

# Cosminexus サービスプラットフォーム 開発ガイド

手引・操作書

3020-3-M72-60

## マニュアルの購入方法

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

## 対象製品

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows XP , Windows Vista

P-2443-7T74 uCosminexus Service Architect 07-60

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 ( x64 ) , Windows Server 2003 R2 ( x64 )

P-2443-7S74 uCosminexus Service Platform 07-60

適用 OS : AIX 6L V6.1 , AIX 5L V5.3 ( 5300-06 RMP ) , AIX 5L V5.2 ( 5200-10 RMP )

P-1M43-7S71 uCosminexus Service Platform 07-60

適用 OS : Red Hat Enterprise Linux AS 3 (x86) , Red Hat Enterprise Linux ES 3 (x86) , Red Hat Enterprise Linux AS 3 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux ES 3 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux AS 4 (x86) , Red Hat Enterprise Linux ES 4 (x86) , Red Hat Enterprise Linux AS 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux ES 4 (AMD64 & Intel EM64T) , Red Hat Enterprise Linux 5 Advanced Platform ( x86 ) , Red Hat Enterprise Linux 5 ( x86 ) , Red Hat Enterprise Linux 5 Advanced Platform ( AMD/Intel 64 ) , Red Hat Enterprise Linux 5 ( AMD/Intel 64 )

P-9S43-7S71 uCosminexus Service Platform 07-60

印の製品については、サポート時期をご確認ください。

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。

本製品では日立トレース共通ライブラリをインストールします。

## 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。

AMD は、Advanced Micro Devices, Inc. の商標です。

Borland のブランド名および製品名はすべて、米国 Borland Software Corporation 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。

Intel は、Intel Corporation の会社名です。

iWay および iWay Software は、Information Builders, Inc. の米国およびその他の国における登録商標です。

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。

JDK 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。

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。

Microsoft Office Excel は、米国 Microsoft Corporation の商品名称です。

MyEclipse は、米国 Genuitec 社の商品名称です。

OMG, CORBA, IOP, UML, Unified Modeling Language, MDA, Model Driven Architecture は、Object Management Group, Inc. 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。

ORACLE は、米国 Oracle Corporation の登録商標です。  
Oracle は、米国 Oracle Corporation 及びその子会社、関連会社の登録商標です。  
Oracle 10g は、米国 Oracle Corporation の商標です。  
Oracle9i は、米国 Oracle Corporation の商標です。  
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。  
SAP, mySAP, R/3, 記載されているすべての SAP 製品およびサービス名は SAP AG のドイツおよびその他世界各国における登録商標または商標です。  
SOAP ( Simple Object Access Protocol ) は、分散ネットワーク環境において XML ベースの情報を交換するための通信プロトコルの名称です。  
Sun, Sun Microsystems, Java は、米国 Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。  
UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。  
W3C は、World Wide Web Consortium の商標 ( 多数の国において登録された ) です。  
Windows は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。  
Windows Server は、米国 Microsoft Corporation の米国及びその他の国における登録商標です。  
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です。  
Eclipse は、開発ツールプロバイダのオープンコミュニティである Eclipse Foundation, Inc. により構築された開発ツール統合のためのオープンプラットフォームです。  
This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

## 発行

2006 年 5 月 ( 第 1 版 ) 3020-3-M72

2008 年 3 月 ( 第 4 版 ) 3020-3-M72-60

## 著作権

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

## 変更内容

変更内容 (3020-3-M72-60) uCosminexus Service Architect 07-60, uCosminexus Service Platform 07-60

追加・変更内容	変更箇所
AIX のバージョンを追加, 変更した。	-
次の適用 OS を追加した。 <ul style="list-style-type: none"> <li>Red Hat Enterprise Linux 5 Advanced Platform ( x86 )</li> <li>Red Hat Enterprise Linux 5 ( x86 )</li> <li>Red Hat Enterprise Linux 5 Advanced Platform ( AMD/Intel 64 )</li> <li>Red Hat Enterprise Linux 5 ( AMD/Intel 64 )</li> </ul>	-
次の適用 OS を削除した。 <ul style="list-style-type: none"> <li>HP-UX 11i V2 ( IPF )</li> <li>HP-UX 11i V3 ( IPF )</li> </ul>	-
データベースと Cosminexus RM を使用しない構成, またはデータベースを使用して Cosminexus RM を使用しない構成で, システムを構築できるようにした。また, テスト環境と本番環境で, データベースと Cosminexus RM の使用有無の設定が異なる場合でも, テスト環境で使用したリポジトリを本番環境へ移行できるようにした。	1.1.5, 2.2, 2.3(9), 2.4.2(1)(c), 2.4.2(2)(a), 2.4.3(1), 2.4.3(2), 2.4.4, 9.3, 9.3.2
開発から実運用までの流れに, テスト環境の作業を追加した。これに伴い, 開発環境および運用環境での作業を変更した。	1.2, 2.3
既存の業務処理システムのファイルの入出力できるようにしたことに伴い, ファイルアダプタ, および次のソフトウェアを追加した。 <ul style="list-style-type: none"> <li>uCosminexus Service Adapter Architect for Flat Files</li> <li>uCosminexus Service Adapter for Flat Files</li> </ul>	1.4.2(2)(h), 1.4.2(2)(i)
Eclipse への HCSCTE プラグインの組み込みの説明を変更した。	2.1.4
HCSC 簡易セットアップ機能を利用した場合に, 組み込みデータベース内で未使用のセグメントが不足したときの対策方法を変更した。	2.4.3(3)
リポジトリを初期化する前およびインポートするときの, HCSCTE プロジェクトに対する注意事項を追加した。また, エラーが発生した場合の対処方法を追加した。	3.2.1(2), 3.2.3
リポジトリをインポートするときの注意事項を追加した。	3.2.3(1)(b), 3.2.3(2)(b)
開発環境と運用環境で同じリポジトリを共有できないようにした。これに伴い, リポジトリを共有する場合の説明を削除した。	3.2, 9.3.3, 13.5, 付録 A.3
複合内容要素で構成区分が選択構成の場合, 構成要素に設定した開始セバレータから, 利用する構成要素を選択できるようにした。	4.4.2(3), 4.4.2(5), 表 4-5, 11.3.3(1)
ユーザ定義クラスに設定してはいけないファイル名を設定した場合の対処方法を追加した。	5.3.2(5), 付録 A.2(6), 付録 A.3(6)
検証結果で表示されるメッセージの種類を追加した。	5.6.2
ビジネスプロセスのステータスを非永続にするときの注意事項を追加した。	6.2.1(3)
一括してプロセスのフォルト処理を実行する場合の説明を追加した。	6.4.3(2)
ビジネスプロセスのフォルト処理の定義で使用する電文フォーマットについての説明を追加した。	6.4.3(3)

追加・変更内容	変更箇所
<p>次の制限を解除した。</p> <ul style="list-style-type: none"> <li>永続化しないビジネスプロセスでは、分岐アクティビティ、並列処理開始アクティビティ、および繰り返しアクティビティを使用できない。</li> <li>永続化しないビジネスプロセスでは、受付アクティビティと対応する応答アクティビティの組は一つだけとする。</li> </ul>	6.6.2(2), 6.6.3(2), 6.6.11(2), 6.6.12(3), 6.6.13(3), 表 6-19
<p>ビジネスプロセスの検証時に HCSCTE プロジェクトの lib ディレクトリに関する検証をすることを追加した。また、これに伴い HCSCTE プロジェクト内の src ディレクトリおよび lib ディレクトリへのリンクの削除についての注意事項を追加した。</p>	6.6.5(1), 6.9, 表 6-17
<p>データ変換アクティビティのデータ変換定義に関する検証をビジネスプロセスの検証時にすることを追加した。</p>	6.9, 表 6-17
<p>定数ファンクションに指定した値を、複数のマッピング先に代入できるようにした。</p>	7.5.15(1), 表 7-8
<p>くり返しファンクションやマッピング線を削除して、祖先ノードにくり返しを設定されているノードのくり返しを取り消した場合の説明を追加した。</p>	7.6.3, 7.6.3(3)
<p>開発環境での定義と運用環境で配備される HCSC コンポーネントの関係の説明を追加した。</p>	9.1
<p>下位バージョンからの移行手順を、旧バージョンで使用していたリポジトリ情報を開発環境で変更し、変更したリポジトリ情報を使用して、バージョンアップ後に運用する場合の手順に変更した。</p>	9.1, 付録 A
<p>応答が byte[] 型の場合で、応答電文がないときは、0 バイトのバイト配列で応答することを説明した。</p>	10.2.7(2), 10.2.8(1)(a)
<p>単純内容要素として使用するデータのデータ種別が文字列の場合で、サイズが固定のとき、埋め字をスペースと空文字から選択できるようにした。</p>	11.3.2(1)
<p>拡張関数に、bpws:getVariableProperty を追加した。</p>	11.8.12
<p>次のメッセージを変更した。  KECT05003-E, KECT10001-E, KECT10002-E, KECT10004-E, KECT10005-E, KECT10009-E, KECT11008-E, KECT12002-E, KECT12003-E, KECT12006-E, KECT12009-E ~ KECT12011-E, KECT12014-E, KECT12019-E, KECT12021-E, KECT12023-E, KECT12101-I, KECT12500-W ~ KECT12502-W, KECT24002-E, KECT24012-E, KECT24013-E, KECT24018-E, KECT24027-E, KECT24029-E, KECT24030-E, KECT24503-W ~ KECT24505-W, KECT24507-E</p>	13.2, 13.3
<p>次のメッセージを追加した。  KECT24034-E, KECT24035-E, KECT24508-W, KECK72001-E, KECK72499-E, KECK73001-E, KECK73003-E, KECK73005-E, KECK73007-E, KECK73010-E, KECK73499-E ~ KECK73501-I, KECK74001-W, KECK74002-W</p>	13.3, 13.4
<p>BPEL ファイルの要素とビジネスプロセス定義の説明に、sequence 要素を追加した。</p>	表 B-1, 付録 B.3(5)
<p>サンプルプログラムの使用方法を追加した。</p>	付録 C

単なる誤字・脱字などはお断りなく訂正しました。

---

追加・変更内容

---

Cosminexus サービスプラットフォームの全体像、および環境間の関係についての説明を追加した。

サービスリクエストからの実行要求を受け付けるインターフェースとして、任意の形式を定義できる「ユーザ定義受付 (同期受付 (Web サービス))」を利用できるようにした。

Windows 2000 を適用 OS から除外し、新たに Windows Vista を適用 OS として追加した。

データ変換定義で、変換テーブルを利用して値を変換する値置換えファンクションを利用できるようにした。

既存の TPBroker システム (Object Wrapper システム) と接続する場合に使用する、次のソフトウェアを追加した。

- uCosminexus Service Adapter Architect for Object Access
- uCosminexus Service Adapter for Object Access

他社 ERP パッケージと連携する場合に使用する、iWay アダプタを追加した。

テスト環境の構築を支援する簡易構築機能を追加した。

使用する Eclipse のバージョンを変更した。

Cosminexus サービスプラットフォームをバージョンアップするときに、下位バージョンで使用していたリポジトリをバージョンアップ後も使用できるようにするための移行機能を追加した。

サービスで利用する XML スキーマや WSDL で、外部の XML スキーマを参照できるようにした。

エクスポートされた ZIP ファイルから必要な HCSC コンポーネントの情報だけをインポートできるようにした。

DB アダプタの利用時に、サービスリクエストからの 1 回の実行要求で複数の SQL 文を実行できるようにした。

バイナリフォーマット定義ファイルの作成で、扱うバイナリデータが CSV 形式の場合の定義方法を変更した。

複合内容要素の構成要素である単純内容要素が、要素ごとに区切り文字 (セパレータ) で区切られたデータ (セパレータデータ) の場合、複合内容要素にセパレータを設定できるようにした。

サービスアダプタ定義画面の標準電文の表示 / 非表示の切り替え機能を削除した。

サービスアダプタ (MDB (WS-R)) を利用して、Cosminexus Reliable Messaging のローカルキューを実行できるようにした。

BPMN を用いたモデリングツールで作成した BPEL ファイルをインポートして、ビジネスプロセスの定義を新規追加できるようにした。

割り当て済みの変数・相関セットの情報を更新できるようにした。

呼び出すサービスまたはユーザ定義受付の要求電文、応答電文、またはフォルト電文に設定されている電文フォーマットを変数として設定する場合、電文フォーマットの取込ダイアログを利用して設定できるようにした。

次のメッセージを追加した。

KECT12503-E, KECT24033-E, KECT24507-E

次のメッセージを変更した。

KECT10003-E

---

---

追加・変更内容

---

ルーティング機能を削除した。

使用できる HiRDB または HiRDB サーバのバージョンを変更した。

サービス側の電文フォーマットがバイナリ形式の場合にデータ形式を変換するための次の機能について追加した。

- バイナリ形式の電文フォーマット定義ファイルを作成する機能を追加した。
- バイナリ形式のデータと XML 形式のデータを変換する場合に使用する関連ソフトウェアとして、次のソフトウェアを追加した。
  - ・ uCosminexus 日立コード変換 - Development Kit
  - ・ uCosminexus 日立コード変換 - Server Runtime
  - ・ uCosminexus 日立コード変換 - Runtime

既存の OpenTP1 システムと接続する場合に使用する、次のソフトウェアを追加した。

- uCosminexus Service Adapter Architect for TP1
- uCosminexus Service Adapter for TP1
- uCosminexus TP1 Gateway

また、TP1/Server Base Enterprise Option を uCosminexus TP1/Server Base Enterprise Option に変更した。

DB アダプタから操作できるデータベースとして、Oracle および HiRDB Version 8 を追加した。

document-literal 型の WSDL 取得方法を追加した。

SOAPFault のエラー情報を取得するための方法を追加した。

Web サービスのサービス部品の電文フォーマットで複数のルート要素を利用できるようにした。

変数の部分定義の一意性の範囲を変更し、サービスの非同期呼び出しで、異なる変数から同じ相関セットを利用できるようにした。

ビジネスプロセスを定義するためのフォルト送出アクティビティ、および無操作アクティビティを追加した。

ビジネスプロセスの再実行を要求するサービスリクエストの作成方法を追加した。

ロードバランス機能を利用した HCSC サーバの構成を追加した。

次のメッセージを追加した。

KECT10008-E, KECT10009-E, KECT13100-E ~ KECT13103-W, KECT17100-E, KECT18100-E ~ KECT18103-E, KECT21100-E, KECT24000-E ~ KECT24022-E, KECT24027-E ~ KECT24032-E, KECT24100-I, KECT24500-W, KECT24501-W ~ KECT24506-W

次のメッセージを変更した。

KECT10003-E

第 2 版では第 1 版の内容のうち、4 章および 6 章から電文フォーマット定義に関連する内容を抽出し、新たに 4 章にまとめました。第 1 版との対応は次のようになっています。

第 1 版	第 2 版
1. SOA に基づいたシステム開発の概要	1. SOA に基づいたシステム開発の概要
2. システムを開発する前に	2. システムを開発する前に
3. プロジェクトの管理とリポジトリの管理	3. プロジェクトの管理とリポジトリの管理
-	4. 電文フォーマットの作成 (第 1 版の 4 章および 6 章から抽出して新規作成)
4. アダプタの定義	5. アダプタの定義
5. ビジネスプロセスの定義	6. ビジネスプロセスの定義
6. データ変換	7. データ変換の定義
7. サービスコンポーネントのパッケージングと配備定義	8. サービスコンポーネントのパッケージングと配備定義
8. サービスリクエストの作成	9. サービスリクエストの作成
9. 開発環境で使用する画面	10. 開発環境で使用する画面
10. コマンド	11. コマンド
11. メッセージ	12. メッセージ
付録 A 用語解説	付録 A 用語解説

# はじめに

---

このマニュアルは、Cosminexus が提供する機能を利用して、SOA に対応したサービス統合環境（サービスプラットフォーム）を開発する方法について説明しています。Cosminexus サービスプラットフォームでは、次に示すプログラムプロダクトを使用してシステムを開発します。

- P-2443-7S74 uCosminexus Service Platform 07-60
- P-2443-7T74 uCosminexus Service Architect 07-60
- P-1M43-7S71 uCosminexus Service Platform 07-60
- P-9S43-7S71 uCosminexus Service Platform 07-60

また、オペレーティングシステム（OS）の種類によって、機能が異なる場合があります。OS ごとの違いがある場合の表記方法については、「適用 OS の違いによる機能相違点の表記」を参照してください。

## 対象読者

Cosminexus サービスプラットフォームを利用して、SOA に対応したシステムを開発する方を対象としています。

対象読者の前提知識として必要な知識を次に示します。

- SOA に関する基本的な知識
- J2EE に関する基本的な知識
- SQL およびリレーショナルデータベースに関する基本的な知識
- XML に関する基本的な知識
- Java によるプログラム開発に関する基本的な知識

なお、このマニュアルの前提である、マニュアル「Cosminexus サービスプラットフォーム 概説」を必ずお読みください。

## 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 SOA に基づいたシステム開発の概要

SOA に基づいたシステム開発の概要について説明しています。

### 第 2 章 システムを開発する前に

Cosminexus サービスプラットフォームのシステムを開発する前に実施する作業、開発手順、および利用できるサービス部品の種類と適用範囲について説明しています。

### 第 3 章 プロジェクトの管理とリポジトリの管理

プロジェクトの作成、プロパティの設定など、プロジェクトの管理について説明しています。また、リポジトリの設定など、リポジトリの管理について説明しています。

はじめに

#### 第 4 章 電文フォーマットの作成

データ変換定義に利用する電文フォーマットの作成方法について説明しています。

#### 第 5 章 アダプタの定義

サービスアダプタおよび DB アダプタの定義方法について説明しています。

#### 第 6 章 ビジネスプロセスの定義

ビジネスプロセスの定義方法について説明しています。

#### 第 7 章 データ変換の定義

データ変換 (マッピング) 定義の方法について説明しています。

#### 第 8 章 ユーザ定義受付の定義

ユーザ定義受付の定義方法について説明しています。

#### 第 9 章 HCSC コンポーネントのパッケージングと配備定義

定義した HCSC コンポーネントのパッケージング方法, およびパッケージングした HCSC コンポーネントを実行環境に配備するための定義方法について説明しています。

#### 第 10 章 サービスリクエストの作成

HCSC サーバの標準受付およびユーザ定義受付にサービス部品を要求するサービスリクエストの作成について説明しています。

#### 第 11 章 開発環境で使用する画面

Cosminexus サービスプラットフォームの開発環境で使用する画面の詳細および画面での操作方法について説明しています。

#### 第 12 章 コマンド

Cosminexus サービスプラットフォームの開発環境で使用するコマンドの文法について説明しています。

#### 第 13 章 メッセージ

Cosminexus サービスプラットフォームの開発環境で出力されるメッセージとその対処について説明しています。

#### 付録 A 下位バージョンからのリポジトリの移行

下位のバージョンで使用していたリポジトリ情報を開発環境で変更し, 変更したリポジトリ情報を使用して, Cosminexus サービスプラットフォームをバージョンアップする方法について説明しています。

#### 付録 B 上流ツールとの連携で使用する BPEL のサポート範囲

BPMN を用いたモデリングツールで作成した BPEL ファイルを, Cosminexus サービスプラットフォームで利用できるビジネスプロセスに変換するときに, 情報がどのように変換されるのかについて説明しています。

#### 付録 C Cosminexus サービスプラットフォームのサンプルプログラム

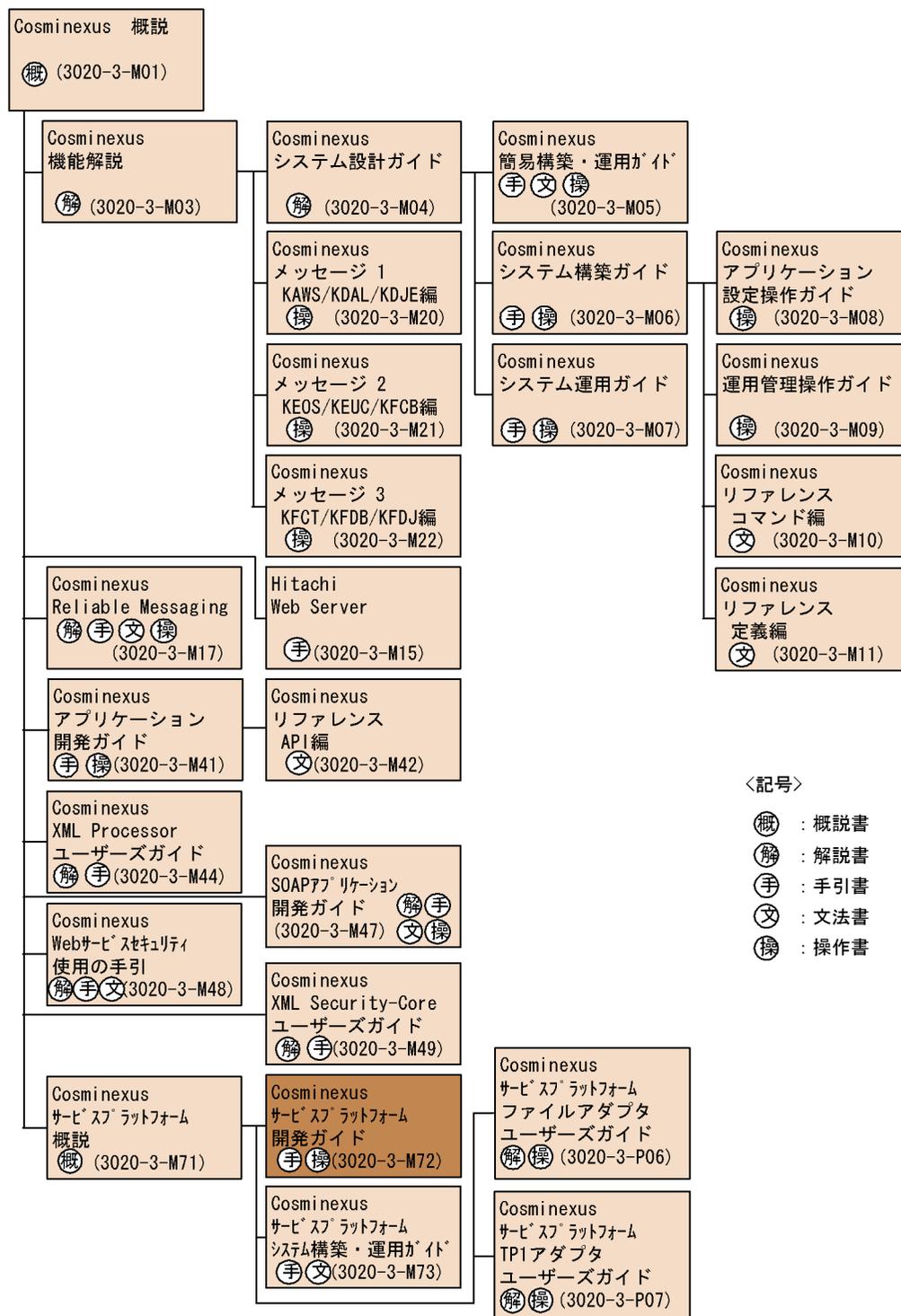
Cosminexus サービスプラットフォームを理解するために, サンプルプログラムを実際に動かしてみる方法について説明しています。

## 付録 D 用語解説

このマニュアルで使用している用語について説明しています。

### 関連マニュアル

Cosminexus のマニュアル体系について、次の図に示します。



それぞれのマニュアルで説明している内容の概要を次に示します。

## Cosminexus 概説

Cosminexus の製品概要について説明しています。

## Cosminexus 機能解説

Cosminexus の実行環境で提供する機能について説明しています。

## Cosminexus システム設計ガイド

システム設計時に、システムの目的に応じたシステム構成や運用方法を検討するための指針について説明しています。また、チューニングの方法についても説明しています。

## Cosminexus 簡易構築・運用ガイド

Smart Composer 機能を使用して、システムを構築・運用する手順、および構築・運用時に使用するコマンドやファイルについて説明しています。

## Cosminexus システム構築ガイド

運用管理ポータル、またはファイル編集とコマンド実行によって、システムを構築する手順について説明しています。

## Cosminexus システム運用ガイド

Cosminexus を使用したシステムの運用方法について説明しています。

## Cosminexus アプリケーション設定操作ガイド

Cosminexus Component Container のサーバ管理コマンド、および Server Plug-in を使用した操作について説明しています。

## Cosminexus 運用管理操作ガイド

Cosminexus Component Container の運用管理ポータルの画面および画面操作について説明しています。

## Cosminexus リファレンス コマンド編

Cosminexus のシステムを構築・運用するときに使用するコマンドについて説明しています。

## Cosminexus リファレンス 定義編

Cosminexus のシステムを構築・運用するとき、またはアプリケーションを開発するときに、使用するファイルの形式について説明しています。

## Cosminexus メッセージ 1 KAWS / KDAL / KDJE 編, Cosminexus メッセージ 2 KEOS / KEUC / KFCB 編, Cosminexus メッセージ 3 KFCT / KFDB / KFDJ 編

Cosminexus で出力されるメッセージについて説明しています。

## Hitachi Web Server

Hitachi Web Server (Web サーバ) の構築、管理方法について説明しています。

## Cosminexus Reliable Messaging

Cosminexus RM を使用したメッセージの非同期通信によるアプリケーションの連携方法について説明しています。

はじめに

#### Cosminexus アプリケーション開発ガイド

構築した Cosminexus のシステムで動作させる，アプリケーションの開発方法について説明しています。

#### Cosminexus リファレンス API 編

アプリケーションの開発で使用する API およびタグについて説明しています。

#### Cosminexus XML Processor ユーザーズガイド

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ています。

#### Cosminexus SOAP アプリケーション開発ガイド

Cosminexus が提供する機能を利用して，SOAP アプリケーションを開発，実行する方法について説明しています。

#### Cosminexus Web サービスセキュリティ 使用の手引

Web サービスセキュリティ機能について説明しています。

#### Cosminexus XML Security - Core ユーザーズガイド

XML 署名データの生成・検証機能，および XML 暗号化・復号化機能について説明しています。

#### Cosminexus サービスプラットフォーム 概説

Cosminexus サービスプラットフォームで SOA を適用したシステムを実現するための機能，および開発，運用の概要について説明しています。

#### Cosminexus サービスプラットフォーム システム構築・運用ガイド

Cosminexus サービスプラットフォームの機能，運用環境・実行環境の構築方法，および運用方法について説明しています。

#### Cosminexus サービスプラットフォーム ファイルアダプタ ユーザーズガイド

Cosminexus サービスプラットフォーム ファイルアダプタの機能，ファイルアダプタの作成および運用について説明しています。

#### Cosminexus サービスプラットフォーム TP1 アダプタ ユーザーズガイド

Cosminexus サービスプラットフォーム TP1 アダプタの機能，アダプタの作成および運用について説明しています。

また，マニュアル体系図に示したマニュアル以外で，このマニュアルと関連する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。

- TP1/Server Base Enterprise Option 使用の手引 (3000-3-982)
- TP1/Server Base Enterprise Option プログラム作成の手引 (3000-3-983)
- スケーラブルデータベースサーバ HiRDB Version 8 システム導入・設計ガイド (Windows(R) 用) (3020-6-352)
- スケーラブルデータベースサーバ HiRDB Version 8 システム定義 (Windows(R) 用) (3020-6-353)
- スケーラブルデータベースサーバ HiRDB Version 8 システム運用ガイド (Windows(R) 用)

( 3020-6-354 )

- スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス ( Windows(R) 用 )( 3020-6-355 )
- スケーラブルデータベースサーバ HiRDB Version 8 UAP 開発ガイド ( 3020-6-356 )
- スケーラブルデータベースサーバ HiRDB Version 8 SQL リファレンス ( 3020-6-357 )
- スケーラブルデータベースサーバ HiRDB Version 8 メッセージ ( 3020-6-358 )

## 読書手順

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

マニュアルを読む目的	記述箇所
概要、システム構成、および uCosminexus Service Architect を利用する場合の動作環境について知りたい。	1 章
uCosminexus Service Architect のインストールなど Cosminexus サービスプラットフォームのシステムを開発する前に実施する作業、開発手順、および利用できるサービス部品の種類と適用範囲について知りたい。	2 章
プロジェクトの管理 ( プロジェクトの作成、プロパティ設定など ) およびリポジトリの管理 ( リポジトリの設定など ) について知りたい。	3 章
電文フォーマットについて知りたい。	4 章
サービスアダプタおよび DB アダプタの定義について知りたい。	5 章
ビジネスプロセスの定義について知りたい。	6 章
データ変換について知りたい。	7 章
ユーザ定義受付の定義について知りたい。	8 章
HCSC コンポーネントのパッケージング方法と配備定義について知りたい。	9 章
サービスリクエストの作成方法について知りたい。	10 章
Cosminexus サービスプラットフォームの開発環境で使用する画面のボタンや項目、操作方法などについて知りたい。	11 章
Cosminexus サービスプラットフォームの開発環境で使用するコマンドの文法について知りたい。	12 章
Cosminexus サービスプラットフォームの開発環境で出力されるメッセージとその対処について知りたい。	13 章
下位のバージョンで使用していたリポジトリ情報を移行して、Cosminexus サービスプラットフォームをバージョンアップする方法について知りたい。	付録 A
BPMN を用いたモデリングツールで作成した BPEL ファイルを、Cosminexus サービスプラットフォームで利用できるビジネスプロセスに変換するとき、情報がどのように変換されるのかについて知りたい。	付録 B
サンプルプログラムの使用方法について知りたい。	付録 C
このマニュアルで使用している用語の意味について知りたい。	付録 D

## このマニュアルでの表記

このマニュアルで使用している表記と、対応する製品名を次に示します。

表記		製品名
Eclipse		Eclipse Platform Runtime Binary 3.2.2 以降 ( 3.3 は対象外 )
Excel		Microsoft(R) Office Excel
HiRDB または HiRDB サーバ	HiRDB/Single Server	HiRDB/Single Server Version 7
		HiRDB/Single Server Version 8
	HiRDB/Parallel Server	HiRDB/Parallel Server Version 7
		HiRDB/Parallel Server Version 8
HiRDB Embedded Server または組み込みデータベース	HiRDB Embedded Server Version 8	
Oracle	Oracle9i	Oracle9i
		Oracle9i R2
	Oracle10g	Oracle 10g
		Oracle 10g R2
TP1/EE		TP1/Server Base Enterprise Option
UNIX	AIX	AIX 6L V6.1
		AIX 5L V5.3 ( 5300-06 RMP )
		AIX 5L V5.2 ( 5200-10 RMP )
	Linux または Linux ( x86 / AMD64 & Intel EM64T )	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 )

表記	製品名
	Red Hat Enterprise Linux ES 4 ( AMD64 & Intel EM64T )
	Red Hat Enterprise Linux 5 Advanced Platform ( x86 )
	Red Hat Enterprise Linux 5 ( x86 )
	Red Hat Enterprise Linux 5 Advanced Platform ( AMD/Intel 64 )
	Red Hat Enterprise Linux 5 ( AMD/ Intel 64 )
Windows Server 2003	Microsoft(R) Windows Server(R) 2003, Enterprise Edition 日本語版
	Microsoft(R) Windows Server(R) 2003, Standard Edition 日本語版
Windows Server 2003 R2	Microsoft(R) Windows Server(R) 2003, Enterprise R2 Edition 日本語 版
	Microsoft(R) Windows Server(R) 2003, Standard R2 Edition 日本語版
Windows Server 2003 ( x64 )	Microsoft(R) Windows Server(R) 2003 , Enterprise x64 Edition Operating System
	Microsoft(R) Windows Server(R) 2003 , Standard x64 Edition Operating System
Windows Server 2003 R2 ( x64 )	Microsoft(R) Windows Server(R) 2003 R2 , Enterprise x64 Edition Operating System
	Microsoft(R) Windows Server(R) 2003 R2 , Standard x64 Edition Operating System
Windows Vista	Microsoft(R) Windows Vista(R) Business 日本語版
	Microsoft(R) Windows Vista(R) Enterprise 日本語版
	Microsoft(R) Windows Vista(R) Ultimate 日本語版
Windows XP	Microsoft(R) Windows(R) XP Professional Operating System
Word	Microsoft(R) Word

Windows Vista , Windows XP , Windows Server 2003 R2 ( x64 ) , Windows Server 2003 ( x64 ) , Windows Server 2003 R2 , Windows Server 2003 を総称して Windows と表記すること

はじめに

があります。

このマニュアルで使用している表記と、対応する Cosminexus の機能名を次に示します。

表記	Cosminexus の機能名
Cosminexus RM	Cosminexus Reliable Messaging
CTM	Cosminexus Component Transaction Monitor
HCSC	Cosminexus Service Coordinator
HCSC-Business Process, ビジネスプロセス基盤, または BP 基盤	Cosminexus Service Coordinator - Business Process
HCSC-Data Transform またはデータ変換基盤	Cosminexus Service Coordinator - Data Transform
HCSC-DB Adapter または DB アダプタ連携基盤	Cosminexus Service Coordinator - Data Base Adapter
HCSC-Manager または HCSC-MNG	Cosminexus Service Coordinator - Manager
HCSC-Messaging, HCSC-MSG, またはメッセージング基盤	Cosminexus Service Coordinator - Messaging
HCSC-TE	Cosminexus Service Coordinator Tools for Eclipse
MyEclipse	MyEclipse for Cosminexus
TPBroker	Cosminexus TPBroker

このマニュアルで使用している表記と、対応する Java 関連用語を次に示します。

表記	Java 関連用語
EAR	Enterprise ARchive
EJB または Enterprise JavaBeans	Enterprise JavaBeans™
J2EE	Java™ 2 Platform, Enterprise Edition
J2SE	Java™ 2 Platform, Standard Edition
JAR	Java™ Archive
Java	Java™
JMS	Java™ Message Service

### 適用 OS の違いによる機能相違点の表記

このマニュアルは、適用 OS が Windows, AIX, および Linux の製品に対応するため、OS の種類によって、機能が異なる場合があります。OS ごとの違いがある場合は、次に示す表記を使用して、それぞれの説明に OS 名を明記しています。

表記	意味
Windows の場合	Windows に該当する表記です。
AIX の場合	AIX に該当する表記です。

表記	意味
Linux の場合	Linux に該当する表記です。
UNIX の場合	UNIX ( AIX および Linux ) に該当する表記です。

### パスの表記について

このマニュアルでは、原則として Windows のパスの区切り (「¥」) を使用しています。UNIX の場合は、「¥」を「/」に置き換えてお読みください。

### このマニュアルで使用している略語

このマニュアルで使用している英略語を次に示します。

英略語	英字での表記
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
ASCII	<u>A</u> merican <u>S</u> tandard <u>C</u> ode for <u>I</u> nformation <u>I</u> nterchange
BMT	<u>B</u> ean <u>M</u> anaged <u>T</u> ransaction
BP	<u>B</u> usiness <u>P</u> rocess
BP <sub>E</sub> L	<u>B</u> usiness <u>P</u> rocess <u>E</u> xecution <u>L</u> anguage
BPMN	<u>B</u> usiness <u>P</u> rocess <u>M</u> odeling <u>N</u> otation
CMT	<u>C</u> ontainer <u>M</u> anaged <u>T</u> ransaction
DB	<u>D</u> atabase
HTML	<u>H</u> yper <u>T</u> ext <u>M</u> arkup <u>L</u> anguage
HTTP	<u>H</u> yper <u>T</u> ext <u>T</u> ransfer <u>P</u> rotocol
HTTPS	<u>H</u> yper <u>T</u> ext <u>T</u> ransfer <u>P</u> rotocol <u>S</u> ecurity
IDE	<u>I</u> ntegrated <u>D</u> evelopment <u>E</u> nvironment
IIOP	<u>I</u> nternet <u>I</u> nter- <u>O</u> rb <u>P</u> rotocol
MDB	<u>M</u> essage- <u>D</u> riven <u>B</u> ean
OS	<u>O</u> perating <u>S</u> ystem
RDB	<u>R</u> elational <u>D</u> atabase
RMI	<u>R</u> emote <u>M</u> ethod <u>I</u> nvoication
RPC	<u>R</u> emote <u>P</u> rocedure <u>C</u> all
SOA	<u>S</u> ervice <u>O</u> riented <u>A</u> rchitecture
SOAP	<u>S</u> imple <u>O</u> bject <u>A</u> ccess <u>P</u> rotocol
SQL	<u>S</u> tructured <u>Q</u> uery <u>L</u> anguage
UNC	<u>U</u> niversal <u>N</u> aming <u>C</u> onvention
URL	<u>U</u> niform <u>R</u> esource <u>L</u> ocator

英略語	英字での表記
UTF	<u>U</u> CS <u>T</u> ransformation <u>F</u> ormat
VM	<u>V</u> irtual <u>M</u> achine
W3C	<u>W</u> orld <u>W</u> ide <u>W</u> eb <u>C</u> onsortium
WSDL	<u>W</u> eb <u>S</u> ervice <u>D</u> escription <u>L</u> anguage
WS-I	<u>W</u> eb <u>S</u> ervices <u>I</u> nteroperability
WS-R	<u>W</u> eb <u>S</u> ervices <u>R</u> eliability
WST	<u>W</u> eb <u>S</u> tandard <u>T</u> ools
XML	<u>E</u> xtensible <u>M</u> arkup <u>L</u> anguage
XPath	<u>X</u> ML <u>P</u> ath Language

## このマニュアルで使用している記号

### ! 注意事項

間違いやすい点、および間違うとトラブルが発生するような項目について説明しています。

### ポイント

その説明の要点について説明しています。

### 参考

補足的な情報を説明しています。

## このマニュアルの図中で使用している記号

このマニュアルの図中で使用している記号を、次のように定義します。

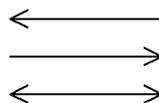
## ●マシン



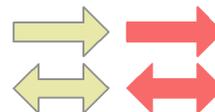
## ●入出力の動作



## ●制御の流れ



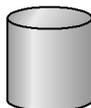
## ●データ、情報の流れ



## ●プログラム



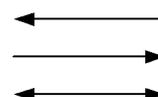
## ●リポジトリ、データベース



## ●工程、作業項目の流れ



## ●その他の流れ



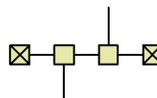
## ●サーバ



## ●ネットワーク



## ●LAN



## ●CD-ROM



## このマニュアルの画面や操作の説明で使用している記号

このマニュアルでは、次に示す記号を使用して画面や操作を説明しています。

記号	意味
[ ]	ダイアログ名、キー名、メニュー項目名、ボタン名、および画面に表示されている項目名に使用します。
[ ] - [ ]	- の前に示したメニューから、- の後ろのメニューを選択することを表します。
< >	< > 内の名称がユーザの環境によって異なることを表します。
「 」	入力値、可変値、またはメッセージなどのユーザが入力する内容を表します。
+	+ の前のキーを押したまま、後ろのキーを押すことを表します。 (例) [Ctrl] キー + [V] キー

## 文法の説明で使用する記号

このマニュアルの文法の説明で使用する記号について次に示します。

記号	意味
	横に並べられた複数の項目に対する項目間の区切りを示し、「または」を意味します。 (例) A   B A または B を指定することを示します。

記号	意味
{ }	この記号で囲まれている複数の項目のうちから一つを選択することを示します。項目が横に並べられ、記号   で区切られている場合は、そのうちの一つを選択します。 (例) { A   B   C } A, B または C のどれかを指定することを示します。
[ ]	この記号で囲まれている項目は省略してもよいことを示します。 (例) [ A ] 「何も指定しない」が「A を指定する」ことを示します。
...	記述が省略されていることを示します。 (例) ABC... ABC の後ろに記述があり、その記述が省略されていることを示します。
(( ))	項目で指定できる値の範囲を示します。
< >	この記号で囲まれている項目は、該当する要素を指定することを示します。 (例) < プロパティ > プロパティを記述します。

## このマニュアルで使用している構文要素

このマニュアルで使用する構文要素の種類と意味を次に示します。

種類	定義
英字	A ~ Z a ~ z
英小文字	a ~ z
英大文字	A ~ Z
数字	0 ~ 9
英数字	A ~ Z a ~ z 0 ~ 9
記号	! " # \$ % & ' ( ) + , _ . / : ; < = > @ [ ] ^ - { } タブ 空白
文字列	任意の文字の配列

注 すべての半角文字を使用してください。

## 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。

宛先(あてさき) 個所(かしよ) 桁(けた) 進捗(しんちよく) 揃える(そろえる) 貼り付け(はりつけ) 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<sup>2</sup> バイト, 1,024<sup>3</sup> バイト, 1,024<sup>4</sup> バイトです。

# 目次

<b>1</b>	<b>SOA に基づいたシステム開発の概要</b>	<b>1</b>
1.1	概要	2
1.1.1	Cosminexus サービスプラットフォームの全体像	2
1.1.2	Cosminexus サービスプラットフォームを構成する環境	3
1.1.3	開発環境・実行環境・運用環境の関係	4
1.1.4	開発環境とテスト環境との関係	6
1.1.5	テスト環境と本番環境との関係	7
1.2	開発から実運用までの流れ	10
1.3	システム全体と開発環境の関係	16
1.4	動作環境	19
1.4.1	前提 OS	19
1.4.2	ソフトウェアの構成	19
<b>2</b>	<b>システムを開発する前に</b>	<b>25</b>
2.1	開発環境を使用するためのセットアップ	26
2.1.1	前提条件	27
2.1.2	インストール	27
2.1.3	Eclipse の設定ファイルの確認	28
2.1.4	Eclipse への HCSCTE プラグインの組み込み	28
2.1.5	環境設定	30
2.1.6	uCosminexus 日立コード変換 - Development Kit のセットアップ	30
2.1.7	組み込みデータベースの構築	32
2.1.8	アンインストール	35
2.2	構成の形態と選択	36
2.3	開発手順	38
2.4	テスト環境の簡易セットアップ	43
2.4.1	HCSC 簡易セットアップ機能で構築できる環境	43
2.4.2	HCSC 簡易セットアップ機能の実行	45
2.4.3	HCSC 簡易セットアップ機能でセットアップしたテスト環境の運用	51
2.4.4	テスト環境のカスタマイズ	57
2.5	利用できるサービス部品の種類と適用範囲	61
2.5.1	Web サービスを利用したサービス部品の適用範囲	61
2.5.2	SessionBean を利用したサービス部品の適用範囲	63

2.5.3	Cosminexus RM のローカルキューを利用したサービス部品の適用範囲	67
2.5.4	DB キューを利用したサービス部品の適用範囲	67
2.5.5	XML スキーマの適用範囲	68

## 3

<b>プロジェクトの管理とリポジトリの管理</b>		<b>71</b>
3.1	プロジェクトの管理	72
3.1.1	プロジェクトを作成する	72
3.1.2	プロパティを設定する	75
3.1.3	プロジェクトをエクスポート/インポートする	76
3.1.4	プロジェクトを削除する	77
3.2	リポジトリの管理	78
3.2.1	リポジトリを設定する	78
3.2.2	リポジトリをエクスポートする	80
3.2.3	リポジトリをインポートする	81

## 4

<b>電文フォーマットの作成</b>		<b>91</b>
4.1	電文フォーマットとデータ変換	92
4.2	電文フォーマットの種類	95
4.3	電文フォーマット (XML フォーマット定義ファイル) の作成方法	96
4.3.1	標準電文を作成する	96
4.3.2	サービス部品電文を作成する (Web サービスの場合)	96
4.3.3	サービス部品電文を作成する (SessionBean の場合)	98
4.3.4	サービス部品電文を作成する (MDB (WS-R または DB キュー) の場合)	98
4.3.5	サービス部品電文を作成する (DB アダプタ電文の場合)	98
4.4	電文フォーマット (バイナリフォーマット定義ファイル) の作成方法	111
4.4.1	バイナリフォーマット定義ファイルを新規作成する	112
4.4.2	要素を定義する (CSV 以外の形式の場合)	113
4.4.3	要素を定義する (CSV 形式の場合)	119
4.4.4	バイナリフォーマット定義ファイルを編集する	122
4.4.5	バイナリフォーマット定義ファイルを検証する	122

## 5

<b>アダプタの定義</b>		<b>133</b>
5.1	定義の流れ	134
5.2	アダプタの追加	136
5.2.1	サービスアダプタを新規に追加する	136

5.2.2	DB アダプタを新規に追加する	137
5.2.3	定義済みアダプタを利用してアダプタを追加する	138
5.3	アダプタの内容の定義	139
5.3.1	サービスアダプタ (Web サービス) を定義する	139
5.3.2	サービスアダプタ (SessionBean) を定義する	142
5.3.3	サービスアダプタ (MDB (WS-R)) を定義する	147
5.3.4	サービスアダプタ (MDB (DB キュー)) を定義する	150
5.3.5	DB アダプタを定義する	151
5.3.6	アダプタ定義での設定一覧	155
5.4	アダプタの保存	161
5.5	アダプタの編集	162
5.6	アダプタの検証	163
5.6.1	検証方法	163
5.6.2	検証内容の表示	163
5.7	アダプタの削除	164

## 6

	ビジネスプロセスの定義	165
6.1	定義の流れ	166
6.2	ビジネスプロセスの追加	168
6.2.1	ビジネスプロセスを新規に追加する	168
6.2.2	定義済みビジネスプロセスを利用してビジネスプロセスを追加する	172
6.3	ビジネスプロセスの内容の定義	173
6.4	アクティビティの配置と連結	174
6.4.1	アクティビティを配置する	174
6.4.2	アクティビティを連結する	174
6.4.3	フォルト処理を定義する	178
6.5	変数・相関セットの定義	185
6.5.1	変数の定義	185
6.5.2	相関セットの定義	191
6.6	アクティビティの定義	200
6.6.1	開始アクティビティ	201
6.6.2	受付アクティビティ	201
6.6.3	応答アクティビティ	203
6.6.4	サービス呼出アクティビティ	204
6.6.5	Java 呼出アクティビティ	206
6.6.6	データ変換アクティビティ	210

6.6.7	代入アクティビティ	212
6.6.8	無操作アクティビティ	214
6.6.9	フォルト送出アクティビティ	215
6.6.10	スコープアクティビティ	216
6.6.11	繰り返しアクティビティ	217
6.6.12	分岐アクティビティ	219
6.6.13	並列処理アクティビティ	222
6.6.14	終了アクティビティ	227
6.6.15	順次処理アクティビティ	227
6.7	ビジネスプロセスの保存	228
6.8	ビジネスプロセスの編集	229
6.8.1	ビジネスプロセスおよびアクティビティの定義情報を変更する	229
6.8.2	アクティビティ名を変更する	230
6.8.3	運用中のビジネスプロセス定義を変更する	230
6.9	ビジネスプロセスの検証	233
6.9.1	検証内容	233
6.9.2	検証方法	244
6.9.3	検証内容の表示	244
6.10	ビジネスプロセスの削除	246

## 7

データ変換の定義	247	
7.1	データ変換に必要なファイルおよび定義	248
7.2	電文フォーマット定義ファイルの作成	249
7.3	データ変換（マッピング）定義	250
7.3.1	データ変換（マッピング）の定義手順	250
7.4	マッピングする	253
7.4.1	変換元ノードの値を変換先ノードに直接マッピングする	253
7.4.2	変換元ノードの値を加工して変換先ノードにマッピングする	254
7.4.3	マッピングを取り消す	259
7.4.4	マッピング元の表示形式	260
7.4.5	マッピングの注意事項	261
7.5	ファンクションを使用した値の加工	262
7.5.1	複数の文字列を連結する	262
7.5.2	一つの文字列から一部分の文字列を取り出す	264
7.5.3	文字列の文字数を代入する	265
7.5.4	指定文字列の有無、または指定文字列の開始を調査する	266

7.5.5	文字列の空白を除去する	268
7.5.6	数値をフォーマット変換する	270
7.5.7	数値を演算する	271
7.5.8	小数点以下の端数を処理する	272
7.5.9	複数のノードセットのノードの数値を合計する	274
7.5.10	ノード数を代入する	275
7.5.11	ノード名を代入する	276
7.5.12	ノードの有無を調査する	277
7.5.13	くり返しを対応づける	278
7.5.14	条件によって異なる値を出力する	280
7.5.15	指定した値を代入する	283
7.5.16	変換テーブルを利用して値を変換する	285
7.5.17	変換元ノードの値にある値を付加する	288
7.5.18	変換元ノードの値を倍にする	289
7.6	くり返しの設定	291
7.6.1	[くり返し設定]ダイアログを使用してマッピングする	291
7.6.2	くり返しを合成する	292
7.6.3	くり返し依存対象の対応づけ	293
7.6.4	くり返し依存対象を確認する	297
7.6.5	くり返し依存対象を変更する	298
7.6.6	くり返し依存対象が設定された変換元ノードのパスの表示	298
7.7	ノード条件を設定する	303
7.8	マッピングの条件	305
7.8.1	マッピング対象とマッピング対象外	305
7.8.2	マッピングできるノードおよびファンクションの対応	306
7.8.3	接続できるマッピング線の数	310
8	ユーザ定義受付の定義	311
8.1	ユーザ定義受付の概要	312
8.2	定義の流れ	314
8.2.1	ビジネスプロセスを定義する前にユーザ定義受付を定義する	314
8.2.2	ビジネスプロセスの定義後にユーザ定義受付を定義する	317
8.3	WSDL の作成	320
8.3.1	WSDL の作成に使用するビジネスプロセスの例	320
8.3.2	WSDL の作成例	322
8.4	ユーザ定義受付の追加	341

8.4.1 ユーザ定義受付を新規に追加する	341
8.5 ユーザ定義受付の内容の確認	344
8.6 ユーザ定義受付の検証	345
8.6.1 検証内容	345
8.6.2 検証方法	346
8.7 ユーザ定義受付の削除	347

## 9

HCSC コンポーネントのパッケージングと配備定義	349
9.1 パッケージングと配備定義	350
9.2 パッケージング	352
9.3 HCSC コンポーネントの配備定義	353
9.3.1 配備できるクラスタ（または単一の HCSC サーバ）	354
9.3.2 クラスタに HCSC コンポーネントを追加する	355
9.3.3 クラスタから HCSC コンポーネントを削除する	356
9.4 HCSC コンポーネントの情報の参照	357
9.4.1 参照できる HCSC コンポーネントの情報	357
9.4.2 HCSC コンポーネントの情報を表示する	358
9.4.3 HCSC コンポーネントの一覧を更新する	358

## 10

サービスリクエストの作成	361
10.1 サービスリクエスト作成の概要	362
10.2 標準の同期受付（Web サービス）に要求を出すサービスリクエスト	364
10.2.1 サービスリクエスト（標準の同期受付（Web サービス））の作成手順	364
10.2.2 WSDL を取得する	366
10.2.3 スタブを生成する	367
10.2.4 オブジェクトを生成する	368
10.2.5 パラメタを設定する	371
10.2.6 要求電文を作成する	373
10.2.7 応答電文を取得する	375
10.2.8 エラー情報を取得する	377
10.2.9 ビジネスプロセスの再実行を要求するサービスリクエストの作成（Web サービス）	381
10.3 標準の同期受付（SessionBean）に要求を出すサービスリクエスト	385
10.3.1 サービスリクエスト（標準の同期受付（SessionBean））の作成手順	385
10.3.2 スタブを取得する	387

10.3.3	インスタンスを生成する	387
10.3.4	パラメタを設定する	389
10.3.5	要求電文を作成する	391
10.3.6	応答電文を取得する	392
10.3.7	エラー情報を取得する	392
10.3.8	ビジネスプロセスの再実行を要求するサービスリクエストの作成 (SessionBean)	398
10.4	標準の非同期受付 (MDB (WS-R)) に要求を出すサービスリクエスト	402
10.4.1	サービスリクエスト (標準の非同期受付 (MDB (WS-R))) の作成手順	402
10.4.2	転送キューを作成する	404
10.4.3	JMS メッセージを生成する	405
10.4.4	パラメタを設定する	406
10.4.5	要求電文を作成する	411
10.4.6	JMS メッセージを送信する	412
10.4.7	応答用のキューを設定する	413
10.4.8	応答を取り出す	414
10.4.9	応答電文を取得する	415
10.5	標準の非同期受付 (DB キュー) に要求を出すサービスリクエスト	417
10.5.1	サービスリクエスト (標準の非同期受付 (DB キュー)) の作成手順	418
10.5.2	送信用共用キューを作成する (JMS)	422
10.5.3	JMS メッセージを生成する (JMS)	423
10.5.4	バイナリデータを作成する (TP1/EE, JMS)	423
10.5.5	パラメタを設定する (TP1/EE, JMS)	427
10.5.6	要求電文を作成する (TP1/EE, JMS)	427
10.5.7	バイナリデータを受信用共用キューに設定する (TP1/EE)	429
10.5.8	JMS メッセージを送信する (JMS)	429
10.5.9	応答用のキューを設定する (TP1/EE, JMS)	429
10.5.10	応答を取り出す (TP1/EE, JMS)	431
10.5.11	応答電文を取得する (TP1/EE, JMS)	434
10.6	ユーザ定義受付 (Web サービス) に要求を出すサービスリクエスト	435
10.6.1	サービスリクエスト (ユーザ定義受付 (Web サービス)) の作成手順	436
10.6.2	WSDL を編集する	437
10.6.3	スタブを生成する	438
10.6.4	オブジェクトを生成する	438
10.6.5	応答電文を取得する	438
10.6.6	エラー情報を取得する	439
10.7	DB アダプタに要求を出すサービスリクエスト	443

10.7.1	要求電文の形式	443
10.7.2	応答電文の形式	447

# 11

開発環境で使用する画面	453
-------------	-----

11.1	画面構成	454
11.1.1	バイナリフォーマット定義ファイルを作成する画面の構成	454
11.1.2	HCSC コンポーネントおよびデータ変換定義を作成する画面の構成	456
11.2	画面一覧	458
11.2.1	バイナリフォーマット定義画面	459
11.2.2	サービアダプタ定義画面	464
11.2.3	ビジネスプロセス定義画面	484
11.2.4	データ変換(マッピング)定義画面	489
11.2.5	ユーザ定義受付定義画面	493
11.2.6	HCSC コンポーネント情報表示画面	494
11.2.7	ユーザ定義受付情報表示画面	497
11.3	バイナリフォーマット定義に関連するダイアログ	499
11.3.1	フォーマットダイアログ	499
11.3.2	単純内容要素ダイアログ	500
11.3.3	複合内容要素ダイアログ	505
11.3.4	構成要素ダイアログ	509
11.3.5	条件値設定ダイアログ	515
11.3.6	ノード選択ダイアログ(バイナリフォーマット定義)	516
11.3.7	要素選択ダイアログ	520
11.3.8	セパレータ選択ダイアログ	521
11.3.9	セパレータ追加/変更ダイアログ	522
11.4	ビジネスプロセスに関連するダイアログ	525
11.4.1	変数・相関セット一覧ダイアログ	525
11.4.2	割当相関セット群ダイアログ	528
11.4.3	変数の表示ダイアログ	529
11.4.4	電文フォーマットの取込ダイアログ	530
11.4.5	フォルト処理の割当ダイアログ	532
11.4.6	受付アクティビティダイアログ	533
11.4.7	応答アクティビティダイアログ	535
11.4.8	サービス呼出アクティビティダイアログ	537
11.4.9	Java 呼出アクティビティダイアログ	539
11.4.10	データ変換アクティビティダイアログ	541

11.4.11	代入アクティビティダイアログ	543
11.4.12	代入アクティビティサブダイアログ	544
11.4.13	無操作アクティビティダイアログ	547
11.4.14	フォルト送出アクティビティダイアログ	547
11.4.15	スコープアクティビティダイアログ	548
11.4.16	繰り返しアクティビティダイアログ	549
11.4.17	条件設定ダイアログ (繰り返しアクティビティ)	550
11.4.18	分岐アクティビティダイアログ	552
11.4.19	条件設定ダイアログ (分岐アクティビティ)	553
11.4.20	並列処理アクティビティダイアログ	555
11.4.21	リンクダイアログ	556
11.4.22	条件設定ダイアログ (並列処理アクティビティ)	557
11.4.23	パレット設定ダイアログ	558
11.5	データ変換 (マッピング) に関連するダイアログ	561
11.5.1	ルート要素選択ダイアログ	561
11.5.2	マッピング元設定ダイアログ	562
11.5.3	くり返し設定ダイアログ	564
11.5.4	ノード選択ダイアログ (データ変換)	565
11.5.5	ファンクション選択ダイアログ	566
11.5.6	ノード条件設定ダイアログ	567
11.5.7	条件設定ダイアログ (ノード条件の設定)	569
11.5.8	設定ダイアログ	572
11.5.9	文字列連結ダイアログ	574
11.5.10	部分文字列取得ダイアログ	576
11.5.11	文字数取得ダイアログ	578
11.5.12	文字列調査ダイアログ	580
11.5.13	空白除去ダイアログ	581
11.5.14	数値フォーマット変換ダイアログ	583
11.5.15	数値演算ダイアログ	585
11.5.16	端数処理ダイアログ	587
11.5.17	総和ダイアログ	588
11.5.18	ノード数取得ダイアログ	590
11.5.19	ノード名取得ダイアログ	591
11.5.20	ノード調査ダイアログ	592
11.5.21	くり返しダイアログ	594
11.5.22	ソート条件追加 / 編集ダイアログ	596
11.5.23	選択ダイアログ	597

11.5.24	条件追加 / 編集ダイアログ	600
11.5.25	定数ダイアログ	602
11.5.26	値置換えダイアログ	604
11.6	ユーザ定義受付に関連するダイアログ	608
11.6.1	ユーザ定義受付削除ダイアログ	608
11.7	バイナリフォーマット定義画面の操作	610
11.7.1	要素をコピーして貼り付ける	610
11.7.2	要素を削除する	612
11.7.3	要素を移動する	612
11.7.4	操作を元に戻す	616
11.7.5	操作をやり直す	616
11.8	ビジネスプロセス定義画面の操作	617
11.8.1	オブジェクトをコピーする	617
11.8.2	オブジェクトを切り取る	617
11.8.3	オブジェクトを貼り付ける	617
11.8.4	オブジェクトを削除する	619
11.8.5	オブジェクトを整列する	619
11.8.6	操作を元に戻す	619
11.8.7	操作をやり直す	620
11.8.8	表示を拡大する	621
11.8.9	表示を縮小する	621
11.8.10	表示を元のサイズに戻す	621
11.8.11	パレットの表示を設定する	621
11.8.12	XPath を指定する	625
11.9	データ変換 (マッピング) 定義画面の操作	629
11.9.1	操作を元に戻す	629
11.9.2	操作をやり直す	629
11.9.3	定義を保存する	629
11.9.4	マッピング線の色を変更する	629

## 12 コマンド 631

コマンドの入力形式	632
コマンドの文法の記述形式	634
csamkxmls (DB アダプタ用の XML フォーマット定義ファイルの生成)	635

<b>13</b>	<b>メッセージ</b>	<b>637</b>
13.1	メッセージの種類と形式	638
13.1.1	メッセージの種類	638
13.1.2	メッセージの形式	638
13.2	KECT02100 ~ KECT12999 のメッセージ	640
13.3	KECT13100 ~ KECT24999 のメッセージ	656
13.4	KECK72001 ~ KECK74999 のメッセージ	669
13.5	エラーメッセージダイアログが出力された場合の注意事項	673

<b>付録</b>		<b>675</b>
付録 A	下位バージョンからのリポジトリの移行	676
付録 A.1	移行の対象となるバージョン	676
付録 A.2	下位バージョンからの移行手順	676
付録 A.3	下位バージョンで開発環境と運用環境のリポジトリを共有していた場合の移行手順	679
付録 B	上流ツールとの連携で使用する BPEL のサポート範囲	683
付録 B.1	ビジネスプロセス全体の定義に関連する要素の変換	684
付録 B.2	基本アクティビティに関連する要素の変換	687
付録 B.3	構造アクティビティに関連する要素の変換	691
付録 C	Cosminexus サービスプラットフォームのサンプルプログラム	694
付録 C.1	サンプルプログラムを動かしてみましょう	694
付録 C.2	サンプルプログラムを構成している要素	711
付録 C.3	定義してみましょう	713
付録 C.4	サンプルプログラムのファイルの構成	768
付録 D	用語解説	775

<b>索引</b>		<b>783</b>
-----------	--	------------



# 1

## SOA に基づいたシステム開発の概要

この章では、SOA に基づいたシステム開発の概要について説明します。

---

1.1 概要

---

1.2 開発から実運用までの流れ

---

1.3 システム全体と開発環境の関係

---

1.4 動作環境

---

## 1.1 概要

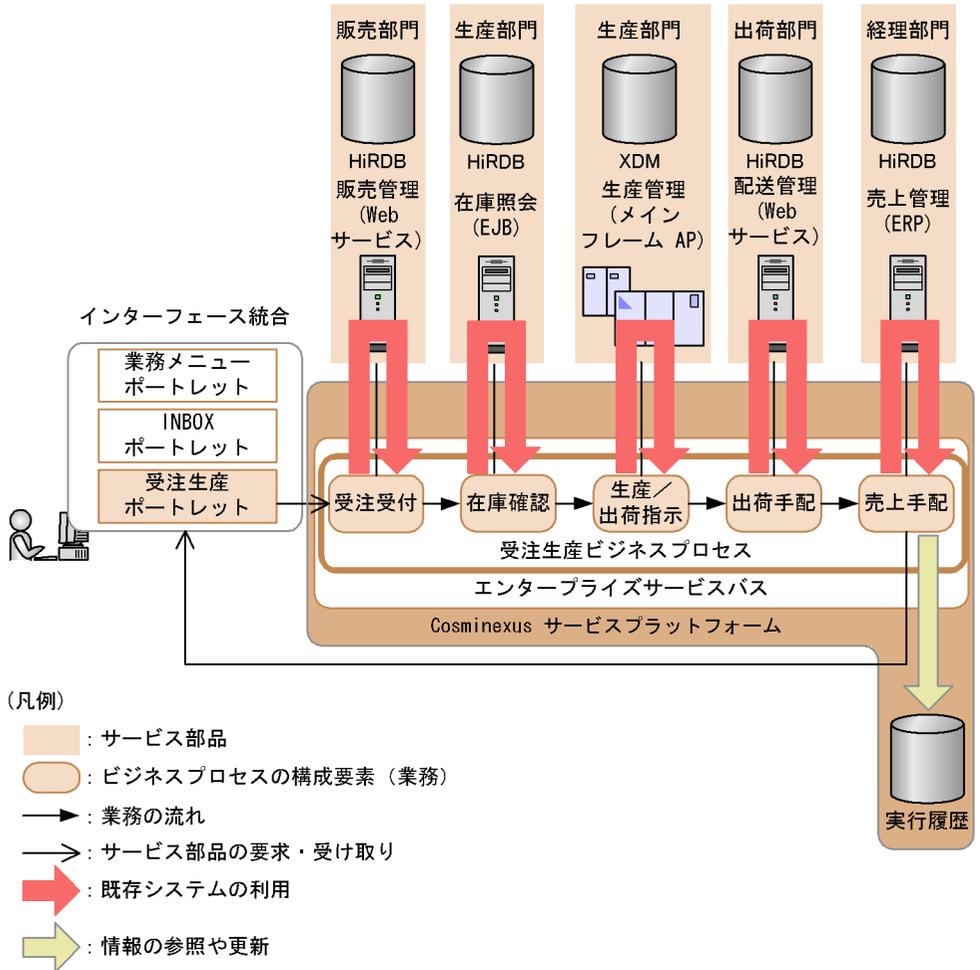
---

ここでは、Cosminexus サービスプラットフォーム、およびシステム開発環境の概要について説明します。

### 1.1.1 Cosminexus サービスプラットフォームの全体像

Cosminexus サービスプラットフォームとは、SOA を実現するシステムの開発・運用の基盤となる製品です。統一された開発・運用環境でビジネスプロセスからサービスの接続までを構築・実行できます。そのため、SOA の利点を引き出して、サービスを柔軟に組み合わせて新しいシステムを迅速に構築・実行できます。この中心となるのが、サービスをプロセスで統合する「プロセス統合」です。プロセス統合を実現するのが、Cosminexus サービスプラットフォームです。Cosminexus サービスプラットフォームを使用したシステムの実現例を次の図に示します。

図 1-1 Cosminexus サービスプラットフォームを使用したシステムの実現例



### 1.1.2 Cosminexus サービスプラットフォームを構成する環境

Cosminexus サービスプラットフォームは、次に示す三つの環境から構成されています。

- 開発環境**  
 サービスを統合するために必要な HCSC コンポーネント（開発環境で作成するサービスアダプタ、ビジネスプロセス、およびユーザ定義受付の総称）を作成して、EAR ファイルにパッケージングするための環境です。
- 実行環境**  
 要求に応じたサービス部品やビジネスプロセスを呼び出して、業務を実行するための環境です。

## 1. SOA に基づいたシステム開発の概要

- 運用環境

開発環境で作成した HCSC コンポーネントを、実行環境に配備したり、以降の運用操作を実行したりするための環境です。また、実行環境から情報を収集して、サービス部品の稼働状態を確認できます。

### 1.1.3 開発環境・実行環境・運用環境の関係

開発環境、実行環境、運用環境をそれぞれ異なるマシンに構築します。

開発環境には uCosminexus Service Architect と Eclipse を、実行環境には uCosminexus Service Platform を、運用環境には uCosminexus Operator と Eclipse をインストールします。

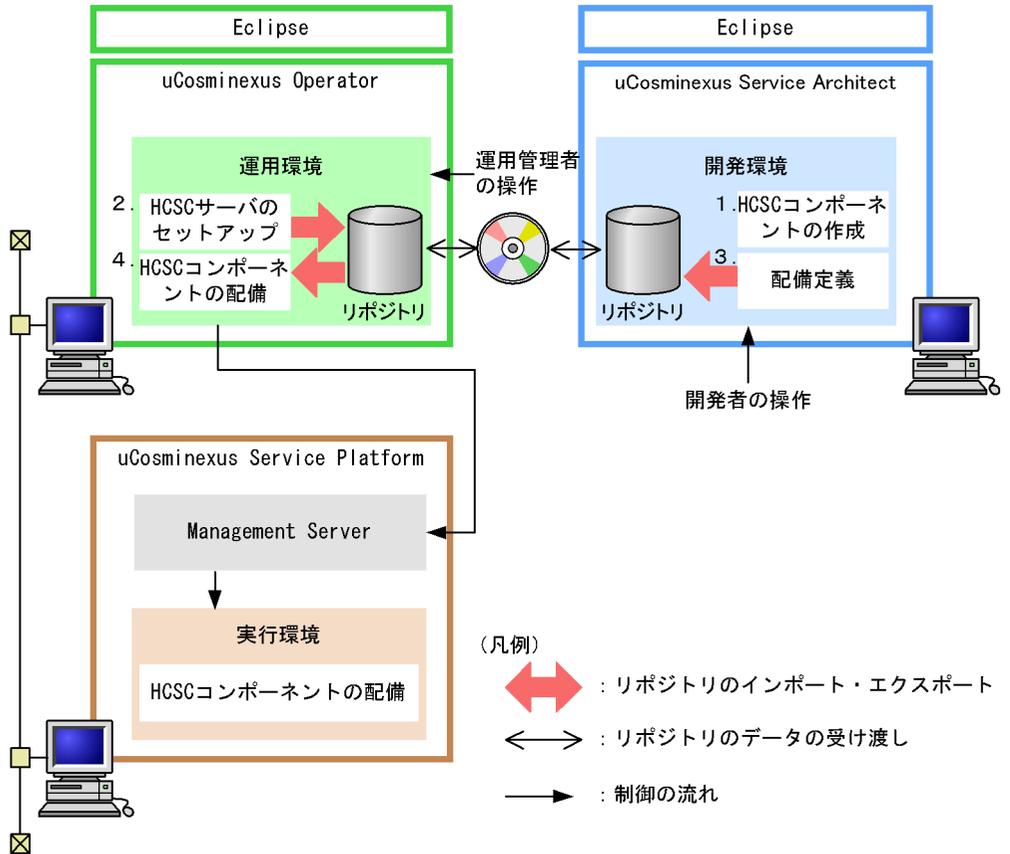
運用環境は、システム内に一つだけ構築します。また、運用の変更に対応するためにも運用環境と実行環境は、異なるマシンに構築しておくことを推奨します。

なお、運用環境から実行環境を操作する場合、Management Server を経由します。Management Server は、運用管理の対象となるサーバを一括管理するものです。HCSC サーバ（実行環境）を構築したマシンには、必ず Management Server を配置します。Management Server の機能の詳細については、マニュアル「Cosminexus 機能解説」の Management Server の機能に関する内容を参照してください。

また、運用環境から実行環境を操作するには、リポジトリというデータモデルが必要です。

Cosminexus サービスプラットフォームを構成する環境を次の図に示します。

図 1-2 Cosminexus サービスプラットフォームを構成する環境



これらの環境は、相互に連携してサービスの統合環境を実現しています。各環境は、環境構築後、次に示す流れを経て実際に運用できるようになります。

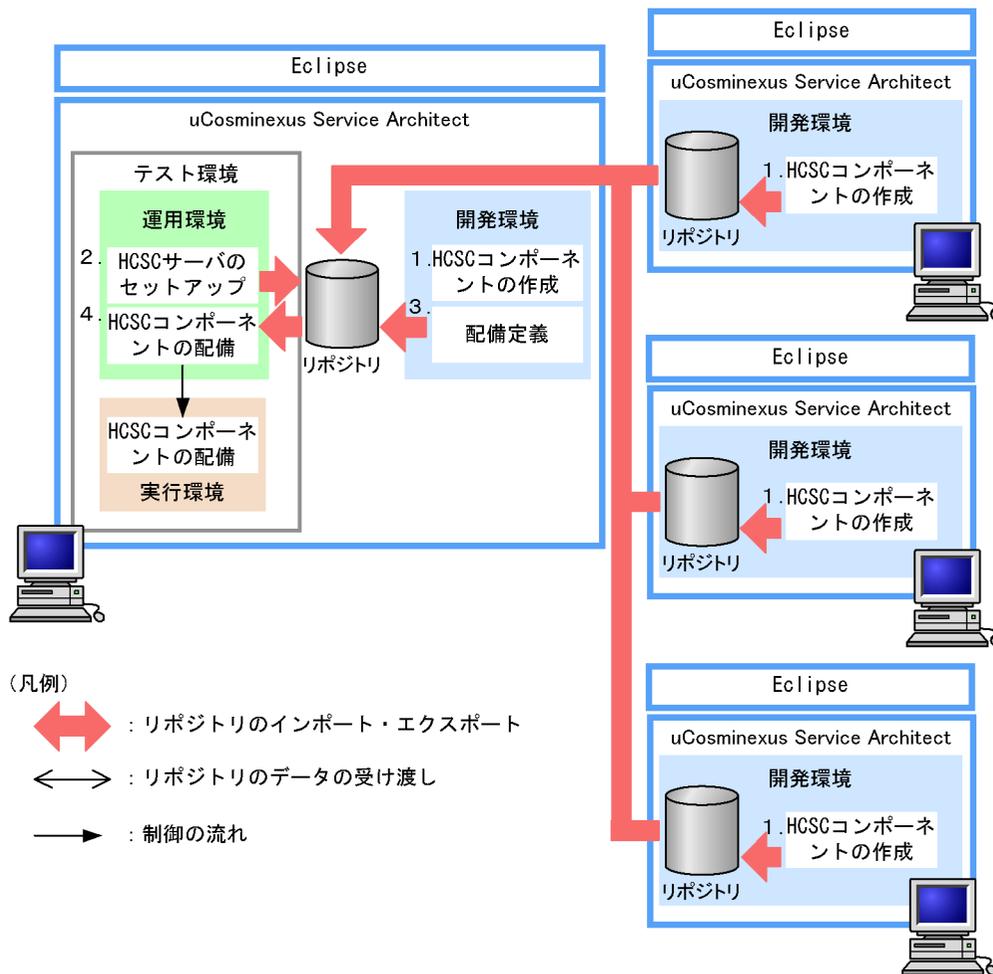
1. 開発環境で HCSC コンポーネントを作成します。
2. 運用環境で、HCSC サーバをセットアップし、システムの構成を定義します。
3. 運用環境からリポジトリをエクスポートし、運用環境で定義したシステム構成定義を開発環境にインポートします。このシステム構成定義を基に、システム構成のどこに配備するかを定義して更新します（配備定義）。
4. 開発環境で定義した配備定義を含むリポジトリをエクスポートし、運用環境にインポートします。この配備定義を基に、運用環境から実行環境にコンポーネントを配備します。

環境間の情報の受け渡しには、環境間で共有する情報を格納したリポジトリを使用します。リポジトリは媒体を経由して、ZIP ファイル形式で保存したり、読み込んだりします。

### 1.1.4 開発環境とテスト環境との関係

開発環境には、複数台のマシンで作成した HCSC コンポーネントの情報を、リポジトリを通じて一つにまとめることができます。そして、開発環境と同じマシンに、作成した HCSC コンポーネントのテスト・デバッグを実施するためのテスト環境を構築できます。テスト環境は、テストに必要な簡易的な環境で、一括構築できます。開発環境とテスト環境との関係を次の図に示します。

図 1-3 開発環境とテスト環境との関係



開発環境とテスト環境も、図 1-2 に示す場合と同様の流れを経て運用します。

1. 開発環境で HCSC コンポーネントを作成します。
2. テスト環境を構築します (HCSC 簡易セットアップ機能を使うことで、HCSC サーバのセットアップや、システムの構成が定義できます)。テスト環境の構築時にも、本番用の実行環境を想定します。

3. テスト環境からリポジトリをエクスポートし、テスト環境で定義したシステム構成定義を開発環境にインポートします。このシステム構成定義を基に、システム構成のどこに配備するかを定義し更新します（配備定義）。
4. 開発環境で定義した配備定義を含むリポジトリをエクスポートし、テスト環境にインポートします。この配備定義を基に、テスト環境にコンポーネントを配備します。

この場合のような環境間の情報の受け渡しにも、リポジトリを使用します。開発環境とテスト環境は同じマシンに構築されるため、媒体を使用しないで情報を受け渡しできます。また、テスト環境で使用したリポジトリは、本番用のシステム構成定義に合わせて配備定義をし直し、本番用の運用環境および実行環境で使用します。

### 1.1.5 テスト環境と本番環境との関係

Cosminexus サービスプラットフォームでは、まず、テスト環境を構築してテストやデバッグを実施します。テスト環境の構築には、HCSC 簡易セットアップ機能を使用できます。

実際のシステム開発では、テスト環境のほかに本番で使用する環境を構築する必要があります。そのため、テスト環境で使用したリポジトリを、本番環境に移行する必要があります。

次のようにテスト環境と本番環境で、データベースと Cosminexus RM の使用有無を同じ設定で構築した場合、テスト環境で使用したリポジトリをそのまま本番環境へ移行できます。

- テスト環境と本番環境で、データベースと Cosminexus RM の両方を使用する場合
- テスト環境と本番環境で、データベースと Cosminexus RM の両方を使用しない場合
- テスト環境と本番環境で、データベースを使用し Cosminexus RM を使用しない場合

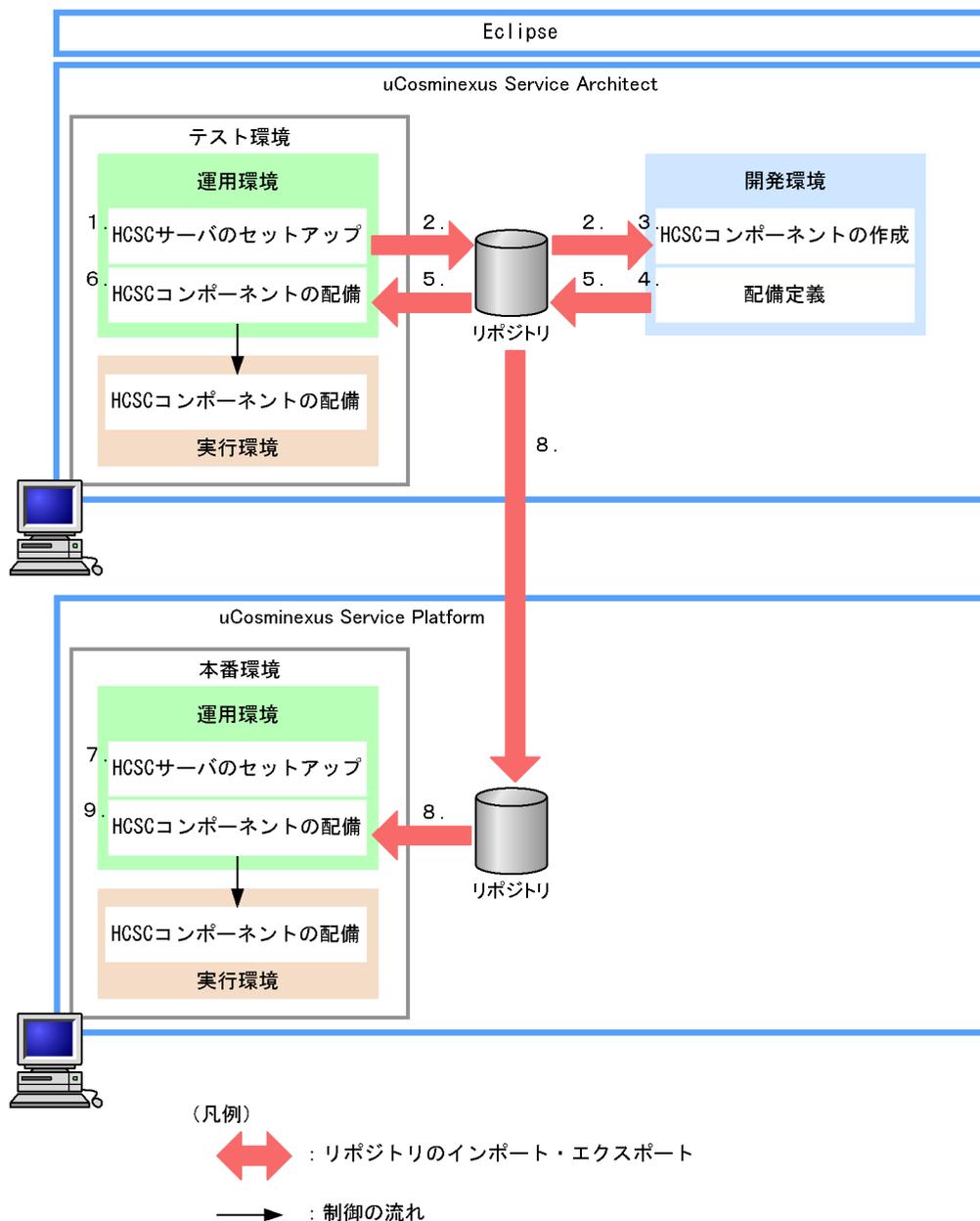
注 この構成のテスト環境は、HCSC 簡易セットアップ機能では構築できません。

なお、テスト環境と本番環境で、データベースと Cosminexus RM の使用有無の設定が同じでない場合でも、リポジトリは移行できます。詳細については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。

テスト環境と本番環境で、データベースと Cosminexus RM の使用有無を同じ設定で構築した場合の移行の流れを次の図に示します。

# 1. SOA に基づいたシステム開発の概要

図 1-4 データベースと Cosminexus RM の使用有無を同じ設定で構築した場合の移行の流れ



1. テスト環境の HCSC サーバをセットアップし、システムの構成を定義します。HCSC 簡易セットアップ機能を使うことで、HCSC サーバのセットアップや、システムの構成が定義できます。ただし、データベースを使用して Cosminexus RM を使用しない構成を構築する場合、HCSC 簡易セットアップ機能は使用できません。

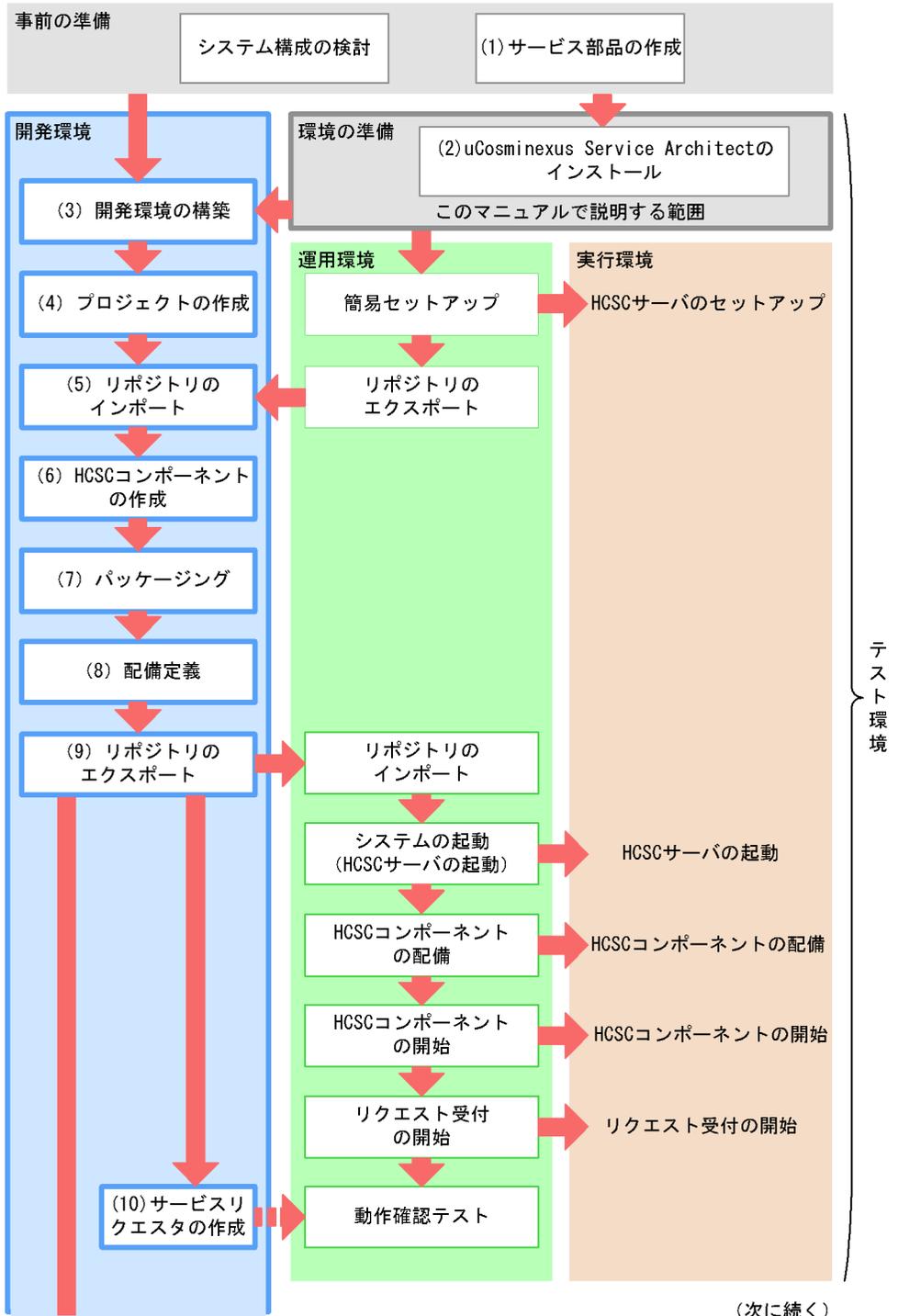
2. 運用環境からリポジトリをエクスポートし、テスト環境で定義したシステム構成定義を開発環境にインポートします。
3. 開発環境で、HCSC コンポーネントを作成します。
4. テスト環境で定義したシステム構成定義を基に、システム構成のどこに配備するかを定義し更新します（配備定義）。
5. 開発環境で定義した配備定義を含むリポジトリをエクスポートし、運用環境にインポートします。
6. 開発環境で定義した配備定義を基に、テスト環境に HCSC コンポーネントを配備します。
7. 本番環境の HCSC サーバをセットアップし、システムの構成を定義します。  
テスト環境と本番環境で、データベースと Cosminexus RM の使用有無の設定を同じにします。
8. 開発環境で定義した配備定義を含むリポジトリをエクスポートし、運用環境にインポートします。  
標準インポートを使用すると、HCSC サーバ名や IP アドレスなどが異なる場合でも、テスト環境のリポジトリをそのまま本番環境へ移行できます。
9. 開発環境で定義した配備定義を基に、本番環境に HCSC コンポーネントを配備します。

## 1.2 開発から実運用までの流れ

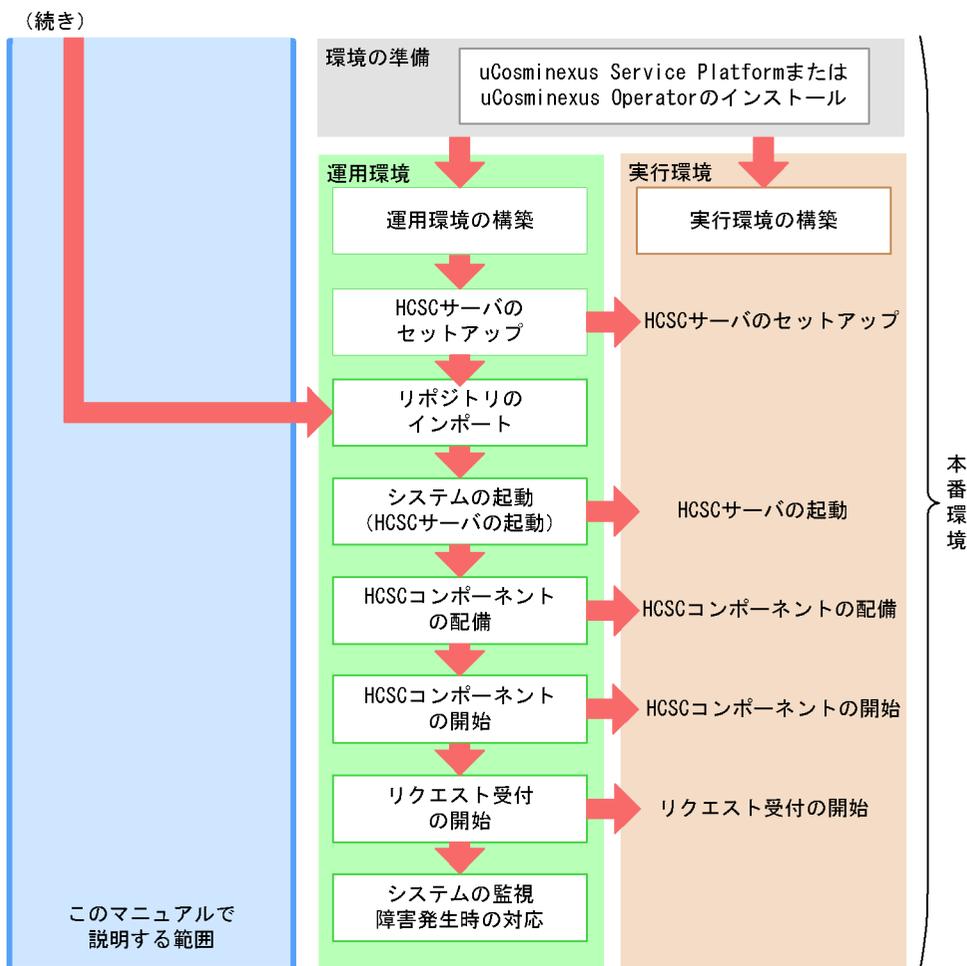
---

Cosminexus サービスプラットフォームを利用したシステムの開発から実運用までの流れを次の図に示します。

図 1-5 開発から実運用までの流れ



# 1. SOA に基づいたシステム開発の概要



このマニュアルでは、図中の「開発環境」で実施する作業の手順について説明しています。運用環境および実行環境で実施する作業については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。

図中の各作業の概要を次に説明します。

## (1) サービス部品の作成

Cosminexus サービスプラットフォームで利用するサービス部品を作成します。サービス部品にはインターフェース情報が定義されている必要があります。利用できるサービス部品の種類などについては、「2.5 利用できるサービス部品の種類と適用範囲」を参照してください。

## (2) uCosminexus Service Architect のインストール

開発環境として利用するマシンに uCosminexus Service Architect をインストールしま

す。

インストールの詳細については、「2.1.2 インストール」を参照してください。

### (3) 開発環境の構築

開発環境の構築では、Eclipse の設定ファイルの確認やプラグインの組み込みを実施します。

詳細については、「2.1.3 Eclipse の設定ファイルの確認」、および「2.1.4 Eclipse への HCSCTE プラグインの組み込み」を参照してください。

### (4) プロジェクトの作成

HCSCTE を組み込んだ Eclipse で、HCSCTE のプロジェクトを作成します。HCSC コンポーネントを開発する前に、プロジェクトを作成して、プロパティを設定します。

プロジェクトの作成の詳細については、「3.1.1 プロジェクトを作成する」を、プロパティの設定の詳細については、「3.1.2 プロパティを設定する」を参照してください。

### (5) リポジトリのインポート

運用環境で簡易セットアップをして設定したリポジトリを、開発環境にインポートします。

リポジトリのインポート方法の詳細については、「3.2.3 リポジトリをインポートする」を参照してください。

### (6) HCSC コンポーネントの作成

事前に作成したサービス部品を呼び出すアダプタ、複数のサービス部品を呼び出すビジネスプロセス、サービスリクエストからの実行要求を受け付けるユーザ定義受付などの HCSC コンポーネントを作成します。

なお、それぞれの HCSC コンポーネントを定義する前に、サービス部品を呼び出す電文の形式である電文フォーマットを作成します。

#### (a) 電文フォーマットの作成

サービス部品を呼び出す電文の形式（電文フォーマット）を作成します。電文フォーマットは、呼び出すサービス部品の種類および利用する電文の種類で作成方法が異なります。

電文フォーマットの作成については、「4. 電文フォーマットの作成」を参照してください。また、電文フォーマットを作成するときに使用する画面、ダイアログの詳細については、「11. 開発環境で使用する画面」を参照してください。

なお、DB アダプタ（データベースを操作するためのアダプタ）を利用する場合の電文フォーマットの作成には、コマンドを利用します。コマンドの詳細については、「12.

## 1. SOA に基づいたシステム開発の概要

コマンド」を参照してください。

### (b) アダプタの定義

サービス部品（データベースの操作を含む）を呼び出すためのアダプタを作成し、詳細を定義します。アダプタには、サービス部品を呼び出すためのサービスアダプタ、データベースを操作するための DB アダプタがあります。

サービスアダプタ、DB アダプタの定義の詳細については、「5. アダプタの定義」を参照してください。アダプタを定義するときに使用する画面、ダイアログの詳細については、「11. 開発環境で使用する画面」を参照してください。

### (c) ビジネスプロセスの定義

ビジネスプロセスを作成し、詳細を定義します。

ビジネスプロセスの定義の詳細については、「6. ビジネスプロセスの定義」を参照してください。

また、ビジネスプロセスを定義するときに使用する画面、ダイアログの詳細については、「11. 開発環境で使用する画面」を参照してください。

### (d) データ変換定義の作成

サービス部品を呼び出す要求電文の電文フォーマットと、呼び出すサービス部品側の電文フォーマットが異なる場合には、データ変換定義を作成します。

データ変換定義の作成の詳細については、「7. データ変換の定義」を参照してください

また、データ変換定義を作成するときに使用する画面、ダイアログの詳細については、「11. 開発環境で使用する画面」を参照してください。

### (e) ユーザ定義受付の定義

サービス部品実行要求を受け付け、応答を返すインターフェースとして任意の形式を定義する場合、ユーザ定義受付を定義します。サービスリクエストとユーザ定義受付の通信プロトコルは Web サービスだけです。また、ユーザ定義受付でリクエストを受け付けることができる HCSC コンポーネントは、ビジネスプロセスだけです。

ユーザ定義受付の定義の詳細については、「8. ユーザ定義受付の定義」を参照してください

また、ユーザ定義受付を定義するときに使用する画面、ダイアログの詳細については、「11. 開発環境で使用する画面」を参照してください。

## (7) パッケージング

作成した HCSC コンポーネントは、EAR ファイルにパッケージングして、運用環境から実行環境に配備できる状態にします。

HCSC コンポーネントのパッケージングについては、「9. HCSC コンポーネントのパッ

「パッケージングと配備定義」を参照してください。

#### (8) 配備定義

パッケージングした HCSC コンポーネントを、運用環境から実行環境にどのように配備するかを定義（配備定義）し、システム構成定義を更新します。

配備定義する方法については、「9. HCSC コンポーネントのパッケージングと配備定義」を参照してください。

#### (9) リポジトリのエクスポート

HCSC コンポーネントの作成、パッケージング、および配備定義をしたリポジトリを、運用環境にエクスポートします。

リポジトリのエクスポート方法の詳細については、「3.2.2 リポジトリをエクスポートする」を参照してください。

#### (10) サービスリクエストの作成

実行環境で業務担当者からのサービス部品の実行要求を受け付け、HCSC コンポーネントに実行要求を送信するためのサービスリクエストを作成します。

サービスリクエストの作成の詳細については、「10. サービスリクエストの作成」を参照してください。

## 1.3 システム全体と開発環境の関係

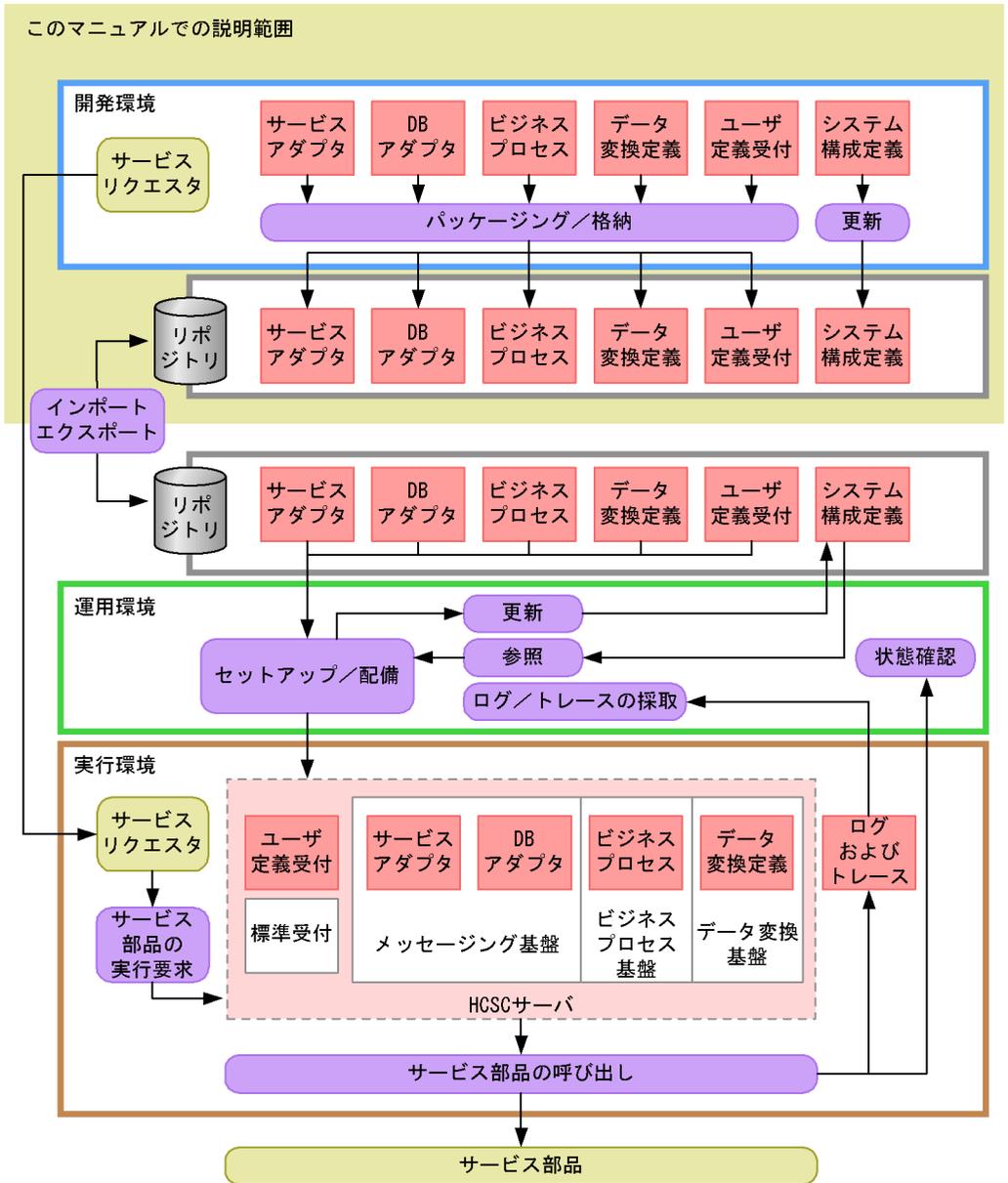
---

Cosminexus サービスプラットフォームでは、開発環境、運用環境および実行環境が相互に関連してシステム全体を構成します。その中での開発環境の位置づけについて説明します。

### (1) 運用上の位置づけ

システム全体の運用の中で、開発環境がどのように位置づけられるかを、次の図に示します。

図 1-6 運用上の位置づけ



(凡例)

- : 作成、格納されるデータ
- : ユーザーが準備するプログラム
- : 実施する操作
- : データの流れ

開発環境では、作成した HCSC コンポーネントや配備定義をリポジトリに格納して、運用環境にデータを受け渡します。開発環境で作成し、リポジトリに格納されたデータは、運用環境から実行環境に配備されます。

## 1. SOA に基づいたシステム開発の概要

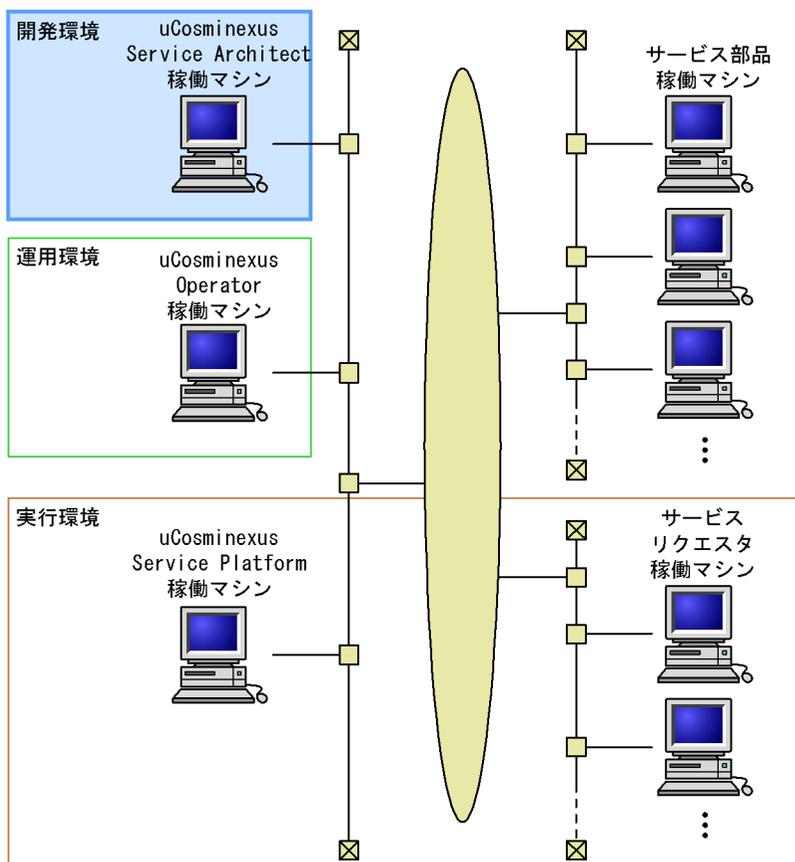
なお、配備定義の作成には、あらかじめ運用環境でセットアップした HCSC サーバの情報などが必要になります。この情報はリポジトリを利用して運用環境から取得する必要があります。リポジトリを利用した運用環境との情報の受け渡しについては、「3.2 リポジトリの管理」を参照してください。

### (2) ネットワーク上の位置づけ

開発環境、運用環境および実行環境は、インターネットやイントラネットなどのネットワークを使って連結してシステムを構成します。

システム全体における開発環境のネットワーク上の位置づけについて次の図に示します。

図 1-7 ネットワーク上の位置づけ



注 上の例は、uCosminexus Operatorを利用して、運用環境と実行環境を異なるマシンに構築した場合の例です。uCosminexus Service Platformが稼働しているマシンの一つに、運用環境と実行環境を構築することもできます。また、開発環境は別のネットワーク上に構築することもできます。

## 1.4 動作環境

Cosminexus サービスプラットフォームの開発環境（uCosminexus Service Architect）を利用する場合の動作環境について説明します。

### 1.4.1 前提 OS

uCosminexus Service Architect の前提 OS について、次の表に示します。

表 1-1 uCosminexus Service Architect の前提 OS

前提 OS	前提 OS のバージョン
Windows	Windows Server 2003 Windows Server 2003 R2 Windows XP Windows Vista

### 1.4.2 ソフトウェアの構成

uCosminexus Service Architect は、構成ソフトウェアと呼ばれる複数のソフトウェアで構成されています。uCosminexus Service Architect を構成するソフトウェアとその位置づけについて説明します。

#### （1）構成ソフトウェア

##### （a）構成ソフトウェアの種類

uCosminexus Service Architect を構成するソフトウェアを次に示します。

- Cosminexus Application Development Plug-in
- Cosminexus cFramework
- Cosminexus Component Container（DB Connector も含む）
- Cosminexus Component Transaction Monitor
- Cosminexus DABroker Library
- Cosminexus Developer's Kit for Java™
- Cosminexus Performance Tracer
- Cosminexus Reliable Messaging
- Cosminexus Service Coordinator
- Cosminexus Service Development Plug-in
- Cosminexus TPBroker
- Cosminexus Web Services - Security
- Cosminexus XML Processor
- HiRDB Embedded Server Version 8
- Hitachi Web Server

## 1. SOA に基づいたシステム開発の概要

- MyEclipse for Cosminexus

Cosminexus サービスプラットフォームの開発環境でアダプタ、ビジネスプロセス、データ変換などを定義する機能は、Cosminexus Service Development Plug-in で提供されます。

そのほかの構成ソフトウェアの機能の詳細については、それぞれの関連マニュアルを参照してください。

### **!** 注意事項

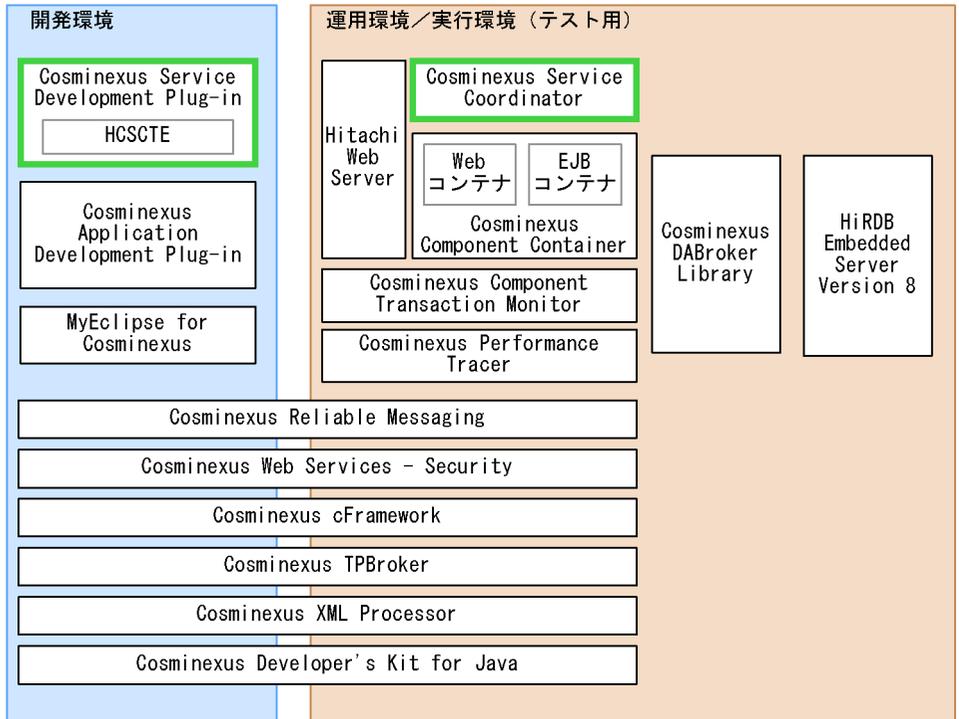
Cosminexus Service Coordinator は、Cosminexus サービスプラットフォームの運用環境および実行環境で利用する機能を提供する構成ソフトウェアです。

ただし、uCosminexus Service Architect に含まれる Cosminexus Service Coordinator は開発した HCSC コンポーネントのテスト・デバッグ用の運用環境および実行環境だけを構築できます。業務システムの実際の運用には使用できません。

#### (b) 構成ソフトウェアの位置づけ

uCosminexus Service Architect を構成するソフトウェアの位置づけを次の図に示します。

図 1-8 uCosminexus Service Architect を構成するソフトウェアの位置づけ



(凡例)

- : Cosminexus サービスプラットフォームの中心となる構成ソフトウェア
- : そのほかの構成ソフトウェア

## (2) 関連ソフトウェア

uCosminexus Service Architect の関連ソフトウェアについて説明します。

### (a) uCosminexus TP1/Server Base Enterprise Option

Cosminexus サービスプラットフォームの実行環境と OpenTP1 の間で、データベースを介した異なるコンポーネントでの通信を行う場合に利用します。

TP1/Server Base Enterprise Option は DB キューのプロトコルをサポートしています。Cosminexus サービスプラットフォームの実行環境とは、DB キューの受け付けとサービスアダプタで連携できます。

### (b) uCosminexus 日立コード変換 - Development Kit

Cosminexus サービスプラットフォームの開発環境で、文字コードを変換するためのマッピングファイルを作成する場合に利用します。パイナリ形式 (XML 形式以外) のデータを扱うとき、またはデータ変換で値置換えファンクションを使用するときに必要な

## 1. SOA に基づいたシステム開発の概要

です。

### (c) uCosminexus 日立コード変換 - Server Runtime

Cosminexus サービスプラットフォームの実行環境で、文字コードを変換する場合に利用します。Windows の場合で、バイナリ形式 (XML 形式以外) のデータを扱うとき、またはデータ変換で値置換えファンクションを使用するときに必要です。

### (d) uCosminexus 日立コード変換 - Runtime

Cosminexus サービスプラットフォームの実行環境で、文字コードを変換する場合に利用します。UNIX の場合で、バイナリ形式 (XML 形式以外) のデータを扱うとき、またはデータ変換で値置換えファンクションを使用するときに必要です。

### (e) uCosminexus Service Adapter Architect for TP1

Cosminexus サービスプラットフォームの開発環境で、既存の OpenTP1 システムや XDM/DCCM3 内にあるサービス部品を呼び出すためのアダプタを作成する場合に利用します。

### (f) uCosminexus Service Adapter for TP1

Cosminexus サービスプラットフォームの実行環境から既存の OpenTP1 システムや XDM/DCCM3 内にあるサービス部品を呼び出す場合に利用します。

### (g) uCosminexus TP1 Gateway

Cosminexus サービスプラットフォームの実行環境で、OpenTP1 などの既存システムから、HCSC サーバを経由してサービス部品を呼び出す場合に利用します。HCSC サーバに対して、CTM を経由した同期応答型の送信、または DB キューを介した非同期応答型の送信ができます。

### (h) uCosminexus Service Adapter Architect for Flat Files

Cosminexus サービスプラットフォームの開発環境で、既存の業務処理システムのファイルからデータを読み込んだり、ファイルへデータを書き込んだりするためのアダプタを作成する場合に利用します。

### (i) uCosminexus Service Adapter for Flat Files

Cosminexus サービスプラットフォームの実行環境で、既存の業務処理システムのファイルからデータを読み込んだり、ファイルへデータを書き込んだりする場合に利用します。ファイルインターフェースを持つシステムをサービスプラットフォームと連携する場合でも、連携先システムのインターフェースを変更することなく、ファイルアクセスをサービス部品として実行でき、ファイルの入出力を伴う既存システムとほかのサービス部品との連携ができるようになります。

(j) uCosminexus Service Adapter Architect for Object Access

Cosminexus サービスプラットフォームの開発環境で、既存の TPBroker システム (Object Wrapper システム) の CORBA サーバの業務メソッドを呼び出すためのアダプタを作成する場合に利用します。

(k) uCosminexus Service Adapter for Object Access

Cosminexus サービスプラットフォームの実行環境で、既存の TPBroker システム (Object Wrapper システム) で動作する CORBA サーバの業務メソッドをサービス部品として連携する場合に利用します。

(l) iWay アダプタ

Cosminexus サービスプラットフォームの実行環境で、他社 ERP パッケージをサービス部品として連携する場合に利用します。SAP R/3, および Oracle EBS との Web サービスによる連携ができるため、既存 ERP 資産を活用したシステム間連携ができます。



# 2

## システムを開発する前に

この章では，Cosminexus サービスプラットフォームのシステムを開発する前に実施する作業，および知っておく必要のある内容について説明します。

---

2.1 開発環境を使用するためのセットアップ

---

2.2 構成の形態と選択

---

2.3 開発手順

---

2.4 テスト環境の簡易セットアップ

---

2.5 利用できるサービス部品の種類と適用範囲

---

## 2.1 開発環境を使用するためのセットアップ

---

Cosminexus サービスプラットフォームの開発環境を使用するためのセットアップ手順を次に示します。

1. uCosminexus Service Architect をインストールします。  
詳細は、「2.1.2 インストール」を参照してください。
2. MyEclipse for Cosminexus をセットアップします。  
詳細は、マニュアル「Cosminexus アプリケーション開発ガイド」のインスタントセットアップ機能を使用しない環境の構築手順に関する内容を参照してください。
3. Eclipse の設定ファイルを確認します。  
詳細は、「2.1.3 Eclipse の設定ファイルの確認」を参照してください。
4. HCSCTE プラグインを Eclipse へ組み込みます。  
詳細は、「2.1.4 Eclipse への HCSCTE プラグインの組み込み」を参照してください。
5. 環境設定をします。  
詳細は、「2.1.5 環境設定」を参照してください。
6. XML データの電文フォーマットとバイナリデータの電文フォーマットのデータ変換を定義する場合は、uCosminexus 日立コード変換 - Development Kit をセットアップします。  
詳細は、「2.1.6 uCosminexus 日立コード変換 - Development Kit のセットアップ」を参照してください。
7. 必要に応じてテスト環境を構築します。  
HCSC 簡易セットアップ機能を利用する場合  
詳細は、「2.4 テスト環境の簡易セットアップ」を参照してください。  
HCSC 簡易セットアップ機能を利用しない場合  
マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照して、運用・実行環境を構築してください。  
また、テスト環境で組み込みデータベースを利用する場合、「2.1.7 組み込みデータベースの構築」を参照してください。

**!** 注意事項

テスト環境では、HCSC サーバ (J2EE サーバ) に、テストで使用するアプリケーション (サービスリクエストおよびサービス部品) をデプロイできます。ただし、次に示す名称は予約されているため、デプロイするアプリケーションの名称として利用できません。

- 「CSC」で始まる名称
- ビジネスプロセス、サービスアダプタのサービス ID
- ユーザ定義受付の受付 ID

## 2.1.1 前提条件

Cosminexus サービスプラットフォームの開発環境を使用するための前提条件を次に示します。

### 開発環境を利用するユーザ

開発環境を利用するユーザは、「Administrators」または「Power Users」グループに属している必要があります。

### インストールおよびセットアップが必要なソフトウェア

開発環境を使用する前に、次のソフトウェアがインストールおよびセットアップされている必要があります。

- MyEclipse for Cosminexus
- WSDL4J 1.5.1

### SOAP 通信基盤の動作モード

Cosminexus が提供している SOAP 通信基盤の動作モードは、標準モードにする必要があります。次のコマンドを実行すると、動作モードを標準モードに設定できます。

```
<Cosminexusインストールディレクトリ>%c4web%bin%setup_mode std
```

## 2.1.2 インストール

uCosminexus Service Architect は、日立総合インストーラからインストールします。日立総合インストーラの指示に従ってインストールしてください。なお、インストール作業は Administrator 権限が設定されたユーザが実施してください。

インストール先を指定する場合、半角英数字で 50 文字までのパス名を指定してください。

インストール種別を選択する画面では「標準」を選択してください。

uCosminexus Service Architect のインストール終了後、再起動してください。

## 2. システムを開発する前に

### ポイント

---

以前のバージョンからバージョンアップする場合、バージョンアップ前の環境で使用していたリポジトリ情報を、バージョンアップ後も使用できる形式に移行できます。

移行の方法については「付録 A 下位バージョンからのリポジトリの移行」を参照してください。

---

### 2.1.3 Eclipse の設定ファイルの確認

Eclipse の設定ファイルを確認します。eclipse.ini ファイルに次の引数が設定されていることを確認してください。

eclipse.ini ファイルの格納先

```
<Eclipse のインストールディレクトリ>¥eclipse.ini
```

引数

```
-vm  
<Cosminexusのインストールディレクトリ>¥jdk¥bin¥javaw.exe  
-Djava.endorsed.dirs=<Cosminexusのインストールディレクトリ>¥jaxp¥lib
```

引数が設定されている eclipse.ini ファイルの例

```
-vm  
C:¥Program Files¥Hitachi¥Cosminexus¥jdk¥bin¥javaw.exe  
-product  
com.genuitec.myeclipse.product.ide  
-vmargs  
-Xms128M  
-Xmx512M  
-XX:PermSize=64M  
-XX:MaxPermSize=128M  
-Dosgi.splashLocation=C:¥MyEclipse¥eclipse¥MyEclipseSplash.bmp  
-Djava.endorsed.dirs=C:¥Program  
Files¥Hitachi¥Cosminexus¥jaxp¥lib
```

### 2.1.4 Eclipse への HCSCTE プラグインの組み込み

HCSCTE プラグインを Eclipse に組み込むには、MyEclipse for Cosminexus をセットアップして、Server Plug-in (<Cosminexus のインストールディレクトリ>¥plugins¥eclipse) を Eclipse に組み込みます。Server Plug-in を Eclipse に組み込む方法については、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。

HCSCTE プラグインが Eclipse に正しく組み込まれたかどうかは、次に示す方法で確認してください。

1. メニューから、[ヘルプ] - [ソフトウェアの更新] - [構成の管理] を選択します。

[製品構成] ダイアログが表示されます。

2. ダイアログ左側に表示されている <Cosminexus インストールディレクトリ> のツリーを展開します。
3. ツリーに表示されている「Cosminexus Service Coordinator Tools for Eclipse」のバージョン番号が「7.6.0」になっていることを確認します。

Cosminexus Service Coordinator Tools for Eclipse のバージョン番号が「7.6.0」ではない場合は、次に示す方法でバージョンを変更してください。

1. メニューから、[ヘルプ] - [ソフトウェアの更新] - [構成の管理] を選択します。  
[製品構成] ダイアログが表示されます。
2. ダイアログ左側に表示されている <Cosminexus インストールディレクトリ> のツリーを展開します。
3. ツリーに表示されている「Cosminexus Service Coordinator Tools for Eclipse」を選択します。
4. ダイアログ右側に表示されている [別のバージョンとの置換] をクリックします。  
[製品構成] ダイアログ (別のバージョンとの置換) が表示されます。
5. 「Cosminexus Service Coordinator Tools for Eclipse 7.6.0」を選択し、[終了] ボタンをクリックします。
6. Eclipse を再起動します。
7. メニューから、[ヘルプ] - [ソフトウェアの更新] - [構成の管理] を選択します。  
[製品構成] ダイアログが表示されます。
8. ダイアログ左側に表示されている <Cosminexus インストールディレクトリ> のツリーを展開します。
9. 「Cosminexus Service Coordinator Tools for Eclipse」のバージョン番号が「7.6.0」になっていることを確認します。

Eclipse に組み込まれた HCSCTE プラグインは、使用不可の状態に設定できます。HCSCTE プラグインを使用不可の状態に設定する方法については、マニュアル「Cosminexus アプリケーション開発ガイド」のプラグイン使用不可の設定に関する内容を参照してください。

#### **!** 注意事項

HCSCTE プラグインがインストールされた複数の Eclipse インスタンスは、同時に起動できません。

## 2.1.5 環境設定

Eclipse を組み込んだ HCSCTE の環境を設定します。環境設定では、次の作業を実施します。

- HCSCTE プロジェクトの作成
- プロパティの設定 (リポジトリの設定および検証設定)

環境設定の詳細については、「3. プロジェクトの管理とリポジトリの管理」を参照してください。

### 参考

電文フォーマットを表示する場合で、電文フォーマット定義ファイルにマルチバイト文字が含まれているときは、次の操作を行って、エンコードを「UTF-8」に変更してください。

1. [ ウィンドウ ] - [ 設定 ] - [ 一般 ] - [ ワークスペース ] - [ テキスト・ファイル・エンコード ] を選択します。
2. 「その他 (UTF-8)」を設定します。

## 2.1.6 uCosminexus 日立コード変換 - Development Kit のセットアップ

XML データの電文フォーマットとバイナリデータの電文フォーマットのデータ変換を定義する場合、uCosminexus 日立コード変換 - Development Kit をセットアップします。

### (1) インストール

uCosminexus 日立コード変換 - Development Kit をインストールします。

インストール方法の詳細は、uCosminexus 日立コード変換 - Development Kit に添付されているドキュメントを参照してください。

### (2) 環境設定

環境設定として、次の作業を実施します。

uCosminexus 日立コード変換 - Development Kit の jar ファイルのコピー

uCosminexus 日立コード変換 - Development Kit の jar ファイルを uCosminexus Service Architect のインストールディレクトリにコピーします。コピー先を次に示します。

```
<Cosminexusのインストールディレクトリ>¥CSCTE¥lib
```

変換テーブル (ユーザマッピングファイル) の格納パスの追加

Eclipse の初期設定ファイル (eclipse.ini) に次の定義を追加します。

```
-Dcscte.dt.codetablepath=<変換テーブルの格納パス>
```

例えば、uCosminexus 日立コード変換 - Development Kit を「C:\Program Files\HITACHI\CodeConvJ」にインストールした場合、次のように定義を追加します。

```
-Dcscte.dt.codetablepath= C:\Program Files\HITACHI\CodeConvJ\Table
```

### (3) ユーザマッピングファイル（外字ファイル）の作成と格納

uCosminexus 日立コード変換 - Development Kit でユーザマッピングファイルを作成し、所定のフォルダに格納します。

#### ユーザマッピングファイルの作成

ユーザマッピングファイルの作成については、uCosminexus 日立コード変換 - Development Kit に添付されているドキュメントを参照してください。

#### 作成時の注意事項

- ユーザマッピングファイルのファイル名は、uCosminexus 日立コード変換 - Development Kit で定義されているデフォルトのファイル名にします。
- 必ず Unicode へのマッピングファイルを作成します。例えば、JIS 漢字から KEIS へのマッピングを定義する場合、「JIS 漢字から Unicode」と「Unicode から KEIS」のユーザマッピングを定義する必要があります。

#### ユーザマッピングファイルの格納

作成したユーザマッピングファイルは、次のディレクトリに格納します。

#### 開発環境

「2.1.6(2) 環境設定」で Eclipse の初期設定ファイル（eclipse.ini）に追加したパス（-Dcscte.dt.codetablepath に指定したパス）に格納します。

#### 実行環境

実行環境で uCosminexus 日立コード変換 - Server Runtime（Windows の場合）、または uCosminexus 日立コード変換 - Runtime（UNIX の場合）をセットアップするとき、システムプロパティファイル（usrconf.properties）に追加したパスに格納します。

実行環境での uCosminexus 日立コード変換 - Server Runtime、または uCosminexus 日立コード変換 - Runtime のセットアップについては、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の uCosminexus 日立コード変換 - Server Runtime または uCosminexus 日立コード変換 - Runtime のセットアップに関する内容を参照してください。

## 2.1.7 組み込みデータベースの構築

uCosminexus Service Architect には、開発した HCSC コンポーネントをテストするためのテスト環境として利用できる、組み込みデータベースが含まれています。

開発環境でのテスト用に HCSC サーバを構築する場合、組み込みデータベースを利用すると、別途 DB サーバを用意・構築する必要がありません。

### ！ 注意事項

- HCSC 簡易セットアップ機能を利用してテスト環境をセットアップする場合、以降で説明する組み込みデータベースの設定を行う必要はありません。  
HCSC 簡易セットアップ機能の詳細については、「2.4 テスト環境の簡易セットアップ」を参照してください。
- テスト用の HCSC サーバと組み込みデータベースは、開発環境の同じマシン上に構築してください。ほかのマシンに DB サーバを構築して組み込みデータベースを利用することはできません。  
また、組み込みデータベースは HiRDB を基にしています。ほかのデータベース (Oracle) を使用する場合は、ほかのマシンに DB サーバを構築してください。

組み込みデータベースを利用する場合に必要な設定について、以降で説明します。

### (1) DB サーバ側の設定

組み込みデータベースは、uCosminexus Service Architect をインストールした段階でインストールされます。そのため、組み込みデータベースそのもののインストールは必要ありません。

組み込みデータベースを利用するには、次の作業が必要です。

- HiRDB SQL Executer のインストールおよび環境変数の設定
- DB サーバの構築および環境設定
- ユーザの設定およびスキーマの定義
- RD エリアの準備
- 組み込みデータベースの再起動

各作業について、次に説明します。

#### (a) HiRDB SQL Executer のインストールおよび環境変数の設定

組み込みデータベースに含まれる HiRDB SQL Executer をインストールします。

組み込みデータベースを構築する場合の HiRDB SQL Executer のインストール方法については、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。

インストールが完了したら、次に示す環境変数 PATH を追加・設定します。

```
<Cosminexusのインストールディレクトリ>%DB%BIN;
<Cosminexusのインストールディレクトリ>%DB%CLIENT%UTL;
```

#### (b) DB サーバの構築および環境設定

組み込みデータベースのDBサーバの構築、環境設定には、Eclipse を利用します。

DBサーバの構築および環境設定の方法については、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。

なお、DBサーバ構築の設定画面では、構築するデータベースのサイズとして [ Large (2GB) ] を選択します。

#### (c) ユーザの設定およびスキーマの定義

データベースを利用するユーザとスキーマを定義します。

組み込みデータベースのユーザの設定およびスキーマの定義の方法については、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。

#### (d) RD エリアの準備

HCSC Messaging の管理情報テーブルを格納するために、必要に応じて RD エリアを作成、追加または拡張します。また、RD エリアを作成、追加または拡張した場合に、正しく処理されたかどうか確認します。

RD エリアを作成、追加または拡張する手順について、次に示します。

1. Windows の [ スタート ] メニューから [ Cosminexus ] - [ Application Development Plug-in ] - [ 組み込み DB プロンプト ] を選択します。  
組み込みデータベース専用コマンドプロンプトが起動されます。
2. 作業に応じて、コマンドプロンプトで次のコマンドを実行します。
  - pdfmkfs コマンド ( RD エリアを作成する場合 )
  - pdmod コマンド ( RD エリアを追加、拡張する場合 )
  - pddbls コマンド ( RD エリアの状態を表示する場合 )

上記のコマンドについては、マニュアル「スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス ( Windows(R) 用 )」を参照してください。

3. RD エリアを追加・拡張した場合、必要に応じて組み込みデータベースの DB 定義ファイル ( HiRDB システム定義ファイル ) の pdbuffer オペランドを追加して、組み込みデータベースのグローバルバッファを拡張します。  
組み込みデータベースの DB 定義ファイルは、DBサーバの構築時に指定した DB 定義ファイルディレクトリの配下にあります。

#### (e) 組み込みデータベースの再起動

組み込みデータベースを停止して、再度起動します。

## 2. システムを開発する前に

組み込みデータベースの停止および起動の方法については、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。

### (2) DB クライアント (HCSC サーバ) 側の設定

組み込みデータベースを利用する場合、DB クライアント (HCSC サーバ) の設定として、次の作業が必要です。

- DB クライアントの環境変数の設定
- 環境変数グループの設定
- 組み込みデータベースの同時接続数の設定
- Cosminexus RM 用のユーザ権限の設定および RD エリアの準備

各作業について、次に説明します。

#### (a) DB クライアントの環境変数の設定

環境変数 PDXAMODE および PDTXACANUM を設定します。

#### (b) 環境変数グループの設定

次に示すパスの「クライアント環境変数登録ツール」で環境変数のグループを登録します。

<Cosminexusのインストールディレクトリ>¥DB¥CLIENT¥UTL

クライアント環境変数登録ツールで環境変数グループを登録する方法の詳細については、マニュアル「スケーラブルデータベースサーバ HiRDB Version 8 UAP 開発ガイド」を参照してください。

なお、登録した環境変数グループ名は、DB Connector のプロパティの XA\_OPEN 文字列で指定します。DB Connector のプロパティの XA\_OPEN 文字列については、マニュアル「Cosminexus アプリケーション設定操作ガイド」の DB Connector のプロパティの定義に関する内容を参照してください。

#### (c) 組み込みデータベースの同時接続数の設定

必要に応じて、組み込みデータベースの同時接続数を増やします。組み込みデータベースの DB 定義ファイルのオペランドを変更します。DB 定義ファイルは、「2.1.7(1)(b) DB サーバの構築および環境設定」の作業を行うときに、DB サーバ構築の設定画面で設定するディレクトリにあります。

DB 定義ファイルで変更するオペランドを次に示します。

- pd\_max\_users (最大同時接続数)
- pd\_max\_server\_process (最大同時起動サーバプロセス数)
- pd\_process\_count (常駐プロセス数)

あわせて、環境変数 PDXACANUM の値は、pd\_max\_users の指定値以上に設定します。

#### (d) Cosminexus RM 用のユーザ権限の設定および RD エリアの準備

必要に応じて、Cosminexus RM 用のユーザ権限の設定および RD エリアの準備を行いません。

ユーザ権限の設定方法は、「2.1.7(1)(c) ユーザの設定およびスキーマの定義」に示す方法と同じです。

RD エリアの準備の方法は「2.1.7(1)(d) RD エリアの準備」に示す方法と同じです。

## 2.1.8 アンインストール

uCosminexus Service Architect をアンインストールする方法について説明します。

アンインストール作業は Administrator 権限が設定されたユーザが実施してください。

また、J2EE サーバ、Management Server、PRF など実行環境の構成要素を停止してからアンインストールしてください。組み込みデータベースを使用している場合は、組み込みデータベースを停止してからアンインストールしてください。

アンインストールの手順について、次に示します。

1. 次のどちらかの操作を実施します。
  - Windows の [ スタート ] メニューから、[ Cosminexus ] - [ uCosminexus Service Architect アンインストール ] を選択する
  - [ コントロールパネル ] の [ プログラムの追加と削除 ] から、[ uCosminexus Service Architect ] を選択する

uCosminexus Service Architect のアンインストールを確認するダイアログが表示されます。

2. [ はい ] ボタン、または [ いいえ ] ボタンをクリックします。

[ はい ] ボタンをクリックした場合

アンインストールが開始され、uCosminexus Service Architect の構成ソフトウェアがすべて削除されます。

[ いいえ ] ボタンをクリックした場合

アンインストールする構成ソフトウェアを選択するダイアログが表示されます。アンインストールする構成ソフトウェアを選択して、[ 次へ ] ボタンをクリックすると、アンインストールが開始され、選択した構成ソフトウェアが削除されます。

## 2.2 構成の形態と選択

Cosminexus サービスプラットフォームのどの機能を使用し、どのような運用をしたいかによって、データベースと Cosminexus RM の使用有無が異なります。使用する機能ごとに、データベースと Cosminexus RM の使用有無を次の表に示します。

表 2-1 使用したい機能ごとのデータベースと Cosminexus RM の使用有無

使用したい機能	データベース	Cosminexus RM
同期受付を使用したい	×	×
非同期受付を使用したい		
同期サービスアダプタを使用したい	×	×
非同期サービスアダプタを使用したい		
ビジネスプロセスを使用したい	永続化するビジネスプロセスを使用したい	
	永続化しないビジネスプロセスを使用したい <sup>1</sup>	×
プロセスインスタンスの実行履歴を管理したい		
電文の実行履歴を管理したい		×
DB アダプタを使用したい	×	×

(凡例)

：必須です。

×：必須ではありません。

注 1

永続化しないビジネスプロセスを使用する場合、開発環境で定義できるプロセスに制限がありますので注意してください。詳細は「6.2.1(3) ステータスの永続化の設定」を参照してください。

注 2

DB アダプタを使用する場合、HCSC サーバが稼働しているマシン上にデータベースは不要です。しかし、DB アダプタから接続するサービス部品稼働マシンでは、データベースが必要になります。

表 2-1 から、データベースと Cosminexus RM の使用有無による構成形態は、次の三つのパターンあることがわかります。

- データベースと Cosminexus RM の両方を使用する
- データベースと Cosminexus RM の両方を使用しない
- データベースを使用して Cosminexus RM を使用しない

本番環境をどのパターンで構築するかによって、テスト環境の設定内容が異なります。テスト環境と本番環境で設定する、データベースと Cosminexus RM の使用有無については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイ

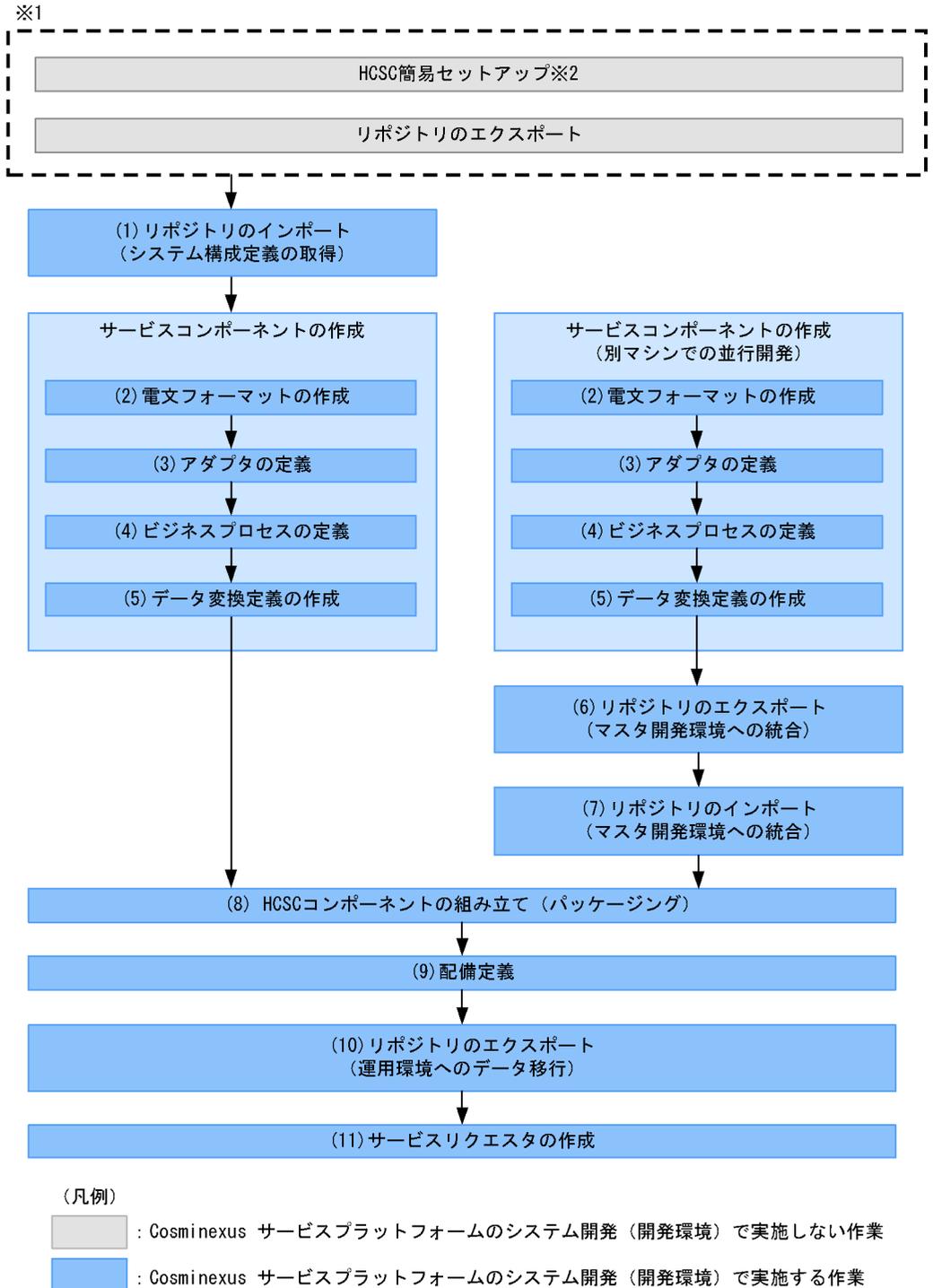
ド」を参照してください。また、テスト環境での設定内容については、「2.4 テスト環境の簡易セットアップ」を参照してください。

## 2.3 開発手順

---

Cosminexus サービスプラットフォームを利用してシステムを開発する手順を次に示します。

図 2-1 システムの開発手順



## 2. システムを開発する前に

### 注 1

マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。

### 注 2

HCSC 簡易セットアップ機能を使用してテスト環境を構築します。HCSC 簡易セットアップ機能については、「2.4 テスト環境の簡易セットアップ」を参照してください。HCSC 簡易セットアップ機能を使用しない場合は、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照して、運用・実行環境を構築してください。

各工程の作業を次に示します。

### (1) リポジトリのインポート（システム構成定義の取得）

運用環境で作成した HCSC サーバの情報をマスタ開発環境に統合するため、システム構成定義だけをインポートします。リポジトリのインポートについては、「3.2.3 リポジトリをインポートする」を参照してください。

### (2) 電文フォーマットの作成

サービスリクエスト、サービスアダプタ、およびサービス部品の間でやり取りされる電文の形式（電文フォーマット）を定義します。作成する電文フォーマットは、サービス部品の実行に利用する電文で XML 形式のデータを扱うか、バイナリ形式（XML 形式以外）のデータを扱うかによって異なります。電文フォーマットの作成の詳細については、「4. 電文フォーマットの作成」を参照してください。

### (3) アダプタの定義

アダプタを使用してシステムを開発する場合、アダプタ追加用のウィザードおよび定義済みのアダプタを利用して、アダプタを追加します。追加したアダプタは、アダプタ定義画面で定義します。アダプタの定義の詳細については、「5. アダプタの定義」を参照してください。

### (4) ビジネスプロセスの定義

ビジネスプロセスを使用してシステムを開発する場合、ビジネスプロセス追加用のウィザードおよび定義済みのビジネスプロセスを利用して、ビジネスプロセスを追加します。追加したビジネスプロセスは、ビジネスプロセス定義画面で定義します。ビジネスプロセスの定義の詳細については、「6. ビジネスプロセスの定義」を参照してください。

また、ビジネスプロセスの実行要求を標準受付以外で受け付ける場合、ユーザ定義受付を定義する必要があります。ユーザ定義受付の定義の詳細については、「8. ユーザ定義受付の定義」を参照してください。

### (5) データ変換定義の作成

サービス部品との電文のやり取りでデータ変換が必要な場合は、データ変換を定義します。データ変換の定義の詳細については、「7. データ変換の定義」を参照してください。

### (6) リポジトリのエクスポート（マスタ開発環境への統合）

複数の開発環境で HCSC コンポーネントを開発（分散開発）する場合、マスタ開発環境以外の環境で開発した HCSC コンポーネントが含まれるリポジトリをエクスポートします。

リポジトリのエクスポートについては、「3.2.2 リポジトリをエクスポートする」を参照してください。

### (7) リポジトリのインポート（マスタ開発環境への統合）

マスタ開発環境以外のマシンで分散開発した HCSC コンポーネントの情報をマスタ開発環境に統合するため、(6) でエクスポートしたリポジトリの一部をマスタ開発環境にインポートします。リポジトリのインポートについては、「3.2.3 リポジトリをインポートする」を参照してください。

### (8) HCSC コンポーネントの組み立て（パッケージング）

(3) ~ (5) で定義した HCSC コンポーネントの定義に関連する定義ファイル、および実行環境が提供するファイルをまとめて EAR ファイルにパッケージングします。パッケージングの方法については、「9. HCSC コンポーネントのパッケージングと配備定義」を参照してください。

### (9) 配備定義

HCSC コンポーネントを配備するクラスタ（または単一の HCSC サーバ）の情報を定義します。配備定義の詳細については、「9. HCSC コンポーネントのパッケージングと配備定義」を参照してください。

#### 参考

配備する前に、配備するクラスタ（または単一の HCSC サーバ）に設定されている、データベースと Cosminexus RM の使用有無を確認する場合の手順を次に示します。

1. Eclipse のメニューから、[ HCSC-Definer ] - [ システム構成定義一覧 ] を選択します。  
ツリービューのシステム構成定義一覧に、配備するクラスタ（または単一の HCSC サーバ）の一覧が表示されます。
2. 配備するクラスタ（または単一の HCSC サーバ）の一覧から、確認したいクラスタ（または単一の HCSC サーバ）を選択し、ダブルクリックします。  
配備するクラスタ（または単一の HCSC サーバ）に設定されている、データベースと Cosminexus RM の使用有無の情報が表示されます。

## 2. システムを開発する前に

### (10)リポジトリのエクスポート(運用環境へのデータ移行)

マスタ開発環境で、HCSC コンポーネントの配備に必要な定義ファイルを格納したリポジトリをエクスポートします。マスタ開発環境からエクスポートした情報は、運用環境にインポートして利用されます。リポジトリのエクスポートについては、「3.2.2 リポジトリをエクスポートする」を参照してください。

### (11)サービスリクエストの作成

HCSC コンポーネントにサービス部品の実行を要求する電文を送信するためのサービスリクエストを作成します。なお、サービスリクエストの作成は、アダプタの定義やビジネスプロセスの定義と同時に行うこともできます。同時に行う場合、アダプタの定義やビジネスプロセスの定義で設定するインターフェース情報などをあらかじめ確認して、サービスリクエストでも同じ設定にしてください。HCSC コンポーネントの作成方法については、「10. サービスリクエストの作成」を参照してください。

## 2.4 テスト環境の簡易セットアップ

---

実行環境に配備して実運用を始める前に、uCosminexus Service Architect で開発したアダプタやビジネスプロセスは、テストやデバッグを実施することを推奨します。そのためには、テストやデバッグを実施するための運用・実行環境（テスト環境）を構築する必要があります。

uCosminexus Service Architect には、テスト環境の構築を支援する HCSC 簡易セットアップ機能が備わっています。テストやデバッグを実施するための運用・実行環境には、さまざまな情報を設定する必要があるため、構築には相応の時間が掛かります。HCSC 簡易セットアップ機能を利用すると、テスト環境に必要なさまざまな情報を自動的に設定できます。また、HCSC 簡易セットアップ機能には、HCSC 簡易セットアップ機能を利用してセットアップしたテスト環境を自動的にアンセットアップする機能もあります。

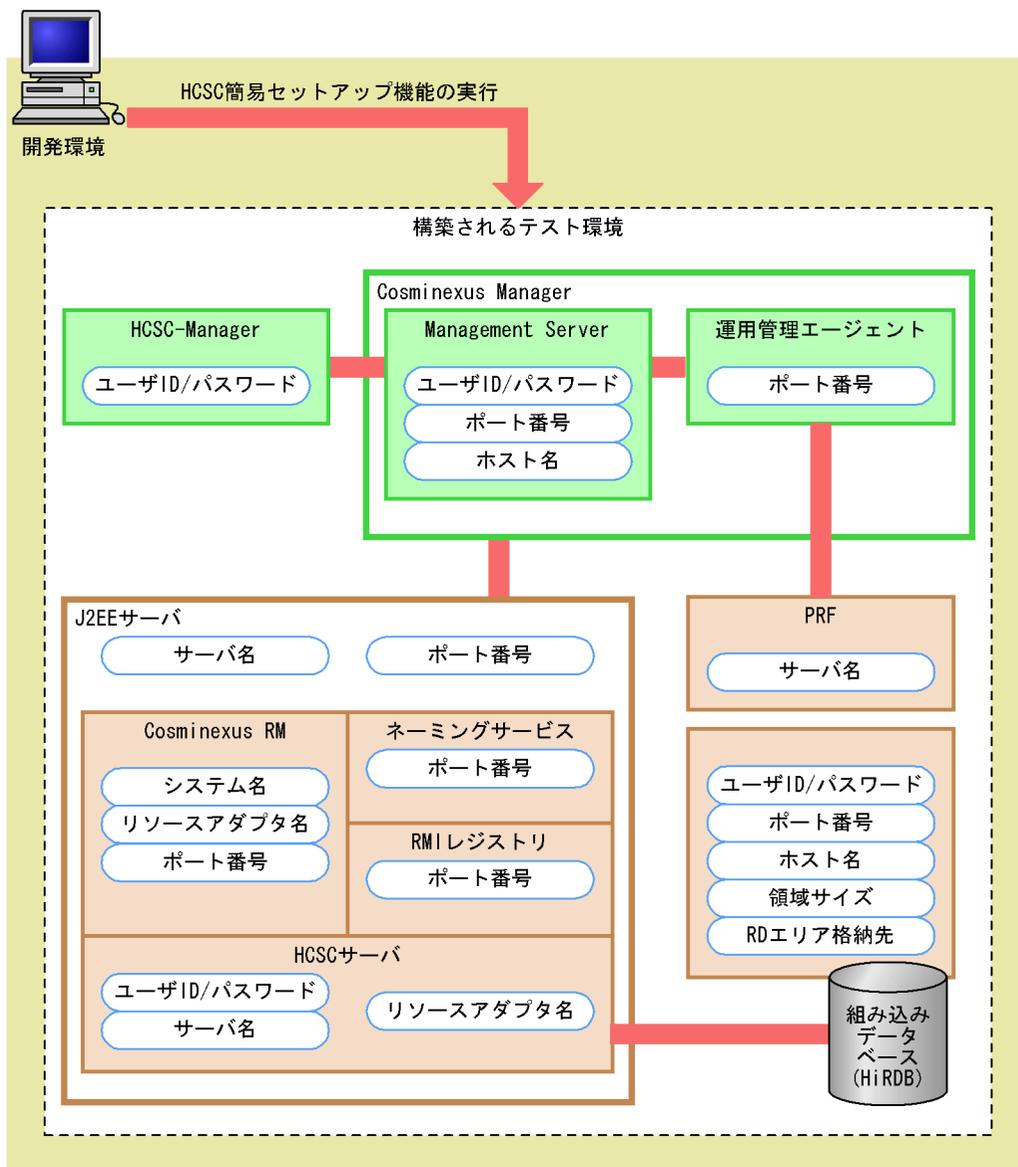
この節では、HCSC 簡易セットアップ機能の概要と利用方法について説明します。

### 2.4.1 HCSC 簡易セットアップ機能で構築できる環境

HCSC 簡易セットアップ機能を利用して構築されるテスト環境は、次の図のような構成になります。次の図に示すプロセス以外のものをテスト環境として利用する場合、別途設定が必要になります。

2. システムを開発する前に

図 2-2 HCSC 簡易セットアップ機能を利用して構築されるテスト環境の構成



(凡例)

 : HCSC簡易セットアップ機能で設定される情報

HCSC 簡易セットアップ機能を利用すると、図 2-2 に示すテスト環境の各プロセスに、ユーザ ID、パスワード、ポート番号、ホスト名などの情報を自動的に設定できます。HCSC 簡易セットアップ機能を利用した場合にテスト環境に設定される情報の詳細については、「2.4.3(2) テスト環境の運用に必要な情報」を参照してください。

## 2.4.2 HCSC 簡易セットアップ機能の実行

HCSC 簡易セットアップ機能を利用してテスト環境をセットアップ、またはアンセットアップするための前提条件と、テスト環境のセットアップおよびアンセットアップの方法について説明します。

### (1) テスト環境のセットアップ

HCSC 簡易セットアップ機能を利用してテスト環境をセットアップする場合の前提条件、およびセットアップ方法について説明します。

#### (a) 前提条件

テスト環境をセットアップするための前提条件を次に示します。

#### テスト環境の前提ソフトウェア

HCSC 簡易セットアップ機能を利用してテスト環境をセットアップするには、次のソフトウェアがインストールされている必要があります。

- Cosminexus Component Container <sup>1</sup>
- Cosminexus XML Processor <sup>1</sup>
- Cosminexus TPBroker <sup>1</sup>
- Cosminexus Performance Tracer <sup>1</sup>
- Cosminexus Reliable Messaging <sup>1</sup>
- Cosminexus Service Coordinator <sup>1</sup>
- WSDL4J 1.5.1 <sup>2</sup>

#### 注 1

これらのソフトウェアは、uCosminexus Service Architect の構成ソフトウェアです。uCosminexus Service Architect をインストールしている環境であれば、改めて上記のソフトウェアをインストールする必要はありません。

上記のソフトウェアを改めてインストールする場合、日立総合インストーラからインストールします。日立総合インストーラの指示に従ってインストールしてください。なお、インストール作業は Administrator 権限が設定されたユーザが実施してください。

#### 注 2

WSDL4J 1.5.1 は、これ以外の前提ソフトウェアをインストールしたあとにインストールしてください。

#### HCSC 簡易セットアップ機能を利用するときのシステムの状態

HCSC 簡易セットアップ機能を利用してテスト環境をセットアップする場合、システムが次の状態である必要があります。

- Cosminexus を新規にインストールした直後であること。
- Cosminexus に関する設定をしていないこと。

## 2. システムを開発する前に

ただし、システム環境変数で、変数名「TZ」の値が有効になっている必要があります。環境変数「TZ」にはタイムゾーンを設定します。タイムゾーンには「JST-9」と指定してください。

### 注

テスト環境をアンセットアップしたあとは、HCSC 簡易セットアップ機能を利用してテスト環境を再度セットアップできます。テスト環境のアンセットアップについては「2.4.2(2) テスト環境のアンセットアップ」を参照してください。

### (b) セットアップ方法

HCSC 簡易セットアップ機能を利用したテスト環境のセットアップには、HCSC 簡易セットアップ画面を利用します。テスト環境のセットアップ方法を次に示します。

1. コマンドプロンプトで次のコマンドを実行し、HCSC 簡易セットアップ画面を表示させます。

```
<Cosminexusインストールディレクトリ>%CSC%bin%csceasysetup.bat
```

2. HCSC 簡易セットアップ画面で必要な情報を入力します。

HCSC 簡易セットアップ画面で入力する項目については、「2.4.2(1)(c) HCSC 簡易セットアップ画面の入力項目」を参照してください。

3. [セットアップ] ボタンをクリックします。

テスト環境のセットアップが開始されます。セットアップの状況は、HCSC 簡易セットアップ画面のコンソールに表示されます。コンソールに「HCSC 簡易セットアップ機能のセットアップを終了します」と表示されたら、テスト環境は正常にセットアップされています。

### ! 注意事項

コンソールにエラーが表示され、テスト環境のセットアップが異常終了した場合、再セットアップする必要があります。異常終了したときに、HCSC 簡易セットアップ画面の [セットアップ] ボタンが活性か、非活性かによって、再セットアップの手順が異なります。

[セットアップ] ボタンが活性の場合

[セットアップ] ボタンをクリックして、再セットアップできます。

[セットアップ] ボタンが非活性の場合

[アンセットアップ] ボタンをクリックして、一度アンセットアップします。そのあとで、再度セットアップしてください。

### (c) HCSC 簡易セットアップ画面の入力項目

HCSC 簡易セットアップ画面の入力項目について、HCSC 簡易セットアップ機能で構築される環境内の各プロセスとの関連を用いて説明します。

HCSC 簡易セットアップ画面を次の図に示します。

図 2-3 HCSC 簡易セットアップ画面

HCSC 簡易セットアップ

操作 その他

標準モデル  ESBモデル

**組み込みデータベース**

データ格納先(660MB以上必要)

DB接続ポート番号  
 <5001~65535>

**Management Server**

HCSCサーバ運用ポート番号  
 <1~65535>

論理サーバ運用ポート番号  
 <1~65535>

終了要求受信ポート番号(内部管理用)  
 <1~65535>

内部通信ポート番号(内部管理用)  
 <1~65535>

**運用管理エージェント**

エージェント接続ポート番号(内部管理用)  
 <1~65535>

**HCSCサーバ**

Webサービス/MDB(WS-R)受付ポート番号  
 <1~65535>

SessionBean受付ポート番号  
 <1~65535>

MDB(DBキュー)受付ポート番号  
 <1024~65535>

稼働確認ポート番号  
 <1~65535>

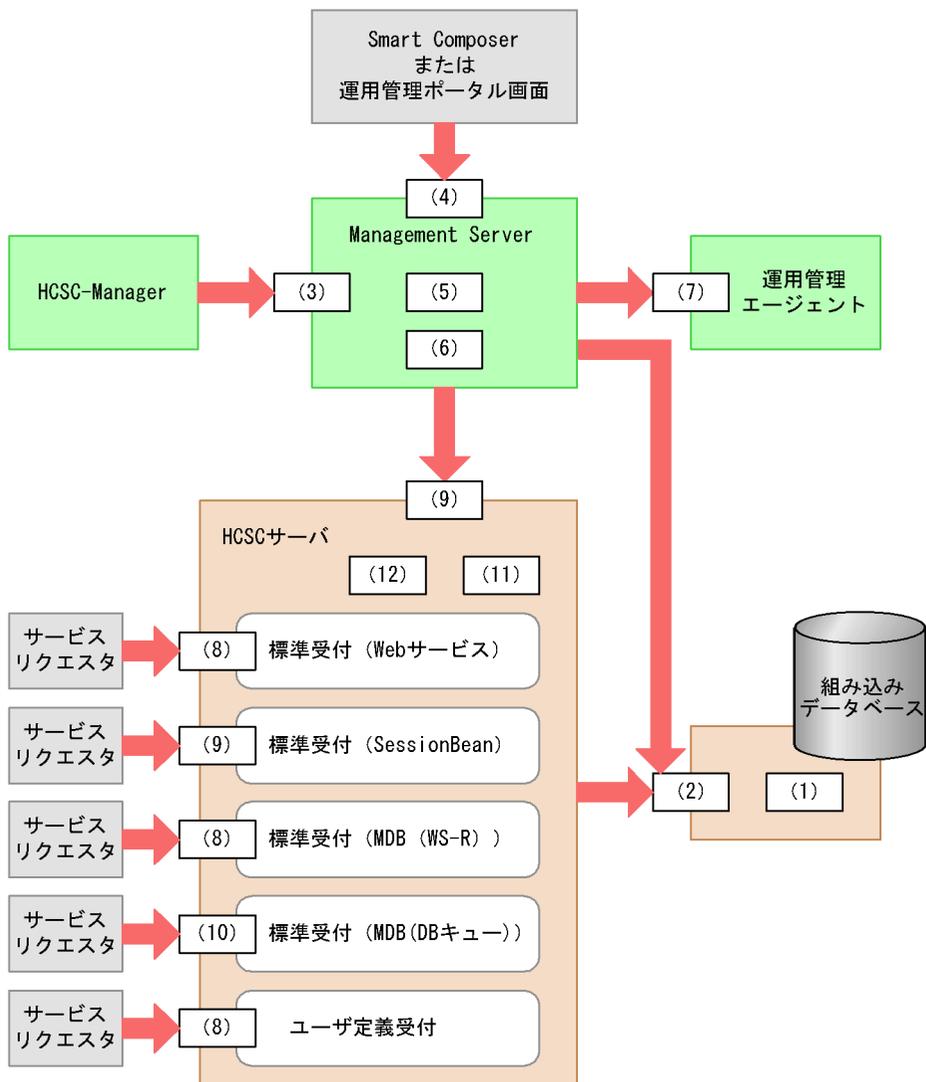
簡易Webサーバポート番号(内部管理用)  
 <1~65535>

コンソール

HCSC 簡易セットアップ画面で入力する項目は、次の図のように、構築される環境の各プロセスに設定されます。設定される値の関連を次の図に示します。

2. システムを開発する前に

図 2-4 HCSC 簡易セットアップ画面の入力項目の関連



(凡例)

(n) : HCSC簡易セットアップ画面で入力する情報 ((n)は、表2-6の項番)

➡ : 通信の方向

[標準モデル] または [ESB モデル] のどちらを選択したかによって、設定する項目が異なります。図 2-4 内で (n) で示されている設定値 (HCSC 簡易セットアップ画面での入力項目) の詳細を次の表に示します。

表 2-2 HCSC 簡易セットアップ画面で入力する項目

入力項目			説明	初期値 <sup>2</sup>
分類	項番	HCSC 簡易 セットアップ 画面の項目名 <sup>1</sup>		
モデル	-	標準モデル	データベースと Cosminexus RM の両方を使用する場合に選択します。	選択
	-	ESB モデル	データベースと Cosminexus RM の両方を使用しない場合に選択します。	-
組み込み データベース	(1)	データ格納先	RD エリアを作成するディレクトリおよびシステムファイルを作成するディレクトリを指定します。660MB 以上使用できるディレクトリを指定します。 存在しないディレクトリを指定した場合、指定したディレクトリが新規作成されます。 [標準モデル] を選択した場合だけ指定します。	<Cosminexus インストール ディレクトリ >¥CSC¥DB¥ar ea
	(2)	DB 接続ポート 番号	Management Server または HCSC サーバから組み込みデータベースにアクセスする場合に使用するポート番号を指定します。5001 ~ 65535 の整数で指定します。 [標準モデル] を選択した場合だけ指定します。	22200
Management Server	(3)	HCSC サーバ 運用ポート番号	HCSC-Manager から Management Server にアクセスする場合に使用するポート番号を指定します。1 ~ 65535 の整数で指定します。	28099
	(4)	論理サーバ運 用ポート番号	Smart Composer または運用管理ポータル画面から Management Server にアクセスする場合に使用するポート番号を指定します。1 ~ 65535 の整数で指定します。	28080
	(5)	終了要求受信 ポート番号 (内部管理用)	Management Server が内部の管理用に利用するポート番号を指定します。1 ~ 65535 の整数で指定します。	28005
	(6)	内部通信ポ ート番号 (内 部管理用)	Management Server が内部の管理用に利用するポート番号を指定します。1 ~ 65535 の整数で指定します。	28009
運用管理 エージェント	(7)	エージェント 接続ポート番 号 (内部管理 用)	Management Server から運用管理エージェントにアクセスする場合に使用する内部管理用のポート番号を指定します。1 ~ 65535 の整数で指定します。	20295

## 2. システムを開発する前に

入力項目			説明	初期値 <sup>2</sup>
分類	項番	HCSC 簡易 セットアップ 画面の項目名 <sup>1</sup>		
HCSC サーバ	(8)	Web サービス / MDB (WS-R) 受付ポート番号	サービスリクエストから標準受付 (Web サービスまたはMDB (WS-R)), またはユーザ定義受付にアクセスする場合に使用するポート番号を指定します。1 ~ 65535 の整数で指定します。	80
	(9)	SessionBean 受付ポート番号	Management Server から HCSC サーバに, またはサービスリクエストから標準受付 (SessionBean) にアクセスする場合に使用するポート番号を指定します。1 ~ 65535 の整数で指定します。 [標準モデル] を選択した場合だけ指定します。	900
	(10)	MDB (DB キュー) 受付ポート番号	サービスリクエストから標準受付 (MDB (DB キュー)) にアクセスする場合に使用するポート番号を指定します。1024 ~ 65535 の整数で指定します。	20351
	(11)	稼働確認ポート番号	HCSC サーバが内部の管理用に利用するポート番号を指定します。1 ~ 65535 の整数で指定します。	23152
	(12)	簡易 Web サーバポート番号 (内部管理用)	HCSC サーバが内部の管理用に利用するポート番号を指定します。1 ~ 65535 の整数で指定します。	8080

(凡例) - : 該当しません。

### 注 1

HCSC 簡易セットアップ画面の対応する項目の上でマウスカーソルを一時停止すると, その項目の説明文が一時的に表示されます。

### 注 2

初期値は, 初めて HCSC 簡易セットアップ画面を表示したときに初期表示される値です。初期値を変更してセットアップした場合, 次回の HCSC 簡易セットアップ画面には前回の設定値が初期値として表示されます。

## (2) テスト環境のアンセットアップ

HCSC 簡易セットアップ機能を利用してセットアップしたテスト環境をアンセットアップする場合の前提条件, およびアンセットアップ方法について説明します。

**!** 注意事項

HCSC 簡易セットアップ機能を利用してアンセットアップすると、テスト環境はセットアップ前の状態に戻ります。テスト環境で利用していたサービスアダプタやビジネスプロセスなどは自動的に削除されるので、必要なデータは、アンセットアップ前に必ずバックアップを取得してください。

## (a) 前提条件

テスト環境をアンセットアップするための前提条件を次に示します。

- アンセットアップの対象が、HCSC 簡易セットアップ機能を利用してセットアップしたテスト環境であること。
- Management Server、および運用管理エージェントが起動していること。
- HCSC 簡易セットアップ画面で [ 標準モデル ] を選択してテスト環境をセットアップした場合、組み込みデータベースのサービス部品が起動していること。

## (b) アンセットアップ方法

テスト環境のアンセットアップには、HCSC 簡易セットアップ画面を利用します。テスト環境のアンセットアップ方法を次に示します。

1. コマンドプロンプトで次のコマンドを実行し、HCSC 簡易セットアップ画面を表示させます。

```
<Cosminexusインストールディレクトリ>%CSC%\bin\csceasysetup.bat
```

2. HCSC 簡易セットアップ画面の [ アンセットアップ ] ボタンをクリックします。  
テスト環境のアンセットアップが開始されます。アンセットアップの状況は、HCSC 簡易セットアップ画面のコンソールに表示されます。コンソールに「HCSC 簡易セットアップ機能のアンセットアップを終了します」と表示されたら、テスト環境は正常にアンセットアップされています。

### 2.4.3 HCSC 簡易セットアップ機能でセットアップしたテスト環境の運用

HCSC 簡易セットアップ機能を利用してセットアップしたテスト環境の運用方法、および運用に必要な情報について説明します。

## (1) テスト環境の起動と停止

テスト環境の起動および停止方法について、次の表に示します。

## 2. システムを開発する前に

表 2-3 テスト環境の起動および停止方法

起動・停止の対象	起動・停止方法
組み込みデータベース	<p>コマンドプロンプトで次のコマンドを実行します。</p> <p>起動する場合：            &lt;Cosminexusのインストールディレクトリ            &gt;%CSC%¥DB¥bats¥cddbstart.bat</p> <p>停止する場合：            &lt;Cosminexusのインストールディレクトリ            &gt;%CSC%¥DB¥bats¥cddbstop.bat</p> <p>組み込みデータベースの起動・停止方法および運用方法の詳細は、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。</p>
組み込みデータベースのサービス	<p>テスト環境のマシンを起動または停止すると、自動的に起動または停止されます。</p>
<ul style="list-style-type: none"> <li>• Management Server</li> <li>• 運用管理エージェント</li> </ul>	<p>テスト環境のマシンを起動すると、自動的に起動されます。また、テスト環境のマシンを停止すると、自動的に停止されます。</p>
<ul style="list-style-type: none"> <li>• Cosminexus Performance Tracer</li> <li>• J2EE サーバ</li> <li>• HCSC サーバ</li> </ul>	<p>コマンドプロンプトで次のコマンドを実行します。</p> <p>起動する場合：            &lt;Cosminexusのインストールディレクトリ            &gt;%CSC%¥bin¥cscsvstart.bat -system</p> <p>停止する場合：            &lt;Cosminexusインストールディレクトリ            &gt;%CSC%¥bin¥cscsvstop.bat -system</p> <p>Cosminexus Performance Tracer、および J2EE サーバの起動・停止方法および運用方法の詳細は、マニュアル「Cosminexus システム運用ガイド」を参照してください。HCSC サーバの起動・停止方法および運用方法の詳細は、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。</p>

### 注

HCSC 簡易セットアップ画面で [標準モデル] を選択した場合だけ実施します。

## (2) テスト環境の運用に必要な情報

HCSC 簡易セットアップ機能を利用してテスト環境をセットアップした場合、必要な情報は自動的に設定されます。テスト環境を使ってテストおよびデバッグを行うには、HCSC 簡易セットアップ機能でテスト環境をセットアップする際に設定される情報を知っておく必要があります。

HCSC 簡易セットアップ機能を利用してセットアップしたテスト環境に設定される情報を、次に説明します。

## ポイント

HCSC 簡易セットアップ機能を利用してセットアップしたテスト環境に設定される情報は、テスト環境のセットアップ後にカスタマイズできます。カスタマイズの方法については、「2.4.4 テスト環境のカスタマイズ」を参照してください。

## (a) ユーザ ID およびパスワードの情報

HCSC 簡易セットアップ機能でセットアップしたテスト環境に設定されるユーザ ID およびパスワードの情報を次の表に示します。

表 2-4 テスト環境に設定されるユーザ ID およびパスワード

設定先	設定されるユーザ ID またはパスワード	初期値	説明
組み込みデータベース <sup>1</sup>	ユーザ ID	admin	テーブル所有者の認識別子（環境変数の USRID）。
	パスワード	admin	テーブル所有者のパスワード（環境変数の PSWD）。
Management Server	管理ユーザ ID	admin	Management Server にログインするための管理ユーザ ID（ <code>cmx_build_system -change</code> コマンド <sup>2</sup> などで使用）。
	パスワード	admin	Management Server にログインするためのパスワード（ <code>cmx_build_system -change</code> コマンド <sup>2</sup> などで使用）。
HCSC-Manager	HCSC-Manager ログインユーザ ID	admin	HCSC-Manager にログインするためのユーザ ID（ <code>cscsvstart</code> コマンド <sup>3</sup> などで使用）。
	HCSC-Manager ログインパスワード	admin	HCSC-Manager にログインするためのパスワード（ <code>cscsvstart</code> コマンド <sup>3</sup> などで使用）。
HCSC サーバ	HCSC サーバが使用するデータベースのユーザ ID	admin	HCSC サーバが使用するデータベースのユーザ ID（ <code>csmlslocation</code> コマンド <sup>3</sup> などで使用）。
	HCSC サーバが使用するデータベースのパスワード	admin	HCSC サーバが使用するデータベースのパスワード（ <code>csmlslocation</code> コマンド <sup>3</sup> などで使用）。

注 1

HCSC 簡易セットアップ画面で [標準モデル] を選択した場合だけ設定されます。

注 2

このコマンドについては、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

## 2. システムを開発する前に

### 注 3

これらのコマンドについては、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。

### (b) ポート番号の情報

HCSC 簡易セットアップ機能でセットアップしたテスト環境には、次に示すポート番号が設定されます。ポート番号は、HCSC 簡易セットアップ画面で任意の値を指定できます。HCSC 簡易セットアップ画面での指定値の詳細については、「表 2-2 HCSC 簡易セットアップ画面で入力する項目」を参照してください。

組み込みデータベース（HCSC 簡易セットアップ画面で [標準モデル] を選択した場合だけ）

- DB 接続ポート番号

#### Management Server

- HCSC サーバ運用ポート番号
- 論理サーバ運用ポート番号
- 終了要求受信ポート番号
- 内部通信ポート番号

#### 運用管理エージェント

- エージェント接続ポート番号

#### J2EE サーバ

- Web サービス /MDB (WS-R) 受付ポート番号
- SessionBean 受付ポート番号
- MDB (DB キュー) 受付ポート番号
- 稼働確認ポート番号
- 簡易 Web サーバポート番号

### (c) 名称の情報

HCSC 簡易セットアップ機能でセットアップしたテスト環境に設定されるホスト名やサーバ名などの情報を次の表に示します。

表 2-5 テスト環境に設定される名称

設定先	設定される名称	初期値	説明
組み込みデータベース <sup>1</sup>	ホスト名	localhost	組み込みデータベースのホスト名（環境変数の DB_HOST）。
Management Server	ホスト名	localhost	Management Server のホスト名（cmx_build_system -change コマンド <sup>2</sup> などで使用）。

設定先	設定される名称	初期値	説明
HCSC-Manager	Cosminexus Manager 名	MyMNG	HCSC-Manager が Cosminexus Manager を認識するための HCSC-Manager 独自の識別名 ( cscsvstart コマンド <sup>4</sup> などで使用 )。
PRF	サーバ名	MyPRF	PRF のサーバ名称。
J2EE サーバ	サーバ名	MyServer	接続先 J2EE サーバの名称 ( cjstartsv コマンド <sup>3</sup> などで使用 )。
	Cosminexus RM が連携するシステム全体で一意的なシステム名	RM	Cosminexus RM が連携するシステム全体で一意的なシステム名 ( 環境変数の HRM_SYSTEM_NAME )。
	リソースアダプタ名	Cosminexus Reliable Messaging	
DB_Connector_for_HiRDB_Type4			
DB_Connector_for_HiRDB_Type4_Cosminexus_RM			
クラスタ	クラスタ名	MyUnit	クラスタの名称
HCSC サーバ	HCSC サーバ名	MyCSC	HCSC サーバの名称 ( cscsvstart コマンド <sup>4</sup> などで使用 )。

## 注 1

HCSC 簡易セットアップ画面で [ 標準モデル ] を選択した場合だけ設定されます。

## 注 2

このコマンドについては、マニュアル「Cosminexus 簡易構築・運用ガイド」を参照してください。

## 注 3

これらのコマンドについては、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

## 注 4

このコマンドについては、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。

## (d) そのほかの情報

HCSC 簡易セットアップ画面で [ 標準モデル ] を選択した場合、HCSC 簡易セットアップ機能でセットアップしたテスト環境の組み込みデータベースには、次の情報が設定されます。

## 2. システムを開発する前に

### RD エリア格納先

RD エリア格納先の情報は、HCSC 簡易セットアップ画面で任意の値を指定できます。HCSC 簡易セットアップ画面での指定値の詳細については、「表 2-2 HCSC 簡易セットアップ画面で入力する項目」を参照してください。

### 構築するデータベースの領域サイズ

構築するデータベースの領域サイズ（環境変数の DB\_SIZE）が設定されます。初期値は「660MB」です。

## （3）障害対策

HCSC 簡易セットアップ機能を利用して、HCSC サーバのセットアップやアンセットアップ、HCSC コンポーネントの配備や削除、およびリクエストを送信し続けると、組み込みデータベース内で未使用のセグメントが不足するおそれがあります。

未使用のセグメントが不足した場合、次の対策を実施してください。

方法 1：組み込みデータベース内の空きセグメントを解放する

1. 組み込みデータベース内の表を SQL Executer で検索します。  
表の検索方法については、HiRDB SQL Executer のオンラインヘルプの pdsql コマンドを参照してください。
2. 手順 1. で検索した結果から、「CSC」および「RM」で始まるすべての表に対して、pdreclaim コマンドを発行し、空きページを解放して再利用します。  
pdreclaim コマンドの詳細については、マニュアル「スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス」を参照してください。

方法 2：組み込みデータベース内の表を再編成する

1. 組み込みデータベース内の表を SQL Executer で検索します。  
表の検索方法については、マニュアル「HiRDB SQL Executer オンラインヘルプ」の pdsql コマンドを参照してください。
2. 手順 1. で検索した結果から、「CSC」および「RM」で始まるすべての表に対して pdrorg コマンドを発行し、表を再編成します。  
表を再編成する方法については、マニュアル「スケーラブルデータベースサーバ HiRDB Version 8 システム運用ガイド」の表の再編成に関する内容を参照してください。  
pdrorg コマンドの詳細については、マニュアル「スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス」を参照してください。

方法 3：HCSC 簡易セットアップ機能で構築したテスト環境を再セットアップする

HCSC 簡易セットアップ機能で構築したテスト環境をいったんアンセットアップし、再度セットアップします。この作業を行うと、組み込みデータベースが再生成されます。

ただし、再セットアップすると、作成したサービスアダプタやビジネスプロセスなども削除されます。そのため、あらかじめリポジトリをエクスポートしてから、再

セットアップし、再セットアップ後にリポジトリをインポートしてください。

## 2.4.4 テスト環境のカスタマイズ

HCSC 簡易セットアップ機能を利用してセットアップしたテスト環境には、「2.4.3(2) テスト環境の運用に必要な情報」に示す情報が設定されています。これらの情報は、テスト環境をセットアップしたあとにカスタマイズできます。

テスト環境をカスタマイズするには、HCSC 簡易セットアップ機能でのテスト環境のセットアップ時に定義ファイルに設定された値を編集します。

編集できる定義ファイルを次の表に示します。各定義ファイルの詳細については、表の「参照先」に示すマニュアルを参照してください。

表 2-6 編集できる定義ファイル

ファイル名	説明	格納先	参照先
adminagent.properties (運用管理エージェントプロパティファイル)	運用管理エージェントとの通信で使用するポート番号が設定されているプロパティファイルです。	A	a
cdsetupconfig.bat (組み込みデータベース設定バッチファイル)	組み込みデータベースの環境変数を設定するバッチファイルです。	B	b
cmdconf.bat (HCSC-Messaging コマンド共通定義ファイル)	uCosminexus Service Platform の csmXXX コマンドで組み込みデータベースにアクセスするために、HiRDB type4 JDBC Driver のクラスパスを設定するバッチファイルです。	C	c
Cosminexus_Reliable_Messaging.xml (Connector 属性ファイル)	リソースアダプタの属性 (コンフィグレーションプロパティの値やプロパティ値) が設定されている XML ファイルです。	D	a
csccmd.properties (HCSC-Manager コマンド共通定義ファイル)	運用環境で使用するコマンドの省略値 (ログインユーザ ID やログインパスワード) が設定されているプロパティファイルです。	E	c
cscmng.properties (HCSC-Manager 定義ファイル)	HCSC-Manager の動作に必要な情報が設定されているプロパティファイルです。	E	c
cscsvconfig.properties (HCSC サーバランタイム定義ファイル)	HCSC サーバの起動時に必要なランタイム情報 (実行履歴管理情報、DB 情報) が設定されているプロパティファイルです。	D	c
cscsvsetup.properties (HCSC サーバセットアップ定義ファイル (標準モデル))	標準モデルで使用する、HCSC サーバのセットアップに必要な情報 (J2EE サーバ、Cosminexus RM、データベースに関連する情報) が設定されているプロパティファイルです。	D	c

## 2. システムを開発する前に

ファイル名	説明	格納先	参照先
cscsvsetup.properties.esb (HCSC サーバセットアップ定義ファイル (ESB モデル))	ESB モデルで使用する, HCSC サーバのセットアップに必要な情報 (J2EE サーバ, Cosminexus RM, データベースに関連する情報) が設定されているプロパティファイルです。	D	c
cscsvsetup.xml (HCSC サーバ構成定義ファイル (標準モデル))	標準モデルで使用する, HCSC サーバの構成情報 (cosminexus-manager, jms-physical-reception, ejb-reception の定義) が設定されている XML ファイルです。	D	c
cscsvsetup.xml.esb (HCSC サーバ構成定義ファイル (ESB モデル))	ESB モデルで使用する, HCSC サーバの構成情報 (cosminexus-manager, jms-physical-reception, ejb-reception の定義) が設定されている XML ファイルです。	D	c
DB_Connector_for_HiRDB_Type4.xml (Connector 属性ファイル)	リソースアダプタの属性 (コンフィグレーションプロパティの値, プロパティ値) が設定されている XML ファイルです。	D	a
DB_Connector_for_HiRDB_Type4_Cosminexus_RM.xml (Connector 属性ファイル)	リソースアダプタの属性 (コンフィグレーションプロパティの値, プロパティ値) が設定されている XML ファイルです。	D	a
grantuser (ユーザ定義ファイル)	HiRDB のユーザの作成, およびスキーマの作成を行うファイルです。	B	b
hrmqueue-transmit.xml (アプリケーション属性ファイル)	アプリケーションの属性が設定されている XML ファイルです。	D	a
input.properties (システム共通情報記載プロパティファイル)	HCSC 簡易セットアップ画面で入力した情報, およびシステム共通の情報設定されているプロパティファイルです。	D	-
inserttableshirdb.sql (テーブルインサート SQL ファイル)	システム管理情報テーブルにレコードをインサートする SQL ファイルです。	D	-
mserver.cfg (Management Server 用オプション定義ファイル)	Management Server を実行する JavaVM の起動オプションが設定されているファイルです。	A	a
mserver.properties (Management Server 環境設定ファイル)	Management Server が使用するポート番号や, 障害検知時コマンドの動作が設定されているプロパティファイルです。	A	a
qconf.txt (キュー定義ファイル)	キュー情報 (DisplayName, QueueName) が設定されているテキストファイルです。	E	d
setupscript (セットアップスクリプトファイル (標準モデル))	標準モデルで使用する, セットアップ処理のタスクの実行順が設定されているスクリプトファイルです。	D	-
setupscript.esb (セットアップスクリプトファイル (ESB モデル))	ESB モデルで使用する, セットアップ処理のタスクの実行順が設定されているスクリプトファイルです。	D	-

ファイル名	説明	格納先	参照先
simple_model.xml (HCSC 簡易セットアップ定義ファイル)	Smart Composer 機能のコマンドで構築する Web システムが定義されている XML ファイルです。	D	e
tablecreate (テーブル定義ファイル)	HiRDB のスキーマ、およびテーブルの作成を行うファイルです。	B	b
unsetupscript (アンセットアップスクリプトファイル (標準モデル))	標準モデルで使用する、アンセットアップ処理のタスクの実行順が設定されているスクリプトファイルです。	D	-
unsetupscript.esb (アンセットアップスクリプトファイル (ESB モデル))	ESB モデルで使用する、アンセットアップ処理のタスクの実行順が設定されているスクリプトファイルです。	D	-

## ( 凡例 )

- A : <Cosminexus インストールディレクトリ >¥manager¥config  
 B : <Cosminexus インストールディレクトリ >¥CSC¥DB¥bats  
 C : <Cosminexus インストールディレクトリ >¥CSC¥config¥msg  
 D : <Cosminexus インストールディレクトリ >¥CSC¥system¥manager¥setup  
 E : <Cosminexus インストールディレクトリ >¥CSC¥config¥manager  
 - : 参照するマニュアルはありません。  
 a : Cosminexus リファレンス 定義編  
 b : Cosminexus アプリケーション開発ガイド  
 c : Cosminexus サービスプラットフォーム システム構築・運用ガイド  
 d : Cosminexus Reliable Messaging  
 e : Cosminexus 簡易構築・運用ガイド

## 注

セットアップ構成関連のプロパティ ( db-use プロパティ , rm-use プロパティ , および hcscserver-data-filepath プロパティ ) は変更できません。

## 参考

テスト環境をカスタマイズする場合、次のディレクトリ内のファイルをカスタマイズの参考として利用できます。

## 2. システムを開発する前に

```
<Cosminexusインストールディレクトリ>%CSC%\log\manager\setup
```

このディレクトリに格納されているファイルを次に示します。

表 2-6 に示す定義ファイルと同じ名称のファイル

HCSC 簡易セットアップ機能を実行したときに設定された定義ファイルと同じ内容のファイル (HCSC 簡易セットアップ機能のログファイル) です。

adminagent.properties.bak ( adminagent.properties のバックアップファイル)

現在の adminagent.properties の一つ前の adminagent.properties の設定が保存されています。

cmdconf.bat.bak ( cmdconf.bat のバックアップファイル)

現在の cmdconf.bat の一つ前の cmdconf.bat の設定が保存されています。

csccmd.properties.bak ( csccmd.properties のバックアップファイル)

現在の csccmd.properties の一つ前の csccmd.properties の設定が保存されています。

cscmng.properties.bak ( cscmng.properties のバックアップファイル)

現在の cscmng.properties の一つ前の cscmng.properties の設定が保存されています。

mserver.cfg.bak ( mserver.cfg のバックアップファイル)

現在の mserver.cfg の一つ前の mserver.cfg の設定が保存されています。

mserver.properties.bak ( mserver.properties のバックアップファイル)

現在の mserver.properties の一つ前の mserver.properties の設定が保存されています。

---

## 2.5 利用できるサービス部品の種類と適用範囲

Cosminexus サービスプラットフォームで利用できるサービス部品の種類を次の表に示します。

表 2-7 利用できるサービス部品の種類

サービス部品の種類	通信形態	プロトコル
Web サービス	同期	SOAP ( HTTP )
SessionBean		RMI-IIOP
MDB	非同期	WS-R
DB キュー		JMS

注

上の表に示すサービス部品以外に、データベースの操作をサービス部品として利用できます。なお、データベースの操作をサービス部品として利用する場合、操作できるデータベースは HiRDB と Oracle です。

以降の項では、サービス部品の種類ごとに適用範囲を説明します。

### 2.5.1 Web サービスを利用したサービス部品の適用範囲

Web サービスを利用したサービス部品の適用範囲を次に示します。

ポイント

このマニュアルで説明する WSDL および XML スキーマでは、特定のプレフィックスと名前空間 URI を対応づけて記述しています。プレフィックスと名前空間 URI の対応を次の表に示します。

表 2-8 プレフィックスと名前空間 URI の対応

プレフィックス	名前空間 URI
wsdl	http://schemas.xmlsoap.org/wsdl/
xsd	http://www.w3.org/2001/XMLSchema
soap	http://schemas.xmlsoap.org/wsdl/soap/

#### (1) 前提となる仕様

Web サービスを利用したサービス部品は、次のバージョンの仕様に従って作成されている必要があります。

## 2. システムを開発する前に

- SOAP 1.1
- WSDL 1.1
- WS-I Basic Profile 1.0a

### (2) SOAP の形式

Web サービスを利用したサービス部品では、SOAP の形式として次の条件を満たしている必要があります。

- 要求電文および応答電文が SOAP の本文に格納されていること。
- SOAP Header 要素を使用していないこと。

### (3) WSDL の形式

Web サービスを利用したサービス部品では、WSDL の形式として次の条件を満たしている必要があります。

- WSDL が、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」の WSDL のサポート範囲に関する内容を満たしていること。
- `wSDL:types` 要素以下のスキーマが、「2.5.5 XML スキーマの適用範囲」に示す条件を満たしていること。スキーマの条件の詳細については、「2.5.5 XML スキーマの適用範囲」を参照してください。
- `wSDL:import` 要素を使用していないこと。
- `mime:multipartRelated` 要素を使用していないこと。
- `rpc/literal` の場合、`wSDL:part` 要素の `type` 属性に `xsd:anyType` および `xsd:anySimpleType` を指定していないこと。
- バインディングとして SOAP バインディング以外が宣言されていないこと。
- `wSDL:types` 要素以下に存在するすべての XML スキーマの名前空間 (`targetNamespace`) に、有効な値が指定されていること。
- オペレーションの `style` 属性が「`rpc`」の場合、`soap:body` 要素の `namespace` 属性に記述された名前空間に「オペレーション名」、または「オペレーション名」+「Response」を `name` 属性として持つ `element` 要素を指定していないこと。
- `wSDL:operation` 要素のオペレーション名が 255 バイト以下であること。
- `schemaLocation` 属性値が、相対ファイルパスで指定されていること。
- `xsd:import` 要素に `schemaLocation` 属性値を指定しない場合、`wSDL:types` 要素以下のスキーマを使用して名前空間を解決していること。
- `targetNamespace` 属性が指定されているスキーマから `targetNamespace` 属性が指定されていないスキーマを `include` していないこと。
- `wSDL:portType` 要素の子要素である `wSDL:operation` 要素で定義した `wSDL:fault` 要素と対応するよう、`wSDL:binding` 要素の子要素である `wSDL:operation` 要素に `wSDL:fault` 要素を定義していること。
- `fault` を使用する場合、`fault` メッセージを定義する `wSDL:part` 要素が、必ず `element` 属性で `xsd:schema` 要素内の `element` 要素を参照していること。

なお、Cosminexus の開発支援コマンドである Java2WSDL コマンドを使用して、オペレーションの style 属性が「rpc」の WSDL ファイルを生成した場合、element 属性ではなく、type 属性で参照する WSDL ファイルが生成されます。そのため、生成された WSDL ファイルをそのまま使用しないで、変換してから使用します。変換の方法については、「4.3.2 サービス部品電文を作成する (Web サービスの場合)」の注意事項を参照してください。

Cosminexus サービスプラットフォームで fault を使用する場合、document スタイルの WSDL を使用することを推奨します。

- fault を使用する場合、WSDL fault 名 (wsdl:fault 要素の name 属性値) と、schema (wsdl:types 内) に定義された例外型 complexType を指す element 名 が、ネームスペースを含めて一致していること。

注

element 名とは、「wsdl:fault が参照する wsdl:part 要素が、element 属性を用いて指定する要素」のことです。WSDL を次の順にたどって特定できる要素を指します。

1. wsdl:fault
2. wsdl:message (この要素をたどらない場合もあります)
3. wsdl:part
4. xsd:element

#### (4) ライフサイクル

Cosminexus サービスプラットフォームの実行環境とサービス部品の間では、セッションを維持するように設定 (DeployScope に「Session」、「Application」を設定) しても、ステートフルな呼び出しはできません。サービスアダプタでは、常に「Request」での呼び出しです。

## 2.5.2 SessionBean を利用したサービス部品の適用範囲

SessionBean を利用したサービス部品の適用範囲を次に示します。

### (1) 前提となる仕様

SessionBean を利用したサービス部品は、EJB 2.0 の仕様に従って作成されている必要があります。

### (2) 利用できるサービス部品の配備先

SessionBean を利用したサービス部品として利用できるのは、Cosminexus 上に配備されているサービス部品です。

### (3) SessionBean の作成条件

SessionBean を利用したサービス部品は、次の条件に従って作成されている必要があります。

## 2. システムを開発する前に

ます。

### 引数・戻り値に指定できるクラス

SessionBean を利用したサービス部品の引数・戻り値に指定するクラスは、次の条件を満たしている必要があります。

- ユーザ定義クラスが基本型クラスで構成されていること。
- 引数・戻り値に指定されるユーザ定義型のクラスが、private なフィールドとそのフィールドに外部からアクセスするための public なアクセサメソッド (setXX, getXX) を持っていること (JavaBeans 形式)。
- 公開するメソッドの引数・戻り値に interface のユーザ定義型を指定していないこと。
- 公開するメソッドの引数・戻り値に abstract クラスのユーザ定義型クラスを指定していないこと。
- 公開するメソッドの引数・戻り値に java.lang.Object 以外のクラスを継承するユーザ定義型クラスを指定していないこと。
- 公開するメソッドの引数・戻り値に、異なるパッケージのユーザ定義型クラスを指定していないこと。
- 次の表に示す Java データ型のうち、利用できる Java データ型だけを利用していること。

表 2-9 Java データ型の利用可否

Java でのデータ型	メソッドの引数・戻り値としての利用の可否		
	直接使用する場合	配列として使用する場合	ユーザ定義データ型クラスのメンバ変数として使用する場合
boolean			
javax.xml.rpc.holders.BooleanHolder		×	×
byte			
javax.xml.rpc.holders.ByteHolder		×	×
byte[]			
javax.xml.rpc.holders.ByteArrayHolder		×	×
double			
javax.xml.rpc.holders.DoubleHolder		×	×
float			
javax.xml.rpc.holders.FloatHolder		×	×
int			
javax.xml.rpc.holders.IntHolder		×	×
long			

Java でのデータ型	メソッドの引数・戻り値としての利用の可否		
	直接使用する場合	配列として使用する場合	ユーザ定義データ型クラスのメンバ変数として使用する場合
javax.xml.rpc.holders.LongHolder		×	×
short			
javax.xml.rpc.holders.ShortHolder		×	×
java.lang.Byte			
javax.xml.rpc.holders.ByteWrapperHolder		×	×
java.lang.Byte[]			
java.lang.Double			
javax.xml.rpc.holders.DoubleWrapperHolder		×	×
java.lang.Float			
javax.xml.rpc.holders.FloatWrapperHolder		×	×
java.lang.Integer			
javax.xml.rpc.holders.IntegerWrapperHolder		×	×
java.lang.Long			
javax.xml.rpc.holders.LongWrapperHolder		×	×
java.lang.Object	×	×	×
javax.xml.rpc.holders.ObjectHolder	×	×	×
java.lang.Object[]	×	×	×
java.lang.Short			
javax.xml.rpc.holders.ShortWrapperHolder		×	×
java.lang.String			
javax.xml.rpc.holders.StringHolder		×	×
java.math.BigDecimal			
javax.xml.rpc.holders.BigDecimalHolder		×	×
java.math.BigInteger			
javax.xml.rpc.holders.BigIntegerHolder		×	×
java.util.Date			
javax.xml.namespace.QName			
javax.xml.rpc.holders.QNameHolder		×	×
java.lang.Boolean			
javax.xml.rpc.holders.BooleanWrapperHolder		×	×
java.util.Calendar			

## 2. システムを開発する前に

Java でのデータ型	メソッドの引数・戻り値としての利用の可否		
	直接使用する場合	配列として使用する場合	ユーザ定義データ型クラスのメンバ変数として使用する場合
javax.xml.rpc.holders.CalendarHolder		x	x

(凡例)

: 利用できます。

x : 利用できません。

### メソッドのオーバーロード

SessionBean を利用したサービス部品では、同じ名称のメソッドは指定できません。

### トランザクション

SessionBean のトランザクション管理種別 (transaction-type) やメソッドに割り当てるトランザクション属性 (trans-attribute) は、HCSC サーバからのトランザクションを引き継がないよう設定します。SessionBean を利用したサービス部品には、トランザクションを引き継ぎません。

サービス部品でのトランザクション属性の指定の可否を次の表に示します。

表 2-10 サービス部品でのトランザクション属性の指定可否

サービス部品のトランザクションモード	Tx 属性	指定の可否
BMT	-	
CMT	Required	x
	RequiresNew	
	Supports	x
	NotSupported	
	Mandatory	x
	Never	x

(凡例)

: 利用できます (EJB の仕様どおりに動作します)。

x : 利用できません。

- : 該当しません。

### メソッドの例外

メソッドの throws 節に記載できる例外は、javax.rmi.RemoteException、および java.lang.RuntimeException の継承クラスだけです。

### SessionBean の種別

SessionBean のサービス部品は、Stateless Session Bean として作成されている必要があります。

Cosminexus サービスプラットフォームの実行環境では、EJB サービスに対する実行要求ごとに接続 (create) と切断 (remove) を発行します。そのため、Stateful Session Bean を使用して内部的な状態を保持するようなサービス部品は利用できません。

#### (4) スキーマの形式

SessionBean を利用したサービス部品で利用するスキーマは、「2.5.5 XML スキーマの適用範囲」に示す条件を満たしている必要があります。スキーマの条件の詳細については、「2.5.5 XML スキーマの適用範囲」を参照してください。

## 2.5.3 Cosminexus RM のローカルキューを利用したサービス部品の適用範囲

Cosminexus RM のローカルキューを利用したサービス部品の適用範囲を次に示します。

### (1) 前提となる仕様

Cosminexus RM のローカルキューを利用したサービス部品は、EJB 2.0 の仕様に従って作成されている必要があります。

### (2) メッセージの種別

サービス部品が Cosminexus RM のローカルキューを利用している場合、サービスリクエストから送信される電文をサービス部品に送信するためのメッセージ種別として、次のメッセージ種別を利用できます。

- BytesMessage
- ObjectMessage
- TextMessage

上記以外のメッセージ種別は利用できません。

### (3) スキーマの形式

Cosminexus RM のローカルキューを利用したサービス部品で利用するスキーマは、「2.5.5 XML スキーマの適用範囲」に示す条件を満たしている必要があります。スキーマの条件の詳細については、「2.5.5 XML スキーマの適用範囲」を参照してください。

## 2.5.4 DB キューを利用したサービス部品の適用範囲

DB キューを利用したサービス部品の適用範囲を次に示します。

## 2. システムを開発する前に

### (1) 前提となる仕様

DB キューを利用したサービス部品は、次のバージョンの仕様に従って作成されている必要があります。

- TP1/Server Base Enterprise Option 02-00 以降
- EJB 2.0 ( サービス部品を MDB で実装した場合 )

### (2) メッセージの種別

サービス部品が DB キューの場合、サービスリクエストから送信される電文をサービス部品に送信するためのメッセージ種別として、`BytesMessage` を利用できます。

`BytesMessage` 以外のメッセージ種別は利用できません。

### (3) 利用できるデータベース

DB キューを作成できるデータベースは `HiRDB` だけです。

### (4) スキーマの形式

DB キューを利用したサービス部品で利用するスキーマは、「2.5.5 XML スキーマの適用範囲」に示す条件を満たしている必要があります。スキーマの条件の詳細については、「2.5.5 XML スキーマの適用範囲」を参照してください。

## 2.5.5 XML スキーマの適用範囲

各サービス部品で利用する XML スキーマの適用範囲について説明します。

### (1) 利用する XML スキーマの形式

XML スキーマは次の条件を満たしている必要があります。

- 次の要素を使用していないこと。
  - `xsd:any` 要素
  - `xsd:anyAttribute` 要素
  - `xsd:redefine` 要素
- `xsd:anyType` 型を使用していないこと。
- `xsd:anySimpleType` 型を使用していないこと。
- `xsd:element` 要素の型が定義されていること。なお、型の定義には、`type` 属性、`ref` 属性または内部定義 (`complexType` または `simpleType`) が使用できます。
- スキーマの XSD 名前空間が「`http://www.w3.org/2001/XMLSchema`」であること。
- `wsdl:part` から参照する要素が、必ず WSDL ファイルの `wsdl:types` 要素以下に定義した内部の XML スキーマに記述されていること。
- 選択した XML スキーマファイルに、必ずルート要素が記述されていること。
- スキーマのファイル名の長さが 128 バイト以下であること。

- XMLスキーマの属性 (attribute) の先頭と末尾に White space 文字 (半角スペース (#x20), タブ (#x9), および改行文字 (#xA または #xD)) を使用していないこと。

また、ビジネスプロセスの受付として標準受付を使用する場合、xsd:schema 要素の elementFormDefault 属性には、qualified を指定することをお勧めします。

## (2) 外部の XML スキーマを参照する場合の注意事項

XML スキーマから外部の XML スキーマを参照して利用する場合の注意事項を、次に示します。

- WSDL ファイルの wsdl:types 要素以下に定義した XML スキーマの xsd:import 要素の schemaLocation 属性が指定されていない場合、wsdl:types 要素以下の XML スキーマを使用して名前空間を解決している必要があります。
- XML スキーマファイルの xsd:import 要素の schemaLocation 属性は必ず指定してください。
- xsd:import 要素には、必ず namespace 属性を指定してください。
- xsd:import 要素および xsd:include 要素の schemaLocation 属性には、相対ファイルパスを指定してください。
- xsd:import 要素および xsd:include 要素では、参照の呼び出し階層を 20 階層以内にしてください。
- xsd:schema 要素の targetNamespace 属性指定されているスキーマからは、targetNamespace 属性が指定されていないスキーマを include しないでください。
- xsd:import 要素の schemaLocation 属性、および xsd:include 要素の schemaLocation 属性には、UTF-8 でパーセントエンコーディングした値でも、パーセントエンコーディングしていない値でも設定できます。ただし、パーセントエンコードし、かつファイルパスにディレクトリを含む場合、ディレクトリデリミタ (/) をエンコードしないでください。
- schemaLocation 属性のディレクトリデリミタには「/」を使用してください(「¥」は使用できません)。
- XML スキーマのファイルパスには、次の文字を使用しないでください。  
「;」、「?」、「:」、「@」、「&」、「=」、「+」、「\$」、「,」、「<」、「>」、「#」、「%」、「"」、「{」、「}」、「|」、「^」、「[」、「」」、「`」



# 3

## プロジェクトの管理とリポジトリの管理

この章では、HCSCTE を組み込んだ Eclipse での、HCSCTE のプロジェクトの作成・削除など、プロジェクトの管理について説明します。また、プロジェクトのプロパティの設定で指定したリポジトリの管理についても説明します。

---

3.1 プロジェクトの管理

---

3.2 リポジトリの管理

---

## 3.1 プロジェクトの管理

---

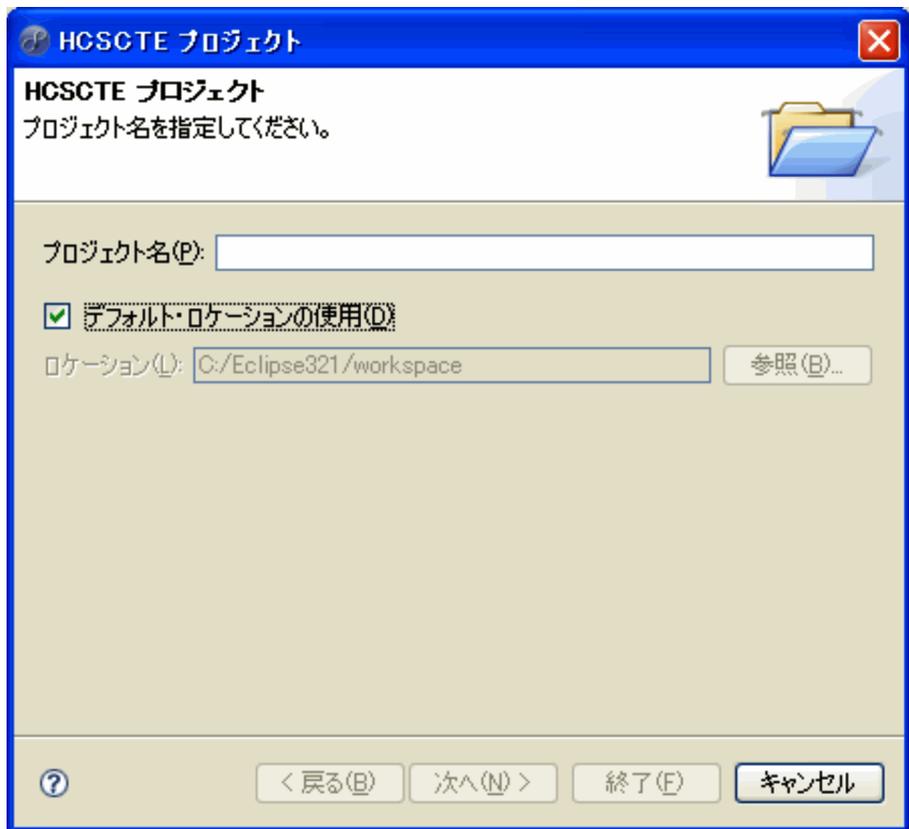
HCSC コンポーネントを開発する前に、プロジェクトを作成して、プロパティを設定します。また、作成したプロジェクトは、必要に応じて、エクスポートおよびインポートします。

### 3.1.1 プロジェクトを作成する

HCSCTE プロジェクトを作成する前に、Eclipse のメニューから [ ウィンドウ ] - [ 設定 ] - [ Java ] - [ ビルド・パス ] を選択し、「ソースおよび出力フォルダー」に「プロジェクト」を設定してください。

プロジェクトを作成する手順を次に示します。

1. Eclipse を起動します。
2. メニューから [ ファイル ] - [ 新規 ] - [ プロジェクト ] を選択します。  
[ 新規プロジェクト ] ダイアログが表示されます。
3. [ HCSCTE プロジェクト ] を選択して、[ 次へ ] ボタンをクリックします。  
[ HCSCTE プロジェクト ] ダイアログ ( HCSCTE プロジェクト新規作成のページ ) が表示されます。



4. 次の項目を設定して、[ 次へ ] ボタンをクリックします。

- プロジェクト名  
任意の名称を指定します。
- プロジェクト・コンテンツ  
[ デフォルトの使用 ] チェックボックスにチェックします。

[ HCSCTE プロジェクト ] ダイアログ ( HCSCTE リポジトリ設定のページ ) が表示されます。

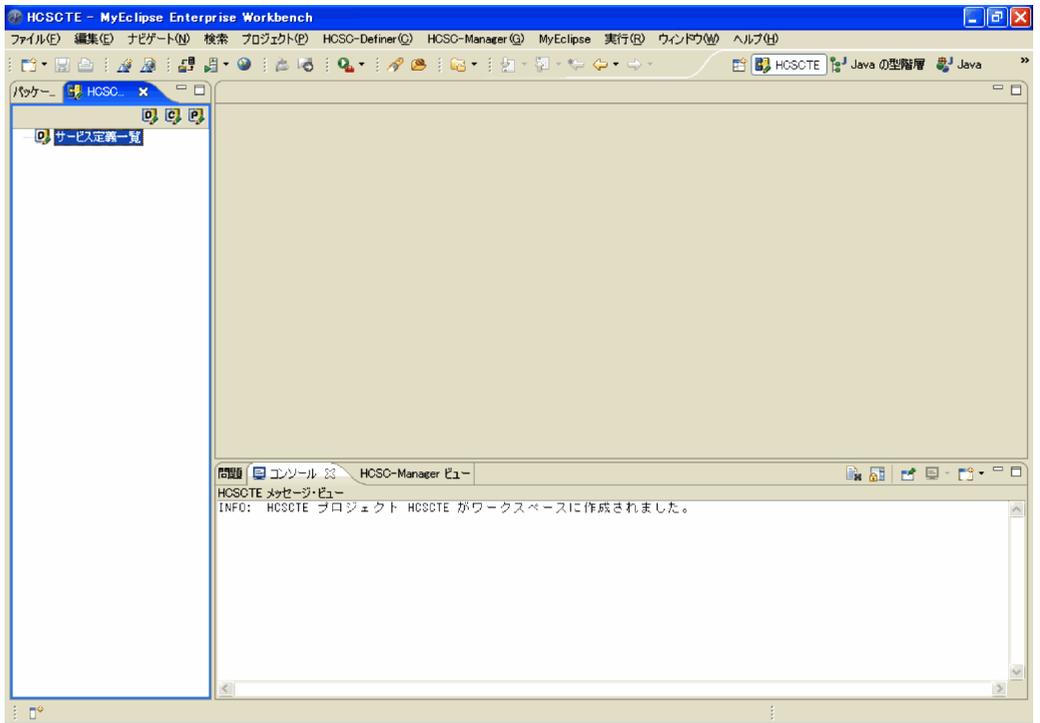
### 3. プロジェクトの管理とリポジトリの管理



5. 次の項目を設定して、[ 終了 ] ボタンをクリックします。

- リポジトリディレクトリ  
リポジトリ情報を格納する任意のディレクトリを指定します。リポジトリディレクトリを指定する場合は、次の点に注意してください。
  - リポジトリディレクトリのパスとプロジェクトのパスには、同じパスを指定しないでください。
  - パスは、絶対パスで指定してください。
  - パスの長さは、正規化された絶対パスでチェックされます。
- ログインユーザ名  
リポジトリログインに使用するユーザ名を指定します。

HCSCTE のプロジェクトが作成され、作成した HCSCTE のプロジェクトのパスペクティブが起動します。



**!** 注意事項

HCSCTE プロジェクトは、ワークスペースに一つしか作成できません。

### 3.1.2 プロパティを設定する

ここでは、HCSCTE のプロジェクトを作成したあとに実施するプロパティの設定について説明します。

#### (1) リポジトリの設定

プロジェクト作成時と異なるリポジトリを使用する場合、必要に応じてリポジトリとして利用するディレクトリを指定します。

リポジトリの設定については、「3.2.1 リポジトリを設定する」を参照してください。

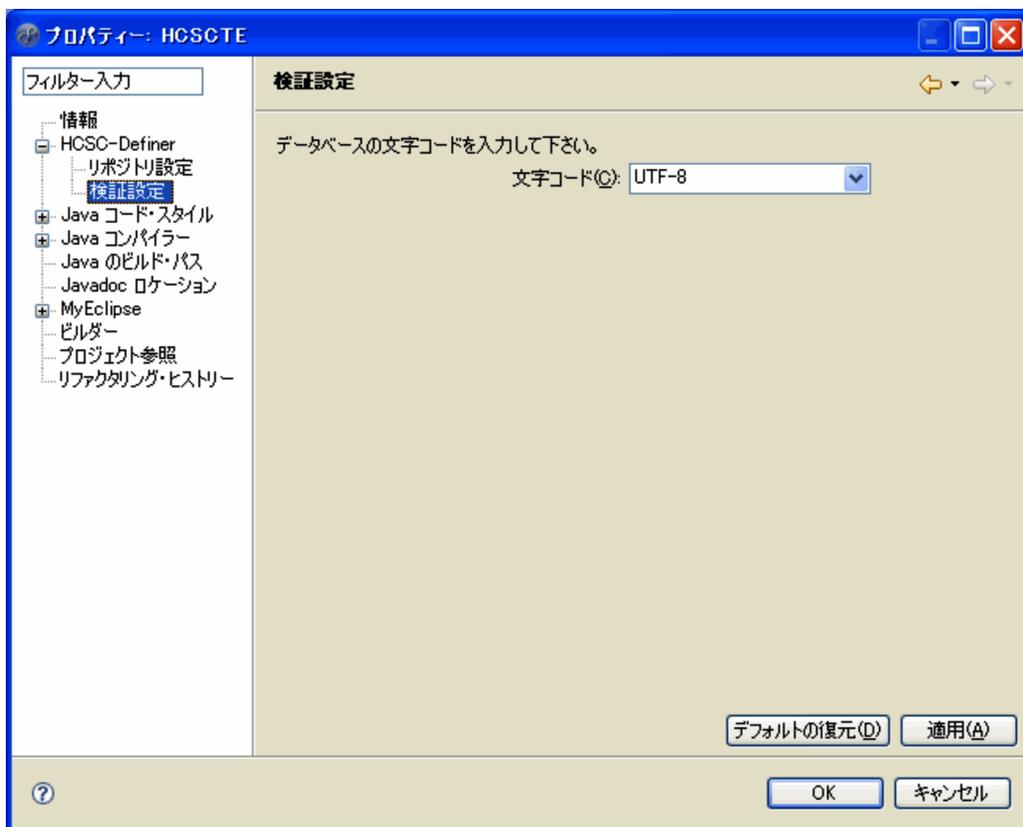
#### (2) 検証設定

検証設定では、データベースで使用する文字コードを指定します。

検証設定で文字コードを指定する手順を次に示します。

### 3. プロジェクトの管理とリポジトリの管理

1. Eclipse のメニューから, [プロジェクト] - [プロパティー] を選択します。  
[HCSCTE のプロパティー] ダイアログが表示されます。
2. ダイアログ左側のツリービューから, [HCSC-Definer] - [検証設定] を選択します。  
ダイアログ右側に, 検証設定に関する設定項目が表示されます。



3. [文字コード] に, データベースで使用する文字コードを指定します。  
ここで指定した値を使用して, サービス名やビジネスプロセスのアクティビティ名などの文字列長の検証が行われます。
4. [適用] ボタン, [OK] ボタンの順にクリックします。

注 「HCSCTE」の部分には, プロジェクト名が表示されます。

#### 3.1.3 プロジェクトをエクスポート / インポートする

Eclipse の標準機能を使用して, HCSCTE のプロジェクトをエクスポートできます。また, エクスポートした HCSCTE のプロジェクトを, インポートすることもできます。ただし, 一つのワークスペースに, 同時に二つ以上の HCSCTE プロジェクトを作成するこ

とはできません。

エクスポート方法およびインポート方法の詳細については、Eclipse のドキュメントを参照してください。

### 3.1.4 プロジェクトを削除する

HCSCTE のプロジェクトを削除する手順を次に示します。

1. パッケージ・エクスプローラーから削除するプロジェクトを選択します。
2. メニューから [ 編集 ] - [ 削除 ] を選択します。  
選択したプロジェクトが削除されます。

## 3.2 リポジトリの管理

---

リポジトリとは、開発環境と運用環境で共有する情報を格納するディレクトリのことをいいます。開発環境では、プロジェクトの作成時にリポジトリを設定します。また、設定したリポジトリは、必要に応じて、エクスポートおよびインポートします。

リポジトリの詳細については、マニュアル「Cosminexus サービスプラットフォーム概説」を参照してください。

### ! 注意事項

バージョン 07-60 以降では、開発環境と運用環境で同じリポジトリを共有できません。下位バージョンでリポジトリを共有していた場合、開発環境と運用環境で別々のリポジトリを用意して、移行してください。移行手順については、「付録 A.3 下位バージョンで開発環境と運用環境のリポジトリを共有していた場合の移行手順」を参照してください。

---

### 3.2.1 リポジトリを設定する

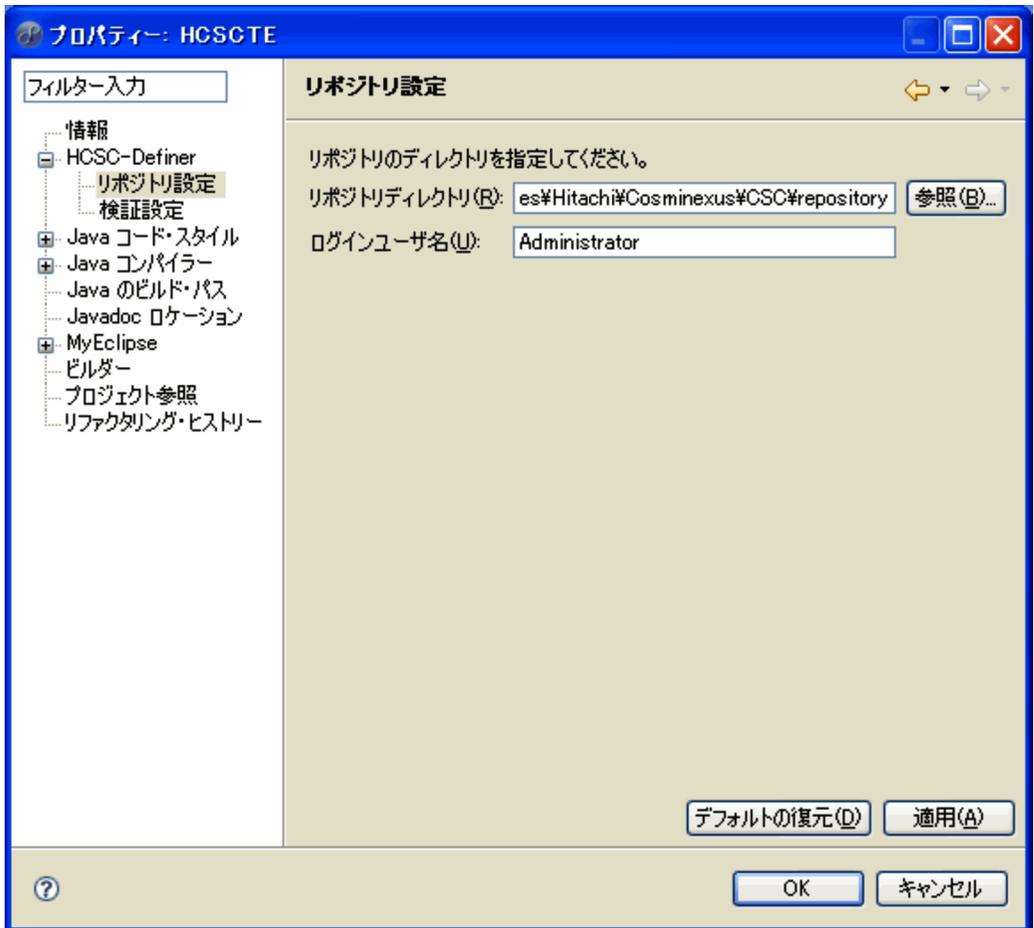
ここでは、プロジェクトの作成時に設定したリポジトリを変更する方法について説明します。

#### (1) リポジトリの指定

リポジトリとするディレクトリを指定します。

リポジトリを指定する手順を次に示します。

1. Eclipse のメニューから、[ プロジェクト ] - [ プロパティ ] を選択します。  
[ HCSCTE のプロパティ ] ダイアログが表示されます。
2. ダイアログ左側のツリービューから、[ HCSC-Definer ] - [ リポジトリ設定 ] を選択します。  
ダイアログ右側に、リポジトリの設定項目が表示されます。



3. [リポジトリディレクトリ]に、リポジトリとする任意のディレクトリを指定します。  
[リポジトリディレクトリ]を指定する場合は、次の点に注意してください。
  - すでに存在するディレクトリを指定してください。
  - リポジトリディレクトリのパスとプロジェクトのパスには、同じパスを指定しないでください。
  - パスは、絶対パスで指定してください。
  - パスの長さは、正規化された絶対パスでチェックされます。
4. [ログインユーザ名]に、ログインに使用する任意のユーザ名を入力します。
5. [適用]ボタン,[OK]ボタンの順にクリックします。

注 「HCSCTE」の部分には、プロジェクト名が表示されます。

## (2) リポジトリの初期化

リポジトリの初期化を行うと、リポジトリの内容をすべて消去し、リポジトリを初期状態に戻すことができます。

### 3. プロジェクトの管理とリポジトリの管理

リポジトリを初期化する前に、必ずすべてのエディタを閉じてください。また、HCSCTE プロジェクトの lib ディレクトリや src ディレクトリにアクセスしないようにしてください。エラーが発生した場合、次の点を確認してください。

- HCSCTE プロジェクトの src ディレクトリ以下のファイルを開いたエディタ（外部エディタを含む）を閉じているか
- HCSCTE プロジェクトの lib ディレクトリ以下のライブラリを利用したプログラムを実行していないか

リポジトリを初期化する手順を次に示します。

1. Eclipse のメニューから、[ HCSC-Definer ] - [ 定義情報管理 ] - [ 定義情報初期化 ] を選択します。  
メッセージダイアログが表示されます。

#### (3) リポジトリの変更

リポジトリとするディレクトリを変更する場合の手順は、(1) のリポジトリの指定と同じ手順になります。リポジトリを変更する場合、アダプタ定義やビジネスプロセス定義など、リポジトリの変更以外の作業は終了させてください。

[ デフォルトの復元 ] ボタンをクリックした場合、前回指定したリポジトリが設定されません。

#### (4) リポジトリの注意事項

リポジトリのファイルおよびディレクトリは、直接操作しないでください。ファイルおよびディレクトリの名前の変更、削除、内容の変更などを行った場合、動作の保証はできません。

## 3.2.2 リポジトリをエクスポートする

リポジトリのエクスポートでは、リポジトリを一つの ZIP ファイルにして、指定したディレクトリに保存することができます。

リポジトリをエクスポートする手順を次に示します。

1. Eclipse のメニューから、[ HCSC-Definer ] - [ 定義情報管理 ] - [ 全定義情報エクスポート ] を選択します。  
[ リポジトリエクスポート ] ダイアログが表示されます。
2. 保存するディレクトリを指定し、保存する ZIP ファイルのファイル名を入力します。
3. [ 保存 ] ボタンをクリックします。  
指定したディレクトリに ZIP ファイルが保存されます。  
保存先のディレクトリに同じファイル名称のファイルがある場合、上書きを確認するダイアログが表示されます。上書きしてよいときは、[ はい ] ボタンをクリックして保存します。

参考

---

必要に応じてリポジトリをエクスポートし、リポジトリのデータをバックアップすることをお勧めします。

---

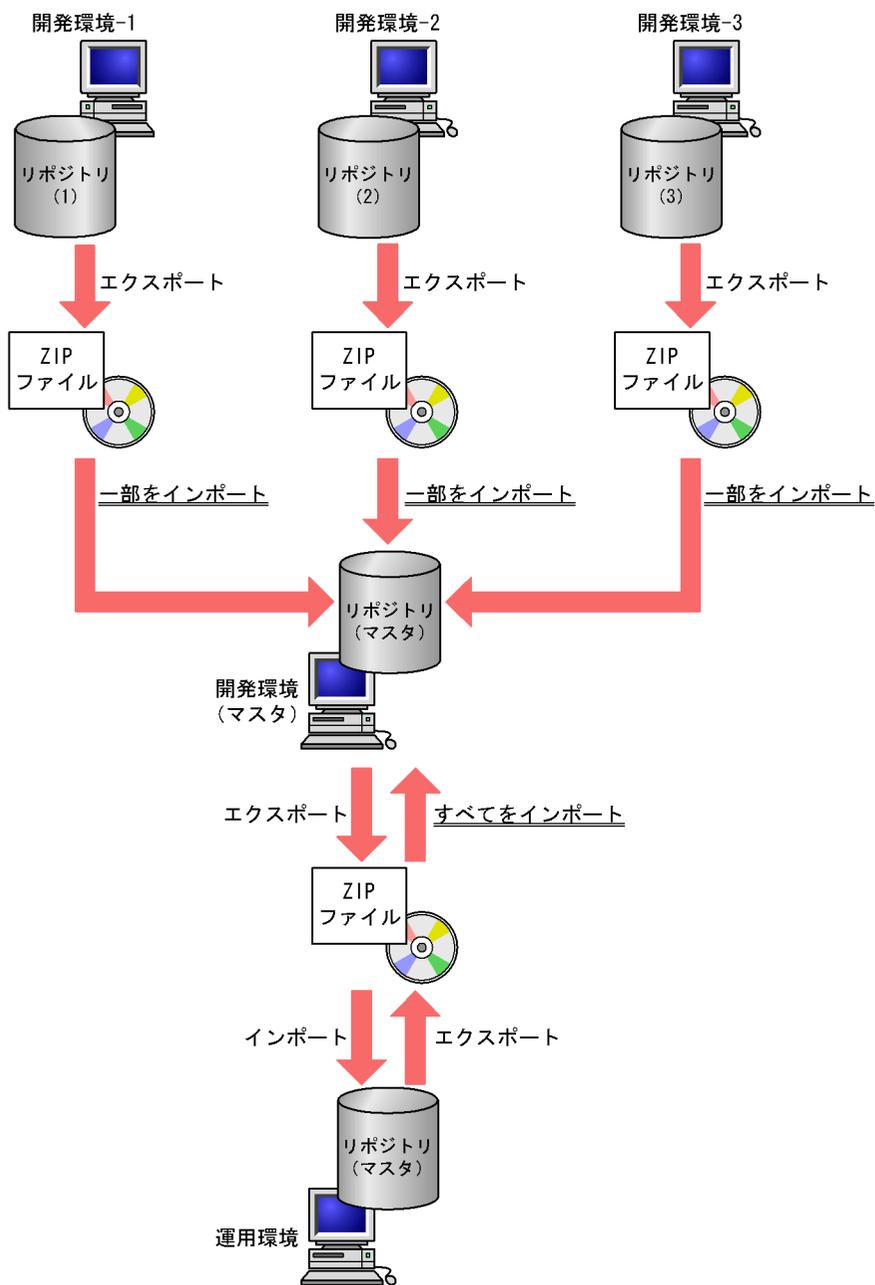
### 3.2.3 リポジトリをインポートする

リポジトリのインポートでは、エクスポートで ZIP ファイルにしたリポジトリを読み込むことができます。インポートできる定義は、リポジトリに格納されたサービス定義やシステム構成定義です。

リポジトリのインポート方法には、エクスポートされた ZIP ファイル内の定義情報をすべてインポートする方法と、サービス定義情報の一部だけをインポートする方法があります。二つのインポート方法の使い分けについて次の図に例示します。

### 3. プロジェクトの管理とリポジトリの管理

図 3-1 二つのインポート方法の使い分け（例）



注：エクスポートした情報は、媒体を利用しないでネットワーク経由で受け渡すこともできます。

図 3-1 の例に示すように、複数の開発環境で同時に HCSC コンポーネントを開発する場合、それぞれの開発環境のリポジトリをエクスポートして、マスターのリポジトリに必要な定義情報だけをインポートします。運用環境でエクスポートされたリポジトリの情報

を開発環境のリポジトリ（マスタ）にインポートする場合、すべての情報をインポートします。

#### ！ 注意事項

リポジトリをインポートする場合、アダプタ定義やビジネスプロセス定義などのすべてのエディタを閉じてください。また、HCSCTE プロジェクトの lib ディレクトリや src ディレクトリにアクセスしないようにしてください。エラーが発生した場合、次の点を確認してください。

- HCSCTE プロジェクトの src ディレクトリ以下のファイルを開いたエディタ（外部エディタを含む）を閉じているか
- HCSCTE プロジェクトの lib ディレクトリ以下のライブラリを利用したプログラムを実行していないか

二つのインポート方法の詳細を次に説明します。

### (1) リポジトリのすべての定義情報をインポートする

#### (a) インポート手順

リポジトリ中のサービス定義情報、システム構成定義情報を一括してインポートする手順を次に示します。

1. Eclipse のメニューから、[ HCSC-Definer ] - [ 定義情報管理 ] - [ 全定義情報インポート ] を選択します。  
インポートを確認するダイアログが表示されます。
2. [ はい ] ボタンをクリックします。  
[ リポジトリインポート ] ダイアログが表示されます。



3. インポートしたい定義のチェックボックスにチェックします。  
初めてインポートする場合は、[ サービス定義 ] と [ システム構成定義 ] の両方にチェックしてください。
4. [ OK ] ボタンをクリックします。

### 3. プロジェクトの管理とリポジトリの管理

[リポジトリインポート] ダイアログが表示されます。

5. 読み込む ZIP ファイルを指定します。

6. [開く] ボタンをクリックします。

7. [OK] ボタンをクリックします。

「3.2.1(1) リポジトリの指定」で指定したディレクトリにリポジトリが読み込まれます。

#### (b) 注意事項

- 旧バージョンのリポジトリをインポートすると、リポジトリ情報が今バージョンに引き継がれます。
- Java 呼出アクティビティで利用するユーザ定義 Java クラス、またはユーザ定義 Java クラスの前提ライブラリをインポートした場合、HCSCTE プロジェクトをビルドしてください。
- サービス定義、システム構成定義のどちらか片方だけをインポートする場合、ZIP ファイル中のインポート対象の定義情報と現在のリポジトリ内の情報（サービス定義またはシステム構成定義）の整合性が保たれている必要があります。不整合がある場合はエラーとなり、サービス定義やシステム構成定義をリポジトリにインポートできません。

リポジトリのインポートで整合性をチェックしている内容を次に示します。

#### サービス定義だけをインポートする場合

リポジトリ内の公開済みサービス定義と、対応する ZIP ファイル中のサービス定義について、整合性をチェックした場合に、エラーになる条件を次に示します。

表 3-1 リポジトリのインポートの整合性チェックでエラーになる条件（サービス定義だけをインポートする場合 1）

項番	エラーになる条件
1	サービス ID が一致しない
2	サービス名が一致しない <sup>1</sup>
3	種別（サービスアダプタまたはビジネスプロセス）が一致しない <sup>1</sup>
4	ビジネスプロセスのバージョンが一致しない <sup>1, 2</sup>
5	ビジネスプロセスに定義されているユーザ定義受付の数が一致しない <sup>1</sup>
6	ビジネスプロセスに定義されているユーザ定義受付の受付 ID が一致しない <sup>1</sup>
7	ビジネスプロセスに定義されているユーザ定義受付の受付名が一致しない <sup>1</sup>
8	ZIP ファイル中のサービス定義がパッケージングされていない <sup>1, 2</sup>

注 1 サービス ID が一致するサービス定義同士をチェックします。

注 2 旧バージョンで作成したリポジトリの場合はチェックしません。

また、リポジトリ内の公開済みサービス定義と、サービス ID が一致する ZIP ファイル中のサービス定義について、整合性をチェックした場合に、エラーにな

る条件を次に示します。なお、リポジトリ内のシステム構成定義で定義されているサーバ構成によって、エラーになる条件が異なります。

表 3-2 リポジトリのインポートの整合性チェックでエラーになる条件（サービス定義だけをインポートする場合 2）

項番	リポジトリ内の情報		公開済みのサービス定義（サービスの種別）	エラーになる条件 (サービス ID が一致する ZIP ファイル中のサービス定義)
	データベースの使用有無	Cosminexus RM の使用有無		
1	使用する	使用する	すべて	-
2	使用しない	使用しない	ビジネスプロセス	ステータスの永続化に yes が設定されている
3			<ul style="list-style-type: none"> <li>• Web サービス</li> <li>• SessionBean</li> <li>• カスタムアダプタ</li> </ul>	サービスの種別に MDB_WSR または MDB_DBQ が設定されている
4	使用する	使用しない	ビジネスプロセス	ステータスの永続化に yes が設定されている
5			<ul style="list-style-type: none"> <li>• Web サービス</li> <li>• SessionBean</li> <li>• カスタムアダプタ</li> </ul>	サービスの種別に MDB_WSR または MDB_DBQ が設定されている

(凡例)

- : エラーになる条件はありません。

エラーが発生した場合の対処方法

現状のリポジトリ内のサービスをすべて非公開にして、再度サービス定義だけインポートをしてください。

システム構成定義だけをインポートする場合

ZIP ファイル中の公開済みサービス定義と、リポジトリ内のサービス定義について、整合性をチェックした場合に、エラーになる条件を次に示します。

表 3-3 リポジトリのインポートの整合性チェックでエラーになる条件（システム構成定義だけをインポートする場合 1）

項番	エラーになる条件
1	サービス ID が一致しない
2	サービス名が一致しない <sup>1</sup>
3	種別（サービスアダプタまたはビジネスプロセス）が一致しない <sup>1</sup>

### 3. プロジェクトの管理とリポジトリの管理

項番	エラーになる条件
4	ビジネスプロセスのバージョンが一致しない <sup>1, 2</sup>
5	ビジネスプロセスに定義されているユーザ定義受付の数が一致しない <sup>1</sup>
6	ビジネスプロセスに定義されているユーザ定義受付の受付 ID が一致しない <sup>1</sup>
7	ビジネスプロセスに定義されているユーザ定義受付の受付名が一致しない <sup>1</sup>
8	ZIP ファイル中のサービス定義がパッケージングされていない <sup>1, 2</sup>

注 1 サービス ID が一致するサービス定義同士をチェックします。

注 2 旧バージョンで作成したリポジトリの場合はチェックしません。

また、ZIP ファイル中の公開済みサービス定義と、サービス ID が一致するリポジトリ内のサービス定義について、整合性をチェックした場合に、エラーになる条件を次に示します。なお、ZIP ファイル中のシステム構成定義で定義されているサーバ構成によって、エラーになる条件が異なります。

表 3-4 リポジトリのインポートの整合性チェックでエラーになる条件（システム構成定義だけをインポートする場合 2）

項番	ZIP ファイル中の情報		エラーになる条件 (サービス ID が一致する ZIP ファイル中のサービス定義)
	システム構成定義で定義されているサーバ構成	公開済みのサービス定義 (サービスの種別)	
	データベースの使用有無	Cosminexus RM の使用有無	
1	使用する	使用する	すべて
2	使用しない	使用しない	ビジネスプロセス
3			<ul style="list-style-type: none"> <li>• Web サービス</li> <li>• SessionBean</li> <li>• カスタムアダプタ</li> </ul>
4	使用する	使用しない	ビジネスプロセス
5			<ul style="list-style-type: none"> <li>• Web サービス</li> <li>• SessionBean</li> <li>• カスタムアダプタ</li> </ul>

(凡例)

- : エラーになる条件はありません。

エラーが発生した場合の対処方法

次の手順を実行して、対処してください。

1. 現状のリポジトリをエクスポート (rep.zip) して退避します。
2. インポートしようとしたアーカイブファイルから、サービス定義とシステム構成定義の両方をインポートします。
3. 手順 2 でインポートしたサービス定義をすべて非公開にします。
4. 手順 1 で退避しておいた rep.zip から、サービス定義だけをインポートします。

## (2) リポジトリのサービス定義情報の一部だけをインポートする

サービス定義情報の一部だけをインポートする場合、インポートの対象として選択できるのは次の情報です。

- HCSC コンポーネント (サービスアダプタ, ビジネスプロセス) に関する定義
- ユーザ定義 Java クラスのソースファイル
- ユーザ定義 Java クラスの前提ライブラリファイル

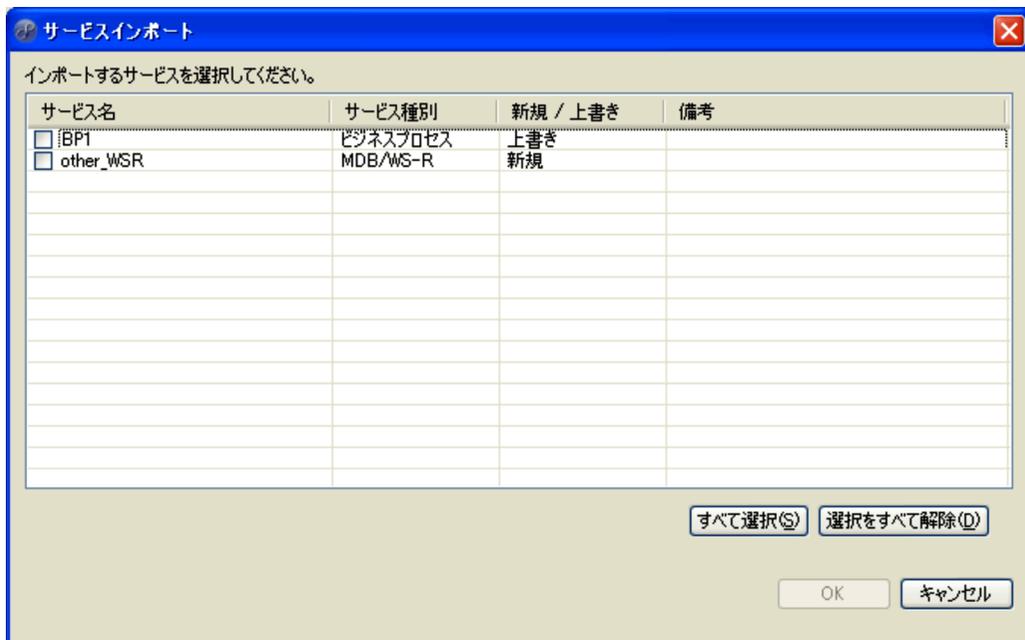
インポートの対象は複数選択できます。インポート対象として選択したビジネスプロセスから呼び出されるサービスアダプタおよびビジネスプロセスは、必要に応じて同時にインポートされます。また、インポート対象のビジネスプロセスに関連するユーザ定義受付も同時にインポートされます。

### (a) インポート手順

リポジトリの一部の情報だけをインポートする手順を次に示します。

1. Eclipse のメニューから, [ HCSC-Definer ] - [ 定義情報管理 ] - [ サービスインポート ] を選択します。  
アーカイブファイル (ZIP ファイル) を選択するダイアログが表示されます。
2. インポートする情報を含むアーカイブファイルを選択して, [ 開く ] ボタンをクリックします。  
[ サービスインポート ] ダイアログ (サービスの選択) が表示されます。

### 3. プロジェクトの管理とリポジトリの管理

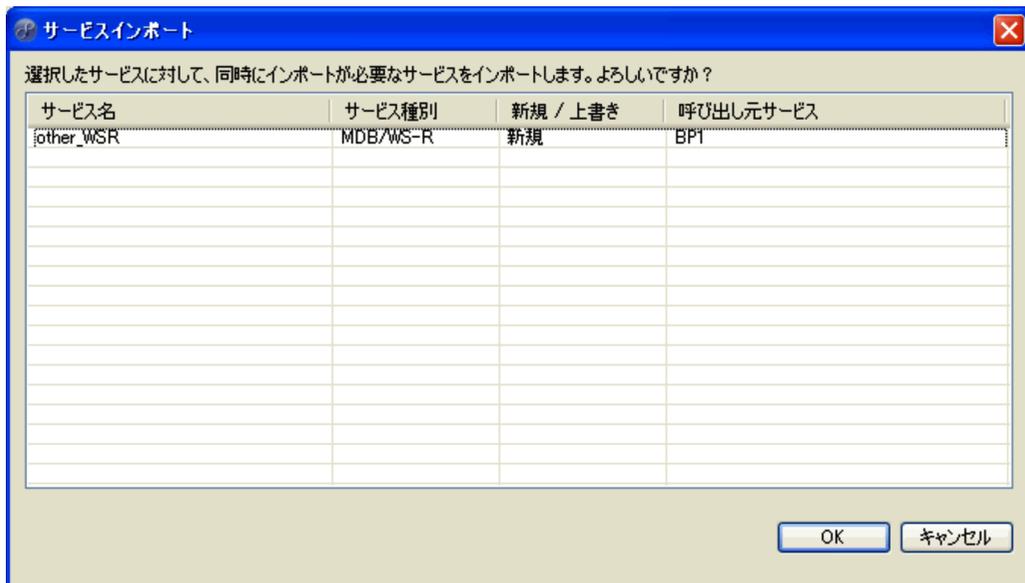


3. インポートするサービスのチェックボックスにチェックします。

4. [OK] ボタンをクリックします。

インポート必須サービスがある場合

[サービスインポート] ダイアログ (インポート必須サービスの確認) が表示されます。手順 5. に進んでください。



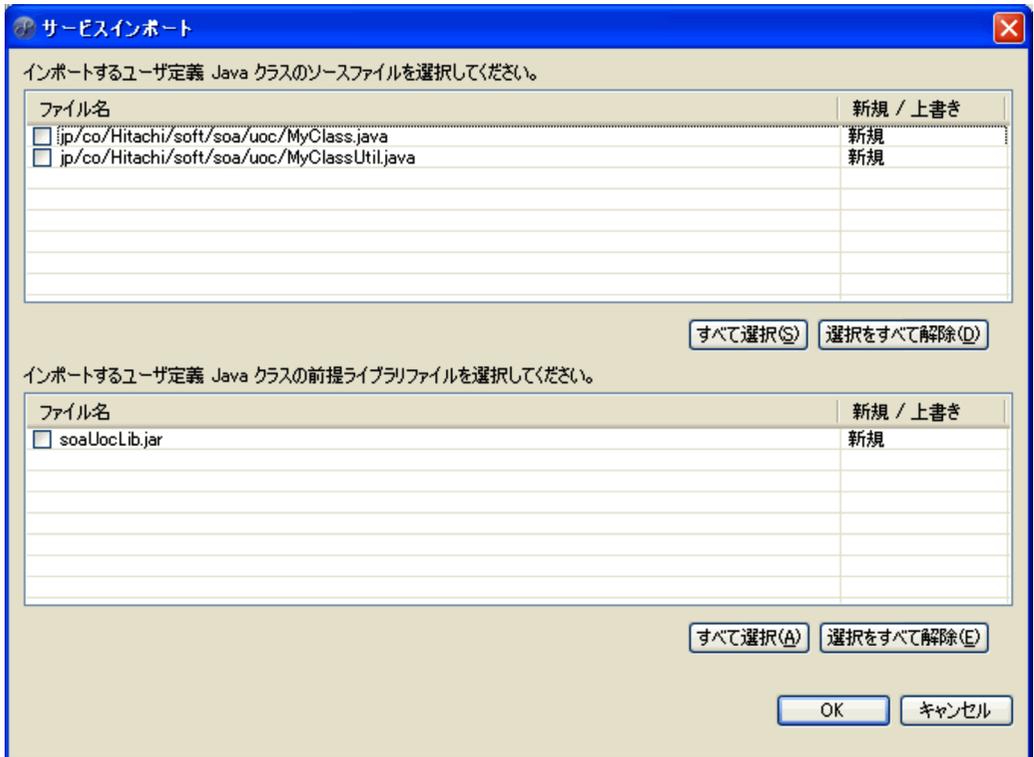
## インポート必須サービスがない場合

ユーザ定義 Java クラスの有無がチェックされます。

ユーザ定義 Java クラスがある場合、[ サービスインポート ] ダイアログ ( ユーザ定義 Java クラスの選択 ) が表示されます。手順 6. に進んでください ([ サービスインポート ] ダイアログ ( ユーザ定義 Java クラスの選択 ) は手順 5. の画面図を参照)。

ユーザ定義 Java クラスがない場合、「3.2.1(1) リポジトリの指定」で指定したディレクトリに、選択したリポジトリの内容が読み込まれます。

- 表示されたインポート必須サービスを確認して、[ OK ] ボタンをクリックします。  
ユーザ定義 Java クラスの有無がチェックされます。  
ユーザ定義 Java クラスがある場合、[ サービスインポート ] ダイアログ ( ユーザ定義 Java クラスの選択 ) が表示されます。  
ユーザ定義 Java クラスがない場合、「3.2.1(1) リポジトリの指定」で指定したディレクトリに、選択したリポジトリの内容が読み込まれます。



- インポートするユーザ定義 Java クラス、およびその前提ライブラリファイルのチェックボックスにチェックします。
- [ OK ] ボタンをクリックします。  
「3.2.1(1) リポジトリの指定」で指定したディレクトリに、選択したリポジトリの内容が読み込まれます。

### 3. プロジェクトの管理とリポジトリの管理

#### ポイント

---

- 同じ名称の HCSC コンポーネントがリポジトリで公開されている場合など、インポートの対象として選択できない場合、[ サービスインポート ] ダイアログ ( サービスの選択 ) 中のサービスの表示は非活性になります。[ 備考 ] の内容を参考に、原因を取り除いてから再度インポートしてください。
  - インポートした HCSC コンポーネント ( サービスアダプタ、ビジネスプロセス ) は、必ずパッケージングしてください。パッケージングの方法については、「9.2 パッケージング」を参照してください。
  - Java 呼出アクティビティで利用するユーザ定義 Java クラス、またはユーザ定義 Java クラスの前提ライブラリをインポートした場合、HCSCTE プロジェクトをビルドしてください。
- 

#### (b) 注意事項

- リポジトリ中で同じ名称の HCSC コンポーネントが公開済みの場合、インポートできません。インポートする前に非公開にしてください。
- リポジトリのサービス定義情報の一部だけをインポートする場合、システム構成定義情報はインポートされません。すでに公開している HCSC コンポーネントをインポートすると、インポートした HCSC コンポーネントは非公開になります。
- インポート先のリポジトリに、インポート対象の HCSC コンポーネントと同じ名称の HCSC コンポーネントがある場合、内容がすべて上書きされます。サービス ID、フォーマット ID、ユーザ定義受付 ID はすべて更新されます。
- バージョンアップしたビジネスプロセスをインポートした場合、インポートしたビジネスプロセスのバージョンは「1」になります。
- 古いバージョンで作成したリポジトリから、サービス定義情報の一部だけをインポートすることはできません。次の手順でリポジトリを移行し、サービス定義情報の一部をインポートしてください。
  1. 現在のリポジトリをアーカイブファイルにエクスポートします。
  2. 旧バージョンのアーカイブファイルから、すべての定義情報をインポートします。
  3. 手順 2 でインポートした定義情報を別のアーカイブファイルにエクスポートします。
  4. 手順 1 でエクスポートしたアーカイブファイルから、すべての定義情報をインポートします。
  5. 手順 3 でエクスポートしたアーカイブファイルから、サービス定義情報の一部をインポートします。

# 4

## 電文フォーマットの作成

この章では、電文フォーマットの作成について説明します。

---

4.1 電文フォーマットとデータ変換

---

4.2 電文フォーマットの種類

---

4.3 電文フォーマット (XML フォーマット定義ファイル) の作成方法

---

4.4 電文フォーマット (バイナリフォーマット定義ファイル) の作成方法

---

## 4.1 電文フォーマットとデータ変換

---

サービスリクエスト、サービスアダプタ、およびサービス部品の間でやり取りされる電文とその形式（電文フォーマット）、およびデータ変換の概要について説明します。

### （1）電文の種類

サービスリクエストからサービスアダプタを経由して、サービス部品の実行を要求するための電文を要求電文といいます。

逆に、サービスアダプタを経由して、サービス部品の実行結果をサービスリクエストに応答するための電文を応答電文といいます。

### （2）電文フォーマット

サービスリクエスト、サービスアダプタ、およびサービス部品の間でやり取りされる電文の形式を電文フォーマットといいます。

サービスアダプタには、リクエストを受け付けるための標準的な電文フォーマットがあります。これを標準電文といいます。また、サービス部品独自の電文のフォーマットをサービス部品電文といいます。サービス部品電文には、データベースの操作をサービス部品として利用する場合のデータベース側の電文フォーマットである DB アダプタ電文があります。

電文フォーマットを定義したファイルを、電文フォーマット定義ファイルといいます。

電文フォーマット定義ファイルには、扱うデータによって XML フォーマット定義ファイルとバイナリフォーマット定義ファイルの 2 種類があります。電文フォーマット定義ファイルの種類については、「4.2 電文フォーマットの種類」を参照してください。

### （3）データ変換

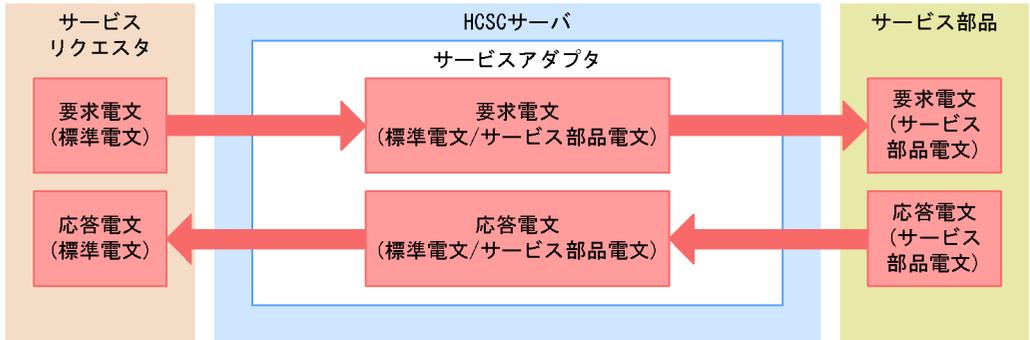
通常は、サービスリクエストからサービス部品の呼び出し要求があった場合は、標準電文でサービス部品を呼び出せます。しかし、サービスリクエストから受け付けた要求電文（電文フォーマットは標準電文）でサービス部品を呼び出せない場合、要求電文の電文フォーマットをサービス部品の電文フォーマット（サービス部品電文）に変換する必要があります。この変換をデータ変換といいます。同様に、サービス部品からの応答電文をサービスリクエストに返せない場合も、電文フォーマットを変換して標準電文の電文フォーマットに変換する必要があります。

データ変換の定義方法については、「7. データ変換の定義」を参照してください。

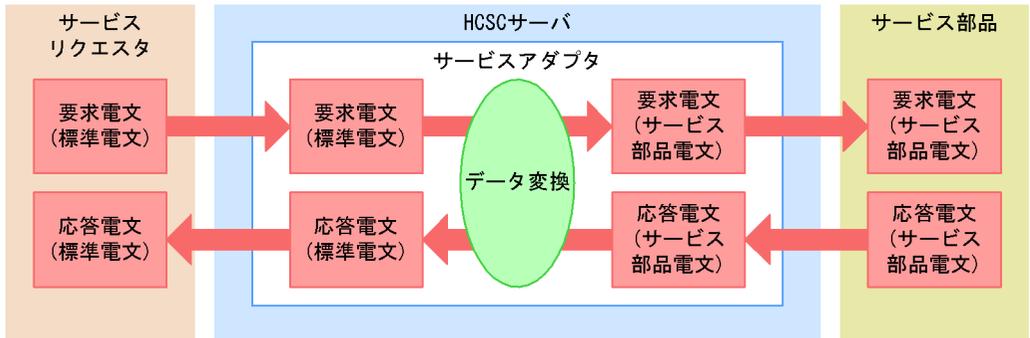
サービスアダプタを利用する場合のサービスリクエストとサービス部品間の電文の流れを次の図に示します。

図 4-1 サービスリクエスタとサービス部品間の電文の流れ（サービスアダプタを利用する場合）

●標準電文とサービス部品電文の形式が同一の場合（データ変換は不要）



●標準電文とサービス部品電文の形式が異なる場合（データ変換が必要）



(凡例)

- : 電文（および電文フォーマット）  
 : 電文の流れ

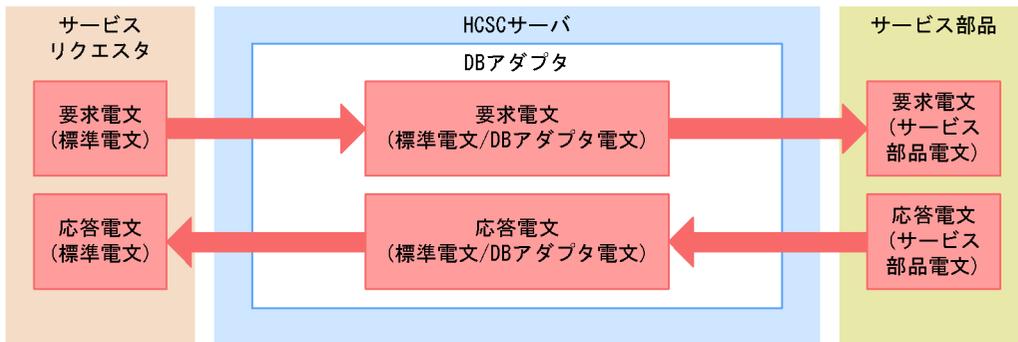
DB アダプタを利用する場合は、標準電文と DB アダプタ電文の流れになります。電文の流れは、基本的にはサービスアダプタと同様です。

DB アダプタを利用する場合のサービスリクエスタとサービス部品（データベース）間の電文の流れを次の図に示します。

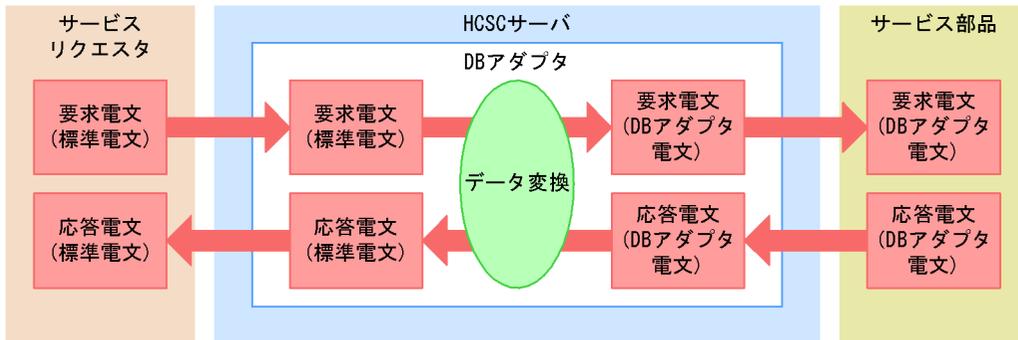
#### 4. 電文フォーマットの作成

図 4-2 サービスリクエスタとサービス部品間の電文の流れ (DB アダプタを利用する場合)

- 標準電文とDBアダプタ電文の形式が同一の場合 (データ変換は不要)



- 標準電文とDBアダプタ電文の形式が異なる場合 (データ変換が必要)



(凡例)

- : 電文 (および電文フォーマット)
- : 電文の流れ

## 4.2 電文フォーマットの種類

---

サービス部品の実行に利用する電文で XML 形式のデータを扱うか、バイナリ形式 (XML 形式以外) のデータを扱うかによって、作成する電文フォーマットは異なります。

### (1) XML 形式のデータを扱う場合

XML 形式のデータを扱う場合、XML フォーマット定義ファイルを作成します。

XML フォーマット定義ファイルは、XML スキーマファイル (拡張子: .xsd) として作成します。作成する XML フォーマット定義ファイルの内容や作成方法は、実行するサービス部品の種類によって異なります。サービス部品ごとの XML フォーマット定義ファイルの作成方法については、「4.3 電文フォーマット (XML フォーマット定義ファイル) の作成方法」を参照してください。

### (2) バイナリ形式のデータを扱う場合

バイナリ形式 (XML 形式以外) のデータを扱う場合、バイナリフォーマット定義ファイルを作成します。

バイナリフォーマット定義ファイルは、バイナリ形式のデータ内の値の格納形式 (値の長さや、ほかの値との関係など) を定義し、定義した情報を XML スキーマファイルに付加したファイル (拡張子: .fdx) です。バイナリフォーマット定義ファイルは、XML の仕様にあわせて作成されるため、データ変換を定義するときの変換元および変換先として、XML フォーマット定義ファイルと同様に利用できます。

バイナリフォーマット定義ファイルの作成方法については、「4.4 電文フォーマット (バイナリフォーマット定義ファイル) の作成方法」を参照してください。

なお、バイナリ形式のデータを扱う場合、またはデータ変換で値置換えファンクションを使用する場合、開発環境には「uCosminexus 日立コード変換 - Development Kit」、実行環境には「uCosminexus 日立コード変換 - Server Runtime」(Windows の場合) または「uCosminexus 日立コード変換 - Runtime」(UNIX の場合) が必要です。

## 4.3 電文フォーマット (XML フォーマット定義ファイル) の作成方法

---

サービス部品の実行に利用する電文で XML 形式のデータを利用する場合の、電文フォーマット (XML フォーマット定義ファイル) の作成方法について、以降の項で説明します。

### 4.3.1 標準電文を作成する

標準電文の XML フォーマット定義ファイルは、実行するサービス部品の種類に関係なく、Eclipse が提供する WST (Web Standard Tools) などを利用して作成します。

作成する XML フォーマット定義ファイルは、「2.5.5 XML スキーマの適用範囲」に示す条件を満たしている必要があります。条件の詳細については、「2.5.5 XML スキーマの適用範囲」を参照してください。

サービスアダプタおよびビジネスプロセスの定義で、利用する電文フォーマットファイルを選択したときにエラー (メッセージ ID が KECX で始まるエラーメッセージ) が発生した場合、XML スキーマファイルにエラーがあります。この場合、マニュアル「Cosminexus XML Processor ユーザーズガイド」のメッセージを参照して対処してください。

### 4.3.2 サービス部品電文を作成する (Web サービスの場合)

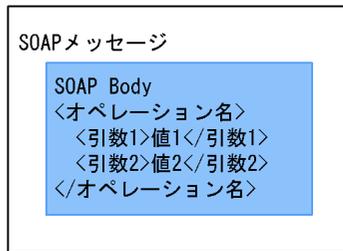
利用するサービス部品が、Cosminexus が提供している SOAP アプリケーション開発支援機能を使用して作成した SOAP アプリケーションに対応する Web サービスの場合、サービス部品電文の XML フォーマット定義ファイルを作成する必要はありません。

ただし、SOAP の WSDL 生成機能 (Java2WSDL コマンド) で WSDL を作成しておく必要があります。Java2WSDL コマンドで WSDL を生成する機能については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。

Web サービスの場合、オブジェクト間の送受信に使用するメッセージとして、SOAP メッセージを使用します。

通信スタイルが rpc 形式の場合の電文の例を次の図に示します。

図 4-3 電文の形式



SOAP メッセージおよびその形式については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。

注意事項：ユーザ定義の例外を throw する Web サービスの場合

SOAP の WSDL 生成機能で生成した WSDL の通信スタイルが rpc 形式の場合、WSDL を修正し、対応するアダプタを作成する必要があります。

生成した WSDL の通信スタイルが document 形式の場合、WSDL を修正する必要はありません。Cosminexus サービスプラットフォームで fault を使用する場合、document 形式の WSDL を使用することを推奨します。

WSDL の修正手順を次に示します。

1. xsd:element を定義します。  
xsd:element 要素は、wsdl:definitions 要素の targetNamespace 属性で示されたネームスペースと同じ targetNamespace 属性を持つ wsdl:types 要素中の xsd:schema 要素内に定義してください。
2. wsdl:portType 要素内の wsdl:operation 要素中の wsdl:fault 要素の message 属性の値から対応する wsdl:message 要素を取得し、内部に含まれる wsdl:part 要素を確認します。
3. 手順 1. で追加した xsd:element 要素に name 属性を追加します。  
値は手順 2. で取得した wsdl:part 要素の type 属性のローカル名と同じ値にします。
4. 手順 3. と同様に、type 属性を追加します。  
値は手順 2. で取得した wsdl:part 要素の type 属性と同じ値にします。
5. 手順 2. で確認した wsdl:part 要素から type 属性を削除し、新たに element 属性を追加します。  
ネームスペースプレフィックスは、手順 1. で参照した wsdl:types 要素内の xsd:schema 要素を示すプレフィックスにします。また、ローカル名は手順 3. で追加した name 属性と同じ値にします。

WSDL を修正したあとは、Web サービスそのものを再作成する必要があります。次に示すコマンドを実行して、修正した WSDL から Web サービス用のスケルトンを再生成してください。

#### 4. 電文フォーマットの作成

```
WSDL2Java -s <WSDLファイル>
```

WSDL2Java コマンドの詳細については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」のスケルトンの生成に関する内容を参照してください。

### 4.3.3 サービス部品電文を作成する（SessionBean の場合）

利用するサービス部品が SessionBean の場合、サービス部品電文の XML フォーマット定義ファイルを作成する必要はありません。XML フォーマット定義ファイルは、サービスアダプタの作成時に自動的に作成されます。

### 4.3.4 サービス部品電文を作成する（MDB（WS-R または DB キュー）の場合）

サービス部品電文の XML フォーマット定義ファイルは、Eclipse が提供する WST（Web Standard Tools）などを利用して作成します。

作成する XML フォーマット定義ファイルは、「2.5.5 XML スキーマの適用範囲」に示す条件を満たしている必要があります。条件の詳細については、「2.5.5 XML スキーマの適用範囲」を参照してください。

### 4.3.5 サービス部品電文を作成する（DB アダプタ電文の場合）

DB アダプタ電文の XML フォーマット定義ファイル（DB アダプタ用の XML フォーマット定義ファイル）は、SQL オペレーション定義ファイルを作成してから、csamkxmls コマンドで作成します。

SQL オペレーション定義ファイルは、実行環境で DB アダプタを介してアクセスするデータベースの情報や、SQL の実行形式を定義するファイルです。DB アダプタ用の XML フォーマット定義ファイルを作成するときの基となるファイルです。

SQL オペレーション定義ファイルと DB アダプタ用の XML フォーマット定義ファイルの作成方法を次に説明します。

#### （1）SQL オペレーション定義ファイルを作成する

SQL オペレーション定義ファイルは、テキストエディタなどを使用して XML 文書形式で作成します。XML 文書で使用できない「<」、「>」、「&」などの文字を使用する場合、エスケープするか、CDATA セクションを利用する必要があります。

要素、属性およびテキストデータは、W3C の XML の仕様に従って指定します。

なお、XML 名前空間は使用できません。

作成する XML 文書の詳細を説明します。

(a) ファイル名

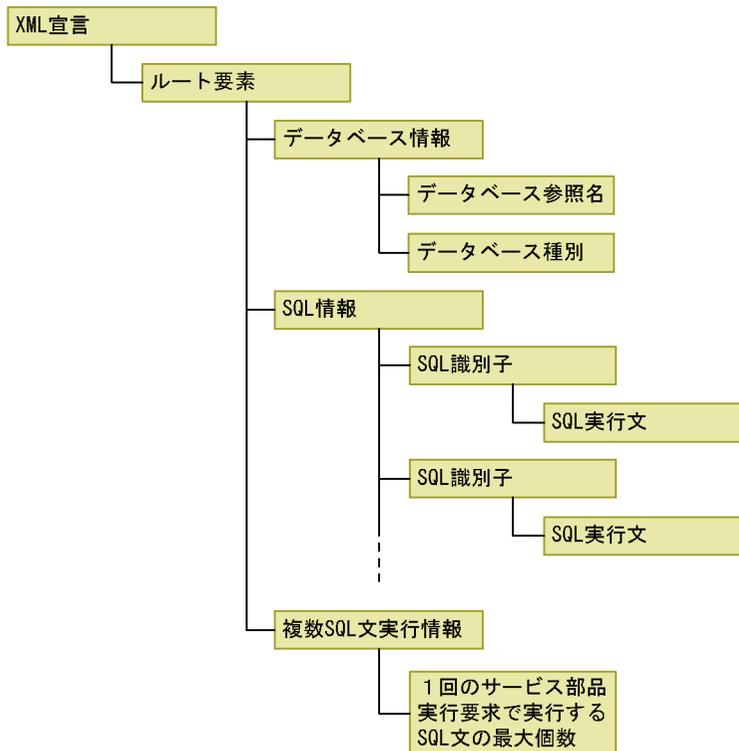
SQL オペレーション定義ファイルのファイル名は、次の形式で指定します。

```
csa_sql_<任意の名称>.xml
```

(b) 構造

SQL オペレーション定義ファイルの構造を次に示します。

図 4-4 SQL オペレーション定義ファイルの構造



(c) SQL オペレーション定義ファイルの形式

SQL オペレーション定義ファイルの形式を次に示します。なお、斜体の文字は可変値を表します。

#### 4. 電文フォーマットの作成

```
<?xml version="XMLのバージョン" encoding="文字コード"?>
<DBadapter_SQL_OPERATION>
  <DATABASE_DATA>
    <DB_NAME>データベース参照名</DB_NAME>
    <DB_TYPE>データベース種別</DB_TYPE>
  </DATABASE_DATA>
  <SQL_DATA>
    <SQL識別子 out_maxOccurs="検索結果出力最大件数">
      SQL命令 <引数要素名 dba_inf="引数の種別" data_type="データ型"/>
    </SQL識別子>
    :
  </SQL_DATA>
  <MULTI_SQL_DATA>
    <MAX_SQL_NO>1回のサービス部品実行要求で実行するSQL文の最大個数</MAX_SQL_NO>
  </MULTI_SQL_DATA>
</DBadapter_SQL_OPERATION>
```

SQL オペレーション定義ファイルで定義する内容について次に説明します。

##### XML 宣言

XML のバージョンには、XML の言語仕様に従って XML のバージョンを指定します。文字コードには UTF-8 を指定してください。なお、文字コードの指定は省略できます。省略した場合の値は「UTF-8 または UTF-16」になります。

```
<?xml version="XMLのバージョン" encoding="文字コード"?>
```

##### ルート要素 (要素名 : DBadapter\_SQL\_OPERATION)

SQL オペレーション定義ファイルのルート要素です。

```
<DBadapter_SQL_OPERATION>
:
</DBadapter_SQL_OPERATION>
```

##### データベース情報 (要素名 : DATABASE\_DATA)

下位の要素 (DB\_NAME および DB\_TYPE) で、データベースの情報を指定します。

```
<DATABASE_DATA>
  データベース参照名
  データベース種別
</DATABASE_DATA>
```

##### データベース参照名 (要素名 : DB\_NAME)

DB Connector のデータベース参照名を指定します。

```
<DB_NAME>データベース参照名</DB_NAME>
```

## データベース種別 (要素名: DB\_TYPE)

データベースの種別として次のどれかを指定します。

- HIRDB  
DBMS として HiRDB を使用する場合 (DB Connector として DBConnector\_DABJ\_CP/XA.rar を使用する場合) に指定します。
- HIRDB-TYPE4  
DBMS として HiRDB を使用する場合 (DB Connector として DBConnector\_HiRDB\_Type4\_CP/XA.rar を使用する場合) に指定します。
- ORACLE  
DBMS として Oracle を使用する場合 (DB Connector として DBConnector\_DABJ\_CP/XA.rar を使用する場合) に指定します。
- ORACLE-THIN  
DBMS として Oracle を使用する場合 (DB Connector として DBConnector\_Oracle\_CP/XA.rar を使用する場合) に指定します。

```
<DB_TYPE>データベース種別</DB_TYPE>
```

## SQL 情報 (要素名: SQL\_DATA)

下位の要素 (SQL 識別子および SQL 実行文) で、実行する SQL の情報を指定します。

```
<SQL_DATA>
  SQL識別子
  SQL実行文
</SQL_DATA>
```

## SQL 識別子 (要素名: 任意 ((1 ~ 256 文字)))

実行する SQL を識別するための要素です。下位の引数要素で SQL 実行文を指定します。属性として out\_maxOccurs を指定できます。また、実行する SQL 命令の数にあわせて複数回指定できます。

なお、SQL 識別子の名称として「DBadapter」および「DBA\_MULTI\_SQL」は指定できません。

```
<SQL識別子 out_maxOccurs="検索結果出力最大件数">
  SQL実行文
</SQL識別子>
:
```

## out\_maxOccurs

検索結果の最大出力件数を指定します。省略した場合は 1000 が設定されます。0 を指定した場合は、2147483647 が設定されます。

#### 4. 電文フォーマットの作成

##### SQL 実行文

SQL 実行文を指定します。SQL の構文は、利用するデータベースの仕様に従います。SQL 実行文の表や列の名称およびデータなどの可変部分（引数要素）を指定します。

なお、SQL 文の終わりを示すセミコロン（;）は必要ありません。

```
SQL命令 <引数要素名 dba_inf="引数の種別" data_type="データ型"/>
```

なお、SQL 実行文の先頭は「SELECT」または「INSERT」のどちらかを指定します（は半角スペース）。ただし、「SELECT」または「INSERT」の直後に引数要素がある場合、「」（半角スペース）は省略できます。指定内容を次に示します。

##### SQL 命令

指定できる SQL 命令は SELECT または INSERT です。

SELECT で検索できるデータ型について、次の表に示します。

表 4-1 SELECT で検索できるデータ型（HiRDB の場合）

HiRDB のデータ型	検索の可否	
	DBPSV の場合	HiRDB Type4 JDBC Driver の場合
INT [ EGER ]		
SMALLINT		
[ LARGE ] DEC [ IMAL ] NUMERIC		
FLOAT DOUBLE PRECISION		
SMALLFLT REAL		
CHAR [ ACTER ]		
VARCHAR CHAR [ ACTER ] VARYING		
NCHAR NATIONAL CHAR		
NVARCHAR NATIONAL CHAR [ ACTER ] VARYING NCHAR VARYING		
MCHAR		
MVARCHAR		
DATE		

HiRDB のデータ型	検索の可否	
	DBPSV の場合	HiRDB Type4 JDBC Driver の場合
TIME		
TIMESTAMP		
INTERVAL YEAR TO DAY	-	-
INTERVAL HOUR TO SECOND	-	-
BLOB BINARY LARGE OBJECT	-	-
BINARY	-	-
BOOLEAN	-	-

( 凡例 )

: SELECT で検索できます。

- : SELECT で検索できません。

注

[ ] で囲まれた部分は省略できることを意味します。

注

DBPSV とは、Cosminexus DABroker Library の JdbcDbpsvDriver です。

表 4-2 SELECT で検索できるデータ型 ( Oracle の場合 )

Oracle のデータ型	検索の可否	
	DBPSV <sup>1</sup> の場合	Oracle JDBC Thin Driver の場合
VARCHAR2		
NVARCHAR2		
NUMBER		
LONG		
DATE		
BINARY_FLOAT <sup>2</sup>	-	-
BINARY_DOUBLE <sup>2</sup>	-	-
TIMESTAMP	-	-
TIMESTAMP WITH TIME ZONE	-	-
TIMESTAMP WITH LOCAL TIME ZONE	-	-
INTERVAL YEAR TO MONTH	-	-
INTERVAL DAY TO SECOND	-	-
RAW	-	-

#### 4. 電文フォーマットの作成

Oracle のデータ型	検索の可否	
	DBPSV <sup>1</sup> の場合	Oracle JDBC Thin Driver の場合
LONG RAW	-	-
ROWID		
UROWID	- <sup>3</sup>	-
CHAR		
NCHAR		
CLOB		
NCLOB		
BLOB	-	-
BFILE	-	-
ユーザ定義型 (オブジェクト型)	-	-
ユーザ定義型 (REF データ型)	-	-
ユーザ定義型 (VARRAY)	-	-
ユーザ定義型 (ネストした表)	-	-

(凡例)

: SELECT で検索できます。

- : SELECT で検索できません。

注 1

DBPSV とは、Cosminexus DABroker Library の JdbcDbpsvDriver です。

注 2

接続するデータベースが Oracle10g の場合だけ使用できます。

注 3

SELECT で検索してもエラーにはなりませんが、検索した場合の動作は保証されません。

#### ポイント

DB アダプタで検索できないデータ型でも、データベースが提供する関数 (ファンクション) を使用して検索できるデータ型に変換できれば、検索できるようになります。

Oracle の TIMESTAMP 型 (列名: c\_ts) を TO\_CHAR ファンクションで VARCHAR2 型に変換して検索する場合の例を次に示します。

```
SELECT TO_CHAR(c_ts) FROM 表名
```

#### 引数要素名

任意の引数要素の名称を指定します。複数の引数要素を同一名称で指定した場合、同じ引数として扱われます。同一名称の引数要素で、それぞれの属性

dba\_inf および data\_type に異なる値を設定した場合はエラーになります。

dba\_inf = " 引数の種別 "

引数の種別を指定します。

引数の種別の種類として指定できる内容を次に示します。

- table  
引数がデータベースに実在するテーブルの名称の場合に指定します。引数として利用するデータは string 型です。
- column  
引数がデータベースに実在する列の名称の場合に指定します。引数として利用するデータは string 型です。
- data  
引数がテーブルおよび列の名称以外のデータの場合に指定します。この値を指定した場合、引数として利用するデータのデータ型を属性 data\_type に指定する必要があります。また、引数として利用するデータを表 4-1 および表 4-2 に示す java のデータ型またはクラスに変換して、SQL 文に埋め込みます。引数として利用するデータを変換できない場合は、エラーとなります。なお、この値を指定した場合、引数要素を IN パラメタ (? パラメタまたはバインド変数) として扱います。SQL 文中に IN パラメタを指定できる個所は、利用するデータベースの仕様に従います。指定できない個所に IN パラメタを定義した場合、エラーになります。なお、SQL の実行時に、引数要素は「?」に置き換えられて実行されます。
- preset  
引数がテーブルおよび列の名称以外のデータの場合に指定します。この値を指定した場合、引数として利用するデータをそのまま SQL 文に埋め込みます。引数として利用するデータは string 型です。

data\_type=" データ型 "

属性 dba\_inf に data を指定した場合に、引数として利用するデータの型を指定します。データベースのデータ型と data\_type に指定できるデータの型の対応を次の表に示します。

表 4-3 data\_type に指定できるデータ型 (HiRDB の場合)

HiRDB のデータ型	data_type に指定できるデータ型	引数として利用するデータの変換先のデータ型およびクラス
INT { EGER }	INTEGER	int
SMALLINT	SMALLINT	short
{ LARGE } DEC { IMAL } NUMERIC	DECIMAL	java.math.BigDecimal
FLOAT DOUBLE PRECISION	FLOAT	double

#### 4. 電文フォーマットの作成

HiRDB のデータ型	data_type に指定できるデータ型	引数として利用するデータの変換先のデータ型およびクラス
SMALLFLT REAL	REAL	float
CHAR (ACTER)	CHAR	java.lang.String
NCHAR NATIONAL CHAR		
MCHAR		
VARCHAR CHAR (ACTER) VARYING	VARCHAR	java.lang.String
NVARCHAR NATIONAL CHAR (ACTER) VARYING NCHAR VARYING		
MVARCHAR		
DATE		
TIME	TIME	java.sql.Time
TIMESTAMP	TIMESTAMP	java.sql.Timestamp

注 ( ) で囲まれた部分は省略できることを意味します。

表 4-4 data\_type に指定できるデータ型 ( Oracle の場合 )

Oracle のデータ型	data_type に指定できるデータ型		引数として利用するデータの変換先のデータ型およびクラス
	DBPSV <sup>1</sup> の場合	Oracle JDBC Thin Driver の場合	
LONG	LONGVARCHAR	LONGVARCHAR	java.io.Reader
NUMBER	NUMERIC	NUMERIC	java.math.BigDecimal
VARCHAR2	VARCHAR	VARCHAR	java.lang.String
NVARCHAR2	VARCHAR	-	
ROWID	CHAR	CHAR	
CHAR	CHAR	CHAR	
NCHAR	CHAR	-	
CLOB	CLOB	CLOB <sup>2</sup>	<ul style="list-style-type: none"> <li>• DBPSV <sup>1</sup> の場合 java.sql.Clob</li> <li>• Oracle JDBC Thin Driver の場合 java.io.Reader</li> </ul>
NCLOB	CLOB	-	
DATE	DATE <sup>3</sup>	DATE <sup>3</sup>	java.sql.Date

Oracle のデータ型	data_type に指定できるデータ型		引数として利用するデータの変換先のデータ型およびクラス
	DBPSV <sup>1</sup> の場合	Oracle JDBC Thin Driver の場合	
	TIME <sup>4</sup>	TIME <sup>4</sup>	java.sql.Time
	TIMESTAMP <sup>5</sup>	TIMESTAMP <sup>5</sup>	java.sql.Timestamp

(凡例)

- : 指定できません。

注 1

DBPSV とは、Cosminexus DABroker Library の JdbcDbpsvDriver です。

注 2

JDBC ドライバのバージョンが 10.2 の場合にだけ指定できます。10.2 以外のバージョンの JDBC ドライバを使用した場合、データが破損するおそれがあります。

注 3

設定する値の形式が「yyyy-mm-dd」の場合に指定します。

注 4

設定する値の形式が「hh:mm:ss」の場合に指定します。

注 5

設定する値の形式が「yyyy-mm-dd hh:mm:ss」の場合に指定します。

複数 SQL 文実行情報 (要素名: MULTI\_SQL\_DATA)

サービスリクエストからの 1 回のサービス部品実行要求で複数の SQL 文を実行する場合に、実行する SQL 文の最大個数を、下位の要素 (MAX\_SQL\_NO) で指定します。単一の SQL 文を実行する場合には、この要素は省略できます。

```
<MULTI_SQL_DATA>
  実行するSQL文の最大個数
</MULTI_SQL_DATA>
```

実行する SQL 文の最大個数 (要素名: MAX\_SQL\_NO)

実行する SQL 文の最大個数 (サービスリクエストからの要求電文の DBA\_MULTI\_SQL 要素の最大出現回数) を、1 ~ 65535 の範囲内で指定します。省略した場合の値は「1024」になります。

```
<MAX_SQL_NO>1回のサービス部品実行要求で実行するSQL文の最大個数</MAX_SQL_NO>
```

(d) 作成例

SQL オペレーション定義ファイルの作成例を次に示します。

#### 4. 電文フォーマットの作成

##### 作成例の条件

データベース参照名：DB\_SERVER1

スキーマ名：DBA

テーブル名：受注表

テーブル構成：

受注番号 (INTEGER)	得意先コード (CHAR)	商品コード (CHAR)	受注個数 (INTEGER)
1	AA001	0001	5
2	AB002	0001	1
3	AA001	0102	3
4	XA005	0103	1
5	AA001	0105	1

実行する SQL の形式：

```
OPERATION1 : SELECT * FROM DBA.受注表 WHERE <val1> <val2> <val3>
OPERATION2 : SELECT <val1> SUM(<val2>) FROM DBA.受注表 GROUP BY <val3>
OPERATION3 : INSERT INTO DBA.受注表 VALUES (<val1>,<val2>,<val3>,<val4>)
```

##### SQL オペレーション定義ファイルの作成例 (単一の SQL 文を実行する場合)

```
<?xml version="1.0" encoding="UTF-8" ?>
<DBadapter_SQL_OPERATION>
  <DATABASE_DATA>
    <DB_NAME>DB_SERVER1</DB_NAME>
    <DB_TYPE>HIRDB</DB_TYPE>
  </DATABASE_DATA>
  <SQL_DATA>
    <OPERATION1>
      SELECT * FROM DBA.受注表 WHERE <val1 dba_inf="column"/>
        <val2 dba_inf="preset"/>
        <val3 dba_inf="data" data_type="CHAR"/>
    </OPERATION1>
    <OPERATION2>
      SELECT <val1 dba_inf="column"/> SUM(<val2 dba_inf="column"/>)
        FROM DBA.受注表 GROUP BY <val3 dba_inf="column"/>
    </OPERATION2>
    <OPERATION3>
      INSERT INTO DBA.受注表
        VALUES (<val1 dba_inf="data" data_type="INTEGER" />,<br>
          <val2 dba_inf="data" data_type="CHAR"/>,<br>
          <val3 dba_inf="data" data_type="CHAR"/>,<br>
          <val4 dba_inf="data" data_type="INTEGER"/>)
    </OPERATION3>
  </SQL_DATA>
</DBadapter_SQL_OPERATION>
```

SQL オペレーション定義ファイルの作成例（複数の SQL 文を実行する場合）

次の例では、1 回のサービス部品実行要求で実行する SQL 文の最大個数を「10」にしています。

```
<?xml version="1.0" encoding="UTF-8" ?>
<DBadapter_SQL_OPERATION>
  <DATABASE_DATA>
    <DB_NAME>DB_SERVER1</DB_NAME>
    <DB_TYPE>HIRDB</DB_TYPE>
  </DATABASE_DATA>
  <SQL_DATA>
    <OPERATION1>
      SELECT * FROM DBA.受注表 WHERE <val1 dba_inf="column"/>
        <val2 dba_inf="preset"/>
        <val3 dba_inf="data" data_type="CHAR"/>
    </OPERATION1>
    <OPERATION2>
      SELECT <val1 dba_inf="column"/> SUM(<val2 dba_inf="column"/>)
        FROM DBA.受注表 GROUP BY <val3 dba_inf="column"/>
    </OPERATION2>
    <OPERATION3>
      INSERT INTO DBA.受注表
        VALUES(<val1 dba_inf="data" data_type="INTEGER" />,
          <val2 dba_inf="data" data_type="CHAR"/>,
          <val3 dba_inf="data" data_type="CHAR"/>,
          <val4 dba_inf="data" data_type="INTEGER"/>)
    </OPERATION3>
  </SQL_DATA>
  <MULTI_SQL_DATA>
    <MAX_SQL_NO>10</MAX_SQL_NO>
  </MULTI_SQL_DATA>
</DBadapter_SQL_OPERATION>
```

## (2) DB アダプタ用の XML フォーマット定義ファイルを作成する

DB アダプタ用の XML フォーマット定義ファイルを作成するには、あらかじめ作成した SQL オペレーション定義ファイルの名称を -o オプションに指定して csamkxmls コマンドを実行します。

csamkxmls コマンドの詳細については、「12. コマンド」の「csamkxmls (DB アダプタ用の XML フォーマット定義ファイルの生成)」を参照してください。

#### 4. 電文フォーマットの作成

##### ! 注意事項

csamkxmls コマンドを実行したとき、SQL オペレーション定義ファイルのファイルサイズが大きいと、「java.lang.OutOfMemoryError」のメッセージが表示されることがあります。このエラーが表示された場合、環境変数 HCSCDBA\_XMX に、現在の JavaVM のメモリ割り当てサイズよりも大きい値を設定して、再度コマンドを実行してください。環境変数 HCSCDBA\_XMX には、JavaVM のヒープサイズ (-Xmx) に設定される値（単位：Mbyte）を指定します。省略した場合、JavaVM のヒープサイズは Java コマンドを実行する環境に依存します。

設定例

```
set HCSCDBA_XMX=64
```

csamkxmls コマンドの実行形式を次に示します。

```
csamkxmls -o <SQLオペレーション定義ファイルの名称> -x <DBアダプタ用のXMLフォーマット定義ファイルの名称> -f
```

-f オプションは、同一名称の DB アダプタ用の XML フォーマット定義ファイルがある場合を上書きするオプションです。上書きしない場合には省略できます。

##### ! 注意事項

DB アダプタ用の XML フォーマット定義ファイルは、SQL オペレーション定義ファイルで指定した SQL 識別子の数だけフォーマットを含みます。単一の SQL 文を実行する場合、データ変換を定義するときには、SQL 識別子ごとにオペレーションを変更（サービスアダプタ定義画面でオペレーションを追加）して定義する必要があります。複数の SQL 文を実行する場合、データ変換を定義するときには、SQL 識別子ごとにオペレーションを変更して定義する必要はありません。

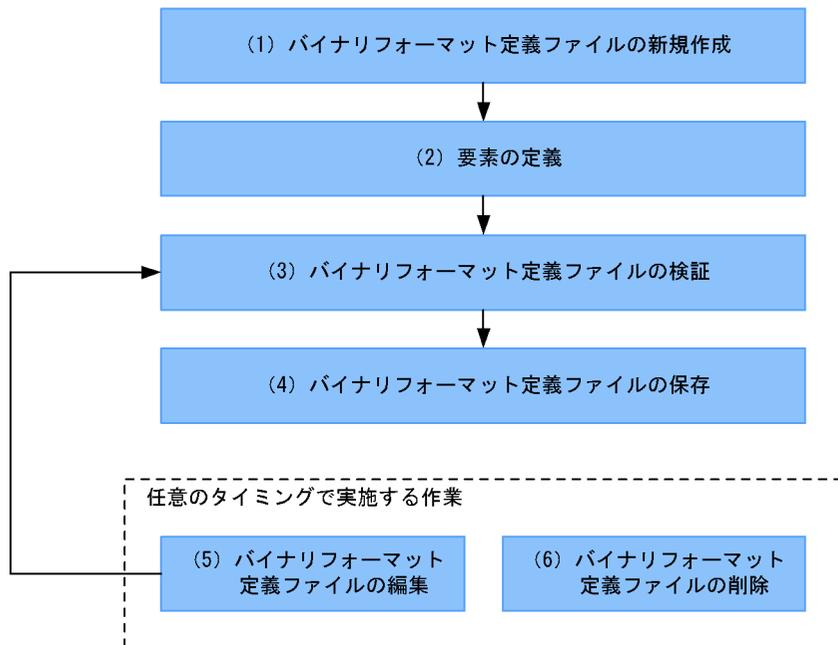
データ変換の定義については、「7. データ変換の定義」を、サービスアダプタ定義画面については、「11.2.2 サービスアダプタ定義画面」を参照してください。

## 4.4 電文フォーマット（バイナリフォーマット定義ファイル）の作成方法

サービス部品の実行に利用する電文でバイナリ形式（XML形式以外）のデータを利用する場合、バイナリフォーマット定義ファイルを作成します。

バイナリフォーマット定義ファイルの作成の流れを次の図に示します。

図 4-5 バイナリフォーマット定義ファイルの作成の流れ



### (1) バイナリフォーマット定義ファイルの新規作成

ウィザードを利用してバイナリフォーマット定義ファイルを新規作成します。バイナリフォーマット定義ファイルの新規作成の方法については、「4.4.1 バイナリフォーマット定義ファイルを新規作成する」を参照してください。

### (2) 要素の定義

新規作成したバイナリフォーマット定義ファイルに要素を定義します。要素の定義の方法は、利用するバイナリデータが CSV 以外の形式の場合と、CSV 形式の場合とで異なります。

要素の定義方法については、「4.4.2 要素を定義する（CSV 以外の形式の場合）」または「4.4.3 要素を定義する（CSV 形式の場合）」を参照してください。

### (3) バイナリフォーマット定義ファイルの検証

作成したバイナリフォーマット定義ファイルの整合性を検証します。バイナリ

#### 4. 電文フォーマットの作成

フォーマット定義ファイルの要素の定義中や定義終了後など、任意のタイミングで検証できます。

バイナリフォーマット定義ファイルの検証方法については、「4.4.5 バイナリフォーマット定義ファイルを検証する」を参照してください。

##### (4) バイナリフォーマット定義ファイルの保存

要素を定義したバイナリフォーマット定義ファイルを保存します。バイナリフォーマット定義ファイルは、アダプタやビジネスプロセスと異なり、リポジトリに保存されません。定義したバイナリフォーマット定義ファイルの保存には、Eclipse のパッケージ・エクスプローラーを利用してください。

バイナリフォーマット定義ファイルを保存するときには、自動的に検証が行われます（この場合、検証結果はコンソールビューに表示されません）。

##### (5) バイナリフォーマット定義ファイルの編集

すでに作成したバイナリフォーマット定義ファイルは、必要に応じて、定義内容を変更できます。バイナリフォーマット定義ファイルの編集方法については、「4.4.4 バイナリフォーマット定義ファイルを編集する」を参照してください。

##### (6) バイナリフォーマット定義ファイルの削除

不要になったバイナリフォーマット定義ファイルは、必要に応じて削除できます。バイナリフォーマット定義ファイルの削除には、Eclipse のパッケージ・エクスプローラーを利用してください。

### 4.4.1 バイナリフォーマット定義ファイルを新規作成する

バイナリフォーマット定義ファイルを新規作成する方法を次に示します。

なお、バイナリフォーマット定義ファイルを作成する前には、プロジェクトを作成しておく必要があります。

1. Eclipse のメニューから、[ファイル] - [新規] - [その他] を選択します。  
ウィザードを選択するダイアログが表示されます。
2. [HCSCTE フォーマット定義] の [バイナリフォーマット定義ファイル] を選択して、[次へ] ボタンをクリックします。  
[新規バイナリフォーマット定義ファイル] ウィザードが表示されます。
3. [親フォルダーを入力または選択] で、バイナリフォーマット定義ファイルを保存するフォルダを入力または選択します。
4. 保存するバイナリフォーマット定義ファイルのフォーマット名を [フォーマット名] に指定します。  
フォーマット名を指定すると、指定したフォーマット名に拡張子「.fdx」を付けた名称が [ファイル名] に自動設定されます。
5. 利用するバイナリデータが CSV 形式の場合、[CSV 形式] チェックボックスをチェックします。

なお、CSV 形式のデータとして扱えるのは、次の区切り文字を使用しているデータです。

- 中間区切り文字：コンマ(,)
- 終了文字：改行

要素の値に中間区切り文字と同じ文字を含む場合は、その要素全体をダブルクォーテーション(")で囲みます。なお、要素内でダブルクォーテーションを値として使用する場合、ダブルクォーテーションを二つ続けてエスケープしてください。

なお、開始文字を利用している場合は、CSV 形式のデータとして扱えないので注意してください。

#### 6. [終了] ボタンをクリックします。

新しいバイナリフォーマット定義ファイルが作成され、バイナリフォーマット定義画面が表示されます。バイナリフォーマット定義画面で、続けて要素を定義できます。

#### ! 注意事項

バイナリフォーマット定義ファイルは、アダプタやビジネスプロセスと異なり、リポジトリに保存されません。定義したバイナリフォーマット定義ファイルの保存や削除などのファイル管理は、Eclipse のパッケージ・エクスプローラーを利用してください。

## 4.4.2 要素を定義する (CSV 以外の形式の場合)

バイナリフォーマット定義ファイルの内容はバイナリフォーマット定義画面で定義します。バイナリフォーマット定義画面は、Eclipse のパッケージ・エクスプローラーに表示されるバイナリフォーマット定義ファイルをダブルクリックすると表示されます。

バイナリフォーマット定義画面については、「11.2.1 バイナリフォーマット定義画面」を参照してください。

#### 参考

バイナリフォーマット定義画面では、複数のバイナリフォーマット定義ファイルを開いて編集できます。ただし、複数のバイナリフォーマット定義ファイルの間で、定義情報をコピーして貼り付けることはできません。

バイナリフォーマット定義画面では、要素の定義として次の作業を行います。

- フォーマット情報の設定
- グローバル定義の単純内容要素の設定
- 複合内容要素の設定
- 構成要素の設定
- 構成要素の選択条件の設定
- ルート要素の設定
- セパレータの設定

#### 4. 電文フォーマットの作成

定義の方法を次に示します。

##### (1) フォーマット情報を設定する

バイナリフォーマット定義ファイルのフォーマット情報として、バイナリデータで使用されている文字コードや、バイナリデータに含まれる2進整数のエンディアンを設定します。

フォーマット情報の設定方法を次に示します。

1. バイナリフォーマット定義エディタに表示されているフォーマット名を選択して右クリックし、[設定]を選択します。  
[フォーマット]ダイアログが表示されます。
2. [フォーマット]ダイアログに必要な情報を入力します。  
[フォーマット]ダイアログの表示・入力内容については、「11.3.1 フォーマットダイアログ」を参照してください。
3. [OK]ボタンをクリックします。  
[フォーマット]ダイアログに入力した情報が設定されます。

##### (2) グローバル定義の単純内容要素を設定する

単純内容要素とは、使用するバイナリデータの形式をXMLスキーマで表現したときに、単純型の要素(xsd:simpleType要素)であり、かつグローバル定義の要素として表現できるものです。バイナリフォーマット定義ファイルをデータ変換(マッピング)定義画面で表示した場合、単純内容要素は、単純内容のノードとして表示されます。

単純内容要素の設定では、要素のデータ種別やサイズを設定します。

単純内容要素の設定方法を次に示します。

1. バイナリフォーマット定義エディタに表示されているフォーマット名を選択して右クリックし、[単純内容要素の追加]を選択します。  
[単純内容要素]ダイアログが表示されます。
2. [単純内容要素]ダイアログに必要な情報を入力します。  
[単純内容要素]ダイアログの表示・入力内容については、「11.3.2 単純内容要素ダイアログ」を参照してください。
3. [OK]ボタンをクリックします。  
[単純内容要素]ダイアログに入力した情報で単純内容要素が設定されます。また、設定された単純内容要素がバイナリフォーマット定義エディタに表示されます。

##### (3) 複合内容要素を設定する

複合内容要素とは、使用するバイナリデータの形式をXMLスキーマで表現したときに、複合型の要素(xsd:complexType要素)であり、かつグローバル定義の要素として表現できるものです。バイナリフォーマット定義ファイルをデータ変換(マッピング)定義画面で表示した場合、複合内容のノードとして表示されます。

複合内容要素の設定では、複合内容要素を構成する要素の構成区分などを設定します。構成区分には、複合内容要素を構成する要素が逐次構成（xsd:sequence 要素）か、選択構成（xsd:choice 要素）かを設定します。

複合内容要素の設定方法を次に示します。

1. バイナリフォーマット定義エディタに表示されているフォーマット名を選択して右クリックし、[ 複合内容要素の追加 ] を選択します。  
[ 複合内容要素 ] ダイアログが表示されます。
2. [ 複合内容要素 ] ダイアログに必要な情報を入力します。  
[ 複合内容要素 ] ダイアログの表示・入力内容については、「11.3.3 複合内容要素ダイアログ」を参照してください。

構成区分として [ 選択構成 ] を選択した場合

構成区分として [ 選択構成 ] を選択した場合、次のどちらかの方法で構成要素に選択条件を設定する必要があります。

- ・ [ 選択条件ノード ] を指定し、選択条件の判定値を含むノードを選択する

- ・ [ 構成要素の開始セパレータ ] を指定する

選択条件を設定する方法については、「4.4.2(5) 構成要素の選択条件を設定する」を参照してください。

なお、構成要素に選択条件を設定する前に、複合内容要素を構成する要素を設定する必要があります。必ず次の手順 3. を行ってから、「4.4.2(4) 構成要素を設定する」に示す手順に従って構成要素を設定し、そのあとで選択条件を設定してください。

複合内容要素の構成要素である単純内容要素がセパレータデータの場合

複合内容要素の構成要素である単純内容要素が、要素ごとに区切り文字（セパレータ）で区切られたデータ（セパレータデータ）の場合、複合内容要素にセパレータを設定します。

セパレータの設定方法については、「4.4.2(7) セパレータを設定する」を参照してください。

3. [ OK ] ボタンをクリックします。  
[ 複合内容要素 ] ダイアログに入力した情報で複合内容要素が設定されます。また、設定された複合内容要素がバイナリフォーマット定義エディタに表示されます。

#### 参考

---

バイナリフォーマット定義ファイルの要素の定義時に、最初に設定した複合内容要素は、ルート要素になります。ルート要素を変更する場合には、「4.4.2(6) ルート要素を設定する」を参照してください。

---

## (4) 構成要素を設定する

構成要素とは、複合内容要素を構成する要素です。構成要素を設定する方法には次の二つの方法があります。

#### 4. 電文フォーマットの作成

- グローバル定義の要素を構成要素に設定する
- ローカル定義の単純内容要素を設定する

それぞれの方法での構成要素の設定について次に示します。

##### (a) グローバル定義の要素を構成要素に設定する

すでに設定されているグローバルな要素（単純内容要素および複合内容要素）を、複合内容要素の構成要素に設定します。

設定方法を次に示します。

1. バイナリフォーマット定義エディタに表示されている複合内容要素を選択して右クリックし、[ 構成要素の追加 ] を選択します。  
[ 要素選択 ] ダイアログが表示されます。[ 要素選択 ] ダイアログには、複合内容要素の構成要素として選択できるグローバル定義の要素がすべて表示されます。  
[ 要素選択 ] ダイアログの表示・入力内容については、「11.3.7 要素選択ダイアログ」を参照してください。
2. [ 要素選択 ] ダイアログの要素一覧で構成要素として設定する要素を選択します。
3. [ OK ] ボタンをクリックします。  
[ 構成要素 ] ダイアログが表示されます。
4. [ 構成要素 ] ダイアログに必要な情報を入力します。  
[ 構成要素 ] ダイアログの表示・入力内容については、「11.3.4 構成要素ダイアログ」を参照してください。
5. [ OK ] ボタンをクリックします。  
[ 構成要素 ] ダイアログに入力した情報で、グローバル定義の要素が複合内容要素の構成要素として設定されます。設定した構成要素は、バイナリフォーマット定義エディタ上の複合内容要素の下位要素として表示されます。

##### (b) ローカル定義の単純内容要素を構成要素として設定する

ローカル定義の単純内容要素を複合内容要素の構成要素として設定します。

設定方法を次に示します。

1. バイナリフォーマット定義エディタに表示されている複合内容要素を選択して右クリックし、[ ローカル単純内容要素の追加 ] を選択します。  
[ 単純内容要素 ] ダイアログが表示されます。
2. [ 単純内容要素 ] タブを選択し、必要な情報を入力します。  
[ 単純内容要素 ] ダイアログの [ 単純内容要素 ] タブの表示・入力内容については、「11.3.2 単純内容要素ダイアログ」を参照してください。
3. [ 構成要素 ] タブを選択し、必要な情報を入力します。  
[ 構成要素 ] タブでは、構成要素の出現回数およびサイズを設定します。  
[ 単純内容要素 ] ダイアログの [ 単純内容要素 ] タブの表示・入力内容については、「11.3.2 単純内容要素ダイアログ」を参照してください。

なお、出現回数およびサイズは、ほかの要素に依存させることができます。この場合、[ノード選択] ボタンをクリックして表示される [ノード選択] ダイアログで、依存させるノードを選択できます。

[ノード選択] ダイアログの表示・入力内容については、「11.3.6 ノード選択ダイアログ (バイナリフォーマット定義)」を参照してください。

[ノード選択] ダイアログで [OK] ボタンをクリックすると、設定した値が [単純内容要素] ダイアログの [構成要素] タブに反映されます。

#### 4. [OK] ボタンをクリックします。

[単純内容要素] ダイアログに入力した情報で、ローカル定義の単純内容要素が複合内容要素の構成要素として設定されます。設定した構成要素は、バイナリフォーマット定義エディタ上の複合内容要素の下位要素として表示されます。

### (5) 構成要素の選択条件を設定する

複合内容要素を設定する際、構成区分として [選択構成] を選択した場合、構成要素に選択条件を設定する必要があります。選択条件を設定すると、複合内容要素に含まれる複数の構成要素のうち、どの要素を利用するかを別の要素の値から判定できます。

選択条件を設定する方法には、次の二つの方法があります。

- [選択条件ノード] を選択して、選択条件を設定する  
選択条件ノードで選択したノードに設定した条件値から、利用する構成要素を決定します。
- [構成要素の開始セパレータ] を選択して、選択条件を設定する  
構成要素に設定した開始セパレータから、利用する構成要素を決定します。

それぞれの方法での選択条件の設定について次に示します。

#### (a) 選択条件ノードを選択して選択条件を設定する

選択条件ノードを選択して選択条件を設定する方法を次に示します。

1. 選択条件を設定する構成要素を含む複合内容要素をバイナリフォーマット定義エディタ上で選択して右クリックし、[設定] を選択します。  
[複合内容要素] ダイアログが表示されます。
2. [構成区分] で [選択構成] を選択します。
3. [選択種別] で [選択条件ノード] を選択し、[ノード選択] ボタンをクリックします。  
[ノード選択] ダイアログが表示されます。[ノード選択] ダイアログでは、選択条件ノードを選択します。  
[ノード選択] ダイアログの表示・入力内容については、「11.3.6 ノード選択ダイアログ (バイナリフォーマット定義)」を参照してください。  
[ノード選択] ダイアログで [OK] ボタンをクリックすると、設定した値が [複合内容要素] ダイアログの [選択条件ノード] に反映されます。
4. [構成要素一覧] から条件を設定する要素を選択し、[条件値設定] ボタンをクリック

#### 4. 電文フォーマットの作成

します。

[条件値設定] ダイアログが表示されます。[条件値設定] ダイアログでは、条件値を入力します。

[条件値設定] ダイアログの表示・入力内容については、「11.3.5 条件値設定ダイアログ」を参照してください。

[条件値設定] ダイアログで [OK] ボタンをクリックすると、設定した値が [複合内容要素] ダイアログの [選択条件値] に反映されます。ほかの構成要素にも選択条件を設定する場合、手順 3. を繰り返します。

なお、選択条件が設定されていない構成要素は、ほかの構成要素に設定されているとの選択条件にも当てはまらない場合に出現する構成要素になります。このような構成要素は、構成要素の中で一つだけ設定できます。

##### 5. [OK ボタン] をクリックします。

[複合内容要素] ダイアログに入力した情報で、選択条件が設定されます。

##### (b) 構成要素の開始セパレータを選択して選択条件を設定する

構成要素の開始セパレータを選択して選択条件を設定する方法を次に示します。

##### 1. 選択条件を設定する構成要素を含む複合内容要素をバイナリフォーマット定義エディタ上で選択して右クリックし、[設定] を選択します。

[複合内容要素] ダイアログが表示されます。

##### 2. [構成区分] で [選択構成] を選択します。

##### 3. [選択種別] で [構成要素の開始セパレータ] を選択します。

[構成要素一覧] に、構成要素および構成要素に設定した開始セパレータが表示されます。

##### 4. [OK ボタン] をクリックします。

[複合内容要素] ダイアログに入力した情報で、選択条件が設定されます。なお、構成要素（複合内容要素）に開始セパレータを設定していない場合は、バイナリフォーマット定義ファイル検証時にエラーになります。

##### (6) ルート要素を設定する

使用するバイナリデータを XML スキーマで表現したときに、ルート要素として扱う複合内容要素を設定します。

バイナリフォーマット定義ファイルの要素の定義時には、最初に設定した複合内容要素がルート要素になっています。ほかの複合内容要素をルート要素にするには、バイナリフォーマット定義エディタに表示されている複合内容要素を選択して右クリックし、[ルート要素に設定] を選択します。

[ルート要素に設定] を選択すると、バイナリフォーマット定義エディタ上で選択した複合内容要素がルート要素になります。バイナリフォーマット定義エディタ上では、ルート要素として扱う複合内容要素にルート要素のアイコンが表示されます。

なお、ある複合内容要素がすでにルート要素として設定されている場合、ほかの複合内

容要素をルート要素として設定すると、ルート要素が変更されます。アウトラインビューの表示も、新しいルート要素からの表示に切り替わります。

### (7) セパレータを設定する

複合内容要素の構成要素として設定されている単純内容要素が、要素ごとに区切り文字（セパレータ）で区切られたデータ（セパレータデータ）の場合、複合内容要素にセパレータを設定します。

セパレータを設定する方法を次に示します。

1. 複合内容要素を選択して右クリックし、[ 設定 ] を選択します。  
[ 複合内容要素 ] ダイアログが表示されます。
2. [ セパレータ ] タブを選択します。  
[ 複合内容要素 ] ダイアログの [ セパレータ ] タブの内容が表示されます。
3. 使用するセパレータの [ 使用する ] チェックボックスをチェックし、[ 選択 ] ボタンをクリックします。  
[ セパレータ選択 ] ダイアログが表示されます。[ セパレータ選択 ] ダイアログの表示・入力内容については「11.3.8 セパレータ選択ダイアログ」を参照してください。
4. 使用するセパレータを一覧から選択し、[ OK ] ボタンをクリックします。  
[ 複合内容要素 ] ダイアログのセパレータタブに選択したセパレータの情報が表示されます。  
なお、[ セパレータ選択 ] ダイアログに、使用するセパレータがない場合、[ セパレータ選択 ] ダイアログの [ 追加 ] ボタンをクリックします。表示される [ セパレータ追加 / 変更 ] ダイアログでセパレータを追加 / 変更できます。  
[ セパレータ追加 / 変更 ] ダイアログの表示・入力内容については、「11.3.9 セパレータ追加 / 変更ダイアログ」を参照してください。
5. 必要に応じて、手順 3. と手順 4. を繰り返します。
6. [ OK ] ボタンをクリックします。  
[ 複合内容要素 ] ダイアログの [ セパレータ ] タブに入力した情報で、セパレータが設定されます。

## 4.4.3 要素を定義する（CSV 形式の場合）

CSV 形式の場合、バイナリフォーマット定義ファイルの内容は、バイナリフォーマット定義画面で定義します。バイナリフォーマット定義画面は、Eclipse のパッケージ・エクスプローラーに表示されるバイナリフォーマット定義ファイルをダブルクリックすると表示されます。

バイナリフォーマット定義画面については、「11.2.1 バイナリフォーマット定義画面」を参照してください。

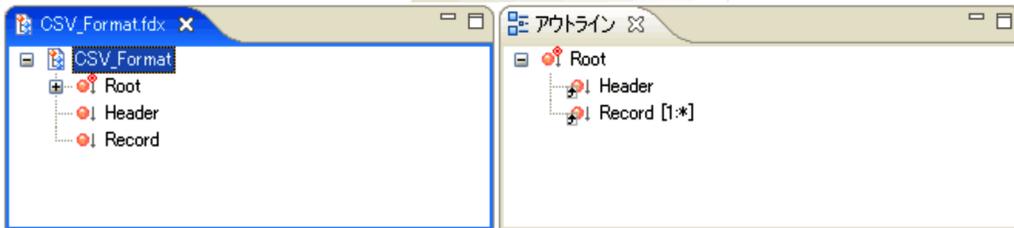
## 4. 電文フォーマットの作成

### 参考

バイナリフォーマット定義画面では、複数のバイナリフォーマット定義ファイルを開いて編集できます。ただし、複数のバイナリフォーマット定義ファイルの間で、定義情報をコピーして貼り付けることはできません。

CSV 形式の場合、バイナリフォーマット定義ファイルを新規作成して、バイナリフォーマット定義画面を表示すると、ルート要素、ヘッダー要素、およびレコード要素が配置された状態で表示されます。

図 4-6 CSV 形式のデータを利用する場合のバイナリフォーマット定義画面の初期表示



ルート要素、ヘッダー要素、およびレコード要素はすべて複合内容要素です。ヘッダー要素およびレコード要素はルート要素の構成要素として設定されています。ヘッダー要素およびレコード要素には、ローカル定義の単純内容要素を構成要素として設定します。

バイナリフォーマット定義画面では次の作業を行います。

- フォーマット情報の設定
- ヘッダー要素およびレコード要素の出現回数設定
- ヘッダー要素およびレコード要素の構成要素の設定
- ルート要素の設定

### ポイント

CSV 形式のデータを利用する場合、バイナリフォーマット定義画面に初期表示されるルート要素、ヘッダー要素、およびレコード要素の構成は、変更できません。

定義の方法を次に示します。

#### (1) フォーマット情報を設定する

バイナリフォーマット定義ファイルのフォーマット情報として、バイナリデータで使用されている文字コードや改行を指定します。

設定の方法は、CSV 以外の形式のバイナリデータを利用する場合と同じです。フォーマット情報の設定方法については、「4.4.2(1) フォーマット情報を設定する」を参照してください。

## (2) ヘッダー要素およびレコード要素の出現回数を設定する

ヘッダー要素およびレコード要素の出現回数の設定方法を次に示します。

1. ルート要素の構成要素として表示されているヘッダー要素またはレコード要素を選択して右クリックし、[ 設定 ] を選択します。  
[ 構成要素 ] ダイアログが表示されます。
2. [ 構成要素 ] ダイアログで出現回数を入力します。  
なお、ヘッダー要素の出現回数は、CSV データにヘッダーがない場合は「0」を、ヘッダーがある場合は「1」を設定します。  
[ 構成要素 ] ダイアログの表示・入力内容については、「11.3.4 構成要素ダイアログ」を参照してください。
3. [ OK ] ボタンをクリックします。  
[ 構成要素 ] ダイアログに入力した出現回数が設定されます。

## (3) ヘッダー要素およびレコード要素の構成要素を設定する

バイナリフォーマット定義エディタに、ルート要素の構成要素として表示されているヘッダー要素およびレコード要素の構成要素の設定方法を次に示します。

1. ルート要素の構成要素として表示されているヘッダー要素またはレコード要素を選択して右クリックし、[ ローカル単純内容要素の追加 ] を選択します。  
[ 単純内容要素 ] ダイアログが表示されます。
2. [ 単純内容要素 ] ダイアログに必要な情報を入力します。  
[ 単純内容要素 ] ダイアログの表示・入力内容については、「11.3.2 単純内容要素ダイアログ」を参照してください。
3. [ OK ] ボタンをクリックします。  
[ 単純内容要素 ] ダイアログに入力した情報が設定されます。

## (4) ルート要素を設定する

CSV データを利用する場合、バイナリフォーマット定義画面に初期表示されたルート要素以外の要素を、ルート要素に設定することはできません。ただし、ルート要素の名称だけ変更できます。

ルート要素の名称の変更方法を次に示します。

1. ルート要素を選択して右クリックし、[ 設定 ] を選択します。  
[ 複合内容要素 ] ダイアログが表示されます。
2. [ 名称 ] を入力します。
3. [ OK ] ボタンをクリックします。  
入力した名称が設定されます。

#### 4.4.4 バイナリフォーマット定義ファイルを編集する

すでに作成、保存されているバイナリフォーマット定義ファイルの内容を変更できます。

バイナリフォーマット定義ファイルの内容を変更するには、Eclipse のパッケージ・エクスプローラーで編集するバイナリフォーマット定義ファイルをダブルクリックします。表示されるバイナリフォーマット定義画面で内容を変更します。

バイナリフォーマット定義画面で設定できる内容については、「4.4.2 要素を定義する (CSV 以外の形式の場合)」, または「4.4.3 要素を定義する (CSV 形式の場合)」を参照してください。

なお、すでに作成済みのバイナリフォーマット定義ファイルをコピーして内容を変更し、別のバイナリフォーマット定義ファイルを作成することもできます。ただし、作成済みのバイナリフォーマット定義ファイルの定義内容の一部をコピーして流用することはできません。

#### 4.4.5 バイナリフォーマット定義ファイルを検証する

必要な情報が設定されていなかったり、設定の内容や関係が正しくない場合、バイナリフォーマット定義ファイルを利用したデータ変換定義の作成や、データ変換が正しく行われません。

そのため、作成したバイナリフォーマット定義ファイルは、設定の内容が妥当かどうか検証します。検証は、任意のタイミングで実施できます。また、各要素を設定するダイアログでの入力時には、入力された値が自動的に検証されます。

##### (1) 検証内容

バイナリフォーマット定義ファイルの各要素の設定内容が検証できます。

検証内容とエラー発生時の対処、およびどの検証内容がどの検証方法の場合に実施されるかについて、次の表に示します。

表 4-5 バイナリフォーマット定義ファイルの検証内容とエラー発生時の対処

検証の対象	検証内容	エラー発生時の対処	検証方法	
			自動	任意
全体	名称が NCName 型か。	フォーマット名称および要素の名称を NCName 型で指定します。		-
	逐次構成の複合内容要素が設定されているか。	逐次構成の複合内容要素を設定します。	-	
	ルート要素が設定されているか。	バイナリフォーマット定義内の複合内容要素をルート要素として設定します。	-	

検証の対象	検証内容	エラー発生時の対処	検証方法	
			自動	任意
単純内容要素	同じ名称の要素が同じ階層にないか。	同じ階層でユニークな名称を指定します。		-
	グローバル定義の単純内容要素は、複合内容要素の構成要素に設定されているか。	該当する単純内容要素を複合内容要素の構成要素に設定します。		-
	サイズが固定の場合、指定したサイズの値は次の式を満たすか。 $0 < \text{サイズ} \leq 2,147,483,647$	サイズを 1 ~ 2,147,483,647 で設定します。		-
	次の条件をすべて満たす場合、指定したサイズの値が偶数か。 <ul style="list-style-type: none"> <li>データ種別が文字列、整数、実数、固定小数部数値、または日付時刻</li> <li>サイズが固定</li> <li>文字コードが UTF-16 (BE/LE)</li> </ul>	文字コードに UTF-16 (BE/LE) 以外を指定するか、該当する単純内容要素のサイズに偶数を指定します。		
	次の条件をすべて満たす場合、指定した小数部桁数が、サイズに指定した値から求められる最大の小数部桁数以下になっているか。 <ul style="list-style-type: none"> <li>データ種別が固定小数部数値、ゾーン形式数値、またはパック形式数値</li> <li>サイズが固定</li> </ul>	サイズに小数部桁数のバイト数以上の値を設定します。		
	データ種別が固定小数部数値、ゾーン形式数値、またはパック形式数値の場合、指定した小数部桁数は次の式を満たすか。 $0 \leq \text{小数部桁数} \leq 33$	小数部桁数を 0 ~ 33 で設定します。		-
	データ種別が日付時刻で、秒小数部桁数が指定されている場合、指定した小数部桁数は次の式を満たすか。 $0 \leq \text{秒小数部桁数} \leq 3$	秒小数部桁数を 0 ~ 3 で設定します。		-
	データ種別が日付時刻で、サイズが固定の場合、指定したサイズは次の式を満たすか。 $\text{サイズ} = \text{形式サイズ} + \text{小数点サイズ} + \text{秒小数部桁数}$	サイズに日付時刻形式のバイト数以上の値を設定します。		
複合内容要素	同じ名称の要素が同じ階層にないか。	同じ階層でユニークな名称を指定します。		-
	ルート要素に設定されている複合内容要素が、ほかの複合内容要素の構成要素として設定されていないか。	複合内容要素の構成要素から、ルート要素に設定されている複合内容要素を削除します。	-	

#### 4. 電文フォーマットの作成

検証の対象	検証内容	エラー発生時の対処	検証方法	
			自動	任意
	ルート要素からたどれるノードの数は、1,024 以下か。	ルート要素からたどれるツリー上のノードを削除し、ノード数を 1,024 以下にします。	-	
	複合内容要素はほかの複合内容要素の構成要素に設定されているか(ただしルート要素の場合は除く)。	該当する複合内容要素を、ほかの複合内容要素の構成要素に設定します。	-	
	構成要素は設定されているか。	該当する複合内容要素に構成要素を設定します。	-	
	選択構成の場合、選択条件ノードが設定されているか。	該当する複合内容要素に選択条件ノードを設定します。	-	
	選択構成の場合、選択条件ノードに設定されている要素は単純内容要素か。	選択条件ノードとして単純内容要素のノードを設定します。		
	選択構成の場合、各構成要素に選択条件値が設定されているか(ただし、一つだけは選択条件値を設定しなくてもよい)。	各構成要素に選択条件値を設定し、選択条件値が設定されていない構成要素を一つ以下にします。	-	
	選択構成の場合、選択条件ノードに設定されている単純内容要素のデータ種別と、各構成要素の選択条件値に指定した値が一致するか。	選択条件ノードに設定されている単純内容要素のデータ種別によって検証の内容と対処が異なります。詳細については「表 4-6 選択条件値に指定した値に関する検証内容と対処」を参照してください。		
	選択構成の場合、各構成要素の選択条件値に重複はないか。	選択条件値が重複した構成要素に、ほかの構成要素で設定されていない選択条件値を設定します。	-	
	選択構成の場合、選択条件ノード、およびそのすべての祖先ノードの出現回数が固定で、かつ 1 回か。	選択条件ノードとその祖先ノードの出現回数を 1 回固定になるように設定します。	-	
	選択構成の場合、選択条件ノードのサイズが固定か。	選択条件ノードのサイズを固定に設定します。	-	
	選択構成の場合、選択条件ノードのパスが正しいか。	選択条件ノードを示すパスが不正であるおそれがあります。対処方法については「表 4-7 パスが不正である場合の原因と対処」を参照してください。		
	選択構成の場合、選択条件ノードのパスに選択構成の複合内容要素が含まれていないか(ただし、パスが絶対パスの場合、設定元のノードの祖先ノードは除く)。	選択条件ノードが実際のバイナリデータで出現しないおそれがあります。実際のバイナリデータで選択条件ノードに該当するデータが出現するか確認します。	-	

検証の対象	検証内容	エラー発生時の対処	検証方法	
			自動	任意
	選択構成の場合、コード変換ライブラリが使用できるか（クラスをロードできるか）。	uCosminexus 日立コード変換 - Development Kit の jar ファイルを正しく設定したあと、Eclipse を再起動します。		
	コード変換テーブルのパスが正しく設定されているか。	コード変換テーブルパスに日立コード変換の変換用テーブルのパスを正しく指定してください。		
	選択構成の場合、構成要素に設定されている単純内容要素の数は一つ以下か。	構成要素に設定する単純内容要素を一つ以下にします。	-	
	選択構成の場合、各構成要素（複合内容要素）に開始セパレータが設定されているか（ただし、構成要素に単純内容要素が設定されていない場合、一つだけは開始セパレータを設定しなくてもよい）。	構成要素に単純内容要素が設定されている場合、各構成要素に開始セパレータを設定します。構成要素に単純内容要素が設定されていない場合、開始セパレータが設定されていない構成要素を一つ以下にします。	-	
	選択構成の場合、各構成要素に設定されている開始セパレータの値に重複はないか。	開始セパレータが重複した構成要素に、ほかの構成要素で設定されていない開始セパレータの値を設定します。	-	
構成要素	出現回数が固定の場合、指定した出現回数は次の式を満たすか。 $0 < \text{出現回数} \leq 2,147,483,647$	出現回数を 1 ~ 2,147,483,647 で設定します。		-
	出現回数が「出現回数ノード」の場合、出現回数ノードが設定されているか。	該当する構成要素に出現回数ノードを設定します。		-
	出現回数が「出現回数ノード」の場合、出現回数ノードのデータ種別が数値型か（数値型とは、整数、実数、固定小数部数値、ゾーン形式数値、バック形式数値、符号付き 2 進整数、または符号なし 2 進整数）。	出現回数ノードには、データ種別が数値型の単純内容要素を設定します。		
	出現回数が「出現回数ノード」の場合、出現回数ノード、およびそのすべての祖先ノードの出現回数が固定、かつ 1 回か。	出現回数ノードとその祖先ノードの出現回数を 1 回固定になるように設定します。		-
	出現回数が「出現回数ノード」の場合、出現回数ノードのパスが正しいか。	出現回数ノードを示すパスが不正であるおそれがあります。対処方法については「表 4-7 パスが不正である場合の原因と対処」を参照してください。		

#### 4. 電文フォーマットの作成

検証の対象	検証内容	エラー発生時の対処	検証方法	
			自動	任意
	出現回数が「出現回数ノード」の場合、出現回数ノードのパスに選択構成の複合内容要素が含まれていないか。(ただし、パスが絶対パスの場合、設定元のノードの祖先ノードは除く)	出現回数ノードが実際のバイナリデータで出現しないおそれがあります。実際のバイナリデータで出現回数ノードに該当するデータが出現するか確認します。	-	
	再帰構造になっていないか。	再帰的に設定されている構成要素を削除します。		-
	構成要素が単純内容要素の場合で、サイズが「サイズノード」のとき、サイズノードが設定されているか。	該当する構成要素にサイズノードを設定します。	-	
	構成要素が単純内容要素の場合で、サイズが「サイズノード」のとき、サイズノードのデータ種別が数値型か(数値型とは、整数、実数、固定小数部数値、ゾーン形式数値、パック形式数値、符号付き2進整数、または符号なし2進整数)	サイズノードには、データ種別が数値型の単純内容要素を設定します。		
	構成要素が単純内容要素の場合で、サイズが「サイズノード」のとき、サイズノード、およびそのすべて祖先ノードの出現回数が固定、かつ1回か。	サイズノードとその祖先ノードの出現回数を1回固定になるように設定します。	-	
	構成要素が単純内容要素の場合で、サイズが「サイズノード」のとき、サイズノードのパスが正しいか。	サイズノードを示すパスが不正であるおそれがあります。対処方法については「表 4-7 パスが不正である場合の原因と対処」を参照してください。		
	構成要素が単純内容要素の場合で、サイズが「サイズノード」のとき、サイズノードのパスに選択構成の複合内容要素が含まれていないか(ただし、パスが絶対パスの場合、設定元のノードの祖先ノードは除く)	サイズノードが実際のバイナリデータで出現しないおそれがあります。実際のバイナリデータでサイズノードに該当するデータが出現するか確認します。	-	
	構成要素が単純内容要素の場合で、出現回数が「データの終端まで」の場合、出現の終わりを示すセパレータが設定されていること。	出現回数が可変の要素で、出現の終わりを示すセパレータを指定します。		
	構成要素が単純内容要素の場合で、サイズが「データの終端まで」の場合、データの終了を示すセパレータが設定されていること。	サイズが可変長の要素で、データの終了を示すセパレータを指定します。		
	構成要素が複合内容要素の場合で、出現回数が「データの終端まで」の場合、出現の終わりを示すセパレータが設定されていること。	出現回数が可変の要素で、出現の終わりを示すセパレータを指定します。		

(凡例)

- : 検証されます。
- : 検証されません。

表 4-6 選択条件値に指定した値に関する検証内容と対処

データの種別	検証内容	対処
文字列	次の式を満たすか。 文字コード変換後のサイズ 指定のサイズ	選択条件値のバイトサイズが選択条件ノードのサイズ以下になるように指定してください。
整数	指定した文字列の形式は $n$ か。 <sup>1 2</sup>	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。
	次の式を満たすか。 文字コード変換後のサイズ 指定のサイズ	選択条件値のバイトサイズが選択条件ノードのサイズ以下になるように指定してください。
	符号の有無が「なし」の場合、 $0$ 値か。	選択条件値に正の値を指定してください。
	整数は 34 桁以下か。	選択条件値に有効桁数が 34 桁以内の数値を指定してください。
実数	指定した文字列の形式は $n$ または $n.m$ か。 <sup>1 2</sup>	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。
	次の式を満たすか。 文字コード変換後のサイズ 指定のサイズ	選択条件値のバイトサイズが選択条件ノードのサイズ以下になるように指定してください。
	符号の有無が「なし」の場合、 $0$ 値か。	選択条件値に正の値を指定してください。
	整数部桁 + 小数部桁が 34 桁以下の実数か。	選択条件値に有効桁数が 34 桁以内の数値を指定してください。
固定小数部数値	指定した文字列の形式は $n$ または $n.m$ か。 <sup>1 2</sup>	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。
	次の式を満たすか。 暗黙的小数部付き数値に変換後のサイズ 指定のサイズ	選択条件値のバイトサイズが選択条件ノードのサイズ以下になるように指定してください。
	符号の有無が「なし」の場合、 $0$ 値か。	選択条件値に正の値を指定してください。
	整数部桁 + 小数部桁が 34 桁以下の実数か。	選択条件値に有効桁数が 34 桁以内の数値を指定してください。
	小数部桁は「小数部桁数」に指定した範囲内か。	選択条件値の小数部の桁数を小数部桁数以下で指定してください。
ゾーン形式数値	指定した文字列の形式は $n$ または $n.m$ か。 <sup>1</sup>	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。

#### 4. 電文フォーマットの作成

データの種別	検証内容	対処
	次の式を満たすか。 ゾーン形式に変換後のサイズ 指定のサイズ	選択条件値のバイトサイズが選択条件ノードのサイズ以下になるように指定してください。
	整数部桁 + 小数部桁が 34 桁以下の実数か。	選択条件値に有効桁数が 34 桁以内の数値を指定してください。
	小数部桁は「小数部桁数」に指定した範囲内か。	選択条件値の小数部の桁数を小数部桁数以下で指定してください。
パック形式数値	指定した文字列の形式は n または n.m か。 1	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。
	次の式を満たすか。 パック形式に変換後のサイズ 指定のサイズ	選択条件値のバイトサイズが選択条件ノードのサイズ以下になるように指定してください。
	整数部桁 + 小数部桁が 34 桁以下の実数か。	選択条件値に有効桁数が 34 桁以内の数値を指定してください。
	小数部桁は「小数部桁数」に指定した範囲内か。	選択条件値の小数部の桁数を小数部桁数以下で指定してください。
符号付き 2 進整数	指定した文字列の形式は n または -n か。 1	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。
	サイズが 1 バイトの場合、次の式を満たす整数か。 $-2^7$ 値 $2^7-1$	選択条件値に有効な値を指定してください。
	サイズが 2 バイトの場合、次の式を満たす整数か。 $-2^{15}$ 値 $2^{15}-1$	選択条件値に有効な値を指定してください。
	サイズが 4 バイトの場合、次の式を満たす整数か。 $-2^{31}$ 値 $2^{31}-1$	選択条件値に有効な値を指定してください。
	サイズが 8 バイトの場合、次の式を満たす整数か。 $-2^{63}$ 値 $2^{63}-1$	選択条件値に有効な値を指定してください。
符号なし 2 進整数	指定した文字列の形式は n か。 1	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。

データの種別	検証内容	対処
	サイズが 1 バイトの場合、次の式を満たす整数か。 $0 \text{ 値} < 2^8$	選択条件値に有効な値を指定してください。
	サイズが 2 バイトの場合、次の式を満たす整数か。 $0 \text{ 値} < 2^{16}$	選択条件値に有効な値を指定してください。
	サイズが 4 バイトの場合、次の式を満たす整数か。 $0 \text{ 値} < 2^{32}$	選択条件値に有効な値を指定してください。
	サイズが 8 バイトの場合、次の式を満たす整数か。 $0 \text{ 値} < 2^{64}$	選択条件値に有効な値を指定してください。
バイト列	指定した文字列の形式は 0 ~ 9, a ~ f, A ~ F か。	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。
	次の式を満たすか。 バイト変換後のサイズ=指定のサイズ	条件値のバイトサイズが選択条件ノードのバイトサイズと同じになるように定義してください。
日付時刻	形式が CCYYMMDD の場合、指定した文字列形式は CCYYMMDD か。	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。
	形式が YYMMDD の場合、指定した文字列形式は YYMMDD か。	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。
	形式が hhmmss の場合、指定した文字列形式は hhmmss か。	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。
	形式が CCYYMMDDhhmmss の場合、指定した文字列形式は CCYYMMDDhhmmss か。	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。
	形式が YYMMDDhhmmss の場合、指定した文字列形式は YYMMDDhhmmss か。	選択条件値に選択条件ノードのデータ種別と一致した文字列を指定してください。
	秒小数点の指定は、秒小数点の桁数と一致するか。	選択条件値の秒小数点桁数と選択条件ノードに指定した値を一致させてください。
	指定した日付時刻は正しい日付時刻か。	選択条件値に正しい日付時刻を指定してください。

#### 4. 電文フォーマットの作成

注 1

n, m は 0 ~ 9 の数値を表します。ただし, n の先頭, および m の末尾は 1 ~ 9 の数値です。

注 2

条件値に付ける符号は, 選択条件ノードの「符号の有無」の設定によって異なります。「常にある」の場合は必ず符号を付けます。「なし」の場合, 符号は指定できません。「負だけあり」の場合は「-」だけ指定できます。

表 4-7 パスが不正である場合の原因と対処

パス不正の原因	対処
選択条件, 出現回数, またはサイズノードに指定したパスのノードが定義上から削除されている。	選択条件, 出現回数, またはサイズノードを指定し直してください。
選択条件, 出現回数, またはサイズノードを指定したあとに, 階層構造が変更された。	選択条件, 出現回数, またはサイズノードを指定し直してください。
選択条件, 出現回数, またはサイズノードに指定したパスが, 設定先の要素よりもあとに出現する。	選択条件, 出現回数, またはサイズノードに, 設定先より前に出現する要素を指定してください。
ノードパス設定先のノードを参照する構成要素が複数定義されている。	ノードパス設定先を参照する構成要素を一つにしてください。
絶対パスの最上位要素がルート要素ではない。	ルート要素を絶対パスの最上位要素に設定するか, パスを設定し直してください。
設定元のノードの祖先に基準ノードが存在しない。	ノードパスを設定し直してください。
基準パスの基準ノードが逐次構成の要素ではない。	基準ノードが逐次構成になるパスを設定してください。
パスと設定元のノードの共通の親が, 選択構成の複合内容である。	パスと設定元のノードの共通の親を逐次構成の複合内容に設定してください。

### (2) 検証方法

検証方法には, 次の二つがあります。検証方法によって検証される内容は異なります。詳細について「4.4.5(1) 検証内容」を参照してください。

#### ダイアログの入力時の検証

バイナリフォーマット定義ファイルの各要素を設定するときに利用するダイアログでは, 入力された値が逐次自動的に検証されます。

#### 任意のタイミングでの検証

バイナリフォーマット定義ファイルの作成後, または作成途中の任意のタイミングで検証を実施できます。

検証を実施するには, バイナリフォーマット定義エディタ上の要素を選択して右クリックし, [ 検証 ] を選択します。バイナリフォーマット定義エディタに表示されているバイナリフォーマット定義ファイルが検証されます。

なお, 検証するときにバイナリフォーマット定義エディタ上で選択する要素は任意です。どの要素を選択しても, バイナリフォーマット定義ファイル全体が検証されます。

検証結果の表示については、「4.4.5(3) 検証結果の表示」を参照してください。

### (3) 検証結果の表示

検証を実施した結果は、画面およびダイアログ上に表示されます。

#### (a) ダイアログの入力時の自動検証の結果の表示

ダイアログの入力時の自動検証の結果は、バイナリフォーマット定義に関連するダイアログの [ インフォメーション ] に表示されます。問題がない場合は、何も表示されません。

バイナリフォーマット定義に関連するダイアログについては、「11.3 バイナリフォーマット定義に関連するダイアログ」を参照してください。

#### (b) 任意のタイミングで検証した結果の表示

検証の結果が画面に表示されます。画面の表示内容については、「11.1 画面構成」および「11.2.1 バイナリフォーマット定義画面」を参照してください。

#### 検証が成功した場合

コンソールビューに検証が成功したことを示すメッセージが表示されます。

#### 検証が失敗した場合

コンソールビューにエラー、警告、および情報通知の内容を示すメッセージが表示されます。また、画面上の各部分にエラー、警告、および情報通知の内容が表示されます。

#### バイナリフォーマット定義エディタ、アウトラインビュー

問題のあったノードにエラーまたは警告を示すアイコンが表示されます。エラーと警告の両方がある場合、エラーのアイコンが優先して表示されます。

#### パッケージ・エクスプローラー

問題のあったファイルとその祖先のフォルダにエラーまたは警告を示すアイコンが表示されます。エラーと警告の両方がある場合、エラーのアイコンが優先して表示されます。

#### 問題ビュー

問題点を示すメッセージ、リソース名（ファイル名）およびフォルダ名が一覧表示されます。



# 5

## アダプタの定義

この章では、サービスアダプタおよびDBアダプタの定義について説明します。

---

5.1 定義の流れ

---

5.2 アダプタの追加

---

5.3 アダプタの内容の定義

---

5.4 アダプタの保存

---

5.5 アダプタの編集

---

5.6 アダプタの検証

---

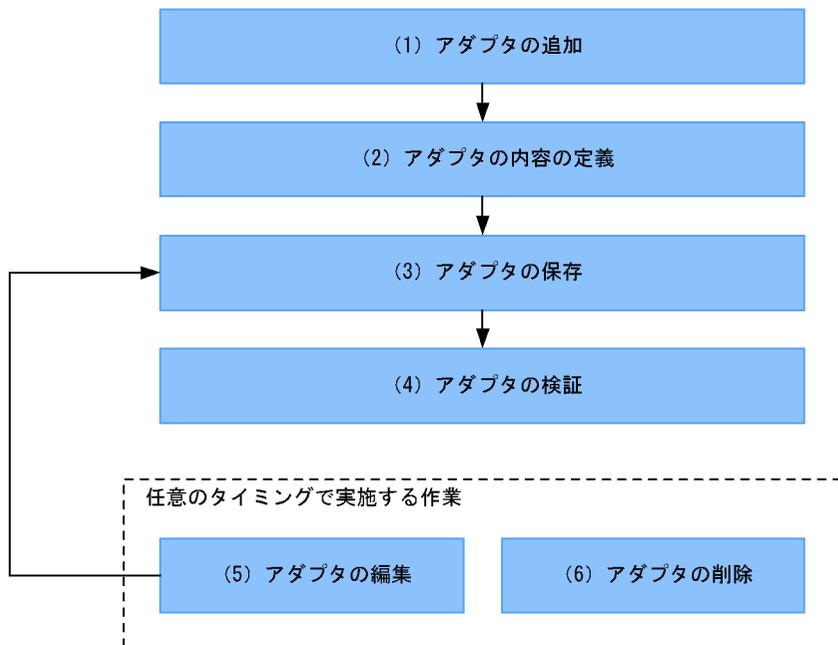
5.7 アダプタの削除

---

## 5.1 定義の流れ

サービスアダプタおよび DB アダプタの定義の流れを次の図に示します。

図 5-1 アダプタの定義の流れ



アダプタの定義に関する作業について次に説明します。

### (1) アダプタの追加

アダプタを追加するには、次のどちらかの方法を利用します。

#### 新規追加する場合

ウィザードを利用してアダプタを新規に追加します。アダプタを新規に追加するウィザードは、アダプタが呼び出すサービス部品の種類（Web サービス，SessionBean，MDB（WS-R または DB キュー））によって異なります。アダプタを新規に追加する方法の詳細については、「5.2.1 サービスアダプタを新規に追加する」または「5.2.2 DB アダプタを新規に追加する」を参照してください。

#### 定義済みのアダプタを利用する場合

定義済みのアダプタを複製してアダプタを追加できます。定義済みのアダプタを複製すると、同じ定義内容のアダプタが追加されます。複製したアダプタの定義内容を編集することもできます。

定義済みのアダプタを利用してアダプタを追加する方法の詳細については、「5.2.3 定義済みアダプタを利用してアダプタを追加する」を参照してください。

## (2) アダプタの内容の定義

サービスアダプタ定義画面でアダプタの内容を定義します。

定義の方法および定義の内容の詳細については、「5.3 アダプタの内容の定義」を参照してください。サービスアダプタ定義画面の詳細については、「11.2.2 サービスアダプタ定義画面」を参照してください。

## (3) アダプタの保存

編集したアダプタの定義情報は、必要に応じてリポジトリに保存する必要があります。

アダプタの定義情報の保存方法と保存されるタイミングについては、「5.4 アダプタの保存」を参照してください。

## (4) アダプタの検証

定義したアダプタの整合性を検証します。アダプタの定義中や定義が終わったあとなど、任意のタイミングで検証できます。

検証方法の詳細については、「5.6 アダプタの検証」を参照してください。

## (5) アダプタの編集

定義済みのアダプタは、必要に応じて定義内容を編集して変更できます。

アダプタの編集方法の詳細については、「5.5 アダプタの編集」を参照してください。

## (6) アダプタの削除

不要になったアダプタは、必要に応じて削除できます。

アダプタの削除方法については、「5.7 アダプタの削除」を参照してください。

## 5.2 アダプタの追加

---

サービスアダプタおよび DB アダプタを追加します。

アダプタを新規に追加する場合、アダプタ追加ウィザードを利用します。また、すでに作成されているアダプタを利用して新規追加することもできます。

### 5.2.1 サービスアダプタを新規に追加する

サービスアダプタを新規に追加するには、まずツリービューにサービス定義一覧を表示させます。サービス定義一覧の表示方法を次に示します。

1. Eclipse のメニューから、[ ウィンドウ ] - [ ビューの表示 ] - [ その他 ] を選択します。  
[ ビューの表示 ] ダイアログが表示されます。
2. [ HCSC-Definer ] - [ HCSCTE ビュー ] を選択し、[ OK ] ボタンをクリックします。  
ツリービューにサービス定義一覧が表示されます。

以降の操作は、利用するサービス部品の種別によって異なります。サービスアダプタを新規追加する方法を利用するサービス部品別に次に示します。

#### (1) Web サービスの場合

1. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。  
追加するサービスアダプタから利用するサービス種別を設定するダイアログが表示されます。
2. ドロップダウンリストから「Web サービス」を選択します。
3. [ 次へ ] ボタンをクリックします。  
サービスアダプタの追加に必要な情報を入力するダイアログが表示されます。
4. サービス名と WSDL ファイルを入力します。  
WSDL ファイルには、拡張子が「.wsdl」のファイルを絶対パスで指定します。  
相対パス（例：wsldir¥wsdlfile.wsdl）や UNC 形式（例：  
¥¥mypc¥wsldir¥wsdlfile.wsdl）で指定しないでください。
5. [ 次へ ] ボタンをクリックします。  
ポートを指定するダイアログが表示されます。  
[ 次へ ] ボタンをクリックしたときに、「WSDL ファイルの解析に失敗しました。」というメッセージが表示された場合、WSDL ファイルにエラーがあります。  
WSDL の形式については、「2.5.1(3) WSDL の形式」、およびマニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。
6. ドロップダウンリストでポートを選択します。

7. [終了] ボタンをクリックします。  
必要なファイルが作成され、リポジトリに保存されます。  
サービスアダプタ定義画面が表示されます。

## (2) SessionBean の場合

1. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。  
追加するサービスアダプタから利用するサービス種別を設定するダイアログが表示されます。
2. ドロップダウンリストから「SessionBean」を選択します。
3. [次へ] ボタンをクリックします。  
サービスアダプタの追加に必要な情報を入力するダイアログが表示されます。
4. サービス名と EAR ファイルを入力します。  
EAR ファイルには、拡張子が「.ear」のファイルを絶対パスで指定します。
5. [次へ] ボタンをクリックします。  
SessionBean を指定するダイアログが表示されます。
6. ドロップダウンリストで SessionBean を選択します。
7. [終了] ボタンをクリックします。  
必要なファイルが作成され、リポジトリに保存されます。  
サービスアダプタ定義画面が表示されます。

## (3) MDB (WS-R または DB キュー) の場合

1. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。  
追加するサービスアダプタから利用するサービス種別を設定するダイアログが表示されます。
2. ドロップダウンリストから「MDB/WS-R」または「MDB/DB キュー」を選択します。
3. [次へ] ボタンをクリックします。  
サービスアダプタの追加に必要な情報を入力するダイアログが表示されます。
4. サービス名を入力します。
5. [終了] ボタンをクリックします。  
必要なファイルが作成され、リポジトリに保存されます。  
サービスアダプタ定義画面が表示されます。

## 5.2.2 DB アダプタを新規に追加する

DB アダプタを新規追加する方法を次に示します。

## 5. アダプタの定義

1. Eclipse のメニューから、[ ウィンドウ ] - [ ビューの表示 ] - [ その他 ] を選択します。  
[ ビューの表示 ] ダイアログが表示されます。
2. [ HCSC-Definer ] - [ HCSCTE ビュー ] を選択し、[ OK ] ボタンをクリックします。  
ツリービューにサービス定義一覧が表示されます。
3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。  
アダプタの新規追加ダイアログが表示されます。
4. ドロップダウンリストから「汎用カスタムアダプタ」を選択します。
5. [ 次へ ] ボタンをクリックします。  
DB アダプタの追加に必要な情報を入力するダイアログが表示されます。
6. サービス名と EAR ファイルを入力します。  
EAR ファイルには、拡張子が「.ear」のファイルを絶対パスで指定します。  
指定する EAR ファイルは「<Cosminexus のインストールディレクトリ  
>¥CSC¥lib¥cscdba.ear」です。
7. [ 終了 ] ボタンをクリックします。  
必要なファイルが作成され、リポジトリに保存されます。  
サービスアダプタ定義画面が表示されます。

### 5.2.3 定義済みアダプタを利用してアダプタを追加する

すでに定義されているアダプタを複製して、アダプタを追加できます。

1. ツリービューのサービス定義一覧で、複製するアダプタを選択して右クリックします。  
サービス一覧のポップアップメニューが表示されます。
2. ポップアップメニューから「複製」を選択します。  
選択したアダプタの複製が作成されます。複製されたアダプタのサービス名やサービス ID などは、システム内で競合しないよう、自動的に別の情報が割り当てられます。

## 5.3 アダプタの内容の定義

サービスリクエストから受け付けた要求は、クラスタ（または単一の HCSC サーバ）内に配備されるサービスアダプタを特定することで、サービス部品へ送ることができます。サービス部品には、そのサービス部品に対応したアダプタが必要です。この節では、アダプタの定義方法について説明します。

サービス部品には、Web サービス、SessionBean および MDB があります。また、データベースへのアクセスをサービス部品として利用することもできます。

各サービス部品固有のプロトコルに対応するサービスアダプタは、サービス部品の WSDL などを基に、開発環境から自動生成できます。これらを標準提供アダプタといいます。サービス部品の種類によって、使用するアダプタの種類やアダプタの定義項目が次のように異なります。

### サービスアダプタ

Web サービス、SessionBean、および MDB の場合に使用します。各サービス部品には、固有のプロトコルがあります。Web サービスの場合は SOAP、SessionBean の場合は RMI-IIOP、MDB の場合は WS-R および DB キューのプロトコルが、それぞれあります。サービスアダプタは、これらのサービス部品（プロトコル）の種類によって、定義内容が少しずつ異なります。サービスアダプタは、サービスアダプタ定義画面で定義します。

### DB アダプタ

データベースへアクセスする場合に使用します。DB アダプタは、サービスアダプタ定義画面で定義します。

### 5.3.1 サービスアダプタ（Web サービス）を定義する

Web サービスに対するアダプタの定義について説明します。Web サービスに対するアダプタは、サービスアダプタ定義画面で定義できます。

#### （1）サービス部品の電文フォーマット

Web サービスの場合、WSDL から作成された電文フォーマット定義ファイルが、サービス部品の電文フォーマット定義ファイルとなります。そのため、要求電文および応答電文ともに、サービス部品の電文フォーマットを設定する必要はありません。

サービス部品の電文フォーマットの作成方法については、「4.3.2 サービス部品電文を作成する（Web サービスの場合）」を参照してください。

#### （2）データ変換

サービスリクエストの電文の形式と、サービスアダプタが呼び出すサービス部品の電文の形式が異なる場合、データ変換が必要です。

## 5. アダプタの定義

要求電文のデータ変換を行う場合、サービスアダプタ定義画面で標準電文にサービスリクエストから入力される電文を設定し、データ変換定義画面でデータ変換（マッピング）を定義します。応答電文のデータ変換を行う場合、サービス部品から返される電文に標準電文を設定し、データ変換定義画面でデータ変換（マッピング）を定義します。

データ変換の詳細については、「7. データ変換の定義」を参照してください。

### （3）クライアント定義ファイルの作成と設定

サービス部品の利用者は、クライアント（サービスリクエスト）側の動作を制御するクライアント定義ファイルを、「c4webcl.properties」というファイル名称で作成し、サービスアダプタ定義画面で設定する必要があります。「c4webcl.properties」はすべて小文字で入力します。クライアント定義ファイルの作成については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。

クライアント定義ファイルの作成に関する注意事項を次の表に示します。

表 5-1 クライアント定義ファイルの作成に関する注意事項

設定項目	キー名称	設定値 (デフォルト 値)	注意事項	設定 可否
多重参照	c4web.common.do_multirefs	false	デフォルト値以外の値を指定した場合の動作は保証できません。	×
データ型定義	c4web.common.send_xsi_types	true		×
文字参照形式オプション	c4web.common.character_reference	false		×
HTTP セッションの維持	c4web.application.app_maintainsession	false		×
SOAP ヘッダの名前修飾チェックオプション	c4web.common.enable_soapheader_check	true	ユーザ定義受付の場合、このオプションは、false として動作します。	
トレースファイル、アプリケーションログのプレフィクス	c4web.logger.log_file_prefix	サービス ID	値は必ずサービス ID に変更されるため、設定する必要はありません。	×

（凡例）

：設定できます。

×：設定できません。

### （4）ベーシック認証の設定

対象となるサービス部品がユーザ認証を必要とする Web サーバ上にある場合は、ベーシック認証（ユーザ名およびパスワード）の設定が必要になります。

## (5) サービスアダプタ定義画面の表示

サービスアダプタを定義するために、サービスアダプタ定義画面を表示します。サービスアダプタ定義画面は、ツリービューのサービス定義一覧から、該当するサービスアダプタを選択し、ダブルクリックすると表示されます。

## (6) サービスアダプタ定義画面での操作

Web サービスに対するアダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。  
サービスアダプタ定義画面の表示方法については、「5.3.1(5) サービスアダプタ定義画面の表示」を参照してください。
2. サービス部品制御情報の [ サービス名 ], [ サービス ID ], および [ 最大インスタンス数 ] を必要に応じて編集します。
3. サービス部品制御情報の [ オペレーション ] のドロップダウンリストから、編集したいオペレーションを選択します。
4. オペレーション情報に表示されている内容を確認します。
5. 要求電文について、手順 6. から手順 16. までを実施します。
6. 次に示す操作をします。  
標準電文のフォーマットを指定する場合  
手順 7. ~ 10. を実施して、手順 11. に進みます。  
標準電文のフォーマットを指定しない場合  
手順 11. に進みます。
7. 標準電文の [ 使う ] チェックボックスをチェックします。
8. 標準電文の [ フォーマット ID ] を指定します。
9. 標準電文の [ ... ] ボタンをクリックして、[ 電文フォーマット ] に標準電文のフォーマットを指定します。  
なお、外部の XML スキーマを参照している電文フォーマットを設定する場合、必ずルートスキーマに当たるファイルを設定してください。ルートスキーマから参照される外部の XML スキーマファイルは、自動的に取り込まれます。  
指定できる電文フォーマットの形式については、「2.5.5 XML スキーマの適用範囲」を参照してください。
10. 標準電文の [ 表示 ] ボタンをクリックします。  
標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。
11. サービス部品電文の [ フォーマット ID ] を指定します。
12. サービス部品電文の [ 表示 ] ボタンをクリックします。  
サービス部品電文のフォーマットが表示されます。必要に応じてサービス部品電文の

## 5. アダプタの定義

フォーマットを確認してください。

13.次に示す操作をします。

標準電文の [ 使う ] チェックボックスをチェックした場合  
手順 14. ~ 16. を実施して、手順 17. に進みます。

標準電文の [ 使う ] チェックボックスをチェックしない場合  
手順 17. に進みます。

14.データ変換定義のファイル名を入力します。

15.[ 編集 ] ボタンをクリックします。

データ変換定義画面が表示されます。初めて定義する場合は、[ ルート要素選択 ] ダイアログが表示されます。

16.標準電文とサービス部品電文の内容をマッピングします。

17.応答電文についても同様に、手順 6. から手順 16. までを実施します。

18.サービスアダプタ定義 ( 詳細 ) タブをクリックします。

19.[ ... ] ボタンをクリックして、クライアント定義ファイル ( c4webcl.properties ) を指定します。

20.サービス部品がユーザ認証を必要とする Web サーバ上にあるかどうかで、次に示す操作をします。

ユーザ認証が必要な場合

手順 21 ~ 23. を実施して、手順 24. に進みます。

ユーザ認証が不要な場合

手順 24. に進みます。

21.ベーシック認証の [ 使う ] チェックボックスをチェックします。

22.ユーザ名およびパスワードの欄に、ベーシック認証のために必要なユーザ名およびパスワードを入力します。

23.パスワード ( 確認 ) 欄に、パスワードの欄で入力したパスワードと同じパスワードを入力します。

24.定義内容を確認して、Eclipse のメニューから [ ファイル ] - [ 保管 ] を選択し、定義内容を保存します。

### 5.3.2 サービスアダプタ ( SessionBean ) を定義する

SessionBean に対するアダプタの定義について説明します。SessionBean に対するアダプタは、サービスアダプタ定義画面で定義できます。

#### ( 1 ) サービス部品の電文フォーマット

SessionBean の場合、EAR ファイルから自動作成された電文フォーマット定義ファイル

が、サービス部品の電文フォーマット定義ファイルとなります。そのため、サービス部品の電文フォーマットを設定する必要はありません。

サービス部品の電文フォーマットの作成方法については、「4.3.3 サービス部品電文を作成する (SessionBean の場合)」を参照してください。

## (2) データ変換

サービスリクエストの電文の形式と、サービスアダプタが呼び出すサービス部品の電文の形式が異なる場合、データ変換が必要です。

要求電文のデータ変換を行う場合、サービスアダプタ定義画面で標準電文にサービスリクエストから入力される電文を設定し、データ変換定義画面でデータ変換 (マッピング) を定義します。応答電文のデータ変換を行う場合、サービス部品から返される電文に標準電文を設定し、データ変換定義画面でデータ変換 (マッピング) を定義します。

データ変換の詳細については、「7. データ変換の定義」を参照してください。

## (3) JNDI 名前空間に登録されている呼び出すサービス部品の名称の指定

サービスアダプタ定義画面では、JNDI 名前空間に登録されている呼び出すサービス部品の名称を指定します。リモート呼び出しとなるようにアドレスを指定してください。サービス部品 (SessionBean) のローカル呼び出しはできません。指定例を次に示します。

- 正しい例 (myhost はリモートホスト名を表します)

```
corbaname::myhost:900#HITACHI_EJB/SERVERS/MyServer/EJB/slbsmpl/MyAdder
```

- 誤った例

```
corbaname::localhost:900#HITACHI_EJB/SERVERS/MyServer/EJB/slbsmpl/MyAdder
```

```
corbaname::127.0.0.1:900#HITACHI_EJB/SERVERS/MyServer/EJB/slbsmpl/MyAdder
```

```
corbaname::HITACHI_EJB/SERVERS/MyServer/EJB/slbsmpl/MyAdder
```

JNDI 名前空間に登録されている名称については、マニュアル「Cosminexus 機能解説」のネーミング管理に関する内容、および「Cosminexus アプリケーション開発ガイド」の EJB コンテナ名前空間に関する内容を参照してください。

## (4) クライアント定義ファイルの作成と設定

Web サービスと同様に、サービス部品の利用者は、クライアント (サービスリクエスト) 側の動作を制御するクライアント定義ファイルを、「c4webcl.properties」というファイ

## 5. アダプタの定義

ル名称で作成し、サービスアダプタ定義画面で設定する必要があります。

「c4webcl.properties」はすべて小文字で入力します。クライアント定義ファイルの作成については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。

なお、サービスアダプタ (SessionBean) の場合、クライアント定義ファイルに設定できる項目とできない項目があります。サービスアダプタ (SessionBean) の場合に設定できるクライアント定義ファイルの項目を次の表に示します。

表 5-2 クライアント定義ファイルの設定項目

キー名称	内容	設定可否
c4web.logger.log_level	トレースファイル出力の重要度	
c4web.logger.aplog_level	アプリケーションログ出力の重要度	
c4web.logger.aplog_error_record	異常発生時のアプリケーションログ出力	
c4web.logger.log_file_dir	トレースファイルおよびアプリケーションログの出力先ディレクトリ	
c4web.logger.log_file_num	トレースファイルの面数	
c4web.logger.log_file_size	トレースファイルのサイズ	
c4web.logger.aplog_file_num	アプリケーションログの面数	
c4web.logger.aplog_file_size	アプリケーションログのサイズ	
c4web.logger.log_file_prefix	トレースファイル、アプリケーションログのプレフィックス	×
c4web.common.do_multirefs	多重参照 (実行時オプション)	×
c4web.common.send_xsi_types	データ型定義 (実行時オプション)	×
c4web.application.app_maintainsession	HTTP セッションの維持 (実行時オプション)	×
c4web.application.proxy_host	プロキシサーバのホスト名 (実行時オプション)	×
c4web.application.non_proxy_hosts	プロキシサーバを使用しないホスト名群 (実行時オプション)	×
c4web.application.proxy_port	プロキシサーバのポート番号 (実行時オプション)	×
c4web.application.proxy_user	プロキシサーバの認証ユーザ ID (実行時オプション)	×
c4web.application.proxy_password	プロキシサーバの認証ユーザ ID に対応するパスワード (実行時オプション)	×
c4web.common.enable_soapheader_check	SOAP ヘッダの名前修飾チェックオプション (実行時オプション)	×
c4web.application.socket_timeout	クライアントのソケットタイムアウト値 (実行時オプション)	×
c4web.attachment.file_body_encoding	ボディエンコーディング種別	×

キー名称	内容	設定可否
c4web.attachment.attachment_temp_directory	退避先ディレクトリ	×
c4web.attachment.attachment_file_delete_by_date	不要退避ファイルの経過日数	×
c4web.common.prf_trace_level	性能解析トレース出力オプション	

(凡例)

○ : 必ず設定してください。

○ : 設定できます。

× : 設定できません。設定した場合は無視されます。

注 値は必ずサービス ID に変更されるため、設定する必要はありません。

### (5) ユーザ定義クラス (JAR ファイル) の設定

SessionBean の場合、スタブを利用してサービス部品を呼び出すために、ユーザ定義クラス (JAR ファイル) を設定する必要があります。

なお、サービスアダプタ定義画面 (SessionBean の詳細画面) で JAR ファイルを設定するとき、選択するファイルの拡張子は、すべて小文字の「jar」としてください。

また、JAR ファイルとして次に示すファイル名は設定しないでください。

- csmsvcadpdef.jar
- cscmsg\_adpejb.jar

上記のファイル名を設定した場合、警告メッセージが表示されます。警告メッセージが表示された場合、上記のファイル名をユーザ定義クラスから削除します。そのあと、上記のファイル名を変更して、再度設定してください。

### (6) RMI-IIOP スタブおよびインターフェースの取得

J2EE サーバの `cjgetstubsjar` コマンドを使って RMI-IIOP スタブおよびインターフェースを取得します。`cjgetstubsjar` コマンドの詳細について、マニュアル「Cosminexus リファレンス コマンド編」を参照してください。

### (7) サービスアダプタ定義画面での操作

SessionBean に対するアダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。  
サービスアダプタ定義画面の表示方法については、「5.3.1(5) サービスアダプタ定義画面の表示」を参照してください。
2. サービス部品制御情報の [ サービス名 ], [ サービス ID ], および [ 最大インスタンス数 ] を必要に応じて編集します。
3. サービス部品制御情報の [ アドレス ] を指定します。

## 5. アダプタの定義

[ アドレス ] には、JNDI 名前空間に登録されている呼び出すサービス部品の名称を取得して、入力します。JNDI 名前空間に登録されている名称については、マニュアル「Cosminexus 機能解説」のネーミング管理に関する内容、および「Cosminexus アプリケーション開発ガイド」の EJB コンテナ名前空間に関する内容を参照してください。

4. サービス部品制御情報の [ オペレーション ] のドロップダウンリストから、編集したいオペレーションを選択します。

5. オペレーション情報に表示されている内容を確認します。

6. 要求電文について、手順 7. から手順 17. までを実施します。

7. 次に示す操作をします。

標準電文のフォーマットを指定する場合

手順 8. ~ 11. を実施して、手順 12. に進みます。

標準電文のフォーマットを指定しない場合

手順 12. に進みます。

8. 標準電文の [ 使う ] チェックボックスをチェックします。

9. 標準電文の [ フォーマット ID ] を指定します。

10. 標準電文の [ ... ] ボタンをクリックして、[ 電文フォーマット ] に標準電文のフォーマットを指定します。

なお、外部の XML スキーマを参照している電文フォーマットを設定する場合、必ずルートスキーマに当たるファイルを設定してください。ルートスキーマから参照される外部の XML スキーマファイルは、自動的に取り込まれます。

指定できる電文フォーマットの形式については、「2.5.5 XML スキーマの適用範囲」を参照してください。

11. 標準電文の [ 表示 ] ボタンをクリックします。

標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。

12. サービス部品電文の [ フォーマット ID ] を指定します。

13. サービス部品電文の [ 表示 ] ボタンをクリックします。

サービス部品電文のフォーマットが表示されます。必要に応じてサービス部品電文のフォーマットを確認してください。

14. 次に示す操作をします。

標準電文の [ 使う ] チェックボックスをチェックした場合

手順 15. ~ 17. を実施して、手順 18. に進みます。

標準電文の [ 使う ] チェックボックスをチェックしない場合

手順 18. に進みます。

15. データ変換定義のファイル名を入力します。

- 16.[編集] ボタンをクリックします。  
データ変換定義画面が表示されます。初めて定義する場合は,[ルート要素選択] ダイアログが表示されます。
- 17.標準電文とサービス部品電文の内容をマッピングします。
- 18.応答電文についても同様に,手順 6. から手順 17. までを実施します。
- 19.サービスアダプタ定義(詳細)タブをクリックします。
- 20.[...] ボタンをクリックして,クライアント定義ファイル(c4webcl.properties)を指定します。
- 21.[追加] ボタンをクリックして,該当するユーザ定義クラスを追加します。  
ユーザ定義クラスは,「5.3.2(6) RMI-IIOP スタブおよびインターフェースの取得」で取得したスタブおよびインターフェースのJAR ファイルを指定します。
- 22.定義内容を確認して,Eclipse のメニューから [ファイル] - [保管] を選択し,定義内容を保存します。

### 5.3.3 サービスアダプタ (MDB (WS-R)) を定義する

MDB (WS-R) に対するアダプタの定義について説明します。MDB (WS-R) に対するアダプタは,サービスアダプタ定義画面で定義できます。

#### (1) キュー情報およびメッセージ種別の取得

サービス部品がMDB (WS-R) の場合,サービス部品のキュー情報およびメッセージ種別を開発者から取得する必要があります。取得した情報は,サービスアダプタを定義するときに使用します。

取得する情報は次のとおりです。

- キュー名称
- 宛先のアドレス (アダプタとサービス部品との間でネットワーク通信をする場合)
- ベーシック認証のためのユーザ名およびパスワード (アダプタとサービス部品との間でネットワーク通信をし,かつベーシック認証を使用する場合)
- メッセージ種別

#### (2) サービス部品の電文フォーマット

MDB (WS-R) の場合,サービス部品の電文フォーマットを作成して,設定する必要があります。

サービス部品の電文フォーマットの作成方法については,「4.3.4 サービス部品電文を作成する (MDB (WS-R) または DB キュー) の場合)」を参照してください。

#### (3) データ変換

サービスリクエストの電文の形式と,サービスアダプタが呼び出すサービス部品の電文

## 5. アダプタの定義

の形式が異なる場合、データ変換が必要です。

データ変換を行う場合、サービスアダプタ定義画面で標準電文にサービスリクエストから入力される電文を設定し、データ変換定義画面でデータ変換（マッピング）を定義します。

データ変換の詳細については、「7. データ変換の定義」を参照してください。

### (4) 呼出しの設定

利用するサービス部品の呼出し方法を設定できます。リモート呼出しまたはローカル呼出しを選択できます。リモート呼出しを選択した場合、最大メッセージ数を設定します。また必要に応じて、ベーシック認証を設定します。

### (5) 最大メッセージ数の設定

サービス部品の呼出し方法としてリモート呼出しを設定した場合、一度に処理できる電文の数を初期値から変更した数に設定できます。設定できる値は1～65535です。

### (6) ベーシック認証の設定

サービス部品の呼出し方法としてリモート呼出しを設定した場合で、MDB (WS-R) のメッセージ転送でユーザ認証を必要とするときは、ベーシック認証（ユーザ名およびパスワード）の設定が必要になります。

### (7) サービスアダプタ定義画面での操作

MDB (WS-R) に対するアダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。  
サービスアダプタ定義画面の表示方法については、「5.3.1(5) サービスアダプタ定義画面の表示」を参照してください。
2. サービス部品制御情報の [ サービス名 ], [ サービス ID ], および [ 最大インスタンス数 ] を必要に応じて編集します。
3. サービス部品制御情報の [ アドレス ] を指定します。  
[ アドレス ] には、「5.3.3(1) キュー情報およびメッセージ種別の取得」で取得したキュー名称を入力します。
4. オペレーション情報に表示されている内容を確認します。
5. 要求電文について次に示す操作をします。  
標準電文のフォーマットを指定する場合  
手順 6. ~ 9. を実施して、手順 10. に進みます。  
標準電文のフォーマットを指定しない場合  
手順 10. に進みます。
6. 標準電文の [ 使う ] チェックボックスをチェックします。

7. 標準電文の [フォーマット ID] を指定します。
8. 標準電文の [...] ボタンをクリックして, [電文フォーマット] に標準電文のフォーマットを指定します。  
なお, 外部の XML スキーマを参照している電文フォーマットを設定する場合, 必ずルートスキーマに当たるファイルを設定してください。ルートスキーマから参照される外部の XML スキーマファイルは, 自動的に取り込まれます。  
指定できる電文フォーマットの形式については, 「2.5.5 XML スキーマの適用範囲」を参照してください。
9. 標準電文の [表示] ボタンをクリックします。  
標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。
10. サービス部品電文の [フォーマット ID] を指定します。
11. サービス部品電文の [...] ボタンをクリックして, [電文フォーマット] にサービス部品電文のフォーマットを指定します。
12. サービス部品電文の [表示] ボタンをクリックします。  
サービス部品電文のフォーマットが表示されます。必要に応じて指定したサービス部品電文のフォーマットを確認してください。
13. サービスアダプタ定義 (詳細) タブをクリックします。
14. [JMS メッセージ種別] のドロップダウンリストから, JMS メッセージ種別を選択します。
15. [呼出し設定] でサービス部品の呼出方法を選択します。  
[リモート呼出し] ラジオボタンを選択した場合  
手順 16. に進みます。  
[ローカル呼出し] ラジオボタンを選択した場合  
手順 24. に進みます。
16. [最大メッセージ数] に, サービス部品が送信できる電文の数を指定します。
17. [宛先 URL] に「5.3.3(1) キュー情報およびメッセージ種別の取得」で取得した, 宛先のアドレスを入力します。
18. サービス部品がユーザ認証を必要とするかどうかで, 次に示す操作をします。  
ユーザ認証が必要な場合  
手順 19. ~ 21. を実施して, 手順 22. に進みます。  
ユーザ認証が不要な場合  
手順 22. に進みます。
19. ベーシック認証の [使う] チェックボックスをチェックします。
20. ユーザ名およびパスワードの欄に, ベーシック認証のために必要なユーザ名およびパスワードを入力します。

## 5. アダプタの定義

- 21.パスワード（確認）欄に、パスワードの欄で入力したパスワードと同じパスワードを入力します。
- 22.転送キュー作成先 RD エリア名を必要に応じて入力します。
- 23.[ 順序保証 ] チェックボックスを必要に応じてチェックします。
- 24.定義内容を確認して、Eclipse のメニューから [ ファイル ] - [ 保管 ] を選択し、定義内容を保存します。

### 5.3.4 サービスアダプタ（MDB（DB キュー））を定義する

MDB（DB キュー）に対するアダプタの定義について説明します。MDB（DB キュー）に対するアダプタは、サービスアダプタ定義画面で定義できます。

#### （1）キュー名称の取得

サービス部品が MDB（DB キュー）の場合、サービス部品のキュー名称を開発者から取得する必要があります。キュー名称は、サービスアダプタを定義するときに使用します。

#### （2）サービス部品の電文フォーマット

MDB（DB キュー）の場合、サービス部品の電文フォーマットを作成して、設定する必要があります。

サービス部品の電文フォーマットの作成方法については、「4.3.4 サービス部品電文を作成する（MDB（WS-R または DB キュー）の場合）」を参照してください。

#### （3）データ変換

サービスリクエストの電文の形式と、サービスアダプタが呼び出すサービス部品の電文の形式が異なる場合、データ変換が必要です。

データ変換を行う場合、サービスアダプタ定義画面で標準電文にサービスリクエストから入力される電文を設定し、データ変換定義画面でデータ変換（マッピング）を定義します。

データ変換の詳細については、「7. データ変換の定義」を参照してください。

#### （4）サービスアダプタ定義画面での操作

MDB（DB キュー）に対するアダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。  
サービスアダプタ定義画面の表示方法については、「5.3.1(5) サービスアダプタ定義画面の表示」を参照してください。
2. サービス部品制御情報の [ サービス名 ], [ サービス ID ], および [ 最大インスタンス数 ] を必要に応じて編集します。

3. サービス部品制御情報の [ アドレス ] を指定します。  
[ アドレス ] には、共用キューを使用する場合の登録先キュー名を入力します。共用キューを使用する場合の登録先キュー名については、マニュアル「Cosminexus Reliable Messaging」を参照してください。
4. サービス部品制御情報の [ オペレーション ] のドロップダウンリストから、編集したいオペレーションを選択します。
5. オペレーション情報に表示されている内容を確認します。
6. 要求電文について次に示す操作をします。  
標準電文のフォーマットを指定する場合  
手順 7. ~ 10. を実施して、手順 11. に進みます。  
標準電文のフォーマットを指定しない場合  
手順 11. に進みます。
7. 標準電文の [ 使う ] チェックボックスをチェックします。
8. 標準電文の [ フォーマット ID ] を指定します。
9. 標準電文の [ ... ] ボタンをクリックして、[ 電文フォーマット ] に標準電文のフォーマットを指定します。  
なお、外部の XML スキーマを参照している電文フォーマットを設定する場合、必ずルートスキーマに当たるファイルを設定してください。ルートスキーマから参照される外部の XML スキーマファイルは、自動的に取り込まれます。  
指定できる電文フォーマットの形式については、「2.5.5 XML スキーマの適用範囲」を参照してください。
10. 標準電文の [ 表示 ] ボタンをクリックします。  
標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。
11. サービス部品電文の [ フォーマット ID ] を指定します。
12. サービス部品電文の [ ... ] ボタンをクリックして、[ 電文フォーマット ] にサービス部品電文のフォーマットを指定します。
13. サービス部品電文の [ 表示 ] ボタンをクリックします。  
サービス部品電文のフォーマットが表示されます。必要に応じて指定したサービス部品電文のフォーマットを確認してください。
14. 定義内容を確認して、Eclipse のメニューから [ ファイル ] - [ 保管 ] を選択し、定義内容を保存します。

### 5.3.5 DB アダプタを定義する

DB アダプタの定義について説明します。DB アダプタは、サービスアダプタ定義画面で定義できます。

## 5. アダプタの定義

### (1) サービス部品の電文フォーマット

DB アダプタの場合、WSDL が生成されないため、サービス部品の電文フォーマットも自動生成されません。そのため、サービス部品の電文フォーマット (DB アダプタ用の XML フォーマット定義ファイル) を作成して、設定する必要があります。

DB アダプタ用の XML フォーマット定義ファイルの作成方法については、「4.3.5 サービス部品電文を作成する (DB アダプタ電文の場合)」を参照してください。

### (2) データ変換

サービスリクエストの電文の形式と、DB アダプタ電文の形式が異なる場合、データ変換が必要です。

要求電文のデータ変換を行う場合、サービスアダプタ定義画面で標準電文にサービスリクエストから入力される電文を設定し、データ変換定義画面でデータ変換 (マッピング) を定義します。応答電文のデータ変換を行う場合、サービス部品から返される電文に標準電文を設定し、データ変換定義画面でデータ変換 (マッピング) を定義します。

データ変換の詳細については、「7. データ変換の定義」を参照してください。

### (3) 独自定義ファイルの作成と設定

サービス部品の利用者は、次のファイルを作成し、サービスアダプタ定義画面で設定する必要があります。

#### (a) SQL オペレーション定義ファイル

SQL オペレーション定義ファイルの作成については、「4.3.5(1) SQL オペレーション定義ファイルを作成する」を参照してください。

#### (b) アプリケーション統合属性ファイル

アプリケーション統合属性ファイルは、サンプルファイルをテキストエディタなどで編集して作成します。アプリケーション統合属性ファイルは「cscadapter\_property.xml」という名称で、次の場所に格納されています。

```
<Cosminexusのインストールディレクトリ>\¥CSC¥config¥dba¥templates
```

なお、アプリケーション統合属性ファイルの詳細については、マニュアル「Cosminexus アプリケーション設定操作ガイド」を参照してください。

アプリケーション統合属性ファイルに設定する項目について、次の表に示します。

表 5-3 アプリケーション統合属性ファイルの設定項目

タグ	説明	設定内容
<hitachi-application-all-property>	ルートタグ	-

タグ	説明	設定内容
<ejb-jar>	EJB に関する情報の定義の開始タグ	-
<hitachi-session-bean-property>	SessionBean の属性の定義の開始タグ	-
<resource-ref>	リソース参照についての定義の開始タグ	-
<res-ref-name>	リソース参照名	SQL オペレーション定義ファイルの <DB_NAME> タグに指定したデータベース参照名。
<res-sharing-scope>	参照したリソースを共有するかどうかの指定	DB アダプタが参照する DB Connector を共有するかどうかの指定。
<linked-to>	データソース表示名	DB Connector のプロパティ定義 (Display name) に指定したリソースアダプタの名称。
</resource-ref>	リソース参照についての定義の終了タグ	-
<session-runtime>	runtime の定義の開始タグ	-
<stateless>	stateless の定義の開始タグ	-
<pooled-instance>	プール内のインスタンスについての定義の開始タグ	-
<minimum>	プール内のインスタンスの最小数	DB アダプタのインスタンスの最小数。
<maximum>	プール内のインスタンスの最大数	DB アダプタのインスタンスの最大数。
</pooled-instance>	プール内のインスタンスについての定義の終了タグ	-
</stateless>	stateless の定義の終了タグ	-
</session-runtime>	runtime の定義の終了タグ	-
</hitachi-session-bean-property>	SessionBean の属性の定義の終了タグ	-
</ejb-jar>	EJB に関する情報の定義の終了タグ	-
</hitachi-application-all-property>	ルートタグ	-

( 凡例 )

- : 変更できません。

**!** 注意事項

表 5-2 で示した設定・変更できる項目以外は、サンプルファイルとして提供したアプリケーション統合属性ファイルを設定・変更しないでください。

#### (4) サービスアダプタ定義画面での操作

DB アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。  
サービスアダプタ定義画面の表示方法については、「5.3.1(5) サービスアダプタ定義画面の表示」を参照してください。
2. サービス部品制御情報の [ サービス名 ] および [ サービス ID ] を必要に応じて編集します。
3. サービス部品制御情報の [ 追加 ] ボタンをクリックして、オペレーションを追加します。
4. オペレーション情報の [ 通信モデル ] のドロップダウンリストから、「同期」または「非同期」を選択します。
5. 要求電文について、手順 6. から手順 17. までを実施します。
6. 次に示す操作をします。  
標準電文のフォーマットを指定する場合  
手順 7. ~ 10. を実施して、手順 11. に進みます。  
標準電文のフォーマットを指定しない場合  
手順 11. に進みます。
7. 標準電文の [ 使う ] チェックボックスをチェックします。
8. 標準電文の [ フォーマット ID ] を指定します。
9. 標準電文の [ ... ] ボタンをクリックして、[ 電文フォーマット ] に標準電文のフォーマットを指定します。  
指定できる電文フォーマットの形式については、「2.5.5 XML スキーマの適用範囲」を参照してください。なお、外部の XML スキーマを参照している電文フォーマットは指定できません。
10. 標準電文の [ 表示 ] ボタンをクリックします。  
標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。
11. サービス部品電文の [ フォーマット ID ] を指定します。
12. サービス部品電文の [ ... ] ボタンをクリックして、[ 電文フォーマット ] にサービス部

品電文のフォーマットを指定します。

[電文フォーマット] に指定できる電文種別は、XML 電文だけです。

13. サービス部品電文の [表示] ボタンをクリックします。

サービス部品電文のフォーマットが表示されます。必要に応じて指定したサービス部品電文のフォーマットを確認してください。

14. 次に示す操作をします。

標準電文の [使う] チェックボックスをチェックした場合  
手順 15. ~ 17. を実施して、手順 18. に進みます。

標準電文の [使う] チェックボックスをチェックしない場合  
手順 18. に進みます。

15. データ変換定義のファイル名を入力します。

16. [編集] ボタンをクリックします。

データ変換定義画面が表示されます。初めて定義する場合は、[ルート要素選択] ダイアログが表示されます。

17. 標準電文とサービス部品電文の内容をマッピングします。

18. [通信モデル] で「同期」を選択した場合、応答電文についても同様に、手順 6. から手順 17. までを実施します。

19. サービスアダプタ定義 (詳細) タブをクリックします。

20. カスタムアダプタ (EJB-JAR ファイル) の名称を確認します。ユーティリティクラス (JAR ファイル) の設定は不要です。

21. 独自定義ファイルの [追加] ボタンをクリックして、次のファイルを追加します。

- SQL オペレーション定義ファイル
- アプリケーション統合属性ファイル

22. 定義内容を確認して、Eclipse のメニューから [ファイル] - [保管] を選択し、定義内容を保存します。

### 5.3.6 アダプタ定義での設定一覧

サービスアダプタおよび DB アダプタの定義で使用する「サービスアダプタ定義画面」での設定について説明します。

## 5. アダプタの定義

### ! 注意事項

- [ 電文フォーマット ID ] には「format」で始まる名称を指定しないでください。「format」で始まる名称を指定した場合、システムが内部で利用している ID と重複し、エラーが発生するおそれがあります。
- サービスアダプタ定義画面で指定する電文フォーマット定義ファイルが、外部の XML スキーマを参照している電文フォーマットの場合、必ずルートスキーマに当たるファイルを設定してください。ルートスキーマから参照される外部の XML スキーマファイルは、自動的に取り込まれます。
- サービスアダプタ定義画面で指定する電文フォーマット定義ファイルは、「2.5.5 XML スキーマの適用範囲」に示す条件を満たしている必要があります。スキーマの条件の詳細については、「2.5.5 XML スキーマの適用範囲」を参照してください。

### (1) サービスアダプタの場合

各サービス部品のサービスアダプタ定義画面での設定を次の表に示します。

表 5-4 各サービス部品のサービスアダプタ定義画面での設定

分類	項目	Web サービス	Session Bean	MDB (WS-R)	MDB (DB キュー)
		同期	同期	非同期	非同期
基本 画面	サービス名				
	サービス ID				
	サービス種別	×	×	×	×
	アドレス	×			
	最大インスタンス数				
	サービスクラス名	×	×	×	×
	オペレーション				
オペレ ーション情 報	オペレーション名	×	×	×	×
	通信モデル	×	×	×	×
要求電文	使う (チェックボックス)				
	標準フォーマット ID	1	1	1	1
	標準電文フォーマット	1	1	1	1
	サービス部品フォーマット ID				
	サービス部品電文フォーマット				

分類	項目	Web サービス	Session Bean	MDB (WS-R)	MDB (DB キュー)	
		同期	同期	非同期	非同期	
	データ変換定義	1	1	1	1	
	応答電文	使う(チェックボックス)			-	-
		標準フォーマットID	2	2	-	-
		標準電文フォーマット	2	2	-	-
		サービス部品フォーマットID			-	-
		サービス部品電文フォーマット			-	-
		データ変換定義	2	2	-	-
	フォルト電文	フォルト名	3	-	-	-
		フォルト電文フォーマット	3	-	-	-
	詳細画面	Web サービス制御情報	クライアント定義ファイル		-	-
			使う(チェックボックス)		-	-
			ユーザ名	4	-	-
パスワード			4	-	-	
パスワード(確認)			4	-	-	
SessionBean 制御情報		クライアント定義ファイル	-		-	-
		ユーザ定義クラス(JARファイル)	-		-	-
MDB 制御情報		JMS メッセージ種別	-	-		-
		リモート呼出し(ラジオボタン)	-	-	5	-
		ローカル呼出し(ラジオボタン)	-	-	5	-
		最大メッセージ数	-	-	6	-
		宛先 URL	-	-	6	-
	ベーシック認証(チェックボックス)	-	-	7	-	
	ユーザ名	-	-	4	-	

## 5. アダプタの定義

分類	項目	Web サービス	Session Bean	MDB (WS-R)	MDB (DB キュー)
		同期	同期	非同期	非同期
	パスワード	-	-	4	-
	パスワード(確認)	-	-	4	-
	転送キュー作成先 RD エリア名	-	-	7	-
	順序保証(チェックボックス)	-	-	7	-

### (凡例)

- : 必ず設定します。
  - : 任意で設定します。
  - : 場合によっては必ず設定します。
  - ×: 表示されている内容を確認します。
  - : 該当しません。
- 注 1 要求電文の [使う](チェックボックス) にチェックした場合, 必ず設定します。
- 注 2 応答電文の [使う](チェックボックス) にチェックした場合, 必ず設定します。
- 注 3 サービス部品電文のフォルト電文があるときにだけ設定します。
- 注 4 次の場合に必ず設定します。
- Web サービス制御情報の場合  
ベーシック認証の [使う](チェックボックス) にチェックした場合, 必ず設定します。
  - MDB 制御情報の場合  
[リモート呼出し](ラジオボタン) を選択し, ベーシック認証の [使う](チェックボックス) にチェックした場合, 必ず設定します。
- 注 5 [リモート呼出し] または [ローカル呼出し] のどちらか一方のラジオボタンを必ず選択します。
- 注 6 [リモート呼出し] ラジオボタンを選択した場合, 必ず設定します。
- 注 7 [リモート呼出し] ラジオボタンを選択した場合にだけ, 任意で設定できます。

### ! 注意事項

サービスアダプタ定義画面では, 入力項目の前後に半角スペースや全角スペースを指定できません。ただし, MDB (WS-R) の詳細画面の [転送キュー作成先 RD エリア名] には, 末尾に半角スペースを指定できます。

## (2) カスタムアダプタの場合

カスタムアダプタ (DB アダプタ) を定義する場合の, サービスアダプタ定義画面での設定を次の表に示します。

表 5-5 カスタムアダプタのサービスアダプタ定義画面での設定

分類	項目	DB アダプタ		
		同期	非同期	
基本画面	サービス部品制御 情報	サービス名		
		サービス ID		
		サービス種別	×	×
		アドレス	×	×
		最大インスタンス数 <sup>1</sup>		
		サービスクラス名	×	×
		オペレーション		
	オペレーション情報	オペレーション名	×	×
		通信モデル	2	3
	要求電文	使う (チェックボックス)		
		標準フォーマット ID	4	4
		標準電文フォーマット	4	4
		サービス部品フォーマット ID		
		サービス部品電文フォーマット <sup>6</sup>		
		データ変換定義	4	4
	応答電文	使う (チェックボックス)		-
		標準フォーマット ID	5	-
		標準電文フォーマット	5	-
		サービス部品フォーマット ID		-
		サービス部品電文フォーマット <sup>6</sup>		-
		データ変換定義	5	-
フォルト電文	フォルト名	-	-	
	サービス部品電文フォーマット	-	-	

## 5. アダプタの定義

分類	項目	DB アダプタ		
		同期	非同期	
詳細画面	カスタムアダプタ 制御情報	カスタムアダプタ (EJB-JAR ファイル)	×	×
		ユーティリティクラス (JAR ファイル)	-	-
		独自定義ファイル		

### (凡例)

- : 必ず設定します。
- : 任意で設定します。
- : 場合によっては必ず設定します。
- × : 表示されている内容を確認します。
- : 該当しません。

注 1 サービスアダプタ定義画面で設定した値は適用されません。最大インスタンス数は、アプリケーション統合属性ファイルの要素「hitachi-application-all-property/ejb-jar/hitachi-session-bean-property/session-runtime/stateless/pooled-instance/maximum」で指定した値が設定されます。

注 2 「同期」を設定します。

注 3 「非同期」を設定します。

注 4 要求電文の [使う] (チェックボックス) にチェックした場合、必ず設定します。

注 5 応答電文の [使う] (チェックボックス) にチェックした場合、必ず設定します。

注 6 電文種別が XML 電文の電文フォーマットを設定します。

## 5.4 アダプタの保存

---

サービスアダプタ定義画面で編集中のサービスアダプタや DB アダプタの内容を、編集途中や編集終了時に保存できます。アダプタの内容はリポジトリに保存されます。

データ変換定義で生成したデータ変換定義ファイルは、サービスアダプタ定義画面と同時に保存されます。

アダプタを保存する手順を次に示します。

### 方法 1

Eclipse のメニューから、[ ファイル ] - [ 保管 ] を選択します。

### 方法 2

Eclipse のメニューから、[ ファイル ] - [ すべて保管 ] を選択します。

### 方法 3

サービスアダプタ定義画面で、[ Ctrl ] キー + [ S ] キーを押します。

なお、アダプタを定義する画面を閉じてアダプタの定義を終了する場合に、アダプタが保存されていないときは、リソース保管ダイアログが表示されます。リソース保管ダイアログでは、保存されていないアダプタの定義を保存できます。

### 注意事項

不正なデータが入力されている場合、アダプタを保存できないことがあります。その場合、表示されているメッセージに従って対処してください。

---

## 5.5 アダプタの編集

---

保存したサービスアダプタや DB アダプタの内容を変更できます。アダプタの内容を変更するには、サービスアダプタ定義画面を表示して、内容を編集します。サービスアダプタ定義画面は、ツリービューのサービス定義一覧から、該当するアダプタを選択し、ダブルクリックすると表示されます。

アダプタの内容の定義方法については、「5.3 アダプタの内容の定義」を参照してください。

## 5.6 アダプタの検証

作成したサービスアダプタやDBアダプタの内容が妥当かどうか検証できます。

必要なサービスアダプタ定義やDBアダプタ定義がない場合、または定義の関係が正しくない場合は正常に動作しません。そのため、実行環境で実行する前に、作成したサービスアダプタやDBアダプタに必要な項目がそろっているか、それらの関係が正しいかどうかを検証できます。検証は、必要に応じて、任意のタイミングで検証できます。

### 5.6.1 検証方法

検証する手順を次に示します。

1. ツリービューのサービス定義一覧で、右クリックします。  
サービス一覧のポップアップメニューが表示されます。
2. ポップアップメニューから「検証」を選択します。  
検証結果がコンソールビューに表示されます。

また、パッケージングを実施した場合も、自動的に検証されます。

なお、検証する前にアダプタが保存されていないときは、リソース保管ダイアログが表示されます。リソース保管ダイアログでは、保存されていないアダプタの定義を保存できます。

### 5.6.2 検証内容の表示

検証結果として、メッセージがコンソールビューに表示されます。メッセージに従って、修正します。

表示されるメッセージの種類を次の表に示します。

表 5-6 メッセージの種類

種類	説明
エラー	次のどちらかの場合に表示されます。 <ul style="list-style-type: none"> <li>• 定義の内容が間違っている。</li> <li>• 正しい構文で定義しているが、実行できない定義をしている。</li> </ul>
警告	実行時にエラーになるおそれがある定義をしている場合に表示されます。
情報	追加情報が表示されます。

## 5.7 アダプタの削除

---

作成したサービスアダプタや DB アダプタを削除できます。

### (1) 削除方法

削除する手順を次に示します。次に示す二つの方法があります。

#### 方法 1

1. ツリービューのサービス定義一覧で、サービスを選択します。
2. [ Delete ] キーを押します。  
削除を確認するダイアログボックスが表示されます。
3. [ はい ] ボタンをクリックします。  
指定したサービスアダプタや DB アダプタが削除されます。

#### 方法 2

1. ツリービューのサービス定義一覧で、サービスを選択して右クリックします。  
サービス一覧のポップアップメニューが表示されます。
2. ポップアップメニューから「削除」を選択します。  
削除を確認するダイアログボックスが表示されます。
3. [ はい ] ボタンをクリックします。  
指定したサービスアダプタや DB アダプタが削除されます。

### (2) 削除できないアダプタ

次に示す場合、サービスアダプタや DB アダプタは削除できません。

- すでに配備定義されている場合
- ビジネスプロセスのサービス呼出アクティビティから呼び出す対象として設定されている場合

# 6

## ビジネスプロセスの定義

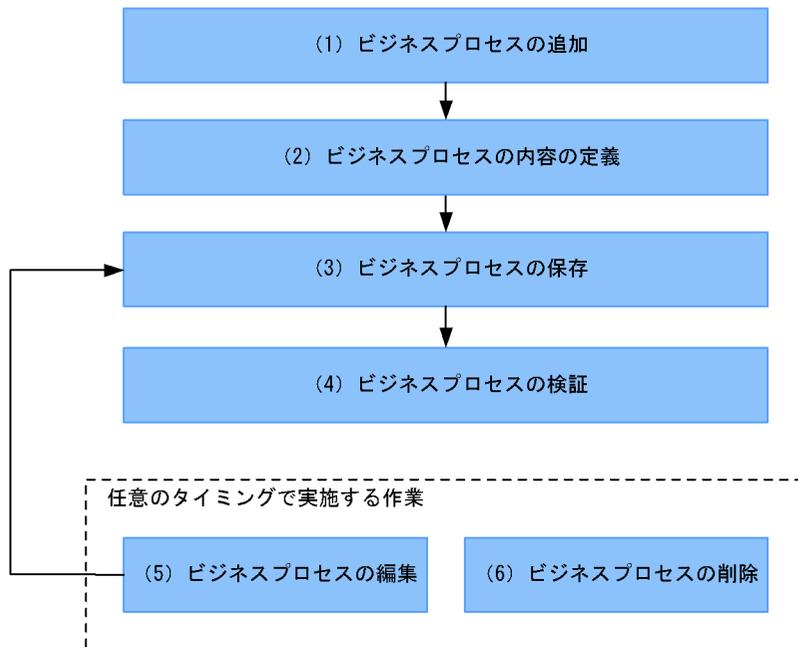
この章では、ビジネスプロセスの定義について説明します。

- 
- 6.1 定義の流れ
  - 6.2 ビジネスプロセスの追加
  - 6.3 ビジネスプロセスの内容の定義
  - 6.4 アクティビティの配置と連結
  - 6.5 変数・相関セットの定義
  - 6.6 アクティビティの定義
  - 6.7 ビジネスプロセスの保存
  - 6.8 ビジネスプロセスの編集
  - 6.9 ビジネスプロセスの検証
  - 6.10 ビジネスプロセスの削除
-

## 6.1 定義の流れ

ビジネスプロセスの定義の流れを次の図に示します。

図 6-1 ビジネスプロセスの定義の流れ



ビジネスプロセスの定義に関する作業について次に説明します。

### (1) ビジネスプロセスの追加

ビジネスプロセスを追加するには、次のどちらかの方法を利用します。

#### 新規に追加する場合

ウィザードを利用してビジネスプロセスを新規に追加します。

ウィザードを利用してビジネスプロセスを新規に追加するには、未定義のビジネスプロセスを追加する方法と、あらかじめ BPMN を用いたモデリングツールで作成した BPEL ファイルをインポートして、ビジネスプロセスの定義に変換して追加する方法があります。

ビジネスプロセスを新規に追加する方法の詳細については、「6.2.1 ビジネスプロセスを新規に追加する」を参照してください。

#### 定義済みのビジネスプロセスを利用する場合

定義済みのビジネスプロセスを複製してビジネスプロセスを追加できます。定義済みのビジネスプロセスを複製すると、同じ定義内容のビジネスプロセスが追加されます。複製したビジネスプロセスの定義内容を編集することもできます。

定義済みのビジネスプロセスを利用してビジネスプロセスを追加する方法の詳細は、

「6.2.2 定義済みビジネスプロセスを利用してビジネスプロセスを追加する」を参照してください。

## (2) ビジネスプロセスの内容の定義

ビジネスプロセス定義画面でビジネスプロセスの内容を定義します。

定義の方法および定義の内容の詳細については、「6.3 ビジネスプロセスの内容の定義」を参照してください。ビジネスプロセス定義画面の詳細については、「11.2.3 ビジネスプロセス定義画面」を参照してください。

## (3) ビジネスプロセスの保存

編集したビジネスプロセスの定義情報は、必要に応じてリポジトリに保存する必要があります。

ビジネスプロセスの定義情報の保存方法と保存されるタイミングについては、「6.7 ビジネスプロセスの保存」を参照してください。

## (4) ビジネスプロセスの検証

定義したビジネスプロセスの整合性を検証します。ビジネスプロセスの定義中や定義が終わったあとなど、任意のタイミングで検証できます。

検証方法の詳細については、「6.9 ビジネスプロセスの検証」を参照してください。

## (5) ビジネスプロセスの編集

定義済みのビジネスプロセスは、必要に応じて定義内容を編集して変更できます。

ビジネスプロセスの編集方法の詳細については、「6.8 ビジネスプロセスの編集」を参照してください。

## (6) ビジネスプロセスの削除

不要になったビジネスプロセスは、必要に応じて削除できます。

ビジネスプロセスの削除方法については、「6.10 ビジネスプロセスの削除」を参照してください。

## 6.2 ビジネスプロセスの追加

---

ビジネスプロセスを追加します。

ビジネスプロセスを新規に追加する場合、ビジネスプロセス追加ウィザードを利用します。また、すでに作成されているビジネスプロセスを複製して、新しいビジネスプロセスを追加することもできます。

### 6.2.1 ビジネスプロセスを新規に追加する

ビジネスプロセス追加ウィザードを利用してビジネスプロセスを新規に追加する方法には、次の二つがあります。

- 未定義のビジネスプロセスの新規追加  
未定義のビジネスプロセスを新規に追加する方法です。この方法で新規に追加されるビジネスプロセスは、何も定義されていない状態になります。
- BPEL ファイルをインポートして新規追加  
あらかじめ BPMN を用いたモデリングツールで BPEL ファイルを作成します。その後、BPEL ファイルをインポートしてビジネスプロセスの定義に変換して新規に追加する方法です。この方法で新規に追加されるビジネスプロセスには、基となった BPEL ファイルの定義情報が引き継がれます。

それぞれの方法について、次に説明します。

#### (1) 未定義のビジネスプロセスを新規に追加する

何も定義されていないビジネスプロセスを新規に追加する方法を次に示します。

1. ツリービューのサービス定義一覧で、右クリックして [ ビジネスプロセス追加 ] を選択します。  
ビジネスプロセス定義を追加するためのダイアログが表示されます。
2. ビジネスプロセス名を入力し、ステータスを永続化するかどうか ( ビジネスプロセスの実行状態をデータベースに保存するかどうか ) を選択します。  
ビジネスプロセス名 ( ビジネスプロセスのサービス名 ) は、64 バイト以内で入力します。
3. [ 終了 ] ボタンをクリックします。  
必要なファイルが作成され、リポジトリに保存されます。  
ビジネスプロセスが正常に保存されると、ビジネスプロセス定義画面が表示されます。

#### (2) BPEL ファイルをインポートしてビジネスプロセスを新規に追加する

あらかじめモデリングツールで BPEL ファイルを作成し、BPEL ファイルをインポートしてビジネスプロセスを新規に追加する方法について説明します。

## (a) BPEL ファイルの作成

モデリングツールで BPEL1.1 に準拠した BPEL ファイルを作成します。BPEL ファイルの作成方法については、利用するモデリングツールのドキュメントを参照してください。

BPEL ファイルに定義した要素および属性の内容と、ビジネスプロセス定義の内容との関連については、「付録 B 上流ツールとの連携で使用する BPEL のサポート範囲」を参照してください。

## (b) ビジネスプロセスの新規追加

BPEL ファイルをインポートして、ビジネスプロセスを新規に追加する方法を次に示します。

1. ツリービューのサービス定義一覧で、右クリックして [ ビジネスプロセス追加 ] を選択します。  
ビジネスプロセス定義を追加するためのダイアログが表示されます。
2. ビジネスプロセス名を入力し、ステータスを永続化するかどうか ( ビジネスプロセスの実行状態をデータベースに保存するかどうか ) を選択します。  
ビジネスプロセス名 ( ビジネスプロセスのサービス名 ) は、64 バイト以内で入力します。
3. [ BPEL ファイル ] の [ インポートする ] のチェックボックスをチェックします。
4. [ ファイル名 ] にインポートする BPEL ファイルを指定します。  
[ ... ] ボタンをクリックして、インポートする BPEL ファイルを選択することもできます。
5. アクティビティの配置方法を選択します。
6. [ 終了 ] ボタンをクリックします。  
BPEL ファイルの定義が、ビジネスプロセスの定義に変換されます。変換が終わり、必要なファイルが作成されると、リポジトリに保存されます。  
ビジネスプロセスが正常に保存されると、ビジネスプロセス定義画面が表示されます。表示されるビジネスプロセス定義画面のキャンパスには、手順 5. で選択した方法でアクティビティが配置されます。

### ! 注意事項

BPEL ファイルをインポートする場合、次の点に注意が必要です。

#### 変換されない BPEL の定義

Cosminexus サービスプラットフォームの開発環境でサポートしていない BPEL の定義は、無視されるか、無操作アクティビティに変換されます。

BPEL ファイルに定義した要素および属性の内容と、ビジネスプロセス定義の内容との関連については、「付録 B 上流ツールとの連携で使用する BPEL のサポート範囲」を参照してください。

なお、次の要素や属性の配下に定義されている要素や属性はすべて無視されます。

BPEL ファイルで使用しないでください。

- pick
- compensationHandler
- eventHandler

#### BPEL ファイルでメッセージ型の変数が定義されていた場合

モデリングツールでメッセージの名称を定義し、BPEL ファイルにメッセージ型の変数が定義されていた場合、BPEL ファイルをインポートして作成したビジネスプロセスでは、変数の種別が文字列型 (string) に変換されます。

この場合、インポート後に変数の種別をメッセージ型 (messageType) に変更してください。変数の定義については、「6.5.1(6) 変数の定義方法」を参照してください。

#### サービス呼出アクティビティが定義されていた場合

BPEL ファイルをインポートしてビジネスプロセスを作成するとき、BPEL ファイルの invoke 要素はサービス呼出アクティビティに変換されます。invoke 要素の portType 属性のローカル名が呼び出すサービス名に、operation 属性がオペレーション名に変換されます。

ただし、リポジトリに該当するサービスやオペレーションがない場合には、サービス名やオペレーション名は変換されません

サービス名やオペレーション名が変換されなかった場合、BPEL ファイルをインポートしたあと、呼び出す対象のサービスアダプタやビジネスプロセスを作成して、サービス名およびオペレーション名を割り当てる必要があります。

#### BPEL ファイルに長い名称のアクティビティが多数定義されていた場合

配置方法で横方向が指定されていて、BPEL ファイルに長い名称のアクティビティが多数定義されている場合、インポート後にキャンパスの端でアクティビティが重なってしまうことがあります。その場合は、アクティビティを配置し直してください。

### (3) ステータスの永続化の設定

ビジネスプロセスには、データベースにプロセスの実行状態および実行履歴を永続化するビジネスプロセスと、永続化しないビジネスプロセスの 2 種類があります。永続化とは、プロセスの実行状態および実行履歴をデータベースに記録することです。

永続化するビジネスプロセスは、プロセスの実行状態と実行履歴の記録があるため、障害が発生した場合に、プロセスの実行の進捗度がわかり、プロセスを再実行できます。

永続化しないビジネスプロセスは、プロセスの実行状態と実行履歴の記録を採らないた

め、高性能を実現する必要があるビジネスプロセスの場合に設定します。

永続化するビジネスプロセスではできることが、永続化しないビジネスプロセスではできないといった制約事項があります。永続化しないビジネスプロセスは、次の条件を満たす場合に定義できます。

1. 非同期の受付アクティビティを含まない。
2. 非同期のサービスアダプタに対して、サービス呼び出すサービス呼出アクティビティを含まない。
3. 応答アクティビティよりあとにアクティビティ（分岐（終了）アクティビティおよび終了アクティビティを除く）が存在しない。  
例を次に示します。

（例1）「スコープ」の中と外で「受付」と「応答」を使用する場合は、スコープ内の「応答1」のあとに、スコープ外の「受付2」を実行することになるため、永続化しないビジネスプロセスとして定義できません。



（例2）「並列処理」の中に「受付」と「応答」を使用する場合は、次に示すように、並列処理が必ず同時に実行して終了するとは限らないため、永続化しないビジネスプロセスとして定義できません。

- 「応答1」を実行したあとに「受付2」、「サービス呼出2」、または「応答2」のどれかを実行する。
- 「応答2」を実行したあとに「受付1」、「サービス呼出1」、または「応答1」のどれかを実行する。



（例3）「並列処理」の中に「サービス呼出」だけを使用する場合は、永続化しないビジネスプロセスとして定義できます。

## 6. ビジネスプロセスの定義



(例4)「分岐」の中に「応答」を使用する場合は、「応答1」と「応答2」のどちらか一方だけを実行することになるため、永続化しないビジネスプロセスとして定義できません。



4. プロセスインスタンスの再実行を使用しない（障害発生時は、サービスリクエストからサービス呼び出し要求（ビジネスプロセスの呼び出し）を再度実行する）。

### 6.2.2 定義済みビジネスプロセスを利用してビジネスプロセスを追加する

すでに定義されているビジネスプロセスを複製して、ビジネスプロセスを追加できます。

1. ツリービューのサービス定義一覧で、複製するビジネスプロセスを選択して右クリックします。  
サービス一覧のポップアップメニューが表示されます。
2. ポップアップメニューから [複製] を選択します。  
選択したビジネスプロセスの複製が作成されます。複製されたビジネスプロセス名（ビジネスプロセスのサービス名）やサービス ID などは、システム内で競合しないよう、自動的に別の情報が割り当てられます。

#### 注

アダプタおよびビジネスプロセスを追加するときに、自動的に割り当てられるシステム内で一意の番号です。サービス ID は変更できますが、すでに使用されているサービス ID には変更できません。

#### ！ 注意事項

ビジネスプロセスを複製すると、複製したビジネスプロセスに含まれるユーザ定義受付もあわせて複製されます。ただし、ユーザ定義受付だけを複製することはできません。

## 6.3 ビジネスプロセスの内容の定義

---

ビジネスプロセスの内容はビジネスプロセス定義画面で定義します。

ビジネスプロセス定義画面は、ツリービューのサービス定義一覧で表示されたサービス内のビジネスプロセスをダブルクリックすると表示されます。

ビジネスプロセス定義画面の詳細については、「11.2.3 ビジネスプロセス定義画面」を参照してください。

ビジネスプロセス定義画面では、アクティビティと変数・相関セットを定義します。

### (1) アクティビティ

ビジネスプロセスは、複数のアクティビティを連結して定義します。

アクティビティとは、ビジネスプロセスの構成要素となる部品であり、処理の構造も表現したものです。複数のアクティビティを連結して、ビジネスプロセスの処理フローを定義します。ビジネスプロセスは、アクティビティを画面上で配置・連結して定義します。

アクティビティの配置、連結方法の詳細については、「6.4 アクティビティの配置と連結」を参照してください。

### (2) 変数・相関セット

ビジネスプロセスには、変数と相関セットを定義できます。

#### (a) 変数

ビジネスプロセスの処理の中で、条件式の項の値などに変数値を利用する場合、アクティビティの定義で変数を宣言します。

ビジネスプロセス内で利用できる変数の種類、アクティビティとの関係および定義方法については、「6.5.1 変数の定義」を参照してください。

#### (b) 相関セット

相関セットとは、サービスリクエストから HCSC サーバを介してサービス部品に送信する要求電文を一意に識別するために利用する文字列です。

相関セットとアクティビティとの関係や定義方法については、「6.5.2 相関セットの定義」を参照してください。

#### ！ 注意事項

ビジネスプロセスの定義では、定義中に入力値が正しいかどうかのチェックはされません。入力値が正しいかどうかは、ビジネスプロセスの検証でチェックされます。入力値が正しくない場合、検証でエラーになります。

## 6.4 アクティビティの配置と連結

---

ビジネスプロセス定義画面では、アクティビティをキャンバスに配置し、連結してビジネスプロセスの処理フローを定義します。また、フォルト処理を設定する場合には、アクティビティ間の連結にフォルト処理を定義します。

以降の項では、アクティビティの配置方法、アクティビティの連結方法およびフォルト処理の定義方法について説明します。

### 6.4.1 アクティビティを配置する

ビジネスプロセス定義画面でのアクティビティの配置方法を次に示します。

1. パレットの基本アクティビティまたは構造アクティビティの中から、キャンバスに配置するアクティビティをクリックします。  
クリックしたアクティビティが選択された状態になります。
2. キャンバス上の適当な場所をクリックします。  
選択したアクティビティがキャンバス上に配置されます。配置されたアクティビティは、ドラッグ&ドロップで場所を移動できます。

なお、各アクティビティで定義できる内容の詳細については、「6.6 アクティビティの定義」を参照してください。

#### ポイント

---

モデリングツールで作成した BPEL ファイルをインポートしてビジネスプロセスを追加した場合、ビジネスプロセスを追加するときに選択した配置方法によって、キャンバスに初期表示されるアクティビティの配置が異なります。

「縦方向」を選択した場合

上から下に処理が流れる形でアクティビティが初期表示されます。

「横方向」を選択した場合

左から右に処理が流れる形でアクティビティが初期表示されます。

必要に応じてドラッグ&ドロップしたり、アクティビティを整列したりしてください。ビジネスプロセス定義画面（キャンバス）での操作については、「11.8 ビジネスプロセス定義画面の操作」を参照してください。

---

### 6.4.2 アクティビティを連結する

キャンバスに配置された二つのアクティビティを連結すると、ビジネスプロセスの実行順序が定義されます。

#### (1) 連結の種類

アクティビティの連結には、次の種類があります。

- コネクションでの連結  
通常のアクティビティの連結に利用します。
- フォルトコネクションでの連結  
アクティビティでフォルトが発生したときのためのフォルト処理を定義する場合に利用します。フォルト処理の定義の詳細については、「6.4.3 フォルト処理を定義する」を参照してください。
- リンクコネクションでの連結  
並列処理アクティビティを利用した並列処理で、リンクを定義する場合に利用します。並列処理とリンクについては、「6.6.13 並列処理アクティビティ」を参照してください。

連結方法によって、キャンバス上の連結線の表示は異なります。連結方法と連結線の関係を次の表に示します。

表 6-1 連結の画面表示

連結方法	表示される連結線	線の色	線の種類	終端
コネクションでの連結		黒	実線	三角の矢印
フォルトコネクションでの連結		赤	破線	三角の矢印
リンクコネクションでの連結		青	点線	線の矢印

## (2) 連結の設定方法

アクティビティを連結する方法を次に示します。

1. パレットの [コネクション], [リンク], または [フォルト] をクリックします。  
コネクション, フォルトコネクション, またはリンクコネクションが選択された状態になります。  
この状態で、アクティビティにマウスポインタを合わせると、そのアクティビティを連結元に設定できる場合、アクティビティの背景色が変わります。
2. キャンバスに配置したアクティビティのうち、連結元となるアクティビティをクリックします。  
連結元となるアクティビティが設定されます。  
この状態で、ほかのアクティビティにマウスポインタを合わせると、そのアクティビティを連結先に設定できる場合、アクティビティの背景色が変わります。
3. キャンバスに配置したアクティビティのうち、連結先となるアクティビティをクリックします。  
連結元となるアクティビティと連結先となるアクティビティが連結されます。

## (3) 連結の変更方法

すでに設定されている連結を変更するには、連結線の始点または終点をドラッグし、ほかのアクティビティにドロップします。

#### (4) 連結線の折り曲げ

連結線は、折り曲げて表示できます。アクティビティの連結が複雑になり、キャンバス上の連結線を整理する場合などに利用できます。連結線を折り曲げる方法について次に示します。

1. キャンバス上で連結線を選択します。  
連結線の始点と終点の間に折り曲げられる個所（バンドポイント）が表示され  
ます。
2. バンドポイントに選択ツールのポイントを合わせます。  
十字の矢印（）が表示されます。
3. 矢印を任意の位置にドラッグ&ドロップします。  
連結線がバンドポイントから折れ曲がります。  
バンドポイントで折り曲げた連結線では、次の個所もバンドポイントとして利用でき  
ます。
  - 連結線の始点とバンドポイントの中間
  - バンドポイントと連結線の終点の中間
  - バンドポイントとバンドポイントの中間

#### (5) アクティビティの連結条件

アクティビティと利用する連結の種類によって、アクティビティを連結元または連結先として利用できるかどうか異なります。各アクティビティの連結元または連結先としての利用可否について、次の表に示します。

表 6-2 連結元または連結先として利用できるアクティビティ

アクティビティ	コネクション		フォルトコネクション		リンクコネクション	
	連結元	連結先	連結元	連結先	連結元	連結先
開始		-	-	-	-	-
受付			-			
応答			-			
サービス呼出						
Java 呼出						
データ変換			-			
代入			-			
無操作			-			
フォルト送出	-		-			
スコープ						
繰り返し			-			
分岐（開始）			-			

アクティビティ	コネクション		フォルトコネクション		リンクコネクション	
	連結元	連結先	連結元	連結先	連結元	連結先
分岐（終了）			-	-	-	-
並列処理（開始）			-			
並列処理（終了）			-	-	-	-
終了	-		-	-	-	-

（凡例）

- ：連結元または連結先としてアクティビティを利用できます。
- ：連結元または連結先としてアクティビティを利用できません。

また、アクティビティをコネクションで連結するとき、あるアクティビティへの遷移元となるアクティビティを遷移元アクティビティ、あるアクティビティからの遷移先となるアクティビティを遷移先アクティビティといいます。

コネクション（フォルトコネクションおよびリンクコネクションは除く）で連結できる遷移元アクティビティおよび遷移先アクティビティの数は、アクティビティごとに異なります。

アクティビティごとの連結できる遷移元・遷移先アクティビティの数を次の表に示します。

表 6-3 連結できる遷移元・遷移先アクティビティの数

アクティビティ	連結できる遷移元アクティビティの数	連結できる遷移先アクティビティの数
開始	0	1
受付	1	1
応答	1	1
サービス呼出	1	1
Java 呼出	1	1
データ変換	1	1
代入	1	1
無操作	1	1
フォルト送出	1	0
スコープ	1	1
繰り返し	1	1
分岐（開始）	1	1 以上
分岐（終了）	1 以上	1
並列処理（開始）	1	1 以上

## 6. ビジネスプロセスの定義

アクティビティ	連結できる遷移元アクティビティの数	連結できる遷移先アクティビティの数
並列処理（終了）	1 以上	1
終了	1	0

### ！ 注意事項

ビジネスプロセスで最初に行える基本アクティビティは、インスタンスを生成する受付アクティビティだけです。

#### (6) コネクション使用時の注意事項

- リンクコネクションでリンク先の遷移条件が「あり」に設定した場合、遷移条件を設定してください。
- リンクコネクションは、循環しないように設定してください。
- リンク名はビジネスプロセス内で一意になるよう設定してください。
- リンクコネクションは、フォルト処理外からフォルト処理内へ張らないでください。
- コネクションで連結したアクティビティ同士を近づけると、連結線の矢印の向きが設定と逆に見える場合があります。連結したアクティビティ同士は、ある程度の距離をとって配置してください。

### 6.4.3 フォルト処理を定義する

アクティビティでフォルトが発生したときに何らかの処理を実行する場合、フォルト処理用のアクティビティを配置してフォルト処理を実行します。

フォルト処理は、フォルトが発生するアクティビティと、フォルト処理を行うアクティビティを連結して定義します。連結にはフォルトコネクションを利用します。

#### (1) フォルト処理の定義手順

フォルト処理の定義手順を次に示します。

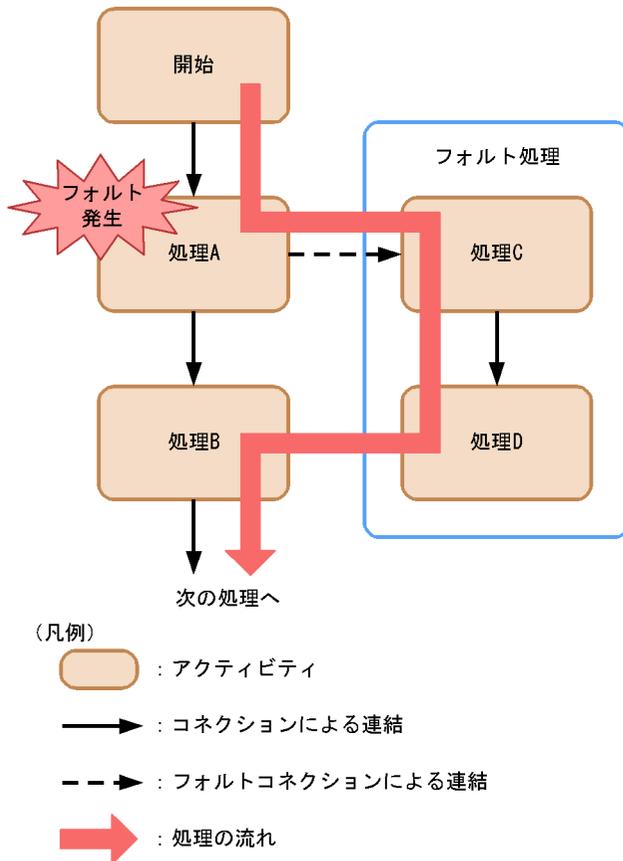
1. フォルトが発生するアクティビティとフォルト処理を行うアクティビティをキャンバスに配置し、詳細を定義します。  
フォルト処理を行うアクティビティは複数定義できます。  
アクティビティの配置方法については、「6.4.1 アクティビティを配置する」を参照してください。各アクティビティの定義方法の詳細については、「6.6 アクティビティの定義」を参照してください。
2. フォルトコネクションを利用してフォルトが発生するアクティビティとフォルト処理を行うアクティビティを連結します。  
アクティビティの連結方法については、「6.4.2(2) 連結の設定方法」を参照してください。また、フォルトコネクションを利用できるアクティビティについては、「6.4.2(5) アクティビティの連結条件」を参照してください。

3. 連結線をダブルクリックします。  
[フォルト処理の割当]ダイアログが表示されます。  
[フォルト処理の割当]ダイアログの入力・表示内容の詳細については、「11.4.5 フォルト処理の割当ダイアログ」を参照してください。なお、フォルト処理に割り当てられる変数はメッセージ型だけです。
4. フォルト処理の条件を定義します。  
フォルトの対象となる変数と、フォルトが発生した場合に処理する連結先のアクティビティを、それぞれ [割当変数] および [遷移先] ドロップダウンリストで指定します。  
定義していないフォルトが発生した場合のフォルト処理を設定する場合、[割当変数] ドロップダウンリストで「catch-all」を選択します。なお、フォルトの発生するアクティビティが Java 呼出アクティビティの場合、定義できるのは「catch-all」のフォルト処理だけです。
5. [OK] ボタンをクリックします。

次の図に示すように、フォルトが発生した場合、フォルト処理を実行したあとに、フォルトが発生したアクティビティの次のアクティビティに処理が移ります。

## 6. ビジネスプロセスの定義

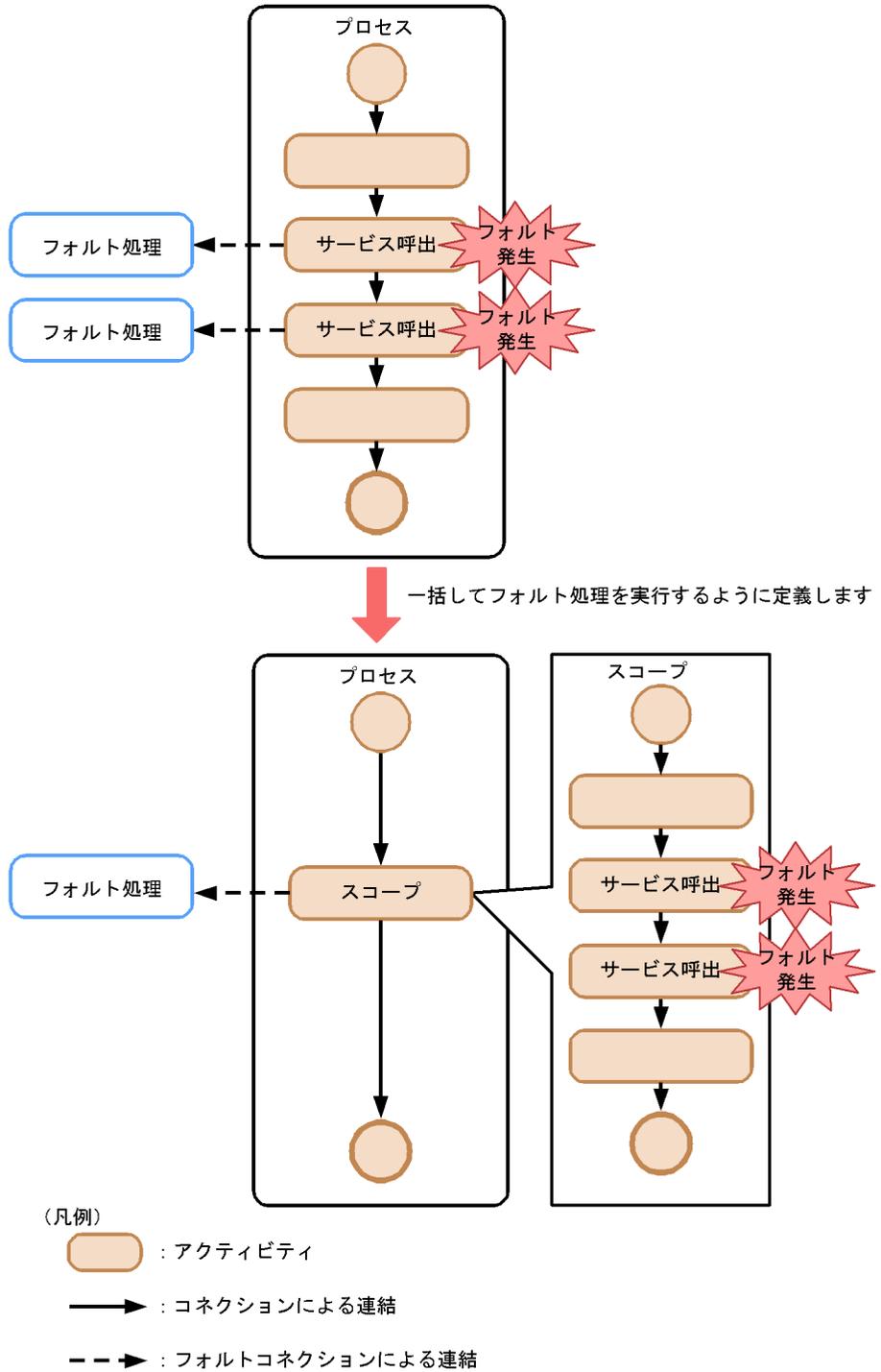
図 6-2 フォルト処理を実行したあとの処理の流れ



### (2) 一括してプロセスのフォルト処理を実行する場合の定義

プロセス内にフォルトが返るおそれのあるアクティビティが複数あった場合、アクティビティごとにフォルト処理を定義するのではなく、一括してプロセスのフォルト処理を実行できます。その場合、次の図に示すようにスコープアクティビティを使って、プロセス内の処理を一つのまとまりにして、フォルト処理を定義します。

図 6-3 スコープアクティビティを使って定義したフォルト処理



### (3) ビジネスプロセスのフォルト処理の定義で使用する電文フォーマット

ビジネスプロセスのフォルト処理の定義で使用する電文フォーマットについて説明します。

#### (a) 複数のフォルト名に対応する電文フォーマット

複数のユーザ定義例外を throw する Java プログラムから、SOAP アプリケーション開発支援機能 (Java2WSDL コマンド) を使用して Document スタイルの WSDL を生成し、これを入力としてアダプタを定義すると、アダプタ定義画面の [フォルト電文] に表示される複数のフォルト名に対応する電文フォーマットは、同一のものとなります。この場合、電文フォーマットは、次の例に示すような複数のルート要素を含む形式になっています。

複数のフォルト名に対応する電文フォーマット (ルートスキーマ) の例

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://service"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://service"
xmlns:tns2="http://data.service"
xmlns:tns3="http://fault.service">
  <xsd:import namespace="http://data.service"
    schemaLocation="cscformat2.xsd"/>
  <xsd:import namespace="http://fault.service"
    schemaLocation="cscformat3.xsd"/>
  <xsd:element name="order">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="in0" type="tns2:OrderData"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="orderResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="orderReturn" type="tns2:OrderResult"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="StockShortageFault"
    type="tns3:StockShortageFault"/>
  <xsd:element name="InvalidCustomerFault"
    type="tns:InvalidCustomerFault"/>
  <xsd:complexType name="InvalidCustomerFault">
    <xsd:sequence>
      <xsd:element name="CustomerName" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

上記の例では、「StockShortageFault」と「InvalidCustomerFault」という二つのフォルトに対応したフォルト電文が、「cscformat1.xsd」という一つの電文フォーマットで扱われています。

SOAP アプリケーション開発支援機能 (Java2WSDL コマンド) を使用して rpc スタイルの WSDL を生成し、フォルトを扱えるように、「4.3.2 サービス部品電文を作成する (Web サービスの場合)」の注意事項の手順で修正した WSDL を基にアダプタを定義した場合も、同様の形式となります。

発生するフォルトに応じて処理を切り分けるために、この電文フォーマットをフォルトごとに別々の変数として定義し、[フォルト処理の割当] ダイアログで二つの catch の割当変数として定義した場合、二つの catch に割り当てられている変数のフォーマットは同一になってしまいます。そのため、どちらのフォルトが発生しても先に定義されている catch に適合してしまい、要因に応じたフォルト処理が呼び出されません。

フォルト電文がこのようなフォーマットになっている場合は、次に示す方法でビジネスプロセスを定義して対処してください。

#### (b) 定義方法

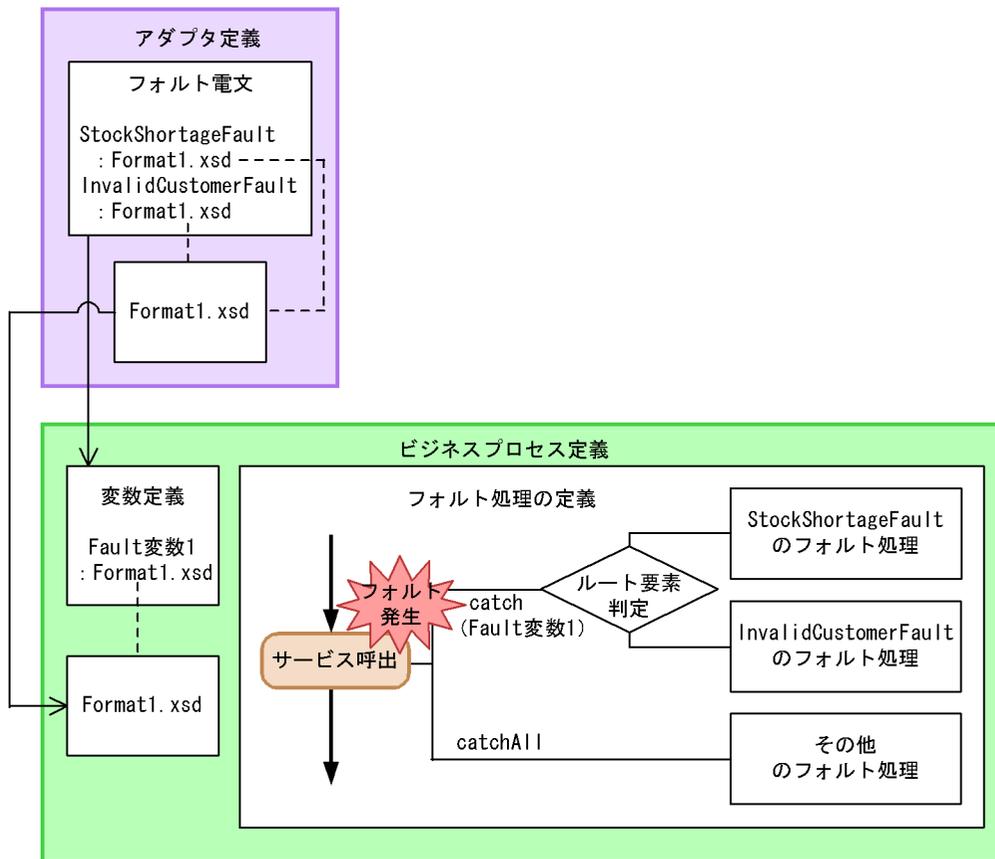
フォルトごとに変数を定義しないで、電文フォーマットが同一のフォルトに対して共通となる変数を一つ定義します。フォルト処理の遷移先も一つにし、[フォルト処理の割当] ダイアログで、共通となる変数を割り当てます。ただし、想定外のフォルト処理のために、別途 catchAll の遷移先を設定することもできます。

フォルト処理の遷移先には分岐アクティビティを配置し、ルートの要素名などで発生したフォルトを切り分ける分岐条件を記述します。

共通となる変数を割り当てた場合の概念図を次に示します。

## 6. ビジネスプロセスの定義

図 6-4 共通となる変数を割り当てた場合の概念図



(凡例)

-  : アクティビティ
-  : コネクションによる連結
-  : アダプタ定義からビジネスプロセス定義へのつながり

ルート要素名を取得して一致判定 ( 図中の「ルート要素判定」) を実施する分岐条件の記述例を次に示します。

```
csc:getVariableData('Fault変数1', 'local-name(/
*)')="StockShortageFault"
```

「Fault 変数 1」の部分にフォルト共通変数の変数名, 「StockShortageFault」の部分にルート要素名を指定します。

## 6.5 変数・相関セットの定義

---

### 6.5.1 変数の定義

ビジネスプロセスを定義する場合、変数には、文字、メッセージに含まれる数値や条件式の項となる数値、またはメッセージを入れることができます。変数の値は、データベースに記録できるため、実行履歴として管理できます。

#### (1) 変数の種類

変数にはグローバル変数とローカル変数の2種類があります。

- グローバル変数

グローバル変数は、一つのプロセスインスタンス内であればどこからでも参照できます。また、プロセスが完了しても変数の値はデータベースに記録されているため、ビジネスプロセス全体の進行状況を分析できます。変数名は、プロセス内で一意にします。

- ローカル変数

ローカル変数は、宣言したスコープ内で参照できます。宣言しているスコープが終了すると、変数の値もデータベースから削除されるため、プロセス実行中の条件判定など一時的に値を参照するために利用できます。変数名は、スコープ内で一意にします。スコープが異なれば同じ変数名を宣言できます。ただし、変数を参照する場合に、参照個所で最も内側に近いスコープで宣言された変数を参照します。

#### (2) 変数の型

変数には次に示す型の値を定義できます。変数は、定義した型の値だけを保持できます。

- boolean 型：真偽（true または false）
- numeric 型：数値（64 ビット浮動小数）
- string 型：文字列
- message 型：変数に代入する値は、変数に関連づけられたフォーマット定義に従います。値が XML データの場合、XML スキーマでフォーマットを定義します。

#### (3) 変数を割り当てられるアクティビティ

次の表に示すアクティビティの項目について変数を割り当てられます。

## 6. ビジネスプロセスの定義

表 6-4 変数を割り当てられるアクティビティと項目

アクティビティ	項目	割り当てられる内容	備考
受付アクティビティ	要求電文	サービスリクエスタから受け取る要求電文を参照するメッセージ型の変数	関連セットの指定にも利用されます。
応答アクティビティ	応答電文	サービスリクエスタへの応答電文またはフォルト電文を参照するメッセージ型の変数	
サービス呼出アクティビティ（同期）	要求電文	同期サービスを呼び出す際の要求電文を参照するメッセージ型の変数	-
	応答電文	呼び出した同期サービスから受け取る応答電文を参照するメッセージ型の変数	
	フォルト電文	呼び出した同期サービスでフォルトが発生した場合に受け取る応答電文を参照するメッセージ型の変数	
サービス呼出アクティビティ（非同期）	要求電文	非同期サービスを呼び出す際の要求電文を参照するメッセージ型の変数	関連セットの指定にも利用されます。
Java 呼出アクティビティ	引数	ユーザの作成した Java クラスを呼び出す際の引数となる値を参照する変数	-
	戻り値	呼び出した Java クラスからの戻り値を参照する変数	-
データ変換アクティビティ	変換元データ	データ変換アクティビティでデータ変換を行う際の変換元データを参照するメッセージ型の変数で、複数指定が可能	-
	変換先データ	データ変換アクティビティでデータ変換を行う際の変換先データを参照するメッセージ型の変数	-
代入アクティビティ	コピー元	代入アクティビティでデータのコピーを行う際の、コピー元となるデータを参照する変数	-
	コピー先	代入アクティビティでデータのコピーを行う際の、コピー先となるデータを参照する変数	-
フォルト送出アクティビティ	フォルト電文	フォルト送出アクティビティでフォルトを送出する際に、フォルト電文を参照するメッセージ型の変数	-
繰り返しアクティビティ	繰り返し条件	繰り返しのループ変数や終端条件式の項となる値を参照する変数	-
分岐アクティビティ	分岐条件	分岐条件の式の項となる値を参照する変数	-

（凡例）

- : 該当しません。

### （４）変数の表示とパスの取得

変数・関連セット一覧ダイアログの [ 表示 ] ボタンをクリックすると、変数の表示ダイ

アログが表示されます。変数と一致した XML スキーマをツリービューとして表示できます。ツリービューで表示された変数を選択すると、[ 選択パス ] に選択した変数のパスがルート要素からの絶対パスで表示されます。

また、変数を選択して、右クリックし、[ パスの取得 ] を選択すると、ルート要素からの絶対パスを取得できます。取得したパスは、任意の個所に貼り付けることができます。

#### 注意事項

変数の表示ダイアログで表示しようとしている変数の XML スキーマで、要素が再帰になっている場合、2 階層目以降の要素は表示されません。

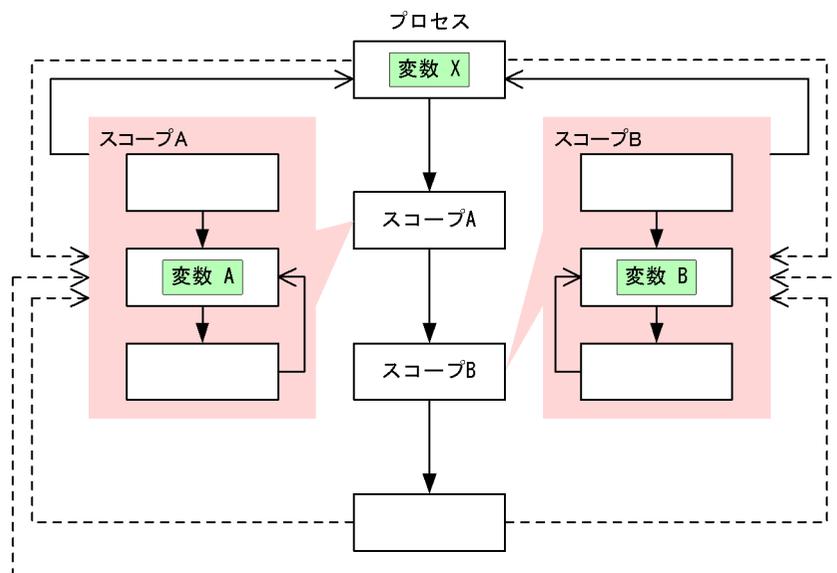
### (5) 変数を使用する場合の注意事項

#### (a) スコープアクティビティを使用している場合の変数の使用

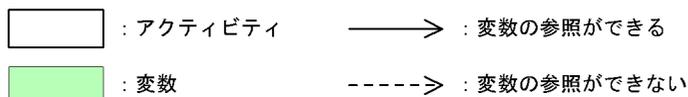
ビジネスプロセス内またはスコープ内のアクティビティで使用する変数を追加したり、編集したりできます。このとき、外側のスコープで宣言した変数は、内側のスコープから参照できますが、内側のスコープで宣言した変数はスコープ外からは参照できません。また、スコープが異なる場合は、同じ変数名を宣言できます。次に示す図の場合、スコープ A やスコープ B はプロセス（外側）の変数を参照できますが、プロセス（外側）からはスコープ A やスコープ B の変数は参照できません。

## 6. ビジネスプロセスの定義

図 6-5 スコープと変数の関係



(凡例)



### (b) フォルト処理を定義する場合の変数の使用

スコープアクティビティにフォルト処理を定義する場合、[フォルト処理の割当]ダイアログで割り当て変数として使用する変数を選択します。選択できる変数は、フォルトの対象となるスコープ(スコープアクティビティを含む)、またはその外側のスコープで定義された変数です。ただし、[フォルト処理の割当]ダイアログで選択した変数と同じ名前の変数が、フォルトの対象のスコープでも定義されている場合、[フォルト処理の割当]ダイアログで選択した変数ではなく、フォルトの対象となるスコープで定義された変数が実行時に使用され、意図しない動作をする場合があります。したがって、スコープアクティビティにフォルト処理を定義する場合、[フォルト処理の割当]ダイアログで選択した変数と同じ名前の変数が、フォルトの対象となるスコープに定義されていないことを確認してください。

### (c) 変数の定義で設定する電文フォーマット

#### 電文フォーマットのファイル名

変数の定義時には、同じファイル名で異なる XML スキーマの電文フォーマットである定義ファイルを複数の変数で設定できます。しかし、実行時には、どれか一つの XML スキーマが使用されるため、意図しない動作をする場合があります。したがって、変数の定義で設定する電文フォーマットのファイル名と XML スキーマは、必ず 1 対 1 になるように設定してください。

## 外部のXMLスキーマの参照を含む電文フォーマットの設定

外部のXMLスキーマを参照している電文フォーマットを設定する場合、必ずルートスキーマに当たるファイルを設定してください。ルートスキーマから参照される外部のXMLスキーマファイルは、自動的に取り込まれます。

## 電文フォーマットの形式

変数に使用する電文フォーマットは「2.5.5 XMLスキーマの適用範囲」に示す条件を満たしている必要があります。条件の詳細については、「2.5.5 XMLスキーマの適用範囲」を参照してください。

## (d) 変数の定義情報の変更

変数の定義情報は、変数の使用個所によって、変更または削除できない場合があります。変数の定義情報を変更または削除できないケースを次の表に示します。なお、次の表に記載されていない個所で使用されている変数については、いつでも変数の定義情報を変更または削除できます。

表 6-5 変数の定義情報を変更または削除できないケース

変数の使用個所	条件	削除の可否	変更の可否			
			変数名	型	電文フォーマット	部分
受付アクティビティ 応答アクティビティ	<ul style="list-style-type: none"> <li>バージョンアップしたビジネスプロセスである。</li> <li>バージョンアップ前に定義された受付・応答アクティビティである。</li> </ul>	×	×	×	×	
データ変換アクティビティ	データ変換定義ファイルが定義済みである。	×	×	×	×	
関連セット	<ul style="list-style-type: none"> <li>バージョンアップしたビジネスプロセスである。</li> <li>バージョンアップ前のビジネスプロセスの受付・応答アクティビティで使用されている関連セットである。</li> </ul>	×	×	×	×	<sup>2</sup>

## (凡例)

- ：変更または削除できます。
- ×：変更または削除できません。

## 注 1

電文フォーマットの変更とは、ダイアログで新しい電文フォーマットを選択することです。もし内容が同じ電文フォーマットを選択しても、変更として扱われます。

## 6. ビジネスプロセスの定義

### 注 2

使用している部分が変更されていなければ、変更できます。部分の変更は、部分名、指定式および型の変更です。順番の変更は含みません。

### ！ 注意事項

変数の定義情報を変更すると、その変数を割り当てた個所に、再度割り当てが必要になる場合があります。

例えば、あるアクティビティに変数 X を割り当て、その後、その変数 X の名前を変数 Y に変更した場合、次のどちらかを行ってください。

- アクティビティへ再度変数を割り当てる。
- 再度変数 X を定義する。

## (6) 変数の定義方法

### (a) 新規に変数を定義する場合

新規に変数を定義する手順を次に示します。なお、変数は、繰り返しアクティビティ内のキャンバスでは定義できません。

1. 次のどちらかの方法で、変数・相関セット一覧ダイアログを表示します。
  - ビジネスプロセス定義画面のキャンバス上の [変数・相関セット] アイコンをダブルクリックします。
  - 次に示すアクティビティダイアログの [編集] ボタンをクリックします。
    - 受付アクティビティ
    - 応答アクティビティ
    - サービス呼出アクティビティ
    - Java 呼出アクティビティ
    - データ変換アクティビティ
    - 代入アクティビティ
    - フォルト送出アクティビティ
2. 一覧から「変数一覧」を選択します。
3. 変数名を入力します。
4. 変数の種別をドロップダウンリストから選択します。
5. 必要に応じて、次のどちらかの操作をします。

変数の種別がメッセージ型の場合で、呼び出すサービス部品またはユーザ定義受付の要求電文、応答電文、またはフォルト電文に設定されている電文フォーマットを、変数に設定する場合

[取込] ボタンをクリックし、[電文フォーマットの取込] ダイアログを表示します。[電文フォーマットの取込] ダイアログで、使用する電文フォーマットを指定します。

[電文フォーマットの取込] ダイアログの入力・表示項目の詳細については、「11.4.4 電文フォーマットの取込ダイアログ」を参照してください。

[ 電文フォーマットの取込 ] ダイアログでの操作の終了後、手順 6. に進みます。

変数の種別がメッセージ型の場合で、上記以外の場合

[ ... ] ボタンをクリックして、変数を使用する電文フォーマットの定義ファイルを設定します。

6. 変数の種別がメッセージ型の場合、[ 部分指定 ] チェックボックスにチェックします。
7. [ 行追加 ] ボタンをクリックし、部分名、指定式、および型を指定します。
8. [ 追加 ] ボタンをクリックします。  
変数一覧に、追加した変数が表示されます。
9. [ OK ] ボタンをクリックします。

(b) 電文フォーマットの定義ファイルを変更した場合

変数の定義で設定した電文フォーマットの定義ファイルを変更した場合、変更後の定義ファイルの内容は、次の手順で反映できます。

1. 変数・相関セット一覧ダイアログを表示します。  
変数・相関セット一覧ダイアログを表示する方法については、「6.5.1(6)(a) 新規に変数を定義する場合」の手順 1. を参照してください。
2. 次のどちらかの操作をします。

変数の種別がメッセージ型の場合で、呼び出すサービス部品またはユーザ定義受付の要求電文、応答電文、またはフォルト電文に設定されている電文フォーマットを、変数に設定する場合

[ 取込 ] ボタンをクリックし、[ 電文フォーマットの取込 ] ダイアログを表示します。[ 電文フォーマットの取込 ] ダイアログで、使用する電文フォーマットを指定します。

[ 電文フォーマットの取込 ] ダイアログの入力・表示項目の詳細については、「11.4.4 電文フォーマットの取込ダイアログ」を参照してください。

[ 電文フォーマットの取込 ] ダイアログでの操作の終了後、手順 6. に進みます。

変数の種別がメッセージ型の場合で、上記以外の場合

[ ... ] ボタンをクリックして、変数を使用する電文フォーマットの定義ファイルを設定します。

3. [ 更新 ] ボタンをクリックします。

## 6.5.2 相関セットの定義

相関セットは、サービスリクエストから送信された要求電文を一意に識別するために利用する文字列です。

### (1) 相関セットによるプロセスインスタンスの識別

ビジネスプロセスの実行時、HCSC サーバでは、受付アクティビティダイアログの [ イ

## 6. ビジネスプロセスの定義

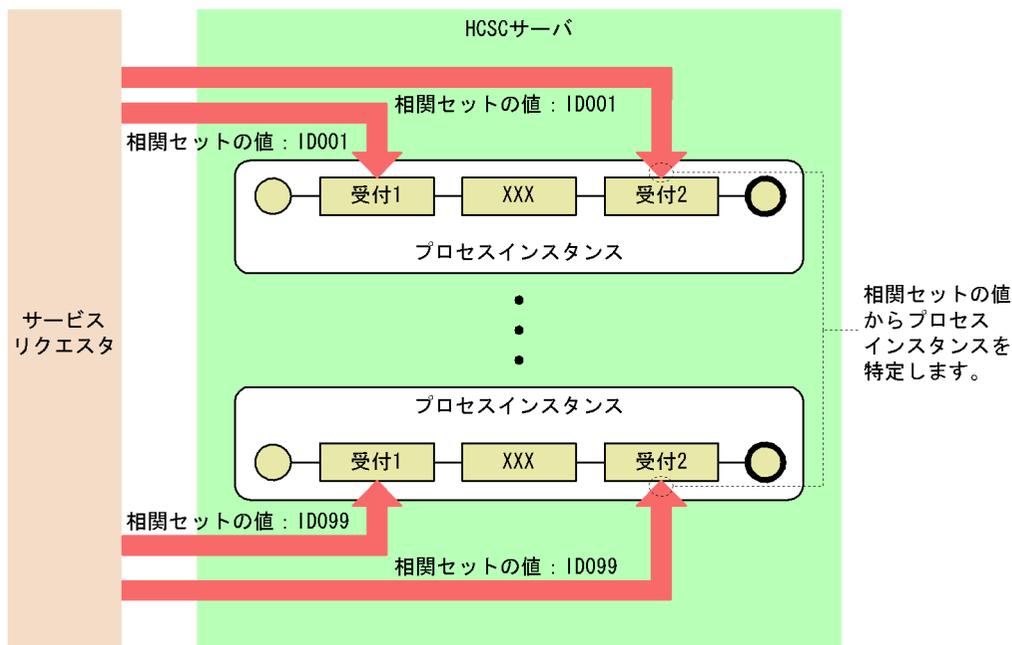
インスタンス生成]に [yes] を指定した受付アクティビティへのリクエストごとにプロセスインスタンスを生成します。

サービスリクエスタからメッセージを受け取れるプロセスインスタンスが HCSC サーバ上に複数ある場合、HCSC サーバはメッセージに含まれる関連セットの値を利用してプロセスインスタンスを特定します。

例えば、あるビジネスプロセスに、非同期にメッセージを受信する受付アクティビティが二つ定義されているとします。一つ目の受付アクティビティへのメッセージに含まれる関連セットの値と、二つ目の受付アクティビティへのメッセージに含まれる関連セットの値が同じ場合、HCSC サーバはこれら二つメッセージが同一のプロセスインスタンスへのリクエストであると特定できます。

関連セットによるプロセスインスタンスの識別の例を次の図に示します。

図 6-6 関連セットによるプロセスインスタンスの識別の例



(凡例)

➡ : サービスリクエスタからのリクエスト

このように、非同期にメッセージを送受信するビジネスプロセスを実現するには、送受信するメッセージからプロセスインスタンスを一意に特定するキーとなるように、関連セットを設計し、定義する必要があります。

関連セットは、メッセージに含まれる一つの部分または複数の部分の連結によって構成することができます。

ビジネスプロセスで関連セットを使用すると、関連セットの値をキーとして、プロセスインスタンスの実行履歴を検索できます。実行履歴を検索することで、特定のリクエストの実行状態を確認できます。プロセスインスタンスの実行履歴の詳細については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。

## (2) 関連セットを割り当てられるアクティビティ

ビジネスプロセスでは、受付アクティビティ、応答アクティビティおよびサービス呼出アクティビティに関連セットを設定できます。また、一つの電文には複数の関連セットを設定できます。

アクティビティごとに、関連セットの設定の要否、および関連セットの役割が異なります。

### (a) 受付アクティビティ、応答アクティビティの場合

受付アクティビティおよび応答アクティビティでは、インスタンスの生成の設定、割り当てた関連セットを初期化するかどうかの設定によって、関連セットの指定の要否および関連セットの内容が異なります。

インスタンスの生成の設定は、受付アクティビティダイアログで設定できます。また、関連セットを初期化するかどうかの設定は、割当関連セット群ダイアログで設定できます。

各ダイアログの詳細については、「11.4.6 受付アクティビティダイアログ」、「11.4.7 応答アクティビティダイアログ」、または「11.4.2 割当関連セット群ダイアログ」を参照してください。

表 6-6 関連セットの設定の要否と内容（受付・応答）

アクティビティ	インスタンスの生成	初期化の設定	設定の要否	最大設定数	設定した関連セットの内容
受付	yes	yes	任意	1	プロセスインスタンスを生成するとき、プロセスインスタンスを識別するための関連セットが、変数として定義されている受信メッセージから生成されます。
		no	設定不可	-	-
	no	yes	任意	1	生成済みのプロセスインスタンスを検索したあと、検索されたプロセスインスタンスに対する新たな関連セットが、変数として定義されている受信メッセージから生成されます。
		no	必須	1	指定した関連セットは、生成済みのプロセスインスタンスを検索するために、変数として定義されている受信メッセージ内の関連セットとして利用されます。
応答	-	yes	任意	1	サービスリクエストに応答を返すとき、変数として定義されている応答メッセージから新たな関連セットが生成されます。

## 6. ビジネスプロセスの定義

アクティビティ	インスタンスの生成	初期化の設定	設定の要否	最大設定数	設定した関連セットの内容
	-	no	任意	1	サービスリクエストに応答を返すとき、変数として定義されている応答メッセージ内の関連セットの値と、プロセスインスタンスの関連セットの値とが一致しているかどうかチェックされます。

(凡例)

- : 該当しません。

(b) サービス呼出アクティビティの場合

サービス呼出アクティビティでは、割り当てた関連セットのパターンの設定、および関連セットを初期化するかどうかの設定によって、関連セットの指定の要否および関連セットの内容が異なります。

割り当てた関連セットのパターンの設定、および関連セットを初期化するかどうかの設定は、割当関連セット群ダイアログで設定できます。

割当関連セット群ダイアログの詳細については、「11.4.2 割当関連セット群ダイアログ」を参照してください。

表 6-7 関連セットを割り当てられるアクティビティ (サービス呼出)

アクティビティ	パターンの設定	初期化の設定	設定の要否	最大設定数	設定した関連セットの内容
サービス呼出	out	yes	任意	1	サービス部品にメッセージを送信するとき、変数として定義されている送信メッセージから新たな関連セットが生成されます。
		no	任意	1	サービス部品にメッセージを送信するとき、変数として定義されている送信メッセージ内の関連セットの値と、プロセスインスタンスの関連セットの値とが一致しているかどうかチェックされます。
	in	yes	任意	1	サービス部品から応答を受けたとき、変数として定義されている応答メッセージから新たな関連セットが生成されます。
		no	任意	1	サービス部品から応答を受けたとき、変数として定義されている応答メッセージ内の関連セットの値と、プロセスインスタンスの関連セットの値とが一致しているかどうかチェックされます。

アクティビティ	パターンの設定	初期化の設定	設定の要否	最大設定数	設定した関連セットの内容
	out-in	yes	任意	1	送信時 サービス部品にメッセージを送信するとき、変数として定義されている送信メッセージから新たな関連セットが生成されます。 受信時 サービス部品から応答を受けたとき、変数として定義されている応答メッセージ内の関連セットの値と、プロセスインスタンスの関連セットの値とが一致しているかどうかチェックされます。
		no	任意	1	送信時 サービス部品にメッセージを送信するとき、変数として定義されている送信メッセージ内の関連セットの値と、プロセスインスタンスの関連セットの値とが一致しているかどうかチェックされます。 受信時 サービス部品から応答を受けたとき、変数として定義されている応答メッセージ内の関連セットの値と、プロセスインスタンスの関連セットの値とが一致しているかどうかチェックされます。

### (3) 関連セットを使用する場合の注意事項

#### (a) スコープアクティビティを使用している場合の関連セットの使用

ビジネスプロセス内またはスコープ内のアクティビティで使用する関連セットを、変数と同様に追加したり、編集したりできます。詳細は、「6.5.1(5)(a) スコープアクティビティを使用している場合の変数の使用」を参照してください。

#### (b) 関連セットの定義情報の変更

次のすべての条件に該当する場合、関連セットの定義情報の変更や削除はできません。

- バージョンアップしたビジネスプロセスの受付・応答アクティビティに設定されている関連セットである。
- 関連セットを使用している受付・応答アクティビティが、バージョンアップ前に定義されたものである。

また、関連セットの定義で使用されている変数の定義情報を変更できない場合もあります。詳細は、「6.5.1(5)(d) 変数の定義情報の変更」を参照してください。

**!** 注意事項

関連セットの定義情報を変更すると、その関連セットを割り当てた個所に、再度割り当てが必要になる場合があります。

例えば、あるアクティビティに関連セット A を割り当て、その後、その関連セット A の名前を関連セット B に変更した場合、次のどちらかを行ってください。

- アクティビティへ再度関連セットを割り当てる。
- 再度関連セット A を定義する。

## (4) 関連セットの定義方法

### (a) 関連セットの新規定義

関連セットを定義するには、あらかじめ変数および関連セットとして利用する部分の部分指定を定義しておきます。新規に関連セットを定義する手順を次に示します。なお、関連セットは、繰り返しアクティビティ内のキャンバスでは定義できません。

1. 次のどちらかの方法で、変数・関連セット一覧ダイアログを表示します。
  - ビジネスプロセス定義画面のキャンバス上の [ 変数・関連セット ] アイコンをクリックします。
  - 次に示すアクティビティダイアログの [ 編集 ] ボタンをクリックします。
    - 受付アクティビティ
    - 応答アクティビティ
    - サービス呼出アクティビティ
2. 一覧から「関連セット一覧」を選択します。
3. 関連セット名を入力します。
4. [ 行追加 ] ボタンをクリックし、[ 取得部 ] に利用変数および部分名を追加します。
5. 利用するメッセージ型の変数が複数ある場合は、利用変数および部分名をドロップダウンリストから選択します。

なお、関連セットに指定できるのは、メッセージ型の変数、かつメッセージ型以外の部分名です。
6. [ 追加 ] ボタンをクリックします。

関連セット一覧に、追加した関連セットが表示されます。
7. [ OK ] ボタンをクリックします。

### (b) アクティビティからの関連セットの設定

「表 6-7 関連セットを割り当てられるアクティビティ ( サービス呼出 )」に示すアクティビティから、関連セットを設定する手順を次に示します。

1. 各アクティビティのダイアログの [ 設定 ] ボタンをクリックします。

割当関連セット群ダイアログが表示されます。
2. アクティビティに割り当てる関連セットを [ 関連セット群 ] から選択します。

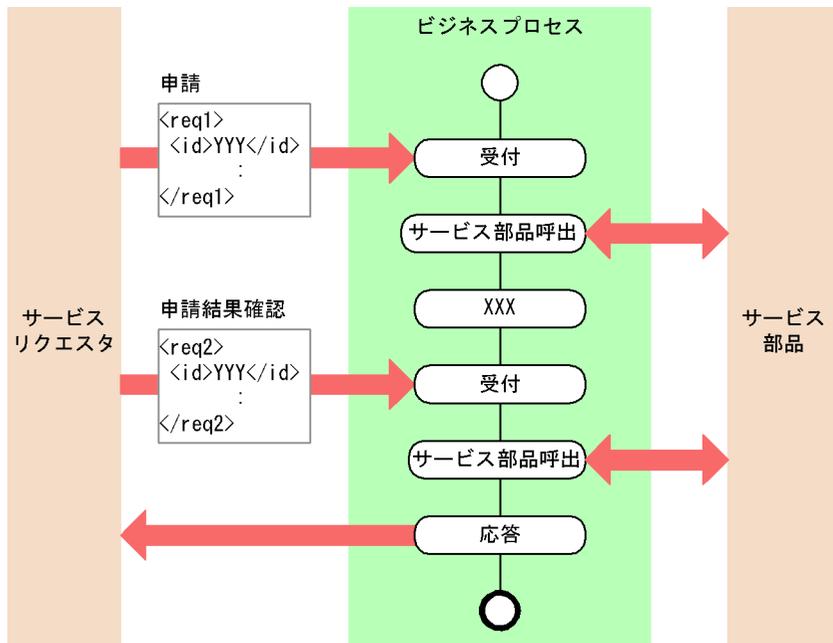
3. [ 相関セット群 ] がない場合は, [ 編集 ] ボタンをクリックします。
4. 変数・相関セット一覧ダイアログが表示されるので, 相関セットを追加します。
5. 割当相関セット群ダイアログで初期化するかどうかを指定します。

### (5) 相関セットを利用した定義の例

相関セットを利用するビジネスプロセスの定義について, 例を用いて説明します。

次の図に示すように, 申請とその結果を非同期で確認する申請結果確認のリクエストを受け付けるビジネスプロセスを例として利用します。

図 6-7 定義するビジネスプロセスの例



(凡例)

➡ : 電文の流れ      □ : 要求電文

上記の図に示したビジネスプロセスを定義する場合の変数, 相関セット, および受付アクティビティの定義内容を次に示します。

#### 変数および相関セットの定義

次の表に示すように, 利用する変数と相関セットを定義します。変数および相関セットは, [ 変数・相関セット一覧 ] ダイアログで定義します。[ 変数・相関セット一覧 ] ダイアログについては, 「11.4.1 変数・相関セット一覧ダイアログ」を参照してください。

## 6. ビジネスプロセスの定義

表 6-8 申請の要求電文に対応する変数の定義内容（例）

[ 変数・相関セット一覧 ] ダイアログ（変数情報の表示）での設定項目		設定する値
変数名		申請メッセージ
種別		messageType
電文フォーマット		申請メッセージ.xsd
部分指定	部分名	申請 ID
	指定式	/req1/id
	型	string

表 6-9 申請結果確認の要求電文に対応する変数の定義内容（例）

[ 変数・相関セット一覧 ] ダイアログ（変数情報の表示）での設定項目		設定する値
変数名		申請確認メッセージ
種別		messageType
電文フォーマット		申請確認メッセージ.xsd
部分指定	部分名	申請 ID
	指定式	/req2/id
	型	string

表 6-10 相関セットの定義内容（例）

[ 変数・相関セット一覧 ] ダイアログ（相関セット情報の表示）での設定項目		設定する値
相関セット名		申請相関 ID
取得部	利用変数	申請メッセージ
	部分名	申請 ID

### 注

利用変数には「申請メッセージ」と「申請確認メッセージ」のどちらでも指定できます。

### 受付アクティビティの定義

次の表に示すように、受付アクティビティを定義します。受付アクティビティは受付アクティビティダイアログで定義します。受付アクティビティダイアログについては、「11.4.6 受付アクティビティダイアログ」を参照してください。

表 6-11 申請のリクエストを受け付ける受付アクティビティの定義内容(例)

[受付アクティビティ]ダイアログでの設定項目		設定する値
アクティビティ名		申請受付
オペレーション名		request
割当変数		申請メッセージ
割当関連セット群	関連セット名	申請関連 ID
	初期化	yes
通信モデル		非同期
インスタンス生成		yes

表 6-12 申請結果確認のリクエストを受け付ける受付アクティビティの定義内容(例)

[受付アクティビティ]ダイアログでの設定項目		設定する値
アクティビティ名		確認受付
オペレーション名		confirm
割当変数		申請確認メッセージ
割当関連セット群	関連セット名	申請関連 ID
	初期化	no
通信モデル		同期
インスタンス生成		no

## 6.6 アクティビティの定義

アクティビティは、キャンバス上に配置・連結するとともに、詳細内容の定義をする必要があります。

定義できるアクティビティと定義内容の概要を次の表に示します。

表 6-13 定義できるアクティビティ

アクティビティ	定義内容
開始	ビジネスプロセスの開始を表します。
受付	サービスリクエストからの要求電文を受け付けるためのインターフェースを定義します。
応答	サービスリクエストから同期受信した要求電文に対する応答またはフォルトを返すためのインターフェースを定義します。
サービス呼出	定義済みの HCSC コンポーネントへの要求電文の送信を定義します。
Java 呼出	専用インターフェースを実装した Java クラスの呼び出しを定義します。
データ変換	データ変換処理を定義します。
代入	次の処理を実行するための定義をします。 <ul style="list-style-type: none"> <li>• ある変数（基本型、メッセージ型）をほかの変数に代入する</li> <li>• ある変数の一部をほかの変数に代入する</li> <li>• 値（数値、文字列、真偽値）を作成し、変数に代入する</li> </ul>
無操作アクティビティ	実行しても何の動作も行わないことを定義します。
フォルト送出アクティビティ	上位のスコープアクティビティへのフォルトの通知を定義します。
スコープ	一つまたは複数のアクティビティから構成される処理フローを、ひとつの単位（スコープ）として定義します。
繰り返し	一つまたは複数のアクティビティを、条件が満たされるまで繰り返し処理するための定義をします。
分岐（開始）	条件式の結果によってビジネスプロセスの処理の分岐を定義します。分岐の開始個所と終了個所で定義が必要です。
分岐（終了）	
並列処理（開始）	処理フローを複数のシーケンスに分けて同時に実行する（処理フローの並列化）を定義します。 並列処理の開始個所と終了個所で定義が必要です。
並列処理（終了）	
終了	ビジネスプロセスの終了を表します。

### 注

ビジネスプロセスにステータスの永続化が設定されている場合、受付、応答、サービス呼出アクティビティの処理が、アクティビティの状態を確定するタイミング（トランザクションの開始と終了のタイミング）になります。

トランザクションの開始および終了のタイミングを次に示します。

表 6-14 トランザクションの開始および終了のタイミング

トランザクション	タイミング
開始	受付アクティビティでメッセージを受信したとき。
	同期サービス呼び出すサービス呼出アクティビティで、メッセージを送信する前にトランザクションを終了した直後。 なお、非同期サービス呼び出すサービス呼出アクティビティ場合、トランザクションの開始のタイミングにはなりません。
	応答アクティビティのあとのアクティビティを実行する直前。
	プロセスを再実行するとき。
終了	応答アクティビティでメッセージを応答する直前。
	同期サービス呼び出すサービス呼出アクティビティでメッセージを送信する直前。 なお、非同期サービス呼び出すサービス呼出アクティビティの場合、トランザクションの終了のタイミングにはなりません。
	すべてのアクティビティを実行し、ビジネスプロセスの処理を終了する直前。

各アクティビティの定義の詳細について、以降の項で説明します。

### 6.6.1 開始アクティビティ

ビジネスプロセスの開始を表すアクティビティです。キャンバス上に一つだけ配置されます。このアクティビティは削除できません。

なお、開始アクティビティには、定義する内容はありません。

### 6.6.2 受付アクティビティ

ビジネスプロセスがサービスリクエストからの要求電文を受け付けるためのインターフェースを定義するアクティビティです。一つのビジネスプロセス内にサービスリクエストに公開するオペレーションが複数ある場合、そのオペレーションの数だけ定義が必要です。

受付アクティビティは、[ 受付アクティビティ ] ダイアログで詳細を定義します。

#### (1) 定義手順

定義の手順を次に示します。

1. 受付アクティビティをキャンバスに配置します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
2. 次のどちらかの方法で [ 受付アクティビティ ] ダイアログを表示させます。

## 6. ビジネスプロセスの定義

- キャンパスの受付アクティビティをダブルクリックする
- キャンパスの受付アクティビティを選択して右クリックし、[設定]を選択する

[受付アクティビティ]ダイアログが表示されます。

### 3. [受付アクティビティ]ダイアログに必要な情報を入力します。

[受付アクティビティ]ダイアログの表示・入力内容の詳細については、「11.4.6 受付アクティビティダイアログ」を参照してください。

- [割当変数]に設定する変数の内容を編集する場合

[編集]ボタンをクリックします。表示される[変数・関連セット一覧]ダイアログで、変数の内容を編集できます。[変数・関連セット一覧]ダイアログの詳細については、「11.4.1 変数・関連セット一覧ダイアログ」を参照してください。

なお、該当するオペレーションを持つユーザ定義受付を定義している場合、ユーザ定義受付の電文フォーマットを変数に設定できます。設定は、[電文フォーマットの取込]ダイアログで行います。[電文フォーマットの取込]ダイアログは、[変数・関連セット一覧]ダイアログの[取込]ボタンをクリックすると表示されません。

[電文フォーマットの取込]ダイアログを表示する際、[受付アクティビティ]ダイアログでオペレーション名を設定しておく、ユーザ定義受付内にある該当するオペレーションの電文フォーマットが選択された状態になります。

[電文フォーマットの取込]ダイアログの詳細については、「11.4.4 電文フォーマットの取込ダイアログ」を参照してください。

- [割当関連セット群]を設定する場合

[設定]ボタンをクリックします。表示される[割当関連セット群]ダイアログで割当関連セットを設定します。[割当関連セット群]ダイアログの詳細については、「11.4.2 割当関連セット群ダイアログ」を参照してください。

### 4. [OK]ボタンをクリックします。

## (2) 定義時の注意事項

- アクティビティ名はビジネスプロセス(スコープ内も含む)内で一意になるように指定してください。
- アクティビティ名は64バイト以内にしてください。
- 必ずオペレーション名を設定してください。
- 同期の受付アクティビティの場合、そのオペレーションに対応する応答アクティビティを設定してください。
- 同期の受付アクティビティの場合、そのオペレーションに対応する応答アクティビティの間には、別の受付アクティビティを設定しないでください。
- インスタンスを生成するすべての受付アクティビティは、使用する関連セットを同じになるように設定してください。
- インスタンスを生成する受付アクティビティを一つ以上作成してください。
- 永続化しないビジネスプロセスの場合、非同期の受付アクティビティを設定しないでください。

### 6.6.3 応答アクティビティ

ビジネスプロセスがサービスリクエストから同期受信した要求電文に対して、応答またはフォルトを返すためのインターフェースを定義するアクティビティです。対応する受付アクティビティの通信モデルに同期受付を定義した場合、必ず定義します。

応答アクティビティは、[ 応答アクティビティ ] ダイアログで詳細を定義します。

#### (1) 定義手順

定義の手順を次に示します。

1. 応答アクティビティをキャンパスに配置します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
2. 次のどちらかの方法で [ 応答アクティビティ ] ダイアログを表示させます。
  - キャンパスの応答アクティビティをダブルクリックする
  - キャンパスの応答アクティビティを選択して右クリックし、[ 設定 ] を選択する
 [ 応答アクティビティ ] ダイアログが表示されます。
3. [ 応答アクティビティ ] ダイアログに必要な情報を入力します。  
[ 応答アクティビティ ] ダイアログの表示・入力内容の詳細については、「11.4.7 応答アクティビティダイアログ」を参照してください。
  - [ 割当変数 ] に設定する変数の内容を編集する場合  
[ 編集 ] ボタンをクリックします。表示される [ 変数・相関セット一覧 ] ダイアログで、変数の内容を編集できます。[ 変数・相関セット一覧 ] ダイアログの詳細については、「11.4.1 変数・相関セット一覧ダイアログ」を参照してください。  
なお、該当するオペレーションを持つユーザ定義受付を定義している場合、ユーザ定義受付の電文フォーマットを変数に設定できます。設定は、[ 電文フォーマットの取込 ] ダイアログで行います。[ 電文フォーマットの取込 ] ダイアログは、[ 変数・相関セット一覧 ] ダイアログの [ 取込 ] ボタンをクリックすると表示されます。  
[ 電文フォーマットの取込 ] ダイアログを表示する際、[ 応答アクティビティ ] ダイアログでオペレーション名を設定しておくこと、ユーザ定義受付内にある該当するオペレーションの電文フォーマットが選択された状態になります。  
[ 電文フォーマットの取込 ] ダイアログの詳細については、「11.4.4 電文フォーマットの取込ダイアログ」を参照してください。
  - [ 割当相関セット群 ] を設定する場合  
[ 設定 ] ボタンをクリックします。表示される [ 割当相関セット群 ] ダイアログで割当相関セットを設定します。[ 割当相関セット群 ] ダイアログの詳細については、「11.4.2 割当相関セット群ダイアログ」を参照してください。
4. [ OK ] ボタンをクリックします。

### (2) 定義時の注意事項

- アクティビティ名はビジネスプロセス（スコープ内も含む）内で一意になるように指定してください。
- アクティビティ名は 64 バイト以内にしてください。
- 必ずオペレーション名を設定してください。
- フォルト名が設定されると、フォルトを返します。
- 必ず割当変数を指定してください。
- 同一のオペレーション名を持つ応答アクティビティで、同じ割当変数名を設定してください。
- 同一オペレーションの応答アクティビティのフォルト名に対応する割当変数を一つに設定してください。
- 同期の受付アクティビティを設定した場合、その受付アクティビティのオペレーションに対応する応答アクティビティを設定してください。
- 受付アクティビティとそのそのオペレーションに対応する応答アクティビティの間には、別の応答アクティビティを設定しないでください。
- 永続化しないビジネスプロセスの場合、応答アクティビティのあとに実行されるアクティビティを設定しないでください。

## 6.6.4 サービス呼出アクティビティ

定義済みの HCSC コンポーネントへの要求電文の送信について定義するアクティビティです。

サービス呼出アクティビティは、[ サービス呼出アクティビティ ] ダイアログで詳細を定義します。

### (1) 定義手順

定義の手順を次に示します。

1. サービス呼出アクティビティをキャンバスに配置します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
2. 次のどちらかの方法で [ サービス呼出アクティビティ ] ダイアログを表示させます。
  - キャンバスのサービス呼出アクティビティをダブルクリックする
  - キャンバスのサービス呼出アクティビティを選択して右クリックし、[ 設定 ] を選択する[ サービス呼出アクティビティ ] ダイアログが表示されます。
3. [ サービス呼出アクティビティ ] ダイアログに必要な情報を入力します。  
[ サービス呼出アクティビティ ] ダイアログの表示・入力内容の詳細については、「11.4.8 サービス呼出アクティビティダイアログ」を参照してください。
  - [ 要求電文用割当変数 ] および [ 応答電文用割当変数 ] に設定する変数の内容を編集する場合

[編集] ボタンをクリックします。表示される [変数・相関セット一覧] ダイアログで、変数の内容を編集できます。[変数・相関セット一覧] ダイアログの詳細については、「11.4.1 変数・相関セット一覧ダイアログ」を参照してください。

なお、呼び出すサービス部品で定義されている電文フォーマットを変数に設定する場合、[電文フォーマットの取込] ダイアログで設定できます。[電文フォーマットの取込] ダイアログを利用して呼び出すサービス部品で定義されている電文フォーマットを変数に設定する手順については、この手順の次に記載されている「ポイント」を参照してください。

[電文フォーマットの取込] ダイアログの詳細については、「11.4.4 電文フォーマットの取込ダイアログ」を参照してください。

- [割当相関セット群] を設定する場合

[設定] ボタンをクリックします。表示される [割当相関セット群] ダイアログで割当相関セットを設定します。[割当相関セット群] ダイアログの詳細については、「11.4.2 割当相関セット群ダイアログ」を参照してください。

4. [OK] ボタンをクリックします。

#### ポイント

---

呼び出すサービス部品で定義されている電文フォーマットを変数に設定する方法を次に示します。なお、要求電文の場合で説明していますが、応答電文についても同様の手順です。

1. [サービス呼出アクティビティ] ダイアログで呼び出すサービス名およびオペレーション名を設定します。
  2. [要求電文用割当変数] の [編集] ボタンをクリックします。  
[変数・相関セット一覧] ダイアログが表示されます。
  3. 変数名を設定します。
  4. [取込] ボタンをクリックします。  
[電文フォーマットの取込] ダイアログが表示されます。  
[サービス名] および [オペレーション名] には、手順 2. で設定した値が設定されています。また、[電文種別] に「要求電文」が設定されています。
  5. [電文フォーマット] に電文フォーマット名を設定します。
  6. [OK] ボタンをクリックします。  
[電文フォーマットの取込] ダイアログが閉じられ、[変数・相関セット一覧] ダイアログに戻ります。
  7. [追加] ボタンをクリックします。  
定義した変数が [変数一覧] に追加されます。
  8. 変数一覧から追加した変数を選択します。
  9. [OK] ボタンをクリックします。  
[変数・相関セット一覧] ダイアログが閉じられ、[サービス呼出アクティビティ] ダイアログに戻ります。
- 

#### (2) 定義時の注意事項

- アクティビティ名はビジネスプロセス（スコープ内も含む）内で一意になるように指定してください。
- サービス名およびオペレーション名には、定義済みの HCSC コンポーネントとそのオ

## 6. ビジネスプロセスの定義

ペレーションを設定してください。また、設定したオペレーションの通信モデルと定義されているオペレーションの通信モデルを一致させてください。

- アクティビティ名は 64 バイト以内にしてください。
- 呼び出すサービス名を設定してください。
- 同期呼び出しの場合、必ず応答電文用の割当変数を設定してください。
- 永続化しないビジネスプロセスの場合、非同期のサービス呼出アクティビティを設定しないでください。

### 6.6.5 Java 呼出アクティビティ

専用インターフェース (jp.co.Hitachi.soft.csc.bp.receiver.ejb.CustomClassInterface) を実装した Java クラスの呼び出しを定義するアクティビティです。

Java 呼出アクティビティは、[Java 呼出アクティビティ] ダイアログで詳細を定義します。

#### (1) Java 呼出アクティビティの定義前の準備

Java 呼出アクティビティを定義する前には、次の準備をする必要があります。

##### HCSCTE プロジェクトの作成

[Java 呼出アクティビティ] ダイアログで指定する Java クラスは、HCSCTE プロジェクトの src ディレクトリに格納されます。そのため、必ず HCSCTE プロジェクトを作成しておく必要があります。

HCSCTE プロジェクトの作成方法については、「3.1 プロジェクトの管理」を参照してください。

##### Java クラスの作成

Java 呼出アクティビティで利用する Java クラスを作成します。利用する Java クラスのインターフェースについては「6.6.5(2) 利用する Java クラスのインターフェース」を参照してください。

なお、コンパイルするには、次に示す JAR ファイルを、HCSCTE プロジェクトプロパティーの [Java のビルド・パス] に追加しておく必要があります。

```
<Cosminexusのインストールディレクトリ>%CSC%lib%cscbp_ejb.jar
```

**!** 注意事項

Java クラスは、リポジトリ内のすべての HCSC コンポーネントで共有されます。そのため、既存の Java クラスを変更した場合、変更前の内容のクラスと変更後の内容のクラスの HCSC コンポーネントが混在してしまうおそれがあります。

Java クラスを変更した場合、変更した Java クラスを呼び出す Java 呼出アクティビティが定義されているビジネスプロセスを、再度パッケージングしてください。

また、Java 呼出アクティビティを削除しても、利用している Java クラスは削除されません。使用していない不要な Java クラスは、別途削除してください。

## ライブラリの追加

ビジネスプロセスをパッケージングして作成される EAR ファイルにライブラリを追加したい場合、HCSCTE プロジェクトの lib ディレクトリにライブラリをコピーしておいてください。

lib ディレクトリにコピーしたライブラリは、ビジネスプロセスをパッケージングすると、自動的に EAR ファイルに追加されます。

なお、lib ディレクトリには、ディレクトリや次の名前のライブラリを登録しないでください。

- csbdef.jar
- csbp\_ejb.jar
- csbjava.jar

**!** 注意事項

HCSCTE プロジェクト内の src ディレクトリおよび lib ディレクトリへのリンクは削除しないでください。

## (2) 利用する Java クラスのインターフェース

Java 呼出アクティビティで呼び出す Java クラスは、次に示すインターフェースを実装している必要があります。

```
package jp.co.Hitachi.soft.csc.bp.receiver.ejb;

public interface CustomClassInterface {
    public Object invoke(
        String processName,
        int version,
        String activityName,
        Object inputData
    ) throws CSBUserException, CSBSystemException;
}
```

Java クラスの内容について次に説明します。

## 6. ビジネスプロセスの定義

### 引数

仮引数名	説明
processName	ビジネスプロセス名
version	ビジネスプロセスのバージョン
activityName	アクティビティ名
inputData	Java 呼出アクティビティダイアログの [ 引数用割当変数 ] で指定した変数

### 戻り値

Java 呼出アクティビティダイアログの [ 戻り値用割当変数 ] に指定した変数に代入されます。

### 例外

throw される例外が CSBUserException か、CSBSystemException かによって処理が異なります。

#### CSBUserException が throw された場合

Java 呼出アクティビティにフォルト処理が設定されている場合、CSBUserException が throw されると、フォルトコネクションでフォルト処理として定義されているアクティビティが実行されます。  
Java 呼出アクティビティにフォルト処理が設定されていない場合、invokeJavaFault が発生したもものとして処理が進められます。  
CSBUserException のインターフェースを次に示します。

```
package jp.co.Hitachi.soft.csc.bp;

public class CSBUserException extends Exception {
    public CSBUserException() { super(); }
    public CSBUserException(String message) { super(message); }
}
```

#### CSBSystemException が throw された場合

CSBSystemException が throw されると、システム例外が発生したもものとして処理が中断されます。このとき、プロセスの実行状態を永続化している場合には、ロールバックが実行されます。  
CSBSystemException のインターフェースを次に示します。

```

package jp.co.Hitachi.soft.csc.bp;

public class CSBSystemException extends Exception {
    public CSBSystemException() { super(); }
    public CSBSystemException(String message) { super(message); }
    public CSBSystemException(String message, Throwable cause) {
        super(message, cause);
    }
    public CSBSystemException(Throwable cause) { super(cause); }
}

```

CSBUserException および CSBSystemException 以外の例外が発生した場合、処理が中断されます。このとき、プロセスの実行状態を永続化している場合には、ロールバックが実行されます。

#### 注意事項

- Java 呼出アクティビティダイアログの [ 引数用割当変数 ] および [ 戻り値用割当変数 ] に指定する変数の型と、このインターフェースの引数および戻り値の型の対応を次に示します。

[ 引数用割当変数 ] および [ 戻り値用割当変数 ] の変数型	引数および戻り値の型
boolean	java.lang.Boolean
numeric	java.lang.Double
string	java.lang.String
message	byte[]

上記の型以外を指定した場合、エラー（KDEC20030-E）になります。

- 利用する Java クラスのインターフェースは cscbp\_ejb.jar に含まれます。そのため、コンパイルするときには、cscbp\_ejb.jar をクラスパスに追加する必要があります。
- Java プログラムでは、呼び出しごとにデフォルトコンストラクタでインスタンスが生成されます。そのため、インスタンスにデータを保持することはできません。
- Java プログラムの呼び出しは、コンテナ拡張ライブラリの延長で行われます。開発および実行にあたっては、Cosminexus におけるコンテナ拡張ライブラリの利用の指針に従ってください。コンテナ拡張ライブラリの利用の指針については、マニュアル「Cosminexus 機能解説」を参照してください。

### (3) 定義手順

Java 呼出アクティビティの定義の手順を次に示します。

1. Java 呼出アクティビティをキャンバスに配置します。

アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を

## 6. ビジネスプロセスの定義

参照してください。

2. 次のどちらかの方法で [Java 呼出アクティビティ] ダイアログを表示させます。
  - キャンパスの Java 呼出アクティビティをダブルクリックする
  - キャンパスの Java 呼出アクティビティを選択して右クリックし、[設定] を選択する

[Java 呼出アクティビティ] ダイアログが表示されます。

3. [Java 呼出アクティビティ] ダイアログに必要な情報を入力します。

[Java 呼出アクティビティ] ダイアログの表示・入力内容の詳細については、「11.4.9 Java 呼出アクティビティダイアログ」を参照してください。

[引数用割当変数] および [戻り値用割当変数] に設定する変数の内容を編集する場合、[編集] ボタンをクリックします。表示される [変数・関連セット一覧] ダイアログで、変数の内容を編集できます。[変数・関連セット一覧] ダイアログの詳細については、「11.4.1 変数・関連セット一覧ダイアログ」を参照してください。

4. [Java 呼出アクティビティ] ダイアログの [OK] ボタンをクリックします。

[Java 呼出アクティビティ] ダイアログが閉じられます。
5. キャンパスの Java 呼出アクティビティを右クリックし、[Java エディタ起動] を選択します。

Eclipse の Java エディタが起動されます。

Java エディタには、[Java 呼出アクティビティ] ダイアログで指定したクラスのソースコードが表示されます。[Java 呼出アクティビティ] ダイアログで指定したクラスが初めて編集するクラスの場合、Java エディタを起動するとソースコードのテンプレートが表示されます。
6. Java 呼出アクティビティから呼び出される Java クラスのソースコードを、Java エディタで編集します。
7. 編集した Java クラスのソースコードをコンパイルします。
8. 編集したソースコードおよびコンパイルしたクラスファイルを保存して、Java エディタを終了します。

### (4) 定義時の注意事項

- アクティビティ名はビジネスプロセス（スコープ内も含む）内で一意になるように指定してください。
- アクティビティ名は 64 バイト以内になしてください。
- 必ず Java クラス名を設定してください。

## 6.6.6 データ変換アクティビティ

データ変換処理を定義するアクティビティです。

データ変換アクティビティは、[データ変換アクティビティ] ダイアログで詳細を定義し

ます。

また、データ変換定義画面でデータの変換方法の定義情報（データ変換定義ファイル）を作成します。

### （１）定義手順

定義の手順を次に示します。

1. データ変換アクティビティをキャンバスに配置します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
2. 次のどちらかの方法で [ データ変換アクティビティ ] ダイアログを表示させます。
  - キャンバスのデータ変換アクティビティをダブルクリックする
  - キャンバスのデータ変換アクティビティを選択して右クリックし、[ 設定 ] を選択する

[ データ変換アクティビティ ] ダイアログが表示されます。
3. [ データ変換アクティビティ ] ダイアログに必要な情報を入力します。  
[ データ変換アクティビティ ] ダイアログの表示・入力内容の詳細については、「11.4.10 データ変換アクティビティダイアログ」を参照してください。  
[ 変換元変数 ] および [ 変換先変数 ] に設定する変数の内容を編集する場合、[ 編集 ] ボタンをクリックします。表示される [ 変数・相関セット一覧 ] ダイアログで、変数の内容を編集できます。[ 変数・相関セット一覧 ] ダイアログの詳細については、「11.4.1 変数・相関セット一覧ダイアログ」を参照してください。
4. キャンバスのデータ変換アクティビティを右クリックし、[ マッピング定義起動 ] を選択します。  
[ データ変換アクティビティ ] ダイアログで指定した内容が保存され、ダイアログが閉じられます。また、データ変換定義画面が表示されます。
5. データ変換定義画面でデータ変換定義ファイルを作成します。  
データ変換定義ファイルの作成方法の詳細については、「7.3 データ変換（マッピング）定義」を参照してください。  
データ変換定義画面の詳細については、「11.2.4 データ変換（マッピング）定義画面」を参照してください。  
なお、作成されたデータ変換定義ファイルは、ビジネスプロセスを保存するまで一時ディレクトリに保存されます。ビジネスプロセスを保存したときにリポジトリに格納されます。

**!** 注意事項

[データ変換アクティビティ] ダイアログの [データ変換定義] に指定したデータ変換定義ファイルがリポジトリに格納されたあとは、[変換元変数] および [変換先変数] に設定されている変数の情報のうち、変数の部分指定に関する情報だけ変更できます。変数の部分指定に関する情報以外を変更したい場合、[ファイルを削除] ボタンをクリックしてください。なお、この場合に [ファイルを削除] をクリックしても、データ変換定義ファイルはリポジトリから削除されません。データ変換定義ファイルは、ビジネスプロセスを保存したときにリポジトリから削除されます。

## (2) 定義時の注意事項

- アクティビティ名はビジネスプロセス (スコープ内も含む) 内で一意になるように指定してください。
- アクティビティ名は 64 バイト以内にしてください。
- 必ず変換先変数を設定してください。
- 必ず変換元変数を設定してください。
- 必ずデータ変換定義を設定してください。

## 6.6.7 代入アクティビティ

次に示す処理を実行するための定義をするアクティビティです。

- 変数 (基本型・メッセージ型) をほかの変数に代入する
- 変数の一部をほかの変数に代入する
- 一般式を作成し、変数に代入する  
一般式の評価結果は、文字列型として取得 (java.lang.String ヘマッピング) されます。

代入アクティビティは、[代入アクティビティ] ダイアログで詳細を定義します。

なお、変数に値を代入する際に、変数の型が異なっても、自動的に型を変換して代入できる場合があります。変換の可否と変換規則を次の表に示します。

表 6-15 代入値の変換規則

代入する値のデータ型	変換後のデータ型			
	boolean	numeric	string	message
boolean	-	1	2	×
numeric	3	-	4	×
string	5	6	-	×
message	×	×	×	-

(凡例)

- : 変換の必要はありません。
- × : 変換して代入できません。

- : 変換して代入できます。
- : 場合によって変換して代入できます。

注 1

true の場合 1, false の場合 0 が代入されます。

注 2

true または false が代入されます。

注 3

0 の場合は false, それ以外の場合は true が代入されます。

注 4

java.lang.Double クラスの toString(double) メソッドで得られる値が代入されます。

注 5

java.lang.Boolean クラスの parseBoolean(String) メソッドで得られる値が代入されます。

注 6

java.lang.Double クラスの parseDouble(String) メソッドで得られる値が代入されます。

代入アクティビティでは, コピー元とコピー先に指定できない組み合わせがあります。コピー元とコピー先の組み合わせを次の表に示します。

表 6-16 コピー元とコピー先の組み合わせ

コピー元	コピー先			
	変数 (メッセージ型)	変数 (基本型)	変数の部分 (メッセージ型)	変数の部分 (基本型)
変数 (メッセージ型)		×		×
変数 (基本型)	×		×	
変数の部分 (メッセージ型)		×		×
変数の部分 (基本型)	×		×	
式	×		×	

(凡例)

: 指定できます。

×: 指定できません。

注 コピー先の変数が初期化されていない場合は, コピーできません。

## (1) 定義手順

定義の手順を次に示します。

1. 代入アクティビティをキャンバスに配置します。  
アクティビティを配置する方法については, 「6.4.1 アクティビティを配置する」を参照してください。
2. 次のどちらかの方法で [代入アクティビティ] ダイアログを表示させます。

## 6. ビジネスプロセスの定義

- キャンパスの代入アクティビティをダブルクリックする
- キャンパスの代入アクティビティを選択して右クリックし、[ 設定 ] を選択する

[ 代入アクティビティ ] ダイアログが表示されます。

3. [ 代入アクティビティ ] ダイアログに必要な情報を入力します。  
[ 代入アクティビティ ] ダイアログの表示・入力内容の詳細については、「11.4.11 代入アクティビティダイアログ」を参照してください。  
[ コピー元 ] および [ コピー先 ] のデータを追加または編集する場合は、[ 追加 ] ボタンまたは [ 編集 ] ボタンをクリックしてください。表示されるダイアログ ([ 代入アクティビティ ] サブダイアログ) で、[ コピー元 ] および [ コピー先 ] に設定するデータを追加、編集できます。また、設定する値が変数で、変数の内容を編集する場合は、[ 代入アクティビティ ] サブダイアログの [ 編集 ] ボタンをクリックします。表示される [ 変数・関連セット一覧 ] ダイアログで、変数の内容を編集できます。  
[ 代入アクティビティ ] サブダイアログの詳細については、「11.4.12 代入アクティビティサブダイアログ」を参照してください。また、[ 変数・関連セット一覧 ] ダイアログの詳細については、「11.4.1 変数・関連セット一覧ダイアログ」を参照してください。
4. [ OK ] ボタンをクリックします。

### (2) 定義時の注意事項

- アクティビティ名はビジネスプロセス ( スコープ内も含む ) 内で一意になるように指定してください。
- アクティビティ名は 64 バイト以内にしてください。
- 一つ以上の代入操作を設定してください。
- 必ずコピー元を設定してください。
- 必ずコピー先を設定してください。
- コピー元のタイプとして、変数と式のどちらかを設定してください。
- コピー元の変数が選択されている場合、必ず変数名を設定してください。
- コピー元の式が選択されている場合、必ず式を設定してください。
- コピー先とコピー元の変数の型が異なる場合、暗黙的に変換されたり、実行時にエラーが発生したりする場合があります。
- 変数 ( メッセージ型 ) から変数 ( メッセージ型 ) への代入を定義する場合、それぞれの変数の定義で設定する電文フォーマットは、同じ XML スキーマかつ同じファイル名の定義ファイルを指定してください。

## 6.6.8 無操作アクティビティ

実行しても何の動作も行わないことを定義するアクティビティです。無操作アクティビティは、分岐アクティビティを使用して分岐した処理や、フォルトコネクションを使用したフォルト処理などの中で、何も処理しないことを示すために使用します。

無操作アクティビティは、[ 無操作アクティビティ ] ダイアログで詳細を定義します。

### (1) 定義手順

定義の手順を次に示します。

1. 無操作アクティビティをキャンバスに配置します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
2. 次のどちらかの方法で [無操作アクティビティ] ダイアログを表示させます。
  - キャンバスの無操作アクティビティをダブルクリックする
  - キャンバスの無操作アクティビティを選択して右クリックし、[設定] を選択する
 [無操作アクティビティ] ダイアログが表示されます。
3. [無操作アクティビティ] ダイアログに必要な情報を入力します。  
[無操作アクティビティ] ダイアログの表示・入力内容の詳細については、「11.4.13 無操作アクティビティダイアログ」を参照してください。
4. [OK] ボタンをクリックします。

### (2) 定義時の注意事項

- アクティビティ名はビジネスプロセス（スコープ内も含む）内で一意になるように指定してください。
- アクティビティ名は 64 バイト以内にしてください。

## 6.6.9 フォルト送出アクティビティ

ビジネスプロセス内でフォルトを定義、送出して、上位のスコープアクティビティにフォルトを通知するためのアクティビティです。

フォルト送出アクティビティは、[フォルト送出アクティビティ] ダイアログで詳細を定義します。

### (1) 定義手順

定義の手順を次に示します。

1. フォルト送出アクティビティをキャンバスに配置します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
2. 次のどちらかの方法で [フォルト送出アクティビティ] ダイアログを表示させます。
  - キャンバスのフォルト送出アクティビティをダブルクリックする
  - キャンバスのフォルト送出アクティビティを選択して右クリックし、[設定] を選択する
 [フォルト送出アクティビティ] ダイアログが表示されます。
3. [フォルト送出アクティビティ] ダイアログに必要な情報を入力します。  
[フォルト送出アクティビティ] ダイアログの表示・入力内容の詳細については、

## 6. ビジネスプロセスの定義

「11.4.14 フォルト送出アクティビティダイアログ」を参照してください。

[ 割当変数 ] には、フォルトを送出する際に割り当てる変数を設定します。[ 割当変数 ] に設定する変数の内容を編集する場合、[ 編集 ] ボタンをクリックします。表示される [ 変数・相関セット一覧 ] ダイアログで、変数の内容を編集できます。[ 変数・相関セット一覧 ] ダイアログの詳細については、「11.4.1 変数・相関セット一覧ダイアログ」を参照してください。

4. [ OK ] ボタンをクリックします。

### (2) 定義時の注意事項

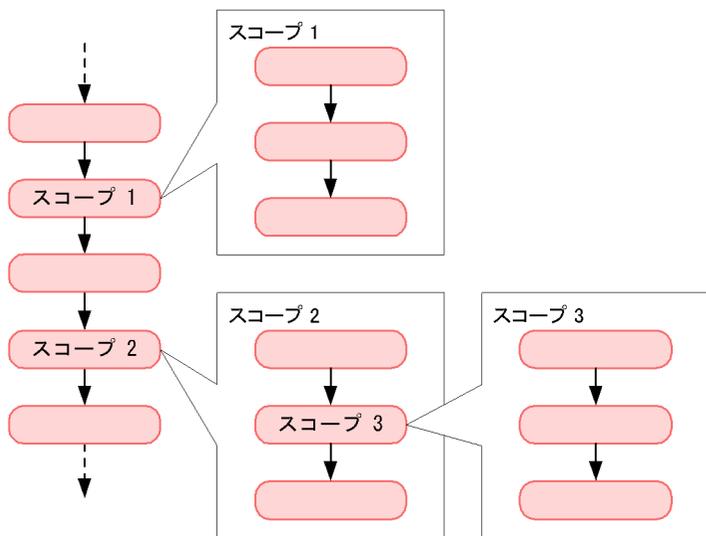
- アクティビティ名はビジネスプロセス（スコープ内も含む）内で一意になるように指定してください。
- アクティビティ名は 64 バイト以内にしてください。
- フォルト送出アクティビティを含む上位のスコープに、対応する割当変数または catchAll を利用したフォルト処理を定義してください。

## 6.6.10 スコープアクティビティ

一つまたは複数のアクティビティを、ひとまとまりの処理単位として定義するためのアクティビティです。

スコープアクティビティを利用した処理単位について、次の図に示します。

図 6-8 スコープアクティビティを利用した処理単位の設定



(凡例)

- : アクティビティ
- : コネクションによる連結

スコープアクティビティを定義すると、スコープ内の処理フローを一つのまとまりとして扱うことができます。また、上図のスコープ3のように、スコープ内の処理フローにさらにスコープを定義することもできます。

なお、スコープ内で定義した変数は、そのスコープ内でだけ有効になります。スコープと変数の有効範囲についての詳細は、「6.5.1(5)(a) スコープアクティビティを使用している場合の変数の使用」を参照してください。

### (1) 定義手順

定義の手順を次に示します。

1. スコープアクティビティをキャンパスに配置します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
2. 次のどちらかの方法で [ スコープアクティビティ ] ダイアログを表示させます。
  - キャンパスのスコープアクティビティをダブルクリックする
  - キャンパスのスコープアクティビティを選択して右クリックし、[ 設定 ] を選択する

[ スコープアクティビティ ] ダイアログが表示されます。
3. アクティビティ名を入力します。
4. [ OK ] ボタンをクリックします。
5. キャンパスのスコープアクティビティを選択して右クリックし、[ 開く ] を選択します。  
キャンパスの下部にスコープアクティビティ名のタブが表示されます。
6. スコープアクティビティ名のタブをクリックします。  
スコープ内の処理を設定するキャンパスが表示されます。
7. 任意のアクティビティを配置・連結・定義し、スコープ内の処理を設定します。

### (2) 定義時の注意事項

アクティビティ名は64バイト以内にしてください。

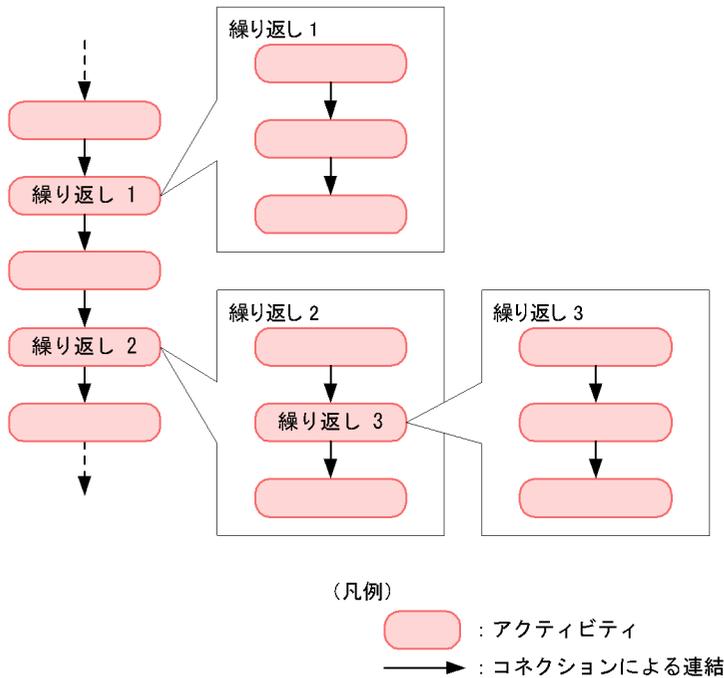
## 6.6.11 繰り返しアクティビティ

一つまたは複数のアクティビティを、条件に従って繰り返すための定義をするアクティビティです。

繰り返しアクティビティを利用した繰り返し処理について、次の図に示します。

## 6. ビジネスプロセスの定義

図 6-9 繰り返しアクティビティを利用した繰り返し処理



繰り返しアクティビティを定義すると、一つまたは複数の処理を一定の条件で繰り返すことができます。また、上図の繰り返し 3 のように、繰り返す処理の中に繰り返し処理を定義することもできます。

### (1) 定義手順

定義の手順を次に示します。

1. 繰り返しアクティビティをキャンバスに配置します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
2. 次のどちらかの方法で [ 繰り返しアクティビティ ] ダイアログを表示させます。
  - キャンバスの繰り返しアクティビティをダブルクリックする
  - キャンバスの繰り返しアクティビティを選択して右クリックし、[ 設定 ] を選択する[ 繰り返しアクティビティ ] ダイアログが表示されます。
3. [ 繰り返しアクティビティ ] ダイアログに必要な情報を入力します。  
[ 繰り返しアクティビティ ] ダイアログの詳細については、「11.4.16 繰り返しアクティビティダイアログ」を参照してください。
4. [ 繰り返し条件設定 ] ボタンをクリックします。
5. 繰り返しの条件を設定するための [ 条件設定 ] ダイアログが表示されます。

[条件設定] ダイアログの表示・入力内容の詳細については、「11.4.17 条件設定ダイアログ (繰り返しアクティビティ)」を参照してください。

6. [OK] ボタンをクリックします。
7. キャンパスの繰り返しアクティビティを選択して右クリックし、[開く] を選択します。  
キャンパスの下部に繰り返しアクティビティ名のタブが表示されます。
8. 繰り返しアクティビティ名のタブをクリックします。  
繰り返す処理を設定するキャンパスが表示されます。
9. 任意のアクティビティを配置・連結・定義し、繰り返す処理を設定します。

#### (2) 定義時の注意事項

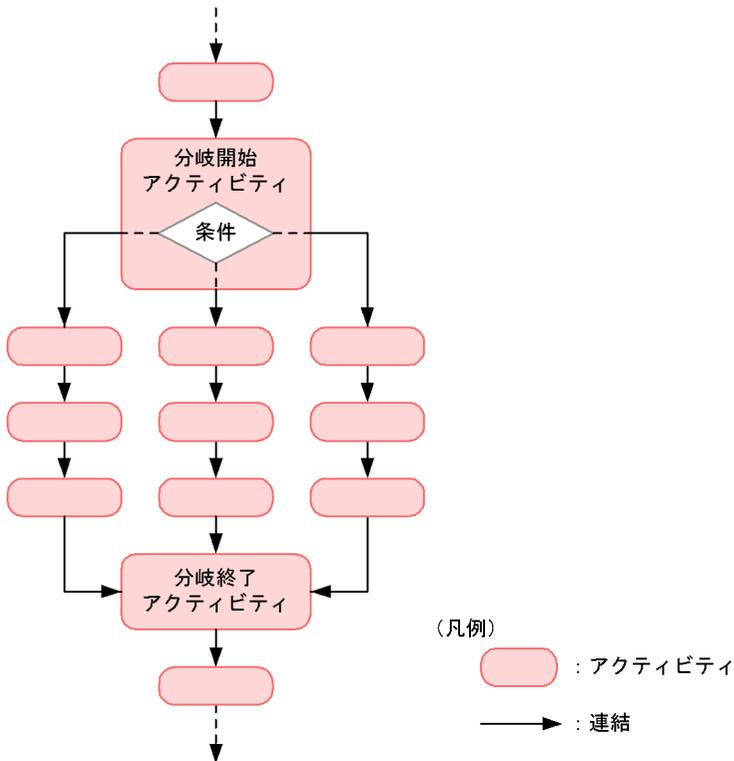
- アクティビティ名はビジネスプロセス (スコープ内も含む) 内で一意になるように指定してください。
- アクティビティ名は 64 バイト以内にしてください。
- 必ず繰り返し条件を設定してください。

### 6.6.12 分岐アクティビティ

条件の判定結果によるビジネスプロセスの処理の分岐を定義するアクティビティで、分岐開始アクティビティと分岐終了アクティビティの二つがあります。分岐の開始個所には分岐開始アクティビティを、終了個所には分岐終了アクティビティを配置します。

分岐アクティビティを利用した処理の分岐について、次の図に示します。

図 6-10 分岐アクティビティを利用した処理の分岐



分岐開始アクティビティの下には幾つかの処理を用意して、分岐開始アクティビティに設定した条件と条件の判定結果によって分岐させます。分岐の終端には、分岐終了アクティビティを配置します。

分岐開始アクティビティから分岐した処理のうち、フォルト送アウトアクティビティで終了する処理は、分岐終了アクティビティに連結する必要はありません。また、分岐開始アクティビティから分岐した処理がすべてフォルト送アウトアクティビティで終了する場合、分岐終了アクティビティを配置する必要はありません。

### (1) 分岐開始アクティビティの定義

分岐開始アクティビティの定義の手順を次に示します。

1. 分岐アクティビティをキャンバスに配置します。  
 アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
2. 分岐アクティビティの連結先となるアクティビティをキャンバスに配置します。  
 アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。

3. 分岐アクティビティと連結先のアクティビティを連結します。  
アクティビティを連結する方法については、「6.4.2(2) 連結の設定方法」を参照してください。
4. 次のどちらかの方法で [分岐アクティビティ] ダイアログを表示させます。
  - キャンパスの分岐アクティビティをダブルクリックする
  - キャンパスの分岐アクティビティを選択して右クリックし、[設定] を選択する
 [分岐アクティビティ] ダイアログが表示されます。
5. [分岐アクティビティ] ダイアログに必要な情報を入力します。  
[分岐アクティビティ] ダイアログの表示・入力内容の詳細については、「11.4.18 分岐アクティビティダイアログ」を参照してください。  
分岐条件を設定する場合、[条件設定] ボタンをクリックします。表示される [条件設定] ダイアログで、分岐条件を設定できます。連結先ごとに条件を設定します。分岐条件は XPath 式で入力します。[条件設定] ダイアログの詳細については、「11.4.19 条件設定ダイアログ (分岐アクティビティ)」を参照してください。
6. [OK] ボタンをクリックします。

## (2) 分岐終了アクティビティの定義

分岐終了アクティビティの定義の手順を次に示します。

1. 分岐アクティビティをキャンパスに配置します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
2. 分岐アクティビティと連結元のアクティビティを連結します。  
アクティビティを連結する方法については、「6.4.2(2) 連結の設定方法」を参照してください。

## (3) 定義時の注意事項

- アクティビティ名はビジネスプロセス (スコープ内も含む) 内で一意になるように指定してください。
- アクティビティ名は 64 バイト以内にしてください。
- 分岐条件を一つ以上設定してください。
- 遷移先を条件分岐またはデフォルトに割り当ててください。
- 必ず分岐の条件名を設定してください。
- 必ず分岐の条件を設定してください。
- 並列処理終了アクティビティに接続しないでください。
- 次に示すコネクションは接続しないでください。
  - 開始アクティビティを元とするコネクション <sup>1</sup>
  - あるアクティビティのあるフォルト処理の開始点を元とするコネクション <sup>1</sup>
- 分岐開始アクティビティは、対応する一つに分岐終了アクティビティに接続してください。 <sup>2</sup>

## 6. ビジネスプロセスの定義

- 対応する分岐処理を分岐終了アクティビティに接続してください。<sup>2</sup>
- 分岐処理は必ず分岐終了アクティビティに接続してください。<sup>2</sup>
- 分岐アクティビティと、対応する分岐終了アクティビティの間には、一つ以上のアクティビティを設定してください。

### 注 1

アクティビティを元とするコネクション、およびあるアクティビティのあるフォルト処理の開始点を元とするコネクションについては、「図 6-13 コネクションの例」を参照してください。

### 注 2

すべての分岐処理の終端がフォルト送出アクティビティの場合を除きます。

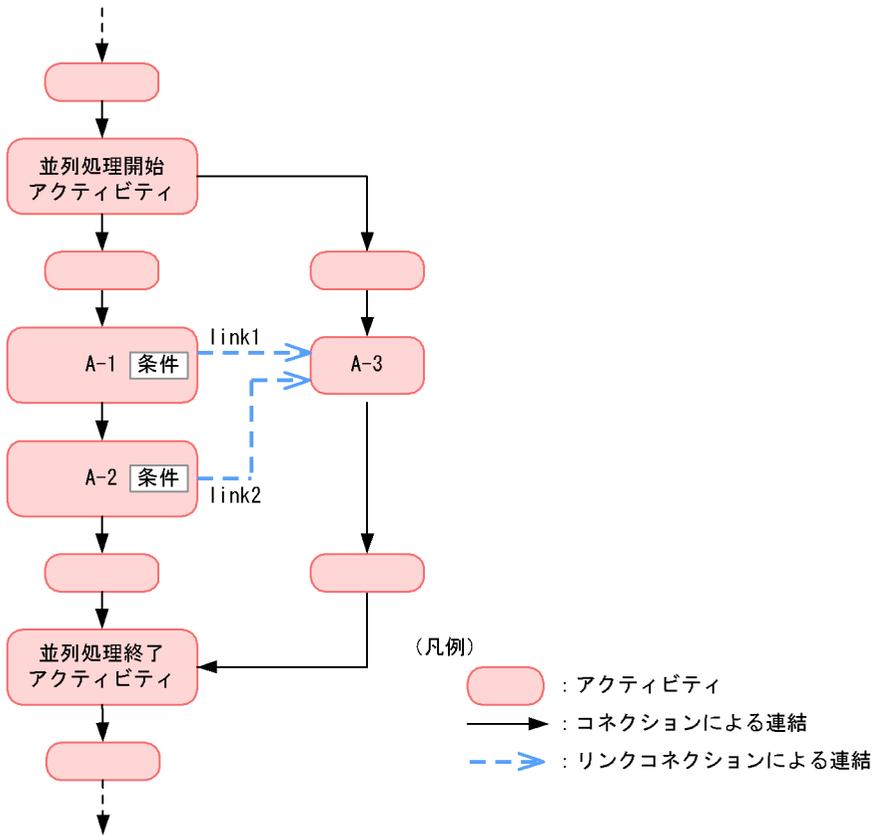
## 6.6.13 並列処理アクティビティ

ビジネスプロセスの処理フローを二つ以上のシーケンスに枝分かれさせて並列実行するための定義をするアクティビティで、並列処理開始アクティビティと並列処理終了アクティビティの二つがあります。並列処理の開始個所には並列処理開始アクティビティを、複数のシーケンスの合流個所には並列処理終了アクティビティを配置します。

並列処理開始アクティビティから枝分かれさせて並列実行するシーケンスのうち、フォルト送出アクティビティで終了するシーケンスは、並列処理終了アクティビティに連結する必要はありません。また、並列処理開始アクティビティから枝分かれさせて並列実行するシーケンスがすべてフォルト送出アクティビティで終了する場合、並列処理終了アクティビティを配置する必要はありません。

並列処理アクティビティを利用した処理の並列実行について、次の図に示します。

図 6-11 並列処理アクティビティを利用した処理の並列実行



並列処理開始アクティビティの下には複数のシーケンスを用意して、並列して処理を実行できます。なお、複数のシーケンス間での処理の順序を設定することもできます。

例えば、上の図の A-1 および A-2 が完了してから A-3 を実行する場合、A-1 および A-2 を連結元、A-3 を連結先として、リンクコネクション（link1 および link2）で連結します。さらに、A-1 および A-2 の処理の終了に加えて、リンクの条件を設定して A-1 および A-2 と A-3 のリンクを制御することもできます。

上の図の例の場合、連結先の A-3 の処理は、次のような過程で開始されます。

1. A-1 および A-2 の処理が完了します。
2. A-1 および A-2 に設定されているリンクの条件が判定されます。  
A-1 および A-2 のリンクの条件が判定されると、それぞれリンクコネクション link1 および link2 が活性化します。リンクコネクションは、条件の判定結果によって真または偽に活性化します。なお、条件が設定されていない場合、リンクコネクションは、リンク元のアクティビティの処理が完了すると真に活性化します。
3. A-3 に連結されているすべてのリンクコネクションが活性化すると、A-3 の処理が開

## 6. ビジネスプロセスの定義

始されます。

ただし、すべてのリンクコネクションが偽に活性化している場合、A-3の処理は開始されずに省略されます。A-3の処理が開始されるのは、リンクコネクションのうち、少なくとも一つのリンクコネクションが真に活性化している場合です。

並列処理終了アクティビティ以降のアクティビティの処理は、並列処理された複数のシーケンスがすべて完了したあとに実行されます。

### (1) 並列処理アクティビティの定義

並列処理アクティビティの定義の手順を次に示します。

1. 並列処理開始アクティビティをキャンバスに配置します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
2. 並列処理開始アクティビティから枝分かれするシーケンスの各アクティビティを配置・定義します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を、アクティビティを定義する方法は、「6.6 アクティビティの定義」の各アクティビティの説明を参照してください。
3. 並列処理終了アクティビティをキャンバスに配置します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
4. 並列処理開始アクティビティから並列処理終了アクティビティまでの各アクティビティを連結します。  
アクティビティを連結する方法については、「6.4.2(2) 連結の設定方法」を参照してください。
5. 次のどちらかの方法で [ 並列処理アクティビティ ] ダイアログを表示させます。
  - キャンバスの並列処理開始アクティビティをダブルクリックする
  - キャンバスの並列処理開始アクティビティを選択して右クリックし、[ 設定 ] を選択する[ 並列処理アクティビティ ] ダイアログが表示されます。
6. [ 並列処理アクティビティ ] ダイアログに必要な情報を入力します。  
[ 並列処理アクティビティ ] ダイアログの表示・入力内容の詳細については、「11.4.20 並列処理アクティビティダイアログ」を参照してください。複数シーケンス間のリンクの設定方法の詳細については、「6.6.13(2) シーケンス間リンクの定義」を参照してください。
7. [ OK ] ボタンをクリックします。

### (2) シーケンス間リンクの定義

並列に処理する複数のシーケンス間でのリンクの定義手順について次に示します。

1. 次のどちらかの方法で [ 並列処理アクティビティ ] ダイアログを表示させます。
  - キャンパスの並列処理開始アクティビティをダブルクリックする
  - キャンパスの並列処理開始アクティビティを選択して右クリックし、[ 設定 ] を選択する

[ 並列処理アクティビティ ] ダイアログが表示されます。
2. 並列処理内で設定するリンクの数だけ、リンク名を入力します。
3. [ OK ] ボタンをクリックし、[ 並列処理アクティビティ ] ダイアログを閉じます。
4. リンクさせるアクティビティを連結します。  
アクティビティを配置する方法については、「6.4.1 アクティビティを配置する」を参照してください。
5. 次のどちらかの方法で [ リンク ] ダイアログを表示させます。
  - 手順 4. で設定した連結線をダブルクリックする
  - 手順 4. で設定した連結線を選択して右クリックし、[ 設定 ] を選択する

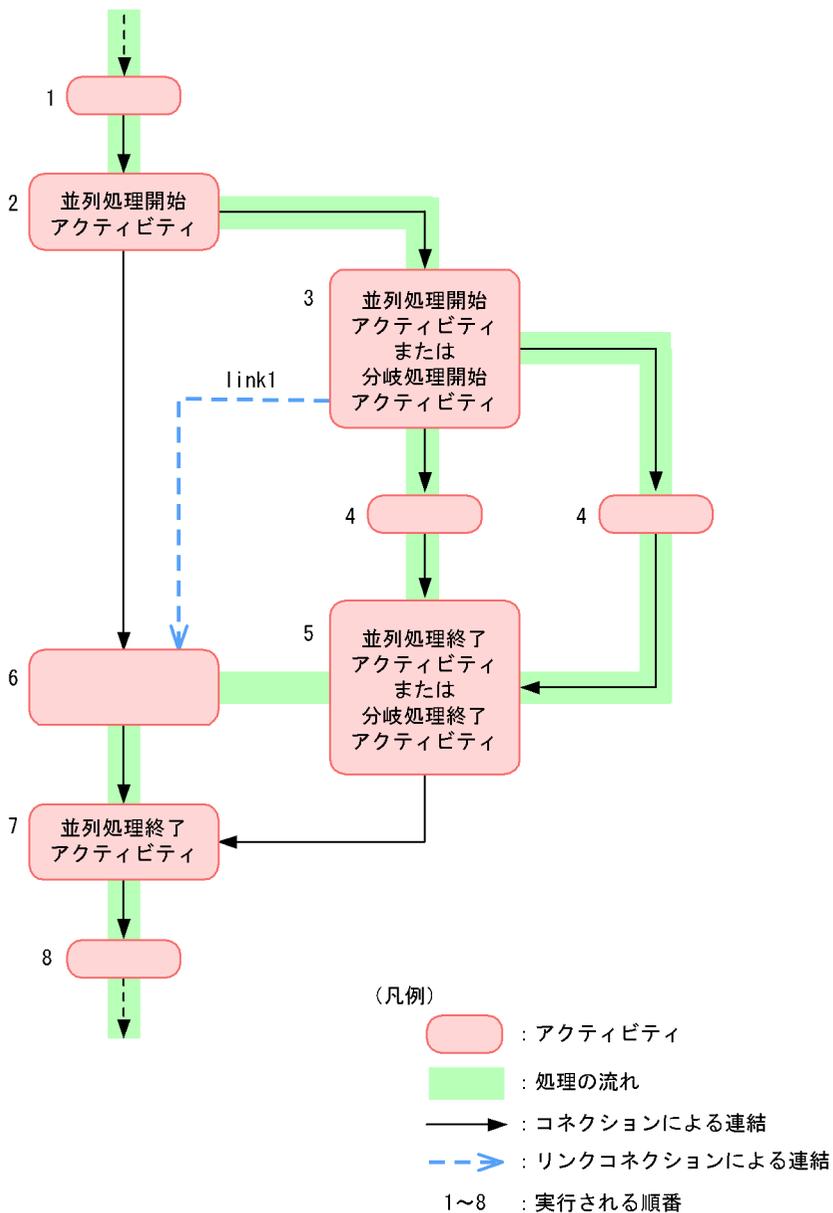
[ リンク ] ダイアログが表示されます。
6. [ リンク ] ダイアログに必要な情報を入力します。  
[ リンク ] ダイアログの表示・入力内容の詳細については、「11.4.21 リンクダイアログ」を参照してください。  
リンクの条件を設定する場合、[ 遷移条件設定 ] ボタンをクリックします。表示される [ 条件設定 ] ダイアログで、リンクの条件を設定できます。リンクの条件は XPath 式で入力します。[ 条件設定 ] ダイアログの詳細については、「11.4.22 条件設定ダイアログ ( 並列処理アクティビティ )」を参照してください。
7. [ OK ] ボタンをクリックします。

**!** 注意事項

並列処理開始アクティビティまたは分岐処理開始アクティビティからリンクが定義された場合、処理の流れは次の図のようになります。

## 6. ビジネスプロセスの定義

図 6-12 並列処理開始アクティビティまたは分岐処理開始アクティビティからリンクが定義された場合の処理の流れ



### (3) 定義時の注意事項

- アクティビティ名はビジネスプロセス（スコープ内も含む）内で一意になるように指定してください。
- すでに [ リンク ] ダイアログで定義されているリンクについては、[ 並列処理アクティビティ ] ダイアログでリンク名を変更・削除できません。

- アクティビティ名は 64 バイト以内にしてください。
- 設定するリンク名の長さは 64 バイト以内にしてください。
- 分岐終了アクティビティに接続しないでください。
- 次に示すコネクションは接続しないでください。
  - 開始アクティビティを元とするコネクション <sup>1</sup>
  - あるアクティビティのあるフォルト処理の開始点を元とするコネクション <sup>1</sup>
- 並列処理開始アクティビティは、対応する一つの並列処理終了アクティビティに接続してください。 <sup>2</sup>
- 対応する並列処理を並列処理終了アクティビティに接続してください。 <sup>2</sup>
- 並列処理は必ず並列処理終了アクティビティに接続してください。 <sup>2</sup>
- 並列処理開始アクティビティと、対応する並列処理終了アクティビティの間には、一つ以上のアクティビティを設定してください。
- リンクコネクションに有効なリンク名を設定してください。

注 1

アクティビティを元とするコネクション、およびあるアクティビティのあるフォルト処理の開始点を元とするコネクションについては、「[図 6-13 コネクションの例](#)」を参照してください。

注 2

並列処理するシーケンスの終端がフォルト送出アクティビティの場合を除きます。

## 6.6.14 終了アクティビティ

ビジネスプロセスの終了を表します。キャンバス上に一つだけ配置されます。このアクティビティは削除できません。

なお、終了アクティビティには、定義する内容はありません。

## 6.6.15 順次処理アクティビティ

順次処理される一連のアクティビティの情報を保持するためのアクティビティです。順序処理アクティビティは、必要に応じて自動で生成されます。順序処理アクティビティは、次に示すアクティビティの情報を保持します。

- 開始アクティビティと終了アクティビティの間に定義された一連のアクティビティ（スコープおよび繰り返しのキャンバスに定義されたアクティビティも同様）
- 分岐開始アクティビティと分岐終了アクティビティの間に定義された各分岐での一連のアクティビティ
- 並列処理開始アクティビティと並列処理終了アクティビティの間に定義された各分岐での一連のアクティビティ
- フォルトコネクションからつながるフォルト処理を行う一連のアクティビティ

## 6.7 ビジネスプロセスの保存

---

ビジネスプロセス定義画面で編集中のビジネスプロセスの内容を、編集途中や編集終了時に新規保存したり、上書き保存したりできます。保存した内容はリポジトリに保存されます。

変数定義で指定した電文フォーマット定義ファイル、およびデータ変換定義で生成したデータ変換定義ファイルは、ビジネスプロセス定義画面と同時に保存されます。

ビジネスプロセスを保存する手順を次に示します。

### 方法 1

Eclipse のメニューから、[ ファイル ] - [ 保管 ] を選択します。

### 方法 2

Eclipse のメニューから、[ ファイル ] - [ すべて保管 ] を選択します。

### 方法 3

ビジネスプロセス定義画面で、[ Ctrl ] キー + [ S ] キーを押します。

なお、ビジネスプロセス定義画面を閉じてビジネスプロセスの定義を終了する場合に、ビジネスプロセスが保存されていないときは、リソース保管ダイアログが表示されます。リソース保管ダイアログでは、保存されていないビジネスプロセスの定義を保存できません。

### 注意事項

不正なデータが入力されている場合、ビジネスプロセスを保存できないことがあります。その場合、表示されているメッセージに従って対処してください。

---

## 6.8 ビジネスプロセスの編集

保存したビジネスプロセスの内容を変更できます。ビジネスプロセスの内容を変更するには、ビジネスプロセス定義画面を表示して、内容を編集します。ビジネスプロセス定義画面は、ツリービューのサービス定義一覧から、該当するビジネスプロセスを選択し、ダブルクリックすると表示されます。

ビジネスプロセスの内容の定義方法については、「6.3 ビジネスプロセスの内容の定義」を参照してください。

### 6.8.1 ビジネスプロセスおよびアクティビティの定義情報を変更する

ビジネスプロセス、およびアクティビティの定義情報を変更できます。

#### (1) ビジネスプロセスの定義情報を変更する

ビジネスプロセスの定義情報を変更するには、次の二つの方法があります。

##### 方法 1

1. ツリービューのサービス定義一覧で表示されたサービスから、該当するビジネスプロセスを選択し、ダブルクリックします。  
選択したビジネスプロセスのビジネスプロセス定義画面が表示されます。
2. 内容を変更します。

##### 方法 2

プロパティから、次に示すビジネスプロセスの定義情報を変更できます。次の情報以外を変更する場合は、方法 1 を実施します。

- ビジネスプロセス名
  - サービス ID
  - ステータスの永続化
1. ツリービューのサービス定義一覧で表示されたサービスから、該当するビジネスプロセスを選択します。  
選択したビジネスプロセスに関するプロパティがプロパティビューに表示されます。
  2. 変更する項目を選択して変更します。

#### (2) アクティビティの定義情報を変更する

アクティビティの定義情報を変更するには、次の二つの方法があります。

##### 方法 1

1. ツリービューのサービス定義一覧で表示されたサービスから、該当するビジネスプロセスを選択し、ダブルクリックします。  
ビジネスプロセス定義画面が表示されます。

## 6. ビジネスプロセスの定義

2. キャンバス上の該当するアクティビティを選択します。
3. ダブルクリック、または右クリックで「設定」を選択します。  
アクティビティのダイアログが表示されます。
4. 内容を変更します。

### 方法 2

開始アクティビティおよび終了アクティビティ以外のアクティビティの場合、プロパティからサービス名を変更できます。サービス名以外の情報を変更する場合は、方法 1 を実施します。

アクティビティの定義情報をプロパティから変更する手順を次に示します。

1. ビジネスプロセス定義画面のキャンバス上のアクティビティを一つ選択します。  
選択したアクティビティに関するプロパティがプロパティビューに表示されず。
2. 変更する項目をダブルクリックして変更します。

## 6.8.2 アクティビティ名を変更する

開始アクティビティおよび終了アクティビティを除く、すべてのアクティビティ名を変更できます。

アクティビティ名は、ビジネスプロセス内で一意である必要があります。

アクティビティ名を変更するには、次の二つの方法があります。

### 方法 1

1. キャンバス上でアクティビティを選択します。
2. アクティビティを選択したまま、左クリックまたは [ F2 ] キーを押します。
3. アクティビティ名を編集します。
4. [ Enter ] キーを押すか、カーソルを外します。  
アクティビティ名が変更されます。

変更後のアクティビティ名が、ビジネスプロセス内ですでに使用されている場合や空白にした場合は、エラーになり、変更前のアクティビティに戻ります。

### 方法 2

1. キャンバス上でアクティビティを選択します。  
プロパティビューに選択したアクティビティのプロパティが表示されます。
2. プロパティビューに表示されるアクティビティ名を選択して、アクティビティ名を編集します。
3. [ Enter ] キーを押すか、カーソルを外します。

## 6.8.3 運用中のビジネスプロセス定義を変更する

業務の変更や改善に応じて運用中のビジネスプロセスの定義を変更する方法には、次の二つの方法があります。

#### 変更前後のビジネスプロセスを並行運用する方法

既存のビジネスプロセスとバージョンアップしたビジネスプロセスを並行運用します。

#### 変更前後のビジネスプロセスを並行運用しない方法

既存のビジネスプロセス定義に属する HCSC コンポーネントを実行環境からすべて削除して、定義を変更したビジネスプロセスに入れ替えます。

それぞれの方法を次に示します。

### (1) 並行運用する方法

1. 運用環境からリポジトリ情報をエクスポートし、開発環境にインポートします。
2. ビジネスプロセスの配備定義を削除します。  
ビジネスプロセスの配備定義の削除については、「9.3 HCSC コンポーネントの配備定義」を参照してください。
3. ツリービューのサービス定義一覧のビジネスプロセスを右クリックし、[バージョンアップ]を選択します。
4. 手順 3. で選択したビジネスプロセスをビジネスプロセス定義画面で編集します。
5. ビジネスプロセスをパッケージングします。  
パッケージングについては、「9.2 パッケージング」を参照してください。
6. ビジネスプロセスの配備定義をします。  
配備定義については、「9.3 HCSC コンポーネントの配備定義」を参照してください。
7. 運用環境で既存のビジネスプロセスを停止および削除します。
8. 手順 2. ~ 4. で変更したビジネスプロセスを含むリポジトリ情報をエクスポートし、運用環境にインポートします。
9. ビジネスプロセスを運用環境から実行環境に配備します。  
運用環境での操作については、マニュアル「Cosminexus サービスプラットフォームシステム構築・運用ガイド」を参照してください。

### (2) 並行運用しない方法

1. 運用環境からリポジトリ情報をエクスポートし、開発環境にインポートします。
2. 既存のビジネスプロセスの配備定義を削除します。  
ビジネスプロセスの配備定義の削除については、「9.3 HCSC コンポーネントの配備定義」を参照してください。
3. 既存のビジネスプロセス定義をビジネスプロセス定義画面で変更します。
4. ビジネスプロセスをパッケージングします。  
パッケージングについては、「9.2 パッケージング」を参照してください。

## 6. ビジネスプロセスの定義

5. ビジネスプロセスの配備定義をします。  
配備定義については、「9.3 HCSC コンポーネントの配備定義」を参照してください。
6. 運用環境で、既存のビジネスプロセス定義に属するプロセスがすべて完了したのを確認してから、実行環境に配備された HCSC コンポーネントを削除します。
7. 手順 2. および手順 3. で変更したビジネスプロセスを含むリポジトリ情報をエクスポートし、運用環境にインポートします。
8. ビジネスプロセスを運用環境から実行環境に配備します。  
運用環境での操作については、マニュアル「Cosminexus サービスプラットフォームシステム構築・運用ガイド」を参照してください。

### (3) 並行運用する場合のバージョンアップ時の注意事項

並行運用する場合、バージョンアップを実行すると、ビジネスプロセス定義画面で新バージョンのビジネスプロセスを編集できます。ただし、旧バージョンとの整合性を保つために、次に示す注意事項があります。

- プロパティから、サービス名またはステータスの永続化は変更できません。
- 既存の受付アクティビティの定義情報の変更や、既存の受付アクティビティの削除はできません。アクティビティ名の直接編集もできません。
- 既存の応答アクティビティの定義情報の変更や、削除はできません。
- 既存の受付アクティビティや応答アクティビティで使用している変数の定義情報の変更や、変数の削除はできません。ただし、使用している部分が変更されていなければ、部分名、指定式および型を変更できます。
- 既存の受付アクティビティや応答アクティビティで使用している関連セットの定義情報の変更や、関連セットの削除はできません。
- 既存のビジネスプロセスでユーザ定義受付を作成したあとにバージョンアップした場合、既存のユーザ定義受付は削除できません。

## 6.9 ビジネスプロセスの検証

---

作成したビジネスプロセスの内容が妥当かどうか検証できます。

必要なビジネスプロセス定義がない場合、または定義の関係が正しくない場合は正常に動作しません。そのため、実行環境で実行する前に、すべてのビジネスプロセス定義を検証する必要があります。

作成したビジネスプロセスに必要な項目がそろっているか、それらの関係が正しいかどうかを検証します。必要に応じて任意のタイミングで検証できます。

ビジネスプロセスの検証として、次の検証を行います。

- ビジネスプロセス定義の検証  
ビジネスプロセスの定義内容を検証します。必須項目、ビジネスプロセスが構造化されているか、およびそのほかの検証を行います。
- 呼出サービス部品の検証  
サービス呼出アクティビティで呼び出しているサービス部品との整合性を検証します。
- Java の検証  
Java 呼出アクティビティで使用する Java クラス、および Java クラスから使用するライブラリについて検証します。
- データ変換定義検証  
データ変換アクティビティのデータ変換定義の内容を検証します。
- ユーザ定義受付検証  
検証対象のビジネスプロセスに含まれるユーザ定義受付の定義内容を検証します。

ビジネスプロセス定義、呼出サービス部品、および Java の検証内容については、「6.9.1 検証内容」を参照してください。

ユーザ定義受付の検証の内容については、「8.6 ユーザ定義受付の検証」を参照してください。

### 6.9.1 検証内容

ビジネスプロセスの検証では、次の内容を検証します。

- 必須項目の検証
- 定義されたビジネスプロセスが構造化されているかの検証
- そのほかの検証

検証中にエラーが 100 件を超えた場合、検証途中でも検証が中止されます。この場合は、作成したビジネスプロセスを見直してください。

ビジネスプロセスの検証ではエラーにならなくても、ビジネスプロセスを実行したときに初めてエラーとなる場合があります。このような場合、ログを採取し、ログに出力されたメッセージに従って対処します。

## 6. ビジネスプロセスの定義

なお、ビジネスプロセスの検証は、基本的に定義情報だけの検証であり、ビジネスプロセスの実行時にエラーが発生しないことを保証するものではありません。よって、運用する前に十分なテストを行ってください。

### (1) 必須項目の検証

ビジネスプロセス定義画面に記述したビジネスプロセスの各アクティビティに、必須の項目が定義されているかどうかを検証されます。また、任意の項目は、設定されている場合は検証されます。必須項目の検証内容と対処を次の表に示します。

表 6-17 必須項目の検証内容と対処

項番	検証内容	エラーになった場合の対処
1	ビジネスプロセス名の長さが、64 バイト以内か。	次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>• ビジネスプロセス名を 64 バイト以下に変更。</li> <li>• 使用する文字コードを変更。</li> </ul>
2	変数・相関セット一覧ダイアログで定義した変数名の長さが、64 バイト以内か。	次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>• 変数名を 64 バイト以下に変更。</li> <li>• 使用する文字コードを変更。</li> </ul>
3	変数・相関セット一覧ダイアログで定義した変数の型がメッセージ型の場合、変数に対応する電文フォーマットが設定されているか。	変数・相関セット一覧ダイアログで、メッセージ型の変数に対応する電文フォーマットを設定します。
4	変数・相関セット一覧ダイアログで定義した変数の部分名が設定されているか。	変数・相関セット一覧ダイアログで、変数の部分名を設定します。
5	変数・相関セット一覧ダイアログで定義した変数の部分名に、指定式が設定されているか。	変数・相関セット一覧ダイアログで、変数の部分名に指定式を設定します。
6	変数・相関セット一覧ダイアログで定義した変数の部分名に、型が設定されているか。	変数・相関セット一覧ダイアログで、変数の部分名に型を設定します。
7	変数・相関セット一覧ダイアログで定義した相関セット名の長さの長さが 64 バイト以内か。	次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>• 相関セット名を 64 バイト以下に変更。</li> <li>• 使用する文字コードを変更。</li> </ul>
8	変数・相関セット一覧ダイアログで定義した相関セットに、部分名が一つ以上設定されているか。	変数・相関セット一覧ダイアログで、相関セットに、部分名を一つ以上設定します。
9	変数・相関セット一覧ダイアログで定義した相関セットに、利用変数名が設定されているか。	変数・相関セット一覧ダイアログで、相関セットに利用変数を設定します。
10	変数・相関セット一覧ダイアログで定義した相関セットに、有効な変数名が設定されているか。	変数・相関セット一覧ダイアログで、有効な変数名を設定します。
11	変数・相関セット一覧ダイアログで定義した相関セットで指定されている変数に、基本型の部分があるか。	変数・相関セット一覧ダイアログで、基本型の部分を持つ変数を設定します。
12	変数・相関セット一覧ダイアログで定義した相関セットに、部分名が設定されているか。	変数・相関セット一覧ダイアログで、相関セットに部分名を設定します。

項番	検証内容	エラーになった場合の対処
13	変数・相関セット一覧ダイアログで定義した相関セットに、有効な部分名が設定されているか。	変数・相関セット一覧ダイアログで、有効な部分名を設定します。
14	アクティビティのフォルト処理で、有効な変数が設定されているか。	フォルト処理の割当ダイアログで、有効な変数を設定します。
15	アクティビティのフォルト処理で、遷移先が設定されているか。	フォルト処理の割当ダイアログで、遷移先を設定します。
16	アクティビティのフォルト処理で、割当変数が設定されているか。	フォルト処理の割当ダイアログで、割当変数を設定します。
17	アクティビティのフォルト処理で、遷移先が設定されているか。	フォルト処理の割当ダイアログで、遷移先を設定します。
18	受付アクティビティで、オペレーション名が設定されているか。	受付アクティビティで、オペレーション名を設定します。
19	アクティビティ名の長さが 64 バイト以内か。	次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>• アクティビティ名を 64 バイト以下に変更。</li> <li>• 使用する文字コードを変更。</li> </ul>
20	受付アクティビティで、有効な割当変数が設定されているか。	受付アクティビティで、有効な割当変数を設定します。
21	受付アクティビティで、有効な割当相関セットが設定されているか。	受付アクティビティで、有効な相関セットを設定します。
22	リンクコネクションで、リンク名が指定されているか。	リンクコネクションに、リンク名を設定します。
23	リンクコネクションで、リンク先の遷移条件が「あり」に設定されている場合に、XPath が設定されているか。	リンクコネクションについて、次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>• 遷移条件を設定。</li> <li>• 遷移条件に「なし」を選択。</li> </ul>
24	応答アクティビティで、オペレーション名が設定されているか。	応答アクティビティで、オペレーション名を設定します。
25	応答アクティビティで、有効な割当変数が設定されているか。	応答アクティビティで、有効な割当変数を設定します。
26	応答アクティビティで、有効な割当相関セットが設定されているか。	応答アクティビティで、有効な相関セットを設定します。
27	応答アクティビティで、割当変数が指定されているか。	応答アクティビティで、割当変数を設定します。
28	サービス呼出アクティビティで、呼び出すサービス名が設定されているか。	サービス呼出アクティビティで、呼び出すサービス名を設定します。
29	サービス呼出アクティビティで、有効な要求電文用の割当変数が設定されているか。	サービス呼出アクティビティで、有効な要求電文用割当変数を設定します。
30	サービス呼出アクティビティで、有効な応答電文用の割当変数が設定されているか。	サービス呼出アクティビティで、有効な応答電文用割当変数を設定します。
31	サービス呼出アクティビティで、同期呼び出しの場合に、応答電文用の割当変数が設定されているか。	サービス呼出アクティビティで、同期呼び出しの場合、応答電文用割当変数を設定します。

## 6. ビジネスプロセスの定義

項番	検証内容	エラーになった場合の対処
32	サービス呼出アクティビティで、有効な相関セットが設定されているか。	サービス呼出アクティビティで、有効な相関セットを設定します。
33	代入アクティビティで、一つ以上の代入操作が設定されているか。	代入アクティビティで、一つ以上の代入操作を設定します。
34	代入アクティビティで、コピー元が設定されているか。	代入アクティビティで、コピー元を設定します。
35	代入アクティビティで、コピー先が設定されているか。	代入アクティビティで、コピー先を設定します。
36	代入アクティビティで、有効なコピー元変数が設定されているか。	代入アクティビティで、有効なコピー元変数を設定します。
37	代入アクティビティで、コピー元に有効な部分名が設定されているか。	代入アクティビティで、コピー元に有効な部分名を設定します。
38	データ変換アクティビティで、変換元変数が設定されているか。	データ変換アクティビティで、変換元変数を設定します。
39	データ変換アクティビティで、有効な変換元変数が設定されているか。	データ変換アクティビティで、有効な変換元変数を設定します。
40	データ変換アクティビティで、データ変換定義が設定されているか。	データ変換アクティビティで、データ変換定義を設定します。
41	データ変換アクティビティのデータ変換定義が正しいか。	検証で出力されたメッセージを参考にして、正しい定義に修正します。
42	代入アクティビティで、コピー元のタイプとして、変数と式のどちらかが選択されているか。	代入アクティビティで、コピー元のタイプとして、変数と式のどちらかを設定します。
43	代入アクティビティで、コピー元の変数が選択されている場合、変数名が設定されているか。	代入アクティビティで、コピー元の変数を設定します。
44	代入アクティビティで、コピー元の式が選択されている場合、式が設定されているか。	代入アクティビティで、コピー元の式を設定します。
45	代入アクティビティで、コピー先の変数名が設定されている場合、有効な変数が設定されているか。	代入アクティビティで、有効なコピー先変数を設定します。
46	データ変換アクティビティで、有効な変換先変数が設定されているか。	データ変換アクティビティで、有効な変換先変数を設定します。
47	代入アクティビティで、コピー先に有効な部分名が設定されているか。	代入アクティビティで、コピー先に有効な部分名を設定します。
48	代入アクティビティのコピー先、またはデータ変換アクティビティの変換先のどちらかで、変数が設定されているか。	次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>代入アクティビティで、コピー先の変数を設定します。</li> <li>データ変換アクティビティで、変換先の変数を設定します。</li> </ul>
49	並列処理開始アクティビティで、設定するリンク名の長さが 64 バイト以内か。	並列処理開始アクティビティで、定義されている、リンク名を次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>リンク名を 64 バイト以下に変更。</li> <li>使用する文字コードを変更。</li> </ul>

項番	検証内容	エラーになった場合の対処
50	分岐アクティビティで、分岐条件を一つ以上設定しているか。	分岐アクティビティで、一つ以上の分岐条件を設定します。
51	分岐アクティビティで、条件分岐またはデフォルトに割り当てられていない遷移先があるか。	分岐アクティビティで、遷移先を条件分岐またはデフォルトに割り当てます。
52	分岐アクティビティで、条件名が設定されているか。	分岐アクティビティで、分岐の条件名を設定します。
53	分岐アクティビティで、条件が設定されているか。	分岐アクティビティで、分岐の条件を設定します。
54	繰り返しアクティビティで、繰り返し条件が設定されているか。	繰り返しアクティビティで、繰り返し条件を設定します。
55	Java 呼出アクティビティで、Java クラス名が設定されているか。	Java 呼出アクティビティで、Java クラス名を設定します。
56	Java 呼出アクティビティで、有効な引数用割当変数が設定されているか。	Java 呼出アクティビティで、有効な引数用割当変数を設定します。
57	Java 呼出アクティビティで、有効な戻り値用割当変数が設定されているか。	Java 呼出アクティビティで、有効な戻り値用割当変数を設定します。
58	フォルト送出アクティビティで、有効な割当変数が設定されているか。	フォルト送出アクティビティで、有効な割当変数を設定します。
59	HCSCTE プロジェクトの lib ディレクトリにディレクトリがないか。	HCSCTE プロジェクトの lib ディレクトリからディレクトリを取り除きます。
60	HCSCTE プロジェクトの lib ディレクトリに次の名前のファイルがないか。 <ul style="list-style-type: none"> <li>• csbdef.jar</li> <li>• csbp_ejb.jar</li> <li>• csbjava.jar</li> </ul>	ファイルを削除するか、ファイル名を変更し、HCSCTE プロジェクトの lib ディレクトリに左記のファイルが含まれないようにします。

## (2) 定義されたビジネスプロセスが構造化されているかの検証

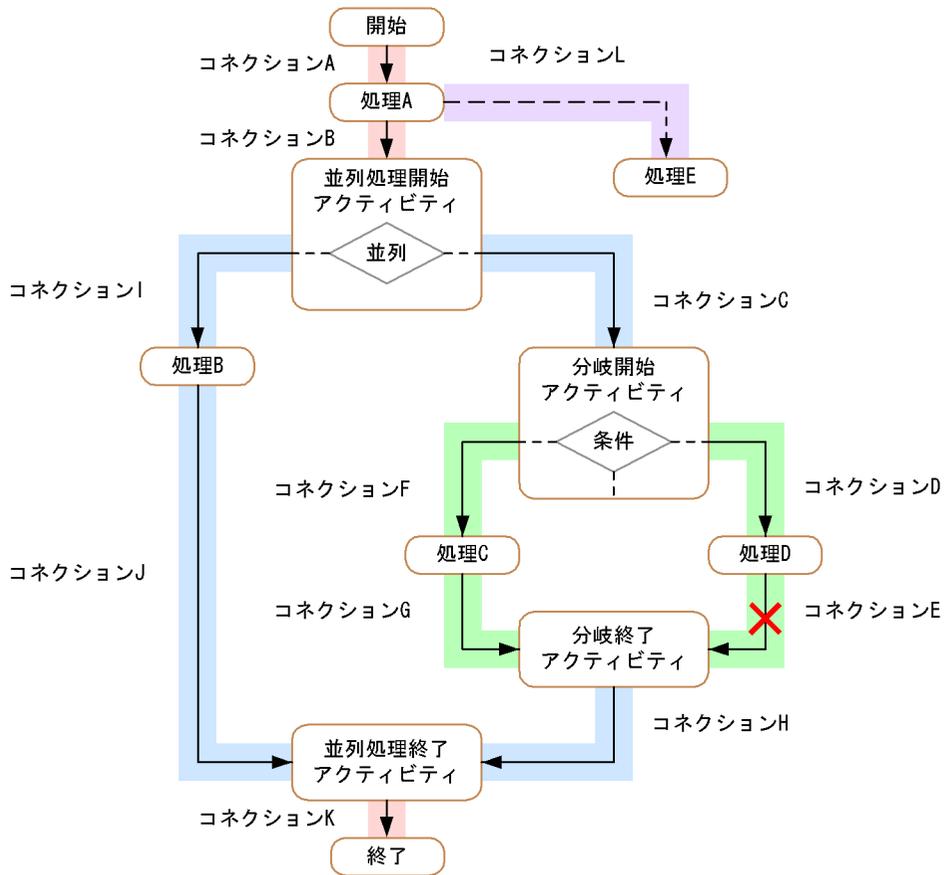
ビジネスプロセスが、コネクションを使用して構造化されているかどうかを検証されます。

検証でエラーになった場合、エラーとなったコネクションの元のコネクションもエラーとみなされます。

例えば、次の図のように、コネクション E がなかった場合は、分岐アクティビティや並列アクティビティなどに正常なコネクションが含まれていても、分岐アクティビティを元とするコネクション、並列アクティビティを元とするコネクション、および開始アクティビティを元とするコネクションすべてがエラーとみなされます。ただし、処理 D がフォルト送出アクティビティの場合、コネクション E は設定できないためエラーにはなりません。

6. ビジネスプロセスの定義

図 6-13 コネクションの例



(凡例)

- : アクティビティ
- : コネクション
- : フォルトコネクション
- : 開始アクティビティを元とするコネクション
- : 並列処理開始アクティビティを元とするコネクション
- : 分岐アクティビティを元とするコネクション
- : フォルト処理の開始点を元とするコネクション

定義されたビジネスプロセスの構造化の検証内容と対処を次の表に示します。

表 6-18 定義されたビジネスプロセスの構造化の検証内容と対処

項番	分類	検証内容	エラーになった場合の対処
1	次の接続先	遷移元が開始アクティビティの場合、次の接続先があるか。	開始アクティビティを先頭にして、接続移先のアクティビティを設定します。
2		遷移元が並列処理開始アクティビティの場合、次の接続先があるか。	並列処理開始アクティビティ、およびリンクコネクションで、接続するアクティビティを設定します。
3		遷移元が分岐開始アクティビティの場合、次の接続先があるか。	分岐開始アクティビティで、接続するアクティビティを設定します。
4		遷移元が上記以外のアクティビティの場合、次の接続先があるか。	遷移元のアクティビティに接続するアクティビティを設定します。
5	並列処理	並列処理が分岐終了アクティビティに接続していないか。	並列処理、対応する並列処理終了アクティビティに接続します。
6		次に示すコネクションが、並列処理終了アクティビティに接続していないか。 <ul style="list-style-type: none"> <li>開始アクティビティを元とするコネクション</li> <li>あるアクティビティのあるフォルト処理の開始点を元とするコネクション</li> </ul>	次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>並列処理終了アクティビティを削除。</li> <li>並列処理終了アクティビティを、別のアクティビティに変更。</li> </ul>
7		並列処理開始アクティビティからつながる並列処理終了アクティビティが複数ないか。	並列処理開始アクティビティを、対応する一つの並列処理終了アクティビティに接続します。
8		並列処理終了アクティビティが、対応する並列処理以外からのコネクションも受け付けていないか。	対応する並列処理を並列処理終了アクティビティに接続します。
9		並列処理が終了アクティビティに接続していないか。	並列処理を並列処理終了アクティビティに接続します。
10		並列処理開始アクティビティから直接、並列処理終了アクティビティに接続していないか。	並列処理開始アクティビティと、対応する並列処理終了アクティビティの間には、一つ以上のアクティビティを設定します。
11		分岐処理	分岐処理が並列処理終了アクティビティに接続していないか。
12	次に示すコネクションが、分岐処理終了アクティビティに接続していないか。 <ul style="list-style-type: none"> <li>開始アクティビティを元とするコネクション</li> <li>あるアクティビティのあるフォルト処理の開始点を元とするコネクション</li> </ul>		次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>分岐終了アクティビティを削除。</li> <li>分岐終了アクティビティを、別のアクティビティに変更。</li> </ul>
13	分岐開始アクティビティからつながる分岐終了アクティビティが複数ないか。		分岐開始アクティビティを、対応する一つに分岐終了アクティビティに接続します。
14	分岐終了アクティビティが、対応する分岐処理以外からのコネクションも受け付けていないか。		対応する分岐処理を分岐終了アクティビティに接続します。

## 6. ビジネスプロセスの定義

項番	分類	検証内容	エラーになった場合の対処
15		分岐処理が終了アクティビティに接続していないか。	分岐処理を分岐終了アクティビティに接続します。
16		分岐開始アクティビティから直接、分岐終了アクティビティに接続していないか。	分岐アクティビティと、対応する分岐終了アクティビティの間には、一つ以上のアクティビティを設定します。
17	フォルト処理	あるアクティビティのあるフォルト処理の開始点を元とするコネクションを処理中に、終了となっていないか。	フォルト処理の接続先から、終了アクティビティを削除します。

注 アクティビティを元とするコネクション、およびあるアクティビティのあるフォルト処理の開始点を元とするコネクションについては、「図 6-13 コネクションの例」を参照してください。

### (3) そのほかの検証

そのほかに次の表に示す内容が検証されます。

表 6-19 そのほかの検証内容と対処

項番	分類	検証内容	エラーになった場合の対処
1	ビジネスプロセス共通	ビジネスプロセスの最初に行われる基本アクティビティが、インスタンスを生成する受付アクティビティであるか。	開始アクティビティの次に、インスタンスを生成する受付アクティビティを接続します。
2		リンクコネクションに有効なリンク名が設定されているか。	リンク元の並列処理開始アクティビティで、リンク名を設定します。
3		代入アクティビティのコピー元とコピー先の変数の型が等しいか。	代入アクティビティで、コピー先とコピー元の変数の型を同じに設定します。
4		同一のオペレーション名を持つ応答アクティビティの変数の型が同一か。	同一のオペレーション名を持つ応答アクティビティで、同じ割当変数名を設定します。
5		同一オペレーションの応答アクティビティのフォルト名に対応付けられている変数が一つか。 (正しくない例) <ul style="list-style-type: none"> <li>• フォルト名 faultA に変数 variableX と variableY の二つが対応している場合</li> </ul> (正しい例) <ul style="list-style-type: none"> <li>• フォルト名 faultA に変数 variableX, フォルト名 faultB に変数 variableY が対応している場合</li> <li>• フォルト名 faultA に変数 variableX, フォルト名 faultB に変数 variableX が対応している場合</li> </ul>	フォルト処理の割当ダイアログで、同一オペレーションの応答アクティビティのフォルト名に対応する割当変数を一つに設定します。

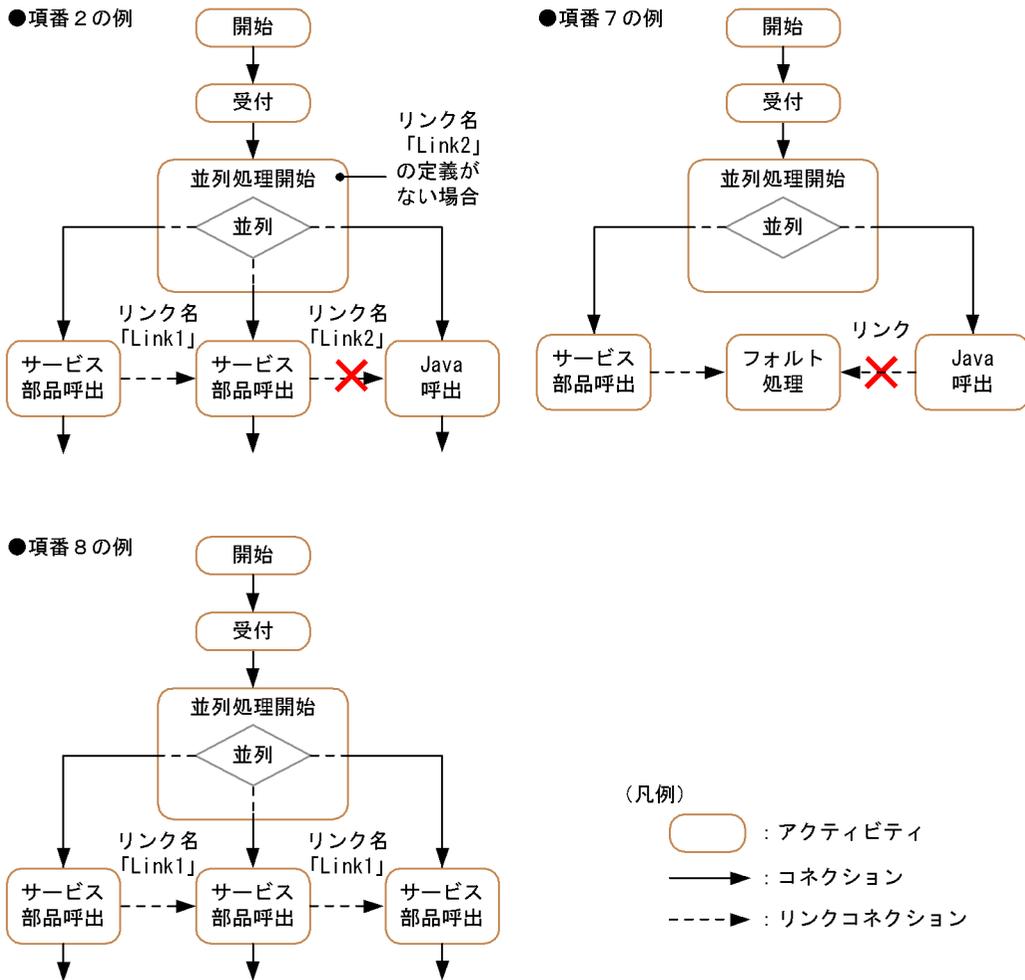
項番	分類	検証内容	エラーになった場合の対処
6		循環するリンクコネクションが設定されていないか。	リンクコネクションが循環しないように設定します。
7		フォルト処理外からフォルト処理内へのリンクコネクションが張られていないか。	フォルト処理からフォルト処理内へのリンクコネクションを削除するように設定します。
8		リンク名が複数回利用されていないか。	リンクコネクションで、リンク名がビジネスプロセス内で一意になるよう変更します。
9		同期の受付アクティビティのオペレーションに対応する応答アクティビティがあるか。	同期の受付アクティビティのオペレーションに対応する応答アクティビティを作成します。
10		応答アクティビティのオペレーションに対応する、同期の受付アクティビティがあるか。	次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>応答アクティビティのオペレーションに対応する同期の受付アクティビティを作成します。</li> <li>応答アクティビティのオペレーション名を変更します。</li> </ul>
11		インスタンスを生成するすべての受付アクティビティで、使用されている相関セットが同じか。	インスタンスを生成するすべての受付アクティビティで、使用する相関セットが同じになるように設定します。
12		インスタンスを生成する受付アクティビティがあるか。	次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>受付アクティビティで、インスタンス生成を「yes」に変更します。</li> <li>インスタンスを生成する受付アクティビティを作成します。</li> </ul>
13		ビジネスプロセスの開始点からコネクションを追っても到達できないアクティビティがあるか。	次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>到達できないアクティビティを削除します。</li> <li>到達できるようにコネクションを設定します。</li> </ul>
14		フォルト送出アクティビティから送出されたフォルト変数が、上位のスキープのフォルト処理でキャッチされるか。	次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>フォルト処理に適切な割当変数を設定します。</li> <li>フォルト処理に割当変数として「catch-all」を設定します。</li> </ul>
15	永続化しないビジネスプロセスの場合	永続化しないビジネスプロセスで、応答アクティビティのあとに実行されるアクティビティがないか。	応答アクティビティのあとにある実行されるアクティビティを削除します。
16		永続化しないビジネスプロセスに、非同期の受付アクティビティや非同期のサービス呼出アクティビティがないか。	次のどちらかの方法で対処します。 <ul style="list-style-type: none"> <li>非同期の受付アクティビティや非同期のサービス呼出アクティビティを同期に変更します。</li> <li>非同期の受付アクティビティや非同期のサービス呼出アクティビティを削除します。</li> </ul>

「表 6-19 そのほかの検証内容と対処」に示した検証の対象となるリンクの状態を次の図

## 6. ビジネスプロセスの定義

に示します。

図 6-14 検証の対象となるリンクの状態



### (4) 呼出サービス部品の検証

サービス呼出アクティビティに割り当てられているサービス部品のサービス名およびオペレーション名について検証されます。

呼出サービス部品の検証内容と対処を次の表に示します。

表 6-20 呼出サービス部品の検証内容と対処

項番	検証内容	エラーになった場合の対処
1	サービス呼出アクティビティに割り当てられているサービス名がリポジトリ内に実在するか。	サービス呼出アクティビティに、リポジトリ内に実在するサービス名を割り当ててください。
2	サービス呼出アクティビティに割り当てられているサービス名がリポジトリ内に実在する場合、アクティビティに割り当てられているオペレーション名が呼出先サービス部品に存在するか。	サービス呼出アクティビティに、リポジトリ内に実在するオペレーション名を割り当ててください。
3	サービス呼出アクティビティに割り当てられているサービス名およびオペレーション名がリポジトリ内に実在する場合、アクティビティに設定されている通信モデルが呼出先のサービス部品およびオペレーションの通信モデルと一致するかどうか。	サービス呼出アクティビティで、適切なサービス部品およびオペレーションを選択し直してください。

### (5) Java の検証

Java 呼出アクティビティで設定されている Java クラスについて検証されます。

Java の検証内容と対処を次の表に示します。

表 6-21 Java の検証内容と対処

項番	検証内容	エラーになった場合の対処
1	Java 呼出アクティビティの「Java クラス名」に指定されているクラスが、HCSCTE プロジェクトの classes ディレクトリ内に存在するか。	次のどちらかの対処を実施してください。 <ul style="list-style-type: none"> <li>「Java エディタ起動」から Java クラスを作成します。</li> <li>Java 呼出アクティビティに HCSCTE プロジェクトの classes ディレクトリ内に存在するクラスを指定します。</li> </ul>
2	Java 呼出アクティビティで指定されているクラスで、コンパイルエラーが検出されていないか。	Java 呼出アクティビティで指定されているクラスのコンパイルエラーを解消するか、コンパイルエラーが発生しないクラスを指定してください。

注 Java 呼出アクティビティで指定されているクラスだけが検証の対象です。そのクラスから参照しているほかのクラスは検証の対象外です。

### (6) 検証できない内容

ビジネスプロセスの検証ではエラーにならなくても、ビジネスプロセスを実行したときに初めてエラーとなる場合があります。このような場合、ログを採取し、ログに出力されたメッセージに従って対処します。ログの出力先については、マニュアル「Cosminexus システム運用ガイド」の J2EE サーバの稼働ログに関する内容を参照してください。

ビジネスプロセスの検証でエラーにならないが、ビジネスプロセスを実行したときにエ

## 6. ビジネスプロセスの定義

ラーになる例を次に示します。

- アクティビティが生成したインスタンスの数が 32,762 を超えた。
- サービスリクエストで指定したオペレーション名の受付アクティビティに到達できない。
- サービスリクエストで指定したオペレーション名の受付アクティビティが待ち受け状態となっている。
- サービスリクエストで指定したオペレーション名の受付アクティビティに対応する応答アクティビティに到達できない。
- プロセスインスタンスを生成する受付アクティビティの前に、受付アクティビティ以外の基本アクティビティを実行した。
- サービスリクエスト側で応答を待っていない状態で、応答アクティビティが実行された。
- サービスリクエストで指定した受付アクティビティのオペレーション名と、サービスリクエストに応答として返す応答アクティビティのオペレーション名が一致しない。
- 定められた回数を超えて繰り返しを実行した。

### 6.9.2 検証方法

検証する手順を次に示します。

1. ツリービューのサービス定義一覧で、右クリックします。  
サービス一覧のポップアップメニューが表示されます。
2. ポップアップメニューから「検証」を選択します。  
検証結果がコンソールビューに表示されます。

また、パッケージングを実施した場合も、自動的に検証されます。

なお、検証する前にビジネスプロセスが保存されていないときは、リソース保管ダイアログが表示されます。リソース保管ダイアログでは、保存されていないビジネスプロセスの定義を保存できます。

### 6.9.3 検証内容の表示

検証結果として、メッセージがコンソールビューに表示されます。メッセージに従って、修正します。

表示されるメッセージの種類を次の表に示します。

表 6-22 メッセージの種類

種類	説明
エラー	次のどちらかの場合に表示されます。 <ul style="list-style-type: none"><li>• 定義の内容が間違っている。</li><li>• 正しい構文で定義しているが、実行できない定義をしている。</li></ul>
警告	実行時にエラーとなる可能性がある定義をしている場合に表示されます。

種類	説明
情報	追加情報が表示されます。

## 6.10 ビジネスプロセスの削除

---

ビジネスプロセスをバージョンアップする場合などに、ビジネスプロセスを削除できません。

定義したビジネスプロセスは削除できます。

### (1) 削除方法

削除する手順を次に示します。次に示す二つの方法があります。

#### 方法 1

1. ツリービューのサービス定義一覧で、ビジネスプロセスを選択します。
2. [ Delete ] キーを押します。  
削除を確認するダイアログボックスが表示されます。
3. [ はい ] ボタンをクリックします。  
指定したビジネスプロセスが削除されます。

#### 方法 2

1. ツリービューのサービス定義一覧で、ビジネスプロセスを選択して右クリックします。  
サービス一覧のポップアップメニューが表示されます。
2. ポップアップメニューから「削除」を選択します。  
削除を確認するダイアログボックスが表示されます。
3. [ はい ] ボタンをクリックします。  
指定したビジネスプロセスが削除されます。

### ! 注意事項

バージョンアップ時、最新のビジネスプロセスだけを削除することはできません。すべてのバージョンのビジネスプロセスが削除されます。

---

### (2) 削除できないビジネスプロセス

次に示す場合、ビジネスプロセスは削除できません。

- すでに配備定義されている場合
- サービス呼出アクティビティで使用されている場合

# 7

## データ変換の定義

この章では、データ変換に必要な電文フォーマット定義ファイルおよびデータ変換（マッピング）定義について説明します。

---

7.1 データ変換に必要なファイルおよび定義

---

7.2 電文フォーマット定義ファイルの作成

---

7.3 データ変換（マッピング）定義

---

7.4 マッピングする

---

7.5 ファンクションを使用した値の加工

---

7.6 くり返しの設定

---

7.7 ノード条件を設定する

---

7.8 マッピングの条件

---

## 7.1 データ変換に必要なファイルおよび定義

---

データ変換に必要なファイルの作成および定義の手順を次に示します。

1. 電文フォーマット定義ファイルを作成します。  
詳細については、「7.2 電文フォーマット定義ファイルの作成」を参照してください。
2. データ変換（マッピング）を定義します。  
詳細については、「7.3 データ変換（マッピング）定義」を参照してください。

## 7.2 電文フォーマット定義ファイルの作成

データ変換（マッピング）定義の基となる電文フォーマット定義ファイルを作成します。電文フォーマット定義ファイルには、変換元の電文フォーマット定義ファイルと変換先の電文フォーマット定義ファイルがあります。

電文フォーマット定義ファイルの種類および作成方法については、「4. 電文フォーマットの作成」を参照してください。

作成した電文フォーマット定義ファイルは、サービスアダプタ定義画面またはビジネスプロセス定義画面で設定します。サービスアダプタ定義画面、ビジネスプロセス定義画面での設定方法については、それぞれ「5.3 アダプタの内容の定義」、「6.3 ビジネスプロセスの内容の定義」を参照してください。

電文フォーマット定義ファイルの形式

データ変換の定義に利用する電文フォーマット定義ファイルは、「2.5.5 XMLスキーマの適用範囲」に示す条件を満たしている必要があります。スキーマの条件の詳細については、「2.5.5 XMLスキーマの適用範囲」を参照してください。

画面に表示されない要素

annotation, appinfo, documentation, および notation は、変換元スキーマツリービューおよび変換先スキーマツリービューにツリー表示されません。

セパレータが設定されているバイナリ電文フォーマットに関する注意事項

セパレータが設定されているバイナリ電文フォーマットを利用する場合、次の点に注意が必要です。

- 出現回数に [ データの終端まで ] を定義した場合で、次の二つの条件を満たす場合、繰り返しによる空要素の生成は一つだけになります。
  - 弟要素が存在しない。
  - 中間区切り文字の直後に繰り返しの終了文字が出現する
- 入力するバイナリデータにセパレータ文字と同じ値のバイトデータが含まれている場合（マルチバイト文字の構成バイトの一部である場合も含む）、バイナリデータの解析に失敗します。
- エスケープ文字を付加しなくてもデータと認識される場合があるので注意してください。

例：開始文字と同じ文字列が、開始文字の直後に設定されている場合

## 7.3 データ変換（マッピング）定義

---

データ変換（マッピング）定義画面で、変換元の電文フォーマット定義ファイルおよび変換先の電文フォーマット定義ファイルを設定して、データ変換（マッピング）を定義します。データ変換（マッピング）定義画面は、アダプタ定義画面、またはビジネスプロセス定義画面のデータ変換アクティビティから表示します。データ変換（マッピング）定義画面、アダプタ定義画面、およびビジネスプロセス定義画面のデータ変換アクティビティの詳細については、「11. 開発環境で使用する画面」を参照してください。

### 7.3.1 データ変換（マッピング）の定義手順

データ変換（マッピング）定義画面でデータ変換（マッピング）を定義する手順を次に示します。

#### （1）新規にデータ変換（マッピング）を定義する場合

新規にデータ変換（マッピング）を定義する手順を次に示します。

1. アダプタ定義画面、またはビジネスプロセス定義画面のデータ変換アクティビティから、データ変換（マッピング）定義画面を表示します。  
データ変換（マッピング）定義画面が表示されたあと、[ルート要素選択] ダイアログが表示されます。  
データ変換（マッピング）定義画面の詳細については「11.2.4 データ変換（マッピング）定義画面」、[ルート要素選択] ダイアログの詳細については、「11.5.1 ルート要素選択ダイアログ」を参照してください。
2. すべてのスキーマ論理名のルート要素を選択します。  
[OK] ボタンをクリックできる状態になります。
3. [OK] ボタンをクリックします。  
変換元スキーマツリービューアおよび変換先スキーマツリービューアに、選択したルート要素のスキーマがツリー表示されます。
4. マッピングします。  
マッピングについては、「7.4 マッピングする」を参照してください。
5. データ変換（マッピング）定義画面の変換元スキーマツリービューア、マッピングビューア、または変換先スキーマツリービューアの適当な場所で右クリックして、[検証] を選択します。  
検証が実行されます。

**!** 注意事項

手順 3. で [キャンセル] ボタンをクリックした場合、変換元スキーマツリービューおよび変換先スキーマツリービューには、スキーマ論理名だけが表示されます。その場合、[ルート要素選択] ダイアログでもう 1 度ルート要素を選択してください。[ルート要素選択] ダイアログは、変換元スキーマツリービューまたは変換先スキーマツリービューのスキーマ論理名を右クリックして、[ルート要素選択] を選択すると表示されます。

**(2) 定義済みのデータ変換 (マッピング) を編集する場合**

定義済みのデータ変換 (マッピング) を編集する手順を次に示します。

1. アダプタ定義画面、またはビジネスプロセス定義画面のデータ変換アクティビティから、データ変換定義画面を表示します。  
定義済みの状態のデータ変換 (マッピング) 定義画面が表示されます。  
データ変換 (マッピング) 定義画面の詳細については、「11.2.4 データ変換 (マッピング) 定義画面」を参照してください。
2. 必要に応じてルート要素を変更します。  
ルート要素を変更する方法については、「7.3.1(3) ルート要素を変更する場合」を参照してください。
3. マッピングを編集します。  
マッピングについては、「7.4 マッピングする」を参照してください。
4. データ変換 (マッピング) 定義画面の変換元スキーマツリービュー、マッピングビュー、または変換先スキーマツリービューの適当な場所で右クリックして、[検証] を選択します。  
検証が実行されます。

**(3) ルート要素を変更する場合**

ルート要素は、ルート要素を選択してスキーマをツリー表示させたあとに、変更することができます。ルート要素を変更する手順を次に示します。

1. 変換元スキーマツリービューまたは変換先スキーマツリービューのスキーマ論理名を右クリックして、[ルート要素選択] を選択します。  
[ルート要素選択] ダイアログが表示されます。
2. スキーマ論理名のルート要素を変更します。
3. [OK] ボタンをクリックします。  
変換元スキーマツリービューおよび変換先スキーマツリービューに、変更したルート要素のスキーマがツリー表示されます。  
変更前の状態でマッピングしていた場合、変更前の編集内容が破棄されることを知らせるダイアログが表示されます。変更前の編集内容を破棄して、新しく選択したルート要素のスキーマをツリー表示する場合は、[OK] ボタンをクリックします。

## 7. データ変換の定義

### **!** 注意事項

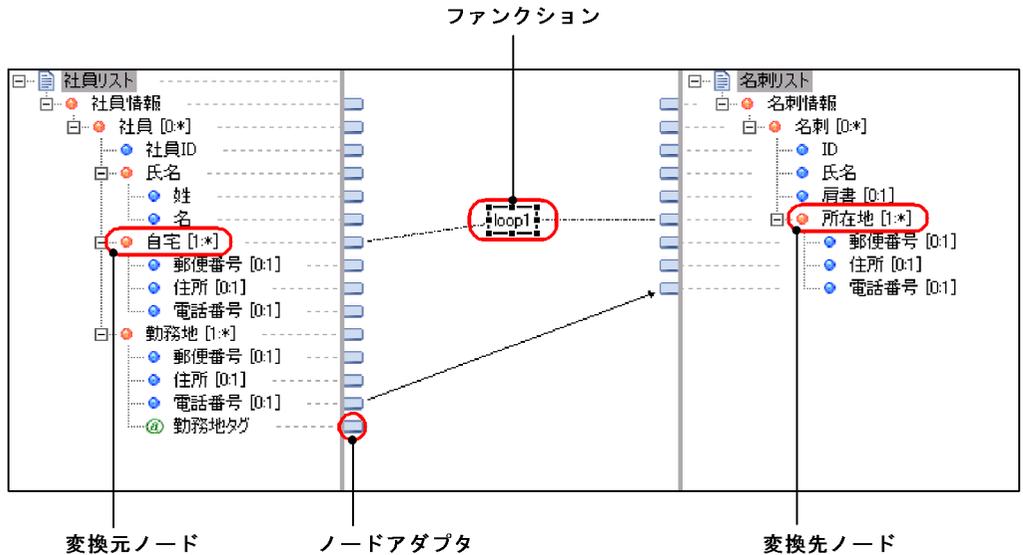
DBアダプタの定義の場合、変換元（DBアダプタ側）のルート要素は変更しないでください。ルート要素を変更した場合、エラーになります。

---

## 7.4 マッピングする

マッピングとは、変換元（マッピング元）の値を変換先（マッピング先）の値に対応づけることです。マッピングの画面を次に示します。

図 7-1 マッピングの画面



マッピングには、変換元ノードの値を変換先ノードに直接代入する場合と、変換元ノードの値を加工して変換先ノードに代入する場合があります。変換元ノードの値を加工する場合は、ファンクションを使用します。

また、マッピングの方法には、パレットのツールによる方法とダイアログによる方法があります。

### 7.4.1 変換元ノードの値を変換先ノードに直接マッピングする

#### (1) パレットのツールを使用する場合

パレットのツールを使用したマッピングの手順を次に示します。

1. パレットから [マッピング] を選択します。
2. マッピング元となる変換元ノードのノードアダプタをクリックします。  
マッピング元には、マッピング対象の変換元ノードを指定してください。マッピング元できないノードを指定しようとした場合、マウスポインタに  が付き、指定できない状態になります。

3. マッピング先となる変換先ノードのノードアダプタをクリックします。  
マッピング線  (代入線) が設定されます。なお、マッピング線の色は変更できません。マッピング線の色を変更する方法については、「11.9.4 マッピング線の色を変更する」を参照してください。  
また、マッピング先には、マッピング対象 かつ「7.8.2 マッピングできるノードおよびファンクションの対応」で指定された条件を満たす変換先ノードを指定してください。マッピング先にできないノードを指定しようとした場合、マウスポインタに  が付き、指定できない状態になります。

注

マッピング対象については、「7.8.1 マッピング対象とマッピング対象外」を参照してください。

## (2) ダイアログを使用する場合

[マッピング元設定] ダイアログを使用してマッピングします。[マッピング元設定] ダイアログを使用したマッピングの手順を次に示します。

1. 変換先スキーマツリービューアのマッピング先となる変換先ノードを右クリックして、[マッピング元] を選択します。  
[マッピング元設定] ダイアログが表示されます。
2. [ノード追加] ボタンをクリックします。  
[ノード選択] ダイアログが表示されます。
3. マッピング元となる変換元ノードを指定して、[OK] ボタンをクリックします。  
[パス/ファンクション名] に指定したノードのパス名が表示されます。パス名の表示形式の詳細については、「7.4.4 マッピング元の表示形式」を参照してください。  
マッピング元には、マッピング対象 の変換元ノードを指定してください。
4. [OK] ボタンをクリックします。  
マッピング線  (代入線) が設定されます。なお、マッピング線の色は変更できません。マッピング線の色を変更する方法については、「11.9.4 マッピング線の色を変更する」を参照してください。

注

マッピング対象については、「7.8.1 マッピング対象とマッピング対象外」を参照してください。

## 7.4.2 変換元ノードの値を加工して変換先ノードにマッピングする

変換元ノードの値を加工して変換先ノードにマッピングする場合、ファンクションを使用します。ファンクションの詳細については、「7.5 ファンクションを使用した値の加工」を参照してください。

## (1) パレットのツールを使用する場合

パレットのツールを使用したマッピングの手順を次に示します。

1. パレットから使用するファンクションを選択します。
2. マッピングビューアで適当な場所をクリックします。  
ファンクションがマッピングビューアに配置されます。配置されたファンクションは、ドラッグ&ドロップで移動できます。
3. パレットから [ マッピング ] を選択します。
4. マッピング元となる変換元ノードのノードアダプタまたはファンクションをクリックします。  
マッピング元には、次の変換元ノードまたはファンクションを指定してください。
  - 変換元ノード  
マッピング対象 <sup>1</sup> の変換元ノード
  - ファンクション  
出力側にマッピング線が設定されていないファンクション
 マッピング元にできないノードまたはファンクションを指定しようとした場合、マウスポインタに  が付き、指定できない状態になります。
5. マッピング先となる変換先ノードのノードアダプタまたはファンクションをクリックします。  
マッピング線  (代入線) が設定されます。くり返しファンクションを使用した場合は、マッピング線  (くり返し対応線) が設定されます。なお、マッピング線の色は変更できます。マッピング線の色を変更する方法については、「11.9.4 マッピング線の色を変更する」を参照してください。  
また、マッピング先には、次の変換先ノードまたはファンクションを指定してください。
  - 変換先ノード  
マッピング対象 <sup>1</sup> かつ「7.8.2 マッピングできるノードおよびファンクションの対応」で指定された条件を満たす変換先ノード
  - ファンクション  
「7.8.2 マッピングできるノードおよびファンクションの対応」および「7.8.3 接続できるマッピング線の数」で指定された条件を満たし、かつファンクション同士の接続で閉路 <sup>2</sup> とならないファンクション
 マッピング先にできないノードまたはファンクションを指定しようとした場合、マウスポインタに  が付き、指定できない状態になります。

注 1

マッピング対象については、「7.8.1 マッピング対象とマッピング対象外」を参照してください。

### 注 2

ファンクション A からファンクション B に、ファンクション B からファンクション C に、ファンクション C からファンクション A にのように、ファンクション同士の接続が閉路を作成するマッピングです。

### ! 注意事項

ファンクションを配置する前にマッピング線を設定してしまった場合、マッピング線を削除して、もう 1 度やり直してください。

### 参考

ファンクションをマッピングビューアに配置すると、ファンクション名の末尾に番号が付きます。この番号は、ファンクション名が重複しないように、マッピングした順に 1 から付けられる自然数の通番です。ファンクションが削除された場合、削除されたファンクションの番号は欠番となります。

## (2) ダイアログを使用する場合

ダイアログを使用したマッピングは、マッピング元とマッピング先の種類によって使用するダイアログが次のように異なります。

- マッピング元が変換元ノードでマッピング先がファンクションのとき  
マッピング先のファンクションを設定するダイアログを使用します。  
マッピング先が選択ファンクションの場合は、「7.5.14 条件によって異なる値を出力する」を参照してください。
- マッピング元およびマッピング先がファンクションのとき  
マッピング先のファンクションを設定するダイアログを使用します。  
マッピング先が選択ファンクションの場合は、「7.5.14 条件によって異なる値を出力する」を参照してください。
- マッピング元がファンクションでマッピング先が変換先ノードのとき  
[マッピング元設定] ダイアログを使用します。  
マッピング元がくり返しファンクションまたは選択ファンクションの場合は、[くり返し設定] ダイアログも使用できます。[くり返し設定] ダイアログによるマッピングの詳細については、「7.6.1 [くり返し設定] ダイアログを使用してマッピングする」を参照してください。

### 注

くり返しファンクションに接続されている、または何も接続されていない選択ファンクションになります。くり返しファンクション以外のファンクションに接続されている場合、[くり返し設定] ダイアログは使用できません。

## (a) マッピング元が変換元ノードでマッピング先がファンクションのとき

マッピング元が変換元ノードでマッピング先がファンクションのときのマッピングの手順を次に示します。

1. パレットから使用するファンクションを選択します。
2. マッピングビューアで適当な場所をクリックします。  
ファンクションがマッピングビューアに配置されます。配置されたファンクションは、ドラッグ&ドロップで移動できます。
3. 次のどちらかの方法でマッピング先のファンクションを設定するダイアログを表示させます。
  - マッピング先のファンクションを右クリックして、[ 設定 ] を選択します。
  - マッピング先のファンクションをダブルクリックします。
 マッピング先のファンクションを設定するダイアログが表示されます。
4. [ ノード追加 ] ボタンまたは [ ノード選択 ] ボタンをクリックします。  
[ ノード選択 ] ダイアログが表示されます。
5. マッピング元となる変換元ノードを指定して、[ OK ] ボタンをクリックします。  
マッピング先のファンクションを設定するダイアログの [ 入力 ] (くり返しファンクションの場合は [ 基準パス ]) に、マッピング元の変換元ノードが設定されます。変換元ノードはパス名で表示されます。パス名の表示形式の詳細については、「7.4.4 マッピング元の表示形式」を参照してください。  
マッピング元には、マッピング対象 かつ「7.8.2 マッピングできるノードおよびファンクションの対応」で指定された条件を満たす変換元ノードを指定してください。
6. [ OK ] ボタンをクリックします。  
マッピング線  (代入線) が設定されます。くり返しファンクションを使用した場合は、マッピング線  (くり返し対応線) が設定されます。なお、マッピング線の色は変更できます。マッピング線の色を変更する方法については、「11.9.4 マッピング線の色を変更する」を参照してください。

## 注

マッピング対象については、「7.8.1 マッピング対象とマッピング対象外」を参照してください。

## (b) マッピング元およびマッピング先がファンクションのとき

マッピング元およびマッピング先がファンクションのときのマッピングの手順を次に示します。

1. パレットから使用するファンクションを選択します。
2. マッピングビューアで適当な場所をクリックします。  
ファンクションがマッピングビューアに配置されます。配置されたファンクション

## 7. データ変換の定義

は、ドラッグ&ドロップで移動できます。

3. 次のどちらかの方法でマッピング先のファンクションを設定するダイアログを表示させます。

- マッピング先のファンクションを右クリックして、[ 設定 ] を選択します。
- マッピング先のファンクションをダブルクリックします。

マッピング先のファンクションを設定するダイアログが表示されます。

4. [ ファンクション追加 ] ボタンまたは [ ファンクション選択 ] ボタンをクリックします。

[ ファンクション選択 ] ダイアログが表示されます。

5. マッピング元となるファンクションを指定して、[ OK ] ボタンをクリックします。  
マッピング先のファンクションを設定するダイアログの [ 入力 ] に、マッピング元のファンクションが設定されます。定数ファンクションの場合はアポストロフィ ( ' ) で囲んだ定数値、定数ファンクション以外のファンクションの場合はファンクション名が表示されます。

マッピング元には、「7.8.2 マッピングできるノードおよびファンクションの対応」で指定された条件を満たし、かつ出力側にマッピング線が設定されていないファンクションを指定してください。

6. [ OK ] ボタンをクリックします。

マッピング線  ( 代入線 ) が設定されます。くり返しファンクションを使用した場合は、マッピング線  ( くり返し対応線 ) が設定されます。なお、マッピング線の色は変更できます。マッピング線の色を変更する方法については、「11.9.4 マッピング線の色を変更する」を参照してください。

### (c) マッピング元がファンクションでマッピング先が変換先ノードのとき

1. パレットから使用するファンクションを選択します。
2. マッピングビューアで適当な場所をクリックします。  
ファンクションがマッピングビューアに配置されます。配置されたファンクションは、ドラッグ&ドロップで移動できます。
3. 変換先スキーマツリービューアのマッピング先となる変換先ノードを右クリックして、[ マッピング元 ] を選択します。

[ マッピング元設定 ] ダイアログが表示されます。

4. [ ファンクション追加 ] ボタンをクリックします。  
[ ファンクション選択 ] ダイアログが表示されます。

5. マッピング元となるファンクションを指定して、[ OK ] ボタンをクリックします。

[ パス / ファンクション名 ] に指定したファンクション名が設定されます。

マッピング元には、「7.8.2 マッピングできるノードおよびファンクションの対応」で指定された条件を満たし、かつ出力側にマッピング線が設定されていないファンクションを指定してください。ただし、選択ファンクションで条件として指定したた

め、出力側にマッピング線 .....(条件線)が設定されているファンクションは指定できます。

6. [ OK ] ボタンをクリックします。

マッピング線 ——→(代入線)が設定されます。なお、マッピング線の色は変更できません。マッピング線の色を変更する方法については、「11.9.4 マッピング線の色を変更する」を参照してください。

#### ! 注意事項

ファンクションを配置する前にマッピング線を設定してしまった場合、マッピング線を削除して、もう1度やり直してください。

#### 参考

ファンクションをマッピングビューアに配置すると、ファンクション名の末尾に番号が付きます。この番号は、ファンクション名が重複しないように、マッピングした順に1から付けられる自然数の通番です。ファンクションが削除された場合、削除されたファンクションの番号は欠番となります。

## 7.4.3 マッピングを取り消す

マッピング線やファンクションを削除して、マッピングを取り消すことができます。

### (1) マッピング線を削除する

次のどちらかの方法で、マッピング線を削除します。

#### 方法 1

削除したいマッピング線を右クリックして、[ 削除 ] を選択します。

#### 方法 2

削除したいマッピング線を選択して、[ Delete ] キーを押します。

#### ! 注意事項

条件線はユーザが削除できません。条件線は、条件の設定を取り消した場合に自動的に削除されます。

### (2) ファンクションを削除する

次のどちらかの方法で、ファンクションを削除します。

## 7. データ変換の定義

### 方法 1

削除したいファンクションを右クリックして、[ 削除 ] を選択します。

### 方法 2

削除したいファンクションを選択して、[ Delete ] キーを押します。

### ! 注意事項

条件としてファンクションを指定 ([ 条件設定 ] ダイアログの [ 条件 ] でファンクションを指定) した場合、そのファンクションは削除できません。

## 7.4.4 マッピング元の表示形式

[ マッピング元設定 ] ダイアログや各ファンクションを設定するダイアログで、マッピング元となる変換元ノードまたはファンクションを指定した場合、次のように表示されません。

### (1) 変換元ノードの場合

スキーマツリーのノードパスが表示されます。ノードパスは次のように表示されます。

#### (a) 変換元ノードに条件が設定されているとき

条件が設定されたノード名の後ろに、角括弧 ( [ ] ) で囲まれた条件が表示されます。

(例) /aa/bb[position()= ' 1 ']/cc

条件としてファンクションを指定 ([ 条件設定 ] ダイアログの [ 条件 ] でファンクションを指定) した場合、ファンクション名が表示されます。

(例) /aa/bb[length1= ' 5 ']

#### (b) スキーマ論理名が複数あるとき

ノードパスの先頭に、「\$< スキーマ論理名 >」が表示されます。

(例) \$source1/aa/bb/cc

#### (c) 属性のとき

ノード名の前に「@」が表示されます。

(例) /aa/bb/@cc

### (2) ファンクションの場合

定数ファンクションのときと定数ファンクション以外のファンクションのときで、表示形式が異なります。

(a) 定数関クションのとき

[ 文字列 ] または [ 数値 ] を選択

アポストロフィ ( ' ) で囲んだ定数値が表示されます。

[ 論理値 ] を選択

次のように表示されます。

true の場合 : true()

false の場合 : false()

[ 特殊ノード ] を選択

ファンクション名が表示されます。

(b) 定数関クション以外のファンクションのとき

ファンクション名が表示されます。

## 7.4.5 マッピングの注意事項

マッピングしたあとに、データ変換 ( マッピング ) 定義の内容が妥当かどうか検証をします。検証でエラーが発生しないように、次の点を考慮してマッピングしてください。

- 「7.8 マッピングの条件」に従ってマッピングしてください。
- マッピング元の変換元ノードとマッピング先の変換先ノードの出現回数は一致させてください。
- 最少出現回数が 2 回以上に設定されている複合内容の要素は、必ずマッピングしてください。ただし、祖先に最少出現回数が 0 回のノードが出現する場合は除きます。
- 最小出現回数が 1 回以上に設定されている単純内容の要素は、必ずマッピングしてください。
- choice 要素の子ノードまたは子孫ノードに対してマッピングする場合、複数の子要素が同時に出現するか、またはどの子要素も出現しないようなマッピングを定義しないでください。
- choice 要素の複数の子要素または子孫ノードに対してマッピングする場合は、複数の子要素が同時に出現しないようにしてください。
- choice 要素のすべての子要素の最少出現回数が 1 回以上に設定されていて、一つの子要素の子孫要素がマッピングされている場合、その子要素に対して選択関クションを使用するときは、複数の子要素が同時に出現しないようにしてください。
- マッピング先の変換先ノードには、次の名前空間を定義しないでください。
  - targetNamespace に xsi と xsi の URI が指定されている
  - targetNamespace の名前空間修飾名が x から始まる
- インスタンス属性 ( xsi:nil , xsi:type , xsi:schemaLocation , xsi:noNamespaceSchemaLocation ) は、マッピング元の変換元ノードで定義されていても、マッピング先の変換先ノードには出現しません。

## 7.5 ファンクションを使用した値の加工

変換元ノードの値を加工して変換先ノードにマッピングする場合、ファンクションを使用します。ここでは、どのような場合にどのファンクションを使用するかを説明します。ファンクションを使用したマッピング方法については、「7.4.2 変換元ノードの値を加工して変換先ノードにマッピングする」を参照してください。

ファンクションの一覧を次の表に示します。

表 7-1 ファンクションの一覧

分類	ファンクション名	説明
文字列系	文字列連結	複数の文字列を連結します。
	部分文字列取得	一つの文字列から一部分の文字列を取り出します。
	文字数取得	文字列の文字数をマッピング先に代入します。
	文字列調査	文字列に指定した文字列が含まれているか、または指定した文字列で開始されているかを調査して、マッピング先に論理値を代入します。
	空白除去	文字列の先頭と末尾の空白を除去します。また、文字列間の連続した空白を一つの半角スペースに置換します。
数値系	数値フォーマット変換	数値をフォーマット変換します。
	数値演算	数値を演算 (+, -, *, div, mod) します。
	端数処理	小数点以下の端数を処理 (四捨五入, 切り捨て, および切り上げ) します。
	総和	複数のノードセットのノードの数値を合計します。
ノード系	ノード数取得	ノード数をマッピング先に代入します。
	ノード名取得	ノード名をマッピング先に代入します。
	ノード調査	指定した変換元ノードが存在するかを調査し、マッピング先にその論理値を代入します。
制御系	くり返し	くり返しを対応づけます。
	選択	条件によって異なる値を出力します。
その他	定数	指定した値をマッピング先に代入します。
	値置換え	変換テーブルの指定に基づいて、マッピング元の値を変換してマッピング先に代入します。

### 7.5.1 複数の文字列を連結する

複数のマッピング元の文字列を連結して、一つの文字列にします。

#### (1) 使用するファンクション

複数のマッピング元の文字列を連結するには、文字列連結ファンクションを使用します。文字列連結ファンクションを使用した例を次に示します。

図 7-2 文字列連結ファンクションの使用例



(凡例)

→ : マッピング元とマッピング先の接続

## (2) 設定手順

複数のマッピング元の文字列を連結して、一つの文字列にする手順を次に示します。

1. パレットから文字列連結ファンクション (concat) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [文字列連結] ダイアログを表示させます。
  - 文字列連結ファンクションを右クリックして、[設定] を選択します。
  - 文字列連結ファンクションをダブルクリックします。

[文字列連結] ダイアログが表示されます。

[文字列連結] ダイアログの詳細については、「11.5.9 文字列連結ダイアログ」を参照してください。

4. [入力] に連結させたい文字列のマッピング元を追加する場合、次のように設定します。
  - マッピング元が変換元ノードの場合  
[ノード追加] ボタンをクリックすると、[ノード選択] ダイアログが表示されます。[ノード選択] ダイアログで、マッピング元となる変換元ノードを選択します。
  - マッピング元がファンクションの場合  
[ファンクション追加] ボタンをクリックすると、[ファンクション選択] ダイアログが表示されます。[ファンクション選択] ダイアログで、マッピング元となるファンクションを選択します。

[入力] に複数のマッピング元を指定した場合、[入力] に表示された連結対象は上から順に連結されます。一番上に表示された連結対象は、連結した文字列の左側に設定されます。

5. [入力] に変換元ノードが指定されている場合、その変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたいとき、[ノード条件設定] ボタンをクリックします。

[ノード条件設定] ダイアログが表示されます。

ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

## 7. データ変換の定義

い。また、[ノード条件設定]ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。

6. [OK] ボタンをクリックします。

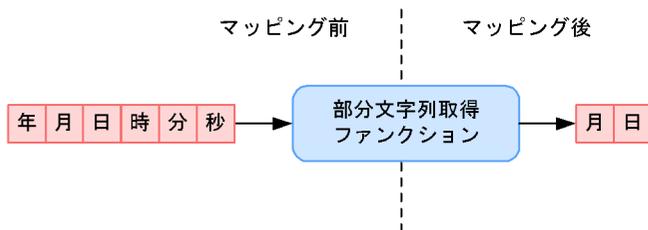
### 7.5.2 一つの文字列から一部分の文字列を取り出す

一つのマッピング元の文字列から、一部分の文字列を取り出します。

#### (1) 使用するファンクション

一つのマッピング元の文字列から、一部分の文字列を取り出すには、部分文字列取得ファンクションを使用します。部分文字列取得ファンクションを使用した例を次に示します。

図 7-3 部分文字列取得ファンクションの使用例



(凡例)

→ : マッピング元とマッピング先の接続

#### (2) 設定手順

一つのマッピング元の文字列から、一部分の文字列を取り出す手順を次に示します。

1. パレットから部分文字列取得ファンクション (substr) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [部分文字列取得] ダイアログを表示させます。
  - 部分文字列取得ファンクションを右クリックして [設定] を選択します。
  - 部分文字列取得ファンクションをダブルクリックします。

[部分文字列取得] ダイアログが表示されます。

[部分文字列取得] ダイアログの詳細については、「11.5.10 部分文字列取得ダイアログ」を参照してください。

4. [入力] に変換元ノードが指定されている場合、その変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたいとき、[ノード条件設定] ボタンをクリックします。

[ノード条件設定] ダイアログが表示されます。

ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また、「[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。

5. 取り出す文字列を指定します。
  - 取り出す範囲を文字列の先頭から指定する場合  
 [ 先頭からの範囲指定 ] ラジオボタンを選択し、[ 開始位置 ] に取り出す文字列の開始文字の位置を指定します。取り出す文字列の文字数を指定する場合、[ 文字数 ] に取り出す文字数を指定します。[ 開始位置 ] で指定した開始文字の位置から末尾までの文字列を取り出す場合、[ 開始位置から末尾までを取得 ] チェックボックスをチェックします。
  - 取り出す範囲を文字列の末尾から指定する場合  
 [ 末尾からの範囲指定 ] ラジオボタンを選択し、[ 開始位置 ] に取り出す文字列の開始文字の位置を指定します。取り出す文字列の文字数を指定する場合、[ 文字数 ] に取り出す文字数を指定します。[ 開始位置 ] で指定した開始文字の位置から先頭までの文字列を取り出す場合、[ 開始位置から先頭までを取得 ] チェックボックスをチェックします。
  - 文字列を二つに分割して、分割した前の部分があとの部分かのどちらかを指定する場合  
 [ 分割文字列指定 ] ラジオボタンを選択し、[ 分割文字列 ] にマッピング元の文字列を分割する文字（文字列）を指定します。先頭の文字から [ 分割文字列 ] で指定した文字（文字列）より前の文字列を取り出したい場合、[ 取得部位 ] に「前」を指定します。[ 分割文字列 ] で指定した文字（文字列）のあとから末尾までの文字列を取り出したい場合、[ 取得部位 ] に「後」を指定します。
6. [ OK ] ボタンをクリックします。

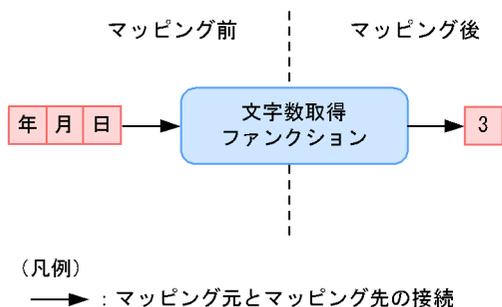
### 7.5.3 文字列の文字数を代入する

マッピング元の文字列の文字数をマッピング先に代入します。

#### (1) 使用するファンクション

マッピング元の文字列の文字数をマッピング先に代入するには、文字数取得ファンクションを使用します。文字数取得ファンクションを使用した例を次に示します。

図 7-4 文字数取得ファンクションの使用例



## (2) 設定手順

マッピング元の文字列の文字数をマッピング先に代入する手順を次に示します。

1. パレットから文字数取得ファンクション (length) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [文字数取得] ダイアログを表示させます。
  - 文字数取得ファンクションを右クリックして、[設定] を選択します。
  - 文字数取得ファンクションをダブルクリックします。

[文字数取得] ダイアログが表示されます。

[文字数取得] ダイアログの詳細については、「11.5.11 文字数取得ダイアログ」を参照してください。

4. [入力] に変換元ノードが指定されている場合、その変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたいとき、[ノード条件設定] ボタンをクリックします。

[ノード条件設定] ダイアログが表示されます。

ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また、[ノード条件設定] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。

5. [OK] ボタンをクリックします。

## 7.5.4 指定文字列の有無、または指定文字列の開始を調査する

マッピング元の文字列に指定した文字列が含まれているか、または指定した文字列で開始されているかを調査して、マッピング先にその論理値を代入します。

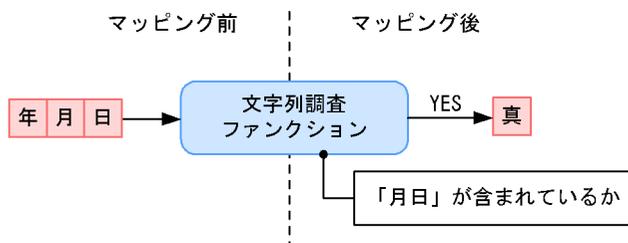
### (1) 使用するファンクション

マッピング元の文字列に指定した文字列が含まれているか、または指定した文字列で開

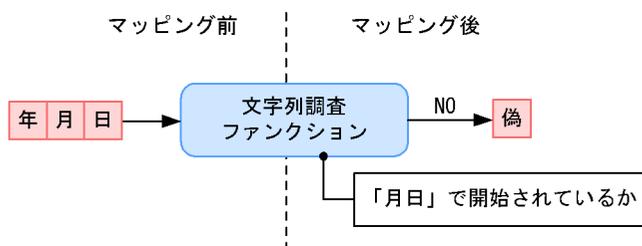
始されているかを調査して、マッピング先にその論理値を代入するには、文字列調査ファンクションを使用します。文字列調査ファンクションを使用した例を次に示します。

図 7-5 文字列調査ファンクションの使用例

- 指定した文字列が含まれているか調査する場合



- 指定した文字列で開始されているか調査する場合



(凡例)

→ : マッピング元とマッピング先の接続

## (2) 設定手順

マッピング元の文字列を調査し、マッピング先に論理値を代入する手順を次に示します。

1. パレットから文字列調査ファンクション (contain) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [文字列調査] ダイアログを表示させます。
  - 文字列調査ファンクションを右クリックして、[設定] を選択します。
  - 文字列調査ファンクションをダブルクリックします。

[文字列調査] ダイアログが表示されます。

[文字列調査] ダイアログの詳細については、「11.5.12 文字列調査ダイアログ」を参照してください。

4. [入力] に変換元ノードが指定されている場合、その変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたいとき、[ノード条件設定] ボタンをク

## 7. データ変換の定義

リックします。

[ ノード条件設定 ] ダイアログが表示されます。

ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また,[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。

5. [ 調査種別 ] に次のどちらかを選択します。
  - [ 調査文字列 ] で指定した文字列が含まれているか調査する場合  
「指定文字列を含む」を選択します。
  - [ 調査文字列 ] で指定した文字列で開始されているか調査する場合  
「指定文字列から開始」を選択します。
6. [ 調査文字列 ] に調査対象となる文字列を指定します。
7. [ OK ] ボタンをクリックします。

### 7.5.5 文字列の空白を除去する

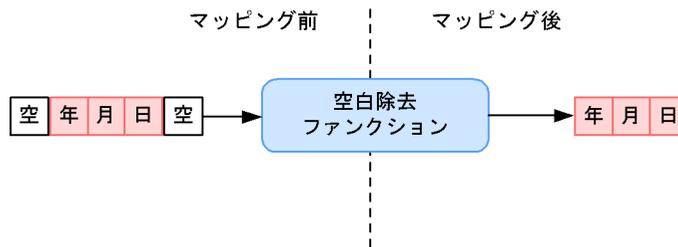
マッピング元の文字列の先頭と末尾の空白（半角スペース，タブ，復帰（Carriage Return），および改行（Line Feed））を除去します。また，文字列間の連続した空白を一つの半角スペースに置換します。

#### （1）使用するファンクション

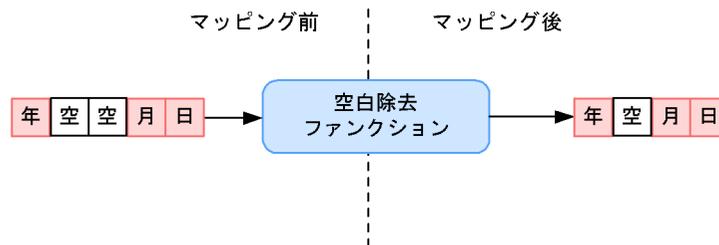
文字列の空白を除去するには，空白除去ファンクションを使用します。空白除去ファンクションを使用した例を次に示します。

図 7-6 空白除去ファンクションの使用例

- 文字列の先頭と末尾の空白を除去する場合



- 文字列間の連続した空白を一つの半角スペースに置換する場合



(凡例)

→ : マッピング元とマッピング先の接続

## (2) 設定手順

文字列の空白を除去する手順を次に示します。

1. パレットから空白除去ファンクション (trim) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [空白除去] ダイアログを表示させます。
  - 空白除去ファンクションを右クリックして、[設定] を選択します。
  - 空白除去ファンクションをダブルクリックします。

[空白除去] ダイアログが表示されます。

[空白除去] ダイアログの詳細については、「11.5.13 空白除去ダイアログ」を参照してください。

4. [入力] に変換元ノードが指定されている場合、その変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたいとき、[ノード条件設定] ボタンをクリックします。
 

[ノード条件設定] ダイアログが表示されます。

ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また、[ノード条件設定] ダイアログについては、「11.5.6 ノード条件設定ダイ

## 7. データ変換の定義

アログ」を参照してください。

5. [OK] ボタンをクリックします。

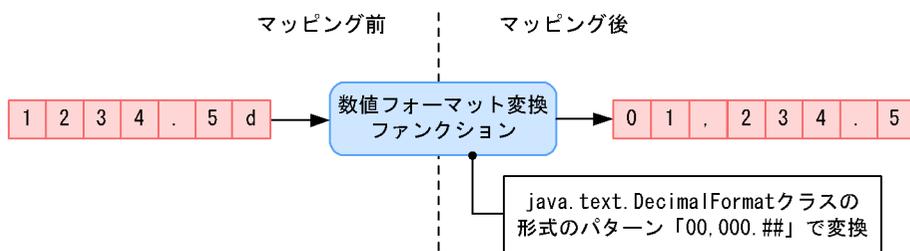
### 7.5.6 数値をフォーマット変換する

マッピング元の数値を `java.text.DecimalFormat` クラスの形式のパターンでフォーマット変換します。また、パターンに指定する小数点文字または桁区切り文字をデフォルトから変更できます。`java.text.DecimalFormat` クラスの形式のパターンの詳細については、Java の API に関するドキュメントを参照してください。

#### (1) 使用するファンクション

マッピング元の数値を `java.text.DecimalFormat` クラスの形式のパターンでフォーマット変換するには、数値フォーマット変換ファンクションを使用します。数値フォーマット変換ファンクションを使用した例を次に示します。

図 7-7 数値フォーマット変換ファンクションの使用例



(凡例)

→ : マッピング元とマッピング先の接続

#### (2) 設定手順

マッピング元の数値を `java.text.DecimalFormat` クラスの形式のパターンでフォーマット変換する手順を次に示します。

1. パレットから数値フォーマット変換ファンクション (format) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [数値フォーマット変換] ダイアログを表示させます。
  - 数値フォーマット変換ファンクションを右クリックして、[設定] を選択します。
  - 数値フォーマット変換ファンクションをダブルクリックします。

[数値フォーマット変換] ダイアログが表示されます。

[数値フォーマット変換] ダイアログの詳細については、「11.5.14 数値フォーマット変換ダイアログ」を参照してください。

4. [入力] に変換元ノードが指定されている場合、その変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたいとき、[ノード条件設定] ボタンをクリックします。  
[ノード条件設定] ダイアログが表示されます。  
ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また、[ノード条件設定] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。
5. [パターン] に変換後の `java.text.DecimalFormat` クラスの形式のパターン文字列を指定します。
6. 小数点文字または桁区切り文字をデフォルトから変更する場合、[記号を変更する] チェックボックスをチェックし、次のように指定します。
  - 小数点文字を変更する場合  
[小数点文字] に変更後の小数点文字の記号を指定します。省略した場合、デフォルトの記号となります。
  - 桁区切り文字を変更する場合  
[桁区切り文字] に変更後の桁区切り文字の記号を指定します。省略した場合、デフォルトの記号となります。
7. [OK] ボタンをクリックします。

### 7.5.7 数値を演算する

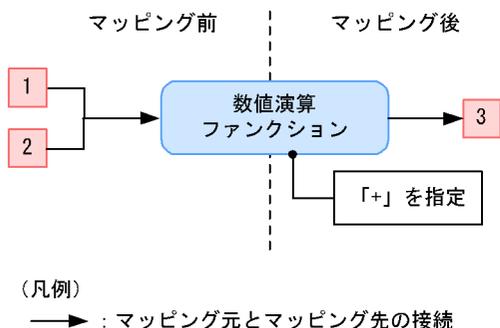
二つのマッピング元の数値に対して、指定した演算をします。指定できる演算の種類を次に示します。

- + (加算)
- - (減算)
- \* (乗算)
- / (除算)
- % (剰余)

#### (1) 使用するファンクション

二つのマッピング元の数値に対して、指定した演算をするには、数値演算ファンクションを使用します。数値演算ファンクションを使用した例を次に示します。

図 7-8 数値演算ファンクションの使用例



## (2) 設定手順

二つのマッピング元の数値に対して、指定した演算をする手順を次に示します。

1. パレットから数値演算ファンクション (calc) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [数値演算] ダイアログを表示させます。
  - 数値演算ファンクションを右クリックして、[設定] を選択します。
  - 数値演算ファンクションをダブルクリックします。

[数値演算] ダイアログが表示されます。

[数値演算] ダイアログの詳細については、「11.5.15 数値演算ダイアログ」を参照してください。

4. [演算] に実行したい演算を指定します。
5. [入力 1] または [入力 2] に変換元ノードが指定されている場合、その変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたいとき、[ノード条件設定] ボタンをクリックします。

[ノード条件設定] ダイアログが表示されます。

ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また、[ノード条件設定] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。
6. [OK] ボタンをクリックします。

### 7.5.8 小数点以下の端数を処理する

マッピング元の数値が小数の場合、小数点以下の端数を処理します。指定できる端数の処理を次に示します。

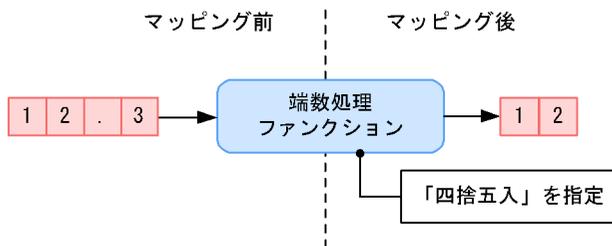
- 四捨五入
- 切り捨て

- 切り上げ

## (1) 使用するファンクション

小数点以下の端数を処理するには、端数処理ファンクションを使用します。端数処理ファンクションを使用した例を次に示します。

図 7-9 端数処理ファンクションの使用例



(凡例)

→ : マッピング元とマッピング先の接続

## (2) 設定手順

小数点以下の端数を処理する手順を次に示します。

1. パレットから端数処理ファンクション (round) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [ 端数処理 ] ダイアログを表示させます。
  - 端数処理ファンクションを右クリックして、[ 設定 ] を選択します。
  - 端数処理ファンクションをダブルクリックします。

[ 端数処理 ] ダイアログが表示されます。

[ 端数処理 ] ダイアログの詳細については、「11.5.16 端数処理ダイアログ」を参照してください。

4. [ 入力 ] に指定されている変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたい場合、[ ノード条件設定 ] ボタンをクリックします。  
[ ノード条件設定 ] ダイアログが表示されます。  
ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また、[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。
5. [ 種別 ] に実行したい端数の処理を指定します。
6. [ OK ] ボタンをクリックします。

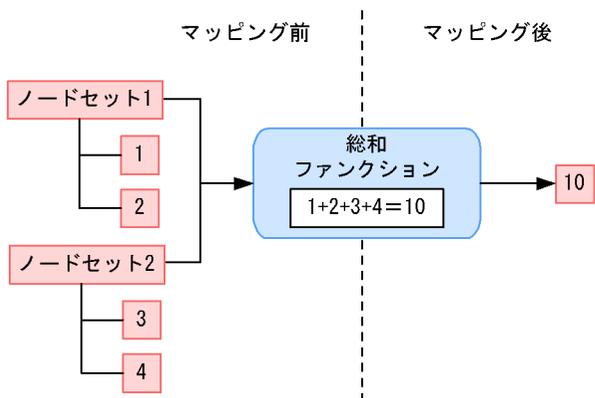
## 7.5.9 複数のノードセットのノードの数値を合計する

複数のマッピング元のノードセットを指定し、そのノードセットに含まれているノードの数値を合計します。

### (1) 使用するファンクション

複数のノードセットのノードの数値を合計するには、総和ファンクションを使用します。総和ファンクションを使用した例を次に示します。

図 7-10 総和ファンクションの使用例



(凡例)

→ : マッピング元とマッピング先の接続

### (2) 設定手順

複数のノードセットのノードの数値を合計する手順を次に示します。

1. パレットから総和ファンクション (sum) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [ 総和 ] ダイアログを表示させます。
  - 総和ファンクションを右クリックして、[ 設定 ] を選択します。
  - 総和ファンクションをダブルクリックします。

[ 総和 ] ダイアログが表示されます。

[ 総和 ] ダイアログの詳細については、「11.5.17 総和ダイアログ」を参照してください。

4. [ 入力 ] に合計させたい数値のマッピング元を追加する場合、[ ノード追加 ] ボタンをクリックします。  
[ ノード選択 ] ダイアログが表示されます。

5. マッピング元となるノードセットを指定します。  
[入力] に合計させたい数値のマッピング元が設定されます。
6. [入力] に指定されている変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたい場合、[ノード条件設定] ボタンをクリックします。  
[ノード条件設定] ダイアログが表示されます。  
ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また、[ノード条件設定] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。
7. [OK] ボタンをクリックします。

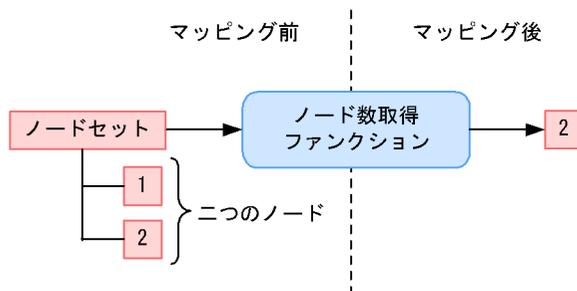
## 7.5.10 ノード数を代入する

マッピング元のノードセットのノード数をマッピング先に代入します。

### (1) 使用するファンクション

マッピング元のノードセットのノード数をマッピング先に代入するには、ノード数取得ファンクションを使用します。ノード数取得ファンクションを使用した例を次に示します。

図 7-11 ノード数取得ファンクションの使用例



(凡例)

→ : マッピング元とマッピング先の接続

### (2) 設定手順

マッピング元のノードセットのノード数をマッピング先に代入する手順を次に示します。

1. パレットからノード数取得ファンクション (count) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [ノード数取得] ダイアログを表示させます。
  - ノード数取得ファンクションを右クリックして、[設定] を選択します。
  - ノード数取得ファンクションをダブルクリックします。

## 7. データ変換の定義

[ ノード数取得 ] ダイアログが表示されます。

[ ノード数取得 ] ダイアログの詳細については、「11.5.18 ノード数取得ダイアログ」を参照してください。

4. [ 入力 ] に指定されている変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたい場合、[ ノード条件設定 ] ボタンをクリックします。

[ ノード条件設定 ] ダイアログが表示されます。

ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また、[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。

5. [ OK ] ボタンをクリックします。

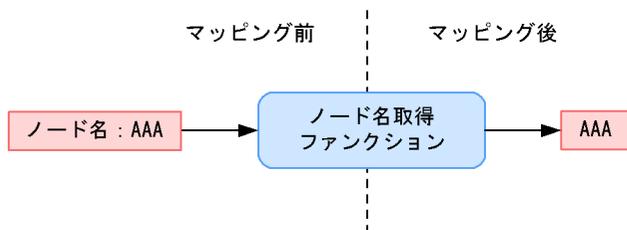
### 7.5.11 ノード名を代入する

マッピング元のノード名をマッピング先に代入します。マッピング元（変換元ノード）のインスタンスが存在しない場合、空文字列を代入します。

#### (1) 使用するファンクション

マッピング元のノード名をマッピング先に代入するには、ノード名取得ファンクションを使用します。ノード名取得ファンクションを使用した例を次に示します。

図 7-12 ノード名取得ファンクションの使用例



(凡例)

→ : マッピング元とマッピング先の接続

#### (2) 設定手順

マッピング元のノード名をマッピング先に代入する手順を次に示します。

1. パレットからノード名取得ファンクション (name) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [ ノード名取得 ] ダイアログを表示させます。
  - ノード名取得ファンクションを右クリックして、[ 設定 ] を選択します。
  - ノード名取得ファンクションをダブルクリックします。

- [ ノード名取得 ] ダイアログが表示されます。  
 [ ノード名取得 ] ダイアログの詳細については、「11.5.19 ノード名取得ダイアログ」を参照してください。
4. [ 入力 ] に指定されている変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたい場合、[ ノード条件設定 ] ボタンをクリックします。  
 [ ノード条件設定 ] ダイアログが表示されます。  
 ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また、[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。
5. [ OK ] ボタンをクリックします。

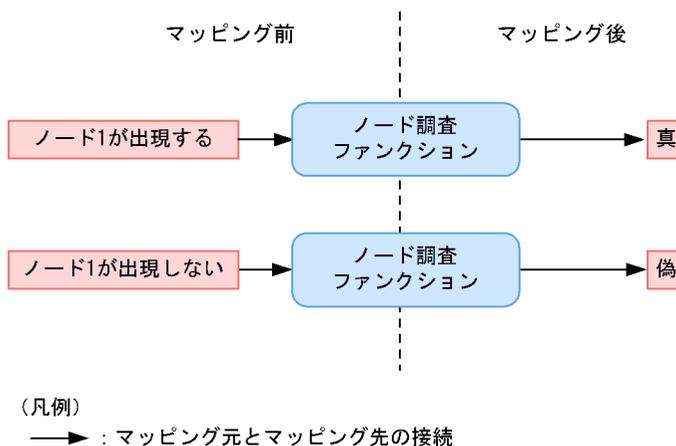
## 7.5.12 ノードの有無を調査する

指定したマッピング元の変換元ノードが存在するかを調査して、マッピング先にその論理値を代入します。

### (1) 使用するファンクション

指定したマッピング元の変換元ノードが存在するかを調査して、マッピング先にその論理値を代入するには、ノード調査ファンクションを使用します。ノード調査ファンクションを使用した例を次に示します。

図 7-13 ノード調査ファンクションの使用例



### (2) 設定手順

指定したマッピング元の変換元ノードが存在するか調査する手順を次に示します。

1. パレットからノード調査ファンクション (exist) を選択して、マッピングビューアに配置します。

## 7. データ変換の定義

2. マッピング線を設定します。
3. 次のどちらかの方法で [ ノード調査 ] ダイアログを表示させます。
  - ノード調査ファンクションを右クリックして、[ 設定 ] を選択します。
  - ノード調査ファンクションをダブルクリックします。

[ ノード調査 ] ダイアログが表示されます。  
[ ノード調査 ] ダイアログの詳細については、「11.5.20 ノード調査ダイアログ」を参照してください。
4. [ 入力 ] に指定されている変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたい場合、[ ノード条件設定 ] ボタンをクリックします。

[ ノード条件設定 ] ダイアログが表示されます。  
ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また、[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。
5. [ OK ] ボタンをクリックします。

### 7.5.13 くり返しを対応づける

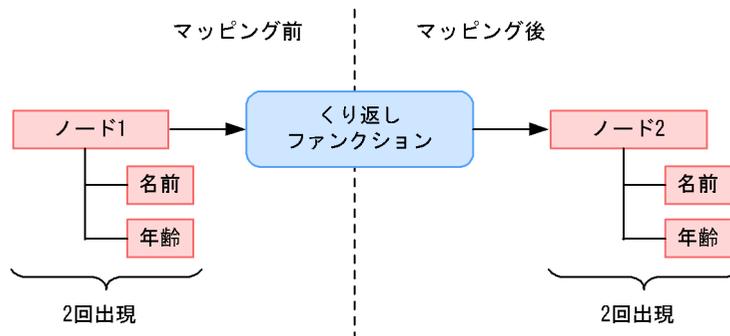
複数回出現する変換先ノードに対して、複数回出現する変換元ノードをマッピング元として、くり返しを対応づけます。また、複数回出現する変換元ノードのインスタンスをソートすることもできます。

ここでは、くり返しを対応づける基本的な方法を説明します。くり返しの合成、くり返し依存対象などの内容については、「7.6 くり返しの設定」を参照してください。

#### (1) 使用するファンクション

複数回出現する変換先ノードに対して、複数回出現する変換元ノードをマッピング元として、くり返しを対応づけるには、くり返しファンクションを使用します。くり返しファンクションを使用した例を次に示します。

図 7-14 くり返しファンクションの使用例



(凡例)

→ : マッピング元とマッピング先の接続

## (2) 設定手順

複数回出現する変換先ノードに対して、複数回出現する変換元ノードをマッピング元として、くり返しを対応づける手順を次に示します。

1. パレットからくり返しファンクション (loop) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [くり返し] ダイアログを表示させます。
  - くり返しファンクションを右クリックして、[設定] を選択します。
  - くり返しファンクションをダブルクリックします。

[くり返し] ダイアログが表示されます。

[くり返し] ダイアログの詳細については、「11.5.21 くり返しダイアログ」を参照してください。

4. 変換元ノードのインスタンスをソートする場合、[追加] ボタンをクリックします。  
[ソート条件追加/編集] ダイアログが表示されます。  
[ソート条件追加/編集] ダイアログの詳細については、「11.5.22 ソート条件追加/編集ダイアログ」を参照してください。
5. [ノード選択] ボタンをクリックします。  
[ノード選択] ダイアログが表示されます。[ノード選択] ダイアログで、ソートさせたいインスタンスの変換元ノードを指定します。
6. 次の項目を設定します。
  - [順序]  
[昇順] または [降順] を選択します。
  - [言語]  
[日本語]、[英語]、または [自動] を選択します。

## 7. データ変換の定義

- [データタイプ]  
[テキスト]または[数値]を選択します。
  - [優先順位]  
[大文字]または[小文字]を選択します。
7. [OK] ボタンをクリックします。  
[くり返し]ダイアログの[ソート条件]に,[ソート条件追加/編集]ダイアログで指定した内容が設定されます。
  8. [ソート条件]にキーを追加する場合,手順4~手順7.をくり返します。  
キーは八つまで設定できます。
  9. [ソート条件]の順番を変える場合,[上へ]ボタンまたは[下へ]ボタンをクリックします。  
[ソート条件]で一番上に設定した条件から順にソートされます。  
例えば,一番上に日付,二番目に名前を条件として設定した場合,まず全体を日付でソートします。次に,ソートした中に同じ日付があった場合,その日付の中でさらに名前ですべてソートします。
  - 10.[OK] ボタンをクリックします。

### ! 注意事項

- 同じ変換先ノードに対して,くり返しをネストして定義しないでください。つまり,同じ変換先ノードに対して,くり返しファンクションを直列に接続しないでください。
- 同じ変換先ノードに対して,同じくり返し依存対象の代入線を複数設定しないでください。くり返し依存対象については,「7.6.3 くり返し依存対象の対応づけ」を参照してください。
- 同じ変換先ノードに対して,同じくり返し依存対象のくり返し対応線および代入線を同時に設定しないでください。くり返し依存対象については,「7.6.3 くり返し依存対象の対応づけ」を参照してください。
- 変換先ノードにくり返しを定義した場合,変換先ノードの子孫ノードにはくり返し依存対象を設定してください。くり返し依存対象については,「7.6.3 くり返し依存対象の対応づけ」を参照してください。

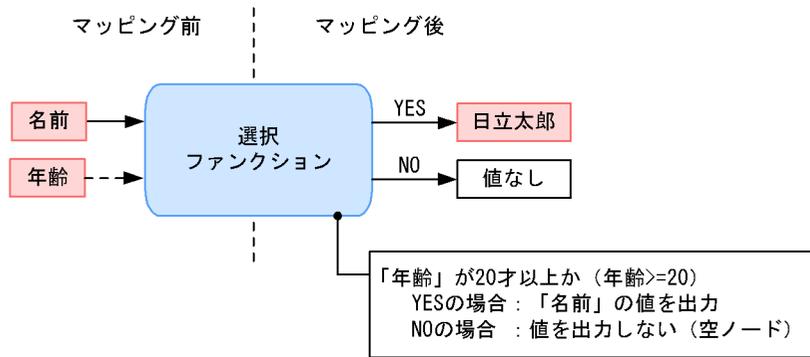
## 7.5.14 条件によって異なる値を出力する

ある条件を設定して,その条件に合致する場合または合致しない場合で,異なる値を出力します。

### (1) 使用するファンクション

ある条件を設定して,その条件に合致する場合または合致しない場合で,異なる値を出力させるには,選択ファンクションを使用します。選択ファンクションを使用した例を次に示します。

図 7-15 選択ファンクションの使用例



(凡例)

- : マッピング元とマッピング先の接続
- : 条件に設定されている変換元ノードまたはファンクションと選択ファンクションの接続

## (2) 設定手順

ある条件を設定して、その条件に合致する場合または合致しない場合で、異なる値を出力させる手順を次に示します。

1. パレットから選択ファンクション (choose) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。  
パレットのツールではなく、[ 選択 ] ダイアログでマッピング元を設定してマッピング線を設定する場合は、手順 .7 ~ 手順 .11 を実施します。
3. 次のどちらかの方法で [ 選択 ] ダイアログを表示させます。
  - 選択ファンクションを右クリックして、[ 設定 ] を選択します。
  - 選択ファンクションをダブルクリックします。

[ 選択 ] ダイアログが表示されます。  
手順 2. でマッピング線を設定した場合は、[ 条件と出力値 ] の [ 出力値 ] にマッピング元が設定されています。  
[ 選択 ] ダイアログの詳細については、「11.5.23 選択ダイアログ」を参照してください。
4. 次のどちらかの方法で [ 条件追加 / 編集 ] ダイアログを表示させます。
  - [ 条件と出力値 ] の [ 出力値 ] にマッピング元が設定されている場合  
[ 条件と出力値 ] の [ 出力値 ] に設定されているマッピング元を選択して、[ 編集 ] ボタンをクリックします。
  - [ 条件と出力値 ] の [ 出力値 ] にマッピング元が設定されていない場合  
[ 追加 ] ボタンをクリックします。

## 7. データ変換の定義

[条件追加/編集] ダイアログの詳細については、「11.5.24 条件追加/編集ダイアログ」を参照してください。

5. [条件設定] ボタンをクリックします。  
[条件設定] ダイアログが表示されます。

6. 条件を設定します。  
[条件設定] ダイアログでの条件設定の方法については、「11.5.7 条件設定ダイアログ (ノード条件の設定)」を参照してください。

7. [条件が true の場合] 欄で次のように指定します。
  - 変換先ノードを出力させる場合  
[ノードを出力する] チェックボックスにチェックします。
  - 変換先ノードを出力させない場合  
[ノードを出力しない] チェックボックスにチェックします。

なお、[条件と出力値] の [出力値] ですでにマッピング元が設定されている場合、[値] にマッピング元が表示されます。マッピング元を変更しないときは、手順 10 に進んでください。

8. 手順 7. で [ノードを出力する] チェックボックスにチェックした場合、[出力値] 欄で次のように指定します。
  - 変換先ノードに値を出力させる場合  
[値を出力する] チェックボックスにチェックします。
  - 変換先ノードに値を出力させない場合  
[空ノード] チェックボックスにチェックします。

9. 手順 8. で [値を出力する] チェックボックスにチェックした場合、[値] を次のように指定します。
  - 変換元ノードの値を出力させる場合  
[ノード選択] ボタンをクリックすると、[ノード選択] ダイアログが表示されます。[ノード選択] ダイアログで、値を出力させたい変換元ノードを選択します。
  - ファンクションの値を出力させる場合  
[ファンクション選択] ボタンをクリックすると、[ファンクション選択] ダイアログが表示されます。[ファンクション選択] ダイアログで、値を出力させたいファンクションを選択します。

10. [値] に変換元ノードが指定されている場合、その変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたいとき、[ノード条件設定] ボタンをクリックします。  
[ノード条件設定] ダイアログが表示されます。  
ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また、[ノード条件設定] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。

11. [OK] ボタンをクリックします。  
[条件追加/編集] ダイアログが閉じます。

12. 条件を追加または編集する場合、手順 4. ~ 手順 .11 をくり返します。

[ 条件と出力値 ] に複数の条件およびマッピング元を指定した場合、一番上に表示された条件から順に、条件を判定します。

13. どの条件にも合致しない場合のマッピング元を、[ 選択 ] ダイアログの [ どの条件にも合致しない場合 ] 欄で設定します。

設定方法は、[ 条件追加 / 編集 ] ダイアログの [ 条件が true の場合 ] 欄と同じです。

14. [ OK ] ボタンをクリックします。

[ 選択 ] ダイアログで設定した条件中に、マッピング元以外の変換元ノードまたはファンクションを指定した場合、その条件中の変換元ノードまたはファンクションと選択ファンクションの間にはマッピング線 ..... (条件線) が設定されます。

### ! 注意事項

- 複合内容の変換先ノードにマッピングする場合、[ 値を出力する ] を選択して [ 値 ] を指定しないでください。また、複合内容以外の変換先ノードにマッピングする場合、[ 値を出力する ] を選択したときは、必ず [ 値 ] を指定してください。
- すべての子要素の最小出現回数が 1 以上に設定されている choice 要素にマッピングする場合、[ 空ノード ] を指定しないでください。
- 最小出現回数が 1 以上に設定されている子要素を持つ要素 ( choice 要素以外 ) にマッピングする場合、[ 空ノード ] を指定しないでください。
- 最小出現回数が 1 以上に設定されているノードにマッピングする場合、[ ノードを出力しない ] を指定しないでください。
- 出力値にくり返しファンクションを一つでも設定した場合、ほかの出力値もすべてくり返しファンクションを設定してください。また、出力値にくり返しファンクション以外を設定した場合、ほかの出力値もすべてくり返しファンクション以外を設定してください。

## 7.5.15 指定した値を代入する

マッピング先に指定した値を代入します。指定できる値を次に示します。

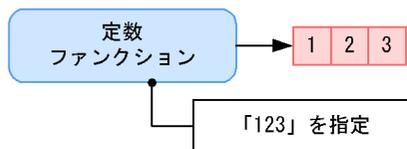
- 文字列
- 数値
- 論理値 ( 「真」または「偽」 )
- 特殊ノード ( 「ノード出力なし」または「空ノード」 )

### (1) 使用するファンクション

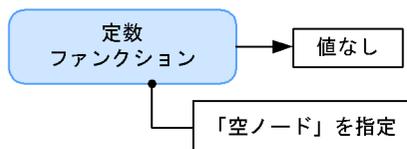
マッピング先に、指定した値を代入するには、定数ファンクションを使用します。定数ファンクションに指定した値は、複数のマッピング先に代入できます。定数ファンクションにはマッピング元がありません。したがって、定数ファンクションの値がそのままマッピング先に代入されます。定数ファンクションの使用例を次の図に示します。

図 7-16 定数ファンクションの使用例

●数値を指定した場合



●特殊ノードを指定した場合



(凡例)

→ : 選択ファンクションとマッピング先の接続

## (2) 設定手順

マッピング先に、指定した値を代入する手順を次に示します。

1. パレットから定数ファンクション (const) を選択して、マッピングビューアに配置します。

2. マッピング線を設定します。

3. 次のどちらかの方法で [定数] ダイアログを表示させます。

- 定数ファンクションを右クリックして、[設定] を選択します。
- 定数ファンクションをダブルクリックします。

[定数] ダイアログが表示されます。

[定数] ダイアログの詳細については、「11.5.25 定数ダイアログ」を参照してください。

4. マッピング先に代入する値を指定します。

- 文字列を指定する場合

[文字列] ラジオボタンを選択し、[値] に代入する値を指定します。

- 数値を指定する場合

[数値] ラジオボタンを選択し、[値] に代入する数値を指定します。

- 論理値を指定する場合

[論理値] ラジオボタンを選択し、[論理値] 欄で [真] または [偽] を選択します。

- 特殊ノードを指定する場合

[特殊ノード] ラジオボタンを選択し、[値] 欄で [ノード出力なし] または [空ノード] を選択します。

5. [ OK ] ボタンをクリックします。

### ! 注意事項

- 特殊ノードを指定する場合、マッピング先には変換先ノードを指定してください。
- 特殊ノード以外を指定する場合、マッピング先には複合型ノード以外のノードを指定してください。
- 特殊ノードでノード出力なしを指定した場合、マッピング先には最小出現回数が 0 回の変換先ノードを指定してください。
- すべての子要素の最小出現回数が 1 回以上に設定されている choice 要素にマッピングする場合、特殊ノードの空ノード以外を指定してください。
- 最小出現回数が 1 回以上に設定されている子要素を持つ要素（choice 要素以外）にマッピングする場合、特殊ノードの空ノード以外を指定してください。

## 7.5.16 変換テーブルを利用して値を変換する

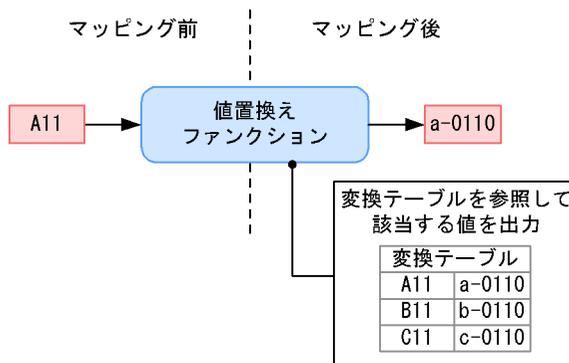
変換テーブルを利用して、マッピング元の値を変換してマッピング先に代入します。

### (1) 使用するファンクション

変換テーブルを利用して、マッピング元の値を変換してマッピング先に代入するには、値置換えファンクションを使用します。

値置換えファンクションを使用した例を次に示します。

図 7-17 値置換えファンクションの使用例



(凡例)

→ : マッピング元とマッピング先の接続

### (2) 設定手順

変換テーブルを利用して、マッピング元の値を変換してマッピング先に代入するには、まず、変換テーブルを作成・保存します。作成した変換テーブルの情報をシステムプロパティファイル（usrconf.properties）に記述すると、変換テーブルが利用できる状態になります。変換テーブルが利用できる状態になったあと、値置換えファンクションを定

## 7. データ変換の定義

義します。

### (a) 変換テーブルの作成と保存

変換テーブルの作成と保存について説明します。

#### ポイント

作成した変換テーブルはパッケージングの対象にはなりません。そのため、実行環境での実運用開始後にも、任意のタイミングで編集できます。編集された内容は、次のシステム起動時に反映されます。

#### 変換テーブルの作成

変換テーブルは、csv ファイルで作成します。csv ファイルの形式を次に示します。

```
"値A01", "値B01"  
"値A02", "値B02"  
"値A03", "値B03"  
:
```

同じ行に記述した「値 Ann」と「値 Bnn」を一对の組み合わせとして値を置き換えます。置き換え対象の文字列はダブルクォーテーション(")で囲んで記述します。置き換え対象の文字列にダブルクォーテーションが含まれる場合、ダブルクォーテーションをエスケープ文字として使用できます。エスケープ対象外の文字をエスケープした場合、エスケープ文字は無視されて、以降の文字が処理されます。なお、変換テーブルに使用できる文字コードは次のとおりです。

- MS932
- UTF8
- UTF16 (ビッグエンディアン, またはリトルエンディアン)

変換テーブルの記述に使用する文字について次の表に示します。

表 7-2 変換テーブルに使用する文字

対象	使用する文字
開始文字	開始文字は不要です。
終了文字	次のどちらかの改行コードです。 <ul style="list-style-type: none"><li>• LF ( 0x0A )</li><li>• CR + LF ( 0x0D0A )</li></ul> 実行する OS が UNIX の場合は LF, Windows の場合は CR+LF を使用します。 注 LF: ラインフィード CR: キャリッジリターン
中間区切り文字	コンマ(,)を利用します。

**!** 注意事項

- 変換テーブルに記述した変換対象は文字列として扱います。そのため、値置換えファンクションへの入力値が数値の場合、文字列に変換して処理されます。
- 変換対象および入力値に、空要素は指定できません。
- 属性値を入力する場合、入力値は正規化（タブをスペースに置き換えたり、連続したスペースを一つのスペースに置き換えたりするなど）した値で扱います。そのため、変換テーブルに記述した検索キー値も、正規化後の値を考慮する必要があります。

## ポイント

「値 Ann」から「値 Bnn」の変換にするか、「値 Bnn」から「値 Ann」の変換にするかは、値置換えファンクションを定義するときに設定できます。

## 変換テーブルの保存

作成した変換テーブルは、任意の場所に保存します。

## (b) システムプロパティファイル (usrconf.properties) への登録

変換テーブルを作成したあとは、usrconf.properties に作成した変換テーブルを登録します。変換テーブルの登録方法の詳細については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の変換テーブルの登録に関する内容を参照してください。

## (c) 値置換えファンクションの定義

値置換えファンクションを利用して、マッピング元の値を変換してマッピング先に代入するための手順を次に示します。

1. パレットから値置換えファンクション (replace) を選択して、マッピングビューアに配置します。
2. マッピング線を設定します。
3. 次のどちらかの方法で [ 値置換え ] ダイアログを表示させます。
  - 値置換えファンクションを右クリックして、[ 設定 ] を選択します。
  - 値置換えファンクションをダブルクリックします。

[ 値置換え ] ダイアログが表示されます。

[ 値置換え ] ダイアログの詳細については、「11.5.26 値置換えダイアログ」を参照してください。

4. [ 入力 ] に変換元ノードが指定されている場合、その変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせたいとき、[ ノード条件設定 ] ボタンをクリックします。

[ ノード条件設定 ] ダイアログが表示されます。

ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。また、[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイ

## 7. データ変換の定義

「ログ」を参照してください。

5. [ 変換テーブル ID ] にシステムプロパティファイル (usrconf.properties) に記述した変換テーブルの名称を指定します。
6. [ 検索キー列指定 ] で、変換テーブルの左列と右列のどちらを値の置き換え元とするか選択します。
7. [ 検索失敗時の動作 ] で、マッピング元からの入力値が変換テーブルに存在しなかった場合のシステムの動作を選択します。

デフォルト値を出力してマッピング先に代入する場合

[ デフォルト値を代入 ] をラジオボタンで選択し、[ 値 ] にデフォルト値として利用する文字列を入力します。

変換エラーにする場合

[ 変換エラー ] をラジオボタンで選択します。

8. [ OK ] ボタンをクリックします。

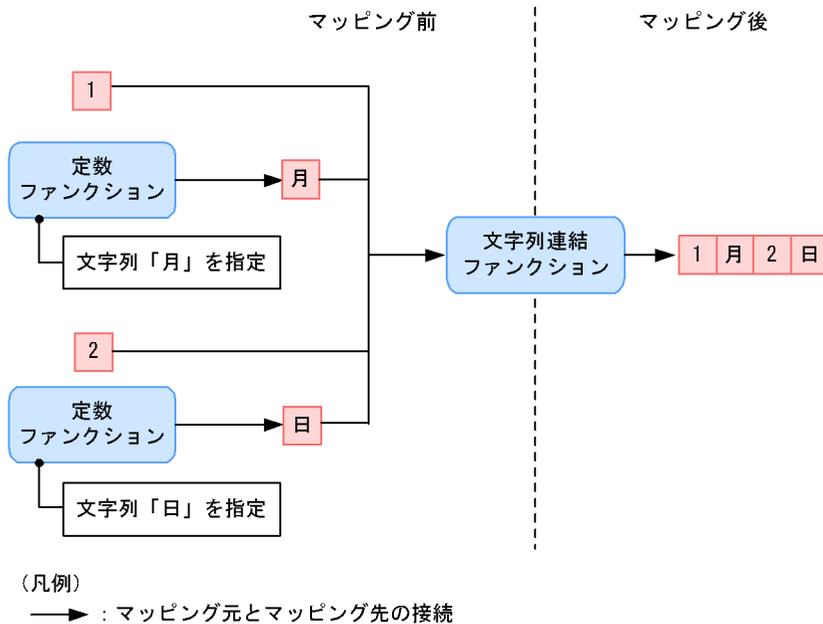
### 7.5.17 変換元ノードの値にある値を付加する

変換元ノードの値にある値を付加して、マッピング先に代入します。

#### (1) 使用するファンクション

変換元ノードの値にある値を付加して、マッピング先に代入するには、文字列連結ファンクションおよび定数ファンクションを組み合わせで使用します。文字列連結ファンクションおよび定数ファンクションを組み合わせで使用した例を次に示します。

図 7-18 文字列連結ファンクションおよび定数ファンクションを組み合わせた使用例



## (2) 設定手順

変換元ノードの値にある値を付加して、マッピング先に代入する手順を次に示します。

1. パレットから定数ファンクション (const) を選択して、マッピングビューアに配置します。
2. 定数ファンクションで値を指定します。  
値を指定する方法については、「7.5.15 指定した値を代入する」を参照してください。
3. パレットから文字列連結ファンクション (concat) を選択して、マッピングビューアに配置します。
4. 定数ファンクションで指定した値と変換元ノードの値を、文字列連結ファンクションで連結します。  
連結方法については、「7.5.1 複数の文字列を連結する」を参照してください。

## 7.5.18 変換元ノードの値を倍にする

変換元ノードの値を倍にして、マッピング先に代入します。

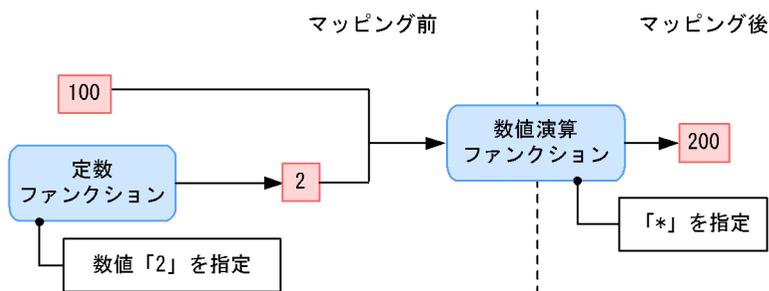
### (1) 使用するファンクション

変換元ノードの値を倍にして、マッピング先に代入するには、数値演算ファンクションおよび定数ファンクションを組み合わせて使用します。数値演算ファンクションおよび

## 7. データ変換の定義

定数ファンクションを組み合わせて使用した例を次に示します。

図 7-19 演算ファンクションおよび定数ファンクションを組み合わせた使用例



(凡例)

→ : マッピング元とマッピング先の接続

### (2) 設定手順

変換元ノードの値を倍にして、マッピング先に代入する手順を次に示します。

1. パレットから定数ファンクション (const) を選択して、マッピングビューアに配置します。
2. 定数ファンクションで値を指定します。  
値を指定する方法については、「7.5.15 指定した値を代入する」を参照してください。
3. パレットから数値演算ファンクション (calc) を選択して、マッピングビューアに配置します。
4. 定数ファンクションで指定した値と変換元ノードの値を、数値演算ファンクションで演算します。  
演算方法については、「7.5.7 数値を演算する」を参照してください。

## 7.6 くり返しの設定

---

くり返しファンクションを使用したマッピングでは、くり返しを合成したり、くり返し依存対象を変更したりできます。

### 7.6.1 [くり返し設定] ダイアログを使用してマッピングする

くり返しファンクションまたは選択ファンクション から変換先ノードへのマッピングは、[くり返し設定] ダイアログを使用することもできます。

[くり返し設定] ダイアログを使用したマッピングの手順を次に示します。

1. パレットからくり返しファンクションまたは選択ファンクションを選択します。
2. マッピングビューアで適当な場所をクリックします。  
ファンクションがマッピングビューアに配置されます。配置されたファンクションは、ドラッグ&ドロップで移動できます。
3. 変換先スキーマツリービューア上のマッピング先となる変換先ノードを右クリックして、[くり返し]を選択します。  
[くり返し設定] ダイアログが表示されます。
4. [ファンクション追加] ボタンをクリックします。  
[ファンクション選択] ダイアログが表示されます。
5. マッピング元となるくり返しファンクションまたは選択ファンクションを指定して、[OK] ボタンをクリックします。  
[ファンクション名] に指定したファンクション名が設定されます。
6. [OK] ボタンをクリックします。  
マッピング線 .....(くり返し対応線) が設定されます。なお、マッピング線の色は変更できます。マッピング線の色を変更する方法については、「11.9.4 マッピング線の色を変更する」を参照してください。

#### 注

くり返しファンクションに接続されている、または何も接続されていない選択ファンクションになります。くり返しファンクション以外のファンクションに接続されている場合、[くり返し設定] ダイアログは使用できません。

### ! 注意事項

変換元ノードにくり返しを設定した場合、その変換元ノードの子孫のノードもくり返しの対象となり、変換先ノードに出現します。

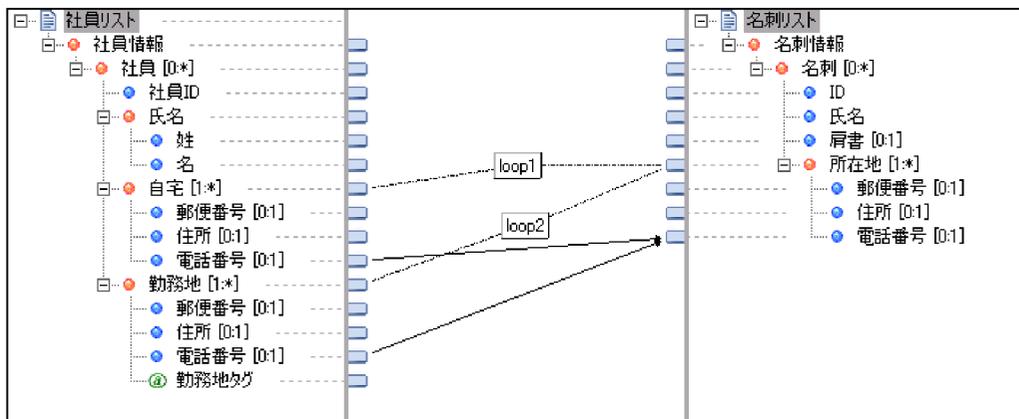
ただし、[マッピング元設定]ダイアログでマッピング元に絶対パスを指定した場合は、常に絶対パスで指定したノードの値が代入されます。[マッピング元設定]ダイアログでマッピング元に相対パスが表示されている場合には、くり返しを設定した変換元ノードの子孫のノードがくり返し対象となり、くり返しに対応したノードの値が代入されます。

変換元ノードのパスの表示については、「7.6.6 くり返し依存対象が設定された変換元ノードのパスの表示」を参照してください。

## 7.6.2 くり返しを合成する

複数の変換元ノードのくり返しを合成して、一つの変換先ノードに対応づけることができます。くり返しを合成した場合の画面を次に示します。

図 7-20 くり返しを合成した場合の画面



くり返しを合成した場合、変換先ノードに出現させる変換元ノードの順序を[くり返し設定]ダイアログで設定できます。図 7-20 の場合、loop1 に対応した変換元ノード（自宅）と loop2 に対応した変換元ノード（勤務地）のどちらを先に変換先ノードに出現させるか決めることができます。なお、くり返しに対応した変換元ノードの子孫のノードもくり返しの対象となり、変換先ノードに出現します。

変換先ノードに代入する変換元ノードの順序を指定する手順を次に示します。

1. 変換先スキーマツリービュー上のマッピング先となる変換先ノードを右クリックして、[くり返し]を選択します。  
[くり返し設定]ダイアログが表示されます。
2. [くり返し]に表示されたファンクション名を、変換先ノードに出現させる順番に並べ替えます。  
一番上に表示されたファンクション名に対応する変換元ノードから順に、変換先ノード

ドに出現します。

- 順番を上にした場合  
上にしたファンクション名を選択して [ 上へ ] ボタンをクリックします。
- 順番を下にした場合  
下にしたファンクション名を選択して [ 下へ ] ボタンをクリックします。

3. [ OK ] ボタンをクリックします。

### 7.6.3 くり返し依存対象の対応づけ

くり返しファンクションを使用して変換元ノードから変換先ノードへマッピングした場合、そのくり返しファンクションは、自動的に子孫ノードのマッピング元のくり返し対象として対応づけられます。また、くり返しファンクションやマッピング線を削除して、あるノードのくり返しを取り消したとします。その場合、その子孫ノードには、くり返しを取り消したノードの祖先ノードに設定されているくり返しファンクションが、自動的にマッピング元のくり返し対象として対応づけられます。これをくり返しの依存対象と言います。

くり返しの依存対象は次の場合に設定されます。

- あるノードにくり返しファンクションを使用してくり返しを設定したあと、その子孫ノードをマッピングした場合  
あるノードで使用したくり返しファンクションは、自動的にその子孫ノードのマッピング元のくり返し依存対象となります。
- あるノードをマッピングしたあと、その祖先ノードにくり返しファンクションを使用してくり返しを設定した場合  
祖先ノードで使用したくり返しファンクションは、その子孫ノードのマッピング元のくり返し依存対象となります。
- 祖先ノードにくり返しが設定されているノードのくり返しを取り消した場合  
祖先ノードに設定されているくり返しファンクションは、くり返しを取り消したノードの子孫ノードで、マッピング元のくり返し依存対象となります。

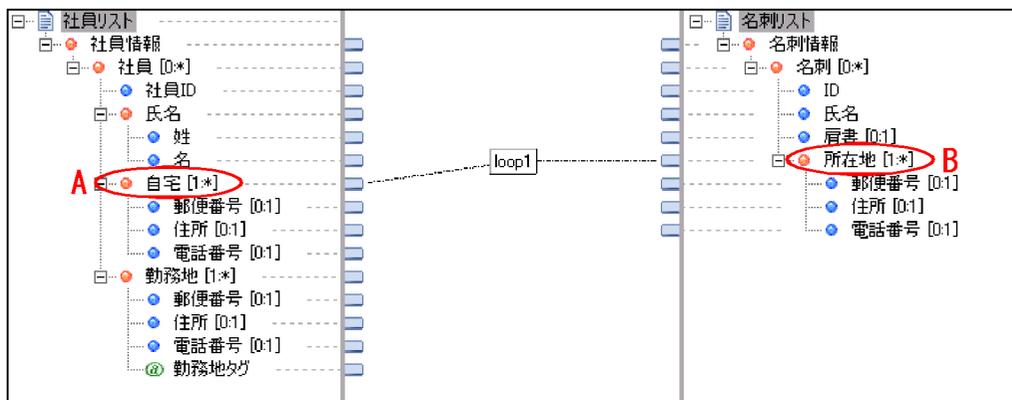
それぞれの場合のくり返し依存対象が対応づけられるタイミングについて説明します。

#### (1) あるノードにくり返しを設定したあと、その子孫ノードをマッピングした場合

あるノードにくり返しを設定したあと、その子孫ノードをマッピングした場合の、くり返し依存対象が対応づけられるタイミングを次に示します。

## 7. データ変換の定義

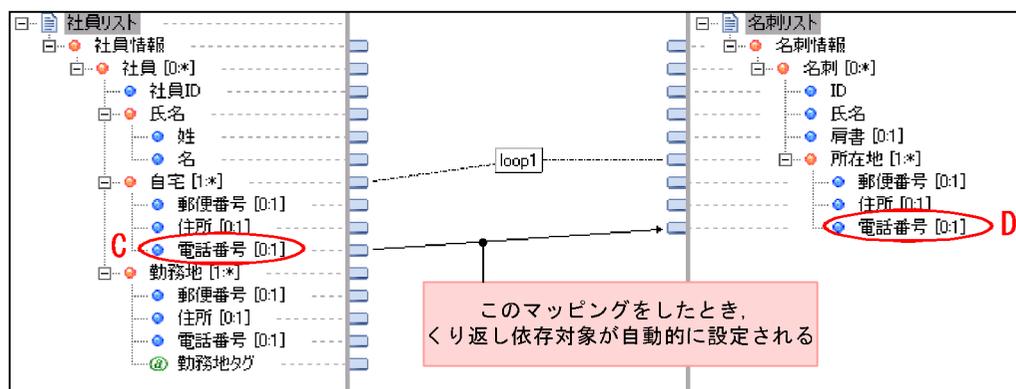
図 7-21 くり返しファンクションを使用したマッピング



まず、くり返しファンクションを使用して、変換元ノード A から変換先ノード B へマッピングします。

次に、変換元ノード A の子孫ノードであるノード C から、変換先ノード B の子孫ノードであるノード D へマッピングします。

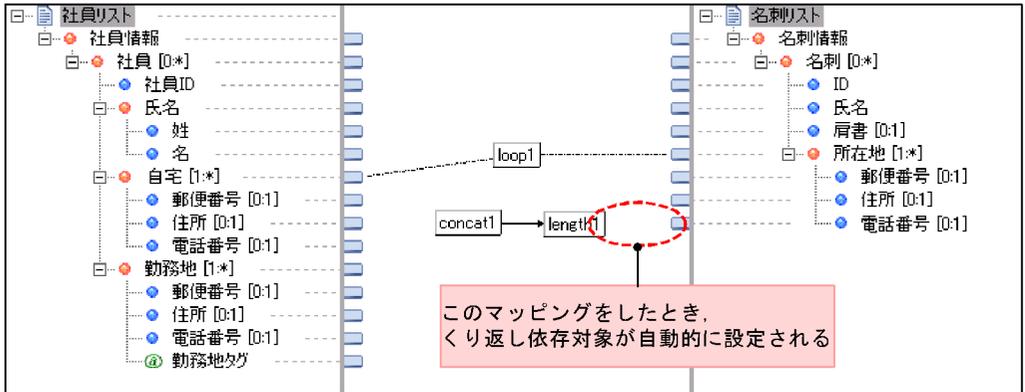
図 7-22 くり返し依存対象が対応づけられるタイミング (子孫ノードにファンクションがない場合)



ノード D に接続されたとき、子孫ノードに対してくり返し依存対象が自動的に設定されます。図 7-22 の場合、子孫ノード D のマッピング元のくり返し依存対象に「loop1」が設定されます。

子孫の変換元ノードから変換先ノードへのマッピングで複数のファンクションを使用した場合は、変換先ノードに接続された時点でくり返し依存対象が自動的に設定されます。

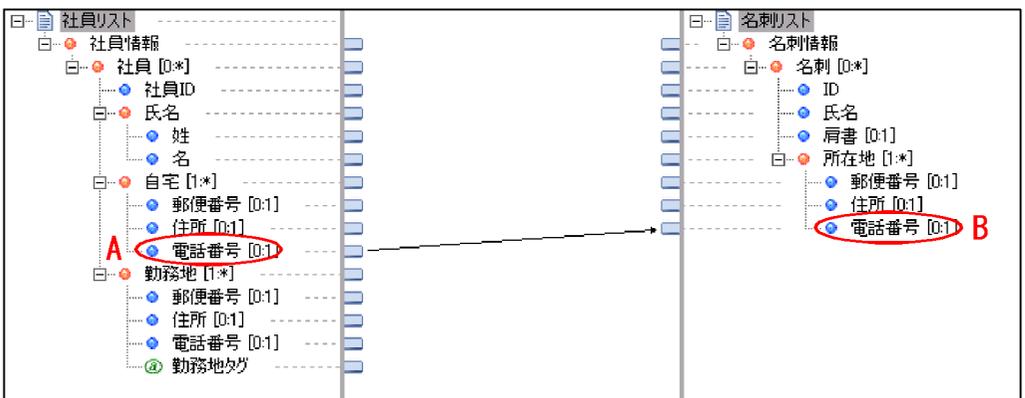
図 7-23 くり返し依存対象が対応づけられるタイミング（子孫のノードに複数のファンクションがある場合）



(2) あるノードをマッピングさせたあと、その祖先ノードにくり返しを設定した場合

あるノードをマッピングさせたあと、その祖先ノードにくり返しを設定した場合の、くり返し依存対象が対応づけられるタイミングを次に示します。

図 7-24 くり返しファンクションを使用しないマッピング

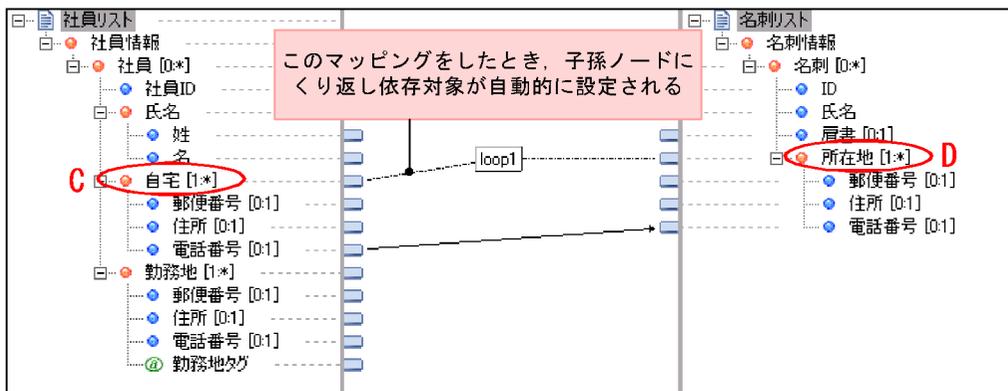


まず、ある変換元ノード A から変換先ノード B へマッピングします。

次に、変換元ノード A の祖先ノードであるノード C から、変換先ノード B の祖先ノードであるノード D へマッピングします。

## 7. データ変換の定義

図 7-25 くり返し依存対象が対応づけられるタイミング（祖先ノードのマッピングをあとにした場合）

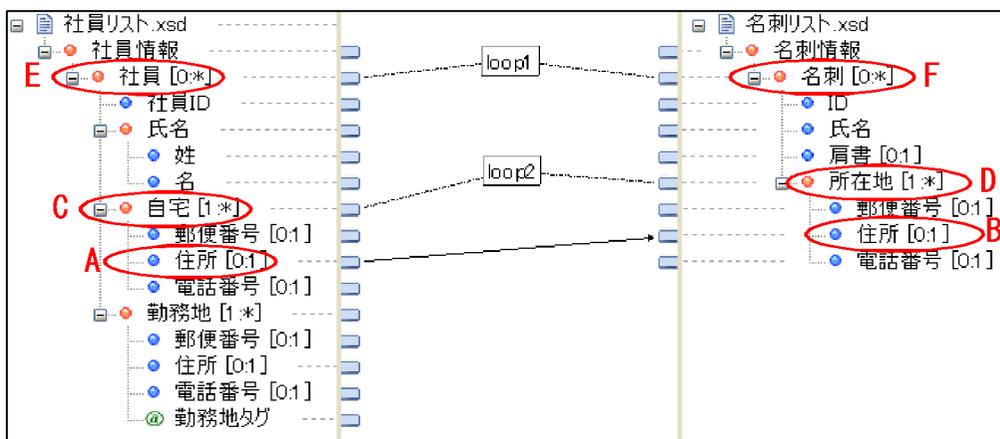


ノード C からノード D にマッピングしたとき、子孫ノードに対してくり返し依存対象が自動的に設定されます。図 7-25 の場合、子孫ノード B に対してくり返し依存対象「loop1」が設定されます。

### (3) 祖先ノードにくり返しが設定されているノードのくり返しを取り消した場合

祖先ノードにくり返しが設定されているノードのくり返しを取り消した場合の、くり返し依存対象が対応づけられるタイミングを次に示します。

図 7-26 くり返しファンクションを使用したマッピング、およびくり返しファンクションを使用しないマッピング

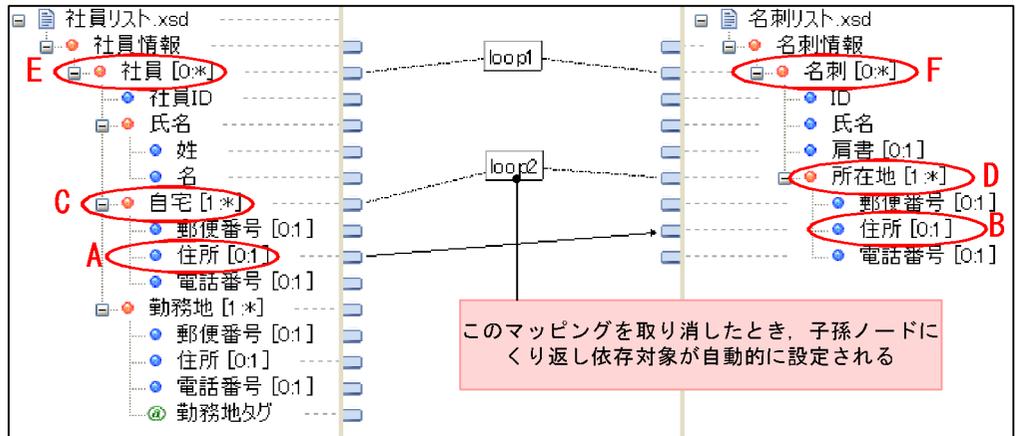


まず、ある変換元ノード A から変換先ノード B へマッピングします。

次に、くり返しファンクションを使用して、変換元ノード A の祖先ノードであるノード C から、変換先ノード B の祖先ノードであるノード D へマッピングします。

そのあと、くり返しファンクションを使用して、変換元ノード A および変換元ノード C の祖先ノードである変換元ノード E から、変換先ノード B および変換先ノード D の祖先ノードである変換先ノード F へマッピングします。

図 7-27 くり返し依存対象が対応づけられるタイミング (マッピングを取り消した場合)

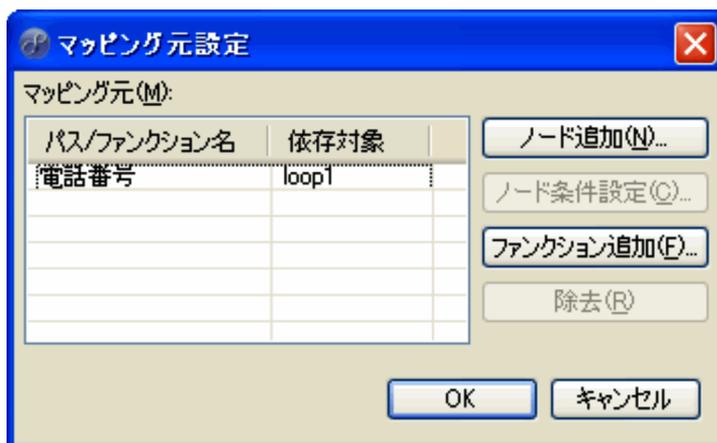


ノード C からノード D へのマッピングを取り消したとき、マッピングを取り消したノードの子孫ノードに対して、くり返し依存対象が自動的に設定されます。図 7-27 の場合、子孫ノード B に対してくり返し依存対象「loop1」が設定されます。

#### 7.6.4 くり返し依存対象を確認する

自動的に設定されたくり返し依存対象を確認する手順を次に示します。

- くり返しを設定した変換先ノードの子孫ノードをスキーマツリービューア上で右クリックして、[ マッピング元 ] を選択します。  
[ マッピング元設定 ] ダイアログが表示されます。



## 7. データ変換の定義

[パス/ファンクション名]には、くり返しファンクションが対応づけられた変換元ノードからの相対パスが表示されます。パス名の表示については、「7.6.6 くり返し依存対象が設定された変換元ノードのパスの表示」を参照してください。[依存対象]にはくり返し依存対象となるファンクション名が表示されます。

### 7.6.5 くり返し依存対象を変更する

複数の変換元ノードから一つの変換先ノードにマッピングしている場合、変換元ノードに対して自動的に設定されたくり返し依存対象を変更することができます。

くり返し依存対象を変更する手順を次に示します。

1. くり返しを設定した変換先ノードの子孫ノードをスキーマツリービューア上で右クリックして、[マッピング元]を選択します。  
[マッピング元設定]ダイアログが表示されます。
2. [マッピング元]の[依存対象]を変更します。  
変更した依存対象のパス名が絶対パスになります。パス名の表示については、「7.6.6 くり返し依存対象が設定された変換元ノードのパスの表示」を参照してください。
3. [OK]ボタンをクリックします。

### 7.6.6 くり返し依存対象が設定された変換元ノードのパスの表示

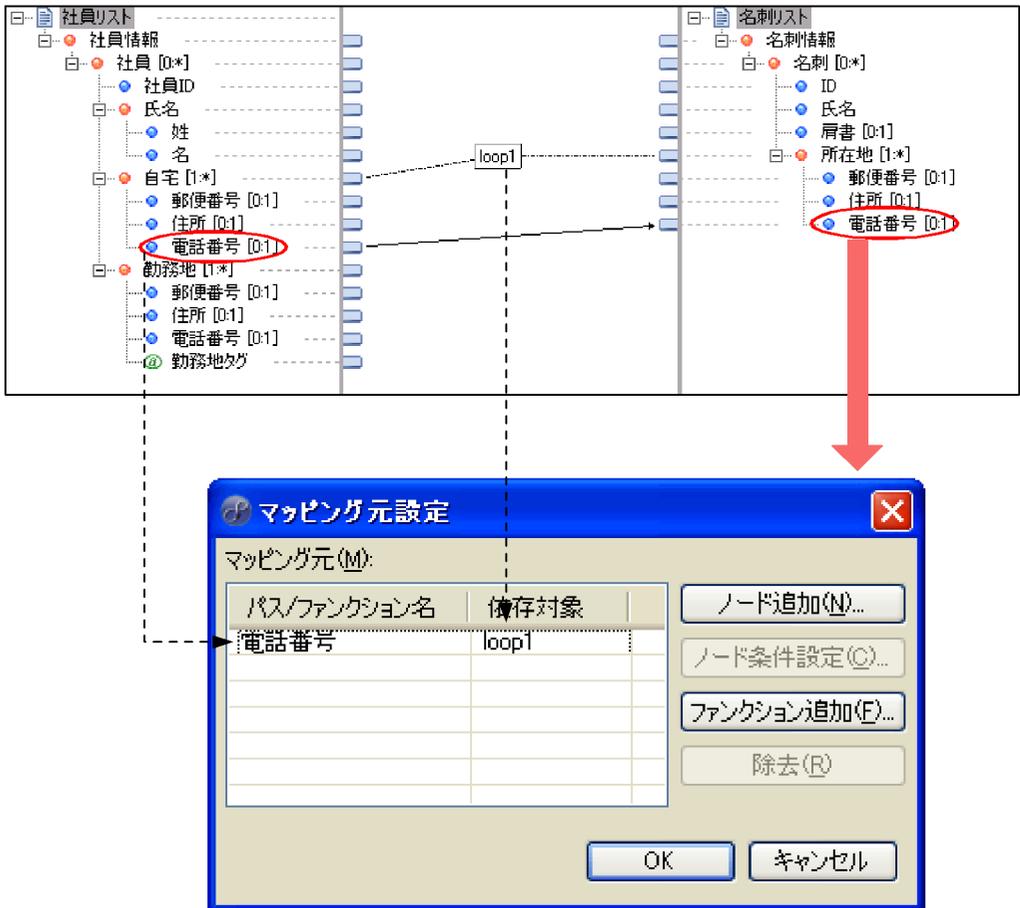
[マッピング元設定]ダイアログの[パス/ファンクション名]には、相対パスまたは絶対パスが表示されます。どのような場合に相対パスまたは絶対パスが表示されるかについて説明します。

#### (1) 相対パス

くり返しファンクションが対応づけられた変換元ノードの子孫の変換元ノードに対してくり返し依存対象が設定された場合、くり返しファンクションが対応づけられた変換元ノードからの相対パスが表示されます。

具体的な例を次に示します。

図 7-28 相対パスの例



自宅ノードにくり返しファンクション「loop1」を対応づけると、自宅ノードの子孫の電話番号ノードにくり返し依存対象「loop1」が設定されます。

このとき、マッピング後の電話番号ノードから [ マッピング元設定 ] ダイアログを表示させると、[ パス/ファンクション名 ] には、自宅ノードからの相対パスが表示されます。

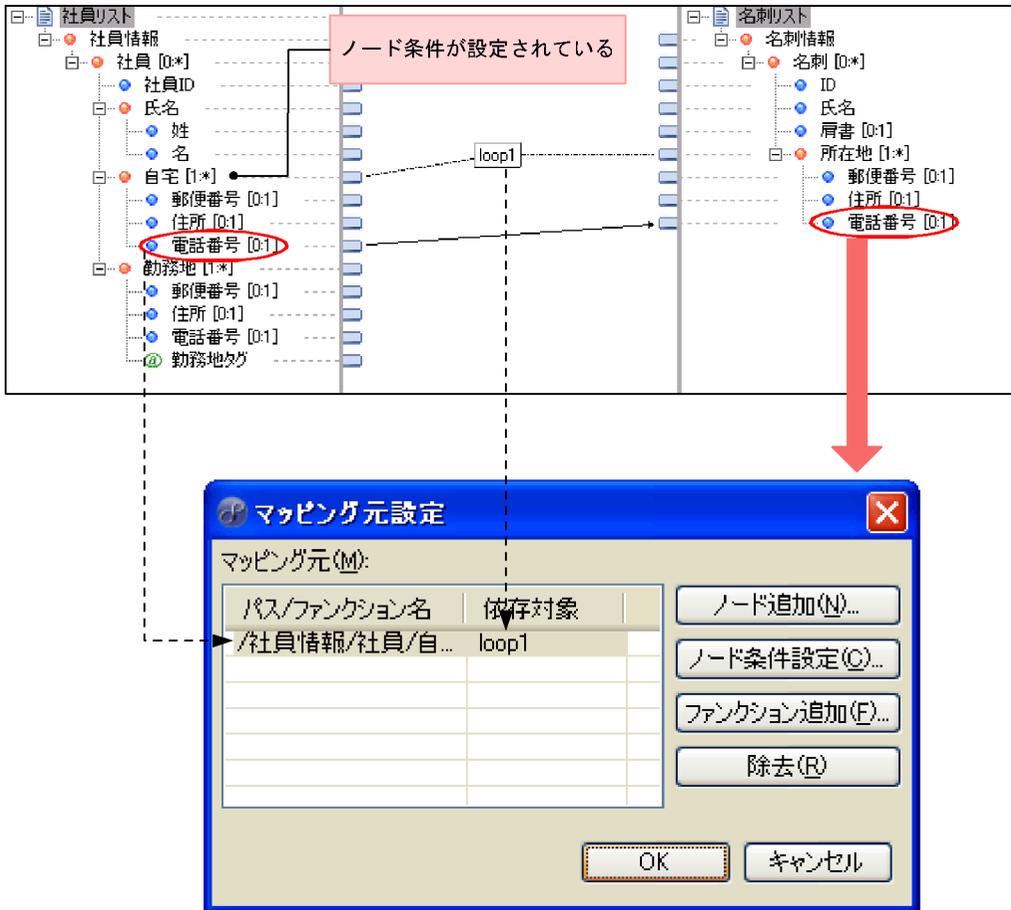
## (2) 絶対パス

次の場合、絶対パスが表示されます。

- ・くり返し依存対象のくり返しファンクションが対応づけられた変換元ノードまたはその変換元ノードの祖先の変換元ノードに、ノード条件が設定されている場合  
具体的な例を次に示します。

## 7. データ変換の定義

図 7-29 絶対パスの例 1

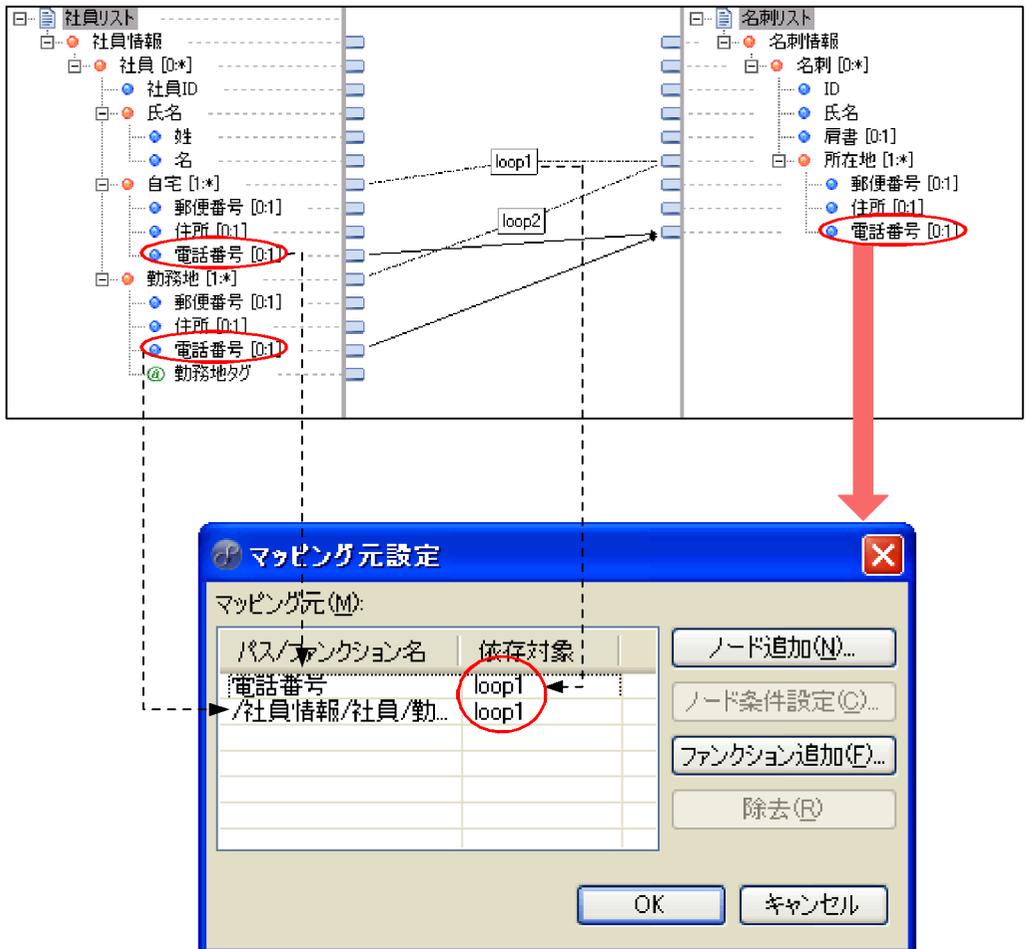


自宅ノードに繰り返しファンクション「loop1」を対応づけると、自宅ノードの子孫の電話番号ノードに繰り返し依存対象「loop1」が設定されます。なお、自宅ノードにはノード条件が設定されています。

このとき、マッピング後の電話番号ノードから [ マッピング元設定 ] ダイアログを表示させると、[ パス/ファンクション名 ] には、絶対パスが表示されます。

- 繰り返し依存対象が設定された変換元ノードが、繰り返し依存対象の繰り返しファンクションが対応づけられた変換元ノードの子孫のノードでない場合  
具体的な例を次に示します。

図 7-30 絶対パスの例 2 (依存対象の変更前)

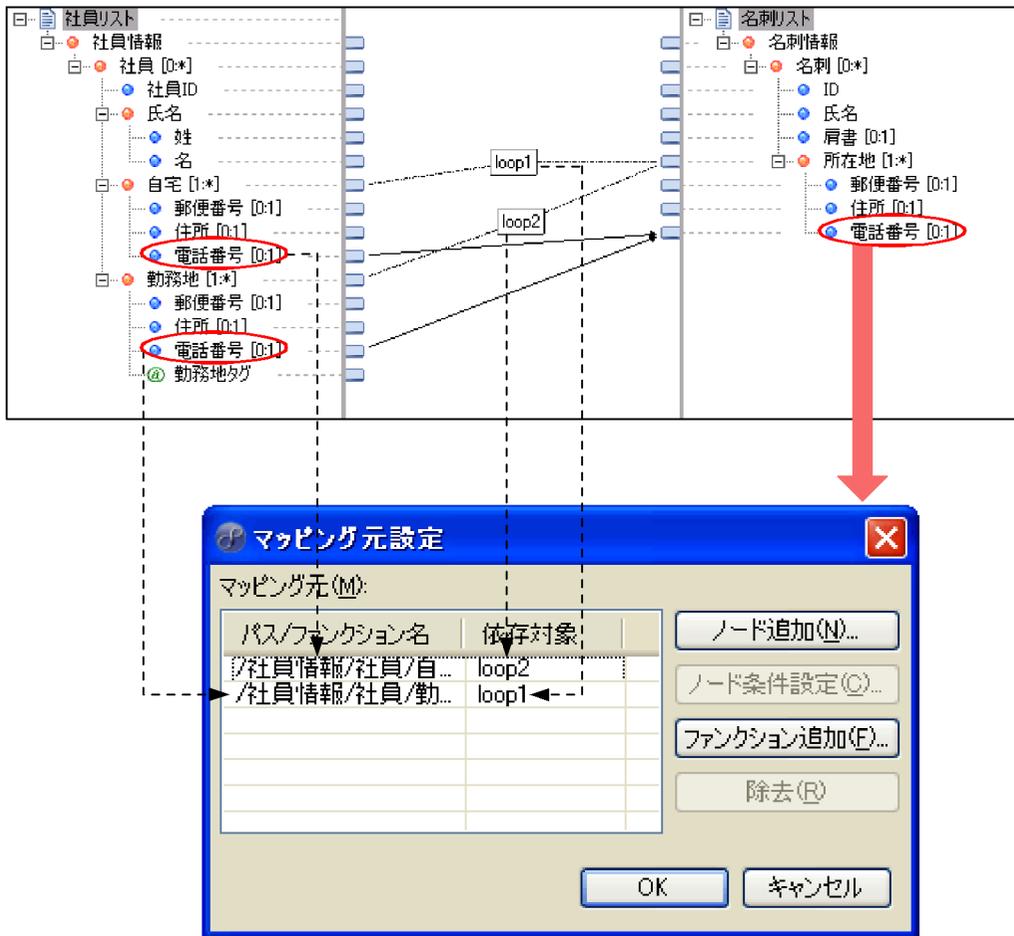


まず自宅ノードにくり返しファンクション「loop1」を対応づけ、次に勤務地ノードにくり返しファンクション「loop2」を対応づけます。これによって、自宅ノードの子孫の電話番号ノードおよび勤務地ノードの子孫の電話番号ノードには、最初に対応づけたくり返しファンクション「loop1」がくり返し依存対象に設定されます。

このとき、マッピング後の電話番号ノードから [マッピング元設定] ダイアログを表示させると、[パス/ファンクション名] には、次のように表示されます。

- 自宅ノード（「loop1」が対応づけられている）の子孫の電話番号ノード  
自宅ノードからの相対パスが表示されます。
- 勤務地ノード（「loop2」が対応づけられている）の子孫の電話番号ノード  
絶対パスが表示されます。

図 7-31 絶対パスの例 2 (依存対象の変更後)



自宅ノードの子孫の電話番号ノードの依存対象を、勤務地ノードのくり返しファンクションである「loop2」に設定します。

このとき、マッピング後の電話番号ノードから [ マッピング元設定 ] ダイアログを表示させると、[ パス/ファンクション名 ] の表示が、次のように変更されます。

- 自宅ノード（「loop1」が対応づけられている）の子孫の電話番号ノード  
自宅ノードからの相対パスが絶対パスになります。

## 7.7 ノード条件を設定する

マッピング元の変換元ノードに対して条件を設定して、その条件を満たす場合だけマッピングします。変換元ノードに条件を設定することをノード条件の設定といいます。ノード条件の設定は、マッピング先によって設定を開始する方法が次のように異なります。

- マッピング先が変換先ノードの場合  
[ マッピング元設定 ] ダイアログからノード条件の設定を開始します。[ マッピング元設定 ] ダイアログは、変換先ノードを右クリックして、[ マッピング元 ] を選択すると表示されます。
- マッピング先がファンクションの場合  
各ファンクションを設定するダイアログからノード条件の設定を開始します。各ファンクションを設定するダイアログは、次のどちらかの方法で表示されます。
  - 各ファンクションを右クリックして、[ 設定 ] を選択します。
  - 各ファンクションをダブルクリックします。

ノード条件を設定する手順を次に示します。

1. [ マッピング元設定 ] ダイアログまたは各ファンクションを設定するダイアログで、ノード条件を設定する変換元ノードを選択します。
2. [ ノード条件設定 ] ボタンをクリックします。  
[ ノード条件設定 ] ダイアログが表示されます。  
[ ノード条件設定 ] ダイアログには、[ マッピング元設定 ] ダイアログで選択した変換元ノードとその祖先のノードが表示されます。ノード条件は、[ ノード条件設定 ] ダイアログで表示されたすべてのノードに設定できます。
3. ノード条件を設定するノードを選択して、[ 条件設定 ] ボタンをクリックします。  
[ 条件設定 ] ダイアログが表示されます。
4. 条件を設定します。  
[ 条件 ] で設定する条件を次のラジオボタンから選択し、条件の詳細を設定します。
  - [ boolean 値を返すノードまたはファンクション ] ラジオボタン
  - [ 条件式 ] ラジオボタン
  - [ 二つの条件の論理積・論理和 ] ラジオボタン
 設定した条件は、[ 生成する条件 ] に表示されます。  
 なお、[ 条件 ] で変換元ノードまたはファンクションを指定する場合、次の変換元ノードまたはファンクションを指定してください。
  - 変換元ノード  
マッピング対象 であり、かつ単純内容または属性のノード
  - ファンクション  
出力側にマッピング線が設定されていない制御系以外のファンクション

## 7. データ変換の定義

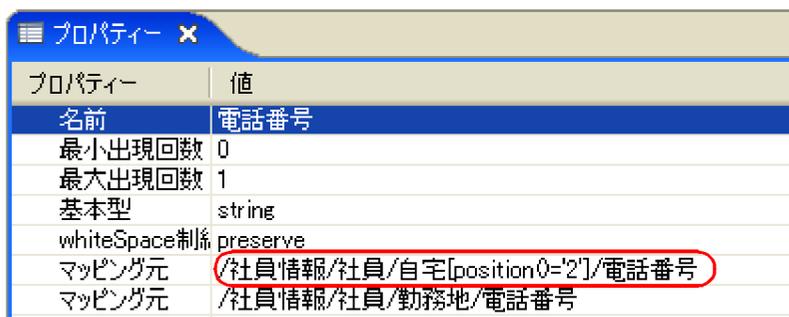
5. [生成する条件]に表示された条件と反対の内容をノード条件としたい場合,[条件の否定をとる]チェックボックスにチェックします。
6. [OK] ボタンをクリックします。  
[ノード条件設定]ダイアログの[条件]に,[条件設定]ダイアログで設定した条件が表示されます。
7. ほかのノードのノード条件を設定する場合,手順3.~手順6を繰り返します。
8. [ノード条件設定]ダイアログで,[OK] ボタンをクリックします。  
ノード条件が設定されます。  
[条件設定]ダイアログで設定した条件中に,マッピング元以外の変換元ノードまたはファンクションを指定した場合,その条件中の変換元ノードまたはファンクションとマッピング先の間にもマッピング線 .....(条件線)が設定されます。

### 注

マッピング対象については、「7.8.1 マッピング対象とマッピング対象外」を参照してください。

### ポイント

条件を設定した変換先ノードを選択すると,プロパティ領域の「マッピング元」に,設定したノード条件が表示されます。ノード条件は,条件を設定したノード名の後ろに角括弧([ ])で囲まれて表示されます。なお,条件として指定したノード([条件設定]ダイアログの[条件]で指定)が,条件を設定したノード([ノード条件設定]ダイアログで指定)の子孫ノードの場合,条件を設定したノードからの相対パスが表示されます。それ以外は絶対パスが表示されます。



プロパティ	値
名前	電話番号
最小出現回数	0
最大出現回数	1
基本型	string
whiteSpace制御	preserve
マッピング元	/社員情報/社員/自宅[position0='2']/電話番号
マッピング先	/社員情報/社員/勤務地/電話番号

## 7.8 マッピングの条件

マッピング元 (変換元ノードまたはファンクション), マッピング先 (変換先ノードまたはファンクション), およびマッピング線の種類や組み合わせによってはマッピングできない場合があります。ここでは, マッピングできる条件について説明します。

### 7.8.1 マッピング対象とマッピング対象外

マッピングでは, マッピング対象の要素 (ノード) とマッピング対象外の要素 (ノード) があります。変換元ノードおよび変換先ノードに, マッピング対象外の要素は指定できません。マッピング対象外の要素を次の表に示します。

表 7-3 マッピング対象外の要素

項目	内容
出現回数が 1 回に固定されていない, かつ名前が指定されていない要素 (コンポジタ)	出現回数が 1 回に固定されていない, かつ名前が指定されていない complexType 要素下の次の要素 (コンポジタ) はマッピング対象外とします。 <ul style="list-style-type: none"> <li>• sequence</li> <li>• choice</li> <li>• all</li> </ul>
出現回数が 1 回に固定されていない sequence 要素 group 参照, または choice 要素 group 参照	次の group 参照はマッピング対象外とします。 <ul style="list-style-type: none"> <li>• sequence 要素下の, 出現回数が 1 回に固定されていない sequence 要素を定義している group の参照</li> <li>• choice 要素を定義している group の参照</li> </ul>
sequence 要素または choice 要素下の名前が指定されていない要素 (コンポジタ)	sequence 要素または choice 要素下の名前が指定されていない次の要素 (コンポジタ) はマッピング対象外とします。 <ul style="list-style-type: none"> <li>• sequence</li> <li>• choice</li> </ul>
complexContent 要素による複合型の拡張	complexContent 要素によって複合型の拡張を定義する場合, 次の要素はマッピング対象外とします。 <ul style="list-style-type: none"> <li>• complexContent 要素下および複合型の拡張で定義された要素が両方とも, 出現回数が 1 回以外の sequence 要素</li> <li>• complexContent 要素下および複合型の拡張で定義された要素の両方またはどちらか一方が choice 要素</li> </ul>
再帰構造	再帰構造が定義されていた場合, 再帰の開始要素だけを表示し, 開始要素はマッピング対象外とします。
混在内容	混在内容である mixed 属性に true が指定されていた場合, 要素間テキストはマッピング対象外とします。なお, 要素間テキストは表示されません。
代替グループ	abstract 属性に true を指定した要素はマッピング対象外とします。また, SubstitutionGroup 属性の指定は無視されるため, SubstitutionGroup 属性を指定した要素はスキーマツリービューアに表示されません。
any 要素および anyAttribute 要素	any 要素および anyAttribute 要素はマッピング対象外とします。

## 7. データ変換の定義

項目	内容
2段階以上ネストした union 要素または list 要素	2段階以上ネストした union 要素または list 要素 ( union 要素または list 要素の属性がさらに定義されている ) はマッピング対象外とします。
fixed が指定されている単純内容および属性の変換先ノード	fixed が指定されている単純内容および属性の変換先ノードはマッピング対象外とします。

### 7.8.2 マッピングできるノードおよびファンクションの対応

マッピング時は、マッピングできる変換元ノード、変換先ノード、およびファンクションの対応を考える必要があります。マッピングできる変換元ノード、変換先ノード、およびファンクションの対応を表 7-4 ~ 表 7-7 に示します。

表 7-4 変換元ノードから変換先ノードへのマッピング

マッピング元	マッピング先		
変換元ノード	変換先ノード		
	単純内容	複合内容	属性
単純内容		×	
複合内容	×	×	×
属性		×	

(凡例)

○ : マッピングできます。

× : マッピングできません。

注

単純内容、複合内容、および属性については、「11.2.4(1) 変換元スキーマツリービューア、マッピングビューア、および変換先スキーマツリービューア」を参照してください。

表 7-5 変換元ノードからファンクションへのマッピング

マッピング元	マッピング先															
変換元ノード	ファンクション															
	文字列系					数値系				ノード系			制御系		その他	
	文字列連結	部分文字列取得	文字数取得	文字列調査	空白除去	数値フォーマット変換	数値演算	端数処理	総和	ノード数取得	ノード名取得	ノード調査	くり返し	選択	定数	値置換え
単純内容																×
複合内容	×	×	×	×	×	×	×	×	×					×	×	×
属性													×		×	

(凡例)

: マッピングできます。

× : マッピングできません。

注

単純内容、複合内容、および属性については、「11.2.4(1) 変換元スキーマツリービューア、マッピングビューア、および変換先スキーマツリービューア」を参照してください。

表 7-6 ファンクションから変換先ノードへのマッピング

マッピング元	マッピング先			
ファンクション	変換先ノード			
	単純内容	複合内容	属性	
文字列系	文字列連結	×		
	部分文字列取得	×		
	文字数取得	×		
	文字列調査	×		
	空白除去	×		
数値系	数値フォーマット変換	×		
	数値演算	×		
	端数処理	×		
	総和	×		
ノード系	ノード数取得	×		

7. データ変換の定義

マッピング元		マッピング先		
ファンクション		変換先ノード		
		単純内容	複合内容	属性
	ノード名取得		×	
	ノード調査		×	
制御系	くり返し			×
	選択			
その他	定数			
	値置換え		×	

(凡例)

: マッピングできます。

x: マッピングできません。

注

単純内容、複合内容、および属性については、「11.2.4(1) 変換元スキーマツリービューア、マッピングビューア、および変換先スキーマツリービューア」を参照してください。

表 7-7 ファンクションからファンクションへのマッピング

マッピング元		マッピング先															
ファンクション		ファンクション															
		文字列系				数値系			ノード系			制御系		その他			
		文字列連結	部分文字列取得	文字数取得	文字列調査	空白除去	数値フォーマット変換	数値演算	端数処理	総和	ノード数取得	ノード名取得	ノード調査	くり返し	選択	定数	値置換え
文字列系	文字列連結								×	×	×	×	×			×	
	部分文字列取得								×	×	×	×	×			×	
	文字数取得								×	×	×	×	×			×	
	文字列調査								×	×	×	×	×			×	
	空白除去								×	×	×	×	×			×	

マッピング元		マッピング先															
ファンクション		ファンクション															
		文字列系					数値系				ノード系			制御系		その他	
		文字列連結	部分文字列取得	文字数取得	文字列調査	空白除去	数値フォーマット変換	数値演算	端数処理	総和	ノード数取得	ノード名取得	ノード調査	くり返し	選択	定数	値置換え
数値系	数値フォーマット変換									×	×	×	×	×		×	
	数値演算									×	×	×	×	×		×	
	端数処理									×	×	×	×	×		×	
	総和									×	×	×	×	×		×	
ノード系	ノード数取得									×	×	×	×	×		×	
	ノード名取得									×	×	×	×	×		×	
	ノード調査									×	×	×	×	×		×	
制御系	くり返し	×	×	×	×	×	×	×	×	×	×	×	×	×		×	×
	選択	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
その他	定数									×	×	×	×	×		×	
	値置換え									×	×	×	×	×		×	

(凡例)

: マッピングできます。

× : マッピングできません。

注

マッピング元のファンクションにくり返しファンクションを一つでも指定した場合、ほかのマッピング元もすべてくり返しファンクションを指定してください。

また、マッピング元のファンクションにくり返しファンクション以外を設定した場合、ほかのマッピング元もすべてくり返しファンクション以外を指定してください。

### 7.8.3 接続できるマッピング線の数

変換元ノード、変換先ノード、およびファンクションを接続するマッピング線は、接続できる数が決められています。出力として接続できるマッピング線の数を表 7-8 に、入力として接続できるマッピング線の数を表 7-9 に示します。

表 7-8 出力として接続できるマッピング線の数

ノードまたはファンクション	接続できるマッピング線の数
変換元ノード	複数 <sup>1</sup>
すべてのファンクション	1 <sup>2</sup>

注 1 上限値はありません。

注 2 定数ファンクションは代入線だけ複数接続できます。

表 7-9 入力として接続できるマッピング線の数

ノードまたはファンクション	接続できるマッピング線の数	
変換先ノード	複数	
文字列系	文字列連結	複数
	部分文字列取得	1
	文字数取得	1
	文字列調査	1
	空白除去	1
数値系	数値フォーマット変換	1
	数値演算	2
	端数処理	1
	総和	複数
ノード系	ノード数取得	1
	ノード名取得	1
	ノード調査	1
制御系	くり返し	1
	選択	複数
その他	定数	0
	値置換え	1

注 上限値はありません。

# 8

## ユーザ定義受付の定義

この章では、ユーザ定義受付の定義について説明します。

---

8.1 ユーザ定義受付の概要

---

8.2 定義の流れ

---

8.3 WSDL の作成

---

8.4 ユーザ定義受付の追加

---

8.5 ユーザ定義受付の内容の確認

---

8.6 ユーザ定義受付の検証

---

8.7 ユーザ定義受付の削除

---

## 8.1 ユーザ定義受付の概要

---

ユーザ定義受付は、ユーザが定義した任意の形式でサービス部品実行要求を受け付け、応答を返すインターフェースです。

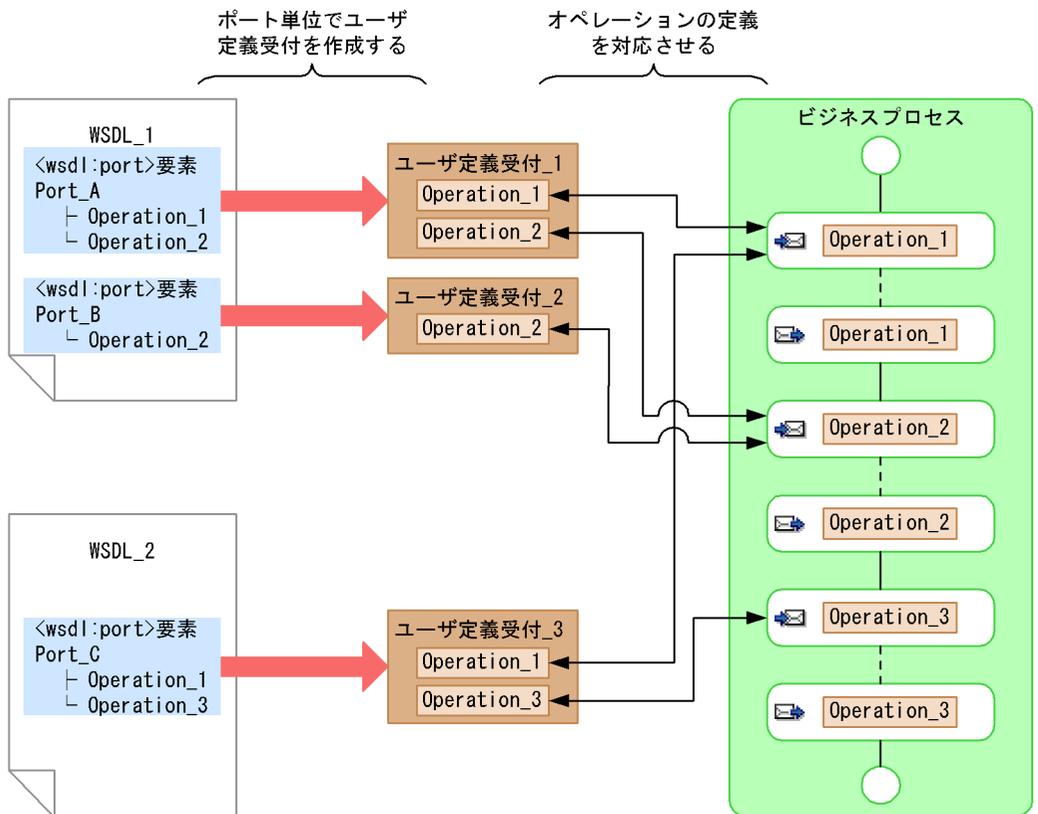
Cosminexus サービスプラットフォームでは、サービスリクエストからのサービス部品の実行要求を受け付ける固定のインターフェースとして、4種類の標準受付（同期受付（Web サービス / SessionBean）、非同期受付（WS-R / DB キュー））を備えています。標準受付を利用するには、それぞれの標準受付の形式に合わせてサービスリクエストを作成し、さらにサービス部品の入出力形式に合わせて電文を作成、解析する必要があります。

これに対し、ユーザ定義受付は、個々のサービス部品およびビジネスプロセスの入出力形式に合わせた形式でサービスリクエストからの要求を受け付けることができます。

なお、サービスリクエストとユーザ定義受付の通信プロトコルは Web サービスだけです。また、ユーザ定義受付でリクエストを受け付けることができる HCSC コンポーネントは、ビジネスプロセスだけです。

ユーザ定義受付、WSDL、ビジネスプロセスの定義の関係について、次の図に示します。

図 8-1 ユーザ定義受付, WSDL, ビジネスプロセスの定義の関係



(凡例)

 : 受付アクティビティ

 : 応答アクティビティ

図 8-1 に示すように、ユーザ定義受付は、WSDL のポート (wsdl:port 要素) ごとに作成します。一つのユーザ定義受付に wsdl:port 要素に含まれるオペレーションがすべて含まれます。

また、図 8-1 の Operation\_2 は、Port\_A および Port\_B の二つの wsdl:port 要素に含まれます。このように、異なる wsdl:port 要素に同じオペレーションが含まれる場合、複数のユーザ定義受付で同じオペレーションを利用できます。図 8-1 では、Operation\_2 は、ユーザ定義受付\_1 およびユーザ定義受付\_2 で利用しています。

## 8.2 定義の流れ

---

ユーザ定義受付の定義の流れには、次の二つがあります。

### ビジネスプロセスの定義前にユーザ定義受付を定義する

次の二つの条件を満たす場合、先に WSDL を基にユーザ定義受付を作成し、そのあとでビジネスプロセスの受付アクティビティと応答アクティビティを定義します。

- サービスリクエストとのインターフェースが規定済み
- 規定したインターフェースに従った WSDL が作成済み

この場合、ユーザ定義受付の作成時に入力した WSDL のインターフェースに合うように、受付アクティビティと応答アクティビティのオペレーションおよび電文定義をビジネスプロセス上で定義する必要があります。

### ビジネスプロセスの定義後にユーザ定義受付を定義する

すでに定義されたビジネスプロセスがある場合、定義済みのビジネスプロセスに合ったユーザ定義受付を定義します。ユーザ定義受付を追加する前に、ビジネスプロセスの受付アクティビティと応答アクティビティのオペレーションおよび割当変数の電文定義に合う WSDL を作成しておく必要があります。

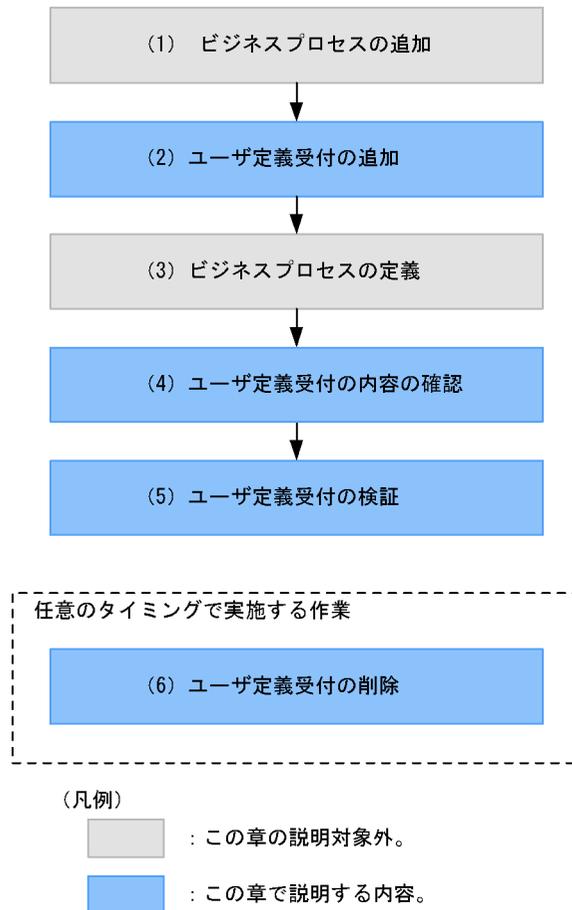
ユーザ定義受付は、ビジネスプロセスの受付アクティビティと応答アクティビティに定義されているオペレーション、および割当変数を呼び出せるように定義します。

ユーザ定義受付の定義の流れとして、これら二つの定義の流れを次に説明します。

### 8.2.1 ビジネスプロセスを定義する前にユーザ定義受付を定義する

ビジネスプロセスを定義する前にユーザ定義受付を定義し、そのインターフェースに合うようにビジネスプロセスの受付アクティビティ、および応答アクティビティを定義する流れについて、次の図に示します。

図 8-2 ユーザ定義受付の定義の流れ（ビジネスプロセスの定義前に定義する場合）



ビジネスプロセスを定義する前にユーザ定義受付を定義する場合の作業について次に説明します。

### (1) ビジネスプロセスの追加

ユーザ定義受付を定義するビジネスプロセスを作成します。(2)で定義するユーザ定義受付の内容と合うインターフェースとなるようにビジネスプロセスを定義します。そのため、ここではアクティビティの配置や変数の定義、各アクティビティの詳細を定義するダイアログでの情報の入力などをする必要はありません。

ビジネスプロセスの追加については、「6.2.1 ビジネスプロセスを新規に追加する」を参照してください。

### (2) ユーザ定義受付の追加

作成済みの WSDL をユーザ定義受付追加ウィザードで入力して、ユーザ定義受付を追加します。

## 8. ユーザ定義受付の定義

ユーザ定義受付を新規に追加する方法の詳細については、「8.4 ユーザ定義受付の追加」を参照してください。

### (3) ビジネスプロセスの定義

ユーザ定義受付で定義されているすべてのオペレーションに対応する受付アクティビティ、および応答アクティビティを定義します。

受付アクティビティおよび応答アクティビティで、ユーザ定義受付で規定された電文フォーマットを持つ変数を定義し、割当変数として設定する手順を次に示します（次の手順は受付アクティビティの場合です。応答アクティビティの場合も同様の手順です）。

1. [ 受付アクティビティ ] ダイアログで、ユーザ定義受付で定義されているオペレーション名を設定します。
2. [ 割当変数 ] の [ 編集 ] ボタンをクリックします。  
[ 変数・相関セット一覧 ] ダイアログが表示されます。
3. 変数名を設定します。
4. [ 取込 ] ボタンをクリックします。  
[ 電文フォーマットの取込 ] ダイアログが表示されます。[ 電文フォーマットの取込 ] ダイアログには次の情報が表示されています。

#### 受付名

手順 1. で設定したオペレーションを持つユーザ定義受付の受付名が設定されています。

#### オペレーション名

手順 1. で設定した値が設定されています。

#### 電文種別

受付アクティビティの場合は要求電文が、応答アクティビティの場合は応答電文が選択されています。

#### 電文フォーマット名

変数名が設定されています。

なお、手順 1. で設定したオペレーション名がユーザ定義受付に存在しない場合、受付名およびオペレーション名が空白になります。この場合、手順 1 に戻って正しいオペレーション名を指定してください。

また、電文フォーマット名には、初期値として変数名が設定されています。必要に応じて適当な名称に変更してください。

5. [ OK ] ボタンをクリックします。  
[ 電文フォーマットの取込 ] ダイアログが閉じられ、[ 変数・相関セット一覧 ] ダイアログが操作できる状態になります。
6. [ 変数・相関セット一覧 ] ダイアログの [ 追加 ] ボタンをクリックします。
7. [ 変数・相関セット一覧 ] ダイアログの変数一覧から追加した変数を選択します。

8. [変数・相関セット一覧] ダイアログの [OK] ボタンをクリックします。

この操作で利用するダイアログについては、「11.4 ビジネスプロセスに関連するダイアログ」を参照してください。

ポイント

---

ユーザ定義受付のオペレーションにフォルトが含まれている場合、ビジネスプロセスではフォルト名を指定した応答アクティビティを定義します。[応答アクティビティ] ダイアログの [フォルト名] に、ユーザ定義受付定義画面の [フォルト電文] の [フォルト名] に表示されている値を指定してから、手順 2. ~ 8. と同様の方法で変数を割り当てます。

また、ユーザ定義受付のオペレーションに複数のフォルトが含まれている場合、含まれるフォルトの数だけ応答アクティビティを追加し、それぞれに対応するフォルト名を設定する必要があります。また、各応答アクティビティで、手順 2. ~ 8. と同様の方法で変数を割り当てます。

---

#### (4) ユーザ定義受付の内容の確認

追加されたユーザ定義受付の内容を確認します。ユーザ定義受付の内容は、ビジネスプロセスの定義内容と合致している必要があります。

ユーザ定義受付の確認の詳細については、「8.5 ユーザ定義受付の内容の確認」を参照してください。

#### (5) ユーザ定義受付の検証

定義したユーザ定義受付とビジネスプロセス定義との整合性を検証できます。

ユーザ定義受付の検証方法の詳細については、「8.6 ユーザ定義受付の検証」を参照してください。

#### (6) ユーザ定義受付の削除

不要になったユーザ定義受付は、必要に応じて削除できます。

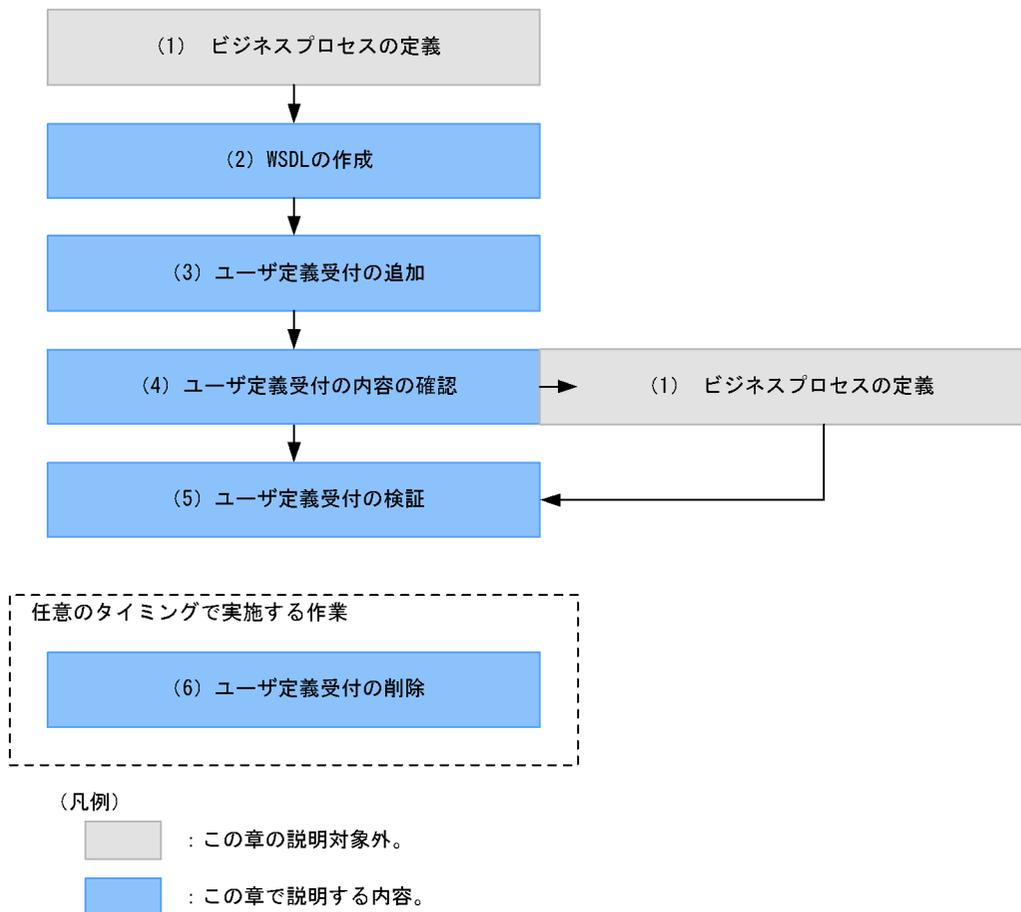
ユーザ定義受付の削除方法については、「8.7 ユーザ定義受付の削除」を参照してください。

## 8.2.2 ビジネスプロセスの定義後にユーザ定義受付を定義する

定義済みのビジネスプロセスのインターフェースに合うようにユーザ定義受付を定義する流れについて、次の図に示します。

## 8. ユーザ定義受付の定義

図 8-3 ユーザ定義受付の定義の流れ（ビジネスプロセスの定義後に定義する場合）



ビジネスプロセスの定義後にユーザ定義受付を定義する場合の作業について次に説明します。

### (1) ビジネスプロセスの定義

ユーザ定義受付を定義する前に、ビジネスプロセスを定義しておく必要があります。ビジネスプロセスが定義途中の場合でも、サービスリクエストとのインターフェースとなる受付アクティビティおよび応答アクティビティを配置し、オペレーション名および割当変数の設定まで行ってください。

ビジネスプロセスの定義については、「6.3 ビジネスプロセスの内容の定義」を参照してください。

### (2) WSDL の作成

ユーザ定義受付を定義する際に利用する WSDL を作成します。

WSDLの作成の詳細については、「8.3 WSDLの作成」を参照してください。

### (3) ユーザ定義受付の追加

ユーザ定義受付追加ウィザードを利用して、ユーザ定義受付を追加します。

ユーザ定義受付を新規に追加する方法の詳細については、「8.4 ユーザ定義受付の追加」を参照してください。

### (4) ユーザ定義受付の内容の確認

追加されたユーザ定義受付の内容を確認します。ユーザ定義受付の内容は、ビジネスプロセスの定義内容と合致している必要があります。

ユーザ定義受付の確認の詳細については、「8.5 ユーザ定義受付の内容の確認」を参照してください。

### (5) ユーザ定義受付の検証

定義したユーザ定義受付とビジネスプロセス定義との整合性を検証できます。

ユーザ定義受付の検証方法の詳細については、「8.6 ユーザ定義受付の検証」を参照してください。

### (6) ユーザ定義受付の削除

不要になったユーザ定義受付は、必要に応じて削除できます。

ユーザ定義受付の削除方法については、「8.7 ユーザ定義受付の削除」を参照してください。

## 8.3 WSDL の作成

ユーザ定義受付を利用してサービス部品を呼び出すときのインターフェース情報を定義する WSDL ファイルを作成します。

ここでは、例を用いて WSDL ファイルの作成方法について説明します。

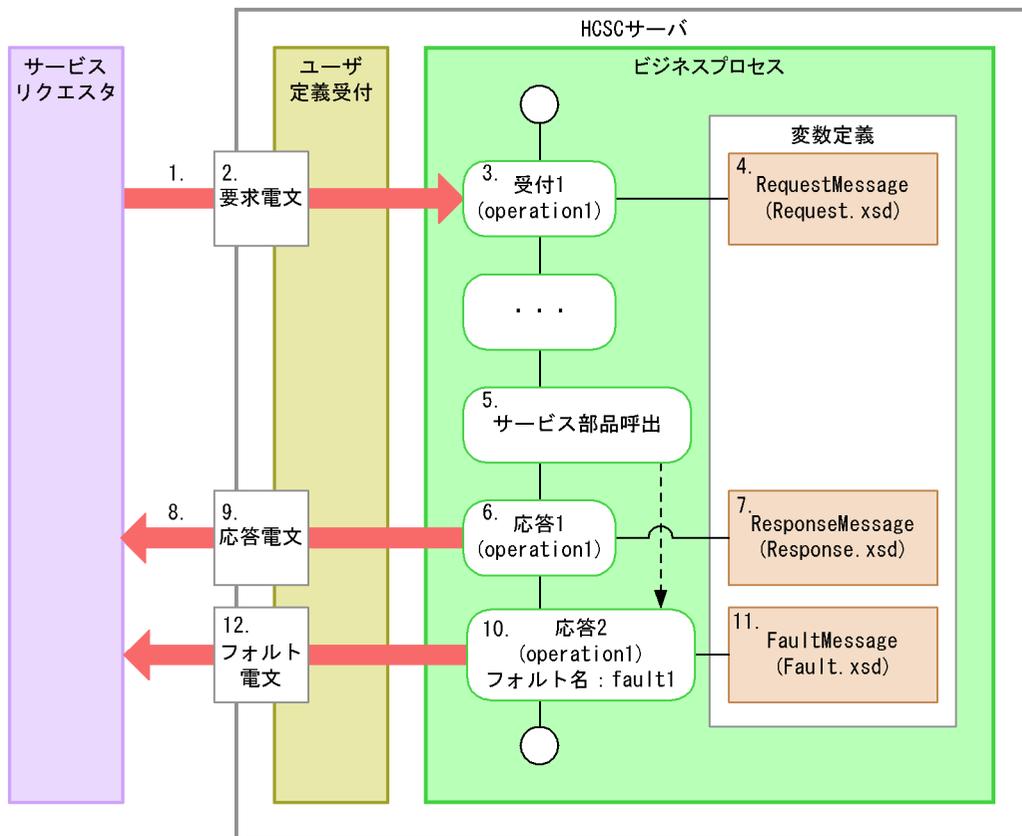
### 8.3.1 WSDL の作成に使用するビジネスプロセスの例

WSDL ファイルの作成方法の説明に使用するビジネスプロセスの例の内容について説明します。

#### (1) ビジネスプロセスの例

WSDL の作成の例として使用するビジネスプロセスを次の図に示します。

図 8-4 WSDL の作成の例として使用するビジネスプロセス



1. サービスリクエスタがビジネスプロセスにサービス部品の実行を要求します。
2. サービス部品の実行要求では、XML の要求電文が送付されます。

3. ビジネスプロセスは XML の要求電文を受付 1 という受付アクティビティで受け取ります。このときのオペレーション名は operation1 です。
4. 受付 1 には変数 RequestMessage を割り当て、要求電文を規定する電文フォーマットとして Request.xsd が指定されています。
5. ビジネスプロセスからサービス部品を呼び出します。フォルトが発生した場合は項番 10. でフォルトが返されます。
6. ビジネスプロセスが、サービス部品の実行結果を XML の応答電文として構成し、応答 1 という応答アクティビティで返します。
7. 応答 1 には変数 ResponseMessage を割り当て、応答電文を規定する電文フォーマットとして Response.xsd が指定されています。
8. サービスリクエストへの応答が返されます。
9. サービスリクエストへの応答では、XML の応答電文が送付されます。
10. 項番 5. でフォルトが発生した場合、応答 2 という応答アクティビティから fault1 というフォルトが返されます。
11. 応答 2 には変数 FaultMessage を割り当て、フォルト電文を規定する電文フォーマットとして Fault.xsd が指定されています。
12. サービスリクエストにフォルト電文が送付されます。

## (2) 電文フォーマットの内容の例

図 8-4 に示すビジネスプロセスで利用する電文フォーマットの例を次に示します。

### Request.xsd

- XML スキーマソース

文字列の値を持つ単一の要素 <Request1> を規定する要求電文用の電文フォーマットです。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.example.org/request"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Request1" type="xsd:string" />
</xsd:schema>
```

- この電文フォーマットの規定に従って送付される XML

```
<Request1>Sample request message</Request1>
```

### Response.xsd

- XML スキーマソース

boolean 値を持つ単一の要素 <Response1> を規定する応答電文用の電文フォーマットです。

## 8. ユーザ定義受付の定義

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.example.org/response"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Response1" type="xsd:boolean" />
</xsd:schema>
```

- この電文フォーマットの規定に従って送付される XML

```
<Response1>true</Response1>
```

### Fault.xsd

- XML スキーマソース  
 <Fault1> という要素を規定し、さらにその子要素として int 値を持つ要素  
 <param1> を規定するフォルト電文用の電文フォーマットです。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.example.org/service"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:svc="http://www.example.org/service">
  <xsd:element name="Fault1" type="svc:SampleData"/>
  <xsd:complexType name="SampleData">
    <xsd:sequence>
      <xsd:element name="param1" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

- この電文フォーマットの規定に従って送付される XML

```
<Fault1>
  <param1>123</param1>
</Fault1>
```

## 8.3.2 WSDL の作成例

ここでは、「8.3.1 WSDL の作成に使用するビジネスプロセスの例」に示すビジネスプロセスに合わせた WSDL の作成例を示します。WSDL の形式に関する条件や注意事項については、「2.5.1(3) WSDL の形式」を参照してください。

WSDL の作成例は「Sample\_Reception.wsdl」とします。Sample\_Reception.wsdl の概要を次に示します。

- 「8.3.1 WSDL の作成に使用するビジネスプロセスの例」に示す内容を前提としています。

- ユーザ定義受付はオペレーション単位に作成することもできますが、Sample\_Reception.wsdl は、一つのユーザ定義受付にすべてのオペレーション含むときの WSDL です。
- 作成する WSDL の通信スタイルは「document」スタイルを推奨しています。そのため、Sample\_Reception.wsdl は「document」スタイルの WSDL となっています。

WSDL の作成例「Sample\_Reception.wsdl」を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  targetNamespace="http://www.example.org/service"
  xmlns:svc="http://www.example.org/service"
  xmlns:req="http://www.example.org/request"
  xmlns:res="http://www.example.org/response"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xsd:schema targetNamespace="http://www.example.org/request">
      <xsd:element name="Request1" type="xsd:string" />
    </xsd:schema>
    <xsd:schema targetNamespace="http://www.example.org/response">
      <xsd:element name="Response1" type="xsd:boolean" />
    </xsd:schema>
    <xsd:schema targetNamespace="http://www.example.org/service">
      <xsd:element name="Fault1" type="svc:SampleData"/>
      <xsd:complexType name="SampleData">
        <xsd:sequence>
          <xsd:element name="param1" type="xsd:int"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:schema>
  </wsdl:types>

  <wsdl:message name="ReqMsg1">
    <wsdl:part element="req:Request1" name="reqParam"/>
  </wsdl:message>
  <wsdl:message name="ResMsg1">
    <wsdl:part element="res:Response1" name="resParam"/>
  </wsdl:message>
  <wsdl:message name="FltMsg1">
    <wsdl:part element="svc:Fault1" name="fltParam"/>
  </wsdl:message>
  <wsdl:portType name="SamplePortType">
    <wsdl:operation name="operation1">
      <wsdl:input message="svc:ReqMsg1"/>
      <wsdl:output message="svc:ResMsg1"/>
      <wsdl:fault message="svc:FltMsg1" name="Fault1"/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

## 8. ユーザ定義受付の定義

```
<wsdl:binding name="SampleBinding" type="svc:SamplePortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="operation1">
    <soap:operation soapAction=""/>
    <wsdl:input>
      <soap:body/>
    </wsdl:input>
    <wsdl:output>
      <soap:body/>
    </wsdl:output>
    <wsdl:fault name="Fault1">
      <soap:fault name="Fault1"/>
    </wsdl:fault>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="SampleService">
  <wsdl:port binding="svc:SampleBinding" name="UserInfo">
    <soap:address location="http://localhost:80/SampleService/services/
UserInfo" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

この作成例「Sample\_Reception.wsdl」は、次の(1)～(8)の作業を行って作成します。

### (1) オペレーションとフォルト名のリストアップ

ビジネスプロセスの受付アクティビティ、および応答アクティビティで定義されている同期のオペレーションをリストアップします。また、フォルト名が定義されている応答アクティビティがある場合は、フォルト名もリストアップします。

作成例ではオペレーションは「operation1」が該当します。また、フォルト名は「Fault1」が該当します。

### (2) 割当変数の電文フォーマット定義ファイルのリストアップ

受付アクティビティ、および応答アクティビティが使用している割当変数の電文フォーマット定義ファイルをリストアップします。

作成例では、「Request.xsd」、「Response.xsd」、および「Fault.xsd」が該当します。

**!** 注意事項

- リストアップした電文フォーマット定義ファイルの中に、同じ名前空間のスキーマが複数存在する場合、一つのスキーマに集約する必要があります。  
このとき、次の二つの条件を満たす場合、割当変数の電文フォーマット定義ファイルを名前の重複がなくなるよう変更してください。
  - ・スキーマの中の要素について名前の重複がある。
  - ・その要素の内容が異なる。
 なお、割当変数の電文フォーマット定義ファイルが複数ファイルの場合（外部の XML ファイルを参照している場合）、外部の XML ファイルのスキーマも含めて要素の重複がなくなるよう変更してください。  
また、それぞれのスキーマの属性が相反する値を持つ場合、値の変更による影響を考慮してスキーマを集約してください。
- リストアップした電文フォーマット定義ファイルの中に、targetNamespace 属性を持たないスキーマ（カメレオンスキーマ）がある場合、割当変数の電文フォーマット定義ファイルは、targetNamespace 属性を指定したスキーマに変更してください。
- リストアップしたフォルト電文の電文フォーマット定義ファイルのルート要素が複合型でない場合、割当変数の電文フォーマット定義ファイルのルート要素を、複合型になるように変更してください。
- フォルト電文のルート要素の名前空間は、wsdl:definitions 要素の名前空間と一致させる必要があります。  
このため、次の二つの条件を満たす場合、その割当変数の電文フォーマットファイルを変更してください。
  - ・フォルト電文が複数存在する。
  - ・ルート要素の名前空間が異なる。
 異なるオペレーションのフォルト電文同士の名前空間が異なる場合、オペレーションごとに WSDL ファイルを分けて作成し、ユーザ定義受付を複数作成して対応できます。

**(3) WSDL ファイルのひな形の作成（wsdl:definitions 要素の定義）**

WSDL ファイルのひな形として、XML 宣言および wsdl:definitions 要素を定義します。  
wsdl:definitions 要素には、次の表に示す属性を追加します。

表 8-1 wsdl:definitions 要素に追加する属性

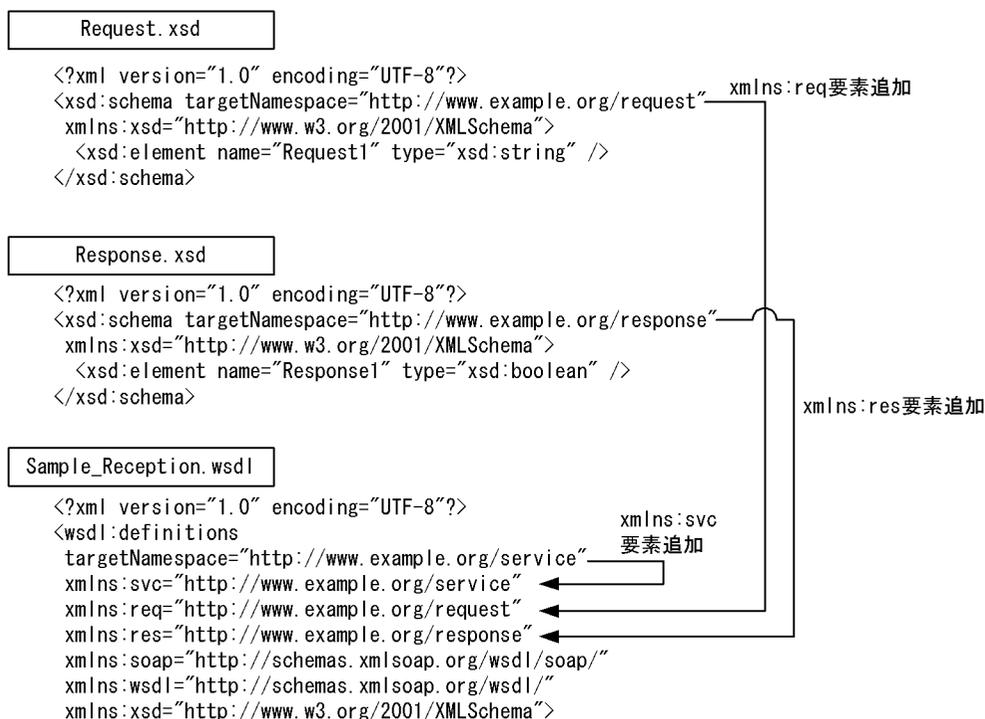
属性	意味	値	個数
targetNamespace	サービス部品の名前空間 URI	任意の URI を定義できます。ただし、フォルト電文がある場合には、フォルト電文の電文フォーマット定義ファイルの xsd:schema 要素が持つ targetNamespace 属性値と同じ URI を定義します。	1
xmlns:svc	サービス部品のプレフィックス宣言	targetNamespace 属性と同じ値を定義します。	1
xmlns:req	要求電文の電文フォーマットプレフィックス宣言	要求電文の電文フォーマット定義ファイルの xsd:schema 要素が持つ targetNamespace 属性値と同じ URI を定義します。	1

## 8. ユーザ定義受付の定義

属性	意味	値	個数
xmlns:res	応答電文の電文フォーマットプレフィックス宣言	応答電文の電文フォーマット定義ファイルの <code>xsd:schema</code> 要素が持つ <code>targetNamespace</code> 属性値と同じ URI を定義します。	1
xmlns:soap	SOAP のプレフィックス宣言	既定の値「 <code>http://schemas.xmlsoap.org/wsdl/soap/</code> 」を定義します。	1
xmlns:wsdl	WSDL のプレフィックス宣言	既定の値「 <code>http://schemas.xmlsoap.org/wsdl/</code> 」を定義します。	1
xmlns:xsd	XML スキーマのプレフィックス宣言	既定の値「 <code>http://www.w3.org/2001/XMLSchema</code> 」を定義します。	1

`wsdl:definitions` 要素の定義例を次の図に示します。

図 8-5 `wsdl:definitions` 要素の定義例



### 複数のオペレーションを使用する場合

作成例 (`Sample_Reception.wsdl`) では、オペレーションを一つだけ使用していますが、複数のオペレーションを使用する場合、複数の要求電文、応答電文を扱うこととなります。これらの電文が異なる複数の名前空間を使用している場合には、次の表に示すように、それぞれのプレフィックス宣言を `wsdl:definitions` 要素に定義

します。

表 8-2 wsdl:definitions 要素に追加する属性（複数のオペレーションを使用する場合）

属性	意味	値
xmlns:reqN(N>0)	要求電文が複数の名前空間に分かれる場合のプレフィックス宣言	要求電文の電文フォーマット定義ファイルの xsd:schema 要素が持つ targetNamespace 属性値と同じ URI を定義します。 例 http://www.example.org/request1
xmlns:resN(N>0)	応答電文が複数の名前空間に分かれる場合のプレフィックス宣言	応答電文の電文フォーマット定義ファイルの xsd:schema 要素が持つ targetNamespace 属性値と同じ URI を定義します。 例 http://www.example.org/response1

定義した内容は wsdl:message 要素の子要素である wsdl:part 要素の element 属性に定義します。wsdl:message 要素の定義については、「8.3.2(5) wsdl:message 要素の定義」を参照してください  
複数のオペレーションを使用する場合の定義例を次の図に示します。

図 8-6 wsdl:definitions 要素の定義例 (複数のオペレーションを使用する場合)

```

Sample_Reception.wsdl
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  targetNamespace="http://www.example.org/service"
  xmlns:svc="http://www.example.org/service"
  xmlns:req1="http://www.example.org/request1"
  xmlns:req2="http://www.example.org/request2" } プレフィックス宣言
  :
<wsdl:types>
  <xsd:schema targetNamespace="http://www.example.org/request">
    <xsd:element name="Request1" type="xsd:string" />
  </xsd:schema>
  <xsd:schema targetNamespace="http://www.example.org/response">
    <xsd:element name="Response1" type="xsd:boolean" />
  </xsd:schema>
  <xsd:schema targetNamespace="http://www.example.org/service">
    <xsd:element name="Fault1" type="svc:SampleData"/>
    <xsd:complexType name="SampleData">
      <xsd:sequence>
        <xsd:element name="param1" type="xsd:int"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:schema>
</wsdl:types>

<wsdl:message name="ReqMsg1">
  <wsdl:part name="reqParam" element="req1:Request1" />
</wsdl:message>
<wsdl:message name="ReqMsg2">
  <wsdl:part name="reqParam" element="req2:Request2" />
</wsdl:message>

```

プレフィックスを使用

#### (4) wsdl:types 要素の定義

wsdl:types 要素とその下位要素、および属性を定義します。

- (3) で定義した wsdl:definitions 要素の子要素として、wsdl:types 要素を定義します。
- (2) でリストアップした電文フォーマット定義ファイルの xsd:schema 要素を targetNamespace 属性で分類し、それぞれの targetNamespace の値ごとに一つずつ xsd:schema 要素を定義します。
- 手順 2. で定義した xsd:schema 要素の属性として、電文フォーマット定義ファイルの xsd:schema 要素が持つすべての属性を追加します。
- 手順 2. で定義した xsd:schema 要素の子要素として、電文フォーマット定義ファイルの xsd:schema 要素が持つすべての子要素を追加します。
- 複数のフォルト電文があり、名前空間が異なる場合、フォルト電文の名前空間を wsdl:definitions 要素の targetNamespace 属性の値に変更します。

wsdl:types 要素以下に定義する要素と属性を次の表に示します。

表 8-3 wsdl:types 要素以下に定義する要素と属性

要素	属性	意味	値	個数
wsdl:types	-	型定義要素	-	1
xsd:schema	-	スキーマ要素	-	1 ~ 電文フォーマット定義ファイルが使う名前空間の数
	targetNamespace	名前空間属性	電文の targetNamespace と同じ値を定義します。	1
	xsd:schema の属性	属性	電文の属性と同じ値を定義します。	1 ~ 属性の数
xsd:schema の子要素	-	電文の内容	電文が持つ xsd:schema 要素のすべての子要素をコピーします。	1 ~ 同一名前空間を使用する電文フォーマットの xsd:schema 要素の子要素の数

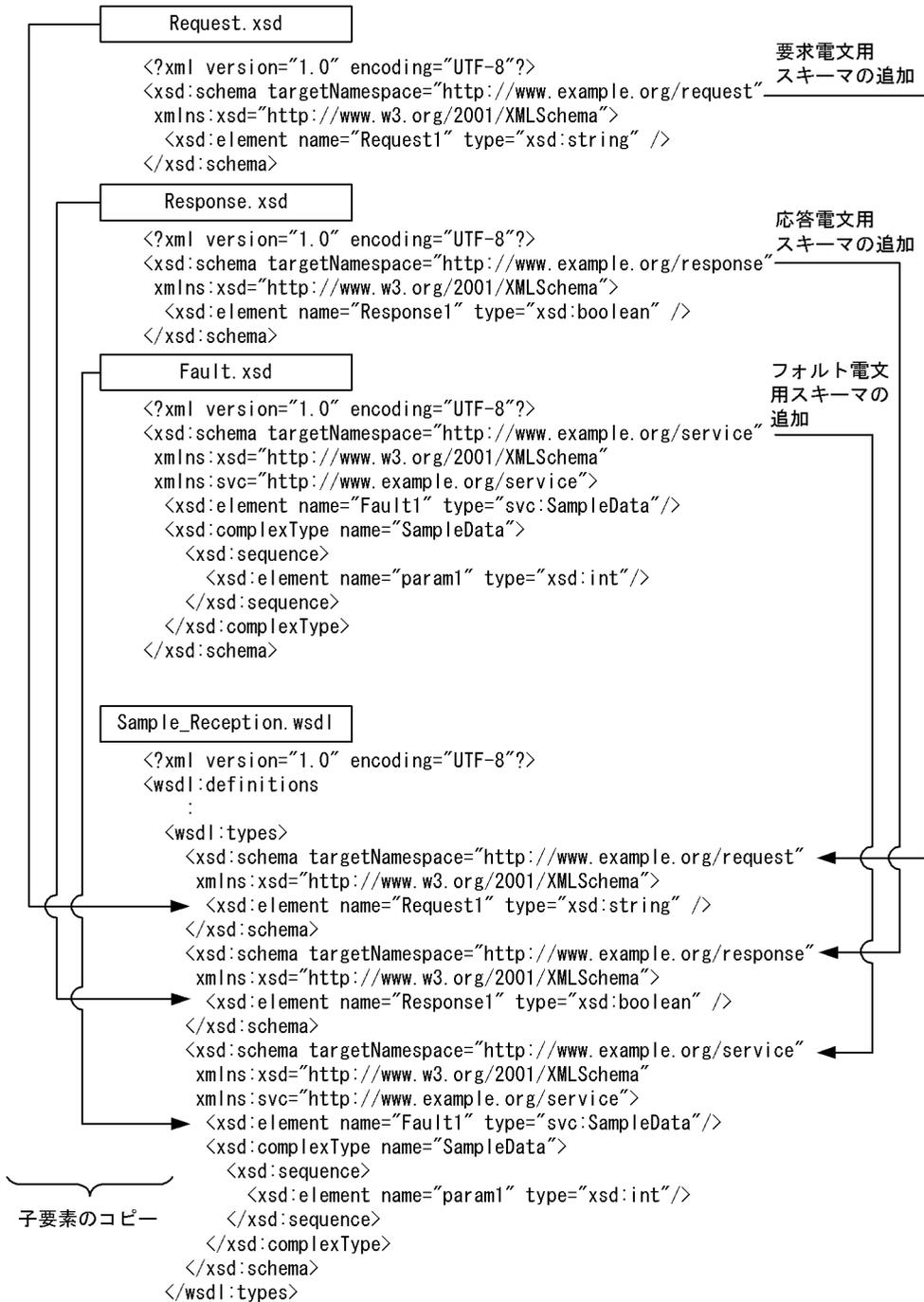
(凡例)

- : 該当しません。

wsdl:types 要素の定義例を次の図に示します。

8. ユーザ定義受付の定義

図 8-7 wsdl:types 要素の定義例



### (5) wsdl:message 要素の定義

wsdl:message 要素とその下位要素，および属性を定義します。

- (3) で定義した wsdl:definitions 要素の子要素として wsdl:message 要素を定義します。  
wsdl:message 要素は WSDL で定義するオペレーションが持つすべての要求電文，応答電文，およびフォルト電文の数だけ定義します。  
作成例では，オペレーションが一つあります。そのオペレーションには，要求電文が一つ，応答電文が一つ，フォルト電文が一つあるので，wsdl:message 要素を三つ定義します。
- wsdl:message 要素に name 属性を定義します。  
name 属性の値は，ファイル内で一意な値になるよう，ほかの wsdl:message 要素の name 属性とは異なる値を定義します。  
作成例では，operation1 の要求電文には ReqMsg1，応答電文には ResMsg1 と定義しています。フォルト電文は個数に合わせて FltMsgN (N は 1 以上) とします。
- wsdl:message 要素の子要素として，wsdl:part 要素を一つずつ定義します。
- wsdl:part 要素に name 属性と element 属性を定義します。  
name 属性には任意の文字列を指定します。  
element 属性の値には，(4) でコピーした xsd:schema 要素の子要素から，ルート要素を選択して指定します。表 8-1 に示すプレフィックスとルート要素名を使用して QName で指定します。

wsdl:message 要素以下に定義する要素と属性を次の表に示します。

表 8-4 wsdl:message 要素以下に定義する要素と属性

要素	属性	意味	値	個数
wsdl:message	-	メッセージ定義要素	-	1
	name	メッセージ名属性	ほかの wsdl:message 要素の name 属性とは異なる一意の文字列を定義します。	1
wsdl:part	-	メッセージパート定義要素	-	1
	name	メッセージパート名属性	任意の文字列 を定義します。	1
	element	要素参照属性	該当する電文フォーマットファイルのルート要素を定義します。	1

(凡例)

- : 該当しません。

注

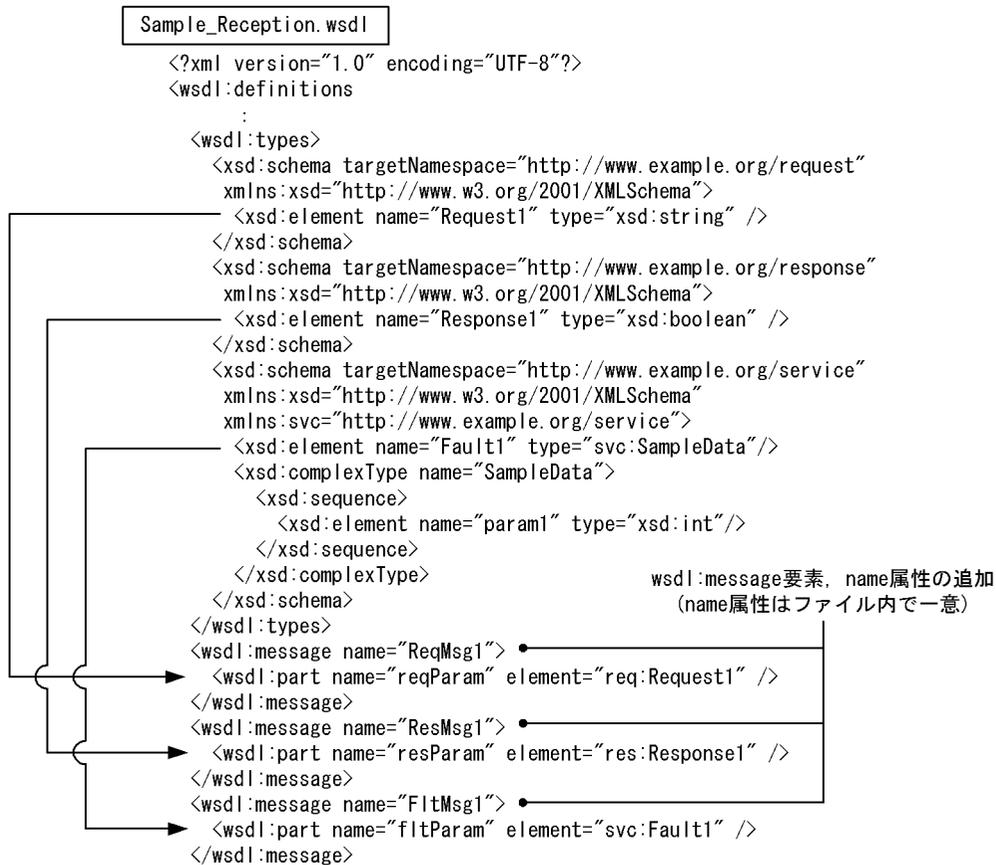
使用できる文字種には制限があります。使用できる文字種の詳細については，マニュアル「Cosminexus SOAP アプリケーション開発ガイド」の WSDL 定義の名称に関する内容を参照

## 8. ユーザ定義受付の定義

してください。

wsdl:message 要素の定義例を次の図に示します。

図 8-8 wsdl:message 要素の定義例



- wsdl:part要素の追加
- element属性にルート要素を指定

### (6) wsdl:portType 要素の定義

wsdl:portType 要素とその下位要素, および属性を定義します。

1. (3) で定義した wsdl:definitions 要素の子要素として, wsdl:portType 要素を一つ定義します。
2. wsdl:portType 要素に name 属性を定義します。  
name 属性の値には, 任意の文字列を指定できます。

3. wsdl:portType 要素の子要素として、wsdl:operation 要素を定義します。  
wsdl:operation 要素は、(1) でリストアップしたオペレーションの数だけ定義します。
4. wsdl:operation 要素に name 属性を定義します。  
name 属性の値には、(1) でリストアップしたオペレーションのオペレーション名を指定します。  
作成例では、wsdl:operation 要素を一つ定義し、name 属性は「operation1」とします。
5. wsdl:operation 要素の子要素として、wsdl:input 要素、wsdl:output 要素、および wsdl:fault 要素を定義します。  
wsdl:input 要素、wsdl:output 要素、wsdl:fault 要素の順序で定義します。  
フォルト電文用の応答アクティビティが存在しない場合、wsdl:fault 要素は省略できます。フォルト電文用の応答アクティビティが複数存在する場合、その数だけ定義します。  
作成例では、wsdl:input 要素、wsdl:output 要素、および wsdl:fault 要素を一つずつ定義します。
6. wsdl:input 要素および wsdl:output 要素に message 属性を定義します。  
message 属性の値には、(5) で定義した wsdl:message 要素の name 属性の値を QName で指定します。
7. wsdl:fault 要素に name 属性と message 属性を定義します。  
name 属性の値には、(1) でリストアップしたフォルト名を指定します。  
message 属性の値には、(5) で定義した wsdl:message 要素の name 属性の値を QName で指定します。

wsdl:portType 要素以下に定義する要素と属性を次の表に示します。

表 8-5 wsdl:portType 要素以下に定義する要素と属性

要素	属性	意味	値	個数
wsdl:portType	-	ポートタイプ定義要素	-	1
	name	ポートタイプ名属性	任意の文字列 を定義します。	1
wsdl:operation	-	オペレーション定義要素	-	1 ~ オペレーションの個数
	name	オペレーション名属性	オペレーション名を定義します。	1
wsdl:input	-	要求メッセージ定義要素	-	1
	message	メッセージ参照属性	(5) で定義した wsdl:message 要素の name 属性値を QName で指定します。	1

## 8. ユーザ定義受付の定義

要素	属性	意味	値	個数
wsdl:output	-	応答メッセージ定義要素	-	1
	message	メッセージ参照属性	(5) で定義した wsdl:message 要素の name 属性値を QName で指定します。	1
wsdl:fault	-	フォルトメッセージ定義要素	-	1 ~ フォルトの個数
	name	フォルトメッセージ名属性	フォルト名を定義します。	1
	message	フォルトメッセージ参照属性	(5) で定義した wsdl:message 要素の name 属性値を QName で指定します。	1

(凡例)

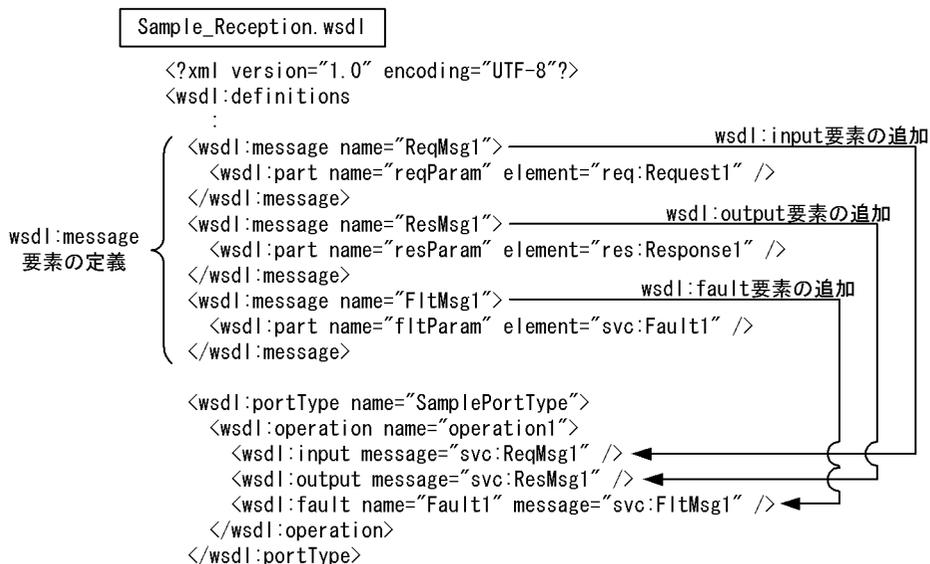
- : 該当しません。

注

使用できる文字種には制限があります。使用できる文字種の詳細については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」の WSDL 定義の名称に関する内容を参照してください。

wsdl:portType 要素の定義例を次の図に示します。

図 8-9 wsdl:portType 要素の定義例



## (7) wsdl:binding 要素の定義

wsdl:binding 要素とその下位要素，および属性を定義します。

1. (3) で定義した wsdl:definitions 要素の子要素として，wsdl:binding 要素を定義します。
2. wsdl:binding 要素に name 属性と type 属性を定義します。  
name 属性の値には，任意の文字列を指定できます。  
type 属性の値には，(6) で定義した wsdl:portType 要素の name 属性の値を QName で指定します。
3. wsdl:binding 要素の子要素として，soap:binding 要素を定義します。
4. soap:binding 要素に transport 属性を定義します。  
transport 属性の値は固定値です。「http://schemas.xmlsoap.org/soap/http」と定義します。
5. wsdl:binding 要素の子要素として，wsdl:operation 要素を定義します。  
wsdl:operation 要素は，(1) でリストアップしたオペレーションの数だけ定義します。
6. wsdl:operation 要素に name 属性を定義します。  
name 属性の値には，(6) で定義した wsdl:portType 要素の子要素である wsdl:operation 要素の name 属性の値と同じ値を指定します。
7. wsdl:operation 要素の子要素として，soap:operation 要素を定義します。
8. soap:operation 要素に soapAction 属性を指定します。  
soapAction 属性の値は空にします。
9. wsdl:operation 要素の子要素として，wsdl:input 要素，wsdl:output 要素，および wsdl:fault 要素を定義します。  
wsdl:input 要素，wsdl:output 要素，wsdl:fault 要素の順序で定義します。  
フォルト電文用の応答アクティビティが存在しない場合，wsdl:fault 要素は省略できます。フォルト電文用の応答アクティビティが複数存在する場合，その数だけ定義します。  
作成例では，wsdl:input 要素，wsdl:output 要素，および wsdl:fault 要素を一つずつ定義します。
10. wsdl:input 要素と wsdl:output 要素の子要素として，soap:body 要素を定義します。  
soap:body 要素には，値は指定しません。
11. wsdl:fault 要素に name 属性を定義します。  
name 属性の値には，(6) で定義した wsdl:operation 要素の子要素である wsdl:fault 要素の name 属性の値と同じ値を指定します。
12. wsdl:fault 要素の子要素として，soap:fault 要素を定義します。
13. soap:fault 要素に name 属性を定義します。  
name 属性の値には，親要素である wsdl:fault 要素の name 属性と同じ値を指定しま

8. ユーザ定義受付の定義

す。

wSDL:binding 要素以下に定義する要素と属性を次の表に示します。

表 8-6 wSDL:binding 要素以下に定義する要素と属性

要素	属性	意味	値	個数
wSDL:binding	-	バインディング定義要素	-	1
	name	バインディング名属性	任意の文字列 を定義します。	1
	type	ポートタイプ参照属性	(6) で定義した wSDL:portType の name 属性値を QName で指定します。	1
SOAP:binding	-	SOAP バインディング定義要素	-	1
	transport	SOAP バインディング伝送形態定義属性	「http://schemas.xmlsoap.org/soap/http」を定義します (固定値)。	1
wSDL:operation	-	オペレーション定義要素	-	1 ~ オペレーションの個数
	name	オペレーション名属性	オペレーション名を定義します。 (6) で定義した wSDL:operation 要素の子要素である wSDL:fault 要素の name 属性の値と同じ値を指定します。	1
SOAP:operation	-	SOAP バインディングオペレーション定義要素	-	1
	soapAction	SOAPAction ヘッダ定義属性	-	1
wSDL:input	-	要求メッセージ定義要素	-	1
SOAP:body	-	SOAP バインディング Body 要素定義要素	-	1
wSDL:output	-	応答メッセージ定義要素	-	1
SOAP:body	-	SOAP バインディング Body 要素定義要素	-	1

要素	属性	意味	値	個数
wsdl:fault	-	フォルトメッセージ定義要素	-	1 ~ フォルトの個数
	name	フォルトメッセージ名属性	フォルト名を定義します。	1
soap:fault	-	SOAP バインディング Fault 要素定義要素	-	1
	name	SOAP バインディング名前属性	フォルト名を定義します。	1

( 凡例 )

- : 該当しません。

注

使用できる文字種には制限があります。使用できる文字種の詳細については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」の WSDL 定義の名称に関する内容を参照してください。

wsdl:binding 要素の定義例を次の図に示します。

図 8-10 wsdl:binding 要素の定義例



## 8. ユーザ定義受付の定義

style 属性および use 属性を使用する場合

wsdl:binding 要素以下に style 属性および use 属性を使用して、document または literal を明示することもできます。

図 8-11 wsdl:binding 要素の定義例 ( style 属性および use 属性を使用する場合 )

Sample\_Reception.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  :
  <wsdl:binding name="SampleBinding" type="svc:SamplePortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="operation1">
      <soap:operation soapAction="" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
      <wsdl:fault name="Fault1">
        <soap:fault name="Fault1" />
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
```

style="document" を明示

use="literal" を明示

### ( 8 ) wsdl:service 要素の定義

wsdl:service 要素とその下位要素、および属性を定義します。

1. (3) で定義した wsdl:definitions 要素の子要素として、wsdl:service 要素を定義します。
2. wsdl:service 要素に name 属性を定義します。  
name 属性の値には、任意の文字列を指定できます。
3. wsdl:service 要素の子要素として、wsdl:port 要素を一つ定義します。
4. wsdl:port 要素に name 属性と binding 属性を定義します。  
name 属性の値には、任意の文字列を指定できます。  
binding 属性の値には、(7) で定義した wsdl:binding 要素の name 属性の値を QName で指定します。
5. wsdl:port 要素の子要素として、soap:address 要素を一つ定義します。
6. soap:address 要素に location 属性を定義します。  
location 属性の値には、仮の値として「http://localhost:80/SampleService/services/ <PortName>」を指定します。  
<PortName> には、手順 4. で定義した name 属性と同じ値を指定してください。

## ポイント

WSDL を作成した段階では、サービスロケーションの値 (wsdl:port 要素内の soap-address 要素の location 属性値) は決定していません。このため、仮の値を設定します。この値は、Cosminexus が提供している SOAP 通信基盤でエラーにならない値です。

サービスロケーションの値は、ユーザ定義ファイルを含む HCSC コンポーネントを配備定義したあとに確定します。WSDL を編集して、確定したサービスロケーションの値を設定してから、WSDL を基にサービスリクエストを作成してください。WSDL の編集の詳細については、「10.6.2 WSDL を編集する」を参照してください。

wsdl:service 要素以下に定義する要素と属性を次の表に示します。

表 8-7 wsdl:service 要素以下に定義する要素と属性

要素	属性	意味	値	個数
wsdl:service	-	サービス定義要素	-	1
	name	サービス名属性	任意の文字列 を定義します。	1
wsdl:port	-	ポート定義要素	-	1
	name	ポート名属性	任意の文字列 を定義します。	1
	binding	バインディング参照属性	(7) で定義した wsdl:binding 要素の name 属性値を QName で指定します。	1
soap:address	-	SOAP バインディングアドレス定義要素	-	1
	location	サービス位置定義属性	仮の値として、 「http://localhost:80/ SampleService/services/ <PortName>」を定義します。 <PortName> には、wsdl:port 要素の name 属性と同じ値を指定します。	1

(凡例)

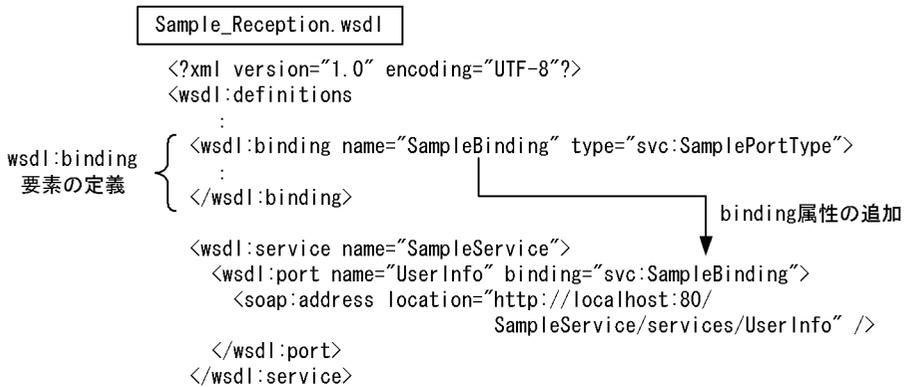
- : 該当しません。

注

使用できる文字種には制限があります。使用できる文字種の詳細については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」の WSDL 定義の名称に関する内容を参照してください。

wsdl:service 要素の定義例を次の図に示します。

図 8-12 wsdl:service 要素の定義例

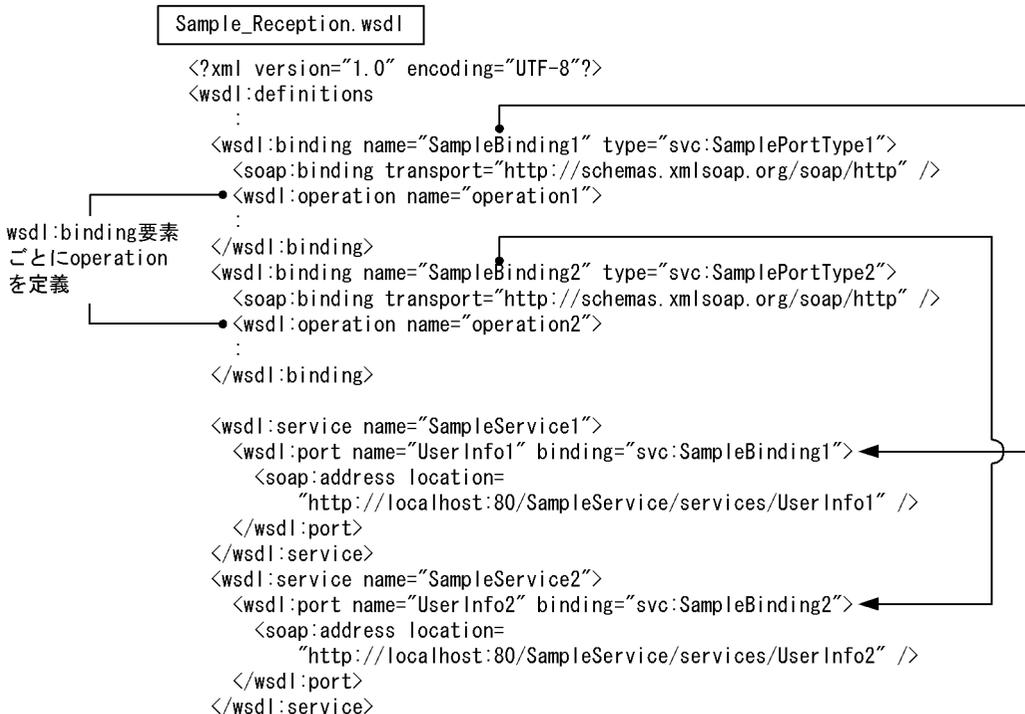


複数の wsdl:port を使用する場合

作成例 ( Sample\_Reception.wsdl ) では、一つの wsdl:port 要素にすべてを定義していますが、複数の wsdl:port 要素を使用して、それぞれにオペレーションを定義することもできます。

複数の wsdl:port 要素を使用する場合の定義例を次の図に示します。

図 8-13 wsdl:service 要素の定義例 ( 複数の wsdl:port を使用する場合 )



## 8.4 ユーザ定義受付の追加

ユーザ定義受付を追加します。ユーザ定義受付を新規に追加するには、ユーザ定義受付追加ウィザードを利用します。ユーザ定義受付追加ウィザードでは、追加するユーザ定義受付でどの WSDL ファイルを使うか、WSDL 内のどの wsdl:port 要素を利用するかを定義します。

### ポイント

追加するユーザ定義受付の定義内容は、ビジネスプロセスの受付アクティビティおよび応答アクティビティの定義に合わせる必要があります。次のどちらかの順で、ユーザ定義受付とビジネスプロセスの定義内容を合わせてください。

- ユーザ定義受付を追加する前に、ビジネスプロセスの受付アクティビティおよび応答アクティビティを定義して、受付アクティビティおよび応答アクティビティのオペレーションや電文の定義に合わせてユーザ定義受付を追加します。
- ユーザ定義受付を追加したあとに、ユーザ定義受付の内容に合わせてビジネスプロセスの受付アクティビティおよび応答アクティビティの定義を変更します。

### ! 注意事項

ユーザ定義受付は、新規に追加した時点で定義内容が確定します。追加したユーザ定義受付の定義内容は変更できません。変更する場合は、該当するユーザ定義受付を削除して、再度新規に追加してください。

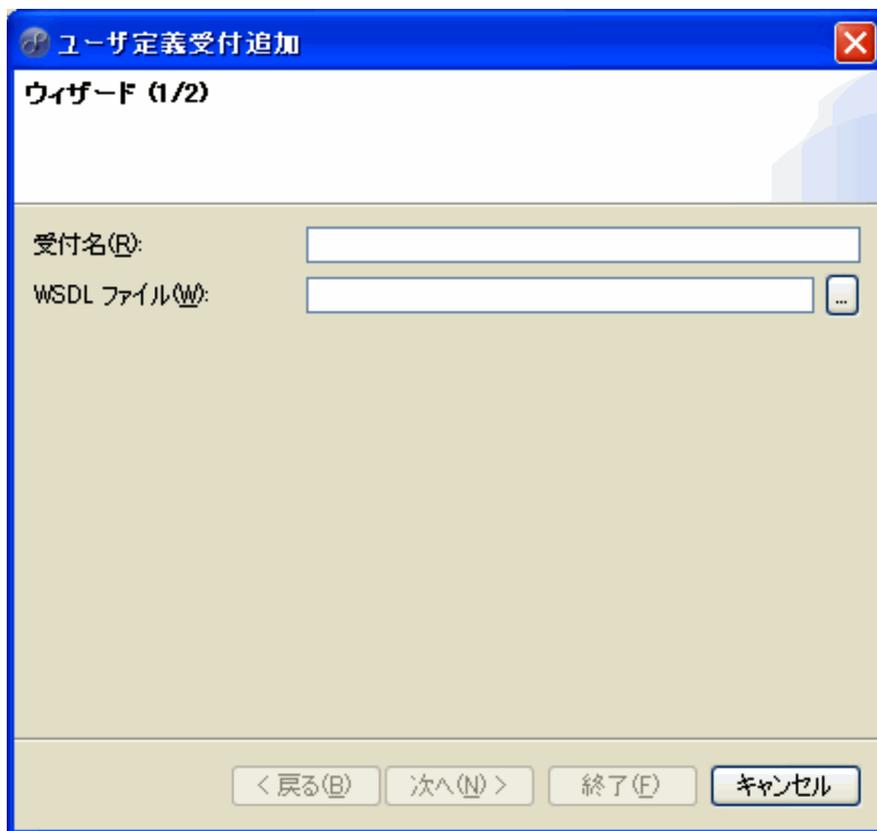
ユーザ定義受付の削除方法については、「8.7 ユーザ定義受付の削除」を参照してください。

### 8.4.1 ユーザ定義受付を新規に追加する

ユーザ定義受付を新規に追加する方法を次に示します。

1. ツリービューのサービス定義一覧で、ユーザ定義受付を追加するサービス（ビジネスプロセス）を選択して、右クリックします。  
サービス一覧のポップアップメニューが表示されます。
2. ポップアップメニューから [ユーザ定義受付追加] を選択します。  
ユーザ定義受付追加ウィザード（1/2）が表示されます。

図 8-14 ユーザ定義受付追加ウィザード ( 1/2 )



3. [ 受付名 ] と [ WSDL ファイル ] を指定します。

- 受付名

ユーザ定義受付の名称を XML スキーマの NCName 定義文字で指定します。1 ~ 40 バイトで指定します。受付名は、ビジネスプロセス内で一意である必要があります。

- WSDL ファイル

ユーザ定義受付で利用する WSDL ファイルを絶対パスで指定します。相対パス (例 : wsddir¥wsdlfile.wsdl) や UNC 形式 (例 : ¥¥mypc¥wsddir¥wsdlfile.wsdl) で指定しないでください。

[ ... ] ボタンをクリックして表示されるダイアログから WSDL ファイルを選択することもできます。

4. [ 次へ ] ボタンをクリックします。

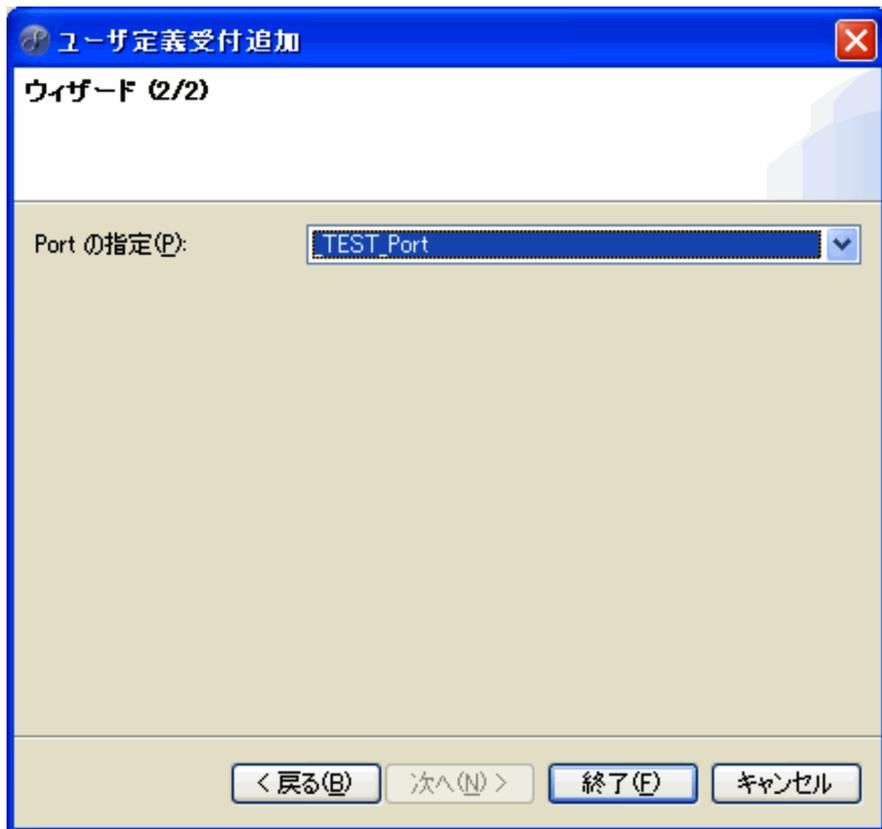
ユーザ定義受付追加ウィザード ( 2/2 ) が表示されます。

[ 次へ ] ボタンをクリックしたときに、「WSDL ファイルの解析に失敗しました。」というメッセージが表示された場合、WSDL ファイルにエラーがあります。

WSDL の形式については、「2.5.1(3) WSDL の形式」、およびマニュアル

「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。

図 8-15 ユーザ定義受付追加ウィザード (2/2)



5. [ Port の指定 ] のドロップダウンリストから、利用するポートを選択します。  
ドロップダウンリストには、手順 3. で指定した WSDL ファイルに定義されているポート名 (wsdl:port 要素の name 属性の値) が表示されます。
6. [ 終了 ] ボタンをクリックします。  
ユーザ定義受付がビジネスプロセスに追加され、ユーザ定義受付定義画面が表示されます。  
ユーザ定義受付定義画面の詳細については「11.2.5 ユーザ定義受付定義画面」を参照してください。

## 8.5 ユーザ定義受付の内容の確認

---

ユーザ定義受付を追加したあとには、定義された内容を確認してください。ユーザ定義受付の内容は、ユーザ定義受付定義画面で確認できます。ユーザ定義受付定義画面は、ユーザ定義受付追加ウィザードでユーザ定義受付を追加したあとに表示されます。

ユーザ定義受付定義画面を任意のタイミングで表示したい場合は、次の手順で表示させてください。

1. ツリービューのサービス定義一覧で、ユーザ定義受付を含むビジネスプロセスをダブルクリックします。  
ビジネスプロセス定義画面が表示されます。
2. ビジネスプロセス定義画面のキャンパスの下部にあるユーザ定義受付の受付名が表示されたタブをクリックします。  
ユーザ定義受付定義画面に画面の表示が切り替わります。

ユーザ定義受付定義画面の詳細については「11.2.5 ユーザ定義受付定義画面」を参照してください。

ユーザ定義受付定義画面では、ユーザ定義受付の定義内容を確認して、受付アクティビティおよび応答アクティビティの定義と不整合になっていないか確認してください。不整合になっている場合、受付アクティビティまたは応答アクティビティの定義を変更してください。

### 注意事項

ユーザ定義受付の定義内容は変更できません。ユーザ定義受付の内容を変更したい場合は、該当するユーザ定義受付を削除して、再度新規に追加してください。

---

## 8.6 ユーザ定義受付の検証

uCosminexus Service Architect では、作成したユーザ定義受付の内容が妥当かどうか検証できます。

### 8.6.1 検証内容

ユーザ定義受付の検証での検証内容と対処を次の表に示します。

表 8-8 ユーザ定義受付の検証内容と対処

項番	検証内容	エラーになった場合の対処
1	ユーザ定義受付のオペレーション名と一致するオペレーション名がビジネスプロセス内に定義されているか。	ビジネスプロセスの受付アクティビティおよびユーザ定義受付のオペレーション名を見直し、一致するオペレーション名を定義します。
2	ユーザ定義受付のオペレーションの通信モデル（同期 / 非同期）と、ビジネスプロセスのオペレーションの通信モデル（同期 / 非同期）が一致しているか。	ビジネスプロセスの受付アクティビティおよびユーザ定義受付の定義を見直し、オペレーションの通信モデル（同期 / 非同期）を合わせます。
3	ユーザ定義受付に対応する受付アクティビティ、および応答アクティビティ（フォルト電文用を含む）の割当変数の種別がメッセージ型か。	ビジネスプロセスの定義を見直し、該当するアクティビティの割当変数をユーザ定義受付に対応する変数に変更します。
4	ユーザ定義受付に対応する応答電文用（フォルト電文用は含まない）の応答アクティビティが存在するか。	ビジネスプロセスの定義を見直し、ユーザ定義受付のオペレーションに対応する応答アクティビティを定義します。
5	ユーザ定義受付のオペレーションのフォルト名と一致するフォルト名がビジネスプロセスの応答アクティビティに存在するか。	ビジネスプロセスおよびユーザ定義受付の定義を見直し、次のどちらかの対処をします。 <ul style="list-style-type: none"> <li>ユーザ定義受付のオペレーションのフォルト名に対応する応答アクティビティを定義します。</li> <li>ユーザ定義受付のオペレーションから不要なフォルトの定義を削除します。この場合、一度ユーザ定義受付を削除し、フォルトの定義を削除した WSDL ファイルからもう一度ユーザ定義受付を作成します。</li> </ul>
6	ユーザ定義受付のオペレーションに存在しないフォルト名がビジネスプロセスの応答アクティビティに定義されていないか。	ビジネスプロセスおよびユーザ定義受付の定義を見直し、次のどちらかの対処をします。 <ul style="list-style-type: none"> <li>ユーザ定義受付のオペレーションで定義されていないフォルトに対応する応答アクティビティを削除します。</li> <li>ユーザ定義受付のオペレーションにフォルトの定義を追加します。この場合、一度ユーザ定義受付を削除し、フォルトの定義を追加した WSDL ファイルからもう一度ユーザ定義受付を作成します。</li> </ul>

## 8.6.2 検証方法

ユーザ定義受付の検証は、ビジネスプロセスを検証するときにあわせて実施されます。ビジネスプロセスを検証すると、検証対象のビジネスプロセス内のユーザ定義受付が検証されます。

ビジネスプロセスの検証方法の詳細については、「6.9.2 検証方法」を参照してください。

また、ユーザ定義受付の検証結果は、ビジネスプロセスの検証結果と同様に、コンソールビューに表示されます。検証結果の表示の詳細については、「6.9.3 検証内容の表示」を参照してください。

## 8.7 ユーザ定義受付の削除

---

ユーザ定義受付を削除します。なお、次のユーザ定義受付は削除できません。

- すでに公開されているビジネスプロセスが持つユーザ定義受付
- 以前のバージョンのビジネスプロセスで作成したユーザ定義受付

ユーザ定義受付の削除方法について、次に示します。

1. ツリービューのサービス定義一覧で、ユーザ定義受付を削除するビジネスプロセスを選択して、右クリックします。  
サービス一覧のポップアップメニューが表示されます。
2. ポップアップメニューから「ユーザ定義受付削除」を選択します。  
[ユーザ定義受付削除]ダイアログが表示されます。[ユーザ定義受付削除]ダイアログの入力・表示内容の詳細については、「11.6.1 ユーザ定義受付削除ダイアログ」を参照してください。
3. 削除するユーザ定義受付のチェックボックスをチェックします。
4. [OK] ボタンをクリックします。  
指定したユーザ定義受付がビジネスプロセスから削除されます。

### ! 注意事項

ビジネスプロセスを削除すると、削除したビジネスプロセスに含まれるユーザ定義受付も削除されます。

---



# 9

## HCSC コンポーネントのパッケージングと配備定義

この章では、HCSC コンポーネントのパッケージングと配備定義について説明します。

---

9.1 パッケージングと配備定義

---

9.2 パッケージング

---

9.3 HCSC コンポーネントの配備定義

---

9.4 HCSC コンポーネントの情報の参照

---

## 9.1 パッケージングと配備定義

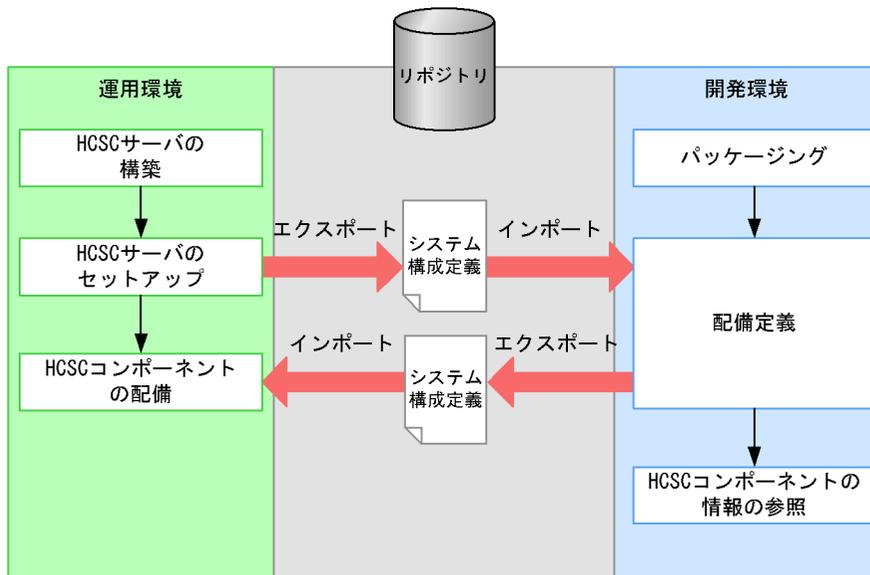
定義した HCSC コンポーネントは、パッケージングしたり、どのクラスタ（または単一の HCSC サーバ）に配備するか決めたり（配備定義）します。

ユーザは、配備定義をする前にリポジトリをインポートする必要があります。また、配備定義したあとにリポジトリをエクスポートして、運用環境に配備定義したシステム構成定義を提供します。システム構成定義の内容に従って、運用環境から実行環境のクラスタ（または単一の HCSC サーバ）に HCSC コンポーネントを配備します。

開発環境では、配備定義をしたあとに、配備定義をした HCSC コンポーネントの情報を参照できます。

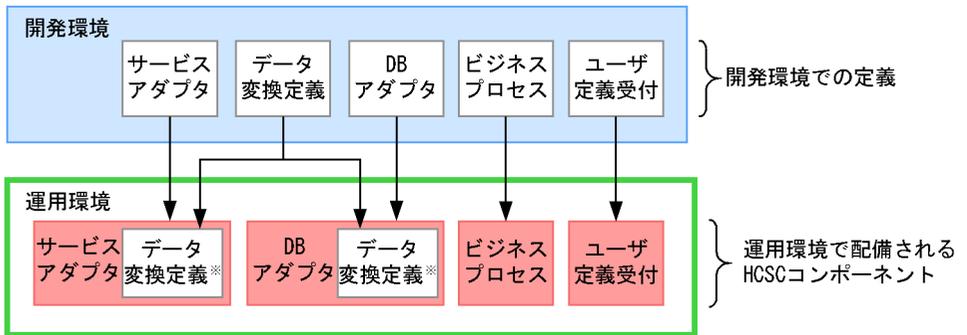
パッケージングから配備定義までの流れを次の図に示します。

図 9-1 パッケージングから配備定義までの流れ



なお、開発環境で定義した内容は、運用環境で HCSC コンポーネントとして配備されません。開発環境での定義と運用環境で配備される HCSC コンポーネントの関係を次の図に示します。

図 9-2 開発環境での定義と運用環境で配備される HCSC コンポーネントの関係



(凡例)

: HCSCコンポーネント

注※

データ変換定義は、HCSCコンポーネントではありません。

運用環境では、サービスアダプタおよびDBアダプタに含まれる形になるため、運用環境からはデータ変換定義が見えません。

### ! 注意事項

リポジトリの一部の情報だけをインポートした場合には、インポートしたリポジトリ情報に含まれる HCSC コンポーネントを、必ずパッケージングする必要があります。

リポジトリの一部の情報だけのインポートについては、「3.2.3 リポジトリをインポートする」を参照してください。

## 9.2 パッケージング

---

実行環境に配備するために必要な HCSC コンポーネントに関するファイルを組み立てて、EAR ファイルを作成します。これをパッケージングと呼びます。ここでは、アダプタまたはビジネスプロセスをまとめて HCSC コンポーネントと呼びます。

HCSC コンポーネントのパッケージングに伴い、パッケージング対象の HCSC コンポーネントと関連するデータ変換定義およびユーザ定義受付もパッケージングされます。

パッケージングを実行すると、HCSC コンポーネントが検証されます。よって、HCSC コンポーネントが正しく定義されている場合はパッケージングできますが、正しく定義されていない場合はパッケージング時の検証でエラーが発生します。パッケージングの前に、あらかじめ、HCSC コンポーネントを検証しておく、円滑にパッケージングできます。

パッケージングの方法を次に示します。

1. ツリービューのサービス定義一覧で、パッケージングする HCSC コンポーネントを右クリックして、「パッケージング」を選択します。  
メッセージダイアログが表示されます。
2. 次のどちらかの操作をします。
  - パッケージングが成功した場合  
[OK] ボタンをクリックします。  
すでに EAR ファイルがある場合、上書きを確認するダイアログが表示されます。  
上書きしてよいときは、[はい] ボタンをクリックします。
  - パッケージングが失敗した場合  
ダイアログのメッセージに従って対処し、パッケージングを再実行します。

なお、パッケージングする前にアダプタおよびビジネスプロセスが保存されていないときは、リソース保管ダイアログが表示されます。リソース保管ダイアログでは、保存されていないアダプタおよびビジネスプロセスの定義を保存できます。

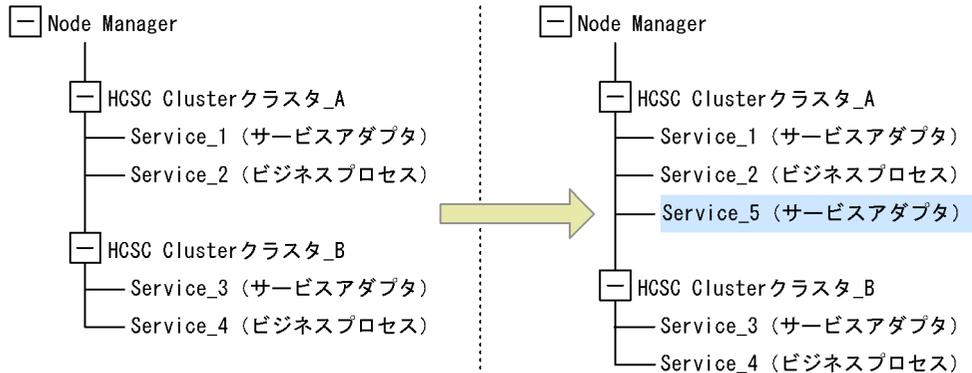
## 9.3 HCSC コンポーネントの配備定義

配備定義とは、定義した HCSC コンポーネントをどのクラスタ（または単一の HCSC サーバ）に配備するか決めることです。

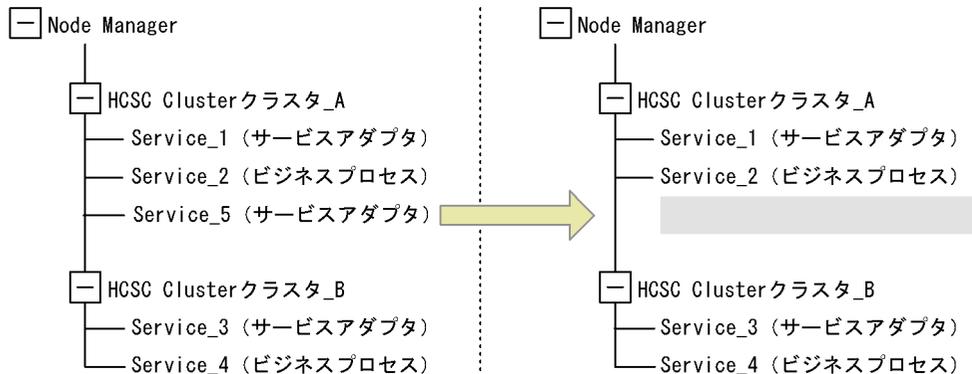
ツリー形式で表示されるツリービューのシステム構成定義一覧で、クラスタ（または単一の HCSC サーバ）の下位に表示される HCSC コンポーネントは、そのクラスタ（または単一の HCSC サーバ）に配備するよう定義されていることを表します。ツリービューのシステム構成定義一覧のクラスタ（または単一の HCSC サーバ）の下位に、配備する HCSC コンポーネントを追加・削除して配備定義を行います。HCSC コンポーネントの追加と削除の例を次の図に示します。

図 9-3 HCSC コンポーネントの追加と削除

- HCSC Clusterクラスタ\_AIにService\_5を追加する場合



- HCSC Clusterクラスタ\_AからService\_5を削除する場合



開発環境で配備定義し、更新されたシステム構成定義の内容に従って、運用環境でクラスタ（または単一の HCSC サーバ）へ HCSC コンポーネントが配備されます。クラスタ（または単一の HCSC サーバ）への配備方法の詳細については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」のサービスアダ

## 9. HCSC コンポーネントのパッケージングと配備定義

プタの配備およびビジネスプロセスの配備に関する内容を参照してください。

ここではクラスタ（または単一の HCSC サーバ）に HCSC コンポーネントを追加・削除して配備定義する方法を説明します。

なお、システム構成定義で定義されているサーバ構成とサービスの組み合わせによっては、配備定義できない場合があります。システム構成定義で定義されているサーバ構成とサービスの組み合わせによる配備定義の可否を次の表に示します。

表 9-1 配備定義の可否

項番	システム構成定義で定義されているサーバ構成		サービスの種別	配備定義の可否
	データベースの使用有無	Cosminexus RM の使用有無		
1	使用する	使用する	すべて	
2	使用しない	使用しない	ビジネスプロセス（永続化する）	×
3			ビジネスプロセス（永続化しない）	
4			Web サービス	
5			SessionBean	
6			MDB（WS-R）	×
7			MDB（DB キュー）	×
8			カスタムアダプタ	
9			使用する	使用しない
10	ビジネスプロセス（永続化しない）			
11	Web サービス			
12	SessionBean			
13	MDB（WS-R）	×		
14	MDB（DB キュー）	×		
15	カスタムアダプタ			

（凡例）

：配備定義できます。

×：配備定義できません。

### 9.3.1 配備できるクラスタ（または単一の HCSC サーバ）

配備できるクラスタ（または単一の HCSC サーバ）は、ツリービューのシステム構成定義一覧に表示されます。表示される情報は、運用環境でセットアップしたクラスタ（または単一の HCSC サーバ）の構成情報です。この情報は、運用環境でエクスポートしたリポジトリ情報を、開発環境でインポートして取得します。

リポジトリのインポートについては、「3.2.3 リポジトリをインポートする」を参照してください。

### 9.3.2 クラスタに HCSC コンポーネントを追加する

クラスタ（または単一の HCSC サーバ）に HCSC コンポーネントを追加する方法を次に示します。なお、HCSC コンポーネントをクラスタ（または単一の HCSC サーバ）に追加すると、追加した HCSC コンポーネントと関連するデータ変換定義およびユーザ定義受付もあわせて追加されます。

1. Eclipse のメニューから、[ HCSC-Definer ] - [ システム構成定義一覧 ] を選択します。  
ツリービューのシステム構成定義一覧に、クラスタ（または単一の HCSC サーバ）の一覧が表示されます。
2. クラスタ（または単一の HCSC サーバ）の一覧から HCSC コンポーネントを追加するクラスタ（または単一の HCSC サーバ）を選択し、ダブルクリックします。  
クラスタの情報が表示されます。

種別	HCSC サーバ
サーバ名	MyHCSC
DB の使用	使用する
RM の使用	使用する

3. 手順 2. で選択したクラスタ（または単一の HCSC サーバ）を右クリックして、[ サービス追加 ] を選択します。  
[ サービス追加 ] ダイアログが表示されます。  
定義済みの HCSC コンポーネントを、パッケージングしたあとに編集した場合、その HCSC コンポーネントを追加対象にしてよいか確認するダイアログが表示されます。追加対象にしてよいときは、[ はい ] ボタンをクリックします。
4. 追加する HCSC コンポーネントを [ 追加するサービス ] のドロップダウンリストから選択します。
5. [ OK ] ボタンをクリックします。  
追加した HCSC コンポーネント情報が表示されます。

サービス名	BP1
サービス種別	ビジネスプロセス

### 9.3.3 クラスタから HCSC コンポーネントを削除する

クラスタ（または単一の HCSC サーバ）から HCSC コンポーネントを削除する方法を次に示します。なお、HCSC コンポーネントをクラスタ（または単一の HCSC サーバ）から削除すると、追加した HCSC コンポーネントと関連するデータ変換定義およびユーザ定義受付もあわせて削除されます。

1. ツリービューのシステム構成定義一覧から削除する HCSC コンポーネントを選択します。
2. 次のどちらかの方法で削除します。
  - HCSC コンポーネントを右クリックして、[ サービス削除 ] を選択します。
  - [ Delete ] キーを押します。

削除を確認するダイアログが表示されます。削除してよいときは、[ はい ] ボタンをクリックします。

## 9.4 HCSC コンポーネントの情報の参照

---

配備定義したあとに、配備定義をした HCSC コンポーネントの情報を参照できます。HCSC コンポーネントの情報は、HCSC コンポーネント情報表示画面に表示されます。HCSC コンポーネント情報表示画面の詳細については、「11.2.6 HCSC コンポーネント情報表示画面」を参照してください。

また、HCSC コンポーネントに含まれるユーザ定義受付の情報も参照できます。ユーザ定義受付の情報は、ユーザ定義受付情報表示画面に表示されます。ユーザ定義受付情報表示画面の詳細については、「11.2.7 ユーザ定義受付情報表示画面」を参照してください。

### 9.4.1 参照できる HCSC コンポーネントの情報

参照できる HCSC コンポーネントの情報を次に示します。

#### (1) サービス部品の情報

サービス部品の情報として、次の情報を参照できます。

- インターフェース情報  
サービス名 (HCSC コンポーネント名) とオペレーションの組み合わせに対応した、オペレーション情報、要求電文、応答電文、およびフォルト電文の情報を参照できます。
- サーバ情報  
サービス部品を呼び出すことができるクラスタ (または単一の HCSC サーバ) のサービス種別ごとの宛先を参照できます。
- オペレーション情報  
インターフェース情報で指定したオペレーションの名称、および通信モデルを参照できます。
- 要求電文  
標準電文フォーマット、および電文フォーマット ID を参照できます。
- 応答電文  
標準電文フォーマット、および電文フォーマット ID を参照できます。
- フォルト電文  
フォルト名、および電文フォーマットを参照できます。

#### (2) ユーザ定義受付の情報

ユーザ定義受付の情報として、次の情報を参照できます。

- ユーザ定義受付情報  
ユーザ定義受付の受付名、ポート名、および URL 情報を参照できます。

## 9.4.2 HCSC コンポーネントの情報を表示する

HCSC コンポーネントの情報を表示させる方法を次に示します。

1. Eclipse のメニューから、[ HCSC-Definer ] - [ 公開サービス一覧 ] を選択します。  
ツリービューの公開サービス一覧に、HCSC コンポーネントの一覧が表示されます。
2. HCSC コンポーネントの一覧から、情報を表示する HCSC コンポーネントを選択し、ダブルクリックします。  
HCSC コンポーネント情報表示画面に、HCSC コンポーネントの情報が表示されます。
3. 次に示す操作をします。

[ オペレーション情報 ], [ 要求電文 ], [ 応答電文 ], および [ フォルト電文 ] の情報を参照する場合

[ インタフェース情報 ] の [ オペレーション ] のドロップダウンリストから、情報を参照するオペレーションを選択します。

電文の内容を参照するときは、参照する電文（要求電文、応答電文、またはフォルト電文）の [ 表示 ] ボタンをクリックします。

クラスタ（または単一の HCSC サーバ）のサービス種別ごとの宛先を参照する場合

[ サーバ情報 ] の [ サーバ名 ] に宛先を参照するクラスタ名が表示されます。

Web サービスで使用する WSDL ファイルを取得するときは、[ WSDL 取得 ] ボタンをクリックします。

SessionBean で使用する RMI-IIOP スタブファイルを取得するときは、[ スタブ取得 ] ボタンをクリックします。

ユーザ定義受付の情報を参照する場合

HCSC コンポーネント情報表示画面の下部の [ ユーザ定義受付情報 ] タブをクリックして、ユーザ定義受付情報表示画面を表示させます。

## 9.4.3 HCSC コンポーネントの一覧を更新する

HCSC コンポーネントが変更された場合、リポジトリ内の情報とツリービューのシステム構成定義一覧に表示されている情報が不一致になる場合があります。このような場合、ツリービューのシステム構成定義一覧の表示内容を最新の情報に更新できます。

更新方法を次に示します。

1. ツリービューのシステム構成定義一覧で、表示されている HCSC コンポーネントを選択しないで右クリックして、[ 更新 ] を選択します。  
ツリービューのシステム構成定義一覧に表示されている HCSC コンポーネントの一覧が更新されます。

**!** 注意事項

ツリービューのシステム構成定義一覧に表示されるクラスタ（または単一の HCSC サーバ）の構成情報は、運用環境でセットアップしたクラスタ（または単一の HCSC サーバ）の構成情報です。

ツリービューのシステム構成定義一覧に最新のサーバ構成一覧を表示するには、必要に応じて、運用環境でエクスポートしたリポジトリ情報を開発環境でインポートします。インポートすると、自動的に表示が更新されます。

リポジトリのインポートについては、「3.2.3 リポジトリをインポートする」を参照してください。

---



# 10 サービスリクエストの作成

この章では、HCSC サーバの標準受付およびユーザ定義受付にサービス部品を要求するサービスリクエストの作成について説明します。

---

10.1 サービスリクエスト作成の概要

---

10.2 標準の同期受付 (Web サービス) に要求を出すサービスリクエスト

---

10.3 標準の同期受付 (SessionBean) に要求を出すサービスリクエスト

---

10.4 標準の非同期受付 (MDB (WS-R)) に要求を出すサービスリクエスト

---

10.5 標準の非同期受付 (DB キュー) に要求を出すサービスリクエスト

---

10.6 ユーザ定義受付 (Web サービス) に要求を出すサービスリクエスト

---

10.7 DB アダプタに要求を出すサービスリクエスト

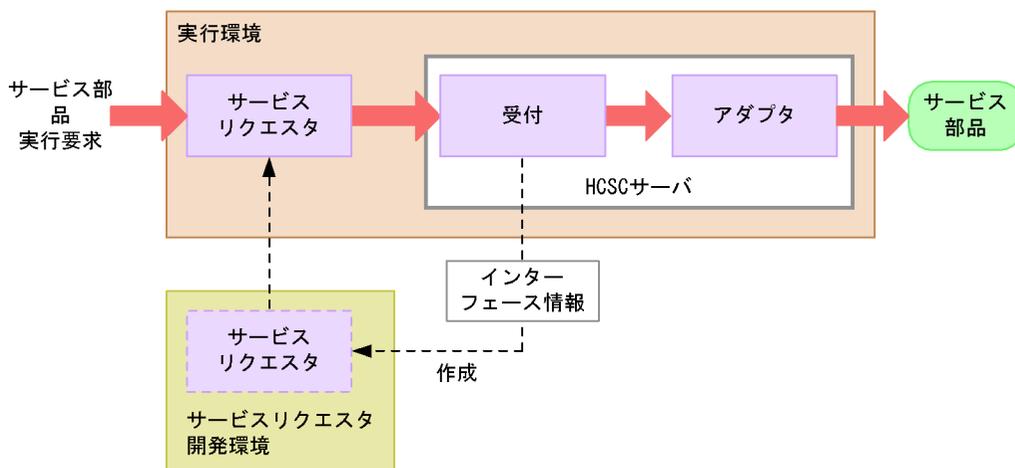
---

## 10.1 サービスリクエスト作成の概要

サービスリクエストとは、業務担当者からの要求を受け付けて、アダプタやビジネスプロセスなどの各 HCSC コンポーネントにサービス部品の実行要求を送信するアプリケーションです。サービスリクエストは、実行環境が提供するインターフェース情報を基に、開発環境で作成し、実行環境に配備して利用します。サービスリクエストの開発は、Java のプログラムを開発できる環境で行ってください。

サービスリクエストの概要を次の図に示します。

図 10-1 サービスリクエストの概要



サービスリクエストは、HCSC サーバの標準受付およびユーザ定義受付にサービス部品の実行を要求します。なお、サービスリクエストが使用するプロトコルによって、要求先は異なります。サービスリクエストが使用するプロトコルと標準受付およびユーザ定義受付の種類を次の表に示します。

表 10-1 サービスリクエストが使用するプロトコルと標準受付およびユーザ定義受付の種類

	受付	プロトコル
標準受付	同期受付 (Web サービス)	SOAP (HTTP)
	同期受付 (SessionBean)	RMI-IIOP
	非同期受付 (MDB (WS-R))	WS-R
	非同期受付 (DB キュー)	<ul style="list-style-type: none"> <li>Cosminexus RM 独自のプロトコル</li> <li>JMS</li> </ul>
ユーザ定義受付	同期受付 (Web サービス)	SOAP (HTTP)

**!** 注意事項

運用時の処理の流れを考慮して、サービスリクエストを開発する必要があります。また、障害発生時にすぐに対策できるように、よく理解した上でエラー処理を作成する必要があります。詳細については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の HCSC サーバのサービス呼び出し時の運用、および障害対策の内容を参照してください。

---

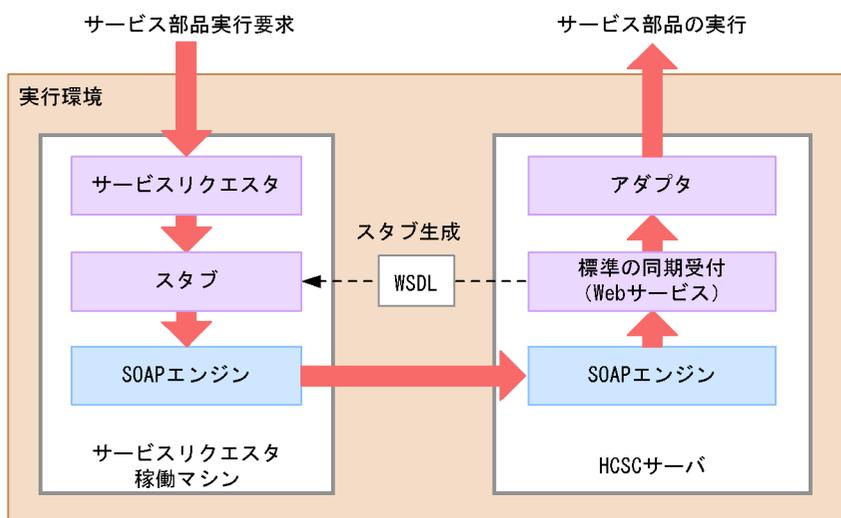
## 10.2 標準の同期受付（Web サービス）に要求を出すサービスリクエスト

標準の同期受付（Web サービス）へ要求電文を送信するサービスリクエストは、SOAPを利用して標準受付と通信します。サービスリクエストから標準受付へサービス部品実行の要求電文が送信され、HCSCサーバがサービス部品の実行を行います。

SOAPを利用する場合、同期受付（Web サービス）のインターフェース情報をWSDLから取得します。取得したWSDLからスタブを生成し、そのスタブを利用して、同期受付（Web サービス）に要求を出します。したがって、サービスリクエストは、生成したスタブを利用できるように実装する必要があります。

標準の同期受付（Web サービス）に要求を出すサービスリクエストとHCSCサーバの関係を次の図に示します。

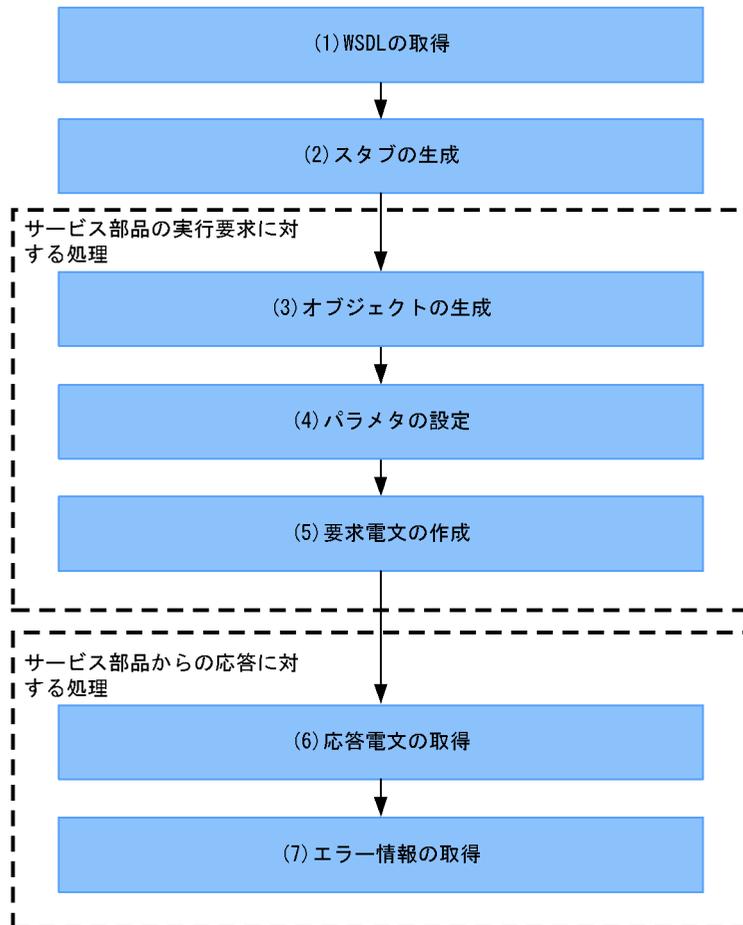
図 10-2 標準の同期受付（Web サービス）に要求を出すサービスリクエストとHCSCサーバの関係



### 10.2.1 サービスリクエスト（標準の同期受付（Web サービス））の作成手順

標準の同期受付（Web サービス）にサービス部品の実行を要求し、サービス部品を呼び出すためのサービスリクエストの作成手順を次に示します。

図 10-3 サービスリクエストの作成手順（標準の同期受付（Web サービス））



各工程の作業を次に示します。

### (1) WSDL の取得

サービス部品を呼び出す HCSC サーバの同期受付（Web サービス）のインターフェース情報を WSDL から取得します。WSDL の取得については、「10.2.2 WSDL を取得する」を参照してください。

### (2) スタブの生成

(1) で取得した WSDL からスタブを生成します。スタブの生成については、「10.2.3 スタブを生成する」を参照してください。

### (3) オブジェクトの生成

同期受付（Web サービス）のメソッドを呼び出すために、(2) で生成したスタブからオブジェクトを生成します。オブジェクトの生成については、「10.2.4 オブジェクトを生成

## 10. サービスリクエストの作成

する」を参照してください。

### (4) パラメタの設定

同期受付 (Web サービス) のメソッドの引数となるパラメタを設定します。パラメタの設定については、「10.2.5 パラメタを設定する」を参照してください。

### (5) 要求電文の作成

サービス部品の実行を要求する要求電文を作成します。要求電文の作成については、「10.2.6 要求電文を作成する」を参照してください。

### (6) 応答電文の取得

同期受付 (Web サービス) から、サービス部品の実行要求に対する応答電文を取得します。応答電文の取得については、「10.2.7 応答電文を取得する」を参照してください。

### (7) エラー情報の取得

要求先のサービス部品、HCSC サーバ、および SOAP エンジンでエラーが発生した場合、エラー情報を取得し、その情報に従って対処します。エラー情報の取得については、「10.2.8 エラー情報を取得する」を参照してください。

## 10.2.2 WSDL を取得する

サービス部品を呼び出す HCSC サーバの同期受付 (Web サービス) のインターフェース情報を WSDL から取得します。WSDL を取得するには、次の二つの方法があります。

### 方法 1

1. Eclipse のメニューから、[ HCSC-Definer ] - [ 公開サービス一覧 ] を選択します。  
ツリービューの公開サービス一覧に、HCSC コンポーネントの一覧が表示されます。
2. HCSC コンポーネントの一覧から、呼び出すサービス部品 (HCSC コンポーネント) を選択し、ダブルクリックします。  
HCSC コンポーネント情報表示画面に、HCSC コンポーネントの情報が表示されます。HCSC コンポーネント情報表示画面の詳細については、「11.2.6 HCSC コンポーネント情報表示画面」を参照してください。  
[ サーバ情報 ] には、サービス部品の要求先となる HCSC サーバの情報が表示されません。
3. [ WSDL 取得 ] ボタンをクリックします。  
[ WSDL ファイルの保存 ] ダイアログが表示されます。  
保存先を指定して、WSDL を取得します。

**!** 注意事項

方法 1 では、document-literal 型の WSDL は取得できません。document-literal 型の WSDL を取得するには方法 2 を実施してください。

## 方法 2

1. HCSC-Messaging をインストールしたディレクトリから、サンプルとして提供されている WSDL を取得します。

HCSC-Messaging をインストールしたディレクトリには、rpc-literal 型の cscmsg\_ws.wsdl ファイルと document-literal 型の cscmsg\_ws\_doc.wsdl ファイルが WSDL のサンプルとして提供されています。用途に合わせてどちらかの WSDL を取得してください。

HCSC-Messaging をインストールしたディレクトリについては、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の uCosminexus Service Platform のインストールに関する内容を参照してください。

2. WSDL の <soap:address> の location の内容を次のように変更します。

```
http://<ホスト名>:<ポート番号> 1/2<コンテキストルート>
      /services/CSCMsgSyncServiceDeliveryWSImpl
```

## 注 1

HCSC サーバの URL です。

## 注 2

開発環境および運用環境で HCSC サーバ構成配備時に決定される HCSC サーバのクラスタ名です。

方法 2 では WSDL の内容を変更する必要がありますが、方法 1 では定義されている正確な WSDL の情報を取得できます。rpc-literal 型の WSDL を取得する場合、できる限り方法 1 の手順を実施してください。

### 10.2.3 スタブを生成する

取得した WSDL からスタブを生成します。スタブは、Cosminexus が開発支援コマンドとして提供する WSDL2Java コマンドで生成します。

コマンドの入力例（WSDL のスタイルが rpc-literal の場合）を次に示します。

```
WSDL2Java cscmsg_ws.wsdl
```

## 10. サービスリクエストの作成

WSDL のスタイルが document-literal の場合、上記の例の「cscmsg\_ws.wsdl」の部分には「cscmsg\_ws\_doc.wsdl」と設定します。

WSDL2Java コマンドのオプションなどの詳細については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。

このコマンドを実行すると、次のディレクトリおよびファイルが作成されます。

```
/カレントディレクトリ
  jp
    co
      Hitachi
        soft
          csc
            msg
              message
                reception
                  ejb
                    CSCMsgSyncServiceDeliveryWSImpl.java 1
                    CSCMsgSyncServiceDeliveryWSImplService.java 2
                    CSCMsgSyncServiceDeliveryWSImplServiceLocator.java 3
                    CSCMsgSyncServiceDeliveryWSImplSoapBindingStub.java 4
                    CSCMsgServerFaultException.java 4
```

注 1  
ユーザ定義のデータクラス（サービスリクエストのインターフェースクラス）です。

注 2  
サービス部品のインターフェースクラスです。

注 3  
サービスクラス（サービス部品のインターフェースクラス）です。サービス部品への接続情報を保持します。

注 4  
スタブクラスです。

### 10.2.4 オブジェクトを生成する

同期受付（Web サービス）のメソッドを呼び出すために、生成したスタブを使用して、オブジェクトを生成します。

#### （1）使用するスタブ

次の二つのスタブを使用して、オブジェクトを生成します。

CSCMsgSyncServiceDeliveryWSImplServiceLocator.java クラス

このクラスは、サービス部品への接続先（endpoint）情報の参照および設定をします。このクラスでは次のメソッドを提供します。

表 10-2 CSCMsgSyncServiceDeliveryWSImplServiceLocator.java クラスのメソッド

メソッド名	機能説明
getCSCMsgSyncServiceDeliveryWSImplAddress()	サービス部品への接続先情報を返します。 戻り値： java.lang.String
getCSCMsgSyncServiceDeliveryWSImpl()	サービスクラスへのインターフェースクラスのオブジェクトポインタを返します。 戻り値： インターフェースクラスオブジェクト (CSCMsgSyncServiceDeliveryWSImpl オブジェクト)
getCSCMsgSyncServiceDeliveryWSImpl(java.net.URL portAddress)	設定されたサービス部品への接続先情報を使用して、サービスクラスへのオブジェクトポインタを返します。 戻り値： インターフェースクラスオブジェクト (CSCMsgSyncServiceDeliveryWSImpl オブジェクト)

#### CSCMsgSyncServiceDeliveryWSImpl.java クラス

サービスクラスとして利用できるメソッドの一覧が記述されています。このクラスを使用して SOAP サービスを利用します。

#### (2) オブジェクトの生成手順

同期受付 (Web サービス) のメソッドを呼び出すためのオブジェクトを生成する手順を次に示します。

##### 1. サービス部品のインターフェースクラスである

CSCMsgSyncServiceDeliveryWSImplServiceLocator クラスのオブジェクトを生成します。

(例)

```
CSCMsgSyncServiceDeliveryWSImplServiceLocator locator
    = new CSCMsgSyncServiceDeliveryWSImplServiceLocator();
```

##### 2. サービス部品のインターフェースクラスのオブジェクトを使用して、サービスリクエストのインターフェースクラスである CSCMsgSyncServiceDeliveryWSImpl.java クラスのオブジェクトを生成します。

なお、生成・取得したサービスリクエストのインターフェースクラスのインスタンスは、複数のスレッドで共有できません。

## 10. サービスリクエストの作成

(例)

```
CSCMsgSyncServiceDeliveryWSImpl ws = null;
try {
    ws = locator.getCSCMsgSyncServiceDeliveryWSImpl();
} catch (ServiceException e) {
    e.printStackTrace();
    return;
}
```

サービス部品への接続先は、WSDL 定義内の Service 要素の address 子要素である location 属性となります。サービスリクエストのプログラム内で、サービス部品への接続先情報を取得する場合は、次のように記述します。

(例)

```
String url = locator.getCSCMsgSyncServiceDeliveryWSImplAddress();
```

また、サービスリクエストのプログラム内で、サービス部品への接続先を変更する場合は、次のように記述します。

(例)

```
java.net.URL endpoint
= new java.net.URL("http://hostname:80/<コンテキストルート>
                  /services/CSCMsgSyncServiceDeliveryWSImpl");
CSCMsgSyncServiceDeliveryWSImpl locator
= locator.getCSCMsgSyncServiceDeliveryWSImpl(endpoint);
```

3. 生成した CSCMsgSyncServiceDeliveryWSImpl.java クラスのオブジェクトを使用して、同期受付 (Web サービス) のメソッドを呼び出します。

(例) 要求電文が XML の場合

```
String result = ws.invokeXML(           // メソッドの呼び出し
    serviceName,                       // サービス名
    clientID,                           // クライアント相関ID
    requestFormatID,                   // リクエストフォーマットID
    responseFormatID,                 // レスポンスフォーマットID
    operationName,                     // オペレーション名
    userData);                          // ユーザ電文
```

## (例) 要求電文がバイナリの場合

```
byte[] resultBinary = ws.invokeBinary( // メソッドの呼び出し
    serviceName, // サービス名
    clientId, // クライアント関連ID
    requestFormatID, // リクエストフォーマットID
    responseFormatID, // レスポンスフォーマットID
    operationName, // オペレーション名
    userDataBinary.length, // ユーザ電文長
    userDataBinary); // ユーザ電文
```

**!** 注意事項

バイナリ電文の要求電文を送信できるのは、サービス部品側で使用している電文形式がバイナリ形式のときだけです。

## 10.2.5 パラメタを設定する

同期受付 (Web サービス) のメソッドを呼び出すため、メソッドの引数となるパラメタを設定します。パラメタの詳細を次の表に示します。

表 10-3 パラメタの詳細 (標準の同期受付 (Web サービス))

パラメタ名	データ型	パラメタ		説明
		invokeXML	invokeBinary	
サービス名 (serviceName)	java.lang.String	in0 (type="xsd:string")		要求先のサービス名です。 このパラメタは必ず指定してください。 要求先のサービス名は、開発環境で定義したアダプタまたはビジネスプロセスを指定してください。
クライアント関連ID (clientId)	java.lang.String	in1 (type="xsd:string")		サービスリクエストからの要求電文を一意に識別するための関連識別子です。 255文字以内の英数字、アンダーライン(_), ピリオド(.)およびハイフン(-)で指定してください。 サービスリクエストからの要求電文と、HCSCサーバで管理している実行履歴やログ・トレースを対応づけるために使用します。したがって、HCSCサーバに送信した要求電文ごとに異なるIDを指定してください。 クライアント関連IDを省略したい場合、NULLを指定してください。
リクエストフォーマットID (requestFormatID)	java.lang.String	in2 (type="xsd:string")		サービスリクエストからの要求電文フォーマットを一意に識別するためのIDです。 このパラメタにはNULLを指定してください。

## 10. サービスリクエストの作成

パラメタ名	データ型	パラメタ		説明
		invokeXML	invokeBinary	
レスポンスフォーマット ID (responseFormat ID)	java.lang.String	in3 ( type="xsd:string" )		HCSC サーバからの応答電文フォーマットを一意に識別するための ID です。 このパラメタには NULL を指定してください。
オペレーション名 (operationName)	java.lang.String	in4 ( type="xsd:string" )		要求先のサービス名に対するオペレーション名です。 オペレーション名は、開発環境で定義したサービス部品のオペレーション名を指定します。255 バイト以内の XMLSchema の NCName 定義文字で指定してください。 要求先のサービス部品が同期のサービス部品 ( Web サービスまたは SessionBean ) またはビジネスプロセスの場合、必ず指定してください。 要求先のサービス部品が非同期のサービス部品の場合、オペレーション名を省略できます。省略する場合、NULL を指定してください。
ユーザ電文 (userData)	java.lang.String	in5 ( type="xsd:string" )	×	サービスリクエストからの要求電文です。 要求電文が XML の場合に指定します。要求電文がない場合、NULL または空文字 ( "" ) を指定してください。要求電文の詳細については、「10.2.6 要求電文を作成する」を参照してください。
ユーザ電文長 (userDataBinary.length)	int	×	in5 ( type="xsd:int" )	要求電文の長さです。 要求電文がバイナリの場合に指定します。要求電文がバイナリの場合、必ず指定してください。 要求電文がない場合、0 を指定してください。
ユーザ電文 (userDataBinary)	byte[]	×	in6 ( type="base64Binary" )	サービスリクエストからの要求電文です。 要求電文がバイナリの場合に指定します。要求電文の詳細については、「10.2.6 要求電文を作成する」を参照してください。 要求電文がない場合、NULL または 0 バイトのバイト配列を指定してください。

( 凡例 )

inX ( X=1 ~ 6 ): スタブで示しているメソッドごとのパラメタです。

× : 指定できません。

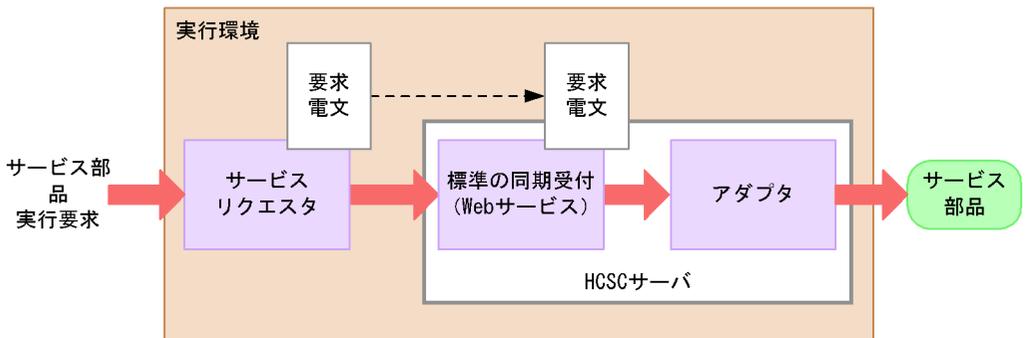
注

要求先のサービス部品のプロトコルが SOAP の場合、ユーザ電文のルート要素の名称（データ変換を行う場合、データ変換後の電文のルート要素の名称）から呼び出すオペレーションを決定します。そのため、ユーザ電文のルート要素に誤った名称を指定した場合、意図しないオペレーションを呼び出してしまうので注意してください。

## 10.2.6 要求電文を作成する

サービスリクエストから HCSC サーバの同期受付（Web サービス）に、サービス部品を要求する要求電文を作成します。サービスリクエストから送信する要求電文の内容は、サービス部品側と同じ電文フォーマットで作成してください。要求電文の送信を次の図に示します。

図 10-4 要求電文の送信（標準の同期受付（Web サービス））



正常な要求電文の伝わり方については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の Web サービス（SOAP 通信）を使ったサービス呼び出しの標準受付の場合に関する内容を参照してください。

また、サービスリクエストから同期受付（Web サービス）に送信する要求電文は、電文種別に XML 電文とバイナリ電文の 2 種類があります。どちらの電文種別を使用するかは、サービス部品側で使用しているプロトコルによって異なります。電文種別が XML 電文の要求電文とバイナリ電文の要求電文を次に説明します。

### （1）XML 電文の要求電文

サービス部品側のプロトコルが次に示すものの場合、およびビジネスプロセスに対して要求を行う場合、XML 電文の要求電文を送信できます。

- SOAP（サービスアダプタ（Web サービス）の場合）<sup>1 2</sup>
- RMI-IIOP（サービスアダプタ（SessionBean）の場合）<sup>1 2</sup>
- WS-R（サービスアダプタ（MDB（WS-R））の場合）<sup>2</sup>
- DB キュー（サービスアダプタ（MDB（DBQ））の場合）<sup>2</sup>

注 1

## 10. サービスリクエストの作成

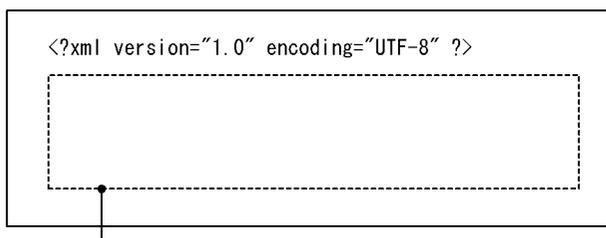
必ず要求電文を指定してください。

### 注 2

要求電文を指定しなかった場合、サービスアダプタでデータ変換処理が行われません。

XML 電文の要求電文を次の図に示します。

図 10-5 XML 電文の要求電文



サービス部品側と同じ電文フォーマットで作成した要求電文 XMLのタグにサービス部品に渡すパラメタの値を設定する

### ! 注意事項

指定する XML 文書（要求電文）の先頭には、必ず XML 宣言を記述してください。XML 宣言がない場合、または XML 文書中のユーザ電文に XML 宣言がある場合、動作の保証はできません。

また、XML 宣言の文字コードの指定（encoding）には、必ず「UTF-8」を指定してください。

## （2）バイナリ電文の要求電文

バイナリ電文の要求電文を送信できるのは、サービス部品側で使用しているプロトコルが DB キューのときだけです。

なお、要求電文を指定しなかった場合、サービスアダプタでデータ変換処理が行われません。

バイナリ電文の要求電文を次の図に示します。

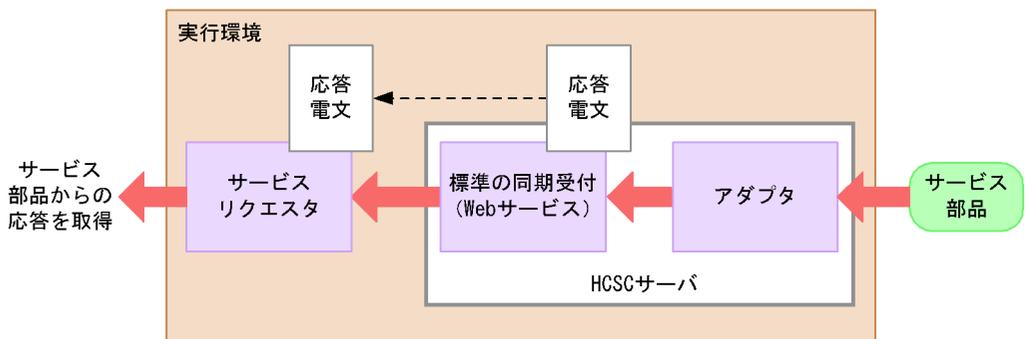
図 10-6 バイナリ電文の要求電文



## 10.2.7 応答電文を取得する

HCSC サーバの同期受付 (Web サービス) から、サービス部品の実行要求に対する応答電文を取得します。サービスリクエストは、サービス部品側と同じ電文フォーマットの応答電文を取得します。応答電文の取得を次の図に示します。

図 10-7 応答電文の取得 (標準の同期受付 (Web サービス))

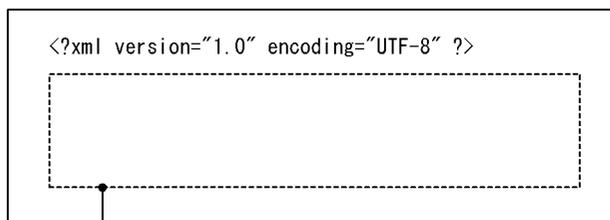


また、サービスリクエストは、電文種別が XML 電文の応答電文を取得します。サービス部品からの応答電文がない場合は、NULL または 0 バイトのバイト配列を受信します。応答電文がある場合とない場合に分けて次に説明します。

### (1) 応答電文がある場合

サービスリクエストが取得する、XML 電文の応答電文を次に示します。

図 10-8 XML 電文の応答電文



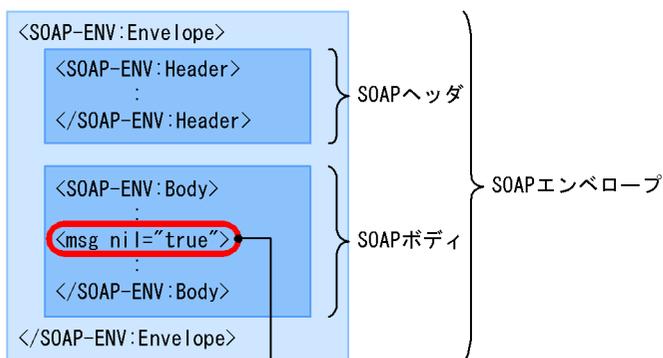
サービス部品側と同じ電文フォーマットの応答電文  
XMLのタグにサービスからの戻り値が設定されています。

## (2) 応答電文がない場合

応答電文がない場合、応答が String 型か byte[] 型によって、サービスリクエストが受信する形式が異なります。

- 応答が String 型の場合  
サービスリクエストは、次の図に示すような形式の NULL を受信します。

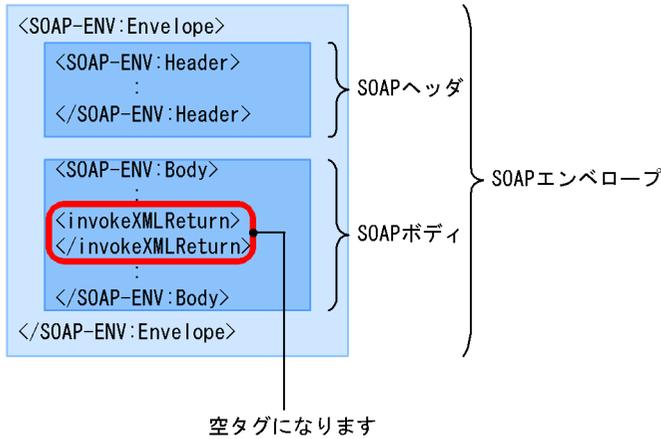
図 10-9 受信する NULL の形式



nil属性があるXMLのタグは、NULLを示します

- 応答が byte[] 型の場合  
サービスリクエストは、次の図に示すような形式の 0 バイトのバイト配列を受信します。SOAP エンベロープは次に示すように空タグとなります。

図 10-10 応答が byte[] 型の形式



### ! 注意事項

サービスリクエストで実装している SOAP エンジンによっては、図 10-9 および図 10-10 と取得形式が異なる場合があります。

## 10.2.8 エラー情報を取得する

要求先のサービス部品、HCSC サーバ、および SOAP エンジンでエラーが発生した場合、エラー情報を取得し、その情報に従って対処します。エラーの伝わり方については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の Web サービス（SOAP 通信）実行時の障害対策に関する内容を参照してください。

### (1) サービスリクエスト側の実装例

エラー情報の取得方法は、SOAP 通信基盤の種類によって異なります。

#### (a) Cosminexus が提供している SOAP 通信基盤を使用している場合

Cosminexus が提供している SOAP 通信基盤を使用している場合、CSCMsgServerFaultException オブジェクトをキャッチして、SOAPFault のエラー情報を取得します。

エラー情報を取得するには、サービスリクエスト側で次のように実装する必要があります。

## 10. サービスリクエストの作成

```
        :
    } catch (CSCMsgServerFaultException e) {
        System.err.println("Exception ErrorMessage = "
            + e.getErrorMessage() );
        System.err.println("Exception ErrorCode = "
            + e.getErrorCode() );
        System.err.println("Exception ProcessInstanceID = "
            + e.getProcessInstanceID() );
        System.err.println("Exception FaultCode = "
            + e.getCscmsgFaultCode() );
        System.err.println("Exception FaultString = "
            + e.getCscmsgFaultString() );
        System.err.println("Exception FaultActor = "
            + e.getCscmsgFaultActor() );
        System.err.println("Exception FaultDetails = "
            + new String(e.getCscmsgFaultDetail(), "UTF-8"));
        System.err.println("Exception FaultName = "
            + e.getFaultName() );
    }
        :
```

各メソッドについて、次に説明します。

### getErrorMessage

#### 説明

エラーメッセージを取得します。

次の例外の内容を取得する場合に使用します。

- HCSC-Messaging 内で検知した例外
- サービス部品やビジネスプロセスからのフォルト

#### 形式

```
public java.lang.String getErrorMessage()
```

### getErrorCode

#### 説明

エラーコードを取得します。

次の例外の内容に対するエラーコードを取得する場合に使用します。

- HCSC-Messaging 内で検知した例外
- サービス部品やビジネスプロセスからのフォルト

#### 形式

```
public java.lang.String getErrorCode()
```

### getProcessInstanceID

#### 説明

ビジネスプロセスインスタンス ID を取得します。

#### 形式

```
public String getProcessInstanceID()
```

`getCscmsgFaultCode`

## 説明

サービス部品 (Web サービス), ビジネスプロセス, またはカスタムアダプタからの `FaultCode` 情報を取得します。

## 形式

```
public java.lang.String getCscmsgFaultCode()
```

`getCscmsgFaultString`

## 説明

サービス部品 (Web サービス), ビジネスプロセス, またはカスタムアダプタからの `FaultString` 情報を取得します。

## 形式

```
public java.lang.String getCscmsgFaultString()
```

`getCscmsgFaultActor`

## 説明

サービス部品 (Web サービス), ビジネスプロセス, またはカスタムアダプタからの `FaultActor` 情報を取得します。

## 形式

```
public java.lang.String getCscmsgFaultActor()
```

`getCscmsgFaultDetail`

## 説明

サービス部品 (Web サービス), ビジネスプロセス, またはカスタムアダプタからの `Detail` 情報を取得します。

サービス部品が設定した `Detail` 情報をバイト配列でサービスリクエストに引き渡します。したがって, 取得したバイト配列を文字列に変換する必要があります。

なお, サービス部品 (Web サービス), ビジネスプロセス, またはカスタムアダプタからの `Detail` 情報がない場合は, 0 バイトのバイト配列 (SOAP メッセージの空タグ) で応答します。

## 形式

```
public byte[] getCscmsgFaultDetail()
```

`getFaultName`

## 説明

サービス部品 (Web サービス) またはビジネスプロセスからの例外名を取得します。

## 形式

```
public String getFaultName()
```

## 10. サービスリクエストの作成

(b) Cosminexus が提供している SOAP 通信基盤を使用していない場合

取得するエラー情報は、サービスリクエスト側で実装している SOAP エンジンに依存します。

### (2) エラー情報 (SOAPFault) の形式

エラー情報 (SOAPFault) の形式を次の表に示します。

表 10-4 エラー情報 (SOAPFault) の形式

要素	名称	内容
faultcode	Fault コード	SOAP エンジンに依存する値です。 message part のメッセージ型属性が参照する QName です。
faultstring	Fault 文字列	SOAP エンジンに依存する値です。 KDCCP00015-E を出力します。
faultactor	Fault 生成者	SOAP エンジンに依存する値です。値はありません。
detail	Fault 詳細	wsdl:fault に対応する内容です。

注

要素「detail」はエラー情報の詳細です。次の表に示す構造体で格納されます。

表 10-5 エラー情報 (SOAPFault) の詳細

フィールド名	説明	
		サービス部品、ビジネスプロセス、カスタムアダプタ、または連携する HCSC サーバ (クラスタ) からのエラー (フォルト)
errorMessage	次に示すエラーの内容です。 <ul style="list-style-type: none"> <li>HCSC サーバ内で検知したエラー</li> <li>サービス部品、ビジネスプロセス、およびカスタムアダプタからのエラー</li> </ul>	
errorCode	次に示す例外の内容に対応するエラーコードです。 <ul style="list-style-type: none"> <li>HCSC サーバ内で検知したエラー</li> <li>サービス部品、ビジネスプロセス、およびカスタムアダプタからのエラー</li> </ul>	
processInstanceID	ビジネスプロセスのインスタンス ID の情報です。 ビジネスプロセスでエラーが発生した場合に値が設定されます。	
cscmsgFaultCode	サービス部品 (Web サービス)、ビジネスプロセス、またはカスタムアダプタからの FaultCode 情報です。	-
cscmsgFaultString	サービス部品 (Web サービス)、ビジネスプロセス、またはカスタムアダプタからの FaultString 情報です。	-
cscmsgFaultActor	サービス部品 (Web サービス)、ビジネスプロセス、またはカスタムアダプタからの FaultActor 情報です。	-

フィールド名	説明	
	サービス部品、ビジネスプロセス、カスタムアダプタ、または連携する HCSC サーバ (クラスタ) からのエラー (フォルト)	HCSC サーバで検知したエラー (例外)
cscmsgFaultDetail	サービス部品 (Web サービス)、ビジネスプロセス、またはカスタムアダプタからの Detail 情報です。	-
faultName	<p>サービス部品 (Web サービスまたは SessionBean) またはビジネスプロセスからのフォルト名 (例外名) 情報です。次の場合に値が設定されます。</p> <ul style="list-style-type: none"> <li>サービス部品 (Web サービスまたは SessionBean) からのユーザ定義例外の SOAP Fault の場合</li> <li>ビジネスプロセスからのフォルトの場合</li> </ul> <p>SOAP Fault 動作定義ファイルに targetNamespace の URI を定義した Web サービスからの SOAP Fault の場合、値は設定されません。SOAP Fault 動作定義ファイルについては、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。</p>	-

(凡例) - : 該当しません。

## 10.2.9 ビジネスプロセスの再実行を要求するサービスリクエストの作成 (Web サービス)

標準の同期受付 (Web サービス) に要求を出すサービスリクエストとして、ビジネスプロセスの再実行を要求するサービスリクエストを作成できます。

標準の同期受付 (Web サービス) にビジネスプロセスの再実行を要求するサービスリクエストの作成手順は、標準の同期受付 (Web サービス) に要求を出す、通常のサービスリクエストを作成する場合と同じです。作成手順については、「10.2.1 サービスリクエスト (標準の同期受付 (Web サービス)) の作成手順」を参照してください。

手順ごとに実施する内容を次に示します。

### (1) WSDL の取得

WSDL の取得方法は、標準の同期受付 (Web サービス) に要求を出す、通常のサービスリクエストを作成する場合と同じです。

WSDL の取得方法については、「10.2.2 WSDL を取得する」を参照してください。

## (2) スタブの生成

スタブの生成方法は、標準の同期受付（Web サービス）に要求を出す、通常のサービスリクエストを作成する場合と同様に、Cosminexus が開発支援コマンドとして提供する WSDL2Java コマンドで生成します。

スタブの生成については、「10.2.3 スタブを生成する」を参照してください。

## (3) オブジェクトの生成

オブジェクトの生成方法は、標準の同期受付（Web サービス）に要求を出す、通常のサービスリクエストを作成する場合と同じです。

オブジェクトの生成については、「10.2.4 オブジェクトを生成する」を参照してください。

なお、ビジネスプロセスの再実行を要求するメソッドは、`invokeBPXML()` を使用して呼び出します。

ビジネスプロセスの再実行を要求するメソッドの例を次に示します。

### (例) ビジネスプロセスの再実行を要求する場合

```
String result = ws.invokeBPXML(           // メソッドの呼び出し
    serviceName,                          // サービス名
    bpRequestType,                        // ビジネスプロセス用要求タイプ
    bpProcessId,                          // ビジネスプロセス用プロセスID
    clientId,                              // クライアント関連ID
    requestFormatID,                      // リクエストフォーマットID
    responseFormatID,                    // レスポンスフォーマットID
    operationName,                        // オペレーション名
    userData);                             // ユーザ電文
```

## (4) パラメタの設定

メソッドの引数となるパラメタは、標準の同期受付（Web サービス）に要求を出す、通常のサービスリクエストを作成する場合と異なります。同期受付（Web サービス）にビジネスプロセスの再実行を要求するサービスリクエストの場合に設定するパラメタの詳細を次に示します。

表 10-6 パラメタの詳細（標準の同期受付（Web サービス）/ビジネスプロセス再実行要求）

パラメタ名	データ型	パラメタ ( invokeBPXML )	説明
サービス名 ( serviceName )	java.lang. String	in0 ( type="xsd:string" )	要求先のサービス名です。 このパラメタは必ず指定してください。 要求先のサービス名は、開発環境で定義したビジネスプロセスを指定してください。
ビジネスプロセス 用要求タイプ ( bpRequestType )	java.lang. String	in1 ( type="xsd:string" )	要求電文の種類を示します。 ビジネスプロセスの再実行を要求する場合、文字列「RECOVER」を指定してください。 <sup>1</sup>
ビジネスプロセス 用プロセス ID ( bpProcessId )	java.lang. String	in2 ( type="xsd:string" )	ビジネスプロセスインスタンス ID です。 エラー情報から取得した値、または、メッセージログに出力された値を指定してください。
クライアント関連 ID ( clientID )	java.lang. String	in3 ( type="xsd:string" )	サービスリクエストからの要求電文を一意に識別するための相関識別子です。 このパラメタには NULL を指定してください。 <sup>1</sup>
リクエストフォー マット ID ( requestFormatI D )	java.lang. String	in4 ( type="xsd:string" )	サービスリクエストからの要求電文フォーマットを一意に識別するための ID です。 このパラメタには NULL または空文字 ( "" ) を指定してください。 <sup>1</sup>
レスポンスフォー マット ID ( responseFormat ID )	java.lang. String	in5 ( type="xsd:string" )	HCSC サーバからの応答電文フォーマットを一意に識別するための ID です。 このパラメタには NULL または空文字 ( "" ) を指定してください。 <sup>1</sup>
オペレーション名 ( operationName )	java.lang. String	in6 ( type="xsd:string" )	要求先のサービス名に対するオペレーション名です。 <sup>2</sup> このパラメタには NULL または空文字 ( "" ) を指定してください。 <sup>1</sup>
ユーザ電文 ( userData )	java.lang. String	in7 ( type="xsd:string" )	サービスリクエストからの要求電文です。 <sup>2</sup> このパラメタには NULL または空文字 ( "" ) を指定してください。 <sup>1</sup>

( 凡例 )

inX ( X=0 ~ 7 ): スタブで示しているメソッドごとのパラメタです。

注 1

これらのパラメタは、指定する値として固定値（「RECOVER」、NULL または空文字（ "" ））が決められています。決められた値以外の値を指定すると、正常に動作しません。

## 10. サービスリクエストの作成

### 注 2

要求先のサービス部品のプロトコルが SOAP の場合、ユーザ電文のルート要素の名称（データ変換を行う場合、データ変換後の電文のルート要素の名称）から呼び出すオペレーションを決定します。そのため、ユーザ電文のルート要素に誤った名称を指定した場合、意図しないオペレーションを呼び出してしまうので注意してください。

### (5) 応答電文の取得

応答電文の取得方法は、標準の同期受付（Web サービス）に要求を出す、通常のリクエストを作成する場合と同じです。

応答電文の取得については、「10.2.7 応答電文を取得する」を参照してください。

### (6) エラー情報の取得

エラー情報を取得するためのサービスリクエストの実装は、標準の同期受付（Web サービス）に要求を出す、通常のリクエストを作成する場合と同じです。

エラー情報の取得については、「10.2.8 エラー情報を取得する」を参照してください。

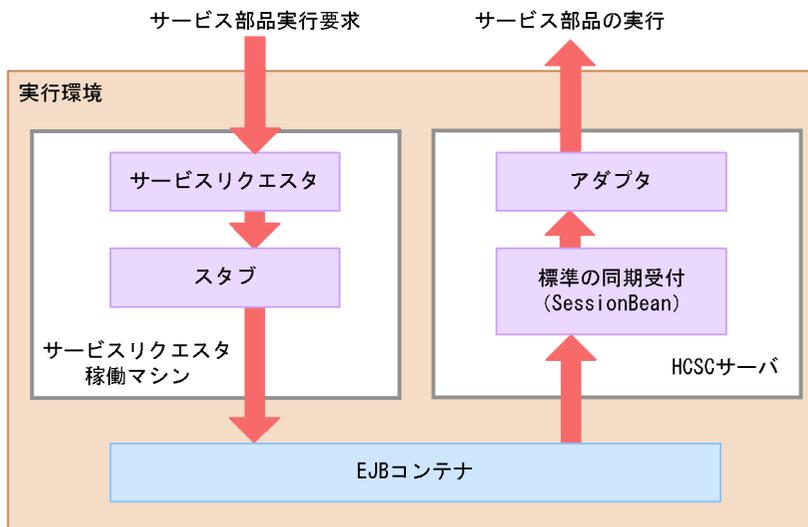
## 10.3 標準の同期受付（SessionBean）に要求を出すサービスリクエスト

標準の同期受付（SessionBean）へ要求電文を送信するサービスリクエストは、RMI-IIOP を利用して標準受付と通信します。サービスリクエストから標準受付へサービス部品実行の要求電文が送信され、HCSC サーバがサービス部品の実行を行います。

ユーザは開発環境からスタブを取得し、そのスタブを利用して、同期受付（SessionBean）に要求を出します。したがって、サービスリクエストは、取得したスタブを利用できるように実装する必要があります。

標準の同期受付（SessionBean）に要求を出すサービスリクエストと HCSC サーバの関係を次の図に示します。

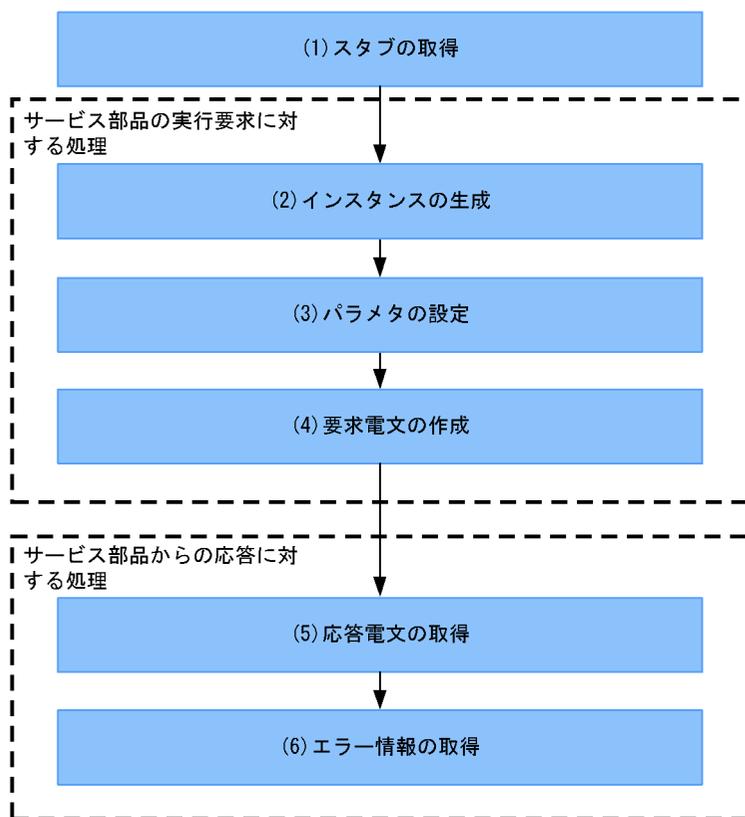
図 10-11 標準の同期受付（SessionBean）に要求を出すサービスリクエストと HCSC サーバの関係



### 10.3.1 サービスリクエスト（標準の同期受付（SessionBean））の作成手順

標準の同期受付（SessionBean）にサービス部品の実行を要求し、サービス部品を呼び出すためのサービスリクエストの作成手順を次に示します。

図 10-12 サービスリクエストの作成手順 (標準の同期受付 (SessionBean))



各工程の作業を次に示します。

### (1) スタブの取得

開発環境の HCSC コンポーネント情報表示画面から、HCSC サーバが配備されている J2EE サーバ (JNDI 名) に対応したスタブを取得します。スタブの取得については、「10.3.2 スタブを取得する」を参照してください。

### (2) インスタンスの生成

同期受付 (SessionBean) のメソッドを呼び出すために、(1) で取得したスタブから Enterprise Bean のインスタンスを生成します。インスタンスの生成については、「10.3.3 インスタンスを生成する」を参照してください。

### (3) パラメタの設定

同期受付 (SessionBean) のメソッドの引数となるパラメタを設定します。パラメタの設定については、「10.3.4 パラメタを設定する」を参照してください。

#### (4) 要求電文の作成

サービス部品の実行を要求する要求電文を作成します。要求電文の作成については、「10.3.5 要求電文を作成する」を参照してください。

#### (5) 応答電文の取得

同期受付 (SessionBean) から、サービス部品の実行要求に対する応答電文を取得します。応答電文の取得については、「10.3.6 応答電文を取得する」を参照してください。

#### (6) エラー情報の取得

要求先のサービス部品、HCSC サーバ、および EJB コンテナでエラーが発生した場合、エラー情報を取得し、その情報に従って対処します。エラー情報の取得については、「10.3.7 エラー情報を取得する」を参照してください。

### 10.3.2 スタブを取得する

HCSC サーバが配備されている J2EE サーバ (JNDI 名) に対応したスタブを取得します。スタブを取得する方法を次に示します。

1. Eclipse のメニューから、[ HCSC-Definer ] - [ 公開サービス一覧 ] を選択します。  
ツリービューの公開サービス一覧に、HCSC コンポーネントの一覧が表示されます。
2. HCSC コンポーネントの一覧から、呼び出すサービス部品 (HCSC コンポーネント) を選択し、ダブルクリックします。  
HCSC コンポーネント情報表示画面に、HCSC コンポーネントの情報が表示されます。HCSC コンポーネント情報表示画面の詳細については、「11.2.6 HCSC コンポーネント情報表示画面」を参照してください。  
[ サーバ情報 ] には、サービス部品の要求先となる HCSC サーバの情報が表示されません。
3. [ スタブ取得 ] ボタンをクリックします。  
[ スタブファイルの保存 ] ダイアログが表示されます。  
保存先を指定して、スタブを取得します。

取得したスタブの構成を次に示します。

```
/カレントディレクトリ
cscmsg_ejb_client.jar
```

### 10.3.3 インスタンスを生成する

同期受付 (SessionBean) のメソッドを呼び出すために、取得したスタブを使用して、Enterprise Bean のインスタンスを生成します。Enterprise Bean のインスタンスを生成

## 10. サービスリクエストの作成

する手順を次に示します。

1. EJB ホームオブジェクトのリファレンスの検索に利用する JNDI ネーミングコンテキストを生成します。

(例)

```
javax.naming.Context ctx = new javax.naming.InitialContext();
```

2. 生成した JNDI ネーミングコンテキストを使用して、EJB ホームオブジェクトのリファレンスを取得します。

EJB ホームオブジェクトのリファレンスは、ユーザ指定名前空間または EJB コンテナ名前空間を利用して検索し、取得します。EJB ホームオブジェクトのリファレンスの検索・取得の詳細については、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。

(例)

```
Object objref
= initial.lookup("HITACHI_EJB/SERVERS/" + "<J2EEサーバ名称>"
+ "/EJB/CSCMsgSyncServiceDelivery/CSCMsgSyncServiceDeliveryEJB");

CSCMsgSyncServiceDeliveryHome home
= (CSCMsgSyncServiceDeliveryHome) PortableRemoteObject
.narrow(objref, CSCMsgSyncServiceDeliveryHome.class);
```

3. EJB ホームオブジェクトの create メソッドによって、Enterprise Bean のインスタンスを生成します。生成した Enterprise Bean のインスタンスを使用して、同期受付 (SessionBean) のメソッド (Enterprise bean のメソッド) を呼び出します。

(例) 要求電文が XML の場合

```
CSCMsgSyncServiceDelivery reception = home.create();
// Enterprise Beanのインスタンスを生成
String result = reception.invokeXML( // メソッドの呼び出し
    serviceName, // サービス名
    clientID, // クライアント関連ID
    requestFormatID, // リクエストフォーマットID
    responseFormatID, // レスポンスフォーマットID
    operationName, // オペレーション名
    userData); // ユーザ電文
```

## (例) 要求電文がバイナリの場合

```

CSCMsgSyncServiceDelivery reception = home.create();
// Enterprise Beanのインスタンスを生成
byte[] resultBinary = reception .invokeBinary( // メソッドの呼び出し
    serviceName, // サービス名
    clientID, // クライアント関連ID
    requestFormatID, // リクエストフォーマットID
    responseFormatID, // レスポンスフォーマットID
    operationName, // オペレーション名
    userDataBinary.length, // メッセージ長
    userDataBinary); // ユーザ電文

```

**!** 注意事項

バイナリ電文の要求電文を送信できるのは、サービス部品側で使用している電文形式がバイナリ形式のときだけです。

## 10.3.4 パラメタを設定する

同期受付 (SessionBean) のメソッドを呼び出すため、メソッドの引数となるパラメタを設定します。パラメタの詳細を次の表に示します。

表 10-7 パラメタの詳細 (標準の同期受付 (SessionBean))

パラメタ名	データ型	パラメタ		説明
		invokeXML	invokeBinary	
サービス名	java.lang.String	serviceName		要求先のサービス名です。 このパラメタは必ず指定してください。 要求先のサービス名は、開発環境で定義したアダプタまたはビジネスプロセスを指定してください。
クライアント関連ID	java.lang.String	clientID		サービスリクエストからの要求電文を一意に識別するための関連識別子です。 255文字以内の英数字、アンダーライン(_), ピリオド(.)およびハイフン(-)で指定してください。 サービスリクエストからの要求電文と、HCSCサーバで管理している実行履歴やログ・トレースを対応づけるために使用します。したがって、HCSCサーバに送信した要求電文ごとに異なるIDを指定してください。 クライアント関連IDを省略したい場合、NULLを指定してください。

## 10. サービスリクエストの作成

パラメタ名	データ型	パラメタ		説明
		invokeXML	invokeBinary	
リクエストフォーマット ID	java.lang.String	cscRequestFormatID		サービスリクエストからの要求電文フォーマットを一意に識別するための ID です。 このパラメタには NULL を指定してください。
レスポンスフォーマット ID	java.lang.String	cscResponseFormatID		HSCS サーバからの応答電文フォーマットを一意に識別するための ID です。 このパラメタには NULL を指定してください。
オペレーション名	java.lang.String	serviceOperationName		要求先のサービス名に対するオペレーション名です。 オペレーション名は、開発環境で定義したサービス部品のオペレーション名を指定します。255 バイト以内の XMLSchema の NCName 定義文字で指定してください。 要求先のサービス部品が同期のサービス部品 (Web サービスまたは SessionBean) またはビジネスプロセスの場合、必ず指定してください。 要求先のサービス部品が非同期のサービス部品の場合、オペレーション名を省略できます。省略する場合、NULL を指定してください。
ユーザ電文	java.lang.String	msg	x	サービスリクエストからの要求電文です。 要求電文が XML の場合に指定します。要求電文がない場合、NULL または空文字 ("") を指定してください。要求電文の詳細については、「10.3.5 要求電文を作成する」を参照してください。
ユーザ電文長	int	x	requestMessageLength	要求電文の長さです。 要求電文がバイナリの場合に指定します。要求電文がバイナリの場合は、必ず指定してください。 要求電文がない場合、0 を指定してください。

パラメタ名	データ型	パラメタ		説明
		invokeXML	invokeBinary	
ユーザ電文	byte[]	x	msg	サービスリクエストからの要求電文です。 要求電文がバイナリの場合に指定します。要求電文の詳細については、「10.3.5 要求電文を作成する」を参照してください。 要求電文がない場合、NULLまたは0バイトのバイト配列を指定してください。

(凡例)

x : 指定できません。

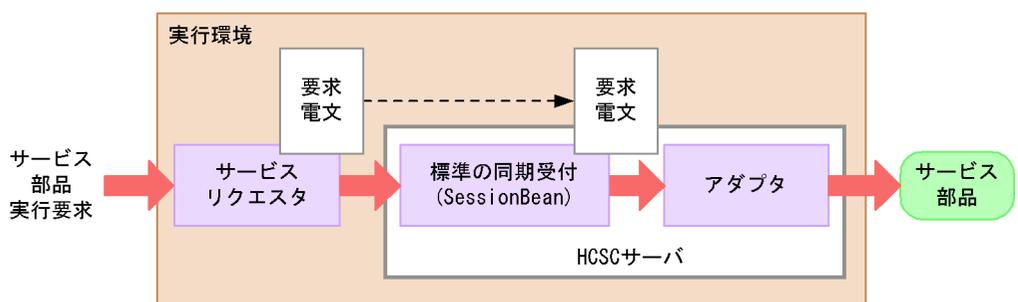
注

要求先のサービス部品のプロトコルが SOAP の場合、ユーザ電文のルート要素の名称（データ変換を行う場合、データ変換後の電文のルート要素の名称）から呼び出すオペレーションを決定します。そのため、ユーザ電文のルート要素に誤った名称を指定した場合、意図しないオペレーションを呼び出してしまうので注意してください。

### 10.3.5 要求電文を作成する

サービスリクエストから HCSC サーバの同期受付 (SessionBean) に、サービス部品を要求する要求電文を作成します。サービスリクエストから送信する要求電文の内容は、サービス部品側と同じ電文フォーマットで作成してください。要求電文の送信を次の図に示します。

図 10-13 要求電文の送信 (標準の同期受付 (SessionBean))



また、サービスリクエストから同期受付 (SessionBean) に送信する要求電文は、電文種別に XML 電文とバイナリ電文の 2 種類があります。どちらの電文種別を使用するかは、サービス部品側で使用しているプロトコルによって異なります。

XML 電文およびバイナリ電文の要求電文については、「10.2.6 要求電文を作成する」を参照してください。

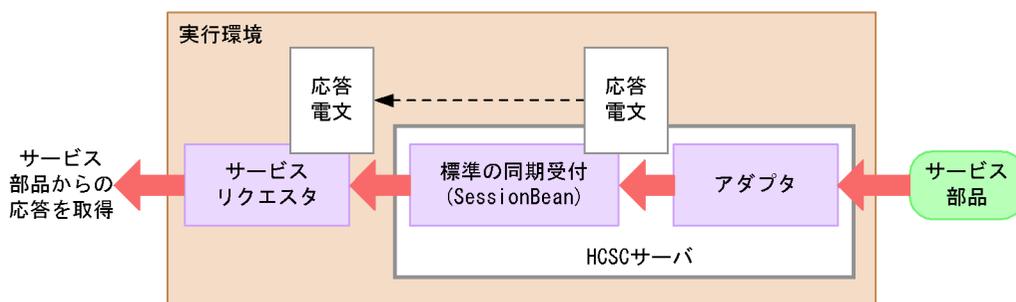
## 10. サービスリクエストの作成

正常な要求電文の伝わり方については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の SessionBean を使ったサービス呼び出しに関する内容を参照してください。

### 10.3.6 応答電文を取得する

HCSC サーバの同期受付 (SessionBean) から、サービス部品の実行要求に対する応答電文を取得します。サービスリクエストは、サービス部品側と同じ電文フォーマットの応答電文を取得します。応答電文の取得を次の図に示します。

図 10-14 応答電文の取得 (標準の同期受付 (SessionBean))



また、サービスリクエストは、電文種別が XML 電文の応答電文を取得します。サービス部品からの応答電文がない場合は、NULL を受信します。

XML 電文の応答電文および NULL の受信については、「10.2.7 応答電文を取得する」を参照してください。

### 10.3.7 エラー情報を取得する

要求先のサービス部品、HCSC サーバ、および EJB コンテナでエラーが発生した場合、エラー情報を取得し、その情報に従って対処します。

エラー情報は、サービスリクエスト側で CSCMsgServerException クラスを取得することによって取得できます。CSCMsgServerException クラスについては「10.3.7(2) CSCMsgServerException クラス」を参照してください。エラーの伝わり方については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の SessionBean 実行時の障害対策に関する内容を参照してください。

エラーの詳細については、getErrorMessage メソッドおよび getErrorCode メソッドを使用して取得します。また、Web サービスからのエラー情報 (SOAPFault) を取得する場合は、checkSoapFault メソッドを使用します。

#### (1) サービスリクエスト側の実装例

CSCMsgServerException クラスを取得する場合のサービスリクエスト側の実装例を次

に示します。

```

:
} catch (CSCMsgServerException e) {
    System.err.println("Exception ErrorMessage = "
        + e.getMessage() );
    System.err.println("Exception ErrorCode = "
        + e.getErrorCode() );
    if (e.checkFaultMessage() == true) {
        System.err.println("Exception ProcessInstanceID = "
            + e.getProcessInstanceID() );
        System.err.println("Exception FaultCode = "
            + e.getFaultCode() );
        System.err.println("Exception FaultString = "
            + e.getFaultString() );
        System.err.println("Exception FaultActor = "
            + e.getFaultActor() );
        System.err.println("Exception FaultDetails = "
            + new String(e.getFaultDetail(), "UTF-8"));
        System.err.println("Exception FaultName = "
            + e.getFaultName() );
    }
}
:

```

## (2) CSCMsgServerException クラス

サービスリクエストが取得する例外クラスです。

### (a) クラス定義

パッケージ

```
jp.co.Hitachi.soft.csc.msg.message.reception
```

クラス

```
public class CSCMsgServerException
    extends java.lang.Exception
```

### (b) フィールドの一覧

フィールド名	データ型	説明
errorMessage	java.lang.String	次に示す例外の内容です。 <ul style="list-style-type: none"> <li>HCSC-Messaging 内で検知した例外</li> <li>サービス部品またはビジネスプロセスからのフォルト</li> </ul>
errorCode	java.lang.String	次に示す例外の内容に対応するエラーコードです。 <ul style="list-style-type: none"> <li>HCSC-Messaging 内で検知した例外</li> <li>サービス部品またはビジネスプロセスからのフォルト</li> </ul>
processInstanceID	java.lang.String	ビジネスプロセスのインスタンス ID の情報です。 ビジネスプロセスでエラーが発生した場合に値が設定されま ず。
cscmsgFaultCode	java.lang.String	サービス部品 (Web サービス), ビジネスプロセス, またはカ スタムアダプタからの FaultCode 情報です。
cscmsgFaultString	java.lang.String	サービス部品 (Web サービス), ビジネスプロセス, またはカ スタムアダプタからの FaultString 情報です。

## 10. サービスリクエストの作成

フィールド名	データ型	説明
cscmsgFaultActor	java.lang.String	サービス部品 (Web サービス), ビジネスプロセス, またはカスタムアダプタからの FaultActor 情報です。
cscmsgFaultDetail	byte[]	サービス部品 (Web サービス), ビジネスプロセス, またはカスタムアダプタからの Detail 情報です。
faultName	java.lang.String	サービス部品 (Web サービスまたは SessionBean), またはビジネスプロセスからのフォルト名 (例外名) 情報です。次の場合に値が設定されます。 <ul style="list-style-type: none"> <li>サービス部品 (Web サービスまたは SessionBean) からのユーザ定義例外の SOAP Fault の場合</li> <li>ビジネスプロセスからのフォルトの場合</li> </ul> SOAP Fault 動作定義ファイルに targetNamespace の URI を定義した Web サービスからの SOAP Fault の場合, 値は設定されません。SOAP Fault 動作定義ファイルについては, マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。

### (c) メソッドの一覧

メソッド名	データ型	説明
getErrorMessage	java.lang.String	エラーメッセージを取得します。
getErrorCode	java.lang.String	エラーコードを取得します。
checkFaultMessage	boolean	サービス部品, ビジネスプロセス, またはカスタムアダプタからのフォルト情報があるかどうかをチェックします。
getFaultCode	java.lang.String	FaultCode 情報を取得します。
getFaultString	java.lang.String	FaultString 情報を取得します。
getFaultActor	java.lang.String	FaultActor 情報を取得します。
getFaultDetail	byte[]	Detail 情報を取得します。
getProcessInstanceId	java.lang.String	ビジネスプロセスのインスタンス ID を取得します。
getFaultName	java.lang.String	フォルト名 (例外名) を取得します。

### (d) メソッドの詳細

#### getErrorMessage

##### 説明

エラーメッセージを取得します。

次の例外の内容を取得する場合に使用します。

- HCSC-Messaging 内で検知した例外
- サービス部品またはビジネスプロセスからのフォルト

##### 形式

```
public java.lang.String getErrorMessage()
```

パラメタ

なし

戻り値

エラーメッセージ

例外

なし

getErrorCode

説明

エラーコードを取得します。

次の例外の内容に対するエラーコードを取得する場合に使用します。

- HCSC-Messaging 内で検知した例外
- サービス部品またはビジネスプロセスからのフォルト

形式

```
public java.lang.String getErrorCode()
```

パラメタ

なし

戻り値

エラーコード

例外

なし

checkFaultMessage

説明

サービス部品、ビジネスプロセス、またはカスタムアダプタからのフォルト情報があるかどうかをチェックします。

サービス部品またはビジネスプロセスが Web サービスの場合、SOAPFault として返されたエラーの内容が含まれているかどうかをチェックします。

また、カスタムアダプタからの個別のエラーが含まれているかどうかをチェックします。

形式

```
public boolean checkFaultMessage()
```

パラメタ

なし

戻り値

true : Fault メッセージ情報が存在します。

false : Fault メッセージ情報が存在しません。

例外

なし

## 10. サービスリクエストの作成

### getFaultCode

#### 説明

サービス部品 ( Web サービス ), ビジネスプロセス, またはカスタムアダプタからの FaultCode 情報を取得します。

#### 形式

```
public java.lang.String getFaultCode()
```

#### パラメタ

なし

#### 戻り値

faultcode

#### 例外

なし

### getFaultString

#### 説明

サービス部品 ( Web サービス ), ビジネスプロセス, またはカスタムアダプタからの FaultString 情報を取得します。

#### 形式

```
public java.lang.String getFaultString()
```

#### パラメタ

なし

#### 戻り値

faultstring

#### 例外

なし

### getFaultActor

#### 説明

サービス部品 ( Web サービス ), ビジネスプロセス, またはカスタムアダプタからの FaultActor 情報を取得します。

#### 形式

```
public java.lang.String getFaultActor()
```

#### パラメタ

なし

#### 戻り値

faultactor

#### 例外

なし

### getFaultDetail

#### 説明

サービス部品 (Web サービス), ビジネスプロセス, またはカスタムアダプタからの Detail 情報を取得します。

サービス部品が設定した Detail 情報をバイト配列でサービスリクエストに引き渡します。したがって, 取得したバイト配列を文字列に変換する必要があります。

#### 形式

```
public byte[] getFaultDetail()
```

#### パラメタ

なし

#### 戻り値

detail

#### 例外

なし

### getProcessInstanceID

#### 説明

ビジネスプロセスのインスタンス ID を取得します。

#### 形式

```
public String getProcessInstanceID()
```

#### パラメタ

なし

#### 戻り値

String

#### 例外

なし

### getFaultName

#### 説明

フォルト名 (例外名) を取得します。

#### 形式

```
public String getFaultName()
```

#### パラメタ

なし

#### 戻り値

String

#### 例外

なし

## (e) サブクラスの一覧

クラス名	サブクラス名と説明	
CSCMsgServerException	CSCMsgServiceException サービス部品例外	CSCMsgServiceUserException サービス部品からのフォルト情報または Exception
		CSCMsgBusinessProcessUserException ビジネスプロセスからのフォルト情報または Exception
		CSCMsgConnectServerException HCSC サーバ（クラスタ）間連携におけるフォルト情報または Exception
		CSCMsgServiceExecuteException サービス部品（Web サービス）からユーザ定義例外以外の例外が返ってきた場合の例外
	CSCMsgServiceDeliveryException サービス部品要求振り分けエラー	CSCMsgLocationSearchException ロケーション検索失敗
		CSCMsgRoutingExecException ルーティング失敗
		CSCMsgDelivererExecException メッセージ配送失敗
		CSCMsgDBQServiceDeliveryException 非同期 DB キューの標準受付から DB キューのサービス部品を呼んだ場合の例外
	CSCMsgServerParameterException 入力パラメタエラー	-
	CSCMsgServerInternalException 上記以外のエラー	-

(凡例) - : サブクラスはありません。

### 10.3.8 ビジネスプロセスの再実行を要求するサービスリクエストの作成 (SessionBean)

標準の同期受付 (SessionBean) に要求を出すサービスリクエストとして、ビジネスプロセスの再実行を要求するサービスリクエストを作成できます。

標準の同期受付 (SessionBean) にビジネスプロセスの再実行を要求するサービスリクエストの作成手順は、標準の同期受付 (SessionBean) に要求を出す、通常のサービスリクエストを作成する場合と同じです。作成手順については、「10.3.1 サービスリクエ

スタ (標準の同期受付 (SessionBean)) の作成手順」を参照してください。

手順ごとに実施する内容を次に示します。

### (1) スタブの取得

スタブの取得方法は、標準の同期受付 (SessionBean) に要求を出す、通常のサービスリクエストを作成する場合と同じです。

スタブの取得については、「10.3.2 スタブを取得する」を参照してください。

### (2) インスタンスの生成

インスタンスの生成方法は、標準の同期受付 (SessionBean) に要求を出す、通常のサービスリクエストを作成する場合と同じです。

インスタンスの生成については、「10.3.3 インスタンスを生成する」を参照してください。

なお、ビジネスプロセスの再実行を要求するメソッドは、`invokeBPXML()` を使用して呼び出します。

ビジネスプロセスの再実行を要求するメソッドの例を次に示します。

(例) ビジネスプロセスの再実行を要求する場合

```
String result = ws.invokeBPXML(           // メソッドの呼び出し
    serviceName,                          // サービス名
    bpRequestType,                        // ビジネスプロセス用要求タイプ
    bpProcessId,                          // ビジネスプロセス用プロセスID
    clientID,                              // クライアント相関ID
    requestFormatID,                      // リクエストフォーマットID
    responseFormatID,                    // レスポンスフォーマットID
    operationName,                        // オペレーション名
    userData);                            // ユーザ電文
```

### (3) パラメタの設定

メソッドの引数となるパラメタは、標準の同期受付 (SessionBean) に要求を出す、通常のサービスリクエストを作成する場合と異なります。同期受付 (SessionBean) にビジネスプロセスの再実行を要求するサービスリクエストの場合に設定するパラメタの詳細を次に示します。

## 10. サービスリクエストの作成

表 10-8 パラメタの詳細（標準の同期受付（SessionBean）/ ビジネスプロセス再実行要求）

パラメタ名	データ型	パラメタ（invokeBPXML）	説明
サービス名	java.lang.String	serviceName	要求先のサービス名です。 このパラメタは必ず指定してください。 要求先のサービス名は、開発環境で定義したビジネスプロセスを指定してください。
ビジネスプロセス用要求タイプ	java.lang.String	cscBpRequestType	要求電文の種類を示します。 ビジネスプロセスの再実行を要求する場合、文字列「RECOVER」を指定してください。
ビジネスプロセス用プロセス ID	java.lang.String	cscBpProcessID	ビジネスプロセスインスタンス ID です。 エラー情報から取得した値、または、メッセージログに出力された値を指定してください。
クライアント相関 ID	java.lang.String	clientID	サービスリクエストからの要求電文を一意に識別するための相関識別子です。 このパラメタには NULL を指定してください。
リクエストフォーマット ID	java.lang.String	cscRequestFormatID	サービスリクエストからの要求電文フォーマットを一意に識別するための ID です。 このパラメタには NULL または空文字（" "）を指定してください。
レスポンスフォーマット ID	java.lang.String	cscResponseFormatID	HCSC サーバからの応答電文フォーマットを一意に識別するための ID です。 このパラメタには NULL または空文字（" "）を指定してください。
オペレーション名	java.lang.String	serviceOperationName	要求先のサービス名に対するオペレーション名です。 このパラメタには NULL または空文字（" "）を指定してください。
ユーザ電文	java.lang.String	msg	サービスリクエストからの要求電文です。 このパラメタには NULL または空文字（" "）を指定してください。

### 注

これらのパラメタは、指定する値として固定値（「RECOVER」、NULL または空文字（" "））が決められています。決められた値以外の値を指定すると、正常に動作しません。

#### (4) 要求電文の作成

要求電文の作成方法は、標準の同期受付 (SessionBean) に要求を出す、通常のリクエストを作成する場合と同じです。

要求電文の作成については、「10.3.5 要求電文を作成する」を参照してください。

#### (5) 応答電文の取得

応答電文の作成方法は、標準の同期受付 (SessionBean) に要求を出す、通常のリクエストを作成する場合と同じです。

応答電文の取得については、「10.3.6 応答電文を取得する」を参照してください。

#### (6) エラー情報の取得

エラー情報を取得するためのサービスリクエストの実装は、標準の同期受付 (SessionBean) に要求を出す、通常のリクエストを作成する場合と同じです。

エラー情報の取得については、「10.3.7 エラー情報を取得する」を参照してください。

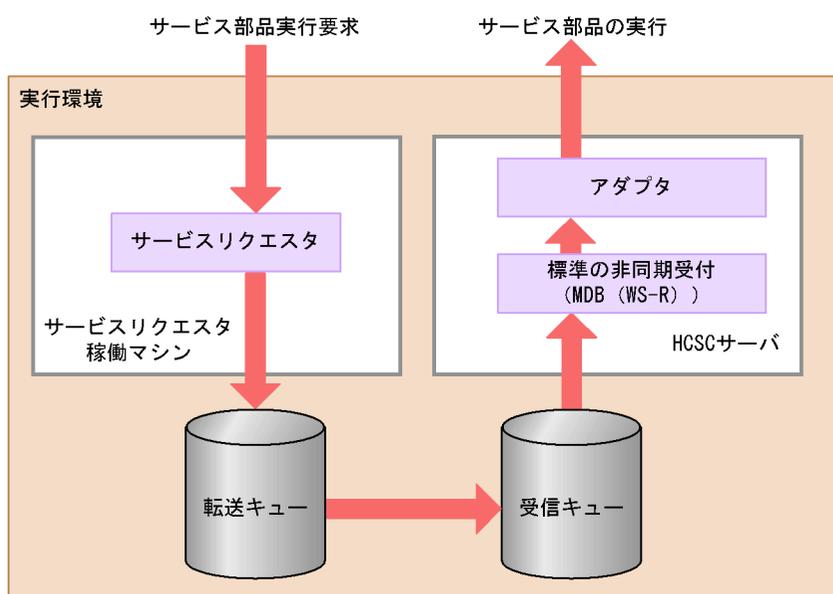
## 10.4 標準の非同期受付 (MDB (WS-R)) に要求を出すサービスリクエスト

標準の非同期受付 (MDB (WS-R)) へ要求電文を送信するサービスリクエストは、転送キューおよび受信キューを通して標準受付と通信を行います。

サービスリクエストから転送キューに送信された要求電文は、受信キューを通して非同期受付 (MDB (WS-R)) に送信されます。

標準の非同期受付 (MDB (WS-R)) に要求を出すサービスリクエストと HCSC サーバの関係を表す図に示します。

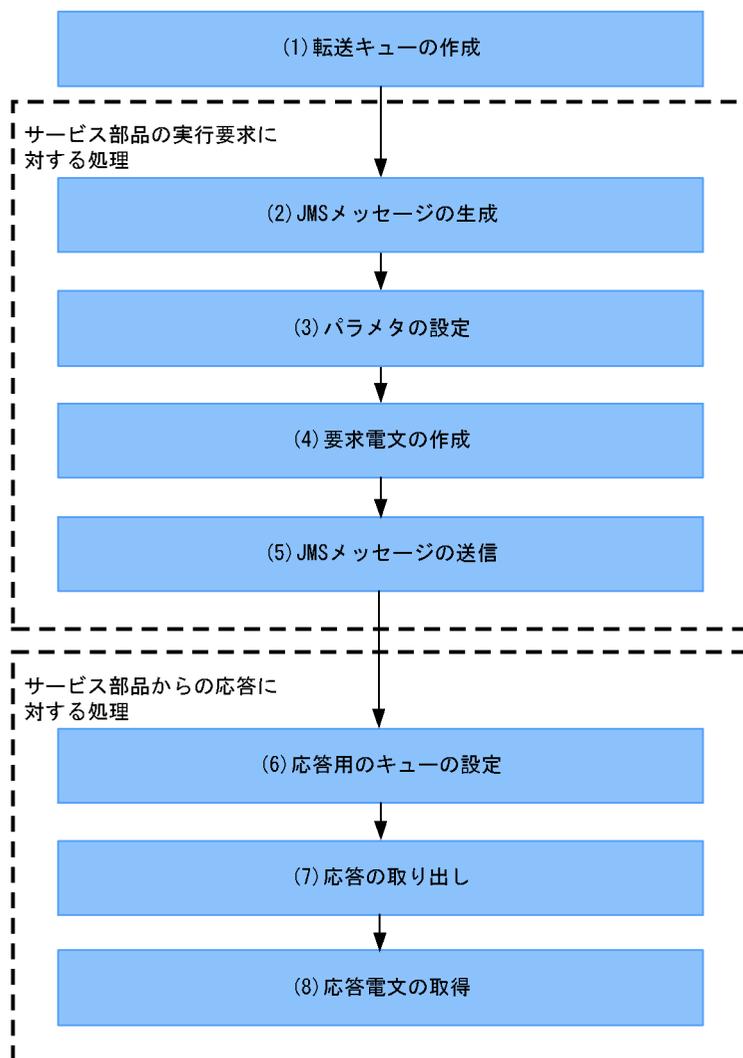
図 10-15 標準の非同期受付 (MDB (WS-R)) に要求を出すサービスリクエストと HCSC サーバの関係



### 10.4.1 サービスリクエスト (標準の非同期受付 (MDB (WS-R))) の作成手順

標準の非同期受付 (MDB (WS-R)) にサービス部品の実行を要求し、サービス部品を呼び出すためのサービスリクエストの作成手順を次に示します。

図 10-16 サービスリクエストの作成手順（標準の非同期受付（MDB（WS-R）））



各工程の作業を次に示します。

### （１）転送キューの作成

サービスリクエストから要求電文を送信するために必要な転送キューを作成します。転送キューの作成については、「10.4.2 転送キューを作成する」を参照してください。

### （２）JMS メッセージの生成

非同期受付（MDB（WS-R））にサービス部品の実行を要求するために、QueueSender のオブジェクトおよび JMS メッセージを生成します。QueueSender のオブジェクトおよび JMS メッセージの生成については、「10.4.3 JMS メッセージを生成する」を参照

## 10. サービスリクエストの作成

してください。

### (3) パラメタの設定

(2) で生成した JMS メッセージにパラメタを設定します。パラメタの設定については、「10.4.4 パラメタを設定する」を参照してください。

### (4) 要求電文の作成

サービス部品の実行を要求する要求電文を作成します。要求電文の作成については、「10.4.5 要求電文を作成する」を参照してください。

### (5) JMS メッセージの送信

JMS メッセージを転送キューに送信します。JMS メッセージの送信については、「10.4.6 JMS メッセージを送信する」を参照してください。

### (6) 応答用のキューの設定

同期のサービス部品から応答を受け取るために、応答用のキュー（転送キュー）を設定します。応答用のキューの設定については、「10.4.7 応答用のキューを設定する」を参照してください。

### (7) 応答の取り出し

応答用のキュー（転送キュー）から受信キューに送信された応答を取り出します。応答の取り出しについては、「10.4.8 応答を取り出す」を参照してください。

### (8) 応答電文の取得

取り出した応答から応答電文を取得します。応答電文の取得については、「10.4.9 応答電文を取得する」を参照してください。

## 10.4.2 転送キューを作成する

サービスリクエストから要求電文を送信するために必要な転送キューを作成します。転送キューは、サービスリクエストからの要求電文を、HCSC サーバが提供する受信キューに送信します。

転送キューの作成には、Cosminexus RM の `hrmmkaddr` コマンドおよび `hrmmkque` コマンドを使用します。

ここでは、`hrmmkaddr` コマンドおよび `hrmmkque` コマンドのオプションの指定値について説明します。ここで説明していないオプションについては、マニュアル「Cosminexus Reliable Messaging」を参照してください。

### (1) hrmmkaddr コマンドのオプションの指定値

hrmmkaddr コマンドのオプションの指定値を次に示します。

#### -u 宛先アドレス

HCSC サーバの宛先アドレス (転送先システムのキュー間転送用 Web アプリケーションの URL) を指定します。

```
http://<ホスト名>:<ポート番号> 1/<コンテキストルート> 2/services/HRMReceiver/
https://<ホスト名>:<ポート番号> 1/<コンテキストルート> 2/services/HRMReceiver/
```

#### 注 1

HCSC サーバの URL です。

#### 注 2

受信側 Cosminexus RM のキュー間転送用 Web アプリケーションのコンテキストルートです。デフォルトは「uCosminexusRM」です。

### (2) hrmmkque コマンドのオプションの指定値

#### -t キューの種類

transmit (転送キュー) を指定します。

#### -m キューの永続性

persistent (永続キュー属性) を指定します。

#### -v 転送先キュー名

次のどちらかを指定します。

CSC<HCSC サーバ名>ACPT\_RCVQ

HCSC サーバがロードバランスクラスタ構成の場合に指定します。

CSC<HCSC クラスタ名>ACPT\_RCVQ

HCSC サーバが HA クラスタ構成の場合に指定します。

HCSC クラスタ名は、開発環境および運用環境で HCSC サーバ構成配備時に決定される HCSC サーバのクラスタ名です。

#### -i キューの転送モード

Compatible (互換モード) を指定します。

## 10.4.3 JMS メッセージを生成する

標準の非同期受付 (MDB (WS-R)) にサービス部品の実行を要求するためには、QueueSender のオブジェクトおよび JMS メッセージを生成し、生成した JMS メッセージを転送キューに送信します。JMS メッセージを送信するアプリケーションの開発につ

## 10. サービスリクエストの作成

いては、マニュアル「Cosminexus Reliable Messaging」を参照してください。

JMS メッセージを生成する手順を次に示します。

1. QueueConnectionFactory と Queue の開始処理 (lookup) を定義します。

(例)

```
Context ic = new InitialContext();
:
QueueConnectionFactory qcFactory
    = (QueueConnectionFactory) ic.lookup("java:comp/env/jms/qcf");
:
Queue queue = (Queue) ic.lookup("java:comp/env/jms/queue");
:
```

2. QueueSession を生成します。

(例)

```
QueueSession qSession
    = qConnection.createQueueSession(true, Session.AUTO_ACKNOWLEDGE);
```

3. QueueSession から QueueSender を作成します。

(例)

```
qSender = qSession.createSender(queue);
```

4. QueueSession から JMS メッセージを生成します。

(例) 要求電文が XML の場合

```
TextMessage textMessage = qSession.createTextMessage();
```

(例) 要求電文がバイナリの場合

```
BytesMessage bytesMessage = qSession.createBytesMessage();
```

### 10.4.4 パラメタを設定する

非同期受付 (MDB (WS-R)) のメソッドを呼び出すため、JMS メッセージにパラメタを設定します。

## (1) プロパティの設定

JMS メッセージのプロパティを設定します。設定するプロパティの詳細については、「10.4.4(4) パラメタの詳細」を参照してください。

(例) 要求電文が XML の場合

```

textMessage = qSession.createTextMessage();
textMessage.setStringProperty("CSCServiceName", serviceName);
// サービス部品名
textMessage.setStringProperty("CSCCorrelationID", clientId);
// クライアント関連ID
textMessage.setStringProperty("CSCRequestFormatID", requestFormatID);
// リクエストフォーマットID
textMessage.setStringProperty("CSCResponseFormatID", responseFormatID);
// レスポンスフォーマットID
textMessage.setStringProperty("CSCServiceOperationName", operationName);
// オペレーション名
textMessage.setStringProperty("CSCReplyToQueueName", replyToQueueName);
// 応答用のキュー名
textMessage.setStringProperty("CSCMessageType", "XML");
// 電文種別

```

### ! 注意事項

バイナリ電文の要求電文を送信できるのは、サービス部品側で使用している電文形式がバイナリ形式のときだけです。DB キューのサービス部品に送信する要求電文はバイナリ変換します。バイナリ変換する場合は、サービスリクエスト側とサービス部品側で同じエンコードの方式を使用してください。

## (2) 引き継がれるヘッダおよびプロパティ

次のヘッダおよびプロパティは、要求時の内容がサービス部品側のキューまで引き継がれます。したがって、必要に応じて次のヘッダおよびプロパティを設定してください。

- JMSReplyTo
- JMSCorrelationID <sup>1</sup>
- JMSType
- JMSXGroupID
- JMSXGroupSeq
- ユーザ固有のプロパティ <sup>2</sup>

注 1

指定できる範囲は、255 バイトまでです。

注 2

ただし、次のプロパティはサービス部品側のキューまで引き継がれません。

- JMSX で始まる JMS 定義によるプロパティ名
- JMS\_ で始まるプロバイダ指定のプロパティ名

## 10. サービスリクエストの作成

- CSC で始まるプロパティ名
- HCSC で始まるプロパティ名

### (3) ペイロードの設定

JMS メッセージのペイロードに要求電文 (ユーザ電文) を設定します。ペイロードの詳細については、「10.4.4(4) パラメタの詳細」を参照してください。

(例) 要求電文が XML の場合

```
textMessage.setText( userData );
```

### (4) パラメタの詳細

パラメタの詳細を次の表に示します。

表 10-9 パラメタの詳細 (標準の非同期受付 (MDB (WS-R)))

パラメタ名	データ型	パラメタ		プロパティ / ペイロード	説明
		TextMessage	BytesMessage		
サービス名	String	StringProperty ("CSCServiceName")		プロパティ	要求先のサービス名です。このパラメタは必ず指定してください。要求先のサービス名は、開発環境で定義したアダプタまたはビジネスプロセスを指定してください。
クライアント関連 ID	String	StringProperty ("CSCCorrelationID")		プロパティ	サービスリクエストからの要求電文を一意に識別するための関連識別子です。255 文字以内の英数字、アンダーライン (_), ビリオド (.) およびハイフン (-) で指定してください。サービスリクエストからの要求電文と、HCSC サーバで管理している実行履歴やログ・トレースを対応づけるために使用します。したがって、HCSC サーバに送信した要求電文ごとに異なる ID を指定してください。なお、このパラメタを指定しないことで、クライアント関連 ID を省略できます。

パラメタ名	データ型	パラメタ		プロパティ / ペイロード	説明
		TextMessage	BytesMessage		
リクエストフォーマット ID	String	StringProperty ("CSCRequestFormatID")		プロパティ	サービスリクエストからの要求電文フォーマットを一意に識別するための ID です。 1,024 文字以内の英数字, アンダーライン (_), ピリオド (.) およびハイフン (-) で指定してください。 このパラメタには NULL を指定してください。なお, このパラメタを指定しないことで, NULL が設定されます。
レスポンスフォーマット ID	String	StringProperty ("CSCResponseFormatID")		プロパティ	HCSC サーバからの応答電文フォーマットを一意に識別するための ID です。 1,024 文字以内の英数字, アンダーライン (_), ピリオド (.) およびハイフン (-) で指定してください。 このパラメタには NULL を指定してください。なお, このパラメタを指定しないことで, NULL が設定されます。
オペレーション名	String	StringProperty ("CSCServiceOperationName")		プロパティ	要求先のサービス名に対するオペレーション名です。 オペレーション名は, 開発環境で定義したサービス部品のオペレーション名を指定します。 255 バイト以内の XMLSchema の NCName 定義文字で指定してください。 要求先のサービス部品が同期のサービス部品 (Web サービスまたは SessionBean) またはビジネスプロセスの場合, 必ず指定してください。 要求先のサービス部品が非同期のサービス部品の場合, オペレーション名を省略できます。 このパラメタを指定しないことで, オペレーション名を省略できます。

## 10. サービスリクエストの作成

パラメタ名	データ型	パラメタ		プロパティ / ペイロード	説明
		TextMessage	BytesMessage		
応答用のキュー名	String	StringProperty ("CSCReplyToQueueName")		プロパティ	<p>要求先のサービス部品が同期 ( Web サービスまたは SessionBean ) またはビジネスプロセスの場合に、サービス部品またはビジネスプロセスからの応答を受け取るキュー名です。要求先のサービス部品が同期か非同期かわからない場合にこの指定をすると、応答を受け取ることができます。</p> <p>応答用のキューの詳細については、「10.4.7 応答用のキューを設定する」を参照してください。なお、このパラメタを指定しないことで、応答用のキュー名を省略できます。省略した場合、NULL が設定され、応答を受け取ることができません。</p>
電文種別	String	StringProperty ("CSCMessageType")		プロパティ	<p>次に示す要求電文の種別を指定してください。</p> <ul style="list-style-type: none"> <li>XML 電文 ( ペイロードが TextMessage ) の場合: XML</li> <li>バイナリ電文 ( ペイロードが BytesMessage ) の場合: Binary</li> </ul> <p>要求電文がある場合、このパラメタは必ず指定してください。要求電文がない場合、このパラメタを指定しないことで、電文種別を省略することができます。</p>
ユーザ電文	-	TextMessage	x	ペイロード	<p>サービスリクエストからの要求電文です。要求電文が XML の場合に指定します。要求電文がない場合、指定する必要はありません。要求電文の詳細については、「10.4.5 要求電文を作成する」を参照してください。</p>
ユーザ電文長	long	x	LongProperty ("CSCMessageLength")	プロパティ	<p>要求電文の長さです。要求電文がバイナリの場合に指定します。</p>

パラメタ名	データ型	パラメタ		プロパティ / ペイロード	説明
		TextMessage	BytesMessage		
ユーザ電文	-	x	BytesMessage	ペイロード	サービスリクエストからの要求電文です。 要求電文がバイナリの場合に指定します。要求電文がない場合、指定する必要はありません。要求電文の詳細については、「10.4.5 要求電文を作成する」を参照してください。

(凡例)

- : 該当しません。
- x : 指定できません。

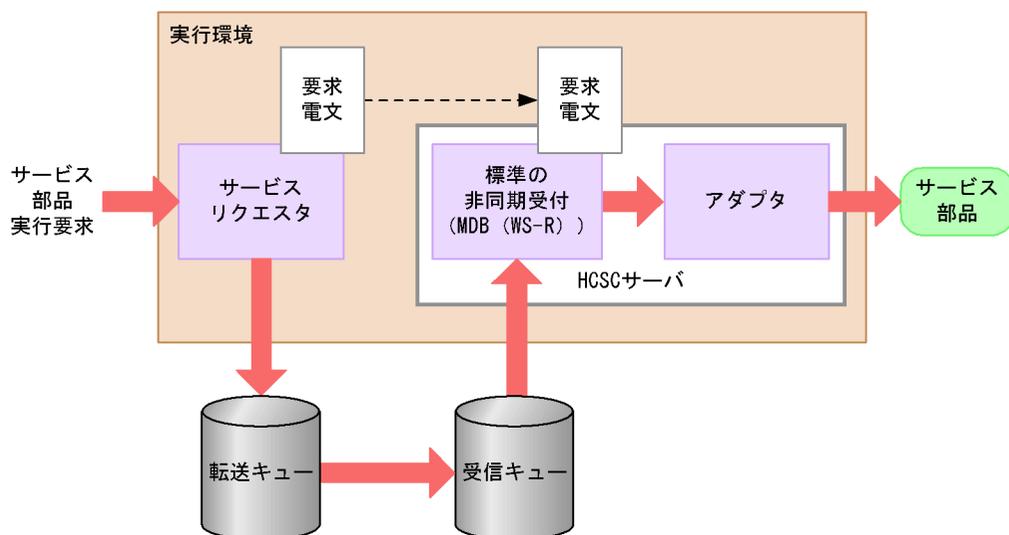
注

要求先のサービス部品のプロトコルが SOAP の場合、ユーザ電文のルート要素の名称（データ変換を行う場合、データ変換後の電文のルート要素の名称）から呼び出すオペレーションを決定します。そのため、ユーザ電文のルート要素に誤った名称を指定した場合、意図しないオペレーションを呼び出してしまうので注意してください。

### 10.4.5 要求電文を作成する

サービスリクエストから HCSC サーバの非同期受付 (MDB (WS-R)) に、サービス部品を要求する要求電文を作成します。サービスリクエストから送信する要求電文の内容は、サービス部品側と同じ電文フォーマットで作成してください。要求電文の送信を次の図に示します。

図 10-17 要求電文の送信（標準の非同期受付（MDB（WS-R）））



また、サービスリクエストから非同期受付（MDB（WS-R））に送信する要求電文は、電文種別に XML 電文とバイナリ電文の 2 種類があります。どちらの電文種別を使用するかは、サービス部品側で使用しているプロトコルによって異なります。

XML 電文およびバイナリ電文の要求電文については、「10.2.6 要求電文を作成する」を参照してください。

正常な要求電文の伝わり方については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の MDB（WS-R）を使ったサービス呼び出しに関する内容を参照してください。

## 10.4.6 JMS メッセージを送信する

パラメタを設定した JMS メッセージを転送キューに送信します。

JMS メッセージを送信する手順を次に示します。

1. ローカルトランザクションの開始および JMS メッセージの登録を行います。

（例）

```
qSender.send(message);
```

2. 終了処理を定義します。

(例)

```

qSession.commit();           // ローカルトランザクションのコミット
qSender.close();            // QueueSenderの解放
qSession.close();          // QueueSessionの解放
qConnection.close();       // QueueConnectionの解放

```

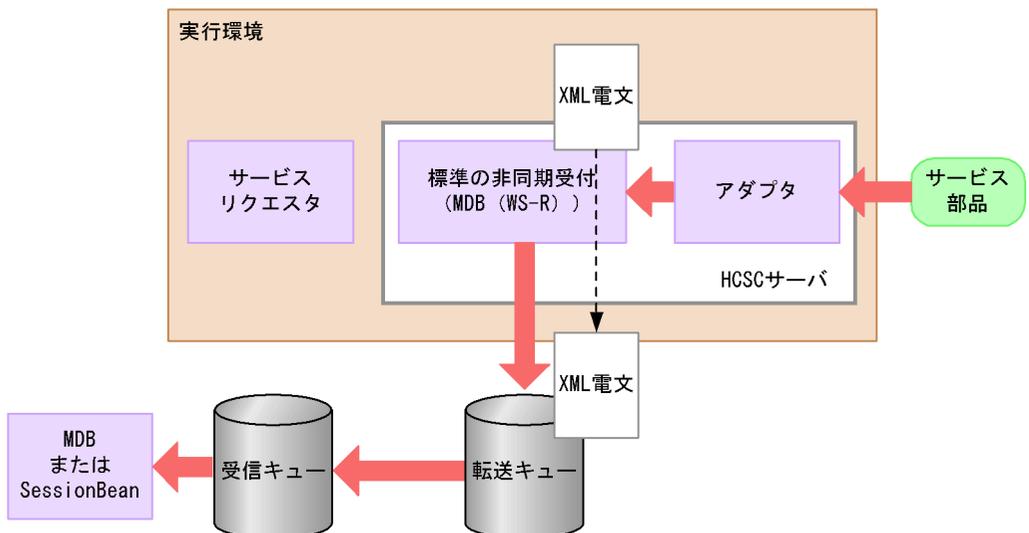
### 10.4.7 応答用のキューを設定する

標準の非同期受付 (MDB (WS-R)) は、非同期の要求電文の標準受付であり、基本的にはサービス部品からの応答がない場合に使用します。しかし、非同期受付 (MDB (WS-R)) から同期のサービス部品 (Web サービスまたは SessionBean) を要求した場合、同期のサービス部品から応答が返るときがあります。このような場合、要求時に応答用のキュー (転送キュー) を指定することで、サービスリクエストは同期のサービス部品から応答を受け取ることができます。要求時の応答用のキューの指定については、「10.4.4 パラメタを設定する」を参照してください。

標準の非同期受付 (MDB (WS-R)) は、同期サービス部品からの応答である XML 電文 (ペイロードが TextMessage) を転送キューに送信します。転送キューで受け取った応答は、受信キューに送信します。受信キューに送信した応答を取り出すことで、サービス部品からの応答電文を取得できます。応答の取り出しについては、「10.4.8 応答を取り出す」を参照してください。

標準の非同期受付 (MDB (WS-R))、転送キュー、および受信キューの関係を次に示します。

図 10-18 標準の非同期受付 (MDB (WS-R))、転送キュー、および受信キューの関係



## 10. サービスリクエストの作成

### 転送キュー（応答用のキュー）

応答用のキューとなる転送キューを、HCSC サーバが稼働している J2EE サーバ（Cosminexus RM）内に作成します。

転送キューの転送先（宛先アドレス）には、受信キューを指定します。転送先以外の転送キューの設定は、ユーザの運用に合わせ、必要に応じて行ってください。

転送キューの作成方法については、「10.4.2 転送キューを作成する」、およびマニュアル「Cosminexus Reliable Messaging」の転送キューに関する内容を参照してください。

### 受信キュー（ローカルキュー）

サービスリクエストが稼働している J2EE サーバ（Cosminexus RM）内に、受信キュー（ローカルキュー）を作成します。

受信キューの設定は、ユーザの運用に合わせ、必要に応じて行ってください。

受信キューの作成方法については、マニュアル「Cosminexus Reliable Messaging」のローカルキューに関する内容を参照してください。

## 10.4.8 応答を取り出す

受信キューから応答を取り出すには、MDB で実装する方法と SessionBean で実装する方法があります。正常な応答電文の伝わり方については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の MDB（WS-R）を使ったサービス呼び出しの ReplyTo キューに関する内容を参照してください。

### （1）応答の内容

サービス部品からは次の内容が応答として送信されます。

- 電文種別（StringProperty プロパティ（"CSCMessageType"））  
応答電文の電文種別は XML 電文のため、StringProperty プロパティには「XML」が設定されています。
- サービス部品からの応答電文（ペイロード（TextMessage））  
XML 電文（TextMessage）の応答電文が設定されています。  
同期のサービス部品を実行したがその応答電文がない場合は、ペイロードなしの TextMessage が送信されます。

### （2）引き継がれるヘッダおよびプロパティ

次のヘッダおよびプロパティは、サービスリクエストからサービス部品を要求したときに設定した内容が、応答用のキューを通し、受信キューまで引き継がれます。

#### （a）HCSC サーバ関連のプロパティ

- StringProperty（"CSCServiceName"）  
サービスリクエストが指定した、要求先のサービス名です。
- StringProperty（"CSCCorrelationID"）  
サービスリクエストが指定した、サービスリクエストからの要求電文を一意に識別す

るための相関識別子です。

- StringProperty ("CSCResponseFormatID")  
NULL が設定されています。
- StringProperty ("CSCServiceOperationName")  
サービスリクエストが指定した、要求先のサービス名に対するオペレーション名です。

(b) Cosminexus RM 関連のヘッダ、およびプロパティ

- JMSReplyTo
- JMSCorrelationID
- JMSType
- JMSXGroupID
- JMSXGroupSeq
- ユーザ固有のプロパティ名称

### 10.4.9 応答電文を取得する

取り出した応答から応答電文を取得します。取得する応答電文は、次のどちらかになります。

- サービス部品からの応答電文
- サービス部品、ビジネスプロセス、およびカスタムアダプタからのエラー情報（フォルト情報）

#### (1) サービス部品からの応答電文

取得する応答電文は、サービス部品側と同じ電文フォーマットの XML 電文になります。XML 電文の応答電文については、「10.2.7 応答電文を取得する」を参照してください。

同期のサービス部品（Web サービスまたは SessionBean）からの応答電文がない場合は、ペイロードなしの TextMessage を受信します。

非同期のサービス部品の場合は、応答がありません。

#### (2) サービス部品、ビジネスプロセス、およびカスタムアダプタからのエラー情報（フォルト情報）

取得する電文（エラー情報）は、非同期受付（MDB（WS-R））が応答用のキューに送信した XML 電文になります。取得する XML 電文（エラー情報）の形式を次に示します。

表 10-10 取得する XML 電文（エラー情報）の形式

タグ	説明
errorcode	サービス部品、ビジネスプロセス、およびカスタムアダプタからのエラーであることを示すエラーコードです。
errorstring	サービス部品、ビジネスプロセス、およびカスタムアダプタからのエラーであることを示すエラーメッセージです。

## 10. サービスリクエストの作成

タグ	説明
processinstanceid	ビジネスプロセスのインスタンス ID の情報です。
cscmsgcode	サービス部品 (Web サービス), ビジネスプロセス, またはカスタムアダプタからの FaultCode 情報です。
cscmsgstring	サービス部品 (Web サービス), ビジネスプロセス, またはカスタムアダプタからの FaultString 情報です。
cscmsgactor	サービス部品 (Web サービス), ビジネスプロセス, またはカスタムアダプタからの FaultActor 情報です。
cscmsgdetail	サービス部品 (Web サービス), ビジネスプロセス, またはカスタムアダプタからの Detail 情報です。

取得する XML 電文 (エラー情報) のスキーマを次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="cscmsgerror">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="errorcode" type="xsd:string"/>
        <xsd:element name="errorstring" type="xsd:string"/>
        <xsd:element name="processinstanceid" type="xsd:string"/>
        <xsd:element ref="errordetail"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="errordetail">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="cscmsgcode" type="xsd:string"/>
        <xsd:element name="cscmsgstring" type="xsd:string"/>
        <xsd:element name="cscmsgactor" type="xsd:string"/>
        <xsd:element name="cscmsgdetail" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

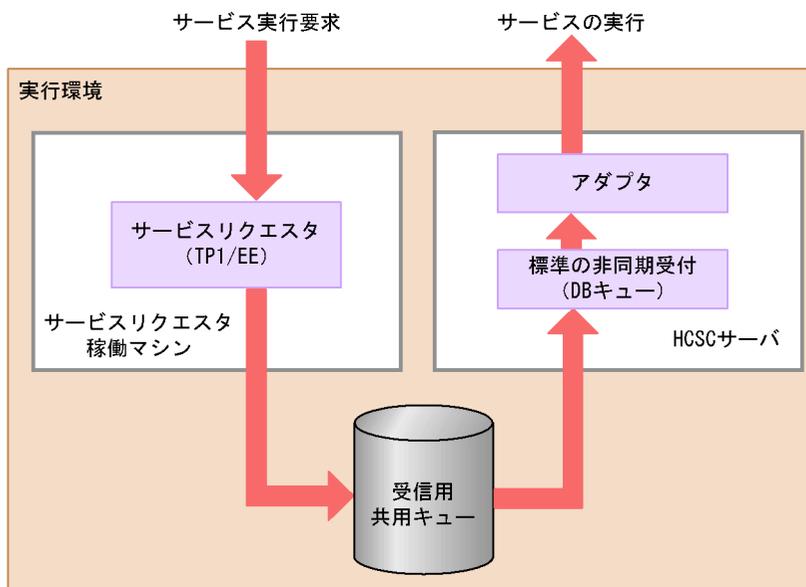
## 10.5 標準の非同期受付（DB キュー）に要求を出すサービスリクエスト

サービス部品を呼び出す HCSC サーバの標準の非同期受付（DB キュー）に要求を出すサービスリクエストは、TP1/Server Base Enterprise Option（TP1/EE）と JMS の二つがあります。

TP1/Server Base Enterprise Option（TP1/EE）のサービスリクエスト

サービスリクエスト（TP1/EE）とサービス部品を呼び出す HCSC サーバの標準の非同期受付（DB キュー）の間は、受信用共用キューを通して通信を行います。標準の非同期受付（DB キュー）に要求を出すサービスリクエスト（TP1/EE）と HCSC サーバの関係を次の図に示します。

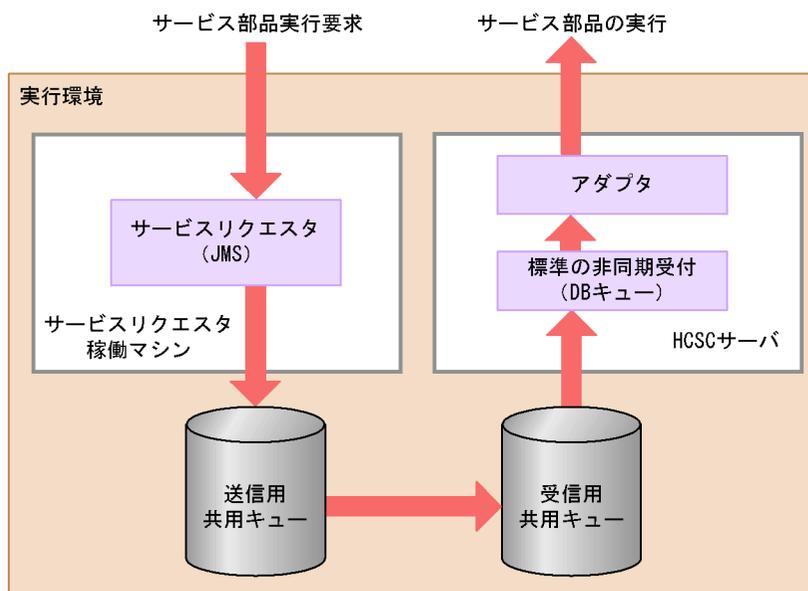
図 10-19 標準の非同期受付（DB キュー）に要求を出すサービスリクエスト（TP1/EE）と HCSC サーバの関係



JMS のサービスリクエスト

サービスリクエスト（JMS）とサービス部品を呼び出す HCSC サーバの標準の非同期受付（DB キュー）の間は、送信用共用キューおよび受信用共用キューを通して通信を行います。標準の非同期受付（DB キュー）に要求を出すサービスリクエスト（JMS）と HCSC サーバの関係を次の図に示します。

図 10-20 非同期受付 (DB キュー) に要求を出すサービスリクエスト (JMS) と HCSC サーバの関係



### 10.5.1 サービスリクエスト (標準の非同期受付 (DB キュー)) の作成手順

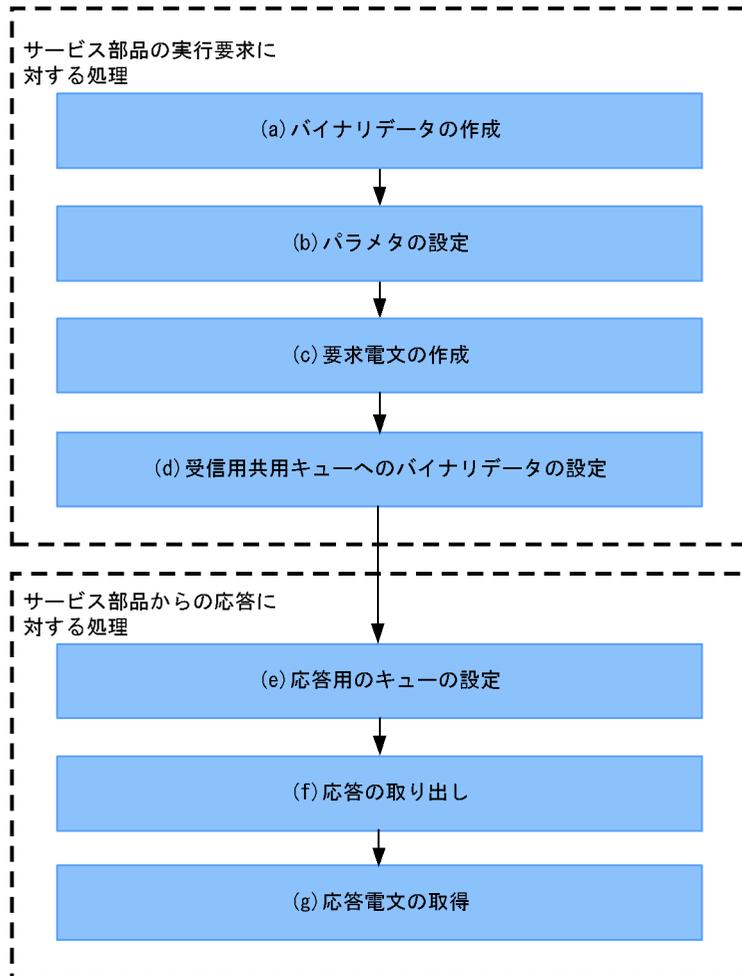
標準の非同期受付 (DB キュー) にサービス部品の実行を要求し、サービス部品を呼び出すためのサービスリクエストの作成手順を次に示します。

作成手順は、サービスリクエストが TP1/EE の場合と JMS の場合で異なります。

#### (1) TP1/EE の場合

サービスリクエストが TP1/EE の場合に、標準の非同期受付 (DB キュー) にサービス部品の実行を要求し、サービス部品を呼び出すためのサービスリクエストの作成手順を次に示します。

図 10-21 サービスリクエスト (TP1/EE) の作成手順 (標準の非同期受付 (DB キュー))



(a) バイナリデータの作成

標準の非同期受付 (DB キュー) に要求電文を送信する場合、バイナリデータを作成する必要があります。バイナリデータの作成については、「10.5.4 バイナリデータを作成する (TP1/EE, JMS)」を参照してください。

(b) パラメタの設定

(a) で作成したバイナリデータの各タグに対応するパラメタを設定して、バイナリデータを組み立てます。パラメタの設定については、「10.5.5 パラメタを設定する (TP1/EE, JMS)」を参照してください。

## 10. サービスリクエストの作成

### (c) 要求電文の作成

サービス部品の実行を要求する要求電文を作成します。要求電文の作成については、「10.5.6 要求電文を作成する (TP1/EE, JMS)」を参照してください。

### (d) 受信用共用キューへのバイナリデータの設定

パラメタを設定したバイナリデータを TP1/EE の API (DB キュー制御) を使用して、受信用共用キューがあるデータベースに設定します。受信用共用キューへのバイナリデータの設定については、「10.5.7 バイナリデータを受信用共用キューに設定する (TP1/EE)」を参照してください。

### (e) 応答用のキューの設定

同期のサービス部品 (Web サービスまたは SessionBean) から応答を受け取るために、応答用のキュー (受信用共用キュー) を設定します。応答用のキューの設定については、「10.5.9 応答用のキューを設定する (TP1/EE, JMS)」を参照してください。

### (f) 応答の取り出し

TP1/EE の API (DB キュー制御) を使用して、応答用のキュー (受信用共用キュー) があるデータベースを参照して、応答のバイナリデータを取り出します。応答の取り出しについては、「10.5.10 応答を取り出す (TP1/EE, JMS)」を参照してください。

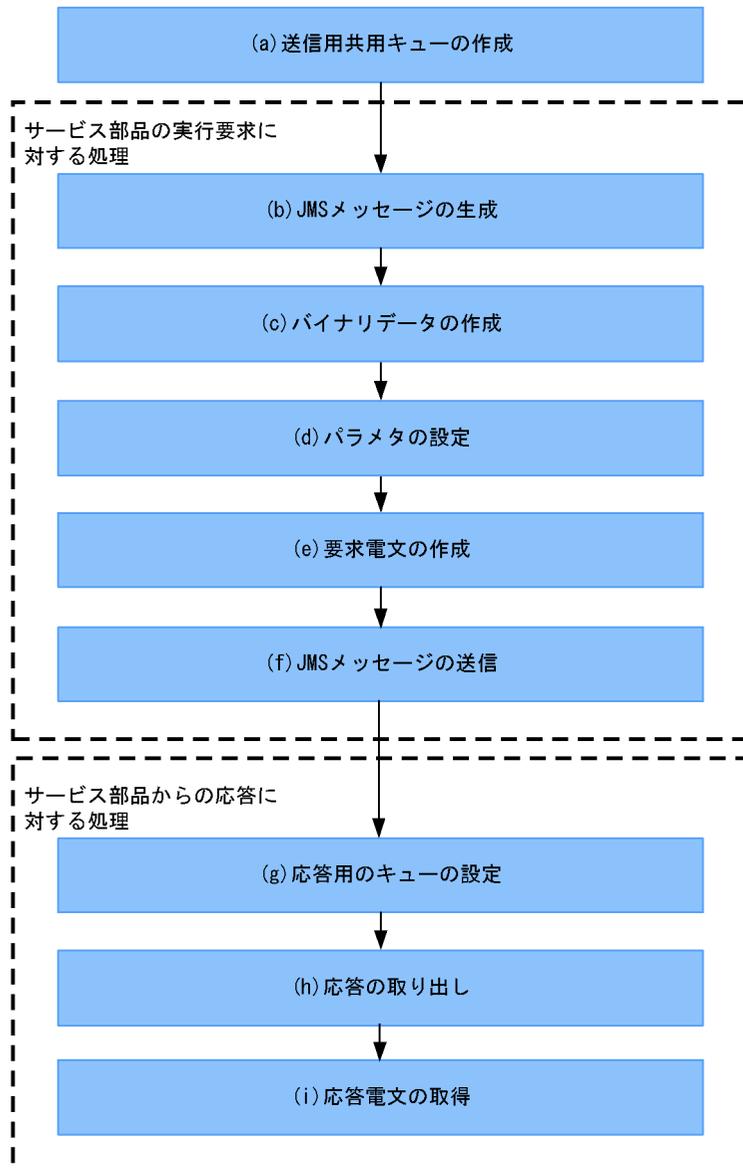
### (g) 応答電文の取得

取り出した応答から応答電文を取得します。応答電文の取得については、「10.5.11 応答電文を取得する (TP1/EE, JMS)」を参照してください。

## (2) JMS の場合

サービスリクエストが JMS の場合に、標準の非同期受付 (DB キュー) にサービス部品の実行を要求し、サービス部品を呼び出すためのサービスリクエストの作成手順を次に示します。

図 10-22 サービスリクエスト (JMS) の作成手順 (標準の非同期受付 (DB キュー))

**(a) 送信用共用キューの作成**

サービスリクエスト (JMS) から要求電文を送信するために必要な送信用共用キューを作成します。送信用共用キューの作成については、「10.5.2 送信用共用キューを作成する (JMS)」を参照してください。

**(b) JMS メッセージの生成**

非同期受付 (DB キュー) にサービス部品の実行を要求するために、QueueSender のオ

## 10. サービスリクエストの作成

プロジェクトおよび JMS メッセージを生成します。QueueSender のオブジェクトおよび JMS メッセージの生成については、「10.5.3 JMS メッセージを生成する (JMS)」を参照してください。

### (c) バイナリデータの作成

非同期受付 (DB キュー) に要求電文を送信する場合、バイナリデータを作成する必要があります。バイナリデータの作成については、「10.5.4 バイナリデータを作成する (TP1/EE, JMS)」を参照してください。

### (d) パラメタの設定

(c) で作成したバイナリデータの各タグに対応するパラメタを設定して、バイナリデータを組み立てます。パラメタの設定については、「10.5.5 パラメタを設定する (TP1/EE, JMS)」を参照してください。

### (e) 要求電文の作成

サービス部品の実行を要求する要求電文を作成します。要求電文の作成については、「10.5.6 要求電文を作成する (TP1/EE, JMS)」を参照してください。

### (f) JMS メッセージの送信

ペイロードにバイナリデータを設定した JMS メッセージを送信用共用キューに送信します。JMS メッセージの送信については、「10.5.8 JMS メッセージを送信する (JMS)」を参照してください。

### (g) 応答用のキューの設定

同期のサービス部品 (Web サービスまたは SessionBean) から応答を受け取るために、応答用のキュー (送信用共用キュー) を設定します。応答用のキューの設定については、「10.5.9 応答用のキューを設定する (TP1/EE, JMS)」を参照してください。

### (h) 応答の取り出し

応答用のキュー (送信用共用キュー) から受信用共用キューに送信された応答を取り出します。応答の取り出しについては、「10.5.10 応答を取り出す (TP1/EE, JMS)」を参照してください。

### (i) 応答電文の取得

取り出した応答から応答電文を取得します。応答電文の取得については、「10.5.11 応答電文を取得する (TP1/EE, JMS)」を参照してください。

## 10.5.2 送信用共用キューを作成する (JMS)

サービスリクエストから要求電文を送信するために必要な送信用共用キューを作成します。送信用共用キューは、サービスリクエストからの要求電文を、HCSC サーバが提供

する受信用共用キューに送信します。

送信用共用キューの作成には、Cosminexus RM の `hrmmkque` コマンドを使用します。ここでは、`hrmmkque` コマンドのオプションの指定値について説明します。ここで説明していないオプションについては、マニュアル「Cosminexus Reliable Messaging」を参照してください。

### (1) `hrmmkque` コマンドのオプションの指定値

-t キューの種類

`shr_send` (送信用共用キュー) を指定します。

-b 共用キューを使用する場合の登録先キュー名

次のどちらかを指定します。

< Cosminexus RM システム名 > `_SHR_CSC`<HCSC サーバ名 >`ACPT_DBQ`  
HCSC サーバがロードバランスクラスタ構成の場合に指定します。

< Cosminexus RM システム名 > `_SHR_CSC`<HCSC クラスタ名 >`ACPT_DBQ`  
HCSC サーバが HA クラスタ構成の場合に指定します。

HCSC クラスタ名は、開発環境および運用環境で HCSC サーバ構成配備時に決定される HCSC サーバのクラスタ名です。

## 10.5.3 JMS メッセージを生成する (JMS)

標準の非同期受付 (DB キュー) にサービス部品の実行を要求するためには、`QueueSender` のオブジェクトおよび JMS メッセージを生成し、送信用共用キューに送信します。JMS メッセージを送信するアプリケーションの開発については、マニュアル「Cosminexus Reliable Messaging」を参照してください。

生成した JMS メッセージのペイロードには、パラメタを設定したバイナリデータを指定します。なお、JMS メッセージは `BytesMessage` で生成し、パラメタを設定したバイナリデータを `BytesMessage` のペイロードに指定してください。要求電文がない場合でも、バイナリデータを `BytesMessage` のペイロードに指定してください。

また、JMS メッセージのプロパティを設定しても、プロパティの値は無視されます。

JMS メッセージを生成する手順については、「10.4.3 JMS メッセージを生成する」を参照してください。

## 10.5.4 バイナリデータを作成する (TP1/EE, JMS)

標準の非同期受付 (DB キュー) に要求電文を送信する場合、次の表に示す形式のバイナリデータを作成する必要があります。

## 10. サービスリクエストの作成

表 10-11 バイナリデータの形式（要求電文）

項番	項目	長さ (バイト)	データ型 <sup>1</sup>	説明	必須
1	ヘッダタグ	4	char	サービス部品要求のメッセージ振り分け用の先頭タグです（アイキャッチャ）。 「DBQH」を指定します。 ASCII コードを使用してください。	
2	バイトオーダーフラグ	1	byte	ヘッダ内の数値データのバイトオーダー（エンディアン）識別フラグです。 ヘッダ内の数値データをバイナリにエンコードする方式を次のどちらかから指定します。 • 0b00000001：ビッグエンディアン • 0b00000010：リトルエンディアン	
3	予備	3	byte	予備領域です。	
4	長さ	4	int	サービス部品要求のメッセージングのヘッダの長さ（項番 5 から項番 22 までの長さ）です。 バイト単位で指定してください。	
5	タグ	8	char	パラメタ識別用のタグです（アイキャッチャ）。 「ServiceN」を指定します。 ASCII コードを使用してください。	
6	長さ	4	int	次のエリア（項番 7）の長さです。 バイト単位で指定してください。	
7	サービス名	任意	String	要求先のサービス名です。 要求先のサービス名は、開発環境で定義したアダプタまたはビジネスプロセスを指定してください。 UTF-8 コードを使用してください。	
8	タグ	8	byte	パラメタ識別用のタグです（アイキャッチャ）。 「ClientID」を指定します。 ASCII コードを使用してください。	
9	長さ	4	int	次のエリア（項番 10）の長さです。 バイト単位で指定してください。	
10	クライアント関連 ID	任意	String	サービスリクエストからの要求電文を一意に識別するための関連識別子です。 255 文字以内の英数字、アンダーライン（_）、ピリオド（.）およびハイフン（-）で指定してください。 サービスリクエストからの要求電文と、HCSC サーバで管理している実行履歴やログ・トレースを対応づけるために使用します。したがって、HCSC サーバに送信した要求電文ごとに異なる ID を指定してください。 UTF-8 コードを使用してください。 省略する場合、このパラメタ、長さ、およびタグの、バイナリデータを組み立てる必要はありません。	
11	タグ	8	byte	パラメタ識別用のタグです（アイキャッチャ）。 「ReqFmtID」を指定します。 ASCII コードを使用してください。	×

項番	項目	長さ (バイト)	データ型 <sup>1</sup>	説明	必須
12	長さ	4	int	次のエリア (項番 13) の長さです。 バイト単位で指定してください。	
13	HCSC リクエスト フォーマット ID	任意	String	要求電文のフォーマット定義を一意に識別するための ID です。 このパラメタ, 長さ, およびタグのバイナリデータは組み立てないでください。	
14	タグ	8	byte	パラメタ識別用のタグです (アイキャッチャ)。 「ResFmtID」を指定します。 ASCII コードを使用してください。	×
15	長さ	4	int	次のエリア (項番 16) の長さです。 バイト単位で指定してください。	
16	HCSC レスポンス フォーマット ID	任意	String	応答電文のフォーマット定義を一意に識別するための ID です。 このパラメタ, 長さ, およびタグのバイナリデータは組み立てないでください。	
17	タグ	8	byte	パラメタ識別用のタグです (アイキャッチャ)。 「ReplyToQ」を指定します。 ASCII コードを使用してください。	
18	長さ	4	int	次のエリア (項番 19) の長さです。 バイト単位で指定してください。	
19	応答用の キュー名	任意	String	要求先のサービス部品が同期 (Web サービスまたは SessionBean) またはビジネスプロセスの場合に, サービス部品またはビジネスプロセスからの応答を受け取るキュー名です。 要求先のサービス部品が同期か非同期かわからない場合にこの指定をすると, 応答を受け取ることができません。 詳細については, 「10.5.9 応答用のキューを設定する (TP1/EE, JMS)」を参照してください。 UTF-8 コードを使用してください。 省略すると, NULL が設定され, 応答を受け取ることができません。省略する場合, このパラメタ, 長さ, およびタグの, バイナリデータを組み立てる必要はありません。	
20	タグ	8	byte	パラメタ識別用のタグです (アイキャッチャ)。 「OperatiN」を指定します。 ASCII コードを使用してください。	
21	長さ	4	int	次のエリア (項番 22) の長さです。 バイト単位で指定してください。	

## 10. サービスリクエストの作成

項番	項目	長さ (バイト)	データ型 <sup>1</sup>	説明	必須
22	サービスオペレーション名	任意	String	<p>要求先のサービス名に対するオペレーション名です。<sup>2</sup></p> <p>オペレーション名は、開発環境で定義したサービス部品のオペレーション名を指定します。255 バイト以内の XMLSchema の NCName 定義文字で指定してください。</p> <p>要求先のサービス部品が同期のサービス部品 (Web サービスまたは SessionBean) またはビジネスプロセスの場合、必ず指定してください。</p> <p>UTF-8 コードを使用してください。</p> <p>要求先のサービス部品が非同期のサービス部品の場合、応答用のキュー名を省略できます。省略する場合、このパラメタ、長さ、およびタグの、バイナリデータを組み立てる必要はありません。</p>	
23	タグ	8	byte	<p>パラメタ識別用のタグです (アイキャッチャ)。 「MessageT」を指定します。</p> <p>ASCII コードを使用してください。</p>	
24	長さ	4	int	<p>次のエリア (項番 25) の長さです。</p> <p>バイト単位で指定してください。</p>	
25	電文種別	任意	String	<p>ユーザ電文 (要求電文) の種別です。</p> <p>XML 電文 (ペイロードが TextMessage) の場合は 「XML」 を、バイナリ電文 (ペイロードが BytesMessage) の場合は 「Binary」 を指定します。</p> <p>メッセージ (要求電文) がある場合、必ず指定してください。</p> <p>UTF-8 コードを使用してください。</p> <p>要求電文がない場合、省略することができます。省略する場合、このパラメタ、長さ、およびタグの、バイナリデータを組み立てる必要はありません。</p>	
26	タグ	8	byte	<p>パラメタ識別用のタグです (アイキャッチャ)。 「MgLength」を指定します。</p> <p>ASCII コードを使用してください。</p>	
27	長さ	4	int	<p>次のエリア (項番 28) の長さです。</p> <p>「4」を指定してください。</p>	
28	メッセージ長	4	int	<p>バイナリ変換したユーザ電文 (要求電文) のメッセージの長さを指定します。</p> <p>ユーザ電文 (要求電文) がない場合、0 を指定してください。</p>	
29	メッセージ	任意	byte[]	<p>ユーザ電文 (要求電文) です。<sup>2</sup></p> <p>DB キューのヘッダ (項番 1 ~ 28) の後ろに、バイナリ変換したユーザ電文 (要求電文) を指定します。</p> <p>なお、XML 電文をバイナリに変換したメッセージの場合は、UTF-8 コードでエンコードしてください。</p> <p>ユーザ電文 (要求電文) がない場合、省略できます。</p>	

(凡例)

- : 必ず指定してください。
- : 必ず指定しなければいけない場合があります。説明の内容を確認してください。
- × : 指定しないでください。

注 1

Java のデータ型です。

注 2

要求先のサービス部品のプロトコルが SOAP の場合、ユーザ電文のルート要素の名称 (データ変換を行う場合、データ変換後の電文のルート要素の名称) から呼び出すオペレーションを決定します。そのため、ユーザ電文のルート要素に誤った名称を指定した場合、意図しないオペレーションを呼び出してしまいますので注意してください。

### 10.5.5 パラメタを設定する (TP1/EE, JMS)

作成したバイナリデータの各タグ (表 10-11 の項番 5, 8, 11, 17, 20, 23, および 26) に対応するパラメタを設定して、バイナリデータを組み立てます。

バイナリデータのタグの形式を次の図に示します。

図 10-23 バイナリデータのタグの形式



バイナリデータ中の各タグの出現順序は任意です。ただし、タグ、長さ、値の順序は、図 10-23 のように固定されています。

また、DB キューのヘッダ (項番 1 ~ 28) の後ろに、バイナリ変換したメッセージ (要求電文) を指定します。要求電文がない場合、項番 26 のタグに MgLength を指定し、その長さを 0 にすることもできます。

### 10.5.6 要求電文を作成する (TP1/EE, JMS)

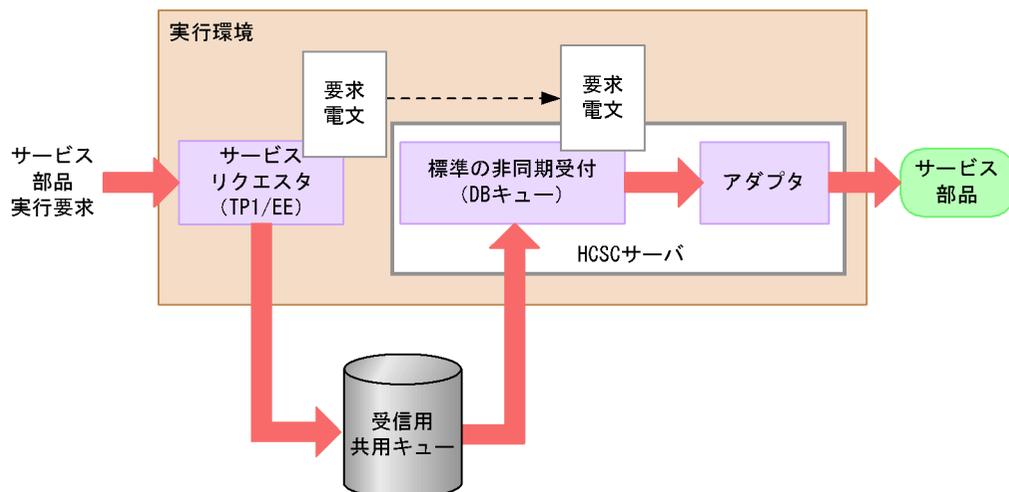
サービスリクエストから HCSC サーバの非同期受付 (DB キュー) に、サービス部品を要求する要求電文を作成します。サービスリクエストから送信する要求電文の内容は、サービス部品側と同じ電文フォーマットで作成してください。

サービスリクエストが TP1/EE の場合と JMS の場合の、要求電文の送信を次の図に示します。

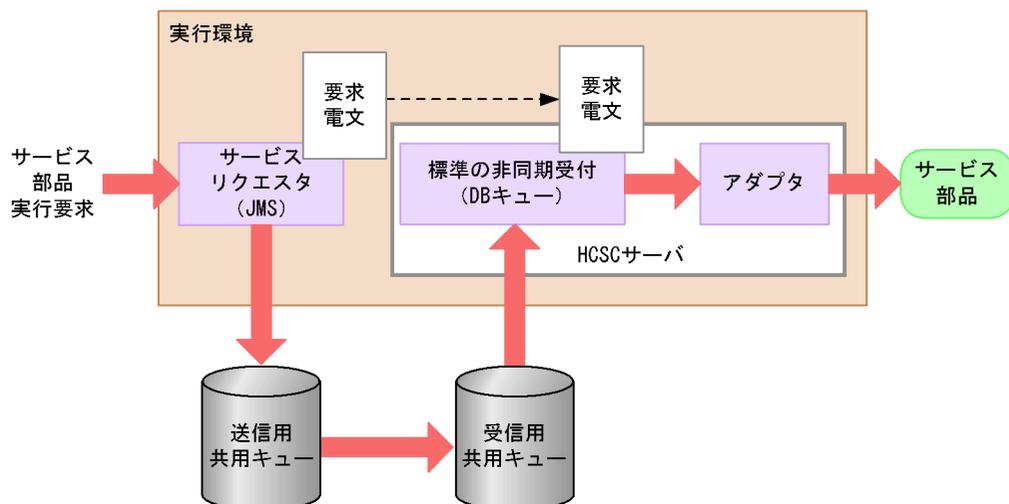
## 10. サービスリクエストの作成

図 10-24 要求電文の送信（標準の非同期受付（DB キュー））

### ●TP1/EEの場合



### ●JMSの場合



正常な要求電文の伝わり方については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」のMDB（DB キュー）を使ったサービス呼び出しに関する内容を参照してください。

また、サービスリクエストでは、電文種別がXML電文の要求電文を作成します。作成した要求電文はバイナリ変換します。バイナリ変換する場合は、サービスリクエスト側とサービス部品側で同じエンコードの方式を使用してください。XML電文の要求電文については、「10.2.6 要求電文を作成する」を参照してください。

なお、プロトコルの種類が DB キューのサービス部品の場合、非同期受付 (DB キュー) から要求電文を送信することはできません。非同期受付 (DB キュー) から要求電文を送信できるのは、サービス部品側で次のプロトコルを使用しているときになります。

- SOAP
- RMI-IIOP
- WS-R

### 10.5.7 バイナリデータを受信用共用キューに設定する (TP1/EE)

パラメタを設定したバイナリデータを TP1/EE の API (DB キュー制御) を使用して、受信用共用キューがあるデータベースに設定します。

詳細は、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

### 10.5.8 JMS メッセージを送信する (JMS)

ペイロードにバイナリデータを設定した JMS メッセージを、送信用共用キューに送信します。

JMS メッセージを送信する手順については、「10.4.6 JMS メッセージを送信する」を参照してください。

### 10.5.9 応答用のキューを設定する (TP1/EE, JMS)

標準の非同期受付 (DB キュー) は、非同期の要求電文の標準受付であり、基本的にはサービス部品からの応答がない場合に使用します。しかし、非同期受付 (DB キュー) から同期のサービス部品 (Web サービスまたは SessionBean) を要求した場合、同期のサービス部品から応答が返るときがあります。このような場合、要求時に応答用のキューを指定することで、サービスリクエストは同期のサービス部品から応答を受け取ることができます。要求時の応答用のキューの指定については、「10.4.6 JMS メッセージを送信する」を参照してください。

#### (1) TP1/EE の場合

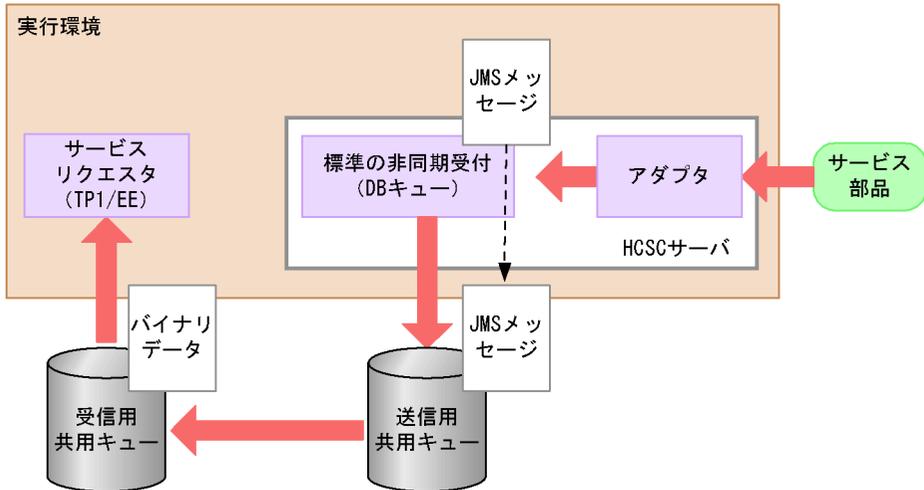
非同期受付 (DB キュー) は、応答電文を含むバイナリデータがペイロードに設定された JMS メッセージを応答用のキュー (送信用共用キュー) に送信します。

送信用共用キューは、受け取った応答からバイナリデータを取り出し、受信用共用キューに送信します。受信用共用キューに送信した応答を取り出して、サービス部品からの応答電文を取得できます。応答の取り出しについては、「10.5.10 応答を取り出す (TP1/EE, JMS)」を参照してください。

## 10. サービスリクエストの作成

標準の非同期受付（DB キュー）と送信用共用キュー，および受信用共用キューの関係を次に示します。

図 10-25 標準の非同期受付（DB キュー）と送信用共用キュー，および受信用共用キューの関係（TP1/EE の場合）



### 送信用共用キュー（応答用のキュー）

応答用のキューとなる送信用共用キューを，HCSC サーバが稼働している J2EE サーバ（Cosminexus RM）内に作成します。送信用共用キューの登録先キューには，受信用共用キューを指定します。

送信用共用キューの作成方法については，マニュアル「Cosminexus Reliable Messaging」の送信用共用キューに関する内容を参照してください。

### 受信用共用キュー

TP1/EE の DB キューを，送信用共用キューがある（Cosminexus RM が使用している）データベースの同一のスキーマ定義内に作成します。詳細は，マニュアル「Cosminexus Reliable Messaging」の受信用共用キューに関する内容を参照してください。

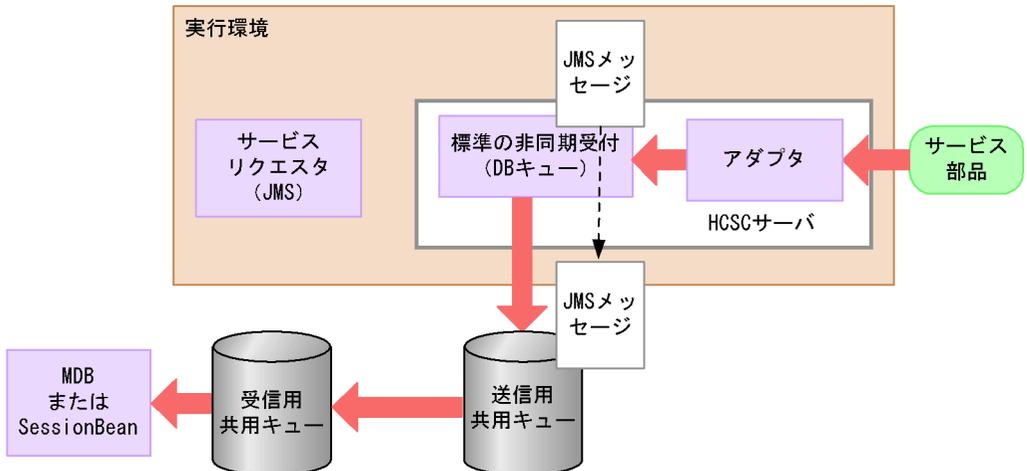
また，DB キューの作成の詳細については，マニュアル「TP1/Server Base Enterprise Option 使用の手引」の DB キューに関する内容を参照してください。

## （2）JMS の場合

標準の非同期受付（DB キュー）は応答として，応答電文を含むバイナリデータがペイロードに設定された JMS メッセージを応答用のキュー（送信用共用キュー）に送信します。送信用共用キューで受け取った応答は，受信用共用キューに送信します。受信用共用キューに送信した応答を取り出すことで，サービス部品からの応答電文を取得できます。応答の取り出しについては，「10.5.10 応答を取り出す（TP1/EE，JMS）」を参照してください。

標準の非同期受付（DB キュー）と送信用共用キュー，および受信用共用キューの関係を次に示します。

図 10-26 標準の非同期受付（DB キュー）と送信用共用キュー，および受信用共用キューの関係（JMS の場合）



#### 送信用共用キュー（応答用のキュー）

応答用のキューとなる送信用共用キューを，HCSC サーバが稼働している J2EE サーバ（Cosminexus RM）内に作成します。送信用共用キューの登録先キューには，受信用共用キューを指定します。

送信用共用キューの作成方法については，マニュアル「Cosminexus Reliable Messaging」の送信用共用キューに関する内容を参照してください。

#### 受信用共用キュー

サービスリクエストが稼働している J2EE サーバ（Cosminexus RM）内に，受信用共用キューを作成します。

受信用共用キューの設定は，ユーザの運用に合わせ，必要に応じて行ってください。

受信用共用キューの作成方法については，マニュアル「Cosminexus Reliable Messaging」の受信用共用キューに関する内容を参照してください。

## 10.5.10 応答を取り出す（TP1/EE，JMS）

### （1）TP1/EE の場合

TP1/EE の API（DB キュー制御）を使用して，応答用のキュー（受信用共用キュー）があるデータベースを参照して，応答のバイナリデータを取り出します。正常な応答電文の伝わり方については，マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」の MDB（DB キュー）を使ったサービス呼び出しの ReplyTo キューに関する内容を参照してください。

## 10. サービスリクエストの作成

詳細については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

応答のバイナリデータの形式を次の表に示します。

表 10-12 バイナリデータの形式（応答電文）

項番	項目	長さ (バイト)	データ型	説明
1	ヘッダタグ	4	char	HCSC メッセージ振り分け用の先頭タグです（アイキャッチャ）。 「DBQH」が設定されます。 ASCII コードでエンコードされているバイナリデータが設定されます。
2	バイトオーダーフラグ	1	byte	要求時に指定したバイトオーダーフラグが設定されます。
3	予備	3	byte	予備領域です。
4	長さ	4	int	HCSC メッセージング用のヘッダの長さ（項番 5 から項番 19 までの長さ）が設定されます。
5	タグ	8	char	パラメタ識別用のタグです（アイキャッチャ）。 「ServiceN」が設定されます。 ASCII コードでエンコードされているバイナリデータが設定されます。
6	長さ	4	int	次のエリア（項番 7）の長さです。 バイト単位で設定されます。
7	サービス名	任意	String	要求時に指定したアダプタまたはビジネスプロセスが設定されます。 UTF-8 コードでエンコードされているバイナリデータが設定されます。
8	タグ	8	byte	パラメタ識別用のタグです（アイキャッチャ）。 「ClientID」が設定されます。 ASCII コードでエンコードされているバイナリデータが設定されます。
9	長さ	4	int	次のエリア（項番 10）の長さです。 バイト単位で設定されます。
10	クライアント 相関 ID	任意	String	サービスリクエストからの要求電文を一意に識別するための 相関識別子です。 255 文字以内の英数字、アンダーライン（_）、ピリオド（.） およびハイフン（-）で設定されます。 UTF-8 コードでエンコードされているバイナリデータが設定 されます。
11	タグ	8	byte	パラメタ識別用のタグです（アイキャッチャ）。 「ResFmtID」を指定します。 ASCII コードを使用してください。
12	長さ	4	int	次のエリア（項番 13）の長さです。 バイト単位で指定してください。

項番	項目	長さ (バイト)	データ型	説明
13	HCSC レスポンスフォーマット ID	任意	String	応答電文のフォーマット定義を一意に識別するための ID です。 このパラメタ、長さ、およびタグは設定されません。
14	タグ	8	byte	パラメタ識別用のタグです (アイキャッチャ)。 「OperatiN」が設定されます。 ASCII コードでエンコードされているバイナリデータが設定されます。
15	長さ	4	int	次のエリア (項番 16) の長さです。 バイト単位で設定されます。
16	サービスオペレーション名	任意	String	要求先のサービス名に対するオペレーション名です。 255 バイトの XMLschema の NCName 定義文字で設定されます。 UTF-8 コードでエンコードされているバイナリデータが設定されます。
17	タグ	8	byte	パラメタ識別用のタグです (アイキャッチャ)。 「MessageT」が設定されます。 ASCII コードでエンコードされているバイナリデータが設定されます。
18	長さ	4	int	次のエリア (項番 19) の長さです。 バイト単位で設定されます。
19	応答電文種別	任意	String	メッセージ (応答電文) の種別です。 電文種別が XML 電文の場合、「XML」が設定されます。 メッセージ (応答電文) がいない場合、このパラメタは省略されます。 UTF-8 コードでエンコードされているバイナリデータが設定されます。
20	タグ	8	byte	パラメタ識別用のタグです (アイキャッチャ)。 「MgLength」が設定されます。 ASCII コードでエンコードされているバイナリデータが設定されます。
21	長さ	4	int	次のエリア (項番 22) の長さです。 「4」が設定されます。
22	メッセージ長	4	int	メッセージ (応答電文) の長さが設定されます。 メッセージ (応答電文) がいない場合、0 バイトが設定されます。
23	メッセージ	任意	byte []	次のどちらかの情報が設定されます。 <ul style="list-style-type: none"> <li>サービス部品からの応答電文</li> <li>サービス部品、ビジネスプロセス、およびカスタムアダプタからのエラー情報 (フォルト情報)</li> </ul> UTF-8 コードでエンコードされているバイナリデータが設定されます。

注 Java のデータ型です。

## (2) JMS の場合

受信用共有キューから応答 (JMS メッセージ) を取り出すには、MDB で実装する方法と SessionBean で実装する方法があります。MDB および SessionBean の実装については、マニュアル「Cosminexus Reliable Messaging」を参照してください。

なお、取り出した応答 (JMS メッセージ) のペイロードに設定されたバイナリデータの形式については、表 10-12 を参照してください。

### 10.5.11 応答電文を取得する (TP1/EE, JMS)

取り出した応答から応答電文を取得します。取得する応答電文は、次のどちらかになります。

- サービス部品からの応答電文
- サービス部品、ビジネスプロセス、およびカスタムアダプタからのエラー情報 (フォルト情報)

#### (1) サービス部品からの応答電文

取得する応答電文は、サービス部品側と同じ電文フォーマットの XML 電文になります。XML 電文の応答電文については、「10.2.7 応答電文を取得する」を参照してください。

#### (2) サービス部品、ビジネスプロセス、およびカスタムアダプタからのエラー情報 (フォルト情報)

取得する電文 (エラー情報) は、非同期受付 (DB キュー) が応答用のキューに送信した XML 電文になります。取得する XML 電文 (エラー情報) の形式については、「10.4.9 応答電文を取得する」を参照してください。

## 10.6 ユーザ定義受付（Web サービス）に要求を出すサービスリクエスト

ユーザ定義受付（Web サービス）へ要求電文を送信するサービスリクエストは、SOAP を利用してユーザ定義受付と通信します。サービスリクエストからユーザ定義受付へサービス部品実行の要求電文が送信され、HCSC サーバがサービス部品の実行を行います。

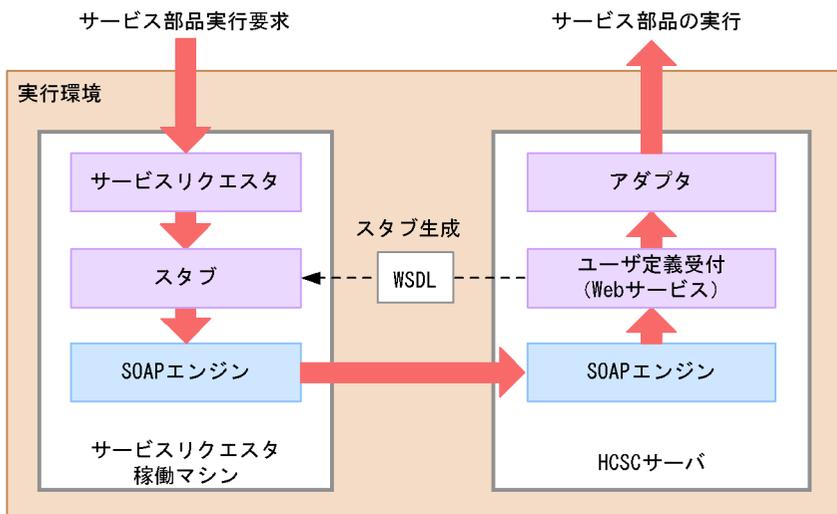
ユーザ定義受付（Web サービス）のインターフェース情報は、開発環境でユーザ定義受付を定義した際に指定した WSDL になります。WSDL からスタブを生成し、そのスタブを利用して、ユーザ定義受付（Web サービス）に要求を出します。

### ポイント

ユーザ定義受付（Web サービス）に要求を出すサービスリクエストを作成する場合、標準受付とは異なり、要求電文を作成する必要はありません。サービスリクエストのスタブの中で WSDL に合わせた形式の要求電文（SOAP メッセージ）が自動的に生成され、サービス部品の実行要求が出されます。

ユーザ定義受付（Web サービス）に要求を出すサービスリクエストと HCSC サーバの関係を次の図に示します。

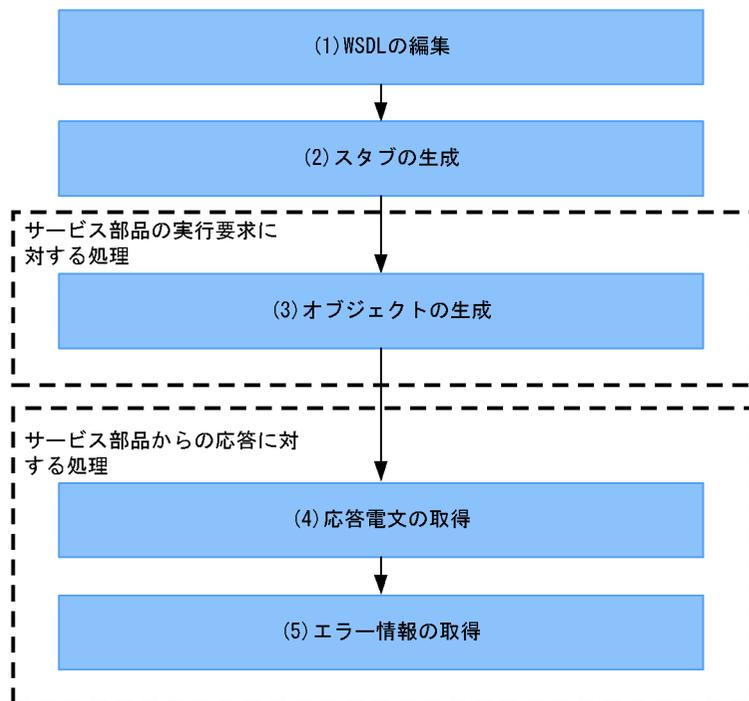
図 10-27 ユーザ定義受付（Web サービス）に要求を出すサービスリクエストと HCSC サーバの関係



## 10.6.1 サービスリクエスト（ユーザ定義受付（Web サービス））の作成手順

ユーザ定義受付（Web サービス）にサービス部品の実行を要求し、サービス部品を呼び出すためのサービスリクエストの作成手順を次に示します。

図 10-28 サービスリクエストの作成手順（ユーザ定義受付（Web サービス））



各工程の作業を次に示します。

### (1) WSDL の編集

ユーザ定義受付の定義内容を確認して、スタブの生成に必要な WSDL の内容を編集します。WSDL の編集については、「10.6.2 WSDL を編集する」を参照してください。

### (2) スタブの生成

(1) で編集した WSDL からスタブを生成します。スタブの生成については、「10.6.3 スタブを生成する」を参照してください。

### (3) オブジェクトの生成

ユーザ定義受付（Web サービス）のメソッドを呼び出すために、(2) で生成したスタブからオブジェクトを生成します。オブジェクトの生成については、「10.6.4 オブジェクトを生成する」を参照してください。

#### (4) 応答電文の取得

ユーザ定義受付 (Web サービス) から、サービス部品の実行要求に対する応答電文を取得します。応答電文の取得については、「10.6.5 応答電文を取得する」を参照してください。

#### (5) エラー情報の取得

要求先のサービス部品、HCSC サーバ、および SOAP エンジンでエラーが発生した場合、エラー情報を取得し、その情報に従って対処します。エラー情報の取得については、「10.6.6 エラー情報を取得する」を参照してください。

### 10.6.2 WSDL を編集する

ユーザ定義受付 (Web サービス) に実行要求を出すサービスリクエストを作成するには、開発環境でユーザ定義受付を定義した際に指定した WSDL を利用します。WSDL の作成については「8.3 WSDL の作成」を参照してください。

開発環境でユーザ定義受付を定義した際に指定した WSDL は、サービスロケーションの値 (wsdl:port 要素内の soap-address 要素の location 属性に設定している値) に、仮の値を設定しています。そのため、WSDL のサービスロケーションの値を、使用するユーザ定義受付の URL 情報に編集する必要があります。

HCSC サーバをセットアップする際に、HCSC サーバセットアップ定義ファイルにユーザ定義受付 (Web サービス) を使用するように指定 (request-userdef-soap=ON を指定) している場合は、開発環境のユーザ定義受付情報表示画面の [Web サービス] に URL が表示されます。表示される URL がユーザ定義受付の URL 情報です。この URL を WSDL のサービスロケーションの値に設定します。

HCSC サーバをセットアップしていない場合、または HCSC サーバをセットアップしたが HCSC サーバセットアップ定義ファイルでユーザ定義受付 (Web サービス) を使用しないように指定している場合は、次の規則に従ってサービスロケーションの値を設定してください。

```
http://<ホスト名>:<ポート番号>1/2<コンテキストルート>  
/services/CSCMsgUserDefinedReception
```

注 1

HCSC サーバの URL (ホスト名およびポート番号) です。

注 2

開発環境でユーザ定義受付を定義したときに割り当てられた受付 ID です。ユーザ定義受付情報表示画面の [受付 ID] に表示されます。

ユーザ定義受付情報表示画面の表示方法は、「9.4.2 HCSC コンポーネントの情報を表示する」を参照してください。

### 10.6.3 スタブを生成する

編集した WSDL からスタブを生成します。スタブは、Cosminexus が開発支援コマンドとして提供する WSDL2Java コマンドで生成できます。

コマンドの入力例を次に示します。

```
WSDL2Java xxxxx.wsdl
```

このコマンドを実行すると、指定した WSDL に記述されている内容に基づいてディレクトリおよびファイルが作成されます。

WSDL2Java コマンドのオプションなどの詳細、および生成されるスタブの内容については、マニュアル「Cosminexus SOAP アプリケーション開発ガイド」を参照してください。

### 10.6.4 オブジェクトを生成する

ユーザ定義受付 (Web サービス) のメソッドを呼び出すために、生成したスタブを使用して、オブジェクトを生成します。

ユーザ定義受付 (Web サービス) のメソッドを呼び出すためのオブジェクトを生成する手順を次に示します。

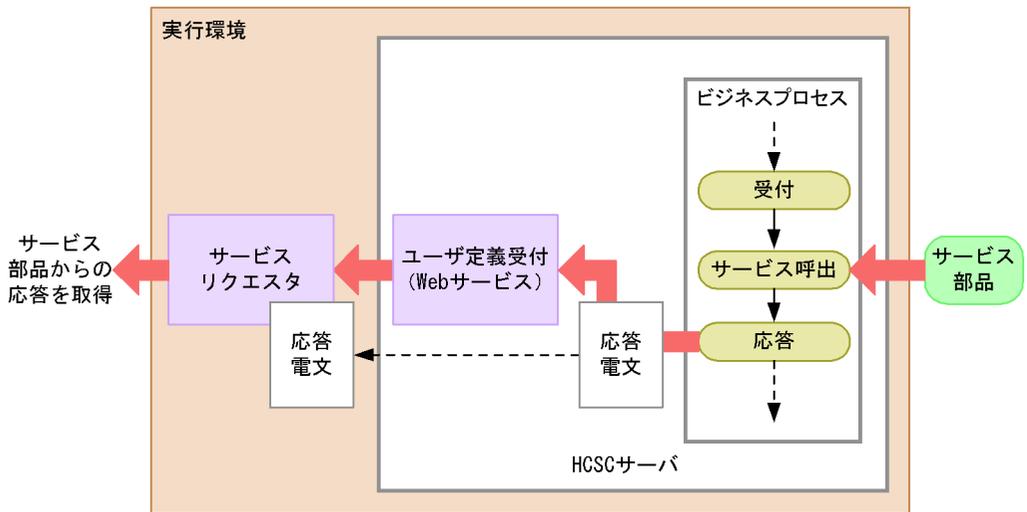
1. インターフェースクラスである Locator クラスのオブジェクトを生成します。
2. Locator クラスを使って、ユーザ定義受付のインターフェースクラスのオブジェクトを生成します。  
なお、生成・取得したサービスリクエストのインターフェースクラスのインスタンスは、複数のスレッドで共有できません。

生成したオブジェクトのメソッドを呼び出すことで、ユーザ定義受付にサービス部品の実行が要求されます。スタブのオブジェクト内で、WSDL で定義された形式の要求電文 (SOAP メッセージ) を自動生成します。

### 10.6.5 応答電文を取得する

ユーザ定義受付 (Web サービス) から、サービス部品の実行要求に対する応答電文を取得します。スタブのオブジェクトは、ユーザ定義受付からの応答電文として WSDL で定義された形式の応答電文 (SOAP メッセージ) を受け取り、サービスリクエストに応答を返します。応答電文の取得を次の図に示します。

図 10-29 応答電文の取得（ユーザ定義受付（Web サービス））



サービス部品からの応答電文がない場合は、NULL を受信します。応答電文がある場合とない場合に分けて次に説明します。

#### (1) 応答電文がある場合

サービスリクエストは、生成したオブジェクトのメソッドを呼び出した戻り値としてビジネスプロセスから応答電文を取得します。

#### (2) 応答電文がない場合

サービスリクエストは、生成したオブジェクトのメソッドを呼び出した戻り値として NULL を取得します。

### 10.6.6 エラー情報を取得する

要求先のサービス部品、HCSC サーバ、および SOAP エンジンでエラーが発生した場合、エラー情報を取得し、その情報に従って対処します。

エラーの伝わり方については、マニュアル「Cosminexus サービスプラットフォームシステム構築・運用ガイド」の Web サービス（SOAP 通信）実行時の障害対策のユーザ定義受付の場合に関する内容を参照してください。

#### (1) サービスリクエスト側の実装例

エラー情報の取得方法は、SOAP 通信基盤の種類によって異なります。

## 10. サービスリクエストの作成

(a) Cosminexus が提供している SOAP 通信基盤を使用している場合

### サービス部品からの SOAPFault の取得方法

WSDL に定義されているエラー情報用の例外オブジェクトをキャッチして SOAPFault のエラー情報を取得します。SOAPFault の形式は、WSDL に定義されている Fault 形式（サービス部品側の Fault 形式）です。

サービス部品からの SOAPFault を取得するには、サービスリクエスト側で次のように実装する必要があります。

```
{
  try {
    // Webサービス呼び出し
  } catch (xxxxxxxxxxxxException e) {
  } catch (C4Fault e) {
  }
}
```

### HCSC サーバ内で発生した例外の取得方法

Cosminexus の SOAP 通信基盤で提供する C4Fault オブジェクトをキャッチしてエラー情報を取得します。

サービスリクエスト側の処理の流れを次に示します。

1. C4Fault または RuntimeException でキャッチします。
2. C4Fault から getFaultDetails() メソッドを使用して Element[] を取り出し、各配列に対して getChildNodes() メソッドで NodeList を取得します。
3. NodeList から Node を取り出し、個々の Node に対して getTextContent() メソッドでエラー情報を取り出します。

HCSC サーバ内で発生した例外を取得する場合の、サービスリクエスト側の実装の例を次に示します。

```

{
    try {
        :
    } catch (xxxxxxxException e) {
        :
    } catch (C4Fault e) {
        // C4FaultからFault情報を取得
        System.out.println("C4Fault Message           = " + e.getMessage());
        System.out.println("C4Fault FaultCode        = " + e.getFaultCode());
        System.out.println("C4Fault FaultActor      = " + e.getFaultActor());
        System.out.println("C4Fault FaultString    = " + e.getFaultString());
        // C4FaultからDetailを取得
        Element ele[] = e.getFaultDetails();
        printElement(ele);
    }
}

/**
 * Print Element
 */
private static void printElement(Element[] ele) {
    int eleNumber = ele.length;
    System.out.println("Element count = " + eleNumber);
    for (int i=0; i<eleNumber; i++) {
        // Element配列からNodeListを取得
        if (ele[i] != null) {
            printNodeList(ele[i].getChildNodes());
        } else {
            System.out.println("Element[" + i + "] = null");
        }
    }
}

/**
 * Print NodeList
 */
private static void printNodeList(NodeList nodelist) {
    int nodelistcount = nodelist.getLength();
    for (int j=0; j<nodelistcount; j++) {
        Node node1 = nodelist.item(j);
        NodeList nodelist1 = node1.getChildNodes();
        int nodelist1len = nodelist1.getLength();

        if (nodelist1len == 0) {
            System.out.println("empty ChildNode");
        } else if (nodelist1len == 1) {
            Node node2 = node1.getFirstChild();
            if (node2 != null) {
                System.out.println("ChildNode[" + j + "] getTextContent = "
                    + node2.getTextContent());
            }
        } else {
            System.out.println("ChildNode[" + j + "] = null");
        }
    }
}
}

```

getFaultDetails() メソッドを使用して取得した Element[] の構造を次に示します。

表 10-13 getFaultDetails() メソッドを使用して取得した Element[] の構造

名称	説明
errorMessage	HCSC サーバ内で発生した例外に設定されているエラーメッセージです。
errorCode	HCSC サーバ内で発生した例外に対応するエラーコードです。
processInstanceID	ビジネスプロセスのインスタンス ID の情報です。 HCSC サーバ内で発生した例外の場合は設定されていないため、null (nil 属性) になります。

(b) Cosminexus が提供している SOAP 通信基盤を使用していない場合

サービス部品からの SOAPFault の取得方法

WSDL に定義されているエラー情報用の例外オブジェクトをキャッチして SOAPFault のエラー情報を取得します。SOAPFault の形式は、WSDL に定義されている Fault 形式 (サービス部品側の Fault 形式) です。

HCSC サーバ内で発生した例外の取得方法

取得する方法は、サービスリクエスト側で実装している SOAP エンジンに依存します。

## 10.7 DB アダプタに要求を出すサービスリクエスト

DB アダプタに要求を出すサービスリクエストは、要求電文を受け付ける標準受付の種類によって作成方法が異なります。それぞれの標準受付の種類ごとの作成方法を参照して作成してください。参照先を次の表に示します。

表 10-14 DB アダプタに要求を出すサービスリクエストの作成方法

標準受付の種類		参照先
同期受付	Web サービス	10.2
	SessionBean	10.3
非同期受付	MDB (WS-R)	10.4
	DB キュー	10.5

以降の項では、DB アダプタ電文の形式について説明します。

DB アダプタ用のサービスリクエストの電文の形式が、DB アダプタ電文の形式と異なる場合、データ変換が必要です。データ変換の詳細については、「7. データ変換の定義」を参照してください。

### 10.7.1 要求電文の形式

要求電文は、XML 文書形式で作成します。

要素、属性およびテキストデータは、W3C の XML の仕様に従って指定します。XML 文書で使用できない「<」、「>」、「&」などの文字を使用する場合、エスケープするか、CDATA セクションを利用する必要があります。

#### (1) 要求電文の形式

要求電文の形式は、単一の SQL 文の実行を要求する場合と、複数の SQL 文の実行を要求する場合とで異なります。

要求電文の形式を次に示します。なお、斜体の文字は可変値を表します。

単一の SQL 文の実行を要求する場合

ルート要素 (DBadapter 要素) の下位要素として *SQL 識別子* 要素を記述します。記述できる *SQL 識別子* 要素の数は、一つだけです。

## 10. サービスリクエストの作成

```
<DBadapter>
  <SQL識別子 out_maxOccurs="検索結果出力最大件数">
    <DBA_IN_DATA>
      <引数要素 nulldata="空要素の扱い">引数要素のデータ</引数要素>
      :
    </DBA_IN_DATA>
  </SQL識別子>
</DBadapter>
```

### 複数の SQL 文の実行を要求する場合

ルート要素 (DBadapter 要素) の下位要素として DBA\_MULTI\_SQL 要素を記述します。複数の DBA\_MULTI\_SQL 要素を記述できます。記述できる数は、SQL オペレーション定義ファイルの MAX\_SQL\_NO 要素に設定した値以下です。

```
<DBadapter>
  <DBA_MULTI_SQL>
    <SQL識別子 out_maxOccurs="検索結果出力最大件数">
      <DBA_IN_DATA>
        <引数要素 nulldata="空要素の扱い">引数要素のデータ</引数要素>
        :
      </DBA_IN_DATA>
    </SQL識別子>
  </DBA_MULTI_SQL>
  :
</DBadapter>
```

## (2) 要求電文の要素と属性

要求電文の要素と属性について次の表に示します。なお、要素および属性の指定は、SQL オペレーション定義ファイルの設定値に依存します。SQL オペレーション定義ファイルについては「4.3.5(1) SQL オペレーション定義ファイルを作成する」を参照してください。

表 10-15 要求電文の要素と属性 (DB アダプタ用のサービスリクエスト)

要素名・属性名	種別	内容と設定値	設定値の型	省略可否
DBadapter	要素	ルート要素です。次の下位要素があります。 単一 SQL 文の実行要求の場合 下位要素は <i>SQL 識別子</i> 要素です。 複数 SQL 文の実行要求の場合 下位要素は DBA_MULTI_SQL 要素です。	-	×
DBA_MULTI_SQL	要素	サービスリクエストからの要求が、複数の SQL 文を実行する要求であることを示す要素です。 <i>SQL 識別子</i> 以降の下位要素で入力情報を設定します。SQL オペレーション定義ファイルの MAX_SQL_NO 要素に設定した値以内で複数記述できます。	-	×

要素名・属性名	種別	内容と設定値	設定値の型	省略可否
<i>SQL 識別子</i>	要素	<i>SQL 識別子</i> には SQL オペレーション定義ファイルで指定した SQL 識別子の名称を指定します。 DBA_IN_DATA 以降の下位要素で、入力情報を設定します。 属性に out_maxOccurs があります。	-	×
out_maxOccurs	属性	検索結果出力最大件数を指定します。0 以上の値を指定します。 省略した場合 SQL オペレーション定義ファイルで指定した out_maxOccurs の値が設定されます。 「0」を設定した場合 最大件数が 2,147,483,647 件に設定されます。	xsd:int	
DBA_IN_DATA	要素	<i>引数要素</i> 以降の下位要素で、SQL オペレーション定義ファイルで設定した引数要素のデータを設定します。	-	×
<i>引数要素</i>	要素	<i>引数要素</i> には SQL オペレーション定義ファイルで指定した引数要素の名称を指定します。 この要素には、SQL オペレーション定義ファイルで指定した引数要素のデータが格納されます。 属性に nulldata があります。 注 SQL オペレーション定義ファイルの設定で、該当する SQL 識別子に引数要素がない場合、この要素は設定できません。	xsd:string または data_type の設定値 <sup>2</sup>	×
nulldata	属性	SQL オペレーション定義ファイルで指定した dba_inf (引数の種別) が「data」の場合に、空要素のデータを「null」として扱うかどうかを指定します (dba_inf が「data」以外の場合は、指定できません)。 「null」として扱う場合 「"Y"」を指定します。 「null」として扱わない場合 「"N"」を指定します。 なお、省略した場合、この属性には「"N"」が設定されます。	xsd:string {Y   N}	

## ( 凡例 )

- : 設定値はありません。
- : 省略できます。
- × : 省略できません。

## 注 1

複数の SQL 文の実行を要求する場合には必ず記述します。単一の SQL 文の実行を要求する場合には記述できません。

## 注 2

格納する引数要素の型は、SQL オペレーション定義ファイルで、該当する引数要素に指定した dba\_inf (引数の種別) によって異なります。dba\_inf が table, column または preset の場合、

## 10. サービスリクエストの作成

xsd:string 型になります。dba\_inf が data の場合、data\_type (データ型) の設定値に従ってください。data\_type の設定値と引数要素の型の対応については、表 10-16 および表 10-17 を参照してください。

また、dba\_inf が data の場合、引数要素に格納するデータをシングルクォーテーション (') で囲む必要はありません。

表 10-16 data\_type の設定値と引数要素の型の対応 (HiRDB の場合)

data_type の設定値	引数要素の型
INTEGER	xsd:int
SMALLINT	xsd:short
DECIMAL	xsd:string
FLOAT	xsd:double
REAL	xsd:float
CHAR	xsd:string
VARCHAR	xsd:string
DATE	xsd:date
TIME	xsd:string
TIMESTAMP	xsd:string

### 注

記述した型に加え、空要素の設定も許可されます。

表 10-17 data\_type の設定値と引数要素の型の対応 (Oracle の場合)

data_type の設定値	引数要素の型
NUMERIC	xsd:string
CHAR	xsd:string
VARCHAR	xsd:string
DATE	xsd:date
TIME	xsd:string
TIMESTAMP	xsd:string
LONGVARCHAR	xsd:string
CLOB	xsd:string

### 注

記述した型に加え、空要素の設定も許可されます。

なお、DB アダプタに送信する要求電文で設定する引数要素の値は、次の表のとおり

います。

表 10-18 DB アダプタに送信する要求電文での引数要素の扱い

要求電文の引数要素の値	SQL オペレーション定義ファイルの設定値		要求電文の nulldata の属性値	引数要素の扱い
	dba_inf の属性値	data_type の属性値		
空要素	data	-	"Y" または "Y "	null
			"Y" または "Y " 以外 (省略時も含む)	空文字列 (" ")
	data 以外	-	-	空文字列 (" ")
空要素以外	data	CHAR, VARCHAR, LONGVARCHAR, CLOB	-	値の前後の空白文字は削除しません。<引数要素> ABC </引数要素> は、空白文字を削除しないで " ABC " として扱います。
		CHAR, VARCHAR, LONGVARCHAR, CLOB 以外	-	値の前後の空白文字を削除します。<引数要素> 123 </引数要素> は、空白文字を削除して "123" として扱います。
	data 以外	-	-	値の前後の空白文字は削除しません。<引数要素> ABC </引数要素> は、空白文字を削除しないで " ABC " として扱います。

( 凡例 )

- : 判定の材料ではありません。
- : 一つ以上の空白文字 (半角空白文字, 改行文字, タブ文字) です。

注

nulldata の属性値の前後の空白文字は無視します。

## 10.7.2 応答電文の形式

応答電文は、XML 文書形式で出力されます。

なお、検索結果 (列名など) に XML 文書で使用できない「<」や「&」などの文字が含まれていた場合、「&lt;」や「&amp;」などにエスケープして応答電文に設定されます。そのため、データベースに格納されているデータをエスケープする必要はありません。

### (1) 応答電文の形式

応答電文の形式は、単一の SQL 文の実行要求に対する応答の場合と、複数の SQL 文の

## 10. サービスリクエストの作成

実行要求に対する応答の場合とで異なります。

応答電文の形式を次に示します。なお、斜体の文字は可変値を表します。

単一の SQL 文の実行要求に対する応答の場合

ルート要素 (DBadapter 要素) の下位要素として *SQL 識別子* 要素が出力されます。

```
<DBadapter>
  <SQL識別子 out_maxOccurs="検索結果出力最大件数">
    <DBA_IN_DATA>
      <引数要素 nulldata="空要素の扱い">引数要素のデータ</引数要素>
      :
    </DBA_IN_DATA>
    <DBA_OUT_DATA>
      <DBA_ResultSetNo>SQLリターン値</DBA_ResultSetNo>
      <DBA_ResultSetXmlNo>検索結果の件数</DBA_ResultSetXmlNo>
      <DBA_ResultSetName>
        <DBA_ResultSetColumnName cid="列番号">列の名称</DBA_ResultSetColumnName>
        :
      </DBA_ResultSetName>
      <DBA_ResultSet lid="行番号">
        <DBA_ResultSetColumn cid="列番号">該当行n列目の検索結果</DBA_ResultSetColumn>
        :
      </DBA_ResultSet>
      :
    </DBA_OUT_DATA>
  </SQL識別子>
</DBadapter>
```

複数の SQL 文の実行要求に対する応答の場合

ルート要素 (DBadapter 要素) の下位要素として DBA\_MULTI\_SQL 要素が出力されます。要求電文で記述した DBA\_MULTI\_SQL 要素と同数の DBA\_MULTI\_SQL 要素が出力されます。

```
<DBadapter>
  <DBA_MULTI_SQL>
    <SQL識別子 out_maxOccurs="検索結果出力最大件数">
      <DBA_IN_DATA>
        <引数要素 nulldata="空要素の扱い">引数要素のデータ</引数要素>
        :
      </DBA_IN_DATA>
      <DBA_OUT_DATA>
        <DBA_ResultSetNo>SQLリターン値</DBA_ResultSetNo>
        <DBA_ResultSetXmlNo>検索結果の件数</DBA_ResultSetXmlNo>
        <DBA_ResultSetName>
          <DBA_ResultSetColumnName cid="列番号">列の名称</DBA_ResultSetColumnName>
          :
        </DBA_ResultSetName>
        <DBA_ResultSet lid="行番号">
          <DBA_ResultSetColumn cid="列番号">該当行n列目の検索結果</DBA_ResultSetColumn>
          :
        </DBA_ResultSet>
        :
      </DBA_OUT_DATA>
    </SQL識別子>
  </DBA_MULTI_SQL>
  :
</DBadapter>
```

## (2) 応答電文の要素と属性

応答電文の要素と属性について次の表に示します。

表 10-19 応答電文の要素と属性 (DB アダプタ用のサービスリクエスト)

要素名・属性名	種別	内容と設定値	格納される値の型	省略の有無
DBadapter	要素	ルート要素です。次の下位要素があります。 単一 SQL 文の実行要求に対する応答の場合 下位要素として <i>SQL 識別子</i> 要素が出力されます。 複数 SQL 文の実行要求に対する応答の場合 下位要素として DBA_MULTI_SQL 要素が出力されます。	-	×
DBA_MULTI_SQL	要素	サービスリクエストからの複数の SQL 文の実行要求に対する応答であることを示す要素です。 <i>SQL 識別子</i> 以降の下位要素に入力情報 (要求電文での設定情報) および出力情報 (SQL の実行結果情報) が出力されます。 要求電文で記述した DBA_MULTI_SQL 要素と同数の DBA_MULTI_SQL 要素が出力されます。	-	×
<i>SQL 識別子</i>	要素	下位の要素「DBA_IN_DATA」および「DBA_OUT_DATA」に、それぞれ入力情報 (要求電文での設定情報) および出力情報 (SQL の実行結果情報) が格納されます。 <i>SQL 識別子</i> は、要求電文に設定した SQL 識別子です。 なお、属性 out_maxOccurs (検索結果出力最大件数) は、要求電文で設定した場合にだけ格納されます。	-	×
DBA_IN_DATA	要素	要求電文で設定した入力情報がこの要素の下位要素に格納されます。	-	×
DBA_OUT_DATA	要素	下位の要素に SQL 実行時の出力情報が格納されます。実行した SQL によって、格納される出力情報が異なります。 INSERT の場合 DBA_ResultSetNo が下位要素として格納されます。 SELECT の場合 次の要素が下位要素として格納されます。 • DBA_ResultSetNo • DBA_ResultSetXmlNo • DBA_ResultSetName • DBA_ResultSet	-	×

10. サービスリクエストの作成

要素名・属性名	種別	内容と設定値	格納される値の型	省略の有無
DBA_ResultSetNo	要素	<p>実行した SQL によって、格納される出力情報が異なります。</p> <p>INSERT の場合 更新した行数 ( PreparedStatement クラスの executeUpdate で取得した値 ) が格納されます。格納される要素の値が 2,147,483,647 を超えた場合に格納される値は、DB Connector の使用する JDBC ドライバの PreparedStatement クラスの executeUpdate の仕様に従います。</p> <p>SELECT の場合 検索結果の件数が格納されます。格納される要素の値が 2,147,483,647 を超えた場合、2,147,483,647 が格納されます。</p>	xsd:int	x
DBA_ResultSetXmlNo	要素	<p>実行した SQL が SELECT の場合、実際の検索結果の件数 ( DBA_ResultSetNo の値 ) のうち、この応答電文に格納されている検索結果の件数が格納されます。</p> <p>DBA_ResultSetNo の値が、要求電文または SQL オペレーション定義ファイルの out_maxOccurs に指定した値よりも大きい場合、out_maxOccurs に指定した値が格納されます。</p> <p>また、格納される要素の値が 2,147,483,647 を超えた場合、2,147,483,647 が格納されます。</p>	xsd:int	
DBA_ResultSetName	要素	<p>実行した SQL が SELECT の場合、検索結果の列の名称が、下位要素 DBA_ResultColumnName に格納されます。</p> <p>なお、検索結果が 0 件の場合、この要素は格納されません。</p>	-	
DBA_ResultColumnName	要素	<p>実行した SQL が SELECT の場合、該当する列の名称が格納されます。</p> <p>なお、結果の列がデータベースの列の場合、「列名」の形式で設定されます。また、結果の列名が取得されない場合、SQL を飛行した先のデータベースの仕様に従った名称が返されません。</p> <p>属性に cid があります。</p>	xsd:string	
cid	属性	DBA_ResultColumnName に格納される値の列番号を示します。	xsd:int	
DBA_ResultSet	要素	<p>実行した SQL が SELECT の場合、一行ごとの結果情報が、下位要素の DBA_ResultColumn に格納されます。なお、検索結果が 0 件の場合、この要素は格納されません。</p> <p>属性に lid があります。</p>	-	

要素名・属性名	種別	内容と設定値	格納される値の型	省略の有無
lid	属性	DBA_ResultSet に格納される値の行番号を示します。 また、格納される要素の値が 2,147,483,647 を超えた場合、2,147,483,647 が格納されます。	xsd:int	
DBA_ResultColumn	要素	該当する行 (DBA_ResultSet で示される行) の $n$ 列目の検索結果が格納されます。 <sup>2</sup> 属性に cid および nulldata があります。	xsd:string	
cid	属性	DBA_ResultColumn に格納される値の列番号を示します。	xsd:int	
nulldata	属性	DBA_ResultColumn に格納されたデータが null であるかどうかを示します。 null の場合「Y」が設定されます。null でない場合、この属性は設定されません。	xsd:string	

## (凡例)

- : 値は格納されません。
- : SQL が SELECT の場合にだけ出力されます。
- : SQL が SELECT の場合で、実行結果が 1 件以上のときに出力されます。
- × : 必ず出力されます。省略されません。

## 注 1

複数の SQL 文の実行要求に対する応答の場合には必ず出力されます。単一の SQL 文の実行要求に対する応答の場合には出力されません。

## 注 2

ResultSet クラスの getString で取得した値が格納されます。そのため、検索結果の書式は DB Connector の使用する JDBC ドライバの ResultSet クラスの getString の仕様に従います。検索結果の書式を変更する場合、データベースが提供する関数 (ファンクション) を使用して検索してください。

HiRDB の TIMESTAMP 型 (列名: c\_ts) の書式を VARCHAR\_FORMAT 関数で変更する場合の例を次に示します。

```
SELECT VARCHAR_FORMAT(c_ts, 'YYYY-MONTH-DD HH:MI') FROM 表名
```



# 11 開発環境で使用する画面

この章では、Cosminexus サービスプラットフォームの開発環境で使用する画面について説明します。

---

11.1 画面構成

---

11.2 画面一覧

---

11.3 バイナリフォーマット定義に関連するダイアログ

---

11.4 ビジネスプロセスに関連するダイアログ

---

11.5 データ変換（マッピング）に関連するダイアログ

---

11.6 ユーザ定義受付に関連するダイアログ

---

11.7 バイナリフォーマット定義画面の操作

---

11.8 ビジネスプロセス定義画面の操作

---

11.9 データ変換（マッピング）定義画面の操作

---

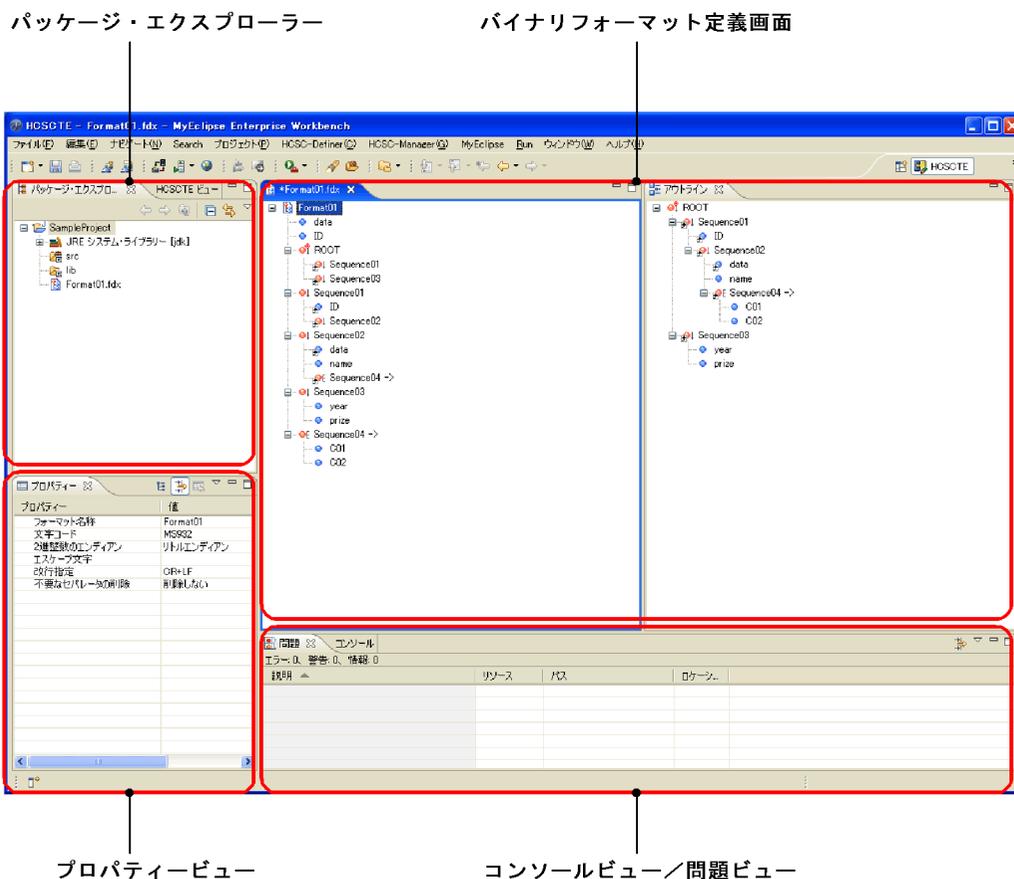
## 11.1 画面構成

開発環境で使用する画面の構成について説明します。

### 11.1.1 バイナリフォーマット定義ファイルを作成する画面の構成

バイナリフォーマット定義ファイルを作成する画面の構成を次の図に示します。

図 11-1 バイナリフォーマット定義ファイルを作成する画面の構成



パッケージ・エクスプローラー

バイナリフォーマット定義ファイルの一覧が表示されます。

バイナリフォーマット定義画面

バイナリフォーマット定義ファイルの内容を定義する画面です。バイナリフォーマット定義エディタおよびアウトラインビューから構成されます。

バイナリフォーマット定義画面の詳細については、「11.2.1 バイナリフォーマット

定義画面」を参照してください。

#### プロパティビュー

バイナリフォーマット定義画面で定義したバイナリフォーマット定義ファイルの情報や各要素の情報が表示されます。

#### コンソールビュー

バイナリフォーマット定義ファイルの検証結果が表示されます。

#### 問題ビュー

バイナリフォーマット定義ファイルを検証した結果、問題があった場合に、問題点の一覧が表示されます。

#### 参考

---

Eclipse のパースペクティブによっては、初期状態で表示されないビューがあります。この場合、Eclipse のメニューから [ ウィンドウ ] - [ ビューの表示 ] を選択して、必要なビューを開いてください。

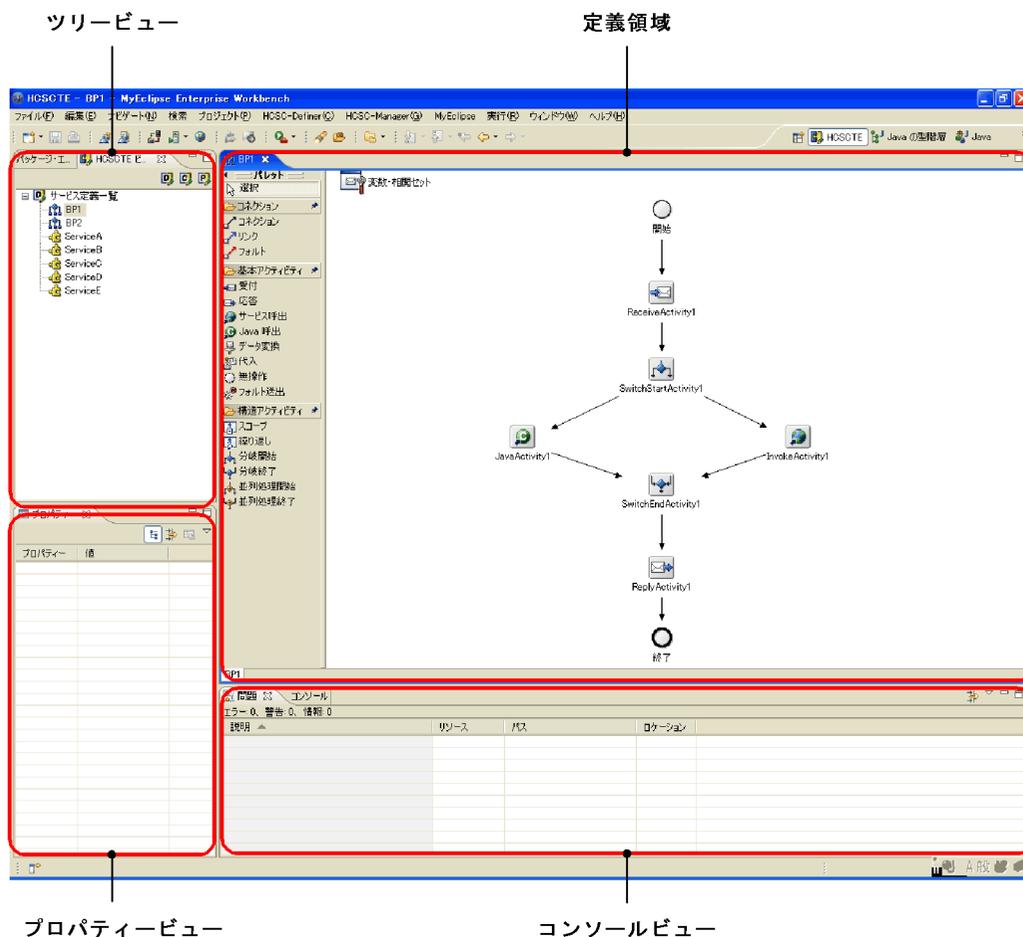
---

## 11. 開発環境で使用する画面

### 11.1.2 HCSC コンポーネントおよびデータ変換定義を作成する画面の構成

HCSC コンポーネントおよびデータ変換定義を作成する画面の構成を次の図に示します。

図 11-2 HCSC コンポーネントおよびデータ変換定義を作成する画面の構成



#### ツリービュー

サービス定義一覧、公開サービス一覧、およびシステム構成定義一覧が表示されます。

#### 定義領域

次のどれかの画面が表示されます。

- サービスアダプタ定義画面
- ビジネスプロセス定義画面
- データ変換（マッピング）定義画面

- ユーザ定義受付定義画面
- HCSC コンポーネント情報表示画面
- ユーザ定義受付情報表示画面

それぞれの画面の詳細については「11.2 画面一覧」を参照してください。

#### プロパティビュー

定義領域で定義した HCSC コンポーネントの情報などが表示されます。

#### コンソールビュー

検証結果やエラー情報が表示されます。

## 11.2 画面一覧

開発環境で使用する画面の一覧を次の表に示します。

表 11-1 開発環境で使用する画面の一覧

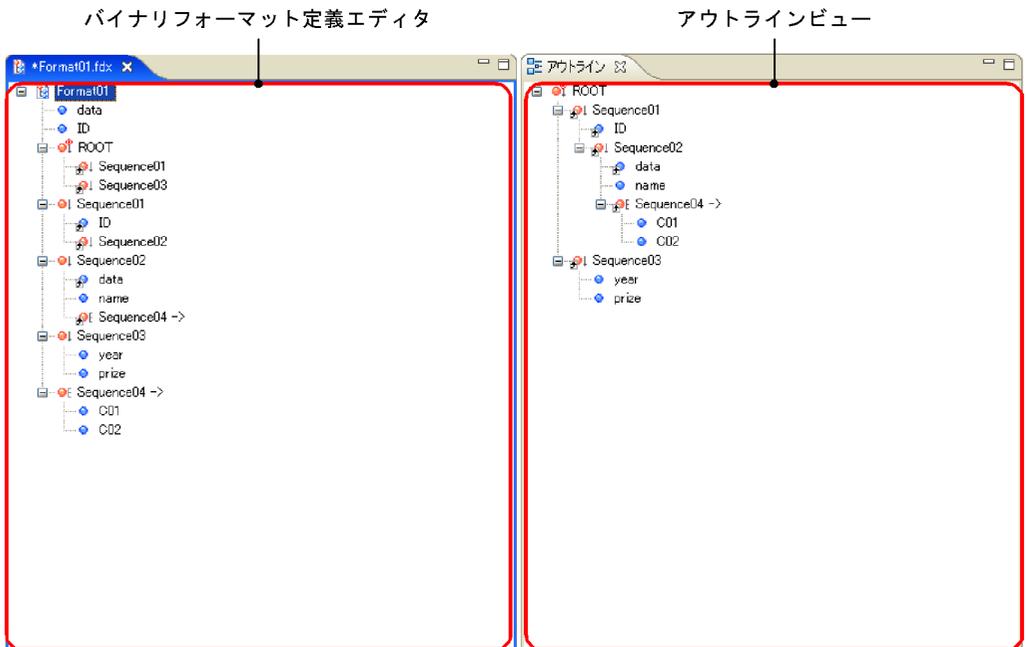
画面		説明
画面名	画面の構成要素（タブなど）	
バイナリフォーマット定義画面	バイナリフォーマット定義エディタ	バイナリフォーマット定義ファイルの内容を定義します。
	アウトラインビュー	バイナリフォーマット定義エディタで定義した各要素の関係が表示されます。
サービスアダプタ定義画面 (Web サービスの場合)	基本定義	Web サービスの基本項目を定義します。
	詳細定義	Web サービスの詳細項目を定義します。
サービスアダプタ定義画面 (SessionBean の場合)	基本定義	SessionBean の基本項目を定義します。
	詳細定義	SessionBean の詳細項目を定義します。
サービスアダプタ定義画面 (MDB (WS-R) の場合)	基本定義	MDB (WS-R) の基本項目を定義します。
	詳細定義	MDB (WS-R) の詳細項目を定義します。
サービスアダプタ定義画面 (MDB (DB キュー) の場合)	基本定義	MDB (DB キュー) の基本項目を定義します。
サービスアダプタ定義画面 (カスタムアダプタの場合)	基本定義	DB アダプタの基本項目を定義します。
	詳細定義	DB アダプタの詳細項目を定義します。
ビジネスプロセス定義画面		ビジネスプロセスを定義します。
データ変換 (マッピング) 定義画面		データ変換 (マッピング) を定義します。
ユーザ定義受付定義画面		ユーザ定義受付の定義内容が表示されます。
HCSC コンポーネント情報表示画面		HCSC サーバの構成および HCSC コンポーネントの情報が表示されます。
ユーザ定義受付情報表示画面		ユーザ定義受付の情報が表示されます。

## 11.2.1 バイナリフォーマット定義画面

バイナリフォーマット定義ファイルを作成するためのバイナリフォーマット定義画面について説明します。

バイナリフォーマット定義ファイルの作成方法については、「4.4 電文フォーマット (バイナリフォーマット定義ファイル) の作成方法」を参照してください。

図 11-3 バイナリフォーマット定義画面



バイナリフォーマット定義画面は、バイナリフォーマット定義エディタとアウトラインビューから構成されます。

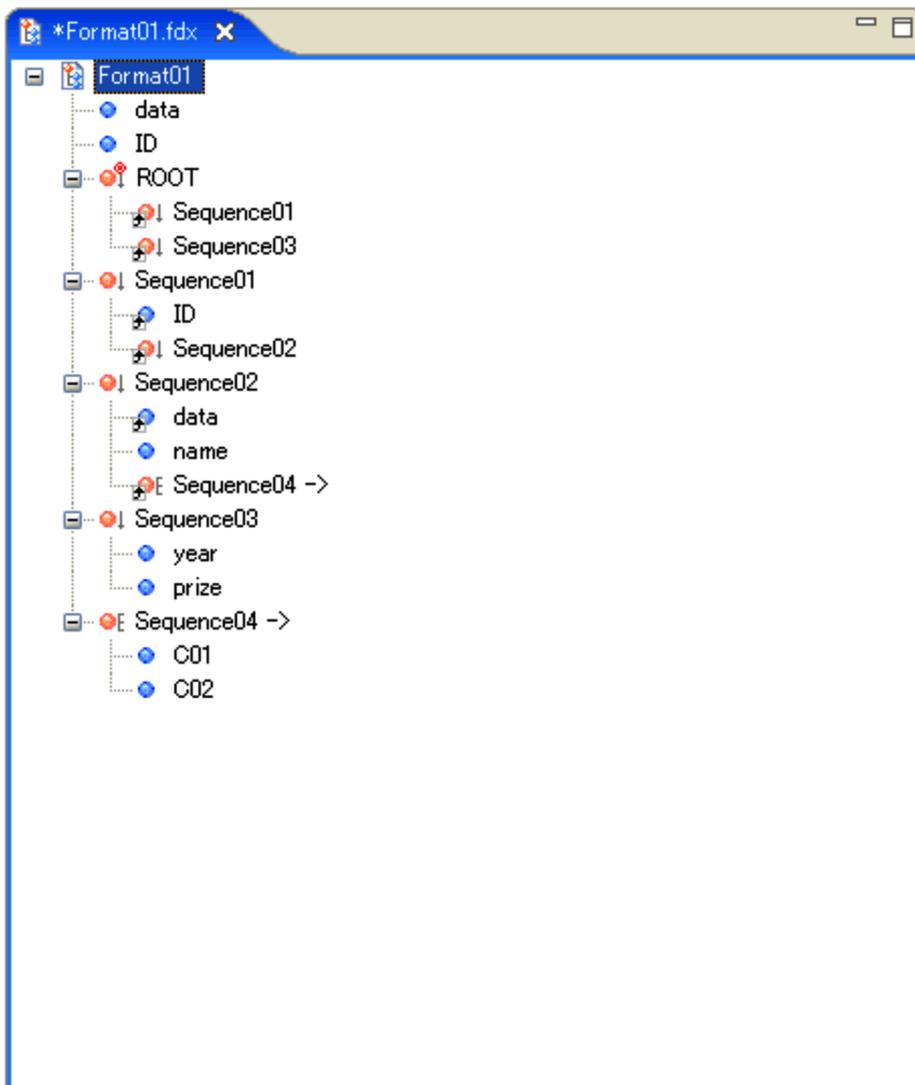
それぞれの表示内容について、次に説明します。

### (1) バイナリフォーマット定義エディタ

要素を設定してバイナリフォーマット定義ファイルの内容を定義します。定義した内容はツリー形式で表示されます。

バイナリフォーマット定義エディタの表示内容について説明します。

図 11-4 バイナリフォーマット定義エディタ



#### バイナリフォーマット定義ファイル名

バイナリフォーマット定義ファイルの名称です。

バイナリフォーマットエディタは、複数のバイナリフォーマット定義ファイルを開いて編集できます。編集するバイナリフォーマット定義ファイルの名称をクリックすると、編集の対象がアクティブになります。また、アウトラインビューの表示も切り替わります。

#### フォーマット ( )

バイナリフォーマット定義ファイルのフォーマット名です。

## ノード

バイナリフォーマット定義ファイル内に定義されている要素です。

フォーマットを基準に一つ目の階層はグローバル定義の要素です。二つ目の階層は複合内容要素の構成要素です。各ノードには、アイコンと要素名が表示されます。

なお、バイナリフォーマット定義エディタでノードを選択すると、アウトラインビューでも同じノードが選択状態になります。

各ノードの表示内容を次の表に示します。

表 11-2 バイナリフォーマット定義エディタに表示されるノード

区分	アイコン	説明
グローバル定義の要素		単純内容要素です。
		逐次構成の複合内容要素です。
		ルート要素に設定されている逐次構成の複合内容要素です。
		選択構成の複合内容要素です。
		ルート要素に設定されている選択構成の複合内容要素です。
複合内容要素の構成要素		構成要素として設定されている単純内容要素です。
		構成要素として設定されている逐次構成の複合内容要素です。
		構成要素として設定されている選択構成の複合内容要素です。
		ローカル定義の単純内容要素です。

検証の結果として、警告またはエラーが発生しているノードには、上で示すアイコンに警告またはエラーの発生を示すアイコン（警告：，エラー：）が付与されて表示されます。

## 出現回数

ノードに出現回数が設定されている場合に、次の形式で表示されます。

[最小出現回数:最大出現回数]

構成要素を設定したときに、出現回数をどのように設定したかによって、出現回数の表示は異なります。出現回数の表示と設定内容の関係を次の表に示します。

表 11-3 出現回数の表示と設定内容の関係

表示形式	出現回数の指定方法	説明
[n:n]	固定	n は固定の出現回数です。 最小出現回数と最大出現回数に同じ値が表示されます。例えば、出現回数が 3 回で固定の場合、[3:3] と表示されません。

## 11. 開発環境で使用する画面

表示形式	出現回数の指定方法	説明
[n:*]	データの終端まで	データを省略できる場合 最小出現回数 n が 0 になり、[0:*] と表示されます。 データを省略できない場合 最小出現回数 n が 1 になり、[1:*] と表示されます。
[0:*]->	ノード指定	最小出現回数は 0、最大出現回数は * で表示されます。また、ほかのノードを参照するノードであることを示す「->」が表示されます。
[n:m]	範囲指定	出現回数が n ~ m の範囲であることを示しています。

### 依存関係

ほかのノードを参照しているノード（依存ノード）の場合、「->」が表示されます。ただし、参照されているノード（ターゲットノード）には何も表示されません。なお、依存関係の詳細情報（パスなど）は、プロパティビューに表示されます。

### (2) アウトラインビュー

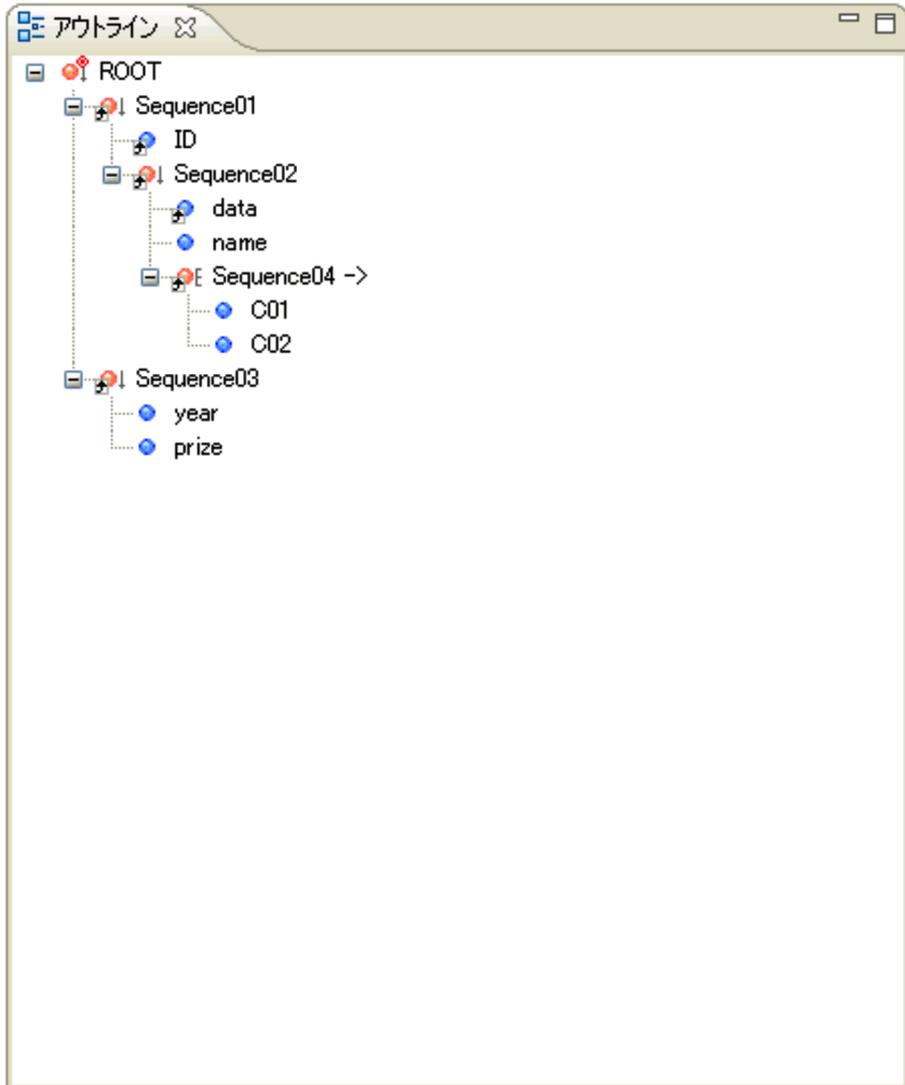
バイナリフォーマット定義エディタで定義した各要素の関係を確認できます。各要素の関係は、ルート要素を起点としたツリー形式で表現されます。

アウトラインビューの表示は、バイナリフォーマット定義エディタで要素を定義すると、自動的に更新されます。アウトラインビューでは要素を定義できません。

なお、アウトラインビューに表示されているノード（ルート要素および構成要素）を選択すると、バイナリフォーマット定義エディタでも同じノードが選択状態になります。

アウトラインビューの表示内容について説明します。

図 11-5 アウトラインビュー



## ノード

バイナリフォーマット定義ファイル内に定義されている要素です。  
各ノードの表示内容を次の表に示します。

表 11-4 アウトラインビューに表示されるノード

区分	アイコン	説明
ルート要素		ルート要素に設定されている逐次構成の複合内容要素です。
		ルート要素に設定されている選択構成の複合内容要素です。

## 11. 開発環境で使用する画面

区分	アイコン	説明
構成要素		構成要素として設定されている単純内容要素です。
		構成要素として設定されている逐次構成の複合内容要素です。
		構成要素として設定されている選択構成の複合内容要素です。
		ローカル定義の単純内容要素です。

検証の結果として、警告またはエラーが発生しているノードには、上で示すアイコンに警告またはエラーの発生を示すアイコン（警告：, エラー：）が付与されて表示されます。

### 11.2.2 サービスアダプタ定義画面

サービスアダプタまたは DB アダプタ（カスタムアダプタ）を定義するときのサービスアダプタ定義画面について説明します。

画面の表示方法、およびサービスアダプタの定義方法については、「5.3.1 サービスアダプタ（Web サービス）を定義する」、「5.3.2 サービスアダプタ（SessionBean）を定義する」、「5.3.3 サービスアダプタ（MDB（WS-R））を定義する」、または「5.3.4 サービスアダプタ（MDB（DB キュー））を定義する」を参照してください。

DB アダプタの定義方法については、「5.3.5 DB アダプタを定義する」を参照してください。

#### 注意事項

サービスアダプタ定義画面では、入力項目の前後に半角スペースや全角スペースを指定できません。ただし、MDB（WS-R）の詳細画面の[転送キュー作成先 RD エリア名]には、末尾に半角スペースを指定できます。

#### (1) Web サービスの場合

Web サービスに対応するサービスアダプタを定義する場合の画面について説明します。

##### (a) Web サービスの基本画面

Web サービスの基本画面の項目について説明します。

図 11-6 サービスアダプタ定義画面 (Web サービスの基本画面)

The screenshot shows a configuration window for 'ServiceA' with the following sections:

- サービス部品制御情報**
  - サービス名: ServiceA
  - サービス ID: 1
  - サービス種別: Web サービス
  - アドレス: http://localhost:80/2ee/C4WebSer
  - 最大インスタンス数: 0
  - サービスクラス名: UserInfo
  - オペレーション: eetUserData
- オペレーション情報**
  - オペレーション名: eetUserData
  - 通信モデル: 同期
- 要求電文**
  - 標準:  使う
  - フォーマット ID: format1\_web
  - 電文フォーマット: F1.xsd
  - サービス部品:
    - フォーマット ID: format2\_web
    - 電文フォーマット: cscformat2.xsd
  - データ変換定義: mapping1
- 応答電文**
  - 標準:  使う
  - フォーマット ID: format3\_web
  - 電文フォーマット: F2.xsd
  - サービス部品:
    - フォーマット ID: format4\_web
    - 電文フォーマット: cscformat3.xsd
  - データ変換定義: mapping2
- フォルト電文**
  - フォルト名: FAULT1
  - 電文フォーマット: cscformat1.xsd

## (i) [ サービス部品制御情報 ]

サービスアダプタ新規作成時に指定したサービス部品についての情報が表示されます。

## [ サービス名 ]

HCSC コンポーネントの名称が表示されます。

## [ サービス ID ]

HCSC コンポーネントの ID が表示されます。

## [ サービス種別 ]

サービス部品の種別として「Web サービス」が表示されます。

## [ アドレス ]

サービス部品のある URL が表示されます。

## [ 最大インスタンス数 ]

サービス部品の最大インスタンス数が表示されます。

## [ サービスクラス名 ]

サービス部品のクラス名が表示されます。

## [ オペレーション ]

オペレーション名が表示されます。サービス部品が持つオペレーション名をドロップダウンリストから選択できます。

(ii) [ オペレーション情報 ]

サービス部品のオペレーションについての情報が表示されます。

[ オペレーション名 ]

サービス部品制御情報のオペレーション欄で選択したオペレーションの名称が表示されます。

[ 通信モデル ]

サービス部品制御情報のオペレーション欄で選択したオペレーションの通信モデルとして「同期」が表示されます。

(iii) [ 要求電文 ]

サービスリクエストから送信される要求電文の情報が表示されます。

・[ 標準電文 ]

[ 使う ] チェックボックス

サービスリクエストから受け付けた標準電文の内容を確認したり、標準電文をサービス部品電文のフォーマットにデータ変換したりする場合にチェックします。

[ フォーマット ID ]

サービスリクエストから受け付けた標準電文のフォーマット ID が表示されます。

[ 電文フォーマット ]

サービスリクエストから受け付けた標準電文のフォーマット名が表示されます。

[ ... ] ボタン

標準電文フォーマットを設定するダイアログが表示されます。

[ 表示 ] ボタン

標準電文のフォーマットが表示されます。

・[ サービス部品 ]

[ フォーマット ID ]

サービス部品の電文フォーマットの ID が「formatN」( N は 1 以上の整数 ) の形式で自動的に割り当てられます。フォーマットの ID は変更できます。

[ 電文フォーマット ]

サービス部品の電文フォーマット名が表示されます。電文フォーマットは、サービス部品電文のオペレーションの要求電文に該当する WSDL から自動生成されます。変更はできません。

[ ... ] ボタン

このボタンは使用しません。

[ 表示 ] ボタン

サービス部品電文のフォーマットが表示されます。

- ・[ データ変換 ]

- [ データ変換定義 ]

- 標準電文からサービス部品電文へデータ変換するための定義ファイル名を入力します。

- [ 編集 ] ボタン

- データ変換定義画面が表示されます。データ変換定義欄で設定した定義ファイルの内容を編集できます。初めて定義する場合は、[ ルート要素選択 ] ダイアログが表示されます。

- [ 削除 ] ボタン

- [ 編集 ] ボタンで設定したデータ変換定義を削除します。

- (iv) [ 応答電文 ]

- サービス部品から送信される応答電文の情報が表示されます。

- ・[ 標準電文 ]

- [ 使う ] チェックボックス

- サービス部品から受け付けたサービス部品電文の内容を確認したり、サービス部品電文を標準電文のフォーマットにデータ変換したりする場合にチェックします。

- [ フォーマット ID ]

- サービス部品から受け付けた標準電文のフォーマット ID が表示されます。

- [ 電文フォーマット ]

- サービス部品から受け付けた標準電文のフォーマット名が表示されます。

- [ ... ] ボタン

- 標準電文フォーマットを設定するダイアログが表示されます。

- [ 表示 ] ボタン

- 標準電文のフォーマットが表示されます。

- ・[ サービス部品 ]

- [ フォーマット ID ]

- サービス部品の電文フォーマットの ID が「formatN」( N は 1 以上の整数 ) の形式で自動的に割り当てられます。フォーマットの ID は変更できます。

- [ 電文フォーマット ]

- サービス部品の電文フォーマット名が表示されます。電文フォーマットは、サービス部品電文のオペレーションの応答電文に該当する WSDL から自動生成されます。変更はできません。

- [ ... ] ボタン

- このボタンは使用しません。

- [ 表示 ] ボタン

- サービス部品電文のフォーマットが表示されます。

## 11. 開発環境で使用する画面

### ・[ データ変換 ]

#### [ データ変換定義 ]

サービス部品電文から標準電文へデータ変換するための定義ファイル名を入力します。

#### [ 編集 ] ボタン

データ変換定義画面が表示されます。データ変換定義欄で設定した定義ファイルの内容を編集できます。初めて定義する場合は、[ ルート要素選択 ] ダイアログが表示されます。

#### [ 削除 ] ボタン

[ 編集 ] ボタンで指定したデータ変換定義を削除します。

### (v) [ フォルト電文 ]

サービス部品のフォルト電文についての情報が表示されます。

#### [ フォルト名 ]

サービス部品のフォルトの名称が表示されます。

#### [ 電文フォーマット ]

サービス部品のフォルト電文のフォーマット名が表示されます。

#### [ ... ] ボタン

このボタンは使用しません。

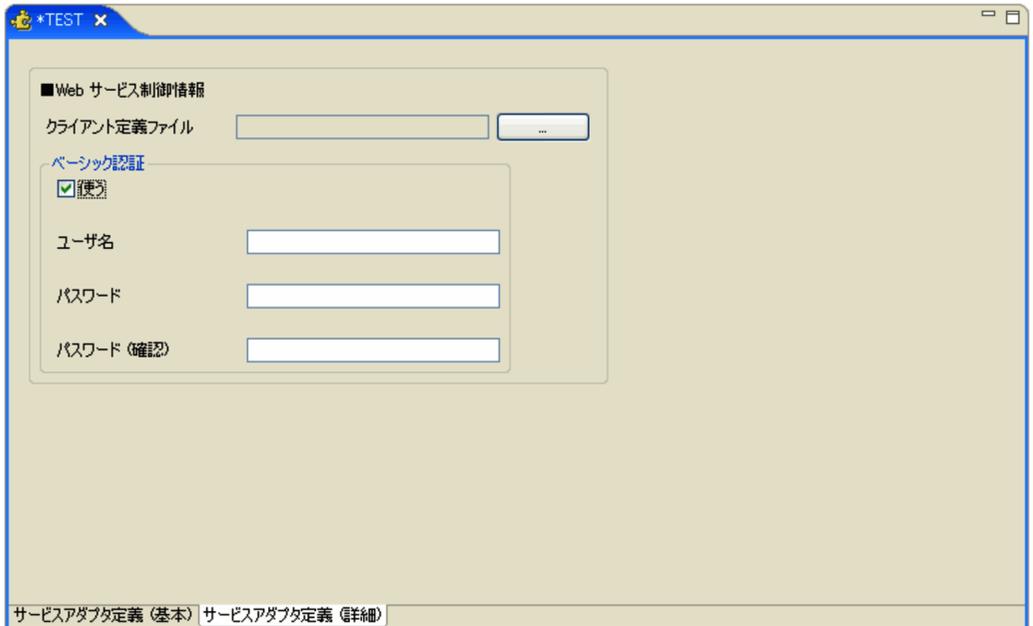
#### [ 表示 ] ボタン

フォルト電文のフォーマットが表示されます。

## (b) Web サービスの詳細画面

Web サービスの詳細画面の項目について説明します。

図 11-7 サービスアダプタ定義画面 (Web サービスの詳細画面)



## (i) [ Web サービス制御情報 ]

サービスアダプタ新規作成時に指定したサービス部品について、サービス種別ごとの情報が表示されます。

- ・クライアント定義ファイル

## [ クライアント定義ファイル ]

[ ... ] ボタンで設定したクライアント定義ファイル名が表示されます。

## [ ... ] ボタン

クライアント定義ファイルを設定するダイアログが表示されます。

- ・ベーシック認証

## [ ベーシック認証 ]

## [ 使う ] チェックボックス

サービス部品で、ベーシック認証を使用している場合にチェックします。

## [ ユーザ名 ]

16 バイト以内の英数字を半角で入力します。

## 11. 開発環境で使用する画面

### [ パスワード ]

16 バイト以内の英数字を半角で入力します。

### [ パスワード ( 確認 ) ]

パスワード欄で入力したパスワードを再度入力します。

## (2) SessionBean の場合

SessionBean に対応するサービスアダプタを定義する場合の画面について説明します。

### (a) SessionBean の基本画面

SessionBean の基本画面の項目について説明します。

図 11-8 サービスアダプタ定義画面 (SessionBean の基本画面)

The screenshot shows the 'ServiceB' configuration window. It contains the following sections:

- サービス部品制御情報**: Service Name (ServiceB), Service ID (2), Service Type (SessionBean), Address (Address\_sb), Max Instance Count (0), Service Class Name (myejb.adder.Adder), Operation (add).
- オペレーション情報**: Operation Name (add), Communication Model (同期).
- 要求電文**:
  - 標準:  使う
  - フォーマット ID: format1\_sb
  - 電文フォーマット: F1.xsd
  - サービス部品:
    - フォーマット ID: format2\_sb
    - 電文フォーマット: cscformat1.xsd
  - データ変換定義: mapping1 (編集, 削除)
- 応答電文**:
  - 標準:  使う
  - フォーマット ID: format3\_sb
  - 電文フォーマット: F2.xsd
  - サービス部品:
    - フォーマット ID: format4\_sb
    - 電文フォーマット: cscformat2.xsd
  - データ変換定義: mapping2 (編集, 削除)
- フォルト電文**:
  - フォルト名: (dropdown)
  - 電文フォーマット: (text field)

#### (i) [ サービス部品制御情報 ]

サービスアダプタ新規作成時に指定したサービス部品についての情報が表示されます。

#### [ サービス種別 ]

サービス部品の種別として「SessionBean」が表示されます。

#### [ アドレス ]

JNDI 名前空間に登録されている呼び出すサービス部品の名称を入力します。リモート呼び出しとなるようにアドレスを指定します (ローカル呼び出しはできません)。

SessionBean の場合は、フォルト電文の情報は表示されません。

上記以外の各項目は、Web サービスの基本画面と同様です。Web サービスの基本画面の項目については、「11.2.2(1)(a) Web サービスの基本画面」を参照してください。

## 11. 開発環境で使用する画面

### (b) SessionBean の詳細画面

SessionBean の詳細画面の項目について説明します。

図 11-9 サービスアダプタ定義画面 (SessionBean の詳細画面)



#### (i) [ SessionBean 制御情報 ]

サービスアダプタ新規作成時に指定したサービス部品について、サービス種別ごとの情報が表示されます。

##### ・クライアント定義ファイル

[ クライアント定義ファイル ]

クライアント定義ファイル名が表示されます。

[ ... ] ボタン

クライアント定義ファイルを設定するダイアログが表示されます。

##### ・ユーザ定義クラス

[ ユーザ定義クラス (JAR ファイル) ]

[ 追加 ] ボタンで設定したユーザ定義クラス (JAR ファイル) 名が表示されません。

[ 追加 ] ボタン

ユーザ定義クラス (JAR ファイル) を設定するダイアログが表示されます。選択する JAR ファイルの拡張子は、すべて小文字の「jar」としてください。

[ 削除 ] ボタン

ユーザ定義クラス欄で選択した JAR ファイルを削除します。

## 11. 開発環境で使用する画面

### (3) MDB (WS-R) の場合

MDB (WS-R) に対応するサービスアダプタを定義する場合の画面について説明します。

#### (a) MDB (WS-R) の基本画面

MDB (WS-R) の基本画面の項目について説明します。

図 11-10 サービスアダプタ定義画面 (MDB (WS-R) の基本画面)

The screenshot shows the 'ServiceC' configuration window with the following sections:

- サービス部品制御情報**
  - サービス名: ServiceC
  - サービス ID: 3
  - サービス種別: MDB/WS-R
  - アドレス: Address\_mdb
  - 最大インスタンス数: 0
  - サービスクラス名: (empty)
  - オペレーション: send
- オペレーション情報**
  - オペレーション名: send
  - 通信モデル: 非同期
- 要求電文**
  - 標準:  使う
  - フォーマット ID: format1\_wsr
  - 電文フォーマット: F1.xsd (表示...)
  - サービス部品:  使う
  - フォーマット ID: format2\_wsr
  - 電文フォーマット: F2.xsd (表示...)
  - データ変換定義: mappine (編集... 削除)
- 応答電文**
  - 標準:  使う
  - フォーマット ID: (empty)
  - 電文フォーマット: (empty) (表示...)
  - サービス部品:  使う
  - フォーマット ID: (empty)
  - 電文フォーマット: (empty) (表示...)
  - データ変換定義: (empty) (編集... 削除)
- フォルト電文**
  - フォルト名: (empty)
  - 電文フォーマット: (empty) (表示...)

At the bottom, there are tabs: 'サービスアダプタ定義 (基本)' and 'サービスアダプタ定義 (詳細)'.

#### (i) [ サービス部品制御情報 ]

サービスアダプタ新規作成時に指定したサービス部品についての情報が表示されます。

#### [ サービス種別 ]

サービス部品の種別として「MDB/WS-R」が表示されます。

#### [ アドレス ]

サービス部品に関連づけられているキュー名を入力します。

#### [ サービスクラス名 ]

この項目は使用しません。

#### (ii) [ オペレーション情報 ]

サービス部品のオペレーションについての情報が表示されます。

[ 通信モデル ]

サービス部品制御情報のオペレーション欄で選択したオペレーションの通信モデルとして「非同期」が表示されます。

(iii) [ 要求電文 ]

・[ サービス部品 ]

[ フォーマット ID ]

サービス部品の電文フォーマットの ID を入力します。

[ 電文フォーマット ]

サービス部品電文のフォーマット名を入力します。

[ ... ] ボタン

サービス部品電文フォーマットを設定するダイアログが表示されます。

MDB ( WS-R ) の場合は、応答電文、およびフォルト電文の情報は表示されません。

上記以外の項目は、Web サービスの基本画面と同様です。Web サービスの基本画面の項目については、「11.2.2(1)(a) Web サービスの基本画面」を参照してください。

## 11. 開発環境で使用する画面

### (b) MDB (WS-R) の詳細画面

MDB (WS-R) の詳細画面の項目について説明します。

図 11-11 サービスアダプタ定義画面 (MDB (WS-R) の詳細画面)

The screenshot shows a web-based configuration interface for ServiceC. The main panel is titled "MDB 制御情報". It includes a dropdown for "JMS メッセージ種別" (JMS Message Type) set to "テキスト" (Text). Under "呼出し設定" (Call Settings), "リモート呼出し" (Remote Call) is selected. The "リモート呼出し設定" (Remote Call Settings) section contains a "最大メッセージ数" (Max Message Count) field with the value "1024" and a "宛先 URL" (Destination URL) field. The "バージョン認証" (Version Authentication) section has a checked "バージョン認証" checkbox and three password fields: "ユーザ名" (Username), "パスワード" (Password), and "パスワード (確認)" (Confirm Password). At the bottom, there is a "転送キュー作成先 RD エリア" (Destination RD Area for Queue Creation) field and an unchecked "順序保証" (Order Guarantee) checkbox. The window title bar reads "ServiceC". The bottom status bar indicates the current view is "サービスアダプタ定義 (詳細)".

#### (i) [ MDB 制御情報 ]

サービスアダプタ新規作成時に指定したサービス部品について、サービス種別ごとの情報が表示されます。

#### [ JMS メッセージ種別 ]

サービス部品の JMS メッセージ種別が表示されます。次に示すメッセージ種別をドロップダウンリストから選択します。

- テキスト
- バイト
- オブジェクト

#### ・[ 呼出し設定 ]

##### [ リモート呼出し ]

サービス部品をリモート呼出しする場合に選択します。選択すると、リモート呼出し設定の各項目が活性化し、入力できる状態になります。

##### [ ローカル呼出し ]

サービス部品をローカル呼出しする場合に選択します。選択すると、リモート

呼出し設定の各項目が非活性化し、入力できない状態になります。

・[ リモート呼出し設定 ]

[ 最大メッセージ数 ]

サービス部品が送信できる電文の最大数を設定します。設定できる値は 1 ~ 65535 です。

[ 宛先 URL ]

非同期時の応答結果の受信先 URL を設定します。

・[ ベーシック認証 ]

[ 使う ] チェックボックス

サービス部品で、ベーシック認証を使用している場合にチェックします。

[ ユーザ名 ]

16 バイト以内の英数字を半角で入力します。

[ パスワード ]

16 バイト以内の英数字を半角で入力します。

[ パスワード ( 確認 ) ]

パスワード欄で入力したパスワードを再度入力します。

[ 転送キュー作成先 RD エリア名 ]

転送キューの作成先 RD エリア名を指定します。なお、この項目の末尾には半角スペースを指定できます。

[ 順序保証 ] チェックボックス

Cosminexus RM で転送の順序を保障する場合にチェックします。

## 11. 開発環境で使用する画面

### (4) MDB (DB キュー) の場合

MDB (DB キュー) に対応するサービスアダプタを定義する場合の画面について説明します。

#### (a) MDB (DB キュー) の基本画面

MDB (DB キュー) の基本画面の項目について説明します。

図 11-12 サービスアダプタ定義画面 (MDB (DB キュー) の基本画面)

The screenshot shows the 'ServiceD' configuration window with the following sections:

- サービス部品制御情報**
  - サービス名: ServiceD
  - サービス ID: 4
  - サービス種別: MDB/DB キュー
  - アドレス: Address\_dbq
  - 最大インスタンス数: 0
  - サービスクラス名:
  - オペレーション: send
- オペレーション情報**
  - オペレーション名: send
  - 通信モデル: 非同期
- 要求電文**
  - 標準:  (使う)
  - フォーマット ID: format1\_dbq
  - 電文フォーマット: F1.xsd
  - サービス部品:  (使う)
  - フォーマット ID: format2\_dbq
  - 電文フォーマット: F2.xsd
  - データ型定義: mapping
- 応答電文**
  - 標準:  (使う)
  - フォーマット ID:
  - 電文フォーマット:
  - サービス部品:  (使う)
  - フォーマット ID:
  - 電文フォーマット:
  - データ型定義:
- フォルト電文**
  - フォルト名:
  - 電文フォーマット:

#### (i) [ サービス部品制御情報 ]

サービスアダプタ新規作成時に指定したサービス部品についての情報が表示されます。

#### [ サービス種別 ]

サービス部品の種別として「MDB/DB キュー」が表示されます。

上記以外の項目は、MDB (WS-R) の基本画面と同様です。MDB (WS-R) の基本画面の項目については、「11.2.2(3)(a) MDB (WS-R) の基本画面」を参照してください。

なお、MDB (DB キュー) には、詳細画面はありません。

## (5) カスタムアダプタの場合

カスタムアダプタ (DB アダプタ) を定義する場合の画面について説明します。

### (a) カスタムアダプタの基本画面

カスタムアダプタの基本画面の項目について説明します。

図 11-13 サービスアダプタ定義画面 (カスタムアダプタの基本画面)

#### (i) [ サービス部品制御情報 ]

DB アダプタ新規作成時に指定したサービス部品についての情報が表示されます。

##### [ サービス名 ]

HCSC コンポーネントの名称が表示されます。

##### [ サービス ID ]

HCSC コンポーネントの ID が表示されます。

##### [ サービス種別 ]

サービス部品の種別として「汎用カスタムアダプタ」が表示されます。

##### [ アドレス ]

この項目は使用しません。

##### [ 最大インスタンス数 ]

サービス部品の最大インスタンス数が表示されます。

## 11. 開発環境で使用する画面

### [ サービスクラス名 ]

この項目は使用しません。

### [ オペレーション ]

オペレーション名が表示されます。サービス部品が持つオペレーション名をドロップダウンリストから選択できます。

### [ 追加 ] ボタン

オペレーションを追加するダイアログが表示されます。

### [ 削除 ] ボタン

オペレーション欄で選択したオペレーションを削除します。

## (ii) [ オペレーション情報 ]

サービス部品のオペレーションについての情報が表示されます。

### [ オペレーション名 ]

サービス部品制御情報のオペレーション欄で選択したオペレーションの名称が表示されます。

### [ 通信モデル ]

サービス部品制御情報のオペレーション欄で選択したオペレーションの通信モデルが表示されます。同期または非同期をドロップダウンリストから選択できます。選択内容によって、次のように表示が変わります。

- 同期を選択した場合：応答電文とフォルト電文の情報が表示されます。
- 非同期を選択した場合：応答電文とフォルト電文の情報は表示されません。非同期のまま、アダプタ定義を保存すると、応答電文とフォルト電文の全項目は消えます。

## (iii) [ 要求電文 ]

サービスリクエストから送信される要求電文の情報が表示されます。

### ・[ 標準電文 ]

#### [ 使う ] チェックボックス

サービスリクエストから受け付けた標準電文の内容を確認したり、標準電文をDBアダプタ電文のフォーマットにデータ変換したりする場合にチェックします。

#### [ フォーマット ID ]

サービスリクエストから受け付けた標準電文のフォーマット ID が表示されます。

#### [ 電文フォーマット ]

サービスリクエストから受け付けた標準電文のフォーマット名が表示されます。

#### [ ... ] ボタン

標準電文フォーマットを設定するダイアログが表示されます。

#### [ 表示 ] ボタン

標準電文のフォーマットが表示されます。

## ・[ サービス部品 ]

## [ フォーマット ID ]

DB アダプタ電文のフォーマット (DB アダプタ用の XML フォーマット定義ファイル) ID を入力します。

## [ 電文フォーマット ]

DB アダプタ電文のフォーマット (DB アダプタ用の XML フォーマット定義ファイル) 名を入力します。

## [ ... ] ボタン

DB アダプタ電文のフォーマットファイル (DB アダプタ用の XML フォーマット定義ファイル) を設定するダイアログが表示されます。

## [ 表示 ] ボタン

DB アダプタ電文のフォーマット (DB アダプタ用の XML フォーマット定義ファイル) が表示されます。

## ・[ データ変換 ]

## [ データ変換定義 ]

標準電文から DB アダプタ電文へデータ変換するための定義ファイル名を入力します。

## [ 編集 ] ボタン

データ変換定義画面が表示されます。データ変換定義欄で設定した定義ファイルの内容を編集できます。初めて定義する場合は、[ ルート要素選択 ] ダイアログが表示されます。

## [ 削除 ] ボタン

[ 編集 ] ボタンで設定したデータ変換定義を削除します。

## (iv) [ 応答電文 ]

サービス部品から送信される応答電文の情報が表示されます。

オペレーション情報の通信モデル欄で、「同期」を選択した場合だけ、設定できます。

## ・[ 標準電文 ]

## [ 使う ] チェックボックス

サービス部品から受け付けた DB アダプタ電文の内容を確認したり、DB アダプタ電文を標準電文のフォーマットにデータ変換したりする場合にチェックします。

## [ フォーマット ID ]

サービス部品から受け付けた標準電文のフォーマット ID が表示されます。

## [ 電文フォーマット ]

サービス部品から受け付けた標準電文のフォーマット名が表示されます。

## 11. 開発環境で使用する画面

### [ ... ] ボタン

標準電文フォーマットを設定するダイアログが表示されます。

### [ 表示 ] ボタン

標準電文フォーマットが表示されます。

## ・[ サービス部品 ]

### [ フォーマット ID ]

サービス部品の電文フォーマット (DB アダプタ用の XML フォーマット定義ファイル) ID が表示されます。

### [ 電文フォーマット ]

サービス部品の電文フォーマット (DB アダプタ用の XML フォーマット定義ファイル) 名が表示されます。

### [ ... ] ボタン

DB アダプタ電文のフォーマットファイル (DB アダプタ用の XML フォーマット定義ファイル) を設定するダイアログが表示されます。

### [ 表示 ] ボタン

DB アダプタ電文のフォーマット (DB アダプタ用の XML フォーマット定義ファイル) が表示されます。

## ・[ データ変換 ]

### [ データ変換定義 ]

DB アダプタ電文から標準電文へデータ変換するための定義ファイル名を入力します。

### [ 編集 ] ボタン

データ変換定義画面が表示されます。データ変換定義欄で設定した定義ファイルの内容を編集できます。初めて定義する場合は、[ ルート要素選択 ] ダイアログが表示されます。

### [ 削除 ] ボタン

[ 編集 ] ボタンで指定したデータ変換定義を削除します。

## (v) [ フォルト電文 ]

サービス部品のフォルト電文についての情報が表示されます。

### [ 追加 ] ボタン

フォルト名を追加するダイアログが表示されます。DB アダプタの定義では、このボタンは使用しません。

### [ 削除 ] ボタン

フォルト名欄で選択したフォルト名を削除します。DB アダプタの定義では、このボタンは使用しません。

### [ フォルト名 ]

サービス部品のフォルトの名称が表示されます。DB アダプタの定義では、この

欄は使用しません。

[ 電文フォーマット ]

サービス部品のフォルト電文のフォーマット名が表示されます。DB アダプタの定義では、この欄は使用しません。

[ ... ] ボタン

フォルト電文のフォーマットを設定するダイアログが表示されます。DB アダプタの定義では、このボタンは使用しません。

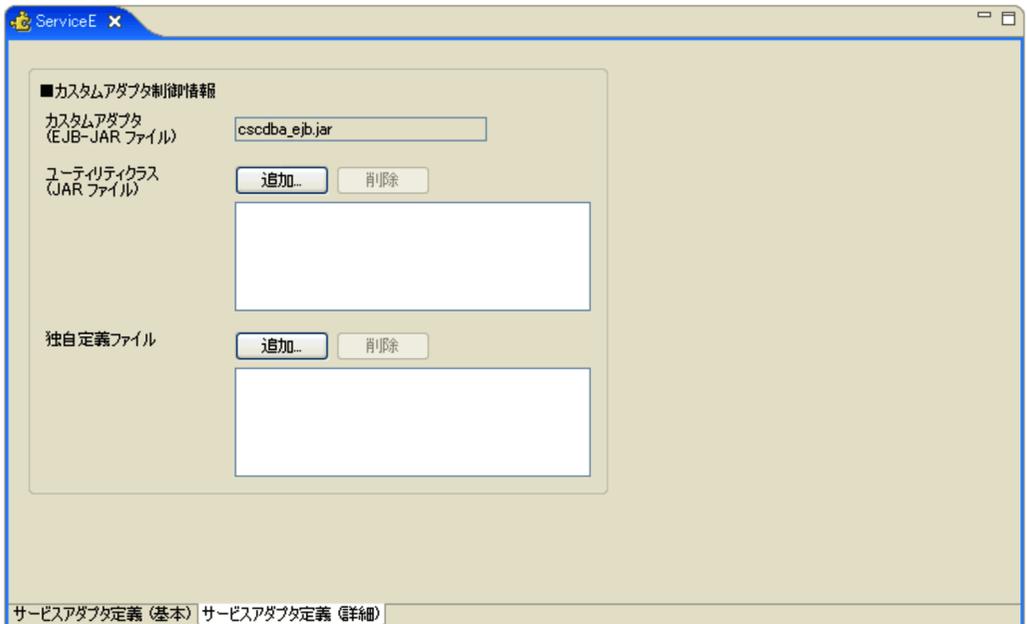
[ 表示 ] ボタン

フォルト電文のフォーマットが表示されます。DB アダプタの定義では、このボタンは使用しません。

(b) カスタムアダプタの詳細画面

カスタムアダプタの詳細画面の項目について説明します。

図 11-14 サービスアダプタ定義画面 (カスタムアダプタの詳細画面)



(i) [ カスタムアダプタ制御情報 ]

DB アダプタ新規作成時に指定したサービス部品についての情報が表示されます。

・ カスタムアダプタ

[ カスタムアダプタ (EJB-JAR ファイル) ]

DB アダプタの EJB-JAR ファイルの名称が表示されます。

## 11. 開発環境で使用する画面

### ・ユーティリティクラス

[ユーティリティクラス (JAR ファイル)]

[追加] ボタンで設定したユーティリティクラスが表示されます。DB アダプタの定義では、この欄は使用しません。

[追加] ボタン

ユーティリティクラスを追加するダイアログが表示されます。DB アダプタの定義では、このボタンは使用しません。

[削除] ボタン

ユーティリティクラス欄で選択したファイルを削除します。DB アダプタの定義では、このボタンは使用しません。

### ・独自定義ファイル

[独自定義ファイル]

[追加] ボタンで設定した SQL オペレーション定義ファイル、およびアプリケーション統合属性ファイルが表示されます。

[追加] ボタン

独自定義ファイル (SQL オペレーション定義ファイルおよびアプリケーション統合属性ファイル) を追加するダイアログが表示されます。

[削除] ボタン

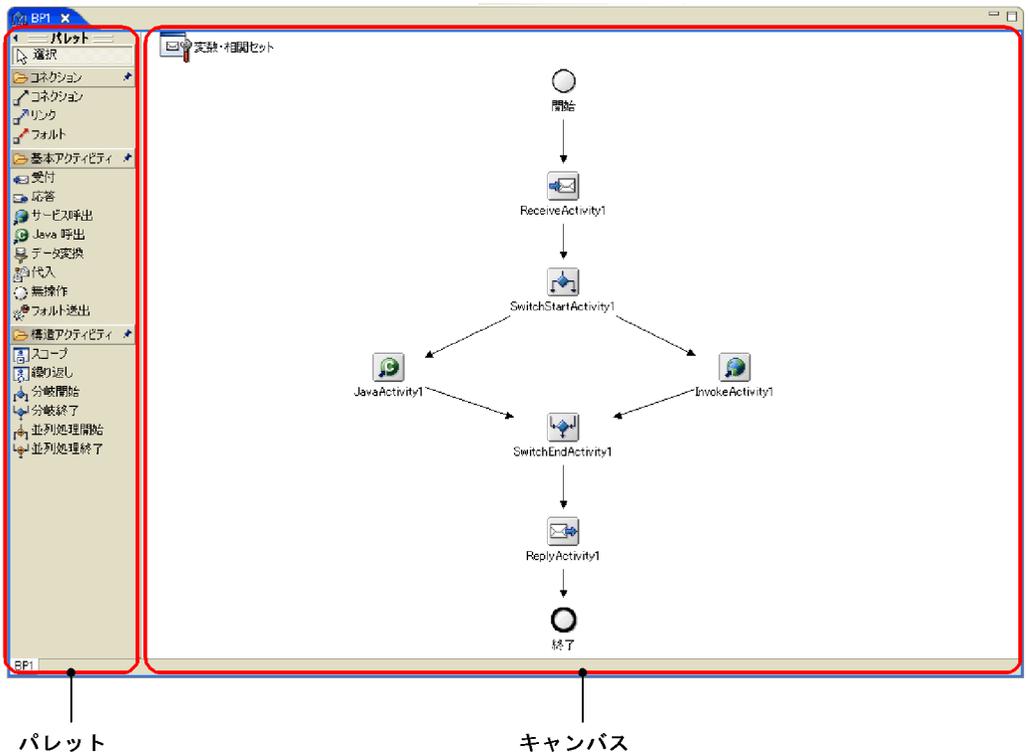
独自定義ファイル欄で選択したファイルを削除します。

## 11.2.3 ビジネスプロセス定義画面

ビジネスプロセスを定義するときのビジネスプロセス定義画面について説明します。

操作方法については、「11.8 ビジネスプロセス定義画面の操作」を参照してください。

図 11-15 ビジネスプロセス定義画面

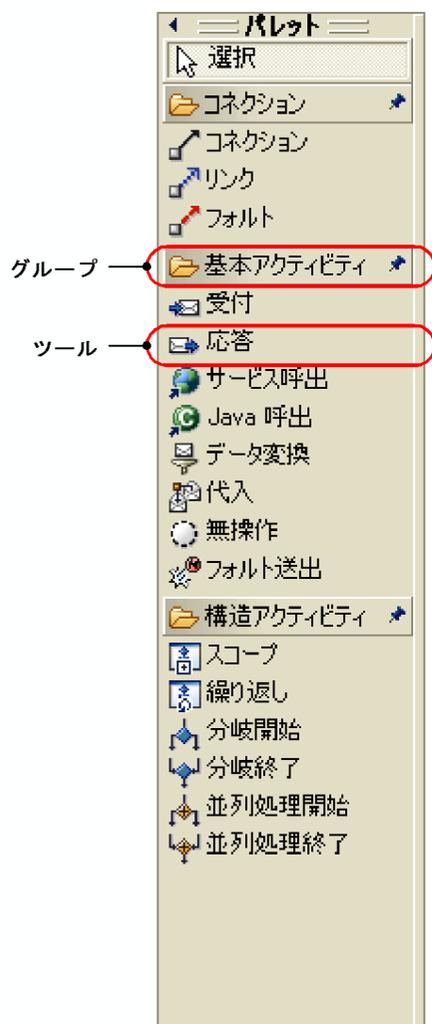


### (1) パレット

キャンバスに配置するアクティビティやコネクションを、パレットのツールから選択します。

パレットの各項目について説明します。

図 11-16 パレット (ビジネスプロセス定義画面)



#### グループ

コネクションまたは同じ種類のアクティビティをまとめたグループです。先頭に  が付いています。

#### ツール

アクティビティやコネクションなどです。選択したアクティビティをキャンバスに配置したり、キャンバスに配置したアクティビティをコネクションで連結したりします。

パレットのグループおよびツールの一覧を次の表に示します。

表 11-5 パレットのグループおよびツール (ビジネスプロセス定義画面)

アイコン	表示名	説明
	選択	キャンバス上のアクティビティおよびコネクションを選択します。
	コネクション	通常のコネクション、リンクコネクション、およびフォルトコネクションをまとめたグループです。
	コネクション	通常のアクティビティを連結するコネクションです。
	リンク	リンクを定義する場合にアクティビティを連結するリンクコネクションです。
	フォルト	フォルト処理を定義する場合にアクティビティを連結するフォルトコネクションです。
	基本アクティビティ	受付アクティビティや応答アクティビティなど、ビジネスプロセスで基本的な定義をするアクティビティをまとめたグループです。
	受付	受付アクティビティです。
	応答	応答アクティビティです。
	サービス呼出	サービス呼出アクティビティです。
	Java 呼出	Java 呼出アクティビティです。
	データ変換	データ変換アクティビティです。
	代入	代入アクティビティです。
	無操作	無操作アクティビティです。
	フォルト送出	フォルト送出アクティビティです。
	構造アクティビティ	スコープアクティビティや繰り返しアクティビティなど、ビジネスプロセスで構造的な定義をするアクティビティをまとめたグループです。
	スコープ	スコープアクティビティです。
	繰り返し	繰り返しアクティビティです。
	分岐開始	分岐開始アクティビティです。
	分岐終了	分岐終了アクティビティです。
	並列処理開始	並列処理開始アクティビティです。

## 11. 開発環境で使用する画面

アイコン	表示名	説明
	並列処理終了	並列処理終了アクティビティです。

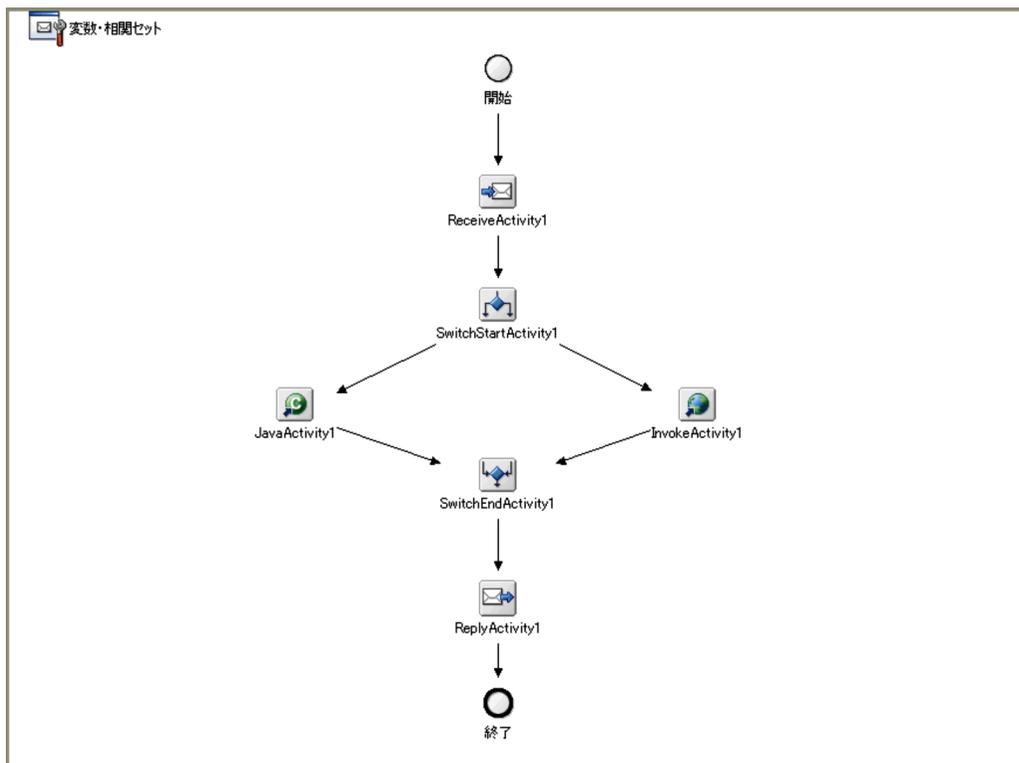
注 詳細については、「付録 D 用語解説」を参照してください。

### (2) キャンバス

キャンバスに、パレットから選択したアクティビティを配置します。また、配置したアクティビティをコネクションで連結します。

キャンバスには、変数・相関セット、開始アクティビティ、および終了アクティビティがすでに配置されています。

図 11-17 キャンバス



変数・相関セット (  )

変数・相関セットを定義する場合に使用します。クリックすると、[変数・相関セット一覧] ダイアログが表示されます。

開始 (  )

開始アクティビティです。このアクティビティは削除できません。

終了 (  )

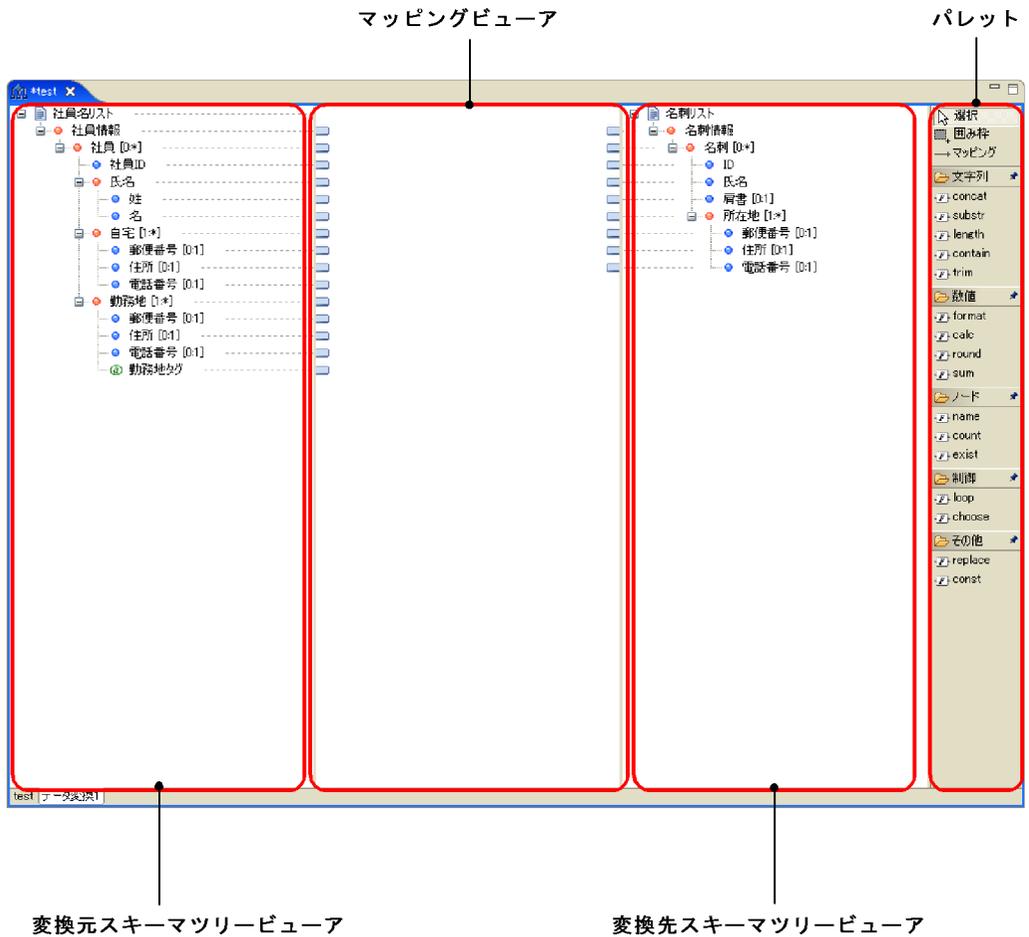
終了アクティビティです。このアクティビティは削除できません。

## 11.2.4 データ変換 (マッピング) 定義画面

データ変換 (マッピング) を定義するときのデータ変換定義画面について説明します。

操作方法については、「11.9 データ変換 (マッピング) 定義画面の操作」を参照してください。

図 11-18 データ変換定義画面



### (1) 変換元スキーマツリービューア, マッピングビューア, および変換先スキーマツリービューア

変換元スキーマツリービューアのノードから変換先スキーマツリービューアのノードへマッピングします。マッピングビューアでは, パレットから選択したファンクションを配置したり, ノードアダプタやファンクションをマッピング線で接続したりします。

変換元スキーマツリービューア, マッピングビューア, および変換先スキーマツリービューアの各項目について説明します。

図 11-19 変換元スキーマツリービューア, マッピングビューア, および変換先スキーマツリービューア



#### ノード

XML スキーマのスキーマ論理名, 要素, および属性です。

ノードに表示されるアイコンおよびアイコンの意味を次に示します。

- スキーマ論理名 ( )

XML スキーマファイルを識別する論理名を表します。

- 複合内容 ( )

複合内容を持つ複合型の要素を表します。

- 単純内容 ( )

単純型の要素, および単純内容を持つ複合型の要素を表します。

- 属性 ( )

要素の属性を表します。

ノードの最後に表示される角括弧 ( [ ] ) の値は, ノードの出現回数を示します。出現回数は次の形式で表示されます。

[ 最小出現回数 : 最大出現回数 ]

最大出現回数に上限がない場合, アスタリスク ( \* ) が表示されます。

最小出現回数, 最大出現回数とともに 1 回の場合, 出現回数は表示されません。

#### ノード対応線

ノードとノードアダプタの対応を示す線です。

### ノードアダプタ

ノードアダプタはすべてのノードに対して表示されます。変換元のノードアダプタと変換先のノードアダプタをマッピング線で接続して、マッピングします。ノードアダプタには、マッピング対象のノードアダプタ (  ) とマッピング対象外のノードアダプタ (  ) の2種類があります。マッピング対象およびマッピング対象外については、「7.8.1 マッピング対象とマッピング対象外」を参照してください。

## (2) パレット

マッピングビューアに配置するファンクションをパレットのツールから選択します。また、マッピング線を選択して、ノードアダプタやファンクションを接続します。

パレットの各項目について説明します。

図 11-20 パレット (データ変換定義画面)



### グループ

同じ種類のファンクションをまとめたグループです。先頭に  が付いています。クリックすると、グループ下のツール一覧が非表示になります。ツール一覧が非表示の場合は、ツールの一覧が表示されます。

### ツール

ファンクションやマッピング線などです。選択したファンクションをマッピングビューアに配置したり、ノードアダプタやファンクションをマッピング線で接続したりします。

パレットのグループおよびツールの一覧を次の表に示します。

表 11-6 パレットのグループおよびツール（データ変換定義画面）

アイコン	表示名	説明
	選択	マッピングビューア上のファンクションおよびマッピング線を選択します。
	囲み枠	四角で囲んだ範囲に含まれるファンクションおよびマッピング線を選択します。
	マッピング	ノードやファンクションを接続するマッピング線です。
	文字列	文字列に関するファンクションをまとめたグループです。
	concat	文字列連結ファンクションです。
	substr	部分文字列取得ファンクションです。
	length	文字数取得ファンクションです。
	contain	文字列調査ファンクションです。
	trim	空白除去ファンクションです。
	数値	数値に関するファンクションをまとめたグループです。
	format	数値フォーマット変換ファンクションです。
	calc	数値演算ファンクションです。
	round	端数処理ファンクションです。
	sum	総和ファンクションです。
	ノード	ノードに関するファンクションをまとめたグループです。
	name	ノード名取得ファンクションです。
	count	ノード数取得ファンクションです。
	exist	ノード調査ファンクションです。

アイコン	表示名	説明
	制御	マッピングを制御するファンクションをまとめたグループです。
	loop	くり返しファンクションです。
	choose	選択ファンクションです。
	その他	どのグループにも当てはまらないファンクションをまとめたグループです。
	const	定数ファンクションです。
	replace	値置換えファンクションです。

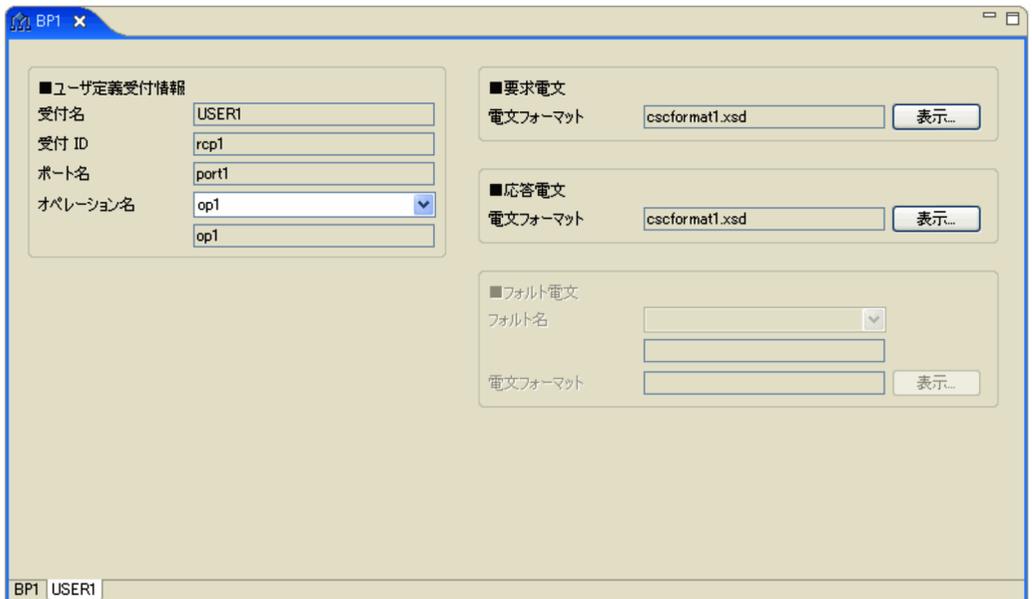
## 11.2.5 ユーザ定義受付定義画面

ユーザ定義受付の定義情報を表示するユーザ定義受付定義画面について説明します。

### ！ 注意事項

追加したユーザ定義受付は変更できません。この画面では、情報の確認だけ行います。

図 11-21 ユーザ定義受付定義画面



BP1 x

■ ユーザ定義受付情報

受付名

受付 ID

ポート名

オペレーション名  ▼

■ 要求電文

電文フォーマット

■ 応答電文

電文フォーマット

■ フォルト電文

フォルト名

電文フォーマット

BP1 USER1

[ ユーザ定義受付情報 ]

[ 受付名 ]

ユーザ定義受付を追加するときにユーザ定義受付追加ウィザードに指定した受付名が表示されます。

[ 受付 ID ]

ユーザ定義受付を追加したときに付与される ID が表示されます。

[ ポート名 ]

ユーザ定義受付を追加するウィザードの [ Port の指定 ] で選択したポート名が表示されます。

[ オペレーション名 ]

オペレーション名が表示されます。ユーザ定義受付に定義されているオペレーション名をドロップダウンリストから選択できます。

[ 要求電文 ]

[ 電文フォーマット ]

要求電文の電文フォーマット名が表示されます。電文フォーマットは、WSDL から自動生成されます。

[ 表示 ] ボタン

要求電文の電文フォーマットが表示されます。

[ 応答電文 ]

[ 電文フォーマット ]

応答電文の電文フォーマット名が表示されます。電文フォーマットは、WSDL から自動生成されます。

[ 表示 ] ボタン

応答電文の電文フォーマットが表示されます。

[ フォルト電文 ]

[ フォルト名 ]

ユーザ定義受付のフォルトの名称が表示されます。

[ 電文フォーマット ]

フォルト電文のフォーマット名が表示されます。電文フォーマットは、WSDL から自動生成されます。

[ 表示 ] ボタン

フォルト電文のフォーマットが表示されます。

## 11.2.6 HCSC コンポーネント情報表示画面

HCSC コンポーネントの情報を表示する HCSC コンポーネント情報表示画面について説明します。

図 11-22 HCSC コンポーネント情報表示画面

The screenshot shows a configuration window titled "ServiceA" with several sections:

- インタフェース情報**
  - サービス名: ServiceA
  - オペレーション: eetUserData
- サーバ情報**
  - サーバ名: MyHCSC
  - 同期: [ ]
  - Web サービス: http://10.209.10.42:80/MyHCSC/s [ WSDL 取得 ]
  - EJB: HITACHI\_EJB/SERVERS/HCSC/ [ スタブ取得 ]
  - 非同期: [ ]
  - MDB/WS-R: http://10.209.10.42:80/ucosminex
  - MDB/DB キュー: CSCHCSCACPT\_DBQ
- オペレーション情報**
  - オペレーション名: eetUserData
  - 通信モデル: 同期
- 要求電文**
  - 標準電文フォーマット: F1.xsd [ 表示... ]
  - フォーマット ID: format1\_web
- 応答電文**
  - 標準電文フォーマット: F2.xsd [ 表示... ]
  - フォーマット ID: format3\_web
- フォルト電文**
  - フォルト名: FAULT1
  - 電文フォーマット: cscformat1.xsd [ 表示... ]

At the bottom left, there is a tab labeled "サービス情報".

## (1) インタフェース情報

インタフェース情報の各項目について説明します。

### [ サービス名 ]

ツリービューの公開サービス一覧で選択した HCSC コンポーネントの名称が表示されます。

### [ オペレーション ]

[ サービス名 ] に表示された HCSC コンポーネントのオペレーション名が表示されます。情報を表示するオペレーションを、ドロップダウンリストから選択します。

## (2) サーバ情報

サーバ情報の各項目について説明します。

### [ サーバ名 ]

サービス部品を呼び出すことができる HCSC サーバが表示されます。情報を表示する HCSC サーバを、ドロップダウンリストから選択します。

### [ Web サービス ]

[ サーバ名 ] で選択した HCSC サーバの Web サービスの受付宛先情報が表示されます。宛先情報には URL が表示されます。

## 11. 開発環境で使用する画面

### [ WSDL 取得 ] ボタン

Web サービスで使用する WSDL ファイルを取得する [ WSDL ファイルの保存 ] ダイアログが表示されます。

### [ EJB ]

[ サーバ名 ] で選択した HCSC サーバの SessionBean の受付宛先情報が表示されます。宛先情報には JNDI 名称が表示されます。

### [ スタブ取得 ] ボタン

SessionBean で使用する RMI-IIOP スタブファイルを取得する [ スタブファイルの保存 ] ダイアログが表示されます。

### [ MDB/WS-R ]

[ サーバ名 ] で選択した HCSC サーバの MDB ( WS-R ) の受付宛先情報が表示されます。宛先情報には URL が表示されます。

### [ MDB/DB キュー ]

[ サーバ名 ] で選択した HCSC サーバの MDB ( DB キュー ) の受付宛先情報が表示されます。宛先情報にはキュー名称が表示されます。

### 注

HCSC サーバがロードバランスクラスタ構成の場合、ロードバランスクラスタ経由で非同期の実行要求 ( MDB ( WS-R ) または MDB ( DB キュー ) ) を受け付けられません。非同期の実行要求は、個々の HCSC サーバで直接受け付けます。

そのため、HCSC サーバがロードバランスクラスタ構成の場合、個々の HCSC サーバの受付宛先情報はコンマ ( , ) 区切りで表示されます。

## ( 3 ) オペレーション情報

[ インタフェース情報 ] の [ オペレーション ] で選択したオペレーションの情報が表示されます。

### [ オペレーション名 ]

[ インタフェース情報 ] の [ オペレーション ] で選択したオペレーション名称が表示されます。

### [ 通信モデル ]

[ インタフェース情報 ] の [ オペレーション ] で選択したオペレーションの通信モデルが表示されます。同期または非同期のどちらかが表示されます。

## ( 4 ) 要求電文

[ インタフェース情報 ] の [ オペレーション ] で選択したオペレーションの情報が表示されます。

[ 標準電文フォーマット ]

標準の要求電文のフォーマット名が表示されます。

[ 表示 ] ボタン

[ 標準電文フォーマット ] に表示された要求電文の内容を表示します。

[ 標準フォーマット ID ]

標準の要求電文のフォーマット ID が表示されます。

## (5) 応答電文

[ インタフェース情報 ] の [ オペレーション ] で選択したオペレーションの情報が表示されます。

[ 標準電文フォーマット ]

標準の応答電文のフォーマット名が表示されます。

[ 表示 ] ボタン

[ 標準電文フォーマット ] に表示された応答電文の内容を表示します。

[ 標準フォーマット ID ]

標準の応答電文のフォーマット ID が表示されます。

## (6) フォルト電文

[ インタフェース情報 ] の [ オペレーション ] で選択したオペレーションの情報が表示されます。

[ フォルト名 ]

フォルト名が表示されます。情報を表示するフォルト名を、ドロップダウンリストから選択します。

[ 電文フォーマット ]

[ フォルト名 ] で選択した電文のフォーマット名が表示されます。

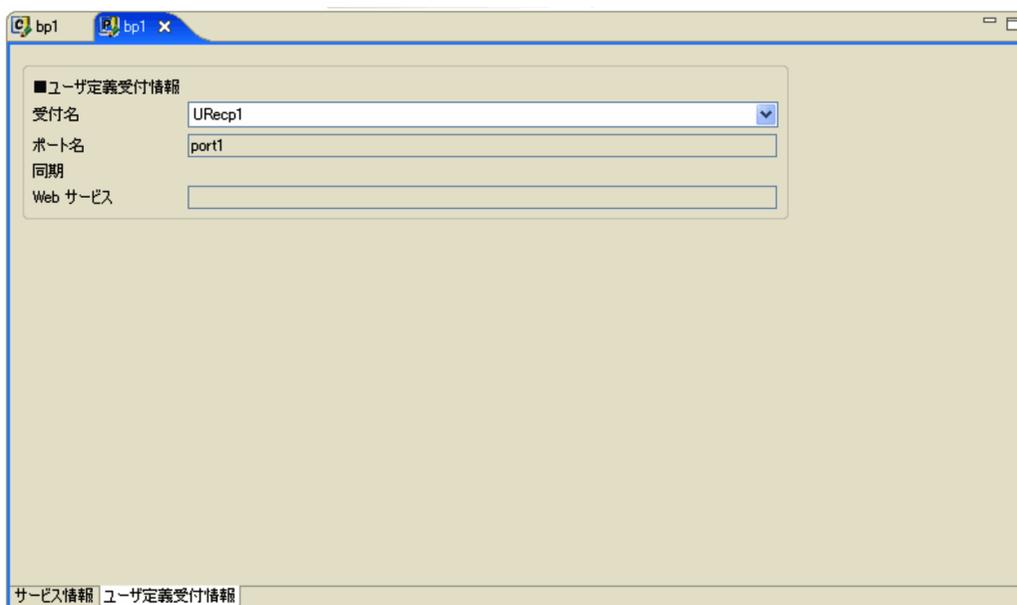
[ 表示 ] ボタン

[ 電文フォーマット ] に表示された電文の内容を表示します。

## 11.2.7 ユーザ定義受付情報表示画面

パッケージング後のユーザ定義受付の定義情報を表示するユーザ定義受付情報表示画面について説明します。

図 11-23 ユーザ定義受付情報表示画面



[ ユーザ定義受付情報 ]

[ 受付名 ]

ユーザ定義受付を追加するときにユーザ定義受付追加ウィザードに指定した受付名が表示されます。

[ ポート名 ]

ユーザ定義受付を追加するときにユーザ定義受付追加ウィザードに指定したポート名が表示されます。

[ 同期 ]

情報は表示されません。

[ Web サービス ]

WSDL のサービスロケーションの値 ( wsdl:port 要素内の soap-address 要素の location 属性に設定している値 ) に設定する URL が表示されます。

サービスリクエストを作成する前に、WSDL を編集して、この項目に表示された値をサービスロケーションの値として設定します。

WSDL の編集については、「10.6.2 WSDL を編集する」を参照してください。

## 11.3 バイナリフォーマット定義に関連するダイアログ

バイナリフォーマット定義に関連するダイアログについて説明します。

### 11.3.1 フォーマットダイアログ

フォーマットダイアログは、バイナリフォーマット定義ファイルのフォーマット名称や文字コードを設定するときに使用します。

操作手順については、「4.4.2(1) フォーマット情報を設定する」を参照してください。

図 11-24 フォーマットダイアログ

使用するバイナリデータがCSV形式以外の場合



使用するバイナリデータがCSV形式の場合



#### [ インフォメーション ]

フォーマットダイアログの入力情報に問題がある場合に、ダイアログの最上段に問題点を示すメッセージが表示されます。表示されるメッセージと対策については、「4.4.5(1) 検証内容」を参照してください。

#### [ フォーマット名称 ]

バイナリフォーマット定義ファイルのフォーマット名称を指定します。NCName型で256文字以内で指定します。

#### [ 文字コード ]

バイナリデータで使用される文字コードをドロップダウンリストから選択します。

#### [ フォーマット ]

使用するバイナリデータがCSV形式の場合、「CSV形式」と表示されます。CSV以

## 11. 開発環境で使用する画面

外の場合は表示されません。

### [ 2 進整数のエンディアン ]

バイナリデータに含まれる 2 進整数のエンディアンをラジオボタンで選択します。  
この項目は、使用するバイナリデータが CSV 形式以外の場合にだけ選択できます。

### [ セパレータ ]

#### [ エスケープ文字 ]

セパレータとして使用する文字がデータ内に含まれる場合に、エスケープ文字として使用する文字を指定します。エスケープ文字として使用できる文字は、1 文字のラテン文字 (JIS X 0201 の「0x21」～「0x7E」) または半角スペース (JIS X 0201 の「0x20」) です。

この項目は、使用するバイナリデータが CSV 形式以外の場合に指定できます。

#### [ 改行指定 ]

セパレータで使用する改行文字をドロップダウンリストで選択します。使用できる改行コードは、CR + LF、または LF です。

#### [ 不要なセパレータの削除 ]

末尾のセパレータを出力する場合に、チェックボックスをチェックします。

#### [ 出力時のダブルクォーテーション「"」の囲み ]

データ項目をダブルクォーテーションで囲むかどうかを指定します。

この項目は、使用するバイナリデータが CSV 形式の場合に指定できます。

### [ OK ] ボタン

指定した内容を設定してフォーマットダイアログを閉じます。

### [ キャンセル ] ボタン

指定した内容を設定しないでフォーマットダイアログを閉じます。

## 11.3.2 単純内容要素ダイアログ

単純内容要素ダイアログは、バイナリフォーマット定義ファイルに単純内容要素を設定するときに使用します。

操作手順については、「4.4.2(2) グローバル定義の単純内容要素を設定する」、または「4.4.2(4)(b) ローカル定義の単純内容要素を構成要素として設定する」を参照してください。

単純内容要素ダイアログは、グローバル定義の単純内容要素を設定する場合と、ローカル定義の単純内容要素を設定する場合で表示されるタブが異なります。

### (1) 単純内容要素タブ

単純内容要素タブは、設定する単純内容要素がグローバル定義でもローカル定義でも表示されます。単純内容要素の情報を指定します。

図 11-25 単純内容要素ダイアログ (単純内容要素タブ)



## [ データ種別 ]

単純内容要素として使用するデータの種別をドロップダウンリストから選択します。選択できるデータ種別を次に示します。

- 文字列
- 整数
- 実数
- 固定小数部数値
- ゾーン形式数値
- パック形式数値
- 符号付 2 進整数
- 符号無 2 進整数
- バイト列
- 日付時刻

## [ サイズ ]

単純内容要素として使用するデータのサイズを指定します。

- 固定値の場合  
ラジオボタンで [ 固定 ] を選択し、サイズ (バイト) を指定します。  
データ種別と文字コードによって指定できるサイズの範囲が異なります。指定できるサイズの範囲を次の表に示します。

表 11-7 固定値の単純内容要素に指定できるサイズの範囲

データ種別		文字コード	指定できる範囲 (バイト)
文字列, 整数, 実数, 固定小数部数値		UTF16 以外	1 ~ 2,147,483,647
		UTF16	2 ~ 2,147,483,646
ゾーン形式数値, パック形式数値		-	1 ~ 2,147,483,647
符号付 2 進整数, 符号無 2 進整数		-	1, 2, 4, 8 のどれか
バイト列		-	1 ~ 2,147,483,647
日付時刻	CCYYMMDD	UTF16 以外	8 ~ 2,147,483,647
		UTF16	16 ~ 2,147,483,646
	YYMMDD	UTF16 以外	6 ~ 2,147,483,647
		UTF16	12 ~ 2,147,483,646
	hhmmss	UTF16 以外	6 ~ 2,147,483,647
		UTF16	12 ~ 2,147,483,646
	CCYYMMDDhhmmss	UTF16 以外	14 ~ 2,147,483,647
		UTF16	28 ~ 2,147,483,646
	YYMMDDhhmmss	UTF16 以外	12 ~ 2,147,483,647
		UTF16	24 ~ 2,147,483,646

(凡例)

- : 指定できる範囲が文字コードに依存しないことを示します。

## • 可変値の場合

ラジオボタンで [ 可変 ] を選択します。サイズの決定方法は、構成要素タブ (または構成要素ダイアログ) の [ サイズ ] で設定します。

## [ 文字列 ]

データ種別が文字列の場合で、サイズが固定のとき、埋め字の種類を設定します。

## [ 埋め字 ]

単純内容要素として使用するデータが指定したサイズに満たない場合、余った部分を埋める文字をドロップダウンリストから選択します。空白で埋める場合は [ スペース ] を、空文字 (0x00) で埋める場合は [ 空文字 ] を選択します。

## [ 数値 ]

データ種別が固定小数部数値, ゾーン形式数値, またはパック形式数値の場合, 詳細を設定します。

## [ 小数部桁数 ]

小数部分の桁数を 0 ~ 33 で指定します。

## [ 符号の有無 ]

データ種別が固定小数部数値の場合、正負の符号の有無をラジオボタンで選択します。

- 符号がない場合  
[ なし ] を選択します。
- 負の符号だけある場合  
[ 負だけあり ] を選択します。
- 常に正負の符号がある場合  
[ 常にあり ] を選択します。

## [ 埋め字 ]

データ種別が固定小数部数値の場合、単純内容要素として使用するデータが指定したサイズに満たないとき、余った部分を埋める文字をラジオボタンで選択します。空白で埋める場合は [ スペース ] を、0 で埋める場合は [ 0 ] を選択します。

## [ 日付時刻 ]

データ種別が日付時刻の場合、日付時刻の形式を設定します。

## [ 形式 ]

日付時刻の形式をドロップダウンリストから選択します。

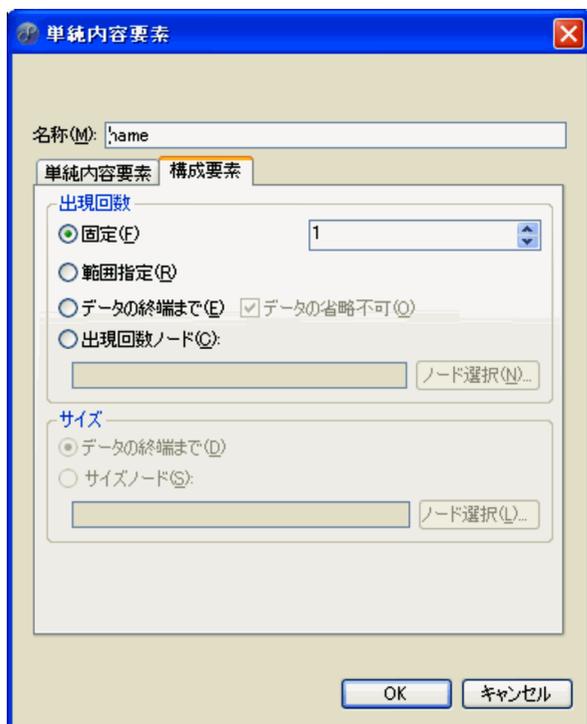
## [ 秒小数部桁数 ]

日付時刻の形式に時刻 (hhmmss) が含まれる場合、秒の小数部分の桁数を 0 ~ 3 で指定します。

## (2) 構成要素タブ

構成要素タブは、ローカル定義の単純内容要素を設定する場合にだけ表示されます。構成要素の情報を指定します。

図 11-26 単純内容要素ダイアログ (構成要素タブ)



指定する内容は、構成要素ダイアログの構成要素タブと同じです。

構成要素ダイアログの構成要素タブについては、「11.3.4 構成要素ダイアログ」を参照してください。

### (3) 各タブで共通の内容

各タブで共通の内容を次に示します。

#### [ インフォメーション ]

単純内容要素ダイアログの入力情報に問題がある場合に、ダイアログの最上段に問題点を示すメッセージが表示されます。表示されるメッセージと対策については、「4.4.5(1) 検証内容」を参照してください。

#### [ 名称 ]

単純内容要素の名称を指定します。NCName 型で利用できる文字を利用して 256 文字以内で指定します。

#### [ OK ] ボタン

指定した内容を設定して単純内容要素ダイアログを閉じます。

#### [ キャンセル ] ボタン

指定した内容を設定しないで単純内容要素ダイアログを閉じます。

### 11.3.3 複合内容要素ダイアログ

複合内容要素ダイアログは、バイナリフォーマット定義ファイルに複合内容要素を設定するときに使用します。また、複合内容要素にセパレータを設定するときにも使用します。

操作手順については、「4.4.2(3) 複合内容要素を設定する」または「4.4.2(7) セパレータを設定する」を参照してください。

複合内容要素ダイアログは、複合内容要素を設定する場合と、セパレータを設定する場合で操作するタブが異なります。

#### (1) 複合内容要素タブ

複合内容要素タブは、複合内容要素の情報を指定するときに使用します。

構成区分が [ 選択構成 ] の場合、利用する構成要素を何から選択するかによって、表示される画面が変わります。

図 11-27 複合内容要素ダイアログの複合内容要素タブ（構成要素を選択条件ノードの値から選択する場合）

名称(M): Sequence02

複合内容要素 | セパレータ

構成区分

逐次構成(S)

選択構成(C)

選択種別

選択条件ノード(Q):

Sequence01/ID      ノード選択(N)...

構成要素の開始セパレータ(E)

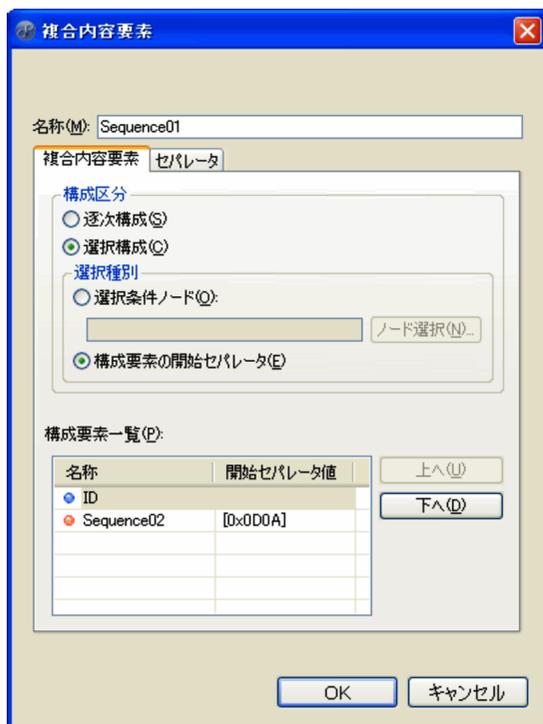
構成要素一覧(P):

名称	選択条件値
<input checked="" type="radio"/> name	1
<input checked="" type="radio"/> data	5

上へ(U)      下へ(D)      条件値設定(V)...

OK      キャンセル

図 11-28 複合内容要素ダイアログの複合内容要素タブ（構成要素を開始セパレータの値から選択する場合）



[ 構成区分 ]

複合内容要素に設定された構成要素の構成区分をラジオボタンで選択します。

- 逐次構成の場合  
[ 逐次構成 ] を選択します。
- 選択構成の場合  
[ 選択構成 ] を選択します。

[ 選択種別 ]

構成区分が [ 選択構成 ] の場合、選択構成の選択種別をラジオボタンで選択します。

- 選択条件ノードの値から、利用する構成要素を選択する場合  
[ 選択条件ノード ] を選択し、[ ノード選択 ] ボタンをクリックして選択条件ノードを指定します。
- 構成要素の開始セパレータの値から、利用する構成要素を選択する場合  
[ 構成要素の開始セパレータ ] を選択します。

[ ノード選択 ] ボタン

ノード選択ダイアログが表示されます。構成要素の選択条件ノードを選択するときに使用します。

ノード選択ダイアログで選択したノードは次のように表示されます。

- 基準パスで選択した場合

基準ノードからの相対パスが表示されます。

- 絶対パスで選択した場合  
ルート要素からのパスが表示されます。

#### [ 構成要素一覧 ]

複合内容要素に設定された構成要素の一覧が表示されます。

#### [ 名称 ]

構成要素に設定されている単純内容要素および複合内容要素の名称が表示されます。

#### [ 選択条件値 ]

構成区分が [ 選択構成 ] で選択種別が [ 選択条件ノード ] の場合、構成要素に設定されている選択条件値が表示されます。構成区分が [ 逐次構成 ] の場合、および構成区分が [ 選択構成 ] で選択種別が [ 構成要素の開始セパレータ ] の場合、この項目は表示されません。

構成区分が [ 選択構成 ] で選択種別が [ 選択条件ノード ] の場合、[ 選択条件値 ] に表示されているどの値にも当てはまらないときは、選択条件値が表示されていない構成要素が出現します。このような構成要素は、構成要素の中で一つだけ設定できます。

#### [ 開始セパレータ値 ]

構成区分が [ 選択構成 ] で選択種別が [ 構成要素の開始セパレータ ] の場合、構成要素に設定されている開始セパレータ値が表示されます。構成区分が [ 逐次構成 ] の場合、および構成区分が [ 選択構成 ] で選択種別が [ 選択条件ノード ] の場合、この項目は表示されません。

構成区分が [ 選択構成 ] で選択種別が [ 構成要素の開始セパレータ ] の場合で、[ 開始セパレータ値 ] に表示されているどの値にも当てはまらないとき、次に示す構成要素が出現します。

- 開始セパレータ値が表示されていない構成要素
- 単純内容要素である構成要素

このような構成要素は、構成要素の中で一つだけ設定できます。

#### [ 上へ ] ボタン

構成要素一覧で選択した構成要素の配置を一つ上に変更します。

#### [ 下へ ] ボタン

構成要素一覧で選択した構成要素の配置を一つ下に変更します。

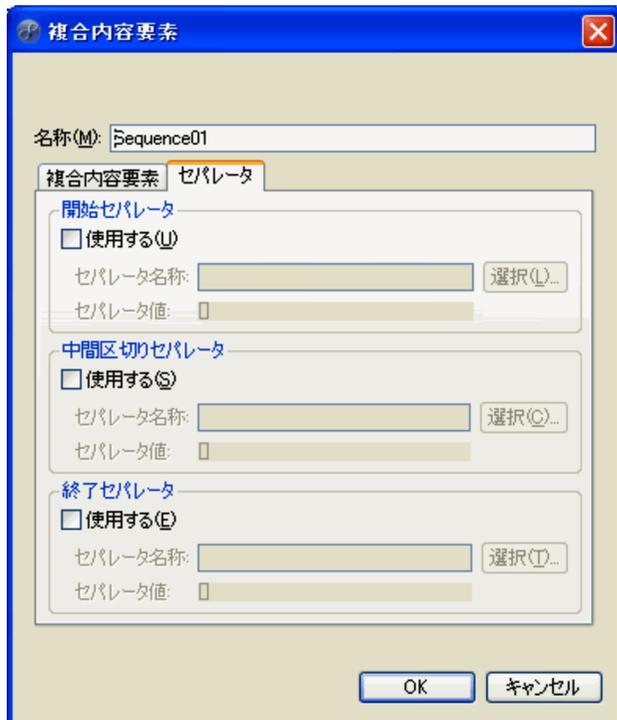
#### [ 条件値設定 ] ボタン

構成区分が [ 選択構成 ] で選択種別が [ 選択条件ノード ] の場合、条件値設定ダイアログが表示されます。構成区分が [ 逐次構成 ] の場合、および [ 選択構成 ] で選択種別が [ 構成要素の開始セパレータ ] の場合、このボタンは表示されません。構成要素一覧で選択した構成要素の選択条件値を設定するときに使用します。

## (2) セパレータタブ

セパレータタブは、複合内容要素にセパレータを設定する場合に使用します。

図 11-29 複合内容要素ダイアログ (セパレータタブ)



開始セパレータ、中間区切りセパレータ、終了セパレータとして使用するセパレータを、次の項目を利用してそれぞれ指定します。

### [ 使用する ] チェックボックス

各セパレータを使用する場合にチェックします。

### [ セパレータ名称 ]

使用するセパレータの名称を指定します。

### [ 選択 ] ボタン

セパレータ選択ダイアログが表示されます。

### [ セパレータ値 ]

[ セパレータ名称 ] を指定した場合、定義されているセパレータの値が表示されます。セパレータの値は [ ] で囲まれて表示されます。

## (3) 各タブで共通の内容

各タブで共通の内容を次に示します。

## [ インフォメーション ]

複合内容要素ダイアログの入力情報に問題がある場合に、ダイアログの最上段に問題点を示すメッセージが表示されます。表示されるメッセージと対策については、「4.4.5(1) 検証内容」を参照してください。

## [ 名称 ]

複合内容要素の名称を指定します。NCName 型で利用できる文字を利用して 256 文字以内で指定します。

## [ OK ] ボタン

指定した内容を設定して複合内容要素ダイアログを閉じます。

## [ キャンセル ] ボタン

指定した内容を設定しないで複合内容要素ダイアログを閉じます。

### 11.3.4 構成要素ダイアログ

構成要素ダイアログは、グローバル定義の単純内容要素または複合内容要素を、複合内容要素の構成要素に設定するときに使用します。

操作手順については、「4.4.2(4)(a) グローバル定義の要素を構成要素に設定する」を参照してください。

構成要素ダイアログは、グローバル定義の単純内容要素を構成要素に設定する場合と、複合内容要素を構成要素に設定する場合で表示されるタブが異なります。

#### (1) 単純内容要素タブ

単純内容要素タブは、単純内容要素を構成要素に設定する場合にだけ表示されます。構成要素に設定する単純内容要素の情報が表示されます。内容は変更できません。

図 11-30 構成要素ダイアログ (単純内容要素タブ)



表示される情報は、構成要素に設定するグローバル定義の単純内容要素を設定するとき、単純内容要素ダイアログの単純内容要素タブで設定した値です。

単純内容要素ダイアログの単純内容要素タブについては、「11.3.2 単純内容要素ダイアログ」を参照してください。

## (2) 複合内容要素タブ

複合内容要素タブは、複合内容要素を構成要素に設定する場合にだけ表示されます。構成要素に設定する複合内容要素の情報が表示されます。内容は変更できません。

図 11-31 構成要素ダイアログ（複合内容要素タブ）



表示される情報は、構成要素に設定する複合内容要素を設定するときに、複合内容要素ダイアログで設定した値です。

複合内容要素ダイアログについては、「11.3.3 複合内容要素ダイアログ」を参照してください。

### (3) 構成要素タブ

構成要素タブは、構成要素に設定する要素がグローバル定義の単純内容要素でも複合内容要素でも表示されます。構成要素の情報を指定します。

図 11-32 構成要素ダイアログ (構成要素タブ)



## [ 出現回数 ]

構成要素の出現回数を指定します。

- 出現回数が固定の場合  
ラジオボタンで [ 固定 ] を選択し, [ 最小 ] に出現回数を 1 ~ 2,147,483,647 で指定します。バイナリデータとして CSV 形式のデータを扱う場合, CSV データにヘッダーがないときは 0 を, ヘッダーがあるときは 1 を設定します。
- 出現回数を範囲で指定する場合  
ラジオボタンで [ 範囲指定 ] を選択し, [ 最小 ] と [ 最大 ] で出現回数の範囲を指定します。設定できる範囲は, 1 ~ 2,147,483,647 です。
- データの端末までを要素とする場合  
ラジオボタンで [ データの端末まで ] を選択します。なお, 必ず 1 回以上出現させる場合は, [ データの省略不可 ] チェックボックスにチェックします。
- 出現回数をほかの要素に依存させる場合  
ラジオボタンで [ 出現回数ノード ] を選択し, [ ノード選択 ] ボタンをクリックして依存させるノードを選択します。

## [ ノード選択 ] ボタン

ノード選択ダイアログが表示されます。構成要素の出現回数をほかの要素に依存させる場合, 依存させるノードを選択するときに使用します。

ノード選択ダイアログで選択したノードは次のように表示されます。

- 基準パスで選択した場合

基準ノードからの相対パスが表示されます。

- 絶対パスで選択した場合  
ルート要素からのパスが表示されます。

#### [ サイズ ]

単純内容要素タブの [ サイズ ] で [ 可変 ] を選択した場合に、構成要素である単純内容要素として使用するデータのサイズを設定します。

- データの終端までを要素とする場合  
ラジオボタンで [ データの終端まで ] を選択します。
- サイズをほかの要素に依存させる場合  
ラジオボタンで [ サイズノード ] を選択し、[ ノード選択 ] ボタンをクリックして依存させるノードを選択します。

#### [ ノード選択 ] ボタン

ノード選択ダイアログが表示されます。構成要素である単純内容要素のサイズをほかの要素に依存させる場合、依存させるノードを選択するときに使用します。

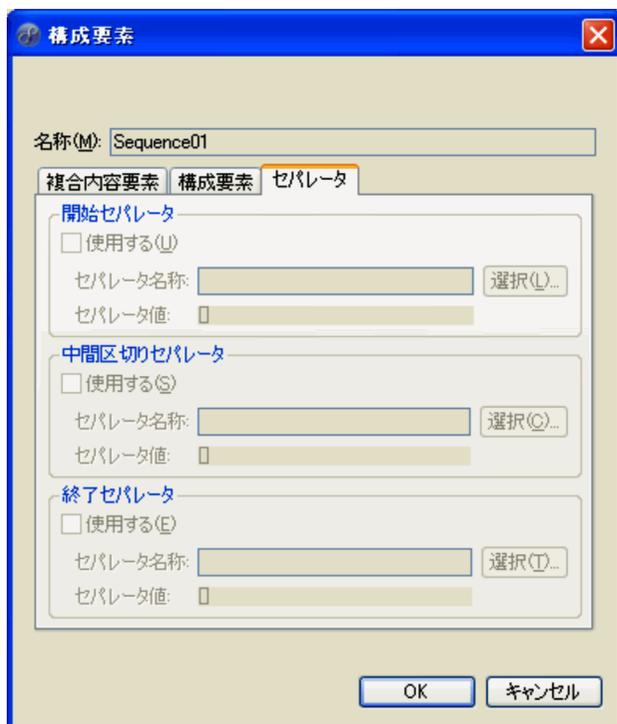
ノード選択ダイアログで選択したノードは次のように表示されます。

- 基準パスで選択した場合  
基準ノードからの相対パスが表示されます。
- 絶対パスで選択した場合  
ルート要素からのパスが表示されます。

### (4) セパレータタブ

セパレータタブは、複合内容要素を構成要素に設定する場合にだけ表示されます。構成要素に設定する複合内容要素のセパレータの情報が表示されます。内容は変更できません。

図 11-33 構成要素ダイアログ (セパレータタブ)



表示される情報は、構成要素に設定する複合内容要素にセパレータを設定するときに、複合内容要素ダイアログで設定した値です。

複合内容要素ダイアログについては、「11.3.3 複合内容要素ダイアログ」を参照してください。

#### (5) 各タブで共通の内容

各タブで共通の内容を次に示します。

##### [ インフォメーション ]

構成要素ダイアログの入力情報に問題がある場合に、ダイアログの最上段に問題点を示すメッセージが表示されます。表示されるメッセージと対策については、「4.4.5(1) 検証内容」を参照してください。

##### [ 名称 ]

構成要素に設定する要素の名称が表示されます。

##### [ OK ] ボタン

指定した内容を設定して構成要素ダイアログを閉じます。

##### [ キャンセル ] ボタン

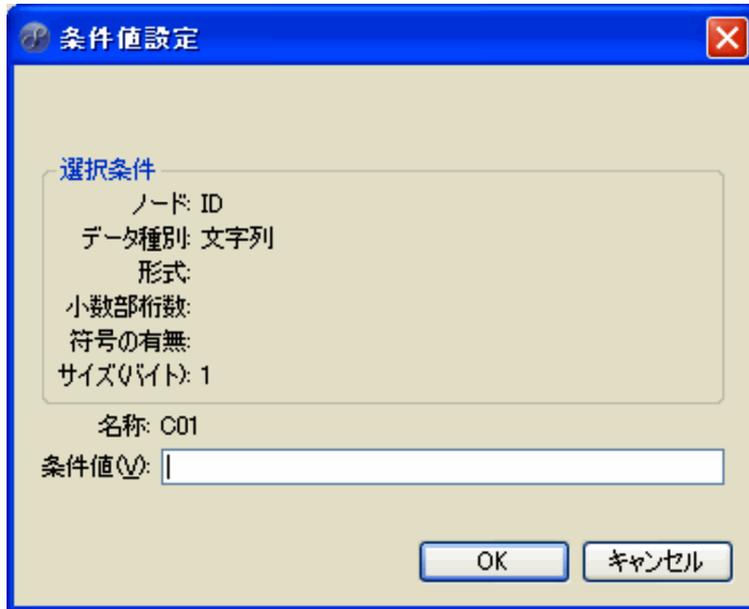
指定した内容を設定しないで構成要素ダイアログを閉じます。

### 11.3.5 条件値設定ダイアログ

条件値設定ダイアログは、選択構成の複合内容要素の構成要素に選択条件値を設定するとき 사용됩니다。

操作手順については、「4.4.2(5) 構成要素の選択条件を設定する」を参照してください。

図 11-34 条件値設定ダイアログ



#### [ インフォメーション ]

条件値設定ダイアログの入力情報に問題がある場合に、ダイアログの最上段に問題点を示すメッセージが表示されます。表示されるメッセージと対策については、「4.4.5(1) 検証内容」を参照してください。

#### [ 選択条件 ]

##### [ ノード ]

選択条件ノードが表示されます。

##### [ データ種別 ]

選択条件ノードのデータ種別が表示されます。

##### [ 形式 ]

選択条件ノードのデータ種別が日付時刻の場合、日付時刻の形式が表示されます。

##### [ 小数部桁数 ]

選択条件ノードのデータ種別が固定小数部数値、ゾーン形式数値、パック形式数値、または日付時刻（時刻を含む形式）の場合、小数部分の桁数が表示されます。

## 11. 開発環境で使用する画面

### [ 符号の有無 ]

選択条件ノードのデータ種別が整数，実数，または固定小数部数値の場合，符号の有無が表示されます。

### [ サイズ (バイト) ]

選択条件ノードのサイズが固定値の場合，その値が表示されます。可変値の場合には「可変値」と表示されます。

### [ 名称 ]

選択条件値を設定する構成要素の名称が表示されます。

### [ 条件値 ]

選択条件値を 1,024 文字以内で指定します。指定できる条件値は，選択条件ノードのデータ種別によって異なります。

指定できる条件値については，「4.4.5(1) 検証内容」を参照してください。

### [ OK ] ボタン

指定した内容を設定して条件値設定ダイアログを閉じます。

### [ キャンセル ] ボタン

指定した内容を設定しないで条件値設定ダイアログを閉じます。

## 11.3.6 ノード選択ダイアログ (バイナリフォーマット定義)

ノード選択ダイアログは，構成要素の出現回数やサイズをほかの要素から参照する場合のターゲットノードや，構成要素の選択条件値を持つ選択条件ノードを選択するときに使用します。

次のダイアログにある [ ノード選択 ] ボタンをクリックした場合に表示されます。

- 単純内容要素ダイアログ (構成要素タブ)
- 構成要素ダイアログ (構成要素タブ)
- 複合内容要素ダイアログ

図 11-35 ノード選択ダイアログ (バイナリフォーマット定義)



## [ インフォメーション ]

ノード選択ダイアログの入力情報に問題がある場合に、ダイアログの最上段に問題点を示すメッセージが表示されます。表示されるメッセージと対策については、「4.4.5(1) 検証内容」を参照してください。

## [ パス指定 ]

ターゲットノードまたは選択条件ノードの指定方法をラジオボタンで選択します。

## [ 基準パス ]

基準ノードからのパスでターゲットノードまたは選択条件ノードを指定する場合に選択します。

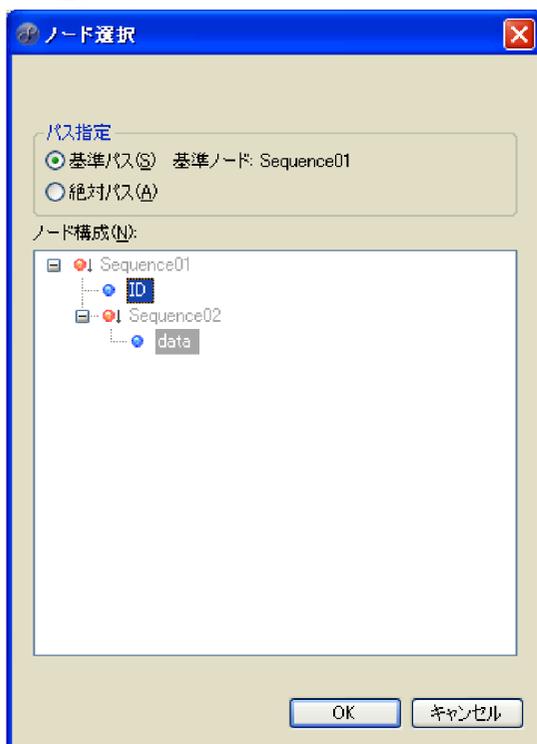
基準ノードとは、参照元の構成要素（依存ノード）がツリー全体に複数ある場合に、共通するツリー構成部分の最上位ノードです。

[ 基準パス ] を選択すると、[ ノード構成 ] には基準ノードからの構成がツリー表示されます。

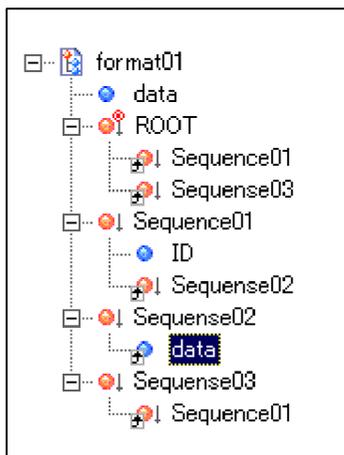
[ 基準パス ] を選択した場合の例を次の図に示します。

図 11-36 基準パスを選択した場合の例

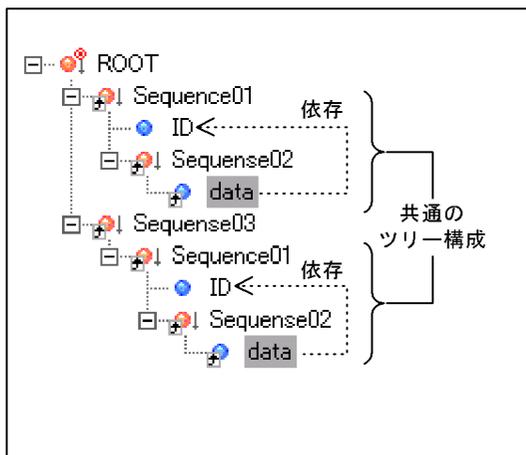
・ノード選択ダイアログ



・フォーマット定義エディタ



・アウトラインビュー



[ 絶対パス ]

ルート要素からのパスでターゲットノードまたは選択条件ノードを指定する場合に選択します。

[ 絶対パス ] を選択すると , [ ノード構成 ] にはルート要素からの構成がツリー表示されます。

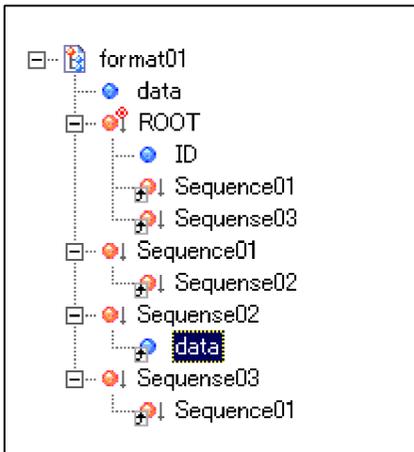
[絶対パス] を選択した場合の例を次の図に示します。

図 11-37 絶対パスを選択した場合の例

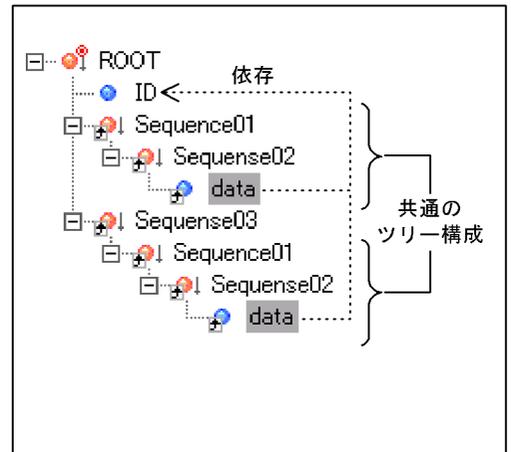
・ノード選択ダイアログ



・フォーマット定義エディタ



・アウトラインビュー



[ノード構成]

[パス指定] の選択に従って、ノードの構成がツリー表示されます。表示されたツ

## 11. 開発環境で使用する画面

リーからターゲットノードまたは選択条件ノードを選択します。

### [ OK ] ボタン

指定した内容を設定してノード選択ダイアログを閉じます。

### [ キャンセル ] ボタン

指定した内容を設定しないでノード選択ダイアログを閉じます。

## 11.3.7 要素選択ダイアログ

要素選択ダイアログは、構成要素として設定するグローバル定義の要素を選択するときに使用します。

操作手順については、「4.4.2(4)(a) グローバル定義の要素を構成要素に設定する」を参照してください。

図 11-38 要素選択ダイアログ



### [ インフォメーション ]

要素選択ダイアログの入力情報に問題がある場合に、ダイアログの最上段に問題点を示すメッセージが表示されます。表示されるメッセージと対策については、「4.4.5(1) 検証内容」を参照してください。

### [ 要素一覧 ]

構成要素として設定する要素を選択します。

## [ 名称 ]

構成要素として設定できるグローバル定義の要素が表示されます。

## [ 要素 ]

単純内容要素の場合は「単純内容」、複合内容要素の場合は「複合内容」と表示されます。

## [ OK ] ボタン

指定した内容を設定して要素選択ダイアログを閉じます。

## [ キャンセル ] ボタン

指定した内容を設定しないで要素選択ダイアログを閉じます。

### 11.3.8 セパレータ選択ダイアログ

セパレータ選択ダイアログは、複合内容要素の開始セパレータ、中間区切りセパレータおよび終了セパレータとして設定するセパレータを選択するダイアログです。

複合内容要素ダイアログ（セパレータタブ）の、各セパレータの指定欄にある [ 選択 ] ボタンをクリックした場合に表示されます。

図 11-39 セパレータ選択ダイアログ



## [ インフォメーション ]

セパレータ選択ダイアログの入力情報に問題がある場合に、ダイアログの最上段に

## 11. 開発環境で使用する画面

問題点を示すメッセージが表示されます。表示されるメッセージと対策については、「4.4.5(1) 検証内容」を参照してください。

### [ セパレータ名称一覧 ]

#### [ 名称 ]

定義されているセパレータの名称が表示されます。

#### [ 値 ]

定義されているセパレータの値が表示されます。セパレータの値は [ ] で囲まれて表示されます。

### [ 追加 ] ボタン

セパレータとして使用する文字を新たに追加するときに使用します。セパレータ追加 / 変更ダイアログが表示されます。

### [ 変更 ] ボタン

すでに定義されているセパレータの名称およびセパレータの値を変更するときに使用します。[ セパレータ名称一覧 ] から情報を変更するセパレータを選択してクリックすると、セパレータ追加 / 変更ダイアログが表示されます。

なお、次のセパレータの情報は変更できません。

- 「LineFeed」および「Comma」(初期設定されているセパレータ)
- 同じ複合内容要素の中で、すでにほかのセパレータとして指定されているセパレータ

### [ 削除 ] ボタン

すでに定義されているセパレータを削除するときに使用します。[ セパレータ名称一覧 ] から削除するセパレータを選択してクリックすると、選択したセパレータが削除されます。

なお、次のセパレータは削除できません。

- 「LineFeed」および「Comma」(初期設定されているセパレータ)
- 同じ複合内容要素の中で、すでにほかのセパレータとして指定されているセパレータ

### [ OK ] ボタン

指定した内容を設定してセパレータ選択ダイアログを閉じます。

### [ キャンセル ] ボタン

指定した内容を設定しないでセパレータ選択ダイアログを閉じます。

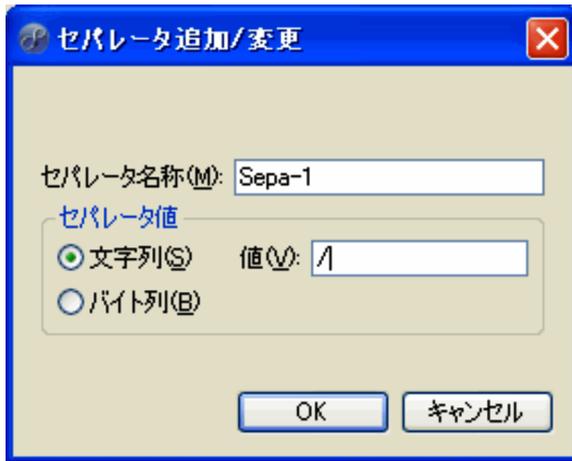
## 11.3.9 セパレータ追加 / 変更ダイアログ

セパレータ追加 / 変更ダイアログは、複合内容要素の開始セパレータ、中間区切りセパレータおよび終了セパレータとして使用するセパレータを追加したり、セパレータの情報を変更したりするダイアログです。

セパレータ選択ダイアログの、[ 追加 ] ボタンまたは [ 変更 ] ボタンをクリックした場合

に表示されます。

図 11-40 セパレータ追加 / 変更ダイアログ



#### [ インフォメーション ]

セパレータ追加 / 変更ダイアログの入力情報に問題がある場合に、ダイアログの最上段に問題点を示すメッセージが表示されます。表示されるメッセージと対策については、「4.4.5(1) 検証内容」を参照してください。

#### [ セパレータ名称 ]

セパレータの名称を指定します。NCName 型で 256 文字以内で指定します。

すでに定義されているセパレータの情報を変更する場合には、セパレータ名称が初期表示されます。セパレータの新規追加する場合には、「Sepa-」という接頭語に通し番号が付いた名称が初期表示されます。

#### [ セパレータ値 ]

##### [ 文字列 ] ラジオボタン

セパレータの値として文字列を使用する場合に選択します。

##### [ バイト列 ] ラジオボタン

セパレータの値としてバイト列を使用する場合に選択します。

##### [ 値 ]

セパレータの値として使用する文字列またはバイト列を指定します。次の値を指定できます。文字列の場合、指定できる文字は、ラテン文字 (JIS X 0201 の「0x21」～「0x7E」) または半角スペース (JIS X 0201 の「0x20」) で 1,024 文字以内です。

セパレータの値として ASCII 文字列の「012」を表現したい場合の指定例を次に示します。

- 文字列の場合

「012」と、文字列をそのまま指定します。

## 11. 開発環境で使用する画面

- バイト列の場合

「303132」と、16進数の0x表記なしの形式で指定します。

[OK] ボタン

指定した内容を設定してセパレータ追加 / 変更ダイアログを閉じます。

[キャンセル] ボタン

指定した内容を設定しないでセパレータ追加 / 変更ダイアログを閉じます。

## 11.4 ビジネスプロセスに関連するダイアログ

ビジネスプロセスに関連するダイアログについて説明します。

### ！ 注意事項

ビジネスプロセスに関連するダイアログでは、入力項目の前後に半角スペースや全角スペースを指定できません。

### 11.4.1 変数・相関セット一覧ダイアログ

変数・相関セット一覧ダイアログは、変数や相関セットを追加、変更したり、一覧を表示したりするときに使用します。

操作手順については、「6.5.1(6) 変数の定義方法」、および「6.5.2(4)(a) 相関セットの新規定義」を参照してください。

図 11-41 変数・相関セット一覧ダイアログ（変数情報の表示）



図 11-42 変数・相関セット一覧ダイアログ（相関セット情報の表示）



[ 変数一覧 ]

プロセスまたはスコープに属する定義済み変数の一覧が表示されます。

[ 変数名 ]

変数名を 64 バイト以内で指定します。

[ 種別 ]

変数の種別をメッセージ型 ( message Type ), 数値 ( numeric ), 文字列 ( string ), 真偽値 ( boolean ) の 4 種類から選択します。

[ 電文フォーマット ]

変数の種別がメッセージ型の場合に、電文フォーマットとなる XML スキーマファイルが表示されます。

[ 取込 ] ボタン

変数の種別がメッセージ型の場合に、呼び出すサービス部品またはユーザ定義受付の要求電文、応答電文またはフォルト電文に設定されている電文フォーマットを、変数に設定する場合に使用します。

電文フォーマットの取込ダイアログが表示されます。電文フォーマットの取込ダイアログでは、変数に設定する電文フォーマットを指定します。

[ ... ] ボタン

変数の種別がメッセージ型の場合に、電文フォーマットとなる XML スキーマファイルを設定します。

## [ 表示 ] ボタン

変数の種別がメッセージ型の場合、変数のパスを取得するときに使用します。変数の表示ダイアログが表示されます。

## [ 部分指定 ] チェックボックス

変数の種別がメッセージ型の場合、部分指定をするためにチェックします。

## [ 部分名 ]

部分指定をする場合、部分名を指定します。

部分名は、次の条件に従って指定します。

部分の型がメッセージ型 ( message Type ) の場合

部分名はビジネスプロセス内で一意である必要があります。

部分の型が数値 ( numeric ) , 文字列 ( string ) または真偽値 ( boolean ) の場合

- 電文フォーマットが異なる変数で、同じ部分の型である場合、同じ部分名を使用できます。
- 上記以外の場合、部分名は一意である必要があります。

## [ 指定式 ]

部分指定をする場合、部分名に対応する式を指定します。指定式には、変数の表示ダイアログの [ 選択パス ] で取得したパスを指定します。

## [ 型 ]

変数の種別がメッセージ型の場合、部分名に対応する型を選択します。メッセージ型 ( message Type ) , 数値 ( numeric ) , 文字列 ( string ) , 真偽値 ( boolean ) の 4 種類から選択します。

## [ 関連セット一覧 ]

プロセスまたはスコープに属する定義済み関連セットの一覧が表示されます。

## [ 関連セット名 ]

関連セット名を 64 バイト以内で指定します。

## [ 取得部 ]

## [ 利用変数 ]

関連セットの定義に利用する電文フォーマットを持つ変数を指定します。所属しているスコープおよびその親スコープで定義済みの、基本型の部分を持つメッセージ型の変数が表示されます。

## [ 部分名 ]

[ 利用変数 ] で選択した変数に対して、関連セットとして利用する基本型の部分の部分名を指定します。

## [ 行追加 ] ボタン

部分指定または取得部指定をするための行を追加します。

## 11. 開発環境で使用する画面

### [ 行削除 ] ボタン

不要な部分指定または取得部指定の行を削除します。

### [ 追加 ] ボタン

定義した変数や相関セットの情報を新規追加します。

### [ 削除 ] ボタン

定義した変数や相関セットの情報を削除します。

### [ 更新 ] ボタン

定義済みの変数や相関セットの情報の内容を更新します。

### [ OK ] ボタン

定義した内容で、変数・相関セット一覧ダイアログを閉じます。

### [ キャンセル ] ボタン

定義した内容を取り消し、定義前の内容で変数・相関セット一覧ダイアログを閉じます。

## 11.4.2 割当相関セット群ダイアログ

割当相関セット群ダイアログは、変数・相関セット一覧ダイアログで定義した相関セットグループをアクティビティに割り当てるときに使用します。

受付アクティビティ、応答アクティビティ、およびサービス呼出アクティビティの [ 設定 ] ボタンをクリックした場合に表示されます。

操作手順については、「6.5.2(4)(b) アクティビティからの相関セットの設定」を参照してください。

図 11-43 割当相関セット群ダイアログ



## [ 関連セット ]

要求電文や応答電文に割り当てる関連セットを、ドロップダウンリストから選択します。ドロップダウンリストには、変数・関連セット一覧ダイアログで定義済みの関連セットだけが表示されます。

## [ 追加 ] ボタン

[ 関連セット ] で選択した関連セットを [ 関連セット名 ] に追加します。

## [ 編集 ] ボタン

変数・関連セット一覧ダイアログが表示されます。

## [ 削除 ] ボタン

[ 関連セット名 ] に表示された関連セットを行ごと削除します。一度に複数の関連セットの行を選択して削除できます。

## [ 関連セット名 ]

[ 関連セット ] で選択した関連セットが表示されます。

## [ 初期化 ]

関連セットを初期化するかどうかを選択します。初期化する場合は「yes」を、初期化しない場合は「no」を選択します。

## [ パターン ]

関連セットのパターンを選択します。受付アクティビティ、および応答アクティビティから関連セットの割当があった場合は、「-」が表示されます。

## [ OK ] ボタン

定義した内容で、割当関連セット群ダイアログを閉じます。

## [ キャンセル ] ボタン

定義した内容を取り消し、定義前の内容で割当関連セット群ダイアログを閉じます。

### 11.4.3 変数の表示ダイアログ

変数の表示ダイアログでは、変数のパスを取得する場合に使用します。

変数・関連セット一覧ダイアログの [ 表示 ] ボタンをクリックした場合に表示されます。

操作手順については、「6.5.1(4) 変数の表示とパスの取得」を参照してください。

図 11-44 変数の表示ダイアログ



[ 変数の内容 ]

変数・相関セット一覧ダイアログで選択した変数の内容がツリービューで表示されます。

表示された変数を選択し、右クリックして [パスの取得] を選択すると、ルート要素からの絶対パスを取得できます。取得したパスは、任意の個所に貼り付けることができます。

[ 選択パス ]

ツリービューで表示された変数を選択すると、選択した変数のパスがルート要素からの絶対パスで表示されます。

[ 閉じる ] ボタン

変数の表示ダイアログを閉じます。

## 11.4.4 電文フォーマットの取込ダイアログ

呼び出すサービス部品またはユーザ定義受付の、要求電文、応答電文またはフォルト電

文に設定されている電文フォーマットを、変数に設定するときに使用します。

変数・相関セット一覧ダイアログの [ 取込 ] ボタンをクリックした場合に表示されます。

変数の定義方法については、「6.5.1(6) 変数の定義方法」を参照してください。

図 11-45 電文フォーマットの取込ダイアログ

#### [ サービス / 受付 ]

##### [ サービス名 ] ラジオボタン

呼び出すサービス部品の要求電文、応答電文またはフォルト電文に設定されている電文フォーマットを、変数に設定する場合に選択します。ドロップダウンリストでは、呼び出すサービス部品を選択します。

##### [ 受付名 ] ラジオボタン

ユーザ定義受付に設定されている電文フォーマットを、変数に設定する場合に選択します。ドロップダウンリストでは、取り込むユーザ定義受付を選択します。

#### [ 取込対象 ]

##### [ オペレーション名 ]

[ サービス / 受付 ] で指定したサービス部品、またはユーザ定義受付のオペレーションのうち、取り込むオペレーション名をドロップダウンリストで選択します。

## 11. 開発環境で使用する画面

### [ 電文種別 ]

変数に設定する電文フォーマットの種別（要求電文，応答電文，フォルト電文）をドロップダウンリストで選択します。

### [ フォルト名 ]

[ 電文種別 ] でフォルト電文を選択した場合，[ オペレーション名 ] で選択したオペレーションのうち，取り込むフォルト名を選択します。

### [ 電文フォーマット ]

電文フォーマットとなる XML スキーマファイルの名称を入力します。

### [ OK ] ボタン

ダイアログを閉じます。設定した内容は，変数・相関セット一覧ダイアログに反映されます。

### [ キャンセル ] ボタン

設定した内容を保存しないでダイアログを閉じます。

## 11.4.5 フォルト処理の割当ダイアログ

フォルト処理の条件を設定するダイアログです。

フォルトコネクションを利用して設定した連結線をダブルクリックした場合には表示されます。

フォルト処理の設定方法については、「6.4.3 フォルト処理を定義する」を参照してください。

図 11-46 フォルト処理の割当ダイアログ



### [ 割当変数 ]

フォルトの対象となる変数をドロップダウンリストで選択します。ドロップダウン

リストには、該当するビジネスプロセスに定義されている変数が表示されます。  
なお、定義していないフォルトが発生した場合のフォルト処理を設定する場合、ド  
ロップダウンリストから「catch-all」を選択します。フォルトの発生するアクティ  
ビティがJava 呼出アクティビティの場合、定義できるのは「catch-all」のフォルト  
処理だけです。

[ 遷移先 ]

[ 割当変数 ] に指定したフォルトが発生したときに実行するフォルト処理を定義した  
アクティビティのアクティビティ名称を選択します。

[ クリア ] ボタン

選択している [ 割当変数 ] の行に指定されている内容をクリアします。

[ OK ] ボタン

定義内容を保存してダイアログを閉じます。

[ キャンセル ] ボタン

定義内容を保存しないでダイアログを閉じます。

## 11.4.6 受付アクティビティダイアログ

受付アクティビティの詳細を定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- キャンバスの受付アクティビティをダブルクリックする
- キャンバスの受付アクティビティを選択して右クリックし、[ 設定 ] を選択する

受付アクティビティの詳細の定義方法については、「6.6.2 受付アクティビティ」を参照  
してください。

図 11-47 受付アクティビティダイアログ

## [ アクティビティ名 ]

アクティビティの名称を指定します。使用できる文字は、XMLSchema の NCName 定義文字です。また、アクティビティの名称は、ビジネスプロセス（スコープ内も含む）内で一意になるように指定します。

## [ オペレーション名 ]

サービスリクエストからサービス部品を呼び出すときに使用するオペレーション名を 255 バイト以内で指定します。使用できる文字は、XMLSchema の NCName 定義文字です。

オペレーション名には、ビジネスプロセス内のほかの受付アクティビティに設定されていない名称を指定します。

## [ 割当変数 ]

ビジネスプロセスの要求電文に割り当てる変数をドロップダウンリストで選択します。ドロップダウンリストには、変数・関連セット一覧ダイアログで定義した変数のうち、利用可能な変数だけが表示されます。

## [ 編集 ] ボタン

[ 変数・関連セット一覧 ] ダイアログが表示されます。[ 割当変数 ] に設定する変数の内容を編集する場合にクリックします。

[ 変数・関連セット一覧 ] ダイアログの詳細については「11.4.1 変数・関連セット一覧ダイアログ」を参照してください。

## [ 割当関連セット群 ]

[ 設定 ] ボタンをクリックして定義済みの関連セットグループをアクティビティに割

り当てます。

[ 設定 ] ボタン

[ 割当関連セット群 ] ダイアログが表示されます。必要に応じて、アクティビティに割り当てる関連セットを [ 割当関連セット群 ] ダイアログで設定します。また、割り当てる関連セットを初期化するかどうかを設定します。

なお、[ インスタンス生成 ] で [ no ] を選択した場合、初期化しない関連セットを一つだけ設定する必要があります。

[ 割当関連セット群 ] ダイアログの詳細については、「11.4.2 割当関連セット群ダイアログ」を参照してください。

[ 通信モデル ]

オペレーションの同期または非同期を選択します。

[ インスタンス生成 ]

要求電文を受け付けたときに、プロセスを初期化するかどうかを選択します。

[ OK ] ボタン

定義内容を保存してダイアログを閉じます。

[ キャンセル ] ボタン

定義内容を保存しないでダイアログを閉じます。

## 11.4.7 応答アクティビティダイアログ

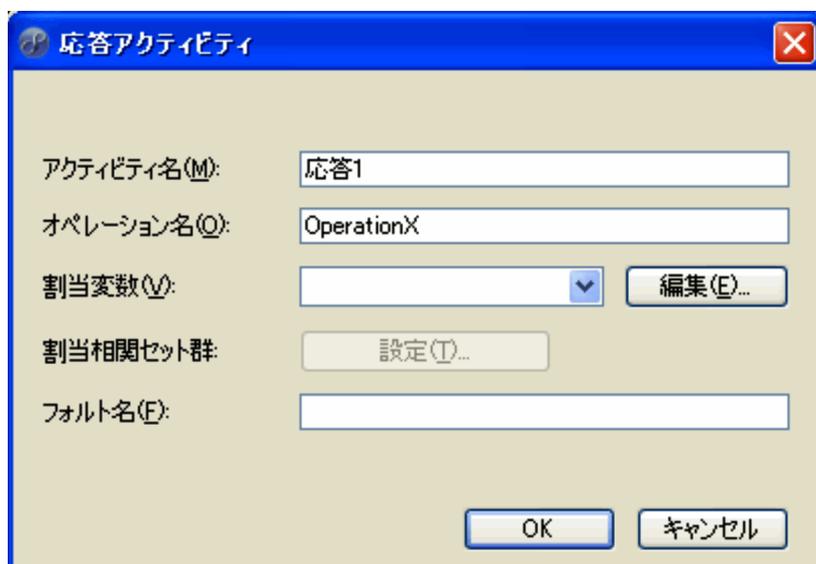
応答アクティビティの詳細を定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- キャンパスの応答アクティビティをダブルクリックする
- キャンパスの応答アクティビティを選択して右クリックし、[ 設定 ] を選択する

応答アクティビティの詳細の定義方法については、「6.6.3 応答アクティビティ」を参照してください。

図 11-48 応答アクティビティダイアログ



[ アクティビティ名 ]

アクティビティの名称を指定します。使用できる文字は、XMLSchema の NCName 定義文字です。また、アクティビティの名称は、ビジネスプロセス（スコープ内も含む）内で一意になるように指定します。

[ オペレーション名 ]

対応する受付アクティビティに指定したオペレーション名を、255 バイト以内で指定します。使用できる文字は、XMLSchema の NCName 定義文字です。

[ 割当変数 ]

応答電文に割り当てる変数をドロップダウンリストで選択します。ドロップダウンリストには、変数・関連セット一覧ダイアログで定義した変数のうち、利用可能な変数だけが表示されます。

[ 編集 ] ボタン

[ 変数・関連セット一覧 ] ダイアログが表示されます。[ 割当変数 ] に設定する変数の内容を編集する場合にクリックします。

[ 変数・関連セット一覧 ] ダイアログの詳細については「11.4.1 変数・関連セット一覧ダイアログ」を参照してください。

[ 割当関連セット群 ]

[ 設定 ] ボタンをクリックして定義済みの関連セットグループをアクティビティに割り当てます。

[ 設定 ] ボタン

[ 割当関連セット群 ] ダイアログが表示されます。必要に応じて、アクティビティに割り当てる関連セットを [ 割当関連セット群 ] ダイアログで設定します。また、割

り当てる相関セットを初期化するかどうかを設定します。

[ 割当相関セット群 ] ダイアログの詳細については、「11.4.2 割当相関セット群ダイアログ」を参照してください。

#### [ フォルト名 ]

フォルト処理として応答アクティビティを定義して、サービスリクエストにフォルトが発生したことを示す応答電文を送信する場合、フォルトの名称を指定します。

XMLSchema の NCName 定義文字でローカル名を指定します。

フォルト名を指定しない場合、応答電文は正常な応答となります。

#### [ OK ] ボタン

定義内容を保存してダイアログを閉じます。

#### [ キャンセル ] ボタン

定義内容を保存しないでダイアログを閉じます。

## 11.4.8 サービス呼出アクティビティダイアログ

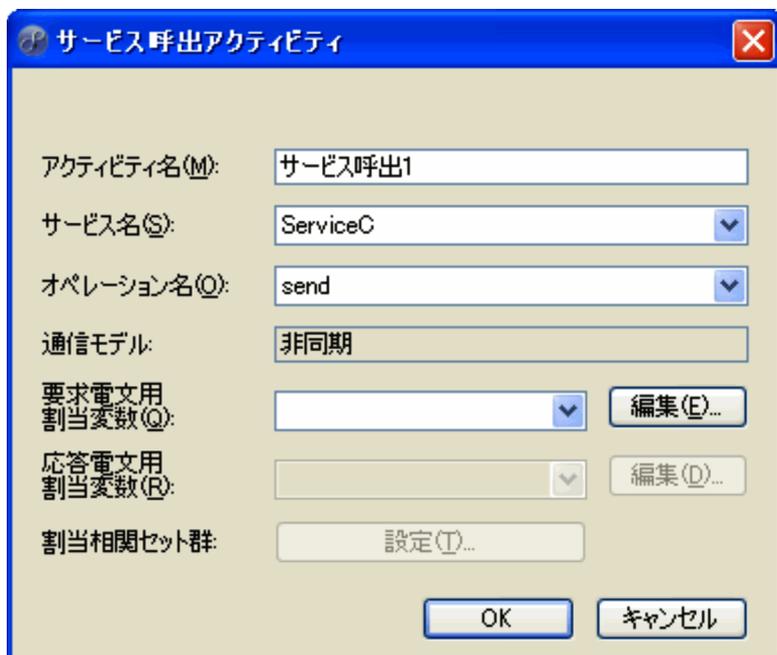
サービス呼出アクティビティの詳細を定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- キャンバスのサービス呼出アクティビティをダブルクリックする
- キャンバスのサービス呼出アクティビティを選択して右クリックし、[ 設定 ] を選択する

サービス呼出アクティビティの詳細の定義方法については、「6.6.4 サービス呼出アクティビティ」を参照してください。

図 11-49 サービス呼出アクティビティダイアログ



[ アクティビティ名 ]

アクティビティの名称を指定します。使用できる文字は、XMLSchema の NCName 定義文字です。また、アクティビティの名称は、ビジネスプロセス（スコープ内も含む）内で一意になるように指定します。

[ サービス名 ]

要求電文を送信して呼び出すサービス部品をドロップダウンリストで選択します。ドロップダウンリストには、ツリービューのサービス定義一覧に表示されるサービス部品が表示されます。

[ オペレーション名 ]

[ サービス名 ] で指定したサービス部品のオペレーションのうち、実際に呼び出すオペレーションの名称を指定します。

[ 通信モデル ]

[ オペレーション名 ] で選択したオペレーションに設定されている通信モデル（同期または非同期）が表示されます。

[ 要求電文用割当変数 ]

サービス部品を呼び出す要求電文に割り当てる変数をドロップダウンリストで選択します。ドロップダウンリストには、変数・相関セット一覧ダイアログで定義した変数のうち、利用可能な変数だけが表示されます。

[ 応答電文用割当変数 ]

同期オペレーションから受け取る応答電文に割り当てる変数をドロップダウンリス

トで選択します。ドロップダウンリストには、変数・相関セット一覧ダイアログで定義した変数のうち、利用可能な変数だけが表示されます。

#### [編集] ボタン

[変数・相関セット一覧] ダイアログが表示されます。[要求電文用割当変数] または [応答電文用割当変数] に設定する変数の内容を編集する場合にクリックします。[変数・相関セット一覧] ダイアログの詳細については「11.4.1 変数・相関セット一覧ダイアログ」を参照してください。

#### [割当相関セット群]

[設定] ボタンをクリックして定義済みの相関セットグループを割り当てます。相関セットグループは、サービス部品を呼び出す要求電文、およびサービス部品から受け取る応答電文に割り当てられます。

#### [設定] ボタン

[割当相関セット群] ダイアログが表示されます。サービス部品を呼び出す要求電文、およびサービス部品から受け取る応答電文に割り当てる相関セットを [割当相関セット群] ダイアログで設定します。また、割り当てる相関セットを初期化するかどうか、およびパターンを設定します。

[割当相関セット群] ダイアログの詳細については、「11.4.2 割当相関セット群ダイアログ」を参照してください。

#### [OK] ボタン

定義内容を保存してダイアログを閉じます。

#### [キャンセル] ボタン

定義内容を保存しないでダイアログを閉じます。

## 11.4.9 Java 呼出アクティビティダイアログ

Java 呼出アクティビティの詳細を定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- キャンパスの Java 呼出アクティビティをダブルクリックする
- キャンパスの Java 呼出アクティビティを選択して右クリックし、[設定] を選択する

Java 呼出アクティビティの詳細の定義方法については、「6.6.5 Java 呼出アクティビティ」を参照してください。

図 11-50 Java 呼出アクティビティダイアログ



## [ アクティビティ名 ]

アクティビティの名称を指定します。使用できる文字は、XMLSchema の NCName 定義文字です。また、アクティビティの名称は、ビジネスプロセス（スコープ内も含む）内で一意になるように指定します。

## [ Java クラス名 ]

Java クラス名を指定します。指定する Java クラス名にはパッケージ名称を含みません。なお、拡張子（「.class」、「.java」など）は指定しないでください。

## [ 引数用割当変数 ]

[ Java クラス名 ] で指定した Java クラスを呼び出すときの引数に割り当てる変数をドロップダウンリストで選択します。ドロップダウンリストには、変数・相関セット一覧ダイアログで定義した変数のうち、利用可能な変数だけが表示されます。

## [ 戻り値用割当変数 ]

[ Java クラス名 ] で指定した Java クラスを呼び出すときの戻り値に割り当てる変数をドロップダウンリストで選択します。ドロップダウンリストには、変数・相関セット一覧ダイアログで定義した変数のうち、利用可能な変数だけが表示されます。

## [ 編集 ] ボタン

[ 変数・相関セット一覧 ] ダイアログが表示されます。[ 引数用割当変数 ] または [ 戻り値用割当変数 ] に設定する変数の内容を編集する場合にクリックします。  
[ 変数・相関セット一覧 ] ダイアログの詳細については「11.4.1 変数・相関セット一覧ダイアログ」を参照してください。

## [ OK ] ボタン

定義内容を保存してダイアログを閉じます。

## [ キャンセル ] ボタン

定義内容を保存しないでダイアログを閉じます。

## 11.4.10 データ変換アクティビティダイアログ

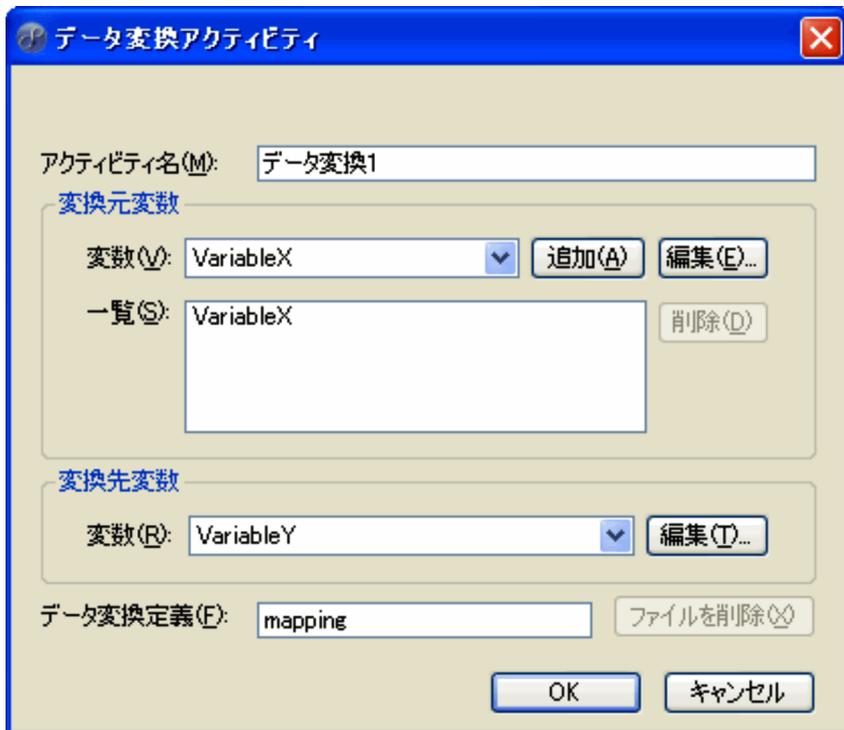
データ変換アクティビティの詳細を定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- キャンパスのデータ変換アクティビティをダブルクリックする
- キャンパスのデータ変換アクティビティを選択して右クリックし、[ 設定 ] を選択する

データ変換アクティビティの詳細の定義方法については、「6.6.6 データ変換アクティビティ」を参照してください。

図 11-51 データ変換アクティビティダイアログ



### [ アクティビティ名 ]

アクティビティの名称を指定します。使用できる文字は、XMLSchema の NCName 定義文字です。また、アクティビティの名称は、ビジネスプロセス（スコープ内も含む）内で一意になるように指定します。

### [ 変換元変数 ]

次の項目を利用して、データ変換の変換元となる変数を指定します。変数は複数指定できます。

### [ 変数 ]

データ変換の変換元になる変数をドロップダウンリストで選択します。ドロップ

## 11. 開発環境で使用する画面

プルダウンリストには、変数・相関セット一覧ダイアログで定義した変数のうち、利用可能な変数だけが表示されます。

選択した変数は、[追加] ボタンをクリックすると [一覧] に表示されます。

### [追加] ボタン

[変数] で選択した変数を [一覧] に追加します。

### [編集] ボタン

[変数・相関セット一覧] ダイアログが表示されます。[変数] に設定する変数の内容を編集する場合にクリックします。

[変数・相関セット一覧] ダイアログの詳細については「11.4.1 変数・相関セット一覧ダイアログ」を参照してください。

### [一覧]

データ変換の変換元になる変数が表示されます。

### [削除] ボタン

[一覧] で選択した変数を削除します。複数選択して [削除] ボタンをクリックすると、選択したすべての変数が削除されます。

### [変換先変数]

次の項目を利用して、データ変換の変換先となる変数を指定します。

#### [変数]

データ変換の変換先になる変数をドロップダウンリストで選択します。ドロップダウンリストには、変数・相関セット一覧ダイアログで定義した変数のうち、利用可能な変数だけが表示されます。

#### [編集] ボタン

[変数・相関セット一覧] ダイアログが表示されます。[変数] に設定する変数の内容を編集する場合にクリックします。

[変数・相関セット一覧] ダイアログの詳細については「11.4.1 変数・相関セット一覧ダイアログ」を参照してください。

### [データ変換定義]

データ変換定義ファイルの論理名を指定します。XML で無効となる文字は指定できません。また、データ変換定義ファイルの名称は、ビジネスプロセス内で一意になるように指定します。

論理名とは、データ変換定義ファイルに仮想的に付与する名称です。論理名には対応するファイル名が自動的に付与されます。

データ変換定義画面では、指定したデータ変換定義ファイルの名称でデータ変換定義を作成します。

### [ファイルを削除] ボタン

[データ変換定義] に指定したデータ変換定義ファイルを削除します。

### [OK] ボタン

定義内容を保存してダイアログを閉じます。

## [キャンセル] ボタン

定義内容を保存しないでダイアログを閉じます。

**!** 注意事項

[データ変換定義] に指定したデータ変換定義ファイルがリポジトリに格納されたあとは、[変換元変数] および [変換先変数] が非活性になります。[変換元変数] および [変換先変数] を変更したい場合、[ファイルを削除] ボタンをクリックしてください。

なお、この場合に [ファイルを削除] をクリックしても、データ変換定義ファイルはリポジトリから削除されません。データ変換定義ファイルは、ビジネスプロセスを保存したときにリポジトリから削除されます。

そのため、[ファイルを削除] をクリックしてからビジネスプロセスを保存する前に、[データ変換アクティビティ] ダイアログを再表示した場合、[変換元変数] および [変換先変数] は非活性のままになります。

## 11.4.11 代入アクティビティダイアログ

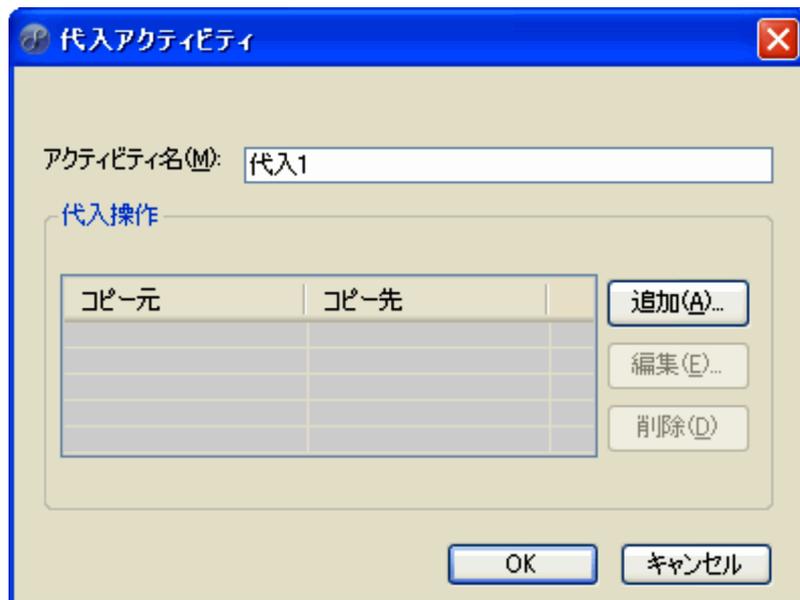
代入アクティビティの詳細を定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- キャンパスの代入アクティビティをダブルクリックする
- キャンパスの代入アクティビティを選択して右クリックし、[設定] を選択する

代入アクティビティの詳細の定義方法については、「6.6.7 代入アクティビティ」を参照してください。

図 11-52 代入アクティビティダイアログ



[ アクティビティ名 ]

アクティビティの名称を指定します。使用できる文字は、XMLSchema の NCName 定義文字です。また、アクティビティの名称は、ビジネスプロセス（スコープ内も含む）内で一意になるように指定します。

[ コピー元 ]

代入元となる変数または値が表示されます。[ 追加 ] ボタンをクリックして表示される [ 代入アクティビティ ] サブダイアログで設定します。

[ コピー先 ]

代入先となる変数が表示されます。[ 追加 ] ボタンをクリックして表示される [ 代入アクティビティ ] サブダイアログで設定します。

[ 追加 ] ボタン

[ 代入アクティビティ ] サブダイアログが表示されます。[ コピー先 ] および [ コピー元 ] に変数または値を追加する場合にクリックします。[ 代入アクティビティ ] サブダイアログについては、「11.4.12 代入アクティビティサブダイアログ」を参照してください。

[ 編集 ] ボタン

[ 代入アクティビティ ] サブダイアログが表示されます。[ コピー先 ] および [ コピー元 ] の変数または値を編集する場合にクリックします。[ 代入アクティビティ ] サブダイアログについては、「11.4.12 代入アクティビティサブダイアログ」を参照してください。

[ 削除 ] ボタン

[ コピー先 ] および [ コピー元 ] で選択した行を削除します。複数の行を選択して [ クリア ] ボタンをクリックすると、選択したすべての行が削除されます。

[ OK ] ボタン

定義内容を保存してダイアログを閉じます。

[ キャンセル ] ボタン

定義内容を保存しないでダイアログを閉じます。

## 11.4.12 代入アクティビティサブダイアログ

代入アクティビティダイアログで定義する [ コピー元 ] および [ コピー先 ] を追加または編集するダイアログです。

次のどちらかの操作をした場合に表示されます。

- 代入アクティビティダイアログの [ 追加 ] ボタンをクリックする
- 代入アクティビティダイアログの [ 編集 ] ボタンをクリックする

代入アクティビティの詳細の定義方法については、「6.6.7 代入アクティビティ」を参照してください。

図 11-53 代入アクティビティサブダイアログ



#### [ コピー元 ]

次の項目を利用して、代入アクティビティダイアログの [ コピー元 ] に設定する値を指定します。

##### [ 変数 ] ラジオボタン

変数名、または変数名と部分名で [ コピー元 ] に変数を設定する場合に選択します。

##### [ 名前 ]

[ コピー元 ] に設定する変数名をドロップダウンリストで選択します。ドロップダウンリストには、変数・相関セット一覧ダイアログで定義した変数のうち、利用可能な変数だけが表示されます。

##### [ 編集 ] ボタン

[ 変数・相関セット一覧 ] ダイアログが表示されます。[ 名前 ] に設定する変数の内容を編集する場合にクリックします。

[ 変数・相関セット一覧 ] ダイアログの詳細については「11.4.1 変数・相関セット一覧ダイアログ」を参照してください。

##### [ 部分名 ]

[ 名前 ] で選択した変数がメッセージ型の場合、部分名をドロップダウンリスト

で選択できます。ドロップダウンリストには、[名前] で選択した変数に設定されている部分名だけが表示されます。

[式] ラジオボタン

[コピー元] に値 (数値, 文字列, 真偽値) を設定する場合, または [コピー元] にメッセージ型の変数の一部を絶対パス (XPath 式) で指定して設定する場合に選択します。

[変数の表示]

[コピー元] の [式] で変数を利用する場合, 表示する変数名をドロップダウンリストで選択します。ドロップダウンリストには, 変数・相関セット一覧ダイアログで定義したメッセージ型の変数のうち, 利用可能な変数だけが表示されます。

[表示] ボタン

[変数の表示] ダイアログが表示されます。[変数の表示] ダイアログでは, [変数の表示] で指定したメッセージ型の変数の内容から, コピー元に設定する内容を選択して, 絶対パス (XPath 式) を取得できます。

[変数の表示] ダイアログについては, 「11.4.3 変数の表示ダイアログ」を参照してください。

[値]

[コピー元] に数値, 文字列, 真偽値を設定する場合, それぞれの値を指定します。

[変数の表示] でメッセージ型の変数を選択した場合とメッセージ型以外の変数を選択した場合とで, 指定する値が異なります。指定する値については, 「11.8.12 XPath を指定する」を参照してください。

[コピー先]

次の項目を利用して, 代入アクティビティダイアログの [コピー先] に設定する値を指定します。

[変数名]

[コピー先] に設定する変数名をドロップダウンリストで選択します。ドロップダウンリストには, 変数・相関セット一覧ダイアログで定義した変数のうち, 利用可能な変数だけが表示されます。

[編集] ボタン

[変数・相関セット一覧] ダイアログが表示されます。[変数名] に設定する変数の内容を編集する場合にクリックします。

[変数・相関セット一覧] ダイアログの詳細については「11.4.1 変数・相関セット一覧ダイアログ」を参照してください。

[部分名]

[変数名] で選択した変数がメッセージ型の場合, 部分名をドロップダウンリストで選択できます。ドロップダウンリストには, [変数名] で選択した変数に設定されている部分名だけが表示されます。

## [ OK ] ボタン

定義内容を [ 代入アクティビティ ] ダイアログに反映してダイアログを閉じます。

## [ キャンセル ] ボタン

定義内容を [ 代入アクティビティ ] ダイアログに反映しないでダイアログを閉じます。

### 11.4.13 無操作アクティビティダイアログ

無操作アクティビティの詳細を定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- キャンパスの無操作アクティビティをダブルクリックする
- キャンパスの無操作アクティビティを選択して右クリックし、[ 設定 ] を選択する

無操作アクティビティの詳細の定義方法については、「6.6.8 無操作アクティビティ」を参照してください。

図 11-54 無操作アクティビティダイアログ



## [ アクティビティ名 ]

アクティビティの名称を指定します。使用できる文字は、XMLSchema の NCName 定義文字です。また、アクティビティの名称は、ビジネスプロセス（スコープ内も含む）内で一意になるように指定します。

## [ OK ] ボタン

定義内容を保存してダイアログを閉じます。

## [ キャンセル ] ボタン

定義内容を保存しないでダイアログを閉じます。

### 11.4.14 フォルト送出アクティビティダイアログ

フォルト送出アクティビティの詳細を定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- キャンパスのフォルト送出アクティビティをダブルクリックする

## 11. 開発環境で使用する画面

- キャンバスのフォルト送出アクティビティを選択して右クリックし、[設定]を選択する

フォルト送出アクティビティの詳細の定義方法については、「6.6.9 フォルト送出アクティビティ」を参照してください。

図 11-55 フォルト送出アクティビティダイアログ



### [アクティビティ名]

アクティビティの名称を指定します。使用できる文字は、XMLSchema の NCName 定義文字です。また、アクティビティの名称は、ビジネスプロセス（スコープ内も含む）内で一意になるように指定します。

### [割当変数]

上位のスコープまたはビジネスプロセスに送出するフォルトに割り当てる変数を、ドロップダウンリストで選択します。ドロップダウンリストには、変数・相関セット一覧ダイアログで定義した変数のうち、利用可能な変数だけが表示されます。

### [編集] ボタン

[変数・相関セット一覧] ダイアログが表示されます。[割当変数] に設定する変数の内容を編集する場合にクリックします。

[変数・相関セット一覧] ダイアログの詳細については「11.4.1 変数・相関セット一覧ダイアログ」を参照してください。

### [OK] ボタン

定義内容を保存してダイアログを閉じます。

### [キャンセル] ボタン

定義内容を保存しないでダイアログを閉じます。

## 11.4.15 スコープアクティビティダイアログ

スコープアクティビティの詳細を定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- キャンパスのスコープアクティビティをダブルクリックする
- キャンパスのスコープアクティビティを選択して右クリックし,[設定]を選択する

スコープアクティビティの詳細の定義方法については、「6.6.10 スコープアクティビティ」を参照してください。

図 11-56 スコープアクティビティダイアログ



[ アクティビティ名 ]

アクティビティの名称を指定します。使用できる文字は、XMLSchema の NCName 定義文字です。また、アクティビティの名称は、ビジネスプロセス（スコープ内も含む）内で一意になるように指定します。

[ OK ] ボタン

定義内容を保存してダイアログを閉じます。

[ キャンセル ] ボタン

定義内容を保存しないでダイアログを閉じます。

## 11.4.16 繰り返しアクティビティダイアログ

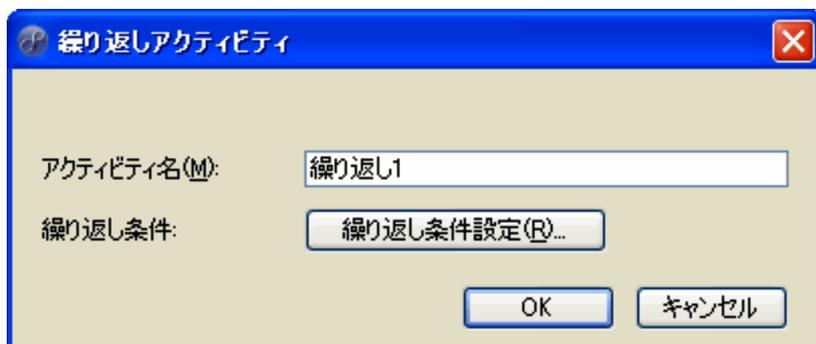
繰り返しアクティビティの詳細を定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- キャンパスの繰り返しアクティビティをダブルクリックする
- キャンパスの繰り返しアクティビティを選択して右クリックし,[設定]を選択する

繰り返しアクティビティの詳細の定義方法については、「6.6.11 繰り返しアクティビティ」を参照してください。

図 11-57 繰り返しアクティビティダイアログ



[ アクティビティ名 ]

アクティビティの名称を指定します。使用できる文字は、XMLSchema の NCName 定義文字です。また、アクティビティの名称は、ビジネスプロセス（スコープ内も含む）内で一意になるように指定します。

[ 繰り返し条件設定 ] ボタン

[ 条件設定 ] ダイアログが表示されます。繰り返しの条件を設定するときにクリックします。[ 条件設定 ] ダイアログについては、「11.4.17 条件設定ダイアログ（繰り返しアクティビティ）」を参照してください。

[ OK ] ボタン

定義内容を保存してダイアログを閉じます。

[ キャンセル ] ボタン

定義内容を保存しないでダイアログを閉じます。

## 11.4.17 条件設定ダイアログ（繰り返しアクティビティ）

繰り返しアクティビティを利用して繰り返し処理を設定する場合の条件を定義するダイアログです。

繰り返しアクティビティダイアログの [ 繰り返し条件設定 ] ボタンをクリックした場合に表示されます。

繰り返しアクティビティの詳細の定義方法については、「6.6.11 繰り返しアクティビティ」を参照してください。

図 11-58 [条件設定] ダイアログ (繰り返しアクティビティ)



## [ 変数 ]

条件に変数を利用した式を設定する場合、変数をドロップダウンリストで選択します。ドロップダウンリストには、変数・相関セット一覧ダイアログで定義したメッセージ型の変数のうち、利用可能な変数だけが表示されます。

## [ 表示 ] ボタン

[ 変数の表示 ] ダイアログが表示されます。[ 変数の表示 ] ダイアログでは、[ 変数 ] で指定したメッセージ型の変数の内容を確認できます。

[ 変数の表示 ] ダイアログについては、「11.4.3 変数の表示ダイアログ」を参照してください。

## [ 条件式 ]

繰り返し条件の条件式を XPath 式で指定します。XPath 式の中では、拡張関数を使用して変数内の情報を取得し、利用することができます。拡張関数の使用方法については、「11.8.12 XPath を指定する」を参照してください。

## [ 最大繰り返し回数 ]

繰り返しの回数の最大値を半角の数字で指定します。指定できる値は 1 ~ 126789 です。

## [ OK ] ボタン

定義内容を [ 繰り返しアクティビティ ] ダイアログに反映してダイアログを閉じます。

[キャンセル] ボタン

定義内容を [ 繰り返しアクティビティ ] ダイアログに反映しないでダイアログを閉じます。

## 11.4.18 分岐アクティビティダイアログ

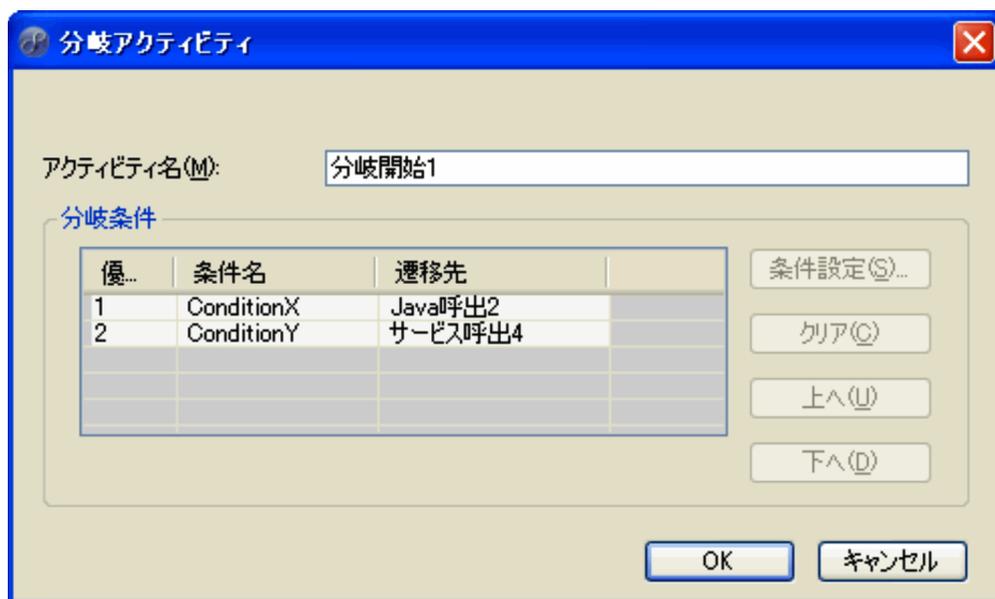
分岐アクティビティの詳細を定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- キャンパスの分岐アクティビティをダブルクリックする
- キャンパスの分岐アクティビティを選択して右クリックし, [ 設定 ] を選択する

分岐アクティビティの詳細の定義方法については, 「6.6.12 分岐アクティビティ」を参照してください。

図 11-59 分岐アクティビティダイアログ



[ アクティビティ名 ]

アクティビティの名称を指定します。使用できる文字は, XMLSchema の NCName 定義文字です。また, アクティビティの名称は, ビジネスプロセス ( スコープ内も含む ) 内で一意になるように指定します。

[ 優先順位 ]

[ 条件名 ] に設定される条件の優先順位が, 優先順位の高い順に表示されます。条件の優先順位は, [ 上へ ] ボタンおよび [ 下へ ] ボタンで変更できます。

[ 条件名 ]

条件の名称が表示されます。条件は, [ 条件設定 ] ボタンをクリックして表示される

[条件設定] ダイアログで設定します。

[遷移先]

その条件を満たす場合に実行するアクティビティのアクティビティ名を選択します。

[条件設定] ボタン

[条件設定] ダイアログが表示されます。分岐の条件を設定するときにクリックします。[条件設定] ダイアログについては、「11.4.19 条件設定ダイアログ (分岐アクティビティ)」を参照してください。

[クリア] ボタン

選択した条件を削除します。

[上へ] ボタン

選択した条件の優先順位を一つ上げます。

[下へ] ボタン

選択した条件の優先順位を一つ下げます。

[OK] ボタン

定義内容を保存してダイアログを閉じます。

[キャンセル] ボタン

定義内容を保存しないでダイアログを閉じます。

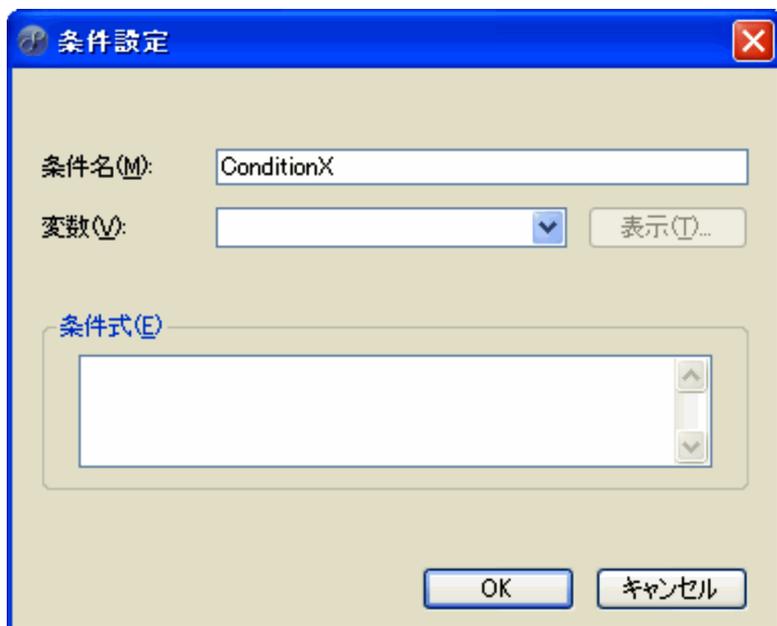
## 11.4.19 条件設定ダイアログ (分岐アクティビティ)

分岐アクティビティダイアログで定義する分岐処理の分岐条件を設定するダイアログです。

分岐アクティビティダイアログの [条件設定] ボタンをクリックした場合に表示されます。

分岐アクティビティの詳細の定義方法については、「6.6.12 分岐アクティビティ」を参照してください。

図 11-60 条件設定ダイアログ (分岐アクティビティ)



[ 条件名 ]

[ 分岐アクティビティ ] ダイアログの [ 条件名 ] に設定する条件の名称を指定します。条件名は、分岐アクティビティの定義内で一意になるように指定します。

[ 変数 ]

条件に変数を利用した式を設定する場合、変数をドロップダウンリストで選択します。ドロップダウンリストには、変数・相関セッター一覧ダイアログで定義したメッセージ型の変数のうち、利用可能な変数だけが表示されます。

[ 表示 ] ボタン

[ 変数の表示 ] ダイアログが表示されます。[ 変数の表示 ] ダイアログでは、[ 変数 ] で指定したメッセージ型の変数の内容を確認できます。

[ 変数の表示 ] ダイアログについては、「11.4.3 変数の表示ダイアログ」を参照してください。

[ 条件式 ]

分岐条件の条件式を XPath 式で指定します。XPath 式の中では、拡張関数を使用して変数内の情報を取得し、利用することができます。拡張関数の使用方法については、「11.8.12 XPath を指定する」を参照してください。

[ OK ] ボタン

定義内容を [ 分岐アクティビティ ] ダイアログに反映してダイアログを閉じます。

[ キャンセル ] ボタン

定義内容を [ 分岐アクティビティ ] ダイアログに反映しないでダイアログを閉じま

す。

## 11.4.20 並列処理アクティビティダイアログ

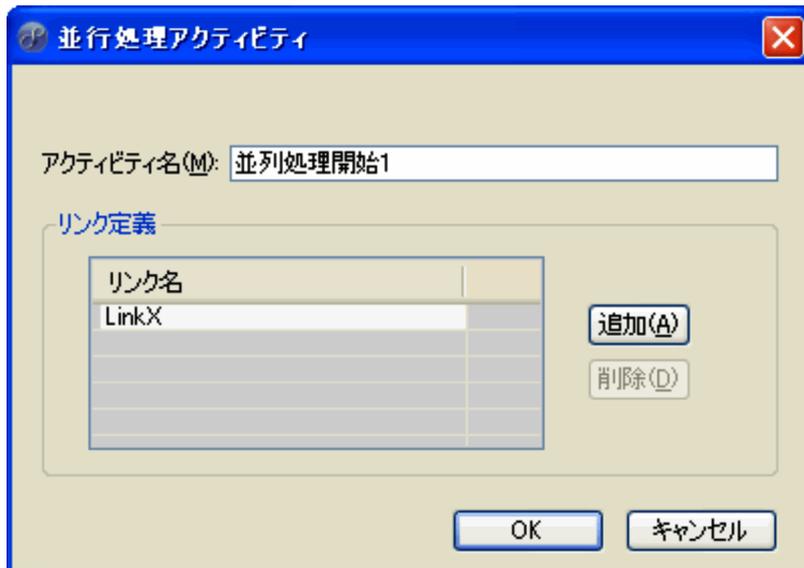
並列処理アクティビティの詳細を定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- キャンパスの並列処理開始アクティビティをダブルクリックする
- キャンパスの並列処理開始アクティビティを選択して右クリックし、[ 設定 ] を選択する

並列処理アクティビティの詳細の定義方法については、「6.6.13 並列処理アクティビティ」を参照してください。

図 11-61 並列処理アクティビティダイアログ



### [ アクティビティ名 ]

アクティビティの名称を指定します。使用できる文字は、XMLSchema の NCName 定義文字です。また、アクティビティの名称は、ビジネスプロセス (スコープ内も含む) 内で一意になるように指定します。

### [ リンク名 ]

リンクの名称を指定します。リンクの名称は、ビジネスプロセス内で一意になるように指定します。

### [ 追加 ] ボタン

[ リンク名 ] に入力行を一行追加します。

## 11. 開発環境で使用する画面

### [削除] ボタン

選択したリンク名を削除します。削除するリンク名は複数指定できます。

### [OK] ボタン

定義内容を保存してダイアログを閉じます。

### [キャンセル] ボタン

定義内容を保存しないでダイアログを閉じます。

## 11.4.21 リンクダイアログ

並列処理する複数のシーケンス間でのリンクを定義するダイアログです。

次のどちらかの操作をした場合に表示されます。

- リンク対象のアクティビティ同士を連結する連結線をダブルクリックする
- リンク対象のアクティビティ同士を連結する連結線を選択して右クリックし、[設定]を選択する

並列処理時のリンクの定義方法については、「6.6.13(2) シーケンス間リンクの定義」を参照してください。

図 11-62 リンクダイアログ



### [割当リンク名]

割り当てるリンクの名称をドロップダウンリストで選択します。ドロップダウンリストには、該当するスコープ内で設定したリンク名が表示されます。

### [リンク元]

リンクコネクションを利用して連結されているアクティビティのうち、連結元に当たるアクティビティの名称が表示されます。

## [ リンク先 ]

リンクコネクションを利用して連結されているアクティビティのうち、連結先に当たるアクティビティの名称が表示されます。

## [ 遷移条件 ]

リンクによる遷移に条件を定義するかどうかを選択します。

## [ 遷移条件設定 ] ボタン

[ 条件設定 ] ダイアログが表示されます。リンクによる遷移の条件を設定するときにクリックします。条件が設定できるのは、[ 遷移条件 ] で [ あり ] を選択した場合だけです。[ 条件設定 ] ダイアログについては、「11.4.22 条件設定ダイアログ ( 並列処理アクティビティ )」を参照してください。

## [ OK ] ボタン

定義内容を保存してダイアログを閉じます。

## [ キャンセル ] ボタン

定義内容を保存しないでダイアログを閉じます。

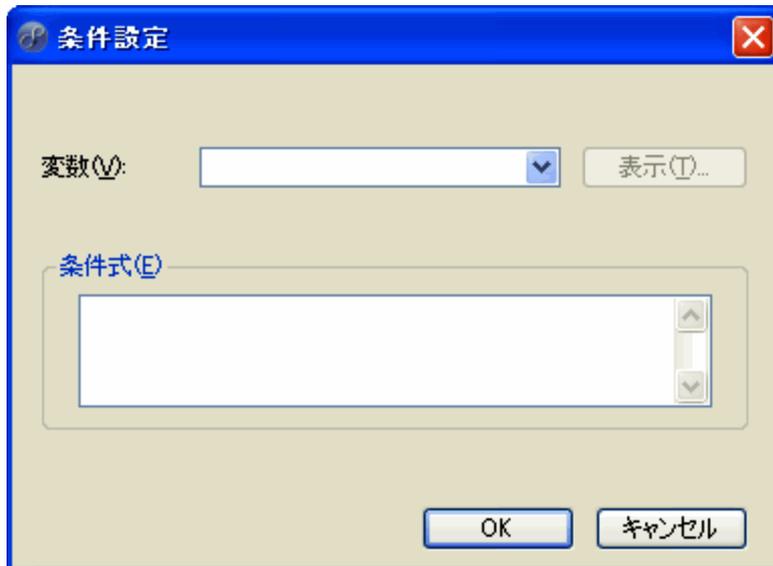
## 11.4.22 条件設定ダイアログ ( 並列処理アクティビティ )

並列処理でリンクを設定した場合のリンクの条件を定義するダイアログです。

リンクダイアログの [ 遷移条件設定 ] ボタンをクリックした場合に表示されます。

並列処理時のリンクの定義方法については、「6.6.13(2) シーケンス間リンクの定義」を参照してください。

図 11-63 条件設定ダイアログ ( 並列処理アクティビティ )



## 11. 開発環境で使用する画面

### [ 変数 ]

条件に変数を利用した式を設定する場合、変数をドロップダウンリストで選択します。ドロップダウンリストには、変数・相関セット一覧ダイアログで定義したメッセージ型の変数のうち、利用可能な変数だけが表示されます。

### [ 表示 ] ボタン

[ 変数の表示 ] ダイアログが表示されます。[ 変数の表示 ] ダイアログでは、[ 変数 ] で指定したメッセージ型の変数の内容を確認できます。

[ 変数の表示 ] ダイアログについては、「11.4.3 変数の表示ダイアログ」を参照してください。

### [ 条件式 ]

遷移条件の条件式を XPath 式で指定します。XPath 式の中では、拡張関数を使用して変数内の情報を取得し、利用することができます。拡張関数の使用方法については、「11.8.12 XPath を指定する」を参照してください。

### [ OK ] ボタン

定義内容を [ リンク ] ダイアログに反映してダイアログを閉じます。

### [ キャンセル ] ボタン

定義内容を [ リンク ] ダイアログに反映しないでダイアログを閉じます。

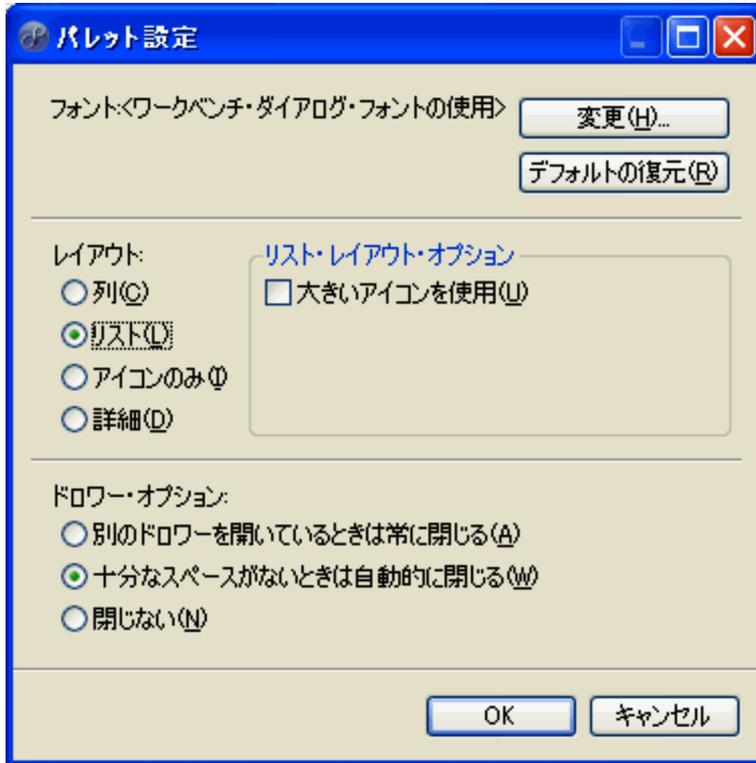
## 11.4.23 パレット設定ダイアログ

パレット設定ダイアログは、パレットの表示を設定する場合に使用します。

パレットのグループまたはツールを右クリックして、[ 設定 ] を選択した場合に表示されます。

操作手順については、「11.8.11 パレットの表示を設定する」を参照してください。

図 11-64 パレット設定ダイアログ



## [ フォント ]

## [ 変更 ] ボタン

パレットのグループおよびツールのフォントを変更する場合に使用します。クリックすると、[ 設定 ] ダイアログが表示されます。

## [ デフォルトの復元 ] ボタン

[ 変更 ] ボタンで変更したフォントを Eclipse で設定したフォントに戻す場合に使用します。

## [ レイアウト ]

次のどれかのラジオボタンを選択して、パレットのグループおよびツールのレイアウトを設定します。

## [ 列 ] ラジオボタン

すべてのグループおよびツールを水平に整列します。

## [ リスト ] ラジオボタン

すべてのグループおよびツールを垂直に整列します。

## [ アイコンのみ ] ラジオボタン

すべてのツールに対して、ツール名を表示しないでアイコンだけの表示にします。

## 11. 開発環境で使用する画面

### [ 詳細 ] ラジオボタン

すべてのツールに対して、詳細なフォーマットおよび記述を表示します。

### [ リスト・レイアウト・オプション ]

#### [ 大きいアイコンを使用 ] チェックボックス

パレットのツールのアイコンサイズを大きくする場合にチェックします。

### [ ドロワー・オプション ]

次のどれかのラジオボタンを選択して、パレットのツール一覧の表示方法を設定します。

#### [ 別のドロワーを開いているときは常に閉じる ] ラジオボタン

一つのグループのツール一覧だけを表示します。

#### [ 十分なスペースがないときは自動的に閉じる ] ラジオボタン

複数のグループのツール一覧を表示しているためパレットに十分な表示スペースがない場合、自動的にツール一覧が閉じます。

#### [ 閉じない ] ラジオボタン

常にツール一覧を表示します。

### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

### [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## 11.5 データ変換（マッピング）に関連するダイアログ

データ変換（マッピング）に関連するダイアログについて説明します。

### 11.5.1 ルート要素選択ダイアログ

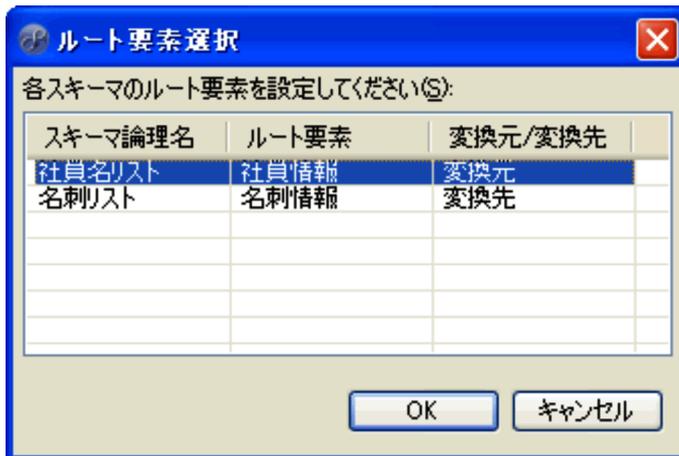
ルート要素選択ダイアログでは、マッピングするスキーマ論理名のルート要素を選択します。

次のどちらかの操作をした場合に表示されます。

- アダプタ定義画面またはビジネスプロセス定義画面のデータ変換アクティビティから、データ変換定義画面を表示する（新規にデータ変換（マッピング）を定義する場合）
- 変換元マッピングビューアまたは変換先マッピングビューアのスキーマ論理名を右クリックし、[ルート要素選択]を選択する

ルート要素選択ダイアログを次に示します。

図 11-65 ルート要素選択ダイアログ



#### (1) 表示項目

##### [ スキーマ論理名 ]

マッピング対象となる XML スキーマファイルのスキーマ論理名が表示されます。

##### [ ルート要素 ]

ルート要素をドロップダウンリストから選択します。選択したルート要素のスキーマがデータ変換（マッピング）定義画面でツリー表示されます。

## 11. 開発環境で使用する画面

### [ 変換元 / 変換先 ]

マッピング対象となる XML スキーマファイルが、変換元 XML スキーマファイルの場合は「変換元」、変換先 XML スキーマファイルの場合は「変換先」と表示されません。

### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

### [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

ルート要素選択ダイアログでは、ルート要素を選択および変更することができます。選択および変更すると、ルート要素のスキーマがデータ変換（マッピング）定義画面でツリー表示されます。ルート要素の選択および変更方法については、「7.3.1 データ変換（マッピング）の定義手順」を参照してください。

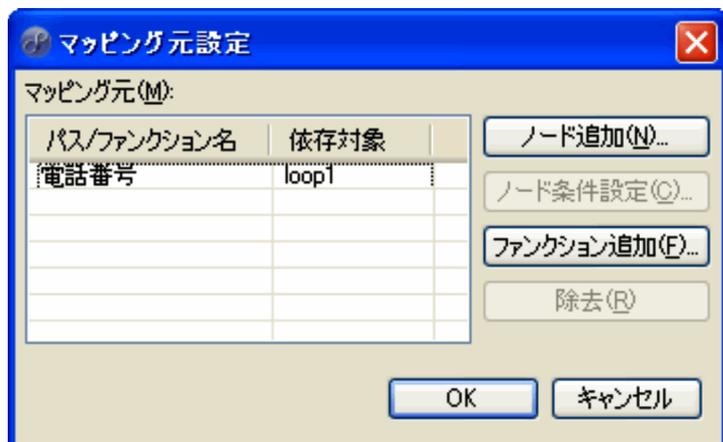
## 11.5.2 マッピング元設定ダイアログ

マッピング元設定ダイアログでは、マッピング元となる変換元ノードおよびファンクションを指定します。

マッピング元設定ダイアログは、変換先ノードを右クリックして、[ マッピング元 ] を選択した場合に表示されます。

マッピング元設定ダイアログを次に示します。

図 11-66 マッピング元設定ダイアログ



## (1) 表示項目

### [パス/ファンクション名]

マッピング元となる変換先ノードおよびファンクションを指定します。すでにマッピングされている場合、そのマッピング元が表示されます。

[パス/ファンクション名]での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

### [依存対象]

繰り返しファンクションを使用した場合、[パス/ファンクション名]で指定したくり返しファンクションに対応するくり返し依存対象が表示されます。くり返し依存対象の詳細については、「7.6 くり返しの設定」を参照してください。

### [ノード追加]ボタン

[ノード選択]ダイアログが表示されます。[パス/ファンクション名]にマッピング元となる変換元ノードを追加する場合にクリックします。

[ノード選択]ダイアログについては、「11.5.4 ノード選択ダイアログ(データ変換)」を参照してください。

### [ノード条件設定]ボタン

[ノード条件設定]ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ノード条件設定]ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

### [ファンクション追加]ボタン

[ファンクション選択]ダイアログが表示されます。[パス/ファンクション名]にマッピング元となるファンクションを追加する場合にクリックします。

[ファンクション選択]ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

### [除去]ボタン

[パス/ファンクション名]で選択した変換元ノードまたはファンクションが[パス/ファンクション名]から削除されます。

### [OK]ボタン

設定した内容を有効にして、ダイアログを閉じます。

### [キャンセル]ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

マッピング元設定ダイアログでは、次のことができます。

- マッピングする

マッピング元設定ダイアログを使用したマッピング方法については、「7.4 マッピングする」を参照してください。

- くり返し依存対象を確認する  
くり返し依存対象の確認方法については、「7.6.4 くり返し依存対象を確認する」を参照してください。
- くり返し依存対象を変更する  
くり返し依存対象の変更方法については、「7.6.5 くり返し依存対象を変更する」を参照してください。

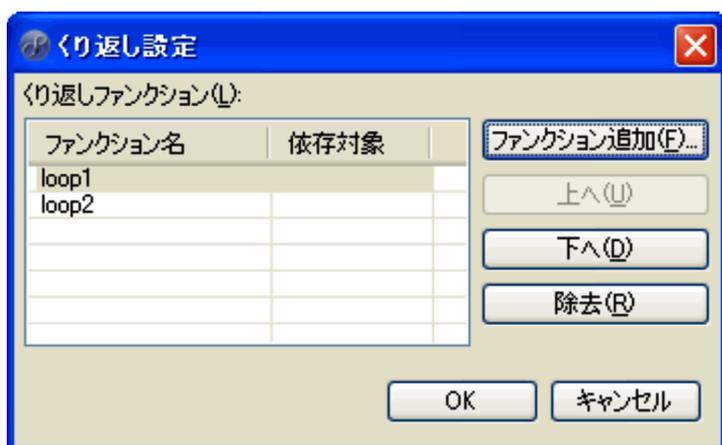
### 11.5.3 くり返し設定ダイアログ

くり返し設定ダイアログでは、マッピング元となるくり返しファンクションを指定します。くり返し設定ダイアログを次に示します。

くり返し設定ダイアログは、くり返しファンクションを使用した場合、変換先ノードを右クリックして、「くり返し」を選択したときに表示されます。

くり返し設定ダイアログを次に示します。

図 11-67 くり返し設定ダイアログ



#### (1) 表示項目

##### [ ファンクション名 ]

マッピング元となるくり返しファンクションを指定します。くり返しファンクション名が表示されます。

##### [ 依存対象 ]

[ ファンクション名 ] で指定したくり返しファンクションに対応するくり返し依存対象が表示されます。くり返し依存対象の詳細については、「7.6 くり返しの設定」を参照してください。

## [ ファンクション追加 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。[ ファンクション名 ] にマッピング元となるくり返しファンクションを追加する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

## [ 上へ ] ボタン

[ ファンクション名 ] で選択したくり返しファンクションの位置が一つ上に移動します。このボタンは、くり返しを合成する場合に使用することがあります。

## [ 下へ ] ボタン

[ ファンクション名 ] で選択したくり返しファンクションの位置が一つ下に移動します。このボタンは、くり返しを合成する場合に使用することがあります。

## [ 除去 ] ボタン

[ ファンクション名 ] で選択したくり返しファンクションが [ ファンクション名 ] から削除されます。

## [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

## [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

くり返し設定ダイアログでは、次のことができます。

- マッピングする
- くり返しを合成する
- くり返し依存対象を確認する
- くり返し依存対象を変更する

上記の方法については、「7.6 くり返しの設定」を参照してください。

## 11.5.4 ノード選択ダイアログ ( データ変換 )

ノード選択ダイアログでは、マッピング元となる変換元ノードを指定します。

次のどちらかの操作をした場合に表示されます。

- [ マッピング元設定 ] ダイアログの [ ノード追加 ] ボタンをクリックする
- 各ファンクションの設定ダイアログの [ ノード追加 ] ボタンまたは [ ノード選択 ] ボタンをクリックする

ノード選択ダイアログを次に示します。

図 11-68 ノード選択ダイアログ (データ変換)



### (1) 表示項目

#### [ 変換元ノード ]

マッピング元の変換元ノードが表示されます。

マッピング元として選択できないノードは、グレーで表示されます。グレーで表示されたノードを選択した場合、[ OK ] ボタンはクリックできません。

#### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

#### [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

### (2) このダイアログでできる内容

ノード選択ダイアログでは、マッピング元となる変換元ノードを指定して、[ マッピング元設定 ] ダイアログまたは各ファクションの設定ダイアログに指定した変換元ノードを設定します。

## 11.5.5 ファンクション選択ダイアログ

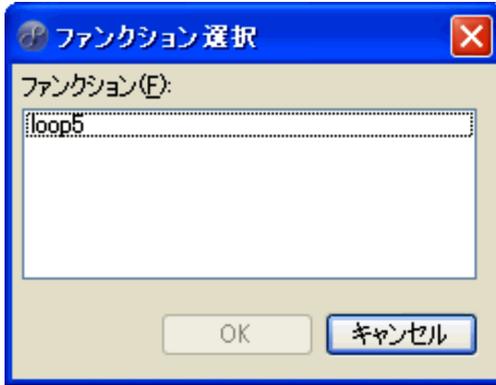
ファンクション選択ダイアログは、マッピング元となるファンクションを指定します。

次のどちらかの操作をした場合に表示されます。

- [ マッピング元設定 ] ダイアログの [ ファンクション追加 ] ボタンをクリックする
- 各ファンクションの設定ダイアログの [ ファンクション追加 ] ボタンまたは [ ファンクション選択 ] ボタンをクリックする

ファンクション選択ダイアログを次に示します。

図 11-69 ファンクション選択ダイアログ



### (1) 表示項目

#### [ ファンクション ]

マッピング元として選択できるファンクションが表示されます。

#### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

#### [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

### (2) このダイアログでできる内容

ファンクション選択ダイアログでは、マッピング元となるファンクションを指定して、[ マッピング元設定 ] ダイアログまたは各ファンクションの設定ダイアログに指定したファンクションを設定します。

## 11.5.6 ノード条件設定ダイアログ

ノード条件設定ダイアログでは、ノード条件を設定する変換元ノードを指定し、条件を設定します。

次のどちらかの操作をした場合に表示されます。

- [ マッピング元設定 ] ダイアログの [ ノード条件設定 ] ボタンをクリックする
- 各ファンクションの設定ダイアログの [ ノード条件設定 ] ボタンをクリックする

ノード条件設定ダイアログを次に示します。

図 11-70 ノード条件設定ダイアログ



### (1) 表示項目

#### [ ノード ]

[ マッピング元設定 ] ダイアログまたは各ファンクションを設定するダイアログで選択したノードおよびその子孫ノードが表示されます。

#### [ 条件 ]

[ 条件設定 ] ダイアログで設定した条件が表示されます。

#### [ 条件設定 ] ボタン

[ 条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定する場合にクリックします。

[ 条件設定 ] ダイアログについては、「11.5.7 条件設定ダイアログ (ノード条件の設定)」を参照してください。

#### [ 条件削除 ] ボタン

[ 条件 ] で選択した条件を [ 条件 ] から削除します。

#### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

#### [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

### (2) このダイアログでできる内容

ノード条件設定ダイアログでは、ノード条件を設定する変換元ノードを指定して、ノード条件を設定します。条件を設定した変換元ノードは、その条件を満たす場合だけマッピングします。ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

## 11.5.7 条件設定ダイアログ（ノード条件の設定）

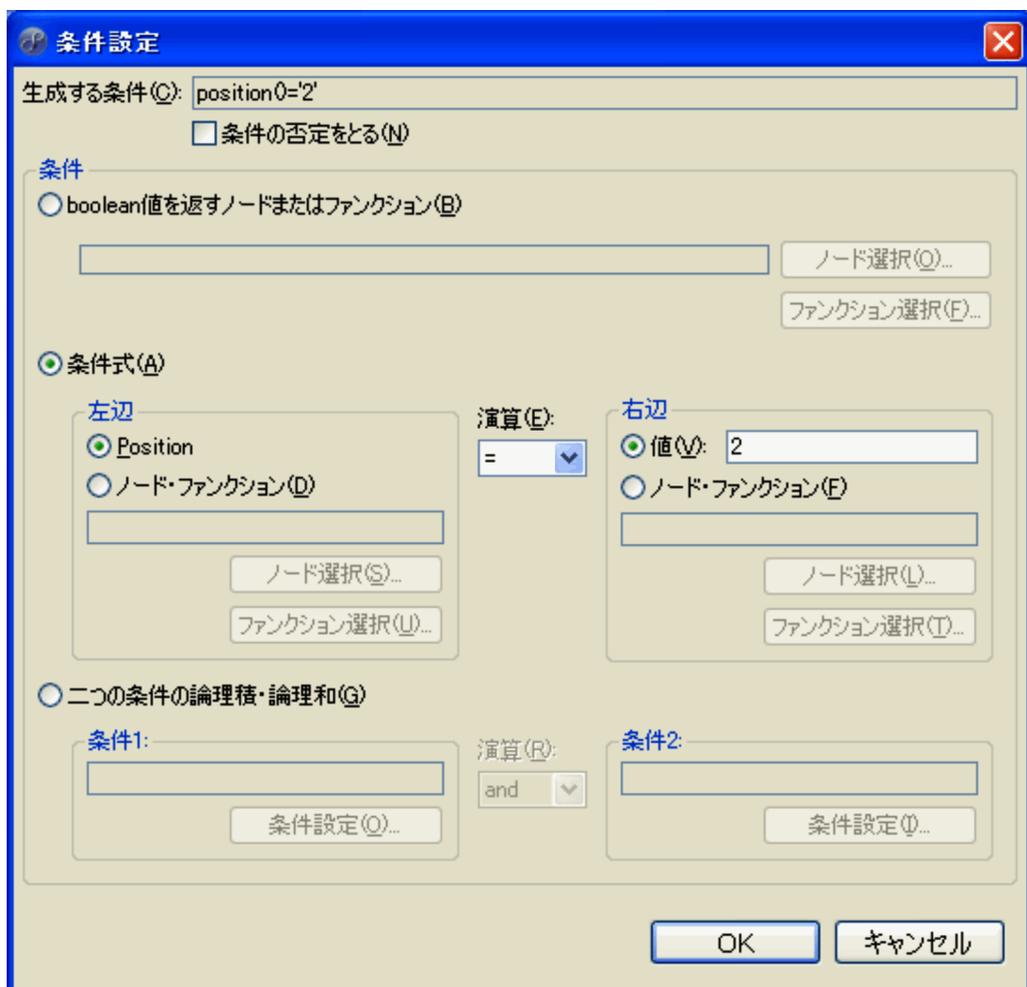
条件設定ダイアログでは、条件の内容を決めます。

次のどれかの操作をした場合に表示されます。

- [ ノード条件設定 ] ダイアログの [ 条件設定 ] ボタンをクリックする
- [ 条件追加 / 編集 ] ダイアログの [ 条件設定 ] ボタンをクリックする
- 条件設定ダイアログの [ 条件設定 ] ボタンをクリックする

条件設定ダイアログを次に示します。

図 11-71 条件設定ダイアログ（ノード条件の設定）



## (1) 表示項目

### [生成する条件]

[条件] で設定した条件が表示されます。

[条件の否定をとる] チェックボックスにチェックした場合、条件の先頭に「not」が表示されます。

### [条件の否定をとる] チェックボックス

[条件] で設定した条件と反対の内容を条件とする場合にチェックします。

### [条件]

### [boolean 値を返すノードまたはファンクション] ラジオボタン

論理値を条件として設定する場合に選択します。

[ノード選択] ボタンまたは [ファンクション選択] ボタンで指定したマッピング元の値によって、次の論理値が設定されます。

- マッピング元の値が「true」、または「0」、「-0」、もしくは「NaN」以外の数値の場合  
真が条件として設定されます。
- マッピング元の値が「true」以外の文字列、または「0」、「-0」、もしくは「NaN」の数値の場合  
偽が条件として設定されます。

### [ノード選択] ボタン

[ノード選択] ダイアログが表示されます。マッピング元に変換元ノードを指定する場合にクリックします。

[ノード選択] ダイアログについては、「11.5.4 ノード選択ダイアログ(データ変換)」を参照してください。

### [ファンクション選択] ボタン

[ファンクション選択] ダイアログが表示されます。マッピング元にファンクションを指定する場合にクリックします。

[ファンクション選択] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

### [条件式] ラジオボタン

指定した条件式を条件として設定する場合に選択します。

### [左辺]

### [Position] ラジオボタン

マッピング元のインスタスの出現する順番を条件として設定する場合に選択します。

例えば、「Position=2」(このラジオボタンを選択し、右辺の値に「2」を指定)と指定した場合、2番目に出現するインスタスを条件として設定します。

### [ノード・ファンクション] ラジオボタン

変換元ノードまたはファンクションの値を条件として設定する場合に選択しま

す。

[ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。条件式の左辺に変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

[ ファンクション選択 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。条件式の左辺にファンクションを指定する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

[ 演算 ] ドロップダウンリスト

条件式に設定する演算子が表示されます。表示される演算子を次に示します。

- =
- !=
- >
- <
- >=
- <=

[ 右辺 ]

[ 値 ] ラジオボタン

ユーザが指定した値を条件として設定する場合に選択します。指定できる文字数は、1,024 文字までです。

[ ノード・ファンクション ] ラジオボタン

変換元ノードまたはファンクションの値を条件として設定する場合に選択します。

[ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。条件式の右辺に変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

[ ファンクション選択 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。条件式の右辺にファンクションを指定する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

[ 二つの条件の論理積・論理和 ] ラジオボタン

二つの条件を設定して、その条件の論理積または論理和を条件に設定する場合に選択します。

## 11. 開発環境で使用する画面

### [条件1]

一つ目の条件が表示されます。

### [条件設定] ボタン

条件設定ダイアログが表示されます。一つ目の条件を設定する場合にクリックします。

### [演算] ドロップダウンリスト

and (論理積) または or (論理和) が表示されます。

### [条件2]

二つ目の条件が表示されます。

### [条件設定] ボタン

条件設定ダイアログが表示されます。二つ目の条件を設定する場合にクリックします。

### [OK] ボタン

設定した内容を有効にして、ダイアログを閉じます。

### [キャンセル] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

条件設定ダイアログでは、次のことができます。

- ノード条件を設定する  
ノード条件を設定する方法については、「7.7 ノード条件を設定する」を参照してください。
- 選択ファンクションの条件を設定する  
選択ファンクションの条件を設定する方法については、「7.5.14 条件によって異なる値を出力する」を参照してください。

## 11.5.8 設定ダイアログ

設定ダイアログは、データ変換定義画面の表示を設定するときに使用します。

設定ダイアログを表示する方法については、「11.9.4 マッピング線の色を変更する」を参照してください。

設定ダイアログを次に示します。

図 11-72 設定ダイアログ



## (1) 表示項目

### [ 線の色 ]

#### [ 代入 ]

[ 代入 ] の右側にあるボタンをクリックすると、[ 色の設定 ] ダイアログが表示されます。

#### [ くり返し ]

[ くり返し ] の右側にあるボタンをクリックすると、[ 色の設定 ] ダイアログが表示されます。

#### [ 条件 ]

[ 条件 ] の右側にあるボタンをクリックすると、[ 色の設定 ] ダイアログが表示されます。

### [ デフォルトの復元 ] ボタン

クリックするとマッピング線の色がデフォルトに戻ります。デフォルトはグレーで

## 11. 開発環境で使用する画面

す。

### [適用] ボタン

設定した内容を有効にします。ダイアログは閉じません。

### [OK] ボタン

設定した内容を有効にして、ダイアログを閉じます。

### [キャンセル] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

設定ダイアログでは、マッピング線の色を変更することができます。マッピング線の色を変更する方法については、「11.9.4 マッピング線の色を変更する」を参照してください。

## 11.5.9 文字列連結ダイアログ

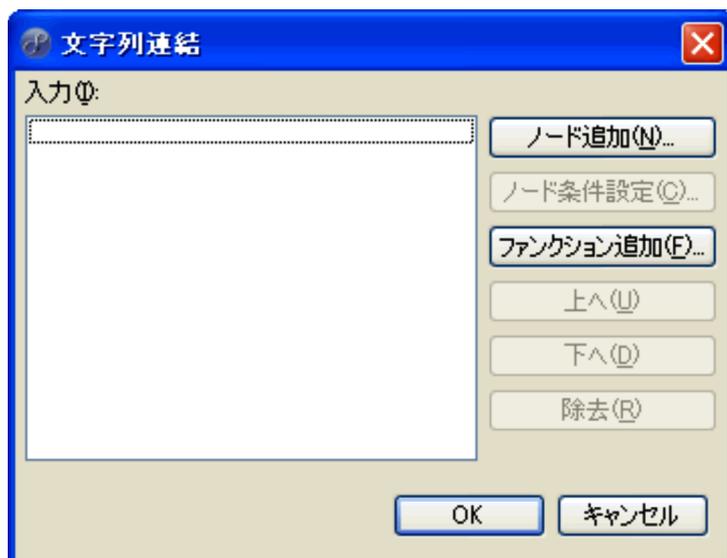
文字列連結ダイアログでは、連結対象となる文字列を指定します。

次のどちらかの操作をした場合に表示されます。

- 文字列連結ファンクションを右クリックし、[設定]を選択する
- 文字列連結ファンクションをダブルクリックする

文字列連結ダイアログを次に示します。

図 11-73 文字列連結ダイアログ



## (1) 表示項目

### [入力]

連結対象となる文字列の変換元ノードおよびファンクションを指定します。すでに連結対象と文字列連結ファンクションがマッピングされている場合、その連結対象が表示されます。

[入力]での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

### [ノード追加]ボタン

[ノード選択]ダイアログが表示されます。[入力]に連結対象となる文字列の変換元ノードを追加する場合にクリックします。

[ノード選択]ダイアログについては、「11.5.4 ノード選択ダイアログ(データ変換)」を参照してください。

### [ノード条件設定]ボタン

[ノード条件設定]ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ノード条件設定]ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

### [ファンクション追加]ボタン

[ファンクション選択]ダイアログが表示されます。[入力]に連結対象となる文字列のファンクションを追加する場合にクリックします。

[ファンクション選択]ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

### [上へ]ボタン

[入力]で選択した変換元ノードまたはファンクションの位置が一つ上に移動します。

### [下へ]ボタン

[入力]で選択した変換元ノードまたはファンクションの位置が一つ下に移動します。

### [除去]ボタン

[入力]で選択した変換元ノードまたはファンクションが[入力]から削除されます。

### [OK]ボタン

設定した内容を有効にして、ダイアログを閉じます。

### [キャンセル]ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

文字列連結ダイアログでは、文字列連結ファンクションを設定できます。文字列連結ファンクションの設定方法については、「7.5.1 複数の文字列を連結する」を参照してください。

### 11.5.10 部分文字列取得ダイアログ

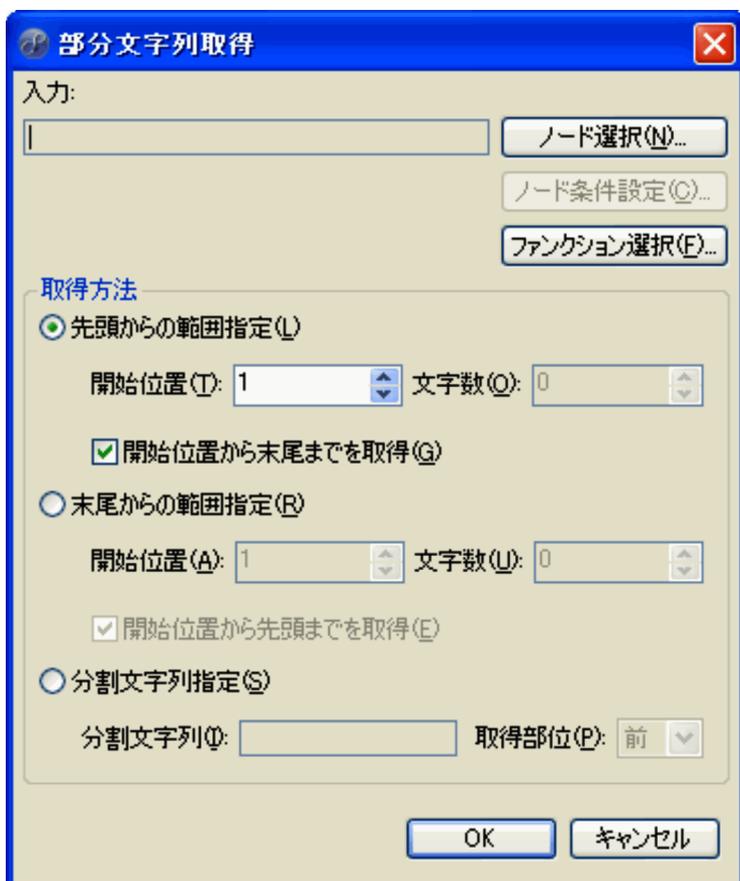
部分文字列取得ダイアログでは、一つの文字列から取り出す一部分の文字列を指定します。

次のどちらかの操作をした場合に表示されます。

- 部分文字列取得ファンクションを右クリックし、[設定]を選択する
- 部分文字列取得ファンクションをダブルクリックする

部分文字列取得ダイアログを次に示します。

図 11-74 部分文字列取得ダイアログ



## (1) 表示項目

### [ 入力 ]

一部分の文字列の取り出し元となる変換元ノードおよびファンクションを指定します。すでに取り出し元と部分文字列取得がマッピングされている場合、その取り出し元が表示されます。

[ 入力 ] での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

### [ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ 入力 ] に一部分の文字列の取り出し元となる変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

### [ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

### [ ファンクション選択 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。[ 入力 ] に一部分の文字列の取り出し元となるファンクションを指定する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

### [ 取得方法 ]

#### [ 先頭からの範囲指定 ] ラジオボタン

取り出す範囲を文字列の先頭から指定する場合に選択します。

#### [ 開始位置 ]

取り出す文字列の開始文字の位置を指定します。

#### [ 文字数 ]

取り出す文字列の文字数を指定します。指定できる文字数は0から2,147,483,647です。

#### [ 開始位置から末尾までを取得 ] チェックボックス

[ 開始位置 ] で指定した開始文字の位置から末尾までの文字列を取り出したい場合にチェックします。

#### [ 末尾からの範囲指定 ] ラジオボタン

取り出す範囲を文字列の末尾から指定する場合に選択します。

#### [ 開始位置 ]

取り出す文字列の開始文字の位置を指定します。

## 11. 開発環境で使用する画面

### [ 文字数 ]

取り出す文字列の文字数を指定します。指定できる文字数は0から2,147,483,647です。

### [ 開始位置から先頭までを取得 ] チェックボックス

[ 開始位置 ] で指定した開始文字の位置から先頭までの文字列を取り出したい場合にチェックします。

### [ 分割文字列指定 ] ラジオボタン

文字列を二つに分割して、分割した前の部分かあとの部分かのどちらかを指定する場合に選択します。

### [ 分割文字列 ]

マッピング元の文字列を分割する文字列を指定します。指定できる文字数は、1,024文字までです。

### [ 取得部位 ]

前または後が表示されます。

- 前  
先頭の文字から [ 分割文字列 ] で指定した文字列より前の文字列を取得します。
- 後  
[ 分割文字列 ] で指定した文字列以降の文字列を取得します。

### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

### [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

部分文字列取得ダイアログでは、部分文字列取得ファンクションを設定できます。部分文字列取得ファンクションの設定方法については、「7.5.2 一つの文字列から一部分の文字列を取り出す」を参照してください。

### 11.5.11 文字数取得ダイアログ

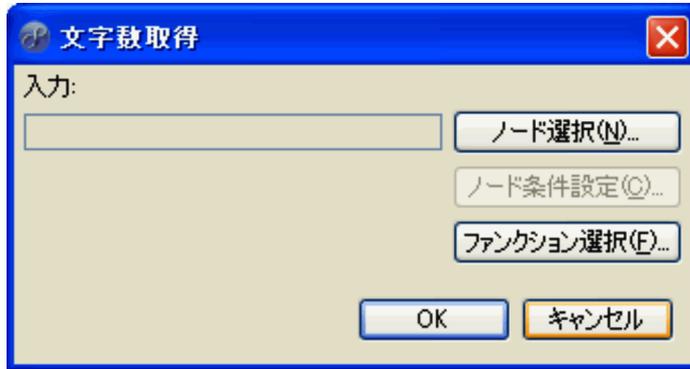
文字数取得ダイアログでは、文字数を取得したい文字列を指定します。

次のどちらかの操作をした場合に表示されます。

- 文字数取得ファンクションを右クリックし、[ 設定 ] を選択する
- 文字数取得ファンクションをダブルクリックする

文字数取得ダイアログを次に示します。

図 11-75 文字数取得ダイアログ



### (1) 表示項目

#### [ 入力 ]

文字数を取得したい文字列の変換元ノードおよびファンクションを指定します。すでに文字数の取得対象と文字数取得ファンクションがマッピングされている場合、その文字数の取得対象が表示されます。

[ 入力 ] での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

#### [ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ 入力 ] に文字数を取得したい文字列の変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

#### [ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

#### [ ファンクション選択 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。[ 入力 ] に文字数を取得したい文字列のファンクションを指定する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

#### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

## 11. 開発環境で使用する画面

### [キャンセル] ボタン

設定した内容を無効にして、ダイアログを閉じます。

### (2) このダイアログでできる内容

文字数取得ダイアログでは、文字数取得ファンクションを設定できます。文字数取得ファンクションの設定方法については、「7.5.3 文字列の文字数を代入する」を参照してください。

## 11.5.12 文字列調査ダイアログ

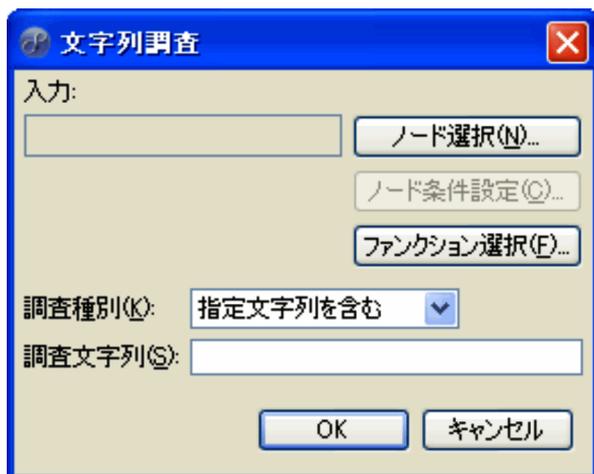
文字列調査ダイアログでは、指定した文字列が含まれているかまたは指定した文字列で開始されているかのどちらかを調査するかを選択し、調査対象となる文字列を指定します。

次のどちらかの操作をした場合に表示されます。

- 文字列調査ファンクションを右クリックし、[設定] を選択する
- 文字列調査ファンクションをダブルクリックする

文字列調査ダイアログを次に示します。

図 11-76 文字列調査ダイアログ



### (1) 表示項目

#### [入力]

調査対象となる文字列の変換元ノードおよびファンクションを指定します。すでに調査対象と文字列調査ファンクションがマッピングされている場合、その調査対象が表示されます。

[入力] での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

## [ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ 入力 ] に調査対象となる文字列の変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

## [ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

## [ ファンクション選択 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。[ 入力 ] に調査対象となる文字列のファンクションを指定する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

## [ 調査種別 ]

[ 調査文字列 ] で指定した文字列に対する調査種別が表示されます。表示される調査種別を次に示します。

- 指定文字列を含む
- 指定文字列から開始

## [ 調査文字列 ]

調査対象となる文字列を指定します。指定できる文字数は、1,024 文字までです。

## [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

## [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

文字列調査ダイアログでは、文字列調査ファンクションを設定できます。文字列調査ファンクションの設定方法については、「7.5.4 指定文字列の有無、または指定文字列の開始を調査する」を参照してください。

## 11.5.13 空白除去ダイアログ

空白除去ダイアログでは、空白の除去対象となる文字列を指定します。

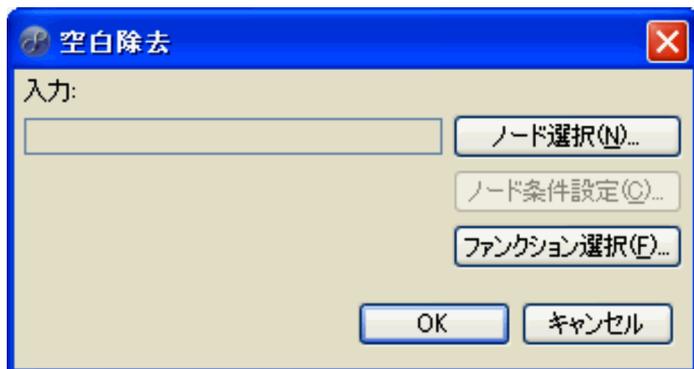
次のどちらかの操作をした場合に表示されます。

- 空白除去ファンクションを右クリックし、[ 設定 ] を選択する

- 空白除去ファンクションをダブルクリックする

空白除去ダイアログを次に示します。

図 11-77 空白除去ダイアログ



## (1) 表示項目

### [ 入力 ]

空白の除去対象となる文字列の変換元ノードおよびファンクションを指定します。すでに空白の除去対象と空白除去ファンクションがマッピングされている場合、その空白の除去対象が表示されます。

[ 入力 ] での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

### [ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ 入力 ] に空白の除去対象となる文字列の変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

### [ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

### [ ファンクション選択 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。[ 入力 ] に空白の除去対象となる文字列のファンクションを指定する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

## [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

## [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

空白除去ダイアログでは、空白除去ファンクションを設定できます。空白除去ファンクションの設定方法については、「7.5.5 文字列の空白を除去する」を参照してください。

### 11.5.14 数値フォーマット変換ダイアログ

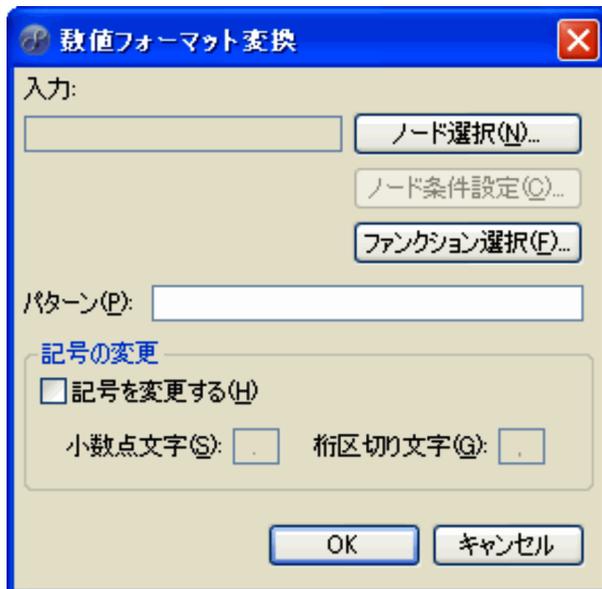
数値フォーマット変換ダイアログでは、変換後の `java.text.DecimalFormat` クラスの形式のパターン文字列を指定します。

次のどちらかの操作をした場合に表示されます。

- 数値フォーマット変換ファンクションを右クリックし、[ 設定 ] を選択する
- 数値フォーマット変換ファンクションをダブルクリックする

数値フォーマット変換ダイアログを次に示します。

図 11-78 数値フォーマット変換ダイアログ



## (1) 表示項目

## [ 入力 ]

変換対象となる数値の変換元ノードおよびファンクションを指定します。すでに変

## 11. 開発環境で使用する画面

換対象と数値フォーマット変換ファンクションがマッピングされている場合、その変換対象が表示されます。

[入力]での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

### [ノード選択] ボタン

[ノード選択] ダイアログが表示されます。[入力]に変換対象となる数値の変換元ノードを指定する場合にクリックします。

[ノード選択] ダイアログについては、「11.5.4 ノード選択ダイアログ(データ変換)」を参照してください。

### [ノード条件設定] ボタン

[ノード条件設定] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ノード条件設定] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

### [ファンクション選択] ボタン

[ファンクション選択] ダイアログが表示されます。[入力]に変換対象となる数値のファンクションを指定する場合にクリックします。

[ファンクション選択] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

### [パターン]

変換後の `java.text.DecimalFormat` クラスの形式のパターン文字列を指定します。指定できる文字数は、1,024 文字までです。

[記号の変更] を指定した場合は、変更後の記号を使用してパターンを記述してください。

### [記号の変更]

#### [記号を変更する] チェックボックス

小数点文字または桁区切り文字をデフォルトから変更する場合にチェックします。

#### [小数点文字]

小数点文字を変更する場合、変更後の小数点文字の記号を指定します。

#### [桁区切り文字]

桁区切り文字を変更する場合、変更後の桁区切り文字の記号を指定します。

### [OK] ボタン

設定した内容を有効にして、ダイアログを閉じます。

### [キャンセル] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

数値フォーマット変換ダイアログでは、数値フォーマット変換ファンクションを設定できます。数値フォーマット変換ファンクションの設定方法については、「7.5.6 数値をフォーマット変換する」を参照してください。

### 11.5.15 数値演算ダイアログ

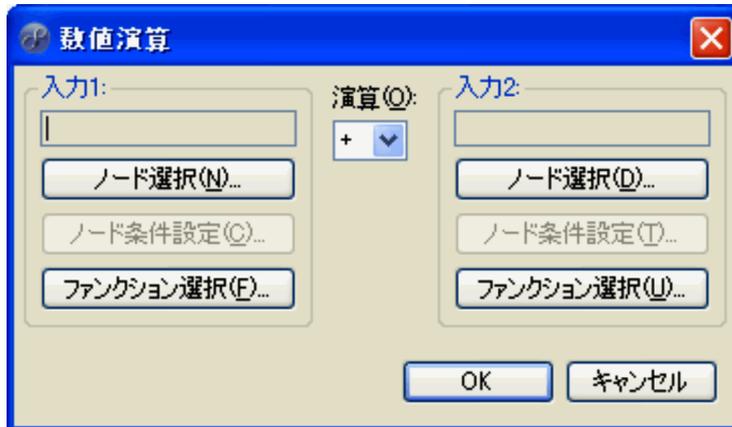
数値演算ダイアログでは、演算対象となる二つの数値と演算の種類を指定します。

次のどちらかの操作をした場合に表示されます。

- 数値演算ファンクションを右クリックし、[ 設定 ] を選択する
- 数値演算ファンクションをダブルクリックする

数値演算ダイアログを次に示します。

図 11-79 数値演算ダイアログ



## (1) 表示項目

### [ 入力 1 ]

演算対象となる一つ目の数値の変換元ノードおよびファンクションを指定します。すでに演算対象と数値演算ファンクションがマッピングされている場合、その演算対象が表示されます。

[ 入力 1 ] での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

### [ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ 入力 1 ] に演算対象となる数値の変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

## 11. 開発環境で使用する画面

### [ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

### [ ファンクション選択 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。[ 入力 1 ] に演算対象となる数値のファンクションを指定する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

### [ 演算 ]

実行できる演算が表示されます。表示される演算を次に示します。

- + (加算)
- - (減算)
- \* (乗算)
- / (除算)
- % (剰余)

### [ 入力 2 ]

演算対象となる二つ目の数値の変換元ノードおよびファンクションを指定します。表示形式は [ 入力 1 ] と同じです。

### [ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ 入力 2 ] に演算対象となる数値の変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

### [ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

### [ ファンクション選択 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。[ 入力 2 ] に演算対象となる数値のファンクションを指定する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

[キャンセル] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

数値演算ダイアログでは、数値演算ファンクションを設定できます。数値演算ファンクションの設定方法については、「7.5.7 数値を演算する」を参照してください。

### 11.5.16 端数処理ダイアログ

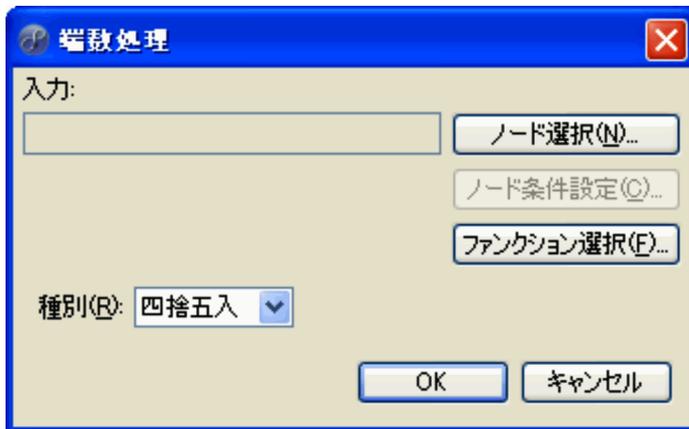
端数処理ダイアログでは、端数処理の対象となる数値を指定します。

次のどちらかの操作をした場合に表示されます。

- 端数処理ファンクションを右クリックし、[設定] を選択する
- 端数処理ファンクションをダブルクリックする

端数処理ダイアログを次に示します。

図 11-80 端数処理ダイアログ



## (1) 表示項目

[入力]

端数処理の対象となる数値の変換元ノードおよびファンクションを指定します。すでに端数処理の対象と端数処理ファンクションがマッピングされている場合、その端数処理の対象が表示されます。

[入力] での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

[ノード選択] ボタン

[ノード選択] ダイアログが表示されます。[入力] に端数処理の対象となる数値の変換元ノードを指定する場合にクリックします。

[ノード選択] ダイアログについては、「11.5.4 ノード選択ダイアログ(データ変

## 11. 開発環境で使用する画面

換)」を参照してください。

### [ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

### [ ファンクション選択 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。[ 入力 ] に端数処理の対象となる数値のファンクションを指定する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

### [ 種別 ]

実行できる端数処理が表示されます。表示される端数処理を次に示します。

- 四捨五入
- 切り捨て
- 切り上げ

### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

### [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

端数処理ダイアログでは、端数処理ファンクションを設定できます。端数処理ファンクションの設定方法については、「7.5.8 小数点以下の端数を処理する」を参照してください。

## 11.5.17 総和ダイアログ

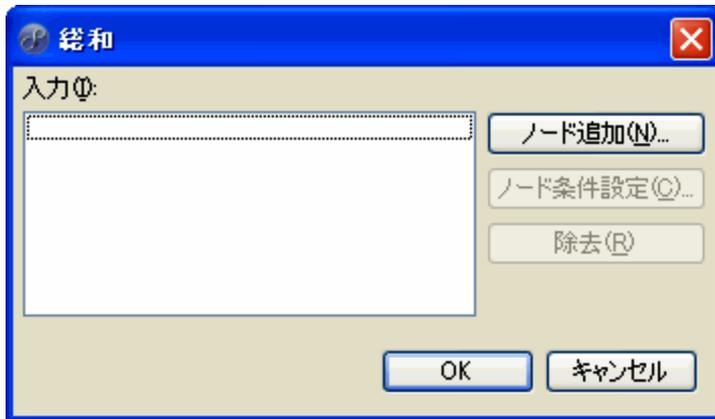
総和ダイアログでは、合計したいノードが含まれているノードセットを指定します。

次のどちらかの操作をした場合に表示されます。

- 総和ファンクションを右クリックし、[ 設定 ] を選択する
- 総和ファンクションをダブルクリックする

総和ダイアログを次に示します。

図 11-81 総和ダイアログ



### (1) 表示項目

#### [ 入力 ]

合計したいノードが含まれているノードセットを指定します。すでにノードセットと総和ファンクションがマッピングされている場合、そのノードセットが表示されます。

[ 入力 ] での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

#### [ ノード追加 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ 入力 ] に合計したいノードが含まれているノードセットを追加する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

#### [ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。ノードセットに条件を設定して、その条件を満たすノードセットだけをマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

#### [ 除去 ] ボタン

[ 入力 ] で選択したノードセットが [ 入力 ] から削除されます。

#### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

#### [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

総和ダイアログでは、総和ファンクションを設定できます。総和ファンクションの設定方法については、「7.5.9 複数のノードセットのノードの数値を合計する」を参照してください。

### 11.5.18 ノード数取得ダイアログ

ノード数取得ダイアログでは、ノード数を取得したいノードセットを指定します。

次のどちらかの操作をした場合に表示されます。

- ノード数取得ファンクションを右クリックし、[設定]を選択する
- ノード数取得ファンクションをダブルクリックする

ノード数取得ダイアログを次に示します。

図 11-82 ノード数取得ダイアログ



#### (1) 表示項目

##### [入力]

ノード数を取得したいノードセットを指定します。すでにノードセットとノード数取得ファンクションがマッピングされている場合、そのノードセットが表示されません。

[入力]での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

##### [ノード選択] ボタン

[ノード選択]ダイアログが表示されます。[入力]にノード数を取得したいノードセットを指定する場合にクリックします。

[ノード選択]ダイアログについては、「11.5.4 ノード選択ダイアログ(データ変換)」を参照してください。

##### [ノード条件設定] ボタン

[ノード条件設定]ダイアログが表示されます。ノードセットに条件を設定して、その条件を満たすノードセットだけをマッピングさせる場合にクリックします。

[ノード条件設定] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

[OK] ボタン

設定した内容を有効にして、ダイアログを閉じます。

[キャンセル] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

ノード数取得ダイアログでは、ノード数取得ファンクションを設定できます。ノード数取得ファンクションの設定方法については、「7.5.10 ノード数を代入する」を参照してください。

## 11.5.19 ノード名取得ダイアログ

ノード名取得ダイアログでは、ノード名を取得したい変換元ノードを指定します。

次のどちらかの操作をした場合に表示されます。

- ノード名取得ファンクションを右クリックし、[設定]を選択する
- ノード名取得ファンクションをダブルクリックする

ノード名取得ダイアログを次に示します。

図 11-83 ノード名取得ダイアログ



### (1) 表示項目

[入力]

ノード名を取得したい変換元ノードを指定します。すでにノード名を取得したい変換元ノードとノード名取得ファンクションがマッピングされている場合、その変換元ノードが表示されます。

[入力]での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

## 11. 開発環境で使用する画面

### [ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ 入力 ] にノード名を取得したい変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

### [ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

### [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

ノード名取得ダイアログでは、ノード名取得ファンクションを設定できます。ノード名取得ファンクションの設定方法については、「7.5.11 ノード名を代入する」を参照してください。

## 11.5.20 ノード調査ダイアログ

ノード調査ダイアログでは、ノードが存在するか調査したいマッピング元の変換元ノードを指定します。

次のどちらかの操作をした場合に表示されます。

- ノード調査ファンクションを右クリックし、[ 設定 ] を選択する
- ノード調査ファンクションをダブルクリックする

ノード調査ダイアログを次に示します。

図 11-84 ノード調査ダイアログ



## (1) 表示項目

### [ 入力 ]

ノードが存在するか調査したい変換元ノードを指定します。すでにノードが存在するか調査したい変換元ノードとノード調査ファンクションがマッピングされている場合、その変換元ノードが表示されます。

[ 入力 ] での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

### [ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ 入力 ] にノードが存在するか調査したい変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

### [ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

### [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

ノード調査ダイアログでは、ノード調査ファンクションを設定できます。ノード調査ファンクションの設定方法については、「7.5.12 ノードの有無を調査する」を参照してください。

## 11.5.21 くり返しダイアログ

くり返しダイアログでは、変換先ノードにくり返しを対応づけたい変換元ノードを指定します。また、複数回出現する変換元ノードのインスタンスのソート条件を設定します。

次のどちらかの操作をした場合に表示されます。

- くり返しファンクションを右クリックし、[ 設定 ] を選択する
- くり返しファンクションをダブルクリックする

くり返しダイアログを次に示します。

図 11-85 くり返しダイアログ



### (1) 表示項目

#### [ 基準パス ]

変換先ノードにくり返しを対応づけたい変換元ノードを指定します。すでにくり返しを対応づけたい変換元ノードとくり返しファンクションがマッピングされている場合、その変換元ノードが表示されます。

[ 基準パス ] での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

#### [ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ 基準パス ] にくり返しを対応づけたい変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変

換)」を参照してください。

[ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

[ ソート条件 ]

[ ソート条件追加 / 編集 ] ダイアログで設定した内容が表示されます。[ ソート条件 ] に設定できるキーは八つまでです。

[ ソート条件追加 / 編集 ] ダイアログについては、「11.5.22 ソート条件追加 / 編集ダイアログ」を参照してください。

[ 追加 ] ボタン

[ ソート条件追加 / 編集 ] ダイアログが表示されます。[ ソート条件 ] にキーを追加して、ソートする条件を設定する場合にクリックします。

[ ソート条件追加 / 編集 ] ダイアログについては、「11.5.22 ソート条件追加 / 編集ダイアログ」を参照してください。

[ 編集 ] ボタン

[ ソート条件追加 / 編集 ] ダイアログが表示されます。[ ソート条件 ] に設定されたキーのソート条件を編集するにクリックします。

[ ソート条件追加 / 編集 ] ダイアログについては、「11.5.22 ソート条件追加 / 編集ダイアログ」を参照してください。

[ 上へ ] ボタン

[ ソート条件 ] で選択したキーの位置が一つ上に移動します。

[ 下へ ] ボタン

[ ソート条件 ] で選択したキーの位置が一つ下に移動します。

[ 除去 ] ボタン

[ ソート条件 ] で選択したキーが [ ソート条件 ] から削除されます。

[ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

[ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

くり返しダイアログでは、くり返しファンクションを設定できます。くり返しファンクションの設定方法については、「7.5.13 くり返しを対応づける」を参照してください。

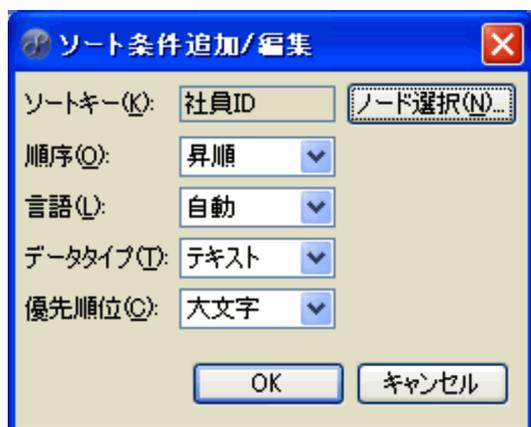
## 11.5.22 ソート条件追加 / 編集ダイアログ

ソート条件追加 / 編集ダイアログでは、[ くり返し ] ダイアログの [ ソート条件 ] にキーを追加して、ソート条件を設定します。また、キーに設定されたソート条件を編集します。

ソート条件追加 / 編集ダイアログは、[ くり返し ] ダイアログの [ 追加 ] ボタンまたは [ 編集 ] ボタンをクリックした場合に表示されます。

ソート条件追加 / 編集ダイアログを次に示します。

図 11-86 ソート条件追加 / 編集ダイアログ



### (1) 表示項目

#### [ ソートキー ]

ソートさせたいインスタンスの元となる変換元ノードを指定します。

#### [ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ ソートキー ] にソートさせたいインスタンスの元となる変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

#### [ 順序 ]

インスタンスをソートする順序が表示されます。表示される順序を次に示します。

- 昇順
- 降順

#### [ 言語 ]

インスタンスをソートする言語が表示されます。表示される言語を次に示します。

- 日本語
- 英語

- 自動

なお、「自動」を選択した場合、デフォルトに設定されているシステムのロケールに対応した言語コードが指定されます。

#### [ データタイプ ]

インスタンスをソートするデータタイプが表示されます。表示されるデータタイプを次に示します。

- テキスト
- 数値

#### [ 優先順位 ]

インスタンスをソートする優先順位が表示されます。[ データタイプ ] が「テキスト」の場合に選択します。表示される優先順位を次に示します。

- 大文字
- 小文字

#### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

#### [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

### (2) このダイアログでできる内容

ソート条件追加 / 編集ダイアログでは、くり返しファンクションのソート条件を設定できます。くり返しファンクションのソート条件の設定方法については、「7.5.13 くり返しを対応づける」を参照してください。

## 11.5.23 選択ダイアログ

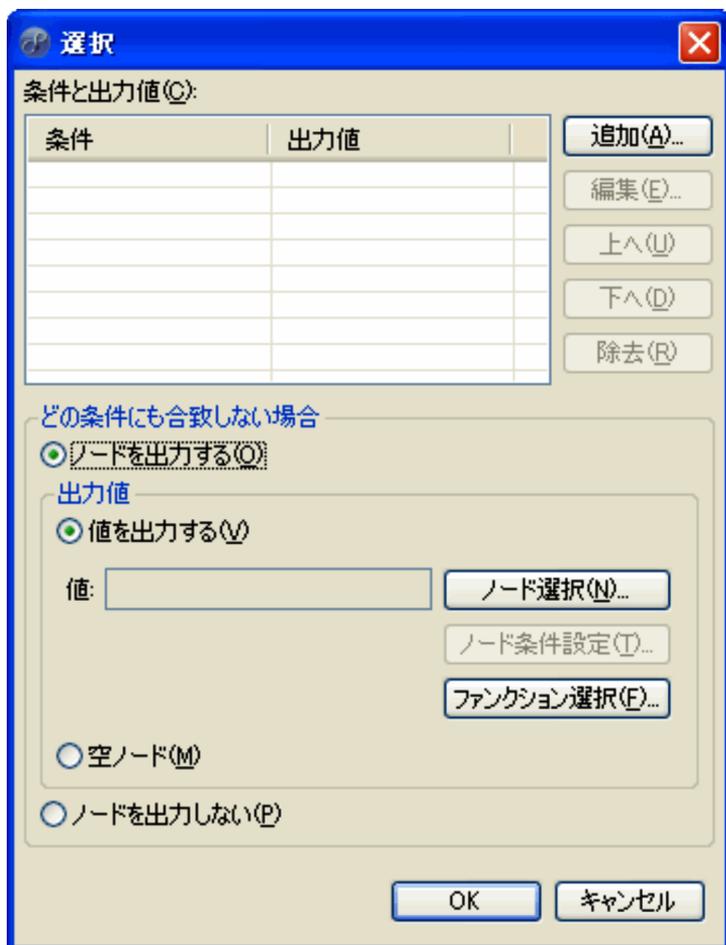
選択ダイアログでは、条件を設定して、その条件によって出力する変換元ノードまたはファンクションを指定します。

次のどちらかの操作をした場合に表示されます。

- 選択ファンクションを右クリックし、[ 設定 ] を選択する
- 選択ファンクションをダブルクリックする

選択ダイアログを次に示します。

図 11-87 選択ダイアログ



### (1) 表示項目

#### [ 条件と出力値 ]

##### [ 条件 ]

[ 条件追加 / 編集 ] ダイアログで設定した条件が表示されます。[ 条件追加 / 編集 ] ダイアログでの設定方法については、「11.5.24 条件追加 / 編集ダイアログ」を参照してください。

##### [ 出力値 ]

条件に合致する場合に出力する出力値の変換元ノードまたはファンクションを指定します。すでに出力値の変換元ノードまたはファンクションと選択ファンクションがマッピングされている場合、その変換元ノードまたはファンクションが表示されます。

[ 出力値 ] に表示される変換元ノードまたはファンクションについては、「7.4.4 マッピング元の表示形式」を参照してください。なお、このダイアログの

[空ノード] ラジオボタンを選択した指定した場合は「空ノード」が,[ノードを出力しない] ラジオボタンを選択した指定した場合は「出力なし」が表示されます。

[追加] ボタン

[条件追加/編集] ダイアログが表示されます。[条件と出力値] に条件および出力値を追加する場合にクリックします。

[編集] ボタン

[条件追加/編集] ダイアログが表示されます。[条件と出力値] の条件または出力値を編集する場合にクリックします。

[上へ] ボタン

[条件と出力値] で選択した条件および出力値の位置が一つ上に移動します。

[下へ] ボタン

[条件と出力値] で選択した条件および出力値の位置が一つ下に移動します。

[除去] ボタン

[条件と出力値] で選択した条件および出力値が [条件と出力値] から削除されます。

[どの条件にも合致しない場合]

[条件と出力値] で指定したすべての条件に合致しない場合の出力を設定します。

[ノードを出力する] ラジオボタン

変換先ノードを出力させるときに選択します。

[ノードを出力しない] ラジオボタン

変換先ノードを出力させないときに選択します。

[出力値]

[どの条件にも合致しない場合] の [ノードを出力する] ラジオボタンを選択した場合に設定します。

[値を出力する] ラジオボタン

変換元ノードまたはファンクションの値を出力させるときに選択します。

[値]

出力値の変換元ノードおよびファンクションを指定します。

出力値の表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

[ノード選択] ボタン

[ノード選択] ダイアログが表示されます。[値] に出力値の変換元ノードを指定する場合にクリックします。

[ノード選択] ダイアログについては、「11.5.4 ノード選択ダイアログ(データ変換)」を参照してください。

[ノード条件設定] ボタン

[ノード条件設定] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

## 11. 開発環境で使用する画面

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

[ ファンクション選択 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。[ 値 ] に出力値のファンクションを指定する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

[ 空ノード ] ラジオボタン

変換先ノードに値を出力させない（空ノードを出力）ときに選択します。

[ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

[ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

### (2) このダイアログでできる内容

選択ダイアログでは、選択ファンクションを設定できます。選択ファンクションの設定方法については、「7.5.14 条件によって異なる値を出力する」を参照してください。

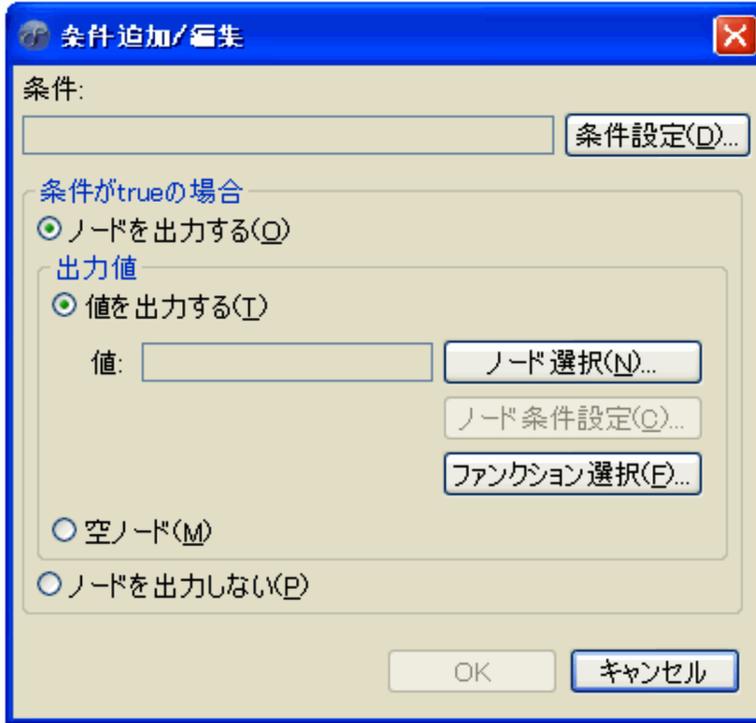
## 11.5.24 条件追加 / 編集ダイアログ

条件追加 / 編集ダイアログでは、[ 選択 ] ダイアログの [ 条件と出力値 ] に条件を追加します。また、条件と出力値の条件または出力値を編集します。

条件追加 / 編集ダイアログは、[ 選択 ] ダイアログの [ 追加 ] ボタンまたは [ 編集 ] ボタンをクリックした場合に表示されます。

条件追加 / 編集ダイアログを次に示します。

図 11-88 条件追加 / 編集ダイアログ



### (1) 表示項目

#### [ 条件 ]

[ 条件設定 ] ダイアログで設定した条件が表示されます。[ 条件設定 ] ダイアログでの設定方法については、「11.5.7 条件設定ダイアログ (ノード条件の設定)」を参照してください。

#### [ 条件が true の場合 ]

[ 条件 ] で指定した条件に合致する場合の出力を設定します。

#### [ ノードを出力する ] ラジオボタン

変換先ノードを出力させるときに選択します。

#### [ ノードを出力しない ] ラジオボタン

変換先ノードを出力させないときに選択します。

#### [ 出力値 ]

[ 条件が true の場合 ] の [ ノードを出力する ] ラジオボタンを選択した場合に設定します。

#### [ 値を出力する ] ラジオボタン

変換元ノードまたはファンクションの値を出力させるときに選択します。

[ 値 ]

出力値の変換元ノードおよびファンクションを指定します。

出力値の表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

[ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ 値 ] に出力値の変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

[ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

[ ファンクション選択 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。[ 値 ] に出力値のファンクションを指定する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

[ 空ノード ] ラジオボタン

変換先ノードに値を出力させない (空ノードを出力) ときに選択します。

[ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

[ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

条件追加 / 編集ダイアログでは、選択ファンクションの条件の追加・編集を設定できません。選択ファンクションの条件の追加・編集の設定方法については、「7.5.14 条件によって異なる値を出力する」を参照してください。

### 11.5.25 定数ダイアログ

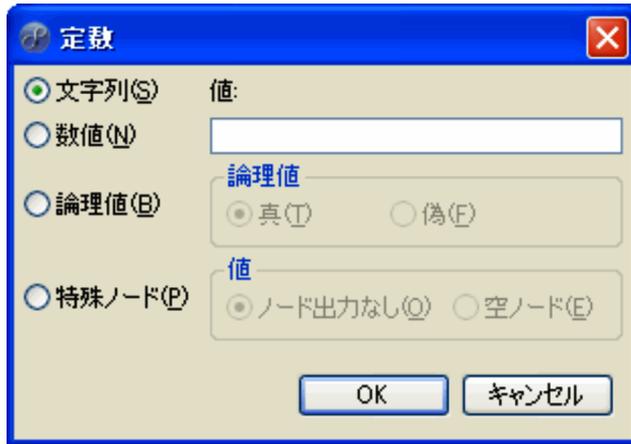
定数ダイアログでは、マッピング先に代入する値を指定します。

次のどちらかの操作をした場合に表示されます。

- 定数ファンクションを右クリックし、[ 設定 ] を選択する
- 定数ファンクションをダブルクリックする

定数ダイアログを次に示します。

図 11-89 定数ダイアログ



### (1) 表示項目

#### [ 文字列 ] ラジオボタン

マッピング先に代入する値が文字列の場合に選択します。

#### [ 値 ]

マッピング先に代入する文字列または数値を指定します。指定できる文字数は、1,024 文字までです。

#### [ 数値 ] ラジオボタン

マッピング先に代入する値が数値の場合に選択します。

#### [ 論理値 ] ラジオボタン

マッピング先に代入する値が論理値の場合に選択します。

#### [ 論理値 ]

##### [ 真 ] ラジオボタン

マッピング先に代入する論理値が真の場合に選択します。

##### [ 偽 ] ラジオボタン

マッピング先に代入する論理値が偽の場合に選択します。

#### [ 特殊ノード ] ラジオボタン

マッピング先に代入する値が特殊ノードの場合に選択します。

#### [ 値 ]

##### [ ノード出力なし ] ラジオボタン

マッピング先にノードを出力させない場合に選択します。

## 11. 開発環境で使用する画面

### [ 空ノード ] ラジオボタン

マッピング先のノードに値を代入しない(空ノード)場合に選択します。

### [ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

### [ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

## (2) このダイアログでできる内容

定数ダイアログでは、定数ファンクションを設定できます。定数ファンクションの設定方法については、「7.5.15 指定した値を代入する」を参照してください。

## 11.5.26 値置換えダイアログ

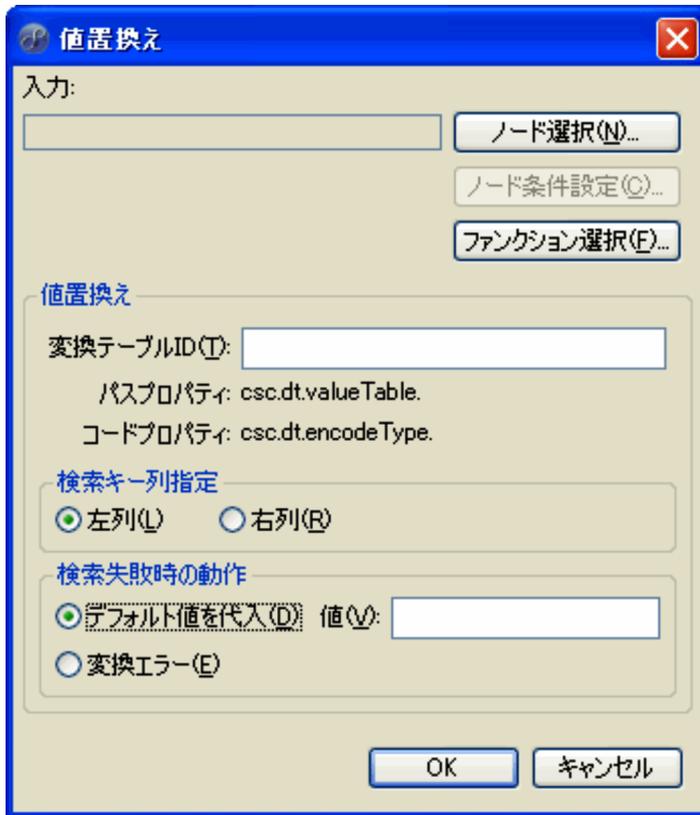
値置換えダイアログでは、マッピング元からの入力値を置き換えてマッピング先に代入するための変換テーブルを指定します。また、置き換え元の文字列を検索する列と、置き換え元の文字列が変換テーブルに存在しなかった場合の対処を指定します。

次のどちらかの操作をした場合に表示されます。

- 値置換えファンクションを右クリックし、[ 設定 ] を選択する
- 値置換えファンクションをダブルクリックする

値置換えダイアログを次に示します。

図 11-90 値置換えダイアログ



## (1) 表示項目

### [ 入力 ]

置き換えの対象となる文字列の変換元ノードおよびファンクションを指定します。すでに置き換えの対象と値置換えファンクションがマッピングされている場合、その置き換えの対象が表示されます。

[ 入力 ] での表示形式については、「7.4.4 マッピング元の表示形式」を参照してください。

### [ ノード選択 ] ボタン

[ ノード選択 ] ダイアログが表示されます。[ 入力 ] に置き換えの対象となる文字列の変換元ノードを指定する場合にクリックします。

[ ノード選択 ] ダイアログについては、「11.5.4 ノード選択ダイアログ (データ変換)」を参照してください。

### [ ノード条件設定 ] ボタン

[ ノード条件設定 ] ダイアログが表示されます。変換元ノードに条件を設定して、その条件を満たす場合だけマッピングさせる場合にクリックします。

[ ノード条件設定 ] ダイアログについては、「11.5.6 ノード条件設定ダイアログ」を

参照してください。また、ノード条件の設定方法については、「7.7 ノード条件を設定する」を参照してください。

[ ファンクション選択 ] ボタン

[ ファンクション選択 ] ダイアログが表示されます。[ 入力 ] に置き換えの対象となる文字列のファンクションを指定する場合にクリックします。

[ ファンクション選択 ] ダイアログについては、「11.5.5 ファンクション選択ダイアログ」を参照してください。

[ 値置換え ]

[ 変換テーブル ID ]

値の置き換えに使用する変換テーブルの変換テーブル ID を指定します。変換テーブル ID は、変換テーブルの作成後にシステムプロパティファイル (usrconf.properties) に記述した変換テーブルの名称です。

[ パスプロパティ ]

[ 変換テーブル ID ] に指定した変換テーブルの格納先が設定されているプロパティ (usrconf.properties のキー) が表示されます。次の形式で表示されます。

```
csc.dt.valueTable.<変換テーブルID>
```

[ コードプロパティ ]

[ 変換テーブル ID ] に指定した変換テーブルで使用している文字コードが設定されているプロパティ (usrconf.properties のキー) が表示されます。次の形式で表示されます。

```
csc.dt.encodeType.<変換テーブルID>
```

[ 検索キー列指定 ]

[ 左列 ] ラジオボタン

置き換え元の文字列を変換テーブルの左列から検索して、右列の文字列に置き換える場合に選択します。

[ 右列 ] ラジオボタン

置き換え元の文字列を変換テーブルの右列から検索して、左列の文字列に置き換える場合に選択します。

[ 検索失敗時の動作 ]

[ デフォルト値を代入 ]

置き換え元の文字列の検索失敗時に、あらかじめ指定した値をマッピング先に代入する場合に選択します。

[ 値 ]

置き換え元の文字列の検索失敗時に、マッピング先に代入するデフォルト値を指定します。[ デフォルト値を代入 ] を選択した場合に指定します。

[ 変換エラー ]

置き換え元の文字列の検索失敗時に、変換エラーを発生させる場合に選択します。

[ OK ] ボタン

設定した内容を有効にして、ダイアログを閉じます。

[ キャンセル ] ボタン

設定した内容を無効にして、ダイアログを閉じます。

(2) このダイアログでできる内容

値置換えダイアログでは、値置換えファンクションを設定できます。値置換えファンクションの設定方法については、「7.5.16 変換テーブルを利用して値を変換する」を参照してください。

## 11.6 ユーザ定義受付に関連するダイアログ

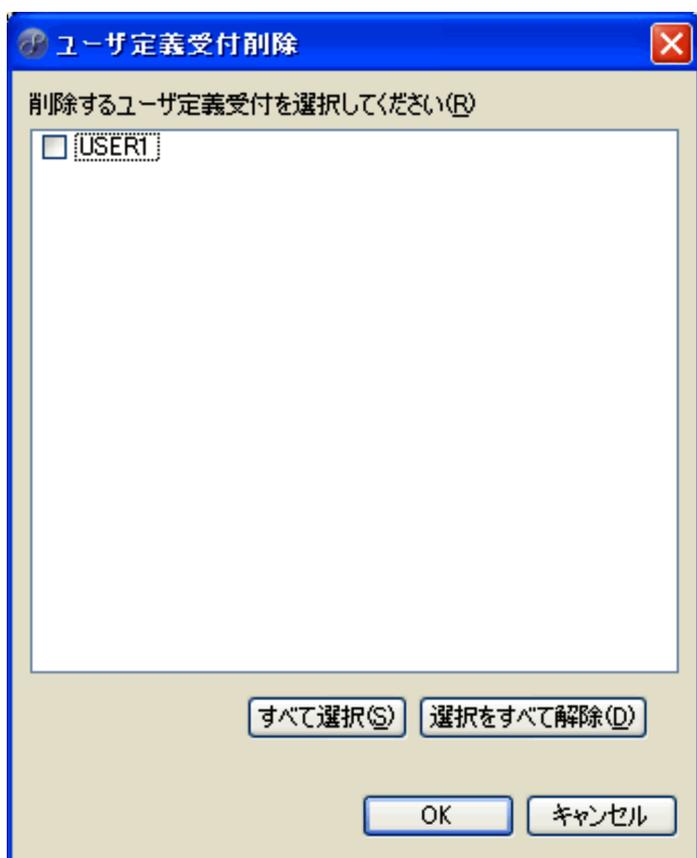
ユーザ定義受付に関連するダイアログについて説明します。

### 11.6.1 ユーザ定義受付削除ダイアログ

ユーザ定義受付を削除するためのダイアログです。

ユーザ定義受付の削除方法の詳細については、「8.7 ユーザ定義受付の削除」を参照してください。

図 11-91 ユーザ定義受付削除ダイアログ



[ 削除するユーザ定義受付を選択してください ]

定義されているユーザ定義受付の一覧が表示されます。削除するユーザ定義受付をチェックボックスで選択します。

[ すべて選択 ] ボタン

ユーザ定義受付の一覧のすべてのチェックボックスがチェックされます。

[ 選択をすべて解除 ] ボタン

ユーザ定義受付の一覧のすべてのチェックボックスのチェックが解除されます。

[ OK ] ボタン

指定したユーザ定義受付を削除して、ダイアログを閉じます。

[ キャンセル ] ボタン

ユーザ定義受付を削除しないでダイアログを閉じます。

## 11.7 バイナリフォーマット定義画面の操作

---

バイナリフォーマット定義画面での操作について説明します。

### ! 注意事項

以降の項での説明は、バイナリフォーマット定義画面内のバイナリフォーマット定義エディタの操作の説明です。バイナリフォーマット定義画面内のアウトラインビューには、バイナリフォーマット定義エディタでの操作結果が表示されます。

---

### 11.7.1 要素をコピーして貼り付ける

バイナリフォーマット定義エディタ上の要素をコピーして貼り付けます。

### ! 注意事項

コピーした要素は、ほかのバイナリフォーマット定義ファイルを編集しているバイナリフォーマット定義エディタ上には貼り付けできません。貼り付けできるのは、コピーした要素があるバイナリフォーマット定義エディタ上だけです。

---

要素をコピーして貼り付ける方法を次に示します。

1. コピーする要素を選択します。
2. 右クリックして [コピー] を選択するか, [Ctrl] キー + [C] キーを押します。
3. 貼り付け先の要素を選択します。
4. 右クリックして [貼り付け] を選択するか, [Ctrl] キー + [V] キーを押します。  
貼り付けた要素の名称は次の形式で自動設定されます (n は要素名をユニークにするための自然数)。

Copy- <n> - <コピー元の要素名>

ただし、ローカル定義の単純内容要素を別の複合内容要素の構成要素として貼り付ける場合で、同じ名称の構成要素がないときは、コピー元の要素と同じ名称の要素として貼り付けられます。

コピー元の要素と貼り付け先の要素の対応を次の表に示します。

表 11-8 コピー元の要素と貼り付け先の要素の対応

コピー元の要素	貼り付け先の要素					
	フォーマット	単純内容要素（グローバル定義）	複合内容要素	構成要素（単純内容要素（グローバル定義））	構成要素（複合内容要素）	構成要素（単純内容要素（ローカル定義））
フォーマット	-	-	-	-	-	-
単純内容要素（グローバル定義）	(1)	(2)	(2)	-	-	-
複合内容要素	(1)	(2)	(2)	-	-	-
構成要素（単純内容要素（グローバル定義））	-	-	-	-	-	-
構成要素（複合内容要素）	-	-	-	-	-	-
構成要素（単純内容要素（ローカル定義））	-	-	(3)	(4)	(4)	(4)

（凡例）

- : コピー，貼り付けはできません。

(1)

コピーしたグローバル定義の単純内容要素または複合内容要素と同じ内容で，グローバル定義の単純内容要素または複合内容要素が追加されます。すべてのグローバル定義の要素の最後尾に追加されます。

なお，複合内容要素をコピーした場合，複合内容要素に設定されている構成要素もコピーされます。

(2)

コピーしたグローバル定義の単純内容要素または複合内容要素と同じ内容で，グローバル定義の単純内容要素または複合内容要素が追加されます。貼り付け先に選択したグローバル定義の要素の上に追加されます。

なお，複合内容要素をコピーした場合，複合内容要素に設定されている構成要素もコピーされます。

(3)

コピーしたローカル定義の単純内容要素と同じ内容で，選択した複合内容要素にローカル定義の単純内容要素が追加されます。選択した複合内容要素の構成要素の最後尾に追加されます。

(4)

コピーしたローカル定義の単純内容要素と同じ内容で，ローカル定義の単純内容要素が追加されます。貼り付け先に選択した構成要素の上に追加されます。

**!** 注意事項

- グローバル定義の単純内容要素をコピーして、ローカル定義の単純内容要素として貼り付けることはできません。逆に、ローカル定義の単純内容要素をコピーして、グローバル定義の単純内容要素として貼り付けることもできません。
- バイナリデータとして CSV 形式のデータを利用する場合、コピーできるのは、複合内容要素（ヘッダー要素またはレコード要素）の構成要素として設定されている単純内容要素だけです。

## 11.7.2 要素を削除する

バイナリフォーマット定義エディタ上の要素を削除します。

要素を削除する方法を次に示します。

1. 削除する要素を選択します。
2. 右クリックして [ 削除 ] を選択するか, [ Delete ] キーを押します。

なお、次の要素の場合は削除できません。

- 構成要素から参照されているグローバル定義の単純内容要素および複合内容要素
- ルート要素に設定されている複合内容要素

また、次の要素の場合、削除してよいかを確認する警告メッセージが表示されます。

- ほかの要素の出現回数、サイズを決定する単純内容要素（ターゲットノード）
- ほかの要素の選択条件値を持つ単純内容要素（選択条件ノード）

## 11.7.3 要素を移動する

バイナリフォーマット定義エディタ上の要素は、移動する要素（ドラッグ元）を移動先（ドロップ先）にドラッグ&ドロップして移動できます。ドラッグ中に [ Esc ] キーを押すと、動作をキャンセルできます。

ドロップ先の種類と、ドラッグ時に [ Ctrl ] キーを押しているかどうかによって、ドラッグ&ドロップしたときの結果は異なります。

**!** 注意事項

- ドラッグ&ドロップできる範囲は、バイナリフォーマット定義エディタ内に限定されま  
す。バイナリフォーマット定義エディタの要素を、別のビューやエディタへドロップする  
ことも、別のビューやエディタの要素をバイナリフォーマット定義エディタにドロップす  
ることもできません。
- ドラッグ&ドロップによって、選択条件などの要素間の依存関係に問題が発生しても、ド  
ラッグ&ドロップした時点ではエラーになりません。バイナリフォーマットファイルの検  
証を実施したときにエラーが出力されます。
- バイナリデータとして CSV 形式のデータを利用する場合、ドラッグできるのはローカル  
単純内容要素だけです。ドラッグしたローカル単純内容要素をドロップできるのは、複合  
内容要素（ヘッダー要素またはレコード要素）、およびその構成要素の上辺と下辺です。

## 参考

ツリーがバイナリフォーマット定義エディタ内に収まらず、ドロップ先の要素が表示されて  
いない場合、ドラッグするときのカーソルをバイナリフォーマット定義エディタの枠の近く  
に移動させると自動的にスクロールされます。

また、ツリーが折りたたまれていてドロップ先の要素が表示されていない場合、ドラッグす  
るときのカーソルを折りたたまれている要素の近くに滞留させておくと自動的にツリーが展  
開されます。

**(1) ドロップ先の種類**

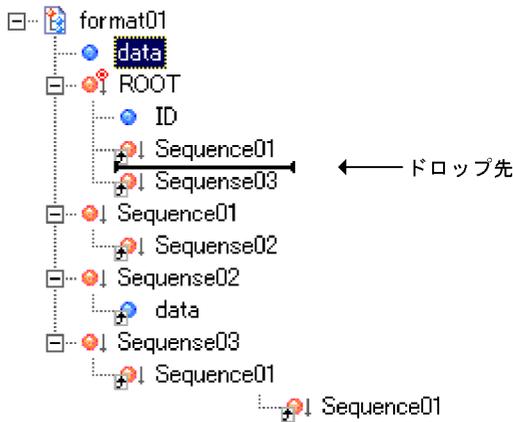
ドロップ先の種類には、要素、上辺および下辺があります。

要素をドロップ先にする場合、ドロップ先の要素にドラッグしたカーソルを移動すると、  
要素の表示色が反転します。

上辺および下辺をドロップ先にする場合、ドロップ位置に線（)が表示されま  
す。

上辺および下辺にドロップする場合の例を次に示します。

図 11-92 上辺および下辺にドロップする場合の例



上の図の例では、ドロップ先は構成要素「Sequence01」の下辺、かつ構成要素「Sequence03」の上辺になります。

## (2) ドラッグ&ドロップしたときの結果

ドラッグ&ドロップしたときの結果は、[Ctrl] キーを押しているかどうかによって異なります。

[Ctrl] キーを押さないでドラッグ&ドロップする場合

### 移動

要素を移動します。ドラッグ元の要素は削除され、ドロップ先に追加されます。

### 参照追加

グローバル定義の複合内容要素に構成要素を追加します。ドラッグ&ドロップして追加された構成要素の内容はデフォルト値になります。

[Ctrl] キーを押さないでドラッグ&ドロップした場合の、ドラッグ元とドロップ先の関係を次の表に示します。

表 11-9 ドラッグ&ドロップしたときの結果 ([Ctrl] キーを利用しない場合)

ドラッグ元	ドロップ先								
	単純内容要素 (グローバル定義)			複合内容要素 (グローバル定義)			構成要素		
	上辺	要素	下辺	上辺	要素	下辺	上辺	要素	下辺
単純内容要素 (グローバル定義)	移動	-	移動	移動	参照追加	移動	参照追加	-	参照追加
複合内容要素 (グローバル定義)	移動	-	移動	移動	参照追加	移動	参照追加	-	参照追加

ドラッグ元	ドロップ先								
	単純内容要素（グローバル定義）			複合内容要素（グローバル定義）			構成要素		
	上辺	要素	下辺	上辺	要素	下辺	上辺	要素	下辺
構成要素（単純内容要素）	-	-	-	-	-	-	移動	-	移動
構成要素（複合内容要素）	-	-	-	-	-	-	移動	-	移動
構成要素（単純内容要素（ローカル定義））	-	-	-	-	-	-	移動	-	移動

（凡例）

- : ドラッグ&ドロップできません。

[ Ctrl ] キーを押したままドラッグ&ドロップする場合

コピー追加

要素をコピーして追加します。

追加された要素の設定情報は、ドラッグ元の要素の情報が引き継がれます。

追加された要素の名称は次の形式で自動設定されます（n は要素名をユニークにするための自然数）。

Copy- <n> - <ドラッグ元の要素名>

複合内容要素をコピー追加した場合、ドラッグ元の複合内容要素に設定されている構成要素も引き継がれます。

[ Ctrl ] キーを押したままドラッグ&ドロップした場合の、ドラッグ元とドロップ先の関係を次の表に示します。

表 11-10 ドラッグ&ドロップしたときの結果 ([ Ctrl ] キーを利用した場合)

ドラッグ元	ドロップ先											
	フォーマット			単純内容要素（グローバル定義）			複合内容要素（グローバル定義）			構成要素		
	上辺	要素	下辺	上辺	要素	下辺	上辺	要素	下辺	上辺	要素	下辺
単純内容要素（グローバル定義）	-		-		-			-		-	-	-
複合内容要素（グローバル定義）	-		-		-			-		-	-	-

## 11. 開発環境で使用する画面

ドラッグ元	ドロップ先											
	フォーマット			単純内容要素 (グローバル定義)			複合内容要素 (グローバル定義)			構成要素		
	上 辺	要 素	下 辺	上 辺	要 素	下 辺	上 辺	要 素	下 辺	上 辺	要 素	下 辺
構成要素 (単純内容要素)	-	-	-	-	-	-	-	-	-	-	-	-
構成要素 (複合内容要素)	-	-	-	-	-	-	-	-	-	-	-	-
構成要素 (単純内容要素 (ローカル定義))	-	-	-	-	-	-	-	-	-	-	-	-

(凡例)

- : ドラッグ&ドロップできません。

: コピー追加できます。

### 11.7.4 操作を元に戻す

バイナリフォーマット定義エディタで行った操作を元に戻します。操作は何回でも元に戻すことができます。

操作を元に戻すには次の二つの方法があります。

方法 1

バイナリフォーマット定義エディタ上で右クリックして、[元に戻す] を選択します。

方法 2

[Ctrl] キー + [Z] キーを押します。

### 11.7.5 操作をやり直す

バイナリフォーマット定義エディタで行った操作をやり直します。操作は何回でもやり直すことができます。

操作をやり直すには次の二つの方法があります。

方法 1

バイナリフォーマット定義エディタ上で右クリックして、[やり直し] を選択します。

方法 2

[Ctrl] キー + [Y] キーを押します。

## 11.8 ビジネスプロセス定義画面の操作

---

ビジネスプロセス定義画面での操作について説明します。

### 11.8.1 オブジェクトをコピーする

キャンバス上のオブジェクト（アクティビティ、コネクション、フォルトコネクション、およびリンクコネクション）をコピーします。オブジェクトを複数選択した場合、選択したすべてのオブジェクトをコピーします。オブジェクトはビジネスプロセス画面のキャンバス以外に、Word、Excel などほかのアプリケーションにもコピーできます。ただし、開始アクティビティおよび終了アクティビティは、ほかのアプリケーションにはコピーできませんが、ビジネスプロセス画面のキャンバスにはコピーはできません。

オブジェクトをコピーする手順を次に示します。

1. コピーしたいオブジェクトを選択します。  
コネクションをコピーする場合、連結元のアクティビティおよび連結先のアクティビティも一緒に選択してください。連結元のアクティビティおよび連結先のアクティビティを一緒に選択しないと、コネクションはコピーされません。
2. 次のどちらかの方法でコピーします。
  - 右クリックして、[ コピー ] を選択します。
  - [ Ctrl ] キー + [ c ] キーを押します。

### 11.8.2 オブジェクトを切り取る

キャンバス上のオブジェクト（アクティビティ、コネクション、フォルトコネクション、およびリンクコネクション）を切り取ります。切り取ることができるオブジェクトなどはコピーの場合と同じです。詳細については、「11.8.1 オブジェクトをコピーする」を参照してください。

オブジェクトを切り取る手順を次に示します。

1. 切り取りたいオブジェクトを選択します。  
コネクションを切り取る場合、連結元のアクティビティおよび連結先のアクティビティも一緒に選択してください。連結元のアクティビティおよび連結先のアクティビティを一緒に選択しないと、コネクションを切り取ることができません。
2. 次のどちらかの方法で切り取ります。
  - 右クリックして、[ 切り取り ] を選択します。
  - [ Ctrl ] キー + [ x ] キーを押します。

### 11.8.3 オブジェクトを貼り付ける

コピーまたは切り取ったオブジェクト（アクティビティ、コネクション、フォルトコネクション、およびリンクコネクション）を、ビジネスプロセス画面のキャンバスまたは

## 11. 開発環境で使用する画面

ほかのアプリケーションに貼り付けます。

オブジェクトを貼り付ける方法を、キャンパスの場合とほかのアプリケーションの場合とに分けて次に示します。

### キャンパスの場合

次のどちらかの方法で貼り付けます。

- アクティビティが選択されていない状態でキャンパス上を右クリックして、[ 貼り付け ] を選択します。
- [ Ctrl ] キー + [ v ] キーを押します。

### ほかのアプリケーションの場合

[ Ctrl ] キー + [ v ] キーを押します。

なお、貼り付け先によって、コピーまたは切り取り元から貼り付けられるアクティビティの内容が異なります。貼り付け先によるアクティビティの内容の違いを次の表に示します。

表 11-11 貼り付け先によるアクティビティの内容の違い

コピーまたは切り取り元の アクティビティ	貼り付け先			
	同じキャンパス 内	同じビジネスブ ロセスの定義内 (別のスコープ または別の繰り 返し)	別のビジネスブ ロセスの定義	別のアプリケー ション
開始アクティビティ	×	×	×	
終了アクティビティ	×	×	×	
データ変換アクティビティ				
スコープアクティビティ				
繰り返しアクティビティ				
上記以外のアクティビティ				

### ( 凡例 )

: 次の内容が貼り付けられます。

- アクティビティのアイコン
- アクティビティの定義情報 ( 変数の情報はコピーされません )

: 次の内容が貼り付けられます。

- アクティビティのアイコン

新規にアクティビティを配置した場合と同じ状態になります。

: 次の内容が貼り付けられます。

- アクティビティのアイコン

コピー先では定義できません。

x : 貼り付けられません。

#### 11.8.4 オブジェクトを削除する

キャンバス上のオブジェクト（アクティビティ、コネクション、フォルトコネクション、およびリンクコネクション）を削除します。オブジェクトを複数選択した場合、選択したすべてのオブジェクトを削除します。

オブジェクトを削除する手順を次に示します。

1. 削除したいオブジェクトを選択します。
2. 次のどちらかの方法で削除します。
  - 右クリックして、[ 削除 ] を選択します。
  - [ Delete ] キーを押します。

なお、連結元のアクティビティまたは連結先のアクティビティを削除すると、連結されていたコネクション、フォルトコネクション、およびリンクコネクションも削除されます。

#### 11.8.5 オブジェクトを整列する

キャンバス上で選択した複数のオブジェクト（アクティビティ、コネクション、フォルトコネクション、およびリンクコネクション）を整列します。整列の種類は、次から選択できます。

- 左揃え
- 中央揃え（水平方向）
- 右揃え
- 上揃え
- 中央揃え（垂直方向）
- 下揃え

なお、コネクション、フォルトコネクション、およびリンクコネクションは、アクティビティの整列に伴い移動されるだけです。

オブジェクトを整列する手順を次に示します。

1. 整列したいオブジェクトを選択します。

整列したいオブジェクトの中には、必ず二つ以上のアクティビティを含めてください。
2. 右クリックして、[ 整列 ] を指定し、整列の種類を選択します。

#### 11.8.6 操作を元に戻す

キャンバス上で行った操作を元に戻します。元に戻す操作の対象は、ビジネスプロセス

## 11. 開発環境で使用する画面

ごとに管理されています。よって、同じビジネスプロセス内の別のスコープアクティビティまたは繰り返しアクティビティの操作の場合は、元に戻すことができます。また、操作は 20 回まで元に戻すことができます。なお、次の操作は元に戻すことができません。

- プロパティビューでの操作。
- スコープのタブまたは繰り返しのタブを、開くまたは閉じる操作。ただし、操作を元に戻してスコープアクティビティおよび繰り返しアクティビティを削除した場合、スコープのタブまたは繰り返しのタブは削除されます。
- データ変換定義画面での操作。
- 表示を拡大または縮小する操作。

操作を元に戻すには、次の二つの方法があります。

### 方法 1

アクティビティが選択されていない状態でキャンバス上を右クリックして、[元に戻す]を選択します。

### 方法 2

[Ctrl] キー + [z] キーを押します。

#### **!** 注意事項

ビジネスプロセスを保存した時点で、操作が初期化されます。したがって、ビジネスプロセスを保存すると、操作を元に戻すことができません

## 11.8.7 操作をやり直す

元に戻した操作をやり直します。やり直す操作の対象は、元に戻す操作と同様に、ビジネスプロセスごとに管理されています。また、やり直すことができる操作の回数は、連続して元に戻した回数までとなります。

操作をやり直すには、次の二つの方法があります。

### 方法 1

アクティビティが選択されていない状態でキャンバス上を右クリックして、[やり直す]を選択します。

### 方法 2

[Ctrl] キー + [y] キーを押します。

#### **!** 注意事項

ビジネスプロセスを保存した時点で、操作が初期化されます。したがって、ビジネスプロセスを保存すると、操作を元に戻すことができません。

### 11.8.8 表示を拡大する

キャンバスの表示を拡大します。表示は 200% まで拡大できます。

表示を拡大するには、次の三つの方法があります。

#### 方法 1

アクティビティが選択されていない状態でキャンバス上を右クリックして、[ 拡大 ] を選択します。

#### 方法 2

[ Ctrl ] キー + [ + ] キーを押します。

#### 方法 3

[ Ctrl ] キー + ホイールマウスを使用します。

### 11.8.9 表示を縮小する

キャンバスの表示を縮小します。表示は 10% まで縮小できます。

表示を縮小するには、次の三つの方法があります。

#### 方法 1

アクティビティが選択されていない状態でキャンバス上を右クリックして、[ 縮小 ] を選択します。

#### 方法 2

[ Ctrl ] キー + [ - ] キーを押します。

#### 方法 3

[ Ctrl ] キー + ホイールマウスを使用します。

### 11.8.10 表示を元のサイズに戻す

拡大または縮小したキャンバスの表示を元のサイズ ( 100% ) に戻します。

次の方法で表示を元サイズに戻します。

#### 方法

アクティビティが選択されていない状態でキャンバス上を右クリックして、[ 元のサイズに戻す ] を選択します。

### 11.8.11 パレットの表示を設定する

パレットの表示を設定します。設定できる内容を次に示します。

- グループおよびツールのレイアウト

## 11. 開発環境で使用する画面

- アイコンのサイズ
- フォント
- グループ下のツール一覧の表示または非表示

パレットの表示項目については、「11.2.3(1) パレット」を参照してください。なお、ここで設定した内容は、すべてのビジネスプロセス画面で適用されます。

### (1) グループおよびツールのレイアウトを設定する

パレットのグループおよびツールは、次のようにレイアウトを設定できます。

- 列  
すべてのグループおよびツールを水平に整列します。
- リスト  
すべてのグループおよびツールを垂直に整列します。
- アイコンのみ  
すべてのツールに対して、ツール名を表示せずアイコンだけの表示にします。
- 詳細  
すべてのツールに対して、詳細なフォーマットおよび記述を表示します。

グループおよびツールのレイアウトを設定するには、次の二つの方法があります。

#### 方法 1

パレットのグループまたはツールを右クリックして、[ レイアウト ] から設定したい項目を選択します。

#### 方法 2

1. パレットのグループまたはツールを右クリックして、[ 設定 ] を選択します。  
[ パレット設定 ] ダイアログが表示されます。
2. [ レイアウト ] から設定したい項目を選択します。

### (2) アイコンのサイズを設定する

パレットのツールのアイコンは、サイズを大きくできます。

ツールのアイコンのサイズを大きくするには、次の二つの方法があります。

#### 方法 1

パレットのグループまたはツールを右クリックして、[ 大きいアイコンを使用 ] を選択します。

#### 方法 2

1. パレットのグループまたはツールを右クリックして、[ 設定 ] を選択します。  
[ パレット設定 ] ダイアログが表示されます。
2. [ リスト・レイアウト・オプション ] の [ 大きいアイコンを使用 ] チェックボックスにチェックします。

### (3) フォントを設定する

パレットのグループおよびツールのフォントは変更できます。また、変更したフォントはデフォルトに戻すことができます。

フォントを設定する方法を、フォントを変更する場合とフォントをデフォルトに戻す場合とに分けて次に示します。

フォントを変更する場合

1. パレットのグループまたはツールを右クリックして、[設定]を選択します。  
[パレット設定]ダイアログが表示されます。
2. [フォント]の[変更]ボタンをクリックします。

フォントをデフォルトに戻す場合

1. パレットのグループまたはツールを右クリックして、[設定]を選択します。  
[パレット設定]ダイアログが表示されます。
2. [フォント]の[デフォルトの復元]ボタンをクリックします。

### (4) ツールの一覧の表示または非表示を設定する

パレットに表示されているツールの一覧を非表示にすることができます。また、非表示にしたツールの一覧を表示させることができます。ツールの一覧の表示および非表示の例を次に示します。

図 11-93 ツールの一覧の表示および非表示



ツールの一覧を表示または非表示するには、次の二つの方法があります。

方法 1

パレットのグループをクリックします。

方法 2

1. パレットのグループを右クリックして、[ 設定 ] を選択します。  
[ パレット設定 ] ダイアログが表示されます。
2. [ ドロワー・オプション ] で次のラジオボタンを選択します。  
[ 別のドロワーを開いているときは常に閉じる ] ラジオボタン  
一つのグループのツール一覧だけを表示させたい場合に選択します。  
[ 十分なスペースがないときは自動的に閉じる ] ラジオボタン

複数のグループのツール一覧を表示しているためパレットに十分な表示スペースがない場合、自動的にツール一覧を閉じたいときに選択します。ただし、ピン留め が設定されているグループは、パレットに十分な表示スペースがない場合でもツール一覧は閉じません。

#### [ 閉じない ] ラジオボタン

グループや表示スペースの影響によって、ツール一覧が閉じないようにしたい場合に選択します。

#### 注

ピン留めを設定するには、次の二つの方法があります。

##### 方法 1

パレットのグループの右側にある  (ピン) をクリックします。

##### 方法 2

パレットのグループを右クリックして、[ ピン留め ] を選択します。

## 11.8.12 XPath を指定する

アクティビティの定義で使用する次のダイアログの入力欄には、XPath を指定する必要があります。

- [ 代入アクティビティ ] サブダイアログの [ コピー元 ] の [ 値 ]
- [ 条件設定 ] ダイアログ ( 分岐アクティビティ , 並列処理アクティビティ , 繰り返しアクティビティ ) の [ 条件式 ]

XPath の中では、拡張関数を使用して変数内の情報を取得し、利用することができます。拡張関数については、「11.8.12(3) 拡張関数」を参照してください。

### (1) 代入アクティビティサブダイアログでの指定

[ 代入アクティビティ ] サブダイアログの [ コピー元 ] の [ 値 ] に指定する XPath は、定義済みの変数がメッセージ型の場合とメッセージ型以外の場合で異なります。

#### (a) メッセージ型の変数の場合

[ 値 ] には拡張関数を指定します。指定できる拡張関数は、「11.8.12(3) 拡張関数」の (i) または (ii) の拡張関数です。

#### (i) の拡張関数を指定する場合

拡張関数の引数には、次のように指定します。

- variableName : 変数名を指定します。
- propertyName : variableName で指定した変数内で定義されている、messageType 型以外の部分名を指定します。

#### 指定例

```
bpws:getVariableProperty('VariableX','PropertyY')
```

(ii) の拡張関数を指定する場合

拡張関数の引数には、次のように指定します。

- variableName : 変数名を指定します。
- locationPath : [ 変数の表示 ] ダイアログの [ 選択パス ] で取得したパスを指定します。

指定例

```
csc:getVariableData('VariableX','input/forInvoke/depositData/id')
```

(b) メッセージ型以外の変数の場合

[ 値 ] には拡張関数を指定します。指定できる拡張関数は、「11.8.12(3) 拡張関数」の(ii)の拡張関数です。

拡張関数の引数には、変数名だけを指定します。

指定例

```
csc:getVariableData('VariableX')
```

## (2) 条件設定ダイアログでの指定

[ 条件設定 ] ダイアログの [ 条件式 ] に指定する XPath は、定義済みの変数がメッセージ型の場合とメッセージ型以外の場合で異なります。

(a) メッセージ型の変数の場合

[ 条件式 ] には拡張関数を指定します。指定できる拡張関数は、「11.8.12(3) 拡張関数」の(i)または(ii)の拡張関数です。

(i) の拡張関数を指定する場合

拡張関数の引数には、次のように指定します。

- variableName : 変数名を指定します。
- propertyName : variableName で指定した変数内で定義されている、messageType 型以外の部分名を指定します。

指定例

```
bpws:getVariableProperty('VariableX','PropertyY') = "HITACHI"
```

(ii) の拡張関数を指定する場合

拡張関数の引数には、次のように指定します。

- variableName : 変数名を指定します。
- locationPath : [ 変数の表示 ] ダイアログの [ 選択パス ] で取得したパスを指定します。

指定例

```
csc:getVariableData('VariableX','input/forInvoke/depositData/userName/firstName') = "Taro"
```

## (b) メッセージ型以外の変数の場合

[ 条件式 ] には拡張関数を指定します。指定できる拡張関数は、「11.8.12(3) 拡張関数」の(ii)の拡張関数です。

拡張関数の引数には、変数名だけを指定します。

## 指定例

```
csc:getVariableData('VariableX') = false()
```

注 VariableX が boolean 型の場合です。

## (3) 拡張関数

拡張関数には、次に示す (i) および (ii) の二つの拡張関数があります。拡張関数の形式、引数、および拡張関数を使用する場合の注意事項を説明します。

## (i) bpws:getVariableProperty

## 形式

```
bpws:getVariableProperty('variableName','propertyName')
```

## 引数

variableName

変数名を指定します。

propertyName

variableName で指定した変数内で定義されている、messageType 型以外の部分名を指定します。

## (ii) csc:getVariableData

## 形式

```
csc:getVariableData('variableName','locationPath')
```

## 引数

variableName

変数名を指定します。

locationPath

XML スキーマのノードを示すパスを指定します。

## 注意事項

- 変数がメッセージ型以外の場合、引数には variableName だけを指定してください。
- 変数がメッセージ型以外の場合、関数は変数の型に応じたデータ型を返します。
- variableName で指定した変数のデータに対して、locationPath で指定したパス

## 11. 開発環境で使用する画面

を評価した結果は、文字列型 (java.lang.String ヘマッピング) で返します。

- locationPath に「'」を含む場合、locationPath を「"」で囲んでください。

## 11.9 データ変換（マッピング）定義画面の操作

---

データ変換（マッピング）定義画面での操作について説明します。

### 11.9.1 操作を元に戻す

データ変換定義画面で行った操作を元に戻します。操作は何回でも元に戻すことができます。また、保存したあとでも操作を元に戻すことができます。

操作を元に戻すには、次の二つの方法があります。

#### 方法 1

データ変換定義画面上で右クリックして、[元に戻す]を選択します。

#### 方法 2

[Ctrl] キー + [z] キーを押します。

### 11.9.2 操作をやり直す

データ変換定義画面で行った操作をやり直します。操作は何回でもやり直すことができます。また、保存したあとでも操作をやり直すことができます。

操作をやり直すには、次の二つの方法があります。

#### 方法 1

データ変換定義画面上で右クリックして、[やり直し]を選択します。

#### 方法 2

[Ctrl] キー + [y] キーを押します。

### 11.9.3 定義を保存する

データ変換定義を保存する方法を次に示します。

#### 方法

Eclipse のメニューから、[保存]を選択します。

保存したデータ変換定義は再編集できます。再編集する場合は、アダプタ定義画面またはビジネスプロセス定義画面のデータ変換アクティビティから、データ変換定義画面を表示して再編集してください。

### 11.9.4 マッピング線の色を変更する

マッピング線の色を変更する方法を次に示します。

### 方法

1. Eclipse のメニューから、[ ウィンドウ ] - [ 設定 ] を選択します。  
[ 設定 ] ダイアログが表示されます。
2. ダイアログ左側から [ HCSC-Definer ] - [ データ変換 ] を選択します。  
データ変換定義画面の設定項目がダイアログ右側に表示されます。データ変換定義画面の設定項目については、「11.5.8 設定ダイアログ」を参照してください。
3. マッピング線の色を設定します。
  - 代入線の色を設定する場合  
[ 代入 ] の右側にあるボタンをクリックすると、各 OS に対応した、色を選択するダイアログが表示されます。表示されたダイアログで、設定したい色を選択します。
  - くり返し対応線の色を設定する場合  
[ くり返し ] の右側にあるボタンをクリックすると、各 OS に対応した、色を選択するダイアログが表示されます。表示されたダイアログで、設定したい色を選択します。
  - 条件線の色を設定する場合  
[ 条件 ] の右側にあるボタンをクリックすると、各 OS に対応した、色を選択するダイアログが表示されます。表示されたダイアログで、設定したい色を選択します。
4. [ 適用 ] ボタンをクリックします。
5. [ OK ] ボタンをクリックします。

# 12 コマンド

この章では、Cosminexus サービスプラットフォームのシステム開発に利用するコマンドについて説明します。

---

コマンドの入力形式

---

コマンドの文法の記述形式

---

## コマンドの入力形式

---

コマンドの入力形式を次に示します。

コマンド名称 [ オプション... ]

各項目について説明します。なお、コマンドプロンプトを「\$」、コマンド名称を「cmd」と表記します。

### コマンド名称

実行するコマンドのファイル名を指定します。

空白を含むパスを指定してコマンドを実行する場合、パス全体を "" で囲む必要があります。

- 誤った指定例：\$ C:\Program Files\Hitachi\Cosminexus\CC\server\bin\cmd
- 正しい指定例：\$ "C:\Program Files\Hitachi\Cosminexus\CC\server\bin\cmd"

### 引数

引数には、オプションも含まれます。オプションの入力形式および指定規則を次に示します。

#### オプションの入力形式

オプションは、「-」(ハイフン)で始まる文字列です。オプションの入力形式には、オプション引数を指定しない形式と、1個のオプション引数を指定する形式があります。

#### オプション引数を指定しない形式

\$ cmd - オプションフラグ

#### 1個のオプション引数を指定する形式

\$ cmd - オプションフラグ <スペースまたはタブ> オプション引数

#### (凡例)

- オプションフラグ  
1文字の半角英数字です。大文字と小文字が区別されます。
- オプション引数  
オプションフラグに対する引数です。

#### オプションの指定規則

- オプションフラグは、一つの「-」(ハイフン)にまとめて指定できません。  
誤った指定例：\$ cmd -abc  
正しい指定例：\$ cmd -a -b -c
- オプション引数を必要とするオプションフラグのオプション引数は、省略できません。
- オプションフラグとオプション引数の間には、スペースまたはタブが必要です。

誤った指定例：`$ cmd -afile`

正しい指定例：`$ cmd -a file`

- 同じオプションフラグは、複数指定できません。

誤った指定例：`$ cmd -a 1 -a 2`

- オプション引数に空白を含む場合、オプション引数全体を `"` で囲む必要があります。

誤った指定例：`$ cmd -a file 1`

正しい指定例：`$ cmd -a "file 1"`

## コマンドの文法の記述形式

---

コマンドの文法について次の形式で説明します。

### 形式

コマンドの入力形式を示します。

### 機能

コマンドの機能について説明します。

### 引数

コマンドの引数およびオプションについて説明します。

### 戻り値

コマンドの戻り値について説明します。

# csamkxmls (DB アダプタ用の XML フォーマット定義ファイルの生成)

## 形式

```
csamkxmls -o <SQLオペレーション定義ファイルの名称>
           -x <DBアダプタ用のXMLフォーマット定義ファイルの名称>
           [-f]
```

## 機能

DB アダプタのサービス部品を実行するための DB アダプタ用の XML フォーマット定義ファイルを作成します。

## 引数

-o <SQL オペレーション定義ファイルの名称> ~ パス名

SQL 実行形式を記載している SQL オペレーション定義ファイルの名称を指定します。SQL オペレーション定義ファイルの名称は、形式が「csa\_sql<任意の名称>.xml」のファイルを絶対パスで指定します。

-x <DB アダプタ用の XML フォーマット定義ファイルの名称> ~ パス名

作成する DB アダプタ用の XML フォーマット定義ファイルの名称を指定します。DB アダプタ用の XML フォーマット定義ファイルの名称は、拡張子が「.xsd」のファイルを絶対パスで指定します。なお、XML フォーマット定義ファイルの名称は、200 バイト 以内で指定してください。

## 注

XML フォーマット定義ファイルの絶対パスを UTF-8 で変換したときのバイト数です。

-f

作成する DB アダプタ用の XML フォーマット定義ファイルと同じ名称の XML フォーマット定義ファイルがすでにある場合、上書きをします。作成する DB アダプタ用の XML フォーマット定義ファイルと同じ名称の XML フォーマット定義ファイルがすでにある場合に、このオプションを指定していないと、エラーになります。

## 戻り値

戻り値	意味
0	正常終了しました。
1	異常終了しました。



# 13 メッセージ

この章では、Cosminexus サービスプラットフォームの開発環境で出力されるメッセージについて説明します。

---

13.1 メッセージの種類と形式

---

13.2 KECT02100 ~ KECT12999 のメッセージ

---

13.3 KECT13100 ~ KECT24999 のメッセージ

---

13.4 KECK72001 ~ KECK74999 のメッセージ

---

13.5 エラーメッセージダイアログが出力された場合の注意事項

---

## 13.1 メッセージの種類と形式

---

### 13.1.1 メッセージの種類

Cosminexus サービスプラットフォームの開発環境で出力されるメッセージには、パイナリフォーマット定義に関するメッセージ、データ変換に関するメッセージ、およびメッセージダイアログによるメッセージがあります。

### 13.1.2 メッセージの形式

このマニュアルでのメッセージの記述形式について説明します。

このマニュアルでのメッセージの記述形式を次に示します。

#### **XXXXnnnnn-Y**

---

メッセージテキスト

メッセージの説明

(S)

Cosminexus サービスプラットフォームの動作

(O)

ユーザの処置

なお、「(S)」および「(O)」はメッセージによって記述しないものもあります。

XXXX

メッセージを出力した環境（開発環境、実行環境、または運用環境）を示す ID（プリフィックス）を表します。プリフィックスは英字 4 文字で示します。

nnnnn

メッセージを出力したプログラムで管理するメッセージ番号を表します。それぞれのメッセージには、5 けたの固有の番号が付いています。

Y

メッセージのレベルを表します。メッセージのレベルは英字 1 文字で示します。メッセージのレベルを示す文字とその意味を次に示します。

E (Error)

エラーレベルのトラブルが発生したことを通知するメッセージです。  
このメッセージが出力されたときは、処理を中断します。

W (Warning)

警告レベルのトラブルが発生したことを通知するメッセージです。

メッセージが出力されたあとも処理を続行します。

I (Information)

システムの動作を通知するメッセージです。

メッセージが出力されたあとも処理を続行します。

メッセージテキスト

Cosminexus サービスプラットフォームが出力したメッセージテキストを示します。

メッセージの説明

メッセージテキストに対する補足説明を示します。

(S)

Cosminexus サービスプラットフォームの動作を示します。

(O)

ユーザの処置を示します。なお、「保守員に連絡してください」とは、購入時の契約に基づいて、システム管理者が弊社問い合わせ窓口へ連絡することを示します。

## 13.2 KECT02100 ~ KECT12999 のメッセージ

---

### KECT02100-I

---

データ変換定義ファイルを生成しました。

データ変換定義ファイルを生成しました。

(S)

データ変換定義ファイルを出力します。

### KECT02101-E

---

マッピング定義ファイルの検証でエラーが発生しました。原因 = エラーの原因となった例外情報

マッピング定義ファイルの検証中に検証エラーが発生しました。

(S)

マッピング定義ファイルの検証後、データ変換定義ファイルのビルドを中止します。

(O)

エラーの原因となった例外情報を参照し、その対処に従ってください。

### KECT02102-E

---

マッピング定義ファイルの検証で致命的なエラーが発生しました。原因 = エラーの原因となった例外情報

マッピング定義ファイルの検証中に継続できない検証エラーが発生しました。

(S)

マッピング定義ファイルの検証を中止し、ビルドを中止します。

(O)

エラーの原因となった例外情報を参照し、その対処に従ってください。

### KECT02103-E

---

マッピング定義ファイルのロードに失敗しました。原因 = エラーの原因となった例外情報

マッピング定義ファイルのロード中にエラーが発生しました。

(S)

データ変換定義ファイルのビルドを中止します。

(O)

エラーの原因となった例外情報を参照し、その対処に従ってください。

**KECT02104-E**

---

データ変換定義ファイルの生成に失敗しました。原因 = エラーの原因となった例外情報  
データ変換定義ファイル生成中にエラーが発生しました。

(S)

データ変換定義ファイルのビルドを中止します。

(O)

エラーの原因となった例外情報を参照し、その対処に従ってください。

**KECT02105-E**

---

予期せぬエラーが発生しました。原因 = エラーの原因となった例外情報  
予期せぬエラーが発生しました。

(S)

データ変換定義ファイルのビルドを中止します。

(O)

保守員に連絡してください。

**KECT02106-E**

---

リポジトリが不正です。詳細 = エラーの詳細情報  
不正なりポジトリを指定しました。

(S)

エラーメッセージを表示して処理を中止します。

(O)

エラーの詳細情報を参照し、そのメッセージの対処に従ってください。

**KECT05001-E**

---

内部エラーが発生しました。詳細 = エラーの詳細情報  
内部エラーが発生しました。

(S)

エラーメッセージを表示して処理を中止します。

(O)

保守員に連絡してください。

**KECT05002-E**

---

ファイルの読み込みに失敗しました。原因 = エラーの原因となった例外情報  
ファイルが読み込めません。

(S)

エラーメッセージを表示して処理を中止します。

### 13. メッセージ

(O)

保守員に連絡してください。

#### **KECT05003-E**

---

ファイルの読み込みに失敗しました。path= ファイルパス  
ファイルが読み込めません。

(S)

エラーメッセージを表示して処理を中止します。

(O)

保守員に連絡してください。

#### **KECT05004-E**

---

ファイルの書き込みに失敗しました。  
ファイルが書き込めません。

(S)

エラーメッセージを表示して処理を中止します。

(O)

保守員に連絡してください。

#### **KECT05005-E**

---

予期せぬエラーが発生しました。原因 = エラーの原因となった例外情報  
予期せぬエラーが発生しました。

(S)

エラーメッセージを表示して処理を中止します。

(O)

保守員に連絡してください。

#### **KECT10001-E**

---

Schema 検証でエラーが発生しました。論理名 =XML Schema 論理名, 詳細情報 = 検証エラーの  
内容

XML Schema 検証でエラーが発生しました。

(S)

エラーメッセージを表示して処理を中止します。

(O)

指定した XML Schema を見直してください。

**KECT10002-E**

---

targetNamespace に不正な URI , または修飾名が使用されています。論理名 =XML Schema 論理名

変換先 XML Schema の targetNamespace 接頭辞が x から始まっているか , targetNamespace に xsl または xsi の URI が指定されています。

(S)

エラーメッセージを表示して処理を中止します。

(O)

指定した XML Schema を見直してください。

**KECT10003-E**

---

redefine は使用できません。論理名 =XML Schema 論理名

XML Schema に未サポート項目である redefine が定義されています。

(S)

エラーメッセージを表示して処理を中止します。

(O)

指定した XML Schema を見直してください。

**KECT10004-E**

---

XML Schema 解析でエラーが発生しました。論理名 =XML Schema 論理名 , 詳細情報 = エラーの内容

XML Schema 解析でエラーが発生しました。

(S)

エラーメッセージを表示して処理を中止します。

(O)

指定した XML Schema を見直してください。

**KECT10005-E**

---

XML Schema 解析で内部エラーが発生しました。論理名 =XML Schema 論理名 , 詳細情報 = 詳細情報

内部エラーが発生しました。

(S)

エラーメッセージを表示して処理を中止します。

(O)

保守員に連絡してください。

### KECT10006-E

---

有効なルート要素候補がありません。論理名 =XML Schema 論理名  
XML Schema に大域定義 element がありません。

(S)

エラーメッセージを表示して処理を中止します。

(O)

指定した XML Schema を見直してください。

### KECT10007-E

---

ツリー中のノード数が最大値ノード数の最大値を超えました。論理名 =XML Schema 論理名  
ルートノード以下のノード数が最大値を超えました。

(S)

エラーメッセージを表示して処理を中止します。

(O)

指定した XML Schema を見直してください。

### KECT10008-E

---

未検証のバイナリフォーマット定義が指定されました。論理名 =XML Schema 論理名  
検証されていないバイナリフォーマット定義が指定されました。

(S)

エラーメッセージを表示して処理を中止します。

(O)

バイナリフォーマット定義を検証後、再実行してください。

### KECT10009-E

---

バイナリフォーマット定義が不正です。論理名 =XML Schema 論理名, 詳細情報 = 詳細情報  
不正なバイナリフォーマット定義が指定されました。

(S)

エラーメッセージを表示して処理を中止します。

(O)

バイナリフォーマット定義が壊れているおそれがあります。再度、検証済みのバイナリフォーマット定義を指定しても同じエラーになる場合は、保守員に連絡してください。

### KECT11000-E

---

コマンドクラスのインスタンス生成に失敗しました。詳細 = エラーが発生した原因  
コマンドクラス (内部クラス) の生成に失敗しました。

- (S) エラーメッセージを表示して処理を中止します。
- (O) 保守員に連絡してください。

### **KECT11001-E**

---

コマンド検索クラスに対応するコマンドクラスが見つかりません。コマンド検索クラス名 = コマンド検索クラス名

- コマンド検索に失敗しました。
- (S) エラーメッセージを表示して処理を中止します。
  - (O) 保守員に連絡してください。

### **KECT11002-E**

---

最少出現回数が 1 回以上の any 要素または anyattribute 属性を含む変換先スキーマへのデータ変換定義は生成できません。詳細 = any 要素または anyattribute 属性のノードパス

- 最少出現回数が 1 回以上の any 要素または anyattribute 属性を持つスキーマを変換先スキーマに指定しました。
- (S) エラーメッセージを表示して処理を中止します。
  - (O) 指定のスキーマを変換先スキーマに指定してデータを変換することはできません。スキーマファイルを指定し直してください。

### **KECT11003-E**

---

use 属性に required が指定されている属性ノードに対して代入線が定義されていません。詳細情報 = 属性ノードパス

- 必須属性に対してマッピングが定義されていません。
- (S) エラーメッセージを表示して処理を中止します。
  - (O) 必須属性に対してマッピングを定義してください。

### **KECT11004-E**

---

複合型要素ノードに対して代入線が定義されています。詳細 = 要素ノードパス  
複合型要素に対して代入マッピングが定義されています。

### 13. メッセージ

(S)

エラーメッセージを表示して処理を中止します。

(O)

複合型要素に対して定義されている代入マッピングを削除してください。

#### **KECT11005-E**

---

選択ファンクションに対して選択ファンクションが接続されています。詳細 = 選択ファンクション名

選択ファンクションに対して選択ファンクションが接続されています。

(S)

エラーメッセージを表示して処理を中止します。

(O)

選択ファンクションのネストはできません。選択ファンクションに接続されている選択ファンクションを削除してください。

#### **KECT11006-E**

---

最少出現回数が 2 回以上の要素に対してマッピングが定義されていません。詳細 = 要素ノードパス

最少出現回数が 2 回以上の要素に対してマッピングが定義されていません。

(S)

エラーメッセージを表示して処理を中止します。

(O)

くり返しファンクションを使用してエラーとなった要素にマッピングを定義してください。

#### **KECT11007-E**

---

最少出現回数が 1 回以上の単純内容要素に対してマッピングが定義されていません。詳細 = 要素ノードパス

最少出現回数が 1 回以上の単純内容要素に対してマッピングが定義されていません。

(S)

エラーメッセージを表示して処理を中止します。

(O)

エラーとなった単純内容要素にマッピングを定義してください。

#### **KECT11008-E**

---

XSLT ノードの追加に失敗しました。追加ノード = 追加した XSLT 要素名, 追加先ノード = 追加先の XSLT 要素名

XSLT 生成時に不正なノード処理をしました。

- (S) エラーメッセージを表示して処理を中止します。
- (O) 保守員に連絡してください。

**KECT11009-E**

---

XSL ツリーの構築時にエラーが発生しました。詳細 = エラーの詳細情報  
XSL ツリーの構築時にエラーが発生しました。

- (S) エラーメッセージを表示して処理を中止します。
- (O) エラーの詳細情報を参照し、そのメッセージの対処に従ってください。

**KECT11010-E**

---

内部エラーが発生しました。詳細 = エラーの詳細情報  
生成処理中に内部エラーが発生しました。

- (S) エラーメッセージを表示して処理を中止します。
- (O) 保守員に連絡してください。

**KECT12000-E**

---

コマンドクラスのインスタンス生成に失敗しました。詳細 = エラーが発生した原因  
コマンドクラス (内部クラス) の生成に失敗しました。

- (S) エラーメッセージを表示して処理を中止します。
- (O) 保守員に連絡してください。

**KECT12001-E**

---

コマンド検索クラスに対応するコマンドクラスが見つかりません。コマンド検索クラス名 = コマンド検索クラス名  
コマンド検索に失敗しました。

- (S) エラーメッセージを表示して処理を中止します。
- (O) 保守員に連絡してください。

### **KECT12002-E**

---

入力に接続されているマッピング線の数本のマッピング線を接続することはできません。名前 = ノードの名前, 接続可能な数 = 接続可能なマッピング線の数

入力側に接続されているマッピング線の数と接続できる線の数一致しません。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

### **KECT12003-E**

---

出力に接続されているマッピング線の数本のマッピング線を接続することはできません。名前 = ノードの名前, 接続可能な数 = 接続可能なマッピング線の数

出力側に接続されているマッピング線の数と接続できる線の数一致しません。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

### **KECT12004-E**

---

接続されているノードの名前を入力ノードに指定することはできません。名前 = ノードの名前  
入力側に接続できないノードが接続されています。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

### **KECT12005-E**

---

接続されているノードの名前を出力ノードに指定することはできません。名前 = ノードの名前  
出力側に接続できないノードが接続されています。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

### **KECT12006-E**

---

必須属性ノードに対してマッピングが定義されていません。名前 = ノードの名前, 依存対象 = 依存先のくり返しファクションの名前

必須属性ノードにマッピング線が定義されていません。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

#### **KECT12007-E**

---

選択条件が定義されていません。名前 = ノードの名前

選択ファンクションにマッピング線は定義されていますが、対応する条件が設定されていません。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

#### **KECT12008-E**

---

必須ノードに対してノード出力なしを定義することはできません。名前 = ノードの名前

最少出現回数が1回以上のノードに対してノード出力なしを定義しました。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

#### **KECT12009-E**

---

必須要素ノードに対してマッピングが定義されていません。名前 = ノードの名前, 依存対象 = 依存先のくり返しファンクションの名前

必須要素ノードにマッピング線が定義されていません。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

#### **KECT12010-E**

---

同じ依存対象を持つ代入線を複数定義することはできません。名前 = ノードの名前, 依存対象 = 依存先のくり返しファンクションの名前

同じ変換先スキーマノードに対して、依存対象が等しい代入線が複数定義されています。

### 13. メッセージ

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

同じ変換先スキーマノードに対して定義できる代入線は、依存対象ごとに一つまでです。

複数定義されている代入線を削除してください。

#### **KECT12011-E**

---

依存対象が等しいくり返し対応線と代入線を同時に定義することはできません。名前 = ノードの名前, 依存対象 = 依存先のくり返し関クションの名前

同じ変換先スキーマノードに対して、依存対象が等しいくり返し対応線と代入線が同時に定義されています。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

同じ変換先スキーマノードに対して、同じ依存対象を持つくり返し対応線と代入線を同時に定義することはできません。

くり返し対応線か代入線のどちらかを削除してください。

#### **KECT12012-E**

---

同じ変換先スキーマノードに対して、くり返し関クションをネストすることはできません。名前 = ノードの名前

同じ変換先スキーマノードに対して、くり返し定義をネストして定義しています。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

同じ変換先スキーマノードに対して、くり返し変換をネストすることはできません。ネストされているくり返し対応線を削除してください。

#### **KECT12013-E**

---

内部エラーが発生しました。詳細 = エラーの詳細情報

検証処理中に内部エラーが発生しました。

(S)

エラーメッセージを表示して、検証処理を中止します。

(O)

保守員に連絡してください。

**KECT12014-E**

---

マッピング定義の変換元と変換先の出現回数が一致しません。名前 = ノードの名前, 依存対象 = 依存先のくり返しファンクションの名前

マッピングを定義している変換元と変換先の出現回数が一致しません。

(S)

エラーメッセージを表示して, 検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

**KECT12015-E**

---

マッピング不可能なノードが必須ノードとして定義されています。名前 = ノードの名前  
変換先スキーマにマッピングできないノードが必須ノードとして定義されています。

(S)

エラーメッセージを表示して, 検証処理を続けます。

(O)

該当ノードの祖先ノードに最少出現回数が0回のノードがある場合は, そのノード以下を出力しないようにマッピングを定義してください。

**KECT12016-E**

---

入力のマッピング線が存在しません。名前 = ノードの名前  
入力用のマッピング線が接続されていません。

(S)

エラーメッセージを表示して, 検証処理を続けます。

(O)

該当ノードに入力用のマッピング線を定義してください。

**KECT12017-E**

---

複合内容要素に対して値を出力することはできません。名前 = ノードの名前  
複合型要素に対して代入マッピングが定義されていません。

(S)

エラーメッセージを表示して, 検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

**KECT12018-E**

---

子ノードに必須要素を持つ要素に対して空ノードを出力することはできません。名前 = ノードの名前

### 13. メッセージ

最少出現回数が0でない要素を子供に持つ要素に対して、空要素出力を定義しています。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

#### **KECT12019-E**

---

choice 構造要素のすべての子孫ノードに対してマッピングが定義されていません。名前 = ノードの名前, 依存対象 = 依存先のくり返しファクションの名前

choice 構造要素のすべての子要素が必須であるにもかかわらず、どの子要素の子孫に対してもマッピングが定義されていません。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

#### **KECT12020-E**

---

すべての子ノードが必須要素である choice 構造要素に対して空ノードを出力することはできません。名前 = ノードの名前

choice 構造要素のすべての子要素が必須であるにもかかわらず、空要素出力を定義しています。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

#### **KECT12021-E**

---

依存対象を持たないマッピング線が存在しました。名前 = ノードの名前, 接続元 = 接続元ノードの名前

依存対象に関連するマッピングを検証中に、依存対象を持たないマッピング線が見付かりました。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

変換元ノードまで接続されていないマッピング線が存在するおそれがあります。すべてのマッピング線が正しく変換元ノードまで接続されているかどうか確認してください。

**KECT12022-E**

---

複合内容要素でないノードに対してマッピングする値が設定されていません。名前 = ノードの名前

複合内容要素でない変換先ノードへ接続している選択ファンクションに出力する値が設定されていない条件が存在します。

(S)

エラーメッセージを表示して、検証処理を続けます。

(O)

該当ノードに接続されているマッピング線を見直してください。

**KECT12023-E**

---

マッピングの数が多いため、データ変換定義を生成することができません。

データ変換定義ファイルのサイズが限界を超えました。

(S)

メッセージを表示して、検証処理を終了します。

(O)

定義しているマッピングの数を少なくしてください。

マッピングの数を減らしてもエラーが発生する場合は、保守員に連絡してください。

**KECT12024-E**

---

ルート要素が見つかりません。名前 = 変換先スキーマの論理名  
変換先スキーマのルート要素が見つかりません。

(S)

メッセージを表示して、検証処理を終了します。

(O)

ルート要素選択ができていないか見直してください。

**KECT12100-I**

---

マッピング定義の検証に成功しました。

検証処理に成功しました。

(S)

メッセージを表示して、検証処理を終了します。

**KECT12101-I**

---

エラー：エラーの個数個、警告：警告の個数個、情報：情報の個数個  
検証で発生したメッセージの個数です。

(S)

メッセージを表示して、検証処理を終了します。

### KECT12500-W

---

マッピング定義の変換元と変換先の出現回数が一致しない可能性があります。名前 = ノードの名前, 依存対象 = 依存先のくり返しファクションの名前

マッピングを定義している変換元と変換先の出現回数が一致しないおそれがあります。

(S)

警告メッセージを表示して、検証処理を続けます。

(O)

マッピングの変換元と変換先の出現回数が一致しないおそれがあるため、変換実行時に検証エラーになるおそれがあります。

変換する XML データ内の該当ノードの出現回数を確認し、実行時に検証エラーにならないことを確認してください。

### KECT12501-W

---

choice 構造要素の複数の子孫ノードに対して同時にマッピングが定義されています。実行時にエラーになる可能性があります。名前 = ノードの名前, 依存対象 = 依存先のくり返しファクションの名前

choice 構造要素の複数の子要素または子孫ノードに対してマッピングが定義されています。

(S)

警告メッセージを表示して、検証処理を続けます。

(O)

choice 構造要素の複数の子要素が同時に出現し、変換実行時に検証エラーになるおそれがあります。

該当する choice 構造以下のマッピングを見直し、同時に複数の子要素が出現しないことを確認してください。

### KECT12502-W

---

choice 構造要素の子ノードに対して選択ファクションが定義されています。名前 = ノードの名前, 依存対象 = 依存先のくり返しファクションの名前

choice 構造要素のすべての子要素が必須ですが、選択ファクションが定義されています。

(S)

警告メッセージを表示して、検証処理を続けます。

(O)

選択ファクションのノード出力なしを定義している場合、変換実行時に検証エラーになるおそれがあります。

選択ファクションのノード出力なしを定義していないことを確認してください。

**KECT12503-E**

---

変換テーブル ID が指定されていません。

変換テーブル ID が指定されていません。

**(S)**

エラーメッセージを表示して、検証処理を続けます。

**(O)**

変換テーブル ID を指定してください。

## 13.3 KECT13100 ~ KECT24999 のメッセージ

---

### KECT13100-E

---

予期せぬエラーが発生しました。原因 = エラーの原因となった例外情報  
予期しないエラーが発生しました。

- (S) エラーメッセージを表示して処理を中止します。
- (O) 保守員に連絡してください。

### KECT13101-E

---

別の構成要素から参照されているため削除できません。  
削除対象の要素を別の構成要素が参照しています。

- (S) 削除処理を中止します。
- (O) 参照している構成要素をすべて削除してから対象の要素を削除してください。

### KECT13102-E

---

ルート要素は削除できません。  
削除対象の要素はルート要素です。

- (S) 削除処理を中止します。
- (O) ルート要素を変更してから削除してください。

### KECT13103-W

---

出現回数ノードまたはサイズノードまたは選択条件ノードとして使用されていますが削除してもよろしいですか？

削除対象の要素が出現回数ノード、サイズノードまたは選択条件ノードとして使用されています。

- (S) 「OK」がクリックされた場合は、削除処理を続けます。「キャンセル」がクリックされた場合は、削除処理を中止します。

(O)

そのまま処理を続けるか、または、出現回数ノード、サイズノードもしくは選択条件ノードとして使用している要素の設定を変更してから削除してください。

### **KECT17100-E**

---

再帰構造になるため要素を追加できません。

指定した構成要素を追加すると、構成が再帰構造になるため追加できません。

(S)

ドラッグ&ドロップ処理を中止します。

(O)

再帰構造にならない要素を指定してください。

### **KECT18100-E**

---

予期せぬエラーが発生しました。原因 = エラーの原因となった例外情報

予期しないエラーが発生しました。

(S)

エラーメッセージを表示して処理を中止します。

(O)

保守員に連絡してください。

### **KECT18101-E**

---

ファイルの読み込みに失敗しました。原因 = エラーの原因となった例外情報

ファイルが読み込めません。

(S)

エラーメッセージを表示して処理を中止します。

(O)

エラーの原因となった例外情報を参照し、その対処に従ってください。

### **KECT18102-E**

---

ファイルの書き込みに失敗しました。原因 = エラーの原因となった例外情報

ファイルが書き込めません。

(S)

エラーメッセージを表示して処理を中止します。

(O)

エラーの原因となった例外情報を参照し、その対処に従ってください。

### **KECT18103-E**

---

保管は完了しませんでした。保管先のファイルが既に別のエディタで開かれています。  
保管先のファイルが別のエディタで開かれています。

(S)

エラーメッセージを表示して処理を中止します。

(O)

保管先ファイルのエディタを閉じるか別のファイル名で保管してください。

### **KECT21100-E**

---

バイナリフォーマット定義生成で内部エラーが発生しました。詳細 = エラーの詳細情報  
内部エラーが発生しています。

(S)

エラーメッセージを表示して処理を中止します。

(O)

保守員に連絡してください。

### **KECT24000-E**

---

フォーマットには逐次構成の複合内容を定義しなければなりません。  
フォーマットに逐次構成の複合内容が定義されていません。

(S)

メッセージを表示して、検証処理を続けます。

(O)

フォーマットに逐次構成の複合内容を定義してください。

### **KECT24001-E**

---

ルート要素が指定されていません。  
ルート要素に指定されている複合内容が存在しません。

(S)

メッセージを表示して、検証処理を続けます。

(O)

フォーマット内のどれかの複合内容をルート要素に設定してください。

### **KECT24002-E**

---

文字コードが UTF-16 の場合は、サイズに偶数の値を指定しなければなりません。  
フォーマットの文字コードが UTF-16 にもかかわらず、サイズに奇数バイトを指定して  
います。

- (S) メッセージを表示して、検証処理を続けます。
- (O) フォーマットの文字コードに UTF-16 以外を指定するか、該当する単純内容のサイズに偶数バイトを指定してください。

#### **KECT24003-E**

---

サイズには小数部桁数のバイトサイズ以上の値を指定しなければなりません。  
データ種別がゾーン形式数値、パック形式数値の単純内容で、小数部桁数のバイト数よりも小さい値をサイズに指定しています。

- (S) メッセージを表示して、検証処理を続けます。
- (O) サイズに小数部桁数のバイト数以上の値を指定してください。

#### **KECT24004-E**

---

サイズには日付時刻の形式と秒小数部桁数を合わせたバイトサイズ以上の値を指定しなければなりません。

日付時刻形式のバイト数よりも小さい値をサイズに指定しています。

- (S) メッセージを表示して、検証処理を続けます。
- (O) サイズに日付時刻形式、小数点および秒小数部桁数を合わせたサイズ以上のバイト数を指定してください。

#### **KECT24005-E**

---

ルート要素から生成されるツリーのノード数が最大値を超えています。  
ルート要素からのツリーのノード数が 1,024 個を超えています。

- (S) メッセージを表示して、検証処理を続けます。
- (O) ルート要素からたどれるツリー上のノードを削除し、最大値以内になるようにしてください。

#### **KECT24006-E**

---

複合内容 複合内容の名前は構成要素が定義されていません。  
複合内容に構成要素が定義されていません。

### 13. メッセージ

(S)

メッセージを表示して、検証処理を続けます。

(O)

該当する複合内容に構成要素を追加してください。

#### **KECT24007-E**

---

複合内容 複合内容の名前は選択条件ノードが指定されていません。

選択構成の複合内容に選択条件ノードが指定されていません。

(S)

メッセージを表示して、検証処理を続けます。

(O)

該当する複合内容に選択条件ノードを指定してください。

#### **KECT24008-E**

---

選択条件ノードは単純内容でなければなりません。

選択条件ノードに単純内容以外のノードが指定されています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

選択条件ノードに単純内容を指定してください。

#### **KECT24009-E**

---

選択条件ノードとその祖先は出現回数が1回のノードでなければなりません。

選択条件ノードまたはその祖先に、出現回数が1回に固定されていないノードが指定されています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

選択条件ノードおよびその祖先は、出現回数が1回に固定されているノードになるように定義してください。

#### **KECT24010-E**

---

選択条件ノードに指定したパスが正しくありません。詳細情報

選択条件ノードに指定したパスが間違っています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

選択条件ノードに指定したパスが壊れているおそれがあります。詳細情報を参考に

選択条件ノードを指定し直してください。

### **KECT24011-E**

---

**構成要素** 構成要素の名前 に対する選択条件値が正しくありません。詳細情報

選択条件値に設定した値が選択条件ノードのデータ型と一致していません。

(S)

メッセージを表示して、検証処理を続けます。

(O)

該当する構成要素の選択条件値に、選択条件ノードのデータ型に合った値を指定してください。

### **KECT24012-E**

---

選択条件値が設定されていない構成要素は一つ以下でなければなりません。

選択条件値を設定していない構成要素が二つ以上存在しています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

選択条件値が設定されていない構成要素の定義を一つ以下にしてください。

### **KECT24013-E**

---

出現回数がデータの終端までの構成要素は、フォーマットの末尾に定義しなければなりません。

出現回数がデータの終端までに指定されている構成要素が、フォーマットの末尾以外に定義されています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

該当する構成要素をフォーマットの末尾に移動してください。

### **KECT24014-E**

---

**構成要素** 構成要素の名前 は出現回数ノードが指定されていません。

出現回数が出現回数ノードに指定されている構成要素に、出現回数ノードが定義されていません。

(S)

メッセージを表示して、検証処理を続けます。

(O)

該当する構成要素に出現回数ノードを指定してください。

### KECT24015-E

---

出現回数ノードには数値型の単純内容を指定しなければなりません。

出現回数ノードに数値型の単純内容以外のノードが指定されています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

出現回数ノードに数値型の単純内容を指定してください。

### KECT24016-E

---

出現回数ノードとその祖先には出現回数が1回のノードを指定しなければなりません。

出現回数ノードまたはその祖先に、出現回数が1回に固定されていないノードが定義されています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

出現回数ノードおよびその祖先は、出現回数が1回に固定されているノードになるように定義してください。

### KECT24017-E

---

出現回数ノードに指定したパスが正しくありません。詳細情報

出現回数ノードに指定したパスが間違っています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

出現回数ノードに指定したパスが壊れているおそれがあります。詳細情報を参考に出現回数ノードを指定し直してください。

### KECT24018-E

---

サイズがデータの終端までの構成要素は、フォーマットの末尾に定義しなければなりません。

サイズがデータの終端までに指定されている構成要素が、フォーマットの末尾以外に定義されています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

該当する構成要素をフォーマットの末尾に移動してください。

### KECT24019-E

---

構成要素 構成要素の名前はサイズノードが指定されていません。

サイズノードに指定されている構成要素にサイズノードが定義されていません。

(S)

メッセージを表示して、検証処理を続けます。

(O)

該当する構成要素にサイズノードを指定してください。

#### **KECT24020-E**

---

サイズノードには数値型の単純内容を指定しなければなりません。

サイズノードに数値型の単純内容以外のノードが指定されています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

サイズノードに数値型の単純内容を指定してください。

#### **KECT24021-E**

---

サイズノードとその祖先には出現回数が1回のノードを指定しなければなりません。

サイズノードまたはその祖先に、出現回数が1回に固定されていないノードが定義されています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

サイズノードおよびその祖先は、出現回数が1回に固定されているノードになるように定義してください。

#### **KECT24022-E**

---

サイズノードに指定したパスが正しくありません。詳細情報

サイズノードに指定したパスが間違っています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

サイズノードに指定したパスが壊れているおそれがあります。詳細情報を参考にサイズノードを指定し直してください。

#### **KECT24027-E**

---

日立コード変換のライブラリが見つからないため、選択条件値を検証することができません。

日立コード変換のJARファイルが見つかりません。

(S)

メッセージを表示して、検証処理を続けます。

### 13. メッセージ

(O)

日立コード変換の JAR ファイルを正しく設定したあと、Eclipse を再起動してください。

#### **KECT24028-E**

---

サイズには符号のバイトサイズよりも大きな値を指定しなければなりません。

データ種別が整数、実数の単純内容で、符号のバイトサイズ以下の値をサイズに指定しています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

サイズに符号のバイトサイズよりも大きな値を指定してください。

#### **KECT24029-E**

---

出現回数がデータの終端までの構成要素の祖先は、出現回数が 1 回のノードでなければなりません。

出現回数がデータの終端までに指定されている構成要素の祖先に、出現回数が 1 回に固定されていないノードが定義されています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

該当する構成要素の祖先は、出現回数が 1 回に固定されているノードになるように定義してください。

#### **KECT24030-E**

---

サイズがデータの終端までの構成要素とその祖先は、出現回数が 1 回のノードでなければなりません。

サイズがデータの終端までに指定されている構成要素およびその祖先に、出現回数が 1 回に固定されていないノードが定義されています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

該当する構成要素およびその祖先の出現回数を 1 回に固定してください。

#### **KECT24031-E**

---

サイズには符号と小数部桁数を合わせたバイトサイズ以上の値を指定しなければなりません。

データ種別が固定小数部数値の単純内容で、符号と小数部桁数を合わせたバイトサイズよりも小さい値をサイズに指定しています。

- (S)           メッセージを表示して、検証処理を続けます。
- (O)           サイズに符号と小数部桁数を合わせたバイトサイズ以上の値を指定してください。

---

**KECT24032-E**

---

選択条件ノードのサイズは固定でなければなりません。  
選択条件ノードのサイズ指定が固定ではありません。

- (S)           メッセージを表示して、検証処理を続けます。
- (O)           選択条件ノードのサイズを固定にしてください。

---

**KECT24033-E**

---

必要なセパレータが指定されていません。  
データを区切るために必要なセパレータが指定されていません。

- (S)           メッセージを表示して、検証処理を続けます。
- (O)           データを区切るために必要なセパレータを設定してください。

---

**KECT24034-E**

---

選択種別で構成要素の開始セパレータを指定した場合、構成要素の中の単純内容は一つ以下でなければなりません。  
構成要素の中に単純内容要素が二つ以上存在しています。

- (S)           メッセージを表示して、検証処理を続けます。
- (O)           構成要素の中の単純内容要素を一つ以下にしてください。

---

**KECT24035-E**

---

選択種別で構成要素の開始セパレータを指定した場合、開始セパレータが設定されていない複合内容または単純内容の構成要素は一つ以下でなければなりません。  
開始セパレータが設定されていない複合内容要素および単純内容要素を合わせた構成要素が二つ以上存在しています。

- (S)           メッセージを表示して、検証処理を続けます。

### 13. メッセージ

(O)

開始セパレータが設定されていない複合内容要素の構成要素を一つ以下にしてください。または、構成要素に単純内容要素を設定しないでください。

#### **KECT24100-I**

---

フォーマット定義の検証に成功しました。

フォーマット定義の検証に成功しました。

(S)

検証処理を終了します。

#### **KECT24500-W**

---

グローバル単純内容 グローバル単純内容の名前 はどこからも参照されていません。

単純内容がほかの複合内容の構成要素として参照されていません。

(S)

メッセージを表示して、検証処理を続けます。

(O)

該当する単純内容をほかの複合内容の構成要素に設定してください。

#### **KECT24501-W**

---

グローバル複合内容 グローバル複合内容の名前 はどこからも参照されていません。

複合内容がほかの複合内容の構成要素として参照されていません。

(S)

メッセージを表示して、検証処理を続けます。

(O)

該当する複合内容をほかの複合内容の構成要素に設定してください。

#### **KECT24502-W**

---

ルート要素 ルート要素に指定された複合内容の名前 が構成要素として定義されています。

ルート要素がほかの複合内容の構成要素として参照されています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

ルート要素に対応する構成要素を複合内容の構成要素から削除してください。

#### **KECT24503-W**

---

選択条件ノードに指定したパスに、選択構成の複合内容 選択構成の複合内容の名前 が含まれています。

選択条件ノードのパスに選択構成の複合内容が含まれています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

選択条件ノードが実際のバイナリデータ中出现しないおそれがあります。実際のバイナリデータ上に、選択条件ノードに該当するデータが出現するか確認してください。

### **KECT24504-W**

---

出現回数ノードに指定したパスに、選択構成の複合内容 選択構成の複合内容の名前 が含まれています。

出現回数ノードのパスに選択構成の複合内容が含まれています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

出現回数ノードが実際のバイナリデータ中出现しないおそれがあります。実際のバイナリデータ上に、出現回数ノードに該当するデータが出現するか確認してください。

### **KECT24505-W**

---

サイズノードに指定したパスに、選択構成の複合内容 選択構成の複合内容の名前 が含まれています。

サイズノードのパスに選択構成の複合内容が含まれています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

サイズノードが実際のバイナリデータ中出现しないおそれがあります。実際のバイナリデータ上に、サイズノードに該当するデータが出現するか確認してください。

### **KECT24506-W**

---

構成要素 構成要素の名前 に対する選択条件値は重複して指定されています。

選択条件値にほかの構成要素で指定されている値を指定しています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

ほかの構成要素で指定されていない値を選択条件値に指定してください。

### **KECT24507-E**

---

日立コード変換のライブラリが見つからないため、検証することができません。

### 13. メッセージ

日立コード変換の JAR ファイルが見つかりません。

(S)

メッセージを表示して、検証処理を続けます。

(O)

日立 コード変換の JAR ファイルを正しく設定したあと、Eclipse を再起動してください。

### **KECT24508-W**

---

構成要素 構成要素の名前 に対する開始セパレータ値は重複して指定されています。

開始セパレータの値に、ほかの構成要素で指定されている値を指定しています。

(S)

メッセージを表示して、検証処理を続けます。

(O)

ほかの構成要素で指定されていない値を開始セパレータの値に指定してください。

## 13.4 KECK72001 ~ KECK74999 のメッセージ

---

### **KECK72001-E**

---

指定したアーカイブファイルは、移行できません。

指定したアーカイブファイルは、移行対象バージョンの製品からエクスポートしたファイルではありません。

(S)

処理を終了します。

(O)

移行対象バージョンの製品からエクスポートしたアーカイブファイルを指定してください。

### **KECK72499-E**

---

内部エラーが発生しました。保守員に連絡してください。

予期しない例外が発生しました。

(S)

処理を終了します。

(O)

保守員に連絡してください。

### **KECK73001-E**

---

アーカイブファイルの展開中にエラーが発生しました。

アーカイブファイルの展開中にエラーが発生しました。

(S)

処理を終了します。

(O)

HCSCTE をインストールしたディレクトリのアクセス権を確認してください。アクセス権を確認しても問題が解決しない場合は、保守員に連絡してください。

### **KECK73003-E**

---

アーカイブファイルの作成中にエラーが発生しました。

アーカイブファイルの作成中にエラーが発生しました。

(S)

処理を終了します。

## 13. メッセージ

(O)

HCSCTE をインストールしたディレクトリのアクセス権を確認してください。アクセス権を確認しても問題が解決しない場合は、保守員に連絡してください。

### **KECK73005-E**

---

HCSCTE インストールディレクトリ以下のファイルへのアクセスで入出力エラーが発生しました。

HCSCTE をインストールしたディレクトリ以下のファイルにアクセスした際に、入出力エラーが発生しました。

(S)

処理を終了します。

(O)

HCSCTE をインストールしたディレクトリのアクセス権を確認してください。アクセス権を確認しても問題が解決しない場合は、保守員に連絡してください。

### **KECK73007-E**

---

アーカイブファイル中のデータへのアクセスで、入出力エラーが発生しました。

アーカイブファイル中のデータにアクセスした際に、入出力エラーが発生しました。

(S)

処理を終了します。

(O)

アーカイブファイルが正しい手順で作成されたかどうかを確認してください。アーカイブファイルが作成された手順を確認しても問題が解決しなかった場合は、保守員に連絡してください。

### **KECK73010-E**

---

内部エラーが発生しました。保守員に連絡してください。

DTD ファイルにアクセスした際、または XML ファイルの解析中に、内部エラーが発生しました。

(S)

処理を終了します。

(O)

保守員に連絡してください。

### **KECK73499-E**

---

内部エラーが発生しました。保守員に連絡してください。

内部エラーが発生しました。

- (S)  
処理を終了します。
- (O)  
保守員に連絡してください。

### **KECK73500-I**

---

移行処理を開始します。  
移行処理を開始します。

- (S)  
処理を続行します。

### **KECK73501-I**

---

移行処理を終了します。  
移行処理を終了します。

- (S)  
処理を続行します。

### **KECK74001-W**

---

以下のサービス中のユーザ定義クラスで指定されているファイル名が禁止ファイル名です。サービス名：サービス部品名 ユーザ定義クラス：ユーザ定義クラス

サービス部品中のユーザ定義クラスで指定されているファイル名が禁止ファイル名です。

- (S)  
処理を続行します。
- (O)  
次に示すファイル名称のファイルをユーザ定義クラスから削除したあと、旧バージョンで設定した jar ファイルの名称を変更し、再度設定してください。
- csmsvcadpdef.jar
  - cscmsg\_adpejb.jar

### **KECK74002-W**

---

以下のサービス中のユーティリティクラスで指定されているファイル名が禁止ファイル名です。サービス名：サービス部品名 ユーティリティクラス：ユーティリティクラス

サービス部品中のユーティリティクラスで指定されているファイル名が禁止ファイル名です。

- (S)  
処理を続行します。
- (O)  
次に示すファイル名称のファイルをユーティリティクラスから削除したあと、旧

## 13. メッセージ

バージョンで設定した jar ファイルの名称を変更し、再度設定してください。

- csmsvcadpdef.jar
- csedba\_ejb.jar
- cscmsg\_adapter.jar

## 13.5 エラーメッセージダイアログが出力された場合の注意事項

---

エラーメッセージダイアログが出力された場合は、メッセージに従って、入力値や操作方法を見直してください。また、エラーが解決しない場合は、次の項目を見直してください。

### (1) 全体に関する内容

- プログラムの動作環境が正しいか見直してください。必要な動作環境については、「1.4.1 前提 OS」、「1.4.2 ソフトウェアの構成」および「2.1.1 前提条件」を参照してください。
- 製品のセットアップが正しく行われているかを見直してください。必要なセットアップの作業については、「2.1 開発環境を使用するためのセットアップ」を参照してください。
- Eclipse の eclipse.ini ファイルに必要なオプションが指定されているか確認してください。必要なオプションについては、「2.1.3 Eclipse の設定ファイルの確認」を参照してください。
- HCSCTE プラグインが有効になっているか確認してください。必要な設定については、「2.1.4 Eclipse への HCSCTE プラグインの組み込み」を参照してください。
- 同一マシン上で複数の HCSCTE を使用していないか確認してください。
- リリースノートの注意事項を確認してください。

### (2) リポジトリに関する内容

- インポートする ZIP ファイルが正しくエクスポートされたものか確認してください。
- リポジトリとして指定したディレクトリまたは Cosminexus のインストールディレクトリに、必要な権限が設定されているか見直してください。
- リポジトリ内のファイルやディレクトリなどをエクスプローラーなどで参照していないか確認してください。
- 複数の HCSCTE が同じリポジトリディレクトリを使用していないか確認してください。
- リポジトリが、古いバージョンの Cosminexus サービスプラットフォームで作成されたものでないか確認してください。古いバージョンで作成したリポジトリを使用している場合は、リポジトリを作成し直すか、新しいバージョンで作成したリポジトリを使用するようにリポジトリディレクトリを設定してください。

### (3) 入出力データに関する内容

- 入力または出力するファイルやフォルダに、必要な権限が設定されているか見直してください。
- 入力するデータが、次に示す制限事項に該当していないか確認してください。

## 13. メッセージ

### 制限事項

- Web サービス  
「2.5.1(3) WSDL の形式」で示した WSDL ファイルを入力する。
- SessionBean  
「2.5.2(3) SessionBean の作成条件」で示した EAR ファイルを入力する。
- 電文に使用する電文フォーマット  
「4.3.1 標準電文を作成する」および「4.3.4 サービス部品電文を作成する (MDB (WS-R または DB キュー) の場合)」で示した電文フォーマットを入力する。

# 付録

---

付録 A 下位バージョンからのリポジトリの移行

---

付録 B 上流ツールとの連携で使用する BPEL のサポート範囲

---

付録 C Cosminexus サービスプラットフォームのサンプルプログラム

---

付録 D 用語解説

---

---

## 付録 A 下位バージョンからのリポジトリの移行

旧バージョンの開発環境で使用していたリポジトリ情報を、バージョンアップ後も使用する場合はバージョンアップ方法について説明します。

なお、バージョンアップ前に使用していたリポジトリ情報をそのまま使用して、バージョンアップ後も運用する場合はバージョンアップ方法については、マニュアル「Cosminexus サービスプラットフォーム システム構築・運用ガイド」を参照してください。

### 付録 A.1 移行の対象となるバージョン

リポジトリ情報の移行の対象となる旧バージョンを次に示します。これらのバージョンのリポジトリ情報を移行して、開発環境を 07-60 にバージョンアップできます。

- uCosminexus Service Architect 07-10
- uCosminexus Service Architect 07-20
- uCosminexus Service Architect 07-50

### 付録 A.2 下位バージョンからの移行手順

次の図に示す (1) ~ (6) の手順に従って、Cosminexus サービスプラットフォーム（開発環境）を旧バージョンから 07-60 へ移行してください。

図 A-1 下位バージョンからの移行手順

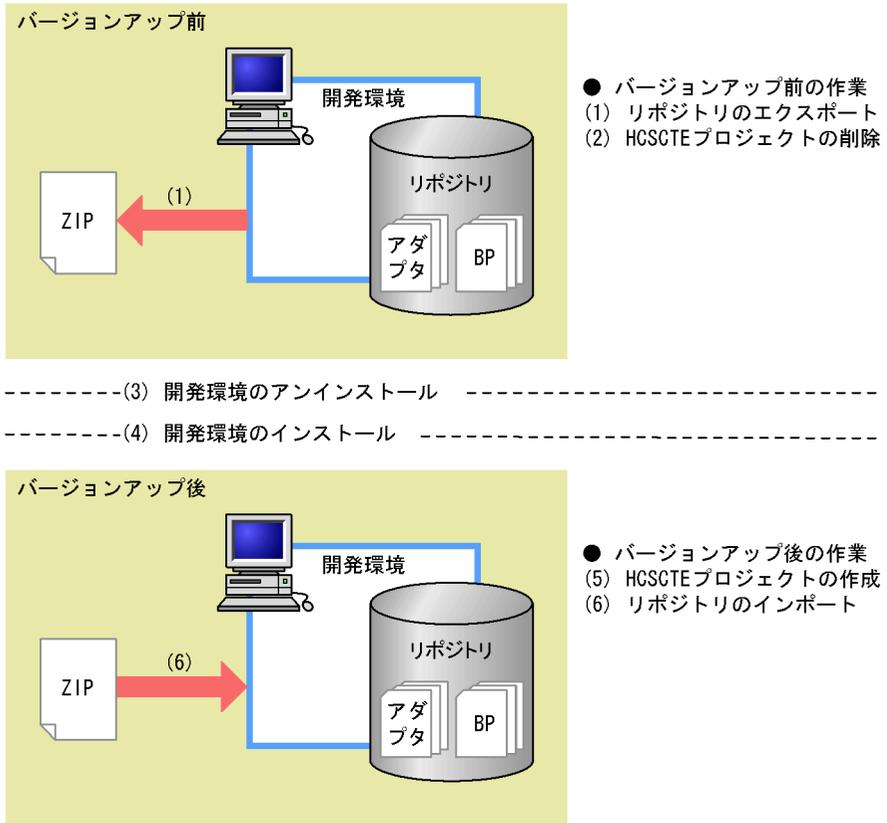


図 A-1 の (1) ~ (6) の手順の詳細を次に示します。

### (1) リポジトリのエクスポート

バージョンアップ前の開発環境のリポジトリをエクスポートして、いったん退避します。複数のリポジトリを使用している場合、必要なリポジトリをすべてエクスポートして、退避します。

リポジトリのエクスポート方法の詳細については、「3.2.2 リポジトリをエクスポートする」を参照してください。

### (2) HCSCTE プロジェクトの削除

HCSCTE プロジェクトをいったん削除します。

手順の詳細は、「3.1.4 プロジェクトを削除する」を参照してください。

### (3) 開発環境のアンインストール

開発環境で旧バージョンの Cosminexus サービスプラットフォームをアンインストール

します。Cosminexus 以下のディレクトリ情報を退避したあとに、旧バージョンの Cosminexus サービスプラットフォームをアンインストールすることをお勧めします。

#### 注意

- J2EE サーバ、Management Server、PRF など実行環境の構成要素を停止してからアンインストールしてください。
- 組み込みデータベースを使用している場合、組み込みデータベースを停止してからアンインストールしてください。

### (4) 開発環境のインストール

開発環境で 07-60 の Cosminexus サービスプラットフォームをインストールしてバージョンアップします。

### (5) HCSCTE プロジェクトの作成

バージョンアップした開発環境で、新たに HCSCTE プロジェクトを作成します。また、リポジトリとして利用するディレクトリを指定します。なお、リポジトリとして利用するディレクトリには、運用環境のリポジトリを指定しないでください。

手順の詳細は、「3.1.1 プロジェクトを作成する」を参照してください。

### (6) リポジトリのインポート

(1) でエクスポートしたリポジトリ情報を開発環境にインポートします。インポートするときは、Eclipse のメニューの [プロジェクト] - [自動的にビルド] をオフにしてから、リポジトリをインポートしてください。リポジトリのインポート終了後は、オンに戻してください。

旧バージョンのリポジトリをインポートすると、パッケージングおよび配備定義が自動で実施され、リポジトリ情報が今バージョンに引き継がれます。なお、パッケージングおよび配備定義が自動で実施されるのは、旧バージョンでパッケージングおよび配備定義した HCSC コンポーネントだけです。

リポジトリのインポート方法については、「3.2.3 リポジトリをインポートする」を参照してください。

#### 注意

07-50 より前のバージョンから移行する場合の注意

サービスアダプタ (SessionBean) のユーザ定義クラスに次のファイル名が設定されているときは、警告メッセージが表示されます。

- csmsvcadpdef.jar
- cscmsg\_adpejb.jar

警告メッセージが表示された場合、サービスアダプタ定義画面で、上記のユーザ定義クラスを削除し、ファイル名を変更してから、再度設定してください。

## メモリについての注意

旧バージョンのリポジトリをインポートするときは、メモリを多く必要とします。そのため、次の手順でメモリサイズが十分かどうかを確認し、メモリが不足している場合は、メモリサイズを増やしてからリポジトリをインポートしてください。

1. Eclipse のメニューから、[ MyEclipse ] - [ ガーベッジ・コレクションの実行 ] を選択します。
2. [ メモリー使用量 (MB) ] ダイアログの [ ガーベッジ・コレクション後 ] に表示されている、「最大許容メモリー・サイズ」および「メモリー使用量」を確認します。
3. 次の条件に当てはまる場合、「2.1.3 Eclipse の設定ファイルの確認」を参照し、推奨値に従ってメモリサイズ (-Xmx の値) を大きくします。条件に当てはまらない場合、対処は不要です。

条件：「メモリー使用量」+ 定義済みサービス部品数 × 3 > 「最大許容メモリー・サイズ」

推奨値：変更後の「-Xmx」の値 = 定義済みサービス部品数 × 3 + 変更前の「-Xmx」の値

開発環境にインポートしたリポジトリ情報を使用して実際に運用する場合、このリポジトリ情報を開発環境から運用環境にエクスポートし、運用環境から HCSC コンポーネントを配備します。

### 付録 A.3 下位バージョンで開発環境と運用環境のリポジトリを共有していた場合の移行手順

バージョン 07-60 以降では、開発環境と運用環境で同じリポジトリを共有できません。開発環境と運用環境で同じリポジトリを共有していた場合、次の図に示す (1) ~ (6) の手順に従って、Cosminexus サービスプラットフォームを旧バージョンから 07-60 へ移行してください。

図 A-2 下位バージョンからの移行手順（開発環境と運用環境のリポジトリを共有していた場合）

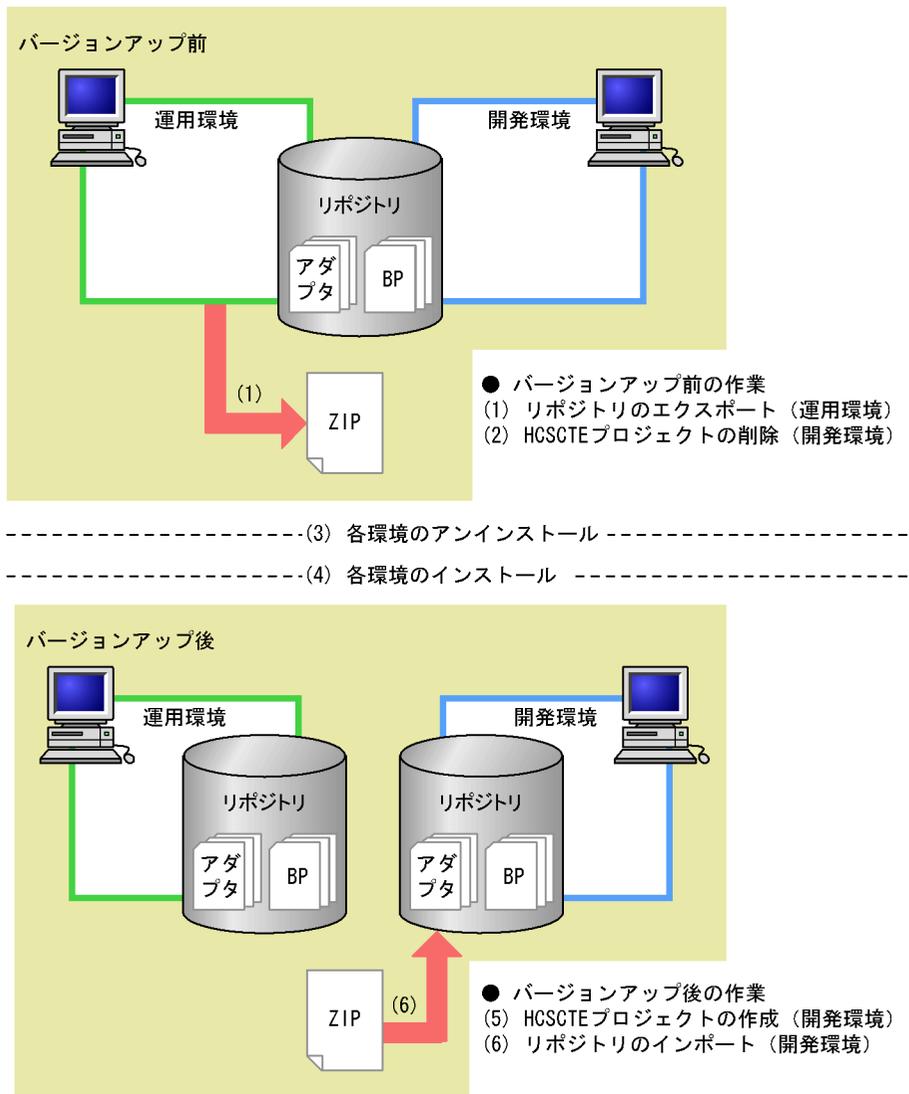


図 A-2 の (1) ~ (6) の手順の詳細を次に示します。

### (1) リポジトリのエクスポート（運用環境）

cscrcptcl コマンド（-export オプション）を運用環境で実行して、バージョンアップ前のリポジトリをエクスポートします。複数のリポジトリを使用している場合、必要なリポジトリをすべてエクスポートして、退避します。

## (2) HCSCTE プロジェクトの削除 (開発環境)

HCSCTE プロジェクトをいったん削除します。

手順の詳細は、「3.1.4 プロジェクトを削除する」を参照してください。

## (3) 各環境のアンインストール

開発環境、運用環境で旧バージョンの Cosminexus サービスプラットフォームをアンインストールします。Cosminexus 以下のディレクトリ情報を退避したあとに、旧バージョンの Cosminexus サービスプラットフォームをアンインストールすることをお勧めします。

注意

- J2EE サーバ、Management Server、PRF など実行環境の構成要素を停止してからアンインストールしてください。
- 組み込みデータベースを使用している場合、組み込みデータベースを停止してからアンインストールしてください。
- HCSC サーバをアンセットアップしたあとに、リポジトリルート配下を削除してください。

## (4) 各環境のインストール

開発環境、運用環境で 07-60 の Cosminexus サービスプラットフォームをインストールしてバージョンアップします。

## (5) HCSCTE プロジェクトの作成 (開発環境)

バージョンアップした開発環境で、新たに HCSCTE プロジェクトを作成します。また、リポジトリとして利用するディレクトリを指定します。なお、リポジトリとして利用するディレクトリには、運用環境のリポジトリを指定しないでください。

手順の詳細は、「3.1.1 プロジェクトを作成する」を参照してください。

## (6) リポジトリのインポート (開発環境)

(1) でエクスポートしたリポジトリ情報を開発環境にインポートします。インポートするときは、Eclipse のメニューの [プロジェクト] - [自動的にビルド] をオフにしてから、リポジトリをインポートしてください。リポジトリのインポート終了後は、オンに戻してください。

旧バージョンのリポジトリをインポートすると、パッケージングおよび配備定義が自動で実施され、リポジトリ情報が今バージョンに引き継がれます。なお、パッケージングおよび配備定義が自動で実施されるのは、旧バージョンでパッケージングおよび配備定義した HCSC コンポーネントだけです。

リポジトリのインポート方法については、「3.2.3 リポジトリをインポートする」を参照してください。

## 注意

### 07-50 より前のバージョンから移行する場合の注意

サービスアダプタ (SessionBean) のユーザ定義クラスに次のファイル名が設定されているときは、警告メッセージが表示されます。

- csmsvcadpdef.jar
- cscmsg\_adpejb.jar

警告メッセージが表示された場合、サービスアダプタ定義画面で、上記のユーザ定義クラスを削除し、ファイル名を変更してから、再度設定してください。

### メモリについての注意

旧バージョンのリポジトリをインポートするときは、メモリを多く必要とします。そのため、次の手順でメモリサイズが十分かどうかを確認し、メモリが不足している場合は、メモリサイズを増やしてからリポジトリをインポートしてください。

1. Eclipse のメニューから、[ MyEclipse ] - [ ガーベッジ・コレクションの実行 ] を選択します。
2. [ メモリー使用量 (MB) ] ダイアログの [ ガーベッジ・コレクション後 ] に表示されている、「最大許容メモリー・サイズ」および「メモリー使用量」を確認します。
3. 次の条件に当てはまる場合、「2.1.3 Eclipse の設定ファイルの確認」を参照し、推奨値に従ってメモリサイズ (-Xmx の値) を大きくします。条件に当てはまらない場合、対処は不要です。  
条件: 「メモリー使用量」+ 定義済みサービス部品数 × 3 > 「最大許容メモリー・サイズ」  
推奨値: 変更後の「-Xmx」の値 = 定義済みサービス部品数 × 3 + 変更前の「-Xmx」の値

開発環境にインポートしたリポジトリ情報を使用して実際に運用する場合、このリポジトリ情報を開発環境から運用環境にエクスポートし、運用環境から HCSC コンポーネントを配備します。

## 付録 B 上流ツールとの連携で使用する BPEL のサポート範囲

ここでは、モデリングツールで作成した BPEL ファイルをインポートする場合の、BPEL ファイルのサポート範囲およびビジネスプロセス定義への変換ルールについて説明します。

### ! 注意事項

BPEL ファイルの要素・属性のうち、以降の説明に登場しない要素・属性は、ビジネスプロセスの定義に変換されません。

BPEL ファイル内の要素が、ビジネスプロセスのどの定義内容に変換されるのかの関係を次の表に示します。

表 B-1 BPEL ファイルの要素とビジネスプロセス定義の関係

変換前の要素 (BPEL ファイル)	変換後の定義内容 (ビジネスプロセス)	変換方法の参照先	
process 要素	variables 要素	変数	付録 B.1(2)
	correlationSets 要素	相関セット	付録 B.1(3)
	faultHandlers 要素	フォルト処理	付録 B.1(4)
	receive 要素	受付アクティビティ	付録 B.2(1)
	reply 要素	応答アクティビティ	付録 B.2(2)
	invoke 要素	サービス呼出アクティビティ	付録 B.2(3)
	assign 要素	代入アクティビティ	付録 B.2(4)
	empty 要素	無操作アクティビティ	付録 B.2(5)
	wait 要素		
	terminate 要素		
	compensate 要素		
	pick 要素		
	throw 要素		
	scope 要素	スコープアクティビティ	付録 B.3(1)
	while 要素	繰り返しアクティビティ	付録 B.3(2)
	switch 要素	分岐アクティビティ	付録 B.3(3)
	flow 要素	並列処理アクティビティ	付録 B.3(4)
	sequence 要素	順次処理アクティビティ	付録 B.3(5)

ポイント

- ビジネスプロセス内の各アクティビティを連結するコネクションの定義内容は、BPEL ファイルの構造、各要素の定義などから判断して設定されます。
- BPEL ファイルをインポートしてビジネスプロセスを作成した場合に、Java 呼出アクティビティおよびデータ変換アクティビティが定義されることはありません。

以降の項で、BPEL ファイルの各要素と、ビジネスプロセス定義の内容との関係の詳細について説明します。

## 付録 B.1 ビジネスプロセス全体の定義に関連する要素の変換

BPEL ファイルに定義されている要素のうち、ビジネスプロセス全体に関連する内容（ビジネスプロセスそのものの設定、利用する変数など）の変換について次に説明します。

### (1) process 要素の変換

process 要素およびその下位要素・属性は、ビジネスプロセス全体の定義に関連する内容に変換されます。

変換の詳細を次の表に示します。

表 B-2 process 要素の変換

BPEL ファイルの要素		ビジネスプロセスの定義内容	
		定義内容	説明
process 要素	variables 要素	変数	詳細は「付録 B.1(2) variables 要素の変換」を参照してください。
	correlationSets 要素	相関セット	詳細は「付録 B.1(3) correlationSets 要素の変換」を参照してください。
	faultHandlers 要素	フォルト処理	詳細は「付録 B.1(4) faultHandlers 要素の変換」を参照してください。
-	-	ビジネスプロセス名	ビジネスプロセス定義を追加するためのダイアログで指定したビジネスプロセス名が設定されます。
-	-	ビジネスプロセスのバージョン	「1」が設定されます。
-	-	永続化	ビジネスプロセス定義を追加するためのダイアログで指定した永続化の有無が設定されます。

(凡例)

- : 該当する要素はありません。定義内容は、自動設定されるか、またはビジネスプロセス定義を追加するためのダイアログで指定した内容が設定されます。

ビジネスプロセスの各アクティビティに変換される要素については、「付録 B.2 基本アクティビティに関連する要素の変換」および「付録 B.3 構造アクティビティに関連する要素の変換」を参照してください。

## (2) variables 要素の変換

variables 要素およびその下位要素・属性は、ビジネスプロセス（またはスコープ内）に設定されている変数の定義に変換されます。インポートしたあとは、定義内容を変数・関連セット一覧ダイアログで変更できます。

### ! 注意事項

Cosminexus サービスプラットフォームでは、スコープ内のフォルト処理を構成するアクティビティの割当変数として定義できる変数は、フォルト処理のあるスコープの上位スコープで定義されている変数です。

そのため、BPEL ファイルの `scope/faultHandlers` 要素以下の要素で、変数として同じ `scope` 要素内の変数を利用している場合、インポートしたあとにビジネスプロセスで割当変数を再定義してください。

変換の詳細を次の表に示します。表内の「項目」に示す名称は、変数・関連セット一覧ダイアログの項目名です。

表 B-3 variables 要素の変換

variables 要素の下位要素または属性	ビジネスプロセスの定義内容	
	項目	説明
name 属性	変数名	変数の名称が設定されます。
messageType 属性	種別	この属性が定義されていた場合、type 属性は文字列型 (string) に変換されます。そのため、変換後に変数の種別をメッセージ型 (messageType) に変更し、電文フォーマットを登録する必要があります。
type 属性	種別	次に示す変数の種別が設定されます。 <ul style="list-style-type: none"> <li>• boolean の場合 「BOOLEAN_TYPE_ID」が設定されます。</li> <li>• double で表現可能な型 の場合 「NUMERIC_TYPE_ID」が設定されます。</li> <li>• 上記以外または未定義の場合 「STRING_TYPE_ID」が設定されます。</li> </ul>

variables 要素の下位要素または属性	ビジネスプロセスの定義内容	
	項目	説明
-	部分指定	変数の部分指定には、値は設定されません。部分指定を利用する場合、インポートしたあとに変数・相関セット一覧ダイアログで設定してください。

(凡例)

- : 該当する要素はありません。

注

次の型が該当します。

int , short , byte , unsignedInt , unsignedShort , unsignedByte , float , double

### (3) correlationSets 要素の変換

correlationSets 要素の定義内容は、ビジネスプロセスの定義では「相関セット」に当たる情報です。BPEL ファイルをインポートしてビジネスプロセスを作成しても値は設定されません。

ビジネスプロセスで相関セットを利用する場合、インポートしたあとに変数・相関セット一覧ダイアログで相関セットを定義し、相関セットを使用するアクティビティの割当相関セット群ダイアログで相関セットを割り当ててください。

### (4) faultHandlers 要素の変換

faultHandlers 要素およびその下位要素・属性は、ビジネスプロセス内のフォルト処理の定義内容に変換されます。変換後の定義内容は、フォルト処理の割当ダイアログで変更できます。

#### ! 注意事項

- process 要素の直下に faultHandlers 要素が定義されている場合、最上位のビジネスプロセスにスコープが作成され、そのスコープのフォルト処理として定義されます。その場合、process 要素の直下のアクティビティは作成されたスコープ内に移動されます。
- scope 要素の配下に faultHandlers 要素が定義されている場合、上位のスコープ内にフォルト処理が設定されます。

変換の詳細を次の表に示します。表内の「項目」に示す名称は、フォルト処理の割当ダイアログの項目名です。

表 B-4 faultHandlers 要素の変換

faultHandlers 要素の下位要素または属性		ビジネスプロセスの定義内容	
		項目	説明
catch 要素	faultVariable 属性	割当変数	変数については「付録 B.1(2) variables 要素の変換」の注意事項も参照してください。
catchAll 要素		割当変数	この要素がある場合、割当変数に「catch-all」が定義されます。
-		遷移先	catch 要素または catchAll 要素が定義されている場合、フォルト処理の遷移先が自動的に設定されます。

( 凡例 )

- : 該当する要素はありません。定義内容は自動で設定されます。

## 付録 B.2 基本アクティビティに関連する要素の変換

BPEL ファイルに定義されている要素のうち、基本アクティビティに変換される要素の変換について次に説明します。

### ( 1 ) receive 要素の変換

receive 要素およびその下位要素・属性は、受付アクティビティの定義内容に変換されます。インポートしたあとは、定義内容を受付アクティビティダイアログで変更できます。

#### ! 注意事項

同一の receive 要素に portType 属性および operation 属性がすでに定義されている場合、receive 要素は無操作アクティビティに変換されます。

変換の詳細を次の表に示します。表内の「項目」に示す名称は、受付アクティビティダイアログの項目名です。

表 B-5 receive 要素の変換

receive 要素の下位要素または属性	ビジネスプロセスの定義内容	
	項目	説明
operation 属性	オペレーション名	portType 属性が異なり、かつ operation 属性が同一の receive 要素がすでに定義されている場合、オペレーション名の末尾に <n> ( n は 1 以上の整数 ) が付与されます。
variable 属性	割当変数	変数については「付録 B.1(2) variables 要素の変換」の注意事項も参照してください。

receive 要素の下位要素または属性		ビジネスプロセスの定義内容	
		項目	説明
createInstance 属性		インスタンス生成	属性値に関係なく「yes」に変換されます。
correlations 要素	correlation 要素	割当相関セット群	BPEL ファイルの定義内容はビジネスプロセスに反映されません。インポートしたあとに、割当相関セット群ダイアログで設定してください。
-		通信モデル	ビジネスプロセス定義の内容から判断して「同期」または「非同期」が設定されます。

(凡例)

- : 該当する要素はありません。定義内容は自動で設定されます。

## (2) reply 要素の変換

reply 要素およびその下位要素・属性は、応答アクティビティの定義内容に変換されます。インポートしたあとは、定義内容を応答アクティビティダイアログで変更できます。

変換の詳細を次の表に示します。表内の「項目」に示す名称は、応答アクティビティダイアログの項目名です。

表 B-6 reply 要素の変換

reply 要素の下位要素または属性		ビジネスプロセスの定義内容	
		項目	説明
operation 属性		オペレーション名	portType 属性と operation 属性が一致する receive 要素のオペレーション名に <n> が付与された場合、オペレーション名の末尾に <n> (n は 1 以上の整数) が付与されます。
variable 属性		割当変数	変数については「付録 B.1(2) variables 要素の変換」の注意事項も参照してください。
faultName 属性		フォルト名	プレフィックスが付与されている場合は、ローカル名に変換されます。
correlations 要素	correlation 要素	割当相関セット群	BPEL ファイルの定義内容はビジネスプロセスに反映されません。インポートしたあとに、割当相関セット群ダイアログで設定してください。

## (3) invoke 要素の変換

invoke 要素およびその下位要素・属性は、サービス呼出アクティビティの定義内容に変換されます。インポートしたあとは、定義内容をサービス呼出アクティビティダイアログまたはフォルト処理の割当ダイアログで変更できます。

また、invoke 要素の下位要素として、catch 要素または catchAll 要素が定義されている場合、インポートしたあとのサービス呼出アクティビティにはフォルト処理（フォルトコネクションで連結したアクティビティ）が設定されます。

変換の詳細を次の表に示します。表内の「項目」に示す名称は、サービス呼出アクティビティダイアログまたはフォルト処理の割当ダイアログの項目名です。

表 B-7 invoke 要素の変換

invoke 要素の下位要素または属性		ビジネスプロセスの定義内容	
		項目	説明
portType 属性		サービス名	該当するサービス部品やオペレーションがリポジトリにある場合にだけ変換されます。ローカル名に変換されます。
operation 属性		オペレーション名	該当するサービス部品やオペレーションがリポジトリにある場合にだけ変換されます。
inputVariable 属性		要求電文用割当変数	サービス部品とオペレーションがリポジトリに存在する場合に設定されます。 変数については「付録 B.1(2) variables 要素の変換」の注意事項も参照してください。
outputVariable 属性		応答電文用割当変数	サービス部品とオペレーションがリポジトリに存在し、かつ同期の場合に設定されます。 変数については「付録 B.1(2) variables 要素の変換」の注意事項も参照してください。
correlations 要素	correlation 要素	割当相関セット群	BPEL ファイルの定義内容はビジネスプロセスに反映されません。インポートしたあとに、割当相関セット群ダイアログで設定してください。
catch 要素	faultVariable 属性	割当変数	フォルト処理の割当ダイアログの項目です。 変数については「付録 B.1(2) variables 要素の変換」の注意事項も参照してください。
catchAll 要素		割当変数	フォルト処理の割当ダイアログの項目です。 この要素がある場合、割当変数に「catch-all」が定義されます。
-		遷移先	フォルト処理の割当ダイアログの項目です。 catch 要素または catchAll 要素が定義されている場合、フォルト処理の遷移先が自動的に設定されます。

(凡例)

- : 該当する要素はありません。定義内容は自動で設定されます。

#### (4) assign 要素の変換

assign 要素およびその下位要素・属性は、代入アクティビティの定義内容に変換されません。インポートしたあとは、定義内容を代入アクティビティダイアログまたは代入アクティビティサブダイアログで変更できます。

変換の詳細を次の表に示します。表内の「項目」に示す名称は、代入アクティビティダイアログまたは代入アクティビティサブダイアログの項目名です。

表 B-8 assign 要素の変換

assign 要素の下位要素または属性			ビジネスプロセスの定義内容	
			項目	説明
copy 要素	from 要素	variable 属性	名前 (コピー元の変数)	代入アクティビティサブダイアログの項目です。変数については「付録 B.1(2) variables 要素の変換」の注意事項も参照してください。
		expression 属性	値 (式)	代入アクティビティサブダイアログの項目です。variable 属性が定義されている場合は変換されません。
		タグ値	値 (式)	代入アクティビティサブダイアログの項目です。variable 属性および expression 属性が定義されていない場合、タグ値が式の値として設定されます。
	to 要素	variable 属性	変数名 (コピー先)	代入アクティビティサブダイアログの項目です。変数については「付録 B.1(2) variables 要素の変換」の注意事項も参照してください。

注

from 要素のタグ値が変換元になります。

#### (5) empty 要素の変換

empty 要素は、無操作アクティビティの定義内容に変換されます。インポートしたあとは、定義内容を無操作アクティビティダイアログで変更できます。

なお、empty 要素には、変換される下位要素・属性はありません。

**!** 注意事項

BPEL ファイルの要素のうち、wait 要素、terminate 要素、compensate 要素、および pick 要素は、empty 要素と同様に無操作アクティビティに変換されます。

また、receive 要素の portType 属性および operation 属性が、reply 要素の portType 属性および operation 属性と同じ場合には、receive 要素は無操作アクティビティに変換されます。

**(6) throw 要素の変換**

throw 要素およびその下位要素・属性は、フォルト送出アクティビティの定義内容に変換されます。インポートしたあとは、定義内容をフォルト送出アクティビティダイアログで変更できます。

変換の詳細を次の表に示します。表内の「項目」に示す名称は、フォルト送出アクティビティダイアログの項目名です。

表 B-9 throw 要素の変換

throw 要素の下位要素または属性	ビジネスプロセスの定義内容	
	項目	説明
faultVariable 属性	割当変数	変数については「付録 B.1(2) variables 要素の変換」の注意事項も参照してください。

**付録 B.3 構造アクティビティに関連する要素の変換**

BPEL ファイルに定義されている要素のうち、基本アクティビティに変換される要素の変換について次に説明します。

**(1) scope 要素の変換**

scope 要素およびその下位要素・属性は、スコープアクティビティの定義内容に変換されます。インポートしたあとは、定義内容をスコープアクティビティダイアログで変更できます。

また、scope 要素内に定義されている変数、相関セット、フォルト処理の情報は、変数・相関セット一覧ダイアログおよびフォルト処理の割当ダイアログで変更できます。

変換の詳細を次の表に示します。

表 B-10 scope 要素の変換

scope 要素の下位要素または属性	ビジネスプロセスの定義内容	
	項目	説明
variables 要素	変数	スコープ内で利用する変数の情報です。変数については「付録 B.1(2) variables 要素の変換」を参照してください。

scope 要素の下位要素または属性	ビジネスプロセスの定義内容	
	項目	説明
correlationSets 要素	相関セット	スコープ内で使用する相関セットの情報です。相関セットの情報については「付録 B.1(3) correlationSets 要素の変換」を参照してください。
faultHandlers 要素	フォルト処理	スコープ内のフォルト処理の情報です。フォルト処理の情報については「付録 B.1(4) faultHandlers 要素の変換」を参照してください。

## (2) while 要素の変換

while 要素およびその下位要素・属性は、繰り返しアクティビティの定義内容に変換されます。インポートしたあとは、定義内容を繰り返しアクティビティダイアログおよび条件設定ダイアログで変更できます。

変換の詳細を次の表に示します。表内の「項目」に示す名称は、繰り返しアクティビティダイアログまたは条件設定ダイアログの項目名です。

表 B-11 while 要素の変換

while 要素の下位要素または属性	ビジネスプロセスの定義内容	
	項目	説明
condition 属性	条件式	条件設定ダイアログの項目です。
-	最大繰り返し回数	「100」が自動的に設定されます。

(凡例)

- : 該当する要素はありません。定義内容は自動で設定されます。

## (3) switch 要素の変換

switch 要素およびその下位要素・属性は、分岐開始アクティビティの定義内容に変換されます。インポートしたあとは、定義内容を分岐アクティビティダイアログおよび条件設定ダイアログで変更できます。

### ! 注意事項

分岐終了アクティビティは、switch 要素以下が分岐開始アクティビティに変換されたときに自動的に設定されます。その際、分岐終了アクティビティの名称として、分岐開始アクティビティの名称に「\_終了」を付けた名称が設定されます。

変換の詳細を次の表に示します。表内の「項目」に示す名称は、分岐アクティビティダ

イアログまたは条件設定ダイアログの項目名です。

表 B-12 switch 要素の変換

switch 要素の下位要素または属性		ビジネスプロセスの定義内容	
		項目	説明
case 要素	condition 属性	条件式	条件設定ダイアログの項目です。
-		遷移先	分岐アクティビティダイアログの項目です。case 要素および otherwise 要素の定義から判断して設定されます。
-		条件名	分岐アクティビティダイアログの項目です。「condition<n>」が自動的に設定されます。<n> は分岐内でユニークになる整数が設定されます。
-		優先順位	分岐アクティビティダイアログの項目です。switch 要素内の条件の出現順に設定されます。

(凡例)

- : 該当する要素はありません。定義内容は自動で設定されます。

#### (4) flow 要素の変換

flow 要素は、並列処理開始アクティビティの定義内容に変換されます。インポートしたあとは、定義内容を並列処理アクティビティダイアログで変更できます。

なお、flow 要素には、変換される下位要素・属性はありません。

#### ! 注意事項

- リンクの定義 (BPEL ファイル内の links 要素以下の定義) は変換されません。
- 並列処理終了アクティビティは、flow 要素以下が並列処理開始アクティビティに変換されたときに自動的に設定されます。その際、並列処理終了アクティビティの名称として、並列処理開始アクティビティの名称に「\_終了」を付けた名称が設定されます。

#### (5) sequence 要素の変換

sequence 要素は、順次処理アクティビティの定義内容に変換されます。定義内容は変更できません。

なお、sequence 要素には、変換される下位属性はありません。

Cosminexus サービスプラットフォームでは、「6.6.15 順次処理アクティビティ」に相当する位置に記述された sequence 要素だけが、ビジネスプロセスの定義に変換されません。

---

## 付録 C Cosminexus サービスプラットフォームのサンプルプログラム

サンプルプログラムを実際に動かして、Cosminexus サービスプラットフォームを理解していきましょう。

### 付録 C.1 サンプルプログラムを動かしてみよう

サンプルプログラムとして、次の 3 種類を用意しています。

- HelloServiceAdapter サンプルプログラム  
サービスリクエスト からサービスアダプタ を介してサービス呼び出すサンプルプログラムです。
- HelloBusinessProcess サンプルプログラム  
ビジネスプロセス からサービスアダプタを介してサービス呼び出すサンプルプログラムです。
- 商品手配サンプルプログラム  
ビジネスプロセスからサービスアダプタを介して商品の在庫を引き当てたり、配送を手配したりするサービス呼び出すサンプルプログラムです。実際の業務に近い型を想定しています。

注 詳細については、「付録 C.2 サンプルプログラムを構成している要素」を参照してください。

最初に、最も基本的な HelloServiceAdapter サンプルプログラムを動かして、イメージをつかんでみましょう。

#### (1) サンプルプログラムを動かすための準備

サンプルプログラムを動かすには、Cosminexus サービスプラットフォームの開発環境 (uCosminexus Service Architect) が必要です。Cosminexus サービスプラットフォームの開発環境を使用するための動作環境、およびセットアップについては、「1.4 動作環境」、および「2.1 開発環境を使用するためのセットアップ」を参照してください。

サンプルプログラムを動かすための手順を次の図に示します。

図 C-1 サンプルプログラムを動かすための手順



## (a) テスト環境を構築します

テスト環境を構築するには、HCSC 簡易セットアップ機能を使います。

<手順>

1. 次のパスを環境変数 PATH に追加します。  
 <Cosminexus インストールディレクトリ>%CSC%bin%
2. コマンドプロンプトで次のコマンドを実行し、簡易構築画面を表示させます。  
 csceasysetup.bat
3. [セットアップ] ボタンをクリックします。  
 テスト環境のセットアップが開始されます。セットアップの状況は、簡易構築画面のコンソールに表示されます。コンソールに「簡易構築機能のセットアップを終了します」と表示されたら、テスト環境は正常にセットアップされています。  
 テスト環境のセットアップが異常終了した場合の手順については、「2.4.2(1) テスト環境のセットアップ」を参照してください。
4. コマンドプロンプトを表示します。
5. 次のコマンドを入力して、テスト環境の組み込みデータベースを起動します。  
 <Cosminexus のインストールディレクトリ>%CSC%DB%bats%cdstart.bat

## 参考

このコマンドを入力した場合、「KFPS01853-W Hostname=< ホスト名 >, unable to execute pdstart command, unit state not OFFLINE」というメッセージが出力される場合があります。これは、同じマシンで二回目以降にサンプルプログラムを動かした場合に、すでに組み込みデータベースが起動されているために出力されるメッセージ

です。そのため、このメッセージが出力されても問題ありません。

6. 次のコマンドを入力して、テスト環境の Cosminexus Performance Tracer, J2EE サーバ、および HCSC サーバを起動します。

<Cosminexus のインストールディレクトリ >%CSC%bin%cscsvstart.bat -system

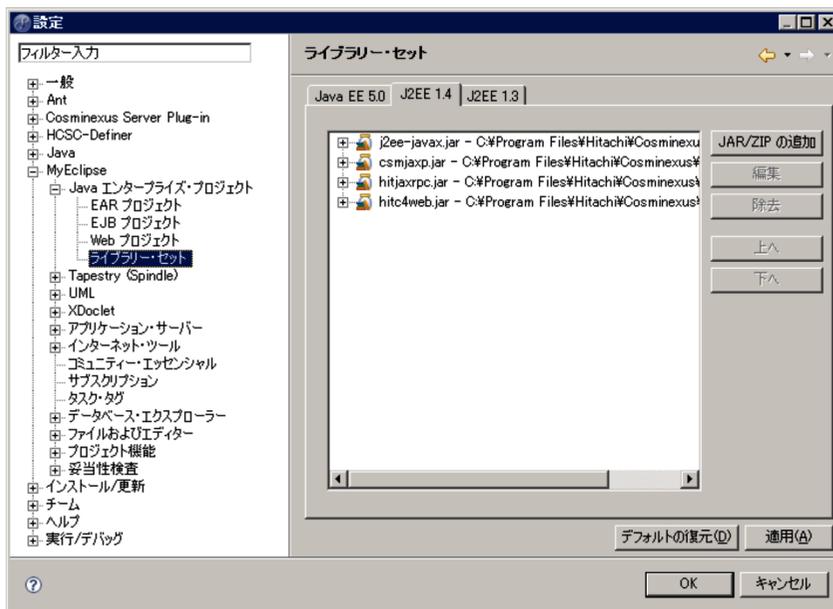
(b) MyEclipse を設定します

MyEclipse の設定では、ライブラリーパスの設定、および J2EE サーバの選択を実施します。

ライブラリーパスの設定

<手順>

1. メニューから [ ウィンドウ ] - [ 設定 ] を選択します。
2. ツリービューから [ MyEclipse ] - [ Java エンタープライズ・プロジェクト ] - [ ライブラリー・セット ] を選択し、[ J2EE 1.4 ] タブを選択します。
3. [ JAR/ZIP の追加 ] ボタンをクリックします。
4. <Cosminexus インストールディレクトリ >%c4web%lib を開いて、配下の「hitjaxrpc.jar」と「hitc4web.jar」を追加します。

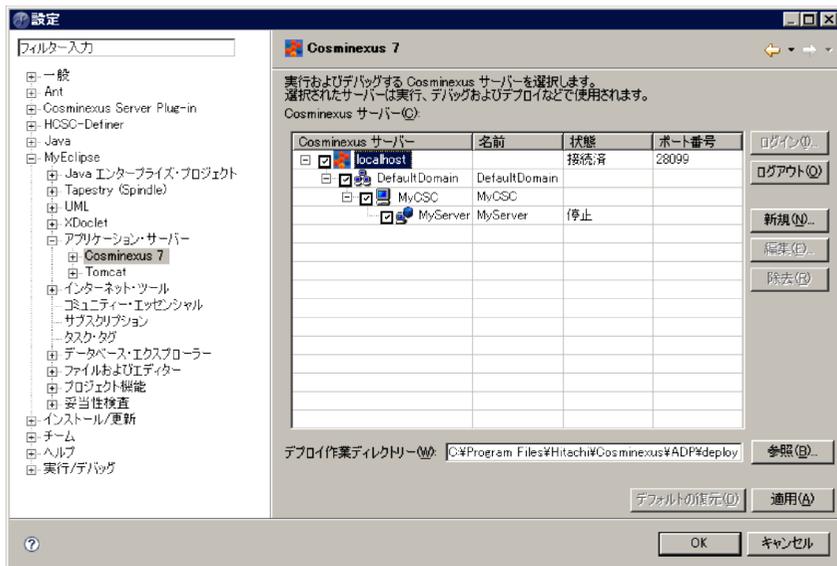


5. MyEclipse を再起動します。

J2EE サーバの選択

<手順>

1. メニューから [ ウィンドウ ] - [ 設定 ] を選択します。
2. ツリービューから [ MyEclipse ] - [ アプリケーション・サーバー ] - [ Cosminexus 7 ] を選択します。
3. Cosminexus サーバを選択し, [ ログイン ] ボタンをクリックします。  
ログイン画面が表示されます。
4. [ 管理ユーザー ID ] に「admin」を, [ パスワード ] に「admin」を入力して, [ OK ] ボタンをクリックします。
5. [ Cosminexus サーバー ] ツリービューで, 使用する J2EE サーバ ( MyServer ) のチェックをオンにして, [ OK ] ボタンをクリックします。

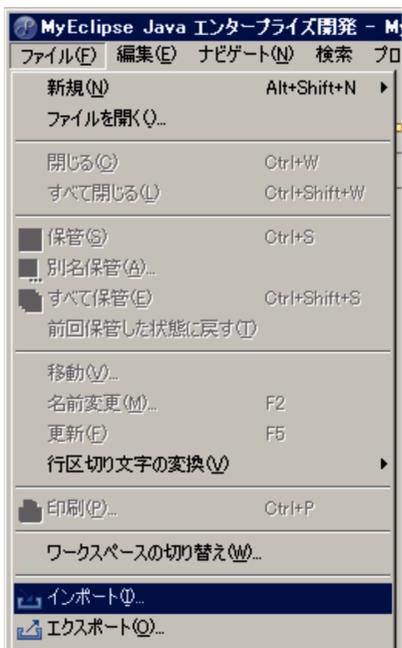


### (c) MyEclipse プロジェクトをインポートします

MyEclipse プロジェクトは, サンプルプログラムごとにインポートします。ここでは, サンプルプログラム ( HelloServiceAdapter サンプルプログラム ) の例で説明しています。

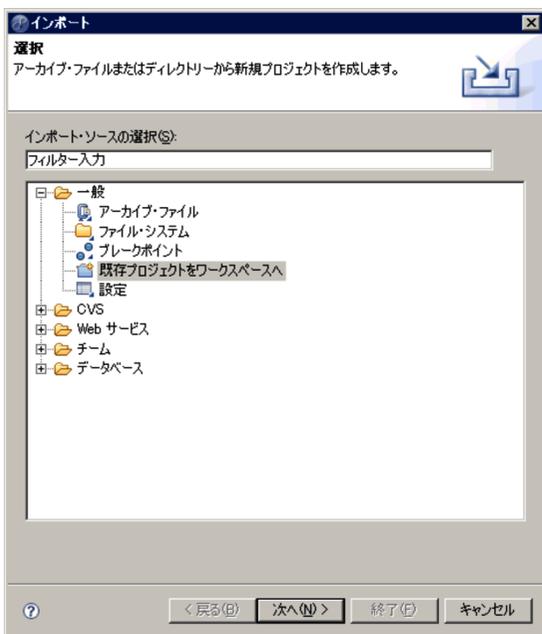
<手順>

1. MyEclipse を起動して, メニューから [ ウィンドウ ] - [ パースペクティブを開く ] - [ MyEclipse Java エンタープライズ開発 ] 選択します。
2. [ OK ] ボタンをクリックします。  
[ MyEclipse Java エンタープライズ開発 ] パースペクティブが表示されます。
3. メニューから [ ファイル ] - [ インポート ] を選択します。



インポート画面が表示されます。

4. [ 一般 ] - [ 既存プロジェクトをワークスペースへ ] を選択します。



5. [ 次へ ] ボタンをクリックします。
6. [ ルートディレクトリの選択 ] を選択し, [ 参照 ] ボタンをクリックして, サンプルプ

プログラムが格納されているディレクトリ (<Cosminexus インストールディレクトリ>¥CSCTE¥Samples¥HelloServiceAdapter) を選択します。

プロジェクト一覧が表示されます。

- すべてのプロジェクトにチェックが入っていることを確認して、[ 終了 ] ボタンをクリックします。



パースペクティブにプロジェクトがインポートされます。

#### (d) Web プロジェクトをデプロイします

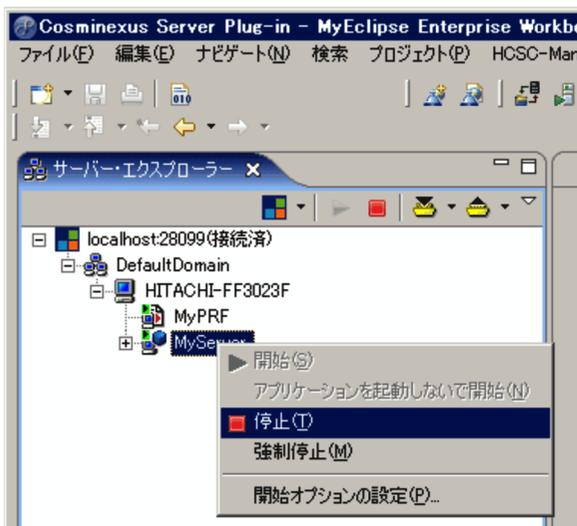
Web プロジェクトは、サンプルプログラムごとにデプロイします。ここでは、サンプルプログラム (HelloServiceAdapter サンプルプログラム) の例で説明しています。

<手順>

- メニューから [ ウィンドウ ] - [ パースペクティブを開く ] を選択し、[ HCSC-Manager ] を選択します。
- [ OK ] ボタンをクリックします。  
[ HCSC-Manager ] パースペクティブが表示されます。
- [ HCSC-Manager ] ビューから [ HCSC-Manager ] を選択し、右クリックして、[ ログイン ] を選択します。  
ログイン画面が表示されます。
- [ ユーザ ID ] に「admin」を、[ パスワード ] に「admin」を入力し、[ OK ] ボタンをクリックします。
- ツリービューの [ Mycsc[localhost] ] をダブルクリックします。

[ MyCSC ] の基本情報ページが表示されます。

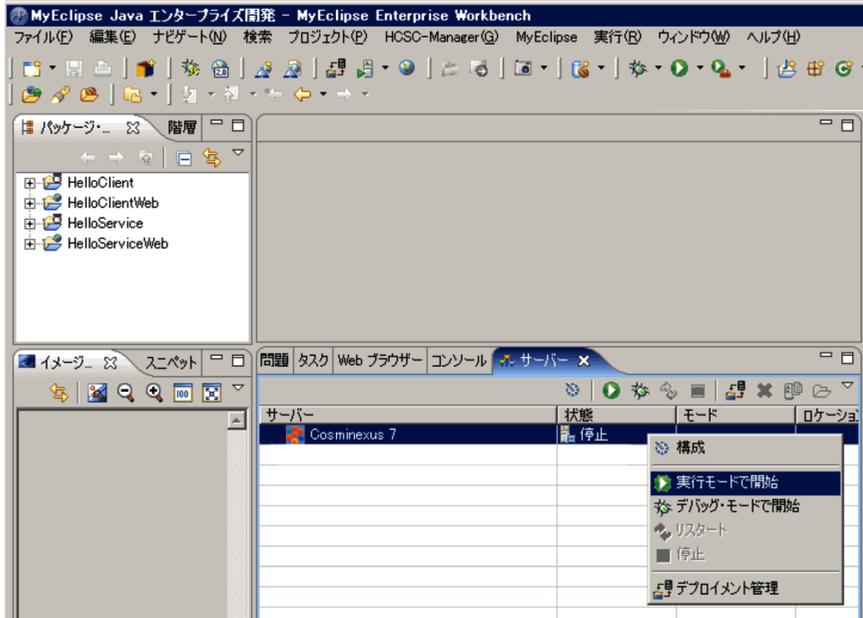
6. [ 稼働運用 ] タブをクリックします。  
稼働運用ページが表示されます。
7. HCSC サーバの [ 稼働状態 ] が [ 停止状態 ] の場合 , [ 起動 ] ボタンをクリックします。  
HCSC サーバが起動します。
8. [ HCSC-Manager ] ビューから [ HCSC-Manager ] を選択し , 右クリックして , [ ログアウト ] を選択します。
9. メニューから [ ウィンドウ ] - [ パースペクティブを開く ] を選択し , [ Cosminexus Server Plug-in ] を選択します。
10. [ OK ] ボタンをクリックします。  
[ Cosminexus Server Plug-in ] パースペクティブが表示されます。
11. [ サーバー・エクスプローラー ] ビューから [ localhost ] を選択し , 右クリックして , [ ログイン ] を選択します。  
ログイン画面が表示されます。
12. [ 管理ユーザー ID ] に「admin」を , [ パスワード ] に「admin」を入力し , [ OK ] ボタンをクリックします。
13. [ サーバー・エクスプローラー ] ビューから [ MyServer ] を選択し , 右クリックして , [ 停止 ] を選択します。



14. メニューから [ ウィンドウ ] - [ パースペクティブを開く ] を選択し , [ MyEclipse Java エンタープライズ開発 ] を選択します。
15. [ OK ] ボタンをクリックします。

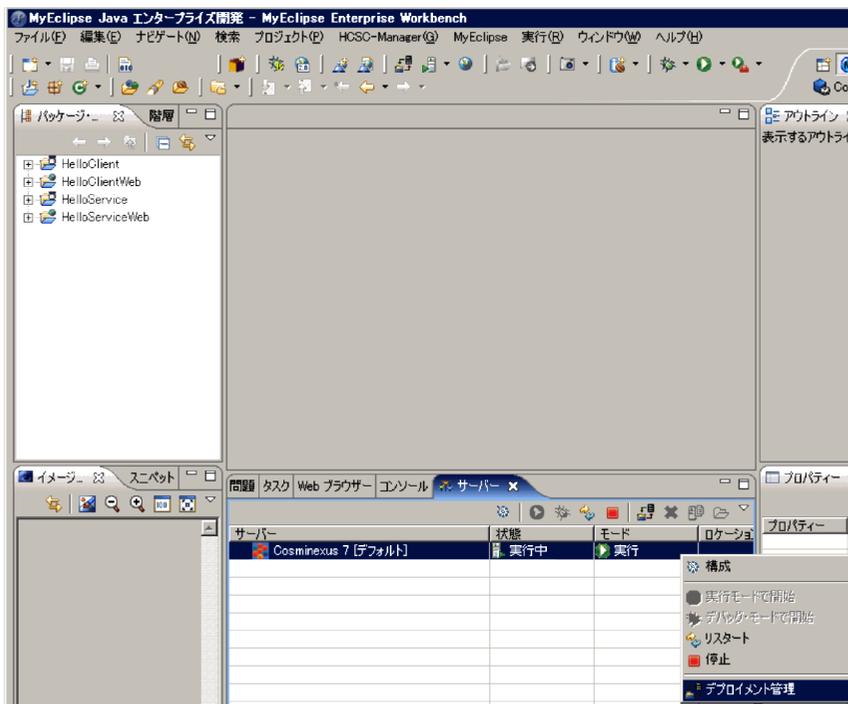
[ MyEclipse Java エンタープライズ開発 ] パースペクティブが表示されます。

- 16.[ サーバー ] ビューから [ Cosminexus 7 ] サーバを選択し、右クリックして [ 実行モードで開始 ] を選択します。  
Cosminexus 7 サーバが表示されていない場合は、「(b) MyEclipse を設定します」の「J2EE サーバの選択」を参照してください。



ログイン画面が表示されます。

- 17.[ 管理ユーザー ID ] に「admin」を、[ パスワード ] に「admin」を入力し、[ OK ] ボタンをクリックします。  
Cosminexus サーバが起動します。
- 18.[ サーバー ] ビューから [ Cosminexus 7 ] サーバを選択し、右クリックして [ デプロイメント管理 ] を選択します。  
[ サーバー・デプロイメント ] ダイアログが表示されます。



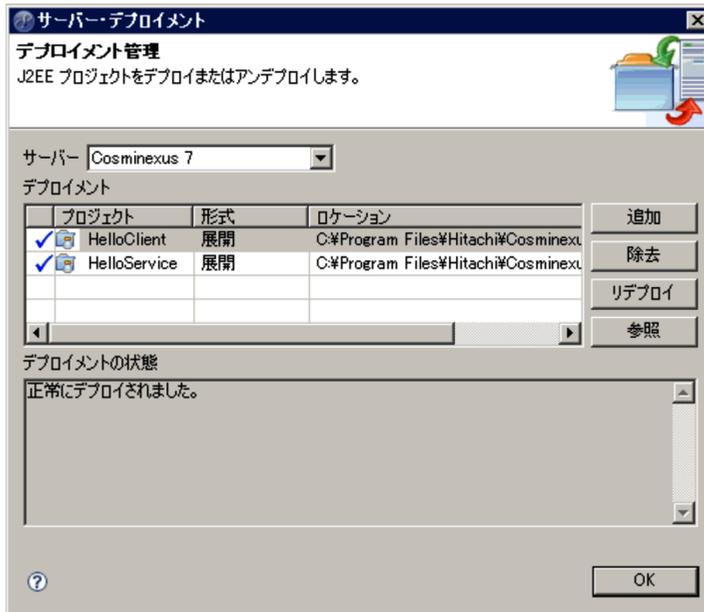
19. [ 追加 ] ボタンをクリックします。

[ 新規デプロイメント ] ダイアログが表示されます。

20. プロジェクトのドロップダウンリストから、[ HelloClient ] を選択し、[ 展開ディレクトリ形式 ] を選択して、[ 終了 ] ボタンをクリックします。



- 21.同様に、プロジェクトのドロップダウンリストから、[ HelloService ] を選択し、[ 展開ディレクトリー形式 ] を選択して、[ 終了 ] ボタンをクリックします。



- 22.[ デプロイメントの状態 ] に「正常にデプロイされました」と表示されていることを確認し、[ OK ] ボタンをクリックします。

- 23.[ サーバー ] ビューの [ Cosminexus 7 ] の配下に [ HelloClient ] と [ HelloService ] プロジェクトが存在することを確認します。

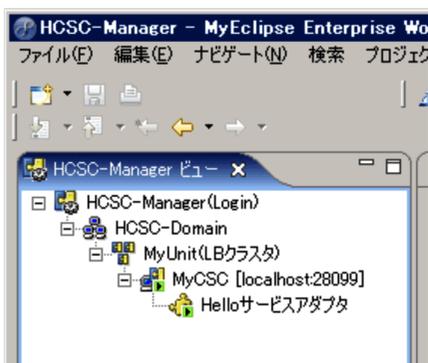
(e) HCSC サーバへ定義内容を配備します

定義内容は、サンプルプログラムごとに HCSC サーバへ配備します。配備する内容は、サンプルプログラムごとに異なります。ここでは、サンプルプログラム (HelloServiceAdapter サンプルプログラム) の例で説明しています。

<手順>

1. 次のパスを環境変数 PATH に追加します。  
<Cosminexus インストールディレクトリ>%CSC%bin
2. HCSC-Manager の画面をログアウトしていることを確認します。  
ログアウトしない場合は、エラーになることがあります。
3. コマンドプロンプトを起動します。
4. HCSC-Manager のコマンドを入力して、リポジトリをインポートします。  
cscreptcl -user admin -pass admin -import <Cosminexus インストールディレクトリ>%CSC%Samples\HelloServiceAdapter\Repository\HelloServiceAdapter.zip  
なお、ほかのサンプルプログラムのリポジトリは、次のところにあります。

- HelloBusinessProcess サンプルプログラム  
<Cosminexus インストールディレクトリ  
>¥CSCTE¥Samples¥HelloServiceBusinessProcess¥Repository¥HelloBusinessProcess.zip
  - 商品手配サンプルプログラム  
<Cosminexus インストールディレクトリ >¥CSCTE¥Samples¥ 商品手配  
¥Repository¥ 商品手配 .zip
5. 次のコマンドを入力して、HCSC サーバに HCSC コンポーネントを配備します。  
csccompodeploy -user admin -pass admin -csc MyCSC -all
  6. 次のコマンドを入力して、HCSC コンポーネントを起動します。  
csccompostart -user admin -pass admin -csc MyCSC -all
  7. 次のコマンドを入力して、標準受付を開始します。  
cscrcptnstart -user admin -pass admin -csc MyCSC
  8. メニューから [ ウィンドウ ] - [ ビューの表示 ] を選択し、[ HCSC-Manager ビュー ] を選択します。
  9. [ OK ] ボタンをクリックします。  
[ HCSC-Manager ] ビューが表示されます。
  - 10.[ HCSC-Manager ] を右クリックして、[ ログイン ] を選択します。  
ログイン画面が表示されます。
  - 11.[ 管理ユーザー ID ] に「admin」を、[ パスワード ] に「admin」を入力して、[ OK ] ボタンをクリックします。
  - 12.[ Hello サービスアダプタ ] が起動していることと、標準受付が開始されていることを確認してログアウトします。



これで、サンプルプログラムを動かすための準備ができました。

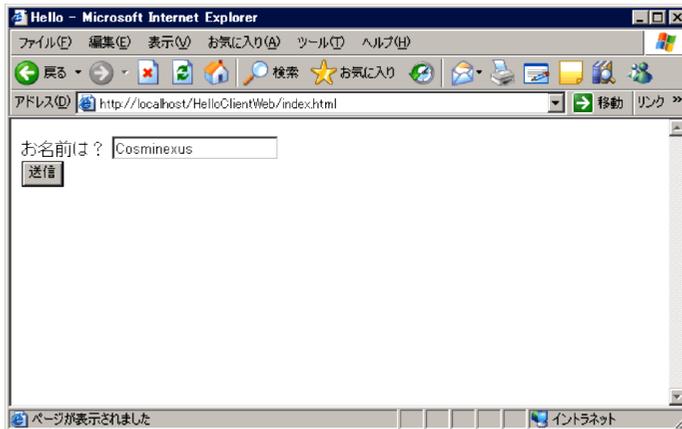
## (2) サンプルプログラムを動かしてみよう

それぞれのサンプルプログラムを動かしてみよう。

(a) HelloServiceAdapter を動かしてみましょう

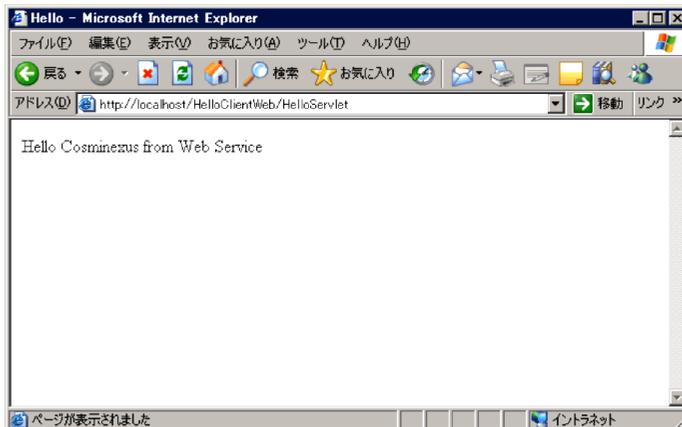
<手順>

1. ブラウザから次に示す URL を入力し，画面を開きます。  
http://localhost/HelloClientWeb/index.html
2. 適当な文字列を入力して，[ 送信 ] ボタンをクリックします。

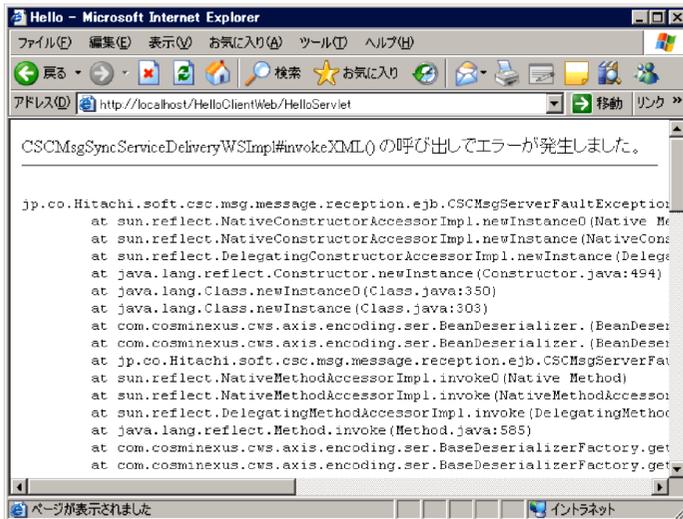


3. 次のように表示されます。

- 正常に応答した場合  
入力した文字列に，文字列が追加され「Hello <入力文字列> from Web Service」( は半角空白) と表示されます。



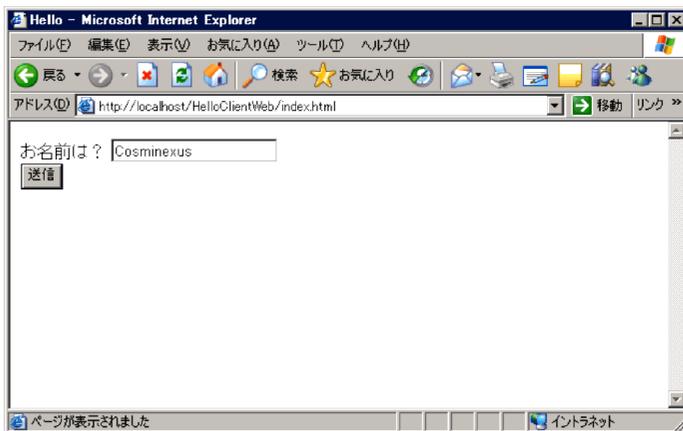
- エラーが発生した場合  
Hello サービスアダプタが起動していない場合、エラーの内容が表示されます。



(b) HelloBusinessProcess サンプルプログラムを動かしてみましょう

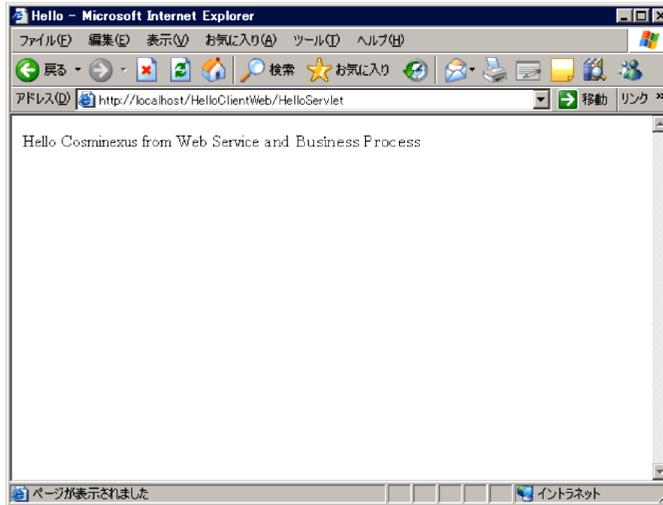
< 手順 >

1. ブラウザから次に示す URL を入力し、画面を開きます。  
`http://localhost/HelloClientWeb/index.html`
2. 適当な文字列を入力して、[ 送信 ] ボタンをクリックします。

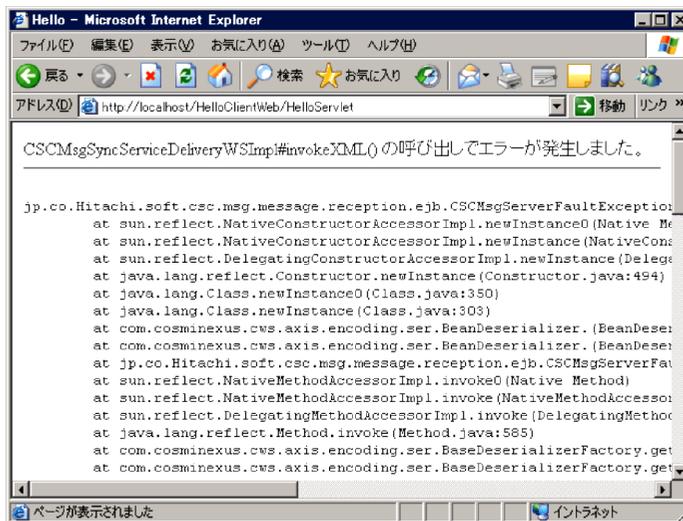


3. 次のように表示されます。

- 正常に応答した場合  
 入力した文字列に、文字列が追加され「Hello <入力文字列> from Web Service and Business Process」( は半角空白 )と表示されます。



- エラーが発生した場合  
 Hello ビジネスプロセスが起動していない場合、エラーの内容が表示されます。



(c) 商品手配サンプルプログラムを動かしてみましょう

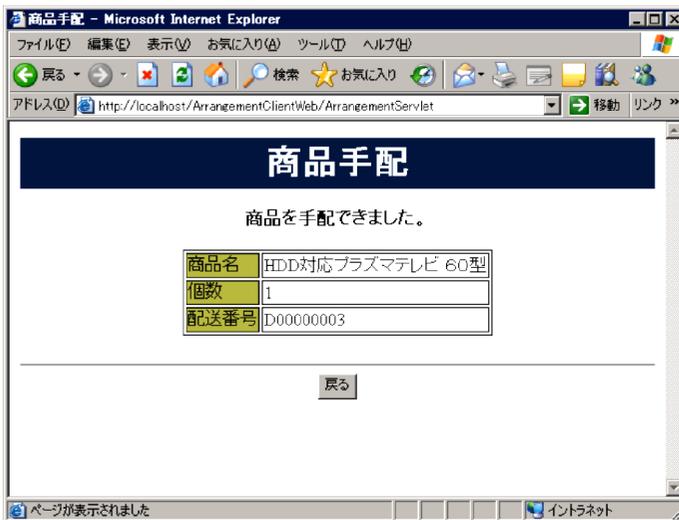
<手順>

1. ブラウザから次に示す URL を入力し、画面を開きます。  
<http://localhost/ArrangementClientWeb/index.html>
2. 商品名と個数を選択して、[ 手配 ] ボタンをクリックします。



3. 次のように表示されます。

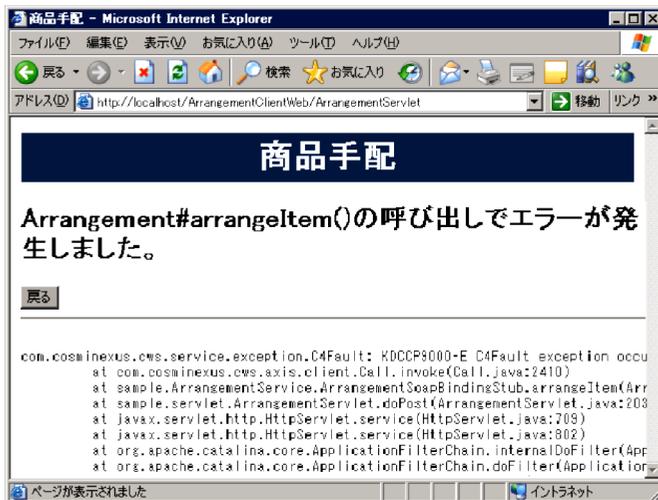
- 正常に応答した（手配が完了した）場合  
手配した商品の商品名，個数，および配送番号が表示されます。



- 正常に応答した（在庫がない）場合  
選択した商品名の個数，および在庫がないことが表示されます。



- エラーが発生した場合  
サービスが起動していない場合，エラーの内容が表示されます。



## 参考

商品手配サンプルプログラムでは、各商品の在庫数は 10 個です。手配が完了するたびに、在庫数から手配が完了した分が減っていきます。そのため、在庫がなくなった場合に、在庫数を元に戻すためには、Cosminexus サーバの再起動、または J2EE プロジェクトのリデプロイのどちらかを実施します。Cosminexus サーバの再起動、または J2EE プロジェクトのリデプロイの方法を次に示します。

## Cosminexus サーバの再起動方法

1. [ MyEclipse Java エンタープライズ開発 ] のパースペクティブを開きます。
2. [ サーバー ] ビューから [ Cosminexus 7 ] サーバを右クリックして、[ リスタート ] を選択します。

## J2EE プロジェクトのリデプロイ方法

1. [ MyEclipse Java エンタープライズ開発 ] のパースペクティブを開きます。
2. [ サーバー ] ビューの [ Cosminexus 7 ] の配下にあるプロジェクトを右クリックして、[ リデプロイ ] を選択します。

## (3) プロジェクトを削除しましょう

サンプルプログラムを動かして確認できたら、プロジェクトを削除しましょう。削除するプロジェクトは、サンプルプログラムごとに異なります。ここでは、サンプルプログラム ( HelloServiceAdapter サンプルプログラム ) の例で説明しています。

<手順>

1. コマンドプロンプトを開きます。
2. 受付を停止します。  
`cscrcptnstop -user admin -pass admin -csc MyCSC`
3. HCSC コンポーネントを停止します。  
`csccompstop -user admin -pass admin -csc MyCSC -all`
4. HCSC サーバへ配備した HCSC コンポーネントを削除します。  
`csccompounddeploy -user admin -pass admin -csc MyCSC -all`

## • 参考

HCSC サーバへ配備した HCSC コンポーネントをすべて削除した場合でも、HCSC-Manager の画面には、HCSC コンポーネントが表示されます。この場合は、ほかの HCSC コンポーネントを配備すると削除されるため、問題ありません。

5. メニューから [ ウィンドウ ] - [ パースペクティブを開く ] を選択し、[ MyEclipse Java エンタープライズ開発 ] を選択します。
6. [ OK ] ボタンをクリックします。  
[ MyEclipse Java エンタープライズ開発 ] パースペクティブが表示されます。
7. [ サーバー ] ビューを開き、[ Cosminexus 7 ] を選択し右クリックして、[ デプロイメント管理 ] を選択します。

[ サーバー・デプロイメント ] ダイアログが表示されます。

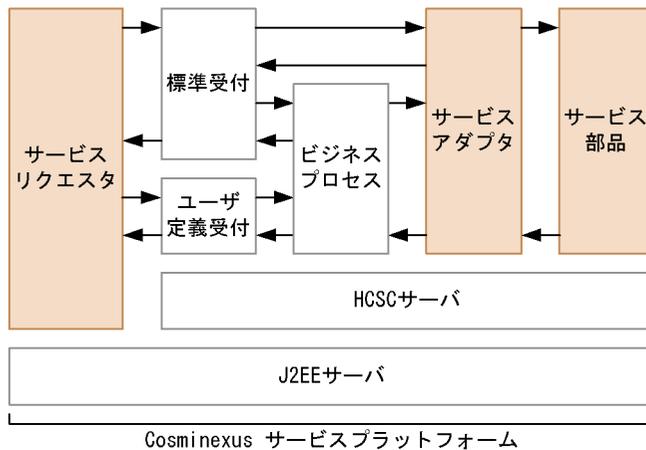
8. [ HelloClient ] プロジェクトを選択して [ 除去 ] ボタンをクリックします。
9. 同様に, [ HelloService ] プロジェクトを選択して [ 除去 ] ボタンをクリックします。
10. [ MyEclipse Java エンタープライズ開発 ] パースペクティブのビューから次の四つのプロジェクトを選択し, 右クリックして削除します。
  - HelloClient
  - HelloClientWeb
  - HelloService
  - HelloServiceWeb

サンプルプログラムをすべて削除したあとに, ほかのサンプルプログラムを動かす場合は, MyEclipse プロジェクトのインポートから始めてください。詳細は, 「付録 C.1(1)(c) MyEclipse プロジェクトをインポートします」を参照してください。

## 付録 C.2 サンプルプログラムを構成している要素

サンプルプログラムで提供している範囲を図 C-2, 図 C-3, および図 C-4 に示します。

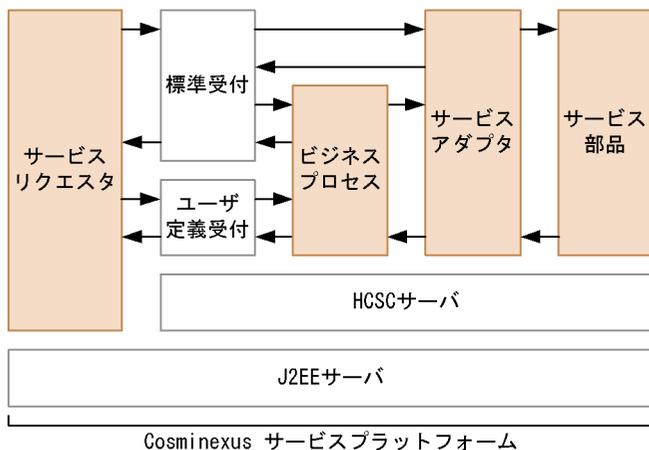
図 C-2 HelloServiceAdapter サンプルプログラムで提供している範囲



(凡例)

- : サービス部品を呼び出す要求・応答の流れ
- : サンプルプログラムの提供範囲

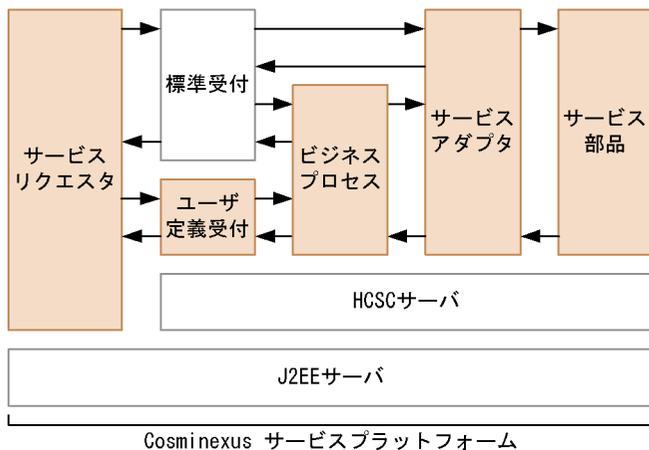
図 C-3 HelloBusinessProcess サンプルプログラムで提供している範囲



(凡例)

- : サービス部品を呼び出す要求・応答の流れ
- : サンプルプログラムの提供範囲

図 C-4 商品手配サンプルプログラムで提供している範囲



(凡例)

- : サービス部品を呼び出す要求・応答の流れ
- : サンプルプログラムの提供範囲

これらのサンプルプログラムが動くためには、どのような要素で構成されているのか、どのような役割をしているかを説明します。

サンプルプログラムは、次に示す要素で構成されています。

- サービスリクエスタ

サービス部品を利用するアプリケーションプログラムのことです。サービス部品を呼び出す要求をします。要求するときに使用する電文が要求電文です。

- HCSC サーバ  
サービス部品を実行するために、サービスアダプタやビジネスプロセスを管理するサーバ機能です。Cosminexus サービスプラットフォームの中にあります。
- 標準受付  
サービスリクエストからの要求電文を受け付けるための機能（インターフェース）です。標準で用意されています。
- ユーザ定義受付  
サービスリクエストからの実行要求を受け付けるための機能（インターフェース）です。HCSC サーバに含まれる標準受付とは異なり、ユーザが任意のインターフェースを定義できます。  
商品手配サンプルプログラムにだけ関係します。
- J2EE サーバ  
J2EE アプリケーション（JSP、サーブレット、EnterpriseBeanなどで構成されるアプリケーション）を実行するためのサーバ機能です。
- ビジネスプロセス  
複数のサービス部品の処理の順番や条件などを定義して、一連の業務の流れとして定義したものです。
- サービスアダプタ  
サービスリクエストやビジネスプロセスからの要求を受け付けて、サービス部品を呼び出すためのものです。HCSC サーバの中にあります。サービス部品を呼び出した結果は、応答電文としてビジネスプロセスまたはサービスリクエストに返します。
- サービス部品  
業務プログラムのことです。サービスリクエストから要求のあった内容を実行します。

サンプルプログラムでは、あらかじめ用意されていますが、実際には、次のものをユーザが作成します。

- サービスリクエスト
- ユーザ定義受付
- ビジネスプロセス
- サービスアダプタ
- サービス部品

## 付録 C.3 定義してみましよう

それでは、実際に用意されているサンプルプログラムを使って、サービスアダプタ、ビジネスプロセスなどを定義してみましよう。

各サンプルプログラムの目的を次に示します。

### (1)HelloServiceAdapter サンプルプログラム

サービスアダプタの定義を習得することを目的にしています。

(2)HelloBusinessProcess サンプルプログラム

ビジネスプロセスの定義を習得することを目的にしています。

(3) 商品手配サンプルプログラム

実際の業務に近い内容での定義を習得することを目的にしています。

( 1 ) HelloServiceAdapter サンプルプログラム

サービスリクエスタからサービスアダプタを呼び出す，HelloServiceAdapter サンプルプログラムを使って，サービスアダプタを定義してみましょう。

HelloServiceAdapter サンプルプログラムの概要を図 C-5 に，HelloServiceAdapter サンプルプログラムのコンポーネントを表 C-1 に示します。

図 C-5 HelloServiceAdapter サンプルプログラムの概要

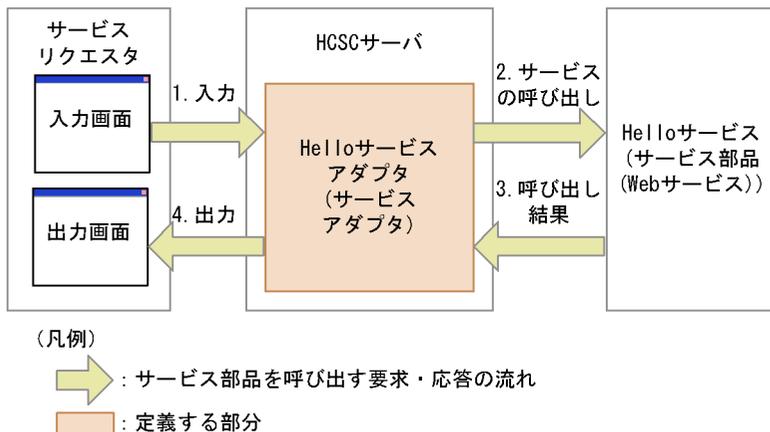


表 C-1 HelloServiceAdapter サンプルプログラムのコンポーネント

コンポーネント	形態	内容
サービスリクエスタ	HTML/ サブレット	入力画面を表示しサービスアダプタを呼び出します。Hello サービスを呼び出した結果をHello サービスアダプタから受け取り，画面に表示します。
Hello サービスアダプタ	サービスアダプタ	Hello サービスを呼び出すためのサービスアダプタです。Hello サービスを呼び出して，サービスリクエスタに応答を返します。
Hello サービス	Web サービス	入力された文字列を編集して返します。

説明

1. サービスリクエスタから文字列を入力します。
2. サービスアダプタから，Web サービスが呼び出されます。
3. Web サービスから編集された文字列が戻り値として取得されます。

## 4. 文字列が画面に表示されます。

HelloServiceAdapter サンプルプログラムでは、次に示すコンポーネントを定義します。

## (a)Hello サービスアダプタ

## ( a ) Hello サービスアダプタを定義してみましょう

サービスアダプタは、呼び出すサービス部品に合わせて作成します。

HelloServiceAdapter サンプルプログラムを使って、Hello サービスアダプタを定義するときに設定する値を次の表に示します。

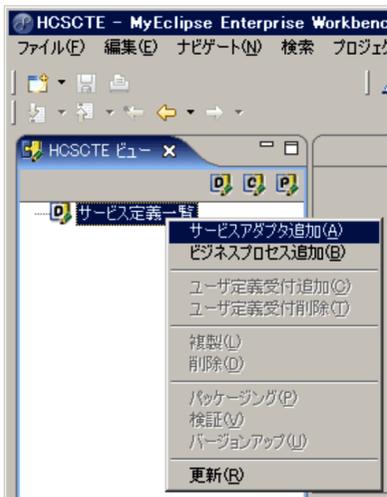
表 C-2 Hello サービスアダプタを定義するときに設定する値

項目名	設定する値	説明
サービス部品の種別	Web サービス	呼び出すサービス部品の種別に合わせて設定したサービスアダプタの種別です。このサンプルプログラムのサービス部品は、Web サービスのためサービスアダプタの種別も Web サービスを設定します。
サービス名	Hello サービスアダプタ	アダプタの名称を指定します。
WSDL ファイル	HelloService.wsdl	WSDL には、Web サービスがどのような機能を持ち、それを利用するためにはどのような要求をすればよいのか、などを記述する方法が定義されています。Hello サービスアダプタは、Hello サービス用 WSDL ( HelloService.wsdl ) を使用して作成します。
Port の指定	Hello	Hello サービスアダプタのポートを指定します。
サービス ID	HelAdp	Hello サービスアダプタの ID を指定します。
クライアント定義ファイル	c4webcl.properties ( 次の内容が設定されています。 「c4web.logger.log_file_prefix=HelloService」)	クライアント定義ファイルは、クライアント側の動作を制御するファイルで、「c4webcl.properties」というファイル名称で利用者が作成します。 このサンプルプログラムでは、トレースファイル、およびアプリケーションログのプレフィクスとして、「HelloService」を設定するクライアント定義ファイルが用意されています。

注 各ファイルの格納場所については、「付録 C.4(a) HelloServiceAdapter サンプルプログラムの構成」を参照してください。

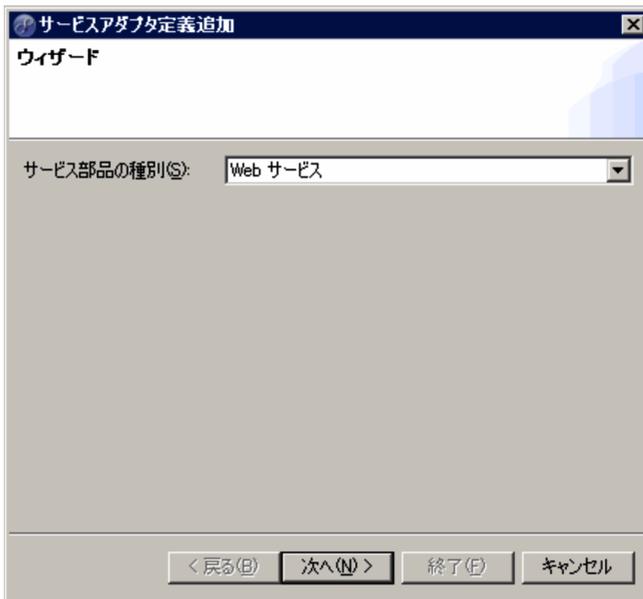
Hello サービスアダプタの追加と定義手順を次に示します。

1. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。



追加するサービスアダプタから利用するサービス種別を設定するダイアログが表示されます。

2. ドロップダウンリストから「Web サービス」を選択して、[次へ] ボタンをクリックします。



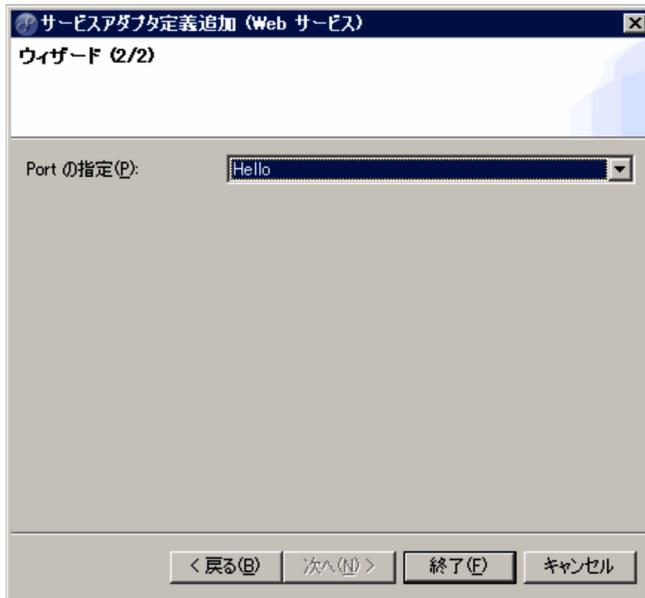
サービスアダプタの追加に必要な情報を入力するダイアログが表示されます。

3. サービス名に「Hello サービスアダプタ」を、WSDL ファイル名には「HelloService.wsdl」を入力して、[次へ] ボタンをクリックします。



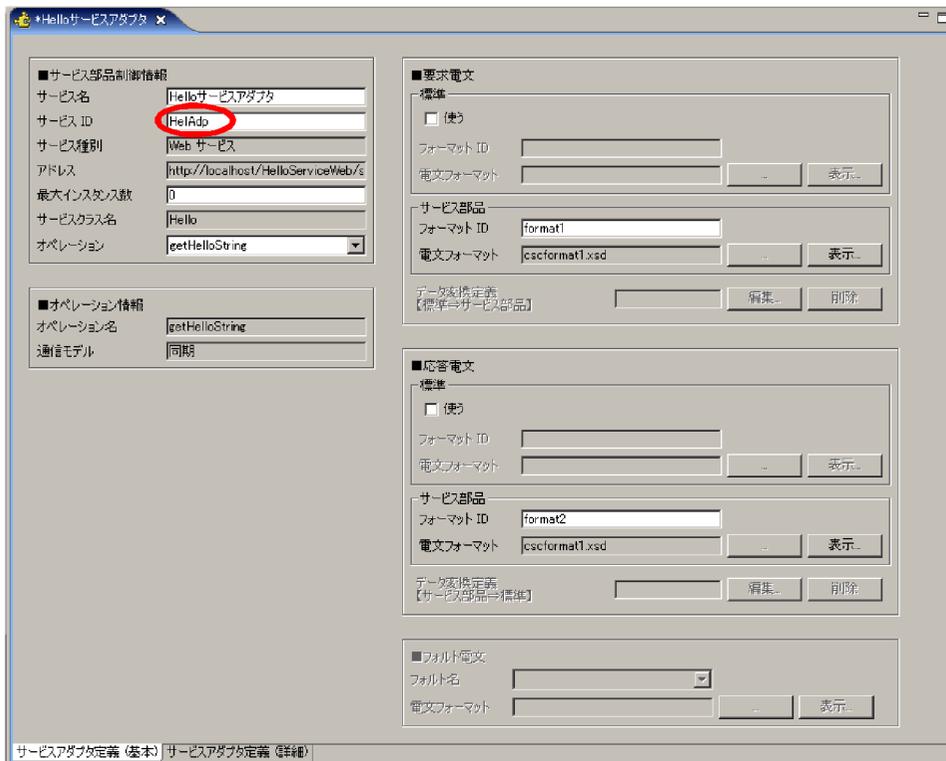
ポートを指定するダイアログが表示されます。

4. ドロップダウンリストでポート「Hello」を選択して、[ 終了 ] ボタンをクリックします。

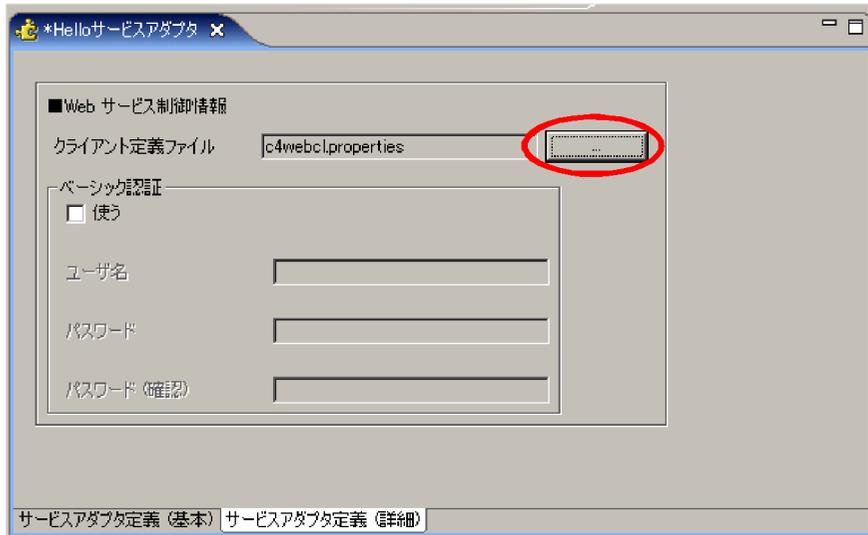


アダプタ「Hello サービスアダプタ」が作成され、サービスアダプタ定義画面が表示されます。

5. サービスアダプタ定義画面の基本画面で、サービス ID を「HelAdp」に変更します。



6. サービスアダプタ定義画面の詳細画面で, [ ... ] ボタンをクリックして, クライアント定義ファイル「c4webcl.properties」を指定します。  
 HelloServiceAdapter サンプルプログラムでは, 次の場所にあります。  
 <Cosminexus インストールディレクトリ  
 >¥CSCTE¥Samples¥HelloServiceAdapter¥Service¥HelloService¥c4webcl.properties

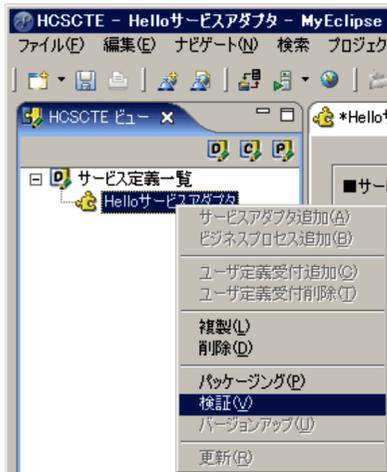


7. [ファイル] - [保管] を選択します。

(b) Hello サービスアダプタの検証とパッケージングをしてみましょう

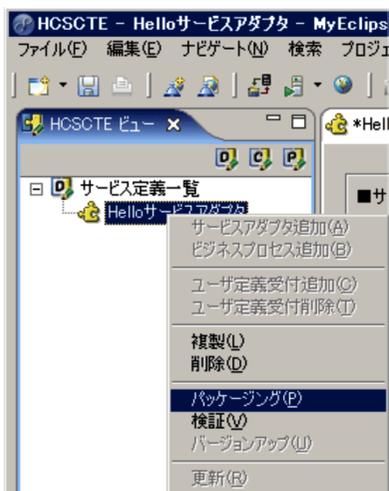
作成した Hello サービスアダプタは、正しいかどうか検証してから、パッケージングします。検証とパッケージングの手順を次に示します。

1. ツリービューのサービス定義一覧で Hello サービスアダプタを選択し、右クリックして、「検証」を選択します。



検証結果がコンソールビューに表示されます。エラーが発生した場合は、メッセージに従って修正します。

2. ツリービューのサービス定義一覧で、「Hello サービスアダプタ」を選択し、右クリックして、「パッケージング」を選択します。



メッセージダイアログが表示されます。

3. 次のどちらかの操作をします。  
パッケージングが成功した場合は、[ OK ] ボタンをクリックします。  
パッケージングが失敗した場合は、ダイアログのメッセージに従って対処し、パッケージングを再実行します。  
これでパッケージングが終了しました。

## (2) HelloBusinessProcess サンプルプログラム

ビジネスプロセスからサービスアダプタを呼び出す、HelloBusinessProcess サンプルプログラムを使って、ビジネスプロセスを定義してみましょう。

HelloBusinessProcess サンプルプログラムの概要を図 C-6 に、HelloBusinessProcess サンプルプログラムのコンポーネントを表 C-3 に示します。

図 C-6 HelloBusinessProcess サンプルプログラムの概要

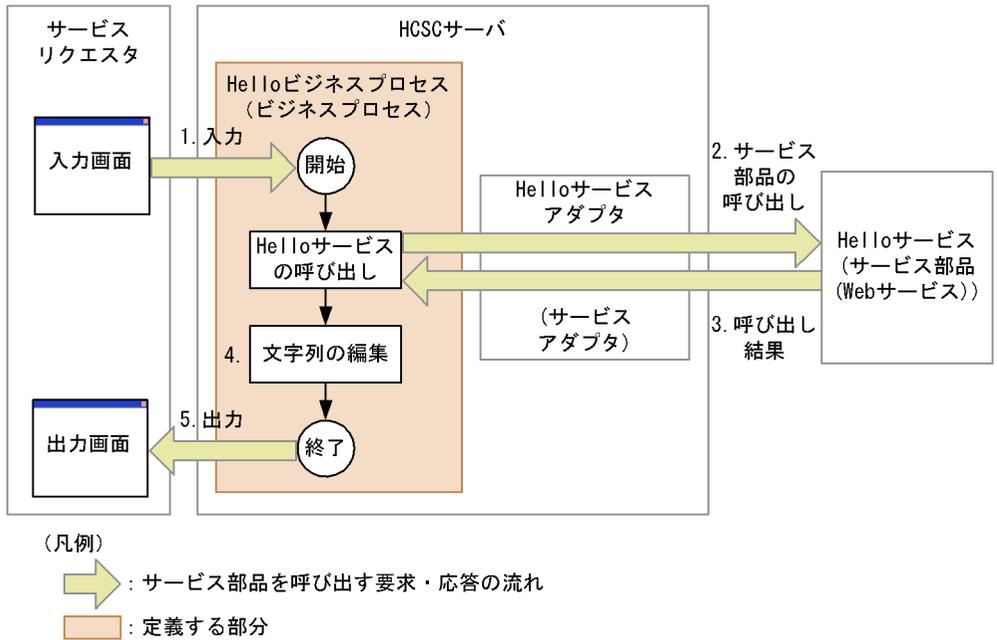


表 C-3 HelloBusinessProcess サンプルプログラムのコンポーネント

コンポーネント	形態	内容
サービスリクエスタ	HTML/ サブレット	入力画面を表示しビジネスプロセスを呼び出します。Hello サービスを呼び出したビジネスプロセスの結果も画面に表示します。
Hello ビジネスプロセス	サービスアダプタ / ビジネスプロセス	サービスリクエスタからの要求を受け取り、Web サービスを呼び出してサービスリクエスタに応答を返します。
Hello サービス	Web サービス	入力された文字列を編集して返します。

### 説明

1. サービスリクエスタから文字列を入力します。

2. ビジネスプロセスからサービスアダプタを介して、Web サービスが呼び出されます。
3. Web サービスから編集された文字列が戻り値として取得されます。
4. 戻り値の文字列が編集されます。
5. ビジネスプロセスで編集された文字列が戻り値として取得され、画面に表示されま  
す。

HelloBusinessProcess サンプルプログラムでは、次に示すコンポーネントを定義します。

(a)Hello ビジネスプロセス

Hello サービスアダプタの定義については、「付録 C.3(1)(a) Hello サービスアダプタを定義してみましょう」を参照してください。

(a) Hello ビジネスプロセスを定義してみましょう

ビジネスプロセスは、どのようなプロセスで処理するかによって、定義する内容が異なります。HelloBusinessProcess サンプルプログラムの Hello ビジネスプロセスでは、次のようなプロセスで処理します。

1. Hello ビジネスプロセスは、サービスリクエストから入力された文字列を受け取りま  
す。
2. Hello サービスアダプタを介して Hello サービスを呼び出します。
3. Hello サービスを呼び出した結果、取得した文字列に、「 and Business  
Process」( は半角空白)という文字列を連結します。

出力文字列: <Hello サービスの出力文字列> and Business Process
--

4. サービスリクエストに文字を連結した結果を返し表示します。

HelloBusinessProcess サンプルプログラムのビジネスプロセスは、次の流れで定義しま  
す。

1. 新規ビジネスプロセスを追加します
2. 変数を設定します
3. アクティビティ を配置します
4. アクティビティ を定義します
5. ビジネスプロセスの定義を終了します

注 アクティビティは、ビジネスプロセスの処理の流れを定義する部品です。

新規ビジネスプロセスを追加します

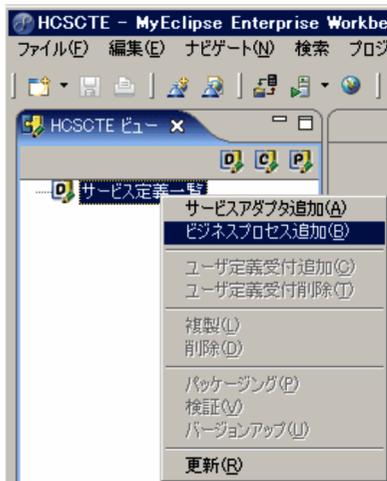
HelloBusinessProcess サンプルプログラムを使って、ビジネスプロセスを追加する  
ときに設定する値を次の表に示します。

表 C-4 ビジネスプロセスを追加するときに設定する値

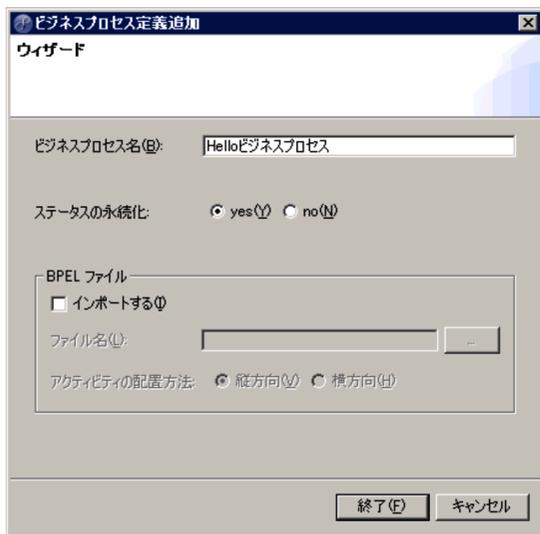
項目名	設定する値	説明
ビジネスプロセス名	Hello ビジネスプロセス	ビジネスプロセスの名称を指定します。
ステータスの永続化	yes	データベースに記録を残すかどうかを指定します。記録をとると、プロセスの進捗状況などを把握できます。このサンプルプログラムでは、データベースに記録を残すため、「yes」を選択します。
BPEL ファイルインポートする	チェックを外します	上流工程でツールを使って作成した BPEL をインポートする場合は、チェックを入れます。インポートすると、ビジネスプロセスに必要なアクティビティが自動的に表示されます。このサンプルプログラムでは、インポートしないので、チェックを外します。
サービス ID	HelBP	ビジネスプロセスの ID を指定します。

Hello ビジネスプロセスの追加手順を次に示します。

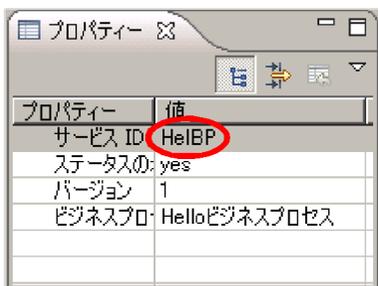
- ツリービューのサービス定義一覧で、右クリックして [ ビジネスプロセス追加 ] を選択します。  
ビジネスプロセス定義を追加するためのダイアログが表示されます。



- [ ビジネスプロセス名 ] に「Hello ビジネスプロセス」を入力し、[ ステータスの永続化 ] で「yes」を選択します。[ BPEL ファイル ] の [ インポートする ] はチェックを外します。



3. [ 終了 ] ボタンをクリックします。  
 ビジネスプロセス「Hello ビジネスプロセス」が作成され、ビジネスプロセス定義画面が表示されます。
4. ビジネスプロセス定義画面のプロパティービューで、サービス ID を「HelBP」に変更します。



変数を設定します

ビジネスプロセスでは、アクティビティを定義するときに変数を使用します。そのため、使用する変数をあらかじめ設定してから、アクティビティを定義します。Hello ビジネスプロセスで使用する変数を次の表に示します。

表 C-5 Hello ビジネスプロセスで使用する変数

変数名	種別	xsd ファイル
入力データ	messageType	入力データ .xsd
出力データ	messageType	出力データ .xsd

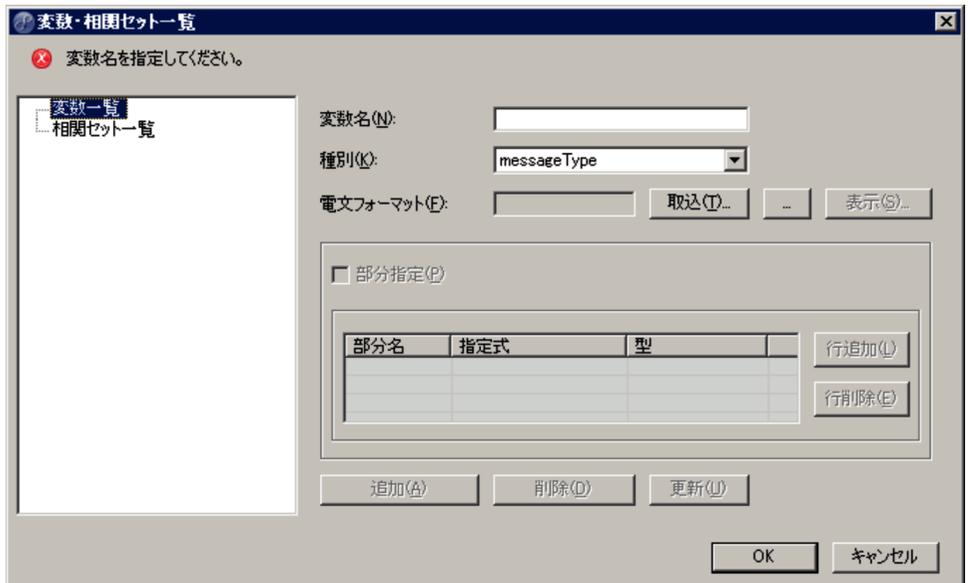
Hello ビジネスプロセスで使用する変数の設定手順を次に示します。

1. ビジネスプロセス定義画面のキャンパス上の [ 変数・相関セット ] アイコンをダブルクリックします。

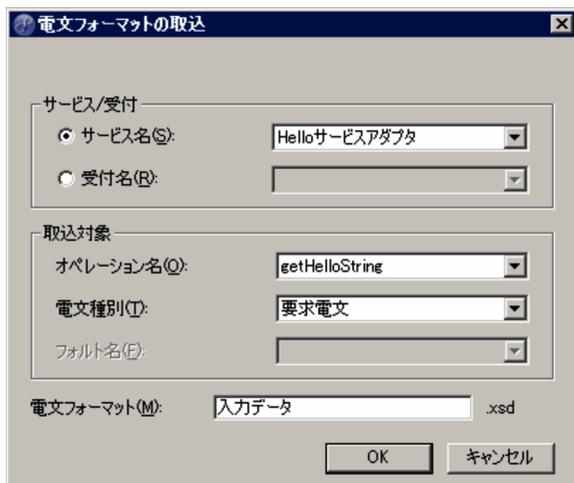


[ 変数・相関セット一覧 ] ダイアログが表示されます。

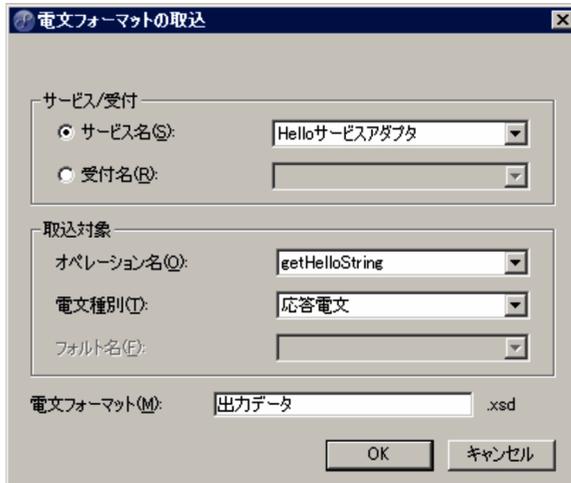
2. [ 変数一覧 ] を選択します。変数名に「入力データ」を入力し、種別は、ドロップダウンリストから「messageType」を選択します。



3. [ 取込 ] ボタンをクリックします。  
[ 電文フォーマットの取込 ] ダイアログが表示されます。
4. [ サービス名 ] を選択して、ドロップダウンリストから「Hello サービスアダプタ」を選択します。
5. [ オペレーション名 ] はドロップダウンリストから「getHelloString」を、[ 電文種別 ] はドロップダウンリストから「要求電文」を選択します。[ 電文フォーマット ] に「入力データ」を入力します。



6. [ OK ] ボタンをクリックします。  
[ 電文フォーマットの取込 ] ダイアログが閉じます。
7. [ 変数・相関セット一覧 ] ダイアログの [ 追加 ] ボタンをクリックします。  
[ 変数一覧 ] に、「入力データ」が追加されます。
8. [ 変数・相関セット一覧 ] ダイアログの [ 変数一覧 ] を選択します。変数名に「出力データ」を入力し、種別は、ドロップダウンリストから「messageType」を選択します。
9. [ 取込 ] ボタンをクリックします。  
[ 電文フォーマットの取込 ] ダイアログが表示されます。
10. [ サービス名 ] を選択して、ドロップダウンリストから「Hello サービスアダプタ」を選択します。
11. オペレーション名はドロップダウンリストから「getHelloString」を、電文種別はドロップダウンリストから「応答電文」を選択します。電文フォーマットに「出力データ」を入力します。



12.[ OK ] ボタンをクリックします。

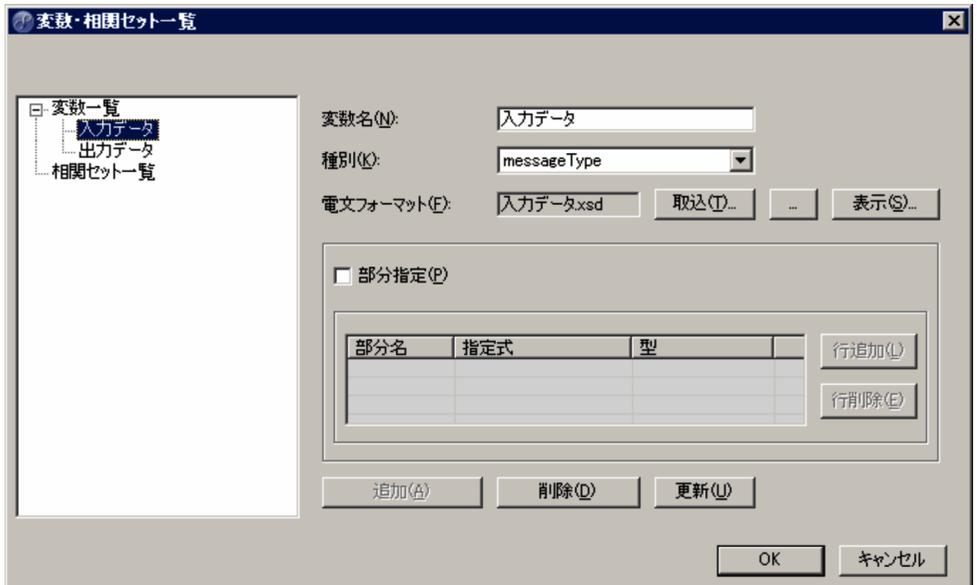
[ 電文フォーマットの取込 ] ダイアログが閉じます。

13.[ 変数・相関セット一覧 ] ダイアログの [ 追加 ] ボタンをクリックします。

[ 変数一覧 ] に、「出力データ」が追加されます。

14.[ 変数・相関セット一覧 ] ダイアログの [ OK ] ボタンをクリックします。

これで変数が設定できました。



アクティビティを配置します

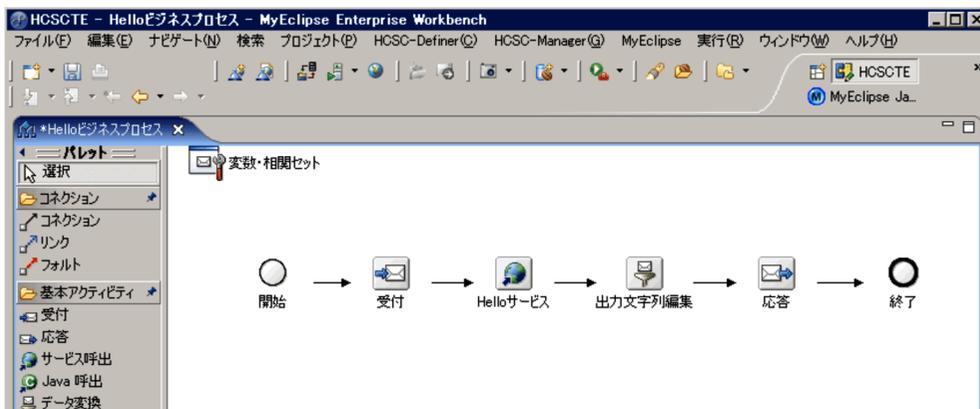
HelloBusinessProcess サンプルプログラムのビジネスプロセスに必要なアクティビティを次の表に示します。

表 C-6 HelloBusinessProcess サンプルプログラムのビジネスプロセスに必要なアクティビティ

設定する値	説明
受付アクティビティ	サービスリクエストからの応答を受け付けます。
サービス呼出アクティビティ	Hello サービスを呼び出します。
データ変換アクティビティ	文字列を編集します。
応答アクティビティ	処理結果をサービスリクエストへ返します。

アクティビティの配置手順を次に示します。

1. パレットから次のアクティビティをクリックし、キャンバス上の適当な位置をクリックして配置します。
  - 受付
  - サービス呼出
  - データ変換
  - 応答
2. アクティビティを連結するために、パレットの [ コネクション ] をクリックします。
3. 連結元である開始アクティビティをクリックします。
4. 連結先である受付アクティビティをクリックします。  
これで開始アクティビティと受付アクティビティを連結できました。
5. 受付アクティビティから終了アクティビティまでを、手順 2. ~ 4. と同様の操作で一つずつアクティビティを連結します。



アクティビティを定義します

キャンバスへ配置した各アクティビティの内容を定義します。

#### < 受付アクティビティ >

1. キャンバスの受付アクティビティをダブルクリックします。  
[ 受付アクティビティ ] ダイアログが表示されます。
2. 次の内容を入力します。

項目名	設定する値	説明
アクティビティ名	受付	アクティビティの名称を指定します。
オペレーション名	getHelloString	サービスリクエストからサービス部品を呼び出すときに利用するオペレーションの名称を指定します。
割当変数	入力データ	ビジネスプロセスの要求電文に割り当てる変数をドロップダウンリストから選択します。
割当関連セット群	なし	関連セットグループをアクティビティに割り当てる場合に設定します。このサンプルプログラムでは使用しないため、設定しません。
通信モデル	同期	オペレーションの通信モデルを指定します。このサンプルプログラムで使用する Hello サービスは、Web サービスのため、「同期」を設定します。
インスタンス生成	yes	要求電文を受け付けたときに、プロセスを初期化するかどうかを選択します。このサンプルプログラムでは、「yes」を設定します。

3. [ OK ] ボタンをクリックします。

#### < サービス呼出アクティビティ >

1. キャンバスのサービス呼出アクティビティをダブルクリックします。  
[ サービス呼出アクティビティ ] ダイアログが表示されます。

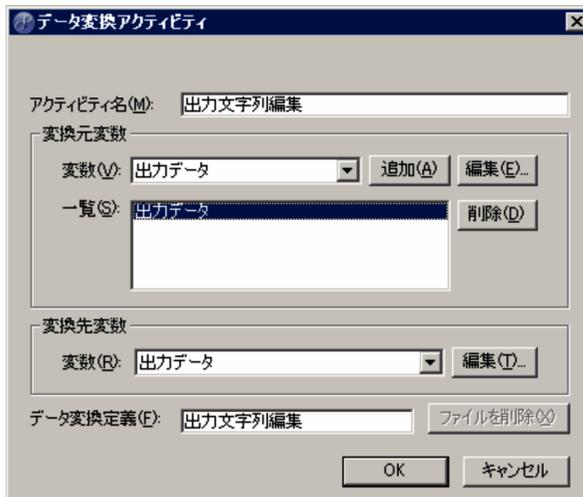
2. 次の内容を入力します。

項目名	設定する値	説明
アクティビティ名	Hello サービス	アクティビティの名称を入力します。
サービス名	Hello サービスアダプタ	要求電文を送信して呼び出すサービス部品の名称をドロップダウンリストから選択します。
オペレーション名	getHelloString	[ サービス名 ] で指定したサービス部品 (Hello サービスアダプタ) のオペレーションのうち、実際に呼び出すオペレーションの名称を指定します。
通信モデル	同期	[ オペレーション名 ] で指定したオペレーションに設定されている通信モデルが表示されます。
要求電文用割当変数	入力データ	在庫管理サービスを呼び出す要求電文に割り当てる変数をドロップダウンリストから選択します。
応答電文用割当変数	出力データ	同期オペレーションから受け取る応答電文に割り当てる変数をドロップダウンリストから選択します。
割当関連セット群	なし	関連セットグループをアクティビティに割り当てる場合に設定します。このサンプルプログラムでは使用しないため、設定しません。

3. [ OK ] ボタンをクリックします。

< データ変換アクティビティ >

1. キャンパスのデータ変換アクティビティをダブルクリックします。  
[ データ変換アクティビティ ] ダイアログが表示されます。
2. 次の内容を入力します。

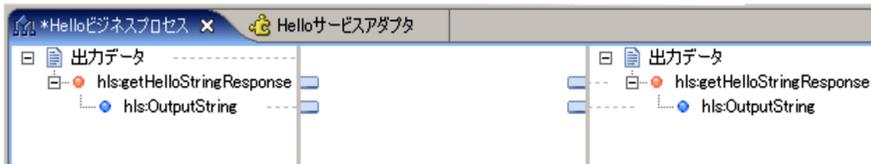


項目名	設定する値	説明
アクティビティ名	出力文字列編集	アクティビティの名称を指定します。
変数（変換元）	出力データ	データの変換元になる変数をドロップダウンリストから選択し、[追加] ボタンをクリックします。
変数（変換先）	出力データ	データの変換先になる変数をドロップダウンリストから選択します。
データ変換定義	出力文字列編集	データ変換定義ファイルに仮想的に付ける名称を指定します。

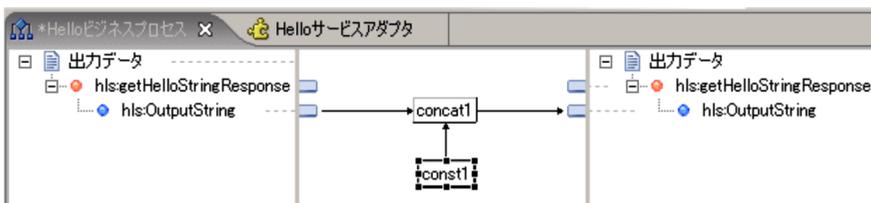
- [ OK ] ボタンをクリックします。
- キャンパスのデータ変換アクティビティを右クリックして、[ マッピング定義起動 ] を選択します。  
[ ルート要素選択 ] ダイアログが表示されます。
- [ 変換元 ] のスキーマ論理名「出力データ」のルート要素として、ドロップダウンリストから「hls:getHelloStringResponse」を選択し、[ 変換先 ] のスキーマ論理名「出力データ」のルート要素として、ドロップダウンリストから「hls:getHelloStringResponse」を選択します。



6. [ OK ] ボタンをクリックします。  
データ変換定義画面が表示されます。



7. データ変換定義画面のパレットから [ concat ] をクリックし、キャンパスの変換元と変換先の間をクリックして配置します。
8. データ変換定義画面のパレットから [ const ] をクリックし、キャンパスの変換元と変換先の間をクリックして配置します。
9. データ変換定義画面のパレットから [ マッピング ] を選択します。
10. マッピング元となる変換元ノードのノードアダプタをクリックします。
11. マッピング先となる [ concat ] をクリックします。  
マッピング線が設定されます。
12. 手順 11. ~ 12. と同様の操作で、[ concat ] から変換先ノードのノードアダプタにマッピング線を設定します。  
[ const ] から [ concat ] へのマッピング線を先に設定すると、出力文字列の順序が変わってしまうため、注意してください。
13. 手順 11. ~ 12. と同様の操作で、[ const ] から [ concat ] にマッピング線を設定します。



14. データ変換定義画面のパレットから [ 選択 ] をクリックします。
15. [ const ] をダブルクリックします。  
[ 定数 ] ダイアログが表示されます。
16. [ 文字列 ] を選択して、「 and Business Process」を入力します（ は半角の空白です）。



17.[ OK ] ボタンをクリックします。

#### < 応答アクティビティ >

- キャンパスの応答アクティビティをダブルクリックします。  
[ 応答アクティビティ ] ダイアログが表示されます。
- 次の内容を入力します。



項目名	設定する値	説明
アクティビティ名	応答	アクティビティの名称を指定します。
オペレーション名	getHelloString	対応する受付アクティビティに指定したオペレーションの名称を指定します。
割当変数	出力データ	応答電文に割り当てる変数をドロップダウンリストから選択します。
割当関連セット群	なし	関連セットグループをアクティビティに割り当てる場合に設定します。このサンプルプログラムでは使用しないため、設定しません。

項目名	設定する値	説明
フォルト名	なし	フォルト処理として応答アクティビティを定義して、サービスリクエストにフォルトが発生したことを示す応答電文を送信する場合のフォルトの名称を指定します。このサンプルプログラムではフォルト処理を使用しないため、設定しません。

3. [ OK ] ボタンをクリックします。

ビジネスプロセスの定義を終了します

すべてのアクティビティを定義したら、[ ファイル ] - [ 保管 ] を選択して、ビジネスプロセスの定義を終了します。

(b) Hello ビジネスプロセスの検証とパッケージングをしてみましょう

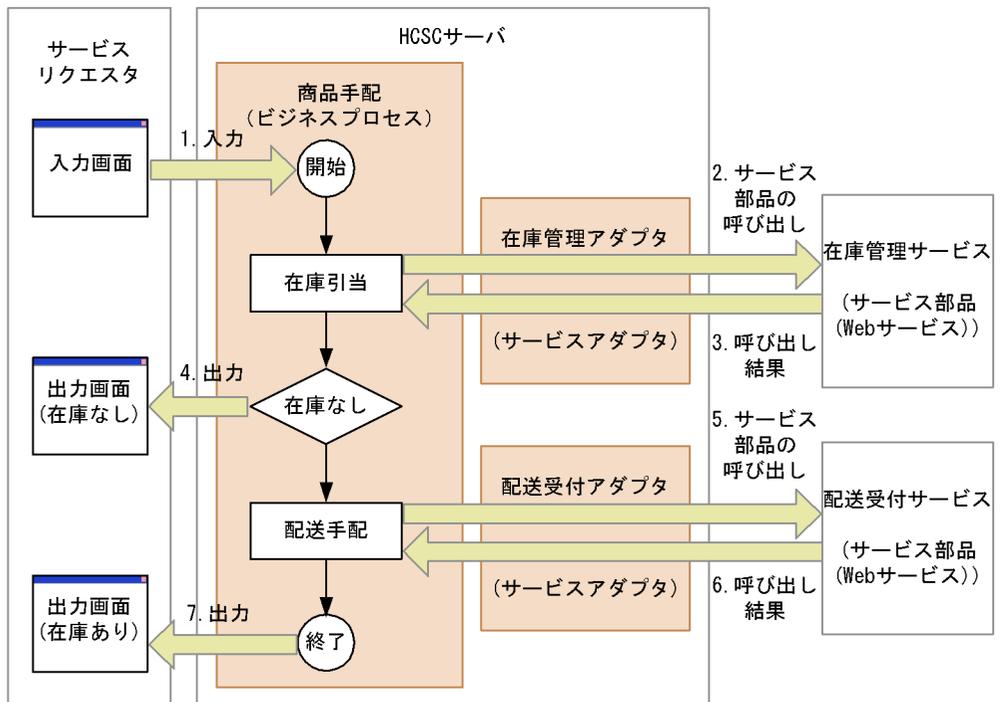
作成したビジネスプロセスは、正しいかどうか検証してから、パッケージングします。検証とパッケージングの方法は、Hello サービスアダプタと同じです。検証とパッケージングの詳細については、「付録 C.3(1)(b) Hello サービスアダプタの検証とパッケージングをしてみましょう」を参照してください。

### (3) 商品手配のサンプルプログラム

実際の業務に近い内容のビジネスプロセスからサービスアダプタを呼び出す、商品手配サンプルプログラムを使って、ビジネスプロセスを定義してみましょう。

ここで紹介する商品手配のサンプルプログラムは、店舗の担当者が業務システムを使って商品の手配をする場合に、在庫引当や配送手配の処理をすることを想定しています。商品手配サンプルプログラムの概要を図 C-7 に、商品手配サンプルプログラムのコンポーネントを表 C-7 に示します。

図 C-7 商品手配サンプルプログラムの概要



(凡例)

- : サービス部品を呼び出す要求・応答の流れ
- : 定義する部分

表 C-7 商品手配サンプルプログラムのコンポーネント

コンポーネント	形態	内容
サービスリクエスタ	HTML/ サブレット /JSP	入力画面を表示しビジネスプロセスを呼び出します。サービス呼び出したビジネスプロセスの結果も画面に表示します。

コンポーネント	形態	内容
商品手配 / 在庫管理・配送受付	サービスアダプタ / ビジネスプロセス	サービスリクエストからの要求を受け取り、在庫管理サービスや配送受付サービスを呼び出して、サービスリクエストに応答を返します。
在庫管理サービス	Web サービス	商品の在庫を引き当て、引当番号を返します。
配送受付サービス	Web サービス	配送を手配し、配送番号を返します。

#### 説明

1. サービスリクエストから商品名および個数を入力すると、商品手配ビジネスプロセスが呼び出されます。
2. ビジネスプロセスから、在庫管理サービスアダプタを介して在庫管理サービスが呼び出されます。
3. ビジネスプロセスには、在庫管理サービスで在庫を引き当てた結果が、引当番号として取得されます。
4. 在庫管理サービスの呼び出し結果から、ビジネスプロセスでは、在庫がない場合は「\*」(在庫無し)をサービスリクエストに返します。
5. 在庫がある場合は、ビジネスプロセスから配送受付サービスアダプタを介して配送受付サービスが呼び出されます。
6. ビジネスプロセスには、配送受付サービスで配送手配した結果が、配送番号として取得されます。
7. 配送受付サービスの呼び出し結果から、配送番号をサービスリクエストに返します。

商品手配サンプルプログラムでは、次に示す三つのコンポーネントを定義します。

(a) 在庫管理サービスアダプタ

(b) 配送受付サービスアダプタ

(c) 商品手配ビジネスプロセス

(a) 在庫管理サービスアダプタを定義してみましょう

在庫管理サービスアダプタは、在庫管理サービス用 WSDL

(InventoryManagementService.wsdl) を使用して定義します。在庫管理サービスアダプタを定義するときに設定する値を次の表に示します。

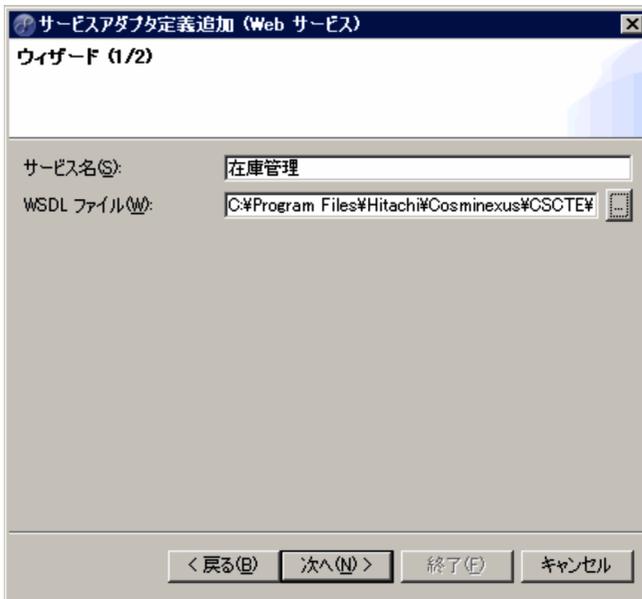
表 C-8 在庫管理サービスアダプタを定義するときに設定する値

項目名	設定する値	説明
サービス部品の種別	Web サービス	呼び出すサービス部品の種別に合わせて設定したサービスアダプタの種別です。このサンプルプログラムのサービス部品は、Web サービスのためサービスアダプタの種別も Web サービスを設定します。
サービス名	在庫管理	アダプタの名称を指定します。
WSDL ファイル	InventoryManagementService.wsdl	WSDL には、Web サービスがどのような機能を持ち、それを利用するためにはどのような要求をすればよいのか、などを記述する方法が定義されています。在庫管理サービスアダプタは、在庫管理サービス用 WSDL ( InventoryManagementService.wsdl ) を使用して作成します。
Port の指定	InventoryManager	在庫管理サービスアダプタのポートを指定します。
サービス ID	InvAdp	在庫管理サービスアダプタの ID を指定します。
クライアント定義ファイル	c4webcl.properties ( 次の内容が設定されています。 「c4webcl.logger.log_file_prefix=InventoryManagementService」)	クライアント定義ファイルは、クライアント側の動作を制御するファイルで、「c4webcl.properties」というファイル名称で利用者が作成します。 このサンプルプログラムでは、トレースファイル、およびアプリケーションログのプレフィクスとして、「InventoryManagementService」を設定する、クライアント定義ファイルが用意されています。

注 各ファイルの格納場所については、「付録 C.4(c) 商品手配サンプルプログラムの構成」を参照してください。

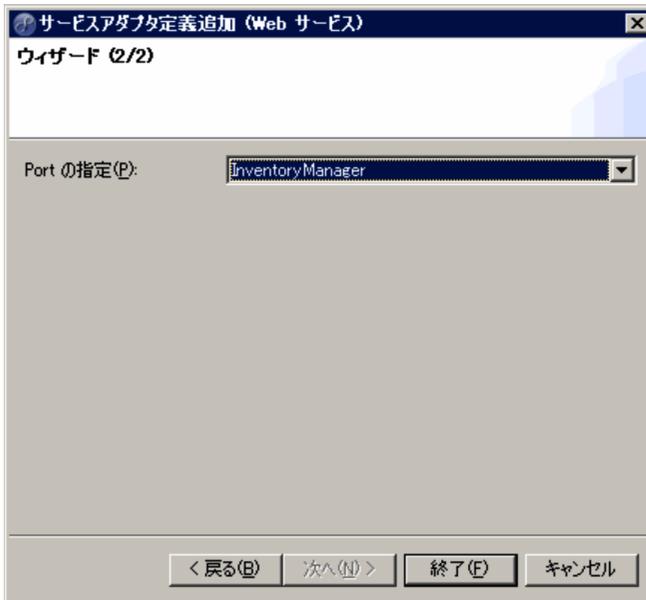
在庫管理サービスアダプタの追加と定義手順を次に示します。

1. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。  
追加するサービスアダプタから利用するサービス種別を設定するダイアログが表示されます。
2. ドロップダウンリストから「Web サービス」を選択して、「[ 次へ ]」ボタンをクリックします。  
サービスアダプタの追加に必要な情報を入力するダイアログが表示されます。
3. サービス名に「在庫管理」を、WSDL ファイルには「InventoryManagementService.wsdl」を入力して、「[ 次へ ]」ボタンをクリックします。



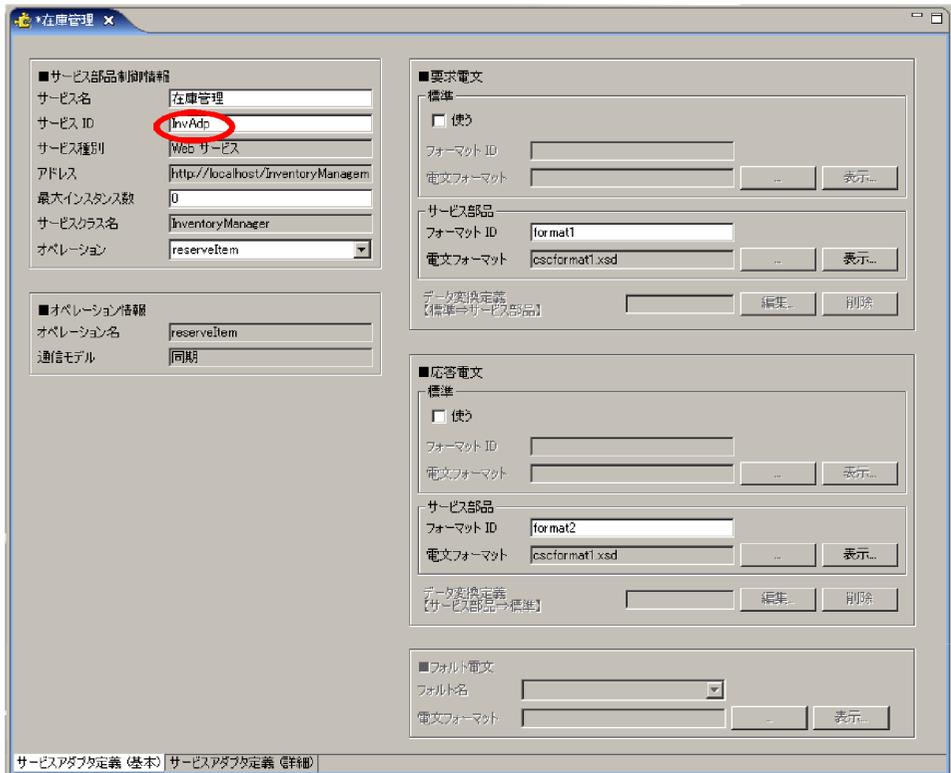
ポートを指定するダイアログが表示されます。

4. ドロップダウンリストでポート「InventoryManager」を選択して、[ 終了 ] ボタンをクリックします。



アダプタ「在庫管理サービスアダプタ」が作成され、サービスアダプタ定義画面が表示されます。

5. サービスアダプタ定義画面の基本画面で、サービス ID を「InvAdp」に変更します。



6. サービスアダプタ定義画面の詳細画面で、[ ... ] ボタンをクリックして、クライアント定義ファイル「c4webcl.properties」を指定します。

商品手配サンプルプログラムでは、次の場所にあります。

<Cosminexus インストールディレクトリ >¥CSCSTE¥Samples¥商品手配¥Service¥InventoryManagementService¥c4webcl.properties

7. [ ファイル ] - [ 保管 ] を選択します。

(b) 配送受付サービスアダプタを定義してみましょう

配送受付サービスアダプタは、配送受付サービス用 WSDL ( DeliveryService.wsdl ) を使用して定義します。配送受付サービスアダプタを定義するときに設定する値を次の表に示します。

表 C-9 配送受付サービスアダプタを定義するときに設定する値

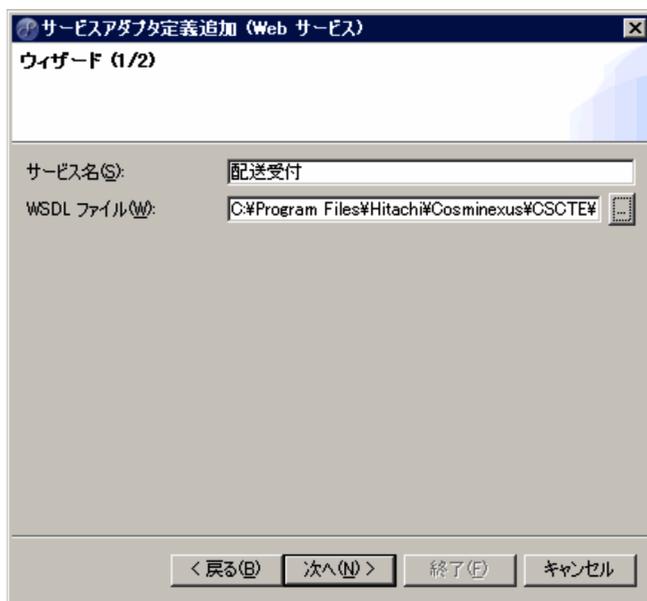
項目名	設定する値	説明
サービス部品の種別	Web サービス	呼び出すサービス部品の種別に合わせて設定したサービスアダプタの種別です。このサンプルプログラムのサービス部品は、Web サービスのためサービスアダプタの種別も Web サービスを設定します。
サービス名	配送受付	アダプタの名称を指定します。

項目名	設定する値	説明
WSDL ファイル	DeliveryService.wsdl	WSDL には、Web サービスがどのような機能を持ち、それを利用するためにはどのような要求をすればよいのか、などを記述する方法が定義されています。配送受付サービスアダプタは、配送受付サービス用 WSDL ( DeliveryService.wsdl ) を使用して作成します。
Port の指定	Delivery	配送受付サービスアダプタのポートを指定します。
サービス ID	DelAdp	配送受付サービスアダプタの ID を指定します。
クライアント定義ファイル	c4webcl.properties ( 次の内容が設定されています。 「 c4webcl.logger.log_file_prefix=DeliveryService 」 )	クライアント定義ファイルは、クライアント側の動作を制御するファイルで、「 c4webcl.properties 」というファイル名称で利用者が作成します。 このサンプルプログラムでは、トレースファイル、およびアプリケーションログのプレフィクスとして、「 DeliveryService 」を設定する、クライアント定義ファイルが用意されています。

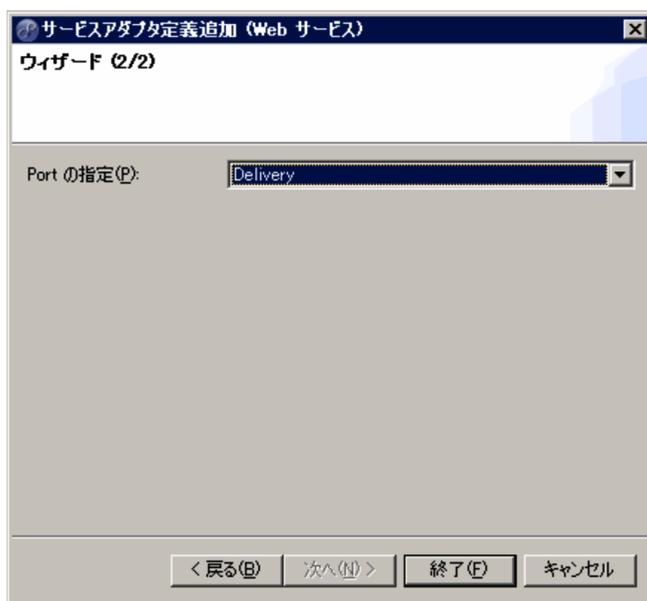
注 各ファイルの格納場所については、「付録 C.4(c) 商品手配サンプルプログラムの構成」を参照してください。

配送受付サービスアダプタの追加と定義手順を次に示します。

1. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。  
追加するサービスアダプタから利用するサービス種別を設定するダイアログが表示されます。
2. ドロップダウンリストから「Web サービス」を選択して、[ 次へ ] ボタンをクリックします。  
サービスアダプタの追加に必要な情報を入力するダイアログが表示されます。
3. サービス名に「配送受付」を、WSDL ファイルには「DeliveryService.wsdl」を入力して、[ 次へ ] ボタンをクリックします。  
ポートを指定するダイアログが表示されます。

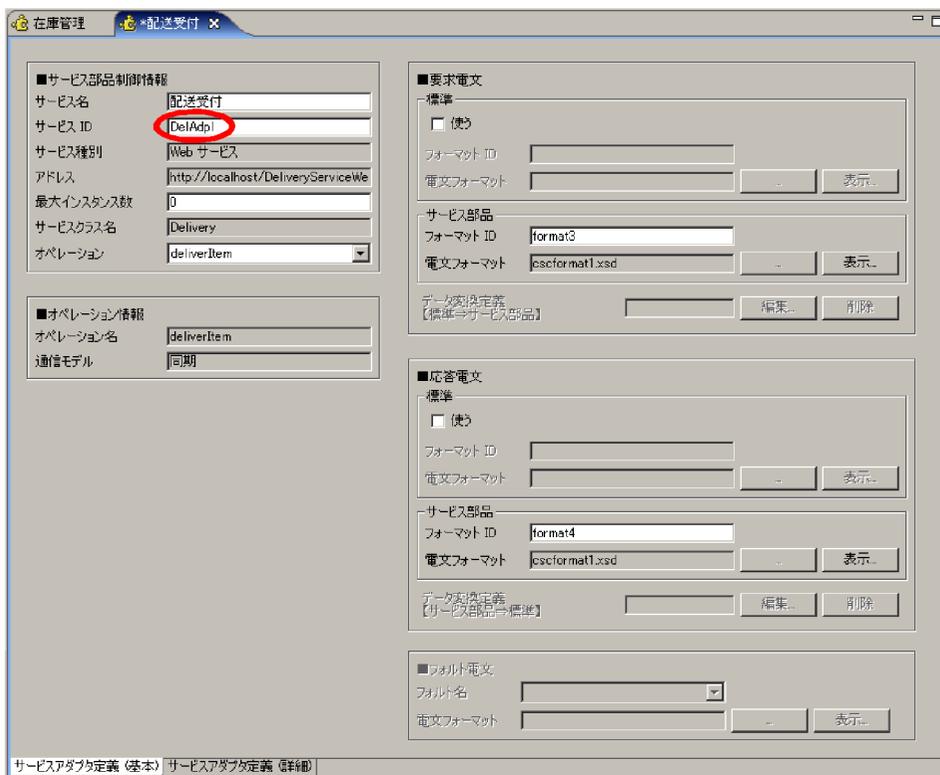


4. ドロップダウンリストでポート「Delivery」を選択して、「終了」ボタンをクリックします。



アダプタ「配送受付サービスアダプタ」が作成され、サービスアダプタ定義画面が表示されます。

5. サービスアダプタ定義画面の基本画面で、サービス ID を「DelAdp」に変更します。



6. サービスアダプタ定義画面の詳細画面で, [ ... ] ボタンをクリックして, クライアント定義ファイル「c4webcl.properties」を指定します。

商品手配サンプルプログラムでは, 次の場所にあります。

<Cosminexus インストールディレクトリ>¥CSCTE¥Samples¥商品手配  
¥Service¥DeliveryService¥c4webcl.properties

7. [ ファイル ] - [ 保管 ] を選択します。

(c) 商品手配ビジネスプロセスを定義してみましょう

商品手配サンプルプログラムの商品手配ビジネスプロセスでは, 次のようなプロセスで処理します。

1. 商品手配ビジネスプロセスは, サービスリクエストから入力された商品名と個数の情報を受け取ります。
2. 在庫管理サービスアダプタを介して在庫管理サービスを呼び出します。
3. 在庫管理サービスを呼び出した結果, 在庫がない場合は, 「\*」(在庫無し) をサービスリクエストに返します。
4. 在庫がある場合は, 配送受付サービスアダプタを介して配送受付サービスを呼び出し, 配送番号を取得します。
5. サービスリクエストに, 取得した配送番号を返します。

なお、商品手配ビジネスプロセスはユーザ定義受付を利用して呼び出すため、WSDL (ArrangementService.wsdl) を使用してユーザ定義受付を作成します。

商品手配サンプルプログラムのビジネスプロセスは、次の流れで定義します。

1. 新規ビジネスプロセスを追加します
2. ユーザ定義受付を追加します
3. 変数を設定します
4. アクティビティを配置します
5. アクティビティを定義します
6. ビジネスプロセスの定義を終了します

#### 新規ビジネスプロセスを追加します

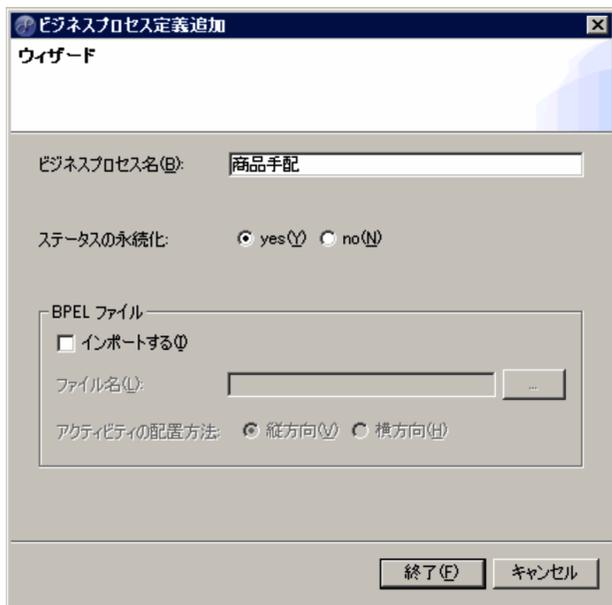
商品手配ビジネスプロセスを追加するときに設定する値を表 C-10 に示します。

表 C-10 商品手配ビジネスプロセスを追加するときに設定する値

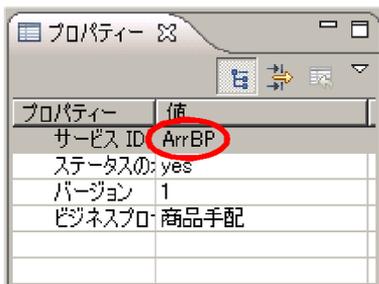
項目名	設定する値	説明
ビジネスプロセス名	商品手配	ビジネスプロセスの名称を指定します。
ステータスの永続化	yes	データベースに記録を残すかどうかを指定します。記録をとると、プロセスの進捗状況などを把握できます。このサンプルプログラムでは、データベースに記録を残すため、「yes」を選択します。
BPEL ファイルをインポートする	チェックオフ	上流工程でツールを使って作成した BPEL をインポートする場合は、チェックを入れます。インポートすると、ビジネスプロセスに必要なアクティビティが自動的に表示されます。このサンプルプログラムでは、インポートしないので、チェックを外します。
サービス ID	ArrBP	ビジネスプロセスの ID を指定します。

商品手配ビジネスプロセスの追加手順を次に示します。

1. ツリービューのサービス定義一覧で、右クリックして [ ビジネスプロセス追加 ] を選択します。  
ビジネスプロセス定義を追加するためのダイアログが表示されます。
2. [ ビジネスプロセス名 ] に「商品手配」を入力し、[ ステータスの永続化 ] で「yes」を選択します。[ BPEL ファイル ] の [ インポートする ] はチェックを外します。



3. [ 終了 ] ボタンをクリックします。  
 ビジネスプロセス「商品手配」が作成され、ビジネスプロセス定義画面が表示されます。
4. ビジネスプロセス定義画面のプロパティービューで、サービス ID を「ArrBP」に変更します。



ユーザ定義受付を追加します

商品手配サンプルプログラムでは、サービス部品に合わせた受付を使用して、サービスリクエストからの要求を受け付けます。そのため、ユーザ定義受付が必要になります。ユーザ定義受付を追加するときに設定する値を表 C-12 に示します。

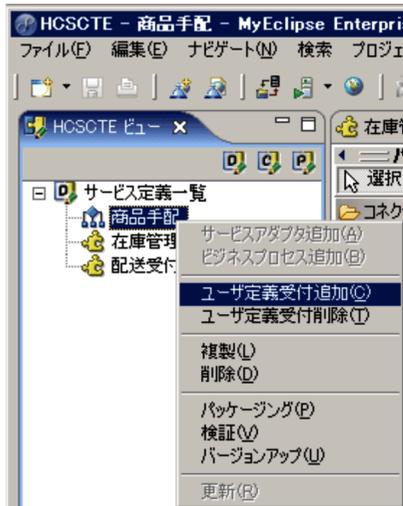
表 C-11 ユーザ定義受付を追加するときに設定する値

受付名	サービス受付	ユーザ定義受付の名称を指定します。
WSDL ファイル	ArrangementService.wsdl	使用する WSDL ファイル名を指定します。

ポート名	Arrangement	ポート名を指定します。
------	-------------	-------------

商品手配のユーザ定義受付の追加手順を次に示します。

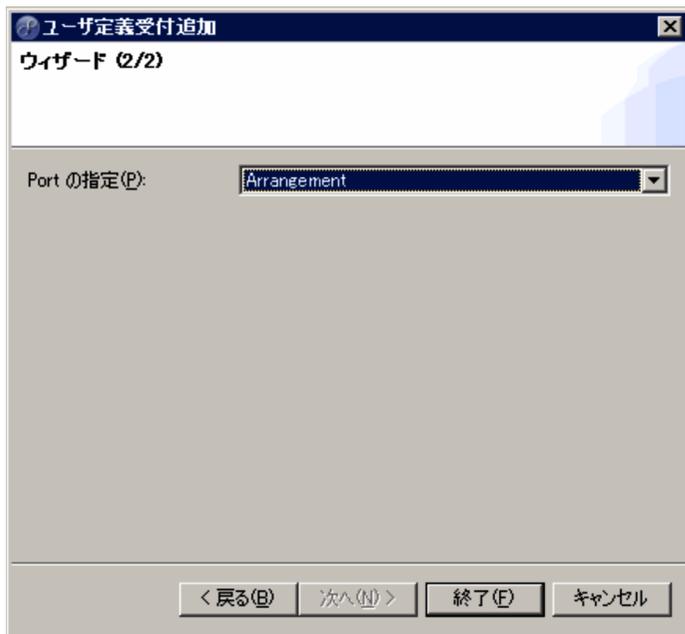
- ツリービューのサービス定義一覧からビジネスプロセス名 [ 商品手配 ] を選択し、右クリックして「ユーザ定義受付追加」を選択します。  
ユーザ定義受付を追加するためのダイアログが表示されます。



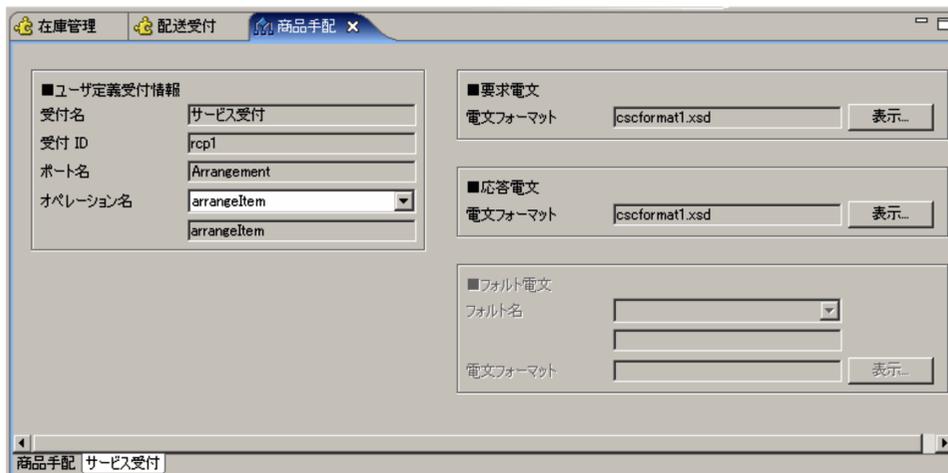
- 受付名に「サービス受付」を入力し、WSDL ファイルに「ArrangementService.wsdl」を入力します。



3. [次へ] ボタンをクリックします。
4. ドロップダウンリストでポート「Arrangement」を選択して、[終了] ボタンをクリックします。



ユーザ定義受付がビジネスプロセスに追加され、ユーザ定義受付定義画面が表示されます。



#### 変数を設定します

ビジネスプロセスでは、アクティビティを定義するときに変数を使用します。そのため、

使用する変数をあらかじめ設定してから、アクティビティを定義します。商品手配ビジネスプロセスで使用する変数を次の表に示します。

表 C-12 商品手配ビジネスプロセスで使用する変数

変数名	種別	xsd ファイル
入力データ	messageType	入力データ .xsd
出力データ	messageType	出力データ .xsd
在庫引当入力データ	messageType	在庫引当入力データ .xsd
在庫引当出力データ	messageType	在庫引当出力データ .xsd
配送手配入力データ	messageType	配送手配入力データ .xsd
配送手配出力データ	messageType	配送手配出力データ .xsd

商品手配ビジネスプロセスで使用する変数の設定手順を次に示します。

1. ビジネスプロセス定義画面のキャンバス上の [ 変数・相関セット ] アイコンをダブルクリックします。  
[ 変数・相関セット一覧 ] ダイアログが表示されます。
2. [ 変数一覧 ] を選択します。変数名に「入力データ」を入力し、種別は、ドロップダウンリストから「messageType」を選択します。
3. [ 取込 ] ボタンをクリックします。  
[ 電文フォーマットの取込 ] ダイアログが表示されます。
4. [ 受付名 ] を選択して、ドロップダウンリストから「サービス受付」を選択します。
5. [ オペレーション名 ] はドロップダウンリストから「arrangeItem」を、[ 電文種別 ] はドロップダウンリストから「要求電文」を選択します。[ 電文フォーマット ] に「入力データ」を入力します。

6. [ OK ] ボタンをクリックします。  
[ 電文フォーマットの取込 ] ダイアログが閉じます。
7. [ 変数・関連セット一覧 ] ダイアログの [ 追加 ] ボタンをクリックします。  
変数一覧に変数「入力データ」が追加されます。
8. 変数「出力データ」、「在庫引当入力データ」、「在庫引当出力データ」、「配送手配入力データ」および「配送手配出力データ」についても、手順 2. ~ 7. と同様の操作で設定します。

電文フォーマットの取込

サービス/受付

サービス名(S):

受付名(R):

取込対象

オペレーション名(O):

電文種別(T):

フォルト名(F):

電文フォーマット(M):

OK キャンセル

電文フォーマットの取込

サービス/受付

サービス名(S):

受付名(R):

取込対象

オペレーション名(O):

電文種別(T):

フォルト名(F):

電文フォーマット(M):

OK キャンセル

**電文フォーマットの取込**

サービス/受付

サービス名(S): 在庫管理

受付名(R):

取込対象

オペレーション名(O): reserveItem

電文種別(T): 応答電文

フォルト名(F):

電文フォーマット(M): 在庫引当出力データ .xsd

OK キャンセル

**電文フォーマットの取込**

サービス/受付

サービス名(S): 配送受付

受付名(R):

取込対象

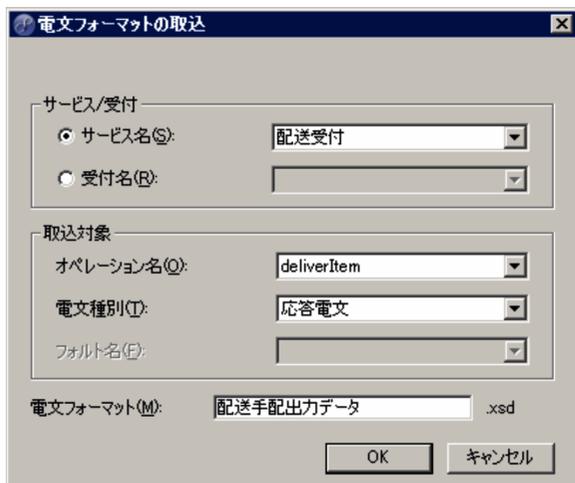
オペレーション名(O): deliverItem

電文種別(T): 要求電文

フォルト名(F):

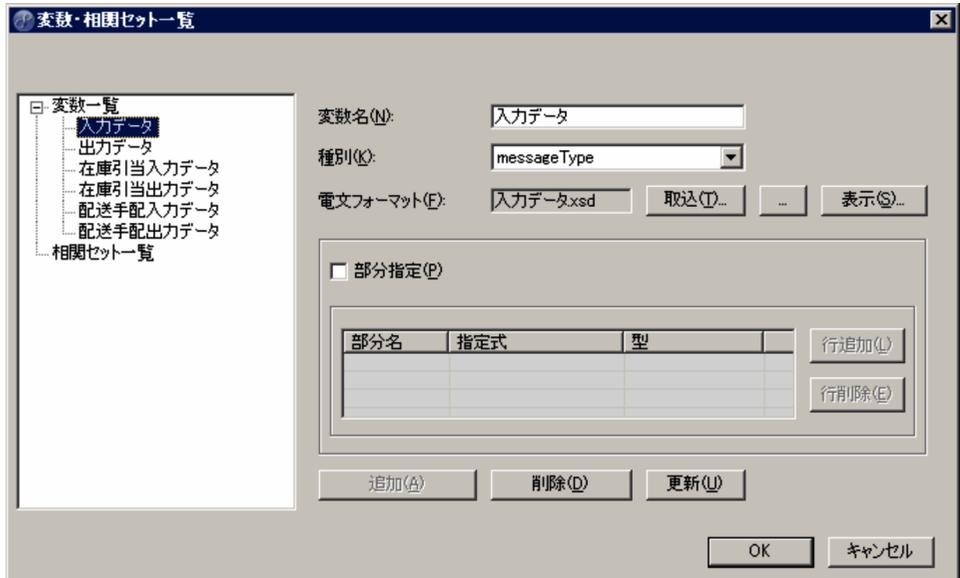
電文フォーマット(M): 配送手配入力データ .xsd

OK キャンセル



[ 変数・相関セット一覧 ] ダイアログ		[ 電文フォーマットの取込 ] ダイアログ				
変数名	種別	サービス / 受付	サービス / 受付の名称	オペレーション名	電文種別	電文フォーマット
出力データ	messageType	受付名	サービス受付	arrangeItem	応答電文	出力データ
在庫引当入力データ	messageType	サービス名	在庫管理	reserveItem	要求電文	在庫引当入力データ
在庫引当出力データ	messageType	サービス名	在庫管理	reserveItem	応答電文	在庫引当出力データ
配送手配入力データ	messageType	サービス名	配送受付	deliverItem	要求電文	配送手配入力データ
配送手配出力データ	messageType	サービス名	配送受付	deliverItem	応答電文	配送手配出力データ

9. [ 変数・相関セット一覧 ] ダイアログの [ OK ] ボタンをクリックします。  
これで変数が設定できました。



#### アクティビティを配置します

商品手配サンプルプログラムのビジネスプロセスに必要なアクティビティを次の表に示します。

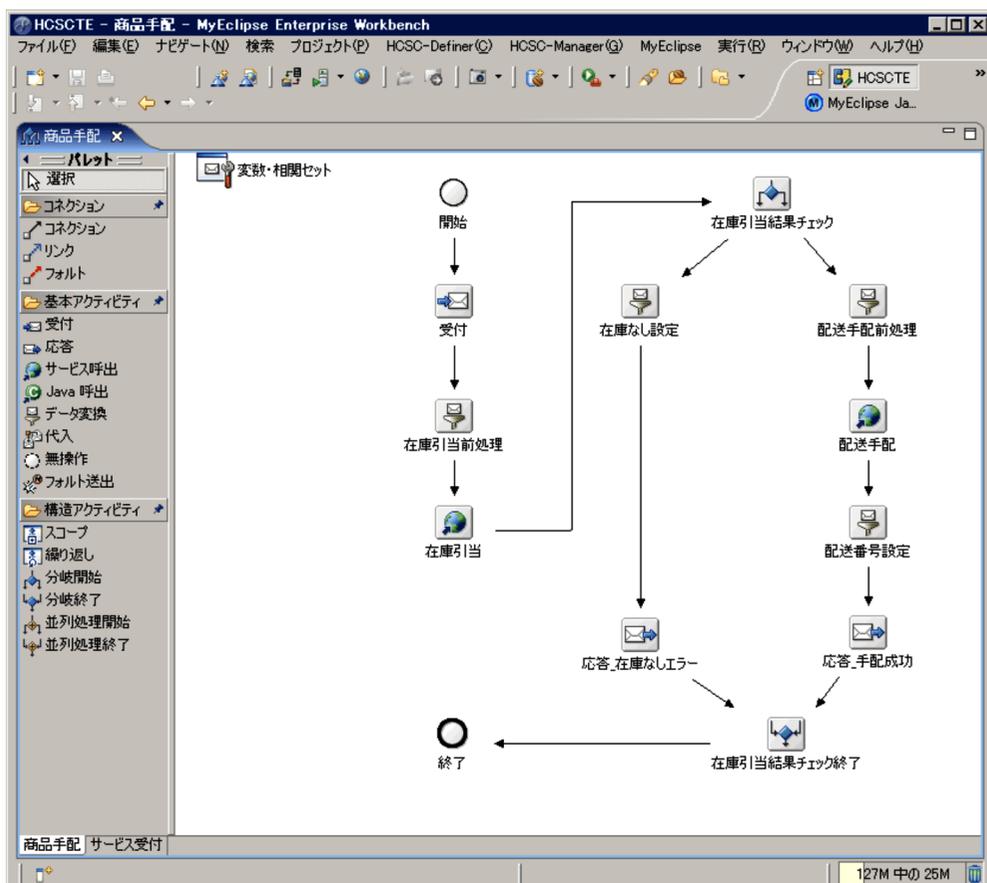
表 C-13 商品手配サンプルプログラムのビジネスプロセスに必要なアクティビティ

設定する値	説明
受付アクティビティ	サービスリクエスタからの応答を受け付けます。
データ変換アクティビティ	入力時、出力時、在庫引当時、および配送手配時のデータを編集します。
サービス呼出アクティビティ	在庫管理サービスまたは配送受付サービスを呼び出します。
分岐開始アクティビティ	条件（在庫の有無）による処理をします。
応答アクティビティ	処理結果をサービスリクエスタへ返します。
分岐終了アクティビティ	条件（在庫の有無）による処理を終了します。

アクティビティの配置手順を次に示します。

- パレットから次のアクティビティをクリックし、キャンバス上の適当な位置をクリックして配置します。
  - 受付アクティビティ
  - データ変換アクティビティ
  - サービス呼出アクティビティ
  - 分岐開始アクティビティ
  - データ変換アクティビティ

- 応答アクティビティ
  - データ変換アクティビティ
  - サービス呼出アクティビティ
  - データ変換アクティビティ
  - 応答アクティビティ
  - 分岐終了アクティビティ
2. アクティビティを連結するために、パレットの [ コネクション ] をクリックします。
  3. 連結元である開始アクティビティをクリックします。
  4. 連結先である受付アクティビティをクリックします。  
これで開始アクティビティと受付アクティビティを連結できました。
  5. 受付アクティビティから終了アクティビティまでを、手順 2. ~ 4. と同様の操作で一つずつアクティビティを連結します。



### アクティビティを定義します

キャンバスへ配置した各アクティビティの内容を定義します。

## &lt; 受付アクティビティ &gt;

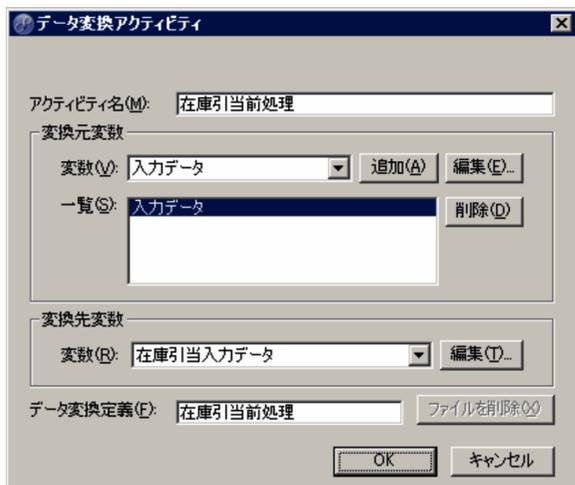
1. キャンパスの受付アクティビティをダブルクリックします。  
[ 受付アクティビティ ] ダイアログが表示されます。
2. 次の内容を入力します。

項目名	設定する値	説明
アクティビティ名	受付	アクティビティの名称を入力します。
オペレーション名	arrangeItem	サービスリクエストから在庫管理サービスを呼び出すときに利用するオペレーションの名称を入力します。
割当変数	入力データ	ビジネスプロセスの要求電文に割り当てる変数をドロップダウンリストから選択します。
割当関連セット群	設定なし	関連セットグループをアクティビティに割り当てる場合に設定します。このサンプルプログラムでは使用しないため、設定しません。
通信モデル	同期	オペレーションの通信モデルを指定します。このサンプルプログラムで使用する商品手配サービスは、Web サービスのため、「同期」を設定します。
インスタンス生成	yes	要求電文を受け付けたときに、プロセスを初期化するかどうかを選択します。このサンプルプログラムでは、初期化するため「yes」を設定します。

3. [ OK ] ボタンをクリックします。

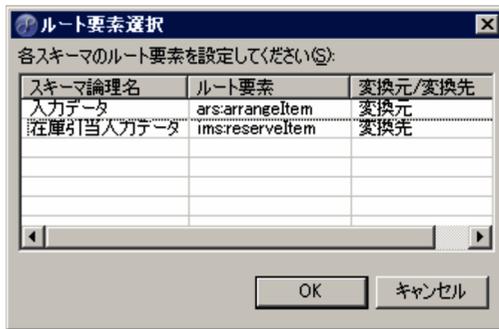
## &lt; データ変換アクティビティ &gt; (在庫引当の前処理の場合)

1. キャンパスのデータ変換アクティビティをダブルクリックします。  
[ データ変換アクティビティ ] ダイアログが表示されます。
2. 次の内容を入力します。

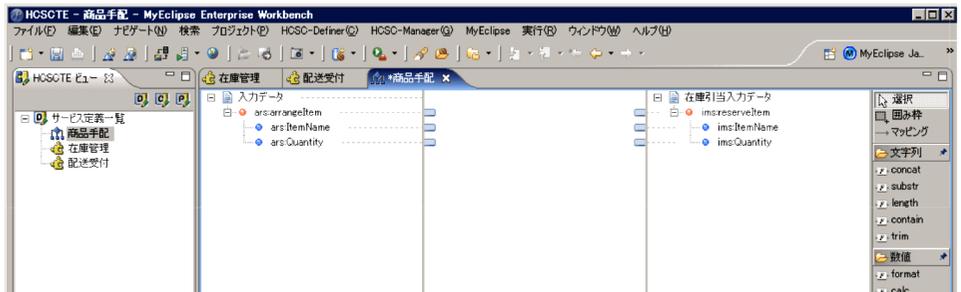


項目名	設定する値	説明
アクティビティ名	在庫引当処理	アクティビティの名称を入力します。
変数 (変換元変数)	入力データ	データの変換元になる変数をドロップダウンリストから選択し,[追加]ボタンをクリックします。
変数 (変換先変数)	在庫引当入力データ	データの変換先になる変数をドロップダウンリストから選択します。
データ変換定義	在庫引当処理	変数の変換に使うデータ変換定義ファイルの名称を入力します。

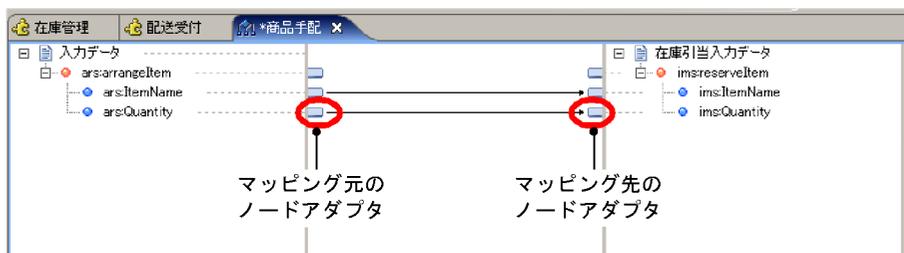
3. [ OK ] ボタンをクリックします。
4. キャンパスのデータ変換アクティビティを右クリックし,[ マッピング定義起動 ] を選択します。  
[ ルート要素選択 ] ダイアログが表示されます。
5. 変換元のスキーマ論理名「入力データ」のルート要素をクリックして,ドロップダウンリストから「ars:arrangeltem」を選択します。
6. 変換先のスキーマ論理名「在庫引当入力データ」のルート要素をクリックして,ドロップダウンリストから「ims:reserveltem」を選択します。



7. [ OK ] ボタンをクリックします。  
データ変換定義画面が表示されます。

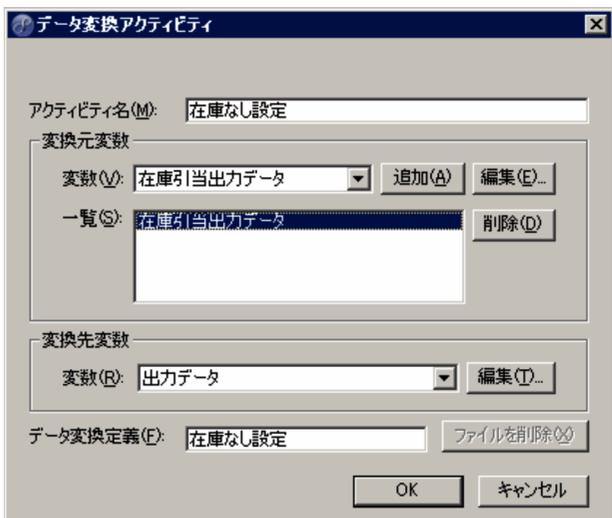


8. データ変換定義画面のパレットから「マッピング」を選択します。
9. マッピング元となる変換元ノードのノードアダプタをクリックします。
10. マッピング先となる変換先ノードのノードアダプタをクリックします。  
マッピング線が設定されます。



< データ変換アクティビティ > (在庫なしの場合)

1. キャンパスのデータ変換アクティビティをダブルクリックします。  
[ データ変換アクティビティ ] ダイアログが表示されます。
2. 次の内容を入力します。

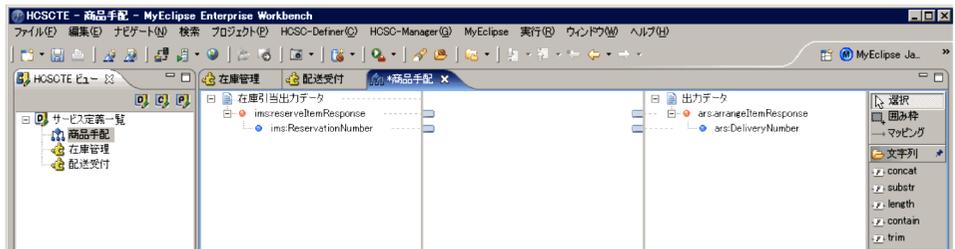


項目名	設定する値	説明
アクティビティ名	在庫なし設定	アクティビティの名称を入力します。
変数 (変換元変数)	在庫引当出力データ	データの変換元になる変数をドロップダウンリストから選択し,[追加]ボタンをクリックします。
変数 (変換先変数)	出力データ	データの変換先になる変数をドロップダウンリストから選択します。
データ変換定義	在庫なし設定	変数の変換に使うデータ変換定義ファイルの名称を入力します。

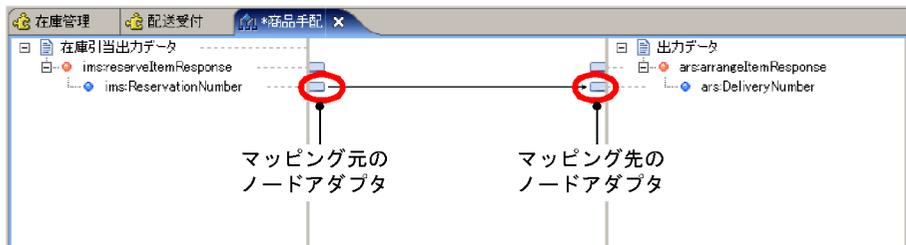
3. [ OK ] ボタンをクリックします。
4. キャンパスのデータ変換アクティビティを右クリックし,[ マッピング定義起動 ] を選択します。  
[ ルート要素選択 ] ダイアログが表示されます。
5. 変換元のスキーマ論理名「在庫引当出力データ」のルート要素をクリックして、ドロップダウンリストから「ims:reserveItemResponse」を選択します。
6. 変換先のスキーマ論理名「出力データ」のルート要素をクリックして、ドロップダウンリストから「ars:arrangeItemResponse」を選択します。



7. [ OK ] ボタンをクリックします。  
データ変換定義画面が表示されます。

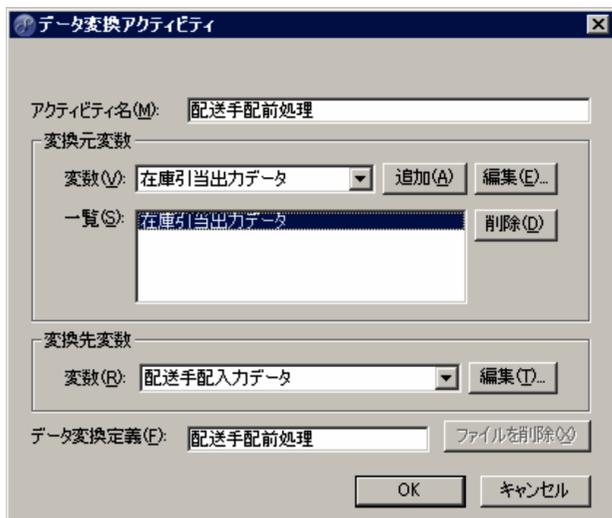


8. データ変換定義画面のパレットから [ マッピング ] を選択します。
9. マッピング元となる変換元ノードのノードアダプタをクリックします。
10. マッピング先となる変換先ノードのノードアダプタをクリックします。  
マッピング線が設定されます。



<データ変換アクティビティ> (配送手配の前処理の場合)

1. キャンパスのデータ変換アクティビティをダブルクリックします。  
[ データ変換アクティビティ ] ダイアログが表示されます。
2. 次の内容を入力します。

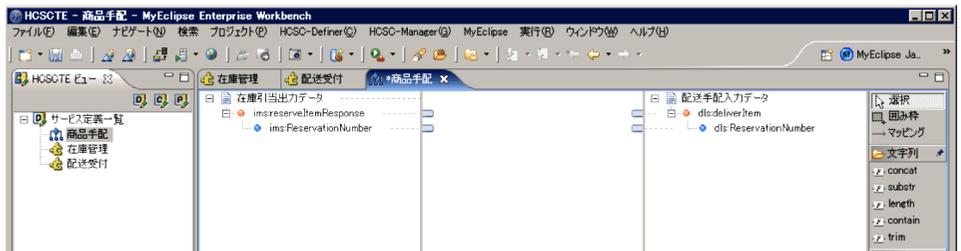


項目名	設定する値	説明
アクティビティ名	配送手配前処理	アクティビティの名称を入力します。
変数 (変換元変数)	在庫引当出力データ	データの変換元になる変数をドロップダウンリストから選択し、[追加] ボタンをクリックします。
変数 (変換先変数)	配送手配入力データ	データの変換先になる変数をドロップダウンリストから選択します。
データ変換定義	配送手配前処理	変数の変換に使うデータ変換定義ファイルの名称を入力します。

3. [ OK ] ボタンをクリックします。
4. キャンパスのデータ変換アクティビティを右クリックし、[ マッピング定義起動 ] を選択します。  
[ ルート要素選択 ] ダイアログが表示されます。
5. 変換元のスキーマ論理名「在庫引当出力データ」のルート要素をクリックして、ドロップダウンリストから「ims:reserveItemResponse」を選択します。
6. 変換先のスキーマ論理名「配送手配入力データ」のルート要素をクリックして、ドロップダウンリストから「dls:deliverItem」を選択します。



7. [ OK ] ボタンをクリックします。  
データ変換定義画面が表示されます。

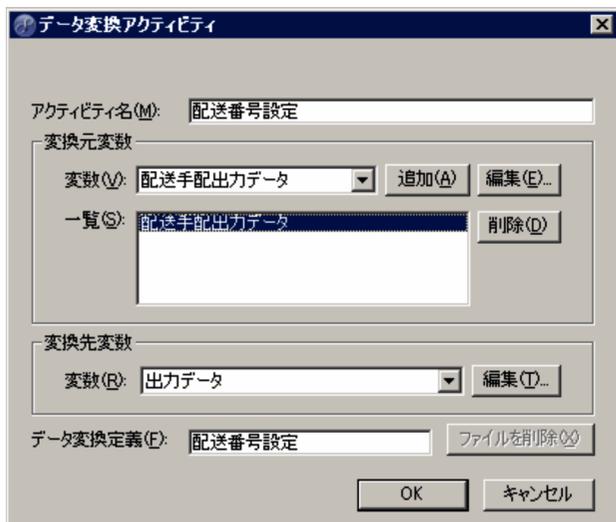


8. データ変換定義画面のパレットから「マッピング」を選択します。
9. マッピング元となる変換元ノードのノードアダプタをクリックします。
10. マッピング先となる変換先ノードのノードアダプタをクリックします。  
マッピング線が設定されます。



<データ変換アクティビティ> ( 配送番号の設定の場合 )

1. キャンパスのデータ変換アクティビティをダブルクリックします。  
[ データ変換アクティビティ ] ダイアログが表示されます。
2. 次の内容を入力します。

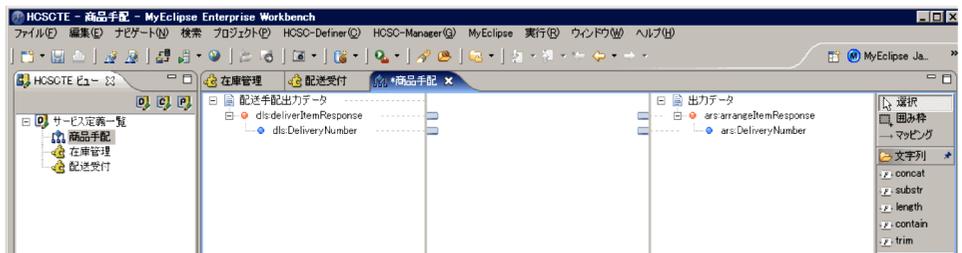


項目名	設定する値	説明
アクティビティ名	配送番号設定	アクティビティの名称を入力します。
変数 (変換元変数)	配送手配出力データ	データの変換元になる変数をドロップダウンリストから選択し,[追加]ボタンをクリックします。
変数 (変換先変数)	出力データ	データの変換先になる変数をドロップダウンリストから選択します。
データ変換定義	配送番号設定	変数の変換に使うデータ変換定義ファイルの名称を入力します。

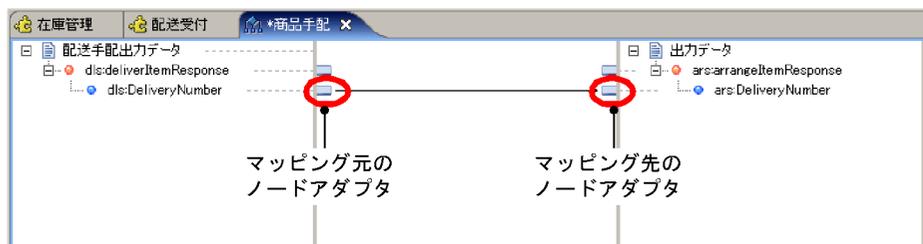
3. [OK] ボタンをクリックします。
4. キャンパスのデータ変換アクティビティを右クリックし,[ マッピング定義起動 ] を選択します。  
[ルート要素選択] ダイアログが表示されます。
5. 変換元のスキーマ論理名「配送手配出力データ」のルート要素をクリックして、ドロップダウンリストから「dls:deliverItemResponse」を選択します。
6. 変換先のスキーマ論理名「出力データ」のルート要素をクリックして、ドロップダウンリストから「ars:arrangeItemResponse」を選択します。



7. [ OK ] ボタンをクリックします。  
データ変換定義画面が表示されます。

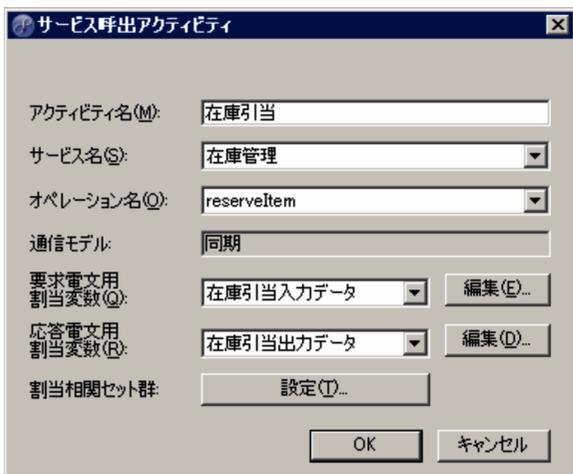


8. データ変換定義画面のパレットから [ マッピング ] を選択します。
9. マッピング元となる変換元ノードのノードアダプタをクリックします。
10. マッピング先となる変換先ノードのノードアダプタをクリックします。  
マッピング線が設定されます。



< サービス呼出アクティビティ > (在庫を引き当てる場合)

1. キャンパスのサービス呼出アクティビティをダブルクリックします。  
[ サービス呼出アクティビティ ] ダイアログが表示されます。
2. 次の内容を入力します。



項目名	設定する値	説明
アクティビティ名	在庫引当	アクティビティの名称を入力します。
サービス名	在庫管理	要求電文を送信して呼び出すサービス部品の名称をドロップダウンリストから選択します。
オペレーション名	reserveItem	[ サービス名 ] で指定したサービス部品 ( 在庫管理 ) のオペレーションのうち、実際に呼び出すオペレーションの名称を指定します。
通信モデル	同期	[ オペレーション名 ] で指定したオペレーションに設定されている通信モデルが表示されます。
要求電文用割当変数	在庫引当入力データ	在庫管理サービスを呼び出す要求電文に割り当てる変数をドロップダウンリストから選択します。
応答電文用割当変数	在庫引当出力データ	同期オペレーションから受け取る応答電文に割り当てる変数をドロップダウンリストから選択します。
割当関連セット群	なし	関連セットグループをアクティビティに割り当てる場合に設定します。このサンプルプログラムでは使用しないため、設定しません。

< サービス呼出アクティビティ > ( 配送を手配する場合 )

1. キャンパスのサービス呼出アクティビティをダブルクリックします。  
[ サービス呼出アクティビティ ] ダイアログが表示されます。
2. 次の内容を入力します。

項目名	設定する値	説明
アクティビティ名	配送手配	アクティビティの名称を入力します。
サービス名	配送受付	要求電文を送信して呼び出すサービス部品の名称をドロップダウンリストから選択します。
オペレーション名	deliverItem	[ サービス名 ] で指定したサービス部品（配送手配）のオペレーションのうち、実際に呼び出すオペレーションの名称を指定します。
通信モデル	同期	[ オペレーション名 ] で指定したオペレーションに設定されている通信モデルが表示されます。
要求電文用割当変数	配送手配入力データ	配送手配サービスを呼び出す要求電文に割り当てる変数をドロップダウンリストから選択します。
応答電文用割当変数	配送手配出力データ	同期オペレーションから受け取る応答電文に割り当てる変数をドロップダウンリストから選択します。
割当関連セット群	なし	関連セットグループをアクティビティに割り当てる場合に設定します。このサンプルプログラムでは使用しないため、設定しません。

< 応答アクティビティ >（在庫がない場合）

1. キャンパスの応答アクティビティをダブルクリックします。  
[ 応答アクティビティ ] ダイアログが表示されます。
2. 次の内容を入力します。



項目名	設定する値	説明
アクティビティ名	応答_在庫なしエラー	アクティビティの名称を入力します。
オペレーション名	arrangeItem	対応する受付アクティビティに指定したオペレーション名を入力します。
割当変数	出力データ	ビジネスプロセスの応答電文に割り当てる変数をドロップダウンリストから選択します。
割当関連セット群 :	設定なし	関連セットグループをアクティビティに割り当てる場合に入力します。このサンプルプログラムでは使用しないため、設定しません。
フォルト名	設定なし	フォルト処理として応答アクティビティを定義して、サービスリクエストにフォルトが発生したことを示す応答電文を送信する場合のフォルトの名称を入力します。このサンプルプログラムではフォルト処理を使用しないため、設定しません。

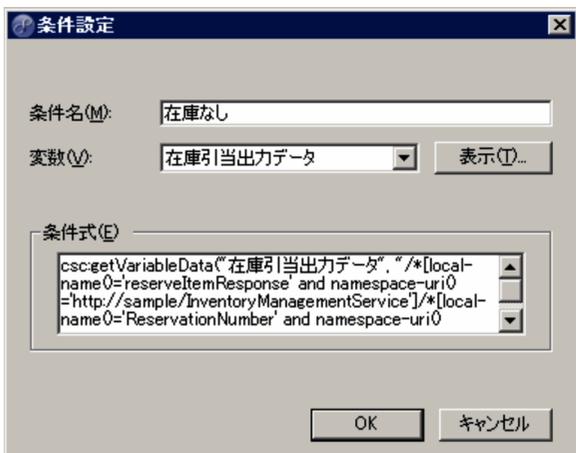
< 応答アクティビティ > ( 配送手配が成功した場合 )

1. キャンパスの応答アクティビティをダブルクリックします。  
[ 応答アクティビティ ] ダイアログが表示されます。
2. 次の内容を入力します。

項目名	設定する値	説明
アクティビティ名	応答_手配成功	アクティビティの名称を入力します。
オペレーション名	arrangeItem	対応する受付アクティビティに指定したオペレーション名を入力します。
割当変数	出力データ	ビジネスプロセスの応答電文に割り当てる変数をドロップダウンリストから選択します。
割当関連セット群 :	設定なし	関連セットグループをアクティビティに割り当てる場合に入力します。このサンプルプログラムでは使用しないため、設定しません。
フォルト名	設定なし	フォルト処理として応答アクティビティを定義して、サービスリクエストにフォルトが発生したことを示す応答電文を送信する場合のフォルトの名称を入力します。このサンプルプログラムではフォルト処理を使用しないため、設定しません。

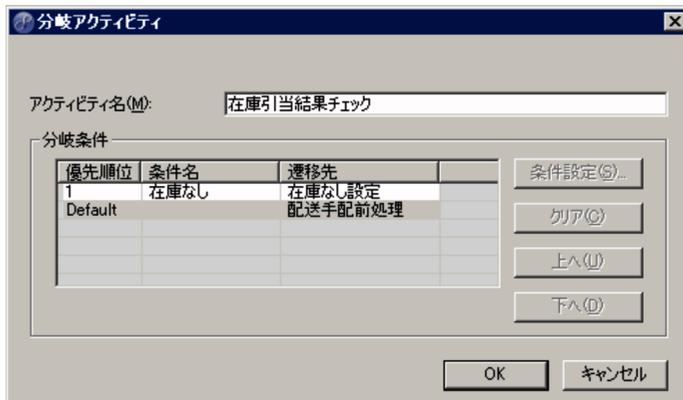
#### <分岐開始クティビティ>

1. キャンパスの分岐開始アクティビティをダブルクリックします。  
[分岐開始アクティビティ] ダイアログが表示されます。
2. アクティビティ名に「在庫引当結果チェック」を入力します。
3. [遷移先] が [在庫なし設定] の行をクリックし、[上へ] ボタンをクリックして、一番上の行に移動します。
4. 一番上の行を選択し、[条件設定] ボタンをクリックします。  
[条件設定] ダイアログが表示されます。
5. 次の内容を入力します。



項目名	設定する値	説明
条件名	在庫なし	在庫引当チェックの結果、在庫がない場合の条件を指定します。
変数	在庫引当出力データ	条件式で使用する変数をドロップダウンリストから選択します。
条件式	csc:getVariableData("在庫引当出力データ", "/*[local-name()='reserveItemResponse' and namespace-uri()='http://sample/InventoryManagementService']/*[local-name()='ReservationNumber' and namespace-uri()='http://sample/InventoryManagementService']")=*"."	在庫がない場合の条件式をXPath式で指定します。

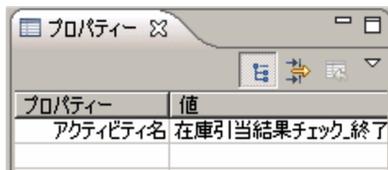
6. [ OK ] ボタンをクリックします。  
分岐アクティビティ画面に戻ります。
7. [ 遷移先 ] が [ 配送手配前処理 ] の [ 優先順位 ] をクリックし、ドロップダウンリストから「Default」を選択します。  
「Default」を選択すると、条件の設定は不要になります。



8. [ OK ] ボタンをクリックします。

#### < 分岐終了アクティビティ >

1. キャンパスの分岐終了アクティビティをクリックします。  
プロパティービューに、分岐終了アクティビティの内容が表示されます。
2. アクティビティ名に、「在庫引当結果チェック\_終了」を入力します。



#### ビジネスプロセスの定義を終了します

すべてのアクティビティを定義したら、[ ファイル ] - [ 保管 ] を選択して、ビジネスプロセスの定義を終了します。

#### (d) 検証とパッケージングをしてみましょう

次に示すものは、定義した内容が正しいかどうか検証してから、パッケージングします。

- 在庫管理サービスアダプタ
- 配送受付サービスアダプタ
- 商品手配ビジネスプロセスおよびユーザ定義受付

検証とパッケージングは、Hello サービスアダプタの検証とパッケージングの手順と同様です。手順の詳細については、「付録 C.3(1)(b) Hello サービスアダプタの検証とパッケージングをしてみましょう」を参照してください。

## 付録 C.4 サンプルプログラムのファイルの構成

各サンプルプログラムは、次の表に示す場所にあります。

表 C-14 サンプルプログラムの格納場所

サンプルプログラム名	サンプルプログラムの格納場所
HelloServiceAdapter	<Cosminexus インストールディレクトリ >¥CSCTE¥Samples¥HelloServiceAdapter
HelloBusinessProcess	<Cosminexus インストールディレクトリ >¥CSCTE¥Samples¥HelloBusinessProcess
商品手配	<Cosminexus インストールディレクトリ >¥CSCTE¥Samples¥ 商品手配

各サンプルプログラムのファイルの構成を次に示します。

なお、太字はサンプルプログラムで実装するファイルです。太字以外のファイルは自動生成で作成したファイルです。

### (a) HelloServiceAdapter サンプルプログラムの構成

```

HelloServiceAdapter
Repository ..... リポジトリのディレクトリ
HelloServiceAdapter.zip ..... リポジトリ
Requester ..... サービスリクエストのディレクトリ
HelloClient ..... MyEclipse Java エンタープライズ・アプリケーション・プロジェクト
    .mymetadata
    .project
    .settings
    org.eclipse.core.runtime.prefs
    META-INF
    application.xml
    MANIFEST.MF

HelloClientWeb ..... MyEclipse Java Web プロジェクト
    .classpath
    .mymetadata
    .project
    .myeclipse
    .settings
    org.eclipse.core.resources.prefs
    org.eclipse.core.runtime.prefs
    org.eclipse.jdt.core.prefs

src
  jp
    co
      Hitachi
        soft
          csc
            msg
    
```

```

message
  reception
    ejb ..... WSDL を基に生成されるクライアントスタブを格納
      CSCMsgServerFaultException.java
      CSCMsgSyncServiceDeliveryWSImpl.java
      CSCMsgSyncServiceDeliveryWSImplService.java
      CSCMsgSyncServiceDeliveryWSImplServiceLocator.java
      CSCMsgSyncServiceDeliveryWSImplSoapBindingStub.java

sample
servlet
  HelloServlet.java ..... サービス部品呼び出しサーブレット
xml
  DomUtil.java
  XmlErrorHandler.java

WebRoot
  index.html ..... 入力画面
  META-INF
  MANIFEST.MF
  WEB-INF
    web.xml ..... DD
  classes
  lib

Service ..... サービス部品のディレクトリ
  HelloService ..... MyEclipse Java エンタープライズ・アプリケーション・プロジェクト
    .metadata
    .project
    c4webcl.properties ..... Hello サービスアダプタで使用するクライアント定義ファイル
    .settings
    org.eclipse.core.runtime.prefs
  META-INF
  application.xml
  MANIFEST.MF

  HelloServiceWeb ..... MyEclipse Java Web プロジェクト
    .classpath
    .mymetadata
    .project
    .myeclipse

    .settings
    org.eclipse.core.runtime.prefs
  src
  sample
  HelloService
  Hello.java ..... サーバスケルトン
  HelloSoapBindingImpl.java ..... Hello サービス (サーバスケルトン)

```

```

WebRoot
  META-INF
    MANIFEST.MF
  WEB-INF
    server-config.xml ..... サービスデプロイ定義ファイル (サーバスケルトン)
    web.xml ..... DD
  classes
  lib

WSDL
cscmsg_ws.wsdl
HelloService.wsdl ..... Hello サービス用 WSDL
    
```

(b) HelloBusinessProcess サンプルプログラムの構成

```

HelloBusinessProcess
  Repository ..... リポジトリのディレクトリ
    HelloBusinessProcess.zip ..... リポジトリ
  Requester ..... サービスリクエストのディレクトリ
  HelloClient ..... MyEclipse Java エンタープライズ・アプリケーション・プロジェクト
    .mymetadata
    .project
    .settings
    org.eclipse.core.runtime.prefs

  META-INF
    application.xml
    MANIFEST.MF

  HelloClientWeb ..... MyEclipse Java Web プロジェクト
    .classpath
    .mymetadata
    .project
    .myeclipse
    .settings
    org.eclipse.core.resources.prefs
    org.eclipse.core.runtime.prefs
    org.eclipse.jdt.core.prefs

  src
    jp
      co
        Hitachi
          soft
            csc
              msg
    
```

```

message
  reception
    ejb ..... WSDL を基に生成されるクライアントスタブを格納
    CSCMsgServerFaultException.java
    CSCMsgSyncServiceDeliveryWSImpl.java
    CSCMsgSyncServiceDeliveryWSImplService.java
    CSCMsgSyncServiceDeliveryWSImplServiceLocator.java
    CSCMsgSyncServiceDeliveryWSImplSoapBindingStub.java

sample
  servlet
    HelloServlet.java ..... サービス部品呼び出しサーブレット
  xml
    DomUtil.java
    XmlErrorHandler.java

WebRoot
  index.html ..... 入力画面
  META-INF
  MANIFEST.MF
  WEB-INF
    web.xml ..... DD
  classes
  lib

Service ..... サービス部品のディレクトリ
  HelloService ..... MyEclipse Java エンタープライズ・アプリケーション・プロジェクト
    .metadata
    .project
    c4webcl.properties ..... Hello サービスアダプタで使用するクライアント定義ファイル
    .settings
    org.eclipse.core.runtime.prefs
  META-INF
  application.xml
  MANIFEST.MF

  HelloServiceWeb ..... MyEclipse Java Web プロジェクト
    .classpath
    .mymetadata
    .project
    .myeclipse
    .settings
    org.eclipse.core.runtime.prefs

src
  sample
    HelloService
    Hello.java ..... サーバスケルトン
    HelloSoapBindingImpl.java ..... Hello サービス (サーバスケルトン)

```

```

WebRoot
  META-INF
    MANIFEST.MF
  WEB-INF
    server-config.xml ..... サービスデプロイ定義ファイル(サーバスケルトン)
    web.xml ..... DD
  classes
  lib

WSDL
  cscmsg_ws.wsdl
  HelloService.wsdl ..... Hello サービス用 WSDL
    
```

(c) 商品手配サンプルプログラムの構成

```

商品手配
  Repository ..... リポジトリのディレクトリ
  商品手配.zip ..... リポジトリ
  Requester ..... サービスリクエストのディレクトリ
  ArrangementClient ..... MyEclipse Java エンタープライズ・アプリケーション・プロジェクト
ト
  .mymetadata
  .project
  .settings
  org.eclipse.core.runtime.prefs
  META-INF
  application.xml
  MANIFEST.MF

ArrangementClientWeb ..... MyEclipse Java Web プロジェクト
  .classpath
  .mymetadata
  .project
  .myeclipse
  .settings
  org.eclipse.core.resources.prefs
  org.eclipse.core.runtime.prefs
  org.eclipse.jdt.core.prefs
  org.eclipse.jdt.ui.prefs

src
  sample
  ArrangementService
  Arrangement.java ..... クライアントスタブ
  ArrangementService.java ..... クライアントスタブ
  ArrangementServiceLocator.java ..... クライアントスタブ
  ArrangementSoapBindingStub.java ..... クライアントスタブ
  servlet
  ArrangementServlet.java ..... サービス部品呼び出しサーブレット
    
```

```

WebRoot
  error.jsp ..... エラー応答画面
  index.html ..... 入力画面
  result.jsp ..... 正常応答画面
META-INF
MANIFEST.MF
WEB-INF
  web.xml ..... DD
  classes
  lib

Service
DeliveryService ..... MyEclipse Java エンタープライズ・アプリケーション・プロジェクト
  .mymetadata
  .project
  c4webcl.properties ..... 配送受付サービスアダプタで使用するクライアント定義ファイル
  .settings
  org.eclipse.core.runtime.prefs
META-INF
  application.xml
  MANIFEST.MF

DeliveryServiceWeb ..... MyEclipse Java Web プロジェクト
  .classpath
  .mymetadata
  .project
  .myeclipse
  .settings
  org.eclipse.core.runtime.prefs

src
  sample
    DeliveryService
      Delivery.java ..... サーバスケルトン
      DeliverySoapBindingImpl.java ..... 配送受付サービス (サーバスケルトン)

WebRoot
META-INF
  MANIFEST.MF
WEB-INF
  server-config.xml ..... サービスデプロイ定義ファイル (サーバスケルトン)
  web.xml ..... DD
  classes
  lib

```

InventoryManagementService	.....	MyEclipse Java エンタープライズ・アプリケーション・プロジェクト
.mymetadata		
.project		
<b>c4webl.properties</b>	.....	在庫管理サービスアダプタで使用するクライアント定義ファイル
.settings		
org.eclipse.core.runtime.prefs		
META-INF		
application.xml		
MANIFEST.MF		
InventoryManagementServiceWeb	.....	MyEclipse Java Web プロジェクト
.classpath		
.mymetadata		
.project		
.myeclipse		
.settings		
org.eclipse.core.runtime.prefs		
src		
sample		
InventoryManagementService		
InventoryManager.java	.....	サーバスケルトン
<b>InventoryManagerSoapBindingImpl.java</b>	...	在庫管理サービス (サーバスケルトン)
WebRoot		
META-INF		
MANIFEST.MF		
WEB-INF		
server-config.xml	.....	サービスデプロイ定義ファイル (サーバスケルトン)
<b>web.xml</b>	.....	DD
classes		
lib		
WSDL		
<b>ArrangementService.wsdl</b>	.....	商品手配用 WSDL
<b>DeliveryService.wsdl</b>	.....	配送受付サービス用 WSDL
<b>InventoryManagementService.wsdl</b>	.....	在庫管理サービス用 WSDL

---

## 付録 D 用語解説

---

### (英字)

---

#### BPEL ( Business Process Execution Language )

XML をベースにしたワークフロー記述言語です。Cosminexus サービスプラットフォームでは、アクティビティとコネクションで定義したビジネスプロセスを BPEL 形式で保存できます。

#### BPMN ( Business Process Modeling Notation )

業務手順をわかりやすく図示して可視化するための表記ルールを定めたものです。ビジネスプロセス・モデリング表記法ともいいます。

#### DB Connector

データベースに接続するためのリソースアダプタです。

#### DB アダプタ

データベースの操作をサービス部品として利用する場合に、操作するデータベースや実行する SQL を定義した HCSC コンポーネントです。

#### DB キュー

データベースを介してシステム間で連携する、TP1/EE ( TP1/Server Base Enterprise Option ) の機能です。DB キュー機能を使用すると、データベースをキューとして、TP1/EE と J2EE との間で、非同期的なメッセージ送受信ができます。

#### EAR ( Enterprise ARchive ) ファイル

J2EE アプリケーションを構成する複数のファイルを EAR ファイル形式でパッケージ化したものです。

Cosminexus サービスプラットフォームでは、HCSC コンポーネントをパッケージングして、EAR ファイルを作成します。

#### Eclipse

Eclipse プロジェクト ( eclipse.org ) が提供するオープンソースの統合開発環境です。ソースコードの編集支援機能やデバッグ機能など、アプリケーションの開発効率を向上させる各種機能を備えています。

Cosminexus サービスプラットフォームでは、Eclipse に必要な機能をプラグインして利用します。

#### EJB ( Enterprise Java Beans )

業務ロジックをプログラムとして記述したビジネスロジック、および永続的データを格納するオブジェクトであるエンティティを Java コンポーネント化したものです。Sun Microsystems, Inc. から EJB 仕様が公開されています。

#### HCSC ( Hitachi Cosminexus Service Coordinator )

uCosminexus Service Platform を構成する構成ソフトウェアの一つです。Cosminexus サービスプラットフォームで SOA を適用したシステムを構築、運用するための機能として、ビジネスプロセス

実行機能，データ変換機能，DB アダプタ機能，実行環境の運用管理機能，およびメッセージング制御機能を提供しています。

### HCSC-Business Process ( ビジネスプロセス基盤 )

ビジネスプロセスの定義に従って，適切なサービス部品を呼び出すための機能です。

### HCSC-Data Transform ( データ変換基盤 )

データ変換定義に従って，要求電文・応答電文の電文フォーマットを変換する機能です。

### HCSC-DB Adapter ( DB アダプタ連携基盤 )

DB アダプタの定義に従って適切なサービス部品 ( データベースの操作 ) を呼び出す機能です。

### HCSCTE

Cosminexus サービスプラットフォームの開発環境で利用する開発支援機能です。HCSCTE で提供される機能は，Eclipse にプラグインして利用します。

アダプタやビジネスプロセスの作成・定義機能などが含まれます。

### HCSC コンポーネント

開発環境で作成するサービスアダプタ，DB アダプタ，ビジネスプロセス，およびユーザ定義受付の総称です。

### HCSC コンポーネントの検証

定義したアダプタ，ビジネスプロセス，データ変換定義の内容が妥当かどうか検証する機能です。必要な定義があるか，また定義の関係が正しいかを，必要に応じて任意のタイミングで検証できます。

### HCSC サーバ

実行環境でサービス部品の実行を管理するサーバ機能です。メッセージング基盤，DB アダプタ連携基盤，ビジネスプロセス基盤，データ変換基盤など，実行環境に必要な機能が含まれます。また，標準受付や連携受付などの機能も含まれます。

### HCSC サーバランタイム定義ファイル ( cscsvconfig.properties )

HCSC サーバの起動時に必要なランタイム情報を定義するファイルです。J2SE のプロパティ形式のファイルとして作成します。

実行履歴を採取するかどうかや，トレースファイルのレベル・出力先などを定義できます。

### HTTP

インターネットで，WWW サーバと WWW クライアントの間で HTML 文書を送受信するための通信プロトコルです。

### IDE ( Integrated Development Environment )

Java の統合開発環境です。Cosminexus サービスプラットフォームでは，Eclipse を利用します。

### J2EE ( Java™ 2 Platform, Enterprise Edition )

Sun Microsystems, Inc. から J2EE 仕様が公開されています。

## J2EE サーバ

アプリケーションを実行するためのサーバ基盤である J2EE コンテナを生成、実行する環境です。Cosminexus サービスプラットフォームは、J2EE サーバとして Cosminexus Component Container を使用します。

## Java

1991 年に Sun Microsystems 社内の研究グループが開発したインタプリタ言語を基に、C++ をより移植性と信頼性が高くなるように改良した言語です。Web の普及によってブラウザに应用されています。

## MDB ( Message-Driven Bean )

JMS と連携するメッセージ駆動タイプの Bean です。EJB コンテナは JMS の Destination からの JMS メッセージ受信を契機に Bean を起動します。

## MyEclipse

Java アプリケーションの開発環境です。MyEclipse には、Eclipse でアプリケーションを開発するときに使用するプラグインがまとめられています。MyEclipse を使用すると、アプリケーションの作成、J2EE サーバへのアプリケーションのデプロイ、J2EE サーバの起動・停止、データベースの操作などができます。

Cosminexus では、構成ソフトウェアとして、MyEclipse の日本語版環境である MyEclipse for Cosminexus を提供しています。

## RMI-IIOP

JavaRMI と JavaIDL を統合した API です。

## SessionBean

セッション単位での処理を行う機能がある、EJB で規定されている Bean の一つです。

## SOAP ( Simple Object Access Protocol )

ネットワーク上のサービス部品を呼び出す際のメッセージを規定したものです。HTTP と XML がベースとなっています。

## SQL オペレーション定義ファイル

DB アダプタを利用してデータベースにアクセスする場合に、実行する SQL を定義するファイルです。DB アダプタ用の XML フォーマット定義ファイルを作成するときの基となるファイルです。

## TP1 アダプタ

Cosminexus サービスプラットフォームの実行環境から既存の OpenTP1 システム内にあるサービス部品を呼び出す場合に利用するアダプタです。

## uCosminexus Service Architect

Cosminexus サービスプラットフォームの開発環境を構築・運用するための製品です。

## uCosminexus Service Platform

Cosminexus サービスプラットフォームの実行環境・運用環境を構築・運用するための製品です。

## W3C ( World Wide Web Consortium )

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。

## Web サービス

インターネット上に分散するアプリケーションが提供するサービス部品を、クライアントに API として提供するサービス部品です。

## WS-R ( Web Services Reliability )

標準化団体 OASIS で標準化された、インターネット向けのオープンな非同期通信プロトコルです。

## WSDL ( Web Service Description Language )

Web サービスを記述するための言語仕様です。XML がベースとなっています。

## XML スキーマ

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 フォーマット定義ファイル

XML 形式のデータを扱う場合に利用する電文フォーマット定義ファイルです。XML スキーマファイルとして作成します。

## XPath

XML 文書の中の特定の要素を示すための記述方法に関する W3C 勧告の標準仕様です。XML 形式のデータのツリー構造をたどって要素や属性にアクセスする手段が規定されています。

# ( ア行 )

---

## アクティビティ

ビジネスプロセスの処理フローを定義する部品です。基本アクティビティと構造アクティビティに分類されます。複数のアクティビティを画面上に配置し、コネクションで連結して、ビジネスプロセスの処理フローを定義します。

## アダプタ

サービスリクエストからの要求電文を受け取り、アダプタ内で定義されているサービス部品へ要求電文を送信して、サービス部品を呼び出すアプリケーションです。アダプタには、標準で使用するサービスアダプタ（標準アダプタ）と、ユーザが必要に応じて使用するカスタムアダプタ（DB アダプタ、TP1 アダプタなど）があります。サービスアダプタは、Web サービス、SessionBean、MDB (WS-R)、および MDB (DB キュー) のサービス部品を呼び出すアダプタです。カスタムアダプタは、それ以外のサービス部品を呼び出すアダプタです。

## 運用環境

開発環境で作成した EAR ファイルをリポジトリから読み込み、実行環境に配備する環境です。また、実行環境で利用する HCSC サーバをセットアップします。実行環境での業務の運用を開始したあとは、システムの起動・停止、および状態を監視します。

## 永続化

ビジネスプロセスのプロセスインスタンスの実行状況を履歴として、データベースに記録すること

です。

## 応答電文

サービス部品の実行結果を返すための電文です。応答電文には、サービス部品から HCSC サーバに送信される電文と、HCSC サーバからサービスリクエストに送信される電文があります。

## (カ行)

---

### 開発環境

実行環境でサービス部品を呼び出すために必要な HCSC コンポーネントを作成したり、作成した HCSC コンポーネントを HCSC サーバおよびクラスタにどのように配備するか定義（配備定義）したりする環境です。

### カスタムアダプタ

呼び出すサービス部品の種類が、Web サービス、SessionBean、MDB (WS-R)、MDB (DB キュー) 以外のアダプタです。

### 基本アクティビティ

受付処理、応答処理、サービス部品の呼び出しなど、ビジネスプロセス内の基本的な処理を定義するためのアクティビティです。

### 構造アクティビティ

分岐処理、並列処理、繰り返し処理など、ビジネスプロセス内の処理の構造を定義するためのアクティビティです。

### コネクション

アクティビティを連結して、アクティビティの実行順序を定義するための部品です。通常のコネクションに加えて、フォルト処理を連結するフォルトコネクションおよび並列処理時のリンクを設定するリンクコネクションがあります。

## (サ行)

---

### サービス

業務と一対一に対応し、業務に必要な機能を再利用できるソフトウェア部品のことです。SOA では、サービスを組み合わせることで業務システムを構築します。これによって、業務の変化に対して、システムを迅速に変更できます。

### サービスアダプタ

どのサービス部品を実行するかを定義した HCSC コンポーネントです。呼び出すサービス部品の種類（Web サービス、SessionBean、MDB (WS-R)、MDB (DB キュー)）ごとに定義する内容は異なります。

### サービス部品

サービスアダプタおよび DB アダプタから実行できる業務処理のことです。Web サービス、SessionBean、MDB (WS-R)、MDB (DB キュー)、およびデータベースの操作をサービス部品と

して利用できます。

### サービス部品電文

サービス部品独自の電文フォーマットを持った電文です。サービス部品がデータベースの操作の場合、DBアダプタ電文といえます。

### サービスリクエスト

HCSC サーバへサービス部品の実行要求（要求電文）を送信するアプリケーションです。HCSC サーバの標準受付が同期標準受付（Web サービス / SessionBean）か、非同期標準受付（MDB（WS-R） / MDB（DB キュー））か、またはユーザ定義受付（Web サービス）かによって、作成するサービスリクエストの形式が異なります。

標準受付にサービス部品の実行を要求するサービスリクエストを作成するときは、HCSC サーバに送信される要求電文の電文フォーマットを、標準電文に合わせる必要があります。

ユーザ定義受付にサービス部品の実行を要求するサービスリクエストを作成するときは、サービス部品またはビジネスプロセスに定義した電文フォーマット（WSDL）に合わせるすることができます。

### システム構成定義

HCSC サーバおよびクラスタのセットアップ情報と、HCSC コンポーネントを HCSC サーバおよびクラスタにどのように配備するかの情報が含まれる定義情報です。

HCSC サーバおよびクラスタのセットアップ情報は、運用環境で更新されます。開発環境では、HCSC サーバおよびクラスタに HCSC コンポーネントをどのように配備するかの情報を定義しません。

### 実行環境

サービスリクエストで受け付けた実行要求に応じて、HCSC サーバを介して適切なサービス部品を呼び出し、業務を実行する環境です。

### 関連セット

サービスリクエストから送信された要求電文を一意に識別するために利用する文字列です。HCSC サーバには、サービスリクエストから関連セットを含む要求電文が送信されます。サービス部品の HCSC サーバは、関連セットを含んだ応答電文をサービスリクエストに返信します。その結果、サービスリクエストは受信した応答電文が、どの要求電文に対する応答であるのかを一意に識別できます。

## ( 太行 )

---

### データ変換

標準電文とサービス部品電文の電文フォーマットが異なる場合に、データ変換定義に従って、標準電文をサービス部品電文の電文フォーマットに変換することです。

### データ変換定義

標準電文とサービス部品電文の電文フォーマットの相違を解消するためのデータ変換方法を定義したものです。標準電文とサービス部品電文の電文フォーマット定義ファイルの各要素をマッピングし、必要に応じてファンクションを設定して定義します。

### 電文フォーマット

要求電文および応答電文のデータ構造です。

サービスリクエストから HCSC サーバに送信される要求電文の電文フォーマットは、標準電文の電文フォーマットに合わせる必要があります。標準電文とサービス部品に実行要求するための電文（サービス部品電文）の電文フォーマットが異なる場合は、データ変換定義を作成して、電文フォーマットの相違を解消します。

### 電文フォーマット定義ファイル

電文の構造（電文フォーマット）を定義したファイルです。XML フォーマット定義ファイルとバイナリフォーマット定義ファイルとがあります。サービス部品の実行に利用する電文で XML 形式のデータを扱う場合は、XML フォーマット定義ファイルを、バイナリ形式（XML 形式以外）のデータを扱う場合は、バイナリフォーマット定義ファイルを利用します。

## (ナ行)

---

### ノード

データ変換定義画面で、ツリービューに表示される各要素のことです。データ変換の元になる変換元ノードと、変換先になる変換先ノードがあります。

## (ハ行)

---

### バイナリフォーマット定義ファイル

バイナリ形式（XML 形式以外）のデータを扱う場合に利用する電文フォーマット定義ファイルです。XML スキーマファイルとして作成します。バイナリ形式のデータ内の値の格納形式を定義し、定義した情報を XML スキーマファイルに付加したファイル（拡張子：.fdx）です。

### 配備

HCSC コンポーネントを適切な HCSC サーバに配置することです。開発環境で作成した HCSC コンポーネントは、運用環境を利用して実行環境（HCSC サーバ）に配備します。

### 配備定義

開発環境で、HCSC コンポーネントを HCSC サーバおよびクラスタにどのように配備するか定義し、システム構成定義を更新することです。運用環境から HCSC コンポーネントを配備するときには、開発環境で配備定義したシステム構成定義の内容に従って配備します。

### パッケージング

開発環境で作成した HCSC コンポーネントを、実行環境に配備できるように EAR ファイルに組み立てることです。

### ビジネスプロセス

複数のサービス部品の処理の順番や条件などを定義して、一連の業務の流れとして定義したものです。Cosminexus サービスプラットフォームの開発環境では、複数のアクティビティを画面上に配置し、コネクションで連結して定義します。

### 標準受付

サービスリクエストからの要求電文を受け付けるための機能（インターフェース）です。HCSC サーバに含まれる機能です。標準受付を開始すると、サービスリクエストからの要求電文を受け付

けられる状態になります。

同期受付 ( Web サービス / SessionBean ) および非同期受付 ( MDB ( WS-R ) / MDB ( DB キュー ) ) の要求を受け付けることができます。

## 標準電文 ( 標準電文フォーマット )

Cosminexus サービスプラットフォームで提供する標準的な電文フォーマットです。

## ファンクション

データ変換を定義する場合に、変換元ノードの値を加工して変換先ノードにマッピングする場合に使用するときの加工の方法を定義するものです。

文字列や数値を変換するファンクション、ノードの名称を取得するファンクション、条件による処理の選択や繰り返しを定義するファンクションなどがあります。

## プロセスインスタンス

ビジネスプロセスで実行している業務の一つの実体です。ワークフローの案件に相当します。

## ( マ行 )

---

### マッピング

データ変換を定義する場合に、変換元と変換先のノードを連結することです。マッピングの始点をマッピング元、終点をマッピング先といいます。

## ( ヤ行 )

---

### ユーザ定義受付

サービスリクエストからの実行要求を受け付けるための機能 ( インターフェース ) です。HCSC サーバに含まれる標準受付とは異なり、ユーザが任意のインターフェースを定義できます。定義したユーザ定義受付を HCSC サーバに配備して開始すると、サービスリクエストからの要求電文を受け付けられる状態になります。

### ユーザ電文

サービスリクエストから送信される要求電文です。

### 要求電文

サービス部品の実行を要求するための電文です。要求電文には、サービスリクエストから HCSC サーバに送信される電文と HCSC サーバ内のアダプタからサービス部品に送信される電文があります。

## ( ラ行 )

---

### リポジトリ

定義した HCSC コンポーネントやシステム構成定義を格納するディレクトリです。格納した情報は、リポジトリの管理機能 ( リポジトリのインポート / エクスポート機能 ) を利用して、開発環境と運用環境との間で受け渡します。

---

# 索引

## A

---

assign 要素の変換 690

## B

---

BPEL〔用語解説〕 775  
BPEL のサポート範囲 683  
BPEL ファイルの作成 169  
BPMN〔用語解説〕 775

## C

---

correlationSets 要素の変換 686  
Cosminexus RM のローカルキューを利用したサービス部品の適用範囲 67  
Cosminexus サービスプラットフォームの全体像 2  
Cosminexus サービスプラットフォームを構成する環境 3  
csamkxmls 635  
CSCMsgServerException クラス 393  
cscsvconfig.properties〔用語解説〕 776

## D

---

DB Connector〔用語解説〕 775  
DB アダプタ〔用語解説〕 775  
DB アダプタ電文 92  
DB アダプタに要求を出すサービスリクエスト 443  
DB アダプタ用の XML フォーマット定義ファイルの生成 635  
DB アダプタ用の XML フォーマット定義ファイルを作成する 109  
DB アダプタ連携基盤〔用語解説〕 776  
DB アダプタを新規に追加する 137  
DB アダプタを定義する 151  
DB キュー〔用語解説〕 775  
DB キューを利用したサービス部品の適用範囲 67

DB クライアント (HCSC サーバ) 側の設定 34  
DB サーバ側の設定 32

## E

---

EAR ファイル〔用語解説〕 775  
Eclipse〔用語解説〕 775  
Eclipse の設定ファイルの確認 28  
Eclipse への HCSCTE プラグインの組み込み 28  
EJB〔用語解説〕 775  
empty 要素の変換 690

## F

---

faultHandlers 要素の変換 686  
flow 要素の変換 693

## H

---

HCSC〔用語解説〕 775  
HCSC-Business Process〔用語解説〕 776  
HCSC-Data Transform〔用語解説〕 776  
HCSC-DB Adapter〔用語解説〕 776  
HCSCTE〔用語解説〕 776  
HCSC 簡易セットアップ画面 47  
HCSC簡易セットアップ画面で入力する項目 49  
HCSC 簡易セットアップ機能 43  
HCSC 簡易セットアップ機能で構築できる環境 43  
HCSC 簡易セットアップ機能でセットアップしたテスト環境の運用 51  
HCSC 簡易セットアップ機能の実行 45  
HCSC コンポーネント 776  
HCSC コンポーネント情報表示画面 494  
HCSC コンポーネントの一覧を更新する 358  
HCSC コンポーネントの情報の参照 357  
HCSC コンポーネントの情報を表示する 358

HCSCコンポーネントを作成する画面の構成  
456

HCSC サーバ〔用語解説〕 776

HCSC サーバランタイム定義ファイル〔用語  
解説〕 776

HTTP〔用語解説〕 776

## I

IDE〔用語解説〕 776

invoke 要素の変換 688

iWay アダプタ 23

## J

J2EE〔用語解説〕 776

J2EE サーバ〔用語解説〕 777

Java〔用語解説〕 777

Java 呼出アクティビティ 206

Java 呼出アクティビティダイアログ 539

JMS のサービスリクエスト 417

JMS メッセージを生成する〔標準の非同期  
受付 (DB キュー)〕 423

JMS メッセージを生成する〔標準の非同期  
受付 (MDB (WS-R))〕 405

JMS メッセージを送信する〔標準の非同期  
受付 (DB キュー)〕 429

JMS メッセージを送信する〔標準の非同期  
受付 (MDB (WS-R))〕 412

## M

MDB〔用語解説〕 777

MyEclipse〔用語解説〕 777

## P

process 要素の変換 684

## R

receive 要素の変換 687

reply 要素の変換 688

RMI-IIOP〔用語解説〕 777

## S

scope 要素の変換 691

SessionBean〔用語解説〕 777

SessionBean を利用したサービス部品の適用  
範囲 63

SOAP〔用語解説〕 777

SQL オペレーション定義ファイル 98

SQL オペレーション定義ファイル〔用語解  
説〕 777

SQL オペレーション定義ファイルの形式 99

SQL オペレーション定義ファイルの構造 99

SQL オペレーション定義ファイルの作成例  
107

SQL オペレーション定義ファイルを作成す  
る 98

switch 要素の変換 692

## T

throw 要素の変換 691

TP1/Server Base Enterprise Option ( TP1/  
EE ) のサービスリクエスト 417

TP1 アダプタ〔用語解説〕 777

## U

uCosminexus Service Adapter Architect for  
Flat Files 22

uCosminexus Service Adapter Architect for  
Object Access 23

uCosminexus Service Adapter Architect for  
TP1 22

uCosminexus Service Adapter for Flat Files  
22

uCosminexus Service Adapter for Object  
Access 23

uCosminexus Service Adapter for TP1 22

uCosminexus Service Architect〔用語解説〕  
777

uCosminexus Service Platform〔用語解説〕  
777

uCosminexus TP1/Server Base Enterprise  
Option 21

uCosminexus TP1 Gateway 22

uCosminexus 日立コード変換 -  
Development Kit 21  
uCosminexus 日立コード変換 -  
Development Kit のセットアップ 30  
uCosminexus 日立コード変換 - Runtime 22  
uCosminexus 日立コード変換 - Server  
Runtime 22

## V

---

variables 要素の変換 685

## W

---

W3C〔用語解説〕 778  
Web サービス〔用語解説〕 778  
Web サービスを利用したサービス部品の適用  
範囲 61  
while 要素の変換 692  
WS-R〔用語解説〕 778  
WSDL〔用語解説〕 778  
WSDL の作成 320  
WSDL の作成に使用するビジネスプロセス  
の例 320  
WSDL の作成例 322  
WSDL を取得する 366  
WSDL を編集する 437

## X

---

XML スキーマ〔用語解説〕 778  
XML スキーマの適用範囲 68  
XML フォーマット定義ファイル 95  
XML フォーマット定義ファイル〔用語解説〕  
778  
XPath〔用語解説〕 778  
XPath を指定する 625

## あ

---

アウトラインビュー 462  
アクティビティ 173  
アクティビティ〔用語解説〕 778  
アクティビティの定義 200  
アクティビティの定義情報を変更する 229

アクティビティの連結条件 176  
アクティビティ名を変更する 230  
アクティビティを配置する 174  
アクティビティを連結する 174  
値置換えダイアログ 604  
値置換えファンクション 285  
アダプタ〔用語解説〕 778  
アダプタ定義での設定一覧 155  
アダプタの検証 163  
アダプタの削除 164  
アダプタの追加 136  
アダプタの定義 133  
アダプタの内容の定義 139  
アダプタの編集 162  
アダプタの保存 161  
アプリケーション統合属性ファイル 152  
アンインストール 35

## い

---

移行の対象となるバージョン 676  
インスタンスを生成する 387  
インストール 27  
インポート〔プロジェクト〕 76  
インポート〔リポジトリ〕 81

## う

---

受付アクティビティ 201  
受付アクティビティダイアログ 533  
運用環境 4  
運用環境〔用語解説〕 778  
運用中のビジネスプロセス定義を変更する  
230

## え

---

永続化〔用語解説〕 778  
エクスポート〔プロジェクト〕 76  
エクスポート〔リポジトリ〕 80  
エラー情報を取得する〔標準の同期受付  
(SessionBean)〕 392  
エラー情報を取得する〔標準の同期受付  
(Web サービス)〕 377

エラー情報を取得する〔ユーザ定義受付 (Web サービス)] 439  
 エラーメッセージダイアログが出力された場合の注意事項 673

## お

---

応答アクティビティ 203  
 応答アクティビティダイアログ 535  
 応答電文 92  
 応答電文〔用語解説〕779  
 応答電文の形式〔DB アダプタ〕447  
 応答電文を取得する〔標準の同期受付 (SessionBean)] 392  
 応答電文を取得する〔標準の同期受付 (Web サービス)] 375  
 応答電文を取得する〔標準の非同期受付 (DB キュー)] 434  
 応答電文を取得する〔標準の非同期受付 (MDB (WS-R))] 415  
 応答電文を取得する〔ユーザ定義受付 (Web サービス)] 438  
 応答用のキューを設定する〔標準の非同期受付 (DB キュー)] 429  
 応答用のキューを設定する〔標準の非同期受付 (MDB (WS-R))] 413  
 応答を取り出す〔標準の非同期受付 (DB キュー)] 431  
 応答を取り出す〔標準の非同期受付 (MDB (WS-R))] 414  
 オブジェクトを切り取る 617  
 オブジェクトをコピーする 617  
 オブジェクトを削除する 619  
 オブジェクトを生成する〔標準の同期受付 (Web サービス)] 368  
 オブジェクトを生成する〔ユーザ定義受付 (Web サービス)] 438  
 オブジェクトを整理する 619  
 オブジェクトを貼り付ける 617

## か

---

開始アクティビティ 201  
 下位バージョンからの移行手順 676

下位バージョンからのリポジトリの移行 676  
 下位バージョンで開発環境と運用環境のリポジトリを共有していた場合の移行手順 679  
 開発から実運用までの流れ 11  
 開発環境 3  
 開発環境〔用語解説〕779  
 開発環境とテスト環境との関係 6  
 開発環境を使用するためのセットアップ 26  
 開発手順 38  
 拡張関数 627  
 カスタムアダプタ〔用語解説〕779  
 画面一覧 458  
 画面構成 454  
 環境設定 30  
 環境の関係 4  
 関連ソフトウェア 21

## き

---

基本アクティビティ 487  
 基本アクティビティ〔用語解説〕779  
 基本アクティビティに関連する要素の変換 687  
 キャンバス 488

## く

---

空白除去ダイアログ 581  
 空白除去ファンクション 268  
 組み込みデータベースの構築 32  
 クライアント定義ファイルの作成と設定 (SessionBean) 143  
 クライアント定義ファイルの作成と設定 (Web サービス) 140  
 クラスタから HCSC コンポーネントを削除する 356  
 クラスタに HCSC コンポーネントを追加する 355  
 繰り返しアクティビティ 217  
 繰り返しアクティビティダイアログ 549  
 くり返し依存対象が設定された変換元ノードのパスの表示 298  
 くり返し依存対象の対応づけ 293  
 くり返し依存対象を確認する 297

くり返し依存対象を変更する 298  
 くり返し設定ダイアログ 564  
 くり返し設定ダイアログを使用してマッピングする 291  
 くり返しダイアログ 594  
 くり返しの設定 291  
 くり返しファンクション 278  
 くり返しを合成する 292  
 くり返しを対応づける 278  
 グローバル定義の単純内容要素を設定する 114  
 グローバル定義の要素を構成要素に設定する 116  
 グローバル変数 185

## け

---

検証〔用語解説〕 776  
 検証結果の表示〔バイナリフォーマット定義ファイル〕 131  
 検証設定 75  
 検証内容〔バイナリフォーマット定義ファイル〕 122  
 検証内容〔ビジネスプロセス〕 233  
 検証内容〔ユーザ定義受付〕 345  
 検証内容の表示〔アダプタ〕 163  
 検証内容の表示〔ビジネスプロセス〕 244  
 検証方法〔アダプタ〕 163  
 検証方法〔バイナリフォーマット定義ファイル〕 130  
 検証方法〔ビジネスプロセス〕 244  
 検証方法〔ユーザ定義受付〕 346

## こ

---

構成ソフトウェア 19  
 構成ソフトウェアの位置づけ 20  
 構成ソフトウェアの種類 19  
 構成の形態と選択 36  
 構成要素 115  
 構成要素ダイアログ 509  
 構成要素の開始セパレータを選択して選択条件を設定する 118  
 構成要素の選択条件を設定する 117

構成要素を設定する 115  
 構造アクティビティ 487  
 構造アクティビティ〔用語解説〕 779  
 構造アクティビティに関連する要素の変換 691  
 コネクション〔用語解説〕 779  
 コマンドの入力形式 632  
 コマンドの文法の記述形式 634  
 コンソールビュー 457  
 コンソールビュー〔バイナリフォーマット定義時〕 455

## さ

---

サービス〔用語解説〕 779  
 サービスアダプタ〔用語解説〕 779  
 サービスアダプタ (MDB (DB キュー)) を定義する 150  
 サービスアダプタ (MDB (WS-R)) を定義する 147  
 サービスアダプタ (SessionBean) を定義する 142  
 サービスアダプタ (Web サービス) を定義する 139  
 サービスアダプタ定義画面 464  
 サービスアダプタ定義画面〔カスタムアダプタの基本画面〕 479  
 サービスアダプタ定義画面〔カスタムアダプタの詳細画面〕 483  
 サービスアダプタ定義画面 (MDB (DB キュー)) の基本画面) 478  
 サービスアダプタ定義画面 (MDB (WS-R)) の基本画面) 474  
 サービスアダプタ定義画面 (MDB (WS-R)) の詳細画面) 476  
 サービスアダプタ定義画面 (SessionBean の基本画面) 471  
 サービスアダプタ定義画面 (SessionBean の詳細画面) 472  
 サービスアダプタ定義画面 (Web サービスの基本画面) 464  
 サービスアダプタ定義画面 (Web サービスの詳細画面) 469

サービスアダプタ定義画面での設定 (カスタムアダプタの場合) 158  
 サービスアダプタ定義画面での設定 (サービスアダプタの場合) 156  
 サービスアダプタ定義画面での操作 [DB アダプタ] 154  
 サービスアダプタ定義画面での操作 [MDB (DB キュー)] 150  
 サービスアダプタ定義画面での操作 [MDB (WS-R)] 148  
 サービスアダプタ定義画面での操作 [SessionBean] 145  
 サービスアダプタ定義画面での操作 [Web サービス] 141  
 サービスアダプタの新規追加 (MDB (WS-R) または DB キュー) の場合 137  
 サービスアダプタの新規追加 (SessionBean の場合) 137  
 サービスアダプタの新規追加 (Web サービスの場合) 136  
 サービス部品 [用語解説] 779  
 サービス部品電文 92  
 サービス部品電文 [用語解説] 780  
 サービス部品電文を作成する (DB アダプタ電文の場合) 98  
 サービス部品電文を作成する (MDB (WS-R) または DB キュー) の場合 98  
 サービス部品電文を作成する (SessionBean の場合) 98  
 サービス部品電文を作成する (Web サービスの場合) 96  
 サービス呼出アクティビティ 204  
 サービス呼出アクティビティダイアログ 537  
 サービスリクエスト 362  
 サービスリクエスト [用語解説] 780  
 サービスリクエスト (標準の同期受付 (SessionBean)) の作成手順 385  
 サービスリクエスト (標準の同期受付 (Web サービス)) の作成手順 364  
 サービスリクエスト (標準の非同期受付 (DB キュー)) の作成手順 418  
 サービスリクエスト (標準の非同期受付 (MDB (WS-R))) の作成手順 402

サービスリクエスト (ユーザ定義受付 (Web サービス)) の作成手順 436  
 サービスリクエストが使用するプロトコルと標準受付およびユーザ定義受付の種類 362  
 サービスリクエストの作成 361  
 削除できないアダプタ 164  
 削除できないビジネスプロセス 246  
 参照できる HCSC コンポーネントの情報 357

---

## し

システム構成定義 [用語解説] 780  
 システム全体と開発環境の関係 16  
 実行環境 3  
 実行環境 [用語解説] 780  
 指定した値を代入する 283  
 指定文字列の有無, または指定文字列の開始を調査する 266  
 終了アクティビティ 227  
 順次処理アクティビティ 227  
 条件設定ダイアログ (繰り返しアクティビティ) 550  
 条件設定ダイアログ (ノード条件の設定) 569  
 条件設定ダイアログ (分岐アクティビティ) 553  
 条件設定ダイアログ (並列処理アクティビティ) 557  
 条件値設定ダイアログ 515  
 条件追加 / 編集ダイアログ 600  
 条件によって異なる値を出力する 280  
 小数点以下の端数を処理する 272

---

## す

数値演算ダイアログ 585  
 数値演算ファンクション 271  
 数値フォーマット変換ダイアログ 583  
 数値フォーマット変換ファンクション 270  
 数値を演算する 271  
 数値をフォーマット変換する 270  
 スキーマ論理名 490  
 スコープアクティビティ 216

スコープアクティビティダイアログ 548  
 スタブを取得する 387  
 スタブを生成する〔標準の同期受付 (Web サービス)〕 367  
 スタブを生成する〔ユーザ定義受付 (Web サービス)〕 438  
 ステータスの永続化 170

## せ

---

接続できるマッピング線の数 310  
 設定ダイアログ 572  
 セパレータ選択ダイアログ 521  
 セパレータ追加 / 変更ダイアログ 522  
 セパレータを設定する 119  
 選択条件ノードを選択して選択条件を設定する 117  
 選択ダイアログ 597  
 選択ファンクション 280  
 前提 OS 19  
 前提条件 27

## そ

---

関連セット 173  
 関連セット〔用語解説〕 780  
 関連セットによるプロセスインスタンスの識別 191  
 関連セットの定義 191  
 関連セットの定義方法 196  
 関連セットを利用した定義の例 197  
 関連セットを割り当てられるアクティビティ 193  
 操作を元に戻す〔データ変換定義画面〕 629  
 操作を元に戻す〔バイナリフォーマット定義エディタ〕 616  
 操作を元に戻す〔ビジネスプロセス定義画面〕 619  
 操作をやり直す〔データ変換定義画面〕 629  
 操作をやり直す〔バイナリフォーマット定義エディタ〕 616  
 操作をやり直す〔ビジネスプロセス定義画面〕 620  
 送信用共用キューを作成する 422

総和ダイアログ 588  
 総和ファンクション 274  
 ソート条件追加 / 編集ダイアログ 596  
 属性 490  
 ソフトウェアの構成 19

## た

---

代入アクティビティ 212  
 代入アクティビティサブダイアログ 544  
 代入アクティビティダイアログ 543  
 単純内容 490  
 単純内容要素 114  
 単純内容要素ダイアログ 500

## つ

---

ツリービュー 456

## て

---

定義済みアダプタを利用してアダプタを追加する 138  
 定義済みビジネスプロセスを利用してビジネスプロセスを追加する 172  
 定義の流れ〔アダプタ〕 134  
 定義の流れ〔ビジネスプロセス〕 166  
 定義の流れ〔ユーザ定義受付〕 314  
 定義領域 456  
 定義を保存する 629  
 定数ダイアログ 602  
 定数ファンクション 283  
 データベースと Cosminexus RM の使用有無を確認する 41  
 データ変換 92  
 データ変換〔用語解説〕 780  
 データ変換 (マッピング) 定義 250  
 データ変換 (マッピング) 定義画面 489  
 データ変換 (マッピング) 定義画面の操作 629  
 データ変換 (マッピング) に関連するダイアログ 561  
 データ変換 (マッピング) の定義手順 250  
 データ変換アクティビティ 210  
 データ変換アクティビティダイアログ 541

データ変換基盤〔用語解説〕 776  
 データ変換定義〔用語解説〕 780  
 データ変換定義を作成する画面の構成 456  
 データ変換に必要なファイルおよび定義 248  
 データ変換の定義 247  
 テスト環境と本番環境との関係 7  
 テスト環境のアンセットアップ 50  
 テスト環境の運用に必要な情報 52  
 テスト環境のカスタマイズ 57  
 テスト環境の簡易セットアップ 43  
 テスト環境の起動と停止 51  
 テスト環境のセットアップ 45  
 テスト環境の前提ソフトウェア 45  
 転送キューを作成する 404  
 電文の種類 92  
 電文の流れ 92  
 電文フォーマット 92  
 電文フォーマット〔用語解説〕 780  
 電文フォーマット (XML フォーマット定義  
 ファイル) の作成方法 96  
 電文フォーマット (バイナリフォーマット定  
 義ファイル) の作成方法 111  
 電文フォーマット定義ファイル 92  
 電文フォーマット定義ファイル〔用語解説〕  
 781  
 電文フォーマット定義ファイルの作成 249  
 電文フォーマットの作成 91  
 電文フォーマットの種類 95  
 電文フォーマットの取込ダイアログ 530

## と

---

動作環境 19

## の

---

ノード〔用語解説〕 781  
 ノード条件設定ダイアログ 567  
 ノード条件を設定する 303  
 ノード数取得ダイアログ 590  
 ノード数取得ファンクション 275  
 ノード数を代入する 275  
 ノード選択ダイアログ (データ変換) 565

ノード選択ダイアログ (バイナリフォーマッ  
 ト定義) 516  
 ノード調査ダイアログ 592  
 ノード調査ファンクション 277  
 ノードの有無を調査する 277  
 ノード名取得ダイアログ 591  
 ノード名取得ファンクション 276  
 ノード名を代入する 276

## は

---

バイナリデータを作成する 423  
 バイナリデータを受信用共用キューに設定す  
 る 429  
 バイナリフォーマット定義エディタ 459  
 バイナリフォーマット定義画面 459  
 バイナリフォーマット定義画面の操作 610  
 バイナリフォーマット定義に関連するダイア  
 ログ 499  
 バイナリフォーマット定義ファイル 95  
 バイナリフォーマット定義ファイル〔用語解  
 説〕 781  
 バイナリフォーマット定義ファイルの作成の  
 流れ 111  
 バイナリフォーマット定義ファイルを検証す  
 る 122  
 バイナリフォーマット定義ファイルを作成す  
 る画面の構成 454  
 バイナリフォーマット定義ファイルを新規作  
 成する 112  
 バイナリフォーマット定義ファイルを編集す  
 る 122  
 配備〔用語解説〕 781  
 配備定義 353  
 配備定義〔用語解説〕 781  
 配備できるクラスタ (または単一の HCSC  
 サーバ) 354  
 端数処理ダイアログ 587  
 端数処理ファンクション 273  
 パッケージ・エクスプローラー 454  
 パッケージング 352  
 パッケージング〔用語解説〕 781  
 パッケージングから配備定義までの流れ 350

パラメタを設定する〔標準の同期受付  
 (SessionBean)〕 389  
 パラメタを設定する〔標準の同期受付 (Web  
 サービス)〕 371  
 パラメタを設定する〔標準の非同期受付  
 (DB キュー)〕 427  
 パラメタを設定する〔標準の非同期受付  
 (MDB (WS-R))〕 406  
 パレット〔データ変換 (マッピング) 定義画  
 面〕 491  
 パレット〔ビジネスプロセス定義画面〕 485  
 パレット設定ダイアログ 558  
 パレットの表示を設定する 621

## ひ

ビジネスプロセス〔用語解説〕 781  
 ビジネスプロセス基盤〔用語解説〕 776  
 ビジネスプロセス全体の定義に関連する要素  
 の変換 684  
 ビジネスプロセス定義画面 484  
 ビジネスプロセス定義画面の操作 617  
 ビジネスプロセスに関連するダイアログ 525  
 ビジネスプロセスの検証 233  
 ビジネスプロセスの再実行を要求するサービ  
 スリクエストの作成 (SessionBean) 398  
 ビジネスプロセスの再実行を要求するサービ  
 スリクエストの作成 (Web サービス) 381  
 ビジネスプロセスの削除 246  
 ビジネスプロセスの追加 168  
 ビジネスプロセスの定義 165  
 ビジネスプロセスの定義情報を変更する 229  
 ビジネスプロセスの内容の定義 173  
 ビジネスプロセスの編集 229  
 ビジネスプロセスの保存 228  
 ビジネスプロセスを新規に追加する 168  
 ビジネスプロセスを新規に追加する〔BPEL  
 ファイルのインポート〕 168  
 ビジネスプロセスを新規に追加する〔未定義  
 のビジネスプロセス〕 168  
 一つの文字列から一部分の文字列を取り出す  
 264  
 標準受付〔用語解説〕 781  
 標準電文 92

標準電文〔用語解説〕 782  
 標準電文フォーマット〔用語解説〕 782  
 標準電文を作成する 96  
 標準の同期受付 (SessionBean) に要求を出  
 すサービスリクエスト 385  
 標準の同期受付 (Web サービス) に要求を出  
 すサービスリクエスト 364  
 標準の非同期受付 (DB キュー) に要求を出  
 すサービスリクエスト 417  
 標準の非同期受付 (MDB (WS-R)) に要求  
 を出すサービスリクエスト 402  
 表示を拡大する 621  
 表示を縮小する 621  
 表示を元のサイズに戻す 621

## ふ

ファンクション〔用語解説〕 782  
 ファンクション選択ダイアログ 566  
 ファンクションを削除する 259  
 ファンクションを使用した値の加工 262  
 フォーマット情報を設定する〔CSV 以外の  
 形式の場合〕 114  
 フォーマット情報を設定する〔CSV 形式の  
 場合〕 120  
 フォーマットダイアログ 499  
 フォルト処理の割当ダイアログ 532  
 フォルト処理を定義する 178  
 フォルト送出アクティビティ 215  
 フォルト送出アクティビティダイアログ 547  
 複合内容 490  
 複合内容要素 114  
 複合内容要素ダイアログ 505  
 複合内容要素を設定する 114  
 複数のノードセットのノードの数値を合計す  
 る 274  
 複数の文字列を連結する 262  
 部分文字列取得ダイアログ 576  
 部分文字列取得ファンクション 264  
 プロジェクトの管理 72  
 プロジェクトを削除する 77  
 プロジェクトを作成する 72  
 プロセスインスタンス〔用語解説〕 782  
 プロセスインスタンスの識別 191

プロパティビュー 457  
 プロパティビュー〔バイナリフォーマット  
 定義時〕455  
 プロパティを設定する 75  
 分岐アクティビティ 219  
 分岐アクティビティダイアログ 552

---

## へ

並列処理アクティビティ 222  
 並列処理アクティビティダイアログ 555  
 ヘッダー要素およびレコード要素の構成要素  
 を設定する 121  
 ヘッダー要素およびレコード要素の出現回数  
 を設定する 121  
 変換先スキーマツリービューア 490  
 変換テーブルの作成 286  
 変換テーブルの保存 287  
 変換テーブルを利用して値を変換する 285  
 変換元スキーマツリービューア 490  
 変換元ノードの値にある値を付加する 288  
 変換元ノードの値を加工して変換先ノードに  
 マッピングする 254  
 変換元ノードの値を倍にする 289  
 変換元ノードの値を変換先ノードに直接マッ  
 pingする 253  
 編集できる定義ファイル 57  
 変数 173  
 変数・相関セット一覧ダイアログ 525  
 変数の型 185  
 変数の種類 185  
 変数の定義 185  
 変数の定義方法 190  
 変数の表示 186  
 変数の表示ダイアログ 529  
 変数を割り当てられるアクティビティ 185

---

## ま

マッピング〔用語解説〕782  
 マッピングする 253  
 マッピング線の色を変更する 629  
 マッピング線を削除する 259  
 マッピング対象とマッピング対象外 305

マッピングできるノードおよびファンクショ  
 ンの対応 306  
 マッピングの条件 305  
 マッピングの注意事項 261  
 マッピングビューア 490  
 マッピング元設定ダイアログ 562  
 マッピング元の表示形式 260  
 マッピングを取り消す 259

---

## む

無操作アクティビティ 214  
 無操作アクティビティダイアログ 547

---

## め

メッセージの形式 638  
 メッセージの種類 638

---

## も

文字数取得ダイアログ 578  
 文字数取得ファンクション 265  
 文字列調査ダイアログ 580  
 文字列調査ファンクション 267  
 文字列の空白を除去する 268  
 文字列の文字数を代入する 265  
 文字列連結ダイアログ 574  
 文字列連結ファンクション 262  
 問題ビュー 455

---

## ゆ

ユーザ定義受付 312  
 ユーザ定義受付〔用語解説〕782  
 ユーザ定義受付（Web サービス）に要求を出  
 すサービスリクエスト 435  
 ユーザ定義受付削除ダイアログ 608  
 ユーザ定義受付情報表示画面 497  
 ユーザ定義受付定義画面 493  
 ユーザ定義受付に関連するダイアログ 608  
 ユーザ定義受付の概要 312  
 ユーザ定義受付の検証 345  
 ユーザ定義受付の削除 347  
 ユーザ定義受付の追加 341

ユーザ定義受付の定義 311  
 ユーザ定義受付の内容の確認 344  
 ユーザ定義受付を新規に追加する 341  
 ユーザ電文〔用語解説〕 782

## よ

---

要求電文 92  
 要求電文〔用語解説〕 782  
 要求電文の形式〔DBアダプタ〕 443  
 要求電文を作成する〔標準の同期受付  
 (SessionBean)〕 391  
 要求電文を作成する〔標準の同期受付 (Web  
 サービス)〕 373  
 要求電文を作成する〔標準の非同期受付  
 (DBキュー)〕 427  
 要求電文を作成する〔標準の非同期受付  
 (MDB (WS-R))〕 411  
 要素選択ダイアログ 520  
 要素を移動する 612  
 要素をコピーする 610  
 要素を削除する 612  
 要素を定義する (CSV 以外の形式の場合)  
 113  
 要素を定義する (CSV 形式の場合) 119  
 要素を貼り付ける 610

## り

---

リポジトリ〔用語解説〕 782  
 リポジトリの管理 78  
 リポジトリの指定 78  
 リポジトリの初期化 79  
 リポジトリの設定 75  
 リポジトリの変更 80  
 リポジトリを設定する 78  
 利用できるサービス部品の種類と適用範囲  
 61  
 リンクダイアログ 556

## る

---

ルート要素選択ダイアログ 561  
 ルート要素を設定する〔CSV 以外の形式の  
 場合〕 118

ルート要素を設定する〔CSV 形式の場合〕  
 121  
 ルート要素を変更する 251

## ろ

---

ローカル定義の単純内容要素を構成要素とし  
 て設定する 116  
 ローカル変数 185

## わ

---

割当相関セット群ダイアログ 528



# ソフトウェアマニュアルのサービス ご案内

## 1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しています。

URL <http://www.hitachi.co.jp/soft/manual/>

ホームページのメニューは次のとおりです。

マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル	日立ソフトウェアマニュアルと製品群別CD-ROMマニュアルの仕様について記載しています。
マニュアルのご購入	マニュアルご購入時のお申し込み方法を記載しています。
オンラインマニュアル	一部製品のマニュアルをインターネットで公開しています。
サポートサービス	ソフトウェアサポートサービスお客様向けページでのマニュアル公開サービスを記載しています。
ご意見・お問い合わせ	マニュアルに関するご意見、ご要望をお寄せください。

## 2. インターネットでのマニュアル公開

2種類のマニュアル公開サービスを実施しています。

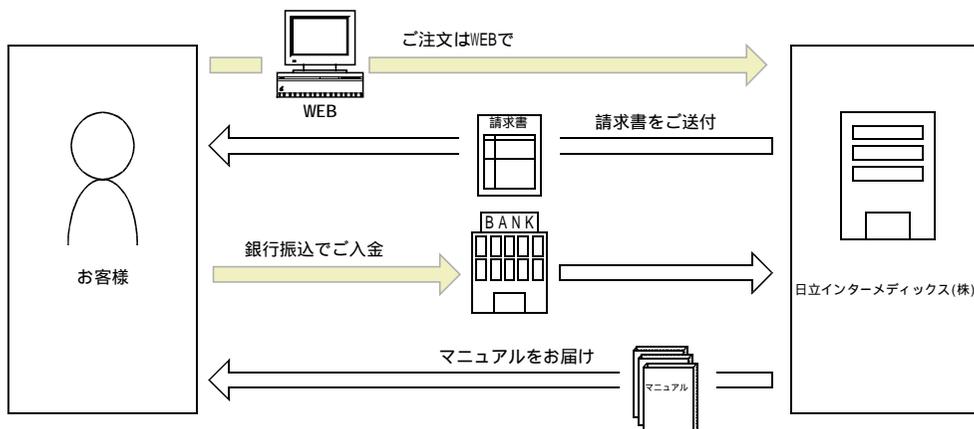
### (1) マニュアル情報ホームページ「オンラインマニュアル」での公開

製品をよりご理解いただくためのご参考として、一部製品のマニュアルを公開しています。

### (2) ソフトウェアサポートサービスお客様向けページでのマニュアル公開

ソフトウェアサポートサービスご契約のお客様向けにマニュアルを公開しています。公開しているマニュアルの一覧、本サービスの対象となる契約の種別などはマニュアル情報ホームページの「サポートサービス」をご参照ください。

## 3. マニュアルのご注文



マニュアル情報ホームページの「マニュアルのご購入」にアクセスし、お申し込み方法をご確認のうえWEBからご注文ください。ご注文先は日立インターメディアックス(株)となります。

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。

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。