

---

# Cosminexus XML Processor ユーザーズガイド

解説・手引書

3020-3-M44-20

## 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。

# HITACHI

## 対象製品

適用 OS : Windows Server 2003 , Windows Server 2003 x64 Editions

P-2443-7D74 uCosminexus Application Server Standard 07-50

P-2443-7K74 uCosminexus Application Server Enterprise 07-50

P-2443-7S74 uCosminexus Service Platform 07-50

適用 OS : Windows Server 2003 , Windows Vista , Windows XP

P-2443-7E74 uCosminexus Developer Standard 07-50

P-2443-7F74 uCosminexus Developer Professional 07-50

P-2443-7U74 uCosminexus Operator 07-50

P-2443-7T74 uCosminexus Service Architect 07-50

適用 OS : Windows Server 2003 , Windows Server 2003 x64 Editions , Windows Vista , Windows XP

P-2443-7H74 uCosminexus Client 07-50

適用 OS : AIX 5L V5.2 , AIX 5L V5.3

P-1M43-7D71 uCosminexus Application Server Standard 07-50

P-1M43-7K71 uCosminexus Application Server Enterprise 07-50

P-1M43-7S71 uCosminexus Service Platform 07-50

適用 OS : HP-UX 11i ( PA-RISC ) , HP-UX 11i V2 ( PA-RISC )

P-1B43-7D71 uCosminexus Application Server Standard 07-50

P-1B43-7K71 uCosminexus Application Server Enterprise 07-50

適用 OS : HP-UX 11i V2 ( IPF ) , HP-UX 11i V3 ( IPF )

P-1J43-7D71 uCosminexus Application Server Standard 07-50

P-1J43-7K71 uCosminexus Application Server Enterprise 07-50

P-1J43-7S71 uCosminexus Service Platform 07-50

適用 OS : Red Hat Enterprise Linux AS 3 ( x86 ) , Red Hat Enterprise Linux AS 4 ( x86 ) , Red Hat Enterprise Linux ES 3 ( x86 ) , Red Hat Enterprise Linux ES 4 ( x86 ) , Red Hat Enterprise Linux AS 3 ( AMD64 & Intel EM64T ) , Red Hat Enterprise Linux AS 4 ( AMD64 & Intel EM64T ) , Red Hat Enterprise Linux ES 3 ( AMD64 & Intel EM64T ) , Red Hat Enterprise Linux ES 4 ( AMD64 & Intel EM64T )

P-9S43-7D71 uCosminexus Application Server Standard 07-50

P-9S43-7K71 uCosminexus Application Server Enterprise 07-50

P-9S43-7S71 uCosminexus Service Platform 07-50

これらのプログラムプロダクトのほかにも、このマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

印の製品については、サポート時期をご確認ください。

## 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

AIX は、米国における米国 International Business Machines Corp. の登録商標です。

HP-UX は、米国 Hewlett-Packard Company のオペレーティングシステムの名称です。

Itanium は、アメリカ合衆国および他の国におけるインテル コーポレーションまたはその子会社の登録商標です。

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あるいは商標です。

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。

Solaris 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。

Sun 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。

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、米国およびその他の国における米国 Microsoft Corp. の商標です。

Windows Vista は、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

## 発行

2006 年 4 月（第 1 版）3020-3-M44

2007 年 6 月（第 2 版）3020-3-M44-20

## 著作権

All Rights Reserved. Copyright (C) 2006, 2007, Hitachi, Ltd.

変更内容

変更内容 (3020-3-M44-20)    uCosminexus Application Server Standard 07-50    uCosminexus Application Server Enterprise 07-50    uCosminexus Service Platform 07-50    uCosminexus Developer Standard 07-50    uCosminexus Developer Professional 07-50    uCosminexus Operator 07-50    uCosminexus Service Architect 07-50    uCosminexus Client 07-50

追加・変更内容	変更箇所
サポートする OS を変更した。	-
高速パース機能をサポートした。	1.4.5 , 3.3 , 4.5 , 6.17
メッセージを変更した。 KECX01024-E , KECX06249-W	7.2.1 , 7.2.5
メッセージを追加した。 KECX01088-E , KECX09601-E , KECX09602-E , KECX09603-E , KECX09604-E , KECX09605-E , KECX09606-E , KECX09607-E , KECX09608-E	7.2.1 , 7.2.7

単なる誤字・脱字などはお断りなく訂正しました。

# はじめに

---

このマニュアルは、Cosminexus XML Processor が提供する XML パーサ・XSLT トランスフォーマの機能，作成方法，および使用方法について説明したものです。Cosminexus XML Processor は，Cosminexus を構成する各製品で提供されています。詳細は，マニュアル「Cosminexus 概説」を参照してください。

## 対象読者

このマニュアルは，Cosminexus XML Processor を使用して，XML 文書进行操作するプログラムを Java 言語で作成されるプログラマの方を対象としています。

また，次に示す知識をお持ちの方を対象としています。

- XML についての基本的な知識
- JAXP についての基本的な知識
- Java 言語によるオブジェクト指向プログラミングの知識

## 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 Cosminexus XML Processor の概要

Cosminexus XML Processor を使用する前に知っておく必要のある，Cosminexus XML Processor の特長，位置づけ，処理の流れ，およびサポート範囲について説明しています。サポート範囲では，使用できる XML パーサ・XSLT トランスフォーマの機能，および処理できる文字コードについて説明し，Cosminexus XML Processor の拡張機能について紹介しています。

### 第 2 章 JAXP の機能

JAXP を使用してプログラムを作成する際に知っておく必要のある JAXP の概要について説明しています。

### 第 3 章 Cosminexus XML Processor の拡張機能

Cosminexus XML Processor の拡張機能について説明しています。

### 第 4 章 フィーチャーおよびプロパティの使用法

JAXP で規定されているフィーチャーおよびプロパティの使用法について説明しています。また，Cosminexus XML Processor の拡張機能に関連したフィーチャーおよびプロパティの使用法についても説明しています。

### 第 5 章 プログラムの作成方法

プログラム作成の流れ，使用するパッケージ，および JAXP の主な機能を用いたサンプルプログラムについて説明しています。

### 第 6 章 Cosminexus XML Processor 使用時の注意事項

Cosminexus XML Processor 使用時の注意事項について説明しています。

はじめに

## 第 7 章 メッセージ

Cosminexus XML Processor で出力されるメッセージの形式，およびメッセージの意味について説明しています。

## 付録 A バージョン間の差異

Cosminexus XML Processor のバージョン間の相違点について説明しています。

## 付録 B 用語解説

このマニュアルで使用している用語の意味を説明しています。

## 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

- Cosminexus 概説 ( 3020-3-M01 )
- Cosminexus リファレンス 定義編 ( 3020-3-M11 )
- Cosminexus アプリケーション開発ガイド ( 3020-3-M41 )

## 読書手順

このマニュアルは、利用目的ごとに章を選択して読むことができます。各章の利用目的を次の表に示します。

章タイトル	利用目的
第 1 章    Cosminexus XML Processor の概要	<ul style="list-style-type: none"> <li>• Cosminexus XML Processor の特長を知りたい</li> <li>• Cosminexus XML Processor の Cosminexus での位置づけを知りたい</li> <li>• Cosminexus XML Processor の処理の流れを知りたい</li> <li>• Cosminexus XML Processor で使用できる文字コードなどサポート範囲を知りたい</li> </ul>
第 2 章    JAXP の機能	<ul style="list-style-type: none"> <li>• JAXP の概要について知りたい</li> <li>• JAXP が規定するパッケージとその機能について知りたい</li> </ul>
第 3 章    Cosminexus XML Processor の拡張機能	<ul style="list-style-type: none"> <li>• Shift_JIS 切り替え機能について知りたい</li> <li>• XSLTC トランスフォーマ機能について知りたい</li> <li>• 高速パース機能について知りたい</li> </ul>
第 4 章    フィーチャーおよびプロパティの使用 方法	<ul style="list-style-type: none"> <li>• JAXP で規定されているフィーチャーおよびプロパティの使用方法について知りたい</li> <li>• SAX2 のフィーチャーおよびプロパティの使用 方法について知りたい</li> <li>• XSLT のフィーチャーの使用方法について知 りたい</li> <li>• XML Schema のプロパティの使用方法について 知りたい</li> <li>• Shift_JIS 切り替え機能のプロパティの使用方 法について知りたい</li> <li>• 高速パース機能のプロパティの使用方 法について知りたい</li> </ul>
第 5 章    プログラムの作成方法	<ul style="list-style-type: none"> <li>• Cosminexus XML Processor を使用してプロ グラムを作成する際の作業の流れについて知 りたい</li> <li>• プログラム作成時に使用するパッケージにつ いて知りたい</li> <li>• 障害対策について知りたい</li> <li>• サンプルプログラムについて知りたい</li> </ul>
第 6 章    Cosminexus XML Processor 使用時の 注意事項	<ul style="list-style-type: none"> <li>• Cosminexus XML Processor を使用するとき の注意事項を知りたい</li> </ul>
第 7 章    メッセージ	<ul style="list-style-type: none"> <li>• メッセージについて知りたい</li> </ul>
付録 A    バージョン間の差異	<ul style="list-style-type: none"> <li>• Cosminexus XML Processor のバージョン間 の相違点について知りたい</li> </ul>
付録 B    用語解説	<ul style="list-style-type: none"> <li>• このマニュアルの本文中で使用されている用 語の意味を知りたい</li> </ul>

## このマニュアルでの表記

- このマニュアルでは、製品名称を次のように表記しています。

このマニュアルでの表記	製品名称
Windows Server 2003	Microsoft(R) Windows Server(TM) 2003 , Enterprise Edition Operating System
	Microsoft(R) Windows Server(TM) 2003 , Standard Edition Operating System
	Microsoft(R) Windows Server(TM) 2003 Release2 , Enterprise Edition Operating System
	Microsoft(R) Windows Server(TM) 2003 Release2 , Standard Edition Operating System
Windows Server 2003 x64 Editions	Microsoft(R) Windows Server(TM) 2003 , Enterprise x64 Edition Operating System
	Microsoft(R) Windows Server(TM) 2003 , Standard x64 Edition Operating System
	Microsoft(R) Windows Server(TM) 2003 Release2 , Enterprise x64 Edition Operating System
	Microsoft(R) Windows Server(TM) 2003 Release2 , Standard x64 Edition Operating System
Windows Vista	Microsoft(R) Windows Vista(TM), Business Edition Operating System
	Microsoft(R) Windows Vista(TM), Enterprise Edition Operating System
	Microsoft(R) Windows Vista(TM), Ultimate Edition Operating System
Windows XP	Microsoft(R) Windows(R) XP Professional Operating System

- このマニュアルでは、Java(TM) 関連の用語を次のように表記しています。

用語	このマニュアルでの表記
EJB(TM)	EJB
Enterprise JavaBeans(TM)	Enterprise JavaBeans
J2EE(TM)	J2EE
Java(TM)	Java
JSP(TM)	JSP



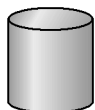
**このマニュアルで使用している略語**  
このマニュアルでは、次に示す英略語を使用します。

英略語	英字の表記
API	Application Programming Interface
DOM	Document Object Model
DTD	Document Type Definition
EJB	Enterprise JavaBeans
IANA	Internet Assigned Numbers Authority
IPF	Itanium(R) Processor Family
J2EE	Java 2 Platform, Enterprise Edition
JAR	Java Archive
JAXP	Java API for XML Processing
JSP	Java Server Pages
SAX	Simple API for XML
SAX1	Simple API for XML 1.0
SAX2	Simple API for XML 2.0
SOAP	Simple Object Access Protocol
TrAX	Transformation API for XML
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations

## 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ています。

●ファイル



●仮想ファイル



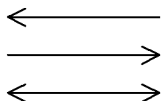
●プログラム



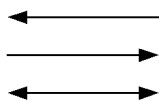
●データの流れ



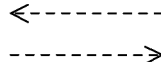
●制御の流れ



●その他の流れ



●参照



## 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）、汎用（はんよう）、必須（ひつす）

## 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。

# 目次

1	Cosminexus XML Processor の概要	1
1.1	Cosminexus XML Processor の特長	2
1.1.1	プラットフォーム間で共通のプログラムを作成できる	2
1.1.2	Java プログラムで XML 文書を容易に解析できる	2
1.1.3	Java プログラムで XML 文書を容易に変換できる	3
1.2	Cosminexus XML Processor の位置づけ	4
1.3	Cosminexus XML Processor の処理の流れ	5
1.4	Cosminexus XML Processor のサポート範囲	7
1.4.1	使用できる XML パーサ, XSLT トランスフォーマなどの機能	7
1.4.2	処理できる文字コード	8
1.4.3	Shift_JIS 切り替え機能	9
1.4.4	XSLTC トランスフォーマ機能	9
1.4.5	高速パース機能	9
2	JAXP の機能	11
2.1	JAXP とは	12
2.2	JAXP が規定するパッケージとその機能	13
2.2.1	javax.xml.parsers パッケージ	14
2.2.2	javax.xml.transform パッケージ	16
2.2.3	javax.xml.transform.dom パッケージ	16
2.2.4	javax.xml.transform.sax パッケージ	17
2.2.5	javax.xml.transform.stream パッケージ	17
2.2.6	javax.xml.validation パッケージ	18
2.2.7	javax.xml.xpath パッケージ	19
2.2.8	javax.xml.namespace パッケージ	19
2.2.9	javax.xml.datatype パッケージ	20
2.2.10	javax.xml パッケージ	20
2.2.11	org.w3c.dom パッケージ	21
2.2.12	org.w3c.dom.bootstrap パッケージ	21
2.2.13	org.w3c.dom.ls パッケージ	22
2.2.14	org.w3c.dom.events パッケージ	23
2.2.15	org.xml.sax パッケージ	23
2.2.16	org.xml.sax.ext パッケージ	24

2.2.17	org.xml.sax.helpers パッケージ	24
--------	---------------------------	----

## 3

Cosminexus XML Processor の拡張機能	25
3.1 Shift_JIS 切り替え機能	26
3.2 XSLTC トランスフォーマ機能	28
3.2.1 XSLTC トランスフォーマの概要	28
3.2.2 XSLTC トランスフォーマを使ったシステムの開発・運用	29
3.2.3 XSLTC トランスフォーマで使用するクラス	29
3.2.4 XSLTC トランスフォーマの使用方法	30
3.3 高速パース機能	32
3.3.1 高速パース機能の概要	32
3.3.2 高速パースのための作業の流れ	34
3.3.3 事前解析用 XML 文書の作成	36
3.3.4 解析結果オブジェクトの生成	44
3.3.5 解析結果オブジェクトの設定	45
3.3.6 XML 文書の解析	45
3.3.7 高速パース機能で使用するクラス	46
3.3.8 高速パース機能を使用するためのコード例	51
3.3.9 事前解析用 XML 文書のチューニング	52
3.3.10 チューニングサマリファイルの詳細	56
3.3.11 チューニング詳細ファイルの詳細	57
3.3.12 高速化定義ファイルを使用した高速パース	60
3.3.13 高速化定義ファイルを使用した高速パースのための作業の流れ	62
3.3.14 高速化定義ファイルの作成	63
3.3.15 高速化定義ファイルの設定	65
3.3.16 事前解析の実行タイミングの設定	65
3.3.17 XML 文書の解析（高速化定義ファイルを使用する場合）	68
3.4 XML Schema のエラーメッセージと W3C 仕様との対応	70

## 4

フィーチャーおよびプロパティの使用方法	71
4.1 SAX2 のフィーチャーおよびプロパティの使用方法	72
4.2 XSLT のフィーチャーの使用方法	74
4.3 XML Schema のプロパティの使用方法	75
4.3.1 DOM パーサに対する XML Schema のプロパティの設定方法	75
4.3.2 SAX パーサに対する XML Schema のプロパティの設定方法	76

4.3.3 スキーマ文書を XML 文書内で設定する方法	77
4.4 Shift_JIS 切り替え機能のプロパティの使用方法	79
4.4.1 DOM パーサで Shift_JIS 切り替え機能を使用する方法	80
4.4.2 SAX パーサで Shift_JIS 切り替え機能を使用する方法	80
4.4.3 XSLT トランスフォーマで Shift_JIS 切り替え機能を使用する方法	81
4.5 高速パース機能のプロパティの使用方法	82
4.5.1 DocumentBuilder に解析結果オブジェクトを設定する方法	82
4.5.2 SAXParser に解析結果オブジェクトを設定する方法	83
4.5.3 XMLReader に解析結果オブジェクトを設定する方法	84

## 5

プログラムの作成方法	85
5.1 プログラム作成の流れ	86
5.2 Cosminexus XML Processor が使用しているパッケージ名	88
5.3 プログラム実行時のトラブルシュート	89
5.4 DOM パーサを使用するサンプルプログラム	90
5.4.1 サンプルプログラムの処理	90
5.4.2 サンプルプログラムの作成の流れ	90
5.4.3 サンプルプログラム ( SampleDOM.java )	91
5.4.4 サンプルプログラム ( SampleDOM.java ) の実行結果	94
5.5 SAX パーサを使用するサンプルプログラム	95
5.5.1 サンプルプログラムの処理	95
5.5.2 サンプルプログラムの作成の流れ	95
5.5.3 使用する XML 文書 ( SampleSAX.xml )	96
5.5.4 サンプルプログラム ( SampleSAX.java )	97
5.5.5 サンプルプログラム ( SampleSAX.java ) の実行結果	100
5.6 XML Schema を使用するサンプルプログラム	101
5.6.1 サンプルプログラムの処理	101
5.6.2 使用する XML 文書 ( purchaseOrder.xml , purchaseOrder-fail.xml )	102
5.6.3 使用するスキーマ文書 ( purchaseOrder.xsd , personalData.xsd )	104
5.6.4 DOM パーサを使用する場合のサンプルプログラム	108
5.6.5 SAX パーサを使用する場合のサンプルプログラム	110
5.6.6 検証に失敗した場合の実行結果	111
5.7 XSLT トランスフォーマを使用するサンプルプログラム	113
5.7.1 サンプルプログラムの処理	113
5.7.2 サンプルプログラムの作成の流れ	113
5.7.3 使用する XML 文書 ( SampleXSLT.xml )	114

5.7.4	使用する XSL ファイル ( SampleXSLT.xml )	115
5.7.5	サンプルプログラム ( SampleXSLT.java )	116
5.7.6	サンプルプログラム ( SampleXSLT.java ) の実行結果	117
5.8	XSLTC トランスフォーマを使用するサンプルプログラム	118

## 6

Cosminexus XML Processor 使用時の注意事項	119
6.1 JAXP1.3 機能の共通の注意事項	121
6.2 DOM パーサに関する注意事項	122
6.3 SAX パーサに関する注意事項	124
6.4 スキーマ検証に関する注意事項	126
6.5 XSLT/XSLTC 共通の注意事項	127
6.6 XSLT に関する注意事項	130
6.6.1 エラーを通知しないケース	130
6.6.2 その他の注意事項	130
6.7 XSLTC に関する注意事項	131
6.7.1 スタイルシートサイズの注意事項	131
6.7.2 変換処理性能に関する注意事項	131
6.7.3 XSLT 要素の注意事項	131
6.7.4 XPath 式の注意事項	134
6.7.5 エラーを通知しないケース	135
6.7.6 その他の注意事項	137
6.8 javax.xml.datatype パッケージに関する注意事項	138
6.9 javax.xml.validation パッケージに関する注意事項	139
6.10 javax.xml.xpath パッケージに関する注意事項	141
6.11 org.w3c.dom パッケージに関する注意事項	143
6.12 org.w3c.dom.bootstrap パッケージに関する注意事項	145
6.13 org.w3c.dom.ls パッケージに関する注意事項	146
6.14 org.xml.sax.ext パッケージに関する注意事項	148
6.15 XInclude に関する注意事項	149
6.16 実装依存仕様に関する注意事項	150
6.17 高速パース機能に関する注意事項	152
6.17.1 解析結果オブジェクトの設定に関する注意事項	152
6.17.2 解析結果オブジェクトを利用する XML パーサに関する注意事項	152
6.17.3 XML1.1 に関する注意事項	152
6.17.4 XInclude に関する注意事項 ( 高速パース機能 )	152

6.17.5	高速化定義ファイルに関する注意事項	153
6.17.6	絶対パスの指定方法に関する注意事項	153
6.17.7	事前解析用 XML 文書に関する注意事項	153
6.17.8	チューニング情報に関する注意事項	154
6.17.9	解析性能に関する注意事項	154

## 7

メッセージ	155
-------	-----

7.1	メッセージの形式	156
7.2	メッセージ一覧	157
7.2.1	KECX01000 ~ KECX01999	157
7.2.2	KECX02000 ~ KECX02999	196
7.2.3	KECX03000 ~ KECX03999	228
7.2.4	KECX04000 ~ KECX04999	246
7.2.5	KECX06000 ~ KECX06999	264
7.2.6	KECX07000 ~ KECX07999	304
7.2.7	KECX09000 ~ KECX09999	321

## 付録

付録 A	バージョン間の差異	323
付録 A.1	DOM パーサ, および SAX パーサ共通の動作の差異	324
付録 A.2	DOM パーサの動作の差異	324
付録 A.3	SAX パーサの動作の差異	325
付録 A.4	スキーマ検証の動作の差異	326
付録 A.5	XSLT の動作の差異	326
付録 B	用語解説	329

## 索引

索引	333
----	-----

## 図目次

図 1-1	JAXP で XML 文書を解析する流れ	3
図 1-2	JAXP で XML 文書を変換する流れ	3
図 1-3	Cosminexus 実行環境での Cosminexus XML Processor の位置づけ	4
図 1-4	XML パーサ機能および XSLT トランスフォーマ機能の処理の流れ	5
図 2-1	DOM パーサでの解析を行うアプリケーションの処理の流れ	14
図 2-2	SAX パーサでの解析を行うアプリケーションの処理の流れ	15
図 2-3	XSLT による変換を行うアプリケーションの処理の流れ	16
図 2-4	javax.xml.validation パッケージに含まれるクラスを用いて XML 文書を検証するアプリケーションの処理の流れ	18
図 2-5	XPath 式を評価するアプリケーションの処理の流れ	19
図 2-6	日付 / 時刻型データを処理するアプリケーションの処理の流れ	20
図 2-7	DOM ツリーを操作するアプリケーションの処理の流れ	21
図 2-8	XML 文書のロード, および保存を行うアプリケーションの処理の流れ	22
図 2-9	SAX イベントを処理するアプリケーションの処理の流れ	23
図 3-1	高速パース機能を利用したシステム	32
図 3-2	解析結果オブジェクトを生成して, 高速パースを実現する流れ	33
図 3-3	高速パース機能を使用するための作業の流れ	35
図 3-4	XML 宣言を記述する例 (事前解析用 XML 文書の作成指針)	36
図 3-5	要素と属性を記述する例 (事前解析用 XML 文書の作成指針)	37
図 3-6	空要素と空要素タグを記述する例 (事前解析用 XML 文書の作成指針)	38
図 3-7	改行文字・空白文字の数や位置が異なる要素を記述する例 (事前解析用 XML 文書の作成指針)	39
図 3-8	要素と属性を記述する例 (事前解析用 XML 文書の作成指針・要素と属性の順序)	40
図 3-9	同じ組み合わせで順序が異なる属性を記述する例 (事前解析用 XML 文書の作成指針)	40
図 3-10	要素の入れ子構造を記述する例 (事前解析用 XML 文書の作成指針)	41
図 3-11	要素の入れ子構造を複数記述する例 (事前解析用 XML 文書の作成指針)	41
図 3-12	要素の繰り返し構造を記述する例 (事前解析用 XML 文書の作成指針)	42
図 3-13	テキスト, CDATA セクション, 属性値を記述する例 (事前解析用 XML 文書の作成指針)	42
図 3-14	空要素を記述する例 (事前解析用 XML 文書の作成指針)	43
図 3-15	DTD, コメント, 処理命令に関する例 (事前解析用 XML 文書の作成指針)	43
図 3-16	XML 文書中から参照されているエンティティの定義を記述する例 (事前解析用 XML 文書の作成指針)	44



図 3-17 高速化定義ファイルを使用した高速パースの処理概要	61
図 3-18 高速化定義ファイルを使用して高速パース機能を使用するための作業の流れ	62
図 3-19 事前解析の実行タイミング	67
図 5-1 Cosminexus XML Processor を使用したプログラム作成の流れ	86
図 5-2 DOM パーサを使用するサンプルプログラムの作成の流れ	90
図 5-3 SAX パーサを使用するサンプルプログラムの作成の流れ	95
図 5-4 XML Schema を使用するサンプルプログラムの処理の流れ	101
図 5-5 スキーマ文書の名前空間の関係	104
図 5-6 XSLT トランスフォーマーを使用するサンプルプログラムの作成の流れ	114
図 5-7 XSLTC トランスフォーマー機能を使用する場合のプログラム変更	118

# 表目次

表 2-1	Cosminexus XML Processor が提供する JAXP の機能	12
表 2-2	JAXP が規定するパッケージに含まれる API の概要	13
表 3-1	XML 文書の encoding 指定と適用される文字エンコーディングとの対応	26
表 3-2	事前解析用 XML 文書を作成するための指針	36
表 3-3	高速パース機能の対象となる parse メソッド	45
表 3-4	通常的高速パースと高速化定義ファイルを使用した高速パースの相違点	61
表 3-5	高速パース機能の対象となる parse メソッド（高速化定義ファイルを使用する場合）	69
表 4-1	Cosminexus XML Processor がサポートしている SAX2 のフィーチャー一覧	72
表 4-2	Cosminexus XML Processor がサポートしている SAX2 のプロパティ一覧	72
表 4-3	JAXP で規定されている XSLT のフィーチャー一覧	74
表 4-4	特定するスキーマ文書の種類と使用する属性の関係	77
表 4-5	設定するプロパティの値とその意味（Shift_JIS 切り替え機能）	79
表 4-6	Shift_JIS 切り替え機能の設定用メソッド	79
表 4-7	設定するプロパティの値とその意味（高速パース機能）	82
表 4-8	解析結果オブジェクトの設定用メソッド	82
表 6-1	JAXP1.3 機能の共通の注意事項	121
表 6-2	DOM パーサに関する注意事項	122
表 6-3	SAX パーサに関する注意事項	124
表 6-4	スキーマ検証に関する注意事項	126
表 6-5	XSLT/XSLTC 共通の注意事項	127
表 6-6	XSLT に関する注意事項（エラーを通知しないケース）	130
表 6-7	XSLT に関する注意事項（その他）	130
表 6-8	XSLTC に関する注意事項（スタイルシートサイズ）	131
表 6-9	XSLTC に関する注意事項（変換処理性能）	131
表 6-10	XSLTC に関する注意事項（XSLT 要素）	131
表 6-11	XSLTC に関する注意事項（XPath 式）	134
表 6-12	XSLTC に関する注意事項（エラーを通知しないケース）	135
表 6-13	XSLTC に関する注意事項（その他）	137
表 6-14	javax.xml.datatype パッケージに関する注意事項	138
表 6-15	javax.xml.validation パッケージに関する注意事項	139
表 6-16	Validator クラスの validate(Source) , validate(Source, Result) の引数の組み合わせ	140
表 6-17	javax.xml.xpath パッケージに関する注意事項	141
表 6-18	org.w3c.dom パッケージに関する注意事項	143

表 6-19	DOMConfiguration オブジェクトに設定可能なパラメタ	144
表 6-20	org.w3c.dom.bootstrap パッケージに関する注意事項	145
表 6-21	org.w3c.dom.ls パッケージに関する注意事項	146
表 6-22	DOMConfiguration オブジェクトに設定可能なパラメタ (LSParser オブジェクト)	147
表 6-23	DOMConfiguration オブジェクトに設定可能なパラメタ (LSSerializer オブジェクト)	147
表 6-24	org.xml.sax.ext パッケージに関する注意事項	148
表 6-25	XInclude に関する注意事項	149
表 6-26	データ型に関する実装依存仕様	150
表 6-27	制約ファセットに関する実装依存仕様	151
表 6-28	出現回数の指定に関する実装依存仕様	151
表 6-29	高速パース機能に関する注意事項 (解析結果オブジェクトの設定)	152
表 6-30	高速パース機能に関する注意事項 (解析結果オブジェクトを利用する XML パーサ)	152
表 6-31	高速パース機能に関する注意事項 (XML1.1)	152
表 6-32	高速パース機能に関する注意事項 (XInclude)	152
表 6-33	高速パース機能に関する注意事項 (高速化定義ファイル)	153
表 6-34	高速パース機能に関する注意事項 (絶対パスの指定方法)	153
表 6-35	高速パース機能に関する注意事項 (事前解析用 XML 文書)	153
表 6-36	高速パース機能に関する注意事項 (チューニング情報)	154
表 6-37	高速パース機能に関する注意事項 (解析性能)	154
表 A-1	DOM パーサ, および SAX パーサ共通の動作の差異	324
表 A-2	DOM パーサの動作の差異	324
表 A-3	SAX パーサの動作の差異	325
表 A-4	スキーマ検証の動作の差異	326
表 A-5	XSLT の動作の差異	326



# 1

## Cosminexus XML Processor の概要

この章では，Cosminexus XML Processor を使用する前に知っておく必要のある Cosminexus XML Processor の特長，位置づけ，処理の流れ，サポート範囲について説明します。サポート範囲では，使用できる XML パーサ・XSLT トランスフォーマの機能，および処理できる文字コードについて説明し，Cosminexus XML Processor の拡張機能について紹介します。

---

1.1 Cosminexus XML Processor の特長

---

1.2 Cosminexus XML Processor の位置づけ

---

1.3 Cosminexus XML Processor の処理の流れ

---

1.4 Cosminexus XML Processor のサポート範囲

---

## 1.1 Cosminexus XML Processor の特長

---

Cosminexus XML Processor は Java 言語用の標準 XML API である JAXP1.3 をサポートし、その実装である XML パーサ機能、および XSLT トランスフォーマ機能などを提供するミドルウェアです。次に Cosminexus XML Processor の特長を説明します。

### 1.1.1 プラットフォーム間で共通のプログラムを作成できる

Cosminexus XML Processor は JAXP1.3 をサポートします。JAXP (Java API for XML Processing) は、米国 Sun Microsystems, Inc. が提供する Java 言語用の標準 XML API です。

JAXP には、XML パーサ (DOM パーサ・SAX パーサ)、XSLT トランスフォーマなどを生成する API が含まれています。XML パーサ、XSLT トランスフォーマなどの実装やコードに依存しないので、XML 文書进行处理・操作するプログラムを作成する際に、使用する XML パーサ、XSLT トランスフォーマなどの実装を意識する必要がありません。そのため、異なる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。

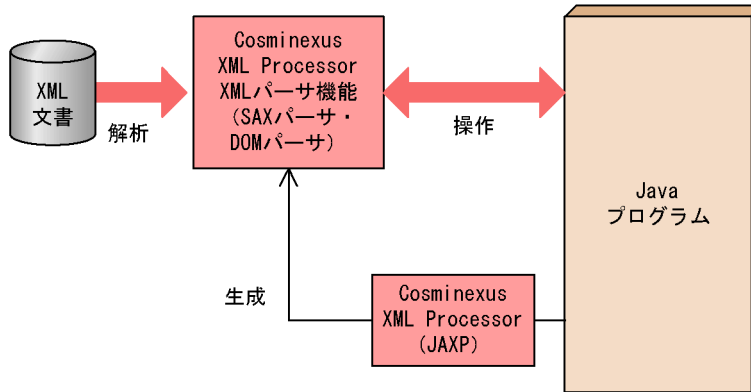
### 1.1.2 Java プログラムで XML 文書を容易に解析できる

Cosminexus XML Processor は JAXP の実装を提供しているため、Java プログラムで XML パーサを使用できます。

XML パーサは、XML 文書を解析し、解析したデータを扱いやすいデータ構造にして、ユーザプログラムに提供するライブラリです。XML 文書を解析する際には、XML 文書内のタグや属性などが、XML Schema や DTD に従っているかどうかを検証できます。

JAXP で XML 文書を解析する流れを図 1-1 に示します。JAXP を使用すると、Java プログラムで XML パーサ (SAX パーサ、または DOM パーサ) を生成して、XML 文書を解析できます。

図 1-1 JAXP で XML 文書を解析する流れ



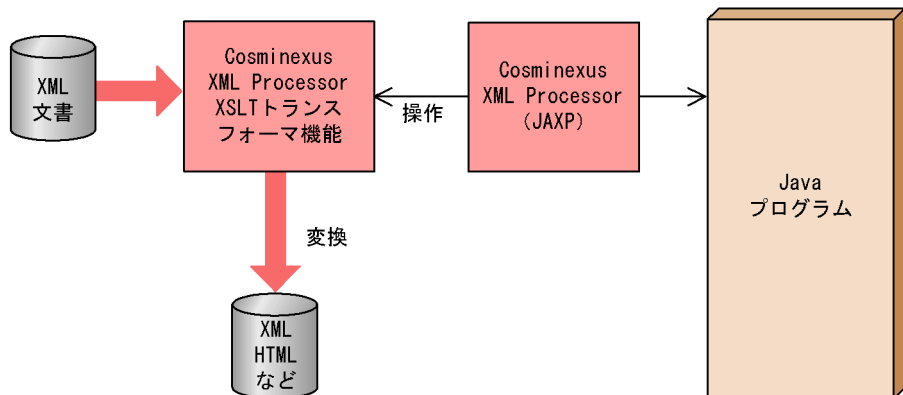
### 1.1.3 Java プログラムで XML 文書を容易に変換できる

Cosminexus XML Processor は JAXP の実装を提供しているため、Java プログラムで XSLT トランスフォーマを使用できます。

XSLT は XML 文書の構造変換を行うための仕様です。XSLT を実装したものを XSLT トランスフォーマと呼びます。XSLT トランスフォーマは XML 文書を読み込み、加工して出力します。XML 文書だけでなく、HTML、テキストも出力できます。

JAXP で XML 文書を変換する流れを図 1-2 に示します。JAXP を使用すると、Java プログラムで XSLT トランスフォーマの XML 文書の変換処理を操作できます。

図 1-2 JAXP で XML 文書を変換する流れ



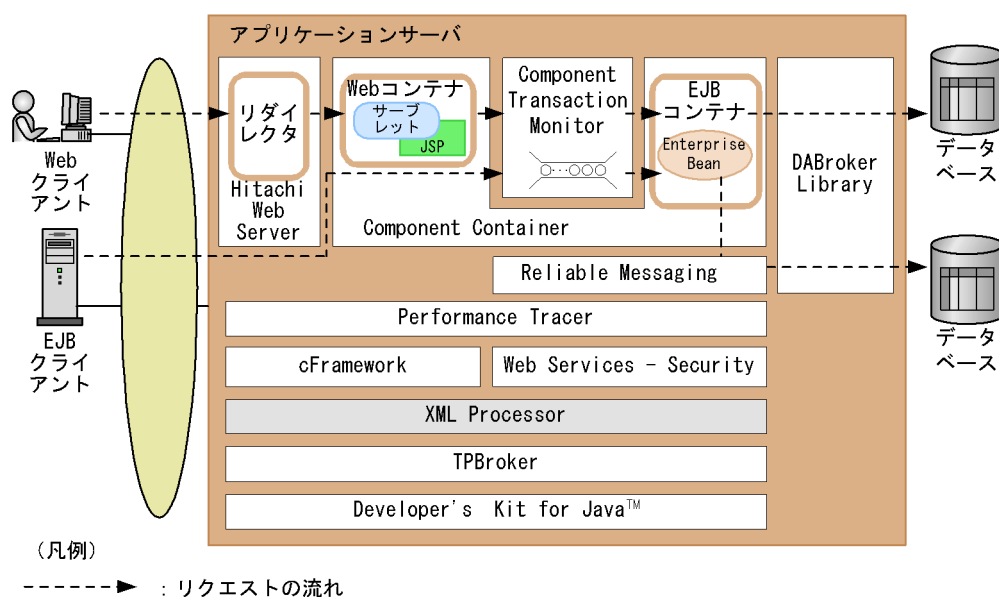
## 1.2 Cosminexus XML Processor の位置づけ

この節では、Cosminexus Version 7 環境での Cosminexus XML Processor の位置づけを説明します。

Cosminexus XML Processor は、Cosminexus Version 7 で XML 文書の解析と XSLT 変換などを必要とする局面で使用します。また、Cosminexus Version 7 のほかの製品と組み合わせることで、SOAP 通信基盤などの、昨今必要とされている XML 基盤のアプリケーション統合を可能にする環境を構築できます。

Cosminexus の実行環境での Cosminexus XML Processor の位置づけを図 1-3 に示します。

図 1-3 Cosminexus 実行環境での Cosminexus XML Processor の位置づけ



注： 図中では、構成ソフトウェア名の「Cosminexus」は省略しています。

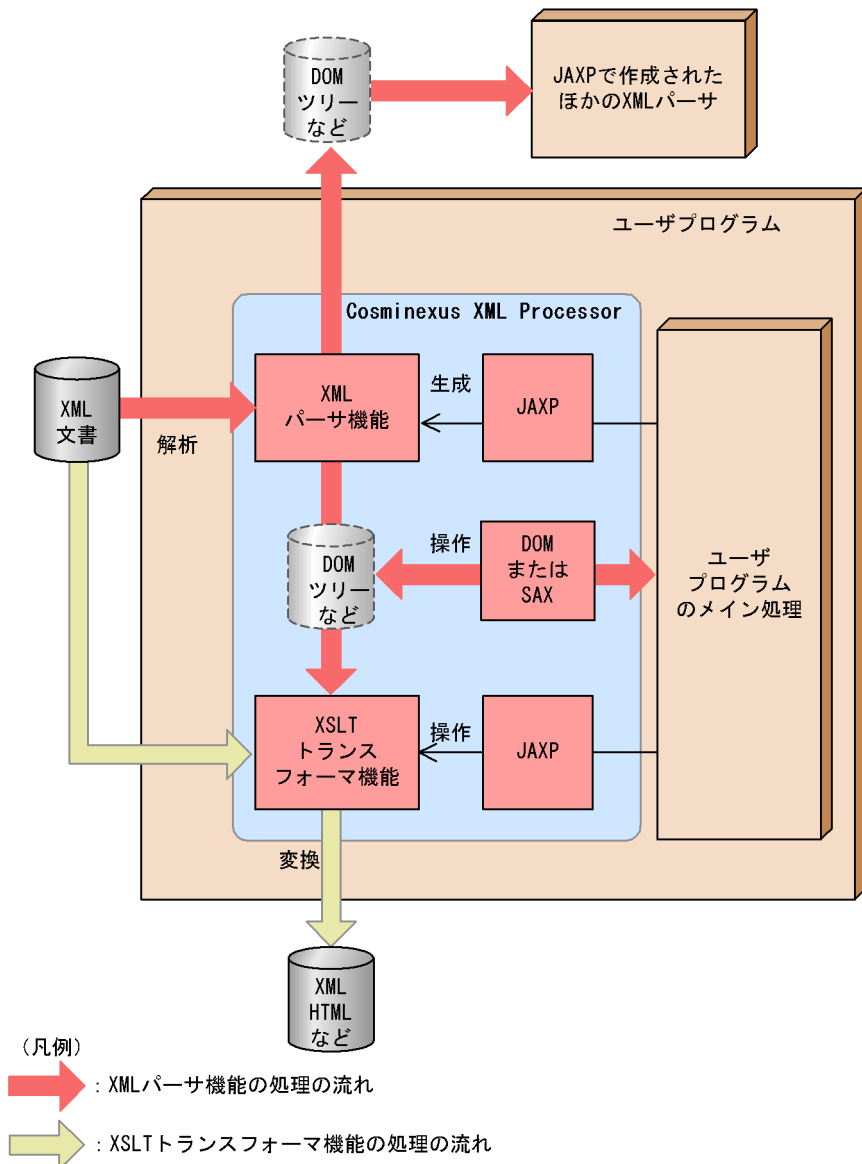


## 1.3 Cosminexus XML Processor の処理の流れ

この節では、Cosminexus XML Processor の処理の流れについて説明します。

Cosminexus XML Processor が提供する機能として、XML パーサ機能、XSLT トランスフォーム機能などがあります。Java で作成されたユーザプログラムで Cosminexus XML Processor を使用する場合の、XML パーサ機能、および XSLT トランスフォーム機能の処理の流れを次の図の例で示します。

図 1-4 XML パーサ機能および XSLT トランスフォーム機能の処理の流れ



## 1. Cosminexus XML Processor の概要

### XML パーサ機能の処理

JAXP によって生成された Cosminexus XML Processor の XML パーサ機能は、XML 文書を解析します。

ユーザプログラムは、解析された XML 文書を DOM API または SAX を使用して処理・操作できます。

また、解析された XML 文書は、JAXP に準拠したほかの XML パーサからも使用できます。

### XSLT トランスフォーマ機能の処理

JAXP によって生成された Cosminexus XML Processor の XSLT トランスフォーマ機能は、XML 文書（XML パーサで解析された XML 文書を含む）を規則に従った XML 文書、HTML などに変換できます。

## 1.4 Cosminexus XML Processor のサポート範囲

この節では、Cosminexus XML Processor のサポート範囲について説明します。

### 1.4.1 使用できる XML パーサ、XSLT トランスフォーマなどの機能

Cosminexus XML Processor のサポート範囲について説明します。

Cosminexus XML Processor は、JAXP1.3 の仕様をサポートしています。JAXP を介して使用される XML パーサ機能、XSLT トランスフォーマ機能などは、次に示す JAXP で定義された仕様の範囲内で使用できます。なお、JAXP1.3 は、従来の JAXP1.2 に対し、若干の仕様差がありますので、ご注意願います。

JAXP で定義された仕様

- JSR 206 Java API for XML Processing (JAXP) 1.3
- Extensible Markup Language (XML)
  - Extensible Markup Language (XML) 1.1 [<http://www.w3.org/TR/xml11/>]
  - Extensible Markup Language (XML) 1.0 (Second Edition) [<http://www.w3.org/TR/2000/REC-xml-20001006/>]
  - Extensible Markup Language (XML) 1.0 (Second Edition) Errata [<http://www.w3.org/XML/xml-V10-2e-errata>]
- Namespaces in XML
  - Namespaces in XML 1.1 [<http://www.w3.org/TR/xml-names11/>]
  - Namespaces in XML 1.0 [<http://www.w3.org/TR/REC-xml-names/>]
  - Namespaces in XML 1.0 Errata [<http://www.w3.org/XML/xml-names-19990114-errata>]
- XML Schema
  - XML Schema Part 1: Structures [<http://www.w3.org/TR/xmlschema-1/>]
  - XML Schema Part 1: Structures Errata [<http://www.w3.org/2001/05/xmlschema-errata#Errata1>]
  - XML Schema Part 2: Datatypes [<http://www.w3.org/TR/xmlschema-2/>]
  - XML Schema Part 2: Datatypes Errata [<http://www.w3.org/2001/05/xmlschema-errata#Errata2>]
- XSL Transformations (XSLT)
  - XSL Transformations (XSLT) Version 1.0 [<http://www.w3.org/TR/xslt>]
- XML Path Language (XPath)
  - XML Path Language (XPath) Version 1.0 [<http://www.w3.org/TR/xpath>]

## 1. Cosminexus XML Processor の概要

- XML Path Language (XPath) Version 1.0 Errata [<http://www.w3.org/1999/11/REC-xpath-19991116-errata>]
- XML Inclusions (XInclude)
  - XML Inclusions (XInclude) Version 1.0 [<http://www.w3.org/TR/xinclude/>]
- Document Object Model (DOM) Level 3
  - Document Object Model (DOM) Level 3 Core [<http://www.w3.org/TR/DOM-Level-3-Core>]
  - Document Object Model (DOM) Level 3 Load and Save [<http://www.w3.org/TR/DOM-Level-3-LS>]
- Simple API for XML (SAX)
  - Simple API for XML (SAX) 2.0.2 (sax2r3) [<http://sax.sourceforge.net/>]
  - Simple API for XML (SAX) 2.0.2 (sax2r3) Extensions [<http://sax.sourceforge.net/sax2-ext.html>]

### 1.4.2 処理できる文字コード

XML 文書の文字コードは、XML パーサや XSLT トランスフォーマの入出力 XML 文書に記述する XML 宣言の encoding 属性 (encoding="XXX" の XXX の部分) に指定します。このほかにも、XML 文書の入力ソースとなる InputSource オブジェクトのエンコーディングに指定する方法などがあります。

Cosminexus XML Processor では、IANA に登録された文字コードのうち次のものを処理できます。文字コードの定義については、IANA の CHARACTER SETS に関するドキュメントを参照してください。

- UTF-8
- UTF-16
- UTF-16BE
- UTF-16LE
- Shift\_JIS <sup>1</sup>
- Windows-31J
- ISO-2022-JP
- EUC-JP
- US-ASCII
- ISO-10646-UCS-2
- ISO-10646-UCS-4 <sup>2</sup>
- ISO-8859-1

注 1

Shift\_JIS を使用するときには適用される文字エンコーディング (SJIS または MS932) の切り替えについては、「1.4.3 Shift\_JIS 切り替え機能」を参照してください

さい。

注 2

バイトオーダーが big-endian または little-endian であり、かつ、Byte-Order-Mark が付加されていない形式だけをサポートします。

**!** 注意事項

上記以外の文字コードを使用した場合の動作は保証しません。

### 1.4.3 Shift\_JIS 切り替え機能

Shift\_JIS 切り替え機能は、XML 文書の encoding 属性に Shift\_JIS が指定されている場合に、Cosminexus XML Processor が適用する文字エンコーディングを切り替える機能です。文字エンコーディングは、XML パーサや XSLT トランスフォーマのインスタンスごとに切り替えられます。

XML 文書の encoding に指定されている文字エンコーディングは、Cosminexus XML Processor が提供する Shift\_JIS 切り替え機能での指定に従って、SJIS (x-sjis-jdk1.1.7) または MS932 (x-sjis-cp932) に切り替えられます。XML 文書の encoding 指定と適用される文字エンコーディングとの対応については、「3.1 Shift\_JIS 切り替え機能」の表 3-1 を参照してください。

Shift\_JIS 切り替え機能の使用方法については、「3.1 Shift\_JIS 切り替え機能」を参照してください。

### 1.4.4 XSLTC トランスフォーマ機能

XSLTC トランスフォーマは、Cosminexus XML Processor が提供している XSLT トランスフォーマと同じ機能を持ちながら、XML 文書の変換処理の性能を向上させたものです。Cosminexus XML Processor では、XSLT トランスフォーマと XSLTC トランスフォーマの両方を使用したアプリケーションの開発が可能です。

XSLTC トランスフォーマ機能の使用方法については、「3.2.4 XSLTC トランスフォーマの使用法」を参照してください。

### 1.4.5 高速パース機能

高速パース機能とは、事前解析によって解析対象の XML 文書の特徴を学習し、以降の XML 文書解析時に学習結果を利用することで、解析処理の実行速度を向上させる機能です。

高速パース機能の使用法については、「3.3 高速パース機能」を参照してください。



# 2

## JAXP の機能

この章では、JAXP を使用してプログラムを作成する際に知っておく必要のある JAXP の概要について説明します。

---

### 2.1 JAXP とは

---

### 2.2 JAXP が規定するパッケージとその機能

---

## 2.1 JAXP とは

JAXP ( Java API for XML Processing ) は、米国 Sun Microsystems, Inc. が提供する Java 言語用の標準 XML API です。

Cosminexus XML Processor が提供する JAXP の機能を表 2-1 に示します。

表 2-1 Cosminexus XML Processor が提供する JAXP の機能

機能の名称	機能の概要
DOM	XML 文書を解析して DOM ツリーを生成します。また、生成した DOM ツリーを操作します。
SAX	XML 文書を解析して SAX のイベントを発生させます。また、発生したイベントを処理します。
XSLT	XML 文書を入力し、スタイルシートに基づいて変換し、ほかの XML 文書、HTML、テキストとして出力します。
XPath	XPath 式を評価します。
Validation	XML 文書をスキーマ文書に基づいて検証します。
Datatype	W3C XML Schema 1.0 で規定された日付 / 時刻型データを処理します。

JAXP には、DOM パーサ、SAX パーサ、XSLT トランスフォーマ、XPath オブジェクト、Validation オブジェクト、および Datatype オブジェクトを生成する API が含まれています。これらの API は JAXP の実装に依存しないので、XML 文書进行处理・操作するプログラムを作成する際に、使用する XML プロセッサの実装を意識する必要がありません。そのため、異なる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。



## 2.2 JAXP が規定するパッケージとその機能

JAXP は、表 2-1 に示した機能を利用するためのパッケージを規定しています。JAXP が規定するパッケージに含まれる API の概要を表 2-2 に示します。

表 2-2 JAXP が規定するパッケージに含まれる API の概要

パッケージの名称	パッケージに含まれる API の概要
javax.xml.parsers	DOM パーサおよび SAX パーサでの解析を行うための API。
javax.xml.transform	XSLT トランスフォーマのための API。
javax.xml.transform.dom	XSLT トランスフォーマの入出力に DOM を使用するための API。
javax.xml.transform.sax	XSLT トランスフォーマの入出力に SAX を使用するための API。
javax.xml.transform.stream	XSLT トランスフォーマの入出力にストリームを使用するための API。
javax.xml.validation	XML 文書を検証するための API。
javax.xml.xpath	XPath 式を評価するための API。
javax.xml.namespace	XML 名前空間を処理するための API。
javax.xml.datatype	W3C XML Schema 1.0 で規定された日付 / 時刻型データを処理するための API。
javax.xml	XML 関連の文字列定数。
org.w3c.dom	DOM ツリーを操作する API。
org.w3c.dom.bootstrap	DOM 実装を取得するための API。
org.w3c.dom.ls	XML 文書をロードおよび保存するための API。
org.w3c.dom.events	DOM のイベント処理のための API。
org.xml.sax	SAX の基本処理のための API。
org.xml.sax.ext	SAX の拡張処理のための API。
org.xml.sax.helpers	SAX の処理のためのヘルパークラス群。

以降では、それぞれのパッケージの概要について説明します。

## 2.2.1 javax.xml.parsers パッケージ

javax.xml.parsers パッケージは、DOM パーサ、および SAX パーサでの解析を行うための API を提供するパッケージです。

DOM パーサでの解析を行うアプリケーションの処理の流れを図 2-1 に、SAX パーサでの解析を行うアプリケーションの処理の流れを図 2-2 に示します。

図 2-1 DOM パーサでの解析を行うアプリケーションの処理の流れ

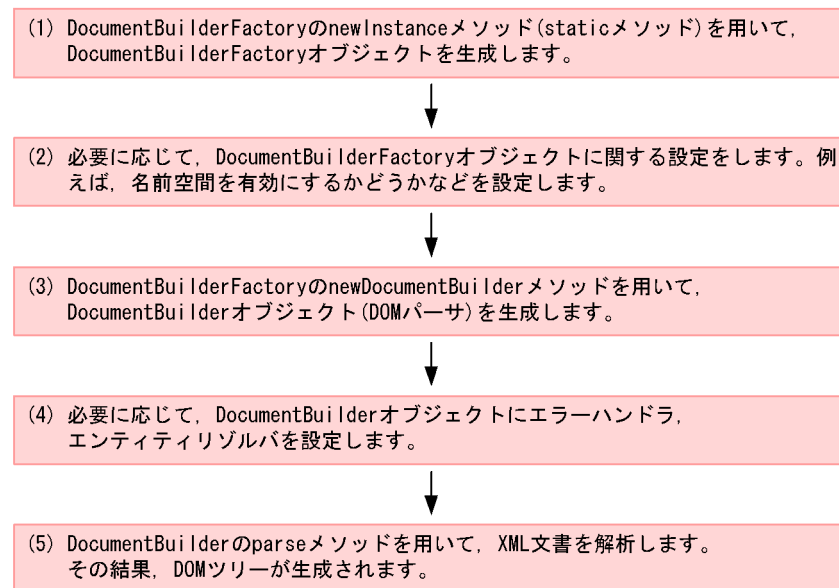
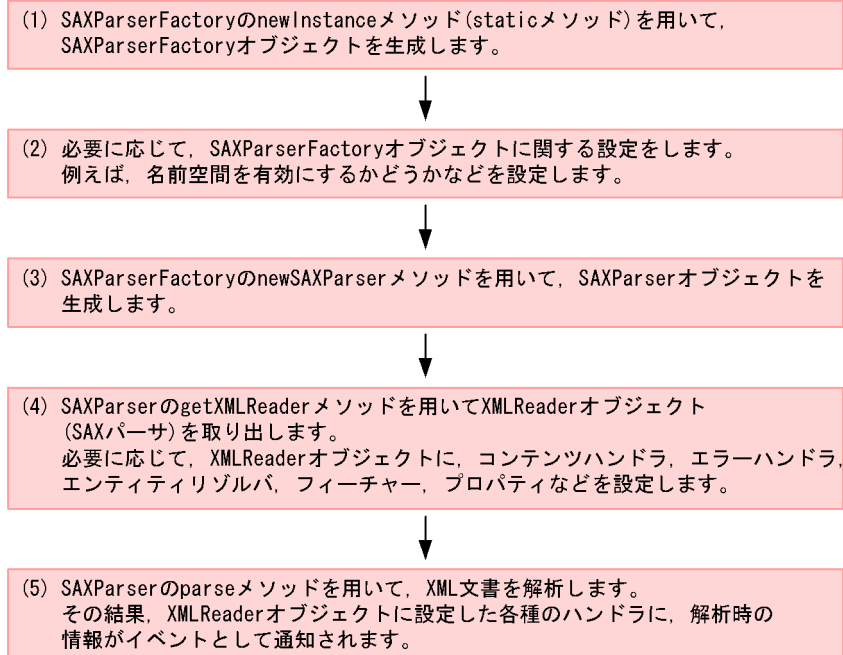


図 2-2 SAX パーサでの解析を行うアプリケーションの処理の流れ



上記のほかに、XMLReaderFactory クラスを用いて SAX パーサを生成する方法なども可能です。

javax.xml.parsers パッケージの詳細については、JSR 206 Java API for XML Processing(JAXP) 1.3 の Chapter 7. Package javax.xml.parsers を参照してください。

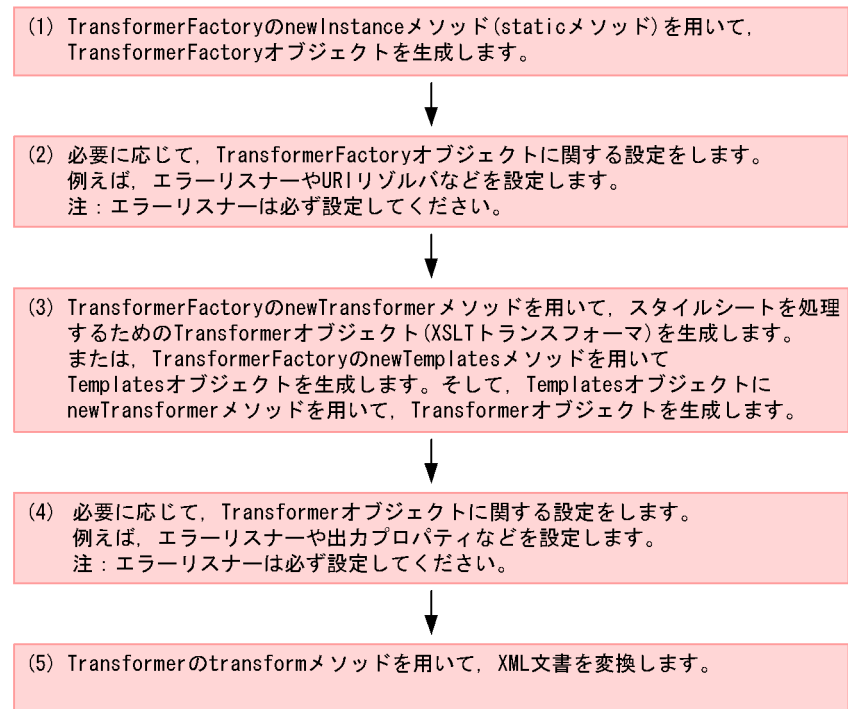
DOM パーサでの解析を行うコーディング例については、「5.4 DOM パーサを使用するサンプルプログラム」を、SAX パーサでの解析を行うコーディング例については、「5.5 SAX パーサを使用するサンプルプログラム」を参照してください。

## 2.2.2 javax.xml.transform パッケージ

javax.xml.transform パッケージは、XSLT トランスフォーマのための API を提供するパッケージです。

XSLT による変換を行うアプリケーションの処理の流れを図 2-3 に示します。

図 2-3 XSLT による変換を行うアプリケーションの処理の流れ



javax.xml.transform パッケージの詳細については、JSR 206 Java API for XML Processing(JAXP) 1.3 の Chapter 8. Package javax.xml.transform を参照してください。

XSLT による変換を行うコーディング例については、「5.7 XSLT トランスフォーマを使用するサンプルプログラム」を参照してください。

## 2.2.3 javax.xml.transform.dom パッケージ

javax.xml.transform.dom パッケージは、XSLT トランスフォーマの入出力に DOM を使用するための API を提供するパッケージです。

javax.xml.transform.dom パッケージの詳細については、JSR 206 Java API for XML Processing(JAXP) 1.3 の Chapter 9. Package javax.xml.transform.dom を参照してください。

## 2.2.4 javax.xml.transform.sax パッケージ

`javax.xml.transform.sax` パッケージは、XSLT トランスフォーマの入出力に SAX を使用するための API を提供するパッケージです。

`javax.xml.transform.sax` パッケージの詳細については、JSR 206 Java API for XML Processing(JAXP) 1.3 の Chapter 10. Package `javax.xml.transform.sax` を参照してください。

## 2.2.5 javax.xml.transform.stream パッケージ

`javax.xml.transform.stream` パッケージは、XSLT トランスフォーマの入出力にストリームを使用するための API を提供するパッケージです。

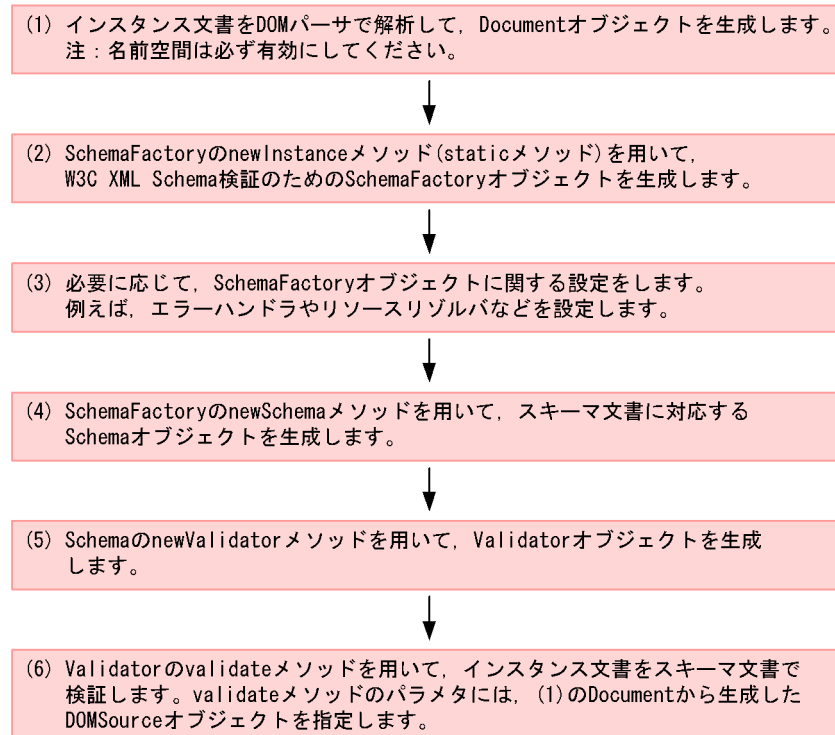
`javax.xml.transform.stream` パッケージの詳細については、JSR 206 Java API for XML Processing(JAXP) 1.3 の Chapter 11. Package `javax.xml.transform.stream` を参照してください。

## 2.2.6 javax.xml.validation パッケージ

javax.xml.validation パッケージは、XML 文書を検証するための API を提供するパッケージです。

javax.xml.validation パッケージに含まれるクラスを用いて XML 文書を検証するアプリケーションの処理の流れを図 2-4 に示します。

図 2-4 javax.xml.validation パッケージに含まれるクラスを用いて XML 文書を検証するアプリケーションの処理の流れ



上記のほかに、DOM パーサでの解析と検証を同時に行うことも可能です。また、SAX パーサでの解析と検証を同時に行うことも可能です。

javax.xml.validation パッケージの詳細については、JSR 206 Java API for XML Processing(JAXP) 1.3 の Chapter 14. Package javax.xml.validation を参照してください。

## 2.2.7 javax.xml.xpath パッケージ

javax.xml.xpath パッケージは、XPath 式を評価するための API を提供するパッケージです。

XPath 式を評価するアプリケーションの処理の流れを図 2-5 に示します。

図 2-5 XPath 式を評価するアプリケーションの処理の流れ

(1) インスタンス文書をDOMパーサで解析して、Documentオブジェクトを生成します。



(2) XPathFactoryのnewInstanceメソッド(staticメソッド)を用いて、XPathFactoryオブジェクトを生成します。



(3) 必要に応じて、XPathFactoryオブジェクトに関する設定をします。  
例えば、名前空間コンテキストや変数リゾルバなどを設定します。



(4) XPathFactoryのnewXPathメソッドを用いて、XPathオブジェクトを生成します。



(5) XPathのevaluateメソッドを用いて、XPath式の評価結果を取得します。  
evaluateメソッドのパラメータには、(1)で生成したDocumentオブジェクトとXPath式を指定します。

または、XPathのcompileメソッドを用いてXPathExpressionオブジェクトを生成します。そして、XPathExpressionオブジェクトにevaluateメソッドを適用して、XPath式の評価結果を取得します。

上記のほかに、InputStream オブジェクトを入力として XPath 式を評価できます。

javax.xml.xpath パッケージの詳細については、JSR 206 Java API for XML Processing(JAXP) 1.3 の Chapter 13. Package javax.xml.xpath を参照してください。

## 2.2.8 javax.xml.namespace パッケージ

javax.xml.namespace パッケージは、XML 名前空間を処理するための API を提供するパッケージです。このパッケージは、javax.xml.xpath パッケージおよび javax.xml.datatype パッケージに含まれる API の引数や戻り値としてだけ使用します。

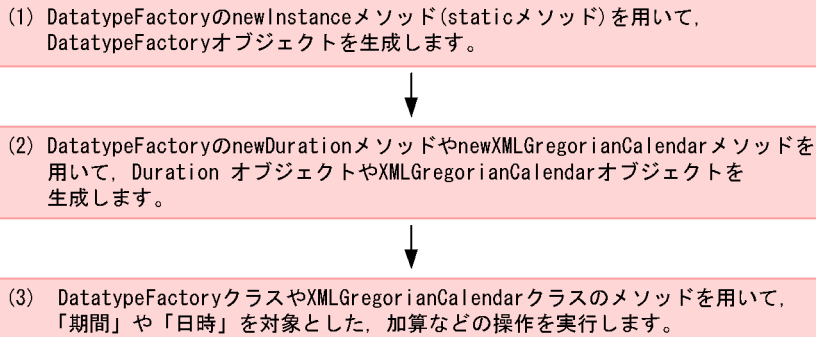
javax.xml.namespace パッケージの詳細については、JSR 206 Java API for XML Processing(JAXP) 1.3 の Chapter 12. Package javax.xml.namespace を参照してください。

## 2.2.9 javax.xml.datatype パッケージ

javax.xml.datatype パッケージは、W3C XML Schema 1.0 で規定された日付 / 時刻型データを処理するための API を提供するパッケージです。

日付 / 時刻型データを処理するアプリケーションの処理の流れを図 2-6 に示します。

図 2-6 日付 / 時刻型データを処理するアプリケーションの処理の流れ



javax.xml.datatype パッケージの詳細については、JSR 206 Java API for XML Processing(JAXP) 1.3 の Chapter 15. Package javax.xml.datatype を参照してください。

## 2.2.10 javax.xml パッケージ

javax.xml パッケージは、XML 関連の文字列定数を定義したパッケージです。

javax.xml パッケージの詳細については、JSR 206 Java API for XML Processing(JAXP) 1.3 の Chapter 16. Package javax.xml を参照してください。



## 2.2.11 org.w3c.dom パッケージ

org.w3c.dom パッケージは、DOM ツリーを操作する API を提供するパッケージです。

DOM ツリーを操作するアプリケーションの処理の流れを図 2-7 に示します。

図 2-7 DOM ツリーを操作するアプリケーションの処理の流れ

(1) インスタンス文書をDOMパーサで解析して、Documentオブジェクトを生成します。



(2) 操作対象のノードを取得します。NodeインタフェースやNodeのサブインタフェース (Element, Attrなど) の、次のような機能を持つメソッドを用います。

- ・ 所定のタグ名やID属性を持つ要素 (Elementノード) を取得する。
- ・ 要素 (Elementノード) の属性 (Attrノード) を取得する。
- ・ ノードの名前、ノードの値を取得する。
- ・ ノードの子ノード、兄弟ノード、親ノードを取得する、etc



(3) 操作対象のノードに対して、次のような機能を持つメソッドを適用することで、アプリケーションに必要な処理を行います。

- ・ 要素 (Elementノード) に属性 (Attrノード) を追加、削除する。
- ・ ノードを追加、削除、置換、移動、比較、複製、インポートする。
- ・ ノードの値を設定する。
- ・ ノードにユーザデータを付加する。
- ・ ノードの子孫テキストノードを取得、置換する、etc

org.w3c.dom パッケージの詳細については、W3C Document Object Model (DOM) Level 3 Core Specification の次に示す個所を参照してください。

- ・ 1.4 Fundamental Interfaces: Core Module
- ・ 1.5 Extended Interfaces: XML Module
- ・ Appendix G: Java Language Binding G.2 Other Core interfaces

DOM ツリーの操作を行うコーディング例は、「5.4 DOM パーサを使用するサンプルプログラム」を参照してください。

## 2.2.12 org.w3c.dom.bootstrap パッケージ

org.w3c.dom.bootstrap パッケージは、DOM 実装を取得するための API を提供するパッケージです。

org.w3c.dom.bootstrap パッケージの詳細については、W3C Document Object Model (DOM) Level 3 Core Specification の次に示す個所を参照してください。

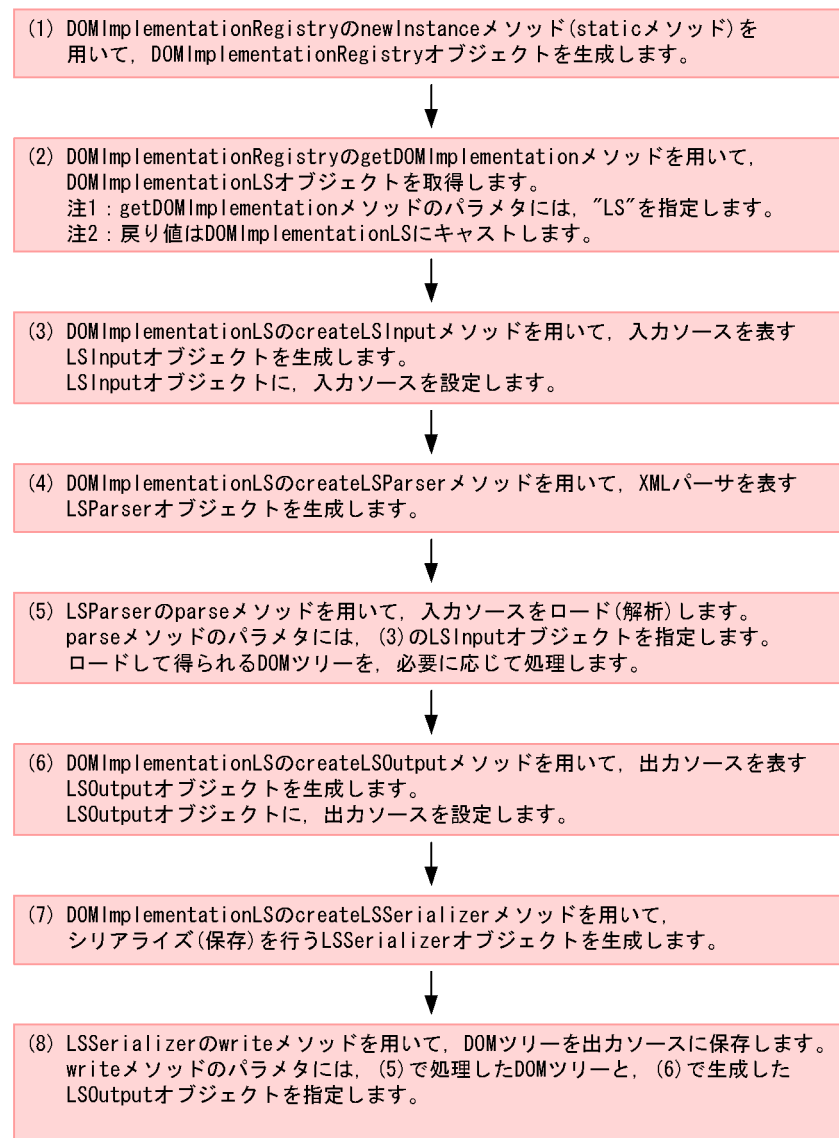
- ・ 1.3.7 Bootstrapping
- ・ Appendix G: Java Language Binding G.1 Java Binding Extension

## 2.2.13 org.w3c.dom.ls パッケージ

org.w3c.dom.ls パッケージは、XML 文書をロードおよび保存するための API を提供するパッケージです。

XML 文書のロード、および保存を行うアプリケーションの処理の流れを図 2-8 に示します。

図 2-8 XML 文書のロード、および保存を行うアプリケーションの処理の流れ



org.w3c.dom.ls パッケージの詳細については、W3C Document Object Model (DOM) Level 3 Load and Save Specification の次に示す個所を参照してください。

- 1.3 Fundamental Interfaces
- Appendix B: Java Language Binding

## 2.2.14 org.w3c.dom.events パッケージ

org.w3c.dom.events パッケージは、org.w3c.dom.ls.LSParser インタフェースを実装した非同期 LSParser オブジェクトが使用します。Cosminexus XML Processor は非同期 LSParser オブジェクトをサポートしないので、このパッケージは使用しません。

org.w3c.dom.events パッケージの詳細については、W3C Document Object Model (DOM) Level 3 Events Specification の次に示す個所を参照してください。

- 1.6 Basic interfaces
- 1.7 Event module definitions
- Appendix D: Java Language Binding

## 2.2.15 org.xml.sax パッケージ

org.xml.sax パッケージは、SAX の基本処理のための API を提供するパッケージです。

SAX イベントを処理するアプリケーションの処理の流れを図 2-9 に示します。

図 2-9 SAX イベントを処理するアプリケーションの処理の流れ

(1) XMLReader オブジェクトに設定した各種のハンドラに、解析時の情報がイベントとして通知されます。



(2) ハンドラが通知する次のようなイベントに対して、アプリケーション側で必要な処理を行います。

- コンテンツハンドラ (ContentHandler)
  - ・XML 文書の開始、終了イベント
  - ・要素の開始、終了イベント
  - ・名前空間バインディングの開始、終了イベント、etc
- DTD ハンドラ (DTDHandler)
  - ・記法宣言の出現イベント、etc
- DTD 拡張ハンドラ (DeclHandler)
  - ・要素型宣言、属性型宣言の出現イベント、etc
- 字句ハンドラ (LexicalHandler)
  - ・CDATA セクションの開始、終了、コメントの出現イベント、etc
- エンティティリゾルバ (EntityResolver)
  - ・エンティティの解決要求イベント
- エラーハンドラ (ErrorHandler)
  - ・warning, error, fatalError 発生イベント

org.xml.sax パッケージの詳細については、Simple API for XML (SAX) 2.0.2 (sax2r3) の Javadoc を参照してください。

SAX イベントを処理するコーディング例は、「5.5 SAX パーサを使用するサンプルプロ

グラム」を参照してください。

### 2.2.16 org.xml.sax.ext パッケージ

org.xml.sax.ext パッケージは、SAX の拡張処理のための API を提供するパッケージです。

org.xml.sax.ext パッケージの詳細については、Simple API for XML (SAX) 2.0.2 (sax2r3) Extensions の Javadoc を参照してください。

### 2.2.17 org.xml.sax.helpers パッケージ

org.xml.sax.helpers パッケージは、SAX の処理のためのヘルパークラス群を提供するパッケージです。

org.xml.sax.ext パッケージの詳細については、Simple API for XML (SAX) 2.0.2 (sax2r3) Extensions の Javadoc を参照してください。

# 3

## Cosminexus XML Processor の拡張機能

この章では，Cosminexus XML Processor の拡張機能について説明します。

---

3.1 Shift\_JIS 切り替え機能

---

3.2 XSLTC トランスフォーマ機能

---

3.3 高速パース機能

---

3.4 XML Schema のエラーメッセージと W3C 仕様との対応

---

## 3.1 Shift\_JIS 切り替え機能

Shift\_JIS 切り替え機能は、XML 文書の encoding 属性に Shift\_JIS が指定されている場合に、Cosminexus XML Processor が適用する文字エンコーディングを SJIS (x-sjis-jdk1.1.7) または MS932 (x-sjis-cp932) に切り替える機能です。XML 文書の encoding 指定と適用される文字エンコーディングとの対応を表 3-1 に示します。

表 3-1 XML 文書の encoding 指定と適用される文字エンコーディングとの対応

XML 文書の encoding 指定	Shift_JIS 切り替え機能の 指定	適用される 文字エンコーディング
Shift_JIS	指定なし	SJIS (デフォルト)
	SJIS	SJIS
	MS932	MS932
Windows-31J	-	MS932

(凡例)

- : 指定できません。

Shift\_JIS 切り替え機能の指定は、XML パーサや XSLT トランスフォーマのインスタンスに特定のプロパティを設定することで行います。プロパティの設定方法については「4.4 Shift\_JIS 切り替え機能のプロパティの使用方法」を参照してください。

Shift\_JIS 切り替え機能によるエンコーディング指定は、次のものに対して適用されます。それ以外には適用されません。

- 入力 XML 文書の encoding 属性
- InputSource オブジェクトに設定されたエンコーディング
- スタイルシート内の xsl:output 要素の encoding 属性に "Shift\_JIS" を指定した場合、または、javax.xml.transform.Transformer クラスの setOutputProperty, setOutputProperties メソッドで "Shift\_JIS" エンコーディングを指定した場合の、出力文字ストリーム

ただし、次の場合は注意が必要です。

1. DOM パーサ、SAX パーサの入力ソースに InputSource オブジェクトを指定する場合  
InputSource オブジェクトに Shift\_JIS 以外のエンコーディングを設定したときには、Shift\_JIS 切り替え機能は無効になります。また、InputSource オブジェクトに Reader が設定されているときにも、Shift\_JIS 切り替え機能は無効になり、Reader のエンコーディングが適用されます。
2. XSLT または XSLTC トランスフォーマの入力ソースに StreamSource オブジェクトを指定する場合  
StreamSource オブジェクトに Reader が設定されているときには、Shift\_JIS 切り替

え機能は無効になり、Reader のエンコーディングが適用されます。

3. XSLT または XSLTC トランスフォーマの入力ソースに DOMSource オブジェクトを指定する場合

DOMSource が生成された時点で XML 文書は文字コード変換済みなので、Shift\_JIS 切り替え機能は無効になります。

Shift\_JIS 切り替え機能を有効にするためには、DOMSource に設定する DOM ノードの生成時に利用するパーサに対して、あらかじめ Shift\_JIS 切り替え機能を指定しておく必要があります。

4. XSLT または XSLTC トランスフォーマの入力ソースに SAXSource オブジェクトを指定する場合

SAXSource オブジェクトに InputSource オブジェクトを設定したときには、1. と同じように注意が必要です。また、SAXSource オブジェクトに XMLReader オブジェクトを設定したときには、XMLReader に指定された Shift\_JIS 切り替え機能が有効になります。

## 3.2 XSLTC トランスフォーマ機能

---

この節では、XSLTC トランスフォーマ機能について説明します。

### 3.2.1 XSLTC トランスフォーマの概要

XSLTC トランスフォーマは、Cosminexus XML Processor が提供している XSLT トランスフォーマと同じ機能を持ちながら、XML 文書の変換処理の性能を向上させたものです。

XSLTC トランスフォーマは、XSLT スタイルシートを解析するコンパイラと、変換処理を行う実行時プロセッサで構成されます。コンパイラは、XSLT スタイルシートを解析して、高速に変換するためのトランスレットと呼ばれる Java のバイトコードをメモリ上に生成します。実行時プロセッサは、トランスレットを実行することによって、高速な変換処理を実現します。

XSLTC トランスフォーマと XSLT トランスフォーマは、次に示す点で動作に差異がありますのでご注意ください。

1. XSLTC トランスフォーマは XSLT トランスフォーマよりもエラーチェックが簡略化されています。
2. XSLT1.0 仕様書で規定されていない部分で、XSLT トランスフォーマと XSLTC トランスフォーマの動作に差異がある場合があります。

XSLTC トランスフォーマを使用することで、XSLT トランスフォーマよりも高い変換処理性能を期待できるケースを次に示します。

#### (1) 同一の XSLT スタイルシートを使って複数の XML 文書を変換する場合

同一の XSLT スタイルシートを使って複数の XML 文書を変換する場合は、XSLT トランスフォーマ、XSLTC トランスフォーマの両方とも、次に示す 1. の方法よりも 2. の方法の方が高い変換性能を期待できます。

1. スタイルシートから Templates オブジェクトや Transformer オブジェクトを毎回生成し、これを用いて変換する。
2. スタイルシートから Templates オブジェクトや Transformer オブジェクトを一度だけ生成し、これを再利用して変換する。

XSLTC トランスフォーマで 2. の方法を用いると、時間の掛かるトランスレット作成処理は一度だけで済みます。その後、トランスレットを用いた高速な変換が繰り返し実行されるため、XSLT トランスフォーマで 2. の方法を用いるよりも高い性能が期待できます。



## 3.2.2 XSLTC トランスフォーマを使ったシステムの開発・運用

XSLTC トランスフォーマは、XSLT トランスフォーマと比べてエラーチェックが厳密ではありません。XSLTC トランスフォーマを使ったシステムを開発・運用する場合は、次に示すように XSLT トランスフォーマでの動作確認も実施してください。

1. XSLT トランスフォーマを使ってスタイルシートのエラーチェックを必ず実施してください。これによって、XSLTC トランスフォーマではチェックできないエラーをチェックすることができ、他システムとの相互運用などで発生する可能性があるエラーチェックの相違を事前にチェックできます。
2. XSLTC トランスフォーマによる変換結果が XSLT トランスフォーマのものと同じになることを確認することを推奨します。これによって、トランスフォーマ固有の動作が変換結果に影響を及ぼすかどうかを確認でき、他システムとの相互運用などで発生する可能性がある動作の差異を事前にチェックできます。

XSLT トランスフォーマと XSLTC トランスフォーマに関する以下の注意事項も参照してください。

- 6.5 XSLT/XSLTC 共通の注意事項
- 6.6 XSLT に関する注意事項
- 6.7 XSLTC に関する注意事項

## 3.2.3 XSLTC トランスフォーマで使用するクラス

XSLTC トランスフォーマでは、TransformerFactoryXSLTC クラスを使用します。

XSLTC トランスフォーマで使用するクラスについて説明します。

### (1) TransformerFactoryXSLTC クラス

#### (a) 説明

XSLTC トランスフォーマのための TransformerFactory インスタンスを作成するクラスです。

#### (b) パッケージ名およびクラス名

com.cosminexus.jaxp.xsltc.TransformerFactoryXSLTC

#### (c) 形式

```
public class TransformerFactoryXSLTC
```

#### (d) メソッド一覧

メソッド名	機能
newInstance	TransformerFactory の新しいインスタンスを作成する

(e) メソッド詳細

newInstance メソッド

説明

XSLTC トランスフォーマのための, TransformerFactory の新しいインスタンスを作成します。

形式

```
public static TransformerFactory newInstance()
```

パラメタ

なし

戻り値

TransformerFactory の新しいインスタンスを返却します。

例外

なし

### 3.2.4 XSLTC トランスフォーマの使用方法

XSLTC トランスフォーマを使用する場合, XSLT トランスフォーマとは TransformerFactory インスタンスの生成方法が異なります。XSLTC トランスフォーマでの TransformerFactory インスタンスの生成方法を次に示します。

```
javax.xml.transform.TransformerFactory factory =  
    com.cosminexus.jaxp.xsltc.TransformerFactoryXSLTC.newInstance();
```

その他の使用法は, XSLT トランスフォーマと同じです。

コード例を次に示します。

```

import javax.xml.transform.*;
import javax.xml.transform.stream.*;
import com.cosminexus.jaxp.xslt.*;
import com.cosminexus.jaxp.xslt.*;
import com.cosminexus.jaxp.xslt.*;
// XSLスタイルシートを設定
StreamSource style = new StreamSource("style.xml");
// TransformerFactoryインスタンスを生成
TransformerFactory factory = TransformerFactoryXSLTC.newInstance();
// Templatesオブジェクトを生成
Templates templates = factory.newTemplates(style);
// XSLTCトランスフォーマを生成
Transformer transformer = templates.newTransformer();
// 同一のXSLスタイルシートを使って複数のXML文書を変換
for (int i = 0; i < 10; i++) {
    // 入力元を設定
    StreamSource source = new StreamSource(args[i]);
    // 出力先を設定
    StreamResult result = new StreamResult("output_" + args[i]);
    // 変換実行
    transformer.transform(source, result);
}

```

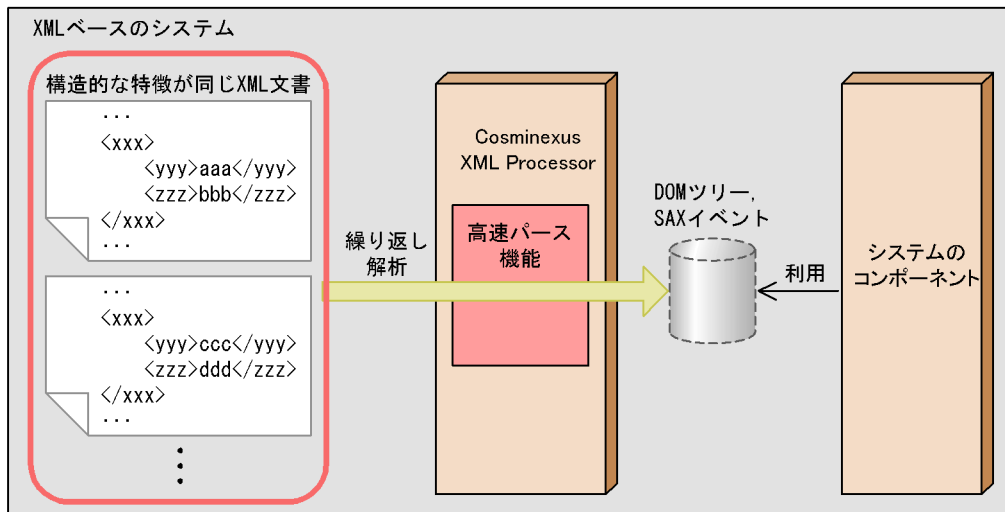
## 3.3 高速パース機能

この節では、高速パース機能について説明します。

### 3.3.1 高速パース機能の概要

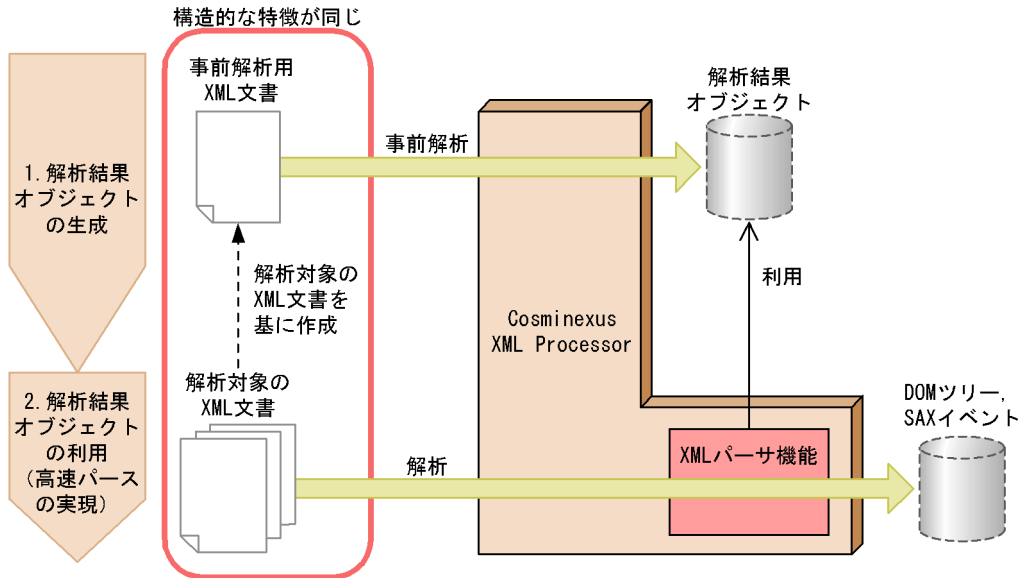
高速パース機能とは、事前解析によって解析対象の XML 文書の特徴を学習し、以降の XML 文書解析時に学習結果を利用することで、解析処理の実行速度を向上させる機能です。この機能は、要素や属性の並び順や入れ子関係などの構造的な特徴が一致している XML 文書を、繰り返し解析するシステムに適しています。高速パース機能を利用したシステムを次の図に示します。

図 3-1 高速パース機能を利用したシステム



事前解析によって学習した XML 文書の特徴は、解析結果オブジェクトに記録します。解析結果オブジェクトを生成して、高速パースを実現する流れを次の図に示します。

図 3-2 解析結果オブジェクトを生成して、高速パースを実現する流れ



#### 1. 解析結果オブジェクトの生成

解析結果オブジェクトは、Cosminexus XML Processor が、事前解析用 XML 文書から要素の並び順、属性の並び順、要素の繰り返しなどの情報を事前解析して生成します。事前解析用 XML 文書とは、解析対象の XML 文書と構造的な特徴が同じ XML 文書です。

事前解析用 XML 文書は、ユーザが作成する必要があります。XML 文書は一般的に自由な構造を持つため、すべての XML 文書に対応できる解析結果オブジェクトは生成できません。このため、解析対象の XML 文書を考慮して、出現頻度の高い文書構造に的を絞って事前解析用 XML 文書を作成します。

#### 2. 解析結果オブジェクトの利用（高速パースの実現）

解析結果オブジェクトを設定した Cosminexus XML Processor の XML パーサ機能で解析を実行すると、解析結果オブジェクトを利用して XML 文書を解析します。解析対象の XML 文書の構造が、解析結果オブジェクトが記録している文書の構造と一致していれば、解析速度が向上します。

図 3-2 に示した流れで高速パース機能を使用する場合、ユーザプログラム中に、解析結果オブジェクトを生成して XML パーサに設定するためのコードを記述する必要があります。ユーザプログラム中にコードを記述して高速パース機能を使用する方法の詳細は、「3.3.2 高速パースのための作業の流れ」を参照してください。

また、ユーザプログラム中にコードを記述しないで高速パース機能を使用することもできます。ユーザプログラム中にコードを記述しないで高速パース機能を使用する方法については、「3.3.12 高速化定義ファイルを使用した高速パース」を参照してください。

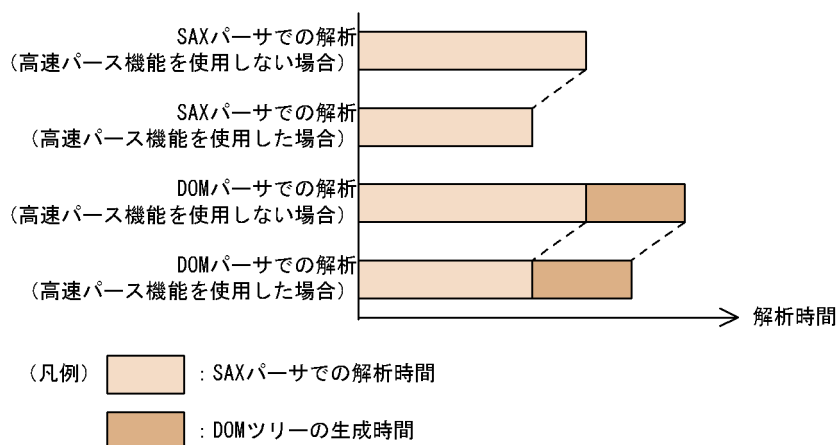
ただし、すべての場合について、通常よりも解析速度を向上できることを保証するものではありません。このため、高速パース機能を使用する場合は、実際に解析対象の XML 文書を使用して性能を評価することを推奨します。

なお、高速パース機能を使用する場合は、「6.17 高速パース機能に関する注意事項」の内容も確認してください。

#### 参考

高速パース機能を使用することで短縮できる解析時間の割合について

高速パース機能は、SAX パーサでの解析時間を短縮します。また、DOM パーサでの解析は、SAX パーサでの解析と DOM ツリーの生成で実装されているため、DOM パーサでの解析時間も短縮できます。ただし、高速パース機能を使用しても DOM ツリーの生成時間は変わりません。このため、次の図に示すように、SAX パーサでの解析と比べて、解析時間全体に対して短縮される時間の割合は小さくなります。

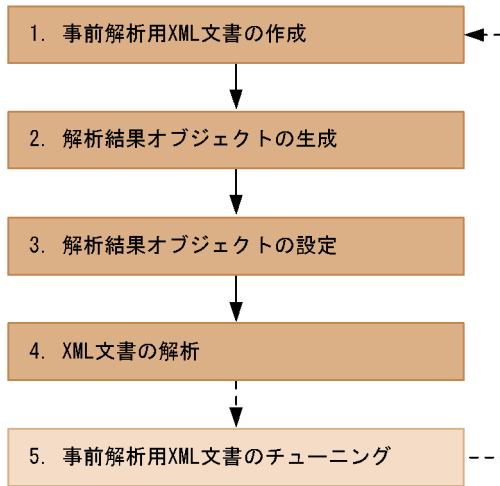


また、妥当性検証を行う場合は、解析時間に占める妥当性検証処理の割合が大きいため、短縮される解析時間の割合は相対的に小さくなります。特に、`javax.xml.validation.Schema` クラスを使わないで妥当性検証を行う場合は、解析時間はほとんど短縮されません。小さいサイズの XML 文書を解析する場合も、解析時間に占める XML パーサの初期化時間の割合が大きいため、短縮される解析時間の割合は相対的に小さくなります。

### 3.3.2 高速パースのための作業の流れ

ユーザプログラム中にコードを記述して高速パース機能を使用する場合の、ユーザが実施する作業の流れを次に示します。なお、ユーザプログラムを変更しないで高速パース機能を使用する方法については、「3.3.12 高速化定義ファイルを使用した高速パース」を参照してください。

図 3-3 高速パース機能を使用するための作業の流れ



(凡例)

→   : 必須の作業    - ->   : 任意の作業

各作業について説明します。

#### 1. 事前解析用 XML 文書の作成

解析対象の XML 文書の構造的な特徴を考慮して、事前解析用 XML 文書を作成します。事前解析用 XML 文書を作成するための指針については、「3.3.3 事前解析用 XML 文書の作成」を参照してください。

#### 2. 解析結果オブジェクトの生成

事前解析用 XML 文書を解析して、解析結果オブジェクトを生成します。解析結果オブジェクトの生成方法については、「3.3.4 解析結果オブジェクトの生成」を参照してください。

#### 3. 解析結果オブジェクトの設定

解析結果オブジェクトを Cosminexus XML Processor の XML パーサに設定します。解析結果オブジェクトの設定方法については、「3.3.5 解析結果オブジェクトの設定」を参照してください。

#### 4. XML 文書の解析

解析結果オブジェクトを設定した XML パーサの parse メソッドで、XML 文書を解析します。高速パース機能の対象となる parse メソッドについては、「3.3.6 XML 文書の解析」を参照してください。

#### 5. 事前解析用 XML 文書のチューニング

必要に応じて、事前解析用 XML 文書のチューニングを行います。チューニング情報を調査して事前解析用 XML 文書を最適化することで、解析速度をさらに向上できます。事前解析用 XML 文書のチューニング方法については、「3.3.9 事前解析用 XML

文書のチューニング」を参照してください。

なお、高速パース機能を使用するためにユーザプログラムに記述するコード例については、「3.3.8 高速パース機能を使用するためのコード例」を参照してください。

### 3.3.3 事前解析用 XML 文書の作成

事前解析用 XML 文書は、解析対象の XML 文書の構造的な特徴を考慮して作成します。事前解析用 XML 文書を作成する際は、次の指針に従ってください。

表 3-2 事前解析用 XML 文書を作成するための指針

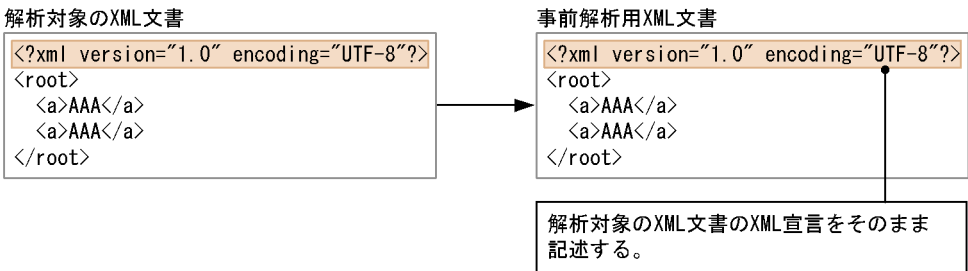
項番	適用箇所	指針
1	XML 宣言	解析対象の XML 文書と一致させる。
2	要素と属性	解析対象の XML 文書に記述されている要素と属性を記述する。
3	要素と属性の順序	解析対象の XML 文書と一致させる。
4	要素の入れ子構造	解析対象の XML 文書と一致させる。
5	要素の繰り返し構造	繰り返す可能性がある要素は、繰り返しの形で記述する。
6	テキスト、CDATA セクション、属性値	解析対象の XML 文書と一致させる必要はない。
7	DTD、コメント、処理命令	記述する必要はない。

それぞれの指針について、具体的な記述例を示して説明します。

#### (1) XML 宣言

解析対象の XML 文書にある XML 宣言を、そのまま事前解析用 XML 文書に記述してください。例を次の図に示します。

図 3-4 XML 宣言を記述する例（事前解析用 XML 文書の作成指針）



#### (2) 要素と属性

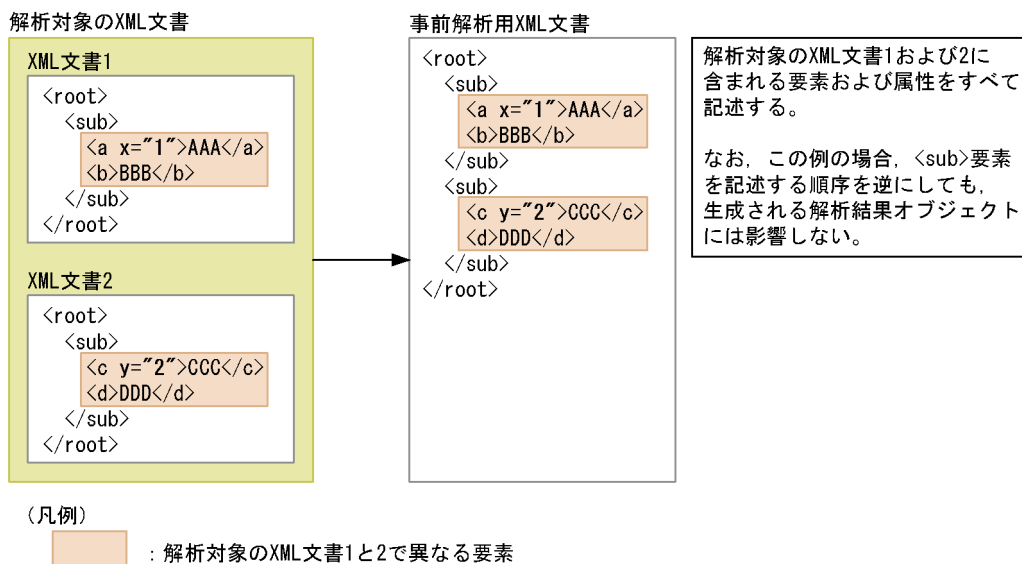
解析対象の XML 文書にある要素や属性を、事前解析用 XML 文書に記述します。事前解



析用 XML 文書に記述された要素や属性の情報は、生成される解析結果オブジェクトに記録されます。

解析対象の XML 文書間で異なる要素や属性がある場合は、すべての要素や属性を事前解析用 XML 文書に記述します。例を次の図に示します。

図 3-5 要素と属性を記述する例（事前解析用 XML 文書の作成指針）



ただし、要素中のテキスト、CDATA セクション、および属性値の内容については、解析対象の XML 文書と一致させる必要はありません。詳細は、「(6) テキスト、CDATA セクション、属性値」を参照してください。

なお、次の記述は XML 規格では区別されませんが、事前解析では異なる要素として区別されます。

- 空要素と空要素タグ（例：<a></a> と <a/>）
- 改行文字・空白文字・タブの数や位置が異なる要素

解析対象の XML 文書にこれらの要素がある場合は、事前解析用 XML 文書には区別して記述する必要があります。例を次の図に示します。

図 3-6 空要素と空要素タグを記述する例（事前解析用 XML 文書の作成指針）

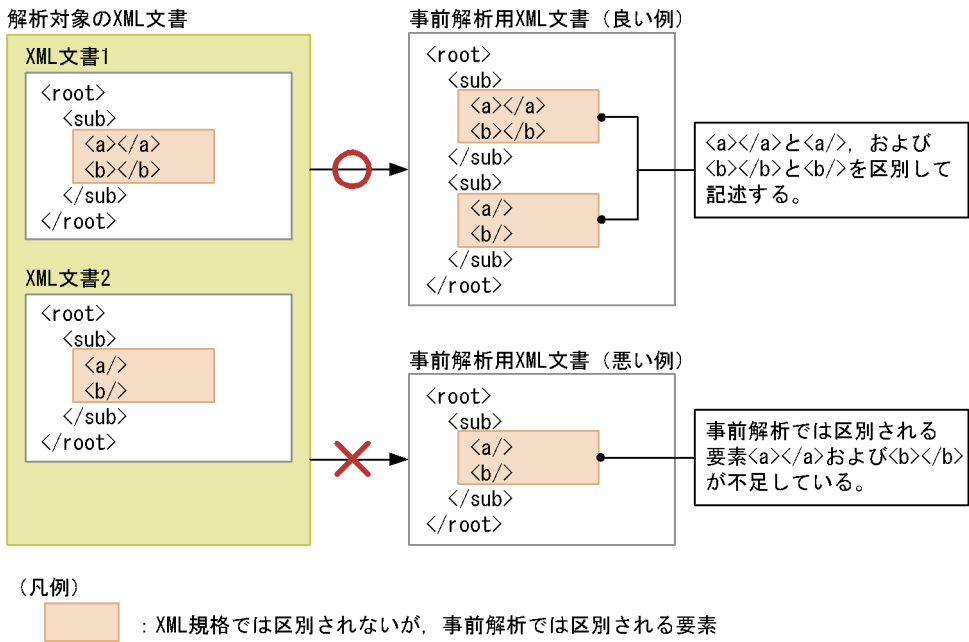
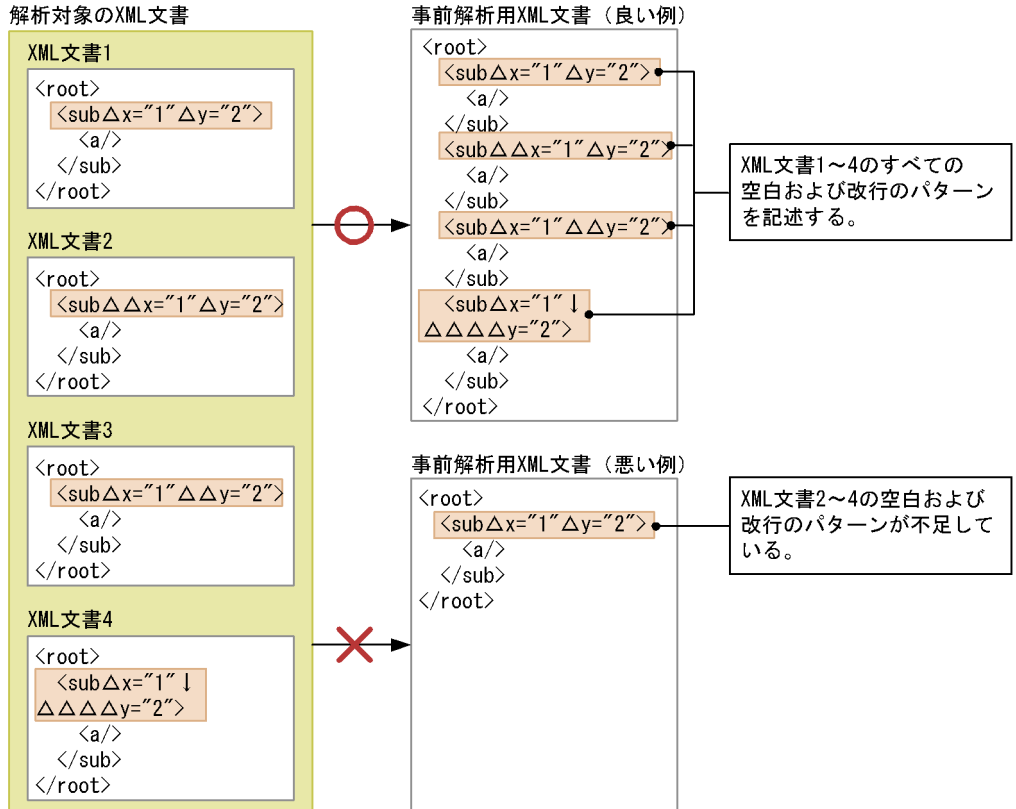


図 3-7 改行文字・空白文字の数や位置が異なる要素を記述する例（事前解析用 XML 文書の作成指針）



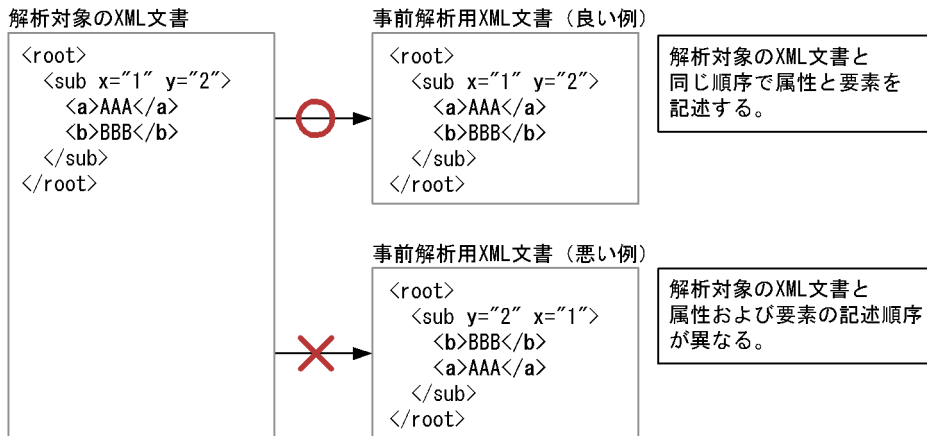
（凡例）

- △ : XML規格では区別されないが、事前解析では区別される要素
- △ : 空白
- ↓ : 改行

### （3）要素と属性の順序

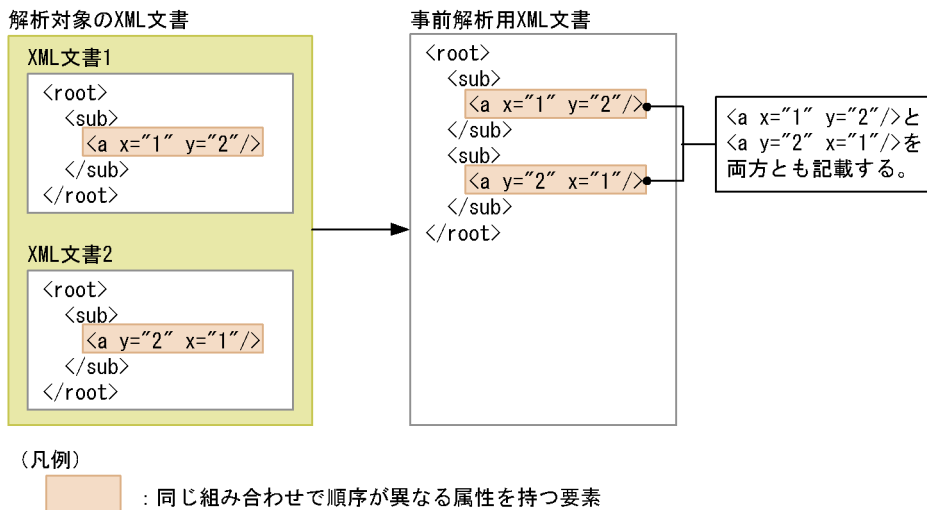
解析対象のXML文書にある要素や属性は、そのままの順序で事前解析用XML文書に記述します。例を次の図に示します。

図 3-8 要素と属性を記述する例（事前解析用 XML 文書の作成指針・要素と属性の順序）



解析対象の XML 文書に、同じ組み合わせで順序が異なる要素や属性が記述されている場合は、それらをすべて事前解析用 XML 文書に記述します。例を次の図に示します。

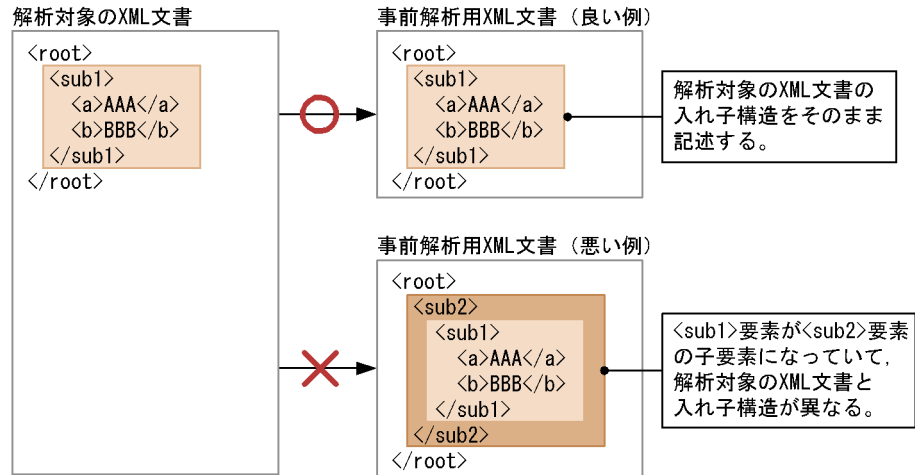
図 3-9 同じ組み合わせで順序が異なる属性を記述する例（事前解析用 XML 文書の作成指針）



#### (4) 要素の入れ子構造

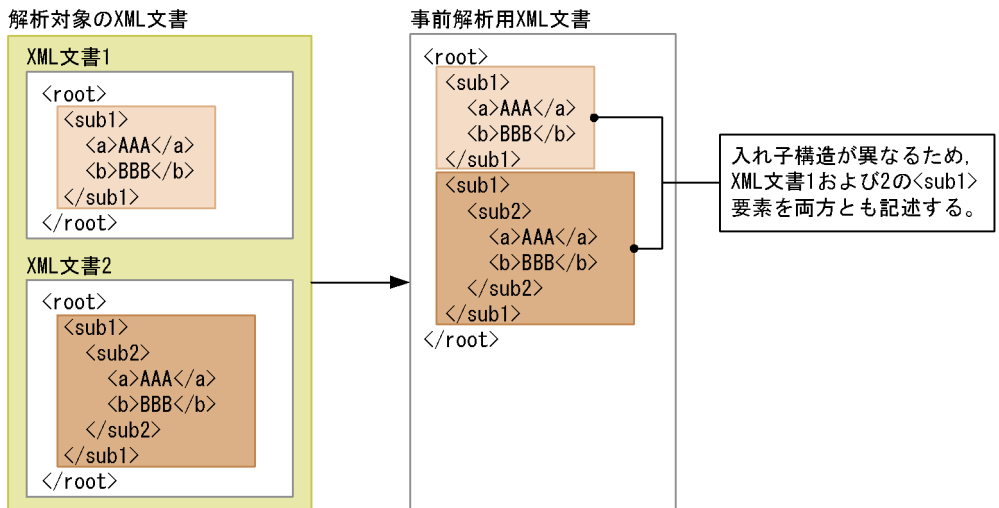
解析対象の XML 文書にある要素の入れ子構造は、そのままの形で事前解析用 XML 文書に記述します。例を次の図に示します。

図 3-10 要素の入れ子構造を記述する例（事前解析用 XML 文書の作成指針）



解析対象の XML 文書に異なる入れ子構造が複数ある場合は、すべての入れ子構造を事前解析用 XML 文書に記述します。例を次の図に示します。

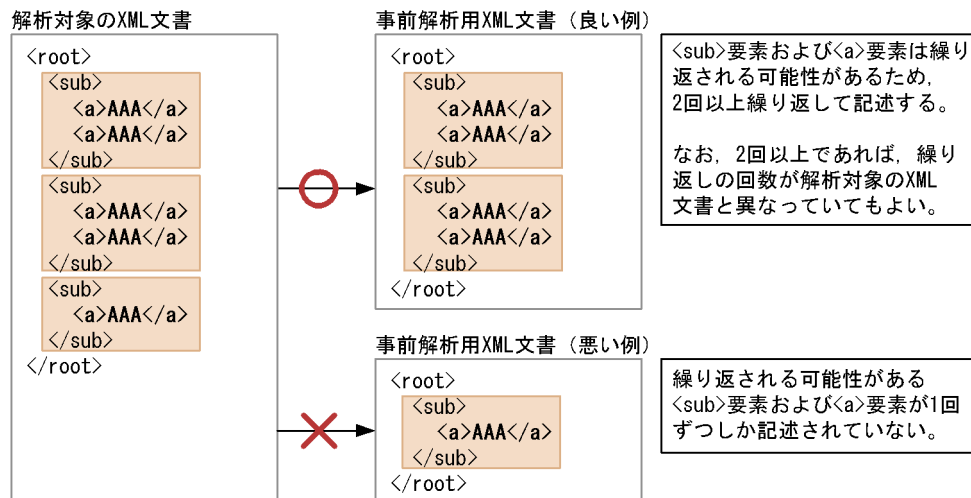
図 3-11 要素の入れ子構造を複数記述する例（事前解析用 XML 文書の作成指針）



## (5) 要素の繰り返し構造

解析対象の XML 文書に繰り返す可能性がある要素がある場合は、繰り返す可能性がある要素を、2 回以上連続で事前解析用 XML 文書に記述します。2 回以上連続で記述することで、解析時に、繰り返す可能性がある要素として扱われます。なお、事前解析用 XML 文書に連続して記述する回数が 2 回以上であれば、解析対象の XML 文書と繰り返しの回数が異なっても、高速パース機能の性能に影響はありません。例を次の図に示します。

図 3-12 要素の繰り返し構造を記述する例（事前解析用 XML 文書の作成指針）

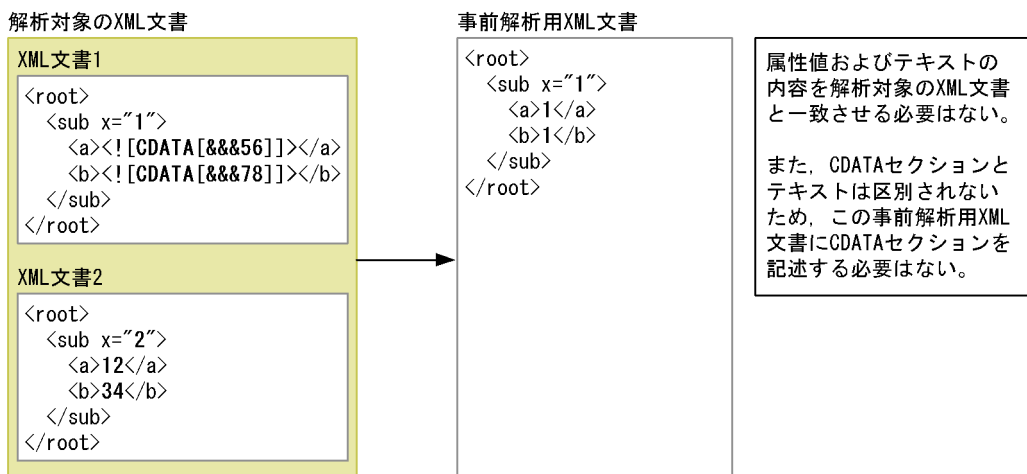


## （6）テキスト，CDATA セクション，属性値

事前解析用 XML 文書に記述されたテキスト，CDATA セクション，および属性値の内容は、生成される解析結果オブジェクトに影響を与えません。このため、事前解析用 XML 文書のテキスト，CDATA セクション，および属性値の内容を、解析対象の XML 文書と一致させる必要はありません。

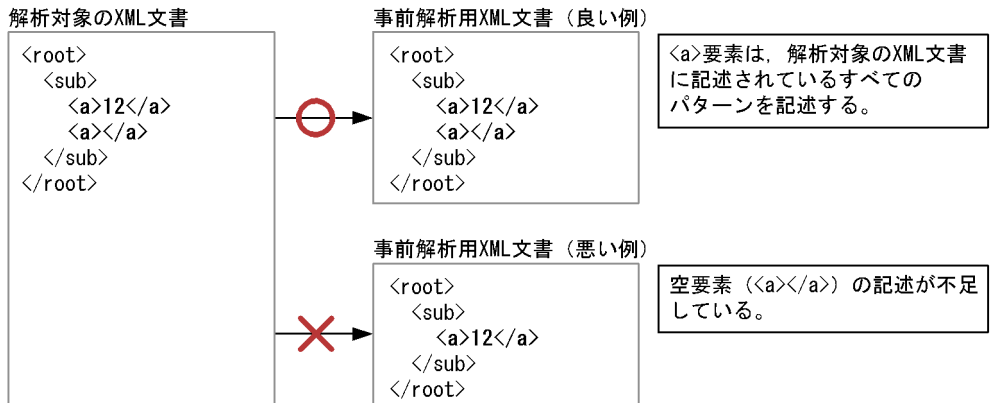
また、テキストと CDATA セクションは区別されません（どちらを記述しても、生成される解析結果オブジェクトは同じです）。例を次の図に示します。

図 3-13 テキスト，CDATA セクション，属性値を記述する例（事前解析用 XML 文書の作成指針）



内容を持つ要素（例：<a>12</a>）と空要素（例：<a></a>）は区別されます。また、改行、タブ、空白はテキストなので、これらのテキストを持つ要素（例：<a> [ 改行 ] </a>）と空要素（例：<a></a>）は区別されます。このため、解析対象の XML 文書にこのような要素が含まれる場合は、事前解析用 XML 文書にすべての要素を区別して記述する必要があります。例を次の図に示します。

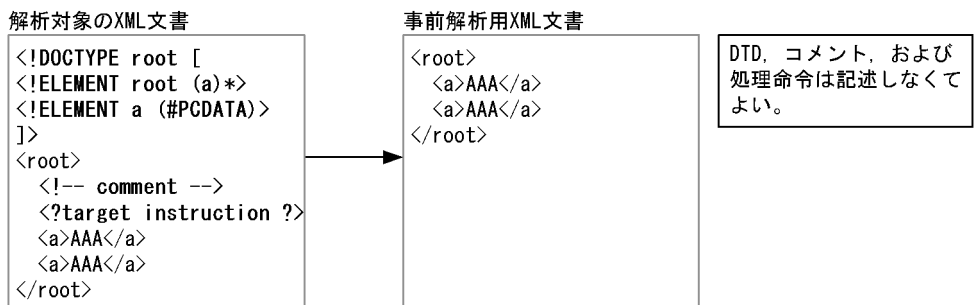
図 3-14 空要素を記述する例（事前解析用 XML 文書の作成指針）



## （7）DTD，コメント，処理命令

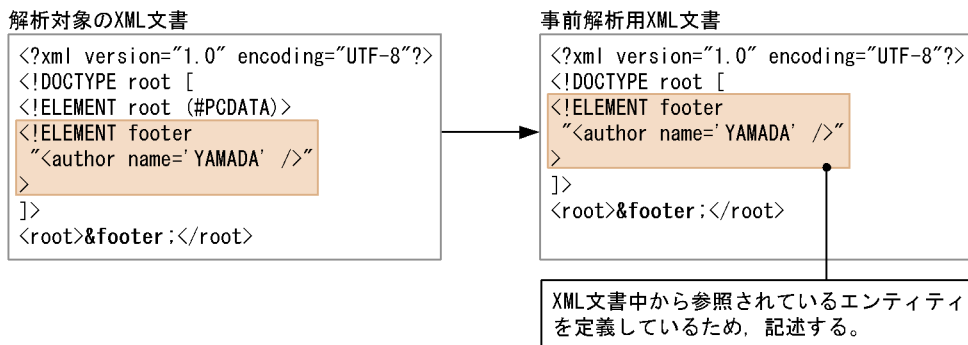
DTD，コメント，および処理命令は、解析結果オブジェクトには含まれません。このため、事前解析用 XML 文書にこれらを記述する必要はありません（記述しても生成される解析結果オブジェクトには影響しません）。例を次の図に示します。

図 3-15 DTD，コメント，処理命令に関する例（事前解析用 XML 文書の作成指針）



ただし、解析対象の XML 文書の中から参照されているエンティティを定義している DTD は、事前解析用 XML 文書に記述する必要があります。例を次の図に示します。

図 3-16 XML 文書中から参照されているエンティティの定義を記述する例（事前解析用 XML 文書の作成指針）



（凡例）

: XML文書中から参照されているエンティティを定義しているDTD

### 3.3.4 解析結果オブジェクトの生成

事前解析用 XML 文書を解析して、解析結果オブジェクトである `PreparsedObject` インスタンスを生成します。

解析結果オブジェクト（`PreparsedObject` インスタンス）を生成するには、次に示すメソッドを使用します。

- `com.cosminexus.jaxp.preparsedxml.PreparsedObjectFactory` クラスの `newInstance` メソッド
- `com.cosminexus.jaxp.preparsedxml.PreparsedObject` クラスの `newPreparsedObject` メソッド

メソッドの詳細は、「3.3.7 高速パース機能で使用するクラス」を参照してください。

解析結果オブジェクトを生成するコードの例を次に示します。

```
// 事前解析用XML文書を用意する
File xml = new File("learning1.xml");
// エンティティリゾルバを用意する
MyEntityResolver entityResolver = new MyEntityResolver();
// エラーハンドラを用意する
MyErrorHandler errorHandler = new MyErrorHandler();
// PreparsedObjectFactoryを生成する
PreparsedObjectFactory pof = PreparsedObjectFactory.newInstance();
// 名前空間を有効にする
pof.setNamespaceAware(true);
// エンティティリゾルバを設定する
pof.setEntityResolver(entityResolver);
// エラーハンドラを設定する
pof.setErrorHandler(errorHandler);
// 解析結果オブジェクトを生成する
PreparsedObject pobj = pof.newPreparsedObject(xml);
```



### 3.3.5 解析結果オブジェクトの設定

解析結果オブジェクトを Cosminexus XML Processor の XML パーサに設定します。解析結果オブジェクトは、次に示す XML パーサに設定できます。

- javax.xml.parsers.DocumentBuilder
- javax.xml.parsers.SAXParser
- org.xml.sax.XMLReader

解析結果オブジェクトを XML パーサに設定するには、<http://cosminexus.com/xml/properties/preparsedobject-load> プロパティを使用します。プロパティの詳細は「4.5 高速パース機能のプロパティの使用方法」を参照してください。

### 3.3.6 XML 文書の解析

解析結果オブジェクトを設定した XML パーサの parse メソッドで、XML 文書を解析します。高速パース機能の対象となる parse メソッドを次の表に示します。



#### 注意事項

表に記載のない parse メソッドを使用して高速パース機能を使用した場合の動作は保証されません。

表 3-3 高速パース機能の対象となる parse メソッド

クラス名	メソッド名
javax.xml.parsers.DocumentBuilder	parse(File f)
	parse(InputSource is)
	parse(InputStream is)
	parse(InputStream is, String systemId)
	parse(String uri)
javax.xml.parsers.SAXParser	parse(File f, DefaultHandler dh)
	parse(InputSource is, DefaultHandler dh)
	parse(InputStream is, DefaultHandler dh)
	parse(InputStream is, DefaultHandler dh, String systemId)
	parse(String uri, DefaultHandler dh)
org.xml.sax.XMLReader	parse(InputSource input)
	parse(String systemId)

### 3.3.7 高速パース機能で使用するクラス

高速パース機能を使用する場合、次に示すクラスを使用します。

- PreparedObjectFactory
- PreparedObject

各クラスについて説明します。

#### (1) PreparedObjectFactory クラス

##### (a) 説明

解析結果オブジェクトを生成するためのファクトリクラスです。

PreparedObjectFactory クラスは、スレッドセーフではありません。このため、複数のスレッドから同時に同じ PreparedObjectFactory にアクセスすることはできません。PreparedObjectFactory クラスは、スレッド間の競合を避けるため、次のどちらかの方法で使用してください。

- 各スレッドに一つの PreparedObjectFactory インスタンスを持つ。
- 各スレッドが排他的に PreparedObjectFactory インスタンスにアクセスする。

##### (b) パッケージ名およびクラス名

com.cosminexus.jaxp.preparedxml.PreparedObjectFactory

##### (c) 形式

```
public class PreparedObjectFactory
```

##### (d) メソッド一覧

メソッド名	機能
newInstance	解析結果オブジェクトを生成するためのファクトリを生成して返却する
setNamespaceAware	事前解析用 XML 文書を解析する際の名前空間の有効・無効を設定する
setErrorHandler	事前解析用 XML 文書の解析で発生するエラーを受け取るエラーハンドラを設定する
setEntityResolver	事前解析用 XML 文書に含まれ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を設定する
setTuningInfoFlag	チューニング情報を出力するかどうかを設定する
newPreparedObject	事前解析用 XML 文書を解析して、解析結果オブジェクトを生成する

##### (e) メソッド詳細

##### newInstance メソッド

##### 説明

解析結果オブジェクトを生成するためのファクトリを生成して返却します。

## 形式

```
public static PreparedObjectFactory newInstance()
```

## パラメタ

なし

## 戻り値

PreparedObjectFactory の新しいインスタンスを返却します。

## 例外

なし

## setNamespaceAware メソッド

## 説明

事前解析用 XML 文書を解析する際の名前空間の有効・無効を設定します。  
デフォルトでは、false が設定されています。

## 形式

```
public void setNamespaceAware(boolean awareness)
```

## パラメタ

- awareness  
名前空間の有効・無効を設定します。  
true : 名前空間を有効にする  
false : 名前空間を無効にする

## 戻り値

なし

## 例外

なし

## setErrorHandler メソッド

## 説明

newPreparedObject メソッドを呼び出す際に、事前解析用 XML 文書の解析で発生するエラーを受け取るエラーハンドラを設定します。  
デフォルトでは、次のエラーハンドラが設定されているように動作します。

```
class DefaultErrorHandler implements ErrorHandler {
    public void fatalError(SAXParseException e) throws SAXException {
        throw e;
    }
    public void error(SAXParseException e) throws SAXException {
        throw e;
    }
    public void warning(SAXParseException e) throws SAXException {
        // 何もしない
    }
}
```

## 形式

```
public void setErrorHandler(org.xml.sax.ErrorHandler  
errorHandler)
```

#### パラメタ

- errorHandler

エラーハンドラを設定します。null を設定した場合、デフォルトの動作になります。

#### 戻り値

なし

#### 例外

なし

### setEntityResolver メソッド

#### 説明

newPreparsedObject メソッドを呼び出す際に、事前解析用 XML 文書に含まれ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を設定します。デフォルトでは、次のエンティティリゾルバが設定されているように動作します。

```
class DefaultEntityResolver implements EntityResolver {  
    InputSource resolveEntity(String publicId, String systemId)  
    throws SAXException, IOException {  
        return null; // 常にnullを返す  
    }  
}
```

#### 形式

```
public void setEntityResolver(org.xml.sax.EntityResolver  
entityResolver)
```

#### パラメタ

- entityResolver

エンティティリゾルバを設定します。null を設定した場合、デフォルトの動作になります。

#### 戻り値

なし

#### 例外

なし

### setTuningInfoFlag メソッド

#### 説明

チューニング情報を出力するかどうかを設定します。

デフォルトでは、false が設定されています。

チューニング情報を出力する場合は、このメソッドで true を設定し、システムプロパティ「com.cosminexus.jaxp.preparsedxml.tuning.level」に INFO を設定する必要があります。チューニング情報の出力方法の詳細は、「3.3.9(3)

チューニング情報の出力方法」を参照してください。

#### 形式

```
public void setTuningInfoFlag(boolean state)
```

#### パラメタ

- state

チューニング情報を出力するかどうかを設定します。

true : チューニング情報を出力する

false : チューニング情報を出力しない

#### 戻り値

なし

#### 例外

なし

### newPreparedObject メソッド

#### 説明

事前解析用 XML 文書を解析して、解析結果オブジェクトを生成します。

#### 形式

```
public PreparedObject newPreparedObject(java.io.File xml)
throws IllegalArgumentException, SAXException, IOException
```

#### パラメタ

- xml

事前解析用 XML 文書を示す File オブジェクトを指定します。null は指定できません。

#### 戻り値

解析結果オブジェクトである PreparedObject の新しいインスタンスを返却します。

#### 例外

- IllegalArgumentException

xml パラメタに null が指定されているときに発生します。

- SAXException

次の場合に発生します。

- ・ 処理中に SAX エラーが発生したとき

- ・ XML1.1 文書を事前解析しようとしたとき（詳細は「6.17.3 XML1.1 に関する注意事項」を参照のこと）

- IOException

入出力エラーが発生したときに発生します。

## (2) PreparsedObject クラス

### (a) 説明

解析結果オブジェクトを示すクラスです。

### (b) パッケージ名およびクラス名

com.cosminexus.jaxp.preparsedxml.PreparsedObject

### (c) 形式

```
public class PreparsedObject
```

### (d) メソッド一覧

メソッド名	機能
isNamespaceAware	事前解析用 XML 文書を解析する際に設定された名前空間の有効・無効を返却する
getErrorHandler	事前解析用 XML 文書を解析する際に設定されたエラーハンドラを返却する
getEntityResolver	事前解析用 XML 文書を解析する際に設定されたエンティティリゾルバを返却する

### (e) メソッド詳細

#### isNamespaceAware メソッド

##### 説明

事前解析用 XML 文書を解析する際に設定された名前空間の有効・無効を返却します。

##### 形式

```
public boolean isNamespaceAware()
```

##### パラメタ

なし

##### 戻り値

true : 名前空間が有効である

false : 名前空間が無効である

##### 例外

なし

#### getErrorHandler メソッド

##### 説明

事前解析用 XML 文書を解析する際に設定されたエラーハンドラを返却します。

##### 形式

```
public ErrorHandler getErrorHandler()
```

パラメタ

なし

戻り値

事前解析用 XML 文書を解析する際に設定されたエラーハンドラを返却します。  
設定されていない場合は、null を返却します。

例外

なし

getEntityResolver メソッド

説明

事前解析用 XML 文書を解析する際に設定されたエンティティリゾルバを返却します。

形式

```
public EntityResolver getEntityResolver()
```

パラメタ

なし

戻り値

事前解析用 XML 文書を解析する際に設定されたエンティティリゾルバを返却します。設定されていない場合は、null を返却します。

例外

なし

### 3.3.8 高速パース機能を使用するためのコード例

高速パース機能を使用するために、ユーザプログラムに記述するコードの例を次に示します。

```

import java.io.File;
import java.io.IOException;
import javax.xml.parsers.SAXParserFactory;
import javax.xml.parsers.SAXParser;
import org.xml.sax.EntityResolver;
import org.xml.sax.ErrorHandler;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;
import com.cosminexus.jaxp.preparedxml.PreparedObjectFactory;
import com.cosminexus.jaxp.preparedxml.PreparedObject;

public class TestSAXParser{
    public static void main(String[] args){
        try{
            // 事前解析用XML文書から解析結果オブジェクトを生成する
            File xml = new File("learning1.xml");
            MyHandler handler = new MyHandler ();
            PreparedObjectFactory pof = PreparedObjectFactory.newInstance();
            pof.setNamespaceAware(true);
            pof.setEntityResolver(handler);
            pof.setErrorHandler(handler);
            PreparedObject pobj = pof.newPreparedObject(xml);

            // XMLパーサを生成する
            SAXParserFactory spf = SAXParserFactory.newInstance();
            spf.setNamespaceAware(true);
            SAXParser sp = spf.newSAXParser();

            // 解析結果オブジェクトをXMLパーサに設定して、解析を実行する
            sp.setProperty("http://cosminexus.com/xml/properties/
preparedobject-load", pobj);
            sp.parse("SampleSAX.xml", handler);
        } catch (IllegalArgumentException iae){
            System.out.println("MSG : " + iae.getMessage());
        } catch (SAXException se){
            System.out.println("MSG : " + se.getMessage());
        } catch (IOException ioe){
            System.out.println("MSG : " + ioe.getMessage());
        } catch (Exception e){
            e.printStackTrace();
        }
    }
}

class MyHandler extends DefaultHandler {
    //エラーハンドラとエンティティリゾルバの実装を記述
}

```

### 3.3.9 事前解析用 XML 文書のチューニング

事前解析用 XML 文書のチューニング情報を参照すると、高速パース機能が効果的に動作しているかどうかを調査できます。調査結果を基に事前解析用 XML 文書を最適化することで、解析速度をさらに向上できます。

事前解析用 XML 文書のチューニング情報の種類、利用方法、および出力方法について説明します。

#### (1) チューニング情報の種類

チューニング情報として出力できるファイルは、次の 2 種類です。

- チューニングサマリファイル
- チューニング詳細ファイル



これらのファイルには、解析時のマッチング処理についての情報が出力されます。マッチング処理とは、解析時に解析結果オブジェクトの情報を利用できたかどうかを判定する処理のことです。解析結果オブジェクトの情報を利用できた場合を「マッチング処理に成功した」、解析結果オブジェクトの情報を利用できなかった場合を「マッチング処理に失敗した」といいます。

各ファイルについて説明します。

#### (a) チューニングサマリファイル

高速パース機能を使用したそれぞれの解析について、マッチング処理の成功率、解析結果オブジェクトの生成に使用した事前解析用 XML 文書のファイル名、解析対象の XML 文書のシステム識別子、チューニング詳細ファイル名などが出力されます。高速パース機能を使用した解析処理で、どのくらい効果的に解析結果オブジェクトが利用されているかを調査できます。

チューニングサマリファイルの詳細は、「3.3.10 チューニングサマリファイルの詳細」を参照してください。

#### (b) チューニング詳細ファイル

高速パース機能を使用した一つの解析について、チューニングサマリファイルの情報に加え、マッチング処理の履歴が出力されます。高速パース機能を使用した解析処理で、マッチングに失敗している個所の詳細を調査できます。

チューニング詳細ファイルの詳細は、「3.3.11 チューニング詳細ファイルの詳細」を参照してください。

### (2) チューニング情報の利用方法

チューニング情報の利用方法を次に示します。

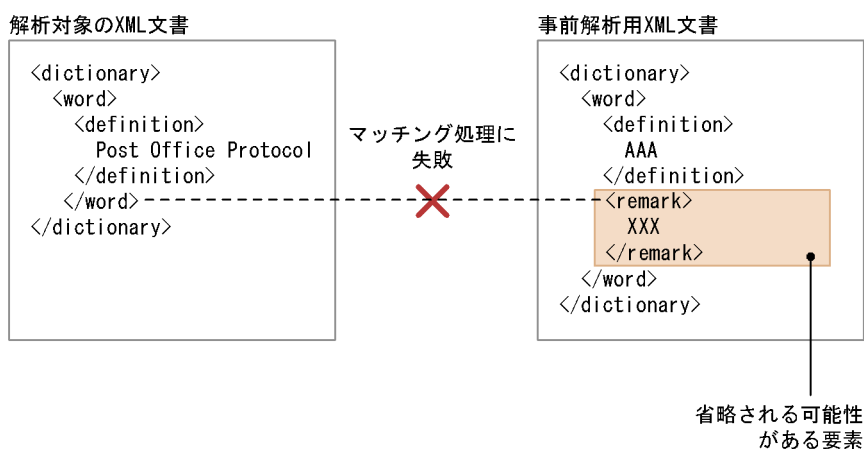
1. チューニング情報を出力する。  
チューニング情報の出力方法については、「(3) チューニング情報の出力方法」を参照してください。
2. チューニングサマリファイルを参照して、マッチング処理の成功率が低い解析を抽出する。  
マッチング処理の成功率が低いということは、解析結果オブジェクトが学習した事前解析用 XML 文書の構造と解析対象の XML 文書の構造の違いが大きく、高速パース機能が効果的に動作していないことを示します。
3. 2. で抽出した各解析について、チューニング詳細ファイルを参照して、マッチングに失敗した個所の履歴を調査する。  
マッチングに失敗した個所の履歴を調査して、事前解析用 XML 文書と解析対象の XML 文書の構造が不一致になっている部分を抽出します。
4. 事前解析用 XML 文書を修正する。  
3. の調査結果を基に、事前解析用 XML 文書を修正します。事前解析用 XML 文書を

作成するための指針については、「3.3.3 事前解析用 XML 文書の作成」を参照してください。

### ! 注意事項

解析対象の XML 文書に記述されているすべての要素や属性が事前解析用 XML 文書に含まれていても、マッチング処理の成功率が低い場合は、高速パース機能の効果は低くなります。このため、高速パース機能を適用するかどうかは、実際のシステムでの解析結果を見て判断してください。

解析対象の XML 文書に記述されているすべての要素や属性が事前解析用 XML 文書に含まれていても、マッチング処理の成功率が低下する場合の例を次に示します。



この例では、解析対象の XML 文書で remark 要素が省略されているため、remark 要素のマッチング処理は失敗し、チューニング詳細ファイルに失敗履歴が記録されます。この結果、マッチング処理の成功率が低下します。

## (3) チューニング情報の出力方法

チューニング情報を出力するには、Cosminexus XML Processor が規定するシステムプロパティを設定する必要があります。また、PreparsedObject を使用した高速パースでチューニング情報を出力する場合は、システムプロパティの設定に加え、PreparsedObjectFactory クラスの setTuningInfoFlag メソッドで true を設定して、PreparsedObject を生成する必要があります。PreparsedObjectFactory クラスの詳細は、「3.3.7(1) PreparsedObjectFactory クラス」を参照してください。

チューニング情報を出力するためのシステムプロパティ、チューニング情報の出力先、およびエラー発生時の動作について説明します。

**!** 注意事項

チューニング情報を出力した場合は、解析性能が劣化します。このため、チューニング情報は、高速パース機能の性能を評価するときだけ出力してください。

## (a) チューニング情報を出力するためのシステムプロパティ

## システムプロパティ名

`com.cosminexus.jaxp.preparsedxml.tuning.level`

## 値

- OFF またはこのプロパティを設定していない場合：チューニング情報を出力しない
- INFO：チューニング情報を出力する

大文字の INFO 以外が指定された場合は、OFF が指定されたものとして扱われます。

## (b) チューニング情報の出力先

チューニング情報は、高速パース機能を利用した解析の終了時に出力されます。チューニング情報の出力先は、次の順序で決定されます。

1. サーバのログ出力ディレクトリ `/jaxp`

注 「サーバのログ出力ディレクトリ」は、動作しているサーバによって異なります。

## J2EE サーバの場合

J2EE サーバのオプション定義ファイル (`usrconf.cfg`) の

`ejb.server.log.directory` キーに指定されているディレクトリです。デフォルトでは、次のディレクトリとなります。

`<Cosminexus 作業ディレクトリ >/ejb/< サーバ名称 >/logs`

## Web コンテナサーバの場合

Web コンテナサーバのオプション定義ファイル (`usrconf.cfg`) の

`web.server.log.directory` キーに指定されているディレクトリです。デフォルトでは、次のディレクトリとなります。

`<Cosminexus のインストールディレクトリ >/CC/web/containers/< サーバ名称 >/logs`

2. `user.dir` システムプロパティで取得できるディレクトリ `/logs/jaxp`

なお、同名のファイルは上書きされます。同名のファイルが存在する場合は、必要に応じて退避してください。ファイル名の規則については、「3.3.10(1) ファイル名 (チューニングサマリファイル)」および「3.3.11(1) ファイル名 (チューニング詳細ファイル)」を参照してください。

また、チューニングサマリファイルおよびチューニング詳細ファイルは自動的に削除されません。必要に応じて削除してください。チューニングサマリファイルのファイルサイズの上限值については、「3.3.10(3) ファイルサイズの上限值」を参照してください。チューニング詳細ファイルの合計ファイルサイズの上限值については、「3.3.11(3) 合計ファイルサイズの上限值」を参照してください。

### (c) エラー発生時の動作

エラーが発生してチューニング情報を出力できない場合の動作を次に示します。

#### 入出力エラーが発生した場合

チューニングサマリファイル、およびチューニング詳細ファイルを出力する際に入出力エラーが発生した場合は、これらのファイルは出力されません。また、IOException 例外が発生します。

#### XML 文書の解析中にエラーが発生した場合

チューニングサマリファイルには情報が出力されません。チューニング詳細ファイルには、エラーが発生する前までの情報が出力されます。

## 3.3.10 チューニングサマリファイルの詳細

チューニングサマリファイルのファイル名、出力形式、およびファイルサイズの上限值を次に示します。

### (1) ファイル名 (チューニングサマリファイル)

csmjaxp\_ppxml\_summary.csv

### (2) 出力形式 (チューニングサマリファイル)

出力行	出力内容
1 行目	チューニングサマリファイルであることを示すヘッダ Cosminexus XML Processor Fast Parse Function Tuning Summary Information
2 行目	チューニングサマリファイルが作成された日時
3 行目	4 行目以降の行ヘッダ matching rate,prepared XML,parsed XML,tuning detail file,date
4 行目以降	次の情報がコンマで区切られて出力される。 1. マッチング処理の成功率 成功回数 ÷ マッチング処理の実行回数 × 100 2. 事前解析用 XML ファイル名 3. 解析対象の XML 文書のシステム識別子 システム識別子が設定されていない場合は、null が出力されます。 4. チューニング詳細ファイルのファイル名 5. 解析が終了した日時

#### 注

出力文字列にコンマ(,)が含まれている場合も、そのまま出力されます。

## 出力例

1	Cosminexus XML Processor Fast Parse Function Tuning Summary Information
2	Create Date: Tue Feb 06 20:57:35 JST 2007
3	matching rate, prepared XML, parsed XML, tuning detail file, date
4	45.5, D:¥tr1.xml, file:/novelA.xml, csmjxap_ppxml_tr1_novelistA_00.txt, Tue Feb 06 20:57:35 JST 2007
5	90.0, D:¥tr1.xml, file:/novelB.xml, ...
6	97.5, D:¥tr2.xml, file:/comicA.xml, ...
7	85.0, D:¥tr2.xml, file:/comicB.xml, ...

## (3) ファイルサイズの上限值

1MB

## 注

1MB を超えた場合、それ以降のデータは追記しません。

なお、ファイルサイズの上限值は、次のシステムプロパティで変更できます。

システムプロパティ名

com.cosminexus.jaxp.preparedxml.tuning.summaryfile\_maxsize

値

0 以上の整数を指定します。単位は MB です。

このプロパティを設定していない場合、および範囲外の値が指定された場合は、1 が設定されます。

## 3.3.11 チューニング詳細ファイルの詳細

チューニング詳細ファイルのファイル名、出力形式、およびファイルサイズの上限値を次に示します。

## (1) ファイル名 (チューニング詳細ファイル)

csmjxap\_ppxml\_{事前解析用XML文書を表す文字列}\_{解析対象のXML文書を表す文字列}\_{連番}.txt

事前解析用 XML 文書を表す文字列

事前解析用 XML 文書の絶対パスの末尾部分から、最後に出現するピリオド(.)以降を除いたものです。

例：事前解析用 XML 文書の絶対パスが「D:¥home¥xml¥training¥tr1.xml」の場合

事前解析用 XML 文書を表す文字列は、「tr1」となります。

解析対象の XML 文書を表す文字列

解析対象となる XML 文書のシステム識別子の末尾部分から、最後に出現するピリオド以降を除いたものです。

例：解析対象の XML 文書のシステム ID が「file:/home/xml/data/

novelA.xml」の場合

解析対象の XML 文書を表す文字列は、「novelA」となります。

#### 連番

00 ~ 99 の連番です。解析ごとに増加します。

なお、J2EE サーバまたは Web コンテナを再起動すると「00」にリセットされます。また、「99」の次は「00」となります。

### (2) 出力形式 (チューニング詳細ファイル)

出力行	出力内容
1 行目	チューニング詳細ファイルであることを示すヘッダ Cosminexus XML Processor Fast Parse Function Tuning Detail Information
2 行目	チューニング詳細ファイルが作成された日時
3 行目	事前解析用 XML 文書のファイル名
4 行目	解析対象の XML 文書のシステム識別子 システム識別子が設定されていない場合は、null が出力されます。
5 行目	マッチング情報の開始を示すヘッダ Matching Information start
6 行目以降	マッチング情報 (マッチング処理の履歴) マッチング処理に成功している場合 o: { マッチング処理の対象とした要素・属性 } マッチング処理に失敗している場合 x: expected { マッチングを試みたパターン } but { 解析対象 XML 文書中に出現したパターン }
最終行から 2 行目	マッチング情報の終了を示すフッタ Matching Information end
最終行	マッチング処理の成功率 成功回数 ÷ マッチング処理の実行回数 × 100 (%) ただし、マッチング処理が 1 回も実行されなかった場合は、0.0% と出力されます

#### 注

要素または属性の一部だけが出力されることがあります。また、要素または属性の前後の文字が出力されることがあります。具体例は出力例を参照してください。

## 出力例

1	Cosminexus XML Processor Fast Parse Function Tuning Detail Information	
2	Create Date: Wed Feb 07 13:12:42 JST 2007	
3	Prepared XML: d:\home\xml\ttraining\ttraining1.xml	
4	Parsed XML: word.xml	
5	Matching Information start	
6	o:note>	
7	o:</note>	
8	o:<word>	要素<word>のマッチング処理に成功したことを示す。
9	o:<name is_acronym=	
10	o:>	
11	o:</name>	
12	o:<definition>	
13	o:</definition>	
14	o:<definition>	
15	o:</definition>	
16	x:expected <definition> but </word>	要素<definition>のマッチング処理に失敗したことを示す。
17	<w	
18	o:</word>	
19	o:<word>	
20	x:expected <name is_acronym= but <update date="200	
21	o:<update date=	
22	o:/>	
23	o:<name is_acronym=	
24	o:>	
25	o:</name>	
26	o:<definition>	
27	o:</definition>	
28	x:expected <definition> but </word>	要素の後ろの文字が出力されることがある。ここでは、</word>の後ろの改行文字と<w>が出力されている。
29	<w	
30	o:</word>	
31	o:<word>	
32	x:expected <name is_acronym= but <update date="200	
33	o:<update date=	
34	x:expected /> but e	
35	o: editor=	
36	o:/>	要素の一部だけが出力されることがある。ここでは、閉じタグを示す/>だけが出力されている。
37	o:<name>	
38	o:</name>	
39	o:<definition>	
40	o:</definition>	
41	x:expected <definition> but </word>	
42	</d	
43	o:</word>	
44	x:expected <word> but <abcdef	要素の一部だけが出力されることがある。ここでは、<abcdef以降の文字ghi/>が出力されていない。
45	x:expected not begin with a but a	文字aが出現したためマッチング処理に失敗したことを示す。
46	Matching Information end	
47	Matching rate: 78.4%	

## (3) 合計ファイルサイズの上限值

32MB

## 注

出力済みのチューニング詳細ファイルの合計ファイルサイズが 32MB を超えた場合、

それ以降はチューニング詳細ファイルが生成されません。

例えば、次の 1 ~ 5 の順にチューニング詳細ファイルが生成される場合、4 のファイルが生成された時点で 1 ~ 4 の合計ファイルサイズが 40MB となり、32MB を超えているため、5 のファイルは生成されません。

1. csmjaxp\_ppxml\_trl\_novelistA\_00.txt ( 10MB )
2. csmjaxp\_ppxml\_trl\_novelistB\_00.txt ( 10MB )
3. csmjaxp\_ppxml\_trl\_novelistC\_00.txt ( 10MB )
4. csmjaxp\_ppxml\_trl\_novelistD\_00.txt ( 10MB )
5. csmjaxp\_ppxml\_trl\_novelistE\_00.txt ( 10MB )

なお、合計ファイルサイズの上限值は、次のシステムプロパティで変更できます。

システムプロパティ名

`com.cosminexus.jaxp.preparsedxml.tuning.detailfile_totalmaxsize`

値

0 以上の整数を指定します。単位は MB です。

このプロパティを設定していない場合、および範囲外の値が指定された場合は、32 が設定されます。

## 3.3.12 高速化定義ファイルを使用した高速パース

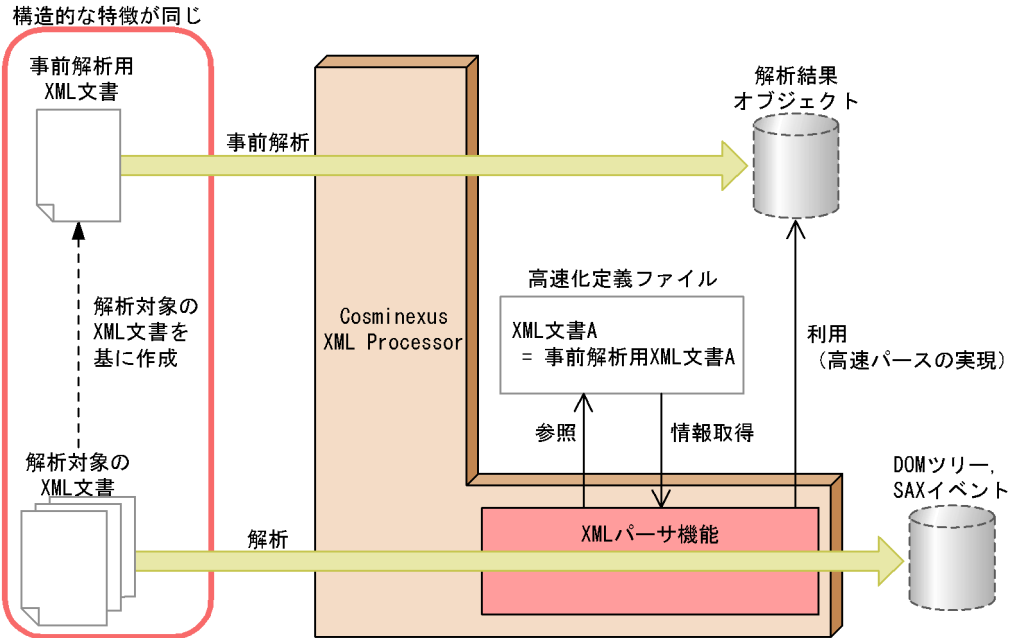
高速パース機能を使用する場合、通常は、ユーザプログラム中に高速パース機能用のコードを追加する必要があります。一方、ユーザプログラムを変更しないで高速パース機能を使用することもできます。

ユーザプログラムを変更しないで高速パース機能を使用するには、高速化定義ファイルを使用します。高速化定義ファイルとは、解析対象の XML 文書と事前解析用 XML 文書の対応や解析時に名前空間を有効にするかどうかなどを定義するファイルです。システムプロパティに高速化定義ファイルを設定しておくことで、高速化定義ファイルで指定された XML 文書の解析時に、高速パースを適用します。

高速化定義ファイルを使用した高速パースの処理概要を次に示します。



図 3-17 高速化定義ファイルを使用した高速パースの処理概要



### (1) 通常的高速パースと高速化定義ファイルを使用した高速パースの相違点

ユーザプログラム中に高速パース機能用のコードを追加する，通常的高速パースと高速化定義ファイルを使用した高速パースの相違点を次に示します。

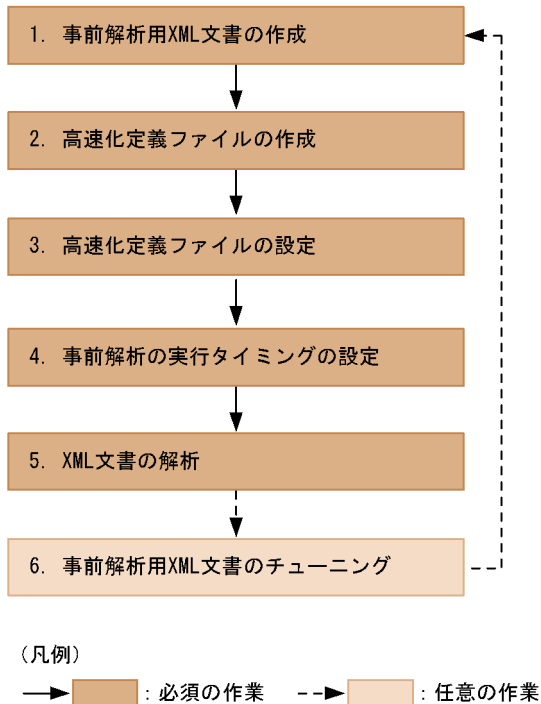
表 3-4 通常的高速パースと高速化定義ファイルを使用した高速パースの相違点

項番	相違点	通常的高速パース	高速化定義ファイルを使用した高速パース
1	高速パース機能を使用できない場合	なし。	解析対象の XML 文書にシステム識別子が設定されていない場合。
2	ユーザが実装したエラーハンドラ，エンティティリゾルバの使用	使用できる。	使用できない。
3	事前解析を実行するタイミング	ユーザプログラム内の任意のタイミング。	次のどちらかを選択する。 ・ J2EE サーバまたは Web コンテナの起動時 ・ 高速パースの対象となる各 XML 文書の初回解析時
4	高速パースを適用した解析を実行するタイミング	ユーザプログラム内の任意のタイミング。	高速化定義ファイルで解析対象として指定した XML 文書の解析時。
5	ユーザプログラムの変更	必要。	不要。

### 3.3.13 高速化定義ファイルを使用した高速パースのための作業の流れ

高速化定義ファイルを使用して高速パース機能を使用する場合の、ユーザが実施する作業の流れを次に示します。

図 3-18 高速化定義ファイルを使用して高速パース機能を使用するための作業の流れ



各作業について説明します。

#### 1. 事前解析用 XML 文書の作成

解析対象の XML 文書の構造的な特徴を考慮して、事前解析用 XML 文書を作成します。これは、通常の高速パースの場合と共通の手順です。事前解析用 XML 文書を作成するための指針については、「3.3.3 事前解析用 XML 文書の作成」を参照してください。

#### 2. 高速化定義ファイルの作成

高速パース機能の対象となる XML 文書と事前解析用 XML 文書の対応を定義した高速化定義ファイルを作成します。高速化定義ファイルの作成方法については、「3.3.14 高速化定義ファイルの作成」を参照してください。

#### 3. 高速化定義ファイルの設定

高速化定義ファイルを Cosminexus XML Processor が規定するシステムプロパティに設定します。高速化定義ファイルの設定方法については、「3.3.15 高速化定義ファイ

ルの設定」を参照してください。

#### 4. 事前解析の実行タイミングの設定

事前解析用 XML 文書を解析するタイミングを Cosminexus のシステムプロパティで設定します。事前解析の実行タイミングの設定については、「3.3.16 事前解析の実行タイミングの設定」を参照してください。

#### 5. XML 文書の解析

XML 文書を解析します。高速化定義ファイルを使用した高速パース機能の対象となる `parse` メソッドについては、「3.3.17 XML 文書の解析 (高速化定義ファイルを使用する場合)」を参照してください。

#### 6. 事前解析用 XML 文書のチューニング

必要に応じて、事前解析用 XML 文書のチューニングを行います。チューニング情報を調査して事前解析用 XML 文書を最適化することで、解析速度をさらに向上できます。高速化定義ファイルを使用する場合の、事前解析用 XML 文書のチューニング方法は、通常の高速パースの場合と同じです。詳細は、「3.3.9 事前解析用 XML 文書のチューニング」を参照してください。

### 3.3.14 高速化定義ファイルの作成

高速化定義ファイルの形式について説明します。なお、高速化定義ファイルを作成する際は、「6.17.5 高速化定義ファイルに関する注意事項」も参照してください。

#### (1) 形式

高速パース機能の対象となるXML文書名=事前解析用XML文書名[,NS]

(凡例) [ ]:[ ]内の文字列が省略できることを示します。

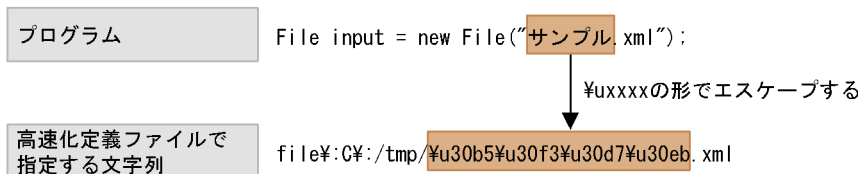
##### 高速パース機能の対象となる XML 文書名

高速パース機能の対象となる XML 文書名を絶対 URI または相対 URI で指定します。「:」を記述する場合は、「¥」でエスケープしてください。

ここに指定する文字列は、実際にユーザプログラム上で解析対象の XML 文書名として使用する文字列 ( `parse` メソッドの引数として指定する URI 文字列や、 `InputSource` に設定するシステム識別子など ) と必ず表記を一致させてください。同じファイルを示す URI であっても、表記上の違いがある場合は、高速パースが適用されません。例えば、 `parse` メソッドの引数として使用する URI が「a.xml」である場合に、高速化定義ファイルで「./a.xml= ~」と指定しても、高速パースは適用されません。

ただし、 `parse` メソッドの引数として `File` オブジェクトを渡す場合は、必ず「file:」の後ろに絶対パスを連結した文字列を指定してください。この場合に、 `File` クラスのコンストラクタ引数に指定した文字列が、Java のプロパティファイルの中で使用できない文字 ( 日本語や複数バイト文字など ) を含むときは、次に示すように

¥uxxxx の形式でエスケープする必要があります。



### 事前解析用 XML 文書名

事前解析用 XML 文書名を絶対パスで指定します。絶対パスを指定する際の注意事項については、「6.17.6 絶対パスの指定方法に関する注意事項」を参照してください。

### NS 名前空間の設定

事前解析用 XML 文書を解析する際、名前空間を有効にするかどうかを指定します。名前空間を有効にする場合は、事前解析用 XML 文書名に続けて、「,NS」と記述してください。「,NS」の記述がない場合は、名前空間を無効にして事前解析を実施します。

なお、「NS」は必ず大文字で指定してください。小文字で指定した場合は、事前解析用 XML 文書名の一部として扱われます。また、「,NS」は、必ず事前解析用 XML 文書名の直後に、空白やタブを入れないで指定してください。

なお、高速化定義ファイルは Java のプロパティファイルであり、基本的な構文規則はプロパティファイルの規則に従います。このため、同一のキーを複数指定した場合は、あとに書かれた指定が有効となります。

## (2) 記述例

高速化定義ファイルの記述例を次に示します。

高速パース機能の対象となる XML 文書名を相対 URI で指定する場合

```
test.xml=C:/preparse/pre.xml
```

事前解析時に名前空間を有効にする場合

```
test.xml=C:/preparse/pre.xml,NS
```

解析対象の XML 文書が File オブジェクトで XML パーサに渡される場合

```
file¥:C¥:/document/example.xml=C:/preparse/pre.xml
```

高速パース機能の対象となる XML 文書名には、「file:」の後ろに絶対パスを連結した文字列を指定します。また、左辺に「:」を記述する場合は、「¥」でエスケープします。

複数の XML 文書に同一の事前解析用 XML 文書を指定する場合

```
test1.xml=C:/preparse/pre.xml
test2.xml=C:/preparse/pre.xml
```

### 3.3.15 高速化定義ファイルの設定

高速化定義ファイルを使用するには、Cosminexus XML Processor が規定するシステムプロパティを設定する必要があります。

高速化定義ファイルを使用するためのシステムプロパティ、およびエラー発生時の動作について説明します。

#### (1) 高速化定義ファイルを使用するためのシステムプロパティ

システムプロパティ名

`com.cosminexus.jaxp.preparedxml.config`

値

高速化定義ファイルを示す絶対パスを指定します。絶対パスを指定する際の注意事項については、「6.17.6 絶対パスの指定方法に関する注意事項」を参照してください。

値が設定されていない場合は、高速化定義ファイルを使用しません。

なお、このシステムプロパティは、システムで最初に事前解析を実行する際に 1 度だけ参照されます。このため、事前解析が 1 度でも実行されたあとにこのシステムプロパティを変更したり、このシステムプロパティに設定されている高速化定義ファイルを更新したりしても、効果はありません。

#### (2) エラー発生時の動作

(1) のシステムプロパティに設定した高速化定義ファイルについて入出力エラーが発生した場合は、事前解析の実行タイミングによって次のように動作します。

J2EE サーバまたは Web コンテナサーバの起動時に事前解析を行う場合

標準エラー出力ストリームにエラーメッセージが出力されて、サーバの起動が中止されます。

高速パースの対象となる各 XML 文書の初回解析時に事前解析を行う場合

IOException 例外が発生します。

事前解析の実行タイミングについては、「3.3.16 事前解析の実行タイミングの設定」を参照してください。

### 3.3.16 事前解析の実行タイミングの設定

高速化定義ファイルを使用した高速パースでは、ユーザプログラムを変更しないため、次のどちらかのタイミングで事前解析用 XML 文書の解析を実行します。

- J2EE サーバまたは Web コンテナサーバの起動時
- 高速パースの対象となる各 XML 文書の初回解析時

事前解析の実行タイミング、および事前解析の実行タイミングの設定方法について説明

します。

### (1) 事前解析の実行タイミング

高速化定義ファイルを使用して高速パース機能を使用する場合の、事前解析の実行タイミングについて説明します。

#### J2EE サーバまたは Web コンテナサーバの起動時

サーバの起動時に、高速化定義ファイルの記述内容に従って、事前解析用 XML 文書の解析を実行します。事前解析で生成された解析結果オブジェクトは、サーバが終了するまでメモリ上に保持されます。

#### 高速パースの対象となる各 XML 文書の初回解析時

高速化定義ファイルに指定した高速パース機能の対象となる XML 文書を初めて解析するタイミングで、事前解析用 XML 文書を解析します。このため、高速パースの対象となる各 XML 文書の初回解析時には、通常の解析時間に加え、解析結果オブジェクトを生成する時間が掛かります。高速パースの対象となる XML 文書の 2 回目以降の解析では、初回に生成した解析結果オブジェクトを利用して高速パースを実現します。

なお、高速パースの対象となる一つの XML 文書を複数のスレッドで解析する場合、すべてのスレッドの中で最初に実行された解析で、事前解析を開始します。この場合、事前解析が完了する前に開始された別のスレッドの解析には、高速パースが適用されません。

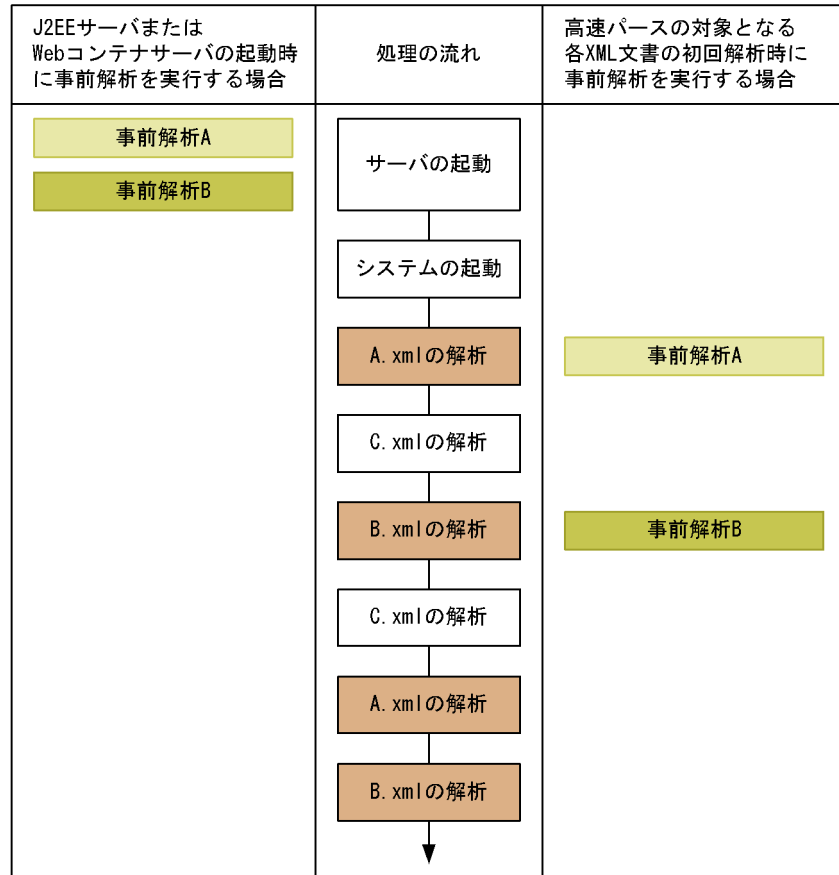
それぞれの事前解析の実行タイミングを次の図に示します。

図 3-19 事前解析の実行タイミング

高速化定義ファイルの設定内容

A. xml = 事前解析用XML文書A  
B. xml = 事前解析用XML文書B

事前解析の実行タイミング



(凡例)

 : 高速パースの対象となる解析処理

 事前解析A : 事前解析用XML文書Aの解析処理

 事前解析B : 事前解析用XML文書Bの解析処理

なお、高速化定義ファイルを使用する場合の事前解析時の外部エンティティ参照は、事前解析のタイミングに関係なく、デフォルトのエンティティリゾルバで解決されます。

## (2) 事前解析の実行タイミングの設定方法

事前解析の実行タイミングは、J2EE サーバまたは Web コンテナサーバのユーザプロパ

ティファイル ( `usrconf.properties` ) のシステムプロパティで設定します。

事前解析の実行タイミングを設定するためのシステムプロパティ , およびエラー発生時の動作について説明します。

( a ) 事前解析の実行タイミングを設定するためのシステムプロパティ

システムプロパティ名

`ejbserver.application.InitTermProcessClasses`

値

- `com.cosminexus.jaxp.preparedxml.PreparedInitializer` を含む J2EE サーバまたは Web コンテナサーバの起動時に事前解析を実行します。
- `com.cosminexus.jaxp.preparedxml.PreparedInitializer` を含まない  
高速パースの対象となる各 XML 文書の初回解析時に事前解析を実行します。

なお , システムプロパティ「 `ejbserver.application.InitTermProcessClasses` 」の詳細は , マニュアル「 [Cosminexus リファレンス 定義編](#) 」を参照してください。

( b ) エラー発生時の動作

事前解析中にエラーが発生した場合は , 事前解析の実行タイミングによって次のように動作します。

J2EE サーバまたは Web コンテナサーバの起動時に事前解析を行う場合

標準エラー出力ストリームにエラーメッセージが出力されて , サーバの起動が中止されます。

高速パースの対象となる各 XML 文書の初回解析時に事前解析を行う場合

`SAXException` 例外または `IOException` 例外が発生します。

### 3.3.17 XML 文書の解析 ( 高速化定義ファイルを使用する場合 )

高速化定義ファイルを使用する場合は , 通常の高速パースを適用していないすべての XML パーサが高速パースの対象となります。XML パーサの入力となる XML 文書が , 高速化定義ファイルで高速パース機能の対象となる XML 文書として指定されている場合に , 自動的に高速パースが適用されます。

高速化定義ファイルを使用する場合に高速パース機能の対象となる `parse` メソッドを次の表に示します。

#### ! 注意事項

表に記載のない `parse` メソッドを使用して解析した場合の動作は保証されません。



表 3-5 高速パース機能の対象となる parse メソッド（高速化定義ファイルを使用する場合）

クラス名	メソッド名
javax.xml.parsers.DocumentBuilder	parse(File f)
	parse(InputSource is)
	parse(InputStream is, String systemId)
	parse(String uri)
javax.xml.parsers.SAXParser	parse(File f, DefaultHandler dh)
	parse(InputSource is, DefaultHandler dh)
	parse(InputStream is, DefaultHandler dh, String systemId)
	parse(String uri, DefaultHandler dh)
org.xml.sax.XMLReader	parse(InputSource input)
	parse(String systemId)

## 注

InputSource にシステム識別子が設定されていない場合は、高速パースの対象外となります。

## 3.4 XML Schema のエラーメッセージと W3C 仕様との対応

---

Cosminexus XML Processor が出力するエラーメッセージには、W3C での仕様個所を特定するための識別子が付いています。これは、スキーマ文書での構文エラーおよび XML 文書での検証エラーに対して、一貫したエラー情報を提供するためです。

エラーメッセージに付いている識別子の一覧は、W3C による XML Schema の仕様書、Part1: Structure Appendices C に記載されています。

例えば、スキーマ文書中で要素内のデータ型を整数型 (`xsd:integer`) と規定したにもかかわらず、XML 文書で文字列データを指定した場合は、次のメッセージが出力されます。

```
KECX06036-E cvc-datatype-valid.1.2.1: 'ABC' is not a valid value for 'integer'.
```

このメッセージでは、「`cvc-datatype-valid.1.2.1`」が識別子となります。これは、「Validation Rule: Datatype Valid」の、1.2.1 に記載された規則に違反していることを示します。

# 4

## フィーチャーおよびプロパティの使用方法

この章では、JAXP で規定されているフィーチャーおよびプロパティの使用方法について説明します。また、Cosminexus XML Processor の拡張機能に関連したフィーチャーおよびプロパティの使用方法についても説明します。

---

4.1 SAX2 のフィーチャーおよびプロパティの使用方法

---

4.2 XSLT のフィーチャーの使用方法

---

4.3 XML Schema のプロパティの使用方法

---

4.4 Shift\_JIS 切り替え機能のプロパティの使用方法

---

4.5 高速パース機能のプロパティの使用方法

---

## 4.1 SAX2 のフィーチャーおよびプロパティの使用法

SAX2 規格では、フィーチャーおよびプロパティという機構により、SAX パーサのオプション機能を使用する方法を規定しています。

フィーチャー名およびプロパティ名は、SAX パーサのオプション機能に対応した URI です。フィーチャー名、プロパティ名の形式は次のとおりです。

フィーチャー名 = `http://xml.org/sax/features/` + フィーチャーID  
 プロパティ名 = `http://xml.org/sax/properties/` + プロパティID

Cosminexus XML Processor がサポートしている SAX2 のフィーチャーを表 4-1 に、Cosminexus XML Processor がサポートしている SAX2 のプロパティを表 4-2 に示します。

表 4-1 Cosminexus XML Processor がサポートしている SAX2 のフィーチャー一覧

項番	フィーチャー名
1	<code>http://xml.org/sax/features/external-general-entities</code>
2	<code>http://xml.org/sax/features/external-parameter-entities</code>
3	<code>http://xml.org/sax/features/namespace</code>
4	<code>http://xml.org/sax/features/namespace-prefixes</code>
5	<code>http://xml.org/sax/features/use-attributes2</code>
6	<code>http://xml.org/sax/features/use-locator2</code>
7	<code>http://xml.org/sax/features/use-entity-resolver2</code>
8	<code>http://xml.org/sax/features/validation</code>

表 4-2 Cosminexus XML Processor がサポートしている SAX2 のプロパティ一覧

項番	プロパティ名
1	<code>http://xml.org/sax/properties/declaration-handler</code>
2	<code>http://xml.org/sax/properties/lexical-handler</code>

それぞれのフィーチャー、およびプロパティの意味については、Simple API for XML (SAX) 2.0.2 (sax2r3) の Javadoc の次に示す個所を参照してください。

- org.xml.sax パッケージの "SAX2 Standard Feature Flags"
- org.xml.sax パッケージの "SAX2 Standard Handler and Property IDs"

## (1) SAX2 のフィーチャーの使用法

SAX2 のフィーチャーを設定するには、org.sax.xml.XMLReader クラスの setFeature メソッドを使用します。フィーチャーの設定方法を次に示します。

```
// SAXパーサを生成する
SAXParserFactory factory = SAXParserFactory.newInstance();
SAXParser parser = factory.newSAXParser();

// SAXパーサに含まれるXMLReaderを取得する
XMLReader reader = parser.getXMLReader();

// "validation"フィーチャーを有効に設定する
reader.setFeature("http://xml.org/sax/features/validation", true);
```

## (2) SAX2 のプロパティの使用法

SAX2 のプロパティを設定するには、org.sax.xml.XMLReader クラスの setProperty メソッドを使用します。プロパティの設定方法を次に示します。

```
// SAXパーサを生成する
SAXParserFactory factory = SAXParserFactory.newInstance();
SAXParser parser = factory.newSAXParser();

// SAXパーサに含まれるXMLReaderを取得する
XMLReader reader = parser.getXMLReader();

// プロパティ設定により、DeclHandlerを登録する。
DeclHandler handler = new MyDeclHandler();
reader.setProperty("http://xml.org/sax/properties/declaration-handler", handler);
```

## 4.2 XSLT のフィーチャーの使用法

JAXP 規格では、フィーチャーという機構により、XSLT トランスフォーマのオプション機能を使用する方法を規定しています。

フィーチャー名は、XSLT トランスフォーマのオプション機能に対応した URI です。JAXP で規定されている XSLT フィーチャー一覧を表 4-3 に示します。

表 4-3 JAXP で規定されている XSLT のフィーチャー一覧

項番	フィーチャー名
1	<code>http://javax.xml.transform.stream.StreamSource/feature</code>
2	<code>http://javax.xml.transform.stream.StreamResult/feature</code>
3	<code>http://javax.xml.transform.dom.DOMSource/feature</code>
4	<code>http://javax.xml.transform.dom.DOMResult/feature</code>
5	<code>http://javax.xml.transform.dom.SAXSource/feature</code>
6	<code>http://javax.xml.transform.dom.SAXResult/feature</code>
7	<code>http://javax.xml.transform.sax.SAXTransformerFactory/feature</code>
8	<code>http://javax.xml.transform.sax.SAXTransformerFactory/feature/xmlfilter</code>

それぞれのフィーチャーの意味については、JSR 206 Java API for XML Processing(JAXP) 1.3 の Chapter 8. Package `javax.xml.transform`, Chapter 9. Package `javax.xml.transform.dom`, Chapter 10. Package `javax.xml.transform.sax` の各クラスに記載されている FEATURE フィールドを参照してください。

### (1) XSLT のフィーチャーの使用法

XSLT のフィーチャーは、設定することはできません。XSLT のフィーチャーを参照するには、`javax.xml.transform.TransformerFactory` クラスの `getFeature` メソッドを使用します。`getFeature` の結果はすべて `true` になります。

## 4.3 XML Schema のプロパティの使用法

JAXP で規定されたプロパティを設定すると、スキーマ検証に使用するスキーマ言語、およびスキーマ文書を特定できます。プロパティの詳細については、JSR 206 Java API for XML Processing(JAXP) 1.3 の Chapter 3. Plugability Layer の Properties For Enabling Schema Validation を参照してください。

### 4.3.1 DOM パーサに対する XML Schema のプロパティの設定方法

DOM パーサに対して XML Schema のプロパティを設定するには、`javax.xml.parsers.DocumentBuilderFactory` クラスの `setAttribute` メソッドを使用します。プロパティの設定方法を次に示します。

コーディング例について次に説明します。各項番はコーディング例のコメントに記述している番号に対応しています。

```
try {
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    dbf.setNamespaceAware(true);
    // 1. 妥当性を検証するように設定します。
    dbf.setValidating(true);
    // 2. スキーマ検証に使用するスキーマ言語を設定します。
    dbf.setAttribute("http://java.sun.com/xml/jaxp/properties/schemaLanguage",
        "http://www.w3.org/2001/XMLSchema");
    // 3. スキーマ文書を特定します。
    dbf.setAttribute("http://java.sun.com/xml/jaxp/properties/schemaSource",
        "purchaseOrder.xsd");
    DocumentBuilder db = dbf.newDocumentBuilder();
    Document doc = db.parse("purchaseOrder.xml");
} catch (Exception e) {
    e.printStackTrace();
}
```

1. 妥当性を検証するように設定します。  
`javax.xml.parsers.DocumentBuilderFactory` クラス、`setValidating` メソッドの引数に `true` を設定します。これによって、XML 文書の妥当性検証の機能が有効になります。
2. スキーマ検証に使用するスキーマ言語を設定します。  
`javax.xml.parsers.DocumentBuilderFactory` クラス、`setAttribute` メソッドで、プロパティ文字列 `"http://java.sun.com/xml/jaxp/properties/schemaLanguage"` に対して、XML Schema の仕様に従って検証を行うことを示す値、`"http://www.w3.org/2001/XMLSchema"` を設定します。これによって、XML Schema による妥当性検証が行われます。
3. スキーマ文書を特定します。

#### 4. フィーチャーおよびプロパティの使用方法

javax.xml.parsers.DocumentBuilderFactory クラス, setAttribute メソッドで, プロパティ文字列 "http://java.sun.com/xml/jaxp/properties/schemaSource" に対して, 検証に使うスキーマ文書を特定します。XML 文書内でスキーマ文書を特定する場合, この指定は不要です。

### 4.3.2 SAX パーサに対する XML Schema のプロパティの設定方法

SAX パーサに対して XML Schema のプロパティを設定するには, javax.xml.parsers.SAXParser クラスの setProperty メソッドを使用します。プロパティの設定方法を次に示します。

```
try {
    SAXParserFactory spf = SAXParserFactory.newInstance();
    spf.setNamespaceAware(true);
    // 1. 妥当性を検証するように設定します。
    spf.setValidating(true);
    SAXParser sp = spf.newSAXParser();
    // 2. スキーマ検証に使用するスキーマ言語を設定します。
    sp.setProperty("http://java.sun.com/xml/jaxp/properties/schemaLanguage",
        "http://www.w3.org/2001/XMLSchema");
    // 3. スキーマ文書を特定します。
    sp.setProperty("http://java.sun.com/xml/jaxp/properties/schemaSource",
        "purchaseOrder.xsd");
    XMLReader reader = sp.getXMLReader();
    reader.parse("purchaseOrder.xml");
} catch (Exception e) {
    e.printStackTrace();
}
```

コーディング例について次に説明します。各項番はコーディング例のコメントに記述している番号に対応しています。

#### 1. 妥当性を検証するように設定します。

javax.xml.parsers.SAXParserFactory クラス, setValidating メソッドの引数に true を設定します。これによって, XML 文書の妥当性検証の機能が有効になります。

#### 2. XML Schema の仕様に従うように設定します。

javax.xml.parsers.SAXParser クラス, setProperty メソッドで, プロパティ文字列 "http://java.sun.com/xml/jaxp/properties/schemaLanguage" に対して, XML Schema の仕様に従って検証を行うことを示す値, "http://www.w3.org/2001/XMLSchema" を設定します。これによって, XML Schema による妥当性検証が行われます。

#### 3. スキーマ文書を特定します。

javax.xml.parsers.SAXParser クラス, setProperty メソッドで, プロパティ文字列 "http://java.sun.com/xml/jaxp/properties/schemaSource" に対して, 検証に使うスキーマ文書を指定します。XML 文書内でスキーマ文書を特定する場合, この指定は不要です。



### 4.3.3 スキーマ文書を XML 文書内で設定する方法

「4.3.1 DOM パーサに対する XML Schema のプロパティの設定方法」、および「4.3.2

SAX パーサに対する XML Schema のプロパティの設定方法」に示した方法では、プロパティを用いてスキーマ文書を設定しています。これに対して、検証の対象となる XML 文書内で、直接スキーマ文書を特定する方法もあります。なお、XML 文書内でスキーマ文書を特定し、かつ、プロパティを用いてスキーマ文書を特定した場合は、プロパティを使って特定したスキーマ文書が優先されます。

検証の対象となる XML 文書内でスキーマ文書を特定するには、`xsi:schemaLocation` 属性または `xsi:noNamespaceSchemaLocation` 属性を使用します。これらの属性は、特定するスキーマ文書内で、名前空間を定義しているかどうかによって異なります。特定するスキーマ文書の種類と使用する属性の関係を表 4-4 に示します。

表 4-4 特定するスキーマ文書の種類と使用する属性の関係

特定するスキーマ文書の種類	使用する属性
名前空間を定義したスキーマ文書	<code>xsi:schemaLocation</code>
名前空間を定義していないスキーマ文書	<code>xsi:noNamespaceSchemaLocation</code>

`xsi:schemaLocation` 属性を使用する場合、特定するスキーマ文書内で宣言されている要素や属性の名前空間の URI と、特定するスキーマ文書名を空白で区切って指定します。

`xsi:schemaLocation` 属性、および `xsi:noNamespaceSchemaLocation` 属性は、どちらも名前空間 "`http://www.w3.org/2001/XMLSchema-instance`" に属します。したがって、これらの属性を使ってスキーマ文書を特定する場合、名前空間 "`http://www.w3.org/2001/XMLSchema-instance`" を宣言する必要があります。

`xsi:schemaLocation` 属性を使用してスキーマ文書を特定する場合の例を次に示します。

```
<? xml version="1.0"?>
<po:purchaseOrder xmlns:po=http://www.myshopping.com/schema/purchaseOrder
xmlns:psd="http://www.myshopping.com/schema/personalData"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ...1.
xsi:schemaLocation="http://www.myshopping.com/schema/purchaseOrder
purchaseOrder.xsd" ...2.
:
:
:
</po:purchaseOrder>
```

1. `xsi:schemaLocation` 属性の名前空間を宣言します。

`xsi:schemaLocation` 属性を使ってスキーマ文書を特定するために、この属性の名前空間 "`http://www.w3.org/2001/XMLSchema-instance`" を宣言します。

2. スキーマ文書を特定します。

特定するスキーマ文書内で宣言されている要素や属性の名前空間の URI、およびスキーマ文書名を空白で区切って指定します。

#### 4. フィーチャーおよびプロパティの使用法

上記の例では、スキーマ文書として purchaseOrder.xsd を、スキーマ文書 purchaseOrder.xsd 内で宣言されている要素や属性の名前空間として "http://www.myshopping.com/schema/purchaseOrder" を指定しています。

## 4.4 Shift\_JIS 切り替え機能のプロパティの使用法

Shift\_JIS 切り替え機能を使用するためには、Cosminexus XML Processor が独自に規定する次のプロパティを設定する必要があります。

プロパティ名 = `http://cosminexus.com/xml/properties/shift_jis_map`

プロパティの値には、文字エンコーディングを設定します。設定するプロパティの値とその意味を表 4-5 に示します。

表 4-5 設定するプロパティの値とその意味 (Shift\_JIS 切り替え機能)

プロパティに設定する値 <sup>1</sup>	意味
SJIS	SJIS エンコーディングを適用する。
MS932	MS932 エンコーディングを適用する。
その他の文字列	指定を無視する。 <sup>2</sup>

注 1

指定された文字列は、大文字・小文字を区別しません。

注 2

指定が無視された場合は、直前に指定されたエンコーディングが有効となります。



### 注意事項

Shift\_JIS 切り替え機能の指定がない場合、デフォルトでは SJIS エンコーディングとなります。

Shift\_JIS 切り替え機能のプロパティは、次に示すクラス、インタフェースのインスタンスごとに設定する必要があります。Shift\_JIS 切り替え機能の設定用メソッドを表 4-6 に示します。表 4-6 では、プロパティの設定に使用するメソッドを、クラス、インタフェースごとに示しています。

表 4-6 Shift\_JIS 切り替え機能の設定用メソッド

クラス、インタフェース	メソッド
<code>javax.xml.parsers.DocumentBuilderFactory</code>	<code>setAttribute</code>
<code>javax.xml.parsers.SAXParser</code>	<code>setProperty</code>
<code>org.xml.sax.XMLReader</code>	<code>setProperty</code>
<code>javax.xml.parsers.TransformerFactory</code>	<code>setAttribute</code>

#### 4. フィーチャーおよびプロパティの使用法

DOM パーサ，SAX パーサ，および XSLT トランスフォーマで Shift\_JIS 切り替え機能を使用する場合について，例を用いて説明します。

##### 4.4.1 DOM パーサで Shift\_JIS 切り替え機能を使用する方法

DOM パーサで Shift\_JIS 切り替え機能を使用する基本的なコーディングの流れについて説明します。

```
// 1. DocumentBuilderFactoryを作成します。
javax.xml.parsers.DocumentBuilderFactory factory =
    javax.xml.parsers.DocumentBuilderFactory.newInstance();
// 2. Shift_JISをMS932に切り替えます。
factory.setAttribute("http://cosminexus.com/xml/properties/shift_jis_map", "MS932");
// 3. DocumentBuilderを作成します。
javax.xml.parsers.DocumentBuilder db = factory.newDocumentBuilder();
```

コーディング例について次に説明します。各項番はコーディング例のコメントに記述している番号に対応しています。

1. DocumentBuilderFactory を作成します。

javax.xml.parsers.DocumentBuilderFactory の static メソッド newInstance() を使用してファクトリを作成します。

2. Shift\_JIS を MS932 に切り替えます。

DOM パーサの設定用メソッド setAttribute の第一引数に設定キー http://cosminexus.com/xml/properties/shift\_jis\_map を，第二引数に MS932 を指定します。

3. DocumentBuilder を作成します。

newDocumentBuilder() メソッドを使用して，DocumentBuilder を作成します。

##### 4.4.2 SAX パーサで Shift\_JIS 切り替え機能を使用する方法

SAX パーサで Shift\_JIS 切り替え機能を使用する基本的なコーディングの流れについて説明します。コーディング例を次に示します。

```
// 1. SAXParserFactoryを作成します。
javax.xml.parsers.SAXParserFactory factory =
    javax.xml.parsers.SAXParserFactory.newInstance();
// 2. SAXParserを作成します。
javax.xml.parsers.SAXParser parser = factory.newSAXParser();
// 3. XMLReaderを取得します。
org.xml.sax.XMLReader reader = parser.getXMLReader();
// 4. Shift_JISをMS932に切り替えます。
reader.setProperty("http://cosminexus.com/xml/properties/shift_jis_map", "MS932");
```

コーディング例について次に説明します。各項番はコーディング例のコメントに記述している番号に対応しています。

1. SAXParserFactory を作成します。

javax.xml.parsers.SAXParserFactory の static メソッド newInstance() を使用して

ファクトリを作成します。

2. SAXParser を作成します。

`newSAXParser()` を使用して SAX パーサを作成します。

3. XMLReader を取得します。

`getXMLReader()` を使用して XMLReader を取得します。

4. Shift\_JIS を MS932 に切り替えます。

SAX パーサの設定用メソッド `setProperty` の第一引数に設定キー `http://cosminexus.com/xml/properties/shift_jis_map` を、第二引数に MS932 を指定します。

### 4.4.3 XSLT トランスフォーマで Shift\_JIS 切り替え機能を使用する方法

XSLT トランスフォーマで Shift\_JIS 切り替え機能を使用する基本的なコーディングの流れについて説明します。コーディング例を次に示します。

```
// 1. TransformerFactoryを作成します。
javax.xml.transform.TransformerFactory factory =
    javax.xml.transform.TransformerFactory.newInstance();
// 2. Shift_JISをMS932に切り替えます。
factory.setAttribute("http://cosminexus.com/xml/properties/shift_jis_map", "MS932");
// 3. Transformerを作成します。
factory.newTransformer();
```

コーディング例について次に説明します。各項番はコーディング例のコメントに記述している番号に対応しています。

1. TransformerFactory を作成します。

`javax.xml.transform.TransformerFactory` の static メソッド `newInstance()` を使用してファクトリを作成します。

2. Shift\_JIS を MS932 に切り替えます。

XSLT トランスフォーマファクトリの設定用メソッド `setAttribute` の第一引数に設定キー `http://cosminexus.com/xml/properties/shift_jis_map` を、第二引数に MS932 を指定します。

3. Transformer を作成します。

`newTransformer()` メソッドを使用して、トランスフォーマを作成します。

## 4.5 高速パース機能のプロパティの使用法

高速パース機能を使用するためには、Cosminexus XML Processor が独自に規定する次のプロパティを設定する必要があります。

```
プロパティ名 = http://cosminexus.com/xml/properties/preparsedobject-load
```

プロパティの値には、null または PreparsedObject オブジェクトを設定します。設定するプロパティの値とその意味を表 4-7 に示します。

表 4-7 設定するプロパティの値とその意味（高速パース機能）

プロパティに設定する値	意味
null	解析時に解析結果オブジェクトを使用しない。
PreparsedObject オブジェクト	解析時に、指定した解析結果オブジェクトを使用する。

### 注

指定が省略された場合は、null が指定されたものとして扱います。

また、null および PreparsedObject オブジェクト以外の値が設定された場合は、プロパティが設定されている XML パーサによって、次の例外が発生します。

- DocumentBuilderFactory の場合：IllegalArgumentException
- SAXParser または XMLReader の場合：SAXNotSupportedException

高速パース機能のプロパティは、次に示す XML パーサのオブジェクトごとに設定する必要があります。解析結果オブジェクトの設定用メソッドを表 4-8 に示します。表 4-8 では、プロパティの設定に使用するメソッドを、クラス、インタフェースごとに示しています。

表 4-8 解析結果オブジェクトの設定用メソッド

クラス、インタフェース	メソッド
javax.xml.parsers.DocumentBuilderFactory	setAttribute
javax.xml.parsers.SAXParser	setProperty
org.xml.sax.XMLReader	setProperty

DocumentBuilder、SAXParser、および XMLReader で高速パース機能を使用する場合について、例を用いて説明します。

### 4.5.1 DocumentBuilder に解析結果オブジェクトを設定する方法

DocumentBuilder に解析結果オブジェクトを設定する場合のコーディング例を次に示し

ます。

```
// 1. 解析結果オブジェクトを生成します。
File xml = new File(...);
PreparedObjectFactory pof = PreparedObjectFactory.newInstance();
pof.setNamespaceAware(...);
pof.setEntityResolver(...);
pof.setErrorHandler(...);
PreparedObject pobj = pof.newPreparedObject(xml);

// 2. DocumentBuilderに解析結果オブジェクトを設定します。
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setAttribute("http://cosminexus.com/xml/properties/preparedobject-load", pobj);
DocumentBuilder db = dbf.newDocumentBuilder();
```

コーディング例について次に説明します。各項番はコーディング例のコメントに記述している番号に対応しています。

1. 解析結果オブジェクトを生成します。  
解析結果オブジェクトを生成する方法については、「3.3.4 解析結果オブジェクトの生成」を参照してください。
2. DocumentBuilder に解析結果オブジェクトを設定します。  
解析結果オブジェクトの設定用メソッド `setAttribute` の第一引数に設定キー `http://cosminexus.com/xml/properties/preparedobject-load` を、第二引数に `PreparedObject` オブジェクトを示す `pobj` を指定します。

## 4.5.2 SAXParser に解析結果オブジェクトを設定する方法

SAXParser に解析結果オブジェクトを設定する場合のコーディング例を次に示します。

```
// 1. 解析結果オブジェクトを生成します。
File xml = new File(...);
PreparedObjectFactory pof = PreparedObjectFactory.newInstance();
pof.setNamespaceAware(...);
pof.setEntityResolver(...);
pof.setErrorHandler(...);
PreparedObject pobj = pof.newPreparedObject(xml);

// 2. SAXParserに解析結果オブジェクトを設定します。
SAXParserFactory spf = SAXParserFactory.newInstance();
SAXParser sp = spf.newSAXParser();
sp.setProperty("http://cosminexus.com/xml/properties/preparedobject-load", pobj);
```

1. 解析結果オブジェクトを生成します。  
解析結果オブジェクトを生成する方法については、「3.3.4 解析結果オブジェクトの生成」を参照してください。
2. SAXParser に解析結果オブジェクトを設定します。  
解析結果オブジェクトの設定用メソッド `setProperty` の第一引数に設定キー `http://cosminexus.com/xml/properties/preparedobject-load` を、第二引数に `PreparedObject` オブジェクトを示す `pobj` を指定します。

### 4.5.3 XMLReader に解析結果オブジェクトを設定する方法

XMLReader に解析結果オブジェクトを設定する場合のコーディング例を次に示します。

```
// 1. 解析結果オブジェクトを生成します。
File xml = new File(...);
PreparedObjectFactory pof = PreparedObjectFactory.newInstance();
pof.setNamespaceAware(...);
pof.setEntityResolver(...);
pof.setErrorHandler(...);
PreparedObject pobj = pof.newPreparedObject(xml);

// 2. XMLReaderに解析結果オブジェクトを設定します。
SAXParserFactory spf = SAXParserFactory.newInstance();
SAXParser sp = spf.newSAXParser();
XMLReader reader = sp.getXMLReader();
reader.setProperty("http://cosminexus.com/xml/properties/preparedobject-load", pobj);
```

1. 解析結果オブジェクトを生成します。

解析結果オブジェクトを生成する方法については、「3.3.4 解析結果オブジェクトの生成」を参照してください。

2. XMLReader に解析結果オブジェクトを設定します。

解析結果オブジェクトの設定用メソッド `setProperty` の第一引数に設定キー `http://cosminexus.com/xml/properties/preparedobject-load` を、第二引数に `PreparedObject` オブジェクトを示す `pobj` を指定します。



# 5

## プログラムの作成方法

この章では、プログラム作成の流れ、使用するパッケージ、および JAXP の主な機能を用いたサンプルプログラムについて説明します。

---

5.1 プログラム作成の流れ

---

5.2 Cosminexus XML Processor が使用しているパッケージ名

---

5.3 プログラム実行時のトラブルシュート

---

5.4 DOM パーサを使用するサンプルプログラム

---

5.5 SAX パーサを使用するサンプルプログラム

---

5.6 XML Schema を使用するサンプルプログラム

---

5.7 XSLT トランスフォーマを使用するサンプルプログラム

---

5.8 XSLTC トランスフォーマを使用するサンプルプログラム

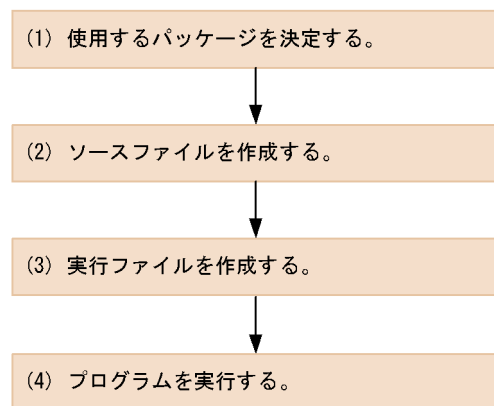
---

## 5.1 プログラム作成の流れ

---

Cosminexus XML Processor を使用したプログラムの作成の流れを図 5-1 に示します。

図 5-1 Cosminexus XML Processor を使用したプログラム作成の流れ



### (1) 使用するパッケージを決定する

Cosminexus XML Processor は、JAXP 仕様で規定されたパッケージ、および XSLTC トランスフォーマ機能のパッケージを提供しています。プログラムを作成する際には、そのプログラムの用途に適したパッケージを選択してください。

例えば、読み込んだ XML 文書をメモリ上に保持し、その構造を変更したり、複雑な操作をしたりする必要がある場合は DOM を、対話的に XML 文書を解析しながら処理をする場合は SAX を使用します。また、XML 文書を HTML などに変換する場合は、XSLT を使用します。

JAXP 仕様で規定されたパッケージについては、「2.2 JAXP が規定するパッケージとその機能」を、XSLTC トランスフォーマ機能のパッケージについては、「3.2.3 XSLTC トランスフォーマで使用するクラス」を参照してください。

### (2) ソースファイルを作成する

使用するパッケージの決定後、それぞれのパッケージに含まれるクラスやインタフェースを使用して、ソースファイルを作成します。

### (3) 実行ファイルを作成する

作成したソースファイルから実行ファイルを作成します。

実行ファイルを作成するには、作成したソースファイルを `javac` コマンドでコンパイルし、使用するクラスやインタフェースに必要なライブラリをリンクする必要があります。

(4) プログラムを実行する

コンパイルしたプログラムを実行します。

## 5.2 Cosminexus XML Processor が使用しているパッケージ名

---

Cosminexus XML Processor は、「com.cosminexus.jaxp.」で始まるパッケージ名を使用しています。

### **!** 注意事項

ユーザプログラムのパッケージ名が、「com.cosminexus.jaxp.」で始まらないようにしてください。

---

## 5.3 プログラム実行時のトラブルシューティング

Cosminexus XML Processor を使用したシステムを運用する場合、システム管理者への連絡が必要になるような障害や、対処方法が不明な障害が発生した場合の障害調査資料として、アプリケーションログには次の情報を採取しておく必要があります。

- 障害発生時刻
- 障害発生時の入力となった XML 文書名（スキーマ文書名、スタイルシート名含む）

上記のアプリケーションログと同時に、次に示す障害情報を取得した上、システム管理者に連絡してください。

- アプリケーションログファイル
- 障害発生時の入力となった XML 文書（スキーマ文書、スタイルシート含む）
- 障害発生時に出力された Cosminexus XML Processor のエラーメッセージ
- サーバ、およびクライアントに設定したシステムプロパティとシステムクラスパス
- サーバ、およびクライアントの標準出力と標準エラー出力
- Cosminexus Component Container、および Web サーバのログ
- Cosminexus Component Container で規定する障害時取得情報

Cosminexus Component Container で規定する障害時取得情報を次に示します。

- J2EE サーバのユーザ定義ファイル
- J2EE サーバの保守情報
- サーバ管理コマンドのユーザ定義ファイル
- サーバ管理コマンドの保守情報
- J2EE サーバの標準出力、標準エラー出力
- J2EE サーバ側の Cosminexus TPBroker のログ
- EJB クライアントアプリケーションに設定したシステムプロパティ、およびシステムクラスパス
- EJB クライアントアプリケーションの標準出力、標準エラー出力
- ネットワークモニタツールで採取したパケットの内容

Cosminexus Component Container で規定する障害時取得情報の取得方法については、マニュアル「Cosminexus アプリケーション開発ガイド」を参照してください。

## 5.4 DOM パーサを使用するサンプルプログラム

この節では、DOM パーサを使用するサンプルプログラムについて説明します。

サンプルプログラムのインストール先、ファイル名は次のとおりです。

インストール先

Cosminexus XML Processor をインストールしたディレクトリ /samples/DOM

ファイル名

SampleDOM.java

### ! 注意事項

サンプルプログラムの実行ファイルの作成、および実行は、DOM ディレクトリ下で行ってください。

### 5.4.1 サンプルプログラムの処理

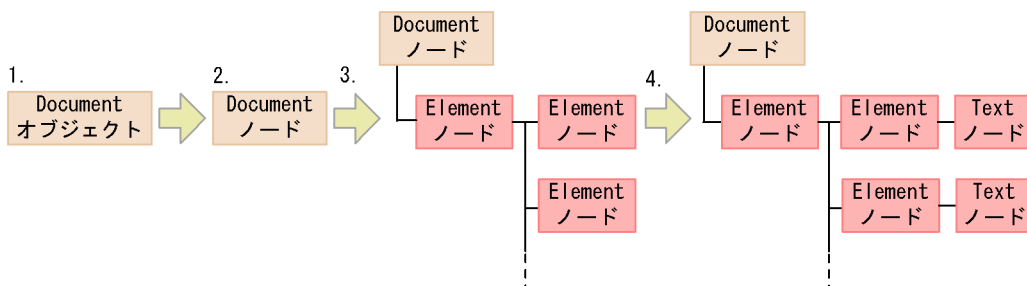
サンプルプログラムは次のような処理を行います。

1. 商品の情報について記述した DOM ツリーを新規に作成します。
2. 作成した DOM ツリーを XML 文書に変換・出力します。
3. 出力した XML 文書を読み込みます。
4. 読み込んだ XML 文書から、商品の合計金額を画面に出力します。

### 5.4.2 サンプルプログラムの作成の流れ

DOM パーサを使用するサンプルプログラムの作成の流れを図 5-2 に示します。

図 5-2 DOM パーサを使用するサンプルプログラムの作成の流れ



1. Document オブジェクトを作成します。  
Document オブジェクトを新規に作成します。
2. Document ノードを作成します。  
1. で作成した Document オブジェクトを使用して DOM ツリーの親となる Document ノードを作成します。
3. Element ノードを作成します。  
必要となる要素ノードを Document ノードから作成し、DOM ツリーを作成します。
4. Text ノードを作成します。  
要素の内容となる文字列を作成します。

### 5.4.3 サンプルプログラム ( SampleDOM.java )

サンプルプログラムを次に示します。

```
import java.io.*;
import javax.xml.parsers.*;
import javax.xml.transform.*;
import javax.xml.transform.dom.*;
import javax.xml.transform.stream.*;
import org.xml.sax.SAXException;
import org.w3c.dom.*;

public class SampleDOM
{
    public static final void main(String[] args)
    {
        try{
            new SampleDOM().CreateXMLFile();
            new SampleDOM().GetStringFromXML();
        }catch(Exception exception){
            exception.printStackTrace();
        }
    }

    //XMLファイルを作成する
    public final void CreateXMLFile()
    {
        try{
            DocumentBuilderFactory dbf =
                DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();

            Document doc = db.newDocument();
            Element parent = doc.createElement("伝票");

            Element sub = doc.createElement("商品");
            Element leaf1 = doc.createElement("商品名");
            Element leaf2 = doc.createElement("単価");
            Element leaf3 = doc.createElement("数量");
            leaf1.appendChild(doc.createTextNode("テレビ"));
```

## 5. プログラムの作成方法

```
sub.appendChild(leaf1);
leaf2.appendChild(doc.createTextNode("15000"));
sub.appendChild(leaf2);
leaf3.appendChild(doc.createTextNode("14"));
sub.appendChild(leaf3);
parent.appendChild(sub);

sub = doc.createElement("商品");
leaf1 = doc.createElement("商品名");
leaf2 = doc.createElement("単価");
leaf3 = doc.createElement("数量");
leaf1.appendChild(doc.createTextNode("ラジオ"));
sub.appendChild(leaf1);
leaf2.appendChild(doc.createTextNode("3500"));
sub.appendChild(leaf2);
leaf3.appendChild(doc.createTextNode("6"));
sub.appendChild(leaf3);
parent.appendChild(sub);

sub = doc.createElement("商品");
leaf1 = doc.createElement("商品名");
leaf2 = doc.createElement("単価");
leaf3 = doc.createElement("数量");
leaf1.appendChild(doc.createTextNode("ビデオ"));
sub.appendChild(leaf1);
leaf2.appendChild(doc.createTextNode("21000"));
sub.appendChild(leaf2);
leaf3.appendChild(doc.createTextNode("4"));
sub.appendChild(leaf3);
parent.appendChild(sub);

doc.appendChild(parent);

OutputStream os =
    new BufferedOutputStream(
        new FileOutputStream("SampleDOM.xml"));
TransformerFactory tf =
    TransformerFactory.newInstance();
Transformer tran = tf.newTransformer();
tran.setOutputProperty("encoding", "Shift_JIS");
tran.setOutputProperty("indent", "yes");
tran.transform(new DOMSource(doc),
    new StreamResult(os));

os.flush();
os.close();

} catch (ParserConfigurationException e) {
    e.printStackTrace();
} catch (TransformerConfigurationException e) {
    e.printStackTrace();
} catch (TransformerException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}
```



```

//CreateXMLFileで作成したXMLファイルからデータを取得して計算する
public final void GetStringFromXML()
{
    try{
        int tvPrice = 0;
        int radioPrice = 0;
        int videoPrice = 0;

        DocumentBuilderFactory factory =
            DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();

        Document doc = builder.parse("SampleDOM.xml");
        Element element1 = doc.getDocumentElement();
        NodeList nodelist1 = element1.getElementsByTagName("商品");

        //商品の数だけループをまわす
        for(int nShouhin = 0 ;
            nShouhin < nodelist1.getLength() ; nShouhin++){
            String str = new java.lang.String();
            int price = 0;
            int number = 0;
            Element element2 = (Element)nodelist1.item(nShouhin);

            //商品名の取得
            NodeList nodelist2 =
                element2.getElementsByTagName("商品名");
            Node node1 = nodelist2.item(0);
            if(node1.getNodeType() == Node.ELEMENT_NODE){
                Element element3 = (Element)node1;
                NodeList nodelist3 = element3.getChildNodes();
                Node node2 = nodelist3.item(0);
                str = node2.getNodeValue();
            }

            //商品名に対する単価の取得
            nodelist2 = element2.getElementsByTagName("単価");
            node1 = nodelist2.item(0);
            if(node1.getNodeType() == Node.ELEMENT_NODE){
                Element element3 = (Element)node1;
                NodeList nodelist3 = element3.getChildNodes();
                Node node2 = nodelist3.item(0);
                price = Integer.parseInt(node2.getNodeValue());
            }

            //商品名に対する数量の取得
            nodelist2 = element2.getElementsByTagName("数量");
            node1 = nodelist2.item(0);
            if(node1.getNodeType() == Node.ELEMENT_NODE){
                Element element3 = (Element)node1;
                NodeList nodelist3 = element3.getChildNodes();
                Node node2 = nodelist3.item(0);
                number = Integer.parseInt(node2.getNodeValue());
            }

            //各商品の小計を計算する
            if(str.equals("テレビ")){
                tvPrice = price * number;
            }
        }
    }
}

```

## 5. プログラムの作成方法

```
        }else if(str.equals("ラジオ")){
            radioPrice = price * number;
        }else if(str.equals("ビデオ")){
            videoPrice = price * number;
        }
    }
    int sum = tvPrice + radioPrice + videoPrice;
    System.out.println("合計は" + sum + "円です。");
} catch (ParserConfigurationException e) {
    e.printStackTrace();
} catch (SAXException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}
```

### 5.4.4 サンプルプログラム ( SampleDOM.java ) の実行結果

このサンプルプログラムの実行結果は標準出力，および sampleDOM.xml に出力されます。

標準出力の内容を次に示します。

\$ 合計は315000円です。

sampleDOM.xml に出力される内容を次に示します。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<伝票>
<商品>
<商品名>テレビ</商品名>
<単価>15000</単価>
<数量>14</数量>
</商品>
<商品>
<商品名>ラジオ</商品名>
<単価>3500</単価>
<数量>6</数量>
</商品>
<商品>
<商品名>ビデオ</商品名>
<単価>21000</単価>
<数量>4</数量>
</商品>
</伝票>
```

## 5.5 SAX パーサを使用するサンプルプログラム

この節では、SAX パーサを使用するサンプルプログラムについて説明します。

サンプルプログラムのインストール先、ファイル名は次のとおりです。

インストール先

Cosminexus XML Processor をインストールしたディレクトリ /samples/SAX

ファイル名

SampleSAX.xml, および SampleSAX.java



### 注意事項

サンプルプログラムの実行ファイルの作成, および実行は, SAX ディレクトリ下で行ってください。

### 5.5.1 サンプルプログラムの処理

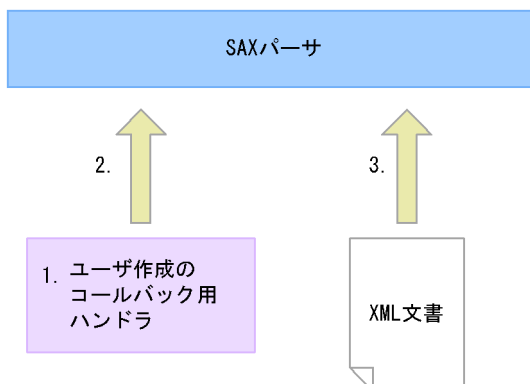
サンプルプログラムは次のような処理を行います。

1. 商品の情報について記述した XML 文書を読み込みます。
2. 読み込んだ XML 文書から, 商品の合計金額を画面に出力します。

### 5.5.2 サンプルプログラムの作成の流れ

SAX パーサを使用するサンプルプログラムの作成の流れを図 5-3 に示します。

図 5-3 SAX パーサを使用するサンプルプログラムの作成の流れ



1. コールバック用ハンドラを作成します。

## 5. プログラムの作成方法

SAX パーサを使用して解析を開始すると、SAX パーサから XML 文書内の情報が通知されます。これを受け取るための、コールバック用ハンドラを作成します。

### 2. コールバック用ハンドラを SAX パーサに登録します。

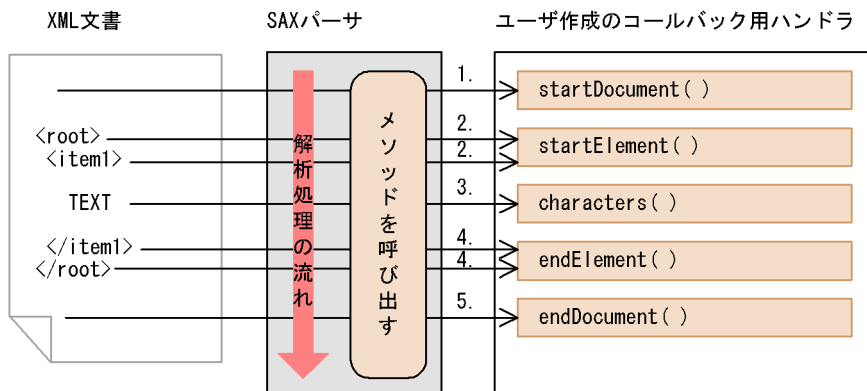
1. で作成したコールバック用ハンドラを SAX パーサに登録します。

### 3. XML 文書を SAX パーサに渡して解析します。

XML 文書を解析します。

#### SAX パーサが呼び出すメソッド

SAX パーサが呼び出すメソッドを次の図に示します。



#### 1. startDocumentメソッド

XML 文書の解析開始時に、startDocumentメソッドが呼び出されます。  
このメソッド内に必要な初期化処理を記述してください。

#### 2. startElementメソッド

解析中に要素の開きタグが見つかるたびに、startElementメソッドが呼び出されます。

#### 3. charactersメソッド

XML 文書内にタグに囲まれたテキストが見つかった場合、charactersメソッドが呼び出され、文字列を取得します。

#### 4. endElementメソッド

解析中に閉じタグが見つかるたびに、endElementメソッドが呼び出されます。

#### 5. endDocumentメソッド

解析終了時に、endDocumentメソッドが呼び出されます。

### 5.5.3 使用する XML 文書 ( SampleSAX.xml )

使用する XML 文書を次に示します。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<伝票>
  <商品>
    <商品名>テレビ</商品名>
    <単価>15000</単価>
    <数量>14</数量>
```

```

</商品>
<商品>
  <商品名>ラジオ</商品名>
  <単価>3500</単価>
  <数量>6</数量>
</商品>
<商品>
  <商品名>ビデオ</商品名>
  <単価>21000</単価>
  <数量>4</数量>
</商品>
</伝票>

```

### 5.5.4 サンプルプログラム ( SampleSAX.java )

サンプルプログラムを次に示します。

```

import javax.xml.parsers.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import org.xml.sax.SAXException;
import java.io.*;

public class SampleSAX{

    public static final void main(String[] args){
        try{
            SAXParserFactory factory = SAXParserFactory.newInstance();
            SAXParser parser = factory.newSAXParser();
            String xml = "SampleSAX.xml";
            DefaultHandler handler = new EventHandler();

            parser.parse(xml, handler);
        }catch(ParserConfigurationException e){
            e.printStackTrace();
        }catch(IOException e){
            e.printStackTrace();
        }catch(SAXException e){
            e.printStackTrace();
        }
    }
}

//合計金額を計算して出力する
class EventHandler extends DefaultHandler{

    int      tvPrice;          //テレビ単価
    int      radioPrice;       //ラジオ単価
    int      videoPrice;       //ビデオ単価
    int      tvNum;            //テレビ数量
    int      radioNum;         //ラジオ数量
    int      videoNum;         //ビデオ数量
    boolean  tvFlag;           //テレビ用フラグ
    boolean  radioFlag;        //ラジオ用フラグ
    boolean  videoFlag;        //ビデオ用フラグ

```

## 5. プログラムの作成方法

```
boolean tankaFlag;        //単価タグ内用フラグ
boolean volFlag;          //数量タグ内用フラグ
boolean nameFlag;         //商品名タグ内用フラグ

String str;               //要素内容の文字列
StringBuffer buffer;      //要素内容の文字列連結用バッファ

//XMLの開始通知イベント
public void startDocument() {
    tvPrice = 0;
    radioPrice = 0;
    videoPrice = 0;
    tvNum = 0;
    radioNum = 0;
    videoNum = 0;
    tvFlag = false;
    radioFlag = false;
    videoFlag = false;
    tankaFlag = false;
    volFlag = false;
    nameFlag = false;
}

//XMLの終了通知イベント
public void endDocument() {
    int sum = tvNum * tvPrice + radioNum * radioPrice +
        videoNum * videoPrice;
    System.out.println("合計は" + sum + "円です。");
}

//エレメント開始(タグの開始)通知イベント
public void startElement(String namespace, String localName,
    String modName, Attributes attr) {
    buffer = new StringBuffer();
    if(modName.equals("単価")){
        tankaFlag = true;
    }
    if(modName.equals("数量")){
        volFlag = true;
    }
    if(modName.equals("商品名")){
        nameFlag = true;
    }
}

//エレメント終了(タグの終了)通知イベント
public void endElement(String namespace, String localName,
    String modName) {
    str = buffer.toString();
    if(nameFlag == true){
        if(str.equals("テレビ")){
            tvFlag = true;
        }
        if(str.equals("ラジオ")){
            radioFlag = true;
        }
        if(str.equals("ビデオ")){
            videoFlag = true;
        }
    }
}
```

```

    }
} else if (tvFlag == true) {
    if (tankaFlag == true) {
        //テレビの単価をセット
        tvPrice = Integer.parseInt(str);
    } else if (volFlag == true) {
        //テレビの数量をセット
        tvNum = Integer.parseInt(str);
    }
} else if (radioFlag == true) {
    if (tankaFlag == true) {
        //ラジオの単価をセット
        radioPrice = Integer.parseInt(str);
    } else if (volFlag == true) {
        //ラジオの数量をセット
        radioNum = Integer.parseInt(str);
    }
} else if (videoFlag == true) {
    if (tankaFlag == true) {
        //ビデオの単価をセット
        videoPrice = Integer.parseInt(str);
    } else if (volFlag == true) {
        //ビデオの数量をセット
        videoNum = Integer.parseInt(str);
    }
}
}
if (modName.equals("単価")) {
    tankaFlag = false;
}
if (modName.equals("数量")) {
    volFlag = false;
}
if (modName.equals("商品名")) {
    nameFlag = false;
}
if (modName.equals("商品")) {
    tvFlag = false;
    radioFlag = false;
    videoFlag = false;
}
}

//文字列通知イベント
public void characters(char[] ch, int start, int length) {
    buffer.append(ch, start, length);
}
}

```

### 5.5.5 サンプルプログラム ( SampleSAX.java ) の実行結果

このサンプルプログラムの実行結果は標準出力に出力されます。標準出力の内容を次に示します。

\$ 合計は315000円です。



## 5.6 XML Schema を使用するサンプルプログラム

この節では、XML Schema を使用するサンプルプログラムについて説明します。

サンプルプログラムのインストール先、ファイル名は次のとおりです。

インストール先

Cosminexus XML Processor をインストールしたディレクトリ /samples/  
XMLSchema

ファイル名

purchaseOrder.xml, purchaseOrder-fail.xml, purchaseOrder.xsd,  
personalData.xsd, SampleValidateDOM.java, および SampleValidateSAX.java

### ！ 注意事項

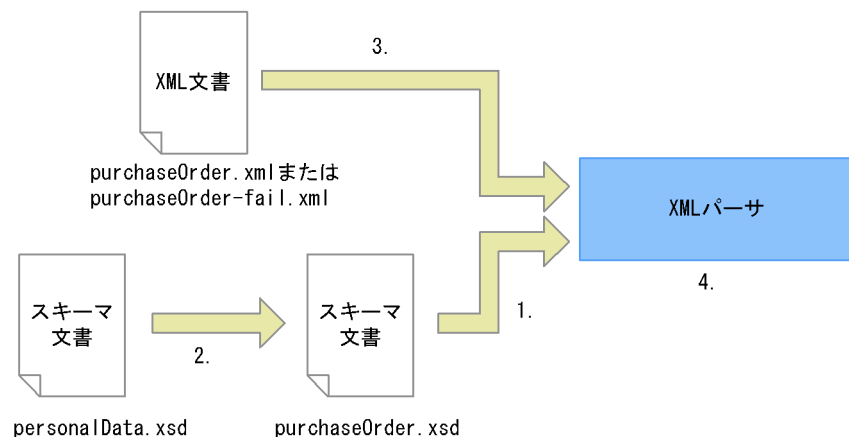
サンプルファイルの実行ファイルの作成、および実行は、XMLSchema ディレクトリ下で行ってください。

### 5.6.1 サンプルプログラムの処理

サンプルプログラムでは、XML 文書として作成された注文明細書の妥当性を検証します。注文明細書は顧客の個人データを共有して使用しているため、注文明細書のスキーマ文書は、個人データのスキーマ文書を読み込みます。なお、スキーマ文書は java プログラム内で特定します。

XML Schema を使用するサンプルプログラムの処理の流れを図 5-4 に示します。

図 5-4 XML Schema を使用するサンプルプログラムの処理の流れ



サンプルプログラムは次のような処理を行います。

1. スキーマ文書 purchaseOrder.xsd を読み込みます。  
purchaseOrder.xsd は、商品の注文明細書を規定するスキーマ文書です。
2. purchaseOrder.xsd は、スキーマ文書 personalData.xsd を読み込みます。  
personalData.xsd は、注文した顧客の個人データを規定するスキーマ文書で、purchaseOrder.xsd とは異なる対象名前空間を持ちます。
3. XML 文書 purchaseOrder.xml (または purchaseOrder-fail.xml) を読み込みます。  
purchaseOrder.xml, および purchaseOrder-fail.xml は、検証される XML 文書となる注文明細書です。
4. XML 文書の妥当性の検証を行います。  
XML 文書がスキーマ文書に対して妥当な場合は「Validation OK」、妥当ではない場合はメッセージとともに「Validation NG」と標準出力に表示します。

### 5.6.2 使用する XML 文書 ( purchaseOrder.xml , purchaseOrder-fail.xml )

Cosminexus XML Processor は、XML Schema による検証の対象となる XML 文書として、検証に成功する XML 文書 ( purchaseOrder.xml ), および検証に失敗する XML 文書 ( purchaseOrder-fail.xml ) を提供します。

#### ( 1 ) XML 文書 ( purchaseOrder.xml )

検証に成功する XML 文書 ( purchaseOrder.xml ) を次に示します。

```
<?xml version="1.0"?>
<po:purchaseOrder
  xmlns:po="http://www.myshopping.com/schema/purchaseOrder"
  xmlns:psd="http://www.myshopping.com/schema/personalData">
  <po:shipTo age="20">
    <psd:firstName>John</psd:firstName>
    <psd:familyName>Doe</psd:familyName>
    <psd:occupation>accountant</psd:occupation>
    <psd:email>johnD@bpl.com</psd:email>
    <psd:tel>050-1234-1234</psd:tel>
    <psd:address country="US">
      <psd:street>Universal street 100</psd:street>
      <psd:city>Carson</psd:city>
      <psd:state>Nevada</psd:state>
      <psd:zip>10-456</psd:zip>
    </psd:address>
  </po:shipTo>
  <po:billTo age="20">
    <psd:firstName>Jane</psd:firstName>
    <psd:familyName>Doe</psd:familyName>
    <psd:occupation>lawyer</psd:occupation>
```

```

<psd:email>janeD@bpl.com</psd:email>
<psd:tel>033-1111-2345</psd:tel>
<psd:address country="US">
  <psd:street>Times Square 555</psd:street>
  <psd:city>New York</psd:city>
  <psd:state>New York</psd:state>
  <psd:zip>155-5600</psd:zip>
</psd:address>
<po:credit year="2007" month="10">
  <po:creditHolder>Jane Doe</po:creditHolder>
  <po:creditNumber>111144442229999</po:creditNumber>
  <po:creditCompany>CardCompanyA</po:creditCompany>
</po:credit>
</po:billTo>
<po:items>
  <po:item>
    <po:productName
      productID="DTPC2000S">DeskTop PC</po:productName>
    <po:quantity>1</po:quantity>
    <po:price>1500</po:price>
    <po:shipDate>2004-05-20</po:shipDate>
  </po:item>
  <po:item>
    <po:productName
      productID="DC500MP">Digital Camera</po:productName>
    <po:quantity>1</po:quantity>
    <po:price>450</po:price>
    <po:shipDate>2004-05-20</po:shipDate>
  </po:item>
  <po:item>
    <po:productName
      productID="LP800S">Printer</po:productName>
    <po:quantity>1</po:quantity>
    <po:price>200</po:price>
    <po:shipDate>2004-05-20</po:shipDate>
  </po:item>
</po:items>
</po:purchaseOrder>

```

## ( 2 ) XML 文書 ( purchaseOrder-fail.xml )

検証に失敗する XML 文書 ( purchaseOrder-fail.xml ) の一部を次に示します。

```

:
:
<po:credit year="2007" month="10">
  <po:creditHolder>Jane Doe</po:creditHolder>
  <po:creditNumber>11114444222999955</po:creditNumber>
  <po:creditCompany>CardCompanyA</po:creditCompany>
</po:credit>
:
:

```

この XML 文書は , <po:creditNumber> で始まるクレジット番号の定義以外は「5.6.2(1)

## 5. プログラムの作成方法

XML 文書 ( purchaseOrder.xml )」と同じです。<po:creditNumber> で始まるクレジット番号の値として 18 桁の数字を使用していますが、サンプルのスキーマ文書ではクレジット番号を 16 桁で規定しているため、検証時にエラーとなります。

この XML 文書を使用した場合の実行結果については「5.6.6 検証に失敗した場合の実行結果」を参照してください。

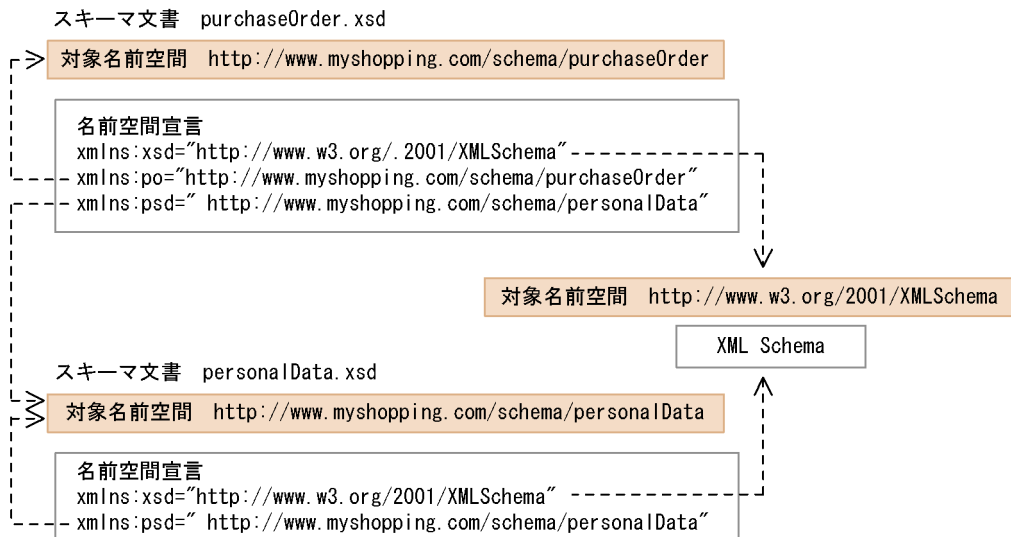
### 5.6.3 使用するスキーマ文書 ( purchaseOrder.xsd , personalData.xsd )

このサンプルでは、スキーマ文書として、purchaseOrder.xsd、および personalData.xsd を使用します。

#### (1) スキーマ文書での各名前空間の関係

purchaseOrder.xsd、および personalData.xsd での各名前空間の関係を図 5-5 に示します。それぞれの矢印は、参照する型定義や要素などが定義された対象名前空間を指しています。

図 5-5 スキーマ文書の名前空間の関係



#### (2) スキーマ文書 ( purchaseOrder.xsd )

使用するスキーマ文書 ( purchaseOrder.xsd ) を次に示します。

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.myshopping.com/schema/purchaseOrder"
  xmlns:po="http://www.myshopping.com/schema/purchaseOrder"
```

```

xmlns:psd="http://www.myshopping.com/schema/personalData">

<xsd:import
  namespace="http://www.myshopping.com/schema/personalData"
  schemaLocation="personalData.xsd"/>

<xsd:element name="purchaseOrder" type="po:purchaseOrderType"/>
<xsd:element name="shipTo" type="psd:personalDataType"/>
<xsd:element name="billTo" type="po:billToPersonalDataType"/>
<xsd:element name="items" type="po:itemsType">
  <xsd:key name="productID_unique">
    <xsd:selector xpath="po:item/po:productName"/>
    <xsd:field xpath="@productID"/>
  </xsd:key>
</xsd:element>

<xsd:element name="item" type="po:itemType"/>
<xsd:element name="productName" type="po:productNameType"/>
<xsd:element name="quantity">
  <xsd:simpleType>
    <xsd:restriction base="xsd:positiveInteger">
      <xsd:maxExclusive value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<xsd:element name="price" type="xsd:positiveInteger"/>
<xsd:element name="shipDate" type="xsd:date"/>
<xsd:element name="credit" type="po:creditType"/>
<xsd:element name="creditHolder" type="xsd:string"/>
<xsd:element name="creditNumber">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="^\d{16}$"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<xsd:element name="creditCompany">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="CardCompanyA"/>
      <xsd:enumeration value="CardCompanyB"/>
      <xsd:enumeration value="CardCompanyC"/>
      <xsd:enumeration value="CardCompanyD"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<xsd:complexType name="purchaseOrderType">
  <xsd:sequence>
    <xsd:element ref="po:shipTo"/>
    <xsd:element ref="po:billTo"/>
    <xsd:element ref="po:items"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="billToPersonalDataType">

```

## 5. プログラムの作成方法

```
<xsd:sequence>
  <xsd:group ref="psd:personalDataGroup"/>
  <xsd:element ref="po:credit"/>
</xsd:sequence>
<xsd:attributeGroup ref="psd:personalAttributeGroup"/>
</xsd:complexType>

<xsd:complexType name="itemsType">
  <xsd:sequence>
    <xsd:element ref="po:item"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="itemType">
  <xsd:sequence>
    <xsd:element ref="po:productName"/>
    <xsd:element ref="po:quantity"/>
    <xsd:element ref="po:price"/>
    <xsd:element ref="po:shipDate"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="productNameType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="productID" type="xsd:string"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="creditType">
  <xsd:sequence>
    <xsd:element ref="po:creditHolder"/>
    <xsd:element ref="po:creditNumber"/>
    <xsd:element ref="po:creditCompany"/>
  </xsd:sequence>
  <xsd:attribute name="year" type="po:yearType" use="required"/>
  <xsd:attribute name="month" type="po:monthType" use="required"/>
</xsd:complexType>

<xsd:simpleType name="yearType">
  <xsd:annotation>
    <xsd:documentation>
      Year set from 2004 to 2008
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="200[4-8]"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="monthType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="0[1-9]|1[0-2]"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
</xsd:schema>
```

### (3) スキーマ文書 (personalData.xsd)

使用するスキーマ文書 (personalData.xsd) を次に示します。

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.myshopping.com/schema/personalData"
  xmlns:psd="http://www.myshopping.com/schema/personalData">

  <xsd:element name="personalData" type="psd:personalDataType"/>
  <xsd:element name="firstName" type="xsd:string"/>
  <xsd:element name="familyName" type="xsd:string"/>
  <xsd:element name="occupation" type="xsd:string"/>
  <xsd:element name="email" type="xsd:string"/>
  <xsd:element name="tel" type="xsd:string"/>
  <xsd:element name="address" type="psd:addressType"/>
  <xsd:element name="building" type="xsd:string"/>
  <xsd:element name="street" type="xsd:string"/>
  <xsd:element name="city" type="xsd:string"/>
  <xsd:element name="state" type="xsd:string"/>
  <xsd:element name="zip" type="xsd:string"/>

  <xsd:complexType name="personalDataType">
    <xsd:group ref="psd:personalDataGroup"/>
    <xsd:attributeGroup ref="psd:personalAttributeGroup"/>
  </xsd:complexType>

  <xsd:group name="personalDataGroup">
    <xsd:sequence>
      <xsd:element ref="psd:firstName"/>
      <xsd:element ref="psd:familyName"/>
      <xsd:element ref="psd:occupation"/>
      <xsd:element ref="psd:email"/>
      <xsd:element ref="psd:tel"/>
      <xsd:element ref="psd:address"/>
    </xsd:sequence>
  </xsd:group>

  <xsd:attributeGroup name="personalAttributeGroup">
    <xsd:attribute name="age"
      type="xsd:positiveInteger" use="optional"/>
    <xsd:attribute name="company"
      type="xsd:string" use="optional"/>
    <xsd:attribute name="department"
      type="xsd:string" use="optional"/>
    <xsd:attribute name="title"
      type="xsd:string" use="optional"/>
    <xsd:attribute name="fax"
      type="xsd:string" use="optional"/>
  </xsd:attributeGroup>

  <xsd:complexType name="addressType">
    <xsd:sequence>
      <xsd:element ref="psd:building" minOccurs="0"/>
```

## 5. プログラムの作成方法

```
<xsd:element ref="psd:street"/>
<xsd:element ref="psd:city"/>
<xsd:element ref="psd:state"/>
<xsd:element ref="psd:zip"/>
</xsd:sequence>
<xsd:attribute name="country" type="xsd:string"
  use="optional" default="US"/>
</xsd:complexType>
</xsd:schema>
```

### 5.6.4 DOM パーサを使用する場合のサンプルプログラム

DOM パーサを使用して XML 文書を検証する場合のサンプルプログラム、サンプルプログラムの実行方法、および実行結果について説明します。

#### (1) サンプルプログラム (SampleValidateDOM.java)

DOM パーサを使用する場合のサンプルプログラム (SampleValidateDOM.java) を次に示します。

```
import javax.xml.parsers.*;
import org.w3c.dom.*;
import org.xml.sax.*;
import java.io.*;

public class SampleValidateDOM implements ErrorHandler{
    public static final void main(String[] argv){
        if (argv.length != 2) {
            System.out.println(
                "Usage: java SampleValidateDOM <xml_file> <schema_file>");
            System.exit(1);
        }

        try{
            DocumentBuilderFactory dbf =
                DocumentBuilderFactory.newInstance();
            dbf.setNamespaceAware(true);
            dbf.setValidating(true);
            dbf.setAttribute(
                "http://java.sun.com/xml/jaxp/properties/schemaLanguage",
                "http://www.w3.org/2001/XMLSchema");
            dbf.setAttribute(
                "http://java.sun.com/xml/jaxp/properties/schemaSource",
                argv[1]);

            DocumentBuilder db = dbf.newDocumentBuilder();
            db.setErrorHandler(new SampleValidateDOM());
            Document doc = db.parse(argv[0]);
            System.out.println("Validation OK");

        }catch(ParserConfigurationException e){
            e.printStackTrace();
        }catch(SAXParseException e){
```



```

        e.printStackTrace();
    } catch (SAXException e) {
        System.out.println("Validation NG: " + e.getMessage());
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public void warning(SAXParseException exception)
    throws SAXException {
    System.out.println("**Parsing Warning**%n" +
        " Line:      " +
        exception.getLineNumber() + "%n" +
        " URI:        " +
        exception.getSystemId() + "%n" +
        " Message: " +
        exception.getMessage());
}

public void error(SAXParseException exception)
    throws SAXException {
    System.out.println("**Parsing Error**%n" +
        " Line:      " +
        exception.getLineNumber() + "%n" +
        " URI:        " +
        exception.getSystemId() + "%n" +
        " Message: " +
        exception.getMessage());
    throw new SAXException("Error encountered");
}

public void fatalError(SAXParseException exception)
    throws SAXException {
    System.out.println("**Parsing Fatal Error**%n" +
        " Line:      " +
        exception.getLineNumber() + "%n" +
        " URI:        " +
        exception.getSystemId() + "%n" +
        " Message: " +
        exception.getMessage());
    throw new SAXException("Fatal Error encountered");
}
}

```

### ❗ 注意事項

サンプルプログラムは、SampleValidateDOM.java があるディレクトリで実行します。

## (2) サンプルプログラムの実行結果

このサンプルプログラムの実行結果は標準出力に出力されます。標準出力の内容を次に示します。

Validation OK

### 5.6.5 SAX パーサを使用する場合のサンプルプログラム

SAX パーサを使用して XML 文書を検証する場合のサンプルプログラム、サンプルプログラムの実行方法、および実行結果について説明します。

#### (1) サンプルプログラム (SampleValidateSAX.java)

SAX パーサを使用する場合のサンプルプログラム (SampleValidateSAX.java) を次に示します。

```
import javax.xml.parsers.*;
import org.xml.sax.*;
import java.io.*;

public class SampleValidateSAX implements ErrorHandler{
    public static final void main(String[] argv){
        if (argv.length != 2) {
            System.out.println(
                "Usage: java SampleValidateSAX <xml_file> <schema_file>");
            System.exit(1);
        }

        try{
            SAXParserFactory spf = SAXParserFactory.newInstance();
            spf.setNamespaceAware(true);
            spf.setValidating(true);
            SAXParser sp = spf.newSAXParser();
            sp.setProperty(
                "http://java.sun.com/xml/jaxp/properties/schemaLanguage",
                "http://www.w3.org/2001/XMLSchema");
            sp.setProperty(
                "http://java.sun.com/xml/jaxp/properties/schemaSource",
                argv[1]);

            XMLReader reader = sp.getXMLReader();
            reader.setErrorHandler(new SampleValidateSAX());
            reader.parse(argv[0]);
            System.out.println("Validation OK");

        }catch(ParserConfigurationException e){
            e.printStackTrace();
        }catch(SAXParseException e){
            e.printStackTrace();
        }catch(SAXException e){
            System.out.println("Validation NG: " + e.getMessage());
        }catch(IOException e){
            e.printStackTrace();
        }
    }

    public void warning(SAXParseException exception)
```

```

        throws SAXException {
    System.out.println("**Parsing Warning**%n" +
        "   Line:      " +
        exception.getLineNumber() + "%n" +
        "   URI:         " +
        exception.getSystemId() + "%n" +
        "   Message: " +
        exception.getMessage());
    }

    public void error(SAXParseException exception)
        throws SAXException{
    System.out.println("**Parsing Error**%n" +
        "   Line:      " +
        exception.getLineNumber() + "%n" +
        "   URI:         " +
        exception.getSystemId() + "%n" +
        "   Message: " +
        exception.getMessage());
    throw new SAXException("Error encountered");
    }

    public void fatalError(SAXParseException exception)
        throws SAXException {
    System.out.println("**Parsing Fatal Error**%n" +
        "   Line:      " +
        exception.getLineNumber() + "%n" +
        "   URI:         " +
        exception.getSystemId() + "%n" +
        "   Message: " +
        exception.getMessage());
    throw new SAXException("Fatal Error encountered");
    }
}

```

### ！ 注意事項

サンプルプログラムは、SampleValidateSAX.java があるディレクトリで実行します。

## (2) サンプルプログラムの実行結果

このサンプルプログラムの実行結果は標準出力に出力されます。標準出力の内容を次に示します。

```
Validation OK
```

### 5.6.6 検証に失敗した場合の実行結果

サンプルの XML 文書 purchaseOrder-fail.xml を対象に検証を行うと、検証に失敗します。サンプルファイルの実行方法は、「5.6.4 DOM パーサを使用する場合のサンプルプ

## 5. プログラムの作成方法

プログラム」, および「5.6.5 SAX パーサを使用する場合のサンプルプログラム」と同じです。

検証に失敗した場合, エラーが標準出力に出力されます。標準出力の内容を次に示します。

```
**Parsing Error**
  Line:      33
  URI:       file:///サンプルが格納されたディレクトリ/
purchaseOrder-fail.xml
  Message: KECX06063-E cvc-pattern-valid: Value
'111144442222999955' is not facet-valid with respect to pattern
'¥d{16}' for type '#AnonType_creditNumber'.
Validation NG: Error encountered
```

## 5.7 XSLT トランスフォーマを使用するサンプルプログラム

---

この節では、XSLT トランスフォーマを使用するサンプルプログラムについて説明します。

サンプルプログラムのインストール先、ファイル名は次のとおりです。

インストール先

Cosminexus XML Processor をインストールしたディレクトリ /samples/XSLT

ファイル名

SampleXSLT.xml , SampleXSLT.xsl , および SampleXSLT.java

### ! 注意事項

サンプルプログラムの実行ファイルの作成、および実行は、XSLT ディレクトリ下で行ってください。

---

### 5.7.1 サンプルプログラムの処理

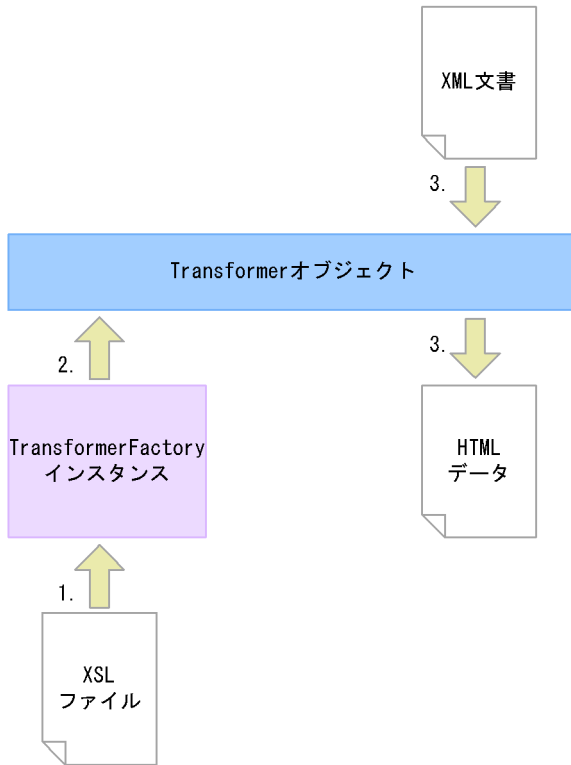
サンプルプログラムは次のような処理を行います。

1. 変換元の XML 文書を読み込みます。
2. 1. で読み込んだ XML 文書を HTML データへ変換します。

### 5.7.2 サンプルプログラムの作成の流れ

XSLT トランスフォーマを使用するサンプルプログラムの作成の流れを図 5-6 に示します。

図 5-6 XSLT トランスフォーマーを使用するサンプルプログラムの作成の流れ



1. TransformerFactory インスタンスを取得します。  
メソッド `newInstance` で TransformerFactory インスタンスを取得します。
2. Transformer オブジェクトを作成します。
3. HTML データへの変換を行います。  
Transformer オブジェクトを使用して XML 文書を HTML データに変換します。

### 5.7.3 使用する XML 文書 ( SampleXSLT.xml )

サンプルプログラムが読み込む XML 文書を次に示します。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<?xml-stylesheet type="text/xsl" href="SampleXSLT.xsl" ?>
<START>
  <伝票>
    <商品>
      <商品名>テレビ</商品名>
      <単価>15000</単価>
      <数量>14</数量>
    </商品>
  </商品>
```

```

    <商品名>ラジオ</商品名>
    <単価>3500</単価>
    <数量>6</数量>
  </商品>
  <商品>
    <商品名>ビデオ</商品名>
    <単価>21000</単価>
    <数量>4</数量>
  </商品>
</伝票>
</START>

```

## 5.7.4 使用する XSL ファイル ( SampleXSLT.xsl )

サンプルプログラムが読み込む XSL ファイルを次に示します。

```

<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="html" indent="yes" encoding="Shift_JIS"/>

  <xsl:template match="/">
    <xsl:apply-templates />
  </xsl:template>

  <!-- STARTをルートとして変換を開始する -->
  <xsl:template match="START">
    <HTML>
      <HEAD>
        <TITLE>XMLをHTMLのテーブルで表示する</TITLE>
      </HEAD>
      <BODY BGCOLOR="#C0C0C0">
        <FONT SIZE="5">発注伝票</FONT>
        <TABLE BORDER="2">
          <TR><TH>商品名</TH><TH>単価</TH>
            <TH>数量</TH><TH>小計</TH></TR>
          <xsl:apply-templates />
          <TR><TH>合計</TH><TD>&#160;</TD><TD>&#160;</TD>
            <TD ALIGN="RIGHT"><xsl:value-of
              select="(//商品[1]/単価 * //商品[1]/数量 +
                //商品[2]/単価 * //商品[2]/数量 +
                //商品[3]/単価 * //商品[3]/数量)" /></TD></TR>
          </TABLE>
        </BODY>
      </HTML>
    </xsl:template>

    <xsl:template match="伝票">
      <xsl:apply-templates />
    </xsl:template>

    <!-- 1商品で1行を作成するし、最後に小計を出力する -->
    <xsl:template match="商品">
      <TR><xsl:apply-templates />
        <TD ALIGN="RIGHT">

```

## 5. プログラムの作成方法

```
        <xsl:value-of select="number(単価)*number(数量)" />
    </TD></TR>
</xsl:template>

<!-- 商品名を出力する -->
<xsl:template match="商品名">
    <TD><xsl:value-of select="."/></TD>
</xsl:template>

<!-- 単価を出力する -->
<xsl:template match="単価">
    <TD ALIGN="RIGHT"><xsl:value-of select="."/></TD>
</xsl:template>

<!-- 数量を出力する -->
<xsl:template match="数量">
    <TD ALIGN="RIGHT"><xsl:value-of select="."/></TD>
</xsl:template>
</xsl:stylesheet>
```

### 5.7.5 サンプルプログラム ( SampleXSLT.java )

サンプルプログラムを次に示します。

```
import java.io.*;
import javax.xml.transform.*;
import javax.xml.transform.stream.*;
import javax.xml.parsers.*;

public class SampleXSLT{
    public static void main(String[] args){
        try{
            File file = new File(args[0]);

            //XMLファイルの読み込み
            Source source = new StreamSource(file);
            Result result = new StreamResult(System.out);
            TransformerFactory factory =
                TransformerFactory.newInstance();

            //HTMLファイルへの変換
            Source style = factory.getAssociatedStylesheet(source,
                null, null, null);

            Transformer transformer = factory.newTransformer(style);
            transformer.transform(source, result);

        } catch(TransformerConfigurationException e){
            e.printStackTrace();
        } catch(TransformerException e){
            e.printStackTrace();
        }
    }
}
```



**!** 注意事項

サンプルプログラムは、SampleXSLT.java があるディレクトリで実行します。

## 5.7.6 サンプルプログラム ( SampleXSLT.java ) の実行結果

このサンプルプログラムの実行結果は標準出力に出力されます。標準出力の内容を次に示します。

```
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html;
charset=Shift_JIS">
<TITLE>XMLをHTMLのテーブルで表示する</TITLE>
</HEAD>
<BODY BGCOLOR="#C0C0C0">
<FONT SIZE="5">発注伝票</FONT>
<TABLE BORDER="2">
<TR>
<TH>商品名</TH><TH>単価</TH><TH>数量</TH><TH>小計</TH>
</TR>
<TR>
<TD>テレビ</TD>
<TD ALIGN="RIGHT">15000</TD>
<TD ALIGN="RIGHT">14</TD>
<TD ALIGN="RIGHT">210000</TD>
</TR>
<TR>
<TD>ラジオ</TD>
<TD ALIGN="RIGHT">3500</TD>
<TD ALIGN="RIGHT">6</TD>
<TD ALIGN="RIGHT">21000</TD>
</TR>
<TR>
<TD>ビデオ</TD>
<TD ALIGN="RIGHT">21000</TD>
<TD ALIGN="RIGHT">4</TD>
<TD ALIGN="RIGHT">84000</TD>
</TR>
<TR>
<TH>合計</TH><TD>&nbsp;</TD><TD>&nbsp;</TD><TD>
ALIGN="RIGHT">315000</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

## 5.8 XSLTC トランスフォーマーを使用するサンプルプログラム

XSLTC トランスフォーマーを使用するプログラムについて説明します。

ここでは、「5.7 XSLT トランスフォーマーを使用するサンプルプログラム」を変更して、XSLTC トランスフォーマー機能を使用する方法を示します。

XSLTC トランスフォーマー機能を使用する場合のプログラム変更を図 5-7 に示します。図 5-7 に示す変更により、XSLTC トランスフォーマー機能を使用できます。プログラムの実行方法、実行結果は「5.7 XSLT トランスフォーマーを使用するサンプルプログラム」と同じです。

図 5-7 XSLTC トランスフォーマー機能を使用する場合のプログラム変更

(変更前)

```
import java.io.*;
import javax.xml.transform.*;
import javax.xml.transform.stream.*;
import javax.xml.parsers.*;

public class SampleXSLT{
    public static void main(String[] args){

        try{
            File file = new File(args[0]);

            //XMLファイルの読み込み
            Source source = new StreamSource(file);
            Result result = new StreamResult(System.out);
            TransformerFactory factory =
                TransformerFactory.newInstance();

            //HTMLファイルへの変換
            Source style = factory.getAssociatedStylesheet(source,
                null, null, null);

            Transformer transformer = factory.newTransformer(style);
            transformer.transform(source, result);

        }catch(TransformerConfigurationException e){
            e.printStackTrace();
        }catch(TransformerException e){
            e.printStackTrace();
        }
    }
}
```

(変更後)

```
import java.io.*;
import javax.xml.transform.*;
import javax.xml.transform.stream.*;
import javax.xml.parsers.*;
import com.cosminexus.jaxp.xsltc.*; //...1

public class SampleXSLT{
    public static void main(String[] args){

        try{
            File file = new File(args[0]);

            //XMLファイルの読み込み
            Source source = new StreamSource(file);
            Result result = new StreamResult(System.out);
            TransformerFactory factory =
                TransformerFactoryXSLTC.newInstance(); //...2

            //HTMLファイルへの変換
            Source style = factory.getAssociatedStylesheet(source,
                null, null, null);

            Transformer transformer = factory.newTransformer(style);
            transformer.transform(source, result);

        }catch(TransformerConfigurationException e){
            e.printStackTrace();
        }catch(TransformerException e){
            e.printStackTrace();
        }
    }
}
```

プログラムの変更箇所について次に説明します。各項番はプログラムのコメントに記述している番号に対応しています。

1. XSLTC トランスフォーマー機能を使用するためのパッケージを import します。
2. TransformerFactoryXSLTC クラスに newInstance メソッドを適用して、TransformerFactory オブジェクトを生成します。

# 6

## Cosminexus XML Processor 使用時の注意事項

この章では、Cosminexus XML Processor をご使用になる際の注意事項について説明します。

- 
- 6.1 JAXP1.3 機能の共通の注意事項
  - 6.2 DOM パーサに関する注意事項
  - 6.3 SAX パーサに関する注意事項
  - 6.4 スキーマ検証に関する注意事項
  - 6.5 XSLT/XSLTC 共通の注意事項
  - 6.6 XSLT に関する注意事項
  - 6.7 XSLTC に関する注意事項
  - 6.8 javax.xml.datatype パッケージに関する注意事項
  - 6.9 javax.xml.validation パッケージに関する注意事項
  - 6.10 javax.xml.xpath パッケージに関する注意事項
  - 6.11 org.w3c.dom パッケージに関する注意事項
  - 6.12 org.w3c.dom.bootstrap パッケージに関する注意事項
  - 6.13 org.w3c.dom.ls パッケージに関する注意事項
  - 6.14 org.xml.sax.ext パッケージに関する注意事項
  - 6.15 XInclude に関する注意事項

## 6. Cosminexus XML Processor 使用時の注意事項

---

### 6.16 実装依存仕様に関する注意事項

---

### 6.17 高速パース機能に関する注意事項

---

## 6.1 JAXP1.3 機能の共通の注意事項

JAXP1.3 機能の共通の注意事項を表 6-1 に示します。

表 6-1 JAXP1.3 機能の共通の注意事項

項番	注意事項
1	JAXP1.3 仕様書で、引数に null を指定したときの動作が規定されていないメソッドがあります。このようなメソッドの引数に null を指定した場合、NullPointerException 例外が発生するなど、動作結果が未定義ですので注意してください。
2	JAXP1.3 仕様書が規定するメソッドで例外が発生したとき、例外オブジェクトに getMessage メソッドを適用することで、多くの場合は詳細メッセージが得られます。ただし、詳細メッセージが必ず得られることを保証するものではありません。したがって、アプリケーションでは、詳細メッセージの内容に依存した例外処理をしないで、例外の種別に応じて適切な処理をしてください。例えば、DOMException 例外の場合は、code フィールドでエラーの詳細要因を判定してください。
3	DocumentBuilder クラスの parse(String uri) メソッドや SAXParser クラスの parse(String uri, ...) メソッドなどの引数に指定する URI は、正しい形式で指定してください。URI として不正な文字列を指定すると、IllegalArgumentException などの RuntimeException 例外が発生する場合があります。
4	DocumentBuilder クラスの parse(File f) メソッドや SAXParser クラスの parse(File f, ...) メソッドなどの引数に指定する File オブジェクトは、引数に「%」「#」「?」を含まない File コンストラクタで生成してください。引数にこれらの文字が含まれる File コンストラクタで生成された File オブジェクトを指定すると、次に示す現象が発生する場合があります。 <ul style="list-style-type: none"> <li>• IllegalArgumentException などの RuntimeException 例外が発生する</li> <li>• SAXException 例外が発生する</li> <li>• XML パーサの動作が不正になる</li> </ul>

## 6.2 DOM パーサに関する注意事項

DOM パーサに関する注意事項を表 6-2 に示します。

表 6-2 DOM パーサに関する注意事項

項番	注意事項
1	<p>マルチスレッドプログラミングをする場合、DocumentBuilderFactory クラスはスレッドセーフではありません。したがって、複数のスレッドが同時に同一の DocumentBuilderFactory インスタンスにアクセスしてはいけません。スレッド間の競合を避けるため、次のどちらかの方法を使用してください。</p> <ul style="list-style-type: none"> <li>各スレッドに一つの DocumentBuilderFactory インスタンスを持つ。</li> <li>各スレッドが排他的に DocumentBuilderFactory インスタンスにアクセスする。</li> </ul>
2	<p>マルチスレッドプログラミングをする場合、DocumentBuilder クラスはスレッドセーフではありません。したがって、複数のスレッドが同時に同一の DocumentBuilder インスタンスを使用してはいけません。スレッド間の競合を避けるため、次の方法を使用してください。</p> <ul style="list-style-type: none"> <li>各スレッドごとに一つの DocumentBuilder インスタンスを持つ。</li> </ul>
3	<p>マルチスレッドプログラミングをする場合、DOM ツリーはスレッドセーフではありません。したがって、parse メソッドにより生成された同一の DOM ツリーに対して、複数のスレッドが同時にアクセスしてはいけません。更新系メソッドだけではなく、参照系メソッドでも同時にアクセスしてはいけません。スレッド間の競合を避けるため、次の方法を使用してください。</p> <ul style="list-style-type: none"> <li>各スレッドが排他的に DOM ツリーにアクセスする。</li> </ul>
4	<p>マルチスレッドプログラミングをする場合、org.w3c.dom、org.w3c.dom.bootstrap、org.w3c.dom.ls パッケージで規定されたオブジェクトはスレッドセーフではありません。したがって、複数のスレッドが同時にこれらのオブジェクトにアクセスしてはいけません。更新系メソッドだけではなく、参照系メソッドでも同時にアクセスしてはいけません。スレッド間の競合を避けるため、次の方法を使用してください。</p> <ul style="list-style-type: none"> <li>各スレッドが排他的にこれらのオブジェクトにアクセスする。</li> </ul>
5	<p>Document インタフェースの createAttribute メソッドまたは createAttributeNS メソッドを使用して Attr ノードを生成する場合は、Attr インタフェースの setValue メソッドで値を設定してください。値が設定されていない状態で、Node インタフェースの getChildNodes メソッドで NodeList を取得した場合、その NodeList の動作は保証しません。</p>
6	<p>Node インタフェースの insertBefore メソッドや appendChild メソッドを使用して、すでに Element ノードを持つ Document ノードに対して要素を追加する場合、Node インタフェースの replaceChild メソッドを使用した場合と同じ結果になります。</p>
7	<p>DocumentBuilder クラスの「InputStream や InputSource を引数とする parse メソッド」でエラーが発生したとき、エラーハンドラへ渡される SAXParseException に getSystemId メソッドを適用すると null が返される場合があります。エラー発生元のシステム識別子を返すようにする場合は、parse メソッドを次のように使用してください。</p> <ul style="list-style-type: none"> <li>parse(InputStream is, String systemId) メソッドで、システム識別子を引数 systemId に指定する。</li> <li>parse(InputSource is) メソッドで、システム識別子を設定した InputSource を引数 is に指定する。</li> </ul>
8	<p>BOM(Byte Order Mark) 付きの UTF-16 で保存された XML 文書を parse(InputSource is) メソッドで解析する場合、InputSource に setEncoding メソッドを適用するときは引数に "UTF-16" を指定してください。</p>

項番	注意事項
9	内部エンティティを定義する文字列中に、477 文字以上の要素名が含まれていると、 <code>java.lang.IndexOutOfBoundsException</code> 例外が発生する場合があります。

## 6.3 SAX パーサに関する注意事項

SAX パーサに関する注意事項を表 6-3 に示します。

表 6-3 SAX パーサに関する注意事項

項番	注意事項
1	<p>マルチスレッドプログラミングをする場合、SAXParserFactory クラスはスレッドセーフではありません。したがって、複数のスレッドが同時に同一の SAXParserFactory インスタンスにアクセスしてはいけません。スレッド間の競合を避けるため、次のどちらかの方法を使用してください。</p> <ul style="list-style-type: none"><li>各スレッドに一つの SAXParserFactory インスタンスを持つ。</li><li>各スレッドが排他的に SAXParserFactory インスタンスにアクセスする。</li></ul>
2	<p>マルチスレッドプログラミングをする場合、SAXParser クラスはスレッドセーフではありません。したがって、複数のスレッドが同時に同一の SAXParser インスタンスを使用してはいけません。スレッド間の競合を避けるため、次の方法を使用してください。</p> <ul style="list-style-type: none"><li>各スレッドに一つの SAXParser インスタンスを持つ。</li></ul>
3	<p>マルチスレッドプログラミングをする場合、org.xml.sax、org.xml.sax.ext、org.xml.sax.helpers パッケージで規定されたオブジェクトはスレッドセーフではありません。したがって、複数のスレッドが同時にこれらのオブジェクトにアクセスしてはいけません。更新系メソッドだけではなく、参照系メソッドでも同時にアクセスしてはいけません。スレッド間の競合を避けるため、次の方法を使用してください。</p> <ul style="list-style-type: none"><li>各スレッドが排他的にこれらのオブジェクトにアクセスする。</li></ul>
4	<p>SAX パーサは、一つの文字列データを複数の文字列（チャンク）に分割し、複数の characters イベントとしてアプリケーションに報告することが許されています。したがって、ContentHandler の実装では、文字列データが分割されることを意識する必要があります。ContentHandler の実装例を次に示します。この例では、characters イベントが発生するたびに文字列データをバッファリングしていき、endElement イベントが報告された時点で文字列データの終端とみなしています。</p> <pre>class MyHandler implements ContentHandler {     String str = null;     StringBuffer buffer = new StringBuffer();     :     public void characters(char c[], int start, int length) throws         SAXException {         buffer.append(c, start, length);     }     :     public void endElement(String uri, String name, String qname) throws         SAXException {         str = buffer.toString();     }     :</pre>



項番	注意事項
5	<p>XMLReaderFactory クラスの createXMLReader(String className) メソッドを使用し、XMLReader の生成を行う場合は、引数 className に "com.cosminexus.jaxp.impl.parsers.parsers.SAXParser" を指定してください。XMLReader の生成例を次に示します。</p> <pre>XMLReader reader = XMLReaderFactory.createXMLReader("com.cosminexus.jaxp.impl.parsers.parsers.SAXParser");</pre>
6	<p>SAXParser クラスの「InputStream や InputSource を引数とする parse メソッド」でエラーが発生したとき、エラーハンドラへ渡される SAXParseException に getSystemId メソッドを適用すると null が返される場合があります。エラー発生元のシステム識別子を返すようにする場合は、parse メソッドを次のように使用してください。</p> <ul style="list-style-type: none"> <li>• parse(InputStream is, ..., String systemId) メソッドで、システム識別子を引数 systemId に指定する。</li> <li>• parse(InputSource is, ...) メソッドで、システム識別子を設定した InputSource を引数 is に指定する。</li> </ul>
7	<p>BOM(Byte Order Mark) 付きの UTF-16 で保存された XML 文書を parse(InputSource is, ...) メソッドで解析する場合、InputSource に setEncoding メソッドを適用するときは引数に "UTF-16" を指定してください。</p>
8	<p>内部エンティティを定義する文字列中に、477 文字以上の要素名が含まれていると、java.lang.IndexOutOfBoundsException 例外が発生する場合があります。</p>

## 6.4 スキーマ検証に関する注意事項

スキーマ検証に関する注意事項を表 6-4 に示します。

表 6-4 スキーマ検証に関する注意事項

項番	注意事項
1	maxOccurs ファセットに大きな値を指定するとメモリを大量に消費します。maxOccurs ファセットには、可能な限り unbounded を指定してください。
2	XML パーサにエンティティリゾルバを登録する場合は、エンティティリゾルバの resolveEntity メソッドの戻り値である InputSource に必ずシステム識別子を設定してください。設定しないと、xsd:import や xsd:include で指定したスキーマ文書の場所の解決に失敗してエラーとなる場合があります。

## 6.5 XSLT/XSLTC 共通の注意事項

XSLT/XSLTC 共通の注意事項を表 6-5 に示します。

表 6-5 XSLT/XSLTC 共通の注意事項

項番	注意事項
1	<p>マルチスレッドプログラミングをする場合、TransformerFactory クラス、および TransformerFactoryXSLTC クラスはスレッドセーフではありません。したがって、複数のスレッドが同時に同一の TransformerFactory インスタンス、および TransformerFactoryXSLTC インスタンスにアクセスしてはいけません。スレッド間の競合を避けるため、次のどちらかの方法を使用してください。</p> <ul style="list-style-type: none"> <li>各スレッドに TransformerFactory インスタンス、および TransformerFactoryXSLTC インスタンスを持つ。</li> <li>各スレッドが排他的に TransformerFactory インスタンス、および TransformerFactoryXSLTC インスタンスにアクセスする。</li> </ul>
2	<p>マルチスレッドプログラミングをする場合、Transformer クラスはスレッドセーフではありません。したがって、複数のスレッドが同時に同一の Transformer インスタンスにアクセスしてはいけません。スレッド間の競合を避けるため、次の方法を使用してください。</p> <ul style="list-style-type: none"> <li>各スレッドに Transformer インスタンスを持つ。</li> </ul>
3	<p>マルチスレッドプログラミングをする場合、SAXTransformerFactory クラス、TemplatesHandler オブジェクト、TransformerHandler オブジェクトはスレッドセーフではありません。したがって、各スレッドは排他的にこれらのオブジェクトにアクセスするようにしてください。</p>
4	<p>XSLT トランスフォーマと XSLTC トランスフォーマは同一の機能を提供しますが、エラー発生時に出力するメッセージは異なる場合があります。</p>
5	<p>xsl:attribute 要素の namespace 属性に "http://www.w3.org/1999/XSL/Transform" を指定した場合、出力される属性の名前空間接頭辞が XSLT と XSLTC とで異なります。ただし、名前空間を有効に設定したパーサにとっては、名前空間接頭辞そのものは意味を持ちません。</p>
6	<p>xsl:attribute 要素および xsl:processing-instruction 要素の name 属性に QName でない名前を指定した場合、XSLT では warning イベントが発生しますが、XSLTC では error イベントが発生します。</p>
7	<p>name 属性以外の属性値が異なる xsl:decimal-format 要素が複数存在する場合に発生するエラーイベントは、XSLT では fatalError イベントが発生し、XSLTC では warning イベントが発生します。</p>
8	<p>xsl:element 要素または xsl:attribute 要素の name 属性には、"xml" で始まる名前空間接頭辞を持つ名前を指定しないでください。</p>
9	<p>xsl:element 要素、xsl:attribute 要素、または xsl:processing-instruction 要素の name 属性が、空文字列になる変数参照である場合、XSLT では warning イベントが発生しますが、XSLTC では error イベントが発生します。</p>
10	<p>xsl:element 要素の name 属性がない場合、XSLT では error イベントが発生しますが、XSLTC では warning イベントが発生します。</p>
11	<p>xsl:namespace-prefix 要素を使った場合、出力される属性の名前空間接頭辞が XSLT と XSLTC とで異なります。ただし、名前空間を有効に設定したパーサにとっては、名前空間接頭辞そのものは意味を持ちません。</p>

## 6. Cosminexus XML Processor 使用時の注意事項

項番	注意事項
12	xsl:number 要素の format 属性に指定するフォーマットトークンには, "1", "1" の前に任意の個数の 0 が続いたもの, "A", "a", "i", "I" のどれかを指定してください。
13	xsl:number 要素の level 属性が "any" の場合, count 属性に属性ノードを表すパターンを指定すると, XSLT と XSLTC とで動作が異なります。XSLT では属性ノードがカウントされませんが, XSLTC では属性ノードがカウントされます。
14	xsl:output 要素の doctype-system 属性に空文字列を指定しないでください。同様に, Transformer クラスの setOutputProperty メソッドを用いて doctype-system プロパティに空文字列を指定しないでください。
15	xsl:output 要素の method 属性には "xml", "html", "text" のうちのどれか一つを指定してください。
16	xsl:processing-instruction 要素の name 属性に, 評価結果が空文字列になるような XPath 式を記述すると, XSLT では warning イベントが発生しますが, XSLTC では error イベントが発生します。
17	xsl:value-of 要素の select 属性に変数参照を記述すると, TransformerException が発生します。
18	XSLT と XSLTC では "document(/)" の評価結果が異なります。XSLT ではスタイルシートのルート要素として評価され, XSLTC では空のノード集合として評価されます。
19	document 関数の第一引数に文字列を指定するとき, その文字列に次に該当する XML 文書を指定した場合, XSLTC は XSLT とエラーレベルが異なります。 <ul style="list-style-type: none"> <li>存在しない XML 文書</li> <li>整形形式でない XML 文書</li> </ul>
20	format-number 関数のフォーマットパターン文字列に通貨記号 ( #x00A4 ) を含めてもエラーにならないことがあります。
21	system-property 関数の引数には, "xsl:version", "xsl:vendor", "xsl:vendor-url" のうちのどれか一つを指定してください。
22	埋め込みスタイルシートを使った変換で, 処理命令 "xml:stylesheet" の擬似属性 "href" で指定した ID に対応するスタイルシート要素が複数存在する場合, XSLT と XSLTC では適用されるテンプレートが異なります。XSLT では最後に現れたテンプレートが適用され, XSLTC では最初に現れたテンプレートが適用されます。
23	トランスフォーマ機能で, 出力プロパティに, 未サポートのエンコーディングが指定されている場合, アプリケーションでトランスフォーマにエラーリスナーを登録しても, エラーメッセージ ( KECX02322-W ) は, エラーリスナーに報告されず, 標準出力ストリームへ出力されます。なお, XSLTC の場合, 上記エラーメッセージが標準出力ストリームへ出力されるのに加えて, エラーメッセージ ( KECX07076-W ) がエラーリスナーに報告されます。
24	変換結果を HTML 形式で出力した場合, XSLT と XSLTC とでは要素間の改行出力に差異があります。ただし, HTML 文書では, 要素間の改行は意味を持たないため問題ありません。
25	次の条件がすべて重なるとき, DOMResult オブジェクトの setNode ( Node node ) メソッドを実行すると, IllegalStateException 例外ではなく, IllegalArgumentException 例外が発生します。 <ul style="list-style-type: none"> <li>DOMResult オブジェクトの nextSibling が null ではない。</li> <li>nextSibling が引数 node の子でない。</li> </ul> <p>この場合は, setNode メソッドの例外を処理する catch 節で, IllegalStateException 例外だけでなく IllegalArgumentException 例外もキャッチしてください。</p>

項番	注意事項
26	スタイルシート内で QName として評価される場所 (xsl:element 要素の name 属性, key 関数の第一引数など) に, XML1.0 規格で名前として使用することが許されていない文字 (Unicode 補助文字, 全角数字など) が使用されても, エラーにならないで処理が続行される場合があります。このような場合, トランスフォーマに出力メソッドとして "xml" を指定し, 出力バージョンとして "1.0" を指定しても, 出力される文書は XML1.0 規格に準拠しません。
27	内部エンティティを定義する文字列中に, 477 文字以上の要素名が含まれていると, java.lang.IndexOutOfBoundsException 例外が発生する場合があります。

## 6.6 XSLT に関する注意事項

XSLT に関する注意事項を次に示します。

### 6.6.1 エラーを通知しないケース

表 6-6 XSLT に関する注意事項（エラーを通知しないケース）

項番	注意事項
1	xsl:copy-of 要素の select 属性に、ノード集合以外のオペランドを " " 演算子で結合した式が書かれた場合、XSLT トランスフォーマはエラーを通知しません。
2	xsl:for-each 要素の select 属性が XPath 式として許されない文字（"~", "#" など）を含む場合、XSLT トランスフォーマはエラーを通知しません。
3	xsl:output 要素の cdata-section-elements 属性に QName 規則に合致しない文字列を指定した場合、XSLT トランスフォーマはエラーを通知しません。
4	element-available 関数および function-available 関数に文字列型以外の引数を指定した場合、XSLT トランスフォーマはエラーを通知しません。

### 6.6.2 その他の注意事項

表 6-7 XSLT に関する注意事項（その他）

項番	注意事項
1	xsl:number 要素の grouping-size 属性に指定できない負の値を指定した場合、正の値に変換してグルーピングを行います。
2	xsl:preserve-space 要素および xsl:strip-space 要素の elements 属性には空文字列以外の文字列を指定してください。
3	XSLT を使って変換を行う場合、xsl:strip-space 要素または xsl:preserve-space 要素の elements 属性に "*" を指定すると、名前空間接頭辞付きの要素が空白の削除対象、または保存対象として選択されない場合があります。名前空間接頭辞付きの要素を空白の削除対象または保存対象とする場合は、elements 属性に名前空間接頭辞付きの要素名を指定してください。
4	format-number 関数の第二引数には、長さが 1 文字以上のフォーマットパターンを指定してください。
5	出力プロパティの method 属性に "html" を指定し、cdata-section-elements 属性に要素名を指定した場合、cdata-section-elements 属性が無視されず、指定した要素の内容が CDATA セクションとして出力されます。

## 6.7 XSLTC に関する注意事項

XSLTC に関する注意事項を次に示します。

### 6.7.1 スタイルシートサイズの注意事項

表 6-8 XSLTC に関する注意事項（スタイルシートサイズ）

項番	注意事項
1	XSLTC には、処理可能なスタイルシートサイズの上限があります。18000 バイトを超えるスタイルシートを処理すると、KECX07058-E エラーが発生して処理を中止する場合があります。ただし、スタイルシートサイズの上限は、スタイルシートの内容によって変動します。

### 6.7.2 変換処理性能に関する注意事項

表 6-9 XSLTC に関する注意事項（変換処理性能）

項番	注意事項
1	「3.2.1（1） 同一の XSLT スタイルシートを使って複数の XML 文書を変換する場合」で示したケースに該当するすべての場合について、XSLTC トランスフォーマーの変換処理性能が XSLT トランスフォーマーより高いことを保証するものではありません。XSLTC トランスフォーマーを使用する場合は、実際に使用する XSLT スタイルシートおよび変換対象の XML 文書を使って性能を評価することを推奨します。

### 6.7.3 XSLT 要素の注意事項

表 6-10 XSLTC に関する注意事項（XSLT 要素）

項番	注意事項
1	xsl:apply-imports 要素が処理されません。
2	テンプレートが xsl:with-param によってパラメタとしてノード集合を受け取った場合、そのパラメタを xsl:apply-templates 要素の select 属性で参照すると、それ以降、同じパラメタを参照してもノード集合を取得できなくなります。
3	xsl:attribute 要素の子要素が xsl:copy の場合、出力される属性値が空になります。
4	xsl:attribute 要素の内容に xsl:value-of 要素を指定する場合は、select 属性に指定する XPath 式を絶対口ケーションパスで記述してください。
5	xsl:attribute-set 要素が use-attribute-sets 属性で自分自身を参照（循環参照）する場合、XSLTC ではスタックオーバーフローが発生します。
6	すべての属性の属性値が等しい xsl:decimal-format 要素を複数個定義した場合、XSLTC はエラーを通知します。
7	xsl:element 要素または xsl:attribute 要素の name 属性には属性値テンプレートを指定しないでください。

項番	注意事項
8	<p>次の条件がすべて重なるとき、要素が属する名前空間が不正になります。</p> <ul style="list-style-type: none"> <li>• <code>xsl:element</code> 要素の子要素に <code>xsl:attribute</code> 要素がある。</li> <li>• <code>xsl:element</code> 要素の <code>name</code> 属性が接頭辞付きの名前で、<code>namespace</code> 属性の指定がある。</li> <li>• <code>xsl:attribute</code> 要素の <code>name</code> 属性が接頭辞付きの名前で、<code>namespace</code> 属性の指定がある。</li> <li>• <code>xsl:element</code> と <code>xsl:attribute</code> の <code>name</code> 属性に指定された接頭辞が同じで、<code>namespace</code> 属性の値が異なる。</li> </ul> <p>上記現象が発生するスタイルシートの例を次に示します。</p> <pre>&lt;xsl:template match="/"&gt;   &lt;xsl:element name="child1" namespace="AAA"&gt;     &lt;xsl:element name="A:child2" namespace="XXX"&gt;       &lt;xsl:attribute name="A:attr" namespace="AAA"&gt;1&lt;/xsl:attribute&gt;     &lt;/xsl:element&gt;   &lt;/xsl:element&gt; &lt;/xsl:template&gt;</pre>
9	<p><code>xsl:for-each</code> 要素の <code>select</code> 属性に述部を持つ XPath 式を指定する場合、述部に「ノードを比較対象とする式」を指定しないでください。この場合、スタイルシートを次のように変更してください。</p> <ol style="list-style-type: none"> <li>1. <code>xsl:for-each</code> 要素の <code>select</code> 属性の XPath 式から述部を削除してください。</li> <li>2. <code>xsl:for-each</code> 要素の子に <code>xsl:if</code> 要素を挿入して、<code>test</code> 属性に 1. で削除した述部の内容を指定してください。</li> </ol> <p>スタイルシートの変更例を次に示します。</p> <p>(変更前)</p> <pre>&lt;xsl:template match="/"&gt;   &lt;xsl:for-each select="element[sub='x']"&gt;     &lt;xsl:value-of select="."/&gt;   &lt;/xsl:for-each&gt; &lt;/xsl:template&gt;</pre> <p>(変更後)</p> <pre>&lt;xsl:template match="/"&gt;   &lt;xsl:for-each select="element"&gt;     &lt;xsl:if test="sub='x'"&gt;       &lt;xsl:value-of select="."/&gt;     &lt;/xsl:if&gt;   &lt;/xsl:for-each&gt; &lt;/xsl:template&gt;</pre>
10	<p><code>xsl:for-each</code> 要素の <code>select</code> 属性に「<code>"descendant::node()"</code> または <code>"descendant-or-self::node()"</code> を含む XPath 式」を指定しないでください。要素ノードおよびテキストノードを選択する目的でこのような XPath 式を使用する場合は、スタイルシートを次のように変更してください。</p> <ul style="list-style-type: none"> <li>• <code>"descendant::node()"</code> を <code>"descendant::node() descendant::text()"</code> に置き換えてください。</li> <li>• <code>"descendant-or-self::node()"</code> を <code>"descendant-or-self::node() descendant-or-self::text()"</code> に置き換えてください。</li> </ul>



項番	注意事項
11	<p>xsl:if 要素または xsl:when 要素の test 属性に「self 軸と述部を両方持つ XPath 式」を指定しないでください。この場合、スタイルシートを次のように変更してください。</p> <ol style="list-style-type: none"> <li>1. xsl:if 要素または xsl:when 要素の test 属性の XPath 式から、述部を削除してください。</li> <li>2. xsl:if 要素または xsl:when 要素の子に xsl:if 要素を挿入して、その test 属性に 1. で削除した述部の内容を指定してください。</li> </ol> <p>スタイルシートの変更例を次に示します。 (変更前)</p> <pre>&lt;xsl:template match="/element/sub"&gt;   &lt;xsl:if test="self::sub[attribute::attr='x']"&gt;     &lt;xsl:value-of select="."/&gt;   &lt;/xsl:if &gt; &lt;/xsl:template&gt;</pre> <p>(変更後)</p> <pre>&lt;xsl:template match="/element/sub"&gt;   &lt;xsl:if test="self::sub"&gt;     &lt;xsl:if test="attribute::attr='x'"&gt;       &lt;xsl:value-of select="."/&gt;     &lt;/xsl:if&gt;   &lt;/xsl:if &gt; &lt;/xsl:template&gt;</pre>
12	<p>xsl:import 要素の href 属性の値が空文字列または空白だけの場合、error イベントが発生し、エラーメッセージとして "null" が出力されます。</p>
13	<p>xsl:number 要素の count 属性に "node()" を含むパターンを指定しないでください。要素ノードを番号付けする目的で "node()" を使用する場合は、代わりに "*" または "要素名" を使用してください。</p>
14	<p>xsl:number 要素または xsl:decimal-format 要素の grouping-separator 属性に複数の空白文字列や 1 文字以上の文字列、または全角文字を指定した場合、エラーにはならず、指定した文字列で区切られた値が出力されます。</p>
15	<p>xsl:number 要素の count 属性に "/" を指定した場合、XSLT では "1" が出力されますが、XSLTC では java.lang.VerifyError が発生します。</p>
16	<p>xsl:number 要素の from 属性に "/" を指定した場合、XSLT では "1" が出力されますが、XSLTC では java.lang.VerifyError が発生します。</p>
17	<p>xsl:output 要素の doctype-public 属性には空文字列以外の文字列を指定してください。同様に、Transformer クラスの setOutputProperty メソッドを用いて doctype-public プロパティを設定する場合も、空文字列以外の文字列を指定してください。</p>
18	<p>xsl:processing-instruction 要素の name 属性に、処理命令のターゲット名として許されない "XML" を指定した場合、"&lt;?XML?&gt;" という不正な処理命令が出力されます。</p>
19	<p>xsl:sort の data-type 属性に不正な値 "NUMBER" (大文字) を指定した場合、デフォルト動作 ("text" が指定された場合と同じ順序でソートする) が行われません。</p>
20	<p>xsl:sort の order 属性に不正な値 "DESCENDING" (大文字) を指定した場合、デフォルト動作 ("ascending" が指定された場合と同じ順序でソートする) が行われません。</p>

項番	注意事項
21	<p>xsl:template 要素の match 属性に「"node()"」を含み、かつ、その後ろにロケーションパスが続く XPath 式」を指定しないでください。要素ノードを選択する目的で "node()" を使用する場合は、代わりに "*" を使用してください。スタイルシートの変更例を次に示します。</p> <p>(変更前)</p> <pre>&lt;xsl:template match="/node()/sub"&gt; &lt;result&gt;&lt;xsl:value-of select="."/&gt;&lt;/result&gt; &lt;/xsl:template&gt;</pre> <p>(変更後)</p> <pre>&lt;xsl:template match="*/sub"&gt; &lt;result&gt;&lt;xsl:value-of select="."/&gt;&lt;/result&gt; &lt;/xsl:template&gt;</pre>
22	xsl:template 要素の子要素としてトップレベル要素を書いた場合、無視されずに有効な子要素として処理されます。
23	xsl:value-of 要素の select 属性に、負のゼロを引数とする string 関数を記述した場合、XSLT では "0" が出力されますが、XSLTC では "-0" が出力されます。
24	xsl:variable 要素の子要素に xsl:call-template 要素を指定し、かつ、xsl:call-template で呼び出されたテンプレート内に名前空間接頭辞を持つリテラル要素がある場合、xsl:copy-of 要素でその変数が示す要素をコピーすると、要素の名前空間宣言が正しくコピーされません。
25	リテラル結果要素に xsl:extension-element-prefixes 属性を指定して拡張要素とした場合、拡張要素の子要素として書かれた xsl:fallback 要素が無視されます。代わりに xsl:stylesheet 要素に extension-element-prefixes 属性を指定してください。
26	スタイルシート内で仕様上、許されない場所に子要素を書いた場合（例：xsl:stylesheet 要素の子要素としてリテラル要素を書いた場合、xsl:namespace-alias 要素の子要素として xsl:text 要素が現れた場合など）、エラーにはならないで、不正な子要素が無視されます。

## 6.7.4 XPath 式の注意事項

表 6-11 XSLTC に関する注意事項 (XPath 式)

項番	注意事項
1	document 関数に XPath 式を続けて指定する場合、"/" は使用しないで、代わりに "/" descendant-or-self::node()" を使用してください。
2	最上位要素の中で current 関数を使用しないでください。
3	XPath 式の中でノードテスト "node()" を使用した場合、ノードを正しく取得できません。
4	XPath 式に、"*" を含むノードテストと論理演算子を組み合わせた XPath 式を書いた場合、XPath 式の構文解析中にエラーが発生します。
5	変数およびパラメタの値が、結果ツリーフラグメント（select 属性ではなく、要素内容として書かれたテンプレートによって定義される値）の場合、そのテンプレートの中に変数参照があると、NullPointerException が発生する場合があります。

## 6.7.5 エラーを通知しないケース

要素や属性の指定内容が不当な場合や、関数の引数が不正な場合など、XSLTC がエラーを通知しないケースがあります。XSLTC がエラーを通知しないケースを表 6-12 に示します。

表 6-12 XSLTC に関する注意事項（エラーを通知しないケース）

項番	注意事項
1	xsl:apply-templates 要素の子要素として xsl:sort 要素と xsl:with-param 要素以外のスタイルシート要素を指定した場合。
2	xsl:apply-templates 要素の select 属性に空文字列を指定した場合。
3	xsl:comment 要素の要素内容にテキスト以外を指定した場合。
4	xsl:decimal-format 要素の次に挙げた属性に 2 文字以上の値を指定した場合。 <ul style="list-style-type: none"> <li>decimal-separator</li> <li>grouping-separator</li> <li>minus-sign</li> <li>percent</li> <li>per-mille</li> <li>zero-digit</li> <li>pattern-separator</li> </ul>
5	xsl:element 要素または xsl:attribute 要素の name 属性に接頭辞で修飾した要素名を指定し、かつ、namespace 属性に空文字列を指定した場合。
6	xsl:for-each 要素の先頭以外の子要素として xsl:sort を指定した場合。
7	xsl:import 要素に要素内容を指定した場合。
8	xsl:import 要素がトップレベル要素の先頭要素でない場合や、xsl:import 要素や xsl:include 要素がトップレベル要素でない場合。
9	xsl:key 要素の use 属性に key 関数を含む XPath 式を指定した場合。
10	xsl:message 要素を xsl:stylesheet 要素の子要素として指定した場合。
11	xsl:message 要素の terminate 属性に "yes", "no" 以外の値を指定した場合。
12	xsl:namespace-alias 要素の stylesheet-prefix 属性や result-prefix 属性に指定した接頭辞に対応する名前空間宣言がない場合。
13	xsl:namespace-alias 要素の stylesheet-prefix 属性や result-prefix 属性の指定がない場合。
14	xsl:namespace-alias 要素に要素内容を指定した場合。
15	xsl:number 要素の grouping-size 属性に Number 規則に合致しない文字列を指定した場合。
16	xsl:number 要素の lang 属性および level 属性に不当な値を指定した場合。
17	xsl:number 要素の letter-value 属性に不当な値を指定した場合。
18	xsl:number 要素の grouping-separator 属性に、値が空文字となる変数参照を指定した場合。
19	xsl:output 要素の version 属性に不当な値を指定した場合。
20	xsl:output 要素の indent 属性に不当な値を指定した場合。
21	xsl:output 要素の standalone 属性に不当な値を指定した場合。

## 6. Cosminexus XML Processor 使用時の注意事項

項番	注意事項
22	xsl:output 要素の omit-xml-declaration 属性に不当な値を指定した場合。
23	xsl:param 要素の select 属性に空文字列を指定した場合。
24	xsl:sort 要素の data-type 属性, order 属性, または case-order 属性に不正な値を指定した場合。
25	xsl:sort 要素の data-type 属性, order 属性, または case-order 属性が空文字列になる変数参照である場合。
26	xsl:sort 要素の data-type 属性, order 属性, または case-order 属性に大文字の値を指定した場合, 小文字で指定されたものとして扱われます。
27	xsl:strip-space 要素, xsl:preserve-space 要素, xsl:apply-imports 要素など, 仕様上は要素内容を持ってない要素に要素内容を指定した場合。
28	xsl:stylesheet 要素の version 属性に不当な値を指定した場合。
29	xsl:stylesheet 要素の exclude-result-prefixes 属性に指定した接頭辞に対応する名前空間 URI が, この属性を持つ要素で有効でない場合。
30	xsl:stylesheet 要素の中に xsl:stylesheet 要素を指定した場合。
31	xsl:template 要素の match 属性および mode 属性に空文字列を指定した場合。
32	xsl:template 要素に name 属性の指定がなく, match 属性の指定もない場合。
33	xsl:template 要素または xsl:apply-template 要素の mode 属性に空文字列を指定した場合。
34	xsl:template 要素の priority 属性に空文字列を指定した場合。
35	最初の xsl:template 要素より前にリテラル結果要素が現れた場合。
36	xsl:template 要素に mode 属性があり, match 属性がない場合。
37	複数の xsl:with-param 要素が同じ値の name 属性を持っている場合。
38	xsl:text 要素の disable-output-escaping 属性に "yes", "no" 以外の値を指定した場合。
39	xsl:value-of 要素の disable-output-escaping 属性に "yes" または "no" 以外の値を指定した場合。
40	xsl:variable 要素の select 属性に空文字列を指定した場合。
41	xsl:variable 要素または xsl:param 要素に, select 属性とテキスト内容の両方が存在する場合。
42	xsl:with-param 要素の select 属性に空文字列を指定した場合。
43	次の関数呼び出しの引数の数が不正な場合。 <ul style="list-style-type: none"> <li>• number 関数</li> <li>• boolean 関数</li> <li>• lang 関数</li> <li>• concat 関数</li> </ul>
44	document 関数の第二引数が空文字列の場合。
45	format-number 関数の第三引数に空文字列を指定した場合。
46	key 関数の第一引数に存在しないキー名を指定した場合。
47	unparsed-entity-uri 関数の引数の個数が不正な場合。

## 6.7.6 その他の注意事項

表 6-13 XSLTC に関する注意事項（その他）

項番	注意事項
1	XSLTC を使った場合、デフォルトのエラーリスナーはエラー時に例外をスローします。
2	ErrorListener の warning メソッドや error メソッドの実装が例外をスローしない場合、スタイルシート解析時に warning や error が発生するとトランスレットの作成ができないため、変換ができません。
3	エラーメッセージ中のエラー ID の前に、エラー原因となったファイルのファイル名と行番号が出力される場合があります。
4	SAXTransformerFactory クラスの newTemplatesHandler メソッドで作成した TemplatesHandler オブジェクトを XMLReader クラスの setContentHandler メソッドでパーサに登録して変換処理を行う場合、エラーのあるスタイルシートを処理してもエラーが出力されません。
5	SAXTransformerFactory を使って変換を行う場合、エラーリスナーの設定にかかわらず、スタイルシートのエラーがエラーリスナーに通知されません。
6	XSLTC を使って変換を行う場合、一つの xsl:namespace-alias 要素の stylesheet-prefix 属性と result-prefix 属性の両方に "#default" を指定すると、NullPointerException が発生します。
7	XSLTC を使って変換を行う場合、インポート優先順位が異なる二つのテンプレートが name 属性と match 属性を持ち、name 属性の値が同じであるとき、インポート優先順位が低い方のテンプレートが無視されます。
8	XSLTC を使って変換を行う場合、スタイルシートと JAXP インタフェースの両方で出力プロパティの cdata-section-elements 属性を設定すると、スタイルシート内で定義された値が JAXP インタフェースで設定された値によって上書きされます。
9	XSLTC を使って変換を行う場合、DocumentFragment から作った DOMSource を返すように実装した URIResolver を使用すると、正しい変換処理が行われません。
10	出力メソッドに HTML 出力を指定した場合、xsl:element 要素の name 属性に "{name()}" または "{name(.)}" を指定すると、NullPointerException が発生します。

## 6.8 javax.xml.datatype パッケージに関する注意事項

javax.xml.datatype パッケージに関する注意事項を表 6-14 に示します。

表 6-14 javax.xml.datatype パッケージに関する注意事項

項番	注意事項
1	マルチスレッドプログラミングをする場合、javax.xml.datatype パッケージで規定されたオブジェクトはスレッドセーフではありません。したがって、各スレッドは排他的にこれらのオブジェクトにアクセスするようにしてください。
2	DatatypeFactory クラスの各メソッドで IllegalArgumentException が発生したとき、getMessage() メソッドでは、各メソッドに指定したパラメタが戻ります。
3	DatatypeFactory クラスの newXMLGregorianCalendar メソッド、newXMLGregorianCalendarDate メソッド、newXMLGregorianCalendarTime メソッドで、月を表すフィールドに 4、6、9、11 のどれかを設定したとき、日のフィールドに 31 を設定してもエラーが発生しません。
4	DatatypeFactory クラスの newXMLGregorianCalendar メソッドおよび newXMLGregorianCalendarTime メソッドの引数 millisecond には、0 から 999 までの値を指定してください。
5	月を表すフィールドが 1 の Duration オブジェクトと、日を表すフィールドが 28 の Duration オブジェクトを比較したとき、結果は次のようになります。 <ul style="list-style-type: none"> <li>compare メソッドで比較したときは、DatatypeConstants.EQUAL が戻ります。</li> <li>equals メソッドで比較したときは、true が戻ります。</li> </ul>
6	XMLGregorianCalendar クラスの newDuration(long durationInMilliSeconds) メソッド、newDurationDayTime(long durationInMilliSeconds) メソッド、または newDurationYearMonth(long durationInMilliSeconds) メソッドで作成した Duration オブジェクトは、DatatypeConstants.DURATION_DAYTIME 型になります。したがって、この Duration オブジェクトの月を表すフィールド、日を表すフィールドの値は 0 になります。
7	XMLGregorianCalendar クラスの setHour メソッド、および setTime メソッドの引数 hour には、0 から 23 までの値を指定してください。また、DatatypeFactory クラスの newXMLGregorianCalendar メソッド、newXMLGregorianCalendarTime メソッドの引数の「時間を表すフィールド」には、0 から 23 までの値を指定してください。
8	JAXP1.3 仕様書では、XMLGregorianCalendar クラスの isValid メソッドは「Validate instance by getXMLSchemaType() constraints.」と定義されています。Cosminexus XML Processor では、次の制約をすべて満たしているときに true を返します。 <ul style="list-style-type: none"> <li>月を表すフィールドが 2 のとき、日を表すフィールドが 28 以内（うるう年以外の場合）、または 29 以内（うるう年の場合）である。</li> <li>年を表すフィールドが 0 ではない。</li> </ul>
9	XMLGregorianCalendar クラスの equals メソッドの引数が null のとき、NullPointerException 例外は発生せずに false が戻ります。
10	DatatypeConstants.GMONTH 型の XMLGregorianCalendar オブジェクトに toString、toXMLFormat メソッドを適用して得られる字句表現は、W3C XML Schema 1.0 Part 2 (second edition) 仕様書のセクション 3.2.14 で規定された文字列「--MM」ではなく、second edition より前の仕様書で規定された文字列「--MM--」になります。

## 6.9 javax.xml.validation パッケージに関する注意事項

javax.xml.validation パッケージに関する注意事項を表 6-15 に示します。

表 6-15 javax.xml.validation パッケージに関する注意事項

項番	注意事項
1	マルチスレッドプログラミングをする場合、SchemaFactory クラス、Validator クラス、ValidatorHandler クラス、TypeInfoProvider クラスはスレッドセーフではありません。したがって、各スレッドは排他的にこれらのオブジェクトにアクセスするようにしてください。
2	SchemaFactory クラスの newInstance(String schemaLanguage) メソッドの引数に指定可能な URI は "http://www.w3.org/2001/XMLSchema" だけです。
3	SchemaFactory クラスの isSchemaLanguageSupported メソッドには、"http://www.w3.org/2001/XMLSchema" を指定してください。その他のスキーマ言語を指定した場合は、false が戻ります。
4	SchemaFactory クラスの newSchema(Source[] schemas) メソッドの引数 schemas の配列要素には、それぞれ異なる targetNamespace を持つスキーマ文書を指定してください。同一の targetNamespace を持つスキーマ文書、または targetNamespace を持たないスキーマ文書を複数個指定すると、インデックスの小さい配列要素のスキーマ文書が優先されます。
5	次の条件で SchemaFactory クラスの newSchema(Source[] schemas) メソッドを実行すると、例外が発生しない場合があります。 <ul style="list-style-type: none"> <li>引数 schemas の配列要素が、正しいスキーマ文書と不正なスキーマ文書の両方を含んでいる。</li> </ul> この問題は、次のようにして回避できます。 <ol style="list-style-type: none"> <li>SchemaFactory オブジェクトのエラーハンドラで、error、fatalError メソッドに渡される例外オブジェクトを保存しておく。</li> <li>newSchema メソッドの呼び出し後に、1. で例外オブジェクトが保存されたかどうかチェックして、保存されていればその例外オブジェクトをスローする。</li> </ol>
6	Validator クラスの validate(Source)、validate(Source, Result) の引数には、表 6-16 に示す組み合わせで指定してください。
7	Validator クラスの validate メソッドによって、ID 型の属性の定義を含むスキーマ文書を用いて XML 文書を検証する場合は、validate(DOMSource, DOMResult) ではなく、validate(SAXSource)、validate(DOMSource)、validate(SAXSource, SAXResult) のどれかのメソッドを使用してください。
8	次の条件がすべて重なるとき、Validator クラスの validate(Source source, Result result) メソッドを実行すると、DOMException 例外が発生する場合があります。 <ol style="list-style-type: none"> <li>引数 result が DOMResult オブジェクトである。</li> <li>DOMResult オブジェクトに設定された「結果ツリーを含む DOM ノード」が子ノードを持つ Document オブジェクトである。</li> </ol> したがって、2. の Document オブジェクトには子ノードを持たせないようにしてください。
9	スキーマ文書の中に xsd:ENTITY 型、および xsd:ENTITIES 型が含まれるとき、ValidatorHandler クラスを用いたスキーマ検証はサポートしていません。このような場合は、例えば Validator クラスなどのほかの方法でスキーマ検証するようにしてください。

項番	注意事項
10	スキーマ文書で定義されていない要素および属性から取得した TypeInfo オブジェクトは、妥当性検証失敗時には、xsd:anyType 型の TypeInfo オブジェクトと同じになります。
11	<p>Validator クラスの validate(Source) および validate(Source, Result) の引数 Source に DOMSource オブジェクトを指定する場合、DOMSource オブジェクトが保持する Node オブジェクトは、次に示すすべての条件に従う必要があります。</p> <ol style="list-style-type: none"> <li>名前空間を有効に設定したパーサによって生成されている。 例えば、setNamespaceAware(true) を適用した DocumentBuilderFactory で生成した DocumentBuilder など。</li> <li>妥当性検証をしないパーサによって生成されている。 例えば、setValidating (false) を適用した DocumentBuilderFactory で生成した DocumentBuilder など。</li> </ol>
12	<p>Validator クラスの validate(Source), validate(Source, Result) の引数 Source に SAXSource オブジェクトを指定する場合に、その SAXSource オブジェクトが XMLReader オブジェクトを使用するとき、XMLReader オブジェクトは次に示すすべての条件に従う必要があります。</p> <ol style="list-style-type: none"> <li>名前空間を有効に設定した XMLReader オブジェクトである。 例えば、setNamespaceAware(true) を適用した SAXParserFactory で SAXParser を生成し、その SAXParser から取り出した XMLReader オブジェクトなど。</li> <li>妥当性検証をしない XMLReader オブジェクトである。 例えば、setValidating (false) を適用した SAXParserFactory で SAXParser を生成し、その SAXParser から取り出した XMLReader オブジェクトなど。</li> </ol>

表 6-16 Validator クラスの validate(Source), validate(Source, Result) の引数の組み合わせ

引数 Result	引数 Source			
	SAXSource	DOMSource	StreamSource	null
SAXResult		×	×	×
DOMResult	×		×	×
StreamResult	×	×	×	×
null			×	×



## 6.10 javax.xml.xpath パッケージに関する注意事項

javax.xml.xpath パッケージに関する注意事項を表 6-17 に示します。

表 6-17 javax.xml.xpath パッケージに関する注意事項

項番	注意事項
1	<p>マルチスレッドプログラミングをする場合、XPathFactory クラスはスレッドセーフではありません。したがって、複数のスレッドが同時に同一の XPathFactory インスタンスにアクセスしてはいけません。スレッド間の競合を避けるため、次のどちらかの方法を使用してください。</p> <ul style="list-style-type: none"> <li>各スレッドに一つの XPathFactory インスタンスを持つ。</li> <li>各スレッドが排他的に XPathFactory インスタンスにアクセスする。</li> </ul>
2	<p>マルチスレッドプログラミングをする場合、XPathFactory クラスの newInstance メソッドで生成される XPath インスタンス、および XPath インタフェースの compile メソッドで生成される XPathExpression インスタンスはスレッドセーフではありません。したがって、複数のスレッドが同時に同一の XPath インスタンス、および XPathExpression インスタンスにアクセスしてはいけません。スレッド間の競合を避けるため、次の方法を使用してください。</p> <ul style="list-style-type: none"> <li>各スレッドに一つの XPath インスタンス、および XPathExpression インスタンスを持つ。</li> </ul>
3	<p>XPathFactory クラスの newInstance(String uri) メソッドの引数に指定可能な URI は "http://java.sun.com/jaxp/xpath/dom" だけです。</p>
4	<p>次のメソッドを用いて、コンテキストを参照するような XPath 式を評価する場合は、引数 item に null ではないコンテキストを指定してください。</p> <ul style="list-style-type: none"> <li>XPath インタフェースの evaluate(String expression, Object item, QName returnType) メソッド</li> <li>XPath インタフェースの evaluate(String expression, Object item) メソッド</li> <li>XPathExpression インタフェースの evaluate(Object item, QName returnType) メソッド</li> <li>XPathExpression インタフェースの evaluate(Object item) メソッド</li> </ul>
5	<p>4 で示したメソッドの引数 item には、Document, DocumentFragment, Element, Text, Attr, ProcessingInstruction, Comment のどれかのオブジェクトを指定してください。</p>
6	<p>XPathFunction インタフェースを実装するクラスの evaluate メソッドの戻り値の型、および、XPathVariableResolver インタフェースを実装するクラスの resolveVariable メソッドの戻り値の型は、java.lang.String, java.lang.Boolean, java.lang.Number, org.w3c.Node, org.w3c.NodeList のどれかにしてください。</p>
7	<p>javax.xml.xpath パッケージのメソッドで例外が発生した場合、発生した例外オブジェクトに getMessage メソッドを適用した戻り値が null になり詳細メッセージを取得できないことがあります。この場合、例外オブジェクトに getCause メソッドを適用して「ラップされた例外オブジェクト」を取得し、これに getMessage メソッドを適用することで、詳細なメッセージを取得できる場合があります。</p>
8	<p>and, or, mod, div 演算子の前後には、空白を入れてください。 (例) "1000 div 10"</p>
9	<p>次のすべての条件が重なる XPath 式はサポートしません。</p> <ol style="list-style-type: none"> <li>演算子のオペランドのそれぞれにノードセットを返す XPath 関数を指定して、ノードセットの和集合を作成する。</li> <li>1. で作成したノードセットの和集合に述語を適用する。</li> </ol>

項番	注意事項
10	<p>次のすべての条件が重なる XPath 式はサポートしません。</p> <ol style="list-style-type: none"> <li>1. local-name, namespace-uri, name のどれかの関数を使用する。</li> <li>2. 1. で示した関数の引数に次の式を指定する。  <code>self::node()/descendant::prefix1:* ( )</code>  この場合, 1. で示した関数の引数を次の式に変更することで回避できます。  <code>self::node()/self::node()/descendant::prefix1:* ( )</code>  注 : prefix1 は, 任意の名前空間接頭辞を表します。</li> </ol>
11	属性ノード, 名前空間ノードに following-sibling 軸を適用した XPath 式はサポートしません。
12	演算子のオペランドがロケーションセット以外の XPath 式はサポートしません。
13	名前空間接頭辞と名前空間 URI の組が等しい名前空間ノードが複数存在するとき, それらの名前空間ノードに軸を適用する XPath 式はサポートしません。

## 6.11 org.w3c.dom パッケージに関する注意事項

org.w3c.dom パッケージに関する注意事項を表 6-18 に示します。

表 6-18 org.w3c.dom パッケージに関する注意事項

項番	注意事項
1	DOMConfiguration オブジェクトに設定可能なパラメータを表 6-19 に示します。設定可能なパラメータは、getParameterNames メソッドの戻り値、および canSetParameter メソッドの戻り値とは一致しません。また、設定不可の値を設定しても、例外が発生しない場合があります。設定不可の値を設定した場合の動作は保証しません。
2	DOMConfiguration オブジェクトのパラメータ名は、大文字と小文字が区別されません。
3	DOM Level 3 では、従来の DOM Level 2 のインタフェースに対して新しいメソッドが追加されています。例えば、Attr インタフェースには isId メソッドが追加されています。通常のアプリケーションプログラムがこれらのインタフェースを implements することはありませんが、もし implements している場合は、DOM Level 3 で追加されたメソッドの実装をアプリケーションに追加する必要があります。
4	XMLSchema で属性のデフォルト値を定義して、その属性に対応する Attr ノードを削除したとき、getSpecified メソッドの戻り値が false である属性ノードは生成されません。
5	Document オブジェクトの normalizeDocument メソッドはサポートしていません。
6	Document オブジェクトの adoptNode メソッドはサポートしていません。代わりに、Document オブジェクトの importNode メソッドと Node オブジェクトの removeChild メソッドを組み合わせ使用してください。
7	Document オブジェクトの getInputEncoding メソッドおよび Entity オブジェクトの getInputEncoding メソッドはサポートしていません。
8	次の条件がすべて重なるとき、Document オブジェクトの renameNode(Node n, String namespaceURI, String qualifiedName) メソッドを実行しても DOMException 例外は発生しません。 <ul style="list-style-type: none"> <li>• 引数 n が名前空間をサポートしない Attr オブジェクトである。</li> <li>• 引数 namespaceURI が null である。</li> <li>• 引数 qualifiedName が不正な修飾名である。</li> </ul> また、引数 n に Element または Attr オブジェクトを指定し、かつ、引数 qualifiedName に null を指定した場合、DOMException 例外ではなく NullPointerException 例外が発生します。
9	Attr オブジェクト、Text オブジェクト、EntityReference オブジェクト、CDATASection オブジェクト、DocumentType オブジェクト、DocumentFragment オブジェクトの getBaseURI メソッドでは URI の取得ができません。
10	Text オブジェクトの replaceWholeText(String) メソッドは EntityReference ノードを含まない DOM ツリーに適用してください。
11	DOMLocator オブジェクトの getByteOffset の戻り値は常に -1 です。
12	DOMErrorHandler オブジェクトの handleError メソッドの引数 DOMError に getRelatedData メソッドを適用したときの戻り値のオブジェクトは、Node オブジェクトとは限りません。エラーメッセージを表す String オブジェクトが戻り値となる場合があります。

表 6-19 DOMConfiguration オブジェクトに設定可能なパラメタ

パラメタの名前	設定可能な値
canonical-form	false
cdata-sections	true , false
check-character-normalization	false
comments	true , false
datatype-normalization	false
element-content-whitespace	true
entities	true , false
error-handler	DOMErrorHandler オブジェクト
infoset	true , false
namespaces	true
namespace-declarations	true , false
normalize-characters	false
schema-location	設定できません
schema-type	設定できません
split-cdata-sections	true , false
validate	false
validate-if-schema	false
well-formed	true
resource-resolver	LSResourceResolver オブジェクト

## 6.12 org.w3c.dom.bootstrap パッケージに関する注意事項

---

org.w3c.dom.bootstrap パッケージに関する注意事項を表 6-20 に示します。

表 6-20 org.w3c.dom.bootstrap パッケージに関する注意事項

項番	注意事項
1	DOMImplementationRegistry クラスの getDOMImplementation メソッド、および getDOMImplementationList メソッドの引数とする文字列には空文字列以外を指定してください。

## 6.13 org.w3c.dom.ls パッケージに関する注意事項

org.w3c.dom.ls パッケージに関する注意事項を表 6-21 に示します。

表 6-21 org.w3c.dom.ls パッケージに関する注意事項

項番	注意事項
1	DOMImplementationLS インタフェースの createLSParser(short mode, String schemaType) メソッドの引数 mode には DOMImplementationLS. MODE_SYNCHRONOUS だけを指定できます。また、引数 schemaType には null, "http://www.w3.org/2001/XMLSchema", "http://www.w3.org/TR/REC-xml" のどれかを指定できます。
2	LSParser オブジェクトに getDomConfig メソッドを適用して得られる DOMConfiguration オブジェクトに設定可能なパラメタを表 6-22 に示します。これに加えて、表 6-19 に示したパラメタも設定可能です。設定可能なパラメタは、getParameterNames メソッドの戻り値、および canSetParameter メソッドの戻り値とは一致しません。また、設定不可の値を設定しても、例外が発生しない場合があります。設定不可の値を設定した場合の動作は保証しません。
3	LSParser オブジェクトの charset-overrides-xml-encoding パラメタに値を設定しても、無視されます。
4	LSParser インタフェースの parseWithContext メソッドはサポートしていません。
5	LSParser オブジェクトの setNewLine メソッドはサポートしていません。
6	LSParser オブジェクトでは、XML1.1 文書を解析できません。
7	LSParser オブジェクトの startElement メソッドは、ルート要素については呼び出されません。
8	LSParser オブジェクトでの解析時、および LSSerializer オブジェクトでの直列化時に LSEException 例外が発生したとき、getMessage() メソッドの戻り値が null になる場合があります。この場合は、LSParserFilter や LSSerializerFilter のユーザ実装クラスで例外が発生している可能性があります。
9	LSSerializer オブジェクトに getDomConfig メソッドを適用して得られる DOMConfiguration オブジェクトに設定可能なパラメタを表 6-23 に示します。これに加えて、表 6-19 に示したパラメタも設定可能です。設定可能なパラメタは、getParameterNames メソッドの戻り値、および canSetParameter メソッドの戻り値とは一致しません。また、設定不可の値を設定しても、例外が発生しない場合があります。設定不可の値を設定した場合の動作は保証しません。
10	LSSerialize オブジェクトの write, writeToString, および writeToURI メソッドの引数 nodeArg に、Document, DocumentFragment, Element 以外のノードを指定した場合、それらのノードは直列化されません。
11	LSSerialize オブジェクトの write, writeToString, および writeToURI メソッドは、名前空間が無効なノードを直列化できません。
12	LSSerializer インタフェースの getNewLine メソッドで得られる行末シーケンス文字のデフォルト値は、"¥n" です。
13	ProcessingInstruction オブジェクトが持つ処理命令の内容に文字列「?>」が含まれるとき、その ProcessingInstruction オブジェクトを直列化してもエラーは通知されません。

表 6-22 DOMConfiguration オブジェクトに設定可能なパラメタ (LSParser オブジェクト)

パラメタの名前	設定可能な値
charset-overrides-xml-encoding	true , false
disallow-doctype	false
ignore-unknown-character-denormalizations	true
supported-media-types-only	false

表 6-23 DOMConfiguration オブジェクトに設定可能なパラメタ (LSSerializer オブジェクト)

パラメタの名前	設定可能な値
discard-default-content	true , false
format-pretty-print	false
ignore-unknown-character-denormalizations	true
xml-declaration	true , false

# 6.14 org.xml.sax.ext パッケージに関する注意事項

org.xml.sax.ext パッケージに関する注意事項を表 6-24 に示します。

表 6-24 org.xml.sax.ext パッケージに関する注意事項

項番	注意事項
1	<p>Attributes2Impl クラスの addAttribute メソッドはサポートしていません。 代わりに、AttributesImpl クラスの addAttribute メソッドを使用し、アプリケーションを次のように変更してください。</p> <p>(変更前)</p> <pre>Attributes2Impl myAtt = new Attributes2Impl(); String type = "ID"; myAtt.addAttribute("http://hitachi-xmlprocessor.com/", "attr",     "ht:attr", type, "value");     :</pre> <p>(変更後)</p> <pre>AttributesImpl myAtt = new AttributesImpl(); String type = "ID"; myAtt.addAttribute("http://hitachi-xmlprocessor.com/", "attr",     "ht:attr", type, "value"); Attributes2Impl myAtt2 = new Attributes2Impl(myAtt); int position = myAtt2.getLength() -1; myAtt2.setSpecified(position, true); myAtt2.setDeclared (position, !"CDATA".equals(type));     :</pre>
2	<p>Locator2Impl クラスの getXMLVersion メソッド, getEncoding メソッドはサポートしていません。</p>



## 6.15 XInclude に関する注意事項

XInclude に関する注意事項を表 6-25 に示します。

表 6-25 XInclude に関する注意事項

項番	注意事項
1	xi:include 要素の xpointer 属性に指定可能なスキーマは、element() スキーマと簡略ポインタだけです。
2	xi:include 要素の xpointer 属性に簡略ポインタを指定する場合、その簡略ポインタは「DTD で ID 型と定義した属性」の属性値である必要があります。「XML Schema で xsd:ID 型と定義した属性」の属性値は指定できません。

## 6.16 実装依存仕様に関する注意事項

XML Schema の実装に依存した仕様を表 6-26 ~ 表 6-28 に示します。

表 6-26 データ型に関する実装依存仕様

データ型			最大値	最大桁数
duration 型 (形式：PnYnMnDTnHnMnS)	nY(年)			-
	nM(月)			
	nD(日)			
	nH(時)			
	nM(分)			
	nS(秒)	1 秒以上		
		1 秒未満		
date 型 (形式：CCYY-MM-DD)	CCYY ("9999" を超える場合，桁数の追加ができる)			
gYearMonth 型 (形式：CCYY-MM)				
gYear 型 (形式：CCYY)				
dateTime 型 (形式：CCYY-MM-DDThh:mm:ss)	1 秒未満 (ss に続く，小数点以下に 1 秒未満の値を指定できる)			
time 型 (形式：hh:mm:ss)				
base64Binary 型				
hexBinary 型				
decimal 型				
integer 型				
nonPositiveInteger 型				
nonNegativeInteger 型				
negativeInteger 型				
positiveInteger 型				

(凡例)

- : 指定できる値は, 最大 2147483647 です。
- : 指定する値の最大値の規定はありません。
- : 指定できる最大桁数は, メモリに依存します。
- : 該当しません。

表 6-27 制約ファセットに関する実装依存仕様

制約ファセット	最大値	最大桁数
length		-
minLength		
maxLength		
totalDigits		
fractionDigits		
maxInclusive		
maxExclusive		
minInclusive		
minExclusive		

( 凡例 )

- : 指定できる値は , 最大 2147483647 です。
- : 指定する値の最大値の規定はありません。
- : 指定できる最大桁数は , メモリに依存します。
- : 該当しません。

表 6-28 出現回数の指定に関する実装依存仕様

出現回数の指定	最大値	最大桁数
minOccurs		-
maxOccurs		

( 凡例 )

- : 指定できる値は , 最大 2147483647 です。
- : 該当しません。

## 6.17 高速パース機能に関する注意事項

高速パース機能に関する注意事項を次に示します。

### 6.17.1 解析結果オブジェクトの設定に関する注意事項

表 6-29 高速パース機能に関する注意事項（解析結果オブジェクトの設定）

項番	注意事項
1	事前解析時の名前空間の有効・無効は、解析時の名前空間の有効・無効に合わせる必要があります。これらが異なる場合、解析結果が不正になるおそれがあります。
2	エンティティ参照を解決するとき、事前解析時のエンティティリゾルバで解決した結果と、解析時のエンティティリゾルバで解決した結果は一致している必要があります。これらの結果が異なる場合、高速化の効果が得られないおそれがあります。

### 6.17.2 解析結果オブジェクトを利用する XML パーサに関する注意事項

表 6-30 高速パース機能に関する注意事項（解析結果オブジェクトを利用する XML パーサ）

項番	注意事項
1	解析結果オブジェクトを利用するパーサの次のフィーチャーを <code>false</code> に設定した場合、高速化の効果が得られない場合があります。 1. <code>http://xml.org/sax/features/external-general-entities</code> 2. <code>http://xml.org/sax/features/external-parameter-entities</code>

### 6.17.3 XML1.1 に関する注意事項

表 6-31 高速パース機能に関する注意事項（XML1.1）

項番	注意事項
1	高速パース機能では、XML1.1 はサポートしていません。このため、XML1.1 文書を事前解析するとエラーが発生します。また、XML1.1 文書を高速パース機能の対象に設定しても、高速パース機能は適用されません。

### 6.17.4 XInclude に関する注意事項（高速パース機能）

表 6-32 高速パース機能に関する注意事項（XInclude）

項番	注意事項
1	XInclude で取り込むファイルの解析に対しては、高速パース機能は適用されません。

### 6.17.5 高速化定義ファイルに関する注意事項

表 6-33 高速パース機能に関する注意事項（高速化定義ファイル）

項番	注意事項
1	高速化定義ファイルのプロパティキー（高速パース機能の対象となる XML 文書名）には、高速パース機能の対象となる parse メソッドで処理する XML 文書の URI を指定してください。高速パース機能の対象となる parse メソッドの一覧は、「3.3.17 XML 文書の解析（高速化定義ファイルを使用する場合）」の表 3-5 を参照してください。 高速パース機能の対象となる parse メソッド以外 <sup>1</sup> で処理する XML 文書 <sup>2</sup> の URI を指定すると、その XML 文書を parse メソッド以外で処理したときの結果が不正になるおそれがあります。
2	高速化定義ファイル内での名前空間の有効・無効の指定と、実際に解析を行うときのパーサに設定する名前空間の有効・無効の設定が一致しない場合、解析の実行結果が不正になるおそれがあります。

注 1

例えば、次のメソッドがあります。

- ・ javax.xml.transform.Transformer#transform メソッド
- ・ javax.xml.validation.SchemaFactory#newSchema メソッド
- ・ javax.xml.xpath.XPath#evaluate メソッド

注 2

スタイルシート、スキーマ文書または XPath 式の適用対象となる XML 文書などを指します。

### 6.17.6 絶対パスの指定方法に関する注意事項

表 6-34 高速パース機能に関する注意事項（絶対パスの指定方法）

項番	注意事項
1	絶対パスを指定する場合は、パス区切り文字に "/" を使用してください。
2	Windows の場合、絶対パスにはドライブ名を含めてください。 Windows の場合と UNIX の場合の、絶対パスの指定例を次に示します。 Windows の場合 D:/home/xml/training/training1.xml UNIX の場合 /home/xml/training/training1.xml

### 6.17.7 事前解析用 XML 文書に関する注意事項

表 6-35 高速パース機能に関する注意事項（事前解析用 XML 文書）

項番	注意事項
1	事前解析用 XML 文書に記述できる要素名および属性名の長さには上限があります。事前解析用 XML 文書に記述されている最も長い要素名および属性名の文字数を足した長さ が 21000 文字を超える場合、KECX09607-E エラーが発生して処理が中止されることがあります。

注

例えば、次の XML 文書では、最も長い要素名 (element または content) の文字数は 7 で、最も長い属性名 (index) の文字数は 5 なので、これらを足した文字数は 12 になります。

```
<root>
  <data index="1"/>
  <element>
    <content>example</content>
  </element>
</root>
```

6.17.8 チューニング情報に関する注意事項

表 6-36 高速パース機能に関する注意事項 (チューニング情報)

項番	注意事項
1	パース対象 XML 文書のシステム識別子のパス区切り文字には "/" を使用してください。
2	コンストラクタ引数に相対パスを指定して作成した File オブジェクトを、PreparsedObjectFactory クラスの newPreparsedObject メソッドに指定すると、チューニングサマリファイルおよび詳細ファイルに出力される事前解析用 XML 文書のファイル名のパス区切り文字が、次に示すように重複して出力される場合があります。 Preparsed XML: C:¥¥a.xml

6.17.9 解析性能に関する注意事項

表 6-37 高速パース機能に関する注意事項 (解析性能)

項番	注意事項
1	外部エンティティ参照を含む XML 文書に高速パース機能を適用すると、通常の解析と比べ、解析性能が向上しない場合があります。外部エンティティ参照を含む XML 文書に高速パース機能を適用する場合は、事前に性能評価を行ってください。

# 7

## メッセージ

この章では、Cosminexus XML Processor で出力されるメッセージの形式、およびメッセージの一覧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

---

## 7.1 メッセージの形式

---

Cosminexus XML Processor で出力されるメッセージは、メッセージ ID とそれに続くメッセージテキストで構成されます。

メッセージの形式

KECXnnnnnn-X メッセージテキスト

KECX

メッセージを出力したプログラムが Cosminexus XML Processor であることを表します。

nnnnnn

各コンポーネントで任意に割り当てられるメッセージ番号を表します。

X

メッセージの種別を表します。各種別を説明します。

E：エラー

W：警告

メッセージテキスト

出力されるメッセージを表します。

メッセージ中の { 数字 } の部分は、実際には該当する文字列が表示されます。該当する文字列に複数の語句が含まれる場合、コンマ ( , ) で区切って出力されます (例えば、メッセージ KECX06223-E では、該当する文字列が '対象名前空間, 名前' のように出力されることがあります)。

XML Schema の検証では、型の名前が不明な場合、メッセージ中の型の名前が挿入される文字列の個所に "AnonType\_xxx" が出力されます (例えば、メッセージ KECX06063-E では、"#AnonType\_xxx" のように出力されます)。

各メッセージは、メッセージ ID、メッセージ和文テキスト、メッセージ英文テキストの順に記載してあります。



## 7.2 メッセージ一覧

---

メッセージの一覧を記載します。

### 7.2.1 KECX01000 ~ KECX01999

#### **KECX01001-E**

---

メッセージキーに対応するエラーメッセージを、見つけることができません。

The error message corresponding to the message key can not be found.

#### **KECX01002-E**

---

次のメッセージをフォーマットしている間に、内部エラーが起きました。

An internal error occurred while formatting the following message:

#### **KECX01003-E**

---

解析途中でファイルの終端に達しました。

Premature end of file.

#### **KECX01004-E**

---

整形式なドキュメントにはルート要素が必要です。

The root element is required in a well-formed document.

#### **KECX01005-E**

---

CDATA セクションに無効な XML 文字 (Unicode:0x{0}) があります。

An invalid XML character (Unicode: 0x{0}) was found in the CDATA section.

#### **KECX01006-E**

---

ドキュメントの element content に無効な XML 文字 (Unicode:0x{0}) があります。

An invalid XML character (Unicode: 0x{0}) was found in the element content of the document.

#### **KECX01010-E**

---

XML 宣言に無効な XML 文字 (Unicode:0x{0}) があります。

An invalid XML character (Unicode: 0x{0}) was found in the XML declaration.

#### **KECX01011-E**

---

文字シーケンス ]]> は、CDATA セクションの終了を示す以外の目的で使用してはいけません。

## 7. メッセージ

The character sequence "]]>" must not appear in content unless used to mark the end of a CDATA section.

### **KECX01014-E**

---

XML 宣言中では、{0} のあとには ' = ' がなければいけません。

The ' = ' character must follow "{0}" in the XML declaration.

### **KECX01015-E**

---

XML 宣言中では、{0} の値は引用符で囲まなければいけません。

The value following "{0}" in the XML declaration must be a quoted string.

### **KECX01016-E**

---

XML 宣言中は ?> で終了する必要があります。

The XML declaration must end with "?>".

### **KECX01017-E**

---

XML 宣言には version が必要です。

The version is required in the XML declaration.

### **KECX01018-E**

---

XML 宣言のバージョン擬似属性の前には、空白が必要です。

White space is required before the version pseudo attribute in the XML declaration.

### **KECX01019-E**

---

XML 宣言のエンコーディング擬似属性の前には、空白が必要です。

White space is required before the encoding pseudo attribute in the XML declaration.

### **KECX01020-E**

---

XML 宣言のエンコーディング擬似属性の前には、空白が必要です。

White space is required before the encoding pseudo attribute in the XML declaration.

### **KECX01021-E**

---

ルート要素より前のドキュメント中の markup は整形形式でなくてはなりません。

The markup in the document preceding the root element must be well-formed.

**KECX01022-E**

---

ルート要素に続くドキュメント中の markup は整形形式でなくてはなりません。

The markup in the document following the root element must be well-formed.

**KECX01023-E**

---

Doctype はすでに宣言されています。

Already seen doctype.

**KECX01024-E**

---

prolog には、content を指定できません。または、parse メソッドの引数に空文字列が指定された可能性があります。

Content is not allowed in prolog.

**KECX01025-E**

---

prolog には、reference を指定できません。

Reference is not allowed in prolog.

**KECX01026-E**

---

trailing section には、content を指定できません。

Content is not allowed in trailing section.

**KECX01027-E**

---

trailing section には、reference を指定できません。

Reference is not allowed in trailing section.

**KECX01028-E**

---

standalone ドキュメント宣言の指定は、yes か no にする必要があります。{0} であってはいけません。

The standalone document declaration value must be "yes" or "no", not "{0}".

**KECX01030-E**

---

要素タイプ {0} は、終了タグ </{0}> で終わらなければいけません。

The element type "{0}" must be terminated by the matching end-tag "</{0}>".

**KECX01031-E**

---

要素タイプ {0} のあとには、> または /> のどちらかがなければいけません。

## 7. メッセージ

Element type "{0}" must be followed by either attribute specifications, ">" or "/>".

### KECX01032-E

---

要素タイプ {0} に関連した属性名 {1} のあとには、'=' がなければいけません。

Attribute name "{1}" associated with an element type "{0}" must be followed by the '=' character.

### KECX01033-E

---

要素タイプ {0} に関連した属性指定 {1} には、開始引用符がありません。

Open quote is expected for attribute "{1}" associated with an element type "{0}".

### KECX01034-E

---

要素タイプ {0} に関連した属性指定 {1} には、終了引用符がありません。

Close quote is expected for attribute "{1}" associated with an element type "{0}".

### KECX01035-E

---

属性 {1} は、すでに要素 {0} で指定されています。

Attribute "{1}" was already specified for element "{0}".

### KECX01036-E

---

名前空間 {2} に境界を付ける属性 {1} は、要素 {0} にすでに結び付けられています。

Attribute "{1}" bound to namespace "{2}" was already specified for element "{0}".

### KECX01037-E

---

要素タイプ {0} の終了タグは、'>' でなければいけません。

The end-tag for element type "{0}" must end with a '>' delimiter.

### KECX01038-E

---

要素の中身は、整形形式の文字または markup でなければいけません。

The content of elements must consist of well-formed character data or markup.

### KECX01039-E

---

要素内容には DOCTYPE を指定できません。

A DOCTYPE is not allowed in content.

### KECX01042-E

---

要素 {0} は、同じエンティティ内で開始・終了しなければいけません。

The element "{0}" must start and end within the same entity.

#### **KECX01043-E**

---

XML 文書は、同じエンティティ内で開始・終了しなければいけません。

XML document structures must start and end within the same entity.

#### **KECX01044-E**

---

要素が {0} の属性 {1} に無効な XML 文字 (Unicode:0x{2}) があります。

An invalid XML character (Unicode: 0x{2}) was found in the value of attribute "{1}" and element is "{0}".

#### **KECX01045-E**

---

コメントに無効な XML 文字 (Unicode:0x{0}) があります。

An invalid XML character (Unicode: 0x{0}) was found in the comment.

#### **KECX01046-E**

---

処理命令に無効な XML 文字 (Unicode:0x{0}) があります。

An invalid XML character (Unicode: 0x{0}) was found in the processing instruction.

#### **KECX01048-E**

---

テキスト宣言に無効な XML 文字 (Unicode:0x{0}) があります。

An invalid XML character (Unicode: 0x{0}) was found in the text declaration.

#### **KECX01050-E**

---

要素タイプ {0} に関連した属性値 {1} は、'<' を含んではいけません。

The value of attribute "{1}" associated with an element type "{0}" must not contain the '<' character.

#### **KECX01052-E**

---

コメントは <!-- で始めなければいけません。

Comment must start with "<!--".

#### **KECX01053-E**

---

コメント内では、-- は使用できません。

The string "--" is not permitted within comments.

### **KECX01056-E**

---

処理命令はターゲット名で始めなければいけません。

The processing instruction must begin with the name of the target.

### **KECX01057-E**

---

処理命令ターゲットとデータの間には空白が必要です。

White space is required between the processing instruction target and data.

### **KECX01059-E**

---

処理命令ターゲットは [xX][mM][lL] の指定ができません。

The processing instruction target matching "[xX][mM][lL]" is not allowed.

### **KECX01062-E**

---

XML バージョン {0} はサポートしていません。XML 1.0 だけをサポートしています。

XML version "{0}" is not supported, only XML 1.0 is supported.

### **KECX01063-E**

---

文字参照では、10 進コードは &# の直後に指定しなければいけません。

A decimal representation must immediately follow the "&#" in a character reference.

### **KECX01064-E**

---

文字参照では 16 進コードは &#x の直後に指定しなければいけません。

A hexadecimal representation must immediately follow the "&#x" in a character reference.

### **KECX01065-E**

---

文字参照は ';' で終了する必要があります。

The character reference must end with the ';' delimiter.

### **KECX01066-E**

---

文字参照 &#{0}; は無効な XML 文字指定です。

Character reference "&#{0};" is an invalid XML character.

### **KECX01067-E**

---

エンティティ名は、エンティティ参照の中で '&' の直後に続かなければいけません。

The entity name must immediately follow the '&' in the entity reference.

**KECX01068-E**

---

エンティティ {0} の参照は、';' で終了する必要があります。

The reference to entity "{0}" must end with the ';' delimiter.

**KECX01070-E**

---

テキスト宣言中では、{0} のあとには '=' がなければいけません。

The '=' character must follow "{0}" in the text declaration.

**KECX01071-E**

---

テキスト宣言中では、{0} の値は引用符で囲まなければいけません。

The value following "{0}" in the text declaration must be a quoted string.

**KECX01072-E**

---

テキスト宣言中の {0} を閉じる引用符がありません。

closing quote in the value following "{0}" in the text declaration is missing.

**KECX01073-E**

---

テキスト宣言中のバージョン擬似属性の前には空白が必要です。

White space is required before the version pseudo attribute in the text declaration.

**KECX01074-E**

---

テキスト宣言中のエンコーディング擬似属性の前には空白が必要です。

White space is required before the encoding pseudo attribute in the text declaration.

**KECX01076-E**

---

テキスト宣言中で、エンコーディング宣言をする必要があります。

The encoding declaration is required in the text declaration.

**KECX01077-E**

---

これ以上の擬似属性指定はできません。

No more pseudo attributes are allowed.

**KECX01078-E**

---

擬似属性指定が足りません。

More pseudo attributes are expected.

### **KECX01079-E**

---

擬似属性指定名称が指定されていません。

A pseudo attribute name is expected.

### **KECX01082-E**

---

{0} は無効なエンコーディング名です。

Invalid encoding name "{0}".

### **KECX01083-E**

---

エンコーディング {0} で与えられたバイトオーダーはサポートしていません。

Given byte order for encoding "{0}" is not supported.

### **KECX01087-E**

---

公開識別子に無効な XML 文字 (Unicode:0x{0}) があります。

An invalid XML character (Unicode: 0x{0}) was found in the public identifier.

### **KECX01088-E**

---

システム識別子で無効な XML 文字 (Unicode:0x{0}) が存在します。

An invalid XML character (Unicode: 0x{0}) was found in the system identifier.

### **KECX01089-E**

---

DOCTYPE 宣言で、キーワード SYSTEM のあとに空白が必要です。

White space is required after keyword SYSTEM in DOCTYPE decl.

### **KECX01090-E**

---

システム識別子は単一引用符か二重引用符で開始する必要があります。

The system identifier must begin with either a single or double quote character.

### **KECX01091-E**

---

システム識別子は単一引用符か二重引用符で終了する必要があります。

The system identifier must end with the matching quote character.

### **KECX01092-E**

---

DOCTYPE 宣言で、キーワード PUBLIC のあとに空白が必要です。

White spaces are required after keyword PUBLIC in DOCTYPE decl.



**KECX01093-E**

---

公開識別子は単一引用符か二重引用符で開始する必要があります。

The public identifier must begin with either a single or double quote character.

**KECX01094-E**

---

公開識別子は単一引用符か二重引用符で終了する必要があります。

The public identifier must end with the matching quote character.

**KECX01096-E**

---

公開識別子とシステム識別子の間には空白が必要です。

White spaces are required between publicId and systemId.

**KECX01097-E**

---

DOCTYPE 宣言で、<!DOCTYPE 指定のあとには空白が必要です。

White space is required after "<!DOCTYPE" in the document type declaration.

**KECX01098-E**

---

DOCTYPE 宣言で、<!DOCTYPE 指定のあとにはルート要素タイプが現れる必要があります。

The root element type must appear after "<!DOCTYPE" in the document type declaration.

**KECX01099-E**

---

ルート要素タイプ {0} の DOCTYPE 宣言は、'>' で終わる必要があります。

The document type declaration for root element type "{0}" must end with '>'.

**KECX01100-E**

---

パラメタエンティティ参照 {0} は、DTD 内部サブセット内の markup の内側には記述できません。

The parameter entity reference "%{0};" cannot occur within markup in the internal subset of the DTD.

**KECX01101-E**

---

markup 宣言は、整形形式なドキュメントタイプ宣言を含むか、またはポイントする必要があります。

The markup declarations contained or pointed to by the document type declaration must be well-formed.

#### **KECX01103-E**

---

要素タイプ宣言で、<!ELEMENT のあとに空白が必要です。

White space is required after "<!ELEMENT" in the element type declaration.

#### **KECX01104-E**

---

要素タイプの指定には、要素タイプ宣言が必要です。

The element type is required in the element type declaration.

#### **KECX01105-E**

---

要素タイプ宣言内で、要素タイプ {0} のあとに空白が必要です。

White space is required after the element type "{0}" in the element type declaration.

#### **KECX01107-E**

---

要素タイプ宣言 {0} は、'>' で終了する必要があります。

The declaration for element type "{0}" must end with '>'.

#### **KECX01108-E**

---

要素タイプ {0} の宣言では、'(' または要素タイプが必要です。

A '(' character or an element type is required in the declaration of element type "{0}".

#### **KECX01109-E**

---

要素タイプ {0} の宣言では、')' が必要です。

A ')' is required in the declaration of element type "{0}".

#### **KECX01110-E**

---

要素タイプ {0} の宣言で、要素タイプの指定が必要です。

An element type is required in the declaration of element type "{0}".

#### **KECX01112-E**

---

混合内容モデル {0} は、子要素のタイプが必要な場合に ) \* で終わらなくてはなりません。

The mixed content model "{0}" must end with ")\*" when the types of child elements are constrained.

#### **KECX01113-E**

---

属性リスト宣言では、<!ATTLIST のあとに空白が必要です。

White space is required after "<!ATTLIST" in the attribute-list declaration.

**KECX01114-E**

---

要素タイプは、属性リスト宣言が必要です。

The element type is required in the attribute-list declaration.

**KECX01115-E**

---

要素 {0} の属性リスト宣言では、属性名の前に空白が必要です。

White space is required before the attribute name in the attribute-list declaration for element "{0}".

**KECX01116-E**

---

属性名は要素 {0} の属性リストを指定する必要があります。

The attribute name must be specified in the attribute-list declaration for element "{0}".

**KECX01117-E**

---

要素 {0} の属性 {1} の宣言で、属性タイプの前に空白が必要です。

White space is required before the attribute type in the declaration of attribute "{1}" for element "{0}".

**KECX01118-E**

---

要素 {0} の属性 {1} の宣言で、属性タイプが必要です。

The attribute type is required in the declaration of attribute "{1}" for element "{0}".

**KECX01119-E**

---

要素 {0} の属性 {1} の宣言で、デフォルト属性の前に空白が必要です。

White space is required before the attribute default in the declaration of attribute "{1}" for element "{0}".

**KECX01120-W**

---

指定された要素 {0} の同じ属性 {1} 宣言のために、一つ以上の属性定義が与えられます。

More than one attribute definition is provided for the same attribute "{1}" of a given element "{0}".

**KECX01121-E**

---

属性宣言 {1} で、NOTATION のあとに空白が必要です。

White space must appear after "NOTATION" in the "{1}" attribute declaration.

**KECX01122-E**

---

属性宣言 {1} で NOTATION に続いて ' (' の指定が必要です。

## 7. メッセージ

The '(' character must follow "NOTATION" in the "{1}" attribute declaration.

### **KECX01123-E**

---

属性宣言 {1} の記法タイプ指定で、記法名が必要です。

The notation name is required in the notation type list for the "{1}" attribute declaration.

### **KECX01124-E**

---

属性宣言 {1} の記法タイプ指定では、記法タイプは ')' で終わらなければいけません。

The notation type list must end with ')' in the "{1}" attribute declaration.

### **KECX01125-E**

---

属性宣言 {1} の列挙されたタイプリストで名称トークンが必要です。

The name token is required in the enumerated type list for the "{1}" attribute declaration.

### **KECX01126-E**

---

属性宣言 {1} の列挙されたタイプリストは ')' で終わらなくてははいけません。

The enumerated type list must end with ')' in the "{1}" attribute declaration.

### **KECX01127-E**

---

属性宣言 {1} で、FIXED のあとに空白が必要です。

White space must appear after "FIXED" in the "{1}" attribute declaration.

### **KECX01128-E**

---

含まれる条件つきセクションは、]]> で終了する必要があります。

The included conditional section must end with "]]>".

### **KECX01129-E**

---

除外される条件つきセクションは、]]> で終了する必要があります。

The excluded conditional section must end with "]]>".

### **KECX01130-E**

---

パラメタエンティティ参照のエンティティ名は、'%' に続いて記述する必要があります。

The entity name must immediately follow the '%' in the parameter entity reference.

### **KECX01131-E**

---

パラメタエンティティ参照 {0} は、';' で終了する必要があります。

The parameter entity reference "%{0};" must end with the ';' delimiter.

---

**KECX01132-E**

---

エンティティ宣言で、<!ENTITY のあとに空白が必要です。

White space is required after "<!ENTITY" in the entity declaration.

---

**KECX01133-E**

---

パラメタエンティティ宣言で、<!ENTITY と '%' の間に空白が必要です。

White space is required between "<!ENTITY" and the '%' character in the parameter entity declaration.

---

**KECX01135-E**

---

エンティティの名前は、エンティティ宣言が必要です。

The name of the entity is required in the entity declaration.

---

**KECX01136-E**

---

エンティティ宣言で、エンティティ名 {0} と定義の間に空白が必要です。

White space is required between the entity name "{0}" and the definition in the entity declaration.

---

**KECX01137-E**

---

エンティティ {0} の宣言で、NDATA と記法名の間に空白が必要です。

White space is required between "NDATA" and the notation name in the declaration for the entity "{0}".

---

**KECX01138-E**

---

エンティティ {0} の宣言で、NDATA の前に空白が必要です。

White space is required before "NDATA" in the declaration for the entity "{0}".

---

**KECX01139-E**

---

エンティティ {0} 宣言で、NDATA のあとに記法名が必要です。

The notation name is required after "NDATA" in the declaration for the entity "{0}".

---

**KECX01140-E**

---

エンティティ {0} の宣言は、>' で終わらなければいけません。

The declaration for the entity "{0}" must end with '>'.

---

**KECX01146-E**

---

記法宣言で、<!NOTATION のあとに空白が必要です。

White space is required after "<!NOTATION" in the notation declaration.

---

**KECX01147-E**

---

記法宣言では記法名が必要です。

The name of the notation is required in the notation declaration.

---

**KECX01148-E**

---

記法宣言で、記法名 {0} のあとに空白が必要です。

White space is required after the notation name "{0}" in the notation declaration.

---

**KECX01149-E**

---

記法 {0} の宣言は、システム識別子または公開識別子を含まなければいけません。

The declaration for the notation "{0}" must include a system or public identifier.

---

**KECX01150-E**

---

表記法 {0} の宣言は、'>' で終わらなければいけません。

The declaration for the notation "{0}" must end with '>'.

---

**KECX01151-E**

---

要素タイプ {1} は、要素宣言 {0} の内容モデルですすでに宣言されています。

The element type "{1}" was already specified in the content model of the element decl "{0}".

---

**KECX01154-E**

---

ID 属性 {0} は、宣言されている #IMPLIED または #REQUIRED のデフォルトでなければいけません。

The ID attribute "{0}" must have a declared default of "#IMPLIED" or "#REQUIRED".

---

**KECX01155-E**

---

ID 型の属性値 {0} は名前でなければいけません。

Attribute value "{0}" of type ID must be a name.

---

**KECX01156-E**

---

ID 型の属性値 {0} は、文書中に同じ名称があってははいけません。

Attribute value "{0}" of type ID must be unique within the document.

### **KECX01157-E**

---

IDREF 型の属性値 {0} は名前でなければいけません。

Attribute value "{0}" of type IDREF must be a name.

### **KECX01158-E**

---

IDREF 型の属性値 {0} は一つ以上の名前でなければいけません。

Attribute value "{0}" of type IDREFS must be one or more names.

### **KECX01159-E**

---

パラメタエンティティ {0} の置換テキストは、適切にネストされた宣言を含まなくてははいけません。

The replacement text of parameter entity "{0}" must include properly nested declarations.

### **KECX01161-E**

---

属性 {1} は要素タイプ {0} の宣言でなくてははいけません。

Attribute "{1}" must be declared for element type "{0}".

### **KECX01162-E**

---

{1} の値を持つ属性 {0} は、{2} のリスト中の値でなければいけません。

Attribute "{0}" with value "{1}" must have a value from the list "{2}".

### **KECX01163-E**

---

属性 {0} の値 {1} は、スタンドアロン文書で正規化によって ({2} に) 変更されてはいけません。

The value "{1}" of attribute "{0}" must not be changed by normalization (to "{2}") in a standalone document.

### **KECX01164-E**

---

要素タイプ {0} のコンテンツが不完全です。{1} と一致してはいけません。

The content of element type "{0}" is incomplete, it must match "{1}".

### **KECX01165-E**

---

要素タイプ {0} のコンテンツは、{1} と一致してはいけません。

The content of element type "{0}" must match "{1}".

---

**KECX01166-E**

---

要素タイプ {0} の属性 {1} はデフォルト値を持っていて、かつスタンドアロン文書で指定されていなければいけません。

Attribute "{1}" for element type "{0}" has a default value and must be specified in a standalone document.

---

**KECX01168-E**

---

要素タイプ {0} は 2 度以上宣言してはいけません。

Element type "{0}" must not be declared more than once.

---

**KECX01169-E**

---

要素タイプ {0} を宣言しなくてははいけません。

Element type "{0}" must be declared.

---

**KECX01170-E**

---

ドキュメントは無効です。文法がありません。

Document is invalid: no grammar found.

---

**KECX01171-E**

---

識別子 {0} を持つ要素が、ドキュメント中になければいけません。

An element with the identifier "{0}" must appear in the document.

---

**KECX01173-E**

---

値 {2} の属性 {1} は、{3} の値を持っていないてはいけません。

Attribute "{1}" with value "{2}" must have a value of "{3}".

---

**KECX01174-E**

---

要素タイプ {0} は ID 型の属性 {1} をすでに持っています。ID 型の第二の属性 {2} は許可されません。

Element type "{0}" already has attribute "{1}" of type ID, a second attribute "{2}" of type ID is not permitted.

---

**KECX01175-E**

---

要素タイプ {0} は NOTATION 型の属性 {1} をすでに持っています。NOTATION 型の第二の属性 {2} は許可されません。



Element type "{0}" already has attribute "{1}" of type NOTATION, a second attribute "{2}" of type NOTATION is not permitted.

---

#### **KECX01176-E**

属性 {0} の記法タイプリストで参照される場合は、記法 {1} を宣言する必要があります。

The notation "{1}" must be declared when referenced in the notation type list for attribute "{0}".

---

#### **KECX01177-E**

属性 {0} の解析対象外エンティティで参照される場合は、記法 {1} を宣言する必要があります。

The notation "{1}" must be declared when referenced in the unparsed entity declaration for "{0}".

---

#### **KECX01178-E**

外部解析対象エンティティで宣言された {0} の参照は、スタンドアロン文書では許可されません。

The reference to entity "{0}" declared in an external parsed entity is not permitted in a standalone document.

---

#### **KECX01179-E**

要素タイプ {0} の指定のために属性 {1} は必要です。

Attribute "{1}" is required and must be specified for element type "{0}".

---

#### **KECX01180-E**

スタンドアロン文書の要素コンテンツを持つ、外部解析対象エンティティの、要素宣言に空白があってははいけません。

White space must not occur between elements declared in an external parsed entity with element content in a standalone document.

---

#### **KECX01181-E**

NMTOKEN 型の属性値 {0} は name token でなければいけません。

Attribute value "{0}" of type NMTOKEN must be a name token.

---

#### **KECX01182-E**

NMTOKENS 型の属性値 {0} は一つ以上の name token でなければいけません。

Attribute value "{0}" of type NMTOKENS must be one or more name tokens.

---

#### **KECX01183-E**

ドキュメントルート要素 {1} は、DOCTYPE root{0} と一致する必要があります。

## 7. メッセージ

Document root element "{1}", must match DOCTYPE root "{0}".

### **KECX01185-E**

---

エンティティ・データタイプの有効性: 有効な文法による初期化メソッドの呼び出しに失敗しました。

ENTITYDatatype Validator: Failed Need to call initialize method with a valid Grammar reference.

### **KECX01186-E**

---

エンティティ {0} は解析対象外ではありません。

ENTITY "{0}" is not unparsed.

### **KECX01188-E**

---

ENTITIES 型, IDREFS 型, NMTOKENS 型の値に, 空のリスト指定はできません。

Value of type ENTITIES, IDREFS, and NMTOKENS cannot be empty list.

### **KECX01189-E**

---

外部エンティティ参照 &{0}; は許可されていない属性です。

The external entity reference "&{0};" is not permitted in an attribute value.

### **KECX01190-E**

---

エンティティ {0} が参照されましたが, 宣言されていません。

The entity "{0}" was referenced, but not declared.

### **KECX01191-E**

---

解析対象外エンティティ参照 &{0}; は許可されません。

The unparsed entity reference "&{0};" is not permitted.

### **KECX01192-E**

---

再帰的なエンティティ参照 {0} です (path{1} を参照してください)。

Recursive entity reference "{0}". (Reference path: {1}),

### **KECX01197-E**

---

要素 {1} の接頭辞 {0} は結び付けられません。

The prefix "{0}" for element "{1}" is not bound.

**KECX01198-E**

---

要素タイプ {0} に関連した属性 {1} の接頭辞 {2} は結び付けられません。

The prefix "{2}" for attribute "{1}" associated with an element type "{0}" is not bound.

**KECX01199-E**

---

属性 {0} 値が無効です。接頭辞を付けた名前空間バインディングは、空ではありません。

The value of the attribute "{0}" is invalid. Prefixed namespace bindings may not be empty.

**KECX01200-E**

---

名前空間接頭辞 {0} が宣言されませんでした。

The namespace prefix "{0}" was not declared.

**KECX01201-E**

---

属性 {0} のデフォルト値 {1} は、この属性タイプで使えない用語です。

The defaultValue "{1}" of attribute "{0}" is not legal as for the lexical constraints of this attribute type.

**KECX01202-E**

---

宣言に開始引用符がありません。

OpenQuoteMissingInDecl

**KECX01203-E**

---

リテラルで無効な文字があります。

InvalidCharInLiteral

**KECX01204-E**

---

未知の属性タイプ {0} が現れました。

!!! unknown attribute type {0}

**KECX01206-W**

---

警告：検証はされましたが、org.xml.sax.ErrorHandler はセットされていません。これは望ましい処理ではありません。パーサはデフォルトの ErrorHandler を使って最初の {0} のエラーを報告します。これをセットするために、setErrorHandler メソッドを呼び出してください。

Warning: validation was turned on but an org.xml.sax.ErrorHandler was not set, which is probably not what is desired. Parser will use a default ErrorHandler to print the first {0} errors. Please call the 'setErrorHandler' method to fix this.

### **KECX01207-E**

---

この値を取得してはいけません。DeferredDocumentTypeImpl#synchronizeInfo:  
node.getNodeType() の値は {0} で、クラス名は {1} です。

Should never get here. : DeferredDocumentTypeImpl#synchronizeInfo: node.getNodeType() = {0},  
class = {1}

### **KECX01208-E**

---

{1}- バイト UTF-8 シーケンスの {0} バイト目が不正です。

Invalid byte {0} of {1}-byte UTF-8 sequence.

### **KECX01209-E**

---

{1}- バイト UTF-8 シーケンスの {0} バイト目が必要です。

Expected byte {0} of {1}-byte UTF-8 sequence.

### **KECX01210-E**

---

UTF-8 シーケンスの高位サロゲートビットは 0x10 を超えてはいませんが、0x{0} を見つけました。

High surrogate bits in UTF-8 sequence must not exceed 0x10 but found 0x{0}.

### **KECX01211-E**

---

オペレーション {0} は {1}reader にサポートされていません。

Operation "{0}" not supported by {1} reader.

### **KECX01212-E**

---

バイト {0} は、7-bit ASCII 文字セットのメンバではありません。

Byte "{0}" is not a member of the (7-bit) ASCII character set.

### **KECX01213-W**

---

エンティティ {0} は、2 度以上宣言されています。

Entity "{0}" is declared more than once.

### **KECX01214-E**

---

要素 {0} の接頭辞に xmlns は指定できません。

Element "{0}" cannot have "xmlns" as its prefix.

**KECX01215-E**

---

接頭辞 xmlns は任意の名前空間に明示的に結び付けることができません。また、xmlns のための名前空間は、任意の接頭辞に明示的に結び付けることができません。

The prefix "xmlns" cannot be bound to any namespace explicitly; neither can the namespace for "xmlns" be bound to any prefix explicitly.

**KECX01216-E**

---

接頭辞 xml は、その通常の名前空間以外の任意の名前空間に結び付けられません。また、xml のための名前空間は、xml 以外に任意の接頭辞に結び付けることができません。

The prefix "xml" cannot be bound to any namespace other than its usual namespace; neither can the namespace for "xml" be bound to any prefix other than "xml".

**KECX01217-E**

---

実体参照が完全な宣言として使用される場合、パラメタ実体 {0} の置換テキストは、適切にネストされた宣言を含まなくてはいけません。

The replacement text of parameter entity "{0}" must include properly nested declarations when the entity reference is used as a complete declaration.

**KECX01218-E**

---

パラメタ実体 {0} の置換テキストは、条件付きのセクション全体、単なる INCLUDE、または IGNORE を含んでいなければいけません。

The replacement text of parameter entity "{0}" must include the entire conditional section or just INCLUDE or IGNORE.

**KECX01219-E**

---

名前空間が有効な場合、コロンは名前 {0} の中で許可されません。

A colon is not allowed in the name '{0}' when namespaces are enabled.

**KECX01220-E**

---

フィーチャー "http://cosminexus.com/xml/features/disallow-doctype-decl" に true がセットされている場合は、DOCTYPE は許可されません。

DOCTYPE is disallowed when the feature "http://cosminexus.com/xml/features/disallow-doctype-decl" set to true.

**KECX01221-E**

---

XML バージョン {0} はサポートしていません。XML 1.0 と XML 1.1 だけをサポートしています。

XML version "{0}" is not supported, only XML 1.0 and XML 1.1 are supported.

### **KECX01222-E**

---

名前空間が有効な場合，ID 型の属性値 {0} は NCName でなければいけません。

Attribute value "{0}" of type ID must be an NCName when namespaces are enabled.

### **KECX01223-E**

---

名前空間が有効な場合，IDREF 型の属性値 {0} は NCName でなければいけません。

Attribute value "{0}" of type IDREF must be an NCName when namespaces are enabled.

### **KECX01224-E**

---

要素または属性は QName 生成と一致しません。QName::=(NCName:'')?NCName

Element or attribute do not match QName production: QName::=(NCName:'')?NCName.

### **KECX01225-E**

---

パーサは，このドキュメント中に {0} 以上のエンティティ拡張を検出しました。これはアプリケーションによって課された限界です。

The parser has encountered more than "{0}" entity expansions in this document; this is the limit imposed by the application.

### **KECX01226-E**

---

ある特定のエンコーディングに含めると決定されたエンティティには，該当のエンコーディングで不正なシーケンスを含めることはできません。

An entity determined to be in a certain encoding must not contain sequences illegal in that encoding.

### **KECX01227-E**

---

列挙値 {1} が要素 {0} の属性 {2} の宣言中で複数回指定されました。一つの列挙属性宣言内の NMTOKENS はすべて異なっている必要があります。

The enumeration value "{1}" was specified more than once in the declaration of attribute "{2}" for element "{0}". The NMTOKENS in a single Enumeration attribute declaration must all be distinct.

### **KECX01228-E**

---

列挙値 {1} が要素 {0} の属性 {2} の宣言中で複数回指定されました。一つの記法型属性宣言内の NOTATION 名はすべて異なっている必要があります。

The enumeration value "{1}" was specified more than once in the declaration of attribute "{2}" for element "{0}". The NOTATION names in a single NotationType attribute declaration must all be distinct.

**KECX01229-E**

---

要素型 {0} の内容は {1} に一致する必要があります。型 {2} の子は許可されていません。

The content of element type "{0}" must match "{1}". Children of type "{2}" are not allowed.

**KECX01230-E**

---

EMPTY と宣言された要素型 {0} は、NOTATION 型の属性 {1} を宣言できません。

Element type "{0}" which was declared EMPTY cannot declare attribute "{1}" of type NOTATION.

**KECX01231-E**

---

記法 {0} の宣言は一意ではありません。同じ名前は複数の記法宣言内で宣言してはいけません。

The declaration for the notation "{0}" is not unique. A given Name must not be declared in more than one notation declaration.

**KECX01301-E**

---

メッセージキーに対応するエラーメッセージが見つかりません。

The error message corresponding to the message key can not be found.

**KECX01302-E**

---

次のメッセージをフォーマットする間に、内部エラーが発生しました。

An internal error occurred while formatting the following message:¥n

**KECX01303-E**

---

テキストの指定された範囲は、DOMString に入りません。

The specified range of text does not fit into a DOMString.

**KECX01304-E**

---

許可されていない箇所に対して、ノードを挿入しようとしています。

An attempt was made to insert a node where it is not permitted.

**KECX01305-E**

---

インデックスまたはサイズが、負の値または許可された値より大きいです。

The index or size is negative, or greater than the allowed value.

**KECX01306-E**

---

ほかの場所で使用されている属性を追加しようとしています。

An attempt is made to add an attribute that is already in use elsewhere.

---

**KECX01308-E**

---

不正な、または無効な XML 文字が指定されています。

An invalid or illegal XML character is specified.

---

**KECX01310-E**

---

存在しない、または使用できないオブジェクトを使用しようとしています。

An attempt is made to use an object that is not, or is no longer, usable.

---

**KECX01311-E**

---

名前空間に関して、誤った方法でオブジェクトを作成または変更しようとしています。

An attempt is made to create or change an object in a way which is incorrect with regard to namespaces.

---

**KECX01312-E**

---

存在しないコンテキスト中のノードを参照しようとしています。

An attempt is made to reference a node in a context where it does not exist.

---

**KECX01313-E**

---

そのインプリメンテーションは、要求された型のオブジェクトまたはオペレーションをサポートしません。

The implementation does not support the requested type of object or operation.

---

**KECX01315-E**

---

修正してはいけない箇所に対して、オブジェクトを修正しようとしています。

An attempt is made to modify an object where modifications are not allowed.

---

**KECX01318-E**

---

ノードを作成した文書以外の文書中でノードが使用されています。

A node is used in a different document than the one that created it.

---

**KECX01319-E**

---

パラメタ {0} は認識されますが、要求された値はセットされません。

The parameter {0} is recognized but the requested value cannot be set.

---

**KECX01320-E**

---

パラメタ {0} は認識されません。



The parameter {0} is not recognized.

---

#### **KECX01323-E**

---

メソッド呼び出し前のイベントの初期化で、イベントの型が指定されませんでした。

The Event's type was not specified by initializing the event before the method was called.

---

#### **KECX01324-E**

---

結果として生じている文字列は長過ぎるので、DOMString に入りません。{0}

The resulting string is too long to fit in a DOMString: '{0}'.

---

#### **KECX01325-E**

---

このパラメタ名の値型は期待された値型と互換性がありません。

The value type for this parameter name is incompatible with the expected value type.

---

#### **KECX01326-E**

---

{1} ノードのテキスト {0} に、無効な XML 文字が含まれています。

The text {0} of the {1} node contains invalid XML characters.

---

#### **KECX01327-E**

---

{1} という名前の {0} ノードに、無効な XML 文字が含まれています。

The {0} node named {1} contains invalid XML characters.

---

#### **KECX01328-E**

---

CDATA セクションに CDATA セクション終了マークの「]]>」が含まれています。

CDATA sections containing the CDATA section termination marker ']]>'

---

#### **KECX01329-E**

---

DOCTYPE 宣言は許可されていません。

DOCTYPE declaration is not allowed.

---

#### **KECX01330-E**

---

エンコーディング {0} はサポートされていません。

The encoding {0} is not supported.

---

#### **KECX01331-E**

---

正規化中、DOM 内に無効な XML 文字 (Unicode: 0x{0}) がありました。

## 7. メッセージ

An invalid XML character (Unicode: 0x{0}) was found in the DOM during normalization.

### **KECX01332-E**

---

属性 {0} 値 {1} が宣言されていないエンティティを参照しました。

The attribute "{0}" value "{1}" referenced an entity that was not declared.

### **KECX01333-E**

---

要素 {0} の名前空間正規化中、null のローカル名がありました。

A null local name was encountered during namespace normalization of element {0}.

### **KECX01334-E**

---

属性 {0} の名前空間正規化中、null のローカル名がありました。

A null local name was encountered during namespace normalization of attribute {0}.

### **KECX01335-E**

---

DOM ツリーの構築に使用されるドキュメントファクトリ {0} のクラス名は  
org.w3c.dom.Document 型ではありません。

The class name of the document factory "{0}" used to construct the DOM tree is not of type  
org.w3c.dom.Document.

### **KECX01336-E**

---

DOM ツリーの構築に使用されるドキュメントファクトリ {0} のクラス名はありませんでした。

The class name of the document factory "{0}" used to construct the DOM tree could not be found.

### **KECX01337-E**

---

{0} という名前のクラスは org.w3c.dom.Document として構築できませんでした。

The class named "{0}" could not be constructed as a org.w3c.dom.Document.

### **KECX01338-E**

---

プロパティ {1} を設定する前にプロパティ {0} を設定する必要があります。

Property '{0}' must be set before setting property '{1}'.

### **KECX01339-E**

---

指定したソースは null にできません。

The source specified cannot be null.

**KECX01401-E**

---

引数 {0} が null です。

Argument '{0}' is null.

**KECX01402-E**

---

serializer に供給される writer がありません。

No writer supplied for serializer.

**KECX01403-E**

---

メソッド {0} はこの factory でサポートされていません。

The method '{0}' is not supported by this factory.

**KECX01404-E**

---

serializer は、シリアライゼーションの最中にリセットされないこともあります。

The serializer may not be reset in the middle of serialization.

**KECX01405-E**

---

内部エラーです。要素の状態がゼロです。

Internal error: element state is zero.

**KECX01406-E**

---

rawName がありません。また、localName は null です。

There is no rawName and localName is null.

**KECX01407-E**

---

要素名 {0} は QName ではありません。

The element name '{0}' is not a QName.

**KECX01408-E**

---

要素 {0} は任意の名前空間に属しません。接頭辞は非宣言とされたか、またはある名前空間に結び付けられました。

Element '{0}' does not belong to any namespace: prefix could be undeclared or bound to some namespace.

**KECX01409-E**

---

属性名 {0} は QName ではありません。

## 7. メッセージ

The attribute name '{0}' is not a QName.

### **KECX01410-E**

---

属性 {0} は、任意の名前空間に属しません。接頭辞は非宣言とされたか、またはある名前空間に結び付けられました。

Attribute '{0}' does not belong to any namespace: prefix could be undeclared or bound to some namespace.

### **KECX01411-E**

---

名前空間宣言の構文に誤りがあります。{0}

Namespace declaration syntax is incorrect: {0}.

### **KECX01412-E**

---

文字シーケンス ]]> は、CDATA セクションの終了を示す以外の目的で使用してはいけません。

The character sequence "]]>" must not appear in content unless used to mark the end of a CDATA section.

### **KECX01413-W**

---

CDATA セクション終了マーカー ]]> を含んでいる CDATA セクションを分割します。

Splitting a CDATA section containing the CDATA section termination marker "]]>".

### **KECX01414-E**

---

リソース {0} はありませんでした。

The resource '{0}' could not be found.

### **KECX01415-E**

---

リソース {0} はロードできませんでした。{1}

The resource '{0}' could not be loaded. {1}

### **KECX01416-E**

---

シリアライゼーションはユーザ要求により停止しました。

Serialization stopped at user request.

### **KECX01417-E**

---

出力指定なし：書き込みデータの出力先は null でした。

no-output-specified: The output destination for data to be written to was null.

**KECX01418-E**

---

エンコーディング未サポート：未サポートのエンコーディングがありました：{0}  
unsupported-encoding: An unsupported encoding is encountered:{0}.

**KECX01419-E**

---

ノードシリアライゼーション不可：ノードはシリアライゼーションできませんでした。  
unable-to-serialize-node: The node could not be serialized.

**KECX01451-E**

---

メッセージキーに対応するエラーメッセージを、見つけることができません。  
The error message corresponding to the message key can not be found.

**KECX01452-E**

---

次のメッセージをフォーマットしている間に、内部エラーが起きました：  
An internal error occurred while formatting the following message:

**KECX01453-E**

---

入力指定がありません。  
no-input-specified

**KECX01600-E**

---

DOM パース中に再び parse メソッドが呼ばれました。  
parse may not be called while parsing.

**KECX01601-E**

---

要素スタックでアンダーフローが発生しました。  
Element stack underflow.

**KECX01701-E**

---

メッセージキーに対応するエラーメッセージが見つかりません。  
The error message corresponding to the message key can not be found.

**KECX01702-E**

---

次のメッセージをフォーマットする間に、内部エラーが発生しました。  
An internal error occurred while formatting the following message:

### **KECX01703-E**

---

include に失敗しました。fallback 要素が見つかりませんでした。

An 'include' failed, and no 'fallback' element was found.

### **KECX01704-E**

---

include 要素の子の一つを超える fallback 要素を含むことはできません。

The [children] of an 'include' element cannot contain more than one 'fallback' element.

### **KECX01705-E**

---

親が include でない fallback 要素がありました。

A 'fallback' element was found that did not have 'include' as the parent.

### **KECX01706-E**

---

名前空間 <http://www.w3.org/2001/Xinclude> からの要素は、fallback 以外は include 要素の子にはできません。しかし {0} がありました。

Elements from namespace 'http://www.w3.org/2001/XInclude', other than 'fallback', are not allowed to be children of 'include' elements. However, '{0}' was found.

### **KECX01707-E**

---

include 要素に href 属性がありません。

The 'href' attribute of an 'include' element is missing.

### **KECX01708-E**

---

再帰的なインクルード派生になっています。ドキュメント {0} はすでに処理されました。

Recursive include detected. Document '{0}' was already processed.

### **KECX01709-E**

---

include 要素上の parse 属性の無効な値です。{0}

Invalid value for 'parse' attribute on 'include' element: '{0}'.

### **KECX01710-E**

---

XML ファイルを解析するエラーです ( href='{0}' )

Error attempting to parse XML file (href='{0}').

**KECX01711-W**

---

インクルード処理が失敗し、フォールバックに戻ります。ファイルを XML として読み込むリソースエラーです (href='{0}')。理由は {1} です。

Include operation failed, reverting to fallback. Resource error reading file as XML (href='{0}').  
Reason: {1}

**KECX01712-E**

---

インクルード処理が失敗し、フォールバックに戻ります。ファイルをテキストとして読み込むリソースエラーです (href='{0}')。理由は {1} です。

Include operation failed, reverting to fallback. Resource error reading file as text (href='{0}').  
Reason: {1}

**KECX01713-E**

---

名前 "{0}" の複数の記法が使用されましたが、それが複写であるとはわかりませんでした。

Multiple notations were used which had the name '{0}', but which were not determined to be duplicates.

**KECX01714-E**

---

名前 "{0}" の複数の解析されていない実体が使用されましたが、それが複写であるとはわかりませんでした。

Multiple unparsed entities were used which had the name '{0}', but which were not determined to be duplicates.

**KECX01715-E**

---

NamespaceContext の型は XInclude の使用と互換性がありません。

XIncludeNamespaceSupport のインスタンスでなければいけません。

The type of the NamespaceContext is incompatible with using XInclude; it must be an instance of XIncludeNamespaceSupport

**KECX01716-E**

---

インクルードリソースのシステム ID を拡張することができませんでした。

Could not expand system id of included resource

**KECX01717-E**

---

名前空間 http://www.w3.org/2001/Xinclude からの要素は、include 以外は fallback 要素の子にできません。しかし、{0} がありました。

## 7. メッセージ

Elements from namespace 'http://www.w3.org/2001/XInclude', other than 'include', are not allowed to be children of 'fallback' elements. However, '{0}' was found.

### **KECX01718-E**

---

href 属性がない場合は、xpointer 属性が必要です。

xpointer attribute must be present when href attribute is absent.

### **KECX01719-E**

---

#x20 ~ #x7E の範囲外の文字は、include 要素の accept 属性の値に含めることができません。

Characters outside the range #x20 through #x7E are not allowed in the value of the 'accept' attribute of an 'include' element.

### **KECX01720-E**

---

#x20 ~ #x7E の範囲外の文字は、include 要素の accept-language 属性の値に含めることができません。

Characters outside the range #x20 through #x7E are not allowed in the value of the 'accept-language' attribute of an 'include' element.

### **KECX01721-E**

---

整形形式ドキュメントはルート要素が必要です。

A well-formed document requires a root element.

### **KECX01722-E**

---

整形形式ドキュメントには複数のルート要素を含めることはできません。

A well-formed document must not contain multiple root elements.

### **KECX01723-E**

---

最上位ソース infoset 内のドキュメント要素として表示される include 要素の代替物は文字を含むことができません。

The replacement of an 'include' element appearing as the document element in the top-level source infoset cannot contain characters.

### **KECX01724-E**

---

最上位ソース infoset 内のドキュメント要素として表示される include 要素の代替物は未拡張のエンティティ参照を含むことができません。

The replacement of an 'include' element appearing as the document element in the top-level source infoset cannot contain unexpanded entity references.



**KECX01725-E**

---

フラグメント識別子は使用できません。href 属性値 {0} は許可されていません。

Fragment identifiers must not be used. The 'href' attribute value '{0}' is not permitted.

**KECX01726-E**

---

href 属性値 '{0}' は構文的に不正です。エスケープ規則の適用後、値は構文的に正しい URI でも IRI でもありません。

href attribute value '{0}' is syntactically invalid. After applying escaping rules the value is neither a syntactically correct URI or IRI.

**KECX01727-E**

---

ソース infoset 内の位置を示す xpointer が指定されました。該当位置はプロセッサのストリーミング性のため、アクセスできません。

An xpointer was specified that points to a location in the source infoset. This location cannot be accessed due to the streaming nature of the processor.

**KECX01728-E**

---

XPointer 解決が失敗しました。

XPointer resolution unsuccessful.

**KECX01801-E**

---

不正な文字です。

Wrong character.

**KECX01802-E**

---

存在しないグループ番号です。

Invalid reference number.

**KECX01803-E**

---

¥ のあとに 1 文字必要です。

A character is required after ¥.

**KECX01804-E**

---

'?' をここに書くことはできません。もしかして '(?:' か '(?=' か '(?!' か '(?<' か '(?#' か '(?>' ですか？

'?' is not expected. '(?:' or '(?=' or '(?!' or '(?<' or '(?#' or '(?>'

---

**KECX01805-E**

---

'(?<=' か '(?<!' が必要です。

'(?<=' or '(?<!' is expected.

---

**KECX01806-E**

---

コメントが閉じていません。

A comment is not terminated.

---

**KECX01807-E**

---

')' が必要です。

') is expected.

---

**KECX01808-E**

---

オプションつきグループの中でパターンが終了しています。

Unexpected end of the pattern in a modifier group.

---

**KECX01809-E**

---

':' が必要です。

':' is expected.

---

**KECX01810-E**

---

条件つきグループの中でパターンが終了しています。

Unexpected end of the pattern in a conditional group.

---

**KECX01811-E**

---

条件パターンには、後方参照・アンカー・先読み・後読みだけを使用できます。

A back reference or an anchor or a lookahead or a lookbehind is expected in a conditional pattern.

---

**KECX01812-E**

---

条件つきグループの中に 3 個以上の選択を書くことはできません。

There are more than three choices in a conditional group.

---

**KECX01813-E**

---

¥c のあとにはコードポイントが U+0040 ~ U+005f の範囲の文字が必要です。

A character in U+0040-U+005f must follow ¥c.

**KECX01814-E**

---

1 文字のカテゴリ文字が '{' が必要です。

A '{' is required before a character category.

**KECX01815-E**

---

プロパティ名が '}' で閉じていません。

A property name is not closed by '}'.

**KECX01816-E**

---

メタ文字が不正なところにあります。

Unexpected meta character.

**KECX01817-E**

---

未知のプロパティ名です。

Unknown property.

**KECX01818-E**

---

POSIX 文字クラスは ']' で閉じる必要があります。

A POSIX character class must be closed by ']'.

**KECX01819-E**

---

文字クラス内でパターンが終了しています。

Unexpected end of the pattern in a character class.

**KECX01820-E**

---

未知の POSIX 文字クラス名です。

Unknown name for a POSIX character class.

**KECX01821-E**

---

ここに '.' を書くことはできません。

'.' is invalid here.

**KECX01822-E**

---

'}' が必要です。

'}' is expected.

#### **KECX01823-E**

---

文字クラス内では '[' を直接書くことはできません。'\[' と書いてください。

'[' is invalid in a character class. Write '\['.

#### **KECX01824-E**

---

文字クラス内では ']' を直接書くことはできません。'\]' と書いてください。

 is invalid in a character class. Write '\]'.

#### **KECX01825-E**

---

文字範囲指定に '-' を直接書くことはできません。'\-' と書いてください。

'-' is an invalid character range. Write '\-'.

#### **KECX01826-E**

---

'[' が必要です。

'[' is expected.

#### **KECX01827-E**

---

')' か '[' か '+' か '&' が必要です。

') or '[' or '+' or '&' is expected.

#### **KECX01828-E**

---

範囲終了コードポイントがスタートコードポイント未満です。

The range end code point is less than the start code point.

#### **KECX01829-E**

---

コードポイントの 16 進数表記中に間違った文字があります。

Invalid Unicode hex notation.

#### **KECX01830-E**

---

コードポイントの数値が大き過ぎます。

Overflow in a hex notation.

#### **KECX01831-E**

---

'\x{' は '}' で閉じなければいけません。

'\x{' must be closed by '}'.

**KECX01832-E**

---

Unicode のコードポイントとして不正です。

Invalid Unicode code point.

**KECX01833-E**

---

アンカーをここに書くことはできません。

An anchor must not be here.

**KECX01834-E**

---

この表現は現在のオプション設定では使用できません。

This expression is not supported in the current option setting.

**KECX01835-E**

---

指定が不正です。数字を指定してください。

Invalid quantifier. A digit is expected.

**KECX01836-E**

---

指定が不正です。不正な量数であるかまたは '}' がありません。

Invalid quantifier. Invalid quantity or a '}' is missing.

**KECX01837-E**

---

指定が不正です。数字または '}' を指定してください。

Invalid quantifier. A digit or '}' is expected.

**KECX01838-E**

---

指定が不正です。min の量数は max の量数以下でなければいけません。

Invalid quantifier. A min quantity must be <= a max quantity.

**KECX01839-E**

---

指定が不正です。量数値がオーバーフローしています。

Invalid quantifier. A quantity value overflow.

**KECX01853-E**

---

{0} は未知のオプションです。

Unknown option: {0}

---

**KECX01901-E**

---

メッセージキーに対応するエラーメッセージを、見つけることができません。

The error message corresponding to the message key can not be found.

---

**KECX01902-E**

---

次のメッセージをフォーマットしている間に、内部エラーが起きました：

An internal error occurred while formatting the following message:

---

**KECX01903-E**

---

指定されたスキーマ言語はサポートされていません。

The specified schema language is not supported.

---

**KECX01904-E**

---

プロパティ {1} を設定する前にプロパティ {0} を設定する必要があります。

Property '{0}' must be set before setting property '{1}'.

---

**KECX01905-E**

---

非 null のスキーマオブジェクトがすでに指定されている場合は、プロパティ {0} は設定できません。

Property '{0}' cannot be set when a non-null Schema object has already been specified.

---

**KECX01906-E**

---

フィーチャー {0} はサポートされていません。

Feature '{0}' is not supported.

---

**KECX01907-E**

---

フィーチャー {0} は認識されていません。

Feature '{0}' is not recognized.

---

**KECX01908-E**

---

フィーチャー {0} の true 状態はサポートされていません。

True state for feature '{0}' is not supported.

---

**KECX01909-E**

---

フィーチャー {0} に false 状態はサポートされていません。

False state for feature '{0}' is not supported.

---

**KECX01910-E**

---

フィーチャー {0} は読み込みのみ可能です。

Feature '{0}' is read only.

---

**KECX01911-E**

---

プロパティ {0} はサポートされていません。

Property '{0}' is not supported.

---

**KECX01912-E**

---

プロパティ {0} は認識されていません。

Property '{0}' is not recognized.

---

**KECX01913-E**

---

プロパティ {0} は読み込みのみ可能です。

Property '{0}' is read only.

---

**KECX01914-E**

---

構文解析中はプロパティ {0} はサポートされていません。

Property '{0}' is not supported while parsing.

---

**KECX01915-E**

---

DOM ノードプロパティは読めません。DOM ツリーがありません。

Cannot read DOM node property. No DOM tree exists.

---

**KECX01916-E**

---

プロパティ {0} に指定された値は {1} に変換できません。

The value specified for property '{0}' cannot be casted to {1}.

## 7.2.2 KECX02000 ~ KECX02999

### **KECX02001-E**

---

エラーです。式の中で '{' を持つことはできません。

Error: Can not have '{' within expression

### **KECX02002-E**

---

{0} に不正な属性が含まれています。{1}

{0} has an illegal attribute: {1}

### **KECX02003-E**

---

xsl:apply-imports で sourceNode が null です。

sourceNode is null in xsl:apply-imports!

### **KECX02004-E**

---

{0} を {1} に追加できません。

Can not add {0} to {1}

### **KECX02005-E**

---

handleApplyTemplatesInstruction で sourceNode が null です。

sourceNode is null in handleApplyTemplatesInstruction!

### **KECX02006-E**

---

{0} には name 属性が必要です。

{0} must have a name attribute.

### **KECX02007-E**

---

{0} という名前のテンプレートが見つかりませんでした。

Could not find template named: {0}

### **KECX02008-E**

---

xsl:call-template で名前 AVT を解決できませんでした。

Could not resolve name AVT in xsl:call-template.



**KECX02009-E**

---

{0} には属性 {1} が必要です。

{0} requires attribute: {1}

**KECX02010-E**

---

{0} には test 属性が必要です。

{0} must have a 'test' attribute.

**KECX02011-E**

---

level 属性に不正な値が指定されています。{0}

Bad value on level attribute: {0}

**KECX02012-E**

---

processing-instruction 名は xml にできません。

processing-instruction name can not be 'xml'

**KECX02013-E**

---

processing-instruction 名は有効な NCName でなければいけません。{0}

processing-instruction name must be a valid NCName: {0}

**KECX02014-E**

---

{0} にモードがある場合は、match 属性が必要です。

{0} must have a match attribute if it has a mode.

**KECX02015-E**

---

{0} は name 属性か、または match 属性を必要とします。

{0} requires either a name or a match attribute.

**KECX02016-E**

---

名前空間の接頭辞 {0} を解決できません。

Can not resolve namespace prefix: {0}

**KECX02017-E**

---

xml:space に不正な値があります。{0}

xml:space has an illegal value: {0}

### **KECX02018-E**

---

子ノードに所有者ドキュメントがありません。  
Child node does not have an owner document!

### **KECX02019-E**

---

ElemTemplateElement エラーです。{0}  
ElemTemplateElement error: {0}

### **KECX02020-E**

---

null である子を追加しようとしてしました。  
Trying to add a null child!

### **KECX02021-E**

---

{0} には select 属性が必要です。  
{0} requires a select attribute.

### **KECX02022-E**

---

xsl:when には test 属性が必要です。  
xsl:when must have a 'test' attribute.

### **KECX02023-E**

---

xsl:with-param には name 属性が必要です。  
xsl:with-param must have a 'name' attribute.

### **KECX02024-E**

---

コンテキストに所有者ドキュメントがありません。  
context does not have an owner document!

### **KECX02025-E**

---

XML TransformerFactory Liaison {0} を作成できませんでした。  
Could not create XML TransformerFactory Liaison: {0}

### **KECX02026-E**

---

XML Processor: 処理は成功しませんでした。  
XML Processor: Process was not successful.

**KECX02027-E**

---

XML Processor: は成功しませんでした。

XML Processor: was not successful.

**KECX02028-E**

---

エンコーディングはサポートされていません。{0}

Encoding not supported: {0}

**KECX02029-E**

---

TraceListener を作成できませんでした。{0}

Could not create TraceListener: {0}

**KECX02030-E**

---

xsl:key には name 属性が必要です。

xsl:key requires a 'name' attribute!

**KECX02031-E**

---

xsl:key には match 属性が必要です。

xsl:key requires a 'match' attribute!

**KECX02032-E**

---

xsl:key には use 属性が必要です。

xsl:key requires a 'use' attribute!

**KECX02033-E**

---

(StylesheetHandler) {0} には elements 属性が必要です。

(StylesheetHandler) {0} requires an 'elements' attribute!

**KECX02034-E**

---

(StylesheetHandler) {0} 属性の 'prefix' がありません。

(StylesheetHandler) {0} attribute 'prefix' is missing

**KECX02035-E**

---

スタイルシートの URL が不正です。{0}

Stylesheet URL is bad: {0}

---

**KECX02036-E**

---

スタイルシートファイルが見つかりませんでした。{0}

Stylesheet file was not found: {0}

---

**KECX02037-E**

---

スタイルシートファイルで入出力例外が発生しました。{0}

Had IO Exception with stylesheet file: {0}

---

**KECX02038-E**

---

(StylesheetHandler) {0} の href 属性が見つかりませんでした。

(StylesheetHandler) Could not find href attribute for {0}

---

**KECX02039-E**

---

(StylesheetHandler) {0} は直接的または間接的に自分自身をインクルードしています。

(StylesheetHandler) {0} is directly or indirectly including itself!

---

**KECX02040-E**

---

StylesheetHandler.processInclude エラーです。{0}

StylesheetHandler.processInclude error, {0}

---

**KECX02041-E**

---

(StylesheetHandler) {0} 属性の 'lang' がありません。

(StylesheetHandler) {0} attribute 'lang' is missing

---

**KECX02042-E**

---

(StylesheetHandler) {0} 要素の場所を間違えた可能性があります。コンテナ要素 'component' がありません。

(StylesheetHandler) misplaced {0} element?? Missing container element 'component'

---

**KECX02043-E**

---

Element , DocumentFragment , Document , または PrintWriter にだけ出力できます。

Can only output to an Element, DocumentFragment, Document, or PrintWriter.

---

**KECX02044-E**

---

StylesheetRoot.process エラーです。

StylesheetRoot.process error

---

**KECX02045-E**

---

UnImplNode エラーです。{0}

UnImplNode error: {0}

---

**KECX02046-E**

---

エラーです。xpath の選択式 (-select) が見つかりませんでした。

Error! Did not find xpath select expression (-select).

---

**KECX02047-E**

---

XSLProcessor をシリアル化できません。

Can not serialize an XSLProcessor!

---

**KECX02048-E**

---

スタイルシートの入力指定がされませんでした。

Stylesheet input was not specified!

---

**KECX02049-E**

---

スタイルシートでの処理に失敗しました。

Failed to process stylesheet!

---

**KECX02050-E**

---

{0} ドキュメントを構文解析できませんでした。

Could not parse {0} document!

---

**KECX02051-E**

---

フラグメント {0} が見つかりませんでした。

Could not find fragment: {0}

---

**KECX02052-E**

---

フラグメント識別子が指すノードが要素ではありませんでした。{0}

Node pointed to by fragment identifier was not an element: {0}

---

**KECX02053-E**

---

for-each には match または name のどちらかの属性が必要です。

## 7. メッセージ

for-each must have either a match or name attribute

---

### KECX02054-E

テンプレートには match または name のどちらかの属性が必要です。

templates must have either a match or name attribute

---

### KECX02055-E

ドキュメントのフラグメントにクローンがありません。

No clone of a document fragment!

---

### KECX02056-E

結果ツリーに項目を作成できません。{0}

Can not create item in result tree: {0}

---

### KECX02057-E

ソース XML の xml:space に不正な値があります。{0}

xml:space in the source XML has an illegal value: {0}

---

### KECX02058-E

{0} に xsl:key 宣言がありません。

There is no xsl:key declaration for {0}!

---

### KECX02059-E

エラーです。{0} の URL を作成できません。

Error! Cannot create url for: {0}

---

### KECX02060-E

xsl:functions はサポートされていません。

xsl:functions is unsupported

---

### KECX02061-E

XSLT TransformerFactory エラーです。

XSLT TransformerFactory Error

---

### KECX02062-E

(StylesheetHandler) {0} はスタイルシート内で許可されません。

(StylesheetHandler) {0} not allowed inside a stylesheet!

---

#### **KECX02063-E**

---

result-ns はサポートされていません。代わりに xsl:output を使用してください。

result-ns no longer supported! Use xsl:output instead.

---

#### **KECX02064-E**

---

default-space はサポートされていません。代わりに xsl:strip-space または xsl:preserve-space を使用してください。

default-space no longer supported! Use xsl:strip-space or xsl:preserve-space instead.

---

#### **KECX02065-E**

---

indent-result はサポートされていません。代わりに xsl:output を使用してください。

indent-result no longer supported! Use xsl:output instead.

---

#### **KECX02066-E**

---

(StylesheetHandler) {0} に不正な属性があります。{1}

(StylesheetHandler) {0} has an illegal attribute: {1}

---

#### **KECX02067-E**

---

未知の XSL 要素です。{0}

Unknown XSL element: {0}

---

#### **KECX02068-E**

---

(StylesheetHandler) xsl:sort は xsl:apply-templates または xsl:for-each とともにだけ使用できません。

(StylesheetHandler) xsl:sort can only be used with xsl:apply-templates or xsl:for-each.

---

#### **KECX02069-E**

---

(StylesheetHandler) xsl:when の場所が誤っています。

(StylesheetHandler) misplaced xsl:when!

---

#### **KECX02070-E**

---

(StylesheetHandler) xsl:when の親は xsl:choose ではありません。

(StylesheetHandler) xsl:when not parented by xsl:choose!

### **KECX02071-E**

---

(StylesheetHandler) xsl:otherwise の場所が誤っています。

(StylesheetHandler) misplaced xsl:otherwise!

### **KECX02072-E**

---

(StylesheetHandler) xsl:otherwise の親は xsl:choose ではありません。

(StylesheetHandler) xsl:otherwise not parented by xsl:choose!

### **KECX02073-E**

---

(StylesheetHandler) {0} はテンプレートの中に置くことを許可されません。

(StylesheetHandler) {0} is not allowed inside a template!

### **KECX02074-E**

---

(StylesheetHandler) {0} 拡張名前空間の接頭辞 {1} が未知です。

(StylesheetHandler) {0} extension namespace prefix {1} unknown

### **KECX02075-E**

---

(StylesheetHandler) スタイルシートの最初の要素でだけインポートが実行できます。

(StylesheetHandler) Imports can only occur as the first elements in the stylesheet!

### **KECX02076-E**

---

(StylesheetHandler) {0} は直接的または間接的に自身をインポートしています。

(StylesheetHandler) {0} is directly or indirectly importing itself!

### **KECX02077-E**

---

(StylesheetHandler) xml:space に不当な値があります。{0}

(StylesheetHandler) xml:space has an illegal value: {0}

### **KECX02078-E**

---

processStylesheet は成功しません。

processStylesheet not successful!

### **KECX02079-E**

---

SAX 例外です。

SAX Exception



**KECX02080-E**

---

未サポート機能です。  
Function not supported!

**KECX02081-E**

---

XSLT エラーです。  
XSLT Error

**KECX02082-E**

---

通貨記号は書式パターン文字列で許可されません。  
currency sign is not allowed in format pattern string

**KECX02083-E**

---

Document 関数は Stylesheet DOM でサポートされていません。  
Document function not supported in Stylesheet DOM!

**KECX02084-E**

---

non-Prefix リゾルバの接頭辞は解決できません。  
Can't resolve prefix of non-Prefix resolver!

**KECX02085-E**

---

Redirect 拡張: ファイル名を取得できませんでした。 - file 属性または select 属性が有効な文字列を戻す必要があります。  
Redirect extension: Could not get filename - file or select attribute must return valid string.

**KECX02086-E**

---

Redirect 拡張で FormatterListener を構築できません。  
Can not build FormatterListener in Redirect extension!

**KECX02087-E**

---

exclude-result-prefixes の接頭辞が有効ではありません。{0}  
Prefix in exclude-result-prefixes is not valid: {0}

**KECX02088-E**

---

指定された接頭辞の名前空間 URI がありません。

## 7. メッセージ

Missing namespace URI for specified prefix

---

### KECX02089-E

---

オプションの引数がありません。{0}

Missing argument for option: {0}

---

### KECX02090-E

---

無効なオプションです。{0}

Invalid option: {0}

---

### KECX02091-E

---

形式の誤った文字列です。{0}

Malformed format string: {0}

---

### KECX02092-E

---

xsl:stylesheet にはバージョン属性が必要です。

xsl:stylesheet requires a 'version' attribute!

---

### KECX02093-E

---

属性 {0} に不正な値があります。{1}

Attribute: {0} has an illegal value: {1}

---

### KECX02094-E

---

xsl:choose には xsl:when が必要です。

xsl:choose requires an xsl:when

---

### KECX02095-E

---

xsl:apply-imports は xsl:for-each で許可されません。

xsl:apply-imports not allowed in a xsl:for-each

---

### KECX02096-E

---

出力 DOM ノードに DTMLiaison を使用できません。代わりに  
com.cosminexus.jaxp.impl.xpath.DOM2Helper を渡します。

Cannot use a DTMLiaison for an output DOM node... pass a  
com.cosminexus.jaxp.impl.xpath.DOM2Helper instead!

**KECX02097-E**

---

入力 DOM ノードに DTMLiaison を使用できません。代わりに  
com.cosminexus.jaxp.impl.xpath.DOM2Helper を渡します。

Cannot use a DTMLiaison for a input DOM node... pass a  
com.cosminexus.jaxp.impl.xpath.DOM2Helper instead!

**KECX02098-E**

---

拡張要素の呼び出しに失敗しました。{0}

Call to extension element failed: {0}

**KECX02099-E**

---

接頭辞は名前空間で解決されなければいけません。{0}

Prefix must resolve to a namespace: {0}

**KECX02100-E**

---

無効な UTF-16 サロゲートが検出されました。{0}

Invalid UTF-16 surrogate detected: {0} ?

**KECX02101-E**

---

xsl:attribute-set{0} を自分自身で使用しているので、無限ループの原因になります。

xsl:attribute-set {0} used itself, which will cause an infinite loop.

**KECX02102-E**

---

非 XML Processor-DOM 入力と XML Processor-DOM 出力を混合できません。

Can not mix non XML Processor-DOM input with XML Processor-DOM output!

**KECX02103-E**

---

addTraceListenersToStylesheet ListenersException が多過ぎます。

addTraceListenersToStylesheet - TooManyListenersException

**KECX02104-E**

---

ElemTemplateElement.readObject に {0} があります。

In ElemTemplateElement.readObject: {0}

### **KECX02105-E**

---

次に示す名前のテンプレートが複数見つかりました。{0}

Found more than one template named: {0}

### **KECX02106-E**

---

無効な関数呼び出しです。recursive key() 呼び出しは許可されません。

Invalid function call: recursive key() calls are not allowed

### **KECX02107-E**

---

変数 {0} は直接的または間接的に自分自身を参照しています。

Variable {0} is directly or indirectly referencing itself!

### **KECX02108-E**

---

newTemplates の DOMSource に対する入力ノードは null にできません。

The input node can not be null for a DOMSource for newTemplates!

### **KECX02109-E**

---

オプション {0} に対するクラスファイルが見つかりません。

Class file not found for option {0}

### **KECX02110-E**

---

必要な要素が見つかりません。{0}

Required Element not found: {0}

### **KECX02111-E**

---

InputStream は null にできません。

InputStream cannot be null

### **KECX02112-E**

---

URI は null にできません。

URI cannot be null

### **KECX02113-E**

---

File は null にできません。

File cannot be null

**KECX02114-E**

---

InputSource は null にできません。

InputSource cannot be null

**KECX02116-E**

---

BSF Manager を初期化できませんでした。

Could not initialize BSF Manager

**KECX02117-E**

---

拡張をコンパイルできませんでした。

Could not compile extension

**KECX02118-E**

---

拡張 {0} を作成できませんでした。原因 {1}

Could not create extension: {0} because of: {1}

**KECX02119-E**

---

メソッド {0} のインスタンスメソッド呼び出しは、最初の引数に Object インスタンスを必要とします。

Instance method call to method {0} requires an Object instance as first argument

**KECX02120-E**

---

無効な要素名が指定されました。{0}

Invalid element name specified {0}

**KECX02121-E**

---

要素名メソッドは static でなければいけません。{0}

Element name method must be static {0}

**KECX02122-E**

---

拡張関数 {0}: {1} は未知です。

Extension function {0} : {1} is unknown

**KECX02123-E**

---

{0} のコンストラクタに最適な一致が複数個あります。

## 7. メッセージ

More than one best match for constructor for {0}

---

### KECX02124-E

メソッド {0} に最適な一致が複数個あります。

More than one best match for method {0}

---

### KECX02125-E

要素メソッド {0} に最適な一致が複数個あります。

More than one best match for element method {0}

---

### KECX02126-E

{0} を評価するのに無効なコンテキストが渡されました。

Invalid context passed to evaluate {0}

---

### KECX02127-E

プールがすでに存在します。

Pool already exists

---

### KECX02128-E

ドライバの Name が指定されていません。

No driver Name specified

---

### KECX02129-E

URL が指定されていません。

No URL specified

---

### KECX02130-E

プールサイズが 1 より小さいです。

Pool size is less than one!

---

### KECX02131-E

無効なドライバ名が指定されました。

Invalid driver name specified!

---

### KECX02132-E

スタイルシートのルートが見つかりませんでした。

Did not find the stylesheet root!

---

**KECX02133-E**

---

xml:space に不正な値です。

Illegal value for xml:space

---

**KECX02134-E**

---

processFromNode が失敗しました。

processFromNode failed

---

**KECX02135-E**

---

リソース {0} は次のものをロードできませんでした。 {1} {2} {3}

The resource [ {0} ] could not load: {1} {2} {3}

---

**KECX02136-E**

---

バッファサイズが 0 以下です。

Buffer size <=0

---

**KECX02137-E**

---

拡張を呼び出すときに未知のエラーが発生しました。

Unknown error when calling extension

---

**KECX02138-E**

---

接頭辞 {0} には対応する名前空間宣言がありません。

Prefix {0} does not have a corresponding namespace declaration

---

**KECX02139-E**

---

要素の内容は lang=javaclass {0} に許可されません。

Element content not allowed for lang=javaclass {0}

---

**KECX02140-E**

---

スタイルシートによって終了が指示されました。

Stylesheet directed termination

---

**KECX02143-E**

---

{0} をロードできませんでした (CLASSPATH を確認してください)。デフォルトだけを使用します。

Could not load {0} (check CLASSPATH), now using just the defaults

---

**KECX02144-E**

---

デフォルトのテンプレートを初期化できません。

Cannot initialize default templates

---

**KECX02145-E**

---

Result は null にはできません。

Result should not be null

---

**KECX02146-E**

---

Result を設定できませんでした。

Result could not be set

---

**KECX02147-E**

---

出力が指定されていません。

No output specified

---

**KECX02148-E**

---

タイプ {0} の Result に変換できません。

Can't transform to a Result of type {0}

---

**KECX02149-E**

---

タイプ {0} の Source に変換できません。

Can't transform a Source of type {0}

---

**KECX02150-E**

---

コンテンツハンドラが null です。

Null content handler

---

**KECX02151-E**

---

エラーハンドラが null です。



Null error handler

---

**KECX02152-E**

---

ContentHandler が設定されていないと構文解析を呼び出すことができません。

parse can not be called if the ContentHandler has not been set

---

**KECX02153-E**

---

フィルタ処理をする親がありません。

No parent for filter

---

**KECX02154-E**

---

スタイルシートが {0} , メディア = {1} に見つかりません。

No stylesheet found in: {0}, media= {1}

---

**KECX02155-E**

---

xml-stylesheet PI が {0} に見つかりませんでした。

No xml-stylesheet PI found in: {0}

---

**KECX02171-E**

---

未サポートです。 {0}

Not supported: {0}

---

**KECX02177-E**

---

プロパティ {0} の値は Boolean インスタンスでなくてはなりません。

Value for property {0} should be a Boolean instance

---

**KECX02195-E**

---

{0} にある外部スクリプトを入手できませんでした。

Could not get to external script at {0}

---

**KECX02196-E**

---

リソース {0} が見つかりませんでした。 {1}

The resource [{0}] could not be found. {1}

---

**KECX02197-E**

---

出力プロパティは認識されていません。 {0}

## 7. メッセージ

Output property not recognized: {0}

---

### KECX02203-E

ElemLiteralResult インスタンスの作成に失敗しました。

Failed creating ElemLiteralResult instance

---

### KECX02204-E

{0} の値には構文解析できる数字を指定してください。

Value for {0} should contain a parsable number

---

### KECX02205-E

{0} の値は yes または no と等しくなければいけません。

Value for {0} should equal yes or no

---

### KECX02206-E

{0} メソッドの呼び出しに失敗しました。

Failed calling {0} method

---

### KECX02207-E

ElemTemplateElement インスタンスの作成に失敗しました。

Failed creating ElemTemplateElement instance

---

### KECX02208-E

ドキュメントのこの部分で、文字列は許可されていません。

Characters are not allowed at this point in the document

---

### KECX02209-E

"{0}" 属性は {1} 要素で許可されていません。

"{0}" attribute is not allowed on the {1} element!

---

### KECX02211-E

{0} の間違った値 {1} です。

{0} bad value {1}

---

### KECX02212-E

{0} 属性値が見つかりません。

{0} attribute value not found

---

**KECX02213-E**

---

{0} 属性値は認識されていません。

{0} attribute value not recognized

---

**KECX02216-E**

---

名前空間の接頭辞を null の URI で生成しようとしています。

Attempting to generate a namespace prefix with a null URI

---

**KECX02217-E**

---

Long の最大値より大きい数をフォーマットしようとしています。

Attempting to format a number bigger than the largest Long integer

---

**KECX02218-E**

---

SAX1 ドライバークラス {0} が見つかりません。

Cannot find SAX1 driver class {0}

---

**KECX02219-E**

---

SAX1 ドライバークラス {0} は見つかりましたがロードできません。

SAX1 driver class {0} found but cannot be loaded

---

**KECX02220-E**

---

SAX1 ドライバークラス {0} はロードしましたが実証できません。

SAX1 driver class {0} loaded but cannot be instantiated

---

**KECX02221-E**

---

SAX1 ドライバークラス {0} は org.xml.sax.Parser をインプリメントしません。

SAX1 driver class {0} does not implement org.xml.sax.Parser

---

**KECX02222-E**

---

システムプロパティの org.xml.sax.parser は指定されませんでした。

System property org.xml.sax.parser not specified

---

**KECX02223-E**

---

パーサ引数は null であってはいけません。

## 7. メッセージ

Parser argument must not be null

### **KECX02224-E**

---

フィーチャーが {0} です。

Feature: {0}

### **KECX02225-E**

---

プロパティが {0} です。

Property: {0}

### **KECX02226-E**

---

Null エンティティリゾルバです。

Null entity resolver

### **KECX02227-E**

---

Null DTD ハンドラです。

Null DTD handler

### **KECX02228-E**

---

ドライバ名が指定されていません。

No Driver Name Specified!

### **KECX02229-E**

---

URL が指定されていません。

No URL Specified!

### **KECX02230-E**

---

プールサイズが 1 より小さいです。

Pool size is less than 1!

### **KECX02231-E**

---

無効なドライバ名が指定されました。

Invalid Driver Name Specified!

### **KECX02232-E**

---

ErrorListener が不正です。

ErrorListener

#### **KECX02233-E**

---

プログラマのエラーです。式は親に ElemTemplateElement を持っていません。

Programmer's error! The expression has no ElemTemplateElement parent!

#### **KECX02234-E**

---

プログラマの RedundentExprEliminator の中の指定です。{0}

Programmer's assertion in RedundentExprEliminator: {0}

#### **KECX02237-E**

---

スタイルシート内のこの位置には、{0} を指定できません。

{0} is not allowed in this position in the stylesheet!

#### **KECX02238-E**

---

スタイルシートのこの位置には空白以外のテキストを指定できません。

Non-whitespace text is not allowed in this position in the stylesheet!

#### **KECX02239-E**

---

CHAR 属性 {0} に不正な値 {1} が使用されました。CHAR タイプの属性は 1 文字でなければいけません。

Illegal value: {1} used for CHAR attribute: {0}. An attribute of type CHAR must be only 1 character!

#### **KECX02242-E**

---

QNAME 属性 {0} に不正な値 {1} が使用されました。

Illegal value: {1} used for QNAME attribute: {0}

#### **KECX02243-E**

---

ENUM 属性 {0} に不正な値 {1} が使用されました。有効な値は {2} です。

Illegal value: {1} used for ENUM attribute: {0}. Valid values are: {2}.

#### **KECX02244-E**

---

NMTOKEN 属性 {0} に不正な値 {1} が使用されました。

Illegal value: {1} used for NMTOKEN attribute: {0}

---

**KECX02245-E**

---

NCNAME 属性 {0} に不正な値 {1} が使用されました。

Illegal value: {1} used for NCNAME attribute: {0}

---

**KECX02246-E**

---

boolean 属性 {0} に不正な値 {1} が使用されました。

Illegal value: {1} used for boolean attribute: {0}

---

**KECX02247-E**

---

number 属性 {0} に不正な値 {1} が使用されました。

Illegal value: {1} used for number attribute: {0}

---

**KECX02248-E**

---

マッチング・パターンの {0} への引数はリテラルでなければいけません。

Argument to {0} in match pattern must be a literal.

---

**KECX02249-E**

---

グローバル変数宣言が重複しています。

Duplicate global variable declaration.

---

**KECX02250-E**

---

変数宣言が重複しています。

Duplicate variable declaration.

---

**KECX02251-E**

---

xsl:template には name または match 属性（あるいはその両方）が必要です。

xsl:template must have a name or match attribute (or both)

---

**KECX02252-E**

---

exclude-result-prefixes 内の接頭辞が無効です。{0}

Prefix in exclude-result-prefixes is not valid: {0}

---

**KECX02253-E**

---

{0} という名前の attribute-set が存在しません。

attribute-set named {0} does not exist

**KECX02254-E**

---

{0} という名前の関数が存在しません。

The function named {0} does not exist

**KECX02256-E**

---

param{0} の値は有効な Java オブジェクトでなければいけません。

The value of param {0} must be a valid Java Object

**KECX02257-E**

---

xsl:namespace-alias 要素の result-prefix 属性は値 #default を持ちますが、該当要素の範囲にデフォルト名前空間の宣言はありません。

The result-prefix attribute of an xsl:namespace-alias element has the value '#default', but there is no declaration of the default namespace in scope for the element

**KECX02258-E**

---

xsl:namespace-alias 要素の result-prefix 属性は値 {0} を持ちますが、該当要素の範囲に接頭辞 {0} の名前空間宣言はありません。

The result-prefix attribute of an xsl:namespace-alias element has the value '{0}', but there is no namespace declaration for the prefix '{0}' in scope for the element.

**KECX02259-E**

---

TransformerFactory.setFeature(String name, boolean value) では、フィーチャー名は null にできません。

The feature name cannot be null in TransformerFactory.setFeature(String name, boolean value).

**KECX02260-E**

---

TransformerFactory.getFeature(String name) では、フィーチャー名は null にできません。

The feature name cannot be null in TransformerFactory.getFeature(String name).

**KECX02261-E**

---

TransformerFactory ではフィーチャー {0} を設定できません。

Cannot set the feature '{0}' on this TransformerFactory.

**KECX02262-E**

---

secure processing フィーチャーが true に設定されている場合は、拡張子要素 {0} の使用は許可されません。

## 7. メッセージ

Use of the extension element '{0}' is not allowed when the secure processing feature is set to true.

### **KECX02263-E**

---

無効な名前空間 URI に対応する接頭辞を取得できません。

Cannot get the prefix for a null namespace uri.

### **KECX02264-E**

---

無効な接頭辞に対応する名前空間 URI を取得できません。

Cannot get the namespace uri for null prefix.

### **KECX02265-E**

---

フィーチャー名を null にできません。

The function name cannot be null.

### **KECX02266-E**

---

引数の数は負数にできません。

The arity cannot be negative.

### **KECX02288-E**

---

要素 {0} はコンテンツと select 属性の両方を持つことはできません。

The {0} element must not have both content and a select attribute.

### **KECX02289-E**

---

{0} の属性 name={1} はすでに同じ先行取り込みで使用されています。

{0} attribute name="{1}" is already used with same import precedence

### **KECX02304-E**

---

getAssociatedStylesheets が失敗しました。

getAssociatedStylesheets failed

### **KECX02309-E**

---

無効な ConnectionPool 名または JNDI Datasource パスです : {0}

Invalid ConnectionPool name or JNDI Datasource path: {0}

### **KECX02311-E**

---

SAXResult の handler は null の指定ができません。



handler can not be null for a SAXResult

---

**KECX02312-E**

---

resource bundles{0} のロードができません。

Could not load any resource bundles.{0}

---

**KECX02313-E**

---

resource bundles のロードができません。

Could not load any resource bundles.

---

**KECX02314-E**

---

ノード {0} のクローンができません。

Can't clone node: {0}

---

**KECX02320-E**

---

拡張仕様 {0} では {1}{2} は見つかりません。

For extension {0}, could not find {1} {2}

---

**KECX02321-W**

---

警告 : charToByteConverterClass を取得できません。

Warning: Could not get charToByteConverterClass!

---

**KECX02322-W**

---

警告 : エンコーディング {0} はサポートされていません。{1} を使用します。

Warning: encoding "{0}" not supported, using {1}

---

**KECX02323-W**

---

警告 : {0}

Warning: {0}

---

**KECX02324-W**

---

警告 : canConvert メソッド内で InvocationTargetException が発生しました。

Warning: InvocationTargetException in canConvert!

---

**KECX02325-W**

---

警告 : canConvert メソッド内で IllegalAccessException が発生しました。

## 7. メッセージ

Warning: IllegalAccessException in canConvert!

### KECX02326-E

---

エラー : xsl:fallback の親は拡張要素または未知の要素でなければいけません。

Error! parent of xsl:fallback must be an extension or unknown element!

### KECX02327-E

---

プログラマのエラーです。countMatchPattern は null であってはいけません。

Programmers error! countMatchPattern should never be null!

### KECX02328-W

---

QNAME のソートデータ種別のためのフックを記述する必要があります。

TODO: Need to write the hooks for QNAME sort data type

### KECX02329-W

---

警告 : -out {0} が {1} をスローしました。

# WARNING: -out {0} threw {1}

### KECX02330-W

---

警告 : -out の引数はファイル名でなければいけません。出力はコンソールに送出されました。

# WARNING: -out argument should have a filename, output sent to console

### KECX02331-W

---

appendEnvironmentReport メソッドが {0} をスローしました。

appendEnvironmentReport threw: {0}

### KECX02501-W

---

'}' を見つけましたが属性テンプレートがオープンしていません。

Found '}' but no attribute template open!

### KECX02502-W

---

警告 :count 属性は xsl:number の祖先と一致しません。Target={0}

Warning: count attribute does not match an ancestor in xsl:number! Target = {0}

### KECX02503-W

---

古い文法です。属性名 expr は select に変更されました。

Old syntax: The name of the 'expr' attribute has been changed to 'select'.

#### **KECX02504-W**

---

XML Processor は format-number 関数内でまだロケール名を処理しません。

XML Processor doesn't yet handle the locale name in the format-number function.

#### **KECX02505-W**

---

警告 : xml:lang={0} のロケールを見つけることができません。

Warning: Could not find locale for xml:lang={0}

#### **KECX02506-W**

---

{0} から URL を作成できません。

Can not make URL from: {0}

#### **KECX02507-W**

---

要求された doc{0} をロードできません。

Can not load requested doc: {0}

#### **KECX02508-W**

---

<sort xml:lang={0} の Collator を見つけられません。

Could not find Collator for <sort xml:lang={0}

#### **KECX02509-W**

---

古い文法です。関数命令は {0} の URL を使わなければいけません。

Old syntax: the functions instruction should use a url of {0}

#### **KECX02510-W**

---

エンコーディング {0} はサポートされていません。UTF-8 を使用します。

encoding not supported: {0}, using UTF-8

#### **KECX02511-W**

---

エンコーディング {0} はサポートされていません。Java{1} を使用します。

encoding not supported: {0}, using Java {1}

#### **KECX02512-W**

---

限定性の矛盾を見つけました。スタイルシート内で最後に使われている {0} が使われています。

## 7. メッセージ

Specificity conflicts found: {0} Last found in stylesheet will be used.

### **KECX02513-W**

---

パーサの実行と準備をしています。{0}

===== Parsing and preparing {0} =====

### **KECX02514-W**

---

Attr Template, {0}

Attr Template, {0}

### **KECX02515-W**

---

xsl:strip-space と xsl:preserve-space との衝突が起きています。

Match conflict between xsl:strip-space and xsl:preserve-space

### **KECX02516-W**

---

XML Processor は {0} 属性を取り扱っていません。

XML Processor does not yet handle the {0} attribute!

### **KECX02517-W**

---

10 進数形式の宣言が見つかりません。{0}

No declaration found for decimal format: {0}

### **KECX02518-E**

---

XSLT の名前空間がないか、または不正です。

Missing or incorrect XSLT Namespace.

### **KECX02519-W**

---

xsl:decimal-format 宣言で許されているデフォルト値は一つだけです。

Only one default xsl:decimal-format declaration is allowed.

### **KECX02520-W**

---

xsl:decimal-format 名はユニークでなければいけません。名称 {0} は多重定義されています。

xsl:decimal-format names must be unique. Name "{0}" has been duplicated.

### **KECX02521-W**

---

{0} が不正な属性 {1} を持っています。

{0} has an illegal attribute: {1}

#### **KECX02522-W**

---

名前空間接頭辞 {0} を解決できませんでした。ノードは無視されます。

Could not resolve namespace prefix: {0}. The node will be ignored.

#### **KECX02523-W**

---

xsl:stylesheet はバージョン属性が必要です。

xsl:stylesheet requires a 'version' attribute!

#### **KECX02524-W**

---

属性名が不正です。{0}

Illegal attribute name: {0}

#### **KECX02525-W**

---

属性 {0} で不正な値が使われています。{1}

Illegal value used for attribute {0}: {1}

#### **KECX02526-W**

---

ドキュメント関数の 2 番目の引数から得られた nodeset が空です。空の node-set を戻します。

Resulting nodeset from second argument of document function is empty. Return an empty node-set.

#### **KECX02527-W**

---

xsl:processing-instruction の name 属性の値は xml にできません。

The value of the name attribute of xsl:processing-instruction name must not be xml

#### **KECX02528-W**

---

xsl:processing-instruction の name 属性の値は、有効な NCName でなくてはなりません。{0}

The value of the name attribute of xsl:processing-instruction must be a valid NCName: {0}

#### **KECX02529-W**

---

子ノードのあと、または要素が現れる前に、属性 {0} を加えることはできません。属性は無視されます。

Cannot add attribute {0} after child nodes or before an element is produced. Attribute will be ignored.

---

**KECX02530-E**

---

修正してはいけない箇所に対して、オブジェクトを修正しようとしています。

An attempt is made to modify an object where modifications are not allowed.

---

**KECX02701-E**

---

createMessage へのパラメタが境界外でした。

Parameter to createMessage was out of bounds

---

**KECX02702-E**

---

messageFormat 呼び出し中に例外がスローされました。

Exception thrown during messageFormat call

---

**KECX02703-E**

---

Xalan-J コマンド行 Process クラス・オプション：

Xalan-J command line Process class options:

---

**KECX02704-W**

---

オプション {0} は XSLTC モードではサポートされていません。

The option {0} is not supported in XSLTC mode.

---

**KECX02705-W**

---

オプション {0} は -XSLTC と一緒にしか使用できません。

The option {0} can only be used with -XSLTC.

---

**KECX02706-E**

---

エラー：スタイルシートがないか入力 xml が指定されていません。使用法の説明については、オプションなしでこのコマンドを実行してください。

Error: No stylesheet or input xml is specified. Run this command without any option for usage instructions.

---

**KECX02707-E**

---

テンプレートのネストが深すぎます。ネスト = {0} , テンプレート {1} {2}

Template nesting too deep. nesting = {0}, template {1} {2}

---

**KECX02708-E**

---

呼び出すことができませんでした。

Could not prime the pump!

### **KECX02709-E**

---

{0} で ParseException が発生しました。

ParseException {0}

## 7.2.3 KECX03000 ~ KECX03999

### **KECX03001-E**

---

current() 関数はパターンのマッチングでは許可されていません。

The current() function is not allowed in a match pattern!

### **KECX03002-E**

---

current() 関数は引数を受け入れません。

The current() function does not accept arguments!

### **KECX03003-E**

---

document() 関数実装は com.cosminexus.jaxp.impl.transform.xslt.FuncDocument に置き換えられました。

document() function implementation has been replaced by  
com.cosminexus.jaxp.impl.transform.xslt.FuncDocument!

### **KECX03004-E**

---

コンテキストに所有者ドキュメントがありません。

context does not have an owner document!

### **KECX03005-E**

---

local-name() の引数が多過ぎます。

local-name() has too many arguments.

### **KECX03006-E**

---

namespace-uri() の引数が多過ぎます。

namespace-uri() has too many arguments.

### **KECX03007-E**

---

normalize-space() の引数が多過ぎます。

normalize-space() has too many arguments.

### **KECX03008-E**

---

number() の引数が多過ぎます。

number() has too many arguments.



**KECX03009-E**

---

name() の引数が多過ぎます。  
name() has too many arguments.

**KECX03010-E**

---

string() の引数が多過ぎます。  
string() has too many arguments.

**KECX03011-E**

---

string-length() の引数が多過ぎます。  
string-length() has too many arguments.

**KECX03012-E**

---

translate() 関数に三つの引数が指定されています。  
The translate() function takes three arguments!

**KECX03013-E**

---

unparsed- エンティティ -uri 関数の引数は一つです。  
The unparsed-entity-uri function should take one argument!

**KECX03014-E**

---

名前空間軸はまだ実装されていません。  
namespace axis not implemented yet!

**KECX03015-E**

---

未知の軸です。{0}  
unknown axis: {0}

**KECX03016-E**

---

未知の照合オペレーションです。  
unknown match operation!

**KECX03017-E**

---

processing-instruction() ノードテストの引数の長さが不正です。  
Arg length of processing-instruction() node test is incorrect!

#### **KECX03018-E**

---

{0} を数字に変換できません。

Can not convert {0} to a number

#### **KECX03019-E**

---

{0} を NodeList に変換できません。

Can not convert {0} to a NodeList!

#### **KECX03020-E**

---

{0} を NodeSetDTM に変換できません。

Can not convert {0} to a NodeSetDTM!

#### **KECX03021-E**

---

{0} を type#{1} に変換できません。

Can not convert {0} to a type#{1}

#### **KECX03022-E**

---

getMatchScore に予期される一致パターンです。

Expected match pattern in getMatchScore!

#### **KECX03023-E**

---

{0} という名前の変数を取得できませんでした。

Could not get variable named {0}

#### **KECX03024-E**

---

エラーです。未知のオプションコード {0}

ERROR! Unknown op code: {0}

#### **KECX03025-E**

---

余分な不正トークンです。{0}

Extra illegal tokens: {0}

#### **KECX03026-E**

---

リテラルの引用符が誤りです。二重引用符が必要です。

misquoted literal... expected double quote!

**KECX03027-E**

---

リテラルの引用符が誤りです。単一引用符が必要です。  
misquoted literal... expected single quote!

**KECX03028-E**

---

式が空です。  
Empty expression!

**KECX03029-E**

---

{0} が予期されていましたが、{1} が見つかりました。  
Expected {0}, but found: {1}

**KECX03030-E**

---

プログラマの指定が不正です。- {0}  
Programmer assertion is incorrect! - {0}

**KECX03031-E**

---

19990709 XPath ドラフトについて、boolean(...) 引数は任意ではありません。  
boolean(...) argument is no longer optional with 19990709 XPath draft.

**KECX03032-E**

---

',' が見つかりましたが、その前に引数がありません。  
Found ',' but no preceding argument!

**KECX03033-E**

---

',' が見つかりましたが、それに続く引数がありません。  
Found ',' but no following argument!

**KECX03034-E**

---

..[predicate] または .[predicate] は不正な構文です。代わりに self::node()[predicate] を使用してください。  
'..[predicate]' or '.[predicate]' is illegal syntax. Use 'self::node()[predicate]' instead.

**KECX03035-E**

---

不正な軸名です。{0}

## 7. メッセージ

illegal axis name: {0}

---

### KECX03036-E

未知のノードタイプです。{0}

Unknown nodetype: {0}

---

### KECX03037-E

リテラル ({0}) パターンは、引用符で囲む必要があります。

Pattern literal ({0}) needs to be quoted!

---

### KECX03038-E

{0} は数字に書式設定できませんでした。

{0} could not be formatted to a number!

---

### KECX03039-E

XML TransformerFactory Liaison を作成できませんでした。{0}

Could not create XML TransformerFactory Liaison: {0}

---

### KECX03040-E

エラーです。xpath 選択式 (-select) が見つかりませんでした。

Error! Did not find xpath select expression (-select).

---

### KECX03041-E

エラーです。OP\_LOCATIONPATH のあとに ENDOP が見つかりませんでした。

ERROR! Could not find ENDOP after OP\_LOCATIONPATH

---

### KECX03042-E

エラーが発生しました。

Error occurred!

---

### KECX03043-E

コンテキストの外で、または定義なしで VariableReference が変数に指定されました。Name = {0}

VariableReference given for variable out of context or without definition! Name = {0}

**KECX03044-E**

---

マッチングパターンには、child:: および attribute:: 軸だけが許可されます。許可されない軸 = {0}  
 Only child:: and attribute:: axes are allowed in match patterns! Offending axes = {0}

**KECX03045-E**

---

key() の引数の数が不正です。  
 key() has an incorrect number of arguments.

**KECX03046-E**

---

count 関数に使用できる引数は一つです。  
 The count function should take one argument!

**KECX03047-E**

---

関数 {0} が見つかりませんでした。  
 Could not find function: {0}

**KECX03048-E**

---

サポートされていないエンコーディング {0} です。  
 Unsupported encoding: {0}

**KECX03049-E**

---

getNextSibling の DTM に問題が発生しました。復元しようとしています。  
 Problem occurred in DTM in getNextSibling... trying to recover

**KECX03050-E**

---

プログラマエラーです。EmptyNodeList に書き込みできません。  
 Programmer error: EmptyNodeList can not be written to.

**KECX03051-E**

---

setDOMFactory は XPathContext でサポートされていません。  
 setDOMFactory is not supported by XPathContext!

**KECX03052-E**

---

接頭辞は名前空間で解決されなければいけません。{0}  
 Prefix must resolve to a namespace: {0}

---

**KECX03053-E**

---

構文解析 (InputSource ソース ) は XPathContext でサポートされていません。{0} をオープンできません。

parse (InputSource source) not supported in XPathContext! Can not open {0}

---

**KECX03057-E**

---

SAX API characters(char ch[]... は DTM で処理されません。

SAX API characters(char ch[]... not handled by the DTM!

---

**KECX03058-E**

---

ignorableWhitespace(char ch[]... は DTM で処理されません。

ignorableWhitespace(char ch[]... not handled by the DTM!

---

**KECX03059-E**

---

DTMLiaison はタイプ {0} のノードを処理できません。

DTMLiaison can not handle nodes of type {0}

---

**KECX03060-E**

---

DOM2Helper はタイプ {0} のノードを処理できません。

DOM2Helper can not handle nodes of type {0}

---

**KECX03061-E**

---

DOM2Helper.parse エラー : SystemID -{0} 行番号 - {1}

DOM2Helper.parse error: SystemID - {0} line - {1}

---

**KECX03062-E**

---

DOM2Helper.parse エラーです。

DOM2Helper.parse error

---

**KECX03065-E**

---

無効な UTF-16 サロゲートが検出されました : {0} ?

Invalid UTF-16 surrogate detected: {0} ?

---

**KECX03066-E**

---

入出力エラーです。

IO error

---

**KECX03067-E**

---

{0} の URL を作成できません。

Cannot create url for: {0}

---

**KECX03068-E**

---

XPath.readObject にあります。{0}

In XPath.readObject: {0}

---

**KECX03069-E**

---

関数トークンが見つかりません。

function token not found.

---

**KECX03071-E**

---

XPath タイプ {0} は処理できません。

Can not deal with XPath type: {0}

---

**KECX03072-E**

---

この NodeSet は可変ではありません。

This NodeSet is not mutable

---

**KECX03073-E**

---

この NodeSetDTM は可変ではありません。

This NodeSetDTM is not mutable

---

**KECX03074-E**

---

解決できない変数です。{0}

Variable not resolvable: {0}

---

**KECX03075-E**

---

エラーハンドラが null です。

Null error handler

---

**KECX03076-E**

---

プログラマの指定は、未知のオプションコードです。{0}

## 7. メッセージ

Programmer's assertion: unknown opcode: {0}

### **KECX03078-E**

---

rtf() は XRTreeFragSelectWrapper ではサポートされていません。

rtf() not supported by XRTreeFragSelectWrapper

### **KECX03079-E**

---

asNodeIterator() は XRTreeFragSelectWrapper ではサポートされていません。

asNodeIterator() not supported by XRTreeFragSelectWrapper

### **KECX03080-E**

---

XStringForChars に fsb() はサポートされていません。

fsb() not supported for XStringForChars

### **KECX03081-E**

---

{0} という名前の変数が見つかりませんでした。

Could not find variable with the name of {0}

### **KECX03082-E**

---

XStringForChars は引数に文字列を使用できません。

XStringForChars can not take a string for an argument

### **KECX03083-E**

---

FastStringBuffer 引数は null にできません。

The FastStringBuffer argument can not be null

### **KECX03085-E**

---

確定する前に変数がアクセスされました。

Variable accessed before it is bound!

### **KECX03086-E**

---

XStringForFSB は引数の string を受け取ることができません。

XStringForFSB can not take a string for an argument!

### **KECX03087-E**

---

エラーです。ウォーカのルートを null にします。



!!!! Error! Setting the root of a walker to null!!!

#### **KECX03088-E**

---

この NodeSetDTM は、直前のノードを繰り返すことができません。

This NodeSetDTM can not iterate to a previous node!

#### **KECX03089-E**

---

この NodeSet は、直前のノードを繰り返すことができません。

This NodeSet can not iterate to a previous node!

#### **KECX03090-E**

---

この NodeSetDTM は、索引づけまたはカウント機能を実行できません。

This NodeSetDTM can not do indexing or counting functions!

#### **KECX03091-E**

---

この NodeSet は、索引づけまたはカウント機能を実行できません。

This NodeSet can not do indexing or counting functions!

#### **KECX03092-E**

---

nextNode を呼び出したあとに、setShouldCacheNodes を呼び出すことはできません。

Can not call setShouldCacheNodes after nextNode has been called!

#### **KECX03093-E**

---

{0} に許可される引数は {1} 個だけです。

{0} only allows {1} arguments

#### **KECX03094-E**

---

getNextStepPos でプログラマが表明した、未知の stepType{0} です。

Programmer's assertion in getNextStepPos: unknown stepType: {0}

#### **KECX03095-E**

---

相対ロケーションパスは '/' または '//' トークンの次に必要です。

A relative location path was expected following the '/' or '/' token.

#### **KECX03096-E**

---

ロケーションパスが必要でしたが、次のトークンが検出されました。{0}

## 7. メッセージ

A location path was expected, but the following token was encountered: {0}

### **KECX03097-E**

---

ロケーションステップは '/' または '/' トークンの次に必要です。

A location step was expected following the '/' or '/' token.

### **KECX03098-E**

---

NCName:\* または QName のどちらかと一致するノードテストが必要です。

A node test that matches either NCName:\* or QName was expected.

### **KECX03099-E**

---

ステップパターンが必要でしたが, '/' が検出されました。

A step pattern was expected, but '/' was encountered.

### **KECX03100-E**

---

相対的なパスパターンが必要です。

A relative path pattern was expected.

### **KECX03103-E**

---

XPath 式 {0} の XPathResult に, ブールに変換できない {1} の XPathResultType があります。

The XPathResult of XPath expression '{0}' has an XPathResultType of {1} which cannot be converted to a boolean.

### **KECX03104-E**

---

XPath 式 {0} の XPathResult に, 単一ノードに変換できない {1} の XPathResultType があります。getSingleNodeValue メソッドは ANY\_UNORDERED\_NODE\_TYPE 型および FIRST\_ORDERED\_NODE\_TYPE 型のみに適用します。

The XPathResult of XPath expression '{0}' has an XPathResultType of {1} which cannot be converted to a single node. The method getSingleNodeValue applies only to types ANY\_UNORDERED\_NODE\_TYPE and FIRST\_ORDERED\_NODE\_TYPE.

### **KECX03105-E**

---

XPathResultType が {1} のため, getSnapshotLength メソッドは XPath 式 {0} の XPathResult に呼び出すことができません。このメソッドは, UNORDERED\_NODE\_SNAPSHOT\_TYPE 型および ORDERED\_NODE\_SNAPSHOT\_TYPE 型のみに適用します。

The method `getSnapshotLength` cannot be called on the `XPathResult` of XPath expression '{0}' because its `XPathResultType` is {1}. This method applies only to types `UNORDERED_NODE_SNAPSHOT_TYPE` and `ORDERED_NODE_SNAPSHOT_TYPE`.

---

#### KECX03106-E

`XPathResultType` が {1} のため、`iterateNext` メソッドは XPath 式 {0} の `XPathResult` に呼び出すことができません。このメソッドは、`UNORDERED_NODE_ITERATOR_TYPE` 型および `ORDERED_NODE_ITERATOR_TYPE` 型のみに適用します。

The method `iterateNext` cannot be called on the `XPathResult` of XPath expression '{0}' because its `XPathResultType` is {1}. This method applies only to types `UNORDERED_NODE_ITERATOR_TYPE` and `ORDERED_NODE_ITERATOR_TYPE`.

---

#### KECX03107-E

結果が返されたあとに、ドキュメントが変更されました。イテレータは無効です。

Document mutated since result was returned. Iterator is invalid.

---

#### KECX03108-E

無効の XPath タイプ引数です。{0}

Invalid XPath type argument: {0}

---

#### KECX03109-E

空の XPath 結果オブジェクトです。

Empty XPath result object

---

#### KECX03110-E

XPath 式 {0} の `XPathResult` は、{2} の指定された `XPathResultType` に強制できない {1} の `XPathResultType` を持ちます。

The `XPathResult` of XPath expression '{0}' has an `XPathResultType` of {1} which cannot be coerced into the specified `XPathResultType` of {2}.

---

#### KECX03111-E

接頭辞なしのリゾルバでは、接頭辞を解決できません。

Unable to resolve prefix with null prefix resolver.

---

#### KECX03112-E

XPath 式 {0} の `XPathResult` は、文字列に変換できない {1} の `XPathResultType` を持ちます。

## 7. メッセージ

The XPathResult of XPath expression '{0}' has an XPathResultType of {1} which cannot be converted to a string.

### **KECX03113-E**

---

XPathResultType が {1} のため、snapshotItem メソッドは XPath 式 {0} の XPathResult に呼び出すことができません。このメソッドは、UNORDERED\_NODE\_SNAPSHOT\_TYPE 型および ORDERED\_NODE\_SNAPSHOT\_TYPE 型のみに適用します。

The method snapshotItem cannot be called on the XPathResult of XPath expression '{0}' because its XPathResultType is {1}. This method applies only to types UNORDERED\_NODE\_SNAPSHOT\_TYPE and ORDERED\_NODE\_SNAPSHOT\_TYPE.

### **KECX03114-E**

---

コンテキストノードは、この XPathEvaluator に結び付けられるドキュメントに属しません。

Context node does not belong to the document that is bound to this XPathEvaluator.

### **KECX03115-E**

---

コンテキストノードタイプはサポートされていません。

The context node type is not supported.

### **KECX03116-E**

---

XPath に未知のエラーが発生しました。

Unknown error in XPath.

### **KECX03117-E**

---

detach() は XRTreeFragSelectWrapper ではサポートされていません。

detach() not supported by XRTreeFragSelectWrapper

### **KECX03118-E**

---

num() は XRTreeFragSelectWrapper ではサポートされていません。

num() not supported by XRTreeFragSelectWrapper

### **KECX03119-E**

---

xstr() は XRTreeFragSelectWrapper ではサポートされていません。

xstr() not supported by XRTreeFragSelectWrapper

**KECX03120-E**

---

str() は XRTreeFragSelectWrapper ではサポートされていません。

str() not supported by XRTreeFragSelectWrapper

**KECX03121-E**

---

ロケーションパスが期待されていたが、XPath 式の終わりが発見されました。

A location path was expected, but the end of the XPath expression was found instead.

**KECX03122-E**

---

XPath 式 {0} の XPathResult に、数字に変換できない {1} の XPathResultType があります。

The XPathResult of XPath expression '{0}' has an XPathResultType of {1} which cannot be converted to a number

**KECX03123-E**

---

拡張子関数：XMLConstants.FEATURE\_SECURE\_PROCESSING フィーチャーが true に設定されている場合、{0} を起動できません。

Extension function: '{0}' can not be invoked when the XMLConstants.FEATURE\_SECURE\_PROCESSING feature is set to true.

**KECX03124-E**

---

変数 {0} の resolveVariable が null を返しました。

resolveVariable for variable {0} returning null

**KECX03125-E**

---

サポートされていないリターン型：{0}

Unsupported Return Type : {0}

**KECX03126-E**

---

ソース型またはリターン型、または両方は null にできません。

Source and/or Return Type can not be null

**KECX03127-E**

---

ソース型またはリターン型、または両方は null にできません。

Source and/or Return Type can not be null

#### **KECX03128-E**

---

{0} 引数は null にできません。

{0} argument can not be null

#### **KECX03129-E**

---

{0}#isObjectModelSupported ( String objectModel ) は objectModel == null と同時に呼び出せません。

{0}#isObjectModelSupported( String objectModel ) cannot be called with objectModel == null

#### **KECX03130-E**

---

{0}#isObjectModelSupported ( String objectModel ) は objectModel == ¥"¥" と同時に呼び出せません。

{0}#isObjectModelSupported( String objectModel ) cannot be called with objectModel == ¥"¥"

#### **KECX03131-E**

---

null 名でフィーチャーを設定しようとした : {0}#setFeature( null, {1})

Trying to set a feature with a null name: {0}#setFeature( null, {1})

#### **KECX03132-E**

---

未知のフィーチャー ¥"{0}¥" を設定しようとした : {1}#setFeature({0},{2})

Trying to set the unknown feature ¥"{0}¥":{1}#setFeature({0},{2})

#### **KECX03133-E**

---

null 名でフィーチャーを取得しようとした : {0}#getFeature(null)

Trying to get a feature with a null name: {0}#getFeature(null)

#### **KECX03134-E**

---

未知のフィーチャー ¥"{0}¥" を取得しようとした : {1}#getFeature({0})

Trying to get the unknown feature ¥"{0}¥":{1}#getFeature({0})

#### **KECX03135-E**

---

null の XPathFunctionResolver を設定しようとした : {0}#setXPathFunctionResolver(null )

Attempting to set a null XPathFunctionResolver:{0}#setXPathFunctionResolver(null)

#### **KECX03136-E**

---

null の XPathVariableResolver を設定しようとした : {0}#setXPathVariableResolver(null)

Attempting to set a null XPathVariableResolver:{0}#setXPathVariableResolver(null)

### **KECX03306-E**

---

この NodeSetDTM は索引づけまたはカウント機能を実行できません。

This NodeSetDTM can not do indexing or counting functions!

### **KECX03307-E**

---

プログラマエラーです。putDocumentInCache によって、doc の再パーシングが見つかりました。{0}

Programmer's Error! putDocumentInCache found reparsing of doc: {0}

### **KECX03314-E**

---

どんな resource bundles もロードできません。

Could not load any resource bundles.

### **KECX03315-E**

---

エラーです。Function.execute は呼ばれてはいけません。

Error! Function.execute should not be called!

### **KECX03316-W**

---

警告：スタックが空の場合の popContextNodeList です。

Warning: popContextNodeList when stack is empty!

### **KECX03501-W**

---

format-number 関数でロケール名はまだ処理されていません。

locale name in the format-number function not yet handled!

### **KECX03502-W**

---

XSL プロパティ {0} は未サポートです。

XSL Property not supported: {0}

### **KECX03503-W**

---

現在、プロパティ {1} の名前空間 {0} で何も実行されていません。

Do not currently do anything with namespace {0} in property: {1}

#### **KECX03504-W**

---

XSL システムプロパティ {0} にアクセスしたときに、SecurityException が発生しました。

SecurityException when trying to access XSL system property: {0}

#### **KECX03505-W**

---

旧構文：quo(...) は XPath では定義されていません。

Old syntax: quo(...) is no longer defined in XPath.

#### **KECX03506-W**

---

nodeTest をインプリメントするには XPath に派生オブジェクトが必要です。

XPath needs a derived object to implement nodeTest!

#### **KECX03507-W**

---

関数トークンが見つかりません。

function token not found.

#### **KECX03508-W**

---

関数 {0} が見つかりませんでした。

Could not find function: {0}

#### **KECX03509-W**

---

URL を {0} から作成できません。

Can not make URL from: {0}

#### **KECX03510-W**

---

DTM パーサで -E オプションはサポートされていません。

-E option not supported for DTM parser

#### **KECX03511-W**

---

コンテキスト外、または定義なしで、変数に VariableReference が指定されました。名前 = {0}

VariableReference given for variable out of context or without definition! Name = {0}

#### **KECX03512-W**

---

サポートされないエンコード：{0}

Unsupported encoding: {0}



**KECX03701-E**

---

createMessage へのパラメタが境界外でした。

Parameter to createMessage was out of bounds

**KECX03702-E**

---

messageFormat 呼び出し中に例外がスローされました。

Exception thrown during messageFormat call

**KECX03901-E**

---

述語またはロケーションステップは NodeSet だけに適用できます。

A predicate or a location step is applicable to only a NodeSet.

**KECX03902-E**

---

変数リゾルバが設定されていません。

A XPathVariableResolver is not set.

**KECX03903-E**

---

関数リゾルバが設定されていません。

A XPathFunctionResolver is not set.

## 7.2.4 KECX04000 ~ KECX04999

### **KECX04012-E**

---

Configuration エラーです。

Configuration Error

### **KECX04014-E**

---

コーディングエラーです。インデックスを記述した DTM は mark/rewind しないでください。

Coding error; Don't try to mark/rewind an indexed DTM

### **KECX04016-E**

---

null/empty スキーマでは URI を構成できません。

Cannot construct URI with null/empty scheme!

### **KECX04017-E**

---

null/empty の scheme-specific では URI を構成できません。

Cannot construct URI with null/empty scheme-specific part!

### **KECX04018-E**

---

{0} が無効です。port は数字だけで指定してください。

{0} is invalid. Port should only contain digits!

### **KECX04019-E**

---

null 文字列では path の初期化はできません。

Cannot initialize path from null string!

### **KECX04020-E**

---

Query string に無効なエスケープシーケンスが含まれています。

Query string contains invalid escape sequence!

### **KECX04021-E**

---

Query string に無効な文字が含まれています。{0}

Query string contains invalid character:{0}

**KECX04022-E**

---

Fragment に無効なエスケープシーケンスが含まれています。{0}

Fragment contains invalid escape sequence!

**KECX04023-E**

---

Fragment に無効な文字が含まれています。{0}

Fragment contains invalid character:{0}

**KECX04024-E**

---

host 指定がない場合 , Userinfo は設定できません。

Userinfo cannot be set when host is null!

**KECX04025-E**

---

Userinfo に無効なエスケープシーケンスが含まれています。{0}

Userinfo contains invalid escape sequence!

**KECX04026-E**

---

Userinfo に無効な文字が含まれています。{0}

Userinfo contains invalid character:{0}

**KECX04027-E**

---

Query string は一般的な URI のセットにだけ使えます。

Query string can only be set for a generic URI!

**KECX04028-E**

---

Query string は path が null の場合には使えません。

Query string cannot be set when path is null!

**KECX04029-E**

---

Query string に無効な文字を含みます。

Query string contains invalid character!

**KECX04030-E**

---

どんな resource bundle もロードできません。

Could not load any resource bundles.

### **KECX04038-W**

---

警告 : ChunkedIntArray({0}) がスロット当たり {1} ワードを消費しています。

\*\*\*\*\*WARNING: ChunkedIntArray({0}) wasting {1} words per slot

### **KECX04039-W**

---

絶対化できない URL です。{0}

Can not absolutize URL: {0}

### **KECX04040-E**

---

ソース位置のコーディングエラーです。{0} != {1}

CODING ERROR in Source Location: {0} != {1}

### **KECX04046-E**

---

DOM エラーです。クラス {0}

DOM ERROR! class: {0}

### **KECX04047-E**

---

どんな resource bundle もロードできません。{0}

Could not load any resource bundles.{0}

### **KECX04101-E**

---

機能はサポートされていません。

Function not supported!

### **KECX04102-E**

---

cause を上書きできません。

Cannot overwrite cause

### **KECX04103-E**

---

デフォルトインプリメンテーションが見つかりません。

No default implementation found

### **KECX04104-E**

---

現在 ChunkedIntArray({0}) はサポートされていません。

ChunkedIntArray({0}) not currently supported

**KECX04105-E**

---

オフセットがスロットより大きいです。

Offset bigger than slot

**KECX04106-E**

---

Coroutine を使用できません。id={0}

Coroutine not available, id={0}

**KECX04107-E**

---

CoroutineManager が co\_exit() 要求を受け取りました。

CoroutineManager received co\_exit() request

**KECX04108-E**

---

co\_joinCoroutineSet() が失敗しました。

co\_joinCoroutineSet() failed

**KECX04109-E**

---

Coroutine パラメタエラーです。({0})

Coroutine parameter error ({0})

**KECX04110-E**

---

予想外 : パーサ doTerminate が {0} を応答しています。

UNEXPECTED: Parser doTerminate answers {0}

**KECX04111-E**

---

parse は構文解析中に呼び出してはいけません。

parse may not be called while parsing

**KECX04112-E**

---

エラーです。軸 {0} の型つきイテレータはインプリメントされていません。

Error: typed iterator for axis {0} not implemented

**KECX04113-E**

---

エラーです。軸 {0} のイテレータはインプリメントされていません。

Error: iterator for axis {0} not implemented

---

**KECX04114-E**

---

イテレータの複製はサポートされていません。

Iterator clone not supported

---

**KECX04115-E**

---

未知の軸トラバースタイプです。{0}

Unknown axis traversal type: {0}

---

**KECX04116-E**

---

軸トラバーサはサポートされていません。{0}

Axis traverser not supported: {0}

---

**KECX04117-E**

---

使用できる DTM ID はこれ以上ありません。

No more DTM IDs are available

---

**KECX04118-E**

---

サポートされていません。{0}

Not supported: {0}

---

**KECX04119-E**

---

getDTMHandleFromNode のノードは非 null でなければいけません。

Node must be non-null for getDTMHandleFromNode

---

**KECX04120-E**

---

ノードをハンドルに解決できませんでした。

Could not resolve the node to a handle

---

**KECX04121-E**

---

startParse は構文解析中に呼び出してはいけません。

startParse may not be called while parsing

---

**KECX04122-E**

---

startParse には null 以外の SAXParser が必要です。

startParse needs a non-null SAXParser

**KECX04123-E**

---

パーサを次で初期化できませんでした。

could not initialize parser with

**KECX04124-E**

---

プールの新規インスタンスを作成中に例外が発生しました。

exception creating new instance for pool

**KECX04125-E**

---

パスに無効なエスケープシーケンスが含まれています。

Path contains invalid escape sequence

**KECX04126-E**

---

スキームが必要です。

Scheme is required!

**KECX04127-E**

---

スキームは URI{0} で見つかりません。

No scheme found in URI: {0}

**KECX04128-E**

---

スキームは URI で見つかりません。

No scheme found in URI

**KECX04129-E**

---

パスに無効文字 {0} が含まれています。

Path contains invalid character: {0}

**KECX04130-E**

---

null 文字列からはスキームを設定できません。

Cannot set scheme from null string

**KECX04131-E**

---

スキームは一致していません。

The scheme is not conformant.

---

**KECX04132-E**

---

ホストは整形形式のアドレスではありません。

Host is not a well formed address

---

**KECX04133-E**

---

ホスト指定がない場合ポートを設定できません。

Port cannot be set when host is null

---

**KECX04134-E**

---

無効なポート番号です。

Invalid port number

---

**KECX04135-E**

---

一般的な URI のフラグメントだけ設定できます。

Fragment can only be set for a generic URI

---

**KECX04136-E**

---

パスが null であるとフラグメントを設定できません。

Fragment cannot be set when path is null

---

**KECX04137-E**

---

フラグメントに無効文字が含まれています。

Fragment contains invalid character

---

**KECX04138-E**

---

パーサはすでに使用中です。

Parser is already in use

---

**KECX04139-E**

---

構文解析中に {0}{1} を変更できません。

Cannot change {0} {1} while parsing

---

**KECX04140-E**

---

自己原因は許可されません。

Self-causation not permitted



**KECX04141-E**

---

ホストが指定されていない場合は Userinfo を指定してはいけません。

Userinfo may not be specified if host is not specified

**KECX04142-E**

---

ホストが指定されていない場合はポートを指定してはいけません。

Port may not be specified if host is not specified

**KECX04143-E**

---

Query string はパスおよび Query string 内に指定できません。

Query string cannot be specified in path and query string

**KECX04144-E**

---

フラグメントはパスとフラグメントの両方に指定できません。

Fragment cannot be specified in both the path and fragment

**KECX04145-E**

---

URI は空のパラメタを使用して初期化できません。

Cannot initialize URI with empty parameters

**KECX04146-E**

---

メソッドは未サポートです。

Method not yet supported

**KECX04147-E**

---

現在 IncrementalSAXSource\_Filter は再始動できません。

IncrementalSAXSource\_Filter not currently restartable

**KECX04148-E**

---

XMLReader が startParse 要求の前ではありません。

XMLReader not before startParse request

**KECX04149-E**

---

軸トラバーサはサポートされていません。{0}

Axis traverser not supported: {0}

---

**KECX04150-E**

---

ListingErrorHandler が null PrintWriter で作成されました。

ListingErrorHandler created with null PrintWriter!

---

**KECX04151-E**

---

SystemId は未知です。

SystemId Unknown

---

**KECX04152-E**

---

エラーのロケーションは未知です。

Location of error unknown

---

**KECX04153-E**

---

接頭辞は名前空間で解決されなければいけません。{0}

Prefix must resolve to a namespace: {0}

---

**KECX04154-E**

---

createDocument() は XPathContext 内で未サポートです。

createDocument() not supported in XPathContext!

---

**KECX04155-E**

---

属性の子に所有者文書がありません。

Attribute child does not have an owner document!

---

**KECX04156-E**

---

属性の子に所有者文書エレメントがありません。

Attribute child does not have an owner document element!

---

**KECX04157-E**

---

警告 : 文書エレメントの前にテキストを出力できません。無視しています ...

Warning: can't output text before document element! Ignoring...

---

**KECX04158-E**

---

DOM では複数のルートを持てません。

Can't have more than one root on a DOM!

**KECX04159-E**

---

引数 `localName` が null です。

Argument 'localName' is null

**KECX04160-E**

---

QNAME 内のローカル名は有効な NCName でなければいけません。

Localname in QNAME should be a valid NCName

**KECX04161-E**

---

QNAME 内の接頭辞は有効な NCName でなければいけません。

Prefix in QNAME should be a valid NCName

**KECX04162-E**

---

`createMessage` へのパラメタが境界外でした。

Parameter to `createMessage` was out of bounds

**KECX04163-E**

---

`messageFormat` 呼び出し中に例外がスローされました。

Exception thrown during `messageFormat` call

**KECX04164-E**

---

シリアライザークラス {0} は `org.xml.sax.ContentHandler` をインプリメントしません。

The serializer class '{0}' does not implement `org.xml.sax.ContentHandler`.

**KECX04165-E**

---

リソース {0} は見つかりませんでした。 {1}

The resource [ {0} ] could not be found. {1}

**KECX04166-E**

---

リソース {0} は次のものをロードできませんでした。 {1} {2} {3}

The resource [ {0} ] could not load: {1} {2} {3}

**KECX04167-E**

---

バッファサイズが 0 以下です。

Buffer size <=0

---

**KECX04168-E**

---

無効な UTF-16 サロゲートが検出されました。{0}

Invalid UTF-16 surrogate detected: {0} ?

---

**KECX04169-E**

---

入出力エラーです。

IO error

---

**KECX04170-E**

---

子ノードのあと、またはエレメントが現れる前に、属性 {0} を加えることはできません。属性は無視されます。

Cannot add attribute {0} after child nodes or before an element is produced. Attribute will be ignored.

---

**KECX04171-E**

---

接頭辞 {0} の名前空間が宣言されていません。

Namespace for prefix '{0}' has not been declared.

---

**KECX04172-E**

---

属性 {0} がエレメントの外側です。

Attribute '{0}' outside of element.

---

**KECX04173-E**

---

名前空間宣言 {0}={1} がエレメントの外側です。

Namespace declaration '{0}'='{1}' outside of element.

---

**KECX04174-E**

---

{0} をロードできませんでした (CLASSPATH を確認してください)。現在は単にデフォルトを使用しています。

Could not load '{0}' (check CLASSPATH), now using just the defaults

---

**KECX04175-E**

---

出力メソッド {1} のプロパティファイル {0} をロードできませんでした (CLASSPATH を確認してください)。

Could not load the property file '{0}' for output method '{1}' (check CLASSPATH) }

**KECX04176-E**

---

整数値 {0} で出力しようとしている文字は、{1} で指定された出力エンコーディングで表現できません。

Attempt to output character of integral value {0} that is not represented in specified output encoding of {1}.

**KECX04177-E**

---

無効なポート番号です。

Invalid port number

**KECX04178-E**

---

ホストが null の場合はポートを設定できません。

Port cannot be set when host is null

**KECX04179-E**

---

ホストが整形形式アドレスではありません。

Host is not a well formed address

**KECX04180-E**

---

スキームが適合しません。

The scheme is not conformant.

**KECX04181-E**

---

null 文字列からスキームを設定できません。

Cannot set scheme from null string

**KECX04182-E**

---

パスに無効なエスケープ・シーケンスが含まれています。

Path contains invalid escape sequence

**KECX04183-E**

---

パスに無効文字が含まれています : {0}

Path contains invalid character: {0}

**KECX04184-E**

---

フラグメントに無効文字が含まれています。

## 7. メッセージ

Fragment contains invalid character

### **KECX04185-E**

---

パスが null であるとフラグメントを設定できません。

Fragment cannot be set when path is null

### **KECX04186-E**

---

一般的な URI の場合のみ、フラグメントを設定できます。

Fragment can only be set for a generic URI

### **KECX04187-E**

---

URI にスキームがありません。

No scheme found in URI

### **KECX04188-E**

---

URI は空のパラメタで初期化できません。

Cannot initialize URI with empty parameters

### **KECX04189-E**

---

フラグメントはパスとフラグメントの両方に指定できません。

Fragment cannot be specified in both the path and fragment

### **KECX04190-E**

---

Query string はパスおよび Query string 内に指定できません。

Query string cannot be specified in path and query string

### **KECX04191-E**

---

ホストが指定されていない場合は、ポートは指定できません。

Port may not be specified if host is not specified

### **KECX04192-E**

---

ホストが指定されていない場合は、userinfo は指定できません。

Userinfo may not be specified if host is not specified

**KECX04193-W**

---

警告：出力ドキュメントのバージョンとして {0} を要求されています。本バージョンの XML はサポートされていません。出力ドキュメントのバージョンは 1.0 となります。

Warning: The version of the output document is requested to be '{0}'. This version of XML is not supported. The version of the output document will be '1.0'.

**KECX04194-E**

---

スキームが必要です。

Scheme is required!

**KECX04195-E**

---

SerializerFactory に渡された Properties オブジェクトは {0} プロパティを持ちません。

The Properties object passed to the SerializerFactory does not have a '{0}' property.

**KECX04196-W**

---

警告：エンコーディング {0} は Java runtime でサポートされていません。

Warning: The encoding '{0}' is not supported by the Java runtime.

**KECX04197-E**

---

メッセージキー {0} はメッセージクラス {1} にありません。

The message key '{0}' is not in the message class '{1}'

**KECX04198-E**

---

メッセージクラス {1} 内のメッセージ {0} のフォーマットは失敗しました。

The format of message '{0}' in message class '{1}' failed.

**KECX04199-E**

---

名前はコロンで始められません。

Name cannot start with a colon

**KECX04301-E**

---

メッセージキーに対応するエラーメッセージを、見つけることができません。

The error message corresponding to the message key can not be found.

**KECX04302-E**

---

次のメッセージをフォーマットしている間に、内部エラーが起きました：¥n

## 7. メッセージ

An internal error occurred while formatting the following message:¥n

### **KECX04303-E**

---

指定されたスキーマ言語は null にできません。

The schema language specified cannot be null.

### **KECX04304-E**

---

指定されたスキーマ言語の長さは 0 文字にできません。

The schema language specified cannot have a length of zero characters.

### **KECX04305-E**

---

Source 配列パラメタは null にできません。

The Source array parameter cannot be null.

### **KECX04306-E**

---

Source 配列パラメタには null の項目を含めることができません。

The Source array parameter cannot contain any items that are null.

### **KECX04307-E**

---

型 {0} の Source パラメタはこの SchemaFactory を認識していません。

Source parameter of type '{0}' is not recognized this SchemaFactory.

### **KECX04308-E**

---

Source パラメタは null にできません。

Source parameter cannot be null.

### **KECX04309-E**

---

型 {0} の Source パラメタはこのバリデータでは受け付けません。

Source parameter of type '{0}' is not accepted by this validator.

### **KECX04310-E**

---

型 {0} の Source パラメタは、型 {1} の結果パラメタと互換性がありません。

Source parameter of type '{0}' is not compatible with result parameter of type '{1}'.

### **KECX04311-E**

---

TypeInfoProvider は startElement コールバックの外で問い合わせできません。



A TypeInfoProvider cannot be queried outside of a startElement callback.

#### **KECX04312-E**

---

フィーチャー名は null にできません。

The feature name cannot be null.

#### **KECX04313-E**

---

プロパティ名は null にできません。

The property name cannot be null.

#### **KECX04314-E**

---

指定された SAXSource には InputSource がありません。

The SAXSource specified contains no InputSource.

#### **KECX04401-E**

---

メッセージキーに対応するエラーメッセージを、見つけることができません。

The error message corresponding to the message key can not be found.

#### **KECX04402-E**

---

次のメッセージをフォーマットしている間に、内部エラーが起きました：¥n

An internal error occurred while formatting the following message:¥n

#### **KECX04403-E**

---

XPointerProcessingError : XPointer 式の処理中にエラーが発生しました。 .

XPointerProcessingError: An error occurred while processing the XPointer expression.

#### **KECX04404-E**

---

InvalidXPointerToken : XPointer 式に無効なトークン {0} が含まれています。

InvalidXPointerToken: The XPointer expression contains the invalid token '{0}'

#### **KECX04405-E**

---

InvalidXPointerExpression : XPointer 式 {0} は無効です。

InvalidXPointerExpression: The XPointer expression '{0}' is invalid.

---

**KECX04406-E**

---

MultipleShortHandPointers : XPointer 式 {0} は無効です。式には二つ以上の ShortHand Pointer が含まれています。

MultipleShortHandPointers: The XPointer expression '{0}' is invalid. It has more than one ShortHand Pointer.

---

**KECX04407-E**

---

SchemeDataNotFollowedByCloseParenthesis : XPointer 式 {0} は無効です。SchemeData の後に ) 文字がありません。

SchemeDataNotFollowedByCloseParenthesis: The XPointer expression '{0}' is invalid. The SchemeData was not followed by a ')' character.

---

**KECX04408-W**

---

SchemeUnsupported : XPointer スキーム {0} はサポートされていません。

SchemeUnsupported: The XPointer scheme '{0}' is not supported.

---

**KECX04409-E**

---

InvalidShortHandPointer : ShortHand Pointer {0} の NCName は無効です。

InvalidShortHandPointer: The NCName of the ShortHand Pointer '{0}' is invalid.

---

**KECX04410-E**

---

UnbalancedParenthesisInXPointerExpression : XPointer 式 {0} は無効です。開かっこの数 {1} は閉かっこの数 {2} と等しくありません。

UnbalancedParenthesisInXPointerExpression: The XPointer expression '{0}' is invalid. The number of open parenthesis '{1}' is not equal to the number of close parenthesis '{2}'.

---

**KECX04411-E**

---

InvalidSchemeDataInXPointer : XPointer 式 {0} に無効な SchemeData が含まれています。

InvalidSchemeDataInXPointer: The XPointer expression '{0}' contains invalid SchemeData.

---

**KECX04412-E**

---

InvalidElementSchemeToken : element() スキーム XPointer 式に無効なトークン {0} が含まれています。

InvalidElementSchemeToken: The element() scheme XPointer expression contains the invalid token '{0}'

**KECX04413-E**

---

InvalidElementSchemeXPointer : Element Scheme XPointer 式 {0} は無効です。

InvalidElementSchemeXPointer: The Element Scheme XPointer expression '{0}' is invalid.

**KECX04414-E**

---

XPointerElementSchemeProcessingError : XPointer element() Scheme 式の処理中にエラーが発生しました。 .

XPointerElementSchemeProcessingError: An error occurred while processing the XPointer element() Scheme expression.

**KECX04415-E**

---

InvalidNCNameInElementSchemeData : element() Scheme には無効な NCName を持つ ShortHand Pointer {0} が含まれています。

InvalidNCNameInElementSchemeData: The element() Scheme contains a ShortHand Pointer '{0}' with an invalid NCName.

**KECX04416-E**

---

InvalidChildSequenceCharacter : element() Scheme には無効な子シーケンス文字 {0} が含まれています。

InvalidChildSequenceCharacter: The element() Scheme contains an invalid child sequence character '{0}'.

## 7.2.5 KECX06000 ~ KECX06999

### **KECX06001-E**

---

メッセージキーに対応するエラーメッセージを、見つけることができません。

The error message corresponding to the message key can not be found.

### **KECX06002-E**

---

次のメッセージをフォーマットしている間に、内部エラーが発生しました。

An internal error occurred while formatting the following message:

### **KECX06003-E**

---

内部エラーが発生しました。{0}

Internal error: {0}.

### **KECX06004-E**

---

空白ファセット値は結合単純型 {0} に使用できません。

Whitespace facet value is not available for the union simpleType '{0}'

### **KECX06005-W**

---

ユーザの文法プールから返された文法の一つが、別の文法と矛盾します。

One of the grammar(s) returned from the user's grammar pool is in conflict with another grammar.

### **KECX06006-E**

---

一意性制約エラーです (cvc-identity-constraint.4.2.1)。要素 {0} に値のないキーが指定されています。

Identity Constraint error (cvc-identity-constraint.4.2.1): element "{0}" has a key with no value.

### **KECX06007-E**

---

フィールド {0} のスコープで重複適合しています。

Duplicate match in scope for field "{0}".

### **KECX06008-E**

---

要素 {1} の一意性制約宣言で、キーの値 {0} が重複しています。

Duplicate key value [{0}] declared for identity constraint of element "{1}".

**KECX06009-E**

---

要素 {1} の一意性制約宣言で、ユニークの値 {0} が重複しています。

Duplicate unique value [{0}] declared for identity constraint of element "{1}".

**KECX06010-E**

---

一意性制約エラーが発生しました。フィールド {0} が、selector の範囲内で複数個一致しました。フィールドはユニークな値でなければいけません。

Identity constraint error: field "{0}" matches more than one value within the scope of its selector; fields must match unique values.

**KECX06011-E**

---

要素の内容が、スキーマ中の要素宣言での "fixed" 属性の値と等しくありません。

The content of this element is not equivalent to the value of the "fixed" attribute in the element's declaration in the schema.

**KECX06012-E**

---

一意性制約エラーです (cvc-identity-constraint.4.2.3)。要素 {0} に空要素指定 nillable に true をセットしている要素と一致するキーが指定されています。

Identity Constraint error (cvc-identity-constraint.4.2.3): element "{0}" has a key which matches an element which has nillable set to true.

**KECX06013-E**

---

要素 {0} の <key name={1}> 一意性制約定義に対して、値の指定が不十分です。

Not enough values specified for <key name="{1}"> identity constraint specified for element "{0}".

**KECX06014-E**

---

値 {1} が指定されたキー {0} は、要素 {2} の一意性制約定義にはありません。

Key '{0}' with value '{1}' not found for identity constraint of element '{2}'.

**KECX06015-E**

---

要素 {0} の <keyref name={1}> 一意性制約定義に対して、値の指定が不十分です。

Not enough values specified for <keyref name="{1}"> identity constraint specified for element "{0}".

**KECX06016-E**

---

一意性制約エラーです。一意性制約 {0} に、スコープ外の key または unique を参照する keyref が指定されました。

## 7. メッセージ

Identity Constraint error: identity constraint "{0}" has a keyref which refers to a key or unique that is out of scope.

### **KECX06017-E**

---

キー参照定義 {0} は、{1} という名前の未知のキーを参照しています。

Key reference declaration "{0}" refers to unknown key with name "{1}".

### **KECX06018-E**

---

要素 {0} の <unique> 一意性制約定義に対して、値の指定が不十分です。

Not enough values specified for <unique> identity constraint specified for element "{0}".

### **KECX06019-E**

---

内部一意性制約エラーです。未知のフィールド {0} です。

Internal identity constraint error; unknown field "{0}".

### **KECX06020-E**

---

要素 {0} の属性 {1} に指定した値 {2} は型 {3} に関して不正です。

cvc-attribute.3: The value '{2}' of attribute '{1}' on element '{0}' is not valid with respect to its type, '{3}'.

### **KECX06021-E**

---

要素 {0} の属性 {1} に指定した値 {2} は、fixed 値制約に関して不正です。属性は {3} の値でなければいけません。

cvc-attribute.4: The value '{2}' of attribute '{1}' on element '{0}' is not valid with respect to its fixed {value constraint}. The attribute must have a value of '{3}'.

### **KECX06022-E**

---

型の内容型が空なので、要素 {0} にテキストまたは子要素情報項目を指定することはできません。

cvc-complex-type.2.1: Element '{0}' must have no character or element information item [children], because the type's content type is empty.

### **KECX06023-E**

---

要素 {0} に子要素を指定することはできません。また、値は妥当である必要があります。

cvc-complex-type.2.2: Element '{0}' must have no element [children], and the value must be valid.

**KECX06024-E**

---

型の内容型が要素だけなので、要素 {0} に子テキストを指定することはできません。

cvc-complex-type.2.3: Element '{0}' cannot have character [children], because the type's content type is element-only.

**KECX06025-E**

---

要素 {0} で始まる内容は不正です。{1} のうちの 하나가期待されます。

cvc-complex-type.2.4.a: Invalid content was found starting with element '{0}'. One of '{1}' is expected.

**KECX06026-E**

---

要素 {0} は完結していません。{1} のうちの 하나가期待されます。

cvc-complex-type.2.4.b: The content of element '{0}' is not complete. One of '{1}' is expected.

**KECX06027-E**

---

任意要素定義で、要素宣言が必要ですが、要素 {0} の宣言が見つかりません。

cvc-complex-type.2.4.c: The matching wildcard is strict, but no declaration can be found for element '{0}'.

**KECX06028-E**

---

要素 {0} で始まる内容は不正です。子要素はこの時点で期待されません。

cvc-complex-type.2.4.d: Invalid content was found starting with element '{0}'. No child element is expected at this point.

**KECX06029-E**

---

要素 {0} の属性 {1} に指定した値 {2} は不正です。属性 {1} は {3} の固定値を持っています。

cvc-complex-type.3.1: Value '{2}' of attribute '{1}' of element '{0}' is not valid with respect to the corresponding attribute use. Attribute '{1}' has a fixed value of '{3}'.

**KECX06030-E**

---

要素 {0} は、属性 {1} に対する属性ワイルドカードを持っていません。

cvc-complex-type.3.2.1: Element '{0}' does not have an attribute wildcard for attribute '{1}'.

**KECX06031-E**

---

属性 {1} は要素 {0} に指定できません。

cvc-complex-type.3.2.2: Attribute '{1}' is not allowed to appear in element '{0}'.

### **KECX06032-E**

---

要素 {0} には属性 {1} が必要です。

cvc-complex-type.4: Attribute '{1}' must appear on element '{0}'.

### **KECX06033-E**

---

要素 {0} の属性 {1} はワイルド ID です。しかし、ワイルド ID {2} がすでにあります。指定できるのは一つだけです。

cvc-complex-type.5.1: In element '{0}', attribute '{1}' is a Wild ID. But there is already a Wild ID '{2}'. There can be only one.

### **KECX06034-E**

---

要素 {0} では、属性 {1} はワイルド ID です。しかし、属性用途の中に、ID から派生した属性 {2} がすでに存在しています。

cvc-complex-type.5.2: In element '{0}', attribute '{1}' is a Wild ID. But there is already an attribute '{2}' derived from ID among the {attribute uses}.

### **KECX06036-E**

---

{0} は {1} の値として不正です。

cvc-datatype-valid.1.2.1: '{0}' is not a valid value for '{1}'.

### **KECX06037-E**

---

{0} は {1} のリスト型として不正です。

cvc-datatype-valid.1.2.2: '{0}' is not a valid value of list type '{1}'.

### **KECX06038-E**

---

{0} は {1} のユニオン型として不正です。

cvc-datatype-valid.1.2.3: '{0}' is not a valid value of union type '{1}'.

### **KECX06039-E**

---

要素 {0} の宣言が見つかりません。

cvc-elt.1: Cannot find the declaration of element '{0}'.

### **KECX06040-E**

---

要素宣言 {0} の属性 abstract の値には、false を指定しないといけません。

cvc-elt.2: The value of {abstract} in the element declaration for '{0}' must be false.



**KECX06041-E**

---

nillable プロパティが false のため、属性 {1} は要素 {0} に指定できません。

cvc-elt.3.1: Attribute '{1}' must not appear on element '{0}', because the {nillable} property of '{0}' is false.

**KECX06042-E**

---

{1} が指定されているため、要素 {0} にはテキストまたは要素情報を指定できません。

cvc-elt.3.2.1: Element '{0}' cannot have character or element information [children], because '{1}' is specified.

**KECX06043-E**

---

{1} が指定されているため、要素 {0} には fixed 値制約があってははいけません。

cvc-elt.3.2.2: There must be no fixed {value constraint} for element '{0}', because '{1}' is specified.

**KECX06044-E**

---

要素 {0} の属性 {1} の値 {2} が、正しい QName ではありません。

cvc-elt.4.1: The value '{2}' of attribute '{1}' of element '{0}' is not a valid QName.

**KECX06045-E**

---

{1} は、要素 {0} の型定義として解決できません。

cvc-elt.4.2: Cannot resolve '{1}' to a type definition for element '{0}'.

**KECX06046-E**

---

型 {1} は要素 {0} の型定義、{2} からの妥当な派生ではありません。

cvc-elt.4.3: Type '{1}' is not validly derived from the type definition, '{2}', of element '{0}'.

**KECX06047-E**

---

要素 {0} の値制約 {2} は、型 {1} のデフォルトに有効ではありません。

cvc-elt.5.1.1: {value constraint} '{2}' of element '{0}' is not a valid default value for type '{1}'.

**KECX06048-E**

---

要素 {0} には子要素情報項目を指定できません。

cvc-elt.5.2.2.1: Element '{0}' must have no element information item [children].

**KECX06049-E**

---

要素 {0} の値 {1} が、固定値制約の値 {2} と一致しません。

## 7. メッセージ

cvc-elt.5.2.2.2.1: The value '{1}' of element '{0}' does not match the fixed {value constraint} value '{2}'.

### **KECX06050-E**

---

要素 {0} の値 {1} が、値制約の値 {2} と一致しません。

cvc-elt.5.2.2.2.2: The value '{1}' of element '{0}' does not match the {value constraint} value '{2}'.

### **KECX06051-E**

---

値 {0} は enumeration ファセットに指定した {1} に関して不正です。enumeration からの値を指定しなければいけません。

cvc-enumeration-valid: Value '{0}' is not facet-valid with respect to enumeration '{1}'. It must be a value from the enumeration.

### **KECX06052-E**

---

値 {0} の小数部桁数 {1} は、fractionDigits ファセットに指定した桁数 {2} を超えています。

cvc-fractionDigits-valid: Value '{0}' has {1} fraction digits, but the number of fraction digits has been limited to {2}.

### **KECX06053-E**

---

IDREF{0} に結合する ID/IDREF がありません。

cvc-id.1: There is no ID/IDREF binding for IDREF '{0}'.

### **KECX06054-E**

---

ID の値 {0} が複数出現しています。

cvc-id.2: There are multiple occurrences of ID value '{0}'.

### **KECX06055-E**

---

一意性制約 {0} のフィールドは要素 {1} と一致しましたが、この要素は単純型を持っていません。

cvc-id.3: A field of identity constraint '{0}' matched element '{1}', but this element does not have a simple type.

### **KECX06056-E**

---

長さが {1} の値 {0} は、型 {3} の length ファセットに指定した長さ {2} と異なります。

cvc-length-valid: Value '{0}' with length = '{1}' is not facet-valid with respect to length '{2}' for type '{3}'.

**KECX06057-E**

---

値 {0} は型 {2} の maxExclusive ファセットに指定した値 {1} の範囲外です。

cvc-maxExclusive-valid: Value '{0}' is not facet-valid with respect to maxExclusive '{1}' for type '{2}'.

**KECX06058-E**

---

値 {0} は型 {2} の maxInclusive ファセットに指定した値 {1} の範囲外です。

cvc-maxInclusive-valid: Value '{0}' is not facet-valid with respect to maxInclusive '{1}' for type '{2}'.

**KECX06059-E**

---

値 {0} の長さ {1} は、型 {3} の maxLength ファセットに指定した長さ {2} の範囲外です。

cvc-maxLength-valid: Value '{0}' with length = '{1}' is not facet-valid with respect to maxLength '{2}' for type '{3}'.

**KECX06060-E**

---

値 {0} は型 {2} の minExclusive ファセットに指定した値 {1} の範囲外です。

cvc-minExclusive-valid: Value '{0}' is not facet-valid with respect to minExclusive '{1}' for type '{2}'.

**KECX06061-E**

---

値 {0} は型 {2} の minInclusive ファセットに指定した値 {1} の範囲外です。

cvc-minInclusive-valid: Value '{0}' is not facet-valid with respect to minInclusive '{1}' for type '{2}'.

**KECX06062-E**

---

値 {0} の長さ {1} は、型 {3} の minLength ファセットに指定した長さ {2} の範囲外です。

cvc-minLength-valid: Value '{0}' with length = '{1}' is not facet-valid with respect to minLength '{2}' for type '{3}'.

**KECX06063-E**

---

値 {0} は型 {2} の pattern ファセットに指定したパターン {1} に一致しません。

cvc-pattern-valid: Value '{0}' is not facet-valid with respect to pattern '{1}' for type '{2}'.

**KECX06064-E**

---

値 {0} の桁数 {1} は、totalDigits ファセットに指定した桁数 {2} を超えています。

cvc-totalDigits-valid: Value '{0}' has {1} total digits, but the number of total digits has been limited to {2}.

---

**KECX06065-E**

---

要素 {0} の型定義は抽象的な型にしてはいけません。

cvc-type.2: The type definition cannot be abstract for element {0}.

---

**KECX06066-E**

---

要素 {0} は単純型のため、その名前空間名が 'http://www.w3.org/2001/XMLSchema-instance' と一致するもの、そのローカル名が type, nil, schemaLocation, noNamespaceSchemaLocation のうちの一つであるものを除いて、属性を指定できません。しかし、属性 {1} が見つかりました。

cvc-type.3.1.1: Element '{0}' is a simple type, so it cannot have attributes, excepting those whose namespace name is identical to 'http://www.w3.org/2001/XMLSchema-instance' and whose [local name] is one of 'type', 'nil', 'schemaLocation' or 'noNamespaceSchemaLocation'. However, the attribute, '{1}' was found.

---

**KECX06067-E**

---

要素 {0} は単純型のため、要素情報項目の子を指定できません。

cvc-type.3.1.2: Element '{0}' is a simple type, so it must have no element information item [children].

---

**KECX06068-E**

---

要素 {0} の値 {1} が不正です。

cvc-type.3.1.3: The value '{1}' of element '{0}' is not valid.

---

**KECX06069-E**

---

ドキュメントが見つからなかった、ドキュメントが読み込めなかった、またはドキュメントのルート要素が <xsd:schema> ではなかったため、スキーマ文書 {0} の読み込みに失敗しました。

schema\_reference.4: Failed to read schema document '{0}', because 1) could not find the document; 2) the document could not be read; 3) the root element of the document is not <xsd:schema>.

---

**KECX06070-E**

---

annotation 要素に指定できるのは、appinfo または documentation だけです。しかし、{0} が見つかりました。

src-annotation: <annotation> elements can only contain <appinfo> and <documentation> elements, but '{0}' was found.

---

**KECX06071-E**

---

属性宣言 {0} に、default と fixed の両方を同時に指定することはできません。それらのうちの一つだけを使用してください。

src-attribute.1: The properties 'default' and 'fixed' cannot both be present in attribute declaration '{0}'. Use only one of them.

### **KECX06072-E**

---

属性 {0} に default が指定されているため、use には optional を指定する必要があります。

src-attribute.2: : The property 'default' is present in attribute '{0}', so the value of 'use' must be 'optional'.

### **KECX06073-E**

---

ローカルな属性宣言に ref または name のどちらか一つが必要です。

src-attribute.3.1: One of "ref" or "name" must be present in a local attribute declaration.

### **KECX06074-E**

---

属性参照 {0} の内容に指定できるのは注釈だけです。

src-attribute.3.2: The content must match (annotation?) for the attribute reference '{0}'.

### **KECX06075-E**

---

属性 {0} は、型属性および匿名型の単純型の子を両方持っています。属性にはこれらのうち一つだけを指定します。

src-attribute.4: Attribute '{0}' has both a 'type' attribute and an anonymous 'simpleType' child. Only one of these is allowed for an attribute.

### **KECX06076-E**

---

ワイルドカードの共通集合は属性グループ {0} には表現できません。

src-attribute\_group.2: The intersection of wildcards is not expressible for attribute group '{0}'.

### **KECX06077-E**

---

属性グループ {0} の参照が無限ループ宣言されています。再帰的に属性グループ参照が続くことは、結局それ自体にリードバックします。

src-attribute\_group.3: Circular definitions detected for attribute group '{0}'. Recursively following attribute group references eventually leads back to itself.

### **KECX06078-E**

---

型 {0} のための複合型定義表現でエラーが発生しました。complexContent が使用されているとき、基底型は複合型でなければいけません。{1} は単純型です。

src-ct.1: Complex Type Definition Representation Error for type '{0}'. When <complexContent> is used, the base type must be a complexType. '{1}' is a simpleType.

### **KECX06079-E**

---

型 {0} のための複合型定義表現でエラーが発生しました。simpleContent が使用される場合、基底型は、その内容型が単純な complexType でなければいけません、または制限が指定される場合だけ、混合内容および空のパーティクルを備えた複合型、または拡張が指定される場合だけ、単純型でないといけません。{1} は、これらの条件のどれも満たしません。

src-ct.2.1: Complex Type Definition Representation Error for type '{0}'. When <simpleContent> is used, the base type must be a complexType whose content type is simple, or, only if restriction is specified, a complex type with mixed content and emptiable particle, or, only if extension is specified, a simple type. '{1}' satisfies none of these conditions.

### **KECX06080-E**

---

型 {0} のための複合型定義表現でエラーが発生しました。simpleContent を備えた complexType が混合内容および空のパーティクルを備えた complexType を制限する場合、それらは制限の子の間に simpleType でなければいけません。

src-ct.2.2: Complex Type Definition Representation Error for type '{0}'. When a complexType with simpleContent restricts a complexType with mixed content and emptiable particle, then there must be a <simpleType> among the children of <restriction>.

### **KECX06081-E**

---

型 {0} のための複合型定義表現でエラーが発生しました。ワイルドカードの共通集合は表現できません。

src-ct.4: Complex Type Definition Representation Error for type '{0}'. The intersection of wildcards is not expressible.

### **KECX06082-E**

---

要素宣言 {0} に、default と fixed の両方を同時に指定することはできません。それらのうちの一つだけを使用してください。

src-element.1: The properties 'default' and 'fixed' cannot both be present in element declaration '{0}'. Use only one of them.

### **KECX06083-E**

---

ローカルな要素宣言に ref または name のどちらか一つが必要です。

src-element.2.1: : One of "ref" or "name" must be present in a local element declaration.

### **KECX06084-E**

---

{0} が ref 属性を含んでいるので、その内容に指定できるのは注釈だけです。しかし、{1} が見つかりました。

src-element.2.2: Since '{0}' contains the 'ref' attribute, its content must match (annotation?). However, '{1}' was found.

---

#### KECX06085-E

要素 {0} は、型属性および匿名型の子の両方を持っています。要素にはこれらのうちの一つだけを指定します。

src-element.3: Element '{0}' has both a 'type' attribute and a 'anonymous type' child. Only one of these is allowed for an element.

---

#### KECX06086-E

import 要素情報項目の名前空間属性 {0} はそれが実在するスキーマの対象名前空間と同じではありません。

src-import.1.1: The namespace attribute '{0}' of an <import> element information item must not be the same as the targetNamespace of the schema it exists in.

---

#### KECX06087-E

ドキュメント {0} のルート要素は、名前空間名 'http://www.w3.org/2001/XMLSchema' およびローカル名 schema を持たなければいけません。

src-import.2: The root element of document '{0}' has to have the namespace name 'http://www.w3.org/2001/XMLSchema' and the local name 'schema'.

---

#### KECX06088-E

import 要素情報項目の名前空間属性 {0} は import されるドキュメントの対象名前空間 {1} と同じでないといけません。

src-import.3.1: The namespace attribute, '{0}', of an <import> element information item must be identical to the targetNamespace attribute, '{1}', of the imported document.

---

#### KECX06089-E

import 要素情報項目に名前空間属性がありません。そのため import されるドキュメントには対象名前空間属性は指定できません。しかし、import されるドキュメントで対象名前空間 {1} が見つかりました。

src-import.3.2: An <import> element information item that had no namespace attribute was found, so the imported document cannot have a targetNamespace attribute. However, the targetNamespace '{1}' was found in the imported document.

---

#### KECX06090-E

ドキュメント {0} のルート要素は、名前空間名 'http://www.w3.org/2001/XMLSchema' およびローカル名 schema を持たなければいけません。

## 7. メッセージ

src-include.1: The root element of document '{0}' has to have the namespace name 'http://www.w3.org/2001/XMLSchema' and the local name 'schema'.

### KECX06091-E

---

現在 {1} の参照づけられたスキーマの対象名前空間は、現在 {0} のインクルードしているスキーマの対象名前空間と同じでなければいけません。

src-include.2.1: The targetNamespace of the referenced schema, currently '{1}', must be identical to that of the including schema, currently '{0}'.

### KECX06092-E

---

ドキュメント {0} のルート要素は、名前空間名 'http://www.w3.org/2001/XMLSchema' およびローカル名 schema を持たなければいけません。

src-redefine.2: The root element of document '{0}' has to have the namespace name 'http://www.w3.org/2001/XMLSchema' and the local name 'schema'.

### KECX06093-E

---

redefine で読まれるスキーマの対象名前空間、現在 {1} は、読み込み元のスキーマの対象名前空間、現在 {0} と同じでなければいけません。

src-redefine.3.1: The targetNamespace of the referenced schema, currently '{1}', must be identical to that of the redefining schema, currently '{0}'.

### KECX06094-E

---

単純型に注釈以外の子が見つかりませんでした。redefine 要素の単純型の子には、それら自身を参照している基底の属性を持つ、制限派生の子孫がいなくてははいけません。

src-redefine.5.a.a: No non-annotation children of <simpleType> were found. <simpleType> children of <redefine> elements must have <restriction> descendants, with "base" attributes that refer to themselves.

### KECX06095-E

---

{0} は子要素として不正です。redefine 要素の単純型の子には、それら自身を参照している基底の属性を持つ、制限派生の子孫がいなくてははいけません。

src-redefine.5.a.b: '{0}' is not a valid child element. <simpleType> children of <redefine> elements must have <restriction> descendants, with 'base' attributes that refer to themselves.

### KECX06096-E

---

{0} は再定義された要素 {1} を参照する基底の属性を持っていません。redefine 要素の単純型の子には、それら自身を参照している基底の属性を持つ、制限派生の子孫がいなくてははいけません。



src-redefine.5.a.c: '{0}' does not have a 'base' attribute that refers to the redefined element, '{1}'.  
 <simpleType> children of <redefine> elements must have <restriction> descendants, with 'base' attributes that refer to themselves.

### KECX06097-E

---

複合型に注釈以外の子が見つかりませんでした。redefine 要素の複合型の子には、それら自身を参照している基底の属性を持つ、拡張派生または制限派生の子孫がいなくてははいけません。

src-redefine.5.b.a: No non-annotation children of <complexType> were found. <complexType> children of <redefine> elements must have <extension> or <restriction> descendants, with "base" attributes that refer to themselves.

### KECX06098-E

---

複合型に注釈以外の孫が見つかりませんでした。redefine 要素の複合型の子には、それら自身を参照している基底の属性を持つ、拡張派生または制限派生の子孫がいなくてははいけません。

src-redefine.5.b.b: No non-annotation grandchildren of <complexType> were found.  
 <complexType> children of <redefine> elements must have <extension> or <restriction> descendants, with "base" attributes that refer to themselves.

### KECX06099-E

---

{0} は孫要素として不正です。redefine 要素の複合型の子には、それら自身を参照している基底の属性を持つ、拡張派生または制限派生の子孫がいなくてははいけません。

src-redefine.5.b.c: '{0}' is not a valid grandchild element. <complexType> children of <redefine> elements must have <extension> or <restriction> descendants, with 'base' attributes that refer to themselves.

### KECX06100-E

---

{0} は再定義された要素 {1} を参照する基底の属性を持っていません。redefine 要素の複合型の子には、それら自身を参照している基底の属性を持つ、拡張派生または制限派生の子孫がいなくてははいけません。

src-redefine.5.b.d: '{0}' does not have a 'base' attribute that refers to the redefined element, '{1}'.  
 <complexType> children of <redefine> elements must have <extension> or <restriction> descendants, with 'base' attributes that refer to themselves.

### KECX06101-E

---

redefine 要素の子の group は、それ自体を参照する group を一つだけ持てます。これは {0} 個持っています。

src-redefine.6.1.1: If a group child of a <redefine> element contains a group referring itself, it must have exactly 1; this one has '{0}'.

---

### KECX06102-E

---

再定義されているグループの参照を含むグループ {0} の minOccurs および maxOccurs は、1 でなければいけません。

src-redefine.6.1.2: The group '{0}', which contains a reference to a group being redefined, must have 'minOccurs' = 'maxOccurs' = 1.

---

### KECX06103-E

---

再定義されたスキーマ中には、{0} と一致する名前のグループはありません。

src-redefine.6.2.1: No group in the redefined schema has a name matching '{0}'.

---

### KECX06104-E

---

グループ {0} は、それが再定義するグループを適切に制限しません。制約は次のものを破りました。{1}

src-redefine.6.2.2: Group '{0}' does not properly restrict the group it redefines; constraint violated: '{1}'.

---

### KECX06105-E

---

redefine 要素の子の attributeGroup は、それ自体を参照する attributeGroup を一つだけ持てます。これは {0} 個持っています。

src-redefine.7.1: If an attributeGroup child of a <redefine> element contains an attributeGroup referring itself, it must have exactly 1; this one has {0}.

---

### KECX06106-E

---

再定義されたスキーマ中には、{0} と一致する名前の attributeGroup はありません。

src-redefine.7.2.1: No attributeGroup in the redefined schema has a name matching '{0}'.

---

### KECX06107-E

---

attributeGroup{0} は、それが再定義する attributeGroup を適切に制限しません。制約は次のものを破りました。{1}

src-redefine.7.2.2: AttributeGroup '{0}' does not properly restrict the attributeGroup it redefines; constraint violated: '{1}'.

---

### KECX06108-E

---

{1} 構成要素で指定した名前 {0} が見つかりません。

src-resolve: Cannot resolve the name '{0}' to a(n) '{1}' component.

**KECX06109-E**

---

構成要素の {2} を解決するエラーです。{2} が名前空間を持っていないことが検知されました。しかし、対象名前空間のないコンポーネントはスキーマドキュメント {0} から参照づけることができません。{2} が名前空間を持つように意図される場合、恐らく、接頭辞を提供する必要があります。{2} が名前空間を持っていないことが意図される場合、namespace 属性のない import は {0} に加えられる必要があります。

src-resolve.4.1: Error resolving component '{2}'. It was detected that '{2}' has no namespace, but components with no target namespace are not referenceable from schema document '{0}'. If '{2}' is intended to have a namespace, perhaps a prefix needs to be provided. If it is intended that '{2}' has no namespace, then an 'import' without a "namespace" attribute should be added to '{0}'.

**KECX06110-E**

---

構成要素の {2} を解決するエラーです。{2} が名前空間 {1} にあることは検知されました。しかし、この名前空間からのコンポーネントはスキーマドキュメント {0} から参照づけることができません。これが正しくない名前空間である場合、恐らく、{2} の接頭辞を変更する必要があります。これが正確な名前空間である場合、適切な import タグは {0} に加えられる必要があります。

src-resolve.4.2: Error resolving component '{2}'. It was detected that '{2}' is in namespace '{1}', but components from this namespace are not referenceable from schema document '{0}'. If this is the incorrect namespace, perhaps the prefix of '{2}' needs to be changed. If this is the correct namespace, then an appropriate 'import' tag should be added to '{0}'.

**KECX06111-E**

---

restriction 要素は、base 属性および子に simpleType 要素の両方を持っています。一つだけが許可されます。

src-simple-type.2.a: A <restriction> element was found that has both a base [attribute] and a <simpleType> element among its [children]. Only one is allowed.

**KECX06112-E**

---

restriction 要素は、base 属性も子に simpleType 要素も持っていません。どちらか一つが必要です。

src-simple-type.2.b: A <restriction> element was found that has neither a base [attribute] nor a <simpleType> element among its [children]. One is required.

**KECX06113-E**

---

list 要素は、itemType 属性および子に simpleType 要素の両方を持っています。一つだけが許可されます。

src-simple-type.3.a: A <list> element was found that has both an itemType [attribute] and a <simpleType> element among its [children]. Only one is allowed.

---

#### KECX06114-E

---

list 要素は、itemType 属性も子に simpleType 要素も持っていません。どちらか一つが必要です。

src-simple-type.3.b: A <list> element was found that has neither an itemType [attribute] nor a <simpleType> element among its [children]. One is required.

---

#### KECX06115-E

---

ファセット {0} は複数回指定されています。

src-single-facet-value: The facet '{0}' is defined more than once.

---

#### KECX06116-E

---

union 要素は、空ではない memberTypes 属性を指定するか、または子に一つ以上の simpleType 要素を指定しなければいけません。

src-union-memberTypes-or-simpleTypes: A <union> element must have either a non-empty memberTypes [attribute] or at least one <simpleType> element among its [children].

---

#### KECX06117-E

---

属性グループ {0} のためのエラーです。同じ名前と対象名前空間で使用する重複属性が指定されています。重複する属性使用の名前は {1} です。

ag-props-correct.2: Error for attribute group '{0}'. Duplicate attribute uses with the same name and target namespace are specified. Name of duplicate attribute use is '{1}'.

---

#### KECX06118-E

---

属性グループ {0} のためのエラーです。二つの属性宣言、{1} および {2} が、ID から派生した型を持っています。

ag-props-correct.3: Error for attribute group '{0}'. Two attribute declarations, '{1}' and '{2}' have types which are derived from ID.

---

#### KECX06119-E

---

属性 {0} の値制約値 {1} は不正です。

a-props-correct.2: Invalid value constraint value '{1}' in attribute '{0}'.

---

#### KECX06120-E

---

属性の { 型定義 } が ID である、または ID から派生しているため、属性 {0} に fixed または default を指定できません。

a-props-correct.3: Attribute '{0}' cannot use 'fixed' or 'default', because the attribute's {type definition} is ID, or is derived from ID.

**KECX06121-E**

---

属性宣言 {0} に固定値 {1} が指定されました。したがって、{0} を参照する属性使用がさらに値制約を持っている場合、それは固定で値は {1} でなければいけません。

au-props-correct.2: In the attribute declaration of '{0}', a fixed value of '{1}' was specified. So if the attribute use referring to '{0}' also has a {value constraint}, it must be fixed and its value must be '{1}'.

**KECX06122-E**

---

all モデルグループは、minOccurs=maxOccurs=1 のパーティクルとならないといけません。そのパーティクルは複合型定義の内容型を構成する一組の一部でなければいけません。

cos-all-limited.1.2: An 'all' model group must appear in a particle with {min occurs} = {max occurs} = 1, and that particle must be part of a pair which constitutes the {content type} of a complex type definition.

**KECX06123-E**

---

all モデルグループ中の要素の maxOccurs には、0 または 1 を指定しなければいけません。要素 {1} の値 {0} は不正です。

cos-all-limited.2: The {max occurs} of an element in an 'all' model group must be 0 or 1. The value '{0}' for element '{1}' is invalid.

**KECX06124-E**

---

型 {1} にはファセット {0} は許可されていません。

cos-applicable-facets: Facet '{0}' is not allowed by type {1}.

**KECX06125-E**

---

型 {0} は型 {1} から拡張によって派生されました。しかし、{1} の final 属性は拡張による派生を禁止しています。

cos-ct-extends.1.1: Type '{0}' was derived by extension from type '{1}'. However, the 'final' attribute of '{1}' forbids derivation by extension.

**KECX06126-E**

---

派生した型の内容型およびその基底の内容型は、共に混合内容であるか共に要素であるかのどちらかです。型 {0} の内容型は要素ですが、基底の内容型は異なります。

cos-ct-extends.1.4.3.2.2.1.a: The content type of a derived type and that of its base must both be mixed or both be element-only. Type '{0}' is element only, but its base type is not.

### **KECX06127-E**

---

派生した型の内容型およびその基底の内容型は、共に混合内容であるか共に要素であるかのどちらかです。型 {0} の内容型は混合内容ですが、基底の内容型は異なります。

cos-ct-extends.1.4.3.2.2.1.b: The content type of a derived type and that of its base must both be mixed or both be element-only. Type '{0}' is mixed, but its base type is not.

### **KECX06128-E**

---

型 {0} のエラーです。名前 {1} を備えた、タイプが異なる複数の要素が、モデルグループにあります。

cos-element-consistent: Error for type '{0}'. Multiple elements with name '{1}', with different types, appear in the model group.

### **KECX06129-E**

---

リスト型 {0} の定義で、型 {1} はアトムックではないため不正なリスト要素型です ({1} はリストを含んでいるリスト型またはユニオン型のどちらかです)。

cos-list-of-atomic: In the definition of list type '{0}', type '{1}' is an invalid list element type because it is not atomic ('{1}' is either a list type, or a union type which contains a list).

### **KECX06130-E**

---

{0} と {1}(またはそれらの代替グループからの要素) は Unique Particle Attribution に違反しています。スキーマを確認中に、二つのパーティクルが規則に合わなくなるおそれがあります。

cos-nonambig: {0} and {1} (or elements from their substitution group) violate "Unique Particle Attribution". During validation against this schema, ambiguity would be created for those two particles.

### **KECX06131-E**

---

派生したパーティクルは空です。また、基底は空ではありません。

cos-particle-restrict.a: Derived particle is empty, and base is not emptiable.

### **KECX06132-E**

---

基底のパーティクルは空です。しかし、派生したパーティクルは空ではありません。

cos-particle-restrict.b: Base particle is empty, but derived particle is not.

### **KECX06133-E**

---

禁止されたパーティクル制限です。{0}

cos-particle-restrict.2: Forbidden particle restriction: '{0}'.

**KECX06134-E**

---

型 {1} はアトミックです。したがって、その基底型定義 {0} はアトミックの単純型定義またはビルトインデータ型でなければいけません。

cos-st-restricts.1.1: The type '{1}' is atomic, so its {base type definition}, '{0}', must be an atomic simple type definition or a built-in primitive datatype.

**KECX06135-E**

---

リスト型 {0} の定義では、それがリストを含んでいるリスト型またはユニオン型のどちらかであるため、型 {1} は無効の項目型です。

cos-st-restricts.2.1: In the definition of list type '{0}', type '{1}' is an invalid item type because it is either a list type, or a union type that contains a list.

**KECX06136-E**

---

項目型定義 {0} の final コンポーネントは、リストを含んでいます。これは、リスト型 {1} のために項目型として {0} を使用できないことを意味します。

cos-st-restricts.2.3.1.1: The {final} component of the {item type definition}, '{0}', contains 'list'. This means that '{0}' cannot be used as an item type for list type '{1}'.

**KECX06137-E**

---

メンバ型定義 {0} の final コンポーネントは、ユニオンを含んでいます。これは、ユニオン型 {1} のためにメンバ型として {0} を使用できないことを意味します。

cos-st-restricts.3.3.1.1: The {final} component of the {member type definitions}, '{0}', contains 'union'. This means that '{0}' cannot be used as an member type for union type '{1}'.

**KECX06138-E**

---

要素 {0} は値制約を持っているため、混合か単純な内容モデルでなければいけません。

cos-valid-default.2.1: Element '{0}' has a value constraint and must have a mixed or simple content model.

**KECX06139-E**

---

要素 {0} が値制約を持ち、その型定義が混合の内容型なので、内容型のパーティクルは空でなければいけません。

cos-valid-default.2.2.2: Since element '{0}' has a {value constraint} and its type definition has mixed {content type}, then the particle of the {content type} must be emptiable.

**KECX06140-E**

---

keyref{0} と key{1} のフィールド指定の数は互いに一致しなければいけません。

c-props-correct.2: Cardinality of Fields for keyref '{0}' and key '{1}' must match each other.

---

### KECX06141-E

複合型 {0} は無限ループ宣言されています。これは、それ自身の型の階層に {0} が含まれることを意味し、それはエラーです。

ct-props-correct.3: Circular definitions detected for complex type '{0}'. This means that '{0}' is contained in its own type hierarchy, which is an error.

---

### KECX06142-E

型 {0} のエラーです。同じ名前と対象名前空間で使用する重複属性が指定されています。重複する属性使用の名前は {1} です。

ct-props-correct.4: Error for type '{0}'. Duplicate attribute uses with the same name and target namespace are specified. Name of duplicate attribute use is '{1}'.

---

### KECX06143-E

型 {0} のエラーです。二つの属性宣言、{1} および {2} が、ID から派生した型を持っています。

ct-props-correct.5: Error for type '{0}'. Two attribute declarations, '{1}' and '{2}' have types which are derived from ID.

---

### KECX06144-E

型 {0} は型 {1} から制限によって派生しました。しかし、{1} は、制限によって派生を禁止する final 属性を持っています。

derivation-ok-restriction.1: Type '{0}' was derived by restriction from type '{1}'. However, '{1}' has a {final} property that forbids derivation by restriction.

---

### KECX06145-E

型 {0} のエラーです。この型での属性使用 {1} は、{2} の use 値を持っています。それは、基底型での適合する属性使用の required 値と矛盾します。

derivation-ok-restriction.2.1.1: Error for type '{0}'. The attribute use '{1}' in this type has a 'use' value of '{2}', which is inconsistent with the value of 'required' in a matching attribute use in the base type.

---

### KECX06146-E

型 {0} のエラーです。この型での属性使用 {1} は、型 {2} を持っています。それは、基底型での適合する属性使用の型 {3} から正当に派生していません。

derivation-ok-restriction.2.1.2: Error for type '{0}'. The attribute use '{1}' in this type has type '{2}', which is not validly derived from '{3}', the type of the matching attribute use in the base type.



**KECX06147-E**

---

型 {0} のエラーです。この型での属性使用 {1} は、固定でない実効値制約を持っています。基底型での適合する属性使用の実行値制約は固定です。

derivation-ok-restriction.2.1.3.a: Error for type '{0}'. The attribute use '{1}' in this type has an effective value constraint which is not fixed, and the effective value constraint of the matching attribute use in the base type is fixed.

**KECX06148-E**

---

型 {0} のエラーです。この型での属性使用 {1} は、値 {2} の固定実効値制約を持っています。基底型での適合する属性使用の固定実行値制約の値 {3} と矛盾します。

derivation-ok-restriction.2.1.3.b: Error for type '{0}'. The attribute use '{1}' in this type has an effective value constraint fixed with a value of '{2}', which is not consistent with the value of '{3}' for the fixed effective value constraint of the matching attribute use in the base type.

**KECX06149-E**

---

型 {0} のエラーです。この型での属性使用 {1} は基底の中で適合する属性を使用していません。また、基底型はワイルドカード属性を持っていません。

derivation-ok-restriction.2.2.a: Error for type '{0}'. The attribute use '{1}' in this type does not have a matching attribute use in the base, and the base type does not have a wildcard attribute.

**KECX06150-E**

---

型 {0} のエラーです。この型での属性使用 {1} は基底の中で適合する属性を使用していません。また、基底型のワイルドカードはこの属性使用の名前空間 {2} を許可しません。

derivation-ok-restriction.2.2.b: Error for type '{0}'. The attribute use '{1}' in this type does not have a matching attribute use in the base, and the wildcard in the base type does not allow the namespace '{2}' of this attribute use.

**KECX06151-E**

---

型 {0} のエラーです。基底型の属性使用 {1} には REQUIRED 指定がありますが、派生した型には一致する属性使用がありません。

derivation-ok-restriction.3: Error for type '{0}'. The attribute use '{1}' in the base type has REQUIRED as true, but there is no matching attribute use in the derived type.

**KECX06152-E**

---

型 {0} のエラーです。派生は属性ワイルドカードを持っていますが、基底は持っていません。

derivation-ok-restriction.4.1: Error for type '{0}'. The derivation has an attribute wildcard, but the base does not have one.

---

### KECX06153-E

---

型 {0} のエラーです。派生内のワイルドカードは基底の妥当なワイルドカードの部分集合ではありません。

derivation-ok-restriction.4.2: Error for type '{0}'. The wildcard in the derivation is not a valid wildcard subset of the one in the base.

---

### KECX06154-E

---

型 {0} のエラーです。派生 ({1}) 中のワイルドカードのプロセッサ検証指定は、基底 ({2}) のプロセッサ検証指定より弱い指定です。

derivation-ok-restriction.4.3: Error for type '{0}'. The process contents of the wildcard in the derivation ({1}) is weaker than that in the base ({2}).

---

### KECX06155-E

---

型 {0} のエラーです。この型の単純な内容型 {1} は、基底の単純な内容型 {2} の不正な制限です。

derivation-ok-restriction.5.2.2.1: Error for type '{0}'. The simple content type of this type, '{1}', is not a valid restriction of the simple content type of the base, '{2}'.

---

### KECX06156-E

---

型 {0} のエラーです。この型の内容型は空ですが、基底の内容型 {1} は空ではありません。

derivation-ok-restriction.5.3.2: Error for type '{0}'. The content type of this type is empty, but the content type of the base, '{1}', is not empty or emptiable.

---

### KECX06157-E

---

型 {0} のエラーです。この型の内容型は混合ですが、基底の内容型 {1} は違います。

derivation-ok-restriction.5.4.1.2: Error for type '{0}'. The content type of this type is mixed, but the content type of the base, '{1}', is not.

---

### KECX06158-E

---

型 {0} のエラーです。型のパーティクルは基底のパーティクルの不正な制限です。

derivation-ok-restriction.5.4.2: Error for type '{0}'. The particle of the type is not a valid restriction of the particle of the base.

---

### KECX06159-E

---

NOTATION 型 ({2}{1}) によって使用される {0} は、この型によって使用される記法要素を指定する enumeration ファセット値を指定しなければいけません。

enumeration-required-notation: The NOTATION type, '{0}' used by {2} '{1}', must have an enumeration facet value which specifies the notation elements used by this type.

**KECX06160-E**

---

enumeration 値 {0} が、基底型 {1} の値空間内ではありません。

enumeration-valid-restriction: Enumeration value '{0}' is not in the value space of the base type, {1}.

**KECX06161-E**

---

要素 {0} 内の値 {1} は値制約として不正です。

e-props-correct.2: Invalid value constraint value '{1}' in element '{0}'.

**KECX06162-E**

---

要素 {0} の型定義は、ヘッド要素 {1} の型定義から正当に派生しません。または、{1} の代替グループ排除の特性はこの派生を許可しません。

e-props-correct.4: The {type definition} of element '{0}' is not validly derived from the {type definition} of the substitutionHead '{1}', or the {substitution group exclusions} property of '{1}' does not allowed this derivation.

**KECX06163-E**

---

要素の { 型定義 } または { 型定義 } の { 内容型 } が ID である、または ID から派生しているので、要素 {0} 上に { 値制約 } があってはいけません。

e-props-correct.5: A {value constraint} must not be present on element '{0}', because the element's {type definition} or {type definition}s {content type} is ID, or is derived from ID.

**KECX06164-E**

---

要素 {0} に無限ループの代替グループが検知されました。

e-props-correct.6: Circular substitution group detected for element '{0}'.

**KECX06165-E**

---

{2} の定義では、fractionDigits ファセットの値 {0} は不正です。祖先の型のうちの一つの中でセットされた fractionDigits の値 {1} 以下でなければいけません。

fractionDigits-valid-restriction: In the definition of {2}, the value '{0}' for the facet 'fractionDigits' is invalid, because it must be <= the value for 'fractionDigits' which was set to '{1}' in one of the ancestor types.

**KECX06166-E**

---

{2} の定義では、fractionDigits ファセットの値 {0} は不当です。値は totalDigits の値 {1} 以下でなければいけません。

fractionDigits-totalDigits: In the definition of {2}, the value '{0}' for the facet 'fractionDigits' is invalid, because the value must be <= the value for 'totalDigits' which is '{1}'.

### **KECX06167-E**

---

length ファセットは、minLength ファセットまたは maxLength ファセットと同時に指定できません。しかし、{0} は length ファセット {1}, minLength ファセット {2} および maxLength ファセット {3} を持っています。

length-minLength-maxLength.a: It is an error to specify both length and either of minLength or maxLength. However, {0} has length = '{1}', minLength = '{2}' and maxLength = '{3}'.

### **KECX06168-E**

---

length ファセットは、minLength ファセットまたは maxLength ファセットと同時に指定できません。しかし、{0} は length ファセット {1} および minLength ファセット {2} を持っています。

length-minLength-maxLength.b: It is an error to specify both length and either of minLength or maxLength. However, {0} has length = '{1}' and minLength = '{2}'.

### **KECX06169-E**

---

length ファセットは、minLength ファセットまたは maxLength ファセットと同時に指定できません。しかし、{0} は length ファセット {1} および maxLength ファセット {2} を持っています。

length-minLength-maxLength.c: It is an error to specify both length and either of minLength or maxLength. However, {0} has length = '{1}' and maxLength = '{2}'.

### **KECX06170-E**

---

型 {2} のエラーです。length ファセットの値 {0} は基底型 {1} の値と同じでなければいけません。

length-valid-restriction: Error for type '{2}'. The value of length = '{0}' must be = the value of that of the base type '{1}'.

### **KECX06171-E**

---

型 {2} のエラーです。maxExclusive 値 {0} は基底型 {1} の maxExclusive 以下でなければいけません。

maxExclusive-valid-restriction.1: Error for type '{2}'. The maxExclusive value = '{0}' must be <= maxExclusive of the base type '{1}'.

### **KECX06172-E**

---

型 {2} のエラーです。maxExclusive 値 {0} は基底型 {1} の maxInclusive 以下でなければいけません。

maxExclusive-valid-restriction.2: Error for type '{2}'. The maxExclusive value = '{0}' must be <= maxInclusive of the base type '{1}'.

**KECX06173-E**

---

型 {2} のエラーです。maxExclusive 値 {0} は基底型 {1} の minInclusive より大きくなければいけません。

maxExclusive-valid-restriction.3: Error for type '{2}'. The maxExclusive value ='{0}' must be > minInclusive of the base type '{1}'.

**KECX06174-E**

---

maxExclusive 値 {0} は基底型 {1} の minExclusive より大きくなければいけません。

maxExclusive-valid-restriction.4: Error for type '{2}'. The maxExclusive value ='{0}' must be > minExclusive of the base type '{1}'.

**KECX06175-E**

---

同一データ型に maxInclusive と maxExclusive を同時に指定できません。{2} には maxInclusive{0} と maxExclusive{1} があります。

maxInclusive-maxExclusive: It is an error for both maxInclusive and maxExclusive to be specified for the same datatype. In {2}, maxInclusive = '{0}' and maxExclusive = '{1}'.

**KECX06176-E**

---

型 {2} のエラーです。maxInclusive 値 {0} は、基底型 {1} の maxInclusive 以下でなければいけません。

maxInclusive-valid-restriction.1: Error for type '{2}'. The maxInclusive value ='{0}' must be <= maxInclusive of the base type '{1}'.

**KECX06177-E**

---

型 {2} のエラーです。maxInclusive 値 {0} は、基底型 {1} の maxExclusive より小さくなければいけません。

maxInclusive-valid-restriction.2: Error for type '{2}'. The maxInclusive value ='{0}' must be < maxExclusive of the base type '{1}'.

**KECX06178-E**

---

型 {2} のエラーです。maxInclusive 値 {0} は、基底型 {1} の minInclusive 以上でなければいけません。

maxInclusive-valid-restriction.3: Error for type '{2}'. The maxInclusive value ='{0}' must be >= minInclusive of the base type '{1}'.

### **KECX06179-E**

---

型 {2} のエラーです。maxInclusive 値 {0} は、基底型 {1} の minExclusive より大きくなければいけません。

maxInclusive-valid-restriction.4: Error for type '{2}'. The maxInclusive value = '{0}' must be > minExclusive of the base type '{1}'.

### **KECX06180-E**

---

{2} の定義では、maxLength 値 {0} は、基底型 {1} の maxLength 値以下でなければいけません。

maxLength-valid-restriction: In the definition of {2}, maxLength value = '{0}' must be <= that of the base type '{1}'.

### **KECX06181-E**

---

グループ {0} の参照が無限ループ宣言されています。再帰的にグループ参照が続くことは、その項目がグループ自体であるパーティクルに結び付きます。

mg-props-correct.2: Circular definitions detected for group '{0}'. Recursively following the {term} values of the particles leads to a particle whose {term} is the group itself.

### **KECX06182-E**

---

{2} の定義では、minExclusive に {0} が指定されていますが、maxExclusive に指定した値 {1} 以下でないといけません。

minExclusive-less-than-equal-to-maxExclusive: In the definition of {2}, minExclusive value = '{0}' must be <= maxExclusive value = '{1}'.

### **KECX06183-E**

---

{2} の定義では、minExclusive に {0} が指定されていますが、maxInclusive に指定した値 {1} 未満でないといけません。

minExclusive-less-than-maxInclusive: In the definition of {2}, minExclusive value = '{0}' must be < maxInclusive value = '{1}'.

### **KECX06184-E**

---

型 {2} のエラーです。minExclusive 値 {0} は、基底型 {1} の minExclusive 以上でなければいけません。

minExclusive-valid-restriction.1: Error for type '{2}'. The minExclusive value = '{0}' must be >= minExclusive of the base type '{1}'.

**KECX06185-E**

---

型 {2} のエラーです。minExclusive 値 {0} は、基底型 {1} の maxInclusive 以下でなければいけません。

minExclusive-valid-restriction.2: Error for type '{2}'. The minExclusive value = '{0}' must be <= maxInclusive of the base type '{1}'.

**KECX06186-E**

---

型 {2} のエラーです。minExclusive 値 {0} は、基底型 {1} の minInclusive 以上でなければいけません。

minExclusive-valid-restriction.3: Error for type '{2}'. The minExclusive value = '{0}' must be >= minInclusive of the base type '{1}'.

**KECX06187-E**

---

型 {2} のエラーです。minExclusive 値 {0} は、基底型 {1} の maxExclusive より小さくなければいけません。

minExclusive-valid-restriction.4: Error for type '{2}'. The minExclusive value = '{0}' must be < maxExclusive of the base type '{1}'.

**KECX06188-E**

---

{2} の定義では、minInclusive に {0} が指定されていますが、maxInclusive に指定した値 {1} 以下でないといけません。

minInclusive-less-than-equal-to-maxInclusive: In the definition of {2}, minInclusive value = '{0}' must be <= maxInclusive value = '{1}'.

**KECX06189-E**

---

{2} の定義では、minInclusive に {0} が指定されていますが、maxExclusive に指定した値 {1} 未満でないといけません。

minInclusive-less-than-maxExclusive: In the definition of {2}, minInclusive value = '{0}' must be < maxExclusive value = '{1}'.

**KECX06190-E**

---

同一データ型に minInclusive と minExclusive を同時に指定できません。{2} には minInclusive{0} と minExclusive{1} があります。

minInclusive-minExclusive: It is an error for both minInclusive and minExclusive to be specified for the same datatype. In {2}, minInclusive = '{0}' and minExclusive = '{1}'.

#### **KECX06191-E**

---

型 {2} のエラーです。minInclusive 値 {0} は、基底型 {1} の minInclusive 以上でなければいけません。

minInclusive-valid-restriction.1: Error for type '{2}'. The minInclusive value = '{0}' must be >= minInclusive of the base type '{1}'.

#### **KECX06192-E**

---

型 {2} のエラーです。minInclusive 値 {0} は、基底型 {1} の maxInclusive 以下でなければいけません。

minInclusive-valid-restriction.2: Error for type '{2}'. The minInclusive value = '{0}' must be <= maxInclusive of the base type '{1}'.

#### **KECX06193-E**

---

型 {2} のエラーです。minInclusive 値 {0} は、基底型 {1} の minExclusive より大きくなければいけません。

minInclusive-valid-restriction.3: Error for type '{2}'. The minInclusive value = '{0}' must be > minExclusive of the base type '{1}'.

#### **KECX06194-E**

---

型 {2} のエラーです。minInclusive 値 {0} は、基底型 {1} の maxExclusive より小さくなければいけません。

minInclusive-valid-restriction.4: Error for type '{2}'. The minInclusive value = '{0}' must be < maxExclusive of the base type '{1}'.

#### **KECX06195-E**

---

{2} の定義では、minLength に指定した値 {0} は、maxLength に指定した値 {1} 未満でないといけません。

minLength-less-than-equal-to-maxLength: In the definition of {2}, value of minLength = '{0}' must be < value of maxLength = '{1}'.

#### **KECX06196-E**

---

{2} の定義では、minLength 値 {0} は、基底型 {1} の minLength 値以上でなければいけません。

minLength-valid-restriction: In the definition of {2}, minLength = '{0}' must be >= than that of the base type, '{1}'.

#### **KECX06197-E**

---

属性宣言の name は、xmlns と一致してはいけません。



no-xmlns: The '{name}' of an attribute declaration must not match "xmlns".

### KECX06198-E

---

属性宣言の対象名前空間は、{0} と一致してはいけません。

no-xsi: The {target namespace} of an attribute declaration must not match '{0}'.

### KECX06199-E

---

{0} の宣言で、minOccurs に指定した {1} は maxOccurs の {2} を超えてはいけません。

p-props-correct.2.1: In the declaration of '{0}', the value of 'minOccurs' is '{1}', but it must not be greater than the value of 'maxOccurs', which is '{2}'.

### KECX06200-E

---

パーティクル間に、完全で機能的なマッピングはありません。

rcase-MapAndSum.1: There is not a complete functional mapping between the particles.

### KECX06201-E

---

グループの出現範囲 ({0},{1}) は、基底のグループの出現範囲 ({2},{3}) の不正な制限です。

rcase-MapAndSum.2: Group's occurrence range, ({0},{1}), is not a valid restriction of base group's occurrence range, ({2},{3}).

### KECX06202-E

---

要素が、異なる名前および対象名前空間を持っています。名前空間 {1} の要素 {0} および名前空間 {3} の要素 {2} です。

rcase-NameAndTypeOK.1: Elements have names and target namespaces which are not the same: Element '{0}' in namespace '{1}' and element '{2}' in namespace '{3}'.

### KECX06203-E

---

項目が要素宣言 {0} のパーティクルエラーです。要素宣言の nillable が true ですが、対応する基底型のパーティクルの要素宣言の nillable は false です。

rcase-NameAndTypeOK.2: Error for the particle whose {term} is the element declaration '{0}'. The element declaration's {nillable} is true, but the corresponding particle in the base type has an element declaration whose {nillable} is false.

### KECX06204-E

---

項目が要素宣言 {0} のパーティクルエラーです。出現範囲 ({1},{2}) は基底の範囲 ({3},{4}) の不正な制限です。

rcase-NameAndTypeOK.3: Error for the particle whose {term} is the element declaration '{0}'. Its occurrence range, ({1},{2}), is not a valid restriction of the range, ({3},{4}), of the corresponding particle in the base type.

### **KECX06205-E**

---

要素 {0} は fixed ではありませんが、対応する基底型の要素は値 {1} の fixed です。

rcase-NameAndTypeOK.4.a: Element '{0}' is not fixed, but the corresponding element in the base type is fixed with value '{1}'.

### **KECX06206-E**

---

要素 {0} は値 {1} の fixed ですが、対応する基底型の要素は値 {2} の fixed です。

rcase-NameAndTypeOK.4.b: Element '{0}' is fixed with value '{1}', but the corresponding element in the base type is fixed with value '{2}'.

### **KECX06207-E**

---

要素 {0} のための一意性制約は、基底のサブセットではありません。

rcase-NameAndTypeOK.5: Identity constraints for element '{0}' are not a subset of those in base.

### **KECX06208-E**

---

要素 {0} は、基底の要素の型から派生していない型を持っています。

rcase-NameAndTypeOK.6: The disallowed substitutions for element '{0}' are not a superset of those in the base.

### **KECX06209-E**

---

要素 {0} の型 {1} は、基底の要素の型 {2} から派生していません。

rcase-NameAndTypeOK.7: The type of element '{0}', '{1}', is not derived from the type of the base element, '{2}'.

### **KECX06210-E**

---

要素 {0} は、基底のワイルドカードで許可されていない名前空間 {1} を持っています。

rcase-NSCompat.1: Element '{0}' has a namespace '{1}' which is not allowed by the wildcard in the base.

### **KECX06211-E**

---

要素宣言 {0} の項目のパーティクルエラーです。出現範囲 ({1},{2}) は、基底型の対応するパーティクルの出現範囲 ({3},{4}) の不正な制限です。

rcase-NSCompat.2: Error for the particle whose {term} is the element declaration '{0}'. Its occurrence range, ({1},{2}), is not a valid restriction of the range, ({3},{4}), of the corresponding particle in the base type.

### KECX06212-E

---

パーティクル間に、完全で機能的なマッピングはありません。

rcase-NSRecurseCheckCardinality.1: There is not a complete functional mapping between the particles.

### KECX06213-E

---

グループの出現範囲は、基底のワイルドカードの範囲の有効な制限ではありません。

rcase-NSRecurseCheckCardinality.2: Group's occurrence range, ({0},{1}), is not a valid restriction of base wildcard's range, ({2},{3}).

### KECX06214-E

---

ワイルドカードは基底の対応するワイルドカードの部分集合ではありません。

rcase-NSSubset.1: Wildcard is not a subset of corresponding wildcard in base.

### KECX06215-E

---

ワイルドカードの出現範囲 ({0},{1}) は、基底の出現範囲 ({2},{3}) の不正な制限です。

rcase-NSSubset.2: Wildcard's occurrence range, ({0},{1}), is not a valid restriction of that in the base, ({2},{3}).

### KECX06216-E

---

ワイルドカードのプロセッサ検証指定 {0} は、基底のワイルドカードのプロセッサ検証指定 {1} より弱い指定です。

rcase-NSSubset.3: Wildcard's process contents, '{0}', is weaker than that in the base, '{1}'.

### KECX06217-E

---

グループの出現範囲 ({0},{1}) は、基底のグループの出現範囲 ({2},{3}) の不正な制限です。

rcase-Recurse.1: Group's occurrence range, ({0},{1}), is not a valid restriction of base group's occurrence range, ({2},{3}).

### KECX06218-E

---

パーティクル間に、完全で機能的なマッピングはありません。

rcase-Recurse.2: There is not a complete functional mapping between the particles.

### **KECX06219-E**

---

グループの出現範囲  $\{0, \{1\}\}$  は、基底のグループの出現範囲  $\{2, \{3\}\}$  の不正な制限です。

rcase-RecurseLax.1: Group's occurrence range,  $\{0, \{1\}\}$ , is not a valid restriction of base group's occurrence range,  $\{2, \{3\}\}$ .

### **KECX06220-E**

---

パーティクル間に完全で機能的なマッピングはありません。

rcase-RecurseLax.2: There is not a complete functional mapping between the particles.

### **KECX06221-E**

---

グループの出現範囲  $\{0, \{1\}\}$  は、基底のグループの出現範囲  $\{2, \{3\}\}$  の不正な制限です。

rcase-RecurseUnordered.1: Group's occurrence range,  $\{0, \{1\}\}$ , is not a valid restriction of base group's occurrence range,  $\{2, \{3\}\}$ .

### **KECX06222-E**

---

パーティクル間に完全で機能的なマッピングはありません。

rcase-RecurseUnordered.2: There is not a complete functional mapping between the particles.

### **KECX06223-E**

---

スキーマは同一名称のグローバルな構成要素を持てません。 $\{0\}$  が重複指定されています。

sch-props-correct.2: A schema cannot contain two global components with the same name; this schema contains two occurrences of ' $\{0\}$ '.

### **KECX06224-E**

---

単純型  $\{0\}$  の参照が無限ループ宣言されています。 $\{0\}$  が自分自身の型の階層に含まれているエラーです。

st-props-correct.2: Circular definitions have been detected for simple type ' $\{0\}$ '. This means that ' $\{0\}$ ' is contained in its own type hierarchy, which is an error.

### **KECX06225-E**

---

型  $\{0\}$  のエラーです。基底型定義  $\{1\}$  の final の値は、制限によって派生を禁止しています。

st-props-correct.3: Error for type ' $\{0\}$ '. The value of {final} of the {base type definition}, ' $\{1\}$ ', forbids derivation by restriction.

### **KECX06226-E**

---

$\{2\}$  の定義では、totalDigits ファセットの値  $\{0\}$  は不正です。祖先の型のうちの一つの中でセットされた totalDigits の値  $\{1\}$  以下でなければいけません。

totalDigits-valid-restriction: In the definition of {2}, the value '{0}' for the facet 'totalDigits' is invalid, because it must be <= the value for 'totalDigits' which was set to '{1}' in one of the ancestor types.

---

#### KECX06227-E

---

{0} の定義では、whiteSpace ファセットの値 {1} が不正です。祖先の型のうちの一つの中で collapse がセットされています。

whiteSpace-valid-restriction.1: In the definition of {0}, the value '{1}' for the facet 'whitespace' is invalid, because the value for 'whitespace' has been set to 'collapse' in one of the ancestor types.

---

#### KECX06228-E

---

{0} の定義では、whiteSpace ファセットの値 preserve が不正です。祖先の型のうちの一つの中で replace がセットされています。

whiteSpace-valid-restriction.2: In the definition of {0}, the value 'preserve' for the facet 'whitespace' is invalid, because the value for 'whitespace' has been set to 'replace' in one of the ancestor types.

---

#### KECX06229-E

---

要素 {0} の属性 {1} に不正な値が指定されています。記録された理由は {2} です。

s4s-att-invalid-value: Invalid attribute value for '{1}' in element '{0}'. Recorded reason: {2}

---

#### KECX06230-E

---

要素 {0} には属性 {1} が必要です。

s4s-att-must-appear: Attribute '{1}' must appear in element '{0}'.

---

#### KECX06231-E

---

属性 {1} は要素 {0} に指定できません。

s4s-att-not-allowed: Attribute '{1}' cannot appear in element '{0}'.

---

#### KECX06232-E

---

要素 {0} はスキーマ文書に指定できない要素です。

s4s-elt-invalid: Element '{0}' is not a valid element in a schema document.

---

#### KECX06233-E

---

{0} の内容は、{1} と一致しなければいけません。この問題は {2} から始まります。

s4s-elt-must-match.1: The content of '{0}' must match {1}. A problem was found starting at: {2}.

---

**KECX06234-E**

---

{0} の内容は、{1} と一致しなければいけません。十分な要素が見つかりませんでした。

s4s-elt-must-match.2: The content of '{0}' must match {1}. Not enough elements were found.

---

**KECX06235-E**

---

{0} の内容が不正です。要素 {1} は無効、不適切、または重複指定されています。

s4s-elt-invalid-content.1: The content of '{0}' is invalid. Element '{1}' is invalid, misplaced, or occurs too often.

---

**KECX06236-E**

---

{0} の内容が不正です。要素 {1} は空を指定することはできません。

s4s-elt-invalid-content.2: The content of '{0}' is invalid. Element '{1}' cannot be empty.

---

**KECX06237-E**

---

型 {0} の要素は schema 要素の子として許されていません。

s4s-elt-invalid-content.3: Elements of type '{0}' cannot appear after declarations as children of a <schema> element.

---

**KECX06238-E**

---

要素 {0} の名前空間はスキーマの名前空間 'http://www.w3.org/2001/XMLSchema' でなければいけません。

s4s-elt-schema-ns: The namespace of element '{0}' must be from the schema namespace, 'http://www.w3.org/2001/XMLSchema'.

---

**KECX06239-E**

---

空白以外の文字は、xs:appinfo および xs:documentation 以外にはスキーマ要素として許可されていません。{0} が該当部分です。

s4s-elt-character: Non-whitespace characters are not allowed in schema elements other than 'xs:appinfo' and 'xs:documentation'. Saw '{0}'.

---

**KECX06240-E**

---

field の値 {0} が不正です。

c-fields-xpaths: The field value = '{0}' is not valid.

---

**KECX06241-E**

---

式 {0} は、XML Schema に支援された XPath 部分集合に関して不正です。

c-general-xpath: The expression '{0}' is not valid with respect to the XPath subset supported by XML Schema.

### **KECX06242-E**

---

XPath 式 {0} での名前空間接頭辞は、名前空間に結び付けられませんでした。

c-general-xpath-ns: A namespace prefix in XPath expression '{0}' was not bound to a namespace.

### **KECX06243-E**

---

selector の値 {0} は不正です。selector の xpath には属性を含むことができません。

c-selector-xpath: The selector value = '{0}' is not valid; selector xpaths cannot contain attributes.

### **KECX06244-W**

---

スキーマ文書 {0} で、対象名前空間に空文字列は指定できません。

EmptyTargetNamespace: In schema document '{0}', the value of the 'targetNamespace' attribute cannot be an empty string.

### **KECX06245-E**

---

型 {0} の宣言で、ファセット {2} の値 {1} は基底型の値 {3} の範囲内でなければいけません。

FacetValueFromBase: In the declaration of type '{0}', value '{1}' of facet '{2}' must be from the value space of the base type, '{3}'.

### **KECX06246-E**

---

{3} の定義では、ファセット {0} の値 {1} は不正です。{0} の値は祖先の型の中で {2} がセットされています。また fixed は true です。

FixedFacetValue: In the definition of {3}, the value '{1}' for the facet '{0}' is invalid, because the value for '{0}' has been set to '{2}' in one of the ancestor types, and {fixed} = true.

### **KECX06247-E**

---

パターン値 {0} は有効な正規表現ではありません。報告されたエラーは次のとおりでした。{1}

InvalidRegex: Pattern value '{0}' is not a valid regular expression. The reported error was: '{1}'.

### **KECX06248-E**

---

パーサの現在の環境設定では、複合型の内容モデルの拡張に {0} ノード以上のノードを含めることはできません。

Current configuration of the parser doesn't allow the expansion of a content model for a complex type to contain more than {0} nodes.

### **KECX06249-W**

---

schemaLocation の値 {0} は、偶数個の URI を持たなければいけません。

SchemaLocation: schemaLocation value = '{0}' must have even number of URI's.

### **KECX06250-E**

---

期待する名前空間は {0} ですが、スキーマ文書の対象名前空間は {1} です。

TargetNamespace.1: Expecting namespace '{0}', but the target namespace of the schema document is '{1}'.

### **KECX06251-E**

---

名前空間を期待していませんが、スキーマ文書に対象名前空間 {1} があります。

TargetNamespace.2: Expecting no namespace, but the schema document has a target namespace of '{1}'.

### **KECX06252-E**

---

Entity{0} は宣言されていません。

UndeclaredEntity: Entity '{0}' is not declared.

### **KECX06253-E**

---

QName として {0} を解決できません。接頭辞 {1} は宣言されていません。

UndeclaredPrefix: Cannot resolve '{0}' as a QName: the prefix '{1}' is not declared.

### **KECX06254-E**

---

型 {0} の場合、length{1} の値を minLength {2} の値より小さい値に設定するとエラーになります。

length-minLength-maxLength.d: For type {0}, it is an error for the value of length '{1}' to be less than the value of minLength '{2}'.

### **KECX06255-E**

---

型 {0} の場合、length{1} の値を maxLength {2} の値より大きい値に設定するとエラーになります。

length-minLength-maxLength.e: For type {0}, it is an error for the value of length '{1}' to be greater than the value of maxLength '{2}'.

### **KECX06256-E**

---

notation 要素には、少なくとも一つの public または system がなければいけません。



PublicSystemOnNotation: At least one of 'public' and 'system' must appear in element 'notation'.

### **KECX06301-E**

---

メッセージキーに対応するエラーメッセージを、見つけることができません。

The error message corresponding to the message key can not be found.

### **KECX06302-E**

---

次のメッセージをフォーマットしている間に、内部エラーが起きました：

An internal error occurred while formatting the following message:

### **KECX06303-E**

---

null パラメタの場合、{0} は呼び出せません。

{0} cannot be called with 'null' parameter.

### **KECX06304-E**

---

未知のフィールドで {0} が呼び出されました：{1}

{0} called with an unknown field:{1}

### **KECX06305-E**

---

年 = {0}, 月 = {1}, 日 = {2}, 時 = {3}, 分 = {4}, 秒 = {5}, 小数秒 = {6}, タイムゾーン = {7} は、XML Gregorian Calendar 値の有効な表記ではありません。

Year = {0}, Month = {1}, Day = {2}, Hour = {3}, Minute = {4}, Second = {5}, fractionalSecond = {6}, Timezone = {7} , is not a valid representation of an XML Gregorian Calendar value.

### **KECX06306-E**

---

年 = {0}, 月 = {1}, 日 = {2}, 時 = {3}, 分 = {4}, 秒 = {5}, 小数秒 = {6}, タイムゾーン = {7} は、XML Gregorian Calendar 値の有効な表記ではありません。

Year = {0}, Month = {1}, Day = {2}, Hour = {3}, Minute = {4}, Second = {5}, fractionalSecond = {6}, Timezone = {7} , is not a valid representation of an XML Gregorian Calendar value.

### **KECX06307-E**

---

XMLGregorianCalendar に無効なフィールドセットが設定されました。

Invalid set of fields set for XMLGregorianCalendar

### **KECX06308-E**

---

小数秒の値 {0} が無効です。

## 7. メッセージ

Invalid value {0} for fractional second.

### **KECX06309-E**

---

{0} は、XML Gregorian Calendar 値の有効な表記ではありません。

{0} is not a valid representation of an XML Gregorian Calendar value.

### **KECX06310-E**

---

{1} フィールドに無効な値 {0}

Invalid value {0} for {1} field.

### **KECX06311-E**

---

{1} フィールドは負です。

{0} field is negative

### **KECX06312-E**

---

すべてのフィールド ( javax.xml.datatype.DatatypeConstants.Field ) が null です。

All the fields ( javax.xml.datatype.DatatypeConstants.Field ) are null.

### **KECX06313-E**

---

{0} 値の {1} はこのインプリメンテーションでは大きすぎてサポートできません。

{0} value "{1}" too large to be supported by this implementation

### **KECX06501-E**

---

内部エラーが発生しました。

ImplementationMessages.VAL\_BST

### **KECX06502-E**

---

内部エラーが発生しました。

ImplementationMessages.VAL\_UST

### **KECX06503-E**

---

内部エラーが発生しました。

ImplementationMessages.VAL\_NIICM

### **KECX06504-E**

---

parse は構文解析中に呼び出してはいけません。

parse may not be called while parsing

---

**KECX06507-E**

---

内部エラーが発生しました。スキーマファクトリクラス {0} は SchemaDVFactory から拡張できません。

Schema factory class {0} does not extend from SchemaDVFactory.

---

**KECX06508-E**

---

内部エラーが発生しました。

internal error

---

**KECX06509-E**

---

スキーマ文書 {0} で、対象名前空間の値は <http://www.w3.org/2001/XMLSchema-instance> であってはいけません。

TargetNamespace.3: In schema document '{0}', the value of the targetNamespace attribute must not be equal to "<http://www.w3.org/2001/XMLSchema-instance>".

## 7.2.6 KECX07000 ~ KECX07999

### **KECX07001-E**

---

複数のスタイルシートが同じファイル内に定義されています。

More than one stylesheet defined in the same file.

### **KECX07002-E**

---

テンプレート {0} はこのスタイルシート内にすでに定義されています。

Template '{0}' already defined in this stylesheet.

### **KECX07003-E**

---

テンプレート {0} はこのスタイルシート内に定義されていません。

Template '{0}' not defined in this stylesheet.

### **KECX07004-E**

---

変数 {0} は同じ有効範囲内に複数回定義されています。

Variable '{0}' is multiply defined in the same scope.

### **KECX07005-E**

---

変数またはパラメタ {0} が未定義です。

Variable or parameter '{0}' is undefined.

### **KECX07006-E**

---

クラス '{0}' が見つかりません。

Cannot find class '{0}'.

### **KECX07007-E**

---

外部メソッド {0} が見つかりません (public でなければいけません)。

Cannot find external method '{0}' (must be public).

### **KECX07008-E**

---

引数 / 戻り型をメソッド {0} への呼び出しで変換できません。

Cannot convert argument/return type in call to method '{0}'

**KECX07009-E**

---

ファイルまたは URI{0} が見つかりません。

File or URI '{0}' not found.

**KECX07010-E**

---

無効な URI{0} です。

Invalid URI '{0}'.

**KECX07011-E**

---

ファイルまたは URI{0} をオープンできません。

Cannot open file or URI '{0}'.

**KECX07012-E**

---

<xsl:stylesheet> または <xsl:transform> 要素が必要です。

<xsl:stylesheet> or <xsl:transform> element expected.

**KECX07013-E**

---

名前空間接頭辞 {0} が宣言されていません。

Namespace prefix '{0}' is undeclared.

**KECX07014-W**

---

関数 {0} への呼び出しを解決できません。

Unable to resolve call to function '{0}'.

**KECX07015-E**

---

{0} への引数はリテラル文字列でなければいけません。

Argument to '{0}' must be a literal string.

**KECX07016-E**

---

XPath 式 {0} を構文解析中にエラーが発生しました。

Error parsing XPath expression '{0}'.

**KECX07017-E**

---

必須属性 {0} がありません。

Required attribute '{0}' is missing.

---

**KECX07018-E**

---

XPath 式内の文字 {0} が正しくありません。

Illegal character '{0}' in XPath expression.

---

**KECX07019-E**

---

命令の処理のための名前 {0} が正しくありません。

Illegal name '{0}' for processing instruction.

---

**KECX07020-W**

---

属性 {0} が要素の外側です。

Attribute '{0}' outside of element.

---

**KECX07021-W**

---

属性 {0} が正しくありません。

Illegal attribute '{0}'.

---

**KECX07022-E**

---

import/include が相互に呼び出し合っています。スタイルシート {0} はすでにロードされています。

Circular import/include. Stylesheet '{0}' already loaded.

---

**KECX07023-W**

---

結果ツリー・フラグメントをソートできません (<xsl:sort> 要素は無視されます)。このノードは結果ツリーの作成時にソートしなければいけません。

Result-tree fragments cannot be sorted (<xsl:sort> elements are ignored). You must sort the nodes when creating the result tree.

---

**KECX07024-W**

---

10 進数フォーマット設定 {0} はすでに定義されています。

Decimal formatting '{0}' is already defined.

---

**KECX07025-E**

---

XSL バージョン {0} は XSLTC によってサポートされていません。

XSL version '{0}' is not supported by XSLTC.

**KECX07026-E**

---

{0} 内の変数 / パラメタの参照が相互に呼び出し合っています。

Circular variable/parameter reference in '{0}'.

**KECX07027-E**

---

2 進数表記のための演算子が不明です。

Unknown operator for binary expression.

**KECX07028-E**

---

関数呼び出しの引数 ( 一つ以上 ) が正しくありません。

Illegal argument(s) for function call.

**KECX07029-E**

---

document() 関数への二つ目の引数はノード集合でなければいけません。

Second argument to document() function must be a node-set.

**KECX07030-E**

---

少なくとも一つの <xsl:when> 要素が <xsl:choose> 内に必要です。

At least one <xsl:when> element required in <xsl:choose>.

**KECX07031-E**

---

<xsl:choose> で許可されている <xsl:otherwise> 要素は一つだけです。

Only one <xsl:otherwise> element allowed in <xsl:choose>.

**KECX07032-E**

---

<xsl:otherwise> を使用できるのは <xsl:choose> 内だけです。

<xsl:otherwise> can only be used within <xsl:choose>.

**KECX07033-E**

---

<xsl:when> を使用できるのは <xsl:choose> 内だけです。

<xsl:when> can only be used within <xsl:choose>.

**KECX07034-E**

---

<xsl:choose> で許可されているのは <xsl:when> および <xsl:otherwise> 要素だけです。

Only <xsl:when> and <xsl:otherwise> elements allowed in <xsl:choose>.

### **KECX07035-E**

---

<xsl:attribute-set> に name 属性がありません。

<xsl:attribute-set> is missing the 'name' attribute.

### **KECX07036-E**

---

子要素が正しくありません。

Illegal child element.

### **KECX07037-W**

---

要素 {0} は呼び出せません。

You cannot call an element '{0}'

### **KECX07038-E**

---

属性 {0} は呼び出せません。

You cannot call an attribute '{0}'

### **KECX07039-E**

---

テキストが最上位の <xsl:stylesheet> 要素の外側にあります。

Text data outside of top-level <xsl:stylesheet> element.

### **KECX07040-E**

---

JAXP パーサは正しく構成されていません。

JAXP parser not configured correctly

### **KECX07041-E**

---

回復できない XSLTC の内部エラーが発生しました。{0}

Unrecoverable XSLTC-internal error: '{0}'

### **KECX07042-E**

---

XSL 要素 {0} はサポートされていません。

Unsupported XSL element '{0}'.

### **KECX07043-E**

---

XSLTC 拡張 {0} は認識されていません。

Unrecognised XSLTC extension '{0}'.



**KECX07044-E**

---

入力文書はスタイルシートではありません (XSL 名前空間はルート要素内で宣言されていません)。

The input document is not a stylesheet (the XSL namespace is not declared in the root element).

**KECX07045-E**

---

スタイルシート・ターゲット {0} が見つかりませんでした。

Could not find stylesheet target '{0}'.

**KECX07046-E**

---

{0} がインプリメントされていません。

Not implemented: '{0}'.

**KECX07047-E**

---

入力文書には XSL スタイルシートが入っていません。

The input document does not contain an XSL stylesheet.

**KECX07048-E**

---

要素 {0} を構文解析できませんでした。

Could not parse element '{0}'

**KECX07049-E**

---

<key> の use 属性は node , node-set , string , または number でなければいけません。

The use attribute of <key> must be node, node-set, string or number.

**KECX07050-E**

---

出力 XML 文書のバージョンは 1.0 である必要があります。

Output XML document version should be 1.0

**KECX07051-E**

---

関係式の演算子が不明です。

Unknown operator for relational expression

**KECX07052-E**

---

存在しない属性集合 {0} を使用しようとしています。

## 7. メッセージ

Attempting to use non-existing attribute set '{0}'.

### **KECX07053-E**

---

属性値テンプレート {0} を構文解析できません。

Cannot parse attribute value template '{0}'.

### **KECX07054-E**

---

クラス {0} のシグニチャ内のデータ型が不明です。

Unknown data-type in signature for class '{0}'.

### **KECX07055-E**

---

データ型 {0} を {1} に変換できません。

Cannot convert data-type '{0}' to '{1}'.

### **KECX07056-E**

---

このテンプレートには有効なトランスレットクラス定義が入っていません。

This Templates does not contain a valid translet class definition.

### **KECX07057-E**

---

このテンプレートには名前が {0} のクラスは入っていません。

This Templates does not contain a class with the name '{0}'.

### **KECX07058-E**

---

トランスレットクラス {0} をロードできませんでした。

Could not load the translet class '{0}'.

### **KECX07059-E**

---

トランスレットクラスがロードされましたが、トランスレットインスタンスを作成できません。

Translet class loaded, but unable to create translet instance.

### **KECX07060-E**

---

{0} の ErrorListener を null に設定しようとしています。

Attempting to set ErrorListener for '{0}' to null

### **KECX07061-E**

---

XSLTC がサポートしているのは StreamSource , SAXSource , および DOMSource だけです。

Only StreamSource, SAXSource and DOMSource are supported by XSLTC

---

**KECX07062-E**

---

{0} に渡されたソースオブジェクトにはコンテンツがありません。

Source object passed to '{0}' has no contents.

---

**KECX07063-E**

---

スタイルシートをコンパイルできませんでした。

Could not compile stylesheet

---

**KECX07064-E**

---

TransformerFactory は属性 {0} を認識しません。

TransformerFactory does not recognise attribute '{0}'.

---

**KECX07065-E**

---

setResult() は startDocument() の前に呼び出されていなければいけません。

setResult() must be called prior to startDocument().

---

**KECX07066-E**

---

変換プログラムにはカプセル化されたトランスレットオブジェクトがありません。

The Transformer has no encapsulated translet object.

---

**KECX07067-E**

---

変換結果の出力ハンドラが定義されていません。

No defined output handler for transformation result.

---

**KECX07068-E**

---

{0} に渡された Result オブジェクトが無効です。

Result object passed to '{0}' is invalid.

---

**KECX07069-E**

---

無効な Transformer プロパティ {0} にアクセスしようとしています。

Attempting to access invalid Transformer property '{0}'.

---

**KECX07070-E**

---

SAX2DOM アダプタ {0} を作成できませんでした。

## 7. メッセージ

Could not create SAX2DOM adapter: '{0}'.

### **KECX07071-E**

---

XSLTSource.build() が systemId を設定しないで呼び出されています。

XSLTSource.build() called without systemId being set.

### **KECX07072-E**

---

param {0} の値は有効な Java Object でなければいけません。

The value of param {0} must be a valid Java Object

### **KECX07073-E**

---

Result パラメタは null であってはいけません。

Result should not be null

### **KECX07075-E**

---

<xsl:sort> を使用できるのは <xsl:for-each> または <xsl:apply-templates> 内だけです。

<xsl:sort> can only be used within <xsl:for-each> or <xsl:apply-templates>.

### **KECX07076-W**

---

出力エンコード {0} はこの JVM ではサポートされていません。

Output encoding '{0}' is not supported on this JVM.

### **KECX07077-E**

---

{0} 内に構文エラーがあります。

Syntax error in '{0}'.

### **KECX07078-E**

---

外部コンストラクタ {0} が見つかりません。

Cannot find external constructor '{0}'.

### **KECX07079-E**

---

非 static Java 関数 {0} への先頭引数は有効なオブジェクト参照子ではありません。

The first argument to the non-static Java function '{0}' is not a valid object reference.

### **KECX07080-E**

---

式 {0} の型を検査中にエラーが発生しました。

Error checking type of the expression '{0}'.

---

**KECX07081-E**

---

不明なロケーションで式の型を検査中にエラーが発生しました。

Error checking type of an expression at an unknown location.

---

**KECX07082-E**

---

コマンド行オプション '{0}' が無効です。

The command-line option '{0}' is not valid.

---

**KECX07083-E**

---

コマンド行オプション '{0}' に必須属性がありません。

The command-line option '{0}' is missing a required argument.

---

**KECX07084-W**

---

警告: {0}, {1}

WARNING:'{0}' :'{1}'

---

**KECX07085-W**

---

警告: {0}

WARNING:'{0}'

---

**KECX07086-E**

---

致命的エラーが発生しました。{0}, {1}

FATAL ERROR:'{0}' :'{1}'

---

**KECX07087-E**

---

致命的エラーが発生しました。{0}

FATAL ERROR:'{0}'

---

**KECX07088-E**

---

エラーが発生しました。{0}, {1}

ERROR:'{0}' :'{1}'

---

**KECX07089-E**

---

エラーが発生しました。{0}

## 7. メッセージ

ERROR:'{0}'

### **KECX07090-E**

---

トランスレット '{0}' を使用する変換

Transform using translet '{0}'

### **KECX07091-E**

---

jar ファイル '{1}' からのトランスレット '{0}' を使用する変換

Transform using translet '{0}' from jar file '{1}'

### **KECX07092-E**

---

TransformerFactory クラス {0} のインスタンスを作成できませんでした。

Could not create an instance of the TransformerFactory class '{0}'.

### **KECX07093-E**

---

コンパイラ・エラーが発生しました。

Compiler errors:

### **KECX07094-W**

---

コンパイラ警告が発生しました。

Compiler warnings:

### **KECX07095-E**

---

トランスレットエラーが発生しました。

Translet errors:

### **KECX07096-E**

---

属性の値が {0} です。この値は QName , または空白で区切られた QName のリストでなければいけません。

An attribute whose value must be a QName or whitespace-separated list of QNames had the value '{0}'

### **KECX07097-E**

---

属性の値が {0} です。この値は NCName でなければいけません。

An attribute whose value must be an NCName had the value '{0}'

**KECX07098-E**

---

TransformerFactory.getFeature(String name) では、フィーチャー名は null にできません。

The feature name cannot be null in TransformerFactory.getFeature(String name).

**KECX07099-E**

---

TransformerFactory.setFeature(String name, boolean value) では、フィーチャー名は null にできません。

The feature name cannot be null in TransformerFactory.setFeature(String name, boolean value).

**KECX07100-E**

---

この TransformerFactory にはフィーチャー {0} を設定できません。

Cannot set the feature '{0}' on this TransformerFactory.

**KECX07101-E**

---

<xsl:output> 要素の method 属性は値 {0} を持っていました。指定できる値は 'xml','html','text' または qname-but-not-ncname でなければいけません。

The method attribute of an <xsl:output> element had the value '{0}'. The value must be one of 'xml', 'html', 'text', or qname-but-not-ncname

**KECX07203-E**

---

クラスをロードできませんでした。

Couldn't load the class

**KECX07204-E**

---

メソッドが見つかりません。

Method not found

**KECX07205-E**

---

メソッドにアクセスできません。

Error: Method is inaccessible

**KECX07206-E**

---

仮引数と実引数の数が一致しません。

Error: Number of actual and formal argument differ

### **KECX07207-E**

---

基底コンストラクタが例外をスローしました。

Error: underlying constructor throws an exception

### **KECX07208-E**

---

コンストラクタが見つかりません。

Constructor not found

### **KECX07209-E**

---

コンストラクタが例外をスローしました。

Error: constructor throws an exception

### **KECX07210-E**

---

コンストラクタにアクセスできません。

Error: constructor is inaccessible

### **KECX07211-E**

---

基底コンストラクタを宣言するクラスは抽象クラスを表しています。

Error: Class that declares the underlying constructor represents an abstract class

### **KECX07212-E**

---

CONSTRUCTOR\_NOT\_FOUND\_ERR クラス名：メソッド名

CONSTRUCTOR\_NOT\_FOUND\_ERR {0} : {1}

### **KECX07213-E**

---

METHOD\_NOT\_FOUND\_ERR クラス名：メソッド名

METHOD\_NOT\_FOUND\_ERR {0} : {1}

### **KECX07301-E**

---

{0} で実行時内部エラーが発生しました。

Run-time internal error in '{0}'

### **KECX07302-E**

---

<xsl:copy> を実行中にエラーが発生しました。

Run-time error when executing <xsl:copy>.



**KECX07303-E**

---

{0} から {1} への変換が無効です。

Invalid conversion from '{0}' to '{1}'.

**KECX07304-E**

---

外部関数 {0} は XSLTC によってサポートされていません。

External function '{0}' not supported by XSLTC.

**KECX07305-E**

---

等式内の引数の型が不明です。

Unknown argument type in equality expression.

**KECX07306-E**

---

{1} を呼び出し中の引数型 {0} が無効です。

Invalid argument type '{0}' in call to '{1}'

**KECX07307-E**

---

数値 {0} をパターン {1} を使用してフォーマット設定しようとしています。

Attempting to format number '{0}' using pattern '{1}'.

**KECX07308-E**

---

イテレータ {0} を複製できません。

Cannot clone iterator '{0}'.

**KECX07309-E**

---

軸 {0} のイテレータはサポートされていません。

Iterator for axis '{0}' not supported.

**KECX07310-E**

---

型付きの軸 {0} のイテレータはサポートされていません。

Iterator for typed axis '{0}' not supported.

**KECX07311-E**

---

属性 {0} が要素の外側です。

Attribute '{0}' outside of element.

---

**KECX07312-E**

---

名前空間宣言 {0}={1} が要素の外側です。

Namespace declaration '{0}'='{1}' outside of element.

---

**KECX07313-E**

---

接頭辞 {0} の名前空間が宣言されていません。

Namespace for prefix '{0}' has not been declared.

---

**KECX07314-E**

---

DOMAdapter が間違っただ型のソース DOM を使用して作成されました。

DOMAdapter created using wrong type of source DOM.

---

**KECX07315-E**

---

使用中の SAX パーサは DTD 宣言イベントを処理しません。

The SAX parser you are using does not handle DTD declaration events.

---

**KECX07316-E**

---

使用中の SAX パーサには XML 名前空間のサポートがありません。

The SAX parser you are using does not have support for XML Namespaces.

---

**KECX07317-E**

---

URI 参照 {0} を解決できませんでした。

Could not resolve the URI reference '{0}'.

---

**KECX07318-E**

---

XSL 要素 {0} はサポートしていません。

Unsupported XSL element '{0}'

---

**KECX07319-E**

---

XSLTC 拡張 {0} は認識されません。

Unrecognized XSLTC extension '{0}'

---

**KECX07320-E**

---

指定されたトランスレット {0} は、使用されている XSLTC ランタイムのバージョンより最近の XSLTC のバージョンを使用して作成されました。stylesheet を再コンパイルするか、またはこ

のトランスレットを実行するために XSLTC のより最近のバージョンを使用しなければいけません。

The specified translet, '{0}', was created using a version of XSLTC more recent than the version of the XSLTC run-time that is in use. You must recompile the stylesheet or use a more recent version of XSLTC to run this translet.

---

#### **KECX07321-E**

QName であるはずの属性が値 {0} を持っていました。

An attribute whose value must be a QName had the value '{0}'

---

#### **KECX07322-E**

NCName であるはずの属性が値 {0} を持っていました。

An attribute whose value must be an NCName had the value '{0}'

---

#### **KECX07323-E**

secure processing フィーチャーが true に設定されている場合は、拡張関数 {0} の使用は許可されません。

Use of the extension function '{0}' is not allowed when the secure processing feature is set to true.

---

#### **KECX07324-E**

secure processing フィーチャーが true に設定されている場合は、拡張要素 {0} の使用は許可されません。

Use of the extension element '{0}' is not allowed when the secure processing feature is set to true.

---

#### **KECX07501-E**

URL を絶対化できません : {0}

Can not absolutize URL: {0}

---

#### **KECX07503-E**

com.cosminexus.jaxp.impl.transform.xslt.trax.SmartTransformerFactoryImpl は com.cosminexus.jaxp.impl.transform.processor.TransformerFactoryImpl を作成できませんでした。

com.cosminexus.jaxp.impl.transform.xslt.trax.SmartTransformerFactoryImpl could not create an com.cosminexus.jaxp.impl.transform.processor.TransformerFactoryImpl.

---

#### **KECX07701-E**

ファイル名を -stdin オプションで提供する必要があります。

## 7. メッセージ

You must supply a filename with the -stdin option.

### **KECX07702-E**

---

ファイル {0} を開けませんでした。

Could not open file {0}

### **KECX07703-E**

---

ファイル {0} に対する権限がありません。

No permission to file {0}

### **KECX07704-E**

---

-stdin オプションがありません。

Missing -stdin option!

### **KECX07705-E**

---

JavaCUP 実行エラー :

Error running JavaCUP:

## 7.2.7 KECX09000 ~ KECX09999

### **KECX09601-E**

---

事前解析時にエラーが発生しました。{0}

An error occurred while pre-parsing the document : {0}

### **KECX09602-E**

---

XML ファイルには null 以外を指定してください。

XML file cannot be null.

### **KECX09603-E**

---

チューニング情報でエラーが発生しました。{0}

Error occurs in Tuning Information : {0}

### **KECX09604-E**

---

ファイル {0} にアクセスできません。

Cannot access to file : {0}

### **KECX09605-E**

---

高速パース機能の一時ディレクトリで入出力エラーが発生しました。{0}

I/O Error occurs in temporary directory of fast parse function : {0}

### **KECX09606-E**

---

preparedobject-load プロパティの値が不正です。

Illegal value for preparedobject-load property.

### **KECX09607-E**

---

カスタムスキャナのロードでエラーが発生しました。{0}

Error occurs in custom scanner loading : {0}

### **KECX09608-E**

---

XML 入カストリームには null 以外を指定してください。

XML InputStream cannot be null.



# 付録

---

付録 A バージョン間の差異

---

付録 B 用語解説

---

## 付録 A バージョン間の差異

Cosminexus XML 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。

### 付録 A.1 DOM パーサ，および SAX パーサ共通の動作の差異

DOM パーサ，および SAX パーサ共通の動作の差異を表 A-1 に示します。

表 A-1 DOM パーサ，および SAX パーサ共通の動作の差異

項番	動作の差異
1	パーサが <code>InputSource</code> オブジェクトを使用してバイトストリームを読み込むとき，06-00 では XML 仕様のアルゴリズムを用いて文字エンコーディングを自動検出しますが，07-00 以降では <code>setEncoding</code> メソッドで指定されたエンコーディングを優先します。
2	「EUC で保存されているが XML 宣言に <code>encoding</code> 指定が存在しない XML 文書」など，エンコーディングが不正な状態で解析したとき，06-00 では <code>IOException</code> 例外が発生しますが，07-00 以降では <code>SAXException</code> 例外が発生します。
3	<code>parse</code> メソッドの引数 <code>uri</code> または引数 <code>systemId</code> に空文字列を指定したとき，06-00 では <code>IOException</code> 例外が発生しますが，07-00 以降では <code>SAXException</code> 例外が発生します。
4	XML 文書の DOCTYPE 宣言のシステム識別子に空文字列を指定したとき，06-00 では <code>IOException</code> 例外が発生しますが，07-00 以降では <code>SAXException</code> 例外が発生します。
5	DTD で <code>EMPTY</code> と宣言した要素の内容がコメントだけのとき，06-00 では妥当性検証時にエラーになりませんが，07-00 以降ではエラーになります。

### 付録 A.2 DOM パーサの動作の差異

DOM パーサの動作の差異を表 A-2 に示します。

表 A-2 DOM パーサの動作の差異

項番	動作の差異
1	同一名の NOTATION 宣言があり，かつ，妥当性検証が有効なとき，06-00 では重複した NOTATION 宣言を破棄するだけですが，07-00 ではエラーになります。
2	あるノードに対して，そのノードもしくは祖先ノードを引数 <code>newChild</code> として <code>appendChild</code> メソッドまたは <code>insertBefore</code> メソッドを実行したとき，発生する <code>DOMException</code> 例外のコードが異なります。06-00 では <code>WRONG_DOCUMENT_ERR</code> ですが，07-00 以降では <code>HIERARCHY_REQUEST_ERR</code> です。
3	Document インタフェースの <code>createElementNS(String namespaceURI, String qualifiedName)</code> メソッドの引数 <code>qualifiedName</code> に <code>null</code> を指定したとき，発生する <code>DOMException</code> 例外のコードが異なります。06-00 では <code>INVALID_CHARACTER_ERR</code> ですが，07-00 以降では <code>NAMESPACE_ERR</code> です。



## 付録 A.3 SAX パーサの動作の差異

SAX パーサの動作の差異を表 A-3 に示します。

表 A-3 SAX パーサの動作の差異

項番	動作の差異
1	XMLReaderFactory クラスの createXMLReader() メソッドは、06-00 ではシステムプロパティ "org.xml.sax.driver" の値を用いて XMLReader インスタンスの生成を試みますが、07-00 以降では以下の順序で XMLReader インスタンスの生成を試みます。 1. システムプロパティ "org.xml.sax.driver" が示すクラス名 2. jarfile 内の META-INF/services/org.xml.sax.driver が示すクラス名 3. デフォルトの XMLReader クラス名 4. ParserFactory クラスの makeParser() メソッドの戻り値を ParserAdapter でラップしたクラス
2	XMLReader オブジェクトに ContentHandler オブジェクトを設定したとき、06-00 ではエンティティをスキップしても ContentHandler オブジェクトの skippedEntity メソッドは呼び出されませんが、07-00 以降では呼び出されます。
3	DeclHandler クラスの externalEntityDecl(String name, String publicId, String systemId) メソッド、および、DTDHandler インタフェースの notationDecl(String name, String publicId, String systemId) メソッドの引数 systemId が URL のとき、06-00 では URL が完全に解決されない場合がありますが、07-00 以降では URL が完全に解決されます。
4	XMLReader クラスの setEntityResolver メソッド、setDTDHandler メソッド、setContentHandler メソッド、setErrorHandler メソッドの引数に null を指定したとき、06-00 では NullPointerException 例外が発生しますが、07-00 以降ではリゾルバやハンドラがクリアされます。
5	ParserAdapter クラスの setEntityResolver メソッド、setDTDHandler メソッド、setContentHandler メソッド、setErrorHandler メソッドの引数に null を指定したとき、06-00 では NullPointerException 例外が発生しますが、07-00 以降ではリゾルバやハンドラがクリアされます。
6	XMLFilterImpl クラスの setEntityResolver メソッド、setDTDHandler メソッド、setContentHandler メソッド、setErrorHandler メソッド、setParent メソッドの引数に null を指定したとき、06-00 では NullPointerException 例外が発生しますが、07-00 以降ではリゾルバやハンドラや親リーダがクリアされます。
7	SAX パーサのフィーチャー "http://xml.org/sax/features/namespaces" を false に設定し、かつ、フィーチャー "http://xml.org/sax/features/namespace-prefixes" を true に設定したとき、接頭辞 "xml","xmlns" に対する startPrefixMapping メソッド、endPrefixMapping メソッドの動作が異なります。06-00 ではこれらのメソッドが呼び出されますが、07-00 以降では呼び出されません。

## 付録 A.4 スキーマ検証の動作の差異

スキーマ検証の動作の差異を表 A-4 に示します。

表 A-4 スキーマ検証の動作の差異

項番	動作の差異
1	スキーマ文書に <code>xsd:key</code> 要素が存在し、かつ、インスタンス文書にキーに該当するテキストが存在しないとき、06-00 ではエラーにならない場合がありますが、07-00 以降ではエラーになります。
2	スキーマ文書に <code>xsd:key</code> 要素と <code>xsd:keyref</code> 要素の組み合わせを指定していて、 <code>xsd:key</code> 要素と <code>xsd:keyref</code> 要素の双方の <code>field</code> 要素の順序と型が一致していないとき、06-00 ではエラーにならない場合がありますが、07-00 以降ではエラーになります。
3	<code>xsd:field</code> 要素の <code>xpath</code> 属性が <code>child</code> 軸を含む XPath 式のと看、06-00 ではエラーになりませんが、07-00 以降では <code>child</code> 軸が指定できない旨のエラーになります。
4	複合型を基準型として制限した複合型では、制限した複合型で制限している要素が基準型に存在しないとき、06-00 ではエラーになりませんが、07-00 以降ではスキーマ文書のエラーになります。
5	<code>xsd:selector</code> 要素の <code>xpath</code> 属性が「 <code>  ns1:element</code> 」のように不当な <code> </code> 演算子を含む XPath 式のと看、06-00 ではエラーにならないか、または、XML 文書の妥当性検証のエラーになりますが、07-00 以降ではスキーマ文書のエラーになります。
6	<code>xsd:redefine</code> 要素または <code>xsd:import</code> 要素の <code>schemaLocation</code> 属性に空文字列を指定したとき、06-00 ではエラーになりませんが、07-00 以降ではエラーになります。

## 付録 A.5 XSLT の動作の差異

XSLT の動作の差異を表 A-5 に示します。

表 A-5 XSLT の動作の差異

項番	動作の差異
1	Transform クラスおよび TransformerFactory クラスのデフォルトのエラーリスナーは、06-00 ではエラー時に例外をスローしますが、07-00 以降ではエラー時に例外をスローしません。
2	TransformerFactory クラスの <code>newTransformer(Source source)</code> メソッド、 <code>newTemplates</code> メソッドを実行したとき、これらのメソッドの入力であるスタイルシートに誤りがあると、 <code>ErrorListener</code> の <code>fatalError</code> メソッドにエラーが報告されます。このとき、 <code>ErrorListener</code> の <code>fatalError</code> メソッドの実装が <code>TransformerException</code> をスローしない場合は、 <code>newTransformer(Source source)</code> メソッド、 <code>newTemplates</code> メソッドの戻り値が <code>null</code> になる場合がありますのでご注意ください。デフォルトの <code>ErrorListener</code> は、 <code>TransformerException</code> をスローしません。したがって、回復できないエラー発生時の正常な動作を確保するため、 <code>ErrorListener</code> を登録して <code>fatalError</code> メソッドの中から <code>TransformerException</code> 例外をスローすることを強くお勧めします。
3	XML をファイルに出力する際、 <code>xsl:output</code> 要素に <code>indent="yes"</code> が指定されていないか、または、Transform クラスの <code>setOutputProperty(OutputKeys.INDENT, "yes")</code> メソッドが実行されていないとき、出力結果が異なります。06-00 では XML 宣言の直後に改行が出力されますが、07-00 以降では出力されません。

項番	動作の差異
4	変換結果を HTML 形式で出力した場合、06-00 と 07-00 以降とでは要素間の改行出力に差異があります。ただし、HTML 文書では、要素間の改行は意味を持たないため問題ありません。
5	system-property('xsl:version') 関数を実行したとき、06-00 では結果は "1" になりますが、07-00 以降では "1.0" になります。
6	xsl:attribute 要素の内容がタブを含む文字列のとき、06-00 では変換結果の属性値がタブ文字そのものになりますが、07-00 以降では文字参照 (&#9;) になります。
7	xsl:number 要素の value 属性に generate-id() 関数を使用したとき、06-00 では "0" が出力されますが、07-00 以降では "NaN" が出力されます。
8	xsl:stylesheet 要素の名前空間宣言で定義した接頭辞が、xsl:element 要素によって作成された要素の中で使われないとき、06-00 ではその名前空間宣言が xsl:element 要素に出力されますが、07-00 以降では出力されません。
9	xsl:version 属性に "1.1" を指定した簡略化スタイルシートの中にトップレベル要素を記述したとき、前方互換処理の動作が異なります。06-00 ではトップレベル要素がインスタンス化されて変換が実行されますが、07-00 以降ではトップレベル要素が無視されます。
10	xsl:attribute-set 要素の use-attribute-sets 属性に「存在しない属性セット名」を指定したとき、06-00 ではエラーになりませんが、07-00 以降では属性セットが存在しない旨のエラーになります。
11	xsl:template 要素に name 属性、match 属性のどちらも指定されていない場合、06-00 ではエラーになりませんが、07-00 以降では name 属性か match 属性のどちらかが必要な旨のエラーになります。
12	exclude-result-prefixes 属性で指定された「除外する名前空間の接頭辞」が、xsl:stylesheet 要素の名前空間宣言で定義されていないとき、06-00 ではエラーになりませんが、07-00 以降ではエラーになります。
13	xsl:stylesheet 要素の version 属性に数値以外を指定したとき、06-00 ではエラーになりませんが、07-00 以降ではエラーになります。
14	xsl:output 要素の method 属性に "xml" を指定し、かつ、xsl:output 要素の version 属性に "1.0"、"1.1" 以外を指定したとき、06-00 ではワーニングは発生しませんが、07-00 以降では "1.0" を仮定する旨のワーニングが発生します。
15	テキスト出力で、出力時のエンコーディングでは表現できない文字を出力するとき、06-00 では文字化けとなりますが、07-00 以降では標準エラー出力にワーニングが出力され、その文字は文字参照として出力されます。
16	Transform クラスの transform(Source xmlSource, Result outputTarget) メソッドの引数 xmlSource に空の Source を指定した場合、06-00 ではエラーになりますが、07-00 以降では DocumentBuilder クラスの newDocument メソッドで生成された空のドキュメントが使われます。
17	TransformerFactory クラスの getFeature メソッドの引数に null を指定した場合、06-00 では false が戻りますが、07-00 以降では NullPointerException 例外が発生します。
18	名前空間宣言のない接頭辞があるとき、06-00 では TransformerException 例外が発生しますが、07-00 以降では XPathStylesheetDOM3Exception 例外が発生する場合があります。この例外は TransformerException 例外のサブクラスなので、TransformerException によりキャッチできます。

項番	動作の差異
19	StreamResult クラスの StreamResult(File f) コンストラクタ、および、setSystemId(File f) メソッドの引数 f に null を指定したとき、06-00 では NullPointerException 例外が発生しますが、07-00 以降では RuntimeException 例外が発生します。
20	xsl:call-template 要素の name 属性で指定した名前付きテンプレートが存在しない場合、06-00 ではエラー KECX02019-E を通知していましたが、07-00 ではエラー KECX02019-E に加えてエラー KECX02015-E を通知するようになります。
21	「namespace:: 局所名」を含む XPath 式の評価結果は、06-00 と 07-00 以降で差異があります。06-00 では、局所名を接頭辞と見なして名前空間 URI を解決し、その名前空間 URI に一致する文字列値を持つ名前空間ノードが評価結果になります。07-00 では、指定した局所名を持つ名前空間ノードが評価結果になります。

---

## 付録 B 用語解説

### (英字)

---

#### DOM

Document Object Model の略です。DOM パーサによって解析された XML 文書の内容を記述したもので、メモリに常駐するツリー形式のデータ構造となっています。このツリー構造を DOM ツリーと呼びます。

#### DOM API

DOM パーサによって生成された DOM にアクセスするための API を DOM API と呼びます。XML 文書内のデータ構造に基づいたデータの処理ができます。

#### DOM ツリー

XML 文書の構造をツリー形式で表したものです。DOM ツリーは DOM パーサを使って解析すると生成されます。

#### DTD

Document Type Definition の略です。XML のスキーマの一つで、XML 文書で使用するタグの属性や階層構造や出現順序などを定義したものです。XML 文書はこの DTD で定義した文法に従って記述します。

#### IANA

Internet Assigned Numbers Authority の略です。インターネットの名前または番号の登録業務を行う組織です。

#### 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、SAX パーサを生成する API が含まれています。パーサ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 トランスフォーマを使用することができます。

#### Namespaces in XML

XML で定義された複数の言語を一つの文書で扱う場合に、要素や属性の名前が衝突することを避けるために用いる修飾子。

#### SAX

Simple API for XML の略です。XML 文書内の構文要素を逐次、通知することで XML 文書の内容にアクセスするためのコールバック API です。SAX の場合、解析中の個所に関する情報だけをユーザプログラムに渡し、DOM のようにツリーを作成しません。このため、DOM に比べてメモリ所要量が少なくなります。

## TrAX

Transformation API for XML の略です。JAXP の仕様に含まれる API で、XSLT トランスフォーマの標準 API です。

TrAX を使用することで Java プログラムから XML 変換処理を実現できます。単純な XML 文書の変換であれば TrAX だけで実現できます。

## URI

Uniform Resource Identifier の略です。URL のように Web 上のデータの位置を指定する方法と、Web 上の特定のデータを固有の名前で指定する方法を持ち合わせた規格です。

## URL

Uniform Resource Locator の略です。Web 上にある特定のデータの位置を指定する規格です。

## W3C

World Wide Web Consortium の略です。XML や DOM を含む Web 関連技術の標準化を推進する非営利団体です。

## XML

Extensible Markup Language の略です。W3C が制定したマークアップ言語で、インターネット上でデータのやり取りを実現します。このマニュアルでは、XML 形式で記述された文書を XML 文書と呼びます。

## XML Schema

W3C 勧告で定められた仕様で、XML 文書内の論理的なデータ構造を規定するための仕組みです。

## XML 仕様

W3C 勧告の「XML 1.0」および「XML 1.1」で定められた仕様です。Cosminexus XML Processor の XML パーサは、XML 仕様に従った XML 文書进行处理できます。

## XML パーサ

XML 文書を解析するプログラムです。

パーサには、DOM API を使用する DOM パーサと、SAX を使用する SAX パーサの 2 種類があります。DOM パーサは、XML 文書の解析結果を DOM ツリーとして出力します。SAX パーサは、DOM ツリーを出力しない代わりに、解析結果をイベントとして SAX に通知します。

このマニュアルではこれらのパーサを総称して XML パーサと呼びます。

## XML 文書

XML 形式で記述された文書です。

## XPath

XML 文書の特定の部分を指し示す構文を規定する。XPath を使用すれば XML 文書中にアンカーを指定しなくても、文書中の任意の位置を指定できます。

## XSL

Extensible Stylesheet Language の略です。XML 用のスタイルシートのことで、表示方法を記述するだけでなく、データのコピー、分岐、抽出などのデータ処理を記述することができます。

## XSLT

XSL Transformations の略です。XML 文書をスタイルシートに従った XML 文書や HTML、テキストに変換するなどの、XML 文書の構造変換を行うための仕様です。

## XSLTC トランスフォーマ

XSLTC トランスフォーマは、Cosminexus XML Processor が提供している XSLT トランスフォーマと同じ機能を持ちながら、XML 文書の変換処理時の性能を向上させたものです。XSLT スタイルシートを解析するコンパイラと変換処理を行う実行時プロセッサで構成されます。

## XSLT トランスフォーマ

XSLT を実装したプログラムです。XML 文書をスタイルシートに従った XML 文書や HTML、テキストに変換することができます。

## (ア行)

---

### エンティティ

XML 文書中で使用される画像データや PDF ファイルなどのファイルや、置換する文字列などのデータを保持する実体です。

## (サ行)

---

### スキーマ文書

XML Schema の仕様に従って記述された、XML 文書内の論理的なデータ構造を規定した XML 文書です。

## (タ行)

---

### データ要素

XML 文書の構成要素となるデータです。タグに挟まれた文字列やタグ属性の値を指します。

### トランスレット

XSLTC トランスフォーマを構成するコンパイラが XSLT スタイルシートを解析してメモリ上に生成する、Java のバイトコードです。XML 文書の変換処理を高速に実行できます。





---

# 索引

## C

---

Cosminexus XML Processor

位置づけ 4

概要 1

サポート範囲 7

使用時の注意事項 119

処理の流れ 5

特長 2

バージョン間の差異 324

Cosminexus XML Processor が使用している  
パッケージ名 88

Cosminexus XML Processor が提供する

JAXP の機能 12

Cosminexus XML Processor 使用時の注意事  
項 119

Cosminexus 実行環境での Cosminexus XML  
Processor の位置づけ 4

## D

---

DocumentBuilder に解析結果オブジェクト  
を設定する方法 82

DOM 329

DOM API 329

DOM ツリー 329

DOM ツリーを操作するアプリケーションの  
処理の流れ 21

DOM パーサ, および SAX パーサ共通の動  
作の差異 324

DOM パーサで Shift\_JIS 切り替え機能を使  
用する方法 80

DOM パーサでの解析を行うアプリケーショ  
ンの処理の流れ 14

DOM パーサに関する注意事項 122

DOM パーサの動作の差異 324

DOM パーサを使用するサンプルプログラム  
90

DOM パーサを使用するサンプルプログラムの  
作成の流れ 90

DOM パーサを使用する場合のサンプルプロ  
グラム 108

DTD 329

## I

---

IANA 329

## J

---

javax.xml.datatype パッケージ 20

javax.xml.datatype パッケージに関する注意  
事項 138

javax.xml.namespace パッケージ 19

javax.xml.parsers パッケージ 14

javax.xml.transform.dom パッケージ 16

javax.xml.transform.sax パッケージ 17

javax.xml.transform.stream パッケージ 17

javax.xml.transform パッケージ 16

javax.xml.validation パッケージ 18

javax.xml.validation パッケージに関する注  
意事項 139

javax.xml.validation パッケージに含まれる  
クラスを用いて XML 文書を検証するアプリ  
ケーションの処理の流れ 18

javax.xml.xpath パッケージ 19

javax.xml.xpath パッケージに関する注意事  
項 141

javax.xml パッケージ 20

JAXP 329

JAXP1.3 機能の共通の注意事項 121

JAXP が規定するパッケージ 13

javax.xml 20

javax.xml.datatype 20

javax.xml.namespace 19

javax.xml.parsers 14

javax.xml.transform 16

javax.xml.transform.dom 16

javax.xml.transform.sax 17

javax.xml.transform.stream 17

javax.xml.validation 18

javax.xml.xpath 19

- org.w3c.dom 21
- org.w3c.dom.bootstrap 21
- org.w3c.dom.events 23
- org.w3c.dom.ls 22
- org.xml.sax 23
- org.xml.sax.ext 24
- org.xml.sax.helpers 24

JAXP が規定するパッケージに含まれる API の概要 13

JAXP で XML 文書を解析する流れ 3

JAXP で XML 文書を変換する流れ 3

JAXP で定義された仕様 7

JAXP とは 12

## M

---

MS932 9

## N

---

Namespaces in XML 329

## O

---

org.w3c.dom.bootstrap パッケージ 21  
org.w3c.dom.bootstrap パッケージに関する  
注意事項 145

org.w3c.dom.events パッケージ 23

org.w3c.dom.ls パッケージ 22

org.w3c.dom.ls パッケージに関する注意事項  
146

org.w3c.dom パッケージ 21

org.w3c.dom パッケージに関する注意事項  
143

org.xml.sax.ext パッケージ 24

org.xml.sax.ext パッケージに関する注意事  
項 148

org.xml.sax.helpers パッケージ 24

org.xml.sax パッケージ 23

## P

---

PreparsedObjectFactory クラス 46

PreparsedObject クラス 50

## S

---

SAX 329

SAX2 のフィーチャー一覧 72

SAX2 のフィーチャーおよびプロパティの使  
用方法 72

SAX2 のフィーチャーの使用方法 73

SAX2 のプロパティ一覧 72

SAX2 のプロパティの使用方法 73

SAXParser に解析結果オブジェクトを設定  
する方法 83

SAX イベントを処理するアプリケーション  
の処理の流れ 23

SAX パーサで Shift\_JIS 切り替え機能を使用  
する方法 80

SAX パーサでの解析を行うアプリケーショ  
ンの処理の流れ 15

SAX パーサに関する注意事項 124

SAX パーサの動作の差異 325

SAX パーサを使用するサンプルプログラム  
95

SAX パーサを使用するサンプルプログラム  
の作成の流れ 95

SAX パーサを使用する場合のサンプルプロ  
グラム 110

Shift\_JIS 切り替え機能 9, 26

Shift\_JIS 切り替え機能の設定用メソッド 79

Shift\_JIS 切り替え機能のプロパティの使用  
方法 79

Shift\_JIS 切り替え機能を使用する方法

DOM パーサ 80

SAX パーサ 80

XSLT トランスフォーマ 81

SJIS 9

## T

---

TransformerFactoryXSLTC クラス 29

TrAX 330

## U

---

URI 330

URL 330

## W

---

W3C 330

## X

---

XInclude に関する注意事項 149

XML 330

XMLReader に解析結果オブジェクトを設定  
する方法 84

XML Schema 330

XML Schema のエラーメッセージと W3C 仕  
様との対応 70

XML Schema のプロパティの使用法 75

XML Schema のプロパティの設定方法

DOM パーサ 75

SAX パーサ 76

XML Schema を使用するサンプルプログラ  
ム 101

XML Schema を使用するサンプルプログラ  
ムの処理の流れ 101

XML 仕様 330

XML パーサ 330

XML パーサ機能および XSLT トランス  
フォーマ機能の処理の流れ 5

XML パーサ機能の処理 6

XML 文書 330

XML 文書 (purchaseOrder-fail.xml) 103

XML 文書 (purchaseOrder.xml) 102

XML 文書の encoding 指定と適用される文字  
エンコーディングとの対応 26

XML 文書の解析 45

XML 文書の解析 (高速化定義ファイルを使  
用する場合) 68

XML 文書のロード, および保存を行うアプ  
リケーションの処理の流れ 22

XPath 330

XPath 式を評価するアプリケーションの処理  
の流れ 19

XSL 330

XSLT 331

XSLT/XSLTC 共通の注意事項 127

XSLTC トランスフォーマ 331

XSLTC トランスフォーマ機能 28

概要 28

システムの開発・運用 29

使用するクラス 29

使用方法 30

XSLTC トランスフォーマ機能を使用する場  
合のプログラム変更 118

XSLTC トランスフォーマを使用するサンプ  
ルプログラム 118

XSLTC に関する注意事項 131

XPath 式 134

XSLT 要素 131

エラーを通知しないケース 135

スタイルシートサイズ 131

その他 137

変換処理性能 131

XSLT トランスフォーマ 331

XSLT トランスフォーマ機能の処理 6

XSLT トランスフォーマで Shift\_JIS 切り替  
え機能を使用する方法 81

XSLT トランスフォーマを使用するサンプ  
ルプログラム 113

XSLT トランスフォーマを使用するサンプ  
ルプログラムの作成の流れ 114

XSLT に関する注意事項 130

エラーを通知しないケース 130

その他 130

XSLT による変換を行うアプリケーションの  
処理の流れ 16

XSLT の動作の差異 326

XSLT のフィーチャー一覧 74

XSLT のフィーチャーの使用法 74

## え

---

エンティティ 331

## か

---

解析結果オブジェクト 32

解析結果オブジェクトの生成 44

解析結果オブジェクトの設定 45

解析結果オブジェクトの設定用メソッド 82

解析結果オブジェクトを設定する方法

DocumentBuilder 82

SAXParser 83  
XMLReader 84

## こ

---

高速化定義ファイルの作成 63  
高速化定義ファイルの設定 65  
高速化定義ファイルを使用した高速パース 60  
高速化定義ファイルを使用した高速パースのための作業の流れ 62  
高速化定義ファイルを使用するためのシステムプロパティ 65  
高速パース機能 32  
高速パース機能で使用するクラス 46  
高速パース機能に関する注意事項 152  
    XInclude 152  
    XML1.1 152  
    解析結果オブジェクトの設定 152  
    解析結果オブジェクトを利用する XML パーサ 152  
    解析性能 154  
    高速化定義ファイル 153  
    事前解析用 XML 文書 153  
    絶対パスの指定方法 153  
    チューニング情報 154  
高速パース機能の概要 32  
高速パース機能の対象となる parse メソッド 45  
高速パース機能の対象となる parse メソッド (高速化定義ファイルを使用する場合) 69  
高速パース機能のプロパティの使用方法 82  
高速パース機能を使用するためのコード例 51  
高速パースのための作業の流れ 34

## さ

---

サンプルプログラム  
    DOM パーサを使用する 90  
    SAX パーサを使用する 95  
    XML Schema を使用する 101  
    XML Schema を使用する (DOM パーサ) 108

XML Schema を使用する (SAX パーサ) 110  
XSLTC トランスフォーマを使用する 118  
XSLT トランスフォーマを使用する 113

## し

---

事前解析の実行タイミング 66  
事前解析の実行タイミングの設定 65  
事前解析の実行タイミングの設定方法 67  
事前解析の実行タイミングを設定するためのシステムプロパティ 68  
事前解析用 XML 文書 33  
事前解析用 XML 文書の作成 36  
事前解析用 XML 文書のチューニング 52  
事前解析用 XML 文書を作成するための指針 36  
実装依存仕様に関する注意事項 150  
使用できる XML パーサ, XSLT トランスフォーマなどの機能 7  
処理できる文字コード 8  
処理の流れ  
    DOM ツリーを操作するアプリケーション 21  
    DOM パーサでの解析を行うアプリケーション 14  
    javax.xml.validation パッケージに含まれるクラスを用いて XML 文書を検証するアプリケーション 18  
    SAX イベントを処理するアプリケーション 23  
    SAX パーサでの解析を行うアプリケーション 15  
    XML 文書のロード, および保存を行うアプリケーション 22  
    XPath 式を評価するアプリケーション 19  
    XSLT による変換を行うアプリケーション 16  
    日付 / 時刻型データを処理するアプリケーション 20

## す

---

スキーマ検証に関する注意事項 126  
 スキーマ検証の動作の差異 326  
 スキーマ文書 331  
 スキーマ文書 (personalData.xsd) 107  
 スキーマ文書 (purchaseOrder.xsd) 104  
 スキーマ文書を XML 文書内で設定する方法 77

## せ

---

設定するプロパティの値とその意味  
 (Shift\_JIS 切り替え機能) 79  
 設定するプロパティの値とその意味 (高速  
 パース機能) 82

## ち

---

### 注意事項

DOM パーサ 122  
 javax.xml.datatype パッケージ 138  
 javax.xml.validation パッケージ 139  
 javax.xml.xpath パッケージ 141  
 JAXP1.3 機能 121  
 org.w3c.dom.bootstrap パッケージ 145  
 org.w3c.dom.ls パッケージ 146  
 org.w3c.dom パッケージ 143  
 org.xml.sax.ext パッケージ 148  
 SAX パーサ 124  
 XInclude 149  
 XSLT 130  
 XSLT/XSLTC 共通 127  
 XSLTC 131  
 高速パース機能 152  
 実装依存仕様 150  
 スキーマ検証 126  
 チューニングサマリファイル 53  
 チューニングサマリファイルの詳細 56  
 チューニング詳細ファイル 53  
 チューニング詳細ファイルの詳細 57  
 チューニング情報の出力先〔高速パース機  
 能〕55

チューニング情報の出力方法〔高速パース機  
 能〕54  
 チューニング情報の種類〔高速パース機能〕  
 52  
 チューニング情報の利用方法〔高速パース機  
 能〕53  
 チューニング情報を出力するためのシステム  
 プロパティ〔高速パース機能〕55

## つ

---

通常的高速パースと高速化定義ファイルを使  
 用した高速パースの相違点 61

## て

---

データ要素 331

## と

---

特定するスキーマ文書の種類と使用する属性  
 の関係 77  
 トラブルシュート 89  
 トランスレット 331

## は

---

### バージョン間の差異

DOM パーサ, および SAX パーサ共  
 通の動作の差異 324  
 DOM パーサの動作の差異 324  
 SAX パーサの動作の差異 325  
 XSLT の動作の差異 326  
 スキーマ検証の動作の差異 326

## ひ

---

日付 / 時刻型データを処理するアプリケー  
 ションの処理の流れ 20

## ふ

---

フィーチャーおよびプロパティの使用方法  
 71  
 SAX2 72  
 Shift\_JIS 切り替え機能 79

XML Schema 75

XSLT 74

高速パース機能 82

プログラム作成の流れ 86

プログラム実行時のトラブルシュート 89

プログラムの作成方法 85

## め

---

メッセージ 155

メッセージ一覧 157

メッセージの形式 156

## よ

---

用語解説 329

# ソフトウェアマニュアルのサービス ご案内

ソフトウェアマニュアルについて、3種類のサービスをご案内します。ご活用ください。

## 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おります。

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。

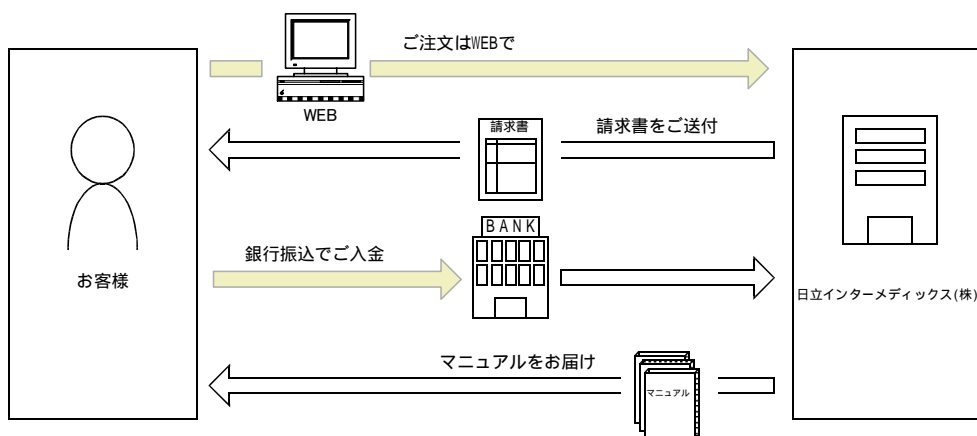
マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル情報	複数マニュアルを格納したCD-ROMマニュアルを提供しています。どの製品に対応したCD-ROMマニュアルがあるか、を参照できます。
マニュアルのご購入	日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からお申し込みできます。 (詳細は「3. マニュアルのご注文」を参照してください。)
Web提供マニュアル一覧	インターネットで参照できるマニュアルの一覧を提供しています。 (詳細は「2. インターネットからのマニュアル参照」を参照してください。)
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. インターネットからのマニュアル参照(ソフトウェアサポートサービス)

ソフトウェアサポートサービスの契約をしていただくと、インターネットでマニュアルを参照できます。本サービスの対象となる契約の種別、及び参照できるマニュアルは、マニュアル情報ホームページでご確認ください。なお、ソフトウェアサポートサービスは、マニュアル参照だけでなく、対象製品に対するご質問への回答、問題解決支援、バージョン更新版の提供など、お客様のシステムの安定的な稼働のためのサービスをご提供しています。まだご契約いただいていない場合は、ぜひご契約いただくことをお勧めします。

## 3. マニュアルのご注文

日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からご注文ください。



下記 URL にアクセスして必要事項を入力してください。

URL [http://www2.himdx.net/manual/privacy.asp?purchase\\_flag=1](http://www2.himdx.net/manual/privacy.asp?purchase_flag=1)

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。

請求書の金額を指定銀行へ振り込んでください。なお、送料は弊社で負担します。

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。