

Cosminexus

ビジネスプロセス管理 / エンタープライズサービスバス V8

サービスプラットフォーム サンプルプログラムガイド

手引・文法・操作書

3020-3-U55-80

対象製品

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows XP , Windows Vista , Windows 7 x86 , Windows 7 x64 ¹

P-2443-7T84 uCosminexus Service Architect 08-70

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows Server 2003 (x64) ¹ , Windows Server 2003 R2 (x64) ¹ , Windows Server 2008 x86 , Windows Server 2008 x64 ¹ , Windows Server 2008 R2

P-2443-7S84 uCosminexus Service Platform 08-70

適用 OS : Windows Server 2003 (x64) , Windows Server 2003 R2 (x64) , Windows Server 2008 x64 , Windows Server 2008 R2

P-2943-7S84 uCosminexus Service Platform 08-70 ²

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

P-1J43-7S81 uCosminexus Service Platform 08-70 ²

適用 OS : AIX V7.1 , AIX V6.1 , AIX 5L V5.3

P-1M43-7S81 uCosminexus Service Platform 08-70 ²

適用 OS : 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) , Red Hat Enterprise Linux Server 6 (32-bit x86) , Red Hat Enterprise Linux Server 6 (64-bit x86_64)

P-9S43-7S81 uCosminexus Service Platform 08-70 ²

適用 OS : Red Hat Enterprise Linux AS 4 (IPF) , Red Hat Enterprise Linux 5 Advanced Platform (Intel Itanium) , Red Hat Enterprise Linux 5 (Intel Itanium)

P-9V43-7S81 uCosminexus Service Platform 08-70 ²

適用 OS : Windows Server 2003 , Windows Server 2003 R2 , Windows XP , Windows Vista , Windows 7 x86 , Windows 7 x64 ¹

P-2443-CL14 uCosminexus Service Adapter Architect for FTP 01-10

適用 OS : Windows Server 2003 , Windows Server 2003 (x64) ¹ , Windows Server 2003 R2 , Windows Server 2003 R2 (x64) ¹ , Windows Server 2008 x86 , Windows Server 2008 x64 ¹ , Windows Server 2008 R2 , HP-UX 11i V2 (IPF) , HP-UX 11i V3 (IPF) , AIX V7.1 , AIX V6.1 , AIX 5L V5.3 , 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) , Red Hat Enterprise Linux Server 6 (32-bit x86) , Red Hat Enterprise Linux Server 6 (64-bit x86_64)

P-2443-CK14 uCosminexus Service Adapter for FTP 01-10

注 1 WOW64 (Windows On Windows 64) 環境だけで使用できます。

注 2 製品の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、米国およびその他の国における International Business Machines Corporation の商標です。

AIX 5L は、米国およびその他の国における International Business Machines Corporation の商標です。

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。

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

Internet Explorer は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Itanium は、アメリカ合衆国およびその他の国における Intel Corporation の商標です。

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。

J2EE は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Java は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

JDBC は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

JDK は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Microsoft は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

MyEclipse は、米国 Genuitec 社の商品名称です。

OMG, CORBA, IIOP, 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。

Red Hat は、米国およびその他の国で Red Hat, Inc. の登録商標もしくは商標です。

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。

UNIX は、The Open Group の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。

WebSphere は、米国およびその他の国における International Business Machines Corporation の商標です。

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

Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

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

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/>).

マイクロソフト製品の表記について

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

表記			製品名
Internet Explorer 6			Microsoft(R) Internet Explorer 6
Internet Explorer 7			Windows(R) Internet Explorer(R) 7
Internet Explorer 8			Windows(R) Internet Explorer(R) 8
Internet Explorer 9			Windows(R) Internet Explorer(R) 9
Microsoft Cluster Service			Microsoft(R) Cluster Service
Windows	Windows 7	Windows 7 x86	Microsoft(R) Windows(R) 7 Enterprise(32 ビット版)
			Microsoft(R) Windows(R) 7 Professional(32 ビット版)
			Microsoft(R) Windows(R) 7 Ultimate(32 ビット版)
		Windows 7 x64	Microsoft(R) Windows(R) 7 Enterprise(64 ビット版)
			Microsoft(R) Windows(R) 7 Professional(64 ビット版)
			Microsoft(R) Windows(R) 7 Ultimate(64 ビット版)
	Windows Server 2003	Windows Server 2003 Enterprise Edition	Microsoft(R) Windows Server(R) 2003, Enterprise Edition Operating System (x86)
		Windows Server 2003 Standard Edition	Microsoft(R) Windows Server(R) 2003, Standard Edition Operating System (x86)
	Windows Server 2003 R2	Windows Server 2003 R2 Enterprise Edition	Microsoft(R) Windows Server(R) 2003 R2, Enterprise Edition Operating System (x86)
		Windows Server 2003 R2 Standard Edition	Microsoft(R) Windows Server(R) 2003 R2, Standard Edition Operating System (x86)
	Windows Server 2003 (x64)	Windows Server 2003 Enterprise x64 Edition	Microsoft(R) Windows Server(R) 2003, Enterprise x64 Edition Operating System
		Windows Server 2003 Standard x64 Edition	Microsoft(R) Windows Server(R) 2003, Standard x64 Edition Operating System
	Windows Server 2003 R2 (x64)	Windows Server 2003 R2 Enterprise x64 Edition	Microsoft(R) Windows Server(R) 2003 R2, Enterprise x64 Edition Operating System

表記			製品名
	Windows Server 2003 R2 Standard x64 Edition		Microsoft(R) Windows Server(R) 2003 R2, Standard x64 Edition Operating System
	Windows Server 2008 x86		Microsoft(R) Windows Server(R) 2008 Enterprise 32-bit
			Microsoft(R) Windows Server(R) 2008 Standard 32-bit
	Windows Server 2008 x64		Microsoft(R) Windows Server(R) 2008 Enterprise x64 Edition
			Microsoft(R) Windows Server(R) 2008 Standard x64 Edition
	Windows Server 2008 R2		Microsoft(R) Windows Server(R) 2008 R2 Enterprise
			Microsoft(R) Windows Server(R) 2008 R2 Standard
	Windows Vista	Windows Vista Business	Microsoft(R) Windows Vista(R) Business
		Windows Vista Enterprise	Microsoft(R) Windows Vista(R) Enterprise
		Windows Vista Ultimate	Microsoft(R) Windows Vista(R) Ultimate
	Windows XP		Microsoft(R) Windows(R) XP Professional Operating System

発行

2011 年 8 月 3020-3-U55-80

著作権

All Rights Reserved. Copyright (C) 2008, 2011, Hitachi, Ltd.

変更内容

変更内容 (3020-3-U55-80) uCosminexus Service Architect 08-70 , uCosminexus Service Platform 08-70 , uCosminexus Service Adapter Architect for FTP 01-10 , uCosminexus Service Adapter for FTP 01-10

追加・変更内容	変更箇所
Cosminexus サービスプラットフォームを構成するソフトウェア製品ごとの前提 OS に、次の OS を追加した。 <ul style="list-style-type: none">• AIX V7.1• Red Hat Enterprise Linux Server 6 (32-bit x86)• Red Hat Enterprise Linux Server 6 (64-bit x86_64)	-
「このマニュアルの読み方」および「このマニュアルで体験できること」を追加した。	1 章
Eclipse Classic 3.6.1 に対応したことに伴い、説明および画面を変更した。	3.2.2 , 3.3 , 3.4.1 , 3.5.1 , 3.5.3 , 3.5.4 , 3.5.5 , 3.5.6 , 3.5.7 , 4.5.3 , 5.2 , 5.3 , 5.4.1 , 5.4.2 , 5.4.3 , 5.5.1 , 5.6.1 , 5.6.2 , 5.6.3 , 5.7 , 5.9.2 , 5.9.3 , 5.9.4 , 5.10.1
各サンプルプログラムのファイルの構成を変更した。	付録 A.1 , 付録 A.2 , 付録 A.3

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは、Cosminexus サービスプラットフォームを使って SOA に対応したシステムを開発する方法について、サンプルプログラムを通じて理解を深めていただくことを目的としています。

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。

- uCosminexus Service Architect
- uCosminexus Service Platform
- uCosminexus Service Adapter Architect for FTP
- uCosminexus Service Adapter for FTP

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
- リレーショナルデータベースおよびデータベース操作言語 (SQL) 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

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、HP-UX 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。
HP-UX の場合	HP-UX に該当する表記です。
Linux の場合	Linux に該当する表記です。
UNIX の場合	UNIX (AIX、HP-UX 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。

画面や操作の説明で使用している記号

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を次に示します。

記号	意味
[]	ダイアログ名，キー名称，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およびツリーの項目から，- の後ろのメニューおよびツリーの項目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

使用している構文要素

このマニュアルで使用してい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 すべて半角文字を使用してください。

目次

1	サンプルプログラムをご使用になる前に	1
1.1	このマニュアルの読み方	2
1.2	このマニュアルで体験できること	3
2	サンプルプログラムの紹介	5
2.1	サンプルプログラムのシステム構成	6
2.2	サンプルプログラムを構成している要素	9
2.3	サンプルプログラムの処理の内容	10
2.3.1	サービスリクエストからサービスを呼び出すパターン	10
2.3.2	ビジネスプロセスからサービスを呼び出すパターン	10
2.3.3	プロセス統合パターン	12
3	サンプルプログラムの環境を準備する	15
3.1	インストールとセットアップの流れ	16
3.2	インストールと前提環境の準備	17
3.2.1	uCosminexus Service Architect のインストール	17
3.2.2	Eclipse のアーカイブファイルのダウンロード	17
3.3	MyEclipse セットアップ機能を使用したセットアップ	19
3.4	開発環境の設定	25
3.4.1	Eclipse の設定	25
3.4.2	WSDL4J のインストール	28
3.5	実行環境の設定	30
3.5.1	テスト環境の構築	31
3.5.2	テスト環境の起動	34
3.5.3	Management Server リモート管理機能へのログイン	34
3.5.4	MyEclipse の設定	35
3.5.5	MyEclipse プロジェクトのインポート	40
3.5.6	Web プロジェクトのデプロイ	45
3.5.7	HCSC サーバへの定義内容の配備	52

4

サンプルプログラムを実行する	57
4.1 サンプルプログラムの実行	58
4.2 ビジネスプロセスを適用しないパターン	59
4.3 ビジネスプロセスを適用するパターン	61
4.4 複数サービスのプロセスを統合したパターン	63
4.5 プロジェクトの削除	67
4.5.1 HCSC サーバへ配備した定義内容の削除	68
4.5.2 Web プロジェクトのアンデプロイ	68
4.5.3 MyEclipse プロジェクトの削除	69
4.6 テスト環境の停止	70

5

サンプルプログラムの開発を体験する	71
5.1 サンプルプログラムの開発の手順	72
5.2 HCSCTE プロジェクトの作成	75
5.3 システム構成定義のエクスポートとインポート	78
5.4 サービスアダプタの定義	80
5.4.1 サービスアダプタの作成	80
5.4.2 サービスアダプタの検証とパッケージング	84
5.4.3 サービスアダプタの配備定義	86
5.5 ビジネスプロセスの定義	87
5.5.1 ビジネスプロセスの作成	87
5.5.2 Hello ビジネスプロセスの検証とパッケージング	101
5.5.3 Hello ビジネスプロセスの配備定義	102
5.6 商品手配システムの開発	103
5.6.1 在庫管理サービスアダプタの定義	103
5.6.2 配送受付サービスアダプタの定義	106
5.6.3 商品手配ビジネスプロセスの定義	109
5.6.4 コンポーネントの検証とパッケージング	134
5.6.5 コンポーネントの配備定義	134
5.7 商品手配システムのデバッグ	135
5.8 開発情報のエクスポートとインポート	146
5.9 任意の Java プログラムを使用したデータ変換の定義	148
5.9.1 CustomFunction サンプルプログラムの定義の流れ	149
5.9.2 カスタムファンクションの準備	149

5.9.3	Hello サービスアダプタの変更	161
5.9.4	Hello ビジネスプロセスの変更	169
5.10	アンセットアップとアンインストール	173
5.10.1	MyEclipse のアンセットアップ	173
5.10.2	uCosminexus Service Architect のアンインストール	175

付録 177

付録 A	サンプルプログラムのファイルの構成	178
付録 A.1	HelloServiceAdapter サンプルプログラムの構成	178
付録 A.2	HelloBusinessProcess サンプルプログラムの構成	181
付録 A.3	商品手配サンプルプログラムの構成	183
付録 A.4	CustomFunction サンプルプログラムの構成	186
付録 B	インスタントセットアップ機能実行時の情報の採取	187
付録 C	このマニュアルの参考情報	188
付録 C.1	関連マニュアル	188
付録 C.2	このマニュアルでの表記	190
付録 C.3	英略語	192
付録 C.4	KB (キロバイト) などの単位表記について	193
付録 D	用語解説	194

索引 205

1

サンプルプログラムをご使用になる前に

この章では、このマニュアルの読み方、およびこのマニュアルで体験できる内容について説明します。

1.1 このマニュアルの読み方

1.2 このマニュアルで体験できること

1.1 このマニュアルの読み方

このマニュアルは、実際にマシンを操作しながら、Cosminexus サービスプラットフォームの環境構築からサンプルプログラムを実行するまでの操作を体験するためのものです。

2章では、この製品で提供している三つのサンプルプログラムの内容、およびこのマニュアルで知っておく必要のある用語について説明しています。

3章では、Cosminexus サービスプラットフォームの環境構築の手順、および三つのサンプルプログラムを実行するための準備について説明しています。

4章では、3章で準備したサンプルプログラムを実行するための手順を説明しています。

5章では、3章および4章で使用したサンプルプログラムを、開発環境で作成する手順について説明しています。

それぞれの章は通読型となっていますので、マシンを操作しながら、順番に読み進めてください。

ポイント

インストールディレクトリについて

uCosminexus Service Architect のデフォルトのインストールディレクトリは、「C:\Program Files\Hitachi\Cosminexus」です。別のディレクトリにインストールする場合には、このマニュアルに記載されている「<uCosminexus Service Architect のインストールディレクトリ>」をご利用のディレクトリに読み替えてください。

1.2 このマニュアルで体験できること

このマニュアルでは、MyEclipse を使用して、サンプルプログラムの環境設定および開発から実行までの操作を体験できます。

それぞれの章で体験できることを次に示します。

- 3 章で体験できること

uCosminexus Service Architect のインストール、MyEclipse セットアップ機能を使用した Eclipse のセットアップ、および HCSC 簡易セットアップ機能を使用した実行環境のセットアップを体験できます。

- 4 章で体験できること

3 章で準備した環境を使用して、Web ブラウザ上でサンプルプログラムを実行する操作を体験できます。また、別のサンプルプログラムを実行する場合に、3 章で準備したサンプルプログラムを削除する操作についても体験できます。

- 5 章で体験できること

手順どおりに操作を進めることで、Cosminexus サービスプラットフォームの開発環境を使用して、三つのサンプルプログラムの開発から実行までの一連の流れを体験できます。

サンプルプログラムの開発の中では、サービスアダプタ、ビジネスプロセス、データ変換定義など、Cosminexus サービスプラットフォームを使用する上で基本的な機能の操作を体験できます。また、サンプルプログラムの一つを例に、Cosminexus サービスプラットフォームが提供するデバッグ機能の操作を体験できます。

2

サンプルプログラムの紹介

この章では，サンプルプログラムの概要について説明します。
また，それぞれのサンプルプログラムで学んでいただきたいこと，およびその目的についても説明します。

2.1 サンプルプログラムのシステム構成

2.2 サンプルプログラムを構成している要素

2.3 サンプルプログラムの処理の内容

2.1 サンプルプログラムのシステム構成

Cosminexus サービスプラットフォームでは、次に示す三つのパターンのサンプルプログラムを用意しています。

サービスリクエスタからサービス部品を呼び出すパターン

ビジネスプロセスからサービス部品を呼び出すパターン

プロセス統合パターン

それぞれのサンプルプログラムについて説明します。

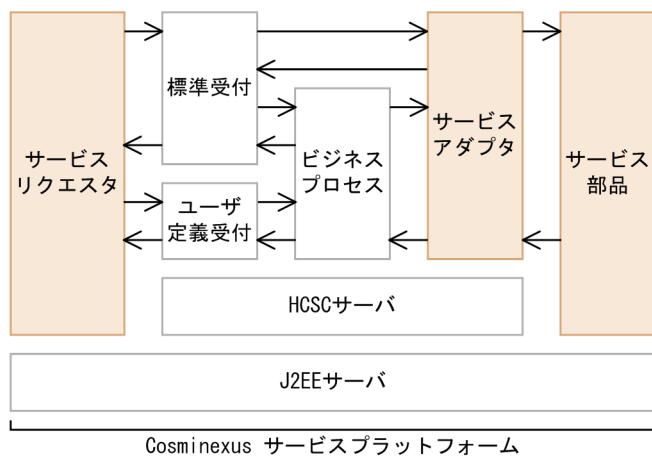
(1) サービスリクエスタからサービス部品を呼び出すパターン

サンプルプログラム名：HelloServiceAdapter

サービスリクエスタからサービスアダプタを介してサービス部品を呼び出すサンプルプログラムです。このサンプルプログラムでは、サービスアダプタの定義を習得することを目的としています。

HelloServiceAdapter のシステム構成を次の図に示します。

図 2-1 HelloServiceAdapter のシステム構成



(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

(2) ビジネスプロセスからサービス部品を呼び出すパターン

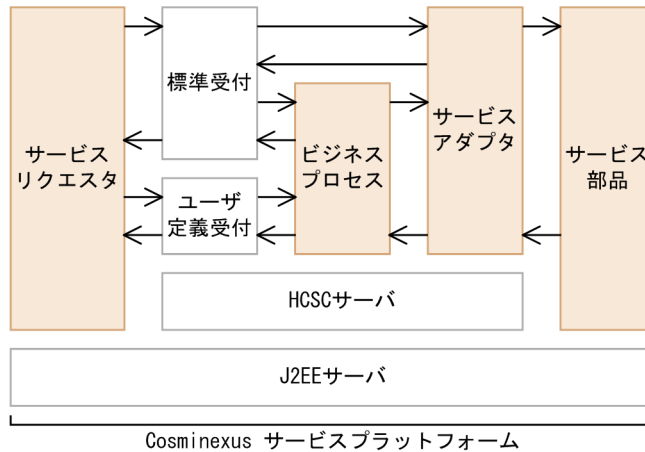
サンプルプログラム名：HelloBusinessProcess

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部品を呼び出すサンプルプロ

グラムです。このサンプルプログラムでは、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。

HelloBusinessProcess のシステム構成を次の図に示します。

図 2-2 HelloBusinessProcess のシステム構成



(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

(3) プロセス統合したパターン

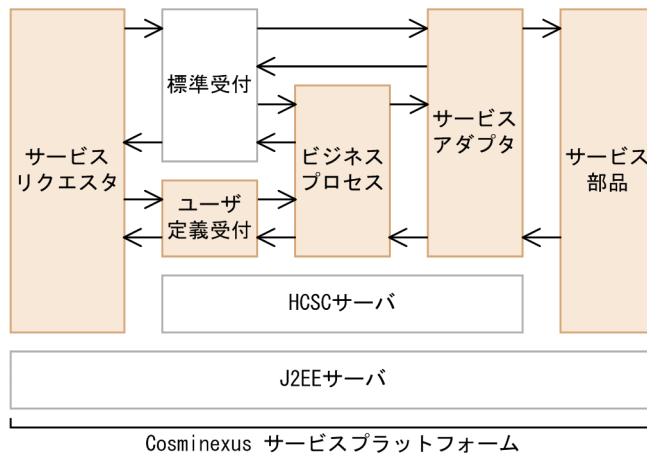
サンプルプログラム名：商品手配サンプルプログラム

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部品を呼び出すサンプルプログラムです。このサンプルプログラムでは、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。

商品手配サンプルプログラムのシステム構成を次の図に示します。

2. サンプルプログラムの紹介

図 2-3 商品手配サンプルプログラムのシステム構成



(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

2.2 サンプルプログラムを構成している要素

「2.1 サンプルプログラムのシステム構成」で説明したサンプルプログラムの構成要素が、それぞれどのような役割を果たしているのかを説明します。

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。

サービスリクエスト

サービスを利用するアプリケーションプログラムのことです。HCSC サーバに対してサービスを呼び出す要求（要求電文）を送ります。サンプルプログラムのサービスリクエストは、通信基盤として SOAP 通信基盤を使用しています。

HCSC サーバ

サービスを実行するために、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、HCSC サーバに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。

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

複数のサービスの処理順序や処理条件などを定義し、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。

！ 注意事項

サンプルプログラムの動作環境は、HCSC 簡易セットアップ機能を利用して構築してください。HCSC 簡易セットアップ機能を利用して環境を構築すると、通信基盤として SOAP 通信基盤が設定されます。

2.3 サンプルプログラムの処理の内容

サンプルプログラムを構成する各要素が、どのような処理を実行しているのかを説明します。

2.3.1 サービスリクエストからサービス呼び出すパターン

HelloServiceAdapter サンプルプログラムの処理の詳細を次の図および表に示します。

図 2-4 HelloServiceAdapter サンプルプログラムの処理詳細

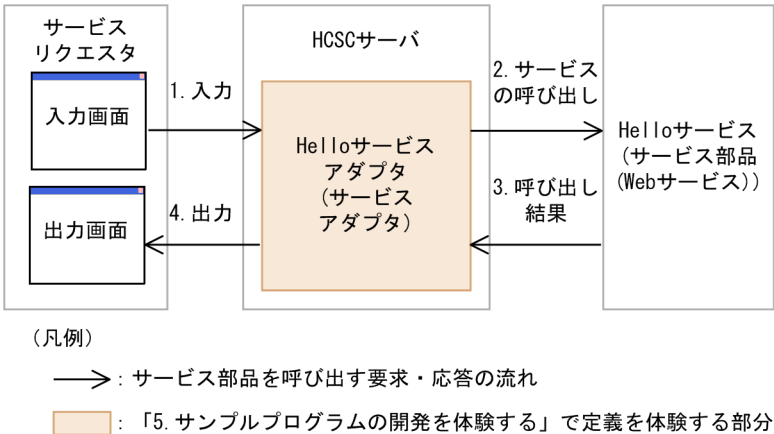


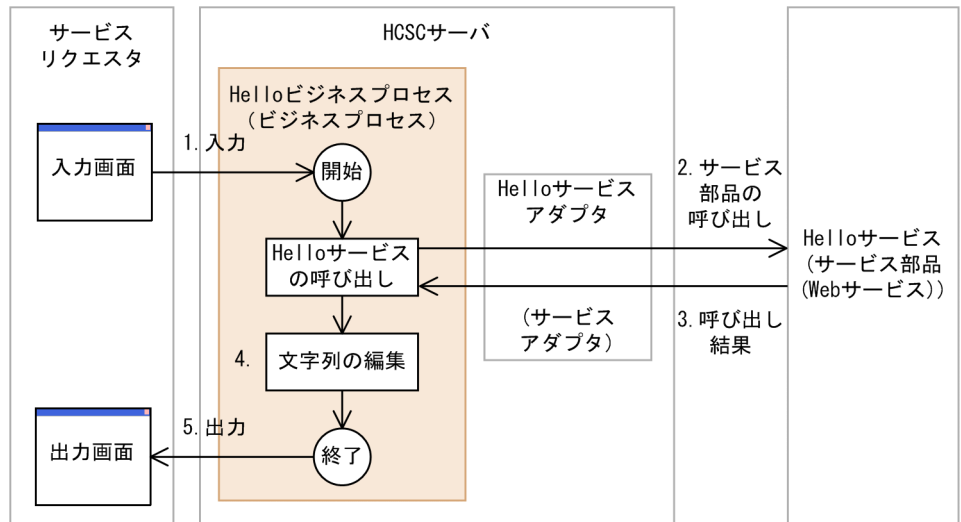
表 2-1 HelloServiceAdapter サンプルプログラムの処理内容

コンポーネント	形態	内容
サービスリクエスト	HTML サーブレット	<ul style="list-style-type: none">入力画面を表示し、サービスアダプタへ Hello サービスを呼び出す要求を送ります。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	<ul style="list-style-type: none">サービスリクエストからの要求を受け取り、Hello サービスを呼び出します。Hello サービスの応答をサービスリクエストへ返します。
Hello サービス	Web サービス	<ul style="list-style-type: none">サービスアダプタからの呼び出しに応答し、入力された文字列を編集して Hello サービスアダプタに返します。

2.3.2 ビジネスプロセスからサービス呼び出すパターン

HelloBusinessProcess サンプルプログラムの処理の詳細を次の図および表に示します。

図 2-5 HelloBusinessProcess サンプルプログラムの処理詳細



(凡例)

→ : サービス部品を呼び出す要求・応答の流れ

[オレンジ色] : 「5. サンプルプログラムの開発を体験する」で定義を体験する部分

表 2-2 HelloBusinessProcess サンプルプログラムの処理内容

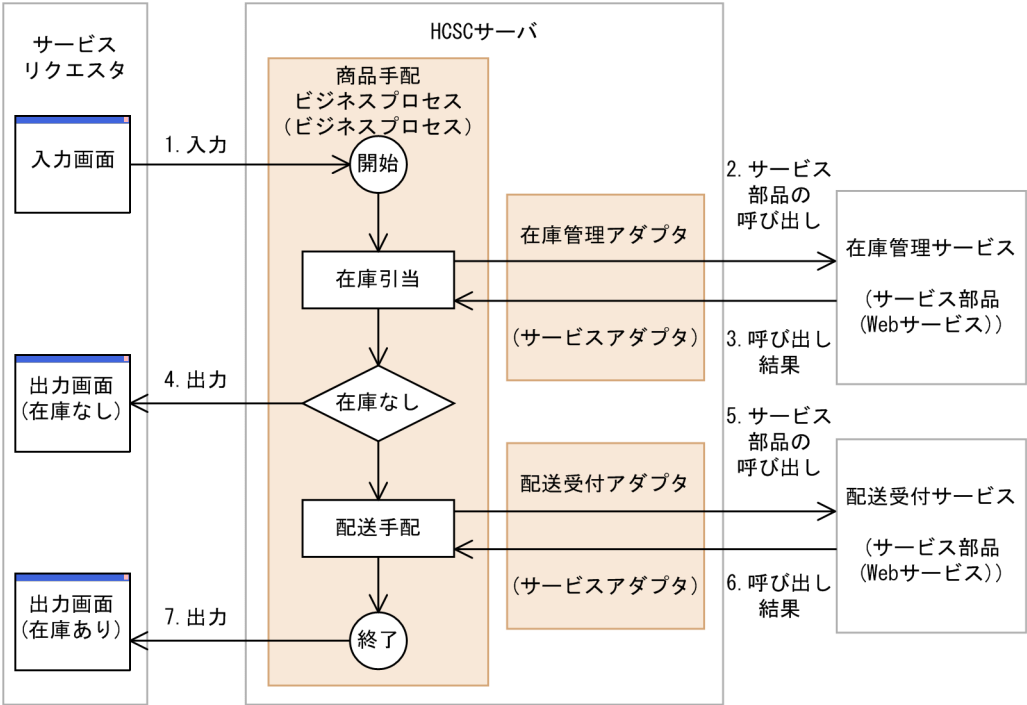
コンポーネント	形態	内容
サービスリクエスタ	HTML サブレット	<ul style="list-style-type: none"> 入力画面を表示し、Hello ビジネスプロセスを呼び出します。 Hello ビジネスプロセスで編集した文字列を戻り値として取得し、出力画面に表示します。
Hello ビジネスプロセス	ビジネスプロセス	<ul style="list-style-type: none"> サービスリクエスタからの要求を受け取り、Hello サービスアダプタを介して Hello サービスを呼び出します。 Hello サービスを呼び出した結果を Hello サービスアダプタを介して戻り値として取得します。 戻り値である文字列を編集し、サービスリクエスタへ返します。
Hello サービスアダプタ	サービスアダプタ	<ul style="list-style-type: none"> Hello ビジネスプロセスからの要求を受け取り、Hello サービスを呼び出します。 Hello サービスの応答を Hello ビジネスプロセスへ返します。
Hello サービス	Web サービス	<ul style="list-style-type: none"> Hello ビジネスプロセスからの呼び出しに応答し、入力された文字列を編集して Hello ビジネスプロセスへ返します。

2.3.3 プロセス統合パターン

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処理の詳細を次の図および表に示します。

図 2-6 商品手配サンプルプログラムの処理詳細



(凡例)

→ : サービス部品を呼び出す要求・応答の流れ

Orange box : 「5. サンプルプログラムの開発を体験する」で定義を体験する部分

表 2-3 商品手配サンプルプログラムの処理内容

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	<ul style="list-style-type: none">入力画面を表示し、商品手配ビジネスプロセスを呼び出します。在庫管理サービスを呼び出した結果を戻り値として取得し、出力画面に表示します。配送受付サービスを呼び出した結果を戻り値として取得し、出力画面に表示します。

コンポーネント	形態	内容
商品手配ビジネスプロセス	ビジネスプロセス	<ul style="list-style-type: none"> サービスリクエストからの要求を受け取り、在庫管理アダプタを介して在庫管理サービス呼び出します。 サービスリクエストからの要求を受け取り、配送受付アダプタを介して配送受付サービス呼び出します。 在庫管理サービス呼び出した結果をサービスリクエストへ返します。 配送受付サービス呼び出した結果をサービスリクエストへ返します。
在庫管理アダプタ	サービスアダプタ	<ul style="list-style-type: none"> 商品手配ビジネスプロセスからの要求を受け取り、在庫管理サービス呼び出します。 在庫管理サービスの応答を商品手配ビジネスプロセスへ返します。
配送受付アダプタ	サービスアダプタ	<ul style="list-style-type: none"> 商品手配ビジネスプロセスからの要求を受け取り、配送受付サービス呼び出します。 配送受付サービスの応答を商品手配ビジネスプロセスへ返します。
在庫管理サービス	Web サービス	<ul style="list-style-type: none"> 商品手配ビジネスプロセスからの呼び出しに回答して商品の在庫を引き当て、商品手配ビジネスプロセスへ引当番号を返します。
配送受付サービス	Web サービス	<ul style="list-style-type: none"> 商品手配ビジネスプロセスからの呼び出しに回答し、配送手配した結果の配送番号を返します。

3

サンプルプログラムの環境を準備する

この章では、サンプルプログラムを動かすために必要な環境構築の方法について説明します。

3.1 インストールとセットアップの流れ

3.2 インストールと前提環境の準備

3.3 MyEclipse セットアップ機能を使用したセットアップ

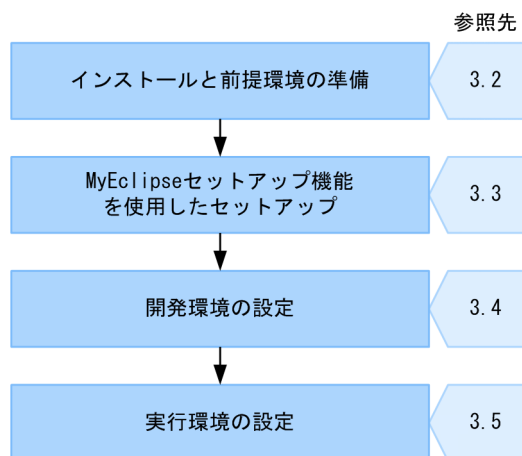
3.4 開発環境の設定

3.5 実行環境の設定

3.1 インストールとセットアップの流れ

サンプルプログラムを動かすための環境構築手順を次の図に示します。

図 3-1 サンプルプログラムを動かすための環境構築手順



1. uCosminexus Service Architect のインストールと前提環境の準備

日立統合インストーラを使用して、uCosminexus Service Architect をインストールします。uCosminexus Service Architect をインストールすると MyEclipse もインストールされます。また、セットアップに必要な Eclipse のアーカイブファイルをダウンロードします。詳細については、「3.2 インストールと前提環境の準備」を参照してください。

2. MyEclipse セットアップ機能を使用したセットアップ

MyEclipse セットアップ機能を使用して、開発環境（MyEclipse）をセットアップします。詳細については、「3.3 MyEclipse セットアップ機能を使用したセットアップ」を参照してください。

3. 開発環境の設定

MyEclipse を使用して J2EE アプリケーションを開発するための設定をします。詳細については、「3.4 開発環境の設定」を参照してください。

4. 実行環境の設定

uCosminexus Service Architect の HCSC 簡易セットアップ機能を利用し、実行環境を構築します。詳細については、「3.5 実行環境の設定」を参照してください。

以降の節で、これらの手順について説明します。

3.2 インストールと前提環境の準備

ここでは、uCosminexus Service Architect のインストールと、セットアップに必要な Eclipse のアーカイブファイルのダウンロードについて説明します。

3.2.1 uCosminexus Service Architect のインストール

uCosminexus Service Architect をインストールします。uCosminexus Service Architect は、日立統合インストーラからインストールします。インストールする前にすべての Windows アプリケーションを終了させてください。また、インストールは、Administrator 権限が設定されたユーザが実施してください。

1. uCosminexus Service Architect の媒体を CD-ROM ドライブにセットします。
日立統合インストーラが起動し、[日立統合インストーラ] ダイアログが表示されます。
2. インストールするソフトウェアとして uCosminexus Service Architect を選択し、[インストール実行] ボタンをクリックします。
インストールを実行してよいかどうかを確認するメッセージが表示されます。
3. [OK] ボタンをクリックします。
uCosminexus Service Architect のセットアッププログラムが起動されます。このあと、画面の指示に従って、処理を進めてください。
 - インストール先フォルダを指定する場合、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。
4. uCosminexus Service Architect のインストール終了後、再起動します。

3.2.2 Eclipse のアーカイブファイルのダウンロード

MyEclipse セットアップ機能を実行する前に Eclipse のアーカイブファイルをダウンロードする必要があります。ダウンロードするアーカイブファイルの種類とダウンロード先について説明します。

参考

ランゲージパックの使用について

Eclipse の環境を日本語化したい場合は、対応するランゲージパックをダウンロードして使用してください。ランゲージパックの適用は、MyEclipse のセットアップが完了してから実施します。なおこのマニュアルでは Eclipse Babel Project が提供する BABEL 日本語ランゲージパック 2010/12/10 版を適用した Eclipse メニュー表記を使用しています。

3. サンプルプログラムの環境を準備する

(1) ダウンロードするアーカイブファイルの種類

Eclipse のアーカイブファイルは、Eclipse.org のダウンロードサイトからダウンロードしてください。また、日本語 MyEclipse のポータルサイトからもダウンロードできます。

次に示すアーカイブファイルをダウンロードしてください。

- Eclipse Classic

! 注意事項

- ダウンロードする Eclipse のバージョン
Eclipse のバージョン 3.6.1 に対応するアーカイブファイルを入手してください。
- ダウンロードした Eclipse のアーカイブファイルの扱い
インスタントセットアップ機能で指定したアーカイブファイルは、削除または名称変更しないでください。

(2) アーカイブファイルのダウンロード先

入手したアーカイブファイルをアーカイブファイル格納用ディレクトリ (<uCosminexus Service Architect のインストールディレクトリ >¥MyEclipse¥archives) に格納しておく、インスタントセットアップ機能実行時に自動的にアーカイブファイルを取り込みます。

アーカイブファイル格納用ディレクトリに同じ種類のアーカイブファイルが存在する場合は、新しいバージョンのアーカイブファイルを優先して取り込みます。

なお、ダウンロードしたアーカイブファイルは、任意のフォルダに格納することもできます。ただし、ダウンロード先のフォルダには、アーカイブファイルと同名のフォルダが存在する場合は、そのフォルダを削除してください。アーカイブファイルと同名のフォルダがあると、そのフォルダがアーカイブファイルとして自動設定されてしまうため、存在しないアーカイブを指定したとしてエラーが発生します。

3.3 MyEclipse セットアップ機能を使用した セットアップ

MyEclipse セットアップ機能を使用して、MyEclipse をセットアップします。

MyEclipse セットアップ機能を使用すると、ダウンロードしておいた Eclipse のアーカイブファイルから Eclipse をインストールし、MyEclipse を組み込みます。なお、MyEclipse セットアップ機能以外の方法でインストールした Eclipse には、MyEclipse セットアップ機能を使用して MyEclipse を組み込みません。

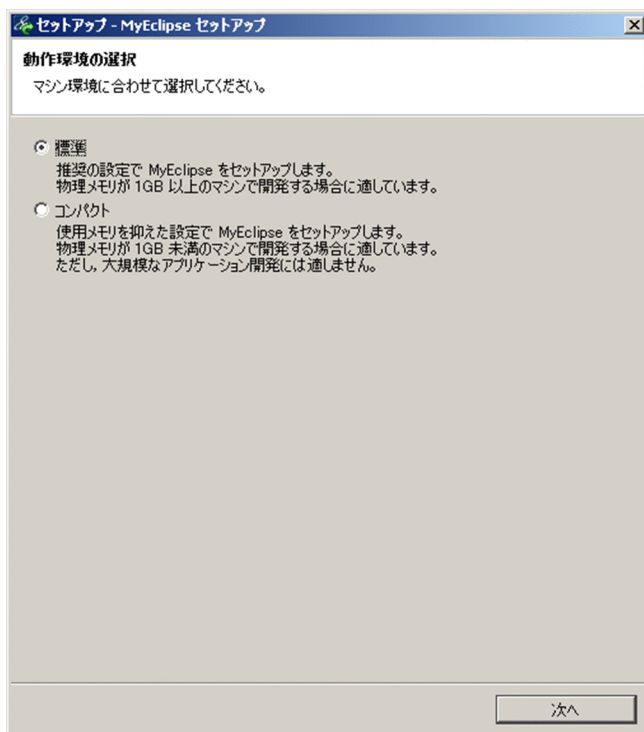
MyEclipse セットアップ機能を使用した場合、デフォルトの Eclipse のインストールディレクトリは次のとおりです。

```
<uCosminexus Service Architect のインストールディレクトリ  
>%MyEclipse%IDE
```

MyEclipse セットアップ機能を使用して、MyEclipse をセットアップする手順を説明します。

3. サンプルプログラムの環境を準備する

1. スタートメニューから [プログラム] - [Cosminexus] - [MyEclipse] - [セットアップ] を選択します。
MyEclipse セットアップ機能が起動して , [セットアップ - MyEclipse セットアップ] ダイアログの [動作環境の選択] ページが表示されます。



2. 動作環境として [標準] を選択します。
推奨の設定で MyEclipse をセットアップします。MyEclipse を動作させるマシンの物理メモリが 1GB 以上必要です。
3. [次へ] ボタンをクリックします。
Eclipse のアーカイブファイルをダウンロードしたかどうかを確認するダイアログが表示されます。Eclipse のアーカイブファイルをダウンロードしていない場合は、表示される URL からアーカイブファイルをダウンロードしてください。
[次回からこのダイアログを表示しない] にチェックを入れた場合、次回起動時にこのダイアログは表示されません。

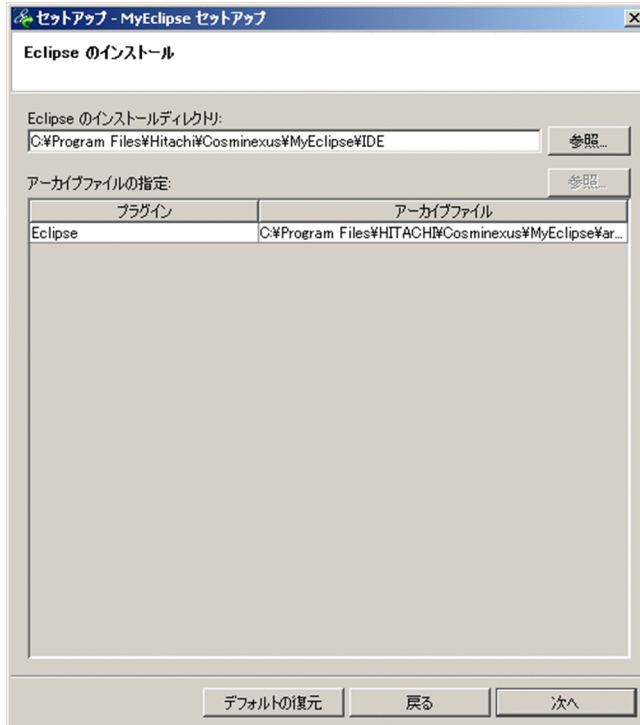
4. 問題がなければ、[了解] ボタンをクリックします。

[Eclipse のインストール] ページが表示されます。

Eclipse のアーカイブファイルをアーカイブファイル格納用ディレクトリ

(<uCosminexus Service Architect のインストールディレクトリ

>¥MyEclipse¥archives) に格納した場合は、次に示すような画面が表示されます。



なお、Eclipse のアーカイブファイルをアーカイブファイル格納用ディレクトリに格納していない場合は、[アーカイブファイルの指定] テーブルで Eclipse のアーカイブファイルを指定する必要があります。

5. Eclipse のインストールディレクトリおよびアーカイブファイルを指定します。

項目名	指定値
Eclipse のインストールディレクトリ	<p>Eclipse のインストールディレクトリを 50 文字以内で指定します。なお、文字列は全角、半角のどちらでもかまいません。 デフォルトの設定から変更したい場合は、[参照] ボタンから指定してください。</p> <p>注意</p> <p>次に示す条件のどちらかに当てはまるディレクトリは指定できません。</p> <ul style="list-style-type: none"> ディレクトリがない。 ディレクトリのパスが UNC で表記されている。

3. サンプルプログラムの環境を準備する

項目名	指定値
Eclipse	[Eclipse] の行を選択して [参照] ボタンから Eclipse Classic のアーカイブファイルを指定します。

注

[アーカイブファイルの指定] テーブルで指定したアーカイブファイルは、削除またはリネームしないでください。

6. [次へ] ボタンをクリックします。
[セットアップの確認] ページが表示されます。



[セットアップの内容] エリアにセットアップされる内容が表示されるので、設定値を確認してください。

7. [実行] ボタンをクリックします。

[進行状況] ページが表示されます。

セットアップが終了すると、[セットアップの完了] ページが表示されます。



8. [終了] ボタンをクリックします。

[セットアップ - MyEclipse セットアップ] ダイアログが閉じます。MyEclipse がセットアップされ、デスクトップ上にMyEclipseのショートカットが生成されます。MyEclipse セットアップ機能実行時のログを確認したい場合は、「付録 B インスタントセットアップ機能実行時の情報の採取」を参照してください。

3. サンプルプログラムの環境を準備する

参考

処理が中断された場合の対処

[中止] ボタンをクリックするなどによって、処理を中断した場合は、アンセットアップを実行してから再度セットアップを実行してください。アンセットアップの手順については、「5.10.1 MyEclipse のアンセットアップ」を参照してください。

エラーによって中断された場合は、コンソールに出力されたエラーメッセージを確認してエラーを取り除いてから、アンセットアップしてください。そのあと、再度セットアップを実行してください。

出力されたエラーメッセージについては、マニュアル「Cosminexus アプリケーションサーバメッセージ3」の「3. KEDT (インスタントセットアップ機能で出力されるメッセージ)」を参照して対処してください。

MyEclipse セットアップ機能の起動オプションの使用

注意事項ダイアログで設定した [次回からこのダイアログを表示しない] の設定を取り消して起動したい場合は、起動オプション `-resetdialog` を指定して、MyEclipse セットアップ機能を起動します。

次の起動オプションを指定して、実行してください。

```
<uCosminexus Service Architectのインストールディレクトリ  
>%MyEclipse%setup%bin%MyEclipseSetup -resetdialog
```

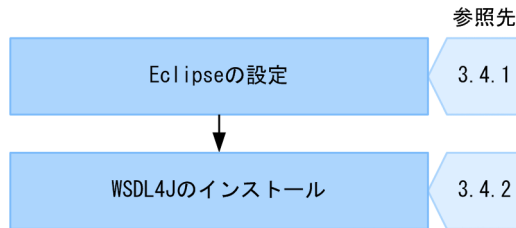
セットアップログの確認方法

MyEclipse セットアップ機能を実行したときのログは、MyEclipse のセットアップログに出力されます。セットアップログの確認方法については「付録 B インスタントセットアップ機能実行時の情報の採取」を参照してください。

3.4 開発環境の設定

ここでは、J2EE アプリケーションを開発するために必要な設定について説明します。設定の流れを次の図に示します。

図 3-2 開発環境の設定の流れ



それぞれの作業の概要を説明します。

1. Eclipse の設定

Eclipse で使用する JDK、およびローカル変数情報の出力について設定します。詳細については、「3.4.1 Eclipse の設定」を参照してください。

2. WSDL4J のインストール

WSDL4J をインストールします。WSDL はドキュメント解析および生成に必要です。詳細については、「3.4.2 WSDL4J のインストール」を参照してください。

以降の項では、この流れに沿って開発環境の設定の手順を説明します。

3.4.1 Eclipse の設定

ここでは、Eclipse 上で JDK の設定を確認する手順を示します。また、J2EE アプリケーションが持つローカル変数情報を出力する場合の設定手順を示します。

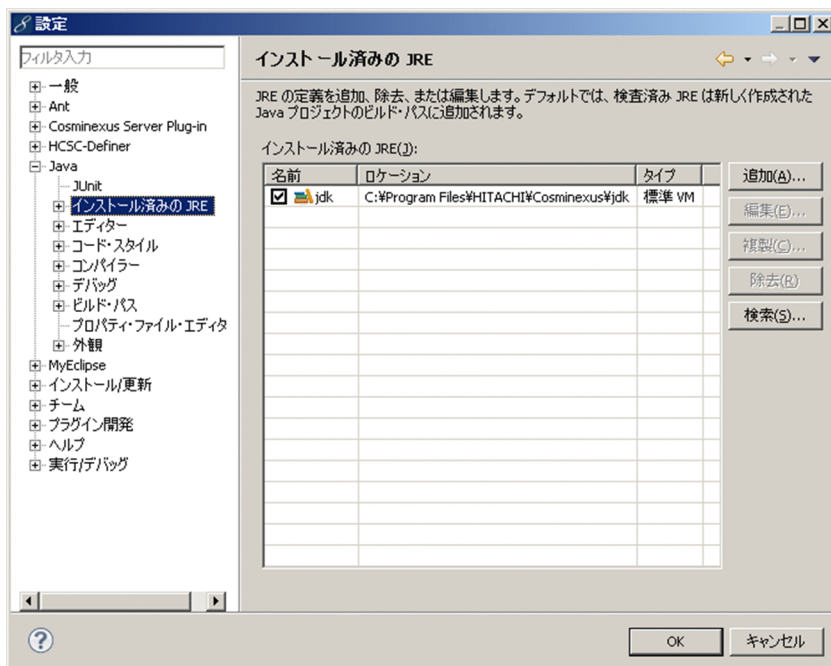
(1) JDK の確認

Eclipse 上で開発するとき使用する JDK が uCosminexus Service Architect で提供されている JDK かどうかを確認します。確認手順を次に示します。

1. MyEclipse 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。

3. サンプルプログラムの環境を準備する

2. 左ペインで [Java] - [インストール済みの JRE] を選択します。
右ペインに [インストール済みの JRE] ページが表示されます。



3. [インストール済みの JRE] が uCosminexus Service Architect で提供されている JDK かどうかを確認します。
[ロケーション] に次のパスが表示されているかどうかを確認します。

<uCosminexus Service Architectのインストールディレクトリ>\jdk

パスが表示されていない場合

[追加] ボタンをクリックして、上記のパスを指定します。パスの指定後、[名前] にチェックを入れます。

パスが表示されている場合

[名前] にチェックが入っているかどうかを確認します。チェックが入っていない場合はチェックを入れてください。

特に、複数の JDK がインストールされている場合、<uCosminexus Service Architect のインストールディレクトリ>\jdk にチェックが入っていないことがあります。チェックが入っていない場合は、チェックを入れてください。

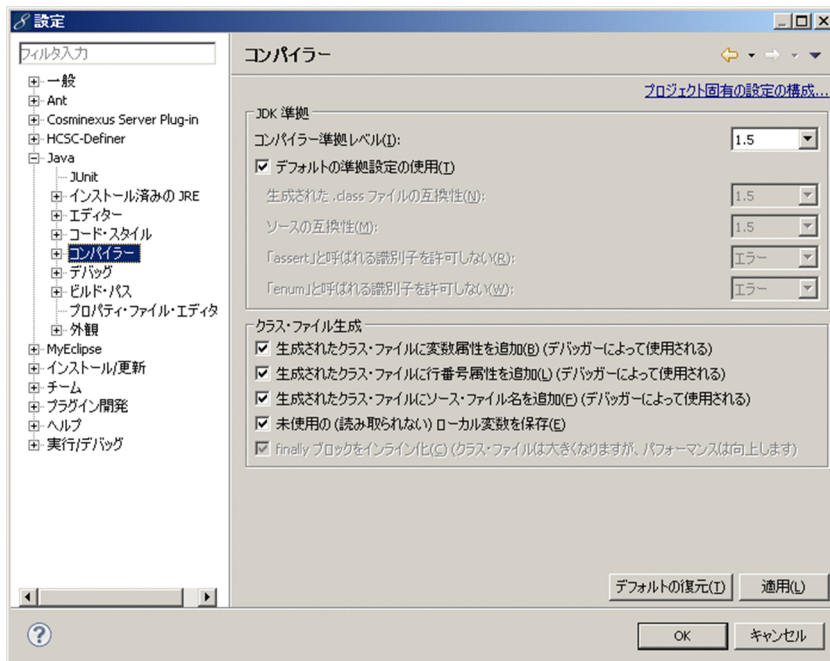
4. [OK] ボタンをクリックします。
設定が保存されます。

(2) ローカル変数情報の出力の設定

Eclipse のコンパイラの設定によって、例外発生時に J2EE アプリケーションが持つローカル変数の情報をスタックトレースに出力できます。

ローカル変数情報をスタックトレースに出力する場合のコンパイラの設定手順を次に示します。

1. MyEclipse 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. 左ペインで [Java] - [コンパイラー] を選択します。
[コンパイラー] ページが表示されます。



3. 次の項目を指定します。

項目名		指定値
JDK 準拠	コンパイラー準拠レベル	「1.5」を選択します。
クラス・ファイル生成		[生成されたクラス・ファイルに変数属性を追加 (デバッガーによって使用される)] にチェックを入れます。 なお、それ以外の項目については、出力したい内容に合わせてチェックを入れてください。

3. サンプルプログラムの環境を準備する

必要に応じて、次の項目を指定してください。

項目名		指定値
JDK 準拠	デフォルトの準拠設定の使用	コンパイラに使用する設定を指定します。 <ul style="list-style-type: none">• チェックを入れる [コンパイラ準拠レベル] で指定したレベルに沿った設定が適用されます。• チェックを入れない 次の項目を手動で指定します。 [生成された .class ファイルの互換性] [ソースの互換性] [「assert」と呼ばれる識別子を許可しない] [「enum」と呼ばれる識別子を許可しない]

4. [適用] ボタンまたは [OK] ボタンをクリックします。
設定が保存されます。

3.4.2 WSDL4J のインストール

WSDL のドキュメント解析および生成に必要な WSDL4J をインストールします。
WSDL4J は、uCosminexus Service Architect とは別の CD-ROM で提供されています。

1. WSDL4J をインストールするために一時的に使用する作業用ディレクトリを作成します。
例として「C:\¥WSDL4J_work」を作成します。

2. WSDL4J の CD-ROM に格納されている wsd4j-bin-1.5.1.zip を操作 1. で作成した WSDL4J 作業用ディレクトリにコピーします。
wsd4j-bin-1.5.1.zip は次のフォルダに格納されています。

<CD-ROMドライブ>:\¥WSDL4J

3. WSDL4J 作業用ディレクトリにコピーした wsd4j-bin-1.5.1.zip を右クリックし、[すべて展開] を選択します。
圧縮フォルダの展開ウィザード画面が表示されます。
4. [次へ] ボタンをクリックします。
展開ウィザード画面が表示されます。
5. [次へ] ボタンをクリックします。
6. [展開されているファイルを表示する] のチェックを外します。
7. [完了] ボタンをクリックします。
8. 展開した JAR ライブラリファイル「wsd4j.jar」、ライセンスファイル「license.html」、および「Readme」をコピーします。
上記三つのファイルが格納されているフォルダ

3. サンプルプログラムの環境を準備する

```
<WSDL4J 作業用ディレクトリ >%wsdl4j-bin-1.5.1%\wsdl4j-1_5_1\lib
```

コピー先のフォルダ

```
<uCosminexus Service Architect のインストールディレクトリ
```

```
>%c4web%\lib
```

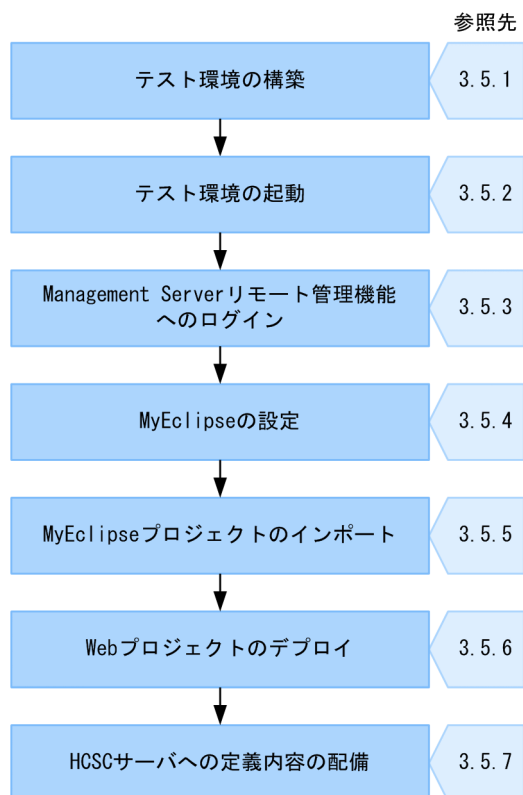
9. 操作 1. で作成した WSDL4J 作業用ディレクトリを削除します。

3.5 実行環境の設定

ここでは、uCosminexus Service Architect の HCSC 簡易セットアップ機能を利用して実行環境を構築する手順について説明します。

設定の流れを次の図に示します。

図 3-3 実行環境の設定の流れ



それぞれの作業の概要を説明します。

1. テスト環境の構築

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。詳細については、「3.5.1 テスト環境の構築」を参照してください。

2. テスト環境の起動

構築したテスト環境を起動します。詳細については、「3.5.2 テスト環境の起動」を参照してください。

3. Management Server リモート管理機能へのログイン

4. の作業を実施するために、MyEclipse から Management Server リモート管理機能へログインします。詳細については、「3.5.3 Management Server リモート管理機能

へのログイン」を参照してください。

4. MyEclipse の設定

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。詳細については、「3.5.4 MyEclipse の設定」を参照してください。

5. MyEclipse プロジェクトのインポート

サンプルプログラムごとに MyEclipse プロジェクトをインポートします。詳細については、「3.5.5 MyEclipse プロジェクトのインポート」を参照してください。

6. Web プロジェクトのデプロイ

サンプルプログラムごとに Web プロジェクトをデプロイします。詳細については、「3.5.6 Web プロジェクトのデプロイ」を参照してください。

7. HCSC サーバへの定義内容の配備

サンプルプログラムごとに定義内容を 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。詳細については、「3.5.7 HCSC サーバへの定義内容の配備」を参照してください。

以降の項では、この流れに沿って実行環境の設定の手順を説明します。

3.5.1 テスト環境の構築

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。テスト環境の構築方法を次に示します。

1. Eclipse を起動している場合は、Eclipse を終了します。
2. 次に示す HCSC 簡易セットアップ機能の起動コマンドが格納されているディレクトリのパスを環境変数 PATH に追加します。

```
<uCosminexus Service Architectのインストールディレクトリ>%CSC%\bin
```

3. コマンドプロンプトで次のコマンドを実行します。

```
csceasysetup
```

3. サンプルプログラムの環境を準備する

HCSC 簡易セットアップの画面の [メイン] ページが表示されます。
サンプルプログラムでは、[メイン] ページに表示された設定項目はデフォルトのまま使します。

HCSC簡易セットアップ

操作 その他

メイン サーバ名称

☒ 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)
%COSMINEXUS_HOME%\CSC\IDBarea 選択

DB接続ポート番号
22200 <5001~65535>

Management Server

HCSCサーバ運用ポート番号
28099 <1~65535>

論理サーバ運用ポート番号
28080 <1~65535>

終了要求受信ポート番号(内部管理用)
28005 <1~65535>

内部通信ポート番号(内部管理用)
28009 <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)
20295 <1~65535>

HCSCサーバ

☒ SOAP 1.1モード ☐ SOAP 1.1/1.2併用モード

Webサービス/MDB(WS-R)受付ポート番号
80 <1~65535>

SessionBean受付ポート番号
900 <1~65535>

MDB(DBキュー)受付ポート番号
20351 <1024~65535>

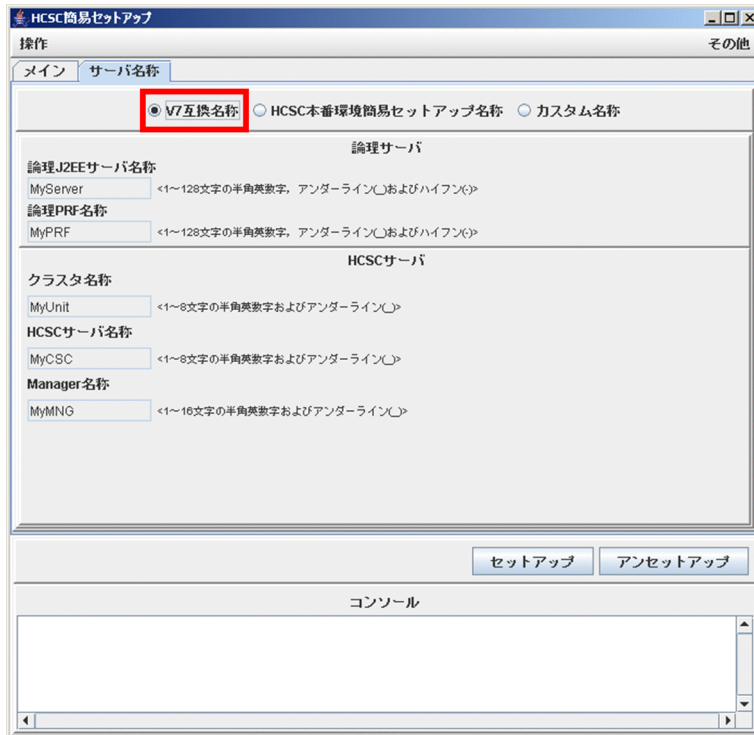
稼働確認ポート番号
23152 <1~65535>

簡易Webサーバポート番号(内部管理用)
8080 <1~65535>

セットアップ アンセットアップ

コンソール

4. [サーバ名称] タブをクリックします。
[サーバ名称] ページが表示されます。
5. [V7 互換名称] をクリックします。
論理サーバおよび HCSC サーバの名称が次のように変更されます。
 - ・ [V7 互換名称] ラジオボタン
 - ・ 論理 J2EE サーバ名称 = MyServer
 - ・ 論理 PRF 名称 = MyPRF
 - ・ クラスタ名称 = MyUnit
 - ・ HCSC サーバ名称 = MyCSC
 - ・ Manager 名称 = MyMNG



6. [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、テスト環境のセットアップは正常に終了です。

注意

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、もう一度セットアップしてください。

- ・[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、アンセットアップします。そのあとで、もう一度セットアップしてください。

7. HCSC 簡易セットアップ画面のメニューから [操作] - [終了] を選択し、HCSC 簡易セットアップ画面を閉じます。

3.5.2 テスト環境の起動

構築したテスト環境を起動する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入力し、テスト環境の組み込みデータベースを起動します。

```
<uCosminexus Service Architectのインストールディレクトリ  
>%CSC%\DB\bats%\cddbstart.bat
```

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2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。

2. コマンドプロンプトで次のコマンドを入力し、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。

```
cscsvstart -system
```

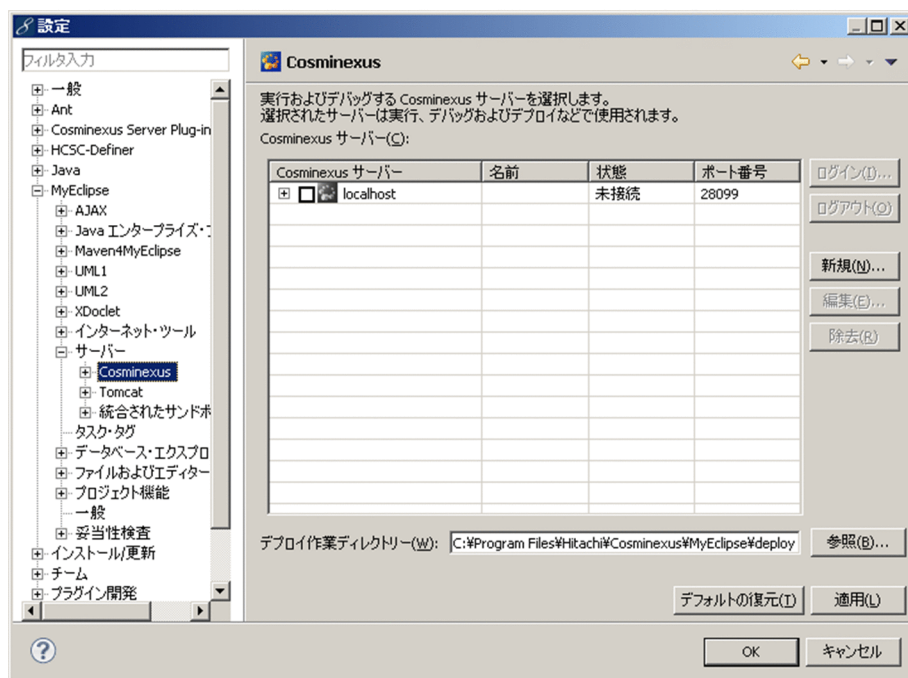
uCosminexus Service Architect の使用を終了する場合は、起動中のテスト環境を停止したあとに Eclipse を終了します。テスト環境を停止する方法については、「4.6 テスト環境の停止」を参照してください。

3.5.3 Management Server リモート管理機能へのログイン

Management Server リモート管理機能へのログイン手順を次に示します。「3.5.4 MyEclipse の設定」以降の手順を実施するためには、Management Server リモート管理機能にログインする必要があります。

1. MyEclipse 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. 左ペインのツリービューで [MyEclipse] - [サーバー] - [Cosminexus] を選択します。
[Cosminexus] ページが表示されます。



3. [Cosminexus] ページの [Cosminexus サーバー] ツリービューで [localhost] の [+] 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、[OK] ボタンをクリックします。
Management Server リモート管理機能に接続します。[Cosminexus サーバー] ツリービューに localhost の運用管理ドメインが表示されます。
5. [OK] ボタンをクリックします。

3.5.4 MyEclipse の設定

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。

(1) ライブラリーパスの設定

MyEclipse では、プロジェクト作成時にビルドパスが設定されます。J2EE 1.4 ライブラリには次に示す Cosminexus のライブラリーパスが設定されます。

3. サンプルプログラムの環境を準備する

```
C:\Program Files\Hitachi\Cosminexus\CC\client\lib\j2ee-javax.jar
C:\Program Files\Hitachi\Cosminexus\jaxp\lib\csmjaxp.jar
C:\Program Files\Hitachi\Cosminexus\jaxp\lib\csmjaxb.jar
C:\Program Files\Hitachi\Cosminexus\jaxp\lib\csmstax.jar
C:\Program Files\Hitachi\Cosminexus\jaxws\lib\cjjaxws.jar
```

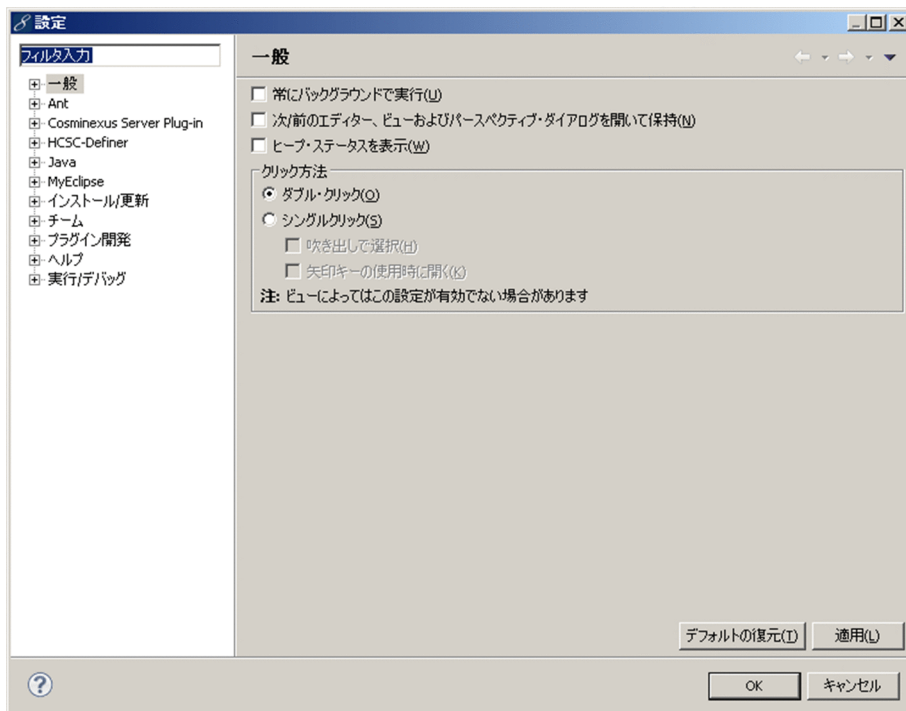
Cosminexus のライブラリーパスの「C:\Program Files\Hitachi\Cosminexus」は、uCosminexus Service Architect のデフォルトのインストールディレクトリです。uCosminexus Service Architect のインストール時にインストールディレクトリを変更している場合は、Cosminexus のライブラリーパスを変更する必要があります。

また、サンプルプログラムのサービスリクエストでは、通信基盤として SOAP 通信基盤を使用しています。プロジェクト作成時には、JAX-WS エンジンのライブラリーパスが設定されているため、JAX-WS エンジンのライブラリーパスを削除したあと SOAP 通信基盤のライブラリーパスを追加する必要があります。

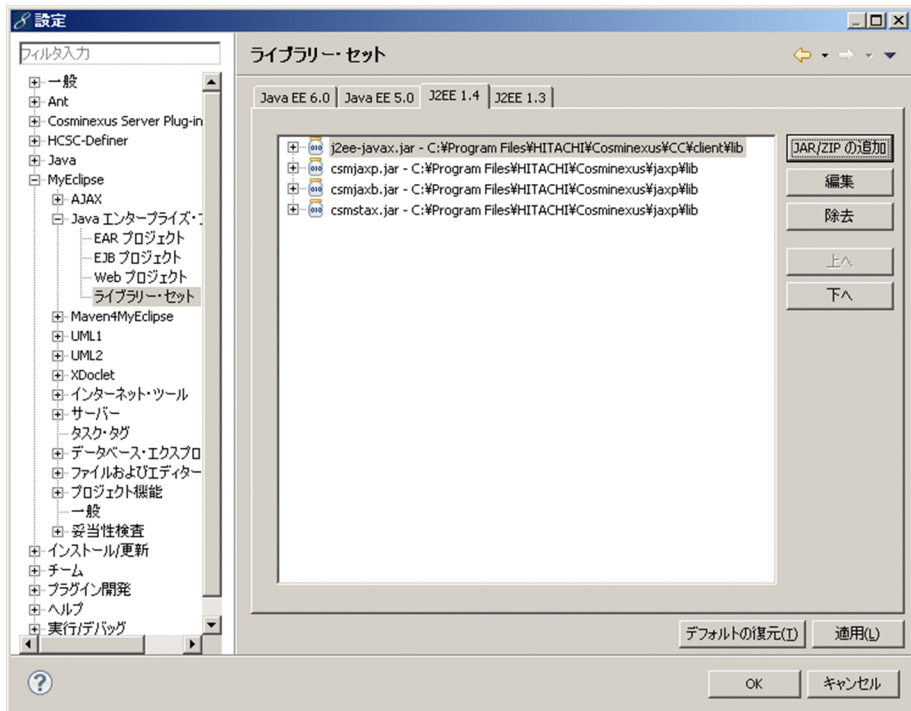
JAX-WS エンジンのライブラリーパスを削除する手順、および SOAP 通信基盤のライブラリーパスを追加する手順を次に示します。

(a) JAX-WS エンジンのライブラリーパスの削除

1. MyEclipse 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. 左ペインのツリービューから [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] を選択します。
右ペインに [ライブラリー・セット] ページが表示されます。
3. [J2EE 1.4] タブを選択します。
現在設定されているライブラリーパスが表示されます。
4. 「cjaxws.jar」を選択し、[除去] ボタンをクリックします。
JAX-WS エンジンのライブラリーパス「cjaxws.jar」が削除されます。
5. 表示されているライブラリーパスと uCosminexus Service Architect のインストールディレクトリが一致しているかどうか確認します。



ライブラリーパスが一致していない場合

表示されているライブラリーパスと uCosminexus Service Architect のインストールディレクトリが一致していない場合は、表示されているライブラリーパスをすべて除去したあと、次に示す Cosminexus のライブラリーパスを再設定してください。

- C:\Program Files\Hitachi\Cosminexus\CC\client\lib\j2ee-javax.jar
- C:\Program Files\Hitachi\Cosminexus\jaxp\lib\csmjaxp.jar
- C:\Program Files\Hitachi\Cosminexus\jaxp\lib\csmjaxb.jar
- C:\Program Files\Hitachi\Cosminexus\jaxp\lib\csmstax.jar

ライブラリーパスを除去する手順は操作 4. と同じです。ライブラリーパスを再設定

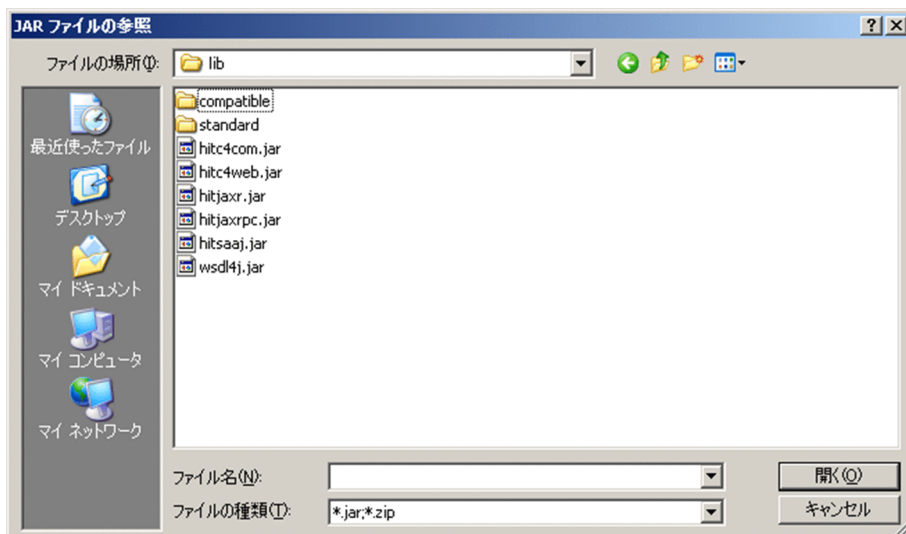
3. サンプルプログラムの環境を準備する

する手順は、「(b) SOAP 通信基盤のライブラリーパスの追加」の操作 1. ~ 2. と同じです。

(b) SOAP 通信基盤のライブラリーパスの追加

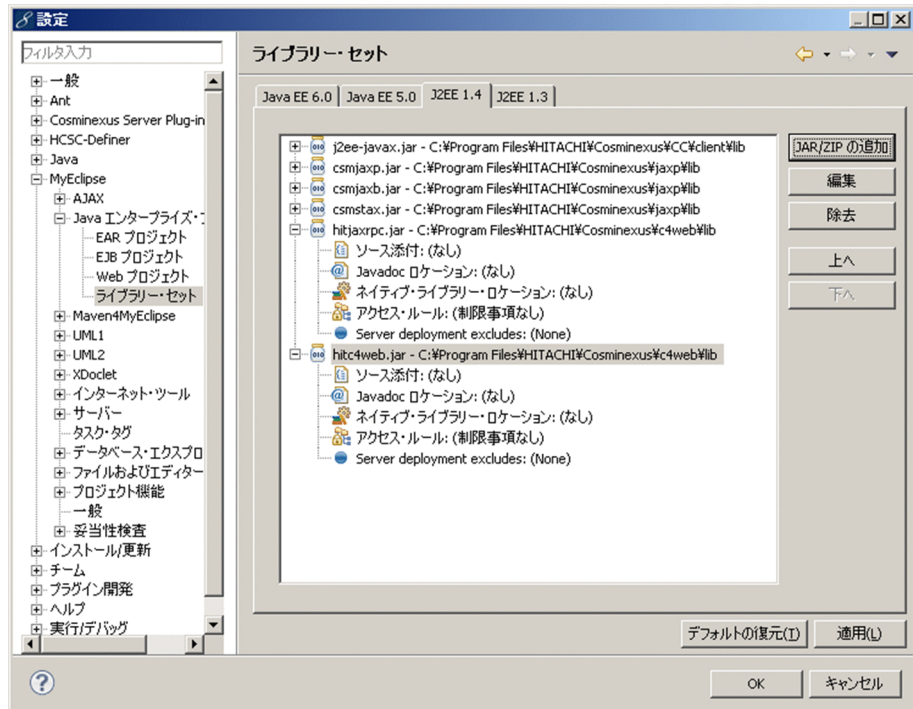
1. [ライブラリー・セット] ページの [J2EE 1.4] タブで [JAR/ZIP の追加] ボタンをクリックします。

[JAR ファイルの参照] ダイアログが表示されます。



2. <uCosminexus Service Architect のインストールディレクトリ
>%c4web%lib%hitjaxrpc.jar を選択し、[開く] ボタンをクリックします。
「hitjaxrpc.jar」が追加されます。

3. 操作 1. ~ 2. と同様の操作で、「hitc4web.jar」を追加します。



4. [OK] ボタンをクリックします。

5. MyEclipse を再起動します。

(2) J2EE サーバの選択

MyEclipse から操作する J2EE サーバを選択します。ここで選択した J2EE サーバに対して、MyEclipse から起動および停止をしたり、J2EE アプリケーションをデプロイしたりします。

設定手順を次に示します。

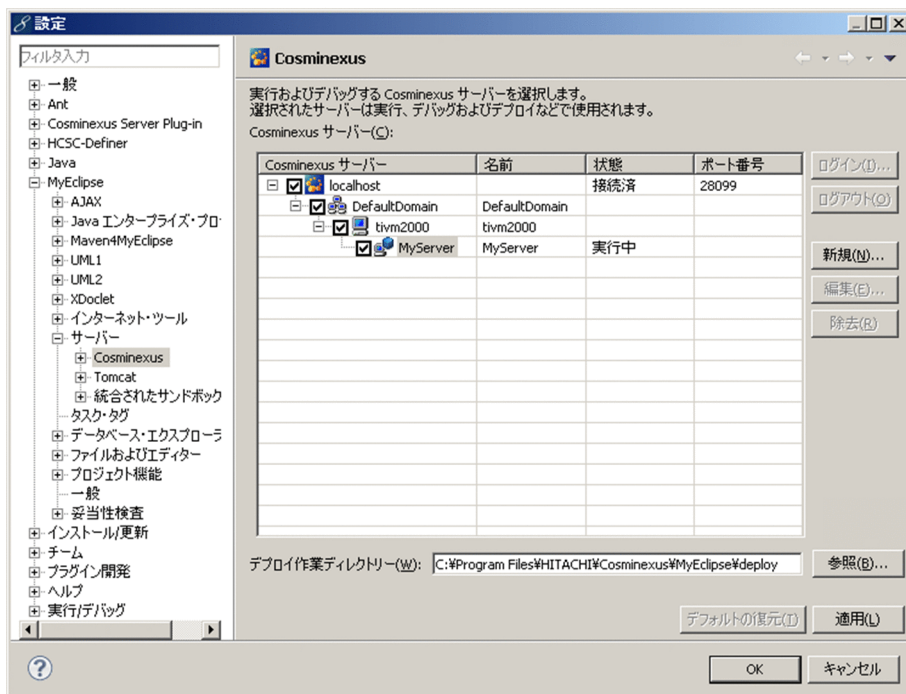
1. MyEclipse のメニューから、[ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューで [MyEclipse] - [サーバー] - [Cosminexus] を選択します。
[Cosminexus] ページが表示されます。
3. [Cosminexus] ページの [Cosminexus サーバー] ツリービューで [localhost] の [+] をクリックします。
ログイン画面が表示されます。

3. サンプルプログラムの環境を準備する

4. [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。

Management Server リモート管理機能に接続します。[Cosminexus サーバー] ツリービューに localhost の運用管理ドメインが表示されます。

5. [Cosminexus サーバー] ツリービューで , 使用する J2EE サーバ (MyServer) のチェックボックスにチェックを入れます。



6. [適用] ボタンまたは [OK] ボタンをクリックします。
設定が保存されます。

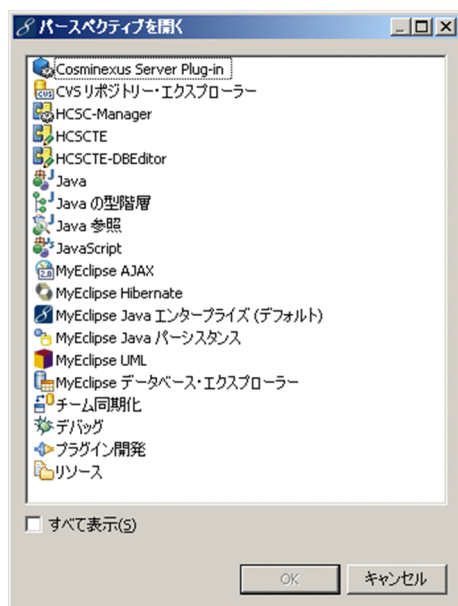
3.5.5 MyEclipse プロジェクトのインポート

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 , HelloServiceAdapter サンプルプログラムの場合を例に , インポート手順を説明します。

参考

対応する XML スキーマが MyEclipse に登録されていない XML ファイルをインポートした場合 , 警告が発生しますが , 問題なくご利用いただけます。

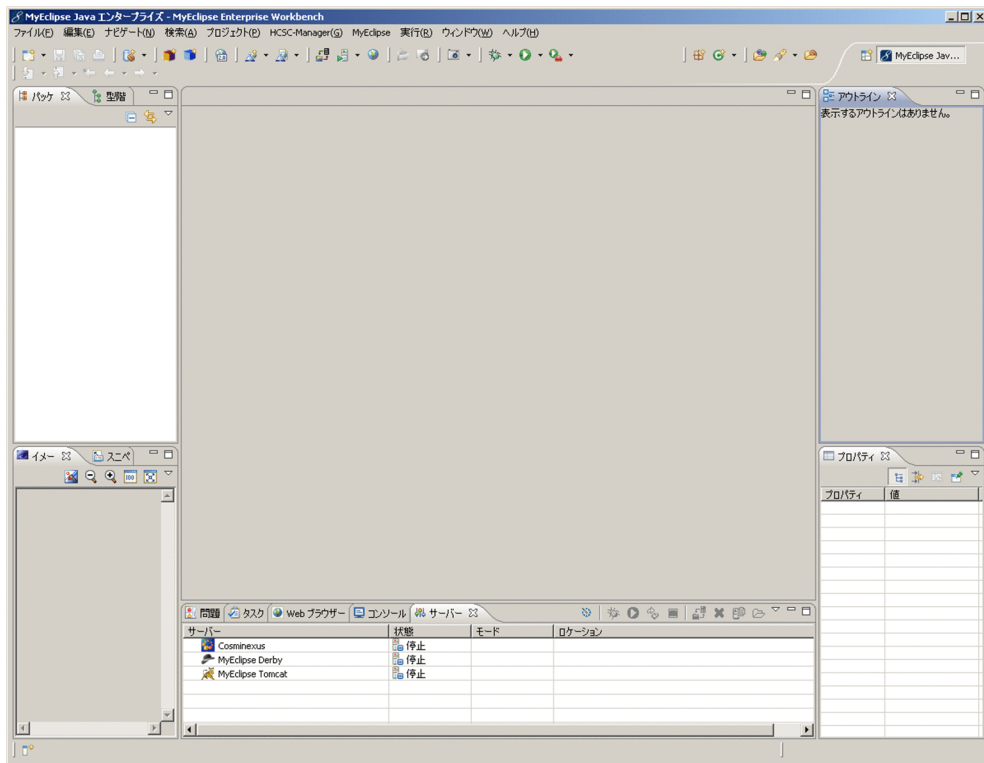
1. MyEclipse のメニューから [ウィンドウ] - [パースペクティブを開く] - [その他] を選択します。
[パースペクティブを開く] ダイアログが表示されます。



3. サンプルプログラムの環境を準備する

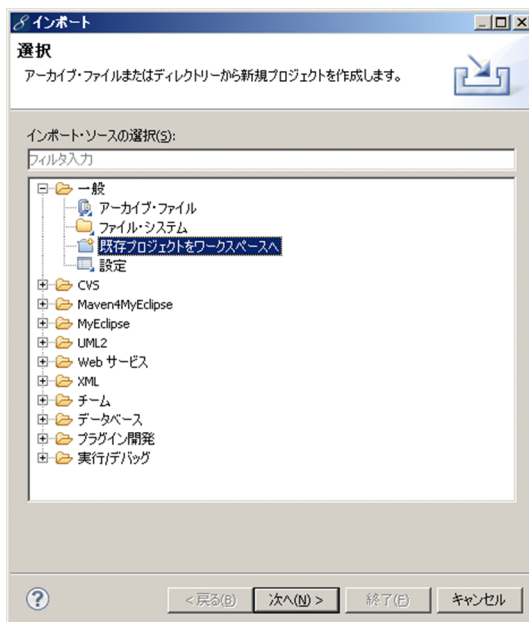
2. [MyEclipse Java エンタープライズ (デフォルト)] を選択し , [OK] ボタンをクリックします。

[MyEclipse Java エンタープライズ] パースペクティブが表示されます。

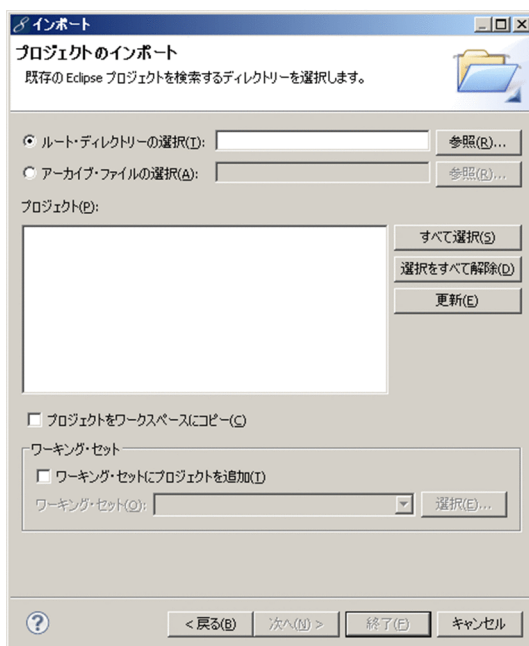


3. メニューから [ファイル] - [インポート] を選択します。
選択画面が表示されます。

4. ツリービューから [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。
プロジェクトのインポート画面が表示されます。



3. サンプルプログラムの環境を準備する

6. [ルート・ディレクトリーの選択] を選択し , [参照] ボタンをクリックします。
ファイルの参照画面が表示されます。
7. サンプルプログラムが格納されているディレクトリ (<uCosminexus Service Architect
のインストールディレクトリ >¥CSCTE¥Samples¥HelloServiceAdapter) を選択し ,
[OK] ボタンをクリックします。
プロジェクトのインポート画面に戻ります。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ください。
なお、ほかのサンプルプログラムが格納されているディレクトリは、次のところにあ
ります。

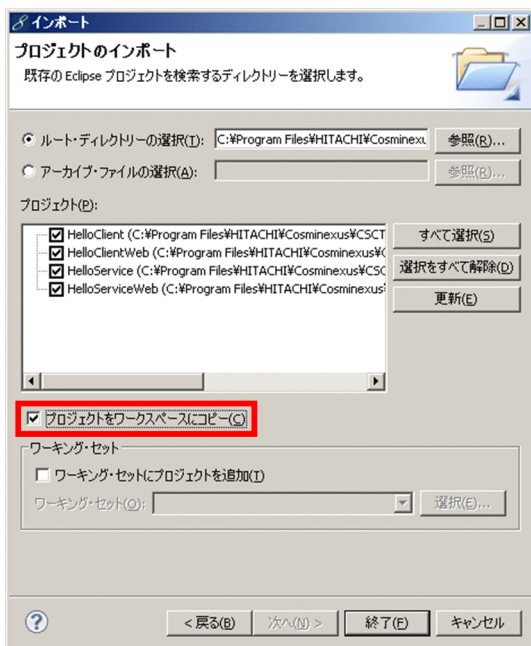
HelloBusinessProcess サンプルプログラム

<uCosminexus Service Architect のインストールディレクトリ
>¥CSCTE¥Samples¥HelloBusinessProcess

商品手配サンプルプログラム

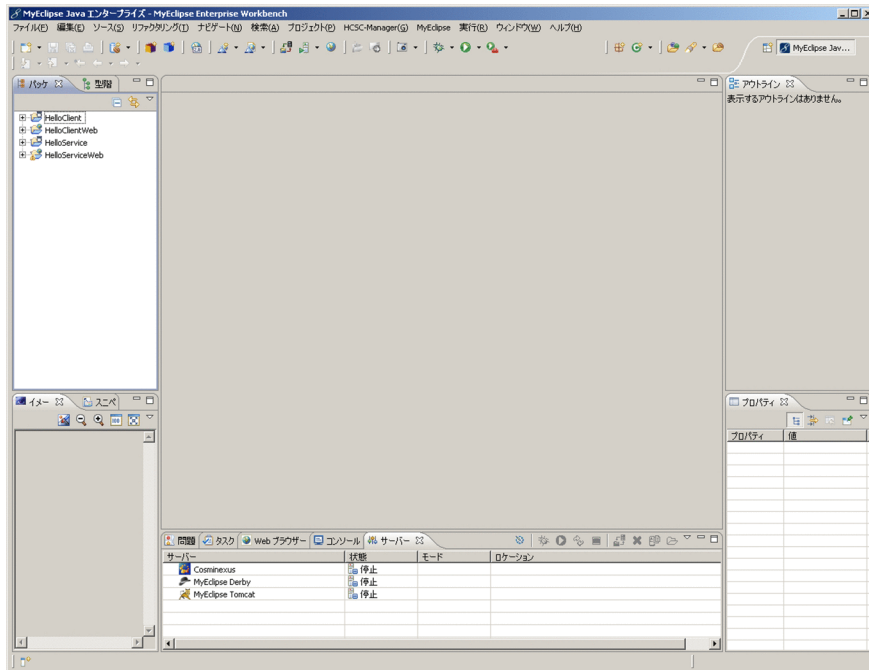
<uCosminexus Service Architect のインストールディレクトリ
>¥CSCTE¥Samples¥ 商品手配

8. [プロジェクトをワークスペースにコピー] のチェックが入っていない場合は ,
チェックを入れます。
チェックを入れないと、サンプルプログラムが削除されるおそれがあります。必ず
チェックを入れてください。



9. [終了] ボタンをクリックします。

パースペクティブに選択したプロジェクトがインポートされます。



3.5.6 Web プロジェクトのデプロイ

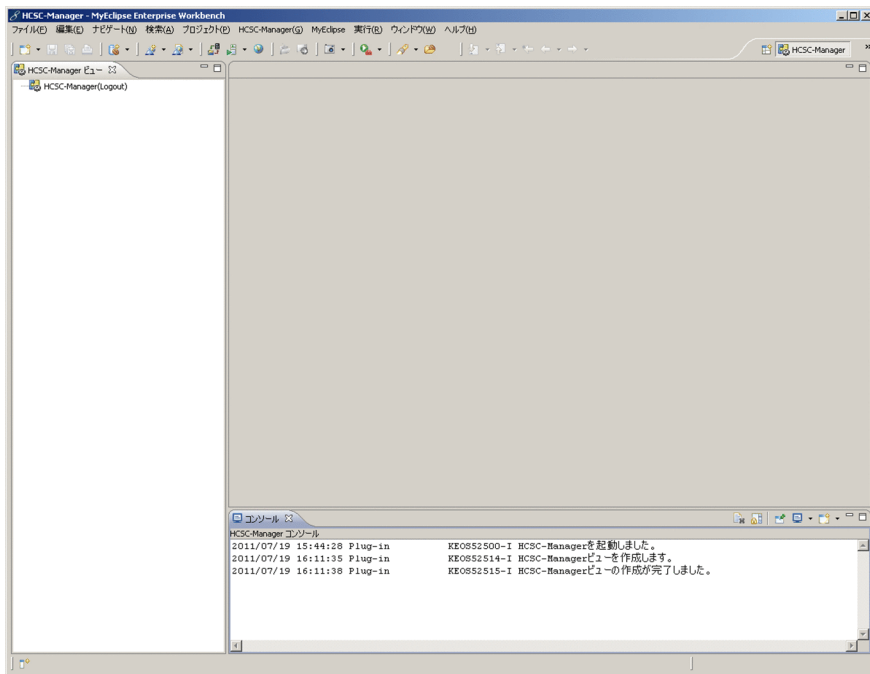
Web プロジェクトは、サンプルプログラムごとにデプロイします。ここでは、HelloServiceAdapter サンプルプログラムの場合を例に、Web プロジェクトのデプロイの手順を説明します。

1. MyEclipse のメニューから [ウィンドウ] - [パースペクティブを開く] - [その他] を選択します。

[パースペクティブを開く] ダイアログが表示されます。

3. サンプルプログラムの環境を準備する

2. [HCSC-Manager] を選択し , [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。

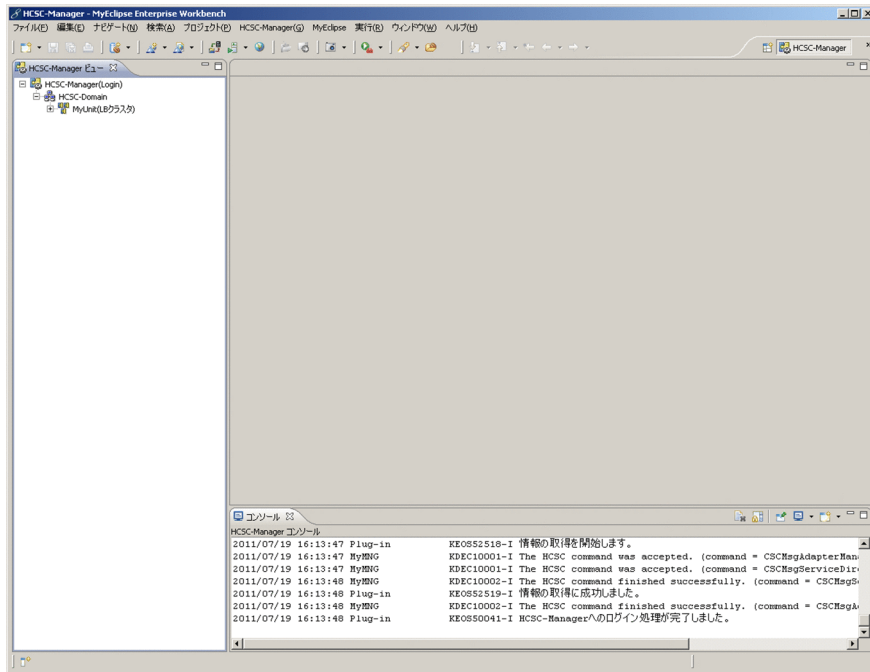


3. [HCSC-Manager] ビューの [HCSC-Manager(Login)] を右クリックし , [ログイン] を選択します。
ログイン画面が表示されます。

3. サンプルプログラムの環境を準備する

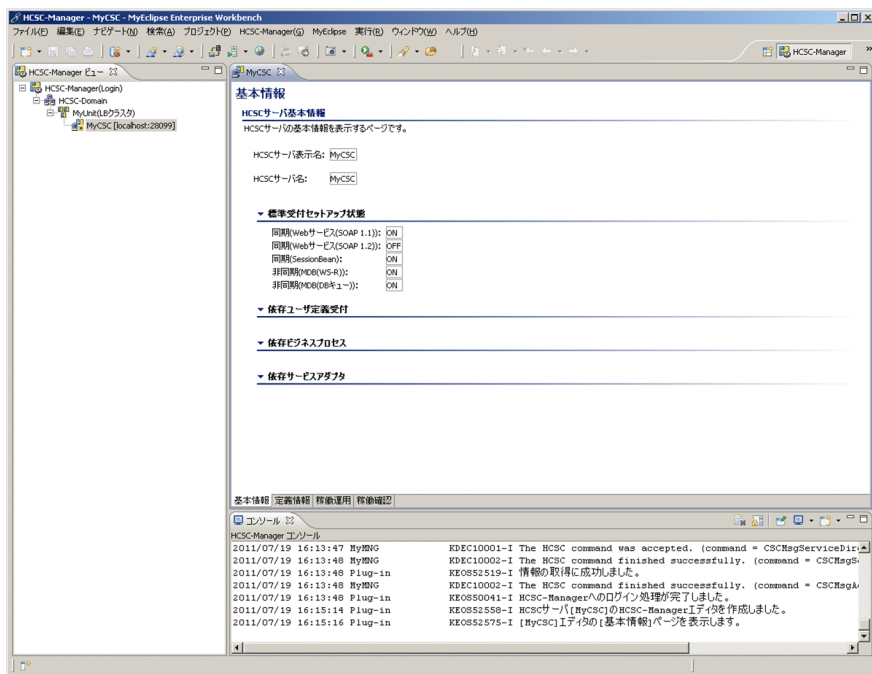
4. [ユーザ ID] に「admin」を,[パスワード] に「admin」を入力し,[OK] ボタンをクリックします。

HCSC-Manager へのログインが完了し,[HCSC-Manager(Login)] と表示されます。

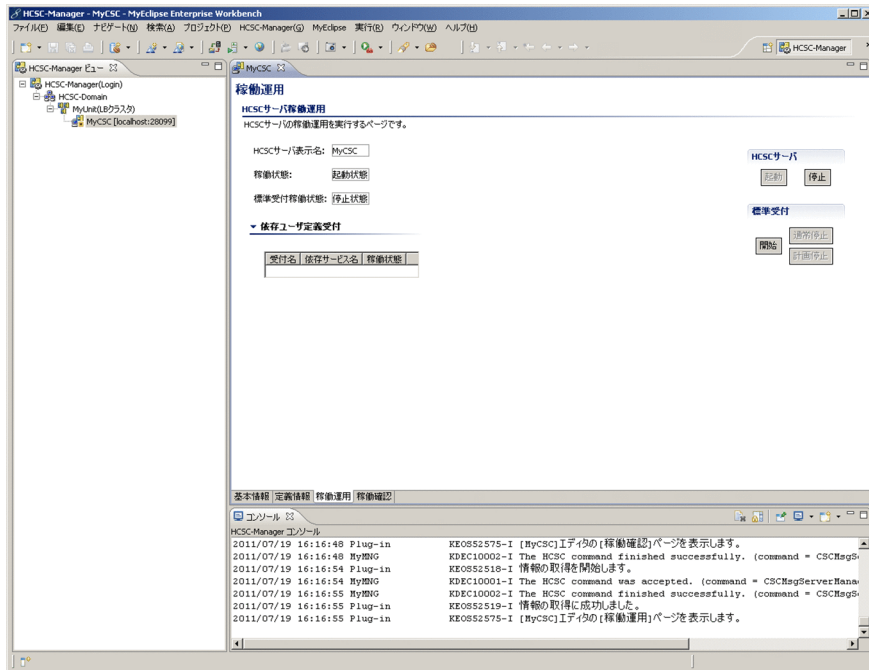


3. サンプルプログラムの環境を準備する

5. [HCSC-Manager] ビューで [MyUnit(LB クラスタ)] の配下の [MyCSC [localhost:28099]] をダブルクリックします。
[MyCSC [localhost:28099]] の基本情報ページが表示されます。



6. ページの下部にある [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] の場合は, [起動] ボタンをクリックします。[起動状態] の場合は, 操作 10. へ進んでください。
HCSC サーバの起動を確認するメッセージが表示されます。
8. [OK] をクリックします。
処理中であることを知らせるメッセージが表示されます。処理が終了すると, HCSC サーバが起動されたことを知らせるメッセージが表示されます。
9. [OK] ボタンをクリックします。
[稼働運用] ページに戻ります。HCSC サーバの [稼働状態] が [起動状態] になっていることを確認してください。
10. [HCSC-Manager] ビューの [HCSC-Manager(Login)] を右クリックし, [ログアウト] を選択します。
HCSC-Manager からのログアウトを確認するメッセージが表示されます。
11. [OK] ボタンをクリックします。
HCSC-Manager からのログアウトが完了し, [HCSC-Manager(Logout)] と表示されます。

3. サンプルプログラムの環境を準備する

12. コマンドプロンプトで次のコマンドを実行して、HCSC サーバを停止します。

```
cscsvstop
```

HCSC サーバが停止します。

13. メニューから [ウィンドウ] - [パースペクティブを開く] - [その他] を選択します。

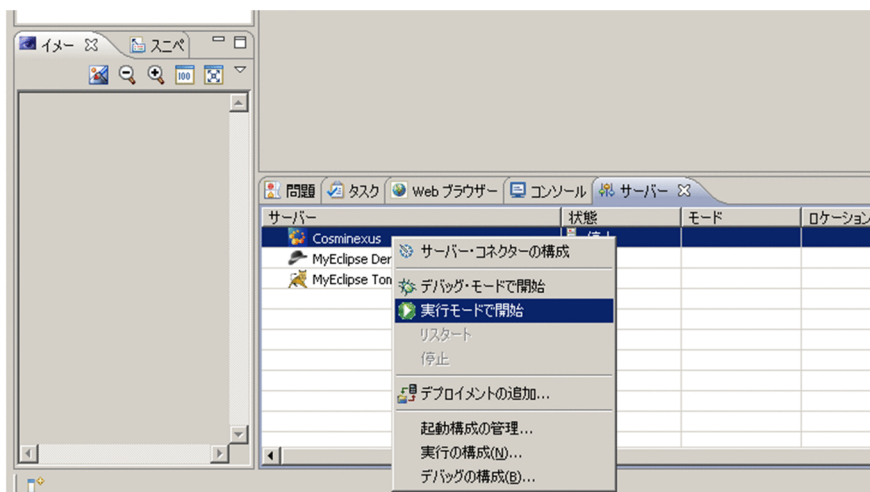
[パースペクティブを開く] ダイアログが表示されます。

14. [MyEclipse Java エンタープライズ (デフォルト)] を選択し、[OK] ボタンをクリックします。

[MyEclipse Java エンタープライズ] パースペクティブが表示されます。

15. [サーバー] ビューの [Cosminexus] サーバを右クリックし、[実行モードで開始] を選択します。

Cosminexus サーバが表示されていない場合は、「3.5.4(2) J2EE サーバの選択」に示す手順を実施してください。



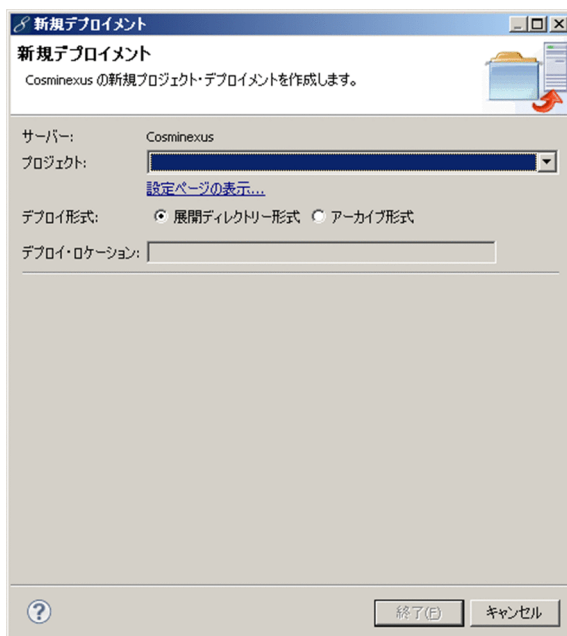
ログイン画面が表示されます。

16. [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。

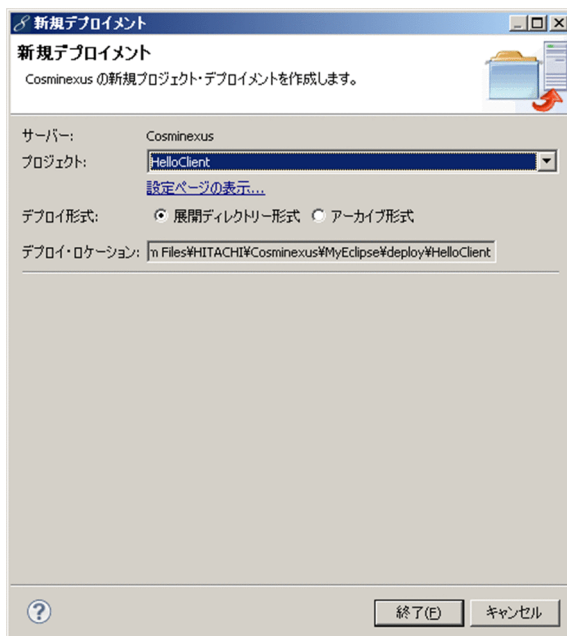
処理中であることを知らせるメッセージが表示されたあと、Cosminexus サーバが起動されます。

17. [サーバー] ビューの [Cosminexus] サーバを右クリックし, [デプロイメントの追加] を選択します。

[新規デプロイメント] ダイアログが表示されます。



18. プロジェクトのドロップダウンリストから [HelloClient] を選択し, デプロイ形式の [展開ディレクトリ形式] を選択します。



3. サンプルプログラムの環境を準備する

19[終了] ボタンをクリックします。

処理中であることを知らせるメッセージが表示されます。処理終了後,[MyEclipse Java エンタープライズ] パースペクティブに戻ります。

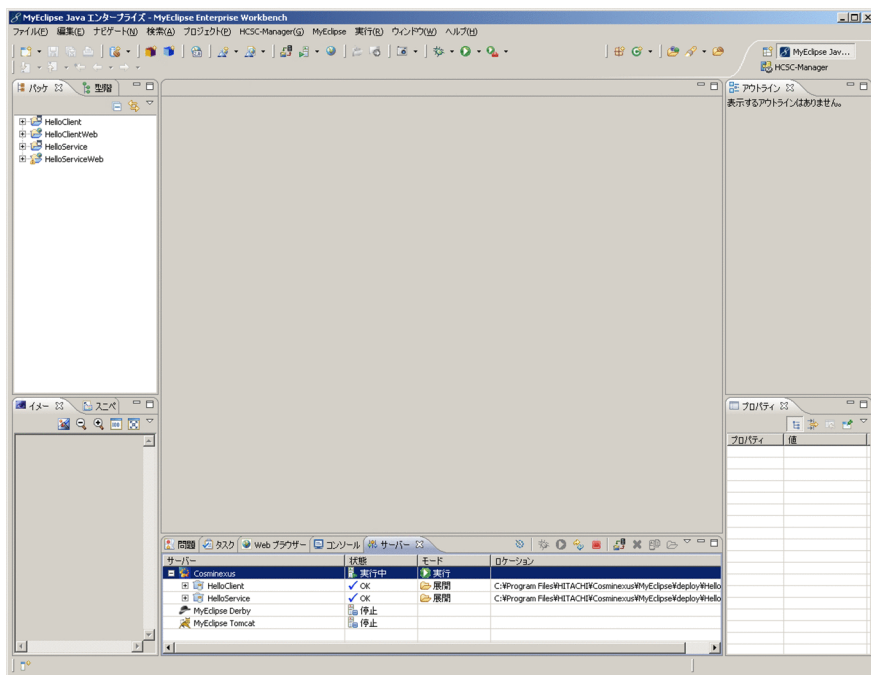
20[サーバー] ビューの [Cosminexus] サーバを右クリックし,[デプロイメントの追加] を選択します。

[新規デプロイメント] ダイアログが表示されます。

21.プロジェクトのドロップダウンリストから [HelloService] を選択し,[展開ディレクトリ形式] を選択します。

22[終了] ボタンをクリックします。

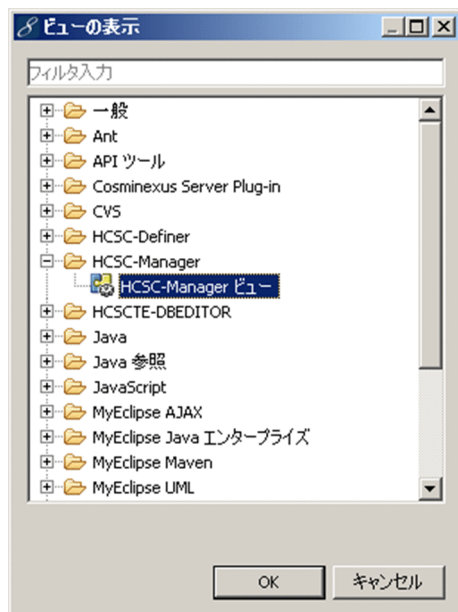
処理中であることを知らせるメッセージが表示されます。処理終了後,[MyEclipse Java エンタープライズ] パースペクティブに戻ります。[サーバー] ビューの [Cosminexus] の配下に [HelloClient] と [HelloService] プロジェクトが存在することを確認してください。



3.5.7 HCSC サーバへの定義内容の配備

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、HelloServiceAdapter サンプルプログラムの場合を例に、配備の手順を説明します。

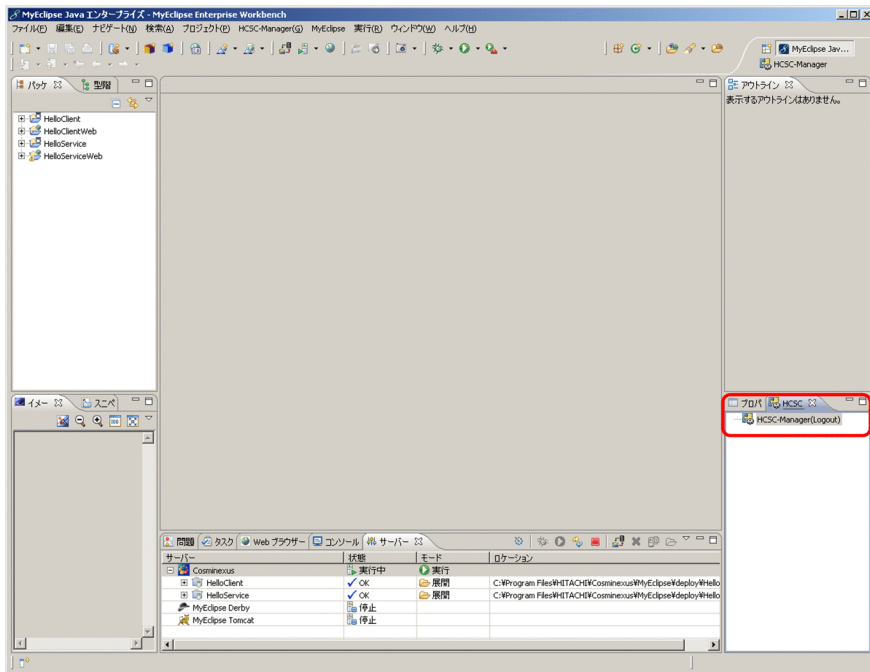
1. メニューから [ウィンドウ] - [ビューの表示] - [その他] を選択します。
[ビューの表示] ダイアログが表示されます。



2. [ビューの表示] ダイアログで [HCSC-Manager] - [HCSC-Manager ビュー] を選択し, [OK] ボタンをクリックします。
[HCSC-Manager] ビューが表示されます。

3. サンプルプログラムの環境を準備する

3. HCSC-Manager からログアウトしていることを確認します。
ログアウトしていない場合は、エラーになることがあります。



4. コマンドプロンプトを起動します。
5. 次のコマンドを入力して、リポジトリをインポートします。

```
cscrcptcl -user admin -pass admin -import <uCosminexus Service Architect  
のインストールディレクトリ  
>%CSCTE%\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

```
<uCosminexus Service Architect のインストールディレクトリ  
>%CSCTE%\Samples\HelloBusinessProcess\Repository\HelloBusinessProcess.zip
```

商品手配サンプルプログラム

```
<uCosminexus Service Architect のインストールディレクトリ  
>%CSCTE%\Samples\商品手配\Repository\商品手配.zip
```

6. 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
```

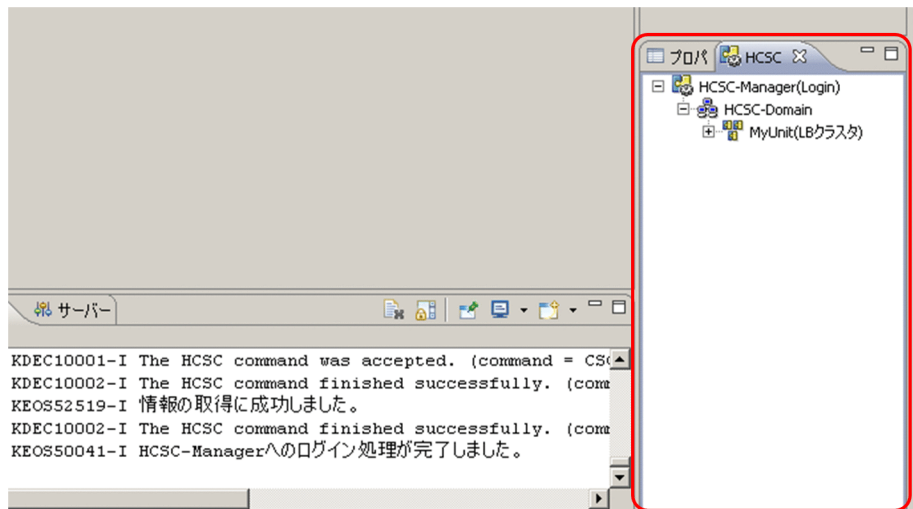

7. 次のコマンドを入力して、配備した HCSC コンポーネントを起動します。

```
csccompostart -user admin -pass admin -csc MyCSC -all
```

8. 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
```

9. [HCSC-Manager] ビューの [HCSC-Manager(Login)] を右クリックし、[ログイン] を選択します。
ログイン画面が表示されます。
- 10.[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。
[HCSC-Manager] ビューが表示されます。



- 11.[Hello サービスアダプタ] が起動していること、および標準受付が開始されていることを確認します。

[Hello サービスアダプタ] の起動を確認する方法

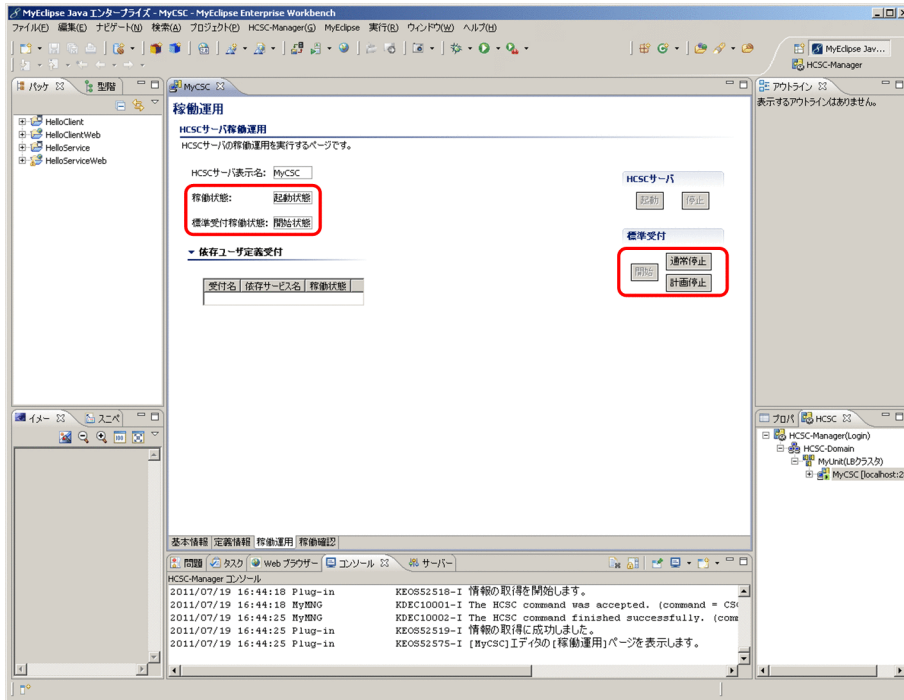
[HCSC-Manager] ビューの [HCSC-Manager(Login)] - [HCSC-Domain] - [MyUnit(LB クラスタ)] - [MyCSC[localhost:28099]] - [Hello サービスアダプタ] をダブルクリックして [基本情報] ページを表示させます。[稼働運用] タブをクリックして [稼働運用] ページを開きます。[稼働状態] が [開始状態] になっているかどうかを確認してください。

標準受付の開始を確認する方法

[HCSC-Manager] ビューの [HCSC-Manager(Login)] - [HCSC-Domain]

3. サンプルプログラムの環境を準備する

- [MyUnit(LB クラスタ)] - [MyCSC[localhost:28099]] をダブルクリックします。[MyCSC] の [稼働運用] ページを表示させ、[標準受付稼働状態] が [開始状態] になっていることを確認してください。[開始] ボタンが非活性になっていれば、[標準受付] は [開始状態] です。



12[HCSC-Manager] ビューの [HCSC-Manager(Login)] を右クリックし、[ログアウト] を選択します。

HCSC-Manager からのログアウトを確認するメッセージが表示されます。

13[OK] ボタンをクリックします。

HCSC-Manager からのログアウトが完了し、[HCSC-Manager(Logout)] と表示されます。

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。

4

サンプルプログラムを実行する

この章では、サンプルプログラムの実行手順について説明します。

4.1 サンプルプログラムの実行

4.2 ビジネスプロセスを適用しないパターン

4.3 ビジネスプロセスを適用するパターン

4.4 複数サービスのプロセスを統合したパターン

4.5 プロジェクトの削除

4.6 テスト環境の停止

4.1 サンプルプログラムの実行

サンプルプログラムの環境を準備したあとは、次の場所を参照してサンプルプログラムを実行します。

HelloServiceAdapter サンプルプログラムを実行する

「4.2 ビジネスプロセスを適用しないパターン」を参照してください。

HelloBusinessProcess サンプルプログラムを実行する

「4.3 ビジネスプロセスを適用するパターン」を参照してください。

商品手配サンプルプログラムを実行する

「4.4 複数サービスのプロセスを統合したパターン」を参照してください。

なお、サンプルプログラムを実行するときは、テスト環境が起動されている必要があります。テスト環境を起動する方法については、「3.5.2 テスト環境の起動」を参照してください。

！ 注意事項

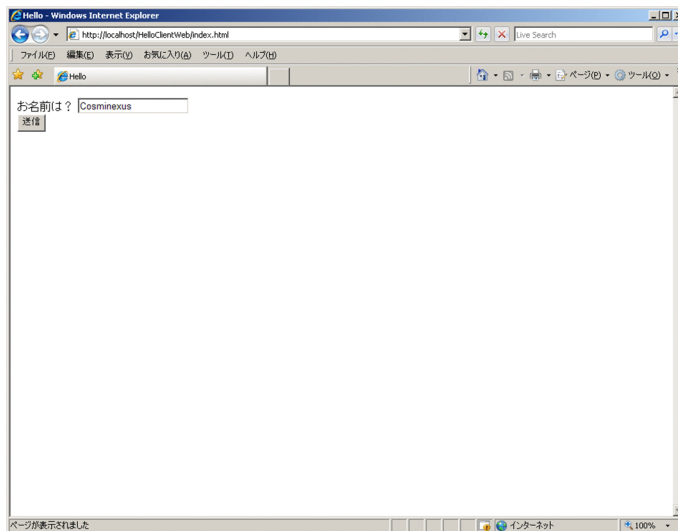
このサンプルプログラムでは、パターンごとにプロジェクトが用意されています。そのため、上記のどれかのサンプルプログラムを実行したあとに、別のサンプルプログラムを実行する場合は、次の手順に従って実行してください。

1. 先に実行したサンプルプログラムのプロジェクトの削除
プロジェクトの削除方法については、「4.5 プロジェクトの削除」を参照してください。
 2. 次に実行するサンプルプログラムの MyEclipse プロジェクトのインポート
MyEclipse プロジェクトのインポート方法については、「3.5.5 MyEclipse プロジェクトのインポート」を参照してください。
 3. 次に実行するサンプルプログラムの Web プロジェクトのデプロイ
Web プロジェクトのデプロイ方法については、「3.5.6 Web プロジェクトのデプロイ」を参照してください。
 4. HCSC サーバへの、次に実行するサンプルプログラムの定義内容の配備
HCSC サーバへの定義内容の配備方法については、「3.5.7 HCSC サーバへの定義内容の配備」を参照してください。
 5. サンプルプログラムの実行
-

4.2 ビジネスプロセスを適用しないパターン

HelloServiceAdapter サンプルプログラムを動かしてみましょう。動作手順を次に示します。

1. ブラウザから次に示す URL を入力します。
`http://localhost/HelloClientWeb/index.html`
HelloServiceAdapter サンプルプログラムが起動されます。
2. 「お名前は？」に文字列を入力します。



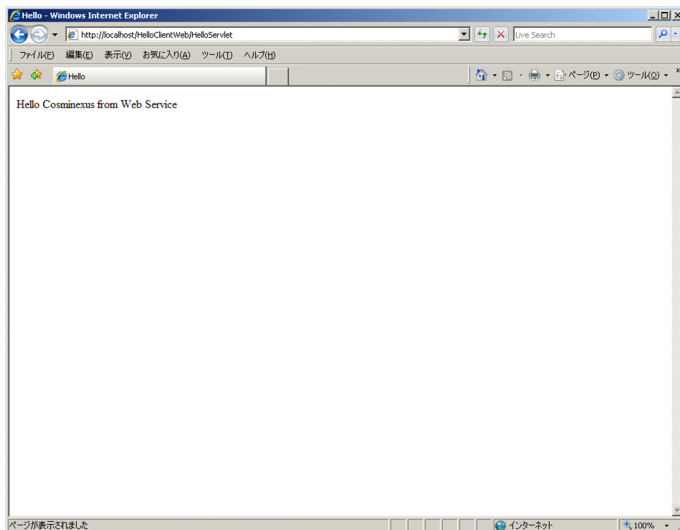
3. [送信] ボタンをクリックします。
次のように表示されます。

4. サンプルプログラムを実行する

正常に応答した場合

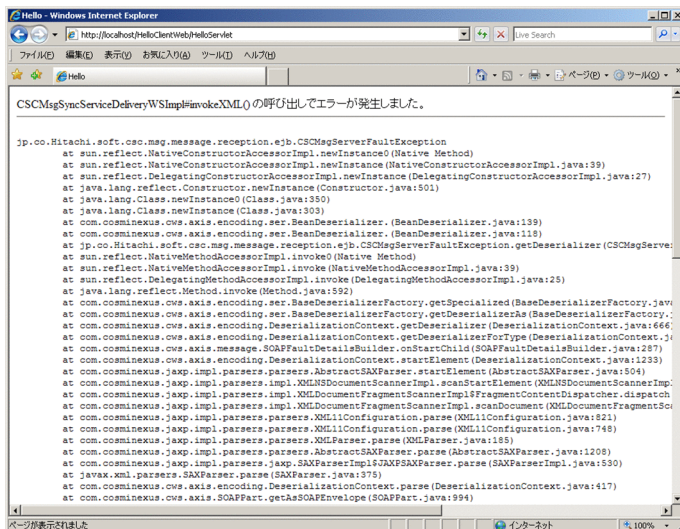
画面に次のように表示されます。

Hello <入力文字列> from Web Service
(< > は半角空白)



エラーが発生した場合

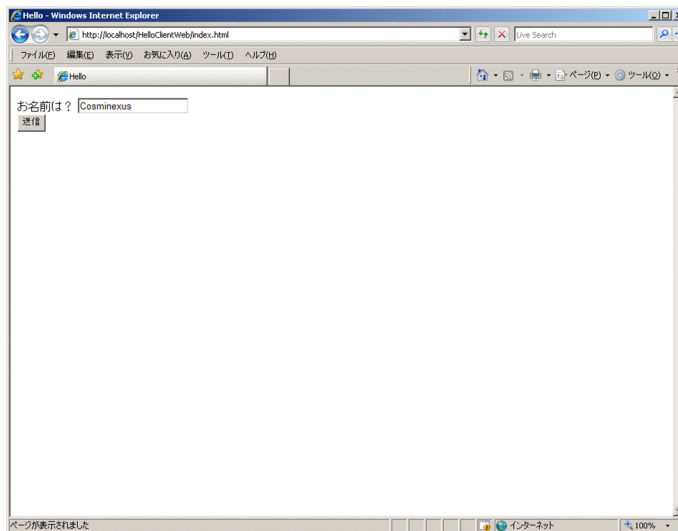
エラーの内容が表示されます。



4.3 ビジネスプロセスを適用するパターン

HelloBusinessProcess サンプルプログラムを動かしてみましょう。動作手順を次に示します。

1. ブラウザから次に示す URL を入力します。
`http://localhost/HelloClientWeb/index.html`
HelloBusinessProcess サンプルプログラムが起動されます。
2. 「お名前は？」に文字列を入力します。



4. サンプルプログラムを実行する

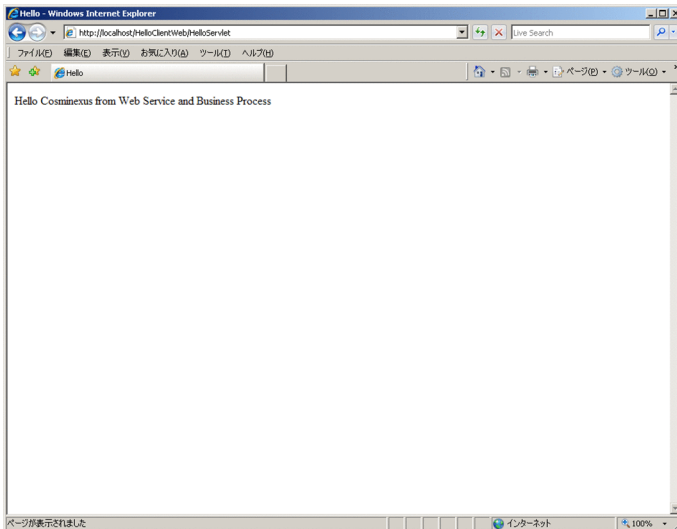
3. [送信] ボタンをクリックします。

次のように表示されます。

正常に応答した場合

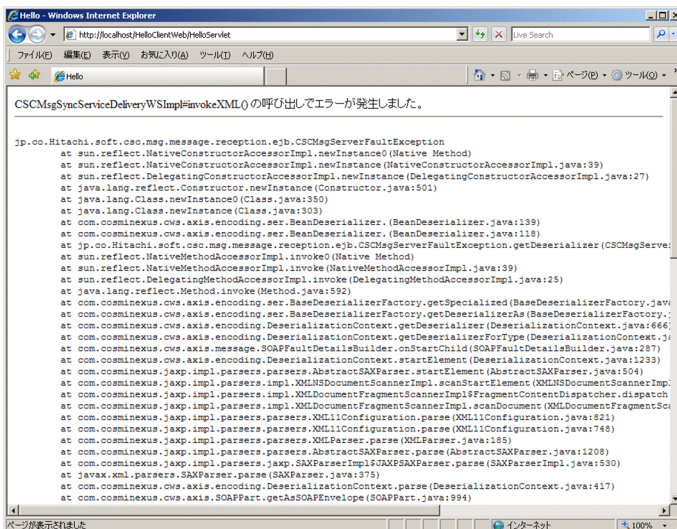
画面に次のように表示されます。

Hello < 入力文字列 > from Web Service and Business
Process
(は半角空白)



エラーが発生した場合

エラーの内容が表示されます。



4.4 複数サービスのプロセスを統合したパターン

商品手配サンプルプログラムを動かしてみましょう。動作手順を次に示します。

1. ブラウザから次に示す URL を入力します。
`http://localhost/ArrangementClientWeb/index.html`
商品手配サンプルプログラムが起動されます。
2. 商品名と個数を選択します。



4. サンプルプログラムを実行する

3. [手配] ボタンをクリックします。

次のように表示されます。

正常に応答した場合（手配が完了した場合）

手配が完了したことを知らせるメッセージ、商品名、個数、および配送番号が表示されます。



正常に応答した場合（在庫がない場合）

在庫がないことを知らせるメッセージ、商品、および個数が表示されます。



エラーが発生した場合

エラーの内容が表示されます。



商品の在庫数を初期値に戻す方法

商品手配サンプルプログラムでは、各商品の総在庫数は 10 個です。手配が完了すると、その個数分、在庫が減ります。在庫がなくなった場合は、Cosminexus サーバを再起動するか、または J2EE プロジェクトをもう一度デプロイ（リデプロイ）してください。在庫数が 10 個に戻ります。Cosminexus サーバの再起動、および J2EE プロジェクトをリデプロイする方法を次に示します。

Cosminexus サーバの再起動方法

1. MyEclipse のメニューから [ウィンドウ] - [パースペクティブを開く] - [その他] を選択します。
[パースペクティブを開く] ダイアログが表示されます。
2. [MyEclipse Java エンタープライズ (デフォルト)] を選択し、[OK] ボタンをクリックします。
[MyEclipse Java エンタープライズ] パースペクティブが表示されます。
3. [サーバー] ビューの [Cosminexus] サーバを右クリックし、[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 のメニューから [ウィンドウ] - [パースペクティブを開く] - [その他] を選択します。
[パースペクティブを開く] ダイアログが表示されます。
2. [MyEclipse Java エンタープライズ (デフォルト)] を選択し、[OK] ボタンをクリックします。
[MyEclipse Java エンタープライズ] パースペクティブが表示されます。
3. [サーバー] ビューの [Cosminexus] の配下にある [HelloClient] および

4. サンプルプログラムを実行する

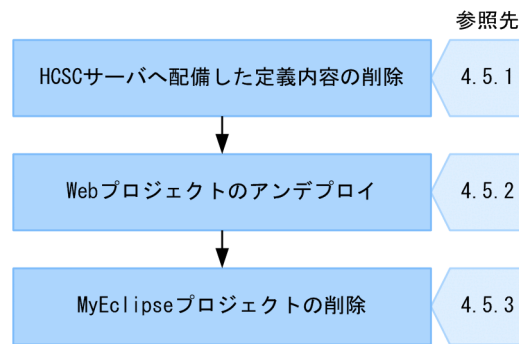
[HelloService] を右クリックし,[リデプロイ] を選択します。

4.5 プロジェクトの削除

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、HelloServiceAdapter サンプルプログラムの場合を例に、プロジェクトの削除の手順を説明します。なお、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。MyEclipse プロジェクトのインポートについては、「3.5.5 MyEclipse プロジェクトのインポート」を参照してください。

プロジェクトの削除の流れを次の図に示します。

図 4-1 プロジェクトの削除の流れ



それぞれの作業の概要を説明します。

1. HCSC サーバへ配備した定義内容の削除

サンプルプログラムごとに、「3.5.7 HCSC サーバへの定義内容の配備」で HCSC サーバへ配備した定義内容を削除します。詳細については、「4.5.1 HCSC サーバへ配備した定義内容の削除」を参照してください。

2. Web プロジェクトのアンデプロイ

サンプルプログラムごとに、「3.5.6 Web プロジェクトのデプロイ」でデプロイした Web プロジェクトをアンデプロイします。詳細については、「4.5.2 Web プロジェクトのアンデプロイ」を参照してください。

3. MyEclipse プロジェクトの削除

サンプルプログラムごとに、「3.5.5 MyEclipse プロジェクトのインポート」でインポートした MyEclipse プロジェクトを削除します。詳細については、「4.5.3 MyEclipse プロジェクトの削除」を参照してください。

以降の項では、この流れに沿ってプロジェクトを削除する手順を説明します。

4.5.1 HCSC サーバへ配備した定義内容の削除

1. コマンドプロンプトで次のコマンドを実行し、標準受付を停止します。

```
cscrcptnstop -user admin -pass admin -csc MyCSC
```

2. コマンドプロンプトで次のコマンドを実行し、HCSC コンポーネントを停止します。

```
csccompostop -user admin -pass admin -csc MyCSC -all
```

3. コマンドプロンプトで次のコマンドを実行し、HCSC サーバへ配備した HCSC コンポーネントを削除します。

```
csccompoundeploy -user admin -pass admin -csc MyCSC -all
```

注意

HCSC サーバへ配備した HCSC コンポーネントをすべて削除しても、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。

4.5.2 Web プロジェクトのアンデプロイ

ここでは、HelloServiceAdapter サンプルプログラムの場合を例に、Web プロジェクトのアンデプロイの手順を説明します。

1. コマンドプロンプトで次のコマンドを実行し、テスト環境の HCSC サーバを停止します。

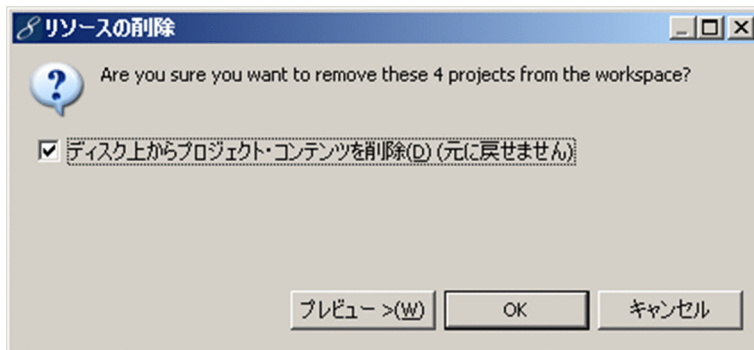
```
cscsvstop
```

2. MyEclipse のメニューから [ウィンドウ] - [パースペクティブを開く] - [その他] をクリックします。
[パースペクティブを開く] ダイアログが表示されます。
3. [MyEclipse Java エンタープライズ (デフォルト)] を選択し、[OK] ボタンをクリックします。
[MyEclipse Java エンタープライズ] パースペクティブが表示されます。

4. [サーバー] ビューの [Cosminexus] の配下にある [HelloClient] を右クリックし ,
[アンデプロイ] を選択します。
ログイン画面が表示されます。
5. [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。
処理中であることを知らせるメッセージが表示されます。同様に [HelloService]
も削除します。

4.5.3 MyEclipse プロジェクトの削除

1. [MyEclipse Java エンタープライズ] パースペクティブのビューから次の四つの
MyEclipse プロジェクトを選択します。
 - HelloClient
 - HelloClientWeb
 - HelloService
 - HelloServiceWeb
2. 右クリックし , [削除] を選択します。
削除を確認する [リソースの削除] ダイアログが表示されます。



3. [ディスク上からプロジェクト・コンテンツを削除] チェックボックスにチェックし ,
[OK] ボタンをクリックします。
[エンタープライズプロジェクトの削除] ダイアログが表示されます。
4. [はい] ボタンをクリックします。
選択した MyEclipse プロジェクトが削除されます。

4.6 テスト環境の停止

uCosminexus Service Architect の使用を終了する場合は、起動中のテスト環境を停止したあと、Eclipse を終了します。

テスト環境を停止する場合は、コマンドプロンプトで次のコマンドを実行します。

```
cscsvstop -system
```

！ 注意事項

テスト環境を停止する前に MyEclipse を終了すると、テスト環境を終了するために、MyEclipse の画面が消えたあともプロセスの終了処理が実施されます。そのため、次に MyEclipse を起動するときに、ほかのプロセスが HCSCTE テンポラリディレクトリを使用している旨のエラーメッセージが表示される場合があります。エラーメッセージが表示された場合の対処方法を次に示します。

1. Windows のタスクマネージャを開きます。
 2. プロセスの一覧から、MyEclipse を動かしている「eclipse.exe」および「javaw.exe」のプロセスが消えるのを待ちます。
 3. プロセスが消えたことを確認したら、MyEclipse を起動します。
-

5

サンプルプログラムの開発を体験する

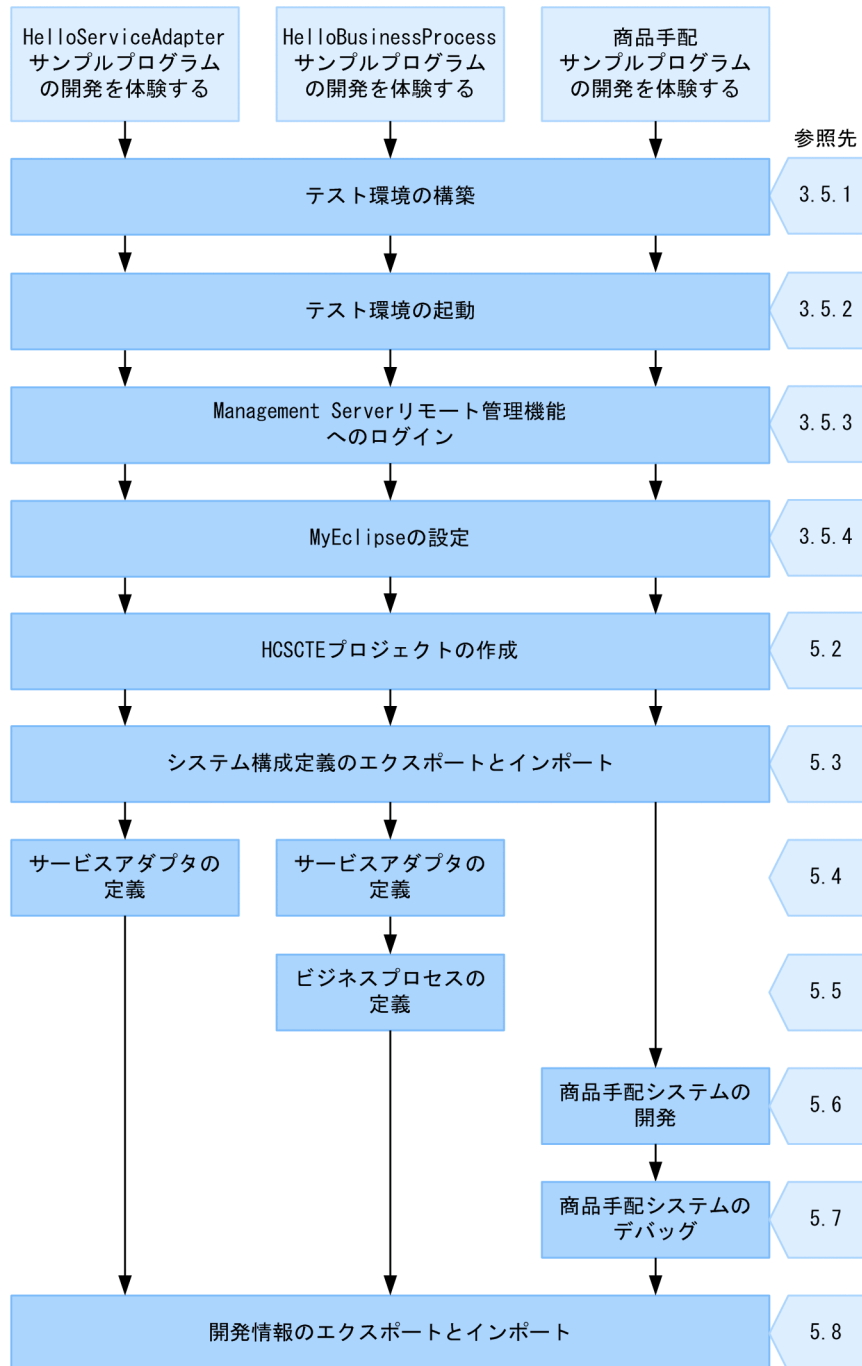
この章では、実際に用意されているサンプルプログラムを使って、サービスアダプタやビジネスプロセスを定義する方法について説明します。

-
- 5.1 サンプルプログラムの開発の手順
 - 5.2 HCSCTE プロジェクトの作成
 - 5.3 システム構成定義のエクスポートとインポート
 - 5.4 サービスアダプタの定義
 - 5.5 ビジネスプロセスの定義
 - 5.6 商品手配システムの開発
 - 5.7 商品手配システムのデバッグ
 - 5.8 開発情報のエクスポートとインポート
 - 5.9 任意の Java プログラムを使用したデータ変換の定義
 - 5.10 アンセットアップとアンインストール
-

5.1 サンプルプログラムの開発の手順

サンプルプログラムの開発の手順を次の図に示します。

図 5-1 サンプルプログラムの開発の手順



1. テスト環境の構築

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。すでにテスト環境を構築している場合は、HCSC 簡易セットアップの画面で一度アンセットアップした

5. サンプルプログラムの開発を体験する

あと、再度構築してください。詳細については、「3.5.1 テスト環境の構築」を参照してください。

2. テスト環境の起動

構築したテスト環境を起動します。詳細については、「3.5.2 テスト環境の起動」を参照してください。

3. Management Server リモート管理機能へのログイン

4. の作業を実施するために、MyEclipse から Management Server リモート管理機能へログインします。詳細については、「3.5.3 Management Server リモート管理機能へのログイン」を参照してください。

4. MyEclipse の設定

サンプルプログラムごとに MyEclipse を設定します。詳細については、「3.5.4 MyEclipse の設定」を参照してください。

5. HCSCTE プロジェクトの作成

サービスアダプタやビジネスプロセスを定義する前に、プロジェクトを作成して、プロパティを設定します。詳細については、「5.2 HCSCTE プロジェクトの作成」を参照してください。

6. システム構成定義のエクスポートとインポート

実行環境のシステム構成定義を開発環境で使用するためにシステム構成定義をエクスポートし、開発環境にインポートします。詳細については、「5.3 システム構成定義のエクスポートとインポート」を参照してください。

7. HCSC コンポーネントの作成

サービス部品を呼び出すサービスアダプタ、複数のサービス部品を呼び出すビジネスプロセスなどの HCSC コンポーネントを作成します。また、作成した HCSC コンポーネントをテスト環境でデバッグします。詳細については、次の個所を参照してください。

- 5.4 サービスアダプタの定義
- 5.5 ビジネスプロセスの定義
- 5.6 商品手配システムの開発
- 5.7 商品手配システムのデバッグ

8. 開発情報のエクスポートとインポート

開発環境の情報を実行環境で使用するために開発環境の情報をエクスポートし、運用環境にインポートします。詳細については、「5.8 開発情報のエクスポートとインポート」を参照してください。

以降の節で、これらの手順について説明します。

また、この章では、任意の Java プログラムを使用したデータ変換を定義するサンプルプログラムの開発、セットアップした環境のアンセットアップ、および uCosminexus Service Architect のアンインストールについても説明します。

5.2 HCSCTE プロジェクトの作成

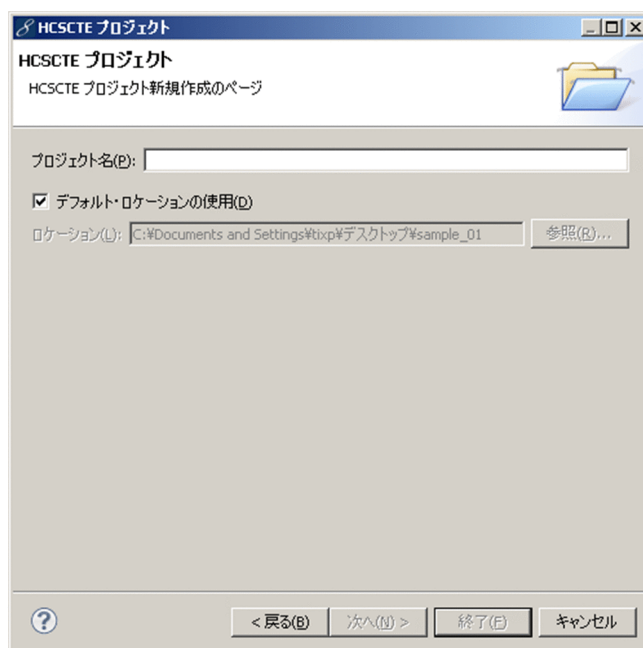
サービスアダプタやビジネスプロセスを定義する前に、HCSCTE プロジェクトを作成します。

！ 注意事項

HCSCTE プロジェクトは、一つのプログラムにつき一つ作成してください。複数のプログラムを開発する場合、各プログラムの HCSCTE プロジェクトのワークスペースは分ける必要があります。同じワークスペースに HCSCTE プロジェクトを複数作成すると、プログラムが正しく動作しません。

HCSCTE 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。



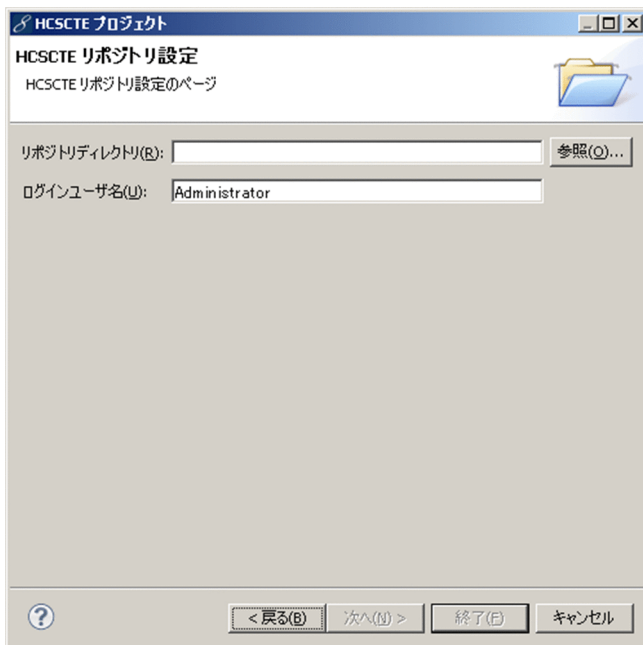
4. 次の項目を設定して、[次へ] ボタンをクリックします。
プロジェクト名

5. サンプルプログラムの開発を体験する

任意の名称を指定します。

デフォルト・ロケーションの使用

[デフォルト・ロケーションの使用] 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。



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。

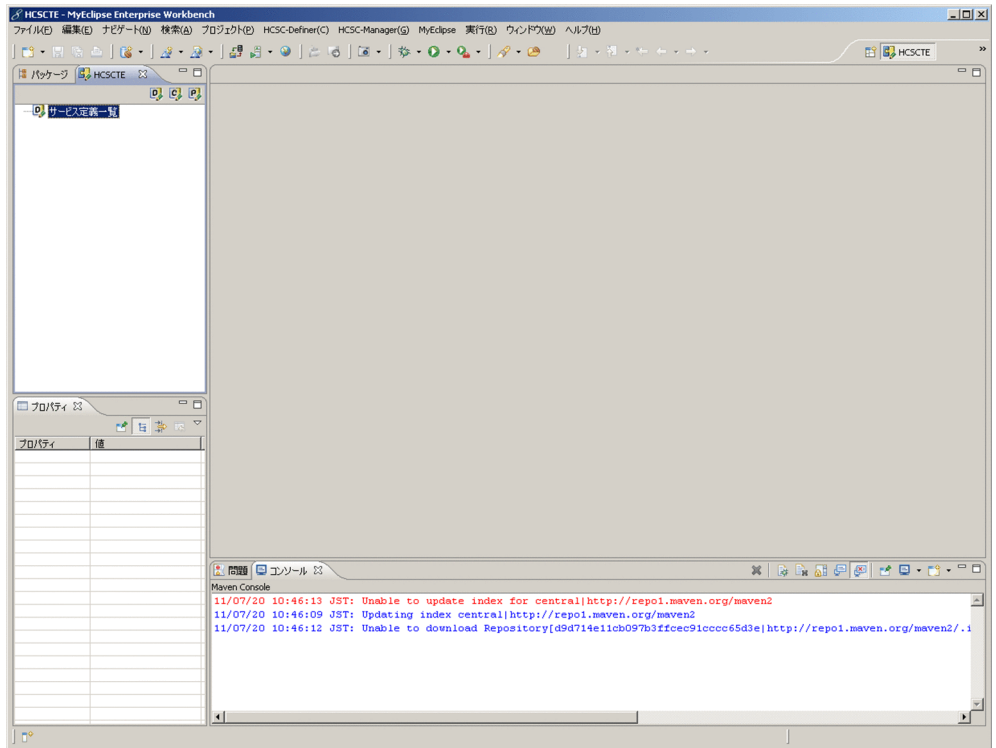
ログインユーザ名

リポジトリログインに使用するユーザ名を指定します。ユーザ名に使用できる文字は半角英数字だけで、長さは 1 ~ 16 文字になります。

パースペクティブを切り替えるかどうかを確認するダイアログが表示された場合は、[はい] ボタンをクリックします。

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ースペクティブが起動します。

5. サンプルプログラムの開発を体験する



5.3 システム構成定義のエクスポートとインポート

uCosminexus Service Architect の HCSC 簡易セットアップ機能を利用して構築した実行環境のシステム構成定義を開発環境で使用します。

ここでは、システム構成定義のエクスポートとインポートの手順を説明します。

(1) エクスポート

実行環境のシステム構成定義は、運用環境のリポジトリに格納されています。実行環境のシステム構成定義を開発環境で使用するために、運用環境で cscreptl コマンドを使用してリポジトリ情報を ZIP 形式のファイルとしてエクスポートします。

cscreptl コマンドは次のように実行します。

```
cscreptl -user admin -pass admin -export <ファイル名>.zip
```

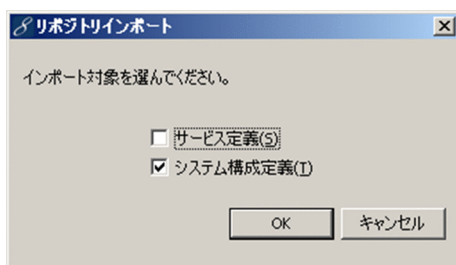
エクスポートした ZIP ファイルはカレントディレクトリに出力されます。

(2) インポート

エクスポートした運用環境のリポジトリ情報の中から、システム構成定義だけを開発環境のリポジトリにインポートします。

手順を次に示します。

1. MyEclipse 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。
リポジトリ情報の ZIP ファイルを選択する [リポジトリインポート] ダイアログが表示されます。
3. 読み込む ZIP ファイルを指定します。
4. [開く] ボタンをクリックします。
インポート対象の定義情報を選択する [リポジトリインポート] ダイアログが表示されます。



5. [システム構成定義] の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。

開発環境のリポジトリとして指定されているディレクトリにシステム構成定義が読み込まれます。

5.4 サービスアダプタの定義

サービスリクエスト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。

サービスアダプタは、呼び出すサービス部品に合わせて定義します。

HelloServiceAdapter サンプルプログラムで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 5-1 HelloServiceAdapter サンプルプログラムで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サービスアダプタの種別を指定します。このサンプルプログラムのサービス部品の種別は Web サービスであるため、「Web サービス」と指定します。
サービス名	Hello サービスアダプタ	定義するサービスアダプタの名称を指定します。
WSDL ファイル	HelloService.wsdl	使用する WSDL ファイルを指定します。 Hello サービスアダプタでは、「HelloService.wsdl (Hello サービス用 WSDL)」を使用します。
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	クライアント定義ファイルを指定します。 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す。ファイル名は「c4webcl.properties」で固定です。 このサンプルプログラムのクライアント定義ファイルには、次の内容が設定されています。 c4web.logger.log_file_prefix=HelloService これは、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」を設定するという処理です。

注

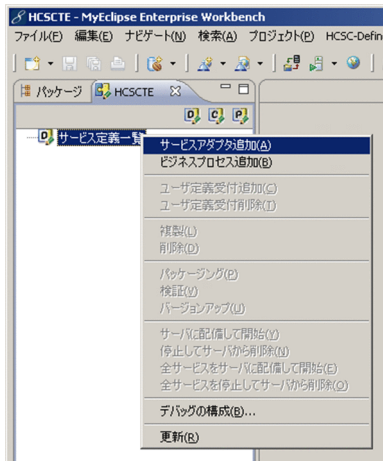
各ファイルの格納場所については、「付録 A.1 HelloServiceAdapter サンプルプログラムの構成」を参照してください。

5.4.1 サービスアダプタの作成

Hello サービスアダプタの作成手順を次に示します。

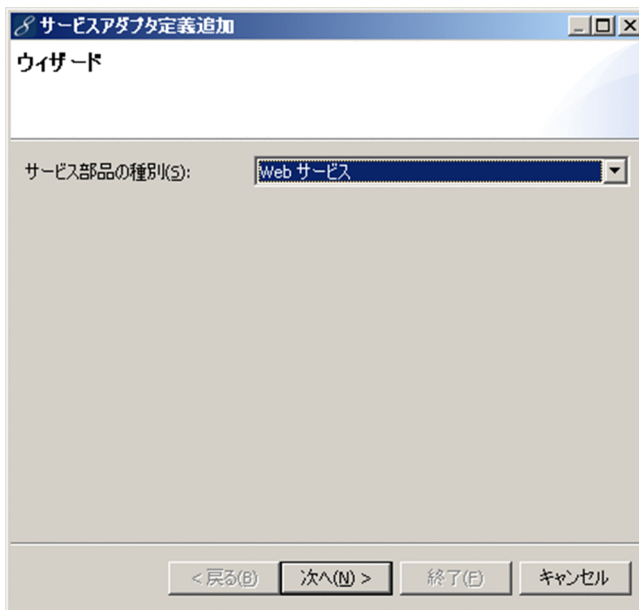
1. MyEclipse を起動します。

- ツリービューの [サービス定義一覧] を選択し、右クリックして、[サービスアダプタ追加] を選択します。



作成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。

- サービス部品の種別で「Web サービス」を選択し、[次へ] ボタンをクリックします。

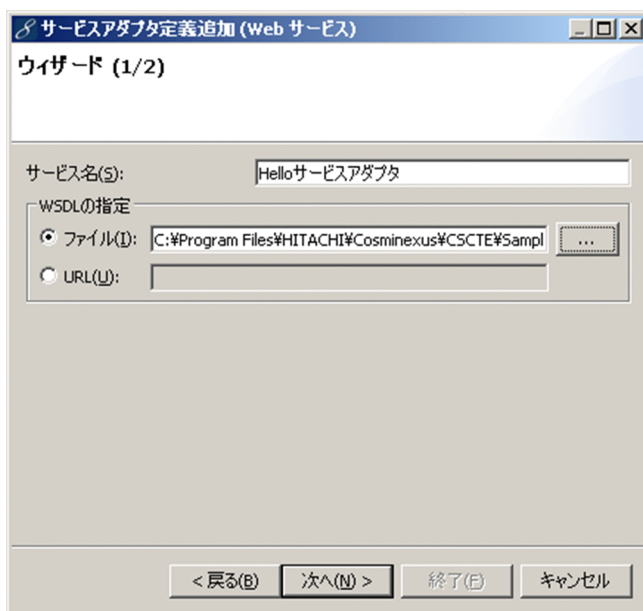


サービスアダプタの追加に必要な情報を入力するダイアログが表示されます。

- サービス名に「Hello サービスアダプタ」を入力します。
- [...] ボタンをクリックします。
[ファイルを開く] ダイアログが表示されます。

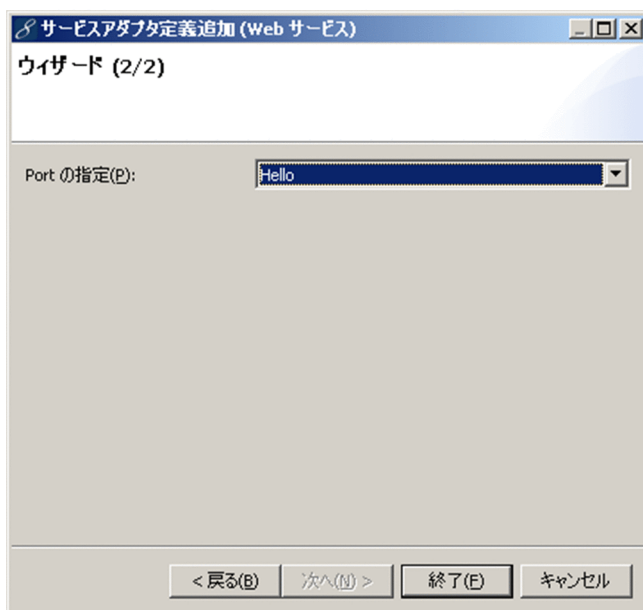
5. サンプルプログラムの開発を体験する

6. 「HelloService.wsdl」を選択し, [開く] ボタンをクリックします。
「HelloService.wsdl」の格納場所は, 「付録 A サンプルプログラムのファイルの構成」を参照してください。
7. [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。

8. Port の指定で「Hello」を選択し, [終了] ボタンをクリックします。



Hello サービス呼び出し用サービスアダプタ「Hello サービスアダプタ」が作成され、サービスアダプタ定義（基本）画面が表示されます。

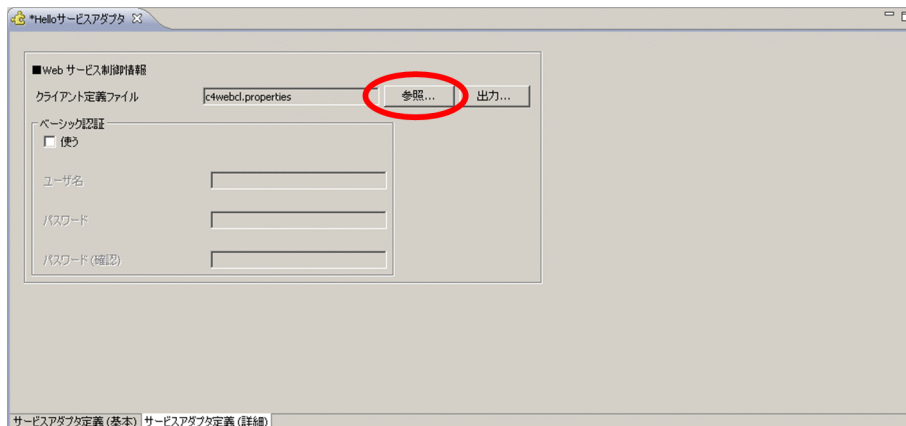
9. サービス ID を「HelAdp」に変更します。

サービスアダプタ定義（基本）画面には、指定した WSDL から読み込んだサービス種別やアクセスするアドレスなどが表示されます。また、WSDL から自動生成した電文フォーマットが要求電文や応答電文の [電文フォーマット] に表示されます。電文フォーマットは、[表示] ボタンをクリックして内容を確認できます。

10. 画面下部の [サービスアダプタ定義（詳細）] タブをクリックします。
サービスアダプタ定義（詳細）画面が表示されます。サービスアダプタ定義（詳細）画面ではクライアント定義ファイルを指定します。

5. サンプルプログラムの開発を体験する

11. サービスアダプタ定義（詳細）画面で , [参照] ボタンをクリックします。



12. 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。

```
<uCosminexus Service Architect のインストールディレクトリ>¥CSCTE¥Samples¥HelloServiceAdapter¥Service¥HelloService¥c4webcl.properties
```

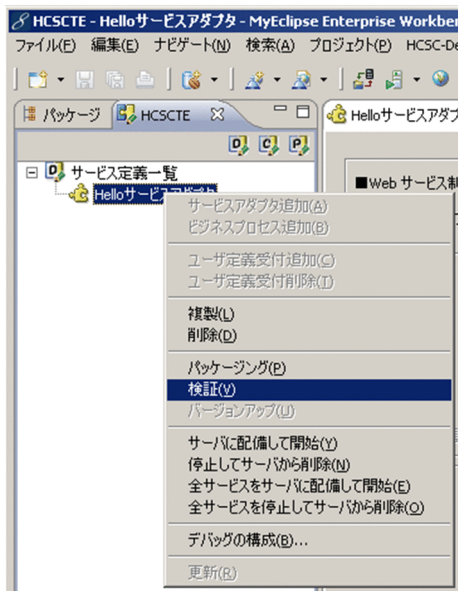
13. メニューで [ファイル] - [保存] を選択します。

サービスアダプタの定義が完了します。

5.4.2 サービスアダプタの検証とパッケージング

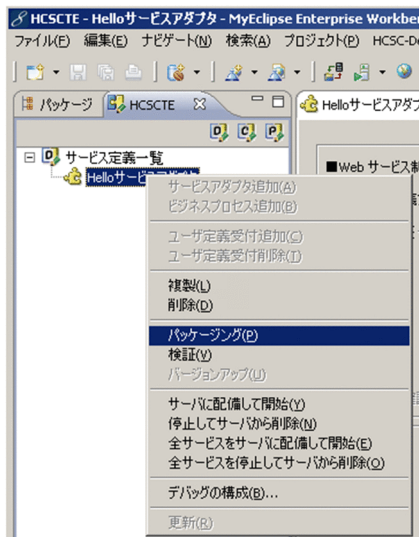
作成した Hello サービスアダプタは , 正しいかどうか検証してから , パッケージングします。検証とパッケージングの手順を次に示します。

1. ツリービューの [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 [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。



パッケージングの処理が開始されます。処理終了後、結果を知らせるメッセージが表示されます。

5. サンプルプログラムの開発を体験する

3. 次のどちらかの操作を実行します。

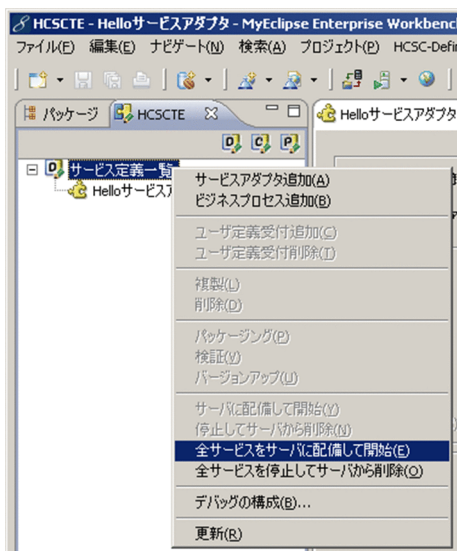
パッケージングが成功した場合は、[OK] ボタンをクリックします。

パッケージングが失敗した場合は、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。

5.4.3 サービスアダプタの配備定義

パッケージングしたサービスアダプタを配備定義します。配備定義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] を右クリックして、[全サービスをサーバに配備して開始] を選択します。



アカウント認証画面が表示されます。

2. [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。
処理中であることを知らせるメッセージが表示されたあと、結果を知らせるメッセージが表示されます。

5.5 ビジネスプロセスの定義

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。

! 注意事項

ビジネスプロセスを定義する前に、サービスアダプタを定義してください。
HelloBusinessProcess サンプルプログラムで使用するサービスアダプタは、Hello サービスアダプタです。Hello サービスアダプタの定義方法については、「5.4 サービスアダプタの定義」を参照してください。

5.5.1 ビジネスプロセスの作成

ビジネスプロセスには、サービスリクエストから受け取ったリクエストをどのようなプロセスで処理していくかを定義します。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、出力画面に表示します。

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。

(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ときに設定する値を次の表に示します。

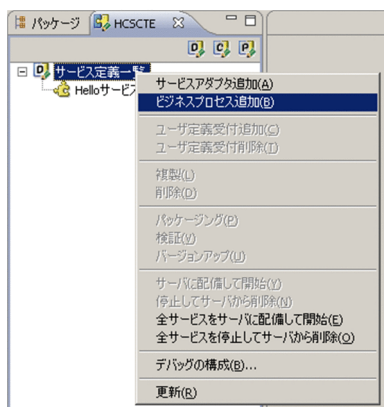
5. サンプルプログラムの開発を体験する

表 5-2 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。記録を残す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 ファイルをインポートするかどうかを指定します。BPEL ファイルをインポートすると、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] を選択し、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「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。

ビジネスプロセス定義追加
ウィザード

ビジネスプロセス名(B): Helloビジネスプロセス

ステータスの永続化: ☒ yes(Y) ☐ no(N)

BPEL ファイル
☐ インポートする(I)
ファイル名(I): ...

アクティビティの配置方法: ☒ 縦方向(V) ☐ 横方向(H)

終了(E) キャンセル

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. ツリービューで「Hello ビジネスプロセス」を選択します。
プロパティビューに Hello ビジネスプロセスのプロパティ一覧が表示されます。
5. プロパティビューで, サービス ID の値のセルをクリックします。
入力できる状態になります。
6. 「HelBP」に変更し, [Enter] を押します。

プロパティ	値
サービス ID	HelBP
ステータスの永続化	yes
バージョン	1
ビジネスプロセス名	Helloビジネスプロセス

5. サンプルプログラムの開発を体験する

7. 変更してよいかどうかを確認するメッセージが表示されるので、[OK] をクリックします。

(2) 変数の設定

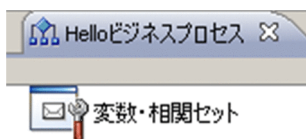
ビジネスプロセスでは、アクティビティを定義するときに変数を使用します。そのため、使用する変数をあらかじめ設定してから、アクティビティを定義します。Hello ビジネスプロセスで使用する変数を次の表に示します。

表 5-3 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	XML	入力データ .xsd
出力データ	XML	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. ツリービューの「変数一覧」を選択します。

3. 変数名に「入力データ」を入力し、種別は「XML」を選択します。

変数・相関セット一覧

変数名(N): 入力データ

種別(K): XML

電文フォーマット(E): 取込(I)... ... 表示(S)...

☒ 部分指定(P)

部分名	指定式	型

行追加(A) 行削除(E)

追加(A) 削除(D) 更新(U)

OK キャンセル

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。
5. [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. サンプルプログラムの開発を体験する

6. [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 (ボディ)」を選択します。[電文フォーマット] には「入力データ」と入力します。

電文フォーマットの取込

サービス/受付

サービス名(S): Helloサービスアダプタ

受付名(R):

取込対象

オペレーション名(O): getHelloString

電文種別(I): 要求電文 (ボディ)

ヘッダルート要素(H):

名前空間(N):

フォルト名(F):

電文フォーマット(M): 入力データ .xsd

OK キャンセル

7. [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。
8. [変数・関連セッの一覧] ダイアログの [追加] ボタンをクリックします。
ツリービューの「変数一覧」に「入力データ」が追加されます。
9. [変数・関連セッの一覧] 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「XML」を選択します。
10. [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。
11. [サービス名] を選択し、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。

- 12.[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 (ボディ)」を選択します。[電文フォーマット] には「出力データ」と入力します。

電文フォーマットの取込

サービス/受付

☒ サービス名(S): Helloサービスアダプタ

☐ 受付名(R):

取込対象

オペレーション名(O): getHelloString

電文種別(I): 応答電文 (ボディ)

ヘッダルート要素(H):

名前空間(N):

フォルト名(F):

電文フォーマット(M): 出力データ .xsd

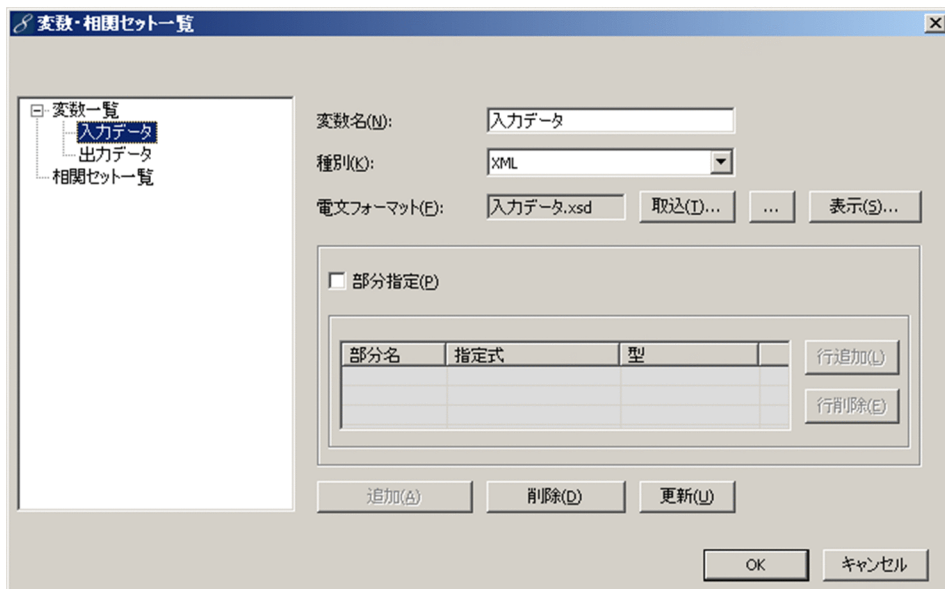
OK キャンセル

- 13.[OK] ボタンをクリックし,[電文フォーマットの取込] ダイアログを閉じます。

- 14.[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。
ツリービューの「変数一覧」に「出力データ」が追加されます。

5. サンプルプログラムの開発を体験する

- 15[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。




(3) アクティビティの配置

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 5-4 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

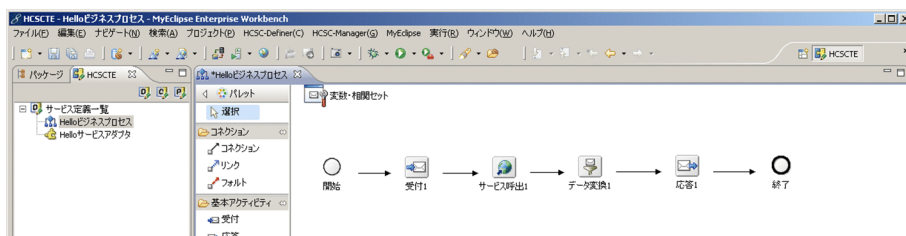
設定する値	説明
受付アクティビティ	サービスリクエストからの応答を受け付けます。
サービス呼出アクティビティ	Hello サービスを呼び出します。
データ変換アクティビティ	文字列を編集します。
応答アクティビティ	処理結果をサービスリクエストへ返します。

アクティビティの配置手順を次に示します。

- 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
- 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、操作 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。



(4) アクティビティの定義

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。

(a) 受付アクティビティ

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。

5. サンプルプログラムの開発を体験する

項目名	設定する値	説明
ボディ割当変数	入力データ	ビジネスプロセスの要求電文のボディ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。
ヘッダ割当変数	設定なし	ビジネスプロセスの要求電文のヘッダに変数を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。
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 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。

(b) 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。

サービス呼出アクティビティ

アクティビティ名(M): Helloサービス

サービス名(S): Helloサービスアダプタ

オペレーション名(O): getHelloString

通信モデル: 同期

要求電文

ボディ割当変数(Q): 入力データ 編集(E)...

ヘッダ割当変数: 設定(H)...

応答電文

ボディ割当変数(R): 出力データ 編集(D)...

ヘッダ割当変数: 設定(I)...

割当関連セット群: 設定(I)...

OK キャンセル

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。
ボディ割当変数 (要求電文)	入力データ	在庫管理サービス呼び出す要求電文のボディ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。
ヘッダ割当変数 (要求電文)	設定なし	在庫管理サービス呼び出す要求電文のヘッダに変数を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。
ボディ割当変数 (応答電文)	出力データ	同期オペレーションから受け取る応答電文のボディ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。
ヘッダ割当変数 (応答電文)	設定なし	同期オペレーションから受け取る応答電文のヘッダに変数を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。

5. サンプルプログラムの開発を体験する

項目名	設定する値	説明
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。

3. [OK] ボタンをクリックします。

(c) 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。

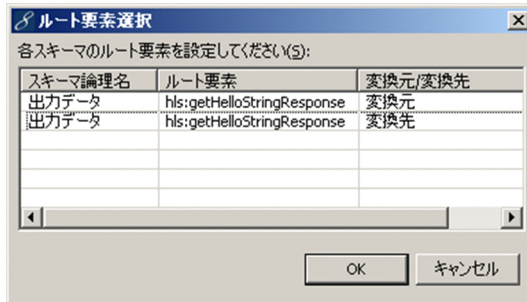
項目名	設定する値	説明
アクティビティ名	出力文字列編集	アクティビティの名称を指定します。
変数 (変換元変数)	出力データ	データの変換元になる変数をドロップダウンリストから選択し, [追加] ボタンをクリックします。
変数 (変換先変数)	出力データ	データの変換先になる変数をドロップダウンリストから選択します。
データ変換定義	出力文字列編集	データ変換定義ファイルに仮想的に付ける名称を指定します。

3. [OK] ボタンをクリックします。

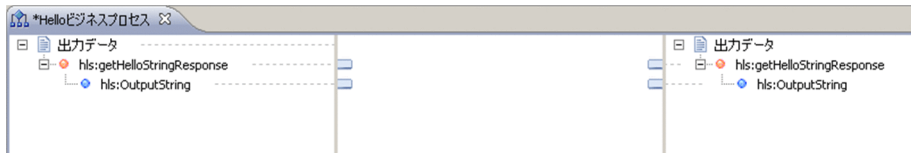
4. キャンパスのデータ変換アクティビティを右クリックして, [マッピング定義起動] を選択します。

[ルート要素選択] ダイアログが表示されます。

5. [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。



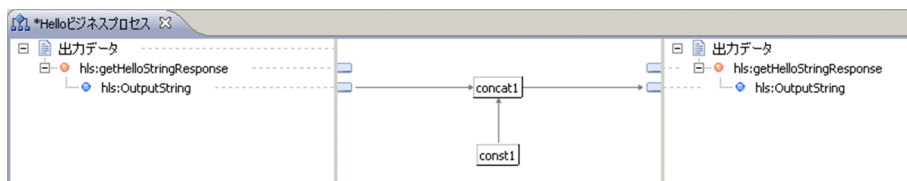
6. [OK] ボタンをクリックします。
データ変換定義画面が表示されます。



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に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に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. 操作 9. ~ 11. 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。

5. サンプルプログラムの開発を体験する

13. 操作 9. ～ 11. 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。

(d) 応答アクティビティ

1. キャンバ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	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。
ボディ割当変数	出力データ	応答電文のボディ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。
ヘッダ割当変数	設定なし	応答電文のヘッダに変数を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。
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。
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。

4. すべてのアクティビティを定義したら、メニューから [ファイル] - [保存] を選択し、ビジネスプロセスの定義を終了します。

5.5.2 Hello ビジネスプロセスの検証とパッケージング

作成したビジネスプロセスは、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。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、「5.4.2 サービスアダプタの検証とパッケージング」を参照してください。

5.5.3 Hello ビジネスプロセスの配備定義

パッケージングしたビジネスプロセスを配備定義します。配備定義の方法は、Hello サービスアダプタと同じです。配備定義の詳細については、「5.4.3 サービスアダプタの配備定義」を参照してください。

5.6 商品手配システムの開発

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。

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。

- 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

5.6.1 在庫管理サービスアダプタの定義

在庫管理サービス用 WSDL (`InventoryManagementService.wsdl`) を使用して在庫管理サービスアダプタを定義します。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 5-5 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	<code>InventoryManagementService.wsdl</code>	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 (<code>InventoryManagementService.wsdl</code>) を使用して作成します。
Port の指定	<code>InventoryManager</code>	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	<code>InvAdp</code>	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	<code>c4webcl.properties</code> (次の内容が設定されています。 「 <code>c4web.logger.log_file_prefix=InventoryManagementService</code> 」)	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、「 <code>c4webcl.properties</code> 」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、「 <code>InventoryManagementService</code> 」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。

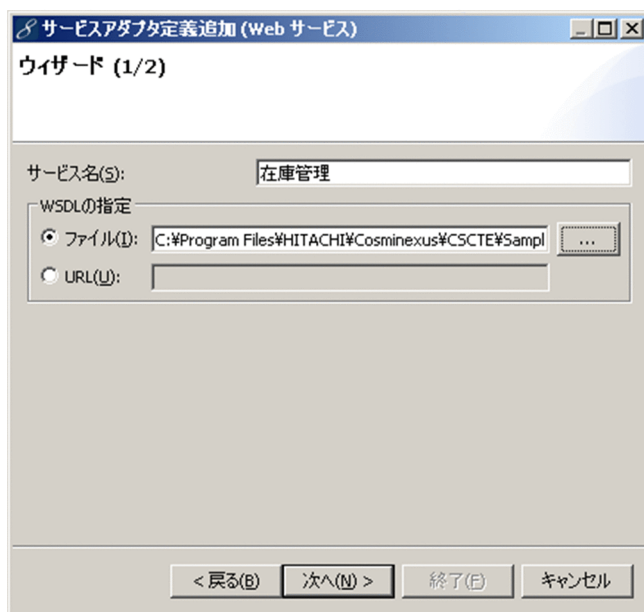
注

5. サンプルプログラムの開発を体験する

各ファイルの格納場所については、「付録 A.3 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。

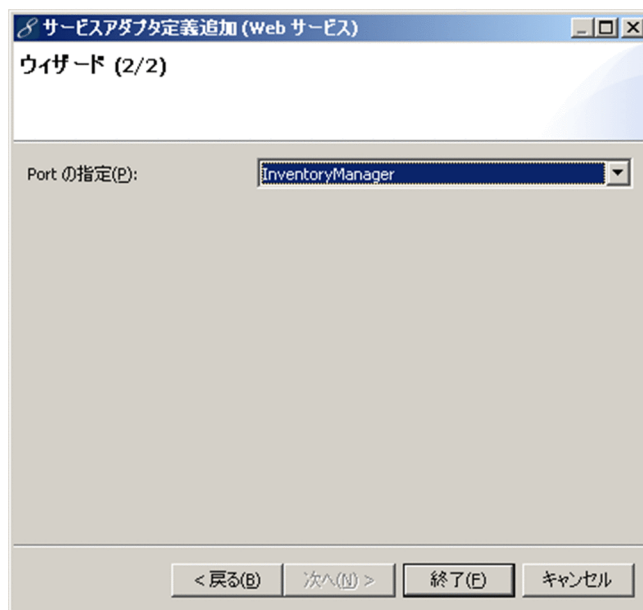
在庫管理サービスアダプタの追加と定義手順を次に示します。

1. ツリービューの [サービス定義一覧] を選択し、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. サンプルプログラムの開発を体験する

5. サービスアダプタ定義（基本）画面で、サービス ID を「InvAdp」に変更します。

■サービス部品制御情報
サービス名 在庫管理
サービス ID InvAdp
サービス種別 Web サービス
アドレス http://localhost/InventoryManager
最大インスタンス数 0
サービスクラス名 InventoryManager
オペレーション reserveItem
☐ システム例外をフォルトに変換する

■オペレーション情報
オペレーション名 reserveItem
通信モデル 同期

■要求電文
ボディ | ヘッダ |
標準
☐ 使う
フォーマット ID
電文フォーマット 参照...
表示... 出力...
サービス部品
フォーマット ID format1
電文フォーマット cscformat1.xsd 参照...
表示... 出力...
データ変換定義
【標準→サービス部品】 編集... 削除

■応答電文
ボディ | ヘッダ |
標準
☐ 使う
フォーマット ID
電文フォーマット 参照...
表示... 出力...
サービス部品
フォーマット ID format2
電文フォーマット cscformat1.xsd 参照...
表示... 出力...
データ変換定義
【サービス部品→標準】 編集... 削除

サービスアダプタ定義 (基本) サービスアダプタ定義 (詳細)

6. サービスアダプタ定義（詳細）画面で、[参照] 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。

<uCosminexus Service Architect のインストールディレクトリ

>¥CSCTE¥Samples¥ 商品手配

¥Service¥InventoryManagementService¥c4webcl.properties

7. メニューから [ファイル] - [保存] を選択します。

5.6.2 配送受付サービスアダプタの定義

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 5-6 配送受付サービスアダプタを定義するときの設定する値

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。
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。 「 c4web.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

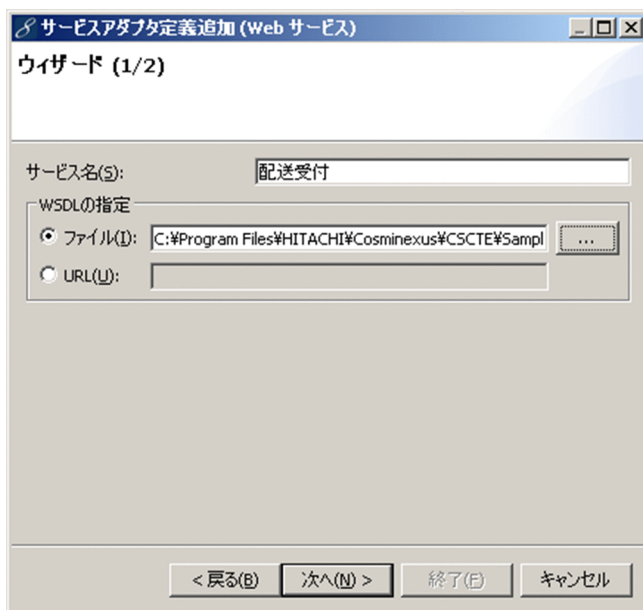
各ファイルの格納場所については、「付録 A.3 商品手配サンプルプログラムの構成」を参照してください。

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。

- ツリービューの [サービス定義一覧] を選択し、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。
- 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。

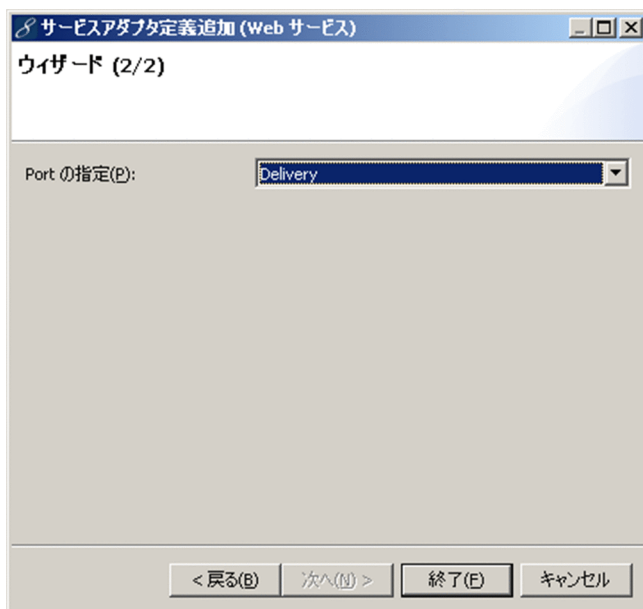
5. サンプルプログラムの開発を体験する

- 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。

- 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（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（詳細）画面で、[参照] 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。
 <uCosminexus Service Architect のインストールディレクトリ
 >¥CSCTE¥Samples¥ 商品手配 ¥Service¥DeliveryService¥c4webcl.properties
7. メニューから [ファイル] - [保存] を選択します。

5.6.3 商品手配ビジネスプロセスの定義

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呼び出します。
3. 在庫管理サービス呼び出した結果、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。

5. サンプルプログラムの開発を体験する

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。

(1) ビジネスプロセスの追加

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次の表に示します。

表 5-7 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] - [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	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] を選択し、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。

ビジネスプロセス定義追加

ウィザード

ビジネスプロセス名(B): 商品手配

ステータスの永続化: ☒ yes(Y) ☐ no(N)

BPEL ファイル

☐ インポートする(I)

ファイル名(L):

アクティビティの配置方法: ☒ 縦方向(V) ☐ 横方向(H)

終了(E) キャンセル

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. ツリービューで「商品手配」を選択します。
プロパティビューに商品手配のプロパティ一覧が表示されます。
5. プロパティビューで、サービス ID の値のセルをクリックします。
入力できる状態になります。
6. 「ArrBP」に変更し,[Enter] を押します。

プロパティ	値
サービス ID	ArrBP
ステータスの永続化	yes
バージョン	1
ビジネスプロセス	商品手配

5. サンプルプログラムの開発を体験する

- 変更してよいかどうかを確認するメッセージが表示されるので、[OK] をクリックします。

(2) ユーザ定義受付の追加

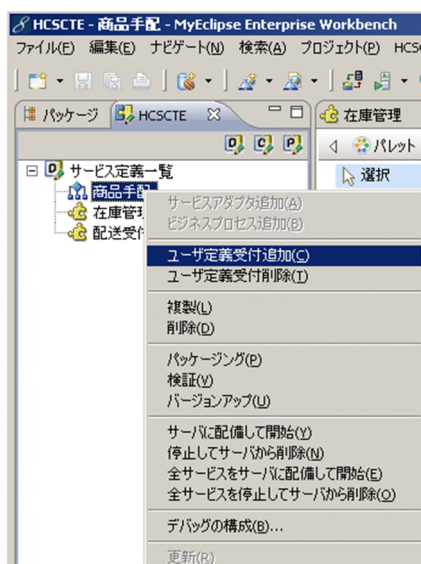
商品手配サンプルプログラムでは、サービス部品に合わせた受付を使用して、サービスリクエストからの要求を受け付けます。そのため、ユーザ定義受付が必要になります。ユーザ定義受付を追加するときに設定する値を次の表に示します。

表 5-8 ユーザ定義受付を追加するときに設定する値

項目名	設定する値	説明
受付種別	SOAP 受付	受付種別を選択します。
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。

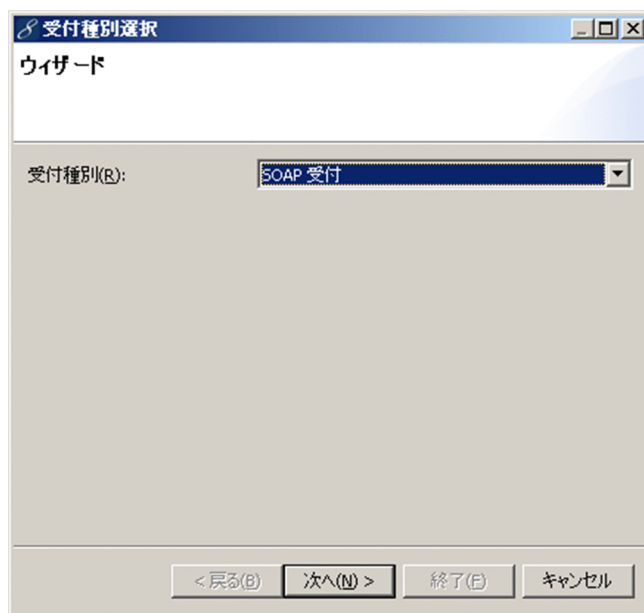
商品手配のユーザ定義受付の追加手順を次に示します。

- 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。

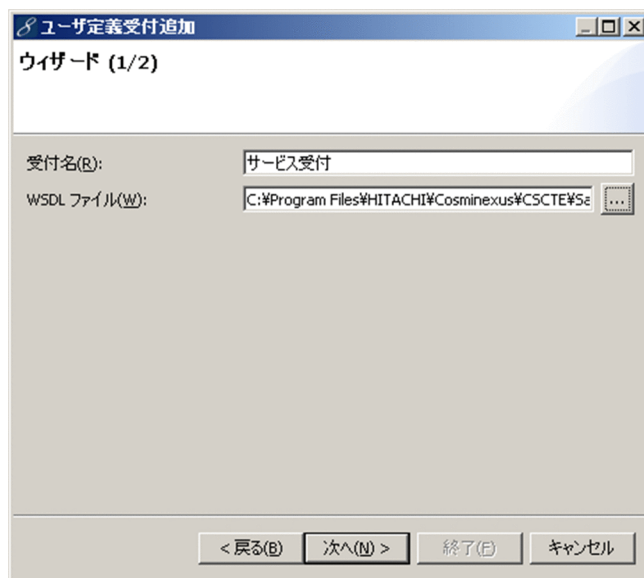


受付種別を選択するためのダイアログが表示されます。

- [受付種別] のドロップダウンリストから、[SOAP 受付] を選択します。



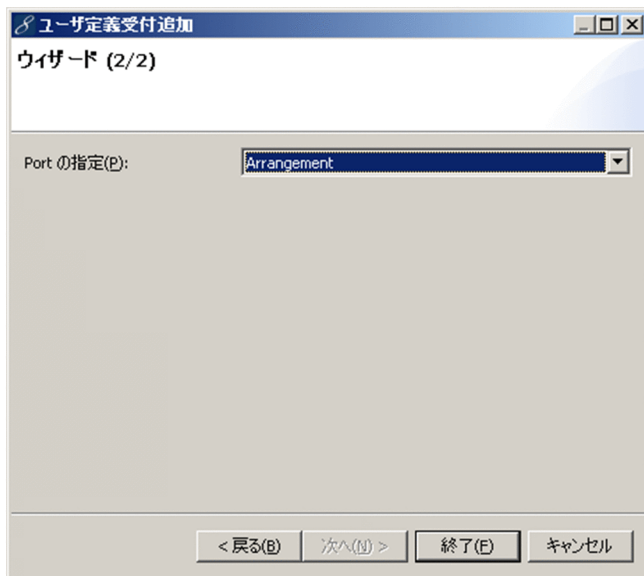
3. [次へ] ボタンをクリックします。
ユーザ定義受付を追加するためのダイアログが表示されます。
4. 受付名に「サービス受付」を入力し、WSDL ファイルに「ArrangementService.wsdl」を入力します。



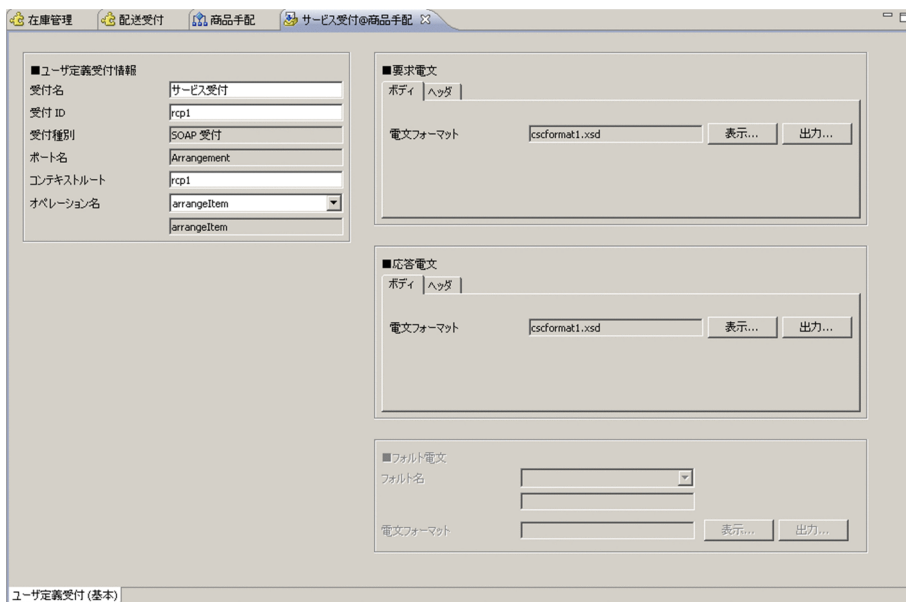
5. [次へ] ボタンをクリックします。

5. サンプルプログラムの開発を体験する

6. 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。



(3) 変数の設定

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 5-9 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	XML	入力データ .xsd
出力データ	XML	出力データ .xsd
在庫引当入力データ	XML	在庫引当入力データ .xsd
在庫引当出力データ	XML	在庫引当出力データ .xsd
配送手配入力データ	XML	配送手配入力データ .xsd
配送手配出力データ	XML	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「XML」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. サンプルプログラムの開発を体験する

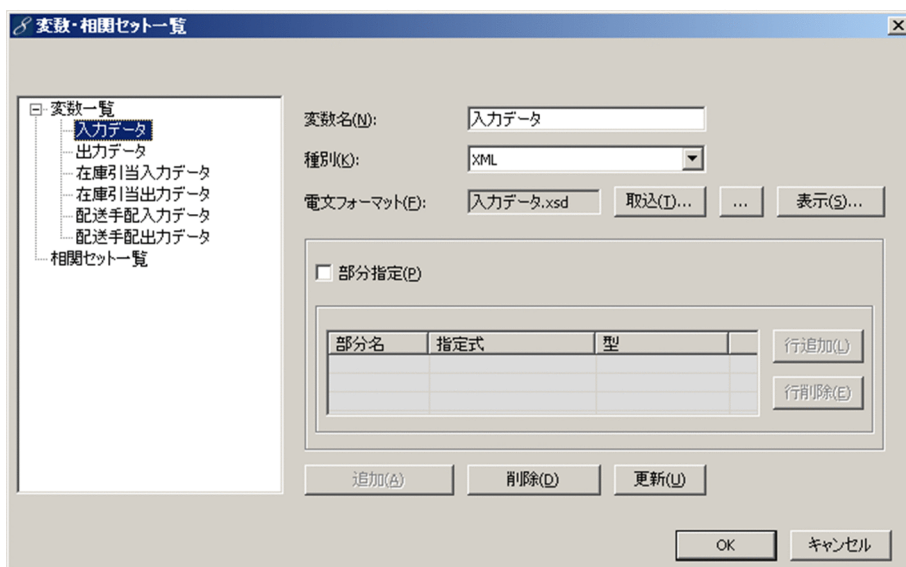
5. [オペレーション名] はドロップダウンリストから「arrangeItem」を,[電文種別] はドロップダウンリストから「要求電文 (ボディ)」を選択します。[電文フォーマット] に「入力データ」を入力します。

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・相関セッー覧] 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, 操作 2. ~ 7. と同様の操作で設定します。
設定値は次のとおりです。
[変数・相関セッー覧] ダイアログで設定する種別はすべて [XML] です。また,
[電文フォーマットの取込] ダイアログでの設定値は次のとおりです。

項目名	変数名				
	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	配送手配 出力データ
サービス / 受付	受付名	サービス名	サービス名	サービス名	サービス名
サービス / 受付の名称	サービス受付	在庫管理	在庫管理	配送受付	配送受付
オペレーション名	arrangeItem	reserveItem	reserveItem	deliverItem	deliverItem

項目名	変数名				
	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	配送手配 出力データ
電文種別	応答電文 (ボディ)	要求電文 (ボディ)	応答電文 (ボ ディ)	要求電文 (ボ ディ)	応答電文 (ボ ディ)
電文フォー マット	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	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。



(4) アクティビティの配置

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。

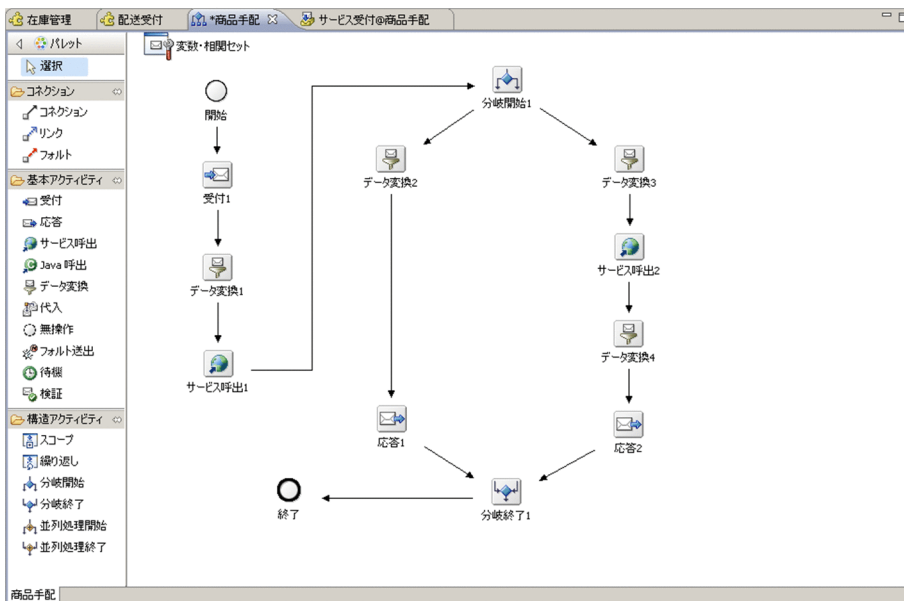
表 5-10 商品手配サンプルプログラムのビジネスプロセスに必要なアクティビティ

設定する値	説明
受付アクティビティ	サービスリクエスト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トへ返します。
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。

5. サンプルプログラムの開発を体験する

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に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操作 2. ~ 4. と同様の操作で一つずつ連結します。
次のようなフローになるよう、アクティビティを連結してください。



(5) アクティビティの定義

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。

(a) 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。
ボディ割当変数	入力データ	ビジネスプロセスの要求電文のボディ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。
ヘッダ割当変数	設定なし	ビジネスプロセスの要求電文のヘッダに変数を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。
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。

5. サンプルプログラムの開発を体験する

項目名	設定する値	説明
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。

(b) データ変換アクティビティ（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。

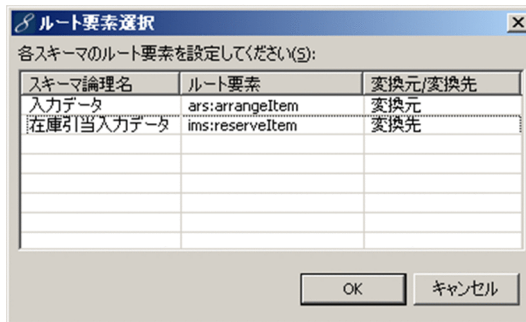
[データ変換アクティビティ] ダイアログが表示されます。

2. 次の内容を入力します。

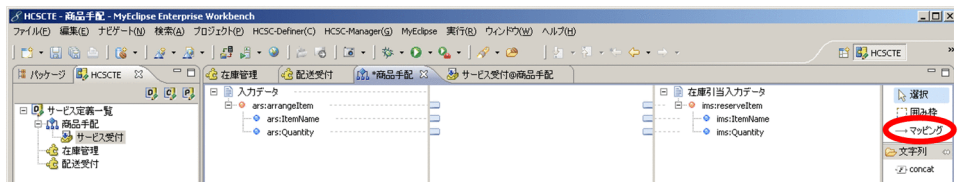
項目名	設定する値	説明
アクティビティ名	在庫引当処理	アクティビティの名称を入力します。
変数（変換元変数）	入力データ	データの変換元になる変数をドロップダウンリストから選択し、[追加] ボタンをクリックします。
変数（変換先変数）	在庫引当入力データ	データの変換先になる変数をドロップダウンリストから選択します。
データ変換定義	在庫引当処理	変数の変換に使うデータ変換定義ファイルの名称を入力します。

3. [OK] ボタンをクリックします。

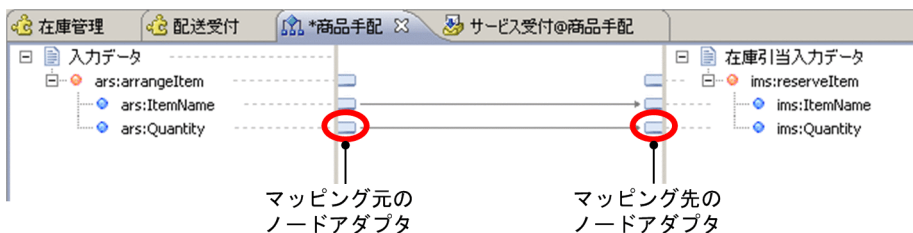
4. キャンパスのデータ変換アクティビティを右クリックして、[マッピング定義起動] を選択します。
[ルート要素選択] ダイアログが表示されます。
5. 変換元のスキーマ論理名「入力データ」のルート要素をクリックして、ドロップダウンリストから「ars:arrangeItem」を選択します。
6. 変換先のスキーマ論理名「在庫引当入力データ」のルート要素をクリックして、ドロップダウンリストから「ims:reserveItem」を選択します。



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。マッピング元のノードアダプタとマッピング先のノードアダプタとの対応は次のとおりです。



5. サンプルプログラムの開発を体験する

(c) データ変換アクティビティ（在庫なしの場合）

1. 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。

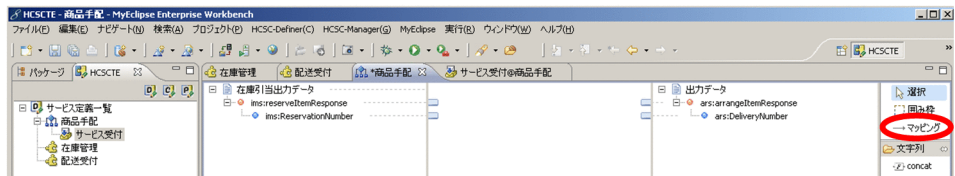
項目名	設定する値	説明
アクティビティ名	在庫なし設定	アクティビティの名称を入力します。
変数（変換元変数）	在庫引当出力データ	データの変換元になる変数をドロップダウンリストから選択し、[追加] ボタンをクリックします。
変数（変換先変数）	出力データ	データの変換先になる変数をドロップダウンリストから選択します。
データ変換定義	在庫なし設定	変数の変換に使うデータ変換定義ファイルの名称を入力します。

3. [OK] ボタンをクリックします。
4. キャンバ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。
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。

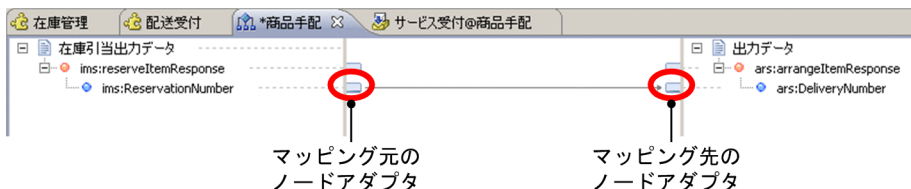
6. 変換先のスキーマ論理名「出力データ」のルート要素をクリックして、ドロップダウンリストから「ars:arrangeItemResponse」を選択します。



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。マッピング元のノードアダプタとマッピング先のノードアダプタとの対応は次のとおりです。



(d) 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。

5. サンプルプログラムの開発を体験する

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。

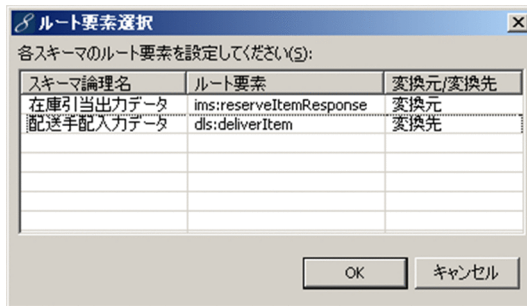
3. [OK] ボタンをクリックします。

4. キャンバ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。

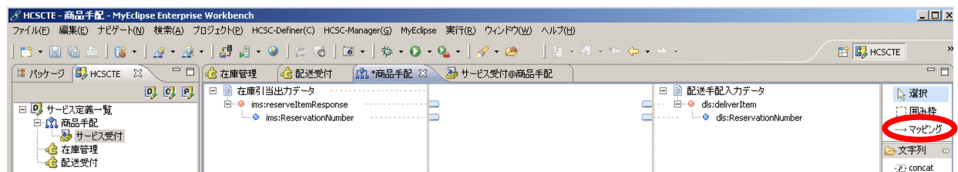
[ルート要素選択] ダイアログが表示されます。

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。

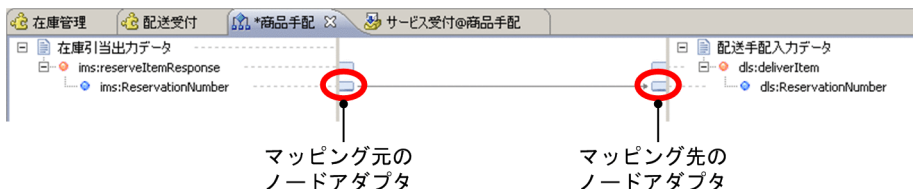
6. 変換先のスキーマ論理名「配送手配入力データ」のルート要素をクリックして、ドロップダウンリストから「dls:deliverItem」を選択します。



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。マッピング元のノードアダプタとマッピング先のノードアダプタとの対応は次のとおりです。



(e) 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。

5. サンプルプログラムの開発を体験する

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。

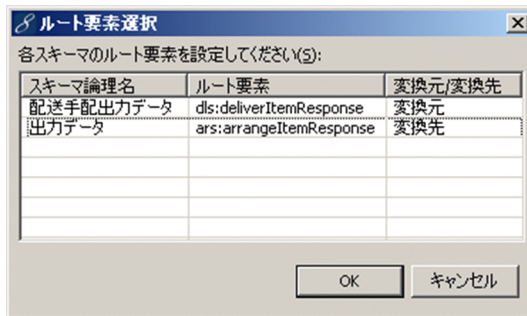
3. [OK] ボタンをクリックします。

4. キャンバ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。

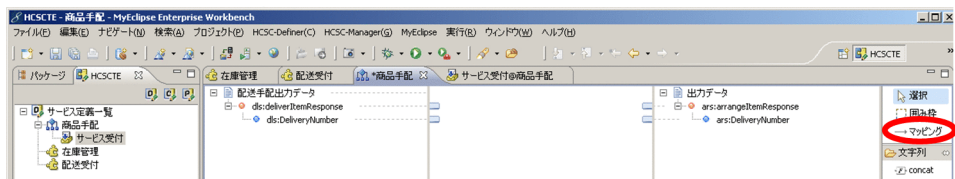
[ルート要素選択] ダイアログが表示されます。

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。

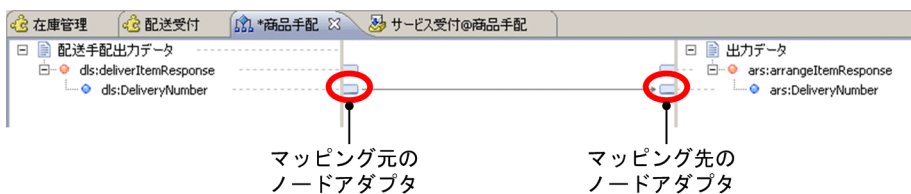
- 変換先のスキーマ論理名「出力データ」のルート要素をクリックして、ドロップダウンリストから「ars:arrangeItemResponse」を選択します。



- [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。
- 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。



- 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。マッピング元のノードアダプタとマッピング先のノードアダプタとの対応は次のとおりです。



(f) サービス呼出アクティビティ（在庫を引き当てる場合）

- 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。

5. サンプルプログラムの開発を体験する

2. 次の内容を入力します。

サービス呼出アクティビティ

アクティビティ名(M): 在庫引当

サービス名(S): 在庫管理

オペレーション名(O): reserveItem

通信モデル: 同期

要求電文

ボディ割当変数(Q): 在庫引当入力データ 編集(E)...

ヘッダ割当変数: 設定(H)...

応答電文

ボディ割当変数(R): 在庫引当出力データ 編集(D)...

ヘッダ割当変数: 設定(D)...

割当関連セット群: 設定(I)...

OK キャンセル

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。
ボディ割当変数（要求電文）	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のボディ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。
ヘッダ割当変数（要求電文）	設定なし	在庫管理サービスを呼び出す要求電文のヘッダに変数を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。
ボディ割当変数（応答電文）	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のボディ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。
ヘッダ割当変数（応答電文）	設定なし	同期オペレーションから受け取る応答電文のヘッダに変数を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
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。

(g) サービス呼出アクティビティ（配送を手配する場合）

- 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。
- 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。
ボディ割当変数（要求電文）	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のボディ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。

5. サンプルプログラムの開発を体験する

項目名	設定する値	説明
ヘッダ割当変数（要求電文）	設定なし	配送手配サービスを呼び出す要求電文のヘッダに変数を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。
ボディ割当変数（応答電文）	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のボディ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。
ヘッダ割当変数（応答電文）	設定なし	同期オペレーションから受け取る応答電文のヘッダに変数を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。
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。

(h) 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。
ボディ割当変数	出力データ	ビジネスプロセスの応答電文のボディ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。
ヘッダ割当変数	設定なし	ビジネスプロセスの応答電文のヘッダに変数を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
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。

(i) 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。
ボディ割当変数	出力データ	ビジネスプロセスの応答電文のボディ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。
ヘッダ割当変数	設定なし	ビジネスプロセスの応答電文のヘッダに変数を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。
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。

5. サンプルプログラムの開発を体験する

項目名	設定する値	説明
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。

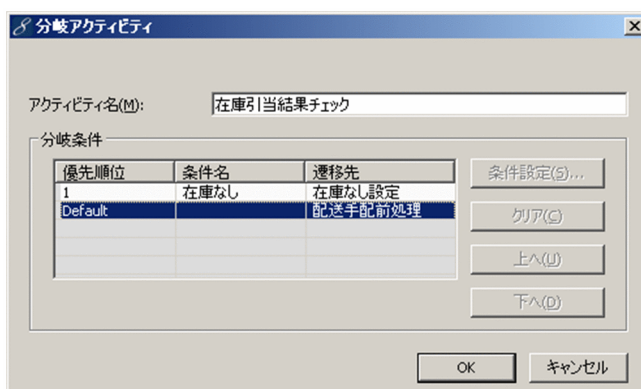
(j) 分岐開始アクティビティ

- 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。
[分岐アクティビティ] ダイアログが表示されます。
- 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。
- [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、いちばん上の行に移動します。
- いちばん上の行を選択し、[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。
- 次の内容を入力します。

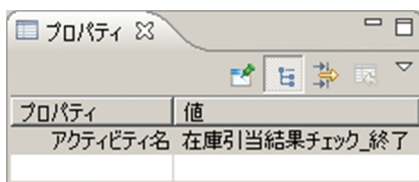
項目名	設定する値	説明
条件名	在庫なし	在庫引当チェックの結果、在庫がない場合の条件を指定します。
変数	在庫引当出力データ	条件式で使用する変数をドロップダウンリストから選択します。

項目名	設定する値	説明
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] ダイアログに戻ります。
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。
- (k) 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。



3. すべてのアクティビティを定義したら、メニューから [ファイル] - [保存] を選択して、ビジネスプロセスの定義を終了します。

5.6.4 コンポーネントの検証とパッケージング

作成したコンポーネント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。ツリービューの [商品手配] ビジネスプロセスに対して、検証・パッケージングしてください。

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。手順の詳細は、「5.4.2 サービスアダプタの検証とパッケージング」を参照してください。

5.6.5 コンポーネントの配備定義

パッケージングしたコンポーネントを配備定義します。配備定義の方法は、Hello サービスアダプタと同じです。配備定義の詳細については、「5.4.3 サービスアダプタの配備定義」を参照してください。

5.7 商品手配システムのデバッグ

「5.6.3 商品手配ビジネスプロセスの定義」で定義したビジネスプロセスをデバッグして、ビジネスプロセスの処理の流れを確認してみましょう。

この節で説明するビジネスプロセスのデバッグの流れを次に示します。

1. アクティビティにブレークポイントを設定します。
2. サービスリクエストをデプロイします。
3. HCSC サーバを起動します。
4. デバッグを起動します。
5. サービスリクエストからリクエストを送信し、プロセスインスタンスの処理を開始します。
6. アクティビティごとにデバッグを実行します。
7. サービスを呼び出さないでデバッグする場合、サービスをエミュレーションします。
8. ビジネスプロセスのデバッグを終了します。

(1) ブレークポイントの設定

プロセスインスタンスの処理を中断するアクティビティにブレークポイントを設定します。

ここでは、サービス呼出アクティビティ（在庫引当）にブレークポイントを設定します。

ブレークポイントを設定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でサービス呼出アクティビティ（在庫引当）を右クリックし、[ブレークポイントの追加] を選択します。
アクティビティにブレークポイントが追加されます。次の図のように、アクティビティの横にブレークポイントを表すチェックが付きます。



在庫引当

(2) サービスリクエストのデプロイ

リクエストを送信するために、商品手配サンプルプログラムの MyEclipse プロジェクトをインポートしたあと、サービスリクエストを J2EE サーバにデプロイします。

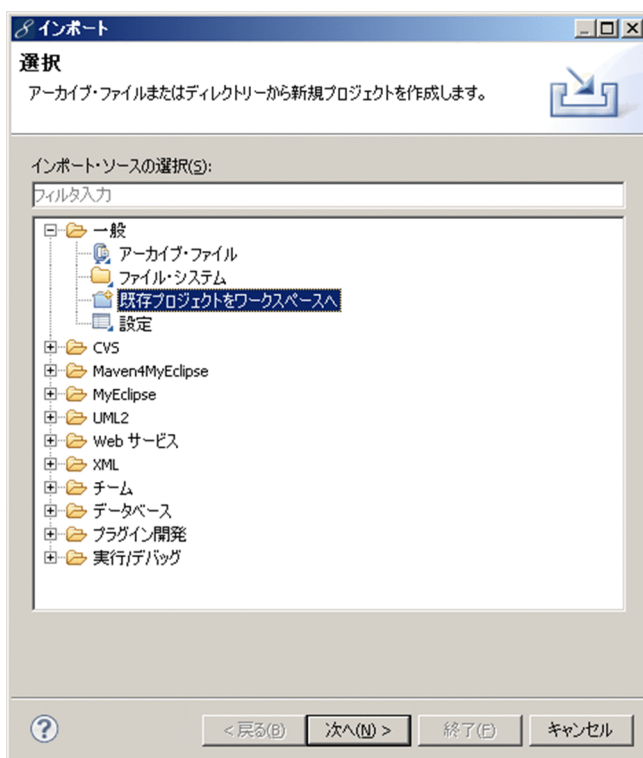
サービスリクエストをデプロイする手順を次に示します。

1. 次のコマンドを入力して HCSC サーバを停止します。

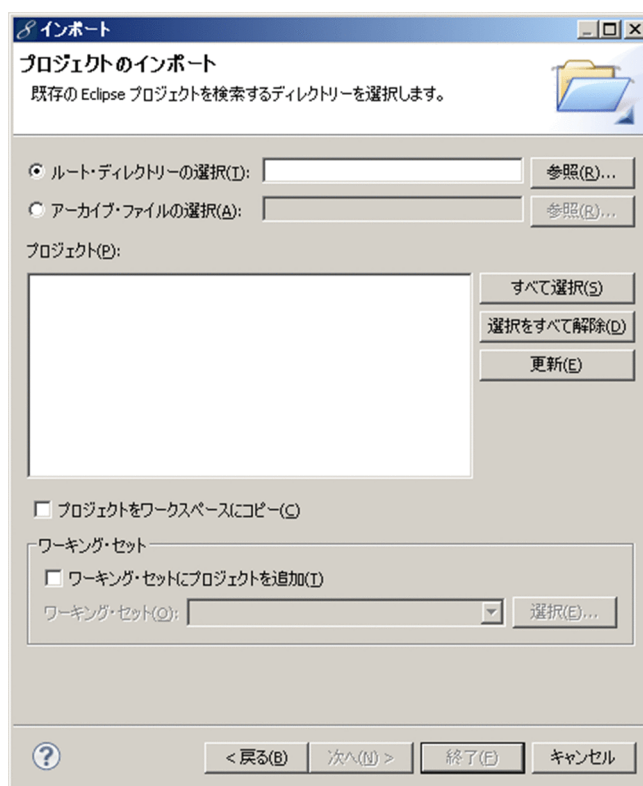
```
cscsvstop
```

5. サンプルプログラムの開発を体験する

2. MyEclipse のメニューから [ウィンドウ] - [パースペクティブを開く] - [その他] を選択します。
[パースペクティブを開く] ダイアログが表示されます。
3. [MyEclipse Java エンタープライズ (デフォルト)] を選択し, [OK] ボタンをクリックします。
[MyEclipse Java エンタープライズ] パースペクティブが表示されます。
4. メニューから [ファイル] - [インポート] を選択します。
選択画面が表示されます。
5. ツリービューから [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。

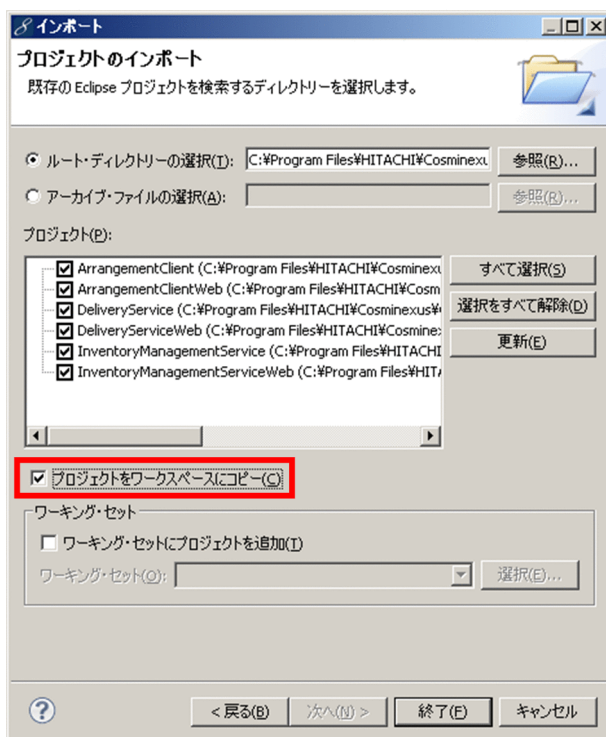


6. [次へ] ボタンをクリックします。
プロジェクトのインポート画面が表示されます。



7. [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ます。
ファイルの参照画面が表示されます。
8. サンプルプログラムが格納されているディレクトリ (<uCosminexus Service Architect
のインストールディレクトリ >¥CSCTE¥Samples¥ 商品手配) を選択し , [OK] ボタ
ンをクリックします。
プロジェクトのインポート画面に戻ります。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ください。
9. [プロジェクトをワークスペースにコピー] のチェックが入っていない場合は ,
チェックを入れます。
チェックを入れないと , サンプルプログラムが削除されるおそれがあります。必ず
チェックを入れてください。

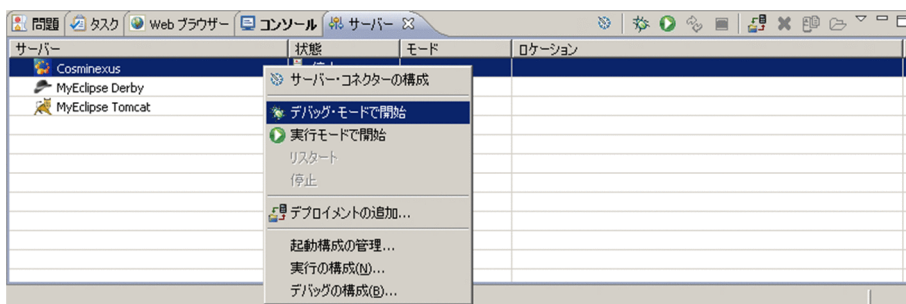
5. サンプルプログラムの開発を体験する



10 [終了] ボタンをクリックします。

パースペクティブに選択したプロジェクトがインポートされます。

11 [サーバー] ビューの [Cosminexus] サーバを右クリックし , [デバッグ・モードで開始] を選択します。

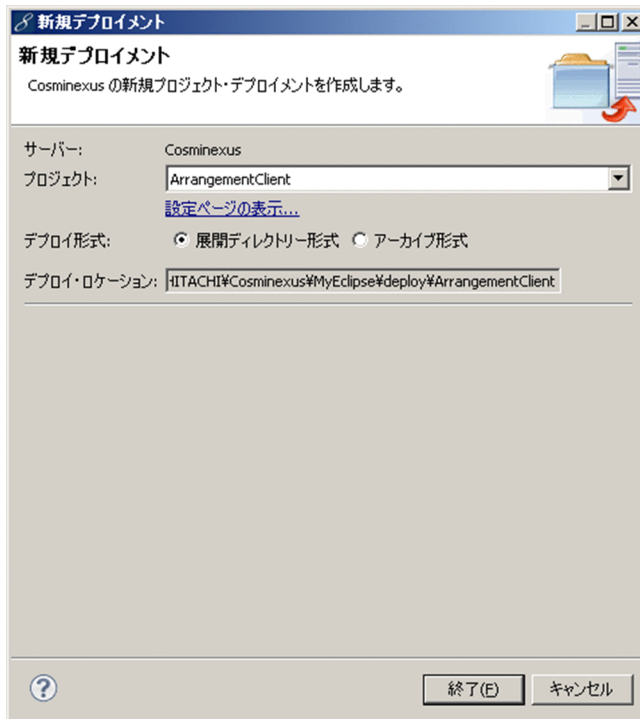


処理中であることを知らせるメッセージが表示されたあと , Cosminexus サーバが起動されます。Cosminexus サーバが表示されていない場合は , 「3.5.4(2) J2EE サーバの選択」を参照してください。

12 [サーバー] ビューの [Cosminexus] サーバを右クリックし , [デプロイメントの追加] を選択します。

[新規デプロイメント] ダイアログが表示されます。

13. プロジェクトのドロップダウンリストから [ArrangementClient] を選択し、デプロイ形式の [展開ディレクトリー形式] を選択します。



14. [終了] ボタンをクリックします。

処理中であることを知らせるメッセージが表示されます。処理終了後、[MyEclipse Java エンタープライズ] パースペクティブに戻ります。

15. 「DeliveryService」, 「InventoryManagementService」についても、操作 12. ~ 14. と同様の操作で設定します。

設定が終わったら、[サーバー] ビューの [Cosminexus] の配下に

[ArrangementClient], [DeliveryService], および

[InventoryManagementService] プロジェクトが存在することを確認してください。

(3) HCSC サーバの起動

デバッガを起動する前に、HCSC サーバを起動します。

1. 次のコマンドを入力して HCSC サーバを起動します。

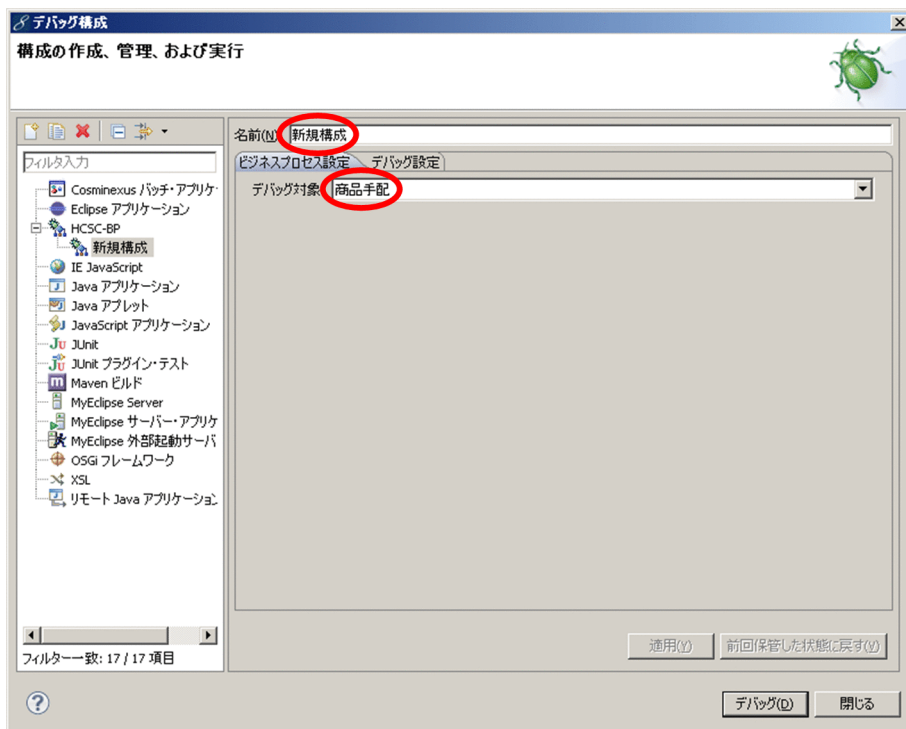
```
cscsvstart
```

5. サンプルプログラムの開発を体験する

(4) デバッガの起動

デバッガを起動する手順を次に示します。

1. MyEclipse のメニューから [実行] - [デバッグの構成] を選択します。
[デバッグ構成] ダイアログが表示されます。
2. [デバッグ構成] ダイアログのメニューから [HCSC-BP] を右クリックし, [新規] を選択します。
3. [名前] に任意の名称を入力し, [デバッグ対象] は, ドロップダウンリストから「商品手配」を選択します。



4. [デバッグ] ボタンをクリックします。
アカウント認証画面が表示されます。
5. [ユーザ ID] に「admin」, [パスワード] に「admin」を入力して, [OK] ボタンをクリックします。
処理中であることを知らせるメッセージが表示されます。2 回目以降にデバッガを起動するときは, アカウント認証画面は表示されないで, すぐに処理が開始されます。
6. [OK] ボタンをクリックします。
デバッガが起動されます。

(5) リクエストの送信

サービスリクエストからリクエストを送信して、プロセスインスタンスの処理を開始します。

リクエストを送信する手順を次に示します。

1. ブラウザから次に示す URL を入力します。
<http://localhost/ArrangementClientWeb/index.html>
 商品手配サンプルプログラムの実行画面が起動されます。
2. 商品名と個数を選択します。



3. [手配] ボタンをクリックします。
 パースペクティブを切り替えるかどうかを確認するダイアログが表示されます。
4. [はい] ボタンをクリックします。
 デバッグ情報や変数を表示するビューが開き、プロセスインスタンスの処理が開始されます。
 サービス呼出アクティビティ（在庫引当）でプロセスインスタンスの処理が中断されます。ビジネスプロセス定義画面のキャンパスのアクティビティに、次のように中断していることを表す色が付きます。



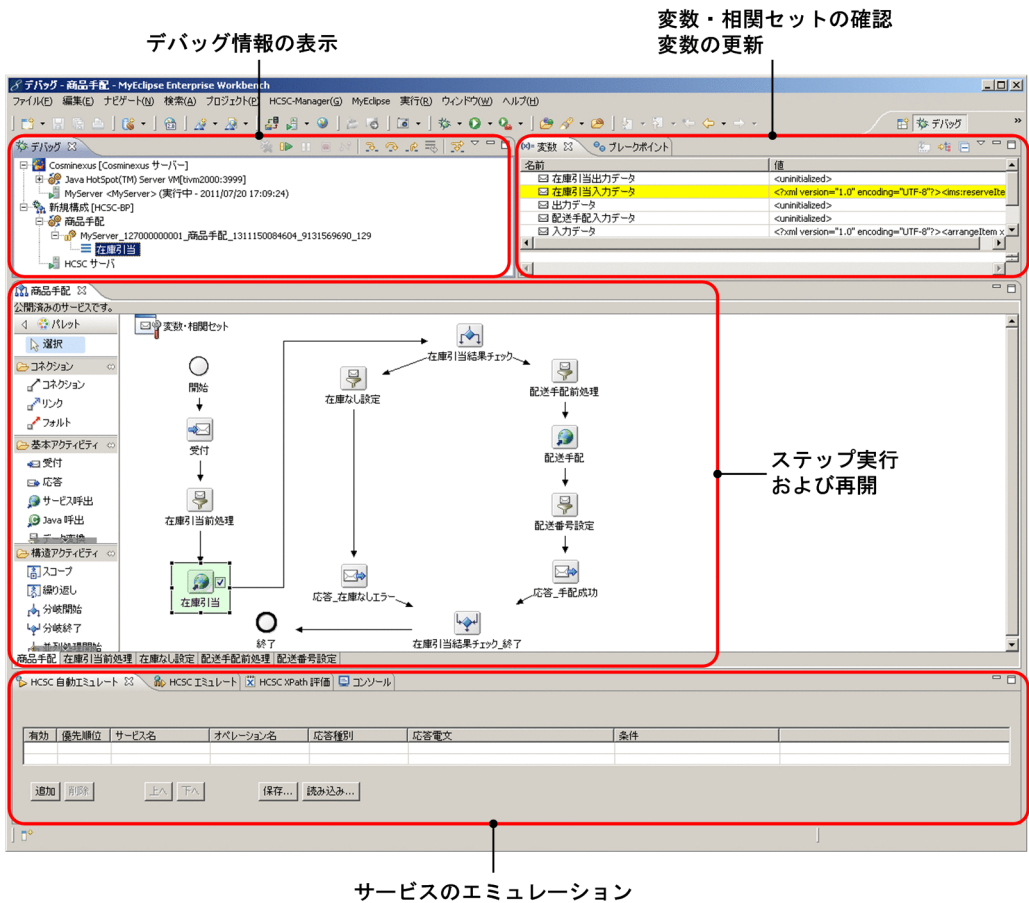
(6) ビジネスプロセスのデバッグの実行

プロセスインスタンスの処理が中断している状態では、次に示す画面でデバッグを実行

5. サンプルプログラムの開発を体験する

できます。





図 5-2 ビジネスプロセスのデバッグ画面



ビジネスプロセスのデバッグで実行できる項目を次の表に示します。

表 5-11 ビジネスプロセスのデバッグで実行できる項目

項目	説明
変数・相関セットの確認	ビジネスプロセスで現在使用している変数，および相関セットを確認できます。
変数の更新	ビジネスプロセスで現在使用している変数の値を更新できます。

項目	説明
ステップ実行および再開	<p>[デバッグ情報の表示] ビューでは次の操作を実行できます。これらの操作によって、アクティビティごとにデバッグを実行してください。</p> <ul style="list-style-type: none"> • ステップイン () 次のアクティビティに進みます。分岐開始アクティビティで実行した場合は、分岐先で最初に処理されるアクティビティに進みます。 • ステップオーバー () 分岐開始アクティビティで実行した場合は、分岐終了アクティビティまでをまとめて処理します。分岐開始アクティビティ以外で実行した場合は、ステップインと同じ動作になります。 • ステップリターン () 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の内部のアクティビティで実行した場合は、所属するアクティビティの次のアクティビティまで処理します。 • 再開 () 次にブレークポイントを設定しているアクティビティまで、プロセスインスタンスの処理を再開します。
サービスのエミュレーション	<p>作成済みの応答電文を利用することで、サービスをエミュレーションできます。サービスをエミュレーションする手順については、「(7) サービスのエミュレーション」を参照してください。</p>

(7) サービスのエミュレーション

サービスを呼び出さないでデバッグする場合、サービスのエミュレーションを設定します。サービスのエミュレーションを設定すると、作成済みの応答電文を実際のサービスを呼び出す代わりに利用できます。

サービスのエミュレーションを利用することで、ビジネスプロセスから呼び出すサービスがなくても、ビジネスプロセスの処理を進められます。

サービスのエミュレーションは、プロセスインスタンスの処理が中断しているときに設定できます。ここでは、サービス呼出アクティビティ（在庫引当）でプロセスインスタンスの処理が中断しているものとして、「在庫引当」サービスをエミュレーションする手順について説明します。

「在庫引当」サービスをエミュレーションする手順を次に示します。

1. XML ファイルを作成します。
2. HCSC エミュレートビューを表示します。
3. 作成した XML ファイルを利用してサービスのエミュレーションを実行します。

サービスのエミュレーションを実行する前に、デバッグを起動して、サービスリクエストからリクエストを送信しておく必要があります。

5. サンプルプログラムの開発を体験する

(a) XML ファイルの作成

サービスの応答として利用する XML ファイルを作成します。

XML ファイルを作成する方法を次に示します。

1. MyEclipse のメニューから、[ファイル] - [新規] - [その他] を選択し、表示されるダイアログで [一般] - [ファイル] を選択します。
[新規ファイル] ダイアログが表示されます。
2. [新規ファイル] ダイアログで XML ファイルを保管するディレクトリを選択します。
3. [ファイル名] に任意のファイル名 (拡張子: .xml) を入力します。
4. [終了] ボタンをクリックします。
XML エディタが表示されます。
5. XML エディタの [ソース] タブをクリックし、次のように入力します (は半角の空白です)

```
<?xml version="1.0" encoding="UTF-8"?><reserveItemResponse  
xmlns="http://sample/InventoryManagementService"  
><ReservationNumber>R00000001</ReservationNumber></reserveItemResponse>
```

6. メニューから [ファイル] - [保存] を選択します。
XML ファイルが保存されます。

(b) HCSC エミュレートビューの表示

サービスのエミュレーションは HCSC エミュレートビューで実行できます。

HCSC エミュレートビューを表示する手順を次に示します。

1. MyEclipse 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. [デバッグ] - [HCSC エミュレート] を選択し、[OK] ボタンをクリックします。
HCSC エミュレートビューが表示されます。

(c) サービスのエミュレーションの実行

1. [デバッグ情報の表示] ビューで在庫引当 (サービス呼出アクティビティ) を選択します。
2. HCSC エミュレートビューの [追加] ボタンをクリックします。
HCSC エミュレートビューのテーブルに行が追加されます。
3. 応答種別のセルを選択し、通常応答を選択します。
4. 応答電文のセルを選択し、[...] ボタンをクリックします。

[ファイル選択] ダイアログが表示されます。

5. 作成した XML ファイルを選択し, [OK] ボタンをクリックします。

6. [ステップ実行] ボタンをクリックします。

サービス呼出アクティビティ (在庫引当) からの応答がエミュレーションされます。
プロセスインスタンスの処理が分岐開始アクティビティ (在庫引当結果チェック) に進みます。

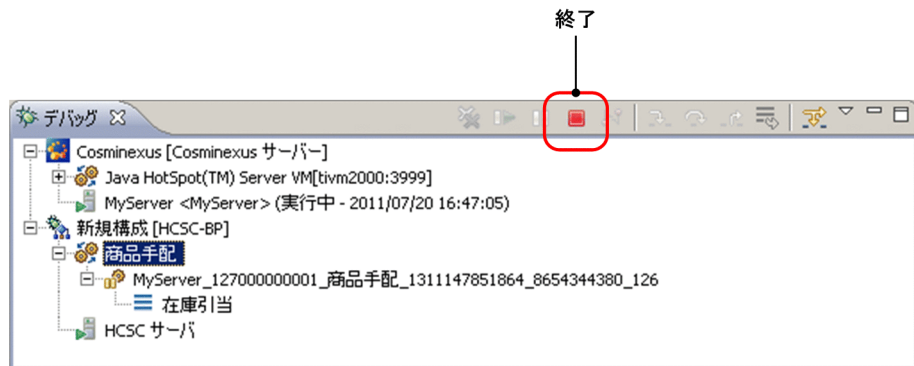
以降のアクティビティへ進むには, [デバッグ情報の表示] ビューを操作してください。
詳細は, 「(6) ビジネスプロセスのデバッグの実行」を参照してください。

(8) ビジネスプロセスのデバッグの終了

ビジネスプロセスの処理の流れを確認したら, ビジネスプロセスのデバッグを終了します。

ビジネスプロセスのデバッグを終了する手順を次に示します。

1. [デバッグ情報の表示] ビューで「商品手配」を選択し, [終了] アイコンをクリックします。



ビジネスプロセスのデバッグが終了します。

5.8 開発情報のエクスポートとインポート

作成したサンプルプログラムを実行環境で動作させるには、開発環境の情報を実行環境と共有する必要があります。

ここでは、開発情報のエクスポートとインポートの手順を説明します。

(1) エクスポート

開発情報を実行環境に渡すため、開発環境のリポジトリ情報を ZIP 形式のファイルとしてエクスポートし、指定したディレクトリに保存します。

手順を次に示します。

1. MyEclipse のメニューから [ウィンドウ] - [パースペクティブを開く] - [その他] を選択します。
[パースペクティブを開く] ダイアログが表示されます。
2. [HCSCTE] を選択し、[OK] ボタンをクリックします。
[HCSCTE] パースペクティブが表示されます。
3. MyEclipse 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。
4. 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。
5. [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。

(2) インポート

エクスポートした開発環境のリポジトリ情報を運用環境のリポジトリにインポートします。インポートには `cscreptcl` コマンドを使用します。

`cscreptcl` コマンドを実行する前に、運用環境のリポジトリとして設定したディレクトリ以下のディレクトリおよびファイルを参照していないことを確認してください。

`cscreptcl` コマンドは次のように実行します。

```
cscreptcl -user admin -pass admin -import <エクスポートしたZIPファイルのファイル名>
```

なお、このあとは、「3.5.5 MyEclipse プロジェクトのインポート」、「3.5.6 Web プロジェクトのデプロイ」、および「3.5.7 HCSC サーバへの定義内容の配備」の手順 7 以

降の手順に従って操作してください。

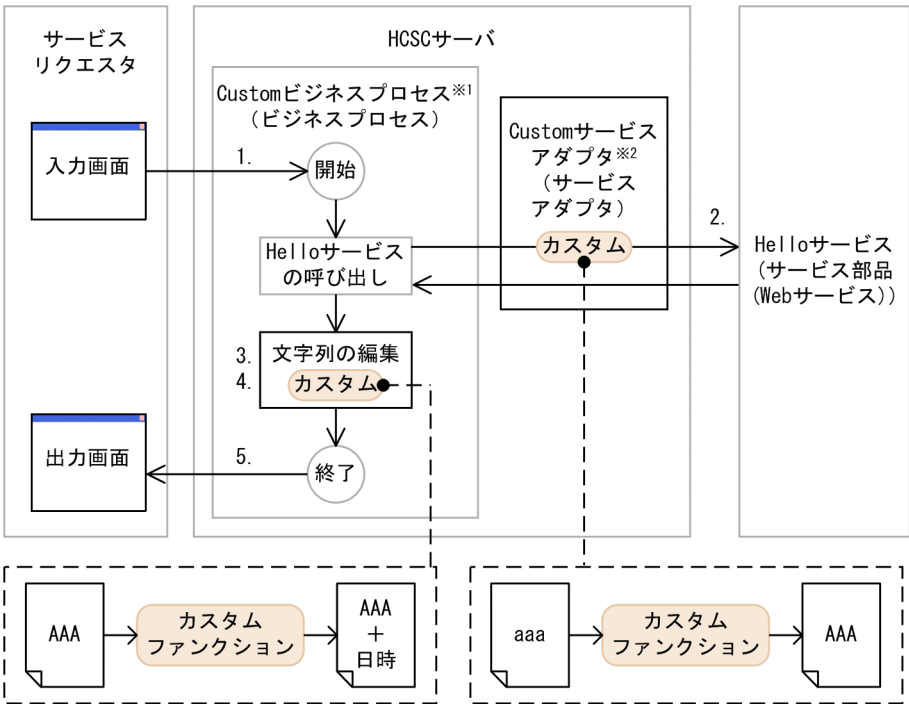
5.9 任意の Java プログラムを使用したデータ変換の定義

CustomFunction サンプルプログラムを使って、任意の Java プログラムを使用するプログラムを開発してみましょう。

CustomFunction サンプルプログラムの開発では、カスタムファンクションを使用します。カスタムファンクションとは、データの加工処理が定義されている任意の Java プログラムを呼び出して、データ変換を定義する機能です。

CustomFunction サンプルプログラムの処理の内容を次の図に示します。

図 5-3 CustomFunction サンプルプログラムの処理内容



(凡例)

→ : サービス部品を呼び出す要求・応答の流れ

カスタム : カスタムファンクション

[] : カスタムファンクションでの処理

注※1 Helloビジネスプロセスの一部を変更して作成します。

注※2 Helloサービスアダプタの一部を変更して作成します。

CustomFunction サンプルプログラムでは、次のようなプロセスで処理します。

1. Custom 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. Custom サービスアダプタを介して Hello サービスを呼び出します。このとき、カスタムファンクションを使用して、文字列を大文字に変換します。
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。
4. さらに、カスタムファンクションを使用して、文字列の末尾に実行時の日時 (yyyy/MM/dd HH:mm:ss.SSS) を連結します。

出力文字列: <Helloサービスの出力文字列 (大文字)> and Business Process yyyy/MM/dd HH:mm:ss.SSS

5. 文字を連結した結果をサービスリクエストに返し、出力画面に表示します。

5.9.1 CustomFunction サンプルプログラムの定義の流れ

CustomFunction サンプルプログラムを定義する流れを説明します。

CustomFunction サンプルプログラムは、Hello サービスアダプタおよび Hello ビジネスプロセスの一部を変更して作成します。したがって、Hello サービスアダプタおよび Hello ビジネスプロセスを作成したあとに、CustomFunction サンプルプログラムを作成してください。Hello サービスアダプタおよび Hello ビジネスプロセスの作成方法については、「5.4 サービスアダプタの定義」および「5.5 ビジネスプロセスの定義」を参照してください。

CustomFunction サンプルプログラムは、次の流れで定義します。

1. カスタムファンクションの準備
2. Hello サービスアダプタの変更
3. Hello ビジネスプロセスの変更

CustomFunction サンプルプログラムを開発する手順について、以降の項で説明します。

5.9.2 カスタムファンクションの準備

カスタムファンクションを使用する場合、最初にカスタムファンクションで使用する次の二つを作成します。

- 呼び出す Java プログラムの構成を定義した変換ファンクション定義ファイル
- Java プログラム

それぞれを作成する手順を次に示します。

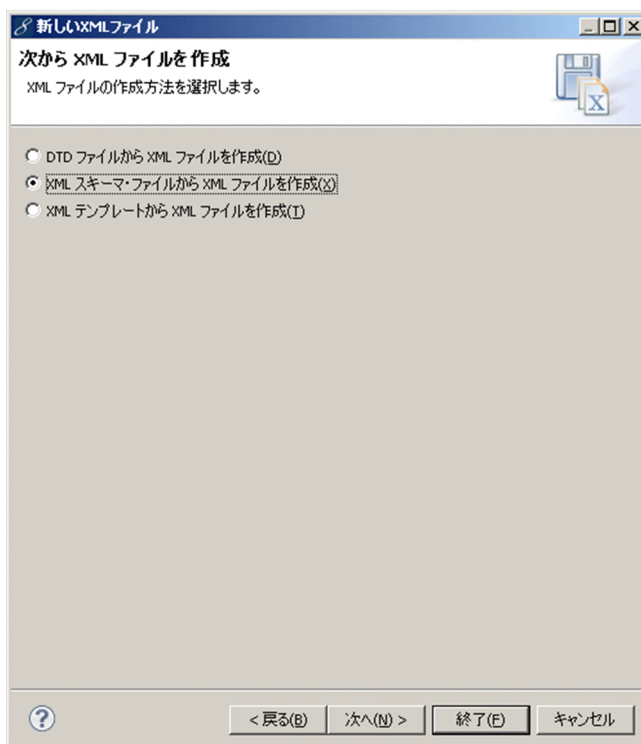
5. サンプルプログラムの開発を体験する

(1) 変換ファンクション定義ファイルの作成

1. MyEclipse のメニューから ,[ファイル] - [新規] - [その他] を選択します。
[新規] ダイアログが表示されます。
2. [MyEclipse] - [XML] - [XML (基本テンプレート)] を選択し ,[次へ] ボタンをクリックします。
[XML] ページが表示されます。

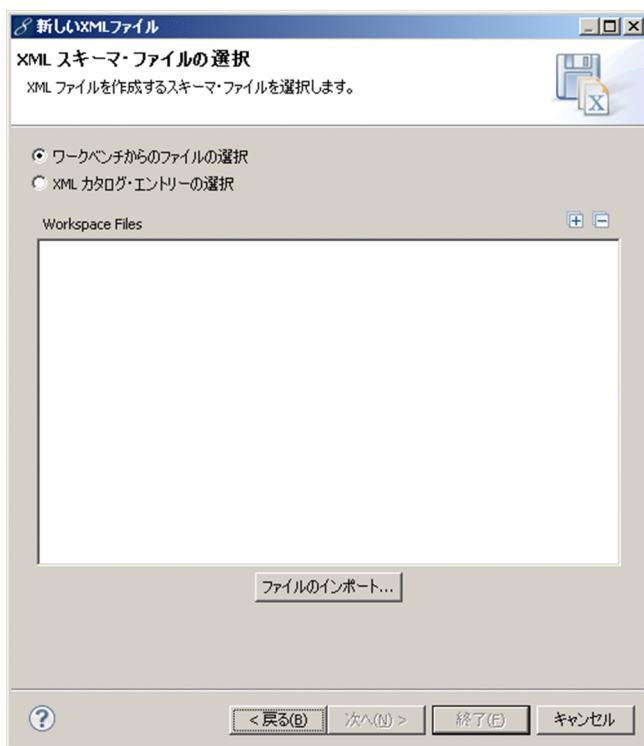


3. 変換ファンクション定義ファイルを保存するディレクトリ , および任意の変換ファンクション定義ファイル名を指定して ,[次へ] ボタンをクリックします。
[次から XML ファイルを作成] ページが表示されます。



4. [XML スキーマ・ファイルから XML ファイルを作成] ラジオボタンを選択し , [次へ] ボタンをクリックします。
[XML スキーマ・ファイルの選択] ページが表示されます。

5. サンプルプログラムの開発を体験する

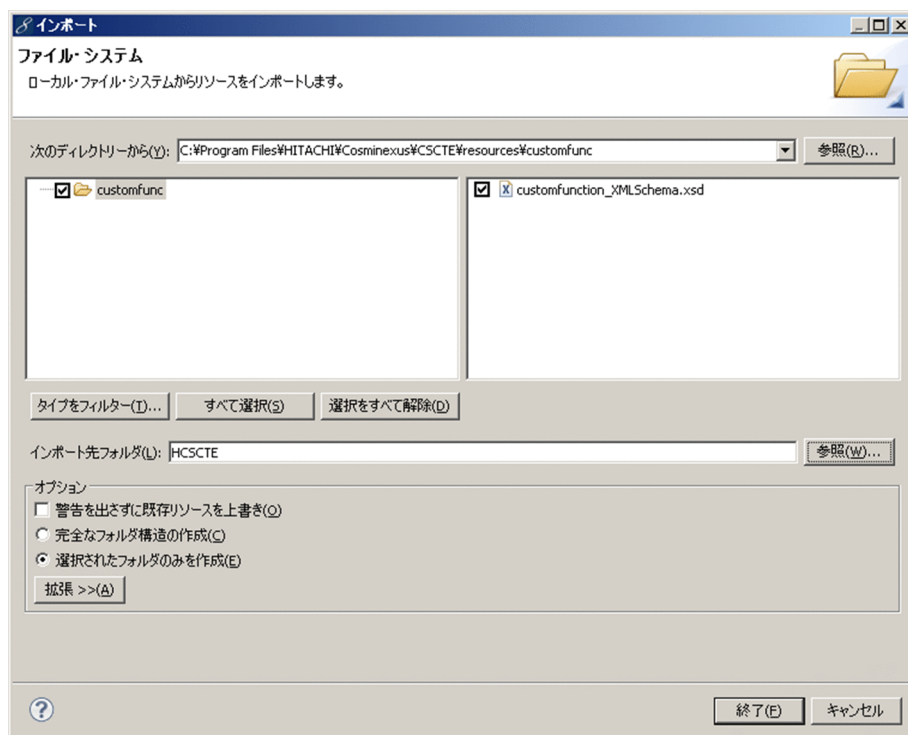


5. [ファイルのインポート] ボタンをクリックします。

[インポート] ダイアログが表示されます。

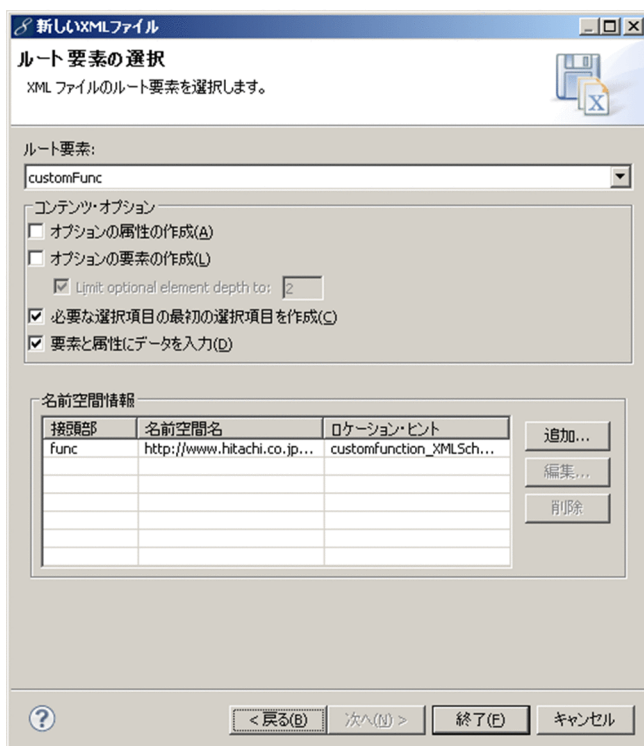
6. [次のディレクトリーから] に次のディレクトリを指定します。

```
<uCosminexus Service Architectのインストールディレクトリ  
>¥CSCTE¥resources¥customfunc
```



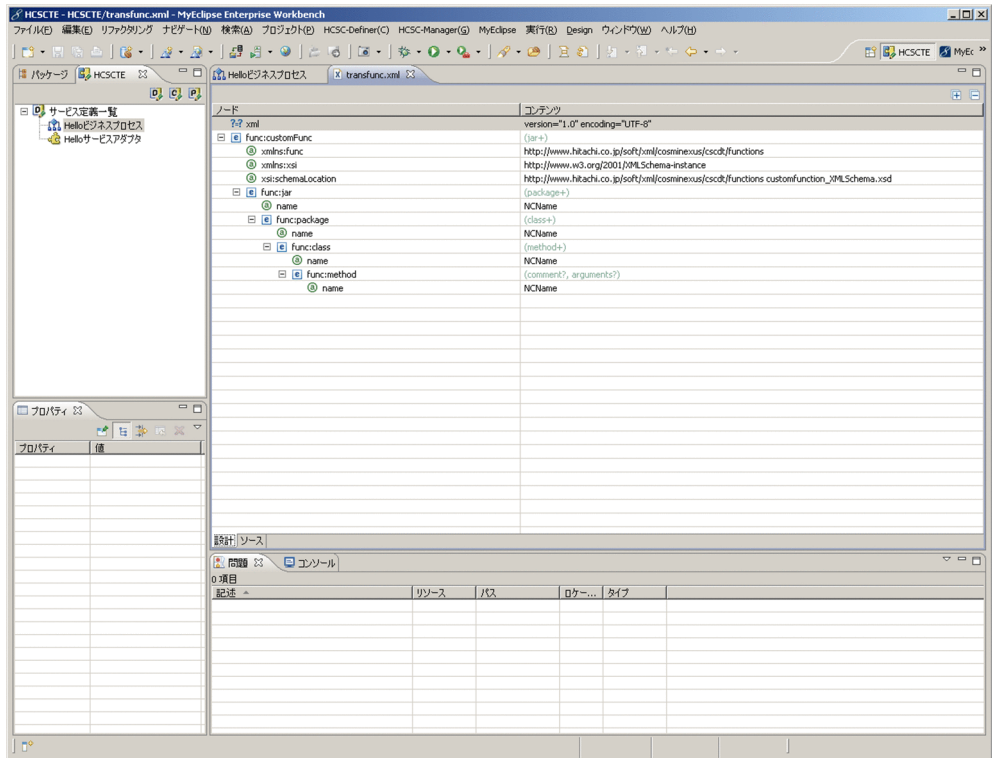
7. [customfunc] チェックボックスおよび [customfunction_XMLSchema.xsd] チェックボックスをチェックします。
8. [インポート先フォルダ] にスキーマファイルをインポートするフォルダを指定します。
9. [終了] ボタンをクリックします。
再度 [XML スキーマ・ファイルの選択] ページが表示されます。
10. [次へ] ボタンをクリックします。
[ルート要素の選択] ページが表示されます。

5. サンプルプログラムの開発を体験する

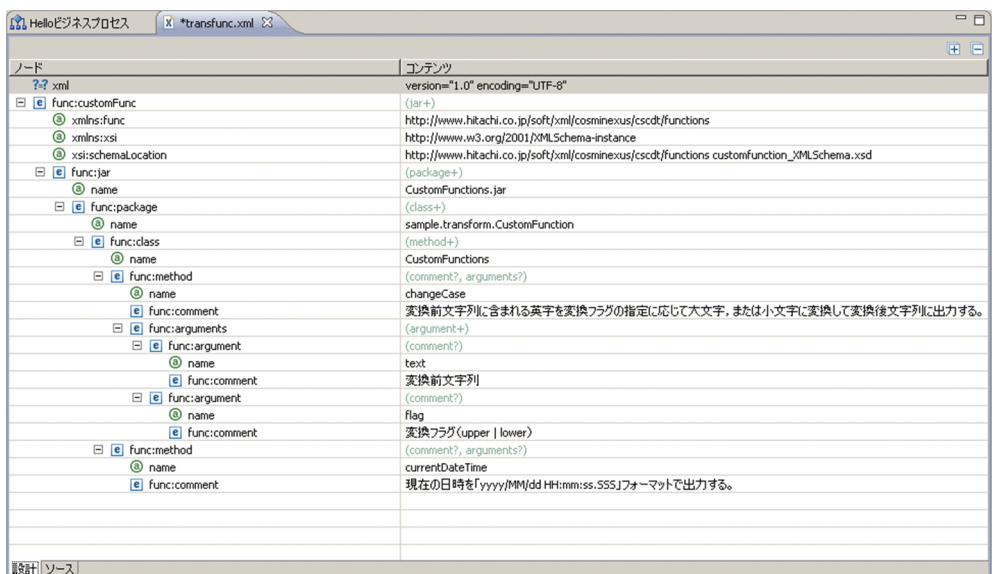


- 11.[ルート要素] で「customFunc」を選択し,[終了] ボタンをクリックします。
変換ファンクション定義ファイルの雛形が生成されます。

5. サンプルプログラムの開発を体験する



12. 変換ファンクション定義ファイルの雛形に必要な内容を定義します。
定義する内容を次に示します。



5. サンプルプログラムの開発を体験する

表 5-12 変換ファンクション定義ファイルに定義する内容

要素	定義する内容
xml	version="1.0" encoding="UTF-8"
func:customFunc	-
xmlns:func	http://www.hitachi.co.jp/soft/xml/cosminexus/cscdt/functions
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.hitachi.co.jp/soft/xml/cosminexus/cscdt/functions customfunction_XMLSchema.xsd
func:jar	-
name	CustomFunctions.jar
func:package	-
name	sample.transform.CustomFunction
func:class	-
name	CustomFunctions
func:method	-
name	changeCase
func:comment	変換前文字列に含まれる英字を変換フラグの指定に応じて大文字，または小文字に変換して変換後文字列に出力する。
func:arguments	-
func:argument	-
name	text
func:comment	変換前文字列
func:argument	-
name	flag
func:comment	変換フラグ (upper lower)
func:method	-
name	currentDateTime
func:comment	現在の日時を「yyyy/MM/dd HH:mm:ss.SSS」フォーマットで出力する。

(凡例)

- : ありません。

13. MyEclipse のメニューで [ファイル] - [保存] を選択します。

変換ファンクション定義ファイルの作成が完了します。

(2) Java プログラムの作成

CustomFunction サンプルプログラムでは、次の Java プログラムを呼び出すカスタムファンクションを作成します。

図 5-4 カスタムファンクションから呼び出す Java プログラム

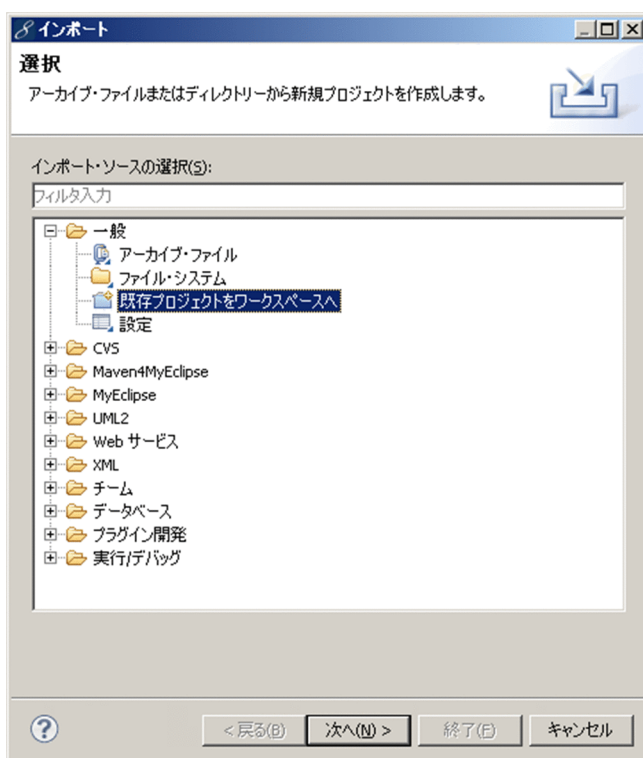
```
CustomFunctions.jar.....jarファイル名
sample.transform.CustomFunction.....パッケージ名
CustomFunctions.....クラス名
    String changeCase.....文字列に含まれる英字を、変換フラグの指定に応じて大文字または
                           小文字に変換するメソッドです。変換フラグには次の値を指定します。
                           ・ upper : 大文字に変換する場合の変換フラグです。
                           ・ lower : 小文字に変換する場合の変換フラグです。
    String currentDateTime.....現在の日時を次のフォーマットで出力するメソッドです。
                           yyyy/MM/dd HH:mm:ss.SSS
```

CustomFunction サンプルプログラムでは、この Java プログラムの処理がコーディングされたソースファイル (CustomFunctions.java) が用意されています。このソースファイルをインポートおよびパッケージングして、カスタムファンクションから呼び出す Java プログラムを作成します。

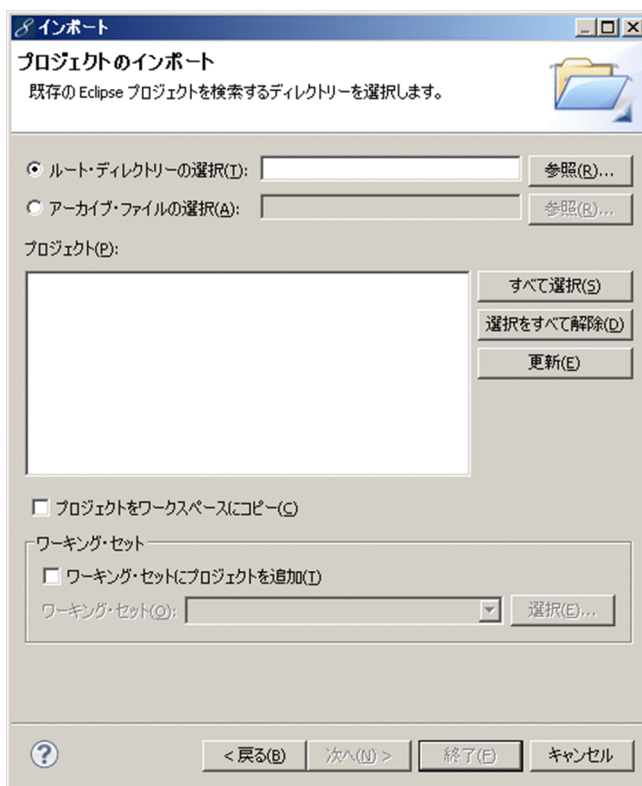
Java プログラムを作成する手順を次に示します。

1. MyEclipse のメニューから、[ウィンドウ] - [パースペクティブを開く] - [その他] を選択します。
[パースペクティブを開く] ダイアログが表示されます。
2. [Java] を選択し、[OK] ボタンをクリックします。
[Java] パースペクティブが表示されます。
3. MyEclipse のメニューから、[ファイル] - [インポート] を選択します。
選択画面が表示されます。
4. ツリービューから [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. サンプルプログラムの開発を体験する

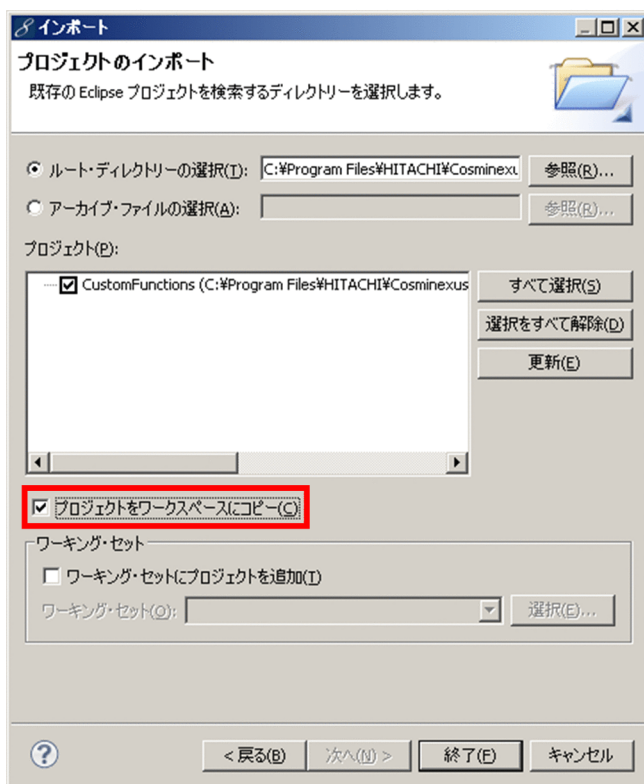


5. [次へ] ボタンをクリックします。
プロジェクトのインポート画面が表示されます。



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ます。
ファイルの参照画面が表示されます。
7. CustomFunction サンプルプログラムが格納されているディレクトリ (<uCosminexus
Service Architect のインストールディレクトリ
>¥CSCTE¥Samples¥CustomFunctions¥CustomFunction¥CustomFunctions) を選択
し , [OK] ボタンをクリックします。
プロジェクトのインポート画面に戻ります。
8. [プロジェクトをワークスペースにコピー] チェックボックスにチェックを入れます。

5. サンプルプログラムの開発を体験する



9. [終了] ボタンをクリックします。

Java プログラムのインポートが完了します。

10. [パッケージ・エクスプローラー] ビューで、インポートした [CustomFunctions] プロジェクト内の [build.xml] を右クリックし、[実行] - [Ant ビルド] を選択します。

Java プログラムがパッケージングされ、jar ファイル (CustomFunctions.jar) が次のディレクトリに作成されます。

<MyEclipse のワークスペースディレクトリ>\CustomFunctions\build\lib

なお、[Ant ビルド] が表示されない場合は、MyEclipse のメニューから [実行] - [外部ツール] - [外部ツールダイアログを開く] を選択して、[Ant ビルド] の設定を実施してください。

11. 作成された jar ファイルを次の場所にコピーします。

<uCosminexus Service Architectのインストールディレクトリ>\CSC\userlib\customfunc

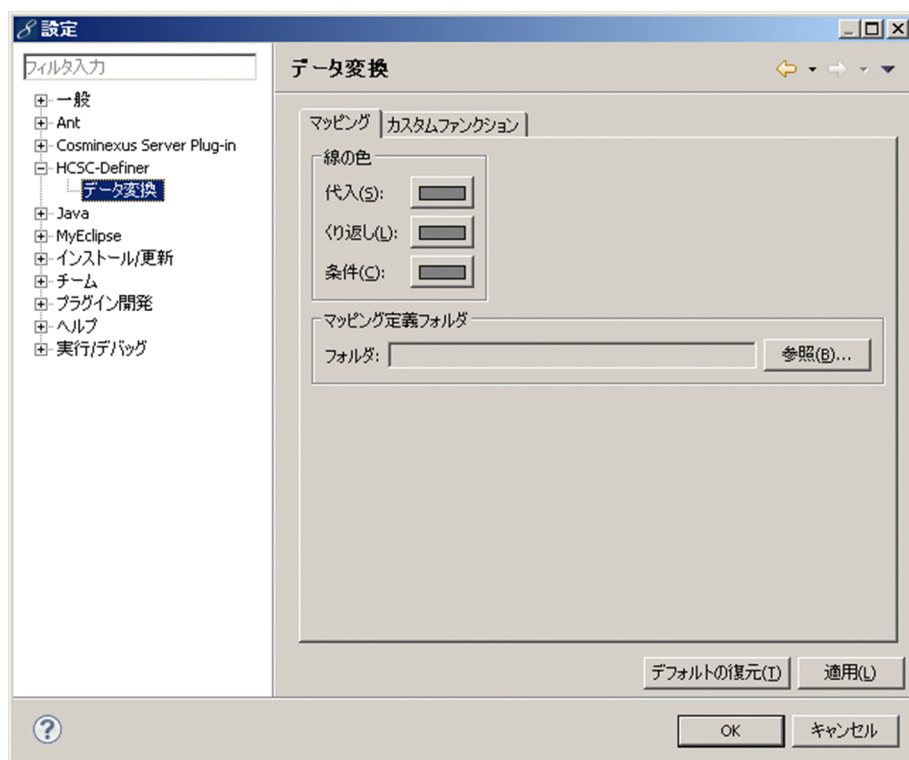
これで Java プログラムの作成が完了しました。

5.9.3 Hello サービスアダプタの変更

CustomFunction サンプルプログラムでは、サービスアダプタを介してサービスを呼び出す際に英字データを大文字に変換します。このためには、Hello サービスアダプタの定義を変更して、データ変換を定義する必要があります。

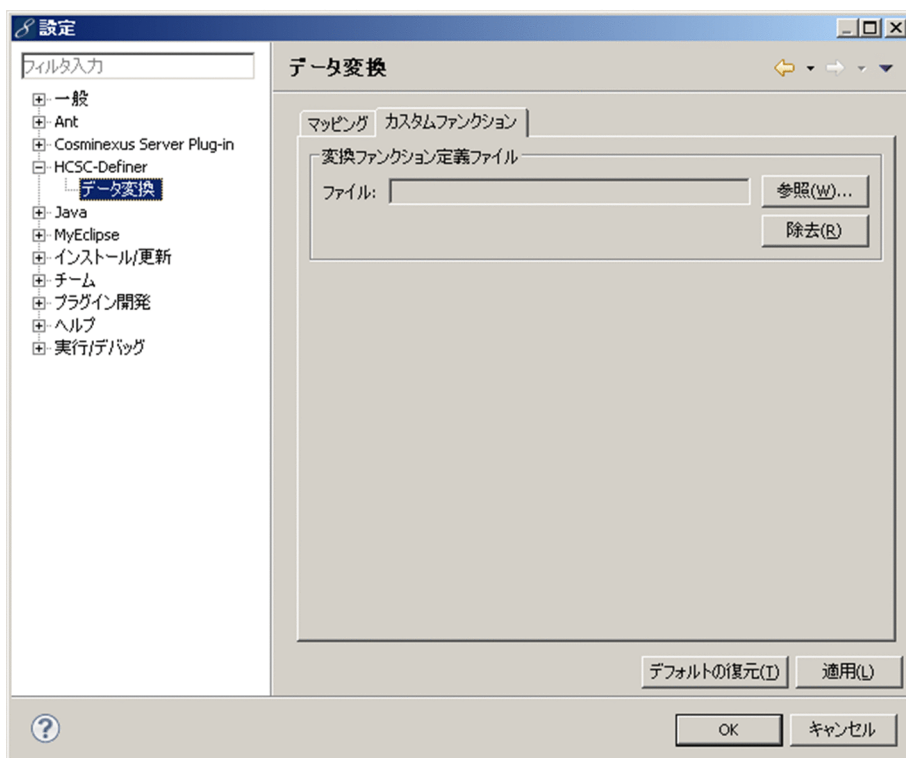
Hello サービスアダプタの変更手順を次に示します。

1. MyEclipse 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. 左ペインで [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表示されます。

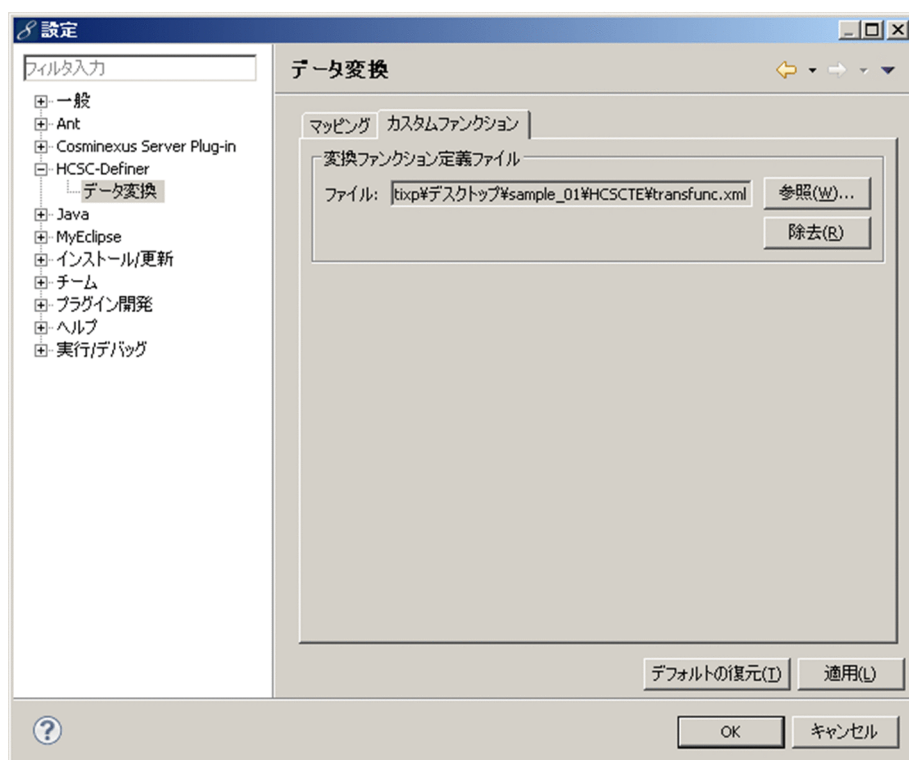


3. [カスタムファンクション] タブをクリックします。
[設定] ダイアログの [カスタムファンクション] タブが表示されます。

5. サンプルプログラムの開発を体験する

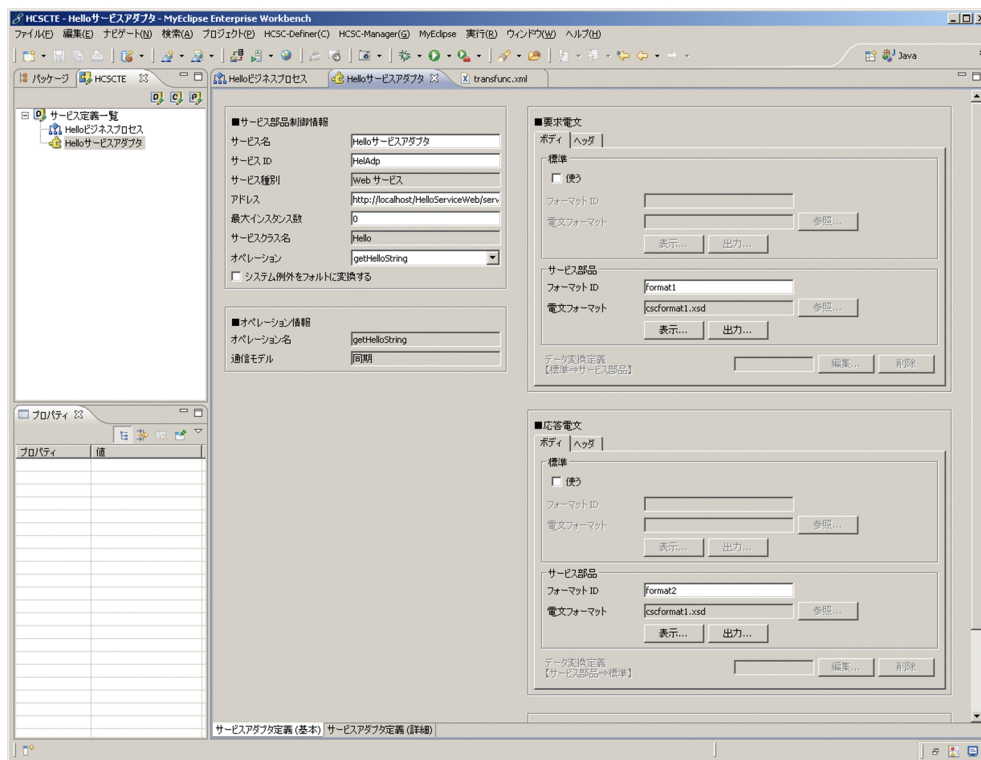


4. [参照] ボタンをクリックします。
ファイルの選択画面が表示されます。
5. 「5.9.2(1) 変換ファンクション定義ファイルの作成」で作成した変換ファンクション定義ファイルを選択し、[OK] ボタンをクリックします。
変換ファンクション定義ファイルが選択されます。



6. [OK] ボタンをクリックします。
変換ファンクション定義ファイルが登録されます。
7. [Hello サービスアダプタ] タブをクリックします。
Hello サービスアダプタの開発で定義したサービスアダプタ定義（基本）画面が表示されます。

5. サンプルプログラムの開発を体験する



8. [要求電文] - [標準] の [使う] チェックボックスにチェックを入れ, [フォーマット ID], [電文フォーマット], および [データ変換定義] に次の値を設定します。

The screenshot shows a configuration window for 'HelloService'. It is divided into several sections:

- サービス部品制御情報 (Service Component Control Information):**
 - サービス名: Helloサービスアダプタ
 - サービス ID: HelAdp
 - サービス種別: Web サービス
 - アドレス: http://localhost/HelloServiceWeb/serv
 - 最大インスタンス数: 0
 - サービスクラス名: Hello
 - オペレーション: getHelloString
 - ☐ システム例外をフォルトに変換する
- オペレーション情報 (Operation Information):**
 - オペレーション名: getHelloString
 - 通信モデル: 同期
- 要求電文 (Request Message):**
 - 標準: ☒ 使う
 - フォーマット ID: hello01
 - 電文フォーマット: HelloService.xsd
 - Buttons: 表示..., 出力..., 参照...
- サービス部品 (Service Component):**
 - フォーマット ID: format1
 - 電文フォーマット: cscformat1.xsd
 - Buttons: 表示..., 出力..., 参照...
- データ変換定義 (Data Transformation Definition):**
 - 定義名: TRANSFORM
 - Buttons: 編集..., 削除
- 応答電文 (Response Message):**
 - 標準: ☐ 使う
 - フォーマット ID: (empty)
 - 電文フォーマット: (empty)
 - Buttons: 表示..., 出力..., 参照...
- サービス部品 (Service Component):**
 - フォーマット ID: format2
 - 電文フォーマット: cscformat1.xsd
 - Buttons: 表示..., 出力..., 参照...
- データ変換定義 (Data Transformation Definition):**
 - 定義名: (empty)
 - Buttons: 編集..., 削除

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)'.

項番	項目	設定値
1	フォーマット ID	hello01
2	電文フォーマット	<uCosminexus Service Architect のインストールディレクトリ>¥CSCTE¥Samples¥CustomFunctions¥CustomFunction¥Schema¥HelloService.xsd
3	データ変換定義	TRANSFORM

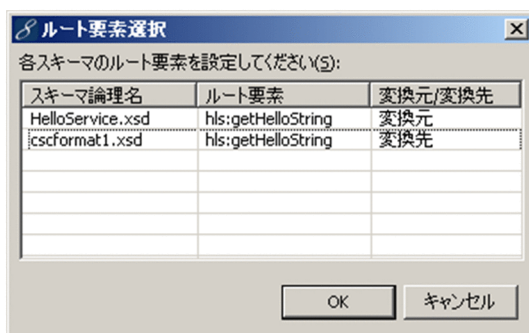
9. [編集] ボタンをクリックします。

[ルート要素選択] ダイアログが表示されます。

10. [変換元] および [変換先] のルート要素として、ドロップダウンリストから次を選択します。

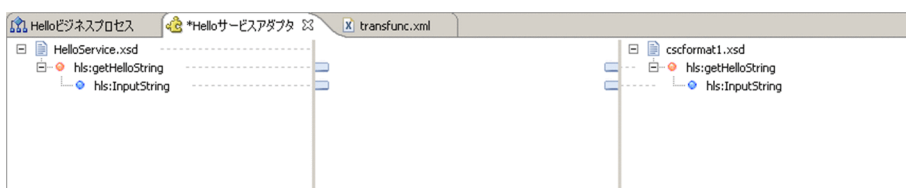
- [変換元] のスキーマ論理名「HelloService.xsd」のルート要素：
hls:getHelloString
- [変換先] のスキーマ論理名「cscformat1.xsd」のルート要素：hls:getHelloString

5. サンプルプログラムの開発を体験する



11.[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。



12.データ変換定義画面のパレットで [custom] をクリックし、キャンパスの変換元と変換先の間にクリックして配置します。

13.データ変換定義画面のパレットで [const] をクリックし、キャンパスの変換元と変換先の間にクリックして配置します。

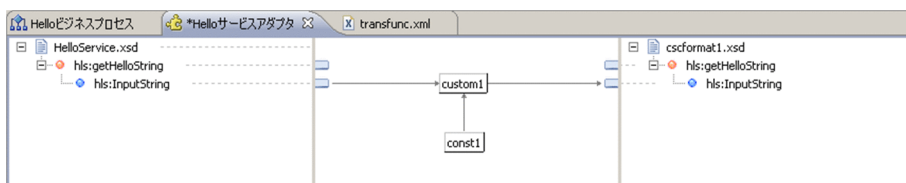
14.データ変換定義画面のパレットで [マッピング] をクリックします。

15.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。

16.マッピング先となる [custom] をクリックします。
マッピング線が設定されます。

17.操作 14. ~ 操作 16. と同様の操作で、[custom] 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。

18.操作 14. ~ 操作 16. と同様の操作で、[const] から [custom] にマッピング線を設定します。

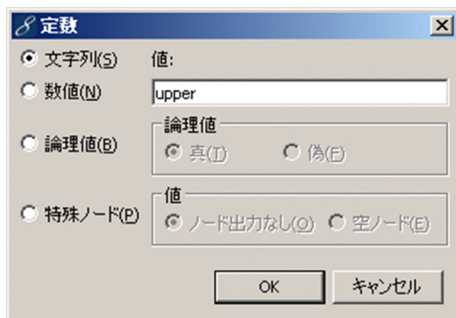


19.データ変換定義画面のパレットで [選択] をクリックします。

20. データ変換定義画面で [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。

21. [文字列] を選択して、「upper」を入力します。



22. [OK] ボタンをクリックします。

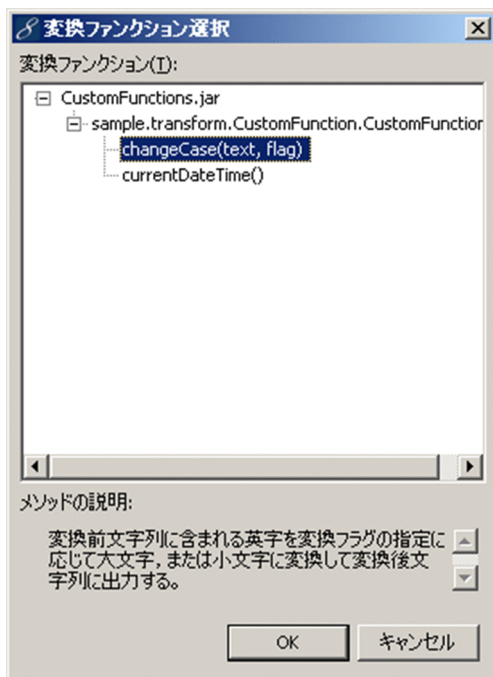
[定数] ダイアログが閉じます。

23. データ変換定義画面で [custom] をダブルクリックします。

[カスタム] ダイアログが表示されます。

24. [変換ファンクション選択] ボタンをクリックします。

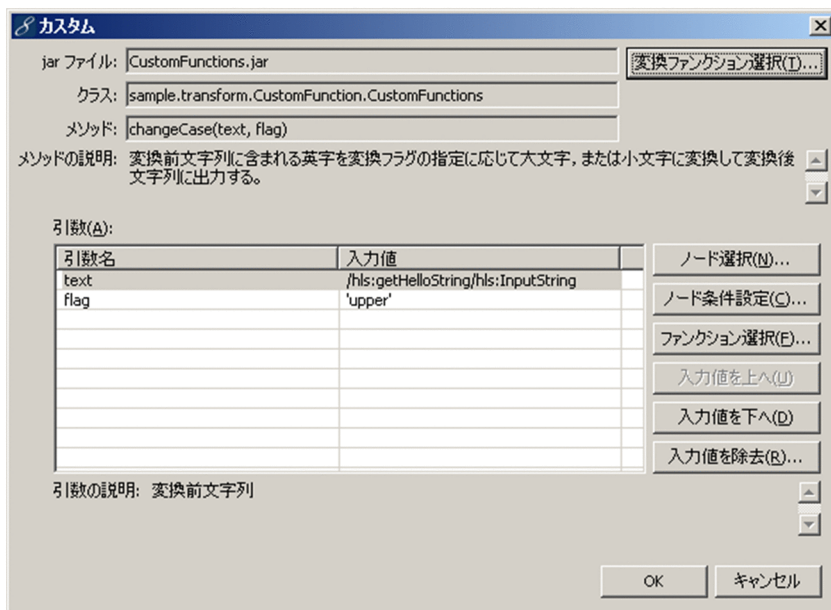
[変換ファンクション選択] ダイアログが表示されます。



25. 変換ファンクションのツリーから「changeCase(text,flag)」を選択し,[OK] ボタンをクリックします。

5. サンプルプログラムの開発を体験する

[変換ファンクション選択] ダイアログが閉じ , [カスタム] ダイアログが再度表示されます。



26 [引数名] と [入力値] が次のように定義されていることを確認します。

[引数名] と [入力値] の組み合わせが異なる場合は , [入力値を上へ] ボタンまたは [入力値を下へ] ボタンで正しい組み合わせにしてください。

項番	引数名	入力値
1	text	/hls:getHelloString/hls:InputString
2	flag	'upper'

27 [OK] ボタンをクリックします。

[カスタム] ダイアログが閉じます。

28. MyEclipse のメニューから [ファイル] - [保存] を選択します。

これで , Hello サービスアダプタの変更が完了しました。

29. 変更したサービスアダプタを検証およびパッケージングします。

検証およびパッケージング方法については , 「5.4.2 サービスアダプタの検証とパッケージング」を参照してください。

30. 変更したサービスアダプタを配備定義します。

配備定義方法については , 「5.4.3 サービスアダプタの配備定義」を参照してください。

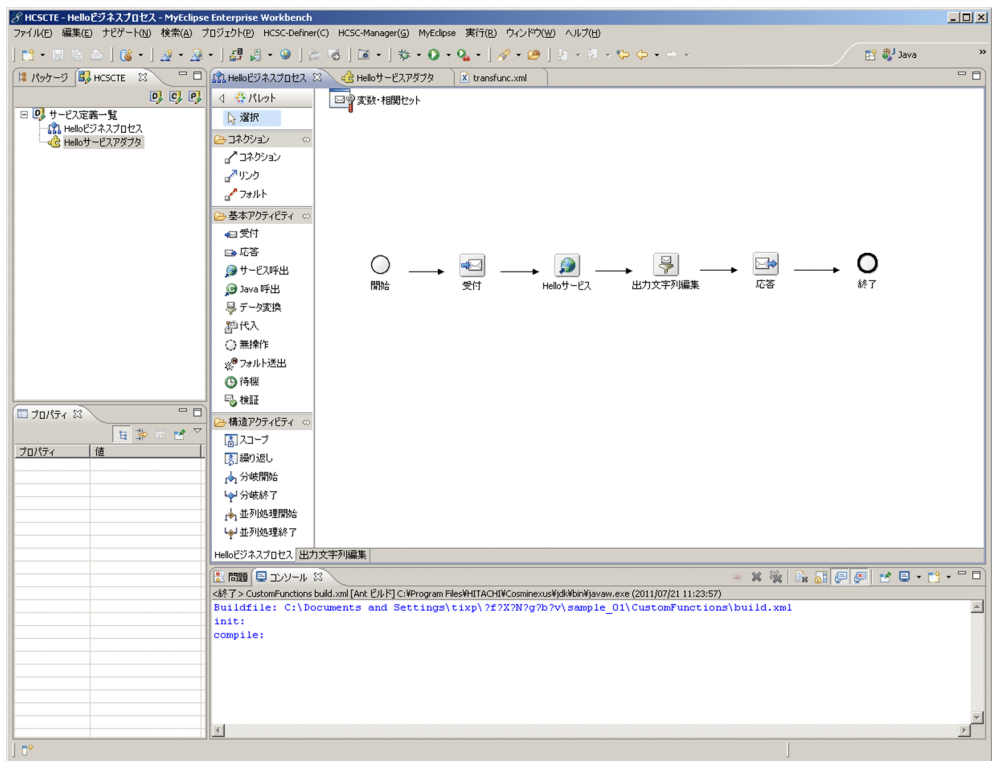
5.9.4 Hello ビジネスプロセスの変更

CustomFunction サンプルプログラムでは、出力する文字列の末尾にビジネスプロセス実行時の日時を連結します。このためには、Hello ビジネスプロセスのデータ変換定義を変更する必要があります。

Hello ビジネスプロセスの変更手順を次に示します。

1. [Hello ビジネスプロセス] タブをクリックします。

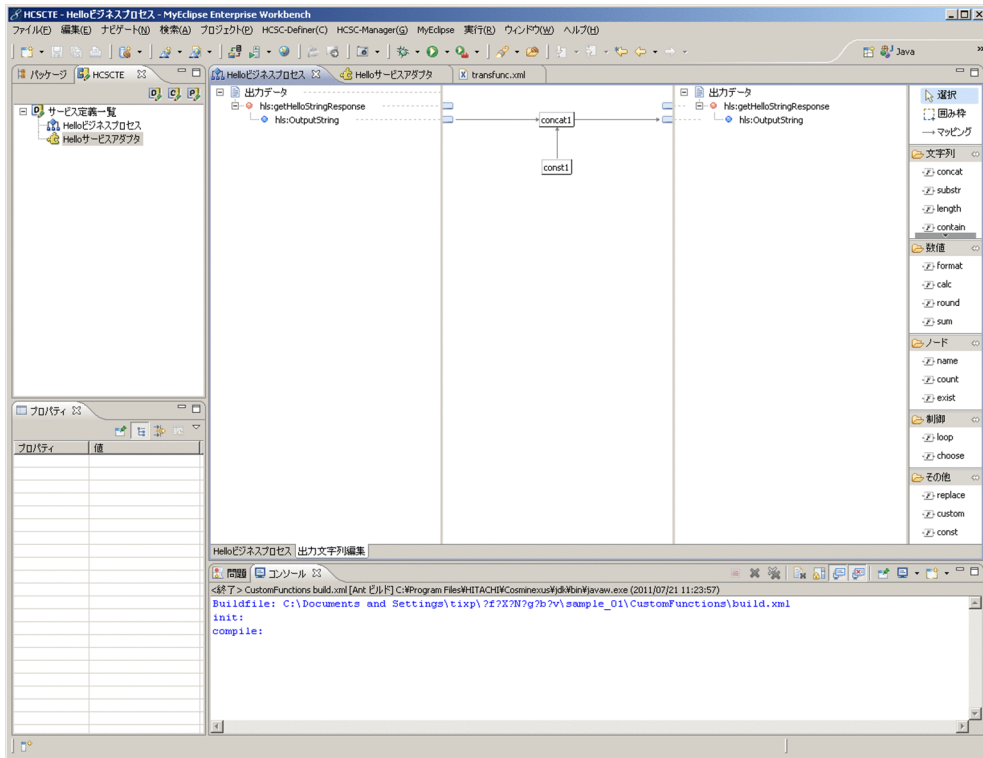
Hello ビジネスプロセスの開発で定義したビジネスプロセス定義画面が表示されます。



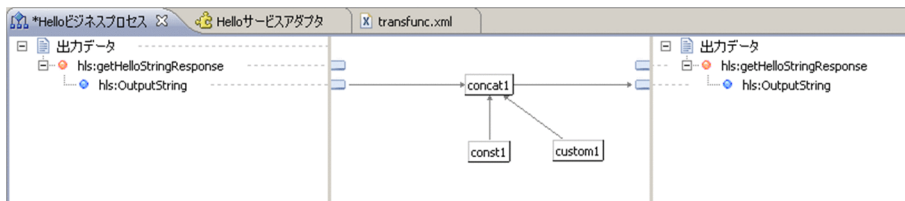
2. [出力文字列編集] タブをクリックします。

Hello ビジネスプロセスの開発で定義したデータ変換定義画面が表示されます。

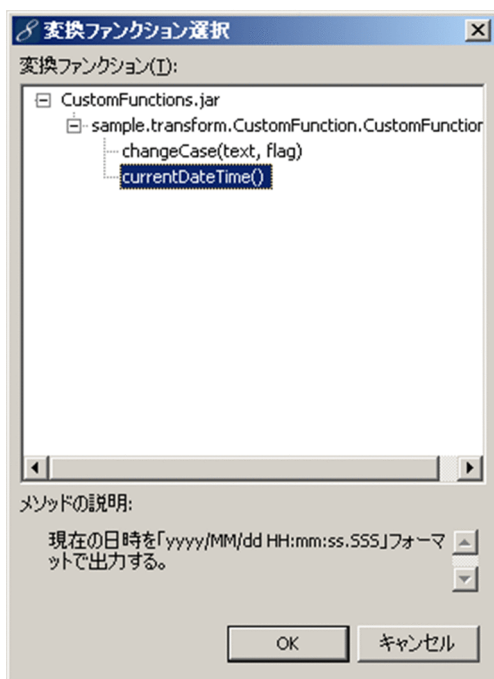
5. サンプルプログラムの開発を体験する



3. データ変換定義画面のパレットで [custom] をクリックし、キャンパスの変換元と変換先の間にクリックして配置します。
4. データ変換定義画面のパレットで [マッピング] をクリックします。
5. マッピング元となる [custom] をクリックします。
6. 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。



7. 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。
8. データ変換定義画面で [custom] をダブルクリックします。
[カスタム] ダイアログが表示されます。
9. [変換ファンクション選択] ボタンをクリックします。
[変換ファンクション選択] ダイアログが表示されます。



10. 変換ファンクションのツリーから「currentDateTime()」を選択し、[OK] ボタンをクリックします。

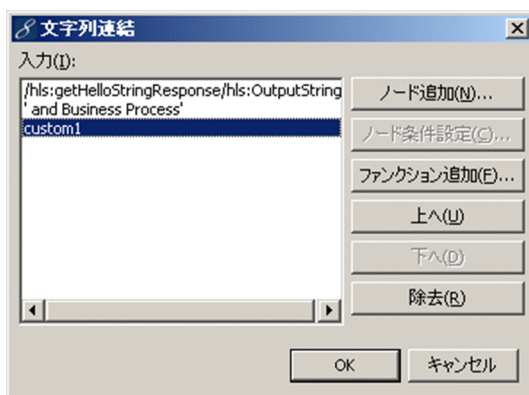
[変換ファンクション選択] ダイアログが閉じ、[カスタム] ダイアログが再度表示されます。

11. [OK] ボタンをクリックします。

[カスタム] ダイアログが閉じます。

12. データ変換定義画面で [concat] 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。



13. [入力] のいちばん下に [custom1] が追加されていることを確認し、[OK] ボタンを

5. サンプルプログラムの開発を体験する

クリックします。

[文字列連結] ダイアログが閉じます。

14. データ変換定義画面で [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。

15. [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)。



16. [OK] ボタンをクリックします。

[定数] ダイアログが閉じます。

17. MyEclipse のメニューから [ファイル] - [保存] を選択します。

これで, Hello ビジネスプロセスの変更が完了しました。

18. 変更したビジネスプロセスを検証およびパッケージングします。

検証およびパッケージング方法については, 「5.4.2 サービスアダプタの検証とパッケージング」を参照してください。

19. 変更したビジネスプロセスを配備定義します。

配備定義方法については, 「5.4.3 サービスアダプタの配備定義」を参照してください。

開発した CustomFunction サンプルプログラムを実行する場合

次の手順に従って, HelloBusinessProcess サンプルプログラムを実行する例と同じように, サンプルプログラムを実行してください。

1. 「5.8 開発情報のエクスポートとインポート」に従って, 開発環境のリポジトリ情報を ZIP 形式のファイルとしてエクスポートしてから, 運用環境のリポジトリにインポートします。
2. 「3.5.5 MyEclipse プロジェクトのインポート」の手順を実施します。
3. 「3.5.6 Web プロジェクトのデプロイ」の手順を実施します。
4. 「3.5.7 HCSC サーバへの定義内容の配備」の手順 7 以降を実施します。

5.10 アンセットアップとアンインストール

ここでは、MyEclipse のアンセットアップと uCosminexus Service Architect のアンインストールの手順について説明します。

5.10.1 MyEclipse のアンセットアップ

MyEclipse セットアップ機能で構築した環境は、MyEclipse セットアップ機能でアンセットアップします。この機能では、MyEclipse および Eclipse のアンセットアップができます。

ここでは、MyEclipse セットアップ機能を使用したアンセットアップの手順について説明します。

注意事項

アンセットアップ実行前に、MyEclipse を終了してください。MyEclipse 起動中にアンセットアップを実行すると、Eclipse のインストールディレクトリが削除されません。なお、アンセットアップ実行後に Eclipse のインストールディレクトリが残っていた場合は、`<Eclipse のインストールディレクトリ>\eclipse` 以下のディレクトリおよびファイルを手動で削除してください。

5. サンプルプログラムの開発を体験する

1. スタートメニューから , [プログラム] - [Cosminexus] - [MyEclipse] - [アンセットアップ] を選択します。

MyEclipse セットアップ機能が起動して , [アンセットアップ - MyEclipse セットアップ] ダイアログの [アンセットアップの確認] ページが表示されます。



2. [アンセットアップの内容] エリアに表示された内容を確認して、[実行] ボタンをクリックします。
[進行状況] ページが表示されます。
アンセットアップが終了すると、[アンセットアップの完了] ページが表示されます。



3. [終了] ボタンをクリックします。
[アンセットアップ - MyEclipse セットアップ] ダイアログが閉じます。

5.10.2 uCosminexus Service Architect のアンインストール

uCosminexus Service Architect をアンインストールします。アンインストールには、Administrator 権限または管理者特権が必要です。

なお、uCosminexus Service Architect をアンインストールする前に、必ず MyEclipse をアンセットアップしてください。

1. Windows の [スタート] メニューから、[Cosminexus] - [uCosminexus Service Architect アンインストール] を選択します。
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。

5. サンプルプログラムの開発を体験する

[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。

付録

付録 A サンプルプログラムのファイルの構成

付録 B インスタントセットアップ機能実行時の情報の採取

付録 C このマニュアルの参考情報

付録 D 用語解説

付録 A 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 A-1 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	<uCosminexus Service Architect のインストールディレクトリ>¥CSCTE¥Samples¥HelloServiceAdapter
HelloBusinessProcess	<uCosminexus Service Architect のインストールディレクトリ>¥CSCTE¥Samples¥HelloBusinessProcess
商品手配	<uCosminexus Service Architect のインストールディレクトリ>¥CSCTE¥Samples¥商品手配
CustomFunction	<uCosminexus Service Architect のインストールディレクトリ>¥CSCTE¥Samples¥CustomFunctions

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。

付録 A.1 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
org.eclipse.wst.common.component
org.eclipse.wst.common.project.facet.core.xml
META-INF
application.xml
MANIFEST.MF

```

HelloClientWeb ..... MyEclipse Java Web プロジェクト
.classpath
.mymetadata
.project
.myeclipse
.settings
.jsdtscope
org.eclipse.core.resources.prefs
org.eclipse.core.runtime.prefs
org.eclipse.jdt.core.prefs
org.eclipse.wst.common.component
org.eclipse.wst.common.project.facet.core.xml

org.eclipse.wst.jsdt.ui.superType.container
org.eclipse.wst.jsdt.ui.superType.name
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
    .mymetadata
    .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
    org.eclipse.wst.common.component
    org.eclipse.wst.common.project.facet.core.xml
  META-INF
    application.xml
    MANIFEST.MF

  HelloServiceWeb ..... MyEclipse Java Web プロジェクト
    .classpath
    .mymetadata
    .project
    .myeclipse

    .settings
    .jsdtscope
    org.eclipse.core.runtime.prefs
    org.eclipse.jdt.core.prefs
    org.eclipse.wst.common.component
    org.eclipse.wst.common.project.facet.core.xml
    org.eclipse.wst.jsdt.ui.superType.container
    org.eclipse.wst.jsdt.ui.superType.name

  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

```

付録 A.2 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
        org.eclipse.wst.common.component
        org.eclipse.wst.common.project.facet.core.xml

    META-INF
        application.xml
        MANIFEST.MF

    HelloClientWeb ..... MyEclipse Java Web プロジェクト
        .classpath
        .mymetadata
        .project
        .myeclipse
        .settings
        .jsdtscope
        org.eclipse.core.resources.prefs
        org.eclipse.core.runtime.prefs
        org.eclipse.jdt.core.prefs
        org.eclipse.wst.common.component
        org.eclipse.wst.common.project.facet.core.xml

        org.eclipse.wst.jsdt.ui.superType.container
        org.eclipse.wst.jsdt.ui.superType.name

    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
    .mymetadata
    .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
      org.eclipse.wst.common.component
      org.eclipse.wst.common.project.facet.core.xml
    META-INF
      application.xml
      MANIFEST.MF

HelloServiceWeb ..... MyEclipse Java Web プロジェクト
  .classpath
  .mymetadata
  .project
  .myeclipse
  .settings
  .jsdtscope
  org.eclipse.core.runtime.prefs
  org.eclipse.jdt.core.prefs
  org.eclipse.wst.common.component
  org.eclipse.wst.common.project.facet.core.xml

  org.eclipse.wst.jsdt.ui.superType.container
  org.eclipse.wst.jsdt.ui.superType.name
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

```

付録 A.3 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
      .mymetadata
      .project
      .settings
      org.eclipse.core.runtime.prefs
      org.eclipse.wst.common.component
      org.eclipse.wst.common.project.facet.core.xml
    META-INF
      application.xml
      MANIFEST.MF
    ArrangementClientWeb ..... MyEclipse Java Web プロジェクト
      .classpath
      .mymetadata
      .project
      .myeclipse
      .settings
      .jsdtscope
      org.eclipse.core.resources.prefs
      org.eclipse.core.runtime.prefs
      org.eclipse.jdt.core.prefs
      org.eclipse.jdt.ui.prefs
      org.eclipse.wst.common.component
      org.eclipse.wst.common.project.facet.core.xml

```

```

org.eclipse.wst.jsdt.ui.superType.container
org.eclipse.wst.jsdt.ui.superType.name
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
    org.eclipse.wst.common.component
    org.eclipse.wst.common.project.facet.core.xml
  META-INF
    application.xml
    MANIFEST.MF
  DeliveryServiceWeb ..... MyEclipse Java Web プロジェクト
    .classpath
    .mymetadata
    .project
    .myeclipse
    .settings
    .jsdtscope
    org.eclipse.core.runtime.prefs
    org.eclipse.jdt.core.prefs
    org.eclipse.wst.common.component
    org.eclipse.wst.common.project.facet.core.xml

```

```

org.eclipse.wst.jsdt.ui.superType.container
org.eclipse.wst.jsdt.ui.superType.name
src
  sample
    DeliveryService
    Delivery.java ..... サービスケルトン
    DeliverySoapBindingImpl.java ..... 配送受付サービス（サービスケルトン）
WebRoot
  META-INF
  MANIFEST.MF
  WEB-INF
    server-config.xml ..... サービスデプロイ定義ファイル（サービスケルトン）
    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
  c4webcl.properties ..... 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
    org.eclipse.wst.common.component
    org.eclipse.wst.common.project.facet.core.xml
  META-INF
    application.xml
    MANIFEST.MF

InventoryManagementServiceWeb ..... MyEclipse Java Web プロジェクト
  .classpath
  .mymetadata
  .project
  .myeclipse
  .settings
    jsdtscope
    org.eclipse.core.runtime.prefs
    org.eclipse.jdt.core.prefs
    org.eclipse.wst.common.component
    org.eclipse.wst.common.project.facet.core.xml

    org.eclipse.wst.jsdt.ui.superType.container
    org.eclipse.wst.jsdt.ui.superType.name
src
  sample
    InventoryManagementService
    InventoryManager.java ..... サービスケルトン
    InventoryManagerSoapBindingImpl.java ... 在庫管理サービス（サービスケルトン）

```

```

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
ArrangementService.wsdl ..... 商品手配用 WSDL
DeliveryService.wsdl ..... 配送受付サービス用 WSDL
InventoryManagementService.wsdl ..... 在庫管理サービス用 WSDL

```

付録 A.4 CustomFunction サンプルプログラムの構成

```

CustomFunctions
  Repository ..... リポジトリのディレクトリ
  HelloCustomFunction.zip ..... リポジトリ
  CustomFunction
    CustomFunctions
      .classpath
      .project
      build.xml ..... Ant ビルド定義ファイル
      bin
      build
      classes
      lib
      CustomFunctions.jar ..... カスタムファンクションの jar ファイル
      src
      sample
      transform
      CustomFunction
      CustomFunctions.java ..... サンプルソース
  Configuration
    CustomFunctions.xml ..... 変換ファンクション定義ファイル
  Schema
    HelloService.xsd ..... Hello サービスアダプタの修正に使用するスキーマ

```

付録 B インスタントセットアップ機能実行時の情報の採取

インスタントセットアップ機能を実行したときに設定した内容（セットアップ内容，設定変更内容，アンセットアップ内容）は，セットアップログとしてファイルに保存されます。

ここでは，MyEclipse のセットアップログの確認方法，およびセットアップログで確認できる内容を説明します。

セットアップログの確認方法を次に説明します。

セットアップログの確認方法

Windows のメニューから，[プログラム] - [Cosminexus] - [MyEclipse] - [セットアップログ] を選択すると，MyEclipse のセットアップログが表示されます。

セットアップログで確認できる内容

MyEclipse のセットアップログでは，次の内容を確認できます。

これらの情報は，実行された順番に一つのファイルに格納されます。このため，最新の情報は，ファイルの最後に格納されています。なお，エラーまたは処理の中断が発生した場合でも，これらの情報は格納されます。

- MyEclipse および Eclipse のセットアップ内容
- アンセットアップした MyEclipse および Eclipse の設定内容

なお，MyEclipse セットアップ機能実行時のエラーメッセージは，次の個所に出力されます。

- MyEclipse セットアップ機能のコンソール
- <uCosminexus Service Architect のインストールディレクトリ
>¥MyEclipse¥setup¥log 下のログファイル

出力されたエラーメッセージについては，マニュアル「Cosminexus アプリケーションサーバメッセージ 3」の「3. KEDT（インスタントセットアップ機能で出力されるメッセージ）」を参照して対処してください。

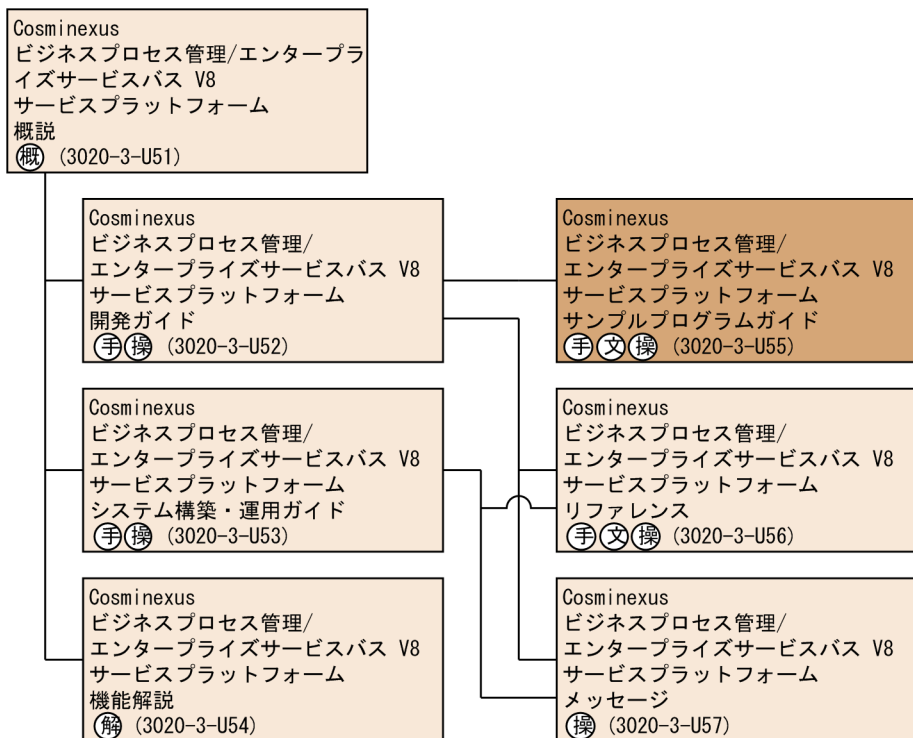
付録 C このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 C.1 関連マニュアル

このマニュアルの関連マニュアルを、次の図に示します。必要に応じてお読みください。

図 C-1 このマニュアルの関連マニュアル



<記号>

- (概) : 概説書
- (解) : 解説書
- (手) : 手引書
- (文) : 文法書
- (操) : 操作書

また、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。

Cosminexus アプリケーションサーバ関連

- Cosminexus アプリケーションサーバ V8 概説 (3020-3-U01)

- Cosminexus アプリケーションサーバ V8 ファーストステップガイド (3020-3-U02)
- Cosminexus アプリケーションサーバ V8 システム設計ガイド (3020-3-U03)
- Cosminexus アプリケーションサーバ V8 システム構築・運用ガイド (3020-3-U04)
- Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (Web コンテナ)
(3020-3-U05)
- Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (EJB コンテナ)
(3020-3-U06)
- Cosminexus アプリケーションサーバ V8 機能解説 基本・開発編 (コンテナ共通機能)
(3020-3-U07)
- Cosminexus アプリケーションサーバ V8 機能解説 拡張編 (3020-3-U08)
- Cosminexus アプリケーションサーバ V8 機能解説 運用 / 監視 / 連携編 (3020-3-U09)
- Cosminexus アプリケーションサーバ V8 機能解説 保守 / 移行 / 互換編 (3020-3-U10)
- Cosminexus アプリケーションサーバ V8 アプリケーション設定操作ガイド
(3020-3-U12)
- Cosminexus アプリケーションサーバ V8 運用管理ポータル操作ガイド (3020-3-U13)
- Cosminexus アプリケーションサーバ V8 リファレンス コマンド編 (3020-3-U14)
- Cosminexus アプリケーションサーバ V8 リファレンス 定義編 (サーバ定義)
(3020-3-U15)
- Cosminexus アプリケーションサーバ V8 リファレンス 定義編 (アプリケーション / リ
ソース定義) (3020-3-U16)
- Hitachi Web Server (3020-3-U17)
- Cosminexus アプリケーションサーバ V8 仮想化システム構築・運用ガイド
(3020-3-U18)
- TPBroker Version 5 トランザクショナル分散オブジェクト基盤 TPBroker ユーザーズ
ガイド (3020-3-U19)
- TPBroker Version 5 トランザクショナル分散オブジェクト基盤 TPBroker 運用ガイド
(3020-3-U20)
- Cosminexus アプリケーションサーバ V8 Cosminexus Reliable Messaging
(3020-3-U21)
- Cosminexus アプリケーションサーバ V8 アプリケーション開発ガイド (3020-3-U25)
- Cosminexus アプリケーションサーバ V8 リファレンス API 編 (3020-3-U26)
- Cosminexus アプリケーションサーバ V8 Cosminexus XML Processor ユーザーズガイ
ド (3020-3-U27)
- VisiBroker Version 5 Borland(R) Enterprise Server VisiBroker(R) デベロッパーズガ
イド (3020-3-U28)
- VisiBroker Version 5 Borland(R) Enterprise Server VisiBroker(R) プログラマーズリ
ファレンス (3020-3-U29)
- Cosminexus アプリケーションサーバ V8 SOAP アプリケーション開発の手引
(3020-3-U30)
- Cosminexus アプリケーションサーバ V8 Web サービス開発の手引 (3020-3-U31)

- Cosminexus アプリケーションサーバ V8 Web サービスセキュリティ 使用の手引 (3020-3-U32)
- Cosminexus アプリケーションサーバ V8 Cosminexus XML Security - Core ユーザーズガイド (3020-3-U33)
- Cosminexus アプリケーションサーバ V8 メッセージ 1 KDAL-KDCG および Hitachi Web Server 編 (3020-3-U41)
- Cosminexus アプリケーションサーバ V8 メッセージ 2 KDJE-KDJW 編 (3020-3-U42)
- Cosminexus アプリケーションサーバ V8 メッセージ 3 KECX-KEDT / KEOS02000-29999 / KEUC-KFRM 編 (3020-3-U43)
- Cosminexus アプリケーションサーバ V8 メッセージ 4 監査ログ編 (3020-3-U44)

なお、このマニュアルでは、次のマニュアルについて、対象 OS およびバージョン番号を省略して表記しています。マニュアルの正式名称とこのマニュアルでの表記を次の表に示します。

正式名称	このマニュアルでの表記
Cosminexus アプリケーションサーバ V8 メッセージ 3 KECX-KEDT / KEOS02000-29999 / KEUC-KFRM 編	Cosminexus アプリケーションサーバ メッセージ 3

付録 C.2 このマニュアルでの表記

(1) 製品名

このマニュアルで使用する表記と、対応する製品名を次に示します。

表記			製品名
Eclipse			Eclipse Classic 3.6.1
IPF			Itanium(R) Processor Family
UNIX	AIX		AIX 5L V5.3
			AIX V6.1
			AIX V7.1
	HP-UX	HP-UX (IPF)	HP-UX 11i V2 (IPF)
			HP-UX 11i V3 (IPF)
	Linux	Linux (IPF)	Red Hat Enterprise Linux(R) AS 4 (IPF)
			Red Hat Enterprise Linux(R) 5 (Intel Itanium(R))
			Red Hat Enterprise Linux(R) 5 Advanced Platform (Intel Itanium(R))

表記			製品名
		Linux (x86 / AMD64 & Intel EM64T)	Red Hat Enterprise Linux(R) AS 4 (x86)
			Red Hat Enterprise Linux(R) 5 Advanced Platform (x86)
			Red Hat Enterprise Linux(R) ES 4 (x86)
			Red Hat Enterprise Linux(R) 5 (x86)
			Red Hat Enterprise Linux(R) AS 4 (AMD64 & Intel EM64T)
			Red Hat Enterprise Linux(R) 5 Advanced Platform (AMD/Intel 64)
			Red Hat Enterprise Linux(R) ES 4 (AMD64 & Intel EM64T)
			Red Hat Enterprise Linux(R) 5 (AMD/Intel 64)
			Red Hat Enterprise Linux(R) Server 6 (32-bit x86)
			Red Hat Enterprise Linux(R) Server 6 (64-bit x86_64)

(2) Cosminexus の機能名

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

表記	Cosminexus の機能名
HCSC	Hitachi Cosminexus Service Coordinator
HCSC-Business Process , ビジネスプロセス基盤 , または BP 基盤	Hitachi Cosminexus Service Coordinator - Business Process
HCSC-Data Transform またはデータ変換基盤	Hitachi Cosminexus Service Coordinator - Data Transform
HCSC-DB Adapter または DB アダプタ	Hitachi Cosminexus Service Coordinator - Database Adapter
HCSC-Manager または HCSC-MNG	Hitachi Cosminexus Service Coordinator - Manager
HCSC-Messaging , HCSC-MSG , またはメッセージング基盤	Hitachi Cosminexus Service Coordinator - Messaging
HCSC-TE	Hitachi Cosminexus Service Coordinator Tools for Eclipse
MyEclipse	MyEclipse for Cosminexus
Service Architect	uCosminexus Service Architect
Service Platform	uCosminexus Service Platform

(3) Java 関連用語

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

表記	Java 関連用語
Connector 1.5	J2EE™ Connector Architecture 1.5
EAR	Enterprise ARchive
EJB または Enterprise JavaBeans	Enterprise JavaBeans
J2EE	J2EE™
	Java™ 2 Platform, Enterprise Edition
J2SE	Java™ 2 Platform, Standard Edition
JAR	Java™ Archive
Java	Java™
JDBC	Java™ Database Connectivity
JDK	JDK™
	Java™ Development Kit
JMS	Java™ Message Service

付録 C.3 英略語

このマニュアルで使用する英略語を次に示します。

英略語	英字での表記
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BMT	Bean Managed Transaction
BP	Business Process
BPEL	Business Process Execution Language
BPMN	Business Process Modeling Notation
CMT	Container Managed Transaction
DB	Database
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Security
IDE	Integrated Development Environment
IIOP	Internet Inter-Orb Protocol

英略語	英字での表記
MDB	Message-Driven Bean
OS	Operating System
RDB	Relational Database
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
UNC	Universal Naming Convention
URL	Uniform Resource Locator
UTF	UCS Transformation Format
VM	Virtual Machine
W3C	World Wide Web Consortium
WS-I	Web Services Interoperability
WS-R	Web Services Reliability
WSDL	Web Service Description Language
WST	Web Standard Tools
XML	Extensible Markup Language
XPath	XML Path Language

付録 C.4 KB（キロバイト）などの単位表記について

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

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。

CTM (Cosminexus Component Transaction Monitor)

構成ソフトウェアの一つです。サービス部品の実行要求が集中しないようにスケジューリングをするための機能です。

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、非同期型の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 JavaBeans)

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。Oracle Corporation から EJB 仕様が公開されています。

FTP アダプタ

Cosminexus サービスプラットフォームと FTP サーバとの間でファイルの送受信をするために利用するアダプタです。

FTP インバウンドアダプタ

FTP 連携で使用するリソースアダプタです。Connector 1.5 仕様に準拠しています。

FTP 受付

FTP インバウンドアダプタを経由して FTP クライアントからの実行要求を受け付ける機能（インターフェース）です。

ファイル操作アダプタや FTP アダプタと連携することで、Cosminexus サービスプラットフォーム上で FTP クライアントと FTP サーバの間のファイル転送ができるようになります。

FTP クライアント

FTP を使用してファイルの送受信を行うクライアント用ソフトウェアです。

FTP サーバ

FTP クライアントからの要求によってファイルを送受信するサーバ用ソフトウェアです。

FTP 連携

Cosminexus サービスプラットフォーム上で FTP クライアントと FTP サーバの間のファイル転送を実現するための機能です。

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)

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。

HCSC-Manager

Cosminexus サービスプラットフォームの運用環境で利用する運用管理機能です。

HCSC サーバのセットアップ機能、HCSC コンポーネントの配備機能、プロセスインスタンスの実行履歴管理機能などが含まれます。

HCSC-Manager コマンド共通定義ファイル (csccmd.properties)

運用環境で使用するコマンドの必須項目の省略値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。

このファイルに省略値を定義しておくと、コマンド入力時に該当するオプションの入力を省略でき

ます。

HCSC-Manager 定義ファイル (cscmng.properties)

運用環境のカスタマイズに必要な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。

リポジトリのインポート先となるディレクトリや、ログファイルの大きさ・面数などを定義できます。

HCSC-Messaging (メッセージング基盤)

サービスアダプタの定義に従って適切なサービス部品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、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、ユーザ定義受付などのリクエスト受付機能や、サービスアダプタ (Web サービス、SessionBean、MDB (WS-R)、MDB (DB キュー)、DB アダプタ) の各種アダプタ機能も含まれます。

HCSC サーバ構成定義ファイル (cscsvsetup.xml)

HCSC サーバの構成を定義するファイルです。XML 形式のファイルとして作成します。

HCSC サーバ、クラスタ、Cosminexus Manager などの情報を定義できます。

HCSC サーバセットアップ定義ファイル (cscsvsetup.properties)

HCSC サーバのセットアップに必要な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。

HCSC サーバに設定するクラスタ名や使用できる標準受付の種類などを定義できます。

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。

iWay アダプタ

Cosminexus サービスプラットフォームの実行環境から他社 ERP パッケージのサービス部品を呼び出す場合に利用するアダプタです。

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 雛形ファイル

カスタムファンクションから呼び出す変換ファンクションの雛形ファイルです。パッケージ、クラス、およびメソッド（引数を含む）の構成が記載されています。

JAX-WS エンジン

Cosminexus が提供する Web サービスの通信基盤です。SOAP1.1 仕様および SOAP1.2 仕様に対応しています。

JDBC (Java Database Connectivity)

Java のプログラムでデータベースを利用するための API です。

JavaSoft が 1996 年前半に仕様を公開しました。Java プログラムから、JDBC API を経由して「JDBC ドライバ」を呼び出し、ドライバがデータベース・サーバへの接続機能を提供します。

JP1

日立の統合システム運用管理ソフトウェアの総称です。

MBean (Managed Bean)

管理対象となるリソースを表す Java オブジェクトです。JMX 仕様に基づいて実装されます。マネージャに対するリソースのインターフェースとして機能し、リソースに関する情報や操作をマネージャに提供します。

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。

Message Queue 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メッセージキュー（IBM WebSphere MQ システム）に対してメッセージの送受信をする場合に利用するアダプタです。

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。

Object Access 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 TPBroker システム (Object Wrapper システム) のサービス部品を呼び出す場合に利用するアダプタです。

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。

SOAP1.1/1.2 併用モード

利用する Web サービスの標準仕様および通信基盤を表すモードです。SOAP1.1 仕様および SOAP1.2 仕様に対応したサービス部品 / サービスリクエストを利用するときに指定します。SOAP1.1/1.2 併用モードでは、通信基盤として Cosminexus が提供する JAX-WS エンジンを利用します。

SOAP1.1 モード

利用する Web サービスの標準仕様および通信基盤を表すモードです。SOAP1.1 仕様に対応したサービス部品 / サービスリクエストを利用するときに指定します。SOAP1.1 モードでは、通信基盤として Cosminexus が提供する SOAP 通信基盤を利用します。

SOAP 通信基盤

Cosminexus が提供する Web サービス (SOAP アプリケーション) の通信基盤です。SOAP1.1 仕様に対応しています。

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/RPC 受付

既存の OpenTP1 システム内にあるサービスリクエストからのサービス部品呼び出し要求を受け付けるための機能 (インターフェース) 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。

TP1 インバウンドアダプタ

TP1 インバウンド連携機能で使用するリソースアダプタです。Connector 1.5 仕様に準拠しています。

TP1 連携システム

OpenTP1 システム内にあるサービスリクエストおよびサービス部品と接続するためのシステム構成です。

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を、クライアントに API として提供するサービスです。

WFA (Work-Flow Architecture)

経済産業省がエンタープライズアーキテクチャで提案している業務フローの記法です。業務にかかわる組織や担当者、業務の流れ、および流れる情報を明示できます。

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。

(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。

受付処理

サービス部品の呼び出し要求を受け付けたり、電文を転送したりする処理を記述した API です。ユーザが作成した受付処理をカスタム受付フレームワーク上で動作させることで、任意のプロトコルに対応した受付を開発できます。

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。

(力行)

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。

カスタム受付

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。ユーザが作成した受付処理をカスタム受付フレームワーク上で動作させることで、任意のプロトコルからの実行要求を受け付けることができます。

カスタム受付フレームワーク

カスタム受付を作成するために Cosminexus サービスプラットフォームで提供している開発用フレームワークです。

カスタムファンクション

データ変換の定義で使用するファンクションの一つです。このファンクションでは、ユーザが作成した任意の Java プログラムを呼び出し、データを加工できます。

共通フォルダ

FTP 連携で、複数のビジネスプロセスからアクセスできるファイルを格納するために HCSC サーバ

に作成する任意のフォルダです。

クラスタ

ある共通の機能を提供するサーバの集合です。

Cosminexus サービスプラットフォームでは、二つの HCSC サーバで実行系と待機系を構成する HA クラスタと、複数の HCSC サーバでクラスタを構成するロードバランスクラスタがあります。

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て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。

サービス型ソフトウェア

ソフトウェアの機能のうち、ユーザが必要とするものだけをサービスとして配布し利用できるようにしたソフトウェアの配布形態です。SaaS (Software as a Service) ともいいます。

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) に合わせられます。

作業フォルダ

FTP 連携で、FTP クライアントや FTP サーバなどから受信したファイルを扱うために HCSC サーバに作成されるフォルダです。

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す。

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。

実行履歴

ビジネスプロセスのプロセスインスタンスの実行履歴と電文の実行履歴とがあります。ビジネスプロセスの実行状況や応答状況を確認したり、サービス部品の情報や電文の配送経路や配送状態を確認したりできます。

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。

(八行)

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。

ファイルアダプタ

Cosminexus サービスプラットフォームの実行環境から HCSC サーバ上のローカルディスクに対して直接ファイルの入出力をする場合に利用するアダプタです。

ファイル操作アダプタ

ファイルのレイアウト変換、文字コード変換、複製、および削除をするためのアダプタです。

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。

変換ファンクション

カスタムファンクションから呼び出す Java プログラムのことです。

変換ファンクション定義ファイル

カスタムファンクションから呼び出す変換ファンクションの構成を定義したファイルです。

(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。

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

ArrangementService.wsdl 110

B

BPEL〔用語解説〕194

BPMN〔用語解説〕194

C

csccmd.properties〔用語解説〕195

cscmng.properties〔用語解説〕196

cscsvconfig.properties〔用語解説〕196

cscsvsetup.properties〔用語解説〕196

cscsvsetup.xml〔用語解説〕196

CTM〔用語解説〕194

CustomFunction サンプルプログラムの処理
内容 148

D

DB Connector〔用語解説〕194

DB アダプタ〔用語解説〕194

DB キュー〔用語解説〕194

DeliveryService.wsdl 106

E

EAR ファイル〔用語解説〕194

Eclipse〔用語解説〕194

Eclipse のアーカイブファイルのダウンロード 17

[Eclipse のインストール] ページ 21

Eclipse の設定 25

EJB〔用語解説〕194

F

FTP アダプタ〔用語解説〕195

FTP インバウンドアダプタ〔用語解説〕195

FTP 受付〔用語解説〕195

FTP クライアント〔用語解説〕195

FTP サーバ〔用語解説〕195

FTP 連携〔用語解説〕195

H

HCSC〔用語解説〕195

HCSC-Business Process〔用語解説〕195

HCSC-Data Transform〔用語解説〕195

HCSC-DB Adapter〔用語解説〕195

HCSC-Manager〔用語解説〕195

HCSC-Manager コマンド共通定義ファイル
〔用語解説〕195

HCSC-Manager 定義ファイル〔用語解説〕
196

HCSC-Messaging〔用語解説〕196

HCSCTE〔用語解説〕196

HCSCTE プロジェクトの作成 75

HCSC 簡易セットアップ機能 31

HCSC コンポーネント〔用語解説〕196

HCSC サーバ 9

HCSC サーバ〔用語解説〕196

HCSC サーバ構成定義ファイル〔用語解説〕
196

HCSC サーバセットアップ定義ファイル〔用
語解説〕196

HCSC サーバへの定義内容の配備 52

HCSC サーバランタイム定義ファイル〔用語
解説〕196

HelloBusinessProcess サンプルプログラム
〔動作手順〕61

HelloBusinessProcess サンプルプログラムの
構成 181

HelloBusinessProcess サンプルプログラムの
処理詳細 11

HelloBusinessProcess サンプルプログラムの
処理内容 11

HelloBusinessProcess のシステム構成 7

HelloServiceAdapter サンプルプログラム
〔動作手順〕59

HelloServiceAdapter サンプルプログラムの
構成 178

HelloServiceAdapter サンプルプログラムの
処理詳細 10
HelloServiceAdapter サンプルプログラムの
処理内容 10
HelloServiceAdapter のシステム構成 6
Hello サービスアダプタの変更 161
Hello ビジネスプロセスの検証とパッケージ
ング 101
Hello ビジネスプロセスの配備定義 102
Hello ビジネスプロセスの変更 169
HTTP〔用語解説〕196

I

InventoryManagementService.wsdl 103
iWay アダプタ〔用語解説〕197

J

J2EE サーバ 9
J2EE サーバ〔用語解説〕197
J2EE サーバの選択 39
Java 雛形ファイル〔用語解説〕197
Java プログラムの作成 157
JAX-WS エンジン〔用語解説〕197
JDBC〔用語解説〕197
JDKの確認 25
JP1〔用語解説〕197

M

MBean〔用語解説〕197
MDB〔用語解説〕197
Message Queue アダプタ〔用語解説〕197
MyEclipse〔用語解説〕197
MyEclipse セットアップ機能 19
MyEclipse セットアップ機能の起動オプションの使用 24
MyEclipse の設定 35
MyEclipse プロジェクトのインポート 40

O

Object Access アダプタ〔用語解説〕198

R

RMI-IIOP〔用語解説〕198

S

SessionBean〔用語解説〕198
SOAP〔用語解説〕198
SOAP1.1/1.2 併用モード〔用語解説〕198
SOAP1.1 モード〔用語解説〕198
SOAP 通信基盤〔用語解説〕198
SQL オペレーション定義ファイル〔用語解
説〕198

T

TP1/RPC 受付〔用語解説〕198
TP1 アダプタ〔用語解説〕198
TP1 インバウンドアダプタ〔用語解説〕198
TP1 連携システム〔用語解説〕199

U

uCosminexus Service Architect〔用語解説〕
199
uCosminexus Service Platform〔用語解説〕
199

W

W3C〔用語解説〕199
Web サービス〔用語解説〕199
Web プロジェクトのデプロイ 45
WFA (Work-Flow Architecture)〔用語解
説〕199
WS-R〔用語解説〕199
WSDL〔用語解説〕199
WSDL4J のインストール 28

X

XML スキーマ〔用語解説〕199
XML フォーマット定義ファイル〔用語解説〕
199

あ

アーカイブファイルのダウンロード先 18
 アクティビティ〔用語解説〕199
 アクティビティの定義 95, 119
 アクティビティの配置 94, 117
 アダプタ〔用語解説〕199
 アンインストール〔uCosminexus Service Architect〕175
 アンセットアップ〔MyEclipse〕173
 [アンセットアップ・MyEclipse セットアップ] ダイアログ 174
 [アンセットアップの確認] ページ 174
 [アンセットアップの完了] ページ 175

い

インストール〔uCosminexus Service Architect〕17
 [インストール済みのJRE] ページ 26
 インストールディレクトリについて 2

う

受付処理〔用語解説〕200
 運用環境〔用語解説〕200

え

永続化〔用語解説〕200

お

応答電文〔用語解説〕200

か

開発環境〔用語解説〕200
 開発環境の設定 25
 開発情報のエクスポートとインポート 146
 カスタムアダプタ〔用語解説〕200
 カスタム受付〔用語解説〕200
 カスタム受付フレームワーク〔用語解説〕200
 カスタムファンクション〔用語解説〕200
 カスタムファンクションの準備 149

き

共通フォルダ〔用語解説〕200

く

クラスタ〔用語解説〕201

け

検証〔用語解説〕196

こ

コネクション〔用語解説〕201
 このマニュアルで体験できること 3
 このマニュアルの読み方 2
 [コンパイラー] ページ 27
 コンポーネントの検証とパッケージング 134
 コンポーネントの配備定義 134

さ

サービス〔用語解説〕201
 サービスアダプタ 9
 サービスアダプタ〔用語解説〕201
 サービスアダプタの検証とパッケージング 84
 サービスアダプタの作成 80
 サービスアダプタの定義 80
 サービスアダプタの配備定義 86
 サービス型ソフトウェア〔用語解説〕201
 サービス部品 9
 サービス部品〔用語解説〕201
 サービスリクエスタ 9
 サービスリクエスタ〔用語解説〕201
 在庫管理サービスアダプタの定義 103
 作業フォルダ〔用語解説〕201
 サンプルプログラムの開発の手順 72
 サンプルプログラムの格納場所 178
 サンプルプログラムのシステム構成 6
 サンプルプログラムの処理の内容 10
 サンプルプログラムを構成している要素 9

し

システム構成定義〔用語解説〕 202
システム構成定義のエクスポートとインポート 78
実行環境〔用語解説〕 202
実行環境の設定 30
実行履歴〔用語解説〕 202
商品手配サンプルプログラム〔動作手順〕 63
商品手配サンプルプログラムの構成 183
商品手配サンプルプログラムのシステム構成 8
商品手配サンプルプログラムの処理詳細 12
商品手配サンプルプログラムの処理内容 12
商品手配システムの開発 103
商品手配システムのデバッグ 135
商品手配ビジネスプロセスの定義 109
処理が中断された場合の対処〔MyEclipse セットアップ機能〕 24

せ

〔セットアップ・MyEclipse セットアップ〕ダイアログ 20
セットアップログ 187
セットアップログの確認方法 24

そ

関連セット〔用語解説〕 202

た

ダウンロードするアーカイブファイルの種類 18

て

定義内容 52
データ変換〔用語解説〕 202
データ変換基盤〔用語解説〕 195
データ変換定義〔用語解説〕 202
テスト環境の起動 34
テスト環境の構築 31
テスト環境の停止 70
電文フォーマット〔用語解説〕 202

電文フォーマット定義ファイル〔用語解説〕 202

と

〔動作環境の選択〕ページ 20

に

任意の Java プログラムを使用したデータ変換の定義 148

は

配送受付サービスアダプタの定義 106
バイナリフォーマット定義ファイル〔用語解説〕 203
配備〔用語解説〕 203
配備定義〔用語解説〕 203
パッケージング〔用語解説〕 203

ひ

ビジネスプロセス 9
ビジネスプロセス〔用語解説〕 203
ビジネスプロセス基盤〔用語解説〕 195
ビジネスプロセスの作成 87
ビジネスプロセスの追加 87, 110
ビジネスプロセスの定義 87
標準受付 9
標準受付〔用語解説〕 203
標準電文〔用語解説〕 203
標準電文フォーマット〔用語解説〕 203

ふ

ファイルアダプタ〔用語解説〕 203
ファイル操作アダプタ〔用語解説〕 203
ファンクション〔用語解説〕 203
プロジェクトの削除 67
プロセスインスタンス〔用語解説〕 204

へ

変換ファンクション〔用語解説〕 204

変換ファンクション定義ファイル〔用語解説〕 204
変換ファンクション定義ファイルの作成 150
変数の設定 90, 114

ま

マッピング〔用語解説〕 204

め

メッセージング基盤〔用語解説〕 196

ゆ

ユーザ定義受付 9
ユーザ定義受付〔用語解説〕 204
ユーザ定義受付の追加 112

よ

要求電文〔用語解説〕 204

ら

ライブラリーパスの設定 35
ランゲージパックの使用について 17

り

リポジトリ〔用語解説〕 204

ろ

ローカル変数情報の出力の設定 27