimals 属性の値です。

数値アイテムと decimals 属性値を基に，precision 属性値を計算する例を次に示します。

数値アイテム	decimals 属性値	precision 属性値の計算結果（n1 ＋ n2 ＋ n3）
123	2	3+0+2=5
123.4567	2	3+0+2=5
123e5	-3	3+5+(-3)=5
123.45e5	-3	3+5+(-3)=5
0.1e-2	5	0+(-2)+5=3
0.001E-2	5	(-2)+(-2)+5=1
0.001e-3	4	(-2)+(-3)+4=-1 結果が 0 未満のため，precision 属性値は 0 となる。

なお，検証時に precision 属性値（または decimals 属性値から推測された precision 属性値）が 0 となった場合は，警告が報告されます。

(b) 概数の算出方法

概数は precision 属性値（省略されている場合は，decimals 属性値から推測された

precision 属性値）から得た有効桁数のあとの数字を四捨五入して算出します。次の場合には，概数は算出されません。

- precision 属性値が INF の場合
- 有効数字が存在しない場合

数値アイテム，precision 属性値および decimals 属性値を基に，概数を算出する例を次に示します。

数値アイテム	precision 属性値	decimals 属性値	概数の算出結果
12.344	4	—	12.34
12.345	4	—	12.35
12.335	4	—	12.34
12.346	4	—	12.35
12.34501	4	—	12.35
12345	4	—	12350
12345	7（decimals 属性値から推測した値）	2	12345.00
0.1e-2	3（decimals 属性値から推測した値）	5	0.00100

（凡例）—：該当しません。

(c) 数値アイテムの比較方法

数値アイテムが v-equal かどうかを検証するには，c-equal および u-equal である数値アイテム A と数値アイテム B が，次の条件を満たしている必要があります。

1. A と B の指定（または推定される精度）を比較し，小さい方の値を N とする
2. A と B をそれぞれ N 桁の有効数字で丸めた概数同士に，x-equal（[XPATH]-equal）が成立する

次に例として，二つの比較値（比較値自体が概数である場合を含む）と，その比較結果を示します。表中の ( ) の数値内は，precision 属性値（または decimals 属性値から推測された precision 属性値）を示します。

比較値 1	比較値 2	比較時に仮定される概数と精度	結果
12345(5)	12350(4)	【概数 1】12350(4),【概数 2】12350(4)	真
12345.00(7)	12345(5)	【概数 1】12345(5),【概数 2】12345(5)	真
0.00100(3)	0.001(1)	【概数 1】0.001(1),【概数 2】0.001(1)	真
12345(5)	12340(4)	【概数 1】12350(4),【概数 2】12340(4)	偽
12345.05(7)	12345.0(6)	【概数 1】12345.1(6),【概数 2】12345.0(6)	偽
12345(5)	1234(4)	【概数 1】12350(4),【概数 2】1234(4)	偽

比較値 1	比較値 2	比較時に仮定される概数と精度	結果
0.00100(3)	0.010(2)	【概数 1】0.0010(2),【概数 2】0.010(2)	偽
12345(0)	12345(5)	概数 1 の精度は 0 のため，丸め処理はしないで常に偽になる	偽

2.4.3　インスタンス文書の形式の検証

XML スキーマ検証，リンクの検証に加え，uCosminexus Business Reporting Processor ではインスタンス文書として正しいかをチェックします。ここでは代表的なチェック内容について説明します。より詳細なチェック内容については，「5.　取り扱える XBRL 文書とデータ型」を参照してください。

(1) DTS の基点となる位置情報の有無の検証

インスタンス文書では，DTS の基点となる位置情報がないと DTS を特定できないため，リンクをたどることができません。そのため，uCosminexus Business Reporting Processor では，DTS の基点となる位置情報がインスタンス文書に指定されているかを確認します。指定されていないインスタンス文書は処理できません。

DTS の基点となる位置情報の詳細については，「2.5　DTS の読み込み」を参照してください。

(2) コンテキスト要素（context）および単位要素（unit）の検証

あるアイテムはあるコンテキストに属している必要があります。uCosminexus Business Reporting Processor では，アイテム要素が参照するコンテキスト要素が，妥当であるかどうかをチェックします。チェック内容は，XBRL のバージョンによって異なります。

(a) XBRL2.0 の場合

XBRL2.0 の場合，コンテキスト要素は，数値コンテキストまたは非数値コンテキストに分かれます。コンテキストとアイテムの従属関係は，コンテキスト要素の id 属性（xsd:ID 型）と，アイテム要素の numericContext 属性値または nonNumericContext 属性値（xsd:IDREF 型）とで関係づけられます。uCosminexus Business Reporting Processor は，アイテムの型と，アイテムが属するコンテキストの型が一致するかどうかを確認します。

例えば，アイテム要素のデータ型が数値型であるのに対して，アイテム要素の numericContext 属性値で指し示すコンテキスト要素が <nonNumericContext> 要素であっても，XML スキーマ検証ではチェックできません。この場合，uCosminexus Business Reporting Processor がチェックして，アイテム要素が参照するコンテキスト要素が <numericContext> 要素でないというエラーを通知します。

XBRL2.0 の場合のコンテキスト検証を次の図に示します。

図 2-4　XBRL2.0 の場合のコンテキスト検証

●数値コンテキスト要素

```
<numericContext cwa="false" id="NC01" precision="9">
<entity>
   <identifier scheme="">SAMP</identifier>
<segment>
 <hi:prefecture>Tokyo</hi:prefecture>
</segment>
</entity>
<period>
   <instant>2003-03-31</instant>
</period>
<unit>
   <measure>iso4217:JPY</measure>
</unit>
</numericContext>
```

この二つの値を検証することで，
コンテキスト要素とアイテム要素が妥当かを
検証します。

●数値コンテキストに対応するアイテム要素

```
<hi:assets id="ITM02" numericContext="NC01">
100000</hi:assets>
<hi:currnetAssets id="ITM03" numericContext="NC01">
50000</hi:currnetAssets>
<hi:investmentsAndAdvances id="ITM04" numericContext="NC01">
20000</hi:investmentsAndAdvances>
```

(b) XBRL2.1 の場合

XBRL2.1 の場合，コンテキスト要素は <xbrli:context> 要素に統合され，新しく <xbrli:unit> 要素が追加されました。そのため，アイテム要素が参照する <xbrli:context> 要素および <xbrli:unit> 要素が妥当であるかをチェックします。

完全な妥当性検証では，アイテムの単位を示す <xbrli:measure> 要素の値（xsd:QName 型）が名前空間「http://www.xbrl.org/2003/iso4217」である場合，ローカルパートは妥当な ISO4217 通貨コードである必要があります。uCosminexus Business Reporting Processor のデフォルトの検証では，java.util.Currency クラスを使ってローカルパートが ISO4217 に準拠した通貨コードかをチェックします。そのため，チェックできる通貨コードは SDK のバージョンに依存します。

デフォルトの検証で不十分な場合には，ユーザー独自の通貨コード一覧を指定して，この一覧に含まれている通貨コードであるかどうか検証します。通貨コード一覧の指定の方法については，「4.16.2(17)　cbrp.parse.validate.full-checking.iso4217」を参照してください。

(3) 脚注リンクの構文の検証

インスタンス文書では，拡張リンクの一つである脚注リンクを記述できます。インスタンス文書でも，タクソノミー文書と同様に，拡張リンク要素内に含まれるはずのない要

素がないかをチェックします。

脚注リンクの構文規則を次の表に示します。

表 2-2　脚注リンクの構文規則

要素名※	日本語名	規則	
		必要な要素	xlink:type 属性値
footnoteLink	脚注リンク	<link:loc> 要素	locator
		<link:footnote> 要素	resource
		<link:footnoteArc> 要素	arc

注※　拡張リンク要素の xlink:type 属性値には，extended を指定します。

(4) リンク構造の検証

インスタンス文書では，タクソノミー文書と同様に単純リンクを記述できます。単純リンクが記述されていた場合，タクソノミー文書で定義されているリンク構造を，さらに上書きし検証する必要があります。

そのため，インスタンス文書でも，タクソノミー文書と同様のリンク構造の検証を実行します。

(5) タプルの検証

通常の検証では，タプルの定義についてだけ検証します。完全な妥当性検証では，次に示す名前空間を持つ属性がタプルに記述されていないかについても検証します。

- http://www.xbrl.org/2003/instance
- http://www.xbrl.org/2003/linkbase
- http://www.xbrl.org/2003/XLink
- http://www.w3.org/1999/xlink

2.5 DTS の読み込み

インスタンス文書を作成するときに，参照するタクソノミー文書が複数必要な場合があります。この場合，複数のタクソノミー文書の集まりを，論理的な一つのタクソノミー文書（DTS）として読み込めます。DTS の読み込み範囲は，XBRL のバージョンによって異なります。この節では，デフォルトの DTS の読み込み範囲，読み込み範囲の変更方法について説明します。

また，uCosminexus Business Reporting Processor では，効率良く XBRL 文書を処理するために，次のことができます。この節では，これらの機能の詳細と，機能を使用する場合の設定方法についても説明します。

DTS のメモリキャッシュ

一度読み込んだ DTS をメモリにキャッシュする機能です。DTS のメモリキャッシュは，同じ DTS を基に複数のインスタンス文書を作成する場合などに利用できます。

リンクベースの遅延読み込み

DTS に定義されているリンクベースを最初にすべて読み込むのではなく，必要なときに使用するリンクベースだけを読み込む機能です。リンクベースの遅延読み込みは，検証済みで信頼できるタクソノミー文書を参照する場合などに利用できます。
XBRL2.1 の場合にリンクベースの遅延読み込みをすると，ユーザーアプリケーションが XBRL DOM を操作する順番によって，読み込まれるリンクベースが異なる場合があります。この場合の対処については，「2.5.5　DTS の拡張読み込み（XBRL2.1 の場合）」を参照してください。

メモリキャッシュと遅延読み込みの使い分け

DTS のメモリキャッシュは，ユーザーアプリケーション単位で指定しますが，リンクベースの遅延読み込みは，解析する文書単位で指定できます。
リンクベースの遅延読み込みと DTS のメモリキャッシュの両方を使用するよう指定した場合は，リンクベースの遅延読み込みが実行されます。メモリキャッシュは実行されません。
確実にメモリキャッシュをするには，リンクベースの遅延読み込みを使用しない設定にしてください。

2.5.1 DTS の読み込み範囲と指定方法（XBRL2.0 の場合）

XBRL2.0 の場合の，DTS の読み込み範囲と，読み込み範囲の指定方法について説明します。

DTS 読み込み時には，XBRL 文書に記載されている属性値を基に，参照先の XBRL 文書が読み込まれます。DTS 読み込み時に参照される要素と属性の一覧を次の表に示します。

表 2-3　DTS 読み込み時に参照される要素と属性の一覧（XBRL2.0 の場合）

XBRL 文書種別	参照される要素と属性	読み込まれる XBRL 文書
インスタンス文書	<xbrli:group> 要素の xsi:schemaLocation 属性	タクソノミー本体
	<link:linkbaseRef> 要素の xlink:href 属性	リンクベース
タクソノミー文書	<xsd:import> 要素の schemaLocation 属性	タクソノミー本体
	<link:linkbaseRef> 要素の xlink:href 属性	リンクベース

次に，各 XBRL 文書の読み込み範囲とその指定方法を説明します。

（1）インスタンス文書を解析する場合

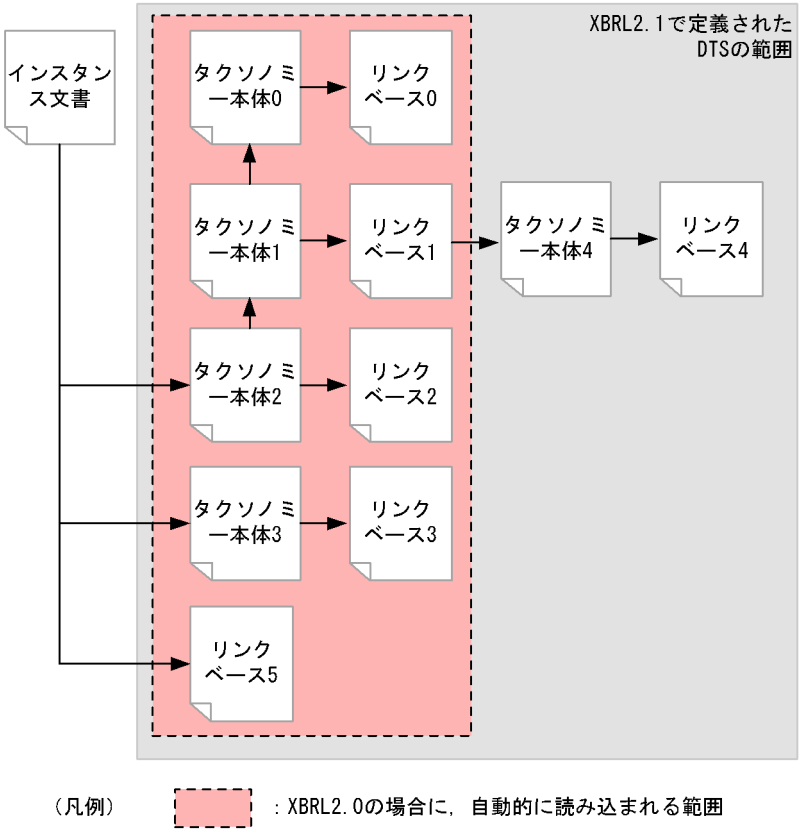
XBRL2.0 のインスタンス文書を解析する場合，次の順番で DTS が読み込まれます。

1. ルート要素である <xbrli:group> 要素の xsi:schemaLocation 属性値を基に，タクソノミー本体が読み込まれます。
2. タクソノミー本体の <xsd:import> 要素をたどり，参照先のタクソノミー本体が読み込まれます。
   タクソノミー本体が，n 階層から成るタクソノミー本体の場合，基底タクソノミー文書までが読み込まれます。
3. タクソノミー本体に <link:linkbaseRef> 要素が記述されていた場合，参照先のリンクベースが読み込まれます。
4. インスタンス文書に <link:linkbaseRef> 要素が記述されていた場合，参照先のリンクベースも DTS の一部として読み込まれます。

また，XBRL International によって提供される XML スキーマ定義では，リンクベースの定義に使用する <link:linkbase> 要素の子要素に，リンクベースの参照に使用する <link:linkbaseRef> 要素を記述できます。しかし，uCosminexus Business Reporting Processor では，<link:linkbase> 要素の子要素として <link:linkbaseRef> 要素が記述されていた場合，警告を通知します。DTS に含めることはできません。

インスタンス文書を解析する場合の DTS の読み込み範囲を次の図に示します。なお，XBRL2.0 の仕様では DTS の範囲が明記されていないため，XBRL2.1 で明記された DTS の範囲と比較して，説明します。

図 2-5　インスタンス文書を解析する場合の DTS の読み込み範囲（XBRL2.0 の場合）



この図は，インスタンス文書の xsi:schemaLocation 属性に，「タクソノミー本体 2」および「タクソノミー本体 3」が指定されている例です。インスタンス文書の <link:linkbaseRef> 要素には，「リンクベース 5」への参照が指定されています。

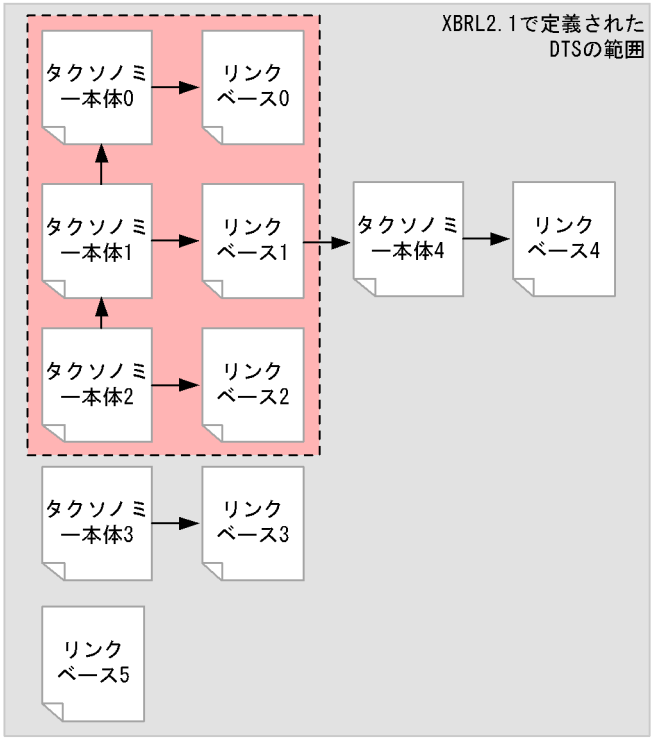
この例では，「タクソノミー本体 2」の基底タクソノミー文書である「タクソノミー本体 0」，および「タクソノミー本体 0」が参照している「リンクベース 0」が読み込まれます。ただし，「リンクベース 1」が参照している「タクソノミー本体 4」は，DTS には含まれません。

(2) タクソノミー文書を解析する場合

XBRL2.0 のタクソノミー文書だけを解析する場合，基点となるタクソノミー本体をすべて指定する必要があります。基点となるタクソノミー本体とは，DTS の読み込み時に出発点となるタクソノミー本体のことです。

タクソノミー文書を解析する場合の DTS の読み込み範囲を次の図に示します。なお，XBRL2.0 の仕様では DTS の範囲が明記されていないため，XBRL2.1 で明記された DTS の範囲と比較して，説明します。

図 2-6　タクソノミー文書を解析する場合の DTS の読み込み範囲（XBRL2.0 の場合）



（凡例）　：XBRL2.0の場合でタクソノミー本体2を指定したときに，
自動的に読み込まれる範囲

この図は，「タクソノミー本体 2」を読み込んだ場合の例です。この例では，「タクソノミー本体 3」は読み込まれません。「タクソノミー本体 3」を DTS に含めたい場合，「タクソノミー本体 2」および「タクソノミー本体 3」を基点となるタクソノミー本体として指定する必要があります。

(3) 読み込み範囲の指定方法

タクソノミー文書は，複数の階層に拡張できます。通常は，タクソノミー文書が複数の階層に拡張されている場合，いちばん基となった基底タクソノミー文書までが読み込まれます。

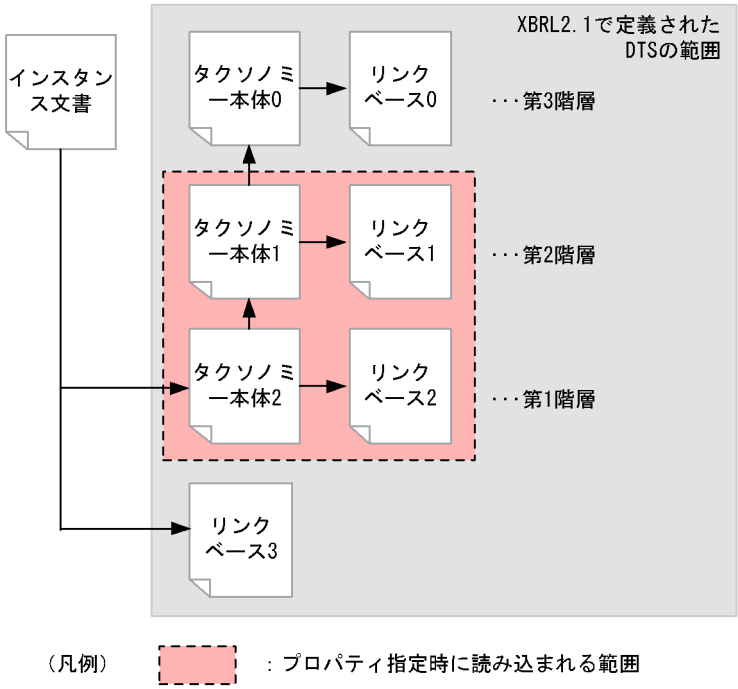
しかし，多くの階層に拡張されたタクソノミー文書を読み込むには，大量のメモリやリソースが必要になります。また，初めて読み込む DTS が予想以上に多くの階層から構成されていてメモリが不足するおそれもあります。

このため，読み込むタクソノミー文書の階層の上限を指定できます。指定した上限値に達した場合，エラーとなり，処理が中断されます。

階層の数え方

タクソノミー文書の階層は，インスタンス文書が参照するタクソノミー文書から基底タクソノミー文書までを数えます。
2 階層まで読み込む指定をした場合，次の図に示す範囲が読み込まれます。

図 2-7　DTS として読み込むタクソノミー文書の階層の数え方



この例では，基底タクソノミー文書は「タクソノミー本体 0」になりますが，「タクソノミー本体 0」は第 3 階層であるため DTS として読み込まれません。
なお，リンクベース 3 は，インスタンス文書からの拡張なので，タクソノミー文書の階層には含まれません。

指定するプロパティ

タクソノミー文書を読み込む階層の上限を指定するには，次のプロパティに階層の最大数を指定します。デフォルトは 2 です。
cbrp.parse.maxTaxonomyExtension
プロパティの指定方法については，「4.16.1　プロパティの指定方法と優先順位」を参照してください。

2.5.2　DTS の読み込み範囲と指定方法（XBRL2.1 の場合）

XBRL2.1 の場合の，DTS の読み込み範囲と，読み込み範囲の指定方法について説明します。

XBRL2.1 の場合，XBRL2.1 仕様で明記された DTS の範囲すべてが読み込まれます。ま

た，XBRL2.0 の場合と大きく異なる点は，リンクベースで <link:loc> 要素などを使用して参照されたタクソノミー本体やリンクベースも，DTS として読み込まれることです。

DTS 読み込み時に参照される要素と属性の一覧を次の表に示します。

表 2-4　DTS 読み込み時に参照される要素と属性の一覧（XBRL2.1 の場合）

XBRL 文書種別	参照される要素と属性	読み込まれる XBRL 文書
インスタンス文書	<link:schemaRef> 要素の xlink:href 属性	タクソノミー本体
	<link:linkbaseRef> 要素の xlink:href 属性	リンクベース
	<link:roleRef> 要素の xlink:href 属性	タクソノミー本体
	<link:arcroleRef> 要素の xlink:href 属性	タクソノミー本体
タクソノミー本体（タクソノミー文書）	<xsd:import> 要素の schemaLocation 属性	タクソノミー本体
	<xsd:include> 要素の schemaLocation 属性	タクソノミー本体
	<link:linkbaseRef> 要素の xlink:href 属性	リンクベース
リンクベース（タクソノミー文書）	<link:loc> 要素の xlink:href 属性	タクソノミー本体またはリンクベース
	<link:roleRef> 要素の xlink:href 属性	タクソノミー本体
	<link:arcroleRef> 要素の xlink:href 属性	タクソノミー本体

ここでは，各 XBRL 文書の読み込み範囲と，その指定方法について説明します。

XBRL2.1 の場合，読み込み範囲の指定には次の方法があります。両方を同時に指定することもできます。

- 読み込まない XML スキーマ定義をあらかじめ登録する方法
- DTS として読み込む文書の最大数を指定する方法

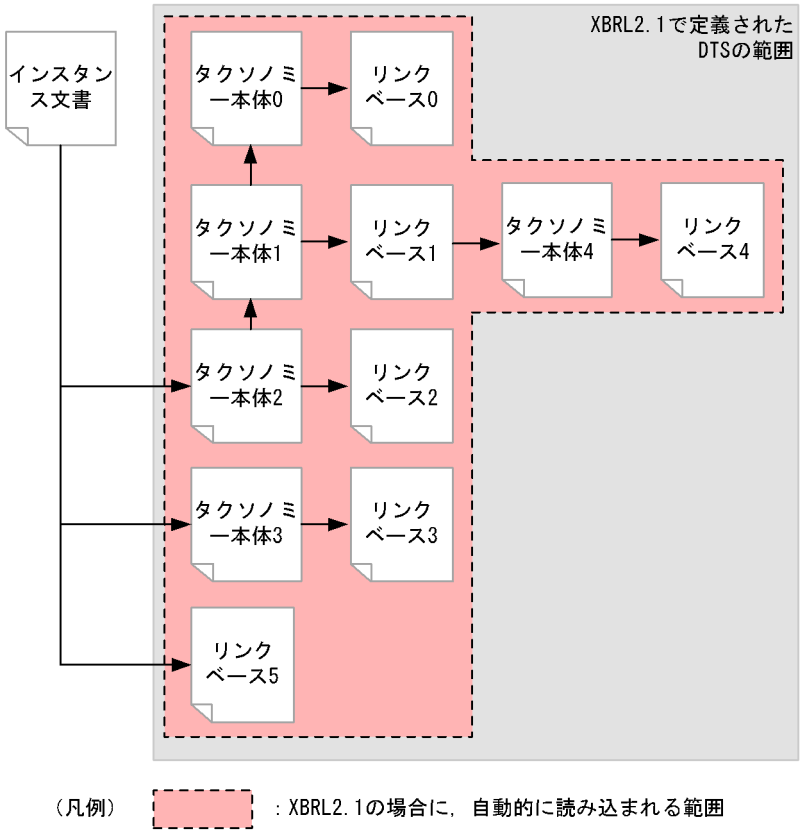
(1) インスタンス文書を解析する場合

XBRL2.1 のインスタンス文書を解析する場合，次の順番で DTS が読み込まれます。

1. XBRL2.1 で新しく追加された <link:schemaRef> 要素，<link:roleRef> 要素，および <link:arcroleRef> 要素を基に，タクソノミー本体が読み込まれます。
2. タクソノミー本体の <xsd:import> 要素をたどり，タクソノミー文書全体（DTS）が読み込まれます。
3. リンクベースで，<link:loc> 要素などを使用してタクソノミー本体やリンクベースが参照されていた場合，参照先のタクソノミー本体やリンクベースも DTS の一部として読み込まれます。

インスタンス文書を解析する場合の DTS の読み込み範囲を次の図に示します。

図 2-8　インスタンス文書を解析する場合の DTS の読み込み範囲（XBRL2.1 の場合）



この図は，インスタンス文書の xsi:schemaLocation 属性に，「タクソノミー本体 2」および「タクソノミー本体 3」が指定されている例です。インスタンス文書の <link:linkbaseRef> 要素には，「リンクベース 5」への参照が指定されています。

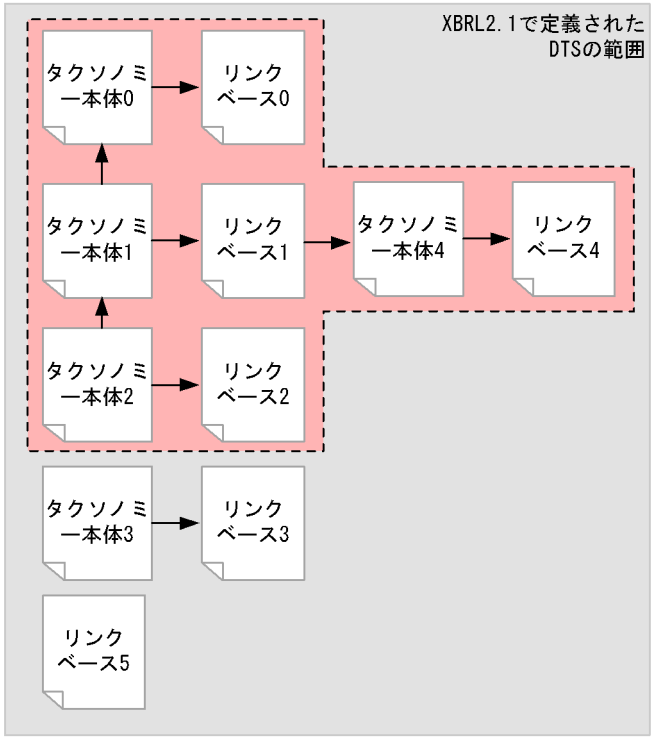
この例では，「リンクベース 1」が参照している「タクソノミー本体 4」，および「タクソノミー本体 4」が参照している「リンクベース 4」が DTS に含まれます。

(2) タクソノミー文書を解析する場合

XBRL2.1 のタクソノミー文書だけを解析する場合，基点となるタクソノミー本体をすべて指定する必要があります。

タクソノミー文書を解析する場合の DTS の読み込み範囲を次の図に示します。

図 2-9 タクソノミー文書を解析する場合の DTS の読み込み範囲（XBRL2.1 の場合）



（凡例） ：XBRL2.1の場合で「タクソノミー本体2」を指定したときに，自動的に読み込まれる範囲

この図は，「タクソノミー本体 2」を指定した場合の例です。「リンクベース 1」から参照されている「タクソノミー本体 4」，および「タクソノミー本体 4」から参照されている「リンクベース 4」が読み込まれます。この例では，「タクソノミー本体 3」は読み込まれません。「タクソノミー本体 3」を DTS に含めたい場合，「タクソノミー本体 2」および「タクソノミー本体 3」を指定する必要があります。

(3) 読み込み範囲の指定方法－読み込まない XML スキーマ定義を登録する

DTS に含めない XML スキーマ定義をあらかじめ登録する方法について説明します。

タクソノミー文書は，<xsd:import> 要素を使用して拡張されるため，通常は <xsd:import> 要素で参照されるタクソノミー本体はすべて DTS として読み込まれます。しかし，<xsd:import> 要素は，タクソノミー本体だけでなく，XML スキーマ定義を指定するためにも使用されます。XML スキーマ定義は，タクソノミー本体の定義や検証には必要ですが，インスタンス文書の操作には必要ありません。

このような，インスタンス文書の操作に必要のない XML スキーマ定義をあらかじめ登録

することで，不要な XML スキーマの読み込みを避けられます。

DTS として読み込まれない XML スキーマ定義

次に示す名前空間に属する XML スキーマ定義は DTS に含まれないため，登録する必要はありません。

XBRL のスキーマ定義（XBRL2.0 の場合）

- http://www.xbrl.org/2001/instance
- http://www.xbrl.org/2001/XLink/xbrllinkbase

XBRL のスキーマ定義（XBRL2.1 の場合）

- http://www.xbrl.org/2003/instance
- http://www.xbrl.org/2003/linkbase
- http://www.xbrl.org/2003/XLink

XML Schema のスキーマ定義

- http://www.w3.org/1999/xlink
- http://www.w3.org/2001/XMLSchema-instance
- http://www.w3.org/2001/XMLSchema
- http://www.w3.org/XML/1998/namespace

指定するプロパティ

DTS として読み込まない XML スキーマ定義を登録するには，次のプロパティに，XML スキーマ定義の名前空間を指定します。デフォルトはなし（null）です。

cbrp.parse.DTS.ignoringSchemas

プロパティの指定方法については，「4.16.1　プロパティの指定方法と優先順位」を参照してください。

（4） 読み込み範囲の指定方法－読み込む文書の最大数を指定する

DTS として読み込む文書の最大数を指定する方法について説明します。

XBRL2.1 の場合，XBRL2.0 の場合に比べて，DTS に含まれるタクソノミー文書の範囲が広くなります。また，リンクベースから参照されるタクソノミー本体も DTS として読み込まれるため，同じ階層を構成する文書数が XBRL 文書の内容によって異なります。そのため，XBRL2.1 の場合は，読み込む文書の最大数を指定することで，読み込み範囲を指定できます。指定した上限値に達した場合，エラーとなり，処理が中断されます。

指定するプロパティ

DTS として読み込む文書の最大数を指定するには，次のプロパティに文書の最大数を指定します。デフォルトは 0（無制限）です。

cbrp.parse.DTS.maxDocuments

プロパティの指定方法については，「4.16.1　プロパティの指定方法と優先順位」を参照してください。

2.5.3 DTS のメモリキャッシュ

DTS のメモリキャッシュとは，一度読み込んだ DTS をメモリにキャッシュすることです。同じ DTS を基に複数のインスタンス文書を作成する場合などに，効率良く文書の作成，編集ができます。

ここでは，メモリキャッシュのタイミングと範囲，メモリキャッシュが解放されるタイミング，メモリキャッシュの条件，およびメモリキャッシュの設定方法について説明します。

(1) メモリキャッシュのタイミングと範囲

タクソノミー文書を解析したときに，タクソノミー文書がメモリにキャッシュされます。インスタンス文書から参照されるリンクベースは，メモリにキャッシュされません。

キャッシュされたタクソノミー文書は，一つのアプリケーション内で有効です。実行環境が uCosminexus Application Server の場合，一つの Web アプリケーション内で有効です。

タクソノミー文書の解析と同時に検証をしてエラー（致命的エラーでないエラー）があった場合，メモリキャッシュされません。ただし，エラーがあった場合でもメモリにキャッシュするよう，設定を変更することもできます。

XBRL2.1 の場合は，タクソノミー文書の解析時にタクソノミー文書を検証しない設定ができます。解析時に検証しない設定の場合，未検証のタクソノミー文書がメモリにキャッシュされます。メモリにキャッシュされている未検証のタクソノミー文書を検証して，キャッシュを上書きすることもできます。

キャッシュされたあとで同じタクソノミー文書を検証した場合，キャッシュされている内容は使用しないで再度解析されます。検証の結果，エラーがないとメモリキャッシュの内容が上書きされます。エラーがあると，メモリキャッシュの内容は変更されません。

(2) メモリキャッシュが解放されるタイミング

メモリキャッシュは次のどちらかのタイミングで解放されます。

- cbrp.cache.maxTaxonomyNum で指定した数を超えたタクソノミー文書を読み込んだ場合，最も古い時刻にキャッシュされたタクソノミー文書がメモリキャッシュから削除されます。
- XBRLDocumentManager オブジェクトの clearCache() メソッドが呼ばれた場合，すべてのタクソノミー文書がメモリキャッシュから削除されます。

(3) メモリキャッシュの条件

メモリキャッシュされたタクソノミー文書を使用する場合，インスタンス文書に指定されているタクソノミー文書の URI を基に，同じタクソノミー文書かどうかが判断されます。そのため，インスタンス文書では次の点に注意してください。

XBRL2.0 の場合

メモリキャッシュを使用するインスタンス文書では，xsi:schemaLocation 属性に同じ値を指定してください。

XBRL2.1 の場合

インスタンス文書の <link:schemaRef> 要素に同じ値を指定するか，または XBRLInstanceDocumentBuilderFactory の setExternalSchemaRefs() メソッドで，足りないタクソノミー文書の URI を指定してください。

(4) メモリキャッシュの指定方法

メモリキャッシュを使用する場合，次のプロパティを指定できます。ここでは，指定できるプロパティについて説明します。プロパティの指定方法については，「4.16.1　プロパティの指定方法と優先順位」を参照してください。

- メモリキャッシュの使用
  タクソノミー文書をメモリにキャッシュするかどうかを次のプロパティに指定します。デフォルトは，true（キャッシュする）です。
  cbrp.cache.enable
- メモリにキャッシュするタクソノミー文書数
  キャッシュするタクソノミー文書数の上限を指定します。指定した上限値までキャッシュされると，最も古い時刻にキャッシュされたタクソノミー文書が一つ削除され，新しいタクソノミー文書がキャッシュされます。デフォルトは 3 です。
  cbrp.cache.maxTaxonomyNum
- エラーがあるタクソノミー文書のキャッシュ
  タクソノミー文書の解析結果にエラー※があった場合，メモリにキャッシュするかどうかを次のプロパティに指定します。
  cbrp.cache.registerErrorDocument
  デフォルトは，false（キャッシュしない）です。
  通常は，エラーがあるタクソノミー文書は訂正して再検証するため，メモリにキャッシュされません。この設定を true にすると，エラーを訂正できないタクソノミー文書をメモリにキャッシュできるため，解析時間を短縮できます。

  注※

  ここでいう「エラー」とは，解析結果として通知される内容（警告，エラー，致命的エラー）のうちの「エラー」を指します。この場合，エラーハンドラでは，error() メソッドが呼ばれます。
  「警告」の場合は，このプロパティの値に関係なくキャッシュされます。
  「致命的エラー」の場合は，このプロパティの値に関係なくキャッシュされません。

2.5.4 リンクベースの遅延読み込み

リンクベースの遅延読み込みとは，タクソノミー文書の解析時に，必要なリンクベース

だけを読み込むことです。解析時にすべてのリンクベースを読み込む場合に比べて，効率良く文書を処理できます。

検証済みであり信頼できるタクソノミー文書を使用する場合，すべてのリンクベースの解析が必要とは限りません。例えば，あるアイテムのラベルを知りたい場合，ラベルリンクだけを解析すれば，ラベルの内容を知ることができます。計算リンクなどの解析は必要ありません。このような場合に，リンクベースの遅延読み込みが利用できます。なお，タクソノミー文書を検証する場合，XML スキーマ検証やリンク検証を実行するため，リンクベースの遅延読み込みは使用できません。

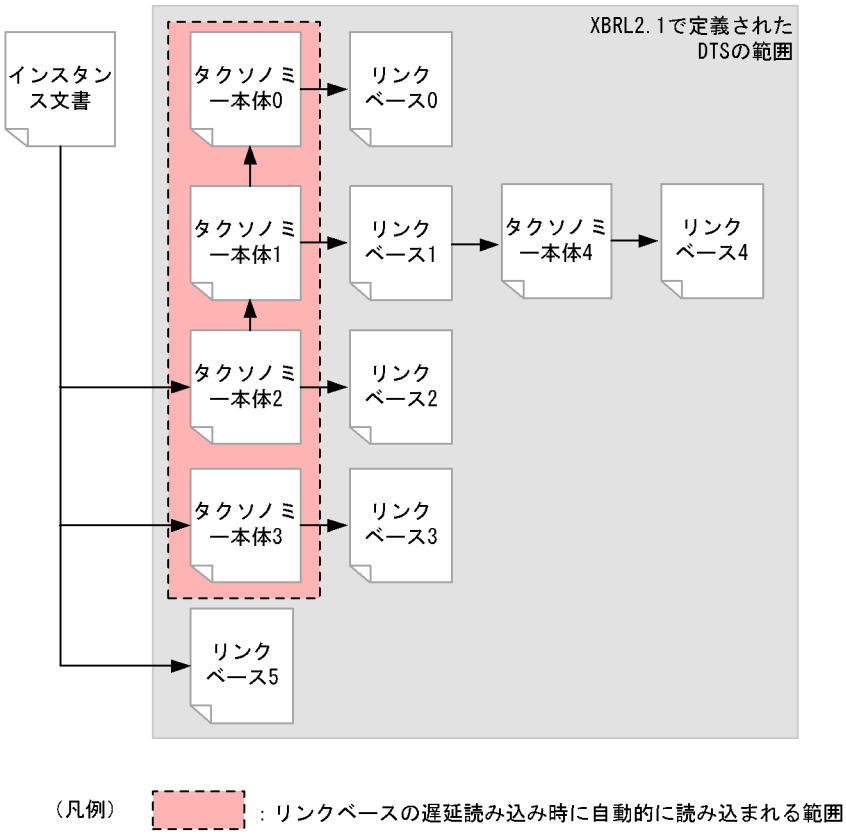
ここでは，リンクベースの遅延読み込み時に最初に読み込まれる範囲，および遅延読み込みの指定方法について説明します。

XBRL2.1 の場合にリンクベースの遅延読み込みを使用すると，DTS の構成によっては，リンクベースが読み込まれるたびに，参照するタクソノミー本体やリンクベースが増加する場合があります。異なるタクソノミー文書が参照されると DTS の内容も変更されてしまいます。この場合，あらかじめタクソノミー文書を登録しておくことで，DTS が変更されるのを防げます。詳細については，「2.5.5　DTS の拡張読み込み（XBRL2.1 の場合）」を参照してください。

（1） 読み込まれる範囲

リンクベースの遅延読み込みを使用した場合に，読み込まれる範囲を次の図に示します。

図 2-10　リンクベースの遅延読み込み時に読み込まれる範囲



なお，XBRL2.1 の場合，リンクベースの遅延読み込みでは，インスタンス文書内の脚注リンクを遅延読み込みできません。

XBRL2.0 の場合の，リンクベースの遅延読み込み時に最初に参照される要素と属性の一覧を次の表に示します。

表 2-5　リンクベースの遅延読み込み時に最初に参照される要素と属性の一覧（XBRL2.0 の場合）

XBRL 文書種別	参照される要素と属性	読み込まれる XBRL 文書
インスタンス文書	<xbrl:group> 要素の xsi:schemaLocation 属性	タクソノミー本体
タクソノミー本体	<xsd:import> 要素の schemaLocation 属性	タクソノミー本体

XBRL2.1 の場合の，リンクベースの遅延読み込み時に最初に参照される要素と属性の一覧を次の表に示します。

表 2-6　リンクベースの遅延読み込み時に最初に参照される要素と属性の一覧（XBRL2.1 の場合）

XBRL 文書種別	参照される要素と属性	読み込まれる XBRL 文書
インスタンス文書	<link:schemaRef> 要素の xlink:href 属性	タクソノミー本体
	<link:roleRef> 要素の xlink:href 属性	タクソノミー本体
	<link:arcroleRef> 要素の xlink:href 属性	タクソノミー本体
タクソノミー本体（タクソノミー文書）	<xsd:import> 要素の schemaLocation 属性	タクソノミー本体
	<xsd:include> 要素の schemaLocation 属性	タクソノミー本体

（2）遅延読み込みの指定方法

リンクベースの遅延読み込みを使用するかどうかは，次のプロパティを指定します。デフォルトは，false（使用しない）です。

cbrp.parse.delayLoad

プロパティの指定方法については，「4.16.1　プロパティの指定方法と優先順位」を参照してください。

2.5.5　DTS の拡張読み込み（XBRL2.1 の場合）

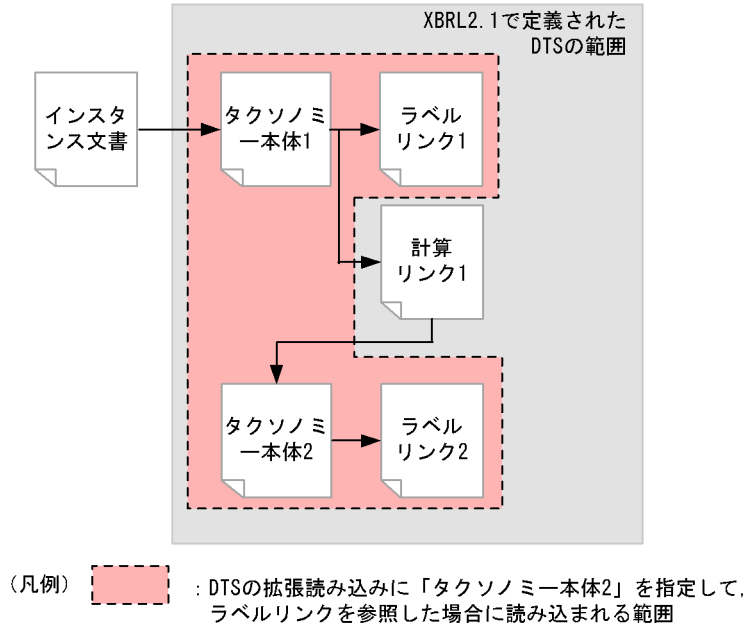
XBRL2.1 の場合でリンクベースの遅延読み込みをするとき，参照したリンクベースから，さらに新しいタクソノミー本体やリンクベースが見つかり，拡張されていくことがあります。そのため，タクソノミー本体やリンクベースの読み込み順序によって，つまりユーザーアプリケーションが XBRL DOM を操作する順序によって，メソッドの実行結果が異なることがあります。

この場合，DTS の拡張読み込みのプロパティであらかじめ参照するタクソノミー本体を指定することで，XBRL DOM の操作順序に関係なく，常に同じ結果を得ることができます。

（1）読み込まれる範囲

DTS の拡張読み込みを指定した場合に読み込まれる範囲の例として，ラベルリンクを参照するときに読み込まれる範囲を次の図に示します。この例では，リンクベースの遅延読み込みをして，DTS の拡張読み込みに「タクソノミー本体 2」を指定していることを想定しています。

図 2-11　ラベルリンクを参照するときに読み込まれる範囲



この例では，DTS の拡張読み込みのプロパティで「タクソノミー本体 2」を指定しているため，「タクソノミー本体 2」も 1 回目の読み込み時に DTS として読み込まれます。この状態でラベルリンクを参照した場合，「ラベルリンク 1」および「ラベルリンク 2」の両方が参照できます。

なお，DTS の拡張読み込みのプロパティを指定しない場合，DTS として読み込まれる文書は「タクソノミー本体 1」だけです。この状態でラベルリンクを参照した場合，「ラベルリンク 1」だけが参照できます。「ラベルリンク 2」は参照できません。

(2) 拡張読み込みの指定方法

拡張読み込みをする場合，次のプロパティを指定します。デフォルトは null（拡張読み込みをしない）です。

cbrp.parse.DTS.externalSchemaRefs

また，このプロパティを使用すると，インスタンス文書から記述が漏れているタクソノミー本体を，DTS の一部として読み込むこともできます。ただし，このプロパティで指定したタクソノミー本体は，アプリケーション内で扱うすべての XBRL 文書で常に読み込まれますので，DTS の構成を把握した上で必要なタクソノミー本体を指定してください。

プロパティの指定方法については，「4.16.1　プロパティの指定方法と優先順位」を参照してください。

2.6 リンクの上書き（XBRL2.0 の場合）

XBRL2.0 の場合，あるタクソノミー文書で定義された，計算リンクなどのリンク（アーク）を上書きすることで，リンク構造を拡張できます。**アーク**とは，拡張リンクでつながれたリソース間の関係を示すものです。

XBRL2.0 の場合，uCosminexus Business Reporting Processor は，リンクの種別ごとにリンクを上書きして，XBRL DOM を作成します。ここでは，リンクの種類ごとに，リンクが上書きされる条件や注意事項などについて説明します。

なお，子要素から親要素への経路が二つ以上あり，共に同一 priority 属性値を持つ場合，uCosminexus Business Reporting Processor は，リンクの上書きを判断できません。このようなリンク構造は，uCosminexus Business Reporting Processor では処理できません。

2.6.1 関係リンクの上書き

ここでは，XBRL2.0 の場合の，関係リンクの上書きについて説明します。

(1) リンクの上書き

関係リンクの場合，アーク型要素が上書きされます。

- 同一リンクが複数記述されていた場合，priority 属性の値が大きい数値のリンクが優先され，上書きされます。
- 同じ priority 属性の値を持つ同一リンクが複数記述されていた場合，エラーとなります。
- アーク型要素の use 属性が prohibited の場合，リンクは上書きされません。上書きするリンクが無効になります。
- 上書きするリンクがない場合，検証時に警告され，無視されます。

(2) リンクが上書きされる条件

リンクは，同一リンクと判断された場合に上書きされます。同一リンクと判断されるのは，次の条件をすべて満たす場合です。

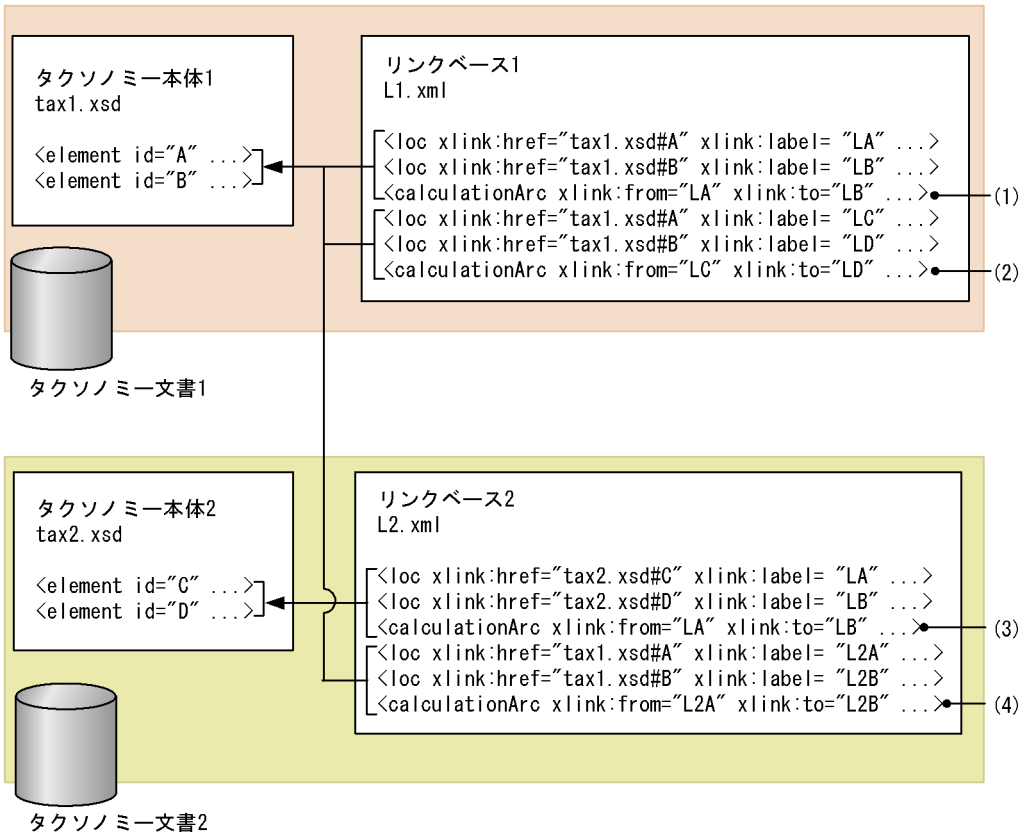
- アーク型要素に記述された xlink:from 属性，および xlink:to 属性が示すロケータ型要素が，同じタクソノミー本体の要素を指している。
  ロケータ型要素が複数の弧を指している場合，同じ数の弧を持つ必要があります。
- xlink:arcrole 属性の値で示される目的が同じ場合である。
  XBRL2.0 の場合，アーク要素は双方向で記述する必要があります。uCosminexus Business Reporting Processor は，双方向の記述を一つの目的と判断します。アークロールの目的を次の表に示します。

表 2-7　アークロールの目的

項番	目的	アークロールの指定
1	親子階層	http://www.xbrl.org/linkprops/arc/parent-child http://www.xbrl.org/linkprops/arc/child-parent
2	同一要素	http://www.xbrl.org/linkprops/arc/dimension-element http://www.xbrl.org/linkprops/arc/element-dimension
3	その他	指定値以外。 アークロールが未知になるため，uCosminexus Business Reporting Processor では片方向のリンクしか記述されていないと判断されます。

関係リンクの同一リンクの判断を次の図に示します。

図 2-12　関係リンクの同一リンクの判断



この例では，(1)，(2) および (4) が示す先は同じなので，ラベル名（xlink:label 属性の値）が異なっても同一リンクとして判断されます。しかし，(3) は，(1) と同じラベル名（LA および LB）を使っていますが，示す先が異なるため別のリンクとして判断されます。

(3) 注意事項

ここでは，リンクの上書きに関する次の注意事項を示します。

- アークロールの違いについて
- priority 属性の値が同じで use 属性の値が prohibited の場合

(a) アークロールの違いについて

アークロールが異なれば，別のリンクとみなされ，リンクは上書きされません。しかし，アークロールに指定されている値が異なっていても目的が同じであれば，リンクは上書きされます。

- アークロールの目的が異なる例
  アークロールの目的が異なる例を次の図に示します。なお，この文書では，xlink:from 属性および xlink:to 属性のラベルの先は，同じタクソノミー本体の要素を参照することを仮定しています。

図 2-13　アークロールの目的が異なる例

```
<calculationLink xlink:type="extended">
(中略)
  <calculationArc xlink:type="arc"
      xlink:from="x1" xlink:to="x2"
      xlink:arcrole="http://www.xbrl.org/linkprops/arc/parent-child"
      weight="1.0" use="optional" priority="0" ...中略... />
</calculationLink>
<calculationLink xlink:type="extended">
(中略)
  <calculationArc xlink:type="arc"
      xlink:from="x1" xlink:to="x2"
      xlink:arcrole="http://www.xbrl.org/linkprops/arc/element-dimension"
      weight="1.0" use="optional" priority="1" ...中略... />
</calculationLink>
```

上下のアーク要素はアークロールの目的が異なるため上書きされない。

上記の文書では，上下のアーク型要素に指定されているアークロールの目的が異なります。そのため，同じタクソノミー本体の要素を参照していても上書きされないで，別のアーク型要素として判断されます。

- アークロールに指定されている値が異なるが，目的が同じである例
  アークロールに指定されている値が異なるが，目的が同じである例を次の図に示します。なお，この文書では，xlink:from 属性および xlink:to 属性のラベルの先は，同じタクソノミー本体の要素を参照することを仮定しています。

図 2-14　アークロールに指定されている値が異なるが，目的が同じである例

```
<calculationLink xlink:type="extended">
(中略)
  <calculationArc xlink:type="arc"
      xlink:from="x1" xlink:to="x2"
      xlink:arcrole="http://www.xbrl.org/linkprops/arc/parent-child"
      weight="1.0" use="optional" priority="0" ...中略... />
</calculationLink>
<calculationLink xlink:type="extended">
(中略)
  <calculationArc xlink:type="arc"
      xlink:from="x2" xlink:to="x1"
      xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent"
      weight="1.0" use="optional" priority="1" ...中略... />
</calculationLink>
```

priority属性の値が大きいので，下側のアーク要素が上書きされる。

上記の文書では，両方のアークロールが親子階層という目的を示しているため，リンクが上書きされます。

(b) priority 属性の値が同じで use 属性の値が prohibited の場合

リンクを上書きするため，uCosminexus Business Reporting Processor は，まず priority 属性の値をチェックします。同一 priority 属性の値が記述されている場合，エラー（KEBR10024-E）が表示されるので，priority 属性の値を正しく修正してから実行してください。

priority 属性の値が同じ場合の例を次の図に示します。なお，この文書では，xlink:from 属性および xlink:to 属性のラベルの先は，同じタクソノミー本体の要素を参照することを仮定しています。

図 2-15　priority 属性の値が同じ場合の例

```
<calculationLink xlink:type="extended">
(中略)
  <calculationArc xlink:type="arc"
      xlink:from="x1" xlink:to="x2"
      xlink:arcrole="http://www.xbrl.org/linkprops/arc/parent-child"
      weight="1.0" use="optional" priority="0" ...中略... />
</calculationLink>
<calculationLink xlink:type="extended">
(中略)
  <calculationArc xlink:type="arc"
      xlink:from="x2" xlink:to="x1"
      xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent"
      weight="1.0" use="prohibited" ...中略... />
</calculationLink>
```

priority属性の値を省略した場合は，0が仮定される。

上記の文書では，下側の要素では，priority 属性の値が省略されているため，priority 属性値に 0 が仮定され，上側の要素と同じ priority 属性値になります。この場合，uCosminexus Business Reporting Processor がリンクを処理する順序によって，どちらかまたは両方のアーク型要素が無視されます。先に下側の要素が参照された場合は，use="prohibited" が指定されているため，両方のアーク型要素が無視されます。

このような場合，表示されるエラー（KEBR10024-E）に従って，priority 属性の値を正しく修正してから実行してください。

2.6.2　ラベルリンクの上書き

ここでは，XBRL2.0 の場合の，ラベルリンクの上書きについて説明します。

(1) リソース型要素（ラベル要素）の追加 / 上書き

ラベルリンクの場合，リンクではなく，リソース型要素のラベル要素が上書きまたは追加されます。

- 同一ラベル要素が複数記述されていた場合は，ラベル要素を指しているアーク型要素の priority 属性の値が大きいラベル要素を優先し，上書きします。
- 同じ priority 属性の値であった場合はエラーとします。
- ラベル要素を指すアーク型要素の use 属性が prohibited の場合は，そのアーク型要素の記述は検証時に警告され，無視されます。上書きはされないため，すでにタクソノミー本体の要素に関連づけているラベル要素があっても削除されません。

(2) ラベル要素が上書きされる条件

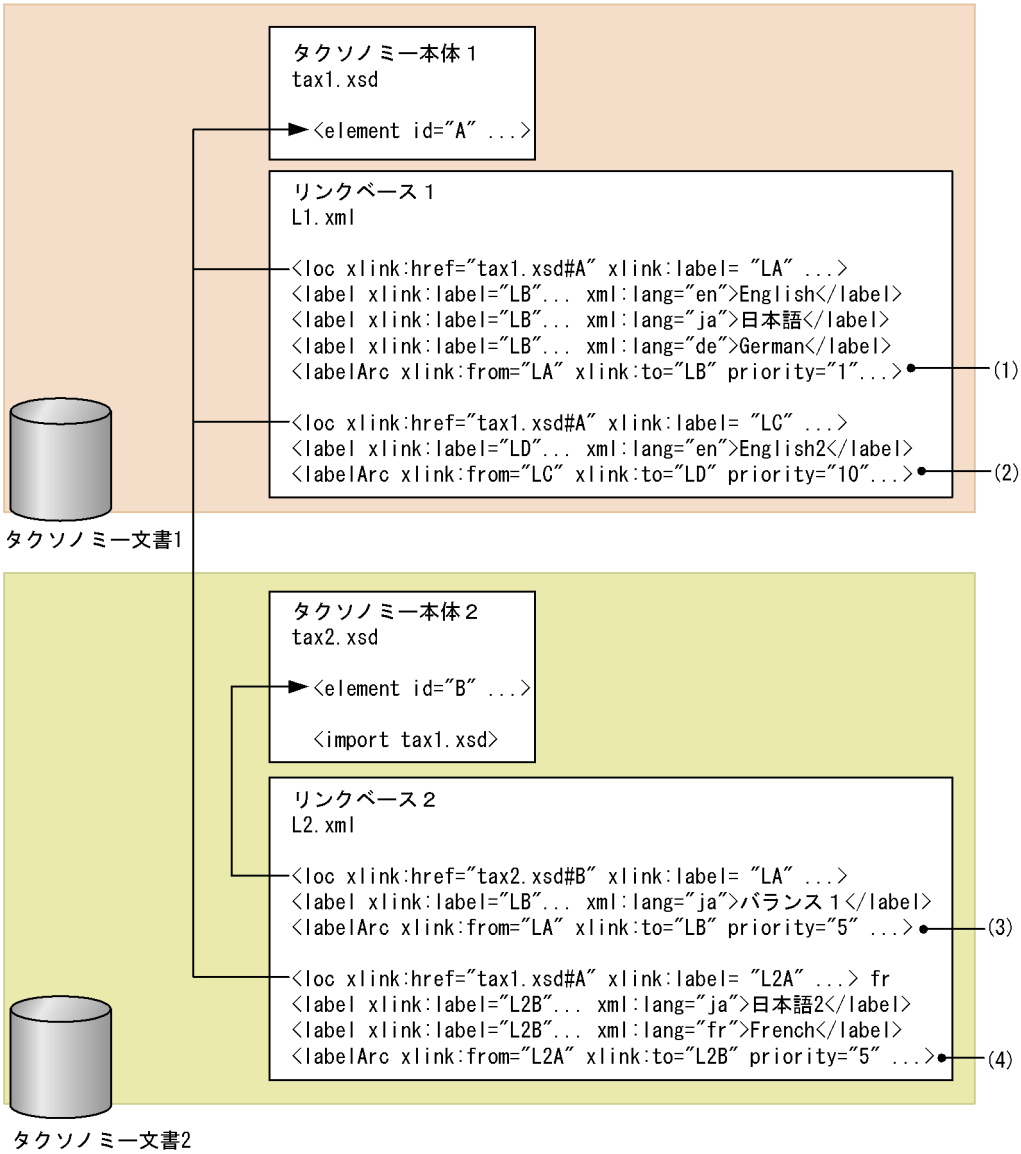
ラベル要素は，ラベル要素が同一であると判断された場合に上書きされます。同一ラベル要素と判断されるのは，次の条件をすべて満たす場合です。

- アーク型要素の xlink:from または xlink:to 属性の先にあるロケータ型要素が，同じタクソノミー本体の要素を指している。
- アーク型要素の xlink:from または xlink:to 属性の先にあるラベル要素が，同じロール（xlink:role 属性の値）および言語（xml:lang 属性の値）である。

ラベル要素のロールおよび言語のどちらか（あるいは両方とも）が異なり，同じタクソノミー本体の要素を指している場合，そのタクソノミー本体の要素にラベル要素のリソースが追加されます。

ラベルリンクの同一リンクの判断を次の図に示します。

図 2-16　ラベルリンクの同一リンクの判断



上記の図では，(1)，(2) および (4) が示すタクソノミー本体の要素は同じため，ラベル名が異なっても同一リンクとして判断されます。この三つのリソース型要素についてロールおよび言語が同じ場合は，ラベル要素が上書きされます。

この例では，ロールを省略していますが，ロールがすべて同じものとして仮定すると，tax1.xsd の id が A のタクソノミー本体の要素に対して，ラベル要素は，English2，日本語 2，German，French となります。

もしロールが異なれば，ラベル要素は上書きされません。ラベル要素は English，日本

語，English2，日本語 2，German，French となり，タクソノミー本体の要素に対して日本語 2 と French が追加されます。

なお，(3) は，(1) と同じラベル名（xlink:label 属性の値）を使っていますが，(1) と示す先（ロケータ要素の xlink:href 属性）が異なるため，別のリンクと判断されます。(1) ではラベル名の示す先が tax1.xsd#A ですが，(3) では tax2.xsd#B であるため，上書きされません。

(3) 注意事項

(a) priority 属性の値が同じで use 属性の値が prohibited の場合

リソース型要素（ラベル要素）を上書きするため，先に priority 属性の値をチェックします。そのため，priority 属性の値が同じ場合，エラーが表示されます。

2.6.3 リファレンスリンクの上書き

ここでは，XBRL2.0 の場合の，リファレンスリンクの上書きについて説明します。

(1) リソース型要素（文献）の追加

リファレンスリンクの場合，タクソノミー本体にリソースが追加されます。同一タクソノミー本体の要素に対して，リソース型要素である <link:reference> 要素が追加されます。

アーク型要素の use 属性が prohibited の場合，そのアーク型要素の記述が無視されます。

(2) リソースが追加される条件

タクソノミー本体が同一であると判断された場合に，該当するタクソノミー本体の要素に記述されていたリソースが追加されます。同一タクソノミー本体と判断されるのは，次の条件を満たす場合です。

- アーク型要素に記述された xlink:from 属性または xlink:to 属性の値が示すロケータ型要素が同じタクソノミー本体の要素を指している。

(3) 注意事項

(a) priority 属性について

priority 属性がアーク型要素に記述されていても無視されます。ただし，文法チェックは実行されるため，priority 属性を記述する場合は文法上の誤りがないように記述する必要があります。

(b) priority 属性の値が同じで use 属性の値が prohibited の場合

priority 属性を指定してもリンクは上書きされないため，use 属性の値が prohibited の

アーク型要素は無視されます。リンクの記述が無効になるため，メッセージは出力されません。

2.6.4 脚注リンクの上書き

ここでは，XBRL2.0 の場合の，脚注リンクの上書きについて説明します。

(1) リソース型要素（脚注）の追加

脚注リンクの場合，アイテムまたはタプル要素が同一であると判断されたときに，該当するアイテムまたはタプル要素に記述されていたリソース（リソース型要素である <link:footnote> 要素）が追加されます。

アーク型要素の use 属性が prohibited の場合，そのアーク型要素の記述が無視されます。

(2) リンクが上書きされる条件

脚注リンクは，アイテムまたはタプル要素が同一であると判断された場合に上書きされます。同一アイテムまたはタプル要素と判断されるのは，次の条件を満たす場合です。

- アーク型要素に記述された xlink:from 属性または xlink:to 属性の値が示すロケータ要素が同じアイテムまたはタプル要素を指している場合。

(3) 注意事項

(a) priority 属性について

priority 属性がアーク型要素に記述されていても無視します。ただし，文法チェックは実行されるため，priority 属性を記述する場合は，文法に誤りのないように記述する必要があります。

(b) priority 属性の値が同じで use 属性の値が prohibited の場合

priority 属性を指定してもリンクは上書きされないため，use 属性の値が prohibited のアーク型要素は無視されます。リンクの記述が無効になるため，メッセージは出力されません。

2.7 リンクの上書き（XBRL2.1 の場合）

タクソノミー文書で定義された，計算リンクなどのリンクを上書きすることで，リンク構造を拡張できます。リンクの上書きでは，タクソノミー文書を拡張したり，インスタンス文書内に目的のリンクベースを指す単純リンク（Simple links）を追加したり，既存のリンクの関係を禁止したりします。**リンクの関係の禁止**とは，リンクの上書きの一種で，すでに定義されているリンクの関係をなくすことです。

XBRL2.1 の場合，uCosminexus Business Reporting Processor は，アーク型要素に記述されているリンクの関係ごとにリンクを上書きして，XBRL DOM を作成します。アーク型要素では，xlink:from 属性と xlink:to 属性を使用して，複数のリソース（ロケータ型要素またはリソース型要素）を指定できます。リンクの関係は，xlink:from 属性と xlink:to 属性の組み合わせごとに作成され，上書きされます。

ここでは，リンクが上書きされる条件，および priority 属性値が同じ場合にどのようにリンクが上書きされるかについて説明します。

2.7.1 リンクが上書きされる条件

ここでは，XBRL2.1 の場合の，リンクが上書きされる条件について説明します。

リンクの上書きは，リンクの関係が同じと判断された場合に実行されます。次の条件をすべて満たしている場合，リンクの関係が同じであると判断されます。

- 同じ種類の拡張リンクであり，拡張リンクの xlink:role 属性の値が同じである。
  同じ種類の拡張リンクとは，同じ名前空間の同じローカル名であるかによって判断されます。
- アーク型要素は同じ種類であり，アーク型要素の xlink:arcrole 属性の値が同じである。
  同じ種類のアーク型要素とは，同じ名前空間の同じローカル名であるかによって判断されます。
- アーク型要素の xlink:from 属性がトラバースされた結果求められる XML フラグメントが同じである。
  詳細は，「2.7.1(2)　トラバーサル先の XML フラグメント」を参照してください。
- アーク型要素の xlink:to 属性がトラバースされた結果求められる XML フラグメントが同じである。
  詳細は，「2.7.1(2)　トラバーサル先の XML フラグメント」を参照してください。
- アーク型要素に記述した属性のうち，XLink によって定義されている属性以外（名前空間識別子が xlink でない属性）が同じである。
  なお，XLink 以外の属性のうち，use 属性と priority 属性は一致する必要はありません。詳細は，「2.7.1(1)　アーク型要素の属性一覧とリンクの上書きで必要な条件」を参照してください。
- アーク型要素下のタイトル型要素は，同じリンクの関係の条件に含まれません。

上書きされる条件が一致すれば，片方のアーク型要素にタイトル要素があり，もう片方になくてもリンクの関係は上書きされます。

(1) アーク型要素の属性一覧とリンクの上書きで必要な条件

リンクが上書きされる条件の一つに，アーク型要素の属性のうち，XLink 以外の属性が一致していることがあります。アーク型要素に指定できる属性一覧と，上書きに必要な条件を次の表に示します。

表 2-8　アーク型要素の属性一覧とリンクの上書きに必要な条件

分類	属性	条件	説明
すべてのアーク型要素共通	xlink:type	○	アーク型要素を示すために，"arc" を指定する必要があります。
	xlink:from	×	指定するラベルの値は一致する必要がありません。
	xlink:to	×	指定するラベルの値は一致する必要がありません。
	xlink:arcrole	○	アークの役割は同じである必要があるため，同じ値を指定します。
	xlink:title	×	一致する必要はありません。
	xlink:show	×	一致する必要はありません。
	xlink:actuate	×	一致する必要はありません。
	order	○	XLink で定義されている属性ではないため，一致する必要があります。
	use	×	リンクの上書きで使用される値であるため，一致する必要はありません。
	priority	×	リンクの上書きで使用される値であるため，一致する必要はありません。
プレゼンテーションリンク	preferredLabel	○	XLink で定義されている属性ではないため，一致する必要があります。 これは，XBRL2.1 だけで使用できる属性です。
計算リンク	weight	○	XLink で定義されている属性ではないため，一致する必要があります。

（凡例）○：値の一致が必要です。　×：値の一致は不要です。

次の場合は，XBRL 文書検証時にエラーとなります。

- xlink:from 属性および xlink:to 属性が同じ値のアーク型要素が複数ある場合
  同一拡張リンク内に，xlink:from 属性および xlink:to 属性の値と同じ値を持つ，別のアーク型要素を記述した場合，致命的エラーとなります。
- xlink:from 属性と xlink:to 属性に同じ値を指定した場合
  xlink:from 属性と xlink:to 属性に同じ値を指定できます。ただし，アーク型要素の種

類によっては同じ値を記述できない場合があり，その場合は，エラーになります。
xlink:from 属性または xlink:to 属性の片方の値が，ほかのアーク型要素と同じ値でもかまいません。

- priority 属性に同じ値を指定した場合
  条件によってはエラーとなります。詳細については，「2.7.2　priority 属性値が同じ場合のリンクの上書き」を参照してください。

(2) トラバーサル先の XML フラグメント

アーク型要素の xlink:from 属性および xlink:to 属性は，同じ拡張リンク内のロケータ型要素またはリソース型要素を参照します。このとき，リンクの関係が同じかどうかの判断材料となる XML フラグメントは，ロケータ型要素ではなく，ロケータ型要素が指す要素になります。そのため，ロケータ型要素が同じタクソノミー本体を指している場合，ロケータ型要素が異なる拡張リンクに記述されていても，リンクの関係は同じであると判断されます。

アーク型要素がリソース型要素を参照する場合でも同様に，XML フラグメントは，リソース型要素ではなく，リソース型要素が指す要素になります。

同じリンクの関係と判断されるロケータ型要素を次に示します。

```
<拡張リンクA xlink:role="http://www.xbrl.org/2003/role/link">
  <loc xlink:href="xxx.xsd#abc" xlink:label="test1" ...>
(中略)
<Arc xlink:from="test1" ....>
</拡張リンクA>

<拡張リンクB xlink:role="http://www.xbrl.org/2003/role/link">
  <loc  xlink:href="xxx.xsd#element(/1/14)" xlink:label="test2" ...>
(中略)
<Arc xlink:from="test2" ....>
</拡張リンクB>
```

この場合，下線部分の xxx.xsd ファイルで，#abc が指す要素と #element(/1/14) が指す要素が同じだった場合，XML フラグメントが同一であると判断され，リンクが上書きされます。

2.7.2　priority 属性値が同じ場合のリンクの上書き

ここでは，XBRL2.1 の場合の，priority 属性値が同じ場合のリンクの上書きについて説明します。

リンクには，優先順位を示す priority 属性を指定できます。priority 属性値は，数値が大きいものほど優先順位が高いとみなされます。ここでは，priority 属性に同じ値が指定されていた場合の，リンクの上書きについて説明します。

priority 属性に同じ値が指定されていた場合，uCosminexus Business Reporting Processor は，use 属性が optional か prohibited かを調べます。prohibited の方が優先

され，リンクの関係が禁止されます。一つでも prohibited が指定されている場合は，すべての関係が無効になります。

priority 属性値が同じ場合の例を次に示します。

(1) 両方とも use="optional" が指定されている場合

同じリンク関係を示す関係 A および関係 B で，priority 属性値が同じで use="optional" が指定されている場合を次に示します。

```
<Arc use="optional" priority="0"...中略>  <!--関係A-->
  <title>xxxx</title>
</Arc>
<Arc use="optional" priority="0"...中略>  <!--関係B-->
```

この場合，どちらか片方のリンクが無効になり，メッセージが出力されます。どちらのリンクが有効になるかは，メッセージ（KEBR10074-W）に出力されます。両方とも同じリンク関係を示すため，XBRL DOM の内容は変更されません。

(2) 片方に use="optional"，もう片方に use="prohibited" が指定されている場合

同じリンク関係を示す関係 A および関係 B で，両方とも priority 属性値が同じで，片方に use="optional"，もう片方に use="prohibited" が指定されている場合を次に示します。

```
<Arc use="optional" priority="0"...中略>  <!--関係A-->
  <title>xxxx</title>
</Arc>
<Arc use="prohibited" priority="0"...中略>  <!--関係B-->
```

この場合，priority 属性値が同じですが，片方に use="prohibited" が指定されているため，両方の関係が無効になります。エラーメッセージは出力されません。

同じリンクの関係が三つ以上ある場合，use="prohibited" が指定されている関係が一つあれば，リンクの関係は禁止されます。

(3) 両方とも use="prohibited" が指定されている場合

同じリンク関係を示す関係 A および関係 B で，priority 属性値が同じで use="prohibited" が指定されている場合を次に示します。

```
<Arc use="prohibited " priority="0" ...中略>  <!--関係A-->
<Arc use="prohibited" priority="0"...中略>  <!--関係B-->
```

この場合，両方のリンクの関係が禁止されるため，関係 A および関係 B で定義されているリンクの関係が両方とも無効になります。エラーメッセージは出力されません。

2.8 リンク構造の参照（XBRL2.1 の場合）

ここでは，アプリケーションからリンク構造を参照する方法について説明します。この機能は，XBRL2.1 の場合だけ使用できます。

W3C DOM を使用してリンク構造を参照するには，タクソノミー本体やリンクベースを検索し，リンク構造を解析して，ノード間の関係を整理する必要があります。

uCosminexus Business Reporting Processor は，ノード間の関係を整理したあとのオブジェクトモデル「XBRLRelationship インターフェース」を提供します。ユーザーアプリケーションから XBRLRelationship インターフェースを使用することで，複雑なリンク関係を解析する必要なく，容易にリンクの関係を参照できます。

XBRLRelationship インターフェースは，ノード間の関係を表現していて，参照元および参照先の両端のノードを参照するメソッドを提供します。また，XBRL 文書で使用する XLink の拡張リンクごとに次のインターフェースを提供しています。

表 2-9　拡張リンクの種類と対応するインターフェース

項番	拡張リンクの種類	対応するインターフェース	インターフェースの説明
1	プレゼンテーションリンク	XBRLPresentationRelationship	<link:presentationArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言とプレゼンテーションリンクとの関係関係を表します。
2	計算リンク	XBRLCalculationRelationship	<link:calculationArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言と計算リンクとの関係を表します。
3	定義リンク	XBRLDefinitionRelationship	<link:definitionArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言と定義リンクとの関係を表します。
4	ラベルリンク	XBRLLabelRelationship	<link:labelArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言とラベルとの関係を表します。
5	リファレンスリンク	XBRLReferenceRelationship	<link:referenceArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言とリファレンスリンクとの関係を表します。
6	脚注リンク	XBRLFootnoteRelationship	<link:footnoteArc> 要素で定義された，インスタンス文書のアイテム（またはタプル）と脚注との関係を表します。

サンプルコードについては，「4.6.3　タクソノミー本体とリンクベースの参照（タクソノミー文書の要素の階層を参照する場合）」を参照してください。API の詳細については，

「6. API リファレンス」を参照してください。

2.9 PTVLI の生成（XBRL2.1 の場合）

タクソノミー文書やリンクベースに対して完全な妥当性検証を実行した場合，その検証結果は PTVLI（Post-Taxonomy Validation Linkbase Infoset）として生成されます。PTVLI の詳しい意味については，XBRL の仕様書を参照してください。

PTVLI には，リンク関係が上書きまたは禁止が発生したあとのリンクベースの情報が含まれます。また，検証時に生じたエラーを無視して処理を続行すると，PTVLI は存在する関係から生成されます。

2.9.1 PTVLI の構造

PTVLI の構造は次のとおりです。

```
ptv:ptvl
  └─ptv:arc
```

● <ptv:ptvl>
文書のルート要素です。xmlns および xsi:schemaLocation 属性に，ptv の名前空間の情報を付加して出力します。

● <ptv:arc>
<ptv:ptvl> の子要素です。リンク（アーク型要素）の情報がリンクの数だけ格納されます。リンク関係が一つもない場合（リンクの禁止によって削除された場合も含む）は作成されません。
<ptv:arc> 要素の属性は次のとおりです。

表 2-10 PTVLI の <ptv:arc> 要素の属性

属性	出力内容
linkType	リンクの種類を次のように出力します。 • label：ラベルリンク • reference：リファレンスリンク • definition：定義リンク • presentation：プレゼンテーションリンク • calculation：計算リンク
arcRole	アーク型要素のアークロール属性の値を出力します。
extRole	PTVLI に変換する前のアーク型要素が属する，拡張リンク上の xlink:role 属性値を出力します。
fromPath	xlink:from 属性の参照先を出力します。 アーク型要素の参照先であるロケータ型要素の xlink:href 属性の値のうち，フラグメントを除いた URI を参照先要素宣言の targetNamespace 属性の値に置換して出力します（{名前空間 URI}# フラグメント）。
labelLang	ラベルリンクの場合に，<link:label> 要素の xml:lang 属性の値を出力します。

属性	出力内容
resRole	リソース型要素のロール属性（xlink:role）の値を出力します。 ただし，xlink:to 属性の参照先がリソースでない場合は出力しません。
weight	weight 属性の値を出力します。 ただし，summation-item アーク以外の場合は出力しません。
order	order 属性の値を出力します。order 属性がない場合は 1 を仮定します。
toPath	xlink:to 属性の参照先を出力します。内容は fromPath と同等です。 ただし，リソースの場合は出力しません。
preferredLabel	preferredLabel 属性の値を出力します。

<ptv:arc> 要素の値は次のとおりです。

- ラベルリンク：<link:label> 要素の値
- リファレンスリンク：<link:reference> 要素の値
- それ以外の場合：値は持たない。

2.9.2 PTVLI の生成例

XBRL で次のように定義した場合を例に，PTVLI の生成例を示します。

```
<linkbase
  xmlns="http://www.xbrl.org/2003/linkbase"
  xmlns:xl="http://www.xbrl.org/2001/XLink"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xml:base="../schema"
  xsi:schemaLocation="http://www.xbrl.org/2003/linkbase ../lib/
xbrl-linkbase-2003-12-31.xsd">
  <labelLink xlink:type="extended" xlink:role="http://www.xbrl.org/2003/role/
link">
    <loc xlink:type="locator"
         xlink:href="LinkbaseRefXMLBase.xsd#fixedAssets"
         xlink:label="aaa" />
    <label xlink:type="resource" xlink:label="aaaLabel"
           xlink:role="http://www.xbrl.org/2003/role/label"
           xml:lang="en">Fixed Assets</label>
    <labelArc xlink:type="arc" xlink:from="aaa" xlink:to="aaaLabel"
              xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"/>
  </labelLink>
</linkbase>
```

このように定義された場合，次に示す PTVLI が生成されます。

```
<ptvl xmlns="http://www.xbrl.org/2003/ptv"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.xbrl.org/2003/ptv ../../lib/
ptv-2003-12-31.xsd">
  <arc arcRole="http://www.xbrl.org/2003/arcrole/concept-label"
         extRole="http://www.xbrl.org/2003/role/link"
         fromPath="http://mycompany.com/xbrl/taxonomy#fixedAssets"
         labelLang="en"
         linkType="label"
         resRole="http://www.xbrl.org/2003/role/label">Fixed Assets</arc>
</ptvl>
```

2.10 PTVI の生成（XBRL2.1 の場合）

完全な妥当性検証を実行した結果，インスタンス文書がどう解釈されたかは，PTVI（Post-Taxonomy Validation Infoset）として生成されます。PTVI の詳しい意味については，XBRL の仕様書を参照してください。

PTVI のアイテムには，推測された precision 属性値や，その要素定義の periodType 属性値などが補完されます。

2.10.1 PTVI の構造

PTVI はインスタンス文書と同様の形式で生成されます。

インスタンス文書から PTVI へ含まれる要素と，その内容を次に示します。

表 2-11 インスタンス文書から PTVI へ含まれる要素とその内容

要素	PTVI への反映	PTVI へ含まれる内容
xbrli:xbrl	○	次の情報が含まれます。※1 • 既存の名前空間宣言 • http://www.xbrl.org/2003/ptv の名前空間宣言 接頭辞は「ptv」です。「ptv」が使用済みの場合は，末尾に 0 から始まる整数を付け，接頭辞が一意となるようにします。 • 名前空間 http://www.xbrl.org/2003/ptv の位置情報※2 既存の xsi:schemaLocation 属性の記述に，http://www.xbrl.org/2003/ptv の位置情報が追加されます。
link:schemaRef	○	記述されている要素がすべて含まれます。追加される属性などはありません。
link:linkbaseRef	×	－
link:roleRef	×	－
link:arcroleRef	×	－
既存のアイテム	○	要素宣言の periodType 属性，balance 属性の内容を名前空間 http://www.xbrl.org/2003/ptv の属性で追加します。
タプル	○	記述されている要素がすべて含まれます。追加される属性などはありません。
xbrli:unit とその子孫要素	○	
xbrli:context とその子孫要素	○	
link:footnoteLink とその子孫要素	×	－

要素	PTVI への反映	PTVI へ含まれる内容
推測したアイテム	○	essence-alias アークから推測したアイテムです。

（凡例）○：PTVI へ反映されます。　×：PTVI へ反映されません。　－：該当する内容はありません。

注※ 1　<xbrli:xbrl> 要素の子要素である兄弟要素の並び順は次のとおりです。

1. link:schemaRef
   複数存在する場合は，既存のインスタンス文書での出現順序に従います。
2. xbrli:unit
   複数存在する場合は，既存のインスタンス文書での出現順序に従います。
3. xbrli:context
   複数存在する場合は，既存のインスタンス文書での出現順序に従います。
4. アイテム／タプル
   推測されたアイテムを含め，その要素名（タグ名）で辞書順にソートされます。ただし，タプルの中はソートされないで，既存のインスタンス文書の出現順序に従います。

注※ 2　名前空間 http://www.xbrl.org/2003/ptv のデフォルトの位置情報は ptv-2003-12-31.xsd です。その場合，xsi:schemaLocation 属性値は次のようになります。

```
<xbrli:xbrl ...
    xsi:schemaLocation=" ... http://www.xbrl.org/2003/ptv  ptv-2003-12-31.xsd"
...
```

また，上位のアプリケーションプログラムからスキーマの位置情報が決定できる場合には，前もって位置情報を設定できます。設定方法については，「4.16.2(18) cbrp.parse.validate.full-checking.PTVx.location」を参照してください。

2.10.2　PTVI のアイテムの作成方法

PTVI の数値アイテム（fractionItemType 型を除く）の作成方法には，次の 2 とおりがあります。

- インスタンス文書に記述されている値のままで作成する
- インスタンス文書に記述されている値を，推測した精度に丸めた概数で作成する

どちらの方法で作成するかは，プロパティ cbrp.parse.validate.full-checking.PTVI.rounded-item-value で指定します。

指定方法の詳細については，「4.16.2(19) cbrp.parse.validate.full-checking.PTVI.rounded-item-value」を参照してください。

なお，fractionItemType 型の数値アイテム，または非数値アイテムを作成する場合には，

プロパティ cbrp.parse.validate.full-checking.PTVI.rounded-item-value での指定に関係なく，インスタンス文書に記述されている値で作成します。

(1) 検証に使用した概数を基にアイテムを作成する

検証に使用した概数を基にアイテムを作成した場合の例を次に示します。既存アイテム値の丸め処理や，decimals 属性値から precision 属性値を推測する方法については，「2.4.2(4) 精度の推測と概数【完全な妥当性検証の場合だけ有効】」を参照してください。

上段がインスタンス文書の記述内容，下段がアイテムの生成結果を示します。

【例 1】通常の場合

```
<item contextRef="c1" precision="4" unitRef="u1">12345</item>
```

```
<item contextRef="c1" unitRef="u1" precision="4"
 ptv:periodType="instant" ptv:balance="credit">12350</item>
```

【例 2】小数点以下の値がない数値に対し，decimals 属性値を記述した場合

```
<item contextRef="c1" decimals="2" unitRef="u1">12345</item>
```

```
<item contextRef="c1" unitRef="u1" precision="7"
 ptv:periodType="instant" ptv:balance="credit">12345.00</item>
```

【例 3】指数表記された数値に対し，decimals 属性値を記述した場合

```
<item contextRef="c1" decimals="5" unitRef="u1">0.1E-2</item>
```

```
<item contextRef="c1" unitRef="u1" precision="3"
 ptv:periodType="instant" ptv:balance="credit">0.00100</item>
```

【例 4】有効数字がない場合

この例では「decimals="3"」という記述に対し，対象となる数値が「0.000123」のため，有効数字がありません。そのため，precision 属性値を次のように算出します。

指数部（0）＋小数点以下の0の数を負にした値（-3）＋decimals値（3）＝0

この算出結果によって「precision="0"」が出力されています。この場合，インスタンス文書に記述されている数字のまま PTVI に出力されます。

```
<item contextRef="c1" decimals="3" unitRef="u1">0.000123</item>
```

```
<item contextRef="c1" unitRef="u1" precision="0"
 ptv:periodType="instant" ptv:balance="credit">0.000123</item>
```

【例 5】有効数字がなく，かつ数値が指数表記の場合

指数表記から展開された値が記述されます。

```
<item contextRef="c1" decimals="4" unitRef="u1">0.00123e-3</item>
```

```
<item contextRef="c1" unitRef="u1" precision="0"
 ptv:periodType="instant" ptv:balance="credit">0.00000123</item>
```

【例 6】数値が 0 で，かつ precision 属性値が記述されている場合

この例では「precision="5"」という記述に対し，有効数字がありません。この場合，小数点以下の精度を precision の値と仮定して，「0.00000」とします。

```
<item contextRef="c1" precision="5" unitRef="u1">0.000</item>
```

```
<item contextRef="c1" unitRef="u1" precision="5"
 ptv:periodType="instant" ptv:balance="credit">0.00000</item>
```

【例 7】数値が 0 で，かつ decimals 属性値が記述されている場合

この例では「decimals="5"」という記述に対し，有効数字がありません。この場合，precision 属性値は次のように算出されます。

指数部（0）＋小数点以下の0の数を負にした値（0）＋decimals値（5）＝5

この算出結果によって「precision="5"」と値「0.00000」が出力されています。

```
<item contextRef="c1" decimals="5" unitRef="u1">0.000</item>
```

```
<item contextRef="c1" unitRef="u1" precision="5"
 ptv:periodType="instant" ptv:balance="credit">0.00000</item>
```

【例 8】数値 0 が指数表記されている場合

0 が指数表記されている場合には，次のように decimals 属性値から precision 属性値を推測します。

指数部（0）＋小数点以下の0の数を負にした値（5）＋decimals値（3）＝8

この算出結果によって「precision="8"」と出力されています。

```
<item contextRef="c1" decimals="3" unitRef="u1">0.00E5</item>
```

```
<item contextRef="c1" unitRef="u1" precision="8"
 ptv:periodType="instant" ptv:balance="credit">0.00000000</item>
```

（2） essence-alias アークの検証時に推測されたアイテムを作成する

拡張機能を使って，essence-alias アークの検証時に存在しないアイテムを推測した場合，推測したアイテムは，通常のアイテムと同じように属性値などが補完され，PTVI に作成されます。作成される値は，アイテムの種類によって次のように異なります。

● 数値アイテム（fractionItemType 型を除く）の場合

プロパティ cbrp.parse.validate.full-checking.PTVI.rounded-item-value での設定に従い，インスタンス文書に記述されている値，または検証に用いた概数（丸めた値）で作成されます。

● それ以外のアイテムの場合
インスタンス文書に記述されている値で作成されます。

また，アイテムは必要最低限の属性だけが PTVI に作成されるため，推測元のアイテムに付いていた id 属性などは削除されます。

A → B への essence-alias アークでアイテム A が推測された場合の PTVI の生成結果を次の例に示します。上段はインスタンス文書の記述内容，下段はアイテムの生成結果です。

```
<B id="item_B" contextRef="c1" precision="3" unitRef="u1">123</B>
```

```
<A contextRef="c1" unitRef="u1" precision="3"
 ptv:periodType="instant" ptv:balance="credit">123</A>
<B contextRef="c1" unitRef="u1" precision="3"
 ptv:periodType="instant" ptv:balance="credit">123</B>
```

2.10.3 PTVI の生成例

ここでは，インスタンス文書の記述内容と PTVI の生成例を示します。

インスタンス文書の記述例を次に示します。

```
<xbrl
  xmlns="http://www.xbrl.org/2003/instance"
  xmlns:cnv="http://www.cnv.gov.ar/xbrl/"
  xmlns:ISO4217="http://www.xbrl.org/2003/iso4217">
<schemaRef href="xbrl-cnv-2003-10-19.xsd"/>
  <unit id="u1">
    <measure>ISO4217:USD</measure>
  </unit>
  <context id="c1">
    <entity>
      <identifier scheme="www.cnv.gov.ar/cnv">norne</identifier>
    </entity>
    <period>
      <instant>2001-12-31</instant>
    </period>
  </context>
  <cnv:Creditos contextRef="c1" precision="7" unitRef="u1">50000
  </cnv:Creditos>
  <cnv:BienesdeUso contextRef="c1" precision="7" unitRef="u1">7000
  </cnv:BienesdeUso>
  <cnv:BienesDiversos contextRef="c1" precision="7" unitRef="u1">6000
  </cnv:BienesDiversos>
</xbrl>
```

この記述から，次に示す PTVI が生成されます。

```
<xbrl
  xmlns="http://www.xbrl.org/2003/instance"
  xmlns:ptv="http://www.xbrl.org/2003/ptv"
  xmlns:cnv="http://www.cnv.gov.ar/xbrl/"
  xmlns:ISO4217="http://www.xbrl.org/2003/iso4217">
  <unit id="u1">
    <measure>ISO4217:USD</measure>
  </unit>
  <context id="c1">
    <entity>
      <identifier scheme="www.cnv.gov.ar/cnv">norne</identifier>
    </entity>
    <period>
      <instant>2001-12-31</instant>
    </period>
  </context>
  <cnv:BienesdeUso
    contextRef="c1" precision="7" unitRef="u1"
    ptv:periodType="instant" ptv:balance="debit">7000
  </cnv:BienesdeUso>
  <cnv:BienesDiversos
    contextRef="c1" precision="7" unitRef="u1"
    ptv:periodType="instant" ptv:balance="credit">6000
  </cnv:BienesDiversos>
  <cnv:Creditos
    contextRef="c1" precision="7" unitRef="u1"
    ptv:periodType="instant" ptv:balance="credit">50000
  </cnv:Creditos>
</xbrl>
```

2.11 XBRL 同等性の判定

XBRL2.1 仕様で定義されている「項目とタプルの重複検出に関連する同等性判定」を XBRLEqualUtil クラスで判定できます。ここでは，この同等性判定を行う際の注意事項を次に示します。

2.11.1 コンテキストの s-equal 判定

コンテキストに定義された時期要素の値として，タイムゾーンの記述を省略した場合は，GMT のタイムゾーンを仮定して判定します。

2.11.2 タプルの duplicate 判定

タプル内に定義されたアイテムの v-equal の判定は，essence-alias アークの検証における v-equal の判定と同等の内容になります。この時，xsi:nil="true" のアイテムがある場合，もう一方の対象としているタプルの下にも，同様に xsi:nil="true" で，v-equal になるアイテムがあれば，v-equal 同等と判定されます。

essence-alias アークの検証における v-equal の判定については，「2.4.2(2) essence-alias アークの検証【完全な妥当性検証の場合だけ有効】」を参照してください。

また，essence-alias アークの検証では，プロパティ ” cbrp.parse.validate.full-checking.ignoringZeroPrecision ” に true が設定されている場合には精度を考慮せず数値比較しますが，XBRLEqualUtil クラスにおいては，引数ありのコンストラクタ，または setIgnoringZeroPrecision メソッドに true を指定することで精度を考慮せず数値比較します。

2.11.3 アイテムの s-equal 判定

アイテムが数値アイテムの場合は，下記の場合に s-equal 同等と判定されます。

- fractionItemType 型同士のアイテムで，分子，分母に対する数値の文字列表現がそれぞれ等しい場合 (fractionItemType とほかの型のアイテムを比較した場合は，s-equal 同等となりません)
- 共に値を持たない同士の場合
- その他の数値型で，その数値の文字列表現が等しい場合 (100 と 100.0 と 1e2 はすべて s-equal となりません)

アイテムが非数値アイテムの場合は，下記の場合に s-equal 同等と判定されます。

- 共に値を持たない同士の場合
- 内容の文字列表現がそれぞれ等しい場合

3 XBRL 文書の作成と編集

この章では，XBRL 文書の作成や編集時に実施される ID の採番や名前空間の正規化について説明します。

3.1 タクソノミー文書の新規作成

3.2 インスタンス文書の新規作成と編集

3.3 ID 管理と採番

3.4 名前空間の正規化

3.1 タクソノミー文書の新規作成

com.hitachi.xbrl.simple.taxonomy パッケージを使えば，XBRL2.1 に準拠した単純なタクソノミー文書を作成できます。ただし，作成したタクソノミー文書を編集することはできません。

com.hitachi.xbrl.simple.taxonomy パッケージで作成できるタクソノミー文書の内容を次に示します。

- タクソノミー本体とラベルリンクベースを一つずつ作成する。
- リンクベースはタクソノミー本体ではなく，別文書に記載する。
- ラベルリンクベースの作成を省略できる。ラベルリンクベースだけを作成することはできない。
- アイテム，ラベルおよびそれぞれの関係から構成される文書を作成する。タプルは使用できない。
- アイテムのデータ型は，xbrl-instance-2003-12-31.xsd に定義されているデータ型である。派生したデータ型は使用できない。
- ラベルの内容は単純な文字列値だけである。
- リンクで使用できるロールおよびアークロールは，標準値または拡張元のタクソノミー文書で定義されている値である。新たに追加することはできない。

3.1.1 タクソノミー文書の出力フォーマット

タクソノミー文書の出力フォーマットについて説明します。

(1) タクソノミー文書の作成時に自動作成される内容

タクソノミー文書の作成時に自動作成される文書項目と要素を次に示します。

● タクソノミー文書の作成時に自動作成される文書項目

文書項目	内容
名前空間	タクソノミー文書の作成時に，ユーザーが名前空間を指定できます。一度指定した名前空間は変更できません。 ただし，次に示す名前空間は使用できません。 • http://www.w3.org/XML/1998/namespace • http://www.w3.org/2000/xmlns/ • http://www.w3.org/2001/XMLSchema • http://www.w3.org/2001/XMLSchema-instance • http://www.w3.org/1999/xlink • http://www.w3.org/1999/xhtml • http://www.xbrl.org/2003/instance • http://www.xbrl.org/2003/linkbase • http://www.xbrl.org/2003/XLink

文書項目	内容
接頭辞	ユーザーが接頭辞を指定できます。 ただし，次に示す接頭辞は使用できません。 • デフォルト（""） • xsi • xl • xlink • xbrli • link • xml で始まる接頭辞
ロケーション	ロケーションを拡張子「xsd」のファイル名で指定できます。このファイル名は，リンクベースを作成する場合にも利用されます。

● タクソノミー文書の作成時に自動作成される要素

要素	内容
xsd:schema	タクソノミー本体のルート要素
xsd:annotation	<xsd:appinfo> 要素のコンテナ
xsd:appinfo	<link:linkbaseRef> 要素のコンテナ

次に，ベースとなるタクソノミー文書がない場合とある場合とに分けて，タクソノミー文書の作成例を示します。実際にはインデントはありません。

(a) ベースとなるタクソノミー文書がない場合

ここでは，ベースとなるタクソノミー文書がない場合について，次の条件でタクソノミー文書を作成する例を示します。

- 文書のファイル名：sample_xbrl_2_1.xsd
- 対象名前空間：http://www.hitachi.com/2004/XBRL
- 対象名前空間接頭辞：hi
- ベースとしたタクソノミー（DTS）：なし

この場合，次のようなタクソノミー文書が作成されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.hitachi.com/2004/XBRL"
        xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns:hi="http://www.hitachi.com/2004/XBRL">
    <import namespace="http://www.xbrl.org/2003/instance"
schemaLocation="xbrl-instance-2003-12-31.xsd"/>
    <import namespace="http://www.xbrl.org/2003/linkbase"
schemaLocation="xbrl-linkbase-2003-12-31.xsd"/>
    <annotation>
        <appinfo/>
    </annotation>
</schema>
```

(b) ベースとなるタクソノミー文書がある場合

ここでは，ベースとなるタクソノミー文書がある場合について，次の条件でタクソノミー文書を作成する例を示します。

- 文書のファイル名：sample2_xbrl_2_1.xsd
- 対象名前空間：http://www.hitachi.com/2004/XBRL/2
- 対象名前空間接頭辞：hi
- ベースとしたタクソノミー（DTS）：sample_xbrl_2_1.xsd

この場合，次のようなタクソノミー文書が作成されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.hitachi.com/2004/XBRL/2"
        xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns:hi="http://www.hitachi.com/2004/XBRL/2">
    <import namespace="http://www.xbrl.org/2003/instance"
schemaLocation="xbrl-instance-2003-12-31.xsd"/>
    <import namespace="http://www.xbrl.org/2003/linkbase"
schemaLocation="xbrl-linkbase-2003-12-31.xsd"/>
    <import namespace="http://www.hitachi.com/2004/XBRL"
schemaLocation="sample_xbrl_2_1.xsd"/>
    <annotation>
        <appinfo/>
    </annotation>
</schema>
```

ベースとしたタクソノミーがある場合は，ベースとしたタクソノミー（DTS）の基点となるタクソノミー本体をインポートするため，<xsd:import> 要素を作成します。その際，namespace 属性値には対象名前空間を設定し，schemaLocation 属性値にはファイル名（絶対 URI でない）だけを埋め込みます。なお，指定された対象名前空間が，基点となったタクソノミー本体と同じであった場合には，<xsd:include> 要素を作成します。

ベースとしたタクソノミーがある場合，タクソノミー本体には，次に示す名前空間とその接頭辞が自動的に付加されます。名前空間宣言は追加・削除・変更ができません。

名前空間	接頭辞
http://www.w3.org/2001/XMLSchema	デフォルト
http://www.w3.org/1999/xlink	xlink
http://www.xbrl.org/2003/instance	xbrli
http://www.xbrl.org/2003/linkbase	link

さらに，タクソノミー（DTS）の基点となったリンクベースがある場合は，<link:linkbaseRef> 要素を作成し，そのリンクベースを参照させます。

(2) ユーザーが作成できる内容

タクソノミー文書の作成時に，ユーザーが作成できる要素と関係は次のとおりです。

(a) ユーザーが作成できる要素

タクソノミー文書でユーザーが作成できる要素は，<xsd:element> 要素と <link:label> 要素だけです。それぞれの説明を次に示します。

● <xsd:element> 要素（アイテム）

<xsd:element> 要素では大域定義だけができます。<xsd:element> 要素で追加・設定できる属性を次に示します。

- name 属性
- type 属性
  XBRL の標準型だけ指定できる。拡張したデータ型は指定できない。
- id 属性
  自動的に設定されるが，ユーザーが設定することもできる。id 属性値のデフォルト形式は次のとおり。
  対象名前空間接頭辞 + "_" + name 属性値
- nillable 属性
- abstract 属性
- substitutionGroup 属性
  属性値は「xbrli:item」だけであり，派生した代替グループは作成できない。
- xbrli:balance 属性
- xbrli:periodType 属性

● <link:label> 要素

<link:label> 要素には次の制限があります。

- xlink:label 属性値は自動的に次の形式で付加される。ユーザーが設定することはできない。
  対象名前空間接頭辞 + "_" + name 属性値 + xml:lang 属性値
  拡張リンク内にすでに同じ値がある場合は，末尾に 1 から始まるシーケンシャル番号を付加する。
  一つのラベルに対して，複数のアイテムを割り当てることはできない。
- xlink:role 属性値には，標準値または拡張元のタクソノミー文書で定義されているカスタムロールが設定される。
- xml:lang 属性値については，ISO639 で定義されている言語コードかどうかはチェックされない。

(b) ユーザーが作成できる関係

タクソノミー文書で作成できる関係は次のとおりです。

- 新規作成するコンセプトと新規作成するラベルの，新たな関係
- 既存のコンセプトと新規作成するラベルの，新たな関係
- 既存のコンセプトと既存のラベルとの関係を否定する関係

既存関係と同じ関係や，既存関係のうち優先順位がいちばん高いものと同じ関係を作ることはできません（ベースとなるタクソノミー文書の関係を否定する場合を除く）。

(3) 関係を追加・削除した場合に生じる影響

アイテムとラベルの関係が追加・削除されると，ラベルリンクベースが作成・破棄されます。ラベルリンクベースが作成された場合，ファイル名は次のようになります。

タクソノミー本体のファイル名（拡張子除く） + "_label.xml"

そのとき作成される要素は次のとおりです。

要素	機能制限
link:linkbaseRef	この要素は schema[1]/annotation[1]/appinfo[1] に作成される。属性値には次の制限がある。 xlink:arcrole 属性値 "http://www.w3.org/1999/xlink/properties/linkbase" 固定。 xlink:href 属性値 各リンクベース文書のファイル名だけ（絶対 URI ではない）。 xlink:role 属性値 "http://www.xbrl.org/2003/role/labelLinkbaseRef" 固定。
link:linkbase	ラベルリンクベースのルート要素として作成される。※
link:labelLink	ロールごとに一つだけ作成する。xlink:role 属性値に設定できる内容は次のとおり。 • "http://www.xbrl.org/2003/role/link" • 拡張元のタクソノミー文書で定義されているカスタムロール
link:loc	xlink:label 属性値 下記の値になる。 関連づけたアイテムまたはタプルの名前（対象名前空間接頭辞 + "_" + name 属性値） ラベルを参照する場合には，参照先の xlink:label 属性値と同じ値とする。 拡張リンク内で値が重複する場合には，1 から始まるシーケンシャル番号を末尾に付加する。 xlink:href 属性値 ショートハンドポインタで作成する。参照先のアイテム・タプル・ラベルに id 属性が存在しない場合は，element() scheme で作成する（新規作成するアイテムやタプルでは，id 属性値を省略することはできない）。
link:labelArc	xlink:arcrole 属性値 作成時に指定された値で作成する。 priority 属性値 常に最大となる整数値を設定する。

注※　<link:linkbase> の出力形式は次のとおりです。

```
<link:linkbase xmlns:link="http://www.xbrl.org/2003/linkbase"
               xmlns:xlink="http://www.w3.org/1999/xlink"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="http://www.xbrl.org/2003/linkbase
xbrl-linkbase-2003-12-31.xsd
                           http://www.w3.org/1999/xlink xlink-2003-12-31.xsd">
```

3.1.2 タクソノミー文書を作成する際の注意事項

(1) name 属性値および id 属性値の一意性

作成されたタクソノミー文書中では，次の属性値が一意となります。

● 要素宣言の name 属性値と id 属性値
要素宣言の name 属性値と id 属性値は一意となるため，name 属性値または id 属性値が重複する要素宣言は，文書に追加できません。
同様に，文書中の name 属性値や id 属性値を変更する場合も，既存の属性値と重複する値へは変更できません。

● ラベル要素の id 属性値
ラベルリンクおよびタクソノミー本体は，タクソノミー文書の作成時に，それぞれ別文書として作成されます。id 属性値は要素ごとに管理されるため，各文書内で一意となります。

(2) アイテムおよび関係リンクの作成・追加・削除

アイテムおよび関係リンクの作成・追加・削除に関する注意事項は次のとおりです。

- 新規作成するタクソノミー文書に関係を作成するには，文書中に対象となるアイテムがなくても，新たな関係を追加することで，関係づけられているアイテム，要素宣言および要素を追加します。
- 関係を削除しても，文書中からアイテムなどは削除されません。
- 文書に関係を追加する場合，かつ，関係づけられているアイテムなども文書に追加する場合，name 属性値および id 属性値が重複しないかチェックされます。重複する場合は追加できません。

3.2 インスタンス文書の新規作成と編集

uCosminexus Business Reporting Processor では，インスタンス文書を新規に作成したり，既存のインスタンス文書を編集したりできます。

どちらの場合も，uCosminexus Business Reporting Processor がインスタンス文書を XBRL DOM に展開し，ユーザーアプリケーションが DOM ツリーを操作します。

インスタンス文書を新規に作成する場合，使用するタクソノミー文書を事前に指定することで，タクソノミー文書に沿ったインスタンス文書を作成できます。

インスタンス文書の作成や編集をする場合，完成したインスタンス文書を一度検証することを推奨します。

3.2.1 インスタンス文書の出力フォーマット

ここでは，uCosminexus Business Reporting Processor がインスタンス文書をファイルまたはストリームに出力する場合の，出力フォーマットを示します。XBRL International で提供されるスキーマに従う内容は，このマニュアルでは説明しません。

(1) 名前空間宣言

ここでは，インスタンス文書を新規に作成する場合と，既存のインスタンス文書を編集する場合の，名前空間宣言の出力について説明します。

インスタンス文書を新規に作成する場合

名前空間宣言は，uCosminexus Business Reporting Processor が自動的に標準値を設定します。標準値以外の名前空間宣言を記述するには，API を使用して追加します。自動設定された標準値の名前空間宣言でも，API を使用して名前空間宣言を削除したり変更したりできます。

設定した名前空間と名前空間接頭辞は，インスタンス文書出力時にルート要素に宣言されます。

既存のインスタンス文書を編集する場合

既存のインスタンス文書を編集する場合，要素に指定されている名前空間宣言はルート要素に宣言し直されます。これを名前空間の正規化と呼びます。名前空間の正規化の詳細については，「3.4 名前空間の正規化」を参照してください。

(2) 要素や属性の出力順序

インスタンス文書をファイルに出力する場合，XBRL DOM を操作して作成した，メモリ上の W3C DOM のツリー構造に従って，出力します。インスタンス文書を参照するだけの場合も，インスタンス文書を XBRL DOM へ展開するときのパースオプションによっては，CDATA ノードが Text ノードに展開された W3C DOM となります。ノードの位置や内容は同じでも，ノードのデータ形式が CDATA か Text かによって，バイナリレベル

では差異が生じます。加えて，要素の各属性については，W3C DOM の実装に依存しており，出現順序が変更されるおそれもあります。

そのため，インスタンス文書をいったん XBRL DOM に展開し出力（保存）したインスタンス文書と，展開前のインスタンス文書とのバイナリ互換は保証できません。ただし，XBRL 文書としての意味や構造を変えるものではありません。

(3) タプル型要素（tuple）の子要素

タプル型要素の子要素には，アイテム型要素またはタプル型要素を含むことができます。タクソノミー文書での定義によっては，出現順序や必須要素などを指定できます。

インスタンス文書をファイルに出力する場合，XBRL DOM を操作して作成した，メモリ上の W3C DOM のツリー構造に従い，出力します。このため，子要素を含むタプル型要素を定義したタクソノミー文書を使用する場合は，ユーザーが要素の出現順序や必須要素を意識して，XBRL DOM を操作する必要があります。

(4) 脚注リンク

脚注リンクは，次の要素で構成されます。

- 脚注リンクのコンテナ要素である <link:footnoteLink> 要素
- 脚注を表す <link:footnote> 要素
- アイテムやタプルを参照する <link:loc> 要素
- アイテムやタプルと脚注をつなぐ <link:footnoteArc> 要素

脚注リンクの出力について，次の点に注意してください。

- 脚注を追加する場合は，アイテムやタプルと関係づける
  アイテムやタプルと関係づけていない脚注を追加した場合には，<link:footnoteLink> 要素と <link:footnote> 要素だけが追加されます。この場合，脚注が関係づけられていないため，意味がありません。そのため，脚注をインスタンス文書に追加する場合は，アイテムやタプルと関係づける必要があります。
- 脚注リンク編集時には，すべての要素が再作成される
  <link:footnoteLink> 要素がすでにある場合は，既存の <link:footnoteLink> 要素およびその子孫要素をすべて削除して，新しく一つの <link:footnoteLink> 要素を作成し，その子孫要素に <link:footnote> 要素などを再作成します。これを脚注リンクの再作成と呼びます。脚注リンクの再作成については，「付録 B　脚注リンクの再作成」を参照してください。

(5) グループ要素（group）（XBRL2.0 の場合）

XBRL2.0 の場合，<xbrli:group> 要素は，ルート要素以外でも，任意のコンテナ要素として使用できます。uCosminexus Business Reporting Processor は，既存インスタンス文書のルート要素の子要素として記述されている <xbrli:group> 要素をそのまま出力します。W3C DOM から削除されません。

（6） アイテム（item）またはタプル（tuple）の出力順序（XBRL2.0 の場合）

インスタンス文書に，アイテムやタプルを新規に追加する場合，ルート要素の子要素として追加されます。すでにアイテムやタプルがインスタンス文書内にある場合，新規に追加する要素は，すでにある要素の前後に追加できます。

ルート要素ではない <xbrli:group> 要素の子要素であるアイテムやタプルの前後に，アイテムやタプルを追加した場合は，それらの親要素である <xbrli:group> 要素の子要素として追加されます。また，ルート要素でない <xbrli:group> 要素から，子要素であるアイテムやタプルをすべて削除した場合，削除後にその <xbrli:group> 要素の子要素として，再度アイテムやタプルを追加できません。

3.3 ID 管理と採番

XML Schema では，XML 文書内で要素を一意に保つために xsd:ID 型属性を使用します。XBRL では，この xsd:ID 型属性が持つ一意性を，アイテムとコンテキストを関係づけるためなどに使用しています。このため，インスタンス文書では，各要素の xsd:ID 型属性である id 属性値（ID）を一意にする必要があります。

uCosminexus Business Reporting Processor は，インスタンス文書内の ID を管理し，インスタンス文書の作成や編集時に自動的に ID を付けます。ユーザーは，API を利用して要素同士を関係づけるだけで，インスタンス文書内の ID が一意であるかを意識することなく，インスタンス文書の作成や編集ができます。

ここでは，ID の管理および採番方法について，説明します。

3.3.1 ID の管理と API を使用した要素同士の関係づけ

uCosminexus Business Reporting Processor は，インスタンス文書内の要素に指定されている ID を一元管理します。ユーザーは，ユーザーアプリケーションで API を使用して要素同士の関係づけを行います。

ここでは，一元管理される ID と，API を使用した要素同士の関係づけについて説明します。また，複数インスタンス文書間での ID 管理についても説明します。複数インスタンス文書間での ID 管理は，XBRL2.1 の場合だけできます。

（1）一元管理される ID と API を使用した要素同士の関係づけ

uCosminexus Business Reporting Processor が一元管理する ID を次の表に示します。

表 3-1　一元管理される ID

項番	要素名	属性名	XBRL バージョン
1	<xbrli:xbrl> 要素	id	2.1
2	アイテム型要素	id	2.0 および 2.1
3	タプル型要素	id	2.0 および 2.1
4	<xbrli:numericContext> 要素	id	2.0
5	<xbrli:nonNumericContext> 要素	id	2.0
6	<xbrli:context> 要素	id	2.1
7	<xbrli:unit> 要素	id	2.1
8	<link:footnoteLink> 要素	id	2.1
9	<link:footnote> 要素	id	2.1
		xlink:label	2.0 および 2.1

項番	要素名	属性名	XBRL バージョン
10	<link:loc> 要素	xlink:label	2.0 および 2.1

上記に示す表のうち，id 属性値である ID は，読み込み時に一元管理されるため，重複する場合はエラーになります。既存のインスタンス文書を参照する場合などには，ID が一意になるように修正してください。

また，脚注リンクの <link:footnote> 要素および <link:loc> 要素に記述する xlink:label 属性値は，インスタンス文書で一意である必要はありません。しかし，uCosminexus Business Reporting Processor では，XLink による関係づけを管理するため，ID として一元管理しています。既存のインスタンス文書の場合，脚注リンクの再作成後に xlink:label 属性値が ID として一元管理されます。脚注リンクの再作成の詳細については，「付録 B　脚注リンクの再作成」を参照してください。

ユーザーが関係づける要素，使用する API，および ID が採番されるタイミングについて説明します。

アイテムとコンテキストとの関係づけ

アイテムとコンテキストとの関係づけには，コンテキストオブジェクトを使用します。XBRL2.0 の場合は XBRLNumericContext オブジェクトまたは XBRLNonNumericContext オブジェクトを，XBRL2.1 の場合は XBRLContext2 オブジェクトを使用します。XBRLInstanceItem オブジェクトに対して，コンテキストオブジェクトを設定することで，アイテムとコンテキストとの従属関係を定義できます。

新しく作成するコンテキストオブジェクトは，オブジェクト作成時に ID が自動採番され，管理されます。XBRL2.1 の場合，ユニットオブジェクトも，オブジェクト作成時に ID が自動採番され，管理されます。

脚注リンクとアイテム（またはタプル）の関係づけ

ユーザーアプリケーションは，<link:footnote> 要素の内容だけに着目して XBRLFootnote オブジェクトを作成します。作成した XBRLFootnote オブジェクトを，関係づけたい XBRLInstanceItem オブジェクト（または XBRLInstanceTuple オブジェクト）に登録することで，脚注リンクとアイテム（またはタプル）を関係づけることができます。

XBRL DOM では，<link:footnoteArc> 要素および <link:loc> 要素を作成するインターフェースはありません。これらの要素は，脚注リンク（<link:footnote> 要素）の作成時に自動的に作成されます。

このとき，自動的に作成する <link:loc> 要素および <link:footnote> 要素の xlink:label 属性値となる，ID が自動採番され，管理されます。また，関係づけるアイテム（またはタプル）に id 属性値が設定されていない場合には，ID が自動採番され，管理されます。

(2) 複数インスタンス文書間での ID 管理（XBRL2.1 の場合）

XBRL2.1 の場合は，一つの XML 文書内に複数のインスタンス文書を含めることができます。そのため，複数のインスタンス文書にわたって ID を一意にする必要があります。

uCosminexus Business Reporting Processor は，インスタンス文書ごとに ID を管理します。また，複数のインスタンス文書にわたって ID を一元管理する手段として，XBRLIdManager クラスを提供します。XBRLIdManager オブジェクトを使用することで，複数のインスタンス文書間で ID を一元管理できます。

3.3.2 ID の採番方法

ID の採番方法には，次があります。

- 自動的に採番された ID を使用する方法
- ユーザーが ID を任意に指定する方法

ここでは，それぞれの場合の ID 採番方法について説明します。

(1) 自動的に採番された ID を使用する方法

自動採番された ID を使用する場合の，採番対象の ID，ID のフォーマットおよび ID の採番規則について説明します。この方法は，XBRL2.0 および 2.1 で使用できます。

ID は，文書内で一意になるように採番されます。しかし，セグメント要素の子孫要素などの ID 自動採番がされない要素で，XML スキーマの xsd:ID 型属性を使用していた場合は，自動採番した値と同一になるおそれがあります。そのため，インスタンス文書作成後は，一度検証することを推奨します。

(a) 採番対象の ID

自動採番される ID は，XBRL のバージョンによって異なります。自動採番される ID と XBRL のバージョンを次の表に示します。

表 3-2 自動採番される ID と XBRL のバージョン

項番	要素名	属性名	エレメント識別子（EID）	XBRL のバージョン
1	アイテム型要素	id	ITM	2.0 および 2.1
2	タプル型要素	id	TPL	2.0 および 2.1
3	<xbrli:numericContext>	id	NUC	2.0
4	<xbrli:nonNumericContext>	id	NNC	2.0
5	<xbrli:context>	id	CNT	2.1
6	<xbrli:unit>	id	UNT	2.1

項番	要素名	属性名	エレメント識別子（EID）	XBRL のバージョン
7	<link:footnote>	xlink:label	FNT	2.0 および 2.1
8	<link:loc>	xlink:label	LOC	2.0 および 2.1

なお，XBRL2.1 では，<xbrli:xbrl> 要素，<link:footnoteLink> 要素，および <link:footnote> 要素に id 属性を指定できますが，uCosminexus Business Reporting Processor は，これら要素の id 属性値を自動作成しません。

(b) ID のフォーマット

自動採番する ID のフォーマットを次の図に示します。

図 3-1 ID のフォーマット

ID のフォーマットの出力内容を，次の表に示します。

表 3-3 ID のフォーマットの出力内容

項番	種別	内容
1	EID	エレメント識別子です。エレメントを識別するための値として，次のどれかが出力されます。 • NUC：<numericContext> 要素 • NNC：<nonNumericContext> 要素 • FNT：<link:footnote> 要素 • LOC：<link:loc> 要素 • ITM：<item> 型要素 • TPL：<tuple> 型要素 • XBL：<xbrli:xbrl> 要素 • CNT：<xbrli:context> 要素 • UNT：<xbrli:unit> 要素 • FTL：<link:footnoteLink> 要素
2	通番	ID の通番（5 桁）が出力されます。 この通番は，採番したすべての ID の通し番号です。エレメント識別子ごとではありません。
3	日付	日時が「yyyymmdd」形式で出力されます。
4	時刻	時刻が「hhmmss」形式で出力されます。
5	SID	サフィックスです。常に CBRP が出力されます。

(c) ID の採番規則

自動的に採番された ID を使用する場合の，ID の採番規則について説明します。

- 通番は 00001 から始まり，10 進数で 99999 までカウントアップします。カウンターは一つです。エレメント識別子ごとの通番ではありません。
- 日付および時刻は，XBRLInstanceDocument オブジェクト生成時の日付および時刻です。通番が最大値 99999 に達した場合は，現在日時を再度取得します。このとき，通番は 00001 に戻されます。
- 日付および時刻を取得し直す場合で，現在日時が取得前と同一日時となった場合，100 ミリ秒待ち，再度取得します。再度取得しても同一日時の場合には，さらに 100 ミリ秒待ち取得し直します。異なる日時が取得できるまで，同様に繰り返します。
- 採番した ID の値がインスタンス文書内にある場合，再度採番します。

(2) ユーザーが ID を任意に指定する方法

ユーザーが ID を任意に指定する方法には，次があります。

- 要素に対応するオブジェクトの setId() メソッドで指定する方法
- ID 採番クラスを作成して，ID が必要になった場合に通知を受け取る方法
  ID 採番クラスを使用する方法は，XBRL2.1 の場合だけ使用できます。

ここでは，ID 採番クラスを作成して，ID が必要になった場合に通知を受け取る方法で必要となる，インターフェースおよびクラスについて説明します。

- XBRLIdCreator
  XBRL 文書の ID を生成するインターフェースです。ユーザーは，このインターフェースを実装して ID 採番クラスを作成します。
  作成した ID 採番クラスのオブジェクトを，ID を一元管理する XBRLIdManager オブジェクトに設定することで，ユーザーアプリケーションで ID を生成できます。
- XBRLIdManager
  ID を一元管理するクラスです。
  作成した ID 採番クラスのオブジェクトを XBRLIdManager オブジェクトに登録することで，ID が必要になった場合に，登録した ID 採番クラスのオブジェクトがコールバックされ，ID を任意に指定できるようになります。
  XBRLIdManager オブジェクトは，XBRLInstanceDocument オブジェクトから取得する方法と，新しくインスタンス化する方法があります。新しくインスタンス化した場合には，この XBRLIdManager オブジェクトで管理する XBRLInstanceDocument オブジェクトを，XBRLIdManager オブジェクトに登録します。

サンプルコードについては，「4.7.7　ID の採番をするインスタンス文書の作成」を参照してください。また，使用する API の詳細については，「6.　API リファレンス」を参照してください。

3.4 名前空間の正規化

XML 仕様では，同じ名前空間に対して複数の名前空間接頭辞を使用したり，各要素に対して局所的な名前空間を宣言したりできます。

複数の名前空間接頭辞が設定されていた場合で，アイテム要素を既存インスタンス文書に追加する場合，どの名前空間接頭辞を設定する必要があるのかをユーザーアプリケーションが判断できません。

このため，uCosminexus Business Reporting Processor は，既存インスタンス文書に複数の名前空間宣言や名前空間接頭辞が設定されていた場合，名前空間を正規化します。名前空間の正規化によって，インスタンス文書内の名前空間が一意になり，ユーザーアプリケーションからインスタンス文書の操作がしやすくなります。

名前空間の正規化では，インスタンス文書内で宣言されている名前空間を取得し，重複している場合は一意の名前空間接頭辞を付け直します。そのため，名前空間の正規化によって，インスタンス文書内の名前空間宣言や名前空間接頭辞が削除または変更されることがあります。この削除や変更によって，インスタンス文書の意味や内容が変更されることはありません。また，一部の要素に指定されている名前空間は正規化されません。

名前空間の正規化は，インスタンス文書の編集で次の操作をした場合に行われます。

- 要素の追加または削除
- xsi:schemaLocation 属性の追加または削除
- 名前空間の追加または削除

すでにある要素に属性を追加したり，属性値を変更したりする場合，名前空間は正規化されません。ただし，アイテム要素が xsd:QName 型または xsd:NOTATION 型の場合には，正規化されます。

ここでは，正規化の準備である名前空間一覧および接頭辞の自動生成，名前空間の正規化規則と例，および正規化されない要素について説明します。

3.4.1 名前空間一覧および接頭辞の自動生成

uCosminexus Business Reporting Processor は，既存インスタンス文書で複数の名前空間宣言や接頭辞が定義されていた場合，名前空間を正規化します。

ここでは，正規化の準備である名前空間一覧の作成と，名前空間接頭辞の自動生成について説明します。

(1) 名前空間一覧の作成

uCosminexus Business Reporting Processor は，インスタンス文書の参照時に，記述されている名前空間を取得し，名前空間に対して接頭辞が一つになるように名前空間一覧を作成します。名前空間一覧の作成は，インスタンス文書の参照時に自動的に行われるため，ユーザーアプリケーションが意識する必要はありません。

名前空間に対して，接頭辞を一つにする規則の詳細については，「3.4.2　名前空間の正規化の例」を参照してください。

XBRL2.1 の場合，XML 文書内に XBRL のインスタンス文書を含めることができます。この構成の場合，インスタンス文書のルート要素である <xbrli:xbrl> 要素の祖先要素まで，名前空間を取得します。この結果，インスタンス文書内で定義されている名前空間がインスタンス文書外にも定義されている場合に，名前空間接頭辞の重複を防げます。なお，インスタンス文書外に定義されている名前空間を変更，削除することはありません。

また，XBRL2.1 では，インスタンス文書にデフォルトの名前空間宣言が定義されていなかった場合，デフォルトの名前空間を解除する宣言（xmlns=""）が，名前空間一覧に自動的に追加されます。この結果，インスタンス文書を XML 文書内で移動した場合に，インスタンス文書の名前空間が変更されるのを防げます。

（2）名前空間接頭辞の自動生成

名前空間一覧を作成する過程では，名前空間は異なりますが，名前空間接頭辞は同じケースがあることが考えられます。この場合，あとに定義されている名前空間に，異なる接頭辞を付け直す必要があります。そのため，uCosminexus Business Reporting Processor は次の規則に従い，名前空間接頭辞を生成します。生成した接頭辞は，名前空間の正規化で使用されます。

- 自動生成する接頭辞は，"pr0" から始まり，"pr1"，"pr2"，・・・"prn" となります。なお n は 0 以上の整数値で，Integer.MAX_VALUE（2,147,483,647）以下の値となります。
- 生成した接頭辞がすでにインスタンス文書で定義されていた場合は，一意となるように，接頭辞を再生成します。

3.4.2　名前空間の正規化の例

名前空間の正規化では，次に示す処理が実行されます。

1. インスタンス文書の読み込み時に作成した名前空間一覧を，ルート要素に追加します。
2. インスタンス文書の読み込み時に作成した名前空間一覧を基に，各要素の名前空間宣言を削除します。
3. 削除した名前空間宣言の代わりとなる名前空間に対応する名前空間接頭辞を，必要に応じて，要素または属性に設定・変更します。

ここでは，名前空間に対して接頭辞を一つにするときの正規化規則と，その例について説明します。

同じ名前空間に対して複数の名前空間宣言が記述されていた場合，最初に出現した名前空間宣言の接頭辞を優先し，名前空間に対する接頭辞を一つにします。このとき，名前空間の重複が排除され，名前空間接頭辞が自動生成されます。これらの例については，

「3.4.2(1)　名前空間宣言の重複排除」および「3.4.2(2)　名前空間接頭辞の自動作成」を参照してください。

ただし，デフォルト名前空間に関しては，次の点が異なります。これらの例外は，グローバル属性や QName 型要素，NOTATION 型要素のためです。

- 最初に出現した名前空間宣言がデフォルト名前空間（接頭辞なし）の場合で，別の名前空間宣言がされていた場合に限り，2 番目以降に出現した名前空間宣言の接頭辞が優先されます。
  この例については，「3.4.2(3)　デフォルト名前空間の例外的処置－グローバル属性への配慮（2 番目以降の宣言を優先）」を参照してください。
- 既存インスタンス文書の走査の過程で，デフォルト名前空間に対して，あとからデフォルト名前空間を取り消す宣言が出現した場合，デフォルト名前空間に割り当てた名前空間に対して，ほかの名前空間と重複しない接頭辞を自動生成し保持します。
  この例については，「3.4.2(4)　デフォルト名前空間の例外的処置－ QName 型などへの配慮（デフォルト名前空間の取り消し）」を参照してください。
- ルート要素にデフォルト名前空間宣言がなく，ルート要素以外でデフォルト名前空間宣言が出現した場合，ルート要素以外でデフォルト名前空間として定義された名前空間に対して，接頭辞を自動生成し保持します。
  この例については，「3.4.2(5)　デフォルト名前空間の例外的処置－ QName 型などへの配慮（名前空間接頭辞の自動生成）」を参照してください。

(1) 名前空間宣言の重複排除

正規化前：

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
      :
  <xbrli:numericContext id='Current_AsOf' precision='18' cwa='true'>
      :
  </xbrli:numericContext>
      :
  <xbrli:group xmlns:p0="http://www.xbrl.org/2001/instance>
    <p0:numericContext id='Prior_Shares' precision='18' cwa='true'>
      :
    </p0:numericContext>
  </xbrli:group>
</xbrli:group>
```

正規化後：

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
      :
  <xbrli:numericContext id='Current_AsOf' precision='18' cwa='true'>
      :
  </xbrli:numericContext>
      :
  <xbrli:group>
    <xbrli:numericContext id='Prior_Shares' precision='18' cwa='true'>
      :
    </xbrli:numericContext>
  </xbrli:group>
</xbrli:group>
```

解説：

名前空間「http://www.xbrl.org/2001/instance」に対して，正規化前では，接頭辞「xbrli」「p0」が混在しています。正規化後では，最初に出現した，接頭辞「xbrli」に統一されています。また，ルート要素の子要素である <group> 要素にあった「xmlns:p0」の名前空間宣言は不要なため削除されます。

（2）名前空間接頭辞の自動作成

正規化前：

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
  :
  <xbrli:group xmlns:xbrli="http://www.sampleCompany.com/fr/xbrli">
    <xbrli:PropertyPlantEquipment
            numericContext='Current_AsOf'>540000</xbrli:PropertyPlantEquipment>
  </xbrli:group>
  :
</xbrli:group>
```

正規化後：

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
             xmlns:pr0="http://www.sampleCompany.com/fr/xbrli"
  :
  <xbrli:group>
    <pr0:PropertyPlantEquipment
          numericContext='Current_AsOf'>540000</pr0:PropertyPlantEquipment>
  </xbrli:group>
  :
</xbrli:group>
```

解説：

正規化前では，名前空間「http://www.xbrl.org/2001/instance」と「http://www.sampleCompany.com/fr/xbrli」に対して，一方は局所的ですが，接頭辞は共に「xbrli」です。正規化後では，最初に出現した名前空間「http://www.xbrl.org/2001/instance」は「xbrli」のままとなり，あとから出現した名前空間「http://www.sampleCompany.com/fr/xbrli」は，局所的な宣言が削除され，かつ，宣言をルート要素に移動し自動生成した接頭辞「pr0」が設定されています。

また，名前空間「http://www.sampleCompany.com/fr/xbrli」に属する要素（<PropertyPlantEquipment>）の接頭辞も「pr0」に変更されています。

（3）デフォルト名前空間の例外的処置－グローバル属性への配慮（2 番目以降の宣言を優先）

正規化前：

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
             xmlns:link="http://www.xbrl.org/2001/XLink/xbrllinkbase"
             xmlns="http://www.w3.org/1999/xlink"
     :
  <xbrli:group xmlns="http://www.xbrl.org/2001/XLink/xbrllinkbase"
               xmlns:xlink="http://www.w3.org/1999/xlink">
    <footnoteLink xlink:type="extended">
      <footnote xlink:type="resource" ...>
        :
    <link:footnoteLink>
  <xbrli:group>
</xbrli:group>
```

正規化後：

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
             xmlns:link="http://www.xbrl.org/2001/XLink/xbrllinkbase"
             xmlns:xlink="http://www.w3.org/1999/xlink"
     :
  <xbrli:group>
    <link:footnoteLink xlink:type="extended">
      <link:footnote xlink:type="resource" ...>
        :
    </link:footnoteLink>
  <xbrli:group>
</xbrli:group>
```

解説：

正規化前は，名前空間「http://www.w3.org/1999/xlink」に対し，ルート要素ではデフォルト名前空間（接頭辞なし）として宣言されています。しかし，ルート要素の子要素である <group> 要素で局所的に接頭辞「xlink」と宣言し直しています。このため，<group> 要素の子孫ノードとなる <link:footnoteLink> などでは，グローバル属性である「xlink:type」が利用できます。

例外的処置をしない場合，先に出現した名前空間宣言が優先されるため，次のようになります。グローバル属性であった「xlink:type」がローカル属性「type」となり，不正なインスタンス文書となってしまいます。

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
             xmlns:link="http://www.xbrl.org/2001/XLink/xbrllinkbase"
             xmlns="http://www.w3.org/1999/xlink"
     :
  <xbrli:group>
    <link:footnoteLink type="extended">
      <link:footnote type="resource" ...>
        :
    </link:footnoteLink>
  <xbrli:group>
</xbrli:group>
```

このため，正規化後では，名前空間「http://www.w3.org/1999/xlink」に対して，最初に出現した宣言はデフォルト名前空間（接頭辞なし）であり，あとから出現した名前空間宣言の接頭辞が「xlink」になります。この場合には，例外的処置が適用され，あとから出現した接頭辞「xlink」を名前空間「http://www.w3.org/1999/xlink」の接頭辞とします。この結果，インスタンス文書の意味は変更されません。

（4）デフォルト名前空間の例外的処置－ QName 型などへの配慮（デフォルト名前空間の取り消し）

正規化前：

```
<group xmlns="http://www.xbrl.org/2001/instance"
       xmlns:p0="http://www.sampleCompany.com/fr/xbrli"
  :
  <p0:tradePayable.notes nonNumericContext='Current_AsOf'
        xmlns="">note</p0:tradePayable.notes>
  :
</group>
```

注　<p0:tradePayable.notes> は，xsd:QName 型とします。

正規化後：

```
<pr0:group xmlns:pr0="http://www.xbrl.org/2001/instance"
           xmlns:p0="http://www.sampleCompany.com/fr/xbrli"
           xmlns=""
  :
  <p0:tradePayable.notes nonNumericContext='Current_AsOf'
        >note</p0:tradePayable.notes>
  :
</pr0:group>
```

解説：

正規化前は，名前空間「http://www.xbrl,rog/2001/instance」に対し，ルート要素ではデフォルト名前空間（接頭辞なし）として宣言されています。しかし，ルート要素の子要素である <tradePayable.notes> 要素で局所的にデフォルト名前空間を取り消す「xmlns=""」と宣言し直しています。このため，<tradePayable.note> 要素の値は，{null, note} であり，名前空間に属さない「note」を意味します。

例外的処置を行わない場合，デフォルト名前空間には，先に「http://www.xbrl.org/2001/instance」が割り当てられているため，「xmlns=""」には自動生成された接頭辞が割り当てられ，次のようになります。

```
<group xmlns="http://www.xbrl.org/2001/instance"
       xmlns:p0="http://www.sampleCompany.com/fr/xbrli"
       xmlns:pr0=""
  :
  <p0:tradePayable.notes nonNumericContext='Current_AsOf'
        >note</p0:tradePayable>
  :
</group>
```

この場合，デフォルト名前空間を局所的に打ち消していたため {null:note} と解釈されていた <tradePayable.note> の値が，{http://www.xbrl.org/2001/instance, note} と解釈されることになり，不正なインスタンス文書となります。

このため，デフォルト名前空間を取り消す宣言「xmlns=""」が出現した場合，デフォルト名前空間であった「http://www.xbrl.org/2001/instance」に対して，自動生成した接頭辞を割り当てる例外処置を適用します。この結果，インスタンス文書の意味は変更されません。

(5) デフォルト名前空間の例外的処置－ QName 型などへの配慮（名前空間接頭辞の自動生成）

正規化前：

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
             xmlns:p0="http://www.sampleCompany.com/fr/xbrli"
   :
  <p0:tradeReceivables.notes xmlns="http://sample/qname"
        nonNumericContext='Current_AsOf'>note1</p0:tradeReceivables.notes>
  <p0:tradePayable.notes
        nonNumericContext='Current_AsOf'>note2</p0:tradePayable.notes>
   :
</xbrli:group>
```

注　<p0:tradeReceivables.notes> および <p0:tradePayable.notes> は，共に xsd:QName 型とします。

正規化後：

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
             xmlns:p0="http://www.sampleCompany.com/fr/xbrli"
             xmlns:pr0="http://sample/qname"
   :
  <p0:tradeReceivables.notes
        nonNumericContext='Current_AsOf'>pr0:note1</p0:tradeReceivables.notes>
  <p0:tradePayable.notes
        nonNumericContext='Current_AsOf'>note2</p0:tradePayable.notes>
   :
</xbrli:group>
```

解説：

正規化前は，<tradeReceivables.notes> 要素に，局所的な名前空間「http://sample.qname」がデフォルト名前空間（接頭辞なし）として宣言されています。このため，<tradeReceivables.note> 要素の値は，{http://sample.qname, note1} であり，名前空間「http://sample.qname」に属する「note1」を意味します。

例外的処置を行わない場合，デフォルト名前空間は宣言されていないため，ルート要素である <group> 要素に「http://sample.qname」が移動され，次のようになります。

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
             xmlns:p0="http://www.sampleCompany.com/fr/xbrli"
             xmlns="http://sample/qname"
   :
  <p0:tradeReceivables.notes
        nonNumericContext='Current_AsOf'>note1</p0:tradeReceivables.notes>
  <p0:tradePayable.notes
        nonNumericContext='Current_AsOf'>note2</p0:tradePayable.notes>
   :
</xbrli:group>
```

<tradeReceivables.notes> 要素の値は，{http://sample.qname, note1} のままであり，例外的処理をしなくても意味は変わりません。しかし，<tradePayable.notes> 要素の値は，名前空間に属さない {null, note2} でしたが，デフォルト名前空間宣言をルート要素に単純に移動したため {http://sample.qname, note2} となってしまいま

す。
このため，正規化後ではルート要素にデフォルト名前空間宣言がなく，ルート要素以外にデフォルト名前空間宣言があった場合の例外的処置を適用します。名前空間「http://sample.qname」に自動生成した接頭辞「pr0」を割り当て，QName 型の値も接頭辞「pr0」を付けています。

3.4.3 正規化されない要素

ここでは，名前空間が正規化されない <scenario> 要素および 要素について説明します。

<scenario> 要素および 要素には任意の XML 要素を記述できるため，<scenario> 要素および 要素の子孫ノード（要素，属性または値）について，名前空間接頭辞の変更，および名前空間宣言の削除をしないで，記述された内容のままとします。

このため，次のすべての条件にあてはまる既存インスタンス文書を編集し，保存した場合，保存した文書が不正になるおそれがあります。

- <scenario>/ 要素およびその子孫ノード（要素や属性，値）がある場合。
- <scenario>/ 要素およびその子孫ノードで使用する名前空間が，<scenario>/ 要素を含む上位ノードで宣言されている場合。
- <scenario>/ 要素およびその子孫ノードで使用する名前空間の接頭辞が，ほかの名前空間宣言の接頭辞と重複しているか，またはすでに別の接頭辞で宣言されている場合。

この場合，あらかじめ名前空間に対する接頭辞を一つに統一しておくことで，文書が不正になることを防げます。

インスタンス文書内で，同じ名前空間に対する（大域または局所的の区別なく）名前空間宣言が複数ない場合や，<scenario> 要素および 要素下位の子孫ノード内で名前空間が閉じている場合は，名前空間の正規化が行われても文書の意味は変わりません。

ただし，XBRL Dimensions1.0 の場合は，次に示す要素，および値だけが名前空間の正規化の対象になります。

- <xbrli:scenario> 要素および 要素の子要素である <xbrldt:typedMember> 要素
- <xbrli:scenario> 要素および 要素の子要素である <xbrldt:explicitMember> 要素
- <xbrldt:explicitMember> 要素の値

（1） 正規化されない例

正規化前：

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
  :
  <xbrli:numericContext id="C03" precision="18" cwa="false">
    <xbrli:entity>
      <xbrli:identifier scheme="www.hitachi.com">HITACHI</xbrli:identifier>
        <xbrli:segment xmlns:my="http://www.sampleCompany.com/segment">
          <my:state xmlns:addr="http://www.sampleCompany.com/address">
            <addr:address>Yokohama</addr:address>
            <addr:zipcode>244-0003</addr:zipcode>
          </my:state>
        </xbrli:segment>
  :
```

正規化後：

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
             xmlns:my="http://www.sampleCompany.com/segment"
  :
  <xbrli:numericContext id="C03" precision="18" cwa="false">
    <xbrli:entity>
      <xbrli:identifier scheme="www.hitachi.com">HITACHI</xbrli:identifier>
        <xbrli:segment>
          <my:state xmlns:addr="http://www.sampleCompany.com/address">
            <addr:address>Yokohama</addr:address>
            <addr:zipcode>244-0003</addr:zipcode>
          </my:state>
        </xbrli:segment>
  :
```

解説：

正規化前は， 要素に，局所的な名前空間「http://www.sampleCompany.com/segment」が宣言されていましたが，正規化後では，この宣言が文書のルート要素である <group> 要素に移動されています。

ただし， 要素下位の子要素である <my:state> 要素で宣言されている「http://www.sampleCompany.com/address」は， 要素下位の子孫ノードが対象外であるため，正規化されません。宣言位置は正規化前と同じです。

(2) 正規化されなかったため，不正な文書となる例

正規化前：

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
             xmlns:my="http://www.sampleCompany.com/segment"
             xmlns:xlink="http://www.w3.org/1999/xlink"
  :
  <my:tradePayable.notes nonNumericContext=NC03'>note</my:tradePayable.notes>
  :
  <xbrli:numericContext id="C03" precision="18" cwa="false">
    <xbrli:entity>
      <xbrli:identifier scheme="www.hitachi.com">HITACHI</xbrli:identifier>
        <xbrli:segment xmlns:xlk="http://www.w3.org/1999/xlink">
          <my:state xmlns:addr="http://www.sampleCompany.com/address">
            <addr:address>Yokohama</addr:address>
            <addr:zipcode>244-0003</addr:zipcode>
          </my:state>
         <my:footnote xlk:type="simple"xlk:href="http://www.sampleCompany.com/
note"/>
        </xbrli:segment>
  :
```

正規化後：

```
<xbrli:group xmlns:xbrli="http://www.xbrl.org/2001/instance"
             xmlns:my="http://www.sampleCompany.com/segment"
             xmlns:xlink="http://www.w3.org/1999/xlink"
  :
  <my:tradePayable.notes nonNumericContext=NC03'>note</my:tradePayable.notes>
  :
  <xbrli:numericContext id="C03" precision="18" cwa="false">
    <xbrli:entity>
      <xbrli:identifier scheme="www.hitachi.com">HITACHI</xbrli:identifier>
        <xbrli:segment>
          <my:state xmlns:addr="http://www.sampleCompany.com/address">
            <addr:address>Yokohama</addr:address>
            <addr:zipcode>244-0003</addr:zipcode>
          </my:state>
         <my:footnote xlk:type="simple"xlk:href="http://www.sampleCompany.com/
note"/>
        </xbrli:segment>
  :
```

解説：

正規化前は， 要素に，局所的な名前空間「http://www.w3.org/1999/xlink」が宣言されていましたが，正規化によって文書のルート要素である <group> 要素に宣言されている名前空間宣言が優先され， 要素の局所的な名前空間宣言に対応するノードは削除されます。

このため正規化によって，大域属性である「xlk:type」「xlk:href」属性の接頭辞「xlk」に対する名前空間宣言のない不正な文書となります。

この例のように，同じ名前空間に対して複数の接頭辞が混在する，既存インスタンス文書を編集する場合，あらかじめ，名前空間に対する接頭辞を一つに統一しておくことで，不具合を防げます。

4 アプリケーションの開発

この章では，uCosminexus Business Reporting Processor が提供する API を利用して，XBRL 文書を操作するアプリケーションを開発する方法について説明します。開発するために必要な環境設定や，トレースログについても説明します。
また，提供されているサンプルソースを使用して，XBRL 文書の作成や検証をするアプリケーションを実行する方法についても説明します。

4.1 **この章の読み方**

4.2 **アプリケーションの開発環境の設定**

4.3 **タクソノミー文書の参照と検証（XBRL2.0 の場合）**

4.4 **インスタンス文書の新規作成と保存（XBRL2.0 の場合）**

4.5 **既存のインスタンス文書の編集（XBRL2.0 の場合）**

4.6 **タクソノミー文書の参照と検証（XBRL2.1 の場合）**

4.7 **インスタンス文書の新規作成と保存（XBRL2.1 の場合）**

4.8 **既存のインスタンス文書の編集（XBRL2.1 の場合）**

4.9 **タクソノミー文書の拡張（XBRL2.1 の場合）**

4.10 **タクソノミー文書の参照と検証（XBRL Dimensions1.0 の場合）**

4.11 **インスタンス文書の新規作成と保存（XBRL Dimensions1.0 の場合）**

4.12 **既存のインスタンス文書の参照と検証（XBRL Dimensions1.0 の場合）**

4.13 **エラーハンドラ**

4.14 エンティティリゾルバ

4.15 サンプルアプリケーションの実行

4.16 アプリケーションの実行環境の設定

4.17 トレースログ

4.1　この章の読み方

この章は，XBRL2.0，XBRL2.1，および XBRL Dimensions1.0 をご利用の場合で，マニュアルを参照する個所が異なります。ご利用のバージョンによって必要な個所をお読みください。

XBRL2.0，XBRL2.1 の場合に参照する内容およびその参照先を次の表に示します。

表 4-1　アプリケーションの開発時に参照する内容（XBRL2.0，XBRL2.1 の場合）

XBRL のバージョン	内容	参照先
2.0	アプリケーションの開発環境の設定	4.2
	タクソノミー文書の参照と検証	4.3
	インスタンス文書の新規作成と保存	4.4
	既存のインスタンス文書の編集	4.5
	エラーハンドラ	4.13
	エンティティリゾルバ	4.14
	サンプルアプリケーションの実行	4.15
	アプリケーションの実行環境の設定	4.16
	トレースログ	4.17
2.1	アプリケーションの開発環境の設定	4.2
	タクソノミー文書の参照と検証	4.6
	インスタンス文書の新規作成と保存	4.7
	既存のインスタンス文書の編集	4.8
	タクソノミー文書の拡張	4.9
	エラーハンドラ	4.13
	エンティティリゾルバ	4.14
	サンプルアプリケーションの実行	4.15
	アプリケーションの実行環境の設定	4.16
	トレースログ	4.17

XBRL Dimensions1.0 をご利用の場合は，XBRL2.1 の場合の内容に加えて，次の表に示す内容もお読みください。

表 4-2　アプリケーションの開発時に参照する内容（XBRL Dimensions1.0 の場合）

XBRL Dimensions のバージョン	内容	参照先
1.0	タクソノミー文書の参照と検証	4.10
	インスタンス文書の新規作成と保存	4.11

XBRL Dimensions のバージョン	内容	参照先
	既存のインスタンス文書の参照と検証	4.12

4.2　アプリケーションの開発環境の設定

ここでは，アプリケーションを開発するための環境設定について説明します。

開発環境の設定では，次が必要です。

- クラスパスの設定
  uCosminexus Business Reporting Processor などが提供するクラスライブラリを使用するための設定です。
- スキーマ定義の準備
  XBRL 文書の検証などに使用するスキーマ定義を準備します。公開されているスキーマ定義をダウンロードするか，公開先にアクセスできる状態にしておく必要があります。必要なスキーマ定義については，「4.15.3　必要なスキーマ定義」を参照してください。

4.2.1　クラスパスの設定

開発環境の設定では，クラスパスを設定します。JavaVM や各種 SDK ツールがクラスライブラリを参照できるように，uCosminexus Business Reporting Processor および前提プログラムが提供する JAR ファイルをクラスパスに追加します。

クラスパスを追加するには，次の方法があります。

- JavaVM 起動時に -classpath オプションを使用する
- CLASSPATH 環境変数を設定する

クラスパスに追加する JAR ファイルとデフォルトの格納場所を次に示します。

● cbrp.jar

uCosminexus Business Reporting Processor が提供するクラスライブラリです。

【Windows の場合】

C:¥Program Files¥HITACHI¥cbrp¥lib¥cbrp.jar

【UNIX の場合】

/opt/hitachi/cbrp/lib/cbrp.jar

● csmxlpr.jar

前提プログラムである uCosminexus XML Link Processor が提供するクラスライブラリです。uCosminexus XML Link Processor は，XLink プロセッサです。

【Windows の場合】

C:¥Program Files¥HITACHI¥xlpr¥lib¥csmxlpr.jar

【UNIX の場合】

/opt/hitachi/xlpr/lib/csmxlpr.jar

なお，P-9S5Z-A144 04-00 Linux 版の場合，csmxlpr.jar ファイルの格納先は設定

しません。

uCosminexus Business Reporting Processor のバージョンが 05-00 以降の場合，csmxlpr.jar ファイルのデフォルトの格納先は次の場所になります。

【Windows の場合】

C:¥Program Files¥HITACHI¥cbrp¥lib¥csmxlpr.jar

● csmjaxp.jar

前提プログラムである Processing Kit for XML，または uCosminexus Application Server（Cosminexus XML Processor）が提供する JAXP のクラスライブラリです。Processing Kit for XML および Cosminexus XML Processor は XML プロセッサです。

Processing Kit for XML のクラスライブラリの格納場所

【Windows の場合】

C:¥Program Files¥HITACHI¥Processing Kit for XML¥jaxp¥lib¥csmjaxp.jar

【UNIX の場合】

/opt/PKX/jaxp/lib/csmjaxp.jar

uCosminexus Application Server（Cosminexus XML Processor）のクラスライブラリの格納場所

【Windows の場合】

C:¥Program Files¥HITACHI¥Cosminexus¥jaxp¥lib¥csmjaxp.jar

【UNIX の場合】

/opt/Cosminexus/jaxp/lib/csmjaxp.jar

● hntrlib2j.jar

日立トレース共通ライブラリのクラスライブラリです。トレース出力で使用します。

【Windows の場合】

C:¥Program Files¥HITACHI¥HNTRLib2¥classes¥hntrlib2j.jar

【UNIX の場合】

/opt/hitachi/HNTRLib2/classes/hntrlib2j.jar

4.3　タクソノミー文書の参照と検証（XBRL2.0の場合）

タクソノミー文書の参照と検証をするアプリケーションの開発について説明します。

ここで示すサンプルは，次の流れでコーディングしています。

1. **タクソノミー文書の読み込み**
2. **タクソノミー本体とリンクベースの参照**

4.3.1　タクソノミー文書の読み込み

XBRL2.0 に準拠したタクソノミー文書を読み込み，XBRL DOM を生成するサンプルコードについて説明します。XBRL DOM の生成時にタクソノミー文書の妥当性を検証するには，setValidating() メソッドを実行します。

また，検証時にエラーや警告が発生した場合のレポートを受け取り，出力するエラーハンドラと，検証時にローカルに保存したスキーマ定義を使用するエンティティリゾルバも実装します。エラーハンドラおよびエンティティリゾルバの詳細については，「4.13 エラーハンドラ」および「4.14　エンティティリゾルバ」を参照してください。

（1）使用するクラスおよびインターフェース

タクソノミー文書の読み込みで使用するクラスおよびインターフェースを次の表に示します。

表 4-3　タクソノミー文書の読み込みで使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.XBRLDocumentManager	XBRL 文書の入出力や，タクソノミー文書のキャッシュを管理します。
2	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory	XBRLTaxonomyDocumentBuilder オブジェクトを生成するための API を定義します。
3	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder	タクソノミー文書から XBRLTaxonomyDocument オブジェクトを取得するための API を定義します。
4	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument	タクソノミー文書全体を表します。

（2）サンプルコード

タクソノミー文書の読み込みのコードは次のとおりです。このサンプルコードでは setValidating() メソッドがコメントアウトされていますが，妥当性検証はデフォルトの

設定で実施されるため，このサンプルコードを使用しても妥当性検証が実行されます。

```
/*
 * タクソノミー文書を読み込む
 */

// 初期化
XBRLDocumentManager.init();
/*
 * プロパティファイルを利用して初期化する場合は，ファイルの絶対パスを指定します
 * 例:
 * XBRLDocumentManager.init("C://Program Files/HITACHI/cbrp/properties/
cbrp.properties");
 */
// XBRLDocumentManagerの取得
XBRLDocumentManager dm = XBRLDocumentManager.getInstance();

// XBRLTaxonomyDocumentBuilderFactoryの作成
XBRLTaxonomyDocumentBuilderFactory tbf =
    dm.newTaxonomyDocumentBuilderFactory();
// タクソノミー文書のパース時に妥当性検証を行う設定
// tbf.setValidating(true);

// XBRLTaxonomyDocumentBuilderの作成
XBRLTaxonomyDocumentBuilder tb = tbf.newTaxonomyDocumentBuilder();

// エラーハンドラの設定
SampleErrorHandler eh = new SampleErrorHandler();
tb.setErrorHandler(eh);

// エンティティリゾルバの設定
SampleEntityResolver er = new SampleEntityResolver();
tb.setEntityResolver(er);

// タクソノミー文書のパース
String filename = "input/sample_taxonomy1.xsd";
File[] taxonomy = { new File(filename) };
XBRLTaxonomyDocument tdoc = tb.parse(taxonomy);

System.out.println("タクソノミー文書のパースに成功しました。");
```

4.3.2 タクソノミー本体とリンクベースの参照

XBRL2.0 に準拠したタクソノミー文書の読み込み時に生成した XBRL DOM から，要素の属性（アイテム名またはタプル名）を参照するサンプルコードについて説明します。要素がアイテムの場合は，その型を参照し，同時にタクソノミー本体に関連づけられたラベルをラベルリンク（リンクベース）から参照します。

タクソノミー本体からアイテム名と型，またはタプル名を，ラベルリンク文書からラベル名（言語）を取り出し，出力します。

(1) 使用するインターフェース

タクソノミー本体とリンクベースの参照で使用するインターフェースを次の表に示します。

表 4-4　タクソノミー本体とリンクベースの参照で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument	タクソノミー文書全体を表します。
2	com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement	タクソノミー文書に記述される要素の要素宣言を表します。
3	com.hitachi.xbrl.taxonomy.XBRLTaxonomyItem	アイテムの要素宣言を表します。
4	com.hitachi.xbrl.taxonomy.XBRLTaxonomyTuple	タプルの要素宣言を表します。
5	com.hitachi.xbrl.taxonomy.XBRLLabel	要素のラベルを表します。

(2) サンプルコード

タクソノミー本体とリンクベースを参照するコードは次のとおりです。

```
// タクソノミー文書のパース
String filename = "input/sample_taxonomy1.xsd";
File[] taxonomy = { new File(filename) };
XBRLTaxonomyDocument tdoc = tb.parse(taxonomy);


System.out.println("タクソノミー文書のパースに成功しました。");


/*
 * ここからタクソノミー文書の内容を参照
 */

System.out.println("タクソノミー文書の要素宣言と，そのラベルを出力します。");
```

```
// 要素宣言(アイテムまたはタプルの定義)の一覧を取得
List<XBRLTaxonomyElement> elist = tdoc.getElements();
for (XBRLTaxonomyElement element : elist) {
    switch (element.getElementType()) {
        case XBRLTaxonomyElement.ITEM_ELEMENT :
            // 要素宣言がアイテム
            XBRLTaxonomyItem item = (XBRLTaxonomyItem) element;
            // 名前の出力
            System.out.println("item:" + item.getName());
            // 型の出力
            if (item.getItemType()
                    == XBRLTaxonomyItem.MONETARY_ITEM) {
                // 型が通貨型(xbrli:monetaryItemType)の場合
                System.out.println(" ItemType:MONETARY_ITEM");
            } else if (
                item.getItemType()
                    == XBRLTaxonomyItem.STRING_ITEM) {
                // 型が文字列型(xbrli:stringItemType)の場合
                System.out.println(" ItemType:STRING_ITEM");
            }
            // 関連付けられているラベルの一覧を取得
            List lbllist = element.getLabels();
            // ラベルの出力
            for (int j = 0; j < lbllist.size(); j++) {
                XBRLLabel lbl = (XBRLLabel) lbllist.get(j);
                // 値と言語を出力
                System.out.println(
                    " Label:"
                        + lbl.getValue()
                        + "("
                        + lbl.getLang()
                        + ")");
            }
            break;


        case XBRLTaxonomyElement.TUPLE_ELEMENT :
            // 要素宣言がタプル
            XBRLTaxonomyTuple tuple = (XBRLTaxonomyTuple) element;
            System.out.println("Tuple:" + tuple.getName());
            break;


    }
}
```

(3) サンプルコードの実行結果

サンプルのタクソノミー文書を参照したときの実行結果画面を次に示します。

```
タクソノミー文書のパースに成功しました。
タクソノミー文書の要素宣言と，そのラベルを出力します。
item:assets
 ItemType:MONETARY_ITEM
 Label:資産合計(ja)
 Label:Assets(en)
item:currentAssets
 ItemType:MONETARY_ITEM
 Label:流動資産(ja)
 Label:Current assets(en)
item:investmentsAndAdvances
 ItemType:MONETARY_ITEM
 Label:投資及び貸付金(ja)
 Label:Investments and advances(en)
item:note
 ItemType:STRING_ITEM
```

4.4 インスタンス文書の新規作成と保存（XBRL2.0 の場合）

ここでは，uCosminexus Business Reporting Processor の API を利用して，XBRL2.0 に準拠したインスタンス文書を新規に作成して保存します。

ここで示すサンプルは，次の流れでコーディングしています。

1. **インスタンス文書の新規作成**
2. **非数値コンテキスト要素（nonNumericContext）の追加**
3. **数値コンテキスト要素（NumericContext）の追加**
4. **非数値コンテキストに対応したアイテム（Item）の追加**
5. **数値コンテキストに対応したアイテム（Item）の追加**
6. **インスタンス文書の保存**

なお，このサンプルではコーディングをわかりやすくするために，通常は自動生成される id 属性などの値を仮定しています。

4.4.1 インスタンス文書の新規作成

XBRL2.0 に準拠したインスタンス文書を新規に作成するサンプルコードについて説明します。

インスタンス文書の新規作成では，まずスキーマとなるタクソノミー文書の XBRL DOM を生成します。生成したタクソノミー文書の XBRL DOM を基に，インスタンス文書の XBRL DOM を生成します。また，名前空間宣言，および xsi:schemaLocation 属性を設定します。

（1） 使用するクラスおよびインターフェース

インスタンス文書の新規作成で使用するクラスおよびインターフェースを次の表に示します。

表 4-5 インスタンス文書の新規作成で使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.XBRLDocumentManager	XBRL 文書の入出力や，タクソノミー文書のキャッシュを管理します。
2	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory	XBRLTaxonomyDocumentBuilder オブジェクトを生成するための API を定義します。

項番	使用するクラスおよびインターフェース	説明
3	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder	タクソノミー文書から XBRLTaxonomyDocument オブジェクトを取得するための API を定義します。
4	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument	タクソノミー文書全体を表します。
5	com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory	XBRLInstanceDocumentBuilder オブジェクトを生成するための API を定義します。
6	com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder	インスタンス文書から XBRLInstanceDocument オブジェクトを取得するための API を定義します。
7	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。
8	com.hitachi.xbrl.util.XBRLNamespaceDeclaration	名前空間接頭辞とそれに対応する名前空間 URI の組を管理します。
9	com.hitachi.xbrl.util.XBRLSchemaLocation	xsi:schemaLocation 属性に指定する名前空間 URI とスキーマ文書の位置を示す URI の組を管理します。

(2) サンプルコード

インスタンス文書の新規作成のコードは次のとおりです。

```
/*
 * 新たなインスタンス文書作成の前準備
 */


// 初期化
XBRLDocumentManager.init();
XBRLDocumentManager dm = XBRLDocumentManager.getInstance();


// XBRLTaxonomyDocumentBuilderFactoryの作成
XBRLTaxonomyDocumentBuilderFactory tbf =
    dm.newTaxonomyDocumentBuilderFactory();


// XBRLTaxonomyDocumentBuilderの作成
XBRLTaxonomyDocumentBuilder tb = tbf.newTaxonomyDocumentBuilder();


// エラーハンドラの設定
SampleErrorHandler eh = new SampleErrorHandler();
tb.setErrorHandler(eh);


// エンティティリゾルバの設定
SampleEntityResolver er = new SampleEntityResolver();
tb.setEntityResolver(er);
```

```
// タクソノミー文書のパース
String filename = "input/sample_taxonomy1.xsd";
File[] taxonomy = { new File(filename) };
XBRLTaxonomyDocument tdoc = tb.parse(taxonomy);

System.out.println("タクソノミー文書のパースに成功しました。");


/*
 * ここから新たなインスタンス文書の作成
 */

// XBRLInstanceDocumentBuilderFactoryの作成
XBRLInstanceDocumentBuilderFactory ibf =
    dm.newInstanceDocumentBuilderFactory();


// XBRLInstanceDocumentBuilderの作成
XBRLInstanceDocumentBuilder ib = ibf.newInstanceDocumentBuilder();

// 新たなインスタンス文書の作成
XBRLInstanceDocument idoc = ib.newInstanceDocument(tdoc);


// 名前空間宣言 - hi - の作成
// (xmlns:hi="http://www.hitachi.com/2003/XBRL")
XBRLNamespaceDeclaration nd1 =
    new XBRLNamespaceDeclaration(
        "hi",
        "http://www.hitachi.com/2003/XBRL");


// 名前空間宣言 - iso4217 - の作成
// (xmlns:iso4217="http://www.xbrl.org/2003/iso4217")
XBRLNamespaceDeclaration iso =
    new XBRLNamespaceDeclaration(
        "iso4217",
        "http://www.xbrl.org/2003/iso4217");


// xsi:schemaLocation属性値の作成
XBRLSchemaLocation sl =
    new XBRLSchemaLocation(
        "http://www.hitachi.com/2003/XBRL",
        "sample_taxonomy1.xsd");


// インスタンス文書に名前空間宣言とxsi:schemaLocation属性値を追加
idoc.addNamespaceDeclaration(nd1);
idoc.addNamespaceDeclaration(iso);
idoc.addSchemaLocation(sl);
```

4.4.2　非数値コンテキスト要素の追加

XBRL2.0 に準拠したインスタンス文書に，非数値コンテキストを追加するサンプルコードについて説明します。

ここでは，「4.4.1　インスタンス文書の新規作成」で生成したインスタンス文書の XBRL DOM に，非数値コンテキストを追加します。非数値コンテキスト要素を追加する場合，非数値コンテキスト要素の子要素（entity，period など）を作成する必要があります。

ここで説明するサンプルコードでは，次の内容の非数値コンテキストを追加します。表中の「名称」欄のインデントは，コーディング結果のインデントを示しています。

表 4-6　非数値コンテキストの値

項番	名称	要素 / 属性	値
1	コンテキスト ID	属性	NNC01
2	時期	要素	2003-03-31
3	エンティティ（組織）	要素	―
4	識別子	要素	SAMP
5	スキーム	属性	指定しない
6	セグメント	要素	―
7	prefecture ※	要素	Tokyo

（凡例）―：値を持つことができません。
注※　"prefecture" の名前空間 URI は "http://www.hitachi.com/2003/XBRL" とします。

このサンプルコードでは，セグメント要素の子要素として <prefecture> 要素を追加し，値を Tokyo にしています。セグメント要素作成時に，org.w3c.dom.Element オブジェクトも同時に作成されます。org.w3c.dom.Element オブジェクトは org.w3c.dom.Node オブジェクトから派生されているため，getContainerNode() メソッドを使用すると，org.w3c.dom.Element オブジェクトを取得できます。

子要素の追加方法を次に示します。この方法を使用して，シナリオ要素などにも任意の子要素を追加できます。ここでは，セグメント要素に <prefecture> 要素を追加する例を使用して説明します。

1. getW3CDOMNode() **メソッドを使用して，インスタンス文書の** org.w3c.dom.Document **オブジェクトを取得します。**
2. org.w3c.dom.Document **オブジェクトの** createElement() **メソッドを使用して，** <prefecture> **要素を作成します。**
3. org.w3c.dom.Node **オブジェクトの** appendChild() **メソッドを使用して，親子関係を設定します。**

ここで使用する createElement() メソッド，および appendChild() メソッドは，org.w3c.dom の API です。

（1）使用するクラスおよびインターフェース

非数値コンテキスト要素の追加で使用するクラスおよびインターフェースを次の表に示します。

表 4-7 非数値コンテキスト要素の追加で使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.types.XBRLDateUnion	XBRL のデータ型 dateUnion を表します。
2	com.hitachi.xbrl.instance.XBRLPeriod	時期要素を表します。
3	com.hitachi.xbrl.instance.XBRLSegment	セグメント要素を表します。
4	com.hitachi.xbrl.instance.XBRLIdentifier	識別子要素を表します。
5	com.hitachi.xbrl.instance.XBRLEntity	エンティティ要素を表します。
6	com.hitachi.xbrl.instance.XBRLNonNumericContext	非数値コンテキストを表します。
7	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

非数値コンテキストを XBRL DOM に追加するコードは次のとおりです。

```
// period要素の作成
XBRLDateUnion date1 = new XBRLDateUnion("2003-03-31");
XBRLPeriod period1 =
    idoc.createPeriod(XBRLPeriod.INSTANT, date1, null, null, null);
// identifier要素の作成
XBRLIdentifier identifier1 = idoc.createIdentifier("", "SAMP");
// segment要素の作成
XBRLSegment segment1 = idoc.createSegment(null);


// segment要素に対応するorg.w3c.dom.Elementの取得
Node container1 = segment1.getContainerNode();
// インスタンス文書に対応するorg.w3c.dom.Documentの取得
Document doc = (Document) idoc.getW3CDOMNode();
// prefecture要素の作成
Element prefecture1 =
    doc.createElementNS(
        "http://www.hitachi.com/2003/XBRL",
        "prefecture");
prefecture1.appendChild(doc.createTextNode("Tokyo"));
// segment要素の子要素にprefecture要素を追加
container1.appendChild(prefecture1);


// entity要素の作成
XBRLEntity entity1 = idoc.createEntity(identifier1, segment1);


// 非数値コンテキストの作成
XBRLNonNumericContext nonNumericContext =
    idoc.createNonNumericContext(
        entity1,
        period1,
        null,   // unit要素なし
        null);  // scenario要素なし
// インスタンス文書に非数値コンテキストを追加
idoc.addContext(nonNumericContext);
```

(3) コーディング結果

コーディング結果として，生成されるインスタンス文書は次のとおりです。

```
<xbrli:nonNumericContext id="NNC01">
    <xbrli:entity>
        <xbrli:identifier scheme="">SAMP</xbrli:identifier>
        <xbrli:segment>
            <prefecture xmlns="http://www.hitachi.com/2003/XBRL">Tokyo</
prefecture>
        </xbrli:segment>
    </xbrli:entity>
    <xbrli:period>
        <xbrli:instant>2003-03-31</xbrli:instant>
    </xbrli:period>
</xbrli:nonNumericContext>
```

4.4.3　数値コンテキスト要素の追加

XBRL2.0 に準拠したインスタンス文書に，数値コンテキストを追加するサンプルコードについて説明します。

ここでは，「4.4.2　非数値コンテキスト要素の追加」で非数値コンテキストを追加した XBRL DOM に，さらに数値コンテキストを追加します。数値コンテキスト要素を追加する場合，非数値コンテキストで追加した要素に加えて，さらに単位要素，precision 属性，cwa 属性が必要になります。

ここで説明するサンプルコードでは，次の内容の数値コンテキストを追加します。表中の「名称」欄のインデントは，コーディング結果のインデントを示しています。

表 4-8　数値コンテキストの値

項番	名称	要素 / 属性	値
1	コンテキスト ID	属性	NC01
2	precision	属性	9
3	cwa	属性	false
4	時期	要素	2003-03-31
5	エンティティ（組織）	要素	―
6	識別子	要素	SAMP
7	スキーム	属性	指定しない
8	セグメント	要素	―
9	prefecture ※1	要素	Tokyo
10	単位	要素	―
11	計量	要素	iso4217:JPY ※2

（凡例）―：値を持つことができません。

注※1　"prefecture" の名前空間 URI は "http://www.hitachi.com/2003/XBRL" とします。
注※2　値の単位を「円」とすることを示します。

(1) 使用するクラスおよびインターフェース

数値コンテキスト要素の追加で使用するクラスおよびインターフェースを次の表に示します。

表 4-9　数値コンテキスト要素の追加で使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.types.XBRLQName	XML Schema のデータ型 QName を表します。
2	com.hitachi.xbrl.instance.XBRLMeasure	計量要素を表します。
3	com.hitachi.xbrl.instance.XBRLUnit	単位要素を表します。
4	com.hitachi.xbrl.instance.XBRLMeasureOrOperator	計量要素または演算子要素を表します。
5	com.hitachi.xbrl.types.XBRLDateUnion	XBRL のデータ型 dateUnion を表します。
6	com.hitachi.xbrl.instance.XBRLPeriod	時期要素を表します。
7	com.hitachi.xbrl.instance.XBRLIdentifier	識別子要素を表します。
8	com.hitachi.xbrl.instance.XBRLSegment	セグメント要素を表します。
9	com.hitachi.xbrl.instance.XBRLEntity	エンティティ要素を表します。
10	com.hitachi.xbrl.instance.XBRLNumericContext	数値コンテキストを表します。
11	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

数値コンテキストを XBRL DOM に追加するコードは次のとおりです。

```
// measure要素の作成
XBRLQName currency =
    new XBRLQName("http://www.xbrl.org/2003/iso4217", "JPY");
XBRLMeasure measure = idoc.createMeasure(currency);
// unit要素の作成
XBRLUnit unit = idoc.createUnit((XBRLMeasureOrOperator) measure);
// period要素の作成
XBRLDateUnion date2 = new XBRLDateUnion("2003-03-31");
XBRLPeriod period2 =
    idoc.createPeriod(XBRLPeriod.INSTANT, date2, null, null, null);
// identifier要素の作成
XBRLIdentifier identifier2 = idoc.createIdentifier("", "SAMP");
// segment要素の作成
XBRLSegment segment2 = idoc.createSegment(null);
// prefecture要素を作成
Node container2 = segment2.getContainerNode();
// prefecture要素は複製で作成
Node prefecture2 = prefecture1.cloneNode(true);
container2.appendChild(prefecture2);
// entity要素の作成
XBRLEntity entity2 = idoc.createEntity(identifier2, segment2);

// 数値コンテキストの作成
XBRLNumericContext numericContext =
    idoc.createNumericContext(
        entity2,
        period2,
        unit,
        null,   // scenario要素なし
        9,      // precision属性 : precision='9'
        false); // cwa属性 : cwa='false'
// インスタンス文書に数値コンテキストを追加
idoc.addContext(numericContext);
```

(3) コーディング結果

コーディング結果として，生成されるインスタンス文書は次のとおりです。

```
<xbrli:numericContext cwa="false" id="NC01" precision="9">
    <xbrli:entity>
        <xbrli:identifier scheme="">SAMP</xbrli:identifier>
        <xbrli:segment>
            <prefecture xmlns="http://www.hitachi.com/2003/XBRL">Tokyo</
prefecture>
        </xbrli:segment>
    </xbrli:entity>
    <xbrli:period>
        <xbrli:instant>2003-03-31</xbrli:instant>
    </xbrli:period>
    <xbrli:unit>
        <xbrli:measure>iso4217:JPY</xbrli:measure>
    </xbrli:unit>
</xbrli:numericContext>
```

4.4.4　非数値コンテキストに対応したアイテムの追加

XBRL2.0 に準拠したインスタンス文書に，アイテムを追加するサンプルコードについて説明します。

ここでは，「4.4.2　非数値コンテキスト要素の追加」で追加した非数値コンテキストに属する，次の非数値アイテム <hi:note> を追加します。

表 4-10 非数値コンテキストに対応したアイテムの値

項番	名称	要素 / 属性	値
1	nonNumericContext	属性	NNC01
2	hi:note	要素	This is a sample!
3	コンテキスト ID	属性	ITM01

(1) 使用するインターフェース

非数値コンテキストに対応したアイテムの追加で使用するインターフェースを次の表に示します。

表 4-11 非数値コンテキストに対応したアイテムの追加で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.instance.XBRLInstanceItem	アイテムを表します。
2	com.hitachi.xbrl.instance.XBRLNonNumericContext	非数値コンテキスト要素を表します。
3	com.hitachi.xbrl.instance.XBRLInstanceElement	インスタンス文書に記述される要素を表します。
4	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

非数値コンテキストに対応したアイテムを XBRL DOM に追加するコードは次のとおりです。

```
// 非数値アイテム - note要素 - の作成
XBRLInstanceItem nonNumericItem =
    idoc.createItem(
        "http://www.hitachi.com/2003/XBRL",
        "note",
        nonNumericContext);
// 非数値アイテムに値を設定
nonNumericItem.setString("This is a sample!");
// インスタンス文書に非数値アイテムを追加
idoc.appendChildElement((XBRLInstanceElement) nonNumericItem);
```

(3) コーディング結果

コーディング結果として，生成されるインスタンス文書は次のとおりです。

```
<hi:note id="ITM01" nonNumericContext="NNC01" >This is a sample!</hi:note>
```

4.4.5 数値コンテキストに対応したアイテムの追加

XBRL2.0 に準拠したインスタンス文書に，アイテムを追加するサンプルコードについて

説明します。

ここでは，「4.4.3　数値コンテキスト要素の追加」で追加した数値コンテキストに属する，次のアイテム要素を追加します。数値コンテキストを追加した際に unit/measure="iso4217:JPY" を指定し，かつタクソノミー文書の該当する要素で type="xbrli:monetaryItemType" を指定していれば，値の単位は円になります。

表 4-12　数値コンテキストに対応したアイテムの値

項番	名称	追加するアイテムとその値		
		hi:assets	hi:currentAssets	hi:investmentsAndAdvances
1	numericContext	NC01	NC01	NC01
2	値	100,000	50,000	20,000
3	コンテキスト ID	ITM02	ITM03	ITM04

(1) 使用するインターフェース

数値コンテキストに対応したアイテムの追加で使用するインターフェースを次の表に示します。

表 4-13　数値コンテキストに対応したアイテムの追加で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.instance.XBRLInstanceItem	アイテムを表します。
2	com.hitachi.xbrl.instance.XBRLInstanceElement	インスタンス文書に記述される要素を表します。
3	com.hitachi.xbrl.instance.XBRLNumericContext	数値コンテキストを表します。
4	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

数値コンテキストに対応したアイテムを XBRL DOM に追加するコードは次のとおりです。

```
// 数値アイテム - assets要素 - の作成
XBRLInstanceItem numericItem1 =
    idoc.createItem(
        "http://www.hitachi.com/2003/XBRL",
        "assets",
        numericContext);
// 数値アイテム - currentAssets要素 - の作成
XBRLInstanceItem numericItem2 =
    idoc.createItem(
        "http://www.hitachi.com/2003/XBRL",
        "currentAssets",
        numericContext);
// 数値アイテム - investmentsAndAdvances要素 - の作成
XBRLInstanceItem numericItem3 =
    idoc.createItem(
        "http://www.hitachi.com/2003/XBRL",
        "investmentsAndAdvances",
        numericContext);

// 数値アイテムに値を設定
BigDecimal v1 = new BigDecimal(100000);
BigDecimal v2 = new BigDecimal(50000);
BigDecimal v3 = new BigDecimal(20000);

numericItem1.setValue((Object) v1);
numericItem2.setValue((Object) v2);
numericItem3.setValue((Object) v3);

// インスタンス文書に数値アイテムを追加
idoc.appendChildElement((XBRLInstanceElement) numericItem1);
idoc.appendChildElement((XBRLInstanceElement) numericItem2);
idoc.appendChildElement((XBRLInstanceElement) numericItem3);
```

(3) コーディング結果

コーディング結果として，生成されるインスタンス文書は次のとおりです。

```
<hi:assets contextRef="C01" unitRef="U01" precision=10>100000</hi:assets>
<hi:currentAssets contextRef="C01" unitRef="U01" precision=10>50000</
hi:currentAssets>
<hi:investmentsAndAdvances contextRef="C01" unitRef="U01" precision=10>20000</
hi:investmentsAndAdvances>
```

4.4.6 インスタンス文書の保存

XBRL2.0 に準拠したインスタンス文書の XBRL DOM を，XML 形式のファイルに書き出して保存するサンプルコードについて説明します。

(1) 使用するインターフェース

インスタンス文書の保存で使用するインターフェースを次の表に示します。

表 4-14 インスタンス文書の保存で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.instance.XBRLInstanceDocumentWriterFactory	XBRLInstanceDocumentWriter オブジェクトを生成するための API を定義します。

項番	使用するインターフェース	説明
2	com.hitachi.xbrl.instance.XBRLInstanceDocumentWriter	インスタンス文書をストリームに書き出します。
3	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

XML 形式のファイルに書き出し，保存するコードは次のとおりです。XBRL DOM の内容をファイル「sample_instance1.xml」に出力します。

```
// XBRLInstanceDocumentWriterFactoryの作成
XBRLInstanceDocumentWriterFactory idwf =
    dm.newInstanceDocumentWriterFactory();

// XBRLInstanceDocumentWriterの作成
XBRLInstanceDocumentWriter idw = idwf.newInstanceDocumentWriter();

// ファイル出力ストリームの作成
FileOutputStream os =
    new FileOutputStream("output/sample_instance1.xml");

// インスタンス文書をファイル出力ストリームに書き込む
idw.write(os, idoc);
// ファイル出力ストリームを閉じる
os.close();
```

4.5　既存のインスタンス文書の編集（XBRL2.0の場合）

ここでは，uCosminexus Business Reporting Processor の API を利用して，既存のインスタンス文書を編集します。

ここで示すサンプルは，次の流れでコーディングしています。

1. **インスタンス文書の読み込み**
2. **脚注リンクの追加**
3. **インスタンス文書の再保存**

4.5.1　インスタンス文書の読み込み

XBRL2.0 に準拠したインスタンス文書を読み込み，XBRL DOM を生成するサンプルコードについて説明します。タクソノミー文書はインスタンス文書内で指定されているため，タクソノミー文書をプログラムで明示的に指定する必要はありません。インスタンス文書の XBRL DOM と同時にタクソノミー文書の XBRL DOM も生成されます。

ここでは，検証時にエラーや警告が発生した場合のレポートを受け取り，出力するエラーハンドラと，検証時にローカルに保存したスキーマ定義を使用させるエンティティリゾルバも実装します。エラーハンドラおよびエンティティリゾルバの詳細については，「4.13　エラーハンドラ」および「4.14　エンティティリゾルバ」を参照してください。

（1）使用するクラスおよびインターフェース

インスタンス文書の読み込みで使用するクラスおよびインターフェースを次の表に示します。

表 4-15　インスタンス文書の読み込みで使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.XBRLDocumentManager	XBRL 文書の入出力や，タクソノミー文書のキャッシュを管理します。
2	com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory	XBRLInstanceDocumentBuilder オブジェクトを生成するための API を定義します。
3	com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder	インスタンス文書から XBRLInstanceDocument オブジェクトを取得するための API を定義します。
4	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

インスタンス文書を読み込み，XBRL DOM を生成するコードは次のとおりです。

```
// 初期化
XBRLDocumentManager.init();
XBRLDocumentManager dm = XBRLDocumentManager.getInstance();

// XBRLInstanceDocumentBuilderFactoryの作成
XBRLInstanceDocumentBuilderFactory ibf =
    dm.newInstanceDocumentBuilderFactory();

// XBRLInstanceDocumentBuilderの作成
XBRLInstanceDocumentBuilder ib = ibf.newInstanceDocumentBuilder();

// エラーハンドラの設定
SampleErrorHandler eh = new SampleErrorHandler();
ib.setErrorHandler(eh);

// エンティティリゾルバの設定
SampleEntityResolver er = new SampleEntityResolver();
ib.setEntityResolver(er);

// インスタンス文書のパース
String filename = "input/sample_instance1.xml";
XBRLInstanceDocument idoc = ib.parse(new File(filename));

System.out.println("インスタンス文書のパースに成功しました。");
```

4.5.2　脚注リンクの追加

「4.5.1　インスタンス文書の読み込み」で読み込んだ XBRL2.0 に準拠するインスタンス文書の XBRL DOM に，脚注リンクを追加するサンプルコードについて説明します。

脚注リンクの構成要素は，ロケータ，リソース，アークから成り立ちます。それぞれ，<link:loc> 要素，<link:footnote> 要素，<link:footnoteArc> 要素を使用して記述します。

<link:footnoteArc> 要素では，一つのリンクを双方向で表現する必要があるため，ロケータ（<link:loc> 要素）からリソース（<link:footnote> 要素）へのアーク，およびリソース（<link:footnote> 要素）からロケータ（<link:loc> 要素）へのアークの両方が必要です。

uCosminexus Business Reporting Processor では，リソース（<link:footnote> 要素）を生成して関連づける要素に登録すると，ロケータ（<link:loc> 要素）とアーク（<link:footnoteArc> 要素）も自動で追加されます。

ここでは，非数値アイテムに日本語と英語の脚注リンクを追加します。二つの脚注リンクを追加するので，リソースおよびアークが 2 要素ずつ追加されます。

脚注リンクを追加する非数値アイテム <hi:note> 要素の値を次の表に示します。

表 4-16 脚注リンクを追加する非数値アイテムの値

項番	名称	要素 / 属性	値
1	nonNumericContext	属性	NNC01
2	hi:note	要素	This is a sample!

追加する <link:footnote> 要素を次の表に示します。

表 4-17 追加する <link:footnote> 要素

項番	属性名	値	
1	xlink:type	resource	
2	xlink:label	dst_point_ja	dst_point_en
3	xml:lang	ja	en

<link:footnote> 要素の値は日本語（ja）「これは脚注リンクです。」，英語（en）「This is a footnote-link.」とします。

追加される <link:loc> 要素および <link:footnoteArc> 要素を次の表に示します。

表 4-18 追加される <link:loc> 要素および <link:footnoteArc> 要素

要素名	属性名	値
<link:loc> 要素	xlink:type	locator
	xlink:label	src_point
	xlink:href ※	ITM01
<link:footnoteArc> 要素（<link:loc> 要素から <link:footnote> 要素へのアーク）	xlink:type	arc
	xlink:from	(locator で設定された label:src_point)
	xlink:to	(footnote で設定された label:dst_point_ja) (footnote で設定された label:dst_point_en)
	xlink:arcrole	http://www.xbrl.org/linkprops/arc/fact-footnote
	xlink:show	replace
	xlink:actuate	onRequest
<link:footnoteArc> 要素（<link:footnote> 要素から <link:loc> 要素へのアーク）	xlink:type	arc
	xlink:from	(footnote で設定された label:dst_point_ja) (footnote で設定された label:dst_point_en)

要素名	属性名	値
	xlink:to	(locator で設定された label:src_point)
	xlink:arcrole	http://www.xbrl.org/linkprops/arc/footnote-fact
	xlink:show	replace
	xlink:actuate	onRequest

注※　xlink:href 属性では，インスタンス文書の本文中の要素を id または xpointer（Xpath 方式）で指し示す必要があります。

(1) 使用するクラスおよびインターフェース

脚注リンクの追加で使用するクラスおよびインターフェースを次の表に示します。

表 4-19　脚注リンクの追加で使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.instance.XBRLFootnote	脚注を表します。
2	com.hitachi.xbrl.instance.XBRLInstanceElement	インスタンス文書に記述される要素を表します。
3	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

インスタンス文書の XBRL DOM に脚注リンクを追加するコードは次のとおりです。

```
// 脚注の作成(日本語)
XBRLFootnote footnote_ja = idoc.createFootnote("ja", null, null);
footnote_ja.setValue("これは脚注リンクです。");
// 脚注の作成(英語)
XBRLFootnote footnote_en = idoc.createFootnote("en", null, null);
footnote_en.setValue("This is a footnote-link.");

// すべてのnote要素(アイテム)を取得
List<XBRLInstanceElement> elist =
    idoc.getElements("http://www.hitachi.com/2003/XBRL", "note");

// 最初の要素に脚注を追加して関連づける
XBRLInstanceElement element = elist.get(0);
element.addFootnote(footnote_ja);
element.addFootnote(footnote_en);
```

(3) コーディング結果

コーディング結果として，生成される脚注リンクは次のとおりです。

```
<link:footnoteLink xlink:type="extended">
 <link:footnote xlink:label="src_point" xlink:type="resource" xml:lang="ja">
    これは脚注リンクです。</link:footnote>
 <link:loc xlink:href="#ITM01" xlink:label="dst_point_ja" xlink:type="locator"
/>
 <link:footnoteArc xlink:actuate="onRequest"
    xlink:arcrole="http://www.xbrl.org/linkprops/arc/footnote-fact"
    xlink:from="src_point" xlink:show="replace" xlink:to="dst_point_ja"
    xlink:type="arc" />
 <link:footnoteArc xlink:actuate="onRequest"
    xlink:arcrole="http://www.xbrl.org/linkprops/arc/fact-footnote"
    xlink:from="dst_point_ja" xlink:show="replace" xlink:to="src_point"
    xlink:type="arc" />
 <link:footnote xlink:label="dst_point_en" xlink:type="resource"
    xml:lang="en">This is a footnote-link.</link:footnote>
 <link:footnoteArc xlink:actuate="onRequest"
    xlink:arcrole="http://www.xbrl.org/linkprops/arc/footnote-fact"
    xlink:from="dst_point_en" xlink:show="replace" xlink:to="dst_point_ja"
    xlink:type="arc" />
 <link:footnoteArc xlink:actuate="onRequest"
    xlink:arcrole="http://www.xbrl.org/linkprops/arc/fact-footnote"
    xlink:from="dst_point_ja" xlink:show="replace" xlink:to="dst_point_en"
    xlink:type="arc" />
</link:footnoteLink>
```

4.5.3　インスタンス文書の再保存

「4.5.2　脚注リンクの追加」で脚注リンクを追加したインスタンス文書の XBRL DOM を，ファイルに保存します。

詳細については，「4.4.6　インスタンス文書の保存」を参照してください。

4.6　タクソノミー文書の参照と検証（XBRL2.1の場合）

ここでは，タクソノミー文書とリンクベースの参照と検証をします。

ここで示すサンプルは，次の流れでコーディングしています。

1. **タクソノミー文書の読み込み**
2. **タクソノミー本体とリンクベースの参照**

XBRL2.1 の場合は，リンクの関係を表す XBRLRelationship インターフェースを使用して，タクソノミー本体とリンクベースを参照することもできます。ここでは，XBRLRelationship インターフェースを使用したサンプルコードについても，説明します。

4.6.1　タクソノミー文書の読み込み

XBRL2.1 に準拠したタクソノミー文書を読み込み，XBRL DOM を生成するサンプルコードについて説明します。XBRL DOM の生成時にタクソノミー文書の妥当性を検証するには，setValidating() メソッドを実行します。また，XBRL2.1 の場合，タクソノミー文書の読み込み時に XBRL のバージョンを指定する必要があります。

ここでは，検証時にエラーや警告が発生した場合のレポートを受け取り，出力するエラーハンドラと，検証時にローカルに保存したスキーマ定義を使用するエンティティリゾルバも実装します。エラーハンドラおよびエンティティリゾルバの詳細については，「4.13　エラーハンドラ」および「4.14　エンティティリゾルバ」を参照してください。

（1）使用するクラスおよびインターフェース

タクソノミー文書の読み込みで使用するクラスおよびインターフェースを次の表に示します。

表 4-20　タクソノミー文書の読み込みで使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.XBRLDocumentManager	XBRL 文書の入出力や，タクソノミー文書のキャッシュを管理します。
2	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory	XBRLTaxonomyDocumentBuilder オブジェクトを生成するための API を定義します。
3	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder	タクソノミー文書から XBRLTaxonomyDocument オブジェクトを取得するための API を定義します。

項番	使用するクラスおよびインターフェース	説明
4	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument	タクソノミー文書全体を表します。
5	com.hitachi.xbrl.util.XBRLConstants	ユーザーアプリケーションで利用できる定数を定義します。

(2) サンプルコード

タクソノミー文書を読み込むコードは次のとおりです。

```
/*
 * タクソノミー文書を読み込む
 */

// 初期化
XBRLDocumentManager.init();
/*
 * プロパティファイルを利用して初期化する場合は，ファイルの絶対パスを指定します
 * 例:
 * XBRLDocumentManager.init("C://Program Files/HITACHI/cbrp/properties/
cbrp.properties");
 */
// XBRLDocumentManagerの取得
XBRLDocumentManager dm = XBRLDocumentManager.getInstance();

// XBRLTaxonomyDocumentBuilderFactoryの作成
XBRLTaxonomyDocumentBuilderFactory tbf =
    dm.newTaxonomyDocumentBuilderFactory();

// XBRLバージョンの設定
tbf.setXBRLVersion(XBRLConstants.XBRL_2_1);

// タクソノミー文書のパース時に妥当性検証を行う設定
// tbf.setValidating(true);

// XBRLTaxonomyDocumentBuilderの作成
XBRLTaxonomyDocumentBuilder tb = tbf.newTaxonomyDocumentBuilder();

// エラーハンドラの設定
SampleErrorHandler eh = new SampleErrorHandler();
tb.setErrorHandler(eh);

// エンティティリゾルバの設定
SampleEntityResolver er = new SampleEntityResolver();
tb.setEntityResolver(er);

// タクソノミー文書のパース
String filename = "input/sample_taxonomy1_xbrl_2_1.xsd";
File[] taxonomy = { new File(filename) };
XBRLTaxonomyDocument tdoc = tb.parse(taxonomy);

System.out.println("タクソノミー文書のパースに成功しました。");
```

さらに，cbrp.parse.validate.full-checking プロパティを使えば，XBRL 2.1 Conformance Suite 1.0 に示されたすべての検証を実行できます。ただし，このプロパティは cbrp.parse.validate プロパティが true の場合だけ有効です。

この検証では，関係ネットワーク上に禁止されているサイクル（循環など）がないかなど，多くの処理時間やメモリリソースを要する検証を実行します。信頼できるタクソノミー文書に対してはこの検証は必要ありませんが，新規作成したタクソノミー文書が

XBRL2.1 に完全に適合しているか検証したい場合などに有効です。

この検証を実行するには，次の内容をコードに追加してください。

```
//タクソノミー文書パース時に完全な妥当性検証を実施する
tbf.setValidating(true);
tbf.setProperty("cbrp.parse.validate.full-checking", Boolean.TRUE);
```

4.6.2 タクソノミー本体とリンクベースの参照（アイテム名またはタプル名を参照する場合）

XBRL2.1 に準拠したタクソノミー文書の読み込み時に生成した XBRL DOM から，要素の属性（アイテム名またはタプル名）を参照するサンプルコードについて説明します。要素がアイテムの場合は，その型を参照し，同時にタクソノミー本体に関連づけられたラベルをラベルリンク（リンクベース）から参照します。

(1) 使用するインターフェース

タクソノミー本体とリンクベースの参照で使用するインターフェースを次の表に示します。

表 4-21 タクソノミー本体とリンクベースの参照で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument	タクソノミー文書全体を表します。
2	com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement	タクソノミー文書に記述される要素の要素宣言を表します。
3	com.hitachi.xbrl.taxonomy.XBRLTaxonomyItem	アイテムの要素宣言を表します。
4	com.hitachi.xbrl.taxonomy.XBRLTaxonomyTuple	タプルの要素宣言を表します。
5	com.hitachi.xbrl.taxonomy.XBRLLabel	要素のラベルを表します。

(2) サンプルコード

タクソノミー本体とリンクベースを参照するコードは次のとおりです。

```
// タクソノミー文書のパース
String filename = "input/sample_taxonomy1_xbrl_2_1.xsd";
File[] taxonomy = { new File(filename) };
XBRLTaxonomyDocument tdoc = tb.parse(taxonomy);

System.out.println("タクソノミー文書のパースに成功しました。");

/*
 * ここからタクソノミー文書の内容を参照
 */

System.out.println("タクソノミー文書の要素宣言と，そのラベルを出力します。");

// 要素宣言(アイテムまたはタプルの定義)の一覧を取得
List elist = tdoc.getElements();
for (int i = 0; i < elist.size(); i++) {
    XBRLTaxonomyElement element =
        (XBRLTaxonomyElement) elist.get(i);
    switch (element.getElementType()) {
        case XBRLTaxonomyElement.ITEM_ELEMENT :
            // 要素宣言がアイテム
            XBRLTaxonomyItem item = (XBRLTaxonomyItem) element;
            // 名前の出力
            System.out.println("item:" + item.getName());
            // 型の出力
            if (item.getItemType()
                    == XBRLTaxonomyItem.MONETARY_ITEM) {
                // 型が通貨型(xbrli:monetaryItemType)の場合
                System.out.println(" ItemType:MONETARY_ITEM");
            } else if (
                item.getItemType()
                    == XBRLTaxonomyItem.STRING_ITEM) {
                // 型が文字列型(xbrli:stringItemType)の場合
                System.out.println(" ItemType:STRING_ITEM");
            }
            // 関連付けられているラベルの一覧を取得
            List lbllist = element.getLabels();
            // ラベルの出力
            for (int j = 0; j < lbllist.size(); j++) {
                XBRLLabel lbl = (XBRLLabel) lbllist.get(j);
                // 値と言語を出力
                System.out.println(
                    " Label:"
                        + lbl.getValue()
                        + "("
                        + lbl.getLang()
                        + ")");
            }
            break;

        case XBRLTaxonomyElement.TUPLE_ELEMENT :
            // 要素宣言がタプル
            XBRLTaxonomyTuple tuple = (XBRLTaxonomyTuple) element;
            System.out.println("Tuple:" + tuple.getName());
            break;

    }
}
```

(3) サンプルコードの実行結果

サンプルのタクソノミー文書を参照したときの実行結果を次に示します。

```
タクソノミー文書のパースに成功しました。
タクソノミー文書の要素宣言と，そのラベルを出力します。
item:assets
 ItemType:MONETARY_ITEM
 Label:資産合計(ja)
 Label:Assets(en)
item:currentAssets
 ItemType:MONETARY_ITEM
 Label:流動資産(ja)
 Label:Current assets(en)
item:investmentsAndAdvances
 ItemType:MONETARY_ITEM
 Label:投資及び貸付金(ja)
 Label:Investments and advances(en)
item:note
 ItemType:STRING_ITEM
```

4.6.3 タクソノミー本体とリンクベースの参照（タクソノミー文書の要素の階層を参照する場合）

XBRL2.1 に準拠したタクソノミー文書の読み込み時に生成した XBRL DOM から，プレゼンテーションリンクを参照して，タクソノミー文書の要素を階層表示するサンプルコードについて説明します。表示名には，ラベルを使用します。

このサンプルコードでは，リンクの関係を表すインターフェースの一つである，XBRLPresentationRelationship インターフェースを使用しています。リンクの関係を表すインターフェース（com.hitachi.xbrl.link パッケージ）は，XBRL2.1 の場合だけ使用できます。

（1）使用するインターフェース

タクソノミー本体とリンクベースの参照で使用するインターフェースを次の表に示します。

表 4-22 タクソノミー本体とリンクベースの参照で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument	タクソノミー文書全体を表します。
2	com.hitachi.xbrl.link.XBRLPresentationRelationship	プレゼンテーションリンクに記述されている関係のうち，一つの関係を示します。
3	com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement	タクソノミー文書に記述される要素の要素宣言を表します。
4	com.hitachi.xbrl.taxonomy.XBRLLabel	要素のラベルを表します。
5	com.hitachi.xbrl.util.XBRLConstants	ユーザーアプリケーションで利用できる定数を定義します。

（2）サンプルコード

タクソノミー本体とリンクベースを参照するコードは次のとおりです。

```
// タクソノミー文書のパース
String filename = "input/sample_taxonomy1_xbrl_2_1.xsd";
File[] taxonomy = { new File(filename) };
XBRLTaxonomyDocument tdoc = tb.parse(taxonomy);

System.out.println("タクソノミー文書のパースに成功しました。");

/*
 * ここからタクソノミー文書の内容を参照
 */

// 標準なプレゼンテーションリンクをツリー表示する
XBRLTaxonomyElement[] elements =
    tdoc.findPresentationRootElements(
        XBRLConstants.XBRL21_ROLE_LINK,
        XBRLConstants.XBRL21_ARCROLE_PARENT_CHILD);

if (elements != null) {
    for (int i = 0; i < elements.length; i++) {
        List<XBRLPresentationRelationship> preList =
            elements[i].getPresentationRelationships(
                XBRLConstants.XBRL21_ROLE_LINK,
                XBRLConstants.XBRL21_ARCROLE_PARENT_CHILD);

        // 標準なプレゼンテーションリンクのルート要素か？
        if (preList.size() > 0) {
            // 一つのツリーを表示する
            ProcessTaxonomy2.printPresentationTree(elements[i], "");
        }
    }
} else {
    System.out.println("標準なプレゼンテーションリンクはありません。");
}

/**
 * プレゼンテーションリンクとラベルを使ったツリー表示
 */
public static void printPresentationTree(
        XBRLTaxonomyElement element, String indent) {

    // 要素の表示
    XBRLLabel label = element.getLabel("ja", null);
    if (label != null) {
        // ラベルで表示
        System.out.println(indent + "+ " + label.getValue());
    } else {
        // 要素名で表示
        System.out.println(indent + "+ " + element.getName());
    }

    // 子孫要素の表示
    List<XBRLPresentationRelationship> preList =
        element.getPresentationRelationships(
            XBRLConstants.XBRL21_ROLE_LINK,
            XBRLConstants.XBRL21_ARCROLE_PARENT_CHILD);

    for (XBRLPresentationRelationship present : preList) {
        XBRLTaxonomyElement child = (XBRLTaxonomyElement) present.getTo();
        printPresentationTree(child, indent + "    ");
    }
}
```

(3) サンプルコードの実行結果

サンプルのタクソノミー文書を参照したときの実行結果を次に示します。

```
タクソノミー文書のパースに成功しました。
+ 資産合計
    + 流動資産
    + 投資及び貸付金
```

4.7 インスタンス文書の新規作成と保存（XBRL2.1 の場合）

ここでは，uCosminexus Business Reporting Processor の API を利用して，XBRL2.1 に準拠したインスタンス文書を新規に作成して保存します。

ここで示すサンプルは，次の流れでコーディングしています。

1. **インスタンス文書の新規作成**
2. **コンテキスト要素（context）の追加**
3. **単位要素（unit）の追加**
4. **非数値アイテム（nonNumericItem）の追加**
5. **数値コンテキストに対応したアイテム（NumericItem）の追加**
6. **インスタンス文書の保存**

また，XBRL2.1 の場合，インスタンス文書の id 属性を任意の形式で付けることもできます。ID に任意の形式を付ける場合のコーディングについても説明します。

ID を任意の形式で付けない場合，通常は id 属性が自動生成されます。しかし，このサンプルではコーディングをわかりやすくするために，id 属性などの値を仮定しています。

4.7.1 インスタンス文書の新規作成

XBRL2.1 に準拠したインスタンス文書を新規に作成するサンプルコードについて説明します。

インスタンス文書の新規作成では，まずスキーマとなるタクソノミー文書の XBRL DOM を生成します。生成したタクソノミー文書の XBRL DOM を基に，インスタンス文書の XBRL DOM を生成します。また，名前空間宣言と xsi:schemaLocation 属性の設定，および XBRL のバージョンの指定をします。

(1) 使用するクラスおよびインターフェース

インスタンス文書の新規作成で使用するクラスおよびインターフェースを次の表に示します。

表 4-23 インスタンス文書の新規作成で使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.XBRLDocumentManager	XBRL 文書の入出力や，タクソノミー文書のキャッシュを管理します。

項番	使用するクラスおよびインターフェース	説明
2	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory	XBRLTaxonomyDocumentBuilder オブジェクトを生成するための API を定義します。
3	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder	タクソノミー文書から XBRLTaxonomyDocument オブジェクトを取得するための API を定義します。
4	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument	タクソノミー文書全体を表します。
5	com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory	XBRLInstanceDocumentBuilder オブジェクトを生成するための API を定義します。
6	com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder	インスタンス文書から XBRLInstanceDocument オブジェクトを取得するための API を定義します。
7	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。
8	com.hitachi.xbrl.util.XBRLNamespaceDeclaration	名前空間接頭辞とそれに対応する名前空間 URI の組を管理します。
9	com.hitachi.xbrl.util.XBRLSchemaLocation	xsi:schemaLocation 属性に指定する名前空間 URI とスキーマ文書の位置を示す URI の組を管理します。
10	com.hitachi.xbrl.instance.XBRLSchemaRef	<link:schemaRef> 要素を表します。

(2) サンプルコード

インスタンス文書を新規に作成するコードは次のとおりです。

```
/*
 * 新たなインスタンス文書作成の前準備
 */


// XBRLバージョンの設定
Properties prop = new Properties();
prop.setProperty("cbrp.parse.XBRLVersion", "2.1");


// 初期化
XBRLDocumentManager.init(prop);
XBRLDocumentManager dm = XBRLDocumentManager.getInstance();


// XBRLTaxonomyDocumentBuilderFactoryの作成
XBRLTaxonomyDocumentBuilderFactory tbf =
    dm.newTaxonomyDocumentBuilderFactory();


// XBRLTaxonomyDocumentBuilderの作成
XBRLTaxonomyDocumentBuilder tb = tbf.newTaxonomyDocumentBuilder();
```

```
// エラーハンドラの設定
SampleErrorHandler eh = new SampleErrorHandler();
tb.setErrorHandler(eh);


// エンティティリゾルバの設定
SampleEntityResolver er = new SampleEntityResolver();
tb.setEntityResolver(er);

// タクソノミー文書のパース
String filename = "input/sample_taxonomy1_xbrl_2_1.xsd";
File[] taxonomy = { new File(filename) };
XBRLTaxonomyDocument tdoc = tb.parse(taxonomy);

System.out.println("タクソノミー文書のパースに成功しました。");


/*
 * ここから新たなインスタンス文書の作成
 */


// XBRLInstanceDocumentBuilderFactoryの作成
XBRLInstanceDocumentBuilderFactory ibf =
    dm.newInstanceDocumentBuilderFactory();


// XBRLInstanceDocumentBuilderの作成
XBRLInstanceDocumentBuilder ib = ibf.newInstanceDocumentBuilder();


// 新たなインスタンス文書の作成
XBRLInstanceDocument idoc = ib.newInstanceDocument(tdoc);


// 名前空間宣言 - hi - の作成
// (xmlns:hi="http://www.hitachi.com/2004/XBRL")
XBRLNamespaceDeclaration nd1 =
    new XBRLNamespaceDeclaration(
        "hi",
        "http://www.hitachi.com/2004/XBRL");


// xsi:schemaLocation属性値の作成
XBRLSchemaLocation sl =
    new XBRLSchemaLocation(
        "http://www.hitachi.com/2004/XBRL",
        "input/sample_taxonomy1_xbrl_2_1.xsd");


// schemaRef要素の作成
XBRLSchemaRef sr =
    idoc.createSchemaRef("input/sample_taxonomy1_xbrl_2_1.xsd", null);

// インスタンス文書に名前空間宣言とxsi:schemaLocation属性値を追加
idoc.addNamespaceDeclaration(nd1);
idoc.addSchemaLocation(sl);
// インスタンス文書にschemaRef要素を追加
idoc.addSchemaRef(sr);
```

4.7.2 コンテキスト要素の追加

XBRL2.1 に準拠したインスタンス文書の XBRL DOM に，コンテキスト要素を追加するサンプルコードについて説明します。コンテキスト要素を追加する場合，コンテキスト要素の子要素（entity，period など）を作成する必要があります。

ここで説明するサンプルコードでは，次の内容のコンテキスト要素を追加します。表中の「名称」欄のインデントは，コーディング結果のインデントを示しています。

表 4-24　コンテキスト要素

項番	名称	要素 / 属性	値
1	コンテキスト ID	属性	C01
2	時期	要素	2004-03-31
3	エンティティ（組織）	要素	―
4	識別子	要素	SAMP
5	スキーム	属性	http://www.hitachi.co.jp/2004/XBRL
6	セグメント	要素	―
7	prefecture ※	要素	Tokyo

（凡例）―：値を持つことができません。
注※　"prefecture" の名前空間 URI は "http://www.hitachi.com/2004/XBRL" とします。

(1) 使用するインターフェース

コンテキスト要素の追加で使用するインターフェースを次の表に示します。

表 4-25　コンテキスト要素の追加で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.types.XBRLDateUnion	XBRL のデータ型 dateUnion を表します。
2	com.hitachi.xbrl.instance.XBRLPeriod	時期要素を表します。
3	com.hitachi.xbrl.instance.XBRLIdentifier	識別子要素を表します。
4	com.hitachi.xbrl.instance.XBRLSegment	セグメント要素を表します。
5	com.hitachi.xbrl.instance.XBRLEntity	エンティティ要素を表します。
6	com.hitachi.xbrl.instance.XBRLContext	コンテキスト要素を表します。
7	com.hitachi.xbrl.instance.XBRLContext2	XBRL2.1 のコンテキスト要素を表します。
8	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

コンテキスト要素を XBRL DOM に追加するコードは次のとおりです。

```
// period要素の作成
XBRLDateUnion date = new XBRLDateUnion("2004-03-31");
XBRLPeriod period = idoc.createPeriod(date);
// identifier要素の作成
XBRLIdentifier identifier =
    idoc.createIdentifier(
        "http://www.hitachi.com/2004/XBRL",
        "SAMP");
// segment要素の作成
XBRLSegment segment = idoc.createSegment();

// segment要素に対応するorg.w3c.dom.Elementの取得
Node container = segment.getContainerNode();
// インスタンス文書に対応するorg.w3c.dom.Documentの取得
Document doc = (Document) idoc.getW3CDOMNode();
// prefecture要素の作成
Element prefecture =
    doc.createElementNS(
        "http://www.hitachi.com/2004/XBRL",
        "prefecture");
prefecture.appendChild(doc.createTextNode("Tokyo"));
// segment要素の子要素にprefecture要素を追加
container.appendChild(prefecture);

// entity要素の作成
XBRLEntity entity = idoc.createEntity(identifier, segment);

// コンテキストの作成
XBRLContext2 context = idoc.createContext(entity, period, null);
// インスタンス文書にコンテキストを追加
idoc.addContext((XBRLContext) context);
```

(3) コーディング結果

コーディング結果として，生成されるインスタンス文書は次のとおりです。

```
<xbrli:context id="C01">
    <xbrli:entity>
     <xbrli:identifier
       schema="http://www.hitachi.com/2004/XBRL">SAMP</xbrli:identifier>
     <xbrli:segment>
       <prefecture xmlns="http://www.hitachi.com/2004/XBRL">Tokyo</prefecture>
     </xbrli:segment>
    </xbrli:entity>
    <xbrli:period>
       <xbrli:instant>2004-03-31</xbrli:instant>
    </xbrli:period>
</xbrli:context>
```

4.7.3　単位要素の追加

XBRL2.1 に準拠したインスタンス文書の XBRL DOM に単位要素（unit）を追加するサンプルコードについて説明します。

ここでは，「4.7.2　コンテキスト要素の追加」でコンテキスト要素を追加した XBRL DOM に，さらに単位要素を追加します。「iso4217:JPY」を指定することで，値の単位は「円」になります。表中の「名称」欄のインデントは，コーディング結果のインデントを示しています。

表 4-26　単位要素

項番	名称	要素 / 属性	値
1	ユニット ID	属性	U01
2	計量	要素	iso4217:JPY

(1) 使用するクラスおよびインターフェース

単位要素の追加で使用するクラスおよびインターフェースを次の表に示します。

表 4-27　単位要素の追加で使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.types.XBRLQName	XML Schema のデータ型 QName を表します。
2	com.hitachi.xbrl.instance.XBRLUnit2	単位要素を表します。
3	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

単位要素を XBRL DOM に追加するコードは次のとおりです。

```
// measure要素の作成
XBRLQName currency =
          new XBRLQName("http://www.xbrl.org/2003/iso4217", "JPY");
XBRLQName[] measures = { currency };

// ユニット(単位)の作成
XBRLUnit2 unit = idoc.createUnit(measures);
// インスタンス文書にユニット(単位)を追加
idoc.addUnit(unit);
```

(3) コーディング結果

コーディング結果として，生成されるインスタンス文書は次のとおりです。

```
<xbrli:unit id="U01">
  <xbrli:measure>iso4217:JPY</xbrli:measure>
</xbrli:unit>
```

4.7.4　非数値アイテムの追加

XBRL2.1 に準拠したインスタンス文書に，コンテキスト要素に対応する非数値アイテムを追加するサンプルコードについて説明します。

ここで説明するサンプルコードでは，「4.7.2　コンテキスト要素の追加」で追加したコンテキスト要素に属する，次に示す <hi:note> という名前の非数値アイテムを追加します。名前空間接頭辞 "hi" の名前空間 URI は，"http://www.hitachi.com/2004/XBRL" です。

表 4-28 追加する非数値アイテムの値

項番	名称	要素 / 属性	値
1	nonNumericContext	属性	C01
2	hi:note	要素	This is a sample!

(1) 使用するインターフェース

非数値アイテムの追加で使用するインターフェースを次の表に示します。

表 4-29 非数値アイテムの追加で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.instance.XBRLInstanceItem	アイテムを表します。
2	com.hitachi.xbrl.instance.XBRLContext2	XBRL2.1 のコンテキスト要素を表します。
3	com.hitachi.xbrl.instance.XBRLInstanceElement	インスタンス文書に記述される要素を表します。
4	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

非数値アイテムを XBRL DOM に追加するコードは次のとおりです。

```
// 非数値アイテム - note要素 - の作成
XBRLInstanceItem nonNumericItem =
    idoc.createNonNumericItem(
        "http://www.hitachi.com/2004/XBRL", "note", context);
// 非数値アイテムに値を設定
nonNumericItem.setString("This is a sample!");
// インスタンス文書に非数値アイテムを追加
idoc.appendChildElement((XBRLInstanceElement) nonNumericItem);
```

(3) コーディング結果

コーディング結果として，生成されるインスタンス文書は次のとおりです。

```
<hi:note contextRef="C01">This is a sample!</hi:note>
```

4.7.5 数値アイテムの追加

XBRL2.1 に準拠したインスタンス文書に，コンテキスト要素に対応する数値アイテムを追加するサンプルコードについて説明します。

ここでは，「4.7.2 コンテキスト要素の追加」で追加したコンテキスト要素に属し，かつ「4.7.3 単位要素の追加」で追加した単位要素に属する，次に示す数値アイテムを追加します。

表 4-30 数値アイテムの値

項番	名称	追加するアイテムとその値		
		hi:assets	hi:currentAssets	hi:investmentsAndAdvances
1	contextRef	C01	C01	C01
2	unitRef	U01	U01	U01
3	precision	10	10	10
4	値	100,000	50,000	20,000

(1) 使用するクラスおよびインターフェース

数値コンテキストに対応したアイテムの追加で使用するクラスおよびインターフェースを次の表に示します。

表 4-31 数値コンテキストに対応したアイテムの追加で使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.types.XBRLPrecision	数値アイテムの precision 属性に指定する値を表します。
2	com.hitachi.xbrl.instance.XBRLInstanceItem	アイテムを表します。
3	com.hitachi.xbrl.instance.XBRLContext2	XBRL2.1 のコンテキスト要素を表します。
4	com.hitachi.xbrl.instance.XBRLUnit2	XBRL2.1 の単位要素を表します。
5	com.hitachi.xbrl.instance.XBRLInstanceElement	インスタンス文書に記述される要素を表します。
6	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

数値コンテキストに対応したアイテムを XBRL DOM に追加するコードは次のとおりです。

```
// precision属性値の作成
XBRLPrecision precision = new XBRLPrecision("10");

// 数値アイテム - assets要素 - の作成
XBRLInstanceItem numericItem1 =
    idoc.createNumericItem(
        "http://www.hitachi.com/2004/XBRL",
        "assets",
        context,
        unit,
        precision,
        null);
// 数値アイテム - currentAssets要素 - の作成
XBRLInstanceItem numericItem2 =
    idoc.createNumericItem(
        "http://www.hitachi.com/2004/XBRL",
        "currentAssets",
        context,
        unit,
        precision,
        null);
// 数値アイテム - investmentsAndAdvances要素 - の作成
XBRLInstanceItem numericItem3 =
    idoc.createNumericItem(
        "http://www.hitachi.com/2004/XBRL",
        "investmentsAndAdvances",
        context,
        unit,
        precision,
        null);

// 数値アイテムに値を設定
BigDecimal v1 = new BigDecimal(100000);
BigDecimal v2 = new BigDecimal(50000);
BigDecimal v3 = new BigDecimal(20000);

numericItem1.setValue((Object) v1);
numericItem2.setValue((Object) v2);
numericItem3.setValue((Object) v3);

// インスタンス文書に数値アイテムを追加
idoc.appendChildElement((XBRLInstanceElement) numericItem1);
idoc.appendChildElement((XBRLInstanceElement) numericItem2);
idoc.appendChildElement((XBRLInstanceElement) numericItem3);
```

(3) コーディング結果

コーディング結果として，生成されるインスタンス文書は次のとおりです。

```
<hi:assets contextRef="C01" unitRef="U01" precision=10>100000</hi:assets>
<hi:currentAssets contextRef="C01" unitRef="U01" precision=10>50000</
hi:currentAssets>
<hi:investmentsAndAdvances contextRef="C01" unitRef="U01" precision=10>20000</
hi:investmentsAndAdvances>
```

4.7.6 インスタンス文書の保存

XBRL2.1 に準拠したインスタンス文書の XBRL DOM を，XML 形式のファイルに書き出して保存するサンプルコードについて説明します。

(1) 使用するインターフェース

インスタンス文書の保存で使用するインターフェースを次の表に示します。

表 4-32 インスタンス文書の保存で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.instance.XBRLInstanceDocumentWriterFactory	XBRLInstanceDocumentWriter オブジェクトを生成するための API を定義します。
2	com.hitachi.xbrl.instance.XBRLInstanceDocumentWriter	インスタンス文書をストリームに書き出します。
3	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

XML 形式のファイルに書き出し，保存するコードは次のとおりです。XBRL DOM の内容をファイル「sample_instance1_xbrl_2_1.xml」に出力します。

```
// XBRLInstanceDocumentWriterFactoryの作成
XBRLInstanceDocumentWriterFactory idwf =
    dm.newInstanceDocumentWriterFactory();

// XBRLInstanceDocumentWriterの作成
XBRLInstanceDocumentWriter idw = idwf.newInstanceDocumentWriter();

// ファイル出力ストリームの作成
FileOutputStream os =
    new FileOutputStream("output/sample_instance1_xbrl_2_1.xml");

// インスタンス文書をファイル出力ストリームに書き込む
idw.write(os, idoc);
// ファイル出力ストリームを閉じる
os.close();
```

4.7.7 ID の採番をするインスタンス文書の作成

インスタンス文書に記述されている ID は，XML 文書内で一意にする必要があります。ID には，アイテムやタプルの id 属性，脚注リンクのロケータの xlink:loc 属性，リソースの xlink:label 属性など，複数あります。XBRL2.1 の場合，uCosminexus Business Reporting Processor では，ユーザー任意の形式でこれらの ID を付けることができます。

任意の形式で id 属性を付ける場合，あらかじめ ID 採番クラスをユーザーが作成しておきます。ID を採番したいインスタンス文書の新規作成時，インスタンス文書を表す XBRLInstanceDocument オブジェクトに ID 採番クラスのオブジェクトを設定することで，インスタンス文書に ID を付けることができます。

ここでは，使用するインターフェース，ID 採番クラスの作成例，およびインスタンス文書の新規作成時に ID 採番クラスを設定する例を示します。

(1) 使用するクラスおよびインターフェース

ID の採番で使用するクラスおよびインターフェースを次の表に示します。

表 4-33 ID の採番で使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.util.XBRLIdCreator	XBRL 文書の ID を生成するためのインターフェースです。ユーザーはこのインターフェースを実装して，ID 採番クラスを作成します。
2	com.hitachi.xbrl.util.XBRLIdManager	XBRL 文書に記述された ID を管理します。ユーザーが新規にインスタンスを生成できます。また，インスタンス文書を表す XBRLInstanceDocument オブジェクトの getIdManager() メソッドを使用して，ID を管理する XBRLIdManager オブジェクトを取得できます。

(2) ID 採番クラスの作成

XBRLIdCreator インターフェースを実装して，ユーザー独自の ID 採番クラスを作成します。実装した ID 採番クラスのオブジェクトを，ID を一元管理する XBRLIdManager オブジェクトに設定します。その結果，ID が必要になったときに採番オブジェクトがコールバックされ，指定した形式で ID が採番されます。

ID 採番クラスのオブジェクトを XBRLIdManager オブジェクトに設定するには，XBRLIdManager クラスの setIdCreator() メソッドを使用します。

XBRLIdCreator インターフェースを実装したサンプルコード（MyIdCreator.java）は次のとおりです。

```
import java.text.NumberFormat;

import com.hitachi.xbrl.util.XBRLIdCreator;
import com.hitachi.xbrl.util.XBRLIdManager;


/**
 * XBRLIdCreator の実装クラス
 */
public class MyIdCreator implements XBRLIdCreator {

    // 要素別カウンター
    private int fnt_counter = 0;
    private int loc_counter = 0;
    private int id_counter = 0;


    // コンストラクタ
    public MyIdCreator() {
    }

    /**
     * IDを生成する
     * フォーマット：{要素を識別する文字列}_{要素別の通し番号}
     */
    public String createId(int type, XBRLIdManager idManager) {
        String rtnValue = null;
```

```
        switch (type) {
        case FOOTNOTE_LABEL:
            rtnValue = createFootnoteLabel(idManager);
            break;
        case LOC_LABEL:
            rtnValue = createLocatorLabel(idManager);
            break;
        case ITEM_ID:
        case TUPLE_ID:
        case XBRL_ID:
        case CONTEXT_ID:
        case UNIT_ID:
        case FOOTNOTE_LINK_ID:
        case FOOTNOTE_ID:
            rtnValue = createElementId(idManager);
            break;
        }


        return rtnValue;
    }


    /**
     * ID の重複を調べて重複しない ID を返す
     */
    private static String checkId(String str, int counter, XBRLIdManager
idManager) {
        String rtnValue = null;


        NumberFormat nf = NumberFormat.getInstance();
        nf.setGroupingUsed(false);
        nf.setMinimumIntegerDigits(3);

        rtnValue = str + nf.format(counter);

        if (idManager.isId(rtnValue)) {
            rtnValue = null;
        }


        return rtnValue;
    }

    /**
     * 脚注のラベルを生成
     */
    private String createFootnoteLabel(XBRLIdManager idManager) {
        final String prefix = "FNT";
        String rtnValue = null;

        while (null == rtnValue) {
            fnt_counter++;
            rtnValue = checkId(prefix, fnt_counter, idManager);
        }


        return rtnValue;
    }


    /**
     * ロケータのラベルを生成
     */
    private String createLocatorLabel(XBRLIdManager idManager) {
        final String prefix = "LOC";
        String rtnValue = null;
```

```
        while (null == rtnValue) {
            loc_counter++;
            rtnValue = checkId(prefix, loc_counter, idManager);
        }


        return rtnValue;
    }

    /**
     * 要素の ID を生成
     */
    private String createElementId(XBRLIdManager idManager) {
        final String prefix = "ID";
        String rtnValue = null;


        while (null == rtnValue) {
            id_counter++;
            rtnValue = checkId(prefix, id_counter, idManager);
        }

        return rtnValue;
    }
}
```

(3) ID 採番クラスのインスタンス文書への組み込み

作成した ID 採番クラスをインスタンス文書に組み込みます。

ID 採番クラスの組み込みでは，次が必要です。

- ID 採番クラスのオブジェクトを，ID を一元管理する XBRLIdManager オブジェクトに設定する。
  XBRLIdManager クラスの setIdCreator() メソッドを使用します。
- インスタンス文書全体を示す XBRLInstanceDocument オブジェクトを XBRLIdManager オブジェクトに登録する。
  XBRLIdManager クラスの registerDocument() メソッドを使用します。

一つの XML 文書内に複数のインスタンス文書がある場合

XBRLIdManager は，複数の XBRLInstanceDocument オブジェクトを管理できます。その結果，一つの XML 文書内にある複数のインスタンス文書間で，ID を一意にできます。

一つの XML 文書内にある複数のインスタンス文書間で ID を一意にしたい場合，同じ ID 採番クラスのオブジェクトを XBRLIdManager オブジェクトに設定し，すべての XBRLInstanceDocument オブジェクトを同じ XBRLIdManager オブジェクトに登録します。

ID 採番クラスを組み込むサンプルコードは次のとおりです。

```
 :
 (中略)
 :
// 新たなインスタンス文書の作成
XBRLInstanceDocument idoc = ib.newInstanceDocument(tdoc);

/*
 * === ID採番方法の設定 ===
 * XBRLIdManagerを生成し，作成した自動採番方法を設定
 * このXBRLIdManagerでインスタンス文書を管理させ，以降設定した自動採番方法でIDを採番する
 */
MyIdCreator myCreator = new MyIdCreator();
XBRLIdManager im = new XBRLIdManager();
im.setIdCreator((XBRLIdCreator) myCreator);
im.registerDocument(idoc);
 :
 (中略)
 :
```

4.8　既存のインスタンス文書の編集（XBRL2.1の場合）

ここでは，uCosminexus Business Reporting Processor の API を利用して，既存のインスタンス文書を編集します。

ここで示すサンプルは，次の流れでコーディングしています。

1. **インスタンス文書の読み込み**
2. **脚注リンクの追加**
3. **インスタンス文書の再保存**

4.8.1　インスタンス文書の読み込み

XBRL2.1 に準拠したインスタンス文書を読み込み，XBRL DOM を生成します。タクソノミー文書はインスタンス文書内で指定されているため，タクソノミー文書をプログラムで明示的に指定する必要はありません。インスタンス文書の XBRL DOM と同時にタクソノミー文書の XBRL DOM も生成されます。ただし，XBRL のバージョンの指定は必要です。

ここでは，検証時にエラーや警告が発生した場合のレポートを受け取り，出力するエラーハンドラと，検証時にローカルに保存したスキーマ定義を使用させるエンティティリゾルバも実装します。エラーハンドラおよびエンティティリゾルバの詳細については，「4.13　エラーハンドラ」および「4.14　エンティティリゾルバ」を参照してください。

(1) 使用するクラスおよびインターフェース

インスタンス文書の読み込みで使用するクラスおよびインターフェースを次の表に示します。

表 4-34　インスタンス文書の読み込みで使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.XBRLDocumentManager	XBRL 文書の入出力や，タクソノミー文書のキャッシュを管理します。
2	com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory	XBRLInstanceDocumentBuilder オブジェクトを生成するための API を定義します。
3	com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder	インスタンス文書から XBRLInstanceDocument オブジェクトを取得するための API を定義します。

項番	使用するクラスおよびインターフェース	説明
4	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。
5	com.hitachi.xbrl.util.XBRLConstants	ユーザーアプリケーションで利用できる定数を定義します。

(2) サンプルコード

インスタンス文書を読み込み，XBRL DOM を生成するコードは次のとおりです。

```
/*
 * 既存インスタンス文書の読み込み
 */

// 初期化
XBRLDocumentManager.init();
XBRLDocumentManager dm = XBRLDocumentManager.getInstance();

// XBRLInstanceDocumentBuilderFactoryの作成
XBRLInstanceDocumentBuilderFactory ibf =
    dm.newInstanceDocumentBuilderFactory();

// XBRLバージョンの設定
ibf.setXBRLVersion(XBRLConstants.XBRL_2_1);

// XBRLInstanceDocumentBuilderの作成
XBRLInstanceDocumentBuilder ib = ibf.newInstanceDocumentBuilder();

// エラーハンドラの設定
SampleErrorHandler eh = new SampleErrorHandler();
ib.setErrorHandler(eh);

// エンティティリゾルバの設定
SampleEntityResolver er = new SampleEntityResolver();
ib.setEntityResolver(er);

// インスタンス文書のパース
String filename = "input/sample_instance1_xbrl_2_1.xml";
XBRLInstanceDocument idoc = ib.parse(new File(filename));

System.out.println("インスタンス文書のパースに成功しました。");
```

4.8.2 脚注リンクの追加

XBRL2.1 に準拠したインスタンス文書の XBRL DOM に脚注リンクを追加します。

脚注リンクの構成要素は，ロケータ，リソース，アークから成り立ちます。それぞれ，<link:loc> 要素，<link:footnote> 要素，<link:footnoteArc> 要素（<link:loc> 要素から <link:footnote> 要素へのアーク）を使用して記述します。

uCosminexus Business Reporting Processor では，リソース（<link:footnote> 要素）を生成して関連づける要素に登録すると，ロケータ（<link:loc> 要素）とアーク（<link:footnoteArc> 要素）も自動で追加されます。

ここでは，「4.8.1　インスタンス文書の読み込み」で読み込んだ XBRL2.1 に準拠したインスタンス文書の XBRL DOM に，日本語と英語の脚注リンクを追加します。二つの脚注リンクを追加するので，リソースおよびアークが 2 要素ずつ追加されます。

脚注リンクを追加する非数値アイテム <hi:note> 要素の値を次の表に示します。

表 4-35　脚注リンクを追加する非数値アイテムの値

項番	名称	要素 / 属性	値
1	contextRef	属性	C01
2	hi:note	要素	This is a sample!
3	コンテキスト ID	属性	NNI01

追加する <link:footnote> 要素を次の表に示します。

表 4-36　追加する <link:footnote> 要素

項番	属性名	値	
1	xlink:type	resource	
2	xlink:label	dst_point_ja	dst_point_en
3	xml:lang	ja	en

<link:footnote> 要素の値は日本語（ja）「これは脚注リンクです。」，英語（en）「This is a footnote-link.」とします。

追加される <link:loc> 要素，および <link:footnoteArc> 要素要素を次の表に示します。

表 4-37　追加される <link:loc> 要素，および <link:footnoteArc> 要素

要素名	属性名	値
<link:loc> 要素	xlink:type	locator
	xlink:label	src_point
	xlink:href ※	NNI01
<link:footnoteArc> 要素（<link:loc> 要素から <link:footnote> 要素へのアーク）	xlink:type	arc
	xlink:from	(locator で設定された label:src_point)
	xlink:to	(footnote で設定された label:dst_point_ja) (footnote で設定された label:dst_point_en)
	xlink:arcrole	http://www.xbrl.org/2003/arcrole/fact-footnote

注※　xlink:href 属性では，インスタンス文書の本文中の要素を id または xpointer（Xpath 方式）

で指し示す必要があります。

(1) 使用するインターフェース

脚注リンクの追加で使用するインターフェースを次の表に示します。

表 4-38 脚注リンクの追加で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.instance.XBRLFootnote	脚注を表します。
2	com.hitachi.xbrl.instance.XBRLInstanceElement	インスタンス文書に記述されている要素を表します。
3	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。

(2) サンプルコード

インスタンス文書の XBRL DOM に脚注リンクを追加するコードは次のとおりです。

```
// 脚注の作成(日本語)
XBRLFootnote footnote_ja = idoc.createFootnote("ja", null, null);
footnote_ja.setValue("これは脚注リンクです。");
// 脚注の作成(英語)
XBRLFootnote footnote_en = idoc.createFootnote("en", null, null);
footnote_en.setValue("This is a footnote-link.");

// すべてのnote要素(アイテム)を取得
List<XBRLInstanceElement> elist =
    idoc.getElements("http://www.hitachi.com/2004/XBRL", "note");

// 最初の要素に脚注を追加して関連づける
XBRLInstanceElement element = elist.get(0);
element.addFootnote(footnote_ja);
element.addFootnote(footnote_en);
```

(3) コーディング結果

コーディング結果として，生成される脚注リンクは次のとおりです。

```
<link:footnoteLink xlink:role="http://www.xbrl.org/2003/role/xlink"
xlink:type="extended">
  <link:footnote xlink:label="dst_point_ja" xlink:type="resource"
xml:lang="ja">
    これは脚注リンクです。</link:footnote>
  <link:loc xlink:href="#NNI01" xlink:label="src_point" xlink:type="locator" />
  <link:footnoteArc order="1"
    xlink:arcrole=http://www.xbrl.org/2003/arcrole/fact-footnote
    xlink:from="src_point" xlink:to="dst_point_ja" xlink:type="arc" />
  <link:footnote xlink:label="dst_point_en" xlink:type="resource"
    xml:lang="en">This is a footnote-link.</link:footnote>
  <link:footnoteArc order="1"
    xlink:arcrole="http://www.xbrl.org/2003/arcrole/fact-footnote"
    xlink:from="src_point" xlink:to="dst_point_en" xlink:type="arc" />
</link:footnoteLink>
```

4.8.3　インスタンス文書の再保存

「4.8.2　脚注リンクの追加」で脚注リンクを追加したインスタンス文書の XBRL DOM をファイルに保存します。

詳細は「4.7.6　インスタンス文書の保存」を参照してください。

4.9 タクソノミー文書の拡張（XBRL2.1 の場合）

この節では，XBRL2.1 に準拠している既存のタクソノミー文書を読み込んでアイテム定義やラベルを追加し，新たなタクソノミー文書を作る方法を説明します。既存のタクソノミー文書の内容は更新しません。

この節では，クラスまたはインターフェースとサンプルコードについて，次の順に説明します。

1. **タクソノミー文書の拡張**
2. **アイテム定義の追加**
3. **ラベルの追加**
4. **関係の禁止**
5. **タクソノミー文書の保存**

4.9.1 タクソノミー文書の拡張

既存のタクソノミー文書を読み込んで編集できる状態にするためのサンプルコードについて説明します。

（1）使用するクラスおよびインターフェース

タクソノミー文書の拡張で使用するクラスおよびインターフェースを次の表に示します。

表 4-39 タクソノミー文書の拡張で使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.XBRLDocumentManager	XBRL 文書の入出力やタクソノミー文書のキャッシュを管理します。
2	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory	XBRLTaxonomyDocumentBuilder オブジェクトを生成するための API を定義します。
3	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder	タクソノミー文書から XBRLTaxonomyDocument オブジェクトを取得するための API を定義します。
4	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument	タクソノミー文書全体を表します。

項番	使用するクラスおよびインターフェース	説明
5	com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomyBuilderFactory	SimpleTaxonomyBuilder オブジェクトを生成するための API を定義します。
6	com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomyBuilder	SimpleTaxonomy オブジェクトを生成するための API を定義します。
7	com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy	編集できる単純なタクソノミー文書全体を表します。
8	com.hitachi.xbrl.util.XBRLConstants	ユーザーアプリケーションで利用できる定数を定義します。

(2) サンプルコード

タクソノミー文書を拡張するコードには，次の処理を定義する必要があります。

1. 拡張元となる既存のタクソノミー文書を読み込み，XBRL DOM を生成する
2. 生成された XBRL DOM を基に，編集可能なタクソノミー文書の XBRL DOM を新たに生成する

タクソノミー文書を拡張するコードは次のとおりです。

```
/*
 * タクソノミー文書を読み込む
 */

// 初期化
XBRLDocumentManager.init();
XBRLDocumentManager dm = XBRLDocumentManager.getInstance();

// XBRLTaxonomyDocumentBuilderFactoryの作成
XBRLTaxonomyDocumentBuilderFactory tbf =
    dm.newTaxonomyDocumentBuilderFactory();

// XBRLバージョンの設定
tbf.setXBRLVersion(XBRLConstants.XBRL_2_1);

// タクソノミー文書のパース時に妥当性検証を行う設定
// tbf.setValidating(true);

//XBRLTaxonomyDocumentBuilderの作成
XBRLTaxonomyDocumentBuilder tb = tbf.newTaxonomyDocumentBuilder();

// エラーハンドラの設定
SampleErrorHandler eh = new SampleErrorHandler();
tb.setErrorHandler(eh);

// エンティティリゾルバの設定
SampleEntityResolver er = new SampleEntityResolver();
tb.setEntityResolver(er);

// タクソノミー文書のパース
String filename = "input/sample_taxonomy1_xbrl_2_1.xsd";
File[] taxonomy = { new File(filename) };
XBRLTaxonomyDocument tdoc = tb.parse(taxonomy);

System.out.println("タクソノミー文書のパースに成功しました。");

/*
 * ここからタクソノミー文書の拡張
 */

// SimpleTaxonomyBuilderFactoryの作成
SimpleTaxonomyBuilderFactory sbf =
    dm.newSimpleTaxonomyBuilderFactory();

// SimpleTaxonomyBuilderの作成
SimpleTaxonomyBuilder sb = sbf.newSimpleTaxonomyBuilder();

// タクソノミー文書(sample_taxonomy1_xbrl_2_1.xsd)を拡張元として
// 単純タクソノミー文書を表すSimpleTaxonomyを作成
SimpleTaxonomy sdoc =
    sb.newSimpleTaxonomy(
        "http://www.hitachi.com/2004/XBRL/extension",
        "hi2",
        "simple_extension_taxonomy.xsd",
        tdoc);
```

4.9.2　アイテム定義の追加

「4.9.1　タクソノミー文書の拡張」で編集可能な状態にしたタクソノミー文書に対して，アイテム定義を追加するためのサンプルコードについて説明します。

(1) 使用するインターフェース

アイテム定義を追加する際に使用するインターフェースを次の表に示します。

表 4-40 アイテム定義を追加する際に使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.simple.taxonomy.SimpleItemDefinition	単純なタクソノミー文書のアイテムの要素宣言を表します。
2	com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy	編集できる単純なタクソノミー文書全体を表します。

(2) サンプルコード

アイテム定義を XBRL DOM に追加するコードは次のとおりです。

```
// アイテムの定義を作成
SimpleItemDefinition item1 =
    sdoc.createItemDefinition(
        "CashDeposits",
        SimpleItemDefinition.MONETARY_ITEM,
        SimpleItemDefinition.DURATION);
// アイテムの定義を追加
sdoc.addElementDefinition(item1);
```

(3) コーディング結果

コーディング結果として，追加されるアイテム定義は次のとおりです。

```
<element id="hi2_CashDeposits"
         name="CashDeposits"
         substitutionGroup="xbrli:item"
         type="xbrli:monetaryItemType"
         xbrli:periodType="duration" />
```

4.9.3 ラベルの追加

ラベルを追加するサンプルコードについて説明します。

(1) 使用するクラスおよびインターフェース

ラベルを追加する際に使用するクラスおよびインターフェースを次の表に示します。

表 4-41 ラベルを追加する際に使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.simple.taxonomy.SimpleLabel	単純なタクソノミー文書のラベルを表します。
2	com.hitachi.xbrl.simple.taxonomy.SimpleLabelEdge	単純なタクソノミー文書を構成するラベルリンクの一辺（一つの関係）を表します。
3	com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy	編集できる単純なタクソノミー文書全体を表します。
4	com.hitachi.xbrl.util.XBRLConstants	ユーザーアプリケーションで利用できる定数を定義します。

(2) サンプルコード

ラベルを XBRL DOM に追加するコードは次のとおりです。

```
// ラベルを作成
SimpleLabel label_ja =
    sdoc.createLabel(
        XBRLConstants.XBRL21_ROLE_LABEL,
        "ja",
        "現預金");
SimpleLabel label_en =
    sdoc.createLabel(
        XBRLConstants.XBRL21_ROLE_LABEL,
        "en",
        "Cash Deposits");

// 関係を作成
SimpleLabelEdge labelEdge_ja =
    sdoc.createLabelEdge(
        XBRLConstants.XBRL21_ROLE_LINK,
        XBRLConstants.XBRL21_ARCROLE_CONCEPT_LABEL,
        item1,
        label_ja);
SimpleLabelEdge labelEdge_en =
    sdoc.createLabelEdge(
        XBRLConstants.XBRL21_ROLE_LINK,
        XBRLConstants.XBRL21_ARCROLE_CONCEPT_LABEL,
        item1,
        label_en);

// 単純なタクソノミー文書に関係を追加
sdoc.addEdge(labelEdge_en);
sdoc.addEdge(labelEdge_ja);
```

このサンプルコードでは,「4.9.2　アイテム定義の追加」で生成したアイテム定義に，次に示す日本語と英語のラベルを関連づけ，拡張したタクソノミー文書に追加します。

```
<link:labelLink xlink:role="http://www.xbrl.org/2003/role/link"
xlink:type="extended">
    <link:loc xlink:href="simple_extension_taxonomy.xsd#hi2_CashDeposits"
           xlink:label="hi2_CashDeposits" xlink:type="locator"/>
    <link:label xlink:label="hi2_CashDeposits_ja" xlink:role="http://
www.xbrl.org/2003/role/label"
           xlink:type="resource" xml:lang="ja">現預金</link:label>
    <link:label xlink:label="hi2_CashDeposits_en"
           xlink:role="http://www.xbrl.org/2003/role/label"
           xlink:type="resource" xml:lang="en">Cash Deposits</link:label>
    <link:labelArc order="1" priority="0" use="optional"
           xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"
           xlink:from="hi2_CashDeposits" xlink:to="hi2_CashDeposits_ja"
           xlink:type="arc"/>
    <link:labelArc order="1" priority="0" use="optional"
           xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"
           xlink:from="hi2_CashDeposits" xlink:to="hi2_CashDeposits_en"
           xlink:type="arc"/>
</link:labelLink>
```

4.9.4 関係の禁止

ここでは，既存の関係を禁止し，新たな関係を追加するサンプルコードについて説明します。

(1) 使用するクラスおよびインターフェース

既存の関係を禁止し，新たな関係を追加するために使用するクラスおよびインターフェースを次の表に示します。

表 4-42 既存の関係の禁止や新たな関係の追加で使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument	タクソノミー文書全体を表します。
2	com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement	タクソノミー文書に記述される要素の要素宣言を表します。
3	com.hitachi.xbrl.taxonomy.XBRLLabel	要素のラベルを表します。
4	com.hitachi.xbrl.link.XBRLLabelRelationship	ラベルリンクに記述されている関係のうち，一つの関係を表します。
5	com.hitachi.xbrl.simple.taxonomy.SimpleEdge	単純なタクソノミー文書を構成する関係ネットワークの一辺（一つの関係）を表します。
6	com.hitachi.xbrl.simple.taxonomy.SimpleLabel	単純なタクソノミー文書のラベルを表します。
7	com.hitachi.xbrl.simple.taxonomy.SimpleLabelEdge	単純なタクソノミー文書を構成するラベルリンクの一辺（一つの関係）を表します。
8	com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy	編集できる単純なタクソノミー文書全体を表します。
9	com.hitachi.xbrl.util.XBRLConstants	ユーザーアプリケーションで利用できる定数を定義します。

(2) サンプルコード

既存の関係を禁止し，新たな関係を XBRL DOM に追加するコードは次のとおりです。

```
// SimpleTaxonomyの作成時に複製されたXBRLTaxonomyDocumentを取得
XBRLTaxonomyDocument baseDoc = sdoc.getBaseTaxonomyDocument();

// 既存の関係を禁止する
XBRLTaxonomyElement item2 =
    baseDoc.getElement(
        "http://www.hitachi.com/2004/XBRL",
        "investmentsAndAdvances");
if (item2 != null) {
    List<XBRLLabelRelationship> list =
        item2.getLabelRelationships(
            XBRLConstants.XBRL21_ROLE_LINK,
            XBRLConstants.XBRL21_ARCROLE_CONCEPT_LABEL);

    for (XBRLLabelRelationship rs : list) {
        XBRLLabel label = (XBRLLabel) rs.getTo();
        if ("ja".equals(label.getLang())) {
            SimpleEdge prohibitedEdge =
                sdoc.createProhibitedEdge(rs);
            sdoc.addEdge(prohibitedEdge);
        }
    }
}

// 新たな関係を作成
SimpleLabel newLabel =
    sdoc.createLabel(
        XBRLConstants.XBRL21_ROLE_LABEL,
        "ja",
        "投資と前貸金");
SimpleLabelEdge newEdge =
    sdoc.createLabelEdge(
        XBRLConstants.XBRL21_ROLE_LINK,
        XBRLConstants.XBRL21_ARCROLE_CONCEPT_LABEL,
        item2,
        newLabel);
sdoc.addEdge(newEdge);
```

このサンプルコードでは，次に示す既存のアイテムに対し，既存の日本語ラベルとの関係を禁止して，新たな日本語ラベルとの関係を作成しています。

```
<element id="investmentsAndAdvances" name="investmentsAndAdvances"
         type="xbrli:monetaryItemType" substitutionGroup="xbrli:item"
         xbrli:periodType="instant" xbrli:balance="credit"/>
```

これによって既存の日本語ラベルとの関係が禁止されます。禁止される関係は次のとおりです。

```
<link:labelLink xlink:role="http://www.xbrl.org/2003/role/link"
xlink:type="extended">
    <link:loc
        xlink:type="locator"
        xlink:href="sample_taxonomy1_xbrl_2_1.xsd#investmentsAndAdvances"
        xlink:label="assets.investmentsAndAdvances"
        xlink:title="assets.investmentsAndAdvances"/>
    <link:label
        xlink:type="resource"
        xlink:label="label_assets.investmentsAndAdvances_ja"
        xlink:title="assets.investmentsAndAdvances_ja"
        xml:lang="ja">投資及び貸付金</link:label>
    <link:labelArc
        xlink:type="arc"
        xlink:from="assets.investmentsAndAdvances"
        xlink:to="label_assets.investmentsAndAdvances_ja"
        xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"
        xlink:title="assets.investmentsAndAdvances to
label_assets.investmentsAndAdvances_ja"/>
</link:labelLink>
```

新たに，次に示す日本語ラベルとの関係が追加されます。

```
<link:labelLink xlink:role="http://www.xbrl.org/2003/role/link"
xlink:type="extended">
   <link:loc xlink:href="sample_taxonomy1_xbrl_2_1.xsd#investmentsAndAdvances"
            xlink:label="hi_investmentsAndAdvances" xlink:type="locator"/>
    <link:loc xlink:href="linkbase_label1_xbrl_2_1.xml#element(/1/1/12)"
            xlink:label="label_assets.investmentsAndAdvances_ja"
xlink:type="locator"/>
    <link:label xlink:label="hi_investmentsAndAdvances_ja"
            xlink:role="http://www.xbrl.org/2003/role/label"
            xlink:type="resource" xml:lang="ja">投資と前貸金</link:label>
    <link:labelArc order="1" priority="1" use="prohibited"
            xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"
            xlink:from="hi_investmentsAndAdvances"
xlink:to="label_assets.investmentsAndAdvances_ja"
            xlink:type="arc"/>
    <link:labelArc order="1" priority="0" use="optional"
            xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"
            xlink:from="hi_investmentsAndAdvances"
xlink:to="hi_investmentsAndAdvances_ja"
```

4.9.5 タクソノミー文書の保存

拡張したタクソノミー文書を保存するサンプルコードについて説明します。

(1) 使用するインターフェース

タクソノミー文書の保存で使用するインターフェースを次の表に示します。

表 4-43 タクソノミー文書の保存で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.util.XBRLDocumentWriterFactory	XBRLDocumentWriter オブジェクトを生成するための API を提供します。
2	com.hitachi.xbrl.util.XBRLDocumentWriter	org.w3c.Document をストリームに書き出します。

項番	使用するインターフェース	説明
3	com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy	編集できる単純なタクソノミー文書全体を表します。

(2) サンプルコード

タクソノミー文書を保存するコードは次のとおりです。

```
// 単純なタクソノミー文書を表すSimpleTaxonomyからorg.w3c.Documentへ変換
Document[] docs = sdoc.transform();

// XBRLDocumentWriterFactoryの作成
XBRLDocumentWriterFactory wf = dm.newDocumentWriterFactory();

// XBRLDocumentWriterの作成
XBRLDocumentWriter writer = wf.newDocumentWriter();
writer.setIndent(true);

// タクソノミースキーマを保存する
FileOutputStream schemaFile =
    new FileOutputStream("output/simple_extension_taxonomy.xsd");
writer.write(schemaFile, docs[0]);
schemaFile.close();

// ラベルリンクを保存する
FileOutputStream labelFile =
    new FileOutputStream("output/simple_extension_taxonomy_label.xml");
writer.write(labelFile, docs[1]);
labelFile.close();
```

このサンプルコードは，編集できるタクソノミー文書全体を表す SimpleTaxonomy オブジェクトの transform() メソッドを使い，タクソノミー本体，ラベルリンクベースをそれぞれに対応する org.w3c.dom.Document オブジェクトへ変換します。変換後の Document オブジェクトは，XML 形式のファイルに次の名称で保存します。

- タクソノミー本体のファイル名
  「4.9.1　タクソノミー文書の拡張」で SimpleTaxonomy オブジェクトを生成したときに指定したファイル名で保存してください。
  なお，このサンプルコードでは，ファイル名は次のように設定されています。
  `simple_extension_taxonomy.xsd`
- ラベルリンクベースのファイル名
  ラベルリンクベースファイルの命名規則に従ったファイル名で保存してください。命名規則については，「3.1.1(3)　関係を追加・削除した場合に生じる影響」を参照してください。
  なお，このサンプルコードでは，ファイル名は次のように設定されています。
  `simple_extension_taxonomy_label.xml`

4.10　タクソノミー文書の参照と検証（XBRL Dimensions1.0 の場合）

ここでは，タクソノミー文書とリンクベースの参照と検証をします。

ここで示すサンプルは，次の流れでコーディングしています。

1. **タクソノミー文書の読み込み**
2. **タクソノミー本体とリンクベースの参照**

ここでは，XBRL Dimensions1.0 の固有の操作である「4.10.1　タクソノミー文書の読み込み」および「4.10.2　タクソノミー本体とリンクベースの参照（DRS の情報を参照する場合）」について説明しています。XBRL2.1 と共通の操作については，「4.6.2　タクソノミー本体とリンクベースの参照（アイテム名またはタプル名を参照する場合）」および「4.6.3　タクソノミー本体とリンクベースの参照（タクソノミー文書の要素の階層を参照する場合）」を参照してください。

4.10.1　タクソノミー文書の読み込み

XBRL Dimensions1.0 に準拠したタクソノミー文書を読み込み，XBRL DOM を生成するサンプルコードについて説明します。XBRL Dimensions1.0 の場合，タクソノミー文書の読み込み時に XBRL Dimensions のバージョンを指定する必要があります。

ここでは，検証時にエラーや警告が発生した場合のレポートを受け取り，出力するエラーハンドラと，検証時にローカルに保存したスキーマ定義を使用するエンティティリゾルバも実装します。エラーハンドラおよびエンティティリゾルバの詳細については，「4.13　エラーハンドラ」および「4.14　エンティティリゾルバ」を参照してください。

(1) 使用するクラスおよびインターフェース

タクソノミー文書の読み込みで使用するクラスおよびインターフェースを次の表に示します。

表 4-44　タクソノミー文書の読み込みで使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.XBRLDocumentManager	XBRL 文書の入出力や，タクソノミー文書のキャッシュを管理します。
2	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory	XBRLTaxonomyDocumentBuilder オブジェクトを生成するための API を定義します。

項番	使用するクラスおよびインターフェース	説明
3	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder	タクソノミー文書からXBRLTaxonomyDocumentオブジェクトを取得するためのAPIを定義します。
4	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument	タクソノミー文書全体を表します。
5	com.hitachi.xbrl.util.XBRLConstants	ユーザーアプリケーションで利用できる定数を定義します。

(2) サンプルコード

タクソノミー文書を読み込むコードは次のとおりです。

```
/*
 * タクソノミー文書を読み込む
 */

// 初期化
XBRLDocumentManager.init();

// XBRLDocumentManagerの取得
XBRLDocumentManager dm = XBRLDocumentManager.getInstance();

// XBRLTaxonomyDocumentBuilderFactoryの作成
XBRLTaxonomyDocumentBuilderFactory tbf =
    dm.newTaxonomyDocumentBuilderFactory();

// XBRLバージョンの設定
tbf.setXBRLVersion(XBRLConstants.XBRL_2_1);
tbf.setDemensionsVersion(XBRLConstants.DIMENSIONS_1_0);

// タクソノミー文書のパース時に妥当性検証を行う設定
tbf.setValidating(true);

// XBRLTaxonomyDocumentBuilderの作成
XBRLTaxonomyDocumentBuilder tb = tbf.newTaxonomyDocumentBuilder();

// エラーハンドラの設定
SampleErrorHandler eh = new SampleErrorHandler();
tb.setErrorHandler(eh);

// エンティティリゾルバの設定
SampleEntityResolver er = new SampleEntityResolver();
tb.setEntityResolver(er);

// タクソノミー文書のパース
String filename = "input/sample_xbrl_dim_taxonomy.xsd";
File[] taxonomy = { new File(filename) };
XBRLTaxonomyDocument tdoc = tb.parse(taxonomy);

System.out.println("タクソノミー文書のパースに成功しました。");
```

4.10.2 タクソノミー本体とリンクベースの参照（DRSの情報を参照する場合）

XBRL Dimensions1.0に準拠したタクソノミー文書の読み込み時に生成したXBRL DOMから，DRSの情報を参照するサンプルコードについて説明します。

このサンプルコードでは，リンクの関係を表すインターフェースの一つである，XBRLDimensionalRelationshipSet インターフェースを使用しています。

(1) 使用するインターフェース

タクソノミー本体とリンクベースの参照で使用するインターフェースを次の表に示します。

表 4-45 タクソノミー本体とリンクベースの参照で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument	タクソノミー文書全体を表します。
2	com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet	プライマリー・アイテムに関連づいている DRS を表します。
3	com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement	タクソノミー文書に記述される要素の要素宣言を表します。
4	com.hitachi.xbrl.link.XBRLDRSDimension	インスタンス文書に記述できるディメンジョンを表します。
5	com.hitachi.xbrl.link.XBRLDRSMember	インスタンス文書に記述できるディメンジョンのメンバを表します。
6	com.hitachi.xbrl.util.XBRLConstants	ユーザーアプリケーションで利用できる定数を定義します。

(2) サンプルコード

タクソノミー本体とリンクベースを参照するコードは次のとおりです。

```
// タクソノミー文書のパース
String filename = "input/sample_xbrl_dim_taxonomy.xsd";
File[] taxonomy = { new File(filename) };
XBRLTaxonomyDocument tdoc = tb.parse(taxonomy);

System.out.println("タクソノミー文書のパースに成功しました。");

/*
 * ここからタクソノミー文書の内容を参照
 */

System.out.println("読み込んだDRS情報を出力します。");

// すべてのDRSの取得
List<XBRLDimensionalRelationshipSet> drssList =
    tdoc.getDimensionalRelationshipSets();
if (drssList.size() > 0) {

    for (int i = 0; i < drssList.size(); i++) {
        System.out.println("--------------------");
        XBRLDimensionalRelationshipSet drss = drssList.get(i);

        // プライマリー・アイテムを取得
        XBRLTaxonomyItem item = drss.getTaxonomyItem();
        System.out.println(
                "(" + i + ") primary item = "
                + item.getName());

        /*
         * このアイテムに対するディメンジョンの一覧を取得する
         */

        // DRSのロールの一覧を取得する
        List<String> roleList = drss.getRoles();
        for (int j = 0; j < roleList.size(); j++) {
            // DRSのロールを出力
            String extRole = roleList.get(j);
            System.out.println(
                    "  + (" + j + ") role = "
                    + extRole);

            // このロールのディメンジョン一覧を取得する
            List<XBRLDRSDimension> dimList =
                drss.getDRSDimensions(extRole);
            for (int n = 0; n < dimList.size(); n++) {
                // ロールのディメンジョンを出力
                XBRLDRSDimension dim = dimList.get(n);
                System.out.println(
                        "    + (" + n + ") dimension = "
                        + dim.getQName());

                // このディメンジョンのメンバを取得する
                List<XBRLDRSMember> memList =
                    drss.getDRSMembers(dim);
                for (int m = 0; m < memList.size(); m++) {
                    // ディメンジョンのメンバを出力
                    XBRLDRSMember mem = memList.get(m);
                    System.out.println(
                            "       + (" + m + ") member = "
                            + mem.getQName());
                }
            }
        }
        System.out.println("");
    }
}
```

(3) サンプルコードの実行結果

サンプルのタクソノミー文書を参照したときの実行結果を次に示します。

```
タクソノミー文書のパースに成功しました。
読み込んだDRS情報を出力します。
--------------------
(0) primary item = NumbersOfSales
  + (0) role = http://www.xbrl.org/2003/role/link
    + (0) dimension = {http://www.hitachi.com/2008/XBRL/dim/sample}DistDim
       + (0) member = {http://www.hitachi.com/2008/XBRL/dim/sample}Tokyo
       + (1) member = {http://www.hitachi.com/2008/XBRL/dim/sample}Kanagawa
       + (2) member = {http://www.hitachi.com/2008/XBRL/dim/sample}Osaka
       + (3) member = {http://www.hitachi.com/2008/XBRL/dim/sample}Kyoto
    + (1) dimension = {http://www.hitachi.com/2008/XBRL/dim/sample}ProductDim
       + (0) member = {http://www.hitachi.com/2008/XBRL/dim/sample}LC
       + (1) member = {http://www.hitachi.com/2008/XBRL/dim/sample}Plasma
```

4.11　インスタンス文書の新規作成と保存（XBRL Dimensions1.0 の場合）

ここでは，uCosminexus Business Reporting Processor の API を利用して，XBRL Dimensions1.0 に準拠したインスタンス文書を新規に作成して保存します。

ここで示すサンプルは，次の流れでコーディングしています。

1. **多次元データを含んだコンテキスト要素の追加**
2. **プライマリー・アイテムの追加**

ここでは，XBRL Dimensions1.0 の固有の操作である「4.11.1　多次元データを含んだコンテキスト要素の追加」および「4.11.2　プライマリー・アイテムの追加」について説明しています。XBRL2.1 と共通の操作については，「4.7　インスタンス文書の新規作成と保存（XBRL2.1 の場合）」を参照してください。

XBRL Dimensions1.0 の場合，インスタンス文書の id 属性を任意の形式で指定することもできます。インスタンス文書の id 属性を任意の形式で指定しない場合，通常は id 属性が自動生成されます。しかし，このサンプルではコーディングをわかりやすくするために，id 属性などの値を仮定しています。

4.11.1　多次元データを含んだコンテキスト要素の追加

XBRL Dimensions1.0 に準拠したインスタンス文書の XBRL DOM に多次元データを含んだコンテキスト要素を追加するサンプルコードについて説明します。

ここでは，<xbrldi:explicitMember> 要素を追加するサンプルコードについて説明します。なお，<xbrldi:typedMember> 要素を追加する場合は，<xbrldi:explicitMember> 要素を <xbrldi:typedMember> 要素に読み替えてください。

ここで説明するサンプルコードでは，次の内容のコンテキスト要素を追加します。表中の「名称」欄のインデントは，コーディング結果のインデントを示しています。

表 4-46　多次元データを含んだコンテキスト要素

項番	名称	要素 / 属性	値
1	コンテキスト ID	属性	C01 および C02
2	時期	要素	2008-09-30
3	エンティティ (組織)	要素	－
4	識別子	要素	SAMP
5	スキーム	属性	"http://www.hitachi.com/2008/XBRL"
6	セグメント	要素	－

項番	名称	要素 / 属性	値
7	explicitMember dimension	要素	p0:Tokyo，p0:LC および p0:Plasma ※

（凡例）－：値を保持しません。
注※　プリフィックス p01 の名前空間 URI は "http://www.hitachi.com/2008/XBRL/dim/sample" とします。なお，プリフィックスと名前空間 URI の組み合わせは，あらかじめ com.hitachi.xbrl.util.XBRLNamespaceDeclaration クラスで作成し，インスタンス文書に追加してください。

(1) 使用するクラスおよびインターフェース

多次元データを含んだコンテキスト要素の追加で使用するクラスおよびインターフェースを次の表に示します。

表 4-47　多次元データを含んだコンテキスト要素の追加で使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。
2	com.hitachi.xbrl.types.XBRLQName	XML Schema のデータ型 QName を表します。
3	com.hitachi.xbrl.instance.XBRLSegment	セグメント要素を表します。
4	com.hitachi.xbrl.instance.XBRLExplicitMember	<xbrldi:explicitMember> 要素を表します。
5	com.hitachi.xbrl.instance.XBRLIdentifier	識別子要素を表します。
6	com.hitachi.xbrl.instance.XBRLEntity	エンティティ要素を表します。
7	com.hitachi.xbrl.types.XBRLDateUnion	XBRL のデータ型 dateUnion を表します。
8	com.hitachi.xbrl.instance.XBRLPeriod	時期要素を表します。
9	com.hitachi.xbrl.instance.XBRLContext2	XBRL2.1 のコンテキスト要素を表します。
10	com.hitachi.xbrl.instance.XBRLContext	コンテキスト要素を表します。

(2) サンプルコード

多次元データを含んだコンテキスト要素を XBRL DOM に追加するコードは次のとおりです。

```
        /*
         * ここからアイテム（売上高）の作成
         */
        XBRLQName LC = new XBRLQName("http://www.hitachi.com/2008/XBRL/dim/
sample", "LC");
       XBRLQName plasma = new XBRLQName("http://www.hitachi.com/2008/XBRL/dim/
sample", "Plasma");

        // 東京の売上
        XBRLQName tokyo = new XBRLQName("http://www.hitachi.com/2008/XBRL/dim/
sample", "Tokyo");
        BigDecimal salesLC = new BigDecimal("10000000");
        createItem(idoc, unit, tokyo, LC, salesLC);

        BigDecimal salesPlasma = new BigDecimal("8000000");
        createItem(idoc, unit, tokyo, plasma, salesPlasma);
                    :
                    :
                    :

private static void createItem(
        XBRLInstanceDocument idoc,
        XBRLUnit2 unit,
        XBRLQName distValue,
        XBRLQName prodValue,
        BigDecimal salesValue) {

    /*
     * ここからコンテキストの作成
     */

    // segment要素の作成
    XBRLSegment segment = idoc.createSegment();

    // explicitMember要素の作成 - DistDim
    XBRLQName distDim = new XBRLQName("http://www.hitachi.com/2008/XBRL/dim/
sample", "DistDim");
    XBRLExplicitMember distExplicit = idoc.createExplicitMember(distDim,
distValue);
    segment.appendDimensionMember(distExplicit);

    // explicitMember要素の作成 - ProductDim
    XBRLQName prodDim = new XBRLQName("http://www.hitachi.com/2008/XBRL/dim/
sample", "ProductDim");
    XBRLExplicitMember prodExplicit = idoc.createExplicitMember(prodDim,
prodValue);
    segment.appendDimensionMember(prodExplicit);

    // identifier要素の作成
    XBRLIdentifier identifier =
        idoc.createIdentifier(
            "http://www.hitachi.com/2008/XBRL",
            "SAMP");

    // entity要素の作成
    XBRLEntity entity = idoc.createEntity(identifier, segment);

    // period要素の作成
    XBRLDateUnion date = new XBRLDateUnion("2008-09-30");
    XBRLPeriod period = idoc.createPeriod(date);

    // コンテキストの作成
    XBRLContext2 context = idoc.createContext(entity, period, null);
    // インスタンス文書にコンテキストを追加
    idoc.addContext((XBRLContext) context);
```

(3) コーディング結果

コーディング結果として，生成されるインスタンス文書は次のとおりです。

```
<xbrli:context id="C01">
    <xbrli:entity>
     <xbrli:identifier
       schema="http://www.hitachi.com/2008/XBRL">SAMP</xbrli:identifier>
     <xbrli:segment>
       <xbrldi:explicitMember dimension="p0:DistDim">p0:Tokyo</
xbrldi:explicitMember>
       <xbrldi:explicitMember dimension="p0:ProductDim">p0:LC</
xbrldi:explicitMember>
     </xbrli:segment>
    </xbrli:entity>
    <xbrli:period>
       <xbrli:instant>2008-09-30</xbrli:instant>
    </xbrli:period>
</xbrli:context>

<xbrli:context id="C02">
    <xbrli:entity>
     <xbrli:identifier
       schema="http://www.hitachi.com/2008/XBRL">SAMP</xbrli:identifier>
     <xbrli:segment>
       <xbrldi:explicitMember dimension="p0:DistDim">p0:Tokyo</
xbrldi:explicitMember>
       <xbrldi:explicitMember dimension="p0:ProductDim">p0:Plasma</
xbrldi:explicitMember>
     </xbrli:segment>
    </xbrli:entity>
    <xbrli:period>
       <xbrli:instant>2008-09-30</xbrli:instant>
    </xbrli:period>
</xbrli:context>
```

4.11.2　プライマリー・アイテムの追加

XBRL Dimensions1.0 に準拠したインスタンス文書の XBRL DOM にプライマリー・アイテムを追加するサンプルコードについて説明します。

ここでは，プライマリー・アイテムとディメンジョンの関係を検証してからプライマリー・アイテムをインスタンス文書へ追加するサンプルコードについて説明します。インスタンス文書全体の検証については，「4.12　既存のインスタンス文書の参照と検証（XBRL Dimensions1.0 の場合）」を参照してください。

なお，「4.11.2(3)　コーディング結果」で示すコーディング結果のコンテキスト要素には「4.11.1　多次元データを含んだコンテキスト要素の追加」で追加したコンテキストが記述されています。

(1) 使用するクラスおよびインターフェース

プライマリー・アイテムの追加で使用するクラスおよびインターフェースを次の表に示します。

表 4-48　プライマリー・アイテムの追加で使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.instance.XBRLContext2	XBRL2.1 のコンテキスト要素を表します。
2	com.hitachi.xbrl.instance.XBRLContext	コンテキスト要素を表します。
3	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。
4	com.hitachi.xbrl.types.XBRLDecimals	数値アイテムの decimals 属性値を表します。
5	com.hitachi.xbrl.instance.XBRLInstanceItem	アイテムを表します。
6	com.hitachi.xbrl.instance.XBRLInstanceElement	インスタンス文書の要素を表します。
7	com.hitachi.xbrl.instance.XBRLDOMException	XBRL DOM の例外を表します。

(2) サンプルコード

プライマリー・アイテムを XBRL DOM に追加するコードは次のとおりです。

```
        // コンテキストの作成
        XBRLContext2 context = idoc.createContext(entity, period, null);
        // インスタンス文書にコンテキストを追加
        idoc.addContext((XBRLContext) context);

        /*
         * ここから数値アイテムの作成
         */

        // decimals属性値の作成
        XBRLDecimals decimals = new XBRLDecimals("0");

        // 数値アイテム - NumbersOfSales要素 - の作成
        XBRLInstanceItem salesItem =
            idoc.createNumericItem(
                "http://www.hitachi.com/2008/XBRL/dim/sample",
                "NumbersOfSales",
                context,
                unit,
                null,
                decimals);

        // 数値アイテムに値を設定
        salesItem.setValue((Object) salesValue);

        try {
            // 妥当性検証
            if (salesItem.validateDimensions()) {
                // インスタンス文書に数値アイテムを追加
                idoc.appendChildElement((XBRLInstanceElement) salesItem);
            } else {
                System.out.println("アイテムとコンテキストの組み合わせは正しくありませ
ん。");
            }

        } catch (XBRLDOMException de) {
            de.printStackTrace();
        }

    }
}
```

(3) コーディング結果

コーディング結果として，生成されるインスタンス文書は次のとおりです。

```
<xbrli:unit id="U01">
  <xbrli:measure>iso4217:JPY</xbrli:measure>
</xbrli:unit>

<xbrli:context id="C01">
  <xbrli:entity>
    <xbrli:identifier
        schema="http://www.hitachi.com/2008/XBRL">SAMP</xbrli:identifier>
    <xbrli:segment>
      <xbrldi:explicitMember dimension="p0:DistDim">p0:Tokyo</
xbrldi:explicitMember>
      <xbrldi:explicitMember dimension="p0:ProductDim">p0:LC</
xbrldi:explicitMember>
    </xbrli:segment>
  </xbrli:entity>
  <xbrli:period>
    <xbrli:instant>2008-09-30</xbrli:instant>
  </xbrli:period>
</xbrli:context>

<xbrli:context id="C02">
  <xbrli:entity>
    <xbrli:identifier
        schema="http://www.hitachi.com/2008/XBRL">SAMP</xbrli:identifier>
    <xbrli:segment>
      <xbrldi:explicitMember dimension="p0:DistDim">p0:Tokyo</
xbrldi:explicitMember>
      <xbrldi:explicitMember dimension="p0:ProductDim">p0:Plasma</
xbrldi:explicitMember>
    </xbrli:segment>
  </xbrli:entity>
  <xbrli:period>
    <xbrli:instant>2008-09-30</xbrli:instant>
  </xbrli:period>
</xbrli:context>

<p0:NumbersOfSales contextRef="C01" unitRef="U01" decimals="0">10000000</
p0:NumbersOfSales>
<p0:NumbersOfSales contextRef="C02" unitRef="U01" decimals="0">8000000</
p0:NumbersOfSales>
```

4.12　既存のインスタンス文書の参照と検証（XBRL Dimensions1.0 の場合）

ここでは，既存のインスタンス文書とリンクベースの参照と検証をします。

ここで示すサンプルは，次の流れでコーディングしています。

1. **インスタンス文書の読み込み**
2. **インスタンス文書とリンクベースの参照**

4.12.1　インスタンス文書の読み込み

XBRL Dimensions1.0 に準拠したタクソノミー文書を読み込み，XBRL DOM を生成するサンプルコードについて説明します。XBRL Dimensions1.0 の場合，タクソノミー文書の読み込み時に XBRL Dimensions のバージョンを指定する必要があります。

ここでは，検証時にエラーや警告が発生した場合のレポートを受け取り，出力するエラーハンドラと，検証時にローカルに保存したスキーマ定義を使用するエンティティリゾルバも実装します。エラーハンドラおよびエンティティリゾルバの詳細については，「4.13　エラーハンドラ」および「4.14　エンティティリゾルバ」を参照してください。

（1）使用するクラスおよびインターフェース

インスタンス文書の読み込みで使用するクラスおよびインターフェースを次の表に示します。

表 4-49　インスタンス文書の読み込みで使用するクラスおよびインターフェース

項番	使用するクラスおよびインターフェース	説明
1	com.hitachi.xbrl.XBRLDocumentManager	XBRL 文書の入出力や，タクソノミー文書のキャッシュを管理します。
2	com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory	XBRLInstanceDocumentBuilder オブジェクトを生成するための API を定義します。
3	com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder	インスタンス文書から XBRLInstanceDocument オブジェクトを取得するための API を定義します。
4	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。
5	com.hitachi.xbrl.util.XBRLConstants	ユーザーアプリケーションで利用できる定数を定義します。

（2）サンプルコード

インスタンス文書を読み込み，XBRL DOM を生成するコードは次のとおりです。

```
/*
 * 既存インスタンス文書の読み込み
 */

// 初期化
XBRLDocumentManager.init();
XBRLDocumentManager dm = XBRLDocumentManager.getInstance();

// XBRLInstanceDocumentBuilderFactoryの作成
XBRLInstanceDocumentBuilderFactory ibf =
    dm.newInstanceDocumentBuilderFactory();

// XBRLバージョンの設定
ibf.setXBRLVersion(XBRLConstants.XBRL_2_1);
ibf.setDimensionsVersion(XBRLConstants.DIMENSIONS_1_0);

// インスタンス文書のパース時に妥当性検証を行う設定
ibf.setValidating(true);

// XBRLInstanceDocumentBuilderの作成
XBRLInstanceDocumentBuilder ib = ibf.newInstanceDocumentBuilder();

// エラーハンドラの設定
SampleErrorHandler eh = new SampleErrorHandler();
ib.setErrorHandler(eh);

// エンティティリゾルバの設定
SampleEntityResolver er = new SampleEntityResolver();
ib.setEntityResolver(er);

// タクソノミー文書のパース
String filename = "input/sample_xbrl_dim_instance.xml";
XBRLInstanceDocument idoc = ib.parse(new File(filename));

System.out.println("インスタンス文書のパースに成功しました。");
```

4.12.2　インスタンス文書とリンクベースの参照（プライマリー・アイテムのディメンジョンとアイテムの値を参照する場合）

XBRL Dimensions1.0 に準拠したタクソノミー文書の読み込み時に生成した XBRL DOM から，プライマリー・アイテムのディメンジョンとアイテムの情報を参照するサンプルコードについて説明します。プライマリー・アイテムおよびディメンジョンの表示名には，ラベルを使用します。

（1）使用するインターフェース

インスタンス文書とリンクベースの参照で使用するインターフェースを次の表に示します。

表 4-50　インスタンス文書とリンクベースの参照で使用するインターフェース

項番	使用するインターフェース	説明
1	com.hitachi.xbrl.instance.XBRLInstanceDocument	インスタンス文書全体を表します。
2	com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument	タクソノミー文書全体を表します。

項番	使用するインターフェース	説明
3	com.hitachi.xbrl.instance.XBRLInstanceElement	インスタンス文書に記述される要素を表します。
4	com.hitachi.xbrl.instance.XBRLInstanceItem	アイテムを表します。
5	com.hitachi.xbrl.instance.XBRLContext	コンテキスト要素を表します。
6	com.hitachi.xbrl.instance.XBRLEntity	エンティティ要素を表します。
7	com.hitachi.xbrl.instance.XBRLSegment	セグメント要素を表します。
8	com.hitachi.xbrl.instance.XBRLExplicitMember	xbrldi:explicitDimension 要素を表します。
9	com.hitachi.xbrl.taxonomy.XBRLLabel	要素のラベルを表します。
10	com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement	タクソノミー文書に記述される要素の要素宣言を表します。
11	com.hitachi.xbrl.types.XBRLQName	XML Schema のデータ型 QName を表します。

(2) サンプルコード

インスタンス文書とリンクベースを参照するコードは次のとおりです。

```
/*
 * ここから内容を表示する
 */
// タクソノミー文書を取得
XBRLTaxonomyDocument tdoc = idoc.getTaxonomyDocument();

// 売上高の一覧を取得する
List<XBRLInstanceElement> elemList =
    idoc.getElements(
            "http://www.hitachi.com/2008/XBRL/dim/sample",
            "NumbersOfSales");
for (int i = 0; i < elemList.size(); i++) {

    System.out.print("(" + i + ")");

    XBRLInstanceItem item = (XBRLInstanceItem) elemList.get(i);

    // コンテキスト要素を取得する
    XBRLContext context = item.getContext();

    // エンティティ要素を取得する
    XBRLEntity entity = context.getEntity();

    // セグメント要素を取得する
    XBRLSegment segment = entity.getSegment();

    // explicitMember要素の一覧を取得する
    List<XBRLExplicitMember> expMemList =
        segment.getExplicitMembers();
    for (XBRLExplicitMember expMem : expMemList) {

        // dimension属性値からディメンジョンを求める
        XBRLQName dimension = expMem.getDimension();
        XBRLTaxonomyElement dimElem =
            tdoc.getElement(
                    dimension.getNamespaceURI(),
                    dimension.getLocalPart());

        // ディメンジョンの日本語ラベルを取得する
        XBRLLabel dimLabel = dimElem.getLabel("ja", null);

        // explicitMember要素の値からメンバを求める
        XBRLQName member = expMem.getValue();
        XBRLTaxonomyElement memElem =
            tdoc.getElement(
                    member.getNamespaceURI(),
                    member.getLocalPart());

        // メンバの日本語ラベルを取得する
        XBRLLabel memLabel = memElem.getLabel("ja", null);

        // ディメンジョンとそのメンバを表示
        System.out.print(
                " " + dimLabel.getValue() +
                ":" + memLabel.getValue() + ",");
    }
    // プライマリー・アイテムの日本語ラベルを取得する
    XBRLLabel label = item.getLabel("ja", null);

    // アイテムとその値を表示
    System.out.println(
            " " + label.getValue() +
            ":" + item.getValue());
}
```

(3) サンプルコードの実行結果

サンプルのインスタンス文書を参照したときの実行結果を次に示します。

```
インスタンス文書のパースに成功しました。
(0) 地域:東京, 製品:液晶, 売上高:10000000
(1) 地域:東京, 製品:プラズマ, 売上高:8000000
(2) 地域:神奈川, 製品:液晶, 売上高:8000000
(3) 地域:神奈川, 製品:プラズマ, 売上高:5000000
(4) 地域:大阪, 製品:液晶, 売上高:9000000
(5) 地域:大阪, 製品:プラズマ, 売上高:8000000
(6) 地域:京都, 製品:液晶, 売上高:7500000
(7) 地域:京都, 製品:プラズマ, 売上高:5500000
```

4.13　エラーハンドラ

エラーハンドラは，XBRL 文書の解析処理で検出されたエラーの通知を受け取る場合に使用します。

エラーハンドラは，XBRLParseErrorHandler を実装して作成します。エラーハンドラは，XBRL DOM の生成時に XBRLDOMException が発生するとそのレポートを受け取り，エラーコードやメッセージなどのレポートの内容をコンソールに出力します。

サンプルコード（SampleErrorHandler.java）は，各サンプルファイルが格納されているフォルダにあります。

<uCosminexus Business Reporting Processor のインストールフォルダ>¥samples¥<XBRL バージョン >¥< サンプル名 >

サンプルファイルが格納されているフォルダの詳細は，「4.15.1　サンプルファイルの格納場所」を参照してください。

エラーハンドラのサンプルコードは次のとおりです。

```
import com.hitachi.xbrl.XBRLDOMException;
import com.hitachi.xbrl.XBRLParseErrorHandler;

public class SampleErrorHandler implements XBRLParseErrorHandler {

    // 警告の通知を受け取る
    public void warning(XBRLDOMException exception) throws XBRLDOMException {
        System.out.println(
            "Warning - code:" + exception.getCode() + " Message:" +
exception.getMessage());
    }

    // エラーの通知を受け取る
    public void error(XBRLDOMException exception) throws XBRLDOMException {
        System.out.println(
            "Error   - code:" + exception.getCode() + " Message:" +
exception.getMessage());
    }

    // 致命的なエラーの通知を受け取る
   public void fatalError(XBRLDOMException exception) throws XBRLDOMException {
        System.out.println(
            "Fatal Error - code:" + exception.getCode() + " Message:" +
exception.getMessage());
    }

}
```

4.14　エンティティリゾルバ

エンティティリゾルバは，直接インターネット上のスキーマ定義にアクセスできない場合や，スキーマ定義をあらかじめローカルに保存して検証する場合に使用します。

4.14.1　サンプルコード

ここでは，エンティティリゾルバのサンプルコードを示します。

このサンプルコードは，XBRL および XML Schema のスキーマをファイルとしてローカルに保存し，解析時にローカルにあるファイルを利用するための簡単なエンティティリゾルバの実装例です。

サンプルコード（SampleEntityResolver.java）は，各サンプルファイルが格納されているフォルダにあります。

<uCosminexus Business Reporting Processor のインストールフォルダ>¥samples¥<XBRL バージョン >¥< サンプル名 >

サンプルファイルが格納されているフォルダの詳細は，「4.15.1　サンプルファイルの格納場所」を参照してください。

エンティティリゾルバのサンプルコードは次のとおりです。

```
import java.io.File;
import java.io.IOException;
import java.net.URI;
import java.util.HashMap;
import java.util.Map;

import org.xml.sax.EntityResolver;
import org.xml.sax.InputSource;

public class SampleEntityResolver implements EntityResolver {

    // スキーマのURI
    private static final String XBRL20_INSTANCE_URI =
        "http://www.xbrl.org/2001/xbrl-instance.xsd";
    private static final String XBRL20_LINKBASE_URI =

        "http://www.xbrl.org/2001/xbrl-linkbase.xsd";
    private static final String XBRL21_INSTANCE_URI =
        "http://www.xbrl.org/2003/xbrl-instance-2003-12-31.xsd";
    private static final String XBRL21_LINKBASE_URI =
        "http://www.xbrl.org/2003/xbrl-linkbase-2003-12-31.xsd";
    private static final String XML_SCHEMA_URI =
        "http://www.w3.org/2001/XMLSchema.xsd";
    private static final String XML_URI =
        "http://www.w3.org/2001/xml.xsd";
    private static final String XBRL_DT_URI =
        "http://xbrl.org/2005/xbrldt-2005.xsd";
    private static final String XBRL_DI_URI =
        "http://xbrl.org/2006/xbrldi-2006.xsd";
```

```
    // ファイルセパレータ
    private static String SEP = File.separator;

    // ファイルの実体に付加するパス
    private static String basePath =
        System.getProperty("user.dir") + SEP + ".." + SEP + "schemas";

    // URIと実体のマッピング
    private static final Map<String, String> HTTPMAP =
        new HashMap<String, String>();

    // 静的初期化子
    static {
        HTTPMAP.put(XBRL20_INSTANCE_URI, "xbrl-instance.xsd");
        HTTPMAP.put(XBRL20_LINKBASE_URI, "xbrl-linkbase.xsd");
        HTTPMAP.put(XBRL21_INSTANCE_URI, "xbrl-instance-2003-12-31.xsd");
        HTTPMAP.put(XBRL21_LINKBASE_URI, "xbrl-linkbase-2003-12-31.xsd");
        HTTPMAP.put(XML_SCHEMA_URI,      "XMLSchema.xsd");
        HTTPMAP.put(XML_URI,             "xml.xsd");
        HTTPMAP.put(XBRL_DT_URI,         "xbrldt-2005.xsd");
        HTTPMAP.put(XBRL_DI_URI,         "xbrldi-2006.xsd");
    };



    /**
     * EntityResolverで定義されたインターフェースの実装です
     * 特定のURIの場合，ローカルファイル名の新しい入力ソースを作成します
     * 特定のURIとは，このオブジェクトのHTTPMAP変数で宣言された値です
     * その他のURIの場合は，nullを返します
     *
     * @param publicId 参照される外部エンティティの公開識別子
     * @param systemId  参照される外部エンティティのシステム識別子
     * @return 新しい入力ソースを記述する InputSource オブジェクトを返す
     *         systemIdに対応するローカルファイル名がない場合は null を返す
     */
    public InputSource resolveEntity(String publicId, String systemId) {


        InputSource is = null;

        if (systemId != null) {
            // ファイル名があればInputSourceを作成して返す。
            String filename = HTTPMAP.get(systemId);
            if (filename != null) {
                // ファイル名を絶対URIにする。
                File file = new File(basePath + SEP + filename);
                try {
                    file = file.getCanonicalFile();
                    URI uri = file.toURI();
                    is = new InputSource(uri.toString());
                } catch (IOException e) {
                    System.err.println(e.getMessage());
                }
            }
        }

        return is;
    }
}
```

4.15 サンプルアプリケーションの実行

ここでは，サンプルアプリケーションの実行方法について説明します。

ここで使用するサンプルは，コマンドラインの Java アプリケーションです。このサンプルアプリケーションでは，事前に定義されたタクソノミー文書を使用して，インスタンス文書を作成します。

サンプルファイルが格納されているフォルダ名，対応する XBRL のバージョン，サンプルソースが実行する内容，および参照先を次の表に示します。

表 4-51 サンプルアプリケーションの詳細

項番	フォルダ名	XBRL のバージョン	XBRL Dimensions のバージョン	実行する内容	参照先
1	ProcessTaxonomy	2.0	該当しない	タクソノミー文書を読み込みます。	4.3
2	ProcessInstance1			インスタンス文書を新規に作成して，要素を追加します。	4.4
3	ProcessInstance2			既存のインスタンス文書に脚注リンクを追加します。	4.5
4	ProcessTaxonomy	2.1	該当しない	タクソノミー文書を読み込みます。	4.6.1 および 4.6.2
5	ProcessInstance1			インスタンス文書を新規に作成して，要素を追加します。	4.7.1 ～ 4.7.6
6	ProcessInstance1WithMyId			インスタンス文書を新規に作成して，要素を追加します。 ユーザーが指定した ID 採番方法を使用します。	4.7.1 ～ 4.7.7
7	ProcessInstance2			既存のインスタンス文書に脚注リンクを追加します。	4.8
8	ProcessDimTaxonomy	2.1	1.0	タクソノミー文書を読み込みます。	4.10
9	ProcessDimTaxonomy1			インスタンス文書を新規に作成して，要素を追加します。	4.11
10	ProcessDimTaxonomy2			既存のインスタンス文書を読み込みます。	4.12

4.15.1 サンプルファイルの格納場所

サンプルファイルが格納されているフォルダは，XBRL のバージョンによって異なりま

す。

XBRL2.0 の場合

<uCosminexus Business Reporting Processor のインストールフォルダ>¥samples¥2_0

XBRL2.1 の場合

<uCosminexus Business Reporting Processor のインストールフォルダ>¥samples¥2_1

XBRL Dimensions1.0 の場合

<uCosminexus Business Reporting Processor のインストールフォルダ>¥samples¥dim

uCosminexus Business Reporting Processor のインストールフォルダは，デフォルトでは「C:¥Program Files¥HITACHI¥cbrp」になります。

サンプルファイルのフォルダ構成を次に示します。

```
<uCosminexus Business Reporting Processorのインストールフォルダ>
└─samples
    │
    ├───2_0-------------------------XBRL2.0のサンプルファイル格納用
    │   │
    │   ├─schemas------------------スキーマ定義ファイル格納用
    │   │
    │   ├─ProcessInstance1---------インスタンス文書新規作成用サンプルソース格納用
    │   │       ├─input----------入力ファイル格納用
    │   │       └─output---------サンプル実行後の出力ファイル格納用
    │   │
    │   ├─ProcessInstance2---------インスタンス文書編集用サンプルソース格納用
    │   │       ├─input----------入力ファイル格納用
    │   │       └─output---------サンプル実行後の出力ファイル格納用
    │   │
    │   └─ProcessTaxonomy----------タクソノミー文書操作用サンプルソース格納用
    │           └─input----------入力ファイル格納用
    │
    ├───2_1-------------------------XBRL2.1のサンプルファイル格納用
    │   │
    │   ├─schemas------------------スキーマ定義ファイル格納用
    │   │
    │   ├─ProcessInstance1---------インスタンス文書新規作成用サンプルソース格納用
    │   │       ├─input----------入力ファイル格納用
    │   │       └─output---------サンプル実行後の出力ファイル格納用
    │   │
    │   ├─ProcessInstance1WithMyId-インスタンス文書新規作成用サンプルソース格納用
    │   │       │                                  (ユーザー指定のID採番を使用)
    │   │       ├─input----------入力ファイル格納用
    │   │       └─output---------サンプル実行後の出力ファイル格納用
    │   │
    │   ├─ProcessInstance2---------インスタンス文書編集用サンプルソース格納用
    │   │       ├─input----------入力ファイル格納用
    │   │       └─output---------サンプル実行後の出力ファイル格納用
    │   │
    │   ├─ProcessTaxonomy----------タクソノミー文書操作用サンプルソース格納用
    │   │       └─input----------入力ファイル格納用
    │   │
    │   ├─ProcessTaxonomy2---------タクソノミー文書操作用サンプルソース格納用
    │   │       └─input----------入力ファイル格納用
    │   │
    │   └─ProcessTaxonomy3---------タクソノミー文書操作用サンプルソース格納用
    │           ├─input----------入力ファイル格納用
    │           └─output---------サンプル実行後の出力ファイル格納用
    │
    └───dim-------------------------XBRL Dimensions1.0のサンプルファイル格納用
        │
        ├─ProcessDimTaxonomy-------タクソノミー文書操作用サンプルソース格納用
        │       └─input----------入力ファイル格納用
        ├─ProcessDimInstance1------インスタンス文書新規作成用サンプルソース格納用
        │       ├─input----------入力ファイル格納用
        │       └─output---------サンプル実行後の出力ファイル格納用
        ├─ProcessDimInstance2------インスタンス文書新規作成用サンプルソース格納用
        │       └─input----------入力ファイル格納用
        └─schemas------------------スキーマ定義ファイル格納用
```

サンプルソース格納用フォルダには，次のファイルが格納されています。

- 各種 XBRL 文書操作用ソースファイル
- エラーハンドラソースファイル（SampleErrorHandler.java）
- エンティティリゾルバソースファイル（SampleEntityResolver.java）
- コンパイル用バッチファイル（build.bat）

- アプリケーション実行用バッチファイル（run.bat）

4.15.2　実行までの流れ

サンプルアプリケーションを実行するまでの流れについて説明します。

1. **クラスパスの設定**
   uCosminexus Business Reporting Processor などが提供するクラスライブラリを使用するための設定です。「4.2.1　クラスパスの設定」を参照してください。

2. **スキーマ定義のダウンロード**
   XBRL 文書の検証などに使用するスキーマ定義をダウンロードします。
   スキーマ定義の詳細については，「4.15.3　必要なスキーマ定義」を参照してください。

3. **コンパイル用バッチファイルの変更**
   コンパイル用バッチファイルを，アプリケーションの実行環境に合わせて変更します。コンパイル用バッチファイル（setEnv.bat）は，次のフォルダに格納されています。XBRL のバージョンに合ったファイルを変更してください。
   <uCosminexus Business Reporting Processor のインストールフォルダ>¥samples¥2_0
   <uCosminexus Business Reporting Processor のインストールフォルダ>¥samples¥2_1
   <uCosminexus Business Reporting Processor のインストールフォルダ>¥samples¥dim

4. **コンパイル**
   コンパイル用バッチファイルを実行します。
   コンパイル用バッチファイル（build.bat）は，サンプルソースが格納されているフォルダ下にあります。

5. **実行**
   実行用バッチファイルを実行します。コンパイルされたバイトコードは，サンプルソースが格納されているフォルダ下の output フォルダに格納されます。
   実行用バッチファイル（run.bat）は，サンプルソースが格納されているフォルダ下にあります。

4.15.3　必要なスキーマ定義

XBRL 文書の検証などには，スキーマ定義が必要です。公開されているスキーマ定義をダウンロードするか，公開先にアクセスできる状態にしておく必要があります。

必要なスキーマ定義とファイル名を次の表に示します。

表 4-52 必要なスキーマ定義とファイル名

スキーマ定義の分類	ファイル名	提供団体
XBRL2.0 のスキーマ定義	xbrl-instance.xsd	XBRL International これらのファイルは，XBRL2.0a 仕様書に添付されています。
	xbrl-linkbase.xsd	
	xl.xsd	
	xlink.xsd	
XBRL2.1 のスキーマ定義	http://www.xbrl.org/2003/xbrl-instance-2003-12-31.xsd	XBRL International
	http://www.xbrl.org/2003/xbrl-linkbase-2003-12-31.xsd	
	http://www.xbrl.org/2003/xl-2003-12-31.xsd	
	http://www.xbrl.org/2003/xlink-2003-12-31.xsd	
XBRL Dimensions1.0 のスキーマ定義	http://xbrl.org/2005/xbrldt-2005.xsd	XBRL International
	http://xbrl.org/2006/xbrldi-2006.xsd	
XML Schema のスキーマ定義	http://www.w3.org/2001/xml.xsd	W3C
	http://www.w3.org/2001/XMLSchema.xsd	
	http://www.w3.org/2001/XMLSchema.dtd	
	http://www.w3.org/2001/XMLSchema-datatypes.xsd	
	http://www.w3.org/2001/datatypes.dtd	

ダウンロードする場合，ファイルの格納先が XBRL のバージョンまたは XBRL Dimensions のバージョンによって異なります。

XBRL2.0 の場合

<uCosminexus Business Reporting Processor のインストールフォルダ>¥samples¥2_0¥schemas

XBRL2.1 の場合

<uCosminexus Business Reporting Processor のインストールフォルダ>¥samples¥2_1¥schemas

XBRL Dimensions1.0 の場合

<uCosminexus Business Reporting Processor のインストールフォルダ>¥samples¥dim¥schemas

4.16 アプリケーションの実行環境の設定

uCosminexus Business Reporting Processor を利用して作成したアプリケーションを実行する場合の環境設定では，開発環境と同様に，uCosminexus Business Reporting Processor や前提プログラムが提供するクラスライブラリをクラスパスに設定します。また，アプリケーション実行時に uCosminexus Business Reporting Processor の動作を変更するためのプロパティがあります。

ここでは，プロパティの指定方法と優先順位，指定できるプロパティ一覧などについて説明します。

4.16.1 プロパティの指定方法と優先順位

プロパティの指定方法は複数あります。ここでは，プロパティの指定方法と優先順位について説明します。

表 4-53 プロパティの指定方法と優先順位

優先順位	指定方法	設定できるプロパティ
1	XBRLTaxonomyDocumentBuilderFactory または XBRLInstanceDocumentBuilderFactory インターフェースの各メソッド	cbrp.parse. で始まるプロパティ
2	XBRLDocumentManager クラスの init() メソッド	すべてのプロパティ
3	JavaVM のシステムプロパティ	cbrp.parse.XBRLVersion
4	デフォルト設定値	該当しない

優先順位 2 の XBRLDocumentManager クラスの init() メソッドは，XBRL DOM を初期化するメソッドです。XBRL DOM の初期化時に，次の方法でプロパティを指定して，プロパティの値を変更できます。

- プロパティリストを指定する
- プロパティファイルを指定する

各プロパティの指定がない場合，つまりプロパティ名（キー）がないか，値が null または空文字列の場合は，デフォルト値が初期値となります。また，値が指定されていた場合で，各プロパティに許容される値以外が指定されていたときはエラー（例外）になります。

4.16.2 プロパティ一覧

uCosminexus Business Reporting Processor を利用して作成したアプリケーションを実行する場合に，設定できるプロパティについて説明します。

設定内容は，大きく次の分類に分けられます。

- トレースログの設定
- XBRL 文書読み込み時の設定
- タクソノミー文書のメモリキャッシュの設定

(1) cbrp.trace.fileName

出力するトレースログファイル名を絶対 URI で指定します。このプロパティに指定した値に「n.log」を付けた文字列がトレースログファイル名となります。n には，1 から始まる通し番号（整数値）が付けられます。

指定できる XBRL バージョン
　2.0 および 2.1

デフォルト値
　<java.io.tmpdir の値 >/cbrp/cbrptrc
　java.io.tmpdir は，Java VM のシステムプロパティの値です。
　デフォルトのトレースログファイル名は，cbrptrc1.log，crbptrc2.log，crbptrc3.log になります。

指定できる値
　任意の文字列

(2) cbrp.trace.level

トレース出力レベルを指定します。レベルには -1，0，10，20，30 または 40 を設定します。レベル 20 以上を指定した場合，各 API の引数および戻り値の値が出力されます。-1 を指定した場合，トレースログが出力されません。

指定できる XBRL バージョン
　2.0 および 2.1

デフォルト値
　10

指定できる値
　-1 以上 100 以下の数字

レベルと設定値 [n] との関係
　レベル 0 ：0<=n<10
　レベル 10：10<=n<20
　レベル 20：20<=n<30
　レベル 30：30<=n<40
　レベル 40：40<=n<=100

（3） cbrp.trace.maxFileSize

トレースログ 1 ファイル当たりの最大ファイルサイズを指定します。

指定できる XBRL バージョン
　2.0 および 2.1

デフォルト値
　2097152

指定できる値
　4096 以上 2147483647（Integer.MAX_VALUE）以下の数字

（4） cbrp.trace.maxFileSegment

トレースログファイルの最大面数を指定します。

指定できる XBRL バージョン
　2.0 および 2.1

デフォルト値
　10

指定できる値
　1 以上 16 以下の数字

（5） cbrp.parse.XBRLVersion

XBRL 文書の解析時に，読み込む XBRL 文書のバージョンを指定します。

指定できる XBRL バージョン
　2.0 および 2.1

デフォルト値
　2.0

指定できる値
　2.0 または 2.1

（6） cbrp.parse.validate

XBRL 文書解析時に，文書が妥当であることを検証するかどうかを指定します。妥当であることを検証する場合は true，検証しない場合は false を設定します。

このプロパティと cbrp.parse.validate.full-checking の両方で true を指定すると，文書が XBRL 2.1 Conformance Suite 1.0 に準じているかを検証（完全な妥当性検証）できます。

指定できる XBRL バージョン

2.0 および 2.1

デフォルト値
true

指定できる値
true または false。大文字，小文字は区別しません。

（7） cbrp.parse.delayLoad

XBRL 文書解析時に，リンクベースの遅延読み込みをするかどうかを指定します。

この指定は，XBRL 文書を検証しない場合（cbrp.parse.validate=false），かつ FRTA/FRIS 検証に必要となる解析を実施しない場合（cbrp.parse.validate.frtafris=false）だけ有効です。検証する場合，または FRTA/FRIS 検証に必要となる解析を実施する場合には無視されます。

指定できる XBRL バージョン
2.0 および 2.1

デフォルト値
false

指定できる値
true または false。大文字，小文字は区別しません。

（8） cbrp.parse.maxTaxonomyExtension

XBRL 文書解析時に読み込むタクソノミー文書の最大階層数を指定します。このプロパティは，XBRL のバージョンが 2.0 の場合だけ使用できます。

DTS（拡張されたタクソノミー文書）のうち，基底タクソノミー文書を含めた階層数を指定します。0 を指定した場合は，読み込む階層の上限を設けないで，基底タクソノミー文書が見つかるまで解析します。

指定できる XBRL バージョン
2.0

デフォルト値
2

指定できる値
0 以上 2147483647（Integer.MAX_VALUE）以下の数字

（9） cbrp.parse.DTS.ignoringSchemas

タクソノミー本体の <xsd:import> 要素に記述された XML スキーマ文書のうち，タクソノミー本体でない文書など，DTS に含める必要がない文書の名前空間

（targetNamespace）を空白区切りで指定します。このプロパティは，XBRL のバージョンが 2.1 の場合だけ使用できます。

指定した名前空間の XML スキーマ文書は，タクソノミー文書の読み込み時に XBRL プロセッサに無視され，DTS に含まれないものとして扱われます。

指定できる XBRL バージョン
　2.1

デフォルト値
　null

指定できる値
　null または任意の文字列

（10）cbrp.parse.DTS.externalSchemaRefs

DTS を構成するタクソノミー本体（スキーマ定義）の絶対 URI または相対 URI のリストを空白区切りで指定します。このプロパティは，XBRL のバージョンが 2.1 の場合だけ使用できます。

インスタンス文書の解析時に，このプロパティで指定されたタクソノミー本体（スキーマ定義）のリストは，どのタクソノミー本体よりも優先して読み込まれ，DTS の一部となります。そのため，リンクベースの <link:loc> 要素で参照されているだけでインスタンス文書に <link:schemaRef> 要素などの位置情報がないタクソノミー本体の場合でも，リンクベースより先にタクソノミー本体（スキーマ）を解析できます。

このプロパティは，XBRL2.1 の場合でリンクベースの遅延読み込みをするときに，使用してください。XBRL2.1 の場合でリンクベースの遅延読み込みをすると，API の実行順序によって読み込むリンクベースおよび DTS の内容が変わってきます。このプロパティであらかじめ必要なタクソノミー本体を指定することで，常に一定の結果を得られるようになります。

指定できる XBRL バージョン
　2.1

デフォルト値
　null

指定できる値
　null または任意の文字列

（11）cbrp.parse.DTS.maxDocuments

DTS を構成する文書（タクソノミー本体およびリンクベース）のうち，読み込む文書の最大文書数を指定します。0 が指定された場合には無制限に読み込まれます。このプロパティは，XBRL のバージョンが 2.1 の場合だけ使用できます。

このプロパティで指定する文書数には，cbrp.parse.DTS.ignoringSchemas で DTS に含めないと指定したタクソノミー本体は含まれません。
cbrp.parse.DTS.externalSchemaRefs で優先して読み込むように指定したタクソノミー本体は含まれます。

指定できる XBRL バージョン
2.1

デフォルト値
0

指定できる値
0 以上 2147483647（Integer.MAX_VALUE）以下の数字

（12）cbrp.cache.enable

タクソノミー文書のメモリキャッシュをするかどうかを示します。メモリキャッシュをする場合は true，しない場合は false を設定します。

この指定は，リンクベースの遅延読み込みが無効の場合（cbrp.parse.delayLoad=false）だけ有効です。リンクベースの遅延読み込みをする場合は，メモリキャッシュされません。

指定できる XBRL バージョン
2.0 および 2.1

デフォルト値
true

指定できる値
true または false。大文字，小文字は区別しません。

（13）cbrp.cache.maxTaxonomyNum

メモリにキャッシュするタクソノミー文書の最大個数を示します。

この指定はメモリキャッシュをする場合（cbrp.cache.enable=true）だけ有効です。メモリキャッシュをしない場合は無視されます。

指定できる XBRL バージョン
2.0 および 2.1

デフォルト値
3

指定できる値
1 以上 2147483647（Integer.MAX_VALUE）以下の数字

（14）cbrp.cache.registerErrorDocument

XBRL 文書解析時に，エラーが発見された DTS をメモリにキャッシュするかどうかを指定します。true の場合にはメモリキャッシュをし，false の場合にはメモリキャッシュをしません。

この指定はメモリキャッシュをする場合（cbrp.cache.enable=true）だけ有効です。メモリキャッシュをしない場合は無視されます。

指定できる XBRL バージョン
　2.0 および 2.1

デフォルト値
　false

指定できる値
　true または false。大文字，小文字は区別しません。

（15）cbrp.parse.validate.full-checking

XBRL 文書解析時に，完全な妥当性検証を実施するかどうかを指定します。文書中にサイクルの循環がないかなど，XBRL 2.1 Conformance Suite 1.0 に完全に準拠しているかを検証したい場合は true を指定します。実施しない場合は false を指定します。

このプロパティは，次の二つのプロパティが指定されている場合だけ有効です。

- cbrp.parse.XBRLVersion=2.1（XBRL 文書の規格バージョンが 2.1 である）
- cbrp.parse.validate=true（妥当性検証を実施する）

指定できる XBRL バージョン
　2.1

デフォルト値
　false

指定できる値
　true または false。大文字，小文字は区別しません。

（16）cbrp.parse.validate.full-checking.role

インスタンス文書解析時に完全な妥当性検証を実施する場合，各種アークの検証を実施する拡張リンクの xlink:role 属性値を指定します。実施する xlink:role 属性値が複数ある場合には，空白区切りで指定します。指定値が null の場合は，すべての xlink:role 属性値の検証を実施します。

指定できる XBRL バージョン
　2.1

デフォルト値
null

指定できる値
null または任意の文字列

（17）cbrp.parse.validate.full-checking.iso4217

インスタンス文書解析時に完全な妥当性検証を実施する場合，単位の検証に使用する ISO4217 のコードを空白区切りですべて指定します。指定値が null の場合は，デフォルトの方法で検証します。

指定できる XBRL バージョン
2.1

デフォルト値
null

指定できる値
null または任意の文字列

（18）cbrp.parse.validate.full-checking.PTVx.location

インスタンス文書解析時に完全な妥当性検証を実施する場合，PTVLI および PTVI に対するスキーマの位置情報を指定します。この位置情報は，PTVLI および PTVI 作成時に使用されます。

指定できる XBRL バージョン
2.1

デフォルト値
ptv-2003-12-31.xsd

指定できる値
任意の文字列

（19）cbrp.parse.validate.full-checking.PTVI.rounded-item-value

インスタンス文書解析時に完全な妥当性検証を実施する場合，PTVI 作成時の数値アイテム（fractionItemType 型を除く）の値を丸めた値とするかどうかを指定します。丸めた値とする場合は true，インスタンス文書の記述のままとする場合は false を指定します。指定値が false の場合は，インスタンス文書の記述のままになります。

指定できる XBRL バージョン
2.1

デフォルト値
false

指定できる値
true または false。大文字，小文字は区別しません。

（20）cbrp.parse.validate.full-checking.essence-alias.infer

インスタンス文書解析時に完全な妥当性検証を実施する場合，essence-alias アークの検証時にインスタンス文書に存在しないアイテムを再帰的に推測して，検証するかどうかを指定します。

指定できる XBRL バージョン
2.1

デフォルト値
false

指定できる値
true または false。大文字，小文字は区別しません。

（21）cbrp.parse.validate.full-checking.ignoringZeroPrecision

インスタンス文書解析時に完全な妥当性検証を実施する場合，essence-alias アークの検証時や，summation-item アークの検証時に precision 属性値または decimals 属性値から推測される精度が "0" となるアイテムを無視するかどうかを指定します。

指定できる XBRL バージョン
2.1

デフォルト値
false

指定できる値
true または false。大文字，小文字は区別しません。

（22）cbrp.parse.validate.frtafris

XBRL 文書解析時に FRTA/FRIS 検証に必要となる解析を実施するかどうかを指定します。

指定できる XBRL バージョン
2.1

デフォルト値
false

指定できる値
true または false。大文字，小文字は区別しません。

(23) cbrp.parse.DimensionsVersion

XBRL Dimensions 文書の解析時に，読み込む XBRL Dimensions1.0 対応文書のバージョンを指定します。このプロパティに 1.0 を指定し，かつ cbrp.parse.delayLoad=false を指定した場合，DRS を生成します。

また，このプロパティは cbrp.parse.XBRLVersion=2.1 の場合だけ有効となります。cbrp.parse.XBRLVersion=2.0 の場合，このプロパティは無視されます。

指定できる XBRL バージョン
　2.1

デフォルト値
　none

指定できる値
　none または 1.0

(24) cbrp.parse.validate.dimensions.role

XBRL Dimensions 1.0 に準拠したインスタンス文書の解析時に完全な妥当性検証を実施する場合，検証を実施する拡張リンクの xlink:role 属性値を指定します。実施する xlink:role 属性値が複数ある場合は，空白で区切って指定します。その場合，左側に書かれた xlink:role 属性値から順番に検証を実施します。指定値が null の場合は，すべての xlink:role 属性値の検証を実施します。指定値が null の場合，順不同で検証を実施します。また，このプロパティは cbrp.parse.XBRLVersion=2.1 かつ cbrp.parse.DimensionsVersion=1.0 の場合だけ有効となります。cbrp.parse.XBRLVersion=2.0 または cbrp.parse.DimensionsVersion=none の場合，このプロパティは無視されます。

指定できる XBRL バージョン
　2.1

デフォルト値
　null

指定できる値
　null または任意の文字列

(25) cbrp.parse.validate.search-schemaRefs-beforehand

インスタンス文書の妥当性検証を実施する場合に，XML スキーマ検証を実施する前に <link:schemaRef> 要素を検索し，<link:schemaRef> 要素の xlink:href 属性で示されるタクソノミー本体を検証に使用するスキーマとするかどうかを指定します。

このプロパティの値に true を指定した場合は，<link:schemaRef> 要素の xlink:href 属

性で示されるタクソノミースキーマを検証に使用するスキーマとします。ただし，このプロパティの値に true を指定した場合でも，<xbrli:xbrl> 要素に xsi:schemaLocation 属性が記述されていたときは，XML スキーマ検証を実施する前に <link:schemaRef> 要素を検索しません。

このプロパティの値に false を指定した場合は，XML スキーマ検証を実施する前に <link:schemaRef> 要素を検索しません。

指定できる XBRL バージョン
　2.1

デフォルト値
　true

指定できる値
　true または false。大文字，小文字は区別しません。

4.16.3 プロパティファイルの記載形式

プロパティファイルは，uCosminexus Business Reporting Processor を使用して作成したアプリケーションを実行する場合の，動作を定義するためのファイルです。

ここでは，プロパティファイルの記載形式，格納場所，および記載例について説明します。

(1) 記載形式

プロパティファイルは J2SE のプロパティファイル形式です。次の形式で記述します。

- 記述形式：「 キー = 値 」
- 記述例：「 cbrp.trace.level=20 」

コメント行は，空白文字だけの行，または最初の非空白文字が ASCII 文字 # または ! の場合とします。コメントの記述形式は次のとおりです。

- コメントの記述形式：「 # コメント 」または，「 ! コメント 」
- コメントの記述例：「 # CBRP Trace Settings 」

(2) 格納場所

デフォルト値を定義したサンプルのプロパティファイルが，次のフォルダに格納されています。

<uCosminexus Business Reporting Processor のインストールフォルダ >¥properties

サンプルのプロパティファイル名は，XBRL のバージョンによって異なります。

- cbrp.properties（XBRL2.0 の場合）

- cbrp.xbrl_2_1.properties（XBRL2.1 の場合）

サンプルのプロパティファイルをコピーして記述を変更してください。uCosminexus Business Reporting Processor で開発したアプリケーションを実行するときに，プロパティファイルを指定することで，動作を設定できます。

（3） 記載例（XBRL2.0 の場合）

XBRL2.0 の場合の，プロパティファイルの記載例を次に示します。

```
# Copyright (C) 2008, Hitachi, Ltd.
# uCosminexus Business Reporting Processor Properties

# CBRP Trace Settings
# cbrp.trace.fileName       =(FULL PATH prefix)
# cbrp.trace.level          =(0, 10, 20, 30, 40)
# cbrp.trace.maxFileSize    =(4096 - 2147483647)
# cbrp.trace.maxFileSegment =(1 - 16)

cbrp.trace.fileName       =c:/tmp/cbrp/log/cbrptrc
cbrp.trace.level          =10
cbrp.trace.maxFileSize    =2097152
cbrp.trace.maxFileSegment =10


# CBRP Parse Option Settings
# cbrp.parse.validate             =(true , false)
# cbrp.parse.delayLoad            =(true , false)
# cbrp.parse.maxTaxonomyExtension =(0 - 2147483647)

cbrp.parse.validate             =true
cbrp.parse.delayLoad            =false
cbrp.parse.maxTaxonomyExtension =2


# CBRP Cache Settings
# cbrp.cache.enable                =(true , false)
# cbrp.cache.maxTaxonomyNum        =(1 - 2147483647)

cbrp.cache.enable                =true
cbrp.cache.maxTaxonomyNum        =3
```

（4） 記載例（XBRL2.1 の場合）

XBRL2.1 の場合の，プロパティファイルの記載例を次に示します。

```
# Copyright (C) 2008, Hitachi, Ltd.
# uCosminexus Business Reporting Processor Properties

# CBRP Trace Settings
# cbrp.trace.fileName       =(FULL PATH prefix)
# cbrp.trace.level          =(0, 10, 20, 30, 40)
# cbrp.trace.maxFileSize    =(4096 - 2147483647)
# cbrp.trace.maxFileSegment =(1 - 16)

cbrp.trace.fileName       =c:/tmp/cbrp/log/cbrptrc
cbrp.trace.level          =10
cbrp.trace.maxFileSize    =2097152
cbrp.trace.maxFileSegment =10


# CBRP Parse Option Settings
# cbrp.parse.XBRLVersion                              =(2.0, 2.1)
# cbrp.parse.validate                                 =(true , false)
# cbrp.parse.validate.search-schemaRefs-beforehand    =(true , false)
# cbrp.parse.delayLoad                                =(true , false)
# cbrp.parse.DTS.ignoringSchemas                      =(targetNamespace)
# cbrp.parse.DTS.externalSchemaRefs                   =(URI)
# cbrp.parse.DTS.maxDocuments                         =(0 - 2147483647)

cbrp.parse.XBRLVersion                                =2.1
cbrp.parse.validate                                   =true
cbrp.parse.validate.search-schemaRefs-beforehand      =true
cbrp.parse.delayLoad                                  =false
cbrp.parse.DTS.ignoringSchemas                        =
cbrp.parse.DTS.externalSchemaRefs                     =
cbrp.parse.DTS.maxDocuments                           =0


# CBRP Parse Full-Checking Option Settings
# cbrp.parse.validate.full-checking                         =(true, false)
# cbrp.parse.validate.full-checking.role                    =(URI)
# cbrp.parse.validate.full-checking.iso4217                 =(iso4217 code)
# cbrp.parse.validate.full-checking.PTVx.location           =(location)
# cbrp.parse.validate.full-checking.PTVI.rounded-item-value =(true, false)
# cbrp.parse.validate.full-checking.essence-alias.infer     =(true, false)

cbrp.parse.validate.full-checking                         =false
cbrp.parse.validate.full-checking.role                    =
cbrp.parse.validate.full-checking.iso4217                 =
cbrp.parse.validate.full-checking.PTVx.location           =ptv-2003-12-31.xsd
cbrp.parse.validate.full-checking.PTVI.rounded-item-value =false
cbrp.parse.validate.full-checking.essence-alias.infer     =false

# CBRP Parse FRTA-FRIS Option Settings
# cbrp.parse.validate.frtafris                              =(true, false)

cbrp.parse.validate.frtafris                              =false

# CBRP Parse Dimensions Option Settings
# cbrp.parse.DimensionsVersion        =(none , 1.0)
# cbrp.parse.validate.dimensions.role =(URI)

cbrp.parse.DimensionsVersion        =none
cbrp.parse.validate.dimensions.role =

# CBRP Cache Settings
# cbrp.cache.enable                =(true , false)
# cbrp.cache.registerErrorDocument =(true , false)
# cbrp.cache.maxTaxonomyNum        =(1 - 2147483647)

cbrp.cache.enable                =true
cbrp.cache.registerErrorDocument =false
cbrp.cache.maxTaxonomyNum        =3
```

4.16.4　文字コードの設定

ここでは，文字コードの設定について説明します。

XML プロセッサは，XML 文書の encoding 属性に "Shift_JIS" が指定された場合，デフォルトの動作では，SJIS(x-sjis-jdk1.1.7) と解釈して UNICODE 変換します。このため，作成された W3C DOM では，Windows で使用できる文字の一部が UNICODE 変換されなくなり，「？」に文字化けすることがあります。

正確に表示するためには，Windows 上で記述された XML 文書の encoding 属性を "MS932" または "Windows-31J" に変更してください。

また，XML プロセッサの設定で，UNICODE 変換に用いる変換テーブルを "SJIS" または "MS932" に切り替えることもできます。切り替え方法の詳細については，uCosminexus Application Server を使用している場合にはマニュアル「Cosminexus XML Processor ユーザーズガイド」を，Processing Kit for XML を使用している場合にはマニュアル「Processing Kit for XML ユーザーズガイド」を参照してください。

4.16.5　JavaVM のカスタマイズ

規模の大きいタクソノミー文書などを扱う場合は，Java VM の引数に使用するメモリサイズを指定する必要があります。

uCosminexus Business Reporting Processor が提供するサンプルアプリケーションを実行する場合などは，特に指定する必要はありません。

メモリサイズの設定例を次の表に示します。

表 4-54　メモリサイズの設定例

項番	設定する内容	設定例	備考
1	最小 VM メモリサイズ	-Xms256m	任意
2	最大 VM メモリサイズ	-Xmx512m	任意

4.17　トレースログ

uCosminexus Business Reporting Processor では，トレースログを出力できます。ここでは，トレースログ出力の設定，およびトレース出力レベルについて説明します。

トレースログ出力の設定

uCosminexus Business Reporting Processor のトレースログを出力するための設定には，次のプロパティがあります。

- トレースログファイル名
- トレース出力レベル
- トレースログ 1 ファイル当たりの最大ファイルサイズ
- トレースログファイルの最大面数

これらのプロパティの詳細，および指定方法については，「4.16　アプリケーションの実行環境の設定」を参照してください。

トレース出力レベル

トレースログを出力する場合，トレース出力レベルを指定します。出力レベルには，レベル 0，レベル 10，レベル 20，レベル 30，レベル 40 と，5 段階のレベルがあります。レベルの数字が大きくなると，より詳細なメッセージが表示されます。

レベル 20 以上を指定した場合，各 API の引数および戻り値の値が，トレースログに出力されます。

5 取り扱えるXBRL文書とデータ型

この章では，uCosminexus Business Reporting Processorで取り扱えるXBRL文書の基本構造，および定義できる要素やその属性値などについて説明します。また，使用できるデータ型についても説明します。
なお，文書の構造などは，XBRLのバージョンによって異なります。

5.1 取り扱えるタクソノミー文書（XBRL2.0の場合）

5.2 取り扱えるインスタンス文書（XBRL2.0の場合）

5.3 取り扱えるタクソノミー文書（XBRL2.1の場合）

5.4 取り扱えるインスタンス文書（XBRL2.1の場合）

5.5 取り扱えるデータ型

5.1 取り扱えるタクソノミー文書（XBRL2.0 の場合）

uCosminexus Business Reporting Processor で取り扱えるタクソノミー文書の形式について説明します。

タクソノミー文書は，次のスキーマ定義に従って記述されている必要があります。

XBRL のスキーマ定義（XBRL2.0 の場合）

XBRL International が提供するスキーマ定義です。XBRL のバージョンによって異なります。

- xbrl-instance.xsd
- xbrl-linkbase.xsd
- xl.xsd
- xlink.xsd

これらのファイルは，XBRL2.0a 仕様書に添付されています。

XML Schema のスキーマ定義

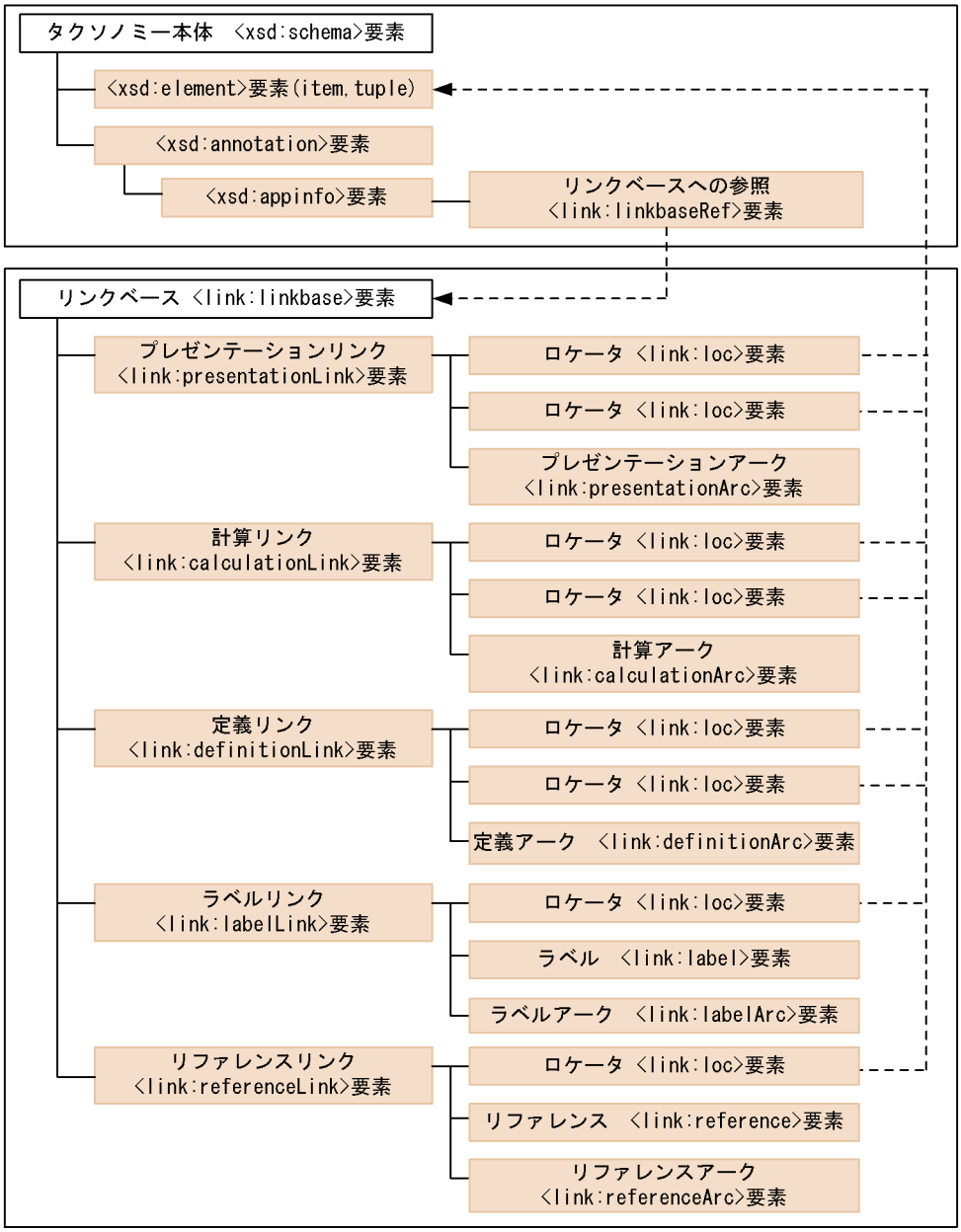
W3C が提供するスキーマ定義です。

- http://www.w3.org/2001/xml.xsd
- http://www.w3.org/2001/XMLSchema.xsd
- http://www.w3.org/2001/XMLSchema.dtd
- http://www.w3.org/2001/XMLSchema-datatypes.xsd
- http://www.w3.org/2001/datatypes.dtd

5.1.1 タクソノミー文書の基本構造

XBRL2.0 の場合の，タクソノミー文書の基本構造を次の図に示します。

図 5-1　タクソノミー文書の基本構造



（凡例）- - - ►　：XLinkまたはXPointerによる参照関係

5.1.2　エレメント要素（element）

ここでは，XBRL2.0 の場合のタクソノミー文書で使用するエレメント要素について説明します。

タクソノミー文書では，XML スキーマの <xsd:element> 要素を使用して，インスタンス文書中に出現できる要素を定義します。エレメント要素では，アイテム型要素，タプル型要素などを定義します。

uCosminexus Business Reporting Processor は，<xsd:element> 要素が <xsd:schema> 要素の子要素として定義されている場合に，エレメント要素を処理します。また，uCosminexus Business Reporting Processor で扱うタクソノミー文書では属性定義が必要で，属性に指定する値も条件を満たす必要があります。uCosminexus Business Reporting Processor は，必須条件を満たす <xsd:element> 要素を処理対象とし，満たさない <xsd:element> 要素は無視します。

(1) アイテム型要素（item）

uCosminexus Business Reporting Processor では，次の条件を満たすアイテム型要素を使用できます。

表 5-1 エレメント要素の属性（アイテム型要素）

項番	属性名	内容	必須条件
1	name	名前属性です。内容は任意です。	必須
2	type	データ型属性です。XBRL データ型またはユーザー独自のデータ型を使用できます。 詳細は，表下の「取り扱えるデータ型」を参照してください。	任意
3	substitutionGroup	代替グループ属性です。総称名となる基本要素として xbrli:item を記述します。	必須
4	id	リンクベースによって関係づけられる場合に必須です。デフォルトは null です。	任意
5	xbrli:balance	指定できる値は debit または credit です。デフォルトは null です。	任意

取り扱えるデータ型

uCosminexus Business Reporting Processor は，type 属性が XBRL データ型である要素宣言を処理できます。認識できないデータ型は，ユーザー独自のデータ型（**ユーザーデータ型**）として解釈します。

ユーザーデータ型は，ユーザーが独自に定義する属性のことです。定義した属性は，XBRL DOM を介して取得できます。定義内容は，スキーマ定義に基づいて検証されます。

ユーザーデータ型を XBRL データ型から派生して作成した場合は，ユーザーデータ型が数値または非数値データ型かどうかが判断されます。それ以外の場合，ユーザーデータ型は文字列として保持・参照されます。

なお，type 属性が指定されていない場合，substitutionGroup 属性によってアイテム型要素として処理され，ユーザーデータ型と同じくその他（OTHER_ITEM）として解釈されます。

取り扱えるユーザーデータ型の詳細については，「5.5　取り扱えるデータ型」を参照してください。

定義例

アイテム型要素の定義例を次に示します。

```
<element id                ="paymentOfdividends.preferredDividends"
         name              ="paymentOfdividends.preferredDividends"
         type              ="xbrli:monetaryItemType"
         substitutionGroup ="xbrli:item"
/>
```

(2) タプル型要素（tuple）

uCosminexus Business Reporting Processor では，次の条件を満たすタプル型要素を使用できます。

表 5-2　エレメント要素の属性（タプル型要素）

項番	属性名	内容	必須条件
1	name	名前属性です。内容は任意です。	必須
2	type	データ型属性です。xbrli:tupleType を記述します。詳細は，表下の「type 属性の指定」を参照してください。	任意
3	substitutionGroup	代替グループ属性です。総称名となる基本要素として xbrli:tuple を記述します。	必須
4	id	リンクベースによって関係づけられる場合に必須です。デフォルトは null です。	任意
5	xbrli:balance	指定できる値は debit または credit です。デフォルトは null です。	任意

type 属性の指定

uCosminexus Business Reporting Processor では，substitutionGroup 属性が xbrli:tuple の場合，type 属性がスキーマ定義どおりであるかを検証します。

type 属性が指定されていない場合，substitutionGroup 属性によってタプル型要素として処理されます。また，substitutionGroup 属性に xbrli:tuple が指定され，type 属性に xbrli:tupleType 以外が指定された場合は，スキーマ検証でエラーとなります。

(3) そのほかのエレメント要素

uCosminexus Business Reporting Processor では，<xsd:schema> の子要素である <xsd:element> 要素を，アイテム型要素またはタプル型要素として一度検証します。

この検証でアイテム型要素またはタプル型要素と認識できない要素の場合，そのほかの要素として扱います。<xsd:schema> 要素の子要素でない <xsd:element> 要素や，<xsd:element> 要素に name 属性を持たない要素については無視します。

そのほかの要素として扱う <xsd:element> 要素は次のとおりです。

- <xsd:schema> の子要素である <xsd:element> 要素
- name 属性を持ち，item 型または tuple 型の条件を満たさない要素
  （substitutionGroup 属性が xbrli:item または xbrli:tuple 以外の要素）

このため，そのほかの要素がリンクベースのトラバース先に指定されている場合でも，リンクベースを処理できます。

5.1.3 リンクベースへの参照定義

タクソノミー本体からリンクベースへの参照定義には，<link:linkbaseRef> 要素を使用します。

(1) <link:linkbaseRef> 要素

uCosminexus Business Reporting Processor は，<xsd:schema>/<xsd:annotation>/<xsd:appinfo> の下位に記述されている <link:linkbaseRef> 要素を処理します。uCosminexus Business Reporting Processor で扱うタクソノミー文書では属性定義が必要で，属性に指定する値も条件を満たす必要があります。<xsd:schema>/<xsd:annotation>/<xsd:appinfo> の下位に記述されている <link:linkbaseRef> 要素が必須条件を満たさない場合は，エラーとして処理を中止します。

タクソノミー文書で必要な <link:linkbaseRef> 要素の属性を次の表に示します。

表 5-3 <link:linkbaseRef> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	simple（単純リンク）固定です。	必須
2	xlink:href	拡張リンク型要素の定義ファイルへの URI またはローカルならファイル名を記述します。値は任意です。	必須
3	xlink:role	リンク種別を意味する文字列です。デフォルトは null です。詳細は，表下の「xlink:role 属性に指定する値」を参照してください。	任意
4	xlink:arcrole	http://www.w3.org/1999/xlink/properties/linkbase 固定です。そのほかの値を指定した場合は，文書解析時にエラーになります。	必須
5	xlink:show	指定できる値は，new，replace，embed，other，または none です。属性を省略できます。	任意
6	xlink:title	任意の文字列を設定します。	任意
7	xlink:actuate	onRequest 固定です。	必須

xlink:role 属性に指定する値

xlink:role 属性を指定する場合，リンク種別ごとに異なる推奨値を使用します。

xlink:role 属性を省略すると，リンクの読み込み時に読み込む必要がないリンクを読み込んでしまうため，xlink:role 属性を指定することを推奨します。
xlink:role 属性のリンク種別ごとの推奨値を次の表に示します。

表 5-4　xlink:role 属性のリンク種別ごとの推奨値

項番	拡張リンク種別	linkbaseRef 要素の xlink:role 属性の推奨値
1	プレゼンテーションリンク	http://www.xbrl.org/linkprops/linkRef/presentation
2	計算リンク	http://www.xbrl.org/linkprops/linkRef/calculation
3	定義リンク	http://www.xbrl.org/linkprops/linkRef/definition
4	ラベルリンク	http://www.xbrl.org/linkprops/linkRef/label
5	リファレンスリンク	http://www.xbrl.org/linkprops/linkRef/reference

uCosminexus Business Reporting Processor では，この推奨値を用いてリンク種別を判断します。リンクベースのファイルの内容が目的のリンクと異なる場合は，エラーになります。また，推奨値以外の値を指定した場合もエラーになります。

定義例

ラベルリンクの定義例を次に示します。

```
<schema
 :
xmlns:link="http://www.xbrl.org/XLink/xbrllinkbase"
 :
<link:linkbaseRef
    xlink:type="simple"・・・・・・・・・・・・・・・・・・・・・・・・・・固定値
    xlink:href="Sample_XBRL_linkbase_label.xml" ・・・・・・・・・・・・任意(URI)
    xlink:role="http://www.xbrl.org/linkprops/linkRef/label"・・・・・・推奨値
    xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"・・固定値
    xlink:actuate="onRequest" />・・・・・・・・・・・・・・・・・・・・・固定値
 :
</schema>
```

5.1.4　リンクベース要素（linkbase）

タクソノミー文書は，インスタンス文書中に出現できる要素宣言のほかに，各要素の関係を示すリンクベースを記述します。ここでは，XBRL2.0 の場合のリンクベースで使用する要素について説明します。

（1） <link:linkbase> 要素

<link:linkbase> 要素は，タクソノミー文書で使用される拡張リンク型要素，または <link:linkbaseRef> 要素を記述するためのコンテナ要素として利用します。また，<link:linkbase> 要素は，拡張リンク型要素を別ドキュメントに記述するためのルート要素です。

<link:linkbase> 要素には，xmlns 属性や xsi:schemaLocation 属性などで，スキーマ検証で使用するタクソノミー文書を指定する必要があります。

(2) 拡張リンク型要素（extend link-type）

タクソノミー文書で使用される拡張リンクのリンクベースは，「5.1.3(1) <link:linkbaseRef> 要素」の <link:linkbaseRef> 要素の xlink:href 属性に記述されたファイルまたは URI に対応する実体です。

タクソノミー文書に記述するどの拡張リンク型要素にも，次の属性定義が必要です。

表 5-5　拡張リンク型要素の共通属性

項番	属性名	内容	必須条件
1	xlink:type	extended（拡張リンク）固定です。	必須
2	xlink:role	リンク種別を意味する文字列で，任意の値です。値を指定しても，uCosminexus Business Reporting Processor の動作は変わりません。デフォルトは null です。	任意
3	xlink:title	任意の文字列を設定します。デフォルトは null です。	任意

XBRL2.0 の仕様で定義されていない子要素がタクソノミー文書に記述されている場合，警告を通知して読み飛ばし，処理を続けます。

拡張リンク型要素は，アーク型要素，ロケータ型要素およびリソース型要素のコンテナ要素となり，リンク種別ごとに子要素として含むことができる要素が異なります。次に，関係リンク，ラベルリンクおよびリファレンスリンクで含むことができる要素について示します。

(a) 関係リンク

- プレゼンテーションリンク
  <link:presentationLink> 要素では，アイテム間の表示順を定義します。
  <link:loc> 要素と <link:presentationArc> 要素を含むことができます。
- 計算リンク
  <link:calculationLink> 要素では，アイテムの数値データの重み付き加減算式を定義します。
  <link:loc> 要素と，<link:calculationArc> 要素を含むことができます。
- 定義リンク
  <link:definitionLink> 要素では，アイテム間の親子関係など，アイテム同士の関係を定義します。
  <link:loc> 要素と <link:definitionArc> 要素を含むことができます。

(b) ラベルリンク（labelLink）

<link:labelLink> 要素では，アイテムの表示名称（ラベル）を定義します。日本語，英語などさまざまな言語でラベルを記述したり，別名を定義したりします。

<link:loc> 要素，<link:label> 要素および <link:labelArc> 要素を含むことができます。

(c) リファレンスリンク（referenceLink）

<link:referenceLink> 要素では，参考文献を定義します。会計概念を定義する根拠となっている文献への参照などに用います。

<link:loc> 要素，<link:reference> 要素および <link:referenceArc> 要素を含むことができます。

(3) <link:loc> 要素

<link:loc> 要素は，拡張リンクでのロケータ型要素で，タクソノミー文書では <link:calculationLink> 要素などの拡張リンク型の子要素です。<link:loc> 要素の属性を次の表に示します。

表 5-6 <link:loc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	locator 固定です。	必須
2	xlink:href	XPointer のショートハンドポインタ（Shorthand Pointer）形式で指定します。記述形式を次に示します。 {URI}#id URI：参照先ドキュメントの絶対 URI または相対 URI です。URI は同一ドキュメント内であれば省略できます。 id：参照先ドキュメント内の ID 型属性値です。	必須
3	xlink:label	NCName 型データです。アーク型要素によって参照しているロケータ型要素（<link:loc>）を識別するために必要です。	必須
4	xlink:role	ツリー構造の開始点（ルート）を明示的に記述するときなどに使用します。XBRL では，ルートとなるロケータ型要素の場合，次の値を推奨しています。 http://www.xbrl.org/linkprops/locator/root ただし，この値によるルートを判定しません。そのため，設定値によって uCosminexus Business Reporting Processor の動作が変わることはありません。そのほかの値を指定した場合も動作は変わりません。デフォルトは null です。	任意
5	xlink:title	任意の値です。デフォルトは null です。	任意

この形式以外のロケータ型要素が出現した場合，エラーとして処理を中止します。

(4) <link:label> 要素

タクソノミー文書での，リソース型要素の一つで，<link:labelLink> 要素の子要素です。<link:label> 要素の属性を次の表に示します。

表 5-7 <link:label> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	resource 固定です。	必須
2	xlink:label	NCName 型データです。アーク型要素によって参照しているリソース型要素（label）を識別するために必要です。同一の値を持つことが許されます。	必須
3	xlink:role	任意の値です。デフォルトは null です。この属性値によって，同一言語であっても用途によって区別できます。 推奨値を次に示します。そのほかの値を指定しても uCosminexus Business Reporting Processor の動作は変わりません。 • http://www.xbrl.org/linkprops/label/standard • http://www.xbrl.org/linkprops/label/total ただし，この値による用途は判定しません。そのため，設定値によって動作が変わることはありません。デフォルトは null です。	任意
4	xml:lang	任意の言語を特定するために使用します。ISO で定義されている言語コードである「ja」,「en」などです。	必須
5	xlink:title	任意の値です。デフォルトは null です。	任意

<link:label> 要素には，任意の文字列を要素の値として記述します。

この形式以外のリソース型要素が出現した場合，エラーとして処理を中止します。

(5) <link:reference> 要素

タクソノミー文書での，リソース型要素の一つで，<link:referenceLink> 要素の子要素です。<link:reference> 要素の属性を次の表に示します。

表 5-8 <link:reference> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	resource 固定です。	必須
2	xlink:label	NCName 型データです。アーク型要素によって参照しているリソース型要素（<link:reference>）を識別するために必要です。同一の値を持つことが許されます。	必須
3	xlink:role	任意の値です。デフォルトは null です。	任意
4	xlink:title	任意の値です。デフォルトは null です。	任意

<link:reference> 要素には，任意の子要素を含むことができます。この任意要素を総称し，抽象的な要素表現である part（パート）と呼びます。

この形式以外のリソース型要素が出現した場合，エラーとして処理を中止します。

（6） <link:presentationArc> 要素

<link:presentationArc> 要素は，タクソノミー文書のアーク型要素の一つで，<link:presentationLink> 要素の子要素です。<link:presentationArc> 要素の属性を次の表に示します。

表 5-9 <link:presentationArc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	arc 固定です。	必須
2	xlink:show	embed または replace を指定します。	必須
3	xlink:actuate	onRequest 固定です。	必須
4	xlink:from	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
5	xlink:to	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
6	xlink:arcrole	次のどれかの値を指定します。 • http://www.xbrl.org/linkprops/arc/parent-child • http://www.xbrl.org/linkprops/arc/child-parent • http://www.xbrl.org/linkprops/arc/dimension-element • http://www.xbrl.org/linkprops/arc/element-dimension	必須
7	xlink:title	任意の値を指定します。デフォルトは null です。	任意
8	use	optional，prohibited または required を指定します。デフォルトは optional です。	任意
9	priority	use 属性の optional，prohibited，required の関係に対してのプライオリティ（優先順位）を指定するための属性です。デフォルトは 0 です。	任意
10	order	複数の兄弟要素間での表示順序を示す，マイナスでない 10 進数を指定します。デフォルトは 1 です。	任意

この形式以外のアーク型要素が出現した場合，エラーとして処理を中止します。

（7） <link:calculationArc> 要素

<link:calculationArc> 要素は，タクソノミー文書のアーク型要素の一つで，<link:calculationLink> 要素の子要素です。<link:calculationArc> 要素の属性を次の表に示します。

表 5-10 <link:calculationArc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	arc 固定です。	必須
2	xlink:show	embed または replace を指定します。	必須

項番	属性名	内容	必須条件
3	xlink:actuate	onRequest 固定です。	必須
4	xlink:from	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
5	xlink:to	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
6	xlink:arcrole	次のどれかの値を指定します。 • http://www.xbrl.org/linkprops/arc/parent-child • http://www.xbrl.org/linkprops/arc/child-parent • http://www.xbrl.org/linkprops/arc/dimension-element • http://www.xbrl.org/linkprops/arc/element-dimension	必須
7	xlink:title	任意の値を指定します。デフォルトは null です。	任意
8	use	optional，prohibited または required を指定します。デフォルトは optional です。	任意
9	priority	use 属性の optional，prohibited，required の関係に対してのプライオリティ（優先順位）を指定するための属性です。デフォルトは 0 です。	任意
10	weight	子要素から親要素に数値を加算するときに，その要素の値に適用する乗数を示します。値「1.0」はその要素の数値を 100% 親要素に加算し，「-1.0」は 100% 減算することを表します。	必須

この形式以外のアーク型要素が出現した場合，エラーとして処理を中止します。

(8) <link:definitionArc> 要素

<link:definitionArc> 要素は，タクソノミー文書のアーク型要素の一つで，<link:definitionLink> 要素の子要素として出現します。<link:definitionArc> 要素の属性を次の表に示します。

表 5-11 <link:definitionArc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	arc 固定です。	必須
2	xlink:show	embed または replace を指定します。	必須
3	xlink:actuate	onRequest 固定です。	必須
4	xlink:from	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
5	xlink:to	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須

項番	属性名	内容	必須条件
6	xlink:arcrole	次のどれかの値とします。 • http://www.xbrl.org/linkprops/arc/parent-child • http://www.xbrl.org/linkprops/arc/child-parent • http://www.xbrl.org/linkprops/arc/dimension-element • http://www.xbrl.org/linkprops/arc/element-dimension	必須
7	xlink:title	任意の値を指定します。デフォルトは null です。	任意
8	use	optional，prohibited または required を指定します。デフォルトは optional です。	任意
9	priority	use 属性の optional，prohibited，required の関係に対してのプライオリティ（優先順位）を指定するための属性です。デフォルトは 0 です。	任意

この形式以外のアーク型要素が出現した場合，エラーとして処理を中止します。

（9） <link:labelArc> 要素

<link:labelArc> 要素は，タクソノミー文書のアーク型要素の一つで，<link:labelLink> 要素の子要素です。<link:labelArc> 要素の属性を次の表に示します。

表 5-12 <link:labelArc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	arc 固定です。	必須
2	xlink:show	embed または replace を指定します。	必須
3	xlink:actuate	onRequest 固定です。	必須
4	xlink:from	NCName 型データです。一つのロケータ型要素（<link:loc>）またはリソース型要素（<link:label>）の xlink:label 属性の内容と同じ値を指定します。	必須
5	xlink:to	NCName 型データです。一つのロケータ型要素（<link:loc>）またはリソース型要素（<link:label>）の xlink:label 属性の内容と同じ値を指定します。	必須
6	xlink:arcrole	次のどちらかの値を指定します。 • http://www.xbrl.org/linkprops/arc/label-element • http://www.xbrl.org/linkprops/arc/element-label	必須
7	xlink:title	任意の値を指定します。デフォルトは null です。	任意
8	use	optional，prohibited または required を指定します。デフォルトは optional です。	任意
9	priority	use 属性の optional，prohibited，および required の関係に対してのプライオリティ（優先順位）を指定するための属性です。デフォルトは 0 です。	任意

この形式以外のアーク型要素が出現した場合，エラーとして処理を中止します。

（10）<link:referenceArc> 要素

<link:referenceArc> 要素は，タクソノミー文書のアーク型要素の一つで，<link:referenceLink> 要素の子要素です。<link:referenceArc> 要素の属性を次の表に示します。

表 5-13　<link:referenceArc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	arc 固定です。	必須
2	xlink:show	embed または replace を指定します。	必須
3	xlink:actuate	onRequest 固定です。	必須
4	xlink:from	NCName 型データです。一つのロケータ型要素（<link:loc>）またはリソース型要素（<link:reference>）の xlink:label 属性の内容と同じ値を指定します。	必須
5	xlink:to	NCName 型データです。一つのロケータ型要素（<link:loc>）またはリソース型要素（<link:reference>）の xlink:label 属性の内容と同じ値を指定します。	必須
6	xlink:arcrole	次のどれかの値を指定します。 • http://www.xbrl.org/linkprops/arc/reference-element • http://www.xbrl.org/linkprops/arc/element-reference • http://www.xbrl.org/linkprops/arc/actual-element ※ • http://www.xbrl.org/linkprops/arc/element-actual ※	必須
7	xlink:title	任意の値を指定します。デフォルトは null です。	任意
8	use	optional，prohibited または required を指定します。 デフォルトは optional です。	任意
9	priority	use 属性の optional，prohibited，required の関係に対してのプライオリティ（優先順位）を指定するための属性です。デフォルトは 0 です。	任意

注※　この値を指定した場合，ロケータで指定された URI を返します。

この形式以外のアーク型要素が出現した場合，エラーとして処理を中止します。

5.2　取り扱えるインスタンス文書（XBRL2.0 の場合）

uCosminexus Business Reporting Processor で取り扱えるインスタンス文書の形式について説明します。ここでは，文書の基本構造，および定義できる要素やその属性値などについて説明します。

インスタンス文書は，次のスキーマに従って記述されている必要があります。

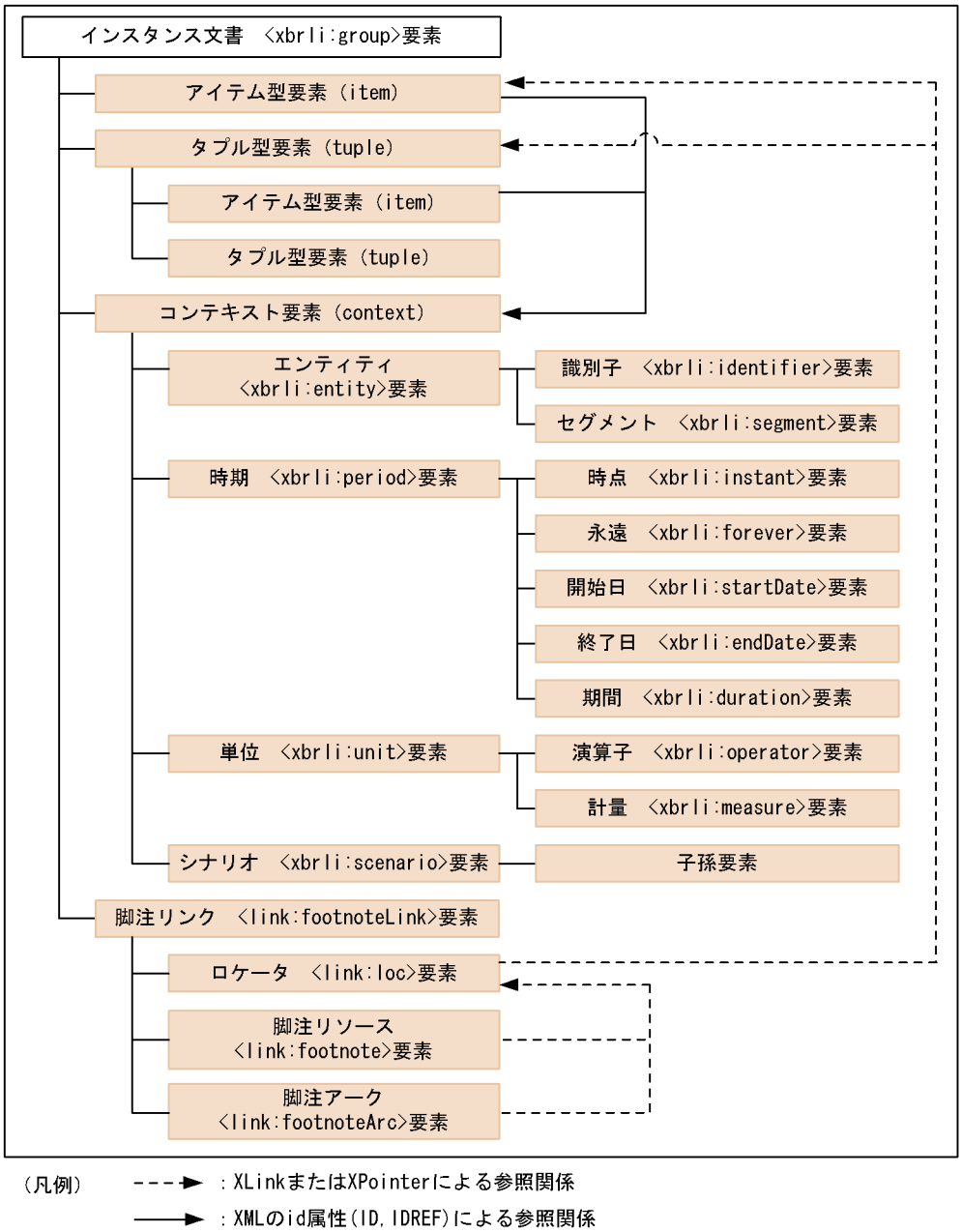
- タクソノミー文書（ユーザー定義スキーマ）
- XBRL のスキーマ定義
- XML Schema のスキーマ定義

XBRL のスキーマ定義および XML Schema のスキーマ定義の詳細については，「5.1　取り扱えるタクソノミー文書（XBRL2.0 の場合）」を参照してください。

5.2.1　インスタンス文書の基本構造

XBRL2.0 の場合の，インスタンス文書の基本構造を次の図に示します。

図 5-2　インスタンス文書の基本構造


インスタンス文書　<xbrli:group>要素
アイテム型要素（item）
タプル型要素（tuple）
アイテム型要素（item）
タプル型要素（tuple）
コンテキスト要素（context）
エンティティ
<xbrli:entity>要素
識別子　<xbrli:identifier>要素
セグメント　要素
時期　<xbrli:period>要素
時点　<xbrli:instant>要素
永遠　<xbrli:forever>要素
開始日　<xbrli:startDate>要素
終了日　<xbrli:endDate>要素
期間　<xbrli:duration>要素
単位　<xbrli:unit>要素
演算子　<xbrli:operator>要素
計量　<xbrli:measure>要素
シナリオ　<xbrli:scenario>要素
子孫要素
脚注リンク　<link:footnoteLink>要素
ロケータ　<link:loc>要素
脚注リソース
<link:footnote>要素
脚注アーク
<link:footnoteArc>要素
（凡例）
：XLinkまたはXPointerによる参照関係
：XMLのid属性(ID,IDREF)による参照関係

5.2.2　ルート要素（group）

XBRL2.0 の仕様では，<xbrli:group> 要素はインスタンス文書のルート要素，または任意の位置に出現できるコンテナ要素として定義されています。

uCosminexus Business Reporting Processor では，ルート要素以外の <xbrli:group> 要素は処理対象としません。インスタンス文書中の任意の場所に出現した場合は，無視して下位要素（次の子孫要素）を処理します。ただし，W3C DOM からは削除しないため，出力時には処理したインスタンス文書の構造は変わりません。

5.2.3 アイテム型要素（item）

ここでは，XBRL2.0 の場合のアイテム型要素について説明します。

アイテム型要素は，一つの事実やビジネス数量を表します。アイテム型要素は，タクソノミー文書で定義された type 属性によって，numericContext 属性または nonNumericContext 属性のどちらかが必須となり，共にコンテキスト要素の id 属性値を記述します。

アイテム型要素の検証内容を次の表に示します。

表 5-14 アイテム型要素の検証内容

項番	属性名	検証内容
1	numericContext	この属性で参照する要素が <xbrli:numericContext> 要素であるかを検証します。
2	nonNumericContext	この属性で参照する要素が <xbrli:nonNumericContext> 要素であるかを検証します。

5.2.4 コンテキスト要素（context）

コンテキスト要素には，エンティティ要素（entity），時期要素（period），単位要素（unit），およびシナリオ要素（scenario）を定義できます。ここでは，これらのうち，XBRL2.0 の場合の，時期要素および単位要素について説明します。

（1） 時期要素（period）

<xbrli:period> 要素は，<xbrli:numericContext> 要素，<xbrli:nonNumericContext> 要素の必須要素で，時期または期間を表現するためのコンテナ要素です。子要素を持ち，属性は持ちません。必須属性は定義されていません。出現位置や出現回数については，スキーマで保証されます。

<xbrli:period> 要素の子要素を次の表に示します。

表 5-15 <xbrli:period> 要素の子要素

項番	要素名	説明
1	<xbrli:instant>	ある特定の時点を示すときに使用し，dateUnion 型（XML の date 型または dateTime 型）の値を指定します。
2	<xbrli:forever>	永遠を表現するための要素です。属性，子要素，値があっても無視します。

項番	要素名	説明
3	<xbrli:startDate> <xbrli:endDate>	開始日と終了日を示し，期間を表現する形式です。共に，dateUnion 型の値を指定します。
4	<xbrli:startDate> <xbrli:duration>	<xbrli:startDate> に開始日を指定し，<xbrli:duration> に期間を指定する形式です。<xbrli:startDate> は dateUnion 型で， <xbrli:duration> 要素は duration 型の値を指定します。
5	<xbrli:duration> <xbrli:endDate>	<xbrli:endDate> に終了日を指定し，<xbrli:duration> に期間を指定する形式です。<xbrli:endDate> は dateUnion 型で， <xbrli:duration> 要素は duration 型の値を指定します。

この組み合わせのうち項番 3 の <xbrli:startDate> 要素と <xbrli:endDate> 要素の組み合わせについて，uCosminexus Business Reporting Processor は日時が逆転していないかどうかをチェックし，エラーを通知します。

(2) 単位要素（unit）

<xbrli:unit> 要素は，<xbrli:numericContext> 要素の必須要素で，計量に関する標準を指定します。<xbrli:nonNumericContext> 要素ではオプションになります。

<xbrli:unit> 要素には，単位を示す ISO4217 で既定されている値を使用する必要があります。ただし，uCosminexus Business Reporting Processor では，<xbrli:unit> 要素に指定された値が，ISO4217 で規定されているかどうかチェックしていません。

5.2.5 脚注リンク要素（footnoteLink）

ここでは，XBRL2.0 の場合の脚注リンク要素について説明します。

<link:footnoteLink> 要素は，インスタンス文書で使用される拡張リンク型要素で，属性定義が必要です。また，<link:footnoteLink> 要素には，ロケータ型要素（loc），脚注リソース型要素（footnote），脚注アーク型要素（footnoteArc）を定義できます。

<link:footnoteLink> 要素の属性を次の表に示します。

表 5-16 <link:footnoteLink> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	extended（拡張リンク）固定です。	必須
2	xlink:role	リンク種別を意味する文字列です。任意の値を指定します。 デフォルトは null です。	任意
3	xlink:title	任意の文字列を指定します。デフォルトは null です。	任意

(1) ロケータ型要素（loc）

<link:loc> 要素は，拡張リンクでのロケータ型要素で，インスタンス文書では

<link:footnoteLink> 要素の子要素として出現します。<link:loc> 要素の属性を次の表に示します。

表 5-17 <link:loc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	locator 固定です。	必須
2	xlink:href	XPointer のショートハンドポインタ（Shorthand Pointer）形式とし，次に示す記述形式を使用できます。 {URI}#id URI：参照先ドキュメントの絶対 URI または相対 URI です。URI は同一ドキュメント内であれば省略できます。 id：参照先ドキュメント内の id 属性値です。	必須
3	xlink:label	NCName 型データです。アーク型要素によって参照しているロケータ型要素を識別するために必要です。	必須
4	xlink:role	ツリー構造の開始点（ルート）を明示的に記述するときなどに使用します。XBRL では，ルートとなるロケータ型要素の場合，次の値を推奨しています。 http://www.xbrl.org/linkprops/locator/root ただし，この値によるルートを判定しません。そのため，設定値によって動作が変わることはありません。デフォルトは null です。	任意
5	xlink:title	任意の値を指定します。デフォルトは null です。	任意

この形式以外のロケータ型要素が出現した場合，エラーとして処理を中止します。

（2）脚注リソース型要素（footnote）

<link:footnote> 要素は，インスタンス文書のリソース型要素で，<link:footnoteLink> 要素の子要素として出現します。<link:footnote> 要素には，任意の文字列を要素の値として記述します。<link:footnote> 要素の属性を次の表に示します。

表 5-18 <link:footnote> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	resource 固定です。	必須
2	xlink:label	NCName 型データです。アーク型要素によって参照しているリソース型要素（<link:footnote>）を識別するために必要です。同一の値を持つことが許されます。	必須
3	xlink:role	任意の値を指定します。デフォルトは null です。	任意
4	xml:lang	任意の言語を特定するために使用します。ISO で定義されている言語コードである「ja」,「en」などを指定します。	必須
5	xlink:title	任意の値を指定します。デフォルトは null です。	任意

この形式以外のリソース型要素が出現した場合，エラーとして処理を中止します。

(3) 脚注アーク型要素（footnoteArc）

<link:footnoteArc> 要素は，インスタンス文書のアーク型要素で，<link:footnoteLink> 要素の子要素として出現します。<link:footnoteArc> 要素の属性を次の表に示します。

表 5-19 <link:footnoteArc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	arc 固定です。	必須
2	xlink:show	embed，または replace を指定します。	必須
3	xlink:actuate	onRequest 固定です。	必須
4	xlink:from	NCName 型データです。一つのロケータ型要素（<link:loc>），またはリソース型要素（<link:footnote>）の xlink:label 属性の内容と同じ値を指定します。	必須
5	xlink:to	NCName 型データです。一つのロケータ型要素（<link:loc>），またはリソース型要素（<link:footnote>）の xlink:label 属性の内容と同じ値を指定します。	必須
6	xlink:arcrole	次のどちらかの値を指定します。 • http://www.xbrl.org/linkprops/arc/fact-footnote • http://www.xbrl.org/linkprops/arc/footnote-fact	必須
7	xlink:title	任意の値を指定します。デフォルトは null です。	任意
8	use	optional，prohibited，または required を指定します。デフォルトは optional です。	任意
9	priority	use 属性の optional，prohibited，または required の関係に対してのプライオリティ（優先順位）を指定するための属性です。デフォルトは 0 です。	任意

この形式以外のアーク型要素が出現した場合，エラーとして処理を中止します。

5.3 取り扱えるタクソノミー文書（XBRL2.1 の場合）

uCosminexus Business Reporting Processor で取り扱えるタクソノミー文書の形式について説明します。

タクソノミー文書は，次のスキーマ定義に従って記述されている必要があります。

XBRL のスキーマ定義（XBRL2.1 の場合）

XBRL International が提供するスキーマ定義です。XBRL のバージョンによって異なります。

- http://www.xbrl.org/2003/xbrl-instance-2003-12-31.xsd
- http://www.xbrl.org/2003/xbrl-linkbase-2003-12-31.xsd
- http://www.xbrl.org/2003/xl-2003-12-31.xsd
- http://www.xbrl.org/2003/xlink-2003-12-31.xsd

XML Schema のスキーマ定義

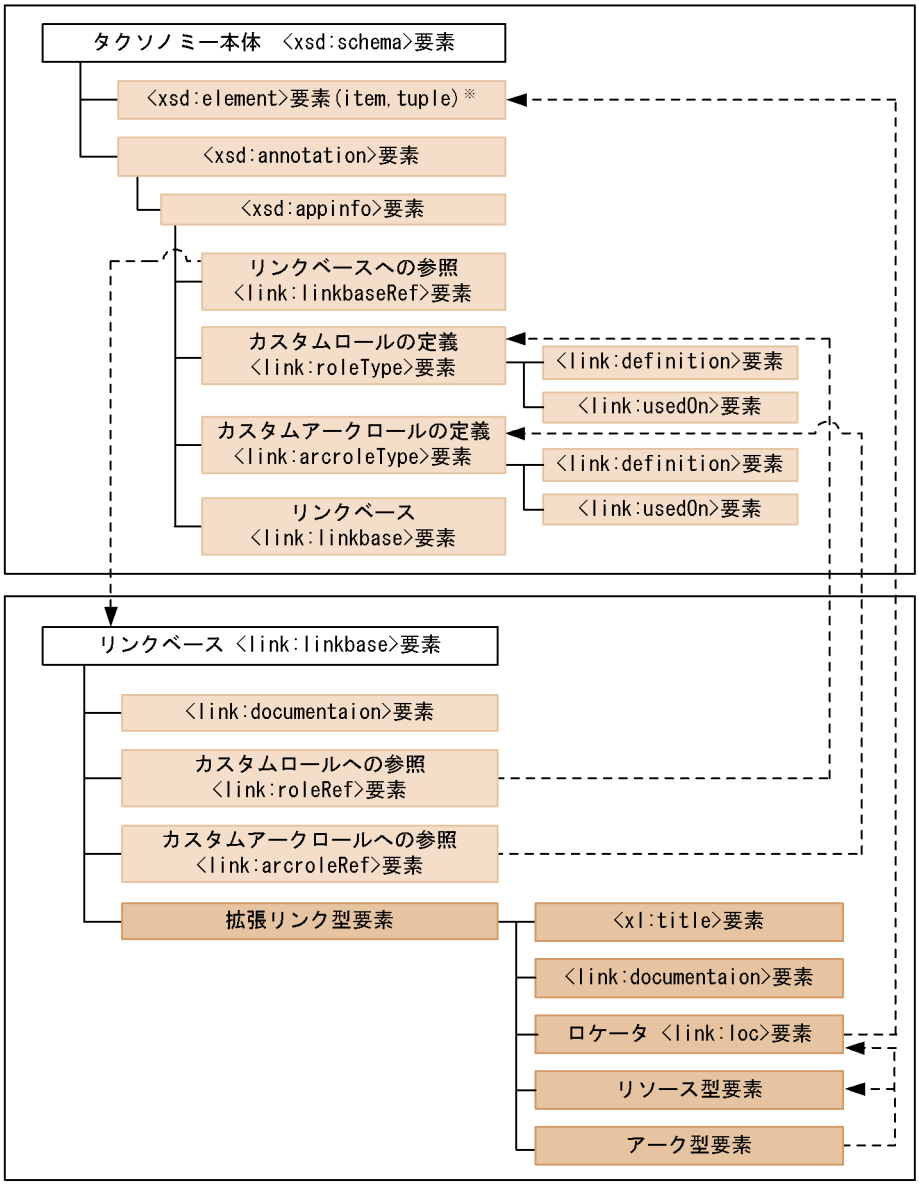
W3C が提供するスキーマ定義です。

- http://www.w3.org/2001/xml.xsd
- http://www.w3.org/2001/XMLSchema.xsd
- http://www.w3.org/2001/XMLSchema.dtd
- http://www.w3.org/2001/XMLSchema-datatypes.xsd
- http://www.w3.org/2001/datatypes.dtd

5.3.1 タクソノミー文書の基本構造

XBRL2.1 の場合の，タクソノミー文書の基本構造を次の図に示します。

図 5-3 タクソノミー文書の基本構造



(凡例) ----► : XLinkまたはXPointerによる参照関係

注※ XBRL Dimensions1.0の場合は，hypercube，dimensionの定義も含みます。

上記の図のうち，拡張型リンク要素の部分（リンクベース）には次があります。

- <link:presentationLink> 要素（プレゼンテーションリンク）
- <link:calculationLink> 要素（計算リンク）
- <link:definitionLink> 要素（定義リンク）

- <link:labelLink> 要素（ラベルリンク）
- <link:referenceLink> 要素（リファレンスリンク）

5.3.2 エレメント要素（element）

ここでは，XBRL2.1 の場合のタクソノミー文書で使用するエレメント要素について説明します。

タクソノミー文書では，XML スキーマの <xsd:element> 要素を使用して，インスタンス文書中に出現できる要素を定義します。エレメント要素では，アイテム型要素，タプル型要素などを定義します。

uCosminexus Business Reporting Processor は，<xsd:element> 要素が <xsd:schema> 要素の子要素として定義されている場合に，エレメント要素を処理します。また，uCosminexus Business Reporting Processor で扱うタクソノミー文書では属性定義が必要で，属性に指定する値も条件を満たす必要があります。uCosminexus Business Reporting Processor は，必須条件を満たす <xsd:element> 要素を処理対象とし，満たさない <xsd:element> 要素は無視します。

（1）アイテム型要素（item）

uCosminexus Business Reporting Processor では，次の条件を満たすアイテム型要素を使用できます。

表 5-20 エレメント要素の属性（アイテム型要素）

項番	属性名	内容	必須条件
1	name	名前属性です。内容は任意です。	必須
2	type	データ型属性です。XBRL データ型またはユーザー独自のデータ型を使用できます。 詳細は，表下の「取り扱えるデータ型」を参照してください。	任意
3	substitutionGroup	代替グループ属性です。総称名となる基本要素として xbrli:item を記述します。	必須
4	id	リンクベースによって関係づけられる場合で，かつ xlink:href 属性値によっては必須です。デフォルトは null です。	任意
5	nillable	インスタンス文書で，xsi:nil 属性を使用してニル（nil）であることを宣言できるかどうかを指定します。宣言する場合，true を指定します。指定できる値は，true または false です。 デフォルトは false です。	任意
6	xbrli:periodType	指定できる値は instant または duration です。	必須
7	xbrli:balance	指定できる値は debit または credit です。デフォルトは null です。	任意

取り扱えるデータ型

uCosminexus Business Reporting Processor は，type 属性が XBRL データ型である要素宣言を処理できます。認識できないデータ型は，ユーザー独自のデータ型（**ユーザーデータ型**）として解釈します。

ユーザーデータ型は，ユーザーが独自に定義する属性のことです。定義した属性は，XBRL DOM を介して取得できます。定義内容は，スキーマ定義に基づいて検証されます。

ユーザーデータ型を XBRL データ型から派生して作成した場合は，ユーザーデータ型が数値または非数値データ型かどうかが判断されます。それ以外の場合，ユーザーデータ型は文字列として保持・参照されます。XBRL2.1 の場合は，派生元のデータ型を取得することもできます。

なお，type 属性が指定されていない場合，substitutionGroup 属性によってアイテム型要素として処理され，ユーザーデータ型と同じくその他（OTHER_ITEM）として解釈されます。

取り扱えるユーザーデータ型の詳細については，「5.5　取り扱えるデータ型」を参照してください。

(2) タプル型要素（tuple）

uCosminexus Business Reporting Processor では，次の条件を満たすタプル型要素を使用できます。

表 5-21　エレメント要素の属性（タプル型要素）

項番	属性名	内容	必須条件
1	name	名前属性です。内容は任意です。	必須
2	type	データ型属性です。xbrli:tupleType を記述します。詳細は，表下の「type 属性の指定」を参照してください。	任意
3	substitutionGroup	代替グループ属性です。総称名となる基本要素として xbrli:tuple を記述します。	必須
4	nillable	インスタンス文書で，xsi:nil 属性を使用してニル（nil）であることを宣言できるかどうかを指定します。宣言する場合，true を指定します。指定できる値は，true または false です。 デフォルトは false です。	任意
5	id	リンクベースによって関係づけられる場合で，かつ xlink:href 属性値によっては必須です。デフォルトは null です。	任意
6	xbrli:balance	指定できる値は debit または credit です。デフォルトは null です。	任意

type 属性の指定

uCosminexus Business Reporting Processor では，substitutionGroup 属性が

xbrli:tuple の場合，type 属性がスキーマ定義どおりであるかを検証します。

なお，タクソノミー文書に対して通常の検証を実行した場合，タプル定義については次の内容を検証します。

- タプルが xbrli:tuple から派生していること
- タプルを宣言する要素に，xbrli:periodType 属性と xbrli:balance 属性が含まれていないこと

さらに，完全な妥当性検証を実行すれば，次の内容も検証できます。

- タプルが混合内容または単純な内容を持たないこと
  例えば，次のような宣言はエラーとなる。
  <xsd:complexContent mixed="true">
- タプルの子孫要素に anonymous 型（匿名）宣言を含まないこと
- タプル内要素の宣言が，xbrli:item または xbrli:tuple から派生した要素のグローバル要素であること
  例えば次の場合，element 要素が ref 属性で参照されていないのでエラーとなる。
  <xsd:element name="managementName" type="xbrli:tokenItemType"/>

（3）ハイパーキューブ型要素（hypercube）（XBRL Dimensions1.0 の場合）

uCosminexus Business Reporting Processor では，次の条件を満たすハイパーキューブ型要素を使用できます。

表 5-22　エレメント要素の属性（ハイパーキューブ型要素）

項番	属性名	内容	必須条件
1	abstract	true を指定します。	必須
2	name	名前属性です。内容は任意です。	必須
3	type	データ型属性です。XBRL データ型またはユーザー独自のデータ型を使用できます。 取り扱えるデータ型については，「5.3.2(1)　アイテム型要素（item）」を参照してください。	任意
4	substitutionGroup	代替グループ属性です。総称名となる基本要素として xbrldt:hypercubeItem を記述します。	必須
5	id	リンクベースによって関係づけられる場合で，かつ xlink:href 属性値によっては必須です。デフォルトは null です。	任意
6	nillable	インスタンス文書で，xsi:nil 属性を使用してニル（nil）であることを宣言できるかどうかを指定します。宣言する場合，true を指定します。指定できる値は，true または false です。 デフォルトは false です。	任意
7	xbrli:periodType	指定できる値は instant または duration です。	必須

項番	属性名	内容	必須条件
8	xbrli:balance	指定できる値は debit または credit です。デフォルトは null です。	任意

(4) ディメンジョン型要素（dimension）（XBRL Dimensions1.0 の場合）

uCosminexus Business Reporting Processor では，次の条件を満たすディメンジョン型要素を使用できます。

表 5-23　エレメント要素の属性（ディメンジョン型要素）

項番	属性名	内容	必須条件
1	abstract	true を指定します。	必須
2	name	名前属性です。内容は任意です。	必須
3	type	データ型属性です。XBRL データ型またはユーザー独自のデータ型を使用できます。 取り扱えるデータ型については，「5.3.2(1)　アイテム型要素（item）」を参照してください。	任意
4	substitutionGroup	代替グループ属性です。総称名となる基本要素として xbrldt:dimensionItem を記述します。	必須
5	id	リンクベースによって関係づけられる場合で，かつ xlink:href 属性値によっては必須です。デフォルトは null です。	任意
6	nillable	インスタンス文書で，xsi:nil 属性を使用してニル（nil）であることを宣言できるかどうかを指定します。宣言する場合，true を指定します。指定できる値は，true または false です。 デフォルトは false です。	任意
7	xbrli:periodType	指定できる値は instant または duration です。	必須
8	xbrli:balance	指定できる値は debit または credit です。デフォルトは null です。	任意
9	xbrldt:typedDomainRef	XPointer 式を指定します。 Typed Dimension の場合，必ずこの属性を指定してください。この属性の指定がない場合は，Explicit Dimension として認識されます。 なお，Typed Dimension の内容は，グローバルな要素宣言として定義してください。	任意

(5) そのほかのエレメント要素

uCosminexus Business Reporting Processor では，<xsd:schema> の子要素である <xsd:element> 要素を，アイテム型要素またはタプル型要素として一度検証します。

この検証でアイテム型要素またはタプル型要素と認識できない要素の場合，そのほかの要素として扱います。<xsd:schema> 要素の子要素でない <xsd:element> 要素や，

<xsd:element> 要素に name 属性を持たない要素については無視します。

そのほかの要素として扱う <xsd:element> 要素は次のとおりです。

- <xsd:schema> の子要素である <xsd:element> 要素
- name 属性を持ち，item 型または tuple 型の条件を満たさない要素
  （substitutionGroup 属性が xbrli:item または xbrli:tuple 以外の要素）

このため，そのほかの要素がリンクベースのトラバース先に指定されている場合でも，リンクベースを処理できます。

5.3.3 カスタムロール，カスタムアークロール

XBRL2.1 では，カスタムロールおよびカスタムアークロールを定義できます。

カスタムロールやカスタムアークロールを使用すると，リンクの役割を定義するための要素であるロールやアークロールを独自に定義できます。カスタムロールの定義には <link:roleType> 要素を，カスタムアークロールの定義には <link:arcroleType> 要素を使用します。

（1）カスタムロール（roleType）

<link:roleType> 要素では，xlink:role 属性に記述できる URI（カスタムロール）と，このカスタムロールを使用できる要素を定義します。

uCosminexus Business Reporting Processor は，<xsd:schema>/<xsd:annotation>/<xsd:appinfo> の下位に記述されている <link:roleType> 要素を処理します。

<link:roleType> 要素の属性を次の表に示します。

表 5-24 <link:roleType> 要素の属性

項番	属性名	内容	必須条件
1	roleURI	ロールを表す URI です。任意の URI を指定できます。空文字列は指定できません。	必須
2	id	ID 型の値です。インスタンス文書またはリンクベースの <link:roleRef> 要素によって関係づけられる場合で，xlink:href 属性値によっては必須です。デフォルトは null です。	任意

<link:roleType> 要素の子要素には，<link:definition> 要素と <link:usedOn> 要素があります。

- <link:definition> 要素
  <link:roleType> 要素の説明を記述します。uCosminexus Business Reporting Processor では，この要素は処理されません。また，XBRL DOM で操作する API もありません。

- <link:usedOn> 要素
  この要素は属性を持ちません。この要素の内容は xsd:QName 型データです。
  この要素には，カスタムロールを使用する要素名を記述します。例えば，<link:labelLink> 要素の xlink:role 属性で使用する場合は，link:labelLink となります。
  <link:usedOn> 要素は，<link:roleType> 要素の子要素として複数定義できます。しかし，同じ値を持つ要素の二重定義はできません。

(2) カスタムアークロール（arcroleType）

<link:arcroleType> 要素では，xlink:arcrole 属性に記述できる URI（カスタムアークロール）と，このカスタムアークロールを使用できる要素を定義します。

uCosminexus Business Reporting Processor は，<xsd:schema>/<xsd:annotation>/<xsd:appinfo> の下位に記述されている <link:arcroleType> 要素を処理します。

<link:arcroleType> 要素の属性を次の表に示します。

表 5-25 <link:arcroleType> 要素の属性

項番	属性名	内容	必須条件
1	arcroleURI	カスタムアークロールを表す URI です。任意の URI を指定できます。空文字列は指定できません。	必須
2	cyclesAllowed	リンクのネットワークで許可されるサイクルのタイプを指定します。指定できる値は，any，undirected，または none です。	必須
3	id	ID 型の値です。インスタンス文書またはリンクベースの <link:arcroleRef> 要素によって関係づけられる場合で，xlink:href 属性値によっては必須です。デフォルトは null です。	任意

<link:arcroleType> 要素の子要素には，<link:definition> 要素と <link:usedOn> 要素があります。

- <link:definition> 要素
  <link:arcroleType> 要素の説明を記述します。uCosminexus Business Reporting Processor では，この要素は処理されません。また，XBRL DOM でこの要素を操作する API もありません。
- <link:usedOn> 要素
  この要素は属性を持ちません。この要素の内容は xsd:QName 型データです。
  この要素には，カスタムアークロールを使用する要素名を記述します。例えば，<link:labelArc> 要素の xlink:arcrole 属性で使用する場合は，link:labelArc となります。
  <link:usedOn> 要素は，<link:arcroleType> 要素の子要素として複数定義できます。しかし，同じ値を持つ要素の二重定義はできません。

5.3.4 リンクベースへの参照定義

タクソノミー本体からリンクベースへの参照定義には，<link:linkbaseRef> 要素を使用します。

（1） <link:linkbaseRef> 要素

uCosminexus Business Reporting Processor は，<xsd:schema>/<xsd:annotation>/<xsd:appinfo> の下位に記述されている <link:linkbaseRef> 要素を処理します。

uCosminexus Business Reporting Processor で扱うタクソノミー文書では，次に示す属性定義が必要です。

表 5-26 <link:linkbaseRef> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	simple（単純リンク）固定です。	必須
2	xlink:href	拡張リンク型要素の定義ファイルへの URI またはローカルならファイル名を記述します。URI のフラグメントには XPointer 式を記述できます。値は任意です。	必須
3	xlink:role	リンク種別を意味する文字列です。空文字列は指定できません。デフォルトは null です。詳細は，表下の「xlink:role 属性に指定する値」を参照してください。	任意
4	xlink:arcrole	http://www.w3.org/1999/xlink/properties/linkbase 固定です。そのほかの値を指定した場合は，文書解析時にエラーになります。	必須
5	xlink:actuate	指定できる値は，onLoad，onRequest，other，または none です。属性を省略できます。	任意
6	xlink:show	指定できる値は，new，replace，embed，other，または none です。属性を省略できます。	任意
7	xlink:title	任意の文字列を設定します。	任意
8	xml:base	基底 URI を指定します。デフォルトは null です。	任意

出現位置に記述され処理対象となった <link:linkbaseRef> 要素が，この表で説明した属性の必須条件を満たしていない場合は，エラーとなり処理が中止されます。

xlink:role 属性に指定する値

xlink:role 属性を指定する場合，リンク種別ごとに異なる標準値を使用します。

xlink:role 属性のリンク種別ごとの標準値を次の表に示します。

表 5-27 xlink:role 属性のリンク種別ごとの標準値

項番	拡張リンク種別	linkbaseRef 要素の xlink:role 属性の標準値
1	プレゼンテーションリンク	http://www.xbrl.org/2003/role/presentationLinkbaseRef
2	計算リンク	http://www.xbrl.org/2003/role/calculationLinkbaseRef

項番	拡張リンク種別	linkbaseRef 要素の xlink:role 属性の標準値
3	定義リンク	http://www.xbrl.org/2003/role/definitionLinkbaseRef
4	ラベルリンク	http://www.xbrl.org/2003/role/labelLinkbaseRef
5	リファレンスリンク	http://www.xbrl.org/2003/role/referenceLinkbaseRef
6	任意	任意の拡張リンク型要素

uCosminexus Business Reporting Processor では，この標準値を用いてリンク種別を判断します。xlink:role 属性が省略されていた場合は，リンク先をトラバースして解析し，既存の拡張リンクであれば DTS に含めます。任意の値の場合は，警告と出力してトラバースしません。

5.3.5 リンクベース要素（linkbase）

タクソノミー文書は，インスタンス文書中に出現できる要素宣言のほかに，各要素の関係を示すリンクベースを記述します。ここでは，XBRL2.1 の場合のリンクベースで使用する要素について説明します。

（1） <link:linkbase> 要素

<link:linkbase> 要素は，タクソノミー文書で使用される拡張リンク型要素などを記述するためのコンテナ要素として利用します。なお，XBRL2.1 では，<link:linkbaseRef> 要素を記述できません。

<link:linkbase> 要素が文書のルート要素となる場合には，xmlns 属性や xsi:schemaLocation 属性などで，スキーマ検証で使用するタクソノミー文書を指定する必要があります。また，タクソノミー本体に <link:linkbase> 要素を記述することもできます。タクソノミー本体に記述する場合には，<xsd:schema>/<xsd:annotation>/<xsd:appinfo> の下に記述された <link:linkbase> 要素を処理します。

<link:linkbase> 要素の属性を次の表に示します。

表 5-28 <link:linkbase> 要素の属性

項番	属性名	内容	必須条件
1	id	ID 型の値です。<link:linkbaseRef> 要素によって関係づけられる場合で，xlink:href 属性値によっては必須です。	任意
2	xml:base	基底 URI を指定します。	任意

（2） <link:documentation> 要素

uCosminexus Business Reporting Processor では，この要素は処理されません。また，XBRL DOM でこの要素を操作する API もありません。

(3) カスタムロール，カスタムアークロールへの参照定義

リンクベースでは，カスタムロール，カスタムアークロールを参照できます。カスタムロールへの参照には <link:roleRef> 要素を，カスタムアークロールへの参照には <link:arcroleRef> 要素を使用します。

(a) <link:roleRef> 要素

<link:roleRef> 要素は，XBRL2.1 で新しく追加された要素です。単純リンク型の要素であり，リンクベース内の各要素の xlink:role 属性値で使用できるカスタムロールを参照するために使用します。

uCosminexus Business Reporting Processor は，<link:linkbase> 下に記述されている <link:roleRef> 要素を処理します。<link:roleRef> 要素で参照されるタクソノミー本体は DTS に含まれます。また，そのタクソノミー本体で，<xsd:import> 要素および <xsd:include> 要素で取り込まれるタクソノミー本体も，すべて DTS に含まれます。

uCosminexus Business Reporting Processor では，次の条件を満たす <link:roleRef> 要素を使用できます。

表 5-29 <link:roleRef> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	simple（単純リンク）固定です。	必須
2	xlink:href	カスタムロール定義（<link:roleType> 要素）への参照 URI を記述します。URI のフラグメントには，XPointer 式を記述できます。任意の値を指定できます。	必須
3	xlink:role	リンク種別を意味する URI です。空文字列は指定できません。	任意
4	xlink:arcrole	任意の URI です。空文字列は指定できません。	任意
5	xlink:actuate	指定できる値は，onLoad，onRequest，other，または none です。属性を省略できます。	任意
6	xlink:show	指定できる値は，new，replace，embed，other，または none です。属性を省略できます。	任意
7	xlink:title	任意の文字列を指定します。	任意
8	xml:base	基底 URI を指定します。デフォルトは null です。	任意
9	roleURI	参照するカスタムロール定義のカスタムロールの値（<link:roleType> 要素の roleURI 属性値）を指定します。	必須

(b) <link:arcroleRef> 要素

<link:arcroleRef> 要素は，XBRL2.1 で新しく追加された要素です。単純リンク型の要素であり，リンクベース内の各要素の xlink:arcrole 属性値で使用できるカスタムアーク

ロールを参照するために使用します。

uCosminexus Business Reporting Processor は，<link:linkbase> 下に記述されている <link:linkbaseRef> 要素を処理します。<link:arcroleRef> 要素で参照されるタクソノミー本体は DTS に含まれます。また，そのタクソノミー本体で，<xsd:import> 要素および <xsd:include> 要素で取り込まれるタクソノミー本体も，すべて DTS に含まれます。

uCosminexus Business Reporting Processor では，次の条件を満たす <link:arcroleRef> 要素を使用できます。

表 5-30　<link:arcroleRef> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	simple（単純リンク）固定です。	必須
2	xlink:href	カスタムアークロール定義（<link:arcroleType> 要素）への参照 URI を記述します。URI のフラグメントには，XPointer 式を記述できます。任意の値を指定できます。	必須
3	xlink:role	リンク種別を意味する URI です。空文字列は指定できません。	任意
4	xlink:arcrole	任意の URI です。空文字列は指定できません。	任意
5	xlink:actuate	指定できる値は，onLoad，onRequest，other，または none です。属性を省略できます。	任意
6	xlink:show	指定できる値は，new，replace，embed，other，または none です。属性を省略できます。	任意
7	xlink:title	任意の文字列を指定します。	任意
8	xml:base	基底 URI を指定します。デフォルトは null です。	任意
9	arcroleURI	参照するカスタムアークロール定義のカスタムアークロールの値（<link:arcroleType> 要素の arcroleURI 属性値）を指定します。	必須

(4) 拡張リンク型要素（extend link-type）

タクソノミー文書で使用される拡張リンクのリンクベースは，「5.3.4(1) <link:linkbaseRef> 要素」の xlink:href 属性に記述されたファイルまたは URI に対応する実体です。

uCosminexus Business Reporting Processor では，<link:linkbase> 要素の子要素として記述された拡張リンク型要素を処理します。また，タクソノミー文書に記述するどの拡張リンク型要素にも，次の属性定義が必要です。

表 5-31 拡張リンク型要素の共通属性

項番	属性名	内容	必須条件
1	xlink:type	extended（拡張リンク）固定です。	必須
2	xlink:role	リンク種別を意味する文字列です。空文字列は指定できません。 標準値は http://www.xbrl.org/2003/role/link です。	必須
3	xlink:title	任意の文字列を設定します。デフォルトは null です。	任意
4	id	ID 型の値です。<link:linkbaseRef> 要素によって関係づけられる場合で，xlink:href 属性値によっては必須です。デフォルトは null です。	任意
5	xml:base	基底 URI を指定します。デフォルトは null です。	任意

拡張リンク型要素は，アーク型要素，ロケータ型要素およびリソース型要素のコンテナ要素となり，リンク種別ごとに子要素として含むことができる要素が異なります。

次に，関係リンク，ラベルリンクおよびリファレンスリンクで含むことができる要素について示します。

(a) 関係リンク

関係リンクで含むことができる要素について示します。なお，完全な妥当性検証では，関係ネットワーク上でリンクが循環していないかを検証できます。詳細については「2.3.2(6) 関係ネットワーク上のサイクルの検証【完全な妥当性検証の場合だけ有効】」を参照してください。

- プレゼンテーションリンク
  <link:presentationLink> 要素では，アイテム間の表示順を定義します。
  <link:documentation> 要素，<xl:title> 要素，<link:loc> 要素，および
  <link:presentationArc> 要素を含むことができます。
- 計算リンク
  <link:calculationLink> 要素では，アイテムの数値データの重み付き加減算式を定義します。
  <link:documentation> 要素，<xl:title> 要素，<link:loc> 要素，および
  <link:calculationArc> 要素を含むことができます。
- 定義リンク
  <link:definitionLink> 要素では，アイテム間の親子関係など，アイテム同士の関係を定義します。
  <link:documentation> 要素，<xl:title> 要素，<link:loc> 要素，および
  <link:definitionArc> 要素を含むことができます。

(b) ラベルリンク（labelLink）

<link:labelLink> 要素では，アイテムの表示名称（ラベル）を定義します。日本語，英語などさまざまな言語でラベルを記述したり，別名を定義したりします。

<link:documentation> 要素，<xl:title> 要素，<link:loc> 要素，<link:label> 要素，および <link:labelArc> 要素を含むことができます。

(c) リファレンスリンク（referenceLink）

<link:referenceLink> 要素では，参考文献を定義します。会計概念を定義する根拠となっている文献への参照などに用います。

<link:documentation> 要素，<xl:title> 要素，<link:loc> 要素，<link:reference> 要素，および <link:referenceArc> 要素を含むことができます。

(5) <link:documentation> 要素

拡張リンク型要素の子要素です。uCosminexus Business Reporting Processor では，この要素は処理されません。また，XBRL DOM でこの要素を操作する API もありません。

(6) <xl:title> 要素

拡張リンク型要素の子要素です。uCosminexus Business Reporting Processor では，この要素は処理されません。また，XBRL DOM でこの要素を操作する API もありません。

(7) <link:loc> 要素

<link:loc> 要素は，拡張リンクでのロケータ型要素で，タクソノミー文書では <link:calculationLink> 要素などの拡張リンク型の子要素です。<link:loc> 要素の属性を次の表に示します。

表 5-32 <link:loc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	locator 固定です。	必須
2	xlink:href	アイテム型要素，タプル型要素，またはリソース型要素への参照 URI を記述します。任意の値を指定できます。また，URI のフラグメントには，XPointer 式を記述できます。 XPointer 形式は，ショートハンドポインタ（Shorthand Pointer）形式，または XPointer element() scheme です。	必須
3	xlink:label	NCName 型データです。アーク型要素によって参照しているロケータ型要素（<link:loc>）を識別するために必要です。	必須
4	xlink:role	任意の URI です。<link:roleType> 要素との関連はありません。デフォルトは null です。	任意
5	xlink:title	任意の値です。デフォルトは null です。	任意

この形式以外のロケータ型要素が出現した場合，エラーとして処理を中止します。

(8) <link:label> 要素

タクソノミー文書での，リソース型要素の一つで，<link:labelLink> 要素の子要素です。<link:label> 要素の属性を次の表に示します。

表 5-33 <link:label> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	resource 固定です。	必須
2	xlink:label	NCName 型データです。アーク型要素によって参照しているリソース型要素（label）を識別するために必要です。同一の値を持つことが許されます。	必須
3	xlink:role	カスタムロール値または標準値を指定します。デフォルトは null です。標準値については，表下の「xlink:role 属性に指定する値」を参照してください。	任意
4	xml:lang	任意の言語を特定するために使用します。ISO で定義されている言語コードである「ja」,「en」などです。	必須
5	xlink:title	任意の値です。デフォルトは null です。	任意
6	id	ID 型の値です。デフォルトは null です。	任意

<link:label> 要素には，任意の文字列を要素の値として記述します。また，この形式以外のリソース型要素が出現した場合，エラーとして処理を中止します。

xlink:role 属性に指定する値

<link:label> 要素の xlink:role 属性の標準値を次に示します。

- http://www.xbrl.org/2003/role/label
- http://www.xbrl.org/2003/role/terseLabel
- http://www.xbrl.org/2003/role/verboseLabel
- http://www.xbrl.org/2003/role/positiveLabel
- http://www.xbrl.org/2003/role/positiveTerseLabel
- http://www.xbrl.org/2003/role/positiveVerboseLabel
- http://www.xbrl.org/2003/role/negativeLabel
- http://www.xbrl.org/2003/role/negativeTerseLabel
- http://www.xbrl.org/2003/role/negativeVerboseLabel
- http://www.xbrl.org/2003/role/zeroLabel
- http://www.xbrl.org/2003/role/zeroTerseLabel
- http://www.xbrl.org/2003/role/zeroVerboseLabel
- http://www.xbrl.org/2003/role/totalLabel
- http://www.xbrl.org/2003/role/periodStartLabel
- http://www.xbrl.org/2003/role/periodEndLabel
- http://www.xbrl.org/2003/role/documentation
- http://www.xbrl.org/2003/role/definitionGuidance

- http://www.xbrl.org/2003/role/disclosureGuidance
- http://www.xbrl.org/2003/role/presentationGuidance
- http://www.xbrl.org/2003/role/measurementGuidance
- http://www.xbrl.org/2003/role/commentaryGuidance
- http://www.xbrl.org/2003/role/exampleGuidance

(9) <link:reference> 要素

タクソノミー文書での，リソース型要素の一つで，<link:referenceLink> 要素の子要素です。<link:reference> 要素の属性を次の表に示します。

表 5-34 <link:reference> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	resource 固定です。	必須
2	xlink:label	NCName 型データです。アーク型要素によって参照しているリソース型要素（<link:reference>）を識別するために必要です。同一の値を持つことが許されます。	必須
3	xlink:role	カスタムロール値または標準値を指定します。デフォルトは null です。標準値については，表下の「xlink:role 属性に指定する値」を参照してください。	任意
4	xlink:title	任意の値です。デフォルトは null です。	任意
5	id	ID 型の値です。デフォルトは null です。	任意

<link:reference> 要素には，任意の子要素を含むことができます。この任意要素を総称し，抽象的な要素表現である part（パート）と呼びます。

この形式以外のリソース型要素が出現した場合，エラーとして処理を中止します。

xlink:role 属性に指定する値

<link:reference> 要素の xlink:role 属性の標準値を次に示します。

- http://www.xbrl.org/2003/role/reference
- http://www.xbrl.org/2003/role/definitionRef
- http://www.xbrl.org/2003/role/disclosureRef
- http://www.xbrl.org/2003/role/mandatoryDisclosureRef
- http://www.xbrl.org/2003/role/recommendedDisclosureRef
- http://www.xbrl.org/2003/role/unspecifiedDisclosureRef
- http://www.xbrl.org/2003/role/presentationRef
- http://www.xbrl.org/2003/role/measurementRef
- http://www.xbrl.org/2003/role/commentaryRef
- http://www.xbrl.org/2003/role/exampleRef

（10）<link:presentationArc> 要素

<link:presentationArc> 要素は，タクソノミー文書のアーク型要素の一つで，<link:presentationLink> 要素の子要素です。<link:presentationArc> 要素の属性を次の表に示します。

表 5-35　<link:presentationArc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	arc 固定です。	必須
2	xlink:show	指定できる値は，new，replace，embed，other，または none です。属性を省略できます。	任意
3	xlink:actuate	指定できる値は，onLoad，onRequest，other，または none です。属性を省略できます。	任意
4	xlink:from	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
5	xlink:to	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
6	xlink:arcrole	カスタムアークロール値または標準値を指定します。 標準値は http://www.xbrl.org/2003/arcrole/parent-child です。	必須
7	xlink:title	任意の値を指定します。デフォルトは null です。	任意
8	use	optional または prohibited を指定します。デフォルトは optional です。	任意
9	priority	use 属性の optional または prohibited の関係に対してのプライオリティ（優先順位）を指定するための属性です。デフォルトは 0 です。	任意
10	order	複数の兄弟要素間での表示順序を示す 10 進数を指定します。デフォルトは 1 です。	任意
11	preferredLabel	任意の URI を指定します。デフォルトは null です。	任意

この形式以外のアーク型要素が出現した場合，エラーとして処理を中止します。

（11）<link:calculationArc> 要素

<link:calculationArc> 要素は，タクソノミー文書のアーク型要素の一つで，<link:calculationLink> 要素の子要素です。<link:calculationArc> 要素の属性を次の表に示します。

表 5-36　<link:calculationArc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	arc 固定です。	必須

項番	属性名	内容	必須条件
2	xlink:show	指定できる値は，new，replace，embed，other，または none です。属性を省略できます。	任意
3	xlink:actuate	指定できる値は，onLoad，onRequest，other，または none です。属性を省略できます。	任意
4	xlink:from	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
5	xlink:to	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
6	xlink:arcrole	カスタムアークロール値または標準値を指定します。 標準値は http://www.xbrl.org/2003/arcrole/summation-item です。	必須
7	xlink:title	任意の値を指定します。デフォルトは null です。	任意
8	use	optional または prohibited を指定します。デフォルトは optional です。	任意
9	priority	use 属性の optional または prohibited の関係に対してのプライオリティ（優先順位）を指定するための属性です。デフォルトは 0 です。	任意
10	order	複数の兄弟要素間での表示順序を示す 10 進数を指定します。デフォルトは 1 です。	任意
11	weight	子要素から親要素に数値を加算するときに，その要素の値に適用する乗数を示します。値「1.0」はその要素の数値を 100% 親要素に加算し，「-1.0」は 100% 減算することを表します。0 は指定できません。	必須

この形式以外のアーク型要素が出現した場合，エラーとして処理を中止します。

(12) <link:definitionArc> 要素

<link:definitionArc> 要素は，タクソノミー文書のアーク型要素の一つで，<link:definitionLink> 要素の子要素として出現します。<link:definitionArc> 要素の属性を次の表に示します。

表 5-37 <link:definitionArc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	arc 固定です。	必須
2	xlink:show	指定できる値は，new，replace，embed，other，または none です。属性を省略できます。	任意
3	xlink:actuate	指定できる値は，onLoad，onRequest，other，または none です。属性を省略できます。	任意

項番	属性名	内容	必須条件
4	xlink:from	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
5	xlink:to	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
6	xlink:arcrole	カスタムアークロール値または標準値を指定します。 標準値は次のとおりです。 • http://www.xbrl.org/2003/arcrole/general-special • http://www.xbrl.org/2003/arcrole/essence-alias • http://www.xbrl.org/2003/arcrole/similar-tuples • http://www.xbrl.org/2003/arcrole/requires-element	必須
7	xlink:title	任意の値を指定します。デフォルトは null です。	任意
8	use	optional または prohibited を指定します。デフォルトは optional です。	任意
9	priority	use 属性の optional または prohibited の関係に対してのプライオリティ（優先順位）を指定するための属性です。デフォルトは 0 です。	任意
10	order	複数の兄弟要素間での表示順序を示す 10 進数を指定します。デフォルトは 1 です。	任意
11	xbrldt:contextElement ※	segment または scenario を指定します。デフォルトは null です。	任意
12	xbrldt:targetRole ※	<difinitionLink> 要素の xlink:role 属性値です。デフォルトは null です。	任意
13	xbrldt:closed ※	true または false を指定します。デフォルトは false です。	任意
14	xbrldt:usable ※	true または false を指定します。デフォルトは true です。	任意

この形式以外のアーク型要素が出現した場合，エラーとして処理を中止します。

注※　XBRL Dimensions1.0 の場合だけ指定できます。また，xlink:arcrole 属性値に指定した値によってこの属性が解析対象となるかどうか，およびこの属性の指定が必須になるか任意になるかどうかが異なります。xlink:arcrole 属性値と <link:definitionArc> 要素の属性の関係を次の表に示します。

表 5-38　xlink:arcrole 属性値と <link:definitionArc> 要素の属性の関係

項番	xlink:arcrole 属性値	<link:definitionArc> 要素の属性			
		xbrldt:contextElement	xbrldt:targetRole	xbrldt:closed	xbrldt:usable
1	http://xbrl.org/int/dim/arcrole/all	○	△	△	×
2	http://xbrl.org/int/dim/arcrole/notAll	○	△	△	×

項番	xlink:arcrole 属性値	<link:definitionArc> 要素の属性			
		xbrldt:contextElement	xbrldt:targetRole	xbrldt:closed	xbrldt:usable
3	http://xbrl.org/int/dim/arcrole/hypercube-dimension	×	△	×	×
4	http://xbrl.org/int/dim/arcrole/dimension-domain	×	△	×	△
5	http://xbrl.org/int/dim/arcrole/dimension-default	×	×	×	×
6	http://xbrl.org/int/dim/arcrole/domain-member	×	△	×	△

（凡例）○：解析対象となり，指定は必須です。　△：解析対象となりますが，指定は任意です。
×：解析対象とはなりません。また，属性を指定した場合，指定した値は無視されます。

（13）<link:labelArc> 要素

<link:labelArc> 要素は，タクソノミー文書のアーク型要素の一つで，<link:labelLink> 要素の子要素です。<link:labelArc> 要素の属性を次の表に示します。

表 5-39　<link:labelArc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	arc 固定です。	必須
2	xlink:show	指定できる値は，new，replace，embed，other，または none です。属性を省略できます。	任意
3	xlink:actuate	指定できる値は，onLoad，onRequest，other，または none です。属性を省略できます。	任意
4	xlink:from	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
5	xlink:to	NCName 型データです。一つのロケータ型要素（<link:loc>）またはリソース型要素（<link:label>）の xlink:label 属性の内容と同じ値を指定します。	必須
6	xlink:arcrole	カスタムアークロール値または標準値を指定します。 標準値は http://www.xbrl.org/2003/arcrole/concept-label です。	必須
7	xlink:title	任意の値を指定します。デフォルトは null です。	任意
8	use	optional または prohibited を指定します。デフォルトは optional です。	任意
9	priority	use 属性の optional または prohibited の関係に対してのプライオリティ（優先順位）を指定するための属性です。デフォルトは 0 です。	任意
10	order	複数の兄弟要素間での表示順序を示す 10 進数を指定します。デフォルトは 1 です。	任意

この形式以外のアーク型要素が出現した場合，エラーとして処理を中止します。

（14）<link:referenceArc> 要素

<link:referenceArc> 要素は，タクソノミー文書のアーク型要素の一つで，<link:referenceLink> 要素の子要素です。<link:referenceArc> 要素の属性を次の表に示します。

表 5-40　<link:referenceArc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	arc 固定です。	必須
2	xlink:show	指定できる値は，new，replace，embed，other，または none です。属性を省略できます。	任意
3	xlink:actuate	指定できる値は，onLoad，onRequest，other，または none です。属性を省略できます。	任意
4	xlink:from	NCName 型データです。一つのロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
5	xlink:to	NCName 型データです。一つのロケータ型要素（<link:loc>）またはリソース型要素（<link:reference>）の xlink:label 属性の内容と同じ値を指定します。	必須
6	xlink:arcrole	カスタムアークロール値または標準値を指定します。 標準値は http://www.xbrl.org/2003/arcrole/concept-reference です。	必須
7	xlink:title	任意の値を指定します。デフォルトは null です。	任意
8	use	optional または prohibited を指定します。デフォルトは optional です。	任意
9	priority	use 属性の optional または prohibited の関係に対してのプライオリティ（優先順位）を指定するための属性です。デフォルトは 0 です。	任意
10	order	複数の兄弟要素間での表示順序を示す 10 進数を指定します。デフォルトは 1 です。	任意

この形式以外のアーク型要素が出現した場合，エラーとして処理を中止します。

5.4　取り扱えるインスタンス文書（XBRL2.1 の場合）

uCosminexus Business Reporting Processor で取り扱えるインスタンス文書の形式について説明します。ここでは，文書の基本構造，および定義できる要素やその属性値などについて説明します。

インスタンス文書は，次のスキーマに従って記述されている必要があります。

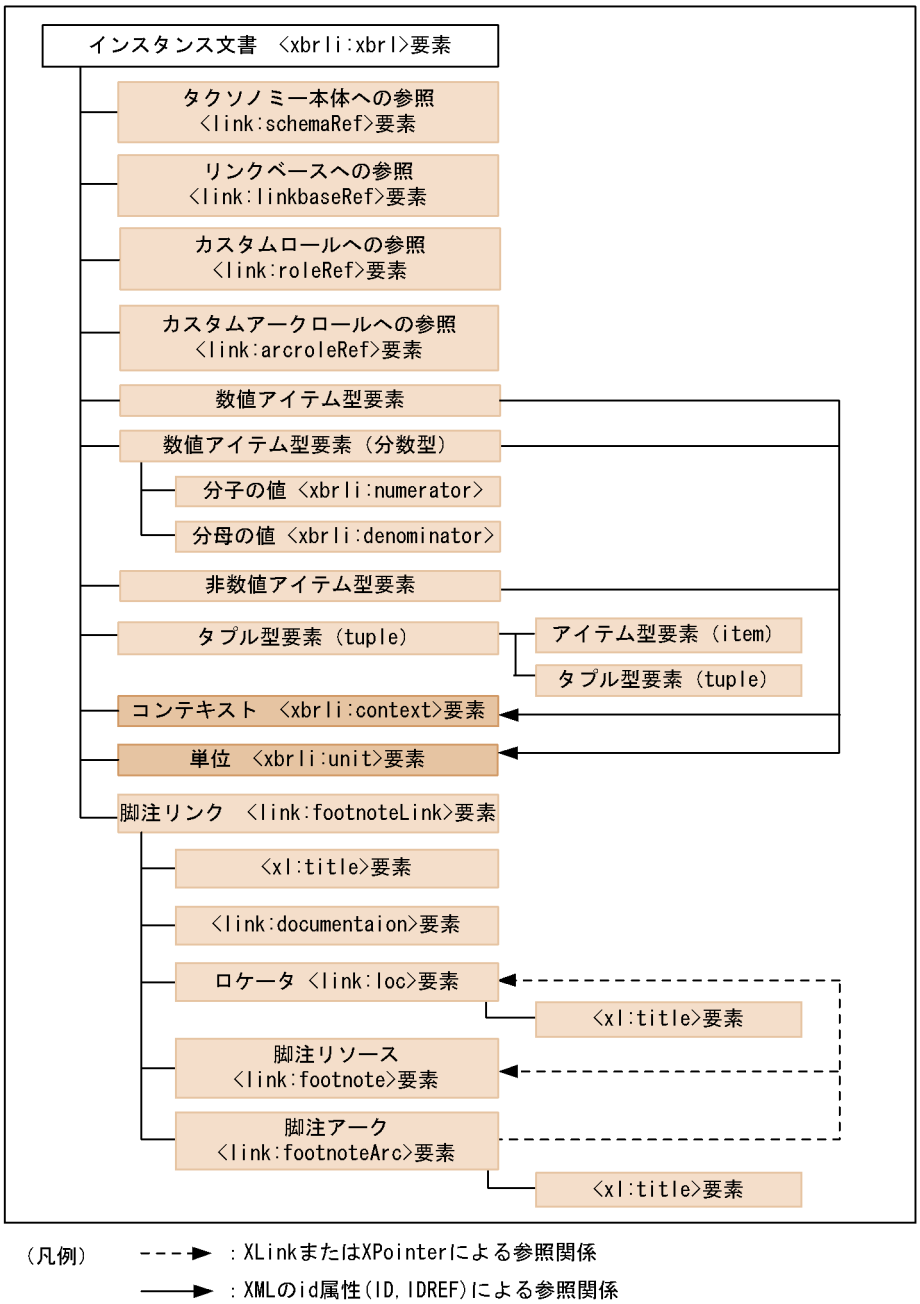
- タクソノミー文書（ユーザー定義スキーマ）
- XBRL のスキーマ定義
- XML Schema のスキーマ定義

XBRL のスキーマ定義および XML Schema のスキーマ定義の詳細については，「5.3　取り扱えるタクソノミー文書（XBRL2.1 の場合）」を参照してください。

5.4.1　インスタンス文書の基本構造

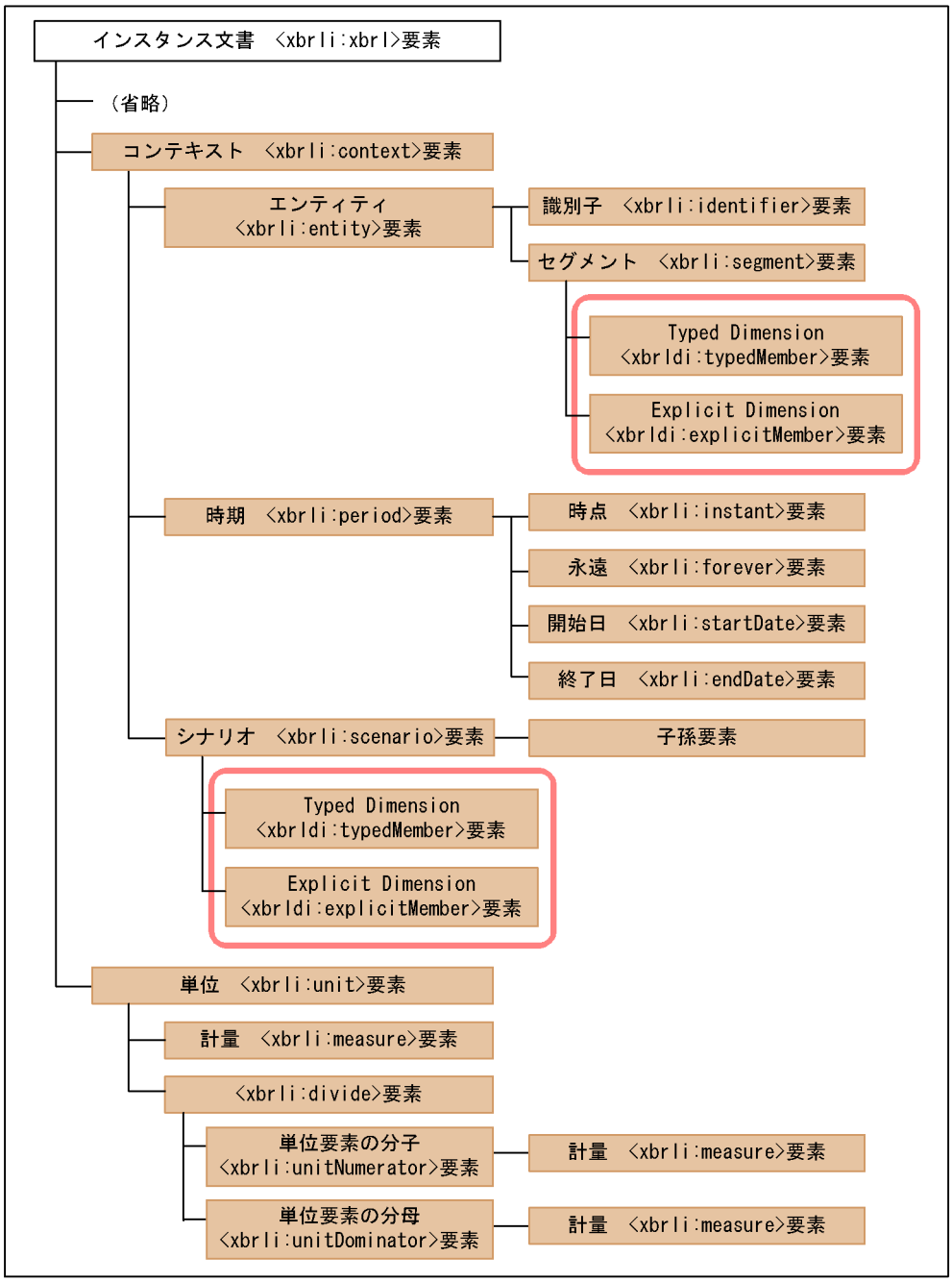
XBRL2.1 の場合の，インスタンス文書の基本構造を次の図に示します。

図 5-4　インスタンス文書の基本構造



上記の図のうち，<xbrli:context> 要素および <xbrli:unit> 要素の構造を次の図に示します。

図 5-5 <xbrli:context> 要素および <xbrli:unit> 要素の構造


インスタンス文書 <xbrli:xbrl>要素
（省略）
コンテキスト <xbrli:context>要素
エンティティ
<xbrli:entity>要素
識別子 <xbrli:identifier>要素
セグメント 要素
Typed Dimension
<xbrldi:typedMember>要素
Explicit Dimension
<xbrldi:explicitMember>要素
時期 <xbrli:period>要素
時点 <xbrli:instant>要素
永遠 <xbrli:forever>要素
開始日 <xbrli:startDate>要素
終了日 <xbrli:endDate>要素
シナリオ <xbrli:scenario>要素
子孫要素
Typed Dimension
<xbrldi:typedMember>要素
Explicit Dimension
<xbrldi:explicitMember>要素
単位 <xbrli:unit>要素
計量 <xbrli:measure>要素
<xbrli:divide>要素
単位要素の分子
<xbrli:unitNumerator>要素
計量 <xbrli:measure>要素
単位要素の分母
<xbrli:unitDominator>要素
計量 <xbrli:measure>要素

（凡例）

：XBRL Dimensions1.0の場合の構造を表します。

5.4.2 ルート要素（xbrl）

XBRL2.0 の仕様では，ルート要素は <xbrli:group> 要素でしたが，XBRL2.1 の仕様では，ルート要素は <xbrli:xbrl> 要素に変更されました。XBRL2.1 の場合，<xbrli:group> 要素は使用できません。

<xbrli:xbrl> 要素を XML 文書の任意の位置に記述したり，複数のインスタンス文書を一つの XML 文書に記述したりできます。uCosminexus Business Reporting Processor では，このような XML 文書を使用できます。

5.4.3 タクソノミー文書への参照定義

インスタンス文書からタクソノミー本体またはリンクベースを参照するための，要素や属性について説明します。それぞれ，次の要素について説明します。

タクソノミー本体への参照定義

- xsi:schemaLocation 属性
- <link:schemaRef> 要素

リンクベースへの参照定義

- <link:linkbaseRef> 要素

（1） xsi:schemaLocation 属性

タクソノミー本体への参照定義をする属性です。

XBRL2.0 の場合，DTS を自動的に読み込むためにルート要素である <xbrli:group> 要素の xsi:schemaLocation 属性の内容を参照していました。XBRL2.1 の場合，DTS を自動的に見つけるための要素として <link:schemaRef> 要素が追加されたため，uCosminexus Business Reporting Processor は，xsi:schemaLocation 属性の内容を確認しません。

ただし，インスタンス文書を検証するためには，xsi:schemaLocation 属性は必須です。

（2） <link:schemaRef> 要素

タクソノミー本体への参照定義をする要素で，XBRL2.1 で新しく追加されました。DTS を自動的に読み込むために参照されます。

<link:schemaRef> 要素は，インスタンス文書に一つ以上定義する必要があります。複数指定されていた場合，指定されているタクソノミー本体すべてが DTS に含まれます。また，<link:schemaRef> 要素で参照されるタクソノミー本体で，<xsd:import> 要素，および <xsd:include> 要素を使用して取り込まれるタクソノミー本体もすべて DTS に含まれます。

uCosminexus Business Reporting Processor では，<xbrli:xbrl> 要素下に記述されてい

る <link:schemaRef> 要素を処理します。

<link:schemaRef> 要素の属性を次の表に示します。

表 5-41 <link:schemaRef> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	simple（単純リンク）固定です。	必須
2	xlink:href	タクソノミー本体への参照 URI を指定します。URI のフラグメントには，XPointer 式を記述できます。任意の値を指定できます。	必須
3	xlink:role	リンク種別を意味する文字列です。空文字列は指定できません。デフォルトは null です。	任意
4	xlink:arcrole	任意の文字列を指定します。空文字列は指定できません。	任意
5	xlink:actuate	指定できる値は，onLoad，onRequest，other，または none です。属性を省略できます。	任意
6	xlink:show	指定できる値は，new，replace，embed，other，または none です。属性を省略できます。	任意
7	xlink:title	任意の文字列を指定します。	任意
8	xml:base	基底 URI を指定します。デフォルトは null です。	任意

(3) <link:linkbaseRef> 要素

リンクベースへの参照定義をする要素です。インスタンス文書では，<link:linkbaseRef> 要素を使用して，インスタンス文書に特化したリンクベースを参照できます。

<link:linkbaseRef> 要素に必要な属性は，タクソノミー文書の場合と同じです。詳細については，「5.3.4 リンクベースへの参照定義」を参照してください。

インスタンス文書の場合，uCosminexus Business Reporting Processor は，<xbrli:xbrl> 要素下に記述されている <link:linkbaseRef> 要素を処理します。

5.4.4 カスタムロール，カスタムアークロールへの参照定義

XBRL2.1 の場合，インスタンス文書から，カスタムロールおよびカスタムアークロールを参照できます。カスタムロールおよびカスタムアークロールは，タクソノミー本体で定義されています。インスタンス文書からカスタムロールへの参照には <link:roleRef> 要素を，カスタムアークロールへの参照には <link:arcroleRef> 要素を使用します。

カスタムロール，カスタムアークロールへの参照定義は，リンクベースからタクソノミー本体へのカスタムロール，カスタムアークロールへの参照定義と同じです。詳細については，「5.3.5 リンクベース要素（linkbase）」を参照してください。

uCosminexus Business Reporting Processor は，次の個所に記載されている要素を処理します。

- <xbrli:xbrl> 要素下に記述されている <link:roleRef> 要素
- <xbrli:xbrl> 要素下に記述されている <link:arcroleRef> 要素

5.4.5 コンテキスト要素（context）と単位要素（unit）

XBRL2.0 の場合は，コンテキスト要素の一部として，単位要素が定義されていました。また，コンテキスト要素を，数値コンテキストまたは非数値コンテキストに区別する必要がありました。

XBRL2.1 の場合は，コンテキストと単位を分離して定義します。また，数値，非数値の区別がなくなり，一つのコンテキスト要素に統合されました。

XBRL2.0 と XBRL2.1 のコンテキスト要素の変更点を次の表に示します。

表 5-42 XBRL2.0 と XBRL2.1 のコンテキスト要素の変更点

分類	XBRL2.0	XBRL2.1	説明
コンテキスト要素	<xbrli:numericContext> <xbrli:nonNumericContext>	<xbrli:context>	XBRL2.1 では，<xbrli:context> 要素に統合されました。
単位要素	<xbrli:unit>	<xbrli:unit>	要素名は同じですが，含むことができる子要素および出現位置が変更されました。

ここでは，XBRL2.1 の場合のコンテキスト要素および単位要素について説明します。

（1） <xbrli:context> 要素

<xbrli:numericContext> 要素に定義されていた次の属性は廃止されました。

廃止された <xbrli:numericContext> 要素の属性を次の表に示します。

表 5-43 廃止された <xbrli:numericContext> 要素の属性

項番	属性名	説明
1	precision	有効桁数は数値アイテム型要素の属性に付けるように変更されたため，この属性は廃止されました。
2	cwa	完全に廃止されました。

（2） 時期要素（period）

XBRL2.0 の場合，<xbrli:period> 要素は，<xbrli:numericContext> 要素および <xbrli:nonNumericContext> 要素の必須要素であり，時期または期間を表現するためのコンテナ要素でした。

XBRL2.1 では，<xbrli:context> 要素の必須要素であり，出現位置も変更されていません。ただし，<xbrli:period> 要素の子要素に変更があります。

<xbrli:period> 要素の子要素を次の表に示します。

表 5-44 <xbrli:period> 要素の子要素

項番	要素名	説明
1	<xbrli:instant>	XBRL2.0 の場合と同様です。 ある特定の時点を示すときに使用し，dateUnion 型（XML の date 型または dateTime 型）の値を指定します。
2	<xbrli:forever>	XBRL2.0 の場合と同様です。 永遠を表現するための要素です。属性，子要素，値があっても無視します。
3	<xbrli:startDate> <xbrli:endDate>	XBRL2.0 の場合と同様です。 開始日と終了日を示し，期間を表現する形式です。共に，dateUnion 型の値を指定します。
4	<xbrli:startDate> <xbrli:duration>	XBRL2.1 では，<xbrli:duration> 要素が廃止されたため，この組み合わせも廃止されました。
5	<xbrli:duration> <xbrli:endDate>	XBRL2.1 では，<xbrli:duration> 要素が廃止されたため，この組み合わせも廃止されました。

この組み合わせのうち項番 3 の <xbrli:startDate> 要素と <xbrli:endDate> 要素の組み合わせについて，uCosminexus Business Reporting Processor は日時が逆転していないかどうかをチェックし，エラーを通知します。なお，XBRL2.0 の場合，同日の日付を指定すると，エラーとして検出されていましたが，XBRL2.1 の場合は，エラーではなく正常データとして扱われます。

(3) 単位要素（unit）

XBRL2.0 の場合，<xbrli:unit> 要素は <xbrli:numericContext> 要素の必須要素であり，計量に関する標準を指定する要素でした。

XBRL2.1 の場合，<xbrli:unit> 要素は，コンテキストと切り離され，ルート要素である <xbrli:xbrl> 要素の子要素に出現位置が変更されました。

XBRL2.0 と XBRL2.1 の単位要素の属性，および子要素の変更点を次の表に示します。

表 5-45 XBRL2.0 と XBRL2.1 の単位要素の属性，および子要素の変更点

項番	属性または子要素	変更点	XBRL のバージョン
1	id	数値アイテム型要素から参照できるように， 単位要素に ID 型の属性が追加されました。	2.1
2	<xbrli:measure>	XBRL2.0 では一つだけ記述できますが， XBRL2.1 では複数記述できるように変更されました。	2.0 および 2.1
3	<xbrli:operator>	XBRL2.1 では廃止されました。 name 属性に乗除算を指定していましたが， <xbrli:divide> 要素に変更されています。	2.0

項番	属性または子要素	変更点	XBRL のバージョン
4	<xbrli:divide>	XBRL2.1 で追加されました。 子要素として分子（<xbrli:unitNumerator>）および分母（<xbrli:unitDenominator>）を表す要素を持ちます。	2.1

数値アイテム型要素が参照する <xbrli:unit> 要素の内容には，幾つかの制約があります。uCosminexus Business Reporting Processor では，記述された <xbrli:unit> 要素が制約を満たしているかをチェックします。

また，<xbrli:unit> 要素には，単位を示す ISO4217 で既定されている値を使用する必要があります。ただし，uCosminexus Business Reporting Processor では，<xbrli:unit> 要素に指定された値が，ISO4217 で規定されているかどうかチェックしません。

5.4.6 アイテム型要素（item）

XBRL2.0 と XBRL2.1 のアイテム型要素の変更点を次に示します。

- コンテキストが <xbrli:context> 要素に変更され，<xbrli:context> 要素を参照するための属性が追加されました。
- コンテキストに数値または非数値の区別がなくなったため，数値アイテム型および非数値アイテム型で，同じコンテキストを参照できるようになりました。
- コンテキストから単位要素が分離され，単位要素が独立しました。それに伴い，数値アイテム型要素は単位要素に属している必要があるため，数値アイテム型要素から単位要素を参照するための属性が追加されました。
- XBRL2.0 では，数値アイテム型要素の場合，<xbrli:numericContext> の precision 属性でその数値の有効桁数を記述していました。XBRL2.1 では，数値アイテム型要素の属性に変更されています。

XBRL2.0 と XBRL2.1 のアイテム型要素の変更点を次の表に示します。

表 5-46　XBRL2.0 と XBRL2.1 のアイテム型要素の変更点

項番	属性	変更点	XBRL バージョン
1	id	XBRL2.0 では，スキーマ定義では定義されていませんでした。XBRL2.1 では，定義されています。	2.0 または 2.1
2	nonNumericContext	contextRef 属性に変更されました。	2.0
3	numericContext	contextRef 属性に変更されました。	2.0
4	contextRef	XBRL2.1 で追加された，<xbrli:context> 要素を参照するための属性です。数値または非数値アイテム型要素に関係なく必須です。値には，<xbrli:context> 要素の id 属性値を指定します。	2.1

項番	属性	変更点	XBRL バージョン
5	unitRef	XBRL2.1 で追加された，<xbrli:unit> 要素を参照するための属性です。数値アイテム型要素では必須です。値には，<xbrli:unit> 要素の id 属性値を指定します。	2.1
6	precision	XBRL2.1 で追加された属性です。数値アイテム型要素の場合に，この要素が持つ値の有効桁数を示します。decimals 属性と排他関係にあります。	2.1
7	decimals	XBRL2.1 で追加された属性です。数値アイテム型要素の場合，この要素が持つ値の小数桁数を示します。precision 属性と排他関係にあります。	2.1
8	xsi:nil	この属性の値が "true" の場合，要素は空である必要があります。	2.1

uCosminexus Business Reporting Processor では，次の内容をチェックします。

- 数値アイテム型に，precision 属性または decimals 属性のどちらかが指定されているかのチェック。
- contextRef 属性の値がコンテキスト要素の id 属性値であるか，unitRef 属性の値が単位要素の id 属性値であるかなどのチェック。

5.4.7 脚注リンク要素（footnoteLink）

ここでは，XBRL2.1 の場合の脚注リンク要素について説明します。

<link:footnoteLink> 要素は，インスタンス文書で使用される拡張リンク型要素で，属性定義が必要です。また，<link:footnoteLink> 要素には，ロケータ型要素（loc），脚注リソース型要素（footnote），脚注アーク型要素（footnoteArc）を定義できます。

uCosminexus Business Reporting Processor は，<xbrli:xbrl> 要素下に記述されている <link:footnoteLink> 要素を処理します。

<link:footnoteLink> 要素の属性を次の表に示します。

表 5-47 <link:footnoteLink> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	extended（拡張リンク）固定です。	必須
2	xlink:role	リンク種別を意味するカスタムロール値です。標準値は http://www.xbrl.org/2003/role/link です。	必須
3	xlink:title	任意の文字列を指定します。デフォルトは null です。	任意
4	id	ID 型の値です。デフォルトは null です。	任意

項番	属性名	内容	必須条件
5	xml:base	基底 URI を記述します。デフォルトは null です。	任意

（1） ロケータ型要素（loc）

<link:loc> 要素は，拡張リンクでのロケータ型要素で，インスタンス文書では <link:footnoteLink> 要素の子要素として出現します。<link:loc> 要素の属性を次の表に示します。

表 5-48　<link:loc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	locator 固定です。	必須
2	xlink:href	アイテム型要素，タプル型要素，またはリソース型要素への参照 URI を記述します。任意の値を指定できます。また，URI のフラグメントには，XPointer 式を記述できます。 XPointer 形式は，ショートハンドポインタ（Shorthand Pointer）形式，または XPointer element() scheme です。	必須
3	xlink:label	NCName 型データです。アーク型要素によって参照しているロケータ型要素（<link:loc>）を識別するために必要です。	必須
4	xlink:role	任意の URI です。<link:roleType> 要素との関連はありません。デフォルトは null です。	任意
5	xlink:title	任意の値を指定します。デフォルトは null です。	任意

この形式以外のロケータ型要素が出現した場合，エラーとして処理を中止します。

（2） 脚注リソース型要素（footnote）

<link:footnote> 要素は，インスタンス文書のリソース型要素で，<link:footnoteLink> 要素の子要素として出現します。<link:footnote> 要素には，任意の文字列を要素の値として記述します。<link:footnote> 要素の属性を次の表に示します。

表 5-49　<link:footnote> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	resource 固定です。	必須
2	xlink:label	NCName 型データです。アーク型要素によって参照しているリソース型要素（<link:footnote>）を識別するために必要です。同一の値を持つことが許されます。	必須
3	xlink:role	カスタムロール値または標準値を指定します。デフォルトは null です。標準値は，http://www.xbrl.org/2003/role/footnote です。	任意

項番	属性名	内容	必須条件
4	xml:lang	任意の言語を特定するために使用します。ISO で定義されている言語コードである「ja」,「en」などを指定します。	必須
5	xlink:title	任意の値を指定します。デフォルトは null です。	任意
6	id	ID 型の値を指定します。デフォルトは null です。	任意

この形式以外のリソース型要素が出現した場合，エラーとして処理を中止します。

(3) 脚注アーク型要素（footnoteArc）

<link:footnoteArc> 要素は，インスタンス文書のアーク型要素で，<link:footnoteLink> 要素の子要素として出現します。<link:footnoteArc> 要素の属性を次の表に示します。

表 5-50 <link:footnoteArc> 要素の属性

項番	属性名	内容	必須条件
1	xlink:type	arc 固定です。	必須
2	xlink:show	指定できる値は，new，replace，embed，other，または none です。属性を省略できます。	任意
3	xlink:actuate	指定できる値は，onLoad，onRequest，other，または none です。デフォルトは null です。属性を省略できます。	任意
4	xlink:from	NCName 型データです。ロケータ型要素（<link:loc>）の xlink:label 属性の内容と同じ値を指定します。	必須
5	xlink:to	NCName 型データです。リソース型要素（<link:footnote>）の xlink:label 属性の内容と同じ値を指定します。	必須
6	xlink:arcrole	カスタムアークロール値または標準値を指定します。 標準値は http://www.xbrl.org/2003/arcrole/fact-footnote です。	必須
7	xlink:title	任意の値を指定します。デフォルトは null です。	任意
8	use	optional または prohibited を指定します。デフォルトは optional です。	任意
9	priority	use 属性の optional または prohibited の関係に対してのプライオリティ（優先順位）を指定するための属性です。デフォルトは 0 です。	任意
10	order	複数の兄弟要素間での表示順序を示す 10 進数を指定します。デフォルトは 1 です。	任意

この形式以外のアーク型要素が出現した場合，エラーとして処理を中止します。

5.4.8 タプル型要素（tuple）

通常の通常の検証では，タプルの定義についてだけ検証します。完全な妥当性検証では，

次に示す名前名前空間を持つ属性がタプルに記述されていないかについても検証します。

- http://www.xbrl.org/2003/instance
- http://www.xbrl.org/2003/linkbase
- http://www.xbrl.org/2003/XLink
- http://www.w3.org/1999/xlink

5.4.9 ディメンジョンの指定（XBRL Dimensions1.0 の場合）

インスタンス文書のディメンジョンは，プライマリー・アイテムと関連づけられた，コンテキストのシナリオ要素またはセグメント要素で指定します。

 要素および <xbrli:scenario> 要素は，任意の要素を子要素にできるコンテナ要素として定義されています。XBRL Dimensions1.0 では 要素および <xbrli:scenario> 要素を利用して，インスタンス文書で次元を表現します。

（1） <xbrldi:typedMember> 要素

<xbrldi:typedMember> 要素は，Typed Dimension を記述するときに使用します。<xbrldi:typedMember> 要素の属性を次の表に示します。

表 5-51 <xbrldi:typedMember> 要素の属性

属性名	内容
dimension	タクソノミー文書で定義した Typed Dimension の QName を指定します。

<xbrldi:typedMember> 要素の内容には，dimension 属性で記述した Typed Dimension の正当な要素および内容を記述してください。正当な要素および内容とは，Typed Dimension で指定する xbrldt:typedDomainRef 属性の参照先の，XML Schema の要素宣言で正当となる任意の形式です。

（2） <xbrldi:explicitMember> 要素

<xbrldi:explicitMember> 要素は，Explicit Dimension を記述するときに使用します。<xbrldi:explicitMember> 要素の属性を次の表に示します。

表 5-52 <xbrldi:explicitMember> 要素の属性

属性名	内容
dimension	タクソノミー文書で定義した Explicit Dimension の QName を指定します。

<xbrldi:explicitMember> 要素の内容には，dimension 属性で記述した Explicit Dimension の正当なメンバの QName を記述してください。

5.5 取り扱えるデータ型

uCosminexus Business Reporting Processor では，XBRL データ型およびユーザー独自のデータ型（**ユーザーデータ型**）を使用できます。

ここでは，使用できる XBRL データ型，およびユーザーデータ型について説明します。

ユーザーデータ型は，XBRL のバージョンによって異なります。操作する XBRL 文書に合ったユーザーデータ型を使用してください。

5.5.1 XBRL データ型

uCosminexus Business Reporting Processor で使用できる XBRL データ型について説明します。

文書中の XBRL データ型と，対応する XBRL DOM で保持するデータ型クラスを，次の表に示します。

表 5-53 XBRL データ型と，対応する XBRL DOM で保持するデータ型クラス

項番	文書中のデータ型	対応する XBRL DOM データ型クラス	XBRL のバージョン
1	monetaryItemType	java.math.BigDecimal	2.0 および 2.1
2	sharesItemType	java.math.BigDecimal	2.0 および 2.1
3	pureItemType	java.math.BigDecimal	2.1
4	fractionItemType	com.hitachi.xbrl.types.XBRLFraction	2.1
5	decimalItemType	java.math.BigDecimal	2.0 および 2.1
6	floatItemType	java.lang.Float	2.0 および 2.1
7	doubleItemType	java.lang.Double	2.0 および 2.1
8	integerItemType	java.math.BigInteger	2.0 および 2.1
9	nonPositiveIntegerItemType	java.math.BigInteger	2.1
10	negativeIntegerItemType	java.math.BigInteger	2.1
11	longItemType	java.lang.Long	2.1
12	intItemType	java.lang.Integer	2.1
13	shortItemType	java.lang.Short	2.1
14	byteItemType	java.lang.Byte	2.1

項番	文書中のデータ型	対応する XBRL DOM データ型クラス	XBRL のバージョン
15	nonNegativeIntegerItemType	java.math.BigInteger	2.1
16	unsignedLongItemType	java.math.BigInteger	2.1
17	unsignedIntItemType	java.math.BigInteger	2.1
18	unsignedShortItemType	java.math.BigInteger	2.1
19	unsignedByteItemType	java.math.BigInteger	2.1
20	positiveIntegerItemType	java.math.BigInteger	2.1
21	stringItemType	java.lang.String	2.0 および 2.1
22	booleanItemType	java.lang.Boolean	2.0 および 2.1
23	hexBinaryItemType	com.hitachi.xbrl.types.XBRLHexBinary	2.0 および 2.1
24	base64BinaryItemType	com.hitachi.xbrl.types.XBRLBase64Binary	2.0 および 2.1
25	anyURIItemType	java.lang.String	2.0 および 2.1
26	uriItemType	java.lang.String	2.0
27	QNameItemType	com.hitachi.xbrl.types.XBRLQName	2.0 および 2.1
28	NOTATIONItemType	com.hitachi.xbrl.types.XBRLQName	2.0
29	durationItemType	com.hitachi.xbrl.types.XBRLDuration	2.0 および 2.1
30	dateTimeItemType	com.hitachi.xbrl.types.XBRLDateUnion	2.0 および 2.1
31	timeItemType	com.hitachi.xbrl.types.XBRLTime	2.0 および 2.1
32	dateItemType	com.hitachi.xbrl.types.XBRLDate	2.0 および 2.1
33	gYearMonthItemType	com.hitachi.xbrl.types.XBRLGYearMonth	2.0 および 2.1
34	gYearItemType	com.hitachi.xbrl.types.XBRLGYear	2.0 および 2.1
35	gMonthDayItemType	com.hitachi.xbrl.types.XBRLGMonthDay	2.0 および 2.1
36	gDayItemType	com.hitachi.xbrl.types.XBRLGDay	2.0 および 2.1
37	gMonthItemType	com.hitachi.xbrl.types.XBRLGMonth	2.0 および 2.1

項番	文書中のデータ型	対応する XBRL DOM データ型クラス	XBRL のバージョン
38	normalizedStringItemType	java.lang.String	2.1
39	tokenItemType	java.lang.String	2.1
40	languageItemType	java.lang.String	2.1
41	NameItemType	java.lang.String	2.1
42	NCNameItemType	java.lang.String	2.1

XBRL DOM で保持するデータ型クラスには，大きく分けて，公開されているデータ型クラスと，XBRL DOM 独自のデータ型クラスとがあります。データ型クラス名が「com.hitachi.xbrl.types」で始まる，XBRL DOM 独自のデータ型クラスの詳細については，「6.　API リファレンス」を参照してください。

5.5.2　ユーザーデータ型（XBRL2.0 の場合）

ユーザーデータ型とは，ユーザーが独自に定義したデータ型のことです。ユーザーは，独自にデータ型の定義をタクソノミー本体に記述します。

ユーザーデータ型を定義する場合，XBRL データ型を拡張する方法と，XBRL データ型に制約を追加する方法があります。

ここでは，XBRL2.0 の場合の，ユーザーデータ型と外部とのインターフェース，使用できるユーザーデータ型の条件，および定義例について説明します。

(1) 外部とのインターフェース

ユーザーデータ型と外部とのインターフェースについて説明します。

ユーザーデータ型と XBRL データ型と区別するために，ユーザーデータ型では，データ型を表すコードが常にその他（OTHER_ITEM）となります。

ユーザーデータ型を XBRL データ型から派生して作成し，使用できるユーザーデータ型の条件に合っている場合は，ユーザーデータ型が数値または非数値データ型かどうかが判断されます。条件に合っていない場合，ユーザーデータ型は文字列として保持・参照されます。

それぞれの場合で使用できるメソッドを次に示します。

- データ型を表すコードを取得するには，XBRLTaxonomyItem および XBRLInstanceItem インターフェースの getItemType() メソッドを使用できます。
- アイテムが数値型かどうかを調べるには，XBRLTaxonomyItem および XBRLInstanceItem インターフェースの isNumeric() メソッドを使用できます。
- 文字列として保持されているユーザーデータ型を参照するには，XBRLInstanceItem インターフェースの getValue() メソッドを使用できます。ユーザーデータ型の内容を

設定および変更するには，setValue() メソッドを使用できます。

(2) 使用できるユーザーデータ型の条件

ユーザーデータ型を定義する場合，次の条件をすべて満たす形式で定義してください。これらの条件をすべて満たす場合，データ型が数値データ型または非数値データ型であるかが判断されます。

- <xsd:element> 要素定義と同一ドキュメント内に定義されたユーザーデータ型。
- <xsd:complexType> 要素で大域複合型として定義されたユーザーデータ型。
- <xsd:simpleContent> 要素で単純内容複合型を派生して単純内容モデルを定義したユーザーデータ型。
- <xsd:extension> 要素で単純内容モデルを拡張したユーザーデータ型，または <xsd:restriction> 要素で単純内容モデルに新しく制約を追加したユーザーデータ型。
- <xsd:extension> および <xsd:restriction> の base 属性が XBRL データ型であるユーザーデータ型。

(3) 拡張したユーザーデータ型の例

```
<schema
 :
    xmlns:cbrpi="http://cosminexus.com/xbrl/"
 :
  <element id                ="paymentOfdividends.preferredDividends"
           name              ="paymentOfdividends.preferredDividends"
           type              ="cbrpi:ProductPrice"
           substitutionGroup ="xbrli:item"
  />
 :
  <complexType name ="ProductPrice">
    <simpleContent>
      <extension base = "xbrli:integerItemType">
        <attribute name="kind" type="integer" use="required"/>
      </extension>
    </simpleContent>
  </complexType>
 :
</schema>
```

(4) 新しく制約を追加したユーザーデータ型の例

```
<schema
 :
    xmlns:cbrpi="http://cosminexus.com/xbrl/"
 :
  <element id                ="paymentOfdividends.preferredDividends"
           name              ="paymentOfdividends.preferredDividends"
           type              ="cbrpi:totalDigits"
           substitutionGroup ="xbrli:item"
  />
 :
  <complexType name ="totalDigits">
    <simpleContent>
      <restriction base = "xbrli:integerItemType">
        <totalDigits value="5">
      </restriction>
    </simpleContent>
  </complexType>
 :
</schema>
```

5.5.3 ユーザーデータ型（XBRL2.1 の場合）

ユーザーデータ型とは，ユーザーが独自に定義したデータ型のことです。ユーザーは，独自にデータ型の定義をタクソノミー本体に記述します。

XBRL2.1 の場合，<xsd:restriction> 要素を使用して制約を追加する方法だけを使用できます。

ここでは，XBRL2.1 の場合の，ユーザーデータ型と外部とのインターフェース，使用できるユーザーデータ型の条件，および定義例について説明します。

(1) 外部とのインターフェース

ユーザーデータ型と外部とのインターフェースについて説明します。

ユーザーデータ型と XBRL データ型と区別するために，ユーザーデータ型では，データ型を表すコードが常にその他（OTHER_ITEM）となります。

ユーザーデータ型を XBRL データ型から派生して作成し，使用できるユーザーデータ型の条件に合っている場合は，ユーザーデータ型が数値または非数値データ型かどうかが判断されます。条件に合っていない場合，ユーザーデータ型は文字列として保持・参照されます。さらに XBRL2.1 の場合は，派生元のデータ型を取得することもできます。

それぞれの場合で使用できるメソッドを次に示します。

- データ型を表すコードを取得するには，XBRLTaxonomyItem および XBRLInstanceItem インターフェースの getItemType() メソッドを使用できます。
- アイテムが数値型かどうかを調べるには，XBRLTaxonomyItem および XBRLInstanceItem インターフェースの isNumeric() メソッドを使用できます。
- 文字列として保持されているユーザーデータ型を参照するには，XBRLInstanceItem インターフェースの getValue() メソッドを使用できます。ユーザーデータ型の内容を設定および変更するには，setValue() メソッドを使用できます。
- 派生元のデータ型を取得するには，XBRLTaxonomyItem および XBRLInstanceItem インターフェースの getBaseItemType() メソッドを使用できます。

(2) 使用できるユーザーデータ型の条件

制約を追加する <xsd:restriction> 要素と，<xsd:element> 要素の定義は同一ドキュメント内である必要はありません。さらに，局所的な宣言も使用できます。

ここでは，大域的な宣言（グローバル宣言）の場合と局所的な宣言（ローカル宣言）に分けて，使用できるユーザーデータ型の条件を示します。

なお，完全な妥当性検証を実行した場合は，ユーザーデータ型に XBRL2.1 で禁止されている定義内容がないかを検証します。この場合，xbrli:fractionItemType 型以外から派生したユーザーデータ型が構造を持っているとエラーとなるため，次に示すユーザーデータ型定義（<xsd:complexType> 要素）の子要素に <xsd:simpleContent> 要素や

<xsd:annotation> 要素以外が使用されているとエラーになります。

- あるアイテムとして認識された <xsd:element> 要素の子要素である <xsd:complexType> 要素（局所定義）
- あるアイテムとして認識された <xsd:element> 要素の type 属性値で参照される <xsd:complexType> 要素（大域定義）

(a) 大域定義の場合

- 次のいずれかのドキュメント内で定義されているユーザーデータ型。
  - <xsd:element> 要素定義と同じドキュメント。
  - <xsd:element> 要素が定義されているドキュメント内の <xsd:include> 要素または <xsd:import> 要素によって取り込まれるドキュメント。
  - <xsd:element> 要素が定義されているドキュメントを，<xsd:include> 要素で取り込むドキュメント。
- <xsd:complexType> 要素で大域複合型として定義されたユーザーデータ型。
- <xsd:simpleContent> 要素で単純内容複合型を派生して単純内容モデルを定義したユーザーデータ型。
  ただし，xbrli:fractionItemType からの派生の場合は，<xsd:complexContent> 要素で複合型を派生した複合内容モデルを定義したユーザーデータ型。
- <xsd:restriction> 要素で単純内容モデルに新しく制約を追加したユーザーデータ型。
- <xsd:restriction> の base 属性が XBRL データ型であるユーザーデータ型，または，その base 属性が XBRL データ型から派生したユーザーデータ型。

(b) 局所定義の場合

- <xsd:element> 要素の子要素として定義されたユーザーデータ型。
- <xsd:complexType> 要素で局所複合型として定義されたユーザーデータ型。
- <xsd:simpleContent> 要素で単純内容複合型を派生して単純内容モデルを定義したユーザーデータ型。
  ただし，xbrli:fractionItemType からの派生の場合は，<xsd:complexContent> 要素で複合型を派生した複合内容モデルを定義したユーザーデータ型。
- <xsd:restriction> 要素で単純内容モデルに新しく制約を追加したユーザーデータ型。
- <xsd:restriction> の base 属性が XBRL データ型であるユーザーデータ型，または，その base 属性が XBRL データ型から派生したユーザーデータ型。

(3) 大域定義の例

```
<schema
 :
    xmlns:cbrpi="http://cosminexus.com/xbrl/"
 :
  <element id                ="paymentOfdividends.preferredDividends"
           name              ="paymentOfdividends.preferredDividends"
           type              ="cbrpi:totalDigits"
           substitutionGroup ="xbrli:item"
  />
 :
  <complexType name ="totalDigits">
    <simpleContent>
      <restriction base ="xbrli:integerItemType">
        <totalDigits value="5">
      </restriction>
    </simpleContent>
  </complexType>
 :
</schema>
```

(4) 局所定義の例

```
<schema
 :
    xmlns:cbrpi="http://cosminexus.com/xbrl/"
 :
  <element id                ="paymentOfdividends.preferredDividends"
           name              ="paymentOfdividends.preferredDividends"
           substitutionGroup ="xbrli:item">
    <complexType>
      <simpleContent>
        <restriction base="xbrli:integerItemType">
          <totalDigits value="5">
        </restriction>
      </simpleContent>
    </complexType>
  </element>
 :
</schema>
```

6 API リファレンス

uCosminexus Business Reporting Processor は，XBRL 文書を操作するための API として XBRL DOM を提供します。
この章では，XBRL DOM で利用できるインターフェースやクラス，およびそのメソッドやフィールドの詳細について説明します。

6.1 パッケージの概要

6.2 API 一覧（パッケージ順）

6.3 メソッド一覧（アルファベット順）

6.4 API リファレンスの形式

6.5 com.hitachi.xbrl パッケージ

6.6 com.hitachi.xbrl.instance パッケージ

6.7 com.hitachi.xbrl.link パッケージ

6.8 com.hitachi.xbrl.taxonomy パッケージ

6.9 com.hitachi.xbrl.types パッケージ

6.10 com.hitachi.xbrl.util パッケージ

6.11 com.hitachi.xbrl.simple.taxonomy パッケージ

6.1 パッケージの概要

ここでは，uCosminexus Business Reporting Processor が提供する XBRL 文書操作用 API である XBRL DOM について説明します。

XBRL DOM を構成するパッケージの概要を次の表に示します。

表 6-1 パッケージの概要

項番	パッケージ	概要
1	com.hitachi.xbrl	XBRL 文書を操作するための API である XBRL DOM のインターフェースを提供します。
2	com.hitachi.xbrl.instance	インスタンス文書を操作するためのインターフェースを提供します。
3	com.hitachi.xbrl.link	XBRL 文書に記述されているリンク情報を参照するためのインターフェースを提供します。
4	com.hitachi.xbrl.taxonomy	タクソノミー文書を参照するためのインターフェースを提供します。
5	com.hitachi.xbrl.types	XML Schema のデータ型および XBRL のデータ型を処理するためのクラスです。
6	com.hitachi.xbrl.util	XBRL DOM で利用できるユーティリティクラスです。
7	com.hitachi.xbrl.simple.taxonomy	単純なタクソノミー文書を新規作成，または拡張した新たなタクソノミー文書を作成するための簡易インターフェースを提供します。

6.1.1 名前空間接頭辞

このマニュアルでは，名前空間 URI を特定の名前空間接頭辞に対応づけて記述しています。名前空間接頭辞に対応する名前空間 URI は，XBRL のバージョンによって異なります。

XBRL2.0 の場合の名前空間接頭辞と名前空間 URI との対応を次の表に示します。

表 6-2 名前空間接頭辞と名前空間 URI との対応（XBRL2.0 の場合）

項番	名前空間接頭辞	名前空間 URI
1	xsd	http://www.w3.org/2001/XMLSchema
2	xsi	http://www.w3.org/2001/XMLSchema-instance
3	xlink	http://www.w3.org/1999/xlink
4	xbrli	http://www.xbrl.org/2001/instance
5	link	http://www.xbrl.org/2001/XLink/xbrllinkbase
6	xl	http://www.xbrl.org/2001/XLink

XBRL2.1 の場合の名前空間接頭辞と名前空間 URI との対応を次の表に示します。

表 6-3　名前空間接頭辞と名前空間 URI との対応（XBRL2.1 の場合）

項番	名前空間接頭辞	名前空間 URI
1	xsd	http://www.w3.org/2001/XMLSchema
2	xsi	http://www.w3.org/2001/XMLSchema-instance
3	xlink	http://www.w3.org/1999/xlink
4	xbrli	http://www.xbrl.org/2003/instance
5	link	http://www.xbrl.org/2003/linkbase
6	xl	http://www.xbrl.org/2003/XLink
7	ptv	http://www.xbrl.org/2003/ptv
8	xbrldt	http://xbrl.org/2005/xbrldt
9	xbrldi	http://xbrl.org/2006/xbrldi
10	xbrldte	http://xbrl.org/2005/xbrldt/errors
11	xbrldie	http://xbrl.org/2005/xbrldi/errors

6.1.2　オブジェクトモデル（XBRL2.0 の場合）

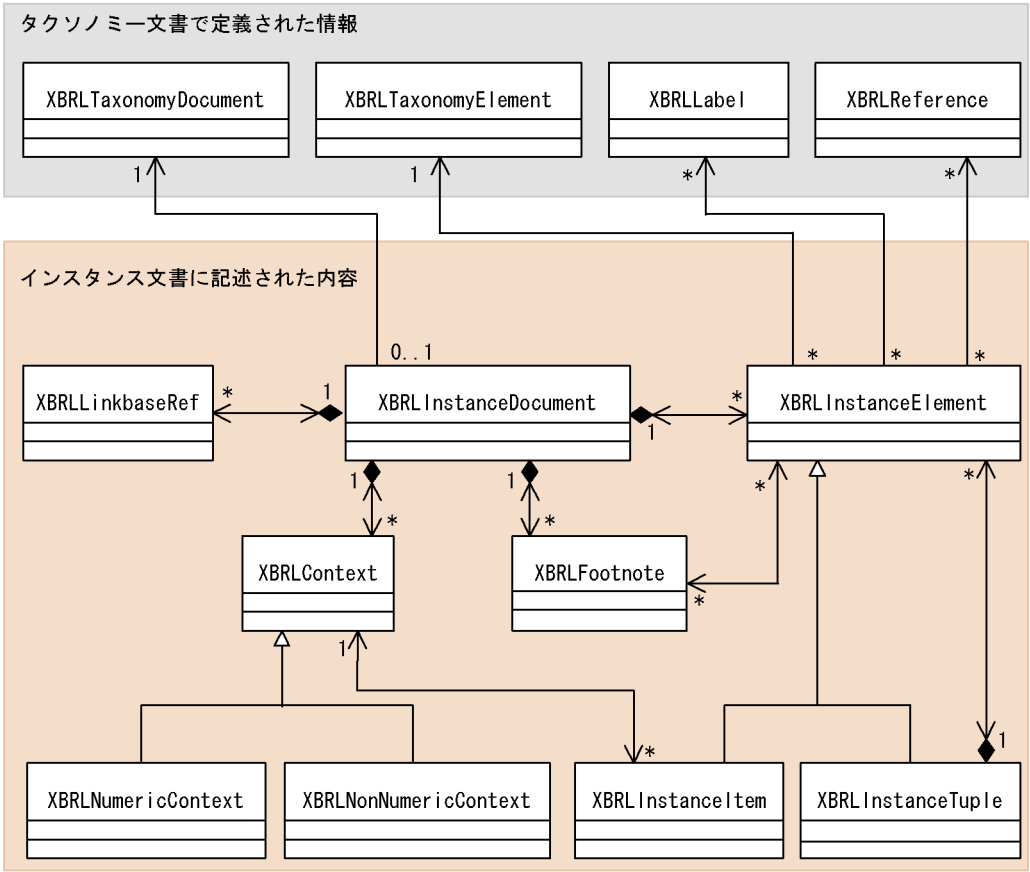
XBRL DOM は，XBRL 文書を操作するために使用する，多数のインターフェースを提供しています。XBRL 文書は，それらのインターフェースを実装したクラスのインスタンス，すなわちオブジェクトによって構築されるオブジェクトモデルとして表現されます。

ここでは，XBRL DOM が提供するインターフェースを，UML で記述したクラス図を示します。

（1）インスタンス文書のオブジェクトモデル

インスタンス文書のオブジェクトモデルを表現した UML のクラス図を次の図に示します。

図 6-1　インスタンス文書のオブジェクトモデルを表現した UML のクラス図（全体）



（凡例）

→ ：使用される方向を表します。

→◆ ：集約の一種でライフサイクルがほぼ一致する場合を表します。

→▷ ：スーパークラスとサブクラスの関係を表します。

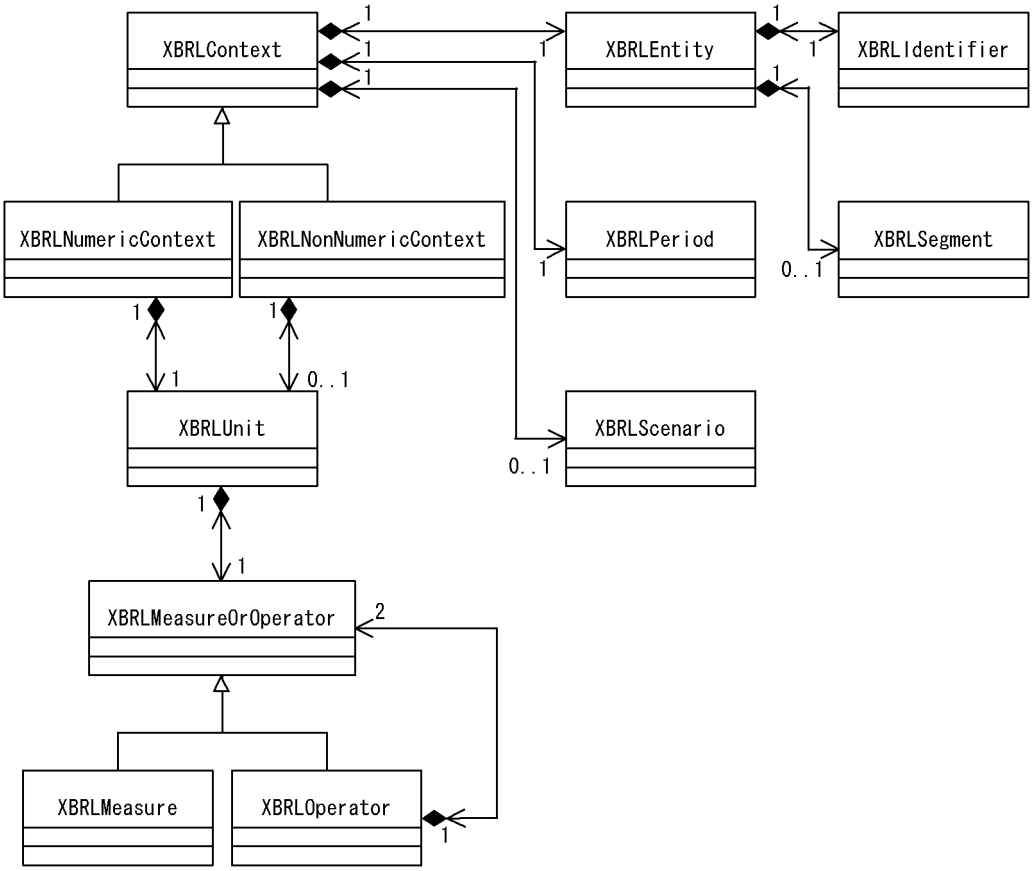
＊　：多重度が0以上を表します。

1　：多重度が1を表します。

0..1　：多重度が0または1を表します。

上記クラス図のうち，コンテキスト要素部分の詳細を次の図に示します。

図 6-2　インスタンス文書のオブジェクトモデルを表現した UML のクラス図（コンテキスト要素部分）



（凡例）

→ ：使用される方向を表します。

→◆ ：集約の一種でライフサイクルがほぼ一致する場合を表します。

→▷ ：スーパークラスとサブクラスの関係を表します。

1　：多重度が1を表します。

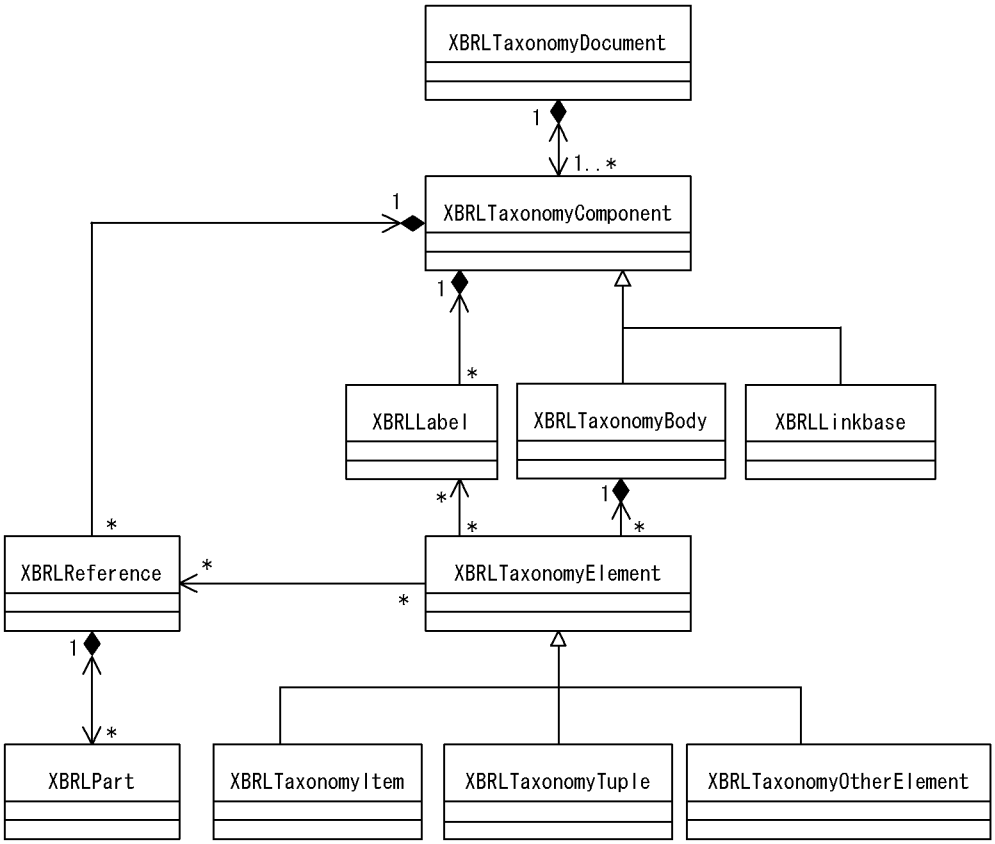
2　：多重度が2を表します。

0..1　：多重度が0または1を表します。

(2) タクソノミー文書のオブジェクトモデル

タクソノミー文書のオブジェクトモデルを表現した UML のクラス図を次の図に示します。

図 6-3　タクソノミー文書のオブジェクトモデルを表現した UML のクラス図



（凡例）

→　：使用される方向を表します。
→◆　：集約の一種でライフサイクルがほぼ一致する場合を表します。
→▷　：スーパークラスとサブクラスの関係を表します。
＊　：多重度が0以上を表します。
1　：多重度が1を表します。
1..＊　：多重度が1以上を表します。

6.1.3　オブジェクトモデル（XBRL2.1 の場合）

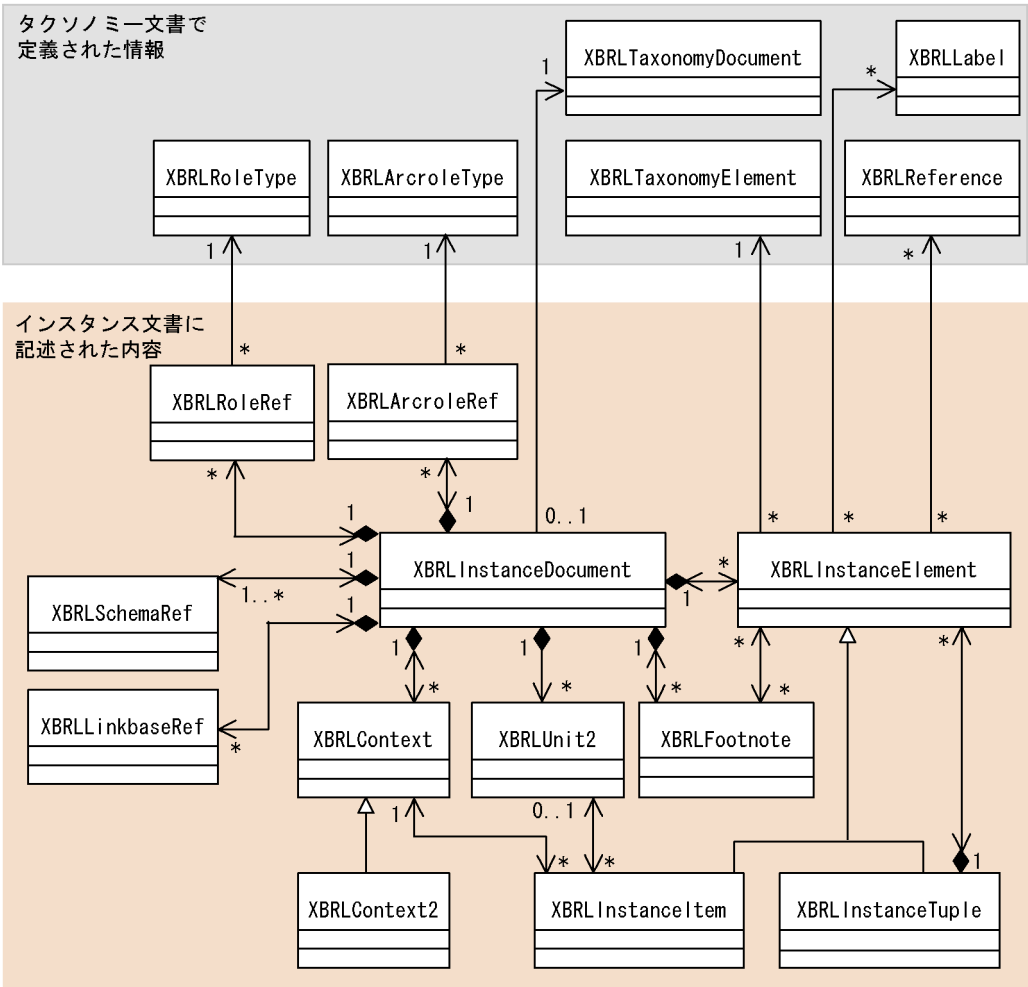
XBRL DOM は，XBRL 文書を操作するために使用する，多数のインターフェースを提供しています。XBRL 文書は，それらのインターフェースを実装したクラスのインスタンス，すなわちオブジェクトによって構築されるオブジェクトモデルとして表現されます。

ここでは，XBRL DOM が提供するインターフェースを，UML で記述したクラス図を示します。

(1) インスタンス文書のオブジェクトモデル

インスタンス文書のオブジェクトモデルを表現した UML のクラス図を次の図に示します。

図 6-4 インスタンス文書のオブジェクトモデルを表現した UML のクラス図（全体）



（凡例）

→ ：使用される方向を表します。

→◆ ：集約の一種でライフサイクルがほぼ一致する場合を表します。

→▷ ：スーパークラスとサブクラスの関係を表します。

* ：多重度が0以上を表します。

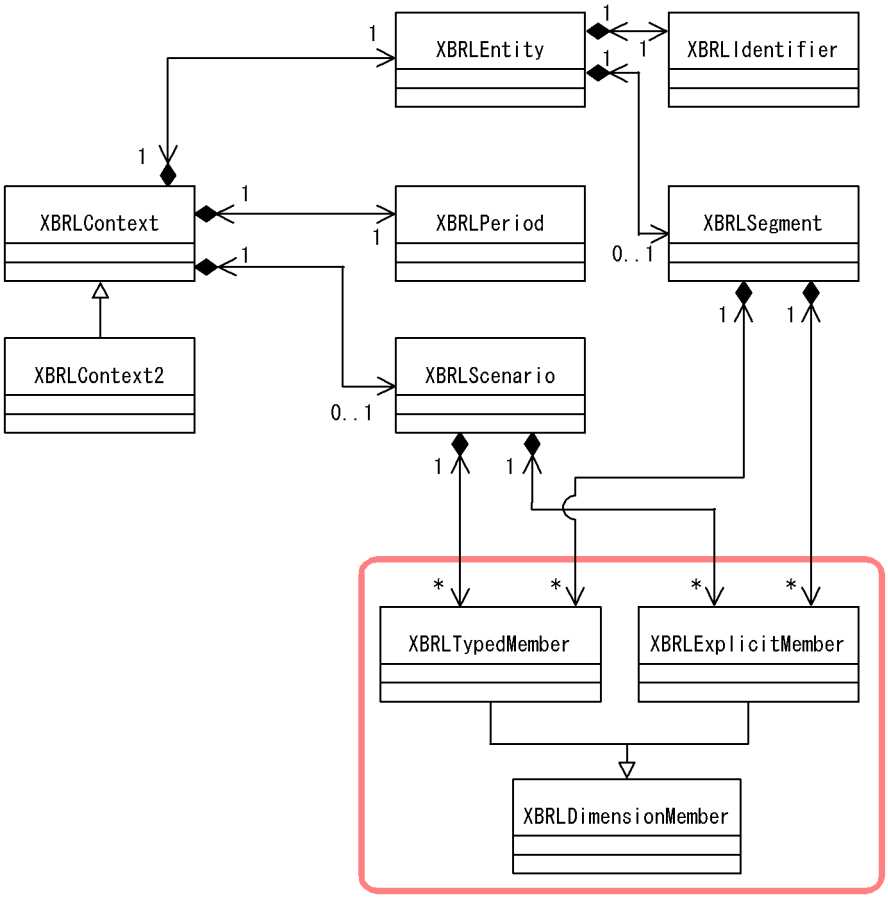
1 ：多重度が1を表します。

0..1 ：多重度が0または1を表します。

1..* ：多重度が1以上を表します。

上記クラス図のうち，コンテキスト要素部分の詳細を次の図に示します。

図 6-5　インスタンス文書のオブジェクトモデルを表現した UML のクラス図（コンテキスト要素部分）



（凡例）

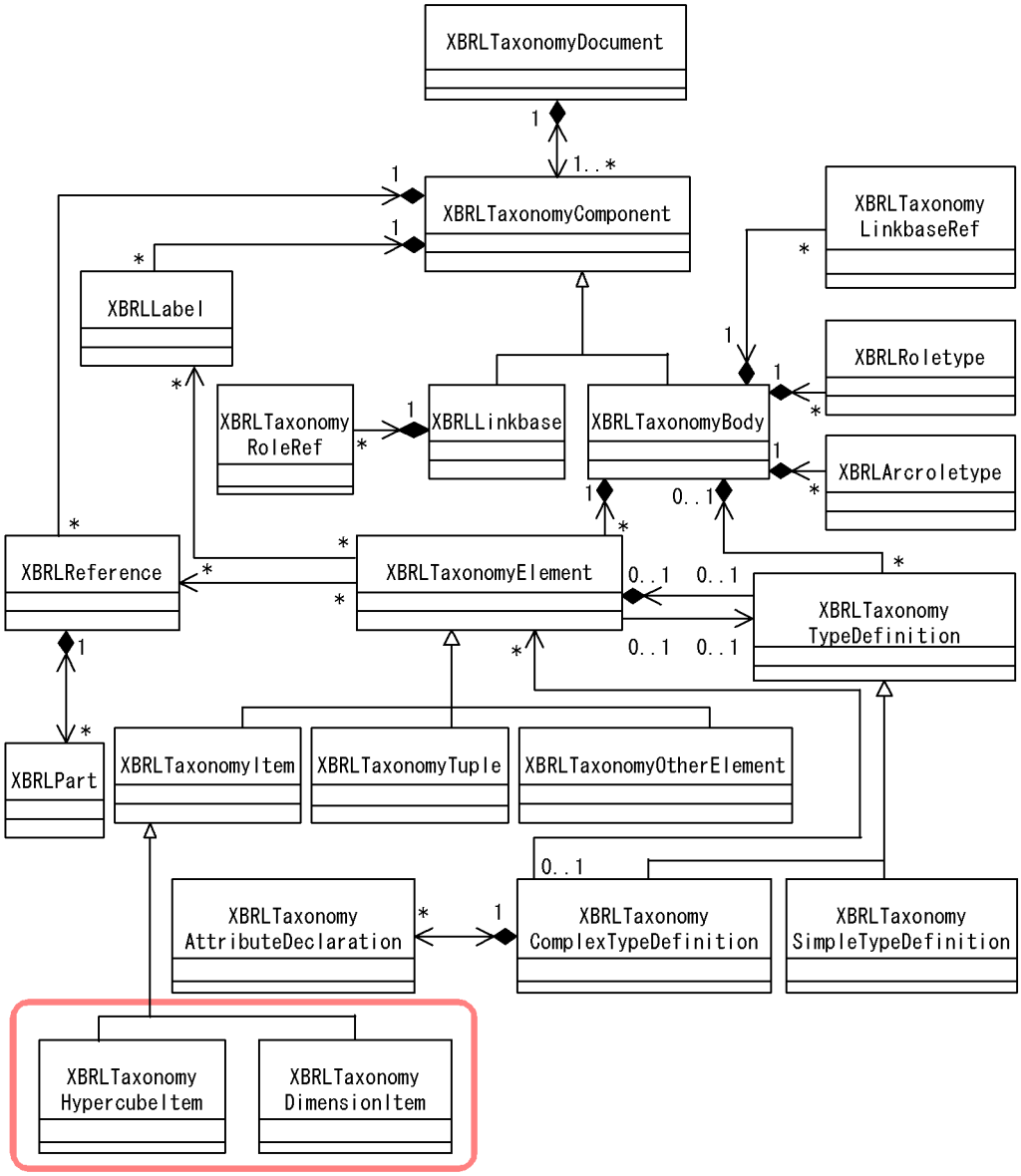
⟶	：使用される方向を表します。
⟶◆	：集約の一種でライフサイクルがほぼ一致する場合を表します。
⟶▷	：スーパークラスとサブクラスの関係を表します。
*	：多重度が0以上を表します。
1	：多重度が1を表します。
0..1	：多重度が0または1を表します。
1..*	：多重度が1以上を表します。
▭	：XBRL Dimensions1.0の場合のクラス図を表します。

(2) タクソノミー文書のオブジェクトモデル

通常のタクソノミー文書のオブジェクトモデルを表現した UML のクラス図を次の図に

示します。

図 6-6　タクソノミー文書のオブジェクトモデルを表現した UML のクラス図

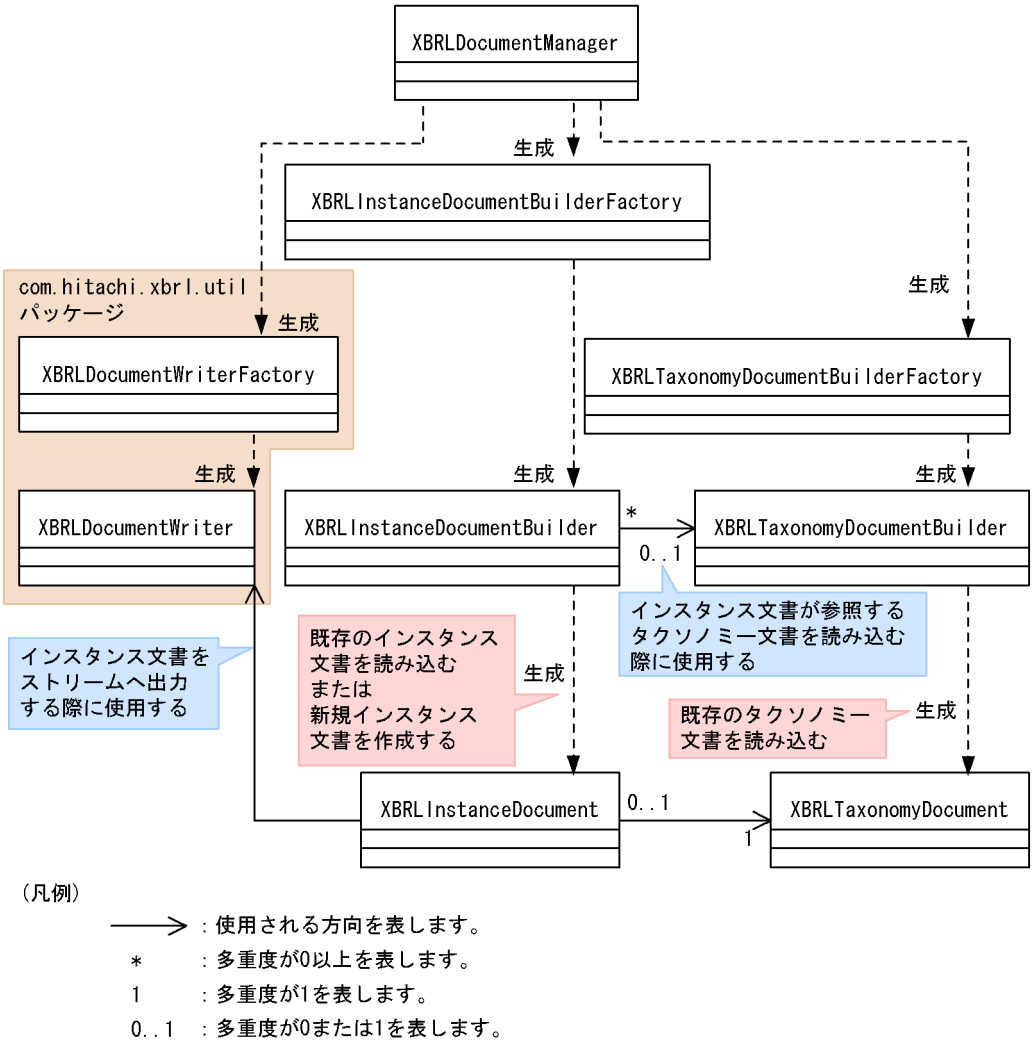


（凡例）

- ：使用される方向を表します。
- ：集約の一種でライフサイクルがほぼ一致する場合を表します。
- ：スーパークラスとサブクラスの関係を表します。
- *　：多重度が0以上を表します。
- 1　：多重度が1を表します。
- 0..1　：多重度が0または1を表します。
- 1..*　：多重度が1以上を表します。
- ：XBRL Dimensions1.0の場合のクラス図を表します。

タクソノミー文書の XBRL 文書管理機能のオブジェクトモデルを表現した UML のクラス図を次の図に示します。

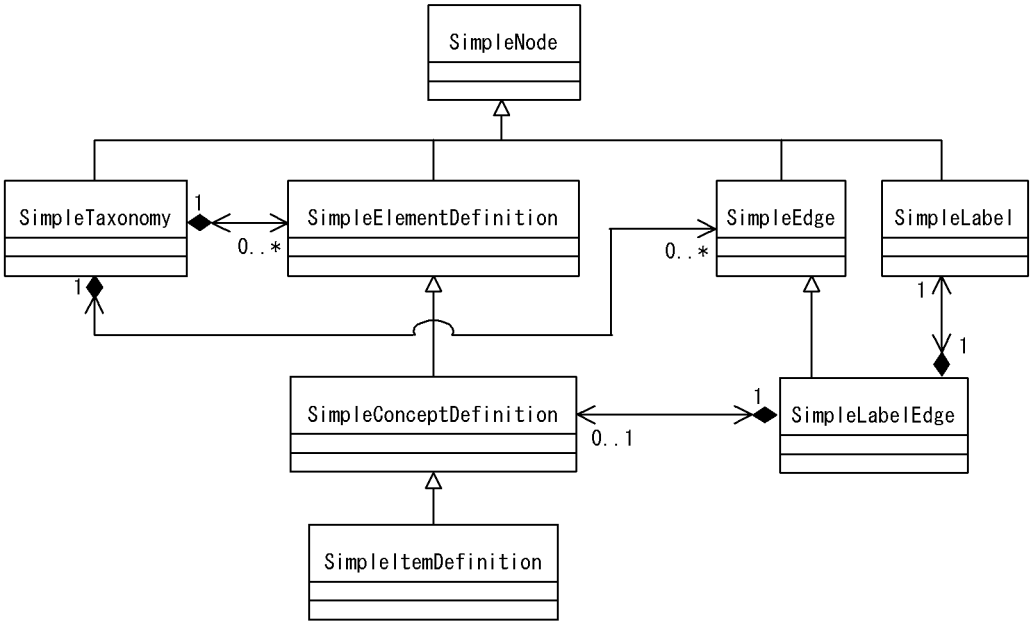
図 6-7 タクソノミー文書の XBRL 文書管理機能のオブジェクトモデルを表現した UML のクラス図



(3) 単純なタクソノミー文書のオブジェクトモデル

単純なタクソノミー文書のオブジェクトモデルを表現した UML のクラス図を次の図に示します。

図 6-8　単純なタクソノミー文書のオブジェクトモデルを表現した UML のクラス図



（凡例）

→ ：使用される方向を表します。

→◆ ：集約の一種でライフサイクルがほぼ一致する場合を表します。

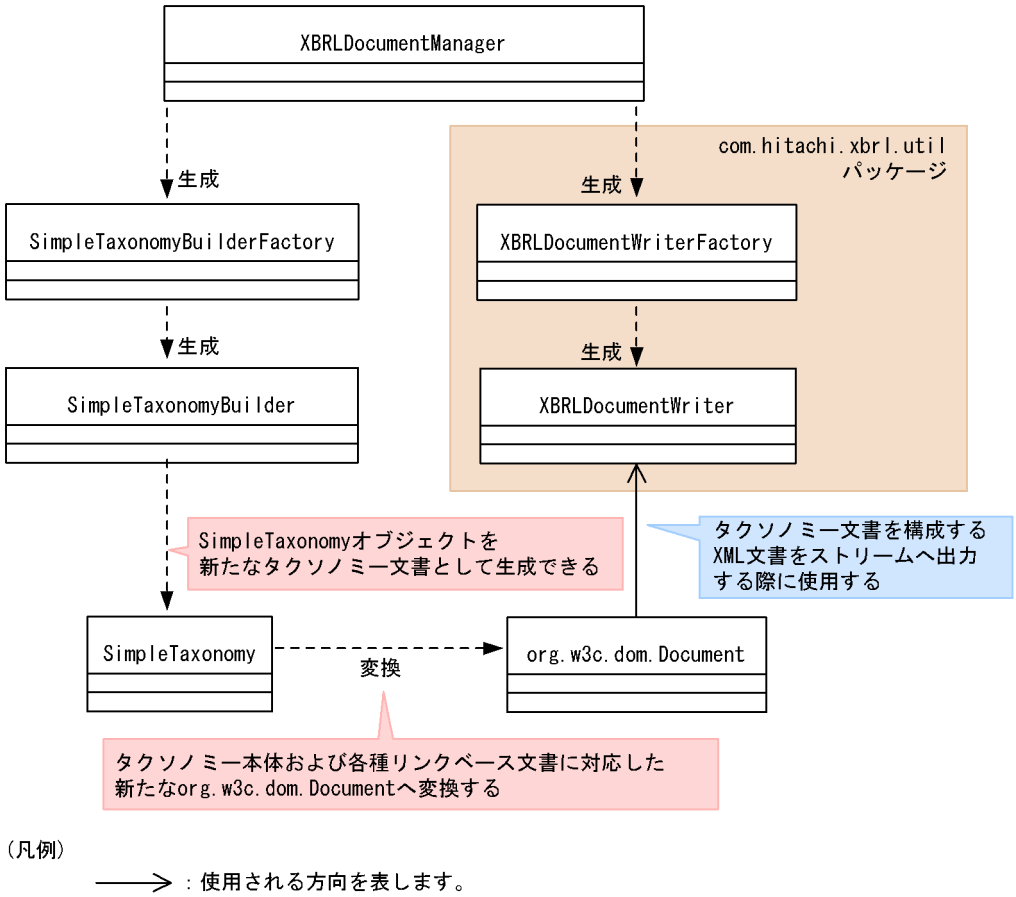
→▷ ：スーパークラスとサブクラスの関係を表します。

1　：多重度が1を表します。

0..1　：多重度が0または1を表します。

単純なタクソノミー文書の XBRL 文書管理機能のオブジェクトモデルを表現した UML のクラス図を次の図に示します。

図 6-9 単純なタクソノミー文書の XBRL 文書管理機能のオブジェクトモデルを表現した UML のクラス図



6.1.4 ノード

インスタンス文書またはタクソノミー文書の構成要素を表すオブジェクトを，ノードといいます。XBRL DOM のオブジェクトモデルでは，XBRLInstanceNode インターフェースまたは XBRLTaxonomyNode インターフェースを継承する各種のインターフェースを実装したオブジェクトによって，ノードが表現されます。

（1） XBRL2.0 の場合

XBRL2.0 の場合の，ノードの表す各インターフェースの継承関係を次の図に示します。

図 6-10　ノードの表す各インターフェースの継承関係（XBRL2.0 の場合）

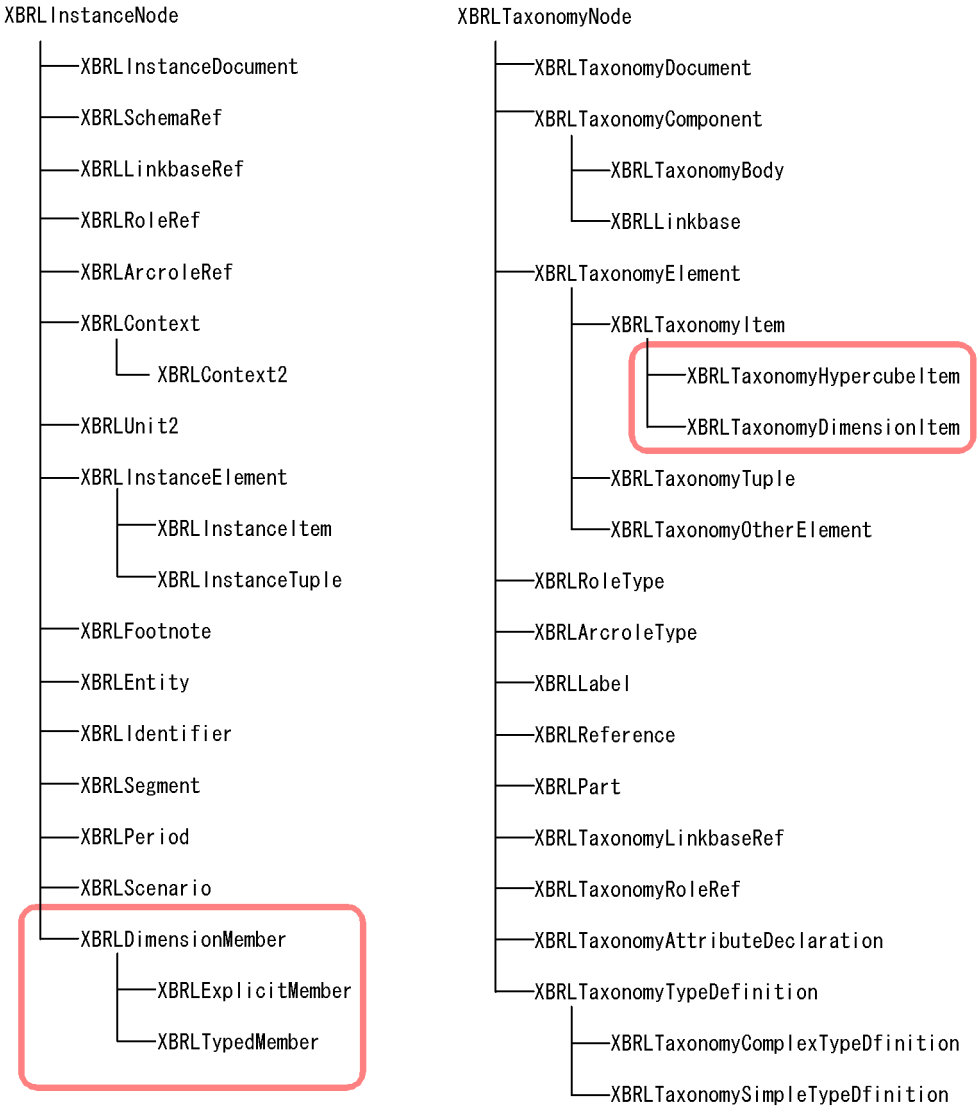
```
XBRLInstanceNode
├──XBRLInstanceDocument
├──XBRLContext
│    ├──XBRLNumericContext
│    └──XBRLNonNumericContext
├──XBRLInstanceElement
│    ├──XBRLInstanceItem
│    └──XBRLInstanceTuple
├──XBRLFootnote
├──XBRLEntity
├──XBRLIdentifier
├──XBRLSegment
├──XBRLPeriod
├──XBRLUnit
├──XBRLMeasureOrOperator
│    ├──XBRLMeasure
│    └──XBRLOperator
├──XBRLScenario
└──XBRLLinkbaseRef
```

```
XBRLTaxonomyNode
├──XBRLTaxonomyDocument
├──XBRLTaxonomyComponent
│    ├──XBRLTaxonomyBody
│    └──XBRLLinkbase
├──XBRLTaxonomyElement
│    ├──XBRLTaxonomyItem
│    ├──XBRLTaxonomyTuple
│    └──XBRLTaxonomyOtherElement
├──XBRLLabel
├──XBRLReference
└──XBRLPart
```

(2) XBRL2.1 の場合

XBRL2.1 の場合の，ノードの表す各インターフェースの継承関係を次の図に示します。

図 6-11　ノードの表す各インターフェースの継承関係（XBRL2.1 の場合）



（凡例）

：XBRL Dimensions1.0の場合の継承関係を表します。

リンクの構造を表す XBRLRelationship インターフェースの継承関係を次の図に示します。

図 6-12　リンクの構造を表す XBRLRelationship インターフェースの継承関係

```
XBRLRelationship
    ├──XBRLCalculationRelationship
    ├──XBRLDefinitionRelationship
    ├──XBRLPresentationRelationship
    ├──XBRLLabelRelationship
    ├──XBRLReferenceRelationship
    └──XBRLFootnoteRelationship
```

単純なタクソノミー文書での，SimpleNode インターフェースの継承関係を次の図に示します。

図 6-13　単純なタクソノミー文書での SimpleNode インターフェースの継承関係

```
SimpleNode
    ├──SimpleTaxonomy
    ├──SimpleElementDefinition
    │      └──SimpleConceptDefinition
    │              └──SimpleItemDefinition
    ├──SimpleLabel
    └──SimpleEdge
           └──SimpleLabelEdge
```

6.2 API 一覧（パッケージ順）

XBRL 文書操作用 API を，パッケージ，インターフェースまたはクラス，メソッドの順に，一覧で表示しています。

6.2.1 com.hitachi.xbrl パッケージ一覧

● XBRLParseErrorHandler インターフェース

warning(XBRLDOMException exception)　警告の通知を受け取ります。

error(XBRLDOMException exception)　回復できるエラーの通知を受け取ります。

fatalError(XBRLDOMException exception)　致命的なエラーの通知を受け取ります。

● XBRLDocumentManager クラス

init()　デフォルト値を使用して，XBRL DOM を初期化します。

init(java.lang.String path)　指定したプロパティファイルを使用して，XBRL DOM を初期化します。

init(java.util.Properties properties)　指定したプロパティリストを使用して，XBRL DOM を初期化します。

getInstance()　XBRLDocumentManager を実装したクラスのインスタンスを取得します。

newInstanceDocumentBuilderFactory()　XBRLInstanceDocumentBuilderFactory オブジェクトを生成します。

newInstanceDocumentWriterFactory()　XBRLInstanceDocumentWriterFactory オブジェクトを生成します。

newTaxonomyDocumentBuilderFactory()　XBRLDocumentManager の新しいインスタンスを取得します。

newDocumentWriterFactory()　XBRLDocumentWriterFactory オブジェクトを生成します。

newSimpleTaxonomyBuilderFactory()　SimpleTaxonomyBuilderFactory オブジェクトを生成します。

clearCache()　キャッシュされているすべてのタクソノミー文書を，キャッシュから削除します。

● XBRLDOMException クラス

getCode()　エラーコードを取得します。

● XBRLDOMRuntimeException クラス

getCode()　エラーコードを取得します。

● XBRLDOMUserException クラス

XBRLDOMUserException(int code)　XBRLDOMUserException オブジェクトを構築します。

XBRLDOMUserException(int code, java.lang.String message)
XBRLDOMUserException オブジェクトを構築します。

XBRLDOMUserException(int code, java.lang.Throwable cause)
XBRLDOMUserException オブジェクトを構築します。

XBRLDOMUserException(int code, java.lang.String message, java.lang.Throwable cause)　XBRLDOMUserException オブジェクトを構築します。

● XBRLDimensionException クラス

getQName()　XBRL Dimensions で規定されているエラーコードを表す QName を取得します。

● XBRLDimensionUserException クラス

XBRLDimensionUserException(int code, javax.xml.namespace.QName qname)
XBRLDimensionUserException オブジェクトを構築します。

XBRLDimensionUserException(int code, java.lang.String message, javax.xml.namespace.QName qname)　XBRLDimensionUserException オブジェクトを構築します。

XBRLDimensionUserException(int code, java.lang.Throwable cause, javax.xml.namespace.QName qname)　XBRLDimensionUserException オブジェクトを構築します。

XBRLDimensionUserException(int code, java.lang.String message, java.lang.Throwable cause, javax.xml.namespace.QName qname)
XBRLDimensionUserException オブジェクトを構築します。

6.2.2　com.hitachi.xbrl.instance パッケージ一覧

● XBRLArcroleRef インターフェース

getHref()　<link:arcroleRef> 要素の xlink:href 属性の値を取得します。

getArcroleURI()　<link:arcroleRef> 要素の arcroleURI 属性の値を取得します。

getBase()　<link:arcroleRef> 要素の xml:base 属性の値を取得します。

getArcroleType()　<link:arcroleRef> 要素が指す <link:arcroleType> 要素を取得します。

● XBRLContext インターフェース

isNumeric()　コンテキスト要素が数値か非数値かを調べます。

getId()　コンテキスト要素の id 属性の値を取得します。

setId(java.lang.String id)　コンテキスト要素の id 属性の値を設定します。

getEntity()　エンティティ要素を取得します。

setEntity(XBRLEntity entity)　エンティティ要素を設定します。

getPeriod()　時期要素を取得します。

setPeriod(XBRLPeriod period)　時期要素を設定します。

getScenario()　シナリオ要素を取得します。

setScenario(XBRLScenario scenario)　シナリオ要素を設定します。

getItems()　このコンテキスト要素を参照しているアイテムのリストを取得します。

getItems(java.lang.String namespaceURI, java.lang.String name)　このコンテキスト要素を参照している，指定した名前を持つアイテムのリストを取得します。

● XBRLContext2 インターフェース

● XBRLEntity インターフェース

getIdentifier()　識別子要素を取得します。

setIdentifier(XBRLIdentifier identifier)　識別子要素を設定します。

getSegment()　セグメント要素を取得します。

setSegment(XBRLSegment segment)　セグメント要素を設定します。

● XBRLFootnote インターフェース

getFootnoteLinkRole()　脚注が属する脚注リンクのロールを取得します。

getLang()　脚注の言語を取得します。

setLang(java.lang.String lang)　脚注の言語を設定します。

getRole()　脚注のロールを取得します。

setRole(java.lang.String role)　脚注のロールを設定します。

getTitle()　脚注のタイトルを取得します。

setTitle(java.lang.String title)　脚注のタイトルを設定します。

getId()　脚注の id 属性を取得します。

setId(java.lang.String id)　脚注の id 属性を設定します。

getValue()　脚注の内容を表す文字列を取得します。

setValue(java.lang.String value)　脚注の内容を表す文字列を設定します。

getContainerNode()　脚注の内容を保持する org.w3c.dom.Node オブジェクトを取得します。

getElements()　脚注と関連づけられている要素のリストを取得します。

getElements(java.lang.String namespaceURI, java.lang.String name)　この脚注と関連づけられている，指定された名前を持つ要素のリストを取得します。

● XBRLIdentifier インターフェース

getScheme()　識別子要素の scheme 属性の値を取得します。

setScheme(java.lang.String namespaceURI)　識別子要素の scheme 属性の値を設定します。

getValue()　識別子要素の内容を取得します。

setValue(java.lang.String value)　識別子要素の内容を設定します。

● XBRLInstanceDocument インターフェース

getOwnerDocumentManager()　インスタンス文書を所有する XBRLDocumentManager オブジェクトを取得します。

getXBRLVersion()　インスタンス文書が準拠している XBRL のバージョンを取得します。

getPTVInstance()　インスタンス文書の解析時に生成された PTVI を取得します。

getIdManager()　インスタンス文書の ID を管理する XBRLIdManager オブジェクトを取得します。

getTaxonomyDocument()　インスタンス文書が参照するタクソノミー文書を取得します。

getInstanceRoot()　インスタンス文書のルート要素を表す org.w3c.dom.Element オブジェクトを取得します。

getNamespaceDeclarations()　インスタンス文書に記述されている名前空間宣言のリ

ストを取得します。

getNamespaceDeclarationByURI(java.lang.String uri)　指定した名前空間 URI の名前空間宣言を取得します。

getNamespaceDeclarationByPrefix(java.lang.String prefix)　指定した名前空間接頭辞の名前空間宣言を取得します。

addNamespaceDeclaration(XBRLNamespaceDeclaration newNamespaceDeclaration)　インスタンス文書のルート要素に名前空間宣言を追加します。

removeNamespaceDeclaration(XBRLNamespaceDeclaration oldNamespaceDeclaration)　インスタンス文書から名前空間宣言を削除します。

getSchemaLocations()　インスタンス文書のルート要素の xsi:schemaLocation 属性に記述されている名前空間 URI と，スキーマ文書の位置を示す URI の組を表すオブジェクトのリストを取得します。

getSchemaLocation(java.lang.String namespaceURI)　インスタンス文書のルート要素の xsi:schemaLocation 属性に記述されている名前空間 URI と，スキーマ文書の位置を示す URI の組を表すオブジェクトのリストの中から，指定した名前空間 URI のオブジェクトを取得します。

addSchemaLocation(XBRLSchemaLocation newSchemaLocation)　インスタンス文書のルート要素の xsi:schemaLocation 属性に，名前空間 URI とスキーマ文書の位置を示す URI の組を追加します。

removeSchemaLocation(XBRLSchemaLocation oldSchemaLocation)　インスタンス文書のルート要素の xsi:schemaLocation 属性から，指定された名前空間 URI とスキーマ文書の位置を示す URI の組を削除します。

getBase()　インスタンス文書のルート要素の xml:base 属性の値を取得します。

setBase(java.lang.String base)　インスタンス文書のルート要素の xml:base 属性の値を設定します。

getId()　インスタンス文書のルート要素の id 属性の値を取得します。

setId(java.lang.String id)　インスタンス文書のルート要素の id 属性の値を設定します。

getSchemaRefs()　インスタンス文書に記述されている <link:schemaRef> 要素のリストを取得します。

addSchemaRef(XBRLSchemaRef newSchemaRef)　インスタンス文書に <link:schemaRef> 要素を追加します。

createSchemaRef(java.lang.String href, java.lang.String base)　<link:schemaRef>

要素を表す XBRLSchemaRef オブジェクトを新規に生成します。

getLinkbaseRefs()　インスタンス文書に記述されている <link:linkbaseRef> 要素のリストを取得します。

getLinkbaseRefs(java.lang.String role)　インスタンス文書に記述されている <link:linkbaseRef> 要素のうち，引数 role と同じ値の xlink:role 属性を持つ <link:linkbaseRef> 要素のリストを取得します。

addLinkbaseRef(XBRLLinkbaseRef newLinkbaseRef)　インスタンス文書に <link:linkbaseRef> 要素を追加します。

removeLinkbaseRef(XBRLLinkbaseRef oldLinkbaseRef)　インスタンス文書に記述されている <link:linkbaseRef> 要素を削除します。

createLinkbaseRef(java.lang.String role, java.lang.String href)　<link:linkbaseRef> 要素を表す XBRLLinkbaseRef オブジェクトを新規に生成します。

createLinkbaseRef(java.lang.String role, java.lang.String href, java.lang.String base)　<link:linkbaseRef> 要素を表す XBRLLinkbaseRef オブジェクトを新規に生成します。

getRoleRefs()　インスタンス文書に記述されている <link:roleRef> 要素のリストを取得します。

getRoleRef(java.lang.String roleURI)　インスタンス文書に記述されている <link:roleRef> 要素のうち，指定した roleURI と同じ値の roleURI 属性を持つ <link:roleRef> 要素を取得します。

addRoleRef(XBRLRoleRef newRoleRef)　インスタンス文書に <link:roleRef> 要素を追加します。

createRoleRef(java.lang.String href, java.lang.String roleURI, java.lang.String base)　<link:roleRef> 要素である XBRLRoleRef オブジェクトを新規に生成します。

getArcroleRefs()　インスタンス文書に記述されている <link:arcroleRef> 要素のリストを取得します。

getArcroleRef(java.lang.String arcroleURI)　インスタンス文書に記述されている <link:arcroleRef> 要素のうち，引数 arcroleURI と同じ値の arcroleURI 属性を持つ <link:arcroleRef> 要素を取得します。

addArcroleRef(XBRLArcroleRef newArcroleRef)　インスタンス文書に <link:arcroleRef> 要素を追加します。

createArcroleRef(java.lang.String href, java.lang.String arcroleURI, java.lang.String base)　<link:arcroleRef> 要素である XBRLArcroleRef オブジェクトを新規に生成します。

getContexts()　インスタンス文書に記述されているコンテキスト要素のリストを取得します。

addContext(XBRLContext newContext)　インスタンス文書にコンテキスト要素を追加します。

removeContext(XBRLContext oldContext)　インスタンス文書からコンテキスト要素を削除します。

createNumericContext(XBRLEntity entity, XBRLPeriod period, XBRLUnit unit, XBRLScenario scenario, int precision, boolean cwa)　数値コンテキスト要素を新規に生成します。

createNonNumericContext(XBRLEntity entity, XBRLPeriod period, XBRLUnit unit, XBRLScenario scenario)　非数値コンテキスト要素を新規に生成します。

createContext(XBRLEntity entity, XBRLPeriod period, XBRLScenario scenario)　コンテキスト要素を新規に生成します。

getUnits()　インスタンス文書に記述されている単位要素のリストを取得します。

addUnit(XBRLUnit2 newUnit)　インスタンス文書に単位要素を追加します。

removeUnit(XBRLUnit2 oldUnit)　インスタンス文書から単位要素を削除します。

createUnit(XBRLQName[] measures)　単位要素である XBRLUnit2 オブジェクトを新規に生成します。

createUnit(XBRLQName[] numerator, XBRLQName[] denominator)　分数を指定して，単位要素である XBRLUnit2 オブジェクトを新規に生成します。

getChildElements()　インスタンス文書のルート要素の子要素のリストを取得します。

getChildElements(java.lang.String namespaceURI, java.lang.String name)　インスタンス文書のルート要素の子要素のうち，指定した名前を持つ子要素のリストを取得します。

appendChildElement(XBRLInstanceElement newElement)　インスタンス文書のルート要素の子要素のリストの末尾に，指定した要素を追加します。

insertChildElementBefore(XBRLInstanceElement newElement, XBRLInstanceElement refElement)　引数 newElement に指定した要素を，このインスタンス文書のルート要素の既存の子要素の前に挿入します。

replaceChildElement(XBRLInstanceElement newElement, XBRLInstanceElement oldElement)　引数 oldElement に指定した子要素を，引数 newElement に指定した要素で置き換えます。

removeChildElement(XBRLInstanceElement oldElement)　インスタンス文書のルー

ト要素の子要素を削除します。

getElements()　このインスタンス文書に存在する要素のリストを取得します。

getElements(java.lang.String namespaceURI, java.lang.String name)　このインスタンス文書に存在する，指定した名前の要素のリストを取得します。

createItem(java.lang.String namespaceURI, java.lang.String name, XBRLContext context)　アイテムを新規に生成します。

createNumericItem(java.lang.String namespaceURI, java.lang.String name, XBRLContext context, XBRLUnit2 unit, XBRLPrecision precision, XBRLDecimals decimals)　数値アイテムを新規に生成します。

createNonNumericItem(java.lang.String namespaceURI, java.lang.String name, XBRLContext2 context)　非数値アイテムを新規に生成します。

createTuple(java.lang.String namespaceURI, java.lang.String name)　タプルを新規に生成します。

createEntity(XBRLIdentifier identifier, XBRLSegment segment)　エンティティ要素を表す XBRLEntity オブジェクトを新規に生成します。

createPeriod(int periodType, XBRLDateUnion instant, XBRLDateUnion startDate, XBRLDateUnion endDate, XBRLDuration duration)　時期要素である XBRLPeriod オブジェクトを新規に生成します。

createPeriod()　時期要素である XBRLPeriod オブジェクトを新規に生成します。

createPeriod(XBRLDateUnion instant)　時期要素である XBRLPeriod オブジェクトを新規に生成します。

createPeriod(XBRLDateUnion startDate, XBRLDateUnion endDate)　時期要素である XBRLPeriod オブジェクトを新規に生成します。

createUnit(XBRLMeasureOrOperator measureOrOperator)　単位要素である XBRLUnit オブジェクトを新規に生成します。

createScenario(java.lang.String name)　シナリオ要素である XBRLScenario オブジェクトを新規に生成します。

createScenario()　シナリオ要素である XBRLScenario オブジェクトを新規に生成します。

createIdentifier(java.lang.String namespaceURI, java.lang.String value)　識別子要素である XBRLIdentifier オブジェクトを新規に生成します。

createSegment(java.lang.String name)　セグメント要素である XBRLSegment オブジェクトを新規に生成します。

createSegment()　セグメント要素である XBRLSegment オブジェクトを新規に生成します。

createMeasure(XBRLQName value)　計量要素である XBRLMeasure オブジェクトを新規に生成します。

createOperator(java.lang.String kind, XBRLMeasureOrOperator lhs, XBRLMeasureOrOperator rhs)　演算子要素である XBRLOperator オブジェクトを新規に生成します。

getFootnotes()　インスタンス文書に記述されている脚注のリストを取得します。

getFootnotesByLang(java.lang.String lang)　インスタンス文書に記述されている脚注のうち，指定した言語の脚注を取得します。

getFootnotesByRole(java.lang.String role)　インスタンス文書に記述されている脚注のうち，指定したロールの脚注を取得します。

getFootnotes(java.lang.String lang, java.lang.String role)　インスタンス文書に記述されている脚注のうち，指定した言語およびロールの脚注を取得します。

addFootnote(XBRLFootnote newFootnote)　インスタンス文書に脚注を追加します。

removeFootnote(XBRLFootnote oldFootnote)　インスタンス文書から脚注を削除します。

createFootnote(java.lang.String lang, java.lang.String role, java.lang.String title)
脚注である XBRLFootnote オブジェクトを新規に生成します。

createFootnote(java.lang.String footnoteLinkRole, java.lang.String lang, java.lang.String role, java.lang.String title)　脚注である XBRLFootnote オブジェクトを新規に生成します。カスタムロールを指定できます。

importNode(XBRLInstanceNode sourceNode, boolean deep)　ほかの XBRLInstanceDocument オブジェクトで生成されたノードをインポートします。

cloneNode(org.w3c.dom.Document w3cdomDocument, boolean deep)　インスタンス文書を複製します。

createExplicitMember(XBRLQName dimension, XBRLQName value)
<xbrldi:explicitMember> 要素を表す XBRLExplicitMember オブジェクトを新規に生成します。

createTypedMember(XBRLQName dimension)　<xbrldi:typedMember> 要素を表す XBRLTypedMember オブジェクトを新規に生成します。

getDimensionsVersion()　インスタンス文書が準拠している XBRL Dimensions のバージョンを取得します。

● XBRLInstanceDocumentBuilder インターフェース

getOwnerDocumentManager()　XBRLInstanceDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

getXBRLVersion()　XBRLInstanceDocumentBuilder オブジェクトが解析または作成するインスタンス文書の XBRL バージョンを取得します。

isValidating()　XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時に文書の妥当性を検証するかどうかを調べます。

isDelayLoading()　XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時にリンクベースの遅延読み込みをするかどうかを調べます。

getExternalSchemaRefs()　XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時に参照するタクソノミー本体の URI のリストを取得します。

getProperty()　XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析または作成する際に使用するプロパティの値を取得します。

getTaxonomyDocumentBuilder()　タクソノミー文書を読み込むために使用する XBRLTaxonomyDocumentBuilder オブジェクトを取得します。

setTaxonomyDocumentBuilder(XBRLTaxonomyDocumentBuilder taxonomyDocumentBuilder)　タクソノミー文書を読み込むために使用する XBRLTaxonomyDocumentBuilder オブジェクトを設定します。

setEntityResolver(org.xml.sax.EntityResolver entityResolver)　インスタンス文書に存在するエンティティを解決するためのエンティティリゾルバ（org.xml.sax.EntityResolver オブジェクト）を設定します。

setErrorHandler(XBRLParseErrorHandler errorHandler)　インスタンス文書解析時のエラーを受け取るエラーハンドラ（XBRLParseErrorHandler オブジェクト）を設定します。

parse(org.xml.sax.InputSource is)　指定した入力ソースからインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

parse(java.io.InputStream is)　指定した入力ストリームをインスタンス文書として解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

parse(java.io.InputStream is, java.lang.String systemId)　指定した入力ストリームをインスタンス文書として解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

parse(java.io.File file)　指定したファイルをインスタンス文書として解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

parse(java.lang.String uri)　指定した URI をインスタンス文書として解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

parse(org.w3c.dom.Element instanceRoot)　指定した要素をルート要素とするインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

parse(org.w3c.dom.Element instanceRoot, java.lang.String systemId)　指定した要素をルート要素とするインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

parse2(org.xml.sax.InputSource is)　指定した入力ソースに含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

parse2(java.io.InputStream is)　指定した入力ストリームに含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

parse2(java.io.InputStream is, java.lang.String systemId)　指定した入力ストリームに含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

parse2(java.io.File file)　指定したファイルに含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

parse2(java.lang.String uri)　指定した URI に含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument)　空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを新規に生成します。

newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument, java.lang.String systemId)　空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを新規に生成します。

newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument, org.w3c.dom.Document w3cdomDocument)　XML 文書の任意の位置にインスタンス文書を追加する場合に使用します。空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを新規に生成します。

newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument, org.w3c.dom.Document w3cdomDocument, java.lang.String systemId)　XML 文書の任意の位置にインスタンス文書を追加する場合に使用します。空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを新規に生成します。

getDimensionsVersion()　XBRLInstanceDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているインスタンス文書を解析または作成するのかを取得します。

setCreatingSchemaLocations(boolean creating)　空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを生成するときに，ルート要素の xsi:schemaLocation 属性の初期値として，タクソノミー文書に含まれる各タクソノミー本体の対象名前空間 URI と位置を示す絶対 URI の組を設定するかどうかを指定します。

isCreatingSchemaLocations()　空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを生成するときに，ルート要素の xsi:schemaLocation 属性の初期値として，タクソノミー文書に含まれる各タクソノミー本体の対象名前空間 URI と位置を示す絶対 URI の組を設定するかどうかを示します。

● XBRLInstanceDocumentBuilderFactory インターフェース

getOwnerDocumentManager()　XBRLInstanceDocumentBuilderFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

setXBRLVersion(int version)　どの XBRL バージョンのインスタンス文書を解析または作成するのかを設定します。

getXBRLVersion()　どの XBRL バージョンのインスタンス文書を解析または作成するのかを取得します。

setValidating(boolean flag)　インスタンス文書の解析時に文書の妥当性を検証するかどうかを設定します。

isValidating()　インスタンス文書の解析時に文書の妥当性を検証するかどうかを調べます。

setDelayLoading(boolean flag)　インスタンス文書の解析時にリンクベースの遅延読み込みをするかどうかを設定します。

isDelayLoading()　インスタンス文書の解析時にリンクベースの遅延読み込みをするかどうかを調べます。

setExternalSchemaRefs(java.lang.String[] uris)　インスタンス文書の解析時に参照するタクソノミー本体の URI のリストを設定します。

getExternalSchemaRefs()　インスタンス文書の解析時に参照するタクソノミー本体の URI のリストを取得します。

setIgnoringComments(boolean flag)　インスタンス文書の解析時にコメントを無視するかどうかを設定します。

isIgnoringComments()　インスタンス文書の解析時にコメントを無視するかどうかを調べます。

setCoalescing(boolean flag)　インスタンス文書の解析時に CDATA ノードを Text ノードに変換するかどうかを設定します。

isCoalescing()　インスタンス文書の解析時に CDATA ノードを Text ノードに変換するかどうかを調べます。

newInstanceDocumentBuilder()　現在設定されているパラメタを使用して，XBRLInstanceDocumentBuilder のインスタンスを新規に生成します。

setProperty(java.lang.String name, java.lang.Object value)
XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析または作成する際に使用するプロパティの値を設定します。

getProperty(java.lang.String name)　XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析または作成する際に使用するプロパティの値を取得します。

setDimensionsVersion(int version)　XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているインスタンス文書を解析または作成するのかを指定します。

getDimensionsVersion()　XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているインスタンス文書を解析または作成するのかを取得します。

● XBRLInstanceDocumentWriter インターフェース

getOwnerDocumentManager()　XBRLInstanceDocumentWriter オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

setEncoding(java.lang.String encoding)　インスタンス文書を書き出す際に使用するエンコーディングを設定します。

getEncoding()　インスタンス文書を書き出す際に使用するエンコーディングを取得します。

setIndent(boolean indent)　インスタンス文書を書き出す際にインデントをするかどうかを設定します。

isIndent()　インスタンス文書を書き出す際にインデントをするかどうかを調べます。

write(java.io.OutputStream os, XBRLInstanceDocument doc)　インスタンス文書をストリームに書き出します。

● XBRLInstanceDocumentWriterFactory インターフェース

getOwnerDocumentManager()　XBRLInstanceDocumentWriterFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

newInstanceDocumentWriter()　現在設定されているパラメタを使用して，XBRLInstanceDocumentWriter のインスタンスを新規に生成します。

● XBRLInstanceElement インターフェース

getTaxonomyElement()　要素の要素宣言を取得します。

getElementType()　オブジェクトの型を表すコードを取得します。

getNamespaceURI()　要素が属する名前空間 URI を取得します。

getPrefix()　要素の名前空間接頭辞を取得します。

getName()　要素の名前を取得します。

getId()　要素の id 属性の値を取得します。

setId(java.lang.String id)　要素の id 属性の値を設定します。

getLabels()　要素に関連づけられているラベルのリストを取得します。

getLabelsByLang(java.lang.String lang)　要素に関連づけられているラベルのうち，指定した言語のラベルのリストを取得します。

getLabelsByRole(java.lang.String role)　要素に関連づけられているラベルのうち，指定したロールを持つラベルのリストを取得します。

getLabel(java.lang.String lang, java.lang.String role)　要素に関連づけられているラベルのうち，指定した言語とロールを持つラベルを取得します。

getLabels(java.lang.String lang, java.lang.String role)　要素に関連づけられているラベルのうち，指定した言語とロールを持つラベルのリストを取得します。

getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole)　要素の要素宣言とラベルの関係を表す XBRLLabelRelationship オブジェクトのリストを取得します。

getReferences()　要素に関連づけられている文献への参照のリストを取得します。

getReferences(java.lang.String role)　要素に関連づけられている文献への参照のうち，指定したロールを持つ文献への参照のリストを取得します。

getReferenceRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole)　要素の要素宣言と文献への参照の関係を表す XBRLReferenceRelationship オブジェクトのリストを取得します。

isNillable()　要素の要素宣言の nillable 属性の値を取得します。

isNil()　要素の xsi:nil 属性の値を取得します。

setNil(java.lang.Boolean nil)　要素の xsi:nil 属性の値を設定します。

getFootnotes()　要素に関連づけられている脚注のリストを取得します。

getFootnotesByLang(java.lang.String lang)　要素に関連づけられている脚注のうち，指定した言語の脚注のリストを取得します。

getFootnotesByRole(java.lang.String role)　要素に関連づけられている脚注のうち，指定したロールの脚注のリストを取得します。

getFootnotes(java.lang.String lang, java.lang.String role)　要素に関連づけられている脚注のうち，指定した言語およびロールの脚注のリストを取得します。

addFootnote(XBRLFootnote newFootnote)　要素に脚注を関連づけます。

addFootnote(XBRLFootnote newFootnote, java.lang.String arcrole, java.math.BigDecimal order)　要素に脚注を関連づけます。カスタムアークロールを使用できます。

removeFootnote(XBRLFootnote oldFootnote)　要素と脚注の関連づけを削除します。

removeFootnote(XBRLFootnote oldFootnote, java.lang.String arcrole, java.math.BigDecimal order)　要素と脚注の関連づけを削除します。

getFootnoteRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole)　要素と脚注の関係を表す XBRLFootnoteRelationship オブジェクトのリストを取得します。

isNilAttributeSpecified()　要素に xsi:nil 属性が定義されていたかどうかを調べます。

getDefinitionRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole)　要素の要素宣言とほかの要素宣言の関係を表す XBRLDefinitionRelationship オブジェクトのリストを取得します。

● XBRLInstanceItem インターフェース

getTaxonomyItem()　アイテムの要素宣言を取得します。

getItemType()　アイテムのデータ型を表すコードを取得します。

getBaseItemType()　アイテムのベースデータ型を表すコードを取得します。

isNumeric()　アイテムが数値または非数値のどちらであるかを調べます。

getContext()　アイテムが参照するコンテキスト要素を取得します。

setContext(XBRLContext context)　アイテムが参照するコンテキスト要素を設定します。

getUnit()　アイテムが参照する単位要素を取得します。

setUnit(XBRLUnit2 unit)　アイテムが参照する単位要素を設定します。

getPrecision()　アイテムの precision 属性値を取得します。

setPrecision(XBRLPrecision precision)　アイテムの precision 属性値を設定します。

getDecimals()　アイテムの decimals 属性値を取得します。

setDecimals(XBRLDecimals decimals)　アイテムの decimals 属性値を設定します。

hasValue()　アイテムに値が設定されているかどうかを調べます。

getValue()　アイテムの値を取得します。

setValue(java.lang.Object value)　アイテムの値を設定します。

getString()　アイテムの値の文字列表現を取得します。

setString(java.lang.String value)　アイテムの値の文字列表現を設定します。

getDimensionalRelationshipSet()　アイテムの要素宣言に関連付いている DRS を取得します。

validateDimensions()　DRS に含まれているすべてのロールでアイテムに対して XBRL Dimensions の妥当性検証を実行します。

validateDimensions(java.lang.String role)　DRS に含まれているロールのうち，引数 role に指定されたロールでアイテムに対して XBRL Dimensions の妥当性検証を実行します。

● XBRLInstanceNode インターフェース

getNodeType()　ノードの型を表すコードを取得します。

getOwnerInstanceDocument()　ノードを所有する XBRLInstanceDocument オブジェクトを取得します。

isActive()　ノードがアクティブかどうか（インスタンス文書に存在し，ツリーをたどって到達できるか）を調べます。

getParentNode()　ノードが関連づけられている親ノードを取得します。

cloneNode(boolean deep)　ノードを複製します。

getW3CDOMNode()　ノードに対応する org.w3c.dom.Node オブジェクトを取得します。

● XBRLInstanceTuple インターフェース

getTaxonomyTuple()　タプルの要素宣言を取得します。

getChildElements()　タプルのすべての子要素のリストを取得します。

getChildElements(java.lang.String namespaceURI, java.lang.String name)　指定し

た名前の子要素のリストを取得します。

appendChildElement(XBRLInstanceElement newElement)　タプルの子要素のリストの末尾に，指定した子要素を追加します。

insertChildElementBefore(XBRLInstanceElement newElement, XBRLInstanceElement refElement)　引数 newElement に指定された要素を，このタプルの既存の子要素の前に挿入します。

replaceChildElement(XBRLInstanceElement newElement, XBRLInstanceElement oldElement)　引数 oldElement に指定した子要素を，引数 newElement に指定した要素で置き換えます。

removeChildElement(XBRLInstanceElement oldElement)　タプルの子要素を削除します。

● XBRLLinkbaseRef インターフェース

getRole()　<link:linkbaseRef> 要素の xlink:role 属性の値を取得します。

getHref()　<link:linkbaseRef> 要素の xlink:href 属性の値を取得します。

getBase()　<link:linkbaseRef> 要素の xml:base 属性の値を取得します。

● XBRLMeasure インターフェース

getValue()　計量要素の内容を取得します。

setValue(XBRLQName value)　計量要素の内容を設定します。

● XBRLMeasureOrOperator インターフェース

getMeasureOrOperatorType()　オブジェクトが計量要素または演算子要素のどちらであるかを調べます。

● XBRLNonNumericContext インターフェース

getUnit()　非数値コンテキスト要素の単位要素を取得します。

setUnit(XBRLUnit unit)　非数値コンテキスト要素の単位要素を設定します。

● XBRLNumericContext インターフェース

getPrecision()　数値コンテキスト要素の precision 属性の値を取得します。

setPrecision(int precision)　数値コンテキスト要素の precision 属性の値を設定します。

getCwa()　数値コンテキスト要素の cwa 属性の値を取得します。

setCwa(boolean cwa)　数値コンテキスト要素の cwa 属性の値を設定します。

getUnit()　数値コンテキスト要素の単位要素を取得します。

setUnit(XBRLUnit unit)　数値コンテキスト要素の単位要素を設定します。

● XBRLOperator インターフェース

getKind()　演算子の種別を取得します。

setKind(java.lang.String kind)　演算子の種別を設定します。

getLhs()　演算子の左辺を取得します。

setLhs(XBRLMeasureOrOperator lhs)　演算子の左辺を設定します。

getRhs()　演算子の右辺を取得します。

setRhs(XBRLMeasureOrOperator rhs)　演算子の右辺を設定します。

● XBRLPeriod インターフェース

getPeriodType()　時期要素の内容の意味を取得します。

getInstant()　時期要素の内容のある特定の時点を取得します。

getStartDate()　時期要素の内容の開始日を取得します。

getEndDate()　時期要素の内容の終了日を取得します。

getDuration()　時期要素の内容の継続期間を取得します。

● XBRLRoleRef インターフェース

getHref()　<link:roleRef> 要素の xlink:href 属性の値を取得します。

getRoleURI()　<link:roleRef> 要素の roleURI 属性の値を取得します。

getBase()　<link:roleRef> 要素の xml:base 属性の値を取得します。

getRoleType()　<link:roleRef> 要素の link:roleType 属性の値を取得します。

● XBRLScenario インターフェース

getName()　シナリオ要素の name 属性を取得します。

setName(java.lang.String name)　シナリオ要素の name 属性を設定します。

getContainerNode()　シナリオ要素の内容を保持するノードを取得します。

getDimensionMembers()　シナリオ要素の子要素である <xbrldi:explicitMember 要素> および <xbrldi:typedMember> 要素のリストを取得します。

getExplicitMembers()　シナリオ要素の子要素である <xbrldi:explicitMember> 要素のリストを取得します。

getTypedMembers()　シナリオ要素の子要素である <xbrldi:typedMember> 要素のリストを取得します。

appendDimensionMember(XBRLDimensionMember newMember)　シナリオ要素の子要素のリストの末尾に，引数 newMember に指定した要素を追加します。

insertDimensionMemberBefore(XBRLDimensionMember newMember, XBRLDimensionMember refMember)　引数 newMember に指定した要素を，引数 refMember に指定した子要素の前に挿入します。

replaceDimensionMember(XBRLDimensionMember newMember, XBRLDimensionMember oldMember)　引数 oldMember に指定した子要素を，引数 newMember に指定した要素で置き換えます。

removeDimensionMember(XBRLDimensionMember oldMember)　引数 oldMember に指定された子要素を，シナリオ要素の子要素のリストから削除します。

hasExplicitMember(XBRLQName dimension)　シナリオ要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在するかどうかを調べます。

hasExplicitMember(XBRLQName dimension, XBRLQName value)　シナリオ要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在するかどうかを調べます。

hasTypedMember(XBRLQName dimension)　シナリオ要素の子要素に，該当する <xbrldi:typedMember> 要素が存在するかどうかを調べます。

● XBRLSchemaRef インターフェース

getHref()　<link:schemaRef> 要素の xlink:href 属性の値を取得します。

getBase()　<link:schemaRef> 要素の xml:base 属性の値を取得します。

● XBRLSegment インターフェース

getName()　セグメント要素の name 属性を取得します。

setName(java.lang.String name)　セグメント要素の name 属性を設定します。

getContainerNode()　セグメント要素の内容を保持するノードを取得します。

getDimensionMembers()　セグメント要素の子要素である <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素のリストを取得します。

getExplicitMembers()　セグメント要素の子要素である <xbrldi:explicitMember> 要素のリストを取得します。

getTypedMembers()　セグメント要素の子要素である <xbrldi:typedMember> 要素のリストを取得します。

appendDimensionMember(XBRLDimensionMember newMember)　セグメント要素

の子要素のリストの末尾に，引数 newMember に指定した要素を追加します。

insertDimensionMemberBefore(XBRLDimensionMember newMember, XBRLDimensionMember refMember)　引数 newMember に指定した要素を，引数 refMember に指定した子要素の前に挿入します。

replaceDimensionMember(XBRLDimensionMember newMember, XBRLDimensionMember oldMember)　引数 oldMember に指定した子要素を，引数 newMember に指定した要素で置き換えます。

removeDimensionMember(XBRLDimensionMember oldMember)　引数 oldMember に指定した子要素を，セグメント要素の子要素のリストから削除します。

hasExplicitMember(XBRLQName dimension)　セグメント要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在するかどうかを調べます。

hasExplicitMember(XBRLQName dimension, XBRLQName value)　セグメント要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在するかどうかを調べます。

hasTypedMember(XBRLQName dimension)　セグメント要素の子要素に，該当する <xbrldi:typedMember> 要素が存在するかどうかを調べます。

● XBRLUnit インターフェース

getMeasureOrOperator()　計量要素または演算子要素を取得します。

setMeasureOrOperator(XBRLMeasureOrOperator measureOrOperator)　計量要素または演算子要素を設定します。

● XBRLUnit2 インターフェース

getId()　単位要素の id 属性の値を取得します。

setId(java.lang.String id)　単位要素の id 属性の値を設定します。

getUnitType()　単位要素の内容を表すコードを取得します。

getMeasures()　単位要素を構成する基本単位のリストを取得します。

getNumerator()　単位要素の分子を構成する基本単位のリストを取得します。

getDenominator()　単位要素の分母を構成する基本単位のリストを取得します。

getItems()　単位要素を参照しているアイテムのリストを取得します。

getItems(java.lang.String namespaceURI, java.lang.String name)　単位要素を参照している，指定した名前のアイテムのリストを取得します。

● XBRLDimensionMember インターフェース

getDimension()　<xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素

の dimension 属性の値を取得します。

setDimension(XBRLQName dimension)　<xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素の dimension 属性の値を設定します。

● XBRLExplicitMember インターフェース

getValue()　<xbrldi:explicitMember> 要素の内容を取得します。

setValue(XBRLQName value)　<xbrldi:explicitMember> 要素の内容を設定します。

● XBRLTypedMember インターフェース

getContainerNode()　<xbrldi:typedMember> 要素の内容を保持するノードを取得します。

6.2.3 com.hitachi.xbrl.link パッケージ一覧

● XBRLCalculationRelationship インターフェース

getWeight()　<link:calculationArc> 要素の weight 属性の値を取得します。

● XBRLDefinitionRelationship インターフェース

● XBRLFootnoteRelationship インターフェース

● XBRLLabelRelationship インターフェース

● XBRLPresentationRelationship インターフェース

getPreferredLabel()　<link:presentationArc> 要素の preferredLabel 属性の値を取得します。

● XBRLReferenceRelationship インターフェース

● XBRLRelationship インターフェース

getArcType()　関係を定義したアークの種類を調べます。

getExtendedLinkRole()　関係を定義したアークを含む拡張リンクのロールを取得します。

getArcrole()　関係を定義したアークのアークロールを取得します。

getOrder()　関係を定義したアークの order 属性の値を取得します。

getFrom()　関係の from 側のリソースを取得します。

getTo()　関係の to 側のリソースを取得します。

getPriority()　関係を定義したアークの priority 属性の値を取得します。

isOrderAttributeSpecified()　関係を定義したアークの order 属性が定義されているかを判定します。

● XBRLDimensionalRelationship インターフェース

● XBRLDimensionalRelationshipSet インターフェース

getTaxonomyItem()　DRS に関連づいているプライマリーアイテムの要素宣言を取得します。

getRoles()　DRS に含まれている has-hypercube 関係が定義された，extended 型要素の xlink:role 属性のリストを取得します。

getHasHypercubeRelationships(java.lang.String role)　DRS に含まれている has-hypercube 関係のうち，引数 role に指定した値と同じ xlink:role 属性を持つ extended 型要素に定義された，has-hypercube 関係のリストを取得します。

getHypercubeDimensionRelationships(XBRLHasHypercubeRelationship relationship)　DRS に含まれている hypercube-dimension 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結びついている hypercube-dimension 関係のリストを取得します。

getDimensionDomainRelationships(XBRLHypercubeDimensionRelationship relationship)　DRS に含まれている dimension-domain 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結びついている dimension-domain 関係のリストを取得します。

getDomainMemberRelationships(XBRLDimensionDomainRelationship relationship)　DRS に含まれている domain-member 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結びついている domain-member 関係のリストを取得します。

getDomainMemberRelationships(XBRLDomainMemberRelationship relationship)　DRS に含まれている domain-member 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結びついている domain-member 関係のリストを取得します。

getDimensionDefaultRelationship(XBRLHypercubeDimensionRelationship relationship)　DRS に含まれている dimension-default 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結びついている dimension-default 関係を取得します。

getDRSDimensions(java.lang.String role, int contextElementType)　DRS に含まれているディメンジョンの要素宣言のうち，インスタンス文書に記述できるディメンジョンの要素宣言のリストを返します。

getDRSDimensions(java.lang.String role)　DRS に含まれているディメンジョンの要

素宣言のうち，インスタンス文書に記述できるディメンジョンの要素宣言のリストを返します。

getDRSMembers(XBRLDRSDimension dimension)　DRS に含まれている要素宣言のうち，指定されたディメンジョンのメンバとして，インスタンス文書に記述できる要素宣言のリストを返します。

getDRSDefaultMember(XBRLDRSDimension dimension)　DRS に含まれている要素宣言のうち，インスタンス文書に記述できる要素宣言のリストを返します。

● XBRLDimensionDefaultRelationship インターフェース

● XBRLDimensionDomainRelationship インターフェース

getUsable()　<link:definitionArc> 要素の xbrldt:usable 属性の値を取得します。

getTargetRole()　<link:definitionArc> 要素の xbrldt:targetRole 属性の値を取得します。

● XBRLDomainMemberRelationship インターフェース

getUsable()　<link:definitionArc> 要素の xbrldt:usable 属性の値を取得します。

getTargetRole()　<link:definitionArc> 要素の xbrldt:targetRole 属性の値を取得します。

● XBRLDRSDimension インターフェース

getContextElementType()　<link:definitionArc> 要素の xbrldt:contextElement 属性の種類を取得します。

getRole()　extended 型要素の xlink:role 属性の値を取得します。

getTaxonomyDimensionItem()　ディメンジョンの要素宣言を取得します。

getQName()　ディメンジョンの要素宣言の QName を取得します。

● XBRLDRSMember インターフェース

getTaxonomyElement()　ディメンジョンのメンバを表す要素宣言を取得します。

getQName()　ディメンジョンのメンバを表す要素宣言の QName を取得します。

● XBRLHasHypercubeRelationship インターフェース

getContextElementType()　<link:definitionArc> 要素の xbrldt:contextElement 属性の種類を取得します。

getClosed()　<link:definitionArc> 要素の xbrldt:closed 属性の値を取得します。

getHasHypercubeRelationshipType()　<link:definitionArc> 要素の xlink:arcrole 属性の種類を取得します。

getTargetRole()　<link:definitionArc> 要素の xbrldt:targetRole 属性の値を取得します。

● XBRLHypercubeDimensionRelationship インターフェース

getTargetRole()　<link:definitionArc> 要素の xbrldt:targetRole 属性の値を取得します。

● XBRLIllegalCycleException クラス

getLinkName()　エラーが検出された関係ネットワークを定義している拡張リンクの種類を取得します。

getExtendedLinkRole()　エラーが検出された関係ネットワークを定義している拡張リンクのロールを取得します。

getArcrole()　エラーが検出された関係ネットワークを定義しているアークのアークロールを取得します。

getCyclesAllowed()　エラーが検出された関係ネットワークで許可されるサイクルの種類を取得します。

getErrorRoutes()　誤っていた経路を取得します。

getRelationRoutes()　誤っていた経路に関連するもう一つの経路を取得します。

● XBRLDimensionIllegalCycleException クラス

getLinkName()　エラーが検出された DRS の関係ネットワークを定義している拡張リンクの種類を取得します。

getExtendedLinkRole()　エラーが検出された DRS の関係ネットワークを定義している拡張リンクのロールを取得します。

getArcrole()　エラーが検出された DRS の関係ネットワークを定義しているアークのアークロールを取得します。

getCyclesAllowed()　エラーが検出された DRS の関係ネットワークで許可されるサイクルの種類を取得します。

getErrorRoutes()　誤っていた経路を取得します。

6.2.4　com.hitachi.xbrl.taxonomy パッケージ一覧

● XBRLArcroleType インターフェース

getId()　<link:arcroleType> 要素の id 属性の値を取得します。

getArcroleURI()　カスタムアークロールの arcroleURI 属性の値を取得します。

getCyclesAllowed()　<link:arcroleType> 要素の cyclesAllowed 属性の値を取得します。

getUsedOn()　カスタムアークロールを使用できる要素のリストを取得します。

isUsedOn(java.lang.String namespaceURI, java.lang.String name)　指定した要素で，カスタムアークロールが使用できるかどうかを調べます。

● XBRLLabel インターフェース

isRemote()　ラベルがラベルリンクの外で定義されているかどうかを調べます。

getHref()　ラベルがラベルリンクの外で定義されている場合に，そのラベルの URI を取得します。

getLang()　ラベルの言語を取得します。

getRole()　ラベルのロールを取得します。

getTitle()　ラベルのタイトルを取得します。

getValue()　ラベルの内容を表す文字列を取得します。

getContainerNode()　ラベルの内容を保持するノードを取得します。

● XBRLLinkbase インターフェース

getExtendedLinksW3CDOMNode(int extendedLinkType)　<link:linkbase> 要素に定義される拡張リンクのうち，引数で指定した種別の拡張リンクに対応する org.w3c.dom.Node オブジェクトのリストを取得します。

● XBRLTaxonomyLinkbaseRef インターフェース

getHref()　<link:linkbaseRef> 要素の xlink:href 属性の値を取得します。

getRole()　<link:linkbaseRef> 要素の xlink:role 属性の値を取得します。

getArcrole()　<link:linkbaseRef> 要素の xlink:arcrole 属性の値を取得します。

getBase()　<link:linkbaseRef> 要素の xml:base 属性の値を取得します。

● XBRLPart インターフェース

getNamespaceURI()　パートが属する名前空間 URI を取得します。

getName()　パートの名前を取得します。

getValue()　パートの値を取得します。

● XBRLReference インターフェース

isRemote()　文献への参照がリファレンスリンクの外で定義されているかどうかを調

べます。

getHref()　文献への参照がリファレンスリンクの外で定義されている場合に，その文献への参照の URI を取得します。

getRole()　文献への参照のロールを取得します。

getTitle()　文献への参照のタイトルを取得します。

getParts()　文献への参照に含まれるパートのリストを取得します。

getParts(java.lang.String namespaceURI, java.lang.String name)　文献への参照に含まれるパートのうち，指定した名前を持つパートのリストを取得します。

● XBRLTaxonomyRoleRef インターフェース

getRoleURI()　<link:roleRef> 要素の roleURI 属性の値を取得します。

getHref()　<link:roleRef> 要素の xlink:href 属性の値を取得します。

getRole()　<link:roleRef> 要素の xlink:role 属性の値を取得します。

getArcrole()　<link:roleRef> 要素の xlink:arcrole 属性の値を取得します。

● XBRLRoleType インターフェース

getId()　<link:roleType> 要素の id 属性の値を取得します。

getRoleURI()　カスタムロールの roleURI 属性の値を取得します。

getUsedOn()　カスタムロールを使用できる要素のリストを取得します。

isUsedOn(java.lang.String namespaceURI, java.lang.String name)　指定した要素で，カスタムロールが使用できるかどうかを調べます。

● XBRLTaxonomyAttributeDeclaration インターフェース

getUse()　属性宣言の use 属性の値を取得します。

getType()　属性宣言の型の修飾名を取得します。

getForm()　属性宣言の form 属性の値を取得します。

● XBRLTaxonomyBody インターフェース

getTargetNamespaceURI()　タクソノミー本体の対象名前空間 URI を取得します。

getElementFormDefault()　タクソノミー本体の elementFormDefault 属性の値を取得します。

getAttributeFormDefault()　タクソノミー本体の attributeFormDefault 属性の値を取得します。

getImportBodies()　タクソノミー本体で <xsd:import> 要素によりインポートするタクソノミー本体を取得します。

getIncludeBodies()　タクソノミー本体で <xsd:include> 要素によりインクルードするタクソノミー本体を取得します。

● XBRLTaxonomyComplexTypeDefinition インターフェース

getAbstract()　型定義の abstract 属性の値を取得します。

getBlock()　型定義の block 属性の値を表すコードを取得します。

getCompositor()　型定義のモデルグループを表すコードを取得します。

getElements()　型定義の要素宣言のリストを取得します。

getTaxonomyAttributeDeclarations()　型定義の属性宣言のリストを取得します。

getTaxonomyAttributeDeclaration(java.lang.String namespaceURI, java.lang.String name)　指定された名前を持つ属性宣言を取得します。

getElementDeclarationsW3CDOMNode()　型定義の要素宣言に対応する org.w3c.dom.Node オブジェクトのリストを取得します。

getDerivationMethod()　型定義の派生方法を表すコードを取得します。

● XBRLTaxonomyComponent インターフェース

getComponentType()　XML 文書の種類を表すコードを取得します。

getAbsoluteURI()　XML 文書の絶対 URI を取得します。

getW3CDOMDocument()　XML 文書の DOM ツリーを取得します。

● XBRLTaxonomyDocument インターフェース

getOwnerDocumentManager()　タクソノミー文書を所有する XBRLDocumentManager オブジェクトを取得します。

getXBRLVersion()　タクソノミー文書が準拠する XBRL バージョンを取得します。

getPTVLinkbase()　タクソノミー文書の解析時に生成された PTVLI を取得します。

getTaxonomyComponents()　タクソノミー文書を構成する XML 文書のリストを取得します。

getTaxonomyBodies()　タクソノミー文書を構成するタクソノミー本体のリストを取得します。

getTaxonomyBody(java.lang.String targetNamespaceURI)　タクソノミー文書を構成するタクソノミー本体のうち，指定した名前空間 URI を持つタクソノミー本体を取得

します。

getLinkbases()　タクソノミー文書を構成するリンクベースのリストを取得します。

getStartingPoints()　タクソノミー文書を構成する際の出発点となったタクソノミー本体およびリンクベースのリストを取得します。

getElements()　タクソノミー文書に記述されている要素の要素宣言のリストを取得します。

getElement(java.lang.String namespaceURI, java.lang.String name)　指定した名前の要素の要素宣言を取得します。

getRoleTypes()　タクソノミー文書に記述されている <link:roleType> 要素のリストを取得します。

getRoleTypes(java.lang.String roleURI)　タクソノミー文書に記述されている <link:roleType> 要素のうち，指定した値のカスタムロールを定義している <link:roleType> 要素のリストを取得します。

getArcroleTypes()　タクソノミー文書に記述されている <link:arcroleType> 要素のリストを取得します。

getArcroleTypes(java.lang.String arcroleURI)　タクソノミー文書に記述されている <link:arcroleType> 要素のうち，指定した値のカスタムアークロールを定義している <link:arcroleType> 要素のリストを取得します。

getLabels()　タクソノミー文書に記述されているラベルのリストを取得します。

getLabelsByLang(java.lang.String lang)　タクソノミー文書に記述されているラベルのうち，指定した言語のラベルのリストを取得します。

getLabelsByRole(java.lang.String role)　タクソノミー文書に記述されているラベルのうち，指定したロールを持つラベルのリストを取得します。

getLabels(java.lang.String lang, java.lang.String role)　タクソノミー文書に記述されているラベルのうち，指定した言語およびロールを持つラベルのリストを取得します。

getReferences()　タクソノミー文書に記述されている文献への参照のリストを取得します。

getReferences(java.lang.String role)　タクソノミー文書に記述されている文献への参照のうち，指定したロールを持つ文献への参照のリストを取得します。

findCalculationRootElements(java.lang.String extendedLinkRole, java.lang.String arcrole)　計算リンクで定義された関係のネットワークでルートとなる要素宣言を検索します。

findDefinitionRootElements(java.lang.String extendedLinkRole, java.lang.String

arcrole)　定義リンクで定義された関係のネットワークでルートとなる要素宣言を検索します。

findPresentationRootElements(java.lang.String extendedLinkRole, java.lang.String arcrole)　プレゼンテーションリンクで定義された関係のネットワークでルートとなる要素宣言を検索します。

getNamespaceDeclarations()　タクソノミー文書に記述されている名前空間宣言のリストを取得します。

getLinkbaseRefs()　タクソノミー文書に存在する <link:linkbaseRef> 要素のリストを取得します。

getRoleRefs()　タクソノミー文書に存在する <link:roleRef> 要素のリストを取得します。

getTaxonomyTypeDefinitions()　タクソノミー文書で定義された型定義のリストを取得します。

getNamespaceDeclarationsByPrefix(java.lang.String prefix)　タクソノミー文書で定義される名前空間宣言のうち，引数で指定した名前空間接頭辞の名前空間宣言のリストを取得します。

getNamespaceDeclarationsByURI(java.lang.String uri)　タクソノミー文書で定義される名前空間宣言のうち，引数で指定した名前空間 URI の名前空間宣言のリストを取得します。

getTaxonomyTypeDefinition(java.lang.String namespaceURI, java.lang.String name)　引数で指定した名前の型定義を取得します。

getDimensionalRelationshipSets()　タクソノミー文書で定義される DRS のリストを取得します。

getDimensionsVersion()　タクソノミー文書が準拠している XBRL Dimensions のバージョンを取得します。

● XBRLTaxonomyDocumentBuilder インターフェース

getProperty()　XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用するプロパティの値を取得します。

getOwnerDocumentManager()　XBRLTaxonomyDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

getXBRLVersion()　XBRLTaxonomyDocumentBuilder オブジェクトが解析するタクソノミー文書の XBRL バージョンを取得します。

isValidating()　XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時に文書の妥当性を検証するかどうかを調べます。

isDelayLoading()　XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時にリンクベースの遅延読み込みをするかどうかを調べます。

getMaxTaxonomyExtension()　タクソノミー文書の解析時に読み込むタクソノミー文書の最大階層数を取得します。

getMaxDocuments()　タクソノミー文書の解析時に読み込むタクソノミー文書の最大数を取得します。

setEntityResolver(org.xml.sax.EntityResolver entityResolver)　タクソノミー文書に記述されているエンティティを解決するために使用するエンティティリゾルバ（org.xml.sax.EntityResolver オブジェクト）を設定します。

setErrorHandler(XBRLParseErrorHandler errorHandler)　タクソノミー文書解析時のエラーを受け取る XBRLParseErrorHandler オブジェクトを設定します。

parse(org.xml.sax.InputSource[] is)　指定された入力ソースからタクソノミー文書を解析し，XBRLTaxonomyDocument オブジェクトを新規に生成します。

parse(java.io.File[] file)　指定されたファイルをタクソノミー文書として解析し，XBRLTaxonomyDocument オブジェクトを新規に生成します。

parse(java.lang.String[] uri)　指定された URI の内容をタクソノミー文書として解析し，XBRLTaxonomyDocument オブジェクトを新規に生成します。

getDimensionsVersion()　XBRLTaxonomyDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているタクソノミー文書を解析するのかを取得します。

● XBRLTaxonomyDocumentBuilderFactory インターフェース

getOwnerDocumentManager()　XBRLTaxonomyDocumentBuilderFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

setXBRLVersion(int version)　どの XBRL バージョンのタクソノミー文書を解析するのかを設定します。

getXBRLVersion()　どの XBRL バージョンのタクソノミー文書を解析するのかを取得します。

setValidating(boolean flag)　タクソノミー文書の解析時に文書の妥当性を検証するかどうかを設定します。

isValidating()　タクソノミー文書の解析時に文書の妥当性を検証するかどうかを調べます。

setDelayLoading(boolean flag)　タクソノミー文書の解析時にリンクベースの遅延読み込みをするかどうかを設定します。

isDelayLoading()　タクソノミー文書の解析時にリンクベースの遅延読み込みをするかどうかを調べます。

setMaxTaxonomyExtension(int max)　タクソノミー文書の解析時に読み込むタクソノミー文書の最大階層数を設定します。

getMaxTaxonomyExtension()　タクソノミー文書の解析時に読み込むタクソノミー文書の最大階層数を取得します。

setMaxDocuments(int max)　タクソノミー文書の解析時に読み込むタクソノミー文書の最大数を設定します。

getMaxDocuments()　タクソノミー文書の解析時に読み込むタクソノミー文書の最大数を取得します。

setIgnoringComments(boolean flag)　タクソノミー文書の解析時にコメントを無視するかどうかを設定します。

isIgnoringComments()　タクソノミー文書の解析時にコメントを無視するかどうかを調べます。

setCoalescing(boolean flag)　タクソノミー文書の解析時に CDATA ノードを Text ノードに変換するかどうかを設定します。

isCoalescing()　タクソノミー文書の解析時に CDATA ノードを Text ノードに変換するかどうかを調べます。

setProperty(java.lang.String name, java.lang.Object value)
XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用するプロパティの値を設定します。

getProperty(java.lang.String name)　XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用するプロパティの値を取得します。

newTaxonomyDocumentBuilder()　現在設定されているパラメタを使用して，XBRLTaxonomyDocumentBuilder のインスタンスを新規に生成します。

setDimensionsVersion(int version)　XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているタクソノミー文書を解析するのかを指定します。

getDimensionsVersion()　XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているタクソノミー文書を解析するのかを取得します。

● XBRLTaxonomyElement インターフェース

getElementType()　オブジェクトの型を表すコードを取得します。

getNamespaceURI()　要素が属する名前空間 URI を取得します。

getName()　要素の名前を取得します。

getType()　要素の型を示す修飾名を取得します。

getSubstitutionGroup()　要素のヘッド要素の修飾名を取得します。

isAbstract()　要素が抽象的かどうかを調べます。

isNillable()　要素の要素宣言の nillable 属性の値を取得します。

getId()　要素の id 属性の値を取得します。

getBalance()　要素宣言の xbrli:balance 属性を取得します。

getLabels()　要素に関連づけられているラベルのリストを取得します。

getLabelsByLang(java.lang.String lang)　要素に関連づけられているラベルのうち，指定した言語のラベルのリストを取得します。

getLabelsByRole(java.lang.String role)　要素に関連づけられているラベルのうち，指定したロールを持つラベルのリストを取得します。

getLabel(java.lang.String lang, java.lang.String role)　要素に関連づけられているラベルのうち，指定した言語とロールを持つラベルを取得します。

getLabels(java.lang.String lang, java.lang.String role)　要素に関連づけられているラベルのうち，指定した言語とロールを持つラベルのリストを取得します。

getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole)　要素の要素宣言とラベルの関係を表す XBRLLabelRelationship オブジェクトのリストを取得します。

getReferences()　要素に関連づけられている文献への参照のリストを取得します。

getReferences(java.lang.String role)　要素に関連づけられている文献への参照のうち，指定したロールを持つ文献への参照のリストを取得します。

getReferenceRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole)　要素の要素宣言と文献への参照の関係を表す XBRLReferenceRelationship オブジェクトのリストを取得します。

getCalculationRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole)　要素宣言とほかの要素宣言との関係を表す XBRLCalculationRelationship オブジェクトのリストを取得します。

getDefinitionRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole)　要素宣言とほかの要素宣言の関係を表す XBRLDefinitionRelationship オブ

ジェクトのリストを取得します。

getPresentationRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole)　要素宣言とほかの要素宣言の関係を表す XBRLPresentationRelationship オブジェクトのリストを取得します。

getTaxonomyTypeDefinition()　要素の型定義を表す XBRLTaxonomyTypeDefinition オブジェクトを取得します。

● XBRLTaxonomyItem インターフェース

getItemType()　アイテムのデータ型を表すコードを取得します。

getBaseItemType()　アイテムのベースデータ型を表すコードを取得します。

isNumeric()　アイテムが数値または非数値のどちらであるかを調べます。

getPeriodType()　要素宣言の xbrli:periodType 属性の値を取得します。

getDimensionalRelationshipSet()　アイテムに関連づいている DRS を取得します。

● XBRLTaxonomyNode インターフェース

getNodeType()　ノードの型を表すコードを取得します。

getOwnerTaxonomyDocument()　ノードを所有する XBRLTaxonomyDocument オブジェクトを取得します。

getParentNode()　ノードが関連づけられている親ノードを取得します。

getW3CDOMNode()　ノードに対応する org.w3c.dom.Node オブジェクトを取得します。

● XBRLTaxonomyOtherElement インターフェース

● XBRLTaxonomySimpleTypeDefinition インターフェース

● XBRLTaxonomyTuple インターフェース

● XBRLTaxonomyTypeDefinition インターフェース

getTypeCategory()　オブジェクトの型を表すコードを取得します。

getId()　型定義の id 属性の値を取得します。

getNamespaceURI()　型定義が属する名前空間 URI を取得します。

getName()　型定義の name 属性の値を取得します。

getFinal()　型定義の final 属性の値を表すコードを取得します。

getBase()　型定義の base 属性で示される型定義の修飾名を取得します。

● XBRLTaxonomyDimensionItem インターフェース

getTypedDomainRef()　要素宣言の xbrldt:typedDomainRef 属性の値を取得します。

getTypedDomainRefElement()　要素宣言の xbrldt:typedDomainRef 属性の値が示す要素宣言を取得します。

● XBRLTaxonomyHypercubeItem インターフェース

6.2.5 com.hitachi.xbrl.types パッケージ一覧

● XBRLAbstractBinary クラス

length()　XBRLAbstractBinary オブジェクトが保持するバイト配列の長さを取得します。

byteAt(int index)　指定したインデックスの位置にあるバイトを取得します。

toByteArray()　XBRLAbstractBinary オブジェクトが保持するバイト配列のコピーを取得します。

toString()　XBRLAbstractBinary オブジェクトが保持するバイト配列を文字列に変換します。

equals(java.lang.Object obj)　XBRLAbstractBinary オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLAbstractBinary オブジェクトのハッシュコードを取得します。

● XBRLAbstractPrecision クラス

isInfinite()　XBRLAbstractPrecision オブジェクトが文字列 "INF" を保持するかどうかを調べます。

getValue()　XBRLAbstractPrecision オブジェクトが保持する数値を取得します。

intValue()　XBRLAbstractPrecision オブジェクトが保持する値を int 型として取得します。

longValue()　XBRLAbstractPrecision オブジェクトが保持する値を long 型として取得します。

floatValue()　XBRLAbstractPrecision オブジェクトが保持する値を float 型として取得します。

doubleValue()　XBRLAbstractPrecision オブジェクトが保持する値を double 型として取得します。

toString()　XBRLAbstractPrecision オブジェクトの文字列表現を取得します。

equals(java.lang.Object obj)　XBRLAbstractPrecision オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLAbstractPrecision オブジェクトのハッシュコードを取得します。

● XBRLBase64Binary クラス

XBRLBase64Binary(java.lang.String base64)　指定した文字列から XBRLBase64Binary オブジェクトを構築します。

XBRLBase64Binary(byte[] bytes)　指定したバイト配列から XBRLBase64Binary オブジェクトを構築します。

XBRLBase64Binary(byte[] bytes, int offset, int length)　指定したバイト配列の一部から XBRLBase64Binary オブジェクトを構築します。

toByteArray(java.lang.String base64)　指定した文字列を Base64 でデコードして，バイト配列を生成します。

toString()　XBRLBase64Binary オブジェクトが保持するバイト配列を，Base64 でエンコードした結果を取得します。

toString(byte[] bytes)　指定したバイト配列を Base64 でエンコードした結果を取得します。

toString(byte[] bytes, int offset, int length)　指定したバイト配列の一部を Base64 でエンコードした結果を取得します。

valueOf(java.lang.String base64)　指定した文字列から XBRLBase64Binary オブジェクトを取得します。

valueOf(byte[] bytes)　指定したバイト配列から XBRLBase64Binary オブジェクトを取得します。

valueOf(byte[] bytes, int offset, int length)　指定したバイト配列の一部から XBRLBase64Binary オブジェクトを取得します。

● XBRLDate クラス

XBRLDate(java.lang.String date)　指定した date 型データの文字列表現から XBRLDate オブジェクトを構築します。

XBRLDate(XBRLDate other)　指定した XBRLDate オブジェクトから XBRLDate オブジェクトを新規に構築します。

XBRLDate(XBRLDateTime other)　指定した XBRLDate オブジェクトから，XBRLDate オブジェクトを新規に構築します。

compareTo(XBRLDate other)　XBRLDate オブジェクトとほかの XBRLDate オブ

ジェクトを比較します。

toString()　date 型データの文字列表現を取得します。

equals(java.lang.Object obj)　XBRLDate オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLDate オブジェクトのハッシュコードを取得します。

add(XBRLDuration duration)　指定した XBRLDuration オブジェクトを加算した XBRLDate オブジェクトを新規に取得します。

● XBRLDateTime クラス

XBRLDateTime(java.lang.String datetime)　指定した dateTime 型データの文字列表現から XBRLDateTime オブジェクトを構築します。

XBRLDateTime(XBRLDateTime other)　指定した XBRLDateTime オブジェクトから XBRLDateTime オブジェクトを新規に構築します。

XBRLDateTime(XBRLDate other)　指定した XBRLDateTime オブジェクトから，時間が "00:00:00.000" で始まる XBRLDateTime オブジェクトを新規に構築します。

compareTo(XBRLDateTime other)　XBRLDateTime オブジェクトとほかの XBRLDateTime オブジェクトを比較します。

toString()　dateTime 型データの文字列表現を取得します。

equals(java.lang.Object obj)　XBRLDateTime オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLDateTime オブジェクトのハッシュコードを取得します。

add(XBRLDuration duration)　指定した XBRLDuration オブジェクトを加算した XBRLDateTime オブジェクトを新規に取得します。

● XBRLDateUnion クラス

XBRLDateUnion(java.lang.String dateunion)　指定した dateUnion 型データの文字列表現から XBRLDateUnion オブジェクトを構築します。

XBRLDateUnion(XBRLDate date)　指定した XBRLDate オブジェクトから XBRLDateUnion オブジェクトを構築します。

XBRLDateUnion(XBRLDateTime datetime)　指定した XBRLDateTime オブジェクトから XBRLDateUnion オブジェクトを構築します。

getMemberType()　dateUnion 型データのメンバの型を調べます。

getMember()　dateUnion 型データのメンバを取得します。

compareTo(XBRLDateUnion other)　XBRLDateUnion オブジェクトとほかの XBRLDateUnion オブジェクトを比較します。

toString()　dateUnion 型データの文字列表現を取得します。

equals(java.lang.Object obj)　XBRLDateUnion オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLDateUnion オブジェクトのハッシュコードを取得します。

● XBRLDecimals クラス

XBRLDecimals(java.math.BigInteger value)　指定した数値から XBRLDecimals オブジェクトを構築します。

XBRLDecimals(java.lang.String decimals)　指定した decimals 属性の値の文字列表現から XBRLDecimals オブジェクトを構築します。

equals(java.lang.Object obj)　XBRLDecimals オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLDecimals オブジェクトのハッシュコードを取得します。

● XBRLDuration クラス

XBRLDuration(java.lang.String duration)　指定した duration 型データの文字列表現から XBRLDuration オブジェクトを構築します。

toString()　duration 型データの文字列表現を取得します。

equals(java.lang.Object obj)　XBRLDuration オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLDuration オブジェクトのハッシュコードを取得します。

● XBRLFraction クラス

XBRLFraction(java.math.BigDecimal numerator, java.math.BigDecimal denominator)　指定した分子要素と分母要素で構成される XBRLFraction オブジェクトを構築します。

XBRLFraction(java.lang.String fraction)　指定した分数の文字列表現から XBRLFraction オブジェクトを構築します。

getNumerator()　XBRLFraction オブジェクトの分子要素を取得します。

getDenominator()　XBRLFraction オブジェクトの分母要素を取得します。

toString()　XBRLFraction オブジェクトの文字列表現を取得します。

equals(java.lang.Object obj)　XBRLFraction オブジェクトとほかのオブジェクトが等

しいかどうかを調べます。

hashCode()　XBRLFraction オブジェクトのハッシュコードを取得します。

● XBRLGDay クラス

XBRLGDay(java.lang.String gday)　指定した gDay 型データの文字列表現から XBRLGDay オブジェクトを構築します。

toString()　gDay 型データの文字列表現を取得します。

equals(java.lang.Object obj)　XBRLGDay オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLGDay オブジェクトのハッシュコードを取得します。

● XBRLGMonth クラス

XBRLGMonth(java.lang.String gmonth)　指定した gMonth 型データの文字列表現から XBRLGMonth オブジェクトを構築します。

toString()　gMonth 型データの文字列表現を取得します。

equals(java.lang.Object obj)　XBRLGMonth オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLGMonth オブジェクトのハッシュコードを取得します。

● XBRLGMonthDay クラス

XBRLGMonthDay(java.lang.String gmonthday)　指定した gMonthDay 型データの文字列表現から XBRLGMonthDay オブジェクトを構築します。

toString()　gMonthDay 型データの文字列表現を取得します。

equals(java.lang.Object obj)　XBRLGMonthDay オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLGMonthDay オブジェクトのハッシュコードを取得します。

● XBRLGYear クラス

XBRLGYear(java.lang.String gyear)　指定した gYear 型データの文字列表現から XBRLGYear オブジェクトを構築します。

toString()　gYear 型データの文字列表現を取得します。

equals(java.lang.Object obj)　XBRLGYear オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLGYear オブジェクトのハッシュコードを取得します。

● XBRLGYearMonth クラス

XBRLGYearMonth(java.lang.String gyearmonth)　指定した gYearMonth 型データの文字列表現から XBRLGYearMonth オブジェクトを構築します。

toString()　gYearMonth 型データの文字列表現を取得します。

equals(java.lang.Object obj)　XBRLGYearMonth オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLGYearMonth オブジェクトのハッシュコードを取得します。

● XBRLHexBinary クラス

XBRLHexBinary(java.lang.String hex)　指定した 16 進数の文字列から XBRLHexBinary オブジェクトを構築します。

XBRLHexBinary(byte[] bytes)　指定したバイト配列から XBRLHexBinary オブジェクトを構築します。

XBRLHexBinary(byte[] bytes, int offset, int length)　指定したバイト配列の一部から XBRLHexBinary オブジェクトを構築します。

toByteArray(java.lang.String hex)　指定した 16 進数の文字列からバイト配列を生成します。

toString()　XBRLHexBinary オブジェクトが保持するバイト配列の 16 進数での表記を取得します。

toString(byte[] bytes)　指定したバイト配列の 16 進数での表記を取得します。

toString(byte[] bytes, int offset, int length)　指定したバイト配列の一部の 16 進数での表記を取得します。

valueOf(java.lang.String hex)　指定した 16 進数の文字列から XBRLHexBinary オブジェクトを取得します。

valueOf(byte[] bytes)　指定したバイト配列から XBRLHexBinary オブジェクトを取得します。

valueOf(byte[] bytes, int offset, int length)　指定したバイト配列の一部から XBRLHexBinary オブジェクトを取得します。

● XBRLPrecision クラス

XBRLPrecision(java.math.BigInteger value)　指定した数値から XBRLPrecision オブジェクトを構築します。

XBRLPrecision(java.lang.String precision)　指定した precision 属性の値の文字列表現から XBRLPrecision オブジェクトを構築します。

equals(java.lang.Object obj)　XBRLPrecision オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLPrecision オブジェクトのハッシュコードを取得します。

● XBRLQName クラス

XBRLQName(java.lang.String qname)　指定した QName の文字列表現から XBRLQName オブジェクトを構築します。

XBRLQName(java.lang.String namespaceURI, java.lang.String localPart)　指定した名前空間 URI およびローカルパートで構成される QName を表す XBRLQName オブジェクトを構築します。

XBRLQName(java.lang.String qname, boolean validate)　指定した QName の文字列表現から XBRLQName オブジェクトを構築します。

XBRLQName(java.lang.String namespaceURI, java.lang.String localPart, boolean validate)　指定した名前空間 URI およびローカルパートで構成される QName を表す XBRLQName オブジェクトを構築します。

getNamespaceURI()　QName の名前空間 URI を取得します。

getLocalPart()　QName のローカルパートを取得します。

toString()　QName の文字列表現を取得します。

toString(java.lang.String namespaceURI, java.lang.String localPart)　指定した名前空間 URI およびローカルパートで構成される，QName の文字列表現を取得します。

valueOf(java.lang.String qname)　指定した QName の文字列表現から XBRLQName オブジェクトを取得します。

valueOf(java.lang.String namespaceURI, java.lang.String localPart)　指定した名前空間 URI とローカルパートで構成される QName を表す XBRLQName オブジェクトを取得します。

equals(java.lang.Object obj)　XBRLQName オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLQName オブジェクトのハッシュコードを取得します。

● XBRLTime クラス

XBRLTime(java.lang.String time)　指定した time 型データの文字列から XBRLTime オブジェクトを構築します。

compareTo(XBRLTime other)　XBRLTime オブジェクトとほかの XBRLTime オブジェクトを比較します。

toString()　time 型データの文字列表現を取得します。

equals(java.lang.Object obj)　XBRLTime オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

hashCode()　XBRLTime オブジェクトのハッシュコードを取得します。

6.2.6 com.hitachi.xbrl.util パッケージ一覧

● XBRLIdCreator インターフェース

createId(int type, XBRLIdManager idManager)　ID を生成します。

● XBRLDocumentWriter インターフェース

getOwnerDocumentManager()　XBRLDocumentWriter オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

setEncoding(java.lang.String encoding)　XML 文書を出力する際に使用するエンコーディングを設定します。

getEncoding()　XML 文書を出力する際に使用するエンコーディングを取得します。

setIndent(boolean indent)　XML 文書を出力する際にインデントを設定するかどうかを指定します。

isIndent()　XML 文書を出力する際にインデントをするかどうかを調べます。

write(java.io.OutputStream os, org.w3c.dom.Document doc)　XML 文書をストリームへ出力します。

write(java.io.OutputStream os, XBRLInstanceDocument doc)　インスタンス文書をストリームへ出力します。

● XBRLDocumentWriterFactory インターフェース

getOwnerDocumentManager()　XBRLDocumentWriterFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

newDocumentWriter()　現在設定されているパラメタを使用して，XBRLDocumentWriter の新しいインスタンスを生成します。

● XBRLConstants クラス

● XBRLIdManager クラス

XBRLIdManager()　XBRL 文書に記述された ID を管理する XBRLIdManager オブジェクトを新規に構築します。

setIdCreator(XBRLIdCreator creator)　XBRLIdCreator インターフェースを継承し

たオブジェクトを，XBRLIdManager オブジェクトに設定します。

getIdCreator()　この XBRLIdManager オブジェクトに設定されている XBRLIdCreator インターフェースを継承したオブジェクトを取得します。

registerDocument(XBRLInstanceDocument doc)　XBRLIdManager オブジェクトで管理する XBRLInstanceDocument オブジェクトを登録します。

unregisterDocument(XBRLInstanceDocument doc)　XBRLIdManager オブジェクトで管理する XBRLInstanceDocument オブジェクトの登録を解除します。

isId(java.lang.String id)　この XBRLIdManager オブジェクトに登録されている XBRLInstanceDocument オブジェクトで，指定した ID がすでに使用されている ID かどうかを調べます。

getIds()　この XBRLIdManager オブジェクトに登録されている XBRLInstanceDocument オブジェクトで，すでに使用されている ID の配列を取得します。

● XBRLNamespaceDeclaration クラス

XBRLNamespaceDeclaration(java.lang.String prefix, java.lang.String uri)　指定した名前空間接頭辞と名前空間 URI から XBRLNamespaceDeclaration オブジェクトを構築します。

XBRLNamespaceDeclaration(java.lang.String prefix, java.lang.String uri, boolean validate)　指定した名前空間接頭辞と名前空間 URI から XBRLNamespaceDeclaration オブジェクトを構築します。

getPrefix()　名前空間接頭辞を取得します。

getURI()　名前空間 URI を取得します。

● XBRLSchemaLocation クラス

XBRLSchemaLocation(java.lang.String namespaceURI, java.lang.String schemaLocation)　XBRLSchemaLocation オブジェクトを構築します。

XBRLSchemaLocation(java.lang.String namespaceURI, java.lang.String schemaLocation, boolean validate)　指定した名前空間 URI とスキーマ文書の位置を示す URI から，XBRLSchemaLocation オブジェクトを構築します。

getNamespaceURI()　名前空間 URI を取得します。

getSchemaLocation()　スキーマ文書の位置を示す URI を取得します。

● PTVInstance クラス

getW3CDOMDocument()　PTVI の DOM ツリーを取得します。

getInferredItems()　PTVI に含まれるアイテムのうち，完全な妥当性検証によって推測されたアイテムのリストを取得します。

● PTVLinkbase クラス

getW3CDOMDocument()　PTVLI の DOM ツリーを取得します。

● XBRLEqualUtil クラス

XBRLEqualUtil()　XBRLEqualUtil オブジェクトを構築します。

XBRLEqualUtil(boolean ignoringZeroPrecision)　XBRLEqualUtil オブジェクトを構築します。

s_equalContext(XBRLContext2 context1, XBRLContext2 context2)　コンテキストが s-equal かどうかを判定します。

s_equalItem(XBRLInstanceItem item1, XBRLInstanceItem item2)　アイテムが s-equal かどうかを判定します。

s_equalTuple(XBRLInstanceTuple tuple1, XBRLInstanceTuple tuple2)　タプルが s-equal かどうかを判定します。

s_equalUnit(XBRLUnit2 unit1, XBRLUnit2 unit2)　単位が s-equal かどうかを判定します。

duplicateItem(XBRLInstanceItem item1, XBRLInstanceItem item2)　アイテムが duplicate かどうかを判定します。

duplicateTuple(XBRLInstanceTuple tuple1, XBRLInstanceTuple tuple2)　タプルが duplicate かどうかを判定します。

setIgnoringZeroPrecision(boolean ignoringZeroPrecision)　XBRLEqualUtil オブジェクトに設定されている精度 0 のアイテムの扱いを設定します。

isIgnoringZeroPrecision()　XBRLEqualUtil オブジェクトに設定されている精度 0 のアイテムの扱いを取得します。

d_equalItem(XBRLInstanceItem item1, XBRLInstanceItem item2, XBRLDimensionMember member)　引数に指定したアイテム同士が s-equal で，かつ，引数に指定したディメンジョンのメンバが両方のアイテムのコンテキストに存在するかどうかを判定します。

6.2.7　com.hitachi.xbrl.simple.taxonomy パッケージ一覧

● SimpleConceptDefinition インターフェース

● SimpleEdge インターフェース

getRole()　一つの関係が属する拡張リンク型要素の xlink:role 属性値を取得します。

getArcrole()　一つの関係を構成するアーク型要素の xlink:arcrole 属性値を取得します。

getUse()　一つの関係を構成するアーク型要素の use 属性値を取得します。

getFrom()　一つの関係を構成するアーク型要素の xlink:from 属性値で参照されるオブジェクトを取得します。

getTo()　一つの関係を構成するアーク型要素の xlink:to 属性値で参照されるオブジェクトを取得します。

getOrder()　一つの関係を構成するアーク型要素の order 属性値を取得します。

setOrder(java.math.BigDecimal order)　一つの関係を構成するアーク型要素の order 属性値を設定します。

● SimpleElementDefinition インターフェース

getName()　要素定義の name 属性値を取得します。

setName(java.lang.String name)　要素定義の name 属性値を設定します。

getType()　要素定義の type 属性値を取得します。

getId()　要素定義の id 属性値を取得します。

setId(java.lang.String id)　要素定義の id 属性値を設定します。

getAbstract()　要素定義の abstract 属性値を取得します。

setAbstract(boolean abstrakt)　要素定義の abstract 属性値を設定します。

getNillable()　要素定義の nillable 属性値を取得します。

setNillable(boolean nillable)　要素定義の nillable 属性値を設定します。

getSubstitutionGroup()　要素定義の substitutionGroup 属性値を取得します。

● SimpleItemDefinition インターフェース

getItemType()　アイテム要素定義の type 属性値に対応するフィールド値を取得します。

setItemType(int type)　アイテム要素定義の type 属性値を設定します。

getPeriodType()　アイテム要素定義の xbrli:periodType 属性値を取得します。

setPeriodType(java.lang.String periodType)　アイテム要素定義の xbrli:periodType 属性値を設定します。

getBalance()　アイテム要素定義の xbrli:balance 属性値を取得します。

setBalance(java.lang.String balance)　アイテム要素定義の xbrli:balance 属性値を設定します。

● SimpleLabel インターフェース

getRole()　ラベル要素（link:label）の xlink:role 属性値を取得します。

setRole(java.lang.String role)　ラベル要素（link:label）の xlink:role 属性値を設定します。

getLang()　ラベル要素（link:label）の xml:lang 属性値を取得します。

setLang(java.lang.String lang)　ラベル要素（link:label）の xml:lang 属性値を設定します。

getTitle()　ラベル要素（link:label）の xlink:title 属性値を取得します。

setTitle(java.lang.String title)　ラベル要素（link:label）の xlink:title 属性値を設定します。

getId()　ラベル要素（link:label）の id 属性値を取得します。

setId(java.lang.String id)　ラベル要素（link:label）の id 属性値を設定します。

getValue()　ラベル要素（link:label）の内容を取得します。

setValue(java.lang.String value)　ラベル要素（link:label）の内容を設定します。

● SimpleLabelEdge インターフェース

● SimpleNode インターフェース

isActive()　ノードがアクティブであるかどうかを調べます。

getNodeType()　ノードの型を表す値を取得します。

getOwnerTaxonomy()　ノードを所有する SimpleTaxonomy オブジェクトを取得します。

getParentNode()　ノードが関連づけられている親ノードを取得します。

● SimpleTaxonomy インターフェース

getOwnerDocumentManager()　単純なタクソノミー文書全体を表す SimpleTaxonomy オブジェクトを所有する，XBRLDocumentManager オブジェクトを取得します。

getBaseTaxonomyDocument()　単純なタクソノミー文書の拡張元であるタクソノミー文書を表す，XBRLTaxonomyDocument オブジェクトを取得します。

getTargetNamespace()　単純なタクソノミー文書を構成する，唯一のタクソノミー本体の targetNamespace 属性値を取得します。

getPrefix()　単純なタクソノミー文書を構成する唯一のタクソノミー本体の targetNamespace 属性値に対応する，名前空間接頭辞を取得します。

getSchemaLocation()　単純なタクソノミー文書を構成する唯一のタクソノミー本体が設定されている場所（ファイル名）を取得します。

createItemDefinition(java.lang.String name, int type, java.lang.String periodType)　単純なタクソノミー文書を構成するアイテム要素定義を表す，新たな SimpleItemDefinition オブジェクトを生成します。

findElementDefinitions(int type)　単純なタクソノミー文書を構成する要素定義を表す SimpleElementDefinition オブジェクトの配列を取得します。

getElementDefinition(java.lang.String name)　単純なタクソノミー文書を構成する要素定義を表す，SimpleElementDefinition オブジェクトを取得します。

addElementDefinition(SimpleElementDefinition obj)　単純なタクソノミー文書に，引数 obj に指定された要素定義を表す SimpleElementDefinition オブジェクトを追加して，アクティブ状態にします。

removeElementDefinition(SimpleElementDefinition obj)　単純なタクソノミー文書から，引数 obj に指定された要素定義を表す SimpleElementDefinition オブジェクトを削除し，非アクティブ状態にします。

createLabel(java.lang.String role, java.lang.String lang, java.lang.String value)　単純なタクソノミー文書を構成するラベル要素を表す，SimpleLabel オブジェクトを生成します。

createLabelEdge(java.lang.String role, java.lang.String arcrole, java.lang.Object from, java.lang.Object to)　単純なタクソノミー文書を構成するラベルリンクの一つの関係（一辺）を表す SimpleLabelEdge オブジェクトを生成します。

createProhibitedEdge(XBRLRelationship obj)　単純なタクソノミー文書を構成する既存の関係の一辺を否定する SimpleEdge オブジェクトを生成します。

findEdges(int type, java.lang.String role, java.lang.String arcrole, java.lang.Object from, java.lang.Object to)　単純なタクソノミー文書を構成するリンクベースの一つの関係（一辺）を表す SimpleEdge オブジェクトの配列をこの SimpleTaxonomy オブジェクトから取得します。

addEdge(SimpleEdge obj)　単純なタクソノミー文書を構成する関係の一辺を表す SimpleEdge オブジェクトを追加します。

removeEdge(SimpleEdge obj)　単純なタクソノミー文書を構成する関係の一辺を表す

SimpleEdge オブジェクトを削除します。

transform()　単純なタクソノミー文書の現在の構成を org.w3c.dom.Document オブジェクトへ変換した，新たな org.w3c.dom.Document オブジェクトの配列を取得します。

● SimpleTaxonomyBuilder インターフェース

getOwnerDocumentManager()　SimpleTaxonomyBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

newSimpleTaxonomy(java.lang.String targetNamespace, java.lang.String prefix, java.lang.String fileName, XBRLTaxonomyDocument doc)　単純なタクソノミー文書を表す SimpleTaxonomy オブジェクトを新規生成します。

● SimpleTaxonomyBuilderFactory インターフェース

getOwnerDocumentManager()　SimpleTaxonomyBuilderFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

newSimpleTaxonomyBuilder()　現在の設定で SimpleTaxonomyBuilder オブジェクトを生成します。

6.3 メソッド一覧（アルファベット順）

XBRL 文書操作用 API のメソッドを，アルファベット順に一覧で表示しています。一覧では，「パッケージ名」「インターフェース／クラス名」「メソッド名」の形式を併記しています。

［A］［B］［C］［D］［E］［F］［G］［H］［I］［L］［N］［P］［R］［S］［T］［U］［V］［W］［X］

A

add　com.hitachi.xbrl.types.XBRLDate クラスの add(XBRLDuration duration) メソッド

add　com.hitachi.xbrl.types.XBRLDateTime クラスの add(XBRLDuration duration) メソッド

addArcroleRef　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの addArcroleRef(XBRLArcroleRef newArcroleRef) メソッド

addContext　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの addContext(XBRLContext newContext) メソッド

addEdge　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの addEdge(SimpleEdge obj) メソッド

addElementDefinition　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの addElementDefinition(SimpleElementDefinition obj) メソッド

addFootnote　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの addFootnote(XBRLFootnote newFootnote) メソッド

addFootnote　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの addFootnote(XBRLFootnote newFootnote) メソッド

addFootnote　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの addFootnote(XBRLFootnote newFootnote, java.lang.String arcrole, java.math.BigDecimal order) メソッド

addLinkbaseRef　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの addLinkbaseRef(XBRLLinkbaseRef newLinkbaseRef) メソッド

addNamespaceDeclaration　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの addNamespaceDeclaration(XBRLNamespaceDeclaration newNamespaceDeclaration) メソッド

addRoleRef　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの

addRoleRef(XBRLRoleRef newRoleRef) メソッド

addSchemaLocation　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの addSchemaLocation(XBRLSchemaLocation newSchemaLocation) メソッド

addSchemaRef　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの addSchemaRef(XBRLSchemaRef newSchemaRef) メソッド

addUnit　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの addUnit(XBRLUnit2 newUnit) メソッド

appendChildElement　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの appendChildElement(XBRLInstanceElement newElement) メソッド

appendChildElement　com.hitachi.xbrl.instance.XBRLInstanceTuple インターフェースの appendChildElement(XBRLInstanceElement newElement) メソッド

appendDimensionMember　com.hitachi.xbrl.instance.XBRLScenario インターフェースの appendDimensionMember(XBRLDimensionMember newMember) メソッド

appendDimensionMember　com.hitachi.xbrl.instance.XBRLSegment インターフェースの appendDimensionMember(XBRLDimensionMember newMember) メソッド

B

byteAt　com.hitachi.xbrl.types.XBRLAbstractBinary クラスの byteAt(int index) メソッド

C

clearCache　com.hitachi.xbrl.XBRLDocumentManager クラスの clearCache() メソッド

cloneNode　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの cloneNode(org.w3c.dom.Document w3cdomDocument, boolean deep) メソッド

cloneNode　com.hitachi.xbrl.instance.XBRLInstanceNode インターフェースの cloneNode(boolean deep) メソッド

compareTo　com.hitachi.xbrl.types.XBRLTime クラスの compareTo(XBRLTime other) メソッド

compareTo　com.hitachi.xbrl.types.XBRLDate クラスの compareTo(XBRLDate other) メソッド

compareTo　com.hitachi.xbrl.types.XBRLDateTime クラスの compareTo(XBRLDateTime other) メソッド

compareTo　com.hitachi.xbrl.types.XBRLDateUnion クラスの

compareTo(XBRLDateUnion other) メソッド

createArcroleRef　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createArcroleRef(java.lang.String href, java.lang.String arcroleURI, java.lang.String base) メソッド

createContext　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createContext(XBRLEntity entity, XBRLPeriod period, XBRLScenario scenario) メソッド

createEntity　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createEntity(XBRLIdentifier identifier, XBRLSegment segment) メソッド

createExplicitMember　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createExplicitMember(XBRLQName dimension, XBRLQName value) メソッド

createFootnote　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createFootnote(java.lang.String lang, java.lang.String role, java.lang.String title) メソッド

createFootnote　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createFootnote(java.lang.String footnoteLinkRole, java.lang.String lang, java.lang.String role, java.lang.String title) メソッド

createId　com.hitachi.xbrl.util.XBRLIdCreator インターフェースの createId(int type, XBRLIdManager idManager) メソッド

createIdentifier　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createIdentifier(java.lang.String namespaceURI, java.lang.String value) メソッド

createItem　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createItem(java.lang.String namespaceURI, java.lang.String name, XBRLContext context) メソッド

createItemDefinition　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの createItemDefinition(java.lang.String name, int type, java.lang.String periodType) メソッド

createLabel　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの createLabel(java.lang.String role, java.lang.String lang, java.lang.String value) メソッド

createLabelEdge　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの createLabelEdge(java.lang.String role, java.lang.String arcrole, java.lang.Object from, java.lang.Object to) メソッド

createLinkbaseRef　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createLinkbaseRef(java.lang.String role, java.lang.String href) メソッド

createLinkbaseRef　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createLinkbaseRef(java.lang.String role, java.lang.String href, java.lang.String base) メソッド

createMeasure　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createMeasure(XBRLQName value) メソッド

createNonNumericContext　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createNonNumericContext(XBRLEntity entity, XBRLPeriod period, XBRLUnit unit, XBRLScenario scenario) メソッド

createNonNumericItem　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createNonNumericItem(java.lang.String namespaceURI, java.lang.String name, XBRLContext2 context) メソッド

createNumericContext　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createNumericContext(XBRLEntity entity, XBRLPeriod period, XBRLUnit unit, XBRLScenario scenario, int precision, boolean cwa) メソッド

createNumericItem　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createNumericItem(java.lang.String namespaceURI, java.lang.String name, XBRLContext context, XBRLUnit2 unit, XBRLPrecision precision, XBRLDecimals decimals) メソッド

createOperator　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createOperator(java.lang.String kind, XBRLMeasureOrOperator lhs, XBRLMeasureOrOperator rhs) メソッド

createPeriod　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createPeriod(int periodType, XBRLDateUnion instant, XBRLDateUnion startDate, XBRLDateUnion endDate, XBRLDuration duration) メソッド

createPeriod　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createPeriod() メソッド

createPeriod　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createPeriod(XBRLDateUnion instant) メソッド

createPeriod　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createPeriod(XBRLDateUnion startDate, XBRLDateUnion endDate) メソッド

createProhibitedEdge　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの createProhibitedEdge(XBRLRelationship obj) メソッド

createRoleRef　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createRoleRef(java.lang.String href, java.lang.String roleURI, java.lang.String base) メソッド

createScenario　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createScenario(java.lang.String name) メソッド

createScenario　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createScenario() メソッド

createSchemaRef　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createSchemaRef(java.lang.String href, java.lang.String base) メソッド

createSegment　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createSegment(java.lang.String name) メソッド

createSegment　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createSegment() メソッド

createTuple　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createTuple(java.lang.String namespaceURI, java.lang.String name) メソッド

createTypedMember　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createTypedMember(XBRLQName dimension) メソッド

createUnit　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createUnit(XBRLQName[] measures) メソッド

createUnit　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createUnit(XBRLQName[] numerator, XBRLQName[] denominator) メソッド

createUnit　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの createUnit(XBRLMeasureOrOperator measureOrOperator) メソッド

D

d_equalItem　com.hitachi.xbrl.util.XBRLEqualUtil クラスの d_equalItem(XBRLInstanceItem item1, XBRLInstanceItem item2, XBRLDimensionMember member) メソッド

doubleValue　com.hitachi.xbrl.types.XBRLAbstractPrecision クラスの doubleValue() メソッド

duplicateItem　com.hitachi.xbrl.util.XBRLEqualUtil クラスの duplicateItem(XBRLInstanceItem item1, XBRLInstanceItem item2) メソッド

duplicateTuple　com.hitachi.xbrl.util.XBRLEqualUtil クラスの duplicateTuple(XBRLInstanceTuple tuple1, XBRLInstanceTuple tuple2) メソッド

E

equals　com.hitachi.xbrl.types.XBRLGDay クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLGMonth クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLGMonthDay クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLGYear クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLGYearMonth クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLPrecision クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLQName クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLTime クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLAbstractBinary クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLAbstractPrecision クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLDate クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLDateTime クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLDateUnion クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLDecimals クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLDuration クラスの equals(java.lang.Object obj) メソッド

equals　com.hitachi.xbrl.types.XBRLFraction クラスの equals(java.lang.Object obj) メソッド

error　com.hitachi.xbrl.XBRLParseErrorHandler インターフェースの error(XBRLDOMException exception) メソッド

F

fatalError　com.hitachi.xbrl.XBRLParseErrorHandler インターフェースの fatalError(XBRLDOMException exception) メソッド

findCalculationRootElements　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの findCalculationRootElements(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッド

findDefinitionRootElements　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの findDefinitionRootElements(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッド

findEdges　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの findEdges(int type, java.lang.String role, java.lang.String arcrole, java.lang.Object from, java.lang.Object to) メソッド

findElementDefinitions　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの findElementDefinitions(int type) メソッド

findPresentationRootElements
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの findPresentationRootElements(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッド

floatValue　com.hitachi.xbrl.types.XBRLAbstractPrecision クラスの floatValue() メソッド

G

getAbsoluteURI　com.hitachi.xbrl.taxonomy.XBRLTaxonomyComponent インターフェースの getAbsoluteURI() メソッド

getAbstract　com.hitachi.xbrl.simple.taxonomy.SimpleElementDefinition インターフェースの getAbstract() メソッド

getAbstract　com.hitachi.xbrl.taxonomy. XBRLTaxonomyComplexTypeDefinition インターフェースの getAbstract() メソッド

getArcrole　com.hitachi.xbrl.link.XBRLRelationship インターフェースの getArcrole() メソッド

getArcrole　com.hitachi.xbrl.link.XBRLIllegalCycleException クラスの getArcrole() メソッド

getArcrole　com.hitachi.xbrl.link.XBRLDimensionIllegalCycleException クラスの getArcrole() メソッド

getArcrole　com.hitachi.xbrl.taxonomy. XBRLTaxonomyLinkbaseRef インターフェースの getArcrole() メソッド

getArcrole　com.hitachi.xbrl.taxonomy. XBRLTaxonomyRoleRef インターフェースの getArcrole() メソッド

getArcrole　com.hitachi.xbrl.simple.taxonomy.SimpleEdge インターフェースの getArcrole() メソッド

getArcroleRef　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getArcroleRef(java.lang.String arcroleURI) メソッド

getArcroleRefs　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getArcroleRefs() メソッド

getArcroleType　com.hitachi.xbrl.instance.XBRLArcroleRef インターフェースの getArcroleType() メソッド

getArcroleTypes　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getArcroleTypes() メソッド

getArcroleTypes　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getArcroleTypes(java.lang.String arcroleURI) メソッド

getArcroleURI　com.hitachi.xbrl.instance.XBRLArcroleRef インターフェースの getArcroleURI() メソッド

getArcroleURI　com.hitachi.xbrl.taxonomy.XBRLArcroleType インターフェースの getArcroleURI() メソッド

getArcType　com.hitachi.xbrl.link.XBRLRelationship インターフェースの getArcType() メソッド

getAttributeFormDefault　com.hitachi.xbrl.taxonomy. XBRLTaxonomyBody インターフェースの getAttributeFormDefault() メソッド

getBalance　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getBalance() メソッド

getBalance　com.hitachi.xbrl.simple.taxonomy.SimpleItemDefinition インターフェースの getBalance() メソッド

getBase　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getBase() メソッド

getBase　com.hitachi.xbrl.instance.XBRLArcroleRef インターフェースの getBase() メソッド

getBase　com.hitachi.xbrl.instance.XBRLLinkbaseRef インターフェースの getBase() メソッド

getBase　com.hitachi.xbrl.instance.XBRLRoleRef インターフェースの getBase() メソッド

getBase　com.hitachi.xbrl.instance.XBRLSchemaRef インターフェースの getBase() メソッド

getBase　com.hitachi.xbrl.taxonomy. XBRLTaxonomyLinkbaseRef インターフェースの getBase() メソッド

getBase　com.hitachi.xbrl.taxonomy. XBRLTaxonomyTypeDefinition インターフェースの getBase() メソッド

getBaseItemType　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの getBaseItemType() メソッド

getBaseItemType　com.hitachi.xbrl.taxonomy.XBRLTaxonomyItem インターフェースの getBaseItemType() メソッド

getBaseTaxonomyDocument　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの getBaseTaxonomyDocument() メソッド

getBlock　com.hitachi.xbrl.taxonomy. XBRLTaxonomyComplexTypeDefinition インターフェースの getBlock() メソッド

getCalculationRelationships　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getCalculationRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッド

getChildElements　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getChildElements() メソッド

getChildElements　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getChildElements(java.lang.String namespaceURI, java.lang.String name) メソッド

getChildElements　com.hitachi.xbrl.instance.XBRLInstanceTuple インターフェースの getChildElements() メソッド

getChildElements　com.hitachi.xbrl.instance.XBRLInstanceTuple インターフェースの getChildElements(java.lang.String namespaceURI, java.lang.String name) メソッド

getClosed　com.hitachi.xbrl.link.XBRLHasHypercubeRelationship インターフェースの getClosed() メソッド

getCode　com.hitachi.xbrl.XBRLDOMException クラスの getCode() メソッド

getCode　com.hitachi.xbrl.XBRLDOMRuntimeException クラスの getCode() メソッド

getComponentType　com.hitachi.xbrl.taxonomy.XBRLTaxonomyComponent インターフェースの getComponentType() メソッド

getCompositor　com.hitachi.xbrl.taxonomy. XBRLTaxonomyComplexTypeDefinition インターフェースの getCompositor() メソッド

getContainerNode　com.hitachi.xbrl.instance.XBRLScenario インターフェースの getContainerNode() メソッド

getContainerNode　com.hitachi.xbrl.instance.XBRLSegment インターフェースの getContainerNode() メソッド

getContainerNode　com.hitachi.xbrl.instance.XBRLTypedMember インターフェースの getContainerNode() メソッド

getContainerNode　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの getContainerNode() メソッド

getContainerNode　com.hitachi.xbrl.taxonomy.XBRLLabel インターフェースの getContainerNode() メソッド

getContext　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの getContext() メソッド

getContextElementType　com.hitachi.xbrl.link.XBRLHasHypercubeRelationship インターフェースの getContextElementType() メソッド

getContextElementType　com.hitachi.xbrl.link.XBRLDRSDimension インターフェースの getContextElementType() メソッド

getContexts　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getContexts() メソッド

getCwa　com.hitachi.xbrl.instance.XBRLNumericContext インターフェースの getCwa() メソッド

getCyclesAllowed　com.hitachi.xbrl.link.XBRLIllegalCycleException クラスの getCyclesAllowed() メソッド

getCyclesAllowed　com.hitachi.xbrl.link.XBRLDimensionIllegalCycleException クラスの getCyclesAllowed() メソッド

getCyclesAllowed　com.hitachi.xbrl.taxonomy.XBRLArcroleType インターフェースの getCyclesAllowed() メソッド

getDecimals　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの getDecimals() メソッド

getDefinitionRelationships　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getDefinitionRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッド

getDefinitionRelationships　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getDefinitionRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッド

getDenominator　com.hitachi.xbrl.instance.XBRLUnit2 インターフェースの getDenominator() メソッド

getDenominator　com.hitachi.xbrl.types.XBRLFraction クラスの getDenominator() メソッド

getDerivationMethod　com.hitachi.xbrl.taxonomy.XBRLTaxonomyComplexTypeDefinition インターフェースの getDerivationMethod() メソッド

getDimension　com.hitachi.xbrl.instance.XBRLDimensionMember インターフェースの getDimension() メソッド

getDimensionalRelationshipSet　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの getDimensionalRelationshipSet() メソッド

getDimensionalRelationshipSet　com.hitachi.xbrl.taxonomy.XBRLTaxonomyItem インターフェースの getDimensionalRelationshipSet() メソッド

getDimensionalRelationshipSets
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getDimensionalRelationshipSets() メソッド

getDimensionDefaultRelationship
com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet インターフェースの getDimensionDefaultRelationship(XBRLHypercubeDimensionRelationship relationship) メソッド

getDimensionDomainRelationships
com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet インターフェースの getDimensionDomainRelationships(XBRLHypercubeDimensionRelationship relationship) メソッド

getDimensionMembers　com.hitachi.xbrl.instance.XBRLScenario インターフェースの getDimensionMembers() メソッド

getDimensionMembers　com.hitachi.xbrl.instance.XBRLSegment インターフェースの getDimensionMembers() メソッド

getDimensionsVersion　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getDimensionsVersion() メソッド

getDimensionsVersion　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの getDimensionsVersion() メソッド

getDimensionsVersion
com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの getDimensionsVersion() メソッド

getDimensionsVersion　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getDimensionsVersion() メソッド

getDimensionsVersion　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの getDimensionsVersion() メソッド

getDimensionsVersion
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの getDimensionsVersion() メソッド

getDomainMemberRelationships
com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet インターフェースの getDomainMemberRelationships(XBRLDimensionDomainRelationship relationship) メソッド

getDomainMemberRelationships
com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet インターフェースの getDomainMemberRelationships(XBRLDomainMemberRelationship relationship) メソッド

getDRSDefaultMember　com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet インターフェースの getDRSDefaultMember(XBRLDRSDimension dimension) メソッド

getDRSDimensions　com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet インターフェースの getDRSDimensions(java.lang.String role) メソッド

getDRSDimensions　com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet インターフェースの getDRSDimensions(java.lang.String role, int contextElementType) メソッド

getDRSMembers　com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet インター

フェースの getDRSMembers(XBRLDRSDimension dimension) メソッド

getDuration　com.hitachi.xbrl.instance.XBRLPeriod インターフェースの getDuration() メソッド

getElement　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getElement(java.lang.String namespaceURI, java.lang.String name) メソッド

getElementDeclarationsW3CDOMNode　com.hitachi.xbrl.taxonomy. XBRLTaxonomyComplexTypeDefinition インターフェースの getElementDeclarationsW3CDOMNode() メソッド

getElementDefinition　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの getElementDefinition(java.lang.String name) メソッド

getElementFormDefault　com.hitachi.xbrl.taxonomy. XBRLTaxonomyBody インターフェースの getElementFormDefault() メソッド

getElements　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getElements() メソッド

getElements　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getElements(java.lang.String namespaceURI, java.lang.String name) メソッド

getElements　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの getElements() メソッド

getElements　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの getElements(java.lang.String namespaceURI, java.lang.String name) メソッド

getElements　com.hitachi.xbrl.taxonomy. XBRLTaxonomyComplexTypeDefinition インターフェースの getElements() メソッド

getElements　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getElements() メソッド

getElementType　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getElementType() メソッド

getElementType　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getElementType() メソッド

getEncoding　com.hitachi.xbrl.instance.XBRLInstanceDocumentWriter インターフェースの getEncoding() メソッド

getEncoding　com.hitachi.xbrl.util.XBRLDocumentWriter インターフェースの getEncoding() メソッド

getEndDate　com.hitachi.xbrl.instance.XBRLPeriod インターフェースの getEndDate() メソッド

getEntity　com.hitachi.xbrl.instance.XBRLContext インターフェースの getEntity() メソッド

getErrorRoutes　com.hitachi.xbrl.link.XBRLIllegalCycleException クラスの getErrorRoutes() メソッド

getErrorRoutes　com.hitachi.xbrl.link.XBRLDimensionIllegalCycleException クラスの getErrorRoutes() メソッド

getExplicitMembers　com.hitachi.xbrl.instance.XBRLScenario インターフェースの getExplicitMembers() メソッド

getExplicitMembers　com.hitachi.xbrl.instance.XBRLSegment インターフェースの getExplicitMembers() メソッド

getExtendedLinkRole　com.hitachi.xbrl.link.XBRLRelationship インターフェースの getExtendedLinkRole() メソッド

getExtendedLinkRole　com.hitachi.xbrl.link.XBRLIllegalCycleException クラスの getExtendedLinkRole() メソッド

getExtendedLinkRole　com.hitachi.xbrl.link.XBRLDimensionIllegalCycleException クラスの getExtendedLinkRole() メソッド

getExtendedLinksW3CDOMNode　com.hitachi.xbrl.taxonomy. XBRLLinkbase インターフェースの getExtendedLinksW3CDOMNode(int extendedLinkType) メソッド

getExternalSchemaRefs　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの getExternalSchemaRefs() メソッド

getExternalSchemaRefs
com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの getExternalSchemaRefs() メソッド

getFinal　com.hitachi.xbrl.taxonomy. XBRLTaxonomyTypeDefinition インターフェースの getFinal() メソッド

getFootnoteLinkRole　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの getFootnoteLinkRole() メソッド

getFootnoteRelationships　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getFootnoteRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッド

getFootnotes　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの

getFootnotes() メソッド

getFootnotes　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getFootnotes(java.lang.String lang, java.lang.String role) メソッド

getFootnotes　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getFootnotes() メソッド

getFootnotes　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getFootnotes(java.lang.String lang, java.lang.String role) メソッド

getFootnotesByLang　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getFootnotesByLang(java.lang.String lang) メソッド

getFootnotesByLang　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getFootnotesByLang(java.lang.String lang) メソッド

getFootnotesByRole　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getFootnotesByRole(java.lang.String role) メソッド

getFootnotesByRole　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getFootnotesByRole(java.lang.String role) メソッド

getForm　com.hitachi.xbrl.taxonomy. XBRLTaxonomyAttributeDeclaration インターフェースの getForm() メソッド

getFrom　com.hitachi.xbrl.link.XBRLRelationship インターフェースの getFrom() メソッド

getFrom　com.hitachi.xbrl.simple.taxonomy.SimpleEdge インターフェースの getFrom() メソッド

getHasHypercubeRelationships
com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet インターフェースの getHasHypercubeRelationships(java.lang.String role) メソッド

getHasHypercubeRelationshipType
com.hitachi.xbrl.link.XBRLHasHypercubeRelationship インターフェースの getHasHypercubeRelationshipType() メソッド

getHref　com.hitachi.xbrl.instance.XBRLArcroleRef インターフェースの getHref() メソッド

getHref　com.hitachi.xbrl.instance.XBRLLinkbaseRef インターフェースの getHref() メソッド

getHref　com.hitachi.xbrl.instance.XBRLRoleRef インターフェースの getHref() メソッド

getHref　com.hitachi.xbrl.instance.XBRLSchemaRef インターフェースの getHref() メソッド

getHref　com.hitachi.xbrl.taxonomy.XBRLLabel インターフェースの getHref() メソッド

getHref　com.hitachi.xbrl.taxonomy. XBRLTaxonomyLinkbaseRef インターフェースの getHref() メソッド

getHref　com.hitachi.xbrl.taxonomy.XBRLReference インターフェースの getHref() メソッド

getHref　com.hitachi.xbrl.taxonomy. XBRLTaxonomyRoleRef インターフェースの getHref() メソッド

getHypercubeDimensionRelationships
com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet インターフェースの getHypercubeDimensionRelationships(XBRLHasHypercubeRelationship relationship) メソッド

getId　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getId() メソッド

getId　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getId() メソッド

getId　com.hitachi.xbrl.instance.XBRLContext インターフェースの getId() メソッド

getId　com.hitachi.xbrl.instance.XBRLUnit2 インターフェースの getId() メソッド

getId　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの getId() メソッド

getId　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getId() メソッド

getId　com.hitachi.xbrl.taxonomy.XBRLArcroleType インターフェースの getId() メソッド

getId　com.hitachi.xbrl.taxonomy.XBRLRoleType インターフェースの getId() メソッド

getId　com.hitachi.xbrl.taxonomy. XBRLTaxonomyTypeDefinition インターフェースの getId() メソッド

getId　com.hitachi.xbrl.simple.taxonomy.SimpleElementDefinition インターフェースの getId(java.lang.String id) メソッド

getId　com.hitachi.xbrl.simple.taxonomy.SimpleLabel インターフェースの getId() メソッド

getIdCreator　com.hitachi.xbrl.util.XBRLIdManager クラスの getIdCreator() メソッド

getIdentifier　com.hitachi.xbrl.instance.XBRLEntity インターフェースの getIdentifier() メソッド

getIdManager　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getIdManager() メソッド

getIds　com.hitachi.xbrl.util.XBRLIdManager クラスの getIds() メソッド

getImportBodies　com.hitachi.xbrl.taxonomy. XBRLTaxonomyBody インターフェースの getImportBodies() メソッド

getIncludeBodies　com.hitachi.xbrl.taxonomy. XBRLTaxonomyBody インターフェースの getIncludeBodies() メソッド

getInferredItems　com.hitachi.xbrl.util.PTVInstance クラスの getInferredItems() メソッド

getInstance　com.hitachi.xbrl.XBRLDocumentManager クラスの getInstance() メソッド

getInstanceRoot　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getInstanceRoot() メソッド

getInstant　com.hitachi.xbrl.instance.XBRLPeriod インターフェースの getInstant() メソッド

getItems　com.hitachi.xbrl.instance.XBRLContext インターフェースの getItems() メソッド

getItems　com.hitachi.xbrl.instance.XBRLContext インターフェースの getItems(java.lang.String namespaceURI, java.lang.String name) メソッド

getItems　com.hitachi.xbrl.instance.XBRLUnit2 インターフェースの getItems() メソッド

getItems　com.hitachi.xbrl.instance.XBRLUnit2 インターフェースの getItems(java.lang.String namespaceURI, java.lang.String name) メソッド

getItemType　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの getItemType() メソッド

getItemType　com.hitachi.xbrl.taxonomy.XBRLTaxonomyItem インターフェースの getItemType() メソッド

getItemType　com.hitachi.xbrl.simple.taxonomy.SimpleItemDefinition インターフェースの getItemType() メソッド

getKind　com.hitachi.xbrl.instance.XBRLOperator インターフェースの getKind() メソッド

getLabel　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getLabel(java.lang.String lang, java.lang.String role) メソッド

getLabel　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getLabel(java.lang.String lang, java.lang.String role) メソッド

getLabelRelationships　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッド

getLabelRelationships　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッド

getLabels　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getLabels(java.lang.String lang, java.lang.String role) メソッド

getLabels　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getLabels() メソッド

getLabels　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getLabels() メソッド

getLabels　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getLabels(java.lang.String lang, java.lang.String role) メソッド

getLabels　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getLabels() メソッド

getLabels　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getLabels(java.lang.String lang, java.lang.String role) メソッド

getLabelsByLang　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getLabelsByLang(java.lang.String lang) メソッド

getLabelsByLang　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getLabelsByLang(java.lang.String lang) メソッド

getLabelsByLang　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getLabelsByLang(java.lang.String lang) メソッド

getLabelsByRole　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getLabelsByRole(java.lang.String role) メソッド

getLabelsByRole　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インター

フェースの getLabelsByRole(java.lang.String role) メソッド

getLabelsByRole　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getLabelsByRole(java.lang.String role) メソッド

getLang　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの getLang() メソッド

getLang　com.hitachi.xbrl.taxonomy.XBRLLabel インターフェースの getLang() メソッド

getLang　com.hitachi.xbrl.simple.taxonomy.SimpleLabel インターフェースの getLang() メソッド

getLhs　com.hitachi.xbrl.instance.XBRLOperator インターフェースの getLhs() メソッド

getLinkbaseRefs　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getLinkbaseRefs() メソッド

getLinkbaseRefs　com.hitachi.xbrl.taxonomy. XBRLTaxonomyDocument インターフェース getLinkbaseRefs() メソッド

getLinkbaseRefs　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getLinkbaseRefs(java.lang.String role) メソッド

getLinkbases　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getLinkbases() メソッド

getLinkName　com.hitachi.xbrl.link.XBRLIllegalCycleException クラスの getLinkName() メソッド

getLinkName　com.hitachi.xbrl.link.XBRLDimensionIllegalCycleException クラスの getLinkName() メソッド

getLocalPart　com.hitachi.xbrl.types.XBRLQName クラスの getLocalPart() メソッド

getMaxDocuments　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの getMaxDocuments() メソッド

getMaxDocuments
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの getMaxDocuments() メソッド

getMaxTaxonomyExtension
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの getMaxTaxonomyExtension() メソッド

getMaxTaxonomyExtension
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの getMaxTaxonomyExtension() メソッド

getMeasureOrOperator　com.hitachi.xbrl.instance.XBRLUnit インターフェースの getMeasureOrOperator() メソッド

getMeasureOrOperatorType　com.hitachi.xbrl.instance.XBRLMeasureOrOperator インターフェースの getMeasureOrOperatorType() メソッド

getMeasures　com.hitachi.xbrl.instance.XBRLUnit2 インターフェースの getMeasures() メソッド

getMember　com.hitachi.xbrl.types.XBRLDateUnion クラスの getMember() メソッド

getMemberType　com.hitachi.xbrl.types.XBRLDateUnion クラスの getMemberType() メソッド

getName　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getName() メソッド

getName　com.hitachi.xbrl.instance.XBRLScenario インターフェースの getName() メソッド

getName　com.hitachi.xbrl.instance.XBRLSegment インターフェースの getName() メソッド

getName　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getName() メソッド

getName　com.hitachi.xbrl.taxonomy.XBRLPart インターフェースの getName() メソッド

getName　com.hitachi.xbrl.taxonomy. XBRLTaxonomyTypeDefinition インターフェースの getName() メソッド

getName　com.hitachi.xbrl.simple.taxonomy.SimpleElementDefinition インターフェースの getName() メソッド

getNamespaceDeclarationByPrefix
com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getNamespaceDeclarationByPrefix(java.lang.String prefix) メソッド

getNamespaceDeclarationByURI　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getNamespaceDeclarationByURI(java.lang.String uri) メソッド

getNamespaceDeclarations　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getNamespaceDeclarations() メソッド

getNamespaceDeclarations　com.hitachi.xbrl.taxonomy. XBRLTaxonomyDocument インターフェースの getNamespaceDeclarations() メソッド

getNamespaceDeclarationsByPrefix　com.hitachi.xbrl.taxonomy. XBRLTaxonomyDocument インターフェースの getNamespaceDeclarationsByPrefix(java.lang.String prefix) メソッド

getNamespaceDeclarationsByURI　com.hitachi.xbrl.taxonomy. XBRLTaxonomyDocument インターフェースの getNamespaceDeclarationsByURI(java.lang.String uri) メソッド

getNamespaceURI　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getNamespaceURI() メソッド

getNamespaceURI　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getNamespaceURI() メソッド

getNamespaceURI　com.hitachi.xbrl.taxonomy.XBRLPart インターフェースの getNamespaceURI() メソッド

getNamespaceURI　com.hitachi.xbrl.taxonomy. XBRLTaxonomyTypeDefinition インターフェースの getNamespaceURI() メソッド

getNamespaceURI　com.hitachi.xbrl.types.XBRLQName クラスの getNamespaceURI() メソッド

getNamespaceURI　com.hitachi.xbrl.util.XBRLSchemaLocation クラスの getNamespaceURI() メソッド

getNillable　com.hitachi.xbrl.simple.taxonomy.SimpleElementDefinition インターフェースの getNillable() メソッド

getNodeType　com.hitachi.xbrl.instance.XBRLInstanceNode インターフェースの getNodeType() メソッド

getNodeType　com.hitachi.xbrl.taxonomy.XBRLTaxonomyNode インターフェースの getNodeType() メソッド

getNodeType　com.hitachi.xbrl.simple.taxonomy.SimpleNode インターフェースの getNodeType() メソッド

getNumerator　com.hitachi.xbrl.instance.XBRLUnit2 インターフェースの getNumerator() メソッド

getNumerator　com.hitachi.xbrl.types.XBRLFraction クラスの getNumerator() メソッド

getOrder　com.hitachi.xbrl.link.XBRLRelationship インターフェースの getOrder() メ

ソッド

getOrder　com.hitachi.xbrl.simple.taxonomy.SimpleEdge インターフェースの getOrder() メソッド

getOwnerDocumentManager　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getOwnerDocumentManager() メソッド

getOwnerDocumentManager
com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの getOwnerDocumentManager() メソッド

getOwnerDocumentManager
com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの getOwnerDocumentManager() メソッド

getOwnerDocumentManager
com.hitachi.xbrl.instance.XBRLInstanceDocumentWriter インターフェースの getOwnerDocumentManager() メソッド

getOwnerDocumentManager
com.hitachi.xbrl.instance.XBRLInstanceDocumentWriterFactory インターフェースの getOwnerDocumentManager() メソッド

getOwnerDocumentManager　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getOwnerDocumentManager() メソッド

getOwnerDocumentManager
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの getOwnerDocumentManager() メソッド

getOwnerDocumentManager
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの getOwnerDocumentManager() メソッド

getOwnerDocumentManager　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの getOwnerDocumentManager() メソッド

getOwnerDocumentManager
com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomyBuilder インターフェースの getOwnerDocumentManager() メソッド

getOwnerDocumentManager
com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomyBuilderFactory インターフェースの getOwnerDocumentManager() メソッド

getOwnerDocumentManager　com.hitachi.xbrl.util.XBRLDocumentWriter インター

フェースの getOwnerDocumentManager() メソッド

getOwnerDocumentManager　com.hitachi.xbrl.util.XBRLDocumentWriterFactory インターフェースの getOwnerDocumentManager() メソッド

getOwnerInstanceDocument　com.hitachi.xbrl.instance.XBRLInstanceNode インターフェースの getOwnerInstanceDocument() メソッド

getOwnerTaxonomy　com.hitachi.xbrl.simple.taxonomy.SimpleNode インターフェースの getOwnerTaxonomy() メソッド

getOwnerTaxonomyDocument　com.hitachi.xbrl.taxonomy.XBRLTaxonomyNode インターフェースの getOwnerTaxonomyDocument() メソッド

getParentNode　com.hitachi.xbrl.instance.XBRLInstanceNode インターフェースの getParentNode() メソッド

getParentNode　com.hitachi.xbrl.taxonomy.XBRLTaxonomyNode インターフェースの getParentNode() メソッド

getParentNode　com.hitachi.xbrl.simple.taxonomy.SimpleNode インターフェースの getParentNode() メソッド

getParts　com.hitachi.xbrl.taxonomy.XBRLReference インターフェースの getParts() メソッド

getParts　com.hitachi.xbrl.taxonomy.XBRLReference インターフェースの getParts(java.lang.String namespaceURI, java.lang.String name) メソッド

getPeriod　com.hitachi.xbrl.instance.XBRLContext インターフェースの getPeriod() メソッド

getPeriodType　com.hitachi.xbrl.instance.XBRLPeriod インターフェースの getPeriodType() メソッド

getPeriodType　com.hitachi.xbrl.taxonomy.XBRLTaxonomyItem インターフェースの getPeriodType() メソッド

getPeriodType　com.hitachi.xbrl.simple.taxonomy.SimpleItemDefinition インターフェースの getPeriodType() メソッド

getPrecision　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの getPrecision() メソッド

getPrecision　com.hitachi.xbrl.instance.XBRLNumericContext インターフェースの getPrecision() メソッド

getPreferredLabel　com.hitachi.xbrl.link.XBRLPresentationRelationship インター

フェースの getPreferredLabel() メソッド

getPrefix　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getPrefix() メソッド

getPrefix　com.hitachi.xbrl.util.XBRLNamespaceDeclaration クラスの getPrefix() メソッド

getPrefix　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの getPrefix() メソッド

getPresentationRelationships　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getPresentationRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッド

getPriority　com.hitachi.xbrl.link.XBRLRelationship インターフェースの getPriority() メソッド

getProperty　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの getProperty() メソッド

getProperty　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの getProperty(java.lang.String name) メソッド

getProperty　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの getProperty(java.lang.String name) メソッド

getProperty　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの getProperty(java.lang.String name) メソッド

getPTVInstance　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getPTVInstance() メソッド

getPTVLinkbase()　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getPTVLinkbase() メソッド

getQName　com.hitachi.xbrl.XBRLDimensionException クラスの getQName() メソッド

getQName　com.hitachi.xbrl.link.XBRLDRSDimension インターフェースの getQName() メソッド

getQName　com.hitachi.xbrl.link.XBRLDRSMember インターフェースの getQName() メソッド

getReferenceRelationships　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getReferenceRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッド

getReferenceRelationships　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getReferenceRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッド

getReferences　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getReferences() メソッド

getReferences　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getReferences(java.lang.String role) メソッド

getReferences　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getReferences() メソッド

getReferences　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getReferences(java.lang.String role) メソッド

getReferences　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getReferences() メソッド

getReferences　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getReferences(java.lang.String role) メソッド

getRelationRoutes　com.hitachi.xbrl.link.XBRLIllegalCycleException クラスの getRelationRoutes() メソッド

getRhs　com.hitachi.xbrl.instance.XBRLOperator インターフェースの getRhs() メソッド

getRole　com.hitachi.xbrl.instance.XBRLLinkbaseRef インターフェースの getRole() メソッド

getRole　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの getRole() メソッド

getRole　com.hitachi.xbrl.link.XBRLDRSDimension インターフェースの getRole() メソッド

getRole　com.hitachi.xbrl.taxonomy.XBRLLabel インターフェースの getRole() メソッド

getRole　com.hitachi.xbrl.taxonomy. XBRLTaxonomyLinkbaseRef インターフェースの getRole() メソッド

getRole　com.hitachi.xbrl.taxonomy.XBRLReference インターフェースの getRole() メソッド

getRole　com.hitachi.xbrl.taxonomy. XBRLTaxonomyRoleRef インターフェースの getRole() メソッド

getRole　com.hitachi.xbrl.simple.taxonomy.SimpleEdge インターフェースの getRole() メソッド

getRole　com.hitachi.xbrl.simple.taxonomy.SimpleLabel インターフェースの getRole() メソッド

getRoleRef　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getRoleRef(java.lang.String roleURI) メソッド

getRoleRefs　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getRoleRefs() メソッド

getRoleRefs　com.hitachi.xbrl.taxonomy. XBRLTaxonomyDocument インターフェースの getRoleRefs() メソッド

getRoles　com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet インターフェースの getRoles() メソッド

getRoleType　com.hitachi.xbrl.instance.XBRLRoleRef インターフェースの getRoleType() メソッド

getRoleTypes　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getRoleTypes() メソッド

getRoleTypes　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getRoleTypes(java.lang.String roleURI) メソッド

getRoleURI　com.hitachi.xbrl.instance.XBRLRoleRef インターフェースの getRoleURI() メソッド

getRoleURI　com.hitachi.xbrl.taxonomy. XBRLTaxonomyRoleRef インターフェースの getRoleURI() メソッド

getRoleURI　com.hitachi.xbrl.taxonomy.XBRLRoleType インターフェースの getRoleURI() メソッド

getScenario　com.hitachi.xbrl.instance.XBRLContext インターフェースの getScenario() メソッド

getSchemaLocation　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getSchemaLocation(java.lang.String namespaceURI) メソッド

getSchemaLocation　com.hitachi.xbrl.util.XBRLSchemaLocation クラスの getSchemaLocation() メソッド

getSchemaLocation　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの getSchemaLocation() メソッド

getSchemaLocations　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getSchemaLocations() メソッド

getSchemaRefs　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getSchemaRefs() メソッド

getScheme　com.hitachi.xbrl.instance.XBRLIdentifier インターフェースの getScheme() メソッド

getSegment　com.hitachi.xbrl.instance.XBRLEntity インターフェースの getSegment() メソッド

getStartDate　com.hitachi.xbrl.instance.XBRLPeriod インターフェースの getStartDate() メソッド

getStartingPoints　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getStartingPoints() メソッド

getString　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの getString() メソッド

getSubstitutionGroup　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getSubstitutionGroup() メソッド

getSubstitutionGroup　com.hitachi.xbrl.simple.taxonomy.SimpleElementDefinition インターフェースの getSubstitutionGroup() メソッド

getTargetNamespace　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの getTargetNamespace() メソッド

getTargetNamespaceURI　com.hitachi.xbrl.taxonomy.XBRLTaxonomyBody インターフェースの getTargetNamespaceURI() メソッド

getTargetRole　com.hitachi.xbrl.link.XBRLHasHypercubeRelationship インターフェースの getTargetRole() メソッド

getTargetRole　com.hitachi.xbrl.link.XBRLHypercubeDimensionRelationship インターフェースの getTargetRole() メソッド

getTargetRole　com.hitachi.xbrl.link.XBRLDimensionDomainRelationship インターフェースの getTargetRole() メソッド

getTargetRole　com.hitachi.xbrl.link.XBRLDomainMemberRelationship インターフェースの getTargetRole() メソッド

getTaxonomyAttributeDeclarations　com.hitachi.xbrl.taxonomy.XBRLTaxonomyComplexTypeDefinition インターフェースの getTaxonomyAttributeDeclarations() メソッド

getTaxonomyAttributeDeclaration　com.hitachi.xbrl.taxonomy. XBRLTaxonomyComplexTypeDefinition インターフェースの getTaxonomyAttributeDeclaration(java.lang.String namespaceURI, java.lang.String name) メソッド

getTaxonomyBodies　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getTaxonomyBodies() メソッド

getTaxonomyBody　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getTaxonomyBody(java.lang.String targetNamespaceURI) メソッド

getTaxonomyComponents　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getTaxonomyComponents() メソッド

getTaxonomyDimensionItem　com.hitachi.xbrl.link.XBRLDRSDimension インターフェースの getTaxonomyDimensionItem() メソッド

getTaxonomyDocument　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getTaxonomyDocument() メソッド

getTaxonomyDocumentBuilder com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの getTaxonomyDocumentBuilder() メソッド

getTaxonomyElement　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの getTaxonomyElement() メソッド

getTaxonomyElement　com.hitachi.xbrl.link.XBRLDRSMember インターフェースの getTaxonomyElement() メソッド

getTaxonomyItem　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの getTaxonomyItem() メソッド

getTaxonomyItem　com.hitachi.xbrl.link.XBRLDimensionalRelationshipSet インターフェースの getTaxonomyItem() メソッド

getTaxonomyTuple　com.hitachi.xbrl.instance.XBRLInstanceTuple インターフェースの getTaxonomyTuple() メソッド

getTaxonomyTypeDefinition　com.hitachi.xbrl.taxonomy. XBRLTaxonomyElement インターフェースの getTaxonomyTypeDefinition() メソッド

getTaxonomyTypeDefinition　com.hitachi.xbrl.taxonomy. XBRLTaxonomyDocument インターフェースの getTaxonomyTypeDefinition(java.lang.String namespaceURI, java.lang.String name) メソッド

getTaxonomyTypeDefinitions　com.hitachi.xbrl.taxonomy. XBRLTaxonomyDocument

インターフェースの getTaxonomyTypeDefinitions() メソッド

getTitle　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの getTitle() メソッド

getTitle　com.hitachi.xbrl.taxonomy.XBRLLabel インターフェースの getTitle() メソッド

getTitle　com.hitachi.xbrl.taxonomy.XBRLReference インターフェースの getTitle() メソッド

getTitle　com.hitachi.xbrl.simple.taxonomy.SimpleLabel インターフェースの getTitle() メソッド

getTo　com.hitachi.xbrl.link.XBRLRelationship インターフェースの getTo() メソッド

getTo　com.hitachi.xbrl.simple.taxonomy.SimpleEdge インターフェースの getTo() メソッド

getType　com.hitachi.xbrl.taxonomy. XBRLTaxonomyAttributeDeclaration インターフェースの getType() メソッド

getType　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの getType() メソッド

getType　com.hitachi.xbrl.simple.taxonomy.SimpleElementDefinition インターフェースの getType() メソッド

getTypeCategory　com.hitachi.xbrl.taxonomy. XBRLTaxonomyTypeDefinition インターフェースの getTypeCategory() メソッド

getTypedDomainRef　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDimensionItem インターフェースの getTypedDomainRef() メソッド

getTypedDomainRefElement
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDimensionItem インターフェースの getTypedDomainRefElement() メソッド

getTypedMembers　com.hitachi.xbrl.instance.XBRLScenario インターフェースの getTypedMembers() メソッド

getTypedMembers　com.hitachi.xbrl.instance.XBRLSegment インターフェースの getTypedMembers() メソッド

getUnit　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの getUnit() メソッド

getUnit　com.hitachi.xbrl.instance.XBRLNonNumericContext インターフェースの getUnit() メソッド

getUnit　com.hitachi.xbrl.instance.XBRLNumericContext インターフェースの getUnit() メソッド

getUnits　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getUnits() メソッド

getUnitType　com.hitachi.xbrl.instance.XBRLUnit2 インターフェースの getUnitType() メソッド

getURI　com.hitachi.xbrl.util.XBRLNamespaceDeclaration クラスの getURI() メソッド

getUsable　com.hitachi.xbrl.link.XBRLDimensionDomainRelationship インターフェースの getUsable() メソッド

getUsable　com.hitachi.xbrl.link.XBRLDomainMemberRelationship インターフェースの getUsable() メソッド

getUse　com.hitachi.xbrl.simple.taxonomy.SimpleEdge インターフェースの getUse() メソッド

getUse　com.hitachi.xbrl.taxonomy. XBRLTaxonomyAttributeDeclaration インターフェースの getUse() メソッド

getUsedOn　com.hitachi.xbrl.taxonomy.XBRLArcroleType インターフェースの getUsedOn() メソッド

getUsedOn　com.hitachi.xbrl.taxonomy.XBRLRoleType インターフェースの getUsedOn() メソッド

getValue　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの getValue() メソッド

getValue　com.hitachi.xbrl.instance.XBRLMeasure インターフェースの getValue() メソッド

getValue　com.hitachi.xbrl.instance.XBRLExplicitMember インターフェースの getValue() メソッド

getValue　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの getValue() メソッド

getValue　com.hitachi.xbrl.instance.XBRLIdentifier インターフェースの getValue() メソッド

getValue　com.hitachi.xbrl.taxonomy.XBRLLabel インターフェースの getValue() メソッド

getValue　com.hitachi.xbrl.taxonomy.XBRLPart インターフェースの getValue() メソッ

ド

getValue　com.hitachi.xbrl.types.XBRLAbstractPrecision クラスの getValue() メソッド

getValue　com.hitachi.xbrl.simple.taxonomy.SimpleLabel インターフェースの getValue() メソッド

getW3CDOMDocument　com.hitachi.xbrl.taxonomy.XBRLTaxonomyComponent インターフェースの getW3CDOMDocument() メソッド

getW3CDOMDocument　com.hitachi.xbrl.util.PTVInstance クラスの getW3CDOMDocument() メソッド

getW3CDOMDocument　com.hitachi.xbrl.util.PTVLinkbase クラスの getW3CDOMDocument() メソッド

getW3CDOMNode　com.hitachi.xbrl.instance.XBRLInstanceNode インターフェースの getW3CDOMNode() メソッド

getW3CDOMNode　com.hitachi.xbrl.taxonomy.XBRLTaxonomyNode インターフェースの getW3CDOMNode() メソッド

getWeight　com.hitachi.xbrl.link.XBRLCalculationRelationship インターフェースの getWeight() メソッド

getXBRLVersion　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの getXBRLVersion() メソッド

getXBRLVersion　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの getXBRLVersion() メソッド

getXBRLVersion　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの getXBRLVersion() メソッド

getXBRLVersion　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocument インターフェースの getXBRLVersion() メソッド

getXBRLVersion　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの getXBRLVersion() メソッド

getXBRLVersion
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの getXBRLVersion() メソッド

H

hasExplicitMember　com.hitachi.xbrl.instance.XBRLScenario インターフェースの

hasExplicitMember(XBRLQName dimension) メソッド

hasExplicitMember　com.hitachi.xbrl.instance.XBRLScenario インターフェースの hasExplicitMember(XBRLQName dimension, XBRLQName value) メソッド

hasExplicitMember　com.hitachi.xbrl.instance.XBRLSegment インターフェースの hasExplicitMember(XBRLQName dimension) メソッド

hasExplicitMember　com.hitachi.xbrl.instance.XBRLSegment インターフェースの hasExplicitMember(XBRLQName dimension, XBRLQName value) メソッド

hashCode　com.hitachi.xbrl.types.XBRLGDay クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLGMonth クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLGMonthDay クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLGYear クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLGYearMonth クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLPrecision クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLQName クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLTime クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLAbstractBinary クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLAbstractPrecision クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLDate クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLDateTime クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLDateUnion クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLDecimals クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLDuration クラスの hashCode() メソッド

hashCode　com.hitachi.xbrl.types.XBRLFraction クラスの hashCode() メソッド

hasTypedMember　com.hitachi.xbrl.instance.XBRLScenario インターフェースの hasTypedMember(XBRLQName dimension) メソッド

hasTypedMember　com.hitachi.xbrl.instance.XBRLSegment インターフェースの hasTypedMember(XBRLQName dimension) メソッド

hasValue　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの hasValue() メソッド

I

importNode　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの importNode(XBRLInstanceNode sourceNode, boolean deep) メソッド

init　com.hitachi.xbrl.XBRLDocumentManager クラスの init() メソッド

init　com.hitachi.xbrl.XBRLDocumentManager クラスの init(java.lang.String path) メソッド

init　com.hitachi.xbrl.XBRLDocumentManager クラスの init(java.util.Properties properties) メソッド

insertChildElementBefore　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの insertChildElementBefore(XBRLInstanceElement newElement, XBRLInstanceElement refElement) メソッド

insertChildElementBefore　com.hitachi.xbrl.instance.XBRLInstanceTuple インターフェースの insertChildElementBefore(XBRLInstanceElement newElement, XBRLInstanceElement refElement) メソッド

insertDimensionMemberBefore　com.hitachi.xbrl.instance.XBRLScenario インターフェースの insertDimensionMemberBefore(XBRLDimensionMember newMember, XBRLDimensionMember refMember) メソッド

insertDimensionMemberBefore　com.hitachi.xbrl.instance.XBRLSegment インターフェースの insertDimensionMemberBefore(XBRLDimensionMember newMember, XBRLDimensionMember refMember) メソッド

intValue　com.hitachi.xbrl.types.XBRLAbstractPrecision クラスの intValue() メソッド

isAbstract　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの isAbstract() メソッド

isActive　com.hitachi.xbrl.instance.XBRLInstanceNode インターフェースの isActive() メソッド

isActive　com.hitachi.xbrl.simple.taxonomy.SimpleNode インターフェースの isActive() メソッド

isCoalescing　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの isCoalescing() メソッド

isCoalescing　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの isCoalescing() メソッド

isCreatingSchemaLocations
com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの isCreatingSchemaLocations() メソッド

isDelayLoading　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの isDelayLoading() メソッド

isDelayLoading　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの isDelayLoading() メソッド

isDelayLoading　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの isDelayLoading() メソッド

isDelayLoading
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの isDelayLoading() メソッド

isId　com.hitachi.xbrl.util.XBRLIdManager クラスの isId(java.lang.String id) メソッド

isIgnoringComments
com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの isIgnoringComments() メソッド

isIgnoringComments
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの isIgnoringComments() メソッド

isIgnoringZeroPrecision　com.hitachi.xbrl.util.XBRLEqualUtil クラスの isIgnoringZeroPrecision() メソッド

isIndent　com.hitachi.xbrl.instance.XBRLInstanceDocumentWriter インターフェースの isIndent() メソッド

isIndent　com.hitachi.xbrl.util.XBRLDocumentWriter インターフェースの isIndent() メソッド

isInfinite　com.hitachi.xbrl.types.XBRLAbstractPrecision クラスの isInfinite() メソッド

isNil　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの isNil() メソッド

isNillable　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの isNillable() メソッド

isNillable　com.hitachi.xbrl.taxonomy.XBRLTaxonomyElement インターフェースの

isNillable() メソッド

isNilAttributeSpecified　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの isNilAttributeSpecified() メソッド

isNumeric　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの isNumeric() メソッド

isNumeric　com.hitachi.xbrl.instance.XBRLContext インターフェースの isNumeric() メソッド

isNumeric　com.hitachi.xbrl.taxonomy.XBRLTaxonomyItem インターフェースの isNumeric() メソッド

isOrderAttributeSpecified　com.hitachi.xbrl.link.XBRLRelationship インターフェースの isOrderAttributeSpecified() メソッド

isRemote　com.hitachi.xbrl.taxonomy.XBRLLabel インターフェースの isRemote() メソッド

isRemote　com.hitachi.xbrl.taxonomy.XBRLReference インターフェースの isRemote() メソッド

isUsedOn　com.hitachi.xbrl.taxonomy.XBRLArcroleType インターフェースの isUsedOn(java.lang.String namespaceURI, java.lang.String name) メソッド

isUsedOn　com.hitachi.xbrl.taxonomy.XBRLRoleType インターフェースの isUsedOn(java.lang.String namespaceURI, java.lang.String name) メソッド

isValidating　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの isValidating() メソッド

isValidating　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの isValidating() メソッド

isValidating　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの isValidating() メソッド

isValidating　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの isValidating() メソッド

L

length　com.hitachi.xbrl.types.XBRLAbstractBinary クラスの length() メソッド

longValue　com.hitachi.xbrl.types.XBRLAbstractPrecision クラスの longValue() メソッド

N

newDocumentWriter　com.hitachi.xbrl.util.XBRLDocumentWriterFactory インターフェースの newDocumentWriter() メソッド

newDocumentWriterFactory　com.hitachi.xbrl.XBRLDocumentManager クラスの newDocumentWriterFactory() メソッド

newInstanceDocument　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument) メソッド

newInstanceDocument　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument, java.lang.String systemId) メソッド

newInstanceDocument　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument, org.w3c.dom.Document w3cdomDocument) メソッド

newInstanceDocument　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument, org.w3c.dom.Document w3cdomDocument, java.lang.String systemId) メソッド

newInstanceDocumentBuilder com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの newInstanceDocumentBuilder() メソッド

newInstanceDocumentBuilderFactory　com.hitachi.xbrl.XBRLDocumentManager クラスの newInstanceDocumentBuilderFactory() メソッド

newInstanceDocumentWriter com.hitachi.xbrl.instance.XBRLInstanceDocumentWriterFactory インターフェースの newInstanceDocumentWriter() メソッド

newInstanceDocumentWriterFactory　com.hitachi.xbrl.XBRLDocumentManager クラスの newInstanceDocumentWriterFactory() メソッド

newSimpleTaxonomy　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomyBuilder インターフェースの newSimpleTaxonomy(java.lang.String targetNamespace, java.lang.String prefix, java.lang.String fileName, XBRLTaxonomyDocument doc) メソッド

newSimpleTaxonomyBuilder com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomyBuilderFactory インターフェースの newSimpleTaxonomyBuilder() メソッド

newSimpleTaxonomyBuilderFactory　com.hitachi.xbrl.XBRLDocumentManager クラスの newSimpleTaxonomyBuilderFactory() メソッド

newTaxonomyDocumentBuilder
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの newTaxonomyDocumentBuilder() メソッド

newTaxonomyDocumentBuilderFactory　com.hitachi.xbrl.XBRLDocumentManager クラスの newTaxonomyDocumentBuilderFactory() メソッド

P

parse　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの parse(org.xml.sax.InputSource is) メソッド

parse　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの parse(java.io.InputStream is) メソッド

parse　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの parse(java.io.InputStream is, java.lang.String systemId) メソッド

parse　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの parse(java.io.File file) メソッド

parse　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの parse(java.lang.String uri) メソッド

parse　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの parse(org.w3c.dom.Element instanceRoot) メソッド

parse　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの parse(org.w3c.dom.Element instanceRoot, java.lang.String systemId) メソッド

parse　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの parse(java.io.File[] file) メソッド

parse　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの parse(java.lang.String[] uri) メソッド

parse　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの parse(org.xml.sax.InputSource[] is) メソッド

parse2　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの parse2(org.xml.sax.InputSource is) メソッド

parse2　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの parse2(java.io.InputStream is) メソッド

parse2　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの parse2(java.io.InputStream is, java.lang.String systemId) メソッド

parse2　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの parse2(java.io.File file) メソッド

parse2　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの parse2(java.lang.String uri) メソッド

R

registerDocument　com.hitachi.xbrl.util.XBRLIdManager クラスの registerDocument(XBRLInstanceDocument doc) メソッド

removeChildElement　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの removeChildElement(XBRLInstanceElement oldElement) メソッド

removeChildElement　com.hitachi.xbrl.instance.XBRLInstanceTuple インターフェースの removeChildElement(XBRLInstanceElement oldElement) メソッド

removeContext　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの removeContext(XBRLContext oldContext) メソッド

removeDimensionMember　com.hitachi.xbrl.instance.XBRLScenario インターフェースの removeDimensionMember(XBRLDimensionMember oldMember) メソッド

removeDimensionMember　com.hitachi.xbrl.instance.XBRLSegment インターフェースの removeDimensionMember(XBRLDimensionMember oldMember) メソッド

removeEdge　SimpleTaxonomy インターフェースの removeEdge(SimpleEdge obj) メソッド

removeElementDefinition　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの removeElementDefinition(SimpleElementDefinition obj) メソッド

removeFootnote　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの removeFootnote(XBRLFootnote oldFootnote) メソッド

removeFootnote　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの removeFootnote(XBRLFootnote oldFootnote) メソッド

removeFootnote　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの removeFootnote(XBRLFootnote oldFootnote, java.lang.String arcrole, java.math.BigDecimal order) メソッド

removeLinkbaseRef　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの removeLinkbaseRef(XBRLLinkbaseRef oldLinkbaseRef) メソッド

removeNamespaceDeclaration　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの removeNamespaceDeclaration(XBRLNamespaceDeclaration oldNamespaceDeclaration) メソッド

removeSchemaLocation　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの removeSchemaLocation(XBRLSchemaLocation oldSchemaLocation) メソッド

removeUnit　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの removeUnit(XBRLUnit2 oldUnit) メソッド

replaceChildElement　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの replaceChildElement(XBRLInstanceElement newElement, XBRLInstanceElement oldElement) メソッド

replaceChildElement　com.hitachi.xbrl.instance.XBRLInstanceTuple インターフェースの replaceChildElement(XBRLInstanceElement newElement, XBRLInstanceElement oldElement) メソッド

replaceDimensionMember　com.hitachi.xbrl.instance.XBRLScenario インターフェースの replaceDimensionMember(XBRLDimensionMember newMember, XBRLDimensionMember oldMember) メソッド

replaceDimensionMember　com.hitachi.xbrl.instance.XBRLSegment インターフェースの replaceDimensionMember(XBRLDimensionMember newMember, XBRLDimensionMember oldMember) メソッド

S

setAbstract　com.hitachi.xbrl.simple.taxonomy.SimpleElementDefinition インターフェースの setAbstract(boolean abstrakt) メソッド

setBalance　com.hitachi.xbrl.simple.taxonomy.SimpleItemDefinition インターフェースの setBalance(java.lang.String balance) メソッド

setBase　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの setBase(java.lang.String base) メソッド

setCoalescing　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの setCoalescing(boolean flag) メソッド

setCoalescing　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの setCoalescing(boolean flag) メソッド

setContext　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの setContext(XBRLContext context) メソッド

setCreatingSchemaLocations
com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの setCreatingSchemaLocations(boolean creating) メソッド

setCwa　com.hitachi.xbrl.instance.XBRLNumericContext インターフェースの setCwa(boolean cwa) メソッド

setDecimals　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの setDecimals(XBRLDecimals decimals) メソッド

setDelayLoading　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの setDelayLoading(boolean flag) メソッド

setDelayLoading
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの setDelayLoading(boolean flag) メソッド

setDimension　com.hitachi.xbrl.instance.XBRLDimensionMember インターフェースの setDimension(XBRLQName dimension) メソッド

setDimensionsVersion
com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの setDimensionsVersion(int version) メソッド

setDimensionsVersion
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの setDimensionsVersion(int version) メソッド

setEncoding　com.hitachi.xbrl.instance.XBRLInstanceDocumentWriter インターフェースの setEncoding(java.lang.String encoding) メソッド

setEncoding　com.hitachi.xbrl.util.XBRLDocumentWriter インターフェースの setEncoding(java.lang.String encoding) メソッド

setEntity　com.hitachi.xbrl.instance.XBRLContext インターフェースの setEntity(XBRLEntity entity) メソッド

setEntityResolver　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの setEntityResolver(org.xml.sax.EntityResolver entityResolver) メソッド

setEntityResolver　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの setEntityResolver(org.xml.sax.EntityResolver entityResolver) メソッド

setErrorHandler　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの setErrorHandler(XBRLParseErrorHandler errorHandler) メソッド

setErrorHandler　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilder インターフェースの setErrorHandler(XBRLParseErrorHandler errorHandler) メソッド

setExternalSchemaRefs
com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの setExternalSchemaRefs(java.lang.String[] uris) メソッド

setId　com.hitachi.xbrl.instance.XBRLInstanceDocument インターフェースの setId(java.lang.String id) メソッド

setId　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの setId(java.lang.String id) メソッド

setId　com.hitachi.xbrl.instance.XBRLContext インターフェースの setId(java.lang.String id) メソッド

setId　com.hitachi.xbrl.instance.XBRLUnit2 インターフェースの setId(java.lang.String id) メソッド

setId　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの setId(java.lang.String id) メソッド

setId　com.hitachi.xbrl.simple.taxonomy.SimpleElementDefinition インターフェースの setId() メソッド

setId　com.hitachi.xbrl.simple.taxonomy.SimpleLabel インターフェースの setId(java.lang.String id) メソッド

setIdCreator　com.hitachi.xbrl.util.XBRLIdManager クラスの setIdCreator(XBRLIdCreator creator) メソッド

setIdentifier　com.hitachi.xbrl.instance.XBRLEntity インターフェースの setIdentifier(XBRLIdentifier identifier) メソッド

setIgnoringComments
com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの setIgnoringComments(boolean flag) メソッド

setIgnoringComments
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの setIgnoringComments(boolean flag) メソッド

setIgnoringZeroPrecision　com.hitachi.xbrl.util.XBRLEqualUtil クラスの setIgnoringZeroPrecision(boolean ignoringZeroPrecision) メソッド

setIndent　com.hitachi.xbrl.instance.XBRLInstanceDocumentWriter インターフェースの setIndent(boolean indent) メソッド

setIndent　com.hitachi.xbrl.util.XBRLDocumentWriter インターフェースの setIndent(boolean indent) メソッド

setItemType　com.hitachi.xbrl.simple.taxonomy.SimpleItemDefinition インターフェースの setItemType(int type) メソッド

setKind　com.hitachi.xbrl.instance.XBRLOperator インターフェースの setKind(java.lang.String kind) メソッド

setLang　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの setLang(java.lang.String lang) メソッド

setLang　com.hitachi.xbrl.simple.taxonomy.SimpleLabel インターフェースの setLang(java.lang.String lang) メソッド

setLhs　com.hitachi.xbrl.instance.XBRLOperator インターフェースの setLhs(XBRLMeasureOrOperator lhs) メソッド

setMaxDocuments
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの setMaxDocuments(int max) メソッド

setMaxTaxonomyExtension
com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの setMaxTaxonomyExtension(int max) メソッド

setMeasureOrOperator　com.hitachi.xbrl.instance.XBRLUnit インターフェースの setMeasureOrOperator(XBRLMeasureOrOperator measureOrOperator) メソッド

setName　com.hitachi.xbrl.instance.XBRLScenario インターフェースの setName(java.lang.String name) メソッド

setName　com.hitachi.xbrl.instance.XBRLSegment インターフェースの setName(java.lang.String name) メソッド

setName　com.hitachi.xbrl.simple.taxonomy.SimpleElementDefinition インターフェースの setName(java.lang.String name) メソッド

setNil　com.hitachi.xbrl.instance.XBRLInstanceElement インターフェースの setNil(java.lang.Boolean nil) メソッド

setNillable　com.hitachi.xbrl.simple.taxonomy.SimpleElementDefinition インターフェースの setNillable(boolean nillable) メソッド

setOrder　com.hitachi.xbrl.simple.taxonomy.SimpleEdge インターフェースの setOrder(java.math.BigDecimal order) メソッド

setPeriod　com.hitachi.xbrl.instance.XBRLContext インターフェースの

setPeriod(XBRLPeriod period) メソッド

setPeriodType　com.hitachi.xbrl.simple.taxonomy.SimpleItemDefinition インターフェースの setPeriodType(java.lang.String periodType) メソッド

setPrecision　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの setPrecision(XBRLPrecision precision) メソッド

setPrecision　com.hitachi.xbrl.instance.XBRLNumericContext インターフェースの setPrecision(int precision) メソッド

setProperty　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの setProperty(java.lang.String name, java.lang.Object value) メソッド

setProperty　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの setProperty(java.lang.String name, java.lang.Object value) メソッド

setRhs　com.hitachi.xbrl.instance.XBRLOperator インターフェースの setRhs(XBRLMeasureOrOperator rhs) メソッド

setRole　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの setRole(java.lang.String role) メソッド

setRole　com.hitachi.xbrl.simple.taxonomy.SimpleLabel インターフェースの setRole(java.lang.String role) メソッド

setScenario　com.hitachi.xbrl.instance.XBRLContext インターフェースの setScenario(XBRLScenario scenario) メソッド

setScheme　com.hitachi.xbrl.instance.XBRLIdentifier インターフェースの setScheme(java.lang.String namespaceURI) メソッド

setSegment　com.hitachi.xbrl.instance.XBRLEntity インターフェースの setSegment(XBRLSegment segment) メソッド

setString　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの setString(java.lang.String value) メソッド

setTaxonomyDocumentBuilder
com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilder インターフェースの setTaxonomyDocumentBuilder(XBRLTaxonomyDocumentBuilder taxonomyDocumentBuilder) メソッド

setTitle　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの setTitle(java.lang.String title) メソッド

setTitle　com.hitachi.xbrl.simple.taxonomy.SimpleLabel インターフェースの

setTitle(java.lang.String title) メソッド

setUnit　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの setUnit(XBRLUnit2 unit) メソッド

setUnit　com.hitachi.xbrl.instance.XBRLNonNumericContext インターフェースの setUnit(XBRLUnit unit) メソッド

setUnit　com.hitachi.xbrl.instance.XBRLNumericContext インターフェースの setUnit(XBRLUnit unit) メソッド

setValidating　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの setValidating(boolean flag) メソッド

setValidating　com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの setValidating(boolean flag) メソッド

setValue　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの setValue(java.lang.Object value) メソッド

setValue　com.hitachi.xbrl.instance.XBRLMeasure インターフェースの setValue(XBRLQName value) メソッド

setValue　com.hitachi.xbrl.instance.XBRLExplicitMember インターフェースの setValue(XBRLQName value) メソッド

setValue　com.hitachi.xbrl.instance.XBRLFootnote インターフェースの setValue(java.lang.String value) メソッド

setValue　com.hitachi.xbrl.instance.XBRLIdentifier インターフェースの setValue(java.lang.String value) メソッド

setValue　com.hitachi.xbrl.simple.taxonomy.SimpleLabel インターフェースの setId(java.lang.String value) メソッド

setXBRLVersion　com.hitachi.xbrl.instance.XBRLInstanceDocumentBuilderFactory インターフェースの setXBRLVersion(int version) メソッド

setXBRLVersion com.hitachi.xbrl.taxonomy.XBRLTaxonomyDocumentBuilderFactory インターフェースの setXBRLVersion(int version) メソッド

s_equalContext　com.hitachi.xbrl.util.XBRLEqualUtil クラスの s_equalContext(XBRLContext2 context1, XBRLContext2 context2) メソッド

s_equalItem　com.hitachi.xbrl.util.XBRLEqualUtil クラスの s_equalItem(XBRLInstanceItem item1, XBRLInstanceItem item2) メソッド

s_equalTuple　com.hitachi.xbrl.util.XBRLEqualUtil クラスの s_equalTuple(XBRLInstanceTuple tuple1, XBRLInstanceTuple tuple2) メソッド

s_equalUnit　com.hitachi.xbrl.util.XBRLEqualUtil クラスの s_equalUnit(XBRLUnit2 unit1, XBRLUnit2 unit2) メソッド

T

toByteArray　com.hitachi.xbrl.types.XBRLHexBinary クラスの toByteArray(java.lang.String hex) メソッド

toByteArray　com.hitachi.xbrl.types.XBRLAbstractBinary クラスの toByteArray() メソッド

toByteArray　com.hitachi.xbrl.types.XBRLBase64Binary クラスの toByteArray(java.lang.String base64) メソッド

toString　com.hitachi.xbrl.types.XBRLGDay クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLGMonth クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLGMonthDay クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLGYear クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLGYearMonth クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLHexBinary クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLHexBinary クラスの toString(byte[] bytes) メソッド

toString　com.hitachi.xbrl.types.XBRLHexBinary クラスの toString(byte[] bytes, int offset, int length) メソッド

toString　com.hitachi.xbrl.types.XBRLQName クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLQName クラスの toString(java.lang.String namespaceURI, java.lang.String localPart) メソッド

toString　com.hitachi.xbrl.types.XBRLTime クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLAbstractBinary クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLAbstractPrecision クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLBase64Binary クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLBase64Binary クラスの toString(byte[] bytes) メソッド

toString　com.hitachi.xbrl.types.XBRLBase64Binary クラスの toString(byte[] bytes, int offset, int length) メソッド

toString　com.hitachi.xbrl.types.XBRLDate クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLDateTime クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLDateUnion クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLDuration クラスの toString() メソッド

toString　com.hitachi.xbrl.types.XBRLFraction クラスの toString() メソッド

transform　com.hitachi.xbrl.simple.taxonomy.SimpleTaxonomy インターフェースの transform() メソッド

U

unregisterDocument　com.hitachi.xbrl.util.XBRLIdManager クラスの unregisterDocument(XBRLInstanceDocument doc) メソッド

V

validateDimensions　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの validateDimensions() メソッド

validateDimensions　com.hitachi.xbrl.instance.XBRLInstanceItem インターフェースの validateDimensions(java.lang.String role) メソッド

valueOf　com.hitachi.xbrl.types.XBRLHexBinary クラスの valueOf(byte[] bytes, int offset, int length) メソッド

valueOf　com.hitachi.xbrl.types.XBRLHexBinary クラスの valueOf(java.lang.String hex) メソッド

valueOf　com.hitachi.xbrl.types.XBRLHexBinary クラスの valueOf(byte[] bytes) メソッド

valueOf　com.hitachi.xbrl.types.XBRLQName クラスの valueOf(java.lang.String qname) メソッド

valueOf　com.hitachi.xbrl.types.XBRLQName クラスの valueOf(java.lang.String namespaceURI, java.lang.String localPart) メソッド

valueOf　com.hitachi.xbrl.types.XBRLBase64Binary クラスの valueOf(byte[] bytes, int offset, int length) メソッド

valueOf　com.hitachi.xbrl.types.XBRLBase64Binary クラスの valueOf(java.lang.String base64) メソッド

valueOf　com.hitachi.xbrl.types.XBRLBase64Binary クラスの valueOf(byte[] bytes) メソッド

W

warning　com.hitachi.xbrl.XBRLParseErrorHandler インターフェースの warning(XBRLDOMException exception) メソッド

write　com.hitachi.xbrl.instance.XBRLInstanceDocumentWriter インターフェースの write(java.io.OutputStream os, XBRLInstanceDocument doc) メソッド

write　com.hitachi.xbrl.util.XBRLDocumentWriter インターフェースの write(java.io.OutputStream os, org.w3c.dom.Document doc) メソッド

write　com.hitachi.xbrl.util.XBRLDocumentWriter インターフェースの write(java.io.OutputStream os, XBRLInstanceDocument doc) メソッド

X

XBRLBase64Binary　com.hitachi.xbrl.types.XBRLBase64Binary クラスの XBRLBase64Binary(java.lang.String base64) コンストラクタ

XBRLBase64Binary　com.hitachi.xbrl.types.XBRLBase64Binary クラスの XBRLBase64Binary(byte[] bytes) コンストラクタ

XBRLBase64Binary　com.hitachi.xbrl.types.XBRLBase64Binary クラスの XBRLBase64Binary(byte[] bytes, int offset, int length) コンストラクタ

XBRLDate　com.hitachi.xbrl.types.XBRLDate クラスの XBRLDate(java.lang.String date) コンストラクタ

XBRLDate　com.hitachi.xbrl.types.XBRLDate クラスの XBRLDate(XBRLDate other) コンストラクタ

XBRLDate　com.hitachi.xbrl.types.XBRLDate クラスの XBRLDate(XBRLDateTime other) コンストラクタ

XBRLDateTime　com.hitachi.xbrl.types.XBRLDateTime クラスの XBRLDateTime(java.lang.String datetime) コンストラクタ

XBRLDateTime　com.hitachi.xbrl.types.XBRLDateTime クラスの XBRLDateTime(XBRLDateTime other) コンストラクタ

XBRLDateTime　com.hitachi.xbrl.types.XBRLDateTime クラスの XBRLDateTime(XBRLDate other) コンストラクタ

XBRLDateUnion　com.hitachi.xbrl.types.XBRLDateUnion クラスの XBRLDateUnion(java.lang.String dateunion) コンストラクタ

XBRLDateUnion　com.hitachi.xbrl.types.XBRLDateUnion クラスの XBRLDateUnion(XBRLDate date) コンストラクタ

XBRLDateUnion　com.hitachi.xbrl.types.XBRLDateUnion クラスの XBRLDateUnion(XBRLDateTime datetime) コンストラクタ

XBRLDecimals　com.hitachi.xbrl.types.XBRLDecimals クラスの XBRLDecimals(java.math.BigInteger value) コンストラクタ

XBRLDecimals　com.hitachi.xbrl.types.XBRLDecimals クラスの XBRLDecimals(java.lang.String decimals) コンストラクタ

XBRLDimensionUserException　com.hitachi.xbrl.XBRLDimensionUserException クラスの XBRLDimensionUserException(int code, javax.xml.namespace.QName qname) コンストラクタ

XBRLDimensionUserException　com.hitachi.xbrl.XBRLDimensionUserException クラスの XBRLDimensionUserException(int code, java.lang.String message, javax.xml.namespace.QName qname) コンストラクタ

XBRLDimensionUserException　com.hitachi.xbrl.XBRLDimensionUserException クラスの XBRLDimensionUserException(int code, java.lang.Throwable cause, javax.xml.namespace.QName qname) コンストラクタ

XBRLDimensionUserException　com.hitachi.xbrl.XBRLDimensionUserException クラスの XBRLDimensionUserException(int code, java.lang.String message, java.lang.Throwable cause, javax.xml.namespace.QName qname) コンストラクタ

XBRLDOMUserException　com.hitachi.xbrl.XBRLDOMUserException クラスの XBRLDOMUserException(int code) コンストラクタ

XBRLDOMUserException　com.hitachi.xbrl.XBRLDOMUserException クラスの XBRLDOMUserException(int code, java.lang.String message) コンストラクタ

XBRLDOMUserException　com.hitachi.xbrl.XBRLDOMUserException クラスの XBRLDOMUserException(int code, java.lang.Throwable cause) コンストラクタ

XBRLDOMUserException　com.hitachi.xbrl.XBRLDOMUserException クラスの XBRLDOMUserException(int code, java.lang.String message, java.lang.Throwable cause) コンストラクタ

XBRLDuration　com.hitachi.xbrl.types.XBRLDuration クラスの XBRLDuration(java.lang.String duration) コンストラクタ

XBRLEqualUtil　com.hitachi.xbrl.util.XBRLEqualUtil クラスの XBRLEqualUtil() コンストラクタ

XBRLEqualUtil　com.hitachi.xbrl.util.XBRLEqualUtil クラスの
XBRLEqualUtil(boolean ignoringZeroPrecision) コンストラクタ

XBRLFraction　com.hitachi.xbrl.types.XBRLFraction クラスの
XBRLFraction(java.math.BigDecimal numerator, java.math.BigDecimal denominator)
コンストラクタ

XBRLFraction　com.hitachi.xbrl.types.XBRLFraction クラスの
XBRLFraction(java.lang.String fraction) コンストラクタ

XBRLGDay　com.hitachi.xbrl.types.XBRLGDay クラスの
XBRLGDay(java.lang.String gday) コンストラクタ

XBRLGMonth　com.hitachi.xbrl.types.XBRLGMonth クラスの
XBRLGMonth(java.lang.String gmonth) コンストラクタ

XBRLGMonthDay　com.hitachi.xbrl.types.XBRLGMonthDay クラスの
XBRLGMonthDay(java.lang.String gmonthday) コンストラクタ

XBRLGYear　com.hitachi.xbrl.types.XBRLGYear クラスの
XBRLGYear(java.lang.String gyear) コンストラクタ

XBRLGYearMonth　com.hitachi.xbrl.types.XBRLGYearMonth クラスの
XBRLGYearMonth(java.lang.String gyearmonth) コンストラクタ

XBRLHexBinary　com.hitachi.xbrl.types.XBRLHexBinary クラスの
XBRLHexBinary(java.lang.String hex) コンストラクタ

XBRLHexBinary　com.hitachi.xbrl.types.XBRLHexBinary クラスの
XBRLHexBinary(byte[] bytes) コンストラクタ

XBRLHexBinary　com.hitachi.xbrl.types.XBRLHexBinary クラスの
XBRLHexBinary(byte[] bytes, int offset, int length) コンストラクタ

XBRLIdManager　com.hitachi.xbrl.util.XBRLIdManager クラスの XBRLIdManager()
コンストラクタ

XBRLNamespaceDeclaration　com.hitachi.xbrl.util.XBRLNamespaceDeclaration ク
ラスの XBRLNamespaceDeclaration(java.lang.String prefix, java.lang.String uri) コン
ストラクタ

XBRLNamespaceDeclaration　com.hitachi.xbrl.util.XBRLNamespaceDeclaration ク
ラスの XBRLNamespaceDeclaration(java.lang.String prefix, java.lang.String uri,
boolean validate) コンストラクタ

XBRLPrecision　com.hitachi.xbrl.types.XBRLPrecision クラスの
XBRLPrecision(java.math.BigInteger value) コンストラクタ

XBRLPrecision　com.hitachi.xbrl.types.XBRLPrecision クラスの XBRLPrecision(java.lang.String precision) コンストラクタ

XBRLQName　com.hitachi.xbrl.types.XBRLQName クラスの XBRLQName(java.lang.String qname) コンストラクタ

XBRLQName　com.hitachi.xbrl.types.XBRLQName クラスの XBRLQName(java.lang.String namespaceURI, java.lang.String localPart) コンストラクタ

XBRLQName　com.hitachi.xbrl.types.XBRLQName クラスの XBRLQName(java.lang.String qname, boolean validate) コンストラクタ

XBRLQName　com.hitachi.xbrl.types.XBRLQName クラスの XBRLQName(java.lang.String namespaceURI, java.lang.String localPart, boolean validate) コンストラクタ

XBRLSchemaLocation　com.hitachi.xbrl.util.XBRLSchemaLocation クラスの XBRLSchemaLocation(java.lang.String namespaceURI, java.lang.String schemaLocation) コンストラクタ

XBRLSchemaLocation　com.hitachi.xbrl.util.XBRLSchemaLocation クラスの XBRLSchemaLocation(java.lang.String namespaceURI, java.lang.String schemaLocation, boolean validate) コンストラクタ

XBRLTime　com.hitachi.xbrl.types.XBRLTime クラスの XBRLTime(java.lang.String time) コンストラクタ

6.4 API リファレンスの形式

このマニュアルでは，API リファレンスを次の形式で記載しています。

6.4.1 パッケージの形式

パッケージの形式を次に示します。

インターフェースの概要：
インターフェースの概要を説明します。

クラスの概要：
クラスが存在するパッケージの場合，クラスの概要を説明します。

例外の概要：
例外が存在するパッケージの場合，クラスの概要を説明します。

6.4.2 インターフェースおよびクラスの形式

インターフェースおよびクラスの形式を次に示します。

XBRL のバージョン：
このインターフェースおよびクラスを使用できる XBRL のバージョンを示します。

XBRL Dimensions のバージョン：
このインターフェースおよびクラスを XBRL Dimensions で使用できる場合に，使用できる XBRL Dimensions のバージョンを示します。この項目が記載されているインターフェースおよびクラスは，XBRL 文書が XBRL Dimensions に準拠している場合だけ使用できます。

説明：
このインターフェースおよびクラスについて説明します。

メソッドの概要：
メソッドが存在するインターフェースおよびクラスの場合，メソッドの概要を説明します。

6.4.3 メソッドの形式

XBRL のバージョン：
このメソッドを使用できる XBRL のバージョンを示します。

XBRL Dimensions のバージョン：
このメソッドを XBRL Dimensions で使用できる場合に，使用できる XBRL Dimensions のバージョンを示します。この項目が記載されているメソッドは，

XBRL 文書が XBRL Dimensions に準拠している場合だけ使用できます。

説明：

このメソッドについて説明します。

引数：

引数が存在するメソッドの場合，引数について説明します。

戻り値

戻り値が存在するメソッドの場合，戻り値について説明します。

例外

例外が存在するメソッドの場合，例外について説明します。

6.5 com.hitachi.xbrl パッケージ

com.hitachi.xbrl パッケージは，XBRL 文書を操作するための XBRL DOM のインターフェースを提供します。

インターフェースの概要：

com.hitachi.xbrl パッケージのインターフェースを次の表に示します。

表 6-4 com.hitachi.xbrl パッケージのインターフェースの概要

インターフェース	説明	XBRL のバージョン
XBRLParseErrorHandler	XBRL 文書の解析時に検出されたエラー通知を受け取るためのインターフェースです。	2.0 および 2.1

クラスの概要：

com.hitachi.xbrl パッケージのクラスを次の表に示します。

表 6-5 com.hitachi.xbrl パッケージのクラスの概要

クラス	説明	XBRL のバージョン
XBRLDocumentManager	XBRL 文書の入出力や，タクソノミー文書のキャッシュを管理します。	2.0 および 2.1

例外の概要：

com.hitachi.xbrl パッケージの例外の概要を次の表に示します。

表 6-6 com.hitachi.xbrl パッケージの例外の概要

項番	クラス	説明	XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLDOMException	解析エラーなど，発生することがあらかじめ予想される種類の例外通知のためにスローされます。	2.0 および 2.1	該当しない
2	XBRLDOMRuntimeException	アプリケーションが正常に動作している状態では発生しない種類の例外通知のためにスローされます。	2.0 および 2.1	該当しない
3	XBRLDOMUserException	ユーザーアプリケーションで XBRLDOMException を明示的にスローしたい場合に使用するクラスです。	2.0 および 2.1	該当しない
4	XBRLDimensionException	XBRL Dimensions の構文解析エラーなど，発生することがあらかじめ予想される種類の例外通知のためにスローされます。	2.1	1.0

項番	クラス	説明	XBRL のバージョン	XBRL Dimensions のバージョン
5	XBRLDimensionUserException	ユーザーアプリケーションで XBRLDimensionException を明示的にスローしたい場合に使用するクラスです。	2.1	1.0

6.5.1 XBRLParseErrorHandler インターフェース

```
public interface XBRLParseErrorHandler
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRL 文書の解析時に検出されたエラーの通知を受け取るためのインターフェースです。

XBRL 文書の解析時に発生したエラーの通知を，ユーザーアプリケーションで受け取りたい場合は，このインターフェースを実装したクラスを用意する必要があります。このインターフェースを実装したクラスのインスタンスを，XBRLInstanceDocumentBuilder や XBRLTaxonomyDocumentBuilder の setErrorHandler() メソッドで指定することで，XBRL 文書の解析時に発生したすべてのエラーについての通知を受け取ることができます。

warning()，error()，fatalError() の各メソッドでは，XBRLDOMException をスローして解析処理を中断させることができます。このとき，warning()，error()，fatalError() の各メソッドからスローされた例外は，XBRLInstanceDocumentBuilder や XBRLTaxonomyDocumentBuilder の parse() メソッドの呼び出し元へそのままスローされます。

メソッドの概要：

XBRLParseErrorHandler インターフェースのメソッドの概要を次の表に示します。

表 6-7　XBRLParseErrorHandler インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	void	warning(XBRLDOMException exception) 警告の通知を受け取ります。	2.0 および 2.1
2	void	error(XBRLDOMException exception) 回復できるエラーの通知を受け取ります。	2.0 および 2.1
3	void	fatalError(XBRLDOMException exception) 致命的なエラーの通知を受け取ります。	2.0 および 2.1

(1) warning

```
public void warning(XBRLDOMException exception)
            throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

警告の通知を受け取ります。このメソッドの呼び出しで例外がスローされなかった場合，XBRL DOM は解析処理を続行します。

引数：

exception - カプセル化された警告情報が指定されます。

例外：

XBRLDOMException -

- 解析処理を中断させる場合は，引数 exception に指定した例外をそのままスローするか，または XBRLDOMUserException をスローしてください。

(2) error

```
public void error(XBRLDOMException exception)
            throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

回復できるエラーの通知を受け取ります。このメソッドの呼び出しで例外がスローされなかった場合，XBRL DOM は解析処理を続行します。

引数：

exception - カプセル化されたエラー情報が指定されます。

例外：

XBRLDOMException -

- 解析処理を中断させる場合は，引数 exception に指定した例外をそのままスローするか，または，XBRLDOMUserException をスローしてください。

(3) fatalError

```
public void fatalError(XBRLDOMException exception)
            throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

致命的なエラーの通知を受け取ります。このメソッドの呼び出しで例外がスローされなかった場合でも，XBRL DOM は必ず解析処理を中断します。このメソッドの呼

び出しで例外がスローされなかった場合は，XBRLInstanceDocumentBuilder または XBRLTaxonomyDocumentBuilder の parse() メソッドの呼び出し元へ，引数 exception に指定した例外がスローされます。

引数：

exception - カプセル化されたエラー情報が指定されます。

例外：

XBRLDOMException -

- 引数 exception に指定した例外をそのままスローするか，または，XBRLDOMUserException をスローしてください。

6.5.2 XBRLDocumentManager クラス

```
public abstract class XBRLDocumentManager
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRL 文書の入出力や，タクソノミー文書のキャッシュを管理します。
XBRL DOM を使用する前に，init() メソッドを呼び出して XBRL DOM を初期化する必要があります。XBRL DOM を初期化すると，XBRLDocumentManager を実装したクラスのインスタンスがシングルトンとして生成されます。生成された XBRLDocumentManager を実装したクラスのインスタンスは getInstance() メソッドで取得できます。

メソッドの概要：

XBRLDocumentManager クラスのメソッドの概要を次の表に示します。

表 6-8 XBRLDocumentManager クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	static void	init() XBRL DOM を初期化します。	2.0 および 2.1
2	static void	init(java.lang.String path) XBRL DOM を初期化します。	2.0 および 2.1
3	static void	init(java.util.Properties properties) XBRL DOM を初期化します。	2.0 および 2.1
4	static XBRLDocument Manager	getInstance() XBRLDocumentManager を実装したクラスのインスタンスを取得します。	2.0 および 2.1

項番	説明		XBRL のバージョン
5	abstract XBRLInstanceDocumentBuilderFactory	newInstanceDocumentBuilderFactory() XBRLInstanceDocumentBuilderFactory オブジェクトを生成します。	2.0 および 2.1
6	abstract XBRLInstanceDocumentWriterFactory	newInstanceDocumentWriterFactory() XBRLInstanceDocumentWriterFactory オブジェクトを生成します。	2.0 および 2.1
7	abstract XBRLTaxonomyDocumentBuilderFactory	newTaxonomyDocumentBuilderFactory() XBRLTaxonomyDocumentBuilderFactory オブジェクトを生成します。	2.0 および 2.1
8	abstract XBRLDocumentWriterFactory	newDocumentWriterFactory() XBRLDocumentWriterFactory オブジェクトを生成します。	2.1
9	abstract SimpleTaxonomyBuilderFactory	newSimpleTaxonomyBuilderFactory() SimpleTaxonomyBuilderFactory オブジェクトを生成します。	2.1
10	abstract void	clearCache() キャッシュされているすべてのタクソノミー文書を，キャッシュから削除します。	2.0 および 2.1

(1) init

```
public static void init()
                    throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRL DOM を初期化します。

このメソッドでは，XBRL DOM の動作を制御するための各プロパティにデフォルト値が設定されたものとして，初期化処理します。プロパティのデフォルト値については，「4.16.2　プロパティ一覧」を参照してください。

各プロパティの値を明示的に指定して初期化処理する場合は，この init() メソッドではなく，init(java.lang.String) メソッドまたは init(java.util.Properties) メソッドを使用します。

例外：

XBRLDOMException -

- XBRL DOM の初期化処理でエラーが発生した場合にスローされます。［エラーコード：39000］

XBRLDOMRuntimeException -

- XBRL DOM がすでに初期化されている場合にスローされます。［エラーコード：20000］

(2) init

```
public static void init(java.lang.String path)
                    throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRL DOM を初期化します。

このメソッドは，指定したプロパティファイルの内容に従って XBRL DOM を初期化します。プロパティファイルに記述できるプロパティの詳細，およびプロパティファイルの記述方法については，「4.16.2　プロパティ一覧」を参照してください。

引数 path に指定したプロパティファイルで定義されていないプロパティについては，デフォルト値が設定されます。そのため，引数 path に指定するプロパティファイルでは，すべてのプロパティを定義する必要はありません。

引数：

path - XBRL DOM のプロパティを記述したプロパティファイルの絶対パス名を指定します。

例外：

XBRLDOMException -

- プロパティファイルの読み込み中にエラーが発生した場合にスローされます。［エラーコード：39200，39203，39204，39205，39206］
- XBRL DOM の初期化処理でエラーが発生した場合にスローされます。［エラーコード：39000］

XBRLDOMRuntimeException -

- XBRL DOM がすでに初期化されている場合にスローされます。［エラーコード：20000］
- 引数 path に null を指定した場合にスローされます。［エラーコード：20002］

(3) init

```
public static void init(java.util.Properties properties)
                    throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRL DOM を初期化します。

このメソッドは，指定したプロパティの設定値に従って XBRL DOM を初期化します。XBRL DOM の動作を制御するために指定できるプロパティの詳細については，「4.16.2　プロパティ一覧」を参照してください。

引数 properties に指定した java.util.Properties オブジェクトで定義されていないプロパティについては，デフォルト値が設定されます。そのため，引数 properties に

指定する java.util.Properties オブジェクトでは，すべてのプロパティを定義する必要はありません。

引数：

properties - XBRL DOM のプロパティを格納している java.util.Properties オブジェクトを指定します。

例外：

XBRLDOMException -

- プロパティの読み取り中にエラーが発生した場合にスローされます。［エラーコード：39200，39206］
- XBRL DOM の初期化処理でエラーが発生した場合にスローされます。［エラーコード：39000］

XBRLDOMRuntimeException -

- XBRL DOM がすでに初期化されている場合にスローされます。［エラーコード：20000］
- 引数 properties に null を指定した場合にスローされます。［エラーコード：20002］

（4） getInstance

```
public static XBRLDocumentManager getInstance()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDocumentManager を実装したクラスのインスタンスを取得します。
このメソッドを呼び出す前に，init() メソッドを呼び出して，XBRL DOM を初期化しておく必要があります。

戻り値：

XBRLDocumentManager を実装したクラスのインスタンスを返します。

例外：

XBRLDOMRuntimeException -

- XBRL DOM が初期化されていない場合にスローされます。［エラーコード：20001］

使用例：

```
XBRLDocumentManager dm = XBRLDocumentManager.getInstance();
XBRLTaxonomyDocumentBuilderFactory tbf =
dm.newTaxonomyDocumentBuilderFactory();
```

（5） newInstanceDocumentBuilderFactory

```
public abstract XBRLInstanceDocumentBuilderFactory
newInstanceDocumentBuilderFactory()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLInstanceDocumentBuilderFactory オブジェクトを生成します。

戻り値：

新しく生成された XBRLInstanceDocumentBuilderFactory オブジェクトを返します。

（6） newInstanceDocumentWriterFactory

```
public abstract XBRLInstanceDocumentWriterFactory
newInstanceDocumentWriterFactory()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLInstanceDocumentWriterFactory オブジェクトを生成します。

戻り値：

新しく生成された XBRLInstanceDocumentWriterFactory オブジェクトを返します。

（7） newTaxonomyDocumentBuilderFactory

```
public abstract XBRLTaxonomyDocumentBuilderFactory
newTaxonomyDocumentBuilderFactory()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLTaxonomyDocumentBuilderFactory オブジェクトを生成します。

戻り値：

新しく生成された XBRLTaxonomyDocumentBuilderFactory オブジェクトを返します。

（8） newDocumentWriterFactory

```
public abstract XBRLDocumentWriterFactory
newDocumentWriterFactory()
```

XBRL のバージョン：

2.1

説明：

XBRLDocumentWriterFactory オブジェクトを生成します。

戻り値：

新たに生成された XBRLDocumentWriterFactory オブジェクトを返します。

(9) newSimpleTaxonomyBuilderFactory

```
public abstract SimpleTaxonomyBuilderFactory
newSimpleTaxonomyBuilderFactory()
```

XBRL のバージョン：

2.1

説明：

SimpleTaxonomyBuilderFactory オブジェクトを生成します。

戻り値：

新たに生成された SimpleTaxonomyBuilderFactory オブジェクトを返します。

(10) clearCache

```
public abstract void clearCache()
```

XBRL のバージョン：

2.0 および 2.1

説明：

キャッシュされているすべてのタクソノミー文書を，キャッシュから削除します。

6.5.3 XBRLDOMException クラス

```
public class XBRLDOMException
extends java.lang.Exception
```

XBRL のバージョン：

2.0 および 2.1

説明：

解析エラーなど，発生することがあらかじめ予想される種類の例外通知のために，uCosminexus Business Reporting Processor がスローします。

ユーザーアプリケーションで XBRLDOMException を明示的にスローしたい場合は，XBRLDOMException から派生した XBRLDOMUserException を使用します。

メソッドの概要：

XBRLDOMException クラスのメソッドの概要を次の表に示します。

表 6-9 XBRLDOMException クラスのメソッドの概要

説明		XBRL のバージョン
int	getCode() エラーコードを取得します。	2.0 および 2.1

(1) getCode

```
public int getCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

エラーコードを取得します。エラーコードは，メッセージ ID「KEBRnnnnn-E」のうち，nnnnn の整数値です。エラーコードの詳細は「7. メッセージ」を参照してください。

戻り値：

エラーコードを返します。

6.5.4 XBRLDOMRuntimeException クラス

```
public class XBRLDOMRuntimeException
extends java.lang.RuntimeException
```

XBRL のバージョン：

2.0 および 2.1

説明：

メソッドへの入力や呼び出し方法に誤りがある場合など，アプリケーションが正常に動作している状態では通常発生しないような種類の例外を通知するために，uCosminexus Business Reporting Processor がスローします。

メソッドの概要：

XBRLDOMRuntimeException クラスのメソッドの概要を次の表に示します。

表 6-10 XBRLDOMRuntimeException クラスのメソッドの概要

説明		XBRL のバージョン
int	getCode() エラーコードを取得します。	2.0 および 2.1

(1) getCode

```
public int getCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

エラーコードを取得します。エラーコードは，メッセージ ID「KEBRnnnnn-E」のうち，nnnnn の整数値です。エラーコードの詳細は「7. メッセージ」を参照してください。

戻り値：

エラーコードを返します。

6.5.5 XBRLDOMUserException クラス

```
public class XBRLDOMUserException
extends XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

ユーザーアプリケーションで XBRLDOMException を明示的にスローしたい場合に使用するクラスです。

XBRLDOMUserException クラスは，XBRLDOMException からの派生クラスです。

XBRLDOMUserException クラスのコンストラクタではエラーコードを指定します。利用できるエラーコードの値の範囲は 90000 ～ 99999 です。ただし，指定したエラーコードが 90000 ～ 99999 の範囲内の値かどうかはチェックされません。このため，範囲外の値を指定してもエラーとはなりません。

コンストラクタの概要：

XBRLDOMUserException クラスのコンストラクタの概要を次の表に示します。

表 6-11 XBRLDOMUserException クラスのコンストラクタの概要

項番	コンストラクタ	説明	XBRL のバージョン
1	XBRLDOMUserException(int code)	XBRLDOMUserException オブジェクトを構築します。	2.0 および 2.1
2	XBRLDOMUserException(int code, java.lang.String message)	XBRLDOMUserException オブジェクトを構築します。	2.0 および 2.1
3	XBRLDOMUserException(int code, java.lang.Throwable cause)	XBRLDOMUserException オブジェクトを構築します。	2.0 および 2.1
4	XBRLDOMUserException(int code, java.lang.String message, java.lang.Throwable cause)	XBRLDOMUserException オブジェクトを構築します。	2.0 および 2.1

(1) XBRLDOMUserException

```
public  XBRLDOMUserException(int code)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDOMUserException オブジェクトを構築します。

引数：

code - エラーコードを指定します。利用できる値の範囲は 90000 ～ 99999 です。

（2） XBRLDOMUserException

```
public  XBRLDOMUserException(int code,
                             java.lang.String message)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDOMUserException オブジェクトを構築します。

引数：

code - エラーコードを指定します。利用できる値の範囲は 90000 ～ 99999 です。

message - 詳細メッセージを指定します。

（3） XBRLDOMUserException

```
public  XBRLDOMUserException(int code,
                             java.lang.Throwable cause)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDOMUserException オブジェクトを構築します。引数 cause に null 以外を指定した場合，詳細メッセージには cause.toString() の戻り値が設定されます。

引数：

code - エラーコードを指定します。利用できる値の範囲は 90000 ～ 99999 です。

cause - エラーの原因となったスローできるオブジェクトを指定します。

（4） XBRLDOMUserException

```
public  XBRLDOMUserException(int code,
                             java.lang.String message,
                             java.lang.Throwable cause)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDOMUserException オブジェクトを構築します。

引数：

code - エラーコードを指定します。利用できる値の範囲は 90000 ～ 99999 です。

message - 詳細メッセージを指定します。
cause - エラーの原因となったスローできるオブジェクトを指定します。

6.5.6 XBRLDimensionException クラス

```
public class XBRLDimensionException
extends XBRLDOMException
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
XBRL Dimensions の構文解析エラーなど，発生することがあらかじめ予想される種類の例外通知のために，uCosminexus Business Reporting Processor がスローします。

フィールドの概要：
XBRLDimensionException クラスのフィールドの概要を次の表に示します。

表 6-12 XBRLDimensionException クラスのフィールドの概要

項番	説明	
1	static javax.xml.namespace.QName	XBRLDTE_DIMENSION_DEFAULT_SOURCE_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:DimensionDefaultSourceError」を保持します。
2	static javax.xml.namespace.QName	XBRLDTE_DIMENSION_DEFAULT_TARGET_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:DimensionDefaultTargetError」を保持します。
3	static javax.xml.namespace.QName	XBRLDTE_DIMENSION_DOMAIN_SOURCE_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:DimensionDomainSourceError」を保持します。
4	static javax.xml.namespace.QName	XBRLDTE_DIMENSION_DOMAIN_TARGET_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:DimensionDomainTargetError」を保持します。
5	static javax.xml.namespace.QName	XBRLDTE_DIMENSION_ELEMENT_IS_NOT_ABSTRACT_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:DimensionElementIsNotAbstractError」を保持します。
6	static javax.xml.namespace.QName	XBRLDTE_DOMAIN_MEMBER_SOURCE_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:DomainMemberSourceError」を保持します。
7	static javax.xml.namespace.QName	XBRLDTE_DOMAIN_MEMBER_TARGET_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:DomainMemberTargetError」を保持します。

項番	説明	
8	static javax.xml.namespace.QName	XBRLDTE_DRS_DIRECTED_CYCLE_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:DRSDirectedCycleError」を保持します。
9	static javax.xml.namespace.QName	XBRLDTE_HAS_HYPERCUBE_MISSING_CONTEXT_ELEMENT_ATTRIBUTE_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:HasHypercubeMissingContextElementAttributeError」を保持します。
10	static javax.xml.namespace.QName	XBRLDTE_HAS_HYPERCUBE_SOURCE_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:HasHypercubeSourceError」を保持します。
11	static javax.xml.namespace.QName	XBRLDTE_HAS_HYPERCUBE_TARGET_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:HasHypercubeTargetError」を保持します。
12	static javax.xml.namespace.QName	XBRLDTE_HYPERCUBE_DIMENSION_SOURCE_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:HypercubeDimensionSourceError」を保持します。
13	static javax.xml.namespace.QName	XBRLDTE_HYPERCUBE_DIMENSION_TARGET_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:HypercubeDimensionTargetError」を保持します。
14	static javax.xml.namespace.QName	XBRLDTE_HYPERCUBE_ELEMENT_IS_NOT_ABSTRACT_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:HypercubeElementIsNotAbstractError」を保持します。
15	static javax.xml.namespace.QName	XBRLDTE_OUT_OF_DTS_SCHEMA_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:OutOfDTSSchemaError」を保持します。
16	static javax.xml.namespace.QName	XBRLDTE_PRIMARY_ITEM_POLYMORPHISM_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:PrimaryItemPolymorphismError」を保持します。
17	static javax.xml.namespace.QName	XBRLDTE_TARGET_ROLE_NOT_RESOLVED_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:TargetRoleNotResolvedError" を保持します。
18	static javax.xml.namespace.QName	XBRLDTE_TOOMANY_DEFAULT_MEMBERS_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:TooManyDefaultMembersError」を保持します。
19	static javax.xml.namespace.QName	XBRLDTE_TYPED_DIMENSION_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:TypedDimensionError」を保持します。
20	static javax.xml.namespace.QName	XBRLDTE_TYPED_DIMENSION_URI_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:TypedDimensionURIError」を保持します。

項番	説明	
21	static javax.xml.namespace.QName	XBRLDTE_TYPED_DOMAIN_REF_ERROR この定数は，XBRL Dimensions で規定されているタクソノミーエラー「xbrldte:TypedDomainRefError」を保持します。
22	static javax.xml.namespace.QName	XBRLDIE_DEFAULT_VALUE_USED_IN_INSTANCE_ERROR この定数は，XBRL Dimensions で規定されているインスタンスエラー「xbrldie:DefaultValueUsedInInstanceError」を保持します。
23	static javax.xml.namespace.QName	XBRLDIE_EXPLICIT_MEMBER_NOT_EXPLICIT_DIMENSION_ERROR この定数は，XBRL Dimensions で規定されているインスタンスエラー「xbrldie:ExplicitMemberNotExplicitDimensionError」を保持します。
24	static javax.xml.namespace.QName	XBRLDIE_EXPLICIT_MEMBER_UNDEFINED_QNAME_ERROR この定数は，XBRL Dimensions で規定されているインスタンスエラー「xbrldie:ExplicitMemberUndefinedQNameError」を保持します。
25	static javax.xml.namespace.QName	XBRLDIE_PRIMARY_ITEM_DIMENSIONALLY_INVALID_ERROR この定数は，XBRL Dimensions で規定されているインスタンスエラー「xbrldie:PrimaryItemDimensionallyInvalidError」を保持します。
26	static javax.xml.namespace.QName	XBRLDIE_REPEATED_DIMENSION_IN_INSTANCE_ERROR この定数は，XBRL Dimensions で規定されているインスタンスエラー「xbrldie:RepeatedDimensionInInstanceError」を保持します。
27	static javax.xml.namespace.QName	XBRLDIE_TYPED_MEMBER_NOT_TYPED_DIMENSION_ERROR この定数は，XBRL Dimensions で規定されているインスタンスエラー「xbrldie:TypedMemberNotTypedDimensionError」を保持します。

メソッドの概要：

XBRLDimensionException クラスのメソッドの概要を次の表に示します。

表 6-13　XBRLDimensionException クラスのメソッドの概要

説明		XBRL のバージョン	XBRL Dimensions のバージョン
javax.xml.namespace.QName	getQName() XBRL Dimensions で規定されているエラーコードを表す QName を取得します。	2.1	1.0

(1) getQName

```
public javax.xml.namespace.QName getQName()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

XBRL Dimensions で規定されているエラーコードを表す QName を取得します。

戻り値：

XBRL Dimensions で規定されているエラーコードを表す QName を返します。
QName が設定されていない場合は null を返します。

6.5.7 XBRLDimensionUserException クラス

```
public class XBRLDimensionUserException
extends XBRLDimensionException
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

ユーザーアプリケーションで XBRLDimensionException を明示的にスローしたい場合に使用するクラスです。
XBRLDimensionUserException クラスは，XBRLDimensionException からの派生クラスです。
XBRLDimensionUserException クラスのコンストラクタではエラーコードを指定します。利用できるエラーコードの値の範囲は 90000 ～ 99999 です。ただし，指定したエラーコードが 90000 ～ 99999 の範囲内の値かどうかはチェックされません。このため，範囲外の値を指定してもエラーとはなりません。
また，XBRLDimensionUserException クラスのコンストラクタでは XBRL Dimensions で規定されているエラーコードを表す QName を指定します。利用できる QName は XBRLDimensionException クラスのフィールドに定義された値です。ただし，指定した QName が XBRLDimensionException クラスのフィールドと同じかどうはチェックされません。このため，XBRLDimensionException クラスのフィールド以外の値を指定してもエラーとはなりません。

コンストラクタの概要：

XBRLDimensionUserException クラスのコンストラクタの概要を次の表に示します。

表 6-14　XBRLDimensionUserException クラスのコンストラクタの概要

項番	説明	XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLDimensionUserException(int code, javax.xml.namespace.QName qname) XBRLDimensionUserException オブジェクトを構築します。	2.1	1.0
2	XBRLDimensionUserException(int code, java.lang.String message, javax.xml.namespace.QName qname) XBRLDimensionUserException オブジェクトを構築します。	2.1	1.0
3	XBRLDimensionUserException(int code, java.lang.Throwable cause, javax.xml.namespace.QName qname) XBRLDimensionUserException オブジェクトを構築します。	2.1	1.0
4	XBRLDimensionUserException(int code, java.lang.String message, java.lang.Throwable cause, javax.xml.namespace.QName qname) XBRLDimensionUserException オブジェクトを構築します。	2.1	1.0

(1) XBRLDimensionUserException

```
public XBRLDimensionUserException(int code,
                                  javax.xml.namespace.QName qname)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

XBRLDimensionUserException オブジェクトを構築します。

引数：

code - エラーコードを指定します。利用できる値の範囲は 90000 ～ 99999 です。

qname - XBRL Dimensions で規定されているエラーコードを表す QName を指定します。利用できる QName は XBRLDimensionException クラスのフィールドに定義された値です。

(2) XBRLDimensionUserException

```
public XBRLDimensionUserException(int code,
                                  java.lang.String message,
                                  javax.xml.namespace.QName qname)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

XBRLDimensionUserException オブジェクトを構築します。

引数：

code - エラーコードを指定します。利用できる値の範囲は 90000 ～ 99999 です。

message - 詳細メッセージを指定します。

qname - XBRL Dimensions で規定されているエラーコードを表す QName を指定します。利用できる QName は XBRLDimensionException クラスのフィールドに定義した値です。

（3） XBRLDimensionUserException

```
public XBRLDimensionUserException(int code,
                                  java.lang.Throwable cause,
                                  javax.xml.namespace.QName qname)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

XBRLDimensionUserException オブジェクトを構築します。

引数：

code - エラーコードを指定します。利用できる値の範囲は 90000 ～ 99999 です。

cause - エラーの原因となったスローできるオブジェクトを指定します。

qname - XBRL Dimensions で規定されているエラーコードを表す QName を指定します。利用できる QName は XBRLDimensionException クラスのフィールドに定義した値です。

（4） XBRLDimensionUserException

```
public XBRLDimensionUserException(int code,
                                  java.lang.String message,
                                  java.lang.Throwable cause,
                                  javax.xml.namespace.QName qname)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

XBRLDimensionUserException オブジェクトを構築します。

引数：

code - エラーコードを指定します。利用できる値の範囲は 90000 ～ 99999 です。

message - 詳細メッセージを指定します。

cause - エラーの原因となったスローできるオブジェクトを指定します。

qname - XBRL Dimensions で規定されているエラーコードを表す QName を指定します。利用できる QName は XBRLDimensionException クラスのフィールドに定義された値です。

6.6 com.hitachi.xbrl.instance パッケージ

com.hitachi.xbrl.instance パッケージは，インスタンス文書を操作するためのインターフェースを提供します。

インターフェースの概要：

com.hitachi.xbrl.instance パッケージのインターフェースを次の表に示します。

表 6-15 com.hitachi.xbrl.instance パッケージのインターフェース

項番	インターフェース	説明	XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLArcroleRef	インスタンス文書に記述されている <link:arcroleRef> 要素を表します。	2.1	該当しない
2	XBRLContext	コンテキスト要素を表します。	2.0 および 2.1	該当しない
3	XBRLContext2	コンテキスト要素を表します。	2.1	該当しない
4	XBRLEntity	エンティティ要素を表します。	2.0 および 2.1	該当しない
5	XBRLFootnote	脚注を表します。	2.0 および 2.1	該当しない
6	XBRLIdentifier	識別子要素を表します。	2.0 および 2.1	該当しない
7	XBRLInstanceDocument	インスタンス文書全体を表します。	2.0 および 2.1	該当しない
8	XBRLInstanceDocumentBuilder	インスタンス文書から XBRLInstanceDocument オブジェクトを取得するための API を定義します。	2.0 および 2.1	該当しない
9	XBRLInstanceDocumentBuilderFactory	XBRLInstanceDocumentBuilder オブジェクトを生成するための API を定義します。	2.0 および 2.1	該当しない
10	XBRLInstanceDocumentWriter	インスタンス文書をストリームに書き出します。	2.0 および 2.1	該当しない
11	XBRLInstanceDocumentWriterFactory	XBRLInstanceDocumentWriter オブジェクトを生成するための API を定義します。	2.0 および 2.1	該当しない
12	XBRLInstanceElement	インスタンス文書に記述されている要素を表します。	2.0 および 2.1	該当しない

項番	インターフェース	説明	XBRL のバージョン	XBRL Dimensions のバージョン
13	XBRLInstanceItem	アイテムを表します。	2.0 および 2.1	該当しない
14	XBRLInstanceNode	インスタンス文書を構成するノードを表します。	2.0 および 2.1	該当しない
15	XBRLInstanceTuple	インスタンス文書に記述されているタプルを表します。	2.0 および 2.1	該当しない
16	XBRLLinkbaseRef	インスタンス文書に記述されている <link:linkbaseRef> 要素を表します。	2.0 および 2.1	該当しない
17	XBRLMeasure	計量要素を表します。	2.0	該当しない
18	XBRLMeasureOrOperator	計量要素または演算子要素を表します。	2.0	該当しない
19	XBRLNonNumericContext	非数値コンテキスト要素を表します。	2.0	該当しない
20	XBRLNumericContext	数値コンテキスト要素を表します。	2.0	該当しない
21	XBRLOperator	演算子要素を表します。	2.0	該当しない
22	XBRLPeriod	時期要素を表します。	2.0 および 2.1	該当しない
23	XBRLRoleRef	インスタンス文書に記述されている <link:roleRef> 要素を表します。	2.1	該当しない
24	XBRLScenario	シナリオ要素を表します。	2.0 および 2.1	該当しない
25	XBRLSchemaRef	<link:schemaRef> 要素を表します。	2.1	該当しない
26	XBRLSegment	セグメント要素を表します。	2.0 および 2.1	該当しない
27	XBRLUnit	単位要素を表します。	2.0	該当しない
28	XBRLUnit2	単位要素を表します。	2.1	該当しない
29	XBRLDimensionMember	<xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素を表します。	2.1	1.0
30	XBRLExplicitMember	<xbrldi:explicitMember> 要素を表します。	2.1	1.0
31	XBRLTypedMember	<xbrldi:typedMember> 要素を表します。	2.1	1.0

6.6.1 XBRLArcroleRef インターフェース

```
public interface XBRLArcroleRef
extends XBRLInstanceNode
```

XBRL のバージョン：

2.1

説明：

インスタンス文書に記述されている <link:arcroleRef> 要素を表します。

メソッドの概要：

XBRLArcroleRef インターフェースのメソッドの概要を次の表に示します。

表 6-16 XBRLArcroleRef インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getHref() <link:arcroleRef> 要素の xlink:href 属性の値を取得します。	2.1
2	java.lang.String	getArcroleURI() <link:arcroleRef> 要素の arcroleURI 属性の値を取得します。	2.1
3	java.lang.String	getBase() <link:arcroleRef> 要素の xml:base 属性の値を取得します。	2.1
4	XBRLArcroleType	getArcroleType() <link:arcroleRef> 要素が指す <link:arcroleType> 要素を取得します。	2.1

(1) getHref

```
public java.lang.String getHref()
```

XBRL のバージョン：

2.1

説明：

<link:arcroleRef> 要素の xlink:href 属性の値を取得します。xlink:href 属性は，この <link:arcroleRef> 要素が指す特定の <link:arcroleType> 要素を参照します。

戻り値：

この <link:arcroleRef> 要素の xlink:href 属性の値を返します。

(2) getArcroleURI

```
public java.lang.String getArcroleURI()
```

XBRL のバージョン：

2.1

説明：

<link:arcroleRef> 要素の arcroleURI 属性の値を取得します。arcroleURI 属性の値は，この <link:arcroleRef> 要素が指す <link:arcroleType> 要素で定義されるカスタムアークロールの値と同じです。

戻り値：

この <link:arcroleRef> 要素の arcroleURI 属性の値を返します。

(3) getBase

```
public java.lang.String getBase()
```

XBRL のバージョン：

2.1

説明：

<link:arcroleRef> 要素の xml:base 属性の値を取得します。

戻り値：

この <link:arcroleRef> 要素の xml:base 属性の値を返します。xml:base 属性がない場合は null を返します。

(4) getArcroleType

```
public XBRLArcroleType getArcroleType()
```

XBRL のバージョン：

2.1

説明：

<link:arcroleRef> 要素が指す <link:arcroleType> 要素を取得します。
この getArcroleType() メソッドは，XBRLArcroleRef オブジェクトがアクティブである場合だけ呼び出せます。XBRLArcroleRef オブジェクトがアクティブでない場合に getArcroleType() メソッドを呼び出したときは，
XBRLDOMRuntimeException がスローされます。

戻り値：

この <link:arcroleRef> 要素が指す <link:arcroleType> 要素を表す
XBRLArcroleType オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- XBRLArcroleRef オブジェクトがアクティブでない場合にスローされます。［エラーコード：20122］

6.6.2 XBRLContext インターフェース

```
public interface XBRLContext
```

```
extends XBRLInstanceNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

コンテキスト要素を表します。コンテキスト要素には，複数の種類があります。各種のコンテキスト要素を表すほかのインターフェースは，このインターフェースを継承します。

- XBRL2.0 の場合：
  コンテキスト要素には，数値コンテキスト要素（XBRLNumericContext）と非数値コンテキスト要素（XBRLNonNumericContext）の 2 種類があります。コンテキスト要素が数値または非数値のどちらであるかを，XBRLContext.isNumeric() メソッドで調べられます。
- XBRL2.1 の場合：
  コンテキスト要素に数値または非数値の区別はありません。XBRLContext2 インターフェース一つだけで表現されます。

コンテキストの id 属性の値は，コンテキストを生成する際に自動的に採番されます。id 属性の値は，コンテキストを生成する際に自動的に採番されますが，setId() メソッドを使用して任意の値を設定することもできます。自動採番の詳細については，「3.3　ID 管理と採番」を参照してください。

メソッドの概要：

XBRLContext インターフェースのメソッドの概要を次の表に示します。

表 6-17　XBRLContext インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	boolean	isNumeric() コンテキスト要素が数値か非数値かを調べます。	2.0
2	java.lang.String	getId() コンテキスト要素の id 属性の値を取得します。	2.0 および 2.1
3	void	setId(java.lang.String id) コンテキスト要素の id 属性の値を設定します。	2.0 および 2.1
4	XBRLEntity	getEntity() エンティティ要素を取得します。	2.0 および 2.1
5	void	setEntity(XBRLEntity entity) エンティティ要素を設定します。	2.0 および 2.1
6	XBRLPeriod	getPeriod() 時期要素を取得します。	2.0 および 2.1

項番	説明		XBRL のバージョン
7	void	setPeriod(XBRLPeriod period) 時期要素を設定します。	2.0 および 2.1
8	XBRLScenario	getScenario() シナリオ要素を取得します。	2.0 および 2.1
9	void	setScenario(XBRLScenario scenario) シナリオ要素を設定します。	2.0 および 2.1
10	java.util.List<XBRLInstanceItem>	getItems() このコンテキスト要素を参照しているアイテムのリストを取得します。	2.0 および 2.1
11	java.util.List<XBRLInstanceItem>	getItems(java.lang.String namespaceURI, java.lang.String name) このコンテキスト要素を参照している，指定した名前を持つアイテムのリストを取得します。	2.0 および 2.1

(1) isNumeric

```
public boolean isNumeric()
```

XBRL のバージョン：

2.0

説明：

コンテキスト要素が数値か非数値かを調べます。

戻り値：

このコンテキストが数値コンテキストである場合は true，非数値コンテキストである場合は false を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。［エラーコード：20100］

(2) getId

```
public java.lang.String getId()
```

XBRL のバージョン：

2.0 および 2.1

説明：

コンテキスト要素の id 属性の値を取得します。

戻り値：

コンテキスト要素の id 属性の値を返します。

(3) setId

```
public void setId(java.lang.String id)
```

XBRL のバージョン：

2.0 および 2.1

説明：

コンテキスト要素の id 属性の値を設定します。

引数：

id - コンテキスト要素の id 属性の値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 id に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 id に指定した値が xsd:NCName 型でない場合にスローされます。［エラーコード：20108］
- このコンテキスト要素がインスタンス文書中に存在していて，かつ，引数 id で指定されたのと同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。［エラーコード：20028］

(4) getEntity

```
public XBRLEntity getEntity()
```

XBRL のバージョン：

2.0 および 2.1

説明：

エンティティ要素を取得します。

戻り値：

エンティティ要素を表す XBRLEntity オブジェクトを返します。

(5) setEntity

```
public void setEntity(XBRLEntity entity)
```

XBRL のバージョン：

2.0 および 2.1

説明：

エンティティ要素を設定します。
この XBRLContext オブジェクトにすでに設定されている XBRLEntity オブジェクトは，親がいない状態となり，getParentNode() メソッドが null を返します。

引数：

entity - エンティティ要素を表す XBRLEntity オブジェクトを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 entity に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 entity に指定した XBRLEntity オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLContext オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 entity に指定した XBRLEntity オブジェクトが，すでにほかの XBRLContext オブジェクトに設定されている場合にスローされます。［エラーコード：20005］

（6） getPeriod

```
public XBRLPeriod getPeriod()
```

XBRL のバージョン：

2.0 および 2.1

説明：

時期要素を取得します。

戻り値：

時期要素を表す XBRLPeriod オブジェクトを返します。

（7） setPeriod

```
public void setPeriod(XBRLPeriod period)
```

XBRL のバージョン：

2.0 および 2.1

説明：

時期要素を設定します。

この XBRLContext オブジェクトにすでに設定されている XBRLPeriod オブジェクトは，親がいない状態となり，getParentNode() メソッドが null を返します。

XBRL2.1 の場合：設定できる時期要素の内容に関する制限

引数 period に設定できる時期要素には，次の表に示す制限があります。

表 6-18　設定できる時期要素の内容に関する制限

項番	条件	制限
1	このコンテキスト要素に設定されている既存の時期要素の内容の意味が XBRLPeriod.INSTANT である場合	引数 period に指定する時期要素の内容の意味は，XBRLPeriod.INSTANT である必要があります。

項番	条件	制限
2	このコンテキスト要素に設定されている既存の時期要素の内容の意味が XBRLPeriod.FOREVER か XBRLPeriod.STARTDATE_ENDDATE である場合	引数 period に指定する時期要素の内容の意味は XBRLPeriod.FOREVER か XBRLPeriod.STARTDATE_ENDDATE である必要があります。

XBRL2.1 では，アイテムの要素宣言の xbrli:periodType 属性に "instant" が指定された場合は，インスタンス文書でそのアイテムが参照するコンテキスト要素の時期要素の内容が XBRLPeriod.INSTANT である必要があります。また，xbrli:periodType 属性に "duration" が指定された場合は，時期要素の内容が XBRLPeriod.FOREVER か XBRLPeriod.STARTDATE_ENDDATE である必要があるという制約があります。

引数：

period - 時期要素を表す XBRLPeriod オブジェクトを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 period に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 period に指定した XBRLPeriod オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLContext オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 period に指定した XBRLPeriod オブジェクトが，すでにほかの XBRLContext オブジェクトに設定されている場合にスローされます。［エラーコード：20005］
- XBRL2.1 の場合：引数 period で指定した XBRLPeriod オブジェクトが，上記の制約に違反している場合にスローされます。［エラーコード：20141］

（8） getScenario

```
public XBRLScenario getScenario()
```

XBRL のバージョン：

2.0 および 2.1

説明：

シナリオ要素を取得します。

戻り値：

シナリオ要素を表す XBRLScenario オブジェクトを返します。シナリオ要素がない場合は null を返します。

(9) setScenario

```
public void setScenario(XBRLScenario scenario)
```

XBRL のバージョン：

2.0 および 2.1

説明：

シナリオ要素を設定します。

この XBRLContext オブジェクトに XBRLScenario オブジェクトが設定されている場合，そのオブジェクトは親がいない状態となり，getParentNode() メソッドが null を返します。

引数：

scenario - シナリオ要素を表す XBRLScenario オブジェクトを指定します。シナリオ要素を削除する場合は null を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 scenario に指定した XBRLScenario オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLContext オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 scenario に指定した XBRLScenario オブジェクトが，すでにほかの XBRLContext オブジェクトに設定されている場合にスローされます。［エラーコード：20005］

(10) getItems

```
public java.util.List<XBRLInstanceItem> getItems()
```

XBRL のバージョン：

2.0 および 2.1

説明：

このコンテキスト要素を参照しているアイテムのリストを取得します。

このコンテキストを参照しているアイテムには，インスタンス文書にない（アクティブでない）ノードも含まれます。そのノードは XBRLInstanceNode.isActive() が false を返します。

このメソッドが返すアイテムのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれるアイテムに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

このコンテキストを参照しているアイテムをすべて含む，XBRLInstanceItem オブ

ジェクトのリストを返します。このコンテキストを参照しているアイテムがない場合は，要素数が 0 のリストを返します。

(11) getItems

```
public java.util.List<XBRLInstanceItem> getItems(java.lang.String
namespaceURI,
                                java.lang.String name)
```

XBRL のバージョン：

2.0 および 2.1

説明：

このコンテキスト要素を参照している，指定した名前を持つアイテムのリストを取得します。

このコンテキストを参照しているアイテムには，インスタンス文書にない（アクティブでない）ノードも含まれます。そのノードは XBRLInstanceNode.isActive() が false を返します。

このメソッドが返すアイテムのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれるアイテムに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

namespaceURI - アイテムの名前空間 URI を指定します。

name - アイテムの名前を指定します。

戻り値：

引数 namespaceURI で指定した名前空間 URI に属し，引数 name で指定した名前を持つアイテムを表す XBRLInstanceItem オブジェクトのリストを返します。指定したアイテムがない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 namespaceURI または name に null を指定した場合にスローされます。［エラーコード：20002］

6.6.3 XBRLContext2 インターフェース

```
public interface XBRLContext2
extends XBRLContext
```

XBRL のバージョン：

2.1

説明：

コンテキスト要素を表します。

6.6.4 XBRLEntity インターフェース

```
public interface XBRLEntity
extends XBRLInstanceNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

エンティティ要素を表します。

メソッドの概要：

XBRLEntity インターフェースのメソッドの概要を次の表に示します。

表 6-19 XBRLEntity インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLIdentifier	getIdentifier() 識別子要素を取得します。	2.0 および 2.1
2	void	setIdentifier(XBRLIdentifier identifier) 識別子要素を設定します。	2.0 および 2.1
3	XBRLSegment	getSegment() セグメント要素を取得します。	2.0 および 2.1
4	void	setSegment(XBRLSegment segment) セグメント要素を設定します。	2.0 および 2.1

(1) getIdentifier

```
public XBRLIdentifier getIdentifier()
```

XBRL のバージョン：

2.0 および 2.1

説明：

識別子要素を取得します。

戻り値：

識別子要素を表す XBRLIdentifier オブジェクトを返します。

(2) setIdentifier

```
public void setIdentifier(XBRLIdentifier identifier)
```

XBRL のバージョン：

2.0 および 2.1

説明：

識別子要素を設定します。

この XBRLEntity オブジェクトにすでに設定されている XBRLIdentifier オブジェクトは，親がいない状態となり，getParentNode() メソッドが null を返します。

引数：

identifier - 識別子要素を表す XBRLIdentifier オブジェクトを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 identifier に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 identifier に指定した XBRLIdentifier オブジェクトを所有している XBRLInstanceDocument オブジェクトがこの XBRLEntity オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 identifier に指定した XBRLIdentifier オブジェクトがすでにほかの XBRLEntity オブジェクトに設定されている場合にスローされます。［エラーコード：20005］

（3） getSegment

```
public XBRLSegment getSegment()
```

XBRL のバージョン：

2.0 および 2.1

説明：

セグメント要素を取得します。

戻り値：

セグメント要素を表す XBRLSegment オブジェクトを返します。セグメント要素がない場合は null を返します。

（4） setSegment

```
public void setSegment(XBRLSegment segment)
```

XBRL のバージョン：

2.0 および 2.1

説明：

セグメント要素を設定します。

この XBRLEntity オブジェクトに XBRLSegment オブジェクトが設定されている場合，そのオブジェクトは親がいない状態となり，getParentNode() メソッドが null を返します。

引数：

segment - セグメント要素を表す XBRLSegment オブジェクトを指定します。セグメント要素を削除する場合は null を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 segment に指定した XBRLSegment オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLEntity オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 segment に指定した XBRLSegment オブジェクトが，すでにほかの XBRLEntity オブジェクトに設定されている場合にスローされます。［エラーコード：20005］

6.6.5 XBRLFootnote インターフェース

```
public interface XBRLFootnote
extends XBRLInstanceNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

脚注を表します。

メソッドの概要：

XBRLFootnote インターフェースのメソッドの概要を次の表に示します。

表 6-20 XBRLFootnote インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getFootnoteLinkRole() 脚注が属する脚注リンクのロールを取得します。	2.1
2	java.lang.String	getLang() 脚注の言語を取得します。	2.0 および 2.1
3	void	setLang(java.lang.String lang) 脚注の言語を設定します。	2.0 および 2.1
4	java.lang.String	getRole() 脚注のロールを取得します。	2.0 および 2.1
5	void	setRole(java.lang.String role) 脚注のロールを設定します。	2.0 および 2.1
6	java.lang.String	getTitle() 脚注のタイトルを取得します。	2.0 および 2.1
7	void	setTitle(java.lang.String title) 脚注のタイトルを設定します。	2.0 および 2.1
8	java.lang.String	getId() 脚注の id 属性を取得します。	2.1
9	void	setId(java.lang.String id) 脚注の id 属性を設定します。	2.1

項番	説明		XBRL のバージョン
10	java.lang.String	getValue() 脚注の内容を表す文字列を取得します。	2.0 および 2.1
11	void	setValue(java.lang.String value) 脚注の内容を表す文字列を設定します。	2.0 および 2.1
12	org.w3c.dom.Node	getContainerNode() 脚注の内容を保持する org.w3c.dom.Node オブジェクトを取得します。	2.0 および 2.1
13	java.util.List<XBRLInstanceElement>	getElements() 脚注と関連づけられている要素のリストを取得します。	2.0 および 2.1
14	java.util.List<XBRLInstanceElement>	getElements(java.lang.String namespaceURI, java.lang.String name) この脚注と関連づけられている，指定された名前を持つ要素のリストを取得します。	2.0 および 2.1

（1） getFootnoteLinkRole

```
public java.lang.String getFootnoteLinkRole()
```

XBRL のバージョン：

2.1

説明：

脚注が属する脚注リンクのロールを取得します。

戻り値：

この脚注が属する脚注リンクのロールを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

（2） getLang

```
public java.lang.String getLang()
```

XBRL のバージョン：

2.0 および 2.1

説明：

脚注の言語を取得します。

戻り値：
この脚注の言語を表す xml:lang 属性の値を返します。

(3) setLang

```
public void setLang(java.lang.String lang)
```

XBRL のバージョン：
2.0 および 2.1

説明：
脚注の言語を設定します。

引数：
lang - この脚注の言語を表す xml:lang 属性の値を指定します。

例外：
XBRLDOMRuntimeException -
- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(4) getRole

```
public java.lang.String getRole()
```

XBRL のバージョン：
2.0 および 2.1

説明：
脚注のロール（役割）を取得します。脚注のロールは，<link:footnote> 要素の xlink:role 属性の値です。

戻り値：
この脚注のロールを表す xlink:role 属性の値を返します。xlink:role 属性の値がない場合は null を返します。

(5) setRole

```
public void setRole(java.lang.String role)
```

XBRL のバージョン：
2.0 および 2.1

説明：
脚注のロール（役割）を設定します。脚注のロールは，<link:footnote> 要素の xlink:role 属性の値です。

引数：
role - この脚注のロールを指定します。ロールを削除する場合は null を指定します。

例外：
XBRLDOMRuntimeException -

- 引数 role に指定した値が xsd:anyURI 型でない場合にスローされます。［エラーコード：20108］
- 引数 role に指定した値が絶対 URI でない場合にスローされます。［エラーコード：20109］
- XBRL2.1 の場合：引数 role に指定したロールが XBRL 仕様で規定された標準値で，かつ，脚注で利用できる値ではない場合にスローされます。［エラーコード：20115］
- XBRL2.1 の場合：引数 role に指定したロールがカスタムロールで，かつ，そのカスタムロールの定義を指す <link:roleRef> 要素がインスタンス文書に記述されていない場合にスローされます。［エラーコード：20117］
- XBRL2.1 の場合：引数 role に指定したロールがカスタムロールで，かつ，<link:roleType> 要素でそのカスタムロールが脚注で利用できるように定義されていない場合にスローされます。［エラーコード：20119］

（6） getTitle

```
public java.lang.String getTitle()
```

XBRL のバージョン：

2.0 および 2.1

説明：

脚注のタイトルを取得します。脚注のタイトルは，<link:footnote> 要素の xlink:title 属性の値です。

戻り値：

この脚注のタイトルを表す xlink:title 属性の値を返します。xlink:title 属性の値がない場合は null を返します。

（7） setTitle

```
public void setTitle(java.lang.String title)
```

XBRL のバージョン：

2.0 および 2.1

説明：

脚注のタイトルを設定します。脚注のタイトルは，<link:footnote> 要素の xlink:title 属性の値です。

引数：

title - この脚注のタイトルを指定します。タイトルを削除する場合は null を指定します。

（8） getId

```
public java.lang.String getId()
```

XBRL のバージョン：
2.1

説明：
脚注の id 属性を取得します。

戻り値：
脚注の id 属性の値を返します。id 属性がない場合は null を返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

(9) setId

```
public void setId(java.lang.String id)
```

XBRL のバージョン：
2.1

説明：
脚注の id 属性を設定します。

引数：
id - この脚注の id 属性の値を指定します。id 属性を削除する場合は，null を指定します。

例外：
XBRLDOMRuntimeException -
- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 id に指定した値が xsd:NCName 型でない場合にスローされます。［エラーコード：20108］
- この脚注がインスタンス文書中に記述されていて，かつ，引数 id で指定されたのと同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに記述されている場合にスローされます。［エラーコード：20028］

(10) getValue

```
public java.lang.String getValue()
```

XBRL のバージョン：
2.0 および 2.1

説明：
脚注の内容を表す文字列を取得します。
脚注の内容に XHTML のタグが含まれている場合，このメソッドは null を返しま

す。脚注の内容に含まれる XHTML のタグを参照したい場合は，
getContainerNode() メソッドを使用してください。

戻り値：
脚注の内容を表す文字列を返します。脚注の内容に XHTML のタグが含まれている場合は null を返します。

（11）setValue

```
public void setValue(java.lang.String value)
```

XBRL のバージョン：
2.0 および 2.1

説明：
脚注の内容を表す文字列を設定します。
引数 value に，XHTML のタグを含む文字列を指定しても，脚注の内容には XHTML のタグは生成されません。脚注に XHTML のタグを含む内容を設定したい場合は，getContainerNode() メソッドで org.w3c.dom.Node オブジェクトを取得し，その子ノードおよび子孫ノードとして XHTML のタグを表す org.w3c.dom.Node オブジェクトを追加します。

引数：
value - この脚注の内容を表す文字列を指定します。

例外：
XBRLDOMRuntimeException -
- 引数 value に null を指定した場合にスローされます。［エラーコード：20002］

（12）getContainerNode

```
public org.w3c.dom.Node getContainerNode()
```

XBRL のバージョン：
2.0 および 2.1

説明：
脚注の内容を保持する org.w3c.dom.Node オブジェクトを取得します。
このメソッドが返す org.w3c.dom.Node オブジェクトの子孫ノード以外のノードに対して，更新を伴う操作をしないでください。このメソッドが返す org.w3c.dom.Node オブジェクトの子孫ノードだけを更新できます。

戻り値：
この脚注の内容を保持する org.w3c.dom.Node オブジェクトを返します。

（13）getElements

```
public java.util.List<XBRLInstanceElement> getElements()
```

XBRL のバージョン：

2.0 および 2.1

説明：

脚注と関連づけられている要素のリストを取得します。要素は，アイテムまたはタプルです。
この脚注と関連づけられている要素には，インスタンス文書にない，XBRLInstanceNode.isActive() が false を返すノードも含まれます。
このメソッドが返す要素のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる要素に増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

この脚注と関連づけられている要素をすべて含む，XBRLInstanceElement オブジェクトのリストを返します。この脚注と関連づけられている要素がない場合は，要素数が 0 のリストを返します。

（14）getElements

```
public java.util.List<XBRLInstanceElement>
getElements(java.lang.String namespaceURI,
                                  java.lang.String name)
```

XBRL のバージョン：

2.0 および 2.1

説明：

この脚注と関連づけられている，指定された名前を持つ要素のリストを取得します。要素は，アイテムまたはタプルです。
この脚注と関連づけられている要素には，インスタンス文書にない，XBRLInstanceNode.isActive() が false を返すノードも含まれます。
このメソッドが返す要素のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる要素に増減があった場合，その変更は直ちに反映されます。このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

namespaceURI - 要素の名前空間 URI を指定します。
name - 要素の名前を指定します。

戻り値：

引数 namespaceURI で指定した名前空間 URI に属し，引数 name で指定した名前を持つ要素を表す XBRLInstanceElement オブジェクトのリストを返します。指定

した要素がない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 namespaceURI または name に null を指定した場合にスローされます。［エラーコード：20002］

6.6.6 XBRLIdentifier インターフェース

```
public interface XBRLIdentifier
extends XBRLInstanceNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

識別子要素を表します。

メソッドの概要：

XBRLIdentifier インターフェースのメソッドの概要を次の表に示します。

表 6-21 XBRLIdentifier インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getScheme() 識別子要素の scheme 属性の値を取得します。	2.0 および 2.1
2	void	setScheme(java.lang.String namespaceURI) 識別子要素の scheme 属性の値を設定します。	2.0 および 2.1
3	java.lang.String	getValue() 識別子要素の内容を取得します。	2.0 および 2.1
4	void	setValue(java.lang.String value) 識別子要素の内容を設定します。	2.0 および 2.1

（1） getScheme

```
public java.lang.String getScheme()
```

XBRL のバージョン：

2.0 および 2.1

説明：

識別子要素の scheme 属性の値を取得します。scheme 属性は，この XBRLIdentifier オブジェクトによって表現される識別子が属するスキーマの名前空間 URI を表します。

戻り値：

スキーマの名前空間 URI を表す scheme 属性の値を返します。

(2) setScheme

```
public void setScheme(java.lang.String namespaceURI)
```

XBRL のバージョン：

2.0 および 2.1

説明：

識別子要素の scheme 属性の値を設定します。scheme 属性は，この XBRLIdentifier オブジェクトによって表現される識別子が属するスキーマの名前空間 URI を表します。

引数：

namespaceURI - スキーマの名前空間 URI を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 namespaceURI に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 namespaceURI に指定した値が xsd:anyURI 型でない場合にスローされます。[エラーコード：20108]
- XBRL2.1 の場合：引数 namespaceURI に空文字列（""）を指定した場合にスローされます。[エラーコード：20110]

(3) getValue

```
public java.lang.String getValue()
```

XBRL のバージョン：

2.0 および 2.1

説明：

識別子要素の内容を取得します。

戻り値：

識別子を返します。

(4) setValue

```
public void setValue(java.lang.String value)
```

XBRL のバージョン：

2.0 および 2.1

説明：

識別子要素の内容を設定します。

引数：

value - 識別子を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 value に null を指定した場合にスローされます。［エラーコード：20002］

6.6.7 XBRLInstanceDocument インターフェース

```
public interface XBRLInstanceDocument
extends XBRLInstanceNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書全体を表します。

既存のインスタンス文書を読み込んで XBRLInstanceDocument オブジェクトを作成する場合は，XBRLInstanceDocumentBuilder の parse() メソッドを使用します。また，インスタンス文書を新規に作成する場合は，XBRLInstanceDocumentBuilder の newInstanceDocument() メソッドを使用して XBRLInstanceDocument オブジェクトを生成します。

名前空間宣言

XBRLInstanceDocument オブジェクトでは，XBRLNamespaceDeclaration オブジェクトを使用して，インスタンス文書に記述される名前空間宣言を管理します。XBRLInstanceDocument オブジェクトに設定された名前空間宣言は，getNamespaceDeclarations() などのメソッドを使用して取得できます。

既存のインスタンス文書を編集する場合の名前空間宣言

既存のインスタンス文書から XBRLInstanceDocument オブジェクトを生成した場合は，XBRLInstanceDocument オブジェクトは，そのインスタンス文書に記述されていた名前空間宣言のリストを保持します。

このとき，同一の名前空間 URI に対して複数の名前空間宣言が記述されていた場合は，そのうちの一つだけが保持され，そのほかの名前空間宣言は無視されます。また，同一の名前空間接頭辞に複数の名前空間 URI が関連づけられていた場合は，名前空間接頭辞を自動的に生成することで，複数の名前空間 URI に対して同一の名前空間接頭辞が重複して関連づけられることがないようにします。

既存のインスタンス文書から XBRLInstanceDocument オブジェクトを生成したあとに，ユーザーアプリケーションが特定の編集操作を実行すると，XBRLInstanceDocument オブジェクトに設定されている名前空間宣言に合わせてインスタンス文書中の XML 要素やグローバル属性の名前空間接頭辞を変更する処理が実行されます。これを**名前空間の正規化**と呼びます。名前空間の正規化では，ルート要素以外の XML 要素から名前空間宣言を削除します。既存のインスタンス文書からの名前空間宣言の読み込み，および名前空間の正規化に関する詳細については，「3.4　名前空間の正規化」を参照してください。

新規にインスタンス文書を作成する場合の名前空間宣言

XBRLInstanceDocumentBuilder の newInstanceDocument() メソッドを使用して XBRLInstanceDocument オブジェクトを生成した場合は，初期状態で標準的な名前空間の宣言が設定されます。初期状態で設定されている名前空間宣言を次の表に示します。

表 6-22　インスタンス文書の新規作成時に初期状態で設定される名前空間宣言（XBRL2.0 の場合）

項番	名前空間接頭辞	名前空間 URI
1	xbrli	http://www.xbrl.org/2001/instance
2	link	http://www.xbrl.org/2001/XLink/xbrllinkbase
3	xlink	http://www.w3.org/1999/xlink
4	xsi	http://www.w3.org/2001/XMLSchema-instance
5	xhtml	http://www.w3.org/1999/xhtml

表 6-23　インスタンス文書の新規作成時に初期状態で設定される名前空間宣言（XBRL2.1 の場合）

項番	名前空間接頭辞	名前空間 URI
1	なし	空文字列（""）
2	xbrli	http://www.xbrl.org/2003/instance
3	link	http://www.xbrl.org/2003/linkbase
4	iso4217	http://www.xbrl.org/2003/iso4217
5	xlink	http://www.w3.org/1999/xlink
6	xsi	http://www.w3.org/2001/XMLSchema-instance
7	xhtml	http://www.w3.org/1999/xhtml
8	xbrldi ※	http://xbrl.org/2006/xbrldi

注※　XBRL Dimensions1.0 に対応している場合だけ存在します。

名前空間宣言の追加および接頭辞の変更は，addNamespaceDeclaration() メソッドを使用します。次の操作をした際に，XBRLNamespaceDeclaration オブジェクトを使用して名前空間宣言を設定していないとエラーが発生します。このため，インスタンス文書で使用するすべての名前空間について，addNamespaceDeclaration() メソッドを使用して，必ず名前空間宣言を追加してください。

- XBRLInstanceDocument の createItem() などのメソッドを使用して XML 要素に対応するノードを生成する際に，その XML 要素が属する名前空間に対応する名前空間宣言が設定されていない場合にエラーが発生します。

- XBRLInstanceItem の setValue() メソッドまたは setString() メソッドを使用して，xsd:QName 型または xsd:NOTATION 型のアイテムに値を設定する際に，その xsd:QName 型データまたは xsd:NOTATION 型データの名前空間に対応する名前空間宣言が設定されていない場合にエラーが発生します。
- XBRLInstanceDocument の addFootnote() メソッドや XBRLInstanceElement の addFootnote() メソッドなどで脚注を操作する際に，脚注リンクを生成するために必要な名前空間 URI（http://www.xbrl.org/2001/XLink/xbrllinkbase および http://www.w3.org/1999/xlink）の名前空間宣言が設定されていない場合にエラーが発生します。

XBRL2.1 の場合：名前空間に属さない要素や修飾名を使用する場合の注意事項

インスタンス文書を編集した結果，次の三つの状態がすべて重なってしまった場合，インスタンス文書をストリームへ出力してから再度読み込むと，名前空間に属していなかった要素や修飾名がデフォルトの名前空間に属するように変更されてしまうおそれがあります。

- 名前空間に属さない要素や修飾名が，インスタンス文書中に存在する
- インスタンス文書のルート要素が，XML 文書のルート要素ではない
- インスタンス文書のルート要素の祖先要素に，デフォルトの名前空間の宣言が存在する

このような問題を回避するために，XBRLInstanceDocument オブジェクト生成時には，デフォルトの名前空間を解除する宣言（xmlns=""）が XBRLInstanceDocument オブジェクトに追加されます。デフォルトの名前空間を解除する宣言が追加されていると，祖先要素で宣言されているデフォルトの名前空間はインスタンス文書中では無効となるため，名前空間に属さない要素や修飾名は再度読み込んだ際にも名前空間に属さないままとなり，前述の問題は発生しません。しかし，デフォルトの名前空間を解除する宣言を明示的に削除した場合や，既存のインスタンス文書でデフォルトの名前空間が宣言されていた場合は，デフォルトの名前空間を解除する宣言が追加されていない状態となるため，前述の問題が顕在化するおそれがあります。
このため，デフォルトの名前空間を解除する宣言を削除することは推奨しません。また，名前空間に属さない要素や修飾名をインスタンス文書中で使用する場合は，デフォルトの名前空間を解除する宣言を XBRLInstanceDocument オブジェクトへ追加しておくことを推奨します。
デフォルトの名前空間を解除する宣言は，次に示すコードによって XBRLInstanceDocument オブジェクトへ追加できます。

```
idoc.addNamespaceDeclaration(new XBRLNamespaceDeclaration(null, ""));
```

xsi:schemaLocation 属性

XBRLInstanceDocument オブジェクトでは，XBRLSchemaLocation オブジェクトを使用してインスタンス文書のルート要素の xsi:schemaLocation 属性の内容を管理します。

既存のインスタンス文書を編集する場合の xsi:schemaLocation 属性

既存のインスタンス文書から XBRLInstanceDocument オブジェクトを生成した場合，XBRLInstanceDocument オブジェクトは，そのインスタンス文書のルート要素の xsi:schemaLocation 属性を読み取り，名前空間 URI とスキーマ文書の位置を示す URI の組のリストを保持します。

インスタンス文書を新規に作成する場合の xsi:schemaLocation 属性

XBRLInstanceDocumentBuilder.newInstanceDocument(XBRLTaxonomyDocument) などのメソッドを使用して XBRLInstanceDocument オブジェクトを生成した場合には，ルート要素の xsi:schemaLocation 属性の初期値としてタクソノミー文書に含まれる各タクソノミー本体の対象名前空間 URI と位置を示す絶対 URI の組が設定されます。xsi:schemaLocation 属性の内容は，addSchemaLocation() メソッドや removeSchemaLocation() メソッドを使用して内容の追加・変更・削除ができるため，必要な場合にはこれらのメソッドを使用して xsi:schemaLocation 属性の内容を適切に設定してください。
また，XBRL2.1 の場合，タクソノミー文書に対象名前空間 URI が同じであるタクソノミー本体が複数含まれる場合があります。
XBRLInstanceDocumentBuilder.newInstanceDocument(XBRLTaxonomyDocument) などのメソッドを使用して XBRLInstanceDocument オブジェクトを生成した際に，複数のタクソノミー本体で対象名前空間 URI が重複している場合は，タクソノミー文書の解析で先に解析されたタクソノミー本体が優先され，ほかのタクソノミー本体は無視されます。
なお，uCosminexus Business Reporting Processor のバージョンが 05-00 以降の場合に，XBRLInstanceDocumentBuilder インターフェースの isCreatingSchemaLocations() メソッドが返す値が true のときは，ルート要素の xsi:schemaLocation 属性の初期値として，タクソノミー文書に含まれる各タクソノミー本体の対象名前空間 URI と位置を示す絶対 URI の組が設定されます。isCreatingSchemaLocations() メソッドが返す値が false のときは，ルート要素の xsi:schemaLocation 属性の初期値として，タクソノミー文書に含まれる各タクソノミー本体の対象名前空間 URI と位置を示す絶対 URI の組が設定されません。

メソッドの概要：

XBRLInstanceDocument インターフェースのメソッドの概要を次の表に示します。

表 6-24　XBRLInstanceDocument インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLDocumentManager	getOwnerDocumentManager() インスタンス文書を所有する XBRLDocumentManager オブジェクトを取得します。	2.0 および 2.1	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
2	int	getXBRLVersion() インスタンス文書が準拠している XBRL のバージョンを取得します。	2.0 および 2.1	該当しない
3	PTVInstance	getPTVInstance() インスタンス文書の解析時に生成された PTVI を取得します。	2.1	該当しない
4	XBRLIdManager	getIdManager() インスタンス文書の ID を管理する XBRLIdManager オブジェクトを取得します。	2.1	該当しない
5	XBRLTaxonomyDocument	getTaxonomyDocument() インスタンス文書が参照するタクソノミー文書を取得します。	2.0 および 2.1	該当しない
6	org.w3c.dom.Element	getInstanceRoot() インスタンス文書のルート要素を表す org.w3c.dom.Element オブジェクトを取得します。	2.0 および 2.1	該当しない
7	java.util.List<XBRLNamespaceDeclaration>	getNamespaceDeclarations() インスタンス文書に記述されている名前空間宣言のリストを取得します。	2.0 および 2.1	該当しない
8	XBRLNamespaceDeclaration	getNamespaceDeclarationByURI(java.lang.String uri) 指定した名前空間 URI の名前空間宣言を取得します。	2.0 および 2.1	該当しない
9	XBRLNamespaceDeclaration	getNamespaceDeclarationByPrefix(java.lang.String prefix) 指定した名前空間接頭辞の名前空間宣言を取得します。	2.0 および 2.1	該当しない
10	XBRLNamespaceDeclaration	addNamespaceDeclaration(XBRLNamespaceDeclaration newNamespaceDeclaration) インスタンス文書のルート要素に名前空間宣言を追加します。	2.0 および 2.1	該当しない
11	XBRLNamespaceDeclaration	removeNamespaceDeclaration(XBRLNamespaceDeclaration oldNamespaceDeclaration) インスタンス文書から名前空間宣言を削除します。	2.0 および 2.1	該当しない
12	java.util.List<XBRLSchemaLocation>	getSchemaLocations() インスタンス文書のルート要素の xsi:schemaLocation 属性に記述されている名前空間 URI と，スキーマ文書の位置を示す URI の組を表すオブジェクトのリストを取得します。	2.0 および 2.1	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
13	XBRLSchemaLocation	getSchemaLocation(java.lang.String namespaceURI) インスタンス文書のルート要素の xsi:schemaLocation 属性に記述されている名前空間 URI と，スキーマ文書の位置を示す URI の組を表すオブジェクトのリストの中から，指定した名前空間 URI のオブジェクトを取得します。	2.0 および 2.1	該当しない
14	XBRLSchemaLocation	addSchemaLocation(XBRLSchemaLocation newSchemaLocation) インスタンス文書のルート要素の xsi:schemaLocation 属性に，名前空間 URI とスキーマ文書の位置を示す URI の組を追加します。	2.0 および 2.1	該当しない
15	XBRLSchemaLocation	removeSchemaLocation(XBRLSchemaLocation oldSchemaLocation) インスタンス文書のルート要素の xsi:schemaLocation 属性から，指定された名前空間 URI とスキーマ文書の位置を示す URI の組を削除します。	2.0 および 2.1	該当しない
16	java.lang.String	getBase() インスタンス文書のルート要素の xml:base 属性の値を取得します。	2.1	該当しない
17	void	setBase(java.lang.String base) インスタンス文書のルート要素の xml:base 属性の値を設定します。	2.1	該当しない
18	java.lang.String	getId() インスタンス文書のルート要素の id 属性の値を取得します。	2.1	該当しない
19	void	setId(java.lang.String id) インスタンス文書のルート要素の id 属性の値を設定します。	2.1	該当しない
20	java.util.List <XBRLSchemaRef>	getSchemaRefs() インスタンス文書に記述されている <link:schemaRef> 要素のリストを取得します。	2.1	該当しない
21	XBRLSchemaRef	addSchemaRef(XBRLSchemaRef newSchemaRef) インスタンス文書に <link:schemaRef> 要素を追加します。	2.1	該当しない
22	XBRLSchemaRef	createSchemaRef(java.lang.String href, java.lang.String base) <link:schemaRef> 要素を表す XBRLSchemaRef オブジェクトを新規に生成します。	2.1	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
23	java.util.List<XBRLLinkbaseRef>	getLinkbaseRefs() インスタンス文書に記述されている <link:linkbaseRef> 要素のリストを取得します。	2.0 および 2.1	該当しない
24	java.util.List<XBRLLinkbaseRef>	getLinkbaseRefs(java.lang.String role) インスタンス文書に記述されている <link:linkbaseRef> 要素のうち，引数 role と同じ値の xlink:role 属性を持つ <link:linkbaseRef> 要素のリストを取得します。	2.0 および 2.1	該当しない
25	XBRLLinkbaseRef	addLinkbaseRef(XBRLLinkbaseRef newLinkbaseRef) インスタンス文書に <link:linkbaseRef> 要素を追加します。	2.0 および 2.1	該当しない
26	XBRLLinkbaseRef	removeLinkbaseRef(XBRLLinkbaseRef oldLinkbaseRef) インスタンス文書に記述されている <link:linkbaseRef> 要素を削除します。	2.0	該当しない
27	XBRLLinkbaseRef	createLinkbaseRef(java.lang.String role, java.lang.String href) <link:linkbaseRef> 要素を表す XBRLLinkbaseRef オブジェクトを新規に生成します。	2.0 および 2.1	該当しない
28	XBRLLinkbaseRef	createLinkbaseRef(java.lang.String role, java.lang.String href, java.lang.String base) <link:linkbaseRef> 要素を表す XBRLLinkbaseRef オブジェクトを新規に生成します。	2.1	該当しない
29	java.util.List<XBRLRoleRef>	getRoleRefs() インスタンス文書に記述されている <link:roleRef> 要素のリストを取得します。	2.1	該当しない
30	XBRLRoleRef	getRoleRef(java.lang.String roleURI) インスタンス文書に記述されている <link:roleRef> 要素のうち，指定した roleURI と同じ値の roleURI 属性を持つ <link:roleRef> 要素を取得します。	2.1	該当しない
31	XBRLRoleRef	addRoleRef(XBRLRoleRef newRoleRef) インスタンス文書に <link:roleRef> 要素を追加します。	2.1	該当しない
32	XBRLRoleRef	createRoleRef(java.lang.String href, java.lang.String roleURI, java.lang.String base) <link:roleRef> 要素である XBRLRoleRef オブジェクトを新規に生成します。	2.1	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
33	java.util.List<XBRLArcroleRef>	getArcroleRefs() インスタンス文書に記述されている<link:arcroleRef> 要素のリストを取得します。	2.1	該当しない
34	XBRLArcroleRef	getArcroleRef(java.lang.String arcroleURI) インスタンス文書に記述されている<link:arcroleRef> 要素のうち，引数 arcroleURI と同じ値の arcroleURI 属性を持つ<link:arcroleRef> 要素を取得します。	2.1	該当しない
35	XBRLArcroleRef	addArcroleRef(XBRLArcroleRef newArcroleRef) インスタンス文書に <link:arcroleRef> 要素を追加します。	2.1	該当しない
36	XBRLArcroleRef	createArcroleRef(java.lang.String href, java.lang.String arcroleURI, java.lang.String base) <link:arcroleRef> 要素である XBRLArcroleRef オブジェクトを新規に生成します。	2.1	該当しない
37	java.util.List<XBRLContext>	getContexts() インスタンス文書に記述されているコンテキスト要素のリストを取得します。	2.0 および 2.1	該当しない
38	XBRLContext	addContext(XBRLContext newContext) インスタンス文書にコンテキスト要素を追加します。	2.0 および 2.1	該当しない
39	XBRLContext	removeContext(XBRLContext oldContext) インスタンス文書からコンテキスト要素を削除します。	2.0 および 2.1	該当しない
40	XBRLNumericContext	createNumericContext(XBRLEntity entity, XBRLPeriod period, XBRLUnit unit, XBRLScenario scenario, int precision, boolean cwa) 数値コンテキスト要素を新規に生成します。	2.0	該当しない
41	XBRLNonNumericContext	createNonNumericContext(XBRLEntity entity, XBRLPeriod period, XBRLUnit unit, XBRLScenario scenario) 非数値コンテキスト要素を新規に生成します。	2.0	該当しない
42	XBRLContext2	createContext(XBRLEntity entity, XBRLPeriod period, XBRLScenario scenario) コンテキスト要素を新規に生成します。	2.1	該当しない
43	java.util.List<XBRLUnit2>	getUnits() インスタンス文書に記述されている単位要素のリストを取得します。	2.1	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
44	XBRLUnit2	addUnit(XBRLUnit2 newUnit) インスタンス文書に単位要素を追加します。	2.1	該当しない
45	XBRLUnit2	removeUnit(XBRLUnit2 oldUnit) インスタンス文書から単位要素を削除します。	2.1	該当しない
46	XBRLUnit2	createUnit(XBRLQName[] measures) 単位要素である XBRLUnit2 オブジェクトを新規に生成します。	2.1	該当しない
47	XBRLUnit2	createUnit(XBRLQName[] numerator, XBRLQName[] denominator) 分数を指定して，単位要素である XBRLUnit2 オブジェクトを新規に生成します。	2.1	該当しない
48	java.util.List<XBRLInstanceElement>	getChildElements() インスタンス文書のルート要素の子要素のリストを取得します。	2.0 および 2.1	該当しない
49	java.util.List<XBRLInstanceElement>	getChildElements(java.lang.String namespaceURI, java.lang.String name) インスタンス文書のルート要素の子要素のうち，指定した名前を持つ子要素のリストを取得します。	2.0 および 2.1	該当しない
50	XBRLInstanceElement	appendChildElement(XBRLInstanceElement newElement) インスタンス文書のルート要素の子要素のリストの末尾に，指定した要素を追加します。	2.0 および 2.1	該当しない
51	XBRLInstanceElement	insertChildElementBefore(XBRLInstanceElement newElement, XBRLInstanceElement refElement) 引数 newElement に指定した要素を，このインスタンス文書のルート要素の既存の子要素の前に挿入します。	2.0 および 2.1	該当しない
52	XBRLInstanceElement	replaceChildElement(XBRLInstanceElement newElement, XBRLInstanceElement oldElement) 引数 oldElement に指定した子要素を，引数 newElement に指定した要素で置き換えます。	2.0 および 2.1	該当しない
53	XBRLInstanceElement	removeChildElement(XBRLInstanceElement oldElement) インスタンス文書のルート要素の子要素を削除します。	2.0 および 2.1	該当しない
54	java.util.List<XBRLInstanceElement>	getElements() このインスタンス文書に存在する要素のリストを取得します。	2.0 および 2.1	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
55	java.util.List<XBRLInstanceElement>	getElements(java.lang.String namespaceURI, java.lang.String name) このインスタンス文書に存在する，指定した名前の要素のリストを取得します。	2.0 および 2.1	該当しない
56	XBRLInstanceItem	createItem(java.lang.String namespaceURI, java.lang.String name, XBRLContext context) アイテムを新規に生成します。	2.0	該当しない
57	XBRLInstanceItem	createNumericItem(java.lang.String namespaceURI, java.lang.String name, XBRLContext context, XBRLUnit2 unit, XBRLPrecision precision, XBRLDecimals decimals) 数値アイテムを新規に生成します。	2.1	該当しない
58	XBRLInstanceItem	createNonNumericItem(java.lang.String namespaceURI, java.lang.String name, XBRLContext2 context) 非数値アイテムを新規に生成します。	2.1	該当しない
59	XBRLInstanceTuple	createTuple(java.lang.String namespaceURI, java.lang.String name) タプルを新規に生成します。	2.0 および 2.1	該当しない
60	XBRLEntity	createEntity(XBRLIdentifier identifier, XBRLSegment segment) エンティティ要素を表す XBRLEntity オブジェクトを新規に生成します。	2.0 および 2.1	該当しない
61	XBRLPeriod	createPeriod(int periodType, XBRLDateUnion instant, XBRLDateUnion startDate, XBRLDateUnion endDate, XBRLDuration duration) 時期要素である XBRLPeriod オブジェクトを新規に生成します。	2.0 および 2.1	該当しない
62	XBRLPeriod	createPeriod() 時期要素である XBRLPeriod オブジェクトを新規に生成します。	2.1	該当しない
63	XBRLPeriod	createPeriod(XBRLDateUnion instant) 時期要素である XBRLPeriod オブジェクトを新規に生成します。	2.1	該当しない
64	XBRLPeriod	createPeriod(XBRLDateUnion startDate, XBRLDateUnion endDate) 時期要素である XBRLPeriod オブジェクトを新規に生成します。	2.1	該当しない
65	XBRLUnit	createUnit(XBRLMeasureOrOperator measureOrOperator) 単位要素である XBRLUnit オブジェクトを新規に生成します。	2.0	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
66	XBRLScenario	createScenario(java.lang.String name) シナリオ要素である XBRLScenario オブジェクトを新規に生成します。	2.0	該当しない
67	XBRLScenario	createScenario() シナリオ要素である XBRLScenario オブジェクトを新規に生成します。	2.1	該当しない
68	XBRLIdentifier	createIdentifier(java.lang.String namespaceURI, java.lang.String value) 識別子要素である XBRLIdentifier オブジェクトを新規に生成します。	2.0 および 2.1	該当しない
69	XBRLSegment	createSegment(java.lang.String name) セグメント要素である XBRLSegment オブジェクトを新規に生成します。	2.0	該当しない
70	XBRLSegment	createSegment() セグメント要素である XBRLSegment オブジェクトを新規に生成します。	2.1	該当しない
71	XBRLMeasure	createMeasure(XBRLQName value) 計量要素である XBRLMeasure オブジェクトを新規に生成します。	2.0	該当しない
72	XBRLOperator	createOperator(java.lang.String kind, XBRLMeasureOrOperator lhs, XBRLMeasureOrOperator rhs) 演算子要素である XBRLOperator オブジェクトを新規に生成します。	2.0	該当しない
73	java.util.List<XBRLFootnote>	getFootnotes() インスタンス文書に記述されている脚注のリストを取得します。	2.0 および 2.1	該当しない
74	java.util.List<XBRLFootnote>	getFootnotesByLang(java.lang.String lang) インスタンス文書に記述されている脚注のうち，指定した言語の脚注を取得します。	2.0 および 2.1	該当しない
75	java.util.List<XBRLFootnote>	getFootnotesByRole(java.lang.String role) インスタンス文書に記述されている脚注のうち，指定したロールの脚注を取得します。	2.0 および 2.1	該当しない
76	java.util.List<XBRLFootnote>	getFootnotes(java.lang.String lang, java.lang.String role) インスタンス文書に記述されている脚注のうち，指定した言語およびロールの脚注を取得します。	2.0 および 2.1	該当しない
77	XBRLFootnote	addFootnote(XBRLFootnote newFootnote) インスタンス文書に脚注を追加します。	2.0 および 2.1	該当しない
78	XBRLFootnote	removeFootnote(XBRLFootnote oldFootnote) インスタンス文書から脚注を削除します。	2.0 および 2.1	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
79	XBRLFootnote	createFootnote(java.lang.String lang, java.lang.String role, java.lang.String title) 脚注である XBRLFootnote オブジェクトを新規に生成します。	2.0 および 2.1	該当しない
80	XBRLFootnote	createFootnote(java.lang.String footnoteLinkRole, java.lang.String lang, java.lang.String role, java.lang.String title) 脚注である XBRLFootnote オブジェクトを新規に生成します。カスタムロールを指定できます。	2.1	該当しない
81	XBRLInstanceNode	importNode(XBRLInstanceNode sourceNode, boolean deep) ほかの XBRLInstanceDocument オブジェクトで生成されたノードをインポートします。	2.0 および 2.1	該当しない
82	XBRLInstanceDocument	cloneNode(org.w3c.dom.Document w3cdomDocument, boolean deep) インスタンス文書を複製します。	2.1	該当しない
83	XBRLExplicitMember	createExplicitMember(XBRLQName dimension, XBRLQName value) <xbrldi:explicitMember> 要素を表す XBRLExplicitMember オブジェクトを新規に生成します。	2.1	1.0
84	XBRLTypedMember	createTypedMember(XBRLQName dimension) <xbrldi:typedMember> 要素を表す XBRLTypedMember オブジェクトを新規に生成します。	2.1	1.0
85	int	getDimensionsVersion() インスタンス文書が準拠している XBRL Dimensions のバージョンを取得します。	2.1	該当しない

(1) getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書を所有する XBRLDocumentManager オブジェクトを取得します。

戻り値：

このインスタンス文書を所有する XBRLDocumentManager オブジェクトを返します。

（2） getXBRLVersion

```
public int getXBRLVersion()
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書が準拠している XBRL のバージョンを取得します。

戻り値：

このインスタンス文書が XBRL2.0 に準拠している場合は XBRLConstants.XBRL_2_0，XBRL2.1 に準拠している場合は XBRLConstants.XBRL_2_1 を返します。

（3） getPTVInstance

```
public PTVInstance getPTVInstance()
```

XBRL のバージョン：

2.1

説明：

インスタンス文書の解析時に生成された PTVI を取得します。

インスタンス文書を解析する際に完全な妥当性検証を実行した場合は，このメソッドで PTVI を取得できます。

ただし，インスタンス文書が次の方法で生成された場合は，PTVI は生成されないため，このメソッドでは PTVI を取得できません。

- インスタンス文書が新規作成されたものである場合
- インスタンス文書がほかのインスタンス文書を複製して生成された場合
- インスタンス文書を解析する際に，完全な妥当性検証が実行されていない場合

完全な妥当性検証および PTVI の詳細については，「2.10　PTVI の生成（XBRL2.1 の場合）」を参照してください。

また，インスタンス文書の内容を変更しても，その変更は PTVI に反映されません。このメソッドが返す PTVI はインスタンス文書の解析時に生成され，その後は変更されません。

戻り値：

インスタンス文書の解析時に生成された PTVI を表す PTVInstance オブジェクトを返します。PTVI が生成されていない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

(4) getIdManager

```
public XBRLIdManager getIdManager()
```

XBRL のバージョン：

2.1

説明：

インスタンス文書の ID を管理する XBRLIdManager オブジェクトを取得します。

戻り値：

このインスタンス文書の ID を管理する XBRLIdManager オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。[エラーコード：20100]

(5) getTaxonomyDocument

```
public XBRLTaxonomyDocument getTaxonomyDocument()
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書が参照するタクソノミー文書を取得します。

戻り値：

このインスタンス文書が参照するタクソノミー文書を表す，
XBRLTaxonomyDocument オブジェクトを返します。

(6) getInstanceRoot

```
public org.w3c.dom.Element getInstanceRoot()
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書のルート要素を表す org.w3c.dom.Element オブジェクトを取得します。

この getInstanceRoot() メソッドが返した org.w3c.dom.Element オブジェクトに対して，属性の追加・変更・削除，または，子要素の追加・削除などの更新操作をしないでください。更新操作をした場合，XBRL DOM が正常に動作しないおそれがあります。

XBRL2.1 の場合：W3C DOM の API を使用して，この getInstanceRoot() メソッドが返した org.w3c.dom.Element オブジェクトが表すインスタンス文書のルート要素を，他インスタンス文書内以外の任意の位置に移動できます。ほかのインスタンス文書内に，インスタンス文書のルート要素を移動することはできません。

戻り値：

インスタンス文書のルート要素を表す org.w3c.dom.Element オブジェクトを返します。

（7） getNamespaceDeclarations

```
public java.util.List<XBRLNamespaceDeclaration>
getNamespaceDeclarations()
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に記述されている名前空間宣言のリストを取得します。
このメソッドが返す名前空間宣言のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる名前空間宣言に増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

名前空間宣言を表す XBRLNamespaceDeclaration オブジェクトのリストを返します。
正規化される文書を対象に実行した場合，戻り値のリストに含まれる XBRLNamespaceDeclaration オブジェクトの接頭辞は，常に正規化後の値になります。

（8） getNamespaceDeclarationByURI

```
public XBRLNamespaceDeclaration
getNamespaceDeclarationByURI(java.lang.String uri)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した名前空間 URI の名前空間宣言を取得します。
指定した名前空間 URI がデフォルトの名前空間を表す場合は，このメソッドが返した XBRLNamespaceDeclaration オブジェクトの getPrefix() は，null を返します。
このメソッドの戻り値が null となるわけではありません。

引数：

uri - 名前空間 URI を指定します。

戻り値：

引数 uri で指定した名前空間 URI の名前空間宣言を表す，XBRLNamespaceDeclaration オブジェクトを返します。引数 uri で指定した名前空間 URI の名前空間宣言がこのインスタンス文書で定義されていない場合は null を返

します。
正規化される文書を対象に実行した場合，戻り値の XBRLNamespaceDeclaration オブジェクトの接頭辞は，常に正規化後の値になります。

例外：
XBRLDOMRuntimeException -
- 引数 uri に null を指定した場合にスローされます。［エラーコード：20002］

（9） getNamespaceDeclarationByPrefix

```
public XBRLNamespaceDeclaration
getNamespaceDeclarationByPrefix(java.lang.String prefix)
```

XBRL のバージョン：
2.0 および 2.1

説明：
指定した名前空間接頭辞の名前空間宣言を取得します。

引数：
prefix - 名前空間接頭辞を指定します。デフォルトの名前空間の宣言を取得する場合は null を指定します。
正規化される文書を対象に実行する場合，引数の接頭辞に正規化後の値を指定する必要があります。

戻り値：
引数 prefix で指定した名前空間接頭辞の名前空間宣言を表す，XBRLNamespaceDeclaration オブジェクトを返します。引数 prefix で指定した名前空間接頭辞の名前空間宣言が，このインスタンス文書にない場合は null を返します。
正規化される文書を対象に実行した場合，戻り値の XBRLNamespaceDeclaration オブジェクトの接頭辞は，常に正規化後の値になります。

（10） addNamespaceDeclaration

```
public XBRLNamespaceDeclaration
addNamespaceDeclaration(XBRLNamespaceDeclaration
newNamespaceDeclaration)
```

XBRL のバージョン：
2.0 および 2.1

説明：
インスタンス文書のルート要素に名前空間宣言を追加します。
追加しようとした名前空間宣言の名前空間 URI が，すでに別の名前空間接頭辞に対応づけられて宣言されている場合は，既存の名前空間宣言が削除されます。引数に指定した XBRLNamespaceDeclaration オブジェクトが，すでにこの XBRLInstanceDocument オブジェクトに追加されている場合は，何もしません。
ある名前空間 URI の名前空間宣言がすでに追加されていて，その名前空間 URI に

対応する名前空間接頭辞を変更したい場合は，同じ名前空間 URI と別の新しい名前空間接頭辞を持つ XBRLNamespaceDeclaration オブジェクトを作成し，それを引数に指定してこのメソッドを呼び出してください。
すでに設定されている名前空間の名前空間接頭辞が変更された場合，この名前空間に属する要素の名前空間接頭辞が変更されます。ただし，<xbrli:scenario> 要素と 要素の内容には任意の XML 要素を記述できるため，これら二つの要素の子孫ノードについては，名前空間接頭辞は設定しません。

引数：

newNamespaceDeclaration - 追加する名前空間宣言を表す XBRLNamespaceDeclaration オブジェクトを指定します。

戻り値：

追加された名前空間宣言を表す XBRLNamespaceDeclaration オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 newNamespaceDeclaration に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newNamespaceDeclaration に指定した名前空間接頭辞が，すでに別の名前空間 URI に対応づけられて宣言されている場合にスローされます。［エラーコード：20037］
- 引数 newNamespaceDeclaration に指定された名前空間接頭辞が null で，かつ，指定された名前空間に属するグローバル属性がインスタンス文書中に存在する場合にスローされます。［エラーコード：20036］
- 引数 newNamespaceDeclaration に指定した名前空間接頭辞が「xml」で始まる場合にスローされます。「xml」の大文字・小文字は区別されません。［エラーコード：20038］
- 引数 newNamespaceDeclaration に指定した名前空間 URI が，http://www.w3.org/XML/1998/namespace または http://www.w3.org/2000/xmlns/ の場合にスローされます。［エラーコード：20039］
- 引数 newNamespaceDeclaration に指定された名前空間接頭辞が null で，かつ，指定された名前空間 URI が空文字列（""）でなく，かつ，名前空間に属さない要素または修飾名がインスタンス文書中に存在する場合にスローされます。［エラーコード：20070］
- 引数 newNamespaceDeclaration に指定された名前空間接頭辞が，空文字列（""）の場合にスローされます。［エラーコード：20027］

（11）removeNamespaceDeclaration

```
public XBRLNamespaceDeclaration
removeNamespaceDeclaration(XBRLNamespaceDeclaration
oldNamespaceDeclaration)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書から名前空間宣言を削除します。

引数：

oldNamespaceDeclaration - 削除する名前空間宣言を表す XBRLNamespaceDeclaration オブジェクトを指定します。

戻り値：

削除された名前空間宣言を表す XBRLNamespaceDeclaration オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 oldNamespaceDeclaration に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 oldNamespaceDeclaration に指定された名前空間宣言が，このインスタンス文書に記述されていない場合にスローされます。［エラーコード：20040］
- 引数 oldNamespaceDeclaration に指定された名前空間接頭辞が，このインスタンス文書で使用されている場合にスローされます。［エラーコード：20041］
- XBRL2.1 の場合：引数 oldNamespaceDeclaration に指定された名前空間接頭辞が null で，かつ指定された名前空間 URI が空文字列（""）で，かつ名前空間に属さない要素または修飾名がインスタンス文書中に存在するときにスローされます。［エラーコード：20107］

(12) getSchemaLocations

```
public java.util.List<XBRLSchemaLocation> getSchemaLocations()
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書のルート要素の xsi:schemaLocation 属性に記述されている名前空間 URI と，スキーマ文書の位置を示す URI の組を表すオブジェクトのリストを取得します。

このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLSchemaLocation オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

XBRLSchemaLocation オブジェクトのリストを返します。

(13) getSchemaLocation

```
public XBRLSchemaLocation getSchemaLocation(java.lang.String
namespaceURI)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書のルート要素の xsi:schemaLocation 属性に記述されている名前空間 URI と，スキーマ文書の位置を示す URI の組を表すオブジェクトのリストの中から，指定した名前空間 URI のオブジェクトを取得します。

引数：

namespaceURI - 名前空間 URI を指定します。

戻り値：

引数 namespaceURI で指定した名前空間 URI と，そのスキーマ文書の位置を示す URI の組を表す，XBRLSchemaLocation オブジェクトを返します。引数 namespaceURI で指定した名前空間 URI の XBRLSchemaLocation オブジェクトがない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 引数 namespaceURI に null を指定した場合にスローされます。［エラーコード：20002］

(14) addSchemaLocation

```
public XBRLSchemaLocation addSchemaLocation(XBRLSchemaLocation
newSchemaLocation)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書のルート要素の xsi:schemaLocation 属性に，名前空間 URI とスキーマ文書の位置を示す URI の組を追加します。追加しようとした名前空間 URI の組がすでにある場合は，最初に既存のものが削除されます。引数に指定した XBRLSchemaLocation オブジェクトがすでにこの XBRLInstanceDocument オブジェクトに追加されている場合は，何もしません。

引数：

newSchemaLocation - 追加する名前空間 URI とスキーマ文書の位置を示す URI の組を表す，XBRLSchemaLocation オブジェクトを指定します。

戻り値：

追加された XBRLSchemaLocation オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 newSchemaLocation に null を指定した場合にスローされます。［エラーコード：20002］

(15) removeSchemaLocation

```
public XBRLSchemaLocation removeSchemaLocation(XBRLSchemaLocation
oldSchemaLocation)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書のルート要素の xsi:schemaLocation 属性から，指定された名前空間 URI とスキーマ文書の位置を示す URI の組を削除します。

引数：

oldSchemaLocation - 削除する名前空間 URI とスキーマ文書の位置を示す URI の組を表す，XBRLSchemaLocation オブジェクトを指定します。

戻り値：

削除された XBRLSchemaLocation オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 oldSchemaLocation に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 oldSchemaLocation に指定した XBRLSchemaLocation オブジェクトが，このインスタンス文書にない場合にスローされます。［エラーコード：20042］

(16) getBase

```
public java.lang.String getBase()
```

XBRL のバージョン：

2.1

説明：

インスタンス文書のルート要素の xml:base 属性の値を取得します。

戻り値：

インスタンス文書のルート要素の xml:base 属性の値を返します。xml:base 属性がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場

合にスローされます。［エラーコード：20100］

（17）setBase

```
public void setBase(java.lang.String base)
```

XBRL のバージョン：

2.1

説明：

インスタンス文書のルート要素の xml:base 属性の値を設定します。

このメソッドで xml:base 属性の値を設定しても，インスタンス文書中に記述されている <link:schemaRef> 要素などの xlink:href 属性に指定された相対 URI を解決し直すことはありません。この setBase() メソッドは，単に xml:base 属性の値を設定するだけです。ただし，この setBase() メソッドによって xml:base 属性の値を変更すると，インスタンス文書を出力して再度解析した際の相対 URI の解決結果が変わるおそれがあるため，注意してください。

引数：

base - xml:base 属性の値を指定します。xml:base 属性を削除する場合は null を指定します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 base に指定した値が xsd:anyURI 型でない場合にスローされます。［エラーコード：20108］

（18）getId

```
public java.lang.String getId()
```

XBRL のバージョン：

2.1

説明：

インスタンス文書のルート要素の id 属性の値を取得します。

戻り値：

インスタンス文書のルート要素の id 属性の値を返します。id 属性がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

(19) setId

```
public void setId(java.lang.String id)
```

XBRL のバージョン：

2.1

説明：

インスタンス文書のルート要素の id 属性の値を設定します。

引数：

id - id 属性の値を指定します。id 属性を削除する場合は null を指定します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 id に指定した値が xsd:NCName 型でない場合にスローされます。［エラーコード：20108］
- 引数 id で指定されたのと同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。［エラーコード：20028］

(20) getSchemaRefs

```
public java.util.List<XBRLSchemaRef> getSchemaRefs()
```

XBRL のバージョン：

2.1

説明：

インスタンス文書に記述されている <link:schemaRef> 要素のリストを取得します。このメソッドが返す <link:schemaRef> 要素のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる <link:schemaRef> 要素に増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このインスタンス文書に記述されている <link:schemaRef> 要素をすべて含む XBRLSchemaRef オブジェクトのリストを返します。このインスタンス文書に <link:schemaRef> 要素が記述されていない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場

合にスローされます。［エラーコード：20100］

(21) addSchemaRef

```
public XBRLSchemaRef addSchemaRef(XBRLSchemaRef newSchemaRef)
                     throws XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

インスタンス文書に <link:schemaRef> 要素を追加します。指定された XBRLSchemaRef オブジェクトがすでにこのインスタンス文書に記述されている場合は，何もしません。

このメソッドを呼び出すと，指定された <link:schemaRef> 要素が指すタクソノミー本体がタクソノミー文書にない場合は，そのタクソノミー本体を読み込んで解析し，タクソノミー文書に追加します。タクソノミー本体の解析処理は，この XBRLInstanceDocument オブジェクトを生成する際に使用された XBRLInstanceDocumentBuilder オブジェクトの設定に従って実行されます。追加されたタクソノミー本体の情報は，この XBRLInstanceDocument オブジェクトが参照する XBRLTaxonomyDocument オブジェクトに直ちに反映されます。

<link:schemaRef> 要素が指すタクソノミー本体を読み込んで解析する際に，そのタクソノミー本体がほかのタクソノミー本体やリンクベースを参照していて，かつ，それらがタクソノミー文書に存在しない場合は，それらのほかのタクソノミー本体やリンクベースも読み込んで解析し，タクソノミー文書へ追加します。

指定された <link:schemaRef> 要素が指すタクソノミー本体がすでにタクソノミー文書に存在する場合は，タクソノミー本体の解析はされません。単にインスタンス文書に <link:schemaRef> 要素が追加されるだけです。

引数：

newSchemaRef - 追加する <link:schemaRef> 要素を表す XBRLSchemaRef オブジェクトを指定します。

戻り値：

追加された <link:schemaRef> 要素を表す XBRLSchemaRef オブジェクトを返します。

例外：

XBRLDOMException -

- 引数 newSchemaRef に指定された XBRLSchemaRef オブジェクトが指すタクソノミー本体の解析処理でエラーが検出された場合に，スローされます。［エラーコード：エラー要因によって異なる※］

  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 newSchemaRef に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newSchemaRef に指定された XBRLSchemaRef オブジェクトを所有する XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］

(22) createSchemaRef

```
public XBRLSchemaRef createSchemaRef(java.lang.String href,
                                     java.lang.String base)
```

XBRL のバージョン：

2.1

説明：

<link:schemaRef> 要素を表す XBRLSchemaRef オブジェクトを新規に生成します。

引数：

href - <link:schemaRef> 要素の xlink:href 属性の値を指定します。

base - <link:schemaRef> 要素の xml:base 属性の値を指定します。xml:base 属性が不要な場合は null を指定します。

戻り値：

新しく生成された XBRLSchemaRef オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 href に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 href または base に指定した値が xsd:anyURI 型でない場合にスローされます。［エラーコード：20108］

(23) getLinkbaseRefs

```
public java.util.List<XBRLLinkbaseRef> getLinkbaseRefs()
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に記述されている <link:linkbaseRef> 要素のリストを取得します。このメソッドが返す <link:linkbaseRef> 要素のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる <link:linkbaseRef> 要素に増減があった場合，その変更は直ちに反映されます。ま

た，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

このインスタンス文書に存在する <link:linkbaseRef> 要素をすべて含む，XBRLLinkbaseRef オブジェクトのリストを返します。このインスタンス文書に <link:linkbaseRef> 要素がない場合は，要素数が 0 のリストを返します。

(24) getLinkbaseRefs

```
public java.util.List<XBRLLinkbaseRef>
getLinkbaseRefs(java.lang.String role)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に記述されている <link:linkbaseRef> 要素のうち，引数 role と同じ値の xlink:role 属性を持つ <link:linkbaseRef> 要素のリストを取得します。このメソッドが返す <link:linkbaseRef> 要素のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる <link:linkbaseRef> 要素に増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

role - 取得したい <link:linkbaseRef> 要素の xlink:role 属性の値を指定します。xlink:role 属性を持たない <link:linkbaseRef> 要素を取得したい場合は null を指定します。

戻り値：

引数 role と同じ値の xlink:role 属性を持つ <link:linkbaseRef> 要素を表す XBRLLinkbaseRef オブジェクトのリストを返します。このインスタンス文書に，引数 role と同じ値の xlink:role 属性を持つ <link:linkbaseRef> 要素がない場合は，要素数が 0 のリストを返します。

(25) addLinkbaseRef

```
public XBRLLinkbaseRef addLinkbaseRef(XBRLLinkbaseRef
newLinkbaseRef)
                       throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に <link:linkbaseRef> 要素を追加します。指定した XBRLLinkbaseRef オブジェクトが，すでにこのインスタンス文書にある場合は，何もしません。
このメソッドを呼び出すと，引数 newLinkbaseRef に指定した XBRLLinkbaseRef オブジェクトが参照するリンクベースを読み込んで解析します。リンクベース解析処理は，この XBRLInstanceDocument オブジェクトを生成する際に使用された，XBRLInstanceDocumentBuilder オブジェクトの設定に従って実行されます。
このメソッドの呼び出しによって追加されたリンクベースの情報は，この XBRLInstanceDocument オブジェクトが参照する XBRLTaxonomyDocument オブジェクトに直ちに反映されます。
XBRL2.1 の場合：<link:linkbaseRef> 要素が指すリンクベースを読み込んで解析をする際に，そのリンクベースがほかのタクソノミー本体やリンクベースを参照していて，かつ，それらのタクソノミー本体やリンクベースがタクソノミー文書に存在しない場合は，それらのほかのタクソノミー本体やリンクベースも読み込んで解析を行い，タクソノミー文書へ追加します。

引数：

newLinkbaseRef - 追加する <link:linkbaseRef> 要素を表す XBRLLinkbaseRef オブジェクトを指定します。

戻り値：

追加された <link:linkbaseRef> 要素を表す XBRLLinkbaseRef オブジェクトを返します。

例外：

XBRLDOMException -

- 引数 newLinkbaseRef に指定した XBRLLinkbaseRef オブジェクトが参照するリンクベースの解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 引数 newLinkbaseRef に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newLinkbaseRef に指定した XBRLLinkbaseRef オブジェクトを所有する XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］

(26) removeLinkbaseRef

```
public XBRLLinkbaseRef removeLinkbaseRef(XBRLLinkbaseRef
oldLinkbaseRef)
```

XBRL のバージョン：

2.0

説明：

インスタンス文書に記述されている <link:linkbaseRef> 要素を削除します。
このメソッドの呼び出しによって削除されたリンクベースの情報は，この XBRLInstanceDocument オブジェクトが参照する XBRLTaxonomyDocument オブジェクトに直ちに反映されます。

引数：

oldLinkbaseRef - 削除する <link:linkbaseRef> 要素を表す XBRLLinkbaseRef オブジェクトを指定します。

戻り値：

削除された <link:linkbaseRef> 要素を表す XBRLLinkbaseRef オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。［エラーコード：20100］
- 引数 oldLinkbaseRef に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 oldLinkbaseRef に指定した <link:linkbaseRef> 要素が，このインスタンス文書にない場合にスローされます。［エラーコード：20047］

（27）createLinkbaseRef

```
public XBRLLinkbaseRef createLinkbaseRef(java.lang.String role,
                                         java.lang.String href)
```

XBRL のバージョン：

2.0 および 2.1

説明：

<link:linkbaseRef> 要素を表す XBRLLinkbaseRef オブジェクトを新規に生成します。
XBRL2.1 の場合，このメソッドで生成された <link:linkbaseRef> 要素には xml:base 属性は付きません。<link:linkbaseRef> 要素に xml:base 属性を付けたい場合は，createLinkbaseRef(java.lang.String role, java.lang.String href, java.lang.String base) メソッドを使用してください。

引数：

role - <link:linkbaseRef> 要素の xlink:role 属性の値を指定します。xlink:role 属性が不要な場合は null を指定します。
href - <link:linkbaseRef> 要素の xlink:href 属性の値を指定します。

戻り値：

新しく生成された XBRLLinkbaseRef オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 href に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 role または href に指定した値が xsd:anyURI 型でない場合にスローされます。［エラーコード：20108］
- 引数 role に指定した値が絶対 URI でない場合にスローされます。［エラーコード：20109］

(28) createLinkbaseRef

```
public XBRLLinkbaseRef createLinkbaseRef(java.lang.String role,
                                         java.lang.String href,
                                         java.lang.String base)
```

XBRL のバージョン：

2.1

説明：

<link:linkbaseRef> 要素を表す XBRLLinkbaseRef オブジェクトを新規に生成します。

引数：

role - <link:linkbaseRef> 要素の xlink:role 属性の値を指定します。xlink:role 属性が不要な場合は null を指定します。

href - <link:linkbaseRef> 要素の xlink:href 属性の値を指定します。

base - <link:linkbaseRef> 要素の xml:base 属性の値を指定します。xml:base 属性が不要な場合は null を指定します。

戻り値：

新しく生成された XBRLLinkbaseRef オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 href に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 role，href または base に指定した値が xsd:anyURI 型でない場合にスローされます。［エラーコード：20108］
- 引数 role に指定した値が絶対 URI でない場合にスローされます。［エラーコード：20109］

(29) getRoleRefs

```
public java.util.List<XBRLRoleRef> getRoleRefs()
```

XBRL のバージョン：

2.1

説明：

インスタンス文書に記述されている <link:roleRef> 要素のリストを取得します。このメソッドが返す <link:roleRef> 要素のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる <link:roleRef> 要素に増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このインスタンス文書に存在する <link:roleRef> 要素をすべて含む XBRLRoleRef オブジェクトのリストを返します。このインスタンス文書に <link:roleRef> 要素が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

（30）getRoleRef

```
public XBRLRoleRef getRoleRef(java.lang.String roleURI)
```

XBRL のバージョン：

2.1

説明：

インスタンス文書に記述されている <link:roleRef> 要素のうち，指定した roleURI と同じ値の roleURI 属性を持つ <link:roleRef> 要素を取得します。

引数：

roleURI - 取得したい <link:roleRef> 要素の roleURI 属性の値を指定します。

戻り値：

このインスタンス文書に存在する <link:roleRef> 要素のうち，引数 roleURI と同じ値の roleURI 属性を持つ <link:roleRef> 要素を返します。引数 roleURI と同じ値の roleURI 属性を持つ <link:roleRef> 要素が存在しない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 roleURI に null を指定した場合にスローされます。［エラーコード：20002］

（31）addRoleRef

```
public XBRLRoleRef addRoleRef(XBRLRoleRef newRoleRef)
                    throws XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

インスタンス文書に <link:roleRef> 要素を追加します。指定された XBRLRoleRef オブジェクトがすでにこのインスタンス文書に存在する場合は，何もしません。

<link:roleRef> 要素は，タクソノミー本体に記述された <link:roleType> 要素を指します。このメソッドを呼び出すと，<link:roleType> 要素が記述されているタクソノミー本体がタクソノミー文書に存在しない場合は，そのタクソノミー本体を読み込んで解析を行い，タクソノミー文書へ追加します。タクソノミー本体の解析処理は，この XBRLInstanceDocument オブジェクトを生成する際に使用された XBRLInstanceDocumentBuilder オブジェクトの設定に従って実行されます。追加されたタクソノミー本体の情報は，この XBRLInstanceDocument オブジェクトが参照する XBRLTaxonomyDocument オブジェクトに直ちに反映されます。

<link:roleRef> 要素が指すタクソノミー本体を読み込んで解析を行う際に，そのタクソノミー本体がほかのタクソノミー本体やリンクベースを参照していて，かつ，それらがタクソノミー文書に存在しない場合は，それらのほかのタクソノミー本体やリンクベースも読み込んで解析を行い，タクソノミー文書へ追加します。

<link:roleType> 要素が記述されているタクソノミー本体がすでにタクソノミー文書に存在する場合は，タクソノミー本体の解析処理は行われません。単にインスタンス文書に <link:roleRef> 要素が追加されるだけです。

追加した <link:roleRef> 要素が指す <link:roleType> 要素は，引数 newRoleRef に指定した XBRLRoleRef オブジェクトに対して getRoleType() メソッドを呼び出すことによって取得できます。

引数：

newRoleRef - 追加する <link:roleRef> 要素を表す XBRLRoleRef オブジェクトを指定します。

戻り値：

追加された <link:roleRef> 要素を表す XBRLRoleRef オブジェクトを返します。

例外：

XBRLDOMException -

- タクソノミー本体の解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］

  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 newRoleRef に null を指定した場合にスローされます。［エラーコード：20002］

- 引数 newRoleRef に指定された XBRLRoleRef オブジェクトを所有する XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合に，スローされます。［エラーコード：20021］
- 引数 newRoleRef に指定された <link:roleRef> 要素の roleURI 属性と同じ値の roleURI 属性を持つ <link:roleRef> 要素がインスタンス文書中に存在する場合にスローされます。［エラーコード：20111］

(32) createRoleRef

```
public XBRLRoleRef createRoleRef(java.lang.String href,
                                 java.lang.String roleURI,
                                 java.lang.String base)
```

XBRL のバージョン：

2.1

説明：

<link:roleRef> 要素である XBRLRoleRef オブジェクトを新規に生成します。

引数：

href - <link:roleRef> 要素の xlink:href 属性の値を指定します。

roleURI - <link:roleRef> 要素の roleURI 属性の値を指定します。

base - <link:roleRef> 要素の xml:base 属性の値を指定します。xml:base 属性が不要な場合は null を指定します。

戻り値：

新しく作成された XBRLRoleRef オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 href または roleURI に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 href，roleURI または base に指定した値が xsd:anyURI 型でない場合にスローされます。［エラーコード：20108］
- 引数 roleURI に指定した値が絶対 URI でない場合にスローされます。［エラーコード：20109］
- 引数 roleURI に指定したロールが XBRL 仕様で規定されている標準値である場合にスローされます。［エラーコード：20113］

(33) getArcroleRefs

```
public java.util.List<XBRLArcroleRef> getArcroleRefs()
```

XBRL のバージョン：

2.1

説明：

インスタンス文書に記述されている <link:arcroleRef> 要素のリストを取得します。このメソッドが返す <link:arcroleRef> 要素のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる <link:arcroleRef> 要素に増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このインスタンス文書に存在する <link:arcroleRef> 要素をすべて含む XBRLArcroleRef オブジェクトのリストを返します。このインスタンス文書に <link:arcroleRef> 要素が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

（34）getArcroleRef

```
public XBRLArcroleRef getArcroleRef(java.lang.String arcroleURI)
```

XBRL のバージョン：

2.1

説明：

インスタンス文書に記述されている <link:arcroleRef> 要素のうち，引数 arcroleURI と同じ値の arcroleURI 属性を持つ <link:arcroleRef> 要素を取得します。

引数：

arcroleURI - 取得したい <link:arcroleRef> 要素の arcroleURI 属性の値を指定します。

戻り値：

このインスタンス文書に存在する <link:arcroleRef> 要素のうち，引数 arcroleURI と同じ値の arcroleURI 属性を持つ <link:arcroleRef> 要素を表す XBRLArcroleRef オブジェクトを返します。引数 arcroleURI と同じ値の arcroleURI 属性を持つ <link:arcroleRef> 要素が存在しない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 arcroleURI に null を指定した場合にスローされます。［エラーコード：

20002]

(35) addArcroleRef

```
public XBRLArcroleRef addArcroleRef(XBRLArcroleRef newArcroleRef)
                      throws XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

インスタンス文書に <link:arcroleRef> 要素を追加します。指定された XBRLArcroleRef オブジェクトがすでにこのインスタンス文書に存在する場合は，何もしません。

<link:arcroleRef> 要素は，タクソノミー本体に記述された <link:arcroleType> 要素を指します。このメソッドを呼び出すと，<link:arcroleType> 要素が記述されているタクソノミー本体がタクソノミー文書に存在しない場合は，そのタクソノミー本体を読み込んで解析を行い，タクソノミー文書へ追加します。タクソノミー本体の解析処理は，この XBRLInstanceDocument オブジェクトを生成する際に使用された XBRLInstanceDocumentBuilder オブジェクトの設定に従って実行されます。追加されたタクソノミー本体の情報は，この XBRLInstanceDocument オブジェクトが参照する XBRLTaxonomyDocument オブジェクトに直ちに反映されます。

<link:arcroleRef> 要素が指すタクソノミー本体を読み込んで解析を行う際に，そのタクソノミー本体がほかのタクソノミー本体やリンクベースを参照していて，かつ，それらがタクソノミー文書に存在しない場合は，それらのほかのタクソノミー本体やリンクベースも読み込んで解析を行い，タクソノミー文書へ追加します。

<link:arcroleType> 要素が記述されているタクソノミー本体がすでにタクソノミー文書に存在する場合は，タクソノミー本体の解析処理は行われません。単にインスタンス文書に <link:arcroleRef> 要素が追加されるだけです。

追加した <link:arcroleRef> 要素が指す <link:arcroleType> 要素は，引数 newArcroleRef に指定した XBRLArcroleRef オブジェクトに対して getArcroleType() メソッドを呼び出すことによって取得できます。

引数：

newArcroleRef - 追加する <link:arcroleRef> 要素を表す XBRLArcroleRef オブジェクトを指定します。

戻り値：

追加された <link:arcroleRef> 要素を表す XBRLArcroleRef オブジェクトを返します。

例外：

XBRLDOMException -

- タクソノミー本体の解析処理でエラーが検出された場合にスローされます。[エラーコード：エラー要因によって異なる※]

注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 newArcroleRef に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newArcroleRef に指定された XBRLArcroleRef オブジェクトを所有する XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］
- 引数 newArcroleRef に指定された <link:arcroleRef> 要素の arcroleURI 属性と同じ値の arcroleURI 属性を持つ <link:arcroleRef> 要素がインスタンス文書中に存在する場合にスローされます。［エラーコード：20112］

(36) createArcroleRef

```
public XBRLArcroleRef createArcroleRef(java.lang.String href,
                                       java.lang.String arcroleURI,
                                        java.lang.String base)
```

XBRL のバージョン：

2.1

説明：

<link:arcroleRef> 要素である XBRLArcroleRef オブジェクトを新規に生成します。

引数：

href - <link:arcroleRef> 要素の xlink:href 属性の値を指定します。

arcroleURI - <link:arcroleRef> 要素の arcroleURI 属性の値を指定します。

base - <link:arcroleRef> 要素の xml:base 属性の値を指定します。xml:base 属性が不要な場合は null を指定します。

戻り値：

新しく作成された XBRLArcroleRef オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 href または arcroleURI に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 href，arcroleURI または base に指定した値が xsd:anyURI 型でない場合にスローされます。［エラーコード：20108］
- 引数 arcroleURI に指定した値が絶対 URI でない場合にスローされます。［エラーコード：20109］

- 引数 arcroleURI に指定したアークロールが XBRL 仕様で規定されている標準値である場合にスローされます。［エラーコード：20114］

(37) getContexts

```
public java.util.List<XBRLContext> getContexts()
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に記述されているコンテキスト要素のリストを取得します。このメソッドが返すコンテキストのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれるコンテキストに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

このインスタンス文書に存在するコンテキストをすべて含む，XBRLContext オブジェクトのリストを返します。このインスタンス文書にコンテキストがない場合は，要素数が 0 のリストを返します。

(38) addContext

```
public XBRLContext addContext(XBRLContext newContext)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書にコンテキスト要素を追加します。指定した XBRLContext オブジェクトがすでにこのインスタンス文書に存在する場合は，何もしません。

引数：

newContext - 追加するコンテキストを表す XBRLContext オブジェクトを指定します。

戻り値：

追加されたコンテキストを表す XBRLContext オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 newContext に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newContext に指定したコンテキストの id 属性と同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。［エラーコード：20028］

- 引数 newContext に指定した XBRLContext オブジェクトを所有する XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］

(39) removeContext

```
public XBRLContext removeContext(XBRLContext oldContext)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書からコンテキスト要素を削除します。
コンテキストを削除すると，そのコンテキストを参照しているアイテムもすべて，インスタンス文書から自動的に削除されます。

引数：

oldContext - 削除するコンテキストを表す XBRLContext オブジェクトを指定します。

戻り値：

削除されたコンテキストを表す XBRLContext オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 oldContext に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 oldContext に指定したコンテキストが，このインスタンス文書にない場合にスローされます。［エラーコード：20043］

(40) createNumericContext

```
public XBRLNumericContext createNumericContext(XBRLEntity entity,
                                               XBRLPeriod period,
                                               XBRLUnit unit,
                                            XBRLScenario scenario,
                                               int precision,
                                               boolean cwa)
```

XBRL のバージョン：

2.0

説明：

数値コンテキスト要素を新規に生成します。
数値コンテキストの id 属性の値は，このメソッドで数値コンテキストを生成する際に自動的に採番されます。id 属性の値は，数値コンテキストを生成する際に自動的に採番されますが，XBRLContext の setId() メソッドを使用して任意の値を設定することもできます。自動採番機能の詳細については，「3.3　ID 管理と採番」を参照してください。

引数：

entity - エンティティ要素を表す XBRLEntity オブジェクトを指定します。

period - 時期要素を表す XBRLPeriod オブジェクトを指定します。

unit - 単位要素を表す XBRLUnit オブジェクトを指定します。

scenario - シナリオ要素を表す XBRLScenario オブジェクトを指定します。シナリオ要素がない場合は null を指定します。

precision - precision 属性の値を指定します。1 以上の値を指定します。

cwa - cwa 属性の値を指定します。

戻り値：

新しく生成された XBRLNumericContext オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。［エラーコード：20100］
- 引数 entity，period または unit に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 entity，period，unit または scenario に指定したオブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］
- 引数 entity，period，unit または scenario に指定したオブジェクトが，すでにほかの XBRLContext オブジェクトに設定されている場合にスローされます。［エラーコード：20005］
- 引数 precision に，0 以下の値を指定した場合にスローされます。［エラーコード：20014］

(41) createNonNumericContext

```
public XBRLNonNumericContext createNonNumericContext(XBRLEntity
entity,
                                                    XBRLPeriod period,
                                                        XBRLUnit unit,
                                                 XBRLScenario scenario)
```

XBRL のバージョン：

2.0

説明：

非数値コンテキスト要素を新規に生成します。

非数値コンテキストの id 属性の値は，このメソッドで非数値コンテキストを生成する際に自動的に採番されます。id 属性の値は，非数値コンテキストを生成する際に自動的に採番されますが，XBRLContext の setId() メソッドを使用して任意の値を設定することもできます。自動採番機能の詳細については，「3.3　ID 管理と採番」を参照してください。

引数：

entity - エンティティ要素を表す XBRLEntity オブジェクトを指定します。

period - 時期要素を表す XBRLPeriod オブジェクトを指定します。

unit - 単位要素を表す XBRLUnit オブジェクトを指定します。単位要素がない場合は null を指定します。

scenario - シナリオ要素を表す XBRLScenario オブジェクトを指定します。シナリオ要素がない場合は null を指定します。

戻り値：

新しく生成された XBRLNonNumericContext オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。［エラーコード：20100］
- 引数 entity，period または unit に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 entity，period，unit または scenario に指定したオブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］
- 引数 entity，period，unit または scenario に指定したオブジェクトが，すでにほかの XBRLContext オブジェクトに設定されている場合にスローされます。［エラーコード：20005］

(42) createContext

```
public XBRLContext2 createContext(XBRLEntity entity,
                                  XBRLPeriod period,
                                  XBRLScenario scenario)
```

XBRL のバージョン：

2.1

説明：

コンテキスト要素を新規に生成します。

コンテキストの id 属性の値は，このメソッドでコンテキストを生成する際に自動的に採番されます。id 属性の値は，コンテキストを生成する際に自動的に採番されますが，XBRLContext の setId() メソッドを使用して任意の値を設定することもできます。自動採番の詳細については，「3.3　ID 管理と採番」を参照してください。

引数：

entity - エンティティ要素を表す XBRLEntity オブジェクトを指定します。

period - 時期要素を表す XBRLPeriod オブジェクトを指定します。

scenario - シナリオ要素を表す XBRLScenario オブジェクトを指定します。シナリオ要素がない場合は null を指定します。

戻り値：

新しく生成された XBRLContext2 オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 entity または period に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 entity，period または scenario に指定されたオブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］
- 引数 entity，period または scenario に指定されたオブジェクトが，すでにほかの XBRLContext オブジェクトに設定されている場合にスローされます。［エラーコード：20005］

(43) getUnits

```
public java.util.List<XBRLUnit2> getUnits()
```

XBRL のバージョン：

2.1

説明：

インスタンス文書に記述されている単位要素のリストを取得します。
このメソッドが返す単位要素のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる単位要素に増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このインスタンス文書に存在する単位要素をすべて含む XBRLUnit2 オブジェクトのリストを返します。このインスタンス文書に単位要素が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

(44) addUnit

```
public XBRLUnit2 addUnit(XBRLUnit2 newUnit)
```

XBRL のバージョン：

2.1

説明：

インスタンス文書に単位要素を追加します。指定された XBRLUnit2 オブジェクトがすでにこのインスタンス文書に存在する場合は，何もしません。

引数：

newUnit - 追加する単位要素を表す XBRLUnit2 オブジェクトを指定します。

戻り値：

追加された単位要素を表す XBRLUnit2 オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 newUnit に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newUnit に指定された XBRLUnit2 オブジェクトを所有する XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］
- 引数 newUnit に指定された単位要素の id 属性と同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。［エラーコード：20028］

(45) removeUnit

```
public XBRLUnit2 removeUnit(XBRLUnit2 oldUnit)
```

XBRL のバージョン：

2.1

説明：

インスタンス文書から単位要素を削除します。単位要素を削除すると，その単位要素を参照しているアイテムもすべて，インスタンス文書から自動的に削除されます。

引数：

oldUnit - 削除する単位要素を表す XBRLUnit2 オブジェクトを指定します。

戻り値：

削除された単位要素を表す XBRLUnit2 オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 oldUnit に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 oldUnit に指定された単位要素が，このインスタンス文書に存在しない場合にスローされます。［エラーコード：20124］

(46) createUnit

```
public XBRLUnit2 createUnit(XBRLQName[] measures)
```

XBRL のバージョン：

2.1

説明：

単位要素である XBRLUnit2 オブジェクトを新規に生成します。
このメソッドで生成する XBRLUnit2 オブジェクトの内容は，一つ以上の基本単位によって構成されます。つまり，XBRLUnit2.getUnitType() メソッドは XBRLUnit2.MULTIPLY を返します。
単位要素の内容には，「基本単位を表す XBRLQName オブジェクトの名前空間 URI が "http://www.xbrl.org/2003/instance" である場合，基本単位を表す XBRLQName オブジェクトのローカルパートは，"pure" または "shares" のどちらかでなければならない」という制約があります。この制約に違反する単位要素を生成しようとした場合は，XBRLDOMRuntimeException がスローされます。
単位要素の id 属性の値は，このメソッドで単位要素を生成する際に自動的に採番されます。id 属性の値は，単位要素を生成する際に自動的に採番されますが，XBRLUnit2 の setId() メソッドを使用して任意の値を設定することもできます。自動採番の詳細については，「3.3　ID 管理と採番」を参照してください。

引数：

measures - 基本単位を表す XBRLQName オブジェクトの配列を指定します。配列の要素数は 1 以上である必要があります。基本単位を表す <xbrli:measure> 要素のインスタンス文書中での出現順は，この引数に指定された配列の中での XBRLQName オブジェクトの順序と同じになります。

戻り値：

新しく生成された XBRLUnit2 オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 measures に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 measures に指定した配列の要素数が 0 である場合にスローされます。［エラーコード：20013］
- 引数 measures に指定した配列中に値が null である要素が存在する場合にスローされます。［エラーコード：20012］
- 引数 measures に指定した配列中の XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］

- 引数 measures に指定した配列中の XBRLQName オブジェクトが名前空間に属さず，かつ，XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］
- 引数 measures に指定した配列の中に，上記の制約に違反する XBRLQName オブジェクトがある場合にスローされます。［エラーコード：20125］

(47) createUnit

```
public XBRLUnit2 createUnit(XBRLQName[] numerator,
                            XBRLQName[] denominator)
```

XBRL のバージョン：

2.1

説明：

分数を指定して，単位要素である XBRLUnit2 オブジェクトを新規に生成します。このメソッドで生成する XBRLUnit2 オブジェクトの内容は，分数です。つまり，XBRLUnit2.getUnitType() メソッドは XBRLUnit2.DIVIDE を返します。

単位要素の内容には次のような制約があります。これらの制約に違反する単位要素を生成しようとした場合は，XBRLDOMRuntimeException がスローされます。

- 分子と分母に同じ基本単位を含めてはいけません。同じ基本単位とは，基本単位を表す XBRLQName オブジェクトの名前空間 URI とローカルパートが同じであるもののことです。
- 基本単位を表す XBRLQName オブジェクトの名前空間 URI が "http://www.xbrl.org/2003/instance" である場合，基本単位を表す XBRLQName オブジェクトのローカルパートは，"pure" または "shares" のどちらかである必要があります。

単位要素の id 属性の値は，このメソッドで単位要素を生成する際に自動的に採番されます。id 属性の値は，単位要素を生成する際に自動的に採番されますが，XBRLUnit2 の setId() メソッドを使用して任意の値を設定することもできます。自動採番の詳細については，「3.3　ID 管理と採番」を参照してください。

引数：

numerator - 分数の分子を構成する基本単位を表す XBRLQName オブジェクトの配列を指定します。配列の要素数は 1 以上である必要があります。分数の分子を構成する基本単位を表す <xbrli:measure> 要素のインスタンス文書中での出現順は，この引数に指定された配列の中での XBRLQName オブジェクトの順序と同じになります。

denominator - 分数の分母を構成する基本単位を表す XBRLQName オブジェクトの配列を指定します。配列の要素数は 1 以上である必要があります。分数の分母を構成する基本単位を表す <xbrli:measure> 要素のインスタンス文書中での出現順は，この引数に指定された配列の中での XBRLQName オブジェクトの順序と同じになります。

戻り値：

新しく生成された XBRLUnit2 オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 numerator または denominator に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 numerator または denominator に指定した配列の要素数が 0 である場合にスローされます。［エラーコード：20013］
- 引数 numerator または denominator に指定した配列中に値が null である要素が存在する場合にスローされます。［エラーコード：20012］
- 引数 numerator または denominator に指定した配列中の XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］
- 引数 numerator または denominator に指定した配列中の XBRLQName オブジェクトが名前空間に属さず，かつ，XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］
- 引数 numerator と denominator で指定した単位要素の内容が，上記の制約に違反している場合にスローされます。［エラーコード：20125，20126］

(48) getChildElements

```
public java.util.List<XBRLInstanceElement> getChildElements()
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書のルート要素の子要素のリストを取得します。子要素は，アイテムまたはタプルです。
このメソッドが返すリスト中での XBRLInstanceElement オブジェクトの順序は，インスタンス文書での要素の出現順と同じです。
このメソッドが返す XBRLInstanceElement オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLInstanceElement オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，
java.lang.UnsupportedOperationException がスローされます。

戻り値：

このインスタンス文書のルート要素の子要素をすべて含む，XBRLInstanceElement

オブジェクトのリストを返します。インスタンス文書のルート要素に子要素がない場合は，要素数が 0 のリストを返します。

(49) getChildElements

```
public java.util.List<XBRLInstanceElement>
getChildElements(java.lang.String namespaceURI,
                                        java.lang.String name)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書のルート要素の子要素のうち，指定した名前を持つ子要素のリストを取得します。子要素は，アイテムまたはタプルです。

インスタンス文書のルート要素の子要素として，同じ名前の要素が二つ以上存在することがあります。このため，このメソッドは，指定した名前の子要素を，XBRLInstanceElement オブジェクトのリストとして返します。リスト中での XBRLInstanceElement オブジェクトの順序は，インスタンス文書での要素の出現順と同じです。

このメソッドが返す XBRLInstanceElement オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLInstanceElement オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

namespaceURI - 子要素の名前空間 URI を指定します。

name - 子要素の名前を指定します。

戻り値：

引数 namespaceURI で指定した名前空間 URI に属し，引数 name で指定した名前を持つ子要素をすべて含む，XBRLInstanceElement オブジェクトのリストを返します。指定した子要素がない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 namespaceURI または name に null を指定した場合にスローされます。［エラーコード：20002］

(50) appendChildElement

```
public XBRLInstanceElement appendChildElement(XBRLInstanceElement
newElement)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書のルート要素の子要素のリストの末尾に，指定した要素を追加します。子要素は，アイテムまたはタプルです。
指定した要素がすでにほかのオブジェクトの子となっている場合は，最初に既存の親から指定した要素が削除されます。

引数：

newElement - 追加する子要素を表す XBRLInstanceElement オブジェクトを指定します。

戻り値：

追加された子要素を表す XBRLInstanceElement オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 newElement に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newElement に指定したオブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］
- 引数 newElement に指定した要素が非アクティブで，かつ，引数 newElement に指定した要素（タプルの場合は子孫要素を含む）の id 属性と同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。［エラーコード：20028］

（51）insertChildElementBefore

```
public XBRLInstanceElement
insertChildElementBefore(XBRLInstanceElement newElement,
                                                XBRLInstanceElement
refElement)
```

XBRL のバージョン：

2.0 および 2.1

説明：

引数 newElement に指定した要素を，このインスタンス文書のルート要素の既存の子要素の前に挿入します。子要素は，アイテムまたはタプルです。
引数 refElement が null の場合，引数 newElement に指定した要素は，このインスタンス文書のルート要素の子要素のリストの末尾に挿入されます。指定した要素がすでにほかのオブジェクトの子となっている場合は，最初に既存の親から指定した要素が削除されます。引数 newElement に指定した XBRLInstanceElement オブジェクトが，引数 refElement に指定した XBRLInstanceElement オブジェクトと同じ場合は，このメソッドは何もしません。

引数：

newElement - 挿入する子要素を表す XBRLInstanceElement オブジェクトを指定し

ます。
refElement - 参照する子要素を表す XBRLInstanceElement オブジェクトを指定します。引数 newElement で指定した子要素は，この子要素の直前に挿入されます。

戻り値：
挿入された子要素を表す XBRLInstanceElement オブジェクトを返します。

例外：
XBRLDOMRuntimeException -
- 引数 newElement に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 refElement に指定した子要素が，このインスタンス文書のルート要素の子要素でない場合にスローされます。［エラーコード：20035］
- 引数 newElement に指定した要素が非アクティブで，かつ引数 newElement に指定した要素（タプルの場合は子孫要素を含む）の id 属性と同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。［エラーコード：20028］
- 引数 newElement に指定したオブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］

(52) replaceChildElement

```
public XBRLInstanceElement replaceChildElement(XBRLInstanceElement
newElement,
                                               XBRLInstanceElement oldElement)
```

XBRL のバージョン：
2.0 および 2.1

説明：
引数 oldElement に指定した子要素を，引数 newElement に指定した要素で置き換えます。子要素は，アイテムまたはタプルです。
指定した要素がすでにほかのオブジェクトの子となっている場合は，最初に既存の親から指定した要素が削除されます。引数 newElement に指定した XBRLInstanceElement オブジェクトが引数 oldElement に指定した XBRLInstanceElement オブジェクトと同じ場合は，このメソッドは何もしません。
置き換える前に，id 属性の値が重複していないかがチェックされます。このチェックは，引数 oldElement を含む置き換えられる前の文書の状態を基に実行されます。
例えば，引数 newElement に指定した要素が非アクティブで，かつ，引数 newElement に指定した要素（タプルの場合は子孫要素を含む）の id 属性の値が，引数 oldElement で指定した要素（タプルの場合は子孫要素を含む）の id 属性の値と重複している場合は，XBRLDOMRuntimeException がスローされます。置き換えの対象となる要素の id 属性と同じ値の id 属性を持つ別の要素をインスタンス文書に追加したい場合は，

XBRLInstanceDocument.removeChildElement(XBRLInstanceElement oldElement) メソッドで置き換えの対象となる要素を削除してから，追加してください。

引数：

newElement - 挿入する子要素を表す XBRLInstanceElement オブジェクトを指定します。

oldElement - 置き換えの対象となる子要素を表す XBRLInstanceElement オブジェクトを指定します。

戻り値：

置き換えの対象となった子要素を表す XBRLInstanceElement オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 newElement または oldElement に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 oldElement に指定した子要素が，このインスタンス文書のルート要素の子要素でない場合にスローされます。［エラーコード：20035］
- 引数 newElement に指定した要素が非アクティブで，かつ引数 newElement に指定した要素（タプルの場合は子孫要素を含む）の id 属性と同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。［エラーコード：20028］
- 引数 newElement に指定したオブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］

(53) removeChildElement

```
public XBRLInstanceElement removeChildElement(XBRLInstanceElement
oldElement)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書のルート要素の子要素を削除します。子要素は，アイテムまたはタプルです。

引数：

oldElement - 削除する子要素を表す XBRLInstanceElement オブジェクトを指定します。

戻り値：

削除された子要素を表す XBRLInstanceElement オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 oldElement に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 oldElement に指定した子要素が，このインスタンス文書のルート要素の子要素でない場合にスローされます。［エラーコード：20035］

(54) getElements

```
public java.util.List<XBRLInstanceElement> getElements()
```

XBRL のバージョン：

2.0 および 2.1

説明：

このインスタンス文書に存在する要素のリストを取得します。要素は，アイテムまたはタプルです。

getChildElements() が，この XBRLInstanceDocument オブジェクトのルート要素の子要素を取得するのに対して，このメソッドは，インスタンス文書に存在する要素を取得する点が異なります。

このメソッドが返す XBRLInstanceElement オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLInstanceElement オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，

java.lang.UnsupportedOperationException がスローされます。

戻り値：

このインスタンス文書に存在する要素をすべて含む XBRLInstanceElement オブジェクトのリストを返します。このインスタンス文書に要素がない場合は，要素数が 0 のリストを返します。

(55) getElements

```
public java.util.List<XBRLInstanceElement>
getElements(java.lang.String namespaceURI,
                                  java.lang.String name)
```

XBRL のバージョン：

2.0 および 2.1

説明：

このインスタンス文書に存在する，指定した名前の要素のリストを取得します。要素は，アイテムまたはタプルです。

getChildElements() が，この XBRLInstanceDocument オブジェクトのルート要素の子要素を取得するのに対して，このメソッドは，インスタンス文書に存在する要素を取得する点が異なります。

このメソッドが返す XBRLInstanceElement オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLInstanceElement オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，
java.lang.UnsupportedOperationException がスローされます。

引数：

namespaceURI - 取得する要素の名前空間 URI を指定します。
name - 取得する要素の名前を指定します。

戻り値：

引数 namespaceURI で指定した名前空間 URI に属し，引数 name で指定した名前を持つ要素をすべて含む，XBRLInstanceElement オブジェクトのリストを返します。このインスタンス文書に指定した要素がない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 namespaceURI または name に null を指定した場合にスローされます。［エラーコード：20002］

(56) createItem

```
public XBRLInstanceItem createItem(java.lang.String namespaceURI,
                                   java.lang.String name,
                                   XBRLContext context)
```

XBRL のバージョン：

2.0

説明：

アイテムを新規に生成します。
このメソッドが返す XBRLInstanceItem オブジェクトは，値が設定されていない状態となります。つまり，XBRLInstanceItem の hasValue() は false を返します。このメソッドで XBRLInstanceItem オブジェクトを生成したあと，
XBRLInstanceItem.setValue() メソッドまたは XBRLInstanceItem.setString() メソッドを使用して適切な値を設定してください。
アイテムの id 属性の値は，このメソッドでアイテムを生成する際に自動的に採番されます。id 属性の値は，アイテムを生成する際に自動的に採番されますが，
XBRLInstanceElement.setId() メソッドを使用して任意の値を設定することもできます。自動採番の詳細については，「3.3　ID 管理と採番」を参照してください。

引数：

namespaceURI - アイテムの名前空間 URI を指定します。
name - アイテムの名前を指定します。

context - アイテムが参照するコンテキストを指定します。

戻り値：

新しく生成された XBRLInstanceItem オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。［エラーコード：20100］
- 引数 namespaceURI，name または context に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 namespaceURI および name で指定したアイテムがタクソノミー文書で定義されていない場合にスローされます。［エラーコード：20044］
- 引数 context に指定したオブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］
- 引数 context が表すコンテキストが，引数 namespaceURI および name で指定したアイテムのデータ型に合わない場合にスローされます。［エラーコード：20032］
- 引数 namespaceURI および name で指定したアイテムが，タクソノミー文書で抽象的な要素として定義されている場合にスローされます。［エラーコード：20065］

(57) createNumericItem

```
public XBRLInstanceItem createNumericItem(java.lang.String
namespaceURI,
                                          java.lang.String name,
                                          XBRLContext context,
                                          XBRLUnit2 unit,
                                          XBRLPrecision precision,
                                          XBRLDecimals decimals)
```

XBRL のバージョン：

2.1

説明：

数値アイテムを新規に生成します。

このメソッドが返す XBRLInstanceItem オブジェクトは数値アイテムを表すため，XBRLInstanceItem.isNumeric() メソッドは true を返します。

このメソッドが返す XBRLInstanceItem オブジェクトは，値が設定されていない状態となります。つまり，XBRLInstanceItem.hasValue() は false を返します。このメソッドで XBRLInstanceItem オブジェクトを生成したあと，XBRLInstanceItem.setValue() メソッドまたは XBRLInstanceItem.setString() メソッドを使用して適切な値を設定してください。

アイテムが参照する単位要素には，アイテムのベースデータ型によって幾つかの制約があります。これらの制約に違反する単位要素を設定しようとした場合は，XBRLDOMRuntimeException がスローされます。単位要素の制約については，

XBRLInstanceItem インターフェースの「6.6.13(8)　setUnit」を参照してください。
このメソッドで生成されたアイテムは，id 属性を持ちません。

引数：

namespaceURI - アイテムの名前空間 URI を指定します。
name - アイテムの名前を指定します。
context - アイテムが参照するコンテキスト要素を指定します。
unit - アイテムが参照する単位要素を指定します。
precision - アイテムのベースデータ型が FRACTION_ITEM 以外である場合は，アイテムの有効桁数を指定します。引数 decimals に小数位を指定する場合は，この引数には null を指定してください。また，アイテムのベースデータ型が FRACTION_ITEM である場合は null を指定してください。
decimals - アイテムのベースデータ型が FRACTION_ITEM 以外である場合は，アイテムの小数位を指定します。引数 precision に有効桁数を指定する場合は，この引数には null を指定してください。また，アイテムのベースデータ型が FRACTION_ITEM である場合は null を指定してください。

戻り値：

新しく生成された XBRLInstanceItem オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 namespaceURI，name，context または unit に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 namespaceURI および name で指定したアイテムがタクソノミー文書で定義されていない場合にスローされます。［エラーコード：20044］
- 引数 namespaceURI および name で指定したアイテムが非数値である場合にスローされます。［エラーコード：20131］
- 引数 namespaceURI および name で指定したアイテムがタクソノミー文書で抽象的な要素として定義されている場合にスローされます。［エラーコード：20065］
- 引数 context または unit に指定したオブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］
- 引数 context に指定したコンテキスト要素の時期要素の内容が，生成するアイテムの要素宣言での xbrli:periodType 属性の指定と矛盾している場合にスローされます。xbrli:periodType 属性の値が "instant" の場合は，時期要素の内容は XBRLPeriod.INSTANT でなければなりません。また，xbrli:periodType 属性の値が "duration" の場合は，時期要素の内容は XBRLPeriod.FOREVER か XBRLPeriod.STARTDATE_ENDDATE である必要があります。［エラーコード：

20140］

- 引数 unit に指定した XBRLUnit2 オブジェクトが，アイテムのベースデータ型による単位要素の制約に違反している場合にスローされます。単位要素の制約については，XBRLInstanceItem インターフェースの「6.6.13(8)　setUnit」を参照してください。［エラーコード：20127］
- 引数 namespaceURI および name で指定したアイテムのベースデータ型が FRACTION_ITEM である場合に，引数 precision または decimals に null 以外の値を指定した場合にスローされます。［エラーコード：20133］
- 引数 namespaceURI および name で指定したアイテムのベースデータ型が FRACTION_ITEM 以外で，かつ，引数 precision および decimals の両方ともに null を指定した場合にスローされます。有効桁数と小数位はどちらか一方を指定する必要があります。［エラーコード：20134］
- 引数 namespaceURI および name で指定したアイテムのベースデータ型が FRACTION_ITEM 以外で，かつ，引数 precision および decimals の両方ともに null 以外の値を指定した場合にスローされます。有効桁数と小数位は，どちらか一方だけを指定する必要があります。［エラーコード：20135］

(58) createNonNumericItem

```
public XBRLInstanceItem createNonNumericItem(java.lang.String
namespaceURI,
                                             java.lang.String name,
                                              XBRLContext2 context)
```

XBRL のバージョン：

2.1

説明：

非数値アイテムを新規に生成します。

このメソッドが返す XBRLInstanceItem オブジェクトは非数値アイテムを表すため，XBRLInstanceItem.isNumeric() メソッドは false を返します。

このメソッドが返す XBRLInstanceItem オブジェクトは，値が設定されていない状態となります。つまり，XBRLInstanceItem.hasValue() は false を返します。このメソッドで XBRLInstanceItem オブジェクトを生成したあと，XBRLInstanceItem.setValue() メソッドまたは XBRLInstanceItem.setString() メソッドを使用して適切な値を設定してください。

このメソッドで生成されたアイテムは，id 属性を持ちません。

引数：

namespaceURI - アイテムの名前空間 URI を指定します。

name - アイテムの名前を指定します。

context - アイテムが参照するコンテキスト要素を指定します。

戻り値：

新しく生成された XBRLInstanceItem オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 namespaceURI，name または context に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 namespaceURI および name で指定されたアイテムがタクソノミー文書で定義されていない場合にスローされます。［エラーコード：20044］
- 引数 namespaceURI および name で指定したアイテムが数値である場合にスローされます。［エラーコード：20132］
- 引数 namespaceURI および name で指定したアイテムがタクソノミー文書で抽象的な要素として定義されている場合にスローされます。［エラーコード：20065］
- 引数 context に指定されたオブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］
- 引数 context に指定したコンテキスト要素の時期要素の内容が，生成するアイテムの要素宣言での xbrli:periodType 属性の指定と矛盾している場合にスローされます。xbrli:periodType 属性の値が "instant" の場合は，時期要素の内容は XBRLPeriod.INSTANT である必要があります。また，xbrli:periodType 属性の値が "duration" の場合は，時期要素の内容は XBRLPeriod.FOREVER か XBRLPeriod.STARTDATE_ENDDATE である必要があります。［エラーコード：20140］

(59) createTuple

```
public XBRLInstanceTuple createTuple(java.lang.String namespaceURI,
                                     java.lang.String name)
```

XBRL のバージョン：

2.0 および 2.1

説明：

タプルを新規に生成します。

- XBRL2.0 の場合：
  タプルの id 属性の値は，このメソッドでタプルを生成する際に自動的に採番されます。自動採番機能の詳細については，「3.3　ID 管理と採番」を参照してください。また，id 属性値は，XBRLInstanceElement の setId() メソッドを使用して任意の値を設定することもできます。
- XBRL2.1 の場合：
  このメソッドで生成されたタプルは id 属性を持ちません。

引数：

namespaceURI - タプルの名前空間 URI を指定します。

name - タプルの名前を指定します。

戻り値：

新しく生成された XBRLInstanceTuple オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 namespaceURI または name に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 namespaceURI および name で指定したタプルがタクソノミー文書で定義されていない場合にスローされます。［エラーコード：20045］
- 引数 namespaceURI および name で指定されたタプルが，タクソノミー文書で抽象的な要素として定義されている場合にスローされます。［エラーコード：20065］

(60) createEntity

```
public XBRLEntity createEntity(XBRLIdentifier identifier,
                               XBRLSegment segment)
```

XBRL のバージョン：

2.0 および 2.1

説明：

エンティティ要素を表す XBRLEntity オブジェクトを新規に生成します。

引数：

identifier - 識別子要素を指定します。

segment - セグメント要素を指定します。セグメント要素がない場合は null を指定します。

戻り値：

新しく生成された XBRLEntity オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 identifier に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 identifier または segment に指定したオブジェクトが，すでにほかの XBRLEntity オブジェクトに関連づけられている場合にスローされます。［エラーコード：20005］
- 引数 identifier または segment に指定したオブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］

(61) createPeriod

```
public XBRLPeriod createPeriod(int periodType,
                               XBRLDateUnion instant,
                               XBRLDateUnion startDate,
                               XBRLDateUnion endDate,
```

```
                                    XBRLDuration duration)
```

XBRL のバージョン：

2.0 および 2.1

説明：

時期要素である XBRLPeriod オブジェクトを新規に生成します。

- XBRL2.1 の場合：
  xbrli:duration 要素を使用できないため，引数 periodType に XBRLPeriod.STARTDATE_DURATION および XBRLPeriod.DURATION_ENDDATE を指定できません。

開始日と終了日のチェック

引数 periodType に XBRLPeriod.STARTDATE_ENDDATE を指定した場合は，引数 endDate で指定された終了日は引数 startDate で指定された開始日よりも遅い日時である必要があります。終了日が開始日よりも遅い日時でない場合は，XBRLDOMRuntimeException がスローされます。このチェックを行う際に，開始日または終了日に時間の部分がない場合（つまり，XBRLDateUnion.getMemberType() メソッドが XBRLDateUnion.DATE を返す場合）は，次の規則が適用されます。

- 開始日に時間の部分がない場合は，日付の部分は同じで，時間の部分は 00:00:00.000 である日時が指定されたものとみなします。
- 終了日に時間の部分がない場合は，XBRL のバージョンによって次のとおりに処理します。
  XBRL2.0 の場合，日付の部分は同じで，時間の部分は 00:00:00.000 である日時が指定されたものとみなします。
  XBRL2.1 の場合，日付の部分は一日足した値で，時間の部分は 00:00:00.000 である日時が指定されたものとみなします。

終了日の判定は XBRL のバージョンによって異なります。このため，終了日が開始日よりも遅いかどうかの判定結果が，XBRL のバージョンによって異なる場合があります。例えば，開始日と終了日がともに 2004-04-01 である場合，開始日は 2004-04-01T00:00:00.000 とみなされますが，終了日は，XBRL2.0 の場合は 2004-04-01T00:00:00.000，XBRL2.1 の場合は 2004-04-02T00:00:00.000 とみなされます。この結果，XBRL2.0 の場合は終了日が開始日よりの遅い日時ではないためエラーとなりますが，XBRL2.1 では終了日は開始日よりも遅い日時となるためエラーとはなりません。

引数：

periodType - 時期要素の内容の意味を示すコードを指定します。

instant - ある特定の時点を指定します。ある特定の時点が不要な場合は null を指定します。

startDate - 開始日を指定します。開始日が不要な場合は null を指定します。

endDate - 終了日を指定します。終了日が不要な場合は null を指定します。

duration - 継続期間を指定します。継続期間が不要な場合は null を指定します。

戻り値：
新しく生成された XBRLPeriod オブジェクトを返します。

例外：
XBRLDOMRuntimeException -
- 引数の組み合わせが不正な場合にスローされます。引数 instant，startDate，endDate，duration の組み合わせは，引数 periodType の値によって異なります。詳細については，「6.6.22　XBRLPeriod インターフェース」を参照してください。［エラーコード：20056］
- 引数 periodType に XBRLPeriod.STARTDATE_ENDDATE を指定したときに，引数 startDate で指定した開始日と引数 endDate で指定した終了日の組み合わせが不正な場合にスローされます。終了日は開始日よりも遅い日付でなければなりません。［エラーコード：20057］
- 引数 periodType に不正な値を指定した場合にスローされます。引数 periodType に指定できる値は，XBRLPeriod.INSTANT，XBRLPeriod.FOREVER，XBRLPeriod.STARTDATE_DURATION，XBRLPeriod.DURATION_ENDDATE，XBRLPeriod.STARTDATE_ENDDATE のどれかです。［エラーコード：20014］
- 引数 periodType に XBRLPeriod.STARTDATE_DURATION または XBRLPeriod.DURATION_ENDDATE を指定し，かつ，引数 duration にゼロ（例：P0Y）や負数（例：-P1Y）を表す XBRLDuration オブジェクトを指定した場合にスローされます。［エラーコード：20071］
- XBRL2.1 の場合：引数 periodType に XBRLPeriod.STARTDATE_DURATION または XBRLPeriod.DURATION_ENDDATE を指定した場合にスローされます。［エラーコード：20142］

(62) createPeriod

```
public XBRLPeriod createPeriod()
```

XBRL のバージョン：
2.1

説明：
時期要素である XBRLPeriod オブジェクトを新規に生成します。
このメソッドで生成される時期要素の内容の意味は " 永遠 " です。つまり，XBRLPeriod.getPeriodType() メソッドは XBRLPeriod.FOREVER を返します。

戻り値：
新しく生成された XBRLPeriod オブジェクトを返します。

例外：
XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

(63) createPeriod

```
public XBRLPeriod createPeriod(XBRLDateUnion instant)
```

XBRL のバージョン：

2.1

説明：

時期要素である XBRLPeriod オブジェクトを新規に生成します。

このメソッドで生成される時期要素の内容の意味は " ある特定の時点 " です。つまり，XBRLPeriod.getPeriodType() メソッドは XBRLPeriod.INSTANT を返します。

引数：

instant - ある特定の時点を指定します。

戻り値：

新しく生成された XBRLPeriod オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 instant に null を指定した場合にスローされます。［エラーコード：20002］

(64) createPeriod

```
public XBRLPeriod createPeriod(XBRLDateUnion startDate,
                               XBRLDateUnion endDate)
```

XBRL のバージョン：

2.1

説明：

時期要素である XBRLPeriod オブジェクトを新規に生成します。

このメソッドで生成される時期要素の内容の意味は " 開始日，終了日 " です。つまり，XBRLPeriod.getPeriodType() メソッドは XBRLPeriod.STARTDATE_ENDDATE を返します。

引数 endDate で指定された終了日は引数 startDate で指定された開始日よりも遅い日時である必要があります。開始日と終了日のチェックに関する詳細については，「6.6.7(61)　createPeriod」を参照してください。

引数：

startDate - 開始日を指定します。

endDate - 終了日を指定します。

戻り値：
新しく生成された XBRLPeriod オブジェクトを返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 startDate または endDate に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 startDate で指定された開始日と引数 endDate で指定された終了日の組み合わせが不正な場合にスローされます。終了日は開始日よりも遅い日付である必要があります。［エラーコード：20057］

(65) createUnit

```
public XBRLUnit createUnit(XBRLMeasureOrOperator measureOrOperator)
```

XBRL のバージョン：
2.0

説明：
単位要素である XBRLUnit オブジェクトを新規に生成します。

引数：
measureOrOperator - 計量要素または演算子要素を指定します。

戻り値：
新しく生成された XBRLUnit オブジェクトを返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。［エラーコード：20100］
- 引数 measureOrOperator に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 measureOrOperator に指定したオブジェクトが，すでにほかの XBRLUnit オブジェクトに関連づけられている場合にスローされます。［エラーコード：20005］
- 引数 measureOrOperator に指定したオブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］

(66) createScenario

```
public XBRLScenario createScenario(java.lang.String name)
```

XBRL のバージョン：

2.0

説明：

シナリオ要素である XBRLScenario オブジェクトを新規に生成します。

引数：

name - name 属性の値を指定します。name 属性が不要な場合は null を指定します。

戻り値：

新しく生成された XBRLScenario オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。［エラーコード：20100］

(67) createScenario

```
public XBRLScenario createScenario()
```

XBRL のバージョン：

2.1

説明：

シナリオ要素である XBRLScenario オブジェクトを新規に生成します。

XBRL 仕様では，シナリオ要素は一つ以上の子要素を持つ必要があると定義されています。しかし，このメソッドでシナリオ要素を生成した直後は，シナリオ要素の内容は空です。このため，このメソッドでシナリオ要素を生成した場合は，必ず子要素を追加してください。シナリオ要素の内容は，XBRLScenario.getContainerNode() メソッドを呼び出して取得した org.w3c.dom.Node オブジェクトを使用して編集できます。詳細については，XBRLScenario インターフェースの「6.6.24(3)　getContainerNode」を参照してください。

戻り値：

新しく生成された XBRLScenario オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

(68) createIdentifier

```
public XBRLIdentifier createIdentifier(java.lang.String
namespaceURI,
                                       java.lang.String value)
```

XBRL のバージョン：

2.0 および 2.1

説明：
識別子要素である XBRLIdentifier オブジェクトを新規に生成します。

引数：
namespaceURI - スキーマの名前空間 URI を指定します。
value - 識別子を指定します。

戻り値：
新しく生成された XBRLIdentifier オブジェクトを返します。

例外：
XBRLDOMRuntimeException -
- 引数 namespaceURI または value に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 namespaceURI に指定した値が xsd:anyURI 型でない場合にスローされます。［エラーコード：20108］
- XBRL2.1 の場合：引数 namespaceURI に空文字列（""）を指定した場合にスローされます。［エラーコード：20110］

(69) createSegment

```
public XBRLSegment createSegment(java.lang.String name)
```

XBRL のバージョン：
2.0

説明：
セグメント要素である XBRLSegment オブジェクトを新規に生成します。

引数：
name - name 属性の値を指定します。name 属性が不要な場合は null を指定します。

戻り値：
新しく生成された XBRLSegment オブジェクトを返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。［エラーコード：20100］

(70) createSegment

```
public XBRLSegment createSegment()
```

XBRL のバージョン：
2.1

説明：

セグメント要素である XBRLSegment オブジェクトを新規に生成します。
XBRL 仕様では，セグメント要素は一つ以上の子要素を持つ必要があると定義されています。しかし，このメソッドでセグメント要素を生成した直後は，セグメント要素の内容は空です。このため，このメソッドでセグメント要素を生成した場合は，必ず子要素を追加してください。セグメント要素の内容は，XBRLSegment.getContainerNode() メソッドを呼び出して取得した org.w3c.dom.Node オブジェクトを使用して編集できます。詳細については，XBRLSegment インターフェースの「6.6.26(3)　getContainerNode」を参照してください。

戻り値：

新しく生成された XBRLSegment オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

(71) createMeasure

```
public XBRLMeasure createMeasure(XBRLQName value)
```

XBRL のバージョン：

2.0

説明：

計量要素である XBRLMeasure オブジェクトを新規に生成します。

引数：

value - 計量を指定します。

戻り値：

新しく生成された XBRLMeasure オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。［エラーコード：20100］
- 引数 value に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 value に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，この XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］
- 引数 value に指定した XBRLQName オブジェクトが名前空間に属さず，かつこの XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］

(72) createOperator

```
public XBRLOperator createOperator(java.lang.String kind,
                                   XBRLMeasureOrOperator lhs,
                                   XBRLMeasureOrOperator rhs)
```

XBRL のバージョン：

2.0

説明：

演算子要素である XBRLOperator オブジェクトを新規に生成します。

引数：

kind - 乗算の場合は XBRLOperator.MULTIPLY，除算の場合は XBRLOperator.DIVIDE を指定します。

lhs - 演算子の左辺を指定します。

rhs - 演算子の右辺を指定します。

戻り値：

新しく生成された XBRLOperator オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。[エラーコード：20100]
- 引数 kind，lhs または rhs に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 lhs または rhs に指定したオブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。[エラーコード：20021]
- 引数 lhs または rhs に指定したオブジェクトが，すでにほかの XBRLOperator オブジェクトに関連づけられている場合にスローされます。[エラーコード：20005]
- 引数 kind に，XBRLOperator.MULTIPLY または XBRLOperator.DIVIDE 以外の文字列を指定した場合にスローされます。[エラーコード：20014]
- 引数 lhs と引数 rhs に同じオブジェクトを指定した場合にスローされます。[エラーコード：20064]

(73) getFootnotes

```
public java.util.List<XBRLFootnote> getFootnotes()
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に記述されている脚注のリストを取得します。

XBRLInstanceDocument インターフェースの getFootnotes() では，文書中に含ま

れる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得する脚注のリストに含まれます。
しかし，XBRLInstanceElement インターフェースの getFootnotes() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得する脚注のリストには含まれません。
このメソッドが返す脚注のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる脚注に増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，
java.lang.UnsupportedOperationException がスローされます。

戻り値：

このインスタンス文書に存在する脚注をすべて含む，XBRLFootnote オブジェクトのリストを返します。このインスタンス文書に脚注がない場合は，要素数が 0 のリストを返します。

(74) getFootnotesByLang

```
public java.util.List<XBRLFootnote>
getFootnotesByLang(java.lang.String lang)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に記述されている脚注のうち，指定した言語の脚注を取得します。
XBRLInstanceDocument インターフェースの getFootnotesByLang() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得する脚注のリストに含まれます。
しかし，XBRLInstanceElement インターフェースの getFootnotesByLang() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得する脚注のリストには含まれません。
このメソッドが返す脚注のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる脚注に増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，
java.lang.UnsupportedOperationException がスローされます。

引数：

lang - 脚注の言語を指定します。

戻り値：

指定した言語の脚注をすべて含む XBRLFootnote オブジェクトのリストを返します。このインスタンス文書に指定した言語の脚注がない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(75) getFootnotesByRole

```
public java.util.List<XBRLFootnote>
getFootnotesByRole(java.lang.String role)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に記述されている脚注のうち，指定したロールの脚注を取得します。

XBRLInstanceDocument インターフェースの getFootnotesByRole() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得する脚注のリストに含まれます。

しかし，XBRLInstanceElement インターフェースの getFootnotesByRole() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得する脚注のリストには含まれません。

このメソッドが返す脚注のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる脚注に増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

role - 脚注のロールを指定します。ロールを持たない脚注を取得する場合は null を指定します。

戻り値：

指定したロールを持つ脚注をすべて含む XBRLFootnote オブジェクトのリストを返します。このインスタンス文書に指定したロールを持つ脚注が存在しない場合は，要素数が 0 のリストを返します。

(76) getFootnotes

```
public java.util.List<XBRLFootnote> getFootnotes(java.lang.String
lang,
                                    java.lang.String role)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に記述されている脚注のうち，指定した言語およびロールの脚注を取得します。

XBRLInstanceDocument インターフェースの getFootnotes() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得する脚注のリストに含まれます。
しかし，XBRLInstanceElement インターフェースの getFootnotes() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得する脚注のリストには含まれません。
このメソッドが返す脚注のリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる脚注に増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，
java.lang.UnsupportedOperationException がスローされます。

引数：

lang - 脚注の言語を指定します。
role - 脚注のロールを指定します。ロールを持たない脚注を取得する場合は null を指定します。

戻り値：

指定した言語とロールを持つ脚注をすべて含む，XBRLFootnote オブジェクトのリストを返します。このインスタンス文書に指定した言語とロールを持つ脚注がない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(77) addFootnote

```
public XBRLFootnote addFootnote(XBRLFootnote newFootnote)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に脚注を追加します。指定した XBRLFootnote オブジェクトがすでにこのインスタンス文書にある場合は，何もしません。

引数：

newFootnote - 追加する脚注を表す XBRLFootnote オブジェクトを指定します。

戻り値：

追加された脚注を表す XBRLFootnote オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 newFootnote に null を指定した場合にスローされます。［エラーコード：

20002］

- 引数 newFootnote に指定した XBRLFootnote オブジェクトを所有する XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではない場合にスローされます。［エラーコード：20021］
- XBRL2.1 の場合：引数 newFootnote に指定された脚注の id 属性と同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。［エラーコード：20028］

(78) removeFootnote

```
public XBRLFootnote removeFootnote(XBRLFootnote oldFootnote)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書から脚注を削除します。
脚注をインスタンス文書から削除すると，その脚注と要素の関連づけもすべて削除されます。削除されるのは関連づけであり，要素がインスタンス文書から削除されるわけではありません。

引数：

oldFootnote - 削除する脚注を表す XBRLFootnote オブジェクトを指定します。

戻り値：

削除された脚注を表す XBRLFootnote オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 oldFootnote に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 oldFootnote に指定した脚注が，このインスタンス文書にない場合にスローされます。［エラーコード：20046］

(79) createFootnote

```
public XBRLFootnote createFootnote(java.lang.String lang,
                                   java.lang.String role,
                                   java.lang.String title)
```

XBRL のバージョン：

2.0 および 2.1

説明：

脚注である XBRLFootnote オブジェクトを新規に生成します。

- XBRL2.1 の場合：
  脚注が属する脚注リンクのロールは，XBRL2.1 の仕様で規定されている標準値である "http://www.xbrl.org/2003/role/link" が仮定されます。脚注が属する脚注リン

クのロールとしてカスタムロールを指定したい場合は，
createFootnote(java.lang.String footnoteLinkRole, java.lang.String lang, java.lang.String role, java.lang.String title) メソッドを使用してください。

引数：

lang - 脚注の言語を指定します。
role - 脚注のロールを指定します。ロールが不要な場合は null を指定します。
title - 脚注のタイトルを指定します。タイトルが不要な場合は null を指定します。

戻り値：

新しく生成された XBRLFootnote オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 role に指定した値が xsd:anyURI 型でない場合にスローされます。［エラーコード：20108］
- 引数 role に指定した値が絶対 URI でない場合にスローされます。［エラーコード：20109］
- XBRL2.1 の場合：引数 role に指定したロールが XBRL 仕様で規定された標準値で，かつ，脚注で利用できるものではない場合にスローされます。［エラーコード：20115］
- XBRL2.1 の場合：引数 role に指定したロールがカスタムロールで，かつ，そのカスタムロールの定義を指す <link:roleRef> 要素がインスタンス文書に存在しない場合にスローされます。［エラーコード：20117］
- XBRL2.1 の場合：引数 role に指定したロールがカスタムロールで，かつ，<link:roleType> 要素でそのカスタムロールが脚注で利用できるように定義されていない場合にスローされます。［エラーコード：20119］

(80) createFootnote

```
public XBRLFootnote createFootnote(java.lang.String
footnoteLinkRole,
                                   java.lang.String lang,
                                   java.lang.String role,
                                   java.lang.String title)
```

XBRL のバージョン：

2.1

説明：

脚注である XBRLFootnote オブジェクトを新規に生成します。カスタムロールを指定できます。
このメソッドで生成された脚注は id 属性を持ちません。

引数：

footnoteLinkRole - 脚注が属する脚注リンクのロールを指定します。

lang - 脚注の言語を指定します。
role - 脚注のロールを指定します。ロールが不要な場合は null を指定します。
title - 脚注のタイトルを指定します。タイトルが不要な場合は null を指定します。

戻り値：
新しく生成された XBRLFootnote オブジェクトを返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 footnoteLinkRole または lang に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 footnoteLinkRole または role に指定した値が xsd:anyURI 型でない場合にスローされます。［エラーコード：20108］
- 引数 footnoteLinkRole または role に指定した値が絶対 URI でない場合にスローされます。［エラーコード：20109］
- 引数 footnoteLinkRole に指定したロールが XBRL 仕様で規定された標準値で，かつ，脚注リンクで利用できるものではない場合にスローされます。［エラーコード：20115］
- 引数 role に指定したロールが XBRL 仕様で規定された標準値で，かつ，脚注で利用できるものではない場合にスローされます。［エラーコード：20115］
- 引数 footnoteLinkRole または role に指定したロールがカスタムロールで，かつ，そのカスタムロールの定義を指す <link:roleRef> 要素がインスタンス文書に存在しない場合にスローされます。［エラーコード：20117］
- 引数 footnoteLinkRole に指定したロールがカスタムロールで，かつ，<link:roleType> 要素でそのカスタムロールが脚注リンクで利用できるように定義されていない場合にスローされます。［エラーコード：20119］
- 引数 role に指定したロールがカスタムロールで，かつ，<link:roleType> 要素でそのカスタムロールが脚注で利用できるように定義されていない場合にスローされます。［エラーコード：20119］

(81) importNode

```
public XBRLInstanceNode importNode(XBRLInstanceNode sourceNode,
                                   boolean deep)
```

XBRL のバージョン：
2.0 および 2.1

説明：
ほかの XBRLInstanceDocument オブジェクトで生成されたノードをインポートします。
このメソッドが返すノードはアクティブではなく，また，親ノードを持ちません。

つまり，このメソッドが返す XBRLInstanceNode オブジェクトの isActive() は false を返し，また，getParentNode() は null を返します。
引数 sourceNode に指定したノードを所有する XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトである場合は，importNode() メソッドの動作は，引数 sourceNode に対して cloneNode() メソッドを呼び出した場合と同じ動作となります。この場合の動作の詳細については，XBRLInstanceNode インターフェースの「6.6.14(5)　cloneNode」を参照してください。
ノードをインポートする処理は，ノードの型によって異なります。ノードのインポートを次の表に示します。

表 6-25　ノードのインポート

項番	ノードの型	説明
1	INSTANCE_DOCUMENT_NODE	この型のノードはインポートできません。インポートしようとすると例外が発生します。
2	SCHEMA_REF_NODE	サブツリーが存在しないため，引数 deep の値は影響しません。
3	LINKBASE_REF_NODE	
4	ROLE_REF_NODE	
5	ARCROLE_REF_NODE	
6	NUMERIC_CONTEXT_NODE	引数 deep の値に関係なく，常にサブツリーがインポートされます。 インポートされたコンテキスト要素の id 属性の値は採番し直されます。
7	NON_NUMERIC_CONTEXT_NODE	
8	CONTEXT2_NODE	
9	UNIT2_NODE	サブツリー※1 が存在しないため，引数 deep の値は影響しません。インポートされた単位要素の id 属性の値は採番し直されます。

項番	ノードの型	説明
10	ITEM_NODE	アイテムにはサブツリー※1が存在しないため，引数 deep の値は影響しません。 アイテムをインポートする際に，インポート元が参照しているコンテキスト要素も同時にインポートされます。 XBRL2.1 の場合：数値アイテムをインポートするときは，インポート元が参照している単位要素も同時にインポートされます。 インポート元のアイテムが脚注と関連づけられていた場合でも，その脚注は自動的にインポートされません。インポートされたアイテムはどの脚注とも関連づけされません。 アイテムをインポートする場合は，インポートしようとしたアイテムと同じ名前空間 URI および名前のアイテムが，インポート先のインスタンス文書が参照するタクソノミー文書で定義されている必要があります。同じ名前空間 URI および名前のアイテムが定義されていない場合は，例外が発生します。 また，同じ名前空間 URI および名前のアイテムが定義されている場合でも，データ型が異なるために，インポート元のアイテムの値をインポート先のアイテムへ設定できない場合は，例外が発生します。 XBRL2.1 の場合：インポート元でのアイテムの要素宣言と，インポート先でのアイテムの要素宣言は，次の定義内容が同じである必要があります。 • ベースデータ型 • xbrli:periodType 属性の値 インポート元とインポート先とで上記の定義内容が異なるアイテムをインポートしようとした場合は，例外が発生します。 XBRL2.1 の場合：インポート元のアイテムの xsi:nil 属性に "true" が指定されていて，かつ，インポート先のアイテムの要素宣言の nillable 属性に "true" が指定されていない場合は，インポート先のアイテムに xsi:nil 属性を設定できないため例外が発生します。 XBRL2.0 の場合：インポートされたアイテムの id 属性の値は採番し直されます。 XBRL2.1 の場合：インポートされたアイテムは id 属性を持ちません。

項番	ノードの型	説明
11	TUPLE_NODE	引数 deep が true の場合は，タプルの子要素もすべてインポートされます。 引数 deep が false の場合は，子要素はインポートされません。 タプルをインポートする場合は，インポートしようとしたタプルと同じ名前空間 URI および名前のタプルが，インポート先のインスタンス文書が参照するタクソノミー文書で定義されている必要があります。同じ名前空間 URI および名前のタプルが定義されていない場合は，例外が発生します。 XBRL2.1 の場合：インポート元のタプルの xsi:nil 属性に "true" が指定されていて，かつ，インポート先のタプルの要素宣言の nillable 属性に "true" が指定されていない場合は，インポート先のタプルに xsi:nil 属性を設定できないため例外が発生します。 XBRL2.0 の場合：インポートされたタプルの id 属性の値は採番し直されます。 XBRL2.1 の場合：インポートされたタプルは id 属性を持ちません。
12	ENTITY_NODE	引数 deep の値に関係なく，常にサブツリーがインポートされます。
13	PERIOD_NODE	
14	UNIT_NODE	
15	OPERATOR_NODE	
16	IDENTIFIER_NODE	サブツリー※1 がないため，引数 deep の値は影響しません。
17	SEGMENT_NODE	XBRL Dimensions1.0 ではない場合：サブツリー※1 が存在しないため，引数 deep の値は影響しません。 XBRL Dimensions1.0 の場合：引数 deep が true の場合は，セグメント要素に含まれている <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素がすべてインポートされます。 引数 deep が false の場合は，セグメント要素に含まれている <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素がすべてインポートされません。
18	MEASURE_NODE	サブツリー※1 がないため，引数 deep の値は影響しません。

項番	ノードの型	説明
19	SCENARIO_NODE	XBRL Dimensions1.0 ではない場合：サブツリー※1 が存在しないため，引数 deep の値は影響しません。 XBRL Dimensions1.0 の場合：引数 deep が true の場合は，シナリオ要素に含まれている <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素がすべてインポートされます。 引数 deep が false の場合は，シナリオ要素に含まれている <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素がすべてインポートされません。
20	FOOTNOTE_NODE	サブツリー※1 がないため，引数 deep の値は影響しません。 なお，インポート元の脚注がアイテムと関連づけられていた場合でも，そのアイテムは自動的にインポートされません。インポートされた脚注はどのアイテムとも関連づけられていない状態となります。 XBRL2.1 の場合：インポートされた脚注は id 属性を持ちません。
21	EXPLICIT_MEMBER_NODE ※2	サブツリー※1 が存在しないため，引数 deep の値は影響しません。
22	TYPED_MEMBER_NODE ※2	サブツリー※1 が存在しないため，引数 deep の値は影響しません。

注※1　ここでのサブツリーとは，XBRL DOM のノード（XBRLInstanceNode オブジェクト）によって構築されるツリーでのサブツリーのことです。W3C DOM のノード（org.w3c.dom.Node オブジェクト）によって構築されるツリーでのサブツリーではないので注意してください。
例えば，単位要素の子として記述される <xbrli:divide> 要素は，XBRL DOM では XBRLUnit2 オブジェクトの子ノードとしては表現されませんが，W3C DOM では <xbrli:divide> 要素は <xbrli:unit> 要素を表すノードの子ノードとして表現されます。そのため，XBRLUnit2 オブジェクトをインポートすると，W3C DOM ではサブツリーとなっている <xbrli:divide> 要素を表すノードやその子孫（<xbrli:unitNumerator> 要素を表すノードや <xbrli:unitDenominator> 要素を表すノード）もインポートされます。
これは，データ型が FRACTION_ITEM であるアイテム，XHTML のタグを内容として持つ脚注，任意の子要素を記述できるセグメント要素やシナリオ要素についても同じです。
注※2　XBRL Dimensions1.0 の場合だけ存在します。

引数：

sourceNode - インポートされるノードを表す XBRLInstanceNode オブジェクトを指定します。

deep - 引数 sourceNode に指定したノードのサブツリーをインポートする場合は true，サブツリーをインポートしない場合は false を指定します。

戻り値：

インポートされたノードを返します。

例外：

XBRLDOMRuntimeException -

- 引数 sourceNode に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 sourceNode に指定されたノードを所有する XBRLInstanceDocument オブジェクトが準拠している XBRL のバージョンと，この XBRLInstanceDocument オブジェクトが準拠している XBRL のバージョンが異なる場合にスローされます。［エラーコード：20104］
- 引数 sourceNode に指定したノードが，インポートできない型である場合にスローされます。［エラーコード：20048］
- XBRL2.1 の場合：引数 sourceNode に指定されたノードの型が ITEM_NODE で，かつ，インポート元でのアイテムの要素宣言とインポート先でのアイテムの要素宣言で，ベースデータ型または xbrli:periodType 属性の値が異なる場合にスローされます。［エラーコード：20145］

(82) cloneNode

```
public XBRLInstanceDocument cloneNode(org.w3c.dom.Document
w3cdomDocument,
                                      boolean deep)
```

XBRL のバージョン：

2.1

説明：

インスタンス文書を複製します。

引数 deep が true の場合は，インスタンス文書のルート要素の内容がすべて複製されます。引数 deep が false の場合は，インスタンス文書のルート要素の内容は複製されません。つまり，複製されたインスタンス文書は空です。ただし，名前空間宣言や xsi:schemaLocation 属性など，ルート要素の属性は常に複製されます。

ルート要素の挿入

複製された XBRLInstanceDocument オブジェクトのルート要素は，複製直後は DOM ツリーにつながっていない状態となっています。このため，ユーザーアプリケーションでは，W3C DOM の API を使用して，複製されたインスタンス文書のルート要素を引数 w3cdomDocument に指定した XML 文書の任意の位置へ挿入する必要があります。ただし，同一 XML 文書に記述されているほかのインスタンス文書の中に挿入してはいけません。インスタンス文書のルート要素を表す org.w3c.dom.Element オブジェクトは，XBRLInstanceDocument.getInstanceRoot() メソッドを使用して取得できます。

出発点となったタクソノミー本体およびリンクベースの指定

引数 deep が false の場合は，複製元の <link:schemaRef> 要素や <link:linkbaseRef> 要素は複製されません。このため，引数 deep に false を指

定してインスタンス文書を複製した場合は，タクソノミー文書を構築する際の出発点となったタクソノミー本体およびリンクベースのリストをXBRLTaxonomyDocument.getStartingPoints() メソッドによって取得し，出発点となったタクソノミー本体およびリンクベースを指す <link:schemaRef> 要素および <link:linkbaseRef> 要素をインスタンス文書に追加する必要があります。このとき，各要素の xlink:href 属性に指定する URI は，絶対 URI，相対 URI のどちらでもかまいません。ただし，インスタンス文書をストリームへ出力してから再度読み込んだ際に，それらの URI は適切な文書へ解決される必要があります。なお，必要な <link:schemaRef> 要素および <link:linkbaseRef> 要素を追加しなかった場合は，インスタンス文書をストリームへ出力してから再度読み込んだ場合に，タクソノミー文書を構成するタクソノミー本体またはリンクベースが不足するおそれがあるため，注意してください。

参照するタクソノミー文書

複製された XBRLInstanceDocument オブジェクトは，複製元の XBRLInstanceDocument オブジェクトが参照している XBRLTaxonomyDocument オブジェクトの複製を参照します。つまり，XBRLInstanceDocument.getTaxonomyDocument() メソッドの戻り値は，複製元の XBRLInstanceDocument オブジェクトと複製先の XBRLInstanceDocument オブジェクトで異なります。ただし，それぞれが参照する XBRLTaxonomyDocument オブジェクトの内容は同じです。

ルート要素の id 属性

複製されたインスタンス文書のルート要素は id 属性を持ちません。引数 deep に true が指定された場合は，インスタンス文書の内容が複製されますが，複製された各ノードが id 属性を持つかどうかについては，XBRLInstanceNode インターフェースの「6.6.14(5)　cloneNode」を参照してください。

XBRLIdCreator オブジェクトの引き継ぎ

インスタンス文書の ID を管理する XBRLIdManager オブジェクトは新規に作成されるため，複製元のインスタンス文書と複製されたインスタンス文書とで ID を管理する XBRLIdManager オブジェクトは異なります。ただし，XBRLIdManager オブジェクトに設定されている XBRLIdCreator オブジェクトは引き継がれます。つまり，複製元の XBRLInstanceDocument オブジェクトと複製された XBRLInstanceDocument オブジェクトとで，getIdManager() メソッドが返す XBRLIdManager オブジェクトは異なりますが，getIdManager().getIdCreator() によって取得できる XBRLIdCreator オブジェクトは同一のものとなります。このため，インスタンス文書を複製する際にコンテキスト要素などの id 属性の値が採番し直されますが，このときに複製元と同じ XBRLIdCreator オブジェクトを使用して ID が生成されることになります。

引数：

w3cdomDocument - インスタンス文書を挿入したい XML 文書を表す

org.w3c.dom.Document オブジェクトを指定します。
deep - ノードのサブツリーを複製する場合は true，サブツリーを複製しない場合は false を指定します。

戻り値：
このノードの複製を返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 w3cdomDocument に null を指定した場合にスローされます。［エラーコード：20002］

（83）createExplicitMember

```
public XBRLExplicitMember createExplicitMember(XBRLQName dimension,
                                               XBRLQName value)
```

XBRL バージョン：
2.1

XBRL Dimensions バージョン：
1.0

説明：
<xbrldi:explicitMember> 要素を表す XBRLExplicitMember オブジェクトを新規に生成します。

引数：
dimension - dimension 属性の値を指定します。
value - <xbrldi:explicitMember> 要素の内容を指定します。

戻り値：
新しく生成された XBRLExplicitMember オブジェクトを返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 dimension または引数 value に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 dimension または引数 value に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，この XBRLInstanceDocument オブジェク

トに登録されていない場合にスローされます。［エラーコード：20023］

- 引数 dimension または引数 value に指定された XBRLQName オブジェクトが名前空間に属さないで，かつ，この XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］
- 引数 dimension で指定したディメンジョンがタクソノミー文書で定義されていない場合にスローされます。［エラーコード：22106］
- 引数 value で指定したアイテムがタクソノミー文書で定義されていない場合にスローされます。［エラーコード：20044］

(84) createTypedMember

```
public XBRLTypedMember createTypedMember(XBRLQName dimension)
```

XBRL バージョン：

2.1

XBRL Dimensions バージョン：

1.0

説明：

<xbrldi:typedMember> 要素を表す XBRLTypedMember オブジェクトを新規に生成します。

引数：

dimension - dimension 属性の値を指定します。

戻り値：

新しく生成された XBRLTypedMember オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 dimension に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 dimension に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，この XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］
- 引数 dimension に指定した XBRLQName オブジェクトが名前空間に属さないで，かつ，XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］
- 引数 dimension で指定したディメンジョンがタクソノミー文書で定義されていな

い場合にスローされます。［エラーコード：22106］

（85）getDimensionsVersion

```
public int getDimensionsVersion()
```

XBRL バージョン：

2.1

説明：

インスタンス文書が準拠している XBRL Dimensions のバージョンを取得します。cbrp.parse.DimensionsVersion プロパティの値は，インスタンス文書が XBRL2.1 に準拠している場合だけ使用されます。インスタンス文書がほかの XBRL のバージョンに準拠している場合は，cbrp.parse.DimensionsVersion プロパティの値は使用されません。

戻り値：

このインスタンス文書が XBRL Dimensions1.0 に準拠している場合は XBRLConstants.DIMENSIONS_1_0 を返します。XBRL Dimensions1.0 に準拠していない場合は XBRLConstants.DIMENSIONS_NONE を返します。

6.6.8 XBRLInstanceDocumentBuilder インターフェース

```
public interface XBRLInstanceDocumentBuilder
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書から XBRLInstanceDocument オブジェクトを取得するための API を定義します。

メソッドの概要：

XBRLInstanceDocumentBuilder インターフェースのメソッドの概要を次の表に示します。

表 6-26 XBRLInstanceDocumentBuilder インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLDocumentManager	getOwnerDocumentManager() XBRLInstanceDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。	2.0 および 2.1
2	int	getXBRLVersion() XBRLInstanceDocumentBuilder オブジェクトが解析または作成するインスタンス文書の XBRL バージョンを取得します。	2.0 および 2.1

項番	説明		XBRL のバージョン
3	boolean	isValidating() XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時に文書の妥当性を検証するかどうかを調べます。	2.0 および 2.1
4	boolean	isDelayLoading() XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時にリンクベースの遅延読み込みをするかどうかを調べます。	2.0 および 2.1
5	java.util.List <java.lang.String>	getExternalSchemaRefs() XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時に参照するタクソノミー本体の URI のリストを取得します。	2.1
6	java.lang.Object	getProperty(java.lang.String name) XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析または作成する際に使用するプロパティの値を取得します。	2.1
7	XBRLTaxonomyDocumentBuilder	getTaxonomyDocumentBuilder() タクソノミー文書を読み込むために使用する XBRLTaxonomyDocumentBuilder オブジェクトを取得します。	2.0 および 2.1
8	void	setTaxonomyDocumentBuilder(XBRLTaxonomyDocumentBuilder taxonomyDocumentBuilder) タクソノミー文書を読み込むために使用する XBRLTaxonomyDocumentBuilder オブジェクトを設定します。	2.0 および 2.1
9	void	setEntityResolver(org.xml.sax.EntityResolver entityResolver) インスタンス文書に存在するエンティティを解決するためのエンティティリゾルバ（org.xml.sax.EntityResolver オブジェクト）を設定します。	2.0 および 2.1
10	void	setErrorHandler(XBRLParseErrorHandler errorHandler) インスタンス文書解析時のエラーを受け取るエラーハンドラ（XBRLParseErrorHandler オブジェクト）を設定します。	2.0 および 2.1
11	XBRLInstanceDocument	parse(org.xml.sax.InputSource is) 指定した入力ソースからインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。	2.0 および 2.1
12	XBRLInstanceDocument	parse(java.io.InputStream is) 指定した入力ストリームをインスタンス文書として解析し，XBRLInstanceDocument オブジェクトを新規に生成します。	2.0 および 2.1
13	XBRLInstanceDocument	parse(java.io.InputStream is, java.lang.String systemId) 指定した入力ストリームをインスタンス文書として解析し，XBRLInstanceDocument オブジェクトを新規に生成します。	2.0 および 2.1
14	XBRLInstanceDocument	parse(java.io.File file) 指定したファイルをインスタンス文書として解析し，XBRLInstanceDocument オブジェクトを新規に生成します。	2.0 および 2.1
15	XBRLInstanceDocument	parse(java.lang.String uri) 指定した URI をインスタンス文書として解析し，XBRLInstanceDocument オブジェクトを新規に生成します。	2.0 および 2.1

項番	説明		XBRL のバージョン
16	XBRLInstanceDocument	parse(org.w3c.dom.Element instanceRoot) 指定した要素をルート要素とするインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。	2.1
17	XBRLInstanceDocument	parse(org.w3c.dom.Element instanceRoot, java.lang.String systemId) 指定した要素をルート要素とするインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。	2.1
18	XBRLInstanceDocument[]	parse2(org.xml.sax.InputSource is) 指定した入力ソースに含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。	2.1
19	XBRLInstanceDocument[]	parse2(java.io.InputStream is) 指定した入力ストリームに含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。	2.1
20	XBRLInstanceDocument[]	parse2(java.io.InputStream is, java.lang.String systemId) 指定した入力ストリームに含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。	2.1
21	XBRLInstanceDocument[]	parse2(java.io.File file) 指定したファイルに含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。	2.1
22	XBRLInstanceDocument[]	parse2(java.lang.String uri) 指定した URI に含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。	2.1
23	XBRLInstanceDocument	newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument) 空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを新規に生成します。	2.0 および 2.1
24	XBRLInstanceDocument	newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument, java.lang.String systemId) 空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを新規に生成します。	2.1
25	XBRLInstanceDocument	newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument, org.w3c.dom.Document w3cdomDocument) XML 文書の任意の位置にインスタンス文書を追加する場合に使用します。空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを新規に生成します。	2.1

項番	説明		XBRL のバージョン
26	XBRLInstanceDocument	newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument, org.w3c.dom.Document w3cdomDocument, java.lang.String systemId) XML 文書の任意の位置にインスタンス文書を追加する場合に使用します。空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを新規に生成します。	2.1
27	int	getDimensionsVersion() XBRLInstanceDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているインスタンス文書を解析または作成するのかを取得します。	2.1
28	void	setCreatingSchemaLocations(boolean creating) 空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを生成するときに，ルート要素の xsi:schemaLocation 属性の初期値として，タクソノミー文書に含まれる各タクソノミー本体の対象名前空間 URI と位置を示す絶対 URI の組を設定するかどうかを指定します。	2.1
29	boolean	isCreatingSchemaLocations() 空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを生成するときに，ルート要素の xsi:schemaLocation 属性の初期値として，タクソノミー文書に含まれる各タクソノミー本体の対象名前空間 URI と位置を示す絶対 URI の組を設定するかどうかを示します。	2.1

(1) getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLInstanceDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

戻り値：

この XBRLInstanceDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトを返します。

(2) getXBRLVersion

```
public int getXBRLVersion()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLInstanceDocumentBuilder オブジェクトが解析または作成するインスタンス

文書の XBRL バージョンを取得します。
このメソッドが返す設定値は，この XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析または作成する際に使用する cbrp.parse.XBRLVersion プロパティの値となります。
この XBRLInstanceDocumentBuilder オブジェクトが使用する cbrp.parse.XBRLVersion プロパティの値は，この XBRLInstanceDocumentBuilder オブジェクトを生成した XBRLInstanceDocumentBuilderFactory オブジェクトの setXBRLVersion() メソッド，または setProperty() メソッドで設定された値です。
XBRLInstanceDocumentBuilderFactory オブジェクトで値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。

戻り値：
XBRL2.0 に準拠したインスタンス文書を解析または作成する場合は XBRLConstants.XBRL_2_0，XBRL2.1 に準拠したインスタンス文書を解析または作成する場合は XBRLConstants.XBRL_2_1 を返します。

（3） isValidating

```
public boolean isValidating()
```

XBRL のバージョン：
2.0 および 2.1

説明：
XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時に文書の妥当性を検証するかどうかを調べます。
このメソッドが返す設定値は，この XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.validate プロパティの値です。
この XBRLInstanceDocumentBuilder オブジェクトが使用する cbrp.parse.validate プロパティの値は，この XBRLInstanceDocumentBuilder オブジェクトを生成した XBRLInstanceDocumentBuilderFactory オブジェクトの setValidating() メソッド，または setProperty() メソッドで設定された値です。
XBRLInstanceDocumentBuilderFactory オブジェクトで値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。

戻り値：
インスタンス文書の解析時に妥当性を検証する場合は true，検証しない場合は false を返します。

（4） isDelayLoading

```
public boolean isDelayLoading()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時にリンクベースの遅延読み込みをするかどうかを調べます。
このメソッドが返す設定値は，この XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.delayLoad プロパティの値です。
この XBRLInstanceDocumentBuilder オブジェクトが使用する cbrp.parse.delayLoad プロパティの値は，この XBRLInstanceDocumentBuilder オブジェクトを生成した XBRLInstanceDocumentBuilderFactory オブジェクトの setDelayLoading() メソッド，または setProperty() メソッドで設定された値です。
XBRLInstanceDocumentBuilderFactory オブジェクトで値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。

戻り値：

インスタンス文書に記述されているリンクの解析を遅延する場合は true，遅延しない場合は false を返します。

(5) getExternalSchemaRefs

```
public java.util.List<java.lang.String> getExternalSchemaRefs()
```

XBRL のバージョン：

2.1

説明：

XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時に参照するタクソノミー本体の URI のリストを取得します。
このメソッドが返す設定値は，この XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.DTS.externalSchemaRefs プロパティの値です。
この XBRLInstanceDocumentBuilder オブジェクトが使用する cbrp.parse.DTS.externalSchemaRefs プロパティの値は，この XBRLInstanceDocumentBuilder オブジェクトを生成した XBRLInstanceDocumentBuilderFactory オブジェクトの setExternalSchemaRefs() メソッド，または setProperty() メソッドで設定された値です。
XBRLInstanceDocumentBuilderFactory オブジェクトで値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。
このメソッドが返す設定値は，解析対象のインスタンス文書が XBRL2.1 に準拠している場合にだけ有効です。解析対象のインスタンス文書がほかのバージョンの

XBRL 仕様に準拠しているものである場合は，このメソッドが返す設定値は使用されません。
このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：
インスタンス文書の解析時に参照するタクソノミー本体の URI を表す文字列のリストを返します。インスタンス文書に記述されている <link:schemaRef> 要素だけを使用して解析処理を行うように設定されている場合は，要素数が 0 のリストを返します。

(6) getProperty

```
public java.lang.Object getProperty(java.lang.String name)
```

XBRL のバージョン：
2.1

説明：
XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析または作成する際に使用するプロパティの値を取得します。このメソッドで値を取得できるプロパティの一覧およびこのメソッドがプロパティの値として返すオブジェクトの詳細については，「4.16.2　プロパティ一覧」を参照してください。

引数：
name - プロパティの名前を指定します。

戻り値：
プロパティの値を返します。

例外：
XBRLDOMRuntimeException -
- 引数 name に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 name に指定したプロパティの名前が正しくない場合にスローされます。［エラーコード：20150］

(7) getTaxonomyDocumentBuilder

```
public XBRLTaxonomyDocumentBuilder getTaxonomyDocumentBuilder()
```

XBRL のバージョン：
2.0 および 2.1

説明：
タクソノミー文書を読み込むために使用する XBRLTaxonomyDocumentBuilder オブジェクトを取得します。
このメソッドが返す設定値は，この XBRLInstanceDocumentBuilder オブジェクトの setTaxonomyDocumentBuilder() メソッドで設定された値となります。

setTaxonomyDocumentBuilder() メソッドの引数に null が指定された場合，または，setTaxonomyDocumentBuilder() メソッドが呼び出されなかった場合は，null を返します。

戻り値：

タクソノミー文書を読み込むために使用する XBRLTaxonomyDocumentBuilder オブジェクトを返します。null を返した場合は，this.getOwnerDocumentManager().newTaxonomyDocumentBuilderFactory().newTaxonomyDocumentBuilder() によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが使用されることを意味します。

（8） setTaxonomyDocumentBuilder

```
public void setTaxonomyDocumentBuilder(XBRLTaxonomyDocumentBuilder
taxonomyDocumentBuilder)
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書を読み込むために使用する XBRLTaxonomyDocumentBuilder オブジェクトを設定します。

このメソッドで設定された XBRLTaxonomyDocumentBuilder オブジェクトは，次の場合に使用されることがあります。

- この XBRLInstanceDocumentBuilder オブジェクトの parse() メソッドを呼び出してインスタンス文書を解析する際に，そのインスタンス文書が参照しているタクソノミー文書を解析する場合
- この XBRLInstanceDocumentBuilder オブジェクトによって生成された XBRLInstanceDocument オブジェクトに <link:linkbaseRef> 要素を追加する場合
- XBRL2.1 の場合：この XBRLInstanceDocumentBuilder オブジェクトによって生成された XBRLInstanceDocument オブジェクトに <link:schemaRef> 要素，<link:roleRef> 要素，および，<link:arcroleRef> 要素を追加する場合

このメソッドを呼び出さなかった場合，もしくは引数 taxonomyDocumentBuilder に null を指定した場合は，this.getOwnerDocumentManager().newTaxonomyDocumentBuilderFactory().newTaxonomyDocumentBuilder() によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが使用されます。このとき，エンティティリゾルバ（org.xml.sax.EntityResolver），エラーハンドラ（XBRLParseErrorHandler），および setValidating() などのメソッドで設定される各種プロパティの値は，この XBRLInstanceDocumentBuilder オブジェクトに設定されたものが引き継がれます。

引数：

taxonomyDocumentBuilder - タクソノミー文書を読み込むために使用する XBRLTaxonomyDocumentBuilder オブジェクトを指定します。null を指定すると，this.getOwnerDocumentManager().newTaxonomyDocumentBuilderFactory().newTaxonomyDocumentBuilder() によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが使用されます。

例外：

XBRLDOMRuntimeException -

- 引数 taxonomyDocumentBuilder に指定した XBRLTaxonomyDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトが，この XBRLInstanceDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトと異なる場合にスローされます。［エラーコード：20051］
- 引数 taxonomyDocumentBuilder に指定された XBRLTaxonomyDocumentBuilder オブジェクトが処理対象とする XBRL のバージョンと，この XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL のバージョンが異なる場合にスローされます。［エラーコード：20102］
- XBRL2.1 の場合：この XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書の解析時に完全な妥当性検証をするように設定されていて，かつ引数 taxonomyDocumentBuilder に指定された XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書の解析時に完全な妥当性検証をするように設定されていない場合にスローされます。［エラーコード：20149］
- 引数 taxonomyDocumentBuilder に指定した XBRLTaxonomyDocumentBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョンと，この XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョンが異なる場合にスローされます。［エラーコード：22101］

（9） setEntityResolver

```
public void setEntityResolver(org.xml.sax.EntityResolver
entityResolver)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に存在するエンティティを解決するためのエンティティリゾルバ（org.xml.sax.EntityResolver オブジェクト）を設定します。

setTaxonomyDocumentBuilder() メソッドを呼び出さなかったか，または setTaxonomyDocumentBuilder() メソッドの引数に null を指定した場合は，タクソノミー文書を解析する際に，エンティティを解決するため，このメソッドで設定した org.xml.sax.EntityResolver オブジェクトが使用されます。タクソノミー文書を解析するために使用される org.xml.sax.EntityResolver オブジェクトでは，

resolveEntity() メソッドの戻り値として返す org.xml.sax.InputSource オブジェクトに，必ずシステム識別子を設定しなければなりません。詳細については，XBRLTaxonomyDocumentBuilder インターフェースの「6.8.14(8) setEntityResolver」を参照してください。

引数：

entityResolver - 解析するインスタンス文書に存在するエンティティを解決するために使用される，org.xml.sax.EntityResolver オブジェクトを指定します。null を指定すると，デフォルトの実装が使用されます。

(10) setErrorHandler

```
public void setErrorHandler(XBRLParseErrorHandler errorHandler)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書解析時のエラーを受け取るエラーハンドラ（XBRLParseErrorHandler オブジェクト）を設定します。
引数 errorHandler に null を指定した場合は，XBRLParseErrorHandler インターフェースのデフォルトの実装が使用されます。デフォルトの実装では，warning()，error() の両メソッドが呼び出された場合は何もしないため，解析処理が続行されます。また，fatalError() メソッドが呼び出された場合は，fatalError() メソッドの引数に指定した例外をスローするため解析処理が中断されます。

引数：

errorHandler - 解析するインスタンス文書に存在するエラーのレポートを受け取る，XBRLParseErrorHandler オブジェクトを指定します。null を指定すると，デフォルトの実装が使用されます。

(11) parse

```
public XBRLInstanceDocument parse(org.xml.sax.InputSource is)
                            throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した入力ソースからインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

- XBRL2.0 の場合：
  解析対象のインスタンス文書のルート要素は，XML 文書のルート要素である必要があります。解析対象のインスタンス文書のルート要素が XML 文書のルート要素でない場合は，致命的エラーが発生します。
- XBRL2.1 の場合：

解析対象のインスタンス文書のルート要素は，XML 文書のルート要素でなくてもかまいません。ただし，XML 文書中に複数のインスタンス文書が含まれている場合は，最初の一つだけが解析され，二つ目以降のインスタンス文書は解析されません。XML 文書中に含まれるすべてのインスタンス文書を解析したい場合は，parse2() メソッドを使用してください。

引数：

is - 入力ソースを表す org.xml.sax.InputSource オブジェクトを指定します。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトを返します。

例外：

XBRLDOMException -

- インスタンス文書の解析処理で，エラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 引数 is に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 is に指定された org.xml.sax.InputSource オブジェクトに設定されているシステム識別子の形式が不正な場合にスローされます。［エラーコード：20147］

(12) parse

```
public XBRLInstanceDocument parse(java.io.InputStream is)
                            throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した入力ストリームをインスタンス文書として解析し，
XBRLInstanceDocument オブジェクトを新規に生成します。

- XBRL2.0 の場合：
  解析対象のインスタンス文書のルート要素は，XML 文書のルート要素である必要があります。解析対象のインスタンス文書のルート要素が XML 文書のルート要素でない場合は，致命的エラーが発生します。
- XBRL2.1 の場合：
  解析対象のインスタンス文書のルート要素は，XML 文書のルート要素でなくてもかまいません。ただし，XML 文書中に複数のインスタンス文書が含まれている場合は，最初の一つだけが解析され，二つ目以降のインスタンス文書は解析されません。XML 文書中に含まれるすべてのインスタンス文書を解析したい場合は，parse2() メソッドを使用してください。

引数：

is - インスタンス文書を含む入力ストリームを表す java.io.InputStream オブジェクトを指定します。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトを返します。

例外：

XBRLDOMException -

- インスタンス文書の解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 引数 is に null を指定した場合にスローされます。［エラーコード：20002］

(13) parse

```
public XBRLInstanceDocument parse(java.io.InputStream is,
                                  java.lang.String systemId)
                            throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した入力ストリームをインスタンス文書として解析し，
XBRLInstanceDocument オブジェクトを新規に生成します。

- XBRL2.0 の場合：
  解析対象のインスタンス文書のルート要素は，XML 文書のルート要素である必要があります。解析対象のインスタンス文書のルート要素が XML 文書のルート要素でない場合は，致命的エラーが発生します。
- XBRL2.1 の場合：
  解析対象のインスタンス文書のルート要素は，XML 文書のルート要素でなくてもかまいません。ただし，XML 文書中に複数のインスタンス文書が含まれている場合は，最初の一つだけが解析され，二つ目以降のインスタンス文書は解析されません。XML 文書中に含まれるすべてのインスタンス文書を解析したい場合は，parse2() メソッドを使用してください。

引数：

is - インスタンス文書を含む入力ストリームを表す java.io.InputStream オブジェクトを指定します。
systemId - インスタンス文書のシステム識別子を指定します。システム識別子は，相対 URI を解決するために使用されます。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトを返します。

例外：

XBRLDOMException -

- インスタンス文書の解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 引数 is または systemId に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 systemId に指定されたシステム識別子の形式が不正な場合にスローされます。［エラーコード：20147］

(14) parse

```
public XBRLInstanceDocument parse(java.io.File file)
                            throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定したファイルをインスタンス文書として解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

- XBRL2.0 の場合：
  解析対象のインスタンス文書のルート要素は，XML 文書のルート要素である必要があります。解析対象のインスタンス文書のルート要素が XML 文書のルート要素でない場合は，致命的エラーが発生します。
- XBRL2.1 の場合：
  解析対象のインスタンス文書のルート要素は，XML 文書のルート要素でなくてもかまいません。ただし，XML 文書中に複数のインスタンス文書が含まれている場合は，最初の一つだけが解析され，二つ目以降のインスタンス文書は解析されません。XML 文書中に含まれるすべてのインスタンス文書を解析したい場合は，parse2() メソッドを使用してください。

引数：

file - インスタンス文書を含むファイルを表す java.io.File オブジェクトを指定します。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトを返します。

例外：

XBRLDOMException -

- インスタンス文書の解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 引数 file に null を指定した場合にスローされます。［エラーコード：20002］

(15) parse

```
public XBRLInstanceDocument parse(java.lang.String uri)
                            throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した URI をインスタンス文書として解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

- XBRL2.0 の場合：
  解析対象のインスタンス文書のルート要素は，XML 文書のルート要素である必要があります。解析対象のインスタンス文書のルート要素が XML 文書のルート要素でない場合は，致命的エラーが発生します。
- XBRL2.1 の場合：
  解析対象のインスタンス文書のルート要素は，XML 文書のルート要素でなくてもかまいません。ただし，XML 文書中に複数のインスタンス文書が含まれている場合は，最初の一つだけが解析され，二つ目以降のインスタンス文書は解析されません。XML 文書中に含まれるすべてのインスタンス文書を解析したい場合は，parse2() メソッドを使用してください。

引数：

uri - インスタンス文書の場所を示す URI を指定します。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトを返します。

例外：

XBRLDOMException -

- インスタンス文書の解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 引数 uri に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 uri に指定された URI の形式が不正な場合にスローされます。［エラーコー

ド：20147］

（16）parse

```
public XBRLInstanceDocument parse(org.w3c.dom.Element instanceRoot)
                            throws XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

指定した要素をルート要素とするインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

このメソッドでインスタンス文書を解析する場合は，DOM ツリーはすでに生成されているため，XML プロセッサによる解析処理は実行されません。このため，この XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書の解析時に妥当性の検証を行うように設定されている場合（isValidating() メソッドが true を返す場合）でも，XML Schema による妥当性の検証はされません。ただし，uCosminexus Business Reporting Processor 独自の検証処理は実行されます。

引数：

instanceRoot - インスタンス文書のルート要素を表す org.w3c.dom.Element オブジェクトを指定します。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトを返します。

例外：

XBRLDOMException -

- インスタンス文書の解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］

  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- この XBRLInstanceDocumentBuilder オブジェクトが，バージョンが 2.1 である XBRL 仕様に準拠したインスタンス文書を解析するように設定されていない場合にスローされます。［エラーコード：20101］
- 引数 instanceRoot に null を指定した場合にスローされます。［エラーコード：20002］

（17）parse

```
public XBRLInstanceDocument parse(org.w3c.dom.Element instanceRoot,
                                  java.lang.String systemId)
                            throws XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

指定した要素をルート要素とするインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。
このメソッドでインスタンス文書を解析する場合は，DOM ツリーはすでに生成されているため，XML プロセッサによる解析処理は実行されません。このため，この XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書の解析時に妥当性の検証を行うように設定されている場合（isValidating() メソッドが true を返す場合）でも，XML Schema による妥当性の検証はされません。ただし，uCosminexus Business Reporting Processor 独自の検証処理は実行されます。

引数：

instanceRoot - インスタンス文書のルート要素を表す org.w3c.dom.Element オブジェクトを指定します。
systemId インスタンス文書のシステム識別子を指定します。システム識別子は，相対 URI を解決するために使用されます。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトを返します。

例外：

XBRLDOMException -

- インスタンス文書の解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- この XBRLInstanceDocumentBuilder オブジェクトが，バージョンが 2.1 である XBRL 仕様に準拠したインスタンス文書を解析するように設定されていない場合にスローされます。［エラーコード：20101］
- 引数 instanceRoot または systemId に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 systemId に指定されたシステム識別子の形式が不正な場合にスローされます。［エラーコード：20147］

(18) parse2

```
public XBRLInstanceDocument[] parse2(org.xml.sax.InputSource is)
                              throws XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

指定した入力ソースに含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。
解析対象のインスタンス文書のルート要素は，XML 文書のルート要素でなくてもかまいません。また，XML 文書中に複数のインスタンス文書を含めることができます。XML 文書中に複数のインスタンス文書が含まれている場合は，含まれているインスタンス文書をすべて解析して XBRLInstanceDocument オブジェクトとして返します。ただし，XML 文書中にインスタンス文書が一つも含まれていない場合は，致命的エラーが発生します。

ID の一元管理

このメソッドが複数の XBRLInstanceDocument オブジェクトを返した場合，インスタンス文書に含まれる ID は，XBRLInstanceDocument オブジェクトごとに別々の XBRLIdManager オブジェクトで管理されます。つまり，XBRLInstanceDocument.getIdManager() メソッドが返す XBRLIdManager オブジェクトは，XBRLInstanceDocument オブジェクトごとに異なります。そのため，ID を一つの XBRLIdManager オブジェクトで一元管理したい場合は，XBRLIdManager.registerDocument(XBRLInstanceDocument doc) メソッドを使用して，各 XBRLInstanceDocument オブジェクトを同じ XBRLIdManager オブジェクトへ登録する必要があります。

引数：

is - インスタンス文書を含む入力ソースを表す org.xml.sax.InputSource オブジェクトを指定します。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトの配列を返します。

例外：

XBRLDOMException -

- インスタンス文書の解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- この XBRLInstanceDocumentBuilder オブジェクトが，バージョンが 2.1 である XBRL 仕様に準拠したインスタンス文書を解析するように設定されていない場合にスローされます。［エラーコード：20101］
- 引数 is に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 is に指定された org.xml.sax.InputSource オブジェクトに設定されているシステム識別子の形式が不正な場合にスローされます。［エラーコード：20147］

（19）parse2

```
public XBRLInstanceDocument[] parse2(java.io.InputStream is)
```

```
                                    throws XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

指定した入力ストリームに含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

解析対象のインスタンス文書のルート要素は，XML 文書のルート要素でなくてもかまいません。また，XML 文書中に複数のインスタンス文書を含めることができます。XML 文書中に複数のインスタンス文書が含まれている場合は，含まれているインスタンス文書をすべて解析して XBRLInstanceDocument オブジェクトとして返します。ただし，XML 文書中にインスタンス文書が一つも含まれていない場合は，致命的エラーが発生します。

ID の一元管理

このメソッドが複数の XBRLInstanceDocument オブジェクトを返した場合，インスタンス文書に含まれる ID は，XBRLInstanceDocument オブジェクトごとに別々の XBRLIdManager オブジェクトで管理されます。つまり，XBRLInstanceDocument.getIdManager() メソッドが返す XBRLIdManager オブジェクトは，XBRLInstanceDocument オブジェクトごとに異なります。そのため，ID を一つの XBRLIdManager オブジェクトで一元管理したい場合は，XBRLIdManager.registerDocument(XBRLInstanceDocument doc) メソッドを使用して，各 XBRLInstanceDocument オブジェクトを同じ XBRLIdManager オブジェクトへ登録する必要があります。

引数：

is - インスタンス文書を含む入力ストリームを表す java.io.InputStream オブジェクトを指定します。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトの配列を返します。

例外：

XBRLDOMException -

- インスタンス文書の解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］

  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- この XBRLInstanceDocumentBuilder オブジェクトが，バージョンが 2.1 である XBRL 仕様に準拠したインスタンス文書を解析するように設定されていない場合にスローされます。［エラーコード：20101］
- 引数 is に null を指定した場合にスローされます。［エラーコード：20002］

(20) parse2

```
public XBRLInstanceDocument[] parse2(java.io.InputStream is,
                                     java.lang.String systemId)
                              throws XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

指定した入力ストリームに含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

解析対象のインスタンス文書のルート要素は，XML 文書のルート要素でなくてもかまいません。また，XML 文書中に複数のインスタンス文書を含めることができます。XML 文書中に複数のインスタンス文書が含まれている場合は，含まれているインスタンス文書をすべて解析して XBRLInstanceDocument オブジェクトとして返します。ただし，XML 文書中にインスタンス文書が一つも含まれていない場合は，致命的エラーが発生します。

ID の一元管理

このメソッドが複数の XBRLInstanceDocument オブジェクトを返した場合，インスタンス文書に含まれる ID は，XBRLInstanceDocument オブジェクトごとに別々の XBRLIdManager オブジェクトで管理されます。つまり，XBRLInstanceDocument.getIdManager() メソッドが返す XBRLIdManager オブジェクトは，XBRLInstanceDocument オブジェクトごとに異なります。そのため，ID を一つの XBRLIdManager オブジェクトで一元管理したい場合は，XBRLIdManager.registerDocument(XBRLInstanceDocument doc) メソッドを使用して，各 XBRLInstanceDocument オブジェクトを同じ XBRLIdManager オブジェクトへ登録する必要があります。

引数：

is - インスタンス文書を含む入力ストリームを表す java.io.InputStream オブジェクトを指定します。

systemId - インスタンス文書を含む XML 文書のシステム識別子を指定します。システム識別子は，相対 URI を解決するために使用されます。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトの配列を返します。

例外：

XBRLDOMException -

- インスタンス文書の解析処理でエラーが検出された場合にスローされます。[エラーコード：エラー要因によって異なる※]

  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- この XBRLInstanceDocumentBuilder オブジェクトが，バージョンが 2.1 である XBRL 仕様に準拠したインスタンス文書を解析するように設定されていない場合にスローされます。［エラーコード：20101］
- 引数 is または systemId に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 systemId に指定されたシステム識別子の形式が不正な場合にスローされます。［エラーコード：20147］

(21) parse2

```
public XBRLInstanceDocument[] parse2(java.io.File file)
                              throws XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

指定したファイルに含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。

解析対象のインスタンス文書のルート要素は，XML 文書のルート要素でなくてもかまいません。また，XML 文書中に複数のインスタンス文書を含めることができます。XML 文書中に複数のインスタンス文書が含まれている場合は，含まれているインスタンス文書をすべて解析して XBRLInstanceDocument オブジェクトとして返します。ただし，XML 文書中にインスタンス文書が一つも含まれていない場合は，致命的エラーが発生します。

ID の一元管理

このメソッドが複数の XBRLInstanceDocument オブジェクトを返した場合，インスタンス文書に含まれる ID は，XBRLInstanceDocument オブジェクトごとに別々の XBRLIdManager オブジェクトで管理されます。つまり，XBRLInstanceDocument.getIdManager() メソッドが返す XBRLIdManager オブジェクトは，XBRLInstanceDocument オブジェクトごとに異なります。そのため，ID を一つの XBRLIdManager オブジェクトで一元管理したい場合は，XBRLIdManager.registerDocument(XBRLInstanceDocument doc) メソッドを使用して，各 XBRLInstanceDocument オブジェクトを同じ XBRLIdManager オブジェクトへ登録する必要があります。

引数：

file - インスタンス文書を含むファイルを表す java.io.File オブジェクトを指定します。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトの配列を返します。

例外：

XBRLDOMException -

- インスタンス文書の解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7. メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- この XBRLInstanceDocumentBuilder オブジェクトが，バージョンが 2.1 である XBRL 仕様に準拠したインスタンス文書を解析するように設定されていない場合にスローされます。［エラーコード：20101］
- 引数 file に null を指定した場合にスローされます。［エラーコード：20002］

(22) parse2

```
public XBRLInstanceDocument[] parse2(java.lang.String uri)
                              throws XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

指定した URI に含まれる複数のインスタンス文書を解析し，XBRLInstanceDocument オブジェクトを新規に生成します。
解析対象のインスタンス文書のルート要素は，XML 文書のルート要素でなくてもかまいません。また，XML 文書中に複数のインスタンス文書を含めることができます。XML 文書中に複数のインスタンス文書が含まれている場合は，含まれているインスタンス文書をすべて解析して XBRLInstanceDocument オブジェクトとして返します。ただし，XML 文書中にインスタンス文書が一つも含まれていない場合は，致命的エラーが発生します。

ID の一元管理

このメソッドが複数の XBRLInstanceDocument オブジェクトを返した場合，インスタンス文書に含まれる ID は，XBRLInstanceDocument オブジェクトごとに別々の XBRLIdManager オブジェクトで管理されます。つまり，XBRLInstanceDocument.getIdManager() メソッドが返す XBRLIdManager オブジェクトは，XBRLInstanceDocument オブジェクトごとに異なります。そのため，ID を一つの XBRLIdManager オブジェクトで一元管理したい場合は，XBRLIdManager.registerDocument(XBRLInstanceDocument doc) メソッドを使用して，各 XBRLInstanceDocument オブジェクトを同じ XBRLIdManager オブジェクトへ登録する必要があります。

引数：

uri - インスタンス文書を含む XML 文書の場所を示す URI を指定します。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトの配列を返します。

例外：

XBRLDOMException -

- インスタンス文書の解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- この XBRLInstanceDocumentBuilder オブジェクトが，バージョンが 2.1 である XBRL 仕様に準拠したインスタンス文書を解析するように設定されていない場合にスローされます。［エラーコード：20101］
- 引数 uri に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 uri に指定された URI の形式が不正な場合にスローされます。［エラーコード：20147］

(23) newInstanceDocument

```
public XBRLInstanceDocument
newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument)
```

XBRL のバージョン：

2.0 および 2.1

説明：

空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを新規に生成します。
生成された XBRLInstanceDocument オブジェクトに対応する org.w3c.dom.Document オブジェクトが新規に作成されます。また，インスタンス文書のルート要素が XML 文書のルート要素として設定されます。

XBRL2.1 の場合：出発点となったタクソノミー本体およびリンクベースの追加

このメソッドを使用して新規にインスタンス文書を生成した場合は，引数 taxonomyDocument に指定した XBRLTaxonomyDocument オブジェクトを構築する際の出発点となったタクソノミー本体およびリンクベースのリストを XBRLTaxonomyDocument.getStartingPoints() メソッドを使用して取得し，出発点となったタクソノミー本体およびリンクベースを指す <link:schemaRef> 要素および <link:linkbaseRef> 要素をインスタンス文書に追加する必要があります。このとき，各要素の xlink:href 属性に指定する URI は，絶対 URI，相対 URI のどちらでもかまいません。ただし，インスタンス文書をストリームへ出力してから再度読み込んだ際に，xlink:href 属性の URI を適切な文書に解決できる必要があります。なお，必要な <link:schemaRef> 要素および <link:linkbaseRef> 要素を追加しなかった場合は，インスタンス文書をストリームへ出力してから再度読み込んだ際に，タクソノミー文書を構成するタクソノミー本体またはリンクベースに不足が生じるおそれがあるため，注意してください。

引数：

taxonomyDocument - タクソノミー文書を表す XBRLTaxonomyDocument オブジェクトを指定します。生成される XBRLInstanceDocument オブジェクトは，この XBRLTaxonomyDocument オブジェクトの複製を参照します。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 taxonomyDocument に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 taxonomyDocument に指定した XBRLTaxonomyDocument オブジェクトを所有する XBRLDocumentManager オブジェクトが，この XBRLInstanceDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトと異なる場合にスローされます。［エラーコード：20051］
- 引数 taxonomyDocument に指定された XBRLTaxonomyDocument オブジェクトが準拠する XBRL のバージョンと，この XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL のバージョンが異なる場合にスローされます。［エラーコード：20103］
- 引数 taxonomyDocument に指定した XBRLTaxonomyDocument オブジェクトが準拠する XBRL Dimensions のバージョンと，この XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョンが異なる場合にスローされます。［エラーコード：22103］

(24) newInstanceDocument

```
public XBRLInstanceDocument
newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument,
                                         java.lang.String systemId)
```

XBRL のバージョン：

2.1

説明：

空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを新規に生成します。

生成された XBRLInstanceDocument オブジェクトに対応する org.w3c.dom.Document オブジェクトが新規に作成されます。また，インスタンス文書のルート要素が XML 文書のルート要素として設定されます。

出発点となったタクソノミー本体およびリンクベースの追加

このメソッドを使用して新規にインスタンス文書を生成した場合は，引数

taxonomyDocument に指定した XBRLTaxonomyDocument オブジェクトを構築する際の出発点となったタクソノミー本体およびリンクベースのリストを XBRLTaxonomyDocument.getStartingPoints() メソッドを使用して取得し，出発点となったタクソノミー本体およびリンクベースを指す <link:schemaRef> 要素および <link:linkbaseRef> 要素をインスタンス文書に追加する必要があります。このとき，各要素の xlink:href 属性に指定する URI は，絶対 URI，相対 URI のどちらでもかまいません。ただし，インスタンス文書をストリームへ出力してから再度読み込んだ際に，xlink:href 属性の URI を適切な文書に解決できる必要があります。なお，必要な <link:schemaRef> 要素および <link:linkbaseRef> 要素を追加しなかった場合は，インスタンス文書をストリームへ出力してから再度読み込んだ際に，タクソノミー文書を構成するタクソノミー本体またはリンクベースに不足が生じるおそれがあるため，注意してください。

引数：

taxonomyDocument - タクソノミー文書を表す XBRLTaxonomyDocument オブジェクトを指定します。生成される XBRLInstanceDocument オブジェクトは，この XBRLTaxonomyDocument オブジェクトの複製を参照します。
systemId - インスタンス文書のシステム識別子を指定します。システム識別子は，相対 URI を解決するために使用されます。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- この XBRLInstanceDocumentBuilder オブジェクトが，バージョンが 2.1 である XBRL 仕様に準拠したインスタンス文書を作成するように設定されていない場合にスローされます。［エラーコード：20101］
- 引数 taxonomyDocument または systemId に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 taxonomyDocument に指定された XBRLTaxonomyDocument オブジェクトを所有する XBRLDocumentManager オブジェクトが，この XBRLInstanceDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトと異なる場合にスローされます。［エラーコード：20051］
- 引数 taxonomyDocument に指定された XBRLTaxonomyDocument オブジェクトが準拠する XBRL のバージョンと，この XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL のバージョンが異なる場合にスローされます。［エラーコード：20103］
- 引数 taxonomyDocument に指定した XBRLTaxonomyDocument オブジェクトが準拠する XBRL Dimensions のバージョンと，この

XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョンが異なる場合にスローされます。［エラーコード：22103］

- 引数 systemId に指定されたシステム識別子の形式が不正な場合にスローされます。［エラーコード：20147］

(25) newInstanceDocument

```
public XBRLInstanceDocument
newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument,
                                                org.w3c.dom.Document
w3cdomDocument)
```

XBRL のバージョン：

2.1

説明：

XML 文書の任意の位置にインスタンス文書を追加する場合に使用します。空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを新規に生成します。

ルート要素の挿入

生成された XBRLInstanceDocument オブジェクトのルート要素は，生成直後は DOM ツリーにつながっていません。そのため，ユーザーアプリケーションでは，W3C DOM の API を使用して，生成されたインスタンス文書のルート要素を，引数 w3cdomDocument に指定した XML 文書の任意の位置へ挿入する必要があります。ただし，同一 XML 文書中に記述されているほかのインスタンス文書の中に挿入しないでください。インスタンス文書のルート要素を表す org.w3c.dom.Element オブジェクトは，XBRLInstanceDocument.getInstanceRoot() メソッドを使用して取得できます。

出発点となったタクソノミー本体およびリンクベースの追加

このメソッドを使用して新規にインスタンス文書を生成した場合は，引数 taxonomyDocument に指定した XBRLTaxonomyDocument オブジェクトを構築する際の出発点となったタクソノミー本体およびリンクベースのリストを XBRLTaxonomyDocument.getStartingPoints() メソッドを使用して取得し，出発点となったタクソノミー本体およびリンクベースを指す <link:schemaRef> 要素および <link:linkbaseRef> 要素をインスタンス文書に追加する必要があります。このとき，各要素の xlink:href 属性に指定する URI は，絶対 URI，相対 URI のどちらでもかまいません。ただし，インスタンス文書をストリームへ出力してから再度読み込んだ際に，xlink:href 属性の URI を適切な文書に解決できる必要があります。なお，必要な <link:schemaRef> 要素および <link:linkbaseRef> 要素を追加しなかった場合は，インスタンス文書をストリームへ出力してから再度読み込んだ際に，タクソノミー文書を構成するタクソノミー本体またはリンクベースに不足が生じるおそれがあるため，注意してください。

引数：

taxonomyDocument - タクソノミー文書を表す XBRLTaxonomyDocument オブジェクトを指定します。生成される XBRLInstanceDocument オブジェクトは，この XBRLTaxonomyDocument オブジェクトの複製を参照します。
w3cdomDocument - インスタンス文書を挿入したい XML 文書を表す org.w3c.dom.Document オブジェクトを指定します。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- この XBRLInstanceDocumentBuilder オブジェクトが，バージョンが 2.1 である XBRL 仕様に準拠したインスタンス文書を作成するように設定されていない場合にスローされます。［エラーコード：20101］
- 引数 taxonomyDocument または w3cdomDocument に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 taxonomyDocument に指定された XBRLTaxonomyDocument オブジェクトを所有する XBRLDocumentManager オブジェクトが，この XBRLInstanceDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトと異なる場合にスローされます。［エラーコード：20051］
- 引数 taxonomyDocument に指定された XBRLTaxonomyDocument オブジェクトが準拠する XBRL のバージョンと，この XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL のバージョンが異なる場合にスローされます。［エラーコード：20103］
- 引数 taxonomyDocument に指定した XBRLTaxonomyDocument オブジェクトが準拠する XBRL Dimensions のバージョンと，この XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョンが異なる場合にスローされます。［エラーコード：22103］

(26) newInstanceDocument

```
public XBRLInstanceDocument
newInstanceDocument(XBRLTaxonomyDocument taxonomyDocument,
                                        org.w3c.dom.Document
w3cdomDocument,
                                    java.lang.String systemId)
```

XBRL のバージョン：

2.1

説明：

XML 文書の任意の位置にインスタンス文書を追加する場合に使用します。空のイン

スタンス文書を表す XBRLInstanceDocument オブジェクトを新規に生成します。

ルート要素の挿入

生成された XBRLInstanceDocument オブジェクトのルート要素は，生成直後は DOM ツリーにつながっていません。そのため，ユーザーアプリケーションでは，W3C DOM の API を使用して，生成されたインスタンス文書のルート要素を，引数 w3cdomDocument に指定した XML 文書の任意の位置へ挿入する必要があります。ただし，同一 XML 文書中に記述されているほかのインスタンス文書の中に挿入しないでください。インスタンス文書のルート要素を表す org.w3c.dom.Element オブジェクトは，XBRLInstanceDocument.getInstanceRoot() メソッドを使用して取得できます。

出発点となったタクソノミー本体およびリンクベースの追加

このメソッドを使用して新規にインスタンス文書を生成した場合は，引数 taxonomyDocument に指定した XBRLTaxonomyDocument オブジェクトを構築する際の出発点となったタクソノミー本体およびリンクベースのリストを XBRLTaxonomyDocument.getStartingPoints() メソッドを使用して取得し，出発点となったタクソノミー本体およびリンクベースを指す <link:schemaRef> 要素および <link:linkbaseRef> 要素をインスタンス文書に追加する必要があります。このとき，各要素の xlink:href 属性に指定する URI は，絶対 URI，相対 URI のどちらでもかまいません。ただし，インスタンス文書をストリームへ出力してから再度読み込んだ際に，xlink:href 属性の URI を適切な文書に解決できる必要があります。なお，必要な <link:schemaRef> 要素および <link:linkbaseRef> 要素を追加しなかった場合は，インスタンス文書をストリームへ出力してから再度読み込んだ際に，タクソノミー文書を構成するタクソノミー本体またはリンクベースに不足が生じるおそれがあるため，注意してください。

引数：

taxonomyDocument - タクソノミー文書を表す XBRLTaxonomyDocument オブジェクトを指定します。生成される XBRLInstanceDocument オブジェクトは，この XBRLTaxonomyDocument オブジェクトの複製を参照します。
w3cdomDocument - インスタンス文書を挿入したい XML 文書を表す org.w3c.dom.Document オブジェクトを指定します。
systemId - インスタンス文書のシステム識別子を指定します。システム識別子は，相対 URI を解決するために使用されます。

戻り値：

新しく生成された XBRLInstanceDocument オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- この XBRLInstanceDocumentBuilder オブジェクトが，バージョンが 2.1 である XBRL 仕様に準拠したインスタンス文書を作成するように設定されていない場合

にスローされます。［エラーコード：20101］

- 引数 taxonomyDocument，w3cdomDocument，または systemId に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 taxonomyDocument に指定された XBRLTaxonomyDocument オブジェクトを所有する XBRLDocumentManager オブジェクトが，この XBRLInstanceDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトと異なる場合にスローされます。［エラーコード：20051］
- 引数 taxonomyDocument に指定された XBRLTaxonomyDocument オブジェクトが準拠する XBRL のバージョンと，この XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL のバージョンが異なる場合にスローされます。［エラーコード：20103］
- 引数 taxonomyDocument に指定した XBRLTaxonomyDocument オブジェクトが準拠する XBRL Dimensions のバージョンと，この XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョンが異なる場合にスローされます。［エラーコード：22103］
- 引数 systemId に指定されたシステム識別子の形式が不正な場合にスローされます。［エラーコード：20147］

(27) getDimensionsVersion

```
public int getDimensionsVersion()
```

XBRL のバージョン：

2.1

説明：

XBRLInstanceDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているインスタンス文書を解析，または作成するのかを取得します。このメソッドが返す値は，XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析，または作成する際に使用する cbrp.parse.DimensionsVersion プロパティの値です。
XBRLInstanceDocumentBuilder オブジェクトが使用する cbrp.parse.DimensionsVersion プロパティの値は，XBRLInstanceDocumentBuilder オブジェクトを生成した XBRLInstanceDocumentBuilderFactory オブジェクトの setDimensionsVersion(int version) メソッド，または setProperty(java.lang.String name, java.lang.Object value) メソッドで設定した値です。
XBRLInstanceDocumentBuilderFactory オブジェクトで値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。
なお，cbrp.parse.DimensionsVersion プロパティの値は，インスタンス文書が

XBRL2.1 に準拠している場合だけ使用されます。インスタンス文書がほかの XBRL のバージョンに準拠している場合は，cbrp.parse.DimensionsVersion プロパティの値は使用されません。

戻り値：

XBRL Dimensions1.0 に準拠したインスタンス文書を解析または作成する場合は，XBRLConstants.DIMENSIONS_1_0 を返します。
XBRL Dimensions1.0 に準拠したインスタンス文書を解析または作成しない場合は，XBRLConstants.DIMENSIONS_NONE を返します。

(28) setCreatingSchemaLocations

```
public void setCreatingSchemaLocations(boolean creating)
```

XBRL のバージョン：

2.1

説明：

XBRLInstanceDocumentBuilder で XBRL2.1 に準拠した空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを生成するときに，ルート要素の xsi:schemaLocation 属性の初期値としてタクソノミー文書に含まれる各タクソノミー本体の対象名前空間 URI と位置を示す絶対 URI の組を設定するかどうかを指定します。デフォルト値は true です。
XBRL2.0 に準拠したインスタンス文書を作成する場合はこの設定値は無視され，ルート要素の xsi:schemaLocation 属性の初期値として，タクソノミー文書に含まれる各タクソノミー本体の対象名前空間 URI と位置を示す絶対 URI の組を設定します。

引数：

creating - XBRL2.1 に準拠した空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを生成するときに，ルート要素の xsi:schemaLocation 属性の初期値として，タクソノミー文書に含まれる各タクソノミー本体の対象名前空間 URI と位置を示す絶対 URI の組を設定する場合は true，設定しない場合は false を指定します。

(29) isCreatingSchemaLocations

```
public boolean isCreatingSchemaLocations()
```

XBRL のバージョン：

2.1

説明：

XBRLInstanceDocumentBuilder で XBRL2.1 に準拠した空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを生成するときに，ルート要素の xsi:schemaLocation 属性の初期値として，タクソノミー文書に含まれる各タクソノ

ミー本体の対象名前空間 URI と位置を示す絶対 URI の組を設定するかどうかを示します。

戻り値：

XBRL2.1 に準拠した空のインスタンス文書を表す XBRLInstanceDocument オブジェクトを生成するときに，ルート要素の xsi:schemaLocation 属性の初期値として，タクソノミー文書に含まれる各タクソノミー本体の対象名前空間 URI と位置を示す絶対 URI の組を設定する場合は true，設定しない場合は false を返します。

6.6.9 XBRLInstanceDocumentBuilderFactory インターフェース

```
public interface XBRLInstanceDocumentBuilderFactory
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLInstanceDocumentBuilder オブジェクトを生成するための API を定義します。

メソッドの概要：

XBRLInstanceDocumentBuilderFactory インターフェースのメソッドの概要を次の表に示します。

表 6-27　XBRLInstanceDocumentBuilderFactory インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLDocumentManager	getOwnerDocumentManager() XBRLInstanceDocumentBuilderFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。	2.0 および 2.1
2	void	setXBRLVersion(int version) どの XBRL バージョンのインスタンス文書を解析または作成するのかを設定します。	2.0 および 2.1
3	int	getXBRLVersion() どの XBRL バージョンのインスタンス文書を解析または作成するのかを取得します。	2.0 および 2.1
4	void	setValidating(boolean flag) インスタンス文書の解析時に文書の妥当性を検証するかどうかを設定します。	2.0 および 2.1
5	boolean	isValidating() インスタンス文書の解析時に文書の妥当性を検証するかどうかを調べます。	2.0 および 2.1

項番	説明		XBRL のバージョン
6	void	setDelayLoading(boolean flag) インスタンス文書の解析時にリンクベースの遅延読み込みをするかどうかを設定します。	2.0 および 2.1
7	boolean	isDelayLoading() インスタンス文書の解析時にリンクベースの遅延読み込みをするかどうかを調べます。	2.0 および 2.1
8	void	setExternalSchemaRefs(java.lang.String[] uris) インスタンス文書の解析時に参照するタクソノミー本体の URI のリストを設定します。	2.1
9	java.util.List<java.lang.String>	getExternalSchemaRefs() インスタンス文書の解析時に参照するタクソノミー本体の URI のリストを取得します。	2.1
10	void	setIgnoringComments(boolean flag) インスタンス文書の解析時にコメントを無視するかどうかを設定します。	2.0 および 2.1
11	boolean	isIgnoringComments() インスタンス文書の解析時にコメントを無視するかどうかを調べます。	2.0 および 2.1
12	void	setCoalescing(boolean flag) インスタンス文書の解析時に CDATA ノードを Text ノードに変換するかどうかを設定します。	2.0 および 2.1
13	boolean	isCoalescing() インスタンス文書の解析時に CDATA ノードを Text ノードに変換するかどうかを調べます。	2.0 および 2.1
14	XBRLInstanceDocumentBuilder	newInstanceDocumentBuilder() 現在設定されているパラメタを使用して，XBRLInstanceDocumentBuilder のインスタンスを新規に生成します。	2.0 および 2.1
15	void	setProperty(java.lang.String name, java.lang.Object value) XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書の解析時または作成時に使用するプロパティの値を設定します。	2.1
16	java.lang.Object	getProperty(java.lang.String name) XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書の解析時または作成時に使用するプロパティの値を取得します。	2.1
17	void	setDimensionsVersion(int version) XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているインスタンス文書を解析または作成するのかを指定します。	2.1

項番	説明		XBRL のバージョン
18	int	getDimensionsVersion() XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているインスタンス文書を解析または作成するのかを取得します。	2.1

(1) getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLInstanceDocumentBuilderFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

戻り値：

この XBRLInstanceDocumentBuilderFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを返します。

(2) setXBRLVersion

```
public void setXBRLVersion(int version)
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，どの XBRL バージョンのインスタンス文書を解析または作成するのかを設定します。
このメソッドで指定した値は，この XBRLInstanceDocumentBuilderFactory オブジェクトによって生成される XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析または作成する際に使用する cbrp.parse.XBRLVersion プロパティの値として設定されます。
cbrp.parse.XBRLVersion プロパティの値は，このメソッドか，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.XBRLVersion プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15) setProperty」の説明を参照してください。
cbrp.parse.XBRLVersion プロパティの値を設定する場合は，newInstanceDocumentBuilder() メソッドを呼び出す前に，このメソッドを呼び出してください。

引数：

version - XBRL2.0 に準拠したインスタンス文書を解析または生成する場合は XBRLConstants.XBRL_2_0，XBRL2.1 に準拠したインスタンス文書を解析または生成する場合は XBRLConstants.XBRL_2_1 を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 version に不正な値を指定した場合にスローされます。［エラーコード：20014］

（3） getXBRLVersion

```
public int getXBRLVersion()
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，どの XBRL バージョンのインスタンス文書を解析または作成するのかを取得します。

このメソッドが返す設定値は，XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析または作成する際に使用する cbrp.parse.XBRLVersion プロパティの値です。

この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが使用する cbrp.parse.XBRLVersion プロパティの値は，この XBRLInstanceDocumentBuilderFactory オブジェクトの setXBRLVersion() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.XBRLVersion プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。

戻り値：

XBRL2.0 に準拠したインスタンス文書を解析または作成する場合は XBRLConstants.XBRL_2_0，XBRL2.1 に準拠したインスタンス文書を解析または作成する場合は XBRLConstants.XBRL_2_1 を返します。

（4） setValidating

```
public void setValidating(boolean flag)
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLInstanceDocumentBuilderFactory によって生成される

XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時に文書の妥当性を検証するかどうかを設定します。このメソッドで指定した値は，このXBRLInstanceDocumentBuilderFactory オブジェクトによって生成されるXBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.validate プロパティの値として設定されます。
cbrp.parse.validate プロパティの値は，このメソッドか，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.validate プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。
cbrp.parse.validate プロパティの値を設定する場合は，newInstanceDocumentBuilder() メソッドを呼び出す前にこのメソッドを呼び出してください。

引数：

flag - インスタンス文書の解析時に妥当性を検証する場合は true，検証しない場合は false を指定します。

(5) isValidating

```
public boolean isValidating()
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLInstanceDocumentBuilderFactory によって生成されるXBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時に文書の妥当性を検証するかどうかを調べます。
このメソッドが返す設定値は，XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.validate プロパティの値です。
この XBRLInstanceDocumentBuilderFactory によって生成されるXBRLInstanceDocumentBuilder オブジェクトが使用する cbrp.parse.validate プロパティの値は，この XBRLInstanceDocumentBuilderFactory オブジェクトのsetValidating() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.validate プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。

戻り値：

インスタンス文書の解析時に妥当性を検証する場合は true，検証しない場合は false を返します。

(6) setDelayLoading

```
public void setDelayLoading(boolean flag)
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時にリンクベースの遅延読み込みをするかどうかを設定します。このメソッドで指定した値は，この XBRLInstanceDocumentBuilderFactory オブジェクトによって生成される XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.delayLoad プロパティの値として設定されます。

cbrp.parse.delayLoad プロパティの値は，このメソッドか，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.delayLoad プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。

XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。

cbrp.parse.delayLoad プロパティの値を設定する場合は，newInstanceDocumentBuilder() メソッドを呼び出す前にこのメソッドを呼び出してください。

インスタンス文書の解析時に妥当性を検証する場合は，リンクの解析を遅延できないため，このメソッドでリンクの解析を遅延する設定にしても，その設定は無視されます。リンクの解析を遅延させたい場合は，必ず妥当性の検証を行わないように設定してください。ただし，解析対象のインスタンス文書が XBRL2.1 に準拠している場合は，インスタンス文書の解析時に脚注リンクの遅延読み込みを行いません。

引数：

flag - インスタンス文書に記述されているリンクの解析を遅延する場合は true，遅延しない場合は false を指定します。

(7) isDelayLoading

```
public boolean isDelayLoading()
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時にリンクベースの遅延読み込みをするかどうかを調べます。このメソッドが返す設定値は，XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.delayLoad プロパティの値です。

この XBRLInstanceDocumentBuilderFactory によって生成される

XBRLInstanceDocumentBuilder オブジェクトが使用する cbrp.parse.delayLoad プロパティの値は，この XBRLInstanceDocumentBuilderFactory オブジェクトの setDelayLoading() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.delayLoad プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。

戻り値：

インスタンス文書に記述されているリンクの解析を遅延する場合は true，遅延しない場合は false を返します。

(8) setExternalSchemaRefs

```
public void setExternalSchemaRefs(java.lang.String[] uris)
```

XBRL のバージョン：

2.1

説明：

この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時に参照するタクソノミー本体の URI のリストを設定します。このメソッドで指定した値は，この XBRLInstanceDocumentBuilderFactory オブジェクトによって生成される XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.DTS.externalSchemaRefs プロパティの値として設定されます。

cbrp.parse.DTS.externalSchemaRefs プロパティの値は，このメソッドか，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.DTS.externalSchemaRefs プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。

引数 uris に null 以外を指定した場合は，それらの URI が設定された xlink:href 属性を持つ <link:schemaRef> 要素がインスタンス文書に記述されているかのように処理されます。つまり，引数 uris で指定した各 URI が指すタクソノミー本体が読み込まれてタクソノミー文書の一部となります。このとき，インスタンス文書に記述されている <link:schemaRef> 要素は，引数 uris に null 以外が指定されたかどうかに関係なく常に処理され，それらが指すタクソノミー本体はタクソノミー文書の一部となります。なお，引数 uris で指定した URI と，インスタンス文書に記述されている <link:schemaRef> 要素が同一のタクソノミー本体を指していた場合，そのタクソノミー本体は一度だけ読み込まれます。同一のタクソノミー本体が重複して読み込まれることはありません。

cbrp.parse.DTS.externalSchemaRefs プロパティの値を設定する場合は，newInstanceDocumentBuilder() メソッドを呼び出す前に，このメソッドを呼び出

してください。
cbrp.parse.DTS.externalSchemaRefs プロパティの値は，解析対象のインスタンス文書が XBRL2.1 に準拠している場合にだけ使用されます。解析対象のインスタンス文書がほかのバージョンの XBRL 仕様に準拠しているものである場合は，cbrp.parse.DTS.externalSchemaRefs プロパティの値は使用されません。

引数：
uris - インスタンス文書の解析時に参照するタクソノミー本体の URI を表す文字列の配列を指定します。インスタンス文書に記述されている <link:schemaRef> 要素だけを使用する場合には null を指定してください。

例外：
XBRLDOMRuntimeException -
- 引数 uris に指定した配列の要素数が 0 である場合にスローされます。［エラーコード：20013］
- 引数 uris に指定した配列に値が null である要素が存在する場合にスローされます。［エラーコード：20012］

（9） getExternalSchemaRefs

```
public java.util.List<java.lang.String> getExternalSchemaRefs()
```

XBRL のバージョン：
2.1

説明：
この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時に参照するタクソノミー本体の URI のリストを取得します。このメソッドが返す値は，XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.DTS.externalSchemaRefs プロパティの値です。
この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが使用する cbrp.parse.DTS.externalSchemaRefs プロパティの値は，この XBRLInstanceDocumentBuilderFactory オブジェクトの setExternalSchemaRefs() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.DTS.externalSchemaRefs プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。
cbrp.parse.DTS.externalSchemaRefs プロパティの値は，解析対象のインスタンス文書が XBRL2.1 に準拠している場合にだけ使用されます。解析対象のインスタンス文書がほかのバージョンの XBRL 仕様に準拠しているものである場合は，cbrp.parse.DTS.externalSchemaRefs プロパティの値は使用されません。
このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変

更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

インスタンス文書の解析時に参照するタクソノミー本体の URI を表す文字列のリストを返します。インスタンス文書に記述されている <link:schemaRef> 要素だけを使用して解析処理を行うように設定されている場合は，要素数が 0 のリストを返します。

(10) setIgnoringComments

```
public void setIgnoringComments(boolean flag)
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時にコメントを無視するかどうかを設定します。このメソッドで指定した値は，この XBRLInstanceDocumentBuilderFactory オブジェクトによって生成される XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.ignoringComments プロパティの値として設定されます。
cbrp.parse.ignoringComments プロパティの値は，このメソッドか，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.ignoringComments プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。
cbrp.parse.ignoringComments プロパティの値を設定する場合は，newInstanceDocumentBuilder() メソッドを呼び出す前に，このメソッドを呼び出してください。
cbrp.parse.ignoringComments プロパティでコメントを無視するように指定しても，無視しないように指定しても，XBRL DOM の動作には影響しません。ただし，XBRLInstanceNode.getW3CDOMNode() メソッドなどで取得した DOM ツリーを参照した場合に Comment ノードが含まれるかどうかについては，cbrp.parse.ignoringComments プロパティの指定によって違いが生じます。

引数：

flag - インスタンス文書の解析時にコメントを無視する場合は true，無視しない場合は false を指定します。

(11) isIgnoringComments

```
public boolean isIgnoringComments()
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時にコメントを無視するかどうかを調べます。このメソッドが返す値は，XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.ignoringComments プロパティの値です。
この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが使用する cbrp.parse.ignoringComments プロパティの値は，この XBRLInstanceDocumentBuilderFactory オブジェクトの setIgnoringComments() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.ignoringComments プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。

戻り値：

インスタンス文書の解析時にコメントを無視する場合は true，無視しない場合は false を返します。

(12) setCoalescing

```
public void setCoalescing(boolean flag)
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時に CDATA ノードを Text ノードに変換するかどうかを設定します。このメソッドで指定した値は，この XBRLInstanceDocumentBuilderFactory オブジェクトによって生成される XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.coalescing プロパティの値として設定されます。
cbrp.parse.coalescing プロパティの値は，このメソッドか，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.coalescing プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。
cbrp.parse.coalescing プロパティの値を設定する場合は，newInstanceDocumentBuilder() メソッドを呼び出す前にこのメソッドを呼び出してください。
cbrp.parse.coalescing プロパティで CDATA ノードを Text ノードに変換するように指定しても，変換しないように指定しても，XBRL DOM の動作には影響しません。

ただし，XBRLInstanceNode.getW3CDOMNode() メソッドなどで取得した DOM ツリーを参照した場合に CDATA ノードが含まれるかどうかについては，cbrp.parse.coalescing プロパティの指定によって違いが生じます。

引数：

flag - インスタンス文書の解析時に CDATA ノードを Text ノードに変換し，隣接する Text ノード（存在する場合）に追加する場合は true，Text ノードに変換しない場合は false を指定します。

(13) isCoalescing

```
public boolean isCoalescing()
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，インスタンス文書の解析時に CDATA ノードを Text ノードに変換するかどうかを調べます。このメソッドが返す値は，XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析する際に使用する cbrp.parse.coalescing プロパティの値です。

この XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが使用する cbrp.parse.coalescing プロパティの値は，この XBRLInstanceDocumentBuilderFactory オブジェクトの setCoalescing() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.coalescing プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。

戻り値：

インスタンス文書の解析時に CDATA ノードを Text ノードに変換する場合は true，変換しない場合は false を返します。

(14) newInstanceDocumentBuilder

```
public XBRLInstanceDocumentBuilder newInstanceDocumentBuilder()
```

XBRL のバージョン：

2.0 および 2.1

説明：

現在設定されているパラメタを使用して，XBRLInstanceDocumentBuilder のインスタンスを新規に生成します。

戻り値：

新しく生成された XBRLInstanceDocumentBuilder オブジェクトを返します。

(15) setProperty

```
public void setProperty(java.lang.String name,
                        java.lang.Object value)
```

XBRL のバージョン：

2.1

説明：

XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書の解析時または作成時に使用するプロパティの値を設定します。このメソッドで値を設定できるプロパティや，値として指定できるオブジェクトについては，「4.16.2　プロパティ一覧」を参照してください。

各プロパティの値は，このメソッドか，または setValidating() などの各プロパティ専用のメソッドで設定できます。これらのメソッドによってプロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時のプロパティの値は，引数 name で指定したプロパティに対し，初期化時に値が指定されていたかどうかで次のように異なります。

- 初期化時にプロパティの値が指定されていた場合
  そのときの指定値が XBRL DOM 初期化時の値となる。
- 初期化時にプロパティの値が指定されていなかった場合
  そのプロパティのデフォルト値が XBRL DOM 初期化時の値となる。

プロパティを設定する場合は，newInstanceDocumentBuilder() メソッドを呼び出す前にこのメソッドを呼び出してください。

このメソッドで設定したプロパティ値は，setValidating() などの各プロパティ専用のメソッドによって上書きできます。逆に，setValidating() などの各プロパティ専用のメソッドで設定した値もこのメソッドで上書きできます。例えば，このメソッドで cbrp.parse.validate プロパティの値として Boolean.TRUE を設定したあと，setValidating() メソッドを呼び出して false を設定した場合，getProperty() メソッドで取得した cbrp.parse.validate プロパティの値は Boolean.FALSE となります。また，isValidating() メソッドも false を返します。

引数：

name - プロパティの名前を指定します。

value - プロパティの値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 name に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 name に指定したプロパティの名前が正しくない場合にスローされます。［エラーコード：20150］
- 引数 name に指定したプロパティが値として null を受け付けないで，かつ引数 value に null を指定した場合にスローされます。［エラーコード：20151］

- 引数 value に指定したオブジェクトの型が正しくない場合にスローされます。［エラーコード：20152］
- 引数 value に指定したオブジェクトが保持している値が正しくない場合にスローされます。［エラーコード：20153］
- 引数 value に指定したオブジェクトが配列で，かつ要素数が 0 の場合にスローされます。［エラーコード：20154］
- 引数 value に指定したオブジェクトが配列で，かつ値が null である要素が存在する場合にスローされます。［エラーコード：20155］

（16）getProperty

```
public java.lang.Object getProperty(java.lang.String name)
```

XBRL のバージョン：

2.1

説明：

XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書の解析時または作成時に使用するプロパティの値を取得します。このメソッドで値を取得できるプロパティや，値として返されるオブジェクトについては，「4.16.2　プロパティ一覧」を参照してください。

引数：

name - プロパティの名前を指定します。

戻り値：

プロパティの値を返します。

例外：

XBRLDOMRuntimeException -

- 引数 name に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 name に指定したプロパティの名前が正しくない場合にスローされます。［エラーコード：20150］

（17）setDimensionsVersion

```
public void setDimensionsVersion(int version)
```

XBRL のバージョン：

2.1

説明：

XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているインスタンス文書を解析または作成するのかを指定します。このメソッドで指定した値は，XBRLInstanceDocumentBuilderFactory オブ

ジェクトによって生成される XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析または作成する際に使用する cbrp.parse.DimensionsVersion プロパティの値として設定されます。
cbrp.parse.DimensionsVersion プロパティの値は，このメソッドか，または setProperty(java.lang.String name, java.lang.Object value) メソッドで設定できます。これらのメソッドによって cbrp.parse.DimensionsVersion プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。
cbrp.parse.DimensionsVersion プロパティの値を設定する場合は，newInstanceDocumentBuilder() メソッドを呼び出す前にこのメソッドを呼び出してください。
cbrp.parse.DimensionsVersion プロパティの値は，インスタンス文書が XBRL2.1 に準拠している場合だけ使用されます。インスタンス文書がほかの XBRL のバージョンに準拠している場合は，cbrp.parse.DimensionsVersion プロパティの値は使用されません。

引数：

version - XBRL Dimensions1.0 に準拠したインスタンス文書を解析または生成する場合は，XBRLConstants.DIMENSIONS_1_0 を指定します。
XBRL Dimensions1.0 に準拠したインスタンス文書を解析または生成しない場合は，XBRLConstants.DIMENSIONS_NONE を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 version に不正な値を指定した場合にスローされます。［エラーコード：20014］

(18) getDimensionsVersion

```
public int getDimensionsVersion()
```

XBRL のバージョン：

2.1

説明：

XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているインスタンス文書を解析または作成するのかを取得します。このメソッドが返す値は，XBRLInstanceDocumentBuilder オブジェクトがインスタンス文書を解析または作成する際に使用する cbrp.parse.DimensionsVersion プロパティの値です。
XBRLInstanceDocumentBuilderFactory によって生成される XBRLInstanceDocumentBuilder オブジェクトが使用する cbrp.parse.DimensionsVersion プロパティの値は，

XBRLInstanceDocumentBuilderFactory オブジェクトの setDimensionsVersion(int version) メソッド，または setProperty(java.lang.String name, java.lang.Object value) メソッドで設定できます。これらのメソッドによって cbrp.parse.DimensionsVersion プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.6.9(15)　setProperty」の説明を参照してください。
cbrp.parse.DimensionsVersion プロパティの値は，インスタンス文書が XBRL2.1 に準拠している場合だけ使用されます。インスタンス文書がほかの XBRL のバージョンに準拠している場合は，cbrp.parse.DimensionsVersion プロパティの値は使用されません。

戻り値：

XBRL Dimensions1.0 に準拠したインスタンス文書を解析または作成する場合は，XBRLConstants.DIMENSIONS_1_0 を返します。
XBRL Dimensions1.0 に準拠したインスタンス文書を解析または作成しない場合は，XBRLConstants.DIMENSIONS_NONE を返します。

6.6.10　XBRLInstanceDocumentWriter インターフェース

```
public interface XBRLInstanceDocumentWriter
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書をストリームに書き出すためのインターフェースを提供します。

メソッドの概要：

XBRLInstanceDocumentWriter インターフェースのメソッドの概要を次の表に示します。

表 6-28　XBRLInstanceDocumentWriter インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLDocument Manager	getOwnerDocumentManager() XBRLInstanceDocumentWriter オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。	2.0 および 2.1
2	void	setEncoding(java.lang.String encoding) インスタンス文書を書き出す際に使用するエンコーディングを設定します。	2.0 および 2.1
3	java.lang.String	getEncoding() インスタンス文書を書き出す際に使用するエンコーディングを取得します。	2.0 および 2.1

項番	説明		XBRL のバージョン
4	void	setIndent(boolean indent) インスタンス文書を書き出す際にインデントをするかどうかを設定します。	2.0 および 2.1
5	boolean	isIndent() インスタンス文書を書き出す際にインデントをするかどうかを調べます。	2.0 および 2.1
6	void	write(java.io.OutputStream os, XBRLInstanceDocument doc) インスタンス文書をストリームに書き出します。	2.0 および 2.1

(1) getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLInstanceDocumentWriter オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

戻り値：

この XBRLInstanceDocumentWriter オブジェクトを所有する XBRLDocumentManager オブジェクトを返します。

(2) setEncoding

```
public void setEncoding(java.lang.String encoding)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書を書き出す際に使用するエンコーディングを設定します。このメソッドが呼び出されなかった場合は，デフォルトのエンコーディングである UTF-8 が使用されます。

XBRLInstanceDocumentWriter は，JAXP に準拠した XSLT プロセッサを使用してインスタンス文書をストリームへ出力します。このため，setEncoding() メソッドで指定できるエンコーディングは，使用する XSLT プロセッサに依存します。

uCosminexus Business Reporting Processor としては，利用できるエンコーディングに制限を設けません。setEncoding() メソッドでは，引数に null を指定しないかぎり，例外は発生しません。XSLT プロセッサでサポートされていないエンコーディングを指定した場合は，write() メソッドで出力処理を実行する際に，例外が発生するおそれがあります。

引数：

encoding - エンコーディングを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 encoding に null を指定した場合にスローされます。［エラーコード：20002］

(3) getEncoding

```
public java.lang.String getEncoding()
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書を書き出す際に使用するエンコーディングを取得します。

戻り値：

インスタンス文書を書き出す際に使用されるエンコーディングを返します。

(4) setIndent

```
public void setIndent(boolean indent)
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書を書き出す際にインデントをするかどうかを設定します。このメソッドが呼び出されなかった場合は，インデントは設定しません。
このメソッドでインデントを設定するように指定すると，
XBRLInstanceDocumentWriter オブジェクトは，XSLT プロセッサに対して
indent プロパティを yes に設定してから出力処理を実行します。

引数：

indent - インスタンス文書を書き出す際にインデントを設定する場合は true，設定しない場合は false を指定します。

(5) isIndent

```
public boolean isIndent()
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書を書き出す際にインデントをするかどうかを調べます。

戻り値：

インスタンス文書を書き出す際にインデントを設定する場合は true，設定しない場合は false を返します。

(6) write

```
public void write(java.io.OutputStream os,
                  XBRLInstanceDocument doc)
            throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書をストリームに書き出します。

注　書き出したインスタンス文書は，一度検証することを推奨します。

- XBRL2.1 の場合：
  インスタンス文書のルート要素が XML 文書のルート要素ではない場合，インスタンス文書が埋め込まれている XML 文書全体をストリームへ書き出します。インスタンス文書のルート要素が DOM ツリーにつながっていない場合は，XBRLDOMRuntimeException がスローされます。

引数：

os - インスタンス文書を書き出すストリームを指定します。

doc - ストリームへ書き出すインスタンス文書を指定します。

例外：

XBRLDOMException -

- 入出力エラーが発生した場合にスローされます。[エラーコード：20052]

XBRLDOMRuntimeException -

- 引数 os または doc に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 doc に指定した XBRLInstanceDocument オブジェクトを所有する XBRLDocumentManager オブジェクトが，この XBRLInstanceDocumentWriter オブジェクトを所有する XBRLDocumentManager オブジェクトと異なる場合にスローされます。[エラーコード：20051]
- XBRL2.1 の場合：引数 doc に指定されたインスタンス文書のルート要素が DOM ツリーにつながっていない場合にスローされます。[エラーコード：20106]

6.6.11 XBRLInstanceDocumentWriterFactory インターフェース

```
public interface XBRLInstanceDocumentWriterFactory
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLInstanceDocumentWriter オブジェクトを生成するための API を定義します。

メソッドの概要：

XBRLInstanceDocumentWriterFactory インターフェースのメソッドの概要を次の表に示します。

表 6-29　XBRLInstanceDocumentWriterFactory インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLDocument Manager	getOwnerDocumentManager() XBRLInstanceDocumentWriterFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。	2.0 および 2.1
2	XBRLInstanceDocumentWriter	newInstanceDocumentWriter() 現在設定されているパラメタを使用して，XBRLInstanceDocumentWriter のインスタンスを新規に生成します。	2.0 および 2.1

(1) getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLInstanceDocumentWriterFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

戻り値：

この XBRLInstanceDocumentWriterFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを返します。

(2) newInstanceDocumentWriter

```
public XBRLInstanceDocumentWriter newInstanceDocumentWriter()
```

XBRL のバージョン：

2.0 および 2.1

説明：

現在設定されているパラメタを使用して，XBRLInstanceDocumentWriter のインスタンスを新規に生成します。

戻り値：

新しく生成された XBRLInstanceDocumentWriter オブジェクトを返します。

6.6.12　XBRLInstanceElement インターフェース

```
public interface XBRLInstanceElement
```

extends XBRLInstanceNode

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に記述されている要素を表します。要素は，タクソノミー文書で定義されるアイテムまたはタプルです。

要素がアイテムまたはタプルのどちらかは，getElementType() で調べることができます。getElementType() が ITEM_ELEMENT を返した場合は XBRLInstanceItem へ，TUPLE_ELEMENT を返した場合は XBRLInstanceTuple へキャストできます。

XBRLInstanceElement オブジェクトは，XBRLInstanceDocument オブジェクトまたは XBRLInstanceTuple オブジェクトの子となることができます。

非アクティブな XBRLInstanceElement オブジェクトを，XBRLInstanceDocument オブジェクトまたはアクティブな XBRLInstanceTuple オブジェクトの子として設定すると，子として設定された XBRLInstanceElement オブジェクトは非アクティブな状態からアクティブな状態へと変化します。

また，XBRLInstanceElement オブジェクトが非アクティブな XBRLInstanceTuple オブジェクトの子である場合はその XBRLInstanceElement オブジェクトも非アクティブです。しかし，親である非アクティブな XBRLInstanceTuple オブジェクトまたはその祖先に当たる XBRLInstanceTuple オブジェクトが，XBRLInstanceDocument オブジェクトまたはアクティブな XBRLInstanceTuple オブジェクトの子として設定され，アクティブになった場合，その子または子孫である XBRLInstanceElement オブジェクトも同時にアクティブな状態へと変化します。

このようにして XBRLInstanceElement オブジェクトが非アクティブな状態からアクティブな状態へ変化する際に，XBRLInstanceElement オブジェクトと関連づけられている XBRLFootnote オブジェクトが非アクティブである場合，その XBRLFootnote オブジェクトは XBRLInstanceDocument の addFootnote() によってインスタンス文書へ自動的に追加されます。

フィールドの概要：

XBRLInstanceElement インターフェースのフィールドの概要を次の表に示します。

表 6-30　XBRLInstanceElement インターフェースのフィールドの概要

項番	説明	
1	static int	ITEM_ELEMENT このオブジェクトは，アイテムを表します。
2	static int	TUPLE_ELEMENT このオブジェクトは，タプルを表します。

メソッドの概要：

XBRLInstanceElement インターフェースのメソッドの概要を次の表に示します。

表 6-31　XBRLInstanceElement インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLTaxonomyElement	getTaxonomyElement() 要素の要素宣言を取得します。	2.0 および 2.1
2	int	getElementType() オブジェクトの型を表すコードを取得します。	2.0 および 2.1
3	java.lang.String	getNamespaceURI() 要素が属する名前空間 URI を取得します。	2.0 および 2.1
4	java.lang.String	getPrefix() 要素の名前空間接頭辞を取得します。	2.0 および 2.1
5	java.lang.String	getName() 要素の名前を取得します。	2.0 および 2.1
6	java.lang.String	getId() 要素の id 属性の値を取得します。	2.0 および 2.1
7	void	setId(java.lang.String id) 要素の id 属性の値を設定します。	2.0 および 2.1
8	java.util.List<XBRLLabel>	getLabels() 要素に関連づけられているラベルのリストを取得します。	2.0 および 2.1
9	java.util.List<XBRLLabel>	getLabelsByLang(java.lang.String lang) 要素に関連づけられているラベルのうち，指定した言語のラベルのリストを取得します。	2.0 および 2.1
10	java.util.List<XBRLLabel>	getLabelsByRole(java.lang.String role) 要素に関連づけられているラベルのうち，指定したロールを持つラベルのリストを取得します。	2.0 および 2.1
11	XBRLLabel	getLabel(java.lang.String lang, java.lang.String role) 要素に関連づけられているラベルのうち，指定した言語とロールを持つラベルを取得します。	2.0 および 2.1
12	java.util.List<XBRLLabel>	getLabels(java.lang.String lang, java.lang.String role) 要素に関連づけられているラベルのうち，指定した言語とロールを持つラベルのリストを取得します。	2.1
13	java.util.List<XBRLLabelRelationship>	getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) 要素の要素宣言とラベルの関係を表す XBRLLabelRelationship オブジェクトのリストを取得します。	2.1
14	java.util.List<XBRLReference>	getReferences() 要素に関連づけられている文献への参照のリストを取得します。	2.0 および 2.1

項番	説明		XBRL のバージョン
15	java.util.List<XBRLReference>	getReferences(java.lang.String role) 要素に関連づけられている文献への参照のうち，指定したロールを持つ文献への参照のリストを取得します。	2.0 および 2.1
16	java.util.List<XBRLReferenceRelationship>	getReferenceRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) 要素の要素宣言と文献への参照の関係を表す XBRLReferenceRelationship オブジェクトのリストを取得します。	2.1
17	boolean	isNillable() 要素の要素宣言の nillable 属性の値を取得します。	2.1
18	boolean	isNil() 要素の xsi:nil 属性の値を取得します。	2.1
19	void	setNil(java.lang.Boolean nil) 要素の xsi:nil 属性の値を設定します。	2.1
20	java.util.List<XBRLFootnote>	getFootnotes() 要素に関連づけられている脚注のリストを取得します。	2.0 および 2.1
21	java.util.List<XBRLFootnote>	getFootnotesByLang(java.lang.String lang) 要素に関連づけられている脚注のうち，指定した言語の脚注のリストを取得します。	2.0 および 2.1
22	java.util.List<XBRLFootnote>	getFootnotesByRole(java.lang.String role) 要素に関連づけられている脚注のうち，指定したロールの脚注のリストを取得します。	2.0 および 2.1
23	java.util.List<XBRLFootnote>	getFootnotes(java.lang.String lang, java.lang.String role) 要素に関連づけられている脚注のうち，指定した言語およびロールの脚注のリストを取得します。	2.0 および 2.1
24	XBRLFootnote	addFootnote(XBRLFootnote newFootnote) 要素に脚注を関連づけます。	2.0 および 2.1
25	XBRLFootnote	addFootnote(XBRLFootnote newFootnote, java.lang.String arcrole, java.math.BigDecimal order) 要素に脚注を関連づけます。カスタムアークロールを使用できます。	2.1
26	XBRLFootnote	removeFootnote(XBRLFootnote oldFootnote) 要素と脚注の関連づけを削除します。	2.0 および 2.1
27	XBRLFootnote	removeFootnote(XBRLFootnote oldFootnote, java.lang.String arcrole, java.math.BigDecimal order) 要素と脚注の関連づけを削除します。	2.1

項番	説明		XBRL のバージョン
28	java.util.List<XBRLFootnoteRelationship>	getFootnoteRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) 要素と脚注の関係を表す XBRLFootnoteRelationship オブジェクトのリストを取得します。	2.1
29	boolean	isNilAttributeSpecified() 要素に xsi:nil 属性が定義されていたかどうかを調べます。	2.1
30	java.util.List<XBRLDefinitionRelationship>	getDefinitionRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) 要素の要素宣言とほかの要素宣言の関係を表す XBRLDefinitionRelationship オブジェクトのリストを取得します。	2.1

(1) getTaxonomyElement

```
public XBRLTaxonomyElement getTaxonomyElement()
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素の要素宣言を取得します。

戻り値：

この要素の要素宣言を表す XBRLTaxonomyElement オブジェクトを返します。

(2) getElementType

```
public int getElementType()
```

XBRL のバージョン：

2.0 および 2.1

説明：

オブジェクトの型を表すコードを取得します。

戻り値：

オブジェクトの型を表すコードを返します。戻り値のコードは,
XBRLInstanceElement インターフェースのフィールド値のどれかです。

(3) getNamespaceURI

```
public java.lang.String getNamespaceURI()
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素が属する名前空間 URI を取得します。

戻り値：

この要素が属する名前空間 URI を返します。

(4) getPrefix

```
public java.lang.String getPrefix()
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素の名前空間接頭辞を取得します。

戻り値：

この要素の名前空間接頭辞を返します。この要素がデフォルトの名前空間に属する場合は null を返します。

インスタンス文書を編集して名前空間が正規化された場合，正規化の前後で戻り値の接頭辞が異なります。正規化前は，既存の文書で指定された接頭辞を返します。正規化後は，正規化によって変更された接頭辞を返します。

(5) getName

```
public java.lang.String getName()
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素の名前を取得します。

戻り値：

この要素の名前を返します。

(6) getId

```
public java.lang.String getId()
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素の id 属性の値を取得します。

戻り値：

この要素の id 属性の値を表す文字列を返します。この要素が id 属性を持たない場合は null を返します。

(7) setId

```
public void setId(java.lang.String id)
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素の id 属性の値を設定します。

id 属性を削除したい場合は，引数 id に null を指定してください。ただし，この要素がアクティブで，かつ，脚注と関連づけられている場合は，id 属性を削除できません。

引数：

id - id 属性の値を指定します。id 属性を削除する場合は null を指定してください。

例外：

XBRLDOMRuntimeException -

- 引数 id に指定した値が xsd:NCName 型でない場合にスローされます。［エラーコード：20108］
- この要素がインスタンス文書中に存在していて，かつ，引数 id で指定されたのと同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。［エラーコード：20028］
- この要素がアクティブで，かつ，脚注と関連づけられていて，かつ，引数 id に null を指定した場合にスローされます。［エラーコード：20146］

（8） getLabels

```
public java.util.List<XBRLLabel> getLabels()
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられているラベルのリストを取得します。リストの各要素は，XBRLLabel へキャストできます。

XBRLInstanceElement および XBRLTaxonomyElement インターフェースの getLabels() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。

しかし，XBRLTaxonomyDocument インターフェースの getLabels() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。

このメソッドが返す XBRLLabel オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabel オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

- XBRL2.1 の場合：

このメソッドでは，要素とラベルの関係を定義している <link:labelArc> 要素のアークロール，および，その <link:labelArc> 要素を含んでいるラベルリンクのロールについては考慮しません。特定の値のロールを持つラベルリンクに記述され，かつ，特定の値のアークロールを持つような <link:labelArc> 要素によって関連づけられたラベルを取得したい場合は，XBRLInstanceElement.getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。

戻り値：

この要素に関連づけられているラベルを表す XBRLLabel オブジェクトのリストを返します。この要素に関連づけられているラベルがない場合は，要素数が 0 のリストを返します。

（9） getLabelsByLang

```
public java.util.List<XBRLLabel> getLabelsByLang(java.lang.String
lang)
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられているラベルのうち，指定した言語のラベルのリストを取得します。リストの各要素は，XBRLLabel へキャストできます。

XBRLInstanceElement および XBRLTaxonomyElement インターフェースの getLabelsByLang() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。

しかし，XBRLTaxonomyDocument インターフェースの getLabelsByLang() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。

このメソッドが返す XBRLLabel オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabel オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

- XBRL2.1 の場合：

  このメソッドでは，要素とラベルの関係を定義している <link:labelArc> 要素のアークロール，および，その <link:labelArc> 要素を含んでいるラベルリンクのロールについては考慮しません。特定の値のロールを持つラベルリンクに記述され，かつ，特定の値のアークロールを持つような <link:labelArc> 要素によって関連づけられたラベルを取得したい場合は，XBRLInstanceElement.getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。

引数：

lang - ラベルの言語を指定します。

戻り値：

この要素のラベルを表す XBRLLabel オブジェクトのリストを返します。指定した言語のラベルがない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

（10）getLabelsByRole

```
public java.util.List<XBRLLabel> getLabelsByRole(java.lang.String
role)
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられているラベルのうち，指定したロールを持つラベルのリストを取得します。リストの各要素は，XBRLLabel へキャストできます。

XBRLInstanceElement および XBRLTaxonomyElement インターフェースの getLabelsByRole() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。

しかし，XBRLTaxonomyDocument インターフェースの getLabelsByRole() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。

このメソッドが返す XBRLLabel オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabel オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

- XBRL2.1 の場合：

  このメソッドでは，要素とラベルの関係を定義している <link:labelArc> 要素のアークロール，および，その <link:labelArc> 要素を含んでいるラベルリンクのロールについては考慮しません。特定の値のロールを持つラベルリンクに記述され，かつ，特定の値のアークロールを持つような <link:labelArc> 要素によって関連づけられたラベルを取得したい場合は，XBRLInstanceElement.getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。

引数：

role - ラベルのロールを指定します。ロールを持たないラベルを取得する場合は null

を指定します。

戻り値：

この要素のラベルを表す XBRLLabel オブジェクトのリストを返します。指定したロールを持つラベルがない場合は，要素数が 0 のリストを返します。

(11) getLabel

```
public XBRLLabel getLabel(java.lang.String lang,
                          java.lang.String role)
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられているラベルのうち，指定した言語とロールを持つラベルを取得します。

一つの要素に対して同じ言語とロールを持つラベルが複数関連づけられていた場合の動作は，XBRL のバージョンによって異なります。詳細については，XBRLTaxonomyElement インターフェースの「6.8.16(13)　getLabel」を参照してください。

- XBRL2.1 の場合：

  このメソッドでは，要素とラベルの関係を定義している <link:labelArc> 要素のアークロール，および，その <link:labelArc> 要素を含んでいるラベルリンクのロールについては考慮しません。特定の値のロールを持つラベルリンクに記述され，かつ，特定の値のアークロールを持つような <link:labelArc> 要素によって関連づけられたラベルを取得したい場合は，XBRLInstanceElement.getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。

引数：

lang - ラベルの言語を指定します。

role - ラベルのロールを指定します。ロールを持たないラベルを取得する場合は null を指定します。

戻り値：

この要素のラベルを表す XBRLLabel オブジェクトを返します。指定した言語とロールを持つラベルがない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(12) getLabels

```
public java.util.List<XBRLLabel> getLabels(java.lang.String lang,
                                           java.lang.String role)
```

XBRL のバージョン：

2.1

説明：

要素に関連づけられているラベルのうち，指定した言語とロールを持つラベルのリストを取得します。

このメソッドでは，要素とラベルの関係を定義している <link:labelArc> 要素のアークロール，および，その <link:labelArc> 要素を含んでいるラベルリンクのロールについては考慮しません。特定の値のロールを持つラベルリンクに記述され，かつ，特定の値のアークロールを持つような <link:labelArc> 要素によって関連づけられたラベルを取得したい場合は，XBRLInstanceElement.getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。

このメソッドが返す XBRLLabel オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabel オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

lang - ラベルの言語を指定します。

role - ラベルのロールを指定します。ロールを持たないラベルを取得する場合は null を指定します。

戻り値：

この要素のラベルを表す XBRLLabel オブジェクトのリストを返します。指定された言語とロールを持つラベルが存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(13) getLabelRelationships

```
public java.util.List<XBRLLabelRelationship>
getLabelRelationships(java.lang.String extendedLinkRole,
                                        java.lang.String arcrole)
```

XBRL のバージョン：

2.1

説明：

要素の要素宣言とラベルの関係を表す XBRLLabelRelationship オブジェクトのリス

トを取得します。
この要素の要素宣言に関連づけられたラベルは，このメソッドが返すリストに含まれる各 XBRLLabelRelationship オブジェクトに対して getTo() メソッドを呼び出すことによって取得できます。また，このメソッドが返すリストに含まれる各 XBRLLabelRelationship オブジェクトに対して getFrom() メソッドを呼び出した場合の戻り値は，この要素の要素宣言を表す XBRLTaxonomyElement オブジェクトとなります。
このメソッドが返すリストに含まれる XBRLLabelRelationship オブジェクトの並びは，各 XBRLLabelRelationship オブジェクトが表す関係を定義しているアークの order 属性の値によって昇順にソートされています。アークに order 属性が指定されていない場合は，1 が指定されたものと仮定します。詳細については，XBRLRelationship インターフェースの「6.7.7(4)　getOrder」を参照してください。アークの order 属性の値が同じである XBRLLabelRelationship オブジェクトが複数ある場合，それらの XBRLLabelRelationship オブジェクトの並び順は不定です。
このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabelRelationship オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

extendedLinkRole - 関係を定義しているアークを含むラベルリンクのロールを指定します。
arcrole - 関係を定義しているアークのアークロールを指定します。

戻り値：

引数 extendedLinkRole で指定されたロールを持つラベルリンクに属し，かつ，引数 arcrole で指定されたアークロールを持つアークによって定義された，この要素の要素宣言とラベルの関係を表す XBRLLabelRelationship オブジェクトのリストを返します。指定された関係が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 extendedLinkRole または arcrole に null を指定した場合にスローされます。［エラーコード：20002］

(14) getReferences

```
public java.util.List<XBRLReference> getReferences()
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられている文献への参照のリストを取得します。
XBRLInstanceElement および XBRLTaxonomyElement インターフェースの getReferences() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。
しかし，XBRLTaxonomyDocument インターフェースの getReferences() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。
このメソッドが返す XBRLReference オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLReference オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

- XBRL2.1 の場合：
  このメソッドでは，要素と文献への参照の関係を定義している <link:referenceArc> 要素のアークロール，および，その <link:referenceArc> 要素を含んでいるリファレンスリンクのロールについては考慮しません。特定の値のロールを持つリファレンスリンクに記述され，かつ，特定の値のアークロールを持つような <link:referenceArc> 要素によって関連づけられた文献への参照を取得したい場合は，
  XBRLInstanceElement.getReferenceRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。

戻り値：

文献への参照を表す XBRLReference オブジェクトのリストを返します。

(15) getReferences

```
public java.util.List<XBRLReference>
getReferences(java.lang.String role)
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられている文献への参照のうち，指定したロールを持つ文献への参照のリストを取得します。リストの各要素は，XBRLReference にキャストできます。
XBRLInstanceElement および XBRLTaxonomyElement インターフェースの getReferences() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。
しかし，XBRLTaxonomyDocument インターフェースの getReferences() では，文

書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。
このメソッドが返す XBRLReference オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLReference オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

- XBRL2.1 の場合：
  このメソッドでは，要素と文献への参照の関係を定義している <link:referenceArc> 要素のアークロール，および，その <link:referenceArc> 要素を含んでいるリファレンスリンクのロールについては考慮しません。特定の値のロールを持つリファレンスリンクに記述され，かつ，特定の値のアークロールを持つような <link:referenceArc> 要素によって関連づけられた文献への参照を取得したい場合は，
  XBRLInstanceElement.getReferenceRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。

引数：

role - 文献への参照のロールを指定します。ロールを持たない文献への参照を取得する場合は null を指定します。

戻り値：

この要素の文献への参照を表す XBRLReference オブジェクトのリストを返します。指定したロールを持つ文献への参照がない場合は，要素数が 0 のリストを返します。

(16) getReferenceRelationships

```
public java.util.List<XBRLReferenceRelationship>
getReferenceRelationships(java.lang.String extendedLinkRole,
                                          java.lang.String arcrole)
```

XBRL のバージョン：

2.1

説明：

要素の要素宣言と文献への参照の関係を表す XBRLReferenceRelationship オブジェクトのリストを取得します。
この要素の要素宣言に関連づけられた文献への参照は，このメソッドが返すリストに含まれる各 XBRLReferenceRelationship オブジェクトに対して getTo() メソッドを呼び出すことによって取得できます。また，このメソッドが返すリストに含まれる各 XBRLReferenceRelationship オブジェクトに対して getFrom() メソッドを呼び出した場合の戻り値は，この要素の要素宣言を表す XBRLTaxonomyElement オブジェクトとなります。
このメソッドが返すリストに含まれる XBRLReferenceRelationship オブジェクトの

並びは，各 XBRLReferenceRelationship オブジェクトが表す関係を定義しているアークの order 属性の値によって昇順にソートされています。アークに order 属性が指定されていない場合は，1 が指定されたものと仮定します。詳細については，XBRLRelationship インターフェースの「6.7.7(4)　getOrder」を参照してください。アークの order 属性の値が同じである XBRLReferenceRelationship オブジェクトが複数ある場合，それらの XBRLReferenceRelationship オブジェクトの並び順は不定です。
このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLReferenceRelationship オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

extendedLinkRole - 関係を定義しているアークを含むリファレンスリンクのロールを指定します。
arcrole - 関係を定義しているアークのアークロールを指定します。

戻り値：

引数 extendedLinkRole で指定されたロールを持つリファレンスリンクに属し，かつ，引数 arcrole で指定されたアークロールを持つアークによって定義された，この要素の要素宣言と文献への参照の関係を表す XBRLReferenceRelationship オブジェクトのリストを返します。指定された関係が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 extendedLinkRole または arcrole に null を指定した場合にスローされます。［エラーコード：20002］

(17) isNillable

```
public boolean isNillable()
```

XBRL のバージョン：

2.1

説明：

要素の要素宣言の nillable 属性の値を取得します。

戻り値：

この要素の要素宣言に指定されている nillable 属性の値を返します。nillable 属性が指定されていない場合は false を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

(18) isNil

```
public boolean isNil()
```

XBRL のバージョン：

2.1

説明：

要素の xsi:nil 属性の値を取得します。

戻り値：

この要素の xsi:nil 属性の値を返します。xsi:nil 属性が指定されていない場合は false を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

(19) setNil

```
public void setNil(java.lang.Boolean nil)
```

XBRL のバージョン：

2.1

説明：

要素の xsi:nil 属性の値を設定します。

要素に xsi:nil 属性を設定できるのは，この要素の要素宣言で nillable 属性に "true" が指定されている場合だけです。要素宣言の nillable 属性に "true" が指定されていない場合は，このメソッドは XBRLDOMRuntimeException をスローします。要素宣言の nillable 属性に "true" が指定されているかどうかは，XBRLInstanceElement.isNillable() メソッドを使用して調べることができます。

引数 nil に java.lang.Boolean.TRUE を指定した場合は，この要素の内容が自動的に削除されます。アイテムの場合は値が設定されていない（XBRLInstanceItem.hasValue() メソッドが false を返す）状態となり，タプルの場合は子要素がすべて削除された状態となります。このとき，この要素がアイテムで，かつ，precision 属性または decimals 属性のどちらかが指定されている場合は，それらの属性も削除されます。

この要素が数値アイテムで，かつ，ベースデータ型が FRACTION_ITEM 以外で，かつ，アイテムの xsi:nil 属性に "true" が指定されている場合に，このメソッドの引数 nil に java.lang.Boolean.FALSE または null を指定すると，自動的にアイテムの

precision 属性に "INF" が設定されます。precision 属性や decimals 属性の値は，XBRLInstanceItem.setPrecision() メソッドや XBRLInstanceItem.setDecimals() メソッドを使用して必要に応じて設定してください。

引数：

nil - xsi:nil 属性に "true" を指定する場合は java.lang.Boolean.TRUE，"false" を指定する場合は java.lang.Boolean.FALSE を指定します。xsi:nil 属性を削除する場合は null を指定してください。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- XBRLInstanceElement.isNillable() メソッドが false を返す場合（この要素の要素宣言で nillable 属性に "false" が指定されているか，または，nillable 属性が指定されていない場合）で，かつ，引数 nil に null 以外の値を指定した場合にスローされます。［エラーコード：20129］

(20) getFootnotes

```
public java.util.List<XBRLFootnote> getFootnotes()
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられている脚注のリストを取得します。

XBRLInstanceElement インターフェースの getFootnotes() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得する脚注のリストには含まれません。

しかし，XBRLInstanceDocument インターフェースの getFootnotes() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得する脚注のリストに含まれます。

このメソッドが返す XBRLFootnote オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLFootnote オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

- XBRL2.1 の場合：

  このメソッドでは，要素と脚注の関係を定義している <link:footnoteArc> 要素のアークロール，および，その <link:footnoteArc> 要素を含んでいる脚注リンクのロールについては考慮しません。特定の値のロールを持つ脚注リンクに記述され，かつ，特定の値のアークロールを持つような <link:footnoteArc> 要素によって関

連づけられた脚注を取得したい場合は，
XBRLInstanceElement.getFootnoteRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。

戻り値：

この要素に関連づけられた脚注を表す XBRLFootnote オブジェクトのリストを返します。この要素に関連づけられた脚注がない場合は，要素数が 0 のリストを返します。

(21) getFootnotesByLang

```
public java.util.List<XBRLFootnote>
getFootnotesByLang(java.lang.String lang)
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられている脚注のうち，指定した言語の脚注のリストを取得します。
XBRLInstanceElement インターフェースの getFootnotesByLang() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得する脚注のリストには含まれません。
しかし，XBRLInstanceDocument インターフェースの getFootnotesByLang() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得する脚注のリストに含まれます。
このメソッドが返す XBRLFootnote オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLFootnote オブジェクトに増減があった場合，その変更は直ちに反映されます。
また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

- XBRL2.1 の場合：
  このメソッドでは，要素と脚注の関係を定義している <link:footnoteArc> 要素のアークロール，および，その <link:footnoteArc> 要素を含んでいる脚注リンクのロールについては考慮しません。特定の値のロールを持つ脚注リンクに記述され，かつ，特定の値のアークロールを持つような <link:footnoteArc> 要素によって関連づけられた脚注を取得したい場合は，
  XBRLInstanceElement.getFootnoteRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。

引数：

lang - 脚注の言語を指定します。

戻り値：

この要素に関連づけられた脚注を表す XBRLFootnote オブジェクトのリストを返し

ます。引数で指定した言語の脚注が関連づけられていない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(22) getFootnotesByRole

```
public java.util.List<XBRLFootnote>
getFootnotesByRole(java.lang.String role)
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられている脚注のうち，指定したロールの脚注のリストを取得します。

XBRLInstanceElement インターフェースの getFootnotesByRole() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得する脚注のリストには含まれません。

しかし，XBRLInstanceDocument インターフェースの getFootnotesByRole() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得する脚注のリストに含まれます。

このメソッドが返す XBRLFootnote オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLFootnote オブジェクトに増減があった場合，その変更は直ちに反映されます。

また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

- XBRL2.1 の場合：

このメソッドでは，要素と脚注の関係を定義している <link:footnoteArc> 要素のアークロール，および，その <link:footnoteArc> 要素を含んでいる脚注リンクのロールについては考慮しません。特定の値のロールを持つ脚注リンクに記述され，かつ，特定の値のアークロールを持つような <link:footnoteArc> 要素によって関連づけられた脚注を取得したい場合は，XBRLInstanceElement.getFootnoteRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。

引数：

role - 脚注のロールを指定します。ロールを持たない脚注を取得する場合は null を指定します。

戻り値：

この要素に関連づけられた脚注を表す XBRLFootnote オブジェクトのリストを返し

ます。指定したロールを持つ脚注が関連づけられていない場合は，要素数が 0 のリストを返します。

(23) getFootnotes

```
public java.util.List<XBRLFootnote> getFootnotes(java.lang.String
lang,
                                    java.lang.String role)
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられている脚注のうち，指定した言語およびロールの脚注のリストを取得します。

XBRLInstanceElement インターフェースの getFootnotes() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得する脚注のリストには含まれません。

しかし，XBRLInstanceDocument インターフェースの getFootnotes() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得する脚注のリストに含まれます。

このメソッドが返す XBRLFootnote オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLFootnote オブジェクトに増減があった場合，その変更は直ちに反映されます。

また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

- XBRL2.1 の場合：

  このメソッドでは，要素と脚注の関係を定義している <link:footnoteArc> 要素のアークロール，および，その <link:footnoteArc> 要素を含んでいる脚注リンクのロールについては考慮しません。特定の値のロールを持つ脚注リンクに記述され，かつ，特定の値のアークロールを持つような <link:footnoteArc> 要素によって関連づけられた脚注を取得したい場合は，XBRLInstanceElement.getFootnoteRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。

引数：

lang - 脚注の言語を指定します。

role - 脚注のロールを指定します。ロールを持たない脚注を取得する場合は null を指定します。

戻り値：

この要素に関連づけられた脚注を表す XBRLFootnote オブジェクトのリストを返します。指定した言語とロールを持つ脚注が関連づけられていない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(24) addFootnote

```
public XBRLFootnote addFootnote(XBRLFootnote newFootnote)
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に脚注を関連づけます。指定した XBRLFootnote オブジェクトが，すでにこの XBRLInstanceElement オブジェクトに関連づけられている場合は，何もしません。この要素がアクティブで，かつ引数 newFootnote に指定した脚注がインスタンス文書にない場合は，脚注はインスタンス文書に自動的に追加されます。

XBRL2.1 の場合，脚注がインスタンス文書へ自動的に追加されるときに，脚注の id 属性と同じ値の id 属性を持つ別の要素がすでにインスタンス文書に存在する場合は，BRLDOMRuntimeException 例外がスローされます。

この要素がアクティブで，かつ，この要素が id 属性を持っていない場合は，ID が自動的に採番されて id 属性に設定されます。また，この要素が非アクティブである場合は，このメソッドを呼び出した時点では id 属性の値は自動採番されませんが，あとでこの要素をインスタンス文書へ追加する際に脚注が関連づけられている場合は，id 属性の値が自動採番されて設定されます。id 属性の値は，XBRLInstanceElement.setId(java.lang.String) を使用して任意の値を設定することもできます。自動採番の詳細については，「3.3　ID 管理と採番」を参照してください。

- XBRL2.1 の場合：
  要素と脚注の関係を定義する <link:footnoteArc> 要素のアークロールは，XBRL2.1 の仕様で規定されている標準値である "http://www.xbrl.org/2003/arcrole/fact-footnote" が使用されます。また，<link:footnoteArc> 要素の order 属性には "1" が設定されます。カスタムアークロールを使用したい場合や，order 属性に "1" 以外の値を設定したい場合は，XBRLInstanceElement.addFootnote(XBRLFootnote newFootnote, java.lang.String arcrole, java.math.BigDecimal order) メソッドを使用してください。

引数：

newFootnote - 脚注を表す XBRLFootnote オブジェクトを指定します。

戻り値：

関連づけられた脚注を表す XBRLFootnote オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 newFootnote に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newFootnote に指定した XBRLFootnote オブジェクトを所有する XBRLInstanceDocument オブジェクトが，この要素を所有する XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］

(25) addFootnote

```
public XBRLFootnote addFootnote(XBRLFootnote newFootnote,
                                java.lang.String arcrole,
                                java.math.BigDecimal order)
```

XBRL のバージョン：

2.1

説明：

要素に脚注を関連づけます。カスタムアークロールを使用できます。

要素と脚注の関係を定義する <link:footnoteArc> 要素のアークロールは，引数 arcrole で指定された値となります。また，<link:footnoteArc> 要素の order 属性は，引数 order で指定された値となります。

引数 newFootnote に指定された XBRLFootnote オブジェクトが，引数 arcrole および order によって指定されたアークロールと order 属性を持つ <link:footnoteArc> 要素によってすでにこの XBRLInstanceElement オブジェクトに関連づけられている場合は，何もしません。

この要素がアクティブで，かつ引数 newFootnote に指定された脚注がインスタンス文書に存在しない場合は，脚注はインスタンス文書に自動的に追加されます。このとき，インスタンス文書へ自動的に追加される脚注の id 属性と同じ値の id 属性を持つ別の要素がすでにインスタンス文書に存在する場合は，XBRLDOMRuntimeException 例外がスローされます。

この要素がアクティブで，かつ，引数 newFootnote に指定された脚注がインスタンス文書に存在しない場合は，脚注はインスタンス文書に自動的に追加されます。

この要素がアクティブで，かつ，この要素が id 属性を持っていない場合は，ID が自動的に採番されて id 属性に設定されます。また，この要素が非アクティブである場合は，このメソッドを呼び出した時点では id 属性の値は自動採番されませんが，あとでこの要素をインスタンス文書へ追加する際に脚注が関連づけられている場合は，id 属性の値が自動採番されて設定されます。id 属性の値は，XBRLInstanceElement.setId(java.lang.String) を使用して任意の値を設定することもできます。自動採番の詳細については，「3.3　ID 管理と採番」を参照してください。

引数：

newFootnote - 脚注を表す XBRLFootnote オブジェクトを指定します。

arcrole - 要素と脚注の関係を定義する <link:footnoteArc> 要素のアークロールを指

定します。
order - 要素と脚注の関係を定義する <link:footnoteArc> 要素の order 属性の値を指定します。

戻り値：
関連づけられた脚注を表す XBRLFootnote オブジェクトを返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 newFootnote，arcrole または order に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newFootnote に指定された XBRLFootnote オブジェクトを所有する XBRLInstanceDocument オブジェクトが，この要素を所有する XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 arcrole に指定した値が xsd:anyURI 型でない場合にスローされます。［エラーコード：20108］
- 引数 arcrole に指定した値が絶対 URI でない場合にスローされます。［エラーコード：20109］
- 引数 arcrole に指定したアークロールが XBRL 仕様で規定された標準値で，かつ，<link:footnoteArc> 要素で利用できるものではない場合にスローされます。［エラーコード：20116］
- 引数 arcrole に指定したアークロールがカスタムアークロールで，かつ，そのカスタムアークロールの定義を指す <link:arcroleRef> 要素がインスタンス文書に存在しない場合にスローされます。［エラーコード：20118］
- 引数 arcrole に指定したアークロールがカスタムアークロールで，かつ，<link:arcroleType> 要素でそのカスタムアークロールが <link:footnoteArc> 要素で利用できるように定義されていない場合にスローされます。［エラーコード：20120］

(26) removeFootnote

```
public XBRLFootnote removeFootnote(XBRLFootnote oldFootnote)
```

XBRL のバージョン：
2.0 および 2.1

説明：
要素と脚注の関連づけを削除します。
- XBRL2.1 の場合：
  要素と脚注の関係を定義する <link:footnoteArc> 要素のアークロールは，XBRL2.1 の仕様で規定されている標準値である "http://www.xbrl.org/2003/

arcrole/fact-footnote" が仮定されます。また，<link:footnoteArc> 要素の order 属性の値には "1" が仮定されます。カスタムアークロールを持つ <link:footnoteArc> 要素によって定義されている関係や，order 属性に "1" 以外の値を持つ <link:footnoteArc> 要素によって定義されている関係を削除したい場合は，XBRLInstanceElement.removeFootnote(XBRLFootnote oldFootnote, java.lang.String arcrole, java.math.BigDecimal order) メソッドを使用してください。

引数：

oldFootnote - 関連づけを削除する脚注を表す XBRLFootnote オブジェクトを指定します。

戻り値：

関連づけが削除された脚注を表す XBRLFootnote オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 oldFootnote に null を指定した場合にスローされます。［エラーコード：20002］
- XBRL2.0 の場合：引数 oldFootnote に指定した脚注が，この要素に関連づけられていない場合にスローされます。［エラーコード：20053］
- XBRL2.1 の場合：引数 oldFootnote に指定された脚注が，アークロールが XBRL 仕様で規定されている標準値である "http://www.xbrl.org/2003/arcrole/fact-footnote" で，かつ，order 属性の値が "1" である <link:footnoteArc> 要素によってこの要素に関連づけられていない場合にスローされます。［エラーコード：20123］

(27) removeFootnote

```
public XBRLFootnote removeFootnote(XBRLFootnote oldFootnote,
                                   java.lang.String arcrole,
                                   java.math.BigDecimal order)
```

XBRL のバージョン：

2.1

説明：

引数 arcrole および order によって指定されたアークロールと order 属性を持つ <link:footnoteArc> 要素によって定義されている，この要素と脚注の関連づけを削除します。

引数：

oldFootnote - 関連づけを削除する脚注を表す XBRLFootnote オブジェクトを指定します。

arcrole - 削除する関係を定義している <link:footnoteArc> 要素のアークロールを指定します。

order - 削除する関係を定義している <link:footnoteArc> 要素の order 属性の値を指定します。

戻り値：

関連づけが削除された脚注を表す XBRLFootnote オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 oldFootnote，arcrole または order に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 oldFootnote に指定された脚注が，引数 arcrole および order によって指定されたアークロールと order 属性を持つ <link:footnoteArc> 要素によってこの要素に関連づけられていない場合にスローされます。［エラーコード：20123］

(28) getFootnoteRelationships

```
public java.util.List<XBRLFootnoteRelationship>
getFootnoteRelationships(java.lang.String extendedLinkRole,
                                          java.lang.String arcrole)
```

XBRL のバージョン：

2.1

説明：

要素と脚注の関係を表す XBRLFootnoteRelationship オブジェクトのリストを取得します。

この要素に関連づけられた脚注は，このメソッドが返すリストに含まれる各 XBRLFootnoteRelationship オブジェクトに対して getTo() メソッドを呼び出すことによって取得できます。また，このメソッドが返すリストに含まれる各 XBRLFootnoteRelationship オブジェクトに対して getFrom() メソッドを呼び出した場合の戻り値は，この要素を表す XBRLInstanceElement オブジェクト自身となります。

このメソッドが返すリストに含まれる XBRLFootnoteRelationship オブジェクトの並びは，各 XBRLFootnoteRelationship オブジェクトが表す関係を定義しているアークの order 属性の値によって昇順にソートされています。アークに order 属性が指定されていない場合は，1 が指定されたものと仮定します。詳細については，XBRLRelationship インターフェースの「6.7.7(4)　getOrder」を参照してください。アークの order 属性の値が同じである XBRLFootnoteRelationship オブジェクトが複数ある場合，それらの XBRLFootnoteRelationship オブジェクトの並び順は不定です。

このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLFootnoteRelationship オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッ

ドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

extendedLinkRole - 関係を定義しているアークを含むリファレンスリンクのロールを指定します。

arcrole - 関係を定義しているアークのアークロールを指定します。

戻り値：

引数 extendedLinkRole で指定されたロールを持つ脚注リンクに属し，かつ，引数 arcrole で指定されたアークロールを持つアークによって定義された，この要素と脚注の関係を表す XBRLFootnoteRelationship オブジェクトのリストを返します。指定された関係が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 extendedLinkRole または arcrole に null を指定した場合にスローされます。［エラーコード：20002］

（29）isNilAttributeSpecified

```
public boolean isNilAttributeSpecified()
```

XBRL のバージョン：

2.1

説明：

要素に xsi:nil 属性が定義されていたかどうかを調べます。

戻り値：

この要素に xsi:nil 属性が定義されていた場合，true を返します。xsi:nil 属性が定義されていない場合は false を返します。

InstanceDocument の createItem メソッド，createNumericItem メソッド，createNonNumericItem で新たに生成した XBRLInstanceElement のオブジェクトにおいて，本メソッドを実行した場合は，常に false を返します。

（30）getDefinitionRelationships

```
public java.util.List<XBRLDefinitionRelationship>
getDefinitionRelationships(java.lang.String extendedLinkRole,
                           java.lang.String arcrole)
```

XBRL のバージョン：

2.1

説明：

要素宣言とほかの要素宣言の関係を表す XBRLDefinitionRelationship オブジェクトのリストを取得します。
この要素宣言に関連づけられたほかの要素宣言は，このメソッドが返すリストに含まれる各 XBRLDefinitionRelationship オブジェクトに対して getTo() メソッドを呼び出すことによって取得できます。また，このメソッドが返すリストに含まれる各 XBRLDefinitionRelationship オブジェクトに対して getFrom() メソッドを呼び出した場合の戻り値は，この XBRLTaxonomyElement オブジェクト自身となります。
このメソッドが返すリストに含まれる XBRLDefinitionRelationship オブジェクトの並びは，各 XBRLDefinitionRelationship オブジェクトが表す関係を定義しているアークの order 属性の値によって昇順にソートされています。アークに order 属性が指定されていない場合は，1 が指定されたものと仮定します。アークの order 属性の値が同じである XBRLDefinitionRelationship オブジェクトが複数ある場合，XBRLDefinitionRelationship オブジェクトの並び順は不定です。
このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLDefinitionRelationship オブジェクトに増減があった場合，その変更は直ちに反映されます。このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

extendedLinkRole - 関係を定義しているアークを含む定義リンクのロールを指定します。
arcrole - 関係を定義しているアークのアークロールを指定します。

戻り値：

引数 extendedLinkRole で指定されたロールを持つ定義リンクに属し，かつ，引数 arcrole で指定されたアークロールを持つアークによって定義された，この要素の要素宣言とほかの要素宣言の関係を表す XBRLDefinitionRelationship オブジェクトのリストを返します。指定された関係が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 extendedLinkRole または arcrole に null を指定した場合にスローされます。［エラーコード：20002］

6.6.13 XBRLInstanceItem インターフェース

```
public interface XBRLInstanceItem
extends XBRLInstanceElement
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムを表します。
XBRLInstanceItem オブジェクトは，XBRLInstanceDocument オブジェクトまたは XBRLInstanceTuple オブジェクトの子となることができます。
非アクティブな XBRLInstanceItem オブジェクトを，XBRLInstanceDocument オブジェクトまたはアクティブな XBRLInstanceTuple オブジェクトの子として設定すると，子として設定された XBRLInstanceItem オブジェクトは非アクティブな状態からアクティブな状態へと変化します。
また，XBRLInstanceItem オブジェクトが非アクティブな XBRLInstanceTuple オブジェクトの子である場合はその XBRLInstanceItem オブジェクトも非アクティブです。しかし，親である非アクティブな XBRLInstanceTuple オブジェクトまたはその祖先に当たる XBRLInstanceTuple オブジェクトが，XBRLInstanceDocument オブジェクトまたはアクティブな XBRLInstanceTuple オブジェクトの子として設定され，アクティブになった場合，その子または子孫である XBRLInstanceItem オブジェクトも同時にアクティブな状態へと変化します。
このようにして XBRLInstanceItem オブジェクトが非アクティブな状態からアクティブな状態へ変化する際に，XBRLInstanceItem オブジェクトが参照する XBRLContext オブジェクトが非アクティブである場合，その XBRLContext オブジェクトは XBRLInstanceDocument の addContext(XBRLContext) によってインスタンス文書へ自動的に追加されます。このとき，インスタンス文書へ自動的に追加される XBRLContext オブジェクトの識別子と同じ識別子を持つ別の XBRLContext オブジェクトがすでにインスタンス文書に存在する場合は，XBRLDOMRuntimeException 例外がスローされます。
XBRL2.1 の場合：XBRLInstanceItem オブジェクトが非アクティブな状態からアクティブな状態へ変化する際に，XBRLInstanceItem オブジェクトが参照する XBRLUnit2 オブジェクトが非アクティブである場合，その XBRLUnit2 オブジェクトは XBRLInstanceDocument.addUnit(XBRLUnit2) によってインスタンス文書へ自動的に追加されます。このとき，インスタンス文書へ自動的に追加される XBRLUnit2 オブジェクトの識別子と同じ識別子を持つ別の要素がすでに存在する場合は，XBRLDOMRuntimeException 例外がスローされます。

フィールドの概要：

XBRLInstanceItem インターフェースのフィールドの概要を次の表に示します。

表 6-32　XBRLInstanceItem インターフェースのフィールドの概要

項番	説明	
1	static int	MONETARY_ITEM このアイテムのデータ型は xbrli:monetary です。

項番	説明	
2	static int	SHARES_ITEM このアイテムのデータ型は xbrli:shares です。
3	static int	PURE_ITEM このアイテムのデータ型は xbrli:pure です。
4	static int	FRACTION_ITEM このアイテムのデータ型は分数です。
5	static int	DECIMAL_ITEM このアイテムのデータ型は xsd:decimal です。
6	static int	FLOAT_ITEM このアイテムのデータ型は xsd:float です。
7	static int	DOUBLE_ITEM このアイテムのデータ型は xsd:double です。
8	static int	INTEGER_ITEM このアイテムのデータ型は xsd:integer です。
9	static int	NONPOSITIVEINTEGER_ITEM このアイテムのデータ型は xsd:nonPositiveInteger です。
10	static int	NEGATIVEINTEGER_ITEM このアイテムのデータ型は xsd:negativeInteger です。
11	static int	LONG_ITEM このアイテムのデータ型は xsd:long です。
12	static int	INT_ITEM このアイテムのデータ型は xsd:int です。
13	static int	SHORT_ITEM このアイテムのデータ型は xsd:short です。
14	static int	BYTE_ITEM このアイテムのデータ型は xsd:byte です。
15	static int	NONNEGATIVEINTEGER_ITEM このアイテムのデータ型は xsd:nonNegativeInteger です。
16	static int	UNSIGNEDLONG_ITEM このアイテムのデータ型は xsd:unsignedLong です。
17	static int	UNSIGNEDINT_ITEM このアイテムのデータ型は xsd:unsignedInt です。
18	static int	UNSIGNEDSHORT_ITEM このアイテムのデータ型は xsd:unsignedShort です。
19	static int	UNSIGNEDBYTE_ITEM このアイテムのデータ型は xsd:unsignedByte です。
20	static int	POSITIVEINTEGER_ITEM このアイテムのデータ型は xsd:positiveInteger です。
21	static int	STRING_ITEM このアイテムのデータ型は xsd:string です。

項番	説明	
22	static int	BOOLEAN_ITEM このアイテムのデータ型は xsd:boolean です。
23	static int	HEXBINARY_ITEM このアイテムのデータ型は xsd:hexBinary です。
24	static int	BASE64BINARY_ITEM このアイテムのデータ型は xsd:base64Binary です。
25	static int	ANYURI_ITEM このアイテムのデータ型は xsd:anyURI です。
26	static int	QNAME_ITEM このアイテムのデータ型は xsd:QName です。
27	static int	NOTATION_ITEM このアイテムのデータ型は xsd:NOTATION です。
28	static int	DURATION_ITEM このアイテムのデータ型は xsd:duration です。
29	static int	DATETIME_ITEM このアイテムのデータ型は xbrli:dateUnion です。
30	static int	TIME_ITEM このアイテムのデータ型は xsd:time です。
31	static int	DATE_ITEM このアイテムのデータ型は xsd:date です。
32	static int	GYEARMONTH_ITEM このアイテムのデータ型は xsd:gYearMonth です。
33	static int	GYEAR_ITEM このアイテムのデータ型は xsd:gYear です。
34	static int	GMONTHDAY_ITEM このアイテムのデータ型は xsd:gMonthDay です。
35	static int	GDAY_ITEM このアイテムのデータ型は xsd:gDay です。
36	static int	GMONTH_ITEM このアイテムのデータ型は xsd:gMonth です。
37	static int	NORMALIZEDSTRING_ITEM このアイテムのデータ型は xsd:normalizedString です。
38	static int	TOKEN_ITEM このアイテムのデータ型は xsd:token です。
39	static int	LANGUAGE_ITEM このアイテムのデータ型は xsd:language です。
40	static int	NAME_ITEM このアイテムのデータ型は xsd:Name です。
41	static int	NCNAME_ITEM このアイテムのデータ型は xsd:NCName です。

項番	説明	
42	static int	OTHER_ITEM このアイテムのデータ型は不明です。

メソッドの概要：

XBRLInstanceItem インターフェースのメソッドの概要を次の表に示します。

表 6-33　XBRLInstanceItem インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLTaxonomyItem	getTaxonomyItem() アイテムの要素宣言を取得します。	2.0 および 2.1	該当しない
2	int	getItemType() アイテムのデータ型を表すコードを取得します。	2.0 および 2.1	該当しない
3	int	getBaseItemType() アイテムのベースデータ型を表すコードを取得します。	2.1	該当しない
4	boolean	isNumeric() アイテムが数値または非数値のどちらであるかを調べます。	2.0 および 2.1	該当しない
5	XBRLContext	getContext() アイテムが参照するコンテキスト要素を取得します。	2.0 および 2.1	該当しない
6	void	setContext(XBRLContext context) アイテムが参照するコンテキスト要素を設定します。	2.0 および 2.1	該当しない
7	XBRLUnit2	getUnit() アイテムが参照する単位要素を取得します。	2.1	該当しない
8	void	setUnit(XBRLUnit2 unit) アイテムが参照する単位要素を設定します。	2.1	該当しない
9	XBRLPrecision	getPrecision() アイテムの precision 属性値を取得します。	2.1	該当しない
10	void	setPrecision(XBRLPrecision precision) アイテムの precision 属性値を設定します。	2.1	該当しない
11	XBRLDecimals	getDecimals() アイテムの decimals 属性値を取得します。	2.1	該当しない
12	void	setDecimals(XBRLDecimals decimals) アイテムの decimals 属性値を設定します。	2.1	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
13	boolean	hasValue() アイテムに値が設定されているかどうかを調べます。	2.0 および 2.1	該当しない
14	java.lang.Object	getValue() アイテムの値を取得します。	2.0 および 2.1	該当しない
15	void	setValue(java.lang.Object value) アイテムの値を設定します。	2.0 および 2.1	該当しない
16	java.lang.String	getString() アイテムの値の文字列表現を取得します。	2.0 および 2.1	該当しない
17	void	setString(java.lang.String value) アイテムの値の文字列表現を設定します。	2.0 および 2.1	該当しない
18	XBRLDimensionalRelationshipSet	getDimensionalRelationshipSet() アイテムの要素宣言に関連づいている DRS を取得します。	2.1	1.0
19	boolean	validateDimensions() DRS に含まれているすべてのロールで，アイテムに対して XBRL Dimensions の妥当性検証を実行します。	2.1	1.0
20	boolean	validateDimensions(java.lang.String role) DRS に含まれているロールのうち，引数 role に指定されたロールで，アイテムに対して XBRL Dimensions の妥当性検証を実行します。	2.1	1.0

（1） getTaxonomyItem

```
public XBRLTaxonomyItem getTaxonomyItem()
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムの要素宣言を取得します。

戻り値：

このアイテムの要素宣言を表す XBRLTaxonomyItem オブジェクトを返します。

（2） getItemType

```
public int getItemType()
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムのデータ型を表すコードを取得します。

このメソッドが返す値は，XBRL のバージョンによって異なります。利用できるデータ型の詳細については，XBRLTaxonomyItem インターフェースの「6.8.17(1) getItemType」を参照してください。

戻り値：
このアイテムの型を表すコードを返します。

(3) getBaseItemType

```
public int getBaseItemType()
```

XBRL のバージョン：
2.1

説明：
アイテムのベースデータ型を表すコードを取得します。
ベースデータ型は，ユーザーデータ型の派生元となっているデータ型のことです。
アイテムがユーザーデータ型である場合（getItemType() メソッドが OTHER_ITEM を返す場合）は，getBaseItemType() メソッドを呼び出すことによって，派生元となっているデータ型を調べることができます。派生元のデータ型が不明な場合は，getBaseItemType() メソッドは OTHER_ITEM を返します。
このアイテムがユーザーデータ型でない場合（getItemType() メソッドが OTHER_ITEM 以外を返す場合）は，このメソッドは getItemType() メソッドが返す値と同じ値を返します。

戻り値：
このアイテムのベースデータ型を表すコードを返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

(4) isNumeric

```
public boolean isNumeric()
```

XBRL のバージョン：
2.0 および 2.1

説明：
アイテムが数値または非数値のどちらであるかを調べます。
このアイテムが数値か非数値のどちらかは，このアイテムのデータ型によって異なります。アイテムのデータ型と isNumeric() メソッドの戻り値との対応関係については，XBRLTaxonomyItem インターフェースの「6.8.17(3)　isNumeric」を参照してください。

戻り値：

このアイテムが数値である場合は true，非数値である場合は false を返します。

（5） getContext

```
public XBRLContext getContext()
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムが参照するコンテキスト要素を取得します。

戻り値：

コンテキスト要素を表す XBRLContext オブジェクトを返します。

- XBRL2.0 の場合：このメソッドが返したオブジェクトは，XBRLNumericContext または XBRLNonNumericContext のどちらか一方にキャストできます。XBRLNumericContext と XBRLNonNumericContext のどちらにキャストできるかは，isNumeric() を呼び出すことで判定できます。
- XBRL2.1 の場合：このメソッドが返したオブジェクトは，XBRLContext2 にキャストできます。

（6） setContext

```
public void setContext(XBRLContext context)
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムが参照するコンテキスト要素を設定します。
このアイテムがアクティブで，かつ引数 context で指定したコンテキスト要素がインスタンス文書にない場合は，コンテキスト要素はインスタンス文書に自動的に追加されます。このとき，インスタンス文書へ自動的に追加されるコンテキスト要素の id 属性と同じ値の id 属性を持つ別のコンテキスト要素が，すでにインスタンス文書に存在する場合は，XBRLDOMRuntimeException がスローされます。

引数：

context - このアイテムが参照するコンテキスト要素を指定します。

- XBRL2.0 の場合：この引数に指定するオブジェクトは，XBRLNumericContext か XBRLNonNumericContext のどちらかでなければなりません。XBRLNumericContext と XBRLNonNumericContext のどちらを指定する必要があるかは，このアイテムのデータ型に依存します。
- XBRL2.1 の場合：この引数に指定するオブジェクトは，XBRLContext2 である必要があります。

例外：

XBRLDOMRuntimeException -

- 引数 context に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 context に指定した XBRLContext オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceItem オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- XBRL2.0 の場合：引数 context が表すコンテキスト要素が，このアイテムのデータ型に適合しない場合にスローされます。［エラーコード：20032］
- XBRL2.1 の場合：引数 context に指定したコンテキスト要素の時期要素の内容が，このアイテムの要素宣言での xbrli:periodType 属性の指定と矛盾している場合にスローされます。xbrli:periodType 属性の値が "instant" の場合は，時期要素の内容は XBRLPeriod.INSTANT である必要があります。また，xbrli:periodType 属性の値が "duration" の場合は，時期要素の内容は XBRLPeriod.FOREVER か XBRLPeriod.STARTDATE_ENDDATE である必要があります。［エラーコード：20140］

（7） getUnit

```
public XBRLUnit2 getUnit()
```

XBRL のバージョン：

2.1

説明：

アイテムが参照する単位要素を取得します。
単位要素を参照するのは数値アイテムだけであり，非数値アイテムは単位要素を参照しません。よって，数値アイテムに対してだけ，このメソッドを呼び出せます。
非数値アイテムに対してこのメソッドを呼び出した場合は，
XBRLDOMRuntimeException がスローされます。アイテムが数値か非数値かについては，XBRLInstanceItem.isNumeric() メソッドで調べることができます。

戻り値：

単位要素を表す XBRLUnit2 オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- このアイテムが非数値の場合にスローされます。［エラーコード：20137］

（8） setUnit

```
public void setUnit(XBRLUnit2 unit)
```

XBRL のバージョン：

2.1

説明：

アイテムが参照する単位要素を設定します。
このアイテムがアクティブで，かつ，引数 unit で指定された単位要素がインスタンス文書に存在しない場合は，単位要素はインスタンス文書に自動的に追加されます。このとき，インスタンス文書へ自動的に追加される単位要素の id 属性と同じ値の id 属性を持つ別の要素がすでにインスタンス文書に存在する場合は，XBRLDOMRuntimeException 例外がスローされます。
単位要素を設定できるのは，数値アイテムに対してだけです。非数値アイテムに対してこのメソッドを呼び出した場合は，XBRLDOMRuntimeException がスローされます。アイテムが数値か非数値かについては，XBRLInstanceItem.isNumeric() メソッドで調べることができます。
アイテムが参照する単位要素には，アイテムのベースデータ型によって次に示す制約があります。これらの制約に違反する単位要素を設定しようとした場合は，XBRLDOMRuntimeException がスローされます。

表 6-34　アイテムのベースデータ型による単位要素の制約

項番	アイテムのベースデータ型	単位要素の制約
1	MONETARY_ITEM	次の制約をすべて満たす必要があります。 • 単位要素の内容は，一つの基本単位によって構成されなければなりません。つまり，XBRLUnit2.getUnitType() メソッドの戻り値が XBRLUnit2.MULTIPLY で，かつ，XBRLUnit2.getMeasures() メソッドが返すリストのサイズは 1 である必要があります。 • 基本単位を表す XBRLQName オブジェクトの名前空間 URI は "http://www.xbrl.org/2003/iso4217" である必要があります。 • 基本単位を表す XBRLQName オブジェクトのローカルパートは，ISO 4217 通貨コードである必要があります。なお，XBRL DOM では，この項目はチェックしません。
2	SHARES_ITEM	次の制約をすべて満たす必要があります。 • 単位要素の内容は，一つの基本単位によって構成されなければなりません。つまり，XBRLUnit2.getUnitType() メソッドの戻り値が XBRLUnit2.MULTIPLY で，かつ，XBRLUnit2.getMeasures() メソッドが返すリストのサイズは 1 である必要があります。 • 基本単位を表す XBRLQName オブジェクトの名前空間 URI は "http://www.xbrl.org/2003/instance" である必要があります。 • 基本単位を表す XBRLQName オブジェクトのローカルパートは "shares" である必要があります。

項番	アイテムのベースデータ型	単位要素の制約
3	PURE_ITEM	次の制約をすべて満たす必要があります。 • 単位要素の内容は，一つの基本単位によって構成されなければなりません。つまり，XBRLUnit2.getUnitType() メソッドの戻り値が XBRLUnit2.MULTIPLY で，かつ，XBRLUnit2.getMeasures() メソッドが返すリストのサイズは 1 である必要があります。 • 基本単位を表す XBRLQName オブジェクトの名前空間 URI は "http://www.xbrl.org/2003/instance" である必要があります。 • 基本単位を表す XBRLQName オブジェクトのローカルパートは "pure" である必要があります。

このアイテムのベースデータ型が上記の表に記載されているもの以外である場合は，このアイテムに設定する単位要素についての制約はありません。

引数：

unit - このアイテムが参照する単位要素を指定します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- このアイテムが非数値の場合にスローされます。［エラーコード：20137］
- 引数 unit に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 unit に指定された XBRLUnit2 オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceItem オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 unit に指定された XBRLUnit2 オブジェクトが，上記の表の制約に違反している場合にスローされます。［エラーコード：20127］

(9) getPrecision

```
public XBRLPrecision getPrecision()
```

XBRL のバージョン：

2.1

説明：

アイテムの precision 属性値を取得します。

このメソッドは，アイテムが数値で，かつ，アイテムのベースデータ型が FRACTION_ITEM 以外である場合に呼び出せます。このアイテムが非数値か，または，ベースデータ型が FRACTION_ITEM である場合にこのメソッドを呼び出すと，XBRLDOMRuntimeException がスローされます。

戻り値：

このアイテムの precision 属性の値を返します。precision 属性がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- このアイテムが非数値か，または，ベースデータ型が FRACTION_ITEM である場合にスローされます。［エラーコード：20136］

（10）setPrecision

```
public void setPrecision(XBRLPrecision precision)
```

XBRL のバージョン：

2.1

説明：

アイテムの precision 属性値を設定します。

precision 属性と decimals 属性はどちらか一方だけを設定できます。decimals 属性が設定されているアイテムに対してこのメソッドを呼び出した場合は，decimals 属性は自動的に削除されます。

このメソッドは，アイテムが数値で，かつ，アイテムのベースデータ型が FRACTION_ITEM 以外である場合に呼び出せます。このアイテムが非数値か，または，ベースデータ型が FRACTION_ITEM である場合にこのメソッドを呼び出すと，XBRLDOMRuntimeException がスローされます。

このアイテムの xsi:nil 属性に "true" が指定されている場合は，precision 属性を設定できません。xsi:nil 属性に "true" が指定されている場合にこのメソッドを呼び出すと，XBRLDOMRuntimeException がスローされます。

引数：

precision - このアイテムの precision 属性に設定する値を指定します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- このアイテムが非数値か，または，ベースデータ型が FRACTION_ITEM である場合にスローされます。［エラーコード：20136］
- このアイテムの xsi:nil 属性に "true" が指定されている場合にスローされます。［エラーコード：20130］
- 引数 precision に null を指定した場合にスローされます。［エラーコード：20002］

（11）getDecimals

```
public XBRLDecimals getDecimals()
```

XBRL のバージョン：

2.1

説明：

アイテムの decimals 属性値を取得します。

このメソッドは，アイテムが数値で，かつ，アイテムのベースデータ型が FRACTION_ITEM 以外である場合に呼び出せます。このアイテムが非数値か，または，ベースデータ型が FRACTION_ITEM である場合にこのメソッドを呼び出すと，XBRLDOMRuntimeException がスローされます。

戻り値：

このアイテムの decimals 属性の値を返します。decimals 属性がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- このアイテムが非数値か，または，ベースデータ型が FRACTION_ITEM である場合にスローされます。［エラーコード：20136］

(12) setDecimals

```
public void setDecimals(XBRLDecimals decimals)
```

XBRL のバージョン：

2.1

説明：

アイテムの decimals 属性値を設定します。

precision 属性と decimals 属性はどちらか一方だけを設定できます。precision 属性が設定されているアイテムに対してこのメソッドを呼び出した場合は，precision 属性は自動的に削除されます。

このメソッドは，アイテムが数値で，かつ，アイテムのベースデータ型が FRACTION_ITEM 以外である場合に呼び出せます。このアイテムが非数値か，または，ベースデータ型が FRACTION_ITEM である場合にこのメソッドを呼び出すと，XBRLDOMRuntimeException がスローされます。

このアイテムの xsi:nil 属性に "true" が指定されている場合は，decimals 属性を設定できません。xsi:nil 属性に "true" が指定されている場合にこのメソッドを呼び出すと，XBRLDOMRuntimeException がスローされます。

引数：

decimals - このアイテムの decimals 属性に設定する値を指定します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- このアイテムが非数値か，または，ベースデータ型が FRACTION_ITEM である場合にスローされます。［エラーコード：20136］
- このアイテムの xsi:nil 属性に "true" が指定されている場合にスローされます。［エラーコード：20130］
- 引数 decimals に null を指定した場合にスローされます。［エラーコード：20002］

（13）hasValue

```
public boolean hasValue()
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムに値が設定されているかどうかを調べます。
XBRLInstanceDocument の createItem() で生成された直後の XBRLInstanceItem オブジェクトには値は設定されていません。つまり，hasValue() は false を返します。

戻り値：

このアイテムに値が設定されている場合は true，設定されていない場合は false を返します。

（14）getValue

```
public java.lang.Object getValue()
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムの値を取得します。
このメソッドが返すオブジェクトは，アイテムのデータ型によって異なります。このアイテムに値が設定されていない場合，このメソッドは XBRLDOMRuntimeException をスローします。
getValue() メソッドが返すオブジェクトを次の表に示します。

表 6-35　getValue() メソッドが返すオブジェクト

項番	アイテムのデータ型	getValue() メソッドが返すオブジェクト
1	MONETARY_ITEM	java.math.BigDecimal
2	SHARES_ITEM	java.math.BigDecimal
3	PURE_ITEM	java.math.BigDecimal
4	FRACTION_ITEM	XBRLFraction

項番	アイテムのデータ型	getValue() メソッドが返すオブジェクト
5	DECIMAL_ITEM	java.math.BigDecimal
6	FLOAT_ITEM	java.lang.Float
7	DOUBLE_ITEM	java.lang.Double
8	INTEGER_ITEM	java.math.BigInteger
9	NONPOSITIVEINTEGER_ITEM	java.math.BigInteger
10	NEGATIVEINTEGER_ITEM	java.math.BigInteger
11	LONG_ITEM	java.lang.Long
12	INT_ITEM	java.lang.Integer
13	SHORT_ITEM	java.lang.Short
14	BYTE_ITEM	java.lang.Byte
15	NONNEGATIVEINTEGER_ITEM	java.math.BigInteger
16	UNSIGNEDLONG_ITEM	java.math.BigInteger
17	UNSIGNEDINT_ITEM	java.math.BigInteger
18	UNSIGNEDSHORT_ITEM	java.math.BigInteger
19	UNSIGNEDBYTE_ITEM	java.math.BigInteger
20	POSITIVEINTEGER_ITEM	java.math.BigInteger
21	STRING_ITEM	java.lang.String
22	BOOLEAN_ITEM	java.lang.Boolean
23	HEXBINARY_ITEM	XBRLHexBinary
24	BASE64BINARY_ITEM	XBRLBase64Binary
25	ANYURI_ITEM	java.lang.String
26	QNAME_ITEM	XBRLQName
27	NOTATION_ITEM	XBRLQName
28	DURATION_ITEM	XBRLDuration
29	DATETIME_ITEM	XBRLDateUnion
30	TIME_ITEM	XBRLTime
31	DATE_ITEM	XBRLDate
32	GYEARMONTH_ITEM	XBRLGYearMonth
33	GYEAR_ITEM	XBRLGYear
34	GMONTHDAY_ITEM	XBRLGMonthDay
35	GDAY_ITEM	XBRLGDay
36	GMONTH_ITEM	XBRLGMonth
37	NORMALIZEDSTRING_ITEM	java.lang.String
38	TOKEN_ITEM	java.lang.String
39	LANGUAGE_ITEM	java.lang.String

項番	アイテムのデータ型	getValue() メソッドが返すオブジェクト
40	NAME_ITEM	java.lang.String
41	NCNAME_ITEM	java.lang.String
42	OTHER_ITEM	java.lang.String

XBRL2.1 の場合：このアイテムがユーザーデータ型（OTHER_ITEM）で，かつ，ベースデータ型が FRACTION_ITEM である場合，このメソッドは，XBRLFraction.toString() メソッドが返す文字列と同じ形式の文字列を表す java.lang.String オブジェクトを返します。ユーザーデータ型で，かつ，ベースデータ型が FRACTION_ITEM であるアイテムの値を表す XBRLFraction オブジェクトを使用したい場合は，このメソッドが返した文字列をそのまま XBRLFraction クラスのコンストラクタに渡します。

戻り値：

アイテムの値を返します。

例外：

XBRLDOMRuntimeException -

- このアイテムに値が設定されていない場合にスローされます。［エラーコード：20026］

（15）setValue

```
public void setValue(java.lang.Object value)
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムの値を設定します。

このメソッドの引数に指定できるオブジェクトは，アイテムのデータ型によって異なります。setValue() メソッドの引数に指定できるオブジェクトを次の表に示します。

表 6-36　setValue() メソッドの引数に指定できるオブジェクト

項番	アイテムのデータ型	setValue() メソッドの引数に指定できるオブジェクト
1	MONETARY_ITEM	java.math.BigDecimal
2	SHARES_ITEM	java.math.BigDecimal
3	PURE_ITEM	java.math.BigDecimal
4	FRACTION_ITEM	XBRLFraction
5	DECIMAL_ITEM	java.math.BigDecimal
6	FLOAT_ITEM	java.lang.Float

項番	アイテムのデータ型	setValue() メソッドの引数に指定できるオブジェクト
7	DOUBLE_ITEM	java.lang.Double
8	INTEGER_ITEM	java.math.BigInteger
9	NONPOSITIVEINTEGER_ITEM	java.math.BigInteger
10	NEGATIVEINTEGER_ITEM	java.math.BigInteger
11	LONG_ITEM	java.lang.Long
12	INT_ITEM	java.lang.Integer
13	SHORT_ITEM	java.lang.Short
14	BYTE_ITEM	java.lang.Byte
15	NONNEGATIVEINTEGER_ITEM	java.math.BigInteger
16	UNSIGNEDLONG_ITEM	java.math.BigInteger
17	UNSIGNEDINT_ITEM	java.math.BigInteger
18	UNSIGNEDSHORT_ITEM	java.math.BigInteger
19	UNSIGNEDBYTE_ITEM	java.math.BigInteger
20	POSITIVEINTEGER_ITEM	java.math.BigInteger
21	STRING_ITEM	java.lang.String
22	BOOLEAN_ITEM	java.lang.Boolean
23	HEXBINARY_ITEM	XBRLHexBinary
24	BASE64BINARY_ITEM	XBRLBase64Binary
25	ANYURI_ITEM	java.lang.String
26	QNAME_ITEM	XBRLQName
27	NOTATION_ITEM	XBRLQName
28	DURATION_ITEM	XBRLDuration
29	DATETIME_ITEM	XBRLDateUnion
30	TIME_ITEM	XBRLTime
31	DATE_ITEM	XBRLDate
32	GYEARMONTH_ITEM	XBRLGYearMonth
33	GYEAR_ITEM	XBRLGYear
34	GMONTHDAY_ITEM	XBRLGMonthDay
35	GDAY_ITEM	XBRLGDay
36	GMONTH_ITEM	XBRLGMonth
37	NORMALIZEDSTRING_ITEM	java.lang.String
38	TOKEN_ITEM	java.lang.String
39	LANGUAGE_ITEM	java.lang.String
40	NAME_ITEM	java.lang.String

項番	アイテムのデータ型	setValue() メソッドの引数に指定できるオブジェクト
41	NCNAME_ITEM	java.lang.String
42	OTHER_ITEM	java.lang.String

インスタンス文書には，このアイテムの値として，引数 value に指定したオブジェクトの toString() が返す文字列が設定されます。ただし，このアイテムが xsd:QName 型または xsd:NOTATION 型の場合は，toString() が返す文字列「{namespaceURI}localPart」ではなく，「prefix:localPart」という形式の文字列が設定されます。
「prefix」の部分に該当する名前空間接頭辞は，XBRLInstanceDocument オブジェクトに登録されている名前空間宣言から取得されます。名前空間宣言は，XBRLInstanceDocument の addNamespaceDeclaration() を使用してあらかじめ登録しておきます。引数 value に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が登録されていない場合は，例外がスローされます。
このアイテムの値としてインスタンス文書に設定される文字列表現を厳密に指定したい場合は，このメソッドの代わりに setString() を使用します。
XBRL2.1 の場合：このアイテムがユーザーデータ型（OTHER_ITEM）で，かつ，ベースデータ型が FRACTION_ITEM である場合，このメソッドに指定する文字列の形式は XBRLFraction クラスのコンストラクタへ指定する文字列と同じ形式である必要があります。XBRLFraction クラスのコンストラクタへ指定できない文字列をこのメソッドに指定した場合は，XBRLDOMRuntimeException がスローされます。ユーザーデータ型で，かつ，ベースデータ型が FRACTION_ITEM であるアイテムに対して，XBRLFraction オブジェクトから値を設定したい場合は，XBRLFraction.toString() メソッドが返す文字列をそのまま setValue() メソッドの引数に指定します。
XBRL2.1 の場合：xsi:nil 属性に "true" が指定されている場合は，アイテムの値は設定できないため，このメソッドを呼び出してはいけません。xsi:nil 属性に "true" が指定されているアイテムに対してこのメソッドを呼び出した場合は，XBRLDOMRuntimeException がスローされます。

引数：

value - アイテムの値を指定します。

例外：

XBRLDOMRuntimeException -

- XBRL2.1 の場合：このアイテムの xsi:nil 属性に "true" が指定されている場合にスローされます。［エラーコード：20130］
- 引数 value に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 value の型が，このアイテムのデータ型に合わない場合にスローされます。［エラーコード：20033］

- このアイテムが xsd:QName 型または xsd:NOTATION 型で，かつ，引数 value に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］
- このアイテムが xsd:QName 型または xsd:NOTATION 型で，かつ引数 value に指定した XBRLQName オブジェクトが名前空間に属さず，かつ XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］
- XBRL2.1 の場合：このアイテムがユーザーデータ型（OTHER_ITEM）で，かつ，ベースデータ型が FRACTION_ITEM である場合に，XBRLFraction クラスのコンストラクタへ指定できない文字列を引数 value に指定した場合にスローされます。［エラーコード：20138］
- XBRL2.1 の場合：このアイテムのデータ型が次のどれかで，かつ，引数 value に指定された整数値がこのアイテムのデータ型の範囲外である場合にスローされます。［エラーコード：20014］
  ・NONPOSITIVEINTEGER_ITEM
  ・NEGATIVEINTEGER_ITEM
  ・NONNEGATIVEINTEGER_ITEM
  ・UNSIGNEDLONG_ITEM
  ・UNSIGNEDINT_ITEM
  ・UNSIGNEDSHORT_ITEM
  ・UNSIGNEDBYTE_ITEM
  ・POSITIVEINTEGER_ITEM
- このアイテムのデータ型が ANYURI_ITEM で，かつ，引数 value に指定された値が xsd:anyURI 型でない場合にスローされます。［エラーコード：20108］
- このアイテムのデータ型が QNAME_ITEM または NOTATION_ITEM で，かつ，引数 value に指定された XBRLQName オブジェクトのローカルパートが xsd:NCName 型でない場合にスローされます。［エラーコード：20108］
- XBRL2.1 の場合：このアイテムのデータ型が NAME_ITEM で，かつ，引数 value に指定された値が xsd:Name 型でない場合にスローされます。［エラーコード：20108］
- XBRL2.1 の場合：このアイテムのデータ型が NCNAME_ITEM で，かつ，引数 value に指定された値が xsd:NCName 型でない場合にスローされます。［エラーコード：20108］

(16) getString

```
public java.lang.String getString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムの値の文字列表現を取得します。

このメソッドが返す文字列は，インスタンス文書にこのアイテムの値として記述されている文字列です。このメソッドが返す文字列は，getValue().toString() が返す文字列と異なる場合があります。例えば，このアイテムが xsd:QName 型または xsd:NOTATION 型である場合，getString() は「prefix:localPart」という形式の文字列を返しますが，getValue().toString() は「{namespaceURI}localPart」という形式の文字列を返します。また，このアイテムが xsd:float 型で，かつインスタンス文書にこのアイテムの値として「1.0e2」と記述されていた場合，getString() は「1.0e2」という文字列を返しますが，getValue().toString() は「100.0」という文字列を返します。

このアイテムに値が設定されていない場合，このメソッドは XBRLDOMRuntimeException をスローします。

XBRL2.1 の場合：アイテムのベースデータ型が FRACTION_ITEM である場合，インスタンス文書に記述される値は文字列ではないため，このメソッドで文字列表現を取得できません。ベースデータ型が FRACTION_ITEM であるアイテムに対してこのメソッドを呼び出した場合は，XBRLDOMRuntimeException がスローされます。

戻り値：

このアイテムの値の文字列表現を返します。

アイテムの型が xsd:QName または xsd:NOTATION の場合，名前空間を正規化すると，正規化の前後で QName の接頭辞が異なります。正規化前は，既存の文書で指定された接頭辞を返します。正規化後は，正規化によって変更された接頭辞を返します。

例外：

XBRLDOMRuntimeException -

- このアイテムに値が設定されていない場合にスローされます。［エラーコード：20026］
- XBRL2.1 の場合：このアイテムのベースデータ型が FRACTION_ITEM である場合にスローされます。［エラーコード：20139］
- このアイテムの内容に子要素が含まれている場合にスローされます。［エラーコード：20061］

（17）setString

```
public void setString(java.lang.String value)
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムの値の文字列表現を設定します。

setValue() では，引数に指定したオブジェクトの toString() の戻り値が，このアイ

テムの値としてインスタンス文書に書き込まれます。一方，setString() では，引数 value に指定した文字列がそのまま，このアイテムの値としてインスタンス文書に書き込まれます。このアイテムの値としてインスタンス文書に書き込まれる文字列表現を厳密に指定したい場合は，setValue() ではなく setString() を使用します。

- XBRL2.1 の場合：
  アイテムのベースデータ型が FRACTION_ITEM である場合，インスタンス文書に記述される値は文字列ではないため，このメソッドで文字列表現を設定できません。ベースデータ型が FRACTION_ITEM であるアイテムに対してこのメソッドを呼び出した場合は，XBRLDOMRuntimeException がスローされます。
- XBRL2.1 の場合：
  xsi:nil 属性に "true" が指定されている場合は，アイテムの値は設定できないため，このメソッドを呼び出さないでください。xsi:nil 属性に "true" が指定されているアイテムに対してこのメソッドを呼び出した場合は，XBRLDOMRuntimeException がスローされます。

引数：

value - このアイテムの値の文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- XBRL2.1 の場合：このアイテムのベースデータ型が FRACTION_ITEM である場合にスローされます。［エラーコード：20139］
- XBRL2.1 の場合：このアイテムの xsi:nil 属性に "true" が指定されている場合にスローされます。［エラーコード：20130］
- 引数 value に null を指定した場合にスローされます。［エラーコード：20002］
- このアイテムが xsd:QName 型または xsd:NOTATION 型で，かつ引数 value に指定した QName の名前空間接頭辞に対応する名前空間宣言が，XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20022］
- 引数 value に指定された文字列表現が不正な場合にスローされます。[エラーコード：20029]

(18) getDimensionalRelationshipSet

```
public XBRLDimensionalRelationshipSet
getDimensionalRelationshipSet()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

アイテムの要素宣言に関連づいている DRS を取得します。

戻り値：

このアイテムの要素宣言に関連づいている DRS を返します。アイテムの要素宣言に関連づいている DRS がない場合は，null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］

（19）validateDimensions

```
public boolean validateDimensions()
                              throws XBRLDOMException
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

アイテムに対して，DRS に含まれているすべてのロールで XBRL Dimensions の妥当性検証を実行します。

戻り値：

このアイテムに対して，DRS に含まれているすべてのロールで XBRL Dimensions の妥当性検証を実行し，妥当である場合は true，妥当でない場合は false を返します。また，このアイテムに DRS がない場合は XBRL Dimensions の妥当性検証を実行しないで，true を返します。

例外：

XBRLDOMException -

- XBRL Dimensions の妥当性検証でエラーが検出された場合にスローされます。
  ［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］

(20) validateDimensions

```
public boolean validateDimensions(java.lang.String role)
  throws XBRLDOMException
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

アイテムに対して，DRS に含まれているロールのうち，引数 role に指定されたロールで XBRL Dimensions の妥当性検証を実行します。

引数：

role - XBRL Dimensions の妥当性検証を実行するロールを指定します。

戻り値：

このアイテムに対して，DRS に含まれているロールのうち，引数 role に指定されたロールで XBRL Dimensions の妥当性検証を実行し，妥当である場合は true，妥当でない場合は false を返します。また，このアイテムに DRS がない場合は XBRL Dimensions の妥当性検証を実行しないで true を返します。

例外：

XBRLDOMException -

- XBRL Dimensions の妥当性検証でエラーが検出された場合にスローされます。
  ［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 role に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 role に空の URI 参照を指定した場合にスローされます。［エラーコード：20110］
- 引数 role に DRS に存在しないロールを指定した場合にスローされます。［エラーコード：22102］

6.6.14　XBRLInstanceNode インターフェース

```
public interface XBRLInstanceNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

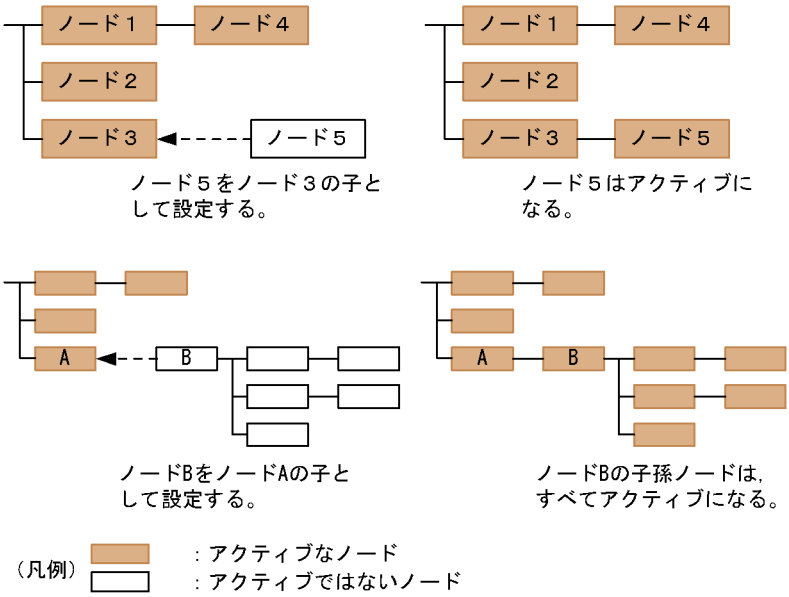
インスタンス文書を構成するノードを表します。

ノードは，ほかのノードの子となることができます。getParentNode() を使用すると，任意のノードの親ノードを取得できます。一方，あるノードの子ノードを取得する方法は，ノードの種類によって異なります。例えば，XBRLEntity オブジェクトの子である XBRLIdentifier オブジェクトを取得する場合は，XBRLEntity.getIdentifier() を使用します。子ノードを取得する方法については，ノードを表す各インターフェースの説明を参照してください。

ノードは，アクティブか非アクティブのどちらかの状態を持ちます。アクティブなノードは，インスタンス文書内に存在するノード，つまり，XBRLInstanceDocument オブジェクトからオブジェクトツリーをたどることで到達できるノードです。一方，非アクティブなノードは，インスタンス文書にないノードです。ノードがアクティブかどうかは，isActive() で調べることができます。

非アクティブなノードをアクティブなノードの子として設定すると，子として設定されたノードはアクティブな状態へと変化します。また，あるノードの親ノードが非アクティブである場合，そのノードも非アクティブですが，親である非アクティブなノードがアクティブな状態へと変化すると，同時にその子ノードもアクティブになります。この状態の変化は，親から子，さらに孫へと伝播します。つまり，あるノードがアクティブになると，その子孫ノードもすべてアクティブになり，あるノードが非アクティブになると，その子孫ノードもすべて非アクティブになります。アクティブになるノードの範囲を次の図に示します。

図 6-14　アクティブになるノードの範囲



あるノードが，非アクティブな状態からアクティブな状態へと変化する際に，次の処理が自動的に実行されます。

- アクティブになったノードが XML 要素に対応する場合，その XML 要素およびその要素が持つグローバル属性の名前空間接頭辞が，XBRLInstanceDocument オブジェクトに設定されている名前空間接頭辞に設定されます。

ノードの種類によっては，アクティブになった時点で，上記以外のさまざまな処理が実行されます。詳細については，ノードを表す各インターフェースの説明を参照してください。

フィールドの概要：

XBRLInstanceNode インターフェースのフィールドの概要を次の表に示します。

表 6-37　XBRLInstanceNode インターフェースのフィールドの概要

項番	説明	
1	static int	INSTANCE_DOCUMENT_NODE このノードは，インスタンス文書を表します。
2	static int	SCHEMA_REF_NODE このノードは，<link:schemaRef> 要素を表します。
3	static int	LINKBASE_REF_NODE このノードは，<link:linkbaseRef> 要素を表します。
4	static int	ROLE_REF_NODE このノードは，<link:roleRef> 要素を表します。
5	static int	ARCROLE_REF_NODE このノードは，<link:arcroleRef> 要素を表します。
6	static int	NUMERIC_CONTEXT_NODE このノードは，数値コンテキスト要素を表します。
7	static int	NON_NUMERIC_CONTEXT_NODE このノードは，非数値コンテキスト要素を表します。
8	static int	CONTEXT2_NODE このノードは，コンテキスト要素を表します。
9	static int	UNIT2_NODE このノードは，単位要素を表します。
10	static int	ITEM_NODE このノードは，アイテムを表します。
11	static int	TUPLE_NODE このノードは，タプルを表します。
12	static int	FOOTNOTE_NODE このノードは，脚注を表します。
13	static int	ENTITY_NODE このノードは，エンティティ要素を表します。
14	static int	IDENTIFIER_NODE このノードは，識別子（identifier）要素を表します。

項番	説明	
15	static int	SEGMENT_NODE このノードは，セグメント要素を表します。
16	static int	PERIOD_NODE このノードは，時期要素を表します。
17	static int	UNIT_NODE このノードは，単位要素を表します。
18	static int	MEASURE_NODE このノードは，計量要素を表します。
19	static int	OPERATOR_NODE このノードは，演算子要素を表します。
20	static int	SCENARIO_NODE このノードは，シナリオ要素を表します。
21	static int	EXPLICIT_MEMBER_NODE このノードは，<xbrldi:explicitMember> 要素を表します。
22	static int	TYPED_MEMBER_NODE このノードは，<xbrldi:typedMember> 要素を表します。

メソッドの概要：

XBRLInstanceNode インターフェースのメソッドの概要を次の表に示します。

表 6-38　XBRLInstanceNode インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	int	getNodeType() ノードの型を表すコードを取得します。	2.0 および 2.1
2	XBRLInstanceDocument	getOwnerInstanceDocument() ノードを所有する XBRLInstanceDocument オブジェクトを取得します。	2.0 および 2.1
3	boolean	isActive() ノードがアクティブかどうか（インスタンス文書に存在し，ツリーをたどって到達できるか）を調べます。	2.0 および 2.1
4	XBRLInstanceNode	getParentNode() ノードが関連づけられている親ノードを取得します。	2.0 および 2.1
5	XBRLInstanceNode	cloneNode(boolean deep) ノードを複製します。	2.0 および 2.1
6	org.w3c.dom.Node	getW3CDOMNode() ノードに対応する org.w3c.dom.Node オブジェクトを取得します。	2.0 および 2.1

（1） getNodeType

```
public int getNodeType()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ノードの型を表すコードを取得します。

このメソッドが返す値は，XBRL のバージョンによって異なります。利用できるノードの型と XBRL のバージョンの関係を次の表に示します。

表 6-39　利用できるノードの型と XBRL のバージョンの関係

項番	ノードの型	XBRL2.0	XBRL2.1
1	INSTANCE_DOCUMENT_NODE	○	○
2	SCHEMA_REF_NODE	×	○
3	LINKBASE_REF_NODE	○	○
4	ROLE_REF_NODE	×	○
5	ARCROLE_REF_NODE	×	○
6	NUMERIC_CONTEXT_NODE	○	×
7	NON_NUMERIC_CONTEXT_NODE	○	×
8	CONTEXT2_NODE	×	○
9	UNIT2_NODE	×	○
10	ITEM_NODE	○	○
11	TUPLE_NODE	○	○
12	FOOTNOTE_NODE	○	○
13	ENTITY_NODE	○	○
14	IDENTIFIER_NODE	○	○
15	SEGMENT_NODE	○	○
16	PERIOD_NODE	○	○
17	UNIT_NODE	○	×
18	MEASURE_NODE	○	×
19	OPERATOR_NODE	○	×
20	SCENARIO_NODE	○	○
21	EXPLICIT_MEMBER_NODE ※	×	○
22	TYPED_MEMBER_NODE ※	×	○

（凡例）○：利用できます。　×：利用できません。
注※　XBRL Dimensions1.0 の場合だけ存在します。

戻り値：

このノードの型を表すコードを返します。

（2） getOwnerInstanceDocument

```
public XBRLInstanceDocument getOwnerInstanceDocument()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ノードを所有する XBRLInstanceDocument オブジェクトを取得します。

戻り値：

このノードを所有する XBRLInstanceDocument オブジェクトを返します。このノードが XBRLInstanceDocument の場合は，常に null を返します。

（3） isActive

```
public boolean isActive()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ノードがアクティブかどうかを調べます。アクティブなノードは，インスタンス文書内に存在するノード，つまり，XBRLInstanceDocument オブジェクトからオブジェクトツリーをたどることで到達できるノードです。

戻り値：

このノードがアクティブな場合は true，アクティブでない場合は false を返します。このノードが XBRLInstanceDocument の場合は，常に true を返します。

（4） getParentNode

```
public XBRLInstanceNode getParentNode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ノードが関連づけられている親ノードを取得します。
このメソッドが返す親ノードの型は，このノードの型によって異なります。
getParentNode メソッドが返す親ノードの型を次の表に示します。

表 6-40　getParentNode メソッドが返す親ノードの型

項番	このノードの型	メソッドが返す親ノードの型
1	INSTANCE_DOCUMENT_NODE	このメソッドは常に null を返します。
2	SCHEMA_REF_NODE	INSTANCE_DOCUMENT_NODE
3	LINKBASE_REF_NODE	INSTANCE_DOCUMENT_NODE

項番	このノードの型	メソッドが返す親ノードの型
4	ROLE_REF_NODE	INSTANCE_DOCUMENT_NODE
5	ARCROLE_REF_NODE	INSTANCE_DOCUMENT_NODE
6	NUMERIC_CONTEXT_NODE	INSTANCE_DOCUMENT_NODE
7	NON_NUMERIC_CONTEXT_NODE	INSTANCE_DOCUMENT_NODE
8	CONTEXT2_NODE	INSTANCE_DOCUMENT_NODE
9	UNIT2_NODE	INSTANCE_DOCUMENT_NODE
10	ITEM_NODE	INSTANCE_DOCUMENT_NODE または TUPLE_NODE
11	TUPLE_NODE	INSTANCE_DOCUMENT_NODE または TUPLE_NODE
12	FOOTNOTE_NODE	INSTANCE_DOCUMENT_NODE
13	ENTITY_NODE	XBRL2.0 の場合：NUMERIC_CONTEXT_NODE または NON_NUMERIC_CONTEXT_NODE XBRL2.1 の場合：CONTEXT2_NODE
14	IDENTIFIER_NODE	ENTITY_NODE
15	SEGMENT_NODE	ENTITY_NODE
16	PERIOD_NODE	XBRL2.0 の場合：NUMERIC_CONTEXT_NODE または NON_NUMERIC_CONTEXT_NODE XBRL2.1 の場合：CONTEXT2_NODE
17	UNIT_NODE	NUMERIC_CONTEXT_NODE または NON_NUMERIC_CONTEXT_NODE
18	MEASURE_NODE	UNIT_NODE または OPERATOR_NODE
19	OPERATOR_NODE	UNIT_NODE または OPERATOR_NODE
20	SCENARIO_NODE	XBRL2.0 の場合：NUMERIC_CONTEXT_NODE または NON_NUMERIC_CONTEXT_NODE XBRL2.1 の場合：CONTEXT2_NODE
21	EXPLICIT_MEMBER_NODE ※	SEGMENT_NODE または SCENARIO_NODE
22	TYPED_MEMBER_NODE ※	SEGMENT_NODE または SCENARIO_NODE

注※　XBRL Dimensions1.0 の場合だけ存在します。

戻り値：

このノードの親ノードを返します。このノードが XBRLInstanceDocument の場合は，常に null を返します。また，このノードがほかのノードの子ノードとして接続されていない場合も，null を返します。

(5) cloneNode

```
public XBRLInstanceNode cloneNode(boolean deep)
```

XBRL のバージョン：

2.0 および 2.1

説明：

ノードを複製します。

このメソッドが返すノードはアクティブではなく，また，親ノードを持ちません。つまり，このメソッドが返す XBRLInstanceNode オブジェクトの isActive() は false を返し，また，getParentNode() は null を返します。

ノードを複製する処理は，ノードの型によって異なります。ノードの複製処理を次の表に示します。

表 6-41　ノードの複製処理

項番	ノードの型	説明
1	INSTANCE_DOCUMENT_NODE	引数 deep が true の場合は，インスタンス文書に含まれているコンテキスト要素，アイテム，タプル，脚注がすべて複製されます。 引数 deep が false の場合は，インスタンス文書に含まれているコンテキスト要素，アイテム，タプル，脚注はすべて複製されません。つまり，複製されたインスタンス文書は空です。ただし，名前空間宣言は複製されます。 複製された XBRLInstanceDocument オブジェクトは，複製元の XBRLInstanceDocument オブジェクトが参照している XBRLTaxonomyDocument オブジェクトの複製を参照します。つまり，XBRLInstanceDocument の getTaxonomyDocument() の戻り値は，複製元の XBRLInstanceDocument オブジェクトと複製先の XBRLInstanceDocument オブジェクトで異なります。ただし，それぞれが参照する XBRLTaxonomyDocument オブジェクトの内容は同じです。 XBRL2.0 の場合：インスタンス文書の読み込み時にリンクベースの遅延解析が指定されていて，かつリンクが未解析の場合は，複製処理の前にインスタンス文書中のリンクベースの遅延解析が実行されます。このとき，インスタンス文書が参照しているタクソノミー文書に含まれるリンクベース，および，インスタンス文書に記述された <link:linkbaseRef> 要素が参照しているリンクベースは，未解析であっても解析されません。 複製された XBRLInstanceDocument オブジェクトに対応する org.w3c.dom.Document オブジェクトは新規に作成されます。また，インスタンス文書のルート要素が XML 文書のルート要素として設定されます。

項番	ノードの型	説明
		XBRL2.1 の場合：複製されたインスタンス文書のルート要素は id 属性を持ちません。 引数 deep に true が指定された場合は，インスタンス文書の内容が複製されます。複製された各ノードが id 属性を持つかどうかについては，この表の各ノードの説明を参照してください。 XBRL2.1 の場合：インスタンス文書の ID を管理する XBRLIdManager オブジェクトは新規に作成されるため，複製元のインスタンス文書と複製されたインスタンス文書とで ID を管理する XBRLIdManager オブジェクトは異なります。ただし，XBRLIdManager オブジェクトに設定されている XBRLIdCreator オブジェクトは引き継がれます。つまり，複製元の XBRLInstanceDocument オブジェクトと複製された XBRLInstanceDocument オブジェクトとで，getIdManager() メソッドが返す XBRLIdManager オブジェクトは異なりますが，getIdManager().getIdCreator() によって取得できる XBRLIdCreator オブジェクトは同一のものとなります。このため，インスタンス文書を複製する際にコンテキスト要素などの id 属性の値が採番し直されます。このときに複製元と同じ XBRLIdCreator オブジェクトを使用して ID が生成されます。 XBRL2.1 の場合：引数 deep が false の場合は，複製元の <link:schemaRef> 要素や <link:linkbaseRef> 要素は複製されません。このため，引数 deep に false を指定してインスタンス文書を複製した場合は，タクソノミー文書を構築する際の出発点となったタクソノミー本体およびリンクベースのリストを XBRLTaxonomyDocument.getStartingPoints() メソッドによって取得し，それらを指す <link:schemaRef> 要素および <link:linkbaseRef> 要素を追加する必要があります。このとき，各要素の xlink:href 属性に指定する URI は，絶対 URI，相対 URI のどちらでもかまいません。ただし，インスタンス文書をストリームへ出力してから再度読み込んだ際に，それらの URI は適切な文書へ解決されなければなりません。なお，必要な <link:schemaRef> 要素および <link:linkbaseRef> 要素を追加しなかった場合は，インスタンス文書をストリームへ出力してから再度読み込んだ場合に，タクソノミー文書を構成するタクソノミー本体またはリンクベースに不足が生じるおそれがあるため注意してください。

項番	ノードの型	説明
2	SCHEMA_REF_NODE	サブツリーが存在しないため，引数 deep の値は影響しません。
3	LINKBASE_REF_NODE	
4	ROLE_REF_NODE	
5	ARCROLE_REF_NODE	
6	NUMERIC_CONTEXT_NODE	引数 deep の値に関係なく，常にサブツリーが複製されます。 複製されたコンテキスト要素の id 属性の値は採番し直されます。
7	NON_NUMERIC_CONTEXT_NODE	
8	CONTEXT2_NODE	
9	UNIT2_NODE	サブツリー※1 が存在しないため，引数 deep の値は影響しません。複製された単位要素の id 属性の値は採番し直されます。
10	ITEM_NODE	アイテムにはサブツリー※1 が存在しないため，引数 deep の値は影響しません。 複製されたアイテムは，複製元と同一のコンテキスト要素を参照します。 XBRL2.1 の場合：数値アイテムを複製した場合，複製されたアイテムは，複製元と同一の単位要素を参照します。 XBRL2.0 の場合：複製されたアイテムの id 属性の値は採番し直されます。 XBRL2.1 の場合：複製されたアイテムは id 属性を持ちません。
11	TUPLE_NODE	引数 deep が true の場合は，タプルの子要素もすべて複製されます。引数 deep が false の場合は，子要素は複製されません。 XBRL2.0 の場合：複製されたタプルの id 属性の値は採番し直されます。 XBRL2.1 の場合：複製されたタプルは id 属性を持ちません。
12	ENTITY_NODE	引数 deep の値に関係なく，常にサブツリーが複製されます。
13	PERIOD_NODE	
14	UNIT_NODE	
15	OPERATOR_NODE	
16	IDENTIFIER_NODE	サブツリー※1 がないため，引数 deep の値は影響しません。

項番	ノードの型	説明
17	SEGMENT_NODE	XBRL Dimensions1.0 ではない場合：サブツリー※1 が存在しないため，引数 deep の値は影響しません。 XBRL Dimensions1.0 の場合：引数 deep が true の場合は，セグメント要素に含まれている <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素がすべて複製されます。 引数 deep が false の場合は，セグメント要素に含まれている <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素がすべて複製されません。
18	MEASURE_NODE	サブツリー※1 がないため，引数 deep の値は影響しません。
19	SCENARIO_NODE	XBRL Dimensions1.0 ではない場合：サブツリー※1 が存在しないため，引数 deep の値は影響しません。 XBRL Dimensions1.0 の場合：引数 deep が true の場合は，シナリオ要素に含まれている <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素がすべて複製されます。 引数 deep が false の場合は，シナリオ要素に含まれている <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素がすべて複製されません。
20	FOOTNOTE_NODE	サブツリー※1 がないため，引数 deep の値は影響しません。 XBRL2.1 の場合：複製された脚注は id 属性を持ちません。
21	EXPLICIT_MEMBER_NODE ※2	サブツリー※1 が存在しないため，引数 deep の値は影響しません。
22	TYPED_MEMBER_NODE ※2	サブツリー※1 が存在しないため，引数 deep の値は影響しません。

注※1　ここでのサブツリーとは，XBRL DOM のノード（XBRLInstanceNode オブジェクト）によって構築されるツリーでのサブツリーのことです。W3C DOM のノード（org.w3c.dom.Node オブジェクト）によって構築されるツリーでのサブツリーではないので注意してください。
例えば，単位要素の子として記述される <xbrli:divide> 要素は，XBRL DOM では XBRLUnit2 オブジェクトの子ノードとしては表現されませんが，W3C DOM では <xbrli:divide> 要素は <xbrli:unit> 要素を表すノードの子ノードとして表現されます。そのため，XBRLUnit2 オブジェクトを複製した場合は，W3C DOM ではサブツリーとなっている <xbrli:divide> 要素を表すノードやその子孫（<xbrli:unitNumerator> 要素を表すノードや <xbrli:unitDenominator> 要素を表すノード）も複製されます。
これは，データ型が FRACTION_ITEM であるアイテム，XHTML のタグを内容として持つ脚注，

任意の子要素を記述できるセグメント要素やシナリオ要素についても同じです。
注※2　XBRL Dimensions1.0 の場合だけ存在します。

引数：

deep - ノードのサブツリーを複製する場合は true，サブツリーを複製しない場合は false を指定します。

戻り値：

このノードの複製を返します。

（6）getW3CDOMNode

```
public org.w3c.dom.Node getW3CDOMNode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ノードに対応する org.w3c.dom.Node オブジェクトを取得します。XBRL の仕様で規定されていない属性や要素を参照したい場合は，このメソッドで取得した DOM ツリーを使用します。
このメソッドが返す org.w3c.dom.Node オブジェクトの型は，このノードの型によって異なります。getW3CDOMNode メソッドが返す org.w3c.dom.Node オブジェクトを次の表に示します。

表 6-42　getW3CDOMNode メソッドが返す org.w3c.dom.Node オブジェクト

項番	XBRLInstanceNode オブジェクトの型	org.w3c.dom.Node オブジェクトの型	説明
1	INSTANCE_DOCUMENT_NODE	org.w3c.dom.Document	インスタンス文書を表します。
2	SCHEMA_REF_NODE	org.w3c.dom.Element	<link:schemaRef> 要素を表します。
3	LINKBASE_REF_NODE	org.w3c.dom.Element	<link:linkbaseRef> 要素を表します。
4	ROLE_REF_NODE	org.w3c.dom.Element	<link:roleRef> 要素を表します。
5	ARCROLE_REF_NODE	org.w3c.dom.Element	<link:arcroleRef> 要素を表します。
6	NUMERIC_CONTEXT_NODE	org.w3c.dom.Element	<xbrli:numericContext> 要素を表します。
7	NON_NUMERIC_CONTEXT_NODE	org.w3c.dom.Element	<xbrli:nonNumericContext> 要素を表します。
8	CONTEXT2_NODE	org.w3c.dom.Element	<xbrli:context> 要素を表します。

項番	XBRLInstanceNode オブジェクトの型	org.w3c.dom.Node オブジェクトの型	説明
9	UNIT2_NODE	org.w3c.dom.Element	<xbrli:unit> 要素を表します。
10	ITEM_NODE	org.w3c.dom.Element	<xbrli:item> 要素の代替要素を表します。
11	TUPLE_NODE	org.w3c.dom.Element	<xbrli:tuple> 要素の代替要素を表します。
12	FOOTNOTE_NODE	org.w3c.dom.Element	<link:footnote> 要素を表します。
13	ENTITY_NODE	org.w3c.dom.Element	<xbrli:entity> 要素を表します。
14	IDENTIFIER_NODE	org.w3c.dom.Element	<xbrli:identifier> 要素を表します。
15	SEGMENT_NODE	org.w3c.dom.Element	要素を表します。
16	PERIOD_NODE	org.w3c.dom.Element	<xbrli:period> 要素を表します。
17	UNIT_NODE	org.w3c.dom.Element	<xbrli:unit> 要素を表します。
18	MEASURE_NODE	org.w3c.dom.Element	<xbrli:measure> 要素を表します。
19	OPERATOR_NODE	org.w3c.dom.Element	<xbrli:operator> 要素を表します。
20	SCENARIO_NODE	org.w3c.dom.Element	<xbrli:scenario> 要素を表します。
21	EXPLICIT_MEMBER_NODE ※	org.w3c.dom.Element	<xbrldi:explicitMember> 要素を表します。
22	TYPED_MEMBER_NODE ※	org.w3c.dom.Element	<xbrldi:typedMember> 要素を表します。

注※　XBRL Dimensions1.0 の場合だけ存在します。

このメソッドが返した org.w3c.dom.Node オブジェクト，およびその org.w3c.dom.Node オブジェクトが属する DOM ツリーに対する更新操作の範囲は限定されます。getW3CDOMNode メソッドが返す org.w3c.dom.Node オブジェクトに対して許可される更新操作を次の表に示します。

表 6-43　org.w3c.dom.Node オブジェクトに対して許可される更新操作

項番	XBRLInstanceNode オブジェクトの型	許可される更新操作
1	INSTANCE_DOCUMENT_NODE	更新操作は許可されません。
2	SCHEMA_REF_NODE	更新操作は許可されません。

項番	XBRLInstanceNode オブジェクトの型	許可される更新操作
3	LINKBASE_REF_NODE	更新操作は許可されません。
4	ROLE_REF_NODE	更新操作は許可されません。
5	ARCROLE_REF_NODE	更新操作は許可されません。
6	NUMERIC_CONTEXT_NODE	更新操作は許可されません。
7	NON_NUMERIC_CONTEXT_NODE	更新操作は許可されません。
8	CONTEXT2_NODE	更新操作は許可されません。
9	UNIT2_NODE	更新操作は許可されません。
10	ITEM_NODE	次の属性以外の属性の追加・更新・削除が許可されます。 • id • XBRL2.0 の場合：numericContext • XBRL2.0 の場合：nonNumericContext • xmlns による名前空間宣言 • XBRL2.1 の場合：contextRef • XBRL2.1 の場合：unitRef • XBRL2.1 の場合：precision • XBRL2.1 の場合：decimals • XBRL2.1 の場合：xsl:nil
11	TUPLE_NODE	次の属性以外の属性の追加・更新・削除が許可されます。 • id • xmlns による名前空間宣言 • XBRL2.1 の場合：xsl:nil
12	FOOTNOTE_NODE	子孫ノードに対する任意の更新操作が許可されます。
13	ENTITY_NODE	更新操作は許可されません。
14	IDENTIFIER_NODE	更新操作は許可されません。
15	SEGMENT_NODE	XBRL Dimensions1.0 ではない場合：子孫ノードに対する任意の更新操作が許可されます。 XBRL Dimensions1.0 の場合：次の子孫ノード以外の子孫ノードに対する任意の更新操作が許可されます。 • <xbrldi:explicitMember> 要素 • <xbrldi:typedMember> 要素
16	PERIOD_NODE	更新操作は許可されません。
17	UNIT_NODE	更新操作は許可されません。
18	MEASURE_NODE	更新操作は許可されません。
19	OPERATOR_NODE	更新操作は許可されません。
20	SCENARIO_NODE	XBRL Dimensions1.0 ではない場合：子孫ノードに対する任意の更新操作が許可されます。 XBRL Dimensions1.0 の場合：次の子孫ノード以外の子孫ノードに対する任意の更新操作が許可されます。 • <xbrldi:explicitMember> 要素 • <xbrldi:typedMember> 要素

項番	XBRLInstanceNode オブジェクトの型	許可される更新操作
21	EXPLICIT_MEMBER_NODE ※	子孫ノードに対する任意の更新操作が許可されます。
22	TYPED_MEMBER_NODE ※	子孫ノードに対する任意の更新操作が許可されます。

注※　XBRL Dimensions1.0 の場合だけ存在します。

許可されていない更新操作をした場合は，XBRL DOM が正常に動作しないおそれがあります。

戻り値：

このノードに対応する org.w3c.dom.Node オブジェクトを返します。

6.6.15　XBRLInstanceTuple インターフェース

```
public interface XBRLInstanceTuple
extends XBRLInstanceElement
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に記述されているタプルを表します。

タプルの子要素として記述できる要素は，タクソノミー本体で XML Schema の構文を使用して定義されます。しかし，XBRL DOM では，任意のアイテムおよびタプルをタプルの子要素として設定できるようになっていて，XBRLInstanceTuple インターフェースを使用してタプルの内容を編集した結果がタクソノミー本体での定義に従っているかどうかについてはチェックしません。

メソッドの概要：

XBRLInstanceTuple インターフェースのメソッドの概要を次の表に示します。

表 6-44　XBRLInstanceTuple インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLTaxonomyTuple	getTaxonomyTuple() タプルの要素宣言を取得します。	2.0 および 2.1
2	java.util.List<XBRLInstanceElement>	getChildElements() タプルのすべての子要素のリストを取得します。	2.0 および 2.1
3	java.util.List<XBRLInstanceElement>	getChildElements(java.lang.String namespaceURI, java.lang.String name) 指定した名前の子要素のリストを取得します。	2.0 および 2.1

項番	説明		XBRL のバージョン
4	XBRLInstanceElement	appendChildElement(XBRLInstanceElement newElement) タプルの子要素のリストの末尾に，指定した子要素を追加します。	2.0 および 2.1
5	XBRLInstanceElement	insertChildElementBefore(XBRLInstanceElement newElement, XBRLInstanceElement refElement) 引数 newElement に指定された要素を，このタプルの既存の子要素の前に挿入します。	2.0 および 2.1
6	XBRLInstanceElement	replaceChildElement(XBRLInstanceElement newElement, XBRLInstanceElement oldElement) 引数 oldElement に指定した子要素を，引数 newElement に指定した要素で置き換えます。	2.0 および 2.1
7	XBRLInstanceElement	removeChildElement(XBRLInstanceElement oldElement) タプルの子要素を削除します。	2.0 および 2.1

（1） getTaxonomyTuple

```
public XBRLTaxonomyTuple getTaxonomyTuple()
```

XBRL のバージョン：

2.0 および 2.1

説明：

タプルの要素宣言を取得します。

戻り値：

このタプルの要素宣言を表す XBRLTaxonomyTuple オブジェクトを返します。

（2） getChildElements

```
public java.util.List<XBRLInstanceElement> getChildElements()
```

XBRL のバージョン：

2.0 および 2.1

説明：

タプルのすべての子要素のリストを取得します。子要素は，アイテムまたはタプルです。

このメソッドが返すリスト中での XBRLInstanceElement オブジェクトの順序は，インスタンス文書での要素の出現順と同じです。

このメソッドが返す XBRLInstanceElement オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLInstanceElement オブジェクトに増減があった場合，その変更は直

ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，
java.lang.UnsupportedOperationException がスローされます。

戻り値：
このタプルの子要素をすべて含む XBRLInstanceElement オブジェクトのリストを返します。このタプルに子要素がない場合は，要素数が 0 のリストを返します。

(3) getChildElements

```
public java.util.List<XBRLInstanceElement>
getChildElements(java.lang.String namespaceURI,
                                        java.lang.String name)
```

XBRL のバージョン：
2.0 および 2.1

説明：
指定した名前の子要素のリストを取得します。子要素は，アイテムまたはタプルです。
タプルの子要素として同じ名前の要素が二つ以上存在することがあります。このため，このメソッドは，指定した名前の子要素を，XBRLInstanceElement オブジェクトのリストとして返します。リスト中での XBRLInstanceElement オブジェクトの順序は，インスタンス文書での要素の出現順と同じです。
このメソッドが返す XBRLInstanceElement オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLInstanceElement オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，
java.lang.UnsupportedOperationException がスローされます。

引数：
namespaceURI - 子要素の名前空間 URI を指定します。
name - 子要素の名前を指定します。

戻り値：
引数 namespaceURI で指定した名前空間 URI に属し，引数 name で指定した名前を持つ子要素をすべて含む XBRLInstanceElement オブジェクトのリストを返します。指定した子要素がない場合は，要素数が 0 のリストを返します。

例外：
XBRLDOMRuntimeException -
- 引数 namespaceURI または name に null を指定した場合にスローされます。［エラーコード：20002］

(4) appendChildElement

```
public XBRLInstanceElement appendChildElement(XBRLInstanceElement
newElement)
```

XBRL のバージョン：

2.0 および 2.1

説明：

タプルの子要素のリストの末尾に，指定した子要素を追加します。子要素は，アイテムまたはタプルです。

指定した要素がすでにほかのオブジェクトの子要素となっている場合は，最初に既存の親から指定した要素が削除されます。

XBRL2.1 の場合：xsi:nil 属性に "true" が指定されている場合は，タプルに子要素を追加できないため，このメソッドを呼び出さないでください。xsi:nil 属性に "true" が指定されているタプルに対してこのメソッドを呼び出した場合は，XBRLDOMRuntimeException がスローされます。

引数：

newElement - 追加する子要素を表す XBRLInstanceElement オブジェクトを指定します。

戻り値：

追加された子要素を表す XBRLInstanceElement オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- XBRL2.1 の場合：このタプルの xsi:nil 属性に "true" が指定されている場合にスローされます。[エラーコード：20130]
- 引数 newElement に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 newElement に指定した XBRLInstanceElement オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceTuple オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。[エラーコード：20004]
- 引数 newElement に指定した XBRLInstanceElement オブジェクトが，この XBRLInstanceTuple オブジェクトの祖先か，またはこの XBRLInstanceTuple オブジェクト自身である場合にスローされます。[エラーコード：20054]
- このタプルがアクティブで，かつ引数 newElement に指定した要素が非アクティブで，かつ引数 newElement に指定した要素（タプルの場合は子孫要素を含む）の id 属性と同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。[エラーコード：20028]

(5) insertChildElementBefore

```
public XBRLInstanceElement
```

```
insertChildElementBefore(XBRLInstanceElement newElement,
                                                XBRLInstanceElement
refElement)
```

XBRL のバージョン：

2.0 および 2.1

説明：

引数 newElement に指定された要素を，このタプルの既存の子要素の前に挿入します。子要素は，アイテムまたはタプルです。

引数 refElement が null の場合は，引数 newElement に指定した要素は，このタプルの子要素のリストの末尾に挿入されます。指定した要素がすでにほかのオブジェクトの子要素となっている場合は，最初に既存の親から指定した要素が削除されます。引数 newElement に指定した XBRLInstanceElement オブジェクトが引数 refElement に指定した XBRLInstanceElement オブジェクトと同じ場合は，このメソッドは何もしません。

XBRL2.1 の場合：xsi:nil 属性に "true" が指定されている場合は，タプルに子要素を追加できないため，このメソッドを呼び出さないでください。xsi:nil 属性に "true" が指定されているタプルに対してこのメソッドを呼び出した場合は，XBRLDOMRuntimeException がスローされます。

引数：

newElement - 挿入する子要素を表す XBRLInstanceElement オブジェクトを指定します。

refElement - 参照する子要素を表す XBRLInstanceElement オブジェクトを指定します。引数 newElement で指定した子要素は，この子要素の直前に挿入されます。

戻り値：

挿入された子要素を表す XBRLInstanceElement オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- XBRL2.1 の場合：このタプルの xsi:nil 属性に "true" が指定されている場合にスローされます。［エラーコード：20130］
- 引数 newElement に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newElement に指定した XBRLInstanceElement オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceTuple オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 newElement に指定した XBRLInstanceElement オブジェクトが，この XBRLInstanceTuple オブジェクトの祖先か，またはこの XBRLInstanceTuple オブジェクト自身である場合にスローされます。［エラーコード：20054］
- このタプルがアクティブで，かつ引数 newElement に指定した要素が非アクティ

ブで，かつ引数 newElement に指定した要素（タプルの場合は子孫要素を含む）の id 属性と同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。［エラーコード：20028］

- 引数 refElement に指定した子要素が，このタプルの子要素でない場合にスローされます。［エラーコード：20055］

（6） replaceChildElement

```
public XBRLInstanceElement replaceChildElement(XBRLInstanceElement
newElement,
                                    XBRLInstanceElement oldElement)
```

XBRL のバージョン：

2.0 および 2.1

説明：

引数 oldElement に指定した子要素を，引数 newElement に指定した要素で置き換えます。子要素は，アイテムまたはタプルです。
指定した要素がすでにほかのオブジェクトの子要素となっている場合は，最初に既存の親から指定した要素が削除されます。引数 newElement に指定した XBRLInstanceElement オブジェクトが引数 oldElement に指定した XBRLInstanceElement オブジェクトと同じ場合は，このメソッドは何もしません。
置き換える前に，id 属性の値が重複していないかがチェックされます。このチェックは，引数 oldElement を含む置き換えられる前の文書の状態を基に実行されます。例えば，このタプルがアクティブで，かつ，引数 newElement に指定した要素が非アクティブで，かつ，引数 newElement に指定した要素（タプルの場合は子孫要素を含む）の id 属性の値が，引数 oldElement で指定した要素（タプルの場合は子孫要素を含む）の id 属性の値と重複している場合は，XBRLDOMRuntimeException がスローされます。置き換えの対象となる要素の id 属性と同じ値の id 属性を持つ別の要素をインスタンス文書に追加したい場合は，XBRLInstanceTuple.removeChildElement(XBRLInstanceElement oldElement) メソッドで置き換えの対象となる要素を削除してから，追加してください。

引数：

newElement - 挿入する子要素を表す XBRLInstanceElement オブジェクトを指定します。
oldElement - 置き換えの対象となる子要素を表す XBRLInstanceElement オブジェクトを指定します。

戻り値：

置き換えの対象となった子要素を表す XBRLInstanceElement オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 newElement または oldElement に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newElement に指定した XBRLInstanceElement オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceTuple オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 newElement に指定した XBRLInstanceElement オブジェクトが，この XBRLInstanceTuple オブジェクトの祖先か，またはこの XBRLInstanceTuple オブジェクト自身である場合にスローされます。［エラーコード：20054］
- このタプルがアクティブで，かつ引数 newElement に指定した要素が非アクティブで，かつ引数 newElement に指定した要素（タプルの場合は子孫要素を含む）の id 属性と同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。［エラーコード：20028］
- 引数 oldElement に指定した子要素が，このタプルの子要素でない場合にスローされます。［エラーコード：20055］

（7） removeChildElement

```
public XBRLInstanceElement removeChildElement(XBRLInstanceElement
oldElement)
```

XBRL のバージョン：

2.0 および 2.1

説明：

タプルの子要素を削除します。子要素は，アイテムまたはタプルです。

引数：

oldElement - 削除する子要素を表す XBRLInstanceElement オブジェクトを指定します。

戻り値：

削除された子要素を表す XBRLInstanceElement オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 oldElement に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 oldElement に指定した子要素が，このタプルの子要素でない場合にスローされます。［エラーコード：20055］

6.6.16 XBRLLinkbaseRef インターフェース

```
public interface XBRLLinkbaseRef
extends XBRLInstanceNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

インスタンス文書に記述されている <link:linkbaseRef> 要素を表します。

メソッドの概要：

XBRLLinkbaseRef インターフェースのメソッドの概要を次の表に示します。

表 6-45　XBRLLinkbaseRef インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getRole() <link:linkbaseRef> 要素の xlink:role 属性の値を取得します。	2.0 および 2.1
2	java.lang.String	getHref() <link:linkbaseRef> 要素の xlink:href 属性の値を取得します。	2.0 および 2.1
3	java.lang.String	getBase() <link:linkbaseRef> 要素の xml:base 属性の値を取得します。	2.1

（1） getRole

```
public java.lang.String getRole()
```

XBRL のバージョン：

2.0 および 2.1

説明：

<link:linkbaseRef> 要素の xlink:role 属性の値を取得します。xlink:role 属性は，この <link:linkbaseRef> 要素が参照するリンクベースの種類を識別します。

戻り値：

この <link:linkbaseRef> 要素の xlink:role 属性の値を返します。xlink:role 属性がない場合は null を返します。

（2） getHref

```
public java.lang.String getHref()
```

XBRL のバージョン：

2.0 および 2.1

説明：

<link:linkbaseRef> 要素の xlink:href 属性の値を取得します。xlink:href 属性は，この <link:linkbaseRef> 要素が参照する特定のリンクベースを指します。

戻り値：

この <link:linkbaseRef> 要素の xlink:href 属性の値を返します。

(3) getBase

```
public java.lang.String getBase()
```

XBRL のバージョン：

2.1

説明：

<link:linkbaseRef> 要素の xml:base 属性の値を取得します。

戻り値：

この <link:linkbaseRef> 要素の xml:base 属性の値を返します。xml:base 属性がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。[エラーコード：20100]

6.6.17 XBRLMeasure インターフェース

```
public interface XBRLMeasure
extends XBRLMeasureOrOperator
```

XBRL のバージョン：

2.0

説明：

計量要素を表します。

メソッドの概要：

XBRLMeasure インターフェースのメソッドの概要を次の表に示します。

表 6-46 XBRLMeasure インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLQName	getValue() 計量要素の内容を取得します。	2.0
2	void	setValue(XBRLQName value) 計量要素の内容を設定します。	2.0

(1) getValue

```
public XBRLQName getValue()
```

XBRL のバージョン：

2.0

説明：
計量要素の内容を取得します。

戻り値：
計量を返します。

（2） setValue

```
public void setValue(XBRLQName value)
```

XBRL のバージョン：
2.0

説明：
計量要素の内容を設定します。

引数：
value - 計量を指定します。

例外：
XBRLDOMRuntimeException -

- 引数 value に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 value に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］
- 引数 value に指定した XBRLQName オブジェクトが名前空間に属さず，かつ XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］

6.6.18 XBRLMeasureOrOperator インターフェース

```
public interface XBRLMeasureOrOperator
extends XBRLInstanceNode
```

XBRL のバージョン：
2.0

説明：
計量要素または演算子要素を表します。

フィールドの概要：
XBRLMeasureOrOperator インターフェースのフィールドの概要を次の表に示します。

表 6-47　XBRLMeasureOrOperator インターフェースのフィールドの概要

項番	説明	
1	static int	MEASURE このオブジェクトは，計量要素を表します。
2	static int	OPERATOR このオブジェクトは，演算子要素を表します。

メソッドの概要：

XBRLMeasureOrOperator インターフェースのメソッドの概要を次の表に示します。

表 6-48　XBRLMeasureOrOperator インターフェースのメソッドの概要

説明		XBRL のバージョン
int	getMeasureOrOperatorType() オブジェクトが計量要素または演算子要素のどちらであるかを調べます。	2.0

(1) getMeasureOrOperatorType

```
public int getMeasureOrOperatorType()
```

XBRL のバージョン：

2.0

説明：

オブジェクトが計量要素または演算子要素のどちらであるかを調べます。

戻り値：

計量要素または演算子要素のどちらを表すかを示すコードを返します。

6.6.19　XBRLNonNumericContext インターフェース

```
public interface XBRLNonNumericContext
extends XBRLContext
```

XBRL のバージョン：

2.0

説明：

非数値コンテキストを表します。

メソッドの概要：

XBRLNonNumericContext インターフェースのメソッドの概要を次の表に示します。

表 6-49　XBRLNonNumericContext インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLUnit	getUnit() 非数値コンテキスト要素の単位要素を取得します。	2.0
2	void	setUnit(XBRLUnit unit) 非数値コンテキスト要素の単位要素を設定します。	2.0

（1） getUnit

```
public XBRLUnit getUnit()
```

XBRL のバージョン：

2.0

説明：

非数値コンテキスト要素の単位要素を取得します。

戻り値：

単位要素を表す XBRLUnit オブジェクトを返します。単位要素がない場合は null を返します。

（2） setUnit

```
public void setUnit(XBRLUnit unit)
```

XBRL のバージョン：

2.0

説明：

非数値コンテキスト要素の単位要素を設定します。

この XBRLNonNumericContext オブジェクトに XBRLUnit オブジェクトが設定されている場合，そのオブジェクトは親がいない状態となり，getParentNode() メソッドが null を返します。

引数：

unit - 単位要素を表す XBRLUnit オブジェクトを指定します。単位要素を削除する場合は null を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 unit に指定した XBRLUnit オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLNonNumericContext オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 unit に指定した XBRLUnit オブジェクトが，すでにほかのオブジェクトの子として設定されている場合にスローされます。［エラーコード：20005］

6.6.20 XBRLNumericContext インターフェース

```
public interface XBRLNumericContext
extends XBRLContext
```

XBRL のバージョン：

2.0

説明：

数値コンテキストを表します。

メソッドの概要：

XBRLNumericContext インターフェースのメソッドの概要を次の表に示します。

表 6-50 XBRLNumericContext インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	int	getPrecision() 数値コンテキスト要素の precision 属性の値を取得します。	2.0
2	void	setPrecision(int precision) 数値コンテキスト要素の precision 属性の値を設定します。	2.0
3	boolean	getCwa() 数値コンテキスト要素の cwa 属性の値を取得します。	2.0
4	void	setCwa(boolean cwa) 数値コンテキスト要素の cwa 属性の値を設定します。	2.0
5	XBRLUnit	getUnit() 数値コンテキスト要素の単位要素を取得します。	2.0
6	void	setUnit(XBRLUnit unit) 数値コンテキスト要素の単位要素を設定します。	2.0

(1) getPrecision

```
public int getPrecision()
```

XBRL のバージョン：

2.0

説明：

数値コンテキスト要素の precision 属性の値を取得します。

戻り値：

precision 属性の値を返します。

(2) setPrecision

```
public void setPrecision(int precision)
```

XBRL のバージョン：

2.0

説明：

数値コンテキスト要素の precision 属性の値を設定します。

引数：

precision - precision 属性の値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 precision に 0 以下の値を指定した場合にスローされます。[エラーコード：20014]

(3) getCwa

```
public boolean getCwa()
```

XBRL のバージョン：

2.0

説明：

数値コンテキスト要素の cwa 属性の値を取得します。

戻り値：

cwa 属性の値を返します。

(4) setCwa

```
public void setCwa(boolean cwa)
```

XBRL のバージョン：

2.0

説明：

数値コンテキスト要素の cwa 属性の値を設定します。

引数：

cwa - cwa 属性の値を指定します。

(5) getUnit

```
public XBRLUnit getUnit()
```

XBRL のバージョン：

2.0

説明：

数値コンテキスト要素の単位要素を取得します。

戻り値：

単位要素を表す XBRLUnit オブジェクトを返します。

(6) setUnit

```
public void setUnit(XBRLUnit unit)
```

XBRL のバージョン：

2.0

説明：

数値コンテキスト要素の単位要素を設定します。

この XBRLNumericContext オブジェクトにすでに設定されている XBRLUnit オブジェクトは，親がいない状態となり，getParentNode() メソッドが null を返します。

引数：

unit - 単位要素を表す XBRLUnit オブジェクトを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 unit に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 unit に指定した XBRLUnit オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLNumericContext オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 unit に指定した XBRLUnit オブジェクトが，すでにほかのオブジェクトの子として設定されている場合にスローされます。［エラーコード：20005］

6.6.21 XBRLOperator インターフェース

```
public interface XBRLOperator
extends XBRLMeasureOrOperator
```

XBRL のバージョン：

2.0

説明：

演算子要素を表します。

フィールドの概要：

XBRLOperator インターフェースのフィールドの概要を次の表に示します。

表 6-51　XBRLOperator インターフェースのフィールドの概要

項番	説明	
1	static java.lang.String	MULTIPLY 演算子の種別は乗算です。
2	static java.lang.String	DIVIDE 演算子の種別は除算です。

メソッドの概要：

XBRLOperator インターフェースのメソッドの概要を次の表に示します。

表 6-52　XBRLOperator インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.Stri ng	getKind() 演算子の種別を取得します。	2.0
2	void	setKind(java.lang.String kind) 演算子の種別を設定します。	2.0
3	XBRLMeasur eOrOperator	getLhs() 演算子の左辺を取得します。	2.0
4	void	setLhs(XBRLMeasureOrOperator lhs) 演算子の左辺を設定します。	2.0
5	XBRLMeasur eOrOperator	getRhs() 演算子の右辺を取得します。	2.0
6	void	setRhs(XBRLMeasureOrOperator rhs) 演算子の右辺を設定します。	2.0

（1）getKind

```
public java.lang.String getKind()
```

XBRL のバージョン：

2.0

説明：

演算子の種別を取得します。

戻り値：

乗算の場合は XBRLOperator.MULTIPLY，除算の場合は XBRLOperator.DIVIDE を返します。

（2）setKind

```
public void setKind(java.lang.String kind)
```

XBRL のバージョン：

2.0

説明：
演算子の種別を設定します。

引数：
kind - 乗算の場合は XBRLOperator.MULTIPLY，除算の場合は XBRLOperator.DIVIDE を指定します。

例外：
XBRLDOMRuntimeException -
- 引数 kind に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 kind に指定した値が XBRLOperator.MULTIPLY か XBRLOperator.DIVIDE 以外の場合にスローされます。［エラーコード：20014］

(3) getLhs

```
public XBRLMeasureOrOperator getLhs()
```

XBRL のバージョン：
2.0

説明：
演算子の左辺を取得します。

戻り値：
演算子の左辺を返します。

(4) setLhs

```
public void setLhs(XBRLMeasureOrOperator lhs)
```

XBRL のバージョン：
2.0

説明：
演算子の左辺を設定します。
この XBRLOperator オブジェクトに左辺としてすでに設定されている XBRLMeasureOrOperator オブジェクトは，親がいない状態となり，getParentNode() メソッドが null を返します。

引数：
lhs - 演算子の左辺を指定します。

例外：
XBRLDOMRuntimeException -
- 引数 lhs に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 lhs に指定した XBRLMeasureOrOperator オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLOperator オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされ

ます。［エラーコード：20004］

- 引数 lhs に指定した XBRLMeasureOrOperator オブジェクトが，すでにほかのオブジェクトの子として設定されている場合にスローされます。［エラーコード：20005］
- 引数 lhs に指定した XBRLMeasureOrOperator オブジェクトが，この XBRLOperator オブジェクトの祖先か，またはこの XBRLOperator オブジェクト自身である場合にスローされます。［エラーコード：20054］

（5） getRhs

```
public XBRLMeasureOrOperator getRhs()
```

XBRL のバージョン：

2.0

説明：

演算子の右辺を取得します。

戻り値：

演算子の右辺を返します。

（6） setRhs

```
public void setRhs(XBRLMeasureOrOperator rhs)
```

XBRL のバージョン：

2.0

説明：

演算子の右辺を設定します。
この XBRLOperator オブジェクトに右辺としてすでに設定されている XBRLMeasureOrOperator オブジェクトは，親がいない状態となり，getParentNode() メソッドが null を返します。

引数：

rhs - 演算子の右辺を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 rhs に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 rhs に指定した XBRLMeasureOrOperator オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLOperator オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 rhs に指定した XBRLMeasureOrOperator オブジェクトが，すでにほかのオブジェクトの子として設定されている場合にスローされます。［エラーコード：20005］

- 引数 rhs に指定した XBRLMeasureOrOperator オブジェクトが，この XBRLOperator オブジェクトの祖先か，またはこの XBRLOperator オブジェクト自身である場合にスローされます。［エラーコード：20054］

6.6.22 XBRLPeriod インターフェース

```
public interface XBRLPeriod
extends XBRLInstanceNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

時期要素を表します。

フィールドの概要：

XBRLPeriod インターフェースのフィールドの概要を次の表に示します。

表 6-53 XBRLPeriod インターフェースのフィールドの概要

項番	説明	
1	static int	INSTANT この時期要素の内容の意味は「ある特定の時点」です。
2	static int	FOREVER この時期要素の内容の意味は「永遠」です。
3	static int	STARTDATE_DURATION この時期要素の内容の意味は「開始日，継続期間」です。
4	static int	DURATION_ENDDATE この時期要素の内容の意味は「継続期間，終了日」です。
5	static int	STARTDATE_ENDDATE この時期要素の内容の意味は「開始日，終了日」です。

メソッドの概要：

XBRLPeriod インターフェースのメソッドの概要を次の表に示します。

表 6-54 XBRLPeriod インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	int	getPeriodType() 時期要素の内容の意味を取得します。	2.0 および 2.1
2	XBRLDateUnion	getInstant() 時期要素の内容のある特定の時点を取得します。	2.0 および 2.1
3	XBRLDateUnion	getStartDate() 時期要素の内容の開始日を取得します。	2.0 および 2.1

項番	説明		XBRL のバージョン
4	XBRLDateUnion	getEndDate() 時期要素の内容の終了日を取得します。	2.0 および 2.1
5	XBRLDuration	getDuration() 時期要素の内容の継続期間を取得します。	2.0

（1） getPeriodType

```
public int getPeriodType()
```

XBRL のバージョン：

2.0 および 2.1

説明：

時期要素の内容の意味を取得します。

このメソッドが返す値は，XBRL のバージョンによって異なります。利用できる時期要素の内容の意味と XBRL のバージョンの関係を次の表に示します。

表 6-55　利用できる時期要素の内容の意味と XBRL のバージョンの関係

項番	時期要素の内容の意味	XBRL2.0	XBRL2.1
1	INSTANT	○	○
2	FOREVER	○	○
3	STARTDATE_DURATION	○	×
4	DURATION_ENDDATE	○	×
5	STARTDATE_ENDDATE	○	○

（凡例）○：利用できます。　×：利用できません。

戻り値：

この時期要素の内容の意味を返します。

（2） getInstant

```
public XBRLDateUnion getInstant()
```

XBRL のバージョン：

2.0 および 2.1

説明：

時期要素の内容のある特定の時点を取得します。

このメソッドは，この時期要素の内容の意味が INSTANT の場合に呼び出せます。それ以外の場合に呼び出した場合は，XBRLDOMRuntimeException 例外がスローされます。この時期要素の内容の意味は，getPeriodType() を呼び出して調べることができます。

戻り値：
この時期要素の内容のある特定の時点を返します。

例外：
XBRLDOMRuntimeException -
- この時期要素の内容の意味が INSTANT でない場合にスローされます。［エラーコード：20030］

(3) getStartDate

```
public XBRLDateUnion getStartDate()
```

XBRL のバージョン：
2.0 および 2.1

説明：
時期要素の内容の開始日を取得します。
このメソッドは，この時期要素の内容の意味が STARTDATE_DURATION または STARTDATE_ENDDATE の場合に呼び出せます。それ以外の場合に呼び出した場合は，XBRLDOMRuntimeException 例外がスローされます。この時期要素の内容の意味は，getPeriodType() を呼び出して調べることができます。

戻り値：
この時期要素の内容の開始日を返します。

例外：
XBRLDOMRuntimeException -
- この時期要素の内容の意味が STARTDATE_DURATION または STARTDATE_ENDDATE のどちらでもない場合にスローされます。［エラーコード：20030］

(4) getEndDate

```
public XBRLDateUnion getEndDate()
```

XBRL のバージョン：
2.0 および 2.1

説明：
時期要素の内容の終了日を取得します。
このメソッドは，この時期要素の内容の意味が DURATION_ENDDATE または STARTDATE_ENDDATE の場合に呼び出せます。それ以外の場合に呼び出した場合は，XBRLDOMRuntimeException 例外がスローされます。この時期要素の内容の意味は，getPeriodType() を呼び出して調べることができます。

戻り値：
この時期要素の内容の終了日を返します。

例外：

XBRLDOMRuntimeException -

- この時期要素の内容の意味が DURATION_ENDDATE または STARTDATE_ENDDATE のどちらでもない場合にスローされます。［エラーコード：20030］

(5) getDuration

```
public XBRLDuration getDuration()
```

XBRL のバージョン：

2.0

説明：

時期要素の内容の継続期間を取得します。

このメソッドは，この時期要素の内容の意味が STARTDATE_DURATION または DURATION_ENDDATE の場合に呼び出せます。それ以外の場合に呼び出した場合は，XBRLDOMRuntimeException 例外がスローされます。この時期要素の内容の意味は，getPeriodType() を呼び出して調べることができます。

戻り値：

この時期要素の内容の継続期間を返します。

例外：

XBRLDOMRuntimeException -

- この時期要素の内容の意味が STARTDATE_DURATION または DURATION_ENDDATE のどちらでもない場合にスローされます。［エラーコード：20030］

6.6.23 XBRLRoleRef インターフェース

```
public interface XBRLRoleRef
extends XBRLInstanceNode
```

XBRL のバージョン：

2.1

説明：

インスタンス文書に記述されている <link:roleRef> 要素を表します。

メソッドの概要：

XBRLRoleRef インターフェースのメソッドの概要を次の表に示します。

表 6-56　XBRLRoleRef インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getHref() <link:roleRef> 要素の xlink:href 属性の値を取得します。	2.1
2	java.lang.String	getRoleURI() <link:roleRef> 要素の roleURI 属性の値を取得します。	2.1
3	java.lang.String	getBase() <link:roleRef> 要素の xml:base 属性の値を取得します。	2.1
4	XBRLRoleType	getRoleType() <link:roleRef> 要素の link:roleType 属性の値を取得します。	2.1

(1) getHref

```
public java.lang.String getHref()
```

XBRL のバージョン：

2.1

説明：

<link:roleRef> 要素の xlink:href 属性の値を取得します。xlink:href 属性は，この <link:roleRef> 要素が参照する特定の <link:roleType> 要素を指します。

戻り値：

この <link:roleRef> 要素の xlink:href 属性の値を返します。

(2) getRoleURI

```
public java.lang.String getRoleURI()
```

XBRL のバージョン：

2.1

説明：

<link:roleRef> 要素の roleURI 属性の値を取得します。roleURI 属性の値は，この <link:roleRef> 要素が指す <link:roleType> 要素で定義されるカスタムロールの値と同じです。

戻り値：

この <link:roleRef> 要素の roleURI 属性の値を返します。

(3) getBase

```
public java.lang.String getBase()
```

XBRL のバージョン：

2.1

説明：

<link:roleRef> 要素の xml:base 属性の値を取得します。

戻り値：

この <link:roleRef> 要素の xml:base 属性の値を返します。xml:base 属性がない場合は null を返します。

（4） getRoleType

```
public XBRLRoleType getRoleType()
```

XBRL のバージョン：

2.1

説明：

<link:roleRef> 要素の link:roleType 属性の値を取得します。

このメソッドは，この XBRLRoleRef オブジェクトがアクティブである場合にだけ呼び出せます。この XBRLRoleRef オブジェクトが非アクティブである場合にこのメソッドを呼び出した場合は，XBRLDOMRuntimeException がスローされます。

戻り値：

この <link:roleRef> 要素が指す <link:roleType> 要素を表す XBRLRoleType オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- この XBRLRoleRef オブジェクトが非アクティブである場合にスローされます。［エラーコード：20121］

6.6.24 XBRLScenario インターフェース

```
public interface XBRLScenario
extends XBRLInstanceNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

シナリオ要素を表します。

メソッドの概要：

XBRLScenario インターフェースのメソッドの概要を次の表に示します。

表 6-57　XBRLScenario インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	java.lang.String	getName() シナリオ要素の name 属性を取得します。	2.0	該当しない
2	void	setName(java.lang.String name) シナリオ要素の name 属性を設定します。	2.0	該当しない
3	org.w3c.dom.Node	getContainerNode() シナリオ要素の内容を保持するノードを取得します。	2.0 および 2.1	該当しない
4	java.util.List<XBRLDimensionMember>	getDimensionMembers() シナリオ要素の子要素である <xbrldi:explicitMember 要素 > および <xbrldi:typedMember> 要素のリストを取得します。	2.1	1.0
5	java.util.List<XBRLExplicitMember>	getExplicitMembers() シナリオ要素の子要素である <xbrldi:explicitMember> 要素のリストを取得します。	2.1	1.0
6	java.util.List<XBRLTypedMember>	getTypedMembers() シナリオ要素の子要素である <xbrldi:typedMember> 要素のリストを取得します。	2.1	1.0
7	XBRLDimensionMember	appendDimensionMember(XBRLDimensionMember newMember) シナリオ要素の子要素のリストの末尾に，引数 newMember に指定した要素を追加します。	2.1	1.0
8	XBRLDimensionMember	insertDimensionMemberBefore(XBRLDimensionMember newMember, XBRLDimensionMember refMember) 引数 newMember に指定した要素を，引数 refMember に指定した子要素の前に挿入します。	2.1	1.0
9	XBRLDimensionMember	replaceDimensionMember(XBRLDimensionMember newMember, XBRLDimensionMember oldMember) 引数 oldMember に指定した子要素を，引数 newMember に指定した要素で置き換えます。	2.1	1.0
10	XBRLDimensionMember	removeDimensionMember(XBRLDimensionMember oldMember) 引数 oldMember に指定された子要素を，シナリオ要素の子要素のリストから削除します。	2.1	1.0

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
11	boolean	hasExplicitMember(XBRLQName dimension) シナリオ要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在するかどうかを調べます。	2.1	1.0
12	boolean	hasExplicitMember(XBRLQName dimension, XBRLQName value) シナリオ要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在するかどうかを調べます。	2.1	1.0
13	boolean	hasTypedMember(XBRLQName dimension) シナリオ要素の子要素に，該当する <xbrldi:typedMember> 要素が存在するかどうかを調べます。	2.1	1.0

（1） getName

```
public java.lang.String getName()
```

XBRL のバージョン：

2.0

説明：

シナリオ要素の name 属性を取得します。

戻り値：

name 属性の値を返します。name 属性がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。［エラーコード：20100］

（2） setName

```
public void setName(java.lang.String name)
```

XBRL のバージョン：

2.0

説明：

シナリオ要素の name 属性を設定します。

引数：

name - name 属性の値を指定します。name 属性を削除する場合は null を指定しま

す。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。［エラーコード：20100］

（3） getContainerNode

```
public org.w3c.dom.Node getContainerNode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

シナリオ要素の内容を保持するノードを取得します。
このメソッドが返すノードの子孫ノード以外のノードに対して，更新を伴う操作を行ってはいけません。このメソッドが返すノードの子孫ノードだけを更新できます。

XBRL2.1 の場合：シナリオ要素の内容に関する制約

XBRL の仕様では，シナリオ要素の内容に次の制約があります。

- シナリオ要素の内容には，http://www.xbrl.org/2003/instance の名前空間に属する要素が含まれてはいけません。
- シナリオ要素の内容には，http://www.xbrl.org/2003/instance の名前空間に属する要素の代替要素が含まれてはいけません。
- シナリオ要素の内容が空ではいけません。つまり，シナリオ要素は一つ以上の子要素を持つ必要があります。

XBRL DOM でインスタンス文書を編集する際，上記の制約についてはチェックされません。このため，シナリオ要素の内容を上記の制約に従っていない状態にすることもできます。しかし，シナリオ要素の内容が上記の制約に従っていない場合には，インスタンス文書をストリームへ出力してから再度解析した際にエラーが発生するおそれがあります。
そのため，このメソッドが返した org.w3c.dom.Node オブジェクトを使用してシナリオ要素の内容を編集する場合は，上記の制約に違反しないように注意してください。

XBRL Dimensions1.0 の場合

このメソッドが返した org.w3c.dom.Node オブジェクトを使用して次の操作を実行した場合，XBRLScenario インターフェースの getDimensionMembers メソッド，getExplicitMember メソッドおよび getTypedMember メソッドで取得できるリストに操作内容は反映されません。

- シナリオ要素の内容に，<xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素を追加する
- シナリオ要素の内容である <xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素を編集する

戻り値：

このシナリオ要素の内容を保持する org.w3c.dom.Node オブジェクトを返します。

（4） getDimensionMembers

```
public java.util.List<XBRLDimensionMember> getDimensionMembers()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

シナリオ要素の子要素である <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素のリストを取得します。

このメソッドが返すリスト中での XBRLDimensionMember オブジェクトの順序は，インスタンス文書の要素の出現順と同じです。XBRLDimensionMember オブジェクトは，<xbrldi:explicitMember> 要素の場合は XBRLExplicitMember オブジェクトになり，<xbrldi:typedMember> 要素の場合は XBRLTypedMember オブジェクトになります。

このメソッドが返す XBRLDimensionMember オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLDimensionMember オブジェクトに増減があった場合，その変更は直ちに反映されます。このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このシナリオ要素の子要素である <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素をすべて含む XBRLDimensionMember オブジェクトのリストを返します。このシナリオ要素の子要素に <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素が存在しない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］

（5） getExplicitMembers

```
public java.util.List<XBRLExplicitMember> getExplicitMembers()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

シナリオ要素の子要素である <xbrldi:explicitMember> 要素のリストを取得します。このメソッドが返すリスト中での XBRLExplicitMember オブジェクトの順序は，インスタンス文書における要素の出現順と同じです。

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このシナリオ要素の子要素である <xbrldi:explicitMember> 要素をすべて含む XBRLExplicitMember オブジェクトのリストを返します。このシナリオ要素の子要素に <xbrldi:explicitMember> 要素が存在しない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］

(6) getTypedMembers

```
public java.util.List<XBRLTypedMember> getTypedMembers()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

シナリオ要素の子要素である <xbrldi:typedMember> 要素のリストを取得します。このメソッドが返すリスト中での XBRLTypedMember オブジェクトの順序は，インスタンス文書における要素の出現順と同じです。

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このシナリオ要素の子要素である <xbrldi:typedMember> 要素をすべて含む XBRLTypedMember オブジェクトのリストを返します。このシナリオ要素の子要素に <xbrldi:typedMember> 要素が存在しない場合は，要素数が 0 であるリストを返

します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］

（7） appendDimensionMember

```
public XBRLDimensionMember
appendDimensionMember(XBRLDimensionMember newMember)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

シナリオ要素の子要素である <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素のリストの末尾に，引数 newMember に指定された要素を追加します。

指定された要素がすでにほかのオブジェクトの子要素となっている場合は，最初に既存の親から指定した要素が削除されます。

引数：

newMember - 追加する子要素を表す XBRLDimensionMember オブジェクトを指定します。

戻り値：

追加された子要素を表す XBRLDimensionMember オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 newMember に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newMember に指定した XBRLDimensionMember オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLScenario オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］

(8) insertDimensionMemberBefore

```
public XBRLDimensionMember
insertDimensionMemberBefore(XBRLDimensionMember newMember,
    XBRLDimensionMember refMember )
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

引数 newMember に指定された要素を，引数 refMember に指定された子要素の前に挿入します。

引数 refMember が null の場合は，引数 newMember に指定した要素は，このシナリオ要素の子要素である <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素のリストの末尾に挿入されます。指定された要素がすでにほかのオブジェクトの子要素となっている場合は，最初に既存の親から指定した要素が削除されます。引数 newMember に指定された XBRLDimensionMember オブジェクトが引数 refMember に指定した XBRLDimensionMember オブジェクトと同じ場合は，このメソッドは何もしません。

引数：

newMember - 挿入する子要素を表す XBRLDimensionMember オブジェクトを指定します。

refMember - 参照する子要素を表す XBRLDimensionMember オブジェクトを指定します。引数 newMember で指定した子要素は，この子要素の直前に挿入されます。

戻り値：

挿入された子要素を表す XBRLDimensionMember オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。[エラーコード：20100]
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。[エラーコード：22100]
- 引数 newMember に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 newMember に指定した XBRLDimensionMember オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLScenario オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。[エラーコード：20004]
- 引数 refMember に指定した子要素が，このシナリオ要素の子要素でない場合にス

ローされます。［エラーコード：22104］

（9） replaceDimensionMember

```
public XBRLDimensionMember
replaceDimensionMember(XBRLDimensionMember newMember,
    XBRLDimensionMember oldMember)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

引数 oldMember に指定された子要素を，引数 newMember に指定した要素で置き換えます。

指定した要素がすでにほかのオブジェクトの子要素となっている場合は，最初に既存の親から指定した要素が削除されます。引数 newMember に指定した XBRLDimensionMember オブジェクトが引数 oldMember に指定した XBRLDimensionMember オブジェクトと同じ場合は，このメソッドは何もしません。

引数：

newMember - 挿入する子要素を表す XBRLDimensionMember オブジェクトを指定します。

oldMember - 置き換えの対象となる子要素を表す XBRLDimensionMember オブジェクトを指定します。

戻り値：

置き換えの対象となった子要素を表す XBRLDimensionMember オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 newMember または oldMember に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newMember に指定した XBRLDimensionMember オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLScenario オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 oldMember に指定した子要素が，このシナリオ要素の子要素でない場合にス

ローされます。［エラーコード：22104］

（10）removeDimensionMember

```
public XBRLDimensionMember
removeDimensionMember(XBRLDimensionMember oldMember)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

引数 oldMember に指定された子要素を，このシナリオ要素の子要素のリストから削除します。

引数：

oldMember - 削除する子要素を表す XBRLDimensionMember オブジェクトを指定します。

戻り値：

削除された子要素を表す XBRLDimensionMember オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 oldMember に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 oldMember に指定した子要素が，このシナリオ要素の子要素でない場合にスローされます。［エラーコード：22104］

（11）hasExplicitMember

```
public boolean hasExplicitMember(XBRLQName dimension)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

シナリオ要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在するかどうかを調べます。

引数：

dimension - 検索する <xbrldi:explicitMember> 要素の dimension 属性の値を指定します。

戻り値：

このシナリオ要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在する場合は true，存在しない場合は false を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 dimension に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 dimension に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，この XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］
- 引数 dimension に指定された XBRLQName オブジェクトが名前空間に属していなく，かつ，この XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］

（12）hasExplicitMember

```
public boolean hasExplicitMember(XBRLQName dimension,
                                 XBRLQName value)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

シナリオ要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在するかどうかを調べます。

引数：

dimension - 検索する <xbrldi:explicitMember> 要素の dimension 属性の値を指定します。

value - 検索する <xbrldi:explicitMember> 要素の内容を指定します。

戻り値：

このシナリオ要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在する場合は true，存在しない場合は false を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 dimension または引数 value に，null を指定した場合にスローされます。［エラーコード：20002］
- 引数 dimension または引数 value に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，この XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］
- 引数 dimension または引数 value に指定した XBRLQName オブジェクトが名前空間に属さないで，かつ，この XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］

(13) hasTypedMember

```
public boolean hasTypedMember(XBRLQName dimension)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

シナリオ要素の子要素に，該当する <xbrldi:typedMember> 要素が存在するかどうかを調べます。

引数：

dimension - 検索する <xbrldi:typedMember> 要素の dimension 属性の値を指定します。

戻り値：

このシナリオ要素の子要素に，該当する <xbrldi:typedMember> 要素が存在する場合は true，存在しない場合は false を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 dimension に null を指定した場合にスローされます。［エラーコード：

20002］

- 引数 dimension に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，この XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］
- 引数 dimension に指定した XBRLQName オブジェクトが名前空間に属さないで，かつ，この XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］

6.6.25 XBRLSchemaRef インターフェース

```
public interface XBRLSchemaRef
extends XBRLInstanceNode
```

XBRL のバージョン：

2.1

説明：

<link:schemaRef> 要素を表します。

メソッドの概要：

XBRLSchemaRef インターフェースのメソッドの概要を次の表に示します。

表 6-58 XBRLSchemaRef インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getHref() <link:schemaRef> 要素の xlink:href 属性の値を取得します。	2.1
2	java.lang.String	getBase() <link:schemaRef> 要素の xml:base 属性の値を取得します。	2.1

（1） getHref

```
public java.lang.String getHref()
```

XBRL のバージョン：

2.1

説明：

<link:schemaRef> 要素の xlink:href 属性の値を取得します。xlink:href 属性は，この <link:schemaRef> 要素が参照する特定のタクソノミー本体を指します。

戻り値：

この <link:schemaRef> 要素の xlink:href 属性の値を返します。

(2) getBase

```
public java.lang.String getBase()
```

XBRL のバージョン：

2.1

説明：

<link:schemaRef> 要素の xml:base 属性の値を取得します。

戻り値：

この <link:schemaRef> 要素の xml:base 属性の値を返します。xml:base 属性がない場合は null を返します。

6.6.26 XBRLSegment インターフェース

```
public interface XBRLSegment
extends XBRLInstanceNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

セグメント要素を表します。

メソッドの概要：

XBRLSegment インターフェースのメソッドの概要を次の表に示します。

表 6-59 XBRLSegment インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	java.lang.String	getName() セグメント要素の name 属性を取得します。	2.0	該当しない
2	void	setName(java.lang.String name) セグメント要素の name 属性を設定します。	2.0	該当しない
3	org.w3c.dom.Node	getContainerNode() セグメント要素の内容を保持するノードを取得します。	2.0 および 2.1	該当しない
4	java.util.List<XBRLDimensionMember>	getDimensionMembers() セグメント要素の子要素である <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素のリストを取得します。	2.1	1.0

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
5	java.util.List<XBRLExplicitMember>	getExplicitMembers() セグメント要素の子要素である<xbrldi:explicitMember> 要素のリストを取得します。	2.1	1.0
6	java.util.List<XBRLTypedMember>	getTypedMembers() セグメント要素の子要素である<xbrldi:typedMember> 要素のリストを取得します。	2.1	1.0
7	XBRLDimensionMember	appendDimensionMember(XBRLDimensionMember newMember) セグメント要素の子要素のリストの末尾に，引数 newMember に指定した要素を追加します。	2.1	1.0
8	XBRLDimensionMember	insertDimensionMemberBefore(XBRLDimensionMember newMember, XBRLDimensionMember refMember) 引数 newMember に指定した要素を，引数 refMember に指定した子要素の前に挿入します。	2.1	1.0
9	XBRLDimensionMember	replaceDimensionMember(XBRLDimensionMember newMember, XBRLDimensionMember oldMember) 引数 oldMember に指定した子要素を，引数 newMember に指定した要素で置き換えます。	2.1	1.0
10	XBRLDimensionMember	removeDimensionMember(XBRLDimensionMember oldMember) 引数 oldMember に指定した子要素を，セグメント要素の子要素のリストから削除します。	2.1	1.0
11	boolean	hasExplicitMember(XBRLQName dimension) セグメント要素の子要素に，該当する<xbrldi:explicitMember> 要素が存在するかどうかを調べます。	2.1	1.0
12	boolean	hasExplicitMember(XBRLQName dimension, XBRLQName value) セグメント要素の子要素に，該当する<xbrldi:explicitMember> 要素が存在するかどうかを調べます。	2.1	1.0
13	boolean	hasTypedMember(XBRLQName dimension) セグメント要素の子要素に，該当する<xbrldi:typedMember> 要素が存在するかどうかを調べます。	2.1	1.0

(1) getName

```
public java.lang.String getName()
```

XBRL のバージョン：

2.0

説明：

セグメント要素の name 属性を取得します。

戻り値：

name 属性の値を返します。name 属性がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。[エラーコード：20100]

(2) setName

```
public void setName(java.lang.String name)
```

XBRL のバージョン：

2.0

説明：

セグメント要素の name 属性を設定します。

引数：

name - name 属性の値を指定します。name 属性を削除する場合は null を指定します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。[エラーコード：20100]

(3) getContainerNode

```
public org.w3c.dom.Node getContainerNode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

セグメント要素の内容を保持するノードを取得します。
このメソッドが返すノードの子孫ノード以外のノードに対して，更新を伴う操作を行ってはいけません。このメソッドが返すノードの子孫ノードだけを更新できます。

XBRL2.1 の場合：セグメント要素の内容に関する制約

XBRL の仕様では，セグメント要素の内容に次の制約があります。

- セグメント要素の内容には，http://www.xbrl.org/2003/instance の名前空間に属する要素が含まれてはいけません。
- セグメント要素の内容には，http://www.xbrl.org/2003/instance の名前空間に属する要素の代替要素が含まれてはいけません。
- セグメント要素の内容が空ではいけません。つまり，セグメント要素は一つ以上の子要素を持つ必要があります。

XBRL DOM でインスタンス文書を編集する際，上記の制約についてはチェックされません。このため，セグメント要素の内容を上記の制約に従っていない状態にすることもできます。しかし，セグメント要素の内容が上記の制約に従っていない場合には，インスタンス文書をストリームへ出力してから再度解析した際にエラーが発生するおそれがあります。
そのため，このメソッドが返した org.w3c.dom.Node オブジェクトを使用してセグメント要素の内容を編集する場合は，上記の制約に違反しないように注意してください。
そのため，このメソッドが返した org.w3c.dom.Node オブジェクトを使用してセグメント要素の内容を編集する場合は，上記の制約に違反しないように注意してください。

XBRL Dimensions1.0 の場合

このメソッドが返した org.w3c.dom.Node オブジェクトを使用して次の操作を実行した場合，XBRLSegment インターフェースの getDimensionMembers メソッド，getExplicitMember メソッド，および getTypedMember メソッドで取得できるリストに操作内容は反映されません。

- セグメント要素の内容に <xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素を追加する
- このセグメント要素の内容である <xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素を編集する

戻り値：

このセグメント要素の内容を保持する org.w3c.dom.Node オブジェクトを返します。

（4） getDimensionMembers

```
public java.util.List<XBRLDimensionMember> getDimensionMembers()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

セグメント要素の子要素である <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素のリストを取得します。
このメソッドが返すリスト中での XBRLDimensionMember オブジェクトの順序は，

インスタンス文書の要素の出現順と同じです。XBRLDimensionMember オブジェクトは，<xbrldi:explicitMember> 要素の場合は XBRLExplicitMember オブジェクトになり，<xbrldi:typedMember> 要素の場合は XBRLTypedMember オブジェクトになります。
このメソッドが返す XBRLDimensionMember オブジェクトのリストは，ライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLDimensionMember オブジェクトに増減があった場合，その変更は直ちに反映されます。このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると
java.lang.UnsupportedOperationException がスローされます。

戻り値：
このセグメント要素の子要素である <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素をすべて含む XBRLDimensionMember オブジェクトのリストを返します。このセグメント要素の子要素に <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素が存在しない場合は，要素数が 0 であるリストを返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］

(5) getExplicitMembers

```
public java.util.List<XBRLExplicitMember> getExplicitMembers()
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
セグメント要素の子要素である <xbrldi:explicitMember> 要素のリストを取得します。
このメソッドが返すリスト中での XBRLExplicitMember オブジェクトの順序は，インスタンス文書の要素の出現順と同じです。
このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：
このセグメント要素の子要素である <xbrldi:explicitMember> 要素をすべて含む

XBRLExplicitMember オブジェクトのリストを返します。このセグメント要素の子要素に <xbrldi:explicitMember> 要素が存在しない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］

（6） getTypedMembers

```
public java.util.List<XBRLTypedMember> getTypedMembers()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

セグメント要素の子要素である <xbrldi:typedMember> 要素のリストを取得します。このメソッドが返すリスト中での XBRLTypedMember オブジェクトの順序は，インスタンス文書の要素の出現順と同じです。
このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このセグメント要素の子要素である <xbrldi:typedMember> 要素をすべて含む XBRLTypedMember オブジェクトのリストを返します。このセグメント要素の子要素に <xbrldi:typedMember> 要素が存在しない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］

（7） appendDimensionMember

```
public XBRLDimensionMember
appendDimensionMember(XBRLDimensionMember newMember)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

引数 newMember に指定された要素を，このセグメント要素の子要素である <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素のリストの末尾に追加します。

指定された要素がすでにほかのオブジェクトの子要素となっている場合は，最初に既存の親から指定した要素が削除されます。

引数：

newMember - 追加する子要素を表す XBRLDimensionMember オブジェクトを指定します。

戻り値：

追加された子要素を表す XBRLDimensionMember オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 newMember に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newMember に指定した XBRLDimensionMember オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLSegment オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］

（8） insertDimensionMemberBefore

```
public XBRLDimensionMember
insertDimensionMemberBefore(XBRLDimensionMember newMember,
    XBRLDimensionMember refMember)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

引数 newMember に指定された要素を，引数 refMember に指定された子要素の前に挿入します。

引数 refMember が null の場合は，引数 newMember に指定された要素は，このセ

グメント要素の子要素である <xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素のリストの末尾に挿入されます。指定した要素がすでにほかのオブジェクトの子要素となっている場合は，最初に既存の親から指定した要素が削除されます。引数 newMember に指定した XBRLDimensionMember オブジェクトが引数 refMember に指定した XBRLDimensionMember オブジェクトと同じ場合は，このメソッドは何もしません。

引数：

newMember - 挿入する子要素を表す XBRLDimensionMember オブジェクトを指定します。

refMember - 参照する子要素を表す XBRLDimensionMember オブジェクトを指定します。引数 newMember で指定された子要素は，この子要素の直前に挿入されます。

戻り値：

挿入された子要素を表す XBRLDimensionMember オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 newMember に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newMember に指定した XBRLDimensionMember オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLSegment オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 refMember に指定した子要素が，このセグメント要素の子要素でない場合にスローされます。［エラーコード：22105］

（9） replaceDimensionMember

```
public XBRLDimensionMember
replaceDimensionMember(XBRLDimensionMember newMember,
    XBRLDimensionMember oldMember)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

引数 oldMember に指定された子要素を，引数 newMember に指定された要素で置

き換えます。
指定した要素がすでにほかのオブジェクトの子要素となっている場合は，最初に既存の親から指定した要素が削除されます。引数 newMember に指定した XBRLDimensionMember オブジェクトが引数 oldMember に指定した XBRLDimensionMember オブジェクトと同じ場合は，このメソッドは何もしません。

引数：
newMember - 挿入する子要素を表す XBRLDimensionMember オブジェクトを指定します。
oldMember - 置き換えの対象となる子要素を表す XBRLDimensionMember オブジェクトを指定します。

戻り値：
置き換えの対象となった子要素を表す XBRLDimensionMember オブジェクトを返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 newMember または oldMember に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 newMember に指定した XBRLDimensionMember オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLSegment オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 oldMember に指定した子要素が，このセグメント要素の子要素でない場合にスローされます。［エラーコード：22105］

（10）removeDimensionMember

```
public XBRLDimensionMember
removeDimensionMember(XBRLDimensionMember oldMember)
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
引数 oldMember に指定された子要素を，このセグメント要素の子要素のリストから削除します。

引数：

oldMember - 削除する子要素を表す XBRLDimensionMember オブジェクトを指定します。

戻り値：

削除された子要素を表す XBRLDimensionMember オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 oldMember に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 oldMember に指定された子要素が，このセグメント要素の子要素でない場合にスローされます。［エラーコード：22105］

（11）hasExplicitMember

```
public boolean hasExplicitMember(XBRLQName dimension)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

セグメント要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在するかどうかを調べます。

引数：

dimension - 検索する <xbrldi:explicitMember> 要素の dimension 属性の値を指定します。

戻り値：

このセグメント要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在する場合は true，存在しない場合は false を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］

- 引数 dimension に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 dimension に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，この XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］
- 引数 dimension に指定した XBRLQName オブジェクトが名前空間に属さないで，かつ，この XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］

（12）hasExplicitMember

```
public boolean hasExplicitMember(XBRLQName dimension,
    XBRLQName value)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

セグメント要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在するかどうかを調べます。

引数：

dimension - 検索する <xbrldi:explicitMember> 要素の dimension 属性の値を指定します。

value - 検索する <xbrldi:explicitMember> 要素の内容を指定します。

戻り値：

このセグメント要素の子要素に，該当する <xbrldi:explicitMember> 要素が存在する場合は true，存在しない場合は false を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 dimension または引数 value に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 dimension または引数 value に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，この XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］
- 引数 dimension または引数 value に指定した XBRLQName オブジェクトが名前

空間に属さないで，かつ，この XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］

（13）hasTypedMember

```
public boolean hasTypedMember(XBRLQName dimension)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

セグメント要素の子要素に，該当する <xbrldi:typedMember> 要素が存在するかどうかを調べます。

引数：

dimension - 検索する <xbrldi:typedMember> 要素の dimension 属性の値を指定します。

戻り値：

このセグメント要素の子要素に，該当する <xbrldi:typedMember> 要素が存在する場合は true，存在しない場合は false を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のインスタンス文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 操作対象のインスタンス文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］
- 引数 dimension に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 dimension に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，この XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］
- 引数 dimension に指定した XBRLQName オブジェクトが名前空間に属さないで，かつ，この XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］

6.6.27 XBRLUnit インターフェース

```
public interface XBRLUnit
extends XBRLInstanceNode
```

XBRL のバージョン：
2.0

説明：
単位要素を表します。

メソッドの概要：
XBRLUnit インターフェースのメソッドの概要を次の表に示します。

表 6-60　XBRLUnit インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLMeasureOrOperator	getMeasureOrOperator() 計量要素または演算子要素を取得します。	2.0
2	void	setMeasureOrOperator(XBRLMeasureOrOperator measureOrOperator) 計量要素または演算子要素を設定します。	2.0

(1) getMeasureOrOperator

```
public XBRLMeasureOrOperator getMeasureOrOperator()
```

XBRL のバージョン：
2.0

説明：
計量要素または演算子要素を取得します。

戻り値：
計量要素または演算子要素を表す XBRLMeasureOrOperator オブジェクトを返します。このメソッドが返したオブジェクトは，XBRLMeasure または XBRLOperator のどちらか一方にキャストできます。XBRLMeasure と XBRLOperator のどちらにキャストできるかは，XBRLMeasureOrOperator の getMeasureOrOperatorType() を呼び出すことで判定できます。

(2) setMeasureOrOperator

```
public void setMeasureOrOperator(XBRLMeasureOrOperator
measureOrOperator)
```

XBRL のバージョン：
2.0

説明：
計量要素または演算子要素を設定します。
この XBRLUnit オブジェクトにすでに設定されている XBRLMeasureOrOperator オブジェクトは，親がいない状態となり，getParentNode() メソッドが null を返します。

引数：

measureOrOperator - 計量要素または演算子要素を表す XBRLMeasureOrOperator オブジェクトを指定します。この引数に指定するオブジェクトは，XBRLMeasure または XBRLOperator のどちらかでなければなりません。

例外：

XBRLDOMRuntimeException -

- 引数 measureOrOperator に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 measureOrOperator に指定した XBRLMeasureOrOperator オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLUnit オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なる場合にスローされます。［エラーコード：20004］
- 引数 measureOrOperator に指定した XBRLMeasureOrOperator オブジェクトが，すでにほかのオブジェクトの子として設定されている場合にスローされます。［エラーコード：20005］

6.6.28 XBRLUnit2 インターフェース

```
public interface XBRLUnit2
extends XBRLInstanceNode
```

XBRL のバージョン：

2.1

説明：

単位要素を表します。

単位要素の id 属性の値は，単位要素を生成する際に自動的に採番されます。id 属性の値は，単位要素を生成する際に自動的に採番されますが，setId(java.lang.String) を使用して任意の値を設定することもできます。自動採番の詳細については，「3.3 ID 管理と採番」を参照してください。

フィールドの概要：

XBRLUnit2 インターフェースのフィールドの概要を次の表に示します。

表 6-61 XBRLUnit2 インターフェースのフィールドの概要

項番	説明	
1	static int	MULTIPLY 単位要素の内容は，一つ以上の基本単位によって構成されることを示します。
2	static int	DIVIDE 単位要素の内容は，分数であることを示します。分数の分子と分母は，一つ以上の基本単位によって構成されます。

メソッドの概要：

XBRLUnit2 インターフェースのメソッドの概要を次の表に示します。

表 6-62　XBRLUnit2 インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getId() 単位要素の id 属性の値を取得します。	2.1
2	void	setId(java.lang.String id) 単位要素の id 属性の値を設定します。	2.1
3	int	getUnitType() 単位要素の内容を表すコードを取得します。	2.1
4	java.util.List<XBRLQName>	getMeasures() 単位要素を構成する基本単位のリストを取得します。	2.1
5	java.util.List<XBRLQName>	getNumerator() 単位要素の分子を構成する基本単位のリストを取得します。	2.1
6	java.util.List<XBRLQName>	getDenominator() 単位要素の分母を構成する基本単位のリストを取得します。	2.1
7	java.util.List<XBRLInstanceItem>	getItems() 単位要素を参照しているアイテムのリストを取得します。	2.1
8	java.util.List<XBRLInstanceItem>	getItems(java.lang.String namespaceURI, java.lang.String name) 単位要素を参照している，指定した名前のアイテムのリストを取得します。	2.1

(1) getId

```
public java.lang.String getId()
```

XBRL のバージョン：

2.1

説明：

単位要素の id 属性の値を取得します。

戻り値：

この単位要素の id 属性の値を返します。

(2) setId

```
public void setId(java.lang.String id)
```

XBRL のバージョン：

2.1

説明：

単位要素の id 属性の値を設定します。

引数：

id - id 属性の値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 id に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 id に指定した値が xsd:NCName 型でない場合にスローされます。［エラーコード：20108］
- この単位要素がインスタンス文書中に存在していて，かつ，引数 id で指定されたのと同じ値を持つ xsd:ID 型属性がインスタンス文書中にすでに存在する場合にスローされます。［エラーコード：20028］

（3） getUnitType

```
public int getUnitType()
```

XBRL のバージョン：

2.1

説明：

単位要素の内容を表すコードを取得します。

戻り値：

この単位要素の内容を表すコードを返します。戻り値のコードは，XBRLUnit2 インターフェースのフィールド値のどちらかです。

（4） getMeasures

```
public java.util.List<XBRLQName> getMeasures()
```

XBRL のバージョン：

2.1

説明：

単位要素を構成する基本単位のリストを取得します。

このメソッドは，この単位要素の内容が XBRLUnit2.MULTIPLY の場合に呼び出せます。それ以外の場合に呼び出した場合は，XBRLDOMRuntimeException 例外がスローされます。この単位要素の内容は，getUnitType() メソッドを呼び出して調べることができます。

このメソッドが返すリスト中での XBRLQName オブジェクトの順序は，基本単位を表す <xbrli:measure> 要素のインスタンス文書中での出現順と同じです。

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

この単位要素を構成する基本単位を表す XBRLQName オブジェクトのリストを返します。

例外：

XBRLDOMRuntimeException -

- この単位要素の内容の意味が XBRLUnit2.MULTIPLY でない場合にスローされます。［エラーコード：20128］

(5) getNumerator

```
public java.util.List<XBRLQName> getNumerator()
```

XBRL のバージョン：

2.1

説明：

単位要素の分子を構成する基本単位のリストを取得します。
このメソッドは，この単位要素の内容が XBRLUnit2.DIVIDE の場合に呼び出せます。それ以外の場合に呼び出した場合は，XBRLDOMRuntimeException 例外がスローされます。この単位要素の内容は，getUnitType() メソッドを呼び出して調べることができます。
このメソッドが返すリスト中での XBRLQName オブジェクトの順序は，分子を構成する基本単位を表す <xbrli:measure> 要素のインスタンス文書中での出現順と同じです。
このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

この単位要素の分子を構成する基本単位を表す XBRLQName オブジェクトのリストを返します。

例外：

XBRLDOMRuntimeException -

- この単位要素の内容の意味が XBRLUnit2.DIVIDE でない場合にスローされます。［エラーコード：20128］

(6) getDenominator

```
public java.util.List<XBRLQName> getDenominator()
```

XBRL のバージョン：

2.1

説明：

単位要素の分母を構成する基本単位のリストを取得します。
このメソッドは，この単位要素の内容が XBRLUnit2.DIVIDE の場合に呼び出せます。それ以外の場合に呼び出した場合は，XBRLDOMRuntimeException 例外がス

ローされます。この単位要素の内容は，getUnitType() メソッドを呼び出して調べることができます。
このメソッドが返すリスト中での XBRLQName オブジェクトの順序は，分母を構成する基本単位を表す <xbrli:measure> 要素のインスタンス文書中での出現順と同じです。
このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：
この単位要素の分母を構成する基本単位を表す XBRLQName オブジェクトのリストを返します。

例外：
XBRLDOMRuntimeException -
- この単位要素の内容の意味が XBRLUnit2.DIVIDE でない場合にスローされます。［エラーコード：20128］

（7） getItems

```
public java.util.List<XBRLInstanceItem> getItems()
```

XBRL のバージョン：
2.1

説明：
単位要素を参照しているアイテムのリストを取得します。
この単位要素を参照しているアイテムには，インスタンス文書に存在しないもの（XBRLInstanceNode.isActive() が false を返すノード）も含まれます。
このメソッドが返すアイテムのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれるアイテムに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：
この単位要素を参照しているアイテムをすべて含む XBRLInstanceItem オブジェクトのリストを返します。この単位要素を参照しているアイテムが存在しない場合は，要素数が 0 のリストを返します。

（8） getItems

```
public java.util.List<XBRLInstanceItem> getItems(java.lang.String
namespaceURI,
                                  java.lang.String name)
```

XBRL のバージョン：
2.1

説明：

単位要素を参照している，指定した名前のアイテムのリストを取得します。
この単位要素を参照しているアイテムには，インスタンス文書に存在しないもの（XBRLInstanceNode.isActive() が false を返すノード）も含まれます。
このメソッドが返すアイテムのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれるアイテムに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

namespaceURI - アイテムの名前空間 URI を指定します。
name - アイテムの名前を指定します。

戻り値：

引数 namespaceURI で指定された名前空間 URI に属し，引数 name で指定された名前を持つアイテムを表す XBRLInstanceItem オブジェクトのリストを返します。指定されたアイテムが存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 namespaceURI または name に null を指定した場合にスローされます。［エラーコード：20002］

6.6.29 XBRLDimensionMember インターフェース

```
public interface XBRLDimensionMember
extends XBRLInstanceNode
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<xbrldi:explicitMember> 要素および <xbrldi:typedMember> 要素を表します。
この要素が <xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素のどちらであるかは，getNodeType() メソッドで調べることができます。getNodeType() メソッドが EXPLICIT_MEMBER_NODE フィールドを返した場合は，XBRLExplicitMember インターフェースにキャストできます。getNodeType() メソッドが TYPED_MEMBER_NODE フィールドを返した場合は，XBRLTypedMember インターフェースにキャストできます。

メソッドの概要：

XBRLDimensionMember インターフェースのメソッドの概要を次の表に示します。

表 6-63　XBRLDimensionMember インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLQName	getDimension() <xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素の dimension 属性の値を取得します。	2.1	1.0
2	void	setDimension(XBRLQName dimension) <xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素の dimension 属性の値を設定します。	2.1	1.0

（1） getDimension

```
public XBRLQName getDimension()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素の dimension 属性の値を取得します。

戻り値：

<xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素の dimension 属性の値を返します。

（2） setDimension

```
public void setDimension(XBRLQName dimension)
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素の dimension 属性の値を設定します。

引数：

dimension - <xbrldi:explicitMember> 要素または <xbrldi:typedMember> 要素の

dimension 属性の値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 dimension に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 dimension に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，この XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。［エラーコード：20023］
- 引数 dimension に指定した XBRLQName オブジェクトが名前空間に属さないで，かつ，この XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。［エラーコード：20069］
- 引数 dimension で指定したディメンジョンがタクソノミー文書で定義されていない場合にスローされます。［エラーコード：22106］

6.6.30 XBRLExplicitMember インターフェース

```
public interface XBRLExplicitMember
extends XBRLDimensionMember
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<xbrldi:explicitMember> 要素を表します。

メソッドの概要：

XBRLExplicitMember インターフェースのメソッドの概要を次の表に示します。

表 6-64 XBRLExplicitMember インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLQName	getValue() <xbrldi:explicitMember> 要素の内容を取得します。	2.1	1.0
2	void	setValue(XBRLQName value) <xbrldi:explicitMember> 要素の内容を設定します。	2.1	1.0

(1) getValue

```
public XBRLQName getValue()
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
<xbrldi:explicitMember> 要素の内容を取得します。

戻り値：
<xbrldi:explicitMember> 要素の内容を返します。

(2) setValue

```
public void setValue(XBRLQName value)
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
<xbrldi:explicitMember> 要素の内容を設定します。

引数：
value - <xbrldi:explicitMember> 要素の内容を指定します。

例外：
XBRLDOMRuntimeException -

- 引数 value に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 value に指定した XBRLQName オブジェクトの名前空間 URI に対応する名前空間宣言が，この XBRLInstanceDocument オブジェクトに登録されていない場合にスローされます。[エラーコード：20023]
- 引数 value に指定した XBRLQName オブジェクトが名前空間に属さないで，かつ，この XBRLInstanceDocument オブジェクトにデフォルトの名前空間の宣言が登録されている場合にスローされます。[エラーコード：20069]
- 引数 value で指定したアイテムがタクソノミー文書で定義されていない場合にスローされます。[エラーコード：20044]

6.6.31 XBRLTypedMember インターフェース

```
public interface XBRLTypedMember
extends XBRLDimensionMember
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：

1.0

説明：

<xbrldi:typedMember> 要素を表します。

メソッドの概要：

XBRLTypedMember インターフェースのメソッドの概要を次の表に示します。

表 6-65　XBRLTypedMember インターフェースのメソッドの概要

説明		XBRL のバージョン	XBRL Dimensions のバージョン
org.w3c.dom.Node	getContainerNode() <xbrldi:typedMember> 要素の内容を保持するノードを取得します。	2.1	1.0

（1） getContainerNode

```
public org.w3c.dom.Node getContainerNode()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<xbrldi:typedMember> 要素の内容を保持するノードを取得します。

このメソッドが返すノードの子孫ノード以外のノードに対して，更新を伴う操作をしないでください。このメソッドが返すノードの子孫ノードだけを更新できます。

戻り値：

この <xbrldi:typedMember> 要素の内容を保持する org.w3c.dom.Node オブジェクトを返します。

6.7 com.hitachi.xbrl.link パッケージ

com.hitachi.xbrl.link パッケージは，XBRL 文書に記述されているリンク情報を参照するためのインターフェースを提供します。

インターフェースの概要：

com.hitachi.xbrl.link パッケージのインターフェースを次の表に示します。

表 6-66 com.hitachi.xbrl.link パッケージのインターフェース

項番	インターフェース	説明	XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLCalculationRelationship	<link:calculationArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言と計算リンクとの関係を表します。	2.1	該当しない
2	XBRLDefinitionRelationship	<link:definitionArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言と定義リンクとの関係を表します。	2.1	該当しない
3	XBRLFootnoteRelationship	<link:footnoteArc> 要素で定義された，インスタンス文書のアイテム（またはタプル）と脚注との関係を表します。	2.1	該当しない
4	XBRLLabelRelationship	<link:labelArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言とラベルとの関係を表します。	2.1	該当しない
5	XBRLPresentationRelationship	<link:presentationArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言とプレゼンテーションリンクとの関係を表します。	2.1	該当しない
6	XBRLReferenceRelationship	<link:referenceArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言とリファレンスリンクとの関係を表します。	2.1	該当しない
7	XBRLRelationship	アークで定義された関係を表します。	2.1	該当しない

項番	インターフェース	説明	XBRL のバージョン	XBRL Dimensions のバージョン
8	XBRLDimensionalRelationship	<link:definitionArc> 要素によって定義された，DRS を構成するアイテム同士の関係を表します。	2.1	1.0
9	XBRLDimensionalRelationshipSet	プライマリー・アイテムに関連づいている DRS を表します。	2.1	1.0
10	XBRLDimensionDefaultRelationship	<link:definitionArc> 要素によって定義された，dimension-default 関係を表します。	2.1	1.0
11	XBRLDimensionDomainRelationship	<link:definitionArc> 要素によって定義された，dimension-domain 関係を表します。	2.1	1.0
12	XBRLDomainMemberRelationship	<link:definitionArc> 要素によって定義された，domain-member 関係を表します。	2.1	1.0
13	XBRLDRSDimension	インスタンス文書に記述できるディメンジョンを表します。	2.1	1.0
14	XBRLDRSMember	インスタンス文書に記述できるディメンジョンのメンバを表します。	2.1	1.0
15	XBRLHasHypercubeRelationship	<link:definitionArc> 要素によって定義された，has-hypercube 関係を表します。	2.1	1.0
16	XBRLHypercubeDimensionRelationship	<link:definitionArc> 要素によって定義された，hypercube-dimension 関係を表します。	2.1	1.0

例外の概要：

com.hitachi.xbrl.link パッケージの例外の概要を次の表に示します。

表 6-67　com.hitachi.xbrl.link パッケージの例外の概要

項番	クラス	説明	XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLIllegalCycleException	関係ネットワークに対するサイクル検証で検出されたエラーを通知するために，uCosminexus Business Reporting Processor によって生成されます。	2.1	該当しない
2	XBRLDimensionIllegalCycleException	DRS の関係ネットワークに対するサイクル検証で検出されたエラーを通知するために，uCosminexus Business Reporting Processor によって生成されます。	2.1	1.0

6.7.1　XBRLCalculationRelationship インターフェース

```
public interface XBRLCalculationRelationship
extends XBRLRelationship
```

XBRL のバージョン：

2.1

説明：

<link:calculationArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言と計算リンクとの関係を表します。

メソッドの概要：

XBRLCalculationRelationship インターフェースのメソッドの概要を次の表に示します。

表 6-68　XBRLCalculationRelationship インターフェースのメソッドの概要

説明		XBRL のバージョン
java.math.BigDecimal	getWeight() <link:calculationArc> 要素の weight 属性の値を取得します。	2.1

（1）getWeight

```
public java.math.BigDecimal getWeight()
```

XBRL のバージョン：

2.1

説明：

<link:calculationArc> 要素の weight 属性の値を取得します。

戻り値：

<link:calculationArc> 要素の weight 属性の値を返します。

6.7.2 XBRLDefinitionRelationship インターフェース

```
public interface XBRLDefinitionRelationship
extends XBRLRelationship
```

XBRL のバージョン：

2.1

説明：

<link:definitionArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言と定義リンクとの関係を表します。

6.7.3 XBRLFootnoteRelationship インターフェース

```
public interface XBRLFootnoteRelationship
extends XBRLRelationship
```

XBRL のバージョン：

2.1

説明：

<link:footnoteArc> 要素で定義された，インスタンス文書のアイテム（またはタプル）と脚注との関係を表します。

6.7.4 XBRLLabelRelationship インターフェース

```
public interface XBRLLabelRelationship
extends XBRLRelationship
```

XBRL のバージョン：

2.1

説明：

<link:labelArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言とラベルとの関係を表します。

6.7.5 XBRLPresentationRelationship インターフェース

```
public interface XBRLPresentationRelationship
extends XBRLRelationship
```

XBRL のバージョン：

2.1

説明：

<link:presentationArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言とプレゼンテーションリンクとの関係関係を表します。

メソッドの概要：

XBRLPresentationRelationship インターフェースのメソッドの概要を次の表に示します。

表 6-69　XBRLPresentationRelationship インターフェースのメソッドの概要

説明		XBRL のバージョン
java.lang.String	getPreferredLabel() <link:presentationArc> 要素の preferredLabel 属性の値を取得します。	2.1

（1） getPreferredLabel

```
public java.lang.String getPreferredLabel()
```

XBRL のバージョン：

2.1

説明：

<link:presentationArc> 要素の preferredLabel 属性の値を取得します。

戻り値：

<link:presentationArc> 要素の preferredLabel 属性の値を返します。
preferredLabel 属性がない場合は null を返します。

6.7.6　XBRLReferenceRelationship インターフェース

```
public interface XBRLReferenceRelationship
extends XBRLRelationship
```

XBRL のバージョン：

2.1

説明：

<link:referenceArc> 要素で定義された，タクソノミー文書のアイテム（またはタプル）の要素宣言とリファレンスリンク（文献への参照）との関係を表します。

6.7.7　XBRLRelationship インターフェース

```
public interface XBRLRelationship
```

XBRL のバージョン：

2.1

説明：

アークで定義された関係を表します。

フィールドの概要：

XBRLRelationship インターフェースのフィールドの概要を次の表に示します。

表 6-70　XBRLRelationship インターフェースのフィールドの概要

項番	説明	
1	static int	LABEL_ARC 関係が <link:labelArc> 要素によって定義されたことを示します。
2	static int	REFERENCE_ARC 関係が <link:referenceArc> 要素によって定義されたことを示します。
3	static int	CALCULATION_ARC 関係が <link:calculationArc> 要素によって定義されたことを示します。
4	static int	DEFINITION_ARC 関係が <link:definitionArc> 要素によって定義されたことを示します。
5	static int	PRESENTATION_ARC 関係が <link:presentationArc> 要素によって定義されたことを示します。
6	static int	FOOTNOTE_ARC 関係が <link:footnoteArc> 要素によって定義されたことを示します。
7	static int	HAS_HYPERCUBE_ARC 関係が <link:definitionArc> 要素によって定義された has-hypercube 関係であることを示します。
8	static int	HYPERCUBE_DIMENSION_ARC 関係が <link:definitionArc> 要素によって定義された hypercube-dimension 関係であることを示します。
9	static int	DIMENSION_DOMAIN_ARC 関係が <link:definitionArc> 要素によって定義された dimension-domain 関係であることを示します。
10	static int	DOMAIN_MEMBER_ARC 関係が <link:definitionArc> 要素によって定義された domain-member 関係であることを示します。
11	static int	DIMENSION_DEFAULT_ARC 関係が <link:definitionArc> 要素によって定義された dimension-default 関係であることを示します。

メソッドの概要：

XBRLRelationship インターフェースのメソッドの概要を次の表に示します。

表 6-71　XBRLRelationship インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	int	getArcType() 関係を定義したアークの種類を調べます。	2.1
2	java.lang.String	getExtendedLinkRole() 関係を定義したアークを含む拡張リンクのロールを取得します。	2.1
3	java.lang.String	getArcrole() 関係を定義したアークのアークロールを取得します。	2.1
4	java.math.BigDecimal	getOrder() 関係を定義したアークの order 属性の値を取得します。	2.1
5	java.lang.Object	getFrom() 関係の from 側のリソースを取得します。	2.1
6	java.lang.Object	getTo() 関係の to 側のリソースを取得します。	2.1
7	java.math.BigInteger	getPriority() 関係を定義したアークの priority 属性の値を取得します。	2.1
8	boolean	isOrderAttributeSpecified() 関係を定義したアークの order 属性が定義されているかを判定します。	2.1

（1） getArcType

```
public int getArcType()
```

XBRL のバージョン：

2.1

説明：

関係を定義したアークの種類を調べます。

この関係を定義したアークの種類に応じて，この XBRLRelationship オブジェクトを，各種のアークに特化したインターフェースへキャストできます。アークの種類とインターフェースの関係を次の表に示します。

表 6-72　アークの種類とインターフェースの関係

項番	関係を定義したアークの種類 （getArcType() メソッドが返すコード）	対応するインターフェース
1	LABEL_ARC	XBRLLabelRelationship
2	REFERENCE_ARC	XBRLReferenceRelationship
3	CALCULATION_ARC	XBRLCalculationRelationship
4	DEFINITION_ARC	XBRLDefinitionRelationship

項番	関係を定義したアークの種類 （getArcType() メソッドが返すコード）	対応するインターフェース
5	PRESENTATION_ARC	XBRLPresentationRelationship
6	FOOTNOTE_ARC	XBRLFootnoteRelationship
7	HAS_HYPERCUBE_ARC ※	XBRLHasHypercubeRelationship
8	HYPERCUBE_DIMENSION_ARC ※	XBRLHypercubeDimensionRelationship
9	DIMENSION_DOMAIN_ARC ※	XBRLDimensionDomainRelationship
10	DOMAIN_MEMBER_ARC ※	XBRLDomainMemberRelationship
11	DIMENSION_DEFAULT_ARC ※	XBRLDimensionDefaultRelationship

注※　XBRL Dimensions1.0 の場合だけ存在します。

戻り値：
　この関係を定義したアークの種類を返します。

（2） getExtendedLinkRole

```
public java.lang.String getExtendedLinkRole()
```

XBRL のバージョン：
　2.1

説明：
　関係を定義したアークを含む拡張リンクのロールを取得します。

戻り値：
　この関係を定義したアークを含む拡張リンクのロールを返します。

（3） getArcrole

```
public java.lang.String getArcrole()
```

XBRL のバージョン：
　2.1

説明：
　関係を定義したアークのアークロールを取得します。

戻り値：
　この関係を定義したアークのアークロールを返します。

（4） getOrder

```
public java.math.BigDecimal getOrder()
```

XBRL のバージョン：

2.1

説明：

関係を定義したアークの order 属性の値を取得します。

戻り値：

この関係を定義したアークの order 属性の値を返します。order 属性がない場合は 1 を返します。

(5) getFrom

```
public java.lang.Object getFrom()
```

XBRL のバージョン：

2.1

説明：

関係の from 側のリソースを取得します。

このメソッドの戻り値は，関係を定義したアークの種類によって異なります。アークの種類と from 側のリソースの関係を，次の表に示します。

表 6-73　アークの種類と from 側のリソースの関係

項番	関係を定義したアークの種類 (getArcType() メソッドが返すコード)	from 側のリソース (getFrom() メソッドが返すオブジェクト)
1	LABEL_ARC	XBRLTaxonomyElement
2	REFERENCE_ARC	XBRLTaxonomyElement
3	CALCULATION_ARC	XBRLTaxonomyElement
4	DEFINITION_ARC	XBRLTaxonomyElement
5	PRESENTATION_ARC	XBRLTaxonomyElement
6	FOOTNOTE_ARC	XBRLInstanceElement
7	HAS_HYPERCUBE_ARC ※	XBRLTaxonomyItem
8	HYPERCUBE_DIMENSION_ARC ※	XBRLTaxonomyHypercubeItem
9	DIMENSION_DOMAIN_ARC ※	XBRLTaxonomyDimensionItem
10	DOMAIN_MEMBER_ARC ※	XBRLTaxonomyItem
11	DIMENSION_DEFAULT_ARC ※	XBRLTaxonomyDimensionItem

注※　XBRL Dimensions1.0 の場合だけ存在します。

戻り値：

この関係の from 側のリソースを表すオブジェクトを返します。

(6) getTo

```
public java.lang.Object getTo()
```

XBRL のバージョン：

2.1

説明：

関係の to 側のリソースを取得します。

このメソッドの戻り値は，関係を定義したアークの種類によって異なります。アークの種類と to 側のリソースの関係を次の表に示します。

表 6-74　アークの種類と to 側のリソースの関係

項番	関係を定義したアークの種類 （getArcType() メソッドが返すコード）	to 側のリソース （getTo() メソッドが返すオブジェクト）
1	LABEL_ARC	XBRLLabel
2	REFERENCE_ARC	XBRLReference
3	CALCULATION_ARC	XBRLTaxonomyElement
4	DEFINITION_ARC	XBRLTaxonomyElement
5	PRESENTATION_ARC	XBRLTaxonomyElement
6	FOOTNOTE_ARC	XBRLFootnote
7	HAS_HYPERCUBE_ARC ※	XBRLTaxonomyHypercubeItem
8	HYPERCUBE_DIMENSION_ARC ※	XBRLTaxonomyDimensionItem
9	DIMENSION_DOMAIN_ARC ※	XBRLTaxonomyItem
10	DOMAIN_MEMBER_ARC ※	XBRLTaxonomyItem
11	DIMENSION_DEFAULT_ARC ※	XBRLTaxonomyItem

注※　XBRL Dimensions1.0 の場合だけ存在します。

戻り値：

この関係の to 側のリソースを表すオブジェクトを返します。

（7） getPriority

```
public java.math.BigInteger getPriority()
```

XBRL のバージョン：

2.1

説明：

関係を定義したアークの priority 属性の値を取得します。

戻り値：

この関係を定義したアークの priority 属性の値を返します。priority 属性がない場合は 0（デフォルト値）を返します。

(8) isOrderAttributeSpecified

```
public boolean isOrderAttributeSpecified()
```

XBRL のバージョン：

2.1

説明：

関係を定義したアークの order 属性が定義されているかを判定します。

戻り値：

この関係を定義したアークの order 属性が定義されている場合は true，order 属性が定義されていない場合は false を返します。XBRLInstanceElement の addFootnote() で生成される XBRLRelationship オブジェクトは常に true を返します。

6.7.8 XBRLDimensionalRelationship インターフェース

```
public interface XBRLDimensionalRelationship
extends XBRLDefinitionRelationship
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<link:definitionArc> 要素によって定義された，DRS を構成するアイテム同士の関係を表します。次の関係のどれかを表します。

- has-hypercube 関係
- hypercube-dimension 関係
- dimension-domain 関係
- domain-member 関係
- dimension-default 関係

6.7.9 XBRLDimensionalRelationshipSet インターフェース

```
public interface XBRLDimensionalRelationshipSet
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

プライマリー・アイテムに関連づいている DRS を表します。

メソッドの概要：

XBRLDimensionalRelationshipSet インターフェースのメソッドの概要を次の表に示します。

表 6-75　XBRLDimensionalRelationshipSet インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLTaxonomyItem	getTaxonomyItem() DRS に関連づいているプライマリー・アイテムの要素宣言を取得します。	2.1	1.0
2	java.util.List<java.lang.String>	getRoles() DRS に含まれている has-hypercube 関係が定義された，extended 型要素の xlink:role 属性のリストを取得します。	2.1	1.0
3	java.util.List<XBRLHasHypercubeRelationship>	getHasHypercubeRelationships(java.lang.String role) DRS に含まれている has-hypercube 関係のうち，引数 role に指定した値と同じ xlink:role 属性を持つ extended 型要素に定義された，has-hypercube 関係のリストを取得します。	2.1	1.0
4	java.util.List<XBRLHypercubeDimensionRelationship>	getHypercubeDimensionRelationships(XBRLHasHypercubeRelationship relationship) DRS に含まれている hypercube-dimension 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている hypercube-dimension 関係のリストを取得します。	2.1	1.0
5	java.util.List<XBRLDimensionDomainRelationship>	getDimensionDomainRelationships(XBRLHypercubeDimensionRelationship relationship) DRS に含まれている dimension-domain 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている dimension-domain 関係のリストを取得します。	2.1	1.0
6	java.util.List<XBRLDomainMemberRelationship>	getDomainMemberRelationships(XBRLDimensionDomainRelationship relationship) DRS に含まれている domain-member 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている domain-member 関係のリストを取得します。	2.1	1.0

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
7	java.util.List<XBRLDomainMemberRelationship>	getDomainMemberRelationships(XBRLDomainMemberRelationship relationship) DRS に含まれている domain-member 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている domain-member 関係のリストを取得します。	2.1	1.0
8	XBRLDimensionDefaultRelationship	getDimensionDefaultRelationship(XBRLHypercubeDimensionRelationship relationship) DRS に含まれている dimension-default 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている dimension-default 関係を取得します。	2.1	1.0
9	java.util.List<XBRLDRSDimension>	getDRSDimensions(java.lang.String role, int contextElementType) DRS に含まれているディメンジョンの要素宣言のうち，インスタンス文書に記述できるディメンジョンの要素宣言のリストを返します。	2.1	1.0
10	java.util.List<XBRLDRSDimension>	getDRSDimensions(java.lang.String role) DRS に含まれているディメンジョンの要素宣言のうち，XBRL インスタンス文書に記述できるディメンジョンの要素宣言のリストを返します。	2.1	1.0
11	java.util.List<XBRLDRSMember>	getDRSMembers(XBRLDRSDimension dimension) DRS に含まれている要素宣言のうち，指定されたディメンジョンのメンバとして，インスタンス文書に記述できる要素宣言のリストを返します。	2.1	1.0
12	XBRLDRSMember	getDRSDefaultMember(XBRLDRSDimension dimension) DRS に含まれている要素宣言のうち，インスタンス文書に記述できる要素宣言のリストを返します。	2.1	1.0

（1） getTaxonomyItem

```
public XBRLTaxonomyItem getTaxonomyItem()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

DRS に関連づいているプライマリー・アイテムの要素宣言を取得します。

戻り値：

この DRS に関連づいているプライマリー・アイテムの要素宣言を表す XBRLTaxonomyItem オブジェクトを返します。

(2) getRoles

```
public java.util.List<java.lang.String> getRoles()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

DRS に含まれている has-hypercube 関係が定義された，extended 型要素の xlink:role 属性のリストを取得します。

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

この DRS に含まれている has-hypercube 関係が定義された，extended 型要素の xlink:role 属性のリストを返します。

(3) getHasHypercubeRelationships

```
public java.util.List<XBRLHasHypercubeRelationship>
getHasHypercubeRelationships ( java.lang.String role )
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

DRS に含まれている has-hypercube 関係のうち，引数 role に指定した値と同じ xlink:role 属性を持つ extended 型要素に定義された，has-hypercube 関係のリストを取得します。

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

role - extended 型要素の xlink:role 属性の値を指定します。

戻り値：

この DRS に含まれている has-hypercube 関係のうち，引数 role に指定した値と同じ xlink:role 属性を持つ extended 型要素に定義された，has-hypercube 関係のリストを返します。該当する has-hypercube 関係がない場合は，要素数が 0 であるリス

トを返します。

例外：

XBRLDOMRuntimeException -

- 引数 role に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 role に空の URI 参照を指定した場合にスローされます。［エラーコード：20110］

（4） getHypercubeDimensionRelationships

```
public java.util.List<XBRLHypercubeDimensionRelationship>
getHypercubeDimensionRelationships ( XBRLHasHypercubeRelationship
relationship )
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

DRS に含まれている hypercube-dimension 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている hypercube-dimension 関係のリストを取得します。

引数 relationship と結び付いている hypercube-dimension 関係とは，次のどちらかのことです。

- 引数 relationship に xbrldt:targetRole 属性が設定されている場合は，その xbrldt:targetRole 属性の値と同じ値の xlink:role 属性を持つ hypercube-dimension 関係を示します。
- 引数 relationship に xbrldt:targetRole 属性が設定されていない場合は，引数 relationship の xlink:role 属性の値と同じ値の xlink:role 属性を持つ hypercube-dimension 関係を示します。

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

relationship - この DRS に含まれている has-hypercube 関係を指定します。

戻り値：

この DRS に含まれている hypercube-dimension 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている hypercube-dimension 関係のリストを返します。該当する hypercube-dimension 関係がない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 relationship に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 relationship に指定した XBRLHasHypercubeRelationship オブジェクトを所有する XBRLTaxonomyDocument オブジェクトが，この XBRLDimensionalRelationshipSet オブジェクトを所有する XBRLTaxonomyDocument オブジェクトと異なる場合にスローされます。［エラーコード：20034］

（5） getDimensionDomainRelationships

```
public java.util.List<XBRLDimensionDomainRelationship>
getDimensionDomainRelationships (
XBRLHypercubeDimensionRelationship relationship )
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

DRS に含まれている dimension-domain 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている dimension-domain 関係のリストを取得します。
引数 relationship と結び付いている hypercube-dimension 関係とは，次のどちらかのことです。

- 引数 relationship に xbrldt:targetRole 属性が設定されている場合は，その xbrldt:targetRole 属性の値と同じ値の xlink:role 属性を持つ dimension-domain 関係を示します。
- 引数 relationship に xbrldt:targetRole 属性が設定されていない場合は，引数 relationship の xlink:role 属性の値と同じ値の xlink:role 属性を持つ dimension-domain 関係を示します。

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

relationship - この DRS に含まれている hypercube-dimension 関係を指定します。

戻り値：

この DRS に含まれている dimension-domain 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている dimension-domain 関係のリストを返します。該当する dimension-domain 関係がない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 relationship に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 relationship に指定した XBRLHypercubeDimensionRelationship オブジェクトを所有している XBRLTaxonomyDocument オブジェクトが，この XBRLDimensionalRelationshipSet オブジェクトを所有している XBRLTaxonomyDocument オブジェクトと異なる場合にスローされます。［エラーコード：20034］

（6） getDomainMemberRelationships

```
public java.util.List<XBRLDomainMemberRelationship>
getDomainMemberRelationships ( XBRLDimensionDomainRelationship
relationship )
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

DRS に含まれている domain-member 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている domain-member 関係のリストを取得します。

引数 relationship と結び付いている domain-member 関係とは，次のどちらかのことです。

- 引数 relationship に xbrldt:targetRole 属性が設定されている場合は，その xbrldt:targetRole 属性の値と同じ値の xlink:role 属性を持つ domain-member 関係を示します。
- 引数 relationship に xbrldt:targetRole 属性が設定されていない場合は，引数 relationship の xlink:role 属性の値と同じ値の xlink:role 属性を持つ domain-member 関係を示します。

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

relationship - この DRS に含まれている dimension-domain 関係を指定します。

戻り値：

この DRS に含まれている domain-member 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている domain-member 関係のリストを返します。該当する domain-member 関係がない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 relationship に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 relationship に指定した XBRLDimensionDomainRelationship オブジェクトを所有している XBRLTaxonomyDocument オブジェクトが，この XBRLDimensionalRelationshipSet オブジェクトを所有している XBRLTaxonomyDocument オブジェクトと異なる場合にスローされます。［エラーコード：20034］

（7） getDomainMemberRelationships

```
public java.util.List<XBRLDomainMemberRelationship>
getDomainMemberRelationships ( XBRLDomainMemberRelationship
relationship )
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

DRS に含まれている domain-member 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている domain-member 関係のリストを取得します。

引数 relationship と結び付いている domain-member 関係とは，次のどちらかのことです。

- 引数 relationship に xbrldt:targetRole 属性が設定されている場合は，その xbrldt:targetRole 属性の値と同じ値の xlink:role 属性を持つ domain-member 関係を示します。
- 引数 relationship に xbrldt:targetRole 属性が設定されていない場合は，引数 relationship の xlink:role 属性の値と同じ値の xlink:role 属性を持つ domain-member 関係を示します。

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

relationship - この DRS に含まれている domain-member 関係を指定します。

戻り値：

この DRS に含まれている domain-member 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている domain-member 関係のリストを返します。該当する domain-member 関係がない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 relationship に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 relationship に指定した XBRLDomainMemberRelationship オブジェクトを所有している XBRLTaxonomyDocument オブジェクトが，この XBRLDimensionalRelationshipSet オブジェクトを所有している XBRLTaxonomyDocument オブジェクトと異なる場合にスローされます。［エラーコード：20034］

（8） getDimensionDefaultRelationship

```
public XBRLDimensionDefaultRelationship
getDimensionDefaultRelationship (
  XBRLHypercubeDimensionRelationship  relationship )
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

DRS に含まれている dimension-default 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている dimension-default 関係を取得します。

引数 relationship と結び付いている dimension-default 関係とは，次のどちらかのことです。

- 引数 relationship に xbrldt:targetRole 属性が設定されている場合は，その xbrldt:targetRole 属性の値と同じ値の xlink:role 属性を持つ dimension-default 関係を示します。
- 引数 relationship に xbrldt:targetRole 属性が設定されていない場合は，引数 relationship の xlink:role 属性の値と同じ値の xlink:role 属性を持つ dimension-default 関係を示します。

引数：

relationship - この DRS に含まれている hypercube-dimension 関係を指定します。

戻り値：

この DRS に含まれている dimension-default 関係のうち，引数 relationship の to 側のリソースを始点に持ち，かつ，引数 relationship と結び付いている dimension-default 関係を返します。該当する dimension-default 関係がない場合は，null を返します。

例外：

XBRLDOMRuntimeException -

- 引数 relationship に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 relationship に指定した XBRLHypercubeDimensionRelationship オブジェクトを所有している XBRLTaxonomyDocument オブジェクトが，この XBRLDimensionalRelationshipSet オブジェクトを所有している XBRLTaxonomyDocument オブジェクトと異なる場合にスローされます。［エラーコード：20034］

(9) getDRSDimensions

```
public java.util.List<XBRLDRSDimension> getDRSDimensions (
java.lang.String  role ,
    int contextElementType )
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

DRS に含まれているディメンジョンの要素宣言のリストを返します。リストに含まれるのは，次の条件をすべて満たしているディメンジョンの要素宣言です。

- この DRS に含まれている has-hypercube 関係のうち，引数 role に指定した値と同じ xlink:role 属性を持つ has-hypercube 関係からたどることができる
- この DRS に含まれている has-hypercube 関係のうち，引数 contextElementType に指定した値と同じ xbrldt:contextElement 属性の種類を持つ has-hypercube 関係からたどることができる
- インスタンス文書に記述できる

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

role - extended 型要素の xlink:role 属性の値を指定します。

contextElementType - <link:definitionArc> 要素の xbrldt:contextElement 属性の種類を指定します。指定できる値は次のどちらかです。

- XBRLHasHypercubeRelationship.CONTEXTELEMENT_SCENARIO
- XBRLHasHypercubeRelationship.CONTEXTELEMENT_SEGMENT

戻り値：

この DRS に含まれているディメンジョンの要素宣言のうち，インスタンス文書に記述できるディメンジョンの要素宣言のリストを返します。該当するディメンジョンの要素宣言がない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 role に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 role に空の URI 参照を指定した場合にスローされます。［エラーコード：20110］
- 引数 contextElementType に不正な値を指定した場合にスローされます。引数 contextElementType に指定できる値は，XBRLHasHypercubeRelationship.CONTEXTELEMENT_SCENARIO，または XBRLHasHypercubeRelationship.CONTEXTELEMENT_SEGMENT です。［エラーコード：20014］

（10）getDRSDimensions

```
public java.util.List<XBRLDRSDimension> getDRSDimensions (
java.lang.String role )
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

DRS に含まれているディメンジョンの要素宣言のリストを返します。リストに含まれるのは，次の条件をすべて満たしているディメンジョンの要素宣言です。

- この DRS に含まれている has-hypercube 関係のうち，引数 role に指定された値と同じ xlink:role 属性を持つ has-hypercube 関係からたどることができる
- インスタンス文書に記述できる

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

role - extended 型要素の xlink:role 属性の値を指定します。

戻り値：

この DRS に含まれているディメンジョンの要素宣言のうち，インスタンス文書に記述できるディメンジョンの要素宣言のリストを返します。該当するディメンジョンの要素宣言がない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 role に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 role に空の URI 参照を指定した場合にスローされます。［エラーコード：20110］

（11）getDRSMembers

```
public java.util.List<XBRLDRSMember> getDRSMembers (
XBRLDRSDimension dimension )
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

DRS に含まれている要素宣言のリストを返します。リストに含まれるのは，次の条件をすべて満たしている要素宣言です。

- この DRS に含まれている has-hypercube 関係のうち，引数 dimension の getRole メソッドで取得した値と同じ xlink:role 属性を持つ has-hypercube 関係からたどることができる，ディメンジョンの要素宣言のメンバである
- この DRS に含まれている has-hypercube 関係のうち，引数 dimension の getContextElementType メソッドで取得した値と同じ xbrldt:contextElement 属性の種類を持つ has-hypercube 関係からたどることができる，ディメンジョンの要素宣言のメンバである
- インスタンス文書に記述できる

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

dimension - インスタンス文書に記述できるディメンジョンを表す XBRLDRSDimension オブジェクトを指定します。

戻り値：

この DRS に含まれている要素宣言のうち，指定されたディメンジョンのメンバとして，インスタンス文書に記述できる要素宣言のリストを返します。該当する要素宣言がない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 dimension に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 dimension に指定された XBRLDRSDimension オブジェクトを所有している XBRLTaxonomyDocument オブジェクトが，この XBRLDimensionalRelationshipSet オブジェクトを所有している XBRLTaxonomyDocument オブジェクトと異なる場合にスローされます。［エラーコード：20034］

（12）getDRSDefaultMember

```
public XBRLDRSMember getDRSDefaultMember ( XBRLDRSDimension
dimension )
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

DRS に含まれている要素宣言のうち，指定されたディメンジョンのデフォルトメンバを返します。

引数：

dimension - インスタンス文書に記述できるディメンジョンを表す XBRLDRSDimension オブジェクトを指定します。

戻り値：

この DRS に含まれている要素宣言のうち，指定されたディメンジョンのデフォルトメンバを返します。該当する要素宣言がない場合は，null を返します。

例外：

XBRLDOMRuntimeException -

- 引数 dimension に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 dimension に指定された XBRLDRSDimension オブジェクトを所有している XBRLTaxonomyDocument オブジェクトが，この XBRLDimensionalRelationshipSet オブジェクトを所有している XBRLTaxonomyDocument オブジェクトと異なる場合にスローされます。［エラーコード：20034］

6.7.10 XBRLDimensionDefaultRelationship インターフェース

```
public interface XBRLDimensionDefaultRelationship
extends XBRLDimensionalRelationship
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<link:definitionArc> 要素によって定義された，dimension-default 関係を表します。

6.7.11 XBRLDimensionDomainRelationship インターフェース

```
public interface XBRLDimensionDomainRelationship
extends XBRLDimensionalRelationship
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<link:definitionArc> 要素によって定義された，dimension-domain 関係を表します。

メソッドの概要：

XBRLDimensionDomainRelationship インターフェースのメソッドの概要を次の表に示します。

表 6-76 XBRLDimensionDomainRelationship インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	boolean	getUsable() <link:definitionArc> 要素の xbrldt:usable 属性の値を取得します。	2.1	1.0
2	java.lang.String	getTargetRole() <link:definitionArc> 要素の xbrldt:targetRole 属性の値を取得します。	2.1	1.0

(1) getUsable

```
public boolean getUsable()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<link:definitionArc> 要素の xbrldt:usable 属性の値を取得します。

戻り値：

<link:definitionArc> 要素の xbrldt:usable 属性の値を返します。xbrldt:usable 属性がない場合は true（デフォルト値）を返します。

（2） getTargetRole

```
public java.lang.String getTargetRole()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<link:definitionArc> 要素の xbrldt:targetRole 属性の値を取得します。

戻り値：

<link:definitionArc> 要素の xbrldt:targetRole 属性の値を返します。
xbrldt:targetRole 属性がない場合は null を返します。

6.7.12 XBRLDomainMemberRelationship インターフェース

```
public interface XBRLDomainMemberRelationship
extends XBRLDimensionalRelationship
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<link:definitionArc> 要素によって定義された，domain-member 関係を表します。

メソッドの概要：

XBRLDimensionDomainRelationship インターフェースのメソッドの概要を次の表に示します。

表 6-77 XBRLDomainMemberRelationship インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	boolean	getUsable() <link:definitionArc> 要素の xbrldt:usable 属性の値を取得します。	2.1	1.0
2	java.lang.String	getTargetRole() <link:definitionArc> 要素の xbrldt:targetRole 属性の値を取得します。	2.1	1.0

（1） getUsable

```
public boolean getUsable()
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
<link:definitionArc> 要素の xbrldt:usable 属性の値を取得します。

戻り値：
<link:definitionArc> 要素の xbrldt:usable 属性の値を返します。xbrldt:usable 属性がない場合は true（デフォルト値）を返します。

(2) getTargetRole

```
public java.lang.String getTargetRole()
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
<link:definitionArc> 要素の xbrldt:targetRole 属性の値を取得します。

戻り値：
<link:definitionArc> 要素の xbrldt:targetRole 属性の値を返します。
xbrldt:targetRole 属性がない場合は null を返します。

6.7.13 XBRLDRSDimension インターフェース

```
public interface XBRLDRSDimension
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
インスタンス文書に記述できるディメンジョンを表します。

メソッドの概要：
XBRLDRSDimension インターフェースのメソッドの概要を次の表に示します。

表 6-78　XBRLDRSDimension インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	int	getContextElementType() <link:definitionArc> 要素の xbrldt:contextElement 属性の種類を取得します。	2.1	1.0
2	java.lang.String	getRole() extended 型要素の xlink:role 属性の値を取得します。	2.1	1.0
3	XBRLTaxonomyDimensionItem	getTaxonomyDimensionItem() ディメンジョンの要素宣言を取得します。	2.1	1.0
4	XBRLQName	getQName() ディメンジョンの要素宣言の QName を取得します。	2.1	1.0

（1） getContextElementType

```
public int getContextElementType()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<link:definitionArc> 要素の xbrldt:contextElement 属性の種類を取得します。

戻り値：

<link:definitionArc> 要素の xbrldt:contextElement 属性の種類を返します。

このメソッドが返す xbrldt:contextElement 属性の種類は，次のどちらかです。

- XBRLHasHypercubeRelationship.CONTEXTELEMENT_SCENARIO
- XBRLHasHypercubeRelationship.CONTEXTELEMENT_SEGMENT

（2） getRole

```
public java.lang.String getRole()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：
extended 型要素の xlink:role 属性の値を取得します。

戻り値：
extended 型要素の xlink:role 属性の値を返します。

（3） getTaxonomyDimensionItem

```
public XBRLTaxonomyDimensionItem getTaxonomyDimensionItem()
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
ディメンジョンの要素宣言を取得します。

戻り値：
ディメンジョンの要素宣言を返します。

（4） getQName

```
public XBRLQName getQName()
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
ディメンジョンの要素宣言の QName を取得します。

戻り値：
ディメンジョンの要素宣言の QName を返します。

6.7.14 XBRLDRSMember インターフェース

```
public interface XBRLDRSMember
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
インスタンス文書に記述できるディメンジョンのメンバを表します。

メソッドの概要：

XBRLDRSMember インターフェースのメソッドの概要を次の表に示します。

表 6-79　XBRLDRSMember インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLTaxonomyElement	getTaxonomyElement() ディメンジョンのメンバを表す要素宣言を取得します。	2.1	1.0
2	XBRLQName	getQName() ディメンジョンのメンバを表す要素宣言の QName を取得します。	2.1	1.0

（1） getTaxonomyElement

```
public XBRLTaxonomyElement getTaxonomyElement()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

ディメンジョンのメンバを表す要素宣言を取得します。

戻り値：

ディメンジョンのメンバを表す要素宣言を返します。

（2） getQName

```
public XBRLQName getQName()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

ディメンジョンのメンバを表す要素宣言の QName を取得します。

戻り値：

ディメンジョンのメンバを表す要素宣言の QName を返します。

6.7.15 XBRLHasHypercubeRelationship インターフェース

```
public interface XBRLHasHypercubeRelationship
extends XBRLDimensionalRelationship
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<link:definitionArc> 要素によって定義された，has-hypercube 関係を表します。

フィールドの概要：

XBRLHasHypercubeRelationship インターフェースのフィールドの概要を次の表に示します。

表 6-80 XBRLHasHypercubeRelationship インターフェースのフィールドの概要

項番	説明	
1	static int	CONTEXTELEMENT_SCENARIO <link:definitionArc> 要素の xbrldt:contextElement 属性の値が "scenario" であることを示します。
2	static int	CONTEXTELEMENT_SEGMENT <link:definitionArc> 要素の xbrldt:contextElement 属性の値が "segment" であることを示します。
3	static int	ARCROLE_ALL <link:definitionArc> 要素の xlink:arcrole 属性の値が "http://xbrl.org/int/dim/arcrole/all" であることを示します。
4	static int	ARCROLE_NOTALL <link:definitionArc> 要素の xlink:arcrole 属性の値が "http://xbrl.org/int/dim/arcrole/notAll" であることを示します。

メソッドの概要：

XBRLHasHypercubeRelationship インターフェースのメソッドの概要を次の表に示します。

表 6-81 XBRLHasHypercubeRelationship インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	int	getContextElementType() <link:definitionArc> 要素の xbrldt:contextElement 属性の種類を取得します。	2.1	1.0

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
2	boolean	getClosed() <link:definitionArc> 要素の xbrldt:closed 属性の値を取得します。	2.1	1.0
3	int	getHasHypercubeRelationshipType() <link:definitionArc> 要素の xlink:arcrole 属性の種類を取得します。	2.1	1.0
4	java.lang.String	getTargetRole() <link:definitionArc> 要素の xbrldt:targetRole 属性の値を取得します。	2.1	1.0

（1） getContextElementType

```
public int getContextElementType()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<link:definitionArc> 要素の xbrldt:contextElement 属性の種類を取得します。

<link:definitionArc> 要素の xbrldt:contextElement 属性の種類を次の表に示します。

表 6-82　<link:definitionArc> 要素の xbrldt:contextElement 属性の種類

項番	<link:definitionArc> 要素の xbrldt:contextElement 属性の種類 （getContextElementType() メソッドが返すコード）	xbrldt:contextElement 属性の値
1	CONTEXTELEMENT_SCENARIO	scenario
2	CONTEXTELEMENT_SEGMENT	segment

戻り値：

<link:definitionArc> 要素の xbrldt:contextElement 属性の種類を返します。

（2） getClosed

```
public boolean getClosed()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<link:definitionArc> 要素の xbrldt:closed 属性の値を取得します。

戻り値：

<link:definitionArc> 要素の xbrldt:closed 属性の値を返します。xbrldt:closed 属性がない場合は false（デフォルト値）を返します。

（3） getHasHypercubeRelationshipType

```
public int getHasHypercubeRelationshipType()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<link:definitionArc> 要素の xlink:arcrole 属性の種類を取得します。

<link:definitionArc> 要素の xlink:arcrole 属性の種類を次の表に示します。

表 6-83　<link:definitionArc> 要素の xlink:arcrole 属性の種類

項番	<link:definitionArc> 要素の xlink:arcrole 属性の種類（getHypercubeRelationshipType() メソッドが返すコード）	xlink:arcrole 属性の値
1	ARCROLE_ALL	http://xbrl.org/int/dim/arcrole/all
2	ARCROLE_NOTALL	http://xbrl.org/int/dim/arcrole/notAll

戻り値：

<link:definitionArc> 要素の xlink:arcrole 属性の種類を返します。

（4） getTargetRole

```
public java.lang.String getTargetRole()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

<link:definitionArc> 要素の xbrldt:targetRole 属性の値を取得します。

戻り値：

<link:definitionArc> 要素の xbrldt:targetRole 属性の値を返します。
xbrldt:targetRole 属性がない場合は null を返します。

6.7.16 XBRLHypercubeDimensionRelationship インターフェース

```
public interface XBRLHypercubeDimensionRelationship
extends XBRLDimensionalRelationship
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
<link:definitionArc> 要素によって定義された，hypercube-dimension 関係を表します。

メソッドの概要：
XBRLHypercubeDimensionRelationship インターフェースのメソッドの概要を次の表に示します。

表 6-84 XBRLHypercubeDimensionRelationship インターフェースのメソッドの概要

説明		XBRL のバージョン	XBRL Dimensions のバージョン
java.lang.String	getTargetRole() <link:definitionArc> 要素の xbrldt:targetRole 属性の値を取得します。	2.1	1.0

(1) getTargetRole

```
public java.lang.String getTargetRole()
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
<link:definitionArc> 要素の xbrldt:targetRole 属性の値を取得します。

戻り値：
<link:definitionArc> 要素の xbrldt:targetRole 属性の値を返します。
xbrldt:targetRole 属性がない場合は null を返します。

6.7.17 XBRLIllegalCycleException クラス

```
public class XBRLIllegalCycleException extends XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

関係ネットワークに対するサイクル検証で検出されたエラーを通知するために，uCosminexus Business Reporting Processor によって生成されます。

この例外クラスは，エラーコードが 15104 または 15105 の場合に使用されます。

メソッドの概要：

XBRLIllegalCycleException クラスのメソッドの概要を次の表に示します。

表 6-85 XBRLIllegalCycleException クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getLinkName() エラーが検出された関係ネットワークを定義している拡張リンクの種類を取得します。	2.1
2	java.lang.String	getExtendedLinkRole() エラーが検出された関係ネットワークを定義している拡張リンクのロールを取得します。	2.1
3	java.lang.String	getArcrole() エラーが検出された関係ネットワークを定義しているアークのアークロールを取得します。	2.1
4	java.lang.String	getCyclesAllowed() エラーが検出された関係ネットワークで許可されるサイクルの種類を取得します。	2.1
5	XBRLRelationship[]	getErrorRoutes() 誤っていた経路を取得します。	2.1
6	XBRLRelationship[]	getRelationRoutes() 誤っていた経路に関連するもう一つの経路を取得します。	2.1

(1) getLinkName

```
public java.lang.String getLinkName()
```

XBRL のバージョン：

2.1

説明：

エラーが検出された関係ネットワークを定義している拡張リンクの種類を取得します。

このメソッドが返す文字列は，拡張リンクを記述している extended 型要素のローカ

ル名となります。例えば，エラーが検出された関係のネットワークがプレゼンテーションリンクによって定義されているものである場合，プレゼンテーションリンクを記述する extended 型要素は link:presentationLink 要素であるため，このメソッドが返す文字列は presentationLink となります。

戻り値：
エラーが検出された関係ネットワークを定義している拡張リンクの種類を返します。

（2） getExtendedLinkRole

```
public java.lang.String getExtendedLinkRole()
```

XBRL のバージョン：
2.1

説明：
エラーが検出された関係ネットワークを定義している拡張リンクのロールを取得します。

戻り値：
エラーが検出された関係ネットワークを定義している拡張リンクのロールを返します。

（3） getArcrole

```
public java.lang.String getArcrole()
```

XBRL のバージョン：
2.1

説明：
エラーが検出された関係ネットワークを定義しているアークのアークロールを取得します。

戻り値：
エラーが検出された関係ネットワークを定義しているアークのアークロールを返します。

（4） getCyclesAllowed

```
public java.lang.String getCyclesAllowed()
```

XBRL のバージョン：
2.1

説明：
エラーが検出された関係ネットワークで許可されるサイクルの種類を取得します。このメソッドには undirected または none のどちらかの文字列が返ります。サイクルの種類については，「2.3.2(6)　関係ネットワーク上のサイクルの検証【完全な妥当性検証の場合だけ有効】」を参照してください。

戻り値：

エラーが検出された関係ネットワークで許可されるサイクルの種類を返します。

(5) getErrorRoutes

```
public XBRLRelationship[] getErrorRoutes()
```

XBRL のバージョン：

2.1

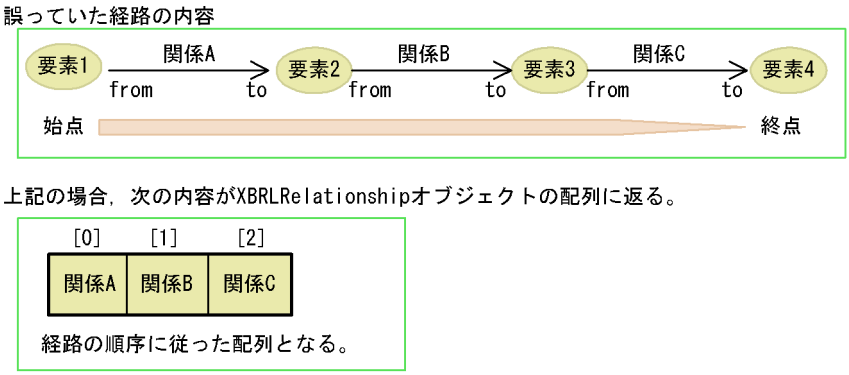
説明：

誤っていた経路を取得します。

エラーコードが 15105（経路が複数ある旨のエラー）の場合は，XBRLIllegalCycleException.getRelationRoutes() メソッドを呼び出すことによって，もう一つの経路を取得できます。

このメソッドは，誤っていた経路を構成している関係を XBRLRelationship オブジェクトの配列として返します。この配列に格納されている XBRLRelationship オブジェクトは，経路の始点から終点へ向かう順番で並んでいます。例を次の図に示します。

図 6-15　XBRLIllegalCycleException.getErrorRoutes() メソッドが返す配列の例



戻り値：

誤っていた経路を示す XBRLRelationship オブジェクトの配列を返します。

(6) getRelationRoutes

```
public XBRLRelationship[] getRelationRoutes()
```

XBRL のバージョン：

2.1

説明：

誤っていた経路に関連するもう一つの経路を取得します。

このメソッドは，エラーコードが 15105（経路が複数ある旨のエラー）の場合に使用します。エラーコードが 15104（経路が循環している旨のエラー）の場合，この

メソッドは null を返します。
このメソッドが返す配列に格納されている XBRLRelationship オブジェクトは，XBRLIllegalCycleException.getErrorRoutes() メソッドと同様に，経路の始点から終点へ向かう順番で並んでいます。

戻り値：
エラーコードが 15105 の場合は，誤っていた経路に関連するもう一つの経路を示す XBRLRelationship オブジェクトの配列を返します。エラーコードが 15104 の場合は null を返します。

6.7.18 XBRLDimensionIllegalCycleException クラス

```
public class XBRLDimensionIllegalCycleException
extends XBRLDimensionException
```

XBRL のバージョン：
2.1

XBRL Dimensions のバージョン：
1.0

説明：
DRS の関係ネットワークに対するサイクル検証で検出されたエラーを通知するために，uCosminexus Business Reporting Processor によって生成されます。

メソッドの概要：
XBRLDimensionIllegalCycleException クラスのメソッドの概要を次の表に示します。

表 6-86 XBRLDimensionIllegalCycleException クラスのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	java.lang.String	getLinkName() エラーが検出された DRS の関係ネットワークを定義している拡張リンクの種類を取得します。	2.1	1.0
2	java.lang.String	getExtendedLinkRole() エラーが検出された DRS の関係ネットワークを定義している拡張リンクのロールを取得します。	2.1	1.0
3	java.lang.String	getArcrole() エラーが検出された DRS の関係ネットワークを定義しているアークのアークロールを取得します。	2.1	1.0

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
4	java.lang.String	getCyclesAllowed() エラーが検出された DRS の関係ネットワークで許可されるサイクルの種類を取得します。	2.1	1.0
5	XBRLRelationship[]	getErrorRoutes() 誤っていた経路を取得します。	2.1	1.0

(1) getLinkName

```
public java.lang.String getLinkName()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

エラーが検出された DRS の関係ネットワークを定義している拡張リンクの種類を取得します。

このメソッドが返す文字列は，拡張リンクを記述している extended 型要素のローカル名です。

戻り値：

エラーが検出された DRS の関係ネットワークを定義している拡張リンクの種類を返します。

(2) getExtendedLinkRole

```
public java.lang.String getExtendedLinkRole()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

エラーが検出された DRS の関係ネットワークを定義している拡張リンクのロールを取得します。

戻り値：

エラーが検出された DRS の関係ネットワークを定義している拡張リンクのロールを返します。

(3) getArcrole

```
public java.lang.String getArcrole()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

エラーが検出された DRS の関係ネットワークを定義しているアークのアークロールを取得します。

戻り値：

エラーが検出された DRS の関係ネットワークを定義しているアークのアークロールを返します。

(4) getCyclesAllowed

```
public java.lang.String getCyclesAllowed()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

エラーが検出された DRS の関係ネットワークで許可されるサイクルの種類を取得します。

このメソッドが返す文字列は次のどちらかです。

- undirected
- none

サイクルの種類については，「2.3.2(6)　関係ネットワーク上のサイクルの検証【完全な妥当性検証の場合だけ有効】」を参照してください。

戻り値：

エラーが検出された DRS の関係ネットワークで許可されるサイクルの種類を返します。

(5) getErrorRoutes

```
public XBRLRelationship[] getErrorRoutes()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

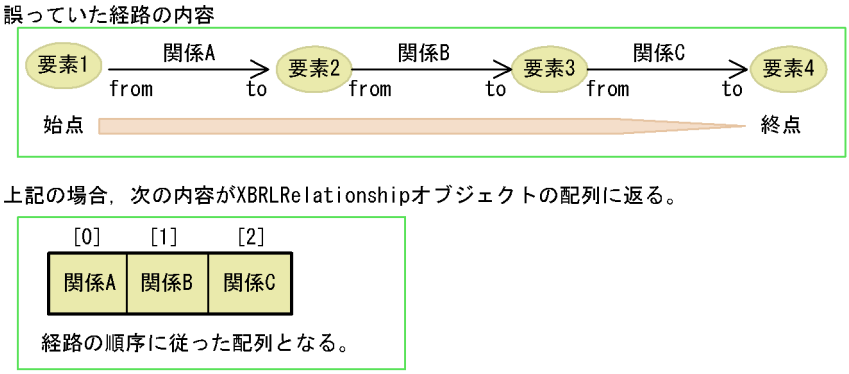
1.0

説明：

誤っていた経路を取得します。

このメソッドは，誤っていた経路を構成している関係を XBRLRelationship オブジェクトの配列として返します。この配列に格納されている XBRLRelationship オブジェクトは，経路の始点から終点へ向かう順番で並んでいます。例を次の図に示します。

図 6-16　XBRLDimensionIllegalCycleException.getErrorRoutes() メソッドが返す配列の例



戻り値：

誤っていた経路を示す XBRLRelationship オブジェクトの配列を返します。

6.8 com.hitachi.xbrl.taxonomy パッケージ

com.hitachi.xbrl.taxonomy パッケージは，タクソノミー文書を参照するためのインターフェースを提供します。

インターフェースの概要：

com.hitachi.xbrl.taxonomy パッケージのインターフェースを次の表に示します。

表 6-87 com.hitachi.xbrl.taxonomy パッケージのインターフェース

項番	インターフェース	説明	XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLArcroleType	カスタムアークロールを定義する <link:arcroleType> 要素を表します。	2.1	該当しない
2	XBRLLabel	要素のラベルを表します。	2.0 および 2.1	該当しない
3	XBRLLinkbase	タクソノミー文書を構成する XML 文書の一つであるリンクベースを表します。	2.0 および 2.1	該当しない
4	XBRLTaxonomyLinkbaseRef	タクソノミー文書に記述される <link:linkbaseRef> 要素を表します。	2.1	該当しない
5	XBRLPart	文献への参照に含まれるパートを表します。	2.0 および 2.1	該当しない
6	XBRLReference	文献への参照を表します。	2.0 および 2.1	該当しない
7	XBRLTaxonomyRoleRef	タクソノミー文書に記述される <link:roleRef> 要素を表します。	2.1	該当しない
8	XBRLRoleType	カスタムロールを定義する <link:roleType> 要素を表します。	2.1	該当しない
9	XBRLTaxonomyAttributeDeclaration	型定義の属性宣言を表します。	2.1	該当しない
10	XBRLTaxonomyBody	タクソノミー文書を構成する XML 文書の一つであるタクソノミー本体を表します。	2.0 および 2.1	該当しない
11	XBRLTaxonomyComplexTypeDefinition	complexType の型定義を表します。	2.1	該当しない
12	XBRLTaxonomyComponent	タクソノミー文書を構成する XML 文書（タクソノミー本体またはリンクベース）を表します。	2.0 および 2.1	該当しない
13	XBRLTaxonomyDocument	タクソノミー文書全体を表します。	2.0 および 2.1	該当しない
14	XBRLTaxonomyDocumentBuilder	タクソノミー文書から XBRLTaxonomyDocument オブジェクトを取得するための API を定義します。	2.0 および 2.1	該当しない

項番	インターフェース	説明	XBRL のバージョン	XBRL Dimensions のバージョン
15	XBRLTaxonomyDocumentBuilderFactory	XBRLTaxonomyDocumentBuilder オブジェクトを生成するための API を定義します。	2.0 および 2.1	該当しない
16	XBRLTaxonomyElement	タクソノミー文書に記述される要素の要素宣言を表します。	2.0 および 2.1	該当しない
17	XBRLTaxonomyItem	アイテムの要素宣言を表します。	2.0 および 2.1	該当しない
18	XBRLTaxonomyNode	タクソノミー文書を構成するノードを表します。	2.0 および 2.1	該当しない
19	XBRLTaxonomyOtherElement	アイテムまたはタプル以外の要素の要素宣言を表します。	2.0 および 2.1	該当しない
20	XBRLTaxonomySimpleTypeDefinition	simpleType の型定義を表します。	2.1	該当しない
21	XBRLTaxonomyTuple	タプルの要素宣言を表します。	2.0 および 2.1	該当しない
22	XBRLTaxonomyTypeDefinition	タクソノミー文書に記述される型定義を表します。	2.1	該当しない
23	XBRLTaxonomyDimensionItem	ディメンジョンの要素宣言を表します。	2.1	1.0
24	XBRLTaxonomyHypercubeItem	ハイパーキューブの要素宣言を表します。	2.1	1.0

6.8.1 XBRLArcroleType インターフェース

```
public interface XBRLArcroleType
extends XBRLTaxonomyNode
```

XBRL のバージョン：

2.1

説明：

カスタムアークロールを定義する <link:arcroleType> 要素を表します。

メソッドの概要：

XBRLArcroleType インターフェースのメソッドの概要を次の表に示します。

表 6-88　XBRLArcroleType インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getId() 　<link:arcroleType> 要素の id 属性の値を取得します。	2.1
2	java.lang.String	getArcroleURI() 　カスタムアークロールの arcroleURI 属性の値を取得します。	2.1
3	java.lang.String	getCyclesAllowed() 　<link:arcroleType> 要素の cyclesAllowed 属性の値を取得します。	2.1
4	java.util.List<XBRLQName>	getUsedOn() 　カスタムアークロールを使用できる要素のリストを取得します。	2.1
5	boolean	isUsedOn(java.lang.String namespaceURI, java.lang.String name) 　指定した要素で，カスタムアークロールが使用できるかどうかを調べます。	2.1

（1） getId

```
public java.lang.String getId()
```

XBRL のバージョン：

2.1

説明：

<link:arcroleType> 要素の id 属性の値を取得します。

戻り値：

この <link:arcroleType> 要素の id 属性の値を返します。この <link:arcroleType> 要素に id 属性がない場合は null を返します。

（2） getArcroleURI

```
public java.lang.String getArcroleURI()
```

XBRL のバージョン：

2.1

説明：

カスタムアークロールの arcroleURI 属性の値を取得します。

戻り値：

この <link:arcroleType> 要素によって定義されるカスタムアークロールの arcroleURI 属性の値を返します。

(3) getCyclesAllowed

```
public java.lang.String getCyclesAllowed()
```

XBRL のバージョン：

2.1

説明：

<link:arcroleType> 要素の cyclesAllowed 属性の値を取得します。

cyclesAllowed 属性の値は，カスタムアークロールを使用して記述された関係のネットワークで，どの種類のサイクルが許可されるかを示します。

戻り値：

この <link:arcroleType> 要素の cyclesAllowed 属性の値を返します。

(4) getUsedOn

```
public java.util.List<XBRLQName> getUsedOn()
```

XBRL のバージョン：

2.1

説明：

カスタムアークロールを使用できる要素のリストを取得します。

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

この <link:arcroleType> 要素によって定義されるカスタムアークロールを使用できる要素のリストを返します。リストには XBRLQName オブジェクトが格納されていて，各 XBRLQName オブジェクトは，カスタムアークロールを使用できる要素の名前空間 URI と要素名を保持します。

(5) isUsedOn

```
public boolean isUsedOn(java.lang.String namespaceURI,
                        java.lang.String name)
```

XBRL のバージョン：

2.1

説明：

指定した要素で，カスタムアークロールが使用できるかどうかを調べます。

引数：

namespaceURI - 要素の名前空間 URI を指定します。

name - 要素の名前を指定します。

戻り値：

この <link:roleType> 要素によって定義されるカスタムアークロールが，引数で指

定した要素で利用できる場合は true，利用できない場合は false を返します。

例外：

XBRLDOMRuntimeException -

- 引数 name に null を指定した場合にスローされます。［エラーコード：20002］

6.8.2 XBRLLabel インターフェース

```
public interface XBRLLabel
extends XBRLTaxonomyNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素のラベルを表します。

- XBRL2.0 の場合：
  ラベルがラベルリンク外で定義されている場合は，その URI だけを取得できます。一方，ラベルがラベルリンク内の <link:label> 要素で定義されている場合は，xml:lang 属性，xlink:role 属性，xlink:title 属性の各属性の値およびラベルの内容を参照できます。ラベルがラベルリンク外で定義されているかどうかは，isRemote() メソッドを呼び出して調べることができます。詳細については，XBRLLabel インターフェースの「6.8.2(1)　isRemote」を参照してください。
- XBRL2.1 の場合：
  ラベルリンクの外で定義されているラベルの内容を参照できます。このため，ラベルがラベルリンクの外で定義されているかどうかに関係なく，isRemote() メソッドは常に false を返します。

フィールドの概要：

XBRLLabel インターフェースのフィールドの概要を次の表に示します。

表 6-89　XBRLLabel インターフェースのフィールドの概要

項番	説明	
1	static java.lang.String	STANDARD このラベルの役割は http://www.xbrl.org/linkprops/label/standard です。XBRL2.0 の場合だけ使用できます。
2	static java.lang.String	TOTAL このラベルの役割は http://www.xbrl.org/linkprops/label/total です。XBRL2.0 の場合だけ使用できます。

メソッドの概要：

XBRLLabel インターフェースのメソッドの概要を次の表に示します。

表 6-90　XBRLLabel インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	boolean	isRemote() ラベルがラベルリンクの外で定義されているかどうかを調べます。	2.0 および 2.1
2	java.lang.String	getHref() ラベルがラベルリンクの外で定義されている場合に，そのラベルの URI を取得します。	2.0
3	java.lang.String	getLang() ラベルの言語を取得します。	2.0 および 2.1
4	java.lang.String	getRole() ラベルのロールを取得します。	2.0 および 2.1
5	java.lang.String	getTitle() ラベルのタイトルを取得します。	2.0 および 2.1
6	java.lang.String	getValue() ラベルの内容を表す文字列を取得します。	2.0 および 2.1
7	org.w3c.dom.Node	getContainerNode() ラベルの内容を保持するノードを取得します。	2.0 および 2.1

(1) isRemote

```
public boolean isRemote()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ラベルがラベルリンクの外で定義されているかどうかを調べます。

- XBRL2.0 の場合：
  ラベルがラベルリンク外で定義されている場合は，このメソッドは true を返します。この場合は，getHref() を呼び出して，ラベルリンク外で定義されるラベルの URI を取得できます。なお，このメソッドが true を返す場合は，getLang()，getRole()，getTitle()，getValue()，getContainerNode() の各メソッドを呼び出してはいけません。
  ラベルがラベルリンク内の <link:label> 要素で定義されている場合は，このメソッドは false を返します。この場合は，getLang()，getRole()，getTitle()，getValue()，getContainerNode() の各メソッドを呼び出して，ラベルの各種情報を取得できます。なお，このメソッドが false を返す場合は，getHref() を呼び出してはいけません。
- XBRL2.1 の場合：
  このメソッドは，ラベルがラベルリンクの外で定義されているかどうかに関係なく，常に false を返します。

戻り値：

このラベルがラベルリンク外で定義されている場合は true，このラベルがラベルリンク内の <link:label> 要素で定義されている場合は false を返します。

（2） getHref

```
public java.lang.String getHref()
```

XBRL のバージョン：

2.0

説明：

ラベルがラベルリンクの外で定義されている場合に，そのラベルの URI を取得します。

このメソッドが返す URI は，ラベルリンクの外で定義されているラベルを指す，<link:loc> 要素の xlink:href 属性の値です。

戻り値：

ラベルリンク外で定義されているラベルの URI を表す xlink:href 属性の値を返します。

例外：

XBRLDOMRuntimeException -

- このラベルがラベルリンク内の <link:label> 要素で定義されている場合にスローされます。［エラーコード：20009］

（3） getLang

```
public java.lang.String getLang()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ラベルの言語を取得します。

XBRL2.0 の場合は，ラベルがラベルリンク内で定義されている場合に，ラベルの言語を取得します。

戻り値：

このラベルの言語を表す xml:lang 属性の値を返します。

例外：

XBRLDOMRuntimeException -

- XBRL2.0 の場合：このラベルがラベルリンク外で定義されている場合にスローされます。［エラーコード：20008］

（4） getRole

```
public java.lang.String getRole()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ラベルのロールを取得します。
XBRL2.0 の場合は，ラベルがラベルリンク内で定義されている場合に，ラベルのロールを取得します。

戻り値：

このラベルのロールを表す xlink:role 属性の値を返します。このラベルが xlink:role 属性の値を持たない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- XBRL2.0 の場合：このラベルがラベルリンク外で定義されている場合にスローされます。［エラーコード：20008］

(5) getTitle

```
public java.lang.String getTitle()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ラベルのタイトルを取得します。ラベルのタイトルは，label 要素の xlink:title 属性の値です。
XBRL2.0 の場合は，ラベルがラベルリンク内で定義されている場合に，ラベルのタイトルを取得します。

戻り値：

このラベルのタイトルを表す xlink:title 属性の値を返します。xlink:title 属性の値がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- XBRL2.0 の場合：このラベルがラベルリンク外で定義されている場合にスローされます。［エラーコード：20008］

(6) getValue

```
public java.lang.String getValue()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ラベルの内容を表す文字列を取得します。
XBRL2.0 の場合は，ラベルがラベルリンク内で定義されている場合に，ラベルの内

容を表す文字列を取得します。
ラベルの内容に XHTML のタグが含まれている場合は，このメソッドは null を返します。ラベルの内容に含まれる XHTML のタグを参照したい場合は，getContainerNode() を使用します。

戻り値：
このラベルの内容を表す文字列を返します。ラベルの内容に XHTML のタグが含まれている場合は null を返します。

例外：
XBRLDOMRuntimeException -
- XBRL2.0 の場合：このラベルがラベルリンク外で定義されている場合にスローされます。［エラーコード：20008］

（7） getContainerNode

```
public org.w3c.dom.Node getContainerNode()
```

XBRL のバージョン：
2.0 および 2.1

説明：
ラベルの内容を保持するノードを取得します。
XBRL2.0 の場合は，ラベルがラベルリンク内で定義されている場合に，ラベルの内容を保持するノードを取得します。
ラベルの内容に XHTML のタグが含まれている場合は，それらを DOM ツリー中のノードとして参照できます。
このメソッドが返すノードに対して，更新を伴う操作を行ってはいけません。

例外：
XBRLDOMRuntimeException -
- XBRL2.0 の場合：このラベルがラベルリンク外で定義されている場合にスローされます。［エラーコード：20008］

6.8.3 XBRLLinkbase インターフェース

```
public interface XBRLLinkbase
extends XBRLTaxonomyComponent
```

XBRL のバージョン：
2.0 および 2.1

説明：
タクソノミー文書を構成する XML 文書の一つであるリンクベースを表します。

フィールドの概要：
XBRLLinkbase インターフェースのフィールドの概要を次の表に示します。

表 6-91　XBRLLinkbase インターフェースのフィールドの概要

項番	説明	
1	static int	CALCULATION_LINK 拡張リンクの種別は計算リンクです。
2	static int	DEFINITION_LINK 拡張リンクの種別は定義リンクです。
3	static int	PRESENTATION_LINK 拡張リンクの種別はプレゼンテーションリンクです。
4	static int	REFERENCE_LINK 拡張リンクの種別は文献リンクです。
5	static int	LABEL_LINK 拡張リンクの種別はラベルリンクです。

メソッドの概要：

XBRLLinkbase インターフェースのメソッドの概要を次の表に示します。

表 6-92　XBRLLinkbase インターフェースのメソッドの概要

説明		XBRL のバージョン
java.util.List<org.w3c.dom.Node>	getExtendedLinksW3CDOMNode(int extendedLinkType) <link:linkbase> 要素に定義される拡張リンクのうち，引数で指定した種別の拡張リンクに対応する org.w3c.dom.Node オブジェクトのリストを取得します。	2.1

(1) getExtendedLinksW3CDOMNode

```
public java.util.List<org.w3c.dom.Node>
getExtendedLinksW3CDOMNode(int extendedLinkType)
```

XBRL のバージョン：

2.1

説明：

<link:linkbase> 要素に定義される拡張リンクのうち，引数で指定した種別の拡張リンクに対応する org.w3c.dom.Node オブジェクトのリストを取得します。

また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

extendedLinkType - 拡張リンクの種別を示すコードを指定します。引数として指定できる値は CALCULATION_LINK，DEFINITION_LINK，PRESENTATION_LINK，REFERENCE_LINK または LABEL_LINK のいずれかの値です。

戻り値：

引数 extendedLinkType で指定された種別の拡張リンクに対応する org.w3c.dom.Node オブジェクトのリストを返します。指定された種別の拡張リンクが存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］
- 引数 extendedLinkType に指定した種別が不正な場合にスローされます。［エラーコード：20156］

6.8.4 XBRLTaxonomyLinkbaseRef インターフェース

```
public interface XBRLTaxonomyLinkbaseRef
extends XBRLTaxonomyNode
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書に記述される <link:linkbaseRef> 要素を表します。

メソッドの概要：

XBRLTaxonomyLinkbaseRef インターフェースのメソッドの概要を次の表に示します。

表 6-93 XBRLTaxonomyLinkbaseRef インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getHref() <link:linkbaseRef> 要素の xlink:href 属性の値を取得します。	2.1
2	java.lang.String	getRole() <link:linkbaseRef> 要素の xlink:role 属性の値を取得します。	2.1
3	java.lang.String	getArcrole() <link:linkbaseRef> 要素の xlink:arcrole 属性の値を取得します。	2.1
4	java.lang.String	getBase() <link:linkbaseRef> 要素の xml:base 属性の値を取得します。	2.1

(1) getHref

```
public java.lang.String getHref()
```

XBRL のバージョン：

2.1

説明：

<link:linkbaseRef> 要素の xlink:href 属性の値を取得します。

戻り値：

この <link:linkbaseRef> 要素の xlink:href 属性の値を返します。

(2) getRole

```
public java.lang.String getRole()
```

XBRL のバージョン：

2.1

説明：

<link:linkbaseRef> 要素の xlink:role 属性の値を取得します。

戻り値：

この <link:linkbaseRef> 要素の xlink:role 属性の値を返します。この <link:linkbaseRef> 要素に xlink:role 属性がない場合は null を返します。

(3) getArcrole

```
public java.lang.String getArcrole()
```

XBRL のバージョン：

2.1

説明：

<link:linkbaseRef> 要素の xlink:arcrole 属性の値を取得します。

戻り値：

この <link:linkbaseRef> 要素の xlink:arcrole 属性の値を返します。

(4) getBase

```
public java.lang.String getBase()
```

XBRL のバージョン：

2.1

説明：

<link:linkbaseRef> 要素の xml:base 属性の値を取得します。

戻り値：

この <link:linkbaseRef> 要素の xml:base 属性の値を返します。この

<link:linkbaseRef> 要素に xml:base 属性がない場合は null を返します。

6.8.5 XBRLPart インターフェース

```
public interface XBRLPart
extends XBRLTaxonomyNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

文献への参照に含まれるパートを表します。

メソッドの概要：

XBRLPart インターフェースのメソッドの概要を次の表に示します。

表 6-94 XBRLPart インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getNamespaceURI() パートが属する名前空間 URI を取得します。	2.0 および 2.1
2	java.lang.String	getName() パートの名前を取得します。	2.0 および 2.1
3	java.lang.String	getValue() パートの値を取得します。	2.0 および 2.1

(1) getNamespaceURI

```
public java.lang.String getNamespaceURI()
```

XBRL のバージョン：

2.0 および 2.1

説明：

パートが属する名前空間 URI を取得します。

戻り値：

このパートが属する名前空間 URI を返します。

(2) getName

```
public java.lang.String getName()
```

XBRL のバージョン：

2.0 および 2.1

説明：

パートの名前を取得します。

戻り値：

このパートの名前を返します。

(3) getValue

```
public java.lang.String getValue()
```

XBRL のバージョン：

2.0 および 2.1

説明：

パートの値を取得します。

戻り値：

このパートの値を返します。

6.8.6 XBRLReference インターフェース

```
public interface XBRLReference
extends XBRLTaxonomyNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

文献への参照を表します。

- XBRL2.0 の場合：
  文献への参照がリファレンスリンクの外で定義されている場合は，その URI だけを取得できます。一方，文献への参照がリファレンスリンク内の <link:reference> 要素で定義されている場合は，xlink:role 属性，xlink:title 属性の各属性の値および文献への参照に含まれるパートを参照できます。文献への参照がリファレンスリンクの外で定義されているかどうかは，isRemote() メソッドを呼び出して調べることができます。詳細については，XBRLReference インターフェースの「6.8.6(1) isRemote」を参照してください。
- XBRL2.1 の場合：
  リファレンスリンクの外で定義されている文献への参照の内容を参照できます。このため，文献への参照がリファレンスリンクの外で定義されているかどうかに関係なく，isRemote() メソッドは常に false を返します。

メソッドの概要：

XBRLReference インターフェースのメソッドの概要を次の表に示します。

表 6-95 XBRLReference インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	boolean	isRemote() 文献への参照がリファレンスリンクの外で定義されているかどうかを調べます。	2.0 および 2.1

項番	説明		XBRL のバージョン
2	java.lang.String	getHref() 文献への参照がリファレンスリンクの外で定義されている場合に，その文献への参照の URI を取得します。	2.0
3	java.lang.String	getRole() 文献への参照のロールを取得します。	2.0 および 2.1
4	java.lang.String	getTitle() 文献への参照のタイトルを取得します。	2.0 および 2.1
5	java.util.List<XBRLPart>	getParts() 文献への参照に含まれるパートのリストを取得します。	2.0 および 2.1
6	java.util.List<XBRLPart>	getParts(java.lang.String namespaceURI, java.lang.String name) 文献への参照に含まれるパートのうち，指定した名前を持つパートのリストを取得します。	2.0 および 2.1

（1） isRemote

```
public boolean isRemote()
```

XBRL のバージョン：

2.0 および 2.1

説明：

文献への参照がリファレンスリンクの外で定義されているかどうかを調べます。

- XBRL2.0 の場合：
  文献への参照がリファレンスリンクの外で定義されている場合は，このメソッドは true を返します。この場合は，getHref() を呼び出して，リファレンスリンクの外で定義される文献への参照の URI を取得できます。なお，このメソッドが true を返す場合は，getRole()，getTitle()，getParts()，getParts(java.lang.String, java.lang.String) の各メソッドを呼び出してはいけません。
  文献への参照がリファレンスリンク内の <link:reference> 要素で定義されている場合は，このメソッドは false を返します。この場合は，getRole()，getTitle()，getParts()，getParts(java.lang.String, java.lang.String) の各メソッドを呼び出して，文献への参照の各種情報を取得できます。なお，このメソッドが false を返す場合は，getHref() を呼び出してはいけません。
- XBRL2.1 の場合：
  このメソッドは，文献への参照がリファレンスリンクの外で定義されているかどうかに関係なく，常に false を返します。

戻り値：

この文献への参照がリファレンスリンクの外で定義されている場合は true，この文

献への参照がリファレンスリンク内の <link:reference> 要素で定義されている場合は false を返します。

(2) getHref

```
public java.lang.String getHref()
```

XBRL のバージョン：

2.0

説明：

文献への参照がリファレンスリンクの外で定義されている場合に，その文献への参照の URI を取得します。

このメソッドが返す URI は，リファレンスリンクの外で定義されている文献への参照を指す，<link:loc> 要素の xlink:href 属性の値です。

戻り値：

リファレンスリンクの外で定義されている文献への参照の URI を示す xlink:href 属性の値を返します。

例外：

XBRLDOMRuntimeException -

- この文献への参照がリファレンスリンク内の <link:reference> 要素で定義されている場合にスローされます。［エラーコード：20011］

(3) getRole

```
public java.lang.String getRole()
```

XBRL のバージョン：

2.0 および 2.1

説明：

文献への参照のロールを取得します。

XBRL2.0 の場合は，文献への参照がリファレンスリンク内で定義されている場合に，文献への参照のロールを取得します。

戻り値：

この文献への参照のロールを表す xlink:role 属性の値を返します。この文献への参照が xlink:role 属性の値を持たない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- XBRL2.0 の場合：この文献への参照がリファレンスリンクの外で定義されている場合にスローされます。［エラーコード：20010］

(4) getTitle

```
public java.lang.String getTitle()
```

XBRL のバージョン：

2.0 および 2.1

説明：

文献への参照のタイトルを取得します。文献への参照のタイトルは，
<link:reference> 要素の xlink:title 属性の値です。
XBRL2.0 の場合は，文献への参照がリファレンスリンク内で定義されている場合に，文献への参照のタイトルを取得します。

戻り値：

この文献への参照のタイトルを表す xlink:title 属性の値を返します。xlink:title 属性の値がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- XBRL2.0 の場合：この文献への参照がリファレンスリンクの外で定義されている場合にスローされます。［エラーコード：20010］

（5） getParts

```
public java.util.List<XBRLPart> getParts()
```

XBRL のバージョン：

2.0 および 2.1

説明：

文献への参照に含まれるパートのリストを取得します。
XBRL2.0 の場合は，文献への参照がリファレンスリンク内で定義されている場合に，文献への参照に含まれるパートのリストを取得します。
パートの順序は，タクソノミー文書での出現順と同じです。
このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

パートを表す XBRLPart オブジェクトのリストを返します。この文献への参照に含まれるパートがない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- XBRL2.0 の場合：この文献への参照がリファレンスリンクの外で定義されている場合にスローされます。［エラーコード：20010］

（6） getParts

```
public java.util.List<XBRLPart> getParts(java.lang.String
namespaceURI,
                                java.lang.String name)
```

XBRL のバージョン：

2.0 および 2.1

説明：

文献への参照に含まれるパートのうち，指定した名前を持つパートのリストを取得します。

XBRL2.0 の場合は，文献への参照がリファレンスリンク内で定義されている場合に，文献への参照に含まれるパートのうち，指定した名前を持つパートのリストを取得します。

同じ名前のパートが二つ以上存在することがあります。このため，このメソッドは，指定した名前のパートを，XBRLPart オブジェクトのリストとして返します。リストでの XBRLPart オブジェクトの順序は，タクソノミー文書での出現順と同じです。

このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

namespaceURI - パートの名前空間 URI を指定します。

name - パートの名前を指定します。

戻り値：

引数 namespaceURI で指定した名前空間 URI に属し，引数 name で指定した名前を持つパートを表す，XBRLPart オブジェクトのリストを返します。指定したパートがない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- XBRL2.0 の場合：この文献への参照がリファレンスリンクの外で定義されている場合にスローされます。［エラーコード：20010］
- 引数 namespaceURI または name に null を指定した場合にスローされます。［エラーコード：20002］

6.8.7 XBRLTaxonomyRoleRef インターフェース

```
public interface XBRLTaxonomyRoleRef
extends XBRLTaxonomyNode
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書に記述される <link:roleRef> 要素を表します。

メソッドの概要：

XBRLTaxonomyRoleRef インターフェースのメソッドの概要を次の表に示します。

表 6-96　XBRLTaxonomyRoleRef インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getRoleURI() <link:roleRef> 要素の roleURI 属性の値を取得します。	2.1
2	java.lang.String	getHref() <link:roleRef> 要素の xlink:href 属性の値を取得します。	2.1
3	java.lang.String	getRole() <link:roleRef> 要素の xlink:role 属性の値を取得します。	2.1
4	java.lang.String	getArcrole() <link:roleRef> 要素の xlink:arcrole 属性の値を取得します。	2.1

（1） getRoleURI

```
public java.lang.String getRoleURI()
```

XBRL のバージョン：

2.1

説明：

<link:roleRef> 要素の roleURI 属性の値を取得します。

戻り値：

この <link:roleRef> 要素の roleURI 属性の値を返します。

（2） getHref

```
public java.lang.String getHref()
```

XBRL のバージョン：

2.1

説明：

<link:roleRef> 要素の xlink:href 属性の値を取得します。

戻り値：

この <link:roleRef> 要素の xlink:href 属性の値を返します。

（3） getRole

```
public java.lang.String getRole()
```

XBRL のバージョン：

2.1

説明：

<link:roleRef> 要素の xlink:role 属性の値を取得します。

戻り値：

この <link:roleRef> 要素の xlink:role 属性の値を返します。この <link:roleRef> 要素に xlink:role 属性がない場合は null を返します。

(4) getArcrole

```
public java.lang.String getArcrole()
```

XBRL のバージョン：

2.1

説明：

<link:roleRef> 要素の xlink:arcrole 属性の値を取得します。

戻り値：

この <link:roleRef> 要素の xlink:arcrole 属性の値を返します。この <link:roleRef> 要素に xlink:arcrole 属性がない場合は null を返します。

6.8.8 XBRLRoleType インターフェース

```
public interface XBRLRoleType
extends XBRLTaxonomyNode
```

XBRL のバージョン：

2.1

説明：

カスタムロールを定義する <link:roleType> 要素を表します。

メソッドの概要：

XBRLRoleType インターフェースのメソッドの概要を次の表に示します。

表 6-97 XBRLRoleType インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getId() <link:roleType> 要素の id 属性の値を取得します。	2.1
2	java.lang.String	getRoleURI() カスタムロールの roleURI 属性の値を取得します。	2.1
3	java.util.List<XBRLQName>	getUsedOn() カスタムロールを使用できる要素のリストを取得します。	2.1
4	boolean	isUsedOn(java.lang.String namespaceURI, java.lang.String name) 指定した要素で，カスタムロールが使用できるかどうかを調べます。	2.1

（1） getId

```
public java.lang.String getId()
```

XBRL のバージョン：

2.1

説明：

<link:roleType> 要素の id 属性の値を取得します。

戻り値：

この <link:roleType> 要素の id 属性の値を返します。この <link:roleType> 要素に id 属性がない場合は null を返します。

（2） getRoleURI

```
public java.lang.String getRoleURI()
```

XBRL のバージョン：

2.1

説明：

カスタムロールの roleURI 属性の値を取得します。

戻り値：

この <link:roleType> 要素によって定義されるカスタムロールの roleURI 属性の値を返します。

（3） getUsedOn

```
public java.util.List<XBRLQName> getUsedOn()
```

XBRL のバージョン：

2.1

説明：

カスタムロールを使用できる要素のリストを取得します。
このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

この <link:roleType> 要素によって定義されるカスタムロールを使用できる要素のリストを返します。リストには XBRLQName オブジェクトが格納されていて，各 XBRLQName オブジェクトは，カスタムロールを使用できる要素の名前空間 URI と要素名を保持します。

（4） isUsedOn

```
public boolean isUsedOn(java.lang.String namespaceURI,
                        java.lang.String name)
```

XBRL のバージョン：

2.1

説明：

指定した要素で，カスタムロールが使用できるかどうかを調べます。

引数：

namespaceURI - 要素の名前空間 URI を指定します。

name - 要素の名前を指定します。

戻り値：

この <link:roleType> 要素によって定義されるカスタムロールが，引数で指定した要素で利用できる場合は true，利用できない場合は false を返します。

例外：

XBRLDOMRuntimeException -

- 引数 name に null を指定した場合にスローされます。［エラーコード：20002］

6.8.9 XBRLTaxonomyAttributeDeclaration インターフェース

```
public interface XBRLTaxonomyAttributeDeclaration
extends XBRLTaxonomyNode
```

XBRL のバージョン：

2.1

説明：

型定義の属性宣言を表します。

メソッドの概要：

XBRLTaxonomyAttributeDeclaration インターフェースのメソッドの概要を次の表に示します。

表 6-98 XBRLTaxonomyAttributeDeclaration インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getUse() 属性宣言の use 属性の値を取得します。	2.1
2	XBRLQName	getType() 属性宣言の型の修飾名を取得します。	2.1
3	java.lang.String	getForm() 属性宣言の form 属性の値を取得します。	2.1

(1) getUse

```
public java.lang.String getUse()
```

XBRL のバージョン：

2.1

説明：

属性宣言の use 属性の値を取得します。

タクソノミー文書の解析時に妥当性の検証を行うように設定されている場合は，このメソッドにより null が返ることはありません。名前空間 ”http://www.w3.org/2001/XMLSchema ” に属する attribute の定義のデフォルト値が返ります。

戻り値：

この属性宣言の use 属性の値を返します。妥当性の検証を行わない場合で，かつこの属性宣言に use 属性がない場合は null を返します。

（2） getType

```
public XBRLQName getType()
```

XBRL のバージョン：

2.1

説明：

属性宣言の型の修飾名を取得します。

戻り値：

この属性宣言の型の修飾名を返します。この属性宣言の型が名前を持たない型（局所型定義）である場合は null を返します。この属性宣言の型が何も指定されていない場合は，xsd:anySimpleType を表す修飾名を返します。

（3） getForm

```
public java.lang.String getForm()
```

XBRL のバージョン：

2.1

説明：

属性宣言の form 属性の値を取得します。

戻り値：

この属性宣言の form 属性の値を返します。この属性宣言に form 属性がない場合は null を返します。

6.8.10 XBRLTaxonomyBody インターフェース

```
public interface XBRLTaxonomyBody
extends XBRLTaxonomyComponent
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書を構成する XML 文書の一つであるタクソノミー本体を表します。

メソッドの概要：

XBRLTaxonomyBody インターフェースのメソッドの概要を次の表に示します。

表 6-99　XBRLTaxonomyBody インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getTargetNamespaceURI() タクソノミー本体の対象名前空間 URI を取得します。	2.0 および 2.1
2	java.lang.String	getElementFormDefault() タクソノミー本体の elementFormDefault 属性の値を取得します。	2.1
3	java.lang.String	getAttributeFormDefault() タクソノミー本体の attributeFormDefault 属性の値を取得します。	2.1
4	java.util.List<XBRLTaxonomyBody>	getImportBodies() タクソノミー本体で <xsd:import> 要素によりインポートするタクソノミー本体を取得します。	2.1
5	java.util.List<XBRLTaxonomyBody>	getIncludeBodies() タクソノミー本体で <xsd:include> 要素によりインクルードするタクソノミー本体を取得します。	2.1

(1) getTargetNamespaceURI

```
public java.lang.String getTargetNamespaceURI()
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー本体の対象名前空間 URI を取得します。対象名前空間 URI は，<xsd:schema> 要素の targetNamespace 属性の値です。

戻り値：

このタクソノミー本体の対象名前空間 URI を返します。

(2) getElementFormDefault

```
public java.lang.String getElementFormDefault()
```

XBRL のバージョン：

2.1

説明：

タクソノミー本体の elementFormDefault 属性の値を取得します。

タクソノミー文書の解析時に妥当性の検証を行うように設定されている場合は，このメソッドにより null が返ることはありません。名前空間 ”http://www.w3.org/

2001/XMLSchema ” に属する schema の定義のデフォルト値が返ります。

戻り値：

このタクソノミー本体の elementFormDefault 属性の値（“ qualified ” または “ unqualified ”）を返します。妥当性の検証を行わない場合で，かつこのタクソノミー本体に elementFormDefault 属性がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］

（3） getAttributeFormDefault

```
public java.lang.String getAttributeFormDefault()
```

XBRL のバージョン：

2.1

説明：

タクソノミー本体の attributeFormDefault 属性の値を取得します。
タクソノミー文書の解析時に妥当性の検証を行うように設定されている場合は，このメソッドにより null が返ることはありません。名前空間 ”http://www.w3.org/2001/XMLSchema ” に属する schema の定義のデフォルト値が返ります。

戻り値：

このタクソノミー本体の attributeFormDefault 属性の値（“ qualified ” または “ unqualified ”）を返します。妥当性の検証を行わない場合で，かつこのタクソノミー本体に attributeFormDefault 属性がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］

（4） getImportBodies

```
public java.util.List<XBRLTaxonomyBody> getImportBodies()
```

XBRL のバージョン：

2.1

説明：

タクソノミー本体で <xsd:import> 要素によりインポートするタクソノミー本体を取

得します。リストの各要素は，XBRLTaxonomyBody へキャストすることができます。
このメソッドが返すリストには，既知のスキーマ，および cbrp.parse.DTS.ignoringSchemas プロパティで指定したスキーマは含まれません。
また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。
継承関係がループになっているタクソノミー本体での本メソッドの呼び出しには注意してください。例えば，本メソッドで取得したタクソノミー本体に対して本メソッドを呼び出すような回帰的な処理では無限ループを引き起こす可能性があるため，注意してください。

戻り値：
このタクソノミー本体で <xsd:import> 要素によりインポートするタクソノミー本体のリストを返します。このタクソノミー本体に xsd:import 要素が存在しない場合は，要素数が 0 のリストを返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］

(5) getIncludeBodies

```
public java.util.List<XBRLTaxonomyBody> getIncludeBodies()
```

XBRL のバージョン：
2.1

説明：
タクソノミー本体で <xsd:include> 要素によりインクルードするタクソノミー本体を取得します。リストの各要素は，XBRLTaxonomyBody へキャストすることができます。
また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。
継承関係がループになっているタクソノミー本体での本メソッドの呼び出しには注意してください。例えば，本メソッドで取得したタクソノミー本体に対して本メソッドを呼び出すような回帰的な処理では無限ループを引き起こす可能性があるため，注意してください。

戻り値：
このタクソノミー本体で <xsd: include> 要素によりインクルードするタクソノミー

本体のリストを返します。このタクソノミー本体に <xsd: include> 要素が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］

6.8.11 XBRLTaxonomyComplexTypeDefinition インターフェース

```
public interface XBRLTaxonomyComplexTypeDefinition
extends XBRLTaxonomyTypeDefinition
```

XBRL のバージョン：

2.1

説明：

complexType の型定義を表します。

フィールドの概要：

XBRLTaxonomyComplexTypeDefinition インターフェースのフィールドの概要を次の表に示します。

表 6-100 XBRL XBRLTaxonomyComplexTypeDefinition インターフェースのフィールドの概要

項番	説明	
1	static int	COMPOSITOR_ALL モデルグループ all を示します。
2	static int	COMPOSITOR_CHOICE モデルグループ choice を示します。
3	static int	COMPOSITOR_SEQUENCE モデルグループ sequence を示します。
4	static int	COMPOSITOR_NONE モデルグループが存在しないことを示します。

メソッドの概要：

XBRLTaxonomyComplexTypeDefinition インターフェースのメソッドの概要を次の表に示します。

表 6-101　XBRLTaxonomyComplexTypeDefinition インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	boolean	getAbstract() 型定義の abstract 属性の値を取得します。	2.1
2	int	getBlock() 型定義の block 属性の値を表すコードを取得します。	2.1
3	int	getCompositor() 型定義のモデルグループを表すコードを取得します。	2.1
4	java.util.List<XBRLTaxonomyElement>	getElements() 型定義の要素宣言のリストを取得します。	2.1
5	java.util.List<XBRLTaxonomyAttributeDeclaration>	getTaxonomyAttributeDeclarations() 型定義の属性宣言のリストを取得します。	2.1
6	XBRLTaxonomyAttributeDeclaration	getTaxonomyAttributeDeclaration(java.lang.String namespaceURI, java.lang.String name) 指定された名前を持つ属性宣言を取得します。	2.1
7	java.util.List<org.w3c.dom.Node>	getElementDeclarationsW3CDOMNode() 型定義の要素宣言に対応する org.w3c.dom.Node オブジェクトのリストを取得します。	2.1
8	int	getDerivationMethod() 型定義の派生方法を表すコードを取得します。	2.1

(1) getAbstract

```
public boolean getAbstract()
```

XBRL のバージョン：

2.1

説明：

型定義の abstract 属性の値を取得します。

戻り値：

この型定義の abstract 属性の値を返します。この型定義に abstract 属性がない場合は，名前空間 ”http://www.w3.org/2001/XMLSchema ” に属する complexType の定義のデフォルト値が返ります。

(2) getBlock

```
public int getBlock()
```

XBRL のバージョン：

2.1

説明：

型定義の block 属性の値を表すコードを取得します。

戻り値：

この型定義の block 属性の値を表すコードを返します。block 属性の指定がなく，<xsd:schema> 要素の blockDefault 属性の指定がある場合は，blockDefault 属性の値を表すコードを返します。
値は XBRLTaxonomyTypeDefinition.DERIVATION_NONE，または XBRLTaxonomyTypeDefinition.DERIVATION_EXTENSION，XBRLTaxonomyTypeDefinition.DERIVATION_RESTRICTION の組み合わせによる論理和です（単一の場合も含みます）。#all を指定した場合は，XBRLTaxonomyTypeDefinition.DERIVATION_EXTENSION，XBRLTaxonomyTypeDefinition.DERIVATION_RESTRICTION の論理和です。

（3） getCompositor

```
public int getCompositor()
```

XBRL のバージョン：

2.1

説明：

型定義のモデルグループを表すコードを取得します。対象となるモデルグループは，以下のいずれかの条件に該当する <xsd:all> 要素，<xsd:choice> 要素，<xsd:sequence> 要素，<xsd:group> 要素が参照するモデルグループです。

- <xsd:complexType> 要素の第 1 子孫
- <xsd:complexContent> 要素の第 1 子孫の <xsd:restriction> の第 1 子孫
- <xsd:complexContent> 要素の第 1 子孫の <xsd:extension> の第 1 子孫

戻り値：

この型定義のモデルグループを表すコードを返します。<xsd:all> 要素，<xsd:choice> 要素，<xsd:sequence> 要素，モデルグループが存在しないことのいずれかを表すコードとなります。

（4） getElements

```
public java.util.List<XBRLTaxonomyElement> getElements()
```

XBRL のバージョン：

2.1

説明：

型定義の要素宣言のリストを取得します。リストの各要素は，XBRLTaxonomyElement へキャストすることができます。
このメソッドが返すリストには，以下のいずれかの条件に該当する <xsd:all> 要素，<xsd:choice> 要素，<xsd:sequence> 要素，<xsd:group> 要素が参照するモデルグループの第 1 子孫である要素宣言の ref 属性で指定された大域要素宣言のみが含まれます。

- <xsd:complexType> 要素の第 1 子孫
- <xsd:complexContent> 要素の第 1 子孫の <xsd:restriction> の第 1 子孫
- <xsd:complexContent> 要素の第 1 子孫の <xsd:extension> の第 1 子孫

また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

この型定義の要素宣言のリストを返します。説明に記載されている条件に該当する要素宣言が存在しない場合は，要素数が 0 のリストを返します。

(5) getTaxonomyAttributeDeclarations

```
public java.util.List<XBRLTaxonomyAttributeDeclaration>
getTaxonomyAttributeDeclarations()
```

XBRL のバージョン：

2.1

説明：

型定義の属性宣言のリストを取得します。リストの各要素は，XBRLTaxonomyAttributeDeclaration へキャストすることができます。

このメソッドが返すリストには，base 属性に指定された型定義の属性宣言は含まれません。また，<xsd:attributeGroup> 要素により参照される属性宣言も含まれません。

また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

この型定義の属性宣言のリストを返します。この型定義に属性宣言が存在しない場合は，要素数が 0 のリストを返します。

(6) getTaxonomyAttributeDeclaration

```
public XBRLTaxonomyAttributeDeclaration
getTaxonomyAttributeDeclaration(java.lang.String
namespaceURI,java.lang.String name)
```

XBRL のバージョン：

2.1

説明：

指定された名前を持つ属性宣言を取得します。base 属性に指定された型定義の属性宣言は取得できません。また，<xsd:attributeGroup> 要素により参照される属性宣言も取得できません。

引数：

namespaceURI - 属性宣言の名前空間 URI を指定します。局所の属性宣言を取得する場合は null を指定します。
name - 属性宣言の名前を指定します。

戻り値：
引数 namespaceURI が null でない場合は，引数 namespaceURI で指定された名前空間 URI に属し，引数 name で指定された名前を持つ大域の属性宣言の参照を表す XBRLTaxonomyAttributeDeclaration オブジェクトを返します。引数 namespaceURI が null の場合は，引数 name で指定された名前を持つ局所の属性宣言を表す XBRLTaxonomyAttributeDeclaration オブジェクトを返します。指定された名前を持つ属性宣言が存在しない場合は null を返します。

例外：
XBRLDOMRuntimeException -
- 引数 name に null を指定した場合にスローされます。［エラーコード：20002］

（7）getElementDeclarationsW3CDOMNode

```
public java.util.List<org.w3c.dom.Node>
getElementDeclarationsW3CDOMNode()
```

XBRL のバージョン：
2.1

説明：
型定義の要素宣言に対応する org.w3c.dom.Node オブジェクトのリストを取得します。
このメソッドが返すリストには，以下のいずれかの条件に該当する <xsd:all> 要素，<xsd:choice> 要素，<xsd:sequence> 要素の第 1 子孫である要素宣言に対応する org.w3c.dom.Node オブジェクトのみが含まれます。
- <xsd:complexType> 要素の第 1 子孫
- <xsd:complexContent> 要素の第 1 子孫の <xsd:restriction> の第 1 子孫
- <xsd:complexContent> 要素の第 1 子孫の <xsd:extension> の第 1 子孫

なお，<xsd:group> 要素で指定されたモデルグループの要素宣言に対応する org.w3c.dom.Node オブジェクトは含まれません。
また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：
この型定義の要素宣言に対応する org.w3c.dom.Node オブジェクトのリストを返します。説明に記載されている条件に該当する要素宣言が存在しない場合は，要素数が 0 のリストを返します。

(8) getDerivationMethod

```
public int getDerivationMethod()
```

XBRL のバージョン：

2.1

説明：

型定義の派生方法を表すコードを取得します。

戻り値：

この型定義の派生方法を表すコードを返します。値は XBRLTaxonomyTypeDefinition.DERIVATION_EXTENSION，または XBRLTaxonomyTypeDefinition.DERIVATION_RESTRICTION です。この型定義に <xsd:restriction> 要素も <xsd:extension> 要素も存在しない場合は，XBRLTaxonomyTypeDefinition.DERIVATION_RESTRICTION を返します。

6.8.12 XBRLTaxonomyComponent インターフェース

```
public interface XBRLTaxonomyComponent
extends XBRLTaxonomyNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書を構成する XML 文書（タクソノミー本体またはリンクベース）を表します。

フィールドの概要：

XBRLTaxonomyComponent インターフェースのフィールドの概要を次の表に示します。

表 6-102 XBRLTaxonomyComponent インターフェースのフィールドの概要

項番	説明	
1	static int	TAXONOMY_BODY_COMPONENT この XML 文書は，タクソノミー本体です。
2	static int	LINKBASE_COMPONENT この XML 文書は，リンクベースです。

メソッドの概要：

XBRLTaxonomyComponent インターフェースのメソッドの概要を次の表に示します。

表 6-103　XBRLTaxonomyComponent インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	int	getComponentType() XML 文書の種類を表すコードを取得します。	2.0 および 2.1
2	java.lang.String	getAbsoluteURI() XML 文書の絶対 URI を取得します。	2.0 および 2.1
3	org.w3c.dom.Document	getW3CDOMDocument() XML 文書の DOM ツリーを取得します。	2.0 および 2.1

（1） getComponentType

```
public int getComponentType()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML 文書の種類を表すコードを取得します。

戻り値：

この XML 文書の種類を表すコードを返します。戻り値のコードは,
XBRLTaxonomyComponent インターフェースのフィールド値のどちらかです。

（2） getAbsoluteURI

```
public java.lang.String getAbsoluteURI()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML 文書の絶対 URI を取得します。

戻り値：

この XML 文書の絶対 URI を返します。

（3） getW3CDOMDocument

```
public org.w3c.dom.Document getW3CDOMDocument()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML 文書の DOM ツリーを取得します。

戻り値：

この XML 文書の DOM ツリーを表す org.w3c.dom.Document を返します。

6.8.13 XBRLTaxonomyDocument インターフェース

```
public interface XBRLTaxonomyDocument
extends XBRLTaxonomyNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書全体を表します。

メソッドの概要：

XBRLTaxonomyDocument インターフェースのメソッドの概要を次の表に示します。

表 6-104 XBRLTaxonomyDocument インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	XBRLDocumentManager	getOwnerDocumentManager() タクソノミー文書を所有する XBRLDocumentManager オブジェクトを取得します。	2.0 および 2.1	該当しない
2	int	getXBRLVersion() タクソノミー文書が準拠する XBRL バージョンを取得します。	2.0 および 2.1	該当しない
3	PTVLinkbase	getPTVLinkbase() タクソノミー文書の解析時に生成された PTVLI を取得します。	2.1	該当しない
4	java.util.List<XBRLTaxonomyComponent>	getTaxonomyComponents() タクソノミー文書を構成する XML 文書のリストを取得します。	2.0 および 2.1	該当しない
5	java.util.List<XBRLTaxonomyBody>	getTaxonomyBodies() タクソノミー文書を構成するタクソノミー本体のリストを取得します。	2.0 および 2.1	該当しない
6	XBRLTaxonomyBody	getTaxonomyBody(java.lang.String targetNamespaceURI) タクソノミー文書を構成するタクソノミー本体のうち，指定した名前空間 URI を持つタクソノミー本体を取得します。	2.0	該当しない
7	java.util.List<XBRLLinkbase>	getLinkbases() タクソノミー文書を構成するリンクベースのリストを取得します。	2.0 および 2.1	該当しない
8	java.util.List<XBRLTaxonomyComponent>	getStartingPoints() タクソノミー文書を構成する際の出発点となったタクソノミー本体およびリンクベースのリストを取得します。	2.1	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
9	java.util.List<XBRLTaxonomyElement>	getElements() タクソノミー文書に記述されている要素の要素宣言のリストを取得します。	2.0 および 2.1	該当しない
10	XBRLTaxonomyElement	getElement(java.lang.String namespaceURI, java.lang.String name) 指定した名前の要素の要素宣言を取得します。	2.0 および 2.1	該当しない
11	java.util.List<XBRLRoleType>	getRoleTypes() タクソノミー文書に記述されている <link:roleType> 要素のリストを取得します。	2.1	該当しない
12	java.util.List<XBRLRoleType>	getRoleTypes(java.lang.String roleURI) タクソノミー文書に記述されている <link:roleType> 要素のうち，指定した値のカスタムロールを定義している <link:roleType> 要素のリストを取得します。	2.1	該当しない
13	java.util.List<XBRLArcroleType>	getArcroleTypes() タクソノミー文書に記述されている <link:arcroleType> 要素のリストを取得します。	2.1	該当しない
14	java.util.List<XBRLArcroleType>	getArcroleTypes(java.lang.String arcroleURI) タクソノミー文書に記述されている <link:arcroleType> 要素のうち，指定した値のカスタムアークロールを定義している <link:arcroleType> 要素のリストを取得します。	2.1	該当しない
15	java.util.List<XBRLLabel>	getLabels() タクソノミー文書に記述されているラベルのリストを取得します。	2.0 および 2.1	該当しない
16	java.util.List<XBRLLabel>	getLabelsByLang(java.lang.String lang) タクソノミー文書に記述されているラベルのうち，指定した言語のラベルのリストを取得します。	2.0 および 2.1	該当しない
17	java.util.List<XBRLLabel>	getLabelsByRole(java.lang.String role) タクソノミー文書に記述されているラベルのうち，指定したロールを持つラベルのリストを取得します。	2.0 および 2.1	該当しない
18	java.util.List<XBRLLabel>	getLabels(java.lang.String lang, java.lang.String role) タクソノミー文書に記述されているラベルのうち，指定した言語およびロールを持つラベルのリストを取得します。	2.0 および 2.1	該当しない
19	java.util.List<XBRLReference>	getReferences() タクソノミー文書に記述されている文献への参照のリストを取得します。	2.0 および 2.1	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
20	java.util.List<XBRLReference>	getReferences(java.lang.String role) タクソノミー文書に記述されている文献への参照のうち，指定したロールを持つ文献への参照のリストを取得します。	2.0 および 2.1	該当しない
21	XBRLTaxonomyElement[]	findCalculationRootElements(java.lang.String extendedLinkRole, java.lang.String arcrole) 計算リンクで定義された関係のネットワークでルートとなる要素宣言を検索します。	2.1	該当しない
22	XBRLTaxonomyElement[]	findDefinitionRootElements(java.lang.String extendedLinkRole, java.lang.String arcrole) 定義リンクで定義された関係のネットワークでルートとなる要素宣言を検索します。	2.1	該当しない
23	XBRLTaxonomyElement[]	findPresentationRootElements(java.lang.String extendedLinkRole, java.lang.String arcrole) プレゼンテーションリンクで定義された関係のネットワークでルートとなる要素宣言を検索します。	2.1	該当しない
24	java.util.List<XBRLNamespaceDeclaration>	getNamespaceDeclarations() タクソノミー文書に記述されている名前空間宣言のリストを取得します。	2.1	該当しない
25	java.util.List<XBRLTaxonomyLinkbaseRef>	getLinkbaseRefs() タクソノミー文書に存在する <link:linkbaseRef> 要素のリストを取得します。	2.1	該当しない
26	java.util.List<XBRLTaxonomyRoleRef>	getRoleRefs() タクソノミー文書に存在する <link:roleRef> 要素のリストを取得します。	2.1	該当しない
27	java.util.List<XBRLTaxonomyTypeDefinition>	getTaxonomyTypeDefinitions() タクソノミー文書で定義された型定義のリストを取得します。	2.1	該当しない
28	java.util.List<XBRLNamespaceDeclaration>	getNamespaceDeclarationsByPrefix(java.lang.String prefix) タクソノミー文書で定義される名前空間宣言のうち，引数で指定した名前空間接頭辞の名前空間宣言のリストを取得します。	2.1	該当しない
29	java.util.List<XBRLNamespaceDeclaration>	getNamespaceDeclarationsByURI(java.lang.String uri) タクソノミー文書で定義される名前空間宣言のうち，引数で指定した名前空間 URI の名前空間宣言のリストを取得します。	2.1	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
30	XBRLTaxonomyTypeDefinition	getTaxonomyTypeDefinition(java.lang.String namespaceURI, java.lang.String name) 引数で指定した名前の型定義を取得します。	2.1	該当しない
31	java.util.List<XBRLDimensionalRelationshipSet>	getDimensionalRelationshipSets() タクソノミー文書で定義される DRS のリストを取得します。	2.1	1.0
32	int	getDimensionsVersion() タクソノミー文書が準拠しているXBRL Dimensions のバージョンを取得します。	2.1	該当しない

(1) getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書を所有する XBRLDocumentManager オブジェクトを取得します。

戻り値：

このタクソノミー文書を所有する XBRLDocumentManager オブジェクトを返します。

(2) getXBRLVersion

```
public int getXBRLVersion()
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書が準拠する XBRL バージョンを取得します。

戻り値：

このタクソノミー文書が XBRL2.0 に準拠している場合は XBRLConstants.XBRL_2_0，XBRL2.1 に準拠している場合は XBRLConstants.XBRL_2_1 を返します。

(3) getPTVLinkbase

```
public PTVLinkbase getPTVLinkbase()
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書の解析時に生成された PTVLI を取得します。

タクソノミー文書を解析する際に完全な妥当性検証を実施した場合は，このメソッドによって PTVLI を取得できます。完全な妥当性検証が実施されていない場合は，PTVLI は生成されていないため，このメソッドでの PTVLI の取得はできません。

完全な妥当性検証および PTVLI の詳細については，「2.9　PTVLI の生成（XBRL2.1 の場合）」を参照してください。

このメソッドが返す PTVLI はタクソノミー文書の解析時に生成されたもので，その後は変更されません。そのため，タクソノミー文書を参照しているインスタンス文書に <link:linkbaseRef> 要素などを追加することでタクソノミー文書の内容を変更しても，その変更は PTVLI に反映されません。

戻り値：

タクソノミー文書の解析時に生成された PTVLI を表す PTVLinkbase オブジェクトを返します。PTVLI が生成されていない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

（4） getTaxonomyComponents

```
public java.util.List<XBRLTaxonomyComponent>
getTaxonomyComponents()
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書を構成する XML 文書のリストを取得します。タクソノミー文書を構成する XML 文書は，タクソノミー本体またはリンクベースです。

このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLTaxonomyComponent オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

このタクソノミー文書を構成する XML 文書を表す XBRLTaxonomyComponent オブジェクトのリストを返します。

（5） getTaxonomyBodies

```
public java.util.List<XBRLTaxonomyBody> getTaxonomyBodies()
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書を構成するタクソノミー本体のリストを取得します。
XBRL2.1 の場合：このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLTaxonomyBody オブジェクトに増減があった場合，その変更は直ちに反映されます。
このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

このタクソノミー文書を構成するタクソノミー本体を表す XBRLTaxonomyBody オブジェクトのリストを返します。

（6） getTaxonomyBody

```
public XBRLTaxonomyBody getTaxonomyBody(java.lang.String
targetNamespaceURI)
```

XBRL のバージョン：

2.0

説明：

タクソノミー文書を構成するタクソノミー本体のうち，指定した名前空間 URI を持つタクソノミー本体を取得します。

引数：

targetNamespaceURI - 対象名前空間 URI を指定します。

戻り値：

指定した対象名前空間 URI を持つタクソノミー本体を表す XBRLTaxonomyBody オブジェクトを返します。指定した対象名前空間 URI を持つタクソノミー本体がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.0 でない場合にスローされます。［エラーコード：20100］
- 引数 targetNamespaceURI に null を指定した場合にスローされます。［エラーコード：20002］

（7） getLinkbases

```
public java.util.List<XBRLLinkbase> getLinkbases()
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書を構成するリンクベースのリストを取得します。
このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLinkbase オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

このタクソノミー文書を構成するリンクベースを表す XBRLLinkbase オブジェクトのリストを返します。このタクソノミー文書にリンクベースがない場合は，要素数が 0 のリストを返します。

(8) getStartingPoints

```
public java.util.List<XBRLTaxonomyComponent> getStartingPoints()
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書を構成する際の出発点となったタクソノミー本体およびリンクベースのリストを取得します。
この XBRLTaxonomyDocument オブジェクトが XBRLTaxonomyDocumentBuilder オブジェクトによって生成されたものである場合，getStartingPoints() メソッドが返すリストには，XBRLTaxonomyDocumentBuilder オブジェクトの parse() メソッドの引数に指定したタクソノミー本体が含まれます。一方，この XBRLTaxonomyDocument オブジェクトが，XBRLInstanceDocumentBuilder オブジェクトを使用してインスタンス文書を解析することによって生成されたものである場合は，getStartingPoints() メソッドが返すリストには，インスタンス文書に記述されている <link:schemaRef> 要素，<link:linkbaseRef> 要素，<link:roleRef> 要素，および，<link:arcroleRef> 要素が指すタクソノミー本体およびリンクベースが含まれています。
このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLTaxonomyComponent オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このタクソノミー文書を構築する際の出発点となったタクソノミー本体およびリンクベースを表す XBRLTaxonomyComponent オブジェクトのリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

（9） getElements

```
public java.util.List<XBRLTaxonomyElement> getElements()
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書に記述されている要素の要素宣言のリストを取得します。
XBRL2.1 の場合：このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLTaxonomyElement オブジェクトに増減があった場合，その変更は直ちに反映されます。
このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

要素宣言を表す XBRLTaxonomyElement オブジェクトのリストを返します。要素宣言がない場合は，要素数が 0 のリストを返します。

（10） getElement

```
public XBRLTaxonomyElement getElement(java.lang.String
namespaceURI,
                                      java.lang.String name)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した名前の要素の要素宣言を取得します。

引数：

namespaceURI - 要素の名前空間 URI を指定します。
name - 要素の名前を指定します。

戻り値：

引数 namespaceURI で指定した名前空間 URI に属し，引数 name で指定した名前を持つ要素の要素宣言を表す XBRLTaxonomyElement オブジェクトを返します。
指定した要素の要素宣言がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 引数 namespaceURI または name に null を指定した場合にスローされます。［エラーコード：20002］

(11) getRoleTypes

```
public java.util.List<XBRLRoleType> getRoleTypes()
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書に記述されている <link:roleType> 要素のリストを取得します。このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLRoleType オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このタクソノミー文書に記述されている <link:roleType> 要素を表す XBRLRoleType オブジェクトのリストを返します。このタクソノミー文書に <link:roleType> 要素が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

(12) getRoleTypes

```
public java.util.List<XBRLRoleType> getRoleTypes(java.lang.String
roleURI)
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書に記述されている <link:roleType> 要素のうち，指定した値のカスタムロールを定義している <link:roleType> 要素のリストを取得します。このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLRoleType オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

roleURI - カスタムロールの値を指定します。

戻り値：

このタクソノミー文書に記述されている <link:roleType> 要素を表す XBRLRoleType オブジェクトのリストを返します。指定されたカスタムロールを定

義する <link:roleType> 要素が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 roleURI に null を指定した場合にスローされます。［エラーコード：20002］

（13）getArcroleTypes

```
public java.util.List<XBRLArcroleType> getArcroleTypes()
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書に記述されている <link:arcroleType> 要素のリストを取得します。このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLArcroleType オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このタクソノミー文書に記述されている <link:arcroleType> 要素を表す XBRLArcroleType オブジェクトのリストを返します。このタクソノミー文書に <link:arcroleType> 要素が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

（14）getArcroleTypes

```
public java.util.List<XBRLArcroleType>
getArcroleTypes(java.lang.String arcroleURI)
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書に記述されている <link:arcroleType> 要素のうち，指定した値のカスタムアークロールを定義している <link:arcroleType> 要素のリストを取得します。

このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLArcroleType オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返す

リストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

arcroleURI - カスタムアークロールの値を指定します。

戻り値：

このタクソノミー文書に記述されている <link:arcroleType> 要素を表す XBRLArcroleType オブジェクトのリストを返します。指定されたカスタムアークロールを定義する <link:arcroleType> 要素が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 arcroleURI に null を指定した場合にスローされます。［エラーコード：20002］

(15) getLabels

```
public java.util.List<XBRLLabel> getLabels()
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書に記述されているラベルのリストを取得します。リストの各要素は，XBRLLabel へキャストできます。

XBRLTaxonomyDocument インターフェースの getLabels() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。

しかし，XBRLInstanceElement および XBRLTaxonomyElement インターフェースの getLabels() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。

このメソッドが返す XBRLLabel オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabel オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

このタクソノミー文書で定義されるラベルを表す XBRLLabel オブジェクトのリストを返します。このタクソノミー文書で定義されるラベルがない場合は，要素数が

0 のリストを返します。

(16) getLabelsByLang

```
public java.util.List<XBRLLabel> getLabelsByLang(java.lang.String
lang)
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書に記述されているラベルのうち，指定した言語のラベルのリストを取得します。

XBRLTaxonomyDocument インターフェースの getLabelsByLang() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。

しかし，XBRLInstanceElement および XBRLTaxonomyElement インターフェースの getLabelsByLang() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。

このメソッドが返す XBRLLabel オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabel オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

lang - ラベルの言語を指定します。

戻り値：

このタクソノミー文書で定義されるラベルを表す XBRLLabel オブジェクトのリストを返します。指定した言語のラベルがない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(17) getLabelsByRole

```
public java.util.List<XBRLLabel> getLabelsByRole(java.lang.String
role)
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書に記述されているラベルのうち，指定したロールを持つラベルの

リストを取得します。
XBRLTaxonomyDocument インターフェースの getLabelsByRole() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。
しかし，XBRLInstanceElement および XBRLTaxonomyElement インターフェースの getLabelsByRole() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。
このメソッドが返す XBRLLabel オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabel オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：
role - ラベルのロールを指定します。ロールを持たないラベルを取得する場合は null を指定します。

戻り値：
このタクソノミー文書で定義されるラベルを表す XBRLLabel オブジェクトのリストを返します。指定したロールを持つラベルがない場合は，要素数が 0 のリストを返します。

(18) getLabels

```
public java.util.List<XBRLLabel> getLabels(java.lang.String lang,
                                           java.lang.String role)
```

XBRL のバージョン：
2.0 および 2.1

説明：
タクソノミー文書に記述されているラベルのうち，指定した言語およびロールを持つラベルのリストを取得します。
XBRLTaxonomyDocument インターフェースの getLabels() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。
しかし，XBRLInstanceElement および XBRLTaxonomyElement インターフェースの getLabels() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。
このメソッドが返す XBRLLabel オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabel オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされ

ます。

引数：

lang - ラベルの言語を指定します。

role - ラベルのロールを指定します。ロールを持たないラベルを取得する場合は null を指定します。

戻り値：

このタクソノミー文書で定義されるラベルを表す XBRLLabel オブジェクトのリストを返します。指定した言語とロールを持つラベルがない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(19) getReferences

```
public java.util.List<XBRLReference> getReferences()
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書に記述されている文献への参照のリストを取得します。

XBRLTaxonomyDocument インターフェースの getReferences() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。

しかし，XBRLInstanceElement および XBRLTaxonomyElement インターフェースの getReferences() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。

このメソッドが返す XBRLReference オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLReference オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

文献への参照を表す XBRLReference オブジェクトのリストを返します。文献への参照がない場合は，要素数が 0 のリストを返します。

(20) getReferences

```
public java.util.List<XBRLReference>
getReferences(java.lang.String role)
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書に記述されている文献への参照のうち，指定したロールを持つ文献への参照のリストを取得します。

XBRLTaxonomyDocument インターフェースの getReferences() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。

しかし，XBRLInstanceElement および XBRLTaxonomyElement インターフェースの getReferences() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。

このメソッドが返す XBRLReference オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLReference オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

role - 文献への参照のロールを指定します。ロールを持たない文献への参照を取得する場合は null を指定します。

戻り値：

このタクソノミー文書で定義される文献への参照を表す XBRLReference オブジェクトのリストを返します。指定したロールを持つ文献への参照がない場合は，要素数が 0 のリストを返します。

(21) findCalculationRootElements

```
public XBRLTaxonomyElement[]
findCalculationRootElements(java.lang.String extendedLinkRole,
                                                    java.lang.String
arcrole)
```

XBRL のバージョン：

2.1

説明：

計算リンクで定義された関係のネットワークでルートとなる要素宣言を検索します。検索対象となる関係を定義しているアークは，引数 extendedLinkRole で指定されたロールを持つ計算リンクに記述され，かつ，引数 arcrole で指定されたアークロールを持つものとします。指定されたアークによって定義された関係がひとつも存在しない場合は，このメソッドは null を返します。

要素宣言が関係のネットワークでルートとなる条件は，「どんな関係でも to 側のリソースとして出現しない」ことです。つまり，そのネットワークで関係の to 側とならない要素宣言はすべてルートとなります。また，ルートとなる要素宣言は，関係

の from 側である必要はありません。関係のネットワークでほかのどの要素宣言とも関連づけられていない要素宣言は，そのネットワークでのルートです。このメソッドが返した配列に含まれる各要素宣言が関係の from 側となっているかどうかについては，このメソッドに指定したものと同じ引数を指定して XBRLTaxonomyElement.getCalculationRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを呼び出すことによって調べることができます。

引数：

extendedLinkRole - 関係を定義しているアークを含む計算リンクのロールを指定します。

arcrole - 関係を定義しているアークのアークロールを指定します。

戻り値：

指定されたアークによって定義された関係のネットワークでルートとなる要素宣言をすべて含む XBRLTaxonomyElement オブジェクトの配列を返します。指定されたアークによって定義された関係のネットワークでルートが存在しない場合は，要素数が 0 の配列を返します。指定されたアークによって定義された関係がひとつも存在しない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 extendedLinkRole または arcrole に null を指定した場合にスローされます。［エラーコード：20002］

（22）findDefinitionRootElements

```
public XBRLTaxonomyElement[]
findDefinitionRootElements(java.lang.String extendedLinkRole,
                                                    java.lang.String
arcrole)
```

XBRL のバージョン：

2.1

説明：

定義リンクで定義された関係のネットワークでルートとなる要素宣言を検索します。検索対象となる関係を定義しているアークは，引数 extendedLinkRole で指定されたロールを持つ定義リンクに記述され，かつ，引数 arcrole で指定されたアークロールを持つものとします。指定されたアークによって定義された関係がひとつも存在しない場合は，このメソッドは null を返します。

要素宣言が関係のネットワーク上でルートとなる条件は，「どんな関係にあっても to 側のリソースとして出現しない」ことです。つまり，そのネットワーク上で関係の to 側とならない要素宣言はすべてルートとなります。また，ルートとなる要素宣言

は，関係の from 側である必要はありません。関係のネットワーク上でほかのどの要素宣言とも関連づけられていない要素宣言は，そのネットワーク上でのルートです。このメソッドが返した配列に含まれる各要素宣言が関係の from 側となっているかどうかについては，このメソッドに指定したものと同じ引数を指定して XBRLTaxonomyElement.getDefinitionRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを呼び出すことによって調べることができます。

引数：

extendedLinkRole - 関係を定義しているアークを含む定義リンクのロールを指定します。

arcrole - 関係を定義しているアークのアークロールを指定します。

戻り値：

指定されたアークによって定義された関係のネットワーク上でルートとなる要素宣言をすべて含む XBRLTaxonomyElement オブジェクトの配列を返します。指定されたアークによって定義された関係のネットワーク上でルートが存在しない場合は，要素数が 0 の配列を返します。指定されたアークによって定義された関係がひとつも存在しない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 extendedLinkRole または arcrole に null を指定した場合にスローされます。［エラーコード：20002］

(23) findPresentationRootElements

```
public XBRLTaxonomyElement[]
findPresentationRootElements(java.lang.String extendedLinkRole,
                                                    java.lang.String
arcrole)
```

XBRL のバージョン：

2.1

説明：

プレゼンテーションリンクで定義された関係のネットワークでルートとなる要素宣言を検索します。

検索対象となる関係を定義しているアークは，引数 extendedLinkRole で指定されたロールを持つプレゼンテーションリンクに記述され，かつ，引数 arcrole で指定されたアークロールを持つものとします。指定されたアークによって定義された関係がひとつも存在しない場合は，このメソッドは null を返します。

要素宣言が関係のネットワーク上でルートとなる条件は，「どんな関係にあっても to 側のリソースとして出現しない」ことです。つまり，そのネットワーク上で関係の

to 側とならない要素宣言はすべてルートとなります。また，ルートとなる要素宣言は，関係の from 側である必要はありません。関係のネットワーク上でほかのどの要素宣言とも関連づけられていない要素宣言は，そのネットワーク上でのルートです。このメソッドが返した配列に含まれる各要素宣言が関係の from 側となっているかどうかについては，このメソッドに指定したものと同じ引数を指定して XBRLTaxonomyElement.getPresentationRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを呼び出すことによって調べることができます。

引数：

extendedLinkRole - 関係を定義しているアークを含むプレゼンテーションリンクのロールを指定します。
arcrole - 関係を定義しているアークのアークロールを指定します。

戻り値：

指定されたアークによって定義された関係のネットワーク上でルートとなる要素宣言をすべて含む XBRLTaxonomyElement オブジェクトの配列を返します。指定されたアークによって定義された関係のネットワーク上でルートが存在しない場合は，要素数が 0 の配列を返します。指定されたアークによって定義された関係がひとつも存在しない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 extendedLinkRole または arcrole に null を指定した場合にスローされます。［エラーコード：20002］

（24）getNamespaceDeclarations

```
public java.util.List<XBRLNamespaceDeclaration>
getNamespaceDeclarations()
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書に記述されている名前空間宣言のリストを取得します。リストの各要素は，XBRLNamespaceDeclaration へキャストすることができます。
このメソッドが返すリストには，<xsd:schema> 要素で指定された名前空間宣言，または，<link:linkbase> 要素で指定された名前空間宣言のみが含まれます。ただし，タクソノミー本体に記述される <link:linkbase> 要素は取得対象になりません。
<xsd:include> 要素でインクルードされたタクソノミー本体が重複する場合，重複するタクソノミー本体に対し名前空間宣言を重複して取得することはありません。
また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリス

トを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

名前空間宣言を表す XBRLNamespaceDeclaration オブジェクトのリストを返します。このタクソノミー文書に名前空間宣言の名前空間 URI が存在しない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］

(25) getLinkbaseRefs

```
public java.util.List<XBRLTaxonomyLinkbaseRef> getLinkbaseRefs()
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書に存在する <link:linkbaseRef> 要素のリストを取得します。リストの各要素は，XBRLTaxonomyLinkbaseRef へキャストすることができます。
また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このタクソノミー文書に存在する <link:linkbaseRef> 要素をすべて含む XBRLTaxonomyLinkbaseRef オブジェクトのリストを返します。このタクソノミー文書に <link:linkbaseRef> 要素が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］

(26) getRoleRefs

```
public java.util.List<XBRLTaxonomyRoleRef> getRoleRefs()
```

XBRL のバージョン：

2.1

説明：

このタクソノミー文書に存在する <link:roleRef> 要素のリストを取得します。リストの各要素は，XBRLTaxonomyRoleRef へキャストすることができます。

また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

このタクソノミー文書に存在する <link:roleRef> 要素をすべて含む XBRLTaxonomyRoleRef オブジェクトのリストを返します。このタクソノミー文書に <link:roleRef> 要素が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］

（27）getTaxonomyTypeDefinitions

```
public java.util.List<XBRLTaxonomyTypeDefinition>
getTaxonomyTypeDefinitions()
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書で定義された型定義のリストを取得します。リストの各要素は，XBRLTaxonomyTypeDefinition へキャストすることができます。

このメソッドが返すリストには，名前を持たない型定義は含まれません。名前を持たない型定義は，その型定義を含む要素宣言の XBRLTaxonomyElement インターフェースから取得することができます。

また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

型定義を表す XBRLTaxonomyTypeDefinition オブジェクトのリストを返します。このタクソノミー文書に型定義（名前を持たない型定義を除く）が存在しない場合は，要素数が 0 であるリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］

(28) getNamespaceDeclarationsByPrefix

```
public java.util.List<XBRLNamespaceDeclaration>
getNamespaceDeclarationsByPrefix(java.lang.String prefix)
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書で定義される名前空間宣言のうち，引数で指定した名前空間接頭辞の名前空間宣言のリストを取得します。リストの各要素は，XBRLNamespaceDeclaration へキャストすることができます。

また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

prefix - 名前空間接頭辞を指定します。デフォルトの名前空間の宣言を取得する場合は null を指定します。

戻り値：

このタクソノミー文書で定義される引数 prefix で指定された名前空間接頭辞の名前空間宣言を表す XBRLNamespaceDeclaration オブジェクトのリストを返します。指定された名前空間接頭辞の名前空間宣言が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］

(29) getNamespaceDeclarationsByURI

```
public java.util.List<XBRLNamespaceDeclaration>
getNamespaceDeclarationsByURI (java.lang.String uri)
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書で定義される名前空間宣言のうち，引数で指定した名前空間 URI

の名前空間宣言のリストを取得します。リストの各要素は，
XBRLNamespaceDeclaration へキャストすることができます。
また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：
uri - 名前空間 URI を指定します。

戻り値：
このタクソノミー文書で定義される引数 uri で指定された名前空間 URI の名前空間宣言を表す XBRLNamespaceDeclaration オブジェクトのリストを返します。指定された名前空間 URI の名前空間宣言が存在しない場合は，要素数が 0 のリストを返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］
- 引数 uri に null を指定した場合にスローされます。［エラーコード：20002］

（30）getTaxonomyTypeDefinition

```
public XBRLTaxonomyTypeDefinition
getTaxonomyTypeDefinition(java.lang.String
namespaceURI,java.lang.String name)
```

XBRL のバージョン：
2.1

説明：
タクソノミー文書で定義された型定義のうち，引数で指定した名前の型定義を取得します。

引数：
namespaceURI - 型定義の名前空間 URI を指定します。
name - 型定義の名前を指定します。

戻り値：
引数 namespaceURI で指定された名前空間 URI に属し，引数 name で指定された名前を持つ型定義を表す XBRLTaxonomyTypeDefinition オブジェクトを返します。指定された型定義が存在しない場合は null を返します。

例外：
XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］
- 引数 namespaceURI または name に null を指定した場合にスローされます。［エラーコード：20002］

(31) getDimensionalRelationshipSets

```
public java.util.List<XBRLDimensionalRelationshipSet>
getDimensionalRelationshipSets()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

タクソノミー文書で定義される DRS のリストを取得します。
このメソッドが返す XBRLDimensionalRelationshipSet オブジェクトのリストは，ライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる必要がある XBRLDimensionalRelationshipSet オブジェクトに増減があった場合，その変更は直ちに反映されます。
このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

このタクソノミー文書で定義される DRS のリストを返します。このタクソノミー文書で定義される DRS がない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- このタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- このタクソノミー文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］

(32) getDimensionsVersion

```
public int getDimensionsVersion()
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書が準拠している XBRL Dimensions のバージョンを取得します。

cbrp.parse.DimensionsVersion プロパティの値は，このタクソノミー文書が XBRL2.1 に準拠している場合にだけ使用されます。XBRL のバージョンが 2.1 以外の場合，cbrp.parse.DimensionsVersion プロパティの値は使用されません。

戻り値：

このタクソノミー文書が XBRL Dimensions1.0 に準拠している場合は XBRLConstants.DIMENSIONS_1_0 を返します。XBRL Dimensions に準拠していない場合は XBRLConstants.DIMENSIONS_NONE を返します。

6.8.14 XBRLTaxonomyDocumentBuilder インターフェース

```
public interface XBRLTaxonomyDocumentBuilder
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書から XBRLTaxonomyDocument オブジェクトを取得するための API を定義します。

メソッドの概要：

XBRLTaxonomyDocumentBuilder インターフェースのメソッドの概要を次の表に示します。

表 6-105 XBRLTaxonomyDocumentBuilder インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLDocumentManager	getOwnerDocumentManager() XBRLTaxonomyDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。	2.0 および 2.1
2	int	getXBRLVersion() XBRLTaxonomyDocumentBuilder オブジェクトが解析するタクソノミー文書の XBRL バージョンを取得します。	2.0 および 2.1
3	boolean	isValidating() XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時に文書の妥当性を検証するかどうかを調べます。	2.0 および 2.1
4	boolean	isDelayLoading() XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時にリンクベースの遅延読み込みをするかどうかを調べます。	2.0 および 2.1
5	int	getMaxTaxonomyExtension() タクソノミー文書の解析時に読み込むタクソノミー文書の最大階層数を取得します。	2.0

項番	説明		XBRL のバージョン
6	int	getMaxDocuments() タクソノミー文書の解析時に読み込むタクソノミー文書の最大数を取得します。	2.1
7	java.lang.Object	getProperty(java.lang.String name) XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用するプロパティの値を取得します。	2.1
8	void	setEntityResolver(org.xml.sax.EntityResolver entityResolver) タクソノミー文書に記述されているエンティティを解決するために使用するエンティティリゾルバ（org.xml.sax.EntityResolver オブジェクト）を設定します。	2.0 および 2.1
9	void	setErrorHandler(XBRLParseErrorHandler errorHandler) タクソノミー文書解析時のエラーを受け取る XBRLParseErrorHandler オブジェクトを設定します。	2.0 および 2.1
10	XBRLTaxonomyDocument	parse(org.xml.sax.InputSource[] is) 指定された入力ソースからタクソノミー文書を解析し，XBRLTaxonomyDocument オブジェクトを新規に生成します。	2.0 および 2.1
11	XBRLTaxonomyDocument	parse(java.io.File[] file) 指定されたファイルをタクソノミー文書として解析し，XBRLTaxonomyDocument オブジェクトを新規に生成します。	2.0 および 2.1
12	XBRLTaxonomyDocument	parse(java.lang.String[] uri) 指定された URI の内容をタクソノミー文書として解析し，XBRLTaxonomyDocument オブジェクトを新規に生成します。	2.0 および 2.1
13	int	getDimensionsVersion() XBRLTaxonomyDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているタクソノミー文書を解析するのかを取得します。	2.1

（1） getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLTaxonomyDocumentBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

戻り値：

この XBRLTaxonomyDocumentBuilder オブジェクトを所有する

XBRLDocumentManager オブジェクトを返します。

(2) getXBRLVersion

```
public int getXBRLVersion()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLTaxonomyDocumentBuilder オブジェクトが解析するタクソノミー文書の XBRL バージョンを取得します。このメソッドが返す値は，この XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.XBRLVersion プロパティの値です。
この XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.XBRLVersion プロパティの値は，この XBRLTaxonomyDocumentBuilder オブジェクトを生成した XBRLTaxonomyDocumentBuilderFactory オブジェクトの setXBRLVersion() メソッド，または setProperty() メソッドで設定された値です。
XBRLTaxonomyDocumentBuilderFactory オブジェクトで値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。

戻り値：

XBRL2.0 に準拠したタクソノミー文書を解析する場合は XBRLConstants.XBRL_2_0，XBRL2.1 に準拠したタクソノミー文書を解析する場合は XBRLConstants.XBRL_2_1 を返します。

(3) isValidating

```
public boolean isValidating()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時に文書の妥当性を検証するかどうかを調べます。このメソッドが返す値は，この XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.validate プロパティの値です。
この XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.validate プロパティの値は，この XBRLTaxonomyDocumentBuilder オブジェクトを生成した XBRLTaxonomyDocumentBuilderFactory オブジェクトの setValidating() メソッド，または setProperty() メソッドで設定された値です。
XBRLTaxonomyDocumentBuilderFactory オブジェクトで値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値について

は，「6.8.15(16)　setProperty」の説明を参照してください。

戻り値：

タクソノミー文書の解析時に妥当性を検証する場合は true，検証しない場合は false を返します。

(4) isDelayLoading

```
public boolean isDelayLoading()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時にリンクベースの遅延読み込みをするかどうかを調べます。このメソッドが返す値は，この XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.delayLoad プロパティの値です。

この XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.delayLoad プロパティの値は，この XBRLTaxonomyDocumentBuilder オブジェクトを生成した XBRLTaxonomyDocumentBuilderFactory オブジェクトの setDelayLoading() メソッド，または setProperty() メソッドで設定された値です。XBRLTaxonomyDocumentBuilderFactory オブジェクトで値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。

戻り値：

タクソノミー文書に含まれるリンクの解析を遅延する場合は true，遅延しない場合は false を返します。

(5) getMaxTaxonomyExtension

```
public int getMaxTaxonomyExtension()
```

XBRL のバージョン：

2.0

説明：

この XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書の解析時に読み込むタクソノミー文書の最大階層数を取得します。このメソッドが返す値は，この XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.maxTaxonomyExtension プロパティの値です。

この XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.maxTaxonomyExtension プロパティの値は，この XBRLTaxonomyDocumentBuilder オブジェクトを生成した XBRLTaxonomyDocumentBuilderFactory オブジェクトの

setMaxTaxonomyExtension() メソッド，または setProperty() メソッドで設定された値です。XBRLTaxonomyDocumentBuilderFactory オブジェクトで値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。
cbrp.parse.maxTaxonomyExtension プロパティの値は，解析対象のタクソノミー文書が XBRL2.0 に準拠している場合だけ使用されます。解析対象のタクソノミー文書がほかのバージョンの XBRL 仕様に準拠しているものである場合は，cbrp.parse.maxTaxonomyExtension プロパティの値は使用されません。

戻り値：
タクソノミー文書の解析時に処理できる拡張されたタクソノミー文書から基底タクソノミー文書までの最大階層を返します。

(6) getMaxDocuments

```
public int getMaxDocuments()
```

XBRL のバージョン：
2.1

説明：
タクソノミー文書の解析時に読み込むタクソノミー文書の最大数を取得します。このメソッドが返す値は，この XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.DTS.maxDocuments プロパティの値です。
この XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.DTS.maxDocuments プロパティの値は，この XBRLTaxonomyDocumentBuilder オブジェクトを生成した XBRLTaxonomyDocumentBuilderFactory オブジェクトの setMaxDocuments() メソッド，または setProperty() メソッドで設定された値です。
XBRLTaxonomyDocumentBuilderFactory オブジェクトで値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。
cbrp.parse.DTS.maxDocuments プロパティの値は，解析対象のタクソノミー文書が XBRL2.1 に準拠している場合だけ使用されます。解析対象のタクソノミー文書がほかのバージョンの XBRL 仕様に準拠しているものである場合は，cbrp.parse.DTS.maxDocuments プロパティの値は使用されません。

戻り値：
タクソノミー文書の解析時に処理できる文書の最大数を返します。

(7) getProperty

```
public java.lang.Object getProperty(java.lang.String name)
```

XBRL のバージョン：

2.1

説明：

XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用するプロパティの値を取得します。このメソッドで値を取得できるプロパティの一覧およびこのメソッドがプロパティの値として返すオブジェクトの詳細については，「4.16.2　プロパティ一覧」を参照してください。

引数：

name - プロパティの名前を指定します。

戻り値：

プロパティの値を返します。

例外：

XBRLDOMRuntimeException -

- 引数 name に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 name に指定したプロパティの名前が不正な場合にスローされます。［エラーコード：20150］

（8） setEntityResolver

```
public void setEntityResolver(org.xml.sax.EntityResolver
entityResolver)
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書に記述されているエンティティを解決するために使用するエンティティリゾルバ（org.xml.sax.EntityResolver オブジェクト）を設定します。org.xml.sax.EntityResolver インターフェースの resolveEntity() メソッドの戻り値として返す org.xml.sax.InputSource オブジェクトには，java.io.InputStream や java.io.Reader を設定できます。このとき，必ずシステム識別子も設定してください。システム識別子を設定しないで，java.io.InputStream や java.io.Reader だけを設定した場合，XML プロセッサがタクソノミー本体を処理する際に，<xsd:import> 要素や <xsd:include> 要素の schemaLocation 属性に指定した URI を解決できないためエラーが発生します。このため，org.xml.sax.EntityResolver インターフェースを実装する際には，org.xml.sax.InputSource オブジェクトに必ずシステム識別子を設定しなければなりません。なお，JAXP の仕様では，org.xml.sax.InputSource オブジェクトに設定するシステム識別子は，絶対 URI でなければなりません。

引数：

entityResolver - 解析するタクソノミー文書に存在するエンティティを解決するために使用される org.xml.sax.EntityResolver オブジェクトを指定します。null を指定すると，デフォルトの実装が使用されます。

(9) setErrorHandler

```
public void setErrorHandler(XBRLParseErrorHandler errorHandler)
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書解析時のエラーを受け取る XBRLParseErrorHandler オブジェクトを設定します。

引数 errorHandler に null を指定した場合は，XBRLParseErrorHandler インターフェースのデフォルトの実装が使用されます。デフォルトの実装では，warning(), error() の両メソッドが呼び出された場合は何もしないため，解析処理が続行されます。また，fatalError() メソッドが呼び出された場合は，fatalError() メソッドの引数に指定した例外をスローし，解析処理が中断されます。

引数：

errorHandler - 解析するタクソノミー文書に存在するエラーのレポートを受け取る XBRLParseErrorHandler オブジェクトを指定します。null を指定すると，デフォルトの実装が使用されます。

(10) parse

```
public XBRLTaxonomyDocument parse(org.xml.sax.InputSource[] is)
                            throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定された入力ソースからタクソノミー文書を解析し，XBRLTaxonomyDocument オブジェクトを新規に生成します。

引数 is に指定した配列の各要素の org.xml.sax.InputSource オブジェクトには，必ずシステム識別子を設定しなければなりません。システム識別子が設定されていない場合は，例外がスローされます。

引数：

is - 入力ソースを表す org.xml.sax.InputSource オブジェクトの配列を指定します。

戻り値：

新しく生成された XBRLTaxonomyDocument オブジェクトを返します。

例外：

XBRLDOMException -

- タクソノミー文書の解析処理でエラーが検出された場合にスローされます。[エラーコード：エラー要因によって異なる※]

  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 引数 is に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 is に指定した配列の要素数が 0 の場合にスローされます。［エラーコード：20013］
- 引数 is に指定した配列に値が null である要素が存在する場合にスローされます。［エラーコード：20012］
- 引数 is に指定した配列にシステム識別子が設定されていない org.xml.sax.InputSource オブジェクトが存在する場合にスローされます。［エラーコード：20016］
- 引数 is に指定した配列中の org.xml.sax.InputSource オブジェクトに設定されているシステム識別子の形式が不正な場合にスローされます。［エラーコード：20148］

（11）parse

```
public XBRLTaxonomyDocument parse(java.io.File[] file)
                            throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定されたファイルをタクソノミー文書として解析し，XBRLTaxonomyDocument オブジェクトを新規に生成します。

引数：

file - タクソノミー文書を含むファイルを表す java.io.File オブジェクトの配列を指定します。

戻り値：

新しく生成された XBRLTaxonomyDocument オブジェクトを返します。

例外：

XBRLDOMException -

- タクソノミー文書の解析処理でエラーが検出された場合にスローされます。［エラーコード：エラー要因によって異なる※］
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 引数 file に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 file に指定した配列の要素数が 0 の場合にスローされます。［エラーコード：20013］
- 引数 file に指定した配列に値が null である要素が存在する場合にスローされます。［エラーコード：20012］

(12) parse

```
public XBRLTaxonomyDocument parse(java.lang.String[] uri)
                            throws XBRLDOMException
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定された URI の内容をタクソノミー文書として解析し，
XBRLTaxonomyDocument オブジェクトを新規に生成します。

引数：

uri - タクソノミー文書の場所を示す URI を表す文字列の配列を指定します。

戻り値：

新しく生成された XBRLTaxonomyDocument オブジェクトを返します。

例外：

XBRLDOMException -

- タクソノミー文書の解析処理でエラーが検出された場合にスローされます。[エラーコード：エラー要因によって異なる※]
  注※　エラー要因については，「7.　メッセージ」の各メッセージの中で，このエラーコードをメッセージ番号とするメッセージ ID を参照してください。

XBRLDOMRuntimeException -

- 引数 uri に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 uri に指定した配列の要素数が 0 の場合にスローされます。[エラーコード：20013]
- 引数 uri に指定した配列に値が null である要素が存在する場合にスローされます。[エラーコード：20012]
- 引数 uri に指定した配列中の URI の形式が不正な場合にスローされます。[エラーコード：20148]

(13) getDimensionsVersion

```
public int getDimensionsVersion()
```

XBRL のバージョン：

2.1

説明：

XBRLTaxonomyDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているタクソノミー文書を解析するのかを取得します。このメソッドが返す値は，XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.DimensionsVersion プロパティの値です。

XBRLTaxonomyDocumentBuilder オブジェクトが使用する

cbrp.parse.DimensionsVersion プロパティの値は，
XBRLTaxonomyDocumentBuilder オブジェクトを生成した
XBRLTaxonomyDocumentBuilderFactory オブジェクトの
setDimensionsVersion(int version) メソッド，または，
setProperty(java.lang.String name, java.lang.Object value) メソッドで設定された値です。
XBRLTaxonomyDocumentBuilderFactory オブジェクトで値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。
cbrp.parse.DimensionsVersion プロパティの値は，このタクソノミー文書が XBRL2.1 に準拠している場合にだけ使用されます。XBRL のバージョンが 2.1 以外の場合，cbrp.parse.DimensionsVersion プロパティの値は使用されません。

戻り値：
XBRL Dimensions1.0 に準拠したタクソノミー文書を解析する場合は XBRLConstants.DIMENSIONS_1_0 を返します。XBRL Dimensions に準拠したタクソノミー文書を解析しない場合は XBRLConstants.DIMENSIONS_NONE を返します。

6.8.15　XBRLTaxonomyDocumentBuilderFactory インターフェース

```
public interface XBRLTaxonomyDocumentBuilderFactory
```

XBRL のバージョン：
2.0 および 2.1

説明：
XBRLTaxonomyDocumentBuilder オブジェクトを生成するための API を定義します。

メソッドの概要：
XBRLTaxonomyDocumentBuilderFactory インターフェースのメソッドの概要を次の表に示します。

表 6-106　XBRLTaxonomyDocumentBuilderFactory インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLDocumentManager	getOwnerDocumentManager() XBRLTaxonomyDocumentBuilderFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。	2.0 および 2.1

項番	説明		XBRL のバージョン
2	void	setXBRLVersion(int version) どの XBRL バージョンのタクソノミー文書を解析するのかを設定します。	2.0 および 2.1
3	int	getXBRLVersion() どの XBRL バージョンのタクソノミー文書を解析するのかを取得します。	2.0 および 2.1
4	void	setValidating(boolean flag) タクソノミー文書の解析時に文書の妥当性を検証するかどうかを設定します。	2.0 および 2.1
5	boolean	isValidating() タクソノミー文書の解析時に文書の妥当性を検証するかどうかを調べます。	2.0 および 2.1
6	void	setDelayLoading(boolean flag) タクソノミー文書の解析時にリンクベースの遅延読み込みをするかどうかを設定します。	2.0 および 2.1
7	boolean	isDelayLoading() タクソノミー文書の解析時にリンクベースの遅延読み込みをするかどうかを調べます。	2.0 および 2.1
8	void	setMaxTaxonomyExtension(int max) タクソノミー文書の解析時に読み込むタクソノミー文書の最大階層数を設定します。	2.0
9	int	getMaxTaxonomyExtension() タクソノミー文書の解析時に読み込むタクソノミー文書の最大階層数を取得します。	2.0
10	void	setMaxDocuments(int max) タクソノミー文書の解析時に読み込むタクソノミー文書の最大数を設定します。	2.1
11	int	getMaxDocuments() タクソノミー文書の解析時に読み込むタクソノミー文書の最大数を取得します。	2.1
12	void	setIgnoringComments(boolean flag) タクソノミー文書の解析時にコメントを無視するかどうかを設定します。	2.0 および 2.1
13	boolean	isIgnoringComments() タクソノミー文書の解析時にコメントを無視するかどうかを調べます。	2.0 および 2.1
14	void	setCoalescing(boolean flag) タクソノミー文書の解析時に CDATA ノードを Text ノードに変換するかどうかを設定します。	2.0 および 2.1
15	boolean	isCoalescing() タクソノミー文書の解析時に CDATA ノードを Text ノードに変換するかどうかを調べます。	2.0 および 2.1

項番	説明		XBRL のバージョン
16	void	setProperty(java.lang.String name, java.lang.Object value) XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用するプロパティの値を設定します。	2.1
17	java.lang.Object	getProperty(java.lang.String name) XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用するプロパティの値を取得します。	2.1
18	XBRLTaxonomyDocumentBuilder	newTaxonomyDocumentBuilder() 現在設定されているパラメタを使用して，XBRLTaxonomyDocumentBuilder のインスタンスを新規に生成します。	2.0 および 2.1
19	void	setDimensionsVersion(int version) XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているタクソノミー文書を解析するのかを指定します。	2.1
20	int	getDimensionsVersion() XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているタクソノミー文書を解析するのかを取得します。	2.1

（1） getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLTaxonomyDocumentBuilderFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

戻り値：

この XBRLTaxonomyDocumentBuilderFactory オブジェクトを所有する，XBRLDocumentManager オブジェクトを返します。

（2） setXBRLVersion

```
public void setXBRLVersion(int version)
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，どの XBRL バージョンのタクソノミー文書を解析するのかを設定します。このメソッドで指定した値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトによって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.XBRLVersion プロパティの値として設定されます。
cbrp.parse.XBRLVersion プロパティの値は，このメソッドか，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.XBRLVersion プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16) setProperty」の説明を参照してください。
cbrp.parse.XBRLVersion プロパティの値を設定する場合は，newTaxonomyDocumentBuilder() メソッドを呼び出す前にこのメソッドを呼び出してください。

引数：

version - XBRL2.0 に準拠したタクソノミー文書を解析する場合は XBRLConstants.XBRL_2_0，XBRL2.1 に準拠したタクソノミー文書を解析する場合は XBRLConstants.XBRL_2_1 を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 version に不正な値を指定した場合にスローされます。［エラーコード：20014］

（3） getXBRLVersion

```
public int getXBRLVersion()
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，どの XBRL バージョンのタクソノミー文書を解析するのかを取得します。このメソッドが返す値は，XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.XBRLVersion プロパティの値です。
この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.XBRLVersion プロパティの値は，この

XBRLTaxonomyDocumentBuilderFactory オブジェクトの setXBRLVersion() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.XBRLVersion プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16) setProperty」の説明を参照してください。

戻り値：

XBRL2.0 に準拠したタクソノミー文書を解析する場合は XBRLConstants.XBRL_2_0，XBRL2.1 に準拠したタクソノミー文書を解析する場合は XBRLConstants.XBRL_2_1 を返します。

(4) setValidating

```
public void setValidating(boolean flag)
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時に文書の妥当性を検証するかどうかを設定します。このメソッドで指定した値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトによって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.validate プロパティの値として設定されます。

cbrp.parse.validate プロパティの値は，このメソッドか，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.validate プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。

cbrp.parse.validate プロパティの値を設定する場合は，newTaxonomyDocumentBuilder() メソッドを呼び出す前にこのメソッドを呼び出してください。

引数：

flag - タクソノミー文書の解析時に妥当性を検証する場合は true，検証しない場合は false を指定します。

(5) isValidating

```
public boolean isValidating()
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時に文書の妥当性を検証するかどうかを調べます。
このメソッドが返す値は，XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.validate プロパティの値です。
この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.validate プロパティの値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトの setValidating() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.validate プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。

戻り値：
タクソノミー文書の解析時に妥当性を検証する場合は true，検証しない場合は false を返します。

(6) setDelayLoading

```
public void setDelayLoading(boolean flag)
```

XBRL のバージョン：
2.0 および 2.1

説明：
この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時にリンクベースの遅延読み込みをするかどうかを設定します。このメソッドで指定した値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトによって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.delayLoad プロパティの値として設定されます。
cbrp.parse.delayLoad プロパティの値は，このメソッドか，または setProperty() メソッドで設定します。これらのメソッドによって cbrp.parse.delayLoad プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。
cbrp.parse.delayLoad プロパティの値を設定する場合は，newTaxonomyDocumentBuilder() メソッドを呼び出す前に，このメソッドを呼び出してください。
タクソノミー文書の解析時に妥当性を検証する場合は，リンクの解析を遅延することはできません。このメソッドでリンクの解析を遅延するように設定しても，その設定は無視されます。リンクの解析を遅延したい場合は，必ず妥当性の検証を行わないように設定します。

引数：

flag - タクソノミー文書に含まれるリンクの解析を遅延する場合は true，遅延しない場合は false を指定します。

(7) isDelayLoading

```
public boolean isDelayLoading()
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時にリンクベースの遅延読み込みをするかどうかを調べます。
このメソッドが返す値は，XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.delayLoad プロパティの値です。
この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.delayLoad プロパティの値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトの setDelayLoading() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.delayLoad プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。

戻り値：

タクソノミー文書に含まれるリンクの解析を遅延する場合は true，遅延しない場合は false を返します。

(8) setMaxTaxonomyExtension

```
public void setMaxTaxonomyExtension(int max)
```

XBRL のバージョン：

2.0

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時に読み込むタクソノミー文書の，基底タクソノミー文書までの最大階層数を設定します。このメソッドで指定した値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトによって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.maxTaxonomyExtension プロパティの値として設定されます。
cbrp.parse.maxTaxonomyExtension プロパティの値は，このメソッドか，または setProperty() メソッドで設定できます。これらのメソッドによって

cbrp.parse.maxTaxonomyExtension プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。
cbrp.parse.maxTaxonomyExtension プロパティの値を設定する場合は，newTaxonomyDocumentBuilder() メソッドを呼び出す前にこのメソッドを呼び出してください。
cbrp.parse.maxTaxonomyExtension プロパティの値は，解析対象のタクソノミー文書が XBRL2.0 に準拠している場合だけ使用されます。解析対象のタクソノミー文書がほかのバージョンの XBRL 仕様に準拠しているものである場合は，cbrp.parse.maxTaxonomyExtension プロパティの値は使用されません。

引数：

max - タクソノミー文書の解析時に処理できる，拡張されたタクソノミー文書から基底タクソノミー文書までの最大階層を指定します。指定できる値の範囲は，0 以上 2,147,483,647（Integer.MAX_VALUE）以下です。

例外：

XBRLDOMRuntimeException -

- 引数 max に負数を指定した場合にスローされます。［エラーコード：20014］

（9）getMaxTaxonomyExtension

```
public int getMaxTaxonomyExtension()
```

XBRL のバージョン：

2.0

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時に読み込むタクソノミー文書の，基底タクソノミー文書までの最大階層数を取得します。
このメソッドが返す値は，XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.maxTaxonomyExtension プロパティの値です。
この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.maxTaxonomyExtension プロパティの値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトの setMaxTaxonomyExtension() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.maxTaxonomyExtension プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。

cbrp.parse.maxTaxonomyExtension プロパティの値は，解析対象のタクソノミー文書が XBRL2.0 に準拠している場合だけ使用されます。解析対象のタクソノミー文書がほかのバージョンの XBRL 仕様に準拠しているものである場合は，cbrp.parse.maxTaxonomyExtension プロパティの値は使用されません。

戻り値：

タクソノミー文書の解析時に処理できる，拡張されたタクソノミー文書から基底タクソノミー文書までの最大階層を返します。

(10) setMaxDocuments

```
public void setMaxDocuments(int max)
```

XBRL のバージョン：

2.1

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時に読み込むタクソノミー文書の最大数を設定します。このメソッドで指定した値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトによって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.DTS.maxDocuments プロパティの値として設定されます。
cbrp.parse.DTS.maxDocuments プロパティの値は，このメソッドか，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.DTS.maxDocuments プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。
cbrp.parse.DTS.maxDocuments プロパティの値を設定する場合は，newTaxonomyDocumentBuilder() メソッドを呼び出す前に，このメソッドを呼び出してください。
cbrp.parse.DTS.maxDocuments プロパティの値は，解析対象のタクソノミー文書が XBRL2.1 に準拠している場合だけ使用されます。解析対象のタクソノミー文書がほかのバージョンの XBRL 仕様に準拠しているものである場合は，cbrp.parse.DTS.maxDocuments プロパティの値は使用されません。

引数：

max - タクソノミー文書の解析時に処理できる文書の最大数を指定します。指定できる値の範囲は，0 以上 2,147,483,647（Integer.MAX_VALUE）以下です。

例外：

XBRLDOMRuntimeException -

- 引数 max に負数を指定した場合にスローされます。［エラーコード：20014］

(11) getMaxDocuments

```
public int getMaxDocuments()
```

XBRL のバージョン：

2.1

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時に読み込むタクソノミー文書の最大数を取得します。

このメソッドが返す値は，XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.DTS.maxDocuments プロパティの値です。

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.DTS.maxDocuments プロパティの値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトの setMaxDocuments() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.DTS.maxDocuments プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。

cbrp.parse.DTS.maxDocuments プロパティの値は，解析対象のタクソノミー文書が XBRL2.1 に準拠している場合だけ使用されます。解析対象のタクソノミー文書がほかのバージョンの XBRL 仕様に準拠しているものである場合は，cbrp.parse.DTS.maxDocuments プロパティの値は使用されません。

戻り値：

タクソノミー文書の解析時に処理できる文書の最大数を返します。

(12) setIgnoringComments

```
public void setIgnoringComments(boolean flag)
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時にコメントを無視するかどうかを設定します。このメソッドで指定した値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトによって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.ignoringComments プロパティの値として設定されます。

cbrp.parse.ignoringComments プロパティの値は，このメソッドか，または

setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.ignoringComments プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。
cbrp.parse.ignoringComments プロパティの値を設定する場合は，newTaxonomyDocumentBuilder() メソッドを呼び出す前にこのメソッドを呼び出してください。
cbrp.parse.ignoringComments プロパティでコメントを無視するように指定しても，無視しないように指定しても，XBRL DOM の動作に影響しません。ただし，XBRLTaxonomyNode.getW3CDOMNode() メソッドなどで取得した DOM ツリーを参照した場合に Comment ノードが含まれるかどうかについては，cbrp.parse.ignoringComments プロパティの指定によって違いが生じます。

引数：

flag - タクソノミー文書の解析時にコメントを無視する場合は true，コメントを無視しない場合は false を指定します。

(13) isIgnoringComments

```
public boolean isIgnoringComments()
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時にコメントを無視するかどうかを調べます。
このメソッドが返す値は，XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.ignoringComments プロパティの値です。
この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.ignoringComments プロパティの値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトの setIgnoringComments() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.ignoringComments プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。

戻り値：

タクソノミー文書の解析時にコメントを無視する場合は true，無視しない場合は false を返します。

(14) setCoalescing

```
public void setCoalescing(boolean flag)
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時に CDATA ノードを Text ノードに変換するかどうかを設定します。このメソッドで指定した値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトによって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.coalescing プロパティの値として設定されます。
cbrp.parse.coalescing プロパティの値は，このメソッドか，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.coalescing プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。
XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。
cbrp.parse.coalescing プロパティの値を設定する場合は，newTaxonomyDocumentBuilder() メソッドを呼び出す前に，このメソッドを呼び出してください。
cbrp.parse.coalescing プロパティで CDATA ノードを Text ノードに変換するように指定しても，変換しないように指定しても，XBRL DOM の動作に影響しません。ただし，XBRLTaxonomyNode.getW3CDOMNode() メソッドなどで取得した DOM ツリーを参照した場合に CDATA ノードが含まれるかどうかについては，cbrp.parse.coalescing プロパティの指定によって違いが生じます。

引数：

flag - タクソノミー文書の解析時に CDATA ノードを Text ノードに変換する場合は true，Text ノードに変換しない場合は false を指定します。

(15) isCoalescing

```
public boolean isCoalescing()
```

XBRL のバージョン：

2.0 および 2.1

説明：

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，タクソノミー文書の解析時に CDATA ノードを Text ノードに変換するかどうかを調べます。
このメソッドが返す値は，XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する cbrp.parse.coalescing プロパティの値です。

この XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.coalescing プロパティの値は，この XBRLTaxonomyDocumentBuilderFactory オブジェクトの setCoalescing() メソッド，または setProperty() メソッドで設定できます。これらのメソッドによって cbrp.parse.coalescing プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。

戻り値：

タクソノミー文書の解析時に CDATA ノードを Text ノードに変換する場合は true，変換しない場合は false を返します。

(16) setProperty

```
public void setProperty(java.lang.String name,
                        java.lang.Object value)
```

XBRL のバージョン：

2.1

説明：

XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用するプロパティの値を設定します。このメソッドで値を設定できるプロパティの一覧およびこのメソッドにプロパティの値として指定するオブジェクトの詳細については，「4.16.2　プロパティ一覧」を参照してください。

各プロパティの値は，このメソッドか，または setValidating() などの各プロパティ専用のメソッドで設定できます。これらのメソッドによってプロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時のプロパティの値は，引数 name で指定したプロパティに対し，初期化時に値が指定されていたかどうかで次のように異なります。

- 初期化時にプロパティの値が指定されていた場合
  そのときの指定値が XBRL DOM 初期化時の値となる。
- 初期化時にプロパティの値が指定されていなかった場合
  そのプロパティのデフォルト値が XBRL DOM 初期化時の値となる。

プロパティを設定する場合は，newTaxonomyDocumentBuilder() メソッドを呼び出す前にこのメソッドを呼び出してください。

このメソッドで設定したプロパティ値は，setValidating() などの各プロパティ専用のメソッドによって上書きできます。逆に，setValidating() などの各プロパティ専用のメソッドで設定した値をこのメソッドで上書きすることもできます。例えば，このメソッドで cbrp.parse.validate プロパティの値として Boolean.TRUE を設定したあと，setValidating() メソッドを呼び出して false を設定した場合，getProperty() メソッドで取得した cbrp.parse.validate プロパティの値は Boolean.FALSE となります。また，isValidating() メソッドも false を返します。

引数：

name - プロパティの名前を指定します。

value - プロパティの値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 name に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 name に指定したプロパティの名前が正しくない場合にスローされます。［エラーコード：20150］
- 引数 name に指定したプロパティが値として null を受け付けないで，かつ引数 value に null を指定した場合にスローされます。［エラーコード：20151］
- 引数 value に指定したオブジェクトの型が正しくない場合にスローされます。［エラーコード：20152］
- 引数 value に指定したオブジェクトが保持している値が正しくない場合にスローされます。［エラーコード：20153］
- 引数 value に指定したオブジェクトが配列で，かつ要素数が 0 である場合にスローされます。［エラーコード：20154］
- 引数 value に指定したオブジェクトが配列で，かつ値が null である要素が存在する場合にスローされます。［エラーコード：20155］

（17）getProperty

```
public java.lang.Object getProperty(java.lang.String name)
```

XBRL のバージョン：

2.1

説明：

XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用するプロパティの値を取得します。このメソッドで値を取得できるプロパティの一覧およびこのメソッドがプロパティの値として返すオブジェクトの詳細については，「4.16.2　プロパティ一覧」を参照してください。

引数：

name - プロパティの名前を指定します。

戻り値：

プロパティの値を返します。

例外：

XBRLDOMRuntimeException -

- 引数 name に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 name に指定したプロパティの名前が正しくない場合にスローされます。［エラーコード：20150］

(18) newTaxonomyDocumentBuilder

```
public XBRLTaxonomyDocumentBuilder newTaxonomyDocumentBuilder()
```

XBRL のバージョン：

2.0 および 2.1

説明：

現在設定されているパラメタを使用して，XBRLTaxonomyDocumentBuilder のインスタンスを新規に生成します。

戻り値：

新しく生成された XBRLTaxonomyDocumentBuilder オブジェクトを返します。

(19) setDimensionsVersion

```
public void setDimensionsVersion(int version)
```

XBRL のバージョン：

2.1

説明：

XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているタクソノミー文書を解析するのかを指定します。このメソッドで指定した値は，XBRLTaxonomyDocumentBuilderFactory オブジェクトによって生成される XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する，cbrp.parse.DimensionsVersion プロパティの値として設定されます。

cbrp.parse.DimensionsVersion プロパティの値は，このメソッドか，または，setProperty(java.lang.String name, java.lang.Object value) メソッドで設定できます。これらのメソッドによって cbrp.parse.DimensionsVersion プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。

cbrp.parse.DimensionsVersion プロパティの値を設定する場合は，newTaxonomyDocumentBuilder() メソッドを呼び出す前にこのメソッドを呼び出してください。

cbrp.parse.DimensionsVersion プロパティの値は，解析対象のタクソノミー文書が XBRL2.1 に準拠している場合だけ使用されます。解析対象のタクソノミー文書がほかのバージョンの XBRL に準拠している場合は，cbrp.parse.DimensionsVersion プロパティの値は使用されません。

引数：

version - XBRL Dimensions1.0 に準拠したタクソノミー文書を解析する場合は XBRLConstants.DIMENSIONS_1_0 を指定します。XBRL Dimensions に準拠した

タクソノミー文書を解析しない場合は XBRLConstants.DIMENSIONS_NONE を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 version に不正な値を指定した場合にスローされます。［エラーコード：20014］

(20) getDimensionsVersion

```
public int getDimensionsVersion()
```

XBRL のバージョン：

2.1

説明：

XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが，どのバージョンの XBRL Dimensions に準拠しているタクソノミー文書を解析するのかを取得します。このメソッドが返す値は，XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書を解析する際に使用する，cbrp.parse.DimensionsVersion プロパティの値です。

XBRLTaxonomyDocumentBuilderFactory によって生成される XBRLTaxonomyDocumentBuilder オブジェクトが使用する cbrp.parse.DimensionsVersion プロパティの値は，XBRLTaxonomyDocumentBuilderFactory オブジェクトの setDimensionsVersion(int version) メソッド，または，setProperty(java.lang.String name, java.lang.Object value) メソッドで設定できます。これらのメソッドによって cbrp.parse.DimensionsVersion プロパティの値が設定されなかった場合は，XBRL DOM 初期化時の値を使用します。XBRL DOM 初期化時の値については，「6.8.15(16)　setProperty」の説明を参照してください。

cbrp.parse.DimensionsVersion プロパティの値は，解析対象のタクソノミー文書が XBRL2.1 に準拠している場合だけ使用されます。解析対象のタクソノミー文書がほかのバージョンの XBRL に準拠している場合は，cbrp.parse.DimensionsVersion プロパティの値は使用されません。

戻り値：

XBRL Dimensions1.0 に準拠したタクソノミー文書を解析する場合は XBRLConstants.DIMENSIONS_1_0 を返します。XBRL Dimensions に準拠したタクソノミー文書を解析しない場合は XBRLConstants.DIMENSIONS_NONE を返します。

6.8.16 XBRLTaxonomyElement インターフェース

```
public interface XBRLTaxonomyElement
extends XBRLTaxonomyNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書に記述される要素の要素宣言を表します。

フィールドの概要：

XBRLTaxonomyElement インターフェースのフィールドの概要を次の表に示します。

表 6-107 XBRLTaxonomyElement インターフェースのフィールドの概要

項番	説明	
1	static int	ITEM_ELEMENT このオブジェクトは，アイテムの要素宣言を表します。
2	static int	TUPLE_ELEMENT このオブジェクトは，タプルの要素宣言を表します。
3	static int	OTHER_ELEMENT このオブジェクトは，アイテムまたはタプル以外の要素の要素宣言を表します。
4	static int	HYPERCUBE_ITEM_ELEMENT このオブジェクトは，ハイパーキューブの要素宣言を表します。
5	static int	DIMENSION_ITEM_ELEMENT このオブジェクトは，ディメンジョンの要素宣言を表します。

メソッドの概要：

XBRLTaxonomyElement インターフェースのメソッドの概要を次の表に示します。

表 6-108 XBRLTaxonomyElement インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	int	getElementType() オブジェクトの型を表すコードを取得します。	2.0 および 2.1
2	java.lang.String	getNamespaceURI() 要素が属する名前空間 URI を取得します。	2.0 および 2.1
3	java.lang.String	getName() 要素の名前を取得します。	2.0 および 2.1
4	XBRLQName	getType() 要素の型を示す修飾名を取得します。	2.0 および 2.1

項番	説明		XBRL のバージョン
5	XBRLQName	getSubstitutionGroup() 要素のヘッド要素の修飾名を取得します。	2.0 および 2.1
6	boolean	isAbstract() 要素が抽象的かどうかを調べます。	2.0 および 2.1
7	boolean	isNillable() 要素の要素宣言の nillable 属性の値を取得します。	2.1
8	java.lang.String	getId() 要素の id 属性の値を取得します。	2.0 および 2.1
9	java.lang.String	getBalance() 要素宣言の xbrli:balance 属性を取得します。	2.0 および 2.1
10	java.util.List<XBRLLabel>	getLabels() 要素に関連づけられているラベルのリストを取得します。	2.0 および 2.1
11	java.util.List<XBRLLabel>	getLabelsByLang(java.lang.String lang) 要素に関連づけられているラベルのうち，指定した言語のラベルのリストを取得します。	2.0 および 2.1
12	java.util.List<XBRLLabel>	getLabelsByRole(java.lang.String role) 要素に関連づけられているラベルのうち，指定したロールを持つラベルのリストを取得します。	2.0 および 2.1
13	XBRLLabel	getLabel(java.lang.String lang, java.lang.String role) 要素に関連づけられているラベルのうち，指定した言語とロールを持つラベルを取得します。	2.0 および 2.1
14	java.util.List<XBRLLabel>	getLabels(java.lang.String lang, java.lang.String role) 要素に関連づけられているラベルのうち，指定した言語とロールを持つラベルのリストを取得します。	2.1
15	java.util.List<XBRLLabelRelationship>	getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) 要素の要素宣言とラベルの関係を表す XBRLLabelRelationship オブジェクトのリストを取得します。	2.1
16	java.util.List<XBRLReference>	getReferences() 要素に関連づけられている文献への参照のリストを取得します。	2.0 および 2.1
17	java.util.List<XBRLReference>	getReferences(java.lang.String role) 要素に関連づけられている文献への参照のうち，指定したロールを持つ文献への参照のリストを取得します。	2.0 および 2.1

項番	説明		XBRL のバージョン
18	java.util.List<XBRLReferenceRelationship>	getReferenceRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) 要素の要素宣言と文献への参照の関係を表す XBRLReferenceRelationship オブジェクトのリストを取得します。	2.1
19	java.util.List<XBRLCalculationRelationship>	getCalculationRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) 要素宣言とほかの要素宣言との関係を表す XBRLCalculationRelationship オブジェクトのリストを取得します。	2.1
20	java.util.List<XBRLDefinitionRelationship>	getDefinitionRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) 要素宣言とほかの要素宣言の関係を表す XBRLDefinitionRelationship オブジェクトのリストを取得します。	2.1
21	java.util.List<XBRLPresentationRelationship>	getPresentationRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) 要素宣言とほかの要素宣言の関係を表す XBRLPresentationRelationship オブジェクトのリストを取得します。	2.1
22	XBRLTaxonomyTypeDefinition	getTaxonomyTypeDefinition() 要素の型定義を表す XBRLTaxonomyTypeDefinition オブジェクトを取得します。	2.1

(1) getElementType

```
public int getElementType()
```

XBRL のバージョン：

2.0 および 2.1

説明：

オブジェクトの型を表すコードを取得します。

戻り値：

オブジェクトの型を表すコードを返します。戻り値のコードは，XBRLTaxonomyElement インターフェースのフィールド値のどれかです。

(2) getNamespaceURI

```
public java.lang.String getNamespaceURI()
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素が属する名前空間 URI を取得します。

戻り値：
この要素が属する名前空間 URI を返します。

(3) getName

```
public java.lang.String getName()
```

XBRL のバージョン：
2.0 および 2.1

説明：
要素の名前を取得します。

戻り値：
この要素の名前を返します。

(4) getType

```
public XBRLQName getType()
```

XBRL のバージョン：
2.0 および 2.1

説明：
要素の型を示す修飾名を取得します。

戻り値：
この要素の型を示す修飾名を返します。要素宣言に type 属性の指定がない場合は null を返します。

(5) getSubstitutionGroup

```
public XBRLQName getSubstitutionGroup()
```

XBRL のバージョン：
2.0 および 2.1

説明：
要素のヘッド要素の修飾名を取得します。

戻り値：
この要素のヘッド要素の修飾名を返します。要素宣言に substitutionGroup 属性の指定がない場合は null を返します。

(6) isAbstract

```
public boolean isAbstract()
```

XBRL のバージョン：
2.0 および 2.1

説明：

要素が抽象的かどうかを調べます。
この要素が抽象的かどうかは，要素宣言の abstract 属性の値によって決まります。abstract 属性の値が true の場合は抽象的です。値が false か，または abstract 属性の指定が省略された場合は抽象的ではありません。

戻り値：
この要素が抽象的の場合は true，抽象的でない場合は false を返します。

(7) isNillable

```
public boolean isNillable()
```

XBRL のバージョン：
2.1

説明：
要素の要素宣言の nillable 属性の値を取得します。

戻り値：
この要素宣言の nillable 属性の値を返します。nillable 属性が指定されていない場合は false を返します。

例外：
XBRLDOMRuntimeException -
- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

(8) getId

```
public java.lang.String getId()
```

XBRL のバージョン：
2.0 および 2.1

説明：
要素の id 属性の値を取得します。

戻り値：
この要素宣言の id 属性の値を表す文字列を返します。この要素宣言が id 属性を持たない場合は null を返します。

(9) getBalance

```
public java.lang.String getBalance()
```

XBRL のバージョン：
2.0 および 2.1

説明：
要素宣言の xbrli:balance 属性の値を取得します。

戻り値：

この要素宣言の xbrli:balance 属性値を返します。この要素宣言が xbrli:balance 属性を持たない場合は null を返します。

(10) getLabels

```
public java.util.List<XBRLLabel> getLabels()
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられているラベルのリストを取得します。
XBRLTaxonomyElement および XBRLInstanceElement インターフェースの getLabels() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。
しかし，XBRLTaxonomyDocument インターフェースの getLabels() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。
XBRL2.1 の場合：このメソッドでは，要素とラベルの関係を定義している <link:labelArc> 要素のアークロール，および，その <link:labelArc> 要素を含んでいるラベルリンクのロールについては考慮しません。特定の値のロールを持つラベルリンクに記述され，かつ，特定の値のアークロールを持つような <link:labelArc> 要素によって関連づけられたラベルを取得したい場合は，XBRLTaxonomyElement.getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。
このメソッドが返す XBRLLabel オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabel オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

この要素に関連づけられているラベルのリストを返します。ラベルがない場合は，要素数が 0 のリストを返します。

(11) getLabelsByLang

```
public java.util.List<XBRLLabel> getLabelsByLang(java.lang.String
lang)
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられているラベルのうち，指定した言語のラベルのリストを取得し

ます。
XBRLTaxonomyElement および XBRLInstanceElement インターフェースの getLabelsByLang() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。
しかし，XBRLTaxonomyDocument インターフェースの getLabelsByLang() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。
XBRL2.1 の場合：このメソッドでは，要素とラベルの関係を定義している <link:labelArc> 要素のアークロール，および，その <link:labelArc> 要素を含んでいるラベルリンクのロールについては考慮しません。特定の値のロールを持つラベルリンクに記述され，かつ，特定の値のアークロールを持つような <link:labelArc> 要素によって関連づけられたラベルを取得したい場合は，XBRLTaxonomyElement.getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。
このメソッドが返す XBRLLabel オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabel オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：
lang - ラベルの言語を指定します。

戻り値：
この要素のラベルを表す XBRLLabel オブジェクトのリストを返します。指定した言語のラベルがない場合は，要素数が 0 のリストを返します。

例外：
XBRLDOMRuntimeException -
- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(12) getLabelsByRole

```
public java.util.List<XBRLLabel> getLabelsByRole(java.lang.String
role)
```

XBRL のバージョン：
2.0 および 2.1

説明：
要素に関連づけられているラベルのうち，指定したロールを持つラベルのリストを取得します。
XBRLTaxonomyElement および XBRLInstanceElement インターフェースの getLabelsByRole() では，アーク要素の use 属性が prohibited の場合は，関連づけ

られていないと判断し，取得するラベルのリストには含まれません。
しかし，XBRLTaxonomyDocument インターフェースの getLabelsByRole() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。
XBRL2.1 の場合：このメソッドでは，要素とラベルの関係を定義している <link:labelArc> 要素のアークロール，および，その <link:labelArc> 要素を含んでいるラベルリンクのロールについては考慮しません。特定の値のロールを持つラベルリンクに記述され，かつ，特定の値のアークロールを持つような <link:labelArc> 要素によって関連づけられたラベルを取得したい場合は，XBRLTaxonomyElement.getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。
このメソッドが返す XBRLLabel オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabel オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

role - ラベルのロールを指定します。ロールを持たないラベルを取得する場合は null を指定します。

戻り値：

この要素のラベルを表す XBRLLabel オブジェクトのリストを返します。指定したロールを持つラベルがない場合は，要素数が 0 のリストを返します。

(13) getLabel

```
public XBRLLabel getLabel(java.lang.String lang,
                          java.lang.String role)
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられているラベルのうち，指定した言語とロールを持つラベルを取得します。
XBRL2.1 の場合：このメソッドでは，要素とラベルの関係を定義している <link:labelArc> 要素のアークロール，および，その <link:labelArc> 要素を含んでいるラベルリンクのロールについては考慮しません。特定の値のロールを持つラベルリンクに記述され，かつ，特定の値のアークロールを持つような <link:labelArc> 要素によって関連づけられたラベルを取得したい場合は，XBRLTaxonomyElement.getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。

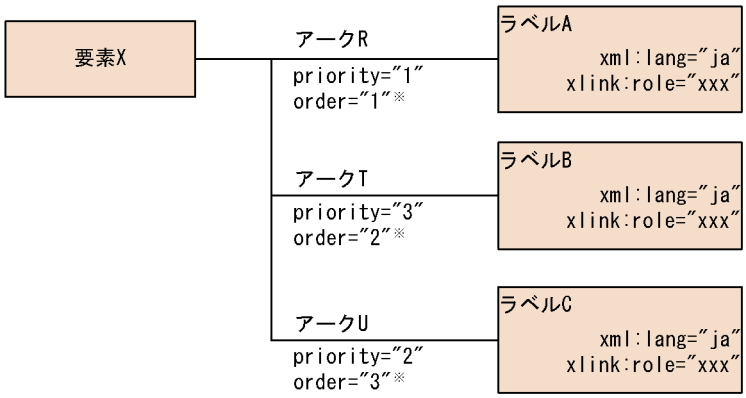
同じ言語とロールを持つ複数のラベルが関連づけられている場合の動作

一つの要素に対して同じ言語とロールを持つラベルが複数関連づけられていた場合の動作は，XBRL のバージョンによって異なります。次に，XBRL の各バージョンでの動作を示します。

- XBRL2.0 の場合
  一つの要素に対して同じ言語とロールを持つラベルが複数関連づけられている場合は，priority 属性の値が最も大きいアークによって関連づけられているラベルがこのメソッドによって取得できます。このとき，priority 属性の値が 2 番目以降のアークによって関連づけられたラベルは上書きされるため，XBRLTaxonomyElement.getLabels() などのほかのメソッドで取得できません。一つの要素に対して同じ言語とロールを持つラベルが複数関連づけられていて，かつ，それらの関連づけを定義しているアークのうち priority 属性の値が最も大きいアークが複数存在する場合は，タクソノミー文書の解析処理でエラーが発生します。このエラーを無視した場合にどちらのアークによって関連づけられたラベルが取得できるかについては，不定です。
- XBRL2.1 の場合
  一つの要素に対して同じ言語とロールを持つラベルが複数関連づけられている場合，order 属性の値が最も小さいアークによって関連づけられているラベルが，このメソッドによって取得できます。このとき，XBRL2.0 の場合とは異なり，各ラベルを関連づけているアークの priority 属性の値は影響しません。一つの要素に対して同じ言語とロールを持つラベルが複数関連づけられていて，かつ，それらの関連づけを定義しているアークのうち order 属性の値が最も小さいアークが複数存在する場合，どのアークによって関連づけられたラベルが取得できるかについては，不定です。この要素に対して同じ言語とロールを持つラベルが複数関連づけられている場合に，このメソッドでは取得できないほかのラベルを取得したい場合は，XBRLTaxonomyElement.getLabels(java.lang.String lang, java.lang.String role) メソッドを使用してください。また，このメソッドでは取得できないほかのラベルは，XBRLTaxonomyElement.getLabels() などのメソッドでも取得できます。

一つの要素に対して同じ言語とロールを持つラベルが複数関連づけられている場合の例を次の図に示します。

図 6-17　一つの要素に対して同じ言語とロールを持つラベルが複数関連づけられている場合の例



注※　アークにorder属性を指定できるのは，XBRL2.1の場合だけです。

この例では，要素 X に同じ言語とロールを持つラベル A，ラベル B，ラベル C が関連づけられています。この要素 X に対して，getLabel("ja",xxx) を呼び出した場合の結果を次に示します。
XBRL2.0 の場合：priority 属性の値が最も大きいアーク T によって関連づけられているラベル B が取得できます。しかし，getLabels() などのほかのメソッドを使用して，ラベル A およびラベル C を取得できません。
XBRL2.1 の場合：order 属性の値が最も小さいアーク R によって関連づけられているラベル A が取得できます。また，getLabels() などのほかのメソッドを使用して，ラベル B およびラベル C を取得することもできます。

引数：

lang - ラベルの言語を指定します。

role - ラベルのロールを指定します。ロールを持たないラベルを取得する場合は null を指定します。

戻り値：

この要素のラベルを表す XBRLLabel オブジェクトを返します。指定した言語とロールを持つラベルがない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(14) getLabels

```
public java.util.List<XBRLLabel> getLabels(java.lang.String lang,
                                           java.lang.String role)
```

XBRL のバージョン：

2.1

説明：

要素に関連づけられているラベルのうち，指定した言語とロールを持つラベルのリストを取得します。
このメソッドでは，要素とラベルの関係を定義している <link:labelArc> 要素のアークロール，および，その <link:labelArc> 要素を含んでいるラベルリンクのロールについては考慮しません。特定の値のロールを持つラベルリンクに記述され，かつ，特定の値のアークロールを持つような <link:labelArc> 要素によって関連づけられたラベルを取得したい場合は，
XBRLTaxonomyElement.getLabelRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。
このメソッドが返す XBRLLabel オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabel オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

lang - ラベルの言語を指定します。
role - ラベルのロールを指定します。ロールを持たないラベルを取得する場合は null を指定します。

戻り値：

この要素のラベルを表す XBRLLabel オブジェクトのリストを返します。指定された言語とロールを持つラベルが存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(15) getLabelRelationships

```
public java.util.List<XBRLLabelRelationship>
getLabelRelationships(java.lang.String extendedLinkRole,
                                          java.lang.String arcrole)
```

XBRL のバージョン：

2.1

説明：

要素の要素宣言とラベルの関係を表す XBRLLabelRelationship オブジェクトのリストを取得します。

この要素宣言に関連づけられたラベルは，このメソッドが返すリストに含まれる各 XBRLLabelRelationship オブジェクトに対して getTo() メソッドを呼び出すことによって取得できます。また，このメソッドが返すリストに含まれる各 XBRLLabelRelationship オブジェクトに対して getFrom() メソッドを呼び出した場合の戻り値は，この XBRLTaxonomyElement オブジェクト自身となります。
このメソッドが返すリストに含まれる XBRLLabelRelationship オブジェクトの並びは，各 XBRLLabelRelationship オブジェクトが表す関係を定義しているアークの order 属性の値によって昇順にソートされています。アークに order 属性が指定されていない場合は，1 が指定されたものと仮定します。アークの order 属性の値が同じである XBRLLabelRelationship オブジェクトが複数ある場合，それらの XBRLLabelRelationship オブジェクトの並び順は不定です。
このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLLabelRelationship オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

extendedLinkRole - 関係を定義しているアークを含むラベルリンクのロールを指定します。
arcrole - 関係を定義しているアークのアークロールを指定します。

戻り値：

引数 extendedLinkRole で指定されたロールを持つラベルリンクに属し，かつ，引数 arcrole で指定されたアークロールを持つアークによって定義された，この要素宣言とラベルの関係を表す XBRLLabelRelationship オブジェクトのリストを返します。指定された関係が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 extendedLinkRole または arcrole に null を指定した場合にスローされます。［エラーコード：20002］

（16）getReferences

```
public java.util.List<XBRLReference> getReferences()
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられている文献への参照のリストを取得します。
XBRLTaxonomyElement および XBRLInstanceElement インターフェースの

getReferences() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。
しかし，XBRLTaxonomyDocument インターフェースの getReferences() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。
XBRL2.1 の場合：このメソッドでは，要素と文献への参照の関係を定義している <link:referenceArc> 要素のアークロール，および，その <link:referenceArc> 要素を含んでいるリファレンスリンクのロールについては考慮しません。特定の値のロールを持つリファレンスリンクに記述され，かつ，特定の値のアークロールを持つような <link:referenceArc> 要素によって関連づけられた文献への参照を取得したい場合は，XBRLTaxonomyElement.getReferenceRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。
このメソッドが返す XBRLReference オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLReference オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

戻り値：

文献への参照を表す XBRLReference オブジェクトのリストを返します。文献への参照がない場合は，要素数が 0 のリストを返します。

(17) getReferences

```
public java.util.List<XBRLReference>
getReferences(java.lang.String role)
```

XBRL のバージョン：

2.0 および 2.1

説明：

要素に関連づけられている文献への参照のうち，指定したロールを持つ文献への参照のリストを取得します。
XBRLTaxonomyElement および XBRLInstanceElement インターフェースの getReferences() では，アーク要素の use 属性が prohibited の場合は，関連づけられていないと判断し，取得するラベルのリストには含まれません。
しかし，XBRLTaxonomyDocument インターフェースの getReferences() では，文書中に含まれる要素の一覧を取得するため，アーク要素の use 属性が prohibited の場合でも，取得するラベルのリストに含まれます。
XBRL2.1 の場合：このメソッドでは，要素と文献への参照の関係を定義している <link:referenceArc> 要素のアークロール，および，その <link:referenceArc> 要素を含んでいるリファレンスリンクのロールについては考慮しません。特定の値のロールを持つリファレンスリンクに記述され，かつ，特定の値のアークロールを持

つような <link:referenceArc> 要素によって関連づけられた文献への参照を取得したい場合は，XBRLTaxonomyElement.getReferenceRelationships(java.lang.String extendedLinkRole, java.lang.String arcrole) メソッドを使用してください。
このメソッドが返す XBRLReference オブジェクトのリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLReference オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると，java.lang.UnsupportedOperationException がスローされます。

引数：

role - 文献への参照のロールを指定します。ロールを持たない文献への参照を取得する場合は null を指定します。

戻り値：

この要素の文献への参照を表す XBRLReference オブジェクトのリストを返します。指定したロールを持つ文献への参照がない場合は，要素数が 0 のリストを返します。

(18) getReferenceRelationships

```
public java.util.List<XBRLReferenceRelationship>
getReferenceRelationships(java.lang.String extendedLinkRole,
                                            java.lang.String arcrole)
```

XBRL のバージョン：

2.1

説明：

要素の要素宣言と文献への参照の関係を表す XBRLReferenceRelationship オブジェクトのリストを取得します。
この要素宣言に関連づけられた文献への参照は，このメソッドが返すリストに含まれる各 XBRLReferenceRelationship オブジェクトに対して getTo() メソッドを呼び出すことによって取得できます。また，このメソッドが返すリストに含まれる各 XBRLReferenceRelationship オブジェクトに対して getFrom() メソッドを呼び出した場合の戻り値は，この XBRLTaxonomyElement オブジェクト自身となります。
このメソッドが返すリストに含まれる XBRLReferenceRelationship オブジェクトの並びは，各 XBRLReferenceRelationship オブジェクトが表す関係を定義しているアークの order 属性の値によって昇順にソートされています。アークに order 属性が指定されていない場合は，1 が指定されたものと仮定します。アークの order 属性の値が同じである XBRLReferenceRelationship オブジェクトが複数ある場合，それらの XBRLReferenceRelationship オブジェクトの並び順は不定です。
このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLReferenceRelationship オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようと

すると java.lang.UnsupportedOperationException がスローされます。

引数：

extendedLinkRole - 関係を定義しているアークを含むリファレンスリンクのロールを指定します。

arcrole - 関係を定義しているアークのアークロールを指定します。

戻り値：

引数 extendedLinkRole で指定されたロールを持つリファレンスリンクに属し，かつ，引数 arcrole で指定されたアークロールを持つアークによって定義された，この要素宣言とラベルの関係を表す XBRLReferenceRelationship オブジェクトのリストを返します。指定された関係が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 extendedLinkRole または arcrole に null を指定した場合にスローされます。［エラーコード：20002］

(19) getCalculationRelationships

```
public java.util.List<XBRLCalculationRelationship>
getCalculationRelationships(java.lang.String extendedLinkRole,
                                              java.lang.String arcrole)
```

XBRL のバージョン：

2.1

説明：

要素宣言とほかの要素宣言との関係を表す XBRLCalculationRelationship オブジェクトのリストを取得します。

この要素宣言に関連づけられたほかの要素宣言は，このメソッドが返すリストに含まれる各 XBRLCalculationRelationship オブジェクトに対して getTo() メソッドを呼び出すことによって取得できます。また，このメソッドが返すリストに含まれる各 XBRLCalculationRelationship オブジェクトに対して getFrom() メソッドを呼び出した場合の戻り値は，この XBRLTaxonomyElement オブジェクト自身となります。

このメソッドが返すリストに含まれる XBRLCalculationRelationship オブジェクトの並びは，各 XBRLCalculationRelationship オブジェクトが表す関係を定義しているアークの order 属性の値によって昇順にソートされています。アークに order 属性が指定されていない場合は，1 が指定されたものと仮定します。アークの order 属性の値が同じである XBRLCalculationRelationship オブジェクトが複数ある場合，それらの XBRLCalculationRelationship オブジェクトの並び順は不定です。

このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出した

あとに，このメソッドが返したリストに含まれる XBRLCalculationRelationship オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

extendedLinkRole - 関係を定義しているアークを含む計算リンクのロールを指定します。

arcrole - 関係を定義しているアークのアークロールを指定します。

戻り値：

引数 extendedLinkRole で指定されたロールを持つ計算リンクに属し，かつ，引数 arcrole で指定されたアークロールを持つアークによって定義された，この要素宣言とほかの要素宣言の関係を表す XBRLCalculationRelationship オブジェクトのリストを返します。指定された関係が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 extendedLinkRole または arcrole に null を指定した場合にスローされます。［エラーコード：20002］

（20）getDefinitionRelationships

```
public java.util.List<XBRLDefinitionRelationship>
getDefinitionRelationships(java.lang.String extendedLinkRole,
                                            java.lang.String arcrole)
```

XBRL のバージョン：

2.1

説明：

要素宣言とほかの要素宣言の関係を表す XBRLDefinitionRelationship オブジェクトのリストを取得します。

この要素宣言に関連づけられたほかの要素宣言は，このメソッドが返すリストに含まれる各 XBRLDefinitionRelationship オブジェクトに対して getTo() メソッドを呼び出すことによって取得できます。また，このメソッドが返すリストに含まれる各 XBRLDefinitionRelationship オブジェクトに対して getFrom() メソッドを呼び出した場合の戻り値は，この XBRLTaxonomyElement オブジェクト自身となります。

このメソッドが返すリストに含まれる XBRLDefinitionRelationship オブジェクトの並びは，各 XBRLDefinitionRelationship オブジェクトが表す関係を定義しているアークの order 属性の値によって昇順にソートされています。アークに order 属性が指定されていない場合は，1 が指定されたものと仮定します。アークの order 属

性の値が同じである XBRLDefinitionRelationship オブジェクトが複数ある場合，それらの XBRLDefinitionRelationship オブジェクトの並び順は不定です。
このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLDefinitionRelationship オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

extendedLinkRole - 関係を定義しているアークを含む定義リンクのロールを指定します。
arcrole - 関係を定義しているアークのアークロールを指定します。

戻り値：

引数 extendedLinkRole で指定されたロールを持つ定義リンクに属し，かつ，引数 arcrole で指定されたアークロールを持つアークによって定義された，この要素宣言とほかの要素宣言の関係を表す XBRLDefinitionRelationship オブジェクトのリストを返します。指定された関係が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 extendedLinkRole または arcrole に null を指定した場合にスローされます。［エラーコード：20002］

(21) getPresentationRelationships

```
public java.util.List<XBRLPresentationRelationship>
getPresentationRelationships(java.lang.String extendedLinkRole,
                                              java.lang.String arcrole)
```

XBRL のバージョン：

2.1

説明：

要素宣言とほかの要素宣言の関係を表す XBRLPresentationRelationship オブジェクトのリストを取得します。
この要素宣言に関連づけられたほかの要素宣言は，このメソッドが返すリストに含まれる各 XBRLPresentationRelationship オブジェクトに対して getTo() メソッドを呼び出すことによって取得できます。また，このメソッドが返すリストに含まれる各 XBRLPresentationRelationship オブジェクトに対して getFrom() メソッドを呼び出した場合の戻り値は，この XBRLTaxonomyElement オブジェクト自身となります。
このメソッドが返すリストに含まれる XBRLPresentationRelationship オブジェク

トの並びは，各 XBRLPresentationRelationship オブジェクトが表す関係を定義しているアークの order 属性の値によって昇順にソートされています。アークに order 属性が指定されていない場合は，1 が指定されたものと仮定します。アークの order 属性の値が同じである XBRLPresentationRelationship オブジェクトが複数ある場合，それらの XBRLPresentationRelationship オブジェクトの並び順は不定です。このメソッドが返すリストはライブオブジェクトです。このメソッドを呼び出したあとに，このメソッドが返したリストに含まれる XBRLPresentationRelationship オブジェクトに増減があった場合，その変更は直ちに反映されます。また，このメソッドが返すリストは読み取り専用です。このメソッドが返したリストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

引数：

extendedLinkRole - 関係を定義しているアークを含むプレゼンテーションリンクのロールを指定します。
arcrole - 関係を定義しているアークのアークロールを指定します。

戻り値：

引数 extendedLinkRole で指定されたロールを持つプレゼンテーションリンクに属し，かつ，引数 arcrole で指定されたアークロールを持つアークによって定義された，この要素宣言とほかの要素宣言の関係を表す XBRLPresentationRelationship オブジェクトのリストを返します。指定された関係が存在しない場合は，要素数が 0 のリストを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- 引数 extendedLinkRole または arcrole に null を指定した場合にスローされます。［エラーコード：20002］

(22) getTaxonomyTypeDefinition

```
public XBRLTaxonomyTypeDefinition getTaxonomyTypeDefinition()
```

XBRL のバージョン：

2.1

説明：

要素の型定義を表す XBRLTaxonomyTypeDefinition オブジェクトを取得します。この要素に名前を持たない型として定義された型定義か，type 属性により指定した型定義が，このタクソノミー文書に名前付きで定義された型定義である場合のみ取得することができます。

戻り値：

この要素の型定義を表す XBRLTaxonomyTypeDefinition オブジェクトを返します。

この要素に string のような事前定義型が指定されている場合に null が返ります。
type 属性，局所型定義のいずれも定義されていない場合にも null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- cbrp.parse.validate.frtafris オプションが false に設定されている場合にスローされます。［エラーコード：22011］

6.8.17 XBRLTaxonomyItem インターフェース

```
public interface XBRLTaxonomyItem
extends XBRLTaxonomyElement
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムの要素宣言を表します。
タクソノミー本体で <xsd:element> 要素によって記述される要素宣言のうち，substitutionGroup 属性が「xbrli:item」のものが，アイテムの要素宣言となります。

フィールドの概要：

XBRLTaxonomyItem インターフェースのフィールドの概要を次の表に示します。

表 6-109 XBRLTaxonomyItem インターフェースのフィールドの概要

項番	説明	
1	static int	MONETARY_ITEM このアイテムのデータ型は xbrli:monetary です。
2	static int	SHARES_ITEM このアイテムのデータ型は xbrli:shares です。
3	static int	PURE_ITEM このアイテムのデータ型は xbrli:pure です。
4	static int	FRACTION_ITEM このアイテムのデータ型は分数です。
5	static int	DECIMAL_ITEM このアイテムのデータ型は xsd:decimal です。
6	static int	FLOAT_ITEM このアイテムのデータ型は xsd:float です。
7	static int	DOUBLE_ITEM このアイテムのデータ型は xsd:double です。
8	static int	INTEGER_ITEM このアイテムのデータ型は xsd:integer です。

項番	説明	
9	static int	NONPOSITIVEINTEGER_ITEM このアイテムのデータ型は xsd:nonPositiveInteger です。
10	static int	NEGATIVEINTEGER_ITEM このアイテムのデータ型は xsd:negativeInteger です。
11	static int	LONG_ITEM このアイテムのデータ型は xsd:long です。
12	static int	INT_ITEM このアイテムのデータ型は xsd:int です。
13	static int	SHORT_ITEM このアイテムのデータ型は xsd:short です。
14	static int	BYTE_ITEM このアイテムのデータ型は xsd:byte です。
15	static int	NONNEGATIVEINTEGER_ITEM このアイテムのデータ型は xsd:nonNegativeInteger です。
16	static int	UNSIGNEDLONG_ITEM このアイテムのデータ型は xsd:unsignedLong です。
17	static int	UNSIGNEDINT_ITEM このアイテムのデータ型は xsd:unsignedInt です。
18	static int	UNSIGNEDSHORT_ITEM このアイテムのデータ型は xsd:unsignedShort です。
19	static int	UNSIGNEDBYTE_ITEM このアイテムのデータ型は xsd:unsignedByte です。
20	static int	POSITIVEINTEGER_ITEM このアイテムのデータ型は xsd:positiveInteger です。
21	static int	STRING_ITEM このアイテムのデータ型は xsd:string です。
22	static int	BOOLEAN_ITEM このアイテムのデータ型は xsd:boolean です。
23	static int	HEXBINARY_ITEM このアイテムのデータ型は xsd:hexBinary です。
24	static int	BASE64BINARY_ITEM このアイテムのデータ型は xsd:base64Binary です。
25	static int	ANYURI_ITEM このアイテムのデータ型は xsd:anyURI です。
26	static int	QNAME_ITEM このアイテムのデータ型は xsd:QName です。
27	static int	NOTATION_ITEM このアイテムのデータ型は xsd:NOTATION です。
28	static int	DURATION_ITEM このアイテムのデータ型は xsd:duration です。

項番	説明	
29	static int	DATETIME_ITEM このアイテムのデータ型は xbrli:dateUnion です。
30	static int	TIME_ITEM このアイテムのデータ型は xsd:time です。
31	static int	DATE_ITEM このアイテムのデータ型は xsd:date です。
32	static int	GYEARMONTH_ITEM このアイテムのデータ型は xsd:gYearMonth です。
33	static int	GYEAR_ITEM このアイテムのデータ型は xsd:gYear です。
34	static int	GMONTHDAY_ITEM このアイテムのデータ型は xsd:gMonthDay です。
35	static int	GDAY_ITEM このアイテムのデータ型は xsd:gDay です。
36	static int	GMONTH_ITEM このアイテムのデータ型は xsd:gMonth です。
37	static int	NORMALIZEDSTRING_ITEM このアイテムのデータ型は xsd:normalizedString です。
38	static int	TOKEN_ITEM このアイテムのデータ型は xsd:token です。
39	static int	LANGUAGE_ITEM このアイテムのデータ型は xsd:language です。
40	static int	NAME_ITEM このアイテムのデータ型は xsd:Name です。
41	static int	NCNAME_ITEM このアイテムのデータ型は xsd:NCName です。
42	static int	OTHER_ITEM このアイテムのデータ型は不明です。

メソッドの概要：

XBRLTaxonomyItem インターフェースのメソッドの概要を次の表に示します。

表 6-110　XBRLTaxonomyItem インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	int	getItemType() アイテムのデータ型を表すコードを取得します。	2.0 および 2.1	該当しない

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
2	int	getBaseItemType() アイテムのベースデータ型を表すコードを取得します。	2.1	該当しない
3	boolean	isNumeric() アイテムが数値または非数値のどちらであるかを調べます。	2.0 および 2.1	該当しない
4	java.lang.String	getPeriodType() 要素宣言の xbrli:periodType 属性の値を取得します。	2.1	該当しない
5	XBRLDimensionalRelationshipSet	getDimensionalRelationshipSet() アイテムに関連づいている DRS を取得します。	2.1	1.0

（1） getItemType

```
public int getItemType()
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムのデータ型を表すコードを取得します。

このメソッドを利用できるアイテムのデータ型は，XBRL のバージョンによって異なります。利用できるアイテムのデータ型と XBRL のバージョンの関係を次の表に示します。

表 6-111　利用できるアイテムのデータ型と XBRL のバージョンの関係

項番	アイテムのデータ型	XBRL2.0	XBRL2.1
1	MONETARY_ITEM	○	○
2	SHARES_ITEM	○	○
3	PURE_ITEM	×	○
4	FRACTION_ITEM	×	○
5	DECIMAL_ITEM	○	○
6	FLOAT_ITEM	○	○
7	DOUBLE_ITEM	○	○
8	INTEGER_ITEM	○	○
9	NONPOSITIVEINTEGER_ITEM	×	○
10	NEGATIVEINTEGER_ITEM	×	○
11	LONG_ITEM	×	○
12	INT_ITEM	×	○

項番	アイテムのデータ型	XBRL2.0	XBRL2.1
13	SHORT_ITEM	×	○
14	BYTE_ITEM	×	○
15	NONNEGATIVEINTEGER_ITEM	×	○
16	UNSIGNEDLONG_ITEM	×	○
17	UNSIGNEDINT_ITEM	×	○
18	UNSIGNEDSHORT_ITEM	×	○
19	UNSIGNEDBYTE_ITEM	×	○
20	POSITIVEINTEGER_ITEM	×	○
21	STRING_ITEM	○	○
22	BOOLEAN_ITEM	○	○
23	HEXBINARY_ITEM	○	○
24	BASE64BINARY_ITEM	○	○
25	ANYURI_ITEM	○	○
26	QNAME_ITEM	○	○
27	NOTATION_ITEM	○	×
28	DURATION_ITEM	○	○
29	DATETIME_ITEM	○	○
30	TIME_ITEM	○	○
31	DATE_ITEM	○	○
32	GYEARMONTH_ITEM	○	○
33	GYEAR_ITEM	○	○
34	GMONTHDAY_ITEM	○	○
35	GDAY_ITEM	○	○
36	GMONTH_ITEM	○	○
37	NORMALIZEDSTRING_ITEM	×	○
38	TOKEN_ITEM	×	○
39	LANGUAGE_ITEM	×	○
40	NAME_ITEM	×	○
41	NCNAME_ITEM	×	○
42	OTHER_ITEM	○	○

（凡例）○：利用できます。　×：利用できません。

戻り値：

このアイテムの型を表すコードを返します。

（2） getBaseItemType

```
public int getBaseItemType()
```

XBRL のバージョン：

2.1

説明：

アイテムのベースデータ型を表すコードを取得します。
ベースデータ型は，ユーザーデータ型の派生元となっているデータ型のことです。
アイテムがユーザーデータ型である場合（getItemType() メソッドが OTHER_ITEM を返す場合）は，getBaseItemType() メソッドを呼び出すことによって，派生元となっているデータ型を調べることができます。派生元のデータ型が不明な場合は，getBaseItemType() メソッドは OTHER_ITEM を返します。
このアイテムがユーザーデータ型でない場合（getItemType() メソッドが OTHER_ITEM 以外を返す場合）は，このメソッドは getItemType() メソッドが返す値と同じ値を返します。

戻り値：

このアイテムのベースデータ型を表すコードを返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

（3） isNumeric

```
public boolean isNumeric()
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムが数値または非数値のどちらであるかを調べます。
このアイテムが参照するコンテキストが数値か非数値のどちらかは，このアイテムのデータ型によって異なります。アイテムのデータ型と isNumeric() メソッドの戻り値との対応関係を次の表に示します。

表 6-112　アイテムのデータ型と isNumeric() メソッドの戻り値との対応関係

項番	アイテムのデータ型	isNumeric() メソッドの戻り値
1	MONETARY_ITEM	true
2	SHARES_ITEM	
3	PURE_ITEM	
4	FRACTION_ITEM	
5	DECIMAL_ITEM	

項番	アイテムのデータ型	isNumeric() メソッドの戻り値
6	FLOAT_ITEM	
7	DOUBLE_ITEM	
8	INTEGER_ITEM	
9	NONPOSITIVEINTEGER_ITEM	
10	NEGATIVEINTEGER_ITEM	
11	LONG_ITEM	
12	INT_ITEM	
13	SHORT_ITEM	
14	BYTE_ITEM	
15	NONNEGATIVEINTEGER_ITEM	
16	UNSIGNEDLONG_ITEM	
17	UNSIGNEDINT_ITEM	
18	UNSIGNEDSHORT_ITEM	
19	UNSIGNEDBYTE_ITEM	
20	POSITIVEINTEGER_ITEM	
21	STRING_ITEM	false
22	BOOLEAN_ITEM	
23	HEXBINARY_ITEM	
24	BASE64BINARY_ITEM	
25	ANYURI_ITEM	
26	QNAME_ITEM	
27	NOTATION_ITEM	
28	DURATION_ITEM	
29	DATETIME_ITEM	
30	TIME_ITEM	
31	DATE_ITEM	
32	GYEARMONTH_ITEM	
33	GYEAR_ITEM	
34	GMONTHDAY_ITEM	
35	GDAY_ITEM	
36	GMONTH_ITEM	
37	NORMALIZEDSTRING_ITEM	
38	TOKEN_ITEM	
39	LANGUAGE_ITEM	
40	NAME_ITEM	

項番	アイテムのデータ型	isNumeric() メソッドの戻り値
41	NCNAME_ITEM	
42	OTHER_ITEM	ユーザーデータ型が数値データ型から派生している場合は true，そうでない場合は false となります。ユーザーデータ型の詳細については，「5.5　取り扱えるデータ型」を参照してください。

戻り値：

このアイテムが数値コンテキストを参照する場合は true，非数値コンテキストを参照する場合は false を返します。

（4） getPeriodType

```
public java.lang.String getPeriodType()
```

XBRL のバージョン：

2.1

説明：

要素宣言の xbrli:periodType 属性の値を取得します。

戻り値：

この要素宣言の xbrli:periodType 属性の値を返します。xbrli:periodType 属性がない場合は null を返します。

例外：

XBRLDOMRuntimeException -

- 操作対象のタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］

（5） getDimensionalRelationshipSet

```
public XBRLDimensionalRelationshipSet
getDimensionalRelationshipSet()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

アイテムに関連づいている DRS を取得します。

戻り値：

このアイテムに関連づいている DRS を返します。このアイテムに関連づいている DRS がない場合は，null を返します。

例外：

XBRLDOMRuntimeException -

- このタクソノミー文書が準拠している XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20100］
- このタクソノミー文書が準拠している XBRL Dimensions のバージョンが 1.0 でない場合にスローされます。［エラーコード：22100］

6.8.18 XBRLTaxonomyNode インターフェース

```
public interface XBRLTaxonomyNode
```

XBRL のバージョン：

2.0 および 2.1

説明：

タクソノミー文書を構成するノードを表します。

フィールドの概要：

XBRLTaxonomyNode インターフェースのフィールドの概要を次の表に示します。

表 6-113 XBRLTaxonomyNode インターフェースのフィールドの概要

項番	説明	
1	static int	TAXONOMY_DOCUMENT_NODE このノードは，タクソノミー文書を表します。
2	static int	TAXONOMY_BODY_NODE このノードは，タクソノミー本体を表します。
3	static int	LINKBASE_NODE このノードは，リンクベースを表します。
4	static int	ITEM_NODE このノードは，アイテムの要素宣言を表します。
5	static int	TUPLE_NODE このノードは，タプルの要素宣言を表します。
6	static int	OTHER_ELEMENT_NODE このノードは，アイテムまたはタプル以外の要素の要素宣言を表します。
7	static int	ROLE_TYPE_NODE このノードは，<link:roleType> 要素を表します。
8	static int	ARCROLE_TYPE_NODE このノードは，<link:arcroleType> 要素を表します。
9	static int	LABEL_NODE このノードは，要素のラベルを表します。
10	static int	REFERENCE_NODE このノードは，文献への参照を表します。

項番	説明	
11	static int	PART_NODE このノードは，文献への参照に含まれるパートを表します。
12	static int	LINKBASE_REF_NODE このノードは，<link:linkbaseRef> 要素を表します。
13	static int	ROLE_REF_NODE このノードは，<link:roleRef> 要素を表します。
14	static int	TAXONOMY_ATTRIBUTE_DECLARATION_NODE このノードは，型定義の属性宣言を表します。
15	static int	TAXONOMY_COMPLEX_TYPE_DEFINITION_NODE このノードは，complexType の型定義を表します。
16	static int	TAXONOMY_SIMPLE_TYPE_DEFINITION_NODE このノードは，simpleType の型定義を表します。
17	static int	HYPERCUBE_ITEM_NODE このノードは，ハイパーキューブの要素宣言を表します。
18	static int	DIMENSION_ITEM_NODE このノードは，ディメンジョンの要素宣言を表します。

メソッドの概要：

XBRLTaxonomyNode インターフェースのメソッドの概要を次の表に示します。

表 6-114　XBRLTaxonomyNode インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	int	getNodeType() ノードの型を表すコードを取得します。	2.0 および 2.1
2	XBRLTaxonomyDocument	getOwnerTaxonomyDocument() ノードを所有する XBRLTaxonomyDocument オブジェクトを取得します。	2.0 および 2.1
3	XBRLTaxonomyNode	getParentNode() ノードが関連づけられている親ノードを取得します。	2.0 および 2.1
4	org.w3c.dom.Node	getW3CDOMNode() ノードに対応する org.w3c.dom.Node オブジェクトを取得します。	2.0 および 2.1

（1） getNodeType

```
public int getNodeType()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ノードの型を表すコードを取得します。

このメソッドが返す値は，XBRL のバージョンによって異なります。利用できるノードの型と XBRL のバージョンの関係を次の表に示します。

表 6-115　利用できるノードの型と XBRL のバージョンの関係

項番	ノードの型	XBRL2.0	XBRL2.1
1	TAXONOMY_DOCUMENT_NODE	○	○
2	TAXONOMY_BODY_NODE	○	○
3	LINKBASE_NODE	○	○
4	ITEM_NODE	○	○
5	TUPLE_NODE	○	○
6	OTHER_ELEMENT_NODE	○	○
7	ROLE_TYPE_NODE	×	○
8	ARCROLE_TYPE_NODE	×	○
9	LABEL_NODE	○	○
10	REFERENCE_NODE	○	○
11	PART_NODE	○	○
12	LINKBASE_REF_NODE	×	○
13	ROLE_REF_NODE	×	○
14	TAXONOMY_ATTRIBUTE_DECLARATION_NODE	×	○
15	TAXONOMY_COMPLEX_TYPE_DEFINITION_NODE	×	○
16	TAXONOMY_SIMPLE_TYPE_DEFINITION_NODE	×	○
17	HYPERCUBE_ITEM_NODE ※	×	○
18	DIMENSION_ITEM_NODE ※	×	○

（凡例）○：利用できます。　×：利用できません。
注※　XBRL Dimensions1.0 の場合だけ存在します。

戻り値：

このノードの型を表すコードを返します。

（2） getOwnerTaxonomyDocument

```
public XBRLTaxonomyDocument getOwnerTaxonomyDocument()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ノードを所有する XBRLTaxonomyDocument オブジェクトを取得します。

戻り値：

このノードを所有する XBRLTaxonomyDocument オブジェクトを返します。このノードが XBRLTaxonomyDocument の場合は，常に null を返します。

（3） getParentNode

```
public XBRLTaxonomyNode getParentNode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ノードが関連づけられている親ノードを取得します。このメソッドが返す親ノードの型は，このノードの型によって異なります。getParentNode メソッドが返す親ノードの型を次の表に示します。

表 6-116　getParentNode メソッドが返す親ノードの型

項番	このノードの型	このメソッドが返す親ノードの型
1	TAXONOMY_DOCUMENT_NODE	このメソッドは常に null を返します。
2	TAXONOMY_BODY_NODE	TAXONOMY_DOCUMENT_NODE
3	LINKBASE_NODE	TAXONOMY_DOCUMENT_NODE
4	ITEM_NODE	TAXONOMY_BODY_NODE
5	TUPLE_NODE	TAXONOMY_BODY_NODE
6	OTHER_ELEMENT_NODE	TAXONOMY_BODY_NODE
7	ROLE_TYPE_NODE	TAXONOMY_BODY_NODE
8	ARCROLE_TYPE_NODE	TAXONOMY_BODY_NODE
9	LABEL_NODE	TAXONOMY_BODY_NODE または LINKBASE_NODE
10	REFERENCE_NODE	TAXONOMY_BODY_NODE または LINKBASE_NODE
11	PART_NODE	REFERENCE_NODE
12	LINKBASE_REF_NODE	TAXONOMY_BODY_NODE
13	ROLE_REF_NODE	LINKBASE_NODE
14	TAXONOMY_ATTRIBUTE_DECLARATION_NODE	TAXONOMY_COMPLEX_TYPE_DEFINITION_NODE
15	TAXONOMY_COMPLEX_TYPE_DEFINITION_NODE	TAXONOMY_BODY_NODE または ITEM_NODE または TUPLE_NODE または OTHER_ELEMENT_NODE
16	TAXONOMY_SIMPLE_TYPE_DEFINITION_NODE	TAXONOMY_BODY_NODE または ITEM_NODE または TUPLE_NODE または OTHER_ELEMENT_NODE
17	HYPERCUBE_ITEM_NODE ※	TAXONOMY_BODY_NODE

項番	このノードの型	このメソッドが返す親ノードの型
18	DIMENSION_ITEM_NODE ※	TAXONOMY_BODY_NODE

注※　XBRL Dimensions1.0 の場合だけ存在します。

戻り値：

このノードの親ノードを返します。関連づけられた親ノードがない場合は null を返します。このノードが XBRLTaxonomyDocument の場合は，常に null を返します。

（4） getW3CDOMNode

```
public org.w3c.dom.Node getW3CDOMNode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

ノードに対応する org.w3c.dom.Node オブジェクトを取得します。XBRL の仕様で規定されていない属性や要素を参照したい場合は，このメソッドで取得した DOM ツリーを使用できます。

このメソッドが返す org.w3c.dom.Node オブジェクトの型は，このノードの型によって異なります。getW3CDOMNode() メソッドが返す org.w3c.dom.Node オブジェクトを，次の表に示します。

表 6-117　getW3CDOMNode() メソッドが返す org.w3c.dom.Node オブジェクト

項番	XBRLTaxonomyNode オブジェクトの型	org.w3c.dom.Node オブジェクトの型	説明
1	TAXONOMY_DOCUMENT_NODE	null	常に null を返します。
2	TAXONOMY_BODY_NODE	org.w3c.dom.Document	タクソノミー本体を表します。
3	LINKBASE_NODE	org.w3c.dom.Document	リンクベースを表します。
4	ITEM_NODE	org.w3c.dom.Element	アイテムを定義している <xsd:element> 要素を表します。
5	TUPLE_NODE	org.w3c.dom.Element	タプルを定義している <xsd:element> 要素を表します。
6	OTHER_ELEMENT_NODE	org.w3c.dom.Element	アイテムまたはタプル以外の要素を定義している <xsd:element> 要素を表します。

項番	XBRLTaxonomyNode オブジェクトの型	org.w3c.dom.Node オブジェクトの型	説明
7	ROLE_TYPE_NODE	org.w3c.dom.Element	<link:roleType> 要素を表します。
8	ARCROLE_TYPE_NODE	org.w3c.dom.Element	<link:arcroleType> 要素を表します。
9	LABEL_NODE	org.w3c.dom.Element	XBRLLabel.isRemote() が true の場合は <link:loc> 要素, false の場合は <link:label> 要素を表します。
10	REFERENCE_NODE	org.w3c.dom.Element	XBRLReference.isRemote() が true の場合は <link:loc> 要素, false の場合は <link:reference> 要素を表します。
11	PART_NODE	org.w3c.dom.Element	<link:part> 要素の代替要素を表します。
12	LINKBASE_REF_NODE	org.w3c.dom.Element	<link:linkbaseRef> 要素を表します。
13	ROLE_REF_NODE	org.w3c.dom.Element	<link:roleRef> 要素を表します。
14	TAXONOMY_ATTRIBUTE_DECLARATION_NODE	org.w3c.dom.Element	<xsd:attribute> 要素を表します。
15	TAXONOMY_COMPLEX_TYPE_DEFINITION_NODE	org.w3c.dom.Element	<xsd:complexType> 要素を表します。
16	TAXONOMY_SIMPLE_TYPE_DEFINITION_NODE	org.w3c.dom.Element	<xsd:simpleType> 要素を表します。
17	HYPERCUBE_ITEM_NODE ※	org.w3c.dom.Element	ハイパーキューブを定義している <xsd:element> 要素を表します。
18	DIMENSION_ITEM_NODE ※	org.w3c.dom.Element	ディメンジョンを定義している <xsd:element> 要素を表します。

注※　XBRL Dimensions1.0 の場合だけ存在します。

このメソッドが返した org.w3c.dom.Node オブジェクト，およびその org.w3c.dom.Node オブジェクトが属する DOM ツリーに対する更新操作は絶対にしてはいけません。更新操作をした場合には，XBRL DOM が正常に動作しないおそれがあります。

戻り値：

このノードに対応する org.w3c.dom.Node オブジェクトを返します。このノードの型が TAXONOMY_DOCUMENT_NODE の場合は null を返します。

6.8.19 XBRLTaxonomyOtherElement インターフェース

```
public interface XBRLTaxonomyOtherElement
extends XBRLTaxonomyElement
```

XBRL のバージョン：

2.0 および 2.1

説明：

アイテムまたはタプル以外の要素の要素宣言を表します。
このインターフェースは，タクソノミー本体に <xsd:element> 要素によって記述される要素宣言のうち，substitutionGroup 属性の値が xbrli:item もしくは xbrli:tuple でない，または，substitutionGroup 属性を持たない要素宣言を表します。

6.8.20 XBRLTaxonomySimpleTypeDefinition インターフェース

```
public interface XBRLTaxonomySimpleTypeDefinition
extends XBRLTaxonomyTypeDefinition
```

XBRL のバージョン：

2.1

説明：

simpleType の型定義を表します。

メソッドの概要：

このインターフェースは固有のメソッドを持ちません。

6.8.21 XBRLTaxonomyTuple インターフェース

```
public interface XBRLTaxonomyTuple
extends XBRLTaxonomyElement
```

XBRL のバージョン：

2.0 および 2.1

説明：

タプルの要素宣言を表します。
タクソノミー本体で <xsd:element> 要素によって記述される要素宣言のうち，substitutionGroup 属性が xbrli:tuple であるものが，タプルの要素宣言となります。

6.8.22 XBRLTaxonomyTypeDefinition インターフェース

```
public interface XBRLTaxonomyTypeDefinition
extends XBRLTaxonomyNode
```

XBRL のバージョン：

2.1

説明：

タクソノミー文書に記述される型定義を表します。

フィールドの概要：

XBRLTaxonomyTypeDefinition インターフェースのフィールドの概要を次の表に示します。

表 6-118 XBRL XBRLTaxonomyTypeDefinition インターフェースのフィールドの概要

項番	説明	
1	static int	COMPLEX_TYPE オブジェクトは，complexType の型定義を表します。
2	static int	SIMPLE_TYPE オブジェクトは，simpleType の型定義を表します。
3	static int	DERIVATION_EXTENSION 拡張による派生を示します。
4	static int	DERIVATION_LIST リストによる派生を示します。
5	static int	DERIVATION_NONE 派生がないことを示します。
6	static int	DERIVATION_RESTRICTION 制限による派生を示します。
7	static int	DERIVATION_SUBSTITUTION 置換を示します。
8	static int	DERIVATION_UNION 共用体による派生を示します。

メソッドの概要：

XBRLTaxonomyTypeDefinition インターフェースのメソッドの概要を次の表に示します

表 6-119 XBRLTaxonomyTypeDefinition インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	int	getTypeCategory() オブジェクトの型を表すコードを取得します。	2.1

項番	説明		XBRL のバージョン
2	java.lang.String	getId() 型定義の id 属性の値を取得します。	2.1
3	java.lang.String	getNamespaceURI() 型定義が属する名前空間 URI を取得します。	2.1
4	java.lang.String	getName() 型定義の name 属性の値を取得します。	2.1
5	int	getFinal() 型定義の final 属性の値を表すコードを取得します。	2.1
6	XBRLQName	getBase() 型定義の base 属性で示される型定義の修飾名を取得します。	2.1

(1) getTypeCategory

```
public int getTypeCategory()
```

XBRL のバージョン：

2.1

説明：

オブジェクトの型を表すコードを取得します。

戻り値：

このオブジェクトの型を表すコードを返します。この値は，<xsd:simpleType> 要素，<xsd:complexType> 要素のいずれかを表すコードを返します。

(2) getId

```
public java.lang.String getId()
```

XBRL のバージョン：

2.1

説明：

型定義の id 属性の値を取得します。

戻り値：

この型定義の id 属性の値を返します。この型定義に id 属性がない場合は null を返します。

(3) getNamespaceURI

```
public java.lang.String getNamespaceURI()
```

XBRL のバージョン：

2.1

説明：
型定義が属する名前空間 URI を取得します。

戻り値：
この型定義が属する名前空間 URI を返します。

(4) getName

```
public java.lang.String getName()
```

XBRL のバージョン：
2.1

説明：
型定義の name 属性の値を取得します。

戻り値：
この型定義の name 属性の値を返します。この型定義に name 属性がない場合は null を返します。

(5) getFinal

```
public int getFinal()
```

XBRL のバージョン：
2.1

説明：
型定義の final 属性の値を表すコードを取得します。

戻り値：
この型定義の final 属性の値を表すコードを返します。final 属性の指定がなく，<xsd:schema> 要素の finalDefault 属性の指定がある場合は，finalDefault 属性の値を表すコードを返します。
<xsd:simpleType> 要素では，DERIVATION_NONE，または DERIVATION_EXTENSION，DERIVATION_RESTRICTION，DERIVATION_LIST，DERIVATION_SUBSTITUTION，DERIVATION_UNION の組み合わせによる論理和です（単一の場合も含みます）。#all を指定した場合は，DERIVATION_EXTENSION，DERIVATION_RESTRICTION，DERIVATION_LIST，DERIVATION_SUBSTITUTION，DERIVATION_UNION の論理和です。
<xsd:complexType> 要素では，DERIVATION_NONE，または DERIVATION_EXTENSION，DERIVATION_RESTRICTION の組み合わせによる論理和です（単一の場合も含みます）。#all を指定した場合は，DERIVATION_EXTENSION，DERIVATION_RESTRICTION の論理和です。

(6) getBase

```
public XBRLQName getBase()
```

XBRL のバージョン：

2.1

説明：

型定義のベースとなる型定義の修飾名を取得します。

戻り値：

この型定義のベースとなる型定義の修飾名を返します。
<xsd:simpleType> 要素では，<xsd:restriction> 要素が存在しない場合，<xsd:anySimpleType> を表す修飾名を返します。<xsd:restriction> 要素に base 属性の指定がない場合は null を返します。<xsd:restriction> 要素に base 属性の指定がない場合，つまり，局所型定義をベースの型としている場合の例を以下に示します。

```
<simpleType name="ST1" >
  <restriction>
    <simpleType>
      <restriction base="string"/>
    </simpleType>
  </restriction>
</simpleType>
```

<xsd:complexType> 要素では，<xsd:restriction> 要素も <xsd:extension> 要素も存在しない場合，<xsd:anyType を表す修飾名を返します。

6.8.23 XBRLTaxonomyDimensionItem インターフェース

```
public interface XBRLTaxonomyDimensionItem
extends XBRLTaxonomyItem
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

ディメンジョンの要素宣言を表します。
タクソノミー本体に <xsd:element> 要素によって記述される要素宣言のうち，substitutionGroup 属性が "xbrldt:dimensionItem" であるものが，ディメンジョンの要素宣言となります。

メソッドの概要：

XBRLTaxonomyDimensionItem インターフェースのメソッドの概要を次の表に示します。

表 6-120　XBRLTaxonomyDimensionItem インターフェースのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	java.lang.String	getTypedDomainRef() 要素宣言の xbrldt:typedDomainRef 属性の値を取得します。	2.1	1.0
2	XBRLTaxonomyElement	getTypedDomainRefElement() 要素宣言の xbrldt:typedDomainRef 属性の値が示す要素宣言を取得します。	2.1	1.0

（1） getTypedDomainRef

```
public java.lang.String getTypedDomainRef()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

要素宣言の xbrldt:typedDomainRef 属性の値を取得します。

戻り値：

この要素宣言の xbrldt:typedDomainRef 属性の値を返します。

xbrldt:typedDomainRef 属性がない場合は null を返します。

（2） getTypedDomainRefElement

```
public XBRLTaxonomyElement getTypedDomainRefElement()
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

要素宣言の xbrldt:typedDomainRef 属性の値が示す要素宣言を取得します。

戻り値：

この要素宣言の xbrldt:typedDomainRef 属性の値が示す要素宣言を返します。

xbrldt:typedDomainRef 属性がない場合は null を返します。

6.8.24 XBRLTaxonomyHypercubeItem インターフェース

```
public interface XBRLTaxonomyHypercubeItem
extends XBRLTaxonomyItem
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

ハイパーキューブの要素宣言を表します。
タクソノミー本体に <xsd:element> 要素によって記述される要素宣言のうち，substitutionGroup 属性が "xbrldt:hypercubeItem" であるものが，ハイパーキューブの要素宣言となります。

6.9 com.hitachi.xbrl.types パッケージ

com.hitachi.xbrl.types パッケージは，XML Schema のデータ型および XBRL のデータ型を処理するためのクラスを提供します。

クラスの概要：

com.hitachi.xbrl.types パッケージのクラス一覧を次の表に示します。

表 6-121 com.hitachi.xbrl.types パッケージのクラス一覧

項番	クラス	説明	XBRL のバージョン
1	XBRLAbstractBinary	任意の方式に従ってエンコードされたバイナリデータを表す抽象クラスです。	2.0 および 2.1
2	XBRLAbstractPrecision	任意の数値精度を表す抽象クラスです。	2.1
3	XBRLBase64Binary	XML Schema のデータ型 base64Binary を表します。	2.0 および 2.1
4	XBRLDate	XML Schema のデータ型 date を表します。	2.0 および 2.1
5	XBRLDateTime	XML Schema のデータ型 dateTime を表します。	2.0 および 2.1
6	XBRLDateUnion	XBRL のデータ型 dateUnion を表します。	2.0 および 2.1
7	XBRLDecimals	数値アイテムの decimals 属性に指定する値を表します。	2.1
8	XBRLDuration	XML Schema のデータ型 duration を表します。	2.0 および 2.1
9	XBRLFraction	データ型が分数であるアイテムの値を表します。	2.1
10	XBRLGDay	XML Schema のデータ型 gDay を表します。	2.0 および 2.1
11	XBRLGMonth	XML Schema のデータ型 gMonth を表します。	2.0 および 2.1
12	XBRLGMonthDay	XML Schema のデータ型 gMonthDay を表します。	2.0 および 2.1
13	XBRLGYear	XML Schema のデータ型 gYear を表します。	2.0 および 2.1
14	XBRLGYearMonth	XML Schema のデータ型 gYearMonth を表します。	2.0 および 2.1
15	XBRLHexBinary	XML Schema のデータ型 hexBinary を表します。	2.0 および 2.1
16	XBRLPrecision	数値アイテムの precision 属性に指定する値を表します。	2.1
17	XBRLQName	XML Schema のデータ型 QName を表します。	2.0 および 2.1
18	XBRLTime	XML Schema のデータ型 time を表します。	2.0 および 2.1

6.9.1 XBRLAbstractBinary クラス

```
public abstract class XBRLAbstractBinary
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

任意の方式に従ってエンコードされたバイナリデータを表す抽象クラスです。エンコードの方式はサブクラスで定義します。

メソッドの概要：

XBRLAbstractBinary クラスのメソッドの概要を次の表に示します。

表 6-122 XBRLAbstractBinary クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	int	length() XBRLAbstractBinary オブジェクトが保持するバイト配列の長さを取得します。	2.0 および 2.1
2	byte	byteAt(int index) 指定したインデックスの位置にあるバイトを取得します。	2.0 および 2.1
3	byte[]	toByteArray() XBRLAbstractBinary オブジェクトが保持するバイト配列のコピーを取得します。	2.0 および 2.1
4	abstract java.lang.String	toString() XBRLAbstractBinary オブジェクトが保持するバイト配列をエンコードして，文字列に変換します。	2.0 および 2.1
5	boolean	equals(java.lang.Object obj) XBRLAbstractBinary オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.0 および 2.1
6	int	hashCode() XBRLAbstractBinary オブジェクトのハッシュコードを取得します。	2.0 および 2.1

(1) length

```
public int length()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLAbstractBinary オブジェクトが保持するバイト配列の長さを取得します。

戻り値：

XBRLAbstractBinary オブジェクトが保持するバイト配列の長さを返します。

(2) byteAt

```
public byte byteAt(int index)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定したインデックスの位置にあるバイトを取得します。

引数：

index - 取得したいバイトのインデックスを指定します。

戻り値：

指定したインデックスの位置にあるバイトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 index によって指定したインデックスが配列の境界を超える場合にスローされます。[エラーコード：25001，25002]

(3) toByteArray

```
public byte[] toByteArray()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLAbstractBinary オブジェクトが保持するバイト配列のコピーを取得します。

戻り値：

XBRLAbstractBinary オブジェクトが保持するバイト配列のコピーを返します。

(4) toString

```
public abstract java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLAbstractBinary オブジェクトが保持するバイト配列をエンコードして，文字列に変換します。変換後の文字列は，サブクラスで定義するエンコードの方式によって異なります。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

XBRLAbstractBinary オブジェクトが保持するバイト配列をエンコードした文字列を返します。

(5) equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLAbstractBinary オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - XBRLAbstractBinary オブジェクトと比較するオブジェクトを指定します。

戻り値：

XBRLAbstractBinary オブジェクトが引数 obj で指定したオブジェクトと等しい場合は true，等しくない場合は false を返します。

(6) hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLAbstractBinary オブジェクトのハッシュコードを取得します。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

XBRLAbstractBinary オブジェクトのハッシュコードを返します。

6.9.2 XBRLAbstractPrecision クラス

```
public abstract class XBRLAbstractPrecision
extends java.lang.Number
```

XBRL のバージョン：

2.1

説明：

任意の数値精度を表す抽象クラスです。数値または文字列 "INF" を保持します。保持する内容の違いを，isInfinite() メソッドで確認できます。

フィールドの概要：

XBRLAbstractPrecision クラスのフィールドの概要を次の表に示します。

表 6-123　XBRLAbstractPrecision クラスのフィールドの概要

説明	
java.lang.String	INFINITY 数値精度が無限大であることを示す文字列 "INF" を保持します。

メソッドの概要：

XBRLAbstractPrecision クラスのメソッドの概要を次の表に示します。

表 6-124　XBRLAbstractPrecision クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	boolean	isInfinite() XBRLAbstractPrecision オブジェクトが文字列 "INF" を保持するかどうかを調べます。	2.1
2	java.math.BigInteger	getValue() XBRLAbstractPrecision オブジェクトが保持する数値を取得します。	2.1
3	int	intValue() XBRLAbstractPrecision オブジェクトが保持する値を int 型として取得します。	2.1
4	long	longValue() XBRLAbstractPrecision オブジェクトが保持する値を long 型として取得します。	2.1
5	float	floatValue() XBRLAbstractPrecision オブジェクトが保持する値を float 型として取得します。	2.1
6	double	doubleValue() XBRLAbstractPrecision オブジェクトが保持する値を double 型として取得します。	2.1
7	java.lang.String	toString() XBRLAbstractPrecision オブジェクトの文字列表現を取得します。	2.1
8	abstract boolean	equals(java.lang.Object obj) XBRLAbstractPrecision オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.1
9	abstract int	hashCode() XBRLAbstractPrecision オブジェクトのハッシュコードを取得します。	2.1

(1) isInfinite

```
public boolean isInfinite()
```

XBRL のバージョン：

2.1

説明：

XBRLAbstractPrecision オブジェクトが文字列 "INF" を保持するかどうかを調べます。

戻り値：

XBRLAbstractPrecision オブジェクトが文字列 "INF" を保持する場合は true，保持しない場合は false を返します。

(2) getValue

```
public java.math.BigInteger getValue()
```

XBRL のバージョン：

2.1

説明：

XBRLAbstractPrecision オブジェクトが保持する数値を取得します。

戻り値：

XBRLAbstractPrecision オブジェクトが保持する数値を返します。
XBRLAbstractPrecision オブジェクトが文字列 "INF" を保持する場合は null を返します。

(3) intValue

```
public int intValue()
```

XBRL のバージョン：

2.1

説明：

XBRLAbstractPrecision オブジェクトが保持する値を int 型として取得します。

オーバーライド：

クラス java.lang.Number 内の intValue()

戻り値：

XBRLAbstractPrecision オブジェクトが保持する値を int 型として返します。保持する数値が大き過ぎて int に適さない場合は，下位の 32 ビットだけを返します。また，文字列 "INF" を保持する場合は java.lang.Integer.MAX_VALUE を返します。

(4) longValue

```
public long longValue()
```

XBRL のバージョン：

2.1

説明：

XBRLAbstractPrecision オブジェクトが保持する値を long 型として取得します。

オーバーライド：

クラス java.lang.Number 内の longValue()

戻り値：

XBRLAbstractPrecision オブジェクトが保持する値を long 型として返します。保持する数値が大き過ぎて long に適さない場合は，下位の 64 ビットだけを返します。また，文字列 "INF" を保持する場合は java.lang.Long.MAX_VALUE を返します。

（5） floatValue

```
public float floatValue()
```

XBRL のバージョン：

2.1

説明：

XBRLAbstractPrecision オブジェクトが保持する値を float 型として取得します。

オーバーライド：

クラス java.lang.Number 内の floatValue()

戻り値：

XBRLAbstractPrecision オブジェクトが保持する値を float 型として返します。保持する数値の絶対値が大き過ぎて float として表せない場合は，
java.lang.Float.NEGATIVE_INFINITY または
java.lang.Float.POSITIVE_INFINITY に変換した値を返します。また，文字列 "INF" を保持する場合は java.lang.Float.POSITIVE_INFINITY を返します。

（6） doubleValue

```
public double doubleValue()
```

XBRL のバージョン：

2.1

説明：

XBRLAbstractPrecision オブジェクトが保持する値を double 型として取得します。

オーバーライド：

クラス java.lang.Number 内の doubleValue()

戻り値：

XBRLAbstractPrecision オブジェクトが保持する値を double 型として返します。

保持する数値の絶対値が大き過ぎて double として表せない場合は，
java.lang.Double.NEGATIVE_INFINITY または
java.lang.Double.POSITIVE_INFINITY に変換した値を返します。また，文字列 "INF" を保持する場合は java.lang.Double.POSITIVE_INFINITY を返します。

(7) toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.1

説明：

XBRLAbstractPrecision オブジェクトの文字列表現を取得します。
オブジェクト構築時に数値の文字列表現が指定された場合に，指定された文字列表現を java.math.BigInteger に変換します。このためオブジェクト構築時に指定された文字列表現と，toString() によって取得した文字列表現が異なることがあります。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

XBRLAbstractPrecision オブジェクトの文字列表現を返します。

(8) equals

```
public abstract boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.1

説明：

XBRLAbstractPrecision オブジェクトとほかのオブジェクトが等しいかどうかを調べます。等価の基準はサブクラスでの定義によって異なります。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - XBRLAbstractPrecision オブジェクトと比較するオブジェクトを指定します。

戻り値：

XBRLAbstractPrecision オブジェクトが引数 obj で指定されたオブジェクトと等しい場合は true，等しくない場合は false を返します。

(9) hashCode

```
public abstract int hashCode()
```

XBRL のバージョン：

2.1

説明：

XBRLAbstractPrecision オブジェクトのハッシュコードを取得します。取得するハッシュコードは，サブクラスでの定義によって異なります。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

XBRLAbstractPrecision オブジェクトのハッシュコードを返します。

6.9.3 XBRLBase64Binary クラス

```
public final class XBRLBase64Binary
extends XBRLAbstractBinary
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML Schema のデータ型 base64Binary を表します。base64Binary 型は，バイナリデータを Base64 でエンコードしたものです。このクラスでは，Base64 でエンコードした文字列とバイト配列を保持します。

コンストラクタの概要：

XBRLBase64Binary クラスのコンストラクタの概要を次の表に示します。

表 6-125 XBRLBase64Binary クラスのコンストラクタの概要

項番	説明	XBRL のバージョン
1	XBRLBase64Binary(java.lang.String base64) 指定した文字列から XBRLBase64Binary オブジェクトを構築します。	2.0 および 2.1
2	XBRLBase64Binary(byte[] bytes) 指定したバイト配列から XBRLBase64Binary オブジェクトを構築します。	2.0 および 2.1
3	XBRLBase64Binary(byte[] bytes, int offset, int length) 指定したバイト配列の一部から XBRLBase64Binary オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLBase64Binary クラスのメソッドの概要を次の表に示します。

表 6-126　XBRLBase64Binary クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	static byte[]	toByteArray(java.lang.String base64) 指定した文字列を Base64 でデコードして，バイト配列を生成します。	2.0 および 2.1
2	java.lang.String	toString() XBRLBase64Binary オブジェクトが保持するバイト配列を Base64 でエンコードして，その結果を取得します。	2.0 および 2.1
3	static java.lang.String	toString(byte[] bytes) 指定したバイト配列を Base64 でエンコードして，その結果を取得します。	2.0 および 2.1
4	static java.lang.String	toString(byte[] bytes, int offset, int length) 指定されたバイト配列の一部を Base64 でエンコードして，その結果を取得します。	2.0 および 2.1
5	static XBRLBase64Binary	valueOf(java.lang.String base64) 指定した文字列から XBRLBase64Binary オブジェクトを取得します。	2.0 および 2.1
6	static XBRLBase64Binary	valueOf(byte[] bytes) 指定したバイト配列から XBRLBase64Binary オブジェクトを取得します。	2.0 および 2.1
7	static XBRLBase64Binary	valueOf(byte[] bytes, int offset, int length) 指定したバイト配列の一部から XBRLBase64Binary オブジェクトを取得します。	2.0 および 2.1

(1) XBRLBase64Binary

```
public  XBRLBase64Binary(java.lang.String base64)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した文字列から XBRLBase64Binary オブジェクトを構築します。

このコンストラクタは，引数 base64 で指定した文字列を Base64 でデコードしてバイト配列を生成し，それを保持します。

引数：

base64 - デコードする文字列（Base64 コード表現の文字列）を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 base64 に null を指定した場合にスローされます。［エラーコード：20002］

- 引数 base64 に指定した文字列をデコードする処理でエラーが検出された場合にスローされます。［エラーコード：25004，25006，25007］

（2）XBRLBase64Binary

```
public  XBRLBase64Binary(byte[] bytes)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定したバイト配列から XBRLBase64Binary オブジェクトを構築します。指定したバイト配列はコピーされます。

引数：

bytes - バイト配列を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 bytes に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 bytes に長さ 0 の配列を指定した場合にスローされます。［エラーコード：25007］

（3）XBRLBase64Binary

```
public  XBRLBase64Binary(byte[] bytes,
                        int offset,
                        int length)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定したバイト配列の一部から XBRLBase64Binary オブジェクトを構築します。バイト配列を指定した部分の内容がコピーされます。

引数：

bytes - バイト配列を指定します。

offset - 読み込み開始位置のインデックスを指定します。

length - 読み込むバイト数を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 bytes に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 offset および引数 length の指定によるインデックスが配列の境界を超える場合にスローされます。［エラーコード：25001，25003，25010］
- 引数 bytes に長さ 0 の配列を指定した場合にスローされます。［エラーコード：25007］

(4) toByteArray

```
public static byte[] toByteArray(java.lang.String base64)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した文字列を Base64 でデコードして，バイト配列を生成します。

引数：

base64 - デコードする文字列（Base64 の表現の文字列）を指定します。

戻り値：

指定した文字列を Base64 でデコードして生成されたバイト配列を返します。

例外：

XBRLDOMRuntimeException -

- 引数 base64 に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 base64 に指定した文字列をデコードする処理で，エラーが検出された場合にスローされます。［エラーコード：25004，25006，25007］

(5) toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLBase64Binary オブジェクトが保持するバイト配列を Base64 でエンコードして，その結果を取得します。

定義：

クラス XBRLAbstractBinary 内の toString()

戻り値：

XBRLBase64Binary オブジェクトが保持するバイト配列を Base64 でエンコードした結果を返します。

(6) toString

```
public static java.lang.String toString(byte[] bytes)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定したバイト配列を Base64 でエンコードして，その結果を取得します。

引数：

bytes - バイト配列を指定します。

戻り値：

指定したバイト配列を Base64 でエンコードした結果を返します。

例外：

XBRLDOMRuntimeException -

- 引数 bytes に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 bytes に長さ 0 の配列を指定した場合にスローされます。［エラーコード：25007］

（7） toString

```
public static java.lang.String toString(byte[] bytes,
                                        int offset,
                                        int length)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定されたバイト配列の一部を Base64 でエンコードして，その結果を取得します。

引数：

bytes - バイト配列を指定します。

offset - 読み込み開始位置のインデックスを指定します。

length - 読み込むバイト数を指定します。

戻り値：

指定したバイト配列の一部を Base64 でエンコードした結果を返します。

例外：

XBRLDOMRuntimeException -

- 引数 bytes に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 offset および length の指定によるインデックスが配列の境界を超える場合にスローされます。［エラーコード：25001，25003，25010］
- 引数 bytes に長さ 0 の配列を指定した場合にスローされます。［エラーコード：25007］

（8） valueOf

```
public static XBRLBase64Binary valueOf(java.lang.String base64)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した文字列から XBRLBase64Binary オブジェクトを取得します。このメソッドが返す XBRLBase64Binary オブジェクトは，指定した文字列を Base64 でデコー

ドして生成したバイト配列を保持します。

引数：

base64 - デコードする文字列（Base64 コード表現の文字列）を指定します。

戻り値：

指定した文字列をデコードして生成した XBRLBase64Binary オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 base64 に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 base64 に指定した文字列をデコードする処理でエラーが検出された場合にスローされます。［エラーコード：25004，25006，25007］

(9) valueOf

```
public static XBRLBase64Binary valueOf(byte[] bytes)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定したバイト配列から XBRLBase64Binary オブジェクトを取得します。指定したバイト配列はコピーされます。

引数：

bytes - バイト配列を指定します。

戻り値：

指定したバイト配列から生成した XBRLBase64Binary オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 bytes に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 bytes に長さ 0 の配列を指定した場合にスローされます。［エラーコード：25007］

(10) valueOf

```
public static XBRLBase64Binary valueOf(byte[] bytes,
                                       int offset,
                                       int length)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定したバイト配列の一部から XBRLBase64Binary オブジェクトを取得します。

バイト配列を指定した部分の内容がコピーされます。

引数：

bytes - バイト配列を指定します。

offset - 読み込み開始位置のインデックスを指定します。

length - 読み込むバイト数を指定します。

戻り値：

指定したバイト配列の一部から生成された XBRLBase64Binary オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 bytes に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 offset および length の指定によるインデックスが配列の境界を超える場合にスローされます。［エラーコード：25001，25003，25010］
- 引数 bytes に長さ 0 の配列を指定した場合にスローされます。［エラーコード：25007］

6.9.4 XBRLDate クラス

```
public final class XBRLDate
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML Schema のデータ型 date を表します。

注　このクラスは equals() と一貫性のない，自然順序づけを持ちます。

コンストラクタの概要：

XBRLDate クラスのコンストラクタの概要を次の表に示します。

表 6-127　XBRLDate クラスのコンストラクタの概要

項番	説明	XBRL のバージョン
1	XBRLDate(java.lang.String date) 指定した date 型データの文字列表現から XBRLDate オブジェクトを構築します。	2.0 および 2.1
2	XBRLDate(XBRLDate other) 指定した XBRLDate オブジェクトから，XBRLDate オブジェクトを新規に構築します。	2.0 および 2.1
3	XBRLDate(XBRLDateTime other) 指定した XBRLDate オブジェクトから，XBRLDate オブジェクトを新規に構築します。	2.0 および 2.1

メソッドの概要：

XBRLDate クラスのメソッドの概要を次の表に示します。

表 6-128　XBRLDate クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	int	compareTo(XBRLDate other) XBRLDate オブジェクトとほかの XBRLDate オブジェクトを比較します。	2.0 および 2.1
2	java.lang.String	toString() date 型データの文字列表現を取得します。	2.0 および 2.1
3	boolean	equals(java.lang.Object obj) XBRLDate オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.0 および 2.1
4	int	hashCode() XBRLDate オブジェクトのハッシュコードを取得します。	2.0 および 2.1
5	XBRLDate	add(XBRLDuration duration) 指定した XBRLDuration オブジェクトを加算した XBRLDate オブジェクトを，新規に取得します。	2.0 および 2.1

(1) XBRLDate

```
public XBRLDate(java.lang.String date)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した date 型データの文字列表現から，XBRLDate オブジェクトを構築します。次の文字列表現だけを指定できます。文字列表現が不正な場合は例外がスローされます。

指定できる文字列表現：BNF 表記

```
dateFormat ::= ('-')?(date | date timezone)
date ::= year '-' month '-' day
timezone ::= 'Z' | ('+'|'-') hour ':' minute
year ::= ([0-9])*[0-9][0-9][0-9][0-9]
month ::= [0][1-9] | [1][0-2]
day ::= [0][1-9] | [1-2][0-9] | [3][0-1]
hour ::= [0-1][0-9] | [2][0-3]
minute ::= [0-5][0-9]
```

year には「0」を指定できません。day については，BNF 表記で表現できない月ごとの最大値もチェックします。うるう年も考慮してチェックします。

引数：

date - date 型データの文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 date に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 date に指定した date 型データの文字列表現が不正な場合にスローされます。［エラーコード：25000］

（2） XBRLDate

```
public  XBRLDate(XBRLDate other)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した XBRLDate オブジェクトから，XBRLDate オブジェクトを新規に構築します。

引数：

other - XBRLDate オブジェクトを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 other に null を指定した場合にスローされます。［エラーコード：20002］

（3） XBRLDate

```
public  XBRLDate(XBRLDateTime other)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した XBRLDate オブジェクトから，XBRLDate オブジェクトを新規に構築します。指定した XBRLDateTime オブジェクトの時間データは無視し，日付データだけで新たなオブジェクトを構築します。

引数：

other - XBRLDateTime オブジェクトを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 other に null を指定した場合にスローされます。［エラーコード：20002］

（4） compareTo

```
public int compareTo(XBRLDate other)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDate オブジェクトとほかの XBRLDate オブジェクトとを比較します。
タイムゾーンオフセットを反映した結果で比較します。タイムゾーン指定のないものについては UTC（協定世界時）と仮定し，比較します。
注　このメソッドの実行結果と equals() とは一貫性がありません。

引数：

other - 比較対象の XBRLDate オブジェクトを指定します。

戻り値：

この XBRLDate オブジェクトが引数 other で指定した XBRLDate オブジェクトと等しい場合は 0 を返します。この XBRLDate オブジェクトが引数 other で指定した XBRLDate オブジェクトよりも小さい場合は 0 より小さい値，大きい場合は 0 より大きい値を返します。

(5) toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

date 型データの文字列表現を取得します。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

この date 型データの文字列表現を返します。

(6) equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDate オブジェクトとほかのオブジェクトが等しいかどうかを調べます。
タイムゾーンを含み同じ文字列表現となる場合だけ，等しいものとみなします。タイムゾーンオフセットを反映した結果が同じ値となる場合でも，タイムゾーンの表現方法が別表現の場合は異なるものとみなします。
注　このメソッドの実行結果と compareTo() とは一貫性がありません。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - この XBRLDate オブジェクトと，比較するオブジェクトを指定します。

戻り値：

この XBRLDate オブジェクトが引数 obj で指定したオブジェクトと等しい場合は true，等しくない場合は false を返します。

(7) hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDate オブジェクトのハッシュコードを取得します。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

この XBRLDate オブジェクトのハッシュコードを返します。

(8) add

```
public XBRLDate add(XBRLDuration duration)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した XBRLDuration オブジェクトを加算した XBRLDate オブジェクトを，新規に取得します。

引数：

duration - XBRLDuration オブジェクトを指定します。

戻り値：

指定した XBRLDuration オブジェクトを加算した新たな XBRLDate オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 duration に null を指定した場合にスローされます。［エラーコード：20002］

6.9.5 XBRLDateTime クラス

```
public final class XBRLDateTime
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML Schema のデータ型 dateTime を表します。

注　このクラスは equals() と一貫性のない，自然順序づけを持ちます。

コンストラクタの概要：

XBRLDateTime クラスのコンストラクタの概要を次の表に示します。

表 6-129　XBRLDateTime クラスのコンストラクタの概要

項番	説明	XBRL のバージョン
1	XBRLDateTime(java.lang.String datetime) 指定した dateTime 型データの文字列表現から，XBRLDateTime オブジェクトを構築します。	2.0 および 2.1
2	XBRLDateTime(XBRLDateTime other) 指定した XBRLDateTime オブジェクトから，XBRLDateTime オブジェクトを新規に構築します。	2.0 および 2.1
3	XBRLDateTime(XBRLDate other) 指定した XBRLDateTime オブジェクトから，時間が "00:00:00.000" で始まる XBRLDateTime オブジェクトを新規に構築します。	2.0 および 2.1

メソッドの概要：

XBRLDateTime クラスのメソッドの概要を次の表に示します。

表 6-130　XBRLDateTime クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	int	compareTo(XBRLDateTime other) XBRLDateTime オブジェクトとほかの XBRLDateTime オブジェクトを比較します。	2.0 および 2.1
2	java.lang.String	toString() dateTime 型データの文字列表現を取得します。	2.0 および 2.1
3	boolean	equals(java.lang.Object obj) XBRLDateTime オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.0 および 2.1
4	int	hashCode() XBRLDateTime オブジェクトのハッシュコードを取得します。	2.0 および 2.1
5	XBRLDateTime	add(XBRLDuration duration) 指定した XBRLDuration オブジェクトを加算した XBRLDateTime オブジェクトを，新規に取得します。	2.0 および 2.1

(1) XBRLDateTime

```
public  XBRLDateTime(java.lang.String datetime)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した dateTime 型データの文字列から，XBRLDateTime オブジェクトを構築します。次の文字列表現だけを指定できます。文字列表現が不正な場合は例外がスローされます。

指定できる文字列表現：BNF 表記

```
datetimeFormat ::= ('-')?(date | date timezone)
date ::= year '-' month '-' day 'T' hour ':' minute ':' second
timezone ::= 'Z' | ('+'|'-') hour ':' minute
year ::= ([0-9])*[0-9][0-9][0-9][0-9]
month ::= [0][1-9] | [1][0-2]
day ::= [0][1-9] | [1-2][0-9] | [3][0-1]
hour ::= [0-1][0-9] | [2][0-3]
minute ::= [0-5][0-9]
second ::= [0-5][0-9] | [0-5][0-9]'.'([0-9])+
```

year には「0」を指定できません。day については，BNF 表記で表現できない月ごとの最大値もチェックします。うるう年も考慮してチェックします。うるう秒は考慮しません。

秒以下の値は，小数第 4 桁で四捨五入しミリ秒単位に丸めます。このためオブジェクト構築時に指定した文字列表現と，toString によって取得した文字列表現が異なることがあります。

引数：

datetime - dateTime 型データの文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 datetime に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 datetime に指定した dateTime 型データの文字列表現が不正な場合にスローされます。［エラーコード：25000］

(2) XBRLDateTime

```
public  XBRLDateTime(XBRLDateTime other)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した XBRLDateTime オブジェクトから，XBRLDateTime オブジェクトを新規に構築します。

引数：

other - XBRLDateTime オブジェクトを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 other に null を指定した場合にスローされます。［エラーコード：20002］

(3) XBRLDateTime

```
public  XBRLDateTime(XBRLDate other)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した XBRLDateTime オブジェクトから，XBRLDateTime オブジェクトを新規に構築します。新たな XBRLDateTime オブジェクトの時間は「00:00:00.000」となります。

引数：

other - XBRLDate オブジェクトを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 other に null を指定した場合にスローされます。［エラーコード：20002］

(4) compareTo

```
public int compareTo(XBRLDateTime other)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDateTime オブジェクトとほかの XBRLDateTime オブジェクトを比較します。

タイムゾーンオフセットを反映した結果で比較します。タイムゾーン指定のないものについては UTC（協定世界時）と仮定し，比較します。

注　このメソッドの実行結果と equals() とは一貫性がありません。

引数：

other - 比較対象の XBRLDateTime オブジェクトを指定します。

戻り値：

この XBRLDateTime オブジェクトが，引数 other で指定した XBRLDateTime オブジェクトと等しい場合は 0 を返します。引数 other で指定した XBRLDateTime オブジェクトよりも小さい場合は 0 より小さい値，大きい場合は 0 より大きい値を返します。

(5) toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

dateTime 型データの文字列を取得します。
秒以下の値は必ず小数第 3 桁で表現します。このためオブジェクト構築時に指定した文字列表現と，toString() によって取得した文字列表現が異なることがあります。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

この dateTime 型データの文字列表現を返します。

(6) equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDateTime オブジェクトとほかのオブジェクトが等しいかどうかを調べます。
タイムゾーンを含み同じ文字列表現となる場合だけを，等しいものとみなします。
タイムゾーンオフセットを反映した結果が同じ値となる場合でも，タイムゾーンの表現方法が別表現の場合は異なるものとみなします。
注　このメソッドの実行結果と compareTo() とは一貫性がありません。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - この XBRLDateTime オブジェクトと比較するオブジェクトを指定します。

戻り値：

この XBRLDateTime オブジェクトが，引数 obj で指定したオブジェクトと等しい場合は true，等しくない場合は false を返します。

(7) hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDateTime オブジェクトのハッシュコードを取得します。

オーバーライド：
クラス java.lang.Object 内の hashCode()

戻り値：
この XBRLDateTime オブジェクトのハッシュコードを返します。

(8) add

```
public XBRLDateTime add(XBRLDuration duration)
```

XBRL のバージョン：
2.0 および 2.1

説明：
指定した XBRLDuration オブジェクトを加算した XBRLDateTime オブジェクトを，新規に取得します。

引数：
duration - XBRLDuration オブジェクトを指定します。

戻り値：
指定した XBRLDuration オブジェクトを加算した，新たな XBRLDateTime オブジェクトを返します。

例外：
XBRLDOMRuntimeException -
- 引数 duration に null を指定した場合にスローされます。［エラーコード：20002］

6.9.6 XBRLDateUnion クラス

```
public final class XBRLDateUnion
extends java.lang.Object
```

XBRL のバージョン：
2.0 および 2.1

説明：
XBRL のデータ型 dateUnion を表します。
XBRL のデータ型 dateUnion は，XML Schema で次のとおり定義されています。

```
<simpleType name="dateUnion">
        <union memberTypes="date dateTime"/>
</simpleType>
```

注　このクラスは equals() と一貫性のない，自然順序づけを持ちます。

フィールドの概要：
XBRLDateUnion クラスのフィールドの概要を次の表に示します。

表 6-131　XBRLDateUnion クラスのフィールドの概要

項番	説明	
1	static int	DATE この dateUnion 型のメンバは，XML Schema の date 型です。
2	static int	DATETIME この dateUnion 型のメンバは，XML Schema の dateTime 型です。

コンストラクタの概要：

XBRLDateUnion クラスのコンストラクタの概要を次の表に示します。

表 6-132　XBRLDateUnion クラスのコンストラクタの概要

項番	説明	XBRL のバージョン
1	XBRLDateUnion(java.lang.String dateunion) 指定した dateUnion 型データの文字列表現から，XBRLDateUnion オブジェクトを構築します。	2.0 および 2.1
2	XBRLDateUnion(XBRLDate date) 指定した XBRLDate オブジェクトから，XBRLDateUnion オブジェクトを構築します。	2.0 および 2.1
3	XBRLDateUnion(XBRLDateTime datetime) 指定した XBRLDateTime オブジェクトから，XBRLDateUnion オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLDateUnion クラスのメソッドの概要を次の表に示します。

表 6-133　XBRLDateUnion クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	int	getMemberType() dateUnion 型データのメンバの型を調べます。	2.0 および 2.1
2	java.lang.Object	getMember() dateUnion 型データのメンバを取得します。	2.0 および 2.1
3	int	compareTo(XBRLDateUnion other) XBRLDateUnion オブジェクトとほかの XBRLDateUnion オブジェクトを比較します。	2.0 および 2.1
4	java.lang.String	toString() dateUnion 型データの文字列表現を取得します。	2.0 および 2.1
5	boolean	equals(java.lang.Object obj) XBRLDateUnion オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.0 および 2.1

項番	説明		XBRL のバージョン
6	int	hashCode() XBRLDateUnion オブジェクトのハッシュコードを取得します。	2.0 および 2.1

(1) XBRLDateUnion

```
public  XBRLDateUnion(java.lang.String dateunion)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した dateUnion 型データの文字列表現から，XBRLDateUnion オブジェクトを構築します。次の文字列表現だけ指定できます。文字列表現が不正な場合は例外がスローされます。

指定できる文字列表現：BNF 表記

```
datetimeFormat ::= ('-')?(date | date timezone)
date ::= year '-' month '-' day 'T' hour ':' minute ':' second
| year '-' month '-' day
timezone ::= 'Z' | ('+'|'-') hour ':' minute
year ::= ([0-9])*[0-9][0-9][0-9][0-9]
month ::= [0][0-9] | [1][0-2]
day ::= [0-2][0-9] | [3][0-1]
hour ::= [0-1][0-9] | [2][0-3]
minute ::= [0-5][0-9]
second ::= [0-5][0-9] | [0-5][0-9]'.'([0-9])+
```

year には「0」を指定できません。day は，BNF 表記で表現できない月ごとの最大値もチェックします。うるう年も考慮してチェックします。うるう秒は考慮しません。

秒以下の値は，小数第 4 桁で四捨五入しミリ秒単位に丸めます。このためオブジェクト構築時に指定した文字列表現と，toString() によって取得した文字列表現が異なることがあります。

引数：

dateunion - dateUnion 型データの文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 dateunion に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 dateunion に指定した dateUnion 型データの文字列表現が不正な場合にスローされます。［エラーコード：25000］

(2) XBRLDateUnion

```
public  XBRLDateUnion(XBRLDate date)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した XBRLDate オブジェクトから，XBRLDateUnion オブジェクトを構築します。

引数：

date - date 型データを表す XBRLDate オブジェクトを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 date に null を指定した場合にスローされます。［エラーコード：20002］

（3） XBRLDateUnion

```
public  XBRLDateUnion(XBRLDateTime datetime)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した XBRLDateTime オブジェクトから，XBRLDateUnion オブジェクトを構築します。

引数：

datetime - dateTime 型データを表す XBRLDateTime オブジェクトを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 datetime に null を指定した場合にスローされます。［エラーコード：20002］

（4） getMemberType

```
public int getMemberType()
```

XBRL のバージョン：

2.0 および 2.1

説明：

dateUnion 型データのメンバの型を調べます。

戻り値：

dateUnion 型データのメンバの型を表すコードを返します。

（5） getMember

```
public java.lang.Object getMember()
```

XBRL のバージョン：

2.0 および 2.1

説明：

dateUnion 型データのメンバを取得します。

戻り値：

この dateUnion 型データのメンバを返します。このメソッドの戻り値は，XBRLDate または XBRLDateTime のどちらかにキャストできます。どちらにキャストできるかは，getMemberType() で調べることができます。

(6) compareTo

```
public int compareTo(XBRLDateUnion other)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDateUnion オブジェクトとほかの XBRLDateUnion オブジェクトを比較します。

タイムゾーンオフセットを反映した結果で比較します。タイムゾーン指定のないものについては UTC（協定世界時）と仮定し，比較します。また，date 型データの場合は，時間を「00:00:00.000」と仮定し比較します。

注　このメソッドの実行結果と equals() とは一貫性がありません。

引数：

other - 比較対象の XBRLDateUnion オブジェクトを指定します。

戻り値：

この XBRLDateUnion オブジェクトが，引数 other で指定した XBRLDateUnion オブジェクトと等しい場合は 0 を返します。引数 other で指定した XBRLDateUnion オブジェクトよりも小さい場合は 0 より小さい値，大きい場合は 0 より大きい値を返します。

(7) toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

dateUnion 型データの文字列表現を取得します。

このオブジェクトで保持する値が dateTime 型データの場合，秒以下の値は必ず小数第 3 桁で表現します。このためオブジェクト構築時に指定した文字列表現と，toString によって取得した文字列表現が異なることがあります。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

dateUnion 型データの文字列表現を返します。

(8) equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDateUnion オブジェクトとほかのオブジェクトが等しいかどうかを調べます。タイムゾーンを含み同じ文字列表現となる場合だけ，等しいものとみなします。dateTime 型データと date 型データが同じ値となる場合でも，オブジェクト構築時に指定した文字列表現またはオブジェクト型が異なる場合には，等しいものとみなしません。タイムゾーンオフセットを反映した結果が同じ値となる場合でも，タイムゾーンの表現方法が別表現の場合は異なるものとみなします。

注　このメソッドの実行結果と compareTo() とは一貫性がありません。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - XBRLDateUnion オブジェクトと比較するオブジェクトを指定します。

戻り値：

XBRLDateUnion オブジェクトが，引数 obj で指定したオブジェクトと等しい場合は true，等しくない場合は false を返します。

(9) hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDateUnion オブジェクトのハッシュコードを取得します。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

XBRLDateUnion オブジェクトのハッシュコードを返します。

6.9.7 XBRLDecimals クラス

```
public final class XBRLDecimals
extends XBRLAbstractPrecision
```

XBRL のバージョン：

2.1

説明：

数値アイテムの decimals 属性に指定する値を表します。整数または文字列 "INF" を保持します。

コンストラクタの概要：

XBRLDecimals クラスのコンストラクタの概要を次の表に示します。

表 6-134　XBRLDecimals クラスのコンストラクタの概要

項番	説明	XBRL のバージョン
1	XBRLDecimals(java.math.BigInteger value) 指定した数値から XBRLDecimals オブジェクトを構築します。	2.1
2	XBRLDecimals(java.lang.String decimals) 指定した decimals 属性の値の文字列表現から XBRLDecimals オブジェクトを構築します。	2.1

メソッドの概要：

XBRLDecimals クラスのメソッドの概要を次の表に示します。

表 6-135　XBRLDecimals クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	boolean	equals(java.lang.Object obj) XBRLDecimals オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.1
2	int	hashCode() XBRLDecimals オブジェクトのハッシュコードを取得します。	2.1

(1) XBRLDecimals

```
public  XBRLDecimals(java.math.BigInteger value)
```

XBRL のバージョン：

2.1

説明：

指定した数値から XBRLDecimals オブジェクトを構築します。

引数：

value - 数値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 value に null を指定した場合にスローされます。［エラーコード：20002］

（2） XBRLDecimals

```
public  XBRLDecimals(java.lang.String decimals)
```

XBRL のバージョン：

2.1

説明：

指定した decimals 属性の値の文字列表現から XBRLDecimals オブジェクトを構築します。

引数 decimals には整数を表す文字列（XML Schema のデータ型 integer），または文字列 "INF" を指定できます。

引数 decimals に数値の文字列表現を指定した場合，構築するオブジェクトは文字列表現を java.math.BigInteger に変換した値を保持します。このためオブジェクト構築時に指定された文字列表現と，toString() によって取得した文字列表現が異なることがあります。

引数：

decimals - decimals 属性の値の文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 decimals に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 decimals に指定された decimals 属性の，値の文字列表現が不正な場合にスローされます。［エラーコード：25000］

（3） equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.1

説明：

XBRLDecimals オブジェクトとほかのオブジェクトが等しいかどうかを調べます。結果が true になるのは，このオブジェクトと文字列表現が等しい XBRLDecimals オブジェクトだけです。

定義：

クラス XBRLAbstractPrecision 内の equals()

引数：

obj - XBRLDecimals オブジェクトと比較するオブジェクトを指定します。

戻り値：

XBRLDecimals オブジェクトが，引数 obj で指定されたオブジェクトと等しい場合

は true，等しくない場合は false を返します。

(4) hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.1

説明：

XBRLDecimals オブジェクトのハッシュコードを取得します。

定義：

クラス XBRLAbstractPrecision 内の hashCode()

戻り値：

XBRLDecimals オブジェクトのハッシュコードを返します。

6.9.8 XBRLDuration クラス

```
public final class XBRLDuration
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML Schema のデータ型 duration を表します。

コンストラクタの概要：

XBRLDuration クラスのコンストラクタの概要を次の表に示します。

表 6-136 XBRLDuration クラスのコンストラクタの概要

説明	XBRL のバージョン
XBRLDuration(java.lang.String duration) 指定した duration 型データの文字列表現から，XBRLDuration オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLDuration クラスのメソッドの概要を次の表に示します。

表 6-137 XBRLDuration クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	toString() duration 型データの文字列表現を取得します。	2.0 および 2.1

項番	説明		XBRL のバージョン
2	boolean	equals(java.lang.Object obj) XBRLDuration オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.0 および 2.1
3	int	hashCode() XBRLDuration オブジェクトのハッシュコードを取得します。	2.0 および 2.1

(1) XBRLDuration

```
public  XBRLDuration(java.lang.String duration)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した duration 型データの文字列表現から，XBRLDuration オブジェクトを構築します。

次の文字列表現だけ指定できます。文字列表現が不正な場合は例外がスローされます。

指定できる文字列表現：BNF 表記

```
durationFormat ::= ('-')? 'P' (date | time | date time)
date ::= year month day | year month | year day | month day
| year | month | day
year ::= ([0-9])+ 'Y'
month ::= ([0-9])+ 'M'
day ::= ([0-9])+ 'D'
time ::= 'T' (hour minute second | hour minute | hour second
| minute second | hour| minute | second)
hour ::= ([0-9])+ 'H'
minute ::= ([0-9])+ 'M'
second ::= ([0-9])+ 'S' | ([0-9])+ '.' ([0-9])+ 'S'
```

秒以下の値は，小数第 4 桁で四捨五入しミリ秒単位に丸めます。このためオブジェクト構築時に指定した文字列表現と，toString() によって取得した文字列表現が異なることがあります。

引数：

duration - duration 型データの文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 duration に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 duration に指定した duration 型データの文字列表現が不正な場合にスローされます。[エラーコード：25000]

(2) toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

duration 型データの文字列表現を取得します。
秒以下の値は必ず小数第 3 桁で表現します。このためオブジェクト構築時に指定した文字列表現と，toString() によって取得した文字列表現が異なることがあります。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

duration 型データの文字列表現を返します。

(3) equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDuration オブジェクトとほかのオブジェクトが等しいかどうかを調べます。
同じ文字列表現となる場合だけ等しいものとみなします。よって，同じ値となる場合でも文字列表現が異なる場合には等しいものとみなしません。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - この XBRLDuration オブジェクトと比較するオブジェクトを指定します。

戻り値：

XBRLDuration オブジェクトが，引数 obj で指定したオブジェクトと等しい場合は true，等しくない場合は false を返します。

(4) hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLDuration オブジェクトのハッシュコードを取得します。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

XBRLDuration オブジェクトのハッシュコードを返します。

6.9.9 XBRLFraction クラス

```
public final class XBRLFraction
extends java.lang.Object
```

XBRL のバージョン：

2.1

説明：

データ型が分数であるアイテムの値を表します。

分数のデータ型は分子要素 numerator と分母要素 denominator で構成されます。

分子要素は 10 進数，分母要素は 0 を除く 10 進数を含んでいる必要があります。

コンストラクタの概要：

XBRLFraction クラスのコンストラクタの概要を次の表に示します。

表 6-138 XBRLFraction クラスのコンストラクタの概要

項番	説明	XBRL のバージョン
1	XBRLFraction(java.math.BigDecimal numerator, java.math.BigDecimal denominator) 指定した分子要素と分母要素で構成される XBRLFraction オブジェクトを構築します。	2.1
2	XBRLFraction(java.lang.String fraction) 指定した分数の文字列表現から，XBRLFraction オブジェクトを構築します。	2.1

メソッドの概要：

XBRLFraction クラスのメソッドの概要を次の表に示します。

表 6-139 XBRLFraction クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	java.math.BigDecimal	getNumerator() XBRLFraction オブジェクトの分子要素を取得します。	2.1
2	java.math.BigDecimal	getDenominator() XBRLFraction オブジェクトの分母要素を取得します。	2.1

項番	説明		XBRL のバージョン
3	java.lang.String	toString() XBRLFraction オブジェクトの文字列を取得します。	2.1
4	boolean	equals(java.lang.Object obj) XBRLFraction オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.1
5	int	hashCode() XBRLFraction オブジェクトのハッシュコードを取得します。	2.1

(1) XBRLFraction

```
public  XBRLFraction(java.math.BigDecimal numerator,
                     java.math.BigDecimal denominator)
```

XBRL のバージョン：

2.1

説明：

指定した分子要素と分母要素で構成される XBRLFraction オブジェクトを構築します。

引数：

numerator - 分数を構成する分子要素を指定します。

denominator - 分数を構成する分母要素を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 numerator または denominator に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 denominator の値が 0 の場合にスローされます。［エラーコード：25011］

(2) XBRLFraction

```
public  XBRLFraction(java.lang.String fraction)
```

XBRL のバージョン：

2.1

説明：

指定した分数の文字列表現から XBRLFraction オブジェクトを構築します。分数の文字列表現は "numerator/denominator" です。

numerator および denominator には，共に XML Schema のデータ型 decimal として妥当な文字列だけを指定できます。ただし，denominator に 0 を表す文字列（"+0.00" や "-0.0" など）は指定できません。

構築されるオブジェクトは，numerator および denominator に指定された文字列を

java.math.BigDecimal に変換した値を保持します。このためオブジェクト構築時に指定された文字列表現と，toString() によって取得した文字列表現が異なることがあります。

引数：

fraction - 分数の文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 fraction に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 fraction に指定された分数の文字列表現が不正な場合にスローされます。［エラーコード：25000］

（3） getNumerator

```
public java.math.BigDecimal getNumerator()
```

XBRL のバージョン：

2.1

説明：

XBRLFraction オブジェクトの分子要素を取得します。

戻り値：

XBRLFraction オブジェクトの分子要素を返します。

（4） getDenominator

```
public java.math.BigDecimal getDenominator()
```

XBRL のバージョン：

2.1

説明：

XBRLFraction オブジェクトの分母要素を取得します。

戻り値：

XBRLFraction オブジェクトの分母要素を返します。

（5） toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.1

説明：

XBRLFraction オブジェクトの文字列表現を取得します。XBRLFraction オブジェクトの文字列表現は "numerator/denominator" です。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

XBRLFraction オブジェクトの文字列表現を返します。

(6) equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.1

説明：

XBRLFraction オブジェクトとほかのオブジェクトが等しいかどうかを調べます。結果が true になるのは，このオブジェクトの正規形（約分した形）と等しい正規形を持つ XBRLFraction オブジェクトが指定された場合だけです。文字列表現が異なる場合でも，分数として計算した結果が同値となる場合には等しいとみなされます。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - XBRLFraction オブジェクトと比較するオブジェクトを指定します。

戻り値：

XBRLFraction オブジェクトが，引数 obj で指定されたオブジェクトと等しい場合は true，等しくない場合は false を返します。

(7) hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.1

説明：

XBRLFraction オブジェクトのハッシュコードを取得します。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

XBRLFraction オブジェクトのハッシュコードを返します。

6.9.10 XBRLGDay クラス

```
public final class XBRLGDay
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML Schema のデータ型 gDay を表します。

コンストラクタの概要：

XBRLGDay クラスのコンストラクタの概要を次の表に示します。

表 6-140　XBRLGDay クラスのコンストラクタの概要

説明	XBRL のバージョン
XBRLGDay(java.lang.String gday) 指定した gDay 型データの文字列表現から，XBRLGDay オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLGDay クラスのメソッドの概要を次の表に示します。

表 6-141　XBRLGDay クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	toString() gDay 型データの文字列表現を取得します。	2.0 および 2.1
2	boolean	equals(java.lang.Object obj) XBRLGDay オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.0 および 2.1
3	int	hashCode() XBRLGDay オブジェクトのハッシュコードを取得します。	2.0 および 2.1

(1) XBRLGDay

```
public  XBRLGDay(java.lang.String gday)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した gDay 型データの文字列表現から，XBRLGDay オブジェクトを構築します。

次の文字列表現だけ指定できます。文字列表現が不正な場合は例外がスローされます。

指定できる文字列表現：BNF 表記

```
gDayFormat ::= '-' '-' '-' (day | day timezone)
day ::= [0][1-9] | [1-2][0-9] | [3][0-1]
timezone ::= 'Z' | ('+'|'-') hour ':' minute
hour ::= [0-1][0-9] | [2][0-3]
```

```
minute ::= [0-5][0-9]
```

day については，BNF 表記で表現できない月ごとの最大値もチェックします。

引数：

gday - gDay 型データの文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 gday に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 gday に指定した gDay 型データの文字列表現が不正な場合にスローされます。［エラーコード：25000］

(2) toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

gDay 型データの文字列表現を取得します。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

この gDay 型データの文字列表現を返します。

(3) equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLGDay オブジェクトとほかのオブジェクトが等しいかどうかを調べます。
タイムゾーンを含み同じ文字列表現となる場合だけ，等しいものとみなします。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - この XBRLGDay オブジェクトと比較するオブジェクトを指定します。

戻り値：

XBRLGDay オブジェクトが引数 obj で指定したオブジェクトと等しい場合は true，
等しくない場合は false を返します。

(4) hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLGDay オブジェクトのハッシュコードを取得します。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

この XBRLGDay オブジェクトのハッシュコードを返します。

6.9.11 XBRLGMonth クラス

```
public final class XBRLGMonth
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML Schema のデータ型 gMonth を表します。

コンストラクタの概要：

XBRLGMonth クラスのコンストラクタの概要を次の表に示します。

表 6-142 XBRLGMonth クラスのコンストラクタの概要

説明	XBRL のバージョン
XBRLGMonth(java.lang.String gmonth) 指定した gMonth 型データの文字列表現から，XBRLGMonth オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLGMonth クラスのメソッドの概要を次の表に示します。

表 6-143 XBRLGMonth クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	toString() gMonth 型データの文字列表現を取得します。	2.0 および 2.1
2	boolean	equals(java.lang.Object obj) XBRLGMonth オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.0 および 2.1

項番	説明		XBRL のバージョン
3	int	hashCode() XBRLGMonth オブジェクトのハッシュコードを取得します。	2.0 および 2.1

(1) XBRLGMonth

```
public  XBRLGMonth(java.lang.String gmonth)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した gMonth 型データの文字列表現から，XBRLGMonth オブジェクトを構築します。

次の文字列表現だけ指定できます。文字列表現が不正な場合は例外がスローされます。

指定できる文字列表現：BNF 表記

```
gMonthFormat ::= '-''-'(month | month timezone)
month ::= ([0][1-9] | [1][0-2])('-''-')?
timezone ::= 'Z' | ('+'|'-') hour ':' minute
hour ::= [0-1][0-9] | [2][0-3]
minute ::= [0-5][0-9]
```

引数：

gmonth - gMonth 型データの文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 gmonth に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 gmonth に指定した gMonth 型データの文字列表現が不正な場合にスローされます。［エラーコード：25000］

(2) toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

gMonth 型データの文字列表現を取得します。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

この gMonth 型データの文字列表現を返します。

(3) equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLGMonth オブジェクトとほかのオブジェクトが等しいかどうかを調べます。タイムゾーンを含み同じ文字列表現となる場合だけ，等しいものとみなします。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - この XBRLGMonth オブジェクトと比較するオブジェクトを指定します。

戻り値：

この XBRLGMonth オブジェクトが，引数 obj で指定したオブジェクトと等しい場合は true，等しくない場合は false を返します。

(4) hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLGMonth オブジェクトのハッシュコードを取得します。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

この XBRLGMonth オブジェクトのハッシュコードを返します。

6.9.12 XBRLGMonthDay クラス

```
public final class XBRLGMonthDay
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML Schema のデータ型 gMonthDay を表します。

コンストラクタの概要：

XBRLGMonthDay クラスのコンストラクタの概要を次の表に示します。

表 6-144　XBRLGMonthDay クラスのコンストラクタの概要

説明	XBRL のバージョン
XBRLGMonthDay(java.lang.String gmonthday) 指定した gMonthDay 型データの文字列表現から，XBRLGMonthDay オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLGMonthDay クラスのメソッドの概要を次の表に示します。

表 6-145　XBRLGMonthDay クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	toString() gMonthDay 型データの文字列表現を取得します。	2.0 および 2.1
2	boolean	equals(java.lang.Object obj) XBRLGMonthDay オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.0 および 2.1
3	int	hashCode() XBRLGMonthDay オブジェクトのハッシュコードを取得します。	2.0 および 2.1

(1) XBRLGMonthDay

```
public  XBRLGMonthDay(java.lang.String gmonthday)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した gMonthDay 型データの文字列表現から，XBRLGMonthDay オブジェクトを構築します。

次の文字列表現だけ指定できます。文字列表現が不正な場合は例外がスローされます。

指定できる文字列表現：BNF 表記

```
gMonthDayFormat ::= monthday | monthday timezone
monthday ::= '-' '-' month '-' day
month ::=  [0][1-9] | [1][0-2]
day ::= [0][1-9] | [1-2][0-9] | [3][0-1]
timezone ::= 'Z' | ('+'|'-') hour ':' minute
hour ::= [0-1][0-9] | [2][0-3]
minute ::= [0-5][0-9]
```

day については，BNF 表記で表現できない月ごとの最大値もチェックします。

引数：

gmonthday - gMonthDay 型データの文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 gmonthday に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 gmonthday に指定した gMonthDay 型データの文字列表現が不正な場合にスローされます。［エラーコード：25000］

（2） toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

gMonthDay 型データの文字列表現を取得します。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

gMonthDay 型データの文字列表現を返します。

（3） equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLGMonthDay オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

タイムゾーンを含み同じ文字列表現となる場合だけ，等しいものとみなします。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - この XBRLGMonthDay オブジェクトと比較するオブジェクトを指定します。

戻り値：

この XBRLGMonthDay オブジェクトが，引数 obj で指定したオブジェクトと等しい場合は true，等しくない場合は false を返します。

（4） hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLGMonthDay オブジェクトのハッシュコードを取得します。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

XBRLGMonthDay オブジェクトのハッシュコードを返します。

6.9.13 XBRLGYear クラス

```
public final class XBRLGYear
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML Schema のデータ型 gYear を表します。

コンストラクタの概要：

XBRLGYear クラスのコンストラクタの概要を次の表に示します。

表 6-146 XBRLGYear クラスのコンストラクタの概要

説明	XBRL のバージョン
XBRLGYear(java.lang.String gyear) 指定した gYear 型データの文字列表現から，XBRLGYear オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLGYear クラスのメソッドの概要を次の表に示します。

表 6-147 XBRLGYear クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	toString() gYear 型データの文字列表現を取得します。	2.0 および 2.1
2	boolean	equals(java.lang.Object obj) XBRLGYear オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.0 および 2.1
3	int	hashCode() XBRLGYear オブジェクトのハッシュコードを取得します。	2.0 および 2.1

(1) XBRLGYear

```
public XBRLGYear(java.lang.String gyear)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した gYear 型データの文字列表現から，XBRLGYear オブジェクトを構築します。

次の文字列表現だけ指定できます。文字列表現が不正な場合は例外がスローされます。

指定できる文字列表現：BNF 表記

```
gYearFormat ::= ('-')?(year | year timezone)
year ::= ([0-9])*[0-9][0-9][0-9][0-9]
timezone ::= 'Z' | ('+'|'-') hour ':' minute
hour ::= [0-1][0-9] | [2][0-3]
minute ::= [0-5][0-9]
```

year には「0」を指定できません。

引数：

gyear - gYear 型データの文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 gyear に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 gyear に指定した gYear 型データの文字列表現が不正な場合にスローされます。［エラーコード：25000］

(2) toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

gYear 型データの文字列表現を取得します。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

gYear 型データの文字列表現を返します。

(3) equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLGYear オブジェクトとほかのオブジェクトが等しいかどうかを調べます。
タイムゾーンを含み同じ文字列表現となる場合だけ，等しいものとみなします。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - この XBRLGYear オブジェクトと比較するオブジェクトを指定します。

戻り値：

この XBRLGYear オブジェクトが，引数 obj で指定したオブジェクトと等しい場合は true，等しくない場合は false を返します。

(4) hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLGYear オブジェクトのハッシュコードを取得します。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

XBRLGYear オブジェクトのハッシュコードを返します。

6.9.14 XBRLGYearMonth クラス

```
public final class XBRLGYearMonth
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML Schema のデータ型 gYearMonth を表します。

コンストラクタの概要：

XBRLGYearMonth クラスのコンストラクタの概要を次の表に示します。

表 6-148　XBRLGYearMonth クラスのコンストラクタの概要

説明	XBRL のバージョン
XBRLGYearMonth(java.lang.String gyearmonth) 指定した gYearMonth 型データの文字列表現から，XBRLGYearMonth オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLGYearMonth クラスのメソッドの概要を次の表に示します。

表 6-149　XBRLGYearMonth クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	toString() gYearMonth 型データの文字列表現を取得します。	2.0 および 2.1
2	boolean	equals(java.lang.Object obj) XBRLGYearMonth オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.0 および 2.1
3	int	hashCode() XBRLGYearMonth オブジェクトのハッシュコードを取得します。	2.0 および 2.1

（1）XBRLGYearMonth

```
public  XBRLGYearMonth(java.lang.String gyearmonth)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した gYearMonth 型データの文字列表現から，XBRLGYearMonth オブジェクトを構築します。

次の文字列表現だけ指定できます。文字列表現が不正な場合は例外がスローされます。

指定できる文字列表現：BNF 表記

```
gYearMonthFormat ::= ('-')?(yearmonth | yearmonth timezone)
yearmonth ::= year '-' month
year ::= ([0-9])*[0-9][0-9][0-9][0-9]
month ::= [0][1-9] | [1][0-2]
timezone ::= 'Z' | ('+'|'-') hour ':' minute
hour ::= [0-1][0-9] | [2][0-3]
minute ::= [0-5][0-9]
```

year には「0」を指定できません。

引数：

gyearmonth - gYearMonth 型データの文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 gyearmonth に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 gyearmonth に指定した gYearMonth 型データの文字列表現が不正な場合にスローされます。［エラーコード：25000］

（2） toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

gYearMonth 型データの文字列表現を取得します。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

gYearMonth 型データの文字列表現を返します。

（3） equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLGYearMonth オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

タイムゾーンを含み同じ文字列表現となる場合だけ，等しいものとみなします。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - この XBRLGYearMonth オブジェクトと比較するオブジェクトを指定します。

戻り値：

この XBRLGYearMonth オブジェクトが，引数 obj で指定したオブジェクトと等しい場合は true，等しくない場合は false を返します。

（4） hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLGYearMonth オブジェクトのハッシュコードを取得します。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

この XBRLGYearMonth オブジェクトのハッシュコードを返します。

6.9.15 XBRLHexBinary クラス

```
public final class XBRLHexBinary
extends XBRLAbstractBinary
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML Schema のデータ型 hexBinary を表します。

hexBinary 型は，バイナリデータを 16 進数で表記したものです。

コンストラクタの概要：

XBRLHexBinary クラスのコンストラクタの概要を次の表に示します。

表 6-150 XBRLHexBinary クラスのコンストラクタの概要

項番	説明	XBRL のバージョン
1	XBRLHexBinary(java.lang.String hex) 指定した 16 進数の文字列から，XBRLHexBinary オブジェクトを構築します。	2.0 および 2.1
2	XBRLHexBinary(byte[] bytes) 指定したバイト配列から，XBRLHexBinary オブジェクトを構築します。	2.0 および 2.1
3	XBRLHexBinary(byte[] bytes, int offset, int length) 指定したバイト配列の一部から，XBRLHexBinary オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLHexBinary クラスのメソッドの概要を次の表に示します。

表 6-151　XBRLHexBinary クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	static byte[]	toByteArray(java.lang.String hex) 指定した 16 進数の文字列からバイト配列を生成します。	2.0 および 2.1
2	java.lang.String	toString() XBRLHexBinary オブジェクトが保持するバイト配列の，16 進数での表記を取得します。	2.0 および 2.1
3	static java.lang.String	toString(byte[] bytes) 指定したバイト配列の，16 進数での表記を取得します。	2.0 および 2.1
4	static java.lang.String	toString(byte[] bytes, int offset, int length) 指定したバイト配列の一部の，16 進数での表記を取得します。	2.0 および 2.1
5	static XBRLHexBinary	valueOf(java.lang.String hex) 指定した 16 進数の文字列から，XBRLHexBinary オブジェクトを取得します。	2.0 および 2.1
6	static XBRLHexBinary	valueOf(byte[] bytes) 指定したバイト配列から，XBRLHexBinary オブジェクトを取得します。	2.0 および 2.1
7	static XBRLHexBinary	valueOf(byte[] bytes, int offset, int length) 指定したバイト配列の一部から，XBRLHexBinary オブジェクトを取得します。	2.0 および 2.1

(1) XBRLHexBinary

```
public  XBRLHexBinary(java.lang.String hex)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した 16 進数の文字列から，XBRLHexBinary オブジェクトを構築します。

引数：

hex - 16 進数の文字列を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 hex に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 hex に指定した文字列が 16 進数として解釈できない場合にスローされます。［エラーコード：25004，25006，25007］

(2) XBRLHexBinary

```
public  XBRLHexBinary(byte[] bytes)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定したバイト配列から，XBRLHexBinary オブジェクトを構築します。指定したバイト配列はコピーされます。

引数：

bytes - バイト配列を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 bytes に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 bytes に長さ 0 の配列を指定した場合にスローされます。［エラーコード：25007］

（3） XBRLHexBinary

```
public  XBRLHexBinary(byte[] bytes,
                      int offset,
                      int length)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定したバイト配列の一部から，XBRLHexBinary オブジェクトを構築します。バイト配列の指定した部分の内容がコピーされます。

引数：

bytes - バイト配列を指定します。

offset - 読み込み開始位置のインデックスを指定します。

length - 読み込むバイト数を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 bytes に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 offset および length の指定によるインデックスが配列の境界を超える場合にスローされます。［エラーコード：25001，25003，25010］
- 引数 bytes に長さ 0 の配列を指定した場合にスローされます。［エラーコード：25007］

（4） toByteArray

```
public static byte[] toByteArray(java.lang.String hex)
```

XBRL のバージョン：

2.0 および 2.1

説明：
指定した 16 進数の文字列からバイト配列を生成します。

引数：
hex - 16 進数の文字列を指定します。

戻り値：
指定した 16 進数の文字列から生成されたバイト配列を返します。

例外：
XBRLDOMRuntimeException -
- 引数 hex に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 hex に指定した文字列が 16 進数として解釈できない場合にスローされます。［エラーコード：25004，25006，25007］

（5） toString

```
public java.lang.String toString()
```

XBRL のバージョン：
2.0 および 2.1

説明：
XBRLHexBinary オブジェクトが保持するバイト配列の，16 進数での表記を取得します。

定義：
クラス XBRLAbstractBinary 内の toString()

戻り値：
XBRLHexBinary オブジェクトが保持するバイト配列の 16 進数による表記を返します。返却される文字列は，必ずアルファベットが大文字になります。

（6） toString

```
public static java.lang.String toString(byte[] bytes)
```

XBRL のバージョン：
2.0 および 2.1

説明：
指定したバイト配列の，16 進数での表記を取得します。

引数：
bytes - バイト配列を指定します。

戻り値：
指定したバイト配列の 16 進数による表記を返します。

例外：

XBRLDOMRuntimeException -

- 引数 bytes に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 bytes に長さ 0 の配列を指定した場合にスローされます。［エラーコード：25007］

（7） toString

```
public static java.lang.String toString(byte[] bytes,
                                        int offset,
                                        int length)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定したバイト配列の一部の，16 進数での表記を取得します。

引数：

bytes - バイト配列を指定します。

offset - 読み込み開始位置のインデックスを指定します。

length - 読み込むバイト数を指定します。

戻り値：

指定したバイト配列の一部の 16 進数による表記を返します。

例外：

XBRLDOMRuntimeException -

- 引数 bytes に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 offset および length の指定によるインデックスが配列の境界を超える場合にスローされます。［エラーコード：25001，25003，25010］
- 引数 bytes に長さ 0 の配列を指定した場合にスローされます。［エラーコード：25007］

（8） valueOf

```
public static XBRLHexBinary valueOf(java.lang.String hex)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した 16 進数の文字列から，XBRLHexBinary オブジェクトを取得します。

引数：

hex - 16 進数の文字列を指定します。

戻り値：

指定した 16 進数の文字列から生成された XBRLHexBinary オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 hex に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 hex に指定した文字列が 16 進数として解釈できない場合にスローされます。［エラーコード：25004，25006，25007］

(9) valueOf

```
public static XBRLHexBinary valueOf(byte[] bytes)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定したバイト配列から，XBRLHexBinary オブジェクトを取得します。指定したバイト配列はコピーされます。

引数：

bytes - バイト配列を指定します。

戻り値：

指定したバイト配列から生成された XBRLHexBinary オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 bytes に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 bytes に長さ 0 の配列を指定した場合にスローされます。［エラーコード：25007］

(10) valueOf

```
public static XBRLHexBinary valueOf(byte[] bytes,
                                    int offset,
                                    int length)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定したバイト配列の一部から，XBRLHexBinary オブジェクトを取得します。バイト配列の指定した部分の内容がコピーされます。

引数：

bytes - バイト配列を指定します。

offset - 読み込み開始位置のインデックスを指定します。

length - 読み込むバイト数を指定します。

戻り値：

指定したバイト配列の一部から生成された XBRLHexBinary オブジェクトを返しま

す。

例外：

XBRLDOMRuntimeException -

- 引数 bytes に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 offset および length の指定によるインデックスが配列の境界を超える場合にスローされます。［エラーコード：25001，25003，25010］
- 引数 bytes に長さ 0 の配列を指定した場合にスローされます。［エラーコード：25007］

6.9.16 XBRLPrecision クラス

```
public final class XBRLPrecision
extends XBRLAbstractPrecision
```

XBRL のバージョン：

2.1

説明：

数値アイテムの precision 属性に指定する値を表します。0 以上の整数，または文字列 "INF" を保持します。

コンストラクタの概要：

XBRLPrecision クラスのコンストラクタの概要を次の表に示します。

表 6-152 XBRLPrecision クラスのコンストラクタの概要

項番	説明	XBRL のバージョン
1	XBRLPrecision(java.math.BigInteger value) 指定した数値から XBRLPrecision オブジェクトを構築します。	2.1
2	XBRLPrecision(java.lang.String precision) 指定した precision 属性の値の文字列表現から XBRLPrecision オブジェクトを構築します。	2.1

メソッドの概要：

XBRLPrecision クラスのメソッドの概要を次の表に示します。

表 6-153 XBRLPrecision クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	boolean	equals(java.lang.Object obj) XBRLPrecision オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.1
2	int	hashCode() XBRLPrecision オブジェクトのハッシュコードを取得します。	2.1

(1) XBRLPrecision

```
public  XBRLPrecision(java.math.BigInteger value)
```

XBRL のバージョン：

2.1

説明：

指定した数値から XBRLPrecision オブジェクトを構築します。

引数：

value - 数値を指定します。指定できる数値は 0 以上の整数です。

例外：

XBRLDOMRuntimeException -

- 引数 value に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 value の値に負数を指定した場合にスローされます。[エラーコード：25001]

(2) XBRLPrecision

```
public  XBRLPrecision(java.lang.String precision)
```

XBRL のバージョン：

2.1

説明：

指定した precision 属性の値の文字列表現から，XBRLPrecision オブジェクトを構築します。引数 precision には，0 以上の整数を表す文字列（XML Schema のデータ型 nonNegativeInteger），または文字列 "INF" を指定できます。

引数 precision に数値の文字列表現が指定された場合，構築されるオブジェクトは文字列表現を java.math.BigInteger に変換した値を保持します。このためオブジェクト構築時に指定された文字列表現と，toString() によって取得した文字列表現が異なることがあります。

引数：

precision - precision 属性の値の文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 precision に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 precision に指定された precision 属性の値の文字列表現が不正な場合にスローされます。[エラーコード：25000]

(3) equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.1

説明：

XBRLPrecision オブジェクトとほかのオブジェクトが等しいかどうかを調べます。結果が true になるのは，このオブジェクトと文字列表現が等しい XBRLPrecision オブジェクトの場合だけです。

定義：

クラス XBRLAbstractPrecision 内の equals()

引数：

obj - この XBRLPrecision オブジェクトと比較するオブジェクトを指定します。

戻り値：

XBRLPrecision オブジェクトが，引数 obj で指定されたオブジェクトと等しい場合は true，等しくない場合は false を返します。

（4） hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.1

説明：

XBRLPrecision オブジェクトのハッシュコードを取得します。

定義：

クラス XBRLAbstractPrecision 内の hashCode()

戻り値：

この XBRLPrecision オブジェクトのハッシュコードを返します。

6.9.17 XBRLQName クラス

```
public final class XBRLQName
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML Schema のデータ型 QName を表します。QName 型は，Namespaces in XML で定義されている修飾名（Qualified Name）を表します。修飾名は，名前空間接頭辞とローカルパートから構成されますが，XBRLQName クラスでは，名前空間接頭辞を保持しません。代わりに，名前空間接頭辞に関連づけられる名前空間 URI を保持します。

コンストラクタの概要：

XBRLQName クラスのコンストラクタの概要を次の表に示します。

表 6-154　XBRLQName クラスのコンストラクタの概要

項番	説明	XBRL のバージョン
1	XBRLQName(java.lang.String qname) 指定した QName の文字列表現から XBRLQName オブジェクトを構築します。	2.0 および 2.1
2	XBRLQName(java.lang.String namespaceURI, java.lang.String localPart) 指定した名前空間 URI およびローカルパートで構成される QName を表す XBRLQName オブジェクトを構築します。	2.0 および 2.1
3	XBRLQName(java.lang.String qname, boolean validate) 指定した QName の文字列表現から XBRLQName オブジェクトを構築します。	2.0 および 2.1
4	XBRLQName(java.lang.String namespaceURI, java.lang.String localPart, boolean validate) 指定した名前空間 URI およびローカルパートで構成される QName を表す XBRLQName オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLQName クラスのメソッドの概要を次の表に示します。

表 6-155　XBRLQName クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getNamespaceURI() QName の名前空間 URI を取得します。	2.0 および 2.1
2	java.lang.String	getLocalPart() QName のローカルパートを取得します。	2.0 および 2.1
3	java.lang.String	toString() QName の文字列表現を取得します。	2.0 および 2.1
4	static java.lang.String	toString(java.lang.String namespaceURI, java.lang.String localPart) 指定した名前空間 URI およびローカルパートで構成される，QName の文字列表現を取得します。	2.0 および 2.1
5	static XBRLQName	valueOf(java.lang.String qname) 指定した QName の文字列表現から XBRLQName オブジェクトを取得します。	2.0 および 2.1
6	static XBRLQName	valueOf(java.lang.String namespaceURI, java.lang.String localPart) 指定した名前空間 URI とローカルパートで構成される QName を表す XBRLQName オブジェクトを取得します。	2.0 および 2.1
7	boolean	equals(java.lang.Object obj) XBRLQName オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.0 および 2.1

項番	説明		XBRL のバージョン
8	int	hashCode() XBRLQName オブジェクトのハッシュコードを取得します。	2.0 および 2.1

(1) XBRLQName

```
public  XBRLQName(java.lang.String qname)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した QName の文字列表現から XBRLQName オブジェクトを構築します。
QName の文字列表現は，"{namespaceURI}localPart" です。"{namespaceURI}" の部分は省略できます。
このコンストラクタでは，引数 qname で指定した QName の文字列表現に不正な文字が含まれるかどうかはチェックしません。

引数：

qname - QName の文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 qname に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 qname に指定した QName の文字列表現が不正なフォーマットの場合にスローされます。[エラーコード：25000]

(2) XBRLQName

```
public  XBRLQName(java.lang.String namespaceURI,
                  java.lang.String localPart)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した名前空間 URI およびローカルパートで構成される QName を表す XBRLQName オブジェクトを構築します。
このコンストラクタでは，引数 namespaceURI および localPart で指定した文字列に，不正な文字が含まれるかどうかはチェックしません。

引数：

namespaceURI - QName の名前空間 URI を指定します。名前空間 URI がない場合は，null を指定します。
localPart - QName のローカルパートを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 localPart に null または空文字列（""）を指定した場合にスローされます。［エラーコード：20002，25000］

(3) XBRLQName

```
public  XBRLQName(java.lang.String qname,
                  boolean validate)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した QName の文字列表現から XBRLQName オブジェクトを構築します。QName の文字列表現は "{namespaceURI}localPart" です。"{namespaceURI}" の部分は省略できます。

引数 validate に true を指定した場合だけ，QName の文字列表現に不正な文字が含まれるかどうかをチェックします。その場合，namespaceURI および localPart には，それぞれ XML Schema のデータ型 anyURI および NCName として妥当な文字列だけを指定できます。ただし namespaceURI に空文字列（""）は指定できません。

引数：

qname - QName の文字列表現を指定します。

validate - QName の文字列表現に不正な文字が含まれているかどうかをチェックする場合は，true を指定します。チェックしない場合は，false を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 qname に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 qname に指定された QName の文字列表現が不正なフォーマットの場合にスローされます。［エラーコード：25000］
- 引数 qname に指定された QName の文字列表現に不正な文字が含まれている場合にスローされます。［エラーコード：25008，25009］

(4) XBRLQName

```
public  XBRLQName(java.lang.String namespaceURI,
                  java.lang.String localPart,
                  boolean validate)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した名前空間 URI およびローカルパートで構成される QName を表す XBRLQName オブジェクトを構築します。

引数 validate に true が指定された場合だけ，名前空間 URI とローカルパートに指定された文字列に不正な文字が含まれるかどうかをチェックします。その場合，引数 namespaceURI および localPart には，それぞれ XML Schema のデータ型 anyURI および NCName として妥当な文字列だけを指定できます。ただし，引数 namespaceURI に空文字列（""）は指定できません。

引数：

namespaceURI - QName の名前空間 URI を指定します。名前空間 URI がない場合は，null を指定します。

localPart - QName のローカルパートを指定します。

validate - 名前空間 URI およびローカルパートに指定された文字列に不正な文字が含まれているかどうかをチェックする場合は，true を指定します。チェックしない場合は，false を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 localPart に null または空文字列（""）を指定した場合にスローされます。［エラーコード：20002，25000］
- 引数 namespaceURI に指定した名前空間 URI，または引数 localPart に指定したローカルパートに不正な文字が含まれている場合にスローされます。［エラーコード：25008，25009］

（5） getNamespaceURI

```
public java.lang.String getNamespaceURI()
```

XBRL のバージョン：

2.0 および 2.1

説明：

QName の名前空間 URI を取得します。

戻り値：

QName の名前空間 URI を返します。名前空間 URI がない場合は null を返します。

（6） getLocalPart

```
public java.lang.String getLocalPart()
```

XBRL のバージョン：

2.0 および 2.1

説明：

QName のローカルパートを取得します。

戻り値：

QName のローカルパートを返します。

(7) toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

QName の文字列表現を取得します。QName の文字列表現は，
"{namespaceURI}localPart" です。"{namespaceURI}" の部分は省略できます。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

QName の文字列表現を返します。

(8) toString

```
public static java.lang.String toString(java.lang.String
namespaceURI,
                                        java.lang.String localPart)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した名前空間 URI およびローカルパートで構成される，QName の文字列表現を取得します。QName の文字列表現は，"{namespaceURI}localPart" です。
"{namespaceURI}" の部分は省略できます。
このメソッドでは，引数 namespaceURI および localPart で指定した文字列に不正な文字が含まれるかどうかはチェックしません。

引数：

namespaceURI -QName の名前空間 URI を指定します。名前空間 URI がない場合は，null を指定します。
localPart - QName のローカルパートを指定します。

戻り値：

指定した名前空間 URI とローカルパートで構成される QName の文字列表現を返します。

例外：

XBRLDOMRuntimeException -

- 引数 localPart に null または空文字列（""）を指定した場合にスローされます。
  ［エラーコード：20002，25000］

(9) valueOf

```
public static XBRLQName valueOf(java.lang.String qname)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した QName の文字列表現から XBRLQName オブジェクトを取得します。
QName の文字列表現は，"{namespaceURI}localPart" です。"{namespaceURI}" の部分は省略できます。
このメソッドでは，引数 qname で指定した QName の文字列表現に不正な文字が含まれるかどうかはチェックしません。

引数：

qname - QName の文字列表現を指定します。

戻り値：

指定した QName の文字列表現から生成された XBRLQName オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 qname に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 qname に指定した QName の文字列表現が不正なフォーマットの場合にスローされます。［エラーコード：25000］

(10) valueOf

```
public static XBRLQName valueOf(java.lang.String namespaceURI,
                                java.lang.String localPart)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した名前空間 URI とローカルパートで構成される QName を表す XBRLQName オブジェクトを取得します。
このメソッドでは，引数 namespaceURI および localPart で指定した文字列に，不正な文字が含まれるかどうかはチェックしません。

引数：

namespaceURI - QName の名前空間 URI を指定します。名前空間 URI がない場合は，null を指定します。
localPart - QName のローカルパートを指定します。

戻り値：

指定した名前空間 URI とローカルパートから生成された XBRLQName オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 localPart に null または空文字列（""）を指定した場合にスローされます。
  ［エラーコード：20002，25000］

（11）equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLQName オブジェクトとほかのオブジェクトが等しいかどうかを調べます。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - この XBRLQName オブジェクトと比較するオブジェクトを指定します。

戻り値：

この XBRLQName オブジェクトが，引数 obj で指定したオブジェクトと等しい場合は true，等しくない場合は false を返します。

（12）hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLQName オブジェクトのハッシュコードを取得します。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

この XBRLQName オブジェクトのハッシュコードを返します。

6.9.18　XBRLTime クラス

```
public final class XBRLTime
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

XML Schema のデータ型 time を表します。

注　このクラスは equals() と一貫性のない，自然順序づけを持ちます。

コンストラクタの概要：

XBRLTime クラスのコンストラクタの概要を次の表に示します。

表 6-156　XBRLTime クラスのコンストラクタの概要

説明	XBRL のバージョン
XBRLTime(java.lang.String time) 指定した time 型データの文字列表現から XBRLTime オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLTime クラスのメソッドの概要を次の表に示します。

表 6-157　XBRLTime クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	int	compareTo(XBRLTime other) XBRLTime オブジェクトとほかの XBRLTime オブジェクトを比較します。	2.0 および 2.1
2	java.lang.String	toString() time 型データの文字列表現を取得します。	2.0 および 2.1
3	boolean	equals(java.lang.Object obj) XBRLTime オブジェクトとほかのオブジェクトが等しいかどうかを調べます。	2.0 および 2.1
4	int	hashCode() XBRLTime オブジェクトのハッシュコードを取得します。	2.0 および 2.1

(1) XBRLTime

```
public  XBRLTime(java.lang.String time)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した time 型データの文字列表現から XBRLTime オブジェクトを構築します。次の文字列表現だけ指定できます。文字列表現が不正な場合は例外がスローされます。

指定できる文字列表現：BNF 表記

```
timeFormat ::= time | time timezone)
time ::= hour ':' minute ':' second
timezone ::= 'Z' | ('+'|'-') hour ':' minute
hour ::= [0-1][0-9] | [2][0-3]
minute ::= [0-5][0-9]
second ::= [0-5][0-9] | [0-5][0-9]'.'([0-9])+
```

秒以下の値は，小数第 4 桁で四捨五入しミリ秒単位に丸めます。このためオブジェクト構築時に指定した文字列表現と，toString() によって取得した文字列表現が異なることがあります。

引数：

time - time 型データの文字列表現を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 time に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 time に指定した time 型データの文字列表現が不正な場合にスローされます。［エラーコード：25000］

(2) compareTo

```
public int compareTo(XBRLTime other)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLTime オブジェクトとほかの XBRLTime オブジェクトを比較します。
タイムゾーンオフセットを反映した結果で比較します。タイムゾーン指定のないものについては UTC（協定世界時）と仮定し，比較します。
注　このメソッドの実行結果と equals() とは一貫性がありません。

引数：

other - 比較対象の XBRLTime オブジェクトを指定します。

戻り値：

この XBRLTime オブジェクトが，引数 other で指定した XBRLTime オブジェクトと等しい場合は 0 を返します。引数 other で指定した XBRLTime オブジェクトよりも小さい場合は 0 より小さい値，大きい場合は 0 より大きい値を返します。

(3) toString

```
public java.lang.String toString()
```

XBRL のバージョン：

2.0 および 2.1

説明：

time 型データの文字列表現を取得します。
秒以下の値は必ず小数第 3 桁で表現します。このためオブジェクト構築時に指定した文字列表現と，toString() によって取得した文字列表現が異なることがあります。

オーバーライド：

クラス java.lang.Object 内の toString()

戻り値：

この time 型データの文字列表現を返します。

(4) equals

```
public boolean equals(java.lang.Object obj)
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLTime オブジェクトとほかのオブジェクトが等しいかどうかを調べます。
タイムゾーンを含み同じ文字列表現となる場合だけ，等しいものとみなします。タイムゾーンオフセットを反映した結果が同じ値となる場合でも，タイムゾーンの表現方法が別表現の場合は異なるものとみなします。
注　このメソッドの実行結果と compareTo() とは一貫性がありません。

オーバーライド：

クラス java.lang.Object 内の equals()

引数：

obj - この XBRLTime オブジェクトと比較するオブジェクトを指定します。

戻り値：

この XBRLTime オブジェクトが，引数 obj で指定したオブジェクトと等しい場合は true，等しくない場合は false を返します。

(5) hashCode

```
public int hashCode()
```

XBRL のバージョン：

2.0 および 2.1

説明：

XBRLTime オブジェクトのハッシュコードを取得します。

オーバーライド：

クラス java.lang.Object 内の hashCode()

戻り値：

XBRLTime オブジェクトのハッシュコードを返します。

6.10 com.hitachi.xbrl.util パッケージ

com.hitachi.xbrl.util パッケージは，XBRL DOM で利用できるユーティリティを提供します。

インターフェースの概要：

com.hitachi.xbrl.util パッケージのインターフェースを次の表に示します。

表 6-158 com.hitachi.xbrl.util パッケージのインターフェース

項番	インターフェース	説明	XBRL のバージョン
1	XBRLIdCreator	XBRL 文書の ID を生成するためのインターフェースです。	2.1
2	XBRLDocumentWriter	XML 文書をストリームへ出力するインターフェースです。	2.1
3	XBRLDocumentWriterFactory	XBRLDocumentWriter オブジェクトを生成するための API を定義するインターフェースです。	2.1

クラスの概要：

com.hitachi.xbrl.util パッケージのクラス一覧を次の表に示します。

表 6-159 com.hitachi.xbrl.util パッケージのクラス一覧

項番	クラス	説明	XBRL のバージョン
1	XBRLConstants	ユーザーアプリケーションで利用できる定数を定義します。	2.0 および 2.1
2	XBRLIdManager	XBRL 文書に記述された ID を管理します。	2.1
3	XBRLNamespaceDeclaration	名前空間接頭辞と対応する名前空間 URI の組を管理します。	2.0 および 2.1
4	XBRLSchemaLocation	xsi:schemaLocation 属性に指定する，名前空間 URI とスキーマ文書の位置を示す URI の組を管理します。	2.0 および 2.1
5	PTVInstance クラス	PTVI の情報を保持します。	2.1
6	PTVLinkbase クラス	PTVLI の情報を保持します。	2.1
7	XBRLEqualUtil クラス	項目とタプルの重複検出に関連する同等判定を可能にする機能を提供します。	2.1

6.10.1 XBRLIdCreator インターフェース

```
public interface XBRLIdCreator
```

XBRL のバージョン：

2.1

説明：

XBRL 文書の ID を生成するためのインターフェースです。

XBRL 文書の編集時に自動採番される ID を，ユーザーアプリケーションで生成したい場合は，このインターフェースを実装したクラスを用意する必要があります。

このインターフェースを実装したクラスのオブジェクトを XBRL 文書の ID を管理する XBRLIdManager オブジェクトに設定すると，ID の自動採番時にコールバックされ，ユーザーアプリケーションで ID を生成できます。

フィールドの概要：

XBRLIdCreator インターフェースのフィールドの概要を次の表に示します。

表 6-160　XBRLIdCreator インターフェースのフィールドの概要

項番	説明	
1	int	FOOTNOTE_LABEL <link:footnote> 要素の xlink:label 属性に設定する値を生成することを示します。
2	int	LOC_LABEL <link:loc> 要素の xlink:label 属性に設定する値を生成することを示します。
3	int	ITEM_ID アイテムの id 属性に設定する値を生成することを示します。
4	int	TUPLE_ID タプルの id 属性に設定する値を生成することを示します。
5	int	XBRL_ID <xbrli:xbrl> 要素の id 属性に設定する値を生成することを示します。
6	int	CONTEXT_ID <xbrli:context> 要素の id 属性に設定する値を生成することを示します。
7	int	UNIT_ID <xbrli:unit> 要素の id 属性に設定する値を生成することを示します。
8	int	FOOTNOTE_LINK_ID <link:footnoteLink> 要素の id 属性に設定する値を生成することを示します。
9	int	FOOTNOTE_ID <link:footnote> 要素の id 属性に設定する値を生成することを示します。

メソッドの概要：

XBRLIdCreator クラスのメソッドの概要を次の表に示します。

表 6-161　XBRLIdCreator クラスのメソッドの概要

説明		XBRL のバージョン
java.lang.String	createId(int type, XBRLIdManager idManager) ID を生成します。	2.1

(1) createId

```
public java.lang.String createId(int type,
                                 XBRLIdManager idManager)
```

XBRL のバージョン：

2.1

説明：

ID を生成します。

インスタンス文書のアイテムや脚注リンクの <link:loc> 要素などの生成時に，id 属性や xlink:label 属性などの値を自動生成する前に呼び出されます。このメソッドの戻り値（文字列）を，id 属性や xlink:label 属性などの値として使用します。

XBRLInstanceDocument.createItem() メソッドや XBRLInstanceDocument.addFootnote() メソッドなどの延長でこのメソッドが呼び出され，戻り値は引数 idManager の XBRLIdManager オブジェクトで管理されます。

このメソッドで採番した ID が不正な場合や実行時例外がスローされた場合には，uCosminexus Business Reporting Processor によって自動採番し直された ID が使用されます。

引数：

type - ID の自動生成が必要な要素の属性を示す値を指定します。詳細を次の表に示します。

表 6-162　ID の自動生成が必要な要素と，type に指定する値

項番	type に指定する値	ID の自動生成が必要な要素と属性
1	XBRLIdCreator.FOOTNOTE_LABEL	<link:footnote>/xlink:label
2	XBRLIdCreator.LOC_LABEL	<link:loc>/xlink:label
3	XBRLIdCreator.ITEM_ID	アイテム型要素 /id
4	XBRLIdCreator.TUPLE_ID	タプル型要素 /id
5	XBRLIdCreator.XBRL_ID	<xbrl>/id
6	XBRLIdCreator.CONTEXT_ID	<context>/id
7	XBRLIdCreator.UNIT_ID	<unit>/id
8	XBRLIdCreator.FOOTNOTE_LINK_ID	<footnoteLink>/id
9	XBRLIdCreator.FOOTNOTE_ID	<footnote>/id

なお，このメソッドが XBRLIdManager オブジェクトから呼び出された場合，次の値が指定されることはありません。

- XBRLIdCreator.XBRL_ID
- XBRLIdCreator.FOOTNOTE_LINK_ID
- XBRLIdCreator.FOOTNOTE_ID

idManager - このメソッドを呼び出した XBRLIdManager オブジェクトを指定します。

戻り値：

各要素の id 属性など，XML 文書内に記述する ID を返します。返した ID が次の場合は，uCosminexus Business Reporting Processor によって自動採番し直された ID が使用されます。

- 引数 idManager の XBRLIdManager オブジェクト内で，一意でない場合
- null の場合
- 空文字列の場合
- xsd:NCName 型でない場合
- 実行時例外がスローされた場合

6.10.2 XBRLDocumentWriter インターフェース

```
public interface XBRLDocumentWriter
```

XBRL のバージョン：

2.1

説明：

XML 文書をストリームへ出力するインターフェースです。

XBRLDocumentWriter は，JAXP に準拠した XSLT プロセッサを使用して XML 文書をストリームへ出力します。

メソッドの概要：

XBRLDocumentWriter インターフェースのメソッドの概要を次の表に示します。

表 6-163 XBRLDocumentWriter インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLDocumentManager	getOwnerDocumentManager() XBRLDocumentWriter オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。	2.1
2	void	setEncoding(java.lang.String encoding) XML 文書を出力する際に使用するエンコーディングを設定します。	2.1

項番	説明		XBRL のバージョン
3	java.lang.String	getEncoding() XML 文書を出力する際に使用するエンコーディングを取得します。	2.1
4	void	setIndent(boolean indent) XML 文書を出力する際にインデントを設定するかどうかを指定します。	2.1
5	boolean	isIndent() XML 文書を出力する際にインデントを設定するよう設定されているかどうかを調べます。	2.1
6	void	write(java.io.OutputStream os, org.w3c.dom.Document doc) XML 文書をストリームへ出力します。	2.1
7	void	write(java.io.OutputStream os, XBRLInstanceDocument doc) インスタンス文書をストリームへ出力します。	2.1

(1) getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.1

説明：

XBRLDocumentWriter オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

戻り値：

XBRLDocumentWriter オブジェクトを所有する XBRLDocumentManager オブジェクトを返します。

(2) setEncoding

```
public void setEncoding(java.lang.String encoding)
```

XBRL のバージョン：

2.1

説明：

XML 文書を出力する際に使用するエンコーディングを設定します。このメソッドが呼び出されなかった場合は，デフォルトのエンコーディングである UTF-8 が使用されます。

XBRLDocumentWriter は，JAXP に準拠した XSLT プロセッサを使用して XML 文書をストリームへ出力します。そのため，このメソッドで指定できるエンコーディングは使用する XSLT プロセッサに依存しますが，uCosminexus Business

Reporting Processor では利用できるエンコーディングに制限はありません。そのため，このメソッドでは引数 encoding に null を指定しないかぎり，例外は発生しません。その代わり，XSLT プロセッサでサポートされないエンコーディングを指定した場合は，XML 文書の出力処理時に例外が発生するおそれがあります。

引数：

encoding - エンコーディングを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 encoding に null を指定した場合にスローされます。［エラーコード：20002］

（3） getEncoding

```
public java.lang.String getEncoding()
```

XBRL のバージョン：

2.1

説明：

XML 文書を出力する際に使用するエンコーディングを取得します。

戻り値：

XML 文書を出力する際に使用されるエンコーディングを返します。

（4） setIndent

```
public void setIndent(boolean indent)
```

XBRL のバージョン：

2.1

説明：

XML 文書を出力する際にインデントを設定するかどうかを指定します。このメソッドが呼び出されなかった場合はインデントは設定しません。インデントを設定するよう指定すると，XBRLDocumentWriter オブジェクトは XSLT プロセッサに対して indent プロパティを yes に設定してから出力処理を実行します。

引数：

indent - インデントを設定する場合は true，インデントを設定しない場合は false を指定します。

（5） isIndent

```
public boolean isIndent()
```

XBRL のバージョン：

2.1

説明：

XML 文書を出力する際にインデントを設定するよう設定されているかどうかを調べます。

戻り値：

XML 文書を出力する際にインデントを設定するよう指定されている場合は true，そうでない場合は false を返します。

(6) write

```
public void write(java.io.OutputStream os,
                  org.w3c.dom.Document doc)
            throws XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

XML 文書をストリームへ出力します。

引数：

os - XML 文書を出力するストリームを指定します。

doc - ストリームへ出力する XML 文書を指定します。

例外：

XBRLDOMException -

- 入出力エラーが発生した場合にスローされます。［エラーコード：20052］

XBRLDOMRuntimeException -

- 引数 os または doc に null を指定した場合にスローされます。［エラーコード：20002］

(7) write

```
public void write(java.io.OutputStream os,
                  XBRLInstanceDocument doc)
            throws XBRLDOMException
```

XBRL のバージョン：

2.1

説明：

インスタンス文書をストリームへ出力します。出力前にインスタンス文書の検証を実行することをお勧めします。

インスタンス文書のルート要素が XML 文書のルート要素ではない場合，インスタンス文書が埋め込まれている XML 文書全体をストリームへ出力します。インスタンス文書のルート要素が DOM ツリーにつながっていない場合は，XBRLDOMRuntimeException がスローされます。

引数：

os - インスタンス文書を出力するストリームを指定します。

doc - ストリームへ出力するインスタンス文書を指定します。

例外：

XBRLDOMException -

- 入出力エラーが発生した場合にスローされます。［エラーコード：20052］

XBRLDOMRuntimeException -

- 引数 os または doc に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 doc に指定された XBRLInstanceDocument オブジェクトを所有する XBRLDocumentManager オブジェクトが，この XBRLDocumentWriter オブジェクトを所有する XBRLDocumentManager オブジェクトと異なる場合にスローされます。［エラーコード：20051］
- 引数 doc に指定されたインスタンス文書のルート要素が DOM ツリーにつながっていない場合にスローされます。［エラーコード：20106］

6.10.3 XBRLDocumentWriterFactory インターフェース

```
public interface XBRLDocumentWriterFactory
```

XBRL のバージョン：

2.1

説明：

XBRLDocumentWriter オブジェクトを生成するための API を定義するインターフェースです。

メソッドの概要：

XBRLDocumentWriterFactory インターフェースのメソッドの概要を次の表に示します。

表 6-164 XBRLDocumentWriterFactory インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLDocumentManager	getOwnerDocumentManager() XBRLDocumentWriterFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。	2.1
2	XBRLDocumentWriter	newDocumentWriter() 現在設定されているパラメタを使用して XBRLDocumentWriter の新しいインスタンスを生成します。	2.1

(1) getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.1

説明：

XBRLDocumentWriterFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

戻り値：

XBRLDocumentWriterFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを返します。

(2) newDocumentWriter

```
public XBRLDocumentWriter newDocumentWriter()
```

XBRL のバージョン：

2.1

説明：

現在設定されているパラメタを使用して，XBRLDocumentWriter の新しいインスタンスを生成します。

戻り値：

新たに生成された XBRLDocumentWriter オブジェクトを返します。

6.10.4 XBRLConstants クラス

```
public abstract class XBRLConstants
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

ユーザーアプリケーションで利用できる定数を定義します。

フィールドの概要：

XBRLConstants クラスのフィールドの概要を次の表に示します。

表 6-165 XBRLConstants クラスのフィールドの概要

項番	説明	
1	static java.lang.String	XBRL_INSTANCE_NS_URI この定数は，名前空間 URI を表す文字列「http://www.xbrl.org/2001/instance」を保持します。この名前空間 URI は，XBRL2.0 では，インスタンス文書の名前空間を示すものとして定義されています。

項番	説明	
2	static java.lang.String	XBRL_LINKBASE_NS_URI この定数は，名前空間 URI を表す文字列「http://www.xbrl.org/2001/XLink/xbrllinkbase」を保持します。この名前空間 URI は，XBRL2.0 では，リンクベースの名前空間を示すものとして定義されています。
3	static java.lang.String	XBRL21_INSTANCE_NS_URI この定数は，名前空間 URI を表す文字列「http://www.xbrl.org/2003/instance」を保持します。この名前空間 URI は，XBRL2.1 では，インスタンス文書の名前空間を示すものとして定義されています。
4	static java.lang.String	XBRL21_LINKBASE_NS_URI この定数は，名前空間 URI を表す文字列「http://www.xbrl.org/2003/linkbase」を保持します。この名前空間 URI は，XBRL2.1 では，リンクベースの名前空間を示すものとして定義されています。
5	static java.lang.String	XBRL21_ISO4217_NS_URI この定数は，名前空間 URI を表す文字列「http://www.xbrl.org/2003/iso4217」を保持します。この名前空間 URI は，XBRL2.1 では，ISO4217 の通貨コードを記述するために定義されています。
6	static java.lang.String	XMLSCHEMA_NS_URI この定数は，名前空間 URI を表す文字列「http://www.w3.org/2001/XMLSchema」を保持します。この名前空間 URI は，W3C の勧告である XML Schema で規定されていて，スキーマ文書を記述するために使用されます。
7	static java.lang.String	XMLSCHEMA_INSTANCE_NS_URI この定数は，名前空間 URI を表す文字列「http://www.w3.org/2001/XMLSchema-instance」を保持します。この名前空間 URI は，W3C の勧告である XML Schema で規定されています。XML 文書でスキーマ文書の所在を示すために使用される schemaLocation などの属性が，この名前空間 URI に含まれます。
8	static java.lang.String	XLINK_NS_URI この定数は，名前空間 URI を表す文字列「http://www.w3.org/1999/xlink」を保持します。この名前空間 URI は，W3C の勧告である XLink 1.0 で規定されています。
9	static java.lang.String	XHTML_NS_URI この定数は，名前空間 URI を表す文字列「http://www.w3.org/1999/xhtml」を保持します。この名前空間 URI は，W3C の勧告である XHTML に従った文書を記述するために使用されます。XBRL では，ラベル（<link:label> 要素）や脚注（<link:footnote> 要素）の内容に XHTML を記述できます。
10	static java.lang.String	XML_NS_URI この定数は，名前空間 URI を表す文字列「http://www.w3.org/XML/1998/namespace」を保持します。この名前空間 URI は，W3C の勧告である Namespaces in XML で，名前空間接頭辞 xml に関連づけられています。
11	static java.lang.String	XMLNS_NS_URI この定数は，名前空間 URI を表す文字列「http://www.w3.org/2000/xmlns/」を保持します。この名前空間 URI は，W3C の勧告である DOM Level 2 で，名前空間接頭辞 xmlns に関連づけられています。

項番	説明	
12	static int	XBRL_2_0 この定数は，XBRL のバージョンが 2.0 であることを示すために使用されます。
13	static int	XBRL_2_1 この定数は，XBRL のバージョンが 2.1 であることを示すために使用されます。
14	static java.lang.String	XBRL21_ROLE_LINK この定数は，XBRL2.1 で規定されている拡張リンクのロールの標準値「http://www.xbrl.org/2003/role/link」を保持します。
15	static java.lang.String	XBRL21_ROLE_LABEL_LINKBASE_REF この定数は，XBRL2.1 で規定されている <link:linkbaseRef> 要素のロールの標準値「http://www.xbrl.org/2003/role/labelLinkbaseRef」を保持します。
16	static java.lang.String	XBRL21_ROLE_REFERENCE_LINKBASE_REF この定数は，XBRL2.1 で規定されている <link:linkbaseRef> 要素のロールの標準値「http://www.xbrl.org/2003/role/referenceLinkbaseRef」を保持します。
17	static java.lang.String	XBRL21_ROLE_CALCULATION_LINKBASE_REF この定数は，XBRL2.1 で規定されている <link:linkbaseRef> 要素のロールの標準値「http://www.xbrl.org/2003/role/calculationLinkbaseRef」を保持します。
18	static java.lang.String	XBRL21_ROLE_DEFINITION_LINKBASE_REF この定数は，XBRL2.1 で規定されている <link:linkbaseRef> 要素のロールの標準値「http://www.xbrl.org/2003/role/definitionLinkbaseRef」を保持します。
19	static java.lang.String	XBRL21_ROLE_PRESENTATION_LINKBASE_REF この定数は，XBRL2.1 で規定されている <link:linkbaseRef> 要素のロールの標準値「http://www.xbrl.org/2003/role/presentationLinkbaseRef」を保持します。
20	static java.lang.String	XBRL21_ROLE_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/label」を保持します。
21	static java.lang.String	XBRL21_ROLE_TERSE_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/terseLabel」を保持します。
22	static java.lang.String	XBRL21_ROLE_VERBOSE_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/verboseLabel」を保持します。
23	static java.lang.String	XBRL21_ROLE_POSITIVE_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/positiveLabel」を保持します。
24	static java.lang.String	XBRL21_ROLE_POSITIVE_TERSE_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/positiveTerseLabel」を保持します。

項番	説明	
25	static java.lang.String	XBRL21_ROLE_POSITIVE_VERBOSE_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/positiveVerboseLabel」を保持します。
26	static java.lang.String	XBRL21_ROLE_NEGATIVE_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/negativeLabel」を保持します。
27	static java.lang.String	XBRL21_ROLE_NEGATIVE_TERSE_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/negativeTerseLabel」を保持します。
28	static java.lang.String	XBRL21_ROLE_NEGATIVE_VERBOSE_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/negativeVerboseLabel」を保持します。
29	static java.lang.String	XBRL21_ROLE_ZERO_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/zeroLabel」を保持します。
30	static java.lang.String	XBRL21_ROLE_ZERO_TERSE_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/zeroTerseLabel」を保持します。
31	static java.lang.String	XBRL21_ROLE_ZERO_VERBOSE_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/zeroVerboseLabel」を保持します。
32	static java.lang.String	XBRL21_ROLE_TOTAL_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/totalLabel」を保持します。
33	static java.lang.String	XBRL21_ROLE_PERIOD_START_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/periodStartLabel」を保持します。
34	static java.lang.String	XBRL21_ROLE_PERIOD_END_LABEL この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/periodEndLabel」を保持します。
35	static java.lang.String	XBRL21_ROLE_DOCUMENTATION この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/documentation」を保持します。
36	static java.lang.String	XBRL21_ROLE_DEFINITION_GUIDANCE この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/definitionGuidance」を保持します。
37	static java.lang.String	XBRL21_ROLE_DISCLOSURE_GUIDANCE この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/disclosureGuidance」を保持します。

項番	説明	
38	static java.lang.String	XBRL21_ROLE_PRESENTATION_GUIDANCE この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/presentationGuidance」を保持します。
39	static java.lang.String	XBRL21_ROLE_MEASUREMENT_GUIDANCE この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/measurementGuidance」を保持します。
40	static java.lang.String	XBRL21_ROLE_COMMENTARY_GUIDANCE この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/commentaryGuidance」を保持します。
41	static java.lang.String	XBRL21_ROLE_EXAMPLE_GUIDANCE この定数は，XBRL2.1 で規定されている <link:label> 要素のロールの標準値「http://www.xbrl.org/2003/role/exampleGuidance」を保持します。
42	static java.lang.String	XBRL21_ROLE_REFERENCE この定数は，XBRL2.1 で規定されている <link:reference> 要素のロールの標準値「http://www.xbrl.org/2003/role/reference」を保持します。
43	static java.lang.String	XBRL21_ROLE_DEFINITION_REF この定数は，XBRL2.1 で規定されている <link:reference> 要素のロールの標準値「http://www.xbrl.org/2003/role/definitionRef」を保持します。
44	static java.lang.String	XBRL21_ROLE_DISCLOSURE_REF この定数は，XBRL2.1 で規定されている <link:reference> 要素のロールの標準値「http://www.xbrl.org/2003/role/disclosureRef」を保持します。
45	static java.lang.String	XBRL21_ROLE_MANDATORY_DISCLOSURE_REF この定数は，XBRL2.1 で規定されている <link:reference> 要素のロールの標準値「http://www.xbrl.org/2003/role/mandatoryDisclosureRef」を保持します。
46	static java.lang.String	XBRL21_ROLE_RECOMMENDED_DISCLOSURE_REF この定数は，XBRL2.1 で規定されている <link:reference> 要素のロールの標準値「http://www.xbrl.org/2003/role/recommendedDisclosureRef」を保持します。
47	static java.lang.String	XBRL21_ROLE_UNSPECIFIED_DISCLOSURE_REF この定数は，XBRL2.1 で規定されている <link:reference> 要素のロールの標準値「http://www.xbrl.org/2003/role/unspecifiedDisclosureRef」を保持します。
48	static java.lang.String	XBRL21_ROLE_PRESENTATION_REF この定数は，XBRL2.1 で規定されている <link:reference> 要素のロールの標準値「http://www.xbrl.org/2003/role/presentationRef」を保持します。
49	static java.lang.String	XBRL21_ROLE_MEASUREMENT_REF この定数は，XBRL2.1 で規定されている <link:reference> 要素のロールの標準値「http://www.xbrl.org/2003/role/measurementRef」を保持します。

項番	説明	
50	static java.lang.String	XBRL21_ROLE_COMMENTARY_REF この定数は，XBRL2.1 で規定されている <link:reference> 要素のロールの標準値「http://www.xbrl.org/2003/role/commentaryRef」を保持します。
51	static java.lang.String	XBRL21_ROLE_EXAMPLE_REF この定数は，XBRL2.1 で規定されている <link:reference> 要素のロールの標準値「http://www.xbrl.org/2003/role/exampleRef」を保持します。
52	static java.lang.String	XBRL21_ROLE_FOOTNOTE この定数は，XBRL2.1 で規定されている <link:footnote> 要素のロールの標準値「http://www.xbrl.org/2003/role/footnote」を保持します。
53	static java.lang.String	XBRL21_ARCROLE_CONCEPT_LABEL この定数は，XBRL2.1 で規定されている <link:labelArc> 要素のアークロールの標準値「http://www.xbrl.org/2003/arcrole/concept-label」を保持します。
54	static java.lang.String	XBRL21_ARCROLE_CONCEPT_REFERENCE この定数は，XBRL2.1 で規定されている <link:referenceArc> 要素のアークロールの標準値「http://www.xbrl.org/2003/arcrole/concept-reference」を保持します。
55	static java.lang.String	XBRL21_ARCROLE_SUMMATION_ITEM この定数は，XBRL2.1 で規定されている <link:calculationArc> 要素のアークロールの標準値「http://www.xbrl.org/2003/arcrole/summation-item」を保持します。
56	static java.lang.String	XBRL21_ARCROLE_GENERAL_SPECIAL この定数は，XBRL2.1 で規定されている <link:definitionArc> 要素のアークロールの標準値「http://www.xbrl.org/2003/arcrole/general-special」を保持します。
57	static java.lang.String	XBRL21_ARCROLE_ESSENCE_ALIAS この定数は，XBRL2.1 で規定されている <link:definitionArc> 要素のアークロールの標準値「http://www.xbrl.org/2003/arcrole/essence-alias」を保持します。
58	static java.lang.String	XBRL21_ARCROLE_SIMILAR_TUPLE この定数は，XBRL2.1 で規定されている <link:definitionArc> 要素のアークロールの標準値「http://www.xbrl.org/2003/arcrole/similar-tuples」を保持します。
59	static java.lang.String	XBRL21_ARCROLE_REQUIRES_ELEMENT この定数は，XBRL2.1 で規定されている <link:definitionArc> 要素のアークロールの標準値「http://www.xbrl.org/2003/arcrole/requires-element」を保持します。
60	static java.lang.String	XBRL21_ARCROLE_PARENT_CHILD この定数は，XBRL2.1 で規定されている <link:presentationArc> 要素のアークロールの標準値「http://www.xbrl.org/2003/arcrole/parent-child」を保持します。

項番	説明	
61	static java.lang.String	XBRL21_ARCROLE_FACT_FOOTNOTE この定数は，XBRL2.1 で規定されている <link:footnoteArc> 要素のアークロールの標準値「http://www.xbrl.org/2003/arcrole/fact-footnote」を保持します。
62	static java.lang.String	DIM10_ARCROLE_ALL この定数は，XBRL Dimensions1.0 で規定されている <link:definitionArc> 要素のアークロールの標準値「http://xbrl.org/int/dim/arcrole/all」を保持します。
63	static java.lang.String	DIM10_ARCROLE_NOTALL この定数は，XBRL Dimensions1.0 で規定されている <link:definitionArc> 要素のアークロールの標準値「http://xbrl.org/int/dim/arcrole/notAll」を保持します。
64	static java.lang.String	DIM10_ARCROLE_HYPERCUBE_DIMENSION この定数は，XBRL Dimensions1.0 で規定されている <link:definitionArc> 要素のアークロールの標準値「http://xbrl.org/int/dim/arcrole/hypercube-dimension」を保持します。
65	static java.lang.String	DIM10_ARCROLE_DIMENSION_DOMAIN この定数は，XBRL Dimensions1.0 で規定されている <link:definitionArc> 要素のアークロールの標準値「http://xbrl.org/int/dim/arcrole/dimension-domain」を保持します。
66	static java.lang.String	DIM10_ARCROLE_DOMAIN_MEMBER この定数は，XBRL Dimensions1.0 で規定されている <link:definitionArc> 要素のアークロールの標準値「http://xbrl.org/int/dim/arcrole/domain-member」を保持します。
67	static java.lang.String	DIM10_ARCROLE_DIMENSION_DEFAULT この定数は，XBRL Dimensions1.0 で規定されている <link:definitionArc> 要素のアークロールの標準値「http://xbrl.org/int/dim/arcrole/dimension-default」を保持します。
68	static java.lang.String	DIM10_TAXONOMY_NS_URI この定数は，名前空間 URI を表す文字列 "http://xbrl.org/2005/xbrldt" を保持します。
69	static java.lang.String	DIM10_INSTANCE_NS_URI この定数は，名前空間 URI を表す文字列「http://xbrl.org/2006/xbrldi」を保持します。
70	static java.lang.String	DIM10_TAXONOMY_ERROR_NS_URI この定数は，名前空間 URI を表す文字列「http://xbrl.org/2005/xbrldt/errors」を保持します。
71	static java.lang.String	DIM10_INSTANCE_ERROR_NS_URI この定数は，名前空間 URI を表す文字列「http://xbrl.org/2005/xbrldi/errors」を保持します。
72	static int	DIMENSIONS_NONE この定数は，XBRL Dimensions に対応しないことを示すために使用されます。

項番	説明	
73	static int	DIMENSIONS_1_0 この定数は，XBRL Dimensions のバージョンが 1.0 であることを示すために使用されます。

6.10.5 XBRLIdManager クラス

```
public class XBRLIdManager
extends java.lang.Object
```

XBRL のバージョン：

2.1

説明：

XBRL 文書に記述された ID を管理します。

コンストラクタの概要：

XBRLIdManager クラスのコンストラクタの概要を次の表に示します。

表 6-166 XBRLIdManager クラスのコンストラクタの概要

説明	XBRL のバージョン
XBRLIdManager() XBRL 文書に記述された ID を管理する XBRLIdManager オブジェクトを新規に構築します。	2.1

メソッドの概要：

XBRLIdManager クラスのメソッドの概要を次の表に示します。

表 6-167 XBRLIdManager クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLIdCreator	setIdCreator(XBRLIdCreator creator) XBRLIdCreator インターフェースを継承したオブジェクトを，XBRLIdManager オブジェクトに設定します。	2.1
2	XBRLIdCreator	getIdCreator() この XBRLIdManager オブジェクトに設定されている XBRLIdCreator インターフェースを継承したオブジェクトを取得します。	2.1
3	void	registerDocument(XBRLInstanceDocument doc) XBRLInstanceDocument オブジェクトを XBRLIdManager オブジェクトに登録します。	2.1
4	void	unregisterDocument(XBRLInstanceDocument doc) XBRLIdManager オブジェクトで管理する XBRLInstanceDocument オブジェクトの登録を解除します。	2.1

項番	説明		XBRL のバージョン
5	boolean	isId(java.lang.String id) この XBRLIdManager オブジェクトに登録されている XBRLInstanceDocument オブジェクトで，指定した ID がすでに使用されている ID かどうかを調べます。	2.1
6	java.lang.String[]	getIds() この XBRLIdManager オブジェクトに登録されている XBRLInstanceDocument オブジェクトで，すでに使用されている ID の配列を取得します。	2.1

(1) XBRLIdManager

```
public XBRLIdManager()
```

XBRL のバージョン：

2.1

説明：

XBRL 文書に記述された ID を管理する XBRLIdManager オブジェクトを新規に構築します。

(2) setIdCreator

```
public XBRLIdCreator setIdCreator(XBRLIdCreator creator)
```

XBRL のバージョン：

2.1

説明：

XBRLIdCreator インターフェースを継承したオブジェクトを，XBRLIdManager オブジェクトに設定します。

XBRLIdManager オブジェクトに設定した XBRLIdCreator オブジェクトは，ID が自動採番される前に createId() メソッドによって呼び出されます。XBRLIdCreator を実装して setIdCreator() メソッドを利用すれば，ユーザーアプリケーションで ID を生成できます。

setIdCreator() メソッドの引数 creator に null を指定すると，uCosminexus Business Reporting Processor による自動採番（未設定）に設定が戻ります。

引数：

creator - 自動採番で利用する XBRLIdCreator オブジェクトを指定します。

戻り値：

設定前の XBRLIdCreator オブジェクトを返します。未設定の場合には，デフォルト実装の XBRLIdCreator オブジェクトを返します。

(3) getIdCreator

```
public XBRLIdCreator getIdCreator()
```

XBRL のバージョン：

2.1

説明：

この XBRLIdManager オブジェクトに設定されている XBRLIdCreator インターフェースを継承したオブジェクトを取得します。

戻り値：

設定されている XBRLIdCreator オブジェクトを返します。設定されていない場合は，デフォルト実装の XBRLIdCreator オブジェクトを返します。

（4） registerDocument

```
public void registerDocument(XBRLInstanceDocument doc)
```

XBRL のバージョン：

2.1

説明：

XBRLInstanceDocument オブジェクトを XBRLIdManager オブジェクトに登録します。

登録時に引数 doc に指定した XBRLInstanceDocument オブジェクトは，既存の XBRLIdManager オブジェクトから自動的に登録を解除されます。一つの XBRLInstanceDocument オブジェクトを，複数の XBRLIdManager オブジェクトでは管理しません。

また，すでにこの XBRLIdManager オブジェクトで管理されている XBRLInstanceDocument オブジェクトを引数 doc に指定しても，例外は発生しません。

引数：

doc - このオブジェクトで管理する XBRLInstanceDocument オブジェクトを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 doc に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 doc に指定された XBRLInstanceDocument オブジェクトが準拠する XBRL のバージョンが 2.1 でない場合にスローされます。［エラーコード：20144］
- このオブジェクトで管理する ID と，引数 doc に指定された XBRLInstanceDocument オブジェクトで使用されている ID が重複した場合にスローされます。［エラーコード：20028］
- 引数 doc に指定された XBRLInstanceDocument オブジェクトの XBRLInstanceDocument.getW3CDOMNode() の結果と，すでに管理対象となっています。オブジェクトの結果が異なる場合にスローされます。［エラーコード：20143］

(5) unregisterDocument

```
public void unregisterDocument(XBRLInstanceDocument doc)
```

XBRL のバージョン：

2.1

説明：

この XBRLIdManager オブジェクトで管理する XBRLInstanceDocument オブジェクトの登録を解除します。
解除後，引数 doc の XBRLInstanceDocument オブジェクトは，デフォルト実装の新たな XBRLIdManager オブジェクトで管理されます。
ただし，この XBRLIdManager オブジェクトに設定されている XBRLIdCreator オブジェクトは，デフォルト実装の新たな XBRLIdManager オブジェクトの XBRLIdCreator オブジェクトとして設定されます。
よって，解除後に uCosminexus Business Reporting Processor による自動採番を利用する場合には，XBRLIdManager.setIdCreator() メソッドで未設定に変更する必要があります。

引数：

doc - このオブジェクトで管理している XBRLInstanceDocument オブジェクトを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 doc に null を指定した場合にスローされます。［エラーコード：20002］

(6) isId

```
public boolean isId(java.lang.String id)
```

XBRL のバージョン：

2.1

説明：

この XBRLIdManager オブジェクトに登録されているすべての XBRLInstanceDocument オブジェクトで，指定した ID がすでに使用されているかどうかを調べます。

引数：

id - この XBRLIdManager オブジェクトに登録されているすべての XBRLInstanceDocument オブジェクトで，すでに使用されているかどうかを調べる ID を指定します。

戻り値：

この XBRLIdManager オブジェクトに登録されているすべての XBRLInstanceDocument で，引数 id がすでに使用されている ID の場合は true,

使用されていない場合には false を返します。

（7） getIds

```
public java.lang.String[] getIds()
```

XBRL のバージョン：

2.1

説明：

この XBRLIdManager オブジェクトに登録されているすべての XBRLInstanceDocument オブジェクトで，すでに使用されている ID の配列を取得します。

戻り値：

この XBRLIdManager オブジェクトに登録されているすべての XBRLInstanceDocument オブジェクトで，すでに使用されている ID の配列を返します。

すでに使用されている ID が存在しない場合は null を返します。

6.10.6 XBRLNamespaceDeclaration クラス

```
public class XBRLNamespaceDeclaration
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

名前空間接頭辞と対応する名前空間 URI の組を管理します。

コンストラクタの概要：

XBRLNamespaceDeclaration クラスのコンストラクタの概要を次の表に示します。

表 6-168 XBRLNamespaceDeclaration クラスのコンストラクタの概要

項番	説明	XBRL のバージョン
1	XBRLNamespaceDeclaration(java.lang.String prefix, java.lang.String uri) 指定した名前空間接頭辞と名前空間 URI から XBRLNamespaceDeclaration オブジェクトを構築します。	2.0 および 2.1
2	XBRLNamespaceDeclaration(java.lang.String prefix, java.lang.String uri, boolean validate) 指定した名前空間接頭辞と名前空間 URI から XBRLNamespaceDeclaration オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLNamespaceDeclaration クラスのメソッドの概要を次の表に示します。

表 6-169 XBRLNamespaceDeclaration クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getPrefix() 名前空間接頭辞を取得します。	2.0 および 2.1
2	java.lang.String	getURI() 名前空間 URI を取得します。	2.0 および 2.1

(1) XBRLNamespaceDeclaration

```
public  XBRLNamespaceDeclaration(java.lang.String prefix,
                                 java.lang.String uri)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した名前空間接頭辞と名前空間 URI から XBRLNamespaceDeclaration オブジェクトを構築します。このコンストラクタでは，引数 prefix および uri に不正な文字が含まれるかどうかはチェックしません。

引数：

prefix - 名前空間接頭辞を指定します。名前空間接頭辞がない場合は null を指定します。

uri - 名前空間 URI を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 uri に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 prefix が null 以外で，かつ引数 uri に空の文字列を指定した場合にスローされます。[エラーコード：20062]

(2) XBRLNamespaceDeclaration

```
public  XBRLNamespaceDeclaration(java.lang.String prefix,
                                 java.lang.String uri,
                                 boolean validate)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した名前空間接頭辞と名前空間 URI から XBRLNamespaceDeclaration オブジェクトを構築します。

このコンストラクタでは，引数 validate に true を指定した場合だけ，名前空間接頭辞と名前空間 URI に指定された文字列に不正な文字が含まれるかどうかをチェック

します。その場合，引数 prefix および uri には，それぞれ XML Schema のデータ型 NCName および anyURI として妥当な文字列だけを指定できます。

引数：

prefix - 名前空間接頭辞を指定します。名前空間接頭辞がない場合は null を指定します。

uri - 名前空間 URI を指定します。

validate - 名前空間接頭辞または名前空間 URI に不正な文字が含まれるかチェックする場合は true を指定します。チェックしない場合は false を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 uri に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 prefix が null 以外で，かつ，引数 uri に空の文字列が指定された場合にスローされます。［エラーコード：20062］
- 引数 prefix に指定した名前空間接頭辞，または引数 uri に指定した名前空間 URI に不正な文字が含まれている場合にスローされます。［エラーコード：25008, 25013］

（3） getPrefix

```
public java.lang.String getPrefix()
```

XBRL のバージョン：

2.0 および 2.1

説明：

名前空間接頭辞を取得します。

戻り値：

名前空間接頭辞を返します。名前空間接頭辞がない場合は null を返します。

XBRLInstanceDocument インターフェースの getNamespaceDeclaration() ／ getNamespaceDeclarationByURI() ／ getNamespaceDeclarationByPrefix() の各メソッドは，常に正規化後の名前空間宣言を表す XBRLNamespaceDeclaration オブジェクトを返します。このため，取得した XBRLNamespaceDeclaration オブジェクトに対して getPrefix() メソッドを実行した場合，戻り値は常に正規化後の接頭辞になります。

（4） getURI

```
public java.lang.String getURI()
```

XBRL のバージョン：

2.0 および 2.1

説明：

名前空間 URI を取得します。

戻り値：

名前空間 URI を返します。

6.10.7 XBRLSchemaLocation クラス

```
public class XBRLSchemaLocation
extends java.lang.Object
```

XBRL のバージョン：

2.0 および 2.1

説明：

xsi:schemaLocation 属性に指定する，名前空間 URI とスキーマ文書の位置を示す URI の組を管理します。

コンストラクタの概要：

XBRLSchemaLocation クラスのコンストラクタの概要を次の表に示します。

表 6-170 XBRLSchemaLocation クラスのコンストラクタの概要

項番	説明	XBRL のバージョン
1	XBRLSchemaLocation(java.lang.String namespaceURI, java.lang.String schemaLocation) 指定した名前空間 URI とスキーマ文書の位置を示す URI から， XBRLSchemaLocation オブジェクトを構築します。	2.0 および 2.1
2	XBRLSchemaLocation(java.lang.String namespaceURI, java.lang.String schemaLocation, boolean validate) 指定した名前空間 URI とスキーマ文書の位置を示す URI から，XBRLSchemaLocation オブジェクトを構築します。	2.0 および 2.1

メソッドの概要：

XBRLSchemaLocation クラスのメソッドの概要を次の表に示します。

表 6-171 XBRLSchemaLocation クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getNamespaceURI() 名前空間 URI を取得します。	2.0 および 2.1
2	java.lang.String	getSchemaLocation() スキーマ文書の位置を示す URI を取得します。	2.0 および 2.1

(1) XBRLSchemaLocation

```
public  XBRLSchemaLocation(java.lang.String namespaceURI,
                           java.lang.String schemaLocation)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した名前空間 URI とスキーマ文書の位置を示す URI から，
XBRLSchemaLocation オブジェクトを構築します。
このコンストラクタでは，引数 namespaceURI および schemaLocation に不正な文字が含まれるかどうかはチェックしません。

引数：

namespaceURI - 名前空間 URI を指定します。
schemaLocation - スキーマ文書の位置を示す URI を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 namespaceURI または schemaLocation に null を指定した場合にスローされます。［エラーコード：20002］

（2）XBRLSchemaLocation

```
public  XBRLSchemaLocation(java.lang.String namespaceURI,
                           java.lang.String schemaLocation,
                           boolean validate)
```

XBRL のバージョン：

2.0 および 2.1

説明：

指定した名前空間 URI とスキーマ文書の位置を示す URI から，
XBRLSchemaLocation オブジェクトを構築します。
このコンストラクタでは引数 validate に true が指定された場合だけ，名前空間 URI とスキーマ文書の位置を示す URI に指定された文字列に不正な文字が含まれるかどうかをチェックします。その場合，引数 namespaceURI および schemaLocation には，共に XML Schema のデータ型 anyURI として妥当で，かつ空文字列（""）以外の文字列だけを指定できます。

引数：

namespaceURI - 名前空間 URI を指定します。
schemaLocation - スキーマ文書の位置を示す URI を指定します。
validate - 名前空間 URI またはスキーマ文書の位置を示す URI に不正な文字が含まれるかどうかをチェックする場合は true を指定します。チェックしない場合は false を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 namespaceURI または schemaLocation に null を指定した場合にスローされ

ます。［エラーコード：20002］

- 引数 namespaceURI に指定した名前空間 URI，または引数 schemaLocation に指定したスキーマ文書の位置を示す URI に不正な文字が含まれている場合にスローされます。［エラーコード：25008，25014］

（3） getNamespaceURI

```
public java.lang.String getNamespaceURI()
```

XBRL のバージョン：

2.0 および 2.1

説明：

名前空間 URI を取得します。

戻り値：

名前空間 URI を返します。

（4） getSchemaLocation

```
public java.lang.String getSchemaLocation()
```

XBRL のバージョン：

2.0 および 2.1

説明：

スキーマ文書の位置を示す URI を取得します。

戻り値：

スキーマ文書の位置を示す URI を返します。

6.10.8 PTVInstance クラス

```
public class PTVInstance
```

XBRL のバージョン：

2.1

説明：

PTVI の情報を保持します。

メソッドの概要：

PTVInstance クラスのメソッドの概要を次の表に示します。

表 6-172　PTVInstance クラスのメソッドの概要

項番	説明		XBRL のバージョン
1	org.w3c.dom.Document	getW3CDOMDocument() PTVI の DOM ツリーを取得します。	2.1
2	java.util.List<org.w3c.dom.Element>	getInferredItems() PTVI に含まれるアイテムのうち，完全な妥当性検証によって推測されたアイテムのリストを取得します。	2.1

（1） getW3CDOMDocument

```
public org.w3c.dom.Document getW3CDOMDocument()
```

XBRL のバージョン：

2.1

説明：

PTVI の DOM ツリーを取得します。

戻り値：

PTVI の DOM ツリーを表す org.w3c.dom.Document オブジェクトを返します。

（2） getInferredItems

```
public java.util.List<org.w3c.dom.Element> getInferredItems()
```

XBRL のバージョン：

2.1

説明：

PTVI に含まれるアイテムのうち，完全な妥当性検証によって推測されたアイテムのリストを取得します。完全な妥当性検証によるアイテムの推測については，「2.3.2 リンクの検証」を参照してください。

このメソッドが返すリストは読み取り専用のため，リストを変更しようとすると java.lang.UnsupportedOperationException がスローされます。

戻り値：

完全な妥当性検証によって推測されたアイテムを表す，org.w3c.dom.Element オブジェクトのリストを返します。その結果，アイテムが一つもなかった場合は，要素数 0 のリストを返します。

6.10.9　PTVLinkbase クラス

```
public class PTVLinkbase
```

XBRL のバージョン：

2.1

説明：

PTVLI の情報を保持します。

メソッドの概要：

PTVLinkbase クラスのメソッドの概要を次の表に示します。

表 6-173　PTVLinkbase クラスのメソッドの概要

説明		XBRL のバージョン
org.w3c.dom.Document	getW3CDOMDocument() PTVLI の DOM ツリーを取得します。	2.1

（1） getW3CDOMDocument

```
public org.w3c.dom.Document getW3CDOMDocument()
```

XBRL のバージョン：

2.1

説明：

PTVLI の DOM ツリーを取得します。

戻り値：

PTVLI の DOM ツリーを表す org.w3c.dom.Document オブジェクトを返します。

6.10.10　XBRLEqualUtil クラス

```
public final class XBRLEqualUtil
extends java.lang.Object
```

XBRL のバージョン：

2.1

説明：

項目とタプルの重複検出に関連する同等判定を可能にする機能を提供します。

コンストラクタの概要：

XBRLEqualUtil クラスのコンストラクタの概要を次の表に示します。

表 6-174　XBRLEqualUtil クラスのコンストラクタの概要

項番	説明	XBRL のバージョン
1	XBRLEqualUtil() XBRLEqualUtil オブジェクトを構築します。	2.1

項番	説明	XBRL のバージョン
2	XBRLEqualUtil(boolean ignoringZeroPrecision) XBRLEqualUtil オブジェクトを構築します。	2.1

メソッドの概要：

XBRLEqualUtil クラスのメソッドの概要を次の表に示します。

表 6-175　XBRLEqualUtil クラスのメソッドの概要

項番	説明		XBRL のバージョン	XBRL Dimensions のバージョン
1	boolean	s_equalContext(XBRLContext2 context1, XBRLContext2 context2) コンテキストが s-equal かどうかを判定します。	2.1	該当しない
2	boolean	s_equalItem(XBRLInstanceItem item1, XBRLInstanceItem item2) アイテムが s-equal かどうかを判定します。	2.1	該当しない
3	boolean	s_equalTuple(XBRLInstanceTuple tuple1, XBRLInstanceTuple tuple2) タプルが s-equal かどうかを判定します。	2.1	該当しない
4	boolean	s_equalUnit(XBRLUnit2 unit1, XBRLUnit2 unit2) 単位が s-equal かどうかを判定します。	2.1	該当しない
5	boolean	duplicateItem(XBRLInstanceItem item1, XBRLInstanceItem item2) アイテムが duplicate かどうかを判定します。	2.1	該当しない
6	boolean	duplicateTuple(XBRLInstanceTuple tuple1, XBRLInstanceTuple tuple2) タプルが duplicate かどうかを判定します。	2.1	該当しない
7	boolean	setIgnoringZeroPrecision(boolean ignoringZeroPrecision) XBRLEqualUtil オブジェクトに設定されている精度 0 のアイテムの扱いを設定します。	2.1	該当しない
8	boolean	isIgnoringZeroPrecision() XBRLEqualUtil オブジェクトに設定されている精度 0 のアイテムの扱いを取得します。	2.1	該当しない
9	boolean	d_equalItem(XBRLInstanceItem item1, XBRLInstanceItem item2, XBRLDimensionMember member) 引数に指定したアイテム同士が s-equal で，かつ，引数に指定したディメンジョンのメンバが両方のアイテムのコンテキストに存在するかどうかを判定します。	2.1	1.0

（1）XBRLEqualUtil

```
public XBRLEqualUtil()
```

XBRL のバージョン：

2.1

説明：

XBRLEqualUtil オブジェクトを構築します。このコンストラクタで生成した場合，精度 0 のアイテムの扱いは false となります。精度 0 のアイテムの扱いについては「6.10.10(2)　XBRLEqualUtil」または「6.10.10(9)　setIgnoringZeroPrecision」を参照してください。

(2) XBRLEqualUtil

```
public XBRLEqualUtil(boolean ignoringZeroPrecision)
```

XBRL のバージョン：

2.1

説明：

XBRLEqualUtil オブジェクトを構築します。

引数：

ignoringZeroPrecision - 精度 0 のアイテムの扱いを指定します。各アイテムの値の判定では，この設定値（デフォルトは false）が false の場合，（精度を考慮し）精度が "0" のアイテムの値は無効な値（NaN と同じ扱い）として，どちらかが精度 "0" の場合は常に不正とします。true の場合は，（精度を考慮せず）値の丸めは行わず記述されている数値の比較とします。

(3) s_equalContext

```
public boolean s_equalContext(XBRLContext2 context1,
                              XBRLContext2 context2)
```

XBRL のバージョン：

2.1

説明：

引数に指定したコンテキスト同士が s-equal かどうかを判定します。s-equal の場合，true を返します。s-equal でない場合は，false を返します。また，引数に指定したコンテキスト同士のインスタンス文書が異なる場合でも，s-equal の判定には影響しません。引数に指定したコンテキスト同士が同一のオブジェクトの場合は，必ず true を返します。

引数：

context1- コンテキストを指定します。

context2- コンテキストを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 context1 または context2 に null を指定した場合にスローされます。［エラー

コード：20002］

（4） s_equalItem

```
public boolean s_equalItem(XBRLInstanceItem item1,
                           XBRLInstanceItem item2)
```

XBRL のバージョン：

2.1

説明：

引数に指定したアイテム同士が s-equal かどうかを判定します。s-equal の場合，true を返します。s-equal でない場合は，false を返します。また，引数に指定したアイテム同士のインスタンス文書が異なる場合でも，s-equal の判定には影響しません。引数に指定したアイテム同士が同一のオブジェクトの場合は，必ず true を返します。

引数：

item1- アイテムを指定します。

item2- アイテムを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 item1 または item2 に null を指定した場合にスローされます。［エラーコード：20002］

（5） s_equalTuple

```
public boolean s_equalTuple(XBRLInstanceTuple tuple1,
                            XBRLInstanceTuple tuple2)
```

XBRL のバージョン：

2.1

説明：

引数に指定したタプル同士が s-equal かどうかを判定します。s-equal の場合，true を返します。s-equal でない場合は，false を返します。また，引数に指定したタプル同士のインスタンス文書が異なる場合でも，s-equal の判定には影響しません。引数に指定したタプル同士が同一のオブジェクトの場合は，必ず true を返します。

引数：

tuple1- タプルを指定します。

tuple2- タプルを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 tuple1 または tuple2 に null を指定した場合にスローされます。［エラーコード：20002］

(6) s_equalUnit

```
public boolean s_equalUnit(XBRLUnit2 unit1,
                           XBRLUnit2 unit2)
```

XBRL のバージョン：

2.1

説明：

引数に指定した単位同士が s-equal かどうかを判定します。s-equal の場合，true を返します。s-equal でない場合は，false を返します。また，引数に指定した単位同士のインスタンス文書が異なる場合でも，s-equal の判定には影響しません。引数に指定した単位同士が同一のオブジェクトの場合は，必ず true を返します。

引数：

unit1- 単位を指定します。

unit2- 単位を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 unit1 または unit2 に null を指定した場合にスローされます。［エラーコード：20002］

(7) duplicateItem

```
public boolean duplicateItem(XBRLInstanceItem item1,
                             XBRLInstanceItem item2)
```

XBRL のバージョン：

2.1

説明：

引数に指定したアイテム同士が duplicate かどうかを判定します。duplicate の場合，true を返します。duplicate でない場合は，false を返します。また，引数に指定したアイテム同士のインスタンス文書が異なる場合，false を返します。引数に指定したアイテム同士が同一のオブジェクトの場合，false を返します。

引数：

item1- アイテムを指定します。

item2- アイテムを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 item1 または item2 に null を指定した場合にスローされます。［エラーコード：20002］

(8) duplicateTuple

```
public boolean duplicateTuple(XBRLInstanceTuple tuple1,
                              XBRLInstanceTuple tuple2)
```

XBRL のバージョン：

2.1

説明：

引数に指定したタプル同士が duplicate かどうかを判定します。duplicate の場合，true を返します。duplicate でない場合は，false を返します。また，引数に指定したタプル同士のインスタンス文書が異なる場合，false を返します。引数に指定したタプル同士が同一のオブジェクトの場合，false を返します。タプルの子要素として，精度 0 のアイテムが記述されている場合，その値の判定はコンストラクタまたは setIgnoringZeroPrecision メソッドで設定した精度 0 のアイテムの扱いに従います。精度 0 のアイテムの扱いについては「6.10.10(2)　XBRLEqualUtil」または 6.10.10(9)　setIgnoringZeroPrecision」を参照してください。

引数：

tuple1- タプルを指定します。

tuple2- タプルを指定します。

例外：

XBRLDOMRuntimeException -

- 引数 tuple1 または tuple2 に null を指定した場合にスローされます。［エラーコード：20002］

(9) setIgnoringZeroPrecision

```
public void setIgnoringZeroPrecision(boolean ignoringZeroPrecision)
```

XBRL のバージョン：

2.1

説明：

XBRLEqualUtil オブジェクトに設定されている精度 0 のアイテムの扱いを設定します。各アイテムの値の判定では，この設定値（デフォルトは false）が false の場合，（精度を考慮し）精度が "0" のアイテムの値は無効な値（NaN と同じ扱い）として，どちらかが精度 "0" の場合は常に不正とします。true の場合は，（精度を考慮せず）値の丸めは行わず記述されている数値の比較とします。

引数：

ignoringZeroPrecision - オブジェクトに設定する精度 0 のアイテムの扱いを指定します。

(10) isIgnoringZeroPrecision

```
public boolean isIgnoringZeroPrecision()
```

XBRL のバージョン：

2.1

説明：

XBRLEqualUtil オブジェクトに設定されている精度 0 のアイテムの扱いを取得します。

(11) d_equalItem

```
public boolean d_equalItem ( XBRLInstanceItem item1 ,
   XBRLInstanceItem item2 ,
   XBRLDimensionMember member )
```

XBRL のバージョン：

2.1

XBRL Dimensions のバージョン：

1.0

説明：

引数に指定したアイテム同士が s-equal で，かつ，引数に指定したディメンジョンのメンバが両方のアイテムのコンテキストに存在するかどうかを判定します。

引数：

item1 - アイテムを指定します。

item 2 - アイテムを指定します。

member - ディメンジョンのメンバを指定します。

戻り値：

引数に指定したアイテム同士が s-equal で，かつ，引数に指定したディメンジョンのメンバが両方のアイテムのコンテキストに存在する場合，true を返します。そうでない場合，false を返します。

引数に指定したアイテム同士のインスタンス文書が異なる場合であっても，このメソッドの判定には影響しません。

例外：

XBRLDOMRuntimeException -

- 引数 item1 または item2 に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 member に null を指定した場合にスローされます。［エラーコード：20002］

6.11 com.hitachi.xbrl.simple.taxonomy パッケージ

com.hitachi.xbrl.simple.taxonomy パッケージは，単純なタクソノミー文書を新規作成，または拡張した新たなタクソノミー文書を作成するためのインターフェースを提供します。

インターフェースの概要：

com.hitachi.xbrl.simple.taxonomy パッケージのインターフェースを次の表に示します。

表 6-176 com.hitachi.xbrl.simple.taxonomy パッケージのインターフェースの概要

項番	インターフェース	説明	XBRL のバージョン
1	SimpleConceptDefinition	単純なタクソノミー文書を構成する要素定義のうち，アイテムを表します。	2.1
2	SimpleEdge	単純なタクソノミー文書を構成する関係ネットワークの一辺（一つの関係）を表します。	2.1
3	SimpleElementDefinition	単純なタクソノミー文書を構成する要素定義を表すインターフェースです。	2.1
4	SimpleItemDefinition	単純なタクソノミー文書を構成するアイテムの要素宣言を表すインターフェースです。	2.1
5	SimpleLabel	単純なタクソノミー文書を構成するラベル要素を表します。	2.1
6	SimpleLabelEdge	単純なタクソノミー文書を構成するラベルリンクの一辺（一つの関係）を表します。	2.1
7	SimpleNode	単純なタクソノミー文書を構成するノードを表します。	2.1
8	SimpleTaxonomy	単純なタクソノミー文書全体を表します。	2.1
9	SimpleTaxonomyBuilder	単純なタクソノミー文書を表す SimpleTaxonomy オブジェクトを生成します。	2.1
10	SimpleTaxonomyBuilderFactory	単純なタクソノミー文書を作成するための SimpleTaxonomyBuilder オブジェクトを生成します。	2.1

6.11.1 SimpleConceptDefinition インターフェース

```
public interface SimpleConceptDefinition
extends SimpleElementDefinition
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成する要素定義のうち，アイテムを表します。

6.11.2 SimpleEdge インターフェース

```
public interface SimpleEdge
extends SimpleNode
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成する関係ネットワークの一辺（一つの関係）を表します。

フィールドの概要：

SimpleEdge インターフェースのフィールドの概要を次の表に示します。

表 6-177 SimpleEdge インターフェースのフィールドの概要

項番	説明	
1	static int	LABEL_EDGE 単純なタクソノミー文書のラベルリンクの一辺を示します。
2	static java.lang.String	USE_OPTIONAL アーク型要素の use 属性の候補値である文字列 optional を示します。
3	static java.lang.String	USE_PROHIBITED アーク型要素の use 属性の候補値である文字列 prohibited を示します。

メソッドの概要：

SimpleEdge インターフェースのメソッドの概要を次の表に示します。

表 6-178 SimpleEdge インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getRole() 一つの関係が属する拡張リンク型要素の xlink:role 属性値を取得します。	2.1
2	java.lang.String	getArcrole() 一つの関係を構成するアーク型要素の xlink:arcrole 属性値を取得します。	2.1
3	java.lang.String	getUse() 一つの関係を構成するアーク型要素の use 属性値を取得します。	2.1

項番	説明		XBRL のバージョン
4	java.lang.Object	getFrom() 一つの関係を構成するアーク型要素の xlink:from 属性値で参照されるオブジェクトを取得します。	2.1
5	java.lang.Object	getTo() 一つの関係を構成するアーク型要素の xlink:to 属性値で参照されるオブジェクトを取得します。	2.1
6	java.math.BigDecimal	getOrder() 一つの関係を構成するアーク型要素の order 属性値を取得します。	2.1
7	void	setOrder(java.math.BigDecimal order) 一つの関係を構成するアーク型要素の order 属性値を設定します。	2.1

（1） getRole

```
public java.lang.String getRole()
```

XBRL のバージョン：

2.1

説明：

一つの関係が属する拡張リンク型要素の xlink:role 属性値を取得します。

戻り値：

一つの関係が属する拡張リンク型要素の xlink:role 属性値を返します。

（2） getArcrole

```
public java.lang.String getArcrole()
```

XBRL のバージョン：

2.1

説明：

一つの関係を構成するアーク型要素の xlink:arcrole 属性値を取得します。

戻り値：

一つの関係を構成するアーク型要素の xlink:arcrole 属性値を返します。

（3） getUse

```
public java.lang.String getUse()
```

XBRL のバージョン：

2.1

説明：

一つの関係を構成するアーク型要素の use 属性値を取得します。

戻り値：
一つの関係を構成するアーク型要素の use 属性値を返します。

(4) getFrom

```
public java.lang.Object getFrom()
```

XBRL のバージョン：
2.1

説明：
一つの関係を構成するアーク型要素の xlink:from 属性値で参照されるオブジェクトを取得します。

戻り値：
一つの関係を構成するアーク型要素の xlink:from 属性値で参照されるオブジェクトを返します。

(5) getTo

```
public java.lang.Object getTo()
```

XBRL のバージョン：
2.1

説明：
一つの関係を構成するアーク型要素の xlink:to 属性値で参照されるオブジェクトを取得します。

戻り値：
一つの関係を構成するアーク型要素の xlink:to 属性値で参照されるオブジェクトを返します。

(6) getOrder

```
public java.math.BigDecimal getOrder()
```

XBRL のバージョン：
2.1

説明：
一つの関係を構成するアーク型要素の order 属性値を取得します。
注　この SimpleEdge オブジェクトが既存の関係を否定するオブジェクトでない場合，order 属性値はオブジェクト生成時に「1」が設定されます。
また，この SimpleEdge オブジェクトが既存の関係を否定するオブジェクトの場合，既存の関係と同じ値が設定されます。

戻り値：
一つの関係を構成するアーク型要素の order 属性値を返します。

(7) setOrder

```
public void setOrder(java.math.BigDecimal order)
```

XBRL のバージョン：

2.1

説明：

一つの関係を構成するアーク型要素の order 属性値を設定します。この SimpleEdge オブジェクトが既存の関係を否定するオブジェクトの場合は，このメソッドを利用できません。

引数：

order - 一つの関係を構成するアーク型要素の order 属性値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 order に null を指定した場合にスローされます。[エラーコード：20002]
- この SimpleEdge オブジェクトが既存の関係を否定するオブジェクトであった場合にスローされます。[エラーコード：22004]

6.11.3 SimpleElementDefinition インターフェース

```
public interface SimpleElementDefinition
extends SimpleNode
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成する要素定義を表すインターフェースです。

メソッドの概要：

SimpleElementDefinition インターフェースのメソッドの概要を次の表に示します。

表 6-179 SimpleElementDefinition インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getName() 要素定義の name 属性値を取得します。	2.1
2	void	setName(java.lang.String name) 要素定義の name 属性値を設定します。	2.1
3	java.lang.String	getType() 要素定義の type 属性値を取得します。	2.1
4	java.lang.String	getId() 要素定義の id 属性値を取得します。	2.1

項番	説明		XBRL のバージョン
5	void	setId(java.lang.String id) 要素定義の id 属性値を設定します。	2.1
6	boolean	getAbstract() 要素定義の abstract 属性値を取得します。	2.1
7	void	setAbstract(boolean abstrakt) 要素定義の abstract 属性値を設定します。	2.1
8	boolean	getNillable() 要素定義の nillable 属性値を取得します。	2.1
9	void	setNillable(boolean nillable) 要素定義の nillable 属性値を設定します。	2.1
10	java.lang.String	getSubstitutionGroup() 要素定義の substitutionGroup 属性値を取得します。	2.1

(1) getName

```
public java.lang.String getName()
```

XBRL のバージョン：

2.1

説明：

要素定義の name 属性値を取得します。

戻り値：

要素定義の name 属性値を返します。

(2) setName

```
public void setName(java.lang.String name)
```

XBRL のバージョン：

2.1

説明：

要素定義の name 属性値を設定します。

引数：

name - 要素定義の name 属性値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 name に null を指定した場合にスローされます。[エラーコード：20002]
- 引数 name に空文字列（""）または xsd:NCName として不正な文字列が指定された場合にスローされます。[エラーコード：20108]
- 次のどちらかの条件に該当する場合にスローされます。[エラーコード：22002]

・この SimpleElementDefinition オブジェクトがアクティブであり，かつこの SimpleElementDefinition オブジェクトを所有する SimpleTaxonomy オブジェクトの中に，引数 name で指定された name 属性値を持つ別のアクティブな SimpleElementDefinition オブジェクトが存在する場合
・同じ対象名前空間で同じ name 属性値を持つ XBRLTaxonomyElement オブジェクトが，拡張元である XBRLTaxonomyDocument オブジェクトに存在する場合

(3) getType

```
public java.lang.String getType()
```

XBRL のバージョン：
2.1

説明：
要素定義の type 属性値を取得します。

戻り値：
要素定義の type 属性値を返します。

(4) getId

```
public java.lang.String getId()
```

XBRL のバージョン：
2.1

説明：
要素定義の id 属性値を取得します。
注　id 属性値には，このオブジェクトを所有する SimpleTaxonomy オブジェクトの getPrefix() で取得される接頭辞と，オブジェクト生成時に指定した name 属性値とを "_" で連結した文字列が設定されています。例えば，接頭辞が hi，name 属性値が balanceSheet の場合，id 属性値は hi_balanceSheet と設定されています。

戻り値：
要素定義の id 属性値を返します。

(5) setId

```
public void setId(java.lang.String id)
```

XBRL のバージョン：
2.1

説明：
要素定義の id 属性値を設定します。

引数：
id - 要素定義の id 属性値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 id に空文字列（""）または xsd:NCName として不正な文字列が指定された場合にスローされます。［エラーコード：20108］
- 次の二つの条件を同時に満たす場合にスローされます。［エラーコード：22003］
  ・この SimpleElementDefinition オブジェクトがアクティブである場合
  ・この SimpleElementDefinition オブジェクトを所有する SimpleTaxonomy オブジェクトの中に，引数 id で指定された id 属性値を持つ別のアクティブな SimpleElementDefinition オブジェクトが存在する場合
- 引数 id に null を指定した場合にスローされます。［エラーコード：20002］

（6） getAbstract

```
public boolean getAbstract()
```

XBRL のバージョン：

2.1

説明：

要素定義の abstract 属性値を取得します。

注　初期状態では false が設定されています。

戻り値：

要素定義の abstract 属性値を返します。

（7） setAbstract

```
public void setAbstract(boolean abstrakt)
```

XBRL のバージョン：

2.1

説明：

要素定義の abstract 属性値を設定します。

abstract 属性値が false の場合，実際のタクソノミー文書ではこの属性は設定されないで，省略されます。

引数：

abstrakt - 要素定義の abstract 属性値を指定します。

（8） getNillable

```
public boolean getNillable()
```

XBRL のバージョン：

2.1

説明：

要素定義の nillable 属性値を取得します。

注　初期状態では false が設定されています。

戻り値：

要素定義の nillable 属性値を返します。

（9） setNillable

```
public void setNillable(boolean nillable)
```

XBRL のバージョン：

2.1

説明：

要素定義の nillable 属性値を設定します。

nillable 属性値が false の場合，実際のタクソノミー文書ではこの属性は設定されないで省略されます。

引数：

nillable - 要素定義の nillable 属性値を指定します。

（10） getSubstitutionGroup

```
public java.lang.String getSubstitutionGroup()
```

XBRL のバージョン：

2.1

説明：

要素定義の substitutionGroup 属性値を取得します。要素定義とその戻り値を次の表に示します。

表 6-180　getSubstitutionGroup の要素定義と戻り値

要素定義	戻り値
アイテム	"xbrli:item"

戻り値：

要素定義の substitutionGroup 属性値を返します。

6.11.4　SimpleItemDefinition インターフェース

```
public interface SimpleItemDefinition
extends SimpleConceptDefinition
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成するアイテムの要素宣言を表すインターフェースで

す。

フィールドの概要：

SimpleElementDefinition インターフェースのフィールドの概要を次の表に示します。

表 6-181　SimpleItemDefinition インターフェースのフィールドの概要

項番	説明	
1	static int	DECIMAL_ITEM アイテム要素定義の type 属性値が xbrli:decimalItemType であることを示します。
2	static int	FLOAT_ITEM アイテム要素定義の type 属性値が xbrli:floatItemType であることを示します。
3	static int	DOUBLE_ITEM アイテム要素定義の type 属性値が xbrli:doubleItemType であることを示します。
4	static int	MONETARY_ITEM アイテム要素定義の type 属性値が xbrli:monetaryItemType であることを示します。
5	static int	SHARES_ITEM アイテム要素定義の type 属性値が xbrli:sharesItemType であることを示します。
6	static int	PURE_ITEM アイテム要素定義の type 属性値が xbrli:pureItemType であることを示します。
7	static int	FRACTION_ITEM アイテム要素定義の type 属性値が xbrli:fractionItemType であることを示します。
8	static int	INTEGER_ITEM アイテム要素定義の type 属性値が xbrli:integerItemType であることを示します。
9	static int	NONPOSITIVEINTEGER_ITEM アイテム要素定義の type 属性値が xbrli:nonPositiveIntegerItemType であることを示します。
10	static int	NEGATIVEINTEGER_ITEM アイテム要素定義の type 属性値が xbrli:negativeIntegerItemType であることを示します。
11	static int	LONG_ITEM アイテム要素定義の type 属性値が xbrli:longItemType であることを示します。
12	static int	INT_ITEM アイテム要素定義の type 属性値が xbrli:intItemType であることを示します。

項番	説明	
13	static int	SHORT_ITEM アイテム要素定義の type 属性値が xbrli:shortItemType であることを示します。
14	static int	BYTE_ITEM アイテム要素定義の type 属性値が xbrli:byteItemType であることを示します。
15	static int	NONNEGATIVEINTEGER_ITEM アイテム要素定義の type 属性値が xbrli:nonNegativeIntegerItemType であることを示します。
16	static int	UNSIGNEDLONG_ITEM アイテム要素定義の type 属性値が xbrli:unsignedLongItemType であることを示します。
17	static int	UNSIGNEDINT_ITEM アイテム要素定義の type 属性値が xbrli:unsignedIntItemType であることを示します。
18	static int	UNSIGNEDSHORT_ITEM アイテム要素定義の type 属性値が xbrli:unsignedShortItemType であることを示します。
19	static int	UNSIGNEDBYTE_ITEM アイテム要素定義の type 属性値が xbrli:unsignedByteItemType であることを示します。
20	static int	POSITIVEINTEGER_ITEM アイテム要素定義の type 属性値が xbrli:positiveIntegerItemType であることを示します。
21	static int	STRING_ITEM アイテム要素定義の type 属性値が xbrli:stringItemType であることを示します。
22	static int	BOOLEAN_ITEM アイテム要素定義の type 属性値が xbrli:booleanItemType であることを示します。
23	static int	HEXBINARY_ITEM アイテム要素定義の type 属性値が xbrli:hexBinaryItemType であることを示します。
24	static int	BASE64BINARY_ITEM アイテム要素定義の type 属性値が xbrli:base64BinaryItemType であることを示します。
25	static int	ANYURI_ITEM アイテム要素定義の type 属性値が xbrli:anyURIItemType であることを示します。
26	static int	QNAME_ITEM アイテム要素定義の type 属性値が xbrli:QNameItemType であることを示します。

項番	説明	
27	static int	DURATION_ITEM アイテム要素定義の type 属性値が xbrli:durationItemType であることを示します。
28	static int	DATETIME_ITEM アイテム要素定義の type 属性値が xbrli:dateTimeItemType であることを示します。
29	static int	TIME_ITEM アイテム要素定義の type 属性値が xbrli:timeItemType であることを示します。
30	static int	DATE_ITEM アイテム要素定義の type 属性値が xbrli:dateItemType であることを示します。
31	static int	GYEARMONTH_ITEM アイテム要素定義の type 属性値が xbrli:gYearMonthItemType であることを示します。
32	static int	GYEAR_ITEM アイテム要素定義の type 属性値が xbrli:gYearItemType であることを示します。
33	static int	GMONTHDAY_ITEM アイテム要素定義の type 属性値が xbrli:gMonthDayItemType であることを示します。
34	static int	GDAY_ITEM アイテム要素定義の type 属性値が xbrli:gDayItemType であることを示します。
35	static int	GMONTH_ITEM アイテム要素定義の type 属性値が xbrli:gMonthItemType であることを示します。
36	static int	NORMALIZEDSTRING_ITEM アイテム要素定義の type 属性値が xbrli:normalizedStringItemType であることを示します。
37	static int	TOKEN_ITEM アイテム要素定義の type 属性値が xbrli:tokenItemType であることを示します。
38	static int	LANGUAGE_ITEM アイテム要素定義の type 属性値が xbrli:languageItemType であることを示します。
39	static int	NAME_ITEM アイテム要素定義の type 属性値が xbrli:NameItemType であることを示します。
40	static int	NCNAME_ITEM アイテム要素定義の type 属性値が xbrli:NCNameItemType であることを示します。

項番	説明	
41	static java.lang.String	INSTANT xbrli:periodType の候補値である文字列 instant であることを示します。
42	static java.lang.String	DURATION xbrli:periodType の候補値である文字列 duration であることを示します。
43	static java.lang.String	DEBIT xbrli:balance の候補値である文字列 debit であることを示します。
44	static java.lang.String	CREDIT xbrli:balance の候補値である文字列 credit であることを示します。

メソッドの概要：

SimpleItemDefinition インターフェースのメソッドの概要を次の表に示します。

表 6-182　SimpleItemDefinition インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	int	getItemType() アイテム要素定義の type 属性値に対応するフィールド値を取得します。	2.1
2	void	setItemType(int type) アイテム要素定義の type 属性値を設定します。	2.1
3	java.lang.String	getPeriodType() アイテム要素定義の xbrli:periodType 属性値を取得します。	2.1
4	void	setPeriodType(java.lang.String periodType) アイテム要素定義の xbrli:periodType 属性値を設定します。	2.1
5	java.lang.String	getBalance() アイテム要素定義の xbrli:balance 属性値を取得します。	2.1
6	void	setBalance(java.lang.String balance) アイテム要素定義の xbrli:balance 属性値を設定します。	2.1

(1) getItemType

```
public int getItemType()
```

XBRL のバージョン：

2.1

説明：

アイテム要素定義の type 属性値に対応するフィールド値を取得します。

戻り値：

アイテム要素定義の type 属性値に対応するフィールド値を返します。

(2) setItemType

```
public void setItemType(int type)
```

XBRL のバージョン：

2.1

説明：

アイテム要素定義の type 属性値を設定します。

アイテム要素定義の type 属性値が xbrli:monetaryItemType の場合，このメソッドで type 属性値を変更すると，設定されていた xbrli:balance 属性値が削除されます。その後このオブジェクトの getBalance() メソッドを実行すると，null が返ります。

引数：

type - アイテム要素定義の type 属性値に対応するフィールド値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 type に指定できない値が指定された場合にスローされます。[エラーコード：20156]

(3) getPeriodType

```
public java.lang.String getPeriodType()
```

XBRL のバージョン：

2.1

説明：

アイテム要素定義の xbrli:periodType 属性値を取得します。

戻り値：

アイテム要素定義の xbrli:periodType 属性値を返します。

(4) setPeriodType

```
public void setPeriodType(java.lang.String periodType)
```

XBRL のバージョン：

2.1

説明：

アイテム要素定義の xbrli:periodType 属性値を設定します。

引数：

periodType - アイテム要素定義の xbrli:periodType 属性値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 periodType に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 periodType に instant または duration 以外の文字列が指定された場合にスローされます。［エラーコード：20014］

（5） getBalance

```
public java.lang.String getBalance()
```

XBRL のバージョン：

2.1

説明：

アイテム要素定義の xbrli:balance 属性値を取得します。xbrli:balance 属性値はオブジェクト生成時には設定されないため，生成直後にこのメソッドを実行した場合には null を返します。

戻り値：

アイテム要素定義の xbrli:balance 属性値を返します。xbrli:balance 属性値が設定されていない場合は，null を返します。

（6） setBalance

```
public void setBalance(java.lang.String balance)
```

XBRL のバージョン：

2.1

説明：

アイテム要素定義の xbrli:balance 属性値を設定します。アイテム要素定義から xlink:balance 属性を削除する場合には，null を指定します。

引数：

balance - このアイテム要素定義の xbrli:balance 属性値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 balance に debit または credit 以外の文字列が指定された場合にスローされます。［エラーコード：20014］
- アイテム要素定義の type 属性値が「xbrli:monetaryItemType」以外の場合にスローされます。［エラーコード：22009］

6.11.5 SimpleLabel インターフェース

```
public interface SimpleLabel
extends SimpleNode
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成するラベル要素を表します。

メソッドの概要：

SimpleLabel インターフェースのメソッドの概要を次の表に示します。

表 6-183　SimpleLabel インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	java.lang.String	getRole() ラベル要素（link:label）の xlink:role 属性値を取得します。	2.1
2	void	setRole(java.lang.String role) ラベル要素（link:label）の xlink:role 属性値を設定します。	2.1
3	java.lang.String	getLang() ラベル要素（link:label）の xml:lang 属性値を取得します。	2.1
4	void	setLang(java.lang.String lang) ラベル要素（link:label）の xml:lang 属性値を設定します。	2.1
5	java.lang.String	getTitle() ラベル要素（link:label）の xlink:title 属性値を取得します。	2.1
6	void	setTitle(java.lang.String title) ラベル要素（link:label）の xlink:title 属性値を設定します。	2.1
7	java.lang.String	getId() ラベル要素（link:label）の id 属性値を取得します。	2.1
8	void	setId(java.lang.String id) ラベル要素（link:label）の id 属性値を設定します。	2.1
9	java.lang.String	getValue() ラベル要素（link:label）の内容を取得します。	2.1
10	void	setValue(java.lang.String value) ラベル要素（link:label）の内容を設定します。	2.1

(1) getRole

```
public java.lang.String getRole()
```

XBRL のバージョン：

2.1

説明：

ラベル要素（link:label）の xlink:role 属性値を取得します。xlink:role 属性値は生成時には設定されないため，生成直後にこのメソッドを実行した場合には null を返します。

戻り値：

ラベル要素（link:label）の xlink:role 属性値を返します。
xlink:role 属性値が設定されていない場合には，null を返します。

（2） setRole

```
public void setRole(java.lang.String role)
```

XBRL のバージョン：
2.1

説明：
ラベル要素（link:label）の xlink:role 属性値を設定します。次に示す標準値とカスタムロール値が設定できます。ラベル要素（link:label）から xlink:role 属性を削除する場合には，null を指定してください。

設定できるカスタムロール値

設定できるカスタムロール値は，拡張元である XBRLTaxonomyDocument オブジェクトの getRoleTypes() メソッドで取得される XBRLRoleType オブジェクトのうち，getUsedOn() メソッドで取得されるリストに次の値を含む，XBRLRoleType オブジェクトの getRoleURI() メソッドで取得される値です。

- 名前空間：http://www.xbrl.org/2003/linkbase
- ローカル名：label

設定できる標準値

設定できる標準値は次のとおりです。

- http://www.xbrl.org/2003/role/label
- http://www.xbrl.org/2003/role/terseLabel
- http://www.xbrl.org/2003/role/verboseLabel
- http://www.xbrl.org/2003/role/positiveLabel
- http://www.xbrl.org/2003/role/positiveTerseLabel
- http://www.xbrl.org/2003/role/positiveVerboseLabel
- http://www.xbrl.org/2003/role/negativeLabel
- http://www.xbrl.org/2003/role/negativeTerseLabel
- http://www.xbrl.org/2003/role/negativeVerboseLabel
- http://www.xbrl.org/2003/role/zeroLabel
- http://www.xbrl.org/2003/role/zeroTerseLabel
- http://www.xbrl.org/2003/role/zeroVerboseLabel
- http://www.xbrl.org/2003/role/totalLabel
- http://www.xbrl.org/2003/role/periodStartLabel
- http://www.xbrl.org/2003/role/periodEndLabel
- http://www.xbrl.org/2003/role/documentation
- http://www.xbrl.org/2003/role/definitionGuidance
- http://www.xbrl.org/2003/role/disclosureGuidance
- http://www.xbrl.org/2003/role/presentationGuidance

- http://www.xbrl.org/2003/role/measurementGuidance
- http://www.xbrl.org/2003/role/commentaryGuidance
- http://www.xbrl.org/2003/role/exampleGuidance

引数：

role - ラベル要素（link:label）の xlink:role 属性値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 role に指定できない値が指定されている場合にスローされます。［エラーコード：20156］

(3) getLang

```
public java.lang.String getLang()
```

XBRL のバージョン：

2.1

説明：

ラベル要素（link:label）の xml:lang 属性値を取得します。

戻り値：

ラベル要素（link:label）の xml:lang 属性値を返します。

(4) setLang

```
public void setLang(java.lang.String lang)
```

XBRL のバージョン：

2.1

説明：

ラベル要素（link:label）の xml:lang 属性値を設定します。

引数：

lang - ラベル要素（link:label）の xml:lang 属性値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 lang に null を指定した場合にスローされます。［エラーコード：20002］

(5) getTitle

```
public java.lang.String getTitle()
```

XBRL のバージョン：

2.1

説明：

ラベル要素（link:label）の xlink:title 属性値を取得します。xlink:title 属性値はオ

ブジェクト生成時には設定されないため，生成直後にこのメソッドを実行した場合には，null を返します。

戻り値：

ラベル要素（link:label）の xml:title 属性値を返します。xlink:title 属性値が設定されていない場合には，null を返します。

（6） setTitle

```
public void setTitle(java.lang.String title)
```

XBRL のバージョン：

2.1

説明：

ラベル要素（link:label）の xlink:title 属性値を設定します。ラベル要素から xlink:title 属性を削除する場合には，null を指定します。

引数：

title - ラベル要素（link:label）の xlink:title 属性値を指定します。

（7） getId

```
public java.lang.String getId()
```

XBRL のバージョン：

2.1

説明：

ラベル要素（link:label）の id 属性値を取得します。id 属性値はオブジェクト生成時には設定されないため，生成直後にこのメソッドを実行した場合には，null を返します。

戻り値：

ラベル要素（link:label）の id 属性値を返します。id 属性値が設定されていない場合には，null を返します。

（8） setId

```
public void setId(java.lang.String id)
```

XBRL のバージョン：

2.1

説明：

ラベル要素（link:label）の id 属性値を設定します。ラベル要素から id 属性を削除する場合には，null を指定します。

引数：

id - このラベル要素（link:label）の id 属性値を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 id に空文字列（""）または xsd:NCName として不正な文字列が指定された場合にスローされます。［エラーコード：20108］
- 次の二つの条件を同時に満たす場合にスローされます。［エラーコード：22003］
  ・この SimpleLabel オブジェクトがアクティブである場合
  ・この SimpleLabel オブジェクトを所有する SimpleTaxonomy オブジェクトの中に，引数 id で指定された id 属性値を持つ別のアクティブな SimpleLabel オブジェクトが存在する場合

(9) getValue

```
public String getValue()
```

XBRL のバージョン：

2.1

説明：

ラベル要素（link:label）の内容を取得します。

戻り値：

ラベル要素（link:label）の内容を返します。

(10) setValue

```
public void setValue(java.lang.String value)
```

XBRL のバージョン：

2.1

説明：

ラベル要素（link:label）の内容を設定します。

引数：

value - ラベル要素（link:label）の内容を指定します。

例外：

XBRLDOMRuntimeException -

- 引数 value に null を指定した場合にスローされます。［エラーコード：20002］

6.11.6 SimpleLabelEdge インターフェース

```
public interface SimpleLabelEdge
extends SimpleEdge
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成するラベルリンクの一辺（一つの関係）を表します。

6.11.7 SimpleNode インターフェース

```
public interface SimpleNode
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成するノードを表します。
com.hitachi.xbrl.simple.taxonomy パッケージのすべてのインターフェースの基底となるインターフェースです。

フィールドの概要：

SimpleNode インターフェースのフィールドの概要を次の表に示します。

表 6-184 SimpleNode インターフェースのフィールドの概要

項番	説明	
1	static int	SIMPLE_TAXONOMY_NODE ノードが SimpleTaxonomy であることを示します。
2	static int	SIMPLE_ITEM_DEFINITION_NODE ノードが SimpleItemDefinition であることを示します。
3	static int	SIMPLE_LABEL_NODE ノードが SimpleLabel であることを示します。
4	static int	SIMPLE_LABEL_EDGE_NODE ノードが SimpleLabelEdge であることを示します。

メソッドの概要：

SimpleNode インターフェースのメソッドの概要を次の表に示します。

表 6-185 SimpleNode インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	boolean	isActive() ノードがアクティブであるかどうかを調べます。	2.1
2	int	getNodeType() ノードの型を表すフィールド値を返します。	2.1
3	SimpleTaxonomy	getOwnerTaxonomy() ノードを所有する SimpleTaxonomy オブジェクトを取得します。ノードが SimpleTaxonomy オブジェクトの場合には，常に null を返します。	2.1
4	SimpleNode	getParentNode() ノードが関連づけられている親ノードを取得します。	2.1

(1) isActive

```
public boolean isActive()
```

XBRL のバージョン：

2.1

説明：

ノードがアクティブであるかどうかを調べます。ノードが SimpleTaxonomy オブジェクトの子孫ノードの場合には true を返し，子孫ノードでない場合には false を返します。

戻り値：

ノードがアクティブな場合は true，そうでない場合は false を返します。このノードが SimpleTaxonomy オブジェクトの場合は，常に true を返します。

(2) getNodeType

```
public int getNodeType()
```

XBRL のバージョン：

2.1

説明：

ノードの型を表す値を取得します。

戻り値：

ノードの型を表す，SimpleNode インターフェースに定義されているフィールド値が返されます。フィールド値と各オブジェクトの対応は，この項の冒頭に示す表「SimpleNode インターフェースのフィールドの概要」を参照してください。

(3) getOwnerTaxonomy

```
public SimpleTaxonomy getOwnerTaxonomy()
```

XBRL のバージョン：

2.1

説明：

ノードを所有する SimpleTaxonomy オブジェクトを取得します。ノードの状態がアクティブ／非アクティブに関係なく，このノードを生成した SimpleTaxonomy オブジェクトを返します。

戻り値：

ノードを所有する SimpleTaxonomy オブジェクトを返します。このノードが SimpleTaxonomy の場合は，常に null を返します。

(4) getParentNode

```
public SimpleNode getParentNode()
```

XBRL のバージョン：

2.1

説明：

ノードが関連づけられている親ノードを取得します。

戻り値：

このノードの親ノードを返します。返される親ノードの型は，ノードの型によって次のように異なります。

項番	ノードの型	親ノードの型
1	SIMPLE_TAXONOMY_NODE	常に null を返します。
2	SIMPLE_ITEM_DEFINITION_NODE	SIMPLE_TAXONOMY_NODE
3	SIMPLE_LABEL_NODE	SIMPLE_LABEL_EDGE_NODE
4	SIMPLE_LABEL_EDGE_NODE	SIMPLE_TAXONOMY_NODE

このノードが SimpleTaxonomy の場合は常に null を返します。また，このノードがほかのノードの子ノードとして接続されていない場合も，null を返します。

6.11.8 SimpleTaxonomy インターフェース

```
public interface SimpleTaxonomy
extends SimpleNode
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書全体を表します。

フィールドの概要：

SimpleTaxonomy インターフェースのフィールドの概要を次の表に示します。

表 6-186 SimpleTaxonomy インターフェースのフィールドの概要

項番	説明	
1	static int	ITEM_DEFINITION 単純なタクソノミー文書の要素定義のうちアイテムを表します。
2	static int	LABEL_EDGE 単純なタクソノミー文書のラベルリンクの一辺を表します。

メソッドの概要：

SimpleTaxonomy インターフェースのメソッドの概要を次の表に示します。

表 6-187　SimpleTaxonomy インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLDocumentManager	getOwnerDocumentManager() 単純なタクソノミー文書全体を表す SimpleTaxonomy オブジェクトを所有する，XBRLDocumentManager オブジェクトを取得します。	2.1
2	XBRLTaxonomyDocument	getBaseTaxonomyDocument() 単純なタクソノミー文書の拡張元であるタクソノミー文書を表す，XBRLTaxonomyDocument オブジェクトを取得します。	2.1
3	java.lang.String	getTargetNamespace() 単純なタクソノミー文書を構成する，唯一のタクソノミー本体の targetNamespace 属性値を取得します。	2.1
4	java.lang.String	getPrefix() 単純なタクソノミー文書を構成する唯一のタクソノミー本体の targetNamespace 属性値に対応する，名前空間接頭辞を取得します。	2.1
5	java.lang.String	getSchemaLocation() 単純なタクソノミー文書を構成する唯一のタクソノミー本体が設定されている場所（ファイル名）を取得します。	2.1
6	SimpleItemDefinition	createItemDefinition(java.lang.String name, int type, java.lang.String periodType) 単純なタクソノミー文書を構成するアイテム要素定義を表す，新たな SimpleItemDefinition オブジェクトを生成します。	2.1
7	SimpleElementDefinition[]	findElementDefinitions(int type) 単純なタクソノミー文書を構成する要素定義を表す SimpleElementDefinition オブジェクトの配列を取得します。	2.1
8	SimpleElementDefinition	getElementDefinition(java.lang.String name) 単純なタクソノミー文書を構成する要素定義を表す，SimpleElementDefinition オブジェクトを取得します。	2.1
9	SimpleElementDefinition	addElementDefinition(SimpleElementDefinition obj) 単純なタクソノミー文書に，引数 obj に指定された要素定義を表す SimpleElementDefinition オブジェクトを追加して，アクティブ状態にします。	2.1
10	SimpleElementDefinition	removeElementDefinition(SimpleElementDefinition obj) 単純なタクソノミー文書から，引数 obj に指定された要素定義を表す SimpleElementDefinition オブジェクトを削除し，非アクティブ状態にします。	2.1
11	SimpleLabel	createLabel(java.lang.String role, java.lang.String lang, java.lang.String value) 単純なタクソノミー文書を構成するラベル要素を表す，SimpleLabel オブジェクトを生成します。	2.1

項番	説明		XBRL のバージョン
12	SimpleLabelEdge	createLabelEdge(java.lang.String role, java.lang.String arcrole, java.lang.Object from, java.lang.Object to) 単純なタクソノミー文書を構成するラベルリンクの一つの関係（一辺）を表す SimpleLabelEdge オブジェクトを生成します。	2.1
13	SimpleEdge	createProhibitedEdge(XBRLRelationship obj) 単純なタクソノミー文書を構成する既存の関係の一辺を否定する SimpleEdge オブジェクトを生成します。	2.1
14	SimpleEdge[]	findEdges(int type, java.lang.String role, java.lang.String arcrole, java.lang.Object from, java.lang.Object to) 単純なタクソノミー文書を構成するリンクベースの一つの関係（一辺）を表す SimpleEdge オブジェクトの配列をこの SimpleTaxonomy オブジェクトから取得します。	2.1
15	SimpleEdge	addEdge(SimpleEdge obj) 単純なタクソノミー文書を構成する関係の一辺を表す SimpleEdge オブジェクトを追加します。	2.1
16	SimpleEdge	removeEdge(SimpleEdge obj) 単純なタクソノミー文書を構成する関係の一辺を表す SimpleEdge オブジェクトを削除します。	2.1
17	org.w3c.dom.Document[]	transform() 単純なタクソノミー文書の現在の構成を org.w3c.dom.Document オブジェクトへ変換した，新たな org.w3c.dom.Document オブジェクトの配列を取得します。	2.1

（1） getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書全体を表す SimpleTaxonomy オブジェクトを所有する，XBRLDocumentManager オブジェクトを取得します。

戻り値：

単純なタクソノミー文書全体を表す SimpleTaxonomy オブジェクトを所有する，XBRLDocumentManager オブジェクトを返します。

（2） getBaseTaxonomyDocument

```
public XBRLTaxonomyDocument getBaseTaxonomyDocument()
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書の拡張元であるタクソノミー文書を表す，XBRLTaxonomyDocument オブジェクトを取得します。これによって，SimpleTaxonomy オブジェクトの生成時に指定した XBRLTaxonomyDocument オブジェクトを取得できます。

戻り値：

単純なタクソノミー文書の拡張元であるタクソノミー文書を表す，XBRLTaxonomyDocument オブジェクトを返します。拡張元が存在しない場合には null を返します。

(3) getTargetNamespace

```
public java.lang.String getTargetNamespace()
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成する，唯一のタクソノミー本体の targetNamespace 属性値を取得します。

戻り値：

単純なタクソノミー文書を構成する唯一のタクソノミー本体の targetNamespace 属性値を返します。

(4) getPrefix

```
public java.lang.String getPrefix()
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成する唯一のタクソノミー本体の targetNamespace 属性値に対応する，名前空間接頭辞を取得します。

戻り値：

単純なタクソノミー文書を構成する唯一のタクソノミー本体の targetNamespace 属性値に対応する，名前空間接頭辞を返します。

(5) getSchemaLocation

```
public java.lang.String getSchemaLocation()
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成する唯一のタクソノミー本体が設定されている場所

（ファイル名）を取得します。

戻り値：

単純なタクソノミー文書を構成する唯一のタクソノミー本体が設定されている場所（ファイル名）を返します。

(6) createItemDefinition

```
public SimpleItemDefinition createItemDefinition(java.lang.String
name,
                                                 int type,
                                        java.lang.String periodType)
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成するアイテム要素定義を表す，新たな SimpleItemDefinition オブジェクトを生成します。

例えば，createItemDefinition("balanceSheet", SimpleItemDefinition.MONETARY_ITEM, "duration"); を実行し，SimpleTaxonomy.getPrefix() の結果が「hi」である場合，タクソノミー本体では次のようなアイテム要素定義が作成されます。

```
<element name="balanceSheet"
         id="hi_balanceSheet"
         substitutionGroup="xbrli:item"
         type="xbrli:monetaryItemType"
         xbrli:periodType="duration"/>
```

引数：

name - 単純なタクソノミー文書を構成するアイテム要素定義の name 属性値を指定します。

type - 単純なタクソノミー文書を構成するアイテム要素定義の type 属性値に対応するフィールド値を指定します。指定できる値は，SimpleItemDefinition オブジェクトの setItemType() メソッドと同じです。

periodType - 単純なタクソノミー文書を構成するアイテム要素定義の xbrli:periodType 属性値を指定します。指定できる値は instant または duration です。

戻り値：

新たに生成された SimpleItemDefinition オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 name または引数 periodType に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 name に空文字列（""）または xsd:NCName として不正な文字列が指定され

た場合にスローされます。［エラーコード：20108］

- 引数 type に指定できない値が指定された場合にスローされます。［エラーコード：20156］
- 引数 periodType に instant または duration 以外の文字列が指定された場合にスローされます。［エラーコード：20014］

（7） findElementDefinitions

```
public SimpleElementDefinition[] findElementDefinitions(int type)
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成する要素定義を表す SimpleElementDefinition オブジェクトの配列を取得します。

引数：

type - 取得する要素定義の種別を指定します。指定できる値は ITEM_DEFINITION だけです。この指定によってオブジェクト SimpleItemDefinition が配列に含まれる可能性があります。

戻り値：

単純なタクソノミー文書を構成する要素定義を表す，SimpleElementDefinition オブジェクトの配列を返します。引数 type に指定された種別の要素定義がこの単純なタクソノミー文書全体を表す SimpleTaxonomy オブジェクトに一つも存在しない場合は，要素数 0 の配列を返します。

例外：

XBRLDOMRuntimeException -

- 引数 type に指定できない値が指定された場合にスローされます。［エラーコード：20156］

（8） getElementDefinition

```
public SimpleElementDefinition
getElementDefinition(java.lang.String name)
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成する要素定義を表す，SimpleElementDefinition オブジェクトを取得します。

このバージョンでは，取得した SimpleElementDefinition オブジェクトは常に SimpleItemDefinition オブジェクトだけにキャストできます。

SimpleTaxonomy オブジェクト（単純なタクソノミー文書全体を表す）に，引数

name に指定された name 属性値を持つ要素定義が存在しない場合は，null を返します。

引数：

name - 単純なタクソノミー文書を構成する要素定義の name 属性値を指定します。

戻り値：

単純なタクソノミー文書を構成する要素定義を表す，SimpleItemDefinition オブジェクトを返します。
SimpleTaxonomy オブジェクト（単純なタクソノミー文書全体を表す）に，引数 name に指定された name 属性値を持つ要素定義が存在しない場合は，null を返します。

例外：

XBRLDOMRuntimeException -

- 引数 name に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 name に空文字列（""）または xsd:NCName として不正な文字列が指定された場合にスローされます。［エラーコード：20108］

（9）addElementDefinition

```
public SimpleElementDefinition
addElementDefinition(SimpleElementDefinition obj)
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書に，引数 obj に指定された要素定義を表す SimpleElementDefinition オブジェクトを追加して，アクティブ状態にします。

引数：

obj - 新たに追加する SimpleElementDefinition オブジェクトを指定します。

戻り値：

新たに追加した SimpleElementDefinition オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 obj に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 obj に指定した SimpleElementDefinition オブジェクトの name 属性値と同じ name 属性値を持つアクティブな SimpleElementDefinition オブジェクトがこの SimpleTaxonomy オブジェクトに存在する場合，または同じ対象名前空間で同じ name 属性値を持つ XBRLTaxonomyElement オブジェクトが拡張元である XBRLTaxonomyDocument オブジェクトに存在する場合にスローされます。［エラーコード：22002］

- 引数 obj に指定した SimpleElementDefinition オブジェクトの id 属性値と同じ id 属性値を持つ，アクティブな SimpleElementDefinition オブジェクトがこの SimpleTaxonomy オブジェクトに存在する場合にスローされます。［エラーコード：22003］
- 引数 obj に指定した SimpleElementDefinition オブジェクトがこの SimpleTaxonomy オブジェクトから生成されたオブジェクトでない場合に，スローされます。［エラーコード：22008］

（10）removeElementDefinition

```
public SimpleElementDefinition
removeElementDefinition(SimpleElementDefinition obj)
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を表す SimpleTaxonomy オブジェクトから，引数 obj に指定された要素定義を表す SimpleElementDefinition オブジェクトを削除し，非アクティブ状態にします。

また，引数 obj に指定された SimpleElementDefinition オブジェクトがこの SimpleTaxonomy オブジェクトに存在するアクティブ状態の SimpleEdge オブジェクトから参照されている場合には，関連するすべての SimpleNode オブジェクトも非アクティブ状態に変更します。ただし，この SimpleElementDefinition オブジェクト以外の SimpleElementDefinition オブジェクトのアクティブ状態は変更しません。

アクティブ状態が変更される可能性のあるオブジェクトは，SimpleEdge オブジェクトおよび SimpleLabel オブジェクトです。

引数：

obj - 削除する SimpleElementDefinition オブジェクトを指定します。

戻り値：

単純なタクソノミー文書から削除された SimpleElementDefinition オブジェクトを返します。削除する要素定義がこの SimpleTaxonomy オブジェクトに存在しない場合には null を返します。

例外：

XBRLDOMRuntimeException -

- 引数 obj に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 obj に指定した SimpleElementDefinition オブジェクトが，この SimpleTaxonomy オブジェクトから生成されたオブジェクトでない場合にスローされます。［エラーコード：22008］

(11) createLabel

```
public SimpleLabel createLabel(java.lang.String role,
                               java.lang.String lang,
                               java.lang.String value)
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成するラベル要素を表す，SimpleLabel オブジェクトを生成します。

引数：

role - 新たに作成するラベル要素の xlink:role 属性値を指定します。指定できる値は SimpleLabel.setRole() メソッドと同じです。省略する場合は null を指定します。

lang - 新たに作成するラベル要素の xml:lang 属性値を指定します。

value - 新たに作成するラベル要素の内容を指定します。

戻り値：

新たに生成された SimpleLabel オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 lang または引数 value に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 role に指定できない値が指定されている場合にスローされます。［エラーコード：20156］

(12) createLabelEdge

```
public SimpleLabelEdge createLabelEdge(java.lang.String role,
                                       java.lang.String arcrole,
                                       java.lang.Object from,
                                       java.lang.Object to)
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成するラベルリンクの一つの関係（一辺）を表す SimpleLabelEdge オブジェクトを生成します。

引数：

role - 新たに作成するラベルリンクの一つの関係が属する，link:labelLink の role 属性値を指定します。指定できるカスタムロール値と標準値は次のとおりです。

指定できるカスタムロール値

ここで指定できるカスタムロールは，拡張元である XBRLTaxonomyDocument オブジェクトの getRoleTypes() メソッドで XBRLRoleType オブジェクトを取

得した際，getUsedOn() メソッドで取得されるリストが次の値を持つものだけです。

- 名前空間：http://www.xbrl.org/2003/linkbase
- ローカル名：labelLink

この条件に該当するカスタムロールについて，XBRLRoleType オブジェクトの getRoleURI() メソッドで取得される値を指定します。

指定できる標準値

http://www.xbrl.org/2003/role/link

arcrole - 新たに作成する <link:labelArc> 要素の xlink:arcrole 属性値を指定します。指定できるカスタムアークロール値と標準値は次のとおりです。

指定できるカスタムアークロール値

ここで指定できるカスタムアークロールは，拡張元である XBRLTaxonomyDocument オブジェクトの getArcroleTypes() メソッドで XBRLArcroleType オブジェクトを取得した際，getUsedOn() メソッドで取得されるリストが次の値を持つものだけです。

- 名前空間：http://www.xbrl.org/2003/linkbase
- ローカル名：labelArc

この条件に該当するカスタムアークロールについて，XBRLArcroleType オブジェクトの getArcroleURI() メソッドで取得される値を指定します。

指定できる標準値

http://www.xbrl.org/2003/arcrole/concept-label

from - 新たに作成する <link:labelArc> 要素の xlink:from 属性値が参照するオブジェクトを指定します。指定できるオブジェクトは次のとおりです。

- SimpleConceptDefinition
- XBRLTaxonomyItem
- XBRLTaxonomyTuple

to - 新たに作成する <link:labelArc> 要素の xlink:to 属性値が参照するオブジェクトを指定します。指定できる値は SimpleLabel だけです。

戻り値：

新たに生成された SimpleLabelEdge オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 role，引数 arcrole，引数 from または引数 to に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 role または引数 arcrole に対し，指定できない値が指定されている場合にスローされます。［エラーコード：20156］
- 引数 from または引数 to に対し，指定できないオブジェクトが指定されている場合にスローされます。［エラーコード：22001］

- 引数 from に指定されたオブジェクトが XBRLTaxonomyItem オブジェクトまたは XBRLTaxonomyTuple オブジェクトであり，かつそのオブジェクトを所有する XBRLTaxonomyDocument オブジェクトがこの SimpleTaxonomy オブジェクトの生成時に指定された XBRLTaxonomyDocument オブジェクトと異なる場合にスローされます。［エラーコード：22007］
- 引数 from または引数 to に指定されたオブジェクトが SimpleConceptDefinition オブジェクトまたは SimpleLabel オブジェクトであり，かつそのオブジェクトを所有する SimpleTaxonomy オブジェクトがこの SimpleTaxonomy オブジェクトと異なる場合にスローされます。［エラーコード：22008］
- 引数 to に指定した SimpleLabel オブジェクトがすでに別の SimpleLabelEdge オブジェクトの子ノードである場合にスローされます。［エラーコード：20005］

（13）createProhibitedEdge

```
public SimpleEdge createProhibitedEdge(XBRLRelationship obj)
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成する既存の関係の一辺を否定する SimpleEdge オブジェクトを生成します。

引数：

obj - 指定できるオブジェクトは XBRLLabelRelationship だけです。それによって生成される SimpleEdge オブジェクトは SimpleLabelEdge です。

戻り値：

新たに生成された SimpleEdge オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 obj に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 obj に指定された XBRLRelationship オブジェクトを所有する XBRLTaxonomyDocument オブジェクトが，この SimpleTaxonomy オブジェクトの生成時に指定された XBRLTaxonomyDocument オブジェクトと異なる場合にスローされます。［エラーコード：22007］
- 引数 obj に XBRLLabelRelationship オブジェクト以外が指定された場合にスローされます。［エラーコード：22001］

（14）findEdges

```
public SimpleEdge[] findEdges(int type,
                              java.lang.String role,
                              java.lang.String arcrole,
                              java.lang.Object from,
                              java.lang.Object to)
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成するリンクベースの一つの関係（一辺）を表す SimpleEdge オブジェクトの配列をこの SimpleTaxonomy オブジェクトから取得します。

引数：

type - 取得するリンクの一辺を表す値を指定します。指定できる値は LABEL_EDGE だけです。この指定によって，オブジェクト SimpleLabelEdge が戻り値の配列に含まれる可能性があります。

role - 取得するリンクが属する拡張リンクの role 属性値を指定します。取得条件に加えない場合には null を指定します。

arcrole - 取得するリンクのアーク型要素の xlink:arcrole 属性値を指定します。取得条件に加えない場合には null を指定します。

from - 取得するリンクのアーク型要素の xlink:from 属性値が参照するオブジェクトを指定します。取得条件に加えない場合には null を指定します。

to - 取得するリンクのアーク型要素の xlink:to 属性値が参照するオブジェクトを指定します。取得条件に加えない場合には null を指定します。

戻り値：

単純なタクソノミー文書を構成するリンクベースの一つの関係（一辺）を表す SimpleEdge オブジェクトの配列を返します。単純なタクソノミー文書全体を表す SimpleTaxonomy オブジェクトに，引数に指定された条件を満たす SimpleEdge オブジェクトが一つも存在しない場合は，要素数 0 の配列を返します。

例外：

XBRLDOMRuntimeException -

- 引数 type に指定できない値が指定されている場合にスローされます。［エラーコード：20156］

(15) addEdge

```
public SimpleEdge addEdge(SimpleEdge obj)
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成する関係の一辺を表す SimpleEdge オブジェクトを追加します。

同じ関係（一辺）を表すアクティブ状態の SimpleEdge オブジェクトが存在した場合，既存の SimpleEdge オブジェクトが最初に SimpleTaxonomy オブジェクトから削除され，引数 obj に指定した SimpleEdge オブジェクトが追加されます。

また，引数 obj に指定した SimpleEdge オブジェクトが保持する SimpleConceptDefinition オブジェクトが非アクティブ状態の場合は，アクティブ状態に変更します。アクティブ状態に変更できなかった場合は，引数 obj に指定された SimpleEdge オブジェクトの追加に失敗します。発生する例外の種類は，addElementDefinition() メソッドと同じです。

引数：

obj - 新たに追加する SimpleEdge オブジェクトを指定します。

戻り値：

新たに追加した SimpleEdge オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 obj に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 obj に SimpleEdge オブジェクトの SimpleLabelEdge を指定した場合に，そのオブジェクトが保持する SimpleLabel オブジェクトの id 属性値を持つ異なるアクティブな SimpleLabel オブジェクトがこの SimpleTaxonomy オブジェクトに存在する場合にスローされます。［エラーコード：22003］
- 引数 obj に指定した SimpleEdge オブジェクトが，この SimpleTaxonomy オブジェクトから生成されたオブジェクトでない場合にスローされます。［エラーコード：22008］

（16）removeEdge

```
public SimpleEdge removeEdge(SimpleEdge obj)
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を構成する関係の一辺を表す SimpleEdge オブジェクトを削除します。ただし，この SimpleEdge オブジェクトが保持する SimpleConceptDefinition オブジェクトのアクティブ状態は変更されないため，オブジェクトを非アクティブ状態へ変えるには removeElementDefinition() メソッドを使用してください。

また，引数 obj が SimpleLabelEdge オブジェクトの場合，この SimpleLabelEdge オブジェクトが参照する SimpleLabel オブジェクトは非アクティブ状態となります。

引数：

obj - 削除する SimpleEdge オブジェクトを指定します。

戻り値：

単純なタクソノミー文書から削除した SimpleEdge オブジェクトを返します。削除

する SimpleEdge オブジェクトがこの SimpleTaxonomy オブジェクトに存在しない場合は，null を返します。

例外：

XBRLDOMRuntimeException -

- 引数 obj に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 obj に指定した SimpleEdge オブジェクトがこの SimpleTaxonomy オブジェクトから生成されたオブジェクトでない場合にスローされます。［エラーコード：22008］

(17) transform

```
public org.w3c.dom.Document[] transform()
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書の現在の構成を org.w3c.dom.Document オブジェクトへ変換した，新たな org.w3c.dom.Document オブジェクトの配列を返します。

配列の要素数は常に 6 です。ドキュメントの格納順序は次のとおりです。

インデックス	格納されるドキュメント
0	タクソノミー本体
1	ラベルリンクベース（ラベルリンクが一つも存在しない場合には null）
2	null
3	null
4	null
5	null

戻り値：

変換後の org.w3c.dom.Document オブジェクトの配列を返します。

6.11.9 SimpleTaxonomyBuilder インターフェース

```
public interface SimpleTaxonomyBuilder
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を表す SimpleTaxonomy オブジェクトを生成します。

メソッドの概要：

SimpleTaxonomyBuilder インターフェースのメソッドの概要を次の表に示します。

表 6-188　SimpleTaxonomyBuilder インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLDocumentManager	getOwnerDocumentManager() SimpleTaxonomyBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。	2.1
2	SimpleTaxonomy	newSimpleTaxonomy(java.lang.String targetNamespace, java.lang.String prefix, java.lang.String fileName, XBRLTaxonomyDocument doc) 単純なタクソノミー文書を表す SimpleTaxonomy オブジェクトを新規生成します。	2.1

(1) getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.1

説明：

SimpleTaxonomyBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。

戻り値：

SimpleTaxonomyBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトを返します。

(2) newSimpleTaxonomy

```
public SimpleTaxonomy newSimpleTaxonomy(java.lang.String
targetNamespace,
                                        java.lang.String prefix,
                                        java.lang.String fileName,
                                        XBRLTaxonomyDocument doc)
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を表す SimpleTaxonomy オブジェクトを新規生成します。このメソッドを使って新しいインスタンス文書が生成された場合は，引数 doc に指定した XBRLTaxonomyDocument オブジェクトを構築する際の出発点となったタクソノミー本体およびリンクベースのリストを XBRLTaxonomyDocument.getStartingPoints() メソッドから取得します。その際，取得されるタクソノミー本体の targetNamespace 属性値と引数 targetNamespace の指定値とが同じ場合，取得されたタクソノミー本体は <xsd:include> 要素によって単純なタクソノミー文書を構成する唯一のタクソノミー本体に取り込まれます。targetNamespace 属性値が異なるタクソノミー本体は <xsd:import> 要素によって

取り込まれます。
引数 doc に指定された XBRLTaxonomyDocument オブジェクトでリンクベースの遅延解析が有効であり，かつリンクが未解析の場合は，SimpleTaxonomy オブジェクトの生成処理の前に，リンクベースの遅延解析が実行されます。

引数：

targetNamespace - 新たに生成する唯一のタクソノミー本体の targetNamespace 属性値を指定します。XBRL2.1 の名前空間など，予約されている名前空間は指定できません。予約されている名前空間については，「3.1.1(1)　タクソノミー文書の作成時に自動作成される内容」を参照してください。
prefix - 新たに生成する唯一のタクソノミー本体の targetNamespace 属性値に対応する，名前空間接頭辞を指定します。
fileName - 新たに生成する唯一のタクソノミー本体のファイル名を指定します。拡張子は「xsd」にしてください（例：sample_taxonomy.xsd）。ただし，指定されたファイル名が，システムで使用できるファイル名かについてはチェックされません。
doc - 単純なタクソノミー文書を生成する際の，拡張元となるタクソノミー文書を表す XBRLTaxonomyDocument オブジェクトを指定します。生成された SimpleTaxonomy オブジェクトは，ここで指定した XBRLTaxonomyDocument オブジェクトの複製を参照します。新規作成の場合には拡張元を必要としないため，null を指定します。

戻り値：

新たに生成された SimpleTaxonomy オブジェクトを返します。

例外：

XBRLDOMRuntimeException -

- 引数 targetNamespace，引数 prefix または引数 fileName に null を指定した場合にスローされます。［エラーコード：20002］
- 引数 targetNamespace が xsd:anyURI 型の値でない場合にスローされます。［エラーコード：20108］
- 引数 prefix が xsd:NCName 型の値でない場合にスローされます。［エラーコード：20108］
- 引数 targetNamespace に空文字列（""）を指定した場合にスローされます。［エラーコード：20110］
- 引数 targetNamespace または引数 prefix に不正な値（予約されている名前空間）が指定されたか，引数 fileName に不正なファイル名が指定された場合にスローされます。［エラーコード：20156］
- 引数 doc に指定された XBRLTaxonomyDocument オブジェクトを所有する XBRLDocumentManager オブジェクトと，この SimpleTaxonomyBuilder オブジェクトを所有する XBRLDocumentManager オブジェクトとが一致しない場合にスローされます。［エラーコード：20051］
- 引数 doc に指定された XBRLTaxonomyDocument オブジェクトが準拠する XBRL

のバージョンと，この SimpleTaxonomyBuilder オブジェクトが処理対象とする XBRL のバージョンとが一致しない場合にスローされます。［エラーコード：20103］

- 引数 doc に指定されたタクソノミー文書の基点となるタクソノミー本体の中に，ほかの XML 文書の一部となっているタクソノミー本体が存在する場合に，スローされます。［エラーコード：22010］
- 引数 doc に指定された XBRLTaxonomyDocument オブジェクトが準拠する XBRL Dimensions のバージョンと，この SimpleTaxonomyBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョンが異なる場合にスローされます。［エラーコード：22103］

6.11.10 SimpleTaxonomyBuilderFactory インターフェース

```
public interface SimpleTaxonomyBuilderFactory
```

XBRL のバージョン：

2.1

説明：

単純なタクソノミー文書を作成するための SimpleTaxonomyBuilder オブジェクトを生成します。SimpleTaxonomyBuilderFactory オブジェクトは，XBRLDocumentManager.newSimpleTaxonomyBuilderFactory() メソッドによってインスタンス化されます。

メソッドの概要：

SimpleTaxonomyBuilderFactory インターフェースのメソッドの概要を次の表に示します。

表 6-189 SimpleTaxonomyBuilderFactory インターフェースのメソッドの概要

項番	説明		XBRL のバージョン
1	XBRLDocumentManager	getOwnerDocumentManager() SimpleTaxonomyBuilderFactory オブジェクトを所有する XBRLDocumentManager オブジェクトを取得します。	2.1
2	SimpleTaxonomyBuilder	newSimpleTaxonomyBuilder() 現在の設定で SimpleTaxonomyBuilder オブジェクトを生成します。	2.1

（1） getOwnerDocumentManager

```
public XBRLDocumentManager getOwnerDocumentManager()
```

XBRL のバージョン：

2.1

説明：

SimpleTaxonomyBuilderFactory オブジェクトを所有する
XBRLDocumentManager オブジェクトを取得します。

戻り値：

SimpleTaxonomyBuilderFactory オブジェクトを所有する
XBRLDocumentManager オブジェクトを返します。

(2) newSimpleTaxonomyBuilder

```
public SimpleTaxonomyBuilder newSimpleTaxonomyBuilder()
```

XBRL のバージョン：

2.1

説明：

現在の設定で SimpleTaxonomyBuilder オブジェクトを生成します。

戻り値：

新たに生成された SimpleTaxonomyBuilder オブジェクトを返します。

7 メッセージ

この章では，uCosminexus Business Reporting Processor が提供する API 利用時に表示されるメッセージについて説明します。

7.1 **メッセージの形式**

7.2 **メッセージ一覧**

7.3 **メッセージ** KEBR10000-KEBR19999

7.4 **メッセージ** KEBR20000-KEBR29999

7.5 **メッセージ** KEBR30000-KEBR39999

7.6 **メッセージ** KEXL00000-KEXL09999

7.7 **メッセージ** KEXL10000-KEXL19999

7.1 メッセージの形式

このマニュアルでは，メッセージを次の形式で記載しています。

メッセージ ID （メッセージの出力先）

メッセージテキスト

要因：

メッセージの内容を説明します。

対策：

メッセージ確認時，ユーザーがする必要のある処置を示します。

メッセージ ID およびメッセージの出力先の詳細について説明します。

メッセージ ID

各メッセージには，次の規則に従った形式のメッセージ ID が付いています。

記述形式：KEBRnnnnn-Z

KEBR

uCosminexus Business Reporting Processor に割り振られたプリフィックスです。

nnnnn

メッセージごとのメッセージ番号です。5 桁の数字です。

メッセージ番号と，対応する機能を次に示します。

10000-19999：XBRL のパース機能

20000-29999：XBRL DOM の API

30000-39999：XBRL DOM の初期化機能

-Z

E：（Error） エラーメッセージ

W：（Warning） 警告メッセージ

I：（Information）通常のメッセージ

記述形式：KEXLnnnnn-E

KEXL

XLink プロセッサに割り振られたプリフィックスです。

nnnnn

メッセージごとのメッセージ番号です。5 桁の数字です。

メッセージ番号と，対応する機能を次に示します。

00000-09999：XLink のエラーメッセージ

10000-19999：XPointer のエラーメッセージ

「E」はエラーメッセージであることを表します。

メッセージの出力先

メッセージが出力される場所です。
F：ログファイル
V：例外に含めるメッセージ

KEXL で始まるメッセージは，KEBR で始まるメッセージの詳細メッセージとして含まれるメッセージのため，メッセージの出力先は記述していません。

7.2 メッセージ一覧

メッセージ ID 順に，メッセージを一覧で示します。

KEBR10000-E　{0} 要素の {1} 属性に指定できない値が指定されました。{2} 個所 ={3} 指定できる値 ={4} 指定された値 ={5}

KEBR10001-E　{0} 要素の {1} 属性に推奨値以外の値が指定されました。{2} 個所 ={3} 指定された値 ={4}

KEBR10002-E　{0} 要素に {1} 要素が存在しません。{2} 個所 ={3}

KEBR10003-E　{0} 要素に {1} 属性が存在しません。{2} 個所 ={3}

KEBR10004-W　{0} 要素に不要な {1} 要素が指定されています。{2} 個所 ={3}

KEBR10005-W　{0} 要素に不要な {1} 属性が指定されています。{2} 個所 ={3}

KEBR10006-E　{0} 要素の {1} 属性の参照先が存在しません。{2} 個所 ={3} 参照先 ={4}

KEBR10007-E　{0} 要素の {1} 属性の値が重複しています。{2} 個所 ={3} 指定された値 ={4}

KEBR10009-E　{0} 要素に使用できない値が指定されました。{1} 個所 ={2} 指定された値 ={3}

KEBR10010-E　period 要素の期間が範囲で指定されていません。{0} 個所 ={1} startDate={2} endDate={3}

KEBR10012-W　リンクが双方向で記述されていません。{0} 個所 ={1}

KEBR10013-E　双方向で記述されたリンクの {0} 属性の値が異なります。{1} リンク１：個所 ={2}, 指定された値 ={3} リンク２：個所 ={4}, 指定された値 ={5}

KEBR10014-W　{0} 対象のリンクが存在しません。{1} 個所 ={2}

KEBR10016-E　指定した xlink:role 属性の値と実際のリンク先の内容が一致していません。{0} 個所 ={1} xlink:role 属性の値 ={2} リンク先の内容 ={3}

KEBR10017-I　group 要素の子要素で記述されている group 要素を無視しました。{0} 個所 ={1}

KEBR10018-E　{0} 要素の {1} 属性に指定されたラベル名に対応する要素が見つかりません。{2} 個所 ={3}

KEBR10019-E　{0} 要素の {1} 属性に指定されたラベル名の要素が {2} ではありません。{3} 個所 ={4}

KEBR10020-W　{0}, id={1} のコンテキスト要素に対応するアイテム要素がありません。

{2} 個所 ={3}

KEBR10021-E　{0} 要素の {1} 属性に {2} 型以外の値が指定されました。{3} 個所 ={4}

KEBR10024-E　{0} 要素の priority={1} を持つ同一のリンクが複数存在します。要素 1：{2}, 個所 ={3} 要素 2：{4}, 個所 ={5}

KEBR10025-E　{0} 要素の {1} 属性に使用できない値が指定されました。{2} 個所 ={3} 指定された値 ={4}

KEBR10029-E　同じ {0} 要素は複数存在できません。{1} 個所 ={2}

KEBR10030-E　XML 文書中で指定されている URI が RFC2396 に違反しています。{0} 個所 ={1} 指定されていた URI={2}

KEBR10031-E　parse() メソッドに指定された URI が RFC2396 に違反しています。指定されていた URI={0}

KEBR10032-E　同一拡張リンク中に同じ弧を持つアーク型要素が指定されています。{0} 個所 ={1}

KEBR10033-E　ルート要素に指定できない要素が指定されました。{0} 指定できる要素 ={1} 指定された要素 ={2}

KEBR10034-E　{0} 要素の値が存在しません。{1} 個所 ={2}

KEBR10035-E　period 要素の子要素 {0} と {1} の関係が正しくありません。{2} 個所 ={3}

KEBR10036-E　{0} 要素の値 {1} が {2} 型ではありません。{3} 個所 ={4}

KEBR10037-E　{0} 要素の {1} 属性の値 {2} が {3} 型ではありません。{4} 個所 ={5}

KEBR10038-E　{0} 要素はタクソノミー文書で定義されていません。{1} 個所 ={2}

KEBR10039-E　{0} の名前空間が宣言されていません。{1}

KEBR10040-E　{0} 要素がパート要素の代替要素としてタクソノミー本体で定義されていません。{1} 個所 ={2}

KEBR10041-E　{0} 要素の {1} 属性が絶対 URI で記述されていません。{2} 個所 ={3}

KEBR10042-W　同じ内容の単純リンクが指定されました。要素 1：{0}, 個所 ={1} 要素 2：{2}, 個所 ={3}

KEBR10043-W　タクソノミー本体を拡張する際に import 要素で記述した URI がループしています。{0} 個所 ={1}

KEBR10044-E　アイテムには子要素を記述できません。{0} 個所 ={1}

KEBR10045-E　{0} 要素はインスタンス文書では使用できません。{1} 個所 ={2}

KEBR10046-E　{0} 要素の {1} 属性に標準値以外の値が指定されました。{2} 個所 ={3} 指定された値 ={4}

KEBR10047-E　{0} 要素の {1} 属性に標準値またはカスタムされた値以外が指定されました。{2} 個所 ={3} 指定された値 ={4}

KEBR10048-E　{0} 要素に記述された値はすでに記述されているため無視しました。{1} 個所 ={2} 指定された値 ={3}

KEBR10049-I　{1} を定義している {0} 要素が複数存在しました。{2} 個所 ={3}

KEBR10050-E　{0} 要素で {1} を再定義しました。{2} 個所 ={3}

KEBR10051-E　{0} 要素が参照する context 要素内の period 要素の指定が誤っています。periodType 属性の値 ={1} 指定できる要素 ={2} {3} 個所 ={4}

KEBR10052-W　タクソノミー本体を拡張する際に import 要素または include 要素で記述した URI がループしています。{0} 個所 ={1}

KEBR10053-E　{0} 属性の範囲に誤りがあります。xlink:from が指す要素の {1} 属性の値 ={2} xlink:to が指す要素の {1} 属性の値 ={3} {4} 個所 ={5}

KEBR10054-E　{0} 要素のアークロール {1} に対応する要素の組み合わせに誤りがあります。xlink:from が指す要素の {2} 属性の値 ={3} xlink:to が指す要素の {2} 属性の値 ={4} {5} 個所 ={6}

KEBR10056-E　{0} 要素の参照先が目的の文書を指定していません。{1} 個所 ={2}

KEBR10057-E　{0} 要素の子要素に {1} 要素の記述はできません。{2} 個所 ={3}

KEBR10058-E　{0} 型の {1} 要素に対応する {2} 要素に誤った値が記述されました。誤った値 ={3} {4} 個所 ={5}

KEBR10059-E　{0} 要素に指定できない値が記述されました。{1} 個所 ={2} 指定されていた値 ={3}

KEBR10061-E　{0} 属性に使用できない値が記述されています。{1} 個所 ={2}

KEBR10062-E　{0} 要素内には {1} 要素の記述ができません。{2} 個所 ={3}

KEBR10064-E　要素宣言 {0} で記述できない {1} 属性を指定しています。{2} 個所 ={3}

KEBR10065-E　要素宣言 {0} で {1} 属性が記述されていません。{2} 個所 ={3}

KEBR10066-E　{0} 要素の値と参照先の {1} 要素の値が一致しません。{2} 個所 ={3}

KEBR10067-E　{0} 要素の記述個所が誤っています。{1} 個所 ={2}

KEBR10068-E　{0} 要素には precision 属性と decimals 属性を同時に指定できません。{2} 個所 ={3}

KEBR10069-E　{1} 属性の値が同じ {0} 要素は複数定義できません。要素 1：{2} 個所 ={3} 要素 2：{4} 個所 ={5} 指定された値 ={6}

KEBR10070-E　接頭辞 {0} に対応する名前空間 URI が宣言されていません。{1} 個所 ={2}

KEBR10071-E　{0} 要素で利用できない要素を関連づけています。{1} 個所 ={2} 参照先 ={3}

KEBR10072-E　{0} 要素に記述された値を無視しました。{1} 個所 ={2}

KEBR10074-W　{0} 要素の priority={1} を持つ同一のリンクが複数存在します。要素 1：{2}, 個所 ={3} 要素 2：{4}, 個所 ={5}

KEBR10075-E　{0} 要素は XBRL ではサポートされていません。{1} 個所 ={2}

KEBR10076-E　{0} 属性に空文字列は指定できません。{1} 個所 ={2}

KEBR10077-E　{0} 要素の {1} 属性の値 {2} に対応する要素宣言の候補が複数存在します。{3} 個所 ={4} 候補 1:{5} 個所 ={6} 候補 2:{7} 個所 ={8}

KEBR10101-W　ユーザーデータ型の定義が同一タクソノミー本体内で定義されていません。{0} 個所 ={1} 使用している型 ={2}

KEBR10102-W　type 属性が省略されているか，サポートされていない形式のユーザーデータ型で定義しています。{0} 個所 ={1}

KEBR10103-E　{0} 要素の {1} 属性の参照先は利用できません。{2} 個所 ={3} 参照先 ={4}

KEBR10105-W　指定されたスキーマ文書のうちタクソノミー本体でないものがありました。{0}

KEBR10106-E　拡張されたタクソノミー文書が最大階層数を超えました。{0}

KEBR10107-E　{0} 属性にタクソノミー文書が指定されていません。{1} 個所 ={2}

KEBR10108-W　アイテムでもタプルでもない要素を参照しています。{0} 個所 ={1} 参照先 ={2}

KEBR10109-W　インスタンス文書で利用できない要素を参照しています。{0} 個所 ={1} 参照先 ={2}

KEBR10110-E　{0} 要素が含まれていないため解析できません。{1}

KEBR10111-W　{0} 要素以下の記述を無視しました。{1} 個所 ={2}

KEBR10112-E　ユーザーデータ型の定義が見つかりません。{0} 個所 ={1} 使用している型 ={2}

KEBR10113-E　タクソノミー文書に含めることのできる XML 文書 (タクソノミー本

体・リンクベース) が最大ファイル数を超えました。最大ファイル数 ={0}

KEBR10115-W　サポートされていないロールが記述されたためリンク先の文書を無視しました。{0} 個所 ={1} ロール ={2}

KEBR10116-E　アーク型要素 {0} が参照するロケータ型要素ではリソース型要素を指してはいけません。{1} 個所 ={2}

KEBR10117-W　ロール {0} はサポートしていません。{1} 個所 ={2}

KEBR11000-I　XBRL 文書にアクセスします。{0} プロセッサ ={1}

KEBR11001-I　XBRL 文書を読み込みました。{0} プロセッサ ={1}

KEBR11002-E　XBRL 文書の読み込みに失敗しました。{0} プロセッサ ={1} 詳細情報 ={2}

KEBR11003-E　XBRL 文書の解析に失敗しました。{0} プロセッサ ={1} 詳細情報 ={2}

KEBR11004-W　XBRL 文書の解析で警告が発生しました。行数 ={0} カラム位置 ={1} 公開識別子 ={2} システム識別子 ={3} 詳細情報 ={4}

KEBR11005-E　XBRL 文書の解析でエラーが発生しました。行数 ={0} カラム位置 ={1} 公開識別子 ={2} システム識別子 ={3} 詳細情報 ={4}

KEBR11006-E　XBRL 文書の解析で致命的エラーが発生しました。行数 ={0} カラム位置 ={1} 公開識別子 ={2} システム識別子 ={3} 詳細情報 ={4}

KEBR11007-E　XPointer の記述に誤りがあります。{0} 個所 ={1} XPointer 式の内容 ={2} 詳細メッセージ ={3}

KEBR11008-W　同じロールの同じ言語の label 要素を一つの要素宣言に複数記述しています。{0} 個所 ={1} 要素宣言 ={2}:{3}

KEBR11009-E　XPointer の記述に誤りがあります。XPointer 式の内容 ={0} 詳細メッセージ ={1}

KEBR11010-E　参照先が存在しません。参照先 ={0}

KEBR11011-E　参照先が目的の文書を指定していません。参照先 ={0}

KEBR11100-I　タクソノミー文書のキャッシュを利用可能にしました。最大キャッシュ数 ={0} 最小キャッシュ数 ={1}

KEBR11101-I　タクソノミー文書のキャッシュは行いません。

KEBR11102-I　キャッシュをクリアしました。

KEBR11103-I　タクソノミー文書をキャッシュに追加しました。{0}

KEBR11104-I　タクソノミー文書をキャッシュから削除しました。{0}

KEBR11105-I　キャッシュからタクソノミー文書を取得しました。{0}

KEBR11106-I　キャッシュ内のタクソノミー文書を更新しました。{0}

KEBR11107-I　XML 文書をキャッシュに追加しました。{0}

KEBR11108-I　XML 文書をキャッシュから削除しました。{0}

KEBR11109-I　キャッシュの XML 文書の参照を開始しました。{0} 参照回数 ={1}

KEBR11110-I　キャッシュの XML 文書の参照を終了しました。{0} 参照回数 ={1}

KEBR11111-I　XLink 文書をキャッシュに追加しました。{0}

KEBR11112-I　XLink 文書をキャッシュから削除しました。{0}

KEBR11113-I　キャッシュの XLink 文書の参照を開始しました。{0} 参照回数 ={1}

KEBR11114-I　キャッシュの XLink 文書の参照を終了しました。{0} 参照回数 ={1}

KEBR11115-I　XML 文書をキャッシュから参照しました。{0}

KEBR11200-I　{0} 処理を開始しました。{1}

KEBR11201-I　{0} 処理が終了しました。{1}

KEBR11202-I　{0} 処理を開始しました。{1} {2}

KEBR11203-I　{0} 処理が終了しました。{1} 返値 ={2}

KEBR11204-I　{0} の解析処理を開始しました。{1}

KEBR11205-I　{0} の解析処理が終了しました。{1}

KEBR11301-E　XBRL 文書の解析で致命的エラーが発生しました。詳細情報 ={0}

KEBR15001-E　{0} の構造が XBRL 仕様に定められている形式に違反しています。{1} 個所 ={2}

KEBR15002-E　ユーザーデータ型は構造を持つことができません。{0} 個所 ={1}

KEBR15101-E　カスタムアークロール {0} の定義が見つからないためサイクルの検証ができません。

KEBR15102-E　DTS 内にカスタムアークロール {0} の定義が複数あるためサイクルの検証ができません。

KEBR15103-I　経路 {0}:{1} {2} 個所 ={3}

KEBR15104-E　経路が循環しています。cyclesAllowed={0} アークロール ={1} {2} のロー

ル ={3} 経路 ={4}

KEBR15105-E　経路が複数あります。cyclesAllowed={0} アークロール ={1} {2} のロール ={3} 経路 ={4}

KEBR15111-E　{0} と {1} の型が異なります。{2} のロール ={3} アイテム 1:{4}, 個所 ={5}, 型名 ={6} アイテム 2:{7}, 個所 ={8}, 型名 ={9} リンク：{10}, 個所 ={11}

KEBR15112-E　{0} と {1} の {2} 属性の値が異なります。{3} のロール ={4} アイテム 1:{5}, 個所 ={6}, 属性値 ={7} アイテム 2:{8}, 個所 ={9}, 属性値 ={10} リンク：{11}, 個所 ={12}

KEBR15121-E　{0} と {1} の型が異なります。{2} のロール ={3} アイテム 1：{4}, 個所 ={5}, 型名 ={6} アイテム 2：{7}, 個所 ={8}, 型名 ={9} リンク：{10}, 個所 ={11}

KEBR15122-E　{0} の型が数値型でありません。{1} のロール ={2} アイテム :{3}, 個所 ={4}, 型名 ={5} リンク：{6}, 個所 ={7}

KEBR15201-E　未知の単位 {0} を記述しました。{1} 個所 ={2}

KEBR15211-E　インスタンス文書の {0} 要素に {1} 属性は指定できません。{2} 個所 ={3}

KEBR15221-E　{0} に対応する {1} が存在しません。{2} のロール ={3} {4} 個所 ={5} リンク：{6}, 個所 ={7}

KEBR15231-E　アイテム {0} とアイテム {1} が参照するユニットが s-equal でありません。{2} {3} のロール ={4} {5} アイテム 1: 個所 ={6},{7} アイテム 2: 個所 ={8},{9}

KEBR15232-E　アイテム {0} とアイテム {1} の値が v-equal でありません。{2} {3} のロール ={4} {5} アイテム 1: 個所 ={6}, 値 ={7} アイテム 2: 個所 ={8}, 値 ={9}

KEBR15233-W　アイテム {0} と p-equal，c-equal のアイテムを無視しました。{1} {2} のロール ={3} {4} 個所 ={5}

KEBR15234-E　推測済みのアイテム {0} の内容が別のアークロールおよびロールで再推測されたアイテムの内容と異なります。{1} 推測したアイテム：{2} のロール ={3}, アークロール ={4}, 値 ={5} 再推測したアイテム：{6} のロール ={7}, アークロール ={8}, 値 ={9}

KEBR15241-E　アイテム {0} の値が検証値と等しくありません。値 ={1} 検証値 ={2} {3} {4} のロール ={5} {6} 個所 ={7}

KEBR15242-W　アイテム {0} と duplicate のアイテムがありました。{1} {2} のロール ={3} {4} 個所 ={5}

KEBR15243-W　アイテム {0} の検証を行いませんでした。{1} {2} のロール ={3} {4} 個所 ={5}

KEBR15251-W　アイテム {0} の精度が 0 のためアイテム {1} の検証値の計算に入れません。{2} {3} のロール ={4} {5} アイテム：値 ={6}, 個所 ={7} 合計値アイテム：個所 ={8}

KEBR15252-W　アイテム {0} の値は {1} と解釈します。{2} 個所 ={3}

KEBR16000-E　ハイパーキューブ要素宣言が抽象要素ではありません。{0}, 個所 ={1}

KEBR16001-E　ディメンジョン要素宣言が抽象要素ではありません。{0}, 個所 ={1}

KEBR16002-E　all アークのソースがアイテムではありません。{0}, 個所 ={1}, 参照先 ={2}

KEBR16003-E　notAll アークのソースがアイテムではありません。{0}, 個所 ={1}, 参照先 ={2}

KEBR16004-E　all アークのターゲットがハイパーキューブではありません。{0}, 個所 ={1}, 参照先 ={2}

KEBR16005-E　notAll アークのターゲットがハイパーキューブではありません。{0}, 個所 ={1}, 参照先 ={2}

KEBR16006-E　hypercube-dimension アークのソースがハイパーキューブではありません。{0}, 個所 ={1}, 参照先 ={2}

KEBR16007-E　hypercube-dimension アークのターゲットがディメンジョンではありません。{0}, 個所 ={1}, 参照先 ={2}

KEBR16008-E　dimension-domain アークのソースが Explicit Dimension ではありません。{0}, 個所 ={1}, 参照先 ={2}

KEBR16009-E　dimension-domainアークのターゲットがアイテムではありません。{0}, 個所 ={1}, 参照先 ={2}

KEBR16010-E　domain-member アークのソースがアイテムではありません。{0}, 個所 ={1}, 参照先 ={2}

KEBR16011-E　domain-member アークのターゲットがアイテムではありません。{0}, 個所 ={1}, 参照先 ={2}

KEBR16012-E　dimension-default アークのソースが Explicit Dimension ではありません。{0}, 個所 ={1}, 参照先 ={2}

KEBR16013-E　dimension-default アークのターゲットがアイテムではありません。{0}, 個所 ={1}, 参照先 ={2}

KEBR16014-E　all アークに xbrldt:contextElement 属性がありません。{0}, 個所 ={1}

KEBR16015-E　notAll アークに xbrldt:contextElement 属性がありません。{0}, 個所 ={1}

KEBR16016-E　属性の値が解決できません。属性 {0}={1}, {2}, 個所 ={3}

KEBR16017-E　属性の値が DTS に含まれていません。属性 {0}={1}, {2}, 個所 ={3}

KEBR16018-E　xbrldt:typedDomainRef 属性がディメンジョン要素宣言以外の場所に指定されています。{0}, 個所 ={1}

KEBR16019-E　xbrldt:typedDomainRef 属性の値が不正です。値 ={0}, {1}, 個所 ={2}

KEBR16020-E　xbrldt:typedDomainRef 属性の URI が不正です。値 ={0}, {1}, 個所 ={2}

KEBR16021-E　経路が循環しています。cyclesAllowed={0}, アークロール ={1}, {2} のロール ={3}, 経路 ={4}

KEBR16022-E　不正な継承関係です。cyclesAllowed={0}, {1} のロール ={2}, 経路 ={3}

KEBR16023-E　デフォルト値が複数あります。{0}, 個所 ={1}

KEBR16200-E　デフォルト値がインスタンス文書に指定されています。デフォルト値 ={0}, {1}, 個所 ={2}

KEBR16201-E　all のハイパーキューブが不正です。アイテム ={0}, ロール ={1}, ハイパーキューブ ={2}, 不正なディメンジョン ={3}, {4}, 個所 ={5}

KEBR16202-E　notAll のハイパーキューブが妥当です。アイテム ={0}, ロール ={1}, ハイパーキューブ ={2}, 妥当なディメンジョン ={3}, {4}, 個所 ={5}

KEBR16203-E　{0} 要素は，リンクベースの定義とインスタンス文書の指定が一致しません。アイテム ={1}, ロール ={2}, {3}, 個所 ={4}

KEBR16204-E　使用できないメンバがインスタンス文書に指定されています。アイテム ={0}, ロール ={1}, ハイパーキューブ ={2}, メンバ ={3}, {4}, 個所 ={5}

KEBR16205-E　ハイパーキューブに定義されていないディメンジョンがインスタンス文書に指定されています。アイテム ={0}, ロール ={1}, ハイパーキューブ ={2}, {3}, 個所 ={4}

KEBR16206-E　DRS に含まれないメンバがインスタンス文書に指定されています。アイテム ={0}, ロール ={1}, ハイパーキューブ ={2}, ディメンジョン ={3}, メンバ ={4}, {5}, 個所 ={6}

KEBR16207-E　一つの context 要素内に同じディメンジョンが複数指定されています。コンテキスト ={0}, ディメンジョン ={1}, {2}, 個所 ={3}

KEBR16208-E　xbrldi:typedMember 要素の dimension 属性の値が Typed Dimension ではありません。コンテキスト ={0}, ディメンジョン ={1}, {2}, 個所 ={3}

KEBR16209-E　xbrldi:explicitMember 要素の dimension 属性が Explicit Dimension ではありません。コンテキスト ={0}, ディメンジョン ={1}, {2}, 個所 ={3}

KEBR16210-E　xbrldi:explicitMember 要素に指定されている QName が DTS にありません。コンテキスト ={0}, ディメンジョン ={1}, メンバ ={2}, {3}, 個所 ={4}

KEBR16211-W　指定されたロールが定義されていません。ロール ={0}, {1}

KEBR16212-I　アイテムの一つのロールについて妥当性を検証しました。アイテム ={0}, ロール ={1}, 結果 ={2}, {3}, 個所 ={4}

KEBR19999-E　内部エラーが発生しました。詳細情報 ={0}

KEBR20000-E　XBRL DOM はすでに初期化されています。

KEBR20001-E　XBRL DOM は初期化されていません。

KEBR20002-E　引数 {0} に null は指定できません。

KEBR20004-E　引数 {0} に指定された {1} オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この {2} オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なります。

KEBR20005-E　引数 {0} に指定された {1} オブジェクトは，すでに他の {2} オブジェクトに設定されています。

KEBR20008-E　このラベルはラベルリンクの外で定義されています。よって，このメソッドは使用できません。

KEBR20009-E　このラベルはラベルリンク内の link:label 要素で定義されています。よって，このメソッドは使用できません。

KEBR20010-E　この文献への参照はリファレンスリンクの外で定義されています。よって，このメソッドは使用できません。

KEBR20011-E　この文献への参照はリファレンスリンク内の link:reference 要素で定義されています。よって，このメソッドは使用できません。

KEBR20012-E　引数 {0} に指定された配列のインデックス {1} の要素に null が指定されました。null は指定できません。

KEBR20013-E　引数 {0} に指定された配列は，長さが 0 の配列で要素を含みません。

KEBR20014-E　引数 {0} に不正な値が指定されました。(指定された値 ={1}, 指定できる値 ={2})

KEBR20016-E　引数 {0} に指定された配列のインデックス {1} の要素にシステム識別子が設定されていない org.xml.sax.InputSource オブジェクトが設定されています。

KEBR20017-E　XBRL 文書の読み込みに失敗しました。{0}

KEBR20018-E　XBRL 文書の解析に失敗しました。{0}

KEBR20019-E　XBRL DOM 内部で予期しないエラーが検出されました。(場所 ={0}, 詳細情報 1 ={1}, 詳細情報 2 ={2}, 詳細情報 3 ={3}, 詳細情報 4 ={4}, 詳細情報 5 ={5}, 詳細情報 6 ={6}, 詳細情報 7 ={7})

KEBR20020-E　このメソッドは {0} ノード型以外では使用できません。

KEBR20021-E　引数 {0} に指定された {1} オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この XBRLInstanceDocument オブジェクトではありません。

KEBR20022-E　名前空間接頭辞が宣言されていません。(名前空間接頭辞 ={0})

KEBR20023-E　名前空間 URI が宣言されていません。(名前空間 URI={0})

KEBR20026-E　XBRLInstanceItem オブジェクトに値が設定されていません。

KEBR20027-E　修飾名の形式が不正です。(修飾名 ={0})

KEBR20028-E　ID が重複しています。(ID={0})

KEBR20029-E　{0} 型のアイテムの文字列表現が不正です。(文字列表現 ={1})

KEBR20030-E　XBRLPeriod オブジェクトの内容が {0} の場合は，{1} メソッドを呼び出せません。

KEBR20031-E　リンクの遅延解析処理でエラーが発生しました。{0}

KEBR20032-E　コンテキスト要素がアイテムの型に適合していません。(コンテキスト要素 ={0}, アイテム ={1}, 名前空間 URI={2}, 要素名 ={3})

KEBR20033-E　アイテムの値として指定されたオブジェクトが，アイテムのデータ型に適合していません。(アイテムのデータ型 ={0}, 適合しないオブジェクト ={1})

KEBR20034-E　引数 {0} に指定された {1} オブジェクトを所有している XBRLTaxonomyDocument オブジェクトが，この {2} オブジェクトを所有している XBRLTaxonomyDocument オブジェクトと異なります。

KEBR20035-E　引数 {0} に指定された要素は，このインスタンス文書のルート要素の子要素ではありません。

KEBR20036-E　グローバル属性の接頭辞には null を設定できません。(名前空間 URI={0})

KEBR20037-E　名前空間接頭辞はすでに他の名前空間 URI で使用されています。(名前空間接頭辞 ={0}, 名前空間 URI={1})

KEBR20038-E　"xml" で始まる名前空間接頭辞は登録できません。

KEBR20039-E　名前空間 URI"http://www.w3.org/XML/1998/namespace" および "http://www.w3.org/2000/xmlns/" の名前空間宣言は登録できません。

KEBR20040-E　指定された XBRLNamespaceDeclaration オブジェクトは登録されていません。

KEBR20041-E　名前空間 URI"{0}" は XBRL 文書中で使用されているため，名前空間宣言を削除できません。

KEBR20042-E　指定された XBRLSchemaLocation オブジェクトは登録されていません。

KEBR20043-E　指定されたコンテキスト要素はインスタンス文書中に存在しません。

KEBR20044-E　指定されたアイテムは，タクソノミー文書で定義されていません。(名前空間 URI={0}, 要素名 ={1})

KEBR20045-E　指定されたタプルは，タクソノミー文書で定義されていません。(名前空間 URI={0}, 要素名 ={1})

KEBR20046-E　指定された脚注はインスタンス文書中に存在しません。

KEBR20047-E　指定された linkbaseRef 要素はインスタンス文書中に存在しません。

KEBR20048-E　このノードはインポートできません。(ノードの型 ={0})

KEBR20050-E　グローバル属性の名前空間接頭辞が指定されていません。(名前空間 URI={0}, 属性名 ={1})

KEBR20051-E　引数 {0} に指定された {1} オブジェクトを所有している XBRLDocumentManager オブジェクトが，この {2} オブジェクトを所有している XBRLDocumentManager オブジェクトと異なります。

KEBR20052-E　XBRL 文書の出力に失敗しました。{0}

KEBR20053-E　指定された脚注はこの要素と関連づけられていません。

KEBR20054-E　引数 {0} に指定されたノードは，このノードの先祖かこのノード自身です。

KEBR20055-E　引数 {0} に指定された要素は，このタプルの子要素ではありません。

KEBR20056-E　period 要素の子要素の組み合わせが不正です。(type={0}, instant={1}, startDate={2}, endDate={3}, duration={4})

KEBR20057-E　終了日は開始日よりもあとの日付でなければなりません。(startDate={0}, endDate={1})

KEBR20058-E　属性値が不正です。(属性名 ={0}, 属性値 ={1})

KEBR20061-E　アイテムの内容に子要素が含まれているため値を取得できません。

KEBR20062-E　引数 prefix が null 以外の場合，引数 uri に空文字列は指定できません。

KEBR20063-E　{0}.{1} は未サポートです。

KEBR20064-E　引数 {0} と {1} に同じオブジェクトを指定できません。

KEBR20065-E　指定された要素はタクソノミー文書で抽象的な要素として定義されているため，インスタンスを生成できません。(名前空間 URI={0}, 要素名 ={1})

KEBR20067-E　通し番号が最大値（2147483647）に達したため，名前空間接頭辞を生成できません。

KEBR20068-W　XBRL 文書の出力で警告が発生しました。詳細情報 ={0}

KEBR20069-E　デフォルトの名前空間が宣言されているため，名前空間に属さない XBRLQName オブジェクトの修飾名を生成できません。(ローカルパート ={0}, デフォルトの名前空間 ={1})

KEBR20070-E　名前空間に属さない要素または修飾名が存在するため，デフォルトの名前空間を宣言できません。

KEBR20071-E　継続期間は正の値でなければなりません。(指定された値 ={0})

KEBR20072-E　XBRL 文書の出力でエラーが発生しました。詳細情報 ={0}

KEBR20073-E　XBRL 文書の出力で致命的なエラーが発生しました。詳細情報 ={0}

KEBR20100-E　操作対象の文書が準拠している XBRL のバージョンが {0} の場合は，{1}.{2} メソッドは利用できません。

KEBR20101-E　バージョン {0} の XBRL 仕様に準拠した文書を処理対象とするように設定されている場合は，{1}.{2} メソッドは利用できません。

KEBR20102-E　指定された XBRLTaxonomyDocumentBuilder オブジェクトが処理対象とする XBRL のバージョンと，この XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL のバージョンが異なります。(XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL のバージョン ={0}, XBRLTaxonomyDocumentBuilder オブジェクトが処理対象とする XBRL のバージョン ={1})

KEBR20103-E　指定された {0} オブジェクトが準拠する XBRL のバージョンと，この {2} オブジェクトが処理対象とする XBRL のバージョンが異なります。({2} オブジェクトが処理対象とする XBRL のバージョン ={3}, {0} オブジェクトが準拠する XBRL のバージョン ={1})

KEBR20104-E　準拠している XBRL のバージョンが異なるため，ノードをインポートできません。(インポート元のバージョン ={0}, インポート先のバージョン ={1})

KEBR20105-E　属性を認識できません。(属性名 ={0})

KEBR20106-E　インスタンス文書のルート要素が DOM ツリーにつながっていないため出力できません。

KEBR20107-E　名前空間に属さない要素または修飾名が存在するため，デフォルトの名

前空間をリセットする宣言（xmlns=""）を削除できません。

KEBR20108-E　引数 {0} に指定された文字列が {1} 型ではありません。(指定された文字列 ={2})

KEBR20109-E　引数 {0} に指定された URI が絶対 URI ではありません。(指定された URI={1})

KEBR20110-E　引数 {0} には空の URI 参照は指定できません。

KEBR20111-E　カスタムロール {0} の定義を参照する link:roleRef 要素がすでに存在します。

KEBR20112-E　カスタムアークロール {0} の定義を参照する link:arcroleRef 要素がすでに存在します。

KEBR20113-E　link:roleRef 要素の roleURI 属性には XBRL 仕様で規定された標準のロールを指定できません。(指定された標準のロール ={0})

KEBR20114-E　link:arcroleRef 要素の arcroleURI 属性には XBRL 仕様で規定された標準のアークロールを指定できません。(指定された標準のアークロール ={0})

KEBR20115-E　XBRL 仕様で規定された標準のロール {0} は {1} 要素で使用できません。

KEBR20116-E　XBRL 仕様で規定された標準のアークロール {0} は {1} 要素で使用できません。

KEBR20117-E　カスタムロール {0} の定義を参照する link:roleRef 要素が見つかりません。

KEBR20118-E　カスタムアークロール {0} の定義を参照する link:arcroleRef 要素が見つかりません。

KEBR20119-E　カスタムロール {0} は {1} 要素で使用できるように定義されていません。

KEBR20120-E　カスタムアークロール {0} は {1} 要素で使用できるように定義されていません。

KEBR20121-E　XBRLRoleRef オブジェクトはアクティブではないため，XBRLRoleType オブジェクトを取得できません。

KEBR20122-E　XBRLArcroleRef オブジェクトはアクティブではないため，XBRLArcroleType オブジェクトを取得できません。

KEBR20123-E　指定された脚注は，指定された link:footnoteArc 要素によってこの要素と関連づけられていません。(脚注リンクのロール ={0}, アークロール ={1}, order 属性 ={2})

KEBR20124-E　指定された単位要素はインスタンス文書中に存在しません。

KEBR20125-E　基本単位のローカルパートが不正です。(名前空間 URI={0}, ローカルパート ={1})

KEBR20126-E　単位要素の分子と分母に同じ内容の基本単位が存在します。(名前空間 URI={0}, ローカルパート ={1})

KEBR20127-E　単位要素の内容が，アイテムのベースデータ型と矛盾しています。(名前空間 URI={0}, 要素名 ={1}, ベースデータ型 ={2})

KEBR20128-E　XBRLUnit2 オブジェクトの内容が {0} の場合は，{1} メソッドを呼び出せません。

KEBR20129-E　要素宣言において nillable 属性に true が指定されていない要素には xsi:nil 属性を設定できません。(名前空間 URI={0}, 要素名 ={1})

KEBR20130-E　xsi:nil 属性に true が設定されている要素に対して {0}.{1} メソッドを呼び出せません。(名前空間 URI={2}, 要素名 ={3})

KEBR20131-E　XBRLInstanceDocument.createNumericItem メソッドでは非数値アイテムを生成できません。(名前空間 URI={0}, 要素名 ={1})

KEBR20132-E　XBRLInstanceDocument.createNonNumericItem メソッドでは数値アイテムを生成できません。(名前空間 URI={0}, 要素名 ={1})

KEBR20133-E　ベースデータ型が FRACTION_ITEM であるアイテムには，precision 属性および decimals 属性は指定できません。(名前空間 URI={0}, 要素名 ={1}, precision={2}, decimals={3})

KEBR20134-E　precision 属性と decimals 属性の両方を省略できません。(名前空間 URI={0}, 要素名 ={1})

KEBR20135-E　precision 属性と decimals 属性の両方を同時に指定できません。(名前空間 URI={0}, 要素名 ={1}, precision={2}, decimals={3})

KEBR20136-E　このアイテムに対して {0} 属性を設定・取得できません。(名前空間 URI={1}, 要素名 ={2}, ベースデータ型 ={3}, 数値 ={4})

KEBR20137-E　非数値アイテムに対する単位要素の設定・取得はできません。(名前空間 URI={0}, 要素名 ={1})

KEBR20138-E　指定された文字列から XBRLFraction オブジェクトを生成できませんでした。(名前空間 URI={0}, 要素名 ={1}, 指定された文字列 ={2})

KEBR20139-E　ベースデータ型が {0} であるアイテムに対して，文字列表現による値の設定・取得はできません。(名前空間 URI={1}, 要素名 ={2})

KEBR20140-E　時期要素の内容が，アイテムの要素宣言の xbrli:periodType 属性における指定と矛盾しています。(名前空間 URI={0}, 要素名 ={1}, xbrli:periodType 属性 ={2},

時期要素の内容 ={3})

KEBR20141-E　時期要素の内容の意味は変更できません。(既存の時期要素の内容 ={0}, 新たに設定しようとした時期要素の内容 ={1})

KEBR20142-E　操作対象の文書が準拠している XBRL のバージョンが {0} の場合は, XBRLPeriod.STARTDATE_DURATION および XBRLPeriod.DURATION_ENDDATE は使用できません。

KEBR20143-E　XBRLIdManager オブジェクトに登録しようとしたインスタンス文書は，すでに管理対象となっているインスタンス文書とは異なる XML 文書に記述されています。

KEBR20144-E　バージョン {0} の XBRL 仕様に準拠した文書を XBRLIdManager オブジェクトへ登録できません。

KEBR20145-E　要素宣言における定義内容が異なるアイテムはインポートできません。(名前空間 URI={0}, 要素名 ={1}, ベースデータ型 ={2}/{3}, xbrli:periodType 属性 ={4}/{5})

KEBR20146-E　この要素に脚注が関連づけられているため，id 属性を削除できません。(名前空間 URI={0}, 要素名 ={1}, id={2})

KEBR20147-E　引数 {0} に指定されたシステム識別子の形式が不正です。(指定されたシステム識別子 ={1})

KEBR20148-E　引数 {0} に指定された配列のインデックス {1} の要素に設定されているシステム識別子の形式が不正です。(指定されたシステム識別子 ={2})

KEBR20149-E　指定された XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書の解析時に完全な妥当性検証を行うように設定されていません。

KEBR20150-E　指定されたプロパティを認識できません。(プロパティ名 ={0})

KEBR20151-E　{0} プロパティの値として null は指定できません。

KEBR20152-E　{0} プロパティの値として指定されたオブジェクトの型が不正です。(不正な型 ={1}, 正しい型 ={2})

KEBR20153-E　{0} プロパティに不正な値が指定されました。(指定された値 ={1}, 指定できる値 ={2})

KEBR20154-E　{0} プロパティの値として指定された配列の要素数が 0 です。要素数は 1 以上でなければなりません。

KEBR20155-E　{0} プロパティの値として指定された配列のインデックス {1} の要素が null です。配列の要素に null は指定できません。

KEBR20156-E　引数 {0} に不正な値が指定されました。(指定された値 ={1})

KEBR22001-E　引数 {0} に {1} オブジェクトは指定できません。

KEBR22002-E　{0} オブジェクトの名前はすでに使用されています。(名前 ={1})

KEBR22003-E　{0} オブジェクトの ID はすでに使用されています。(ID={1})

KEBR22004-E　この {0} オブジェクトは関係を否定するオブジェクトです。よって，このメソッドは使用できません。

KEBR22006-E　引数 {0} に指定された {1} オブジェクトを所有している SimpleTaxonomy オブジェクトが，この {2} オブジェクトを所有している SimpleTaxonomy オブジェクトと異なります。

KEBR22007-E　引数 {0} に指定された {1} オブジェクトを所有している XBRLTaxonomyDocument オブジェクトが，この SimpleTaxonomy オブジェクトを生成したときに指定された XBRLTaxonomyDocument オブジェクトと異なります。

KEBR22008-E　引数 {0} に指定された {1} オブジェクトを所有している SimpleTaxonomy オブジェクトが，この SimpleTaxonomy オブジェクトと異なります。

KEBR22009-E　この SimpleItemDefinition オブジェクトの type 属性が {0} です。よって，このメソッドは使用できません。

KEBR22010-E　拡張元として指定されたタクソノミー文書の基点となるタクソノミースキーマに，他の XML 文書の一部となっているタクソノミースキーマが存在します。(絶対 URI={0})

KEBR22011-E　cbrp.parse.validate.frtafris オプションが false の場合は，{0}.{1} メソッドは利用できません。

KEBR22100-E　操作対象の文書が準拠している XBRL Dimensions のバージョンが {0} の場合は，{1}.{2} メソッドは利用できません。

KEBR22101-E　指定された XBRLTaxonomyDocumentBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョンと，この XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョンが異なります。(XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョン ={0}, XBRLTaxonomyDocumentBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョン ={1})

KEBR22102-E　引数 {0} に指定されたロールは DRS にありません。(ロール ={1})

KEBR22103-E　指定された {0} オブジェクトが準拠する XBRL Dimensions のバージョンと，この {2} オブジェクトが処理対象とする XBRL Dimensions のバージョンが異なります。({2} オブジェクトが処理対象とする XBRL Dimensions のバージョン ={3}, {0} オブジェクトが準拠する XBRL Dimensions のバージョン ={1})

KEBR22104-E　引数 {0} に指定された要素は，このシナリオ要素の子要素ではありません。

KEBR22105-E　引数 {0} に指定された要素は，このセグメント要素の子要素ではありません。

KEBR22106-E　指定されたディメンジョンは，タクソノミー文書で定義されていません。(名前空間 URI={0}, 要素名 ={1})

KEBR25000-E　指定された {0} 型データの文字列表現が不正です。

KEBR25001-E　引数 {0} に負数は指定できません。指定できる値 = 負数以外

KEBR25002-E　保持しているバイト配列の長さ以上の引数 {0} は指定できません。指定できる値 = 保持しているバイト配列の長さ未満

KEBR25003-E　配列 {0} の長さを超える引数 {1}+{2} は指定できません。指定できる値 = 配列 {0} の長さ以下

KEBR25004-E　引数 {0} に {1} の倍数の長さ以外の文字列は指定できません。指定できる値 ={1} の倍数の長さの文字列 {0}

KEBR25005-E　ISO8859-1 の文字セットがサポートされていません。

KEBR25006-E　フォーマットが不正です。

KEBR25007-E　引数 {0} に長さ 0 の引数は指定できません。指定できる値 = 長さ 0 以外の引数 {0}

KEBR25008-E　指定された名前空間 URI に不正な文字列が含まれています。名前空間 URI={0}

KEBR25009-E　指定されたローカルパートに不正な文字列が含まれています。ローカルパート ={0}

KEBR25010-E　引数 {0} に 0 または，負数は指定できません。指定できる値 =0 より大きい値

KEBR25011-E　引数 {0} に 0 は指定できません。指定できる値 =0 以外の値

KEBR25013-E　指定された名前空間接頭辞に不正な文字列が含まれています。名前空間接頭辞 ={0}

KEBR25014-E　指定されたスキーマ文書の位置を示す URI に不正な文字列が含まれています。スキーマ文書の位置を示す URI={0}

KEBR30000-I　XBRLDocumentManager を初期化しました。

KEBR39000-E　An attempt to initialize XBRLDocumentManager has failed.

KEBR39001-E　XBRLDocumentManager is not initialized.

KEBR39002-E　XBRLDocumentManager has already been initialized.

KEBR39100-E　An attempt to initialize a message has failed.

KEBR39101-E　The message is not initialized.

KEBR39102-E　The message has already been initialized.

KEBR39103-E　An attempt to read the cbrp_message.properties has failed.

KEBR39200-E　Property initialization has failed. ({0}:({1}))

KEBR39201-E　The property is not initialized. ({0}.{1})

KEBR39202-E　The property has already been initialized. ({0}.{1})

KEBR39203-E　An attempt to read the defaults.properties file has failed. ({0}):({1})

KEBR39204-E　The specified file was not found. (file name = {0}):({1})

KEBR39205-E　A file permission error occurred. ({0}: file name = {1}):({2})

KEBR39206-E　The specified value is outside the valid range. ({0} = {1}:{2})

KEBR39207-E　The specified parameter is invalid. ({0} = {1})

KEBR39300-E　An attempt to initialize a trace has failed.

KEBR39301-E　The trace is not initialized.

KEBR39302-E　The trace has already been initialized.

KEXL00001-E　XLinkDocumentSet のクローンの作成に失敗しました。

KEXL00002-E　href 属性に指定した URI に接続できません。

KEXL00003-E　同一拡張リンク中に同じ弧を持つ arc 型要素が指定されています。

KEXL00004-E　href 属性に指定したファイルが見つかりません。

KEXL00005-E　actuate 属性の値に，{onLoad | onRequest | other | none} 以外の値が指定されています。

KEXL00006-E　show 属性の値に，{new | replace | embed | other | none} 以外の値が指定されています。

KEXL00007-E　type 属性の値に，
{simple | extended | locator | arc | resource | title | none} 以外の値が指定されています。

KEXL00008-E　{label | from | to} 属性の値が NCName ではありません。

KEXL00009-E　必須項目が指定されていません。locator 型要素に href 属性は必須です。

KEXL00010-E　指定した URI が RFC2396 に違反しています。

KEXL00011-E　{role|arcrole} 属性の値は絶対 URI でないといけません。

KEXL00012-E　{from|to} に指定した要素がみつかりません。

KEXL00013-E　href 属性に指定した XPointer 式が評価できません。

KEXL00014-E　href 属性に指定した XPointer 式が不正です。

KEXL10001-E　XPointerLocation.getStartPoint メソッドは，ノード型が属性ノードまたは名前空間ノードであるノードロケーションには適用できません。

KEXL10002-E　XPointerLocation.getEndPoint メソッドは，ノード型が属性ノードまたは名前空間ノードであるノードロケーションには適用できません。

KEXL10003-E　XPointerLocation.rangeTo メソッドまたは range-to 関数で生成するレンジの開始，終了ポイントが，同一の XML 文書の中にありません。

KEXL10004-E　XPointerManager.evaluate メソッドに指定された org.w3c.dom.Document が不正です。

KEXL10005-E　メソッドのパラメタが不正です。

KEXL10006-E　XPointer 式の構文エラーです。

KEXL10007-E　XPointerManager.evaluate メソッドに指定された org.w3c.dom.Document が here ノード，origin ノードのオーナードキュメントと異なります。

KEXL10008-E　XPointer 式に here 関数がありますが here ノードが指定されていません。

KEXL10009-E　XPointer 式に origin 関数がありますが origin ノードが指定されていません。

KEXL10010-E　"" が閉じられていません。

KEXL10011-E　"" が閉じられていません。

KEXL10012-E　括弧の対応が取れていません。

KEXL10013-E　サーカムフレックスの使用方法に誤りがあります。

KEXL10014-E　{0} 関数に指定できる引数は {1} 個だけです。

KEXL10015-E　xpointer() スキーマ式の中に何も指定されていません。

KEXL10016-E　{0} は誤った数値の書き方です。

KEXL10017-E　{0} は正しい関数名ではありません。

KEXL10018-E　{0} が必要です。{1} は誤りです。

KEXL10019-E　位置パスが必要です。{0} は誤りです。

KEXL10020-E　'/' または '//' の次に位置ステップが必要です。{0}

KEXL10021-E　ノードテスト NCName:* または QName のどちらかが必要です。

KEXL10022-E　'/' または '//' の次に相対位置ステップが必要です。{0}

KEXL10023-E　xpointer() スキーマ式の終わりに余分な文字があります。{0}

KEXL10024-E　',' が見つかりましたが後続の引数がありません。

KEXL10025-E　{0} は正しい軸の名前ではありません。

KEXL10026-E　'.[述部]' は正しくない書き方です。'self::node()[述部]' を使用してください。

KEXL10027-E　{0} は正しいノード型の名前ではありません。

KEXL10028-E　名前空間接頭辞を名前空間名に解決できません。{0}

KEXL10029-E　xpointer() スキーマ式の中に変数を記述することはできません。

KEXL10030-E　XPointerLocation.rangeTo メソッドまたは range-to 関数で生成したレンジの一方のポイントのコンテナノードが要素型，テキスト型，ルート型以外のとき，もう一方のポイントも同一のコンテナノードを持たなければなりません。

KEXL10031-E　XPointerLocation.rangeTo メソッドまたは range-to 関数で生成したレンジの開始ポイントが，終了ポイントよりも文書順で後ろに存ってはなりません。

KEXL10034-E　属性ノードまたは名前空間ノードの開始ポイントを取得することはできません。

KEXL10035-E　属性ノードまたは名前空間ノードの終了ポイントを取得することはできません。

KEXL10036-E　xpointer() スキーマ式の評価結果がロケーションセット型ではありません。

KEXL10037-E　{0} 関数の引数の型はロケーションセット型でなければなりません。

KEXL10038-E　string-range 関数の第 1 引数の型はロケーションセット型でなければなりません。

KEXL10039-E　XPointerLocation.compareByDocOrder メソッドで，異なる XML 文書

のロケーションを比較することはできません。

KEXL10040-E　XPointerLocation.compareByDocOrder メソッドで，同一要素に属する異なる属性ノードまたは名前空間ノードをコンテナノードとする，二つのポイントを比較することはできません。

KEXL10041-E　XPointerLocation.compareByDocOrder メソッドで，属性ノードまたは名前空間ノードと，そのノードをコンテナノードとするポイントを比較することはできません。

KEXL10042-E　比較できないロケーションが同一ロケーションセットに追加されました。

KEXL10043-E　',' が見つかりましたが先行する引数がありません。

KEXL10044-E　{0} が必要です。

KEXL10045-E　文字列，論理値，数値に評価される式をロケーションセットとして使用することはできません。

KEXL19998-E　内部エラーです。

7.3 メッセージ KEBR10000-KEBR19999

KEBR10000-KEBR19999 のメッセージの詳細について番号順に説明します。

KEBR10000-E （F，V）

{0} 要素の {1} 属性に指定できない値が指定されました。{2} 個所 ={3} 指定できる値 ={4} 指定された値 ={5}

{0}：エラーが起きた要素名

{1}：エラーが起きた属性名

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた要素の XML 文書内での位置

{4}：指定できる値

{5}：指定されていた値

要因：

指定できない値が指定されました。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

指定できる値に変更して，再度実行してください。

KEBR10001-E （F，V）

{0} 要素の {1} 属性に推奨値以外の値が指定されました。{2} 個所 ={3} 指定された値 ={4}

{0}：エラーが起きた要素名

{1}：エラーが起きた属性名

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた要素の XML 文書内での位置

{4}：指定されていた値

要因：

推奨値以外の値が指定されました。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

推奨値を指定して，再度実行してください。

KEBR10002-E （F，V）

{0} 要素に {1} 要素が存在しません。{2} 個所 ={3}

{0}：エラーが起きた要素名
{1}：必須要素。要素が複数ある場合は，コンマでセパレートされます。
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：
必須要素がありません。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：
必須要素を追加して再度実行してください。

KEBR10003-E （F，V）

{0} 要素に {1} 属性が存在しません。{2} 個所 ={3}

{0}：エラーが起きた要素名
{1}：必須属性
{2}：エラーが起きた XML 文書の URI
{3}：エラーが起きた要素の XML 文書内での位置

要因：
必須属性がありません。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，fatalError() メソッドまたは，error() メソッドを呼び出します。fatalError() メソッドが呼ばれた場合は，処理を中止します。error() メソッドが呼ばれた場合，このメソッドで例外が上がったときは，そこで処理を中止します。例外が上がらないときは，その要素を無視して処理を続行します。

対策：
必須属性を追加して再度実行してください。

KEBR10004-W （F，V）

{0} 要素に不要な {1} 要素が指定されています。{2} 個所 ={3}

{0}：エラーが起きた要素名
{1}：不要な要素
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：
不要な要素が指定されています。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合

は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。
処理を続行する場合，あとから編集などで利用する可能性のある要素（コンテキスト，ユニット，リソース型要素など）については，XBRL DOM を生成します。それ以外の要素については無視します。

対策：
不要な要素を削除してください。

XBRL2.1 の場合
エラーが起きた要素がコンテキスト要素やユニット要素の場合，アイテムと関連づけられていません。必要に応じて関連づけてから，再度実行してください。

KEBR10005-W （F，V）

{0} 要素に不要な {1} 属性が指定されています。{2} 個所 ={3}

{0}：エラーが起きた要素名
{1}：不要な属性
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：
不要な属性が指定されています。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。
処理を続行する場合は，不要な属性を無視して処理を続行します。

対策：
不要な属性を削除してください。

KEBR10006-E （F，V）

{0} 要素の {1} 属性の参照先が存在しません。{2} 個所 ={3} 参照先 ={4}

{0}：エラーが起きた要素名
{1}：エラーが起きた属性名
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置
{4}：指定されていた値

要因：

指定された参照先がありません。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

必要に応じて参照先を正しく修正して，再度実行してください。

XBRL2.1 の場合

参照先の要素が，同じ <xbrli:xbrl> 要素の下位要素であるかどうかを確認し，必要に応じて参照先を追加するか，修正してください。

KEBR10007-E （F，V）

{0} 要素の {1} 属性の値が重複しています。{2} 個所 ={3} 指定された値 ={4}

{0}：エラーが起きた要素名

{1}：エラーが起きた属性名

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた要素の XML 文書内での位置

{4}：指定されていた値

要因：

一意でなければならない値が重複しています。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

重複がないように値を指定し直して，再度実行してください。

KEBR10009-E （F，V）

{0} 要素に使用できない値が指定されました。{1} 個所 ={2} 指定された値 ={3}

{0}：エラーが起きた要素名

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

{3}：指定されていた値

要因：

使用できない値が指定されました。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

duration 要素に 0 未満の値が指定されています。0 以上の値を指定して，再度実行してください。

KEBR10010-E （F，V）

period **要素の期間が範囲で指定されていません。**{0} **個所** ={1} startDate={2} endDate={3}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

{2}：startDate の値

{3}：endDate の値

要因：

startDate 要素（開始日付）の値と endDate 要素（終了日付）の値の関係が不正です。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

開始日時と終了日時を指定し直して，再度実行してください。

KEBR10012-W （F，V）

リンクが双方向で記述されていません。{0} **個所** ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

要因：

リンクが双方向で記述されていません。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。

XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

処理を続行する場合は，指定されたリンクを使用してツリーノードを作成します。

対策：

双方向となるようにリンクの記述を修正して，再度実行してください。

KEBR10013-E （F，V）

双方向で記述されたリンクの {0} **属性の値が異なります。**{1} **リンク１：個所** ={2}, **指定された値** ={3} **リンク２：個所** ={4}, **指定された値** ={5}

{0}：エラーが起きた属性名

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた階層（リンク１）

{3}：指定された値（リンク１）

{4}：エラーが起きた階層（リンク２）

{5}：指定された値（リンク２）

要因：

双方向で記述されたリンクの属性値が異なります。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

双方向で記述しているリンクの属性の値を合わせて，再度実行してください。

KEBR10014-W （F，V）

{0} 対象のリンクが存在しません。{1} 個所 ={2}

{0}：use 属性の値

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

要因：

上書きする対象のリンクがありません。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。
処理を続行する場合は，このリンクの記述を無視します。

対策：

上書き対象のリンクがあるかどうかを確認してください。

KEBR10016-E （F，V）

指定した xlink:role 属性の値と実際のリンク先の内容が一致していません。{0} 個所 ={1} xlink:role 属性の値 ={2} リンク先の内容 ={3}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

{2}：xlink:role 属性の値

{3}：リンク先の内容。リンク先の要素名を表示します。例えば，calculationLink などのリンクベース内の要素名を表示します。XBRL2.0 の場合は，リンク先が解析済みのタクソノミー本体の URI のとき，taxonomy が表示されます。

要因：

指定した xlink:role 属性の値と実際のリンク先の内容が一致していません。

XBRL2.0 の場合

処理を中止します。もし，XBRLParseErrorHandler の実装を指定して XBRL

文書の解析を行っていた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

XBRL2.1 の場合

XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は，xlink:href 属性で示す link:linkbase 要素以下の拡張リンクを DTS に含めます。

対策：

xlink:role 属性の値とリンク先の内容を一致させてから，再度実行してください。

KEBR10017-I （F）

group 要素の子要素で記述されている group 要素を無視しました。{0} 個所 ={1}

{0}：XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置

要因：

group 要素の子要素で記述されている group 要素を無視しました。処理を続行します。

対策：

なし

KEBR10018-E （F，V）

{0} 要素の {1} 属性に指定されたラベル名に対応する要素が見つかりません。{2} 個所 ={3}

{0}：エラーが起きた要素名
{1}：エラーが起きた属性名
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：

アーク型要素の xlink:from,xlink:to 属性に記述されたラベル名に対応するロケータ型要素またはリソース型要素が見つかりません。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

アーク型要素の種類によって，xlink:from 属性，xlink:to 属性が参照する要素が異

なります。そのため，xlink:from,xlink:to 属性のラベル名，またはロケータ型要素 / リソース型要素で記述しているラベル名を適切に修正したあと，再度実行してください。

XBRL2.0 の場合

関係リンクに記述する xlink:from 属性，xlink:to 属性が参照できる要素は，ロケータ型要素だけです。

ラベルリンクまたはリファレンスリンクに記述する，xlink:from 属性および xlink:to 属性が参照できる要素は，ロケータ型要素またはリソース型要素です。

脚注リンクに記述する xlink:from 属性はロケータ型要素を，xlink:to 属性はリソース型要素を参照できます。

XBRL2.1 の場合

関係リンクに記述する xlink:from 属性，xlink:to 属性が参照できる要素は，ロケータ型要素だけです。

ラベルリンクまたはリファレンスリンクに記述する xlink:from 属性はロケータ型要素を，xlink:to 属性はロケータ型要素またはリソース型要素を参照できます。

脚注リンクに記述する xlink:from 属性はロケータ型要素を，xlink:to 属性はリソース型要素を参照できます。

KEBR10019-E （F，V）

{0} **要素の** {1} **属性に指定されたラベル名の要素が** {2} **ではありません。** {3} **個所** ={4}

{0}：エラーが起きた要素名

{1}：エラーが起きた属性名

{2}：正しい要素

{3}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{4}：エラーが起きた要素の XML 文書内での位置

要因：

使用できない要素が指定されました。処理を中止します。

XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

xlink:arcrole 属性の値と xlink:from,xlink:to 属性に指定されたラベル名に対応する要素が合っていません。xlink:arcrole 属性と xlink:from,xlink:to 属性の関係および，xlink:from,xlink:to 属性の先の要素を見直し，正しく修正して，再度実行してください。

KEBR10020-W （F，V）

{0}, id={1} **のコンテキスト要素に対応するアイテム要素がありません。** {2} **個所** ={3}

{0}：エラーが起きたコンテキストの要素名
{1}：エラーが起きたコンテキストの id
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：

コンテキストに対応するアイテム要素がありません。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。
処理を続行する場合は，指定されたコンテキストの XBRL DOM を作成します。

対策：

必要に応じてコンテキストにアイテムを関連づけてから再度実行してください。

KEBR10021-E （F，V）

{0} 要素の {1} 属性に {2} 型以外の値が指定されました。{3} 個所 ={4}

{0}：エラーが起きた要素名
{1}：エラーが起きた属性名
{2}：正しいデータ型
{3}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{4}：エラーが起きた要素の XML 文書内での位置

要因：

属性に指定された値の型が正しいデータ型ではありません。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

指定した属性の値のデータ型を正しいデータ型に修正して，再度実行してください。

KEBR10024-E （F，V）

{0} 要素の priority={1} を持つ同一のリンクが複数存在します。要素 1：{2}, 個所 ={3} 要素 2：{4}, 個所 ={5}

{0}：エラーが起きた要素名
{1}：エラーが起きた要素の priority 値。priority 属性が省略された場合は，0 が仮定されます。
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた要素の XML 文書内での位置
{4}：エラーが起きた XML 文書の URI（比較した要素の URI）。URI=xx の形式で出力します。URI がない場合は，表示されません。
{5}：エラーが起きた要素（比較した要素）の XML 文書内での位置

要因：

アーク型要素に同じ priority 値を持つ同じリンクが複数あります。
priority 属性を省略した場合，デフォルトとして 0 が仮定されます。また，XBRL2.1 の場合，同じ priority 値を持つ同じリンクのアーク型要素の use 属性に "prohibited" が含まれていない場合に出力されます。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。
処理を続行する場合，関係リンクは，要素 2 で示すリンクを無視します。ラベルリンクは，要素 1 と要素 2 のリンク先のリソースで重なっているリソース型要素のうち，要素 2 で示すリソース型要素を無視します。

対策：

アーク型要素に同じ priority 値を持つリンクが重複しないように，個所 ={3} で示すアーク型要素の priority 属性値の値を適切に修正して，再度実行してください。

KEBR10025-E （F，V）

{0} 要素の {1} 属性に使用できない値が指定されました。{2} 個所 ={3} 指定された値 ={4}

{0}：エラーが起きた要素名
{1}：エラーが起きた属性名
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置
{4}：指定された値

要因：

使用できない値が指定されました。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

- presentationArc 要素の order 属性の場合（XBRL2.0 の場合）
  負の数は使用できません。正の数を指定して，再度実行してください。
- 属性が label,from,to のどれかの場合
  属性の値が NCName で記述されていません。NCName で記述して，再度実行してください。

- priority 属性の場合
  整数で指定されていません。整数に修正して，再度実行してください。
- group 要素（XBRL2.0 の場合）
  schemaLocation 属性で指定された名前空間と URI が組で指定されていません。
  組で指定して，再度実行してください。

KEBR10029-E （F，V）

同じ {0} **要素は複数存在できません。**{1} **個所** ={2}

{0}：エラーが起きた要素名
{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{2}：エラーが起きた要素の XML 文書内での位置

要因：
同じ言語種別かつ同じロール属性の値を持つリソース型要素が複数あります。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：
label 要素の場合は，一つのラベル名に対して，同じ言語種別（xml:lang 属性値）かつ同じロール（xlink:role 属性値）を持つ要素が複数記述されているため，ロールを分けて記述してください。

KEBR10030-E （F，V）

XML **文書中で指定されている** URI **が** RFC2396 **に違反しています。**{0} **個所** ={1} **指定されていた** URI={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：エラーが起きた URI の値

要因：
RFC2396 に違反しているため，符号化，絶対 URI 化に失敗しました。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：
URI の形式を正しく修正して，再度実行してください。

KEBR10031-E （F，V）

parse() **メソッドに指定された** URI **が** RFC2396 **に違反しています。指定されていた** URI={0}

{0}：エラーが起きた URI の値

要因：

RFC2396 に違反しているため，符号化，絶対 URI 化に失敗しました。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

parse() メソッドに指定されている uri を正しく入力してください。

KEBR10032-E （F，V）

同一拡張リンク中に同じ弧を持つアーク型要素が指定されています。{0} **個所** ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置

要因：

同一拡張リンク内に，同じ from と to 属性値を持つアーク型要素があります。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

同一拡張リンク中に同じ from と to 属性値を持つアーク型要素を削除，または修正して，再度実行してください。

KEBR10033-E （F，V）

ルート要素に指定できない要素が指定されました。{0} **指定できる要素** ={1} **指定された要素** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：指定できる要素
{2}：指定されていた要素

要因：

指定できない要素が指定されました。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

XBRL2.0 に準拠した文書の場合，次の内容を変更して，再度実行してください。

- インスタンス文書の場合は，group 要素に変更してください。
- タクソノミー本体の場合は，schema 要素に変更してください。
- リンクベースの場合は，linkbase 要素に変更してください。

KEBR10034-E （F，V）

{0} **要素の値が存在しません。**{1} **個所** ={2}

{0}：エラーが起きた要素名
{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{2}：エラーが起きた要素の XML 文書内での位置

要因：
要素に値が記述されていません。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：
要素に値を記述して，再度実行してください。

KEBR10035-E （F，V）

period **要素の子要素** {0} **と** {1} **の関係が正しくありません。**{2} **個所** ={3}

{0}：エラーが起きた子要素名 1
{1}：エラーが起きた子要素名 2
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：
period 要素の期間の指定方法が正しくありません。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：
period 要素の期間を正しく記述して，再度実行してください。

KEBR10036-E （F，V）

{0} **要素の値** {1} **が** {2} **型ではありません。**{3} **個所** ={4}

{0}：エラーが起きた要素名
{1}：指定された値
{2}：正しい型
{3}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{4}：エラーが起きた要素の XML 文書内での位置

要因：
要素の値に指定できないデータ型が指定されました。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

正しいデータ型の値を記述して，再度実行してください。

KEBR10037-E （F，V）

{0} **要素の** {1} **属性の値** {2} **が** {3} **型ではありません。** {4} **個所** ={5}

{0}：エラーが起きた要素名

{1}：エラーが起きた属性名

{2}：指定された値

{3}：正しい型

{4}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{5}：エラーが起きた要素の XML 文書内での位置

要因：

属性の値に指定できないデータ型が指定されました。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

正しいデータ型の値を記述して，再度実行してください。

KEBR10038-E （F，V）

{0} **要素はタクソノミー文書で定義されていません。** {1} **個所** ={2}

{0}：エラーが起きた要素名

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

要因：

指定された要素はタクソノミー文書で定義されていません。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。
処理を続行する場合は，その要素を無視します。

対策：

タクソノミー文書で定義している要素を確認し，要素名が誤りの場合はインスタンス文書を修正してください。要素の定義が必要であれば，タクソノミー文書に追記して，再度実行してください。

KEBR10039-E （F，V）

{0} **の名前空間が宣言されていません。** {1}

{0}：名前空間 URI
{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

要因：

必要な名前空間 URI が宣言されていません。必要な名前空間 URI については，次を参照してください。

- インスタンス文書に必要な名前空間 URI
  http://www.w3.org/2001/XMLSchema-instance
  http://www.xbrl.org/2001/instance
- タクソノミー文書に必要な名前空間 URI
  http://www.w3.org/2001/XMLSchema
- リンクベースに必要な名前空間 URI
  http://www.xbrl.org/2001/XLink/xbrllinkbase

処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

名前空間 URI の宣言をルート要素の属性に記述して，再度実行してください。

KEBR10040-E （F，V）

{0} **要素がパート要素の代替要素としてタクソノミー本体で定義されていません。** {1} **個所** ={2}

{0}：エラーが起きた要素名
{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{2}：エラーが起きた要素の XML 文書内での位置

要因：

reference 要素の子要素に指定されたパート要素が，タクソノミー本体で定義されていません。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。
処理を続行する場合は，XBRLPart オブジェクトを作成します。

対策：

パート要素の名前を確認し，誤りの場合は正しい名称にしてください。また，タクソノミー本体で定義する必要がある場合は代替要素をタクソノミー本体で宣言してください。

KEBR10041-E （F，V）

{0} **要素の** {1} **属性が絶対** URI **で記述されていません。** {2} **個所** ={3}

{0}：エラーが起きた要素名
{1}：エラーが起きた属性名
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：

要素で示す属性に指定する値は，絶対 URI で記述する必要があります。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

要素で示す属性の内容を絶対 URI に変更して，再度実行してください。

KEBR10042-W （F，V）

同じ内容の単純リンクが指定されました。要素 1：{0}, **個所** ={1} **要素 2：**{2}, **個所** ={3}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：エラーが起きた XML 文書の URI（比較した要素の URI）。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素（比較した要素）の XML 文書内での位置

要因：

タクソノミー本体の simple 型要素で指定されたリンクベースは解析済みです。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。

対策：

同一の単純リンクを指定した simple 型要素を削除して，再度実行してください。

KEBR10043-W （F，V）

タクソノミー本体を拡張する際に import **要素で記述した** URI **がループしています。**{0} **個所** ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた import 要素の XML 文書内での位置

要因：

タクソノミー本体の継承関係がループしています。import 要素に自分自身の URI を記述したか，拡張元のタクソノミー本体の URI に import 済みの URI（解析済み

の URI）が指定されています。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。

対策：
継承関係がループしているタクソノミー本体の import 要素を修正してください。

KEBR10044-E （F，V）

アイテムには子要素を記述できません。{0} **個所** ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置

要因：
アイテムには子要素を記述できません。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：
アイテムの子要素を削除して，再度実行してください。

KEBR10045-E （F，V）

{0} **要素はインスタンス文書では使用できません。**{1} **個所** ={2}

{0}：エラーが起きた要素名
{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{2}：エラーが起きた要素の XML 文書内での位置

要因：
タクソノミー本体で記述した要素のうち，インスタンス文書で使用できない要素を記述しています。
インスタンス文書で使用するには，タクソノミー本体で element 要素を記述する場合に substitutionGroup 属性が xbrli:item または xbrli:tuple である必要があります。また abstract 属性が false であるか，または abstract 属性を指定してはいけません。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：
要素のつづりを確認してください。もし誤りがなければ，次を確認してください。
要素をそのままインスタンス文書で使用する場合は，タクソノミー本体を修正して，再度実行してください。

- element 要素の substitutionGroup 属性が xbrli:item または xbrli:tuple であることを確認してください。
- element 要素の abstract 属性が false か，または abstract 属性が指定されていないことを確認してください。

KEBR10046-E （F，V）

{0} 要素の {1} 属性に標準値以外の値が指定されました。{2} 個所 ={3} 指定された値 ={4}

{0}：エラーが起きた要素名

{1}：エラーが起きた属性名

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた要素の XML 文書内での位置

{4}：指定されていた値

要因：

標準値以外の値が指定されました。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

標準値を記述したあと再度実行してください。

KEBR10047-E （F，V）

{0} 要素の {1} 属性に標準値またはカスタムされた値以外が指定されました。{2} 個所 ={3} 指定された値 ={4}

{0}：エラーが起きた要素名

{1}：エラーが起きた属性名

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた要素の XML 文書内での位置

{4}：指定されていた値

要因：

XBRL 文書で使用できないロール値またはアークロール値が指定されました。
XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は，指定された値をそのまま使用します。

対策：

次の内容を確認して正しい値に修正したあと，再度実行してください。

- 記述している値を確認してください。つづりに誤りがある場合，正しく修正してください。
- カスタムロールまたはカスタムアークロールを使用する場合は，roleRef 要素または arcroleRef 要素が記述されているかどうかを確認し，漏れているときは追加してください。
- roleRef 要素または arcroleRef 要素を記述している場合は，参照先の roleType 要素または arcroleType 要素の usedOn 要素に該当する要素が記載されていないおそれがあります。使用できるロールまたはアークロールに変更するか，usedOn 要素に該当する要素を追加してください。

KEBR10048-E （F，V）

{0} 要素に記述された値はすでに記述されているため無視しました。{1} 個所 ={2} 指定された値 ={3}

{0}：エラーが起きた要素名（roleType /arcroleType）

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

{3}：指定されていた値

要因：

roleType 要素または arcroleType 要素内に記述する usedOn 要素には，兄弟要素であるほかの usedOn 要素と同じ値を記述できません。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。

XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。

処理を続行する場合は，指定された値を無視します。

対策：

usedOn 要素に同じ要素を指す指定を複数記述しているため，不要な記述を削除してから再度実行してください。

なお，名前空間接頭辞が異なっても名前空間 URI が同じ場合，同じ値と判断されます。

KEBR10049-I （F）

{1} を定義している {0} 要素が複数存在しました。{2} 個所 ={3}

{0}：エラーが起きた要素名（roleType /arcroleType）

{1}：エラーが起きた値（roleURI または arcroleURI）

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた要素の XML 文書内での位置

要因：

同じ内容の定義が複数存在します。処理を続行します。

対策：

同じ内容について不要であれば，削除してください。

KEBR10050-E （F，V）

{0} 要素で {1} を再定義しました。{2} 個所 ={3}

{0}：エラーが起きた要素名（roleRef/ roleType /arcroleRef/ arcroleType/ unitDenominator）

{1}：エラーが起きた値（roleURI，arcroleURI，または unitDenominator の値）

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた要素の XML 文書内での位置

要因：

同じ内容の要素を異なる記述で再定義しました。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。

XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。

対策：

次の要素を修正したあと，再度実行してください。

- roleRef 要素 /arcroleRef 要素

  同一 xbrl 要素または linkbase 要素内に，同じ roleURI を持つ roleRef 要素，または同じ arcroleURI を持つ arcroleRef 要素を複数記述しないでください。

  または，標準値を再定義しようとしています。標準値は再定義できないため，適切な値を指定してください。

- roleType 要素 /arcroleType 要素

  XBRL 文書全体で同じ roleURI 属性の値を持つ roleType 要素，または同じ arcroleURI 属性の値を持つ arcroleType 要素があります。しかし，definition 要素および usedOn 要素以下の値が異なるため，必要であれば同じ値を指定してください。この roleType 要素または arcroleType 要素が不要であれば，削除してください。

  または，標準値を再定義しようとしています。標準値は再定義できないため，適切な値を指定してください。

- unitDenominator 要素

  親要素である unit 要素の子要素 unitNumerator 要素と同じ値を持つ unitDenominator 要素があるため，適切に修正してください。

KEBR10051-E （F，V）

{0} 要素が参照する context 要素内の period 要素の指定が誤っています。periodType 属性の値 ={1} 指定できる要素 ={2} {3} 個所 ={4}

{0}：アイテムの要素名

{1}：エラーが起きた periodType の値

{2}：記述できる要素。複数ある場合は，コンマでセパレートされます。

{3}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{4}：エラーが起きた要素（アイテムの要素）の XML 文書内での位置

要因：

periodType 属性の値とアイテムが参照するコンテキスト要素の period 要素の記述に誤りがあります。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。

対策：

次の内容を修正したあと，再度実行してください。

- アイテムが参照するコンテキスト要素が正しいかどうかを確認してください。

periodType 属性の値が "instant" の場合

コンテキスト要素の period 要素の子要素に instant 要素を指定してください。

periodType 属性の値が "duration" の場合

コンテキスト要素の period 要素の子要素に次のどちらかを指定してください。

- forever 要素
- startDate 要素と endDate 要素の組

KEBR10052-W （F，V）

タクソノミー本体を拡張する際に import 要素または include 要素で記述した URI がループしています。{0} 個所 ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた import 要素または include 要素の XML 文書内での位置

要因：

タクソノミー本体の継承関係がループしています。import 要素または include 要素に自分自身の URI を記述したか，拡張元のタクソノミー本体の URI に import/include 済みの URI（解析済みの URI）が指定されています。

XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そ

こで処理を中止します。例外が上がらない場合は処理を続行します。
解析済みのタクソノミー本体を無視し，処理を続行します。

対策：

継承関係がループしているタクソノミー本体の import 要素または include 要素を修正してください。

KEBR10053-E （F，V）

{0} 属性の範囲に誤りがあります。xlink:from が指す要素の {1} 属性の値 ={2} xlink:to が指す要素の {1} 属性の値 ={3} {4} 個所 ={5}

{0}：エラーが起きた属性（weight）

{1}：エラーの要因となった属性（balance）

{2}：xlink:from が指す要素宣言の属性の値。属性の値が省略されている場合は，null が入ります。

{3}：xlink:to が指す要素宣言の属性の値。属性の値が省略されている場合は，null が入ります。

{4}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{5}：エラーが起きた要素（calculationArc）の XML 文書内での位置

要因：

xlink:to 属性および xlink:from 属性で示す要素の balance 属性の値の組み合わせと calculationArc 要素の weight 属性の値の範囲が誤っています。または，calculationArc 要素の weight 属性の値に 0 を指定しています。
XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。

対策：

calculationArc 要素の weight 属性の値に 0 を指定できません。
weight 属性と，calculationArc 要素のロケータ型要素が参照している balance 属性の関係が次のとおりになるように値を修正したあと，再度実行してください。

balance 属性の値		weight 属性の値
from 属性	to 属性	
debit	debit	Positive (> 0)
debit	credit	Negative (< 0)
credit	debit	Negative (< 0)
credit	credit	Positive (> 0)

KEBR10054-E （F，V）

{0} 要素のアークロール {1} に対応する要素の組み合わせに誤りがあります。xlink:from が指す要素の {2} 属性の値 ={3} xlink:to が指す要素の {2} 属性の値 ={4} {5} 個所 ={6}

{0}：エラーが起きた要素（definitionArc）

{1}：エラーが起きた xlink:arcrole 属性（"essence-alias"）

{2}：エラーの要因となった属性名（type）

{3}：xlink:from が指す要素宣言の属性の値

{4}：xlink:to が指す要素宣言の属性の値

{5}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{6}：エラーが起きた要素（calculationArc）の XML 文書内での位置

要因：

xlink:to 属性および xlink:from 属性で示す要素の組み合わせに誤りがあります。
XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は，この指定をそのまま使用します。

対策：

- definitionArc 要素のアークロール属性が "essence-alias" の場合
  xlink:from 側および xlink:to 側の指す要素宣言の type 属性が，同じ型名または同じ型名から派生した型であるように修正してください。

KEBR10056-E （F，V）

{0} 要素の参照先が目的の文書を指定していません。{1} 個所 ={2}

{0}：エラーが起きた要素名

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

要因：

目的の文書とは異なる文書が指定されています。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

次の事項を対策したあと，再度実行してください。

- schemaRef 要素の場合
  目的の文書としてタクソノミー本体を指定してください。
- linkbaseRef 要素の場合

目的の文書としてリンクベースを指定してください。

KEBR10057-E （F，V）

{0} 要素の子要素に {1} 要素の記述はできません。{2} 個所 ={3}

{0}：エラーが起きた要素名

{1}：記述できない要素名

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた要素の XML 文書内での位置

要因：

要素の子要素に記述してはいけない名前空間を持った要素または substitutionGroup に，xbrli:item または xbrli:tuple から派生されたアイテムまたはタプルが記述されました。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は，この指定をそのまま使用します。

対策：

segment 要素または scenario 要素の子要素に指定できない要素が指定されています。次の内容を確認して，正しく修正してから再度実行してください。

- 次の名前空間の要素を含めていないことを確認してください。
  http://www.xbrl.org/2003/instance
- substitutionGroup に xbrli:item または xbrli:tuple から派生したアイテムまたはタプルは記述していないことを確認してください。

KEBR10058-E （F，V）

{0} 型の {1} 要素に対応する {2} 要素に誤った値が記述されました。誤った値 ={3} {4} 個所 ={5}

{0}：エラーが起きたアイテム要素の型名

{1}：エラーが起きたアイテム要素名

{2}：エラーが起きた要素名（measure）

{3}：エラーが起きた要素の値。未指定である場合は，空文字列（""）が出力されます。

{4}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{5}：エラーが起きた要素の XML 文書内での位置

要因：

要素の型に対応する要素の記述に誤りがあります。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上が

らない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は，この指定をそのまま使用します。

対策：

要素の型名に応じて measure 要素に記述できる値が決まります。そのため，要素が参照する unit 要素を正しい値を持つ unit 要素に変更するか，エラーとなった measure 要素に指定する値を正しい値に変更してください。

- monetaryItemType 型
  名前空間 URI が http://www.xbrl.org/2003/iso4217 の値を記述します。
- sharesItemType 型
  名前空間 URI が http://www.xbrl.org/2003/instance の "shares" を記述します。
- pureItemType 型
  名前空間 URI が http://www.xbrl.org/2003/instance の "pure" を記述します。

KEBR10059-E （F，V）

{0} 要素に指定できない値が記述されました。{1} 個所 ={2} 指定されていた値 ={3}

{0}：エラーが起きた要素
{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{2}：エラーが起きた要素の XML 文書内での位置
{3}：要素に記述されていた値

要因：

要素に指定できない値が記述されました。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は，この指定をそのまま使用します。

対策：

unit 要素の子要素である measure 要素の値の接頭辞が次の名前空間 URI を示す場合にローカルパートの値が「pure」か「shares」でなければなりません。値を正しく修正したあと再度実行してください。

- http://www.xbrl.org/2003/instance

KEBR10061-E （F，V）

{0} 属性に使用できない値が記述されています。{1} 個所 ={2}

{0}：エラーが起きた属性名（xbrli:periodType または xbrli:balance または

cyclesAllowed)
{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{2}：エラーが起きた要素の XML 文書内での位置

要因：

アイテムの宣言を行った場合に periodType 属性で "instant" および "duration" 以外の文字（空文字列含む）が指定されました。
または balance 属性で "debit" および "credit" 以外の文字（空文字列含む）が指定されました。
arcroleType 要素の cyclesAllowed 属性で，"any"，"undirected" および "none" 以外の値が指定されました。
XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は，処理を続行します。
処理を続行する場合は，この指定をそのまま使用します。

対策：

次の事項を対策したあと，再度実行してください。

- periodType 属性の場合
  "instant" および "duration" 以外の文字（空文字列含む）が記述されているため正しい値に修正してください。
- balance 属性の場合
  "debit" および "credit" 以外の文字（空文字列含む）が記述されているため正しい値に修正してください。
- cyclesAllowed 属性の場合
  "any"，"undirected" および "none" 以外の文字（空文字列含む）が記述されているため正しい値に修正してください。

KEBR10062-E （F，V）

{0} 要素内には {1} 要素の記述ができません。{2} 個所 ={3}

{0}：拡張リンクの要素名
{1}：エラーが起きた要素名
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：

拡張リンク内に使用できないリソース型要素があります。

- footnoteLink 内には footnote 要素だけ記述できます。
- labelLink 内には，label 要素だけ記述できます。

- referenceLink 内には，reference 要素だけ記述できます。

XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は，この指定を無視します。

対策：

指定できるリソース型要素に修正してから再度実行してください。

KEBR10064-E （F，V）

要素宣言 {0} で記述できない {1} 属性を指定しています。{2} 個所 ={3}

{0}：エラーが起きた要素宣言の種類およびデータ型（xbrli:item または xbrli:tuple）。xbrli:item の場合は，ベースの型名が "," で区切って表示されます。type 属性または局所定義で型を示す値が指定されていない場合は，空文字列になります。もし，type 属性または局所定義の両方が指定された場合は，局所定義の型を示す値が優先されます。
{1}：エラーが起きた属性名
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：

要素宣言で記述できない属性を指定しています。
タプルの宣言の場合，periodType 属性または balance 属性を記述できません。
アイテムの宣言の場合，monetaryItemType 型以外のアイテムには，balance 属性を記述できません。
XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は，この属性を無視します。

対策：

次の事項を対策したあと，再度実行してください。

- タプルの宣言の場合
  periodType 属性または balance 属性の記述を削除します。
- アイテムの宣言の場合
  アイテムが monetaryItemType 型以外であれば，balance 属性の記述を削除します。

KEBR10065-E （F，V）

要素宣言 {0} **で** {1} **属性が記述されていません。**{2} **個所** ={3}

{0}：エラーが起きた要素宣言の種類およびデータ型（xbrli:item または xbrli:tuple）。xbrli:item の場合は，ベースの型名が "," で区切って表示されます。type 属性または局所定義で型を示す値が指定されていない場合は，空文字列になります。もし，type 属性または局所定義の両方が指定された場合は，局所定義の型を示す値が優先されます。

{1}：必要な属性名

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた要素の XML 文書内での位置

要因：

要素宣言の宣言に誤りがあります。

アイテムの宣言の場合，periodType 属性は省略できません。

XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。

XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。

対策：

次の事項を対策したあと，再度実行してください。

- アイテムの宣言の場合
  periodType 属性が省略されている場合は，periodType 属性を記述します。

KEBR10066-E （F，V）

{0} **要素の値と参照先の** {1} **要素の値が一致しません。**{2} **個所** ={3}

{0}：エラーが起きた参照元の要素名（roleRef または arcroleRef）

{1}：エラーが起きた参照先の要素名（roleType または arcroleType）

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた参照元の要素の XML 文書内での位置

要因：

指定されたロールまたはアークロールの値が参照先の要素と一致しません。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

roleRef 要素であれば参照先の roleType 要素と同じ roleURI に修正してから再度実行してください。

arcroleRef 要素であれば参照先の arcroleType 要素と同じ arcroleURI に修正してから再度実行してください。

また，参照先が誤りの場合，正しい参照先に修正してから再度実行してください。

KEBR10067-E （F，V）

{0} 要素の記述個所が誤っています。{1} 個所 ={2}

{0}：エラーが起きた要素名

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

要因：

タクソノミー本体内で記述している要素の記述位置が誤っています。
XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は，この指定を無視します。

対策：

タクソノミー本体に要素を記述する位置は決められています。記述位置と記述できる要素を次に示します。正しい位置に要素を移動したあと，再度実行してください。

- XPath"schema/annotation/appinfo/*" に記述できる要素
  link:linkbaseRef 要素，link:roleType 要素，<link:arcroleType> 要素，および link:linkbase 要素
- XPath"schema/annotation/appinfo/linkbase/*" に記述できる要素
  各種拡張リンク型（presentationLink，definitionLink，calculationLink，labelLink および referenceLink）要素，link:roleRef 要素，link:arcroleRef 要素，および link:documentation 要素

KEBR10068-E （F，V）

{0} 要素には precision 属性と decimals 属性を同時に指定できません。{2} 個所 ={3}

{0}：エラーが起きた要素名

{1}：同時に指定された属性名。コンマ区切りで出力します。

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた要素の XML 文書内での位置

要因：

precision 属性と decimals 属性はどちらか一方しか指定できません。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

precision 属性または decimals 属性のどちらか一方を削除したあと，再度実行してください。

KEBR10069-E （F，V）

{1} 属性の値が同じ {0} 要素は複数定義できません。要素 1：{2} 個所 ={3} 要素 2：{4} 個所 ={5} 指定された値 ={6}

{0}：エラーが起きた要素名

{1}：エラーが起きた属性名

{2}：すでに定義されていた XML 文書の URI.。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：すでに定義されていた要素の XML 文書内での位置

{4}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{5}：エラーが起きた要素の XML 文書内での位置

{6}：記述されていた値

要因：

同じ roleURI または arcroleURI を定義できません。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

要素 1 または要素 2 を削除してください。削除した要素を参照している場合は，残した方を参照するように修正したあと，再度実行してください。

KEBR10070-E （F，V）

接頭辞 {0} に対応する名前空間 URI が宣言されていません。{1} 個所 ={2}

{0}：指定されていた接頭辞

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

要因：

接頭辞に対応する名前空間 URI が宣言されていません。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

個所に対応する要素または要素の値（QName）に対応する名前空間 URI の宣言を確認し，足りなければ追加してから再度実行してください。

KEBR10071-E （F，V）

{0} 要素で利用できない要素を関連づけています。{1} 個所 ={2} 参照先 ={3}

{0}：エラーが起きた要素名

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

{3}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

アーク型要素の参照先が目的のアイテムやタプル要素でありません。または，ロケータ型要素に指定された XPointer 式から複数の要素が求められたか，XPointer 式が要素以外の値（要素の値やコメントなど）を指していました。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

次の内容を確認し，誤っている場合は正しく修正してから再度実行してください。

個所 {2} がアーク型要素を指している場合

このアーク型要素の xlink:arcrole 属性の値と xlink:from，xlink:to 属性に指定された要素の関係が誤っています。

- ラベルリンクやリファレンスリンクなどの，リソース型要素を指すことができるリンクの場合は，xlink:to 属性側でリソース型要素を直接または間接（ロケータ型要素経由）で参照していることを確認してください。
- リソース型の種類が誤っていないか確認してください（ラベルリンクの場合，リソース型要素はラベルしか指せません）。
- アーク型要素の xlink:arcrole 属性の値と xlink:from，xlink:to 属性で参照できる要素は決まっています。正しい内容を参照しているか確認してください（例えば，計算リンクで xlink:arcrole 属性に標準値が指定されている場合，参照できるのはアイテムの要素宣言だけです）。

個所 {2} がロケータ型要素を指している場合

単一の要素宣言，アイテム，タプルまたはリソース型要素を指していないため，単一の値を指していることを確認してください。

KEBR10072-E （F，V）

{0} 要素に記述された値を無視しました。{1} 個所 ={2}

{0}：エラーが起きた要素名

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

要因：

xsi:nil="true" が指定された要素は，値または子要素を持てません。

また，アイテムの種類が数値型アイテムの場合は，precision 属性または decimals 属性を持てません。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
処理を続行する場合は，値または子要素，および属性を無視します。

対策：
不要な値を削除してください。

KEBR10074-W （F，V）

{0} **要素の** priority={1} **を持つ同一のリンクが複数存在します。要素** 1**：**{2}, **個所** ={3} **要素** 2**：**{4}, **個所** ={5}

{0}：エラーが起きた要素名
{1}：エラーが起きた要素の priority 値。priority 属性が省略された場合は，0 が仮定されます。
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置
{4}：エラーが起きた XML 文書の URI（比較した要素の URI）。URI=xx の形式で出力します。URI がない場合は，表示されません。
{5}：エラーが起きた要素（比較した要素）の XML 文書内での位置

要因：
アーク型要素の同じ priority 値を持つ同じリンクが複数あります。
priority 属性を省略した場合，デフォルトとして 0 が仮定されます。
このメッセージは，同じ priority 値を持つ同じリンクのアーク型要素の use 属性に "prohibited" が含まれていない場合に出力されます。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
処理を続行する場合，関係リンクであれば，要素 2 で示すリンクを無視します。ラベルリンクであれば，要素 1 と要素 2 のリンク先のリソースで重複しているリソース型要素のうち，要素 2 で示すリソース型要素を無視します。

対策：
アーク型要素に同じ priority 値を持つリンクが重複しないように，個所 ={3} で示すアーク型要素の priority 属性値の値を適切に修正して，再度実行してください。

KEBR10075-E （F，V）

{0} **要素は** XBRL **ではサポートされていません。**{1} **個所** ={2}

{0}：サポートされていない要素名

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{2}：エラーが起きた要素の XML 文書内での位置

要因：

指定された要素は XBRL の仕様ではサポートされていません。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

指定した要素は，XBRL2.1 仕様では使用しないでください。

KEBR10076-E （F，V）

{0} 属性に空文字列は指定できません。{1} 個所 ={2}

{0}：属性名
{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{2}：エラーが起きた要素の XML 文書内での位置

要因：

targetNamespace 属性が空文字列の場合に発生します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

targetNamespace 属性に空文字列は指定できません。属性値を正しく修正するか，不要な場合は属性を記述しないでください。

KEBR10077-E （F，V）

{0} 要素の {1} 属性の値 {2} に対応する要素宣言の候補が複数存在します。{3} 個所 ={4} 候補 1:{5} 個所 ={6} 候補 2:{7} 個所 ={8}

{0}：エラーが起きた要素名
{1}：エラーが起きた属性名
{2}：指定された値
{3}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{4}：エラーが起きた要素の XML 文書内での位置

{5}：候補 1 の要素宣言がある XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{6}：候補 1 の要素宣言がある XML 文書内での位置
{7}：候補 2 の要素宣言がある XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{8}：候補 2 の要素宣言がある XML 文書内での位置

要因：

属性名が type 属性の場合は，同じ名前空間 URI かつ同じ name 属性値を持つユーザーデータ型の宣言が，異なる文書中に存在している場合に発生します。
属性名が substitutionGroup 属性の場合は，同じ名前空間 URI かつ同じ name 属性値を持つ代替グループのヘッド要素が，異なる文書中に存在している場合に発生します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

アイテム，タプルおよびパートの要素宣言のデータ型および代替グループのヘッド要素が一意となるように，include 要素で取り込まれる文書の関係を適切に修正してください。

KEBR10101-W （F，V）

ユーザーデータ型の定義が同一タクソノミー本体内で定義されていません。{0} **個所** ={1} **使用している型** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：element 要素の type 属性に指定している型名

要因：

element 要素の type 属性にユーザーデータ型が指定されていますが，この型の定義を，同一タクソノミー本体で定義していません。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は，アイテムの型に OTHER_ITEM を仮定して処理を続行します。

対策：

ユーザーデータ型を指定する場合は，使用しているタクソノミー本体と同一文書中で宣言してください。宣言のしかたについては「5.5　取り扱えるデータ型」を参照

してください。

KEBR10102-W （F，V）

type 属性が省略されているか，サポートされていない形式のユーザーデータ型で定義しています。{0} 個所 ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置

要因：

type 属性が省略されているか，またはサポートされていないユーザーデータ型が定義されています。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は，アイテムの型に OTHER_ITEM を仮定して処理を続行します。

対策：

type 属性を追加するか，またはサポートされている形式のユーザーデータ型を指定してください。ユーザーデータ型の形式については，「5.5　取り扱えるデータ型」を参照してください。

KEBR10103-E （F，V）

{0} 要素の {1} 属性の参照先は利用できません。{2} 個所 ={3} 参照先 ={4}

{0}：エラーが起きた要素名
{1}：エラーが起きた属性名
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置
{4}：指定されていた値

要因：

指定されている参照先は利用できません。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

- リンク要素の場合
  参照先の URI を確認してください。必要に応じて参照先を正しく修正して，再度実行してください。
  XBRL2.1 の場合，参照先が存在していてもスコープが異なる場合には，このメッセージが出力されます。一つのインスタンス文書で複数の xbrl 要素があり，片方の xbrl 要素から別の xbrl 要素のアイテムやタプル，コンテキストなどの参照はで

きません。そのため，正しい参照先に修正してから再度実行してください。

- 要素がアイテムの場合
  numericContext 属性には数値コンテキスト，nonNumericContext 属性には非数値コンテキストの ID を指定して，再度実行してください。XBRL2.1 の場合は，context 要素または unit 要素の正しい ID を指定して，再度実行してください。

KEBR10105-W （F，V）

指定されたスキーマ文書のうちタクソノミー本体でないものがありました。{0}

{0}：指定された XML 文書の URI。複数ある場合は，コンマ区切りで出力します。URI=xx の形式で出力します。URI がない場合は，表示されません。

要因：

インスタンス文書で記述された schemaLocation 属性，または XBRLTaxonomyDocumentBuilder で指定された一部のスキーマ文書（URI で示す文書）内に linkbaseRef 要素がありません。linkbaseRef 要素，element 要素または拡張リンクが，一つも含まれていないため，無視しました。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。
処理を続行する場合は，タクソノミー文書とみなされない文書を無視します。

対策：

URI で表示されているスキーマ文書が不要であれば記述から削除してください。

KEBR10106-E （F，V）

拡張されたタクソノミー文書が最大階層数を超えました。{0}

{0}：基底タクソノミー本体の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

要因：

拡張されたタクソノミー文書が最大階層数を超えたため，処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

cbrp.parse.maxTaxonomyExtension プロパティの最大階層数を増やすか，拡張するタクソノミー文書の階層数を減らして，再度実行してください。

KEBR10107-E （F，V）

{0} **属性にタクソノミー文書が指定されていません。**{1} **個所** ={2}

{0}：エラーが起きた属性名

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{2}：エラーが起きた要素の XML 文書内での位置

要因：

schemaLocation 属性にタクソノミー文書が指定されていません。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

schemaLocation 属性にタクソノミー文書を指定し，再度実行してください。

KEBR10108-W （F，V）

アイテムでもタプルでもない要素を参照しています。{0} **個所** ={1} **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：href 属性の値

要因：

指定された参照先の要素の substitutionGroup 属性が xbrli:item または xbrli:tuple でない要素を，拡張リンク内のリンクで参照しています。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。
処理を続行する場合は，loc 要素にそのまま XBRL DOM の関連づけをして処理を続行します。

対策：

参照先を確認し，必要に応じてアイテムまたはタプルの要素宣言を参照するように修正してください。

KEBR10109-W （F，V）

インスタンス文書で利用できない要素を参照しています。{0} **個所** ={1} **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：href 属性の値

要因：

abstract 属性が true で記述された要素を拡張リンク内のリンクで参照しています。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，

warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。
処理を続行する場合は，loc 要素にそのまま XBRL DOM の関連づけをして処理を続行します。

対策：

この要素宣言は抽象要素のため，インスタンス文書では利用できません。
この要素をインスタンス文書で利用する場合は，abstract 属性を false に変更するか，または，abstract 属性を削除してください。

KEBR10110-E （F，V）

{0} 要素が含まれていないため解析できません。{1}

{0}：要素名として "xbrl" が表示されます。
{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

要因：

指定された XML 文書中に xbrl 要素がないため，解析ができませんでした。
または parse() メソッドに渡された要素が xbrl 要素でないため，解析できませんでした。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

xbrl 要素を含めたインスタンス文書を指定してください。
parse() メソッドに直接要素を渡している場合は，この要素を xbrl 要素に変更してください。

KEBR10111-W （F，V）

{0} 要素以下の記述を無視しました。{1} 個所 ={2}

{0}：要素名として "xbrl" が表示されます。
{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{2}：エラーが起きた要素の XML 文書内での位置

要因：

xbrl 要素以下の記述を無視しました。
parse() メソッドでは複数の xbrl 要素を解析できないため，複数の xbrl 要素を含む XML 文書を解析する場合は，parse2() メソッドを使用してください。
XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。

XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は , 指定された要素以下を無視します。

対策：
複数の xbrl 要素を参照する場合は，XBRLInstanceDocumentBuilder クラスの parse2() メソッドを使用してください。

KEBR10112-E （F，V）

ユーザーデータ型の定義が見つかりません。{0} 個所 ={1} 使用している型 ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：element 要素の type 属性または restriction 要素 /extension 要素の base 属性に指定していた発見できない型名

要因：
解析中の文書内，および import 要素または include 要素で取り込まれた文書内に，ユーザーデータ型の定義が見つかりません。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は，項目の型に OTHER_ITEM を仮定して処理を続行します。

対策：
ユーザーデータ型のつづり，およびインポート，インクルード元も含めて定義の有無を確認して，修正したあと再度実行してください。

KEBR10113-E （F，V）

タクソノミー文書に含めることのできる XML 文書 (タクソノミー本体・リンクベース) が最大ファイル数を超えました。最大ファイル数 ={0}

{0}：タクソノミー文書に含めることのできる XML 文書（タクソノミー本体・リンクベース）の最大ファイル数

要因：
タクソノミー文書に含めることのできる XML 文書（タクソノミー本体・リンクベース）が最大ファイル数を超えました。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：
cbrp.parse.DTS.maxDocuments プロパティの最大ファイル数を増やすか，解析するタクソノミー文書のファイル数を減らしてから再度実行してください。

このチェックを行わない場合は，cbrp.parse.DTS.maxDocuments=0 を指定してください。

KEBR10115-W （F，V）

サポートされていないロールが記述されたためリンク先の文書を無視しました。{0} **個所** ={1} **ロール** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：xlink:role 属性の値

要因：

サポートされていないロールが指定されたので，このリンク先の文書を無視しました。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は，このロールをそのまま使用します。

対策：

xlink:role 属性を記述する場合は，サポートされているロールを指定してください。
サポートしているロールについては，XBRL の仕様書を参照してください。

KEBR10116-E （F，V）

アーク型要素 {0} **が参照するロケータ型要素ではリソース型要素を指してはいけません。**{1} **個所** ={2}

{0}：アーク型要素の XML 文書内での位置
{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{2}：ロケータ型要素の XML 文書内での位置

要因：

アーク型要素が参照しているロケータ型要素では，リソース型要素を参照できません。XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は，リンクをそのまま有効にします。

対策：

ロケータ型要素はリソース型要素を指せません。次の点を見直してください。

- use 属性の値に "prohibited" が指定されていることを確認してください。use 属性を省略した場合は use="optional" が仮定されるため，省略しないでください。
- 別の拡張リンクで定義されているリソース型要素を参照していないかを確認してください。参照している場合は，エラーが起きたアーク型要素と同じ拡張リンク内で，リソース型要素を再定義してください。

KEBR10117-W （F，V）

ロール {0} **はサポートしていません。** {1} **個所** ={2}

{0}：単純リンクの記述されていたロール値

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

要因：

単純リンクに記述されたロールはサポートしていません。

XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。

XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。

対策：

単純リンクに記述されたロールはサポートしていません。ロール名が誤っている場合は正しく修正してから再度実行してください。

KEBR11000-I （F）

XBRL **文書にアクセスします。** {0} **プロセッサ** ={1}

{0}：XBRL 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：XML Processor または XLink Processor

要因：

URI で示す XBRL 文書（インスタンス文書，タクソノミー本体，リンクベース）にアクセスします。処理を続行します。

対策：

なし

KEBR11001-I （F）

XBRL **文書を読み込みました。** {0} **プロセッサ** ={1}

{0}：XBRL 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：XML Processor または XLink Processor

要因：

URI で示す XBRL 文書（インスタンス文書，タクソノミー本体，リンクベース）の読み込みに成功しました。処理を続行します。

対策：

なし

KEBR11002-E （F，V）

XBRL **文書の読み込みに失敗しました。**{0} **プロセッサ** ={1} **詳細情報** ={2}

{0}：XBRL 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：XML Processor または XLink Processor。エラーが発生したプロセッサを表示します。

{2}：例外コード（java.io.IOException 例外のメッセージ）

要因：

URI で示す XBRL 文書（インスタンス文書，タクソノミー本体，リンクベース）の読み込みに失敗しました。処理を中止します。

対策：

詳細情報 ={2} で示す例外が発生したため，XBRL 文書の読み込みに失敗しました。詳細情報 ={2} の原因を取り除き，再度実行してください。

KEBR11003-E （F，V）

XBRL **文書の解析に失敗しました。**{0} **プロセッサ** ={1} **詳細情報** ={2}

{0}：XBRL 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：XML Processor または XLink Processor 。エラーが発生したプロセッサを表示します。

{2}：例外コード

要因：

XBRL 文書の解析に失敗しました。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

XBRL 文書の解析で，続行できないエラーが発生しました。XBRL 文書の記述に誤りがあるおそれがあります。このメッセージの前に出力されている，KEBR11004，KEBR11005，KEBR11006 メッセージを参照して原因を取り除き，再度実行してください。修正完了後もこのメッセージが表示される場合は，詳細情報を参照して原因を取り除き，再度実行してください。

KEBR11004-W （F，V）

XBRL **文書の解析で警告が発生しました。行数** ={0} **カラム位置** ={1} **公開識別子** ={2} **システム識別子** ={3} **詳細情報** ={4}

{0}：発生した SAXParseException 例外の getLineNumber() の値が表示されます。
{1}：発生した SAXParseException 例外の getColumnNumber() の値が表示されます。
{2}：発生した SAXParseException 例外の getPublicId() の値が表示されます。
{3}：発生した SAXParseException 例外の getSystemId() の値が表示されます。
{4}：XML プロセッサで発生したエラーまたは警告メッセージ

要因：

XML プロセッサで警告が発生しました。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は，XML 文書の解析を続行します。

対策：

詳細情報 ={4} で示す情報を基に修正し，再度実行してください。

KEBR11005-E （F，V）

XBRL **文書の解析でエラーが発生しました。行数** ={0} **カラム位置** ={1} **公開識別子** ={2} **システム識別子** ={3} **詳細情報** ={4}

{0}：発生した SAXParseException 例外の getLineNumber() の値が表示されます。
{1}：発生した SAXParseException 例外の getColumnNumber() の値が表示されます。
{2}：発生した SAXParseException 例外の getPublicId() の値が表示されます。
{3}：発生した SAXParseException 例外の getSystemId() の値が表示されます。
{4}：XML プロセッサで発生したエラーまたは警告メッセージ

要因：

XML プロセッサでエラーが発生しました。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は XML 文書の解析を続行します。

対策：

詳細情報 ={4} で示す情報を基に修正し，再度実行してください。

KEBR11006-E （F，V）

XBRL 文書の解析で致命的エラーが発生しました。行数 ={0} カラム位置 ={1} 公開識別子 ={2} システム識別子 ={3} 詳細情報 ={4}

{0}：発生した SAXParseException 例外の getLineNumber() の値が表示されます。

{1}：発生した SAXParseException 例外の getColumnNumber() の値が表示されます。

{2}：発生した SAXParseException 例外の getPublicId() の値が表示されます。

{3}：発生した SAXParseException 例外の getSystemId() の値が表示されます。

{4}：XML プロセッサで発生したエラーまたは警告メッセージ

要因：

XML プロセッサで致命的エラーが発生しました。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

詳細情報 ={4} で示す情報を基に修正し，再度実行してください。

KEBR11007-E （F，V）

XPointer の記述に誤りがあります。{0} 個所 ={1} XPointer 式の内容 ={2} 詳細メッセージ ={3}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

{2}：XPointer 式の内容

{3}：誤りの詳細（XLink プロセッサで発生したメッセージ）

要因：

XPointer 式の記述に詳細メッセージに示す誤りがあります。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

詳細メッセージの内容を参照し，正しく修正したあと，再度実行してください。

KEBR11008-W （F，V）

同じロールの同じ言語の label 要素を一つの要素宣言に複数記述しています。{0} 個所 ={1} 要素宣言 ={2}:{3}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

{2}：要素宣言の名前空間 URI（タクソノミー本体の targetNamespace 属性の値）

{3}：要素宣言のローカル名（タクソノミー本体の element 要素の name 属性の値）

要因：

同じロールかつ同じ言語を持つ label 要素を，同じ要素宣言に複数記述しています。
XBRLParseErrorHandler の実装を指定して XBRL 文書の解析を行っていた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書の解析を行っていた場合は処理を続行します。
処理を続行する場合は，この指定をそのまま使用します。

対策：

label 要素の記述を確認し，不要な label 要素があれば，一つの label 要素に修正してください。

KEBR11009-E （F，V）

XPointer の記述に誤りがあります。XPointer 式の内容 ={0} 詳細メッセージ ={1}

{0}：XPointer 式の内容

{1}：誤りの詳細（XLink プロセッサで発生したメッセージ）

要因：

インスタンス文書中に指定された schemaRef 要素または XBRLTaxonomyDocumentBuilder クラスの parse() メソッドの URI のフラグメントに指定された XPointer 式の記述に誤りがあります。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

インスタンス文書の解析中に発生した場合は，インスタンス文書中に記述されている schemaRef 要素の指定を正しい値に変更してから再度実行してください。
XBRLTaxonomyDocumentBuilder クラスの parse() メソッドの延長で発生した場合は，parse() メソッドに渡した URI の指定を正しい値に変更してから再度実行してください。

KEBR11010-E （F，V）

参照先が存在しません。参照先 ={0}

{0}：参照先

要因：

インスタンス文書中に指定された schemaRef 要素または XBRLTaxonomyDocumentBuilder クラスの parse() メソッドで指定された参照先が存在しません。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

インスタンス文書の解析中に発生した場合は，インスタンス文書中に記述されてい

る schemaRef 要素の指定を正しい値に変更してから再度実行してください。
XBRLTaxonomyDocumentBuilder クラスの parse() メソッドの延長で発生した場合は，parse() メソッドに渡した URI の指定を正しい値に変更してから再度実行してください。

KEBR11011-E （F，V）

参照先が目的の文書を指定していません。参照先 ={0}

{0}：参照先

要因：

インスタンス文書中に指定された schemaRef 要素または XBRLTaxonomyDocumentBuilder クラスの parse() メソッドで指定された参照先に，schema 要素が指定されていません。または，XBRL の仕様で決まっている既知のスキーマファイルが指定されました。処理を中止します。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

インスタンス文書の解析中に発生した場合は，インスタンス文書中に記述されている schemaRef 要素の指定を正しい値に変更してから再度実行してください。
XBRLTaxonomyDocumentBuilder クラスの parse() メソッドの延長で発生した場合は，parse() メソッドに渡した URI の指定を正しい値に変更してから再度実行してください。

KEBR11100-I （F）

タクソノミー文書のキャッシュを利用可能にしました。最大キャッシュ数 ={0} 最小キャッシュ数 ={1}

{0}：タクソノミー文書をキャッシュする最大数
{1}：タクソノミー文書をキャッシュする最小数

要因：

タクソノミー文書をキャッシュするため設定を初期化しました。処理を続行します。

対策：

なし

KEBR11101-I （F）

タクソノミー文書のキャッシュは行いません。

要因：

タクソノミー文書をキャッシュしません。処理を続行します。

対策：

なし

KEBR11102-I （F）

キャッシュをクリアしました。

要因：

メモリキャッシュの内容をクリアしました。処理を続行します。

対策：

なし

KEBR11103-I （F）

タクソノミー文書をキャッシュに追加しました。{0}

{0}：キャッシュしたタクソノミー本体の URI。URI=xx の形式で出力されます。複数ある場合は，コンマ「,」で区切られます。

要因：

URI で示すタクソノミー文書をメモリキャッシュに追加しました。処理を続行します。

対策：

なし

KEBR11104-I （F）

タクソノミー文書をキャッシュから削除しました。{0}

{0}：キャッシュから削除したタクソノミー本体の URI。URI=xx の形式で出力されます。複数ある場合は，コンマ「,」で区切られます。

要因：

URI で示すタクソノミー文書をメモリキャッシュから削除しました。処理を続行します。

対策：

なし

KEBR11105-I （F）

キャッシュからタクソノミー文書を取得しました。{0}

{0}：キャッシュしたタクソノミー本体の URI。URI=xx の形式で出力されます。複数ある場合は，コンマ「,」で区切られます。

要因：

メモリキャッシュから URI で示すタクソノミー文書を取得しました。処理を続行します。

対策：

なし

KEBR11106-I （F）

キャッシュ内のタクソノミー文書を更新しました。{0}

{0}：更新したタクソノミー本体の URI。URI=xx の形式で出力されます。複数ある場合は，コンマ「,」で区切られます。

要因：

メモリキャッシュ内に格納されている，URI で示すタクソノミー文書を更新しました。このメッセージは，次のどちらかの条件に該当する場合に出力されます。

- 妥当性検証をしていないタクソノミー文書がメモリキャッシュ内にあり，そのタクソノミー文書を妥当性検証ありで解析した場合。
- 完全な妥当性検証をしていないタクソノミー文書がメモリキャッシュ内にあり，そのタクソノミー文書を完全な妥当性検証ありで解析した場合。

対策：

なし

KEBR11107-I （F）

XML **文書をキャッシュに追加しました。**{0}

{0}：XML 文書の URI。URI=xx の形式で出力されます。

要因：

URI で示す XML 文書をメモリキャッシュに追加しました。処理を続行します。

対策：

なし

KEBR11108-I （F）

XML **文書をキャッシュから削除しました。**{0}

{0}：XML 文書の URI。URI=xx の形式で出力されます。

要因：

URI で示す XML 文書をメモリキャッシュから削除しました。処理を続行します。

対策：

なし

KEBR11109-I （F）

キャッシュの XML **文書の参照を開始しました。**{0} **参照回数** ={1}

{0}：XML 文書の URI。URI=xx の形式で出力されます。

{1}：タクソノミー文書が参照している XML 文書の数

要因：

メモリキャッシュから URI で示す XML 文書の参照を開始しました。処理を続行します。

対策：

なし

KEBR11110-I （F）

キャッシュの XML **文書の参照を終了しました。**{0} **参照回数** ={1}

{0}：XML 文書の URI。URI=xx の形式で出力されます。

{1}：タクソノミー文書が参照している XML 文書の数

要因：

メモリキャッシュから URI で示す XML 文書の参照を終了しました。処理を続行します。

対策：

なし

KEBR11111-I （F）

XLink **文書をキャッシュに追加しました。**{0}

{0}：XLink 文書の URI。URI=xx の形式で出力されます。

要因：

URI で示す XLink 文書をメモリキャッシュに追加しました。処理を続行します。

対策：

なし

KEBR11112-I （F）

XLink **文書をキャッシュから削除しました。**{0}

{0}：XLink 文書の URI。URI=xx の形式で出力されます。

要因：

URI で示す XLink 文書をメモリキャッシュから削除しました。処理を続行します。

対策：

なし

KEBR11113-I （F）

キャッシュの XLink **文書の参照を開始しました。**{0} **参照回数** ={1}

{0}：XLink 文書の URI。URI=xx の形式で出力されます。

{1}：タクソノミー文書が参照している XLink 文書の数

要因：

メモリキャッシュから URI で示す XLink 文書の参照を開始しました。処理を続行します。

対策：

なし

KEBR11114-I （F）

キャッシュの XLink **文書の参照を終了しました。**{0} **参照回数** ={1}

{0}：XLink 文書の URI。URI=xx の形式で出力されます。
{1}：タクソノミー文書が参照している XLink 文書の数

要因：
メモリキャッシュから URI で示す XLink 文書の参照を終了しました。処理を続行します。

対策：
なし

KEBR11115-I （F）

XML **文書をキャッシュから参照しました。**{0}

{0}：XML 文書の URI。URI=xx の形式で出力されます。

要因：
メモリキャッシュから URI で示す XBRL2.1 対応の XML 文書を参照しました。処理を続行します。

対策：
なし

KEBR11200-I （F）

{0} **処理を開始しました。**{1}

{0}：XML プロセッサ，XLink プロセッサ，XML Schema Processor またはパーサー使用者が指定したクラス名
{1}：処理対象の XML 文書または XLink 文書の URI。URI=xx の形式で出力されます。URI がない場合は，表示されません。

要因：
uCosminexus Business Reporting Processor のパーサーが内部的に使用する機能を開始しました。処理を続行します。

対策：
なし

KEBR11201-I （F）

{0} **処理が終了しました。**{1}

{0}：XML プロセッサ，XLink プロセッサ，XML Schema Processor またはパーサー使用者が指定したクラス名
{1}：処理対象の XML 文書または XLink 文書の URI。URI=xx の形式で出力されます。

URI がない場合は，表示されません。

要因：

uCosminexus Business Reporting Processor のパーサーが内部的に使用する機能の処理が終了しました。処理を続行します。

対策：

なし

KEBR11202-I （F）

{0} 処理を開始しました。{1} {2}

{0}：XML プロセッサ，XLink プロセッサまたはパーサー使用者が指定したクラス名

{1}：処理対象の XML 文書または XLink 文書の URI。URI=xx の形式で出力されます。URI がない場合は，表示されません。

{2}：リゾルバの処理が呼び出された場合に resolveEntity() メソッドに渡された「publicId=x systemId=x」の値が表示されます。

ノードを参照している場合は，「HREF=xxx」で参照先が表示されます。

要因：

uCosminexus Business Reporting Processor のパーサーが内部的に使用する機能を開始しました。処理を続行します。

対策：

なし

KEBR11203-I （F）

{0} 処理が終了しました。{1} 返値 ={2}

{0}：XML プロセッサ，XLink プロセッサまたはパーサー使用者が指定したクラス名

{1}：処理対象の XML 文書または XLink 文書の URI。URI=xx の形式で出力されます。URI がない場合は，表示されません。

{2}：{0} で示す処理から戻ってきた際のオブジェクトの情報が出力されます。null が出力された場合は，{0} の処理から null が返ってきたことを示します。それ以外は返されたオブジェクトの情報が出力されます。

要因：

uCosminexus Business Reporting Processor のパーサーが内部的に使用する機能の処理が終了しました。処理を続行します。

対策：

なし

KEBR11204-I （F）

{0} の解析処理を開始しました。{1}

{0}：Instance Document または Taxonomy Document。解析する文書の種類を示します。

{1}：処理対象の XML 文書または XLink 文書の URI。URI=xx の形式で出力されます。複数ある場合は，コンマ「,」で区切られます。URI がない場合は，表示されません。

要因：

uCosminexus Business Reporting Processor のパーサーが指定された文書の解析を開始しました。処理を続行します。

対策：

なし

KEBR11205-I （F）

{0} の解析処理が終了しました。{1}

{0}：Instance Document または Taxonomy Document。解析する文書の種類を示します。

{1}：処理対象の XML 文書または XLink 文書の URI。URI=xx の形式で出力されます。複数ある場合は，コンマ「,」で区切られます。URI がない場合は，表示されません。

要因：

uCosminexus Business Reporting Processor のパーサーが指定された文書の解析を終了しました。処理を続行します。

対策：

なし

KEBR11301-E （F，V）

XBRL 文書の解析で致命的エラーが発生しました。詳細情報 ={0}

{0}：XML Schema プロセッサで発生したエラーまたは警告メッセージ

要因：

XBRL 文書の解析に失敗しました。処理を中止します。XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，処理を中止する前に fatalError() メソッドを呼び出します。

対策：

XBRL 文書の解析において，続行できないエラーが発生しました。XBRL 文書の記述に誤りがある可能性があります。詳細情報を参照して原因を取り除いた後，再度実行してください。

KEBR15001-E （F，V）

{0} の構造が XBRL 仕様に定められている形式に違反しています。{1} 個所 ={2}

{0}：element 要素の名前（{ 名前空間 URI}name 属性の値）

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

要因：

タプルの構造が XBRL 仕様に定められている形式に違反しています。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は error() メソッドを呼び出し，例外が上がった場合はそこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：
XBRL 仕様に定められたタプルの構造に合わせてください。

KEBR15002-E （F，V）

ユーザーデータ型は構造を持つことができません。{0} 個所 ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置

要因：
ユーザーデータ型は fractionItemType 型以外の型から派生する場合は，構造を持つことができません。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は error() メソッドを呼び出し，例外が上がった場合はそこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：
ユーザーデータ型は xsd:simpleContent 要素で正しく定義してください。

KEBR15101-E （F，V）

カスタムアークロール {0} の定義が見つからないためサイクルの検証ができません。

{0}：参照に失敗したカスタムアークロール

要因：
カスタムアークロールが DTS 内にないため，サイクルを検証できませんでした。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は error() メソッドを呼び出し，例外が上がった場合はそこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：
DTS 内に link:arcroleType 要素を追加してください。

KEBR15102-E （F，V）

DTS 内にカスタムアークロール {0} **の定義が複数あるためサイクルの検証ができません。**

{0}：参照に失敗したカスタムアークロール

要因：

cyclesAllowed 属性の値が異なるカスタムアークロールを複数の link:arcroleType 要素で実行しているため，サイクルを検証できませんでした。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

DTS 内に同一の arcroleURI 属性値を持つ link:arcroleType 要素は s-equal のものだけにしてください。

KEBR15103-I （F）

経路 {0}:{1} {2} **個所** ={3}

{0}：経路が複数ある場合に識別番号を出力します。経路が一つ（循環）の場合は表示されません。
{1}：経路として，"->" の前後に "{ 名前空間 URI} ローカルパート " の形式で出力します。左側は xlink:from の値，右側は xlink:to の値が指す要素を示します。
{2}：参照元となったアーク型要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：参照元となったアーク型要素の XML 文書内での位置

要因：

経路の確認結果を通知するメッセージです。この経路はメッセージ KEBR15104-E または KEBR15105-E に対応して出力されるため，サイクルの種類などの情報についてはこれらのメッセージを参照してください。処理を続行します。

対策：

なし

KEBR15104-E （F，V）

経路が循環しています。 cyclesAllowed={0} **アークロール** ={1} {2} **のロール** ={3} **経路** ={4}

{0}：アークロールに対応する cyclesAllowed の値
{1}：アーク型要素のアークロール
{2}：拡張リンクの種類（calculationLink など）
{3}：拡張リンクのロール
{4}：経路として，"->" の前後に "{ 名前空間 URI} ローカルパート " の形式で出力します。

左側は xlink:from の値，右側は xlink:to の値が指す要素を示します。

要因：

経路が循環しています。

XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。

XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

経路が循環しています。メッセージ中に出力された経路を参照し，循環している個所を取り除いてください。

KEBR15105-E （F，V）

経路が複数あります。cyclesAllowed={0} **アークロール** ={1} {2} **のロール** ={3} **経路** ={4}

{0}：アークロールに対応する cyclesAllowed の値

{1}：アーク型要素のアークロール

{2}：拡張リンクの種類（calculationLink など）

{3}：拡張リンクのロール

{4}：経路として，"->" の前後に "{ 名前空間 URI} ローカルパート " の形式で出力します。

左側は xlink:from の値，右側は xlink:to の値が指す要素を示します。

要因：

経路が複数あります。

XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は error() メソッドを呼び出し，例外が上がった場合はそこで処理を中止します。例外が上がらない場合は処理を続行します。

XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は，処理を続行します。

対策：

メッセージ中に出力された経路を参照し，複数経路にならないようにしてください。

KEBR15111-E （F，V）

{0} **と** {1} **の型が異なります。**{2} **のロール** ={3} **アイテム** 1:{4}, **個所** ={5}, **型名** ={6} **アイテム** 2:{7}, **個所** ={8}, **型名** ={9} **リンク：**{10}, **個所** ={11}

{0}：xlink:from 側に対応する element 要素の名前（{ 名前空間 URI} ローカルパート）

{1}：xlink:to 側に対応する element 要素の名前（{ 名前空間 URI} ローカルパート）

{2}：拡張リンクの種類（definitionLink など）

{3}：拡張リンクのロール

{4}：xlink:from 側に対応する element 要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{5}：xlink:from 側に対応する element 要素の XML 文書内での位置
{6}：xlink:from 側に対応する element 要素の型名（{名前空間 URI} ローカルパート）
{7}：xlink:to 側に対応する element 要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{8}：xlink:to 側に対応する element 要素の XML 文書内での位置
{9}：xlink:to 側に対応する element 要素の型名（{名前空間 URI} ローカルパート）
{10}：エラーの元になったアーク型要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{11}：エラーの元になったアーク型要素の XML 文書内での位置

要因：

型が一致しないため，このロールに対応するインスタンス文書中の同一定義の検証を実施しません。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

この検証では，派生元が等しいかは関係なく，element 要素の type 属性で指定された型そのものが一致している必要があります。同じ型名に訂正してください。

KEBR15112-E （F，V）

{0} と {1} の {2} 属性の値が異なります。{3} のロール ={4} アイテム 1:{5}, 個所 ={6}, 属性値 ={7} アイテム 2:{8}, 個所 ={9}, 属性値 ={10} リンク：{11}, 個所 ={12}

{0}：xlink:from 側に対応する element 要素の名前（{名前空間 URI} ローカルパート）
{1}：xlink:to 側に対応する element 要素の名前（{名前空間 URI} ローカルパート）
{2}：エラーとなった element 要素の属性名（balance または periodType）
{3}：拡張リンクの種類（definitionLink など）
{4}：拡張リンクのロール
{5}：xlink:from 側に対応する element 要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{6}：xlink:from 側に対応する element 要素の XML 文書内での位置
{7}：xlink:from 側に対応する element 要素の属性値
{8}：xlink:to 側に対応する element 要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{9}：xlink:to 側に対応する element 要素の XML 文書内での位置
{10}：xlink:to 側に対応する element 要素の属性値
{11}：エラーの元になったアーク型要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{12}：エラーの元になったアーク型要素の XML 文書内での位置

要因：

xsd:element 要素の属性が一致しないため，このロールに対応するインスタンス文書中の同一定義の検証を実施しません。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

タクソノミー文書に記述されている <xsd:element> 要素の属性値を同じにしてください。なお，xbrli:balance 属性の場合は両方を同じ値にしてください（片方または両方にない場合は，このメッセージは出力されません）。

KEBR15121-E （F，V）

{0} **と** {1} **の型が異なります。**{2} **のロール** ={3} **アイテム** 1：{4}, **個所** ={5}, **型名** ={6} **アイテム** 2：{7}, **個所** ={8}, **型名** ={9} **リンク：**{10}, **個所** ={11}

{0}：xlink:from 側に対応する element 要素の名前（{ 名前空間 URI} ローカルパート）
{1}：xlink:to 側に対応する element 要素の名前（{ 名前空間 URI} ローカルパート）
{2}：拡張リンクの種類（calculationLink など）
{3}：拡張リンクのロール
{4}：xlink:from 側に対応する element 要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{5}：xlink:from 側に対応する element 要素の XML 文書内での位置
{6}：xlink:from 側に対応する element 要素の型名（{ 名前空間 URI} ローカルパート）
{7}：xlink:to 側に対応する element 要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{8}：xlink:to 側に対応する element 要素の XML 文書内での位置
{9}：xlink:to 側に対応する element 要素の型名（{ 名前空間 URI} ローカルパート）
{10}：エラーの元になったアーク型要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{11}：エラーの元になったアーク型要素の XML 文書内での位置

要因：

アイテムの型が合わないため，この拡張リンクのロール内の合計値を検証しません。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

型が不一致のため，合計値が正しく計算されていない可能性があります。型を同じ XBRL アイテム型（派生元含む）にしてから検証してください。

KEBR15122-E （F，V）

{0} の型が数値型でありません。{1} のロール ={2} アイテム :{3}, 個所 ={4}, 型名 ={5} リンク：{6}, 個所 ={7}

{0}：非数値型の element 要素の名前（{名前空間 URI} ローカルパート）

{1}：拡張リンクの種類（calculationLink など）

{2}：拡張リンクのロール

{3}：非数値型の element 要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{4}：非数値型の element 要素の XML 文書内での位置

{5}：非数値型の element 要素の型名（{名前空間 URI} ローカルパート）

{6}：エラーの元になったアーク型要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{7}：エラーの元になったアーク型要素の XML 文書内での位置

要因：

アイテムの型が非数値型であるため，この拡張リンクのロール内の合計値を検証しません。

XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。

XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

リンクの関係が誤っている場合は，正しい数値型のアイテムに訂正してください。

タクソノミー文書に記述されている <xsd:element> 要素の型が誤っている場合は，<xsd:element> 要素を数値型に変更してから検証してください。

KEBR15201-E （F，V）

未知の単位 {0} を記述しました。{1} 個所 ={2}

{0}：未知の単位（measure 要素の値のローカルパート部分）

{1}：インスタンス文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：未知の単位のあった measure 要素のインスタンス文書内での位置

要因：

ISO4217 に対応する単位で未知の値が記述されています。

XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，

error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

正しい単位に訂正してください。もし，単位の一覧リストに漏れている場合は，単位の一覧リストに追加してから検証してください。

KEBR15211-E （F，V）

インスタンス文書の {0} **要素に** {1} **属性は指定できません。**{2} **個所** ={3}

{0}：要素名（{名前空間 URI} ローカルパート）

{1}：属性名

{2}：インスタンス文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：不正な属性を指定した要素のインスタンス文書内での位置

要因：

インスタンス文書の要素に記述できない属性を指定された。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

記述できない属性を削除してください。

KEBR15221-E （F，V）

{0} **に対応する** {1} **が存在しません。**{2} **のロール** ={3} {4} **個所** ={5} **リンク：**{6}, **個所** ={7}

{0}：xlink:from 側の element 要素の名前（{名前空間 URI} ローカルパート）

{1}：xlink:to 側の element 要素の名前（{名前空間 URI} ローカルパート）

{2}：拡張リンクの種類（definitionLink など）

{3}：拡張リンクのロール

{4}：インスタンス文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{5}：アイテム / タプルのインスタンス文書内での位置

{6}：エラーの元になったアーク型要素が格納されている XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{7}：エラーの元になったアーク型要素の XML 文書内での位置

要因：

アイテムまたはタプルに対応するアイテムまたはタプルが存在しません。

XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

アイテムまたはタプルに対応するアイテムまたはタプルをインスタンス文書に追加してください。またはこのチェックが不要であれば，リンクの関係を訂正してください。

KEBR15231-E （F，V）

アイテム {0} **とアイテム** {1} **が参照するユニットが** s-equal **でありません。** {2} {3} **のロール** ={4} {5} **アイテム** 1: **個所** ={6},{7} **アイテム** 2: **個所** ={8},{9}

{0}：xlink:from 側のアイテム（{ 名前空間 URI} ローカルパート）
{1}：xlink:to 側のアイテム（{ 名前空間 URI} ローカルパート）
{2}：xlink:from 側のアイテムの contextRef の値（CONTEXT= 値の形式）
{3}：拡張リンクの種類（definitionLink など）
{4}：拡張リンクのロール
{5}：インスタンス文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{6}：xlink:from 側に対応するアイテムのインスタンス文書内での位置
{7}：xlink:from 側に対応するアイテムの unitRef 属性の値（UNIT= 値の形式）
{8}：xlink:to 側に対応するアイテムのインスタンス文書内での位置
{9}：xlink:to 側に対応するアイテムの unitRef 属性の値（UNIT= 値の形式）

要因：

アイテムのユニットが一致しません。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

アイテムが参照しているユニットが s-equal でありません。s-equal の unit に訂正してください。

KEBR15232-E （F，V）

アイテム {0} **とアイテム** {1} **の値が** v-equal **でありません。** {2} {3} **のロール** ={4} {5} **アイテム** 1: **個所** ={6}, **値** ={7} **アイテム** 2: **個所** ={8}, **値** ={9}

{0}：xlink:from 側のアイテム（{ 名前空間 URI} ローカルパート [(INFERRED)]。

(INFERRED) は，アイテムまたはアイテムの値が推測された値の場合に付加されます）
{1}：xlink:to 側のアイテム（{ 名前空間 URI} ローカルパート [(INFERRED)]。
（INFERRED）は，アイテムまたはアイテムの値が推測された値の場合に付加されます）
{2}：xlink:from 側のアイテムの contextRef の値（CONTEXT= 値の形式）
{3}：拡張リンクの種類（definitionLink など）
{4}：拡張リンクのロール
{5}：インスタンス文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{6}：xlink:from 側に対応するアイテムのインスタンス文書内での位置
{7}：xlink:from 側に対応するアイテムの値を「値 (属性情報)」の形式で表示します。ただし，属性情報が数値型（fractionItemType 型を除く）で precision/decimals 属性がある場合は，(属性情報) の部分に (PRECISION= 値 /DECIMALS= 値) と表示されます。非数値型・fractionItemType 型の場合は属性情報がないため，(属性情報) の部分は表示されません。
{8}：xlink:to 側に対応するアイテムのインスタンス文書内での位置
{9}：xlink:to 側に対応するアイテムの値を「値 (属性情報)」の形式で表示します。ただし，属性情報が数値型（fractionItemType 型を除く）で precision/decimals 属性がある場合は，(属性情報) の部分に (PRECISION= 値 /DECIMALS= 値) と表示されます。非数値型・fractionItemType 型の場合は属性情報がないため，(属性情報) の部分は表示されません。

要因：

アイテムの値が一致しません。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

アイテムの値が v-equal でありません。v-equal になる値に訂正してください。

KEBR15233-W （F，V）

アイテム {0} **と** p-equal，c-equal **のアイテムを無視しました。**{1} {2} **のロール** ={3} {4} **個所** ={5}

{0}：アイテム（{ 名前空間 URI} ローカルパート）
{1}：アイテムの contextRef（および unitRef）の値。形式は CONTEXT= 値（および「,UNIT= 値」）
{2}：拡張リンクの種類（definitionLink など）
{3}：拡張リンクのロール
{4}：インスタンス文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{5}：p-equal，c-equal であるアイテムのインスタンス文書内での位置

要因：

同じ名前のアイテムが p-equal，c-equal であるため，このアイテムの指定を無視しました。

XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。

XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

p-equal，c-equal の同じアイテムは同一の検証をしません。このアイテムが不要ならば削除してください。または，アイテムの記述個所を確認し，正しく訂正してください。

KEBR15234-E （F，V）

推測済みのアイテム {0} **の内容が別のアークロールおよびロールで再推測されたアイテムの内容と異なります。**{1} **推測したアイテム：**{2} **のロール** ={3}, **アークロール** ={4}, **値** ={5} **再推測したアイテム：**{6} **のロール** ={7}, **アークロール** ={8}, **値** ={9}

{0}：推測済みのアイテム（{ 名前空間 URI} ローカルパート）

{1}：アイテムの contextRef（および unitRef）の値。形式は CONTEXT= 値（および「,UNIT= 値」）

{2}：推測済みのアイテムの元になった拡張リンクの種類 1（calulationLink など）

{3}：推測済みのアイテムの元になった拡張リンクのロール 1

{4}：推測済みのアイテムの元になったアーク型要素のアークロール 1

{5}：推測済みのアイテムの値を「値 (属性情報)」の形式で表示します。ただし，属性情報が数値型（fractionItemType 型を除く）で precision/decimals 属性がある場合は，(属性情報) の部分に (PRECISION= 値 /DECIMALS= 値) と表示されます。非数値型・fractionItemType 型の場合は属性情報がないため，(属性情報) の部分は表示されません。

{6}：再推測したアイテムの元になった拡張リンクの種類 2（calulationLink など）

{7}：再推測したアイテムの元になった拡張リンクのロール 2

{8}：再推測したアイテムの元になったアーク型要素のアークロール 2

{9}：再推測したアイテムの値を「値 (属性情報)」の形式で表示します。ただし，属性情報が数値型（fractionItemType 型を除く）で precision/decimals 属性がある場合は，(属性情報) の部分に (PRECISION= 値 /DECIMALS= 値) と表示されます。非数値型・fractionItemType 型の場合は属性情報がないため，(属性情報) の部分は表示されません。

要因：

ある関係ネットワークに属するアークから推測されたアイテムの内容と，別の関係ネットワークに属するアークから再推測されたアイテムの内容が異なります。

XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，

error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

拡張リンクの種類およびロールと，アーク型要素のアークロール，推測したアイテムの値を参考に，複数の関係ネットワークから同じアイテムが推測される場合でも，それぞれから推測されるアイテムの内容が同じとなるように訂正してください。

KEBR15241-E （F，V）

アイテム {0} **の値が検証値と等しくありません。値** ={1} **検証値** ={2} {3} {4} **のロール** ={5} {6} **個所** ={7}

{0}：アイテム（{ 名前空間 URI} ローカルパート [(INFERRED)]。(INFERRED) は，アイテムまたはアイテムの値が推測された値の場合に付加されます）
{1}：インスタンス文書中に記述されていた値を「値 (属性情報)」の形式で表示します。ただし，属性情報が数値型（fractionItemType 型を除く）で precision/decimals 属性がある場合は，(属性情報) の部分に (PRECISION= 値 /DECIMALS= 値) と表示されます。非数値型・fractionItemType 型の場合は属性情報がないため，(属性情報) の部分は表示されません。
{2}：検証値を表示します。
{3}：アイテムの contextRef（および unitRef）の値。形式は CONTEXT= 値（および「,UNIT= 値」）
{4}：拡張リンクの種類（calulationLink など）
{5}：拡張リンクのロール
{6}：インスタンス文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{7}：アイテムのインスタンス文書内での位置

要因：

合計値を検証したところ，オリジナルのインスタンス文書中のアイテムの値（合計値）と，計算した結果（検証値）が等しくありません。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，error() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

合計値を確認してください。または，加減算でアイテムの過不足がないか確認してください。

KEBR15242-W （F，V）

アイテム {0} **と** duplicate **のアイテムがありました。**{1} {2} **のロール** ={3} {4} **個所** ={5}

{0}：アイテム（{名前空間 URI} ローカルパート）

{1}：アイテムの contextRef（および unitRef）の値。形式は CONTEXT= 値（および「,UNIT= 値」）

{2}：拡張リンクの種類（calulationLink など）

{3}：拡張リンクのロール

{4}：インスタンス文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{5}：duplicate であるアイテムのインスタンス文書内での位置

要因：

合計値の検証でオリジナルのインスタンス文書中の計算スコープ内に，duplicate のアイテムを検知した場合に発生します。

XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。

XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

また，この警告は duplicate のアイテムを検知するごとに発生します。

対策：

p-equal，u-equal，c-equal の同じアイテムが計算スコープにあります。このアイテムが不要ならば削除してください。または，アイテムの記述個所を確認し，正しく訂正してください。

KEBR15243-W （F，V）

アイテム {0} **の検証を行いませんでした。**{1} {2} **のロール** ={3} {4} **個所** ={5}

{0}：アイテム（{名前空間 URI} ローカルパート）

{1}：アイテムの contextRef（および unitRef）の値。形式は CONTEXT= 値（および「,UNIT= 値」）

{2}：拡張リンクの種類（calulationLink など）

{3}：拡張リンクのロール

{4}：インスタンス文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{5}：アイテムのインスタンス文書内での位置

要因：

合計値の検証でオリジナルのインスタンス文書中の計算スコープ内に，duplicate のアイテムを検知した場合に発生します。

XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで

処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。
また，この警告は検証する合計値の計算スコープ内に duplicate のアイテムが含まれている場合に一度発生します。

対策：

p-equal，u-equal，c-equal の同じアイテムが計算スコープにあります。このアイテムが不要ならば削除してください。または，アイテムの記述個所を確認し，正しく訂正してください。

KEBR15251-W （F，V）

アイテム {0} **の精度が** 0 **のためアイテム** {1} **の検証値の計算に入れません。** {2} {3} **のロール** ={4} {5} **アイテム** : **値** ={6}, **個所** ={7} **合計値アイテム** : **個所** ={8}

{0}：計算に入れないアイテム（{ 名前空間 URI} ローカルパート）
{1}：合計値を示すアイテム（{ 名前空間 URI} ローカルパート）
{2}：アイテムの contextRef（および unitRef の値) の値。形式は CONTEXT= 値（および「,UNIT= 値」）
{3}：拡張リンクの種類（calulationLink など）
{4}：拡張リンクのロール
{5}：インスタンス文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{6}：計算に入れないアイテムを「値 (属性情報)」の形式で表示します。
{7}：計算に入れないアイテムのインスタンス文書内での位置
{8}：合計値を示すアイテムのインスタンス文書内での位置

要因：

アイテムの precision 属性の値に 0 が記述されたか，decimals 属性から推測された precision 属性の結果値が 0 以下になりました。そのため，検証時にはこのアイテムを検証値の算出に入れません。
XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。
XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

precision 属性の値が 0 の場合，有効な値がないため正しく検証できません。
precision 属性（または，decimals 属性の値から推測した結果）の値が 0 にならないように変更してください。

KEBR15252-W （F，V）

アイテム {0} **の値は** {1} **と解釈します。** {2} **個所** ={3}

{0}：アイテム（{名前空間 URI} ローカルパート）

{1}：解釈した値

{2}：インスタンス文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：アイテムのインスタンス文書内での位置

要因：

アイテムの値がシステムの制限値を超えた値である場合に符号付無限大，または，0 として解釈されました。

XBRLParseErrorHandler の実装を指定して XBRL 文書を解析していた場合は，warning() メソッドを呼び出します。このメソッドで例外が上がった場合は，そこで処理を中止します。例外が上がらない場合は処理を続行します。

XBRLParseErrorHandler の実装を指定しないで XBRL 文書を解析していた場合は処理を続行します。

対策：

数値アイテムの値が java.lang.Double の isInfinite() メソッドで true が返される正負の値と，0 の近似値であり小数点以下の桁数が 1024 桁を超える値がないか見直してください。

KEBR16000-E （F，V）

ハイパーキューブ要素宣言が抽象要素ではありません。 {0}, **個所** ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

要因：

要素宣言で，substitutionGroup 属性に xbrldt:hypercubeItem が指定されている場合，abstract 属性に true を指定しなければなりません。

対策：

要素宣言で，substitutionGroup 属性に xbrldt:hypercubeItem が指定されている場合，abstract 属性に true を指定してください。

KEBR16001-E （F，V）

ディメンジョン要素宣言が抽象要素ではありません。 {0}, **個所** ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

要因：

要素宣言で，substitutionGroup 属性に xbrldt:dimensionItem が指定されている場合，abstract 属性に true を指定しなければなりません。

対策：

要素宣言で，substitutionGroup 属性に xbrldt:dimensionItem が指定されている場合，abstract 属性に true を指定してください。

KEBR16002-E （F，V）

all **アークのソースがアイテムではありません。**{0}, **個所** ={1}, **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

{2}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

all アークの xlink:from 側の要素がアイテムではありません。

対策：

all アークの xlink:from 側にはアイテムを定義してください。

KEBR16003-E （F，V）

notAll **アークのソースがアイテムではありません。**{0}, **個所** ={1}, **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

{2}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

notAll アークの xlink:from 側の要素がアイテムではありません。

対策：

notAll アークの xlink:from 側にはアイテムを定義してください。

KEBR16004-E （F，V）

all **アークのターゲットがハイパーキューブではありません。**{0}, **個所** ={1}, **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

{2}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

all アークの xlink:to 側の要素がハイパーキューブではありません。

対策：

all アークの xlink:to 側にはハイパーキューブを定義してください。

KEBR16005-E （F，V）

notAll **アークのターゲットがハイパーキューブではありません。**{0}, **個所** ={1}, **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

notAll アークの xlink:to 側の要素がハイパーキューブではありません。

対策：

notAll アークの xlink:to 側にはハイパーキューブを定義してください。

KEBR16006-E （F，V）

hypercube-dimension **アークのソースがハイパーキューブではありません。**{0}, **個所** ={1}, **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

hypercube-dimension アークの xlink:from 側の要素がハイパーキューブではありません。

対策：

hypercube-dimension アークの xlink:from 側にはハイパーキューブを定義してください。

KEBR16007-E （F，V）

hypercube-dimension **アークのターゲットがディメンジョンではありません。**{0}, **個所** ={1}, **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

hypercube-dimension アークの xlink:to 側の要素がディメンジョンではありません。

対策：

hypercube-dimension アークの xlink:to 側にはディメンジョンを定義してください。

KEBR16008-E （F，V）

dimension-domain **アークのソースが** Explicit Dimension **ではありません。**{0}, **個所** ={1}, **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

dimension-domain アークの xlink:from 側の要素が Explicit Dimension ではありません。

対策：

dimension-domain アークの xlink:from 側の element 要素に Explicit Dimension を指定してください。また，dimension-domain アークの xlink:from 側の要素宣言に xbrldt:typedDomainRef 属性が指定されていないかどうか確認してください。

KEBR16009-E （F，V）

dimension-domain **アークのターゲットがアイテムではありません。**{0}, **個所** ={1}, **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

dimension-domain アークの xlink:to 側の要素がアイテムではありません。

対策：

dimension-domain アークの xlink:to 側にはアイテムを定義してください。

KEBR16010-E （F，V）

domain-member **アークのソースがアイテムではありません。**{0}, **個所** ={1}, **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置
{2}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

domain-member アークの xlink:from 側の要素がアイテムではありません。

対策：

domain-member アークの xlink:from 側にはアイテムを定義してください。

KEBR16011-E （F，V）

domain-member **アークのターゲットがアイテムではありません。**{0}, **個所** ={1}, **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

{2}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

domain-member アークの xlink:to 側の要素がアイテムではありません。

対策：

domain-member アークの xlink:to 側にはアイテムを定義してください。

KEBR16012-E （F，V）

dimension-default **アークのソースが** Explicit Dimension **ではありません。**{0}, **個所** ={1}, **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

{2}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

dimension-default アークの xlink:from 側の要素が Explicit Dimension ではありません。

対策：

dimension-default アークの xlink:from 側に Explicit Dimension を指定してください。また，dimension-default アークの xlink:from 側の要素宣言に xbrldt:typedDomainRef 属性が指定されていないかどうか確認してください。

KEBR16013-E （F，V）

dimension-default **アークのターゲットがアイテムではありません。**{0}, **個所** ={1}, **参照先** ={2}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

{2}：アーク型要素が指しているロケータ型要素の参照先（xlink:href 属性の値）

要因：

dimension-default アークの xlink:to 側の要素がアイテムではありません。

対策：

dimension-default アークの xlink:to 側にはアイテムを定義してください。

KEBR16014-E （F，V）

all **アークに** xbrldt:contextElement **属性がありません。**{0}, **個所** ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置

要因：
all アークに xbrldt:contextElement 属性がありません。

対策：
all アークに xbrldt:contextElement 属性を定義してください。

KEBR16015-E （F，V）

notAll **アークに** xbrldt:contextElement **属性がありません。**{0}, **個所** ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置

要因：
notAll アークに xbrldt:contextElement 属性がありません。

対策：
notAll アークに xbrldt:contextElement 属性を定義してください。

KEBR16016-E （F，V）

属性の値が解決できません。属性 {0}={1}, {2}, **個所** ={3}

{0}：属性名
{1}：属性値
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：
xbrldt:targetRole 属性の値である URI が roleRef 要素で解決できません。

対策：
xbrldt:targetRole 属性の値に対応する roleRef 要素があるかどうか確認してください。

KEBR16017-E （F，V）

属性の値が DTS **に含まれていません。属性** {0}={1}, {2}, **個所** ={3}

{0}：属性名
{1}：属性値

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：

xbrldt:typedDomainRef 属性が指定しているスキーマが DTS に含まれていません。

対策：

xbrldt:typedDomainRef 属性が指定しているスキーマを DTS に含めてください。

KEBR16018-E （F，V）

xbrldt:typedDomainRef **属性がディメンジョン要素宣言以外の場所に指定されています。**{0}, **個所** ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{1}：エラーが起きた要素の XML 文書内での位置

要因：

xbrldt:typedDomainRef 属性がディメンジョン要素宣言以外の場所に指定されています。xbrldt:typedDomainRef 属性を定義できるのは，属性として substitutionGroup="xbrldt:dimensionItem" が定義されている element 要素だけです。

対策：

属性として substitutionGroup="xbrldt:dimensionItem" が定義されていない element 要素には，xbrldt:typedDomainRef 属性を定義しないでください。

KEBR16019-E （F，V）

xbrldt:typedDomainRef **属性の値が不正です。値** ={0}, {1}, **個所** ={2}

{0}：xbrldt:typedDomainRef 属性の値
{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{2}：エラーが起きた要素の XML 文書内での位置

要因：

xbrldt:typedDomainRef 属性の値が不正です。xbrldt:typedDomainRef 属性の値に次のような原因が考えられます。

- 指定しているスキーマがない。
- XML Schema のグローバル要素宣言がない。
- 抽象要素になっている。

対策：

xbrldt:typedDomainRef 属性の値を確認してください。

KEBR16020-E （F，V）

xbrldt:typedDomainRef **属性の** URI **が不正です。値** ={0}, {1}, **個所** ={2}

{0}：xbrldt:typedDomainRef 属性の値

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

要因：

xbrldt:typedDomainRef 属性の URI が不正です。xbrldt:typedDomainRef 属性の URI に次のような原因が考えられます。

- 不正な URI である。
- fragment identifier がない。

対策：

xbrldt:typedDomainRef 属性の URI を確認してください。

KEBR16021-E （F，V）

経路が循環しています。cyclesAllowed={0}, **アークロール** ={1}, {2} **のロール** ={3}, **経路** ={4}

{0}：アークロールに対応する cyclesAllowed の値

{1}：アーク型要素のアークロール

{2}：拡張リンクの種類（definitionLink など）

{3}：拡張リンクのロール

{4}：経路として，"->" の前後に "{ 名前空間 URI} ローカルパート " の形式で出力します。左側は xlink:from の値，右側は xlink:to の値が指す要素を示します。

要因：

DRS に有向循環があります。

対策：

経路が循環しています。このメッセージの例外には，詳細情報として経路が格納されています。この情報から経路を参照して，循環している個所を取り除いてください。

KEBR16022-E （F，V）

不正な継承関係です。cyclesAllowed={0}, {1} **のロール** ={2}, **経路** ={3}

{0}：アークロールに対応する cyclesAllowed の値

{1}：拡張リンクの種類（definitionLink など）

{2}：拡張リンクのロール

{3}：経路として，"->" の前後に "{ 名前空間 URI} ローカルパート " の形式で出力します。左側は xlink:from の値，右側は xlink:to の値が指す要素を示します。

要因：

不正な継承関係です。アイテムが自身のドメイン・メンバにもなっており，循環が

あります。

対策：

経路が循環しています。このメッセージの例外には，詳細情報として経路が格納されています。この情報から経路を参照し，循環している個所を取り除いてください。

KEBR16023-E （F，V）

デフォルト値が複数あります。{0}, **個所** ={1}

{0}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{1}：エラーが起きた要素の XML 文書内での位置

要因：

デフォルト値が複数あります。デフォルト・メンバが一つのディメンジョンに二つ以上あります。

対策：

一つのディメンジョンに定義しているデフォルト値を一つだけにしてください。

KEBR16200-E （F，V）

デフォルト値がインスタンス文書に指定されています。デフォルト値 ={0}, {1}, **個所** ={2}

{0}：デフォルト値

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{2}：エラーが起きた要素の XML 文書内での位置

要因：

デフォルト値がインスタンス文書に指定されています。

対策：

デフォルト値をインスタンス文書に指定しないでください。

KEBR16201-E （F，V）

all **のハイパーキューブが不正です。アイテム** ={0}, **ロール** ={1}, **ハイパーキューブ** ={2}, **不正なディメンジョン** ={3}, {4}, **個所** ={5}

{0}：検証対象のアイテム

{1}：アイテムのロール

{2}：ハイパーキューブ

{3}：不正なディメンジョン

{4}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{5}：エラーが起きた要素の XML 文書内での位置

要因：

all のハイパーキューブが不正です。次に示す原因が考えられます。

- タクソノミー文書でハイパーキューブに定義されているディメンジョンが，インスタンス文書に指定されていない。
- インスタンス文書に指定されているディメンジョンのメンバの値が不正である。

対策：

タクソノミー文書のハイパーキューブの定義に基づいて，インスタンス文書に指定するディメンジョンもしくはメンバの値を見直してください。

KEBR16202-E （F，V）

notAll のハイパーキューブが妥当です。アイテム ={0}, **ロール** ={1}, **ハイパーキューブ** ={2}, **妥当なディメンジョン** ={3}, {4}, **個所** ={5}

{0}：検証対象のアイテム

{1}：アイテムのロール

{2}：ハイパーキューブ

{3}：妥当なディメンジョン

{4}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{5}：エラーが起きた要素の XML 文書内での位置

要因：

notAll のハイパーキューブが妥当です。インスタンス文書に指定されているディメンジョンやメンバが，notAll のハイパーキューブの定義を満たしています。

対策：

タクソノミー文書のハイパーキューブの定義に基づいて，インスタンス文書に指定するディメンジョンもしくはメンバの値を見直してください。

KEBR16203-E （F，V）

{0} **要素は，リンクベースの定義とインスタンス文書の指定が一致しません。アイテム** ={1}, **ロール** ={2}, {3}, **個所** ={4}

{0}：リンクベースの定義と一致しない要素名（segment 要素または scenario 要素）

{1}：検証対象のアイテム

{2}：アイテムのロール

{3}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{4}：エラーが起きた要素の XML 文書内での位置

要因：

リンクベースの xbrldt:contextElement 属性に定義している値である "segment" または "scenario" と，インスタンス文書の context 要素の子要素の要素名である segment 要素または scenario 要素が一致しません。

対策：

リンクベースの定義とインスタンス文書の指定が一致しているかどうか確認してください。

KEBR16204-E （F，V）

使用できないメンバがインスタンス文書に指定されています。アイテム ={0}, **ロール** ={1}, **ハイパーキューブ** ={2}, **メンバ** ={3}, {4}, **個所** ={5}

{0}：検証対象のアイテム

{1}：アイテムのロール

{2}：ハイパーキューブ

{3}：使用できないメンバ

{4}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{5}：エラーが起きた要素の XML 文書内での位置

要因：

タクソノミー文書で xbrldt:usable="false" と定義されているアイテムがインスタンス文書に指定されています。

対策：

タクソノミー文書で xbrldt:usable="false" と定義されているアイテムは，インスタンス文書に指定しないでください。

KEBR16205-E （F，V）

ハイパーキューブに定義されていないディメンジョンがインスタンス文書に指定されています。アイテム ={0}, **ロール** ={1}, **ハイパーキューブ** ={2}, {3}, **個所** ={4}

{0}：検証対象のアイテム

{1}：アイテムのロール

{2}：ハイパーキューブ

{3}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{4}：エラーが起きた要素の XML 文書内での位置

要因：

ハイパーキューブに xbrldt:closed="true" が定義されているが，ハイパーキューブでは定義されていないディメンジョンがインスタンス文書に指定されています。

対策：

タクソノミー文書のハイパーキューブの定義に基づいて，インスタンス文書に指定するディメンジョンの値を見直してください。

KEBR16206-E （F，V）

DRS **に含まれないメンバがインスタンス文書に指定されています。アイテム** ={0}, **ロール** ={1}, **ハイパーキューブ** ={2}, **ディメンジョン** ={3}, **メンバ** ={4}, {5}, **個所** ={6}

{0}：検証対象のアイテム
{1}：アイテムのロール
{2}：ハイパーキューブ
{3}：ディメンジョン
{4}：DRS に含まれないメンバ
{5}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{6}：エラーが起きた要素の XML 文書内での位置

要因：
DTS には含まれているが DRS には含まれていないメンバがインスタンス文書に指定されています。

対策：
DRS に含まれないメンバをインスタンス文書に指定しないでください。

KEBR16207-E （F，V）

一つの context **要素内に同じディメンジョンが複数指定されています。コンテキスト** ={0}, **ディメンジョン** ={1}, {2}, **個所** ={3}

{0}：context 要素の id 属性の値
{1}：dimension 属性の値
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：
一つの context 要素内に，同じ dimension 属性を持つ要素が複数指定されています。

対策：
一つの context 要素内には，同じ dimension 属性を持つ要素を複数指定しないでください。

KEBR16208-E （F，V）

xbrldi:typedMember **要素の** dimension **属性の値が** Typed Dimension **ではありません。コンテキスト** ={0}, **ディメンジョン** ={1}, {2}, **個所** ={3}

{0}：context 要素の id 属性の値
{1}：dimension 属性の値
{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。
{3}：エラーが起きた要素の XML 文書内での位置

要因：
xbrldi:typedMember 要素の dimension 属性の値が Typed Dimension ではありません。

対策：

xbrldi:typedMember 要素の dimension 属性の値に Typed Dimension を指定してください。

KEBR16209-E （F，V）

xbrldi:explicitMember **要素の** dimension **属性が** Explicit Dimension **ではありません。コンテキスト** ={0}, **ディメンジョン** ={1}, {2}, **個所** ={3}

{0}：context 要素の id 属性の値

{1}：dimension 属性の値

{2}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{3}：エラーが起きた要素の XML 文書内での位置

要因：

xbrldi:explicitMember 要素の dimension 属性の値が Explicit Dimension ではありません。

対策：

xbrldi:explicitMember 要素の dimension 属性の値に Explicit Dimension を指定してください。

KEBR16210-E （F，V）

xbrldi:explicitMember **要素に指定されている** QName **が** DTS **にありません。コンテキスト** ={0}, **ディメンジョン** ={1}, **メンバ** ={2}, {3}, **個所** ={4}

{0}：context 要素の id 属性の値

{1}：dimension 属性の値

{2}：xbrldi:explicitMember 要素の要素内容

{3}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{4}：エラーが起きた要素の XML 文書内での位置

要因：

xbrldi:explicitMember 要素の内容に指定されている QName が DTS にありません。

対策：

xbrldi:explicitMember 要素の内容に，DTS にある QName を指定してください。

KEBR16211-W （F，V）

指定されたロールが定義されていません。ロール ={0}, {1}

{0}：指定されたロール

{1}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

要因：

cbrp.parse.validate.dimensions.role プロパティに指定されたロールが定義されていません。

対策：

定義されているロールを指定してください。

KEBR16212-I （F）

アイテムの一つのロールについて妥当性を検証しました。アイテム ={0}, **ロール** ={1}, **結果** ={2}, {3}, **個所** ={4}

{0}：検証対象のアイテム

{1}：アイテムのロール

{2}：アイテムのロールの妥当性について検証した結果。成功の場合には true，失敗の場合には false を出力する。

{3}：エラーが起きた XML 文書の URI。URI=xx の形式で出力します。URI がない場合は，表示されません。

{4}：エラーが起きた要素の XML 文書内での位置

要因：

アイテムの一つのロールについて妥当性を検証しました。

対策：

なし

KEBR19999-E （F，V）

内部エラーが発生しました。詳細情報 ={0}

{0}：詳細コード

要因：

内部エラーが発生しました。処理を中止します。

対策：

前提プログラムプロダクトが正しいかどうか確認してください。

確認後，再度実行しても障害が取り除かれない場合は，システム管理者に連絡してください。

7.4 メッセージ KEBR20000-KEBR29999

KEBR20000-KEBR29999 のメッセージの詳細について番号順に説明します。

KEBR20000-E （F，V）

XBRL DOM はすでに初期化されています。

要因：

XBRLDocumentManager クラスの init() メソッドが複数回呼び出されました。
XBRLDocumentManager クラスの init() メソッドによる初期化処理はアプリケーションで一度しか実行できません。

対策：

XBRLDocumentManager クラスの init() メソッドを一度だけ呼び出すようにアプリケーションを修正してください。

KEBR20001-E （F，V）

XBRL DOM は初期化されていません。

要因：

XBRLDocumentManager クラスの init() メソッドによる初期化処理が実行されていません。

対策：

XBRLDocumentManager クラスの init() メソッドを呼び出して XBRL DOM を初期化して，再度実行してください。

KEBR20002-E （F，V）

引数 {0} に null は指定できません。

{0}：null が指定された引数の名前

要因：

null を指定してはいけない引数に null が指定されました。

対策：

引数に null 以外の値を指定して再度実行してください。

KEBR20004-E （F，V）

引数 {0} に指定された {1} オブジェクトを所有している XBRLInstanceDocument オブジェクトが，この {2} オブジェクトを所有している XBRLInstanceDocument オブジェクトと異なります。

{0}：不正なオブジェクトが指定された引数の名前
{1}：不正なオブジェクトの型
{2}：メソッドを呼び出したオブジェクトの型

要因：

メソッド呼び出しの対象となったオブジェクトを所有している XBRLInstanceDocument オブジェクトと，メソッド呼び出しの引数に指定したオブジェクトを所有している XBRLInstanceDocument オブジェクトは同じでなければなりません。

対策：

所有者が同じオブジェクトを指定して再度実行してください。

KEBR20005-E （F，V）

引数 {0} に指定された {1} オブジェクトは，すでに他の {2} オブジェクトに設定されています。

{0}：不正なオブジェクトが指定された引数の名前

{1}：不正なオブジェクトの型

{2}：不正なオブジェクトの親オブジェクトの型

要因：

引数に指定されたオブジェクトにはすでに親オブジェクトがあります。

対策：

親を持たないオブジェクトを指定して再度実行してください。

KEBR20008-E （F，V）

このラベルはラベルリンクの外で定義されています。よって，このメソッドは使用できません。

要因：

XBRLLabel インターフェースの isRemote() メソッドが true を返す場合は，次のメソッドは呼び出せません。

- getLang()
- getRole()
- getTitle()
- getValue()
- getContainerNode()

対策：

XBRLLabel インターフェースの isRemote() メソッドが true を返す場合は，次のメソッドを呼び出さないようにアプリケーションを修正してください。

- getLang()
- getRole()
- getTitle()
- getValue()
- getContainerNode()

KEBR20009-E （F，V）

このラベルはラベルリンク内の link:label 要素で定義されています。よって，このメソッドは使用できません。

要因：

XBRLLabel インターフェースの isRemote() メソッドが false を返す場合は，getHref() メソッドは呼び出せません。

対策：

XBRLLabel インターフェースの isRemote() メソッドが false を返す場合は，getHref() メソッドを呼び出さないようにアプリケーションを修正してください。

KEBR20010-E （F，V）

この文献への参照はリファレンスリンクの外で定義されています。よって，このメソッドは使用できません。

要因：

XBRLReference インターフェースの isRemote() メソッドが true を返す場合は，getRole()，getTitle()，getParts() の各メソッドは呼び出せません。

対策：

XBRLReference インターフェースの isRemote() メソッドが true を返す場合は，getRole()，getTitle()，getParts() の各メソッドを呼び出さないようにアプリケーションを修正してください。

KEBR20011-E （F，V）

この文献への参照はリファレンスリンク内の link:reference 要素で定義されています。よって，このメソッドは使用できません。

要因：

XBRLReference インターフェースの isRemote() メソッドが false を返す場合は，getHref() メソッドは呼び出せません。

対策：

XBRLReference インターフェースの isRemote() メソッドが false を返す場合は，getHref() メソッドを呼び出さないようにアプリケーションを修正してください。

KEBR20012-E （F，V）

引数 {0} に指定された配列のインデックス {1} の要素に null が指定されました。null は指定できません。

{0}：null を含んだ配列の引数の名前

{1}：配列の引数で null が指定された要素のインデックス

要因：

引数に指定する配列の要素に null が含まれています。

対策：

配列の要素に null 以外の値を設定して再度実行してください。

KEBR20013-E （F，V）

引数 {0} **に指定された配列は，長さが** 0 **の配列で要素を含みません。**

{0}：長さ 0 の配列の名前

要因：

配列の長さは 1 以上でなければなりません。

対策：

引数に長さが 1 以上の配列を指定して再度実行してください。

KEBR20014-E （F，V）

引数 {0} **に不正な値が指定されました。**(**指定された値** ={1}, **指定できる値** ={2})

{0}：指定された引数の名前

{1}：指定された値

{2}：指定できる値

要因：

指定できる値以外の値が引数に指定されました。

対策：

引数に指定できる値を指定して再度実行してください。

KEBR20016-E （F，V）

引数 {0} **に指定された配列のインデックス** {1} **の要素にシステム識別子が設定されていない** org.xml.sax.InputSource **オブジェクトが設定されています。**

{0}：配列の引数の名前

{1}：配列の引数でシステム識別子が設定されていない要素のインデックス

要因：

org.xml.sax.InputSource オブジェクトには必ずシステム識別子を指定する必要があります。

対策：

org.xml.sax.InputSource オブジェクトに適切なシステム識別子を指定して再度実行してください。

KEBR20017-E （F，V）

XBRL **文書の読み込みに失敗しました。**{0}

{0}：XBRL 文書の読み込み処理でスローされた java.io.IOException の詳細メッセージ

要因：

XBRL 文書の読み込み中にエラーが発生したため処理を中断しました。

対策：

エラーの原因を取り除いてから再度実行してください。

KEBR20018-E （F，V）

XBRL **文書の解析に失敗しました。**{0}

{0}：XBRL 文書の解析処理でスローされた org.xml.sax.SAXException の詳細メッセージ

要因：

XBRL 文書の解析中にエラーが発生したため処理を中断しました。

対策：

エラーの原因を取り除いてから再度実行してください。

KEBR20019-E （F，V）

XBRL DOM **内部で予期しないエラーが検出されました。**(**場所** ={0}, **詳細情報１** ={1}, **詳細情報２** ={2}, **詳細情報３** ={3}, **詳細情報４** ={4}, **詳細情報５** ={5}, **詳細情報６** ={6}, **詳細情報７** ={7})

{0}：エラーを検出した場所

{1}：詳細情報１。ない場合は none が出力されます。

{2}：詳細情報２。ない場合は none が出力されます。

{3}：詳細情報３。ない場合は none が出力されます。

{4}：詳細情報４。ない場合は none が出力されます。

{5}：詳細情報５。ない場合は none が出力されます。

{6}：詳細情報６。ない場合は none が出力されます。

{7}：詳細情報７。ない場合は none が出力されます。

要因：

XBRL DOM 内部で予期しないエラーが検出されました。

対策：

システム管理者に連絡してください。

KEBR20020-E （F，V）

このメソッドは {0} **ノード型以外では使用できません。**

{0}：このメソッドが使用できるノード型

要因：

使用できないメソッドで使用されました。

対策：

適切なメソッドを呼び出すようにアプリケーションを修正してください。

KEBR20021-E （F，V）

引数 {0} **に指定された** {1} **オブジェクトを所有している** XBRLInstanceDocument **オブジェクトが，この** XBRLInstanceDocument **オブジェクトではありません。**

{0}：不正なオブジェクトが指定された引数の名前

{1}：不正なオブジェクトの型

要因：

メソッド呼び出しの対象となった XBRLInstanceDocument オブジェクトと，メソッド呼び出しの引数に指定したオブジェクトを所有している XBRLInstanceDocument オブジェクトは同じでなければなりません。

対策：

メソッド呼び出しの対象となった XBRLInstanceDocument オブジェクトが所有するオブジェクトを指定して，再度実行してください。

KEBR20022-E （F，V）

名前空間接頭辞が宣言されていません。(**名前空間接頭辞** ={0})

{0}：名前空間接頭辞

要因：

名前空間接頭辞が宣言されていないので，名前空間接頭辞に対応する名前空間 URI を取得できません。

対策：

適切な名前空間宣言を追加して再度実行してください。

KEBR20023-E （F，V）

名前空間 URI **が宣言されていません。**(**名前空間** URI={0})

{0}：名前空間 URI

要因：

名前空間 URI が宣言されていないので，名前空間 URI に対応する名前空間接頭辞を取得できません。

対策：

適切な名前空間宣言を追加して再度実行してください。

KEBR20026-E （F，V）

XBRLInstanceItem **オブジェクトに値が設定されていません。**

要因：

XBRLInstanceItem オブジェクトに値が設定されていないため，値を取得できません。

対策：

XBRLInstanceItem オブジェクトに値を設定して，再度実行してください。

KEBR20027-E （F，V）

修飾名の形式が不正です。(修飾名 ={0}**)**

{0}：不正な修飾名

要因：

修飾名の形式が不正です。ローカルパートにコロンが含まれているか，名前空間接頭辞またはローカルパートの長さが 0 文字になっています。

対策：

修飾名を正しい形式に修正して，再度実行してください。

KEBR20028-E （F，V）

ID **が重複しています。**(ID={0})

{0}：重複している ID

要因：

ID が重複しています。

対策：

ID が重複しないように，アプリケーションを修正してください。

KEBR20029-E （F，V）

{0} **型のアイテムの文字列表現が不正です。(文字列表現 =**{1}**)**

{0}：データ型の名前

{1}：不正な文字列表現

要因：

XBRLInstanceItem オブジェクトに設定した値の文字列表現が不正です。

対策：

正しい文字列表現を設定して再度実行してください。

KEBR20030-E （F，V）

XBRLPeriod **オブジェクトの内容が** {0} **の場合は，**{1} **メソッドを呼び出せません。**

{0}：xbrli:period 要素の内容

{1}：呼び出せないメソッド

要因：

不正なメソッドが呼び出されました。XBRLPeriod オブジェクトに対して呼び出せるメソッドは，その内容によって異なります。

対策：

XBRLPeriod オブジェクトの内容に応じて正しいメソッドを呼び出すようにアプリ

ケーションを修正してください。

KEBR20031-E （F，V）

リンクの遅延解析処理でエラーが発生しました。{0}

{0}：リンクの遅延解析処理でスローされた java.io.IOException または org.xml.sax.SAXException の詳細メッセージ

要因：

リンクの遅延解析処理でエラーが発生しました。

対策：

エラーの原因を取り除いてから再度実行してください。

KEBR20032-E （F，V）

コンテキスト要素がアイテムの型に適合していません。(コンテキスト要素 ={0}, **アイテム** ={1}, **名前空間** URI={2}, **要素名** ={3})

{0}：numeric または non numeric

{1}：numeric または non numeric

{2}：アイテムの名前空間 URI

{3}：アイテムの要素名

要因：

数値コンテキストを参照する必要があるアイテムに対して非数値コンテキストを設定したり，非数値コンテキストを参照する必要があるアイテムに対して数値コンテキストを設定してはいけません。

対策：

適切なコンテキスト要素を指定して再度実行してください。

KEBR20033-E （F，V）

アイテムの値として指定されたオブジェクトが，アイテムのデータ型に適合していません。(アイテムのデータ型 ={0}, **適合しないオブジェクト** ={1})

{0}：データ型の名前

{1}：合わないオブジェクトの完全修飾名

要因：

アイテムの値として指定されたオブジェクトが，アイテムのデータ型に合いません。XBRLInstanceItem オブジェクトの setValue() メソッドに指定できるオブジェクトの型は，その XBRLInstanceItem のデータ型によって異なります。

対策：

データ型に応じた適切なオブジェクトを指定して再度実行してください。

KEBR20034-E （F，V）

引数 {0} **に指定された** {1} **オブジェクトを所有している** XBRLTaxonomyDocument **オブジェクトが，この** {2} **オブジェクトを所有している** XBRLTaxonomyDocument **オブジェクトと異なります。**

{0}：不正なオブジェクトが指定された引数の名前

{1}：不正なオブジェクトの型

{2}：メソッドを呼び出したオブジェクトの型

要因：

メソッド呼び出しの対象となったオブジェクトを所有している XBRLTaxonomyDocument オブジェクトと，メソッド呼び出しの引数に指定したオブジェクトを所有している XBRLTaxonomyDocument オブジェクトは同じでなければなりません。

対策：

所有者が同じオブジェクトを指定して再度実行してください。

KEBR20035-E （F，V）

引数 {0} **に指定された要素は，このインスタンス文書のルート要素の子要素ではありません。**

{0}：引数の名前

要因：

指定された要素はインスタンス文書のルート要素の子要素でなければなりません。

対策：

インスタンス文書のルート要素の子要素を指定して再度実行してください。

KEBR20036-E （F，V）

グローバル属性の接頭辞には null **を設定できません。**(**名前空間** URI={0})

{0}：接頭辞に null を設定した名前空間 URI

要因：

グローバル属性の接頭辞には null を設定できません。

対策：

グローバル属性が属する名前空間をデフォルトの名前空間として宣言しないように，アプリケーションを修正してください。

KEBR20037-E （F，V）

名前空間接頭辞はすでに他の名前空間 URI **で使用されています。**(**名前空間接頭辞** ={0}, **名前空間** URI={1})

{0}：名前空間接頭辞

{1}：名前空間接頭辞をすでに使用している名前空間 URI

要因：

すでに名前空間 URI に関連づけられている名前空間接頭辞を，別の名前空間 URI に対応づけることはできません。

対策：

一つの名前空間接頭辞を複数の名前空間 URI に対応づけることがないように，名前空間接頭辞を変更し，再度実行してください。

KEBR20038-E （F，V）

"xml" で始まる名前空間接頭辞は登録できません。

要因：

xml で始まる名前空間接頭辞は登録できません。

対策：

名前空間接頭辞を xml で始まるもの以外に変更して再度実行してください。

KEBR20039-E （F，V）

名前空間 URI"http://www.w3.org/XML/1998/namespace" および "http://www.w3.org/2000/xmlns/" の名前空間宣言は登録できません。

要因：

名前空間 URI「http://www.w3.org/XML/1998/namespace」および「http://www.w3.org/2000/xmlns/」の名前空間宣言は登録できません。

対策：

名前空間 URI「http://www.w3.org/XML/1998/namespace」および「http://www.w3.org/2000/xmlns/」の名前空間宣言を登録しないようにアプリケーションを修正してください。

KEBR20040-E （F，V）

指定された XBRLNamespaceDeclaration オブジェクトは登録されていません。

要因：

指定された XBRLNamespaceDeclaration オブジェクトが登録されていません。

対策：

登録されている XBRLNamespaceDeclaration オブジェクトを指定して再度実行してください。

KEBR20041-E （F，V）

名前空間 URI"{0}" は XBRL 文書中で使用されているため，名前空間宣言を削除できません。

{0}：名前空間 URI

要因：

使用中の名前空間 URI を宣言している名前空間宣言を削除できません。

対策：

使用中の名前空間 URI を宣言している名前空間宣言を削除しないように，アプリケーションを修正してください。または，削除したい名前空間宣言で宣言している名前空間 URI に属する要素・属性を，XBRL 文書中から削除して，再度名前空間宣言の削除を実行してください。

KEBR20042-E （F，V）

指定された XBRLSchemaLocation **オブジェクトは登録されていません。**

要因：

指定された XBRLSchemaLocation オブジェクトが登録されていません。

対策：

登録されている XBRLSchemaLocation オブジェクトを指定して再度実行してください。

KEBR20043-E （F，V）

指定されたコンテキスト要素はインスタンス文書中に存在しません。

要因：

指定されたコンテキスト要素がインスタンス文書中にありません。

対策：

インスタンス文書中にあるコンテキスト要素を指定して再度実行してください。

KEBR20044-E （F，V）

指定されたアイテムは，タクソノミー文書で定義されていません。(**名前空間** URI={0}, **要素名** ={1})

{0}：定義されていないアイテムの名前空間 URI

{1}：定義されていないアイテムの要素名

要因：

指定されたアイテムがタクソノミー文書で定義されていません。

対策：

タクソノミー文書で定義されているアイテムを指定して再度実行してください。

KEBR20045-E （F，V）

指定されたタプルは，タクソノミー文書で定義されていません。(**名前空間** URI={0}, **要素名** ={1})

{0}：定義されていないタプルの名前空間 URI

{1}：定義されていないタプルの要素名

要因：

指定されたタプルがタクソノミー文書で定義されていません。

対策：
タクソノミー文書で定義されているタプルを指定して再度実行してください。

KEBR20046-E （F，V）

指定された脚注はインスタンス文書中に存在しません。

要因：
指定された脚注がインスタンス文書中にありません。

対策：
インスタンス文書中にある脚注を指定して再度実行してください。

KEBR20047-E （F，V）

指定された linkbaseRef 要素はインスタンス文書中に存在しません。

要因：
指定された linkbaseRef 要素がインスタンス文書中にありません。

対策：
インスタンス文書中にある linkbaseRef 要素を指定して再度実行してください。

KEBR20048-E （F，V）

このノードはインポートできません。(ノードの型 ={0})

{0}：ノードの型

要因：
インポートできない型のノードをインポートしようとしました。

対策：
インポートできないノードをインポートしないように，アプリケーションを修正してください。または，インポートできるノードを指定して再度実行してください。

KEBR20050-E （F，V）

グローバル属性の名前空間接頭辞が指定されていません。(名前空間 URI={0}, **属性名** ={1})

{0}：接頭辞が指定されていない名前空間 URI
{1}：属性のローカル名

要因：
グローバル属性が属する名前空間が，デフォルトの名前空間として宣言されています。グローバル属性には名前空間接頭辞が必須です。

対策：
グローバル属性が属する名前空間の名前空間接頭辞を宣言する名前空間宣言を追加し，再度実行してください。

KEBR20051-E （F，V）

引数 {0} に指定された {1} オブジェクトを所有している XBRLDocumentManager オブジェクトが，この {2} オブジェクトを所有している XBRLDocumentManager オブジェクトと異なります。

{0}：不正なオブジェクトが指定された引数の名前

{1}：不正なオブジェクトの型

{2}：メソッドを呼び出したオブジェクトの型

要因：

メソッド呼び出しの対象となったオブジェクトを所有している XBRLDocumentManager オブジェクトと，メソッド呼び出しの引数に指定したオブジェクトを所有している XBRLDocumentManager オブジェクトは同じでなければなりません。

対策：

所有者が同じオブジェクトを指定して再度実行してください。

KEBR20052-E （F，V）

XBRL 文書の出力に失敗しました。{0}

{0}：XBRL 文書の出力処理でスローされた javax.xml.transform.TransformerException の詳細メッセージ

要因：

XBRL 文書の出力処理でエラーが発生しました。

対策：

エラーの原因を取り除いてから再度実行してください。

KEBR20053-E （F，V）

指定された脚注はこの要素と関連づけられていません。

要因：

指定された脚注がこの要素と関連づけられていません。

対策：

この要素と関連づけられている脚注を指定して再度実行してください。

KEBR20054-E （F，V）

引数 {0} に指定されたノードは，このノードの先祖かこのノード自身です。

{0}：引数の名前

要因：

あるノードの先祖またはそのノード自身を，そのノードの子として設定できません。

対策：

あるノードの先祖またはそのノード自身を，そのノードの子として設定しないように，アプリケーションを修正してください。

KEBR20055-E （F，V）

引数 {0} **に指定された要素は，このタプルの子要素ではありません。**

{0}：引数の名前

要因：

指定された要素はこのタプルの子要素でなければなりません。

対策：

このタプルの子要素を指定して再度実行してください。

KEBR20056-E （F，V）

period **要素の子要素の組み合わせが不正です。** (type={0}, instant={1}, startDate={2}, endDate={3}, duration={4})

{0}：種類

{1}：instant 要素

{2}：startDate 要素

{3}：endDate 要素

{4}：duration 要素

要因：

period 要素の子要素の組み合わせが不正です。

対策：

period 要素の子要素の正しい組み合わせを指定して再度実行してください。

KEBR20057-E （F，V）

終了日は開始日よりもあとの日付でなければなりません。 (startDate={0}, endDate={1})

{0}：startDate 要素

{1}：endDate 要素

要因：

終了日は開始日よりもあとの日付でなければなりません。

対策：

終了日が開始日よりもあとの日付となるように，開始日と終了日の値を修正してください。

KEBR20058-E （F，V）

属性値が不正です。 (**属性名** ={0}, **属性値** ={1})

{0}：属性名

{1}：属性値

要因：

属性値が不正です。

対策：

正しい属性値を指定して再度実行してください。

KEBR20061-E （F，V）

アイテムの内容に子要素が含まれているため値を取得できません。

要因：

アイテムの内容に子要素が含まれているため値を取得できません。

対策：

子要素が含まれないようにアイテムの内容を修正して，再度実行してください。

KEBR20062-E （F，V）

引数 prefix が null 以外の場合，引数 uri に空文字列は指定できません。

要因：

引数 prefix が null 以外の場合に，引数 uri に空文字列が指定されました。

対策：

引数 prefix が null 以外の場合は，引数 uri に空でない文字列を指定して，再度実行してください。

KEBR20063-E （F，V）

{0}.{1} は未サポートです。

{0}：クラス名またはインターフェース名

{1}：コンストラクタ名またはメソッド名

要因：

サポートしていないコンストラクタまたはメソッドを呼び出しました。

対策：

サポートしていないコンストラクタまたはメソッドを使用しないように，アプリケーションを修正してください。

KEBR20064-E （F，V）

引数 {0} と {1} に同じオブジェクトを指定できません。

{0}：引数の名前

{1}：引数の名前

要因：

複数の引数に同じオブジェクトを指定していますが，これらの引数に同じオブジェクトを指定できません。

対策：

複数の引数に同じオブジェクトを指定しないようにアプリケーションを修正してください。または，それぞれの引数に異なるオブジェクトを指定して再度実行してく

ださい。

KEBR20065-E （F，V）

指定された要素はタクソノミー文書で抽象的な要素として定義されているため，インスタンスを生成できません。(名前空間 URI={0}, **要素名** ={1})

{0}：抽象的な要素の名前空間 URI

{1}：抽象的な要素の要素名

要因：

引数に指定された要素はタクソノミー文書で抽象的な要素として定義されているため，インスタンスを生成できません。抽象的な要素とは，要素宣言（xsd:element 要素）で abstract 属性に true が指定されている要素のことです。

対策：

抽象的な要素のインスタンスを生成しないようにアプリケーションを修正してください。または，抽象的でない要素を指定して再度実行してください。

KEBR20067-E （F，V）

通し番号が最大値（2147483647**）に達したため，名前空間接頭辞を生成できません。**

要因：

インスタンス文書に記述されている名前空間宣言を読み込む処理で，自動生成された名前空間接頭辞の通し番号が最大値（2,147,483,647）に達しました。

対策：

インスタンス文書中に記述された名前空間宣言で，名前空間接頭辞が必要以上に重複していないかどうかを見直してください。

KEBR20068-W （F）

XBRL **文書の出力で警告が発生しました。詳細情報** ={0}

{0}：XSLT プロセッサが生成した javax.xml.transform.TransformerException の詳細メッセージ

要因：

XSLT プロセッサの処理で警告が発生し，エラーイベントリスナの warning() メソッドが呼び出されました。

対策：

警告の原因を調査し，必要に応じて原因を取り除いてから再度実行してください。

KEBR20069-E （F，V）

デフォルトの名前空間が宣言されているため，名前空間に属さない XBRLQName **オブジェクトの修飾名を生成できません。(ローカルパート** ={0}, **デフォルトの名前空間** ={1})

{0}：XBRLQName オブジェクトのローカルパート

{1}：デフォルトの名前空間

要因：

デフォルトの名前空間が宣言されている場合は，名前空間に属さない XBRLQName オブジェクトを次に示す場所に設定できません。

- xsd:QName 型のアイテム
- xsd:NOTATION 型のアイテム
- xbrli:measure 要素
- xbrldi:explicitMember 要素の dimension 属性
- xbrldi:explicitMember 要素の内容
- xbrldi:typedMember 要素の dimension 属性

名前空間に属さない XBRLQName オブジェクトの getNamespaceURI() メソッドは null を返します。

対策：

デフォルトの名前空間が宣言されている場合に，名前空間に属さない XBRLQName オブジェクトを次に示す場所に設定しないようにアプリケーションを見直してください。

- xsd:QName 型のアイテム
- xsd:NOTATION 型のアイテム
- xbrli:measure 要素
- xbrldi:explicitMember 要素の dimension 属性
- xbrldi:explicitMember 要素の内容
- xbrldi:typedMember 要素の dimension 属性

KEBR20070-E （F，V）

名前空間に属さない要素または修飾名が存在するため，デフォルトの名前空間を宣言できません。

要因：

名前空間に属さない要素または修飾名があるため，デフォルトの名前空間を宣言できません。

対策：

名前空間に属さない要素や修飾名がある場合は，デフォルトの名前空間を宣言しないようにアプリケーションを修正してください。

KEBR20071-E （F，V）

継続期間は正の値でなければなりません。(指定された値 ={0})

{0}：指定された不正な継続期間

要因：

xbrli:duration 要素に指定する継続期間は正の値でなければなりません。

対策：

xbrli:duration 要素に設定する継続期間として負の値（例：-P1Y2M）やゼロ（例：P0Y）を指定しないように，アプリケーションを修正してください。

KEBR20072-E （F）

XBRL 文書の出力でエラーが発生しました。詳細情報 ={0}

{0}：XSLT プロセッサが生成した javax.xml.transform.TransformerException の詳細メッセージ

要因：

XSLT プロセッサの処理でエラーが発生し，エラーイベントリスナの error() メソッドが呼び出されました。

対策：

エラーの原因を調査し，必要に応じて原因を取り除いてから再度実行してください。

KEBR20073-E （F）

XBRL 文書の出力で致命的なエラーが発生しました。詳細情報 ={0}

{0}：XSLT プロセッサが生成した javax.xml.transform.TransformerException の詳細メッセージ

要因：

XSLT プロセッサの処理で致命的なエラーが発生し，エラーイベントリスナの fatalError() メソッドが呼び出されました。

対策：

エラーの原因を取り除いてから再度実行してください。

KEBR20100-E （F，V）

操作対象の文書が準拠している XBRL のバージョンが {0} の場合は，{1}.{2} メソッドは利用できません。

{0}：XBRL のバージョン

{1}：クラス名またはインターフェース名

{2}：コンストラクタ名またはメソッド名

要因：

XBRL の特定のバージョンでだけ利用できるメソッドを呼び出しました。しかし，操作対象の XBRL 文書が準拠している XBRL のバージョンでは，そのメソッドは利用できません。

対策：

XBRL の特定のバージョンでだけ利用できるメソッドを使用する場合は，操作対象の XBRL 文書が準拠している XBRL のバージョンを確認してから呼び出すように，アプリケーションを見直してください。

KEBR20101-E （F，V）

バージョン {0} **の** XBRL **仕様に準拠した文書を処理対象とするように設定されている場合は，** {1}.{2} **メソッドは利用できません。**

{0}：XBRL のバージョン

{1}：クラス名またはインターフェース名

{2}：コンストラクタ名またはメソッド名

要因：

XBRL の特定のバージョンを処理対象とするように設定されいる場合にだけ利用できるメソッドを呼び出しました。しかし，処理対象として設定されているバージョンでは，そのメソッドは利用できません。

対策：

XBRL の特定のバージョンを処理対象とするように設定されいる場合にだけ利用できるメソッドを使用する場合は，処理対象として設定されている XBRL のバージョンを確認してから呼び出すように，アプリケーションを見直してください。

KEBR20102-E （F，V）

指定された XBRLTaxonomyDocumentBuilder **オブジェクトが処理対象とする** XBRL **のバージョンと，この** XBRLInstanceDocumentBuilder **オブジェクトが処理対象とする** XBRL **のバージョンが異なります。**(XBRLInstanceDocumentBuilder **オブジェクトが処理対象とする** XBRL **のバージョン** ={0}, XBRLTaxonomyDocumentBuilder **オブジェクトが処理対象とする** XBRL **のバージョン** ={1})

{0}：XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL のバージョン

{1}：XBRLTaxonomyDocumentBuilder オブジェクトが処理対象とする XBRL のバージョン

要因：

XBRLInstanceDocumentBuilder オブジェクトと，XBRLInstanceDocumentBuilder.setTaxonomyDocumentBuilder() メソッドに指定する XBRLTaxonomyDocumentBuilder オブジェクトは，処理対象とする XBRL のバージョンが同じでなければなりません。

対策：

XBRLInstanceDocumentBuilder.setTaxonomyDocumentBuilder() メソッドには，処理対象の XBRL のバージョンが同じである XBRLTaxonomyDocumentBuilder オブジェクトを指定するように，アプリケーションを見直してください。

KEBR20103-E （F，V）

指定された {0} **オブジェクトが準拠する** XBRL **のバージョンと，この** {2} **オブジェクトが処理対象とする** XBRL **のバージョンが異なります。**({2} **オブジェクトが処理対象とする** XBRL **のバージョン** ={3}, {0} **オブジェクトが準拠する** XBRL **のバージョン** ={1})

{0}：XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL のバージョン

{1}：XBRLTaxonomyDocument オブジェクトが準拠する XBRL のバージョン

要因：

XBRLInstanceDocumentBuilder.newInstanceDocument() メソッドに指定する XBRLTaxonomyDocument オブジェクトが準拠する XBRL のバージョンと，XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL のバージョンは同じでなければなりません。

対策：

XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL のバージョンに準拠する XBRLTaxonomyDocument オブジェクトを指定するように，アプリケーションを見直してください。

KEBR20104-E （F，V）

準拠している XBRL **のバージョンが異なるため，ノードをインポートできません。**(**インポート元のバージョン** ={0}, **インポート先のバージョン** ={1})

{0}：インポート元が準拠している XBRL のバージョン

{1}：インポート先が準拠している XBRL のバージョン

要因：

準拠している XBRL のバージョンが異なる文書間ではノードをインポートできません。

対策：

準拠している XBRL のバージョンが異なる文書間でノードをインポートしないように，アプリケーションを見直してください。

KEBR20105-E （F，V）

属性を認識できません。(**属性名** ={0})

{0}：属性名

要因：

指定した属性の名前に誤りがあります。

対策：

属性の名前に誤りがないかどうか確認し，正しい属性名を指定するように，アプリケーションを見直してください。

KEBR20106-E （F，V）

インスタンス文書のルート要素が DOM ツリーにつながっていないため出力できません。

要因：

XBRLInstanceDocumentWriter を使用してインスタンス文書をストリームへ出力する場合は，インスタンス文書のルート要素が DOM ツリーにつながっている必要があります。

対策：

インスタンス文書のルート要素を DOM ツリーにつなげてから出力するように，アプリケーションを見直してください。

KEBR20107-E （F，V）

名前空間に属さない要素または修飾名が存在するため，デフォルトの名前空間をリセットする宣言（xmlns=""）を削除できません。

要因：

名前空間に属さない要素または修飾名が存在するため，デフォルトの名前空間をリセットする宣言（xmlns=""）を削除できません。

対策：

名前空間に属さない要素や修飾名が存在する場合には，デフォルトの名前空間をリセットする宣言（xmlns=""）を削除しないように，アプリケーションを見直してください。

KEBR20108-E （F，V）

引数 {0} に指定された文字列が {1} 型ではありません。(指定された文字列 ={2})

{0}：引数の名前

{1}：データ型の名前

{2}：指定された文字列

要因：

メソッドの引数に不正な文字列を設定しました。

対策：

正しい文字列を指定するように，アプリケーションを見直してください。

KEBR20109-E （F，V）

引数 {0} に指定された URI が絶対 URI ではありません。(指定された URI={1})

{0}：引数の名前

{1}：指定された URI

要因：

絶対 URI を指定しなければならない引数に対して，絶対 URI ではない URI を指定しました。

対策：

絶対 URI を指定するように，アプリケーションを見直してください。

KEBR20110-E （F，V）

引数 {0} には空の URI 参照は指定できません。

{0}：引数の名前

要因：

空の URI 参照を指定できない引数に対して，空文字列を指定しました。

対策：

空文字列を指定しないように，アプリケーションを見直してください。

KEBR20111-E （F，V）

カスタムロール {0} の定義を参照する link:roleRef 要素がすでに存在します。

{0}：カスタムロール

要因：

インスタンス文書中では，roleURI 属性の値が同じである link:roleRef 要素が複数存在できません。このため，すでにインスタンス文書中に存在する link:roleRef 要素の roleURI 属性と同じ値の roleURI 属性を持つ別の link:roleRef 要素をインスタンス文書に追加できません。

対策：

roleURI 属性の値が重複する link:roleRef 要素を追加しないように，アプリケーションを見直してください。

KEBR20112-E （F，V）

カスタムアークロール {0} の定義を参照する link:arcroleRef 要素がすでに存在します。

{0}：カスタムアークロール

要因：

インスタンス文書中では，arcroleURI 属性の値が同じである link:arcroleRef 要素が複数存在できません。このため，すでにインスタンス文書中に存在する link:arcroleRef 要素の arcroleURI 属性と同じ値の arcroleURI 属性を持つ別の link:arcroleRef 要素をインスタンス文書に追加できません。
処理を中止します。

対策：

arcroleURI 属性の値が重複する link:arcroleRef 要素を追加しないように，アプリケーションを見直してください。

KEBR20113-E （F，V）

link:roleRef 要素の roleURI 属性には XBRL 仕様で規定された標準のロールを指定できません。(指定された標準のロール ={0})

{0}：標準のロール

要因：

XBRL 仕様で規定された標準のロールを link:roleType 要素で再定義できません。このため，link:roleType 要素を参照する link:roleRef 要素の roleURI 属性には標準のロールを設定できません。

対策：

link:roleRef 要素の roleURI 属性に標準のロールを指定しないように，アプリケーションを見直してください。

KEBR20114-E （F，V）

link:arcroleRef 要素の arcroleURI 属性には XBRL 仕様で規定された標準のアークロールを指定できません。(指定された標準のアークロール ={0})

{0}：標準のアークロール

要因：

XBRL 仕様で規定された標準のアークロールを link:arcroleType 要素で再定義できません。このため，link:arcroleType 要素を参照する link:arcroleRef 要素の arcroleURI 属性には標準のアークロールを設定できません。

対策：

link:arcroleRef 要素の arcroleURI 属性に標準のアークロールを指定しないように，アプリケーションを見直してください。

KEBR20115-E （F，V）

XBRL 仕様で規定された標準のロール {0} は {1} 要素で使用できません。

{0}：標準のロール

{1}：標準のロールを使用しようとした要素

要因：

XBRL 仕様で規定されている標準のロールを，そのロールを使用できない要素で使用しようとしました。

対策：

この要素で使用できる標準のロールまたはカスタムロールに修正してください。

KEBR20116-E （F，V）

XBRL 仕様で規定された標準のアークロール {0} は {1} 要素で使用できません。

{0}：標準のアークロール

{1}：標準のアークロールを使用しようとした要素

要因：

XBRL 仕様で規定されている標準のアークロールを，そのアークロールを使用できない要素で使用しようとしました。

対策：

この要素で使用できる標準のアークロールまたはカスタムアークロールに修正してください。

KEBR20117-E （F，V）

カスタムロール {0} **の定義を参照する** link:roleRef **要素が見つかりません。**

{0}：カスタムロール

要因：

インスタンス文書でカスタムロールを使用する場合には，カスタムロールの定義を参照する link:roleRef 要素がインスタンス文書中に存在しなければなりません。

対策：

次のどちらかの観点で，アプリケーションを見直してください。

- カスタムロールを使用する前に，カスタムロールの定義を参照する link:roleRef 要素をインスタンス文書に追加する。
- インスタンス文書で使用するカスタムロールを，標準値を用いて指定するか，または定義済みのカスタムロールを用いて指定する。

KEBR20118-E （F，V）

カスタムアークロール {0} **の定義を参照する** link:arcroleRef **要素が見つかりません。**

{0}：カスタムアークロール

要因：

インスタンス文書でカスタムアークロールを使用する場合には，カスタムアークロールの定義を参照する link:arcroleRef 要素がインスタンス文書中に存在しなければなりません。

対策：

次のどちらかの観点で，アプリケーションを見直してください。

- カスタムアークロールを使用する前に，カスタムアークロールの定義を参照する link:arcroleRef 要素を，インスタンス文書に追加する。
- インスタンス文書で使用するカスタムアークロールを，標準値を用いて指定するか，または定義済みのカスタムアークロールを用いて指定する。

KEBR20119-E （F，V）

カスタムロール {0} **は** {1} **要素で使用できるように定義されていません。**

{0}：カスタムロール

{1}：カスタムロールを使用しようとした要素

要因：

カスタムロールを，そのカスタムロールを使用できるように定義されていない要素で使用しようとしました。

対策：

次のどちらかの観点で，アプリケーションを見直してください。

- カスタムロールを使用する前に，カスタムロールの定義を参照する link:roleRef 要素を，インスタンス文書に追加する。
- インスタンス文書で使用するカスタムロールを，標準値を用いて指定するか，または定義済みのカスタムロールを用いて指定する。

KEBR20120-E （F，V）

カスタムアークロール {0} は {1} 要素で使用できるように定義されていません。

{0}：カスタムアークロール

{1}：カスタムアークロールを使用しようとした要素

要因：

カスタムアークロールを，そのアークカスタムロールを使用できるように定義されていない要素で使用しようとしました。

対策：

次のどちらかの観点で，アプリケーションを見直してください。

- カスタムアークロールを使用する前に，カスタムアークロールの定義を参照する link:arcroleRef 要素を，インスタンス文書に追加する。
- インスタンス文書で使用するカスタムアークロールを，標準値を用いて指定するか，または定義済みのカスタムアークロールを用いて指定する。

KEBR20121-E （F，V）

XBRLRoleRef **オブジェクトはアクティブではないため，**XBRLRoleType **オブジェクトを取得できません。**

要因：

XBRLRoleRef.getRoleType() メソッドを使用して link:roleRef 要素が指す link:roleType 要素を取得できるのは，その link:roleRef 要素を表す XBRLRoleRef オブジェクトがアクティブである場合だけです。XBRLRoleRef オブジェクトがアクティブでない場合は，getRoleType() メソッドを呼び出してはいけません。

対策：

非アクティブな XBRLRoleRef オブジェクトに対して getRoleType() メソッドを呼び出さないように，アプリケーションを見直してください。

KEBR20122-E （F，V）

XBRLArcroleRef **オブジェクトはアクティブではないため，**XBRLArcroleType **オブジェクトを取得できません。**

要因：

XBRLArcroleRef.getArcroleType() メソッドを使用して link:arcroleRef 要素が指す link:arcroleType 要素を取得できるのは，その link:arcroleRef 要素を表す XBRLArcroleRef オブジェクトがアクティブである場合だけです。XBRLArcroleRef オブジェクトがアクティブでない場合は，getArcroleType() メソッドを呼び出してはいけません。

対策：

非アクティブな XBRLArcroleRef オブジェクトに対して getArcroleType() メソッドを呼び出さないように，アプリケーションを見直してください。

KEBR20123-E （F，V）

指定された脚注は，指定された link:footnoteArc **要素によってこの要素と関連づけられていません。(** **脚注リンクのロール** ={0}, **アークロール** ={1}, order **属性** ={2})

{0}：脚注リンクのロール

{1}：アークロール

{2}：order 属性の値

要因：

指定された脚注が，指定された link:footnoteArc 要素によってこの要素に関連づけられていないため，処理を続行できません。

対策：

この要素と関連づけられている脚注，および，この要素と脚注を関連づけるために使用されている link:footnoteArc 要素のアークロールと order 属性の値を指定するように，アプリケーションを見直してください。

KEBR20124-E （F，V）

指定された単位要素はインスタンス文書中に存在しません。

要因：

指定された単位要素がインスタンス文書中に存在しないため，処理を続行できません。

対策：

インスタンス文書中に存在する単位要素を指定して再度実行してください。

KEBR20125-E （F，V）

基本単位のローカルパートが不正です。(**名前空間** URI={0}, **ローカルパート** ={1})

{0}：基本単位の名前空間 URI

{1}：基本単位のローカルパート

要因：

単位要素を構成する基本単位として指定された XBRLQName オブジェクトの名前

空間 URI が "http://www.xbrl.org/2003/instance" である場合は，ローカルパートは "pure" または "shares" である必要があります。

対策：

単位要素の名前空間 URI が "http://www.xbrl.org/2003/instance" である場合は，ローカルパートには "pure" または "shares" を設定するように，アプリケーションを見直してください。

KEBR20126-E （F，V）

単位要素の分子と分母に同じ内容の基本単位が存在します。(**名前空間** URI={0}, **ローカルパート** ={1})

{0}：基本単位の名前空間 URI

{1}：基本単位のローカルパート

要因：

単位要素の内容が分数である場合は，分子と分母を構成する基本単位に内容が同じである基本単位を含めることはできません。

対策：

分子と分母に同じ内容の基本単位を含めないように，アプリケーションを見直してください。

KEBR20127-E （F，V）

単位要素の内容が，アイテムのベースデータ型と矛盾しています。(**名前空間** URI={0}, **要素名** ={1}, **ベースデータ型** ={2})

{0}：アイテムの名前空間 URI

{1}：アイテムの要素名

{2}：アイテムのベースデータ型

要因：

アイテムが参照する単位要素の内容には，アイテムのベースデータ型によって幾つかの制約があります。詳細については，XBRLInstanceItem インターフェースの「6.6.13(8)　setUnit」を参照してください。

対策：

アイテムが参照する単位要素の内容がアイテムのベースデータ型と矛盾しないように，アプリケーションを見直してください。

KEBR20128-E （F，V）

XBRLUnit2 **オブジェクトの内容が** {0} **の場合は，**{1} **メソッドを呼び出せません。**

{0}：単位要素の内容

{1}：呼び出せないメソッド

要因：

XBRLUnit2 オブジェクトに対して呼び出せるメソッドは，その内容によって異なります。

対策：

XBRLUnit2 オブジェクトの内容に応じて正しいメソッドを呼び出すように，アプリケーションを見直してください。

KEBR20129-E （F，V）

要素宣言において nillable **属性に** true **が指定されていない要素には** xsi:nil **属性を設定できません。**(**名前空間** URI={0}, **要素名** ={1})

{0}：xsi:nil 属性を設定しようとした要素の名前空間 URI

{1}：xsi:nil 属性を設定しようとした要素の要素名

要因：

要素に xsi:nil 属性を付けるためには，その要素の要素宣言で nillable 属性に true が指定されている必要があります。

対策：

要素宣言で nillable 属性に true が指定されていない要素に xsi:nil 属性を設定しないように，アプリケーションを見直してください。

KEBR20130-E （F，V）

xsi:nil **属性に** true **が設定されている要素に対して** {0}.{1} **メソッドを呼び出せません。**(**名前空間** URI={2}, **要素名** ={3})

{0}：インターフェース名

{1}：メソッド名

{2}：要素の名前空間 URI

{3}：要素の要素名

要因：

xsi:nil 属性に true が指定されている要素の内容は空でなければなりません。このため，その要素がアイテムである場合は値を設定できません。また，その要素がタプルである場合は子要素を追加できません。さらに，xsi:nil 属性に true が指定されている要素がアイテムである場合は，そのアイテムに対して precision 属性および decimals 属性を設定できません。

対策：

xsi:nil 属性に true が設定されている要素の内容を設定しないように，アプリケーションを見直してください。

KEBR20131-E （F，V）

XBRLInstanceDocument.createNumericItem **メソッドでは非数値アイテムを生成できません。**(**名前空間** URI={0}, **要素名** ={1})

{0}：アイテムの名前空間 URI
{1}：アイテムの要素名

要因：
XBRLInstanceDocument.createNumericItem() メソッドでは非数値アイテムを生成できません。非数値アイテムを生成する場合は，XBRLInstanceDocument.createNonNumericItem() メソッドを使用してください。

対策：
非数値アイテムを生成する場合は XBRLInstanceDocument.createNonNumericContext() メソッドを使用するように，アプリケーションを見直してください。

KEBR20132-E （F，V）

XBRLInstanceDocument.createNonNumericItem **メソッドでは数値アイテムを生成できません。(名前空間** URI={0}, **要素名** ={1})

{0}：アイテムの名前空間 URI
{1}：アイテムの要素名

要因：
XBRLInstanceDocument.createNonNumericItem() メソッドでは数値アイテムを生成できません。数値アイテムを生成する場合は，XBRLInstanceDocument.createNumericItem() メソッドを使用してください。

対策：
数値アイテムを生成する場合は XBRLInstanceDocument.createNumericContext() メソッドを使用するように，アプリケーションを見直してください。

KEBR20133-E （F，V）

ベースデータ型が FRACTION_ITEM **であるアイテムには，**precision **属性および** decimals **属性は指定できません。(名前空間** URI={0}, **要素名** ={1}, precision={2}, decimals={3})

{0}：アイテムの名前空間 URI
{1}：アイテムの要素名
{2}：指定された precision 属性の値
{3}：指定された decimals 属性の値

要因：
数値アイテムには precision 属性と decimals 属性のどちらか一方を指定できますが，ベースデータ型が FRACTION_ITEM である場合は，precision 属性および decimals 属性のどちらも指定できません。

対策：
ベースデータ型が FRACTION_ITEM であるアイテムに precision 属性や decimals 属性を指定しないように，アプリケーションを見直してください。

KEBR20134-E （F，V）

precision **属性と** decimals **属性の両方を省略できません。**(**名前空間** URI={0}, **要素名** ={1})

{0}：アイテムの名前空間 URI

{1}：アイテムの要素名

要因：

数値アイテム（ベースデータ型が FRACTION_ITEM のものは除く）には，precision 属性と decimals 属性のどちらか一方を指定する必要があります。両方を省略できません。

対策：

precision 属性と decimals 属性のどちらか一方を指定するように，アプリケーションを見直してください。

KEBR20135-E （F，V）

precision **属性と** decimals **属性の両方を同時に指定できません。**(**名前空間** URI={0}, **要素名** ={1}, precision={2}, decimals={3})

{0}：アイテムの名前空間 URI

{1}：アイテムの要素名

{2}：指定された precision 属性の値

{3}：指定された decimals 属性の値

要因：

数値アイテム（ベースデータ型が FRACTION_ITEM のものは除く）には，precision 属性と decimals 属性のどちらか一方だけを指定できます。両方を同時に指定できません。

対策：

precision 属性と decimals 属性の両方を同時に指定しないように，アプリケーションを見直してください。

KEBR20136-E （F，V）

このアイテムに対して {0} **属性を設定・取得できません。**(**名前空間** URI={1}, **要素名** ={2}, **ベースデータ型** ={3}, **数値** ={4})

{0}："precision" または "decimals"

{1}：アイテムの名前空間 URI

{2}：アイテムの要素名

{3}：アイテムのベースデータ型

{4}：数値かどうか "Yes" or "No"

要因：

アイテムが precision 属性または decimals 属性を持てるのは，アイテムが数値であり，かつ，ベースデータ型が FRACTION_ITEM 以外の場合だけです。アイテムが

非数値か，または，ベースデータ型が FRACTION_ITEM である場合は，precision 属性および decimals 属性を持つことはできません。

対策：

非数値か，または，ベースデータ型が FRACTION_ITEM であるアイテムに対して precision 属性や decimals 属性の操作を行わないように，アプリケーションを見直してください。

KEBR20137-E （F，V）

非数値アイテムに対する単位要素の設定・取得はできません。(名前空間 URI={0}, **要素名** ={1})

{0}：アイテムの名前空間 URI

{1}：アイテムの要素名

要因：

XBRLInstanceItem インターフェースの setUnit() ／ getUnit() メソッドを使用して単位要素の設定・取得ができるのは，数値アイテムに対してだけです。非数値アイテムに対する単位要素の設定・取得はできません。

対策：

非数値アイテムに対して単位要素の設定・取得をしないように，アプリケーションを見直してください。

KEBR20138-E （F，V）

指定された文字列から XBRLFraction **オブジェクトを生成できませんでした。(名前空間** URI={0}, **要素名** ={1}, **指定された文字列** ={2})

{0}：アイテムの名前空間 URI

{1}：アイテムの要素名

{2}：指定された文字列

要因：

XBRLInstanceItem.setValue() メソッドの引数に指定された文字列から XBRLFraction オブジェクトを生成できませんでした。ユーザーデータ型で，かつ，ベースデータ型が FRACTION_ITEM であるアイテムに対して値を設定する場合は，XBRLInstanceItem.setValue() メソッドの引数には，XBRLFraction クラスのコンストラクタに指定するのと同じ形式の文字列を指定しなければなりません。

対策：

正しい文字列を指定するように，アプリケーションを見直してください。

KEBR20139-E （F，V）

ベースデータ型が {0} **であるアイテムに対して，文字列表現による値の設定・取得はできません。(名前空間** URI={1}, **要素名** ={2})

{0}：アイテムのベースデータ型

{1}：アイテムの名前空間 URI
{2}：アイテムの要素名

要因：
ベースデータ型が FRACTION_ITEM であるアイテムに対しては，XBRLInstanceItem インターフェースの setString() メソッドおよび getString() メソッドによる値の設定・取得はできません。setValue() メソッドおよび getValue() メソッドを使用してください。

対策：
ベースデータ型が FRACTION_ITEM であるアイテムに対して，XBRLInstanceItem インターフェースの setString() メソッドおよび getString() メソッドを使用しないように，アプリケーションを見直してください。

KEBR20140-E （F，V）

時期要素の内容が，アイテムの要素宣言の xbrli:periodType **属性における指定と矛盾しています。(** **名前空間** URI={0}, **要素名** ={1}, xbrli:periodType **属性** ={2}, **時期要素の内容** ={3})

{0}：アイテムの名前空間 URI
{1}：アイテムの要素名
{2}：xbrli:periodType 属性の値
{3}：時期要素の内容

要因：
要素宣言の xbrli:periodType 属性に "instant" が指定された場合は，そのアイテムが参照するコンテキスト要素内の時期要素の内容は，XBRLPeriod.INSTANT でなければなりません。また，xbrli:periodType 属性の値が "duration" である場合は，XBRLPeriod.FOREVER または XBRLPeriod.STARTDATE_ENDDATE でなければなりません。

対策：
要素宣言の xbrli:periodType 属性での指定と時期要素の内容が正しい組み合わせとなるように，アプリケーションを見直してください。

KEBR20141-E （F，V）

時期要素の内容の意味は変更できません。(**既存の時期要素の内容** ={0}, **新たに設定しようとした時期要素の内容** ={1})

{0}：既存の時期要素の内容
{1}：新たに設定しようとした時期要素の内容

要因：
XBRLContext.setPeriod() メソッドでコンテキスト要素に時期要素を設定する場合に，新たに設定する時期要素の内容に制約があります。詳細については，XBRLContext インターフェースの「6.6.2(7)　setPeriod」を参照してください。

対策：

制約に従った時期要素を設定するように，アプリケーションを見直してください。

KEBR20142-E （F，V）

操作対象の文書が準拠している XBRL **のバージョンが** {0} **の場合は，** XBRLPeriod.STARTDATE_DURATION **および** XBRLPeriod.DURATION_ENDDATE **は使用できません。**

{0}：XBRL のバージョン

要因：

XBRLPeriod.STARTDATE_DURATION および XBRLPeriod.DURATION_ENDDATE は XBRL のバージョンが 2.0 の場合にだけ使用でき，そのほかのバージョンでは使用できません。

対策：

XBRL のバージョンが 2.0 以外の場合には，XBRLPeriod.STARTDATE_DURATION および XBRLPeriod.DURATION_ENDDATE を使用しないように，アプリケーションを見直してください。

KEBR20143-E （F，V）

XBRLIdManager **オブジェクトに登録しようとしたインスタンス文書は，すでに管理対象となっているインスタンス文書とは異なる** XML **文書に記述されています。**

要因：

XBRLIdManager オブジェクトでは複数のインスタンス文書の ID をまとめて管理できますが，一つの XBRLIdManager オブジェクトが管理するインスタンス文書はすべて同じ XML 文書に記述されたものでなければなりません。つまり，XBRLIdManager オブジェクトで管理するすべてインスタンス文書について，XBRLInstanceDocument.getW3CDOMNode() メソッドが返す org.w3c.dom.Document オブジェクトは同じでなければなりません。

対策：

異なる XML 文書に記述されたインスタンス文書をそれぞれ別々の XBRLIdManager オブジェクトで管理するように，アプリケーションを見直してください。

KEBR20144-E （F，V）

バージョン {0} **の** XBRL **仕様に準拠した文書を** XBRLIdManager **オブジェクトへ登録できません。**

{0}：XBRL のバージョン

要因：

XBRLIdManager オブジェクトでは，XBRL2.0 に準拠したインスタンス文書を管理できません。

対策：

XBRL2.0 に準拠したインスタンス文書を XBRLIdManager オブジェクトへ登録しないように，アプリケーションを見直してください。

KEBR20145-E （F，V）

要素宣言における定義内容が異なるアイテムはインポートできません。(**名前空間** URI={0}, **要素名** ={1}, **ベースデータ型** ={2}/{3}, xbrli:periodType **属性** ={4}/{5})

{0}：アイテムの名前空間 URI

{1}：アイテムの要素名

{2}：インポート元でのベースデータ型

{3}：インポート先でのベースデータ型

{4}：インポート元での xbrli:periodType 属性の値

{5}：インポート先での xbrli:periodType 属性の値

要因：

インポート元でのアイテムの要素宣言とインポート先でのアイテムの要素宣言で，ベースデータ型または xbrli:periodType 属性の値が異なっているため，アイテムをインポートできません。

対策：

インポート元での要素宣言とインポート先での要素宣言とで定義内容が異なるアイテムをインポートしないように，アプリケーションを見直してください。要素宣言での定義内容に誤りがある場合は，タクソノミー文書を修正してください。

KEBR20146-E （F，V）

この要素に脚注が関連づけられているため，id **属性を削除できません。**(**名前空間** URI={0}, **要素名** ={1}, id={2})

{0}：要素の名前空間 URI

{1}：要素の要素名

{2}：id 属性の値

要因：

要素がアクティブで，かつ，脚注と関連づけられている場合は，その要素の id 属性を削除できません。

対策：

脚注と関連づけられている要素の id 属性を削除しないように，アプリケーションを見直してください。

KEBR20147-E （F，V）

引数 {0} に指定されたシステム識別子の形式が不正です。(指定されたシステム識別子 ={1})

{0}：引数の名前

{1}：不正なシステム識別子

要因：

指定したシステム識別子の形式が不正です。

対策：

正しいシステム識別子を指定するように，アプリケーションを見直してください。

KEBR20148-E （F，V）

引数 {0} に指定された配列のインデックス {1} の要素に設定されているシステム識別子の形式が不正です。(指定されたシステム識別子 ={2})

{0}：配列の引数の名前

{1}：配列の引数で不正なシステム識別子が設定されている要素のインデックス

{2}：不正なシステム識別子

要因：

指定したシステム識別子の形式が不正です。

対策：

正しいシステム識別子を指定するように，アプリケーションを見直してください。

KEBR20149-E （F，V）

指定された XBRLTaxonomyDocumentBuilder オブジェクトがタクソノミー文書の解析時に完全な妥当性検証を行うように設定されていません。

要因：

インスタンス文書の解析時に完全な妥当性検証を実行するよう XBRLInstanceDocumentBuilder オブジェクトに設定されていますが，そのオブジェクトに設定する XBRLTaxonomyDocumentBuilder オブジェクトに対して，タクソノミー文書の解析時に完全な妥当性検証を実行するよう設定されていません。処理を中止します。

対策：

XBRLTaxonomyDocumentBuilder オブジェクトが完全な妥当性検証を実行されるよう，設定を見直してください。

KEBR20150-E （F，V）

指定されたプロパティを認識できません。(プロパティ名 ={0})

{0}：プロパティ名

要因：

指定したプロパティ名に誤りがあります。

処理を中止します。

対策：

正しいプロパティ名を指定してください。

KEBR20151-E （F，V）

{0} **プロパティの値として** null **は指定できません。**

{0}：プロパティ名

要因：

null を指定できないプロパティに対して null が指定されました。プロパティ値として null を指定できるかどうかは，プロパティごとに決まっています。

処理を中止します。

対策：

null 以外の正しい値を指定するようにアプリケーションを修正してください。

KEBR20152-E （F，V）

{0} **プロパティの値として指定されたオブジェクトの型が不正です。(** **不正な型** ={1}, **正しい型** ={2})

{0}：プロパティ名

{1}：プロパティ値として指定されたオブジェクトの完全修飾名

{2}：プロパティ値として指定できるオブジェクトの完全修飾名

要因：

プロパティ値として指定されたオブジェクトの型が不正です。指定できるオブジェクトの型は，プロパティごとに決まっています。

処理を中止します。

対策：

正しい型のオブジェクトを指定してください。

KEBR20153-E （F，V）

{0} **プロパティに不正な値が指定されました。(** **指定された値** ={1}, **指定できる値** ={2})

{0}：プロパティ名

{1}：指定された値

{2}：指定できる値

要因：

プロパティ値として不正な値が指定されました。指定できる値は，プロパティごとに決まっています。

処理を中止します。

対策：

正しい値を指定してください。

KEBR20154-E （F，V）

{0} プロパティの値として指定された配列の要素数が 0 です。要素数は 1 以上でなければなりません。

{0}：プロパティ名

要因：

プロパティの値として指定する配列の要素数は，1 以上にする必要があります。
処理を中止します。

対策：

要素数が 1 以上の配列を指定してください。

KEBR20155-E （F，V）

{0} プロパティの値として指定された配列のインデックス {1} の要素が null です。配列の要素に null は指定できません。

{0}：プロパティ名

{1}：null が指定された要素のインデックス

要因：

値が null である要素は，プロパティの値として指定する配列に含めることはできません。
処理を中止します。

対策：

null を含まない配列を指定してください。

KEBR20156-E （F，V）

引数 {0} に不正な値が指定されました。(指定された値 ={1})

{0}：指定された引数の名前

{1}：指定された値

要因：

引数 {0} に対し，指定できない値 {1} が指定されました。
処理を中止します。

対策：

引数の値を指定し直してください。指定できる値については，例外が発生したメソッドの説明を参照してください。

KEBR22001-E （F，V）

引数 {0} に {1} オブジェクトは指定できません。

{0}：引数の名前

{1}：不正なオブジェクトの型または完全修飾名

要因：

引数 {0} に対し，指定できないオブジェクト {1} が指定されました。
処理を中止します。

対策：

正しいオブジェクトを指定してください。
指定できるオブジェクトは，各メソッドの内容を確認してください。

KEBR22002-E （F，V）

{0} オブジェクトの名前はすでに使用されています。(名前 ={1})

{0}：不正なオブジェクトの型
{1}：使用されている名前

要因：

SimpleTaxonomy オブジェクトの中に，オブジェクト {0} が保持する名前（getName() メソッドで取得される値）と同じ名前のアクティブなオブジェクトが存在するため，メソッドの操作に失敗しました。
名前はアクティブな SimpleElementDefinition オブジェクト（派生したインターフェースを含む）で一意でなければなりません。
処理を中止します。

対策：

オブジェクトの保持する名前が一意になるように setName() メソッドで変更し，再度実行してください。

KEBR22003-E （F，V）

{0} オブジェクトの ID はすでに使用されています。(ID={1})

{0}：不正なオブジェクトの型
{1}：使用されている ID

要因：

SimpleTaxonomy オブジェクトの中に，オブジェクト {0} が保持する ID（getId() メソッドで取得される値）と同じ ID を持つアクティブなオブジェクトが存在するため，メソッドの操作に失敗しました。
次の原因が考えられます。

- アクティブな SimpleElementDefinition オブジェクト（派生したインターフェースを含む）で，ID が一意でない。
- 不正なオブジェクトの型が SimpleLabel の場合，アクティブな SimpleLabel オブジェクトで，ID が一意でない。
- 不正なオブジェクトの型が SimpleReference の場合，アクティブな SimpleReference オブジェクトで，ID が一意でない。

処理を中止します。

対策：

オブジェクトが保持する ID が一意になるように setId() メソッドで変更し，再度実行してください。

KEBR22004-E （F，V）

この {0} オブジェクトは関係を否定するオブジェクトです。よって，このメソッドは使用できません。

{0}：このメソッドのオブジェクトの型

要因：

オブジェクト {0} は関係を否定するオブジェクトのため，このメソッドは使用できません。
処理を中止します。

対策：

アプリケーションの処理を見直し，関係を否定するオブジェクトに対して，このメソッドを使用しないようにしてください。

KEBR22006-E （F，V）

引数 {0} に指定された {1} オブジェクトを所有している SimpleTaxonomy オブジェクトが，この {2} オブジェクトを所有している SimpleTaxonomy オブジェクトと異なります。

{0}：不正なオブジェクトが指定された引数の名前
{1}：不正なオブジェクトの型
{2}：このメソッドのオブジェクトの型

要因：

メソッド呼び出しの対象となったオブジェクトを所有している SimpleTaxonomy オブジェクトと，メソッド呼び出しの引数に指定したオブジェクトを所有している SimpleTaxonomy オブジェクトは，同じでなければなりません。
処理を中止します。

対策：

アプリケーションの処理を見直し，所有者が同じであるオブジェクトを指定して再度実行してください。

KEBR22007-E （F，V）

引数 {0} に指定された {1} オブジェクトを所有している XBRLTaxonomyDocument オブジェクトが，この SimpleTaxonomy オブジェクトを生成したときに指定された XBRLTaxonomyDocument オブジェクトと異なります。

{0}：不正なオブジェクトが指定された引数の名前
{1}：不正なオブジェクトの型

要因：

メソッド呼び出しの対象となったオブジェクトを所有している SimpleTaxonomy オブジェクトの生成時に指定された XBRLTaxonomyDocument オブジェクトと，メ

ソッド呼び出しの引数に指定したオブジェクトを所有している
XBRLTaxonomyDocument オブジェクトは，同じでなければなりません。
処理を中止します。

対策：

アプリケーションの処理を見直し，所有者が同じであるオブジェクトを指定して再度実行してください。

KEBR22008-E （F，V）

引数 {0} に指定された {1} オブジェクトを所有している SimpleTaxonomy オブジェクトが，この SimpleTaxonomy オブジェクトと異なります。

{0}：不正なオブジェクトが指定された引数の名前

{1}：不正なオブジェクトの型

要因：

メソッド呼び出しの対象となったオブジェクトを所有している SimpleTaxonomy オブジェクトと，メソッド呼び出しの引数に指定したオブジェクトを所有している SimpleTaxonomy オブジェクトは，同じでなければなりません。
処理を中止します。

対策：

アプリケーションの処理を見直し，所有者が同じであるオブジェクトを指定して再度実行してください。

KEBR22009-E （F，V）

この SimpleItemDefinition オブジェクトの type 属性が {0} です。よって，このメソッドは使用できません。

{0}: このオブジェクトの現在の type 属性値

要因：

SimpleItemDefinition オブジェクトの setBalance() メソッドは，
SimpleElementDefinition.getType() メソッドで取得される値が
"xbrli:monetaryItemType"（SimpleItemDefinition.getItemType() メソッドで取得される値が SimpleItemDefinition.MONETARY_ITEM）以外の場合には使用できません。
処理を中止します。

対策：

アプリケーションの処理を見直し，アイテム要素宣言を表すオブジェクトの type 属性値（getType()/getItemType() で取得される値）を見直すか，または，このメソッドを使用しないようにしてください。

KEBR22010-E （F，V）

拡張元として指定されたタクソノミー文書の基点となるタクソノミースキーマに，他の XML **文書の一部となっているタクソノミースキーマが存在します。**(**絶対** URI={0})

{0}: 不正なタクソノミー本体の絶対 URI

要因：

SimpleTaxonomyBuilder オブジェクトでは，ほかの XML 文書の一部であるタクソノミー本体を基点としたタクソノミー文書については，拡張元として指定できません。

処理を中止します。

対策：

基点となるタクソノミー本体に，ほかの XML 文書の一部になっているタクソノミー本体を含まないタクソノミー文書を指定して，再実行してください。

KEBR22011-E （F，V）

cbrp.parse.validate.frtafris **オプションが** false **の場合は，**{0}.{1} **メソッドは利用できません。**

{0}: クラス名またはインターフェース名

{1}: コンストラクタ名またはメソッド名

要因：

cbrp.parse.validate.frtafris オプションが true の場合だけ利用可能なメソッドを呼び出しました。しかし，現在の cbrp.parse.validate.frtafris オプションの設定では，そのメソッドは利用可能ではありません。

対策：

cbrp.parse.validate.frtafris オプションが true の場合だけ利用可能なメソッドを使用するときは，cbrp.parse.validate.frtafris オプションを true に指定してから呼び出すようにしてください。

KEBR22100-E （F，V）

操作対象の文書が準拠している XBRL Dimensions **のバージョンが** {0} **の場合は，**{1}.{2} **メソッドは利用できません。**

{0}：XBRL Dimensions のバージョン

{1}：クラス名またはインターフェース名

{2}：コンストラクタ名またはメソッド名

要因：

XBRL Dimensions の特定のバージョンだけで利用できるメソッドを呼び出しました。しかし，操作対象の XBRL 文書が準拠している XBRL Dimensions のバージョンでは，そのメソッドは利用できません。

対策：

XBRL Dimensions の特定のバージョンだけで利用できるメソッドを使用する場合

は，操作対象の XBRL 文書が準拠している XBRL Dimensions のバージョンを確認してから呼び出すように，アプリケーションを修正してください。

KEBR22101-E （F，V）

指定された XBRLTaxonomyDocumentBuilder **オブジェクトが処理対象とする** XBRL Dimensions **のバージョンと，この** XBRLInstanceDocumentBuilder **オブジェクトが処理対象とする** XBRL Dimensions **のバージョンが異なります。**(XBRLInstanceDocumentBuilder **オブジェクトが処理対象とする** XBRL Dimensions **のバージョン** ={0}, XBRLTaxonomyDocumentBuilder **オブジェクトが処理対象とする** XBRL Dimensions **のバージョン** ={1})

{0}：XBRLInstanceDocumentBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョン

{1}：XBRLTaxonomyDocumentBuilder オブジェクトが処理対象とする XBRL Dimensions のバージョン

要因：

XBRLInstanceDocumentBuilder オブジェクトと，XBRLInstanceDocumentBuilder.setTaxonomyDocumentBuilder() メソッドに指定する XBRLTaxonomyDocumentBuilder オブジェクトは，処理対象の XBRL Dimensions のバージョンが同じでなければなりません。

対策：

XBRLInstanceDocumentBuilder.setTaxonomyDocumentBuilder() メソッドに，処理対象の XBRL Dimensions のバージョンが同じである XBRLTaxonomyDocumentBuilder オブジェクトを指定するようにアプリケーションを修正してください。

KEBR22102-E （F，V）

引数 {0} **に指定されたロールは** DRS **にありません。**(**ロール** ={1})

{0}：引数の名前

{1}：DRS にないロール

要因：

指定されたロールが DRS にないため，処理を続行できません。

対策：

DRS にあるロールを指定して再度実行してください。

KEBR22103-E （F，V）

指定された {0} **オブジェクトが準拠する** XBRL Dimensions **のバージョンと，この** {2} **オブジェクトが処理対象とする** XBRL Dimensions **のバージョンが異なります。**({2} **オブジェクトが処理対象とする** XBRL Dimensions **のバージョン** ={3}, {0} **オブジェクトが準拠する** XBRL Dimensions **のバージョン** ={1})

{0}：指定されたオブジェクトの型
{1}：指定されたオブジェクトが準拠する XBRL Dimensions のバージョン
{2}：このオブジェクトの型
{3}：このオブジェクトが処理対象とする XBRL Dimensions のバージョン

要因：
メソッドの引数に指定したオブジェクトが準拠する XBRL Dimensions のバージョンと，このオブジェクトが処理対象とする XBRL Dimensions のバージョンは同じでなければなりません。

対策：
このオブジェクトが処理対象とする XBRL Dimensions のバージョンに準拠するオブジェクトを指定するように，アプリケーションを修正してください。

KEBR22104-E （F，V）

引数 {0} に指定された要素は，このシナリオ要素の子要素ではありません。

{0}：引数の名前

要因：
指定された要素はこのシナリオ要素の子要素でなければなりません。

対策：
このシナリオ要素の子要素を指定して再度実行してください。

KEBR22105-E （F，V）

引数 {0} に指定された要素は，このセグメント要素の子要素ではありません。

{0}：引数の名前

要因：
指定された要素はこのセグメント要素の子要素でなければなりません。

対策：
このセグメント要素の子要素を指定して再度実行してください。

KEBR22106-E （F，V）

指定されたディメンジョンは，タクソノミー文書で定義されていません。(名前空間 URI={0}, 要素名 ={1})

{0}：定義されていないディメンジョンの名前空間 URI
{1}：定義されていないディメンジョンの要素名

要因：
指定されたディメンジョンがタクソノミー文書で定義されていないため，処理を続行できません。

対策：
タクソノミー文書で定義されているディメンジョンを指定して再度実行してくださ

い。

KEBR25000-E （F）

指定された {0} 型データの文字列表現が不正です。

{0}：不正値の型の名前

要因：

不正な文字列表現が指定されました。

対策：

正しい文字列表現を指定して再度実行してください。

KEBR25001-E （F）

引数 {0} に負数は指定できません。指定できる値 = 負数以外

{0}：負数が指定された引数の名前

要因：

負数を指定してはいけない引数に負数が指定されました。

対策：

引数に負数以外の値を指定して再度実行してください。

KEBR25002-E （F）

保持しているバイト配列の長さ以上の引数 {0} は指定できません。指定できる値 = 保持しているバイト配列の長さ未満

{0}：指定された引数の名前

要因：

保持しているバイト配列の長さ以上の開始位置の offset（引数 {0}）が指定されました。

対策：

引数 {0} に保持しているバイト配列の長さ未満の値を指定して再度実行してください。

KEBR25003-E （F）

配列 {0} の長さを超える引数 {1}+{2} は指定できません。指定できる値 = 配列 {0} の長さ以下

{0}：引数で指定した配列の名前

{1}：引数で指定した開始位置（offset）

{2}：引数で指定した長さ（length）

要因：

引数 {0} に指定した配列の長さを超える開始位置の offset（引数 {1}）+ 取得する長さ length(引数 {2}) が指定されました。

対策：

引数 {1}+ 引数 {2} が配列 {0} の長さ以下になるように修正して，再度実行してください。

KEBR25004-E （F）

引数 {0} に {1} の倍数の長さ以外の文字列は指定できません。指定できる値 ={1} の倍数の長さの文字列 {0}

{0}：指定された引数の名前

{1}：整数値 (XBRLHexBinary では 2，XBRLBase64Binary では 4)

要因：

引数 {0} に指定した文字列の長さが {1} の倍数ではありません。

対策：

引数 {0} に，XBRLHexBinary では 2 の倍数，XBRLBase64Binary では 4 の倍数の長さの文字列を指定して再度実行してください。

KEBR25005-E （F）

ISO8859-1 **の文字セットがサポートされていません。**

要因：

標準文字セットである ISO8859-1 の文字セットが JavaVM でサポートされていません。

対策：

ISO8859-1 を含む標準文字セットをサポートした JavaVM を使用して，再度実行してください。

KEBR25006-E （F）

フォーマットが不正です。

要因：

XBRLBase64Binary クラスで，バイト配列を符号化したときに，最後から 4 バイト目または 3 バイト目に埋め文字である「=」が指定されました。または，
XBRLHexBinary クラスで，バイト配列を符号化したときに，1 ～ 9 および A ～ F 以外が指定されました。

対策：

XBRLBase64Binary クラスで，バイト配列を符号化した場合，最後から 4 バイト目または 3 バイト目に埋め文字である「=」がない文字列を指定して再度実行してください。
XBRLHexBinary クラスで，バイト配列を符号化した場合，1 ～ 9 および A ～ F を指定して再度実行してください。

KEBR25007-E （F）

引数 {0} に長さ 0 の引数は指定できません。指定できる値 = 長さ 0 以外の引数 {0}

{0}：指定された引数の名前

要因：

引数 {0} に長さ 0 の引数が指定されました。

対策：

引数 {0} に，長さ 0 以外の引数を指定して実行してください。

KEBR25008-E （F）

指定された名前空間 URI に不正な文字列が含まれています。名前空間 URI={0}

{0}：指定された名前空間

要因：

指定された名前空間 URI に不正な文字列が含まれています。

対策：

正しい名前空間 URI を指定して再度実行してください。

KEBR25009-E （F）

指定されたローカルパートに不正な文字列が含まれています。ローカルパート ={0}

{0}：指定されたローカルパート

要因：

指定されたローカルパートに不正な文字列が含まれています。

対策：

正しいローカルパートを指定して再度実行してください。

KEBR25010-E （F）

引数 {0} に 0 または，負数は指定できません。指定できる値 =0 より大きい値

{0}：指定された引数の名前

要因：

指定された引数に 0 または負数が指定されました。

対策：

引数 {0} に，0 より大きい値を指定して再度実行してください。

KEBR25011-E （F）

引数 {0} に 0 は指定できません。指定できる値 =0 以外の値

{0}：指定された引数の名前

要因：

指定された引数に 0 が指定されました。

対策：

引数 {0} に，0 以外の値を指定して実行してください。

KEBR25013-E （F）

指定された名前空間接頭辞に不正な文字列が含まれています。名前空間接頭辞 ={0}

{0}：指定された名前空間接頭辞

要因：

指定された名前空間接頭辞に不正な文字列が含まれています。

対策：

正しい名前空間接頭辞を指定して再度実行してください。

KEBR25014-E （F）

指定されたスキーマ文書の位置を示す URI **に不正な文字列が含まれています。スキーマ文書の位置を示す** URI={0}

{0}：指定されたスキーマ文書の位置を示す URI

要因：

指定されたスキーマ文書の位置を示す URI に不正な文字列が含まれています。

対策：

スキーマ文書の位置を示す URI を正しく指定して再度実行してください。

7.5 メッセージ KEBR30000-KEBR39999

KEBR30000-KEBR39999 のメッセージの詳細について番号順に説明します。

KEBR30000-I （F）

XBRLDocumentManager **を初期化しました。**

要因：

XBRLDocumentManager クラスの init() メソッド実行に成功しました。

対策：

なし

KEBR39000-E （V）

An attempt to initialize XBRLDocumentManager has failed.

要因：

XBRLDocumentManager クラスの init() メソッド実行に失敗しました。

対策：

システム管理者に連絡してください。

KEBR39001-E （V）

XBRLDocumentManager is not initialized.

要因：

init() メソッドを実行しないで，getInstance() メソッドを実行しました。

対策：

getInstance() メソッド実行前に，init() メソッドを実行しているか，アプリケーションを見直してください。

KEBR39002-E （V）

XBRLDocumentManager has already been initialized.

要因：

init() メソッド実行後に，再度 init() メソッドを実行しました。

対策：

init() メソッド実行後に，再度 init() メソッドを実行していないか，アプリケーションを見直してください。

KEBR39100-E （V）

An attempt to initialize a message has failed.

要因：

初期化に失敗しました。

対策：

システム管理者に連絡してください。

KEBR39101-E （V）

The message is not initialized.

要因：

初期化されていません。

対策：

システム管理者に連絡してください。

KEBR39102-E （V）

The message has already been initialized.

要因：

二重に初期化されました。

対策：

システム管理者に連絡してください。

KEBR39103-E （V）

An attempt to read the cbrp_message.properties has failed.

要因：

cbrp_message.properties（リソースファイル）の読み込みに失敗しました。

対策：

アプリケーションのセキュリティポリシーを確認してください。

KEBR39200-E （V）

Property initialization has failed. ({0}:({1}))

{0}：詳細メッセージ（メソッド名，ファイル名，パラメタ名など）

{1}：保守情報

要因：

初期化に失敗しました。

対策：

システム管理者に連絡してください。

KEBR39201-E （V）

The property is not initialized. ({0}.{1})

{0}：クラス名
{1}：メソッド名

要因：
初期化されていません。

対策：
システム管理者に連絡してください。

KEBR39202-E （V）

The property has already been initialized. ({0}.{1})

{0}：クラス名
{1}：メソッド名

要因：
二重に初期化されました。

対策：
システム管理者に連絡してください。

KEBR39203-E （V）

An attempt to read the defaults.properties file has failed. ({0}):({1})

{0}：メソッド名
{1}：保守情報

要因：
デフォルトプロパティファイルの読み込みに失敗しました。

対策：
システム管理者に連絡してください。

KEBR39204-E （V）

The specified file was not found. (file name = {0}):({1})

{0}：指定したファイル名
{1}：保守情報

要因：
指定したプロパティファイルを見つけられませんでした。

対策：
- 指定したプロパティファイルのパスを確認してください。
- アプリケーションのセキュリティポリシーを確認してください。

KEBR39205-E （V）

A file permission error occurred. ({0}: file name = {1}):({2})

{0}：アクセスに失敗した要因（security，load）

{1}：アクセスに失敗したファイル名

{2}：保守情報

要因：

指定したプロパティファイルへのアクセスに失敗しました。

対策：

- 指定したプロパティファイルのアクセス権を確認してください。
- アプリケーションのセキュリティポリシーを確認してください。

KEBR39206-E （V）

The specified value is outside the valid range. ({0} = {1}:{2})

{0}：指定されたプロパティ名

{1}：プロパティに設定されていた値です。取得できない場合は null になります。

{2}：そのプロパティに指定できる値の範囲です。範囲がない場合は表示しません。

要因：

指定したプロパティのプロパティ値が不正な値です。

対策：

指定したプロパティを確認してください。

KEBR39207-E （V）

The specified parameter is invalid. ({0} = {1})

{0}：指定されたプロパティ名

{1}：指定した値

要因：

指定したパラメタが不正です。

対策：

指定したパラメタを確認してください。

KEBR39300-E （V）

An attempt to initialize a trace has failed.

要因：

トレース機能の初期化に失敗しました。

対策：

- アプリケーションのクラスパスに <System Drive>:¥Program Files¥HITACHI¥HNTRLib2¥classes¥hntrlib2j.jar があるかどうか確認してください。ない場合はアプリケーションのクラスパスに hntrlib2j.jar を追加し，再度実行してください。
  【UNIX の場合】/opt/hitachi/HNTRLib2/classes/hntrlib2j.jar
- アプリケーションにセキュリティマネージャを設定している場合，セキュリティ

ポリシー（ファイルまたはフォルダへのアクセス権限）が不正なおそれがあります。アプリケーションのファイルまたはフォルダへの読み取り／書き込み権限を確認し，再度実行してください。

- <System Drive>:¥Program Files¥HITACHI¥HNTRLib2¥classes¥hntrlib2j.jar がインストールされているかどうか確認してください。インストールされていない場合は，uCosminexus Business Reporting Processor を再インストールして再度実行してください。
  【UNIX の場合】/opt/hitachi/HNTRLib2/classes/hntrlib2j.jar

KEBR39301-E （V）

The trace is not initialized.

要因：

初期化されていません。

対策：

システム管理者に連絡してください。

KEBR39302-E （V）

The trace has already been initialized.

要因：

二重に初期化されました。

対策：

システム管理者に連絡してください。

7.6 メッセージ KEXL00000-KEXL09999

KEXL00000-KEXL09999 のメッセージの詳細について番号順に説明します。

KEXL00001-E

XLinkDocumentSet のクローンの作成に失敗しました。

要因：

アプリケーション側での DOM の更新によって，クローン時にエラーが発生しました。内部エラーです。

対策：

システム管理者に連絡してください。

KEXL00002-E

href 属性に指定した URI に接続できません。

要因：

simple 型要素，locator 型要素の href 属性に指定した URI のサーバが見つかりません。

対策：

href 属性に指定したサーバがあるか確認してください。

KEXL00003-E

同一拡張リンク中に同じ弧を持つ arc 型要素が指定されています。

要因：

同一拡張リンク内に，同じ from と to 属性値を持つ arc 型要素があります。

対策：

同一拡張リンク内にある同じ孤を持つ arc 型要素を修正してください。

KEXL00004-E

href 属性に指定したファイルが見つかりません。

要因：

simple 型要素，locator 型要素の href 属性に指定した URI のサーバは見つかりましたが，ファイルが見つかりません。

対策：

href 属性に指定したファイルがあるか確認してください。

KEXL00005-E

actuate 属性の値に，{onLoad|onRequest|other|none} 以外の値が指定されています。

要因：

actuate 属性の値に，{onLoad | onRequest | other | none} 以外の値が指定されています。

対策：

actuate 属性の値を正しく修正してください。

KEXL00006-E

show 属性の値に，{new|replace|embed|other|none} 以外の値が指定されています。

要因：

show 属性の値に，{new | replace | embed | other | none} 以外の値が指定されています。

対策：

show 属性の値を正しく修正してください。

KEXL00007-E

type 属性の値に，{simple|extended|locator|arc|resource|title|none} 以外の値が指定されています。

要因：

type 属性の値に，{simple | extended | locator | arc | resource | title | none} 以外の値が指定されています。

対策：

type 属性の値を正しく修正してください。

KEXL00008-E

{label|from|to} 属性の値が NCName ではありません。

要因：

{label | from | to} 属性値に NCName で許されない文字（「：」など）が指定されています。

対策：

{label | from | to} 属性の値に NCName を指定してください。

KEXL00009-E

必須項目が指定されていません。locator 型要素に href 属性は必須です。

要因：

locator 型要素に href 属性が指定されていません。

対策：

locator 型要素に href 属性を指定してください。

KEXL00010-E

指定した URI が RFC2396 に違反しています。

要因：

- メソッドのパラメタに RFC2396 に違反した URI が指定されているか，または絶対 URI が指定されていません。
- {role | arcrole | href} 属性値が RFC2396 に違反しています。
- XML Base によって絶対 URI 化をするときに，xml:base 属性が RFC2396 に違反しています。

対策：

RFC2396 に従った URI を指定してください。

KEXL00011-E

{role|arcrole} **属性の値は絶対 URI でないといけません。**

要因：

{role | arcrole} 属性値に絶対 URI ではない値が指定されています。

対策：

{role | arcrole} 属性値に絶対 URI を指定してください。

KEXL00012-E

{from|to} **に指定した要素がみつかりません。**

要因：

arc 型要素の {from | to} に指定したラベル名が同一拡張リンク中にありません。

対策：

arc 型要素の {from | to} 属性の値に，同一拡張リンク中にあるラベルを指定してください。

KEXL00013-E

href **属性に指定した** XPointer **式が評価できません。**

要因：

href 属性に指定した XPointer 式が評価できなく，また該当するロケーションがありません。

対策：

href 属性に評価できる XPointer 式を指定してください。

KEXL00014-E

href **属性に指定した** XPointer **式が不正です。**

要因：

href 属性に指定した XPointer 式が XPointer として不正です。

対策：

href 属性に正しい XPointer 式を指定してください。

7.7 メッセージ KEXL10000-KEXL19999

KEXL10000-KEXL19999 のメッセージの詳細について説明します。

KEXL10001-E

XPointerLocation.getStartPoint メソッドは，ノード型が属性ノードまたは名前空間ノードであるノードロケーションには適用できません。

要因：

XPointerLocation.getStartPoint メソッドを，ノード型が属性ノードまたは名前空間ノードであるノードロケーションに適用しています。内部エラーです。

対策：

システム管理者に連絡してください。

KEXL10002-E

XPointerLocation.getEndPoint メソッドは，ノード型が属性ノードまたは名前空間ノードであるノードロケーションには適用できません。

要因：

XPointerLocation.getEndPoint メソッドを，ノード型が属性ノードまたは名前空間ノードであるノードロケーションに適用しています。内部エラーです。

対策：

システム管理者に連絡してください。

KEXL10003-E

XPointerLocation.rangeTo メソッドまたは range-to 関数で生成するレンジの開始，終了ポイントが，同一の XML 文書の中にありません。

要因：

XPointerLocation.rangeTo メソッドまたは range-to 関数で生成するレンジの開始ポイントと終了ポイントが，同じ XML 文書の中にありません。内部エラーの可能性があります。

対策：

range-to 関数によって生成するレンジの開始ポイントと終了ポイントは，同じ XML 文書の中にあるようにしてください。
解決しない場合は，システム管理者に連絡してください。

KEXL10004-E

XPointerManager.evaluate メソッドに指定された org.w3c.dom.Document が不正です。

要因：

XPointerManager.evaluate メソッドの document パラメタに，正しい org.w3c.dom.Document が指定されていないか，または null が指定されています。内部エラーです。

対策：

システム管理者に連絡してください。

KEXL10005-E

メソッドのパラメタが不正です。

要因：

各メソッドのパラメタが不当です。内部エラーです。

対策：

システム管理者に連絡してください。

KEXL10006-E

XPointer **式の構文エラーです。**

要因：

XPointer 式に文法的なエラーがあります。

対策：

XPointer 式の文法的なエラーを取り除いてください。

KEXL10007-E

XPointerManager.evaluate **メソッドに指定された** org.w3c.dom.Document **が** here **ノード，** origin **ノードのオーナードキュメントと異なります。**

要因：

XPointerManager.evaluate メソッドの here パラメタまたは origin パラメタが有効で，here ノードまたは origin ノードのオーナードキュメントが document パラメタと異なります。内部エラーです。

対策：

システム管理者に連絡してください。

KEXL10008-E

XPointer **式に** here **関数がありますが** here **ノードが指定されていません。**

要因：

XPointerManager.evaluate メソッドで，XPointer 式に here 関数が指定されており，かつ here パラメタに null が指定されています。内部エラーです。

対策：

システム管理者に連絡してください。

KEXL10009-E

XPointer 式に origin 関数がありますが origin ノードが指定されていません。

要因：

XPointerManager.evaluate メソッドで，XPointer 式に origin 関数が指定されており，かつ origin パラメタに null が指定されています。内部エラーです。

対策：

システム管理者に連絡してください。

KEXL10010-E

"" が閉じられていません。

要因：

XPointer 式のダブルクォーテーション「"」が閉じられていません。

対策：

文字列を正しくダブルクォーテーション「"」で囲んでください。

KEXL10011-E

"" が閉じられていません。

要因：

XPointer 式のシングルクォーテーション「'」が閉じられていません。

対策：

文字列を正しくシングルクォーテーション「'」で囲んでください。

KEXL10012-E

括弧の対応が取れていません。

要因：

XPointer 式，または XPointer 式の中の「"」や「'」で囲まれたリテラル内の括弧の対応が取れていません。

対策：

括弧の対応を確認し，正しく対応するように修正してください。またはサーカムフレックスでエスケープしてください。

KEXL10013-E

サーカムフレックスの使用方法に誤りがあります。

要因：

XPointer 式の中で，サーカムフレックスの使用方法に誤りがあます。

対策：

括弧をエスケープする目的以外でサーカムフレックスを記述する場合は，サーカム

フレックスを二つ連続して記述（^^）してください。

KEXL10014-E

{0} 関数に指定できる引数は {1} 個だけです。

{0}：関数名

{1}：指定された引数の数

要因：

関数に指定した引数の数が不当です。

対策：

関数に指定できる引数の個数を確認し，正しい個数の引数を指定してください。

KEXL10015-E

xpointer() スキーマ式の中に何も指定されていません。

要因：

xpointer() スキーマ式の中に何も指定されていません。

対策：

xpointer() スキーマ式の中に適切な内容を指定してください。

KEXL10016-E

{0} は誤った数値の書き方です。

{0}：指定された数値

要因：

数値の書き方に誤りがあります。

対策：

数値の書き方を確認し，正しい書き方で指定してください。

KEXL10017-E

{0} は正しい関数名ではありません。

{0}：指定された関数名

要因：

不当な関数名が指定されています。

対策：

関数名を確認し，正しい関数名を指定してください。

KEXL10018-E

{0} が必要です。{1} は誤りです。

{0}：正しい構文の文字

{1}：指定された文字

要因：

構文的に不当な文字があります。

対策：

xpointer() スキーマ式の内容を確認し，正しい構文で記述してください。

KEXL10019-E

位置パスが必要です。{0} **は誤りです。**

{0}：指定されたパス

要因：

位置パスが必要な文脈で，位置パスが指定されていません。

対策：

位置パスが必要な個所に，正しく位置パスを指定してください。

KEXL10020-E

'/' **または** '//' **の次に位置ステップが必要です。**{0}

{0}：不当な文字

要因：

「/」または「//」の次に位置ステップが必要な文脈で，位置ステップが指定されていません。

対策：

「/」または「//」の次に，正しく位置ステップを指定してください。

KEXL10021-E

ノードテスト NCName:* **または** QName **のどちらかが必要です。**

要因：

ノードテストの書き方に誤りがあります。

対策：

ノードテストの書き方を確認し，正しい書き方で指定してください。

KEXL10022-E

'/' **または** '//' **の次に相対位置ステップが必要です。**{0}

{0}：不当な文字

要因：

「/」または「//」の次に相対位置ステップが必要な文脈で，相対位置ステップが指定されていません。

対策：

「/」または「//」の次に，正しく相対位置ステップを指定してください。

KEXL10023-E

xpointer() **スキーマ式の終わりに余分な文字があります。**{0}

{0}：余分な文字

要因：

xpointer() スキーマ式の終わりに余分な文字があります。

対策：

xpointer() スキーマ式の終わりにある余分な文字を削除してください。

KEXL10024-E

',' **が見つかりましたが後続の引数がありません。**

要因：

関数引数の区切り文字である「,」のあとに，後続の引数が指定されていません。

対策：

関数引数の区切り文字である「,」のあとに，後続の引数を指定してください。

KEXL10025-E

{0} **は正しい軸の名前ではありません。**

{0}：指定された軸の名前

要因：

不当な軸の名前が指定されています。

対策：

軸の名前を確認し，正しい軸の名前を指定してください。

KEXL10026-E

'.[**述部**]' **は正しくない書き方です。**'self::node()[**述部**]' **を使用してください。**

要因：

「.［述部］」という書き方が指定されています。

対策：

「.［述部］」という書き方の代わりに，「self::node()［述部］」を指定してください。

KEXL10027-E

{0} **は正しいノード型の名前ではありません。**

{0}：指定されたノード型の名前

要因：

不当なノード型の名前が指定されています。

対策：

ノード型の名前を確認し，正しいノード型の名前を指定してください。

KEXL10028-E

名前空間接頭辞を名前空間名に解決できません。{0}

{0}：名前空間接頭辞

要因：

名前空間接頭辞に対する名前空間が宣言されていません。

対策：

名前空間接頭辞に対する名前空間を宣言してください。または，名前空間接頭辞に誤りがないか確認してください。

KEXL10029-E

xpointer() **スキーマ式の中に変数を記述することはできません。**

要因：

xpointer() スキーマ式の中に変数が記述されています。

対策：

xpointer() スキーマ式の中の変数を削除してください。

KEXL10030-E

XPointerLocation.rangeTo **メソッドまたは** range-to **関数で生成したレンジの一方のポイントのコンテナノードが要素型，テキスト型，ルート型以外のとき，もう一方のポイントも同一のコンテナノードを持たなければなりません。**

要因：

XPointerLocation.rangeTo メソッドまたは range-to 関数で生成したレンジの一方のポイントのコンテナノードが要素型，テキスト型，ルート型以外であり，もう一方のポイントが同一のコンテナノードを持っていません。内部エラーの可能性があります。

対策：

range-to 関数で生成するレンジの一方のポイントのコンテナノードが要素型，テキスト型，ルート型以外の場合は，もう一方のポイントも同一のコンテナノードを持つようにしてください。

解決しない場合は，システム管理者に連絡してください。

KEXL10031-E

XPointerLocation.rangeTo **メソッドまたは** range-to **関数で生成したレンジの開始ポイントが，終了ポイントよりも文書順で後ろに存ってはなりません。**

要因：

XPointerLocation.rangeTo メソッドまたは range-to 関数で生成したレンジの開始ポイントが，終了ポイントよりも文書順で後ろにあります。内部エラーの可能性があります。

対策：

range-to 関数で生成するレンジの開始ポイントが，終了ポイントよりも文書順で前または等しい位置にあるようにしてください。
解決しない場合は，システム管理者に連絡してください。

KEXL10034-E

属性ノードまたは名前空間ノードの開始ポイントを取得することはできません。

要因：

- start-point 関数の引数が，属性ノードまたは名前空間ノードです。
- range-to 関数評価時の文脈ロケーションが，属性ノードまたは名前空間ノードです。
- XPointerLocation.rangeTo メソッドを適用したロケーションが，属性ノードまたは名前空間ノードです。
- 内部エラーの可能性があります。

対策：

- start-point 関数の引数が属性ノード，名前空間ノードにならないようにしてください。
- range-to 関数評価時の文脈ロケーションが属性ノード，名前空間ノードにならないようにしてください。
- 解決しない場合は，システム管理者に連絡してください。

KEXL10035-E

属性ノードまたは名前空間ノードの終了ポイントを取得することはできません。

要因：

end-point，range-to 関数，XPointerLocation.rangeTo メソッドの引数が属性ノードまたは名前空間ノードです。内部エラーの可能性があります。

対策：

end-point 関数，range-to 関数の引数が属性ノードまたは名前空間ノードにならないようにしてください。
解決しない場合は，システム管理者に連絡してください。

KEXL10036-E

xpointer() **スキーマ式の評価結果がロケーションセット型ではありません。**

要因：

xpointer() スキーマ式の評価結果がロケーションセット型以外です。

対策：

xpointer() スキーマ式の評価結果がロケーションセット型になるようにしてください。

KEXL10037-E

{0} 関数の引数の型はロケーションセット型でなければなりません。

{0}：指定された関数名

要因：

count, local-name, name, namespace-uri, sum, start-point, end-point, range, range-inside, range-to 関数の引数の型がロケーションセット型ではありません。

対策：

それぞれの関数の引数の型がロケーションセット型になるようにしてください。

KEXL10038-E

string-range 関数の第 1 引数の型はロケーションセット型でなければなりません。

要因：

string-range 関数の第 1 引数の型がロケーションセット型ではありません。

対策：

string-range 関数の第 1 引数の型がロケーションセット型になるようにしてください。

KEXL10039-E

XPointerLocation.compareByDocOrder メソッドで，異なる XML 文書のロケーションを比較することはできません。

要因：

XPointerLocation.compareByDocOrder メソッドで，異なる XML 文書のロケーションを比較しています。内部エラーです。

対策：

システム管理者に連絡してください。

KEXL10040-E

XPointerLocation.compareByDocOrder メソッドで，同一要素に属する異なる属性ノードまたは名前空間ノードをコンテナノードとする，二つのポイントを比較することはできません。

要因：

XPointerLocation.compareByDocOrder メソッドで，同一要素に属する異なる属性ノードまたは名前空間ノードをコンテナノードとする，二つのポイントを比較しています。内部エラーです。

対策：

システム管理者に連絡してください。

KEXL10041-E

XPointerLocation.compareByDocOrder メソッドで，属性ノードまたは名前空間ノードと，そのノードをコンテナノードとするポイントを比較することはできません。

要因：

XPointerLocation.compareByDocOrder メソッドで，属性ノードまたは名前空間ノードと，そのノードをコンテナノードとするポイントを比較しています。内部エラーです。

対策：

システム管理者に連絡してください。

KEXL10042-E

比較できないロケーションが同一ロケーションセットに追加されました。

要因：

| 演算子などを用いて，比較できないロケーションが同一ロケーションセットに追加されました。

対策：

比較できないロケーションを同一ロケーションセットに追加しないようにしてください。

KEXL10043-E

',' が見つかりましたが先行する引数がありません。

要因：

関数引数の区切り文字である「,」の前に，先行する引数が指定されていません。

対策：

関数引数の区切り文字である「,」の前に，先行する引数を指定してください。

KEXL10044-E

{0} が必要です。

{0}：構文的に必要な文字

要因：

構文的に必要な文字がありません。

対策：

xpointer() スキーマ式の内容を確認し，正しい構文で記述してください。

KEXL10045-E

文字列，論理値，数値に評価される式をロケーションセットとして使用することはできません。

要因：

文字列，論理値，数値に評価される式がロケーションセットとして使用されています。

対策：

ロケーションパスの内容を確認し，文字列，論理値，数値に評価される式をロケーションセットとして使用しないでください。

KEXL19998-E

内部エラーです。

要因：

内部エラーです。

対策：

システム管理者に連絡してください。

付録

付録 A　XBRL 文書の検証内容と対応するメッセージ ID

付録 B　脚注リンクの再作成

付録 C　バージョン間の差異

付録 D　用語解説

付録 A　XBRL 文書の検証内容と対応するメッセージ ID

ここでは，XBRL 文書の検証内容と対応するメッセージ ID について説明します。

uCosminexus Business Reporting Processor では，大きく分けて，次の内容を検証します。

- XBRL 文書共通の検証
- インスタンス文書の検証
- タクソノミー文書の検証
- リンクベースおよびリンクの検証

次に，ここで説明に使用する表の見方について説明します。

●「検証」欄の記載項目

XBRLInstanceDocumentBuilder または XBRLTaxonomyDocumentBuilder の isValidating() が返す内容によって，検証されるかどうかが次のように異なります。

- 「検証」欄が「あり」の項目
  isValidating() が true を返す場合だけ検証されます。
- 「検証」欄が「－」の項目
  isValidating() が false を返す場合でも常に検証されます。

●「メッセージ」欄の「分類」欄の記載項目

XBRLParseErrorHandler を実装したエラーハンドラが呼ばれるかどうかを示します。

- 「分類」欄が「通知」の項目
  エラーハンドラは呼ばれません。ログにだけメッセージが出力されます。
- 「分類」欄が「通知」以外の項目
  エラーハンドラのメソッドが呼ばれます。

付録 A.1　XBRL 文書共通の検証

XBRL 文書共通の検証では，要素の記述内容，ID の値，URI の文法が正しいか，などを検証します。

表 A-1　XBRL 文書共通の検証

項番	検証内容	検証	メッセージ		XBRL のバージョン	
			分類	メッセージ ID	2.0	2.1
1	XML の構文が誤っています。	―	致命的エラー，エラーまたは警告	KEBR11003-E※ KEBR11004-W KEBR11005-E KEBR11006-E	△	△
2	XML プロセッサを使用して XML スキーマ検証をし，誤りがあれば例外を通知します。	―	エラーまたは警告		△	―
3		あり			―	△
4	各要素の必須属性が記述されていません。または必須の要素が記述されていません。	―	致命的エラーまたはエラー	KEBR10002-E KEBR10003-E	○	○
5	各要素の属性の値に使用できない値が指定されました。	―	致命的エラー	KEBR10025-E	○	○
6	各要素の属性の値に固定値以外が指定されました。	―	致命的エラー	KEBR10000-E	○	○
7	各要素・各属性に指定できないデータ型の値が指定されました（空文字列含む）。	―	致命的エラー	KEBR10036-E KEBR10037-E	○	―
8				KEBR10021-E KEBR10036-E	―	○
9	接頭辞に対応する名前空間 URI の宣言が見つかりません。	―	致命的エラー	KEBR10070-E	―	○
10	XBRL2.0 の場合 インスタンス文書中で指定している id 属性が一意ではありません。 XBRL2.1 の場合 タクソノミー本体中，リンクベース中，またはインスタンス文書中で指定している id 属性が一意ではありません。	―	致命的エラー	KEBR10007-E	○	○
11	名前空間接頭辞が異なる要素，要素名が誤った要素，または認識できない要素が指定されました。	あり	警告	KEBR10004-W	○	―
12	xlink:role 属性または xlink:arcrole 属性が絶対 URI ではありません。	―	致命的エラー	KEBR10041-E	○	○

項番	検証内容	検証	メッセージ		XBRL のバージョン	
			分類	メッセージ ID	2.0	2.1
13	指定されている URI が，RFC2396 に違反しています。	―	致命的エラー	KEBR10030-E KEBR10031-E	○	○
14	xlink:href 属性に指定した XPointer 式の記述に誤りがあります。	―	致命的エラー	KEBR11007-E	―	○

（凡例）○：検証します。　△：XML スキーマ検証時に検証します。　―：該当しません。
注※　XML プロセッサで発生した警告 / エラー内容は，XBRLDOMException の cause に設定されます。

また，XBRL 文書が別文書と関係する場合，単純リンクや参照先文書，カスタムロールやカスタムアークロールが正しいかなどを検証します。

表 A-2　別文書と関係する場合の検証

項番	検証内容	検証	メッセージ		XBRL のバージョン	
			分類	メッセージ ID	2.0	2.1
1	<link:schemaRef> 要素の xlink:href 属性に指定されている XML 文書のルート要素が <xsd:schema> 要素ではありません。	―	致命的エラー	KEBR10056-E KEBR11011-E	―	○
2	<link:linkbaseRef> 要素の xlink:href 属性に指定されている XML 文書のルート要素が <link:linkbase> 要素ではありません。					
3	xlink:href 属性のトラバース先が見つかりません。XML 文書は存在しますが，要素が異なります。	―	致命的エラー	KEBR11010-E	―	○
4				KEBR10103-E	○	○
5	単純リンクの xlink:role 属性に指定されている内容と，リンクベースで記述されているリンクの種類が一致しません。	―	致命的エラー	KEBR10016-E	○	―
6			エラー		―	○
7	<link:linkbaseRef> 要素の xlink:role 属性で，標準値以外のロールが指定されました。	―	致命的エラー	KEBR10001-E	○	―

項番	検証内容	検証	メッセージ		XBRL のバージョン	
			分類	メッセージ ID	2.0	2.1
8		—	警告	KEBR10117-W	—	○
9	<link:roleRef> 要素の roleURI 属性値と，参照先の <link:roleType> 要素の roleURI 属性値が不一致です。	—	致命的エラー	KEBR10066-E	—	○
10	<link:arcroleRef> 要素の arcroleURI 属性値と，参照先の <link:arcroleType> 要素の arcroleURI 属性値が不一致です。					
11	次の要素のリンク先が正しく，かつ xlink:href 属性の参照先に，既知のスキーマ※の名前空間 URI が記述された targetNamespace 属性を持つ XML スキーマ文書が指定されました。 • <link:schemaRef> 要素 • <link:roleRef> 要素 • <link:arcroleRef> 要素 • <link:loc> 要素	—	致命的エラー	KEBR10107-E KEBR11011-E	—	○
12	xlink:href 属性で記述した XPointer 式によって求められる要素が複数あります。またはエレメント要素以外が参照されています。	—	致命的エラー	KEBR10071-E	—	○
13	<xbrli:xbrl> 要素，または <link:linkbase> 要素の下位要素内に，次の要素が複数記述されています。 • 同じ roleURI 属性を持つ <link:roleRef> 要素 • 同じ arcroleURI 属性を持つ <link:arcroleRef> 要素	—	致命的エラー	KEBR10069-E	—	○
14	<xsd:schema> 要素の下位要素内に，次の要素が複数記述されています。 • 同じ roleURI 属性を持つ <link:roleRef> 要素 • 同じ arcroleURI 属性を持つ <link:arcroleRef> 要素	—	致命的エラー	KEBR10069-E	—	○
15	<link:roleType> 要素または <link:arcroleType> 要素の下位要素内に，同じ値を持つ <link:usedOn> 要素が複数記述されています。 なお，名前空間接頭辞が異なっても名前空間 URI が同じ場合，同じ値と判断されます。	あり	エラー	KEBR10048-E	—	○
16	同じロールのリンクベースの URI が一つの XBRL 文書内で複数指定されています。	あり	警告	KEBR10042-W	○	—
17	同じ roleURI 属性を持つ <link:roleType> 要素または同じ arcroleURI 属性を持つ <link:arcroleType> 要素が再定義されています。	—	エラー	KEBR10050-E	—	○

項番	検証内容	検証	メッセージ		XBRL のバージョン	
			分類	メッセージ ID	2.0	2.1
18	同じ roleURI 属性を持つ <link:roleType> 要素または同じ arcroleURI 属性を持つ <link:arcroleType> 要素が再定義されていて，<link:definition> 要素および <link:usedOn> 要素下も同じ値が記述されています。	―	通知	KEBR10049-I	―	○
19	xlink:role 属性または xlink:arcrole 属性の値が標準値または使用できるカスタムロール / カスタムアークロールの値ではありません。	あり	エラー	KEBR10047-E	―	○
20	次の要素の roleURI 属性値または arcroleURI 属性値に，XBRL で決められている標準値を再定義しようとしました。 • <link:roleType> 要素 • <link:arcroleType> 要素 • <link:roleRef> 要素 • <link:arcroleRef> 要素	―	エラー	KEBR10050-E	―	○
21	<link:linkbaseRef> 要素の xlink:arcrole 属性に，XBRL2.0 の場合は推奨値以外，XBRL2.1 の場合は標準値以外が指定されました。	―	致命的エラー	KEBR10001-E	○	―
22				KEBR10046-E	―	○

（凡例）○：検証します。　―：該当しません。

注※

表中の既知のスキーマは，次の種類を示します。

XBRL2.0 の場合

- http://www.xbrl.org/2001/instance
- http://www.xbrl.org/2001/XLink/xbrllinkbase
- http://www.w3.org/2001/XMLSchema-instance
- http://www.w3.org/1999/xlink
- http://www.w3.org/2001/XMLSchema
- http://www.w3.org/XML/1998/namespace

XBRL2.1 の場合

- http://www.xbrl.org/2003/instance
- http://www.xbrl.org/2003/linkbase
- http://www.xbrl.org/2003/XLink
- http://www.w3.org/1999/xlink
- http://www.w3.org/2001/XMLSchema-instance

- http://www.w3.org/2001/XMLSchema
- http://www.w3.org/XML/1998/namespace

付録 A.2　インスタンス文書の検証

インスタンス文書の検証では，アイテムやタプルの指定内容などを検証します。

検証方法には，通常の検証と，完全な妥当性検証の 2 とおりがあります。それぞれの検証内容について次に示します。

(1) 通常の検証の場合

表 A-3　インスタンス文書の検証

項番	検証内容	検証	メッセージ		XBRL のバージョン		XBRL Dimensions のバージョン
			分類	メッセージ ID	2.0	2.1	1.0
1	次の名前空間が宣言されていません。 • http://www.xbrl.org/2001/instance	—	致命的エラー	KEBR10039-E	○	—	—
2	ルート要素に <xbrli:group> 要素または <xbrli:xbrl> 要素以外が指定されました。	—	致命的エラー	KEBR10033-E	○	—	—
3	インスタンス文書に <xbrli:xbrl> 要素が含まれていません。	—	致命的エラー	KEBR10110-E	—	○	○
4	parse() メソッドに渡した要素が <xbrli:xbrl> 要素ではありません。						
5	XBRLInstanceDocumentBuilder#parse() メソッドで複数の <xbrli:xbrl> 要素を持つ XML 文書を解析しました。	あり	警告	KEBR10111-W	—	○	○
6	ルート要素以外に <xbrli:group> 要素の記述があります。	—	通知	KEBR10017-I	○	—	—
7	schemaLocation 属性に，既知のスキーマ※だけ，または空文字列が指定されました。	—	致命的エラー	KEBR10107-E	○	—	—
8	schemaLocation 属性に指定した文書に，タクソノミー文書でないものが含まれています。 次の要素が一つも含まれていない場合は，タクソノミー文書でないと判断されます。 • <link:linkbaseRef> 要素 • エレメント要素 • 各種の拡張リンク	—	警告	KEBR10105-W	○	—	—

項番	検証内容	検証	メッセージ		XBRL のバージョン		XBRL Dimensions のバージョン
			分類	メッセージ ID	2.0	2.1	1.0
9	アイテムと関連づけられていないコンテキスト要素の記述があります。	あり	警告	KEBR10020-W	○	―	―
10		あり	警告	KEBR10004-W	―	○	○
11	アイテムと関連づけられていないユニット要素の記述があります。	あり	警告	KEBR10004-W	―	○	○
12	アイテムに指定されたコンテキストに対応するコンテキスト要素がありません。 または，参照している要素が同じ <xbrli:xbrl> 要素のスコープ内にありません。	―	致命的エラー	KEBR10006-E	○	○	○
13	アイテムに指定されたユニットに対応するユニット要素がありません。 または，参照している要素が同じ <xbrli:xbrl> 要素のスコープ内にありません。	―	致命的エラー	KEBR10006-E	―	○	○
14	<xbrli:period> 要素で期間が指定されていません。 XBRL2.0 の場合，同日の日付は致命的エラーになります。 XBRL2.1 の場合，同日の日付は時間が省略されていると解釈され，正常データとして扱われます。	―	致命的エラー	KEBR10010-E	○	○	○
15	<xbrli:period> 要素の日付・期間の値が誤っています。	―	致命的エラー	KEBR10009-E	○	―	―
16	<xbrli:period> 要素の子要素の関係が正しくありません。	―	致命的エラー	KEBR10035-E	○	○	○
17	要素または <xbrli:scenario> 要素の子要素に，次の名前空間 URI の要素が含まれています。 • http://www.xbrl.org/2003/instance	あり	エラー	KEBR10057-E	―	○	○
18	要素または <xbrli:scenario> 要素の子要素に，アイテムまたはタプルが記述されました。						

項番	検証内容	検証	メッセージ		XBRL のバージョン		XBRL Dimensions のバージョン
			分類	メッセージ ID	2.0	2.1	1.0
19	<xbrli:unit> 要素の子要素である <xbrli:measure> 要素の値の接頭辞が次の名前空間 URI を示す場合，ローカルパートの値には「pure」または「shares」を指定する必要があります。 • http://www.xbrl.org/2003/instance	あり	エラー	KEBR10059-E	—	○	○
20	<xbrli:unit> 要素の子要素である <xbrli:unitNumerator> 要素と，同じ値を持つ <xbrli:unitDenominator> 要素の記述があります。	あり	エラー	KEBR10050-E	—	○	○
21	アイテムの型と <xbrli:measure> 要素の値の関係が不正です。詳細については，この表の下の「アイテムの型と <xbrli:measure> 要素の値との対応」を参照してください。	あり	エラー	KEBR10058-E	—	○	○
22	アイテムまたはタプル要素がタクソノミー文書で定義されていません。	—	エラー	KEBR10038-E	○	○	○
23	アイテムで参照しているコンテキスト要素が利用できません。次の場合が考えられます。 • アイテムで使用するコンテキスト要素の参照はできますが，このコンテキスト要素がアイテムに対応していません。 • numericContext 属性に非数値コンテキストの ID を指定しました。 • nonNumericContext 属性に数値コンテキストの ID を指定しました。 • contextRef 属性の参照が，<xbrli:unit> 要素などの目的と異なる要素を指しています（XBRL2.1 の場合）。	—	致命的エラー	KEBR10103-E	○	○	○
24	インスタンス文書中のアイテムに子要素が記述されました。 XBRL2.1 の場合，xbrli:fractionItemType 型には <xbrli:unitNumerator> 要素と <xbrli:unitDenominator> 要素を記述できます。	—	致命的エラー	KEBR10044-E	○	○	○
25	インスタンス文書で指定できない要素がタクソノミー本体に記述されています。 • abstract 属性が true の要素。 • element 要素の substitutionGroup 属性が xbrli:item または xbrli:tuple 以外の要素。	—	致命的エラー	KEBR10045-E	○	○	○

項番	検証内容	検証	メッセージ		XBRL のバージョン		XBRL Dimensions のバージョン
			分類	メッセージ ID	2.0	2.1	1.0
26	数値型アイテムに，precision 属性および decimals 属性が同時に指定されました。なお，xbrli:fractionItemType 型にはどちらの属性も指定できません。	—	致命的エラー	KEBR10068-E	—	○	○
27	数値型アイテムで xsi:nil="true" のときに，precision 属性または decimals 属性が指定されました。指定された属性は無視されます。	—	エラー	KEBR10072-E	—	○	○
28	nillable="true" の要素宣言に対応するタプルに子要素が指定されました。または，アイテムに値が指定されているのに，要素の属性に xsi:nil="true" が指定されました。指定された子要素および値は無視されます。	—	エラー	KEBR10072-E	—	○	○
29	要素宣言の <xbrli:periodType> 要素の指定と，アイテムから参照しているコンテキスト要素の <xbrli:period> 要素の記述が不一致です。 <xbrli:periodType> 要素が instant または duration の場合に検証されます。それ以外の場合は検証されません。	あり	エラー	KEBR10051-E	—	○	○
30	デフォルト値がインスタンス文書に指定されています。	あり	エラー	KEBR16200-E	—	—	○
31	ハイパーキューブの定義に基づいてアイテムを評価した結果が次の点で不正です。 • all のハイパーキューブが不正です。	あり	エラー	KEBR16201-E	—	—	○
32	ハイパーキューブの定義に基づいてアイテムを評価した結果が次の点で不正です。 • notAll のハイパーキューブが妥当です。	あり	エラー	KEBR16202-E	—	—	○
33	タクソノミー文書ではセグメント要素で定義しているアイテムを，インスタンス文書ではシナリオ要素に記述しているなど，Context Container の値が異なっています。	あり	エラー	KEBR16203-E	—	—	○
34	usable="false" が指定されているアイテムをインスタンス文書に指定しています。 なお，同じ二つのアイテムに対して usable="false" と usable="true" が定義されている場合は，usable="false" と解釈します。	あり	エラー	KEBR16204-E	—	—	○

項番	検証内容	検証	メッセージ		XBRL のバージョン		XBRL Dimensions のバージョン
			分類	メッセージ ID	2.0	2.1	1.0
35	closed="true" が指定されているハイパーキューブで，そのハイパーキューブでは定義されていないディメンジョンをインスタンス文書に指定しています。	あり	エラー	KEBR16205-E	―	―	○
36	DTS に含まれているが，DRS には含まれていないメンバをインスタンス文書に指定しています。	あり	エラー	KEBR16206-E	―	―	○
37	一つのコンテキストに同じディメンジョンが複数指定されています。	あり	エラー	KEBR16207-E	―	―	○
38	<xbrldi:typedMember> 要素の dimension 属性が，Typed Dimension ではありません。	あり	エラー	KEBR16208-E	―	―	○
39	<xbrldi:explicitMember> 要素の dimension 属性が，Explicit Dimension ではありません。	あり	エラー	KEBR16209-E	―	―	○
40	<xbrldi:explicitMember> 要素の内容に指定されている QName が，DTS に存在しません。	あり	エラー	KEBR16210-E	―	―	○
41	cbrp.parse.validate.dimensions.role プロパティに指定されたロールが定義されていません。	あり	警告	KEBR16211-W	―	―	○

（凡例）○：検証します。　―：該当しません。

注※

表中の既知のスキーマは，次の種類を示します。

XBRL2.0 の場合

- http://www.xbrl.org/2001/instance
- http://www.xbrl.org/2001/XLink/xbrllinkbase
- http://www.w3.org/2001/XMLSchema-instance
- http://www.w3.org/1999/xlink
- http://www.w3.org/2001/XMLSchema
- http://www.w3.org/XML/1998/namespace

XBRL2.1 の場合

- http://www.xbrl.org/2003/instance
- http://www.xbrl.org/2003/linkbase

- http://www.xbrl.org/2003/XLink
- http://www.w3.org/1999/xlink
- http://www.w3.org/2001/XMLSchema-instance
- http://www.w3.org/2001/XMLSchema
- http://www.w3.org/XML/1998/namespace

アイテムの型と <xbrli:measure> 要素の値との対応

アイテムの型に応じて，対応する <xbrli:unit> 要素の <xbrli:measure> 要素の値が正しいかが検証されます。なお，アイテムの型は，派生元が同型である場合も含まれます。

アイテムの型と，対応する <xbrli:measure> 要素の値を次の表に示します。

表 A-4　アイテムの型と，対応する <xbrli:measure> 要素の値

項番	アイテムの型	<xbrli:measure> 要素の値	
		名前空間接頭辞	ローカルパート
1	xbrli:monetaryItemType 型	次の名前空間 URI を示す接頭辞を指定します。 • http://www.xbrl.org/2003/iso4217	ISO4217 で定義された値
2	xbrli:sharesItemType 型	次の名前空間 URI を示す接頭辞を指定します。 • http://www.xbrl.org/2003/instance	shares
3	xbrli:pureItemType 型		pure
4	上記以外	検証されません。接頭辞に対応する名前空間 URI が見つからない場合は，致命的エラーになります。	

(2) 完全な妥当性検証の場合

インスタンス文書の完全な妥当性検証で検証される内容は次のとおりです。

表 A-5　インスタンス文書の完全な妥当性検証

項番	検証内容	メッセージ	
		分類	メッセージ ID
1	measure 要素の値（xsd:QName 型）の名前空間が ISO4217 のとき，ローカルパートに対応する値が，単位の一覧で指定された値の中にありません。	エラー	KEBR15201-E
2	タプル要素の属性として，次に示す XBRL2.1 規格の名前空間に属する属性が記述されています。 • http://www.xbrl.org/2003/instance • http://www.xbrl.org/2003/linkbase • http://www.xbrl.org/2003/XLink • http://www.xbrl.org/1999/xlink	エラー	KEBR15211-E

項番	検証内容	メッセージ	
		分類	メッセージ ID
3	requires-element アークに，インスタンス文書に必要な要素が記述されていません。	エラー	KEBR15221-E
4	essence-alias アークを検証した結果，インスタンス文書中のアイテム同士が p-equal かつ c-equal ですが，u-equal ではありません。	エラー	KEBR15231-E
5	essence-alias アークを検証した結果，インスタンス文書中のアイテム同士が p-equal かつ c-equal かつ u-equal ですが，v-equal ではありません。	エラー	KEBR15232-E
6	essence-alias アークを検証した結果，p-equal かつ c-equal である同一名称（名前空間とローカル名が同じ）のアイテムが，複数存在しました。	警告	KEBR15233-W
7	essence-alias アークを検証した結果，二つの拡張リンクの誤りによって，推測したアイテムに矛盾がありました。この場合，PTVI には最初に推測した値が含まれます。	エラー	KEBR15234-E
8	summation-item アークを検証した結果，合計値と検証値が等しくありません。	エラー	KEBR15241-E
9	summation-item アークを検証した結果，p-equal，c-equal かつ u-equal である同一名称（名前空間とローカル名が同じ）のアイテムが，複数存在しました。	警告	KEBR15242-W
10	summation-item アークを検証した結果，p-equal，c-equal かつ u-equal である同一名称（名前空間とローカル名が同じ）のアイテムが，複数存在しました。このため，合計値の検証は行われませんでした。	警告	KEBR15243-W
11	summation-item アークを検証した結果，アイテムの precision 属性の値に 0 が記述されていました。または，decimals 属性から推測した precision 属性の値が 0 以下になっていました。	警告	KEBR15251-W
12	数値アイテムの値が，java.lang.Double.isInfinite() メソッドで true となる 正負の値，または，数値アイテムの値が 0 の近似値であり小数点以下の桁数が 1024 桁を超える値が記述されていました。	警告	KEBR15252-W

付録 A.3　タクソノミー文書の検証

タクソノミー文書の検証では，要素の宣言，要素の記述位置，インポートやインクルードの指定などを検証します。

検証方法には，通常の検証と，完全な妥当性検証の 2 とおりがあります。それぞれの検証内容について次に示します。

（1）通常の検証の場合

表 A-6　タクソノミー文書の検証

項番	検証内容	検証	メッセージ		XBRL のバージョン		XBRL Dimensions のバージョン
			分類	メッセージ ID	2.0	2.1	1.0
1	次の名前空間が宣言されていません。 • http://www.w3.org/2001/XMLSchema	－	致命的エラー	KEBR10039-E	○	－	－
2	parse() メソッドで渡されたスキーマ文書に，タクソノミー文書でないものが含まれています。 次の要素が一つも含まれていない場合は，タクソノミー文書でないと判断されます。 • <link:linkbaseRef> 要素 • エレメント要素 • 各種の拡張リンク	－	警告	KEBR10105-W	○	－	－
3	ルート要素に <xsd:schema> 要素以外が指定されました。	－	致命的エラー	KEBR10033-E	○	－	－
4	<xsd:schema> 要素に targetNamespace が記述されていません。XBRL2.1 では，<xsd:include> 要素以外で読み込まれた文書の場合にこのエラーが発生します。	－	致命的エラー	KEBR10003-E	○	○	○
5	<xsd:import> 要素の schemaLocation 属性で指定されたスキーマ文書がタクソノミー文書ではありません。 次の要素が一つも含まれていない場合は，タクソノミー文書でないと判断されます。 • <link:linkbaseRef> 要素 • エレメント要素 • 各種の拡張リンク	－	警告	KEBR10105-W	○	－	－
6	拡張されたタクソノミー文書が最大階層数を超えました。	－	致命的エラー	KEBR10106-E	○	－	－
7	タクソノミー文書に含めることができる XML 文書の数が，最大数を超えました。	－	致命的エラー	KEBR10113-E	－	○	○
8	タクソノミー本体で記述されている <xsd:import> 要素がループしています。	－	警告	KEBR10043-W	○	－	－

項番	検証内容	検証	メッセージ		XBRL のバージョン		XBRL Dimensions のバージョン
			分類	メッセージ ID	2.0	2.1	1.0
9	タクソノミー本体で記述されている <xsd:import> 要素または <xsd:include> 要素がループしています。	あり	警告	KEBR10052-W	―	○	○
10	タクソノミー本体に，同じ名前空間を持つ同じ要素が複数定義されています。	―	致命的エラー	KEBR10007-E	○	○	○
11	要素の型の定義が同一タクソノミー本体内で定義されていません。	―	警告	KEBR10101-W	○	―	―
12	要素の型の定義が見つかりません。<xsd:import> 要素および <xsd:include> 要素で参照している文書内にもありません。	―	エラー	KEBR10112-E	―	○	○
13	サポートされていない形式でユーザーデータ型を定義しています。	―	警告	KEBR10102-W	○	○	○
14	part 要素の代替要素がタクソノミー本体で定義されていません。	―	エラー	KEBR10040-E	○	―	―
15	要素宣言のタプルに，指定できない xbrli:periodType 属性または xbrli:balance 属性が指定されました。	あり	エラー	KEBR10064-E	―	○	○
16	要素宣言のアイテムの型が xbrli:monetaryItemType 型（派生元が xbrli:monetaryItemType 型も含む）以外の場合に，balance 属性が記述されました。	あり	エラー	KEBR10064-E	―	○	○
17	要素宣言のアイテムで，xbrli:periodType 属性が省略されました。	あり	エラー	KEBR10065-E	―	○	○
18	xbrli:periodType 属性で "instant" および "duration" 以外の文字（空文字列含む）が指定されました。または，xbrli:balance 属性で "debit" および "credit" 以外の文字（空文字列含む）が指定されました。	あり	エラー	KEBR10061-E	―	○	○
19	XPath"schema/annotation/appinfo/link:linkbase/*" 以外に，拡張リンク（xlink:type="extended" の要素）が記述されています。	あり	エラー	KEBR10067-E	―	○	○
20	XPath"schema/annotation/appinfo/*" 以外に，<link:linkbaseRef> 要素（xlink:type="simple" の要素）が記述されています。						

項番	検証内容	検証	メッセージ		XBRL のバージョン		XBRL Dimensions のバージョン
			分類	メッセージ ID	2.0	2.1	1.0
21	XPath"schema/annotation/appinfo/link:linkbase/*" 以外に，<link:roleRef> 要素，<link:arcroleRef> 要素（xlink:type="simple" の要素）が記述されています。						
22	XPath"schema/annotation/appinfo/*" 以外に，<link:roleType> 要素，<link:arcroleType> 要素が記述されています。						
23	<link:arcroleType> 要素の cyclesAllowed 属性で "any"，"undirected" および "none" 以外の文字（空文字列含む）が指定されました。	あり	エラー	KEBR10061-E	—	○	○
24	targetNamespace 属性に空文字列が指定されています。	あり	エラー	KEBR10076-E	—	○	○
25	同じ名前空間 URI かつ同じ name 属性値を持つユーザーデータ型の宣言が，異なる文書中に複数あります。	あり	エラー	KEBR10077-E	—	○	○
26	同じ名前空間 URI かつ同じ name 属性値を持つ代替グループのヘッド要素が，異なる文書中に複数あります。						
27	ハイパーキューブの要素宣言に abstract="true" が指定されていません。	あり	エラー	KEBR16000-E	—	—	○
28	ディメンジョンの要素宣言に abstract="true" が指定されていません。	あり	エラー	KEBR16001-E	—	—	○
29	has-hypercube（all）アークのソースがアイテムの要素宣言ではありません。	あり	エラー	KEBR16002-E	—	—	○
30	has-hypercube（notAll）アークのソースがアイテムではありません。	あり	エラー	KEBR16003-E	—	—	○
31	has-hypercube（all）アークのターゲットがハイパーキューブではありません。	あり	エラー	KEBR16004-E	—	—	○
32	has-hypercube（notAll）アークのターゲットがハイパーキューブではありません。	あり	エラー	KEBR16005-E	—	—	○
33	hypercube-dimension アークのソースがハイパーキューブではありません。	あり	エラー	KEBR16006-E	—	—	○
34	hypercube-dimension アークのターゲットがディメンジョンではありません。	あり	エラー	KEBR16007-E	—	—	○

項番	検証内容	検証	メッセージ		XBRL のバージョン		XBRL Dimensions のバージョン
			分類	メッセージ ID	2.0	2.1	1.0
35	dimension-domain アークのソースが Explicit Dimension ではありません。	あり	エラー	KEBR16008-E	—	—	○
36	dimension-domain アークのターゲットがアイテムではありません。	あり	エラー	KEBR16009-E	—	—	○
37	domain-member アークのソースがアイテムではありません。	あり	エラー	KEBR16010-E	—	—	○
38	domain-member アークのターゲットがアイテムではありません。	あり	エラー	KEBR16011-E	—	—	○
39	dimension-default アークのソースが Explicit Dimension ではありません。	あり	エラー	KEBR16012-E	—	—	○
40	dimension-default アークのターゲットがアイテムではありません。	あり	エラー	KEBR16013-E	—	—	○
41	has-hypercube（all）アークに xbrldt:contextElement 属性がありません。	あり	エラー	KEBR16014-E	—	—	○
42	has-hypercube（notAll）アークに xbrldt:contextElement 属性がありません。	あり	エラー	KEBR16015-E	—	—	○
43	xbrldt:targetRole 属性の値である URI が，roleRef で解決できません。	あり	エラー	KEBR16016-E	—	—	○
44	xbrldt:typedDomainRef 属性が指しているスキーマが，DTS に含まれていません。	あり	エラー	KEBR16017-E	—	—	○
45	xbrldt:typedDomainRef 属性が，ディメンジョンの要素宣言以外の場所に記載されています。	あり	エラー	KEBR16018-E	—	—	○
46	xbrldt:typedDomainRef 属性の値に次のどれかの不正が存在します。 • 指しているスキーマが存在しない • XML Schema のグローバル要素宣言がない • 抽象要素になっている。	あり	エラー	KEBR16019-E	—	—	○
47	xbrldt:typedDomainRef 属性の URI に次のどれかの不正が存在します。 • 不正な URI である • fragment identifier がない	あり	エラー	KEBR16020-E	—	—	○
48	DRS に有向循環があります。	—	エラー	KEBR16021-E	—	—	○

項番	検証内容	検証	メッセージ		XBRL のバージョン		XBRL Dimensions のバージョン
			分類	メッセージ ID	2.0	2.1	1.0
49	アイテムが自身のドメイン・メンバにもなっており，循環があります。	―	エラー	KEBR16022-E	―	―	○
50	一つのディメンジョンに二つ以上のデフォルト・メンバがあります。	あり	エラー	KEBR16023-E	―	―	○

（凡例）○：検証します。　―：該当しません。

(2) 完全な妥当性検証の場合

タクソノミー文書の完全な妥当性検証で検証される内容は次のとおりです。

表 A-7　タクソノミー文書の完全な妥当性検証

項番	検証内容	メッセージ	
		分類	メッセージ ID
1	タプルが，混合内容または単純な内容を持てるように定義されています。	エラー	KEBR15001-E
2	タプルの子孫要素が anonymous 型（匿名）宣言を含んでいます。		
3	タプルに含められる要素の宣言が局所定義されています。		
4	タプルに含められる要素の宣言が参照定義になっていますが，その参照先が xbrli:item または xbrli:tuple から派生した要素のグローバル要素ではありません。		
5	xbrli:fractionItemType 型以外のユーザーデータ型定義の <xsd:element> 要素で，その子要素として <xsd:simpleContent> または <xsd:annotation> 要素以外が宣言されています。	エラー	KEBR15002-E

付録 A.4　リンクベースおよびリンクの検証

リンクベースおよびリンクの検証では，拡張リンクやリンクの上書きを検証します。

検証方法には，通常の検証と，完全な妥当性検証の 2 とおりがあります。それぞれの検証内容について次に示します。

（1）通常の検証の場合

表 A-8　リンクベースおよびリンクの検証

項番	検証内容	検証	メッセージ		XBRL のバージョン	
			分類	メッセージ ID	2.0	2.1
1	ルート要素に <link:linkbase> 要素以外が指定されました。	―	致命的エラー	KEBR10033-E	○	―
2	次の名前空間が宣言されていません。 • http://www.xbrl.org/2001/XLink/xbrllinkbase	―	致命的エラー	KEBR10039-E	○	―
3	同一拡張リンク中に同じ弧を持つアーク型要素が記述されています。	―	致命的エラー	KEBR10032-E	○	○
4	同一ファイル内で，リンクが双方向で記述されていません。	―	警告	KEBR10012-W	○	―
5	双方向のリンクの記述が一致していません。	―	致命的エラー	KEBR10013-E	○	―
6	アーク型要素に対応するロケータ型要素またはリソース型要素が見つかりません。	―	致命的エラー	KEBR10018-E	○	○
7	参照先が利用できません。参照先のファイルはありますが，ID に対応する要素が見つかりません。	―	致命的エラー	KEBR10103-E	○	○
8	リソース型要素を指す <link:loc> 要素の記述がありますが，その <link:loc> 要素を指すアーク型要素の use 属性の値が "prohibited" ではありません。	―	エラー	KEBR10116-E	―	○
9	拡張リンクの <link:loc> 要素が，アイテムまたはタプルを指していません。	―	警告	KEBR10108-W	○	―
10	拡張リンクの <link:loc> 要素が，アーク型要素の xlink:role 属性の値と対応した要素を指していません。	―	致命的エラー	KEBR10071-E	―	○
11	拡張リンクのリンク内でタクソノミー本体の要素を指す場合に，参照先の要素が abstract="true" で記述されています。	―	警告	KEBR10109-W	○	―
12	アーク型要素の xlink:from 属性，および xlink:to 属性に指定されたラベル名に対応する要素が誤っています。	―	致命的エラー	KEBR10019-E	○	○
13	アーク型要素からつながらないリソース型要素，またはロケータ型要素があります。	あり	警告	KEBR10004-W	○	○
14	アーク型要素の weight 属性，order 属性に数値以外の値が指定されました。または，範囲に誤りがあります。	―	致命的エラー	KEBR10021-E	○	○
15	<link:calculationArc> 要素の weight 属性に 0 が指定されました。	あり	エラー	KEBR10053-E	―	○

項番	検証内容	検証	メッセージ		XBRL のバージョン	
			分類	メッセージ ID	2.0	2.1
16	<link:calculationArc> 要素の weight 属性と，ロケータ型要素が参照しているアイテムの xbrli:balance 属性の値が不一致です。					
17	<link:definitionArc> 要素のアークロール属性が "essence-alias" の場合，xlink:from 側および xlink:to 側の type 属性が不一致です。	あり	エラー	KEBR10054-E	―	○
18	アーク型要素の use 属性が prohibited の場合に，上書きするリンクがありません。	あり	警告	KEBR10014-W	○	○
19	同じ priority 属性を持つ同じリンクが複数記述されています。	―	エラー	KEBR10024-E	○	―
20	同じ priority 属性を持つ同じリンクが複数記述されていて，どのリンクにも use="prohibited" が指定されていません。	―	警告	KEBR10074-W	―	○
21	同じ xlink:role 属性で同じ xml:lang 属性を持つラベル要素が複数あります。	―	致命的エラー	KEBR10029-E	○	―
22		あり	警告	KEBR11008-W	―	○
23	インスタンス文書以外で脚注リンクの記述があります。	あり	警告	KEBR10004-W	○	○
24	拡張リンク内に，指定できないリソース型要素があります。	あり	エラー	KEBR10062-E	―	○

（凡例）○：検証します。　―：該当しません。

(2) 完全な妥当性検証の場合

リンクベースの完全な妥当性検証で検証される内容は次のとおりです。

表 A-9　リンクベースの完全な妥当性検証

項番	検証内容	メッセージ	
		分類	メッセージ ID
1	関係ネットワークのカスタムアークロールに対応する link:arcroleType 要素が見つかりません。	エラー	KEBR15101-E
2	関係ネットワークのカスタムアークロールに対応する link:arcroleType 要素が複数存在し，cyclesAllowed 属性の値が不一致です。	エラー	KEBR15102-E
3	関係ネットワークに，許可されない循環がありました。	エラー	KEBR15104-E

項番	検証内容	メッセージ	
		分類	メッセージ ID
4	関係ネットワークに，許可されない複数の経路がありました。	エラー	KEBR15105-E
5	essence-alias アークの関係ネットワークに含まれる要素宣言のデータ型が不一致です。データ型の派生元が同一であっても検出されます。	エラー	KEBR15111-E
6	essence-alias アークの関係ネットワークに含まれる要素宣言の periodType 属性値または balance 属性値が不一致です。	エラー	KEBR15112-E
7	summation-item アークの関係ネットワークに含まれる要素宣言のデータ型が不一致です。ただし，データ型の派生元が同一ならば検出されません。	エラー	KEBR15121-E
8	summation-item アークの関係ネットワークに含まれる要素宣言が数値型ではありません。	エラー	KEBR15122-E

付録 B　脚注リンクの再作成

既存のインスタンス文書を編集する場合，すでに脚注リンクがあるときは，既存の脚注リンクが再作成されます。

ここでは，再作成される脚注リンクの各要素の詳細について説明します。

付録 B.1　再作成される脚注リンク（XBRL2.0 の場合）

脚注リンクでは，次の要素を使用しています。

- 脚注リンクのコンテナ要素である <link:footnoteLink> 要素
- 脚注を表す <link:footnote> 要素
- アイテムやタプルを参照する <link:loc> 要素
- アイテムやタプルと脚注をつなぐ <link:footnoteArc> 要素

<link:footnoteLink> 要素は，脚注全体を表し，子要素として <link:footnote> 要素，<link:loc> 要素，および <link:footnoteArc> 要素を持ちます。

（1） <link:footnoteLink> 要素

既存の <link:footnoteLink> 要素を削除して，新たな <link:footnoteLink> 要素を一つ作成します。作成した <link:footnoteLink> 要素には，既存の <link:footnote> 要素および子要素のノードを移動し，<link:footnoteArc> 要素および <link:loc> 要素は関連づけに沿って再作成します。

新しく作成する <link:footnoteLink> 要素の各属性を次の表に示します。

表 B-1　新しく作成する <link:footnoteLink> 要素の各属性

項番	属性名	内容
1	xlink:type	extended 固定。
2	その他	生成しません。

再編成されたあとに，新しく脚注を追加した場合は，すでに作成されている <link:footnoteLink> 要素下に脚注が追加されます。

（2） <link:footnote> 要素

<link:footnote> 要素および子要素のノードは再利用されます。これによって，親ノードは変更されますが，<link:footnote> 要素の子孫ノードに変更はありません。ただし，<link:footnote> 要素の xlink:label 属性値は，自動採番した ID となります。

変更する <link:footnote> 要素の各属性を次の表に示します。

表 B-2　変更する <link:footnote> 要素の各属性

項番	属性名	内容
1	xlink:label	自動採番した ID になります。
2	その他	変更および追加しません。

（3） <link:loc> 要素

既存の <link:loc> 要素は削除して再作成します。一つのアイテムまたはタプル要素に，<link:loc> 要素を一つだけ再作成します。再作成時に設定する属性は，xlink:href 属性，xlink:label 属性，xlink:type 属性だけです。新しく作成する <link:loc> 要素の各属性を次の表に示します。

表 B-3　新しく作成する <link:loc> 要素の各属性

項番	属性名	内容
1	xlink:type	locator 固定。
2	xlink:href	すでに対応するアイテムまたはタプル要素に id 属性がある場合は，「#」＋ id 属性値になります。ただし，既存 id 属性値がない場合，または重複した場合は，「#」＋自動採番した ID になります。
3	xlink:label	自動採番した ID になります。
4	その他	生成しません。

（4） <link:footnoteArc> 要素

既存の <link:footnoteArc> 要素を削除して再作成します。<link:footnoteArc> 要素は，一つのリンクを双方向で表現するため，必ず二つ作成します。

作成する二つの <link:footnoteArc> 要素を次に示します。

- <link:footnote> 要素から <link:loc> 要素への <link:footnoteArc> 要素
- <link:loc> 要素から <link:footnote> 要素への <link:footnoteArc> 要素

新しく作成する，<link:footnote> 要素から <link:loc> 要素への <link:footnoteArc> 要素を次の表に示します。

表 B-4　<link:footnote> 要素から <link:loc> 要素への <link:footnoteArc> 要素

項番	属性名	内容
1	xlink:type	arc 固定。
2	xlink:show	replace 固定。
3	xlink:actuate	onRequest 固定。
4	xlink:from	<link:footnote> 要素の xlink:label 属性値。

項番	属性名	内容
5	xlink:to	<link:loc> 要素の xlink:label 属性値。
6	xlink:arcrole	http://www.xbrl.org/linkprops/arc/footnote-fact 固定。
7	その他	生成しません。

新しく作成する，<link:loc> 要素から <link:footnote> 要素への <link:footnoteArc> 要素を，次の表に示します。

表 B-5　<link:loc> 要素から <link:footnote> 要素への <link:footnoteArc> 要素

項番	属性名	内容
1	xlink:type	arc 固定。
2	xlink:show	replace 固定。
3	xlink:actuate	onRequest 固定。
4	xlink:from	<link:loc> 要素の xlink:label 属性値。
5	xlink:to	<link:footnote> 要素の xlink:label 属性値。
6	xlink:arcrole	http://www.xbrl.org/linkprops/arc/fact-footnote 固定。
7	その他	生成しません。

注　複数の <link:footnote> 要素または <link:loc> 要素の xlink:label 属性に同じ値を指定して，一つの <link:footnoteArc> 要素で二つ以上のリンクを構成することもできますが，uCosminexus Business Reporting Processor では，このような構成にはしません。

付録 B.2　再作成される脚注リンク（XBRL2.1 の場合）

脚注リンクでは，次の要素を使用しています。

- 脚注リンクのコンテナ要素である <link:footnoteLink> 要素
- 脚注を表す <link:footnote> 要素
- アイテムやタプルを参照する <link:loc> 要素
- アイテムやタプルと脚注をつなぐ <link:footnoteArc> 要素

<link:footnoteLink> 要素は，脚注全体を表し，子要素として <link:footnote> 要素，<link:loc> 要素，および <link:footnoteArc> 要素を持ちます。

(1)　<link:footnoteLink> 要素

既存の <link:footnoteLink> 要素を削除して，新たな <link:footnoteLink> 要素を xlink:role 属性ごとに作成します。同じ xlink:role 属性を持つ <link:footnoteLink> 要素は一つにまとめられます。

作成した <link:footnoteLink> 要素には，既存の <link:footnote> 要素および子要素のノードを移動し，<link:footnoteArc> 要素および <link:loc> 要素は関連づけに沿って再作成します。

新しく作成する <link:footnoteLink> 要素の各属性を次の表に示します。

表 B-6　新しく作成する <link:footnoteLink> 要素の各属性

項番	属性名	内容
1	xlink:type	extended 固定。
2	xlink:role	記述されていた値。
3	その他	生成しません。

再編成されたあとに，同じ xlink:role 属性を持つ脚注を新しく追加した場合は，すでに作成されている <link:footnoteLink> 要素下に脚注が追加されます。

異なる xlink:role 属性を持つ脚注を追加した場合は，<link:footnoteLink> 要素を新しく作成し，作成された <link:footnoteLink> 要素の下に脚注が追加されます。

（2） <link:footnote> 要素

<link:footnote> 要素および子要素のノードは再利用されます。これによって，親ノードは変更されますが，<link:footnote> 要素の子孫ノードに変更はありません。ただし，<link:footnote> 要素の xlink:label 属性値は，自動採番した ID となります。

変更する <link:footnote> 要素の各属性を次の表に示します。

表 B-7　変更する <link:footnote> 要素の各属性

項番	属性名	内容
1	xlink:label	自動採番した ID になります。
2	id	基本的には変更されません。すでに id 属性が設定されていた場合でも，追加，変更されません。
3	その他	変更および追加しません。

（3） <link:loc> 要素

既存の <link:loc> 要素は削除して再作成します。<link:loc> 要素は，<link:footnoteArc> 要素の xlink:arcrole 属性に対応する数だけ再作成されます。一つの拡張リンク内でも，一つのアイテムまたはタプル要素に対して，<link:loc> 要素が一つとはかぎりません。

再作成時に設定する属性は，xlink:href 属性，xlink:label 属性，xlink:type 属性だけです。新しく作成する <link:loc> 要素の各属性を次の表に示します。

表 B-8　新しく作成する <link:loc> 要素の各属性

項番	属性名	内容
1	xlink:type	locator 固定。
2	xlink:href	すでに対応するアイテムまたはタプル要素に id 属性がある場合は，「#」＋ id 属性値になります。ただし，既存 id 属性値がない場合，または重複した場合は，「#」＋自動採番した ID に変更されます。 また，XPointer の element スキームで記述されていた場合，ショートハンドポインタに変更されます。
3	xlink:label	自動採番した ID になります。
4	その他	生成しません。

（4） <link:footnoteArc> 要素

既存の <link:footnoteArc> 要素を削除して再作成します。一つの <link:footnoteArc> 要素で複数の関係を表していた場合，関係ごとに一つのリンクを作成します。

新しく作成する <link:footnoteArc> 要素の各属性を次の表に示します。

表 B-9　新しく作成する <link:footnoteArc> 要素の各属性

項番	属性名	内容
1	xlink:type	arc 固定。
2	xlink:from	<link:loc> 要素の xlink:label 属性値。
3	xlink:to	<link:footnote> 要素の xlink:label 属性値。
4	xlink:arcrole	記述されていた値。
5	order	order 属性値が記述されていた場合，記述されていた値で作成されます。記述されていない場合には，デフォルト値 "1" となります。
6	その他	生成しません。

注　複数の <link:footnote> 要素または <link:loc> 要素の xlink:label 属性に同じ値を指定して，一つの <link:footnoteArc> 要素で二つ以上のリンクを構成することもできますが，uCosminexus Business Reporting Processor では，このような構成にはしません。

付録 C　バージョン間の差異

uCosminexus Business Reporting Processor の以前のバージョンとの動作の差異について説明します。動作の差異を次の表に示します。

04-00 と 05-00 の差異

表 C-1　uCosminexus Business Reporting Processor の動作の差異（04-00 と 05-00 の差異）

項番	動作の差異
1	05-00 では，Generics に対応しました。そのため，「6.　API リファレンス」の各メソッドの形式には，Generics に対応したメソッドの戻り値が記載されています。04-00 の場合，Generics に対応していないため戻り値を記載したメソッドの形式は使用できません。

03-00 と 04-00 の差異

表 C-2　uCosminexus Business Reporting Processor の動作の差異（03-00 と 04-00 の差異）

項番	動作の差異
1	インスタンス文書に float 型や double 型の要素，属性があり，値として +NaN，-NaN，Infinity，+Infinity，-Infinity 等を指定している場合，インスタンス文書の読み込みで，03-00 ではエラーは発生しませんでしたが，04-00 ではエラー (KEBR11005-E, KEBR10036-E) が発生します。
2	インスタンス文書等の読み込み時に対象となるスキーマが見つからない場合，03-00 では致命的エラー (KEBR11002-E) のみが発生しますが，04-00 では警告 (KEBR11004-W) と致命的エラー (KEBR11002-E) が発生します。
3	インスタンス文書に anyURI 型の要素，属性があり，値として "|" 等の文字 (XML Schema Part2 仕様書 3.2.17 anyURI にて定められているエスケープ文字) が含まれていた場合，03-00 では，KEBR11005-E(詳細情報 =KECX06036-E，KECX06020-E に起因する) のエラーが発生しますが，04-00 では発生しません。 ただし，03-00 で上記エラーを無視して続行した場合に致命的エラー (KEBR10030-E) となることは変わりません。
4	XBRLInstanceDocumentWriter インターフェース等の write メソッドでファイル出力する場合，03-00 では，XML 宣言の直後に改行が出力されますが，04-00 では出力されません。

付録 D　用語解説

（英字）

cwa

閉世界仮説（Closed World Assumption）の略で，論理学やデータベース理論で使用される用語です。インスタンス文書では，cwa が真（cwa=true）であれば，任意の事実を計算するために必要な事実（アイテム）が，すべてそのインスタンス文書に含まれていることを意味します。

DTS

Discoverable Taxonomy Set の略です。
複数のタクソノミー文書を，論理的に一つのタクソノミー文書としてみなす範囲のことです。

JAXP

Java API for XML Processing の略です。米国 Sun Microsystems, Inc. が提供する Java 言語用の標準 XML API です。DOM，SAX，および XSLT のリファレンス実装そのものを JAXP と呼ぶ場合もあります。
JAXP には，DOM パーサー，SAX パーサーを生成する API が含まれています。パーサーの実装やコードに依存しないので，プラットフォーム間で共通のプログラムを作成できます。また，JAXP には，XSLT トランスフォーマーの API が含まれているので，Java プログラムから XSLT トランスフォーマーを使用できます。

W3C DOM

XML データのツリー構造を操作するための API として，W3C によって規格化されたドキュメントオブジェクトモデル（Document Object Model）です。このマニュアルでは，XBRL DOM と区別するため，W3C DOM と表現します。

XBRL DOM

uCosminexus Business Reporting Processor が提供する，XBRL 文書を操作するためのオブジェクトモデルのことです。

XBRL 文書

インスタンス文書とタクソノミー文書の両方を指す場合の表現です。XBRL のバージョンを明記する必要がない場合の，文書の総称として使用します。

XLink プロセッサ

XML 文書に記述されている XLink 属性値から，XLink 構造を操作するためのオブジェクト構造を作成するソフトウェアのことです。uCosminexus Business Reporting Processor は，XLink プロセッサとして，uCosminexus XML Link Processor が必要です。

XML プロセッサ

XML 文書を W3C の DOM（Document Object Model）に展開するソフトウェアのことです。
uCosminexus Business Reporting Processor は，XML プロセッサとして，Processing Kit for XML または uCosminexus Application Server が必要です。

XSLT プロセッサ

XSLT（eXtensible Stylesheet Language - Transformations）は，XML 文書を別の形式に変換するための W3C の規格です。XSLT プロセッサとは，XSLT で記述された変換ルールに従って，XML 文書などを別の形式に変換するソフトウェアを指します。

（ア行）

アイテム（item）

所定のビジネス実体に関して，所定の期間内に報告された事実を表します。XBRL では抽象的な XML 要素である <xbrli:item> に対応します。

アクティブ

ノードがインスタンス文書中にある場合，アクティブなノードといいます。アクティブなノードは，XBRLInstanceDocument オブジェクトからオブジェクトツリーをたどることによって到達できます。一方，アクティブでないノードは，インスタンス文書に存在しないノードです。

インスタンス（instance）

ビジネスでの事実を指します。財務報告では財務事実に相当し，基本的には報告された金額のことです。

インスタンス文書（instance document）

ビジネスでの事実を報告する XML 文書です。各要素は，タクソノミー文書（XML スキーマ文書）に定義され，この要素と事実（金額など）を含めた文書を指します。

エンティティ（entity）

アイテムの主題であるビジネス実体です。XBRL では，インスタンス文書での XML 要素である <xbrli:entity> に対応します。XML や SGML の構文規則上の entity を意味するときには，その旨を示します。

（カ行）

拡張リンク（extended link）

一つのリソースから任意の数のリソースに付けたリンクです。

協定世界時

協定世界時（UTC）は，グリニッジ標準時を基に，国際協定によって人工的に維持されている世界共通の標準時です。日本標準時（JST）は，協定世界時より 9 時間進んでいます。

コンテキスト（context）

インスタンス文書で報告されている内容を理解するために必要な項目の集まりです。
文書を発行した企業の説明（エンティティまたは entity），どの時点での報告内容なのかを示す時期（period），報告内容が予算なのか実績なのかなどを示す文書の状態（scenario）などを記載します。
XBRL2.0 の場合，コンテキスト要素には，数値コンテキスト（numericContext）要素，非数値コンテキスト（nonNumericContext）要素があります。XBRL では，インスタンス文書での XML 要

素である <xbrli:numericContext> 要素と，<xbrli:nonNumericContext> 要素に対応します。
XBRL2.1 の場合，数値または非数値の区別はなく，<xbrli:context> 要素に対応します。

（サ行）

時期（period）

ある時点または期間を示します。ビジネス報告では，財務数値およびほかの事実は，ある年月日付の時点，または期間に関して報告されます。XBRL では，インスタンス文書での XML 要素である <xbrli:period> に対応します。

事実

タクソノミー文書で <xsd:element> 要素で定義される要素，およびインスタンス文書中に記述されるその要素の内容を指します。

（タ行）

タクソノミー（taxonomy）

ビジネスでの概念を指します。財務報告は，「財務概念（タクソノミー）」と「財務事実（インスタンス）」が報告される内容であり，財務概念は「貸借対照表」「損益計算書」などの概念です。財務事実は，基本的には金額ですが，その意味する数値を正しく理解するためには，財務概念が必須になります。

タクソノミー文書（taxonomy document）

新しい要素（概念）を定義する XML スキーマ文書です。各要素は，インスタンス文書およびタクソノミー文書中で参照できる概念に対応します。タクソノミー文書は，XLink に準拠した情報を含んだ XML スキーマ文書の拡張とみなせます。実際には，各要素を定義した XML スキーマ文書（.xsd）と，XLink を利用して各要素の関係を表現した XML 文書（.xml）から構成されることが多くなります。

タクソノミー本体

タクソノミー文書を構成する XML 文書のうち，XML スキーマ言語によって要素の一覧を定義している XML スキーマ文書を指します。

タプル（tuple）

ある事実（アイテム）が，理解されるために必須となるアイテムの集まりを意味する用語です。
XBRL では抽象的な XML 要素である <xbrli:tuple> に対応し，タプル要素はアイテムをまとめるために使用されるコンテナです。

トラバース（トラバーサル）

XLink の用語で，任意の目的でリンクを使用することや，それをたどることをトラバーサル（traversal）と呼びます。

（ナ行）

ノード

インスタンス文書またはタクソノミー文書の構成要素を表すオブジェクトを指します。XBRL DOM のオブジェクトモデルでは，XBRLInstanceNode インターフェースまたは XBRLTaxonomyNode インターフェースを継承する各種インターフェースを実装したオブジェクトによって，ノードを表現します。なお，文脈によっては，W3C DOM の org.w3c.dom.Node オブジェクトを指す場合もあります。

（ハ行）

ポイント（point）

XML 構成要素がある位置を表します。ノードとノードの間や文字と文字の間にある一点を指します。

（ラ行）

リンク

XLink の仕様に従って記述されたリンクを指します。

リンクベース（linkbase）

タクソノミー文書を構成する XML 文書のうち，各要素の関係を，XLink を利用して表現した XML 文書を指します。XLink の仕様では，インバウンド（リモートからローカル）とサードパーティー（リモートからリモート）リンクのコレクションを含む文書のことを，リンクデータベース，またはリンクベース（linkbase）と呼びます。

レンジ（range）

二つのポイントの間にある XML 構成要素を表します。

ロケーション（location）

XPointer で参照できるノード，ポイント，レンジの総称を指します。

索引

数字

16 進数 885

A

API 一覧 291

B

Base64 843

C

cbrp.cache.enable 204
cbrp.cache.maxTaxonomyNum 204
cbrp.cache.registerErrorDocument 205
cbrp.parse.delayLoad 202
cbrp.parse.DimensionsVersion 208
cbrp.parse.DTS.externalSchemaRefs 203
cbrp.parse.DTS.ignoringSchemas 202
cbrp.parse.DTS.maxDocuments 203
cbrp.parse.maxTaxonomyExtension 202
cbrp.parse.validate 201
cbrp.parse.validate.dimensions.role 208
cbrp.parse.validate.frtafris 207
cbrp.parse.validate.full-checking 205
cbrp.parse.validate.full-checking.essence-alias.infer 207
cbrp.parse.validate.full-checking.ignoringZeroPrecision 207
cbrp.parse.validate.full-checking.iso4217 206
cbrp.parse.validate.full-checking.PTVI.rounded-item-value 206
cbrp.parse.validate.full-checking.PTVx.location 206
cbrp.parse.validate.full-checking.role 205
cbrp.parse.validate.search-schemaRefs-beforehand 208
cbrp.parse.XBRLVersion 201
cbrp.trace.fileName 200
cbrp.trace.level 200
cbrp.trace.maxFileSegment 201
cbrp.trace.maxFileSize 201
com.hitachi.xbrl.instance パッケージ一覧 292
com.hitachi.xbrl.link パッケージ一覧 311
com.hitachi.xbrl.simple.taxonomy パッケージ一覧 333
com.hitachi.xbrl.taxonomy パッケージ一覧 314
com.hitachi.xbrl.types パッケージ一覧 324
com.hitachi.xbrl.util パッケージ一覧 331
com.hitachi.xbrl パッケージ一覧 291
cwa〔用語解説〕 1176

D

date 849
dateTime 854
dateUnion 858
DTS 6
　拡張読み込み 57
　読み込み 44
　読み込み範囲と指定方法〔XBRL2.0 の場合〕 44
　読み込み範囲と指定方法〔XBRL2.1 の場合〕 48
DTS〔用語解説〕 1176
duration 866

F

FRIS 17
FRTA 17

G

gDay 873
gMonth 875
gMonthDay 877
gYear 880
gYearMonth 882

H

hexBinary 885

I

ID の管理 95
ID の採番 97
ISO4217〔XBRL2.0 の場合〕 232
ISO4217〔XBRL2.1 の場合〕 263

J

JavaVM のカスタマイズ 212
JAXP〔用語解説〕 1176

L

link:label 要素 719

P

PTVI 77
PTVInstance クラス 928
PTVLI 74
PTVLinkbase クラス 929

Q

QName 893

S

SimpleConceptDefinition インターフェース 937
SimpleEdge インターフェース 938
SimpleElementDefinitionインターフェース 941
SimpleItemDefinition インターフェース 945
SimpleLabelEdge インターフェース 956
SimpleLabel インターフェース 951
SimpleNode インターフェース 957
SimpleTaxonomyBuilderFactory インターフェース 975
SimpleTaxonomyBuilder インターフェース 972
SimpleTaxonomy インターフェース 959

T

time 900

U

URI
　XPointer を使用した URI の指定方法 26
　エンティティリゾルバ 192
　指定方法 24

W

W3C DOM 8
W3C DOM〔用語解説〕 1176

X

XBRL
　XBRL 文書 2
　概要 2
　データ型 268
XBRLAbstractBinary クラス 836
XBRLAbstractPrecision クラス 838
XBRLArcroleRef インターフェース 411
XBRLArcroleType インターフェース 716
XBRLBase64Binary クラス 843
XBRLCalculationRelationship インターフェース 677
XBRLConstants クラス 912
XBRLContext2 インターフェース 419
XBRLContext インターフェース 412
XBRLDateTime クラス 853
XBRLDateUnion クラス 858
XBRLDate クラス 849
XBRLDecimals クラス 863
XBRLDefinitionRelationship インターフェース 678
XBRLDimensionalRelationshipSet インターフェース 685
XBRLDimensionalRelationship インターフェース 685

XBRLDimensionDefaultRelationship インターフェース 697
XBRLDimensionDomainRelationship インターフェース 698
XBRLDimensionException クラス 402
XBRLDimensionIllegalCycleException クラス 711
XBRLDimensionMemberインターフェース 670
XBRL Dimensions 3
XBRLDimensionUserException クラス 405
XBRLDocumentManager クラス 393
XBRLDocumentWriterFactory インターフェース 911
XBRLDocumentWriter インターフェース 907
XBRL DOM
　概要 8
　ツリーへの展開 16
XBRL DOM〔用語解説〕 1176
XBRLDomainMemberRelationship インターフェース 699
XBRLDOMException クラス 398
XBRLDOMRuntimeException クラス 399
XBRLDOMUserException クラス 400
XBRLDRSDimension インターフェース 700
XBRLDRSMember インターフェース 702
XBRLDuration クラス 866
XBRLEntity インターフェース 420
XBRLEqualUtil クラス 930
XBRLExplicitMember インターフェース 672
XBRLFootnoteRelationship インターフェース 678
XBRLFootnote インターフェース 422
XBRLFraction クラス 869
XBRLGDay クラス 872
XBRLGMonthDay クラス 877
XBRLGMonth クラス 875
XBRLGYearMonth クラス 882
XBRLGYear クラス 880
XBRLHasHypercubeRelationship インターフェース 704
XBRLHexBinary クラス 885
XBRLHypercubeDimensionRelationship インターフェース 707
XBRLIdCreator インターフェース 904
XBRLIdentifier インターフェース 429
XBRLIdManager クラス 919
XBRLIllegalCycleException クラス 708
XBRLInstanceDocumentBuilderFactory インターフェース 536
XBRLInstanceDocumentBuilder インターフェース 507
XBRLInstanceDocumentWriterFactory インターフェース 553
XBRLInstanceDocumentWriter インターフェース 550
XBRLInstanceDocument インターフェース 431
XBRLInstanceElement インターフェース 554
XBRLInstanceItem インターフェース 580
XBRLInstanceNode インターフェース 602
XBRLInstanceTuple インターフェース 616
XBRLLabelRelationship インターフェース 678
XBRLLabel インターフェース 719
XBRLLinkbaseRef インターフェース 622
XBRLLinkbase インターフェース 723
XBRLMeasureOrOperator インターフェース 625
XBRLMeasure インターフェース 624
XBRLNamespaceDeclaration クラス 923
XBRLNonNumericContext インターフェース 626
XBRLNumericContext インターフェース 628
XBRLOperator インターフェース 630
XBRLParseErrorHandlerインターフェース 391
XBRLPart インターフェース 727
XBRLPeriod インターフェース 634
XBRLPrecision クラス 891

XBRLPresentationRelationship インターフェース 678
XBRLQName クラス 893
XBRLReferenceRelationship インターフェース 679
XBRLReference インターフェース 728
XBRLRelationship インターフェース 679
XBRLRoleRef インターフェース 637
XBRLRoleType インターフェース 734
XBRLScenario インターフェース 639
XBRLSchemaLocation クラス 926
XBRLSchemaRef インターフェース 651
XBRLSegment インターフェース 652
XBRLTaxonomyAttributeDeclaration インターフェース 736
XBRLTaxonomyBody インターフェース 737
XBRLTaxonomyComplexTypeDefinition インターフェース 741
XBRLTaxonomyComponent インターフェース 746
XBRLTaxonomyDimensionItem インターフェース 832
XBRLTaxonomyDocumentBuilderFactory インターフェース 780
XBRLTaxonomyDocumentBuilder インターフェース 771
XBRLTaxonomyDocument インターフェース 748
XBRLTaxonomyElement インターフェース 796
XBRLTaxonomyHypercubeItem インターフェース 834
XBRLTaxonomyItem インターフェース 814
XBRLTaxonomyLinkbaseRef インターフェース 725
XBRLTaxonomyNode インターフェース 822
XBRLTaxonomyOtherElement インターフェース 828
XBRLTaxonomyRoleRef インターフェース 732
XBRLTaxonomySimpleTypeDefinition インターフェース 828
XBRLTaxonomyTuple インターフェース 828
XBRLTaxonomyTypeDefinition インターフェース 829
XBRLTime クラス 900
XBRLTypedMember インターフェース 673
XBRLUnit2 インターフェース 665
XBRLUnit インターフェース 663
XBRL 文書〔用語解説〕 1176
XBRL 文書の検証内容 1150
XLink プロセッサ 11
XLink プロセッサ〔用語解説〕 1176
XML
　XML プロセッサ 11
　スキーマ検証〔インスタンス文書〕 33
　スキーマ検証〔タクソノミー文書〕 21
XML プロセッサ〔用語解説〕 1176
XPointer 26
xsi:schemaLocation 属性 926
XSLT プロセッサ〔用語解説〕 1177

あ

アーク型要素の属性 69
アークロール 60
アイテム 4,581
　アイテム型要素〔XBRL2.0 の場合〕 231
　アイテム型要素〔XBRL2.1 の場合〕 263
　要素宣言 814
アイテム（item）〔用語解説〕 1177
アクティブ〔用語解説〕 1177

い

インスタンス（instance）〔用語解説〕 1177
インスタンス文書 5
　ID の採番をするインスタンス文書の作成 155
　オブジェクトモデル〔XBRL2.0 の場合〕 277

オブジェクトモデル〔XBRL2.1 の場合〕 281
カスタムロール，カスタムアークロールへの参照定義 260
基本構造〔XBRL2.0 の場合〕 229
基本構造〔XBRL2.1 の場合〕 256
検証 33
参照 20
出力フォーマット 92
新規作成〔XBRL2.0 の場合〕 122
新規作成〔XBRL2.1 の場合〕 146
タクソノミー文書への参照定義 259
ノード 603
編集〔XBRL2.0 の場合〕 134
編集〔XBRL2.1 の場合〕 160
保存〔XBRL2.0 の場合〕 132
保存〔XBRL2.1 の場合〕 154
読み込み〔XBRL2.0 の場合〕 134
読み込み〔XBRL2.1 の場合〕 160
読み込み〔XBRL Dimensions1.0 の場合〕 186
ルート要素〔XBRL2.1 の場合〕 259
インスタンス文書（instance document）〔用語解説〕 1177

え

エラーの通知 391
エラーハンドラ 191
エレメント要素〔XBRL2.0 の場合〕 217
エレメント要素〔XBRL2.1 の場合〕 237
エンコード 843
演算子要素 625,630
エンティティ（entity）〔用語解説〕 1177
エンティティリゾルバ 192

か

開発環境 115
拡張リンク
構文規則 31
構文の検証 30
拡張リンク（extended link）〔用語解説〕 1177
カスタムアークロール 241
カスタムロール 241
関係リンク
上書き 59
関係リンク〔XBRL2.0 の場合〕 222
関係リンク〔XBRL2.1 の場合〕 247
区分 4
完全な妥当性検証 8

き

期間 231
基底タクソノミー文書 16
脚注 422
脚注リンク 5
上書き 67
脚注リンク要素〔XBRL2.0 の場合〕 232
脚注リンク要素〔XBRL2.1 の場合〕 264
構文規則 43
追加〔XBRL2.0 の場合〕 135
追加〔XBRL2.1 の場合〕 161
脚注リンク（フォーマット） 93
脚注リンクの再作成 1170
協定世界時〔用語解説〕 1177

く

クラスパス 115

け

計算リンク
区分 4
計量 232,262
計量要素 624,625
検証
DTS の基点となる位置情報の有無 41
インスタンス文書 33
脚注リンクの構文 42
コンテキスト要素（context）および単位要素（unit） 41
タクソノミー文書 21
タプル 31,43

リンク〔インスタンス文書〕 33
リンク〔タクソノミー文書〕 24
リンクベース 22

こ

子要素の追加 125
コンテキスト（context）〔用語解説〕 1177
コンテキスト要素 41
コンテキスト要素〔XBRL2.0 の場合〕 231
コンテキスト要素〔XBRL2.1 の場合〕 261
数値コンテキストに対応したアイテムの追加〔XBRL2.0 の場合〕 130
数値コンテキスト要素の追加〔XBRL2.0 の場合〕 127
多次元データを含んだコンテキスト要素の追加〔XBRL Dimensions1.0 の場合〕 179
追加〔XBRL2.1 の場合〕 148
非数値コンテキストに対応したアイテムの追加〔XBRL2.0 の場合〕 129
非数値コンテキスト要素の追加〔XBRL2.0 の場合〕 124
コンパイル 197

さ

参照
インスタンス文書 20
インスタンス文書とリンクベース〔XBRL Dimensions1.0 の場合〕 187
タクソノミー文書 15
タクソノミー本体とリンクベース〔XBRL2.0 の場合〕 118
タクソノミー本体とリンクベース〔XBRL2.1 の場合〕 141
タクソノミー本体とリンクベース〔XBRL Dimensions1.0 の場合〕 175
リンク構造 72
サンプルアプリケーション 194

し

時期 231
時期（period）〔用語解説〕 1178
識別子要素 429
時期要素 231，634
事実〔用語解説〕 1178
実行環境 199
シナリオ要素 639
シングルトン 393

す

数値アイテムの追加 152
数値コンテキスト 628
スキーマ定義 197
スキーマ定義の準備 115
ストリーム 550

せ

正規化 100
セグメント要素 652

た

タクソノミー（taxonomy）〔用語解説〕 1178
タクソノミー文書 4
XBRL DOM ツリーへの展開 15
オブジェクトモデル〔XBRL2.0 の場合〕 279
オブジェクトモデル〔XBRL2.1 の場合〕 282
階層構造 6
拡張 5
カスタムロール，カスタムアークロールへの参照定義 245
基本構造〔XBRL2.0 の場合〕 216
基本構造〔XBRL2.1 の場合〕 235
検証 21
参照 15
出力フォーマット 86
ノード 822
読み込み〔XBRL2.0 の場合〕 117
読み込み〔XBRL2.1 の場合〕 139

読み込み〔XBRL Dimensions1.0 の場合〕 174
タクソノミー文書（taxonomy document）〔用語解説〕 1178
タクソノミー本体 4
リンクベースへの参照定義〔XBRL2.0 の場合〕 220
リンクベースへの参照定義〔XBRL2.1 の場合〕 243
タクソノミー本体〔用語解説〕 1178
タプル 4
タプル（tuple）〔用語解説〕 1178
タプル型
タプル型要素〔XBRL2.1 の場合〕 266
単位〔XBRL2.0 の場合〕 232
単位〔XBRL2.1 の場合〕 263
単位要素 664
単位要素〔XBRL2.0 の場合〕 232
単位要素〔XBRL2.1 の場合〕 261
追加〔XBRL2.1 の場合〕 150
単純なタクソノミー文書
オブジェクトモデル〔XBRL2.1 の場合〕 285
単純リンクのリモートリソースの参照先の検証 30

て

定義リンク
区分 4
定数 912
ディメンジョン 182
データ型
base64Binary 843
任意の数値精度 839
ユーザーデータ型 270

と

トラバース（トラバーサル）〔用語解説〕 1178
トレースログ 213

な

名前空間
一覧の作成 100
正規化されない要素 107
名前空間 URI 923,926
名前空間接頭辞 923
自動生成 101
名前空間 URI との対応 276
名前空間宣言 92

の

ノード〔用語解説〕 1179
ノードのインポート 499

は

パート 727
バイナリデータ 836,843,885
パッケージの概要 276

ひ

非数値アイテムの追加 151
非数値コンテキスト 626

ふ

プライマリー・アイテム 182
プライマリー・アイテムの追加 182
プレゼンテーションリンク
区分 4
プロパティ
一覧 199
指定方法と優先順位 199
プロパティファイルの記載形式 209
文献への参照 727,728

ほ

ポイント（point）〔用語解説〕 1179

め

メソッド一覧 338
メッセージ一覧 980

メッセージの形式 978
メモリキャッシュ 53

ゆ

ユーザーデータ型
　　検証 23
ユーザーデータ型〔XBRL2.0 の場合〕 270
ユーザーデータ型〔XBRL2.1 の場合〕 272

よ

要素 555
　　関係づけ 95
　　二重定義 30

ら

ラベル 719
ラベルリンク
　　上書き 63
　　区分 4
　　ラベルリンク〔XBRL2.0 の場合〕 222
　　ラベルリンク〔XBRL2.1 の場合〕 247

り

リファレンスリンク
　　上書き 66
　　区分 5
　　リファレンスリンク〔XBRL2.0 の場合〕 223
リンク
　　上書き〔XBRL2.0 の場合〕 59
　　上書き〔XBRL2.1 の場合〕 68
　　検証〔インスタンス文書〕 33
　　検証〔タクソノミー文書〕 24
　　リンク構造の検証〔インスタンス文書〕 43
　　リンク構造の検証〔タクソノミー文書〕 31
　　リンク構造の参照 72
リンク〔用語解説〕 1179
リンクベース 4
　　検証 22
　　遅延読み込み 54
　　リンクベース要素〔XBRL2.0 の場合〕 221
　　リンクベース要素〔XBRL2.1 の場合〕 244
リンクベース（linkbase）〔用語解説〕 1179

る

ルート要素〔XBRL2.0 の場合〕 230

れ

例外通知 398，402
レンジ（range）〔用語解説〕 1179

ろ

ロケーション（location）〔用語解説〕 1179

ソフトウェアマニュアルのサービス ご案内

1．マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しています。

URL　http://www.hitachi.co.jp/soft/manual/

ホームページのメニューは次のとおりです。

■マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ，マニュアル名称，資料番号のいずれかから検索できます。
■CD-ROMマニュアル	日立ソフトウェアマニュアルと製品群別CD-ROMマニュアルの仕様について記載しています。
■マニュアルのご購入	マニュアルご購入時のお申し込み方法を記載しています。
■オンラインマニュアル	一部製品のマニュアルをインターネットで公開しています。
■サポートサービス	ソフトウェアサポートサービスお客様向けページでのマニュアル公開サービスを記載しています。
■ご意見・お問い合わせ	マニュアルに関するご意見，ご要望をお寄せください。

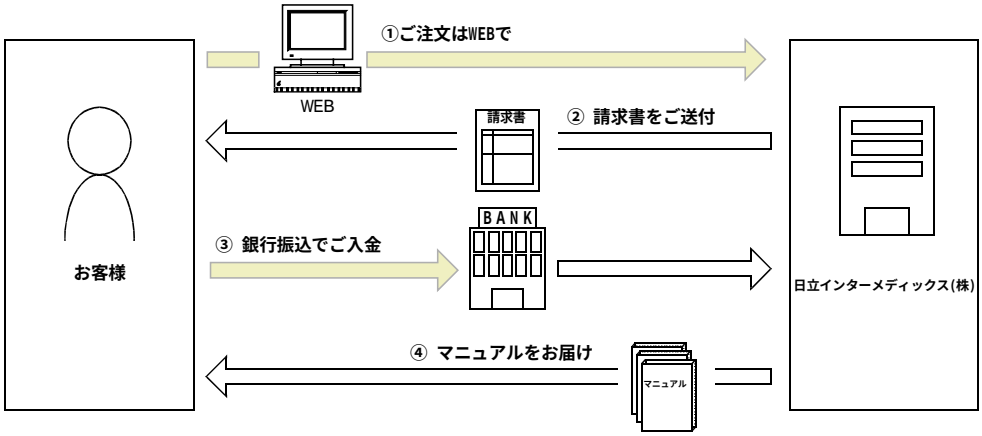
2．インターネットでのマニュアル公開

2 種類のマニュアル公開サービスを実施しています。

(1) マニュアル情報ホームページ「オンラインマニュアル」での公開
製品をよりご理解いただくためのご参考として，一部製品のマニュアルを公開しています。

(2) ソフトウェアサポートサービスお客様向けページでのマニュアル公開
ソフトウェアサポートサービスご契約のお客様向けにマニュアルを公開しています。公開しているマニュアルの一覧，本サービスの対象となる契約の種別などはマニュアル情報ホームページの「サポートサービス」をご参照ください。

3．マニュアルのご注文



① マニュアル情報ホームページの「マニュアルのご購入」にアクセスし，お申し込み方法をご確認のうえ WEB からご注文ください。ご注文先は日立インターメディックス(株)となります。

② ご注文いただいたマニュアルについて請求書をお送りします。

③ 請求書の金額を指定銀行へ振り込んでください。

④ 入金確認後 7 日以内にお届けします。在庫切れの場合は，納期を別途ご案内いたします。