

HITACHI
Inspire the Next



Cosminexus
コズミネクサス

サービスプラットフォーム
サンプルの実行と開発

SOAP連携編

2015.
December **12**

はじめに

■ 対象製品

uCosminexus Service Architect (09-71 以降)

■ 商標類

HITACHI, Cosminexus, uCosminexus は、株式会社 日立製作所の商標または登録商標です。

Microsoft および 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。

Oracle と 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です。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

■ 適用 OS

Microsoft® Windows® 7 Professional

Microsoft® Windows® 7 Enterprise

Microsoft® Windows® 7 Ultimate

Microsoft® Windows® 8 Pro

Microsoft® Windows® 8 Pro

Microsoft® Windows® 8.1 Pro

Microsoft® Windows® 8.1 Enterprise

■ マイクロソフト製品の表記について

本書の適用 OS を Windows と表記します。

なお、32 ビット版の Windows を Windows x86 と表記することがあります。また、64 ビット版の Windows を Windows x64 と表記することがあります。

■ 発行元

株式会社日立製作所 情報・通信システム社 ITプラットフォーム事業本部

■ 著作権

All Rights Reserved. Copyright (C) 2015, Hitachi, Ltd.

目次

はじめに	1
第 1 編 実行編	5
1 サンプルプログラムをご使用になる前に	5
1.1 このホワイトペーパーの読み方	6
1.2 このホワイトペーパーで体験できること	8
2 サンプルプログラムの紹介	9
2.1 サービスプラットフォームを利用する一般的なシステム構成	10
2.2 サンプルプログラムのシステム構成	11
2.3 サンプルプログラムの処理内容	13
3 サンプルプログラムの開発・テスト環境を構築する	15
3.1 サンプルプログラムの開発・テスト環境を構築する流れ	16
3.2 Service Architect のインストール	17
3.3 Eclipse の準備	19
3.3.1 Eclipse をインストールするための準備	19
3.3.2 Eclipse セットアップ機能を使用した開発環境の構築	19
3.3.3 JDK の確認	22
3.3.4 ローカル変数情報の出力の設定	25
3.4 テスト環境の構築	27
3.4.1 HCSC 簡易環境セットアップ	27
3.4.2 テスト環境の起動	29
3.5 開発環境とテスト環境の連動設定	30
3.5.1 開発環境の SOAP モードの設定	30
3.5.2 HCSCTE プロジェクトの作成	31
3.5.3 テスト環境のシステム構成定義を開発環境に取り込むための設定	34
4 サンプルプログラムを実行する準備	38
4.1 ビジネスプロセス・サービスアダプタを準備する	39

4.1.1 HCSC コンポーネントのインポート	39
4.1.2 HCSC コンポーネントの配備と開始	39
4.1.3 HCSC コンポーネントの配備内容の確認	40
4.2 連携する SOAP サービスリクエスタ・サービス部品を準備する	42
4.2.1 Eclipse から J2EE サーバを操作するための設定	42
4.2.2 Eclipse プロジェクトのインポート	48
4.2.3 Eclipse プロジェクトのデプロイ	53
5 サンプルプログラムを実行する	57
<hr/>	
5.1 サンプルプログラムの実行	58
5.2 商品在庫数の初期化方法	61
 第 2 編 開発編	 62
6 システムの開発を体験する	62
<hr/>	
6.1 商品手配システムの開発の手順	63
6.2 商品手配システムの開発	65
6.2.1 InventoryManagement サービスアダプタの定義	65
6.2.2 DeliveryReceipt サービスアダプタの定義	67
6.2.3 ProductStock ビジネスプロセスの定義	69
6.2.4 コンポーネントの検証とパッケージング	93
6.3 ビジネスプロセスのデバッグ	96
6.3.1 サンプルプログラムが提供するサービス部品を利用するデバッグ	96
6.3.2 サービスのエミュレーション機能を利用する場合	105
6.4 開発した商品手配システムを実行する準備	108
 7 サンプルプログラムの環境を削除する	 109
<hr/>	
7.1 プロジェクトの削除	110
7.1.1 Eclipse プロジェクトのアンデプロイと削除	110
7.1.2 HCSC サーバへ配備した定義内容の削除	112
7.2 テスト環境の停止	113
7.3 アンセットアップとアンインストール	114
7.3.1 テスト環境のアンセットアップ	114
7.3.2 Eclipse のアンセットアップ	114
7.3.3 Service Architect のアンインストール	117

A サンプルプログラムのファイルの構成	119
B Eclipse セットアップ機能実行時の情報の採取	122
C 用語解説	123

1 サンプルプログラムをご使用になる前に

この章では、このホワイトペーパーの読み方、およびこのホワイトペーパーで体験できる内容について説明します。

本章の構成

- 1.1 このホワイトペーパーの読み方
- 1.2 このホワイトペーパーで体験できること

1.1 このホワイトペーパーの読み方

このホワイトペーパーは、実際にマシンを操作しながら、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、このホワイトペーパーで紹介するサンプルプログラムの内容、およびサンプルプログラムの構成要素について説明しています。

3章では、サービスプラットフォームの開発・テスト環境を構築する手順について説明しています。

4章では、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。

5章では、4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。

6章では、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。

7章では、3章および4章で準備したサービスプラットフォームの環境、およびサンプルプログラムの環境を削除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。

画面や操作の説明で使用している記号について

ホワイトペーパー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
< >	<>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。

使用している構文要素記号について

ホワイトペーパーで使用している構文要素の種類と定義を次に示します。

種類	定義
英字	A~Z a~z
英数字	A~Z a~z 0~9
文字列	任意の文字の配列

インストールディレクトリについて

サービスプラットフォーム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、このホワイトペーパーに記載されている「<サービスプラットフォーム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。

ランゲージパックの使用について

このホワイトペーパーでは、Eclipse Babel Project が提供する BABEL 日本語ランゲージパック

2014/12/23 版を適用した Eclipse メニュー表記を使用しています。使用するランゲージパックのバージョンによっては、ホワイトペーパー内の表記と異なる場合があります。

1.2 このホワイトペーパーで体験できること

このホワイトペーパーでは、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**

サービスプラットフォームを使うための基本的な設定を体験できます。具体的には、製品のインストール、開発環境のセットアップ、およびテスト環境のセットアップです。この章で体験することは、一般的なシステム開発時の設定手順としても使用できます。

- **4章で体験できること**

サンプルプログラムを動作させるための準備を体験できます。このホワイトペーパーでは、サービスプラットフォームを利用した SOAP プロトコル連携のサンプルプログラムを紹介します。この章では、サービスプラットフォームの要素（ビジネスプロセス・サービスアダプタ）の準備と、連携する SOAP サービスリクエスト・サービス部品の準備を体験できます。

- **5章で体験できること**

3章および4章で準備した環境を使用して、サンプルプログラムを実行する操作を体験できます。

- **6章で体験できること**

手順どおりに操作を進めることで、サービスプラットフォームの開発環境を使用して、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、ビジネスプロセス、サービスアダプタ、データ変換定義など、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。

- **7章で体験できること**

3章および4章で準備したプロジェクトの削除、テスト環境の停止、開発環境のアンセットアップ、および製品のアンインストールを体験できます。

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、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。

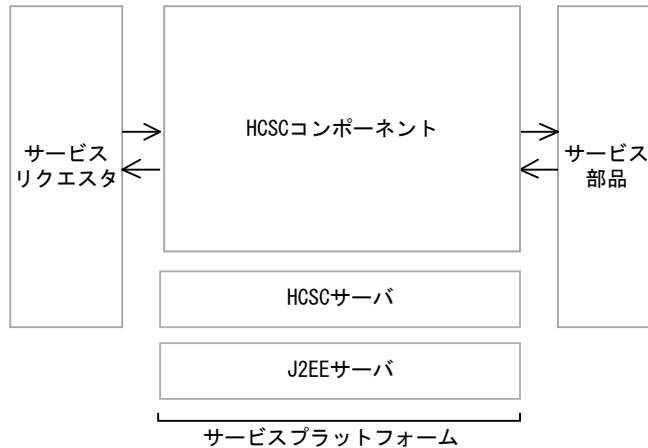
本章の構成

- 2.1 サービスプラットフォームを利用する一般的なシステム構成
- 2.2 サンプルプログラムのシステム構成
- 2.3 サンプルプログラムの処理内容

2.1 サービスプラットフォームを利用する一般的なシステム構成

サービスプラットフォームを利用する一般的なシステム構成について説明します。

サービスプラットフォームを利用する一般的なシステム構成を次の図に示します。HCSC コンポーネントはサービスリクエスタからの要求を受け取り、サービス部品を呼び出します。また、サービス部品の処理結果が HCSC コンポーネントを経由し、サービスリクエスタへ返します。



(凡例)

—→ : サービス部品を呼び出す要求・応答の流れ

図 2-1 一般的なシステム構成

一般的なシステム構成要素が、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。

● サービスリクエスタ

サービスを呼び出すためのクライアントアプリケーションです。サービスプラットフォームを使用する場合、サービスリクエスタは HCSC サーバ上の HCSC コンポーネントへ要求（要求電文）を送ります。

● HCSC コンポーネント

サービスリクエスタからの要求を受け取り、定義した業務の流れや条件に従って要求を受付、サービスを呼び出すためのものです。受付、ビジネスプロセス、サービスアダプタの総称です。

● サービス部品

サービスプラットフォームで接続する連携先サービスのことです。

● HCSC サーバ

ビジネスプロセスやサービスアダプタを管理するサーバ機能です。サービスプラットフォームの中にあります。

● J2EE サーバ

J2EE アプリケーション（JSP、サーブレット、Enterprise Bean 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。HCSC サーバを起動するのに必要です。

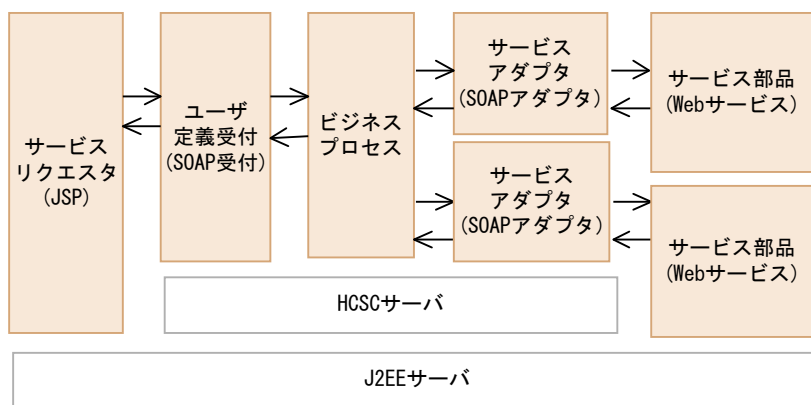
2.2 サンプルプログラムのシステム構成

サンプルプログラムのシステム構成について説明します。

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。



(凡例)

- : サービス部品を呼び出す要求・応答の流れ
- : サンプルプログラムの提供範囲

図 2-2 商品手配サンプルプログラムのシステム構成

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。

● サービスリクエスタ

ビジネスプロセスを呼び出すユーザ定義受付に対してサービスを呼び出す要求（要求電文）を送ります。本サンプルのサービスリクエスタは、JSP（JavaServer Pages）で Web ページを生成し、SOAP プロトコルにより通信を行います。

● ユーザ定義受付

サービスリクエスタからのサービス部品の要求電文を受け付け、ビジネスプロセスを呼び出すための機能（インターフェース）です。ユーザが任意のインターフェースを定義できます。本サンプルプログラムはユーザ定義受付の SOAP 受付を使用します。

● ビジネスプロセス

ビジネスプロセスは、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。本サンプルプログラムでは、サービス部品ごとに異なる電文フォーマットを変換したり、サービス部品からの応答によって処理を分岐したりしています。

- **サービスアダプタ**

サービスアダプタは、ビジネスプロセスからの要求を受け付けて、サービス部品を呼び出すためのものです。本サンプルプログラムは SOAP プロトコルで通信を行うため、SOAP アダプタを使用します。

- **サービス部品**

サービスアダプタから呼び出される Web サービスです。SOAP プロトコルにより通信を行います。

注意事項

一般的なシステムの場合は、サービスリクエスタ、サービス部品をデプロイ・実行するサーバとサービスプラットフォームの J2EE サーバは異なるものでなければなりません。

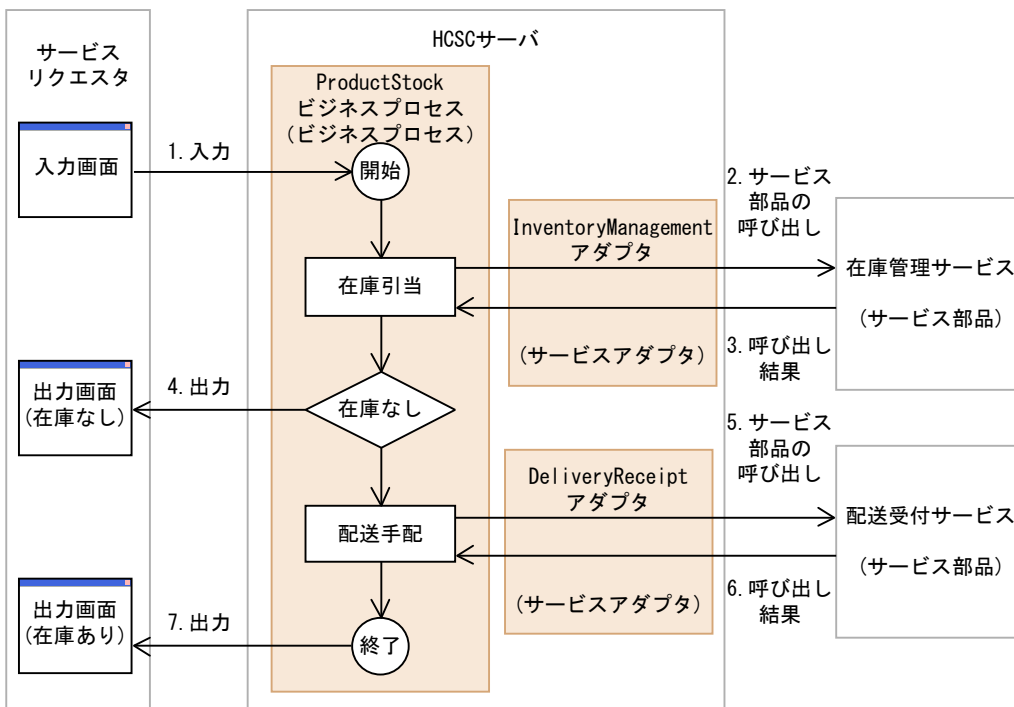
今回のサンプルは便宜上、サービスリクエスタとサービス部品をサービスプラットフォームと同じ J2EE サーバ上でデプロイ・実行します。

2.3 サンプルプログラムの処理内容

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。

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。



(凡例)

→ : サービス部品を呼び出す要求・応答の流れ

■ : 「6. 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

図 2-3 商品手配サンプルプログラムの処理詳細

表 2-1 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	JSP サーブレット	<ul style="list-style-type: none"> 入力画面を表示し、ProductStock 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。
ProductStock 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、InventoryManagement アダプタを介して在庫管理サービスを呼び出します。

		<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， DeliveryReceipt アダプタを介して配送受付サービス呼び出します。 在庫管理サービスを呼び出した結果を InventoryManagement アダプタを介して取得し， サービスリクエスタへ返します。 配送受付サービスを呼び出した結果を DeliveryReceipt アダプタを介して取得し， サービスリクエスタへ返します。
InventoryManagement アダプタ	サービスアダプタ	<ul style="list-style-type: none"> ProductStock ビジネスプロセスからの要求を受け取り， 在庫管理サービスを呼び出します。 在庫管理サービスの応答を ProductStock ビジネスプロセスへ返します。
DeliveryReceipt アダプタ	サービスアダプタ	<ul style="list-style-type: none"> ProductStock ビジネスプロセスからの要求を受け取り， 配送受付サービスを呼び出します。 配送受付サービスの応答を ProductStock ビジネスプロセスへ返します。
在庫管理サービス	サービス部品	<ul style="list-style-type: none"> ProductStock 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， ProductStock 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	サービス部品	<ul style="list-style-type: none"> ProductStock 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開発・テスト環境 を構築する

この章では、サンプルプログラムの開発・テスト環境の構築について説明します。

本章の構成

- 3.1 サンプルプログラムの開発・テスト環境を構築する流れ
- 3.2 Service Architect のインストール
- 3.3 Eclipse の準備
- 3.4 テスト環境の構築
- 3.5 開発環境とテスト環境の連動設定

3.1 サンプルプログラムの開発・テスト環境を構築する流れ

サンプルプログラムの開発・テスト環境を構築する手順を次の図に示します。

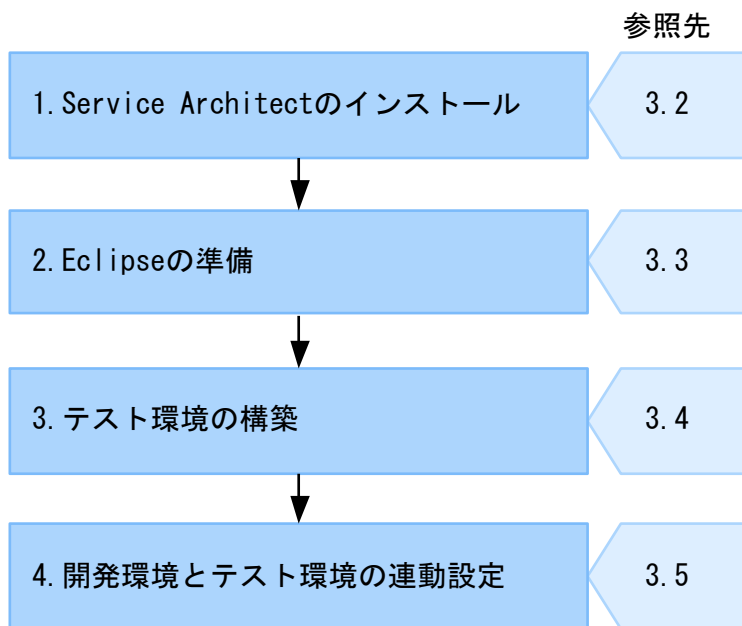


図 3-1 サンプルプログラムの開発・テスト環境を構築する手順

1. Service Architect のインストール

インストーラを使用して、Service Architect をインストールします。詳細については、「3.2 Service Architect のインストール」を参照してください。

2. Eclipse の準備

Eclipse をインストールして開発環境を構築します。Eclipse のインストールは、Service Architect が提供する Eclipse セットアップ機能を利用します。詳細については、「3.3 Eclipse の準備」を参照してください。

3. テスト環境の構築

Service Architect の HCSC 簡易セットアップ機能を利用し、テスト環境を構築します。詳細については、「3.4 テスト環境の構築」を参照してください。

4. 開発環境とテスト環境の連動設定

テスト環境構築後のシステム構成を、開発環境に反映させる必要があります。詳細については、「3.5 開発環境とテスト環境の連動設定」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 Service Architect のインストール

Service Architect のインストール手順を次に示します。

注意事項

Service Architect を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おいてください。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. インストール CD-ROM を、CD-ROM ドライブにセットします。
[日立総合インストーラ] ダイアログに、「選択されたソフトウェアをインストールします。」と表示されます。
[日立総合インストーラ] ダイアログが表示されない場合、エクスプローラを使用して、CD-ROM ディレクトリの「HCD_INST.EXE」をダブルクリックしてください。
2. uCosminexus Service Architect を選択した状態で、[インストール実行] ボタンをクリックします。
[インストール処理開始の確認 - 日立総合インストーラ] ダイアログに、「インストールを開始します。よろしいですか?」と表示されます。
3. [OK] ボタンをクリックします。
[uCosminexus Service Architect セットアッププログラムへようこそ] ページが表示されます。
4. [次へ] ボタンをクリックします。
[インストール先の選択] ページが表示されます。
5. 必要に応じてインストール先ディレクトリを選択して、[次へ] ボタンをクリックします。

ポイント

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

[機能の選択] ページが表示されます。



6. [標準 - このセットアップを推奨します。] の左にあるボタンをクリックします。
[ユーザ情報] ページが表示されます。



7. [ユーザ名] および [会社名] を入力して [次へ] ボタンをクリックします。
[プログラム フォルダの選択] ページが表示されます。
8. 必要に応じてプログラムフォルダ名を変更して、[次へ] ボタンをクリックします。
[インストールの開始] ページが表示されます。
9. 設定した内容を確認して、問題がなければ [次へ] ボタンをクリックします。
インストールが開始されます。インストールが完了すると、[セットアップの完了] ダイアログが表示されます。インストールは数分掛かる場合があります。
10. [完了] ボタンをクリックします。
OS を再起動するかどうかを確認する画面が表示されます。
11. [はい] ボタンをクリックします。
OS が再起動し、Service Architect のインストールが完了します。

これで、Service Architect はインストールされました。

3.3 Eclipse の準備

ここでは、Eclipse セットアップ機能を使用した開発環境の構築について説明します。また、JDK の確認やローカル変数情報の出力の設定も説明します。

3.3.1 Eclipse をインストールするための準備

Eclipse をインストールするための環境を準備します。Eclipse のアーカイブファイルを準備する手順を次に示します。

1. Eclipse のアーカイブファイルを入手します。

次に示す Eclipse のアーカイブファイルを Service Architect に同梱されている添付品または Eclipse.org のダウンロードサイトから入手してください。

Windows x86 (WOW64 環境を含む) の場合

Eclipse IDE for Java EE Developers 4.4.1a (eclipse-jee-luna-SR1a-win32.zip)

Windows x64 の場合

Eclipse IDE for Java EE Developers 4.4.1a (eclipse-jee-luna-SR1a-win32-x86_64.zip)

2. Eclipse のアーカイブファイルを格納します。

入手した Eclipse のアーカイブファイルを次のディレクトリへ格納します。

<サービスプラットフォームのインストールディレクトリ>\ADP\archives

参考

添付品の Eclipse ランゲージパックを Eclipse のアーカイブファイルと同じディレクトリに格納すると、Eclipse セットアップ時に Eclipse を日本語化できます。

これで、Eclipse をインストールする準備ができました。

3.3.2 Eclipse セットアップ機能を使用した開発環境の構築

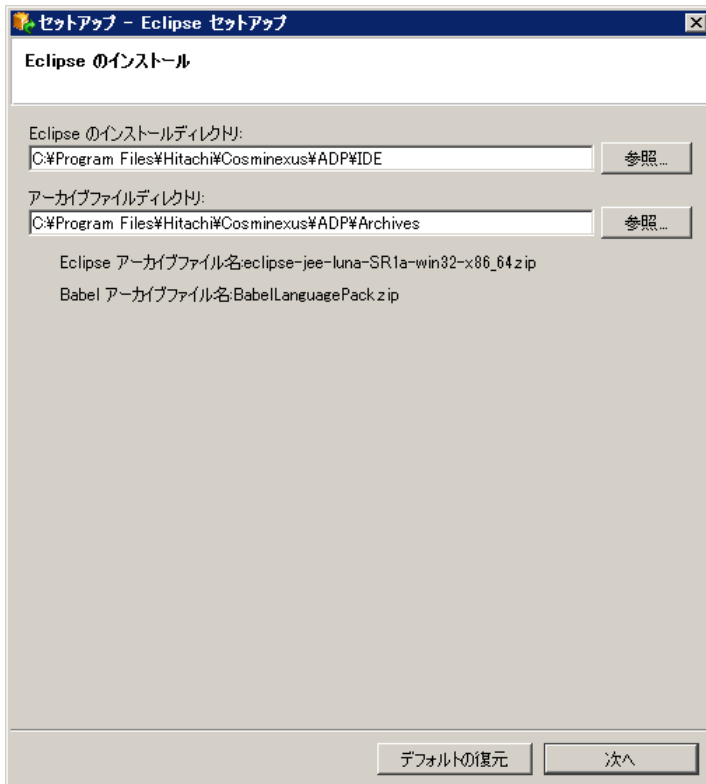
Eclipse セットアップ機能を使用して開発環境を構築します。開発環境を構築する手順を次に示します。

1. 管理者権限を持つユーザ（または管理者特権モード）でマシンを起動します。
2. Windows の [スタート] メニューから、[プログラム] - [Cosminexus] - [環境構築] - [Eclipse セットアップ] を選択します。

Eclipse のアーカイブファイルを準備したかどうかを確認するダイアログが表示されます。



3. 確認ダイアログで、[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。



4. [Eclipse のインストール] ページで、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ディレクトリを指定して、[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。



5. [セットアップの確認] ページでセットアップ内容を確認して, [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, セットアップが実行されます。セットアップが完了すると [セットアップの完了] ページが表示されます。

Eclipse セットアップは数分掛かる場合があります。



6. [セットアップの完了] ページで, [終了] ボタンをクリックします。
[セットアップ - Eclipse セットアップ] ダイアログが閉じて, Eclipse のセットアップが完了します。
デスクトップに Eclipse のショートカットが生成されます。

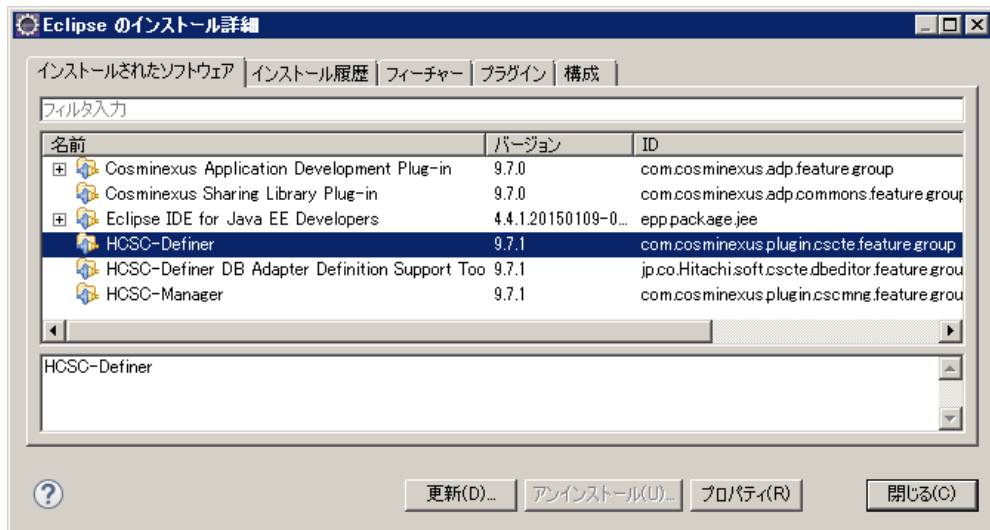
これで、Eclipse 環境が構築できました。

正しくセットアップされたことを確認します。

7. Eclipse を起動します。

8. Eclipse のメニューから、[ヘルプ] - [Eclipse について] - [インストール詳細] を選択し、[Eclipse のインストール詳細] ダイアログを表示します。

HCSC-Definer のバージョン番号がインストールした Service Architect のバージョンになっていることを確認します。



参考：セットアップログの確認方法

Eclipse セットアップ機能を実行したときのログは、Eclipse 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 Eclipse 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。

3.3.3 JDK の確認

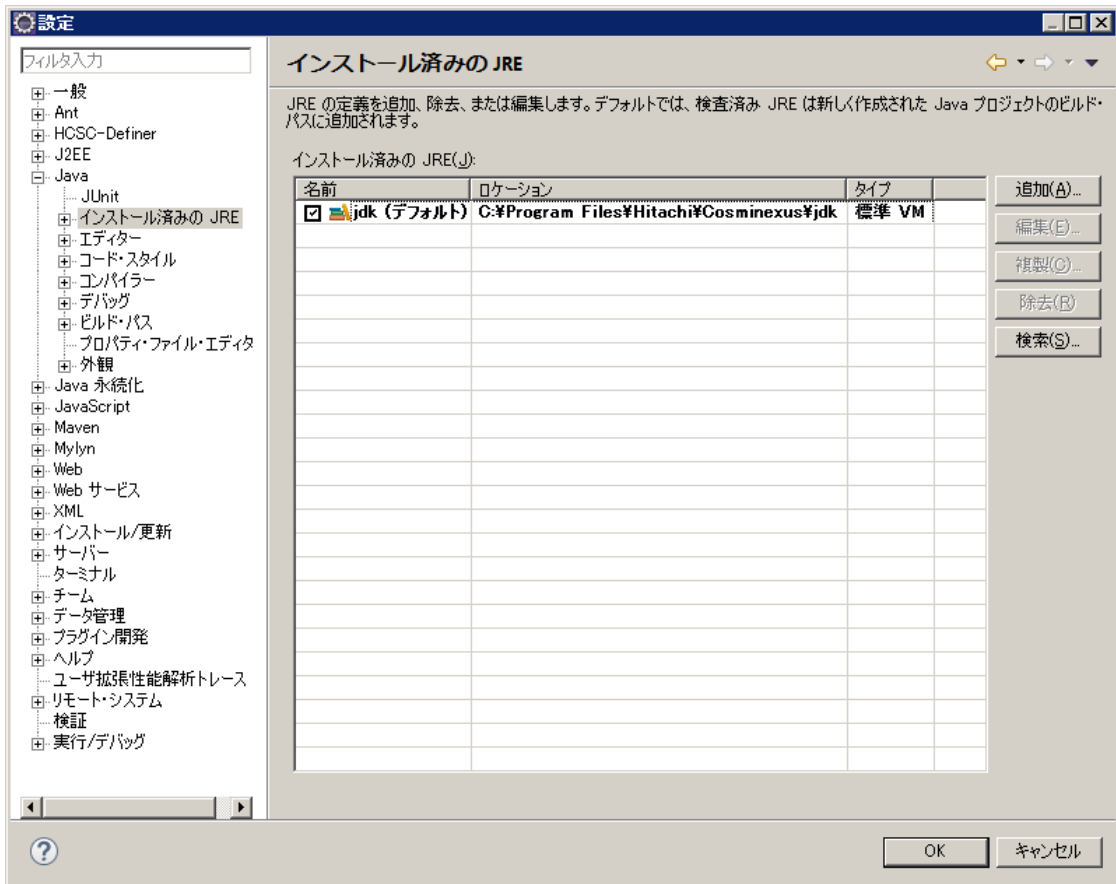
Eclipse 上で開発するとき使用する JDK が Service Architect で提供されている JDK かどうかを確認します。確認手順を次に示します。

1. Eclipse のメニューから [ウィンドウ] - [設定] を選択します。

[設定] ダイアログが表示されます。

2. 左ペインで [Java] - [インストール済みの JRE] を選択します。

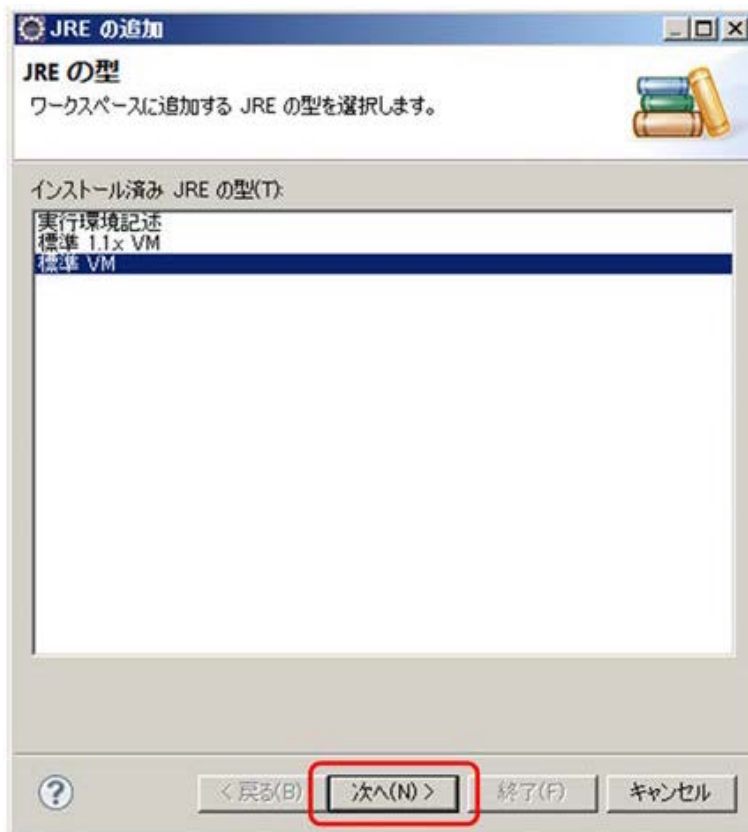
右ペインに [インストール済みの JRE] ページが表示されます。



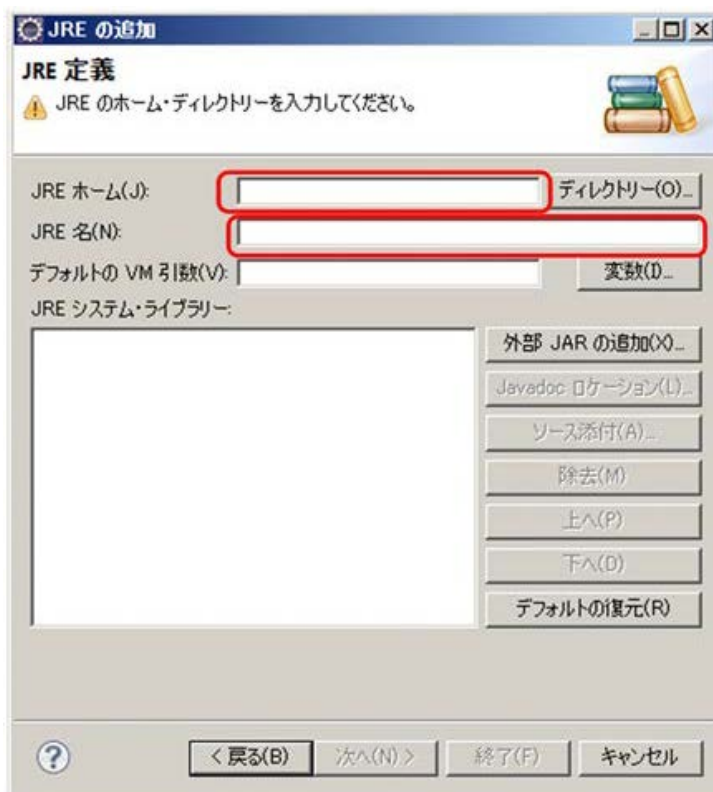
3. [インストール済みの JRE] が Service Architect で提供されている JDK かどうかを確認します。
 [ロケーション] に次のパスが表示されているかどうかを確認します。
 <サービスプラットフォームのインストールディレクトリ>\jdk

パスが表示されていない場合

[追加] ボタンをクリックします。 [JRE の型] ページが表示されます。



「標準 VM」を選択し、[次へ] ボタンをクリックします。[JRE 定義] ページが表示されます。



[JRE ホーム] に「<サービスプラットフォームのインストールディレクトリ>%jdk」を、[JRE 名] に「jdk」を選択し、[終了] ボタンをクリックします。設定後、[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、<サービスプラットフォーム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。

3.3.4 ローカル変数情報の出力の設定

Eclipse のコンパイラーの設定によって、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。

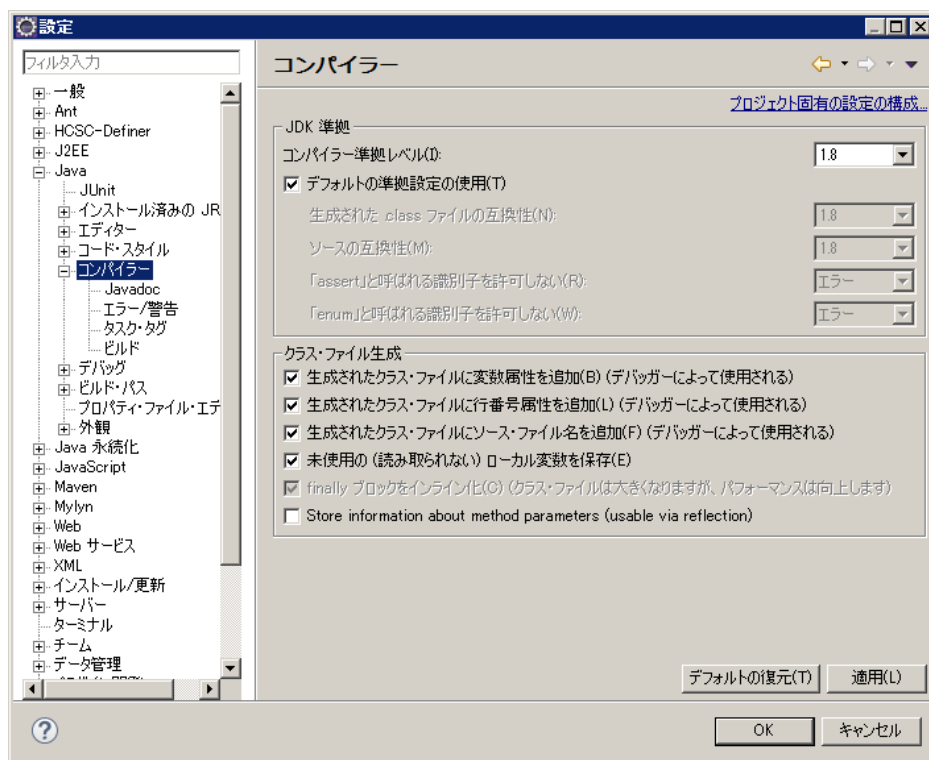
ローカル変数情報をスタックトレースに出力する場合のコンパイラーの設定手順を次に示します。

1. Eclipse のメニューから [ウィンドウ] - [設定] を選択します。

[設定] ダイアログが表示されます。

2. 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	設定値
JDK 準拠 コンパイラー準拠レベル	「1.8」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。

必要に応じて、次の項目を指定してください。

項目名		設定値
JDK 準拠	デフォルトの準拠設定の使用	コンパイラー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 [コンパイラー準拠レベル] 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 テスト環境の構築

3.4.1 HCSC 簡易環境セットアップ

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. [スタート] メニューの [プログラム] から [Cosminexus[※]] - [環境構築] - [テスト環境セットアップ] を選択します。

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。

注※

プログラムフォルダ名を変更している場合は、変更したプログラムフォルダから選択してください。

3. [DB あり/RM なしモデル] ラジオボタンを選択します。
4. [SOAP1.1/1.2 併用モード] ラジオボタンを選択します。

The screenshot shows the 'HCSC Easy Setup' dialog box with the 'Main' tab selected. The 'Database Model' section has three radio buttons: 'DB with/RM none model' (selected), 'DB/RM none model', and 'DB/RM with model'. The 'HCSC Server Mode' section has two radio buttons: 'SOAP 1.1 mode' and 'SOAP 1.1/1.2 concurrent mode' (selected). Below these are several input fields for port numbers, including 'Web Service/MDB(WS-R) Reception Port Number' (80), 'Session Bean Reception Port Number' (900), 'MDB(DB Queue) Reception Port Number' (20351), 'Replication Confirmation Port Number' (23152), 'Simple Web Service Port Number (Internal Management)' (8080), 'Management Server' section with 'HCSC Server Operation Port Number' (28099), 'Argument Server Operation Port Number' (28080), 'Termination Request Reception Port Number (Internal Management)' (28005), 'Internal Communication Port Number (Internal Management)' (28009), and 'In-process Naming Service Port Number (Internal Management)' (28900), and 'Operation Management Agent' section with 'Agent Reception Port Number (Internal Management)' (20295). A 'Select' button is next to the database path field. At the bottom are 'Set Up' and 'Cancel Set Up' buttons, and a 'Console' area.

5. [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。
6. [HCSC 本番環境簡易セットアップ名称] ラジオボタンを選択します。
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。
 - [HCSC 本番環境簡易セットアップ名称] ラジオボタン

- 論理 J2EE サーバ名称 = J2EEServer
- 論理 PRF 名称 = PRF
- クラスタ名称 = Cluster
- HCSC サーバ名称 = HCSC
- Manager 名称 = Manager

The screenshot shows the 'HCSC簡易セットアップ' window with the 'サーバ名称' tab selected. The 'HCSC本番環境簡易セットアップ名称' radio button is checked. The form contains the following fields and values:

- 論理J2EEサーバ名称: J2EEServer
- 論理PRF名称: PRF
- クラスタ名称: Cluster
- HCSCサーバ名称: HCSC
- Manager名称: Manager

Buttons for 'セットアップ' and 'アンセットアップ' are visible at the bottom right. A 'コンソール' (Console) area is at the bottom.

7. [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。

注意事項

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。

8. 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。

3.4.2 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. [スタート] メニューの [プログラム] から [Cosminexus[※]] - [データベース起動] を選択して、テスト環境の組み込みデータベースを起動します。
データベースを起動した場合、「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、すでに組み込みデータベースが起動されているため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. [スタート] メニューの [プログラム] から [Cosminexus[※]] - [テストサーバ起動] を選択して、テスト環境の Performance Tracer, J2EE サーバ, および, 標準受付とユーザ定義受付を含む HCSC サーバを起動します。

注※

プログラムフォルダ名を変更している場合は、変更したプログラムフォルダから選択してください。

注意事項

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、「7.2 テスト環境の停止」を参照してください。

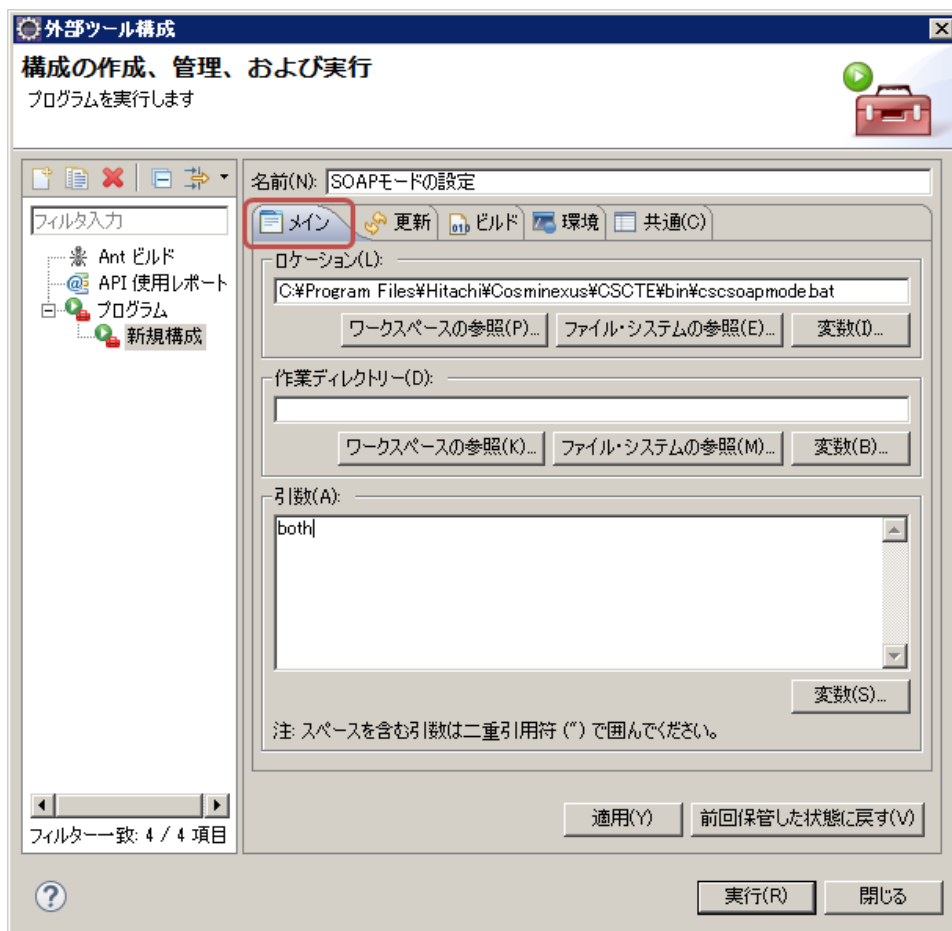
3.5 開発環境とテスト環境の連動設定

3.5.1 開発環境の SOAP モードの設定

開発環境の SOAP モードは「3.4.1 HCSC 簡易環境セットアップ」で設定したテスト環境の SOAP モードと一致させる必要があります。

開発環境の SOAP モードの設定方法を次に示します。

1. Eclipse のメニューから、[ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. ダイアログ左側のツリービューで [HCSC-Definer] を選択します。
ダイアログ右側の [現在の SOAP モード] に使用している SOAP モードが表示されます。現在の SOAP モードは SOAP1.1/1.2 併用モードかどうかを確認してください。現在の SOAP モードが SOAP1.1 モードの場合だけ以降の手順を実施して SOAP モードを SOAP1.1/1.2 併用モードに設定してください。
3. Eclipse のメニューから [実行] - [外部ツール] - [外部ツールの構成] を選択します。
[外部ツール構成] ダイアログが表示されます。
4. 左ペインの [プログラム] を右クリックし、[新規] を選択します。
[構成の作成、管理、および実行] ページが表示されます。



[メイン] タブを選択して、次の内容を入力します。

項目名	設定値
名前	任意の名称を指定します。 ここでは、「SOAP モードの設定」と指定します。
ロケーション	<code>\${env_var: COSMINEXUS_HOME}¥CSCTE¥bin¥cscsoapmode.bat</code> または [ファイルシステムの参照] ボタンから次のファイルを選択します。 <サービスプラットフォームのインストールディレクトリ>¥CSCTE¥bin¥cscsoapmode.bat
引数	both

5. [実行] ボタンをクリックします。

コマンドが登録、および実行され、コンソールビューに実行結果が表示されます。

6. Eclipse を再起動します。

3.5.2 HCSCTE プロジェクトの作成

開発環境で HCSC コンポーネントを作成・編集するためには、HCSCTE プロジェクトが必要となります。HCSCTE プロジェクトを作成する手順を次に示します。

注意事項

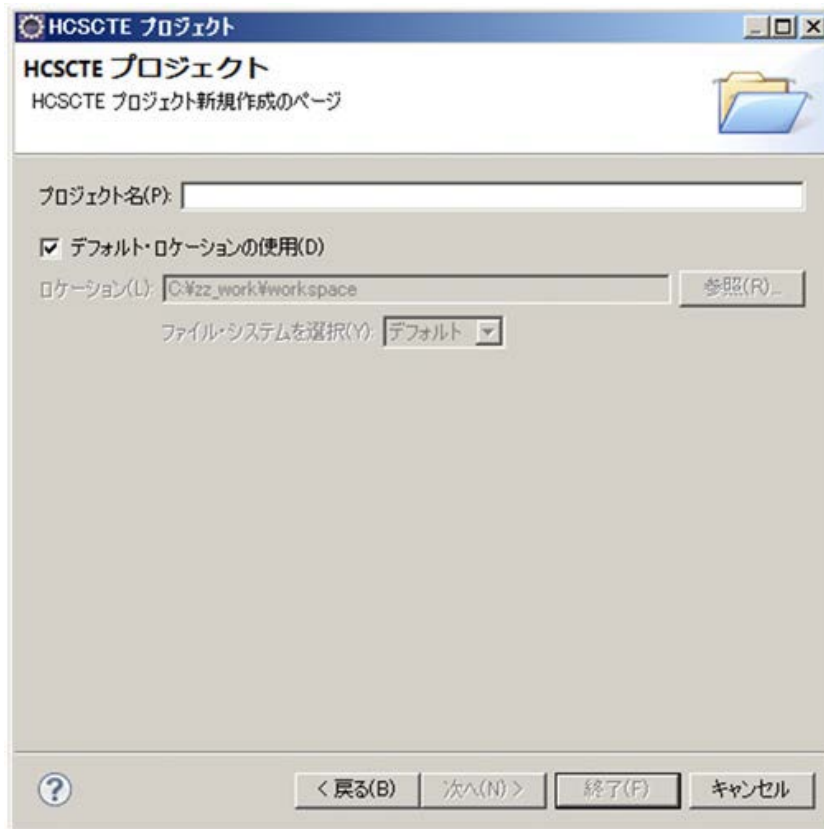
HCSCTE プロジェクトは、1つのプログラムにつき1つ作成してください。複数のプログラムを開発する場合、各プログラムの HCSCTE プロジェクトのワークスペースは分ける必要があります。同じワークスペースに HCSCTE プロジェクトを複数作成すると、プログラムが正しく動作しません。

1. 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。

2. [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。



3. 次の項目を設定して、[次へ] ボタンをクリックします。

プロジェクト名

任意の名称を指定します。ここでは「HCSCTE」とします。

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。



4. 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。ここでは、

「C:¥work¥ProductStock¥repository」とします。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。

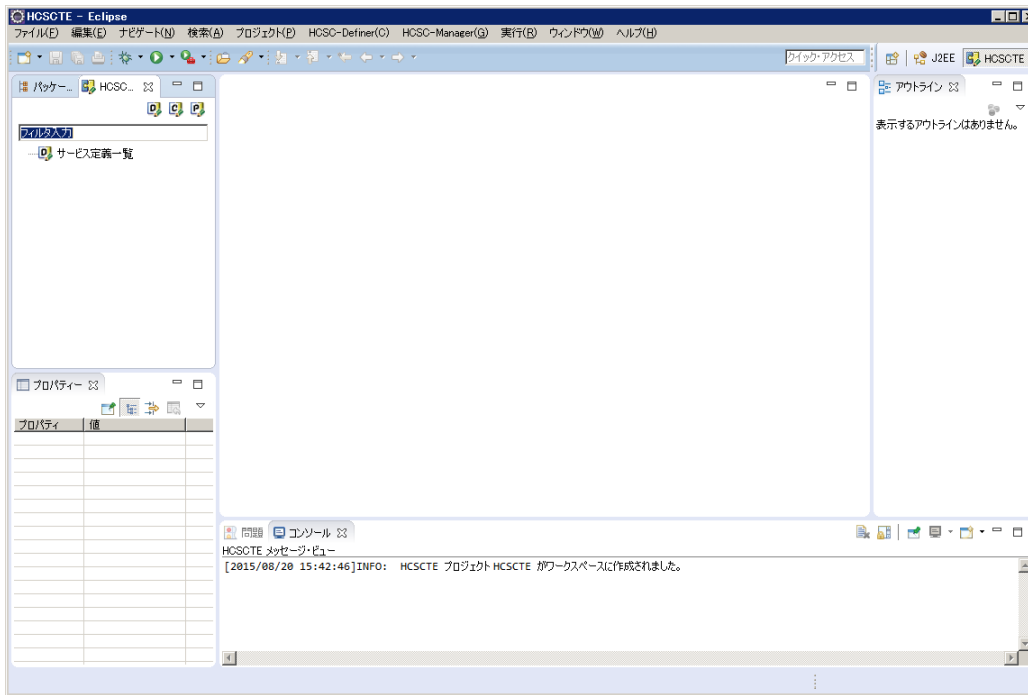
- リポジトリディレクトリのパスとプロジェクトのパスには、同じパスを指定しないでください。
- パスは、絶対パスで指定してください。
- パスの長さは、正規化された絶対パスでチェックされます。

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。

「関連付けられたパースペクティブを開きますか?」という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。

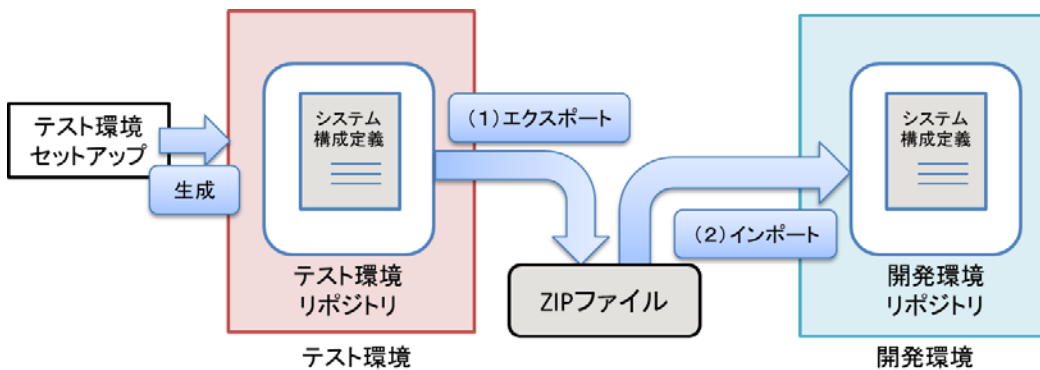
HCSCTE のプロジェクトが作成され、HCSCTE のプロジェクトのパースペクティブが起動します。



3.5.3 テスト環境のシステム構成定義を開発環境に取り込むための設定

開発環境からテスト環境を操作するためには、テスト環境のリポジトリにあるシステム構成定義と開発環境のリポジトリにあるシステム構成定義を同一にする必要があります。

テスト環境のシステム構成定義を開発環境に取り込むための手順を次の図に示します。テスト環境の構築で作成されるシステム構成定義を含むリポジトリをエクスポートし、開発環境にインポートします。



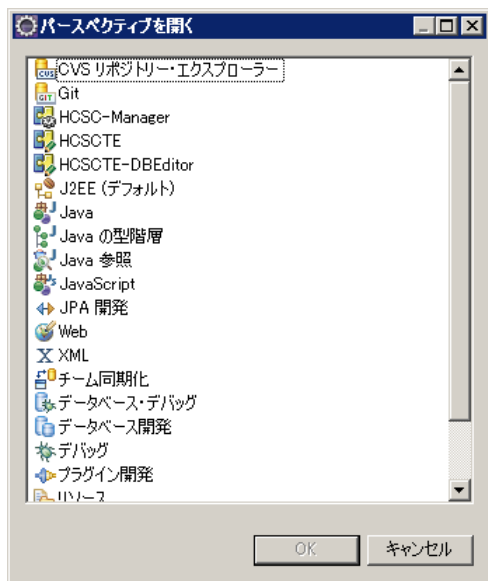
(1) テスト環境からのシステム構成定義のエクスポート

テスト環境のシステム構成定義を開発環境に取り込むため、テスト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ます。エクスポートの手順を次に示します。

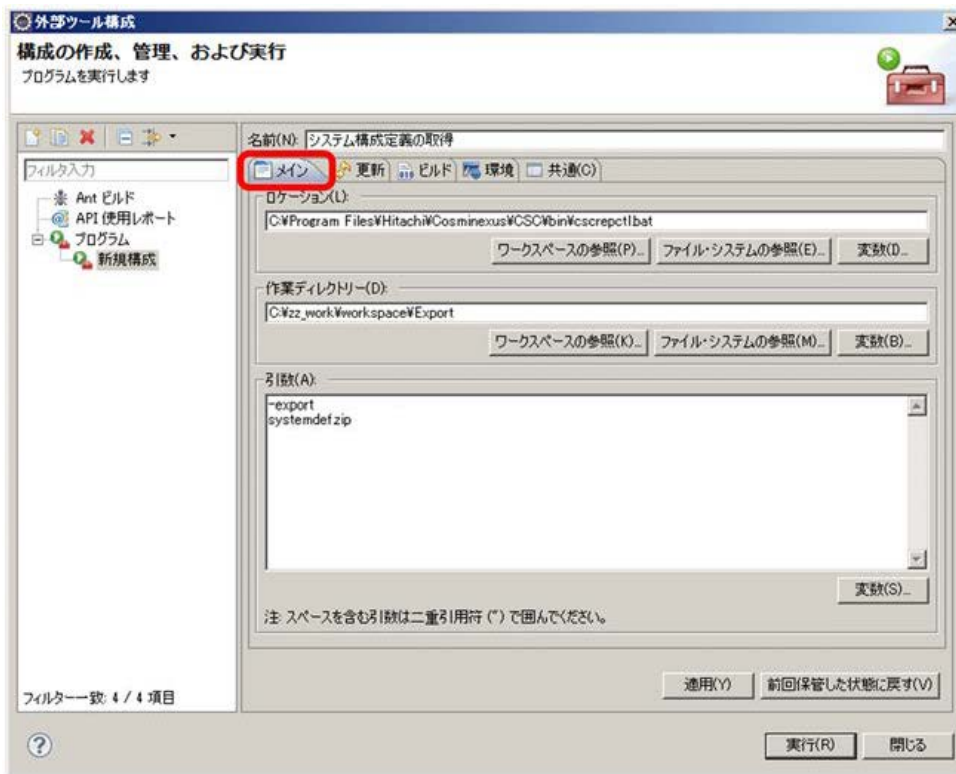
ポイント

システム構成定義には HCSC コンポーネントの配備情報も含まれます。HCSC コンポーネントが配備されていない状態のテスト環境リポジトリを保存しておくため、テスト環境構築後に必ず1回のみ実行し、エクスポートしたファイルを退避しておいてください。

1. Eclipse 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. Eclipse のメニューから [実行] - [外部ツール] - [外部ツールの構成] を選択します。
[外部ツール構成] ダイアログが表示されます。
4. 左ペインの [プログラム] を右クリックし、[新規] を選択します。
[構成の作成、管理、および実行] ページが表示されます。



[メイン] タブを選択して、次の内容を入力します。

項目名	設定値
名前	任意の名称を指定します。 ここでは、「システム構成定義の取得」と指定します。
ロケーション	<code>{env_var: COSMINEXUS_HOME}¥CSC¥bin¥cscreptl.bat</code> または [ファイルシステムの参照] ボタンから次のファイルを選択します。 <サービスプラットフォームのインストールディレクトリ>¥CSC¥bin¥cscreptl.bat
作業ディレクトリ	引数で指定するファイルの出力先ディレクトリを任意で指定します。
引数	<code>-export</code> <任意出力ファイル名>.zip ここでは、「systemdef.zip」と指定します。

5. [実行] ボタンをクリックします。

コマンドが登録、および実行され、テスト環境のリポジトリ情報が ZIP 形式のファイルとして、指定した出力先ディレクトリにエクスポートされます。

(2) 開発環境へのシステム構成定義のインポート

エクスポートしたテスト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。インポートの手順を次に示します

1. Eclipse のメニューから、[HCSC-Definer] - [定義情報管理] - [全定義情報インポート] を選択します。
リポジトリの上書き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。テスト環境からエクスポートした「systemdef.zip」の ZIP ファイルを指定します。

3. [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。



4. [システム構成定義] の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。

テスト環境のシステム構成定義が取り込まれると, 正常に完了したことを示すダイアログが表示されます。

5. [OK] ボタンをクリックします。

これで, サンプルプログラムの開発・テスト環境の構築ができました。

4 サンプルプログラムを実行する準備

この章では、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ます。

本章の構成

4.1 ビジネスプロセス・サービスアダプタを準備する

4.2 連携する SOAP サービスリクエスタ・サービス部品を準備する

4.1 ビジネスプロセス・サービスアダプタを準備する

ビジネスプロセス・サービスアダプタは HCSC サーバ上で動作します。この節では、ビジネスプロセス・サービスアダプタの準備について説明します。

4.1.1 HCSC コンポーネントのインポート

サンプルで提供する HCSC コンポーネントをインポートします。HCSC コンポーネントは、リポジトリ情報（ZIP ファイル）として提供されます。

1. [HCSC-Definer] - [定義情報管理] - [全定義情報インポート] を選択します。

リポジトリの上書きを確認するダイアログが表示されます。

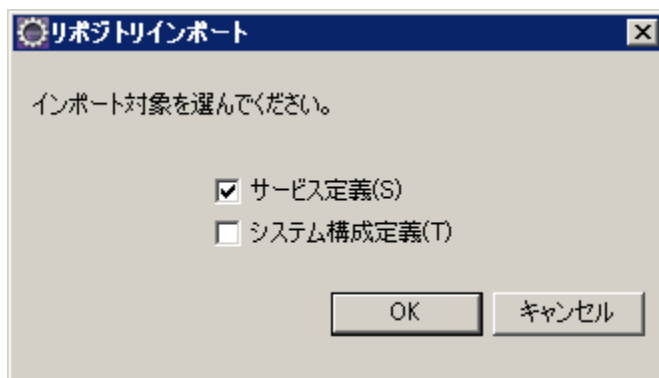
2. [はい] ボタンをクリックします。

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。次の ZIP ファイルを選択してください。

<サービスプラットフォームのインストールディレクトリ>\¥CSCTE¥Samples¥SOAP1.1_1.2mode¥ProductStock¥Repository¥ProductStock.zip

3. [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。「3.6.2 テスト環境のシステム構成定義を開発環境に取り込むための設定」でテスト環境のシステム構成定義を開発環境に取り込んだため、リポジトリからは、サービス定義の情報のみ使います。そのため、[サービス定義] のチェックボックスのみをチェックして、[OK] ボタンをクリックします。



4.1.2 HCSC コンポーネントの配備と開始

HCSC コンポーネントをテスト環境に配備して、開始します。

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。



ログインしていない場合は、アカウント認証画面が表示されます。手順 2 を実施してください。

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。

3. [OK] ボタンをクリックします。

これで配備と開始が完了しました。

4.1.3 HCSC コンポーネントの配備内容の確認

HCSC サーバに、HCSC コンポーネントが配備されていることと、起動していることを確認します。

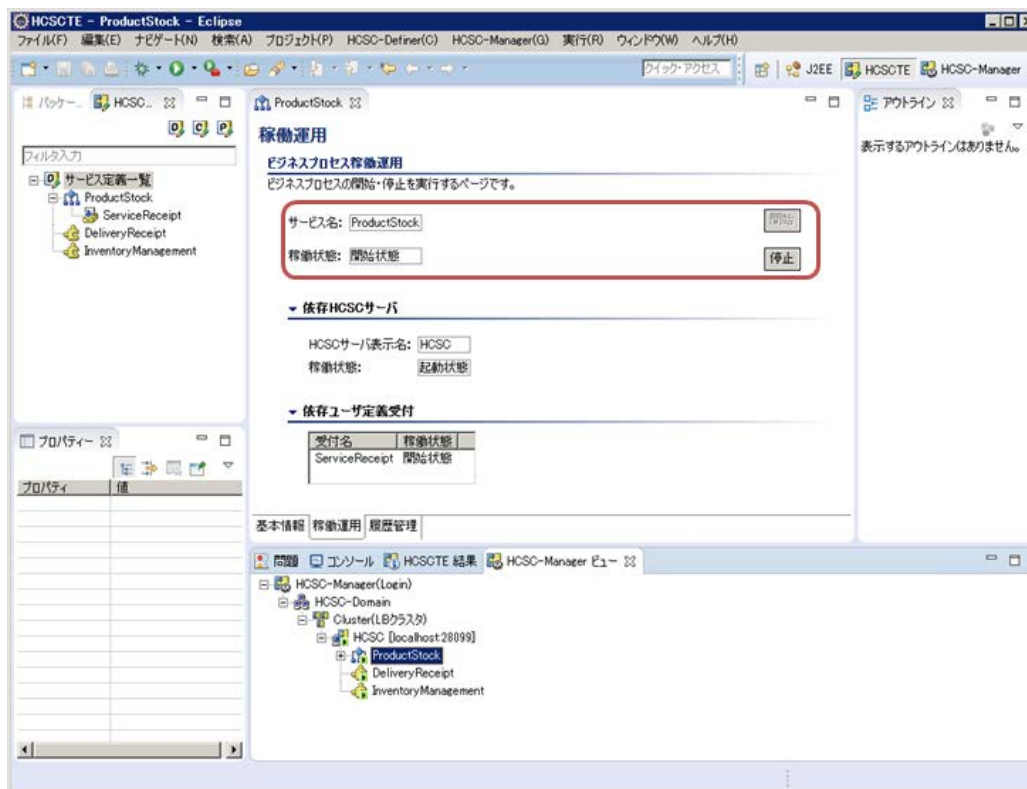
1. メニューから [ウィンドウ] - [ビューの表示] - [その他] を選択します。
[ビューの表示] ダイアログが表示されます。
2. [ビューの表示] ダイアログで [HCSC-Manager] - [HCSC-Manager ビュー] を選択し、[OK] ボタンをクリックします。
[HCSC-Manager] ビューが表示されます。
3. [HCSC-Manager] ビューの [HCSC-Manager(Login)] を右クリックし、[ログイン] を選択します。
ログイン画面が表示されます。
4. [管理ユーザーID] に「admin」を、[パスワード] に「admin」を入力して、[OK] ボタンをクリックします。

HCSC-Manager へのログインが完了します。

5. HCSC コンポーネントが起動していることを確認します。

[HCSC-Manager] ビューの [HCSC-Manager(Login)] - [HCSC-Domain] - [Cluster(LB クラスター)] - [HCSC[localhost:28099]] - [ProductStock] 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ます。

上記と同じ手順で [DeliveryReceipt] アダプタと [InventoryManagement] アダプタの稼働状態を確認します。



6. [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。

7. [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、ビジネスプロセス・サービスアダプタの準備ができました。

4.2 連携する SOAP サービスリクエスタ・サービス部品を準備する

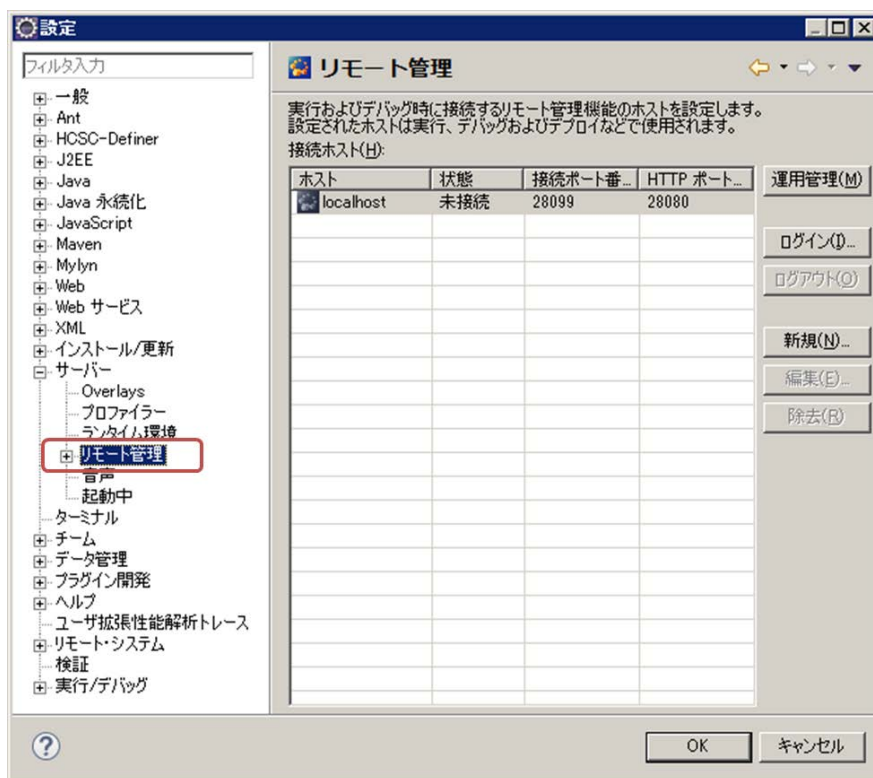
連携する SOAP サービスリクエスタ・サービス部品は J2EE サーバ上で動作します。この節では、Eclipse を使用して SOAP サービスリクエスタ・サービス部品を J2EE サーバ上にデプロイし、実行する方法を説明します。

4.2.1 Eclipse から J2EE サーバを操作するための設定

(1) 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。

1. Eclipse を起動します。
2. Eclipse のメニューから、[ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
3. 左ペインのツリービューで [サーバー] - [リモート管理] を選択します。
[リモート管理] ページが表示されます。

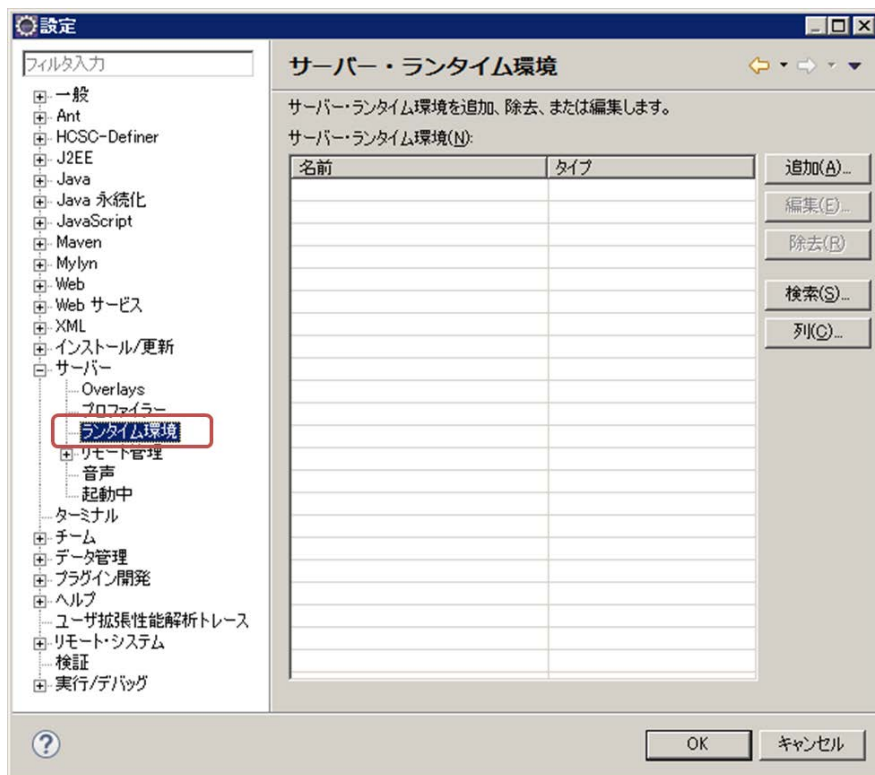


4. 接続する Management Server のホスト (localhost) を選択し、[ログイン] ボタンをクリックします。
ログイン画面が表示されます。
5. [管理ユーザーID] に「admin」を、[パスワード] に「admin」を入力し、[OK] ボタンをクリックします。
Management Server リモート管理機能に接続します。
6. [OK] ボタンをクリックします。

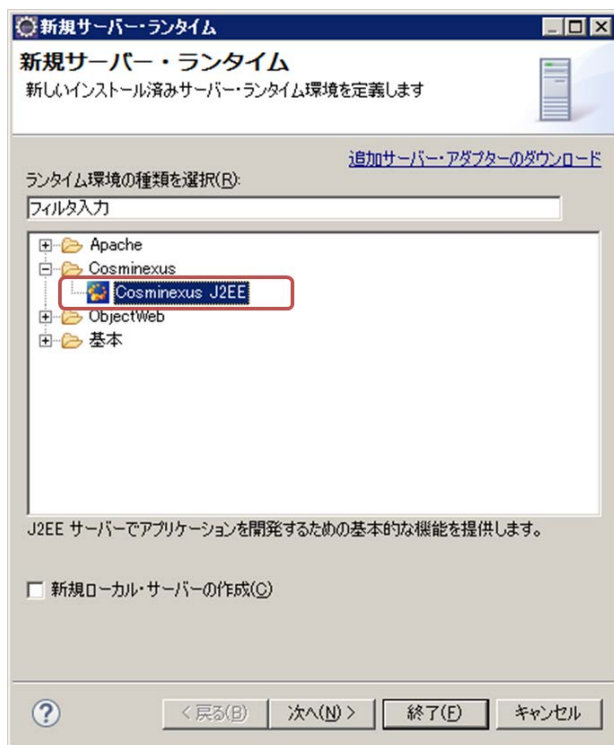
(2) サーバランタイムの作成

J2EE サーバを操作するために、サーバランタイムを作成します。サーバランタイムとは J2EE サーバのインストール先などの環境情報です。作成手順を次に示します。

1. Eclipse のメニューから [ウィンドウ] - [設定] を選択します。
[設定] ダイアログが表示されます。
2. 左ペインのツリービューから [サーバー] - [ランタイム環境] を選択します。
右ペインに [サーバー・ランタイム環境] ページが表示されます。



3. [追加] ボタンをクリックします。
[新規サーバー・ランタイム] ダイアログが表示されます。
4. ツリービューから [Cosminexus] - [Cosminexus J2EE] を選択します。



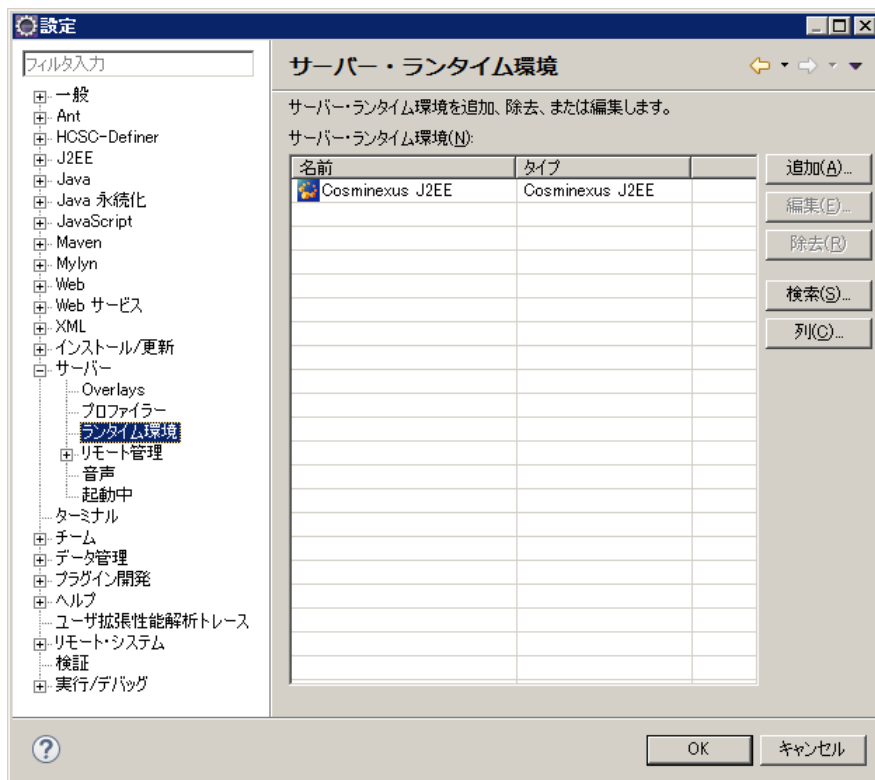
5. [次へ] ボタンをクリックします。
[J2EE サーバー・ランタイム] のダイアログが表示されます。
6. 名前を「Cosminexus J2EE」に設定します。

注※

デフォルトで入力されている名前の末尾が空白である場合は、末尾の空白を削除してください。



7. [終了] ボタンをクリックします。



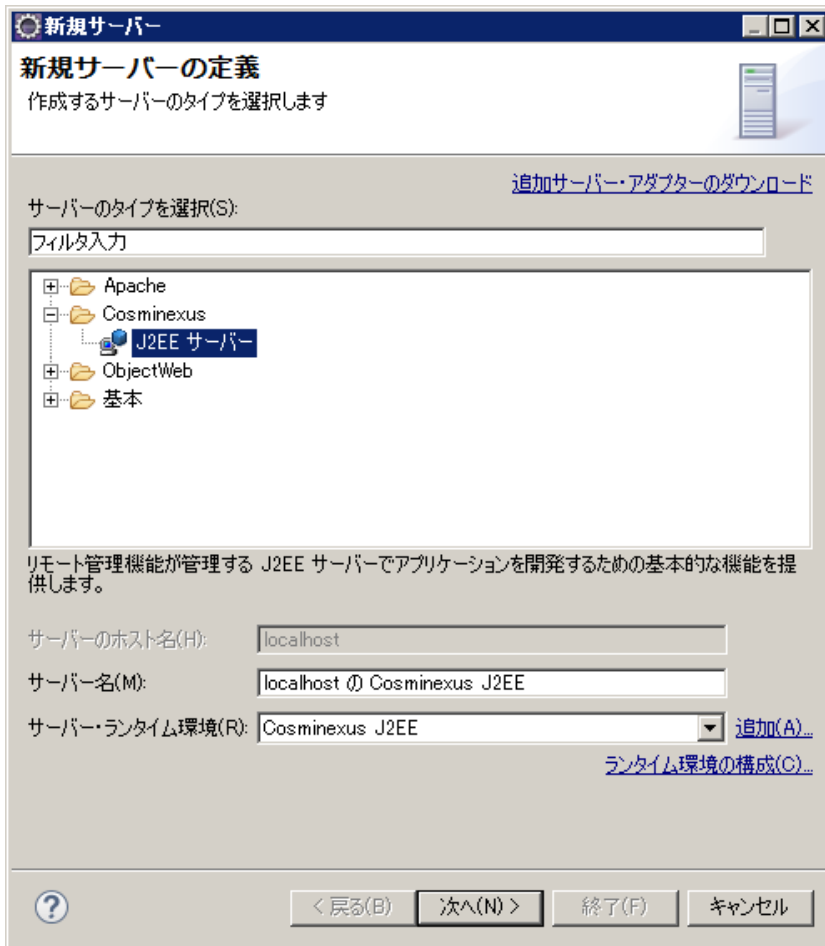
8. [サーバー・ランタイム環境] に「Cosminexus J2EE」が追加されたのを確認し、[OK] ボタンをクリックします。
設定が保存されます。

(3) J2EE サーバの選択

Eclipse から J2EE サーバを操作するための手順です。Eclipse から、ここで選択した J2EE サーバの起動・停止を操作できます。また、J2EE アプリケーションのデプロイや起動・停止を操作できます。

設定手順を次に示します。

1. Eclipse のメニューから、[ファイル] - [新規] - [その他] を選択します。
[新規] ダイアログが表示されます。
2. [サーバー] - [サーバー] を選択して、[次へ] ボタンをクリックします。
[新規サーバー] ダイアログの [新規サーバーの定義] ページが表示されます。

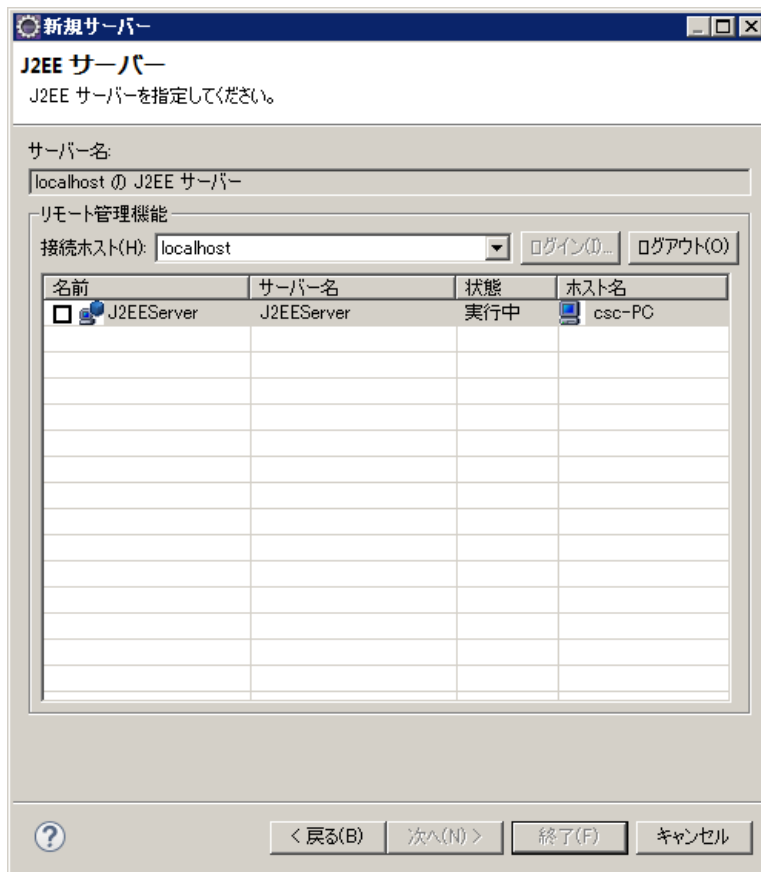


3. 次の項目を指定します。

項目名	設定値
サーバーのタイプを選択	[Cosminexus] - [J2EE サーバー] を選択します。
サーバー名	サーバ名は、[J2EE サーバー] ページの値が使用されるため、ここでは設定不要です。
サーバー・ランタイム環境	「Cosminexus J2EE」サーバランタイムを指定してください。

4. [次へ] ボタンをクリックします。

[J2EE サーバー] ページが表示されます。

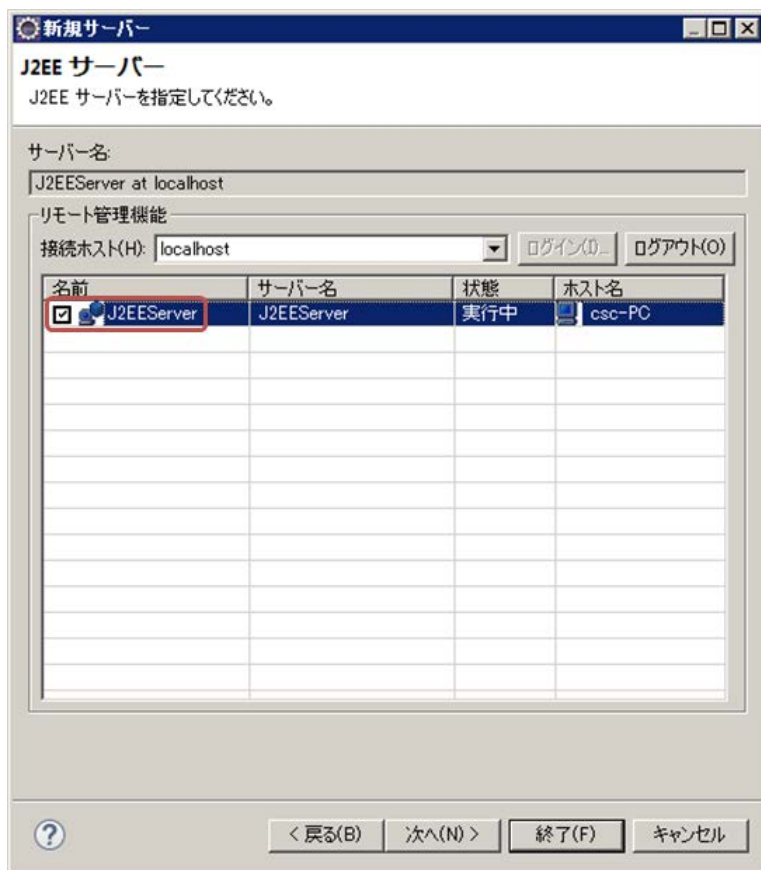


5. リモート管理機能の接続ホストに、接続先のホストを選択します。接続ホストにログインしていない場合は [ログイン] ボタンをクリックします。

[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。接続先の J2EE サーバが表示されます。

6. 使用する J2EE サーバ (J2EEServer) のチェックボックスにチェックを入れます。



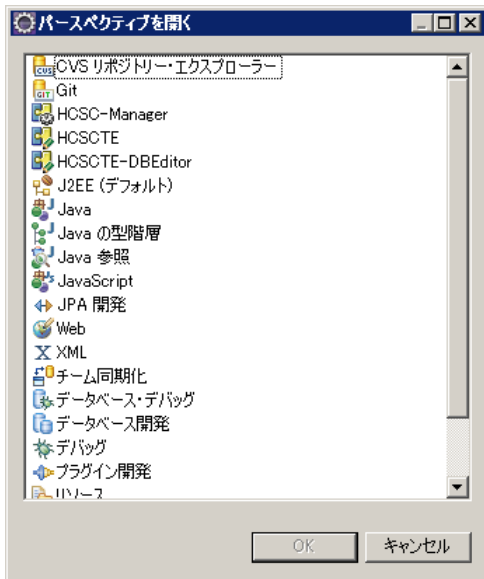
7. [終了] ボタンをクリックします。
設定が保存されます。

4.2.2 Eclipse プロジェクトのインポート

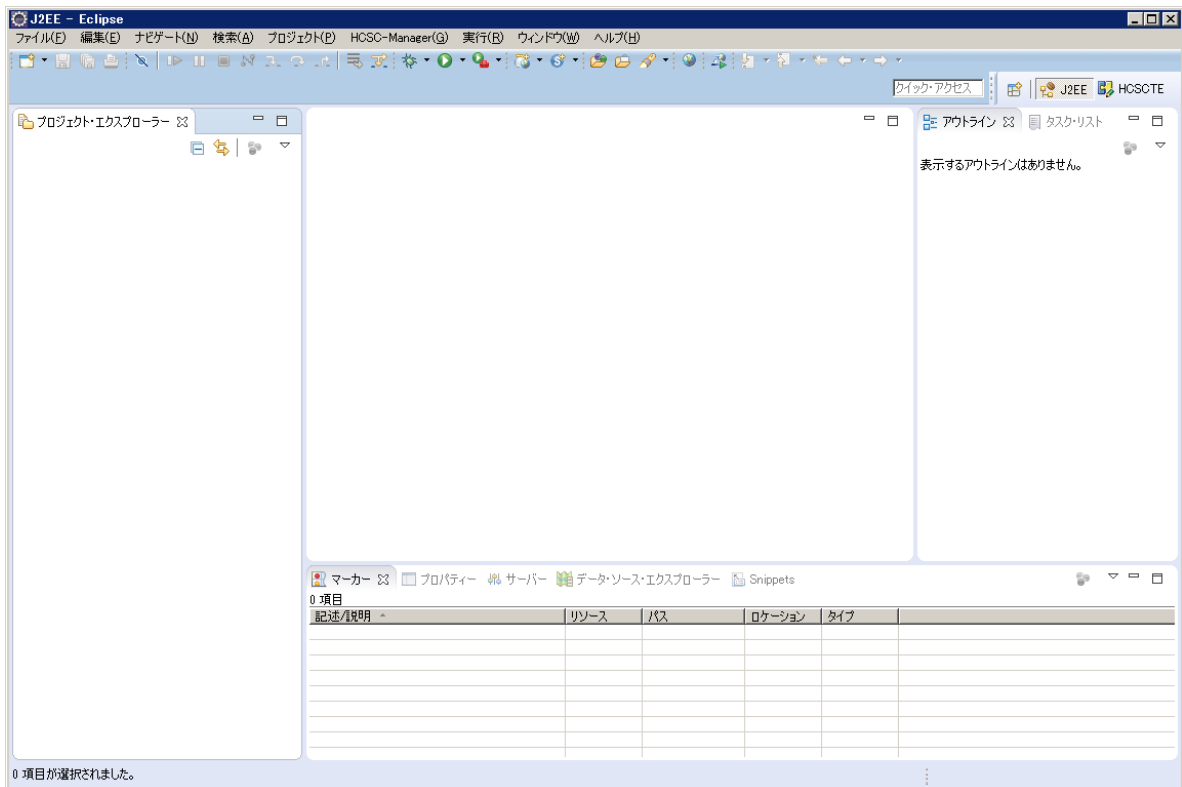
連携する SOAP サービスリクエスタ・サービス部品は Eclipse プロジェクトとしてインポートします。

Eclipse プロジェクトのインポート手順を説明します。

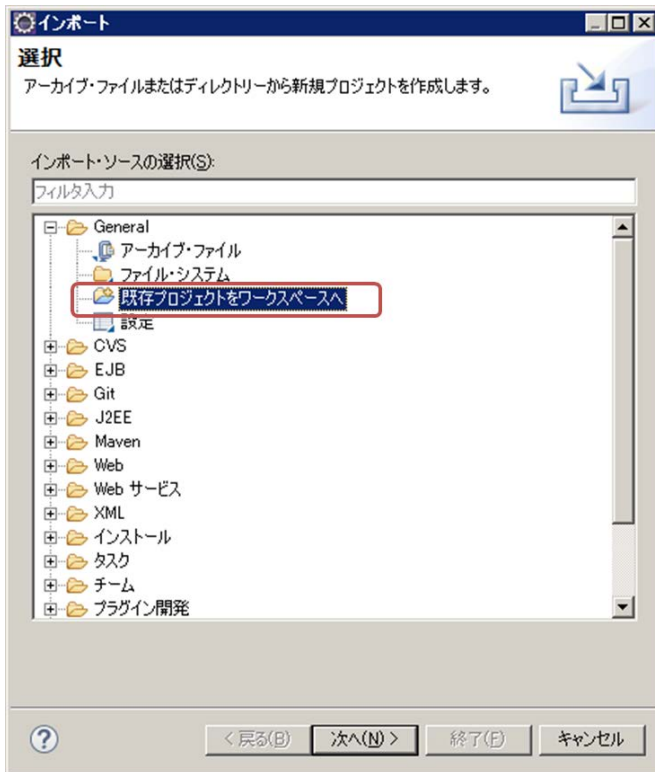
1. Eclipse のメニューから [ウィンドウ] - [パースペクティブを開く] - [その他] を選択します。
[パースペクティブを開く] ダイアログが表示されます。



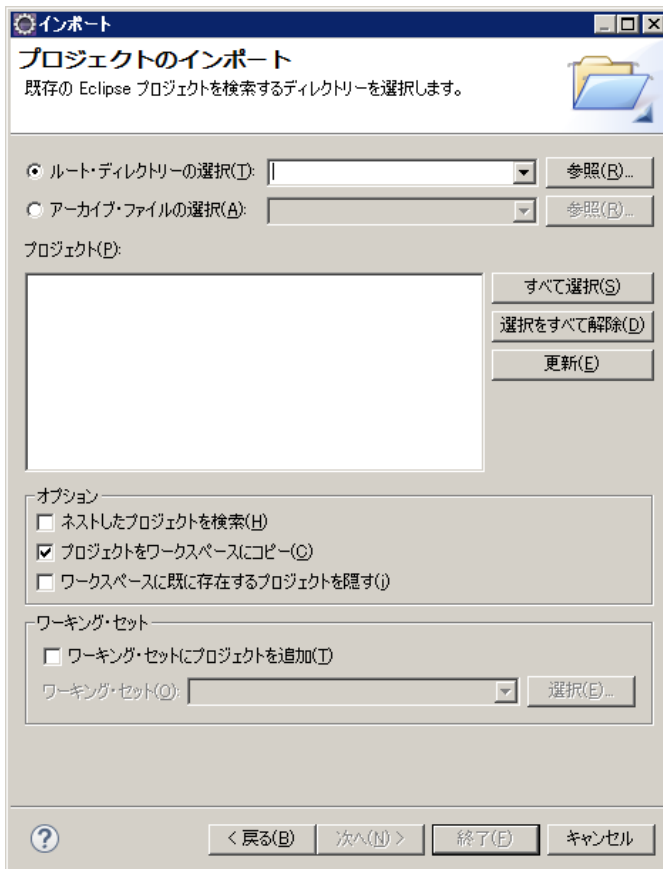
2. [J2EE (デフォルト)] を選択し、[OK] ボタンをクリックします。
[J2EE] パースペクティブが表示されます。



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 [General] - [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。

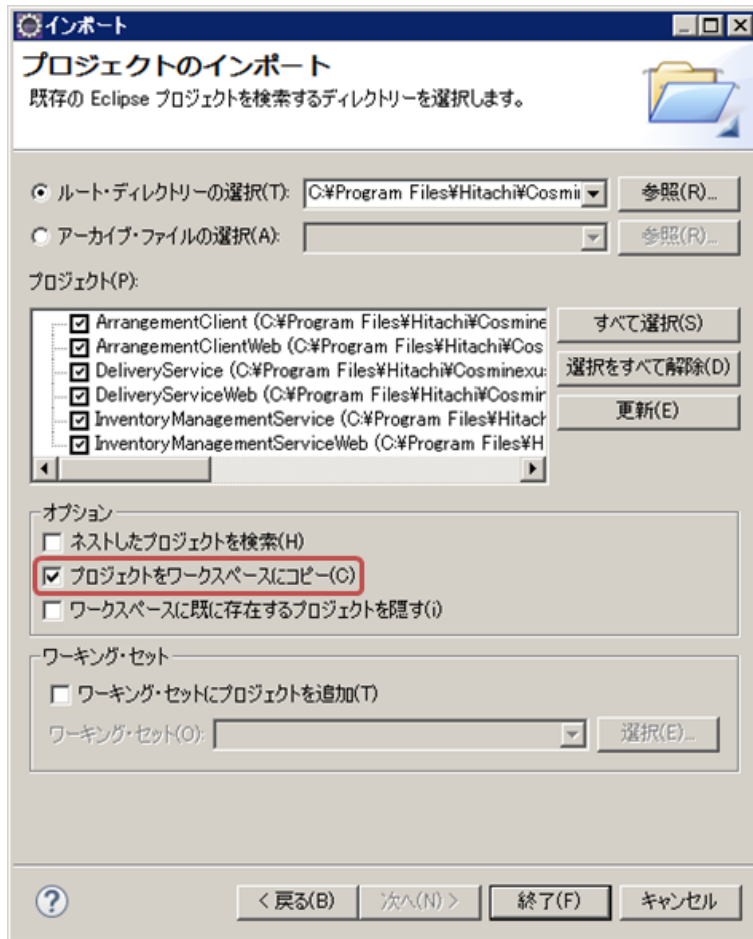


6. [ルート・ディレクトリーの選択] を選択し, [参照] ボタンをクリックします。
ファイルの参照画面が表示されます。
7. サンプルプログラムが格納されているディレクトリ (<サービスプラットフォームのインストールディ

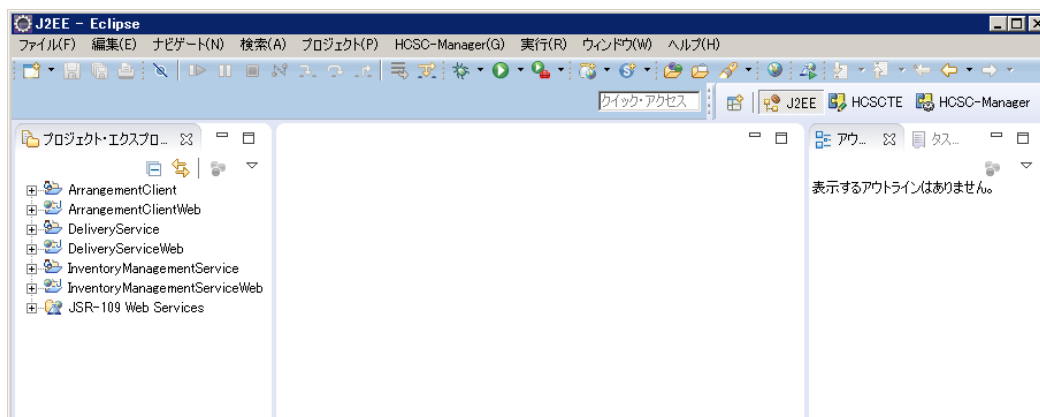
レクトリ>CSCTE¥Samples¥SOAP1.1_1.2mode¥ProductStock) を選択し, [OK] ボタンをクリックします。

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。

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。

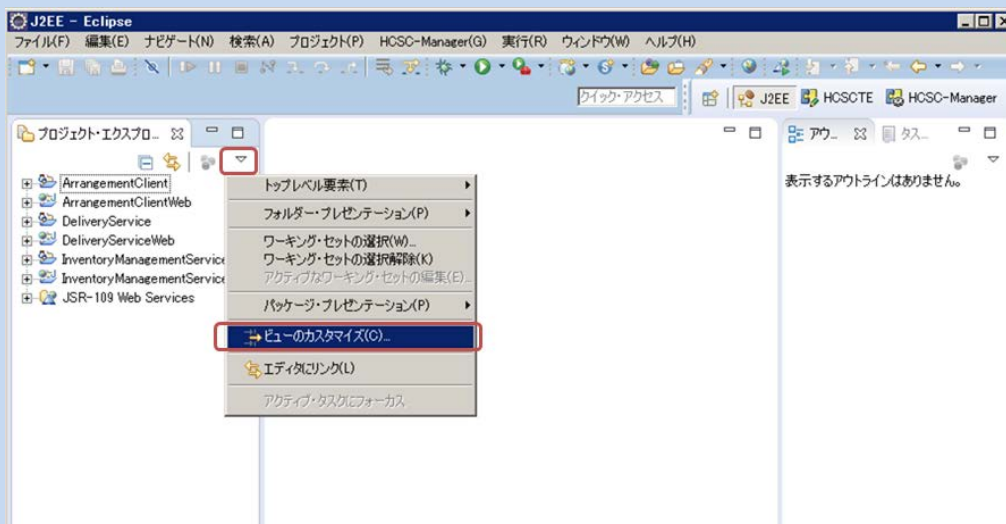


注意事項 1

インポート後に Eclipse のプロジェクト・エクスプローラー上に [JSR-109 web services] プロジェクトが表示されることがあります。本サンプルでは使用しないため、次の手順で非表示にすることを勧めます。

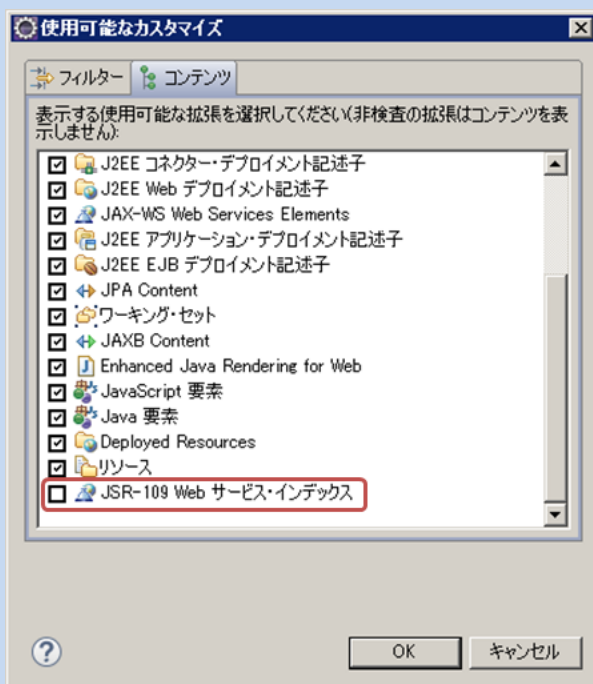
1. [プロジェクト・エクスプローラー] ビューのメニューを開き、[ビューのカスタマイズ] を選択します。

[使用可能なカスタマイズ] ダイアログが表示されます。



2. [コンテンツ] タブを選択して、[JSR-109 web サービス・インデックス] のチェックを外します。

[JSR-109 web services] が非表示になります。



注意事項 2

ArrangementClientWeb プロジェクトの ArrangementService.wsdl に警告が発生しますが、問題なくご利用いただけます。

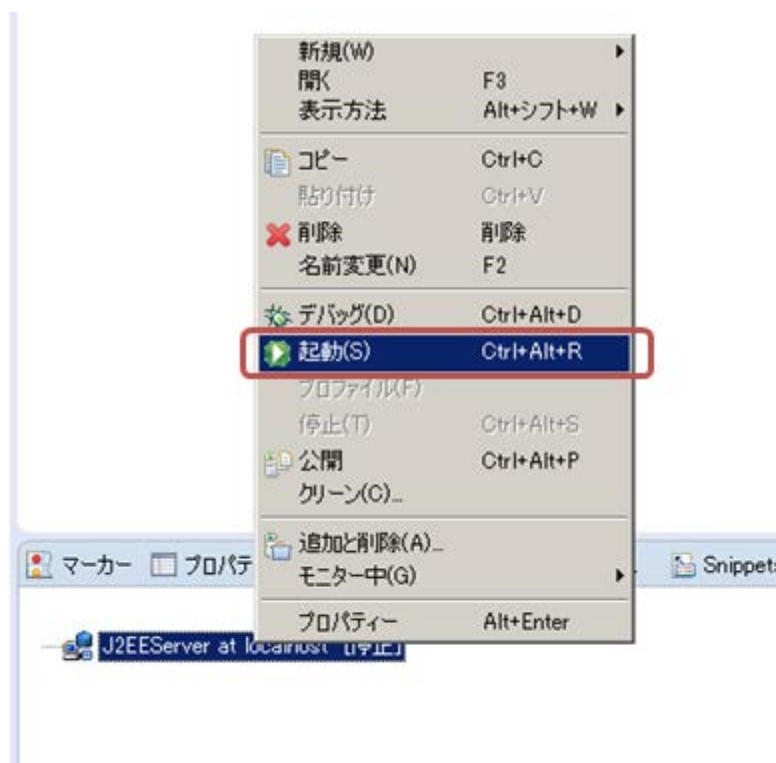
4.2.3 Eclipse プロジェクトのデプロイ

Eclipse プロジェクトのデプロイの手順を説明します。

1. 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. [J2EE (デフォルト)] を選択し、[OK] ボタンをクリックします。
[J2EE] パースペクティブが表示されます。
3. [サーバー] ビューの [J2EEServer at localhost] サーバを右クリックし、[起動] を選択します。
「サーバー 'J2EEServer' が外部で実行中です。サーバー 'J2EEServer' を停止してください。」というエラーダイアログが表示されることがあります。この場合、[スタート] メニューの [プログラム] から [Cosminexus*]-[テストサーバ停止] を選択して、テスト環境の Performance Tracer, J2EE サーバ、および、標準受付とユーザ定義受付を含む HCSC サーバを停止してください。そのあと、再度 [サーバー] ビューの [J2EEServer at localhost] サーバを右クリックし、[起動] を選択します。

注※

プログラムフォルダ名を変更している場合、変更したプログラムフォルダから、選択してください。



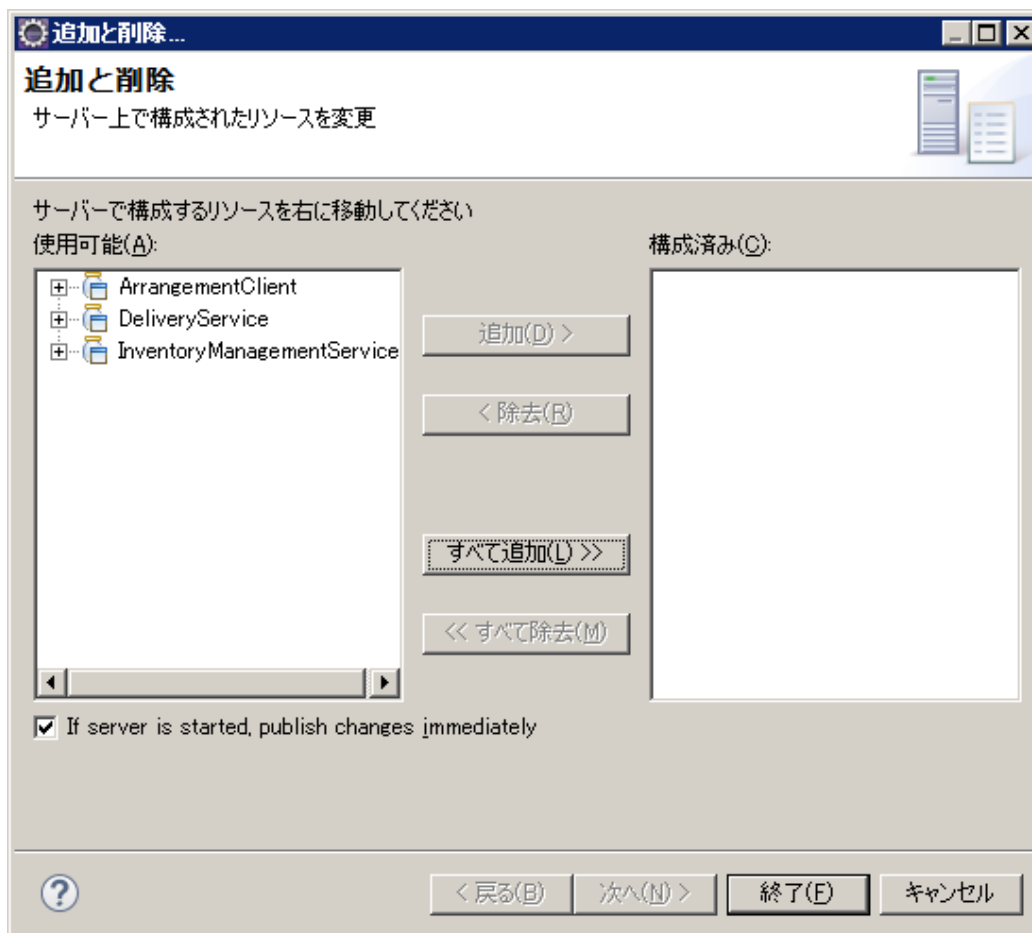
ログイン画面が表示されます。

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。

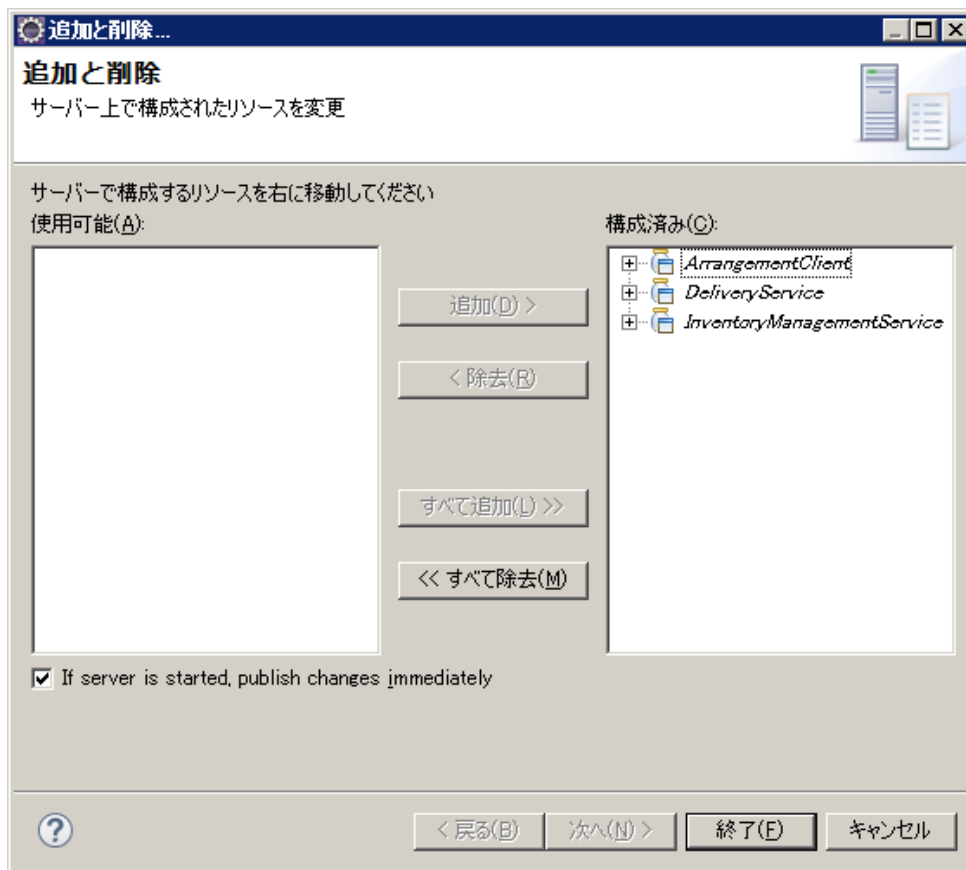
処理中であることを知らせるメッセージが表示されたあと， サーバが起動されます。

5. [サーバー] ビューの [J2EEServer at localhost] サーバを右クリックし， [追加と削除...] を選択します。

[追加と削除...] ダイアログが表示されます。



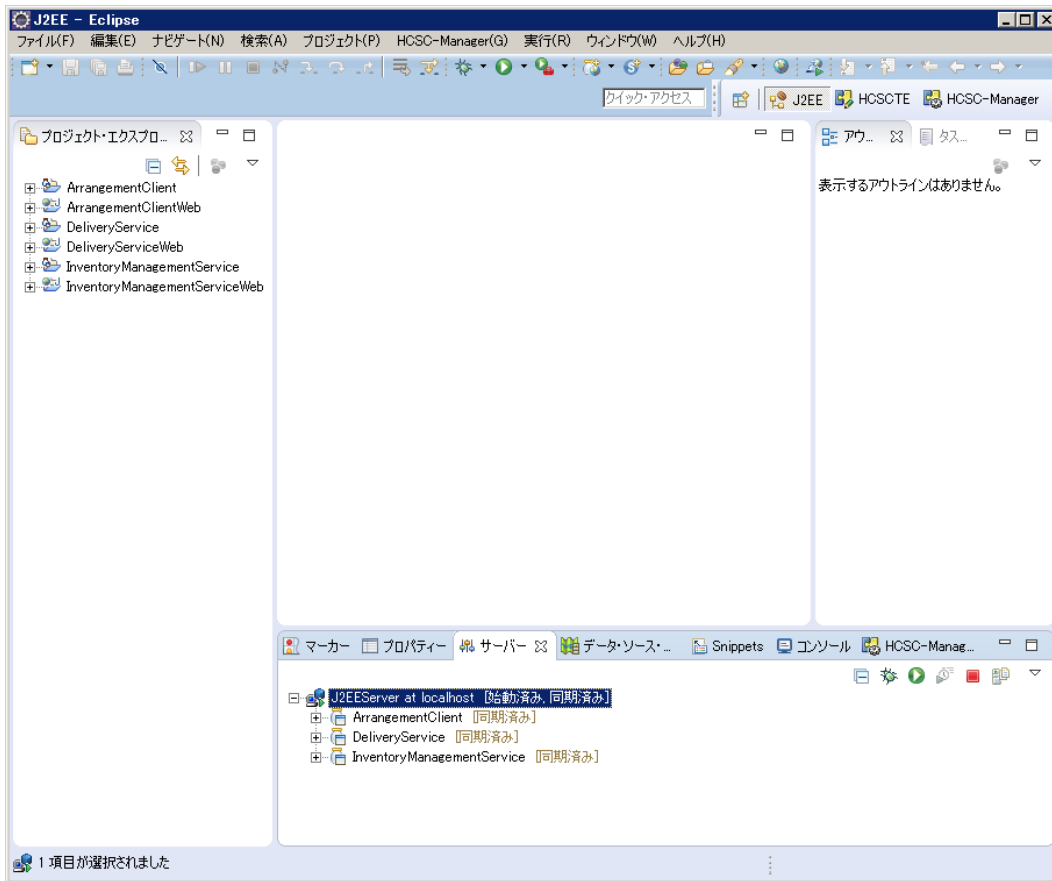
6. [使用可能] から [ArrangementClient] ， [DeliveryService] および [InventoryManagementService] を選択し， [追加] ボタンをクリックします。



[ArrangementClient] , [DeliveryService] および [InventoryManagementService] が [使用可能] から [構成済み] へ移動します。

7. [終了] ボタンをクリックします。

処理終了後, [J2EE] パースペクティブに戻ります。[サーバー] ビューの [J2EEServer at localhost] の配下に [ArrangementClient] , [DeliveryService] および [InventoryManagementService] プロジェクトがあることを確認してください。



8. [サーバー] ビューの [J2EEServer at localhost] サーバを右クリックし、[停止] を選択します。
9. [スタート] メニューの [プログラム] から [Cosminexus[※]] - [テストサーバ起動] を選択して、テスト環境の Performance Tracer, J2EE サーバ、および、標準受付とユーザ定義受付を含む HCSC サーバを起動します。

注※

プログラムフォルダ名を変更している場合、変更したプログラムフォルダから、選択してください。

これで、連携する SOAP サービスリクエスタ、サービス部品の準備ができました。

5 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。

本章の構成

5.1 サンプルプログラムの実行

5.2 商品在庫数の初期化方法

5.1 サンプルプログラムの実行

商品手配サンプルプログラムの動作手順を次に示します。

注意事項

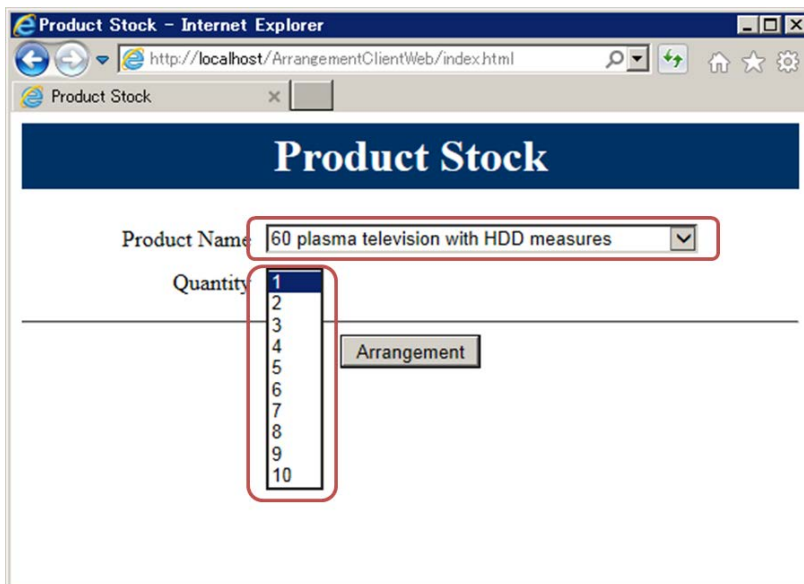
サンプルプログラムを実行するときは、テスト環境が起動されている必要があります。テスト環境を起動する方法については、「3.4.2 テスト環境の起動」を参照してください。サンプルプログラムを実行しないときは、テスト環境を停止しておいてください。テスト環境を停止する方法については、「7.2 テスト環境の停止」を参照してください。

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。

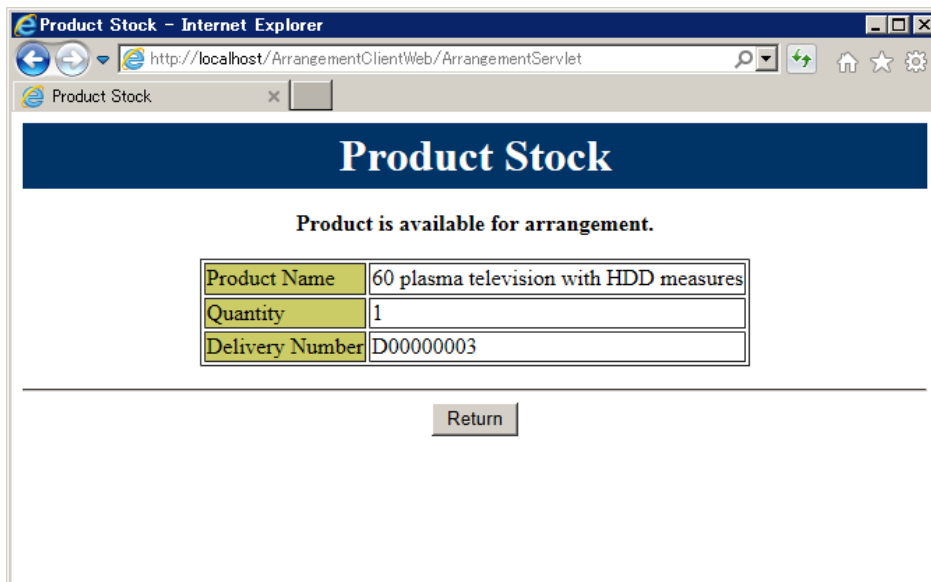


3. [Arrangement] ボタンをクリックします。

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。



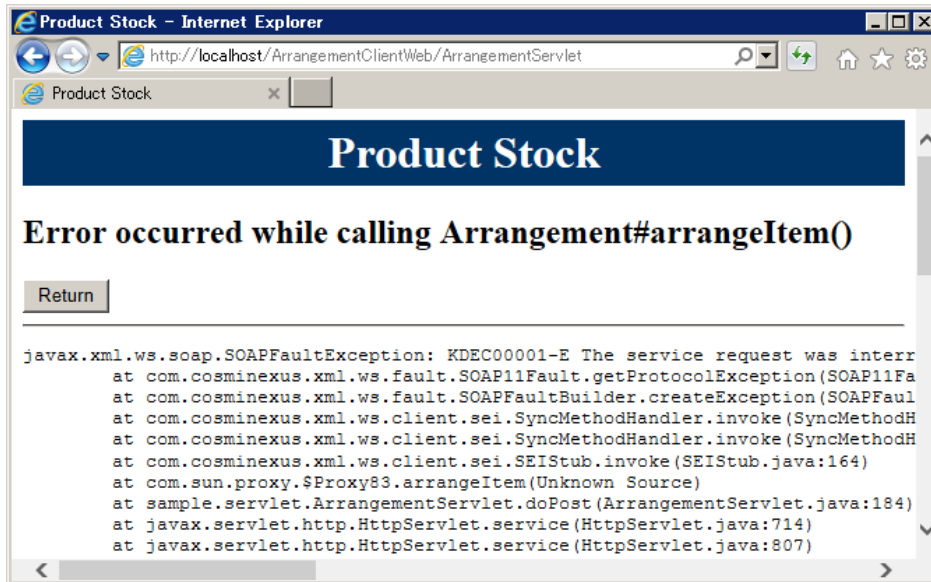
正常に応答した場合（在庫がない場合）

在庫がないことを知らせるメッセージ，商品名，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。



5.2 商品在庫数の初期化方法

商品手配サンプルプログラムでの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、J2EE サーバを再起動するか、または Eclipse 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。J2EE サーバの再起動、および Eclipse プロジェクトをリデプロイする方法を次に示します。

J2EE サーバの再起動方法

1. Eclipse 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. [J2EE (デフォルト)] を選択し、[OK] ボタンをクリックします。
[J2EE] パースペクティブが表示されます。
3. [サーバー] ビューの [J2EE Server at localhost] サーバを右クリックし、[起動] を選択します。

Eclipse プロジェクトのリデプロイ方法

1. Eclipse 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. [J2EE (デフォルト)] を選択し、[OK] ボタンをクリックします。
[J2EE] パースペクティブが表示されます。
3. [サーバー] ビューの [J2EE Server at localhost] サーバを右クリックし、[追加と削除...] を選択します。
[追加と削除...] ダイアログが表示されます。

Eclipse プロジェクトをアンデプロイしたあと、デプロイします。アンデプロイの詳細については、「7.1.1

(1) Eclipse プロジェクトのアンデプロイ」を、デプロイの詳細については、「4.2.3 Eclipse プロジェクトのデプロイ」をそれぞれ参照してください。

6 システ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。

本章の構成

- 6.1 商品手配システムの開発の手順
- 6.2 商品手配システムの開発
- 6.3 ビジネスプロセスのデバッグ
- 6.4 開発した商品手配システムを実行する準備

6.1 商品手配システムの開発の手順

この章では、5章までで実行した製品が提供するサンプルのことを「サンプルプログラム」と呼び、この章で開発体験することで作成するものを「商品手配システム」と呼びます。商品手配システムの開発の手順を次の図に示します。

注意事項

下記の手順を行う前に、「3.2 Service Architect のインストール」と「3.3 Eclipse の準備」が実行済みである必要があります。

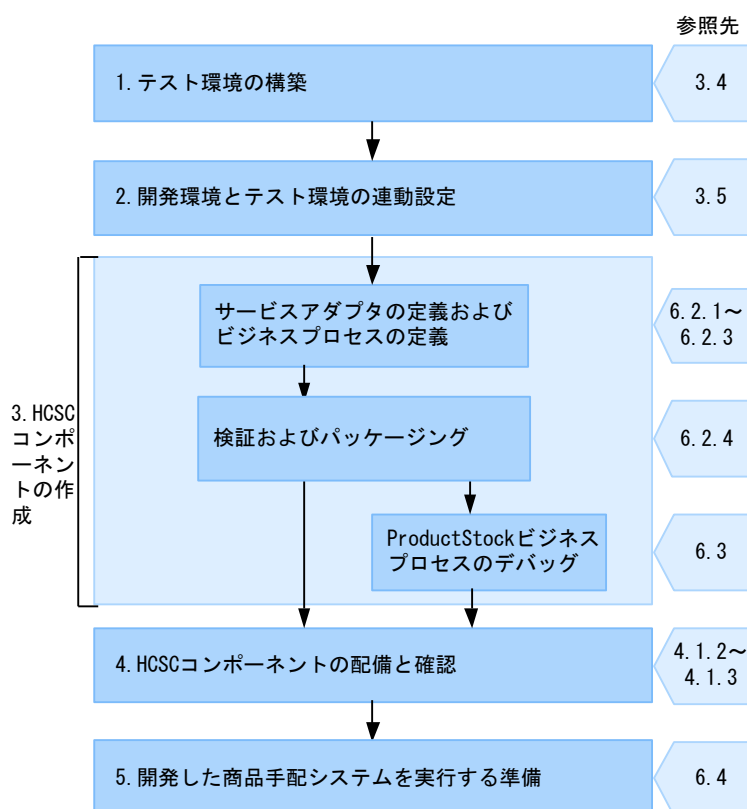


図 6-1 商品手配システムの開発の手順

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あと、再度構築してください。詳細については、「3.4.1 HCSC 簡易環境セットアップ」を参照してください。テスト環境の構築が終わりましたら、テスト環境を起動してください。詳細については、「3.4.2 テスト環境の起動」を参照してください。

2. 開発環境とテスト環境の連動設定

開発環境の SOAP モードの設定とテスト環境のシステム構成定義を開発環境に取り込むための設定をします。詳細については、「3.5 開発環境とテスト環境の連動設定」を参照してください。

3. 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、サービスアダプタを経由して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。詳細については、「6.2 商品手配システムの開発」を参照してください。また、作成したビジネスプロセスをデバッグする場合は、「6.3 ビジネスプロセスのデバッグ」を参照してください。

4. HCSC コンポーネントの配備と確認

HCSC コンポーネントをサーバに配備し、起動状態を確認します。詳細については、「4.1.2 HCSC コンポーネントの配備と開始」、「4.1.3 HCSC コンポーネントの配備内容の確認」を参照してください。

5. 開発した商品手配システムを実行する準備

サンプルプログラムで提供するサービスリクエストおよびサービス部品を利用することで、開発した商品手配システムの動作を確認できます。詳細については、「6.4 開発した商品手配システムを実行する準備」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。

6.2 商品手配システムの開発

サンプルプログラムと同じ動作をする，商品手配システムを開発してみましょう。

商品手配システムでは，次に示す 3 つのコンポーネントを定義します。

- InventoryManagement サービスアダプタ
- DeliveryReceipt サービスアダプタ
- ProductStock ビジネスプロセス

6.2.1 InventoryManagement サービスアダプタの定義

InventoryManagement サービスアダプタは，在庫管理サービス用 WSDL

(InventoryManagementService.wsdl) を使用して定義します。InventoryManagement 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 6-1 InventoryManagement 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。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	InventoryManagement	サービスアダプタの名称を指定します。
WSDL ファイル*	InventoryManagementService.wsdl	WSDL には，Web サービスがどのような機能を持ち，それを利用するためにはどのような要求をすればよいのか，などを記述する方法が定義されています。サービスアダプタは，在庫管理サービス用 WSDL (InventoryManagementService.wsdl) を使用して作成します。
Port の指定	InventoryManager	InventoryManagement サービスアダプタのポートを指定します。
サービス ID	InvAdp	InventoryManagement サービスアダプタの ID を指定します。

注※

各ファイルの格納場所については，「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。

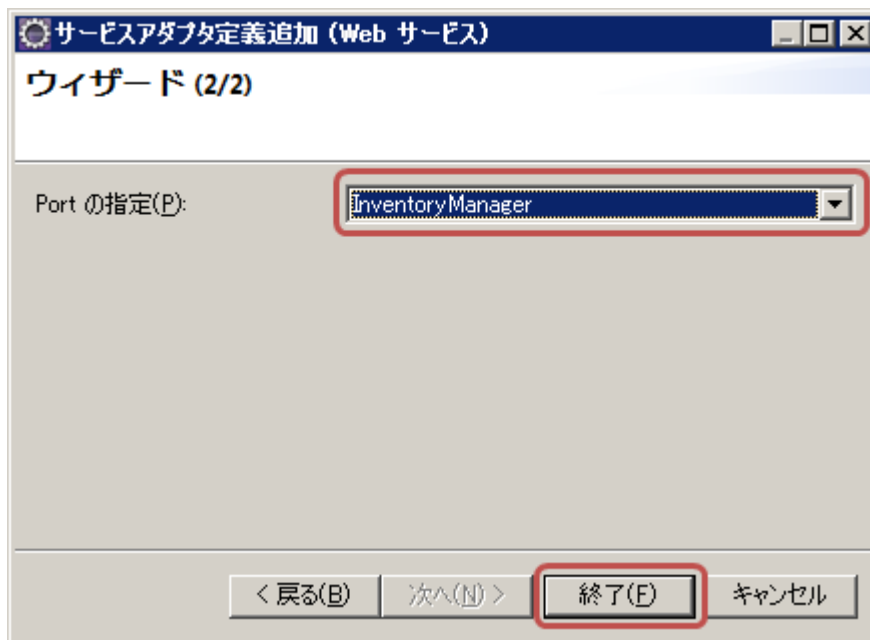
InventoryManagement サービスアダプタの追加と定義手順を次に示します。

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。
SOAP 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「InventoryManagement」を入力し，WSDL ファイルには「InventoryManagementService.wsdl」を指定して， [次へ] ボタンをクリックします。



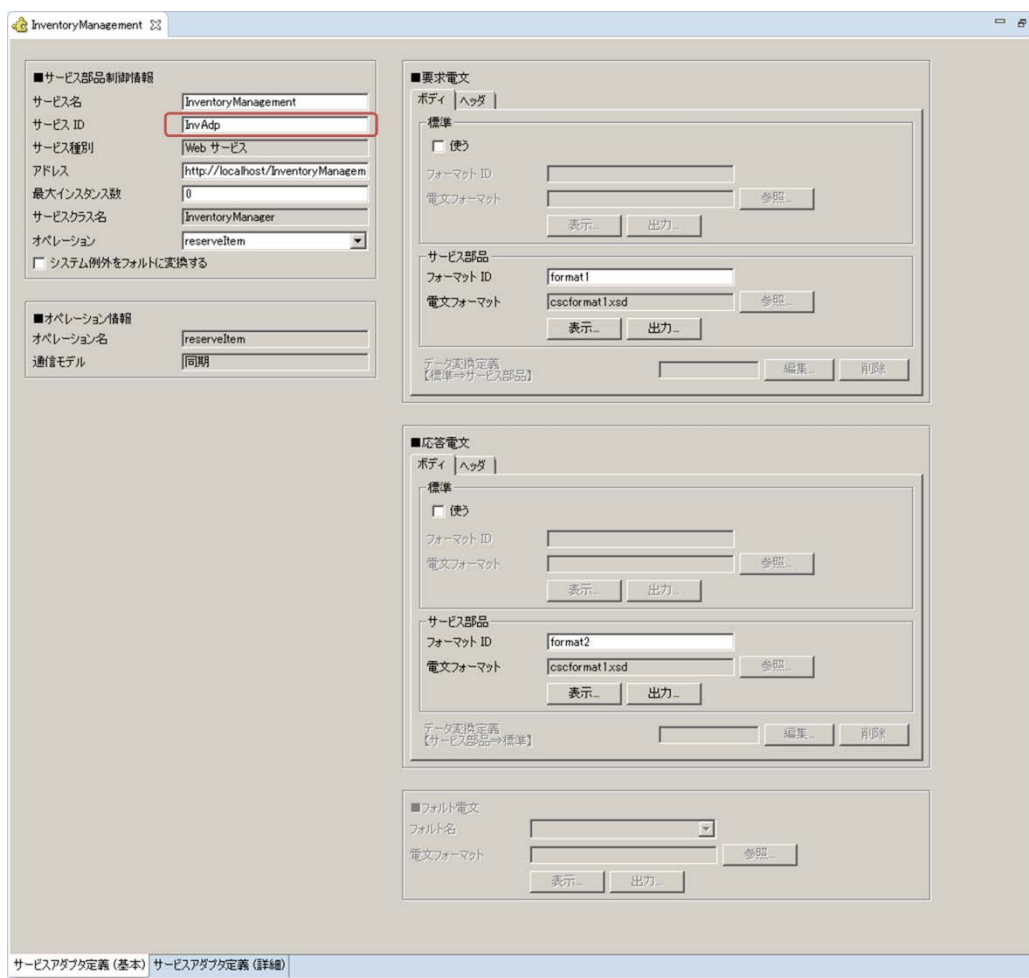
ポートを指定するダイアログが表示されます。

4. ドロップダウンリストからポート「InventoryManager」を選択して、[終了] ボタンをクリックします。



サービスアダプタ「InventoryManagement 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「InvAdp」に変更します。



6. メニューから [ファイル] - [保管] を選択します。

6.2.2 DeliveryReceipt サービスアダプタの定義

DeliveryReceipt 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。DeliveryReceipt 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 6-2 DeliveryReceipt 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。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	DeliveryReceipt	サービス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。DeliveryReceipt 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	DeliveryReceipt サービスアダプタのポートを指定します。
サービス ID	DelAdp	DeliveryReceipt サービスアダプタの ID を指定します。

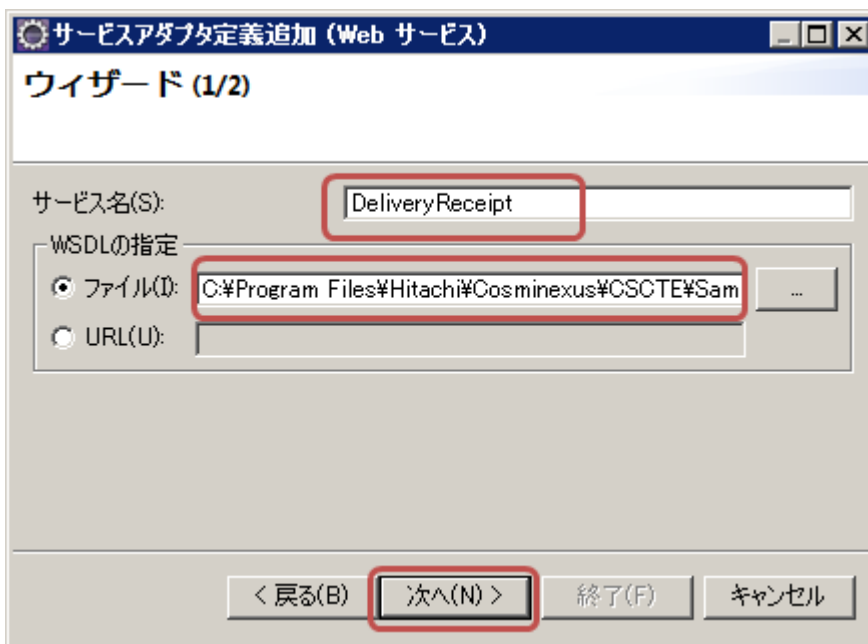
注※

各ファイルの格納場所については、「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。

ださい。

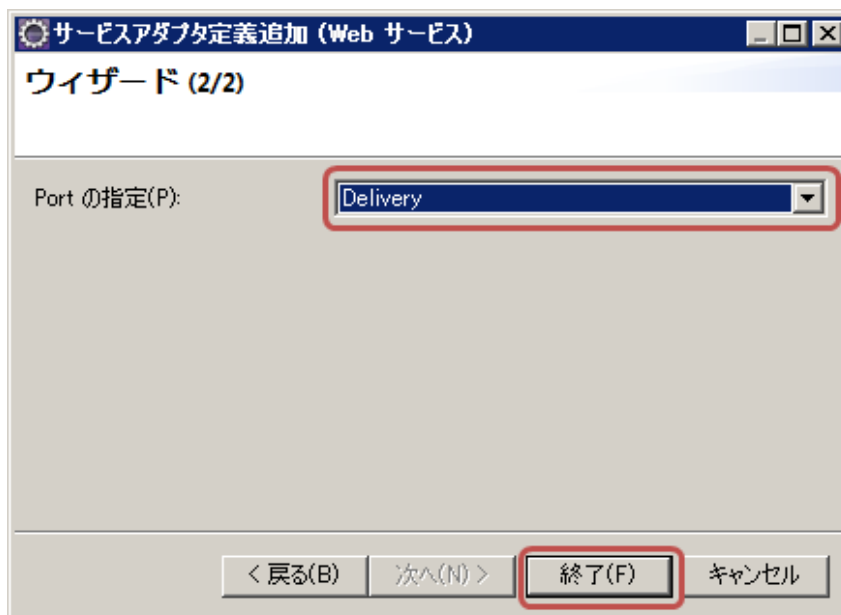
DeliveryReceipt サービスアダプタの追加と定義手順を次に示します。

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。
SOAP 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「DeliveryReceipt」を入力し、WSDL ファイルには「DeliveryService.wsdl」を指定して、「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からポート「Delivery」を選択して、「終了」ボタンをクリックします。



サービスアダプタ「DeliveryReceipt 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。

The screenshot shows the 'DeliveryReceipt' service adapter definition interface. The 'Service ID' field is highlighted with a red box and contains the text 'DelAdp'. The interface is divided into several sections: 'Service Component Information', 'Request Message', 'Response Message', and 'Fault Message'. Each section has tabs for 'Body' and 'Header' and options for 'Standard' and 'Service Component'. The 'Request Message' section shows 'format3' for the service component and 'cscformat1.xsd' for the message format. The 'Response Message' section shows 'format4' for the service component and 'cscformat1.xsd' for the message format. The 'Fault Message' section has a dropdown for the fault name and a message format field.

6. メニューから [ファイル] - [保管] を選択します。

6.2.3 ProductStock ビジネスプロセスの定義

商品手配システムの ProductStock ビジネスプロセスは、次のような処理を実現します。

1. ProductStock 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. InventoryManagement サービスアダプタを介して在庫管理サービスを呼び出します。
3. InventoryManagement サービスを呼び出した結果、在庫がない場合は、「*」（No Stock）をサービスリクエストに返します。
4. 在庫がある場合は、DeliveryReceipt サービスアダプタを介して配送受付サービスを呼び出し、配送番号を取得します。
5. サービスリクエストに、取得した配送番号を返します。

また、商品手配システムのリクエストにはサンプルプログラムと同じサービスリクエストを使用します。そのため、ユーザ定義受付（SOAP 受付）をビジネスプロセスに関連づけて定義して、サンプルプログラムと同様に SOAP プロトコルでリクエストを受付できるようにします。

商品手配システ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

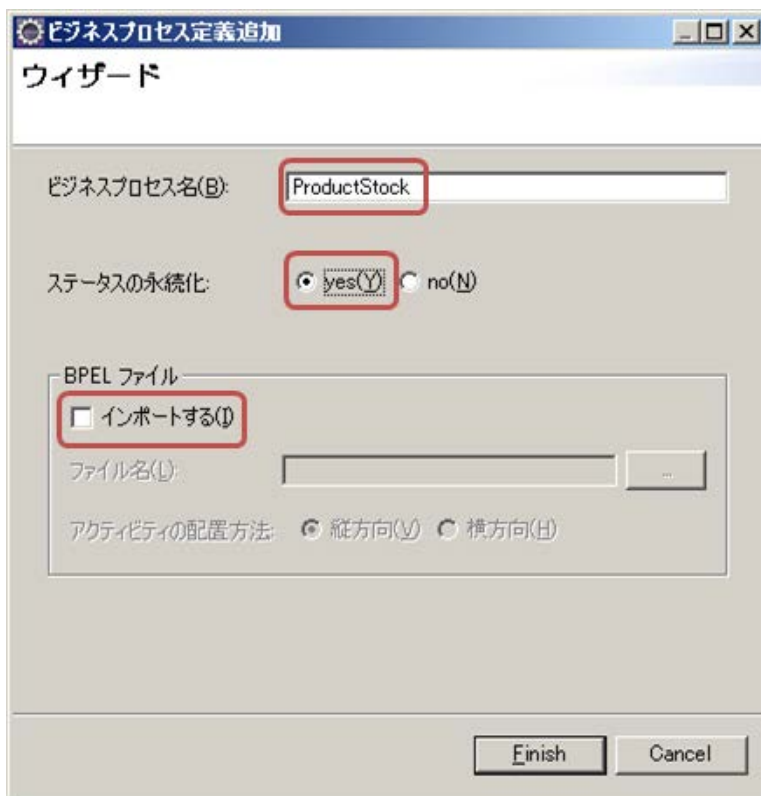
ProductStock ビジネスプロセスを追加するときに設定する値を次の表に示します。

表 6-3 ProductStock 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	ProductStock	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]	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。

ProductStock 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「ProductStock」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] はチェックを外したままに設定してください。



3. [終了] ボタンをクリックします。
ビジネスプロセス「ProductStock」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「ProductStock」を選択します。
プロパティビューに ProductStock 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] キーを押します。

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
ビジネスプロセス名	ProductStock

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) ユーザ定義受付の追加

ユーザ定義受付では、サービスリクエストとのインターフェースをあらかじめ決める必要があります。商品手配システムでは、WSDL ファイルを利用してサービスリクエストとユーザ定義受付とのインターフェースを定義します。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 6-4 ユーザ定義受付を追加するときの設定する値

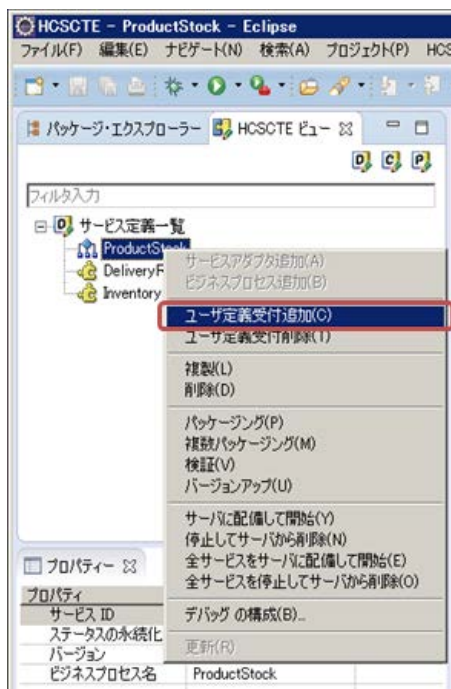
項目名	設定する値	説明
受付種別	SOAP 受付	受付種別を選択します。
受付名	サービス受付	ユーザ定義受付の名称を指定します。
WSDL ファイル※	ArrangementService.wsdl	使用する WSDL ファイル名を指定します。
ポート名	Arrangement	ポート名を指定します。

注※

各ファイルの格納場所については、「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。

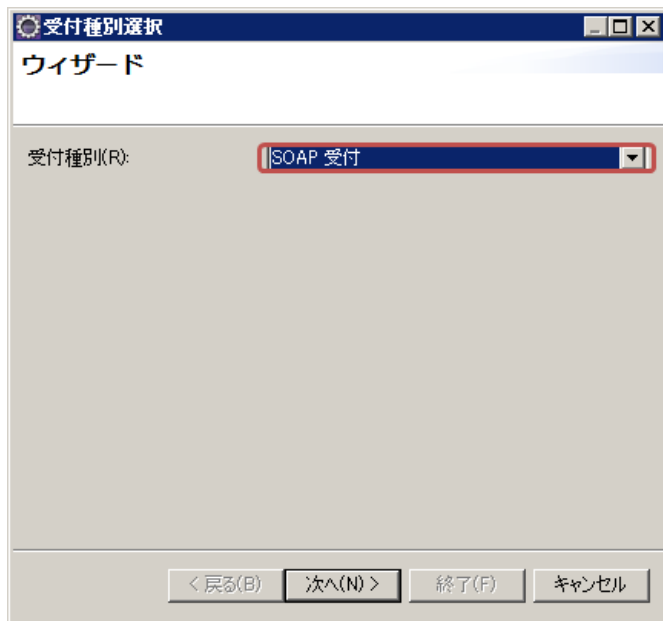
ProductStock のユーザ定義受付の追加手順を次に示します。

1. ツリービューのサービス定義一覧からビジネスプロセス名 [ProductStock] 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。

2. [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。



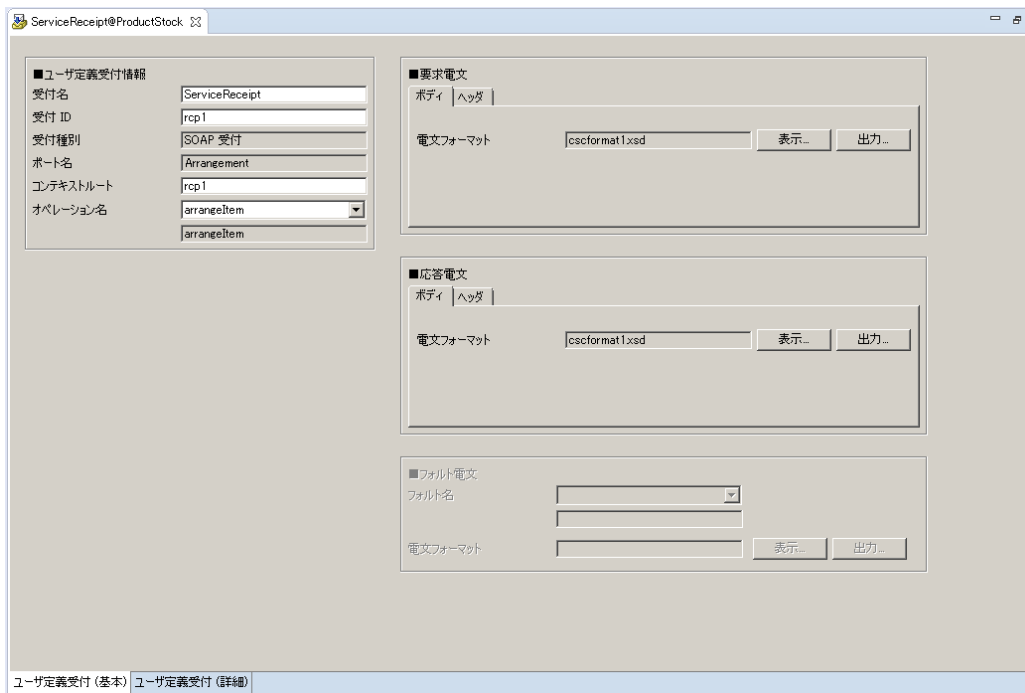
3. [次へ] ボタンをクリックします。
SOAP 受付を追加するためのダイアログが表示されます。
4. 受付名に「ServiceReceipt」を入力し、WSDL ファイルに「ArrangementService.wsdl」を指定します。



5. [次へ] ボタンをクリックします。
6. ドロップダウンリストからポート「Arrangement」を選択して、[終了] ボタンをクリックします。



SOAP 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定義しておく必要があります。ProductStock 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 6-5 ProductStock ビジネスプロセスで使用する変数

変数名	種別
InputData	XML
OutputData	XML
InventoryAllocationInputData	XML
InventoryAllocationOutputData	XML
DeliveryArrangementInputData	XML

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

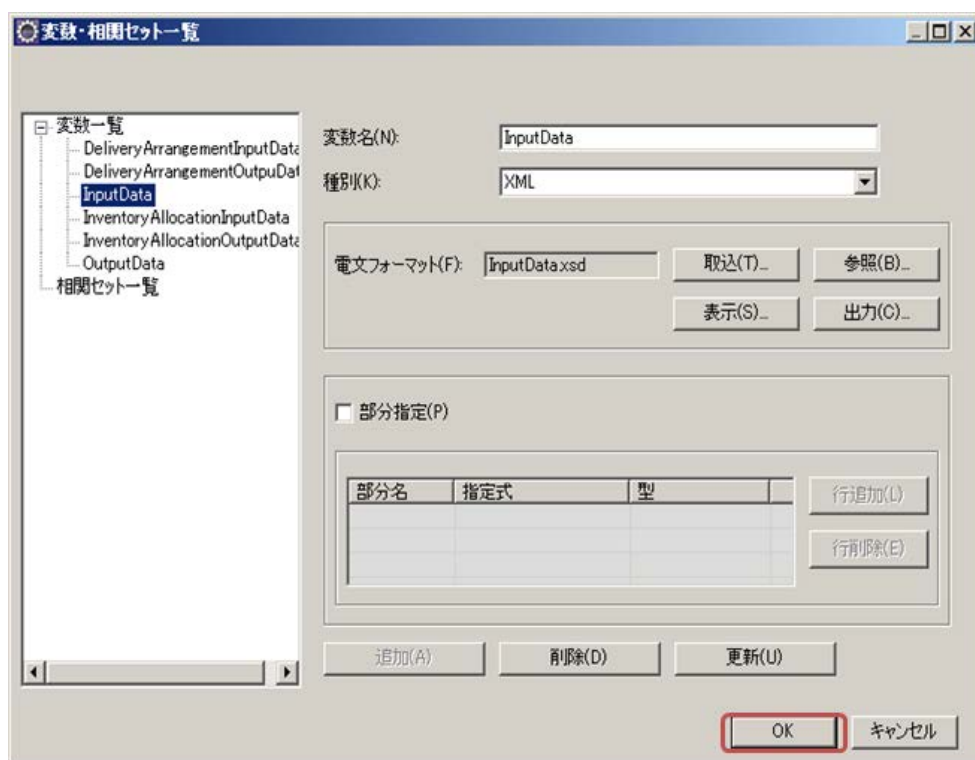


6. [OK] ボタンをクリックします。
[電文フォーマットの取込] ダイアログが閉じます。
7. [変数・相関セット一覧] ダイアログの [追加] ボタンをクリックします。
変数一覧に変数「InputData」が追加されます。
8. 変数「OutputData」, 「InventoryAllocationInputData」, 「InventoryAllocationOutputData」, 「DeliveryArrangementInputData」および「DeliveryArrangementOutputData」についても、手順 2. ~7.と同様の手順で設定します。
設定値は次のとおりです。
[変数・相関セット一覧] 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				
	OutputData	InventoryAllocationInputData	InventoryAllocationOutputData	DeliveryArrangementInputData	DeliveryArrangementOutputData
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	ServiceReceipt	InventoryManagement	InventoryManagement	DeliveryReceipt	DeliveryReceipt
オペレーション名	arrangeItem	reserveItem	reserveItem	deliverItem	deliverItem
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	OutputData	InventoryAllocationInputData	InventoryAllocationOutputData	DeliveryArrangementInputData	DeliveryArrangementOutputData

9. [変数・関連セット一覧] 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

ProductStock 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。

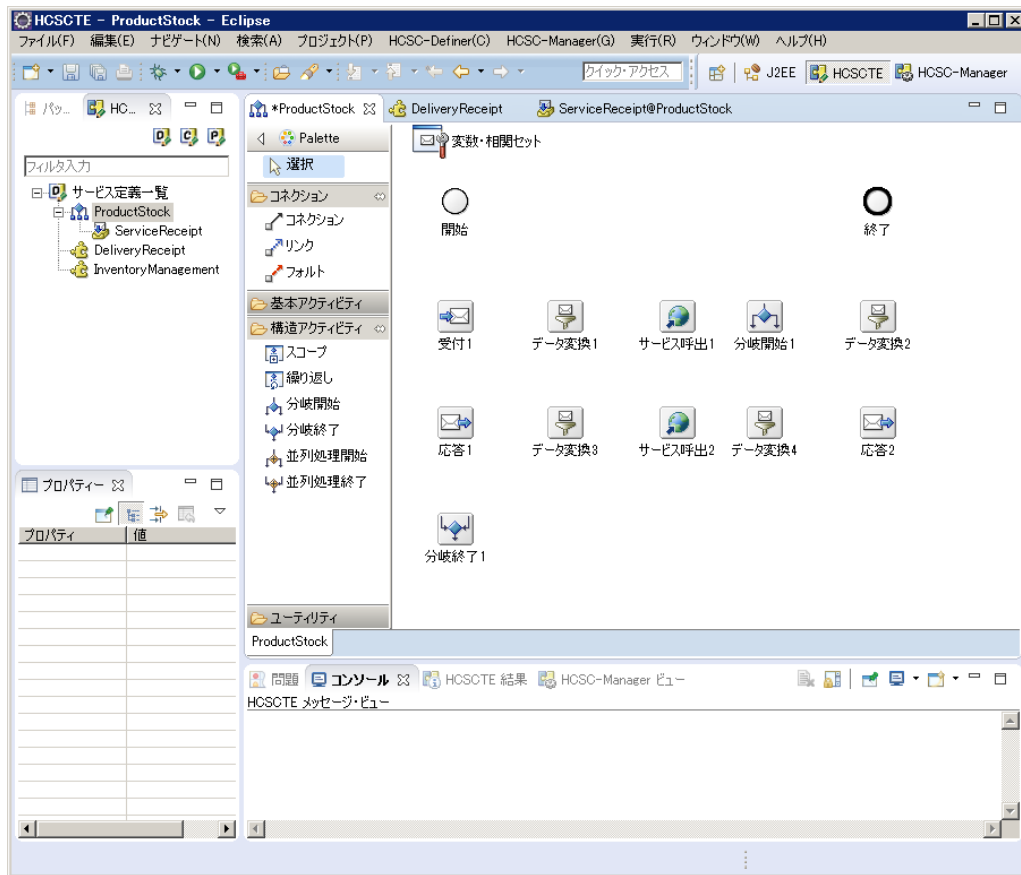
表 6-6 ProductStock ビジネスプロセスに必要なアクティビティ

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。

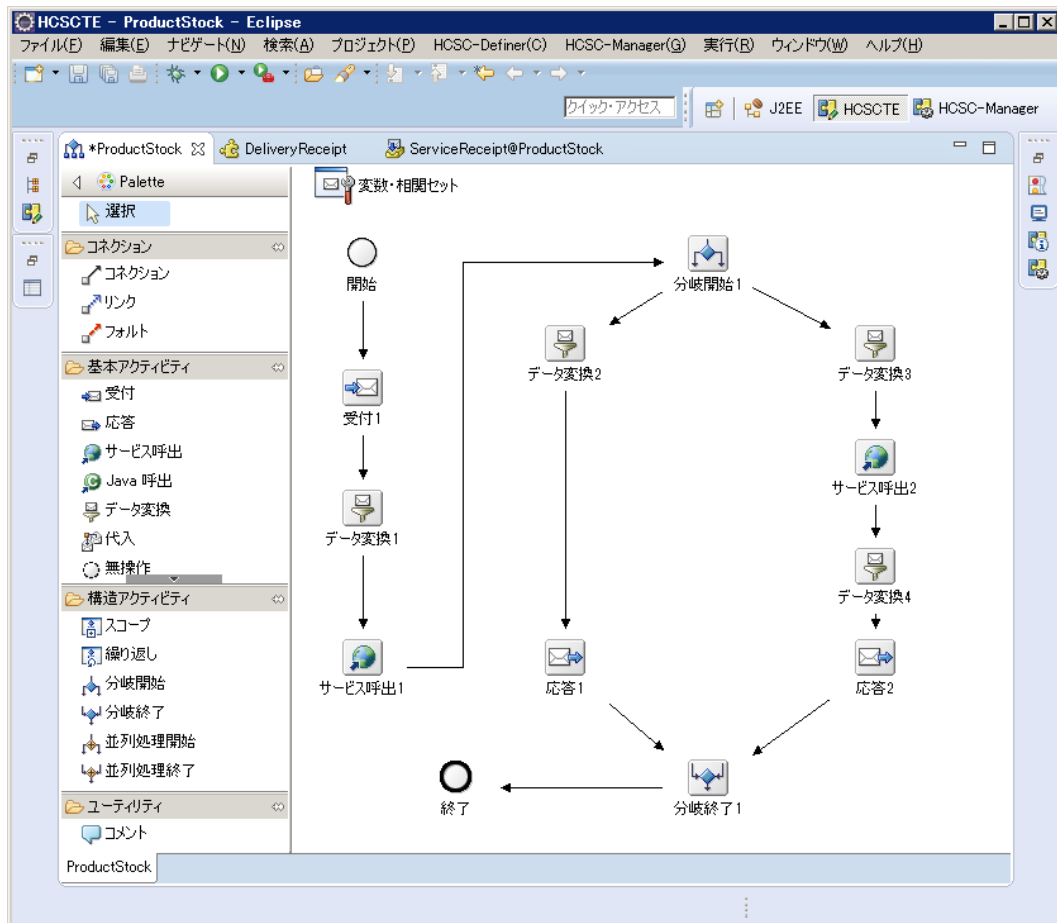
応答アクティビティ	処理結果をサービスリクエストへ返します。
分岐終了アクティビティ	条件（在庫の有無）による処理を終了します。

アクティビティの配置手順を次に示します。

1. パレットからアクティビティをクリックし、図のようにキャンバス上の適切な位置をクリックして配置します。



2. アクティビティを連結するために、パレットの「コネクション」をクリックします。
3. 連結元である開始アクティビティをクリックします。
4. 連結先である受付アクティビティをクリックします。
これで開始アクティビティと受付アクティビティを連結できました。
5. 受付アクティビティから終了アクティビティまでを手順 2.~4.と同様の手順で 1 つずつ連結します。
アクティビティを整列させ、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（受付 1）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	Receive	アクティビティの名称を入力します。
オペレーション名	arrangeItem	サービスリクエストから在庫管理サービスを呼び出すときに利用するオペレーションの名称を入力します。
ボディ割当変数	InputData	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。ProductStock は、各サービス部品の呼び出し結果から応答電文を作成するため、「同期」を設定します。
インスタンス生成	yes	要求電文を受け付けたときに、プロセスを初期化するかどうかを選択します。このサンプルプログラムでは、初期化するため「yes」を設定します。

3. [OK] ボタンをクリックします。

(b) データ変換アクティビティ（在庫引当の前処理の場合）

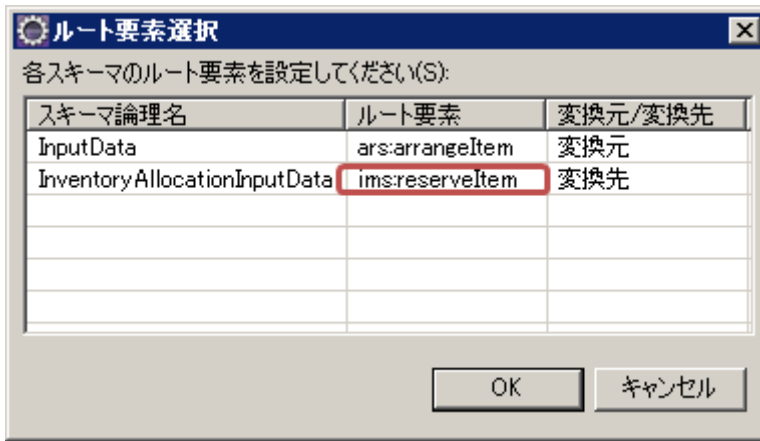
1. キャンバスのデータ変換アクティビティ（データ変換 1）をダブルクリックします。

[データ変換アクティビティ] ダイアログが表示されます。

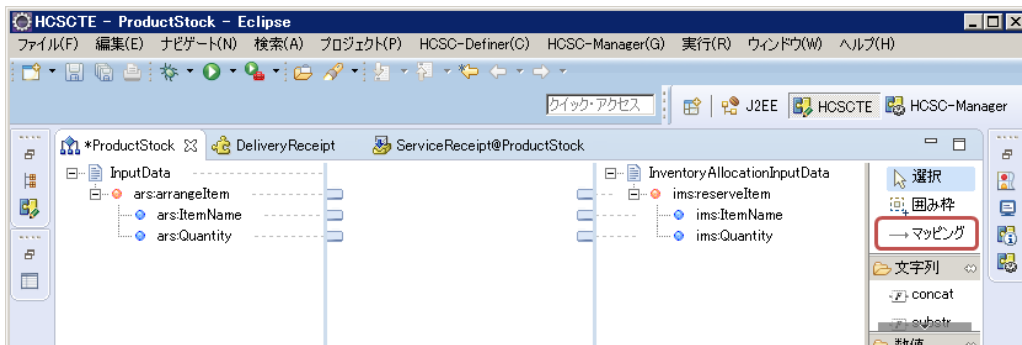
2. 次の内容を入力します。

項目名	設定する値	説明
アクティビティ名	StockAllocationPre-processing	アクティビティの名称を入力します。
変数 (変換元変数)	InputData	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)	InventoryAllocationInputData	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	StockAllocationPre-processing	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。

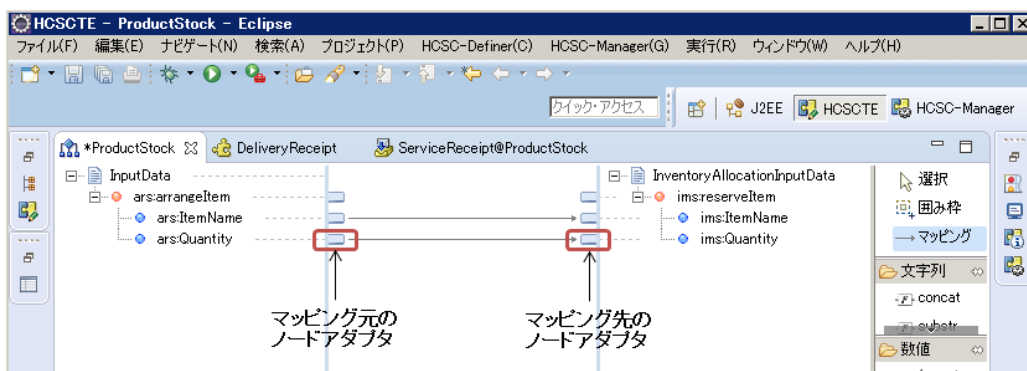
3. [OK] ボタンをクリックします。
4. キャンバスのデータ変換アクティビティを右クリックして、[マッピング定義起動] を選択します。
[ルート要素選択] ダイアログが表示されます。
5. 変換元のスキーマ論理名「InputData」のルート要素をクリックして、ドロップダウンリストから「ars:arrangeItem」を選択します。
6. 変換先のスキーマ論理名「InventoryAllocationInputData」のルート要素をクリックして、ドロップダウンリストから「ims:reserveItem」を選択します。



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。

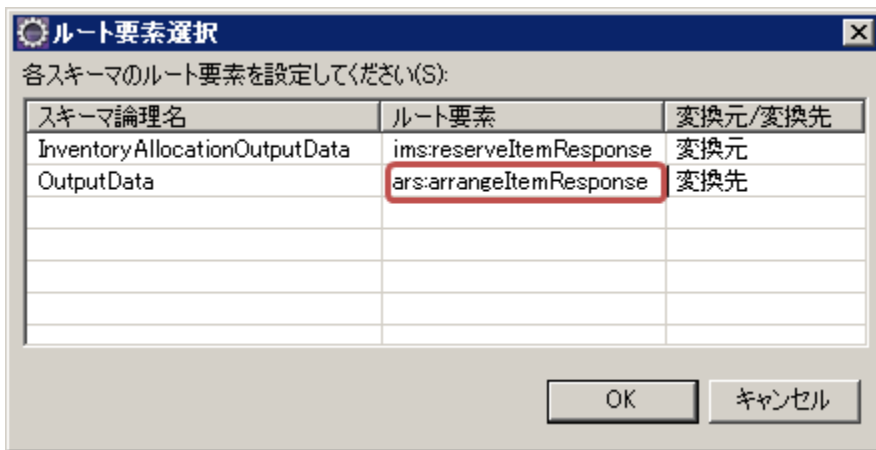


(c) データ変換アクティビティ（在庫なしの場合）

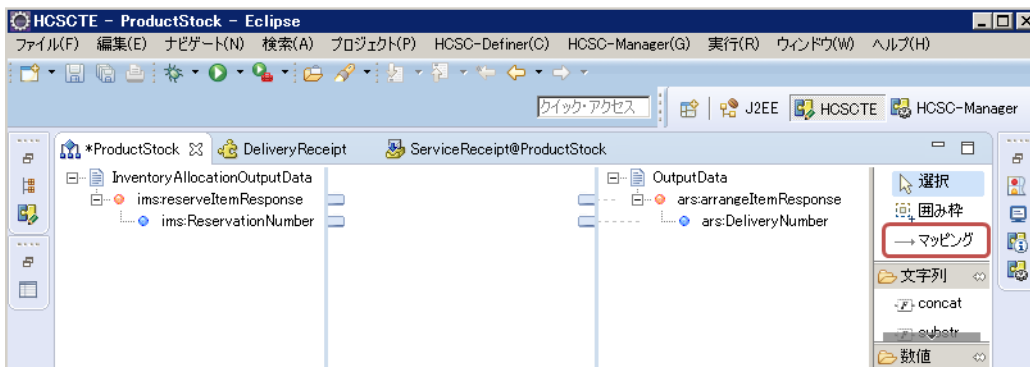
1. キャンバスのデータ変換アクティビティ（データ変換 2）をダブルクリックします。
[データ変換アクティビティ] ダイアログが表示されます。
2. 次の内容を入力します。

項目名	設定する値	説明
アクティビティ名	NoStockArrangement	アクティビティの名称を入力します。
変数 (変換元変数)	InventoryAllocationOutputData	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)	OutputData	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	NoStockArrangement	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。

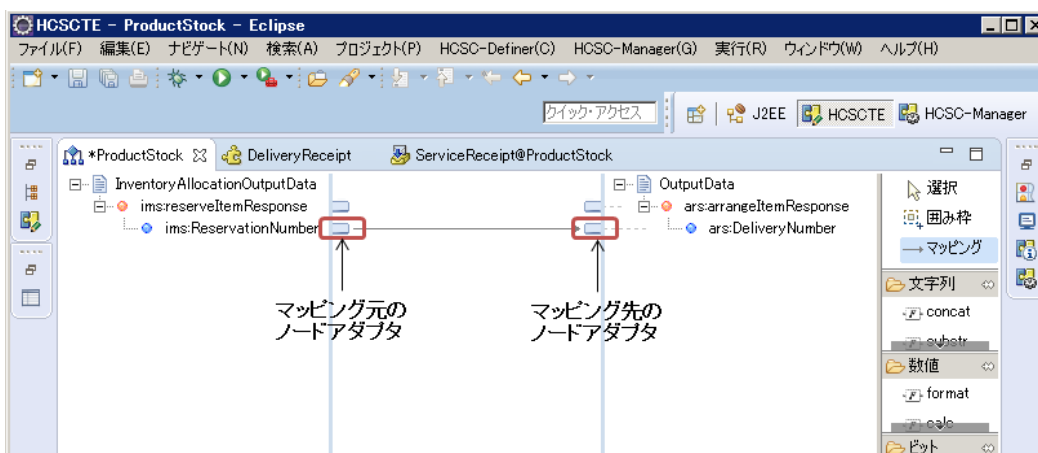
3. [OK] ボタンをクリックします。
4. キャンバスのデータ変換アクティビティを右クリックして、[マッピング定義起動] を選択します。
[ルート要素選択] ダイアログが表示されます。
5. 変換元のスキーマ論理名「InventoryAllocationOutputData」のルート要素をクリックして、ドロップダウンリストから「ims:reserveItemResponse」を選択します。
6. 変換先のスキーマ論理名「OutputData」のルート要素をクリックして、ドロップダウンリストから「ars:arrangeItemResponse」を選択します。



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. キャンバスのデータ変換アクティビティ（データ変換 3）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。

データ変換アクティビティ

アクティビティ名(M): DeliveryArrangementPre-processing

変換元変数

変数(V): InventoryAllocationOutputData 追加(A) 編集(E)...

一覧(S): InventoryAllocationOutputData 削除(D)

変換先変数

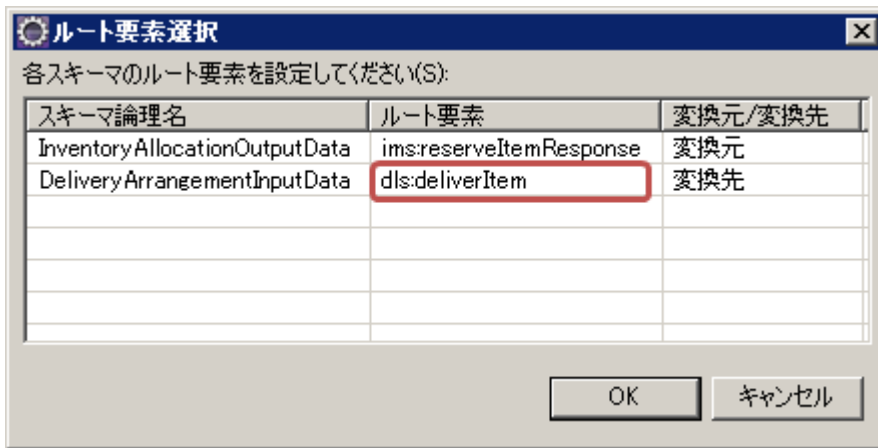
変数(R): DeliveryArrangementInputData 編集(T)...

データ変換定義(F): DeliveryArrangementPre-processing ファイルを削除(X)

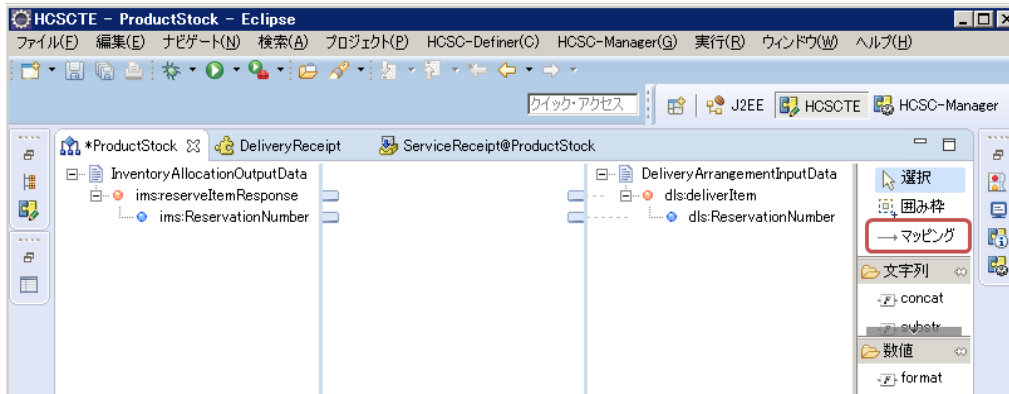
OK キャンセル

項目名	設定する値	説明
アクティビティ名	DeliveryArrangementPre-processing	アクティビティの名称を入力します。
変数 (変換元変数)	InventoryAllocationOutputData	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)	DeliveryArrangementInputData	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	DeliveryArrangementPre-processing	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。

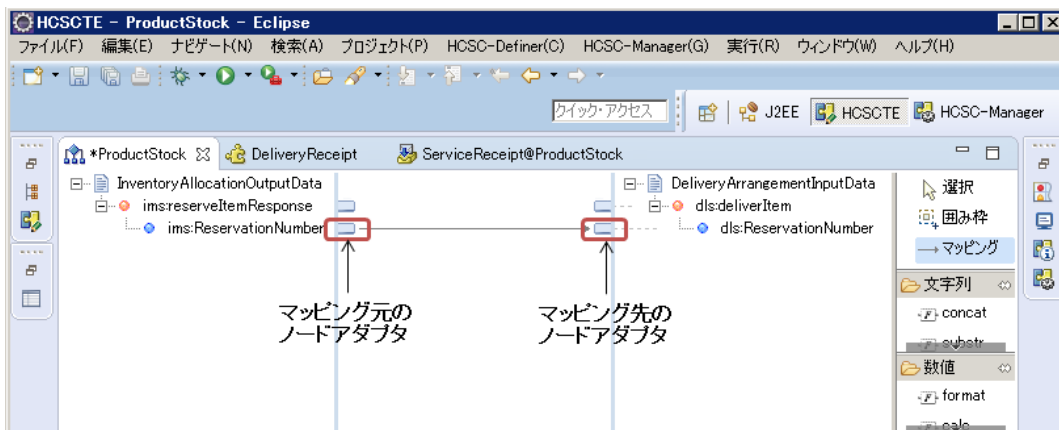
- [OK] ボタンをクリックします。
- キャンバスのデータ変換アクティビティを右クリックして、[マッピング定義起動] を選択します。
[ルート要素選択] ダイアログが表示されます。
- 変換元のスキーマ論理名「InventoryAllocationOutputData」のルート要素をクリックして、ドロップダウンリストから「ims:reserveItemResponse」を選択します。
- 変換先のスキーマ論理名「DeliveryArrangementInputData」のルート要素をクリックして、ドロップダウンリストから「dls:deliverItem」を選択します。



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。



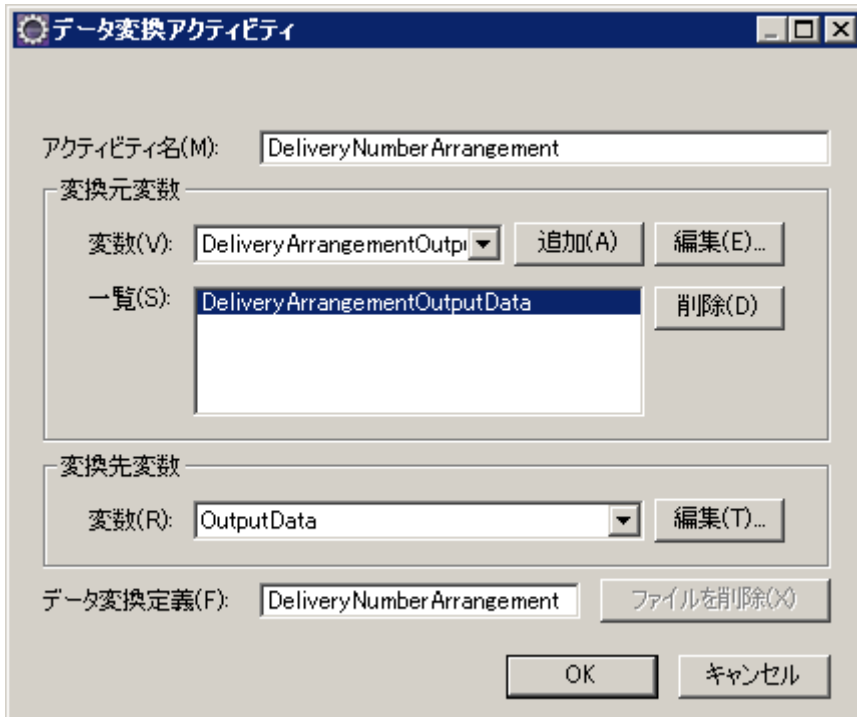
9. マッピング元となる変換元ノードのノードアダプタをクリックします。
10. マッピング先となる変換先ノードのノードアダプタをクリックします。
マッピング線が設定されます。マッピング元のノードアダプタとマッピング先のノードアダプタとの対応は次のとおりです。



(e) データ変換アクティビティ（配送番号の設定の場合）

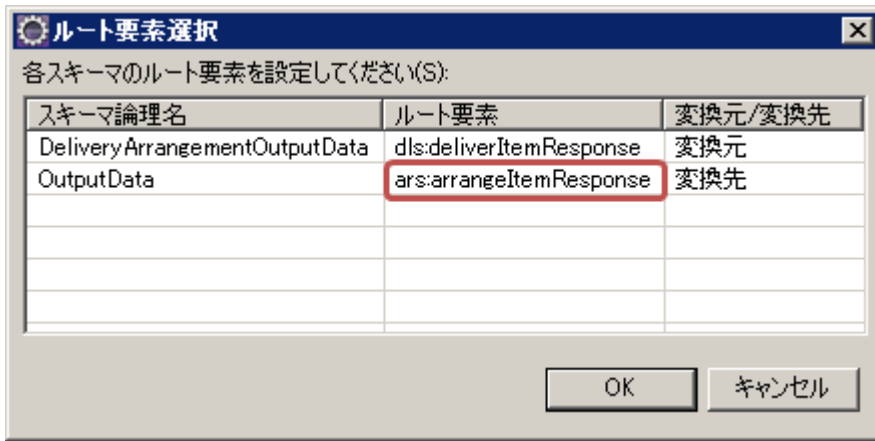
1. キャンバスのデータ変換アクティビティ（データ変換 4）をダブルクリックします。
[データ変換アクティビティ] ダイアログが表示されます。

2. 次の内容を入力します。

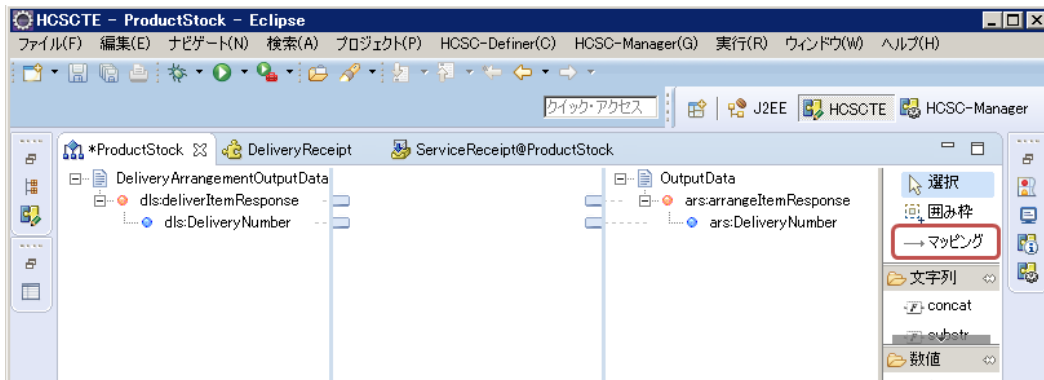


項目名	設定する値	説明
アクティビティ名	DeliveryNumberArrangement	アクティビティの名称を入力します。
変数 (変換元変数)	DeliveryArrangementOutputData	データの変換元になる変数をドロップダウンリストから選択し, [追加] ボタンをクリックします。
変数 (変換先変数)	OutputData	データの変換先になる変数をドロップダウンリストから選択します。
データ変換定義	DeliveryNumberArrangement	変数の変換に使うデータ変換定義ファイルの名称を入力します。

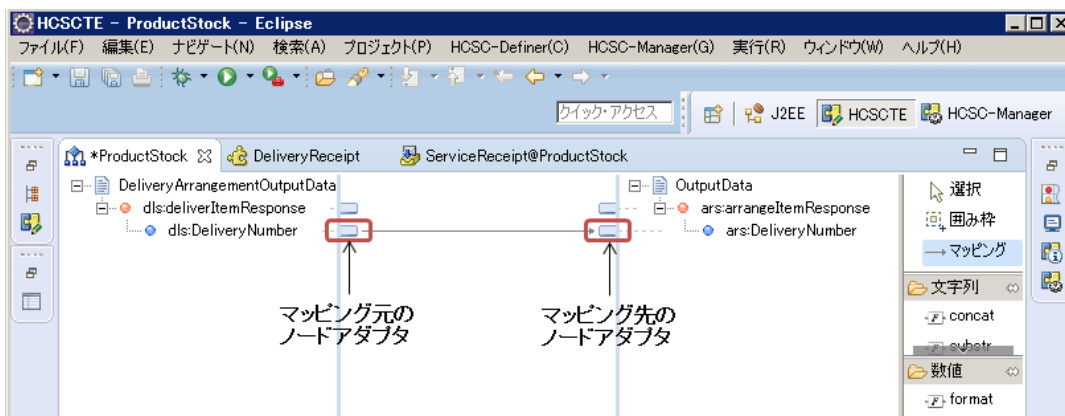
3. [OK] ボタンをクリックします。
4. キャンバスのデータ変換アクティビティを右クリックして, [マッピング定義起動] を選択します。
[ルート要素選択] ダイアログが表示されます。
5. 変換元のスキーマ論理名「DeliveryArrangementOutputData」のルート要素をクリックして, ドロップダウンリストから「dls:deliverItemResponse」を選択します。
6. 変換先のスキーマ論理名「OutputData」のルート要素をクリックして, ドロップダウンリストから「ars:arrangelItemResponse」を選択します。



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。



(f) サービス呼出アクティビティ（在庫を引き当てる場合）

1. キャンバスのサービス呼出アクティビティ（サービス呼出 1）をダブルクリックします。
[サービス呼出アクティビティ] ダイアログが表示されます。
2. 次の内容を入力します。

項目名	設定する値	説明
アクティビティ名	StockAllocation	アクティビティの名称を入力します。
サービス名	InventoryManagement	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)	InventoryAllocationInputData	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)	InventoryAllocationOutputData	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。

3. [OK] ボタンをクリックします。

(g) サービス呼出アクティビティ（配送を手配する場合）

1. キャンパスのサービス呼出アクティビティ（サービス呼出 2）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): DeliveryArrangement

サービス名(S): DeliveryReceipt

オペレーション名(O): deliverItem

通信モデル: 同期

要求電文

ボディ割当変数(Q): DeliveryArrangementInputData 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): DeliveryArrangementOutputData 編集(D)...

ヘッダ割当変数: 設定(D)...

割当関連セット群: 設定(T)...

OK キャンセル

項目名	設定する値	説明
アクティビティ名	DeliveryArrangement	アクティビティの名称を入力します。
サービス名	DeliveryReceipt	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)	DeliveryArrangementInputData	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。
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)	DeliveryArrangementOutputData	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。

3. [OK] ボタンをクリックします。

(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（応答 1）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	Reply_NostockError	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	OutputData	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。
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。

3. [OK] ボタンをクリックします。

(i) 応答アクティビティ（配送手配が成功した場合）

1. キャンバスの応答アクティビティ（応答2）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。

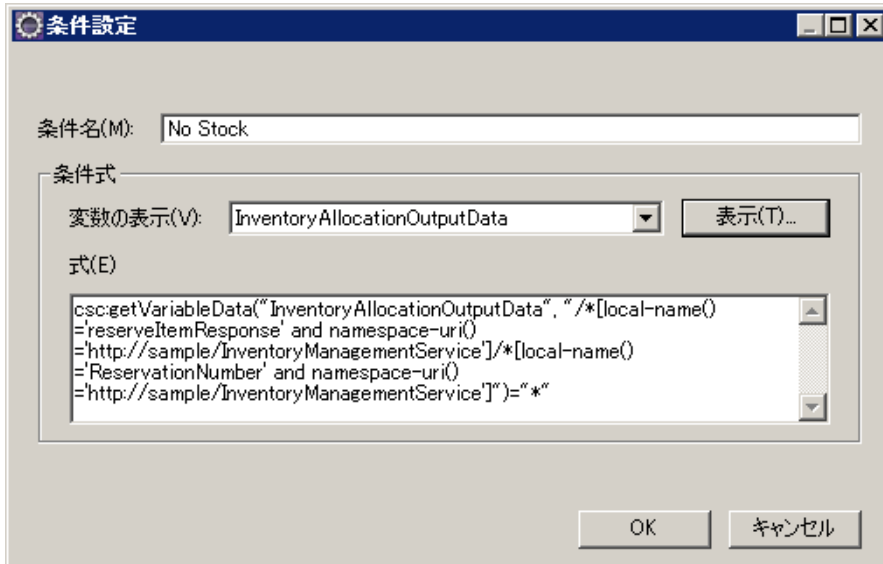
2. 次の内容を入力します。

項目名	設定する値	説明
アクティビティ名	Reply_ArrangementSuccess	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	OutputData	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。
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。

3. [OK] ボタンをクリックします。

(j) 分岐開始アクティビティ

- キャンバスの分岐開始アクティビティ（分岐開始 1）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「StockAllocationResultCheck」を入力します。
- [遷移先] が [NoStockArrangement] 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。

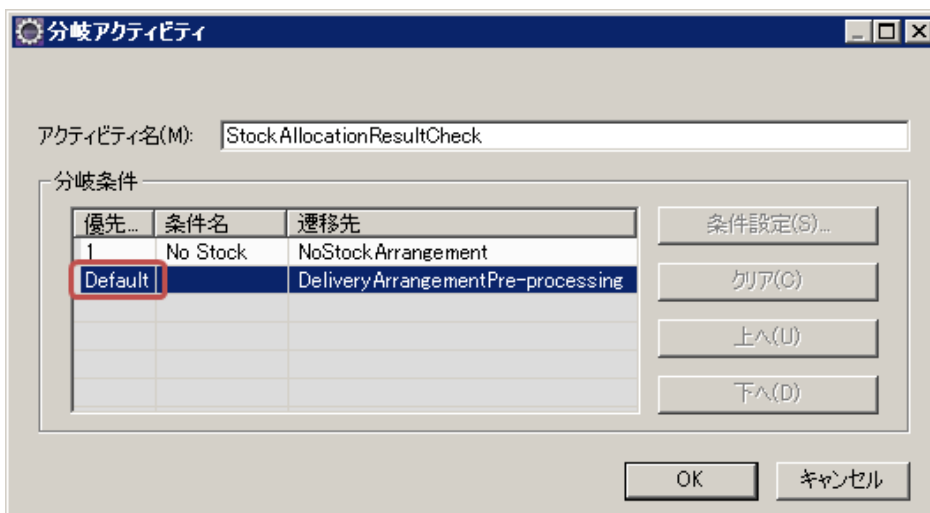


項目名	設定する値	説明
条件名	No Stock	在庫引当チェックの結果、在庫がない場合の条件を指定します。
変数	InventoryAllocationOutputData	条件式で使用する変数をドロップダウンリストから選択します。
条件式*	csc:getVariableData("InventoryAllocationOutputData", "/*[local-name()='reserveItemResponse' and namespace-uri()='http://sample/InventoryManagementService']/*[local-name()='ReservationNumber' 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。

注※

「条件式」の「設定する値」はテキストボックスの幅で折り返して表示されます。

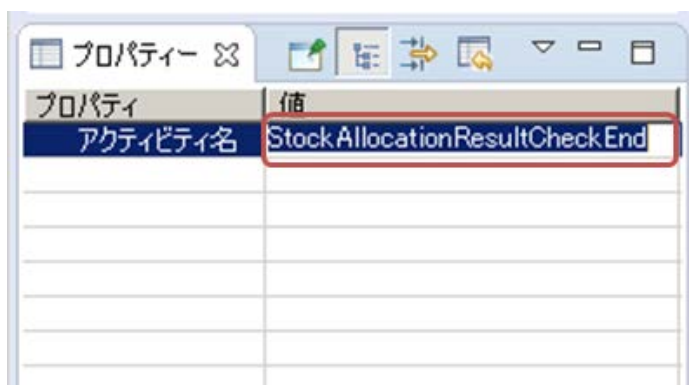
- [OK] ボタンをクリックします。
[分岐アクティビティ] ダイアログに戻ります。
- [遷移先] が [DeliveryArrangementPre-processing] の [優先順位] をクリックし、ドロップダウンリストから「Default」を選択します。
「Default」を選択すると、条件の設定は不要になります。



- [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. キャンバスの分岐終了アクティビティ（分岐終了 1）をクリックします。
プロパティビューに、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、「StockAllocationResultCheckEnd」を入力します。



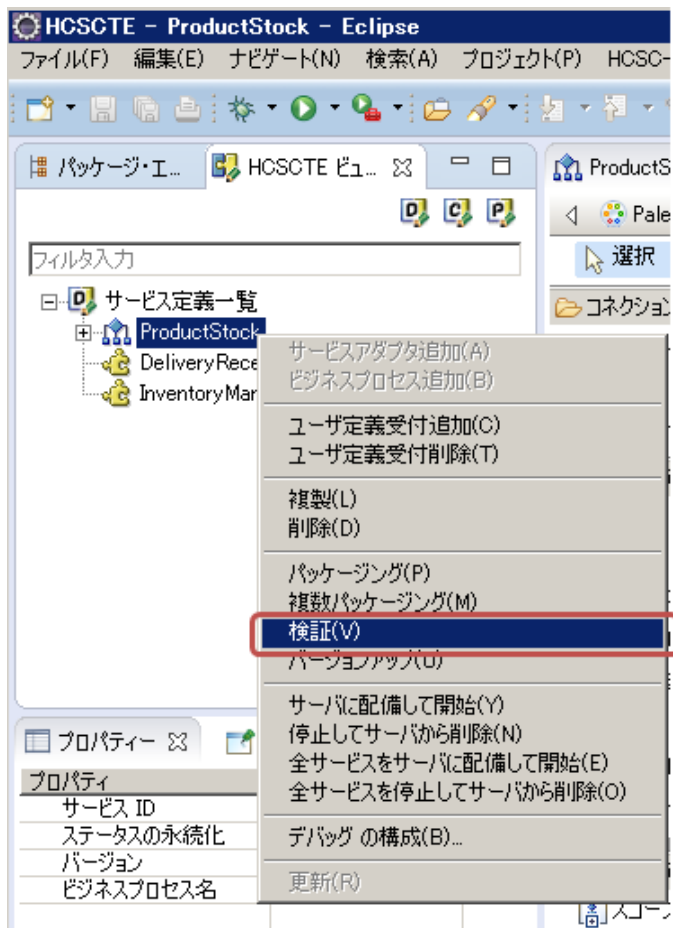
3. すべてのアクティビティを定義したら、メニューから [ファイル] - [保管] を選択して、ビジネスプロセスの定義を終了します。

6.2.4 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。

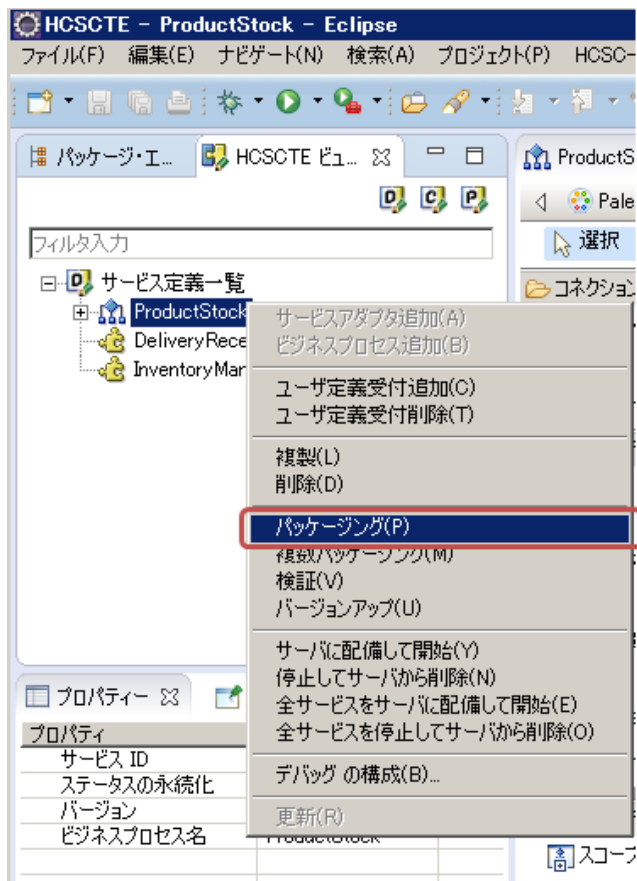
ProductStock ビジネスプロセスの検証とパッケージングの手順を次に示します。

1. ツリービューの [ProductStock] 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 [ProductStock] 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。

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。

4. ツリービューの [DeliveryReceipt] サービスアダプタ、および [InventoryManagement] サービスアダプタに対して、1~3の手順に従って、検証・パッケージングしてください。

6.3 ビジネスプロセスのデバッグ

「6.2.3 ProductStock 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。

ビジネスプロセスをデバッグするためには、サービスリクエストが必要となります。今回は、サンプルプログラムで提供するサービスリクエストを使用します。

6.3.1 では、サンプルプログラムが提供するサービス部品を利用するデバッグについて説明します。サービス部品を使用せずに、サービスエミュレーションの機能を使ってデバッグする場合は、「6.3.2 サービスのエミュレーション機能を利用する場合」を参照してください。

6.3.1 サンプルプログラムが提供するサービス部品を利用するデバッグ

サンプルプログラムが提供するサービス部品を利用する場合、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・サービス部品をデプロイします。
3. HCSC サーバを起動します。
4. デバッガを起動します。
5. サービスリクエストからリクエストを送信し、ビジネスプロセ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. 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

ビジネスプロセ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 (StockAllocation) 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 (StockAllocation) 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。



(2) サービスリクエスト・サービス部品のデプロイ

リクエストの送信およびサービス部品の呼び出しのために、商品手配サンプルプログラムの Eclipse 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・サービス部品を J2EE サーバにデプロイします。

サービスリクエスト・サービス部品をデプロイする手順を次に示します。

1. HCSC サーバを停止します。

[スタート] メニューの [プログラム] から [Cosminexus*] - [テストサーバ停止] を選択して、テスト環境の Performance Tracer, J2EE サーバ, および、標準受付とユーザ定義受付を含む HCSC サーバを停止します。

注※

プログラムフォルダ名を変更している場合、変更したプログラムフォルダから選択してください。

2. Eclipse 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。

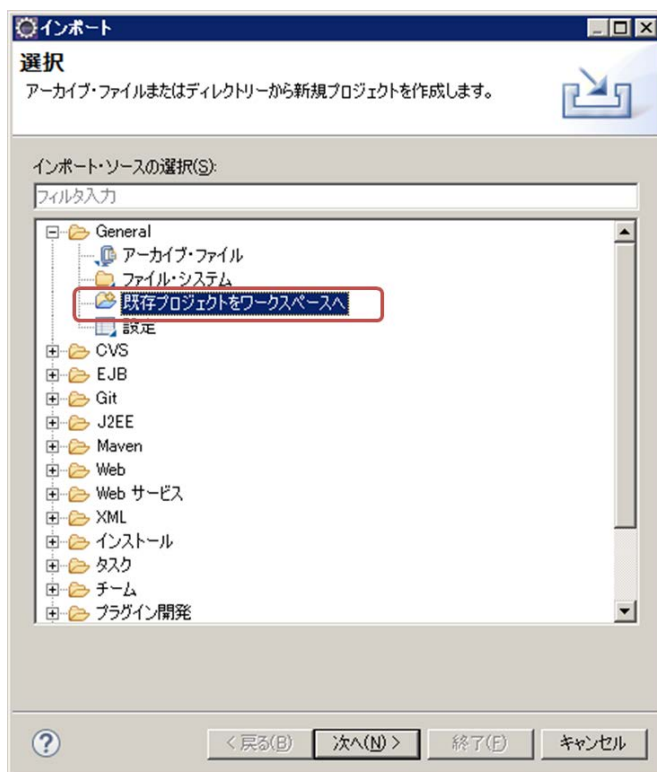
3. [J2EE (デフォルト)] を選択し、[OK] ボタンをクリックします。

[J2EE] 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。

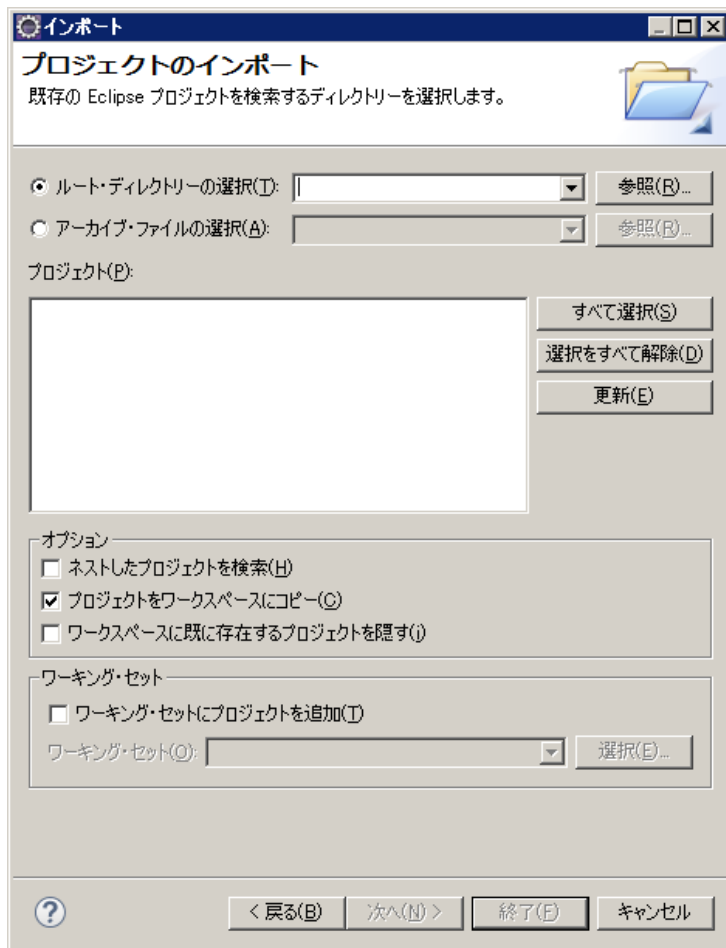
選択画面が表示されます。

5. ツリービューから [General] - [既存プロジェクトをワークスペースへ] を選択します。

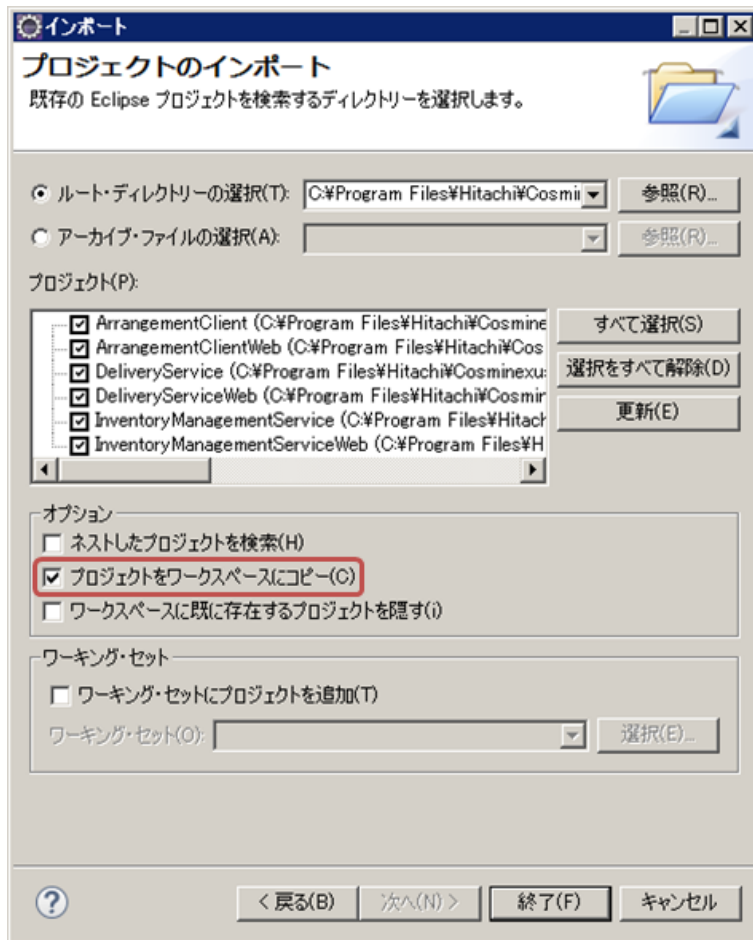


6. [次へ] ボタンをクリックします。

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。
8. サンプルプログラムが格納されているディレクトリ(<サービスプラットフォームのインストールディレクトリ>CSCTE¥Samples¥SOAP1.1_1.2mode¥ProductStock) 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。



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。

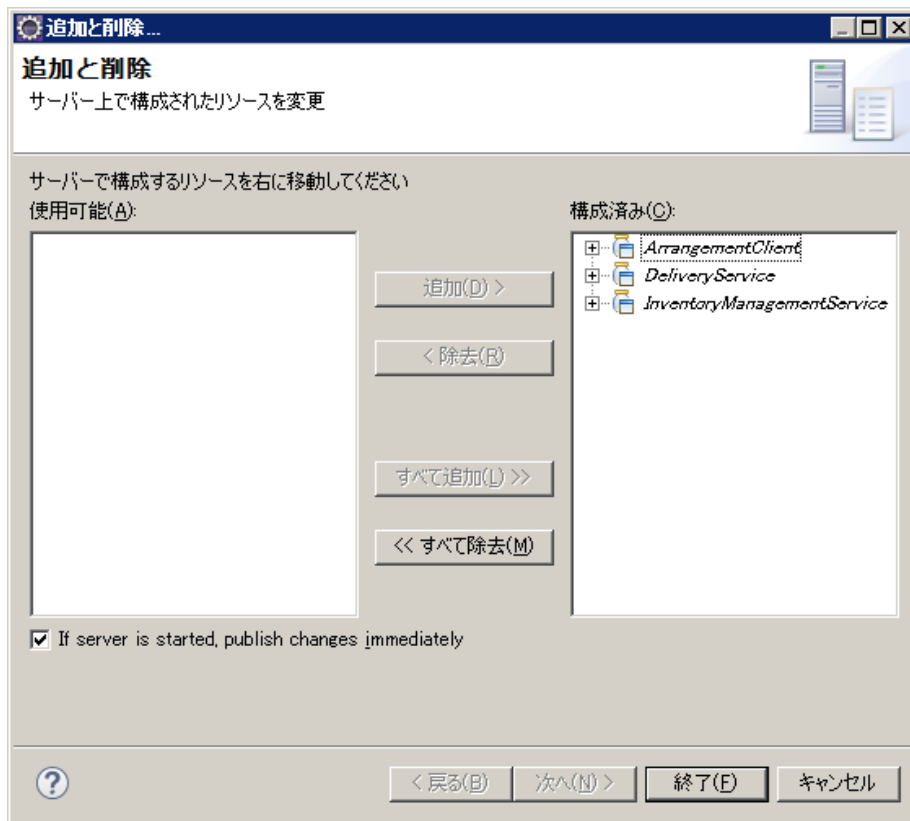
11. [サーバー] ビューの [J2EE Server at localhost] サーバを右クリックし、[デバッグ] を選択します。



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、「4.2.1 (3) 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 [J2EEServer at localhost] サーバを右クリックし、[追加と削除...] を選択します。
[追加と削除...] ダイアログが表示されます。
13. [Available] から [ArrangementClient], [DeliveryService], および [InventoryManagementService] を選択し、[追加] ボタンをクリックします。



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、[J2EE] パースペクティブに戻ります。

設定が終わったら、[サーバー] ビューの [J2EEServer at localhost] の配下に [ArrangementClient]、[DeliveryService]、および [InventoryManagementService] プロジェクトがあ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. HCSC サーバを起動します。

[スタート] メニューの [プログラム] から [Cosminexus*] - [テストサーバ起動] を選択して、テスト環境の Performance Tracer, J2EE サーバ, および、標準受付とユーザ定義受付を含む HCSC サーバを起動します。

注※

プログラムフォルダ名を変更している場合、変更したプログラムフォルダから、選択してください。

(4) デバッガの起動

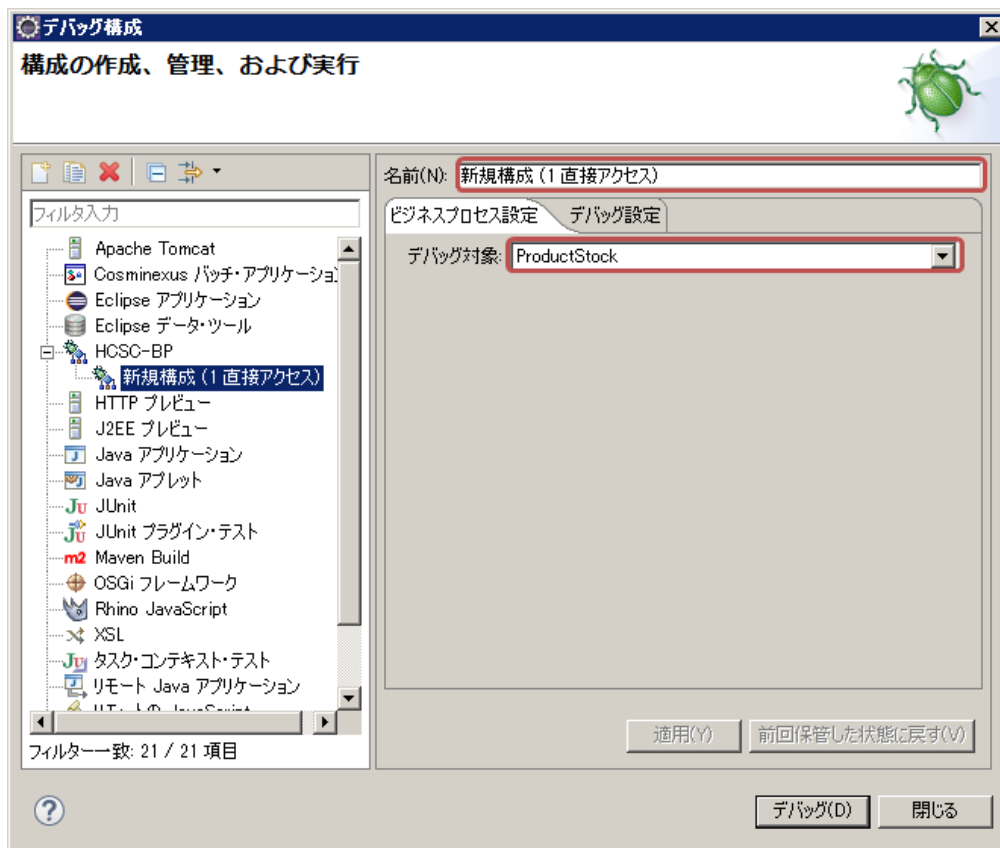
デバッガを起動する手順を次に示します。

1. Eclipse 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「ProductStock」を選択します。



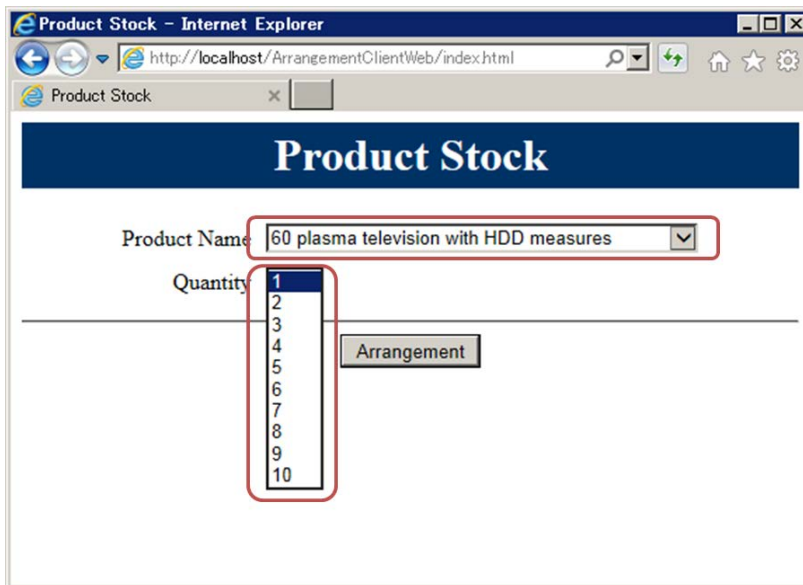
4. [デバッグ] ボタンをクリックします。
アカウント認証画面が表示されます。
5. [ユーザ ID] に「admin」、[パスワード] に「admin」を入力して、[OK] ボタンをクリックします。
処理中であることを知らせるメッセージが表示されます。2 回目以降にデバッガを起動するときは、アカウント認証画面は表示されず、すぐに処理が開始されます。
6. [OK] ボタンをクリックします。
デバッガが起動されます。

(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、ビジネスプロセ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。

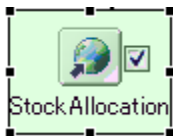


3. [Arrangement] ボタンをクリックします。

Eclipse に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、ビジネスプロセスの処理が開始されます。サービス呼出アクティビティ (StockAllocation) でビジネスプロセ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バ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

ビジネスプロセ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できます。

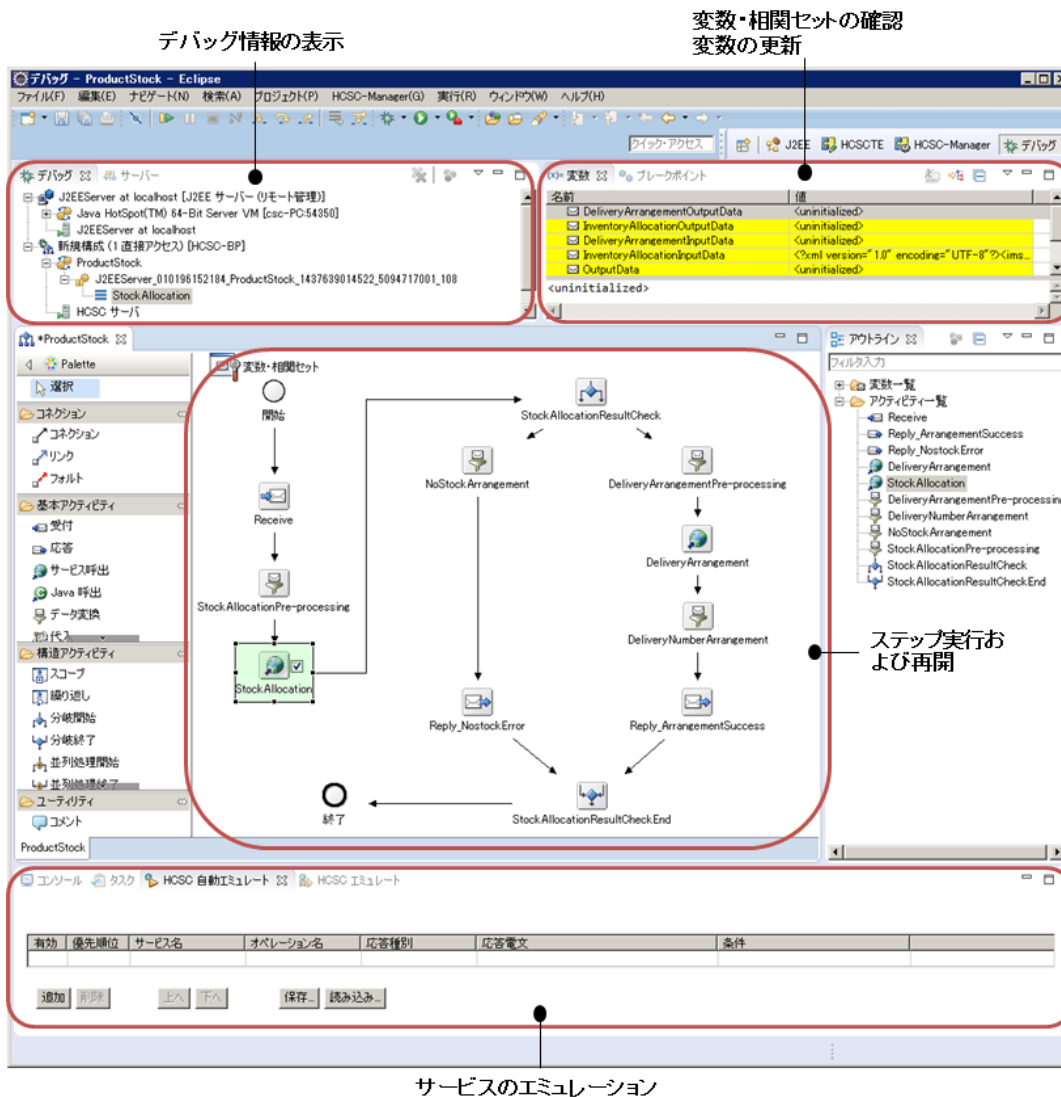






図 6-2 ビジネスプロセスのデバッグ画面

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 6-7 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。
ステップ実行および再開	<p>[デバッグ] 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 () <p>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。</p> <ul style="list-style-type: none"> ステップオーバー () <p>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。</p>

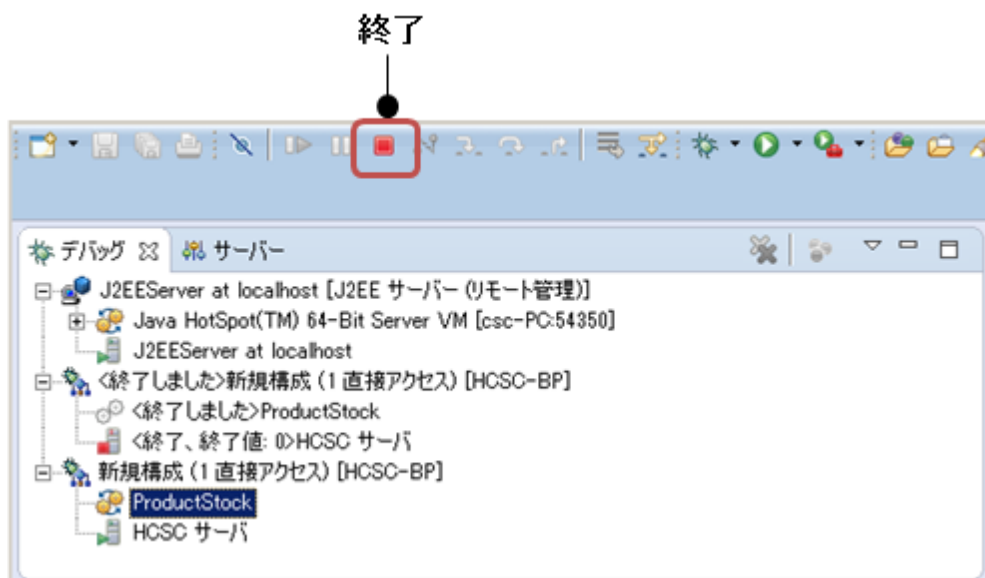
	<ul style="list-style-type: none"> • ステップリターン () <p>スコープアクティビティおよび繰り返しアクティビティの内部のアクティビティで実行した場合は、所属するアクティビティの次のアクティビティまで処理します。</p> <ul style="list-style-type: none"> • 再開 () <p>次にブレークポイントを設定しているアクティビティまで、ビジネスプロセスの処理を再開します。</p>
サービスのエミュレーション	<p>作成済みの応答電文を利用することで、サービスをエミュレーションできます。サービスをエミュレーションする手順については、「6.3.2 サービスのエミュレーション機能を利用する場合」を参照してください。</p>

(7) 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] ビューで「ProductStock」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。

6.3.2 サービスのエミュレーション機能を利用する場合

サービスのエミュレーション機能では、あらかじめサービスの応答電文を作成しておき、設定することで、実際のサービスを呼び出す代わりにすることができます。

サービスのエミュレーション機能を利用することで、サービス呼出アクティビティに設定したサービス部品がなくても、ビジネスプロセスのデバッグ処理を進められます。

サービスのエミュレーションは、ビジネスプロセスの処理が中断しているときに設定できます。ここでは、「6.3.1 (5) リクエストの送信」までの手順を完了し、サービス呼出アクティビティ (StockAllocation) でビジネスプロセスの処理が中断しているものとして、「StockAllocation」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。

「StockAllocation」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、デバッガを起動して、サービスリクエスタからリクエストを送信しておく必要があります。

(1) 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. Eclipse 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] - [保管] を選択します。
XML ファイルが保存されます。

(2) 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. Eclipse のメニューから、[ウィンドウ] - [ビューの表示] - [その他] を選択します。
[ビューの表示] ダイアログが表示されます。
2. [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。

(3) サービスのエミュレーションの実行

1. [デバッグ] ビューで StockAllocation (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 (StockAllocation) からの応答がエミュレーションされます。

ビジネスプロセスの処理が分岐開始アクティビティ (StockAllocationResultCheck) に進みます。

以降のアクティビティへ進むには、[デバッグ] ビューを操作してください。詳細は、「6.3.1 (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。

6.4 開発した商品手配システムを実行する準備

サンプルプログラムで提供するサービスリクエストおよびサービス部品を利用することで、開発したサンプルプログラムの動作を確認できます。

サンプルプログラムの動作を確認する場合は、「4.2 連携する SOAP サービスリクエスト・サービス部品を準備する」の手順を実施したあと、「5 サンプルプログラムを実行する」の手順に従ってサンプルプログラムを実行してください。

注意事項

サンプルで提供するサービスリクエストおよびサービス部品を利用しない場合は、サービスリクエストおよびサービス部品を作成して確認することもできます。サービスリクエストの作成方法はマニュアル「Cosminexus V9 BPM/ESB 基盤 サービスプラットフォーム 開発ガイド基本開発編」を、サービス部品の作成方法はマニュアル「Cosminexus V9 アプリケーションサーバアプリケーション開発ガイド」を参照してください。

7 サンプルプログラムの環境を削除する

この章では、サンプルプログラムの環境の削除手順について説明します。

本章の構成

- 7.1 プロジェクトの削除
- 7.2 テスト環境の停止
- 7.3 アンセットアップとアンインストール

7.1 プロジェクトの削除

サンプルプログラムの削除の手順について説明します。なお、サンプルプログラムをすべて削除したあとに、同じサンプルプログラムを動かす場合は、「4 サンプルプログラムを実行する準備」からやり直してください。

プロジェクトの削除の流れを次の図に示します。

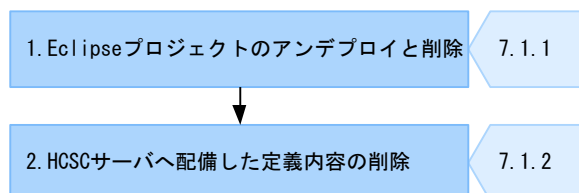


図 7-1 プロジェクトの削除の流れ

それぞれの作業の概要を説明します。プロジェクトの削除を実施する前に **Eclipse** を起動しておいてください。

1. Eclipse プロジェクトのアンデプロイと削除

「4.2.3 Eclipse プロジェクトのデプロイ」でデプロイした **Eclipse** プロジェクトをアンデプロイします。そのあと、「4.2.2 Eclipse プロジェクトのインポート」でインポートした **Eclipse** プロジェクトを削除します。詳細については、「7.1.1 Eclipse プロジェクトのアンデプロイと削除」を参照してください。

2. 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、「4.1.2 HCSC コンポーネントの配備と開始」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、「7.1.2 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。

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。

7.1.1 Eclipse プロジェクトのアンデプロイと削除

Eclipse プロジェクトのアンデプロイと削除の手順を説明します。

(1) Eclipse プロジェクトのアンデプロイ

Eclipse プロジェクトのアンデプロイの手順を次に示します。

注意事項

「3.5.2 HCSCTE プロジェクトの作成」で作成したプロジェクトは削除しないでください。

1. Eclipse 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。

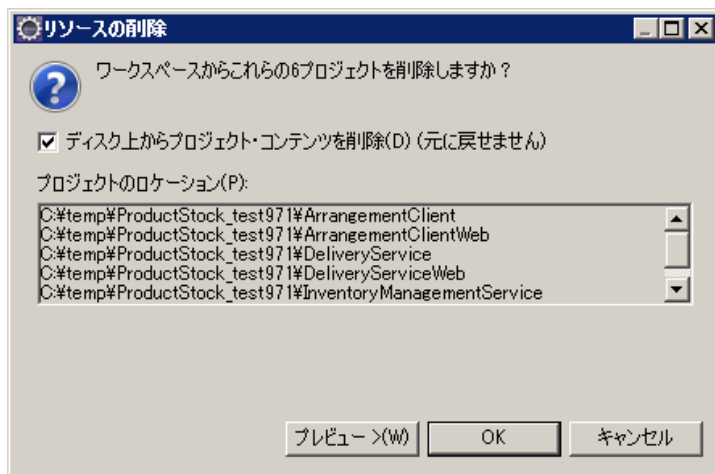
[パースペクティブを開く] ダイアログが表示されます。

2. [J2EE (デフォルト)] を選択し, [OK] ボタンをクリックします。
[J2EE] パースペクティブが表示されます。
3. [サーバー] ビューの [J2EEServer at localhost] を右クリックし, [追加と削除...] を選択します。
[追加と削除...] ダイアログが表示されます。
4. [構成済み] から [ArrangementClient], [DeliveryService] および [InventoryManagementService] を選択して, [除去] ボタンをクリックします。
[ArrangementClient], [DeliveryService] および [InventoryManagementService] が [構成済み] から [使用可能] へ移動します。
5. [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, [J2EE] パースペクティブに戻ります。[サーバー] ビューの [J2EEServer at localhost] の配下から [ArrangementClient], [DeliveryService] および [InventoryManagementService] プロジェクトが削除されていることを確認してください。

(2) Eclipse プロジェクトの削除

Eclipse プロジェクトの削除の手順を次に示します。

1. [プロジェクト・エクスプローラー] ビューから, インポートした Eclipse プロジェクトを選択します。
サンプルプログラムの Eclipse プロジェクトについては, 「4.2.2 Eclipse プロジェクトのインポート」を参照してください。
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。

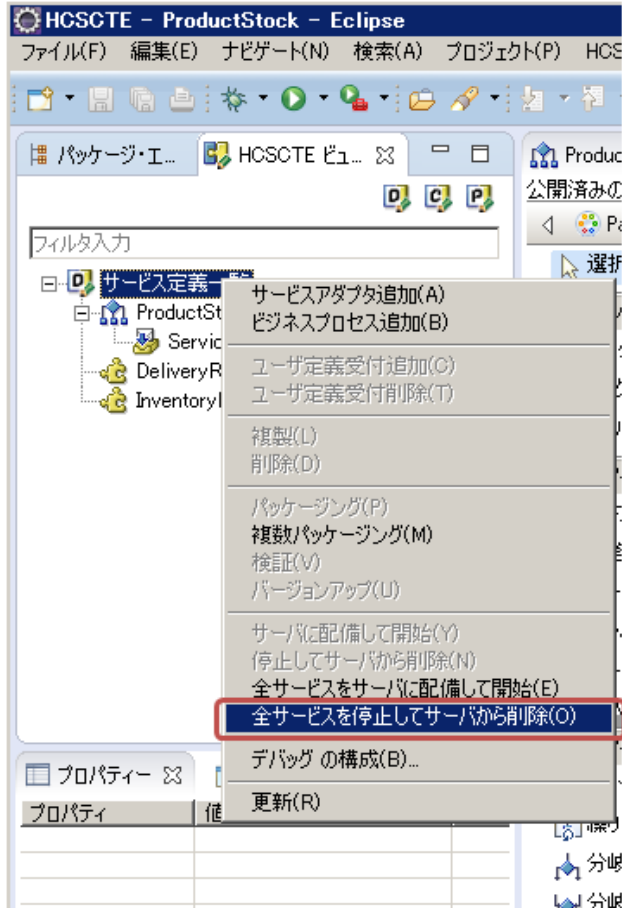


3. [ディスク上からプロジェクト・コンテンツを削除] チェックボックスにチェックし, [OK] ボタンをクリックします。
選択した Eclipse プロジェクトが削除されます。

7.1.2 HCSC サーバへ配備した定義内容の削除

HCSC サーバ上に配備されたサービスアダプタを削除する手順を説明します。

1. HCSCTE ツリービューの [サービス定義一覧] を右クリックして、[全サービスを停止してサーバから削除] 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。これで HCSC サーバへ配備した HCSC コンポーネントの削除が完了しました。

3. HCSCTE プロジェクトから、定義を完全に削除するには、[HCSC-Definer] - [定義情報管理] - [定義情報初期化] を選択します。

初期化を確認するダイアログが表示されるので、[はい] ボタンをクリックします。これで定義が完全に削除されました。

7.2 テスト環境の停止

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手順を次に示します。

1. [スタート] メニューの [プログラム] から [Cosminexus[※]] - [テストサーバ停止] を選択して、テスト環境の Performance Tracer, J2EE サーバ, および, 標準受付とユーザ定義受付を含む HCSC サーバを停止します。
2. [スタート] メニューの [プログラム] から [Cosminexus[※]] - [データベース停止] を選択して、テスト環境の組み込みデータベースを停止します。

注※

プログラムフォルダ名を変更している場合、変更したプログラムフォルダから選択してください。

注意事項

テスト環境を停止する前に Eclipse を終了すると、テスト環境を終了するために、Eclipse 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 Eclipse 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、Eclipse 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、Eclipse を起動します。

7.3 アンセットアップとアンインストール

ここでは、テスト環境のアンセットアップ、Eclipse のアンセットアップ、および Service Architect のアンインストールの手順について説明します。

7.3.1 テスト環境のアンセットアップ

テスト環境をアンセットアップするには、HCSC 簡易セットアップ機能を使います。テスト環境のアンセットアップ方法を次に示します。

1. テスト環境を起動している場合は、テスト環境を停止します。また、Eclipse を起動している場合は、Eclipse を終了します。
2. [スタート] メニューの [プログラム] から [Cosminexus[※]] - [環境構築] - [テスト環境セットアップ] を選択します。
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。
3. [アンセットアップ] ボタンをクリックします。
テスト環境のアンセットアップが開始され、しばらくすると完了します。

注※

プログラムフォルダ名を変更している場合、変更したプログラムフォルダから選択してください。

7.3.2 Eclipse のアンセットアップ

Eclipse セットアップ機能で構築した環境は、Eclipse セットアップ機能でアンセットアップします。

Eclipse セットアップ機能を使用したアンセットアップを実施した場合に削除される項目を次に示します。

表 7-1 Eclipse セットアップ機能のアンセットアップで削除される項目

項目	削除の有無
Eclipse Platform	削除される ^{※1}
Eclipse の configuration フォルダ (ユーザごとのフォルダ)	削除されない ^{※2}
Eclipse のワークスペース	削除されない
Eclipse のショートカット	削除される

注※1

ユーザが作成したファイルも含め、対象となるフォルダ内にあるすべてのデータが削除されます。

注※2

configuration フォルダおよびワークスペースは、Eclipse が生成します。このため、Eclipse セットアップ機能のアンセットアップを実施しても削除されません。削除したい場合は手動で削除してください。手動での削除は、「(2) 手動でアンセットアップする場合」を参照してください。

なお、Eclipse セットアップ機能で構築した環境をアンセットアップする前に、Service Architect をアン

インストールおよび再インストールした場合は、「(2) 手動でアンセットアップする場合」を参照してください。

注意事項

アンセットアップ実行前に、Eclipse を終了してください。Eclipse 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。

(1) Eclipse セットアップ機能を使用する場合

Eclipse セットアップ機能を使用したアンセットアップの手順を次に示します。

1. [スタート]メニューから、[プログラム] - [Cosminexus] - [環境構築] - [Eclipse アンセットアップ] を選択します。

Eclipse セットアップ機能が起動して、[アンセットアップ - Eclipse 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 - Eclipse セットアップ] ダイアログが閉じます。

注意事項

アンセットアップを実行すると、セットアップ実行時にデスクトップに追加した、`eclipse.exe` へのショートカットが削除されますが、画面には表示されたままで、削除されていないように見える場合があります。この場合には、デスクトップを最新の情報に更新すると、ショートカットの削除が画面に反映されます。

(2) 手動でアンセットアップする場合

Eclipse セットアップ機能で構築した環境をアンセットアップする前に、**Service Architect** をアンインストールおよび再インストールした場合は、構築した環境を手動でアンセットアップしてください。手動でアンセットアップする手順を次に示します。

1. Eclipse のショートカットを削除します。
次に示すファイルを管理者が削除します。
C:\Users\Public\Desktop\Eclipse.lnk
2. Eclipse の configuration フォルダを削除します。

フォルダの場所を次に示します。

C:¥Users¥ (全ユーザ分) ¥ADP

3. Eclipse のフォルダを削除します。

デフォルトのフォルダの場所を次に示します。

<サービスプラットフォームのインストールディレクトリ>¥ADP¥IDE¥eclipse

7.3.3 Service Architect のアンインストール

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、Service Architect をアンインストールする前に、必ず Eclipse 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。

アンインストールを確認するダイアログが表示されます。

2. [はい] ボタン、または [いいえ] ボタンをクリックします。

[はい] ボタンをクリックした場合

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

[いいえ] ボタンをクリックした場合

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

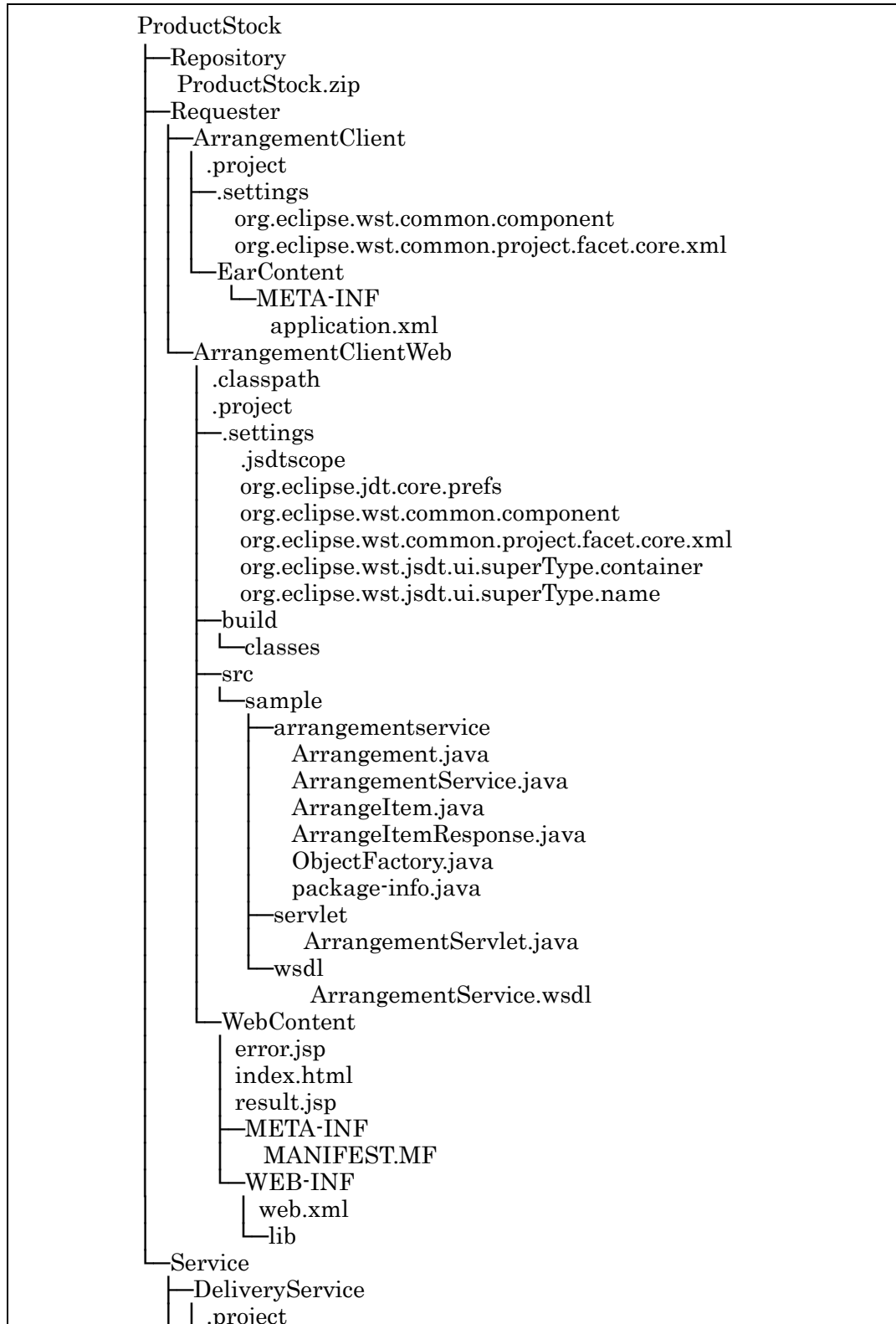
付録

A サンプルプログラムのファイルの構成

商品手配サンプルプログラムは、次の場所にあります。

<サービスプラットフォームのインストールディレクトリ>\¥CSCTE¥Samples¥SOAP1.1_1.2mode¥ProductStock

付録 A.1 商品手配サンプルプログラムの構成




```

├──.settings
│   ├──org.eclipse.wst.common.component
│   └──org.eclipse.wst.common.project.facet.core.xml
├──EarContent
│   └──META-INF
│       └──application.xml
├──DeliveryServiceWeb
│   ├──.classpath
│   ├──.project
│   ├──.settings
│   │   ├──.jsdtscope
│   │   ├──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
│   ├──build
│   │   └──classes
│   ├──src
│   │   └──sample
│   │       ├──deliveryservice
│   │       │   ├──Delivery.java
│   │       │   ├──DeliverItemResponse.java
│   │       │   ├──DeliverItem.java
│   │       │   ├──DeliveryImpl.java
│   │       │   ├──ObjectFactory.java
│   │       └──package-info.java
│   └──WebContent
│       ├──META-INF
│       │   └──MANIFEST.MF
│       ├──WEB-INF
│       │   └──web.xml
│       └──lib
├──InventoryManagementService
│   ├──.project
│   ├──.settings
│   │   ├──org.eclipse.wst.common.component
│   │   └──org.eclipse.wst.common.project.facet.core.xml
│   └──EarContent
│       └──META-INF
│           └──application.xml
├──InventoryManagementServiceWeb
│   ├──.classpath
│   ├──.project
│   ├──.settings
│   │   ├──.jsdtscope
│   │   ├──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
│   ├──build
│   │   └──classes
│   ├──src
│   │   └──sample
│   │       ├──inventorymanagementservice
│   │       │   ├──InventoryManager.java
│   │       │   ├──InventoryManagerImpl.java
│   │       │   ├──ReserveItem.java
│   │       └──ReserveItemResponse.java

```



B Eclipse セットアップ機能実行時の情報の採取

Eclipse 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。

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

Windows の [スタート] メニューから, [プログラム] - [Cosminexus] - [環境構築] - [Eclipse セットアップログ] を選択すると, 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

Eclipse 環境のセットアップログでは, 次の内容を確認できます。

- Eclipse 環境をセットアップしたときの設定内容
- アンセットアップした Eclipse 環境の設定内容

これらの情報は, 実行された順番に 1 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。

なお, Eclipse セットアップ機能実行時のエラーメッセージは, Eclipse セットアップ機能のコンソールに出力されます。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アプリケーションサーバ メッセージ(構築/運用/開発用)」を参照して対処してください。

C 用語解説

ホワイトペーパーで使用する用語については、マニュアル「Cosminexus アプリケーションサーバ & BPM/ESB 基盤 用語解説」を参照してください。