HITACHI
Inspire the Next

Cosminexus V11 BPM/ESB 基盤 サービスプラットフォーム 開発ガイド 受付・アダプタ定義編

手引・操作書

3021-3-J44-80

前書き

■ 対象製品

マニュアル「アプリケーションサーバ & BPM/ESB 基盤 概説」の前書きの対象製品の説明を参照してください。

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、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

Amazon Web Services，AWS，Powered by AWS ロゴ，Amazon Aurora は，Amazon.com, Inc. またはその関連会社の商標です。

Apache(R)，Apache Kafka(R)，および Kafka(R)は，Apache Software Foundation 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。

IBM は，世界の多くの国で登録された International Business Machines Corporation の商標です。

Linux は，Linus Torvalds 氏の米国およびその他の国における登録商標です。

Microsoft，Windows，Windows Server は，マイクロソフト 企業グループの商標です。

Oracle(R)，Java，MySQL 及び NetSuite は，Oracle，その子会社及び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。

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/).

■ ランゲージパックの使用について

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，マニュアル内の表記と異なる場合があります。

■ 発行

2026 年 1 月 3021-3-J44-80

■ 著作権

All Rights Reserved. Copyright (C) 2020, 2026, Hitachi, Ltd.

変更内容

変更内容(3021-3-J44-80) uCosminexus Service Architect 11-70，uCosminexus Service Platform 11-70

追加・変更内容	変更個所
Service Platform および Service Architect から次の適用 OS を削除した。 • Windows 10 x64 • Red Hat Enterprise Linux 8（AMD/Intel 64）	－
XML スキーマを利用した JSON-XML 変換（拡張モード）ができるようにした。 これに伴い，拡張モードに対し，既存の JSON-XML 変換を標準モードとして区別した。	2.5.3
FTP アダプタに RENAME オペレーションを追加し，ファイルのリネームができるようにした。	3.3.10(1)，3.3.19(10)
FTP アダプタのファイル転送の性能向上のため，ログインセッション維持機能を追加した。 これに伴い，MGET オペレーション，MPUT オペレーション，LOGIN オペレーション，および LOGOUT オペレーションの説明を追加した。	3.3.10(1)，3.3.19(10)

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

このマニュアルをお読みになる際の前提情報については，マニュアル「アプリケーションサーバ & BPM/ESB 基盤 概説」のはじめにの説明を参照してください。

目次

前書き　2
変更内容　4
はじめに　5

1　受付およびサービスアダプタの開発の概要　11
1.1　システム全体の中の受付およびサービスアダプタの位置づけ　12
1.2　利用できる受付およびサービスアダプタ　14
1.2.1　利用できる受付　14
1.2.2　利用できるサービスアダプタ　15

2　ユーザ定義受付の定義　17
2.1　ユーザ定義受付の概要　18
2.1.1　SOAP 受付の概要　18
2.1.2　TP1/RPC 受付の概要　19
2.1.3　FTP 受付の概要　20
2.1.4　HTTP 受付の概要　20
2.1.5　Message Queue 受付の概要　20
2.1.6　ファイルイベント受付の概要　20
2.1.7　Kafka 受付の概要　20
2.1.8　スケジュール駆動受付の概要　21
2.1.9　gRPC 受付の概要　21
2.1.10　カスタム受付の概要　21
2.2　SOAP 受付の定義　22
2.2.1　SOAP 受付の定義の流れ　22
2.2.2　SOAP 受付の追加　28
2.2.3　SOAP 受付の定義を編集する　31
2.3　TP1/RPC 受付の定義　32
2.3.1　TP1/RPC 受付の定義の流れ　32
2.3.2　TP1/RPC 受付を追加する　33
2.3.3　TP1/RPC 受付の定義を編集する　35
2.4　FTP 受付の定義　36
2.4.1　FTP 受付の定義の流れ　36
2.4.2　FTP 受付の電文フォーマットを作成する　37
2.4.3　FTP 受付の定義ファイルを作成する　46
2.4.4　FTP 受付を追加する　48

2.5 HTTP 受付の定義 51
2.5.1 HTTP 受付の定義の流れ 51
2.5.2 HTTP 受付の電文フォーマットを作成する 53
2.5.3 HTTP 受付の定義ファイルを作成する 63
2.5.4 HTTP 受付を追加する 64
2.5.5 アクティビティを設定する 66
2.6 Message Queue 受付の定義 70
2.6.1 Message Queue 受付の定義の流れ 70
2.6.2 Message Queue 受付の電文フォーマットを作成する 72
2.6.3 Message Queue 受付の定義ファイルを作成する 75
2.6.4 Message Queue 受付を追加する 76
2.7 ファイルイベント受付の定義 79
2.7.1 ファイルイベント受付の定義の流れ 79
2.7.2 ファイルイベント受付の電文フォーマットを作成する 80
2.7.3 ファイルイベント受付の定義ファイルを作成する 82
2.7.4 ファイルイベント受付を追加する 82
2.8 Kafka 受付の定義 85
2.8.1 Kafka 受付の定義の流れ 85
2.8.2 Kafka 受付の電文フォーマットを作成する 86
2.8.3 Kafka 受付の定義ファイルを作成する 88
2.8.4 Kafka 受付を追加する 88
2.8.5 アクティビティを設定する 90
2.9 スケジュール駆動受付の定義 93
2.9.1 スケジュール駆動受付の定義の流れ 93
2.9.2 スケジュール駆動受付の電文フォーマットを作成する 94
2.9.3 スケジュール駆動受付の定義ファイルを作成する 95
2.9.4 スケジュール駆動受付を追加する 95
2.10 gRPC 受付の定義 98
2.10.1 gRPC 受付の定義の流れ 98
2.10.2 gRPC 受付を追加する 99
2.10.3 フォルト電文フォーマットを修正する 101
2.10.4 アクティビティを設定する 105
2.11 WSDL の作成 111
2.11.1 フォルトが発生した場合にフォルト電文を返す例 111
2.11.2 サービスアダプタの接続先情報を動的に変更する例 130
2.11.3 SOAP 受付の WSDL 作成時の注意事項 135
2.12 ユーザ定義受付の内容の確認 137
2.13 ユーザ定義受付の保存 138
2.14 ユーザ定義受付の検証 139

2.14.1 検証内容 139
2.14.2 検証方法 140
2.15 ユーザ定義受付の情報の変更 141
2.16 ユーザ定義受付の削除 142
2.16.1 1 つのユーザ定義受付を削除する場合 142
2.16.2 複数のユーザ定義受付（呼出先固定）を同時に削除する場合 142

3 サービスアダプタの定義 144
3.1 サービスアダプタの定義の手順 145
3.1.1 定義の流れ 145
3.2 サービスアダプタの追加 149
3.2.1 SOAP アダプタを新規に追加する 149
3.2.2 SessionBean アダプタを新規に追加する 150
3.2.3 MDB（WS-R）アダプタを新規に追加する 151
3.2.4 MDB（DB キュー）アダプタを新規に追加する 151
3.2.5 DB アダプタを新規に追加する 152
3.2.6 TP1 アダプタを新規に追加する 153
3.2.7 ファイルアダプタを新規に追加する 153
3.2.8 Object Access アダプタを新規に追加する 154
3.2.9 Message Queue アダプタを新規に追加する 154
3.2.10 FTP アダプタを新規に追加する 155
3.2.11 ファイル操作アダプタを新規に追加する 156
3.2.12 メールアダプタを新規に追加する 157
3.2.13 HTTP アダプタを新規に追加する 157
3.2.14 コマンドアダプタを新規に追加する 158
3.2.15 SFTP アダプタを新規に追加する 159
3.2.16 Kafka アダプタを新規に追加する 159
3.2.17 gRPC アダプタを新規に追加する 160
3.2.18 汎用カスタムアダプタを新規に追加する 161
3.2.19 定義済みサービスアダプタを利用してサービスアダプタを追加する 161
3.3 サービスアダプタの内容の定義 162
3.3.1 SOAP アダプタを定義する 162
3.3.2 SessionBean アダプタを定義する 167
3.3.3 MDB（WS-R）アダプタを定義する 173
3.3.4 MDB（DB キュー）アダプタを定義する 176
3.3.5 DB アダプタを定義する 179
3.3.6 TP1 アダプタを定義する 216
3.3.7 ファイルアダプタを定義する 223
3.3.8 Object Access アダプタを定義する 242

3.3.9 Message Queue アダプタを定義する 255
3.3.10 FTP アダプタを定義する 273
3.3.11 ファイル操作アダプタを定義する 336
3.3.12 メールアダプタを定義する 357
3.3.13 HTTP アダプタを定義する 377
3.3.14 コマンドアダプタを定義する 404
3.3.15 SFTP アダプタを定義する 411
3.3.16 Kafka アダプタを定義する 421
3.3.17 gRPC アダプタを定義する 427
3.3.18 汎用カスタムアダプタを定義する 438
3.3.19 サービスアダプタ定義での設定一覧 450
3.4 サービスアダプタの保存 497
3.5 サービスアダプタの編集 498
3.6 サービスアダプタの検証 499
3.6.1 検証方法 499
3.6.2 検証内容の表示 499
3.7 サービスアダプタの削除 500

付録 501

付録 A カスタム受付 502
付録 A.1 カスタム受付の概要 502
付録 A.2 カスタム受付の開発 505
付録 A.3 カスタム受付フレームワークの API 511
付録 A.4 カスタム受付の定義 526
付録 A.5 カスタム受付の運用 529
付録 A.6 カスタム受付のチューニング 536
付録 A.7 カスタム受付でのエラーの伝わり方 538
付録 A.8 障害情報の取得（カスタム受付） 543
付録 B カスタムアダプタ開発フレームワーク 555
付録 B.1 カスタムアダプタ開発フレームワークの API 555
付録 B.2 カスタムアダプタ開発フレームワークの定義ファイル 576
付録 B.3 カスタムアダプタ開発フレームワークのサンプルプログラム 584
付録 B.4 汎用カスタムアダプタの定義画面 594
付録 B.5 汎用カスタムアダプタを使用する際の環境設定 601
付録 B.6 トラブルシューティング 601
付録 C DB アダプタ定義支援機能を利用した DB アダプタの定義 606
付録 C.1 DB アダプタ定義支援機能の概要 606
付録 C.2 DB アダプタの定義 606
付録 C.3 分散開発時のエクスポート・インポート 626

付録 D　ファイルアダプタの設定例　627
付録 D.1　サンプルについて　627
付録 D.2　DB アダプタの設定　629
付録 D.3　XML データ読み込みサンプル　632
付録 D.4　CSV データ読み込みサンプル　644
付録 D.5　XML データ書き込みサンプル　657
付録 D.6　CSV データ書き込みサンプル　672
付録 E　ファイル操作アダプタの定義例　689
付録 E.1　XML 形式からバイナリ形式に分割処理方式で変換する場合　689
付録 E.2　バイナリ形式から XML 形式に分割処理方式で変換する場合　693
付録 F　HTTP 受付を利用したビジネスプロセスの設定例　698
付録 F.1　HTTP 受付の設定例（標準モードの場合）　698
付録 F.2　HTTP 受付の設定例（パススルーモードの場合）　709
付録 F.3　HTTP 受付のルーティング機能の設定例　712
付録 G　HTTP 受付および HTTP アダプタを利用したビジネスプロセスの設定例　715
付録 G.1　設定例 1（HTTP サーバから HTTP クライアントにファイルをダウンロードする）　715
付録 G.2　設定例 2（HTTP アダプタの接続先をリクエストごとに動的に変更する）　745
付録 H　FTP 連携で必要なセキュリティの設定　757
付録 H.1　FTPS を使用したセキュアな接続（FTP アダプタ）　757
付録 H.2　セキュアプロトコルによる暗号化・認証（FTP インバウンドアダプタ）　759
付録 H.3　セットアップ時の属性ファイルの定義項目（FTP インバウンドアダプタ）　760
付録 I　HTTP アダプタで必要なセキュリティの設定　762
付録 I.1　HTTPS を使用したセキュアな接続　762
付録 J　メールアダプタで必要なセキュリティの設定　765
付録 J.1　TLS に関する設定　765
付録 J.2　キーストアに関する設定　766
付録 K　gRPC アダプタで必要なセキュリティの設定　768
付録 K.1　キーストアに関する設定　768
付録 L　ユーザ定義受付（呼出先選択）の定義　770
付録 L.1　ユーザ定義受付（呼出先選択）の追加　770
付録 L.2　ユーザ定義受付（呼出先選択）の複製　772
付録 L.3　ユーザ定義受付（呼出先選択）の検証　772
付録 L.4　ユーザ定義受付（呼出先選択）の削除　772
付録 L.5　ユーザ定義受付（呼出先選択）のパッケージングと配備定義　772
付録 L.6　ユーザ定義受付（呼出先選択）のリポジトリ管理　773
付録 M　用語解説　774

索引　775

1 受付およびサービスアダプタの開発の概要

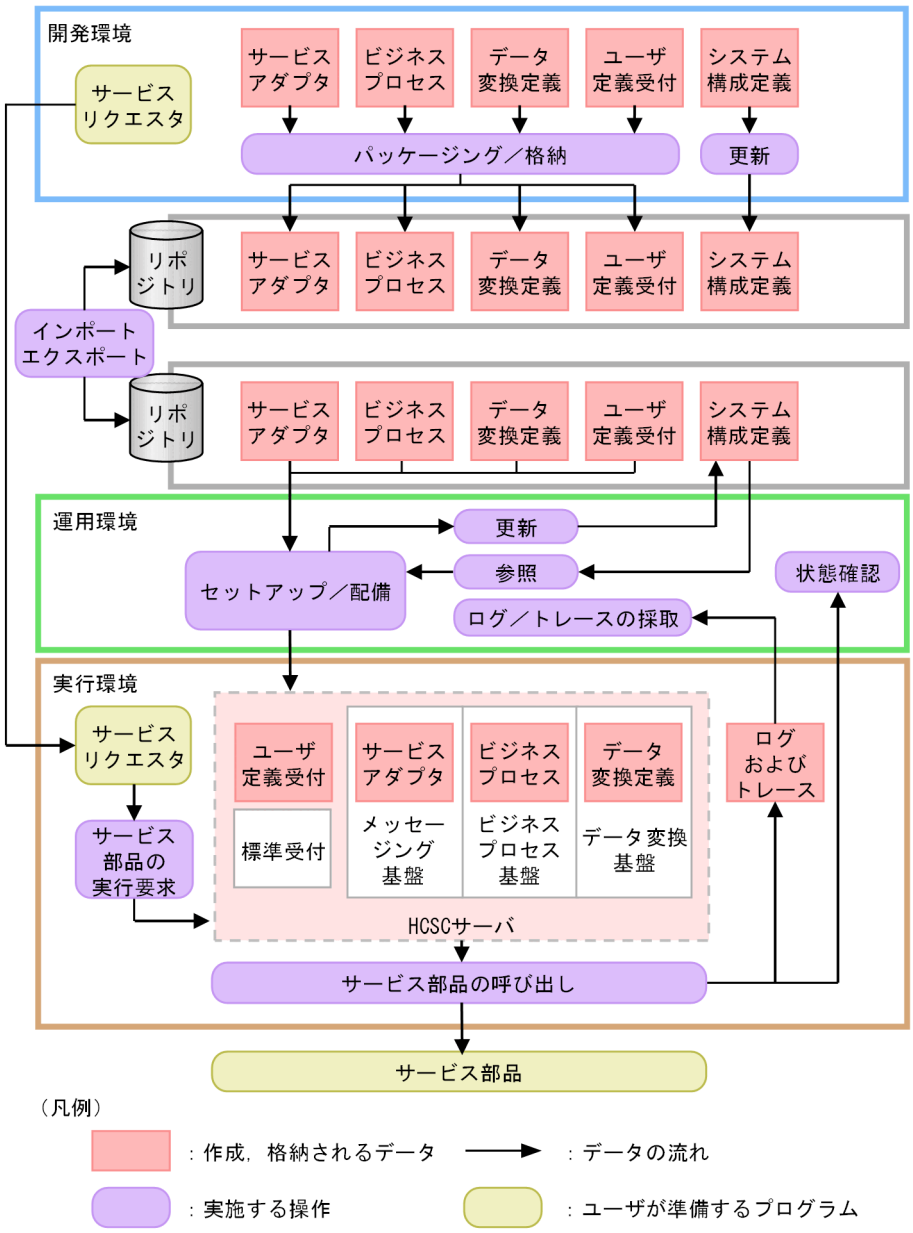
この章では，受付およびサービスアダプタの開発の概要について説明します。

1.1　システム全体の中の受付およびサービスアダプタの位置づけ

サービスプラットフォームでは，受付でサービスリクエスタからの要求電文を受け付け，サービス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します。

受付およびサービスアダプタが，サービスプラットフォームのシステム全体（開発環境，運用環境および実行環境）の中でどのように位置づけられるかを次の図に示します。

図 1-1　受付およびサービスアダプタの位置づけ



受付およびサービスアダプタは開発環境で定義します。定義した受付およびサービスアダプタを含む HCSC コンポーネントや配備定義はリポジトリに格納して，運用環境にデータを渡します。運用環境に渡されたデータは，実行環境に配備されます。

このマニュアルでは，「開発環境」で実施する受付およびサービスアダプタの定義について説明しています。これ以外で「開発環境」で実施する作業については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」を参照してください。また，運用環境および実行環境で実施する作業については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」を参照してください。

1.2　利用できる受付およびサービスアダプタ

サービスプラットフォームで利用できる受付およびサービスアダプタについて説明します。

1.2.1　利用できる受付

受付には，HCSC サーバの機能として提供している標準受付と，ユーザが任意のインターフェースを定義できるユーザ定義受付があります。

ユーザ定義受付には，特定のビジネスプロセスを呼び出すユーザ定義受付（呼出先固定）と，呼び出すビジネスプロセスを選択できるユーザ定義受付（呼出先選択）があります。2 つを区別しない場合は，ユーザ定義受付と表記します。

サービスプラットフォームで利用できる受付の種類を次の表に示します。

表 1-1　利用できる受付の種類

大分類	種別	用途
標準受付	標準受付（Web サービス）	Web サービス（SOAP 通信）を使用したシステムでサービス部品呼び出し要求を受け付ける場合
	標準受付（SessionBean）	SessionBean を使用したシステムでサービス部品呼び出し要求を受け付ける場合
	標準受付（MDB(WS-R)）	MDB（WS-R）を使用したシステムでサービス部品呼び出し要求を受け付ける場合
	標準受付（MDB(DB キュー)）	MDB（DB キュー）を使用したシステムでサービス部品呼び出し要求を受け付ける場合
ユーザ定義受付（呼出先固定）	SOAP 受付	Web サービス（SOAP 通信）を使用したシステムでサービス部品呼び出し要求を受け付ける場合
	TP1/RPC 受付	既存の OpenTP1 システム内にあるサービスリクエスタからのサービス部品呼び出し要求を受け付ける場合
	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場合※1
	HTTP 受付	HTTP クライアントから HTTP 通信を使用してサービス部品呼び出し要求を受け付ける場合
	Message Queue 受付	MQ リソースアダプタを経由して JMS プロバイダ（IBM MQ システム）からの実行要求を受け付ける場合
	ファイルイベント受付	監視対象のフォルダ内にファイルが作成されたことをイベントとして受け付けて，処理を開始する場合
	Kafka 受付	Kafka インバウンドアダプタを経由して Kafka からの実行要求を受け付ける場合
	スケジュール駆動受付	スケジュールを設定してビジネスプロセスを実行する場合

大分類	種別	用途
ユーザ定義受付（呼出先固定）	gRPC 受付	gRPC インバウンドアダプタを経由して，gRPC からの実行要求を受け付ける場合
	カスタム受付	任意のプロトコルからの実行要求を受け付ける場合※2
ユーザ定義受付（呼出先選択）	HTTP 受付	HTTP クライアントから HTTP 通信を使用してサービス部品呼び出し要求を受け付ける場合

注※1

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する場合，FTP アダプタと連携する必要があります。

注※2

任意のプロトコルからの実行要求を受け付けるには，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必要があります。

標準受付を利用するには，それぞれの標準受付の形式に合わせてサービスリクエスタを作成し，さらにサービス部品の入出力形式に合わせて電文を作成，解析する必要があります。標準受付を利用する場合のサービスリクエスタおよび電文の作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「9.　サービスリクエスタの作成」を参照してください。

ユーザ定義受付を利用するには，開発環境で任意のインターフェースを定義したあと，HCSC サーバに配備します。HCSC サーバに配備したあと，ユーザ定義受付を開始すると，サービスリクエスタからの要求電文を受け付けられる状態になります。

ユーザ定義受付を利用するのに必要な定義については，「2.　ユーザ定義受付の定義」を参照してください。

1.2.2　利用できるサービスアダプタ

サービスプラットフォームでは，他システム，データベース，ファイルなど用途に合わせてさまざまなサービスアダプタを提供しています。また，提供しているサービスアダプタでは直接対応できないプロトコルを持つシステムにも対応できるように，カスタムアダプタ開発フレームワークを使用して開発する汎用カスタムアダプタを提供しています。

サービスプラットフォームで利用できるサービスアダプタの種類を次の表に示します。

なお，それぞれのサービスアダプタの定義については，「3.　サービスアダプタの定義」を参照してください。

表 1-2　利用できるサービスアダプタの種類

種別	用途
SOAP アダプタ	Web サービス（SOAP 通信）を利用してサービス部品を呼び出す場合
SessionBean アダプタ	SessionBean を利用してサービス部品を呼び出す場合

種別	用途
MDB（WS-R）アダプタ	WS-R（WS-Reliability）を利用して MDB（Message Driven Bean）で作成したサービス部品を呼び出す場合
MDB（DB キュー）アダプタ	DB キューを利用して非同期のサービス部品を呼び出す場合
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※
TP1 アダプタ	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
ファイルアダプタ	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
Object Access アダプタ	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
Message Queue アダプタ	既存のメッセージキュー（IBM MQ システム）に対してメッセージの送受信をする場合
FTP アダプタ	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場合
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，移動，削除，圧縮，伸張，および一覧情報取得をする場合
メールアダプタ	SMTP プロトコルをサポートしたメールサーバをサービスプラットフォーム上でサービス部品として呼び出す場合
HTTP アダプタ	REST スタイルで公開されている Web サービス（RESTful Web サービス）を呼び出す場合
コマンドアダプタ	オペレーティングシステム上で動作する外部コマンドを起動し，実行結果を取得する場合
SFTP アダプタ	サービスプラットフォームと SFTP サーバとの間で SFTP プロトコルによるファイル転送をする場合
Kafka アダプタ	サービスプラットフォームから Kafka にメッセージを送信する場合
gRPC アダプタ	gRPC で作成されたサービスにメッセージを送信する場合
汎用カスタムアダプタ	サービスプラットフォームで提供するサービスアダプタでは対応できないプロトコルを持つシステムのサービス部品を呼び出す場合

注※

操作するデータベースや実行する SQL を定義する必要があります。

2 ユーザ定義受付の定義

この章では，ユーザ定義受付の定義について説明します。

2.1　ユーザ定義受付の概要

ユーザ定義受付は，個々のサービス部品およびビジネスプロセスの入出力形式に合わせた形式でサービスリクエスタからの要求を受け付けることができます。

ユーザ定義受付には，特定のビジネスプロセスを呼び出すユーザ定義受付（呼出先固定）と，呼び出すビジネスプロセスを選択できるユーザ定義受付（呼出先選択）があります。2 つを区別しない場合は，ユーザ定義受付と表記します。

ユーザ定義受付には次の種類があります。

- SOAP 受付
- TP1/RPC 受付
- FTP 受付
- HTTP 受付※
- Message Queue 受付
- ファイルイベント受付
- Kafka 受付
- スケジュール駆動受付
- gRPC 受付
- カスタム受付

注※　ユーザ定義受付（呼出先選択）は，HTTP 受付で使用できます。

2.1.1　SOAP 受付の概要

SOAP 受付は，ユーザが指定した WSDL 形式のファイルをインターフェースとして使用し，サービスリクエスタと Web サービス（SOAP）で通信します。SOAP 受付でリクエストを受け付けることができる HCSC コンポーネントはビジネスプロセスだけです。

SOAP 受付，WSDL，ビジネスプロセスの定義の関係について，次の図に示します。

図 2-1　SOAP 受付，WSDL，ビジネスプロセスの定義の関係

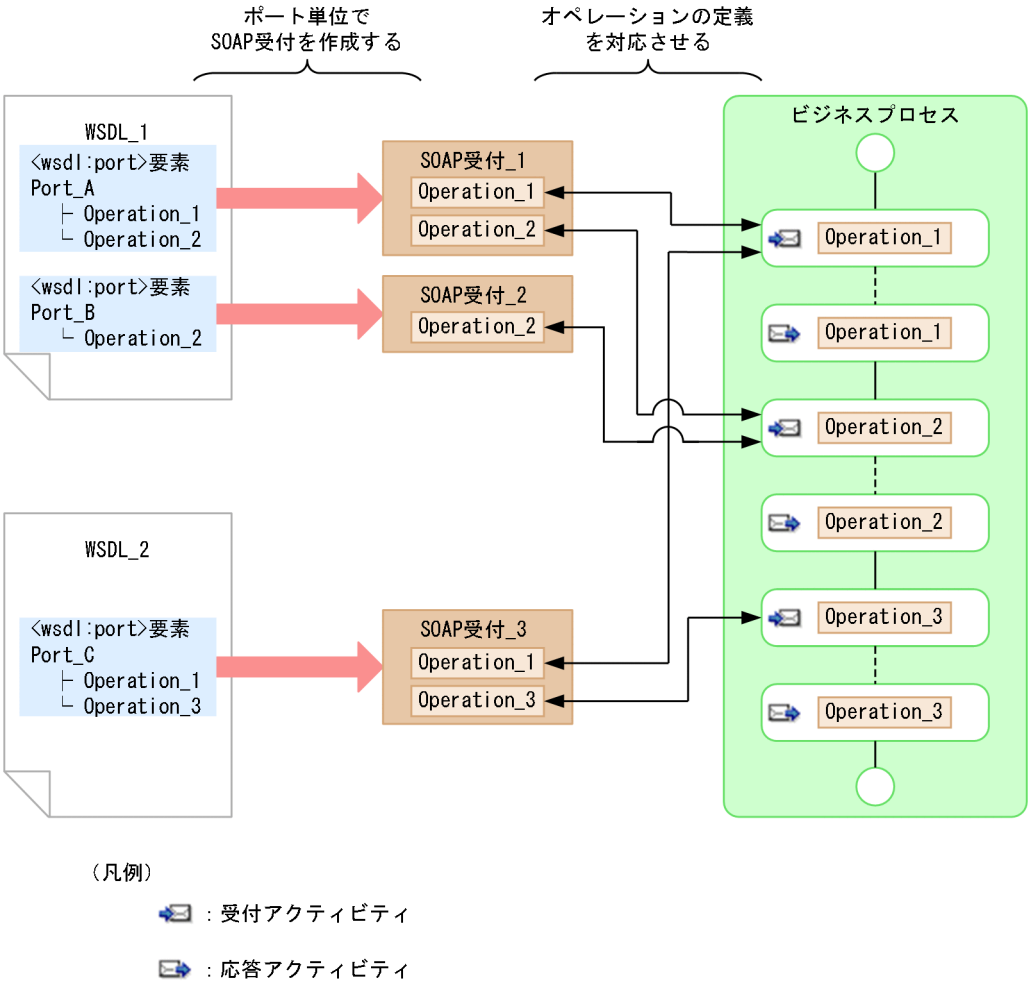


図 2-1 に示すように，SOAP 受付は，WSDL のポート（wsdl:port 要素）ごとに作成します。1 つの SOAP 受付に wsdl:port 要素に含まれるオペレーションがすべて含まれます。

また，図 2-1 の Operation_2 は，Port_A および Port_B の 2 つの wsdl:port 要素に含まれます。このように，異なる wsdl:port 要素に同じオペレーションが含まれる場合，複数の SOAP 受付で同じオペレーションを利用できます。図 2-1 では，Operation_2 は，SOAP 受付_1 および SOAP 受付_2 で利用しています。

2.1.2　TP1/RPC 受付の概要

TP1/RPC 受付は，OpenTP1 システム内にあるサービスリクエスタからの要求を受け付けます。サービスプラットフォームでは，TP1/RPC 受付をユーザ定義受付として提供しています。

TP1/RPC 受付については，「2.3　TP1/RPC 受付の定義」を参照してください。

2.1.3　FTP 受付の概要

FTP 受付は，FTP クライアントと FTP サーバの間でファイル転送をする場合に，FTP クライアントからの要求を受けます。サービスプラットフォームでは，FTP 受付をユーザ定義受付として提供しています。

FTP 受付については，「2.4　FTP 受付の定義」を参照してください。

2.1.4　HTTP 受付の概要

HTTP 受付は，HTTP クライアントからの接続要求を Web フロントシステムや SOAP 受付を経由しないで受け付けることができるインターフェースです。

HTTP 受付については，「2.5　HTTP 受付の定義」を参照してください。

2.1.5　Message Queue 受付の概要

Message Queue 受付は，MQ リソースアダプタを経由して JMS プロバイダ（IBM MQ システム）のキューからの実行要求を受け付けます。サービスプラットフォームでは，Message Queue 受付をユーザ定義受付として提供しています。

Message Queue 受付については，「2.6　Message Queue 受付の定義」を参照してください。

2.1.6　ファイルイベント受付の概要

ファイルイベント受付は，監視対象のフォルダ内にファイルが作成されたことをイベントとして受け付けて，処理を開始します。

ファイルイベント受付については，「2.7　ファイルイベント受付の定義」を参照してください。

2.1.7　Kafka 受付の概要

Kafka 受付は，Kafka インバウンドアダプタを経由して Kafka からの実行要求を受け付けます。サービスプラットフォームでは，Kafka 受付をユーザ定義受付として提供しています。

Kafka 受付については，「2.8　Kafka 受付の定義」を参照してください。

2.1.8　スケジュール駆動受付の概要

スケジュール駆動受付は，スケジュール設定に定義したスケジュールに従って，タスクを実行し，ビジネスプロセスを呼び出します。サービスプラットフォームでは，スケジュール駆動受付をユーザ定義受付として提供しています。

スケジュール駆動受付については，「2.9　スケジュール駆動受付の定義」を参照してください。

2.1.9　gRPC 受付の概要

gRPC 受付は，gRPC クライアントから gRPC インバウンドアダプタを経由して受け取ったメッセージを要求電文に変換して，ビジネスプロセスに渡します。サービスプラットフォームでは，gRPC 受付をユーザ定義受付として提供しています。

gRPC 受付については，「2.10　gRPC 受付の定義」を参照してください。

2.1.10　カスタム受付の概要

カスタム受付は，任意のインターフェースおよび通信プロトコルでサービスリクエスタからの要求を受け付けます。カスタム受付を使用する場合，インターフェースおよび通信プロトコルはユーザが実装する必要があります。

カスタム受付については，「付録 A　カスタム受付」を参照してください。

2.2 SOAP 受付の定義

この節では，SOAP 受付の追加，WSDL の作成など SOAP 受付の定義について説明します。

2.2.1 SOAP 受付の定義の流れ

SOAP 受付の定義の流れには，次の 2 つがあります。

ビジネスプロセスの定義前に SOAP 受付を定義する

次の 2 つの条件を満たす場合，先に WSDL を基に SOAP 受付を作成し，そのあとでビジネスプロセスの受付アクティビティと応答アクティビティを定義します。

- サービスリクエスタとのインターフェースが規定済み
- 規定したインターフェースに従った WSDL が作成済み

この場合，SOAP 受付の作成時に入力した WSDL のインターフェースに合うように，受付アクティビティと応答アクティビティのオペレーションおよび電文定義をビジネスプロセス上で定義する必要があります。

ビジネスプロセスの定義後に SOAP 受付を定義する

すでに定義されたビジネスプロセスがある場合，定義済みのビジネスプロセスに合った SOAP 受付を定義します。SOAP 受付を追加する前に，ビジネスプロセスの受付アクティビティと応答アクティビティのオペレーションおよび割当変数の電文定義に合う WSDL を作成しておく必要があります。

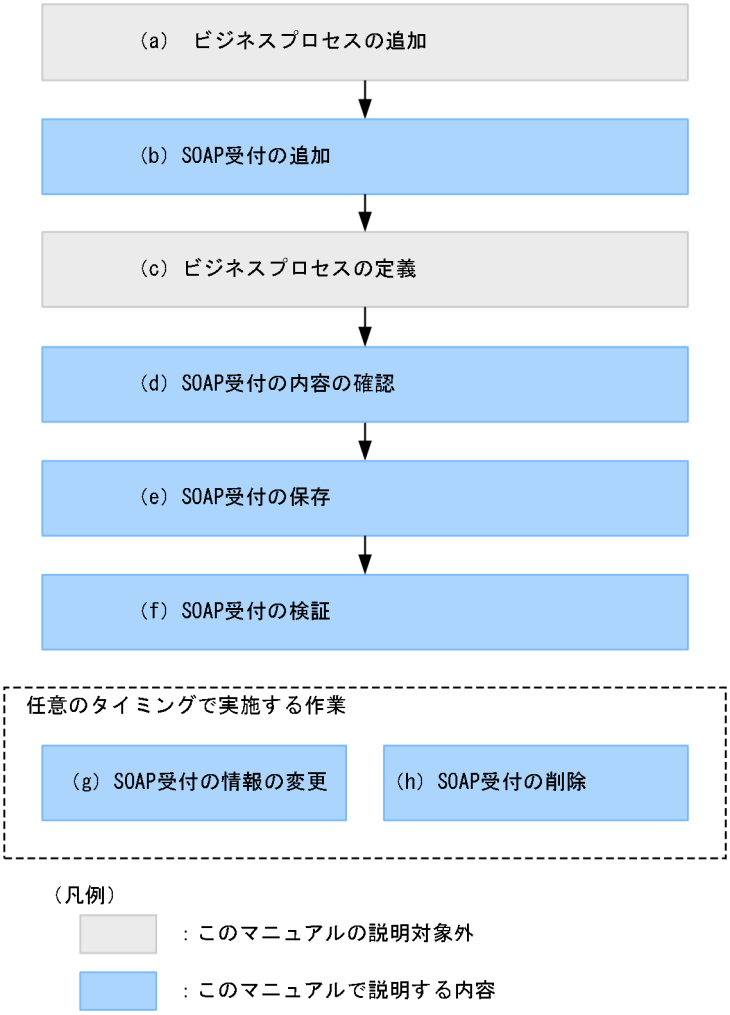
SOAP 受付は，ビジネスプロセスの受付アクティビティと応答アクティビティに定義されているオペレーション，および割当変数を呼び出せるように定義します。

SOAP 受付の定義の流れとして，これら 2 つの定義の流れを次に説明します。

(1) ビジネスプロセスを定義する前に SOAP 受付を定義する

ビジネスプロセスを定義する前に SOAP 受付を定義し，そのインターフェースに合うようにビジネスプロセスの受付アクティビティ，および応答アクティビティを定義する流れについて，次の図に示します。

図 2-2　SOAP 受付の定義の流れ（ビジネスプロセスの定義前に定義する場合）



ビジネスプロセスを定義する前に SOAP 受付を定義する場合の作業について次に説明します。

(a)　ビジネスプロセスの追加

SOAP 受付を定義するビジネスプロセスを作成します。(b)で定義する SOAP 受付の内容と合うインターフェースとなるようにビジネスプロセスを定義します。そのため，ここではアクティビティの配置や変数の定義，各アクティビティの詳細を定義するダイアログでの情報の入力などをする必要はありません。

ビジネスプロセスの追加については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.2.1　ビジネスプロセスを新規に追加する」を参照してください。

(b)　SOAP 受付の追加

作成済みの WSDL をユーザ定義受付追加ウィザードで入力して，SOAP 受付を追加します。

SOAP 受付を新規に追加する方法の詳細については，「2.2.2　SOAP 受付の追加」を参照してください。

(c) ビジネスプロセスの定義

SOAP 受付で定義されているすべてのオペレーションに対応する受付アクティビティ，および応答アクティビティを定義します。

受付アクティビティおよび応答アクティビティで，SOAP 受付で規定された電文フォーマットを持つ変数を定義し，割当変数として設定する手順を次に示します（次の手順は受付アクティビティの場合です。応答アクティビティの場合も同様の手順です）。

1. **[受付アクティビティ] ダイアログで，[オペレーション名] の [取込] ボタンをクリックします。**
   [オペレーション名の取込] ダイアログが表示されます。

2. **[受付名] に該当する SOAP 受付の名称を設定し，[オペレーション名] に取り込み対象となるオペレーション名を設定します。**

3. **[OK] ボタンをクリックします。**

4. **[ボディ割当変数] の [編集] ボタンをクリックします。**
   [変数・相関セット一覧] ダイアログが表示されます。

5. **変数名を設定します。**

6. **[取込] ボタンをクリックします。**
   [電文フォーマットの取込] ダイアログが表示されます。[電文フォーマットの取込] ダイアログには次の情報が表示されています。

   受付名

   手順 1.で設定したオペレーションを持つ SOAP 受付の受付名が設定されています。

   オペレーション名

   手順 1.で設定した値が設定されています。

   電文種別

   受付アクティビティの場合は要求電文が，応答アクティビティの場合は応答電文が選択されています。

   電文フォーマット名

   変数名が設定されています。

   なお，手順 1.で設定したオペレーション名が SOAP 受付に存在しない場合，受付名およびオペレーション名が空白になります。この場合，手順 1 に戻って正しいオペレーション名を指定してください。

   また，電文フォーマット名には，初期値として変数名が設定されています。必要に応じて適当な名称に変更してください。

7. **[OK] ボタンをクリックします。**
   [電文フォーマットの取込] ダイアログが閉じられ，[変数・相関セット一覧] ダイアログが操作できる状態になります。

8. **[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。**

9. [変数・相関セット一覧] ダイアログで電文のヘッダに割り当てる変数を追加します。
   変数を追加する手順については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.5.1　変数の定義」の変数の定義方法の説明を参照してください。

10. [変数・相関セット一覧] ダイアログの変数一覧から追加した変数を選択します。

11. [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。
   ［変数・相関セット一覧］ダイアログが閉じられます。

12. [受付アクティビティ] ダイアログで，［ヘッダ割当変数］の［設定］ボタンをクリックします。
   ［ヘッダ割当変数］ダイアログが表示されます。

13. [ヘッダ割当変数] ダイアログの［追加］ボタンをクリックします。

14. [ヘッダ割当変数] ダイアログの割当変数から追加した変数を選択します。

15. [ヘッダ割当変数] ダイアログの［OK］ボタンをクリックします。

［受付アクティビティ］ダイアログについては，マニュアル「サービスプラットフォーム リファレンス」の「1.4.7　受付アクティビティダイアログ」を参照してください。［変数・相関セット一覧］ダイアログについては，マニュアル「サービスプラットフォーム リファレンス」の「1.4.1　変数・相関セット一覧ダイアログ」を参照してください。［電文フォーマットの取込］ダイアログについては，マニュアル「サービスプラットフォーム リファレンス」の「1.4.5　電文フォーマットの取込ダイアログ」を参照してください。

ポイント

SOAP 受付のオペレーションにフォルトが含まれている場合，ビジネスプロセスではフォルト名を指定した応答アクティビティを定義します。［応答アクティビティ］ダイアログの［フォルト名］に，ユーザ定義受付定義画面の［フォルト電文］の［フォルト名］に表示されている値を指定してから，手順 2.～9.と同様の方法で変数を割り当てます。

また，SOAP 受付のオペレーションに複数のフォルトが含まれている場合，含まれるフォルトの数だけ応答アクティビティを追加し，それぞれに対応するフォルト名を設定する必要があります。また，各応答アクティビティで，手順 2.～9.と同様の方法で変数を割り当てます。

(d)　SOAP 受付の内容の確認

追加された SOAP 受付の内容を確認します。SOAP 受付の内容は，ビジネスプロセスの定義内容と合致している必要があります。

SOAP 受付の確認の詳細については，「2.12　ユーザ定義受付の内容の確認」を参照してください。

(e)　SOAP 受付の保存

定義した SOAP 受付をリポジトリに保存します。

SOAP 受付を保存する方法については，「2.13　ユーザ定義受付の保存」を参照してください。

(f)　SOAP 受付の検証

定義した SOAP 受付とビジネスプロセス定義との整合性を検証できます。

SOAP 受付の検証方法の詳細については，「2.14　ユーザ定義受付の検証」を参照してください。

(g)　SOAP 受付の情報の変更

定義した SOAP 受付は，必要に応じて受付名，受付 ID，およびコンテキストルートを変更できます。

SOAP 受付の情報を変更する方法については，「2.15　ユーザ定義受付の情報の変更」を参照してください。

(h)　SOAP 受付の削除

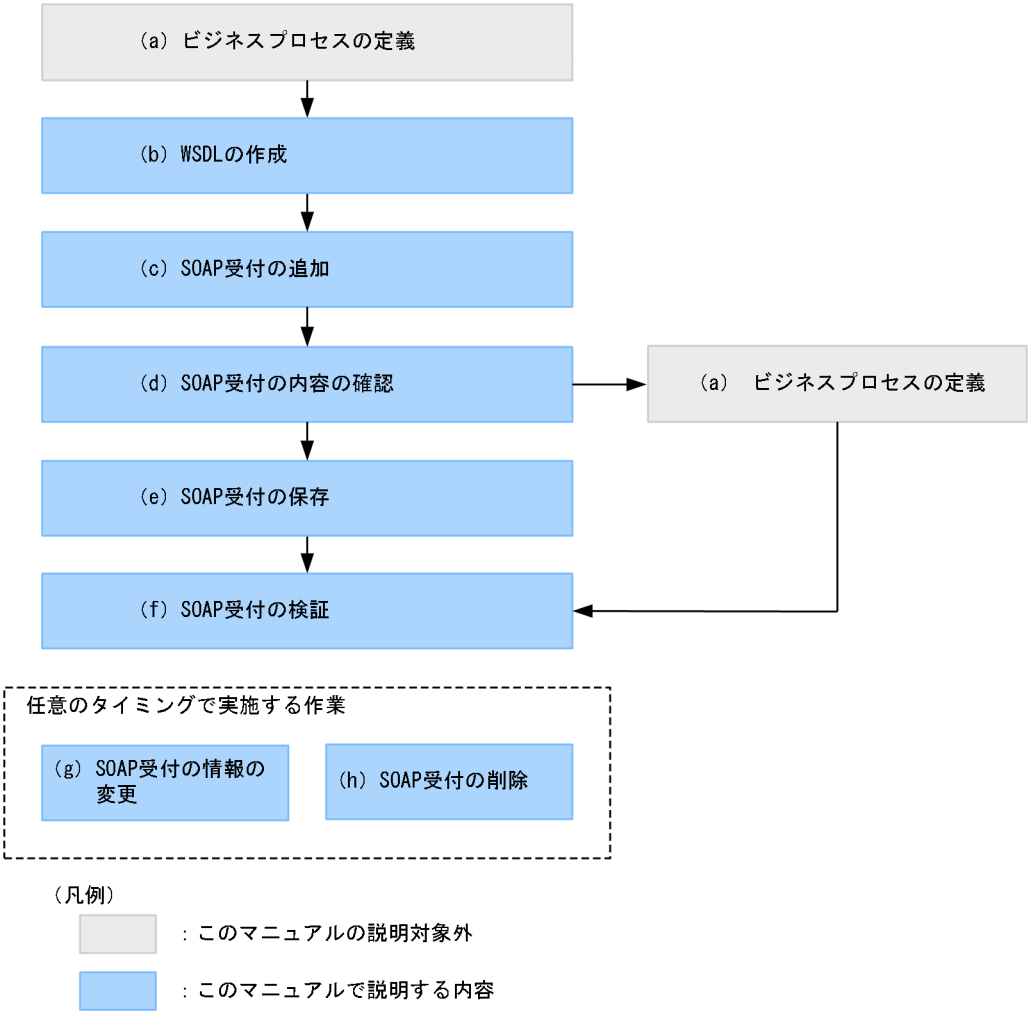
不要になった SOAP 受付は，必要に応じて削除できます。

SOAP 受付の削除方法については，「2.16　ユーザ定義受付の削除」を参照してください。

(2)　ビジネスプロセスの定義後に SOAP 受付を定義する

定義済みのビジネスプロセスのインターフェースに合うように SOAP 受付を定義する流れについて，次の図に示します。

図 2-3　SOAP 受付の定義の流れ（ビジネスプロセスの定義後に定義する場合）



ビジネスプロセスの定義後に SOAP 受付を定義する場合の作業について次に説明します。

(a)　ビジネスプロセスの定義

SOAP 受付を定義する前に，ビジネスプロセスを定義しておく必要があります。ビジネスプロセスが定義途中の場合でも，サービスリクエスタとのインターフェースとなる受付アクティビティおよび応答アクティビティを配置し，オペレーション名および割当変数の設定まで行ってください。

ビジネスプロセスの定義については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.3　ビジネスプロセスの内容の定義」を参照してください。

(b)　WSDL の作成

SOAP 受付を定義する際に利用する WSDL を作成します。

WSDL の作成の詳細については，「2.11　WSDL の作成」を参照してください。

(c)　SOAP 受付の追加

ユーザ定義受付追加ウィザードを利用して，SOAP 受付を追加します。

SOAP 受付を新規に追加する方法については，「2.2.2　SOAP 受付の追加」を参照してください。

(d)　SOAP 受付の内容の確認

追加された SOAP 受付の内容を確認します。SOAP 受付の内容は，ビジネスプロセスの定義内容と合致している必要があります。

SOAP 受付の確認の詳細については，「2.12　ユーザ定義受付の内容の確認」を参照してください。

(e)　SOAP 受付の保存

定義した SOAP 受付をリポジトリに保存します。

SOAP 受付を保存する方法については，「2.13　ユーザ定義受付の保存」を参照してください。

(f)　SOAP 受付の検証

定義した SOAP 受付とビジネスプロセス定義との整合性を検証できます。

SOAP 受付の検証方法の詳細については，「2.14　ユーザ定義受付の検証」を参照してください。

(g)　SOAP 受付の情報の変更

定義した SOAP 受付は，必要に応じて受付名，受付 ID，およびコンテキストルートを変更できます。

SOAP 受付の情報を変更する方法については，「2.15　ユーザ定義受付の情報の変更」を参照してください。

(h)　SOAP 受付の削除

不要になった SOAP 受付は，必要に応じて削除できます。

SOAP 受付の削除方法については，「2.16　ユーザ定義受付の削除」を参照してください。

2.2.2　SOAP 受付の追加

SOAP 受付を追加します。SOAP 受付を新規に追加するには，ユーザ定義受付追加ウィザードを利用します。ユーザ定義受付追加ウィザードでは，追加する SOAP 受付でどの WSDL ファイルを使うか，WSDL 内のどの wsdl:port 要素を利用するかを定義します。

ポイント

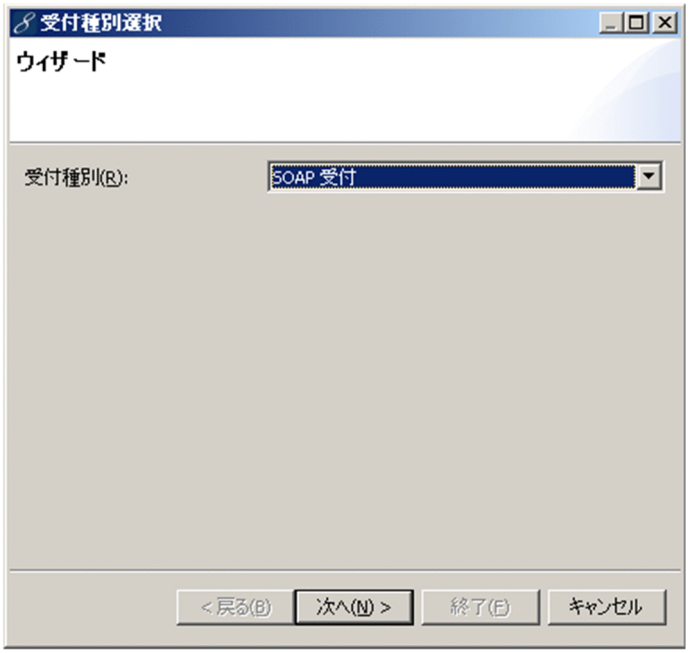
追加する SOAP 受付の定義内容は，ビジネスプロセスの受付アクティビティおよび応答アクティビティの定義に合わせる必要があります。次のどちらかの順で，SOAP 受付とビジネスプロセスの定義内容を合わせてください。

- SOAP 受付を追加する前に，ビジネスプロセスの受付アクティビティおよび応答アクティビティを定義して，受付アクティビティおよび応答アクティビティのオペレーションや電文の定義に合わせて SOAP 受付を追加します。
- SOAP 受付を追加したあとに，SOAP 受付の内容に合わせてビジネスプロセスの受付アクティビティおよび応答アクティビティの定義を変更します。

SOAP 受付を新規に追加する手順を次に示します。

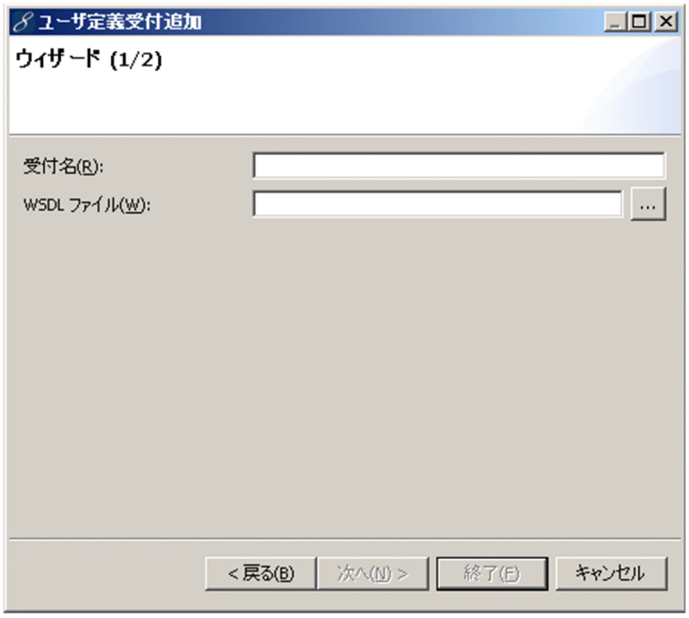
1. ツリービューのサービス定義一覧で，SOAP 受付を追加するサービス（ビジネスプロセス）を選択して，右クリックします。
   サービス一覧のポップアップメニューが表示されます。

2. ポップアップメニューから［ユーザ定義受付（呼出先固定）追加］を選択します。
   受付種別選択ウィザードが表示されます。



3. ［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。

4. ［次へ］ボタンをクリックします。
   ユーザ定義受付追加ウィザード（1/2）が表示されます。



5. [受付名] と [WSDL ファイル] を指定します。

- 受付名

  SOAP 受付の名称を XML スキーマの NCName 定義文字で指定します。1～40 バイトで指定します。受付名は，ビジネスプロセス内で一意である必要があります。

- WSDL ファイル

  SOAP 受付で利用する WSDL ファイルを絶対パスで指定します。相対パス（例：wsdldir¥wsdlfile.wsdl）や UNC 形式（例：¥¥mypc¥wsdldir¥wsdlfile.wsdl）で指定しないでください。

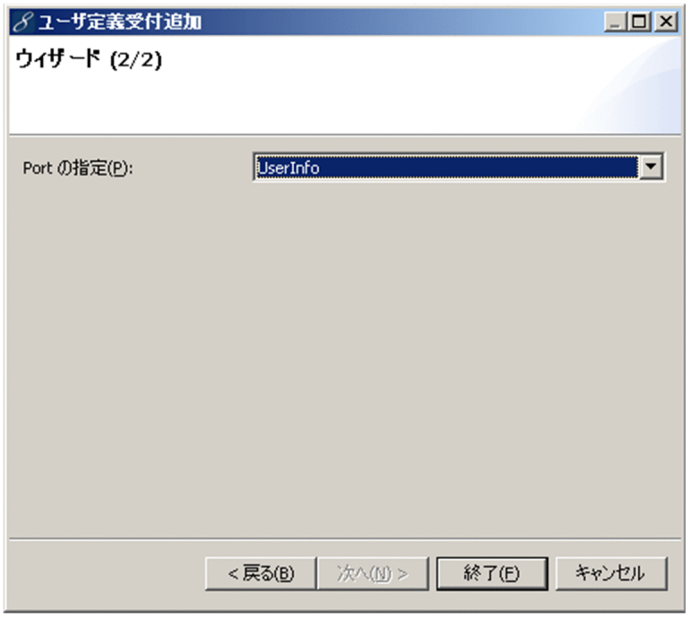
  [...] ボタンをクリックして表示されるダイアログから WSDL ファイルを選択することもできます。

6. [次へ] ボタンをクリックします。

ユーザ定義受付追加ウィザード（2/2）が表示されます。

[次へ] ボタンをクリックしたときに，「WSDL ファイルの解析に失敗しました。」というメッセージが表示された場合，WSDL ファイルにエラーがあります。

Web サービスを利用する場合の WSDL の形式については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.1　Web サービスを利用したサービス部品の適用範囲」を参照してください。ユーザ定義受付を利用する場合の WSDL の形式については，このマニュアルの「2.11.3 SOAP 受付の WSDL 作成時の注意事項」を参照してください。



7. [Port の指定] のドロップダウンリストから，利用するポートを選択します。

   ドロップダウンリストには，手順 5.で指定した WSDL ファイルに定義されているポート名（wsdl:port 要素の name 属性の値）が表示されます。

8. [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。

   ユーザ定義受付定義画面の詳細については，マニュアル「サービスプラットフォーム リファレンス」の「1.2.5　ユーザ定義受付定義画面」を参照してください。

注意事項

警告メッセージが表示された場合は，マニュアル「アプリケーションサーバ メッセージ(構築／運用／開発用)」の次の個所を参照して，表示されたメッセージの内容を確認してください。

- 「KDCCC」で始まる警告メッセージが表示された場合
  「5.2　KDCCC0001 から KDCCC9999 までのメッセージ」
- 「KDJW」で始まる警告メッセージが表示された場合
  「10.　KDJW（Component Container が出力するメッセージ)」

2.2.3　SOAP 受付の定義を編集する

SOAP1.1/1.2 併用モードの場合，ユーザ定義受付（詳細）定義画面で SOAP 受付定義ファイルの編集ができます。SOAP 受付定義ファイルについては，マニュアル「サービスプラットフォーム リファレンス」の「3.6.1　SOAP 受付定義ファイル」を参照してください。

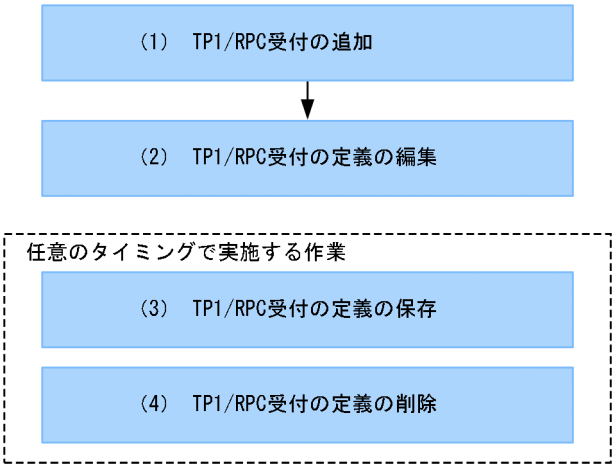
2.3　TP1/RPC 受付の定義

この節では，TP1/RPC 受付の追加，および定義の編集について説明します。

2.3.1　TP1/RPC 受付の定義の流れ

TP1/RPC 受付の定義の流れを次に示します。

図 2-4　TP1/RPC 受付の定義の流れ



TP1/RPC 受付を定義する場合の作業について次に説明します。

(1)　TP1/RPC 受付の追加

TP1/RPC 受付の EAR ファイルをインポートします。

TP1/RPC 受付を新規に追加する方法については，「2.3.2　TP1/RPC 受付を追加する」を参照してください。

(2)　TP1/RPC 受付の定義の編集

インポート済みの TP1/RPC 受付の内容を変更できます。

TP1/RPC 受付の定義を変更する方法については，「2.3.3　TP1/RPC 受付の定義を編集する」を参照してください。

(3)　TP1/RPC 受付の定義の保存

定義した TP1/RPC 受付をリポジトリに保存します。

TP1/RPC 受付を保存する方法については，「2.13　ユーザ定義受付の保存」を参照してください。

(4)　TP1/RPC 受付の定義の削除

不要になった TP1/RPC 受付は，必要に応じて削除できます。

TP1/RPC 受付の削除方法については，「2.16　ユーザ定義受付の削除」を参照してください。

2.3.2　TP1/RPC 受付を追加する

TP1/RPC 受付を新規に追加する手順を次に示します。

1. ツリービューのサービス定義一覧で，TP1/RPC 受付を追加するサービス（ビジネスプロセス）を選択して，右クリックします。
   サービス一覧のポップアップメニューが表示されます。

2. ポップアップメニューから［ユーザ定義受付（呼出先固定）追加］を選択します。
   受付種別選択ウィザードが表示されます。

3. ［受付種別］のドロップダウンリストから，［TP1/RPC 受付］を選択します。
   ［次へ］ボタンをクリックすると，TP1/RPC 受付の追加に必要な情報を入力するダイアログが表示されます。

4. ［受付名］を指定します。
   TP1/RPC 受付の名称を 1～40 バイトで指定します。

5. ［終了］ボタンをクリックします。
   TP1/RPC 受付がビジネスプロセスに追加され，ユーザ定義受付定義画面が表示されます。
   ユーザ定義受付定義画面の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「1.2.5 ユーザ定義受付定義画面」を参照してください。

6. TP1/RPC 受付の定義情報を設定します。
   TP1/RPC 受付の設定項目について次の表に示します。

表 2-1　TP1/RPC 受付のユーザ定義受付定義画面（基本）の設定項目

分類	項目	設定内容	設定
ユーザ定義受付情報	受付名	設定した受付名が表示されます。	◎
	受付 ID	受付 ID が表示されます。必要に応じて変更します。	◎
	デフォルトオペレーション名	デフォルトで呼び出すオペレーションを選択します。	◎
	オペレーション	任意のオペレーションを追加します。	◎
オペレーション情報	通信モデル	「同期」を設定します。	◎
要求電文	［any 型を使う］チェックボックス	チェックしません。	－

分類	項目		設定内容	設定
要求電文	受付	電文フォーマット	サービスリクエスタから TP1/RPC 受付が受け取る要求電文のバイナリフォーマット定義ファイル（拡張子：.fdx）を指定します。 バイナリ形式の電文フォーマットの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.4　電文フォーマット（バイナリフォーマット定義ファイル）の作成方法」を参照してください。	◎
	サービス部品	［使う］チェックボックス	データ変換を行う場合は，チェックします。	○
		電文フォーマット	［使う］チェックボックスをチェックした場合は，指定します。 TP1/RPC 受付からサービス部品に渡すサービス電文のフォーマット定義ファイルを指定します。 バイナリフォーマットから XML フォーマットに変換する場合は，要求電文のバイナリフォーマット定義ファイルから cscfdx2xsd コマンドを使用して生成した XML フォーマット定義ファイルを指定できます。	○
	データ変換定義		［使う］チェックボックスをチェックした場合は，指定します。 受付からサービス部品への要求電文のデータ変換定義を設定します。	○
応答電文	［any 型を使う］チェックボックス		チェックしません。	－
	受付	電文フォーマット	TP1/RPC 受付からサービスリクエスタに渡す応答電文のバイナリフォーマット定義ファイル（拡張子：.fdx）を指定します。 バイナリ形式の電文フォーマットの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.4　電文フォーマット（バイナリフォーマット定義ファイル）の作成方法」を参照してください。	◎
	サービス部品	［使う］チェックボックス	データ変換を行う場合は，チェックします。	○
		電文フォーマット	［使う］チェックボックスをチェックした場合は，指定します。 サービス部品から TP1/RPC 受付が受け取るサービス電文のフォーマット定義ファイルを指定します。 バイナリフォーマットから XML フォーマットに変換する場合は，応答電文のバイナリフォーマット定義ファイルから cscfdx2xsd コマンドを使用して生成した XML フォーマット定義ファイルを指定できます。	○
	データ変換定義		［使う］チェックボックスをチェックした場合は，指定します。 サービス部品から受付への応答電文のデータ変換定義を設定します。	○

（凡例）

◎：必ず設定します。

○：任意で設定します。

－：該当しません。

7. ユーザ定義受付（詳細）タブをクリックします。

ユーザ定義受付定義画面（詳細）が表示されます。

8. TP1/RPC 受付の定義情報を設定します。

TP1/RPC 受付の設定項目について次の表に示します。

表 2-2　TP1/RPC 受付のユーザ定義受付定義画面（詳細）の設定項目

分類	項目	設定内容	設定
ユーザ定義受付制御情報	独自定義ファイル	次の独自定義ファイルが表示されます。必要に応じて変更します。 • cscurecptp1rpc.properties※1	◎
	EAR ファイル	次の EAR ファイルが設定されていることを確認します。 • cscmsg_urecp_tp1rpc.ear※2	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※1

設定されている cscurecptp1rpc.properties はテンプレートファイルです。cscurecptp1rpc.properties を選択して［編集］ボタンをクリックし，独自定義ファイルの内容を修正してください。プロパティファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.4.1　TP1/RPC 受付定義ファイル」を参照してください。

注※2

ファイルの格納先ディレクトリは「<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¥lib」です。

2.3.3　TP1/RPC 受付の定義を編集する

ユーザ定義受付定義画面で，受付名，受付 ID，オペレーション名などインポート済みの TP1/RPC 受付の定義を編集します。TP1/RPC 受付の定義を編集する場合は，カスタム受付制御情報の独自定義ファイルに，TP1/RPC 受付定義ファイル（cscurecptp1rpc.properties）を必ず指定してください。

TP1/RPC 受付の定義の情報を変更する手順については，「2.15　ユーザ定義受付の情報の変更」を参照してください。

注意事項

TP1/RPC 受付では，要求電文および応答電文にバイナリ形式（.fdx）の電文フォーマットを指定する必要があります。バイナリ形式の電文フォーマットの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.4　電文フォーマット（バイナリフォーマット定義ファイル）の作成方法」を参照してください。

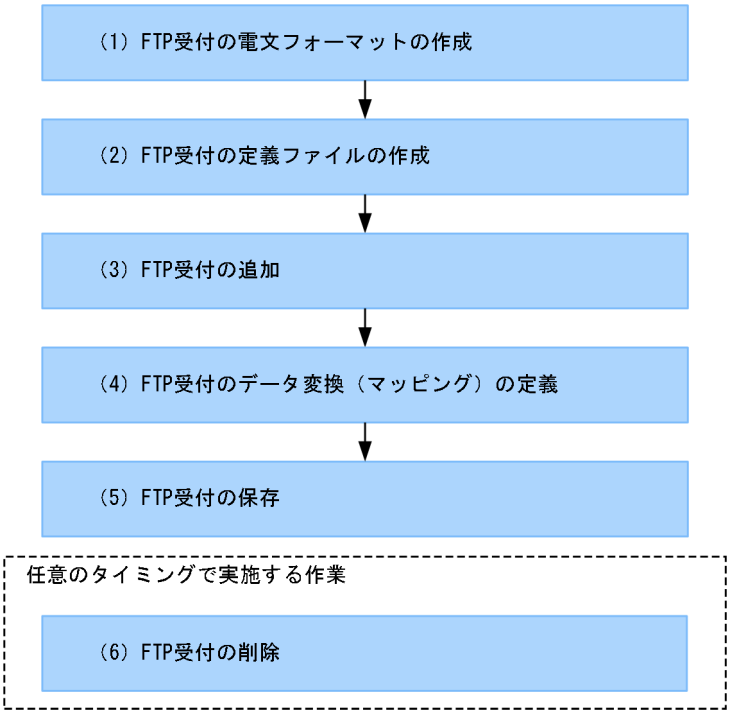
2.4　FTP 受付の定義

この節では，FTP 受付の定義ファイルの作成，FTP 受付の追加など FTP 受付の定義について説明します。

2.4.1　FTP 受付の定義の流れ

FTP 受付の定義の流れを次に示します。

図 2-5　FTP 受付の設定の流れ



FTP 受付を定義する場合の作業について次に説明します。

(1)　FTP 受付の電文フォーマットの作成

FTP 受付で使用する電文のフォーマットを作成します。

FTP 受付の電文フォーマットの作成については，「2.4.2　FTP 受付の電文フォーマットを作成する」を参照してください。

(2)　FTP 受付の定義ファイルの作成

FTP 受付で使用する定義ファイルを作成し，指定された場所に格納します。

FTP 受付の定義ファイルの作成については，「2.4.3　FTP 受付の定義ファイルを作成する」を参照してください。

(3)　FTP 受付の追加

ウィザードおよびサービスアダプタ定義画面を利用して FTP 受付を追加します。

FTP 受付を追加する手順については，「2.4.4　FTP 受付を追加する」を参照してください。

(4)　FTP 受付のデータ変換（マッピング）の定義

データ変換定義画面を利用して，変換元の電文フォーマット定義ファイルおよび変換先の電文フォーマット定義ファイルのデータ変換を定義します。

データ変換の定義については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7. データ変換の定義」を参照してください。

データ変換定義画面については，マニュアル「サービスプラットフォーム リファレンス」の「1.2.4　データ変換定義画面」を参照してください。

(5)　FTP 受付の保存

FTP 受付の定義情報は，必要に応じてリポジトリに保存する必要があります。

FTP 受付の保存については，「2.13　ユーザ定義受付の保存」を参照してください。

(6)　FTP 受付の削除

不要になった FTP 受付は，必要に応じて削除できます。

FTP 受付の削除については，「2.16　ユーザ定義受付の削除」を参照してください。

2.4.2　FTP 受付の電文フォーマットを作成する

FTP 受付の電文フォーマットの定義ファイルはサービスプラットフォームが提供するスキーマを使用します。そのため，電文フォーマットの定義ファイルの作成は不要です。

FTP 受付で使用する電文フォーマットの種類，および内容について説明します。

ファイルの格納場所は「<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¥schema¥ftprecp」です。

(1)　FTP 受付の要求電文フォーマット

FTP 受付の要求電文フォーマットをオペレーションごとに示します。

(a) PUT オペレーション

PUT オペレーションの FTP 受付からビジネスプロセスに渡す要求電文フォーマットを次に示します。この電文フォーマットのファイル名は「urecp_ftp_put_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/ftp/put_request」です。

表 2-3 FTP 受付の要求電文フォーマット（PUT オペレーション）

タグ名	出現回数※	説明
<request>	1 回	−
<ftpclient-ipaddr>	1 回	FTP クライアントの IP アドレスです。 IP アドレスのテキスト表現形式で設定されます。
<ftp-user>	1 回	FTP クライアントが USER コマンドで指定したユーザ名です。 USER コマンドの引数に指定された情報がそのまま設定されます。
<ftp-type>	1 回	FTP クライアントが TYPE コマンドで指定したタイプの情報です。 TYPE コマンドの引数に指定された情報がそのまま設定されます。
<transfer-type>	1 回	転送種別です。次のどちらかが設定されます。 • APPE ファイルへの情報追加 • STOR ファイルの上書き
<transfer-path>	1 回	転送対象ファイルパスです。 FTP クライアントが APPE コマンド，または STOR コマンドで指定したパス情報から，先頭の振分定義部分を取り除いたファイルパスの情報が設定されます。 先頭には必ず「/」が設定されます。
<file-name>	1 回	FTP クライアントから送られてきたファイルのデータが書き出された中間ファイルのファイル名です。
<file-size>	1 回	FTP クライアントから送られてきて，中間ファイルに書き出されたデータのサイズです。
<request-id>	1 回	リクエスト ID です。FTP 受付が生成したリクエスト ID が設定されます。
<ftp-commands-before>	0 または 1 回	転送前に実行する FTP コマンドです。 FTP クライアントから次に示す FTP コマンドが要求されている場合に設定されます。 SITE CSCTHR -B <FTP コマンド>[;<FTP コマンド> …] この場合，「<FTP コマンド>[;<FTP コマンド> …]」の部分がそのままの形式で設定されます。 ただし，SITE CSCTHR コマンドの引数で指定された FTP コマンドが，FTP コマンド許可リスト定義機能で許可されていない場合は設定されません。 FTP クライアントから CSCTHR コマンドが複数要求されている場合は，「<FTP コマンド>[;<FTP コマンド> …]」の部分が SITE CSCTHR コマンドを実行した順番にセミコロン（;）で接続されて追加されます。 CSCTHR コマンドが一度も要求されていない場合はこのタグは出力されません。

タグ名		出現回数※	説明
	<ftp-commands-after>	0 または 1 回	転送後に実行する必要がある FTP コマンドです。 FTP クライアントから次に示すコマンドが要求されている場合に設定されます。 SITE CSCTHR -A <FTP コマンド>[;<FTP コマンド> …] この場合，「<FTP コマンド>[;<FTP コマンド> …]」の部分がそのままの形式で設定されます。 ただし，SITE CSCTHR コマンドの引数で指定された FTP コマンドが，FTP コマンド許可リスト定義機能で許可されていない場合は設定されません。 FTP クライアントから CSCTHR コマンドが複数要求されている場合は，「<FTP コマンド>[;<FTP コマンド> …]」の部分が SITE CSCTHR コマンドを実行した順番にセミコロン（;）で接続されて追加されます。 CSCTHR コマンドが一度も要求されていない場合はこのタグは出力されません。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

(b) GET オペレーション

GET オペレーションの FTP 受付からビジネスプロセスに渡す要求電文フォーマットを次に示します。この電文フォーマットのファイル名は「urecp_ftp_get_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/ftp/get_request」です。

表 2-4 FTP 受付の要求電文フォーマット（GET オペレーション）

タグ名		出現回数※	説明
<request>		1 回	－
	<ftpclient-ipaddr>	1 回	FTP クライアントの IP アドレスです。 IP アドレスのテキスト表現形式で設定されます。
	<ftp-user>	1 回	FTP クライアントが USER コマンドで指定したユーザ名です。 USER コマンドの引数に指定された情報がそのまま設定されます。
	<ftp-type>	1 回	FTP クライアントが TYPE コマンドで指定したタイプの情報です。 TYPE コマンドの引数に指定された情報がそのまま設定されます。
	<transfer-type>	1 回	転送種別です。次の内容が設定されます。 • RETR ファイルの取得
	<transfer-path>	1 回	転送対象ファイルパスです。 FTP クライアントが RETR コマンドで指定したパス情報から，先頭の振分定義部分を取り除いたファイルパスの情報が設定されます。 先頭には必ず「/」が設定されます。

タグ名		出現回数※	説明
	<request-id>	1 回	リクエスト ID です。FTP 受付が生成したリクエスト ID が設定されます。
	<ftp-commands-before>	0 または 1 回	転送前に実行する必要がある FTP コマンドです。 FTP クライアントから次に示す FTP コマンドが要求されている場合に設定されます。 SITE CSCTHR -B <FTP コマンド>[;<FTP コマンド> …] この場合,「<FTP コマンド>[;<FTP コマンド> …]」の部分がそのままの形式で設定されます。 ただし,SITE CSCTHR コマンドの引数で指定された FTP コマンドが,FTP コマンド許可リスト定義機能で許可されていない場合は設定されません。 FTP クライアントから CSCTHR コマンドが複数要求されている場合は,「<FTP コマンド>[;<FTP コマンド> …]」の部分が SITE CSCTHR コマンドを実行した順番にセミコロン（;）で接続されて追加されます。 CSCTHR コマンドが一度も要求されていない場合はこのタグは出力されません。
	<ftp-commands-after>	0 または 1 回	転送後に実行する必要がある FTP コマンドです。 FTP クライアントから次に示すコマンドが要求されている場合に設定されます。 SITE CSCTHR -A <FTP コマンド>[;<FTP コマンド> …] この場合,「<FTP コマンド>[;<FTP コマンド> …]」の部分がそのままの形式で設定されます。 ただし,SITE CSCTHR コマンドの引数で指定された FTP コマンドが,FTP コマンド許可リスト定義機能で許可されていない場合は設定されません。 FTP クライアントから CSCTHR コマンドが複数要求されている場合は,「<FTP コマンド>[;<FTP コマンド> …]」の部分が SITE CSCTHR コマンドを実行した順番にセミコロン（;）で接続されて追加されます。 CSCTHR コマンドが一度も要求されていない場合はこのタグは出力されません。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

(c) GETINFO オペレーション

GETINFO オペレーションの FTP 受付からビジネスプロセスに渡す要求電文フォーマットを次に示します。この電文フォーマットのファイル名は「urecp_ftp_getinfo_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/ftp/getinfo_request」です。

表 2-5 FTP 受付の要求電文フォーマット（GETINFO オペレーション）

タグ名		出現回数※	説明
<request>		1 回	－
	<ftpclient-ipaddr>	1 回	FTP クライアントの IP アドレスです。 IP アドレスのテキスト表現形式で設定されます。

タグ名		出現回数※	説明
	<ftp-user>	1 回	FTP クライアントが USER コマンドで指定したユーザ名です。 USER コマンドの引数に指定された情報がそのまま設定されます。
	<getinfo-type>	1 回	情報取得種別として，次のどちらかが設定されます。 • LIST ファイル情報の一覧を取得します。 • NLST ファイル名の一覧を取得します。
	<getinfo-option>	0 または 1 回	情報取得オプション情報です。 次の条件をすべて満たす場合は，オプション定義ファイルに設定されている文字列と一致した文字列部分が設定されます。 • FTP クライアントが LIST コマンドまたは NLST コマンドで指定した引数の先頭部分が，オプション定義ファイルに設定されている文字列と一致する • 引数の先頭部分の後ろに文字列が続いていない，または空白文字になっている なお，次のどちらかに当てはまる場合，このタグは出力されません。 • FTP クライアントが LIST コマンドまたは NLST コマンドで引数を指定しなかった場合 • FTP クライアントが LIST コマンドまたは NLST コマンドで指定した引数の先頭部分がオプション定義ファイルに設定されている文字列と一致しなかった場合
	<getinfo-path>	0 または 1 回	情報取得パス情報です。 <getinfo-option>タグが出力されていない場合は，FTP クライアントが LIST コマンドまたは NLST コマンドで指定した引数がそのまま設定されます。 <getinfo-option>タグが出力されている場合は，FTP クライアントが LIST コマンドまたは NLST コマンドで指定した引数の先頭から，<getinfo-option>タグに出力されているオプション情報およびその後ろの空白文字を取り除いたパス情報部分が設定されます。 なお，次のどちらかに当てはまる場合，このタグは出力されません。 • FTP クライアントが LIST コマンドまたは NLST コマンドで引数を指定しなかった場合 • FTP クライアントが LIST コマンドまたは NLST コマンドで指定した引数に，オプション情報しか指定されていなかった場合
	<getinfo-path-context>	0 または 1 回	<getinfo-path>タグから抽出した情報取得パス情報の部分文字列です。 <getinfo-path>タグが出力されている場合は，先頭のスラッシュ（/）から次のスラッシュ（/）より前の部分が設定されます。このタグにスラッシュ（/）は含まれません。 <getinfo-path>タグが出力されていない場合，または<getinfo-path>タグの先頭のスラッシュ（/）から次のスラッシュ（/）より前の部分が抽出できない場合は，このタグは出力されません。
	<getinfo-path-relative>	0 または 1 回	<getinfo-path>タグから抽出した情報取得パス情報の部分文字列です。 <getinfo-path>タグが出力されている場合は，<getinfo-path>タグの 2 つ目のスラッシュ（/）より後ろの値が設定されます。 <getinfo-path>タグが出力されていない場合，または<getinfo-path>タグの 2 つ目のスラッシュ（/）より後ろの値が設定されていない場合は，このタグは出力されません。
	<request-id>	1 回	リクエスト ID です。FTP 受付が生成したリクエスト ID が設定されます。

タグ名		出現回数※	説明
	<ftp-commands-before>	0 または 1 回	リストコマンド実行前に実行する必要がある FTP コマンドです。 FTP クライアントから次に示す FTP コマンドが要求されている場合に設定されます。 SITE CSCTHR -B <FTP コマンド>[;<FTP コマンド> …] この場合，「<FTP コマンド>[;<FTP コマンド> …]」の部分がそのままの形式で設定されます。 ただし，SITE CSCTHR コマンドの引数で指定された FTP コマンドが，FTP コマンド許可リスト定義機能で許可されていない場合は設定されません。 FTP クライアントから CSCTHR コマンドが複数要求されている場合は，「<FTP コマンド>[;<FTP コマンド> …]」の部分が SITE CSCTHR コマンドを実行した順番にセミコロン（;）で接続されて追加されます。 CSCTHR コマンドが一度も要求されていない場合は，このタグは出力されません。
	<ftp-commands-after>	0 または 1 回	リストコマンド実行後に実行する必要がある FTP コマンドです。 FTP クライアントから次に示すコマンドが要求されている場合に設定されます。 SITE CSCTHR -A <FTP コマンド>[;<FTP コマンド> …] この場合，「<FTP コマンド>[;<FTP コマンド> …]」の部分がそのままの形式で設定されます。 ただし，SITE CSCTHR コマンドの引数で指定された FTP コマンドが，FTP コマンド許可リスト定義機能で許可されていない場合は設定されません。 FTP クライアントから CSCTHR コマンドが複数要求されている場合は，「<FTP コマンド>[;<FTP コマンド> …]」の部分が SITE CSCTHR コマンドを実行した順番にセミコロン（;）で接続されて追加されます。 CSCTHR コマンドが一度も要求されていない場合は，このタグは出力されません。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

(2) FTP 受付の応答電文フォーマット

FTP 受付の応答電文フォーマットをオペレーションごとに示します。

(a) PUT オペレーション

ビジネスプロセスから PUT オペレーションの FTP 受付に渡す応答電文フォーマットを次に示します。この電文フォーマットのファイル名は「urecp_ftp_put_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/ftp/put_response」です。

表 2-6 FTP 受付の応答電文フォーマット（PUT オペレーション）

タグ名	出現回数※1	説明
<response>	1 回	－

タグ名		出現回数※1	説明
	<message>	0 または 1 回	ビジネスプロセスの終了状態を示すメッセージです。指定できるメッセージの最大長は 1,024 文字です。 最大長を超える長さのメッセージを指定した場合，メッセージが 1,024 文字に収まるようメッセージの後ろの部分は切り捨てられます。 使用できない文字※2 が含まれていた場合，使用できない文字はクエスチョンマーク（?）に置き換えられます。 省略した場合，次のメッセージが設定されます。 • <success>タグの値が true の場合 Transfer complete. • <success>タグの値が false の場合 Requested action aborted.
	<success>	0 または 1 回	ビジネスプロセスの終了状態を示すフラグです。次のどちらかを指定します。 • true ビジネスプロセスの処理が正常に終了した。 • false ビジネスプロセスの処理でエラーが発生した。 省略した場合，true が設定されます。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

次に示す文字以外は使用できません。

半角英数字，半角空白，半角タブ，半角改行，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

(b) GET オペレーション

ビジネスプロセスから GET オペレーションの FTP 受付に渡す応答電文フォーマットを次に示します。この電文フォーマットのファイル名は「urecp_ftp_get_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/ftp/get_response」です。

表 2-7　FTP 受付の応答電文フォーマット（GET オペレーション）

タグ名	出現回数※1	説明
<response>	1 回	－

タグ名		出現回数※1	説明
	<message>	0 または 1 回	ビジネスプロセスの終了状態を示すメッセージです。指定できるメッセージの最大長は 1,024 文字です。 最大長を超える長さのメッセージを指定した場合，メッセージが 1,024 文字に収まるようメッセージの後ろの部分は切り捨てられます。 使用できない文字※2 が含まれていた場合，使用できない文字はクエスチョンマーク（?）に置き換えられます。 省略した場合，次のメッセージが設定されます。 • <success>タグの値が true の場合 Transfer complete. • <success>タグの値が false の場合 Requested action not taken.
	<success>	0 または 1 回	ビジネスプロセスの終了状態を示すフラグです。次のどちらかを指定します。 • true ビジネスプロセスの処理が正常に終了した。 • false ビジネスプロセスの処理でエラーが発生した。 省略した場合，true が設定されます。
	<folder-name>	0 または 1 回	FTP クライアントへ送り返す中間ファイルが格納されている共通フォルダの定義名です。 省略した場合は，作業フォルダ内のファイルが中間ファイルとして使用されます。ただし，<success>タグの値が false の場合，この設定は不要です。もし設定されていても無視されます。
	<file-name>	0 または 1 回	FTP クライアントへ送り返すファイルのデータが書き出された中間ファイルのファイル名です。 作業フォルダを使用する場合は，作業フォルダ直下のファイルを指定してください。 共通フォルダを使用する場合は，共通フォルダルート配下のファイルを相対パスで指定してください。ファイル名の先頭にスラッシュ（/）を使用した場合，スラッシュ（/）は無視されます。また，シンボリックリンク以外のファイルを指定してください。ただし，<success>タグの値が false の場合，この設定は不要です。もし設定されていても無視されます。
	<file-size>	0 または 1 回	FTP クライアントへ送り返すファイルのデータのサイズです。 ただし，<success>タグの値が false の場合，この設定は不要です。もし設定されていても無視されます。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

次に示す文字以外は使用できません。

半角英数字，半角空白，半角タブ，半角改行，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右ア

ングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

(c) GETINFO オペレーション

ビジネスプロセスから GETINFO オペレーションの FTP 受付に渡す応答電文フォーマットを次に示します。この電文フォーマットのファイル名は「urecp_ftp_getinfo_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/ftp/getinfo_response」です。

表 2-8 FTP 受付の応答電文フォーマット（GETINFO オペレーション）

タグ名		出現回数 ※1	説明
<response>		1 回	−
	<reply-code>	0 または 1 回	情報取得処理を実行したときに FTP サーバから受け取ったリプライコード（またはそれに準ずるコード）です。 FTP アダプタの GETINFO オペレーションを使用して情報取得処理を実行する場合は，FTP アダプタの応答電文の<reply-code>の値をこのタグに設定することを推奨しています。 このタグに指定できる値は，200～299 の数値，または 400～599 の数値です。 200～299 を設定した場合は，設定した値が FTP クライアントに返されるリプライコードとなります。 400～599 の値を設定した場合は，設定した値が FTP クライアントに返されるリプライコードとなります。 <success>タグの値が true で，このタグを省略した場合は，FTP 受付がリプライコードに 226 を仮定します。 <success>タグの値が false の場合は，この設定は不要です。もし設定されていても無視されます。
	<message>	0 または 1 回	情報取得処理を実行したときに FTP サーバから受け取ったリプライメッセージ，またはビジネスプロセスの終了状態を示すメッセージです。 指定できるメッセージの最大長は 1,024 文字です。最大長を超える長さのメッセージを指定した場合，メッセージが 1,024 文字に収まるようメッセージの後ろの部分は切り捨てられます。 使用できない文字※2 が含まれていた場合，使用できない文字はクエスチョンマーク（?）に置き換えられます。 省略した場合，次のメッセージが設定されます。 • <success>タグの値が true の場合 Transfer complete. • <success>タグの値が false の場合 Requested action not taken. FTP アダプタの GETINFO オペレーションを使用して情報取得処理を実行する場合は，FTP アダプタの応答電文の<reply-message>の値をこのタグに設定することを推奨しています。
	<success>	0 または 1 回	ビジネスプロセスの終了状態を示すフラグです。次のどちらかを指定します。 • true

タグ名		出現回数 ※1	説明
	<success>	0 または 1 回	ビジネスプロセスの処理が正常に終了した。 • false ビジネスプロセスの処理でエラーが発生した。 省略した場合，true が設定されます。
	<file-name>	0 または 1 回	FTP クライアントへ送り返すデータが書き出された中間ファイルのファイル名です。 このタグに指定したファイルには，次の形式で取得した情報が書き込まれる必要があります。 • 文字コード：UTF-16 • 改行コード：LF※3 または CRLF ただし，次のどちらかに当てはまる場合，この設定は不要です。もし設定されていても無視されます。 • <success>タグの値が false の場合 • <success>タグの値が true で，<reply-code>の値が 400～599 の場合

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

次に示す文字以外は使用できません。

半角英数字，半角空白，半角タブ，半角改行，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

注※3

RFC959 の規則に従い，改行コードが LF の場合は CRLF に変換してクライアントに送り返します。

2.4.3　FTP 受付の定義ファイルを作成する

FTP 受付の定義ファイルの種類，および作成方法について説明します。

(1)　定義ファイルの種類

作成する定義ファイルの種類を次に示します。

- FTP 受付定義ファイル
  FTP 受付をセットアップしたあとに設定変更できる内容を定義するファイルです。
- FTP 実行許可リスト定義ファイル

ビジネスプロセスの実行を許可するユーザや，ビジネスプロセスの実行を許可しないユーザを定義するファイルです。任意で作成します。

- FTP コマンド許可リスト定義ファイル

  転送前または転送後に実行することを許可する FTP コマンドを定義するファイルです。任意で作成します。

- リストコマンドオプション定義ファイル

  リストコマンドの引数でオプションとして指定できる文字列を定義するファイルです。任意で作成します。

- FTP 受付コンフィグファイル

  FTP 受付をセットアップしたあとに設定変更できる内容で，かつ頻繁に設定変更する可能性のある情報を定義するファイルです。任意で作成します。

(2) 定義ファイルの作成

FTP 受付で使用する定義ファイルの作成手順について説明します。リストコマンドオプション定義ファイル以外の定義ファイルは，FTP 受付が提供しているテンプレートファイルを利用して作成します。

(a) FTP 受付定義ファイル

FTP 受付の追加時に，サービスプラットフォームが提供するテンプレートファイル（cscurecpftp.properties）を編集して利用します。そのため，FTP 受付定義ファイルの作成は不要です。

(b) FTP 実行許可リスト定義ファイル

1. テンプレートファイル（<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¥config¥ftprecp¥templates¥ftp_permission_allow.properties）をコピーして，任意のディレクトリに格納します。

2. コピーしたテンプレートファイルのファイル名を任意の名称に変更します。

3. 定義内容を編集して保存します。

   FTP 実行許可リスト定義ファイルで編集できる項目については，マニュアル「サービスプラットフォーム リファレンス」の「6.8.7　FTP 実行許可リスト定義ファイル」を参照してください。

FTP 実行許可リスト定義ファイルは，FTP 受付の開始時に実行環境に反映されます。

(c) FTP コマンド許可リスト定義ファイル（FTP 受付用）

1. テンプレートファイル（<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¥config¥ftprecp¥templates¥ftp_command_allow.properties）をコピーして，任意のディレクトリに格納します。

2. コピーしたテンプレートファイルのファイル名を任意の名称に変更します。

3. 定義内容を編集して保存します。
   FTP コマンド許可リスト定義ファイルで編集できる項目については，マニュアル「サービスプラットフォーム リファレンス」の「6.8.6　FTP コマンド許可リスト定義ファイル」を参照してください。

FTP コマンド許可リスト定義ファイルは，FTP 受付の開始時に実行環境に反映されます。

(d)　リストコマンドオプション定義ファイル

1. 任意の名称でテキストファイルを作成し，任意のディレクトリに格納します。

2. 定義内容を編集して保存します。
   リストコマンドオプション定義ファイルで編集できる項目については，マニュアル「サービスプラットフォーム リファレンス」の「6.8.9　リストコマンドオプション定義ファイル」を参照してください。

リストコマンドオプション定義ファイルは，FTP 受付の開始時に実行環境に反映されます。

(e)　FTP 受付コンフィグファイル

1. テンプレートファイル（<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¥config¥ftprecp¥templates¥ftprecp_config.properties）をコピーして，次のディレクトリに格納します。
   <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>¥CSC¥config¥ftprecp

2. コピーしたテンプレートファイルのファイル名を「<FTP 受付の受付 ID>.properties」に変更します。

3. 定義内容を編集して保存します。
   FTP 受付コンフィグファイルで編集できる項目については，マニュアル「サービスプラットフォーム リファレンス」の「6.8.5　FTP 受付コンフィグファイル」を参照してください。

FTP 受付コンフィグファイルは，FTP 受付の開始時に実行環境に反映されます。

2.4.4　FTP 受付を追加する

FTP 受付を新規に追加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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのビジネスプロセスで右クリックして［ユーザ定義受付（呼出先固定）追加］を選択します。
   受付種別選択ウィザードが表示されます。

4. 受付種別のドロップダウンリストから「FTP 受付」を選択します。

［次へ］ボタンをクリックすると，FTP 受付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. ［受付名］を指定します。

FTP 受付の名称を 1～40 バイトで指定します。

6. ［終了］ボタンをクリックします。

［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。ユーザ定義受付定義画面（基本）が表示されます。

7. FTP 受付の定義情報を設定します。

FTP 受付の設定項目について次の表に示します。

表 2-9　FTP 受付のユーザ定義受付定義画面（基本）の設定項目

分類	項目		設定内容	設定
ユーザ定義受付情報	受付名		設定した受付名が表示されます。	◎
	受付 ID		受付 ID が表示されます。必要に応じて変更します。	◎
	デフォルトオペレーション名		デフォルトで呼び出すオペレーションを選択します。	◎
	オペレーション		次の中から使用するオペレーションを追加します。 • PUT：FTP クライアントから FTP サーバへファイルを転送する • GET：FTP サーバから FTP クライアントへファイルを転送する • GETINFO：FTP サーバの情報を取得し，FTP クライアントへ送付する	◎
オペレーション情報	通信モデル		「同期」を設定します。	◎
要求電文	受付	電文フォーマット	［オペレーション］の設定に応じて，次の要求電文用フォーマット定義ファイルが設定されています。 • PUT の場合：urecp_ftp_put_request.xsd※ • GET の場合：urecp_ftp_get_request.xsd※ • GETINFO の場合：urecp_ftp_getinfo_request.xsd※	×
	サービス部品	［使う］チェックボックス	必要に応じてチェックします。	○
	データ変換定義		サービス部品の電文フォーマットを指定した場合に，受付からサービス部品への要求電文のデータ変換定義を設定します。	○
応答電文	受付	電文フォーマット	［オペレーション］の設定に応じて，次の応答電文用フォーマット定義ファイルが設定されています。 • PUT の場合：urecp_ftp_put_response.xsd※ • GET の場合：urecp_ftp_get_response.xsd※ • GETINFO の場合：urecp_ftp_getinfo_response.xsd※	×

分類	項目		設定内容	設定
応答電文	サービス部品	［使う］チェックボックス	必要に応じてチェックします。	○
	データ変換定義		サービス部品の電文フォーマットを指定した場合に，サービス部品から受付への応答電文のデータ変換定義を設定します。	○

（凡例）

◎：必ず設定します。

○：任意で設定します。

×：表示されている内容を確認します。

注※

ファイルの格納先ディレクトリは「<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¥schema¥ftprecp」です。

上記以外のユーザ定義受付定義画面の設定項目については，マニュアル「サービスプラットフォーム リファレンス」の「1.2.5　ユーザ定義受付定義画面」を参照してください。

8. ユーザ定義受付（詳細）タブをクリックします。

ユーザ定義受付定義画面（詳細）が表示されます。

9. FTP 受付の定義情報を設定します。

FTP 受付の設定項目について次の表に示します。

表 2-10　FTP 受付のユーザ定義受付定義画面（詳細）の設定項目

分類	項目	設定内容	設定
ユーザ定義受付制御情報	独自定義ファイル	次の独自定義ファイルが表示されます。必要に応じて変更します。 • cscurecpftp.properties※1	○
	EAR ファイル	次の EAR ファイルが設定されていることを確認します。 • cscmsg_urecp_ftp.ear※2	×

（凡例）

○：任意で設定します。

×：表示されている内容を確認します。

注※1

設定されている cscurecpftp.properties はテンプレートファイルです。必要に応じて cscurecpftp.properties を選択して［編集］ボタンをクリックし，独自定義ファイルの内容を修正してください。プロパティファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.5.1　FTP 受付定義ファイル」を参照してください。

注※2

ファイルの格納先ディレクトリは「<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¥lib」です。

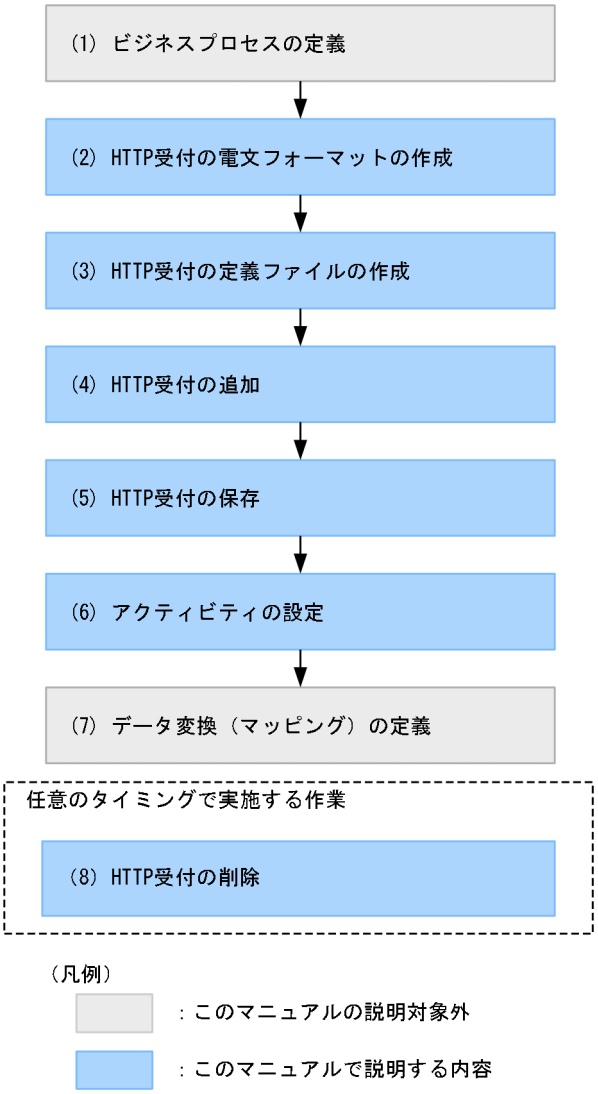
2.5　HTTP 受付の定義

この節では，HTTP 受付の電文フォーマットの作成，HTTP 受付の追加など，HTTP 受付の定義方法について説明します。実際の設定例に基づいて HTTP 受付の動作を確認したい場合は，「付録 F　HTTP 受付を利用したビジネスプロセスの設定例」を参照してください。

2.5.1　HTTP 受付の定義の流れ

HTTP 受付の定義の流れを次に示します。

図 2-6　HTTP 受付の定義の流れ



HTTP 受付を定義する場合の作業について次に説明します。

(1)　ビジネスプロセスの定義

ビジネスプロセス追加用のウィザードおよび定義済みのビジネスプロセスを利用して，ビジネスプロセスを追加します。追加したビジネスプロセスは，ビジネスプロセス定義画面で定義します。ビジネスプロセスの定義の詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.　ビジネスプロセスの定義」を参照してください。

また，ビジネスプロセスからサービス部品を呼び出す場合，サービスアダプタを定義する必要があります。サービスアダプタの定義の詳細は，「3.　サービスアダプタの定義」を参照してください。

(2)　HTTP 受付の電文フォーマットの作成

HTTP 受付で使用する要求電文フォーマットおよび応答電文フォーマットを作成します。

電文フォーマットの作成方法については，「2.5.2　HTTP 受付の電文フォーマットを作成する」を参照してください。

(3)　HTTP 受付の定義ファイルの作成

HTTP 受付で使用する定義ファイルを作成します。作成した定義ファイルはリポジトリに反映されます。

HTTP 受付で使用する定義ファイルの作成方法については，「2.5.3　HTTP 受付の定義ファイルを作成する」を参照してください。

(4)　HTTP 受付の追加

ツリービューのサービス定義一覧で HTTP 受付を追加し，内容を定義します。

特定のビジネスプロセスを呼び出すか，または呼び出しサービス選択機能で呼び出すビジネスプロセスを選択するかで，HTTP 受付を定義する手順が異なります。

特定のビジネスプロセスを呼び出す HTTP 受付の追加および定義手順については，「2.5.4　HTTP 受付を追加する」を参照してください。

呼び出すビジネスプロセスを選択する HTTP 受付の追加および定義手順については，「付録 L　ユーザ定義受付（呼出先選択）の定義」を参照してください。

(5)　HTTP 受付の保存

追加および定義済みの HTTP 受付をリポジトリに保存します。HTTP 受付の保存手順については，「2.13 ユーザ定義受付の保存」を参照してください。

(6)　アクティビティの設定

HTTP 受付の要求電文／応答電文フォーマット用のボディ変数とヘッダ変数をビジネスプロセスのアクティビティに設定します。

アクティビティの設定方法については，「2.5.5　アクティビティを設定する」を参照してください。

(7)　データ変換（マッピング）の定義

電文のやり取りでデータ変換が必要な場合は，データ変換を定義します。データ変換の定義方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.　データ変換の定義」を参照してください。

HTTP 受付を使用する場合のマッピングの例は，「付録 F　HTTP 受付を利用したビジネスプロセスの設定例」を参照してください。

(8)　HTTP 受付の削除

不要になった HTTP 受付は，必要に応じて削除できます。

HTTP 受付の削除については，「2.16　ユーザ定義受付の削除」を参照してください。

2.5.2　HTTP 受付の電文フォーマットを作成する

ここでは，HTTP 受付で使用する電文フォーマットの種類，形式，および作成方法について説明します。

(1)　電文フォーマットの種類

HTTP 受付で使用する電文フォーマットの種類を次の表に示します。電文フォーマットは，サービスプラットフォームが提供する XML スキーマを利用して作成できます。

表 2-11　電文フォーマットの種類

大分類	小分類	説明
要求電文フォーマット	ヘッダ変数用（HTTP 受付用）	HTTP リクエストのヘッダ情報および URL 情報を設定します。
	ボディ変数用（HTTP 受付用）（メイン）	HTTP リクエストのボディ情報を設定します。
	ボディ変数用（HTTP 受付用）（詳細）	要求電文フォーマット（ボディ変数（メイン））の一部として参照されます。
	ボディ変数用（JSON-XML 変換用）	JSON 変換ツールで作成した情報を設定した HTTP リクエストボディを送信したい場合に作成します。 サービスプラットフォームでは XML スキーマファイルを提供していません。
応答電文フォーマット	ヘッダ変数用（HTTP 受付用）	HTTP レスポンスのヘッダ情報を設定します。
	ヘッダ変数用（JSON-XML 変換用）	JSON-XML 変換の定義を設定します。

大分類	小分類	説明
応答電文フォーマット	ボディ変数用（HTTP 受付用）	HTTP レスポンスのボディ情報を設定します。
	ボディ変数用（JSON-XML 変換用）	JSON 変換ツールで作成した情報を設定した HTTP レスポンスボディを送信したい場合に作成します。 サービスプラットフォームでは XML スキーマファイルを提供していません。

ビジネスプロセスで検証アクティビティを使用する場合は，上記とは別の電文フォーマットを作成して使用してください。

(2) 電文フォーマットの形式

ここでは，HTTP 受付で使用する要求電文フォーマットおよび応答電文フォーマットの形式について説明します。

(a) 要求電文フォーマット（ヘッダ変数用）（HTTP 受付用）

HTTP 受付がビジネスプロセスから呼び出されたときに，HTTP 受付に渡されるヘッダ変数の要求電文フォーマットを次に示します。この電文フォーマットのファイル名は「urecp_http_header_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/request」です。

表 2-12 要求電文フォーマット（ヘッダ変数用）（HTTP 受付用）

タグ名	種別	出現回数	説明
<http-header-request>	−	1 回	−
<auth>	−	0 または 1 回	HTTP ヘッダ認証が指定されている場合，このタグが作成されます。
<username>	string	1 回	認証されたユーザのユーザ名が設定されます。ユーザ名情報はプレーンテキストにデコードされて，設定されます。
<method>	string	1 回	HTTP 受付で使用される HTTP メソッド（GET，POST，HEAD，PUT，DELETE，または OPTIONS）が設定されます。
<url>	−	1 回	リクエスト URL からロケーション情報やクエリ文字列が設定されます。
<location>	string	1 回	ロケーション情報が設定されます。
<query>	string	0 または 1 回	クエリ文字列が設定されます。
<content-type>	string	0 または 1 回	Content-Type ヘッダに指定されたメディアタイプが設定されます。

タグ名	種別	出現回数	説明
<content-type>	string	0 または 1 回	content-type タグの charset 属性には Content-Type ヘッダに指定された charset 属性（文字コード）が設定されます。charset 属性が付与されていなければ，この属性は設定されません。
<request-id>	string	0 または 1 回	HTTP 受付が生成したリクエスト ID が設定されます。HTTP 受付が作業フォルダを作成する場合，生成したリクエスト ID が設定されます。 作業フォルダ作成の詳細については，マニュアル「サービスプラットフォーム 解説」の「2.15.6 HTTP 受付のファイル管理」を参照してください。
<files>	−	0 または 1 回	HTTP クライアントから受信したマルチパート型データを基に，作業フォルダ内に生成された中間ファイルのリストが設定されます。
<file>	−	0 回以上	HTTP クライアントから受信したマルチパート型データを基に，作業フォルダ内に生成された中間ファイルが設定されます。
<partID>	string	0 または 1 回	各パートを区別するための識別子が設定されます。 HTTP 受付が受信したマルチパート型データ内で，各パートの Content-Disposition ヘッダの name 属性に対応します。
<file-name>	string	0 または 1 回	HTTP クライアントから受信したファイルデータのファイル名称が設定されます。 HTTP 受付が受信したマルチパート型データ内で，各パートの Content-Disposition ヘッダの filename 属性に対応します。なお，パス指定の場合はファイル名だけが設定されます。
<local-file-name>	string	1 回	作業フォルダ内に生成された中間ファイルの名称が設定されます。 中間ファイルの名称は作業フォルダ内で一意となります。
<content-type>	string	0 または 1 回	HTTP 受付が受信したファイルデータのメディアタイプが設定されます。 HTTP 受付が受信したマルチパート型データ内で，各パートの Content-Type ヘッダの値に対応します。 各パートの Content-Type ヘッダが省略された場合，このタグは生成されません。 content-type タグの charset 属性には，各パートの Content-Type ヘッダに指定された charset 属性（文字コード）が設定されます。charset 属性が付与されていなければ，この属性は設定されません。

タグ名		種別	出現回数	説明
	<http-header>	any	0 または 1 回	HTTP 受付が受信した HTTP ヘッダ情報がヘッダ変数に変換されて設定されます。詳細はマニュアル「サービスプラットフォーム 解説」の「2.15.4 HTTP 受付のリクエスト処理」を参照してください。
	<url-parameters>	−	0 または 1 回	URL パラメタ（HTTP 受付ルーティング定義ファイルの URL パターン内で動的セグメントに対応するリクエスト URL のセグメント）が設定されます。 http-header-urlparametersType は，HTTP 受付ルーティング定義ファイルで指定する URL パターンに応じてユーザが作成する urecp_http_header_urlparameters.xsd で定義します。

（凡例）

−：該当する項目はありません。

(b) 要求電文フォーマット（ボディ変数用）（HTTP 受付用）

HTTP 受付がビジネスプロセスから呼び出されたときに，HTTP 受付に渡されるボディ変数の要求電文フォーマットを次に示します。この電文フォーマットのファイル名は「urecp_http_body_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/request」です。

表 2-13 要求電文フォーマット（ボディ変数用）（HTTP 受付用）

タグ名		種別	出現回数	説明
<http-body-request>		−	1 回	−
	<parameter>	http-body-requestType	0 回以上	リクエストボディの情報がボディ変数に変換されて設定されます。詳細はマニュアル「サービスプラットフォーム 解説」の「2.15.4 HTTP 受付のリクエスト処理」を参照してください。

（凡例）

−：該当する項目はありません。

(c) 応答電文フォーマット（ヘッダ変数用）（HTTP 受付用）

HTTP 受付がビジネスプロセスを呼び出したときに渡すヘッダ変数の応答電文フォーマットを次に示します。この電文フォーマットのファイル名は「urecp_http_header_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/response」です。

表 2-14　応答電文フォーマット（ヘッダ変数用）（HTTP 受付用）

タグ名		種別	出現回数	説明
<http-header-response>		−	1 回	−
	<status-code>	string	0 または 1 回	HTTP レスポンスのステータスコードを指定します。
	<content-type>	string	0 または 1 回	HTTP 受付の応答電文（ボディ変数用）で表すデータの Content-Type ヘッダを指定します。空文字列が指定された場合は，この要素の指定を省略したとみなされます。 データ種別がマルチパート型となる場合，パートに変換した応答電文（ボディ変数用）に対して付与する Content-Type ヘッダとして使用します。 content-type 要素の charset 属性には，Content-Type ヘッダの charset 属性として使用する文字コードを指定します。空文字列が指定された場合は，この要素の指定を省略したとみなされます。content-type 要素の値に charset 属性やそのほかの属性を指定した場合，動作は保証されません。
	<body-partID>	string	0 または 1 回	マルチパート型データを HTTP クライアントに送信する場合に，パートに変換した応答電文（ボディ変数用）に対して付与する Content-Disposition ヘッダの name 属性の値を指定します。 body-partID 要素の値を省略してマルチパート型データに応答電文（ボディ変数用）を格納した場合，この name 属性の値には「csc-body-text」が設定されます。 データ種別がマルチパート型でない場合，この値は無視されます。
	<ignore-bodymsg>	boolean	0 または 1 回	応答電文（ボディ）を無視して HTTP クライアントに HTTP レスポンスを返すかどうかを指定します。 この要素の値に true を指定した場合は，HTTP クライアントに HTTP レスポンスを返す処理で，応答電文（ボディ）が無視されます。false を指定した場合は応答電文（ボディ）が利用されます。 この要素を指定しなかった場合，またはこの要素に無効な文字列を指定した場合は，HTTP 受付定義ファイルの httprecp.response.ignore-bodymsg プロパティで指定した値が適用されます。 応答電文（ヘッダ）の ignore-bodymsg 要素と HTTP 受付定義ファイルの httprecp.response.ignore-bodymsg プロパティの両方に値を指定した場合は応答電文（ヘッダ）の指定が優先されます。
	<files>	string	0 または 1 回	作業フォルダから読み込んで HTTP クライアントに送信する中間ファイルのリストを指定します。

タグ名				種別	出現回数	説明
		<file>		−	0 回以上	作業フォルダから読み込んで，HTTP クライアントに送信する中間ファイルを指定します。
			<partID>	string	0 または 1 回	データ種別がマルチパート型の場合に各パートを識別するための識別子を指定します。 HTTP レスポンスでのマルチパート型データの各パート単位で，Content-Disposition ヘッダの name 属性の値に対して付与する文字列です。 このタグを省略した場合は local-file-name タグの値が設定されます。 ignore-bodymsg 要素の値が「true」または HTTP 受付定義ファイルの httprecp.response.ignore-bodymsg プロパティが「true」（応答電文（ボディ）を無視する設定）で，かつ HTTP クライアントに送信する中間ファイルが 1 つの場合は，マルチパート型とならないため，このタグの指定は無視されます。
			<file-name>	string	0 または 1 回	HTTP クライアントがダウンロードするときのファイル名称を指定します。 HTTP レスポンスでのマルチパート型データの各パートで，Content-Disposition ヘッダの filename 属性の値に対して付与する文字列です。 このタグを省略した場合は filename 属性が設定されません。
			<local-file-name>	string	1 回	HTTP クライアントにダウンロードする作業フォルダ内の中間ファイル名称を指定します。 このタグで指定した名称のファイルは，作業フォルダ内から読み込まれて HTTP レスポンスのボディ部分またはマルチパート型データの各パートのボディ部分に設定されます。 このタグを省略した場合，システム例外が発生します。
			<content-type>	string	0 または 1 回	HTTP クライアントに送信するファイルのメディアタイプを指定します。 HTTP レスポンスでのマルチパート型データの各パートで，Content-Type ヘッダに付与する値です。 このタグを省略した場合はメディアタイプが設定されません。空文字列が指定された場合は，この要素の指定を省略したとみなされます。 content-type 要素の charset 属性には，Content-Type ヘッダの charset 属性として使用する文字コードを指定します。空文字列が指定された場合は，この要素の指定を省略したとみなされます。content-type 要素の値に charset 属性やそのほかの属性を指定した場合，動作は保証されません。

タグ名		種別	出現回数	説明
	<http-header>	any	0 または 1 回	HTTP レスポンスヘッダの情報を指定します。要素名をヘッダフィールド，値をヘッダ値として，<http-header>の子要素として指定します。 指定しない場合は，HTTP レスポンスヘッダ定義ファイルの情報が HTTP レスポンスヘッダとして使用されます。

（凡例）

－：該当する項目はありません。

(d) 応答電文フォーマット（ヘッダ変数用）（JSON-XML 変換用）

JSON-XML 変換の定義を設定するヘッダ変数用の応答電文フォーマットを次に示します。この電文フォーマットのファイル名は「urecp_custom_response.xsd」です。名前空間は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/custom/response」です。

表 2-15 応答電文フォーマット（ヘッダ変数用）（JSON-XML 変換用）

タグ名		種別	出現回数※1	説明
<custom-reception-response>		－	1 回	－
	<json-transfer>	boolean	0 または 1 回	HTTP 受付のレスポンス処理時に，JSON-XML 変換するかどうかを指定するタグです。指定値は次のどちらかです。 true：JSON-XML 変換します。※2 false：JSON-XML 変換しません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 HTTP 受付定義ファイルの httprecp.response.switchover.json-transfer.mode プロパティの値が有効になります。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

HTTP 受付定義ファイルの httprecp.response.switchover.json-transfer.mode プロパティの値に「true」が設定されていない場合，JSON-XML 変換をしません。

(e) 応答電文フォーマット（ボディ変数用）（HTTP 受付用）

HTTP 受付がビジネスプロセスを呼び出したときに渡すボディ変数の応答電文フォーマットを次に示します。この電文フォーマットのファイル名は「urecp_http_dummy_body_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/response」です。

なお，この電文フォーマットは応答電文フォーマット（ボディ変数用）のダミーフォーマットです。応答電文（ヘッダ）の ignore-bodymsg 要素の値が「true」，または HTTP 受付定義ファイルの httprecp.response.ignore-bodymsg プロパティが「true」（応答電文（ボディ）を無視する設定）の場合に使用します。

応答電文（ボディ）を無視する設定の場合，HTTP 受付に応答電文フォーマット（ボディ変数用）を設定していないと，応答電文（ボディ）の変数が存在しないためにビジネスプロセスが変数を返すことができなくなります。このため，このダミーフォーマットを HTTP 受付に設定して応答電文（ボディ）の変数を作成し，応答アクティビティに割り当てることを推奨します。

表 2-16　応答電文フォーマット（ボディ変数用）（応答電文（ボディ）を無視する設定の場合）

タグ名		種別	出現回数	説明
<http-body-response>		－	1 回	－
	<any>	string	0 回以上	－

（凡例）

－：該当する項目はありません。

応答電文（ボディ）を無視しない設定の場合，応答電文フォーマット（ボディ変数用）は HTTP レスポンスに応じてユーザが作成してください。

(3) 電文フォーマットの作成方法

HTTP 受付で使用する電文フォーマットの作成方法について説明します。

電文フォーマットと使用する XML スキーマの関係を次の表に示します。

表 2-17　電文フォーマットと使用する XML スキーマの関係

電文フォーマット		説明	ユーザによる編集
要求電文	ヘッダ変数用（HTTP 受付用）	サービスプラットフォームが提供する次の XML スキーマファイルをそのまま利用します。 • urecp_http_header_request.xsd※3	不要
	ボディ変数用（HTTP 受付用）（メイン）※1	サービスプラットフォームが提供する次の XML スキーマファイルをそのまま利用します。 • urecp_http_body_request.xsd※3	不要

電文フォーマット		説明	ユーザによる編集
要求電文	ボディ変数用（HTTP 受付用）（詳細）※1	要求電文（ボディ）の値を利用する場合，サービスプラットフォームが提供する次の XML スキーマファイルを一部編集して利用します。 • urecp_http_body_detail_request.xsd※3 要求電文（ボディ）の値を利用しない場合はサービスプラットフォームが提供する XML スキーマファイルを利用しません。	任意
	ボディ変数用（JSON-XML 変換用）※2	JSON-XML 変換をする場合に JSON 変換ツールで作成します。 JSON 形式の要求電文，および応答電文のサンプルファイルを入力 JSON ファイルとして用意し，JSON 変換ツールで XSD ファイルを作成してください。	任意
応答電文	ヘッダ変数用（HTTP 受付用）	サービスプラットフォームが提供する次の XML スキーマファイルをそのまま利用します。 • urecp_http_header_response.xsd※3	不要
	ヘッダ変数用（JSON-XML 変換用）	サービスプラットフォームが提供する次の XML スキーマファイルをそのまま利用します。 • urecp_custom_response.xsd※4	不要
	ボディ変数用（HTTP 受付用）※1	サービスプラットフォームが提供する次の XML スキーマファイルをそのまま利用します。 • urecp_http_dummy_body_response.xsd※3 応答電文（ボディ）を無視する設定の場合に利用を推奨します。	不要
		応答電文（ボディ）を無視しない設定の場合に，HTTP リクエストの内容に応じて，ユーザが作成します。	必要
	ボディ変数用（JSON-XML 変換用）※2	JSON-XML 変換をする場合に JSON 変換ツールで作成します。 JSON 形式の要求電文，および応答電文のサンプルファイルを入力 JSON ファイルとして用意し，JSON 変換ツールで XSD ファイルを作成してください。	任意

注※1

HTTP 受付をパススルーモードで使用する場合，HTTP リクエストおよび HTTP レスポンスの内容に応じてユーザが作成する必要があります。

注※2

HTTP 受付で JSON-XML 変換の動作を変更する場合は，JSON-XML 変換定義ファイルを JSON 変換ツールの実行時に指定してください。

JSON 変換ツールの実行形式を次に示します。

```
cscjson2xsd -in 〈入力JSONファイルのパス〉 -out 〈出力XSDファイルのパス〉 [-f 〈JSON-XML変換定義ファイルのパス〉]
```

JSON 変換ツールの詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「14. JSON 変換ツールを使用したデータ変換」を参照してください。

注※3

サービスプラットフォームが提供する電文フォーマットの XML スキーマは，「<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¥custom-reception¥http¥schema」に格納されています。編集する場合はフォルダごと任意の場所にコピーしてから編集してください。

注※4

サービスプラットフォームが提供する電文フォーマットの XML スキーマは，「<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¥schema¥msg」に格納されています。XML スキーマはそのまま使用してください。編集した場合の動作は保障されません。

以降に，次に示す電文フォーマットの XML スキーマの作成方法を示します。

- 要求電文フォーマット（ボディ変数用）（HTTP 受付用）
- ユーザ作成の応答電文フォーマット（ボディ変数用）（HTTP 受付用）（ユーザ作成の場合）

(a) 要求電文フォーマット（ボディ変数用）（HTTP 受付用）の作成

1. XML スキーマファイル（urecp_http_body_detail_request.xsd）を XML エディタなどで開き，次の図の枠内で囲まれた部分を編集します。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema elementFormDefault="qualified"
  targetNamespace="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/request"
  xmlns:hrc="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/request"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:complexType name="http-body-requestType">
   <xsd:sequence>
    <!-- User Customize -->
    <!-- <xsd:element name="key" type="xsd:string" minOccurs="0" /> -->
   </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

（凡例）

：編集個所

XML スキーマファイルの編集時には次の点に注意してください。

- 要素を追加する場合は，<http-body-request>の子要素として定義してください。孫要素などの構造は定義しないでください。
- 定義した要素に対応するキーが HTTP リクエストのクエリ文字列内にない場合は，要求電文内で定義した要素は作成されません。
- 要素として定義していないキーが HTTP リクエストのクエリ文字列内にある場合，要求電文内でキーに対応する要素が作成されます。

2. XML スキーマファイルを上書き保存します。

(b) 応答電文フォーマット（ボディ変数用）（HTTP 受付用）の作成

1. XML エディタなどで任意の XML スキーマファイルを作成します。

応答電文フォーマット（ボディ変数用）の作成例については，「付録 F　HTTP 受付を利用したビジネスプロセスの設定例」を参照してください。

2. 「urecp_http_body_response.xsd」のファイル名で任意のディレクトリに保存します。

2.5.3　HTTP 受付の定義ファイルを作成する

ここでは，HTTP 受付で使用する定義ファイルの種類について説明します。

定義ファイルの種類を次に示します。

表 2–18　定義ファイルの種類

定義ファイル	説明	省略可否
HTTP レスポンスヘッダ定義ファイル	応答電文のヘッダ変数が省略されている場合に，デフォルトの HTTP レスポンスのヘッダ定義を指定するためのファイルです。	可
HTTP 受付定義ファイル	HTTP 受付を使用する際に必要なパラメタを設定するためのファイルです。	可※1
JSON-XML 変換定義ファイル	JSON-XML 変換の動作を変更するためのファイルです。	可※2
HTTP 受付サービス選択定義ファイル	ユーザ定義受付（呼出先選択）の HTTP 受付から呼び出すビジネスプロセスの情報と呼び出し条件を設定するためのファイルです。	可※3※4
HTTP 受付ルーティング定義ファイル	HTTP 受付から呼び出すビジネスプロセスや，HTTP 受付で実行する機能を設定するファイルです。	可※4

注※1

HTTP 受付定義ファイルを省略した場合，HTTP 受付定義ファイルのテンプレートファイルのデフォルト値が使用されます。

注※2

JSON-XML 変換定義ファイルを省略した場合，JSON-XML 変換定義ファイルのテンプレートファイルのデフォルト値が使用されます。

注※3

ユーザ定義受付（呼出先選択）の HTTP 受付を使用する場合は，省略できません。

注※4

HTTP 受付サービス選択定義ファイルと HTTP 受付ルーティング定義ファイルは，同時に使用できません。

次のファイルは，HTTP 受付の追加時に，サービスプラットフォームが提供するテンプレートファイルを編集して利用します。

- HTTP レスポンスヘッダ定義ファイル（cscurecphttp_header.properties）
- HTTP 受付定義ファイル（cscurecphttp.properties）
- HTTP 受付サービス選択定義ファイル（cscurecphttp_selection.properties）
- HTTP 受付ルーティング定義ファイル（cscurecphttp_routing.xml）

そのため，上記のファイルの作成は不要です。

JSON-XML 変換定義ファイルのテンプレートファイル（csc_json_converter.properties）は，「<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¥config¥msg¥templates」に格納されています。使用する場合は，テンプレートファイルを独自定義ファイルとして追加し，ファイルを編集してください。ファイル名は「csc_json_converter.properties」で固定です。

ファイルの内容については，ボディ変数用の電文フォーマットの作成時に使用した JSON-XML 変換定義ファイルと合わせてください。

HTTP 受付の JSON-XML 変換では，拡張モードを使用した JSON-XML 変換と標準モードを使用した JSON-XML 変換を行うことができます。HTTP 受付の JSON-XML 変換を使用するには，それぞれのモードに合わせた JSON 変換ツールを使用する必要があります。標準モードと拡張モードのどちらのモードを使用して JSON-XML 変換を行うかは，HTTP 受付定義ファイルの httprecp.json-xml-convert.mode プロパティで設定します。標準モードと拡張モードの差異については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「11.1.2　標準モードと拡張モードとの差異」を参照してください。

2.5.4　HTTP 受付を追加する

HTTP 受付を新規に追加する場合，特定のビジネスプロセスを呼び出すか，または呼び出しサービス選択機能で呼び出すビジネスプロセスを選択するかで，HTTP 受付を追加する手順が異なります。

ここでは，呼び出しサービス選択機能を使用しないで，特定のビジネスプロセスを呼び出す手順について説明します。

呼び出すビジネスプロセスを選択して HTTP 受付を追加する手順については，「付録 L　ユーザ定義受付（呼出先選択）の定義」を参照してください。

特定のビジネスプロセスを呼び出す HTTP 受付を新規に追加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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのビジネスプロセスで右クリックして［ユーザ定義受付（呼出先固定）追加］を選択します。
   受付種別選択ウィザードが表示されます。

4. 受付種別のドロップダウンリストから「HTTP 受付」を選択します。
   ［次へ］ボタンをクリックすると，HTTP 受付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. ［受付名］を指定します。
   HTTP 受付の名称を 1～40 バイトで指定します。

6. ［終了］ボタンをクリックします。
   ［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。ユーザ定義受付定義画面（基本）が表示されます。

7. HTTP 受付の定義情報を設定します。

HTTP 受付の設定項目について次の表に示します。

表 2-19　HTTP 受付のユーザ定義受付定義画面（基本）の設定項目

分類	項目		設定内容	設定
ユーザ定義受付情報	受付名		設定した受付名が表示されます。	◎
	受付 ID		受付 ID が表示されます。必要に応じて変更します。	◎
	デフォルトオペレーション名		デフォルトで呼び出すオペレーションを選択します。	◎
	オペレーション		任意のオペレーション名を指定します。	◎
オペレーション情報	通信モデル		「同期」を設定します。	◎
要求電文	[any 型を使う] チェックボックス		ボディ電文を空で送信する場合，チェックすることを推奨します。※1	○
	受付	電文フォーマット	urecp_http_body_request.xsd※2 を指定します。 [any 型を使う] チェックボックスをチェックしている場合は，電文フォーマットファイルを指定する必要はありません。	○
	サービス部品	[使う] チェックボックス	チェックしません。	－
応答電文	[any 型を使う] チェックボックス		チェックしません。	－
	受付	電文フォーマット	• 応答電文（ボディ）を無視する設定の場合 任意の電文フォーマットを指定します。サービスプラットフォームが提供する urecp_http_dummy_body_response.xsd※2 の利用を推奨します。 • 応答電文（ボディ）を無視しない設定の場合 「2.5.2(3)(b)　応答電文フォーマット（ボディ変数用）（HTTP 受付用）の作成」で作成した urecp_http_body_response.xsd を指定します。	◎
	サービス部品	[使う] チェックボックス	チェックしません。	－

（凡例）

◎：必ず設定します。

○：任意で設定します。

－：該当しません。

注※1

パススルーモードを有効にしてメッセージボディを指定しない場合，空文字のボディ電文をビジネスプロセスに渡すため，any 型の利用を推奨します。

注※2

ファイルの格納先ディレクトリは「<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¥custom-reception¥http¥schema」です。

上記以外のユーザ定義受付定義画面の設定項目については，マニュアル「サービスプラットフォーム リファレンス」の「1.2.5　ユーザ定義受付定義画面」を参照してください。

8. ユーザ定義受付（詳細）タブをクリックします。

ユーザ定義受付定義画面（詳細）が表示されます。

9. HTTP 受付の定義情報を設定します。

HTTP 受付の設定項目について次の表に示します。

表 2-20　HTTP 受付のユーザ定義受付定義画面（詳細）の設定項目

分類	項目	設定内容	設定
ユーザ定義受付制御情報	独自定義ファイル	次の独自定義ファイルが表示されます。必要に応じて変更します。 • cscurecphttp.properties※1 • cscurecphttp_header.properties※2 ここに設定されているのはテンプレートファイルです。修正が必要な場合はファイルを選択して［編集］ボタンをクリックし，独自定義ファイルの内容を修正してください。	◎
	EAR ファイル	次の EAR ファイルが設定されていることを確認します。 • cscmsg_urecp_http.ear※3	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※1

cscurecphttp.properties の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.11.3　HTTP 受付定義ファイル」を参照してください。

注※2

cscurecphttp_header.properties の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.11.1　HTTP レスポンスヘッダ定義ファイル」を参照してください。

注※3

ファイルの格納先ディレクトリは「<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¥custom-reception¥http¥lib」です。

2.5.5　アクティビティを設定する

ここでは，HTTP 受付を定義したあとに，ビジネスプロセス定義画面で受付アクティビティおよび応答アクティビティを設定する方法について説明します。

アクティビティの設定では，要求電文用および応答電文用の変数を定義したあと，受付アクティビティおよび応答アクティビティに定義した変数を割り当てます。

(1) 変数の定義

［変数・相関セット一覧］ダイアログで次の変数を定義します。

- 要求電文（ヘッダ）の変数
- 応答電文（ヘッダ）の変数
- 要求電文（ボディ）の変数
- 応答電文（ボディ）の変数

変数の定義手順を次に示します。

(a) 要求電文（ヘッダ）および応答電文（ヘッダ）の変数

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. ［変数名］に任意の名称を指定します。
3. ［種別］ドロップダウンリストから「XML」を選択します。
4. ［電文フォーマット］の［参照］ボタンをクリックします。
   電文フォーマットを指定するダイアログが表示されます。
5. 変数に応じて次のファイルを指定します。
   - 要求電文（ヘッダ）の変数
     <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¥custom-reception¥http¥schema¥urecp_http_header_request.xsd
   - 応答電文（ヘッダ）の変数
     <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¥custom-reception¥http¥schema¥urecp_http_header_response.xsd
6. ［変数・相関セット一覧］ダイアログで［追加］ボタンをクリックします。
   変数が追加されます。
   操作 2.から操作 6.と同様の手順で応答電文（ヘッダ）の変数を追加します。

(b) 要求電文（ボディ）および応答電文（ボディ）の変数

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. ［変数名］に任意の名称を指定します。
3. ［種別］ドロップダウンリストから「XML」を選択します。

4. [電文フォーマット] の [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. [受付名] をチェックし，ドロップダウンリストからユーザ定義受付定義画面で設定した HTTP 受付の名称を選択します。

6. [オペレーション名] ドロップダウンリストから HTTP 受付に設定したオペレーションを選択します。

7. 変数に応じて [電文種別] ドロップダウンリストで次のように選択します。
   - 要求電文（ボディ）の変数の場合
     [要求電文（ボディ）] を選択します。
   - 応答電文（ボディ）の変数の場合
     [応答電文（ボディ）] を選択します。

8. [電文フォーマット] に任意の名称を入力します。

9. [OK] ボタンをクリックします。

10. [変数・相関セット一覧] ダイアログで [追加] ボタンをクリックします。
    変数が追加されます。
    操作 2.から操作 10.と同様の手順で応答電文（ボディ）の変数を追加します。

(2) 変数の割り当て

受付アクティビティおよび応答アクティビティに追加した変数を割り当てる手順を次に示します。

(a) 受付アクティビティの場合

1. ビジネスプロセス定義画面のキャンバス上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. [アクティビティ名] に任意の名称を入力します。

3. [オペレーション名] の [取込] ボタンをクリックします。
   [オペレーション名の取込] ダイアログが表示されます。

4. [受付名] に該当する HTTP 受付の名称を設定し，[オペレーション名] に取り込み対象となるオペレーション名を設定します。

5. [OK] ボタンをクリックします。

6. [ボディ割当変数] ドロップダウンリストから，定義した要求電文（ボディ）の変数を選択します。

7. [ヘッダ割当変数] の [設定] ボタンをクリックします。
   [ヘッダ割当変数] ダイアログが表示されます。

8. [追加] ボタンをクリックします。

9. [割当変数] セルのドロップダウンリストから，定義した要求電文（ヘッダ）の変数を選択します。

10. [ルート要素] セルのドロップダウンリストからルート要素を選択します。

11. [OK] ボタンをクリックします。
［ヘッダ割当変数］ダイアログが閉じます。

12. [受付アクティビティ] ダイアログで［OK］ボタンをクリックします。

(b) 応答アクティビティの場合

1. ビジネスプロセス定義画面のキャンバス上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. [アクティビティ名] に任意の名称を入力します。

3. [オペレーション名] の［取込］ボタンをクリックします。
［オペレーション名の取込］ダイアログが表示されます。

4. [ユーザ定義受付から取込] ラジオボタンを選択し，［受付名］を選択します。

5. [オペレーション名] に HTTP 受付に設定したオペレーション名を設定します。

6. [OK] ボタンをクリックします。

7. [ボディ割当変数] ドロップダウンリストから，定義した応答電文（ボディ）の変数を選択します。

8. [ヘッダ割当変数] の［設定］ボタンをクリックします。
［ヘッダ割当変数］ダイアログが表示されます。

9. [追加] ボタンをクリックします。

10. [割当変数] セルのドロップダウンリストから，定義した応答電文（ヘッダ）の変数を選択します。

11. [ルート要素] セルのドロップダウンリストからルート要素を選択します。

12. [OK] ボタンをクリックします。
［ヘッダ割当変数］ダイアログが閉じます。

13. [応答アクティビティ] ダイアログで［OK］ボタンをクリックします。

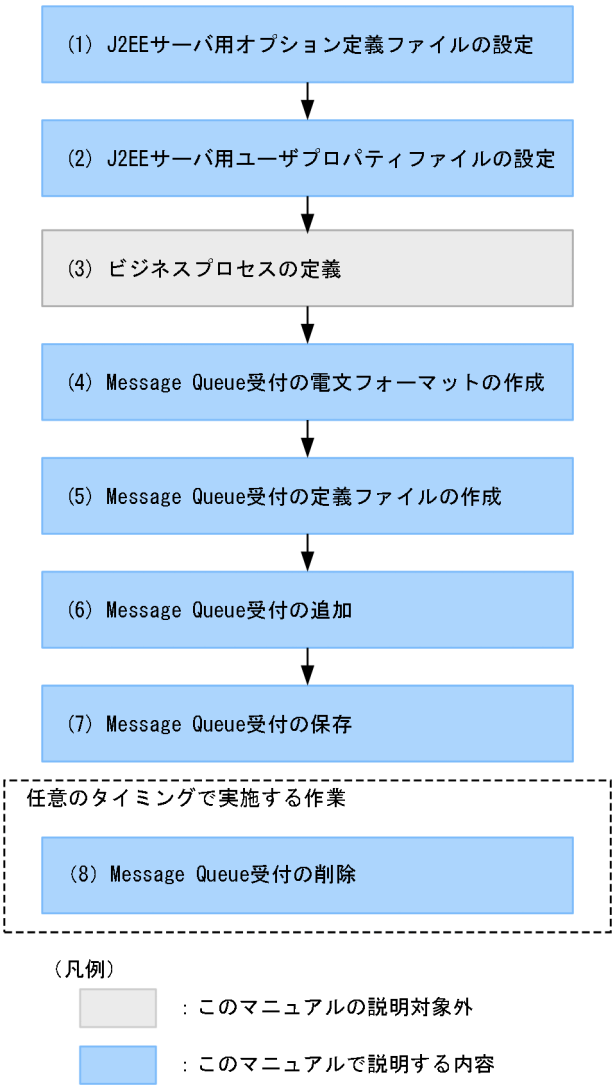
2.6　Message Queue 受付の定義

この節では，Message Queue 受付の電文フォーマットの作成，Message Queue 受付の追加など，Message Queue 受付の定義方法について説明します。

2.6.1　Message Queue 受付の定義の流れ

Message Queue 受付の定義の流れを次に示します。

図 2-7　Message Queue 受付の定義の流れ



Message Queue 受付を定義する場合の作業について次に説明します。

(1)　J2EE サーバ用オプション定義ファイルの設定

実行環境で Message Queue 受付が動作するために，J2EE サーバのクラスパスに IBM MQ のライブラリ（IBM MQ の jar）を追加してください。

(2)　J2EE サーバ用ユーザプロパティファイルの設定

実行環境で Message Queue 受付が動作するために，J2EE サーバを実行する JavaVM のシステムプロパティにライトトランザクション機能（ejbserver.distributedtx.XATransaction.enabled=true）を設定してください。

(3)　ビジネスプロセスの定義

ビジネスプロセス追加用のウィザードおよび定義済みのビジネスプロセスを利用して，ビジネスプロセスを追加します。追加したビジネスプロセスは，ビジネスプロセス定義画面で定義します。ビジネスプロセスの定義の詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.　ビジネスプロセスの定義」を参照してください。

また，ビジネスプロセスからサービス部品を呼び出す場合，サービスアダプタを定義する必要があります。サービスアダプタの定義の詳細は，「3.　サービスアダプタの定義」を参照してください。

(4)　Message Queue 受付の電文フォーマットの作成

Message Queue 受付で使用する電文フォーマットを作成します。

電文フォーマットの作成方法については，「2.6.2　Message Queue 受付の電文フォーマットを作成する」を参照してください。

(5)　Message Queue 受付の定義ファイルの作成

Message Queue 受付で使用する定義ファイルを作成します。作成した定義ファイルはリポジトリに反映されます。

Message Queue 受付で使用する定義ファイルの作成方法については，「2.6.3　Message Queue 受付の定義ファイルを作成する」を参照してください。

(6)　Message Queue 受付の追加

ツリービューのサービス定義一覧で Message Queue 受付を追加，および内容を定義します。

Message Queue 受付の追加および定義手順については，「2.6.4　Message Queue 受付を追加する」を参照してください。

(7)　Message Queue 受付の保存

追加および定義済みの Message Queue 受付をリポジトリに保存します。Message Queue 受付の保存手順については，「2.13　ユーザ定義受付の保存」を参照してください。

(8)　Message Queue 受付の削除

不要になった Message Queue 受付は，必要に応じて削除できます。

Message Queue 受付の削除については，「2.16　ユーザ定義受付の削除」を参照してください。

2.6.2　Message Queue 受付の電文フォーマットを作成する

Message Queue 受付で使用する電文フォーマットの形式，および作成方法について説明します。

(1)　電文フォーマットの形式

Message Queue 受付で扱う JMS メッセージは，JMS ヘッダ，プロパティ，およびボディで構成されます。

Message Queue 受付で使用する電文フォーマットの形式について説明します。

(a)　ヘッダ電文の形式

Message Queue 受付からビジネスプロセスに渡す要求電文のヘッダには，JMS メッセージの JMS ヘッダとプロパティに指定されたデータを設定します。

Message Queue 受付からビジネスプロセスに渡す要求電文フォーマット（ヘッダ変数用）を次に示します。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/mq/header」です。

表 2-21　Message Queue 受付のヘッダ電文の要求電文フォーマット

タグ名			種別	出現回数	説明
<mq_header>			－	1 回	－
	<jms_header>		string	1 回	JMS ヘッダの内容が設定されます。
		<JMSDestination>	string	1 回	メッセージが設定されているキュー名が設定されます。
		<JMSDeliveryMode>	string	1 回	メッセージの永続性を示す値が設定されます。
		<JMSMessageID>	string	1 回	プロバイダによって送信される各メッセージを一意に識別するメッセージ識別子が設定されます。
		<JMSTimestamp>	string	1 回	メッセージの送信時刻を表す値が設定されます。（単位：ミリ秒）
		<JMSExpiration>	string	1 回	メッセージの有効期間となる時刻を表す値が設定されます。（単位：ミリ秒）
		<JMSRedelivered>	string	1 回	メッセージが再配送中であるかどうかを示す値が設定されます。
		<JMSPriority>	string	1 回	メッセージのプライオリティが設定されます。
		<JMSReplyTo>	string	1 回	応答先のあて先が設定されます。 この情報は，Message Queue 受付では使用しません。

タグ名			種別	出現回数	説明
		<JMSCorrelationID>	string	1 回	あるメッセージをほかのメッセージと関連づけるための相関識別子が設定されます。
		<JMSType>	string	1 回	メッセージが送信されるときに，アプリケーションによって提供されるメッセージタイプが設定されます。
		<JMSHeaderExtension>	string	0 回以上	次の値が設定されます。 属性に"name"を持ち，JMS ヘッダの名称が設定されます。 値には，name に設定された JMS ヘッダの値が設定されます。
	<user_property>		−	1 回	ユーザが設定したプロパティの内容が設定されます。
		<property>	string	0 回以上	次の値が設定されます。このタグは設定されたプロパティの個数分生成されます。 属性に"name"を持ち，プロパティの名称が設定されます。 値には，name に設定されたプロパティの値が設定されます。

（凡例）

−：該当する項目はありません。

(b)　ボディ電文の形式

要求電文のボディに格納された本文は，ボディ電文としてビジネスプロセスに渡します。

ボディ電文の種別は，JMS メッセージのインターフェースとユーザ定義受付定義の設定によって異なります。ボディ電文の電文種別を次に示します。

表 2-22　ボディ電文の電文種別

インターフェース名	ユーザ定義受付定義の設定 （要求電文フォーマット指定あり）	
	any 型の指定なし	any 型の指定あり
BytesMessage	バイナリ電文	任意形式の電文
TextMessage	XML 電文	任意形式の電文

(2)　電文フォーマットの作成方法

Message Queue 受付で使用する電文フォーマットの作成方法について説明します。

(a) ヘッダ電文の作成

Message Queue 受付のヘッダ電文の電文フォーマットは，サービスプラットフォームが提供する XML スキーマを編集して利用します。そのため，ヘッダ電文の XML スキーマの作成は不要です。

サービスプラットフォームが提供する電文フォーマットの XML スキーマは，「<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¥custom-reception¥mq¥schema」に格納されています。

サービスプラットフォームが提供する XML スキーマの内容を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- All Rights Reserved. Copyright (C) 2014, Hitachi, Ltd. -->
<xsd:schema elementFormDefault="qualified"
            targetNamespace="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/mq/he
ader"
            xmlns:mhc="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/mq/header"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <xsd:element name="mq_header">
  <xsd:complexType>
   <xsd:sequence>
     <xsd:element name="jms_header" maxOccurs="1" minOccurs="0">
      <xsd:complexType>
       <xsd:sequence>
        <xsd:element name="JMSDestination" maxOccurs="1" minOccurs="0" nillable="true" type=
"xsd:string"/>
        <xsd:element name="JMSDeliveryMode" maxOccurs="1" minOccurs="0" nillable="true" type
="xsd:string"/>
        <xsd:element name="JMSMessageID" maxOccurs="1" minOccurs="0" nillable="true" type="x
sd:string"/>
        <xsd:element name="JMSTimestamp" maxOccurs="1" minOccurs="0" nillable="true" type="x
sd:string"/>
        <xsd:element name="JMSExpiration" maxOccurs="1" minOccurs="0" nillable="true" type="
xsd:string"/>
        <xsd:element name="JMSRedelivered" maxOccurs="1" minOccurs="0" nillable="true" type=
"xsd:string"/>
        <xsd:element name="JMSPriority" maxOccurs="1" minOccurs="0" nillable="true" type="xs
d:string"/>
        <xsd:element name="JMSReplyTo" maxOccurs="1" minOccurs="0" nillable="true" type="xsd
:string"/>
        <xsd:element name="JMSCorrelationID" maxOccurs="1" minOccurs="0" nillable="true" typ
e="xsd:string"/>
        <xsd:element name="JMSType" maxOccurs="1" minOccurs="0" nillable="true" type="xsd:st
ring"/>
        <xsd:element name=" JMSHeaderExtension " minOccurs="0" maxOccurs="unbounded" >
         <xsd:complexType>
          <xsd:simpleContent>
           <xsd:extension base="xsd:string">
            <xsd:attribute name="name" type="xsd:string" use="optional"/>
           </xsd:extension>
          </xsd:simpleContent>
         </xsd:complexType>
        </xsd:element>
       </xsd:sequence>
      </xsd:complexType>
     </xsd:element>
```

```
    <xsd:element name="user_property" maxOccurs="1" minOccurs="0">
     <xsd:complexType>
      <xsd:sequence>
       <xsd:element name="property" minOccurs="0" maxOccurs="unbounded" >
        <xsd:complexType>
         <xsd:simpleContent>
          <xsd:extension base="xsd:string">
           <xsd:attribute name="name" type="xsd:string" use="optional"/>
          </xsd:extension>
         </xsd:simpleContent>
        </xsd:complexType>
       </xsd:element>
      </xsd:sequence>
     </xsd:complexType>
    </xsd:element>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

(b) ボディ電文の作成

Message Queue 受付のボディ電文の電文フォーマットについては，スキーマの形式は任意です。ボディ電文の電文種別に合わせてスキーマを作成する必要があります。なお，電文種別が任意形式（any 型の指定あり）の場合はスキーマの作成は不要です。

2.6.3 Message Queue 受付の定義ファイルを作成する

Message Queue 受付の定義ファイルを次に示します。

- Message Queue 受付定義ファイル

  Message Queue 受付を使用する際に必要なパラメタを設定するためのファイルです。Message Queue 受付配備時に，Message Queue 受付定義ファイルに設定した内容がアプリケーション統合属性ファイルに書き込まれ，Message Queue 受付起動時に有効になります。

Message Queue 受付定義ファイルは，Message Queue 受付の追加時に，サービスプラットフォームが提供するテンプレートファイル（cscurecpmq.properties）を編集して利用します。そのため，Message Queue 受付定義ファイルの作成は不要です。

Message Queue 受付定義ファイルの記載例を次に示します。

```
resource-adapter=IBM_MQ_Resource_Adapter
activation-config.destination=QUEUE.NAME
activation-config.username=user1
```

2.6.4 Message Queue 受付を追加する

Message Queue 受付を新規に追加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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのビジネスプロセスで右クリックして［ユーザ定義受付（呼出先固定）追加］を選択します。
   受付種別選択ウィザードが表示されます。

4. 受付種別のドロップダウンリストから「Message Queue 受付」を選択します。
   ［次へ］ボタンをクリックすると，Message Queue 受付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. ［受付名］を指定します。
   Message Queue 受付の名称を 1～40 バイトで指定します。

6. ［終了］ボタンをクリックします。
   ［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。ユーザ定義受付定義画面（基本）が表示されます。

7. Message Queue 受付の定義情報を設定します。
   Message Queue 受付の設定項目について次の表に示します。

表 2-23　Message Queue 受付のユーザ定義受付定義画面（基本）の設定項目

分類	項目	設定内容	設定
ユーザ定義受付情報	受付名	設定した受付名が表示されます。	◎
	受付 ID	受付 ID が表示されます。必要に応じて変更します。	◎
	デフォルトオペレーション名	デフォルトで呼び出すオペレーションを選択します。	◎
	オペレーション	次の中から追加するオペレーション種別を選択して，オペレーション名を指定します。 • 終了時コミット Message Queue 受付の標準パターンです。JMS プロバイダとのトランザクションは，ビジネスプロセス実行終了後にコミットされます。 ビジネスプロセスの通信モデルは非同期となります。 • 任意時コミット	◎

分類	項目		設定内容	設定
ユーザ定義受付情報	オペレーション		Message Queue 受付の応用パターンです。JMS プロバイダとのトランザクションは，ビジネスプロセスの応答アクティビティ処理完了後にコミットされます。 ビジネスプロセスの通信モデルは同期となります。 応答アクティビティ処理完了後のビジネスプロセス処理で例外が発生した場合，リクエストを再送信する必要があります。	◎
オペレーション情報	通信モデル		「同期」または「非同期」を設定します。	◎
要求電文	[any 型を使う] チェックボックス		任意の形式の要求電文（ボディ）を使用する場合はチェックします。	○
	受付	電文フォーマット	[any 型を使う] チェックボックスをチェックしていない場合は，任意の電文フォーマットファイルを指定します。 [any 型を使う] チェックボックスをチェックしている場合は，電文フォーマットファイルを指定する必要はありません。	○
	サービス部品	[使う] チェックボックス	チェックしません。	－
応答電文	[any 型を使う] チェックボックス		任意の形式の応答電文（ボディ）を使用する場合はチェックします。	○
	受付	電文フォーマット	任意の電文フォーマットファイルを指定します。	○
	サービス部品	[使う] チェックボックス	チェックしません。	－

（凡例）

◎：必ず設定します。

○：任意で設定します。

－：該当しません。

8. ユーザ定義受付（詳細）タブをクリックします。

ユーザ定義受付定義画面（詳細）が表示されます。

9. Message Queue 受付の定義情報を設定します。

Message Queue 受付の設定項目について次の表に示します。

表 2-24　Message Queue 受付のユーザ定義受付定義画面（詳細）の設定項目

分類	項目	設定内容	設定
ユーザ定義受付制御情報	独自定義ファイル	次の独自定義ファイルが表示されます。必要に応じて変更します。 • cscurecpmq.properties※1	◎
	EAR ファイル	次の EAR ファイルが設定されていることを確認します。 • cscmsg_urecp_mq.ear※2	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※1

設定されている cscurecpmq.properties はテンプレートファイルです。必要に応じて cscurecpmq.properties を選択して［編集］ボタンをクリックし，独自定義ファイルの内容を修正してください。cscurecpmq.properties の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.9.1　Message Queue 受付定義ファイル」を参照してください。

注※2

ファイルの格納先ディレクトリは「<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¥custom-reception¥mq¥lib」です。

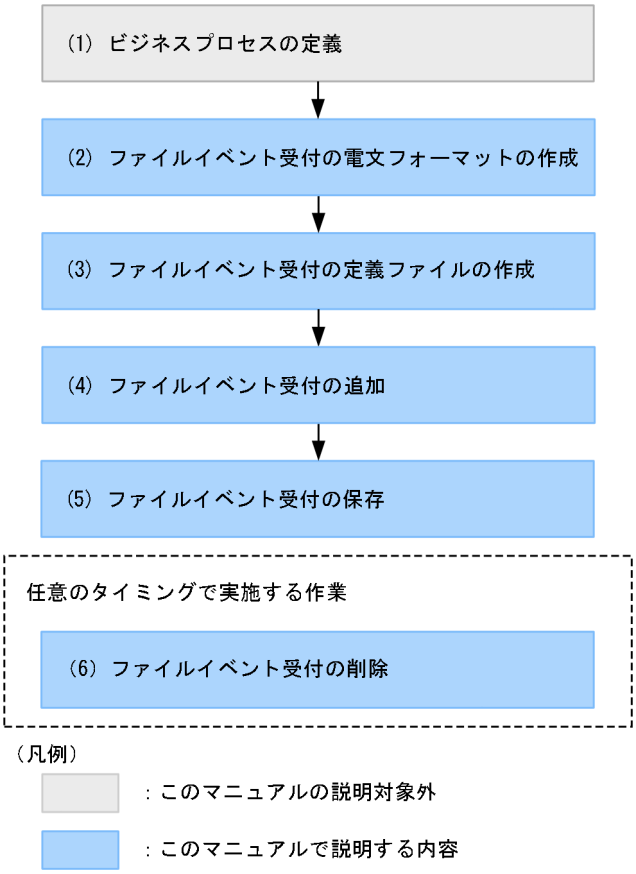
2.7　ファイルイベント受付の定義

この節では，開発環境でのファイルイベント受付の定義方法について説明します。

2.7.1　ファイルイベント受付の定義の流れ

ファイルイベント受付の流れを次に示します。

図 2-8　ファイルイベント受付の定義の流れ



ファイルイベント受付を定義する場合の作業について次に説明します。

(1)　ビジネスプロセスの定義

ビジネスプロセス追加用のウィザードおよび定義済みのビジネスプロセスを利用して，ビジネスプロセスを追加します。追加したビジネスプロセスは，ビジネスプロセス定義画面で定義します。ビジネスプロセスの定義の詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.　ビジネスプロセスの定義」を参照してください。

また，ビジネスプロセスからサービス部品を呼び出す場合，サービスアダプタを定義する必要があります。サービスアダプタの定義の詳細は，「3.　サービスアダプタの定義」を参照してください。

(2)　ファイルイベント受付の電文フォーマットの作成

ファイルイベント受付で使用する要求電文フォーマットおよび応答電文フォーマットを作成します。

電文フォーマットの作成方法については，「2.7.2　ファイルイベント受付の電文フォーマットを作成する」を参照してください。

(3)　ファイルイベント受付の定義ファイルの作成

ファイルイベント受付で使用する定義ファイルを作成します。ファイルイベント受付で使用する定義ファイルの作成方法については，「2.7.3　ファイルイベント受付の定義ファイルを作成する」を参照してください。

(4)　ファイルイベント受付の追加

実行環境でファイルイベント受付を追加してください。追加方法については，「2.7.4　ファイルイベント受付を追加する」を参照してください。

(5)　ファイルイベント受付の保存

追加および定義済みのファイルイベント受付をリポジトリに保存します。ファイルイベント受付の保存手順については，「2.13　ユーザ定義受付の保存」を参照してください。

(6)　ファイルイベント受付の削除

不要になったファイルイベント受付は，必要に応じて削除できます。

ファイルイベント受付の削除については，「2.16　ユーザ定義受付の削除」を参照してください。

2.7.2　ファイルイベント受付の電文フォーマットを作成する

ファイルイベント受付の要求電文および応答電文の電文フォーマットは，サービスプラットフォームが提供するスキーマを使用します。そのため，電文フォーマットの定義ファイルの作成は不要です。

ファイルイベント受付で使用する電文フォーマットの種類，および内容について説明します。

ファイルの格納場所は「<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¥custom-reception¥fileevent¥schema」です。

(1)　ファイルイベント受付の要求電文フォーマット

ファイルイベント受付がビジネスプロセスを呼び出すときに，ビジネスプロセスへ渡す要求電文のフォーマットを次に示します。この電文フォーマットのファイル名は「urecp_fileevent_request.xsd」です。名

前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/fileevent/request」です。

表 2-25　ファイルイベント受付の要求電文フォーマット

タグ名		種別	出現回数	説明
<request>		−	1 回	−
	<common-folder-def-name>	string	1 回	取得対象ファイルが格納されている共通フォルダの定義名です。
	<sub-folder>	string	0 または 1 回	共通フォルダのサブフォルダです。 ファイルイベント受付定義ファイルの urecp-fileevent.sub-folder プロパティが設定されている場合に値が設定されます。 urecp-fileevent.sub-folder プロパティが設定されていない場合は，要素は出現しません。
	<file-name>	string	1 回	取得対象ファイル名です。 共通フォルダを監視する場合は，監視対象はリネーム後のファイル名です。 リネーム後のファイル名については，詳細はマニュアル「サービスプラットフォーム 解説」の「9.1.2　監視対象のファイルの状態管理機能」を参照してください。
	<file-size>	long	1 回	取得対象ファイルのファイルサイズです。

（凡例）

−：該当する項目はありません。

(2)　ファイルイベント受付の応答電文フォーマット

ファイルイベント受付の応答電文のフォーマットを次に示します。この電文フォーマットのファイル名は「urecp_fileevent_response.xsd」です。名前空間は，「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/fileevent/response」です。

表 2-26　ファイルイベント受付の応答電文フォーマット

タグ名		種別	出現回数	説明
<response>		−	1 回	−
	<message>	string	0 または 1 回	ビジネスプロセスの終了状態を示すメッセージです。タグの設定値をファイルイベントトレースの付加情報に出力します。
	<success>	boolean	0 または 1 回	ビジネスプロセスの終了状態を示すフラグです。 次のどちらかを指定します。 • true ビジネスプロセスの処理が正常に終了した。 • false

タグ名		種別	出現回数	説明
	<success>	boolean	0 または 1 回	ビジネスプロセスの処理でエラーが発生した。 省略した場合，または上記以外の値を設定した場合は true が設定されます。 なお，タグの設定値によって，ファイルイベントトレースの出力内容が変わります。ファイルイベントトレースについては，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「7.7.12　ファイルイベント連携システムでの障害対策」を参照してください。

（凡例）

－：該当する項目はありません。

2.7.3　ファイルイベント受付の定義ファイルを作成する

ファイルイベント受付では，次の定義ファイルを作成します。

- ファイルイベント受付定義ファイル

  ファイルイベント受付を使用する際に必要なパラメタを設定するためのファイルです。

  ファイルイベント受付定義ファイルは，ファイルイベント受付の追加時に，サービスプラットフォームが提供するテンプレートファイル（cscurecpfileevent.properties）を編集して利用できるため，新規に作成する必要はありません。

注意事項

ファイルイベント受付定義ファイルの設定を省略すると，テンプレートファイルの各デフォルト値が使用されます。ただし，監視サブフォルダについてはデフォルト値がないため，ファイルイベント受付定義ファイルで設定（監視サブフォルダの指定）を省略しないでください。ファイルイベント受付定義ファイルの設定については，マニュアル「サービスプラットフォーム リファレンス」の「3.12.1　ファイルイベント受付定義ファイル」を参照してください。

2.7.4　ファイルイベント受付を追加する

ファイルイベント受付を新規に追加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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。

   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのビジネスプロセスで右クリックして［ユーザ定義受付（呼出先固定）追加］を選択します。

受付種別選択ウィザードが表示されます。

4. 受付種別のドロップダウンリストから「ファイルイベント受付」を選択します。

［次へ］ボタンをクリックすると，ファイルイベント受付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. ［受付名］を指定します。

ファイルイベント受付の名称を 1～40 バイトで指定します。

6. ［終了］ボタンをクリックします。

［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。ユーザ定義受付定義画面（基本）が表示されます。

7. ファイルイベント受付の定義情報を設定します。

ファイルイベント受付の設定項目について次の表に示します。

表 2-27　ファイルイベント受付のユーザ定義受付定義画面（基本）の設定項目

分類	項目		設定内容	設定
ユーザ定義受付情報	受付名		設定した受付名が表示されます。	◎
	受付 ID		受付 ID が表示されます。必要に応じて変更します。	◎
	デフォルトオペレーション名		「FileEvent」で固定です。	◎
	オペレーション		「FileEvent」で固定です。	◎
オペレーション情報	通信モデル		「同期」または「非同期」を設定します。	◎
要求電文	［any 型を使う］チェックボックス		チェックしません。	－
	受付	電文フォーマット	urecp_fileevent_request.xsd が指定されています。	◎
	サービス部品	［使う］チェックボックス	チェックしません。	－
応答電文	［any 型を使う］チェックボックス		チェックしません。	－
	受付	電文フォーマット	urecp_fileevent_response.xsd が指定されています。	◎※
	サービス部品	［使う］チェックボックス	チェックしません。	－

（凡例）

◎：必ず設定します。

－：該当しません。

注※

通信モデルで「非同期」を指定した場合，応答電文は使用できないため非活性になり，設定できません。

8. ユーザ定義受付（詳細）タブをクリックします。

ユーザ定義受付定義画面（詳細）が表示されます。

9. ファイルイベント受付の定義情報を設定します。

ファイルイベント受付の設定項目について次の表に示します。

表 2-28　ファイルイベント受付のユーザ定義受付定義画面（詳細）の設定項目

分類	項目	設定内容	設定
ユーザ定義受付制御情報	独自定義ファイル	次の独自定義ファイルが表示されます。必要に応じて変更します。 • cscurecpfileevent.properties※1	◎
	EAR ファイル	次の EAR ファイルが設定されていることを確認します。 • cscmsg_urecp_fileevent.ear※2	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※1

設定されている cscurecpfileevent.properties はテンプレートファイルです。必要に応じて cscurecpfileevent.properties を選択して［編集］ボタンをクリックし，独自定義ファイルの内容を修正してください。cscurecpfileevent.properties の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.12.1　ファイルイベント受付定義ファイル」を参照してください。

注※2

ファイルの格納先ディレクトリは「<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¥custom-reception¥fileevent¥lib」です。

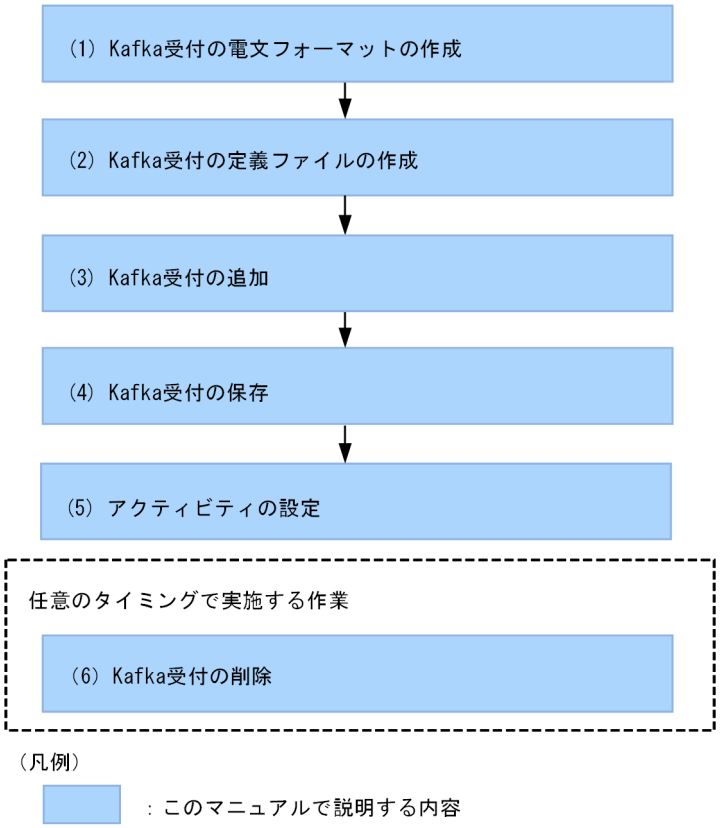
2.8　Kafka 受付の定義

この節では，開発環境での Kafka 受付の定義方法について説明します。

2.8.1　Kafka 受付の定義の流れ

Kafka 受付の流れを次に示します。

図 2-9　Kafka 受付の定義の流れ



Kafka 受付を定義する場合の作業について次に説明します。

(1)　Kafka 受付の電文フォーマットの作成

Kafka 受付で使用する要求電文フォーマットおよび応答電文フォーマットを作成します。

電文フォーマットの作成方法については，「2.8.2　Kafka 受付の電文フォーマットを作成する」を参照してください。

(2)　Kafka 受付の定義ファイルの作成

Kafka 受付で使用する定義ファイルを作成します。Kafka 受付で使用する定義ファイルの作成方法については，「2.8.3　Kafka 受付の定義ファイルを作成する」を参照してください。

(3)　Kafka 受付の追加

実行環境で Kafka 受付を追加します。追加方法については，「2.8.4　Kafka 受付を追加する」を参照してください。

(4)　Kafka 受付の保存

追加および定義済みの Kafka 受付をリポジトリに保存します。Kafka 受付の保存手順については，「2.13 ユーザ定義受付の保存」を参照してください。

(5)　アクティビティの設定

Kafka 受付の要求電文／応答電文フォーマット用のボディ変数とヘッダ変数をビジネスプロセスのアクティビティに設定します。

アクティビティの設定方法については，「2.8.5　アクティビティを設定する」を参照してください。

(6)　Kafka 受付の削除

不要になった Kafka 受付は，必要に応じて削除できます。

Kafka 受付の削除については，「2.16　ユーザ定義受付の削除」を参照してください。

2.8.2　Kafka 受付の電文フォーマットを作成する

Kafka 受付では要求電文フォーマットを使用します。応答電文フォーマットの設定は不要です。

Kafka 受付で使用する要求電文フォーマットの形式と作成方法について説明します。

(1)　ヘッダ電文の電文フォーマット

Kafka 受付のヘッダ電文の電文フォーマットは，サービスプラットフォームが提供するスキーマを使用します。そのため，ヘッダ電文の定義ファイルの作成は不要です。

サービスプラットフォームが提供する電文フォーマットの XML スキーマは，「<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¥custom-reception¥kafka¥schema」に格納されています。

Kafka 受付の要求電文フォーマットの形式を次に示します。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/kafka/request」です。

表 2-29　Kafka 受付のヘッダ電文の要求電文フォーマット

タグ名	種別	出現回数	説明
<request>	−	1 回	−

タグ名	種別	出現回数	説明
<record>	−	1 回	レコードの情報
<topic>	string	1 回	レコードが格納されているトピックのトピック名が設定されます。
<headers>	−	0 または 1 回	ヘッダ一覧が設定されます。 ヘッダが設定されている場合だけ出現します。
<header>	−	1 回以上	headers 要素がある場合だけ出現します。
<key>	string	1 回	ヘッダのキーが設定されます。 header 要素がある場合だけ出現します。
<value>	string	0 または 1 回	ヘッダの値が設定されます。 header 要素がある場合だけ出現します。ただし，値が null の場合は出現しません。 また，空のバイト配列が設定されていた場合は，空要素となります。
<key>	string	0 または 1 回	レコードのキーです。レコードのキーが null の場合は，出現しません。また，キーがバイト配列で，かつ空のバイト配列が設定されていた場合は，空要素となります。
<partition>	int	1 回	レコードが格納されているパーティション ID です。
<offset>	long	1 回	レコードのオフセットです。
<timestamp>	long	1 回	レコードに設定されているタイムスタンプです。

（凡例）

−：該当する項目はありません。

(2) ボディ電文の電文フォーマット

Kafka 受付のボディ電文の電文フォーマットについては，スキーマの形式は任意です。ボディ電文の電文種別に合わせてスキーマを作成する必要があります。

ボディ電文は，次の電文形式を使用できます。

- XML 形式
- バイナリ形式
- any

JSON-XML 変換を行う場合は，ボディ電文の電文フォーマットを JSON 変換ツールで生成する必要があります。JSON 変換ツールの詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「14.　JSON 変換ツールを使用したデータ変換」を参照してください。

2.8.3　Kafka 受付の定義ファイルを作成する

Kafka 受付では，次の定義ファイルを作成します。

- Kafka 受付定義ファイル

  Kafka 受付や Kafka インバウンドアダプタの動作を定義するファイルです。Kafka 受付の配備時，Kafka 受付の起動時，および csccmctl コマンドの実行時に読み込まれます。

  Kafka 受付定義ファイルは，Kafka 受付の追加時に，サービスプラットフォームが提供するテンプレートファイル（cscurecpkafka.properties）を編集して利用します。そのため，Kafka 受付定義ファイルの作成は不要です。
- JSON-XML 変換定義ファイル

  JSON-XML 変換時の動作を設定するファイルです。JSON-XML 変換を行う場合だけ作成します。

2.8.4　Kafka 受付を追加する

Kafka 受付を新規に追加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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。

   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのビジネスプロセスで右クリックして［ユーザ定義受付（呼出先固定）追加］を選択します。

   受付種別選択ウィザードが表示されます。

4. 受付種別のドロップダウンリストから「Kafka 受付」を選択します。

   ［次へ］ボタンをクリックすると，Kafka 受付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. ［受付名］を指定します。

   Kafka 受付の名称を 1～40 バイトで指定します。

6. ［終了］ボタンをクリックします。

   ［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。ユーザ定義受付定義画面（基本）が表示されます。

7. Kafka 受付の定義情報を設定します。

Kafka 受付の設定項目について次の表に示します。

表 2-30　Kafka 受付のユーザ定義受付定義画面（基本）の設定項目

分類	項目		設定内容	設定
ユーザ定義受付情報	受付名		設定した受付名が表示されます。	◎
	受付 ID		受付 ID が表示されます。必要に応じて変更します。	◎
	デフォルトオペレーション名		「Receive」で固定です。	◎
	オペレーション		「Receive」で固定です。	◎
オペレーション情報	通信モデル		「非同期」を設定します。	◎
要求電文	[any 型を使う] チェックボックス		任意の形式の要求電文（ボディ）を使用する場合はチェックします。	○
	受付	電文フォーマット	[any 型を使う] チェックボックスをチェックしていない場合は，任意の電文フォーマットファイルを指定します。 [any 型を使う] チェックボックスをチェックしている場合は，電文フォーマットファイルを指定する必要はありません。	○
	サービス部品	[使う] チェックボックス	必要に応じてチェックします。	○
応答電文	[any 型を使う] チェックボックス		チェックしません。	－
	受付	電文フォーマット	設定しません。	－
	サービス部品	[使う] チェックボックス	チェックしません。	－

（凡例）

◎：必ず設定します。

○：任意で設定します。

－：該当しません。

8. ユーザ定義受付（詳細）タブをクリックします。

ユーザ定義受付定義画面（詳細）が表示されます。

9. Kafka 受付の定義情報を設定します。

Kafka 受付の設定項目について次の表に示します。

表 2-31　Kafka 受付のユーザ定義受付定義画面（詳細）の設定項目

分類	項目	設定内容	設定
ユーザ定義受付制御情報	独自定義ファイル	次の独自定義ファイルが表示されます。必要に応じて変更します。 • cscurecpkafka.properties※1	◎

分類	項目	設定内容	設定
ユーザ定義受付制御情報	独自定義ファイル	JSON-XML 変換を行う場合は，必要に応じて JSON-XML 変換定義ファイル（csc_json_converter.properties※2）を指定します。	◎
	EAR ファイル	次の EAR ファイルが設定されていることを確認します。 • cscmsg_urecp_kafka.ear※3	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※1

設定されている cscurecpkafka.properties はテンプレートファイルです。必要に応じて cscurecpkafka.properties を選択して［編集］ボタンをクリックし，独自定義ファイルの内容を修正してください。cscurecpkafka.properties の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.14.1　Kafka 受付定義ファイル」を参照してください。

注※2

JSON-XML 変換定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.18.1　JSON-XML 変換定義ファイル」を参照してください。

注※3

ファイルの格納先ディレクトリは「<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¥custom-reception¥kafka¥lib」です。

2.8.5　アクティビティを設定する

Kafka 受付を定義したあとに，ビジネスプロセス定義画面で受付アクティビティを設定します。

アクティビティの設定では，要求電文用の変数を定義したあと，受付アクティビティに定義した変数を割り当てます。

(1)　変数の定義

［変数・相関セット一覧］ダイアログで次の変数を定義します。

- 要求電文（ヘッダ）の変数
- 要求電文（ボディ）の変数

変数の定義手順について説明します。

(a)　要求電文（ヘッダ）の変数

要求電文（ヘッダ）の変数の定義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. ［変数名］に任意の名称を指定します。

3. [種別] ドロップダウンリストから「XML」を選択します。

4. [電文フォーマット] の［参照］ボタンをクリックします。
   電文フォーマットを指定するダイアログが表示されます。

5. 次のファイルを指定します。
   <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¥custom-reception¥kafka¥schema¥urecp_kafka_header_request.xsd

6. [変数・相関セット一覧] ダイアログで［追加］ボタンをクリックします。
   変数が追加されます。

(b) 要求電文（ボディ）の変数

要求電文（ボディ）の変数の定義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. [変数名] に任意の名称を指定します。

3. [種別] ドロップダウンリストから「XML」を選択します。

4. [電文フォーマット] の［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. [受付名] をチェックし，ドロップダウンリストからユーザ定義受付定義画面で設定した Kafka 受付の名称を選択します。

6. [オペレーション名] ドロップダウンリストから Kafka 受付に設定したオペレーションを選択します。

7. 変数に応じて［電文種別］ドロップダウンリストで［要求電文（ボディ）］を選択します。

8. [電文フォーマット] に任意の名称を入力します。

9. [OK] ボタンをクリックします。

10. [変数・相関セット一覧] ダイアログで［追加］ボタンをクリックします。
    変数が追加されます。

(2) 変数の割り当て

受付アクティビティに追加した変数を割り当てる手順を次に示します。

1. ビジネスプロセス定義画面のキャンバス上の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. [アクティビティ名] に任意の名称を入力します。

3. [オペレーション名] の［取込］ボタンをクリックします。
［オペレーション名の取込］ダイアログが表示されます。

4. [受付名] に該当する Kafka 受付の名称を設定し，［オペレーション名］に取り込み対象となるオペレーション名を設定します。

5. [OK] ボタンをクリックします。

6. [ボディ割当変数] ドロップダウンリストから，定義した要求電文（ボディ）の変数を選択します。

7. [ヘッダ割当変数] の［設定］ボタンをクリックします。
［ヘッダ割当変数］ダイアログが表示されます。

8. [追加] ボタンをクリックします。

9. [割当変数] セルのドロップダウンリストから，定義した要求電文（ヘッダ）の変数を選択します。

10. [ルート要素] セルのドロップダウンリストからルート要素を選択します。

11. [OK] ボタンをクリックします。
［ヘッダ割当変数］ダイアログが閉じます。

12. [受付アクティビティ］ダイアログで［OK］ボタンをクリックします。

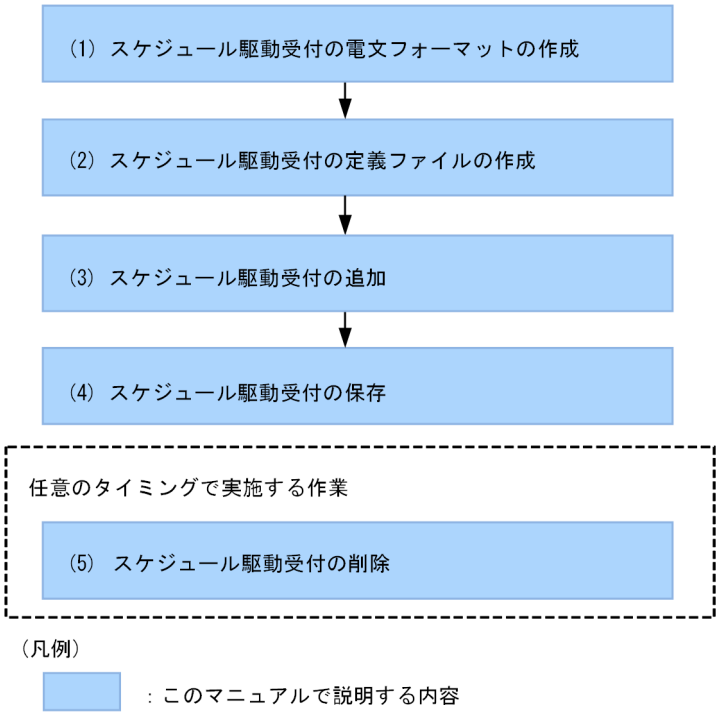
2.9　スケジュール駆動受付の定義

この節では，開発環境でのスケジュール駆動受付の定義方法について説明します。

2.9.1　スケジュール駆動受付の定義の流れ

スケジュール駆動受付の流れを次に示します。

図 2-10　スケジュール駆動受付の定義の流れ



スケジュール駆動受付を定義する場合の作業について次に説明します。

(1)　スケジュール駆動受付の電文フォーマットの作成

スケジュール駆動受付で使用する要求電文フォーマットを作成します。

電文フォーマットの作成方法については，「2.9.2　スケジュール駆動受付の電文フォーマットを作成する」を参照してください。

(2)　スケジュール駆動受付の定義ファイルの作成

スケジュール駆動受付で使用する定義ファイルを作成します。スケジュール駆動受付で使用する定義ファイルの作成方法については，「2.9.3　スケジュール駆動受付の定義ファイルを作成する」を参照してください。

(3)　スケジュール駆動受付の追加

実行環境でスケジュール駆動受付を追加します。追加，および定義情報の設定方法については，「2.9.4 スケジュール駆動受付を追加する」を参照してください。

(4)　スケジュール駆動受付の保存

追加および定義済みのスケジュール駆動受付をリポジトリに保存します。スケジュール駆動受付の保存手順については，「2.13　ユーザ定義受付の保存」を参照してください。

(5)　スケジュール駆動受付の削除

不要になったスケジュール駆動受付は，必要に応じて削除できます。

スケジュール駆動受付の削除については，「2.16　ユーザ定義受付の削除」を参照してください。

2.9.2　スケジュール駆動受付の電文フォーマットを作成する

スケジュール駆動受付では要求電文フォーマットを使用します。応答電文フォーマットの設定は不要です。

スケジュール駆動受付の要求電文フォーマットは，サービスプラットフォームが提供するスキーマを使用します。そのため，電文フォーマットの作成は不要です。

(1)　要求電文の電文フォーマット

スケジュール駆動受付の要求電文フォーマットの形式について説明します。

名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/schedule/request」です。

表 2-32　スケジュール駆動受付の要求電文フォーマット

タグ名		種別	出現回数	説明
<request>		−	1 回	−
	<request-id>	string	0 または 1 回	スケジュール駆動受付が作成したリクエスト ID が設定されます。スケジュール駆動受付定義ファイルの「scdrecp.create.work-dir」に true を設定した場合だけ出現します。
	<cscCorrelationID>	string	1 回	スケジュール駆動受付が作成したクライアント相関 ID が設定されます。
	<receptionName>	string	1 回	スケジュール駆動受付の受付名が設定されます。
	<receptionID>	string	1 回	スケジュール駆動受付の受付 ID が設定されます。

タグ名		種別	出現回数	説明
	<eventOccurrenceTime>	string	1 回	イベントの発生時刻がローカル時刻で設定されます。 「yyyyMMdd_HHmmssSSS」の形式で設定します。

（凡例）

－：該当する項目はありません。

2.9.3 スケジュール駆動受付の定義ファイルを作成する

スケジュール駆動受付では，次の定義ファイルを作成します。

- スケジュール駆動受付定義ファイル

  スケジュール駆動受付の動作を定義するファイルです。スケジュール駆動受付のデプロイ時，スケジュール駆動受付の起動時，および csccmctl コマンドの実行時に読み込まれます。

  スケジュール駆動受付定義ファイルは，スケジュール駆動受付の追加時に，サービスプラットフォームが提供するテンプレートファイル（cscurecpschedule.properties）を編集して利用します。そのため，スケジュール駆動受付定義ファイルの作成は不要です。

- スケジュール駆動受付コンフィグファイル

  スケジュール駆動受付のコンフィグファイルです。スケジュール駆動受付のセットアップ後に，スケジュール駆動受付に関する構成情報の変更に使用します。変更頻度が高い情報を変更できます。スケジュール駆動受付の開始時に読み込まれます。

- スケジュール駆動受付共通コンフィグファイル

  すべてのスケジュール駆動受付で共通して定義できるスケジュール駆動受付コンフィグファイルです。

2.9.4 スケジュール駆動受付を追加する

スケジュール駆動受付を新規に追加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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。

   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのビジネスプロセスで右クリックして［ユーザ定義受付（呼出先固定）追加］を選択します。

   受付種別選択ウィザードが表示されます。

4. 受付種別のドロップダウンリストから「スケジュール駆動受付」を選択します。

［次へ］ボタンをクリックすると，スケジュール駆動受付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. ［受付名］を指定します。

スケジュール駆動受付の名称を 1～40 バイトで指定します。

6. ［終了］ボタンをクリックします。

［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。ユーザ定義受付定義画面（基本）が表示されます。

7. スケジュール駆動受付の定義情報を設定します。

スケジュール駆動受付の設定項目について次の表に示します。

表 2-33　スケジュール駆動受付のユーザ定義受付定義画面（基本）の設定項目

分類	項目		設定内容	設定
ユーザ定義受付情報	受付名		設定した受付名が表示されます。	◎
	受付 ID		受付 ID が表示されます。必要に応じて変更します。	◎
	デフォルトオペレーション名		「Schedule」で固定です。	◎
	オペレーション		「Schedule」で固定です。	◎
オペレーション情報	通信モデル		「非同期」を設定します。	◎
要求電文	［any 型を使う］チェックボックス		チェックしません。	－
	受付	電文フォーマット	urecp_schedule_request.xsd を指定します。	◎
	サービス部品	［使う］チェックボックス	必要に応じてチェックします。	○
応答電文	［any 型を使う］チェックボックス		チェックしません。	－
	受付	電文フォーマット	設定しません。	－
	サービス部品	［使う］チェックボックス	チェックしません。	－

（凡例）

◎：必ず設定します。

○：任意で設定します。

－：該当しません。

8. ユーザ定義受付（詳細）タブをクリックします。

ユーザ定義受付定義画面（詳細）が表示されます。

9. スケジュール駆動受付の定義情報を設定します。

スケジュール駆動受付の制御情報の設定項目について次の表に示します。

表 2-34　スケジュール駆動受付のユーザ定義受付定義画面（詳細）の設定項目

分類	項目	設定内容	設定
ユーザ定義受付制御情報	独自定義ファイル	次の独自定義ファイルが表示されます。必要に応じて変更します。 • cscurecpschedule.properties※1	◎
	EAR ファイル	次の EAR ファイルが設定されていることを確認します。 • cscmsg_urecp_schedule.ear※2	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※1

設定されている cscurecpschedule.properties はテンプレートファイルです。必要に応じて cscurecpschedule.properties を選択して［編集］ボタンをクリックし，独自定義ファイルの内容を修正してください。cscurecpschedule.properties の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.15.1　スケジュール駆動受付定義ファイル」を参照してください。

注※2

ファイルの格納先ディレクトリは「<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¥custom-reception¥schedule¥lib」です。

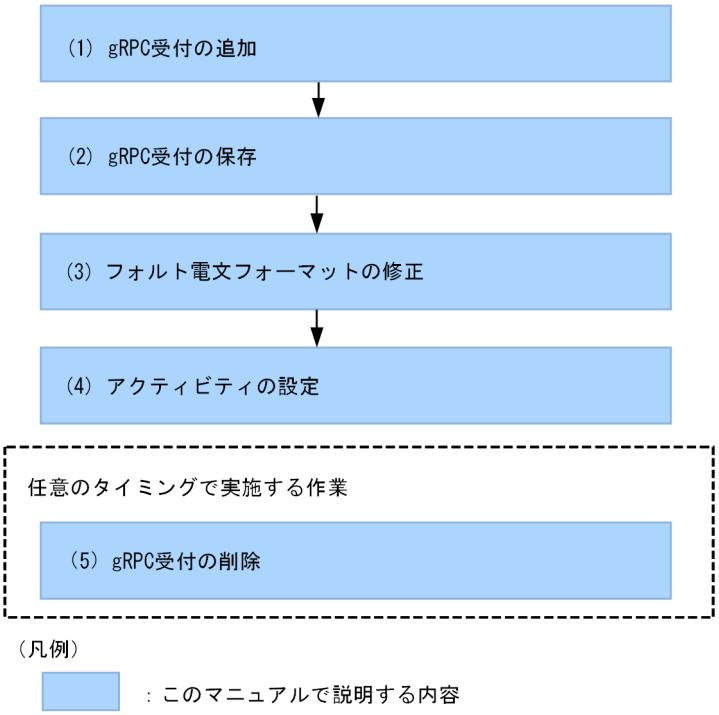
2.10　gRPC 受付の定義

この節では，開発環境での gRPC 受付の定義方法について説明します。

2.10.1　gRPC 受付の定義の流れ

gRPC 受付の流れを次に示します。

図 2-11　gRPC 受付の定義の流れ



gRPC 受付を定義する場合の作業について次に説明します。

(1)　gRPC 受付の追加

実行環境で gRPC 受付を追加します。追加方法については，「2.10.2　gRPC 受付を追加する」を参照してください。

(2)　gRPC 受付の保存

追加および定義済みの gRPC 受付をリポジトリに保存します。gRPC 受付の保存手順については，「2.13 ユーザ定義受付の保存」を参照してください。

(3)　フォルト電文フォーマットの修正

gRPC 受付を作成したあと，必要に応じてフォルト電文フォーマットの XML スキーマファイルを修正し，再登録できます。

フォルト電文フォーマットの修正方法については，「2.10.3　フォルト電文フォーマットを修正する」を参照してください。

(4)　アクティビティの設定

gRPC 受付の要求電文／応答電文フォーマット用のボディ変数とヘッダ変数をビジネスプロセスのアクティビティに設定します。

アクティビティの設定方法については，「2.10.4　アクティビティを設定する」を参照してください。

(5)　gRPC 受付の削除

不要になった gRPC 受付は，必要に応じて削除できます。

gRPC 受付の削除については，「2.16　ユーザ定義受付の削除」を参照してください。

2.10.2　gRPC 受付を追加する

gRPC 受付を新規に追加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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのビジネスプロセスで右クリックして［ユーザ定義受付（呼出先固定）追加］を選択します。
   受付種別選択ウィザードが表示されます。

4. 受付種別のドロップダウンリストから「gRPC 受付」を選択します。
   ［次へ］ボタンをクリックすると，gRPC 受付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. 受付名，proto_path，proto ファイルを入力します。
   受付名には，任意の受付名を入力します。
   proto_path には，proto ファイルが格納されたフォルダのパスを指定します。
   proto ファイルには，proto ファイル名を絶対パスで指定します。
   proto ファイルを設定する際の注意事項については，マニュアル「サービスプラットフォーム 解説」の「2.13.1(3)(c)　proto ファイル使用時の注意事項」を参照してください。

6. ［次へ］ボタンをクリックします。

7. gRPC Service 名，RPC メソッド名を選択します。

8. [終了] ボタンをクリックします。

[終了] ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。ユーザ定義受付定義画面（基本）が表示されます。

9. gRPC 受付の定義情報を設定します。

gRPC 受付の設定項目について次の表に示します。

表 2-35　gRPC 受付のユーザ定義受付定義画面（基本）の設定項目

分類	項目		設定内容	設定
ユーザ定義受付情報	受付名		設定した受付名が表示されます。必要に応じて変更します。	◎
	受付 ID		受付 ID が表示されます。必要に応じて変更します。	◎
要求電文	ボディ	電文フォーマット	cscformatN.xsd が指定されています。	－
応答電文	ボディ	電文フォーマット	cscformatN.xsd が指定されています。	－
フォルト電文	フォルト名		フォルト名が表示されます。	－
	電文フォーマット		urecp_grpc_fault.xsd が指定されています。	○

（凡例）

◎：必ず設定します。

○：任意で設定します。

－：該当しません。

10. ユーザ定義受付（詳細）タブをクリックします。

ユーザ定義受付定義画面（詳細）が表示されます。

11. gRPC 受付の定義情報を設定します。

gRPC 受付の設定項目について次の表に示します。

表 2-36　gRPC 受付のユーザ定義受付定義画面（詳細）の設定項目

分類	項目	設定内容	設定
ユーザ定義受付制御情報	独自定義ファイル	次の独自定義ファイルが表示されます。必要に応じて変更します。 • cscurecpgrpc.properties※1	◎
	proto ファイル	次の zip ファイルが設定されていることを確認します。 • proto.zip※2	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※1

設定されている cscurecpgrpc.properties はテンプレートファイルです。必要に応じて cscurecpgrpc.properties を選択して［編集］ボタンをクリックし，独自定義ファイルの内容を修正してください。cscurecpgrpc.properties の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.17.1　gRPC 受付定義ファイル」を参照してください。

注※2

proto ファイルの詳細については，マニュアル「サービスプラットフォーム 解説」の「2.13.1(3)(b)　proto ファイルの出力」を参照してください。

2.10.3　フォルト電文フォーマットを修正する

gRPC 受付では，gRPC のメッセージ内容※に沿ったフォルト電文フォーマットを提供しています。このフォルト電文フォーマット（XML スキーマファイル）は，gRPC 受付を作成したときに，ユーザ定義受付定義画面（基本）のフォルト電文フォーマットに自動的に設定されます。

注※

gRPC を利用した通信では，google.rpc.Status のメッセージの処理方式によってエラー処理を行っています。google.rpc.Status のメッセージ内容，およびエラー処理については，Google 社の google.rpc.Status に関するサイトを参照してください。

gRPC 受付を作成したあと，必要に応じてフォルト電文フォーマットの XML スキーマファイルを修正し，再登録できます。再登録することで，XML スキーマファイルの記述を簡略化できます。

(1)　フォルト電文フォーマットを修正する手順

XML スキーマファイルを修正する手順を次に示します。

1. 次の XML スキーマファイルを任意の場所にコピーします。
   <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¥custom-reception¥gRPC¥schema¥urecp_grpc_fault.xsd
2. コピーした XML スキーマファイルをテキストエディタで開きます。
3. XML スキーマファイルに定義されている，/fault/status/details 要素以下に定義されている次の要素のうち，使用しない要素を削除します。
   - retry-info
   - debug-info
   - quota-failure
   - error-info
   - precondition-failure
   - bad-request
   - request-info
   - resource-info
   - help
   - localized-message

- additional-message

処理に必要な要素を削除した場合の動作は保証されません。

これらの要素は，Google 社が提供する google/rpc/error_details.proto の内容に対応しています。google/rpc/error_details.proto の内容については，Google 社の google/rpc/error_details.proto に関するサイトを参照してください。

なお，additional-message は，google/rpc/error_details.proto に定義されていない任意の型のメッセージを格納する場合に使用します。

4. **使用しない要素（子要素を含む）を削除した XML スキーマファイルを保存します。**

5. **修正した XML スキーマファイルを，ユーザ定義受付定義画面（基本）の［フォルト電文］に設定します。**

修正した XML スキーマファイル以外を設定した場合，動作は保証されません。

(2) フォルト電文フォーマットの形式

gRPC 受付のフォルト電文フォーマットの形式について説明します。

名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/grpc/fault」です。

表 2-37 gRPC 受付のフォルト電文フォーマット

タグ名	種別	出現回数	説明
<fault>	−	1 回	−
<status>	−	1 回	ステータス情報を指定します。
<code>	文字列	1 回	ステータスコード※を指定します。
<message>	文字列	0～1 回	開発者向けのエラーメッセージを指定します。 設定値は英字で設定してください。英字以外の値を指定した場合の動作は保証されません。
<details>	−	0～1 回	詳細エラーのメッセージのリストを指定します。子要素として「error_details.proto」のメッセージ構造体に応じた要素が出現します。
<retry-info>	−	0 回以上	メッセージ構造体「RetryInfo」の定義内容を指定します。
<retry-delay>	文字列	0～1 回	引数「retry_delay」が設定されます。
<seconds>	文字列	0～1 回	引数「retry_delay」の秒を指定します。 設定値は「-315,576,000,000～315,576,000,000」の範囲で指定してください。
<nanos>	文字列	0～1 回	引数「retry_delay」のナノ秒を指定します。設定値は「-999,999,999～999,999,999」の範囲で指定してください。
<debug-info>	−	0 回以上	メッセージ構造体「DebugInfo」の定義内容を指定します。

タグ名	種別	出現回数	説明
<stack-entries>	文字列	0 回以上	引数「stack_entries」が設定されます。
<detail>	文字列	0～1 回	引数「detail」が設定されます。
<quota-failure>	－	0 回以上	メッセージ構造体「QuotaFailure」の定義内容を指定します。
<violation>	－	0 回以上	メッセージ構造体「Violation」の定義内容を指定します。
<subject>	文字列	0～1 回	引数「subject」が設定されます。
<description>	文字列	0～1 回	引数「description」が設定されます。
<error-info>	－	0 回以上	メッセージ構造体「ErrorInfo」の定義内容を指定します。
<reason>	文字列	0～1 回	引数「reason」を指定します。 設定値は，63 文字以内の「A-Z」「0-9」「_」の範囲の文字列を指定してください。範囲外の値を指定した場合の動作は保証されません。
<domain>	文字列	0～1 回	引数「domain」が設定されます。
<metadata>	－	0 回以上	引数「metadata」が設定されます。
<key>	文字列	1 回	メタデータのキーを指定します。キーの省略（空文字含む）はできません。 設定値は，64 文字以内の「a-z」「A-Z」「0-9」「-」「_」の範囲の文字列を指定してください。範囲外の値を指定した場合の動作は保証されません。
<value>	文字列	1 回	メタデータの値を指定します。
<precondition-failure>	－	0 回以上	メッセージ構造体「PreconditionFailure」の定義内容を指定します。
<violation>	－	0 回以上	メッセージ構造体「Violation」の定義内容を指定します。
<type>	文字列	0～1 回	引数「type」が設定されます。
<subject>	文字列	0～1 回	引数「subject」が設定されます。
<description>	文字列	0～1 回	引数「description」が設定されます。
<bad-request>	－	0 回以上	メッセージ構造体「BadRequest」の定義内容を指定します。
<field-violation>	－	0 回以上	メッセージ構造体「FieldViolation」の定義内容を指定します。
<field>	文字列	0～1 回	引数「field」が設定されます。
<description>	文字列	0～1 回	引数「description」が設定されます。

タグ名	種別	出現回数	説明
<request-info>	－	0 回以上	メッセージ構造体「RequestInfo」の定義内容を指定します。
<request-id>	文字列	0～1 回	引数「request_id」が設定されます。
<serving-data>	文字列	0～1 回	引数「serving_data」が設定されます。
<resource-info>	－	0 回以上	メッセージ構造体「ResourceInfo」の定義内容を指定します。
<resource-type>	文字列	0～1 回	引数「resource_type」が設定されます。
<resource-name>	文字列	0～1 回	引数「resource_name」が設定されます。
<owner>	文字列	0～1 回	引数「owner」が設定されます。
<description>	文字列	0～1 回	引数「description」が設定されます。
<help>	－	0 回以上	メッセージ構造体「Help」の定義内容を指定します。
<link>	－	0 回以上	メッセージ構造体「Link」の定義内容を指定します。
<description>	文字列	0～1 回	引数「description」が設定されます。
<url>	文字列	0～1 回	引数「url」が設定されます。
<localized-message>	－	0 回以上	メッセージ構造体「LocalizedMessage」の定義内容を指定します。
<locale>	文字列	0～1 回	引数「locale」を指定します。 設定値は，IETF の BCP47 に準拠してください。IETF の BCP47 に準拠していない場合の動作は保証されません。
<message>	文字列	0～1 回	引数「message」が設定されます。
<additional-message>	－	0 回以上	「error_details.proto」のメッセージ構造体に定義されていないメッセージの定義内容を指定します。Any 型のメッセージとして扱われます。
<type-url>	文字列	1 回	メッセージのタイプ（URL/resource name）を指定します。
<value>	文字列	1 回	メッセージの値を Base64 エンコードしたデータを指定します。
<metadata>	－	0 回以上	gRPC クライアントに送信するメタデータを指定します。
<key>	文字列	1 回	gRPC クライアントに送信するメタデータのキーを指定します。キーの省略（空文字含む）はできません。 バイナリ形式で送信する場合，この要素値は「-bin」で終了してください。

タグ名			種別	出現回数	説明
		<value>	文字列	1 回	メタデータのキーに対する値を指定します。 テキスト形式の場合，0x20 および 0x21-0x7E の範囲の文字列が指定できます。ただし，空白（0x20）を値の最初または最後に置くことはできません。 指定できない値を指定した場合の動作は保証されません。 バイナリ形式の場合，Base64 でエンコードした値を設定します。

（凡例）

－：該当しません。

注※

フォルト電文に設定できるステータスコードを次に示します。次に示すステータスコード以外の値が設定された場合，リクエストがエラーとなります。

・CANCELLED
・UNKNOWN
・INVALID_ARGUMENT
・DEADLINE_EXCEEDED
・NOT_FOUND
・ALREADY_EXISTS
・PERMISSION_DENIED
・RESOURCE_EXHAUSTED
・FAILED_PRECONDITION
・ABORTED
・OUT_OF_RANGE
・UNIMPLEMENTED
・INTERNAL
・UNAVAILABLE
・DATA_LOSS
・UNAUTHENTICATED

2.10.4　アクティビティを設定する

gRPC 受付を定義したあとに，ビジネスプロセス定義画面で受付アクティビティおよび応答アクティビティを設定する方法について説明します。

アクティビティの設定では，要求電文，応答電文，およびフォルト電文の変数を定義したあと，受付アクティビティおよび応答アクティビティに定義した変数を割り当てます。

(1)　変数の定義

［変数・相関セット一覧］ダイアログで次の変数を定義します。

- 要求電文（ヘッダ）の変数
- 応答電文（ヘッダ）の変数
- 要求電文（ボディ）の変数
- 応答電文（ボディ）の変数
- フォルト電文の変数

変数の定義手順について説明します。

(a) 要求電文（ヘッダ）および応答電文（ヘッダ）の変数

要求電文（ヘッダ）および応答電文（ヘッダ）の変数の定義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. ［変数名］に任意の名称を指定します。
3. ［種別］ドロップダウンリストから「XML」を選択します。
4. ［電文フォーマット］の［参照］ボタンをクリックします。
   電文フォーマットを指定するダイアログが表示されます。
5. 変数に応じて次のファイルを指定します。
   - 要求電文（ヘッダ）の変数
     <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¥custom-reception¥gRPC¥schema¥urecp_grpc_header_metadata_request.xsd
   - 応答電文（ヘッダ）の変数
     <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¥custom-reception¥gRPC¥schema¥urecp_grpc_header_metadata_response.xsd
6. ［変数・相関セット一覧］ダイアログで［追加］ボタンをクリックします。
   変数が追加されます。

(b) 要求電文（ボディ），応答電文（ボディ），およびフォルト電文の変数

要求電文（ボディ），応答電文（ボディ），およびフォルト電文の変数の定義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. ［変数名］に任意の名称を指定します。
3. ［種別］ドロップダウンリストから「XML」を選択します。

4. [電文フォーマット] の [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. [受付名] をチェックし，ドロップダウンリストからユーザ定義受付定義画面で設定した gRPC 受付の名称を選択します。

6. [オペレーション名] ドロップダウンリストから gRPC 受付に設定したオペレーションを選択します。

7. 変数に応じて [電文種別] ドロップダウンリストで次のように選択します。
   - 要求電文（ボディ）の変数の場合
     [要求電文（ボディ）] を選択します。
   - 応答電文（ボディ）の変数の場合
     [応答電文（ボディ）] を選択します。
   - フォルト電文の変数の場合
     [フォルト電文] を選択します。

8. [電文フォーマット] に任意の名称を入力します。

9. [OK] ボタンをクリックします。

10. [変数・相関セット一覧] ダイアログで [追加] ボタンをクリックします。
    変数が追加されます。

(2) 変数の割り当て

受付アクティビティおよび応答アクティビティに追加した変数を割り当てる手順を次に示します。

(a) 受付アクティビティの場合

受付アクティビティに，追加した要求電文の変数を割り当てる手順を次に示します。

1. ビジネスプロセス定義画面のキャンバス上の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. [アクティビティ名] に任意の名称を入力します。

3. [オペレーション名] の [取込] ボタンをクリックします。
   [オペレーション名の取込] ダイアログが表示されます。

4. [受付名] に該当する gRPC 受付の名称を設定し，[オペレーション名] に取り込み対象となるオペレーション名を設定します。

5. [OK] ボタンをクリックします。

6. [ボディ割当変数] ドロップダウンリストから，定義した要求電文（ボディ）の変数を選択します。

7. [ヘッダ割当変数] の［設定］ボタンをクリックします。
   ［ヘッダ割当変数］ダイアログが表示されます。

8. [追加] ボタンをクリックします。

9. [割当変数] セルのドロップダウンリストから，定義した要求電文（ヘッダ）の変数を選択します。

10. [ルート要素] セルのドロップダウンリストからルート要素を選択します。

11. [OK] ボタンをクリックします。
    ［ヘッダ割当変数］ダイアログが閉じます。

12. [受付アクティビティ］ダイアログで［OK］ボタンをクリックします。

(b) 応答アクティビティの場合

応答アクティビティに，追加した応答電文の変数を割り当てる手順を次に示します。フォルト応答用の応答アクティビティについては，「(c) フォルト応答用の応答アクティビティの場合」を参照してください。

1. ビジネスプロセス定義画面のキャンバス上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. [アクティビティ名] に任意の名称を入力します。

3. [オペレーション名] の［取込］ボタンをクリックします。
   ［オペレーション名の取込］ダイアログが表示されます。

4. [ユーザ定義受付から取込] ラジオボタンを選択し，[受付名] を選択します。

5. [オペレーション名] に gRPC 受付に設定したオペレーション名を設定します。

6. [OK] ボタンをクリックします。

7. [ボディ割当変数] ドロップダウンリストから，定義した応答電文（ボディ）の変数を選択します。

8. [ヘッダ割当変数] の［設定］ボタンをクリックします。
   ［ヘッダ割当変数］ダイアログが表示されます。

9. [追加] ボタンをクリックします。

10. [割当変数] セルのドロップダウンリストから，定義した応答電文（ヘッダ）の変数を選択します。

11. [ルート要素] セルのドロップダウンリストからルート要素を選択します。

12. [OK] ボタンをクリックします。
    ［ヘッダ割当変数］ダイアログが閉じます。

13. [応答アクティビティ］ダイアログで［OK］ボタンをクリックします。

(c) フォルト応答用の応答アクティビティの場合

フォルト応答用の応答アクティビティに，追加したフォルト電文の変数を割り当てる手順を次に示します。

1. ビジネスプロセス定義画面のキャンバス上の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. ［アクティビティ名］に任意の名称を入力します。
3. ［オペレーション名］の［取込］ボタンをクリックします。
   ［オペレーション名の取込］ダイアログが表示されます。
4. ［受付名］を選択し，［オペレーション名］に gRPC 受付に設定したオペレーション名を設定します。
5. ［OK］ボタンをクリックします。
6. ［ボディ割当変数］ドロップダウンリストから，定義したフォルト電文の変数を選択します。
7. ［フォルト名］に gRPC 受付で定義されたフォルト名を入力します。
8. ［応答アクティビティ］ダイアログで［OK］ボタンをクリックします。

(3) gRPC 受付のヘッダ電文フォーマット

gRPC 受付で使用するヘッダ電文フォーマットの形式について説明します。

(a) 要求電文のヘッダ電文フォーマット

gRPC 受付の要求電文のヘッダ電文フォーマットは，サービスプラットフォームが提供するスキーマを使用します。そのため，ヘッダ電文の定義ファイルの作成は不要です。

サービスプラットフォームが提供する電文フォーマットの XML スキーマは，「<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¥custom-reception¥gRPC¥schema」に格納されています。

gRPC 受付の要求電文フォーマットの形式を次に示します。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/grpc/header/metadata_request」です。

表 2-38 gRPC 受付のヘッダ電文の要求電文フォーマット

タグ名			種別	出現回数	説明
<grpc-metadata-request>			−	1 回	−
	<metadata>		−	0 回以上	gRPC クライアントから送信されたメタデータが設定されます。
		<key>	string	1 回	gRPC クライアントから送信されたメタデータのキーが設定されます。 バイナリ形式の場合，この要素値は「-bin」で終了します。

タグ名			種別	出現回数	説明
		<value>	string	1 回	メタデータのキーに対する値が設定されます。 バイナリ形式の場合，Base64 でエンコードした値を設定します。

（凡例）

－：該当する項目はありません。

(b) 応答電文のヘッダ電文フォーマット

gRPC 受付の応答電文のヘッダ電文フォーマットは，サービスプラットフォームが提供するスキーマを使用します。そのため，ヘッダ電文の定義ファイルの作成は不要です。

サービスプラットフォームが提供する電文フォーマットの XML スキーマは，「<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¥custom-reception¥gRPC¥schema」に格納されています。

gRPC 受付の応答電文フォーマットの形式を次に示します。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/grpc/header/metadata_response」です。

表 2-39 gRPC 受付のヘッダ電文の応答電文フォーマット

タグ名			種別	出現回数	説明
<grpc-metadata-response>			－	1 回	－
	<metadata>		－	0 回以上	gRPC クライアントに送信するメタデータが設定されます。
		<key>	string	1 回	gRPC クライアントに送信するメタデータのキーが設定されます。キーに指定できる値は半角英数字とアンダーバー（_），ハイフン（-），ピリオド（.）です。キーの省略はできません。また，指定値の大文字・小文字は区別しません。 バイナリ形式で送信する場合，この要素値は「-bin（大文字・小文字を区別する）」で終了します。ただし，「-bin」だけを指定することはできません。
		<value>	string	1 回	メタデータのキーに対する値が設定されます。 テキスト形式の場合の，指定できる値は「0x20」，および「0x21-0x7E」の範囲の文字列です。指定できる値以外を指定した場合は，動作は保証されません。空白（0x20）は，値の最初または最後に置くことはできません。 バイナリ形式の場合，Base64 でエンコードした値を設定します。

（凡例）

－：該当する項目はありません。

2.11　WSDL の作成

ユーザ定義受付を利用してサービス部品を呼び出すときのインターフェース情報を定義する WSDL ファイルを作成します。

ここでは，WSDL ファイルの作成方法について説明します。

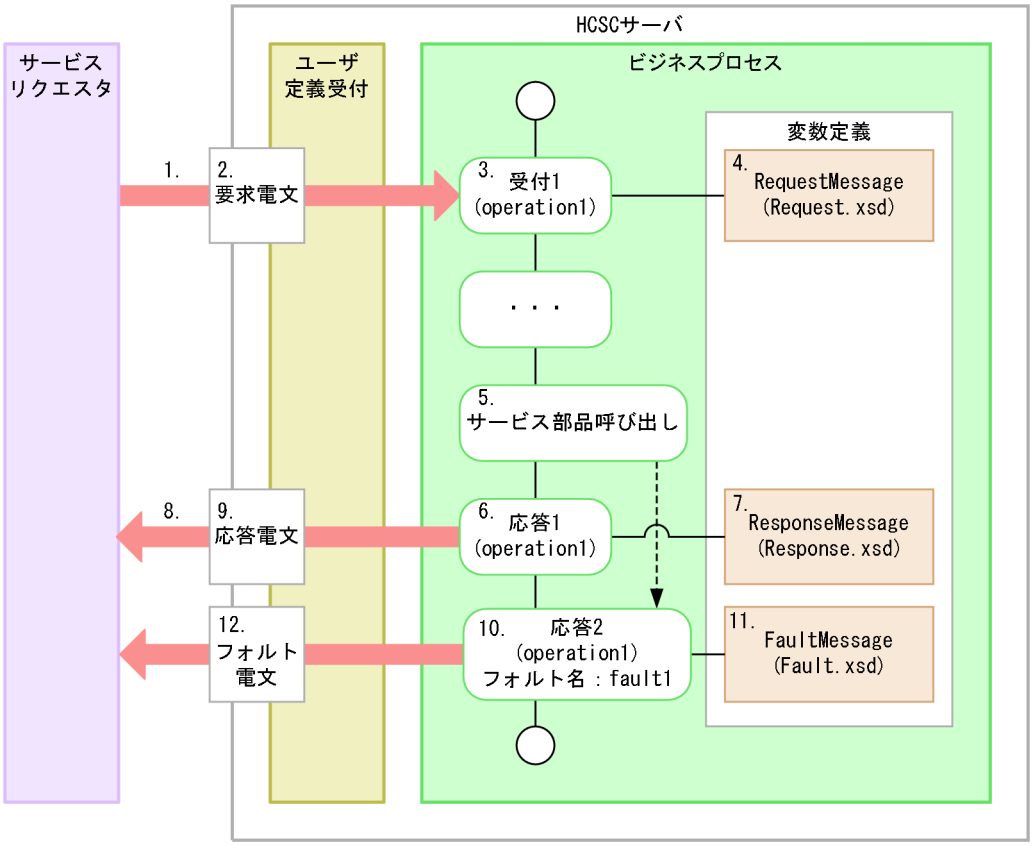
2.11.1　フォルトが発生した場合にフォルト電文を返す例

サービス部品の呼び出しでフォルトが発生したあと，サービスリクエスタにフォルト電文を返す場合を例に，WSDL ファイルの作成方法について説明します。

(1)　ビジネスプロセスの例

WSDL の作成の例として使用するビジネスプロセスを次の図に示します。

図 2-12　WSDL の作成の例として使用するビジネスプロセス



1. サービスリクエスタがビジネスプロセスにサービス部品の実行を要求します。
2. サービス部品の実行要求では，XML の要求電文（ボディ）が送付されます。
3. ビジネスプロセスは XML の要求電文を受付 1 という受付アクティビティで受け取ります。このときのオペレーション名は operation1 です。

4. 受付 1 には変数 RequestMessage を割り当て，要求電文を規定する電文フォーマットとして Request.xsd が指定されています。
5. ビジネスプロセスからサービス部品を呼び出します。フォルトが発生した場合は項番 10.でフォルトが返されます。
6. ビジネスプロセスが，サービス部品の実行結果を XML の応答電文として構成し，応答 1 という応答アクティビティで返します。
7. 応答 1 には変数 ResponseMessage を割り当て，応答電文を規定する電文フォーマットとして Response.xsd が指定されています。
8. サービスリクエスタへの応答が返されます。
9. サービスリクエスタへの応答では，XML の応答電文（ボディ）が送付されます。
10. 項番 5.でフォルトが発生した場合，応答 2 という応答アクティビティから fault1 というフォルトが返されます。
11. 応答 2 には変数 FaultMessage を割り当て，フォルト電文を規定する電文フォーマットとして Fault.xsd が指定されています。
12. サービスリクエスタにフォルト電文が送付されます。

(2) 電文フォーマットの内容の例

「図 2-12 WSDL の作成の例として使用するビジネスプロセス」に示すビジネスプロセスで利用する電文フォーマットの例を次に示します。

Request.xsd

- XML スキーマソース

  文字列の値を持つ単一の要素<Request1>を規定する要求電文用の電文フォーマットです。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.example.org/request"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Request1" type="xsd:string" />
</xsd:schema>
```

- この電文フォーマットの規定に従って送付される XML

```
<Request1>Sample request message</Request1>
```

Response.xsd

- XML スキーマソース

  boolean 値を持つ単一の要素<Response1>を規定する応答電文用の電文フォーマットです。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.example.org/response"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Response1" type="xsd:boolean" />
</xsd:schema>
```

- この電文フォーマットの規定に従って送付される XML

```
<Response1>true</Response1>
```

Fault.xsd

- XML スキーマソース

  <Fault1>という要素を規定し，さらにその子要素として int 値を持つ要素<param1>を規定するフォルト電文用の電文フォーマットです。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.example.org/service"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:svc="http://www.example.org/service">
  <xsd:element name="Fault1" type="svc:SampleData"/>
  <xsd:complexType name="SampleData">
    <xsd:sequence>
      <xsd:element name="param1" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

- この電文フォーマットの規定に従って送付される XML

```
<Fault1>
  <param1>123</param1>
</Fault1>
```

(3)　WSDL の作成例

ここでは，SOAP1.1 を利用する場合を想定して，「(1)　ビジネスプロセスの例」に示すビジネスプロセスに合わせた WSDL の作成例を示します。

WSDL の形式に関する条件や注意事項については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.1　Web サービスを利用したサービス部品の適用範囲」およびこのマニュアルの「2.11.3　SOAP 受付の WSDL 作成時の注意事項」を参照してください。

WSDL の作成例は「Sample_Reception.wsdl」とします。Sample_Reception.wsdl の概要を次に示します。

- 「(1)　ビジネスプロセスの例」に示す内容を前提としています。
- ユーザ定義受付はオペレーション単位に作成することもできますが，Sample_Reception.wsdl は，1 つのユーザ定義受付にすべてのオペレーション含むときの WSDL です。
- 作成する WSDL の通信スタイルは「document」スタイルを推奨しています。そのため，Sample_Reception.wsdl は「document」スタイルの WSDL となっています。

WSDL の作成例「Sample_Reception.wsdl」を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
```

```
targetNamespace="http://www.example.org/service"
xmlns:svc="http://www.example.org/service"
xmlns:req="http://www.example.org/request"
xmlns:res="http://www.example.org/response"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <wsdl:types>
   <xsd:schema targetNamespace="http://www.example.org/request">
     <xsd:element name="Request1" type="xsd:string" />
   </xsd:schema>
   <xsd:schema targetNamespace="http://www.example.org/response">
     <xsd:element name="Response1" type="xsd:boolean" />
   </xsd:schema>
   <xsd:schema targetNamespace="http://www.example.org/service">
     <xsd:element name="Fault1" type="svc:SampleData"/>
     <xsd:complexType name="SampleData">
       <xsd:sequence>
         <xsd:element name="param1" type="xsd:int"/>
       </xsd:sequence>
     </xsd:complexType>
   </xsd:schema>
 </wsdl:types>
 <wsdl:message name="ReqMsg1">
   <wsdl:part element="req:Request1" name="reqParam"/>
 </wsdl:message>
 <wsdl:message name="ResMsg1">
   <wsdl:part element="res:Response1" name="resParam"/>
 </wsdl:message>
 <wsdl:message name="FltMsg1">
   <wsdl:part element="svc:Fault1" name="fltParam"/>
 </wsdl:message>
 <wsdl:portType name="SamplePortType">
   <wsdl:operation name="operation1">
     <wsdl:input message="svc:ReqMsg1"/>
     <wsdl:output message="svc:ResMsg1"/>
     <wsdl:fault message="svc:FltMsg1" name="Fault1"/>
   </wsdl:operation>
 </wsdl:portType>
 <wsdl:binding name="SampleBinding" type="svc:SamplePortType">
   <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
   <wsdl:operation name="operation1">
     <soap:operation soapAction=""/>
     <wsdl:input>
       <soap:body/>
     </wsdl:input>
     <wsdl:output>
       <soap:body/>
     </wsdl:output>
     <wsdl:fault name="Fault1">
       <soap:fault name="Fault1"/>
     </wsdl:fault>
   </wsdl:operation>
 </wsdl:binding>
 <wsdl:service name="SampleService">
   <wsdl:port binding="svc:SampleBinding" name="UserInfo">
     <soap:address location="http://localhost:80/SampleService/services/UserInfo" />
   </wsdl:port>
```

```
  </wsdl:service>
</wsdl:definitions>
```

この作成例「Sample_Reception.wsdl」は，次の作業を行って作成します。

(a)　オペレーションとフォルト名のリストアップ

ビジネスプロセスの受付アクティビティ，および応答アクティビティで定義されている同期のオペレーションをリストアップします。また，フォルト名が定義されている応答アクティビティがある場合は，フォルト名もリストアップします。

作成例ではオペレーションは「operation1」が該当します。また，フォルト名は「Fault1」が該当します。

(b)　割当変数の電文フォーマット定義ファイルのリストアップ

受付アクティビティ，および応答アクティビティが使用している割当変数の電文フォーマット定義ファイルをリストアップします。

作成例では，「Request.xsd」，「Response.xsd」，および「Fault.xsd」が該当します。

注意事項

- リストアップした電文フォーマット定義ファイルの中に，同じ名前空間のスキーマが複数存在する場合，1 つのスキーマに集約する必要があります。

  このとき，次の 2 つの条件を満たす場合，割当変数の電文フォーマット定義ファイルを名前の重複がなくなるよう変更してください。

  ・スキーマの中の要素について名前の重複がある。

  ・その要素の内容が異なる。

  なお，割当変数の電文フォーマット定義ファイルが複数ファイルの場合（外部の XML ファイルを参照している場合），外部の XML ファイルのスキーマも含めて要素の重複がなくなるよう変更してください。

  また，それぞれのスキーマの属性が相反する値を持つ場合，値の変更による影響を考慮してスキーマを集約してください。

- リストアップした電文フォーマット定義ファイルの中に，targetNamespace 属性を持たないスキーマ（カメレオンスキーマ）がある場合，割当変数の電文フォーマット定義ファイルは，targetNamespace 属性を指定したスキーマに変更してください。
- リストアップしたフォルト電文の電文フォーマット定義ファイルのルート要素が複合型でない場合，割当変数の電文フォーマット定義ファイルのルート要素を，複合型になるように変更してください。
- フォルト電文のルート要素の名前空間は，wsdl:definitions 要素の名前空間と一致させる必要があります。

  このため，次の 2 つの条件を満たす場合，その割当変数の電文フォーマットファイルを変更してください。

・フォルト電文が複数存在する。

・ルート要素の名前空間が異なる。

異なるオペレーションのフォルト電文同士の名前空間が異なる場合，オペレーションごとにWSDL ファイルを分けて作成し，ユーザ定義受付を複数作成して対応できます。

(c) WSDL ファイルのひな形の作成（wsdl:definitions 要素の定義）

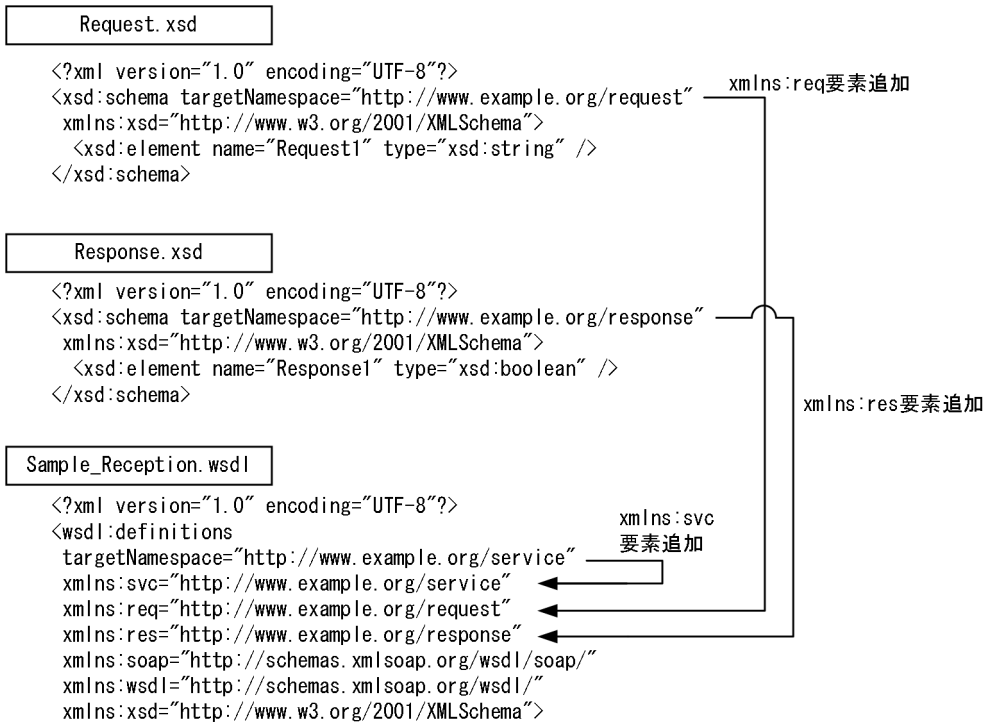
WSDL ファイルのひな形として，XML 宣言および wsdl:definitions 要素を定義します。wsdl:definitions 要素には，次の表に示す属性を追加します。

表 2-40 wsdl:definitions 要素に追加する属性

属性	意味	値	個数
targetNamespace	サービス部品の名前空間 URI	任意の URI を定義できます。ただし，フォルト電文がある場合には，フォルト電文の電文フォーマット定義ファイルの xsd:schema 要素が持つ targetNamespace 属性値と同じ URI を定義します。	1
xmlns:svc	サービス部品のプリフィックス宣言	targetNamespace 属性と同じ値を定義します。	1
xmlns:req	要求電文の電文フォーマットプリフィックス宣言	要求電文の電文フォーマット定義ファイルの xsd:schema 要素が持つ targetNamespace 属性値と同じ URI を定義します。	1
xmlns:res	応答電文の電文フォーマットプリフィックス宣言	応答電文の電文フォーマット定義ファイルの xsd:schema 要素が持つ targetNamespace 属性値と同じ URI を定義します。	1
xmlns:soap	SOAP のプリフィックス宣言	既定の値「http://schemas.xmlsoap.org/wsdl/soap/」を定義します。	1
xmlns:wsdl	WSDL のプリフィックス宣言	既定の値「http://schemas.xmlsoap.org/wsdl/」を定義します。	1
xmlns:xsd	XML スキーマのプリフィックス宣言	既定の値「http://www.w3.org/2001/XMLSchema」を定義します。	1

wsdl:definitions 要素の定義例を次の図に示します。

図 2-13　wsdl:definitions 要素の定義例



複数のオペレーションを使用する場合

作成例（Sample_Reception.wsdl）では，オペレーションを 1 つだけ使用していますが，複数のオペレーションを使用する場合，複数の要求電文，応答電文を扱うことになります。これらの電文が異なる複数の名前空間を使用している場合には，次の表に示すように，それぞれのプリフィックス宣言を wsdl:definitions 要素に定義します。

表 2-41　wsdl:definitions 要素に追加する属性（複数のオペレーションを使用する場合）

属性	意味	値
xmlns:reqN(N>0)	要求電文が複数の名前空間に分かれる場合のプリフィックス宣言	要求電文の電文フォーマット定義ファイルの xsd:schema 要素が持つ targetNamespace 属性値と同じ URI を定義します。 例 http://www.example.org/request1
xmlns:resN(N>0)	応答電文が複数の名前空間に分かれる場合のプリフィックス宣言	応答電文の電文フォーマット定義ファイルの xsd:schema 要素が持つ targetNamespace 属性値と同じ URI を定義します。 例 http://www.example.org/response1

定義した内容は wsdl:message 要素の子要素である wsdl:part 要素の element 属性に定義します。wsdl:message 要素の定義については，「(e)　wsdl:message 要素の定義」を参照してください。

複数のオペレーションを使用する場合の定義例を次の図に示します。

図 2-14　wsdl:definitions 要素の定義例（複数のオペレーションを使用する場合）

Sample_Reception.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
 targetNamespace="http://www.example.org/service"
 xmlns:svc="http://www.example.org/service"
 xmlns:req1="http://www.example.org/request1"
 xmlns:req2="http://www.example.org/request2"
    :
 <wsdl:types>
   <xsd:schema targetNamespace="http://www.example.org/request">
     <xsd:element name="Request1" type="xsd:string" />
   </xsd:schema>
   <xsd:schema targetNamespace="http://www.example.org/response">
     <xsd:element name="Response1" type="xsd:boolean" />
   </xsd:schema>
   <xsd:schema targetNamespace="http://www.example.org/service">
     <xsd:element name="Fault1" type="svc:SampleData"/>
     <xsd:complexType name="SampleData">
       <xsd:sequence>
         <xsd:element name="param1" type="xsd:int"/>
       </xsd:sequence>
     </xsd:complexType>
   </xsd:schema>
 </wsdl:types>

 <wsdl:message name="ReqMsg1">
   <wsdl:part name="reqParam" element="req1:Request1" />
 </wsdl:message>
 <wsdl:message name="ReqMsg2">
   <wsdl:part name="reqParam" element="req2:Request2" />
 </wsdl:message>
```

プレフィックス宣言

プレフィックスを使用

(d)　wsdl:types 要素の定義

wsdl:types 要素とその下位要素，および属性を定義します。

1. 「(c)　WSDL ファイルのひな形の作成（wsdl:definitions 要素の定義）」で定義した wsdl:definitions 要素の子要素として，wsdl:types 要素を定義します。

2. 「(b)　割当変数の電文フォーマット定義ファイルのリストアップ」でリストアップした電文フォーマット定義ファイルの xsd:schema 要素を targetNamespace 属性で分類し，それぞれの targetNamespace の値ごとに 1 つずつ xsd:schema 要素を定義します。

3. 手順 2.で定義した xsd:schema 要素の属性として，電文フォーマット定義ファイルの xsd:schema 要素が持つすべての属性を追加します。

4. 手順 2.で定義した xsd:schema 要素の子要素として，電文フォーマット定義ファイルの xsd:schema 要素が持つすべての子要素を追加します。

5. 複数のフォルト電文があり，名前空間が異なる場合，フォルト電文の名前空間を wsdl:definitions 要素の targetNamespace 属性の値に変更します。

wsdl:types 要素以下に定義する要素と属性を次の表に示します。

表 2-42　wsdl:types 要素以下に定義する要素と属性

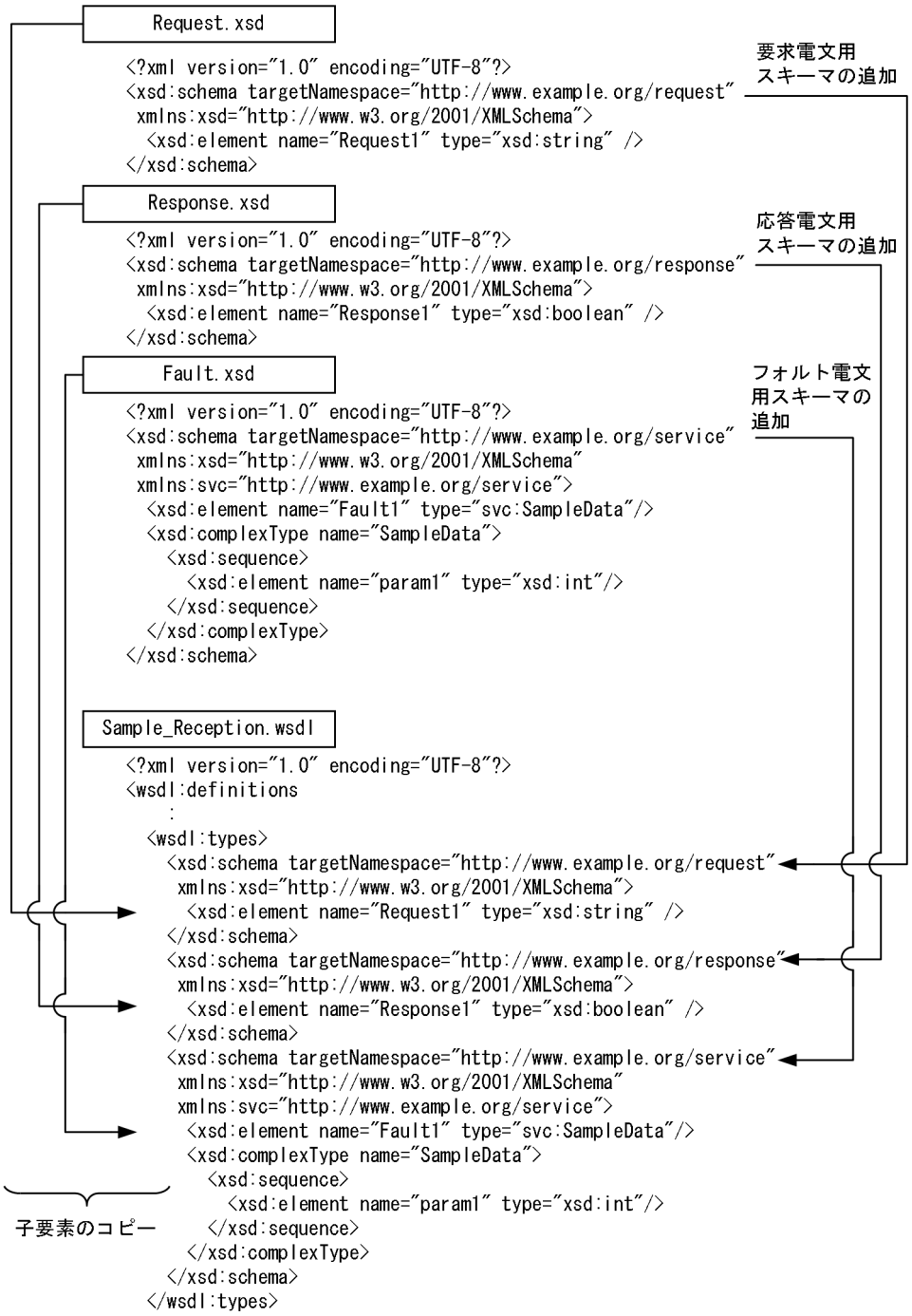
要素			属性	意味	値	個数
wsdl:types			－	型定義要素	－	1
	xsd:schema		－	スキーマ要素	－	1～電文フォーマット定義ファイルが使う名前空間の数
			targetNamespace	名前空間属性	電文の targetNamespace と同じ値を定義します。	1
			xsd:schema の属性	属性	電文の属性と同じ値を定義します。	1～属性の数
		xsd:schema の子要素	－	電文の内容	電文が持つ xsd:schema 要素のすべての子要素をコピーします。	1～同一名前空間を使用する電文フォーマットの xsd:schema 要素の子要素の数

（凡例）

－：該当しません。

wsdl:types 要素の定義例を次の図に示します。

図 2-15　wsdl:types 要素の定義例



(e)　wsdl:message 要素の定義

wsdl:message 要素とその下位要素，および属性を定義します。

1.「(c)　WSDL ファイルのひな形の作成（wsdl:definitions 要素の定義）」で定義した wsdl:definitions 要素の子要素として wsdl:message 要素を定義します。

wsdl:message 要素は WSDL で定義するオペレーションが持つすべての要求電文，応答電文，およびフォルト電文の数だけ定義します。

作成例では，オペレーションが 1 つあります。そのオペレーションには，要求電文が 1 つ，応答電文が 1 つ，フォルト電文が 1 つあるので，wsdl:message 要素を 3 つ定義します。

2. **wsdl:message 要素に name 属性を定義します。**

   name 属性の値は，ファイル内で一意な値になるよう，ほかの wsdl:message 要素の name 属性とは異なる値を定義します。

   作成例では，operation1 の要求電文には ReqMsg1，応答電文には ResMsg1 と定義しています。フォルト電文は個数に合わせて FltMsgN（N は 1 以上）とします。

3. **wsdl:message 要素の子要素として，wsdl:part 要素を 1 つずつ定義します。**

4. **wsdl:part 要素に name 属性と element 属性を定義します。**

   name 属性には任意の文字列を指定します。

   element 属性の値には，「(d)　wsdl:types 要素の定義」でコピーした xsd:schema 要素の子要素から，ルート要素を選択して指定します。「表 2-40　wsdl:definitions 要素に追加する属性」に示すプリフィックスとルート要素名を使用して QName で指定します。

wsdl:message 要素以下に定義する要素と属性を次の表に示します。

表 2-43　wsdl:message 要素以下に定義する要素と属性

要素		属性	意味	値	個数
wsdl:message		－	メッセージ定義要素	－	1
		name	メッセージ名属性	ほかの wsdl:message 要素の name 属性とは異なる一意の文字列を定義します。	1
	wsdl:part	－	メッセージパート定義要素	－	1
		name	メッセージパート名属性	任意の文字列※を定義します。	1
		element	要素参照属性	該当する電文フォーマットファイルのルート要素を定義します。	1

（凡例）

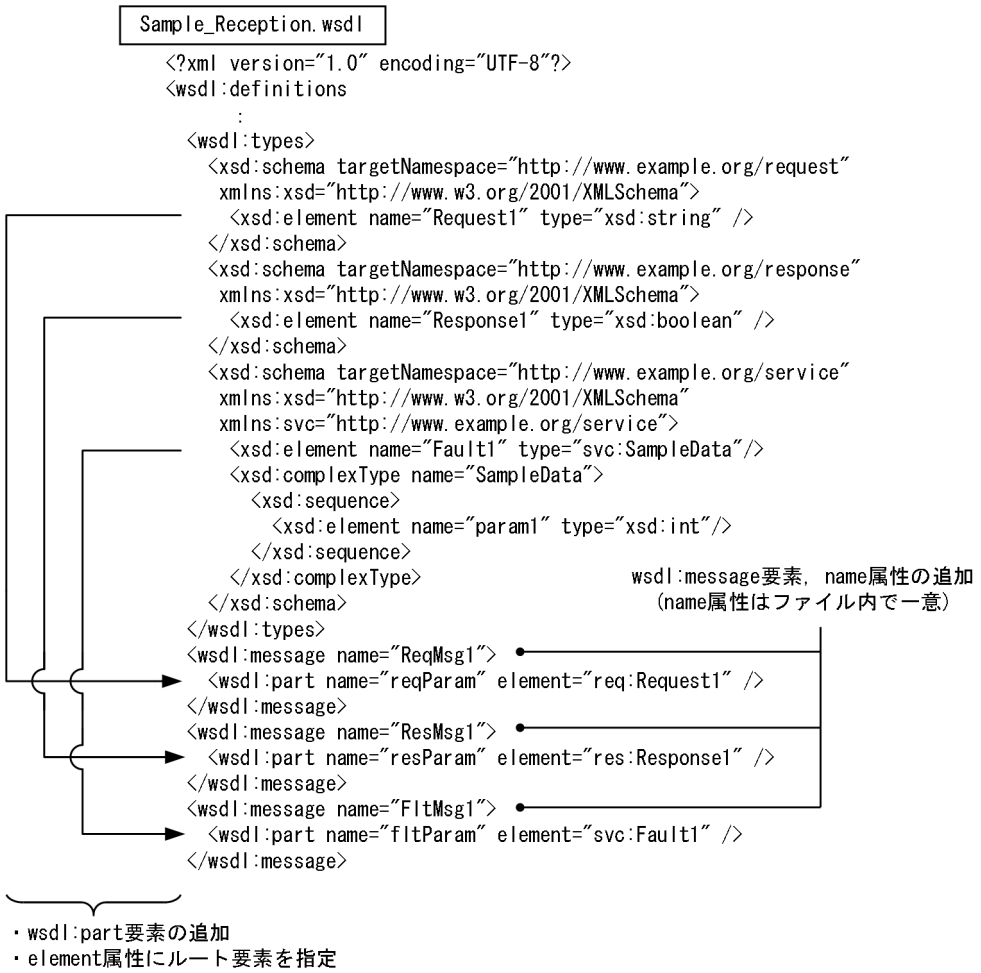
－：該当しません。

注※

使用できる文字種には制限があります。使用できる文字種については，「2.11.3(2)　WSDL 定義で使用できる文字」を参照してください。

wsdl:message 要素の定義例を次の図に示します。

図 2-16　wsdl:message 要素の定義例



(f)　wsdl:portType 要素の定義

wsdl:portType 要素とその下位要素，および属性を定義します。

1. 「(c)　WSDL ファイルのひな形の作成（wsdl:definitions 要素の定義）」で定義した wsdl:definitions 要素の子要素として，wsdl:portType 要素を 1 つ定義します。

2. wsdl:portType 要素に name 属性を定義します。

   name 属性の値には，任意の文字列を指定できます。

3. wsdl:portType 要素の子要素として，wsdl:operation 要素を定義します。

   wsdl:operation 要素は，「(a)　オペレーションとフォルト名のリストアップ」でリストアップしたオペレーションの数だけ定義します。

4. wsdl:operation 要素に name 属性を定義します。

   name 属性の値には，「(a)　オペレーションとフォルト名のリストアップ」でリストアップしたオペレーションのオペレーション名を指定します。

   作成例では，wsdl:operation 要素を 1 つ定義し，name 属性は「operation1」とします。

5. **wsdl:operation 要素の子要素として，wsdl:input 要素，wsdl:output 要素，および wsdl:fault 要素を定義します。**

   wsdl:input 要素，wsdl:output 要素，wsdl:fault 要素の順序で定義します。

   フォルト電文用の応答アクティビティが存在しない場合，wsdl:fault 要素は省略できます。フォルト電文用の応答アクティビティが複数存在する場合，その数だけ定義します。

   作成例では，wsdl:input 要素，wsdl:output 要素，および wsdl:fault 要素を 1 つずつ定義します。

6. **wsdl:input 要素および wsdl:output 要素に message 属性を定義します。**

   message 属性の値には，「(e)　wsdl:message 要素の定義」で定義した wsdl:message 要素の name 属性の値を QName で指定します。

7. **wsdl:fault 要素に name 属性と message 属性を定義します。**

   name 属性の値には，「(a)　オペレーションとフォルト名のリストアップ」でリストアップしたフォルト名を指定します。

   message 属性の値には，「(e)　wsdl:message 要素の定義」で定義した wsdl:message 要素の name 属性の値を QName で指定します。

wsdl:portType 要素以下に定義する要素と属性を次の表に示します。

表 2-44　wsdl:portType 要素以下に定義する要素と属性

要素			属性	意味	値	個数
wsdl:portType			−	ポートタイプ定義要素	−	1
			name	ポートタイプ名属性	任意の文字列※を定義します。	1
	wsdl:operation		−	オペレーション定義要素	−	1～オペレーションの個数
			name	オペレーション名属性	オペレーション名を定義します。	1
		wsdl:input	−	要求メッセージ定義要素	−	1
			message	メッセージ参照属性	「(e)　wsdl:message 要素の定義」で定義した wsdl:message 要素の name 属性値を QName で指定します。	1
		wsdl:output	−	応答メッセージ定義要素	−	1
			message	メッセージ参照属性	「(e)　wsdl:message 要素の定義」で定義した wsdl:message 要素の name 属性値を QName で指定します。	1
		wsdl:fault	−	フォルトメッセージ定義要素	−	1～フォルトの個数
			name	フォルトメッセージ名属性	フォルト名を定義します。	1

要素			属性	意味	値	個数
		wsdl:fault	message	フォルトメッセージ参照属性	「(e)　wsdl:message 要素の定義」で定義した wsdl:message 要素の name 属性値を QName で指定します。	1

（凡例）

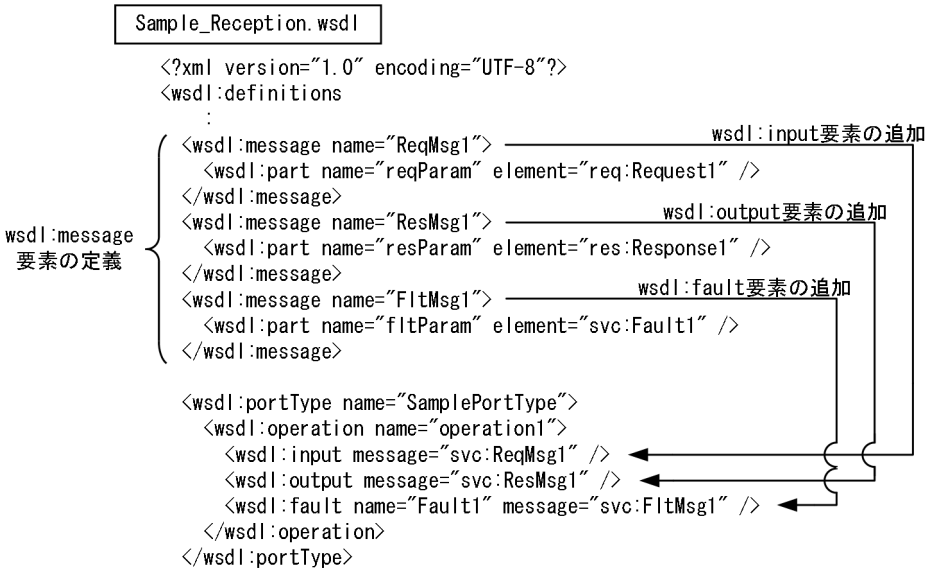
－：該当しません。

注※

使用できる文字種には制限があります。使用できる文字種については，「2.11.3(2)　WSDL 定義で使用できる文字」を参照してください。

wsdl:portType 要素の定義例を次の図に示します。

図 2-17　wsdl:portType 要素の定義例



(g)　wsdl:binding 要素の定義

wsdl:binding 要素とその下位要素，および属性を定義します。

1. 「(c)　WSDL ファイルのひな形の作成（wsdl:definitions 要素の定義）」で定義した wsdl:definitions 要素の子要素として，wsdl:binding 要素を定義します。

2. wsdl:binding 要素に name 属性と type 属性を定義します。

   name 属性の値には，任意の文字列を指定できます。

   type 属性の値には，「(f)　wsdl:portType 要素の定義」で定義した wsdl:portType 要素の name 属性の値を QName で指定します。

3. wsdl:binding 要素の子要素として，soap:binding 要素を定義します。

4. soap:binding 要素に transport 属性を定義します。

   transport 属性の値は固定値です。「http://schemas.xmlsoap.org/soap/http」と定義します。

5. wsdl:binding 要素の子要素として，wsdl:operation 要素を定義します。

   wsdl:operation 要素は，「(a)　オペレーションとフォルト名のリストアップ」でリストアップしたオペレーションの数だけ定義します。

6. wsdl:operation 要素に name 属性を定義します。

   name 属性の値には，「(f)　wsdl:portType 要素の定義」で定義した wsdl:portType 要素の子要素である wsdl:operation 要素の name 属性の値と同じ値を指定します。

7. wsdl:operation 要素の子要素として，soap:operation 要素を定義します。

8. soap:operation 要素に soapAction 属性を指定します。

   soapAction 属性の値は空にします。

9. wsdl:operation 要素の子要素として，wsdl:input 要素，wsdl:output 要素，および wsdl:fault 要素を定義します。

   wsdl:input 要素，wsdl:output 要素，wsdl:fault 要素の順序で定義します。

   フォルト電文用の応答アクティビティが存在しない場合，wsdl:fault 要素は省略できます。フォルト電文用の応答アクティビティが複数存在する場合，その数だけ定義します。

   作成例では，wsdl:input 要素，wsdl:output 要素，および wsdl:fault 要素を 1 つずつ定義します。

10. wsdl:input 要素と wsdl:output 要素の子要素として，soap:body 要素を定義します。

    soap:body 要素には，値は指定しません。

11. wsdl:fault 要素に name 属性を定義します。

    name 属性の値には，「(f)　wsdl:portType 要素の定義」で定義した wsdl:operation 要素の子要素である wsdl:fault 要素の name 属性の値と同じ値を指定します。

12. wsdl:fault 要素の子要素として，soap:fault 要素を定義します。

13. soap:fault 要素に name 属性を定義します。

    name 属性の値には，親要素である wsdl:fault 要素の name 属性と同じ値を指定します。

wsdl:binding 要素以下に定義する要素と属性を次の表に示します。

表 2-45　wsdl:binding 要素以下に定義する要素と属性

要素	属性	意味	値	個数
wsdl:binding	－	バインディング定義要素	－	1
	name	バインディング名属性	任意の文字列※を定義します。	1
	type	ポートタイプ参照属性	「(f)　wsdl:portType 要素の定義」で定義した wsdl:portType の name 属性値を QName で指定します。	1

要素	属性	意味	値	個数
soap:binding	−	SOAP バインディング定義要素	−	1
	transport	SOAP バインディング伝送形態定義属性	「http://schemas.xmlsoap.org/soap/http」を定義します（固定値）。	1
wsdl:operation	−	オペレーション定義要素	−	1～オペレーションの個数
	name	オペレーション名属性	オペレーション名を定義します。「(f)　wsdl:portType 要素の定義」で定義した wsdl:operation 要素の子要素である wsdl:fault 要素の name 属性の値と同じ値を指定します。	1
soap:operation	−	SOAP バインディングオペレーション定義要素	−	1
	soapAction	SOAPAction ヘッダ定義属性	−	1
wsdl:input	−	要求メッセージ定義要素	−	1
soap:body	−	SOAP バインディング Body 要素定義要素	−	1
wsdl:output	−	応答メッセージ定義要素	−	1
soap:body	−	SOAP バインディング Body 要素定義要素	−	1
wsdl:fault	−	フォルトメッセージ定義要素	−	1～フォルトの個数
	name	フォルトメッセージ名属性	フォルト名を定義します。	1
soap:fault	−	SOAP バインディング Fault 要素定義要素	−	1
	name	SOAP バインディング名前属性	フォルト名を定義します。	1

（凡例）

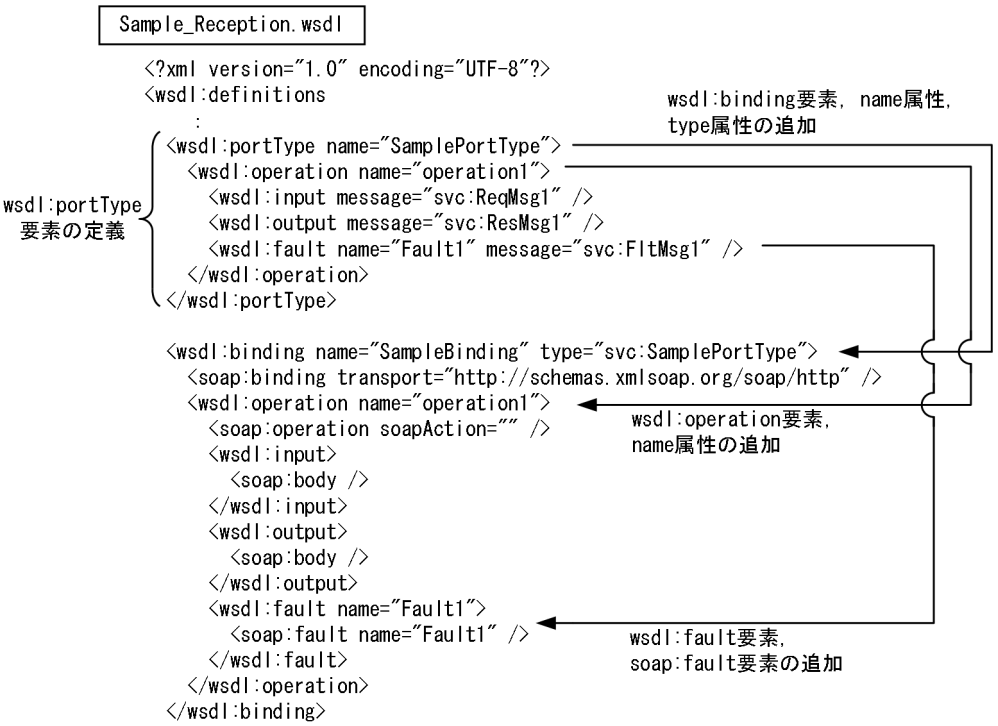
−：該当しません。

注※

使用できる文字種には制限があります。使用できる文字種については，「2.11.3(2)　WSDL 定義で使用できる文字」を参照してください。

wsdl:binding 要素の定義例を次の図に示します。

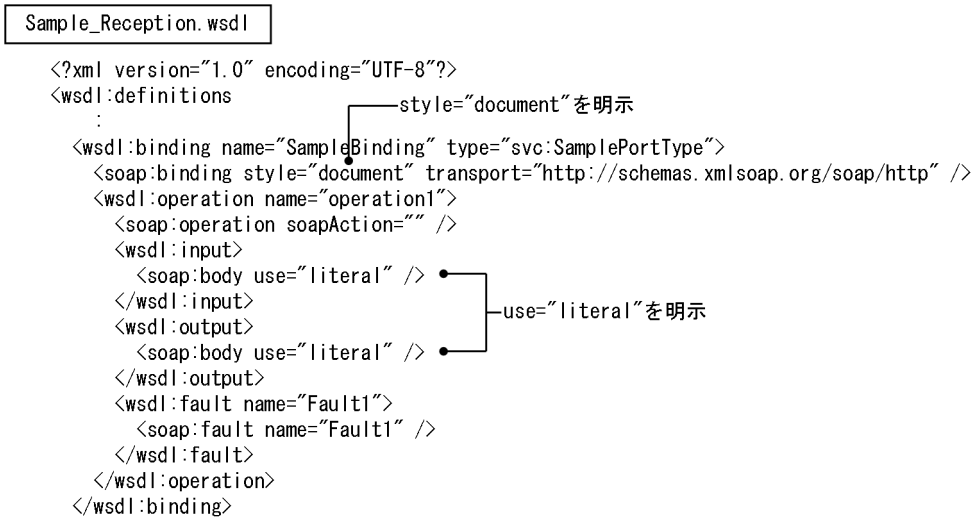
図 2-18　wsdl:binding 要素の定義例



style 属性および use 属性を使用する場合

wsdl:binding 要素以下に style 属性および use 属性を使用して，document または literal を明示することもできます。

図 2-19　wsdl:binding 要素の定義例（style 属性および use 属性を使用する場合）



(h)　wsdl:service 要素の定義

wsdl:service 要素とその下位要素，および属性を定義します。

1.「(c)　WSDL ファイルのひな形の作成（wsdl:definitions 要素の定義）」で定義した wsdl:definitions 要素の子要素として，wsdl:service 要素を定義します。

2. wsdl:service 要素に name 属性を定義します。

name 属性の値には，任意の文字列を指定できます。

3. wsdl:service 要素の子要素として，wsdl:port 要素を 1 つ定義します。

4. wsdl:port 要素に name 属性と binding 属性を定義します。

name 属性の値には，任意の文字列を指定できます。

binding 属性の値には，「(g)　wsdl:binding 要素の定義」で定義した wsdl:binding 要素の name 属性の値を QName で指定します。

5. wsdl:port 要素の子要素として，soap:address 要素を 1 つ定義します。

6. soap:address 要素に location 属性を定義します。

location 属性の値には，仮の値として「http://localhost:80/SampleService/services/<PortName>」を指定します。

<PortName>には，手順 4.で定義した name 属性と同じ値を指定してください。

ポイント

WSDL を作成した段階では，サービスロケーションの値（wsdl:port 要素内の soap-address 要素の location 属性値）は決定していません。このため，仮の値を設定します。この値は，サービスプラットフォームが提供している SOAP 通信基盤および JAX-WS エンジンでエラーにならない値です。

サービスロケーションの値は，ユーザ定義ファイルを含む HCSC コンポーネントを配備定義したあとに確定します。WSDL を編集して，確定したサービスロケーションの値を設定してから，WSDL を基にサービスリクエスタを作成してください。WSDL の編集の詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「9.7.2　WSDL を編集する」を参照してください。

wsdl:service 要素以下に定義する要素と属性を次の表に示します。

表 2-46　wsdl:service 要素以下に定義する要素と属性

要素	属性	意味	値	個数
wsdl:service	－	サービス定義要素	－	1
	name	サービス名属性	任意の文字列※を定義します。	1
wsdl:port	－	ポート定義要素	－	1
	name	ポート名属性	任意の文字列※を定義します。	1
	binding	バインディング参照属性	「(g)　wsdl:binding 要素の定義」で定義した wsdl:binding 要素の name 属性値を QName で指定します。	1

要素			属性	意味	値	個数
		soap:address	－	SOAP バインディングアドレス定義要素	－	1
			location	サービス位置定義属性	仮の値として，「http://localhost:80/SampleService/services/<PortName>」を定義します。 <PortName>には，wsdl:port 要素の name 属性と同じ値を指定します。	1

（凡例）

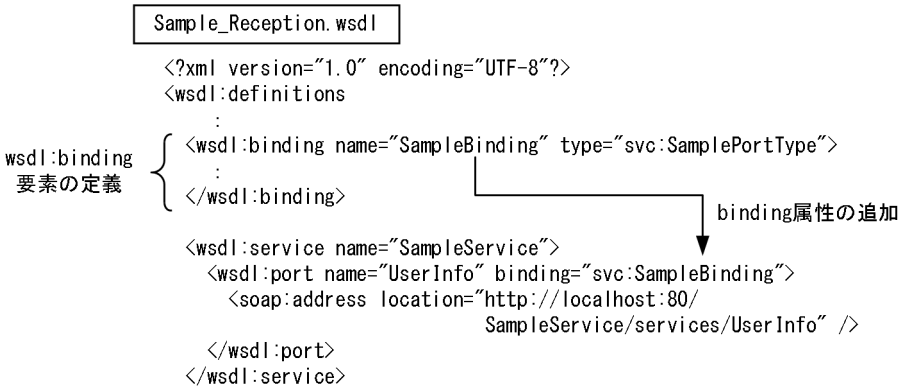
－：該当しません。

注※

使用できる文字種には制限があります。使用できる文字種については，「2.11.3(2)　WSDL 定義で使用できる文字」を参照してください。

wsdl:service 要素の定義例を次の図に示します。

図 2-20　wsdl:service 要素の定義例



複数の wsdl:port を使用する場合

作成例（Sample_Reception.wsdl）では，1 つの wsdl:port 要素にすべてを定義していますが，複数の wsdl:port 要素を使用して，それぞれにオペレーションを定義することもできます。

複数の wsdl:port 要素を使用する場合の定義例を次の図に示します。

図 2-21　wsdl:service 要素の定義例（複数の wsdl:port を使用する場合）



2.11.2　サービスアダプタの接続先情報を動的に変更する例

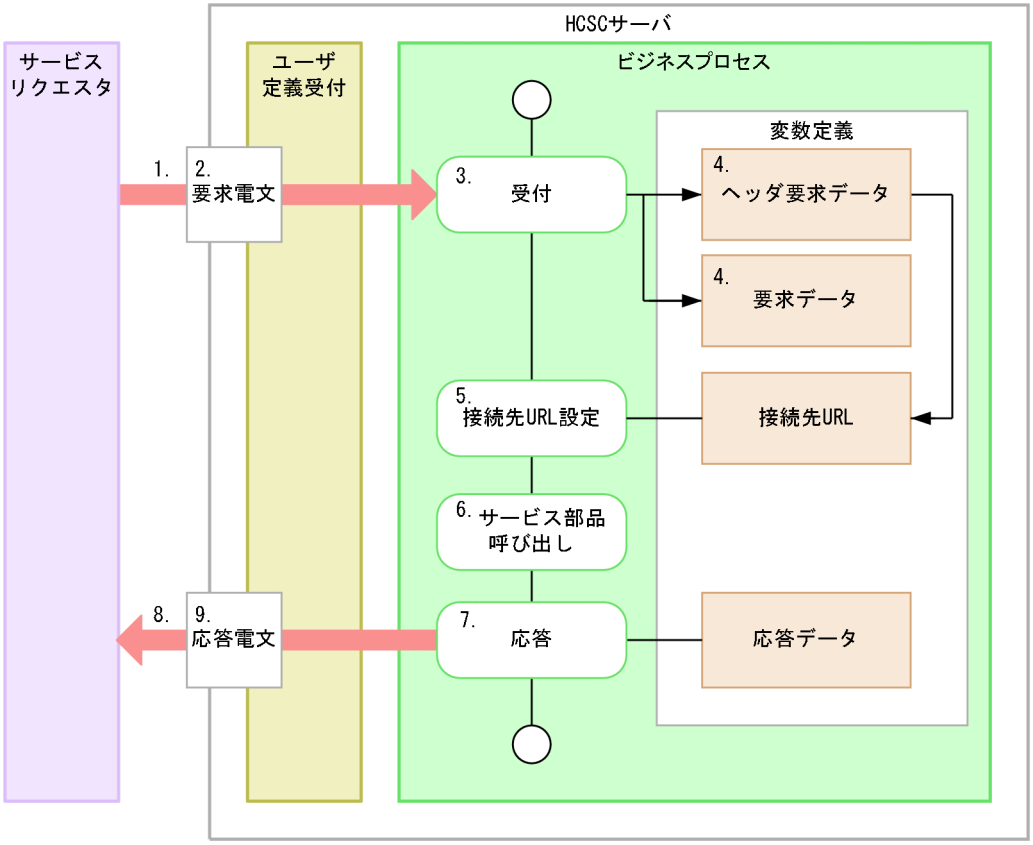
接続先情報ファイルを使用することで，サービスアダプタの接続先情報を動的に変更できます。

ここでは，SOAP 受付およびサービスアダプタを使用した SOAP 通信を行う場合を使って，接続先情報ファイルの使用例を説明します。

(1)　ビジネスプロセスの例

接続先情報ファイルの使用方法の例として使用するビジネスプロセスを次の図に示します。

図 2-22　接続先情報ファイルを使用するビジネスプロセスの例

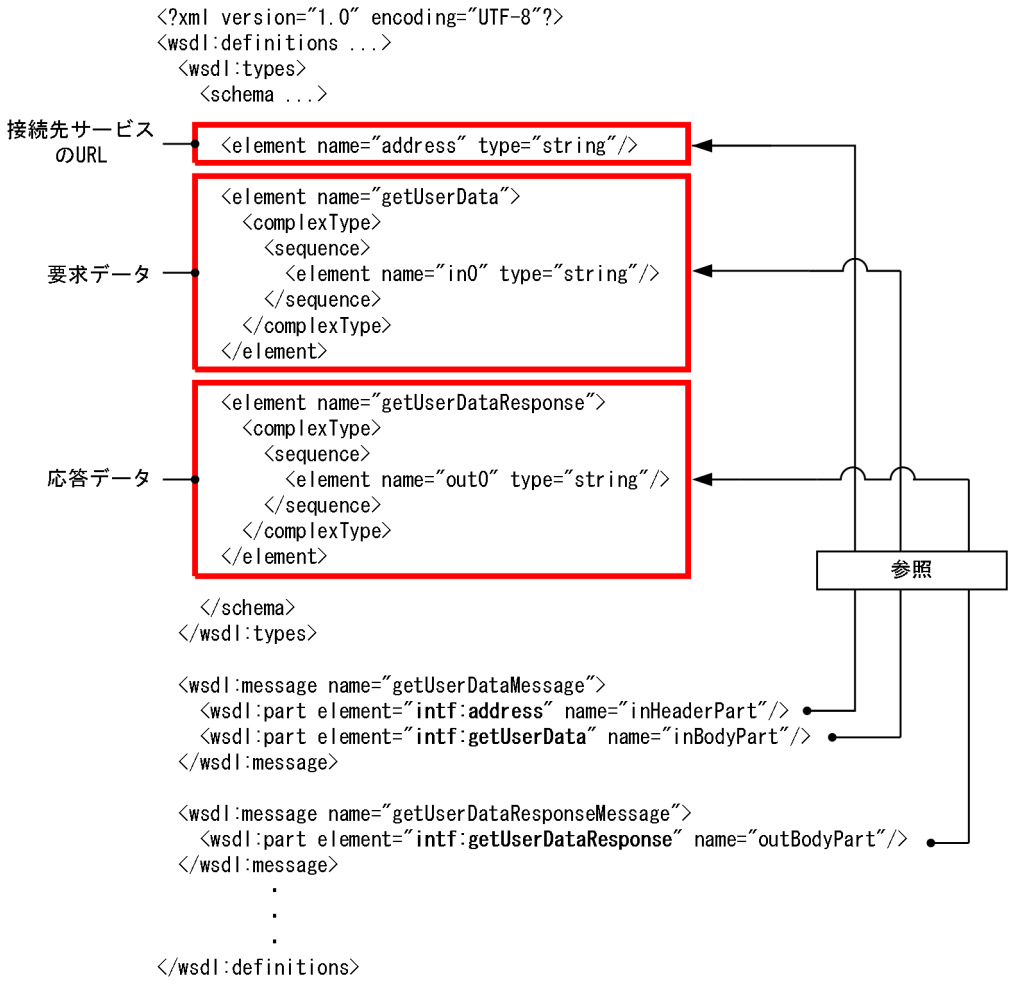


1. サービスリクエスタがビジネスプロセスにサービス部品の実行を要求します。
2. サービスリクエスタから送付する要求電文内のヘッダ電文に接続先 URL を設定します。
3. ビジネスプロセスは XML の要求電文を受付アクティビティで受け取ります。
4. ビジネスプロセスがヘッダ要求データおよび要求データを生成します。
5. ヘッダ要求データの接続先 URL を，データ変換アクティビティなどで接続先 URL 変数に設定します。
6. ビジネスプロセスは接続先 URL 変数から接続先 URL を取得して，サービス部品を呼び出します。
7. ビジネスプロセスが，サービス部品の実行結果を XML の応答電文として構成し，応答アクティビティで返します。
8. サービスリクエスタへの応答が返されます。
9. サービスリクエスタへの応答では，XML の応答電文が送付されます。

(2)　SOAP 受付を生成する WSDL の作成例

SOAP 受付を生成する WSDL の作成例を次に示します。

なお，この例では，接続先となるサービス部品の URL を SOAP ヘッダで送付しています。

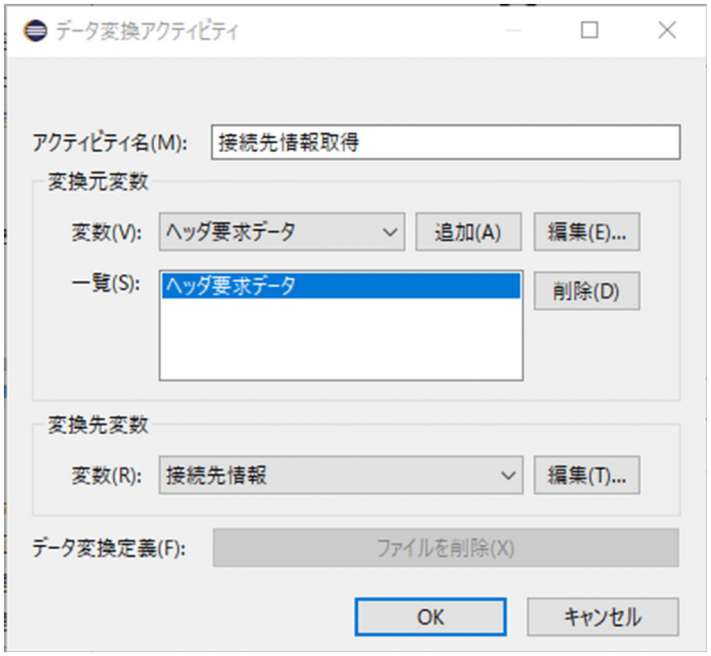


(3)　データ変換アクティビティの設定例

「図 2-22　接続先情報ファイルを使用するビジネスプロセスの例」に示すビジネスプロセスで使用するデータ変換アクティビティの設定例を次に示します。

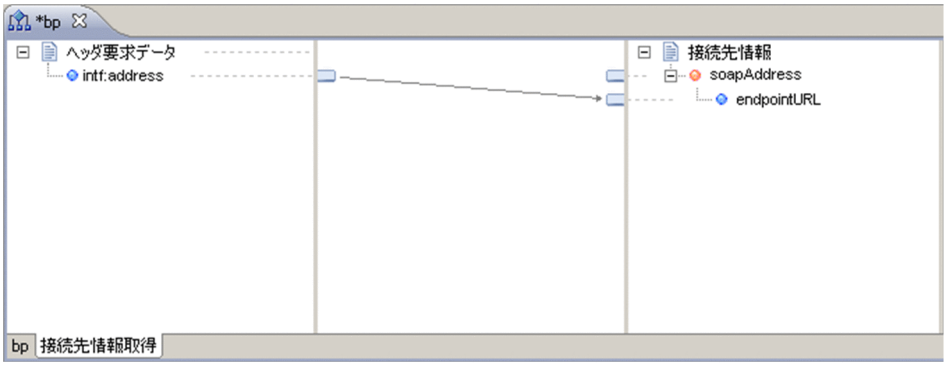
ここでは，変換元に「ヘッダ要求データ」を，変換先に「接続先情報」を指定しています。

図 2-23　データ変換アクティビティの設定例



また，変換のマッピングは，次の図に示すとおりに定義します。

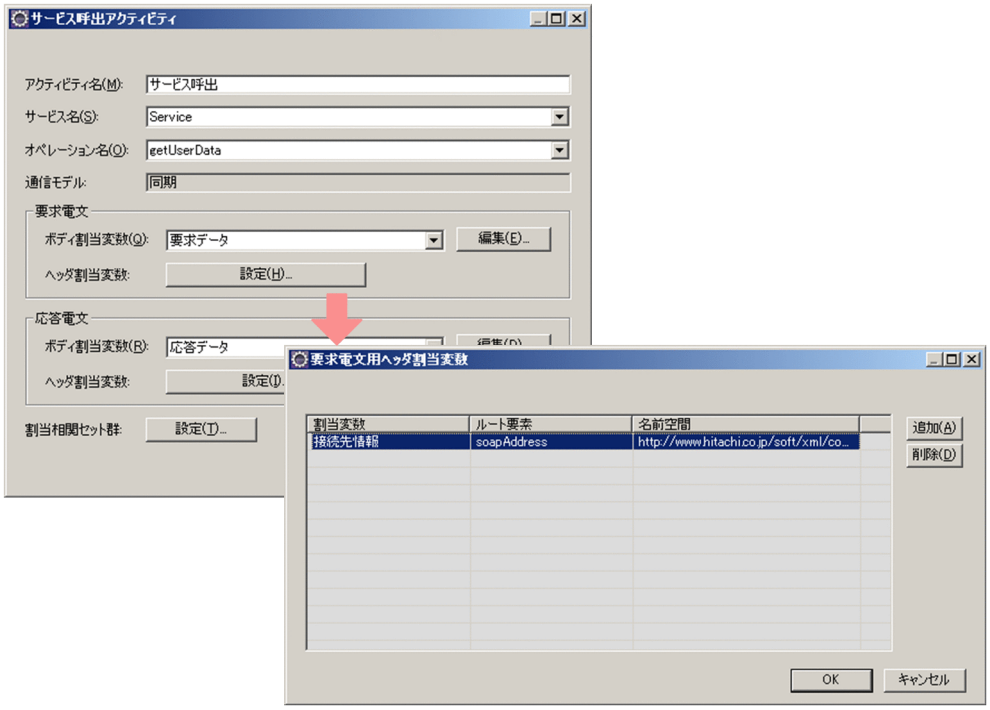
図 2-24　変換のマッピングの設定例



(4)　接続先情報の設定

「図 2-22　接続先情報ファイルを使用するビジネスプロセスの例」に示すビジネスプロセスで使用する接続先情報の設定例を次に示します。なお，接続先情報は，サービス呼出アクティビティのヘッダ割当変数に設定します。ここで設定した情報が，サービスアダプタで参照できる情報になります。

図 2-25　接続先情報の設定例（サービス呼出アクティビティの設定）



各サービスアダプタの接続先情報ファイルの格納先を次に示します。

- DB アダプタ用の接続先情報ファイルの格納先

  <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¥schema¥connection¥connection_db_adapter.xsd

- SOAP アダプタの接続先情報ファイルの格納先

  <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¥schema¥connection¥connection_soap_adapter.xsd

- TP1 アダプタ用の接続先情報ファイルの格納先

  <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¥schema¥connection¥connection_tp1_adapter.xsd

注意事項

サービスアダプタの接続先情報を動的に変更する場合，サービスプラットフォームで提供している接続先情報ファイルを使用してください。提供している接続先情報ファイルを使用しなかった場合，動作は保証されません。

2.11.3　SOAP 受付の WSDL 作成時の注意事項

SOAP1.1 モードを利用する場合で，SOAP 受付の WSDL を作成するときの注意事項と WSDL 定義で使用できる文字について説明します。

SOAP1.1/1.2 併用モードを利用する場合の注意事項については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.1　Web サービスを利用したサービス部品の適用範囲」の WSDL の定義時と使用時の注意（SOAP1.1/1.2 併用モード）に関する説明を参照してください。

(1)　WSDL 作成時の注意事項（SOAP 受付）

SOAP 受付の場合，WSDL を作成するときに次の点に注意する必要があります。

- wsdl:types 要素以下にあるすべての XML スキーマの名前空間（targetNamespace）に，有効な値を指定してください。
- XML スキーマは，マニュアル「アプリケーションサーバ SOAP アプリケーション開発の手引」の「12.2.4　XML Schema のサポート範囲」に記載されている XML スキーマのサポート範囲内で定義してください。
- wsdl:import 要素は指定できません。
- style 属性が「document」の場合，soap:body 要素に namespace 属性を指定しないでください。
- style 属性が「rpc」の場合，soap:body 要素に namespace 属性を指定してください。
- soap:header 要素は指定できません。
- soap:fault 要素の name 属性の値は，その親の wsdl:fault 要素の name 属性の値と一致させてください。また，namespace 属性を指定しないでください。

(2)　WSDL 定義で使用できる文字

WSDL 定義では，RFC3986 で規定されている非予約文字と日本語を使用できます。ここでは，WSDL 定義で使用できる文字，および使用する文字に関する注意事項を示します。

(a)　「document/literal」の場合

半角英数字（A～Z，a～z，0～9），アンダーバー（_），および次に示す文字を使用できます。

- 全角ひらがな，全角カタカナ
- 全角ギリシャ文字，全角ロシア文字など
- 繰り返し符号（「ゝ」，「ゞ」，「々」）
- 漢字（JIS X 0208 の第一水準，第二水準に含まれる範囲）

WSDL のデータ型が anyURI の属性については，次の文字も使用できます。

- ハイフン（-）

- ピリオド（.）
- チルダ（~）

(b) 「rpc/literal」の場合

半角英数字（A～Z，a～z，0～9）およびアンダーバー（_）を使用できます。WSDL のデータ型が anyURI の属性については，次の文字も使用できます。

- ハイフン（-）
- ピリオド（.）
- チルダ（~）

2.12　ユーザ定義受付の内容の確認

ユーザ定義受付を追加したあとには，定義された内容を確認してください。ユーザ定義受付の内容は，ユーザ定義受付定義画面で確認できます。ユーザ定義受付定義画面は，ユーザ定義受付追加ウィザードでユーザ定義受付を追加したあとに表示されます。

ユーザ定義受付定義画面を任意のタイミングで表示したい場合は，次の手順で表示させてください。

1. ツリービューのサービス定義一覧で，ユーザ定義受付をダブルクリックします。

   ユーザ定義受付定義画面が表示されます。

ユーザ定義受付定義画面の詳細については，マニュアル「サービスプラットフォーム リファレンス」の「1.2.5　ユーザ定義受付定義画面」を参照してください。

ユーザ定義受付定義画面では，ユーザ定義受付の定義内容を確認して，受付アクティビティおよび応答アクティビティの定義と不整合になっていないか確認してください。不整合になっている場合，受付アクティビティまたは応答アクティビティの定義を変更してください。

2.13　ユーザ定義受付の保存

ユーザ定義受付定義画面で編集中のユーザ定義受付を編集途中や編集終了時に保存できます。ユーザ定義受付の内容はリポジトリに保存されます。

ユーザ定義受付を保存する手順を次に示します。

方法 1

Eclipse のメニューから，［ファイル］－［保存］を選択します。

方法 2

Eclipse のメニューから，［ファイル］－［すべて保管］を選択します。

方法 3

Eclipse のツールバーの，［保存］アイコンをクリックします。

方法 4

ユーザ定義受付定義画面で，［Ctrl］キー＋［S］キーを押します。

なお，ユーザ定義受付定義画面を閉じてユーザ定義受付の定義を終了する場合に，ユーザ定義受付が保存されていないときは，リソース保管ダイアログが表示されます。リソース保管ダイアログでは，保存されていないユーザ定義受付の定義を保存できます。

注意事項

不正なデータが入力されている場合，ユーザ定義受付を保存できないことがあります。その場合，表示されているメッセージに従って対処してください。

2.14　ユーザ定義受付の検証

サービスプラットフォームでは，作成したユーザ定義受付の内容が妥当かどうか検証できます。

2.14.1　検証内容

ユーザ定義受付の検証での検証内容と対処を次の表に示します。

表 2-47　ユーザ定義受付の検証内容と対処

項番	検証内容	エラーになった場合の対処
1	ユーザ定義受付のオペレーション名と一致するオペレーション名がビジネスプロセス内に定義されているか。	ビジネスプロセスの受付アクティビティおよびユーザ定義受付のオペレーション名を見直し，一致するオペレーション名を定義します。
2	ユーザ定義受付のオペレーションの通信モデル（同期／非同期）と，ビジネスプロセスのオペレーションの通信モデル（同期／非同期）が一致しているか。	ビジネスプロセスの受付アクティビティおよびユーザ定義受付の定義を見直し，オペレーションの通信モデル（同期／非同期）を合わせます。
3	ユーザ定義受付に対応する受付アクティビティ，および応答アクティビティ（フォルト電文用を含む）の割当変数の種別がメッセージ型か。	ビジネスプロセスの定義を見直し，該当するアクティビティの割当変数をユーザ定義受付に対応する変数に変更します。
4	ユーザ定義受付に対応する応答電文用（フォルト電文用は含まない）の応答アクティビティが存在するか。	ビジネスプロセスの定義を見直し，ユーザ定義受付のオペレーションに対応する応答アクティビティを定義します。
5	ユーザ定義受付のオペレーションのフォルト名と一致するフォルト名がビジネスプロセスの応答アクティビティに存在するか。	ビジネスプロセスおよびユーザ定義受付の定義を見直し，次のどちらかの対処をします。 • ユーザ定義受付のオペレーションのフォルト名に対応する応答アクティビティを定義します。 • ユーザ定義受付のオペレーションから不要なフォルトの定義を削除します。この場合，一度ユーザ定義受付を削除し，フォルトの定義を削除した WSDL ファイルからもう一度ユーザ定義受付を作成します。
6	ユーザ定義受付のオペレーションに存在しないフォルト名がビジネスプロセスの応答アクティビティに定義されていないか。	ビジネスプロセスおよびユーザ定義受付の定義を見直し，次のどちらかの対処をします。 • ユーザ定義受付のオペレーションで定義されていないフォルトに対応する応答アクティビティを削除します。 • ユーザ定義受付のオペレーションにフォルトの定義を追加します。この場合，一度ユーザ定義受付を削除し，フォルトの定義を追加した WSDL ファイルからもう一度ユーザ定義受付を作成します。

2.14.2　検証方法

ユーザ定義受付の検証は，ビジネスプロセスを検証するときにあわせて実施されます。ビジネスプロセスを検証すると，検証対象のビジネスプロセス内のユーザ定義受付が検証されます。

ビジネスプロセスの検証方法の詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.11.2　検証方法」を参照してください。

また，ユーザ定義受付の検証結果は，ビジネスプロセスの検証結果と同様に，コンソールビューに表示されます。検証結果の表示の詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.11.3　検証内容の表示」を参照してください。

2.15　ユーザ定義受付の情報の変更

ユーザ定義受付定義画面で，受付名，受付 ID，およびコンテキストルートを変更できます。

ユーザ定義受付の情報を変更する手順を次に示します。

1. ツリービューのサービス定義一覧で，ユーザ定義受付をダブルクリックします。
   ユーザ定義受付定義画面が表示されます。

2. [ユーザ定義受付情報] の［受付名］，［受付 ID］，および［コンテキストルート］のテキストボックスに，変更後の値を入力します。

ユーザ定義受付定義画面および入力できる値については，マニュアル「サービスプラットフォーム リファレンス」の「1.2.5　ユーザ定義受付定義画面」を参照してください。

2.16　ユーザ定義受付の削除

ユーザ定義受付を削除します。なお，次のユーザ定義受付は削除できません。

- すでに公開されているビジネスプロセスが持つユーザ定義受付（呼出先固定）
- サービスとして公開されているユーザ定義受付（呼出先選択）
- 以前のバージョンのビジネスプロセスで作成したユーザ定義受付

2.16.1　1 つのユーザ定義受付を削除する場合

1 つのユーザ定義受付を削除する手順について，次に示します。

方法 1

1. ツリービューのサービス定義一覧で，削除するユーザ定義受付を右クリックします。
   対象のユーザ定義受付のポップアップメニューが表示されます。
2. ポップアップメニューから［削除］を選択します。
   確認ダイアログが表示されます。
3. ［はい］ボタンをクリックします。
   指定したユーザ定義受付がビジネスプロセスから削除されます。

方法 2

1. ツリービューのサービス定義一覧で，削除するユーザ定義受付を選択します。
2. ［Delete］キーを押します。
   確認ダイアログが表示されます。
3. ［はい］ボタンをクリックします。
   指定したユーザ定義受付がビジネスプロセスから削除されます。

2.16.2　複数のユーザ定義受付（呼出先固定）を同時に削除する場合

複数のユーザ定義受付（呼出先固定）を同時に削除する手順について，次に示します。

1. ツリービューのサービス定義一覧で，ユーザ定義受付（呼出先固定）を削除するビジネスプロセスを右クリックします。
   サービス一覧のポップアップメニューが表示されます。
2. ポップアップメニューから［ユーザ定義受付（呼出先固定）削除］を選択します。

［ユーザ定義受付削除］ダイアログが表示されます。［ユーザ定義受付削除］ダイアログの入力・表示内容の詳細については，マニュアル「サービスプラットフォーム リファレンス」の「1.7.1　ユーザ定義受付削除ダイアログ」を参照してください。

3. 削除するユーザ定義受付（呼出先固定）のチェックボックスをチェックします。

4. ［OK］ボタンをクリックします。

   指定したユーザ定義受付（呼出先固定）がビジネスプロセスから削除されます。

注意事項

ビジネスプロセスを削除すると，削除したビジネスプロセスに含まれるユーザ定義受付も削除されます。

3 サービスアダプタの定義

この章では，各種サービスアダプタの定義について説明します。

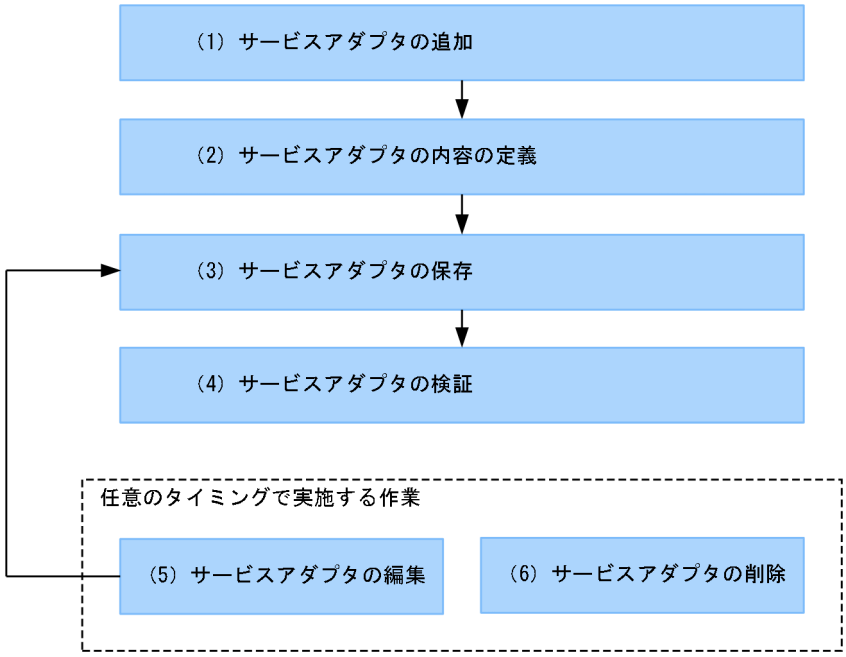
3.1　サービスアダプタの定義の手順

ここでは，サービスアダプタの定義の手順について説明します。

3.1.1　定義の流れ

サービスアダプタの定義の流れを次の図に示します。

図 3-1　サービスアダプタの定義の流れ



サービスアダプタの定義に関する作業について次に説明します。

(1)　サービスアダプタの追加

サービスアダプタを追加するには，次のどちらかの方法を利用します。

新規追加する場合

ウィザードを利用してサービスアダプタを新規に追加します。アダプタには，標準で使用するサービスアダプタと，ユーザが必要に応じて使用する汎用カスタムアダプタがあります。

サービスアダプタを新規に追加するウィザードは，サービスアダプタが呼び出すサービス部品の種類（Web サービス，SessionBean，MDB（WS-R または DB キュー）など）によって異なります。

サービスアダプタを新規に追加する方法の詳細については，「3.2.1　SOAP アダプタを新規に追加する」を参照してください。

サービスアダプタには次の種類があります。

- SOAP アダプタ
- SessionBean アダプタ

- MDB（WS-R）アダプタ
- MDB（DB キュー）アダプタ
- DB アダプタ
- TP1 アダプタ
- ファイルアダプタ
- Object Access アダプタ
- Message Queue アダプタ
- FTP アダプタ
- ファイル操作アダプタ
- メールアダプタ
- HTTP アダプタ
- コマンドアダプタ
- SFTP アダプタ
- Kafka アダプタ
- gRPC アダプタ
- 汎用カスタムアダプタ

それぞれのサービスアダプタを新規に追加する方法の詳細は，次に示す個所を参照してください。

種別	参照個所
SOAP アダプタ	3.2.1　SOAP アダプタを新規に追加する
SessionBean アダプタ	3.2.2　SessionBean アダプタを新規に追加する
MDB（WS-R）アダプタ	3.2.3　MDB（WS-R）アダプタを新規に追加する
MDB（DB キュー）アダプタ	3.2.4　MDB（DB キュー）アダプタを新規に追加する
DB アダプタ	3.2.5　DB アダプタを新規に追加する
TP1 アダプタ	3.2.6　TP1 アダプタを新規に追加する
ファイルアダプタ	3.2.7　ファイルアダプタを新規に追加する
Object Access アダプタ	3.2.8　Object Access アダプタを新規に追加する
Message Queue アダプタ	3.2.9　Message Queue アダプタを新規に追加する
FTP アダプタ	3.2.10　FTP アダプタを新規に追加する
ファイル操作アダプタ	3.2.11　ファイル操作アダプタを新規に追加する
メールアダプタ	3.2.12　メールアダプタを新規に追加する
HTTP アダプタ	3.2.13　HTTP アダプタを新規に追加する
コマンドアダプタ	3.2.14　コマンドアダプタを新規に追加する

種別	参照個所
SFTP アダプタ	3.2.15　SFTP アダプタを新規に追加する
Kafka アダプタ	3.2.16　Kafka アダプタを新規に追加する
gRPC アダプタ	3.2.17　gRPC アダプタを新規に追加する
汎用カスタムアダプタ	3.2.18　汎用カスタムアダプタを新規に追加する

定義済みのサービスアダプタを利用する場合

定義済みのサービスアダプタを複製してサービスアダプタを追加できます。定義済みのサービスアダプタを複製すると，同じ定義内容のサービスアダプタが追加されます。複製したサービスアダプタの定義内容を編集することもできます。

定義済みのサービスアダプタを利用してサービスアダプタを追加する方法の詳細については，「3.2.19 定義済みサービスアダプタを利用してサービスアダプタを追加する」を参照してください。

(2)　サービスアダプタの内容の定義

サービスアダプタ定義画面でサービスアダプタの内容を定義します。

定義の方法および定義の内容の詳細については，「3.3　サービスアダプタの内容の定義」を参照してください。サービスアダプタ定義画面の詳細については，マニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

(3)　サービスアダプタの保存

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

サービスアダプタの定義情報の保存方法と保存されるタイミングについては，「3.4　サービスアダプタの保存」を参照してください。

(4)　サービスアダプタの検証

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

検証方法の詳細については，「3.6　サービスアダプタの検証」を参照してください。

(5)　サービスアダプタの編集

定義済みのサービスアダプタは，必要に応じて定義内容を編集して変更できます。

サービスアダプタの編集方法の詳細については，「3.5　サービスアダプタの編集」を参照してください。

(6)　サービスアダプタの削除

不要になったサービスアダプタは，必要に応じて削除できます。

サービスアダプタの削除方法については，「3.7　サービスアダプタの削除」を参照してください。

3.2　サービスアダプタの追加

サービスアダプタを追加します。

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

注意事項

警告メッセージが表示された場合は，マニュアル「アプリケーションサーバ メッセージ(構築／運用／開発用)」の次の個所を参照して，表示されたメッセージの内容を確認してください。

- 「KDCCC」で始まる警告メッセージが表示された場合
  「5.2　KDCCC0001 から KDCCC9999 までのメッセージ」
- 「KDJW」で始まる警告メッセージが表示された場合
  「10.　KDJW（Component Container が出力するメッセージ)」

3.2.1　SOAP アダプタを新規に追加する

SOAP アダプタを新規に追加する方法を次に示します。

1. Eclipse のメニューから，［ウィンドウ］－［ビューの表示］－［その他］を選択します。
   ［ビューの表示］ダイアログが表示されます。
2. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。
3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。
4. ドロップダウンリストから「Web サービス」を選択します。
5. ［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。
6. サービス名と WSDL ファイルを指定します。
   WSDL ファイルの指定は，次のどちらかを選択します。
   - ファイル
     WSDL ファイルをパスで指定します。

「ファイル」ラジオボタンを選択し，拡張子が「.wsdl」のファイルを絶対パスで指定します。相対パス（例：wsdldir¥wsdlfile.wsdl）や UNC 形式（例：¥¥mypc¥wsdldir¥wsdlfile.wsdl）で指定しないでください。

- URL

  WSDL を URL で指定します。

  「URL」ラジオボタンを選択し，テキストボックスに参照先の URL を指定します。

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。

   [次へ] ボタンをクリックしたときに，「WSDL ファイルの解析に失敗しました。」というメッセージが表示された場合，WSDL ファイルにエラーがあります。

   WSDL の形式については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.1 Web サービスを利用したサービス部品の適用範囲」を参照してください。

8. ドロップダウンリストでポートを選択します。

9. [終了] ボタンをクリックします。

   [終了] ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。

   サービスアダプタ定義画面が表示され，この画面で SOAP アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(1)　SOAP アダプタの場合」を参照してください。

3.2.2　SessionBean アダプタを新規に追加する

SessionBean アダプタを新規に追加する方法を次に示します。

1. Eclipse のメニューから，[ウィンドウ] － [ビューの表示] － [その他] を選択します。

   [ビューの表示] ダイアログが表示されます。

2. [HCSC-Definer] － [HCSCTE ビュー] を選択し，[開く] ボタンをクリックします。

   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。

   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。

4. ドロップダウンリストから「SessionBean」を選択します。

5. [次へ] ボタンをクリックします。

   SessionBean アダプタの追加に必要な情報を入力するダイアログが表示されます。

6. サービス名と EAR ファイルを入力します。

   EAR ファイルには，拡張子が「.ear」のファイルを絶対パスで指定します。

7. [次へ] ボタンをクリックします。
   SessionBean を指定するダイアログが表示されます。

8. ドロップダウンリストで SessionBean を選択します。

9. [終了] ボタンをクリックします。
   [終了] ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面で SessionBean アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(2)　SessionBean アダプタの場合」を参照してください。

3.2.3　MDB（WS-R）アダプタを新規に追加する

MDB（WS-R）アダプタを新規に追加する方法を次に示します。

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

2. [HCSC-Definer] － [HCSCTE ビュー] を選択し，[開く] ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

4. ドロップダウンリストから「MDB/WS-R」を選択します。

5. [次へ] ボタンをクリックします。
   MDB（WS-R）アダプタの追加に必要な情報を入力するダイアログが表示されます。

6. サービス名を入力します。

7. [終了] ボタンをクリックします。
   [終了] ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面で MDB（WS-R）アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(3)　MDB（WS-R）アダプタの場合」を参照してください。

3.2.4　MDB（DB キュー）アダプタを新規に追加する

MDB（DB キュー）アダプタを新規に追加する方法を次に示します。

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

2. [HCSC-Definer] － [HCSCTE ビュー] を選択し，[開く] ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

4. ドロップダウンリストから「MDB/DB キュー」を選択します。

5. [次へ] ボタンをクリックします。
   MDB（DB キュー）アダプタの追加に必要な情報を入力するダイアログが表示されます。

6. サービス名を入力します。

7. [終了] ボタンをクリックします。
   [終了] ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面で MDB（DB キュー）アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(4)　MDB（DB キュー）アダプタの場合」を参照してください。

3.2.5　DB アダプタを新規に追加する

DB アダプタを新規に追加する方法を次に示します。

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

2. [HCSC-Definer] － [HCSCTE ビュー] を選択し，[開く] ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

4. ドロップダウンリストから「DB アダプタ」を選択します。
   [次へ] ボタンをクリックすると，DB アダプタの追加に必要な情報を入力するダイアログが表示されます。

5. サービス名を入力します。
   サービス名は，サービスリクエスタから HCSC サーバを呼ぶ際に必要になる識別情報です。

6. [終了] ボタンをクリックします。
   [終了] ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面で DB アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(5) DB アダプタの場合」を参照してください。

3.2.6　TP1 アダプタを新規に追加する

TP1 アダプタを新規に追加する方法を次に示します。

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

2. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   ［サービスアダプタ定義追加］ダイアログが表示されます。

4. サービス部品の種別のドロップダウンリストから「TP1 アダプタ」を選択します。

5. ［次へ］ボタンをクリックします。
   TP1 アダプタの追加に必要な情報を入力するダイアログが表示されます。

6. サービス名を入力します。
   サービス名には，任意のサービス名を入力します。

7. ［終了］ボタンをクリックします。
   ［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面で TP1 アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(6)　TP1 アダプタの場合」を参照してください。

3.2.7　ファイルアダプタを新規に追加する

ファイルアダプタを新規に追加する方法を次に示します。

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

2. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

4. サービス部品の種別のドロップダウンリストから「ファイルアダプタ」を選択します。
   ［次へ］ボタンをクリックすると，ファイル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. サービス名を入力します。

サービス名には，任意のサービス名を入力します。

6. [終了] ボタンをクリックします。
   [終了] ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面でファイルアダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(7)　ファイルアダプタの場合」を参照してください。

3.2.8　Object Access アダプタを新規に追加する

Object Access アダプタを新規に追加する方法を次に示します。

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

2. [HCSC-Definer] − [HCSCTE ビュー] を選択し，[開く] ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

4. サービス部品の種別のドロップダウンリストから「Object Access アダプタ」を選択します。
   [次へ] ボタンをクリックすると，Object Access 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. サービス名を入力します。
   サービス名には，任意のサービス名を入力します。

6. [終了] ボタンをクリックします。
   [終了] ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面で Object Access アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(8)　Object Access アダプタの場合」を参照してください。

3.2.9　Message Queue アダプタを新規に追加する

Message Queue アダプタを新規に追加する方法を次に示します。

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

2. [HCSC-Definer] − [HCSCTE ビュー] を選択し，[開く] ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

4. サービス部品の種別のドロップダウンリストから「Message Queue アダプタ」を選択します。
   [次へ] ボタンをクリックすると，Message Queue 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. サービス名を入力します。
   サービス名には，任意のサービス名を入力します。

6. [終了] ボタンをクリックします。
   [終了] ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面で Message Queue アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(9)　Message Queue アダプタの場合」を参照してください。

3.2.10　FTP アダプタを新規に追加する

FTP アダプタを新規に追加する方法を次に示します。

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

2. [HCSC-Definer] − [HCSCTE ビュー] を選択し，[開く] ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

4. サービス部品の種別のドロップダウンリストから「FTP アダプタ」を選択します。
   [次へ] ボタンをクリックすると，FTP 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. サービス名を入力します。
   サービス名には，任意のサービス名を入力します。

6. [終了] ボタンをクリックします。
   [終了] ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面で FTP アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(10)　FTP アダプタの場合」を参照してください。

3.2.11　ファイル操作アダプタを新規に追加する

ファイル操作アダプタを新規に追加する方法を次に示します。

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

2. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

4. サービス部品の種別のドロップダウンリストから「ファイル操作アダプタ」を選択します。
   ［次へ］ボタンをクリックすると，ファイル操作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. サービス名を入力します。
   サービス名には，任意のサービス名を入力します。

6. ［終了］ボタンをクリックします。
   ［終了］ボタンをクリックすると，サービスアダプタ定義画面が表示され，この画面でファイル操作アダプタを定義します。オペレーションごとの設定内容は，次の個所を参照してください。

   - ファイル変換オペレーションの場合
     3.3.19(11)(a)　ファイル変換オペレーション
   - ファイル複製オペレーションの場合
     3.3.19(11)(b)　ファイル複製オペレーション
   - ファイル移動オペレーションの場合
     3.3.19(11)(c)　ファイル移動オペレーション
   - ファイル・フォルダ削除オペレーションの場合
     3.3.19(11)(d)　ファイル・フォルダ削除オペレーション
   - ファイル圧縮オペレーションの場合
     3.3.19(11)(e)　ファイル圧縮オペレーション
   - ファイル伸張オペレーションの場合
     3.3.19(11)(f)　ファイル伸張オペレーション
   - ファイル一覧取得オペレーションの場合
     3.3.19(11)(g)　ファイル一覧取得オペレーション
   - ファイル一覧書き込みオペレーションの場合
     3.3.19(11)(h)　ファイル一覧書き込みオペレーション

- ファイル読み込みオペレーションの場合
  3.3.19(11)(i)　ファイル読み込みオペレーション
- ファイル書き込みオペレーションの場合
  3.3.19(11)(j)　ファイル書き込みオペレーション

3.2.12　メールアダプタを新規に追加する

メールアダプタを新規に追加する方法を次に示します。

1. Eclipse のメニューから，［ウィンドウ］－［ビューの表示］－［その他］を選択します。
   ［ビューの表示］ダイアログが表示されます。
2. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。
3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。
4. サービス部品の種別のドロップダウンリストから「メールアダプタ」を選択します。
   ［次へ］ボタンをクリックすると，メール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. サービス名を入力します。
   サービス名には，任意のサービス名を入力します。
6. ［終了］ボタンをクリックします。
   ［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面でメールアダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(12)　メールアダプタの場合」を参照してください。

3.2.13　HTTP アダプタを新規に追加する

HTTP アダプタを新規に追加する方法を次に示します。

1. Eclipse のメニューから，［ウィンドウ］－［ビューの表示］－［その他］を選択します。
   ［ビューの表示］ダイアログが表示されます。
2. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。
3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。

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。

4. サービス部品の種別のドロップダウンリストから「HTTP アダプタ」を選択します。
［次へ］ボタンをクリックすると，HTTP 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. サービス名を入力します。
サービス名には，任意のサービス名を入力します。

6. ［終了］ボタンをクリックします。
［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
サービスアダプタ定義画面が表示され，この画面で HTTP アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(13)　HTTP アダプタの場合」を参照してください。

3.2.14　コマンドアダプタを新規に追加する

コマンドアダプタを新規に追加する方法を次に示します。

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

2. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
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。

4. サービス部品の種別のドロップダウンリストから「コマンドアダプタ」を選択します。
［次へ］ボタンをクリックすると，コマンド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. サービス名を入力します。
サービス名には，任意のサービス名を入力します。

6. ［終了］ボタンをクリックします。
［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
サービスアダプタ定義画面が表示され，この画面でコマンドアダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(14)　コマンドアダプタの場合」を参照してください。

注意事項

コマンドアダプタの利用は，Service Coordinator 09-71-01 以降が前提です。

3.2.15　SFTP アダプタを新規に追加する

SFTP アダプタを新規に追加する方法を次に示します。

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

2. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

4. サービス部品の種別のドロップダウンリストから「SFTP アダプタ」を選択します。
   ［次へ］ボタンをクリックすると，SFTP 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. サービス名を入力します。
   サービス名には，任意のサービス名を入力します。

6. ［終了］ボタンをクリックします。
   ［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面で SFTP アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(15)　SFTP アダプタの場合」を参照してください。

3.2.16　Kafka アダプタを新規に追加する

Kafka アダプタを新規に追加する方法を次に示します。

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

2. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

4. サービス部品の種別のドロップダウンリストから「Kafka アダプタ」を選択します。
   ［次へ］ボタンをクリックすると，Kafka 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. サービス名を入力します。

サービス名には，任意のサービス名を入力します。

6. [終了] ボタンをクリックします。
   [終了] ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面で Kafka アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(16)　Kafka アダプタの場合」を参照してください。

3.2.17　gRPC アダプタを新規に追加する

gRPC アダプタを新規に追加する方法を次に示します。

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

2. [HCSC-Definer] − [HCSCTE ビュー] を選択し，[開く] ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

4. サービス部品の種別のドロップダウンリストから「gRPC アダプタ」を選択します。
   [次へ] ボタンをクリックすると，gRPC 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. サービス名，proto_path，proto ファイルを入力します。
   サービス名には，任意のサービス名を入力します。
   proto_path には，proto ファイルが格納されたフォルダのパスを指定します。
   proto ファイルには，proto ファイル名を絶対パスで指定します。
   proto ファイルを設定する際の注意事項については，マニュアル「サービスプラットフォーム 解説」の「2.13.1(3)(c)　proto ファイル使用時の注意事項」を参照してください。

6. [次へ] ボタンをクリックします。

7. gRPC Service の指定のドロップダウンリストから使用する gRPC サービスを選択します。

8. [終了] ボタンをクリックします。
   [終了] ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面で gRPC アダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(17)　gRPC アダプタの場合」を参照してください。

3.2.18　汎用カスタムアダプタを新規に追加する

汎用カスタムアダプタを新規に追加する方法を次に示します。

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

2. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

4. ドロップダウンリストから「汎用カスタムアダプタ」を選択します。
   ［次へ］ボタンをクリックすると，汎用カスタム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. サービス名を入力します。
   サービス名は，サービスリクエスタから HCSC サーバを呼ぶ際に必要になる識別情報です。

6. EAR ファイルを入力します。
   EAR ファイルには，拡張子が「.ear」のファイルを絶対パスで指定します。
   指定する EAR ファイルは，「3.3.18(6)　EAR ファイルの作成」で作成したファイルです。

7. ［終了］ボタンをクリックします。
   ［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。
   サービスアダプタ定義画面が表示され，この画面で汎用カスタムアダプタを定義します。サービスアダプタ定義画面での設定については，「3.3.19(18)　汎用カスタムアダプタの場合」を参照してください。

3.2.19　定義済みサービスアダプタを利用してサービスアダプタを追加する

すでに定義されているサービスアダプタを複製して，サービスアダプタを追加できます。

1. ツリービューのサービス定義一覧で，複製するサービスアダプタを選択して右クリックします。
   サービス一覧のポップアップメニューが表示されます。

2. ポップアップメニューから「複製」を選択します。
   選択したサービスアダプタの複製が作成されます。複製されたサービスアダプタのサービス名やサービス ID などは，システム内で競合しないよう，自動的に別の情報が割り当てられます。

3.3　サービスアダプタの内容の定義

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

3.3.1　SOAP アダプタを定義する

SOAP アダプタの定義方法について説明します。SOAP アダプタは，サービスアダプタ定義画面で定義できます。

(1)　サービス部品の電文フォーマット

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

サービス部品の電文フォーマットの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.3.2　サービス部品電文を作成する（Web サービスの場合)」を参照してください。

(2)　データ変換

サービスリクエスタの電文の形式と，サービスアダプタが呼び出すサービス部品の電文の形式が異なる場合，データ変換が必要です。

要求電文のデータ変換を行う場合，サービスアダプタ定義画面で標準電文にサービスリクエスタから入力される電文を設定し，データ変換定義画面でデータ変換を定義します。応答電文のデータ変換を行う場合，サービス部品から返される電文に標準電文を設定し，データ変換定義画面でデータ変換を定義します。

データ変換の詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7. データ変換の定義」を参照してください。

注意事項

SOAP アダプタでデータ変換を定義する場合，電文フォーマットのルート要素には次の要素を選択してください。

- document/literal の場合
  要求電文：<wsdl:operation><wsdl:input>が参照する<wsdl:message><wsdl:part>の element 属性が示す要素

応答電文：<wsdl:operation><wsdl:output>が参照する<wsdl:message><wsdl:part>の element 属性が示す要素

- rpc/literal の場合
  要求電文：「オペレーション名」と同じ名前の要素
  応答電文：「オペレーション名＋"Response"」と同じ名前の要素

(3)　ベーシック認証の設定

対象となるサービス部品がユーザ認証を必要とする Web サーバ上にある場合は，ベーシック認証（ユーザ名およびパスワード）の設定が必要になります。

(4)　サービスアダプタ定義画面の表示

サービスアダプタを定義するために，サービスアダプタ定義画面を表示します。サービスアダプタ定義画面は，ツリービューのサービス定義一覧から，該当するサービスアダプタを選択し，ダブルクリックすると表示されます。

(5)　サービスアダプタ定義画面での操作

SOAP アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。
   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービス部品制御情報の［サービス名］，［サービス ID］，［アドレス］，および［最大インスタンス数］を必要に応じて編集します。

3. サービス部品制御情報の［オペレーション］のドロップダウンリストから，編集したいオペレーションを選択します。

4. オペレーション情報に表示されている内容を確認します。

5. サービス部品制御情報の［システム例外をフォルトに変換する］チェックボックスを必要に応じてチェックします。

6. 要求電文について，手順 7.から手順 23.までを実施します。

7. 次に示す操作をします。

   要求電文に SOAP ヘッダ情報を定義する場合

   手順 8.～14.を実施して，手順 15.に進みます。

   要求電文に SOAP ヘッダ情報を定義しない場合

   手順 15.に進みます。

8. 要求電文の［ヘッダ］タブをクリックします。

9. ［ルート要素］のドロップダウンリストからヘッダ情報のルート要素を選択します。

［名前空間］には選択したルート要素の名前空間が表示され，［電文フォーマット］にはヘッダ情報として使用する電文フォーマットのファイル名が表示されます。必要に応じてヘッダ情報のルート要素を追加してください。

10. 次に示す操作をします。

ヘッダ情報のルート要素を追加する場合

手順 11.～13.を実施して，手順 14.に進みます。

ヘッダ情報のルート要素を追加しない場合

手順 14.に進みます。

11. ［ルート要素］の［追加］ボタンをクリックします。

［電文フォーマットファイルの選択］ダイアログが表示されます。

12. ヘッダ情報として使用する電文フォーマットを選択して［開く］ボタンをクリックします。

［ヘッダ情報の追加］ダイアログが表示されます。

13. ［ルート要素］のドロップダウンリストから追加するヘッダ情報のルート要素を選択して［OK］ボタンをクリックします。

追加したルート要素は，要求電文の［ヘッダ］タブの［ルート要素］に追加されます。追加位置は，ドロップダウンリストの末尾になります。

14. 電文フォーマットの［表示］ボタンをクリックします。

ヘッダ情報として使用する電文のフォーマットが表示されます。必要に応じて電文フォーマットを確認してください。

15. 要求電文の［ボディ］タブをクリックします。

16. 標準電文の［使う］チェックボックスをチェックします。

17. 標準電文の［参照］ボタンをクリックして，［電文フォーマット］に標準電文のフォーマットを指定します。

なお，外部の XML スキーマを参照している電文フォーマットを設定する場合，必ずルートスキーマに当たるファイルを設定してください。ルートスキーマから参照される外部の XML スキーマファイルは，自動的に取り込まれます。

指定できる電文フォーマットの形式については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.5　XML スキーマの適用範囲」を参照してください。

18. 標準電文の［表示］ボタンをクリックします。

標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。

19. サービス部品電文の［表示］ボタンをクリックします。

サービス部品電文のフォーマットが表示されます。必要に応じてサービス部品電文のフォーマットを確認してください。

20. 次に示す操作をします。

標準電文の［使う］チェックボックスをチェックした場合

手順 21.～23.を実施して，手順 24.に進みます。

標準電文の［使う］チェックボックスをチェックしない場合

手順 24.に進みます。

21. データ変換定義のファイル名を入力します。

22. ［編集］ボタンをクリックします。

データ変換定義画面が表示されます。

なお，初めて定義する場合は，［ルート要素選択］ダイアログが表示されます。

また，電文フォーマットを変更した場合は，電文フォーマットの変更を反映するかを確認するダイアログが表示されます。電文フォーマットを変更した場合の手順は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.3.2　電文フォーマットを変更した場合の定義手順」を参照してください。

23. 標準電文とサービス部品電文の内容をマッピングします。

24. 応答電文についても同様に，手順 7.から手順 23.までを実施します。

25. サービスアダプタ定義（詳細）タブをクリックします。

26. サービス部品がユーザ認証を必要とする Web サーバ上にあるかどうかで，次に示す操作をします。

ユーザ認証が必要な場合

手順 27.～29.を実施します。

ユーザ認証が不要な場合

以降の手順は不要です。

27. ベーシック認証の［使う］チェックボックスをチェックします。

28. ユーザ名およびパスワードの欄に，ベーシック認証のために必要なユーザ名およびパスワードを入力します。

29. パスワード（確認）欄に，パスワードの欄で入力したパスワードと同じパスワードを入力します。

(6)　クライアント定義ファイルの設定

クライアント定義ファイルは，クライアント（サービスリクエスタ）側の動作を制御するためのファイルです。クライアント定義ファイルは，必要に応じて，次のどちらかの方法で設定します。

- テンプレートファイルを編集する
- 新規に作成してテンプレートファイルに上書き設定する

クライアント定義ファイルの設定は任意です。クライアント定義ファイルを設定しない場合はテンプレートファイルの内容がそのまま使用されます。

(a) 設定手順（テンプレートファイルを編集する場合）

テンプレートファイルを編集する場合の設定手順を示します。

1. サービスアダプタ定義画面（詳細）の［クライアント定義ファイル］で，［編集］ボタンをクリックします。
   クライアント定義ファイルを編集するためのエディタが起動します。
2. エディタ上でクライアント定義ファイルを編集します。
3. Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(b) 設定手順（新規に作成してテンプレートファイルを上書きする場合）

新規に作成してテンプレートファイルを上書きする場合の設定手順を示します。

1. テキストエディタなどで，「c4webcl.properties」というファイル名称でクライアント定義ファイルを作成します。
   「c4webcl.properties」はすべて小文字で入力します。
2. サービスアダプタ定義画面（詳細）の［クライアント定義ファイル］で，［参照］ボタンをクリックし，作成したクライアント定義ファイルを指定します。

(c) 設定内容

クライアント定義ファイルの設定内容は，SOAP1.1 モードを使用する場合と SOAP1.1/1.2 併用モードを使用する場合とでは異なります。それぞれの場合の設定内容を示します。

- SOAP1.1 モードを使用する場合
  クライアント定義ファイルの設定内容については，マニュアル「アプリケーションサーバ SOAP アプリケーション開発の手引」の「10.3　クライアント定義ファイルの設定」を参照してください。
  次の表に，SOAP1.1 モードを使用する場合のクライアント定義ファイル編集時の注意事項を示します。

表 3-1　クライアント定義ファイルの作成に関する注意事項（SOAP1.1 モード）

設定項目	キー名称	設定値 （デフォルト値）	注意事項
多重参照	c4web.common.do_multirefs	false	デフォルト値以外の値を指定した場合の動作は保証されません。
データ型定義	c4web.common.send_xsi_types	true	

設定項目	キー名称	設定値 （デフォルト値）	注意事項
文字参照形式オプション	c4web.common.character_reference	false	デフォルト値以外の値を指定した場合の動作は保証されません。
HTTP セッションの維持	c4web.application.app_maintainsession	false	サービスアダプタの HTTP ヘッダ引き継ぎ機能※を使用して，「JSESSIONID」の Cookie 情報を送受信する場合は，true を設定してください。
SOAP ヘッダの名前修飾チェックオプション	c4web.common.enable_soapheader_check	true	このオプションは，false として動作します。
トレースファイル，アプリケーションログのプリフィックス	c4web.logger.log_file_prefix	サービス ID	値は必ずサービス ID に変更されるため，設定する必要はありません。

注※

サービスアダプタの HTTP ヘッダ引き継ぎ機能を使用して，HTTP ヘッダおよび Cookie 情報を引き継ぐ方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「付録 F　サービスアダプタを利用した HTTP ヘッダおよび Cookie 情報の引き継ぎ」を参照してください。

- SOAP1.1/1.2 併用モードを使用する場合

  クライアント定義ファイルの設定内容については，マニュアル「サービスプラットフォーム リファレンス」の「3.6.2　クライアント定義ファイル」を参照してください。

3.3.2　SessionBean アダプタを定義する

SessionBean アダプタの定義方法について説明します。SessionBean アダプタは，サービスアダプタ定義画面で定義できます。

(1)　サービス部品の電文フォーマット

SessionBean の場合，EAR ファイルから自動作成された電文フォーマット定義ファイルが，サービス部品の電文フォーマット定義ファイルとなります。そのため，サービス部品の電文フォーマットを設定する必要はありません。

(2)　データ変換

サービスリクエスタの電文の形式と，サービスアダプタが呼び出すサービス部品の電文の形式が異なる場合，データ変換が必要です。

要求電文のデータ変換を行う場合，サービスアダプタ定義画面で標準電文にサービスリクエスタから入力される電文を設定し，データ変換定義画面でデータ変換を定義します。応答電文のデータ変換を行う場合，サービス部品から返される電文に標準電文を設定し，データ変換定義画面でデータ変換を定義します。

データ変換の詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7. データ変換の定義」を参照してください。

(3) JNDI 名前空間に登録されている呼び出すサービス部品の名称の指定

サービスアダプタ定義画面では，JNDI 名前空間に登録されている呼び出すサービス部品の名称を指定します。リモート呼び出しとなるようにアドレスを指定してください。サービス部品（SessionBean）のローカル呼び出しはできません。指定例を次に示します。

- 正しい例（myhost はリモートホスト名を表します）

```
corbaname::myhost:900#HITACHI_EJB/SERVERS/J2EEServer/EJB/slsbsmpl/MyAdder
```

- 誤った例

```
corbaname::localhost:900#HITACHI_EJB/SERVERS/J2EEServer/EJB/slsbsmpl/MyAdder

corbaname::127.0.0.1:900#HITACHI_EJB/SERVERS/J2EEServer/EJB/slsbsmpl/MyAdder

corbaname::HITACHI_EJB/SERVERS/J2EEServer/EJB/slsbsmpl/MyAdder
```

JNDI 名前空間に登録されている名称については次の個所を参照してください。

- マニュアル「アプリケーションサーバ 機能解説 基本・開発編(コンテナ共通機能)」の「2. ネーミング管理」
- マニュアル「アプリケーションサーバ 機能解説 基本・開発編(EJB コンテナ)」の「2.5 EJB コンテナの JNDI 名前空間へのリファレンス登録」

(4) ユーザ定義クラス（JAR ファイル）の設定

SessionBean の場合，スタブを利用してサービス部品を呼び出すために，ユーザ定義クラス（JAR ファイル）を設定する必要があります。

なお，サービスアダプタ定義画面（SessionBean の詳細画面）で JAR ファイルを設定するとき，選択するファイルの拡張子は，すべて小文字の「jar」としてください。

また，JAR ファイルとして次に示すファイル名は設定しないでください。

- csmsvcadpdef.jar
- cscmsg_adpejb.jar

上記のファイル名を設定した場合，警告メッセージが表示されます。警告メッセージが表示された場合，上記のファイル名をユーザ定義クラスから削除します。そのあと，上記のファイル名を変更して，再度設定してください。

(5)　RMI-IIOP スタブおよびインターフェースの取得

J2EE サーバの cjgetstubsjar コマンドを使って RMI-IIOP スタブおよびインターフェースを取得します。cjgetstubsjar コマンドの詳細について，マニュアル「アプリケーションサーバ リファレンス コマンド編」の「cjgetstubsjar（アプリケーションの RMI-IIOP スタブおよびインタフェースの取得)」を参照してください。

(6)　サービスアダプタ定義画面での操作

SessionBean アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。
   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービス部品制御情報の［サービス名］，［サービス ID］，および［最大インスタンス数］を必要に応じて編集します。

3. サービス部品制御情報の［アドレス］を指定します。
   ［アドレス］には，JNDI 名前空間に登録されている呼び出すサービス部品の名称を取得して，入力します。JNDI 名前空間に登録されている名称については次の個所を参照してください。
   - マニュアル「アプリケーションサーバ 機能解説 基本・開発編(コンテナ共通機能)」の「2.　ネーミング管理」
   - マニュアル「アプリケーションサーバ 機能解説 基本・開発編(EJB コンテナ)」の「2.5　EJB コンテナの JNDI 名前空間へのリファレンス登録」

4. サービス部品制御情報の［オペレーション］のドロップダウンリストから，編集したいオペレーションを選択します。

5. オペレーション情報に表示されている内容を確認します。

6. サービス部品制御情報の［システム例外をフォルトに変換する］チェックボックスを必要に応じてチェックします。

7. 要求電文について，手順 8.から手順 16.までを実施します。

8. 次に示す操作をします。

   標準電文のフォーマットを指定する場合

   手順 9.～11.を実施して，手順 12.に進みます。

   標準電文のフォーマットを指定しない場合

   手順 12.に進みます。

9. 標準電文の［使う］チェックボックスをチェックします。

10. 標準電文の［参照］ボタンをクリックして，［電文フォーマット］に標準電文のフォーマットを指定します。

なお，外部の XML スキーマを参照している電文フォーマットを設定する場合，必ずルートスキーマに当たるファイルを設定してください。ルートスキーマから参照される外部の XML スキーマファイルは，自動的に取り込まれます。

指定できる電文フォーマットの形式については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.5　XML スキーマの適用範囲」を参照してください。

11. 標準電文の［表示］ボタンをクリックします。

標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。

12. サービス部品電文の［表示］ボタンをクリックします。

サービス部品電文のフォーマットが表示されます。必要に応じてサービス部品電文のフォーマットを確認してください。

13. 次に示す操作をします。

標準電文の［使う］チェックボックスをチェックした場合

手順 14.～16.を実施して，手順 17.に進みます。

標準電文の［使う］チェックボックスをチェックしない場合

手順 17.に進みます。

14. データ変換定義のファイル名を入力します。

15. ［編集］ボタンをクリックします。

データ変換定義画面が表示されます。

なお，初めて定義する場合は，［ルート要素選択］ダイアログが表示されます。

また，電文フォーマットを変更した場合は，電文フォーマットの変更を反映するかを確認するダイアログが表示されます。電文フォーマットを変更した場合の手順は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.3.2　電文フォーマットを変更した場合の定義手順」を参照してください。

16. 標準電文とサービス部品電文の内容をマッピングします。

17. 応答電文についても同様に，手順 8.から手順 16.までを実施します。

18. サービスアダプタ定義（詳細）タブをクリックします。

19. ［追加］ボタンをクリックして，該当するユーザ定義クラスを追加します。

ユーザ定義クラスは，「3.3.2(5)　RMI-IIOP スタブおよびインターフェースの取得」で取得したスタブおよびインターフェースの JAR ファイルを指定します。

(7) クライアント定義ファイルの設定

Web サービスと同様に，クライアント（サービスリクエスタ）側の動作を制御するクライアント定義ファイルを設定します。

クライアント定義ファイルは，必要に応じて，次のどちらかの方法で設定します。

- テンプレートファイルを編集する
- 新規に作成してテンプレートファイルに上書き設定する

クライアント定義ファイルの設定は任意です。クライアント定義ファイルを設定しない場合はテンプレートファイルの内容がそのまま使用されます。

(a) 設定手順（テンプレートファイルを編集する場合）

テンプレートファイルを編集する場合の設定手順を示します。

1. サービスアダプタ定義画面（詳細）の［クライアント定義ファイル］で，［編集］ボタンをクリックします。
   クライアント定義ファイルを編集するためのエディタが起動します。
2. エディタ上でクライアント定義ファイルを編集します。
3. Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(b) 設定手順（新規に作成してテンプレートファイルを上書きする場合）

新規に作成してテンプレートファイルを上書きする場合の設定手順を示します。

1. テキストエディタなどで，「c4webcl.properties」というファイル名称でクライアント定義ファイルを作成します。
   「c4webcl.properties」はすべて小文字で入力します。
2. サービスアダプタ定義画面（詳細）の［クライアント定義ファイル］で，［参照］ボタンをクリックし，作成したクライアント定義ファイルを指定します。

(c) 設定内容

クライアント定義ファイルの設定内容については，マニュアル「アプリケーションサーバ SOAP アプリケーション開発の手引」の「10.3　クライアント定義ファイルの設定」を参照してください。

なお，SessionBean アダプタの場合，クライアント定義ファイルに設定できる項目とできない項目があります。SessionBean アダプタの場合に設定できるクライアント定義ファイルの項目を次の表に示します。

表 3-2　クライアント定義ファイルの設定項目

キー名称	内容	設定可否
c4web.logger.log_level	トレースファイル出力の重要度	○
c4web.logger.aplog_level	アプリケーションログ出力の重要度	○
c4web.logger.aplog_error_record	異常発生時のアプリケーションログ出力	○
c4web.logger.log_file_dir	トレースファイルおよびアプリケーションログの出力先ディレクトリ	○
c4web.logger.log_file_num	トレースファイルの面数	○
c4web.logger.log_file_size	トレースファイルのサイズ	○
c4web.logger.aplog_file_num	アプリケーションログの面数	○
c4web.logger.aplog_file_size	アプリケーションログのサイズ	○
c4web.logger.log_file_prefix	トレースファイル，アプリケーションログのプリフィックス	×※
c4web.common.do_multirefs	多重参照（実行時オプション）	×
c4web.common.send_xsi_types	データ型定義（実行時オプション）	×
c4web.application.app_maintainsession	HTTP セッションの維持（実行時オプション）	×
c4web.application.proxy_host	プロキシサーバのホスト名（実行時オプション）	×
c4web.application.non_proxy_hosts	プロキシサーバを使用しないホスト名群（実行時オプション）	×
c4web.application.proxy_port	プロキシサーバのポート番号（実行時オプション）	×
c4web.application.proxy_user	プロキシサーバの認証ユーザ ID（実行時オプション）	×
c4web.application.proxy_password	プロキシサーバの認証ユーザ ID に対応するパスワード（実行時オプション）	×
c4web.common.enable_soapheader_check	SOAP ヘッダの名前修飾チェックオプション（実行時オプション）	×
c4web.application.socket_write_timeout	クライアントのソケットの書き込みタイムアウト値（実行時オプション）	×
c4web.application.socket_read_timeout	クライアントのソケットの読み込みタイムアウト値（実行時オプション）	×
c4web.application.socket_connect_timeout	クライアントのソケットの接続タイムアウト値（実行時オプション）	×
c4web.attachment.file_body_encoding	ボディエンコーディング種別	×
c4web.attachment.attachment_temp_directory	退避先ディレクトリ	×
c4web.attachment.attachment_file_delete_by_date	不要退避ファイルの経過日数	×
c4web.common.prf_trace_level	性能解析トレース出力オプション	○

（凡例）

○：設定できます。

×：設定できません。設定した場合は無視されます。

注※

値は必ずサービス ID に変更されるため，設定する必要はありません。

3.3.3 MDB（WS-R）アダプタを定義する

MDB（WS-R）アダプタの定義方法について説明します。MDB（WS-R）アダプタは，サービスアダプタ定義画面で定義できます。

(1) キュー情報およびメッセージ種別の取得

サービス部品が MDB（WS-R）の場合，サービス部品のキュー情報およびメッセージ種別を開発者から取得する必要があります。取得した情報は，サービスアダプタを定義するときに使用します。

取得する情報は次のとおりです。

- キュー名称
- 宛先のアドレス（サービスアダプタとサービス部品との間でネットワーク通信をする場合）
- ベーシック認証のためのユーザ名およびパスワード（サービスアダプタとサービス部品との間でネットワーク通信をし，かつベーシック認証を使用する場合）
- メッセージ種別

(2) サービス部品の電文フォーマット

MDB（WS-R）の場合，サービス部品の電文フォーマットを作成して，設定する必要があります。

サービス部品の電文フォーマットの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.3.4　サービス部品電文を作成する（MDB（WS-R または DB キュー）の場合)」を参照してください。

(3) データ変換

サービスリクエスタの電文の形式と，サービスアダプタが呼び出すサービス部品の電文の形式が異なる場合，データ変換が必要です。

データ変換を行う場合，サービスアダプタ定義画面で標準電文にサービスリクエスタから入力される電文を設定し，データ変換定義画面でデータ変換を定義します。

データ変換の詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7. データ変換の定義」を参照してください。

(4)　呼出しの設定

利用するサービス部品の呼出し方法を設定できます。リモート呼出しまたはローカル呼出しを選択できます。リモート呼出しを選択した場合，最大メッセージ数を設定します。また必要に応じて，ベーシック認証を設定します。

(5)　最大メッセージ数の設定

サービス部品の呼出し方法としてリモート呼出しを設定した場合，一度に処理できる電文の数を初期値から変更した数に設定できます。設定できる値は 1～65535 です。

(6)　ベーシック認証の設定

サービス部品の呼出し方法としてリモート呼出しを設定した場合で，MDB（WS-R）のメッセージ転送でユーザ認証を必要とするときは，ベーシック認証（ユーザ名およびパスワード）の設定が必要になります。

注意事項

MDB（WS-R）アダプタでは，ベーシック認証のユーザ名またはパスワードは，16 バイト以内の半角英数字で設定してください。

(7)　サービスアダプタ定義画面での操作

MDB（WS-R）アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。
   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービス部品制御情報の［サービス名］，［サービス ID］，および［最大インスタンス数］を必要に応じて編集します。

3. サービス部品制御情報の［アドレス］を指定します。
   ［アドレス］には，「3.3.3(1)　キュー情報およびメッセージ種別の取得」で取得したキュー名称を入力します。

4. オペレーション情報に表示されている内容を確認します。

5. サービス部品制御情報の［システム例外をフォルトに変換する］チェックボックスを必要に応じてチェックします。

6. 要求電文について次に示す操作をします。

   標準電文のフォーマットを指定する場合

   手順 7.～9.を実施して，手順 10.に進みます。

標準電文のフォーマットを指定しない場合

手順 10.に進みます。

7. 標準電文の［使う］チェックボックスをチェックします。

8. 標準電文の［参照］ボタンをクリックして，［電文フォーマット］に標準電文のフォーマットを指定します。

なお，外部の XML スキーマを参照している電文フォーマットを設定する場合，必ずルートスキーマに当たるファイルを設定してください。ルートスキーマから参照される外部の XML スキーマファイルは，自動的に取り込まれます。

指定できる電文フォーマットの形式については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.5　XML スキーマの適用範囲」を参照してください。

9. 標準電文の［表示］ボタンをクリックします。

標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。

10. サービス部品電文の［参照］ボタンをクリックして，［電文フォーマット］にサービス部品電文のフォーマットを指定します。

11. サービス部品電文の［表示］ボタンをクリックします。

サービス部品電文のフォーマットが表示されます。必要に応じて指定したサービス部品電文のフォーマットを確認してください。

12. 次に示す操作をします。

標準電文の［使う］チェックボックスをチェックした場合

手順 13.～15.を実施して，手順 16.に進みます。

標準電文の［使う］チェックボックスをチェックしない場合

手順 16.に進みます。

13. データ変換定義のファイル名を入力します。

14. ［編集］ボタンをクリックします。

データ変換定義画面が表示されます。

なお，初めて定義する場合は，［ルート要素選択］ダイアログが表示されます。

また，電文フォーマットを変更した場合は，電文フォーマットの変更を反映するかを確認するダイアログが表示されます。電文フォーマットを変更した場合の手順は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.3.2　電文フォーマットを変更した場合の定義手順」を参照してください。

15. 標準電文とサービス部品電文の内容をマッピングします。

16. サービスアダプタ定義（詳細）タブをクリックします。

17. [JMS メッセージ種別] のドロップダウンリストから，JMS メッセージ種別を選択します。

18. [呼出し設定] でサービス部品の呼出方法を選択します。

    [リモート呼出し] ラジオボタンを選択した場合

    手順 19.に進みます。

    [ローカル呼出し] ラジオボタンを選択した場合

    手順 27.に進みます。

19. [最大メッセージ数] に，サービス部品が送信できる電文の数を指定します。

20. [宛先 URL] に「3.3.3(1)　キュー情報およびメッセージ種別の取得」で取得した，宛先のアドレスを入力します。

21. サービス部品がユーザ認証を必要とするかどうかで，次に示す操作をします。

    ユーザ認証が必要な場合

    手順 22.～24.を実施して，手順 25.に進みます。

    ユーザ認証が不要な場合

    手順 25.に進みます。

22. ベーシック認証の［使う］チェックボックスをチェックします。

23. ユーザ名およびパスワードの欄に，ベーシック認証のために必要なユーザ名およびパスワードを入力します。

24. パスワード（確認）欄に，パスワードの欄で入力したパスワードと同じパスワードを入力します。

25. 転送キュー作成先 RD エリア名を必要に応じて入力します。

26. [順序保証] チェックボックスを必要に応じてチェックします。

27. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

3.3.4　MDB（DB キュー）アダプタを定義する

MDB（DB キュー）アダプタの定義方法について説明します。MDB（DB キュー）アダプタは，サービスアダプタ定義画面で定義できます。

(1)　キュー名称の取得

サービス部品が MDB（DB キュー）の場合，サービス部品のキュー名称を開発者から取得する必要があります。キュー名称は，サービスアダプタを定義するときに使用します。

(2)　サービス部品の電文フォーマット

MDB（DB キュー）の場合，サービス部品の電文フォーマットを作成して，設定する必要があります。

サービス部品の電文フォーマットの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.3.4　サービス部品電文を作成する（MDB（WS-R または DB キュー）の場合)」を参照してください。

(3)　データ変換

サービスリクエスタの電文の形式と，サービスアダプタが呼び出すサービス部品の電文の形式が異なる場合，データ変換が必要です。

データ変換を行う場合，サービスアダプタ定義画面で標準電文にサービスリクエスタから入力される電文を設定し，データ変換定義画面でデータ変換を定義します。

データ変換の詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7. データ変換の定義」を参照してください。

(4)　サービスアダプタ定義画面での操作

MDB（DB キュー）アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。

   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービス部品制御情報の［サービス名］，［サービス ID］，および［最大インスタンス数］を必要に応じて編集します。

3. サービス部品制御情報の［アドレス］を指定します。

   ［アドレス］には，共用キューを使用する場合の登録先キュー名を入力します。共用キューを使用する場合の登録先キュー名については，マニュアル「Reliable Messaging」の「8.3.19　hrmmkque（送信用共用キューの作成)」を参照してください。

4. サービス部品制御情報の［オペレーション］のドロップダウンリストから，編集したいオペレーションを選択します。

5. オペレーション情報に表示されている内容を確認します。

6. サービス部品制御情報の［システム例外をフォルトに変換する］チェックボックスを必要に応じてチェックします。

7. 要求電文について次に示す操作をします。

   標準電文のフォーマットを指定する場合

   手順 8.～10.を実施して，手順 11.に進みます。

標準電文のフォーマットを指定しない場合

手順 11.に進みます。

8. 標準電文の［使う］チェックボックスをチェックします。

9. 標準電文の［参照］ボタンをクリックして，［電文フォーマット］に標準電文のフォーマットを指定します。

なお，外部の XML スキーマを参照している電文フォーマットを設定する場合，必ずルートスキーマに当たるファイルを設定してください。ルートスキーマから参照される外部の XML スキーマファイルは，自動的に取り込まれます。

指定できる電文フォーマットの形式については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.5　XML スキーマの適用範囲」を参照してください。

10. 標準電文の［表示］ボタンをクリックします。

標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。

11. サービス部品電文の［参照］ボタンをクリックして，［電文フォーマット］にサービス部品電文のフォーマットを指定します。

12. サービス部品電文の［表示］ボタンをクリックします。

サービス部品電文のフォーマットが表示されます。必要に応じて指定したサービス部品電文のフォーマットを確認してください。

13. 次に示す操作をします。

標準電文の［使う］チェックボックスをチェックした場合

手順 14.～16.を実施して，手順 17.に進みます。

標準電文の［使う］チェックボックスをチェックしない場合

手順 17.に進みます。

14. データ変換定義のファイル名を入力します。

15. ［編集］ボタンをクリックします。

データ変換定義画面が表示されます。

なお，初めて定義する場合は，［ルート要素選択］ダイアログが表示されます。

また，電文フォーマットを変更した場合は，電文フォーマットの変更を反映するかを確認するダイアログが表示されます。電文フォーマットを変更した場合の手順は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.3.2　電文フォーマットを変更した場合の定義手順」を参照してください。

16. 標準電文とサービス部品電文の内容をマッピングします。

17. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

3.3.5 DB アダプタを定義する

DB アダプタの定義方法について説明します。

(1) SQL オペレーション定義ファイルの作成

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

SQL オペレーション定義ファイルの作成方法を次に説明します。

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

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

なお，XML 名前空間名は使用できません。

作成する XML 文書の詳細を説明します。

●ファイル名

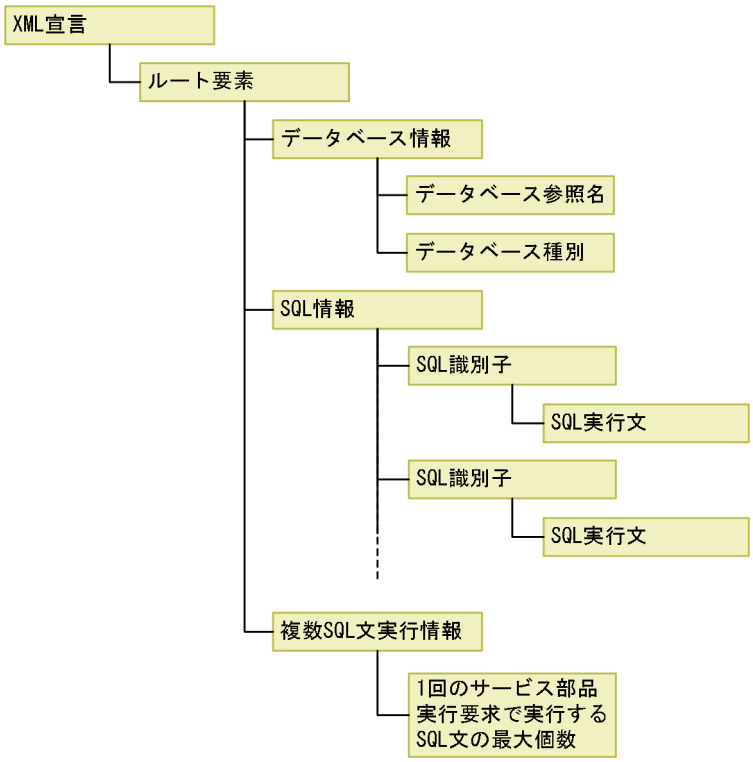
SQL オペレーション定義ファイルのファイル名は，次の形式で指定します。

```
csa_sql_<任意の名称>.xml
```

●構造

SQL オペレーション定義ファイルの構造を次に示します。

図 3-2　SQL オペレーション定義ファイルの構造



●SQL オペレーション定義ファイルの形式

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

```
<?xml version="XMLのバージョン" encoding="文字コード"?>
<DBadapter_SQL_OPERATION dba_separate_transaction="トランザクション分離オプション" sqlope
ration_chk="SQL命令チェックオプション">
    <DATABASE_DATA>
        <DB_NAME dynamic="接続先の動的変更オプション">データベース参照名</DB_NAME>
        <DB_TYPE>データベース種別</DB_TYPE>
    </DATABASE_DATA>
    <SQL_DATA encoding="文字バイナリデータのエンコード・デコード形式">
        <SQL識別子 out_maxOccurs="検索結果出力最大件数">
            SQL命令 <引数要素名 dba_inf="引数の種別" data_type="データ型"/>
        </SQL識別子>
            :
    </SQL_DATA>
    <MULTI_SQL_DATA>
        <MAX_SQL_NO>1回のサービス部品実行要求で実行するSQL文の最大個数</MAX_SQL_NO>
    </MULTI_SQL_DATA>
</DBadapter_SQL_OPERATION>
```

SQL オペレーション定義ファイルで定義する内容について次に説明します。

XML 宣言

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

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

ルート要素（要素名：DBadapter_SQL_OPERATION）

SQL オペレーション定義ファイルのルート要素です。

```
<DBadapter_SQL_OPERATION dba_separate_transaction="Y|N" sqloperation_chk＝"Y/N">
    :
</DBadapter_SQL_OPERATION>
```

- dba_separate_transaction

  DB アダプタのトランザクションをビジネスプロセスとは別に開始するかどうかを指定します。トランザクションを分離する場合は Y を，分離しない場合は N を指定します。指定を省略した場合は，HCSC サーバランタイム定義ファイルの「dba-separate-transaction」の値が有効となります。この属性と，HCSC サーバランタイム定義ファイルに「dba-separate-transaction」の両方が指定してある場合は，この属性の指定値が優先されます。

  なお，この属性に Y を指定する場合は，事前に DB アダプタのアプリケーション統合属性ファイルの trans-attribute 要素の指定値を「Required」から「RequiresNew」に変更してください。

- sqloperation_chk

  DB アダプタで実行する SQL 命令の制限をするかどうかを指定します。SQL 命令を SELECT と INSERT に制限する場合は Y を，SQL 命令で SELECT と INSERT に加えて UPDATE と DELETE を実行する場合は N を指定します。指定を省略した場合は，Y が指定されているときと同じ動作となります。

データベース情報（要素名：DATABASE_DATA）

下位の要素（DB_NAME および DB_TYPE）で，データベースの情報を指定します。

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

データベース参照名（要素名：DB_NAME）

DB Connector のデータベース参照名を指定します。

DB アダプタ実行時に，ここで指定した値を元にアプリケーション統合属性ファイルのリソース参照を検索するため，アプリケーション統合属性ファイルの res-ref-name 要素指定値と同じ値を指定してください。

dynamic 属性に Y が設定されている場合はこの要素の値は省略できます。また，この要素の値がすでに設定されていた場合は指定を無視します。

```
<DB_NAME dynamic="Y|N">データベース参照名</DB_NAME>
```

- dynamic

  接続先を動的に変更するかどうかを指定します。

  接続先を動的に変更する場合は Y を，動的に変更しない場合は N を指定します。指定を省略した場合は，N が指定されたときと同じ動作になります。

データベース種別（要素名：DB_TYPE）

データベースの種別として次のどれかを指定します。

- HIRDB

  DBMS として HiRDB を使用する場合（DB Connector として DBConnector_DABJ_CP/XA.rar を使用する場合）に指定します。

- HIRDB-TYPE4

  DBMS として HiRDB を使用する場合（DB Connector として DBConnector_HiRDB_Type4_CP/XA.rar を使用する場合）に指定します。

- ORACLE

  DBMS として Oracle を使用する場合（DB Connector として DBConnector_DABJ_CP/XA.rar を使用する場合）に指定します。

- ORACLE-THIN

  DBMS として Oracle を使用する場合（DB Connector として DBConnector_Oracle_CP/XA.rar を使用する場合）に指定します。

- MYSQL

  DBMS として MySQL を使用する場合（DB Connector として DBConnector_MySQL_CP.rar を使用する場合）に指定します。J2EE サーバの互換モードが推奨モードの場合は，Amazon Aurora MySQL も使用できます。

- POSTGRESQL

  DBMS として PostgreSQL を使用する場合（DB Connector として DBConnector_PostgreSQL_CP.rar を使用する場合）に指定します。J2EE サーバの互換モードが推奨モードの場合は，Amazon Aurora PostgreSQL も使用できます。

```
<DB_TYPE>データベース種別</DB_TYPE>
```

SQL 情報（要素名：SQL_DATA）

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

```
<SQL_DATA encoding="hexBinary|base64Binary">
    SQL識別子
        SQL実行文
</SQL_DATA>
```

- encoding

  バイナリデータのエンコード・デコード形式を指定します。

  hexBinary 形式でエンコードまたはデコードする場合は hexBinary を，base64Binary 形式でエンコードまたはデコードする場合は base64Binary を指定します。指定を省略した場合は，hexBinary が指定されたときと同じ動作になります。

  なお，この属性が有効になるのは，data_type 属性に次のデータ型が指定された場合だけです。

  ・HiRDB の場合：LONGVARBINARY

  ・Oracle の場合：VARBINARY，LONGVARBINARY，BLOB

  ・MySQL の場合：BINARY，VARBINARY，LONGBARBINARY

  ・PostgreSQL の場合：BINARY

注意事項

バイナリデータをデータ変換する場合，SQL オペレーション定義ファイルの encoding 属性に指定した値とバイナリフォーマット定義ファイルに指定したエンコード・デコード形式を合わせてください。形式が異なる場合の動作は保証されません。

参考

09-50 より前の環境から DB アダプタをバージョンアップした場合に，エンコード・デコード形式を base64binary に変更する手順を次に示します。

1. 開発環境でバイナリフォーマット定義ファイルを再作成します。
2. SQL オペレーション定義ファイルを修正（エンコード・デコード形式を設定）します。
3. 修正した SQL オペレーション定義ファイルを使用して csamkxmls コマンドで XML スキーマを生成します。
4. XML スキーマとバイナリフォーマット定義ファイルを開発環境で DB アダプタに設定します。
5. 開発環境でマッピング定義を再設定します。
6. 開発環境で修正した SQL オペレーション定義ファイルを DB アダプタに設定します。
7. DB アダプタを再配備します。

SQL 識別子（要素名：任意((1～256 文字)))

実行する SQL を識別するための要素です。この要素を省略した場合，SQL 文を実行しません。下位の引数要素で SQL 実行文を指定します。属性として out_maxOccurs を指定できます。また，実行する SQL 命令の数にあわせて複数回指定できます。

なお，SQL 識別子の名称として「DBadapter」および「DBA_MULTI_SQL」は指定できません。

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

- out_maxOccurs

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

SQL 実行文

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

なお，SQL 文の終わりを示すセミコロン（;）は必要ありません。

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

なお，SQL 実行文の先頭は，「SELECT△」または次のどれかを指定します（△は半角スペース）。

- 「INSERT△」
- 「UPDATE△」
- 「DELETE△」

ただし，「SELECT△」，または「INSERT△」，「UPDATE△」，「DELETE△」のどれかの直後に引数要素がある場合，「△」（半角スペース）は省略できます。

指定内容を次に示します。

SQL 命令

指定できる SQL 命令は SELECT，または INSERT，UPDATE，DELETE です。

SELECT で検索できるデータ型について，次の表に示します。

表 3-3　SELECT で検索できるデータ型（HiRDB の場合）

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

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

（凡例）

○：SELECT で検索できます。

−：SELECT で検索できません。

注

〔 〕で囲まれた部分は省略できることを意味します。

表 3-4　SELECT で検索できるデータ型（Oracle の場合）

Oracle のデータ型	検索の可否
	Oracle JDBC Thin Driver の場合
VARCHAR2	○
NVARCHAR2	○
NUMBER	○
LONG	○
DATE	○※
BINARY_FLOAT	−
BINARY_DOUBLE	−
TIMESTAMP	−
TIMESTAMP WITH TIME ZONE	−
TIMESTAMP WITH LOCAL TIME ZONE	−
INTERVAL YEAR TO MONTH	−
INTERVAL DAY TO SECOND	−
RAW	○
LONG RAW	○
ROWID	○
UROWID	−
CHAR	○

Oracle のデータ型	検索の可否
	Oracle JDBC Thin Driver の場合
NCHAR	○
CLOB	○
NCLOB	○
BLOB	○
BFILE	－
ユーザ定義型（オブジェクト型）	－
ユーザ定義型（REF データ型）	－
ユーザ定義型（VARRAY）	－
ユーザ定義型（ネストした表）	－

（凡例）

○：SELECT で検索できます。

－：SELECT で検索できません。

注※

サポート対象バージョンの Oracle JDBC Thin Driver を利用する場合，論理 J2EE サーバの定義で JVM の起動パラメタとして次の値を設定する必要があります。

-Doracle.jdbc.mapDateToTimestamp=false

起動パラメタの設定の詳細については，マニュアル「アプリケーションサーバ 運用管理ポータル操作ガイド」の「10.8.23　起動パラメタの設定（J2EE サーバ）」を参照してください。

表 3-5　SELECT で検索できるデータ型（MySQL の場合）

MySQL のデータ型	検索の可否
TINYINT	○
SMALLINT	△
MEDIUMINT	△
INT[EGER]	○
BIGINT	○
DECIMAL	○
NUMERIC	△
FLOAT	△
DOUBLE	○
BIT	○※1
DATE	○
TIMESTAMP	○

MySQL のデータ型	検索の可否
TIME	○
CHAR	○
VARCHAR	○
VARBINARY	○
TINYBLOB	△
BLOB	△
MEDIUMBLOB	△
LONGBLOB	△
TINYTEXT	△
TEXT	△
MEDIUMTEXT	△
LONGTEXT	△
DATETIME	△
ENUM	△
SET	△
JSON	△
BINARY	○
YEAR	－※2
WKT（Well-KnownText）形式	－※2
WKB（Well-KnownBinary）形式	－※2

（凡例）

○：SELECT で検索できます。

△：SELECT で検索できますが，data_type は別の文字列を指定します。指定できる data_type は，「表 3-9 data_type に指定できるデータ型（MySQL の場合）」を参照してください。

－：SELECT で検索できません。

注※1

サイズが 1 の場合は使用できますが，1 より大きなサイズの場合は使用できません。

注※2

SELECT で検索してもエラーにはなりません。ただし，検索結果は保証できません。

表 3-6　SELECT で検索できるデータ型（PostgreSQL の場合）

PostgreSQL のデータ型	検索の可否
smallint	○

PostgreSQL のデータ型	検索の可否
integer	○
bigint	○
numeric	○
decimal	△
real	○
double precision	△
smallserial	△
serial	△
bigserial	△
varchar	○
character varying	△
char	○
character	△
text	△
bytea	△
timestamp [without time zone]	○
timestamp with time zone	△
date	○
time [without time zone]	○
time with time zone	△
boolean	△
bit	―※
money	―※
interval [fields] [(p)]	―
point	―
line	―
lseg	―
box	―
path	―
polygon	―
circle	―

PostgreSQL のデータ型	検索の可否
cidr	－
inet	－
macaddr	－
macaddr8	－
bit varying (n)	－
tsvector	－
tsquery	－
uuid	－
xml	－
json	－
jsonb	－
pg_lsn	－
txid_snapshot	－
列挙型	＿※
ドメイン型	＿※

（凡例）

○：SELECT で検索できます。

△：SELECT で検索できますが，data_type は別の文字列を指定します。指定できる data_type は，「表 3-10 data_type に指定できるデータ型（PostgreSQL の場合）」を参照してください。

－：SELECT で検索できません。

注※

SELECT で検索してもエラーにはなりません。ただし，検索結果は保証できません。

●ポイント

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

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

```
SELECT TO_CHAR(c_ts) FROM 表名
```

引数要素名

任意の引数要素の名称を指定します。複数の引数要素を同一名称で指定した場合，同じ引数として扱われます。同一名称の引数要素で，それぞれの属性 dba_inf および data_type に異なる値を設定した場合はエラーになります。

dba_inf ＝"引数の種別"

引数の種別を指定します。

引数の種別の種類として指定できる内容を次に示します。

- table

  引数がデータベースに実在するテーブルの名称の場合に指定します。引数として利用するデータは string 型です。

- column

  引数がデータベースに実在する列の名称の場合に指定します。引数として利用するデータは string 型です。

- data

  引数がテーブルおよび列の名称以外のデータの場合に指定します。この値を指定した場合，引数として利用するデータのデータ型を属性 data_type に指定する必要があります。また，引数として利用するデータを表 3-3 および表 3-4 に示す Java のデータ型またはクラスに変換して，SQL 文に埋め込みます。引数として利用するデータを変換できない場合は，エラーとなります。

  なお，この値を指定した場合，引数要素を IN パラメタ（?パラメタまたはバインド変数）として扱います。SQL 文中に IN パラメタを指定できる個所は，利用するデータベースの仕様に従います。指定できない個所に IN パラメタを定義した場合，エラーになります。なお，SQL の実行時に，引数要素は「?」に置き換えられて実行されます。

- preset

  引数がテーブルおよび列の名称以外のデータの場合に指定します。この値を指定した場合，引数として利用するデータをそのまま SQL 文に埋め込みます。引数として利用するデータは string 型です。

data_type="データ型"

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

表 3-7　data_type に指定できるデータ型（HiRDB の場合）

HiRDB のデータ型	data_type に指定できるデータ型	引数として利用するデータの変換先のデータ型およびクラス
INT〔EGER〕	INTEGER	int
SMALLINT	SMALLINT	short
〔LARGE〕 DEC〔IMAL〕 NUMERIC	DECIMAL	java.math.BigDecimal
FLOAT DOUBLE PRECISION	FLOAT	double
SMALLFLT REAL	REAL	float
CHAR〔ACTER〕	CHAR	java.lang.String
NCHAR NATIONAL CHAR		

HiRDB のデータ型	data_type に指定できるデータ型	引数として利用するデータの変換先のデータ型およびクラス
MCHAR	CHAR	java.lang.String
VARCHAR CHAR〔ACTER〕VARYING	VARCHAR	java.lang.String
NVARCHAR NATIONAL CHAR〔ACTER〕VARYING NCHAR VARYING		
MVARCHAR		
DATE	DATE	java.sql.Date
TIME	TIME	java.sql.Time
TIMESTAMP	TIMESTAMP	java.sql.Timestamp
BLOB BINARY LARGE OBJECT	LONGVARBINARY	byte[]
BINARY	LONGVARBINARY	byte[]

注

〔 〕で囲まれた部分は省略できることを意味します。

表 3-8　data_type に指定できるデータ型（Oracle の場合）

Oracle のデータ型	data_type に指定できるデータ型	引数として利用するデータの変換先のデータ型およびクラス
LONG	LONGVARCHAR	java.io.Reader
NUMBER	NUMERIC	java.math.BigDecimal
VARCHAR2	VARCHAR	java.lang.String
NVARCHAR2	−	
ROWID	CHAR	
CHAR	CHAR	
NCHAR	−	
CLOB	CLOB※1	• Oracle JDBC Thin Driver の場合 java.io.Reader
NCLOB	−	
DATE	DATE※2	java.sql.Date
	TIME※3	java.sql.Time
	TIMESTAMP※4	java.sql.Timestamp
RAW	VARBINARY	byte[]

Oracle のデータ型	data_type に指定できるデータ型	引数として利用するデータの変換先のデータ型およびクラス
LONG RAW	LONGVARBINARY	byte[]
BLOB	BLOB	byte[]

（凡例）

－：指定できません。

注※1

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

注※2

設定する値の形式が「yyyy-mm-dd」の場合に指定します。

注※3

設定する値の形式が「hh:mm:ss」の場合に指定します。

注※4

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

表 3-9　data_type に指定できるデータ型（MySQL の場合）

MySQL のデータ型	data_type に指定できるデータ型
TINYINT	TINYINT
SMALLINT，MEDIUMINT，INT[EGER]	INTEGER
INT[EGER] (UNSIGNED)	BIGINT
BIGINT	BIGINT
BIGINT(UNSIGNED)	DECIMAL
DECIMAL NUMERIC	DECIMAL
FLOAT	REAL
DOUBLE	DOUBLE
BIT	BIT
DATE	DATE
TIMESTAMP，DATETIME	TIMESTAMP
TIME	TIME
CHAR，ENUM，SET	CHAR
VARCHAR，TINYTEXT	VARCHAR
BINARY	BINARY
VARBINARY，TINYBLOB	VARBINARY

MySQL のデータ型	data_type に指定できるデータ型
BLOB，MEDIUMBLOB，LONGBLOB	LONGVARBINARY
TEXT，MEDIUMTEXT，LONGTEXT，JSON	LONGVARCHAR
WKT（Well-KnownText）形式	－
WKB（Well-KnownBinary）形式	－

（凡例）

－：指定できません。

表 3-10　data_type に指定できるデータ型（PostgreSQL の場合）

PostgreSQL のデータ型	data_type に指定できるデータ型
bigint，bigserial	BIGINT
bit	－
bit varying [(n)]	－
boolean	BOOLEAN
box	－
bytea	BINARY
character [(n)]	CHAR
character varying [(n)]	VARCHAR
cidr	－
circle	－
date	DATE
double precision	DOUBLE
inet	－
integer	INTEGER
interval [fields] [(p)]	－
json	－
jsonb	－
line	－
lseg	－
macaddr	－
macaddr8	－
money	－

PostgreSQL のデータ型	data_type に指定できるデータ型
numeric [(p, s)]	DECIMAL
path	−
pg_lsn	−
point	−
polygon	−
real	REAL
smallint	SMALLINT
smallserial	SMALLINT
serial	INTEGER
text	VARCHAR
time [(p)] [without time zone]	TIME
time [(p)] with time zone	TIME
timestamp [(p)] [without time zone]	TIMESTAMP
timestamp [(p)] with time zone	TIMESTAMP
tsquery	−
tsvector	−
txid_snapshot	−
uuid	−
xml	−

（凡例）

−：指定できません。

複数 SQL 文実行情報（要素名：MULTI_SQL_DATA）

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

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

実行する SQL 文の最大個数（要素名：MAX_SQL_NO）

実行する SQL 文の最大個数（サービスリクエスタからの要求電文の DBA_MULTI_SQL 要素の最大出現回数）を，1～2,147,483,647 の範囲内で指定します。省略した場合の値は「1024」になります。

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

●作成例

SQL オペレーション定義ファイルの作成例を次に示します。

作成例の条件

データベース参照名：DB_SERVER1

スキーマ名：DBA

テーブル名：受注表

テーブル構成：

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

実行する SQL の形式：

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

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

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

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

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

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

(2)　電文フォーマットの作成

DB アダプタの場合，WSDL が生成されないため，サービス部品の電文フォーマットも自動生成されません。そのため，サービス部品の電文フォーマット（DB アダプタ用の XML フォーマット定義ファイル）を作成して，設定する必要があります。

(a)　DB アダプタ用の XML フォーマット定義ファイルの作成

DB アダプタ用の XML フォーマット定義ファイルを作成するには，「(1)　SQL オペレーション定義ファイルの作成」で作成した SQL オペレーション定義ファイルの名称を-o オプションに指定して csamkxmls コマンドを実行します。

●コマンドの実行例

csamkxmls コマンドの実行形式を次に示します。

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

csamkxmls コマンドの詳細については，マニュアル「サービスプラットフォーム リファレンス」の「csamkxmls（DB アダプタ用の XML フォーマット定義ファイルの生成)」を参照してください。

注意事項

csamkxmls コマンドを実行したとき，SQL オペレーション定義ファイルのファイルサイズが大きいと，「java.lang.OutOfMemoryError」のメッセージが表示されることがあります。

このエラーが表示された場合，環境変数 HCSCDBA_XMX に，現在の JavaVM のメモリ割り当てサイズよりも大きい値を設定して，再度コマンドを実行してください。

環境変数 HCSCDBA_XMX には，JavaVM のヒープサイズ（-Xmx）に設定される値（単位：Mbyte）を指定します。省略した場合，JavaVM のヒープサイズは Java コマンドを実行する環境に依存します。

設定例

```
set HCSCDBA_XMX=64
```

注意事項

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

データ変換の定義については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.　データ変換の定義」を参照してください。サービスアダプタ定義画面については，マニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

(b)　DB アダプタ用の XML フォーマット定義ファイルの作成例

DB アダプタ用の XML フォーマット定義ファイルの作成例を次に示します。

図 3-3　（例 1）SELECT 文を使用し，WHERE 句の値を引数化したい場合

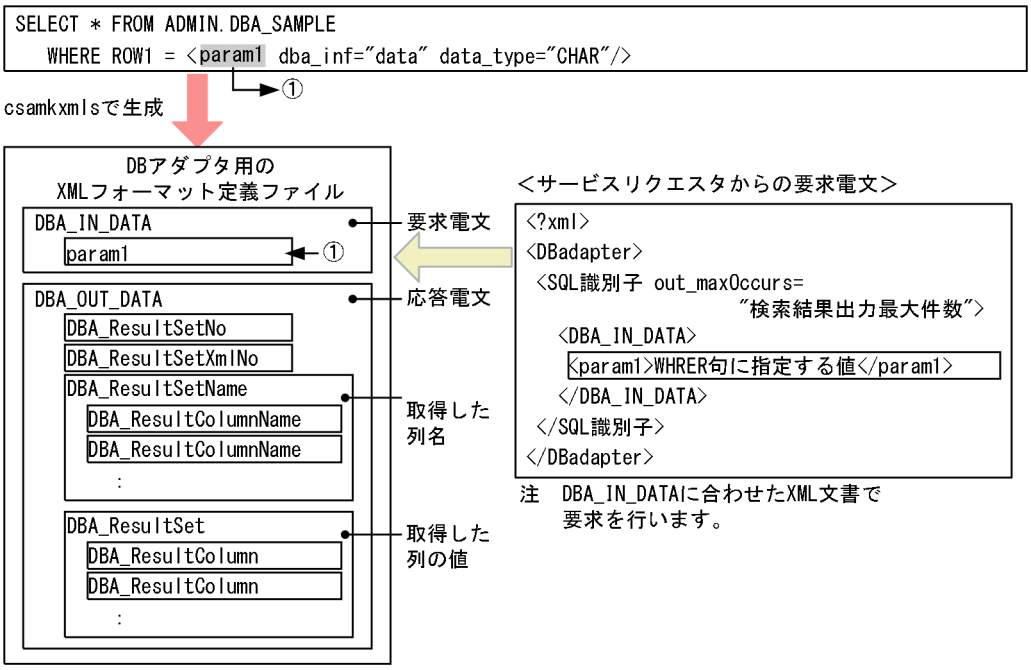


図 3-4　（例 2）SELECT 文を使用し，取得する列名も引数化したい場合

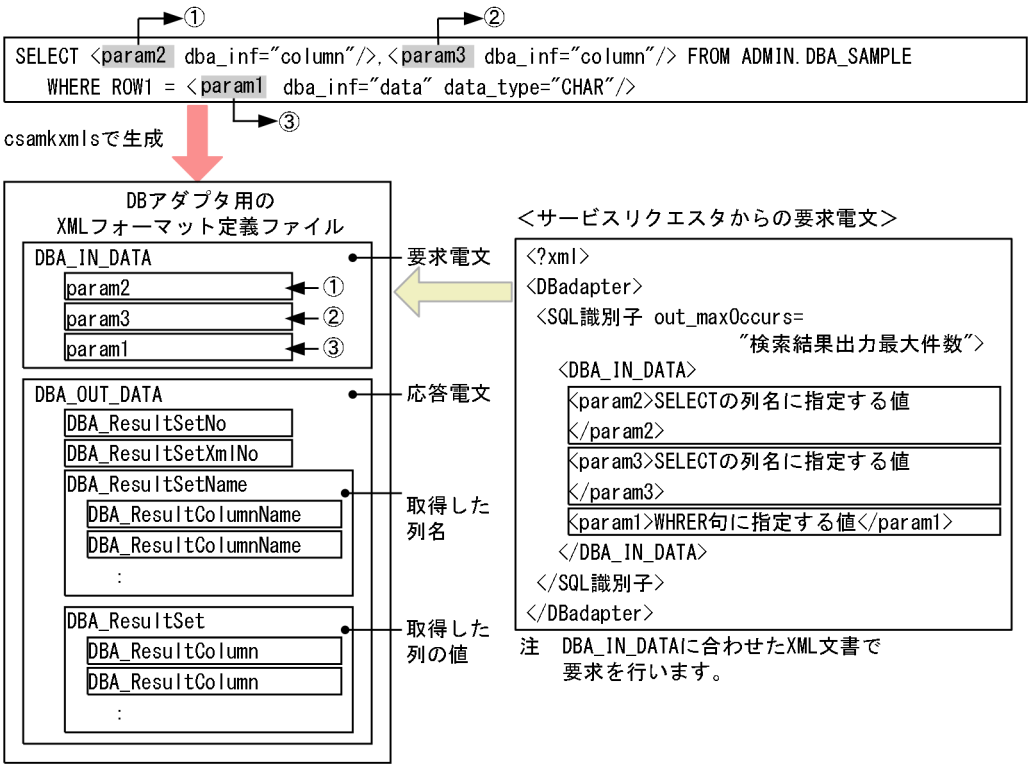
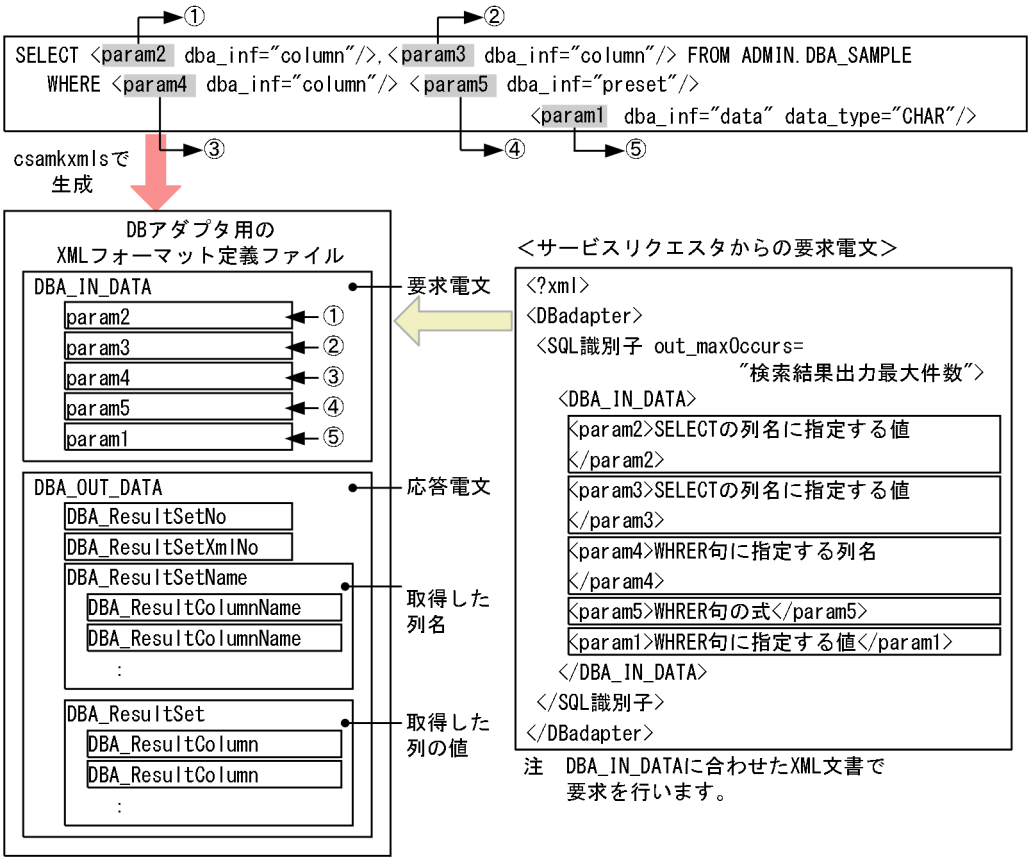


図 3-5　（例 3）SELECT 文を使用し，WHERE 句の列名や式も引数化したい場合



(3)　サービスアダプタ定義画面での操作

DB アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。

   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービス部品制御情報の［サービス名］および［サービス ID］を必要に応じて編集します。

   サービス名は，DB アダプタを追加したときに入力した名称から異なる名称に変更したい場合に入力してください。

   サービス ID は，運用管理者が HCSC サーバの操作（オペレーション）を行うときに必要になる識別情報です。サービス名との関連がわかるような形で命名することを推奨します。

3. サービス部品制御情報の［追加］ボタンをクリックして，オペレーションを追加します。

   オペレーション名は，サービスリクエスタから HCSC サーバを呼ぶときに必要になる識別情報です。

4. サービス部品制御情報の［システム例外をフォルトに変換する］チェックボックスを必要に応じてチェックします。

5. オペレーション情報の［通信モデル］のドロップダウンリストから，「同期」または「非同期」を選択します。

応答が必要な場合は「同期」を選択してください。

6. 要求電文について，手順 7.から手順 16.までを実施します。

7. 次に示す操作をします。

標準電文のフォーマットを指定する場合

手順 8.～10.を実施して，手順 11.に進みます。

標準電文のフォーマットを指定しない場合

手順 11.に進みます。

8. 標準電文の［使う］チェックボックスをチェックします。

9. 標準電文の［参照］ボタンをクリックして，［電文フォーマット］に標準電文のフォーマットを指定します。

指定できる電文フォーマットの形式については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.5　XML スキーマの適用範囲」を参照してください。なお，外部の XML スキーマを参照している電文フォーマットは指定できません。

10. 標準電文の［表示］ボタンをクリックします。

標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。

11. サービス部品電文の［参照］ボタンをクリックして，［電文フォーマット］にサービス部品電文のフォーマットを指定します。

ここには，作成した DB アダプタ用の XML フォーマット定義ファイルを指定します。

［電文フォーマット］に指定できる電文種別は，XML 電文だけです。

12. サービス部品電文の［表示］ボタンをクリックします。

サービス部品電文のフォーマットが表示されます。必要に応じて指定したサービス部品電文のフォーマットを確認してください。

13. 必要に応じて，データ変換定義を行います。

次に示す操作をします。

標準電文の［使う］チェックボックスをチェックした場合

手順 14.～16.を実施して，手順 17.に進みます。

標準電文の［使う］チェックボックスをチェックしない場合

手順 17.に進みます。

14. データ変換定義のファイル名を入力します。

15. ［編集］ボタンをクリックします。

データ変換定義画面が表示されます。

なお，初めて定義する場合は，［ルート要素選択］ダイアログが表示されます。

また，電文フォーマットを変更した場合は，電文フォーマットの変更を反映するかを確認するダイアログが表示されます。電文フォーマットを変更した場合の手順は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.3.2　電文フォーマットを変更した場合の定義手順」を参照してください。

16. 標準電文とサービス部品電文の内容をマッピングします。

17. 手順 5.の［通信モデル］で「同期」を選択した場合，応答電文についても同様に，手順 7.から手順 16.までを実施します。

18. サービスアダプタ定義（詳細）タブをクリックします。

19. サービスアダプタ（EJB-JAR ファイル）および独自定義ファイルの名称を確認します。

    サービスアダプタ（EJB-JAR ファイル）の名称が「cscdba_ejb.jar」，独自定義ファイルの名称が「cscadapter_property.xml」であることを確認してください。

    ユーティリティクラス（JAR ファイル）の設定は不要です。

20. 独自定義ファイルの［追加］ボタンをクリックして，次のファイルを追加します。

    - SQL オペレーション定義ファイル

      [csa_sql_<任意の名称>.xml]

(4)　アプリケーション統合属性ファイルの編集

サービスプラットフォームが提供するテンプレートファイルを利用して，アプリケーション統合属性ファイルを編集します。

アプリケーション統合属性ファイルの編集手順を示します。

1. サービスアダプタ定義画面（詳細）の［独自定義ファイル］で「cscadapter_property.xml」を選択し，［編集］ボタンをクリックします。

   アプリケーション統合属性ファイルを編集するためのエディタが起動します。

2. エディタ上でアプリケーション統合属性ファイルを編集します。

   アプリケーション統合属性ファイルによるプロパティ設定については，マニュアル「アプリケーションサーバ アプリケーション設定操作ガイド」の「9.2　アプリケーション統合属性ファイルによるプロパティ設定」を参照してください。アプリケーション統合属性ファイルの詳細については，マニュアル「アプリケーションサーバ リファレンス 定義編(アプリケーション／リソース定義)」の「3.1　アプリケーション統合属性ファイル」を参照してください。

   アプリケーション統合属性ファイルに設定する項目について，次の表に示します。

表 3-11　アプリケーション統合属性ファイルの設定項目

タグ	説明	設定内容	変更可否	設定例
<hitachi-application-all-property>	ルートタグ	−	否	−
<ejb-jar>	EJB に関する情報の定義の開始タグ	−	否	−
<hitachi-session-bean-property>	SessionBean の属性の定義の開始タグ	−	否	−
<resource-ref>	リソース参照についての定義の開始タグ※1	−	否	−
<res-ref-name>	リソース参照名	SQL オペレーション定義ファイルの<DB_NAME>タグに指定したデータベース参照名	可	DB_SERVER1
<res-sharing-scope>	参照したリソースを共有するかどうかの指定	DB アダプタが参照する DB Connector を共有するかどうかの指定	可	Shareable
<linked-to>	データソース表示名	DB Connector のプロパティ定義（display-name）に指定したリソースアダプタの名称	可	DB_Connector_for_HiRDB_Type4_XA や DB_Connector_for_Oracle_XA など※2 display-name を変更した場合は変更後の名称。
</resource-ref>	リソース参照についての定義の終了タグ	−	否	−
<container-transaction>	コンテナのトランザクションについての定義の開始タグ	−	否	−
<trans-attribute>	メソッドに割り当てるトランザクション属性	トランザクションを引き継ぐ Required	可	Required※3
</container-transaction>	コンテナのトランザクションについての定義の終了タグ	−	否	−
<session-runtime>	runtime の定義の開始タグ	−	否	−
<stateless>	stateless の定義の開始タグ	−	否	−

タグ	説明	設定内容	変更可否	設定例
<pooled-instance>	プール内のインスタンスについての定義の開始タグ	−	否	−
<minimum>	プール内のインスタンスの最小数	DB アダプタのインスタンスの最小数	可	0 または 1≦minimum≦maximum
<maximum>	プール内のインスタンスの最大数	DB アダプタのインスタンスの最大数	可	0（無制限）または 1≦maximum≦maximum-sessions
</pooled-instance>	プール内のインスタンスについての定義の終了タグ	−	否	−
</stateless>	stateless の定義の終了タグ	−	否	−
</session-runtime>	runtime の定義の終了タグ	−	否	−
</hitachi-session-bean-property>	SessionBean の属性の定義の終了タグ	−	否	−
</ejb-jar>	EJB に関する情報の定義の終了タグ	−	否	−
</hitachi-application-all-property>	ルートタグ	−	否	−

（凡例）

−：変更できません。

注※1

SQL オペレーション定義ファイルの dynamic 属性に Y が設定されている場合は，このタグおよびその子要素の定義を省略してください。

注※2

display-name を変更した場合は，変更後の名称を指定してください。

注※3

ビジネスプロセスとは別に DB アダプタのトランザクションを開始する場合は，HCSC サーバランタイム定義ファイルの dba-separate-transaction プロパティに true，または SQL オペレーション定義ファイルの「dba_separate_transaction」属性に Y を設定し，かつアプリケーション統合属性ファイルの<trans-attribute>タグに RequiresNew を設定してください。

注意事項

表 3-11 で示した設定・変更できる項目以外は，設定・変更しないでください。

3. Eclipse のメニューから［ファイル］−［保存］を選択し，定義内容を保存します。

(5) データ変換

要求電文のデータ変換を行う場合，サービスアダプタ定義画面で標準電文にサービスリクエスタから入力される電文を設定し，データ変換定義画面でデータ変換を定義します。応答電文のデータ変換を行う場合，サービス部品から返される電文に標準電文を設定し，データ変換定義画面でデータ変換を定義します。

(a) データ変換が必要なケース

サービスリクエスタの電文の形式と，DB アダプタ電文の形式が異なる場合，データ変換が必要です。

次のような場合は，標準電文のフォーマットを指定してデータ変換を定義します。

- DB アダプタに指定した電文フォーマット（XML フォーマット定義ファイル）と異なる電文フォーマットで DB アダプタを呼びたい場合
- DB アダプタに指定した電文フォーマット（XML フォーマット定義ファイル）と異なる電文フォーマットで DB アダプタからの応答を受け取りたい場合

(b) データ変換の定義

- **要求電文のデータ変換をする場合**

  サービスアダプタ定義画面で標準電文にサービスリクエスタから入力される電文を設定し，データ変換定義画面でデータ変換を定義します。

  要求電文については，DB アダプタに指定した電文フォーマット（XML フォーマット定義ファイル）の DBA_IN_DATA と対応付けます。

- **応答電文のデータ変換をする場合**

  サービス部品から返される電文に標準電文を設定し，データ変換定義画面でデータ変換を定義します。

  応答電文については，DB アダプタに指定した電文フォーマット（XML フォーマット定義ファイル）の DBA_OUT_DATA と対応付けます。

データ変換の詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7. データ変換の定義」を参照してください。

(6) EAR ファイルの入れ替え

DB アダプタの追加時に EAR ファイルを設定します。DB アダプタを追加したあとに EAR ファイルを修正する必要がある場合，DB アダプタのサービスアダプタ定義画面で，EAR ファイルを入れ替えます。

DB アダプタのサービスアダプタ定義画面で，EAR ファイルを入れ替える手順を次に示します。

1. **ツリービューのサービス定義一覧で，EAR ファイルを入れ替える DB アダプタをダブルクリックします。**

   サービスアダプタ定義画面が表示されます。

2. **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。

3. 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。

4. 入れ替える EAR ファイルを指定します。

(7)　サービスリクエスタの作成

DB アダプタに要求を出すサービスリクエスタは，要求電文を受け付ける標準受付の種類によって作成方法が異なります。それぞれの標準受付の種類ごとの作成方法を参照して作成してください。参照先を次の表に示します。

表 3-12　DB アダプタに要求を出すサービスリクエスタの作成方法

標準受付の種類		参照先
同期受付	Web サービス（SOAP1.1）	マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「9.2　標準の同期受付（Web サービス）に要求を出すサービスリクエスタ（SOAP 通信基盤）」
	Web サービス（SOAP1.2）	マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「9.3　標準の同期受付（Web サービス）に要求を出すサービスリクエスタ（JAX-WS エンジン）」
	SessionBean	マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「9.4　標準の同期受付（SessionBean）に要求を出すサービスリクエスタ」
非同期	MDB（WS-R）	マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「9.5　標準の非同期受付（MDB（WS-R））に要求を出すサービスリクエスタ」
	MDB（DB キュー）	マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「9.6　標準の非同期受付（MDB（DB キュー））に要求を出すサービスリクエスタ」

以降の項では，DB アダプタ電文の形式について説明します。

DB アダプタ用のサービスリクエスタの電文の形式が，DB アダプタ電文の形式と異なる場合，データ変換が必要です。データ変換の詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.　データ変換の定義」を参照してください。

(a)　要求電文の形式

要求電文は，XML 文書形式で作成します。

要素，属性およびテキストデータは，W3C の XML の仕様に従って指定します。XML 文書で使用できない「<」，「>」，「&」などの文字を使用する場合，エスケープするか，CDATA セクションを利用する必要があります。

要求電文の形式は，単一の SQL 文の実行を要求する場合と，複数の SQL 文の実行を要求する場合とで異なります。

要求電文の形式を次に示します。なお，斜体の文字は可変値を表します。

単一の SQL 文の実行を要求する場合

ルート要素（DBadapter 要素）の下位要素として *SQL 識別子*要素を記述します。記述できる *SQL 識別子*要素の数は，1 つだけです。

```
<DBadapter>
  <SQL識別子 out_maxOccurs="検索結果出力最大件数">
    <DBA_IN_DATA>
      <引数要素 nulldata="空要素の扱い">引数要素のデータ</引数要素>
        :
    </DBA_IN_DATA>
  </SQL識別子>
</DBadapter>
```

複数の SQL 文の実行を要求する場合

ルート要素（DBadapter 要素）の下位要素として DBA_MULTI_SQL 要素を記述します。複数の DBA_MULTI_SQL 要素を記述できます。記述できる数は，SQL オペレーション定義ファイルの MAX_SQL_NO 要素に設定した値以下です。

```
<DBadapter>
  <DBA_MULTI_SQL>
    <SQL識別子 out_maxOccurs="検索結果出力最大件数">
      <DBA_IN_DATA>
        <引数要素 nulldata="空要素の扱い">引数要素のデータ</引数要素>
          :
      </DBA_IN_DATA>
    </SQL識別子>
  </DBA_MULTI_SQL>
    :
</DBadapter>
```

(b)　要求電文の要素と属性

要求電文の要素と属性について次の表に示します。なお，要素および属性の指定は，SQL オペレーション定義ファイルの設定値に依存します。SQL オペレーション定義ファイルについては「(1)　SQL オペレーション定義ファイルの作成」を参照してください。

表 3-13　要求電文の要素と属性

要素名・属性名	種別	内容と設定値	設定値の型	省略可否
DBadapter	要素	ルート要素です。次の下位要素があります。 単一 SQL 文の実行要求の場合 　下位要素は *SQL 識別子*要素です。 複数 SQL 文の実行要求の場合 　下位要素は DBA_MULTI_SQL 要素です。	－	×

要素名・属性名	種別	内容と設定値	設定値の型	省略可否
DBA_MULTI_SQL	要素	サービスリクエスタからの要求が，複数の SQL 文を実行する要求であることを示す要素です。*SQL 識別子*以降の下位要素で入力情報を設定します。 SQL オペレーション定義ファイルの MAX_SQL_NO 要素に設定した値以内で複数記述できます。 注 単一の SQL 文の実行を要求する場合，この要素は記述できません。	−	○
SQL 識別子	要素	*SQL 識別子*には SQL オペレーション定義ファイルで指定した SQL 識別子の名称を指定します。 DBA_IN_DATA 以降の下位要素で，入力情報を設定します。 属性に out_maxOccurs があります。	−	○
out_maxOccurs	属性	検索結果出力最大件数を指定します。0 以上の値を指定します。 省略した場合 SQL オペレーション定義ファイルで指定した out_maxOccurs の値が設定されます。 「0」を設定した場合 最大件数が 2,147,483,647 件に設定されます。	xsd:int	○
DBA_IN_DATA	要素	*引数要素*以降の下位要素で，SQL オペレーション定義ファイルで設定した引数要素のデータを設定します。	−	×
引数要素	要素	*引数要素*には SQL オペレーション定義ファイルで指定した引数要素の名称を指定します。 この要素には，SQL オペレーション定義ファイルで指定した引数要素のデータが格納されます。 属性に nulldata があります。 注 SQL オペレーション定義ファイルの設定で，該当する SQL 識別子に引数要素がない場合，この要素は設定できません。	xsd:string または data_type の設定値※	×
nulldata	属性	SQL オペレーション定義ファイルで指定した dba_inf（引数の種別）が「data」の場合に，空要素のデータを「null」として扱うかどうかを指定します（dba_inf が「data」以外の場合は，指定できません）。 「null」として扱う場合 「"Y"」を指定します。 「null」として扱わない場合 「"N"」を指定します。 なお，省略した場合，この属性には「"N"」が設定されます。	xsd:string {Y | N}	○

（凡例）

－：設定値はありません。

○：省略できます。

×：省略できません。

注※

格納する引数要素の型は，SQL オペレーション定義ファイルで，該当する引数要素に指定した dba_inf（引数の種別）によって異なります。dba_inf が table，column または preset の場合，xsd:string 型になります。dba_inf が data の場合，data_type（データ型）の設定値に従ってください。data_type の設定値と引数要素の型の対応については，「表 3-14」および「表 3-15」を参照してください。

また，dba_inf が data の場合，引数要素に格納するデータをシングルクォーテーション（'）で囲む必要はありません。

表 3-14　data_type の設定値と引数要素の型の対応（HiRDB の場合）

data_type の設定値	引数要素の型
INTEGER	xsd:int※
SMALLINT	xsd:short※
DECIMAL	xsd:string
FLOAT	xsd:double※
REAL	xsd:float※
CHAR	xsd:string
VARCHAR	xsd:string
DATE	xsd:date※
TIME	xsd:string
TIMESTAMP	xsd:string
LONGVARBINARY	xsd:hexBinary
	xsd:base64Binary

注※

記述した型に加え，空要素の設定も許可されます。

表 3-15　data_type の設定値と引数要素の型の対応（Oracle の場合）

data_type の設定値	引数要素の型
NUMERIC	xsd:string
CHAR	xsd:string
VARCHAR	xsd:string
DATE	xsd:date※
TIME	xsd:string
TIMESTAMP	xsd:string

data_type の設定値	引数要素の型
LONGVARCHAR	xsd:string
CLOB	xsd:string
VARBINARY	xsd:hexBinary
	xsd:base64Binary
LONGVARBINARY	xsd:hexBinary
	xsd:base64Binary
BLOB	xsd:hexBinary
	xsd:base64Binary

注※

記述した型に加え，空要素の設定も許可されます。

表 3-16　data_type の設定値と引数要素の型の対応（MySQL の場合）

data_type の設定値	引数要素の型
TINYINT	xsd:short
INTEGER	xsd:int
BIGINT	xsd:long
REAL	xsd:float
DOUBLE	xsd:double
DECIMAL	xsd:string
BIT	xsd:boolean
CHAR	xsd:string
VARCHAR	xsd:string
LONGVARCHAR	xsd:string
VARBINARY	xsd:hexBinary
	xsd:base64Binary
LONGVARBINARY	xsd:hexBinary
	xsd:base64Binary
DATE	xsd:date
TIMESTAMP	xsd:string
TIME	xsd:string

表 3-17　data_type の設定値と引数要素の型の対応（PostgreSQL の場合）

data_type の設定値	引数要素の型
SMALLINT	xsd:short
INTEGER	xsd:int
BIGINT	xsd:long
REAL	xsd:float
DOUBLE	xsd:double
NUMERIC	xsd:string
BIT	xsd:boolean
BINARY	xsd:hexBinary
CHAR	xsd:string
VARCHAR	xsd:string
DATE	xsd:date
TIME	xsd:string
TIMESTAMP	xsd:string

なお，DB アダプタに送信する要求電文で設定する引数要素の値は，次の表のとおりに扱います。

表 3-18　DB アダプタに送信する要求電文での引数要素の扱い

要求電文の引数要素の値	SQL オペレーション定義ファイルの設定値		要求電文の nulldata の属性値	引数要素の扱い
	dba_inf の属性値	data_type の属性値		
空要素	data	－	"Y"または"△Y△"※	null
			"Y"または"△Y△"以外（省略時も含む）	空文字列（""）
	data 以外	－	－	空文字列（""）
空要素以外	data	CHAR，VARCHAR，LONGVARCHAR，CLOB	－	値の前後の空白文字は削除しません。<*引数要素*>△ABC△</*引数要素*>は，空白文字を削除しないで"△ABC△"として扱います。
		CHAR，VARCHAR，LONGVARCHAR，CLOB 以外	－	値の前後の空白文字を削除します。<*引数要素*>△123△</*引数要素*>は，空白文字を削除して"123"として扱います。
	data 以外	－	－	値の前後の空白文字は削除しません。<*引数要素*>△ABC△</*引数要素*>は，空白文字を削除しないで"△ABC△"として扱います。

（凡例）

－：判定の材料ではありません。

△：1 つ以上の空白文字（半角空白文字，改行文字，タブ文字）です。

注※

nulldata の属性値の前後の空白文字は無視します。

(c) 応答電文の形式

応答電文は，XML 文書形式で出力されます。

なお，検索結果（列名など）に XML 文書で使用できない「<」や「&」などの文字が含まれていた場合，「<」や「&」などにエスケープして応答電文に設定されます。そのため，データベースに格納されているデータをエスケープする必要はありません。

応答電文の形式は，単一の SQL 文の実行要求に対する応答の場合と，複数の SQL 文の実行要求に対する応答の場合とで異なります。

応答電文の形式を次に示します。なお，斜体の文字は可変値を表します。

単一の SQL 文の実行要求に対する応答の場合

ルート要素（DBadapter 要素）の下位要素として *SQL 識別子* 要素が出力されます。

```
<DBadapter>
  <SQL識別子 out_maxOccurs="検索結果出力最大件数">
    <DBA_IN_DATA>
      <引数要素 nulldata="空要素の扱い">引数要素のデータ</引数要素>
        :
    </DBA_IN_DATA>
    <DBA_OUT_DATA>
      <DBA_ResultSetNo>SQL リターン値</DBA_ResultSetNo>
      <DBA_ResultSetXmlNo>検索結果の件数</DBA_ResultSetXmlNo>
      <DBA_ResultSetName>
        <DBA_ResultColumnName cid="列番号">列の名称</DBA_ResultColumnName>
          :
      </DBA_ResultSetName>
      <DBA_ResultSet lid="行番号">
        <DBA_ResultColumn cid="列番号">該当行n列目の検索結果</DBA_ResultColumn>
          :
      </DBA_ResultSet>
        :
    </DBA_OUT_DATA>
  </SQL識別子>
</DBadapter>
```

複数の SQL 文の実行要求に対する応答の場合

ルート要素（DBadapter 要素）の下位要素として DBA_MULTI_SQL 要素が出力されます。要求電文で記述した DBA_MULTI_SQL 要素と同数の DBA_MULTI_SQL 要素が出力されます。

```
<DBadapter>
  <DBA_MULTI_SQL>
    <SQL識別子 out_maxOccurs="検索結果出力最大件数">
      <DBA_IN_DATA>
```

```
        <引数要素 nulldata="空要素の扱い">引数要素のデータ</引数要素>
          :
      </DBA_IN_DATA>
      <DBA_OUT_DATA>
        <DBA_ResultSetNo>SQLリターン値</DBA_ResultSetNo>
        <DBA_ResultSetXmlNo>検索結果の件数</DBA_ResultSetXmlNo>
        <DBA_ResultSetName>
          <DBA_ResultColumnName cid="列番号">列の名称</DBA_ResultColumnName>
            :
        </DBA_ResultSetName>
        <DBA_ResultSet lid="行番号">
          <DBA_ResultColumn cid="列番号">該当行n列目の検索結果</DBA_ResultColumn>
            :
        </DBA_ResultSet>
          :
      </DBA_OUT_DATA>
    </SQL識別子>
  </DBA_MULTI_SQL>
    :
</DBadapter>
```

なお，要求電文で入力情報が０（ゼロ）の場合，応答電文は次の形式で出力されます。

入力情報が0の場合の要求電文

```
<DBadapter>
  <DBA_MULTI_SQL>
    <SQL識別子 out_maxOccurs="検索結果出力最大件数">
      <DBA_IN_DATA>
        <引数要素 nulldata="空要素の扱い">引数要素のデータ</引数要素>
          :
      </DBA_IN_DATA>
      <DBA_OUT_DATA>
        <DBA_ResultSetNo>SQLリターン値</DBA_ResultSetNo>
        <DBA_ResultSetXmlNo>検索結果の件数</DBA_ResultSetXmlNo>
        <DBA_ResultSetName>
          <DBA_ResultColumnName cid="列番号">列の名称</DBA_ResultColumnName>
            :
        </DBA_ResultSetName>
        <DBA_ResultSet lid="行番号">
          <DBA_ResultColumn cid="列番号">該当行n列目の検索結果</DBA_ResultColumn>
            :
        </DBA_ResultSet>
          :
      </DBA_OUT_DATA>
    </SQL識別子>
  </DBA_MULTI_SQL>
    :
</DBadapter>
```

入力情報が0の場合とは，この部分のデータをすべて省略した要求電文のことです。

入力情報が0の場合の応答電文

```
<DBadapter>
</DBadapter>
```

(d)　応答電文の要素と属性

応答電文の要素と属性について次の表に示します。

表 3-19　応答電文の要素と属性（DB アダプタ用のサービスリクエスタ）

要素名・属性名	種別	内容と設定値	格納される値の型	省略
DBadapter	要素	ルート要素です。次の下位要素があります。 単一 SQL 文の実行要求に対する応答の場合 　下位要素として *SQL 識別子* 要素が出力されます。 複数 SQL 文の実行要求に対する応答の場合 　下位要素として DBA_MULTI_SQL 要素が出力されます。	－	×
DBA_MULTI_SQL	要素	サービスリクエスタからの複数の SQL 文の実行要求に対する応答であることを示す要素です。*SQL 識別子* 以降の下位要素に入力情報（要求電文での設定情報）	－	×※1

要素名・属性名	種別	内容と設定値	格納される値の型	省略
DBA_MULTI_SQL	要素	および出力情報（SQL の実行結果情報）が出力されます。 要求電文で記述した DBA_MULTI_SQL 要素と同数の DBA_MULTI_SQL 要素が出力されます。	−	×※1
SQL 識別子	要素	下位の要素「DBA_IN_DATA」および「DBA_OUT_DATA」に，それぞれ入力情報（要求電文での設定情報）および出力情報（SQL の実行結果情報）が格納されます。 *SQL 識別子*は，要求電文に設定した SQL 識別子です。 なお，属性 out_maxOccurs（検索結果出力最大件数）は，要求電文で設定した場合にだけ格納されます。	−	×
DBA_IN_DATA	要素	要求電文で設定した入力情報がこの要素の下位要素に格納されます。	−	×
DBA_OUT_DATA	要素	下位の要素に SQL 実行時の出力情報が格納されます。実行した SQL によって，格納される出力情報が異なります。 INSERT，UPDATE，DELETE の場合 DBA_ResultSetNo が下位要素として格納されます。 SELECT の場合 次の要素が下位要素として格納されます。 • DBA_ResultSetNo • DBA_ResultSetXmlNo • DBA_ResultSetName • DBA_ResultSet	−	×
DBA_ResultSetNo	要素	実行した SQL によって，格納される出力情報が異なります。 INSERT，UPDATE，DELETE の場合 更新した行数（PreparedStatement クラスの executeUpdate で取得した値）が格納されます。格納される要素の値が 2,147,483,647 を超えた場合に格納される値は，DB Connector の使用する JDBC ドライバの PreparedStatement クラスの executeUpdate の仕様に従います。 SELECT の場合 検索結果の件数が格納されます。 格納される要素の値が 2,147,483,647 を超えた場合，2,147,483,647 が格納されます。	xsd:int	×
DBA_ResultSetXmlNo	要素	実行した SQL が SELECT の場合，実際の検索結果の件数（DBA_ResultSetNo の値）のうち，この応答電文に格納されている検索結果の件数が格納されます。	xsd:int	○

要素名・属性名	種別	内容と設定値	格納される値の型	省略
DBA_ResultSetXmlNo	要素	DBA_ResultSetNo の値が，要求電文または SQL オペレーション定義ファイルの out_maxOccurs に指定した値よりも大きい場合，out_maxOccurs に指定した値が格納されます。 また，格納される要素の値が 2,147,483,647 を超えた場合，2,147,483,647 が格納されます。	xsd:int	○
DBA_ResultSetName	要素	実行した SQL が SELECT の場合，検索結果の列の名称が，下位要素 DBA_ResultColumnName に格納されます。 なお，検索結果が 0 件の場合，この要素は格納されません。	−	△
DBA_ResultColumnName	要素	実行した SQL が SELECT の場合，該当する列の名称が格納されます。 なお，結果の列がデータベースの列の場合，「列名」の形式で設定されます。また，結果の列名が取得されない場合，SQL を発行した先のデータベースの仕様に従った名称が返されます。 属性に cid があります。	xsd:string	△
cid	属性	DBA_ResultColumnName に格納される値の列番号を示します。	xsd:int	
DBA_ResultSet	要素	実行した SQL が SELECT の場合，一行ごとの結果情報が，下位要素の DBA_ResultColumn に格納されます。なお，検索結果が 0 件の場合，この要素は格納されません。 属性に lid があります。	−	△
lid	属性	DBA_ResultSet に格納される値の行番号を示します。 また，格納される要素の値が 2,147,483,647 を超えた場合，2,147,483,647 が格納されます。	xsd:int	
DBA_ResultColumn	要素	該当する行（DBA_ResultSet で示される行）の *n* 列目の検索結果が格納されます。※2 属性に cid および nulldata があります。	xsd:string	△
cid	属性	DBA_ResultColumn に格納される値の列番号を示します。	xsd:int	
nulldata	属性	DBA_ResultColumn に格納されたデータが null であるかどうかを示します。 null の場合「Y」が設定されます。null でない場合，この属性は設定されません。	xsd:string	

（凡例）

−：値は格納されません。

○：SQL が SELECT の場合にだけ出力されます。

△：SQL が SELECT の場合で，実行結果が 1 件以上のときに出力されます。

×：必ず出力されます。省略されません。

注※1

複数の SQL 文の実行要求に対する応答の場合には必ず出力されます。単一の SQL 文の実行要求に対する応答の場合には出力されません。

注※2

ResultSet クラスの getString で取得した値が格納されます。そのため，検索結果の書式は DB Connector の使用する JDBC ドライバの ResultSet クラスの getString の仕様に従います。検索結果の書式を変更する場合，データベースが提供する関数（ファンクション）を使用して検索してください。

HiRDB の TIMESTAMP 型（列名：c_ts）の書式を VARCHAR_FORMAT 関数で変更する場合の例を次に示します。

```
SELECT VARCHAR_FORMAT(c_ts, 'YYYY-MONTH-DD HH:MI') FROM 表名
```

3.3.6　TP1 アダプタを定義する

TP1 アダプタの定義方法について説明します。

(1)　電文フォーマットの作成

サービスリクエスタ，TP1 アダプタ，およびサービスの間でやり取りされる電文フォーマットを作成します。電文フォーマットの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.　電文フォーマットの作成」を参照してください。

(2)　データ変換の定義

データ変換定義画面で，変換元の電文フォーマット定義ファイルおよび変換先の電文フォーマット定義ファイルを設定して，データ変換を定義します。

データ変換の定義方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.　データ変換の定義」を参照してください。

(3)　サービスアダプタ定義画面での操作

TP1 アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。

   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービス部品制御情報の［サービス名］および［サービス ID］を必要に応じて編集します。

3. サービス部品制御情報の［追加］ボタンをクリックして，オペレーションを追加します。

4. オペレーション情報の［通信モデル］のドロップダウンリストから，「同期」または「非同期」を選択します。

5. 要求電文について，手順 6.から手順 15.までを実施します。

6. 次に示す操作をします。

   標準電文のフォーマットを指定する場合

   手順 7.～9.を実施して，手順 10.に進みます。

   標準電文のフォーマットを指定しない場合

   手順 10.に進みます。

7. 標準電文の［使う］チェックボックスをチェックします。

8. 標準電文の［参照］ボタンをクリックして，［電文フォーマット］に標準電文のフォーマットを指定します。

   指定できる電文フォーマットの形式については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.5　XML スキーマの適用範囲」を参照してください。

9. 標準電文の［表示］ボタンをクリックします。

   電文フォーマットが XML 形式の場合，標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。

10. サービス部品電文の［参照］ボタンをクリックして，［電文フォーマット］にサービス部品電文のフォーマットを指定します。

11. サービス部品電文の［表示］ボタンをクリックします。

    電文フォーマットが XML 形式の場合，サービス部品電文のフォーマットが表示されます。必要に応じて指定したサービス部品電文のフォーマットを確認してください。

12. 次に示す操作をします。

    標準電文の［使う］チェックボックスをチェックした場合

    手順 13.～15.を実施して，手順 16.に進みます。

    標準電文の［使う］チェックボックスをチェックしない場合

    手順 16.に進みます。

13. データ変換定義のファイル名を入力します。

14. ［編集］ボタンをクリックします。

    データ変換定義画面が表示されます。初めて定義する場合は，［ルート要素選択］ダイアログが表示されます。

15. 標準電文とサービス電文の内容をマッピングします。

16. 応答電文についても，手順 6.から手順 15.までを実施します。

17. 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。

18. サービスアダプタ（EJB-JAR ファイル）の名称を確認します。

19. ユーティリティクラス（JAR ファイル）に次のユーティリティクラスが追加されていることを確認します。
    - adptp1pc.jar（プロトコルコンバータアーカイブファイル）

20. 独自定義ファイルに次のファイルが追加されていることを確認します。
    - adptp1_config.xml（TP1 アダプタ通信構成定義ファイル）
    - customadapter_properties.xml（TP1 アダプタ環境定義ファイル）

(4) TP1 アダプタ環境定義ファイルの編集

TP1 アダプタの環境定義に必要な情報は，XML 形式の TP1 アダプタ環境定義ファイルに定義します。

TP1 アダプタ環境定義ファイルの編集手順を示します。

1. サービスアダプタ定義画面（詳細）の［独自定義ファイル］で「customadapter_properties.xml」を選択し，［編集］ボタンをクリックします。
   TP1 アダプタ環境定義ファイルを編集するためのエディタが起動します。

2. エディタ上で TP1 アダプタ環境定義ファイルを編集します。
   TP1 アダプタ環境定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.4.2　TP1 アダプタ環境定義ファイル」を参照してください。

3. Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(5) TP1 アダプタ通信構成定義ファイルの編集

TP1 アダプタの通信構成定義に必要な情報は，XML 形式の TP1 アダプタ通信構成定義ファイルに定義します。

TP1 アダプタ通信構成定義ファイルの編集手順を示します。

1. サービスアダプタ定義画面（詳細）の［独自定義ファイル］で「adptp1_config.xml」を選択し，［編集］ボタンをクリックします。
   TP1 アダプタ通信構成定義ファイルを編集するためのエディタが起動します。

2. エディタ上で TP1 アダプタ通信構成定義ファイルを編集します。
   TP1 アダプタ通信構成定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.4.3　TP1 アダプタ通信構成定義ファイル」を参照してください。

3. Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(6) TP1 アダプタの設定項目の反映

TP1 アダプタを選択した状態で，「保存」メニューを選択すると設定項目を保存できます。このときに保存されるファイルが EAR ファイルです。EAR ファイルは運用環境から利用できます。

(7) TP1 アダプタ実行環境プロパティファイルの作成

TP1 アダプタ実行環境プロパティファイルの作成方法について説明します。

TP1 アダプタ実行環境プロパティファイルは，サンプルファイルを利用して作成します。サンプルファイルの格納先を次に示します。

```
<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>¥CSC¥custom-adapter¥TP1¥config¥templates¥adptp1.properties
```

TP1 アダプタ実行環境プロパティファイルの作成手順を次に示します。

1. サンプルファイルをコピーして，次のディレクトリに格納します。
   - 全 HCSC サーバ共通の定義の場合

     ```
     <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>¥CSC¥custom-adapter¥TP1¥config
     ```

   - 特定の HCSC サーバ用の定義の場合

     ```
     <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>¥CSC¥custom-adapter¥TP1¥config¥<HCSCサーバ名>※
     ```

     注※　サブディレクトリ<HCSC サーバ名>は，ユーザが作成してください。Windows の場合，ディレクトリ名は大文字と小文字は区別しません。また，HCSC サーバ用のディレクトリには，HCSC サーバの管理者のアクセス権限を設定する必要があります。
2. サンプルファイルの名称を<サービス ID>.properties に変更します。
3. 定義内容を編集して保存します。

TP1 アダプタ実行環境プロパティファイルは，TP1 アダプタを開始する際に実行環境に反映されます。TP1 アダプタ実行環境プロパティファイルの内容を変更する場合は，TP1 アダプタを停止してから作業を行います。TP1 アダプタを再開始すると，変更内容が実行環境に反映されます。

TP1 アダプタ実行環境プロパティファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.7.1　TP1 アダプタ実行環境プロパティファイル」を参照してください。

(8) TP1/Client/J 環境定義の設定

TP1 アダプタの RPC 通信および MHP を接続先とした TCP/IP 通信で使用するサービス要求方式によって，Service Architect で指定する環境定義の値が異なります。必ず各サービス要求方式に対応する TP1/Client/J 環境定義を指定してください。

ここで説明する以外の TP1/Client/J 環境定義については，マニュアル「サービスプラットフォーム 解説」の「付録 A.1　TP1/Client/J の機能」を参照してください。

TP1/Client/J 環境定義は，TP1 アダプタを開始する際に実行環境に反映されます。TP1/Client/J 環境定義の内容を変更する場合は，TP1 アダプタを停止してから作業を行います。TP1 アダプタを再開始すると，変更内容が実行環境に反映されます。

(a)　リモート API 機能を使用する場合

リモート API 機能を使用する場合，RPC 接続の確立方式および常設コネクションの管理方法によって，TP1/Client/J で指定する環境定義の値が異なります。TP1 アダプタの通信構成定義および TP1/Client/J 環境定義の対応を次の表に示します。

表 3-20　TP1 アダプタの通信構成定義および TP1/Client/J 環境定義の対応

TP1/Client/J 環境定義	TP1 アダプタの通信構成定義		
	connect_mode：RAP		
	active_mode：STAND		active_mode：EACH
	standcon_mode：MANUAL	standcon_mode：AUTO	standcon_mode 要素は指定不要
dcnamuse	N（使用しない）	N（使用しない）	N（使用しない）
dcrapdirect	Y（使用する）	Y（使用する）	Y（使用する）
dcscddirect	N（使用しない）	N（使用しない）	N（使用しない）
dcrapautoconnect	N（使用しない）	Y（使用する）	N（使用しない）

（凡例）

：（コロン）は，要素および要素の指定値の対応を示します。例えば，「connect_mode：RAP」は，connect_mode 要素に RAP を指定することを示します。

リモート API 機能を使用する場合の TP1/Client/J 環境定義を次の表に示します。

表 3-21　リモート API 機能使用時の TP1/Client/J 環境定義

TP1/Client/J 環境定義	指定値	説明
dcnamuse	N（使用しない）	ネームサービス機能の使用の有無を指定します。
dcrapdirect	Y（使用する）	リモート API 機能の使用の有無を指定します。
dcscddirect	N（使用しない）	スケジューラダイレクト機能の使用の有無を指定します。
dchost	ホスト名，rap リスナーのポート番号	rap リスナーが存在する接続先システムのホスト名および rap リスナーのポート番号を指定します。dcrapport オペランドで rap リスナーのポート番号を指定している場合，ポート番号の指定は省略できます。
dcrapport	rap リスナーのポート番号	rap リスナーのポート番号を指定します。dchost オペランドでポート番号を指定する場合，このオペランドの指定は省略できます。

TP1/Client/J 環境定義	指定値	説明
dcrapautoconnect※	Y ｜ N	rap サーバとの間の常設コネクションを自動的に確立させるかどうかを指定します。TP1 アダプタの通信構成定義の standcon_mode 要素と指定値を対応させてください。

注※

dcrapautoconnect オペランドの指定による性能についてのメリットおよびデメリットを次の表に示します。

表 3-22　dcrapautoconnect オペランドの指定によるメリットおよびデメリット

dcrapautoconnect オペランドの指定値	メリット	デメリット
Y（オートコネクト）	接続先システムからコネクションを解放されても，サービス要求実行時に再確立を行います。	初回のサービス要求実行時にコネクションの確立を行うため性能劣化のおそれがあります。
N（非オートコネクト）	TP1 アダプタ開始時にすべてのコネクションの確立を行います。初回のサービス要求実行時に，コネクションの確立をする必要なく通信できます。	確立するコネクションの数が多いと TP1 アダプタの開始処理に時間が掛かります。

(b)　ネームサービス機能を使用する場合

ネームサービス機能を使用する場合，TP1 アダプタの通信構成定義の connect_mode 要素に STAND を指定します。ネームサービス機能を使用する場合の TP1/Client/J 環境定義を次の表に示します。

表 3-23　ネームサービス機能使用時の TP1/Client/J 環境定義

TP1/Client/J 環境定義	指定値	説明
dcnamuse	Y（使用する）	ネームサービス機能の使用の有無を指定します。
dcrapdirect	N（使用しない）	リモート API 機能の使用の有無を指定します。
dcscddirect	N（使用しない）	スケジューラダイレクト機能の使用の有無を指定します。
dchost	ホスト名，ネームサービスのポート番号	ネームサービスが存在する接続先システムのホスト名およびネームサービスのポート番号を指定します。dcnamport オペランドでネームサービスのポート番号を指定している場合，ポート番号の指定は省略できます。
dcnamport	ネームサービスのポート番号	ネームサービスのポート番号を指定します。dchost オペランドでポート番号を指定する場合，このオペランドの指定は省略できます。

(c)　スケジューラダイレクト機能を使用する場合

スケジューラダイレクト機能を使用する場合，TP1 アダプタの通信構成定義の connect_mode 要素に STAND を指定します。スケジューラダイレクト機能を使用する場合の TP1/Client/J 環境定義を次の表に示します。

表 3-24　スケジューラダイレクト機能使用時の TP1/Client/J 環境定義

TP1/Client/J 環境定義	指定値	説明
dcnamuse	N（使用しない）	ネームサービス機能の使用の有無を指定します。
dcrapdirect	N（使用しない）	リモート API 機能の使用の有無を指定します。
dcscddirect	Y（使用する）	スケジューラダイレクト機能の使用の有無を指定します。
dchost	ホスト名，スケジューラダイレクトサービスのポート番号	スケジューラダイレクトサービスが存在する接続先システムのホスト名およびスケジューラダイレクトサービスのポート番号を指定します。dcscdport オペランドでスケジューラダイレクトサービスのポート番号を指定している場合，ポート番号の指定は省略できます。
dcscdport	スケジューラダイレクトサービスのポート番号	スケジューラダイレクトサービスのポート番号を指定します。dchost オペランドでポート番号を指定する場合，このオペランドの指定は省略できます。

(d)　通信先を指定した接続をする場合

TP1 アダプタで通信相手と通信先を指定した接続をする場合，TP1 アダプタの通信構成定義の connect_mode 要素に STATIC を指定します。通信先を指定した接続をする場合の TP1/Client/J 環境定義を次の表に示します。

表 3-25　通信先指定による接続時の TP1/Client/J 環境定義

TP1/Client/J 環境定義	指定値	説明
dcnamuse	N（使用しない）	ネームサービス機能の使用の有無を指定します。
dcrapdirect	N（使用しない）	リモート API 機能の使用の有無を指定します。
dcscddirect	Y（使用する）	スケジューラダイレクト機能の使用の有無を指定します。

注意事項

次の表に示すオペランドの定義項目は使用されません。接続先のホスト名，ポート番号の情報は，TP1 アダプタの通信構成定義の static_host および static_port 要素を使用します。また，dcrapautoconnect オペランドの定義は無効となります。

表 3-26　通信先指定による接続時に使用されない TP1/Client/J 環境定義

TP1/Client/J 環境定義	指定値	説明
dchost	ホスト名，スケジューラダイレクトサービスのポート番号	スケジューラダイレクトサービスが存在する接続先システムのホスト名，およびスケジューラダイレクトサービスのポート番号
dcscdport	スケジューラダイレクトサービスのポート番号	スケジューラダイレクトサービスのポート番号

(e) MHP を接続先とした TCP/IP 接続をする場合

TP1 アダプタで MHP を接続先とした TCP/IP 接続をする場合，TP1 アダプタの通信構成定義の connect_mode 要素に TCPIP を指定します。MHP を接続先とした TCP/IP 接続をする場合の TP1/Client/J 環境定義を次の表に示します。

表 3-27　MHP を接続先とした TCP/IP 接続の TP1/Client/J 環境定義

TP1/Client/J 環境定義	指定値	説明
dcsndhost	接続する MHP が存在するノードのホスト名	コネクションを確立して接続する MHP が存在するノードのホスト名を指定します。
dcsndport	接続する MHP のポート番号	コネクションを確立して接続する MHP のポート番号を指定します。
dcsndrcvtype	同期する場合（メッセージを送受信するための環境） DCCLT_SNDRCV 同期しない場合（メッセージを一方送信するための環境） DCCLT_ONEWAY_SND	TCP/IP 通信機能の使用時に初期化する環境を指定します。
dcsockopenatrcv	Y（cltAssemReceive メソッド実行時）	TCP/IP 通信機能の使用時に，1 コネクションで送受信する場合の，受信用ソケットを開設する契機を指定します。

必須の定義項目は，TP1 アダプタの起動時に指定値が適切かどうかをチェックします。次の項目がチェックされます。

- dcsndhost および dcsndport の指定が省略されていないか。
- dcsndrcvtype および dcsockopenatrcv に指定値以外の値が指定されていないか。

適切でない場合，エラーが出力され，アダプタの起動に失敗します。

3.3.7　ファイルアダプタを定義する

ファイルアダプタの定義方法について説明します。

(1)　電文フォーマットの作成

ファイルアダプタで作成する電文フォーマットの内容について説明します。

(a)　ファイルアダプタの要求電文フォーマット

ファイル操作の実行を要求するための要求電文では，次の情報を送信します。

- ファイル操作情報

- ファイル書き込みデータ

●ファイル操作情報

ファイル操作情報は，ファイルにアクセスするために必要な情報です。ファイル操作情報の内容を次の表に示します。

表 3-28　ファイル操作情報

情報	説明
ファイル名	読み込みまたは書き込みが行われるファイル名です。絶対パスで指定します。ファイル名の長さは，UTF-8 へ変換した際に 1～200 バイトの範囲で指定できます。先頭と最後の空白文字は削除されます。Windows の場合，パスの大文字と小文字は区別されません。 次のようなファイル名は指定できません。 • 相対パス，親ディレクトリ表現（「..¥」「../」），およびカレントディレクトリ表現（「.¥」「./」）を含むファイル名。 • シンボリックリンクを含むファイル名。 • Windows の場合，UNC 形式，NTFS のストリーム名，または予約デバイス名を含むファイル名。
読み込みモード	ファイル読み込み時のモードです。次のモードがあります。 • 一括 指定した値の前後の空白文字は削除されます。
書き込みモード	ファイル書き込み時のモードです。次のモードがあります。 • 新規 • 追加 指定した値の前後の空白文字は削除されます。

●ファイル書き込みデータ

XML ファイルまたはバイナリファイルへ書き込み要求をする場合に，要求電文に指定するユーザデータです。

要求電文の内容は，読み込みまたは書き込みを行うファイルの形式や種別によって異なります。このため，電文作成時にはファイル操作情報およびファイル書き込みデータのうち，必要な情報だけを電文に記述します。

XML ファイルの読み込みを行う場合に，要求電文に必要な情報を次の表に示します。

表 3-29　XML ファイル読み込み時の要求電文情報（XML 電文）

情報	電文内のタグ名	指定値	省略時のデフォルト値	指定
ルート要素	ReadRequest	−	−	○
ファイルヘッダ要素	fileheader	−	−	○
ファイル名要素	filename	ファイル名	−	○
ファイルモード要素	filemode	all	all	×

（凡例）

○：必ず指定します。

×：省略できます。

－：該当しません。

XML ファイルの書き込みを行う場合に，要求電文に必要な情報を次の表に示します。

表 3-30　XML ファイル書き込み時の要求電文情報（XML 電文）

情報	電文内のタグ名	指定値	省略時のデフォルト値	指定
ルート要素	WriteRequest	－	－	○
ファイルヘッダ要素	fileheader	－	－	○
ファイル名要素	filename	ファイル名	－	○
ファイルモード要素	filemode	new	new	×
ファイル書き込みデータ	userdata	ユーザ任意	－	○

（凡例）

○：必ず指定します。

×：省略できます。

－：該当しません。

バイナリファイルの読み込みを行う場合に，要求電文に必要な情報を次の表に示します。

表 3-31　バイナリファイル読み込み時の要求電文情報（XML 電文）

情報	電文内のタグ名	指定値	省略時のデフォルト値	指定
ルート要素	ReadRequest	－	－	○
ファイルヘッダ要素	fileheader	－	－	○
ファイル名要素	filename	ファイル名	－	○
ファイルモード要素	filemode	all	all	×

（凡例）

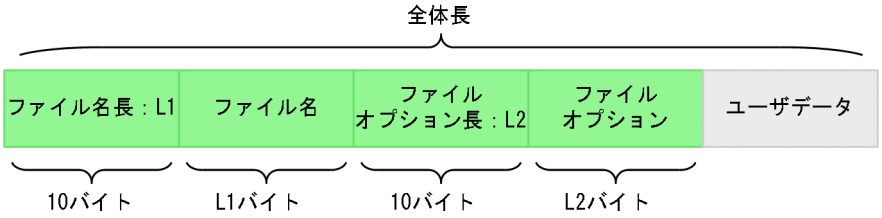
○：必ず指定します。

×：省略できます。

－：該当しません。

バイナリファイルの書き込みを行う場合は，要求電文がバイナリ形式になります。バイナリ形式の要求電文の構造を次の図に示します。

図 3-6　バイナリ形式の要求電文の構造



電文の全体長は次のような関係式が成り立つ必要があります。

```
全体長 ≧ 10 + L1 + 10 + L2
```

バイナリファイルの書き込みを行う場合に，要求電文に必要な情報を次の表に示します。

表 3-32　バイナリファイル書き込み時の要求電文情報（バイナリ電文）

情報	指定値	省略時のデフォルト値	指定
ファイル名長	ファイル名のバイト長を 10 進数値文字列で指定します。	－	○
ファイル名	ファイル名	－	○
ファイルオプション長	ファイルオプションのバイト長を 10 進数値文字列で指定します。	－	○
ファイルオプション	新規書き込み：mode=new 追加書き込み：mode=append	mode=new	×
ユーザデータ	ユーザ任意	－	○※

（凡例）

○：必ず指定します。

×：省略できます。

－：該当しません。

注※

0 バイトの指定ができます。

各要求電文のサンプルは，「(f)　電文例」を参照してください。

(b)　ファイルアダプタの応答電文フォーマット

ファイルアダプタの読み込み情報をファイルアダプタの呼び出し元に返却する**応答電文**では，次の情報を送信します。

- ファイル読み込みデータ
- ファイル書き込み処理結果

●ファイル読み込みデータ

XML ファイルまたはバイナリファイルを読み込む場合に，応答電文に格納され返却されるユーザデータです。

●ファイル書き込み処理結果

XML ファイルまたはバイナリファイルを書き込む場合に，書き込み処理の結果が応答電文へ格納されます。書き込み処理が失敗した場合，ファイルアダプタは HCSC サーバに例外を通知し応答電文は返却しません。応答電文へ書き込み処理の結果が格納されるのは，書き込み処理が成功した場合だけとなります。このため，書き込み時の応答電文の返却値は必ず「OK」になります。

応答電文の内容は，読み込みまたは書き込みを行うファイルの形式や種別によって異なります。ファイルの読み込み，書き込みを行う場合に返信される応答電文の情報を次の表に示します。

表 3-33　応答電文情報

ファイル形式および種別	情報	電文内のタグ名	返却値
XML ファイル読み込み	ファイル読み込みデータ	ユーザ任意	ユーザ任意
XML ファイル書き込み	ファイル書き込み処理の結果	status	OK
バイナリファイル読み込み	ファイル読み込みデータ	－	ユーザ任意
バイナリファイル書き込み	ファイル書き込み処理の結果	status	OK

（凡例）

－：該当しません。

(c)　電文フォーマットの種類

ファイルアダプタ呼び出し元（標準電文）の電文フォーマットがファイルアダプタ（サービス部品電文）の電文フォーマットと異なる場合は，ファイルアダプタ呼び出し元（標準電文）の電文フォーマット定義ファイルおよびファイルアダプタ（サービス部品電文）の電文フォーマット定義ファイルを作成し，データ変換（マッピング）定義画面からそれぞれの電文フォーマットのマッピングを行います。

データ変換を行う場合に作成する電文フォーマット定義ファイルを次の表に示します。

表 3-34　作成する電文フォーマット定義ファイル

ファイル形式および種別	要求電文		応答電文	
	標準電文（呼び出し元）	サービス部品電文（ファイルアダプタ）	標準電文（呼び出し元）	サービス部品電文（ファイルアダプタ）
XML ファイル読み込み	XML※	XML	XML※	XML
XML ファイル書き込み	XML※	XML	XML※	XML
バイナリファイル読み込み	XML※	XML	XML	バイナリ

ファイル形式および種別	要求電文		応答電文	
	標準電文（呼び出し元）	サービス部品電文（ファイルアダプタ）	標準電文（呼び出し元）	サービス部品電文（ファイルアダプタ）
バイナリファイル書き込み	XML	バイナリ	XML※	XML

（凡例）

XML：XML フォーマット定義ファイル

バイナリ：バイナリフォーマット定義ファイル

注※

ファイルアダプタ呼び出し元（標準電文）とファイルアダプタ（サービス部品電文）の電文フォーマットが同じ場合，ファイルアダプタ呼び出し元（標準電文）の電文フォーマット定義ファイルを作成したり，設定したりする必要はありません。

ファイルアダプタで利用する電文で XML 形式のデータを扱うか，バイナリ形式（XML 形式以外）のデータを扱うかによって，作成する電文フォーマット定義ファイルは異なります。

- XML 形式のデータを扱う場合

  XML 形式のデータを扱う場合，XML フォーマット定義ファイルを作成します。XML フォーマット定義ファイルは，XML スキーマファイル（拡張子：.xsd）として作成します。XML フォーマット定義ファイルの作成方法については，「(d)　XML フォーマット定義ファイルの作成」を参照してください。

- バイナリ形式のデータを扱う場合

  バイナリ形式（XML 形式以外）のデータを扱う場合，バイナリフォーマット定義ファイルを作成します。バイナリフォーマット定義ファイルは，バイナリ形式のデータ内の値の格納形式（値の長さや，ほかの値との関係など）を定義し，定義した情報を XML スキーマファイルに付加したファイル（拡張子：.fdx）です。バイナリフォーマット定義ファイルは，XML の仕様に合わせて作成されるため，データ変換を定義するときの変換元および変換先として，XML フォーマット定義ファイルと同様に利用できます。バイナリフォーマット定義ファイルの作成方法については，「(e)　バイナリフォーマット定義ファイルの作成」を参照してください。

(d)　XML フォーマット定義ファイルの作成

標準電文およびファイルアダプタ（サービス部品電文）の作成方法を次に示します。

●標準電文の XML フォーマット定義ファイルの作成

標準電文の XML フォーマット定義ファイルは，Eclipse が提供する WST（Web Standard Tools）などを利用して作成します。

作成する XML フォーマット定義ファイルは，HCSC サーバで使用できる XML スキーマの適用範囲を満たしている必要があります。条件の詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.5　XML スキーマの適用範囲」を参照してください。

ファイルアダプタおよびビジネスプロセスの定義で，利用する電文フォーマットファイルを選択したときにエラー（メッセージ ID が KECX で始まるエラーメッセージ）が発生した場合，XML スキーマファイルにエラーがあります。この場合，マニュアル「アプリケーションサーバ メッセージ(構築／運

用／開発用)」の「11. KECX（XML Processor が出力するメッセージ)」のメッセージを参照して対処してください。

●ファイルアダプタ（サービス部品電文）の XML フォーマット定義ファイルの作成

ファイルアダプタ（サービス部品電文）の XML フォーマット定義ファイルを作成する場合に，ユーザが使用する定義ファイルおよび編集の必要性について次の表に示します。

表 3-35　ユーザが使用する定義ファイルおよび編集の必要性

ファイル操作モード	電文の種類	使用する定義ファイル	ユーザ編集
読み込み	要求電文	adpff_read.xsd	×
	応答電文	ユーザ作成	○
書き込み	要求電文	adpff_header.xsd	×
		adpff_write.xsd	○
	応答電文	adpff_result.xsd	×

（凡例）

○：編集が必要です。

×：編集は不要です。

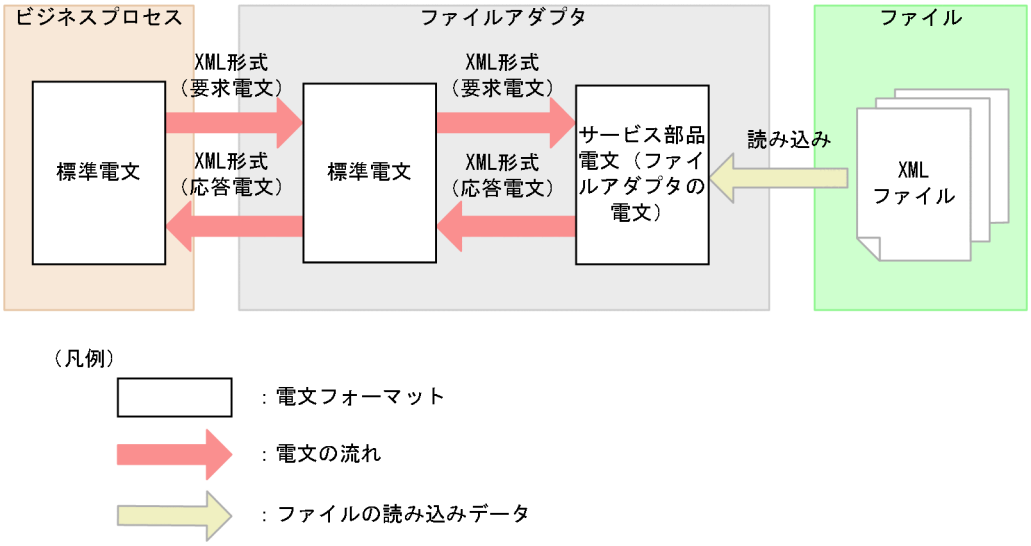
次に，XML フォーマット定義ファイルの作成方法をファイル操作モードごとに説明します。

読み込み用

ファイルアダプタで XML ファイルを読み込む場合，次の電文フォーマット定義ファイルを作成します。

- 要求電文：XML フォーマット定義ファイル
- 応答電文：XML フォーマット定義ファイル

図 3-7　XML ファイル読み込み時の電文の流れ



- 要求電文

  要求電文用の XML フォーマット定義ファイルは，次に示すファイルを使用します。

要求電文用フォーマット定義ファイル（XML・バイナリ形式読み込み用）
<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¥custom-adapter¥File¥schema¥adpff_read.xsd

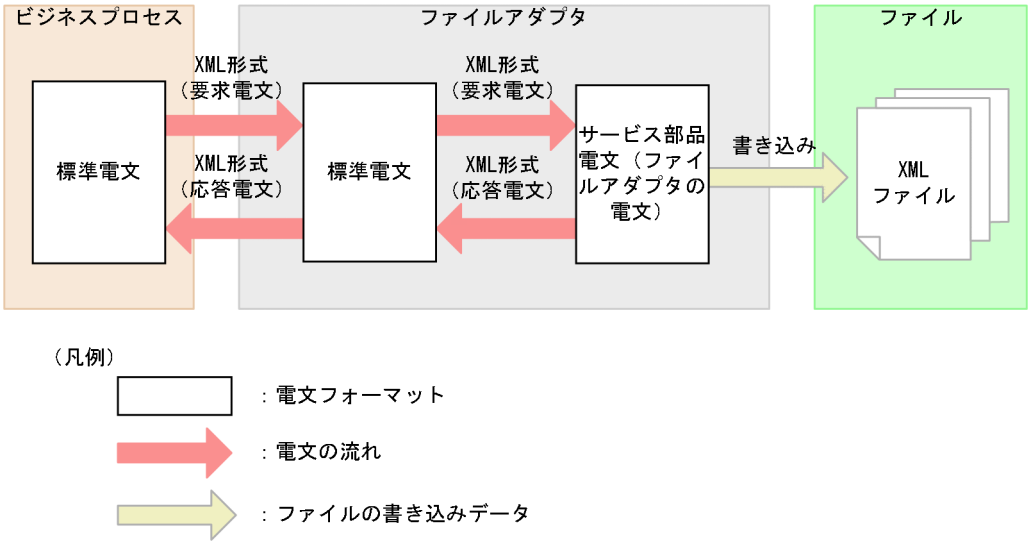
- 応答電文
  応答電文用の XML フォーマット定義ファイルは，ユーザが独自に作成します。

書き込み用

ファイルアダプタで XML ファイルに書き込む場合，次の電文フォーマット定義ファイルを作成します。

- 要求電文：XML フォーマット定義ファイル
- 応答電文：XML フォーマット定義ファイル

図 3-8　XML ファイル書き込み時の電文の流れ



- 要求電文
  要求電文用の XML フォーマット定義ファイルは次に示すファイルとテンプレートファイルを使用して作成します。
  要求電文用ファイル操作情報ファイル（XML 形式書き込み用）
  <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¥custom-adapter¥File¥schema¥adpff_header.xsd
  要求電文用フォーマット定義テンプレートファイル（XML 形式書き込み用）
  <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¥custom-adapter¥File¥schema¥templates¥adpff_write.xsd
  要求電文用フォーマット定義テンプレートファイルは，ユーザの編集が必要です。要求電文用フォーマット定義テンプレートファイルの編集個所を次の図に示します。

図 3-9　要求電文用フォーマット定義テンプレートファイル

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- All Rights Reserved. Copyright (C) 2007, Hitachi, Ltd. -->
3 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
4             xmlns:adpff="http://FlatFiles.cstmadp.csc.soft.Hitachi.co.jp/WriteRequest"
5             xmlns:<!-- prefix -->=<!-- Define targetNamespace. -->
6
7             targetNamespace=<!-- Define targetNamespace. -->>
8
9   <xsd:import namespace="http://FlatFiles.cstmadp.csc.soft.Hitachi.co.jp/WriteRequest"
10              schemaLocation="adpff_header.xsd"/>
11
12
13
14 <xsd:element name="WriteRequest" type="<!-- prefix -->:format"/>
15   <xsd:complexType name="format">
16     <xsd:sequence>
17       <xsd:element name="fileheader" type="adpff:fileheader"/>
18       <xsd:element name="userdata" type="<!-- prefix -->:userdata"/>
19     </xsd:sequence>
20   </xsd:complexType>
21
22   <xsd:complexType name="userdata">
23     <!-- Define XML schema of the user data. -->
24   </xsd:complexType >
25 </xsd:schema>
26
```

（凡例）

：ユーザが編集できる個所

図中の行番号に沿って，要求電文用フォーマット定義テンプレートファイルの説明を次の表に示します。なお，図中の行番号は実際のテンプレートには含まれません。

表 3-36　要求電文用フォーマット定義テンプレートファイルの説明

行数	説明
4 行目	要求電文用ファイル操作情報ファイル（XML 形式書き込み用：adpff_header.xsd）の名前空間宣言をしています。
5 行目	このテンプレートファイルの名前空間宣言をしています。ユーザは 7 行目に指定した targetNamespace 属性の値に任意の接頭辞を宣言します。
6 行目	ほかのスキーマファイルをインポートする場合，そのスキーマの名前空間宣言を記述します。※1，※2
7 行目	このテンプレートファイルの対象名前空間を記述しています。ユーザは targetNamespace 属性の値に一意となる対象名前空間を指定します。
9～10 行目	要求電文用ファイル操作情報ファイル（XML 形式書き込み用：adpff_header.xsd）のインポート宣言を記述しています。このテンプレートファイルは要求電文用ファイル操作情報ファイル（XML 形式書き込み用）をインポートします。ユーザは schemaLocation 属性の値を要求電文用ファイル操作情報ファイル（XML 形式書き込み用）の存在する相対パスに修正します。デフォルトでは同一ディレクトリを指定しています。
14～20 行目	WriteRequest 要素宣言を記述しています。要求電文のルート要素となります。ファイルアダプタは，この要素名を取得し，ファイル入出力種別を判別します（WriteRequest：書き込み，ReadRequest：読み込み）。fileheader 要素と userdata 要素を持ちます。ユーザは userdata 要素の接頭辞として 5 行目で宣言した接頭辞を記述します。
22 行目	userdata 要素宣言を記述しています。
23 行目	ユーザデータ用の XML スキーマ情報を記述します。※1，※2

注※1

記述方法は 2 種類あり，このファイルに直接記述する方法と，ユーザデータ用の XML スキーマ情報を記述するファイルを別に作成し，このファイルからインポートする方法があります。インポートする場合は 6 行目にインポートするスキーマのターゲットネームスペース，12 行目にインポートするスキーマのインポート文を記述してください。記述例は「(f)　電文例」を参照してください。

注※2

ユーザデータ用の XML スキーマ情報を記述する場合，userdata 要素の子要素には複数の要素を宣言できません。userdata 要素の子要素は XML ファイル書き込み時にルート要素となるため，userdata 要素の子要素は必ず 1 つだけ宣言してください。

- 応答電文

  応答電文用の XML フォーマット定義ファイルは次に示すファイルを使用します。

  応答電文用フォーマット定義ファイル（XML・バイナリ形式書き込み用）

  <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¥custom-adapter¥File¥schema¥adpff_result.xsd

(e)　バイナリフォーマット定義ファイルの作成

バイナリフォーマット定義ファイルの作成方法について説明します。

バイナリフォーマット定義ファイルを作成する場合に，ユーザが使用する定義ファイルおよび編集の必要性について次の表に示します。

表 3-37　ユーザが使用する定義ファイルおよび編集の必要性

ファイル操作モード	電文の種類	使用する定義ファイル	ユーザ編集
読み込み	要求電文	adpff_read.xsd	×
	応答電文	ユーザ作成	○
書き込み	要求電文	adpff_write.fdx	○
	応答電文	adpff_result.xsd	×

（凡例）

○：編集が必要です。

×：編集は不要です。

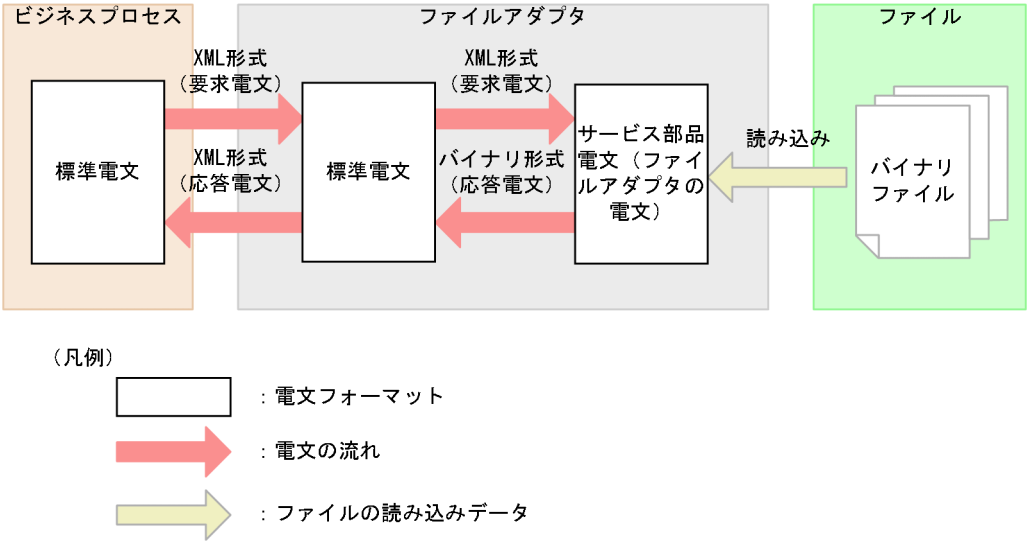
バイナリフォーマット定義ファイルの作成方法を次に示します。

●読み込み用

ファイルアダプタでバイナリファイルを読み込む場合，次の電文フォーマット定義ファイルを作成します。

- 要求電文：XML フォーマット定義ファイル
- 応答電文：バイナリフォーマット定義ファイル

図 3-10　バイナリファイル読み込み時の電文の流れ



要求電文

要求電文用の XML フォーマット定義ファイルは次に示すファイルを使用します。

要求電文用フォーマット定義ファイル（XML・バイナリ形式読み込み用）

<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¥custom-adapter¥File¥schema¥adpff_read.xsd

応答電文

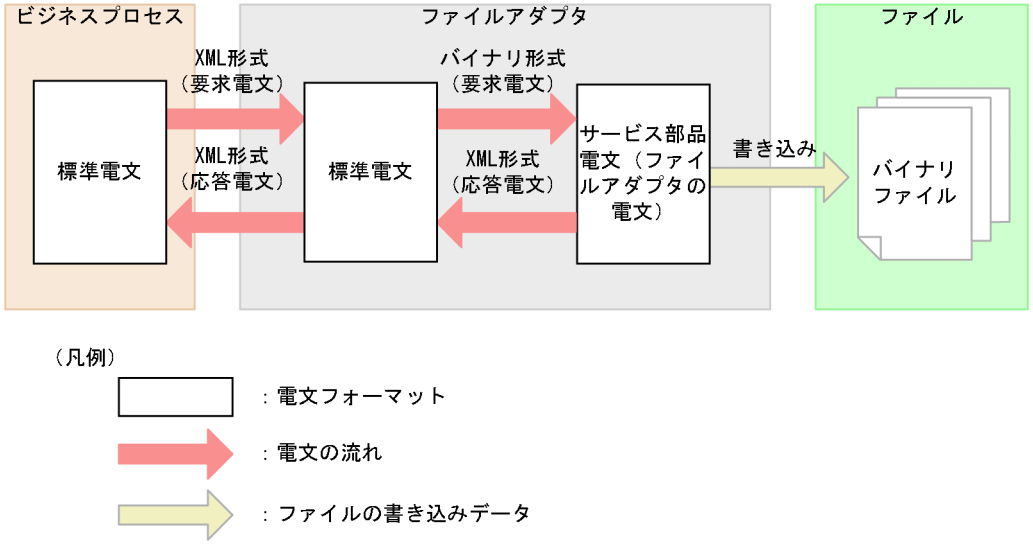
応答電文用のバイナリフォーマット定義ファイルはユーザが独自に作成します。

●書き込み用

ファイルアダプタでバイナリファイルを書き込む場合，次の電文フォーマット定義ファイルを作成します。

- 要求電文：バイナリフォーマット定義ファイル
- 応答電文：XML フォーマット定義ファイル

図 3-11　バイナリファイル書き込み時の電文の流れ



要求電文

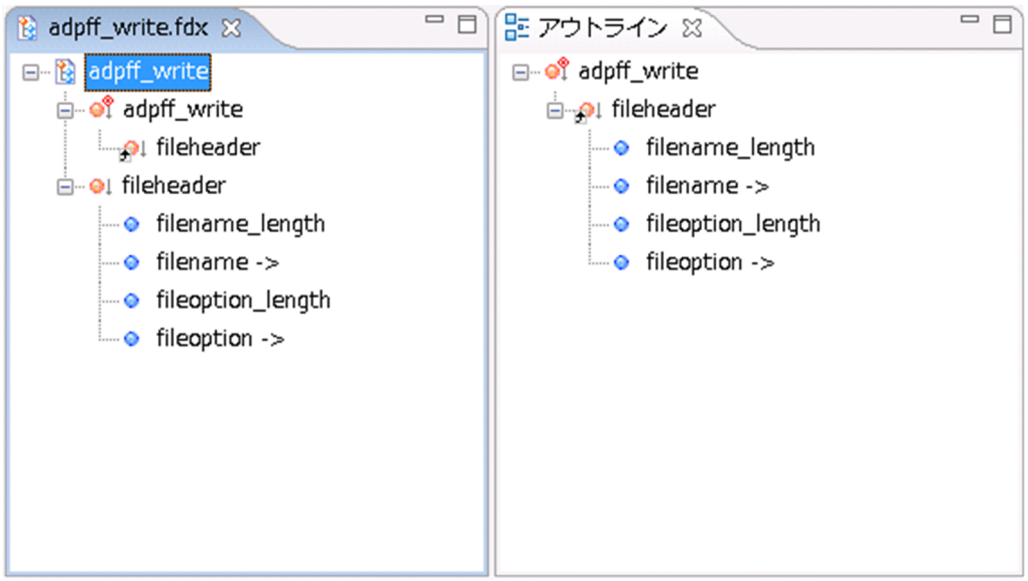
要求電文用のバイナリフォーマット定義ファイルは，次に示すテンプレートファイルをユーザがバイナリフォーマット定義画面で編集することによって作成します。

要求電文用フォーマット定義テンプレートファイル（バイナリ形式書き込み用）

<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¥custom-adapter¥File¥schema¥templates¥adpff_write.fdx

要求電文用フォーマット定義テンプレートファイルの画面を次の図に示します。

図 3-12　要求電文用フォーマット定義テンプレートファイルの画面



要求電文用フォーマット定義テンプレートファイルの設定値を次の表に示します。

表 3-38　要求電文用フォーマット定義テンプレートファイルの設定値

要素名	データ種別	サイズ（byte）	出現回数	説明
adpff_write	−	−	−	ルート要素
fileheader	−	−	1 回	ファイル操作情報要素
filename_length	整数	10	1 回	ファイル名長要素
filename	文字列	可変	1 回	ファイル名要素
fileoption_length	整数	10	1 回	ファイルオプション長要素
fileoption	文字列	可変	1 回	ファイルオプション要素

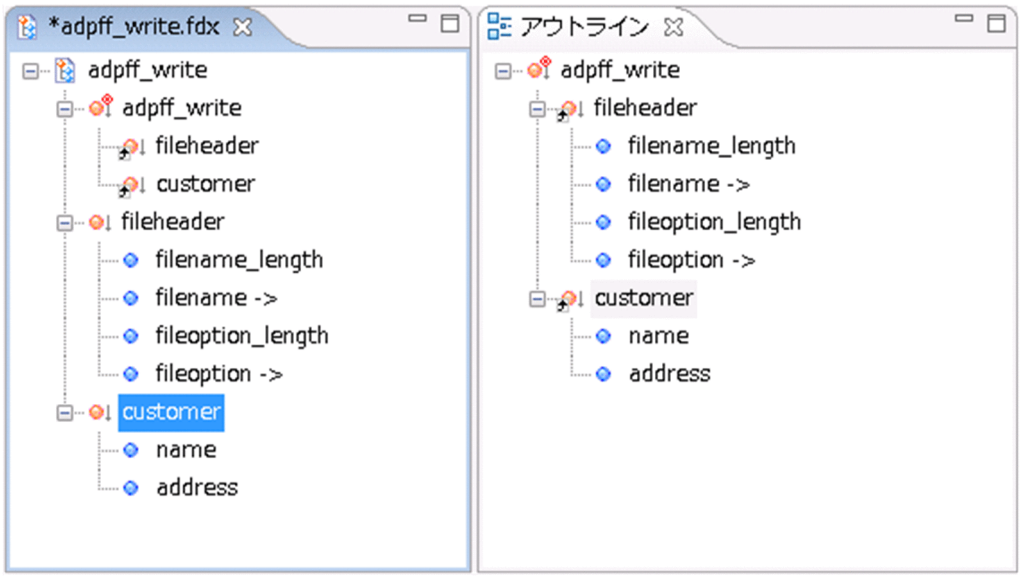
（凡例）

−：該当しません。

- **作成方法**

ユーザはルート要素（adpff_write 要素）の子要素として任意の要素名のユーザデータ要素を作成します。ただし，必ず fileheader 要素が先頭となるように作成してください。fileheader 要素が先頭でない場合の動作は保証されません。バイナリフォーマット定義ファイルを作成したあと，任意の名称で保存し，サービスアダプタ定義画面で指定します。作成完了後のバイナリフォーマット定義ファイルのサンプル画面を次の図に示します。サンプル画面ではユーザデータ要素として customer 要素を宣言し，その子要素として name 要素と address 要素を作成しています。

図 3-13　バイナリフォーマット定義ファイル作成後のサンプル画面



応答電文

応答電文用の XML フォーマット定義ファイルは次に示すファイルを使用します。

応答電文用フォーマット定義ファイル（XML・バイナリ形式書き込み用）

<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¥custom-adapter¥File¥schema¥adpff_result.xsd

(f) 電文例

各ファイル操作を行う場合の要求電文の例を次に示します。

電文例を説明するに当たって，要求電文フォーマットを定義するのに使用する要求電文用フォーマット定義ファイルの一覧を次の表に示します。

表 3-39 提供される要求電文用フォーマット定義ファイル一覧

ファイル名	内容	編集
adpff_read.xsd	要求電文用フォーマット定義ファイル（XML・バイナリ形式読み込み用）	×
adpff_write.xsd	要求電文用フォーマット定義テンプレートファイル（XML 形式書き込み用）	○
adpff_write.fdx	要求電文用フォーマット定義テンプレートファイル（バイナリ形式書き込み用）	○

（凡例）

○：編集して使用するファイルです。

×：編集しないでそのまま使用するファイルです。

●ファイル読み込み時の要求電文

ファイル読み込み時の要求電文の要求電文フォーマットには，要求電文用フォーマット定義ファイル（XML・バイナリ形式読み込み用）を使用します。要求電文フォーマット定義ファイルの内容は変更しないでそのまま使用してください。

ファイル読み込み時の要求電文の記述例を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<adpff:ReadRequest xmlns:adpff="http://FlatFiles.cstmadp.csc.soft.Hitachi.co.jp/ReadRequest">
  <fileheader>
    <filename>C:¥WORK¥ReadFile.xml</filename>
    <filemode>all</filemode>
  </fileheader>
</adpff:ReadRequest>
```

●ファイル書き込み時の要求電文（XML 形式）

XML 形式でのファイル書き込み時の要求電文の要求電文フォーマットには，要求電文用フォーマット定義テンプレートファイル（XML 形式書き込み用）を使用します。要求電文用フォーマット定義ファイルに，ユーザデータの XML スキーマ情報を追記して，要求電文用フォーマット定義ファイルを作成します。

ユーザデータの XML スキーマ情報を追記するには次の 2 つの方法があります。

- 要求電文用フォーマット定義ファイルに XML スキーマ情報を直接記述する
- 要求電文用フォーマット定義ファイルに XML スキーマ情報をインポートする

ユーザデータの XML スキーマ情報を直接記述する場合

要求電文用フォーマット定義ファイルに，ユーザデータ用の XML スキーマ情報を直接記述する場合の例を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- All Rights Reserved. Copyright (C) 2007, Hitachi, Ltd. -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:adpff="http://FlatFiles.cstmadp.csc.soft.Hitachi.co.jp/WriteRequest"
            xmlns:wr="http://WriteRequest"
            targetNamespace="http://WriteRequest">
  <xsd:import namespace="http://FlatFiles.cstmadp.csc.soft.Hitachi.co.jp/WriteRequest"
              schemaLocation="adpff_header.xsd"/>
  <xsd:element name="WriteRequest" type="wr:format"/>

  <xsd:complexType name="format">
    <xsd:sequence>
      <xsd:element name="fileheader" type="adpff:fileheader"/>
      <xsd:element name="userdata" type="wr:userdata"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="userdata">
    <xsd:sequence>
      <xsd:element name="customer" type="wr:customer"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="customer">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string" />
      <xsd:element name="address" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

上記の XML スキーマに適合した要求電文の記述例を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<wr:WriteRequest xmlns:wr="http://WriteRequest">
  <fileheader>
    <filename>C:¥WORK¥writeXML.xml</filename>
    <filemode>new</filemode>
  </fileheader>
  <userdata>
    <customer>
      <name>日立太郎</name>
      <address>横浜市戸塚区</address>
    </customer>
  </userdata>
</wr:WriteRequest>
```

ユーザデータの XML スキーマ情報をインポートする場合

要求電文用フォーマット定義ファイルに，ユーザデータ用の XML スキーマ情報をインポートして記述する場合の例を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- All Rights Reserved. Copyright (C) 2007, Hitachi, Ltd. -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:adpff="http://FlatFiles.cstmadp.csc.soft.Hitachi.co.jp/WriteRequest"
            xmlns:wr="http://WriteRequest"
            targetNamespace="http://WriteRequest"
            xmlns:usr="http://UserSchema">

  <xsd:import namespace="http://FlatFiles.cstmadp.csc.soft.Hitachi.co.jp/WriteRequest"
              schemaLocation="adpff_header.xsd"/>

  <xsd:import namespace="http://UserSchema" schemaLocation="User.xsd"/>

  <xsd:element name="WriteRequest" type="wr:format"/>
  <xsd:complexType name="format">
    <xsd:sequence>
      <xsd:element name="fileheader" type="adpff:fileheader"/>
      <xsd:element name="userdata" type="wr:userdata"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="userdata">
    <xsd:sequence>
      <xsd:element ref="usr:customer"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

インポートされる User.xsd は次の内容です。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:usr="http://UserSchema"
            targetNamespace="http://UserSchema">
  <xsd:element name="customer">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="name" type="xsd:string"/>
        <xsd:element name="address" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

上記の XML スキーマに適合した要求電文の記述例を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<wr:WriteRequest xmlns:wr="http://WriteRequest">
  <fileheader>
    <filename>C:¥WORK¥writeXML.xml</filename>
    <filemode>new</filemode>
  </fileheader>
  <userdata>
    <usr:customer xmlns:usr="http://UserSchema">
```

```
        <name>日立太郎</name>
        <address>横浜市戸塚区</address>
      </usr:customer>
    </userdata>
</wr:WriteRequest>
```

●ファイル書き込み時の要求電文（バイナリ形式）

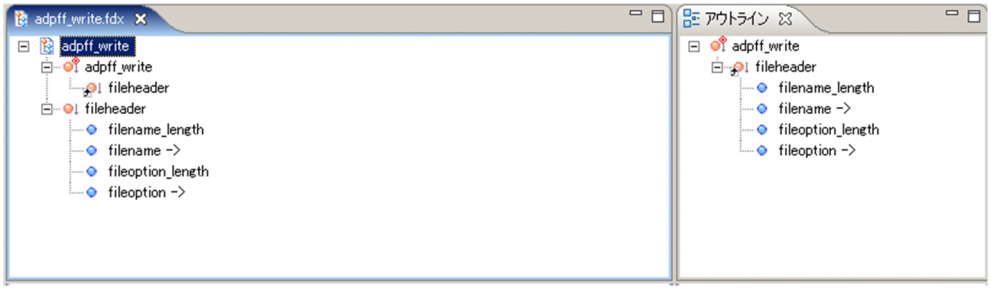
バイナリ形式でのファイル書き込み時の要求電文の要求電文フォーマットには，要求電文用フォーマット定義テンプレートファイル（バイナリ形式書き込み用）を使用します。要求電文用フォーマット定義ファイルを，Eclipse のバイナリフォーマット定義ファイル作成用のプロジェクトにインポートして，ユーザデータ用の定義を追加します。

ビジネスプロセス上では，要求電文が XML 形式で扱われるため，データ変換機能を使用して XML 形式からバイナリ形式に変換することで，ファイルアダプタのバイナリ用の要求電文に変換します。

要求電文用フォーマット定義ファイルの定義例，および要求電文のデータ変換の定義例を次に示します。

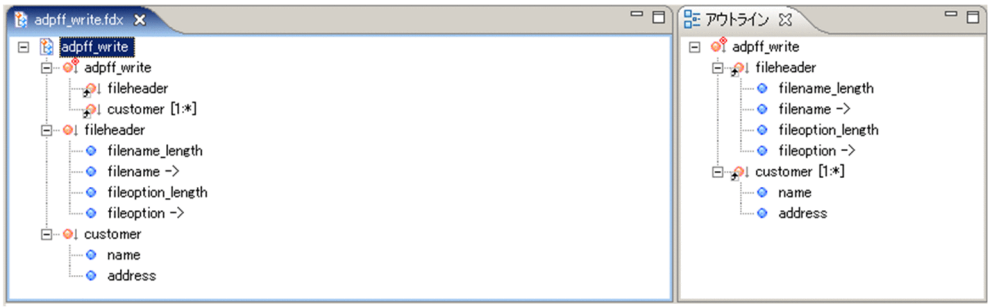
提供されている要求電文用フォーマット定義ファイルをインポートすると，ファイルアダプタのファイル操作情報の定義が表示されます。

図 3-14　提供ファイルをインポートした定義画面



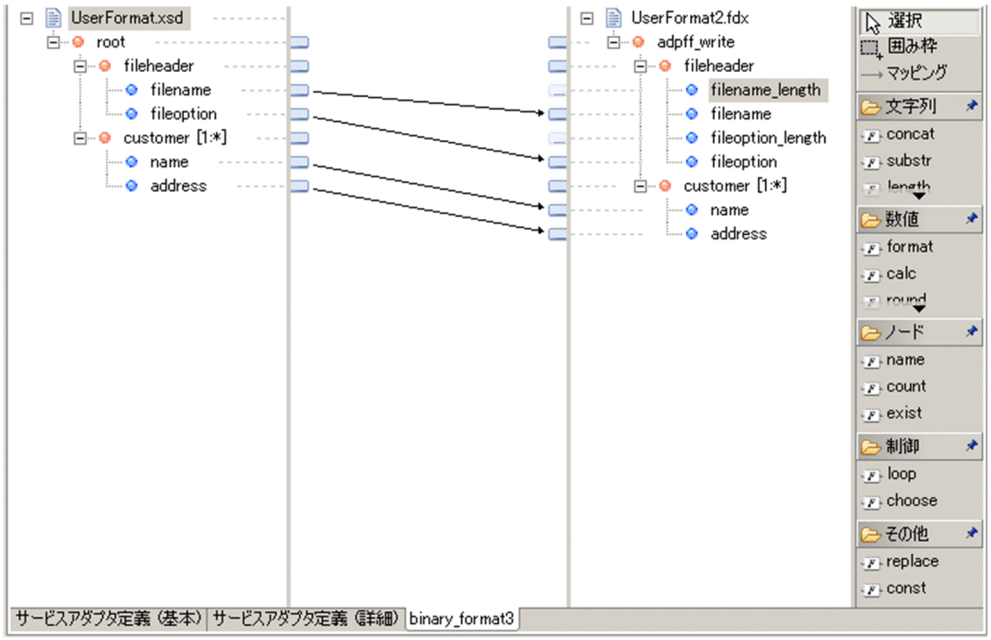
この定義画面に，ユーザデータの定義を追加します。次の例では，customer データとして，name と address の要素を定義しています。

図 3-15　ユーザデータの定義を追加した定義画面



ファイルアダプタの呼び出し元の要求電文フォーマットから，作成したバイナリフォーマット定義ファイルへデータ変換します。filename_length および fileoption_length 要素には，データ変換の機能で自動的に，filename および fileoption 要素のデータ長が設定されます。

図 3-16　データ変換定義画面



(2)　データ変換の定義

データ変換（マッピング）定義画面で，変換元の電文フォーマット定義ファイルおよび変換先の電文フォーマット定義ファイルを設定して，データ変換を定義します。

データ変換の定義方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.　データ変換の定義」を参照してください。

(3)　サービスアダプタ定義画面での操作

ファイルアダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。
   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービス部品制御情報の［サービス名］および［サービス ID］を必要に応じて編集します。

3. サービス部品制御情報の［追加］ボタンをクリックして，定義パターンおよびオペレーション名を指定します。

4. オペレーション情報の［通信モデル］のドロップダウンリストから，「同期」を選択します。

5. 要求電文について，手順 6.から手順 15.までを実施します。

6. 次に示す操作をします。

   標準電文のフォーマットを指定する場合

   手順 7.～9.を実施して，手順 10.に進みます。

標準電文のフォーマットを指定しない場合

手順 10.に進みます。

7. 標準電文の［使う］チェックボックスをチェックします。

8. 標準電文の［参照］ボタンをクリックして，［電文フォーマット］に標準電文のフォーマットを指定します。

指定できる電文フォーマットの形式については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.5　XML スキーマの適用範囲」を参照してください。

9. 標準電文の［表示］ボタンをクリックします。

電文フォーマットが XML 形式の場合，標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。

10. 手順 3.で指定した定義パターンに応じて次の電文フォーマットが設定されていることを確認します。

定義パターンが「XML 読み込み」または「バイナリ読み込み」の場合

要求電文のサービス部品：adpff_read.xsd

応答電文のサービス部品：なし

定義パターンが「XML 書き込み」または「バイナリ書き込み」の場合

要求電文のサービス部品：なし

応答電文のサービス部品：adpff_result.xsd

11. サービス部品電文の［表示］ボタンをクリックします。

電文フォーマットが XML 形式の場合，サービス部品電文のフォーマットが表示されます。必要に応じて指定したサービス部品電文のフォーマットを確認してください。

12. 次に示す操作をします。

標準電文の［使う］チェックボックスをチェックした場合

手順 13.～15.を実施して，手順 16.に進みます。

標準電文の［使う］チェックボックスをチェックしない場合

手順 16.に進みます。

13. データ変換定義のファイル名を入力します。

14. ［編集］ボタンをクリックします。

データ変換定義画面が表示されます。初めて定義する場合は，［ルート要素選択］ダイアログが表示されます。

15. 標準電文とサービス電文の内容をマッピングします。

16. 応答電文についても，手順 6.から手順 15.までを実施します。

17. 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画面（詳細）が表示されます。

18. サービスアダプタ（EJB-JAR ファイル）の名称を確認します。

19. ユーティリティクラス（JAR ファイル）に次のユーティリティクラスが追加されていることを確認します。
    - adpffpc.jar（プロトコルコンバータアーカイブファイル）

20. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(4) ファイルアダプタ実行環境プロパティファイルの作成

ファイルアダプタ実行環境プロパティファイルは，サンプルファイルを利用して作成します。サンプルファイルの格納先を次に示します。

```
<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>¥CSC¥custom-adapter¥File¥config¥templates¥adpff_exe.properties
```

ファイルアダプタ実行環境プロパティファイルの作成手順を次に示します。

1. サンプルファイルをコピーして，次のディレクトリに格納します。

   ```
   <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>¥CSC¥custom-adapter¥File¥config
   ```

2. サンプルファイルの名称を<サービス ID>.properties に変更します。

3. 定義内容を編集して保存します。

ファイルアダプタ実行環境プロパティファイルは，ファイルアダプタを開始する際に実行環境に反映されます。ファイルアダプタ実行環境プロパティファイルの内容を変更する場合は，ファイルアダプタを停止してから作業をします。ファイルアダプタを再開始すると，変更内容が実行環境に反映されます。

ファイルアダプタ実行環境プロパティファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.10.1　ファイルアダプタ実行環境プロパティファイル」を参照してください。

3.3.8 Object Access アダプタを定義する

Object Access アダプタの定義方法について説明します。

(1) 電文フォーマットの作成

各電文のフォーマットについて説明します。

(a) 要求電文

要求電文の電文フォーマットを次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2"
xmlns:tns="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2">
  <xsd:element name="exec2">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="method_name" type="xsd:string"/>
        <xsd:element name="in_list" type="tns:OWB2.NameValueList"/>
        <xsd:element name="session_data" type="tns:OWB2.NameValueList"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="OWB2.NameValueList">
    <xsd:complexContent>
      <xsd:restriction base="tns:EComm2.NameValueList">
       <xsd:sequence>
         <xsd:element name="item" type="tns:EComm2.NameValue"
                      minOccurs="0" maxOccurs="unbounded"/>
       </xsd:sequence>
      <xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="EComm2.NameValueList">
    <xsd:sequence>
      <xsd:element name="item" type="tns:EComm2.NameValue"
                   minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="EComm2.NameValue">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="value" type="tns:CORBA.Any" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CORBA.Any">
    <xsd:sequence>
      <xsd:element name="type" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="value" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

表 3-40　要求電文フォーマット

項目	電文フォーマット
要求電文形式	<?xml version="1.0" encoding="UTF-8"?> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2" xmlns:tns="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2">

項目	電文フォーマット
要求電文形式	[オペレーション情報] [型定義情報]... </xsd:schema>
オペレーション情報	<xsd:element name="exec2"> <xsd:complexType> <xsd:sequence> [入力パラメタ情報]... </xsd:sequence> </xsd:complexType> </xsd:element>
入力パラメタ情報	<xsd:element name="IDLに定義したパラメタ名称" type="[型情報]"/>
型定義情報：any	<xsd:complexType name="CORBA.Any"> <xsd:sequence> <xsd:element name="type" type="xsd:string" minOccurs="1" maxOccurs="1"/> <xsd:element name="value" type="xsd:string" minOccurs="1" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>
型定義情報：struct	<xsd:complexType name="IDLに定義した構造体宣言名称"> <xsd:sequence> <xsd:element name="IDLに定義した構造体変数名称" type="[型情報]" minOccurs="1" maxOccurs="1"/>... </xsd:sequence> </xsd:complexType>
型定義情報：sequence	<xsd:complexType name="IDLに定義したシーケンス宣言名称"> <xsd:sequence> <xsd:element name="item" type="[型情報]" minOccurs="0" maxOccurs="0"/> </xsd:sequence> </xsd:complexType>
型情報：string	xsd:string
型情報：any	tns:CORBA.Any
型情報：struct	tns:IDLに定義した構造体宣言名称
型情報：sequence	tns:IDLに定義したシーケンス宣言名称

要求電文の電文サンプルを次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:exec2 xmlns:tns="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2">
  <method_name>Mtd1</method_name>
  <in_list>
    <item>
      <name>MP7G</name>
      <value>
        <type>long</type>
        <value>200</value>
      </value>
    </item>
    <item>
      <name>GP9AKJ</name>
      <value>
        <type>Ecomm2.OctSeq</type>
        <value>ABCDE</value>
      </value>
    </item>
  </in_list>
  <session_data>
    <item>
      <name>S03U48</name>
      <value>
        <type>long</type>
        <value>212</value>
      </value>
    </item>
    <item>
      <name>H4TF8L</name>
      <value>
        <type>Ecomm2.OctSeq</type>
        <value>OK</value>
      </value>
    </item>
  </session_data>
</tns:exec2>
```

(b) 応答電文

応答電文の電文フォーマットを次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2"
xmlns:tns="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2">
  <xsd:element name="exec2Response">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="result" type="tns:OWB2.NameValueList"/>
        <xsd:element name="out_rec" type="tns:OWB2.RecordValue"/>
        <xsd:element name="session_data" type="tns:OWB2.NameValueList"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="OWB2.NameValueList">
    <xsd:complexContent>
```

```
      <xsd:restriction base="tns:EComm2.NameValueList">
    <xsd:sequence>
          <xsd:element name="item" type="tns:EComm2.NameValue"
                       minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="EComm2.NameValueList">
    <xsd:sequence>
      <xsd:element name="item" type="tns:EComm2.NameValue"
                   minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="EComm2.NameValue">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="value" type="tns:CORBA.Any" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CORBA.Any">
    <xsd:sequence>
      <xsd:element name="type" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="value" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="OWB2.RecordValue">
    <xsd:complexContent>
      <xsd:restriction base="tns:EComm2.RecordValue">
    <xsd:sequence>
          <xsd:element name="name" type="tns:EComm2.NameList" minOccurs="1" maxOccurs="1"/>
          <xsd:element name="record" type="tns:EComm2.RecordList" minOccurs="1" maxOccurs="1
"/>
        </xsd:sequence>
      <xsd:restriction base="tns:EComm2.RecordValue">
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="EComm2.RecordValue">
    <xsd:sequence>
      <xsd:element name="name" type="tns:EComm2.NameList" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="record" type="tns:EComm2.RecordList" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
<xsd:complexType name="EComm2.NameList">
    <xsd:sequence>
      <xsd:element name="item" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="EComm2.RecordList">
    <xsd:sequence>
      <xsd:element name="item" type="tns:EComm2.ValueList"
                   minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="EComm2.ValueList">
    <xsd:sequence>
      <xsd:element name="item" type="tns:CORBA.Any" minOccurs="0" maxOccurs="unbounded"/>
```

```
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

表 3-41　応答電文フォーマット

項目	電文フォーマット
応答電文形式	<?xml version="1.0" encoding="UTF-8"?> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2" xmlns:tns="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2"> [オペレーション情報] [型定義情報]... </xsd:schema>
オペレーション情報	<xsd:element name="exec2Response"> <xsd:complexType> <xsd:sequence> [出力パラメタ情報]... </xsd:sequence> </xsd:complexType> </xsd:element>
出力パラメタ情報	<xsd:element name="IDLに定義したパラメタ名称" type="[型情報]"/> 注　この情報には，リターン情報を含みます。
型定義情報：any	<xsd:complexType name="CORBA.Any"> <xsd:sequence> <xsd:element name="type" type="xsd:string" minOccurs="1" maxOccurs="1"/> <xsd:element name="value" type="xsd:string" minOccurs="1" maxOccurs="1"/> </xsd:sequence> </xsd:complexType>
型定義情報：struct	<xsd:complexType name="IDLに定義した構造体宣言名称"> <xsd:sequence> <xsd:element name="IDLに定義した構造体変数名称" type="[型情報]" minOccurs="1" maxOccurs="1"/>... </xsd:sequence> </xsd:complexType>
型定義情報：sequence	<xsd:complexType name="IDLに定義したシーケンス宣言名称"> <xsd:sequence> <xsd:element name="item" type="[型情報]" minOccurs="0" maxOccurs="0"/> </xsd:sequence> </xsd:complexType>

項目	電文フォーマット
型情報：string	xsd:string
型情報：any	tns:CORBA.Any
型情報：struct	tns:IDLに定義した構造体宣言名称
型情報：sequence	tns:IDLに定義したシーケンス宣言名称

応答電文の電文サンプルを次に示します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<tns:exec2Response xmlns:tns="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2">
  <result>
    <item>
      <name>MP7G</name>
      <value>
        <type>long</type>
        <value>200</value>
      </value>
    </item>
    <item>
      <name>GP9AKJ</name>
      <value>
        <type>Ecomm2.OctSeq</type>
        <value>ABCDE</value>
      </value>
    </item>
  </result>
  <out_rec>
    <name />
    <record />
  </out_rec>
  <session_data>
    <item>
      <name>S03U48</name>
      <value>
        <type>long</type>
        <value>212</value>
      </value>
    </item>
    <item>
      <name>H4TF8L</name>
      <value>
        <type>Ecomm2.OctSeq</type>
        <value>OK</value>
      </value>
    </item>
    <item>
      <name>GW:status</name>
      <value>
        <type>long</type>
```

```
        <value>0</value>
      </value>
    </item>
  </session_data>
</tns:exec2Response>
```

(c) フォルト電文

フォルト電文の電文フォーマットを次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2"
xmlns:tns="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2">
  <xsd:element name="EComm2.Exceptionexec2Fault1">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="code" type="xsd:int"/>
        <xsd:element name="message" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

表 3-42　フォルト電文フォーマット

項目	電文フォーマット
フォルト電文形式	<?xml version="1.0" encoding="UTF-8"?> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2" xmlns:tns="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2"> [例外定義情報] </xsd:schema>
例外定義情報	<xsd:element name="exec2Fault1"> <xsd:complexType> <xsd:sequence> [例外情報]... </xsd:sequence> </xsd:complexType> </xsd:element>
例外情報	<xsd:element name="IDL の例外に定義した変数名称" type="[型情報]"/>
型情報：long	xsd:int
型情報：string	xsd:string

フォルト電文の電文サンプルを次に示します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<tns:exec2Fault1 xmlns:tns="http://ObjectAccess.cstmadp.csc.soft.Hitachi.co.jp/ow/OWB2">
  <code>0</code>
  <message>no such method:Resource 'Base' does not contain method 'Mtd1'.</message>
</tns:exec2Fault1>
```

(d) Object Wrapper 提供 IDL

Object Wrapper 提供の IDL のサンプルを次に示します。

```
module EComm2 {
    typedef sequence<octet>       OctSeq;
    typedef sequence<OctSeq>      NameList;
    typedef sequence<any> ValueList;
    typedef sequence<ValueList>   RecordList;
    struct Binary {
        OctSeq   content;
    };
    struct NameValue {
        OctSeq   name;
        any      value;
    };
    typedef sequence<NameValue>   NameValueList;
    struct RecordValue {
        NameList name;
        RecordList        record;
    };
    exception _Exception
    {
        long     code;
        OctSeq   message;
    };
};
interface OWB2
{
    typedef EComm2::NameValueList        NameValueList;
    typedef EComm2::RecordValue            RecordValue;
    NameValueList exec2 (
        in string                          method_name,
        in NameValueList  in_list,
        out RecordValue           out_rec,
        inout NameValueList       session_data)
        raises ( EComm2::_Exception );
};
```

(2) Object Access アダプタ通信構成定義ファイルの作成

Object Access アダプタの場合，Object Access アダプタ通信構成定義ファイルを作成する必要があります。

Object Access アダプタ通信構成定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.7.1　Object Access アダプタ通信構成定義ファイル」を参照してください。

(3) データ変換の定義

データ変換（マッピング）定義画面で，変換元の電文フォーマット定義ファイルおよび変換先の電文フォーマット定義ファイルを設定して，データ変換を定義します。

データ変換の定義方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7. データ変換の定義」を参照してください。

(4) ユーザ定義ファイルの作成

ユーザ定義ファイルは，次の定義ファイルを jar コマンドでアーカイブして作成します。

- TPBroker 環境定義ファイル
- Object Access アダプタ通信構成定義ファイル

ユーザ定義ファイルの作成手順を説明します。

1. コマンドプロンプトを起動します。

   [スタート] メニューの [プログラム] から [アクセサリ] − [コマンド プロンプト] でコマンドプロンプトを起動します。

2. ユーザ定義ファイルを作業ディレクトリへコピーします。

   任意の作業ディレクトリに，次のディレクトリ構成を作成して，ユーザ定義ファイルをコピーします。

   ```
   jp¥co¥Hitachi¥soft¥csc¥cstmadp¥ObjectAccess¥cmd¥def
   ```

   「C:¥conf」ディレクトリに保存されている各定義ファイルを，「C:¥work」ディレクトリを作業ディレクトリとしてコピーする場合の例を次に示します。

   ```
   >XCOPY C:¥conf¥adpoa_com.xml C:¥work¥jp¥co¥Hitachi¥soft¥csc¥cstmadp¥ObjectAccess¥cmd¥def¥
    /V
   >XCOPY C:¥conf¥adpoa_tpb.properties C:¥work¥jp¥co¥Hitachi¥soft¥csc¥cstmadp¥ObjectAccess¥c
   md¥def¥ /V
   ```

3. ユーザ定義ファイルをアーカイブします。

   サービスプラットフォームが提供する jar コマンドを実行して，ユーザ定義ファイルを作成します。

   「Object Access アダプタ通信構成定義ファイル」だけをアーカイブする場合と，「Object Access アダプタ通信構成定義ファイル」および「TPBroker 環境定義ファイル」をアーカイブする場合の例を次に示します。ここで示す例では，「C:¥work」ディレクトリを作業ディレクトリとして，ユーザ定義ファイルを「C:¥conf」に作成します。

   Object Access アダプタ通信構成定義ファイルだけをアーカイブする場合

   ```
   > jar cf C:¥conf¥adpoadef.jar
     -C C:¥work jp¥co¥Hitachi¥soft¥csc¥cstmadp¥ObjectAccess¥cmd¥def¥adpoa_com.xml
   ```

   Object Access アダプタ通信構成定義ファイルおよび TPBroker 環境定義ファイルをアーカイブする場合

```
> jar cf C:¥conf¥adpoadef.jar
  -C C:¥work jp¥co¥Hitachi¥soft¥csc¥cstmadp¥ObjectAccess¥cmd¥def¥adpoa_com.xml
  -C C:¥work jp¥co¥Hitachi¥soft¥csc¥cstmadp¥ObjectAccess¥cmd¥def¥adpoa_tpb.properties
```

4. 作業ディレクトリを削除します。

   RMDIR コマンドを実行して，ユーザ定義ファイルのアーカイブ用に作成したディレクトリを削除します。

   「C:¥work」ディレクトリを作業ディレクトリとした場合の例を次に示します。

```
> RMDIR /S /Q C:¥work¥jp
```

注意事項を次に示します。

- Object Access アダプタ通信構成定義ファイル名称が，「adpoa_com.xml」であることを確認してください。ほかのファイル名称を指定した場合，実行環境で正常に動作しません。
- TPBroker 環境定義ファイル名称が，「adpoa_tpb.properties」であることを確認してください。ほかのファイル名称を指定した場合，実行環境で正常に動作しません。
- ユーザ定義ファイルの作成で使用する作業ディレクトリおよびユーザ定義ファイルの作成先ディレクトリは，書き込み権限を持つディレクトリを指定してください。

(5)　サービスアダプタ定義画面での操作

Object Access アダプタを定義する場合の手順を次に示します。設定内容の詳細は，「3.3.19(8)　Object Access アダプタの場合」を参照してください。

1. サービスアダプタ定義画面を表示します。

   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービス部品制御情報の［サービス名］および［サービス ID］を必要に応じて編集します。

3. サービス部品制御情報の［追加］ボタンをクリックして，オペレーションを追加します。

   任意のフォルト名を指定してください。

4. オペレーション情報の［通信モデル］のドロップダウンリストから，「同期」を選択します。

5. 要求電文について，手順 6.から手順 15.までを実施します。

6. 次に示す操作をします。

   標準電文のフォーマットを指定する場合

   手順 7.～9.を実施して，手順 10.に進みます。

   標準電文のフォーマットを指定しない場合

   手順 10.に進みます。

7. 標準電文の［使う］チェックボックスをチェックします。

8. 標準電文の［参照］ボタンをクリックして，［電文フォーマット］に標準電文のフォーマットを指定します。

   指定できる電文フォーマットの形式については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.5　XML スキーマの適用範囲」を参照してください。

9. 標準電文の［表示］ボタンをクリックします。

   電文フォーマットが XML 形式の場合，標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。

10. サービス部品電文に次の電文フォーマットが設定されていることを確認します。

    - 要求電文：exec2_request.xsd
    - 応答電文：exec2_response.xsd

11. サービス部品電文の［表示］ボタンをクリックします。

    サービス部品電文のフォーマットが表示されます。必要に応じてサービス部品電文のフォーマットを確認してください。

12. 次に示す操作をします。

    標準電文の［使う］チェックボックスをチェックした場合

    手順 13.～15.を実施して，手順 16.に進みます。

    標準電文の［使う］チェックボックスをチェックしない場合

    手順 16.に進みます。

13. データ変換定義のファイル名を入力します。

14. ［編集］ボタンをクリックします。

    データ変換定義画面が表示されます。初めて定義する場合は，［ルート要素選択］ダイアログが表示されます。

15. 標準電文とサービス電文の内容をマッピングします。

16. 応答電文についても，手順 6.から手順 15.までを実施します。

17. フォルト電文の電文フォーマットに「exec2_fault1.xsd」が設定されていることを確認します。

18. サービスアダプタ定義（詳細）タブをクリックします。

    サービスアダプタ定義画面（詳細）が表示されます。

19. サービスアダプタ（EJB-JAR ファイル）およびユーティリティクラス（JAR ファイル）の名称を確認します。

    サービスアダプタ（EJB-JAR ファイル）の名称が「cscmsg_adpejb.jar」，ユーティリティクラス（JAR ファイル）の名称が「adpoapc.jar」であることを確認してください。

20. ユーティリティクラス（JAR ファイル）の［追加］ボタンをクリックして，次のユーティリティクラスを追加します。
    - adpoadef.jar（ユーザ定義ファイル）
21. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(6) Object Access アダプタ実行環境プロパティファイルの作成

Object Access アダプタの場合，Object Access アダプタ実行環境プロパティファイルを作成する必要があります。

Object Access アダプタ実行環境プロパティファイルの詳細は，マニュアル「サービスプラットフォームリファレンス」の「6.11.1　Object Access アダプタ実行環境プロパティファイル」を参照してください。

(7) TPBroker 環境定義ファイルの作成

Object Access アダプタの ORB 動作の設定は，プロパティ形式の TPBroker 環境定義ファイルとして，ユーザが独自に作成します。定義ファイル名称は"adpoa_tpb.properties"とし，ユーザ定義ファイルにアーカイブします。TPBroker 環境定義ファイルは，ユーザ定義ファイルにアーカイブ後に，HCSC 開発環境でパッケージングし，実行環境へインポートすることで実行環境に反映されます。TPBroker 環境定義ファイルの内容を変更する場合は，ユーザ定義ファイルの再アーカイブ後に，HCSC 開発環境で再度パッケージングして実行環境へインポートします。なお，ORB 動作の設定が必要ない場合は，TPBroker 環境定義ファイルを作成する必要はありません。

TPBroker 環境定義ファイルに設定した ORB 動作は，ほかの環境変数などの定義より優先して適用されます。

ORB の動作を定義します。同一のキーが存在する場合は，後述のキーの値が反映されます。行頭の＃や！はコメントを示します。

```
＜property name＞＝＜property value＞
```

定義できるプロパティについては，マニュアル「Borland(R) Enterprise Server VisiBroker(R) プログラマーズリファレンス」を参照してください。

Object Access アダプタでは，TPBroker の次のプロパティを標準で設定します。プロパティは，Object Access アダプタの TPBroker 環境定義ファイルで変更します。ただし，vbroker.agent.enableLocator の設定は変更できません。Object Access アダプタで，true を設定してサービス実行時にスマートエージェントと通信します。

表 3-43　設定するプロパティ

プロパティ	説明	設定値
vbroker.ce.iiop.ccm.htc.readerPerConnection	RelativeRoundtripTimeoutPolicy で指定したタイムアウト値が満了した場合，コネクションを切断するかを指定します。	true タイムアウト発生時にコネクションをクローズしません。
vbroker.agent.enableLocator	サービス実行時にスマートエージェントと通信するかを指定します。	true サービス実行時にスマートエージェントと通信します。 この設定は変更できません。

3.3.9　Message Queue アダプタを定義する

Message Queue アダプタの定義方法について説明します。

(1)　電文フォーマット定義ファイルの作成

Message Queue アダプタでは，要求電文と応答電文の 2 種類の電文があります。それぞれの内容について説明します。

電文フォーマット作成手順の詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.　電文フォーマットの作成」を参照してください。

Message Queue アダプタの要求電文・応答電文には，XML 形式またはバイナリ形式メッセージが指定できます。

XML 形式のメッセージを送信する場合は XML 形式電文，バイナリメッセージを送信する場合はバイナリ形式電文だけを指定できます。また，XML 形式メッセージを受信する場合は XML 形式電文，バイナリメッセージを受信する場合はバイナリ形式電文だけを指定できます。

注意事項

送信メッセージまたは受信メッセージのユーザデータは 1 バイト以上を指定してください。長さ 0 バイトのユーザデータはエラーになります。

(a)　要求電文

Message Queue アダプタの呼び出し元から，送信キューへのメッセージ送信および受信キューからのメッセージ受信の実行を要求するための電文を要求電文といいます。

要求電文として送信する情報を説明します。

キュー操作情報

キュー操作情報はキューにアクセスするために必要な情報です。キュー操作情報の内容を次に示します。

表 3-44　キュー操作情報

情報	説明
送信キュー特定情報，受信キュー特定情報	アクセスする送信キューまたは受信キューを指定します。アプリケーション統合属性ファイルの<resource-env-ref>タグ内にある<resource-env-ref-name>タグで指定した resource-env-ref の名称を指定します。
メッセージ相関識別子	メッセージ受信要求またはブラウズ・メッセージ受信要求時，メッセージ相関識別子と一致するメッセージを受信する場合に指定します。指定値は 24 バイト以内でメッセージ送信応答時に設定されるメッセージ相関識別子を指定します。メッセージ相関識別子のフォーマットについては，マニュアル「サービスプラットフォーム 解説」の「2.10.2　Message Queue アダプタを使ったメッセージキューへのアクセス」のメッセージ相関識別子の生成方法に関する説明を参照してください。

送信メッセージ

XML 形式またはバイナリ形式メッセージを送信要求する場合に要求電文に指定するユーザデータです。

(b)　要求電文の内容

要求電文の内容は，送信キュー特定情報および受信キュー特定情報の有無やメッセージ相関識別子の指定有無および送信メッセージ形式（バイナリ形式または XML 形式）のうち，必要な情報だけを電文に記述します。

●メッセージ送信要求の場合

XML 形式で送信キュー特定情報を指定する場合の要求電文形式を次に示します。

表 3-45　XML 形式の送信要求電文形式（送信先キュー特定情報指定あり）

情報	電文内のタグ名	指定値	必須
ルート要素	SendRequest	－	○
ヘッダ情報	mqheader	－	○
送信キュー特定情報	sndresname	送信キュー特定情報※	○
送信メッセージ	userdata	ユーザ任意	○

（凡例）

○：必ず指定します。

－：指定値はありません。

注※

Message Queue アダプタ通信構成定義の<res_name>タグを省略した場合，この送信キュー特定情報は指定必須です。
Message Queue アダプタ通信構成定義の<res_name>タグを指定した場合，この指定は無視され，Message Queue アダプタ通信構成定義の指定値が使用されます。

XML 形式で送信キュー特定情報を指定しない場合の要求電文形式を次に示します。

表 3-46　XML 形式の送信要求電文形式（送信先キュー特定情報指定なし）

情報	電文内のタグ名	指定値	必須
ルート要素	SendRequest	−	○
ヘッダ情報	mqheader	−	○
送信メッセージ	userdata	ユーザ任意	○

（凡例）

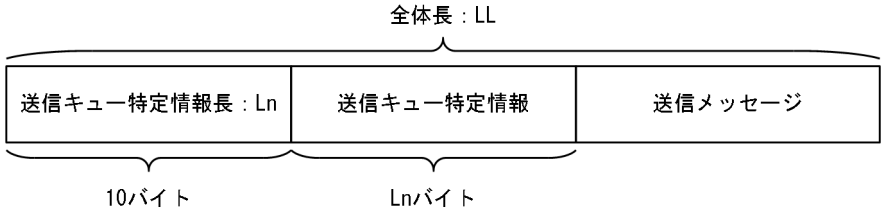
○：必ず指定します。

−：指定値はありません。

バイナリ形式で送信キュー特定情報を指定する場合について説明します。

バイナリメッセージを送信する場合は，要求電文がバイナリ形式になります。バイナリ形式の要求電文の構造を次に示します。

図 3-17　バイナリ形式の送信要求電文の構造



電文の全体長は次の関係式が成り立つ必要があります。

```
LL ≧ 10 + Ln
```

バイナリメッセージを送信する場合に，要求電文に必要な情報を次に示します。

表 3-47　バイナリ形式の送信要求電文形式（送信先キュー特定情報指定あり）

情報	電文内のタグ名	指定値	必須
送信キュー特定情報長	送信キュー特定情報のバイト長を 10 進数文字列で指定	10 バイト	○
送信キュー特定情報	送信キュー特定情報（文字列）	任意	○
送信メッセージ	ユーザ任意	任意	○

（凡例）

○：必ず指定します。

バイナリ形式で送信キュー特定情報を指定しない場合の要求電文形式を次に示します。

表 3-48　バイナリ形式の送信要求電文形式（送信先キュー特定情報指定なし）

情報	電文内のタグ名	指定値	必須
送信メッセージ	ユーザ任意	任意	○

（凡例）

○：必ず指定します。

●メッセージ受信要求の場合

バイナリ形式のメッセージ受信要求は指定できません。

XML 形式のメッセージ受信要求電文形式を次に示します。

表 3-49　XML 形式のメッセージ受信要求電文形式

情報	電文内のタグ名	指定値	必須
ルート要素	RecvRequest	－	○
ヘッダ情報	mqheader	－	○
受信キュー特定情報	rcvresname	受信キュー特定情報※1	×
メッセージ相関識別子	correlationid	受信メッセージの相関識別子※2	×

（凡例）

○：必ず指定します。

×：省略できます。

－：指定値はありません。

注※1

Message Queue アダプタ通信構成定義の<res_name>タグを省略した場合，この受信キュー特定情報は指定必須です。Message Queue アダプタ通信構成定義の<res_name>タグを指定した場合，この指定は無視され，Message Queue アダプタ通信構成定義の指定値が使用されます。

注※2

Message Queue アダプタ通信構成定義の<id_type>タグの指定値が Type2 の場合，このメッセージ相関識別子は指定必須です。Message Queue アダプタ通信構成定義の<id_type>タグを省略または Type1 を指定した場合，この指定は無視されます。

●ブラウズ・メッセージ受信要求の場合

バイナリ形式のブラウズ・メッセージ受信要求は指定できません。

XML 形式のブラウズ・メッセージ受信要求電文形式を次に示します。

表 3-50　XML 形式のブラウズ・メッセージ受信要求電文形式

情報	電文内のタグ名	指定値	必須
ルート要素	BrowseRequest	－	○
ヘッダ情報	mqheader	－	○

情報	電文内のタグ名	指定値	必須
受信キュー特定情報	rcvresname	受信キュー特定情報※1	×
メッセージ相関識別子	correlationid	受信メッセージの相関識別子※2	×

（凡例）

○：必ず指定します。

×：省略できます。

－：指定値はありません。

注※1

Message Queue アダプタ通信構成定義の<res_name>タグを省略した場合，この受信キュー特定情報は指定必須です。Message Queue アダプタ通信構成定義の<res_name>タグを指定した場合，この指定は無視され，Message Queue アダプタ通信構成定義の指定値が使用されます。

注※2

Message Queue アダプタ通信構成定義の<id_type>タグの指定値が Type2 の場合，このメッセージ相関識別子は指定必須です。Message Queue アダプタ通信構成定義の<id_type>タグを省略または Type1 を指定した場合，この指定は無視されます。

●メッセージ送受信要求の場合

XML 形式の送信キュー特定情報および受信キュー特定情報を指定する場合の要求電文形式を次に示します。

表 3-51　XML 形式のメッセージ送受信要求電文形式（送受信キュー特定情報指定あり）

情報	電文内のタグ名	指定値	必須
ルート要素	SndRcvRequest	－	○
ヘッダ情報	mqheader	－	○
送信キュー特定情報	sndresname	送信キュー特定情報※	○
受信キュー特定情報	rcvresname	受信キュー特定情報※	○
送信メッセージ	userdata	任意	○

（凡例）

○：必ず指定します。

－：指定値はありません。

注※

Message Queue アダプタ通信構成定義の<res_inf>タグを省略した場合，この送信キュー特定情報および受信キュー特定情報は指定必須です。Message Queue アダプタ通信構成定義の<res_inf>タグを指定した場合，この指定は無視され，Message Queue アダプタ通信構成定義の指定値が使用されます。

XML 形式の送信キュー特定情報および受信キュー特定情報を指定しない場合の要求電文形式を次に示します。

表 3-52　XML 形式のメッセージ送受信要求電文形式（送受信キュー特定情報指定なし）

情報	電文内のタグ名	指定値	必須
ルート要素	SndRcvRequest	−	○
ヘッダ情報	mqheader	−	○
送信メッセージ	userdata	任意	○

（凡例）

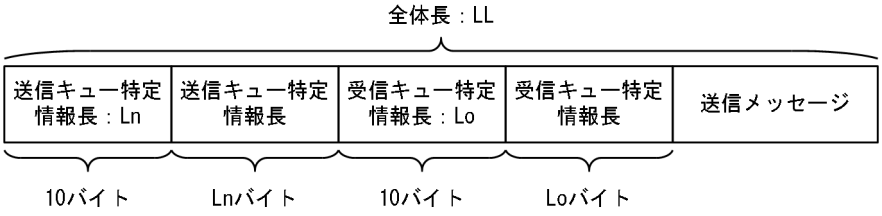
○：必ず指定します。

−：指定値はありません。

バイナリ形式の送信キュー／受信キュー特定情報を指定する場合について説明します。

バイナリメッセージを送受信する場合は，要求電文がバイナリ形式になります。バイナリ形式の要求電文の構造を次に示します。

図 3-18　バイナリ形式の送受信要求電文の構造



電文の全体長は次のような関係式が成り立つ必要があります。

```
LL ≧ 10 + Ln + 10 + Lo
```

バイナリメッセージを送受信する場合に，要求電文に必要な情報を次に示します。

表 3-53　バイナリ形式の送信要求電文形式（送信先キュー/受信キュー特定情報指定あり）

情報	電文内のタグ名	指定値	必須
送信キュー特定情報長	送信キュー特定情報のバイト長を 10 進数文字列で指定	10 バイト	○
送信キュー特定情報	送信キュー特定情報（文字列）	任意	○
受信キュー特定情報長	受信キュー特定情報のバイト長を 10 進数文字列で指定	10 バイト	○
受信キュー特定情報	受信キュー特定情報（文字列）	任意	○
送信メッセージ	ユーザ任意	任意	○

（凡例）

○：必ず指定します。

バイナリ形式の送信キュー／受信キュー特定情報を指定しない場合の要求電文形式を次に示します。

表 3-54　バイナリ形式の送信要求電文形式（送信先キュー／受信キュー特定情報指定なし）

情報	電文内のタグ名	指定値	必須
送信メッセージ	ユーザ任意	任意	○

（凡例）

○：必ず指定します。

(c)　応答電文

Message Queue アダプタの呼び出し元へ返却する電文を応答電文といいます。応答電文として返される情報を説明します。

受信メッセージ

XML 形式，またはバイナリ形式のメッセージを受信する場合，応答電文に格納されるユーザデータです。

メッセージ送信結果

XML 形式，またはバイナリ形式のメッセージを送信する場合，送信処理の結果が応答電文へ格納されます。送信処理が失敗した場合，Message Queue アダプタはカスタムアダプタ開発フレームワークへ例外を通知し応答電文は返しません。応答電文にメッセージ送信結果が格納されるのは，メッセージ送信が成功したときだけとなります。このため，メッセージ送信時の応答電文の返却値は必ず「OK」になります。

メッセージ相関識別子

XML 形式，またはバイナリ形式のメッセージの送信時，JMSCorrelationID ヘッダフィールドに設定したメッセージ相関識別子です。

(d)　応答電文の内容

応答電文の内容は，受信メッセージ形式（バイナリ形式または XML 形式）によって異なります。

なお，バイナリ形式のメッセージ送信応答は指定できません。

XML 形式の送信応答電文形式を次に示します。

●メッセージ送信要求の応答電文の場合

XML 形式の送信応答電文形式を次に示します。

表 3-55　XML 形式の送信応答電文形式

情報	電文内のタグ名	返却値
ルート要素	SendResponse	−
メッセージ送信結果	status	OK
メッセージ相関識別子	correlationid	メッセージ相関識別子※

（凡例）

－：返却値はありません。

注※

Message Queue アダプタ通信構成定義の<id_type>で Type2 を指定した場合に返却されます。

●メッセージ受信要求の応答電文の場合

XML 形式のメッセージ受信応答電文形式を次に示します。

表 3-56　XML 形式のメッセージ受信応答電文形式

情報	電文内のタグ名	返却値
受信メッセージ	ユーザ任意	ユーザ任意

バイナリ形式のメッセージ受信応答電文形式を次に示します。

表 3-57　バイナリ形式のメッセージ受信応答電文形式

情報	電文内のタグ名
受信メッセージ	ユーザ任意

●ブラウズ・メッセージ受信要求の応答電文の場合

XML 形式のブラウズ・メッセージ受信応答電文形式を次に示します。

表 3-58　XML 形式のブラウズ・メッセージ受信応答電文形式

情報	電文内のタグ名	返却値
受信メッセージ	ユーザ任意	ユーザ任意

バイナリ形式のブラウズ・メッセージ受信応答電文形式を次に示します。

表 3-59　バイナリ形式のブラウズ・メッセージ受信応答電文形式

情報	電文内のタグ名
受信メッセージ	ユーザ任意

●メッセージ送受信要求の応答電文

XML 形式のメッセージ送受信応答電文形式を次に示します。

表 3-60　XML 形式のメッセージ送受信応答電文形式

情報	電文内のタグ名	返却値
受信メッセージ	ユーザ任意	ユーザ任意

バイナリ形式のメッセージ送受信応答電文形式を次に示します。

表 3-61　バイナリ形式のメッセージ送受信応答電文形式

情報	電文内のタグ名
受信メッセージ	ユーザ任意

(e)　フォーマット定義ファイルの作成

各電文フォーマットのうち，サービス部品項目に設定する電文フォーマットが Message Queue アダプタの電文フォーマットです。Message Queue アダプタの電文フォーマットの作成方法を説明します。

XML 形式の要求電文

要求電文用の XML フォーマット定義ファイルは，テンプレートファイルを使用して作成するか，製品提供ファイルを使用します。それ以外の場合はユーザが独自に作成してください。

XML フォーマット定義テンプレートファイルを使用する場合

- メッセージの送信要求電文で送信キューを指定します。
- メッセージの送受信要求で送信キューおよび受信キューを指定します。

製品提供ファイルを使用する場合

- メッセージの受信要求電文で受信キューおよびメッセージ相関識別子を指定します。
- ブラウズ・メッセージ受信要求で受信キューおよびメッセージ相関識別子を指定します。

ユーザが独自に作成する場合

上記，XML フォーマット定義テンプレートファイルを使用する場合にも製品提供ファイルを使用する場合にも該当しない場合は，ユーザが独自に作成してください。

XML フォーマット定義テンプレートファイルとカスタマイズの説明を次に示します。

図 3-19　メッセージ送信要求電文用テンプレートファイル（adpmq_snd_request.xsd）

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!-- All Rights Reserved. Copyright (C) 2008, Hitachi, Ltd. -->
3  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
4                        xmlns:<!-- prefix -->=<!-- Define targetNamespace. -->
5
6                        targetNamespace=<!-- Define targetNamespace. -->>
7
8
9
10 <xsd:element name="SendRequest" type="<!-- prefix -->:format"/>
11     <xsd:complexType name="format">
12           <xsd:sequence>
13             <xsd:element name="mqheader" type="<!-- prefix -->:mqheader"/>
14             <xsd:element name="userdata" type="<!-- prefix -->:userdata"/>
15           </xsd:sequence>
16     </xsd:complexType>
17
18     <xsd:complexType name="mqheader">
19           <xsd:sequence>
20             <xsd:element name="sndresname" type="xsd:string" minOccurs="0"/>
21           </xsd:sequence>
22     </xsd:complexType>
23
24     <xsd:complexType name="userdata">
25         <!-- Define XML schema of the user data. -->
26     </xsd:complexType>
27 </xsd:schema>
```

（凡例）

：ユーザの編集個所

表 3-62　メッセージ送信要求電文用テンプレートファイルの内容

行数	説明
4 行目	このテンプレートファイルの名前空間宣言を表します。ユーザは 6 行目に指定した targetNamespace 属性の値に任意の接頭辞を宣言します。
6 行目	このテンプレートファイルの対象名前空間を表します。ユーザは targetNamespace 属性の値に一意となる対象名前空間を指定します。
8 行目	25 行目でユーザデータ用の XML スキーマ情報をインポートする場合，5 行目にインポートするスキーマのターゲットネームスペースを，8 行目にインポートするスキーマのインポート文を記述します。
10～16 行目	SendRequest 要素宣言を表します。要求電文のルート要素となる。mqheader 要素と userdata 要素を持ちます。ユーザは mqheader 要素と userdata 要素の接頭辞として 4 行目で宣言した接頭辞を記述します。
18～22 行目	mqheader 要素宣言を表します。
24 行目	userdata 要素宣言を表します。
25 行目	ユーザデータ用の XML スキーマ情報を記述します。 記述方法 記述方法は 2 種類あり，このファイルに直接記述する方法と，ユーザデータ用の XML スキーマ情報を記述するファイルを別に作成し，このファイルからインポートする方法があります。インポートする場合は 5 行目にインポートするスキーマのターゲットネームスペースを，8 行目にインポートするスキーマのインポート文を記述します。

行数	説明
25 行目	注意事項 ユーザデータ用の XML スキーマ情報を記述する際，userdata 要素の子要素には複数の要素を宣言してはいけません。userdata 要素の子要素は XML ファイル出力時にルート要素となる要素であるため，userdata 要素の子要素は必ず 1 つだけ宣言してください。

図 3-20　メッセージ送受信要求電文用フォーマット定義テンプレートファイル (adpmq_sndrcv_request.xsd)

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!-- All Rights Reserved. Copyright (C) 2008, Hitachi, Ltd. -->
3  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
4                        xmlns:<!-- prefix -->=<!-- Define targetNamespace. -->
5
6                        targetNamespace=<!-- Define targetNamespace. -->>
7
8
9
10 <xsd:element name="SndRcvRequest" type="<!-- prefix -->:format"/>
11     <xsd:complexType name="format">
12           <xsd:sequence>
13             <xsd:element name="mqheader" type="<!-- prefix -->:mqheader"/>
14             <xsd:element name="userdata" type="<!-- prefix -->:userdata"/>
15           </xsd:sequence>
16     </xsd:complexType>
17
18     <xsd:complexType name="mqheader">
19           <xsd:sequence>
20             <xsd:element name="sndresname" type="xsd:string" minOccurs="0"/>
21             <xsd:element name="rcvresname" type="xsd:string" minOccurs="0"/>
22           </xsd:sequence>
23     </xsd:complexType>
24
25     <xsd:complexType name="userdata">
26           <!-- Define XML schema of the user data. -->
27     </xsd:complexType>
28 </xsd:schema>
```

（凡例）

：ユーザの編集個所

表 3-63　メッセージ送受信要求電文用テンプレートファイルの内容

行数	説明
4 行目	このテンプレートファイルの名前空間宣言を表します。ユーザは 6 行目に指定した targetNamespace 属性の値に任意の接頭辞を宣言します。
6 行目	このテンプレートファイルの対象名前空間を表します。ユーザは targetNamespace 属性の値に一意となる対象名前空間を指定します。
8 行目	26 行目でユーザデータ用の XML スキーマ情報をインポートする場合，5 行目にインポートするスキーマのターゲットネームスペースを，8 行目にインポートするスキーマのインポート文を記述します。
10～16 行目	SndRcvRequest 要素宣言を表します。要求電文のルート要素となる mqheader 要素と userdata 要素を持ちます。ユーザは mqheader 要素と userdata 要素の接頭辞として 4 行目で宣言した接頭辞を記述します。
18～23 行目	mqheader 要素宣言を表します。

行数	説明
21 行目	受信キュー特定情報の指定要素を表します。この指定個所がメッセージ送信要求用テンプレートファイルと異なります。
25 行目	userdata 要素宣言を表します。
26 行目	ユーザデータ用の XML スキーマ情報を記述します。 記述方法 記述方法は 2 種類あり，このファイルに直接記述する方法と，ユーザデータ用の XML スキーマ情報を記述するファイルを別に作成し，このファイルからインポートする方法があります。インポートする場合は 5 行目にインポートするスキーマのターゲットネームスペースを，8 行目にインポートするスキーマのインポート文を記述します。 注意事項 ユーザデータ用の XML スキーマ情報を記述する際，userdata 要素の子要素には複数の要素を宣言してはいけません。userdata 要素の子要素は XML ファイル出力時にルート要素となる要素であるため，userdata 要素の子要素は必ず 1 つの要素しか宣言できません。

XML 形式の応答電文

応答電文用の XML フォーマット定義ファイルは，製品提供のフォーマット定義ファイルを使用するか，ユーザが独自に作成してください。

製品提供ファイルを使用する場合

次の条件に当てはまるときに使用します。

- メッセージ送信応答するとき。
- メッセージ受信応答，ブラウズ・メッセージ受信応答またはメッセージ送受信応答で Fault 応答情報電文を受信するとき。

ユーザが独自に作成する場合

上記，製品提供ファイルを使用する場合に該当しない場合は，ユーザが独自に作成してください。

バイナリ形式の要求電文

要求電文用のバイナリフォーマット定義ファイルは，製品提供のテンプレートファイルを使用してユーザがバイナリフォーマット定義画面で編集するか，ユーザが独自に作成してください。

バイナリフォーマット定義テンプレートファイルを使用する場合

- メッセージの送信要求電文で送信キュー特定情報を指定するとき。
- メッセージ送受信要求で送信キュー特定情報および受信キュー特定情報を指定するとき。

ユーザが独自に作成する場合

上記，製品提供ファイルを使用する場合に該当しない場合は，ユーザが独自に作成してください。

バイナリ形式の応答電文

応答電文用バイナリフォーマット定義ファイルは次のケースがありますが，ユーザが独自に作成してください。

- メッセージ受信応答でバイナリメッセージを受信する。

- ブラウズ・メッセージ受信応答でバイナリメッセージを受信する。
- メッセージ送受信応答でバイナリメッセージを受信する。

(2)　データ変換の定義

データ変換（マッピング）定義画面で，変換元の電文フォーマット定義ファイルおよび変換先の電文フォーマット定義ファイルを設定して，データ変換を定義します。

データ変換の定義方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.　データ変換の定義」を参照してください。

(3)　サービスアダプタ定義画面での操作

Message Queue アダプタを定義する場合の手順を次に示します。設定内容の詳細は，「3.3.19(9) Message Queue アダプタの場合」を参照してください。

1. サービスアダプタ定義画面を表示します。

   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービス部品制御情報の［サービス名］および［サービス ID］を必要に応じて編集します。

3. サービス部品制御情報の［追加］ボタンをクリックして，定義パターン，オペレーション名，およびフォルト名※を指定します。

   オペレーション名は Message Queue アダプタ通信構成定義のオペレーション名（op_name）と一致させる必要があります。

   注※　定義パターンが「メッセージ送信」以外の場合に指定します。

4. オペレーション情報の［通信モデル］のドロップダウンリストから，「同期」を選択します。

5. 要求電文について，手順 6.から手順 15.までを実施します。

6. 次に示す操作をします。

   標準電文のフォーマットを指定する場合

   手順 7.～9.を実施して，手順 10.に進みます。

   標準電文のフォーマットを指定しない場合

   手順 10.に進みます。

7. 標準電文の［使う］チェックボックスをチェックします。

8. 標準電文の［参照］ボタンをクリックして，［電文フォーマット］に標準電文のフォーマットを指定します。

   指定できる電文フォーマットの形式については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.5　XML スキーマの適用範囲」を参照してください。

9. 標準電文の［表示］ボタンをクリックします。

電文フォーマットが XML 形式の場合，標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。

10. 手順 3.で指定した定義パターンに応じて次の電文フォーマットが設定されていることを確認します。

定義パターンが「メッセージ送信」の場合

要求電文のサービス部品：なし

応答電文のサービス部品：adpmq_snd_response.xsd

フォルト電文：なし

定義パターンが「メッセージ受信」の場合

要求電文のサービス部品：adpmq_rcv_request.xsd

応答電文のサービス部品：なし

フォルト電文：adpmq_fault_response.xsd

定義パターンが「ブラウズ・メッセージ受信」の場合

要求電文のサービス部品：adpmq_browse_request.xsd

応答電文のサービス部品：なし

フォルト電文：adpmq_fault_response.xsd

定義パターンが「メッセージ送受信」の場合

要求電文のサービス部品：なし

応答電文のサービス部品：なし

フォルト電文：adpmq_fault_response.xsd

11. サービス部品電文の［表示］ボタンをクリックします。

サービス部品電文のフォーマットが表示されます。必要に応じてサービス部品電文のフォーマットを確認してください。

12. 次に示す操作をします。

標準電文の［使う］チェックボックスをチェックした場合

手順 13.～15.を実施して，手順 16.に進みます。

標準電文の［使う］チェックボックスをチェックしない場合

手順 16.に進みます。

13. データ変換定義のファイル名を入力します。

14. ［編集］ボタンをクリックします。

データ変換定義画面が表示されます。初めて定義する場合は，［ルート要素選択］ダイアログが表示されます。

15. 標準電文とサービス電文の内容をマッピングします。

16. 応答電文についても，手順 6.から手順 15.までを実施します。

17. 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画面（詳細）が表示されます。

18. サービスアダプタ（EJB-JAR ファイル）およびユーティリティクラス（JAR ファイル）の名称を確認します。
    サービスアダプタ（EJB-JAR ファイル）の名称が「cscmsg_adpejb.jar」，ユーティリティクラス（JAR ファイル）の名称が「adpmqpc.jar」であることを確認してください。

19. 独自定義ファイルに次のファイルが追加されていることを確認します。
    - customadapter_properties.xml（Message Queue アダプタ環境定義ファイル）
    - adpmq_communication（Message Queue アダプタ通信構成定義ファイル）
    - cscadapter_property.xml（アプリケーション統合属性ファイル）

(4) Message Queue アダプタ環境定義ファイルの編集

Message Queue アダプタの環境定義に必要な情報は，XML 形式の Message Queue アダプタ環境定義ファイルに定義します。

Message Queue アダプタ環境定義ファイルの編集手順を示します。

1. サービスアダプタ定義画面（詳細）の［独自定義ファイル］で「customadapter_properties.xml」を選択し，［編集］ボタンをクリックします。
   Message Queue アダプタ環境定義ファイルを編集するためのエディタが起動します。

2. エディタ上で Message Queue アダプタ環境定義ファイルを編集します。
   Message Queue アダプタ環境定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.8.1　Message Queue アダプタ環境定義ファイル」を参照してください。

3. Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(5) Message Queue アダプタ通信構成定義ファイルの編集

Message Queue アダプタの通信構成定義に必要な情報は，XML 形式の Message Queue アダプタ通信構成定義ファイルに定義します。

Message Queue アダプタ通信構成定義ファイルの編集手順を示します。

1. サービスアダプタ定義画面（詳細）の［独自定義ファイル］で「adpmq_communication.xml」を選択し，［編集］ボタンをクリックします。
   Message Queue アダプタ通信構成定義ファイルを編集するためのエディタが起動します。

2. エディタ上で Message Queue アダプタ通信構成定義ファイルを編集します。

Message Queue アダプタ通信構成定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.8.2　Message Queue アダプタ通信構成定義ファイル」を参照してください。

3. Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(6)　アプリケーション統合属性ファイルの編集

アプリケーション統合属性ファイルの編集方法について説明します。

(a)　編集手順

アプリケーション統合属性ファイルの編集手順を示します。

1. サービスアダプタ定義画面（詳細）の［独自定義ファイル］で「cscadapter_property.xml」を選択し，［編集］ボタンをクリックします。

   アプリケーション統合属性ファイルを編集するためのエディタが起動します。

2. エディタ上でアプリケーション統合属性ファイルを編集します。

   設定内容については「(b) 定義詳細」を参照してください。

3. Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(b)　定義詳細

アプリケーション統合属性ファイルの内容を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hitachi-application-all-property PUBLIC
'-//Hitachi, Ltd.//DTD Application All Property 7.6//EN' 'http://localhost/hitachi-applicati
on-all-property_7_6.dtd'>
<!-- All Rights Reserved. Copyright (C) 2008, Hitachi, Ltd. -->

<hitachi-application-all-property>
  <hitachi-application-property>
    <lookup-name>CSCMQ</lookup-name>
    <security-prop>
      <security-method>no_security_for_methods_without_roles</security-method>
    </security-prop>
    <managed-by-ctm>false</managed-by-ctm>
  </hitachi-application-property>
  <ejb-jar>
    <hitachi-ejb-jar-property>
      <display-name>CSCMsgServiceAdapter</display-name>
    </hitachi-ejb-jar-property>
    <hitachi-session-bean-property>
      <display-name>CSCMsgServiceAdapterEJB</display-name>
      <session-type>Stateless</session-type>
      <transaction-type>Container</transaction-type>
      <resource-ref>
        <res-ref-name>jms/リソースアダプタ特定情報</res-ref-name>
        <res-type>javax.jms.QueueConnectionFactory</res-type>
        <res-auth>Container</res-auth>
```

```
        <res-sharing-scope>Unshareable</res-sharing-scope>
        <linked-to>リソースアダプタ表示名</linked-to>
      </resource-ref>
      <resource-env-ref>
        <resource-env-ref-name>送信キュー特定情報または受信キュー特定情報</resource-env-ref-
name>
        <resource-env-ref-type>javax.jms.Queue</resource-env-ref-type>
        <linked-adminobject>
          <resourceadapter-name>リソースアダプタ表示名</resourceadapter-name>
          <adminobject-name>管理対象オブジェクト名</adminobject-name>
        </linked-adminobject>
      </resource-env-ref>
      <container-transaction>
        <description></description>
        <method>
          <description></description>
          <method-intf></method-intf>
          <method-name>*</method-name>
        </method>
        <trans-attribute>Required</trans-attribute>
      </container-transaction>
      <session-runtime>
        <lookup-name>CSCMsgServiceAdapterEJB</lookup-name>
        <optional-name></optional-name>
        <maximum-sessions>0</maximum-sessions>
        <stateless>
          <pooled-instance>
            <minimum>インスタンス最小数</minimum>
            <maximum>インスタンス最大数</maximum>
          </pooled-instance>
        </stateless>
      </session-runtime>
   </hitachi-session-bean-property>
  </ejb-jar>
</hitachi-application-all-property>
```

定義の設定値について説明します。

●アプリケーション属性<hitachi-application-property>

<lookup-name>

クライアントから EJB をルックアップするときに使う名前として「CSCMQ」を指定します。

<security-method>

セキュリティの管理方法として「no_security_for_methods_without_roles」を指定します。

<managed-by-ctm>

CTM の連帯として「false（CTM と連携しない)」を指定します。

●EJB-JAR 属性<hitachi-ejb-jar-property>

<display-name>

EJB-JAR の表示名として「CSCMsgServiceAdapter」を指定します。

●Session Bean 属性<hitachi-session-bean-property>

<display-name>

Session Bean の表示名として「CSCMsgServiceAdapterEJB」を指定します。

<session-type>

Session Bean の種別として「Stateless」を指定します。

<transaction-type>

トランザクション管理種別として「Container」を指定します。

<res-ref-name>

リソースアダプタ特定情報を指定します。この指定値はリソースアダプタを特定する任意の名称を指定します。指定するときは先頭に「jms/」を付与してください。

この要素と同一の指定値を Message Queue アダプタ通信構成定義の<res_ref_name>要素に設定してください。

<res-type>

リソース種別を指定します。参照先リソースアダプタのコネクション定義識別子（<connectionfactory-interface>タグの設定値）として「javax.QueueConnectionFactory」を指定します。

<res-auth>

認証方式として「Container」を指定します。

<res-sharing-scope>

参照したリソースアダプタ共有有無の指定として「Unshareable」を指定します。

<linked-to>

参照するリソースアダプタの表示名を指定します。

Outbound リソースアダプタは，<リソースアダプタ表示名>!<コネクション定義識別子>で指定します。

<コネクション定義識別子>とは，リソースアダプタのデプロイ・ディスクリプター（ra.xml）の<connectionfactory-interface>タグで指定した値のことです。

<res-env-ref-name>

送信キュー特定情報または受信キュー特定情報を指定します。この指定値は接続先キューを特定する任意の名称を指定します。指定するときは先頭に「jms/」を付与してください。

この要素と同一の指定値を Message Queue アダプタ通信構成定義の<res_name>，<snd_res_name>，または<rcv_res_name>要素に設定してください。

<res-env-ref-type>

リソース環境変数の値のクラスタイプの指定として「javax.jms.Queue」を指定します。

<resourceadapter-name>

リソース環境変数のリソースアダプタの表示名を指定します。

<adminobject-name>
管理対象オブジェクト名を指定します。リソースアダプタのデプロイ・ディスクリプター（ra.xml）の<adminobject-name>と同一の値を指定します。

<trans-attribute>
メソッドに割り当てるトランザクション属性として「Required」を指定します。

<lookup-name>
クライアントから EJB をルックアップする場所に使用する名称として「CSCMsgServiceAdapterEJB」を指定します。

<maximum-sessions>
セッションの最大数として「0（無制限）」を指定します。

<minimum>
プール内のインスタンスの最小数を指定します。

<maximum>
プール内のインスタンスの最大数を指定します。

アプリケーション統合属性ファイルの詳細については，次の個所を参照してください。

- マニュアル「アプリケーションサーバ アプリケーション設定操作ガイド」の「9.2　アプリケーション統合属性ファイルによるプロパティ設定」
- マニュアル「アプリケーションサーバ リファレンス 定義編(アプリケーション／リソース定義)」の「3.1　アプリケーション統合属性ファイル」

(7)　Message Queue アダプタ実行環境プロパティファイルの作成

Message Queue アダプタの場合，Message Queue アダプタ実行環境プロパティファイルを作成する必要があります。

Message Queue アダプタ実行環境プロパティファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.12.1　Message Queue アダプタ実行環境プロパティファイル」を参照してください。

3.3.10　FTP アダプタを定義する

FTP アダプタの定義方法について説明します。

(1)　電文フォーマットの作成

FTP アダプタの電文フォーマットの定義ファイルはサービスプラットフォームが提供するスキーマを使用します。そのため，電文フォーマットの定義ファイルの作成は不要です。

FTP アダプタで使用する電文フォーマットの内容について説明します。

ファイルの格納場所は「<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¥custom-adapter¥FTP¥schema」です。

(a) FTP アダプタの要求電文フォーマット

FTP アダプタの要求電文フォーマットを，オペレーションごとに説明します。

●PUT オペレーション

ビジネスプロセスから FTP アダプタに渡す PUT オペレーションの要求電文フォーマットを次に示します。ファイル名は「ftpadp_put_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/put_request」です。

表 3-64 要求電文フォーマット（PUT オペレーション）

タグ名		出現回数※1	説明
<request>		1 回	−
	<host-ipaddr>	0 または 1 回	接続する FTP サーバの IP アドレスまたはホスト名です。 • タグがあって値がある場合 指定の文字列で connect が発行されます。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。 Windows を使用する場合，ループバック IP アドレス（127.0.0.1 または localhost）を指定すると，FTP 接続に失敗する場合があります。実 IP アドレスまたはホスト名を指定してください。
	<host-con-port>	0 または 1 回	FTP サーバの制御コネクション用ポート番号です。 • タグがあって値がある場合 1 から 65535 までのポート番号を指定します。これ以外の値を指定した場合，システム例外を返します。 • タグがあって値がない場合 デフォルトの 21 が指定されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<file-name-charset>	0 または 1 回	FTP サーバとの制御コネクションで，情報（ファイル名など）を送受信するときに使用する文字セットの名称を示すタグです。指定値は次のどちらかです。 UTF-8：文字セットに「UTF-8」を使用します。 MS932：文字セットに「MS932」を使用します。 • タグがあって値が使用できる文字セット名の場合 情報（ファイル名など）を送受信するときの文字セットとして適用されます。

タグ名			出現回数※1	説明
	<file-name-charset>		0 または 1 回	• タグがあって値が使用できない文字セット名の場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「UTF-8」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<ftps>		0 または 1 回	FTPS に関する設定を示すタグです。タグがない場合，FTPS に関するすべての設定は FTP アダプタ実行環境プロパティファイルの値が有効となります。
		<ftps-enable>	0 または 1 回	FTP サーバとの接続に FTPS を使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：FTP サーバとの接続に FTPS を使用します。 false：FTP サーバとの接続に通常の FTP を使用します。※2 FTPS を使用する場合，制御コネクションおよびデータコネクションの通信は暗号化されます。ただし，接続先の FTP サーバが FTPS による接続をサポートしている必要があります。「true」を指定した場合でも，データコネクションの通信の暗号化が設定されていないときは，データコネクションの通信も暗号化されません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
		<ftps-protocol-name>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，使用するセキュリティ通信用のプロトコルを示すタグです。指定値は次のどちらかです。 TLS：セキュリティ通信用のプロトコルに TLS を使用します。 SSL：セキュリティ通信用のプロトコルに SSL を使用します。 • タグがあって値が「TLS」または「SSL」の場合 指定した値が適用されます。 • タグがあって値が「TLS」または「SSL」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「TLS」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-implicit-mode>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，FTPS の Implicit モードを使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：Implicit モードを使用します。

タグ名			出現回数※1	説明
		<ftps-implicit-mode>	0 または 1 回	false：Explicit モードを使用します。※2 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-data-con-secure>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，データコネクションの通信を暗号化するかどうかを指定するタグです。指定値は次のどちらかです。 true：データコネクションの通信を暗号化します。 false：データコネクションの通信を暗号化しません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「true」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-server-authentication>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，サーバ認証をするかどうかを指定するタグです。指定値は次のどちらかです。 true：サーバ認証をします。 false：サーバ認証をしません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。なお，「true」を指定する場合，キーストアに関する設定が必要です。キーストアの設定方法については，「付録 H.1　FTPS を使用したセキュアな接続(FTP アダプタ)」を参照してください。

タグ名		出現回数※1	説明
	<ftp-user>※3	0 または 1 回	接続する FTP サーバのログインユーザ名を示すタグです。USER コマンドの引数として使用します。このタグに指定するログインユーザには，ファイル操作を実行できる権限が与えられている必要があります。 FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 • タグがあって正常な値がある場合 「USER　"値"」が発行されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値がない場合，または値に不正な文字が含まれている場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。
	<ftp-acct>※3	0 または 1 回	接続する FTP サーバの課金情報を示すタグです。 FTP アダプタアカウント定義ファイルに重複して設定されている場合，要求電文の値が有効になります。 課金情報を必要とするユーザ 条件に応じて次の値が利用されます。 • タグがあって正常な値がある場合 タグの内容が ACCT コマンドに利用されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値に不正な文字が含まれている場合 システム例外を返します。 • タグがあって値がない場合 「ACCT　""」が発行されます。 • タグがない場合 FTP アダプタアカウント定義ファイルで指定されている ACCT の値が利用されます。FTP アダプタアカウント定義ファイルに ACCT がない場合は「ACCT　""」が発行されます。 課金情報を必要としないユーザ 指定は無視されます。
	<ftp-type>	0 または 1 回	転送するデータタイプ（伝送モード）を示すタグです。TYPE コマンドの引数として使用します。FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 指定値は次のどちらかです。 A：ASCII 形式で転送します。 I：バイナリ形式で転送します。 • タグがあって値が「A」または「I」の場合 TYPE コマンドの引数として使用されます。 • タグがあって値が「A」または「I」のどちらでもない場合 システム例外を返します。

タグ名		出現回数※1	説明
	<ftp-type>	0 または 1 回	• タグがあって値がない場合 TYPE コマンドは実行されません。※4 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，TYPE コマンドは実行されません。※4
	<ftp-mode>	0 または 1 回	転送モード（圧縮モード）を示すタグです。MODE コマンドの引数として使用します。FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 指定値は次のどちらかです。 S：ファイルを圧縮しないで伝送します。 C：ファイルを圧縮して伝送します。 • タグがあって値が「S」または「C」の場合 MODE コマンドの引数として使用されます。 • タグがあって値が「S」または「C」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 MODE コマンドは実行されません。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効となります。FTP アダプタ実行環境プロパティファイルにも指定がない場合は，デフォルトの「S」で MODE コマンドが実行されます。
	<pasvport>	0 または 1 回	データコネクションの接続モードを示すタグです。 指定値は次のどちらかです。 PASV：PASV モードでデータコネクションを設定します。 PORT：PORT モードでデータコネクションを設定します。 • タグがあって値が「PASV」または「PORT」の場合 指定した値が適用されます。 • タグがあって値が「PASV」または「PORT」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「PASV」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルにも指定がない場合は，デフォルトの「PASV」でデータコネクションを設定します。 要求電文からデータコネクションの接続モードを設定する場合，FTP アダプタ実行環境プロパティファイルで設定した値に対して，次のプロパティの値のチェック処理が実施されます。 • PASV ftpadp.client-pasv-port.min ftpadp.client-pasv-port.max

タグ名		出現回数※1	説明
	<pasvport>	0 または 1 回	• PORT ftpadp.listen-port.min ftpadp.listen-port.max チェック処理の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.8.3　FTP アダプタ実行環境プロパティファイル」を参照してください。
	<file-size-check>	0 または 1 回	送信側と受信側で，転送されたファイルのサイズが正しいかどうか確認する機能を，実行するかどうかを指定するタグです。 指定値は次のどちらかです。 true：ファイルサイズを確認します。 false：ファイルサイズを確認しません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<transfer-type>	1 回	転送種別を示すタグです。次のどちらかを設定します。 STOR：ファイルの上書き APPE：ファイルへの情報追加 • タグがあって値が「STOR」または「APPE」の場合 転送種別として利用されます。 • タグがない，またはタグがあって値が「STOR」または「APPE」のどちらでもない場合 システム例外を返します。
	<request-id>	0 または 1 回	受付で作成したリクエスト ID を示すタグです。 • タグがあって値がある場合 リクエスト ID として利用されます。 • タグがない，またはタグがあって値がない場合 作業フォルダを利用するときはシステム例外を返します。共通フォルダを利用するときは無視されます。
	<local-folder>	1 回	ローカル側フォルダを示すタグです。 • タグの属性に common="true"が指定されている場合 ローカルフォルダとして共通フォルダを利用し，このタグには「共通フォルダ定義名」を指定します。 指定された共通フォルダ定義名から共通フォルダが取得できないときは，システム例外を返します。 • タグの属性に common="false"が指定されている場合 ローカルフォルダとして作業フォルダを利用します。

タグ名		出現回数※1	説明
	<local-folder>	1 回	• common 属性がない，または値が"true"および"false"のどちらでもない場合 システム例外を返します。
	<local-file-name>	1 回	ローカル側ファイル名を示すタグです。 • 作業フォルダを利用する場合，作業フォルダ直下のファイルを指定してください。ファイル名の先頭以外に区切り文字（スラッシュ（/），円マーク（¥）など）は使用できません。ファイル名の先頭にスラッシュ（/）を使用した場合，スラッシュ（/）は無視されます。 • 共通フォルダを利用する場合，共通フォルダルート配下のファイルを指定してください。シンボリックリンク以外のファイルを指定してください。 • タグがあって値が上記の制限に従っている場合 ファイル名として使用されます。 • タグがあって値が上記の制限に違反する場合 システム例外を返します。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
	<remote-path>※5	1 回	リモートパス名（FTP サーバに転送するときの転送先パス名）を示すタグです。 • タグがあって値がある場合 リモートパス名として利用されます。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
	<ftp-commands-before>	0 または 1 回	ファイル転送前に実行する FTP コマンドとその引数を示すタグです。※6 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します（例：MKD transdir;CWD transdir）。 • タグがあって値がある場合 ファイル転送前に実行するコマンドとして利用されます。 • タグがない，またはタグがあって値がない場合 ファイル転送前に実行するコマンドはありません。
	<ftp-commands-after>	0 または 1 回	ファイル転送後に実行する FTP コマンドとその引数を示すタグです。※6 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します（例：RNFR oldfile.txt;RNTO newfile.txt）。 • タグがあって値がある場合 ファイル転送後に実行するコマンドとして利用されます。 • タグがない，またはタグがあって値がない場合 ファイル転送後に実行するコマンドはありません。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

接続先 FTP サーバが FTPS の Implicit モードで動作している場合に<ftps-enable>タグ，または<ftps-implicit-mode>タグの値に false を指定すると，FTP サーバに接続したあとタイムアウトが発生するまで待ち状態となることがあります。

タイムアウトが発生するまでの時間は，次の値のうちどちらか小さい方です。

・FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値

・接続先 FTP サーバ側で設定されたタイムアウト値

FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値でタイムアウトとなった場合，FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.retry.count プロパティで指定した回数のリトライを行ったあと，KDEK30407-E のエラーが発生します。

接続先 FTP サーバ側で設定されたタイムアウト値でタイムアウトとなった場合，KDEK30428-E のエラーが発生します。

注※3

使用できる文字を次に示します。

半角英数字，半角空白，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

注※4

FTP プロトコルの仕様（RFC959）に従い，デフォルトでは ASCII 形式で転送されます。

注※5

FTP 受付の要求電文の<transfer-path>タグと同じ値を FTP アダプタの<remote-path>タグに設定すると，パスの先頭がスラッシュ（/）になります。

パスの先頭にスラッシュ（/）があると問題がある場合は，ビジネスプロセスのデータ変換アクティビティで部分文字列取得ファンクション（substr）を使用して FTP 受付の要求電文の<transfer-path>タグの値から 2 文字目以降の文字列を取り出して，FTP アダプタの<remote-path>タグに設定してください。パスの先頭にスラッシュ（/）があると問題がある場合とは，相対パスを指定する場合や Windows の絶対パス形式でパスを指定したい場合などです。

注※6

<ftp-commands-before>タグおよび<ftp-commands-after>タグに指定されたコマンドは，そのまま FTP サーバに送られ，FTP アダプタはコマンドのオプションの妥当性やコマンドの実行による影響について関知しません。

そのため，<ftp-commands-before>タグおよび<ftp-commands-after>タグに OPTS UTF {ON|OFF}コマンドを指定した場合は，FTP サーバとの通信に適用される文字セットを次のどちらかで指定しておく必要があります。

・要求電文の<file-name-charset>タグ

・FTP アダプタ実行環境プロパティファイルの ftpadp.file-name.charset プロパティ

●MPUT オペレーション

ビジネスプロセスから FTP アダプタに渡す MPUT オペレーションの要求電文フォーマットを次に示します。ファイル名は「ftpadp_mput_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/mput_request」です。

表 3-65　要求電文フォーマット（MPUT オペレーション）

タグ名	出現回数※1	説明
<request>	1 回	−

タグ名		出現回数※1	説明
<host-ipaddr>		0 または 1 回	接続する FTP サーバの IP アドレスまたはホスト名です。 • タグがあって値がある場合 指定の文字列で connect が発行されます。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。 Windows を使用する場合，ループバック IP アドレス（127.0.0.1 または localhost）を指定すると，FTP 接続に失敗する場合があります。実 IP アドレスまたはホスト名を指定してください。
<host-con-port>		0 または 1 回	FTP サーバの制御コネクション用ポート番号です。 • タグがあって値がある場合 1 から 65535 までのポート番号を指定します。これ以外の値を指定した場合，システム例外を返します。 • タグがあって値がない場合 デフォルトの 21 が指定されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
<file-name-charset>		0 または 1 回	FTP サーバとの制御コネクションで，情報（ファイル名など）を送受信するときに使用する文字セットの名称を示すタグです。指定値は次のどちらかです。 UTF-8：文字セットに「UTF-8」を使用します。 MS932：文字セットに「MS932」を使用します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が使用できる文字セット名の場合 情報（ファイル名など）を送受信するときの文字セットとして適用されます。 • タグがあって値が使用できない文字セット名の場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「UTF-8」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
<ftps>		0 または 1 回	FTPS に関する設定を示すタグです。タグがない場合，FTPS に関するすべての設定は FTP アダプタ実行環境プロパティファイルの値が有効となります。
	<ftps-enable>	0 または 1 回	FTP サーバとの接続に FTPS を使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：FTP サーバとの接続に FTPS を使用します。 false：FTP サーバとの接続に通常の FTP を使用します。※2

タグ名			出現回数※1	説明
		<ftps-enable>	0 または 1 回	FTPS を使用する場合，制御コネクションおよびデータコネクションの通信は暗号化されます。ただし，接続先の FTP サーバが FTPS による接続をサポートしている必要があります。 データコネクションの通信を暗号化するには，<ftps-data-con-secure>を true に設定する必要があります。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
		<ftps-protocol-name>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，使用するセキュリティ通信用のプロトコルを示すタグです。指定値は次のどちらかです。 TLS：セキュリティ通信用のプロトコルに TLS を使用します。 SSL：セキュリティ通信用のプロトコルに SSL を使用します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「TLS」または「SSL」の場合 指定した値が適用されます。 • タグがあって値が「TLS」または「SSL」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「TLS」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-implicit-mode>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，FTPS の Implicit モードを使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：Implicit モードを使用します。 false：Explicit モードを使用します。※2 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。

タグ名			出現回数※1	説明
		<ftps-implicit-mode>	0 または 1 回	この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-data-con-secure>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，データコネクションの通信を暗号化するかどうかを指定するタグです。指定値は次のどちらかです。 true：データコネクションの通信を暗号化します。 false：データコネクションの通信を暗号化しません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「true」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-server-authentication>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，サーバ認証をするかどうかを指定するタグです。指定値は次のどちらかです。 true：サーバ認証をします。 false：サーバ認証をしません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。なお，「true」を指定する場合，キーストアに関する設定が必要です。キーストアの設定方法については，「付録 H.1　FTPS を使用したセキュアな接続(FTP アダプタ)」を参照してください。
	<ftp-user>※3		0 または 1 回	接続する FTP サーバのログインユーザ名を示すタグです。USER コマンドの引数として使用します。このタグに指定するログインユーザには，ファイル操作を実行できる権限が与えられている必要があります。 FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 また，タグの有無に応じて次のように動作します。 • タグがあって正常な値がある場合 「USER　"値"」が発行されます。値に改行が含まれている場合，システム例外を返します。

タグ名		出現回数※1	説明
	<ftp-user>※3	0 または 1 回	• タグがあって値がない場合，または値に不正な文字が含まれている場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。
	<ftp-acct>※3	0 または 1 回	接続する FTP サーバの課金情報を示すタグです。 FTP アダプタアカウント定義ファイルに重複して設定されている場合，要求電文の値が有効になります。 課金情報を必要とするユーザ 条件に応じて次の値が利用されます。 • タグがあって正常な値がある場合 タグの内容が ACCT コマンドに利用されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値に不正な文字が含まれている場合 システム例外を返します。 • タグがあって値がない場合 「ACCT　""」が発行されます。 • タグがない場合 FTP アダプタアカウント定義ファイルで指定されている ACCT の値が利用されます。FTP アダプタアカウント定義ファイルに ACCT がない場合は「ACCT　""」が発行されます。 課金情報を必要としないユーザ 指定は無視されます。
	<ftp-type>	0 または 1 回	転送するデータタイプ（伝送モード）を示すタグです。TYPE コマンドの引数として使用します。FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 指定値は次のどちらかです。 A：ASCII 形式で転送します。 I：バイナリ形式で転送します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「A」または「I」の場合 TYPE コマンドの引数として使用されます。 • タグがあって値が「A」または「I」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 TYPE コマンドは実行されません。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの ftpadp.ftp-type プロパティの値で TYPE コマンドが実行されます。このプロパティの指定がない場合，TYPE コマンドは実行されません。

タグ名		出現回数※1	説明
	<ftp-mode>	0 または 1 回	転送モード（圧縮モード）を示すタグです。MODE コマンドの引数として使用します。FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 指定値は次のどちらかです。 S：ファイルを圧縮しないで伝送します。 C：ファイルを圧縮して伝送します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「S」または「C」の場合 MODE コマンドの引数として使用されます。 • タグがあって値が「S」または「C」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 MODE コマンドは実行されません。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの ftpadp.ftp-mode プロパティに指定された値で MODE コマンドが実行されます。このプロパティの指定がない場合，「MODE S」が実行されます。
	<pasvport>	0 または 1 回	データコネクションの接続モードを示すタグです。 指定値は次のどちらかです。 PASV：PASV モードでデータコネクションを設定します。 PORT：PORT モードでデータコネクションを設定します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「PASV」または「PORT」の場合 指定した値が適用されます。 • タグがあって値が「PASV」または「PORT」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「PASV」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルにも指定がない場合は，デフォルトの「PASV」でデータコネクションを設定します。
	<file-size-check>	0 または 1 回	送信側と受信側で，転送されたファイルのサイズが正しいかどうか確認する機能を，実行するかどうかを指定するタグです。 指定値は次のどちらかです。 true：ファイルサイズを確認します。 false：ファイルサイズを確認しません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。

タグ名			出現回数※1	説明
	<file-size-check>		0 または 1 回	• タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルにも指定がない場合は，デフォルトの「false」が使用されます。
	<transfer-type>		1 回	転送種別を示すタグです。次のどちらかを設定します。 STOR：ファイルの上書き APPE：ファイルへの情報追加 また，タグの有無に応じて次のように動作します。 • タグがあって値が「STOR」または「APPE」の場合 転送種別として利用されます。 • タグがない，またはタグがあって値が「STOR」または「APPE」のどちらでもない場合 システム例外を返します。
	<request-id>		0 または 1 回	受付で作成したリクエスト ID を示すタグです。 • タグがあって値がある場合 リクエスト ID として利用されます。 • タグがない，またはタグがあって値がない場合 作業フォルダを利用するときはシステム例外を返します。共通フォルダを利用するときは無視されます。
	<transfer-unit>		1 回以上	転送する個々のファイルに関する情報を指定します。
		<local-folder>	1 回	ローカル側フォルダを示すタグです。 • タグの属性に common="true"が指定されている場合 ローカルフォルダとして共通フォルダを利用し，このタグには「共通フォルダ定義名」を指定します。 指定された共通フォルダ定義名から共通フォルダが取得できないときは，システム例外を返します。 • タグの属性に common="false"が指定されている場合 ローカルフォルダとして作業フォルダを利用します。 • common 属性がない，または値が「true」および「false」のどちらでもない場合 システム例外を返します。
		<local-file-name>	0 または 1 回	ローカル側ファイル名を示すタグです。 • 作業フォルダを利用する場合，作業フォルダ直下のファイルを指定してください。ファイル名の先頭以外に区切り文字（スラッシュ（/），円マーク（¥）など）は使用できません。ファイル名の先頭にスラッシュ（/）を使用した場合，スラッシュ（/）は無視されます。 • 共通フォルダを利用する場合，共通フォルダルート配下のファイルを指定してください。シンボリックリンク以外のファイルを指定してください。 また，タグの有無に応じて次のように動作します。

タグ名			出現回数※1	説明
		<local-file-name>	0 または 1 回	• タグがあって値が上記の制限に従っている場合 ファイル名として使用されます。 • タグがあって値が上記の制限に違反する場合 システム例外を返します。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
		<remote-path>※4	1 回	リモートパス名（FTP サーバに転送するときの転送先パス名）を示すタグです。 • タグがあって値がある場合 リモートパス名として利用されます。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
	<ftp-commands-before>		0 または 1 回	ファイル転送前に実行する FTP コマンドとその引数を示すタグです。※5 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します（例：MKD transdir;CWD transdir）。 • タグがあって値がある場合 ファイル転送前に実行するコマンドとして利用されます。 • タグがない，またはタグがあって値がない場合 ファイル転送前に実行するコマンドはありません。
	<ftp-commands-after>		0 または 1 回	ファイル転送後に実行する FTP コマンドとその引数を示すタグです。※5 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します（例：RNFR oldfile.txt;RNTO newfile.txt）。 • タグがあって値がある場合 ファイル転送後に実行するコマンドとして利用されます。 • タグがない，またはタグがあって値がない場合 ファイル転送後に実行するコマンドはありません。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

接続先 FTP サーバが FTPS の Implicit モードで動作している場合に<ftps-enable>タグ，または<ftps-implicit-mode>タグの値に false を指定すると，FTP サーバに接続したあとタイムアウトが発生するまで待ち状態となることがあります。

タイムアウトが発生するまでの時間は，次の値のうちどちらか小さい方です。

・FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値

・接続先 FTP サーバ側で設定されたタイムアウト値

FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値でタイムアウトとなった場合，FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.retry.count プロパティで指定した回数のリトライを行ったあと，KDEK30407-E のエラーが発生します。

接続先 FTP サーバ側で設定されたタイムアウト値でタイムアウトとなった場合，KDEK30428-E のエラーが発生します。

注※3

使用できる文字を次に示します。

半角英数字，半角空白，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

注※4

FTP 受付の要求電文の<transfer-path>タグと同じ値を FTP アダプタの<remote-path>タグに設定すると，パスの先頭がスラッシュ（/）になります。

パスの先頭にスラッシュ（/）があると問題がある場合は，ビジネスプロセスのデータ変換アクティビティで部分文字列取得ファンクション（substr）を使用して FTP 受付の要求電文の<transfer-path>タグの値から 2 文字目以降の文字列を取り出して，FTP アダプタの<remote-path>タグに設定してください。パスの先頭にスラッシュ（/）があると問題がある場合とは，相対パスを指定する場合や Windows の絶対パス形式でパスを指定したい場合などです。

注※5

<ftp-commands-before>タグおよび<ftp-commands-after>タグに指定されたコマンドは，そのまま FTP サーバに送られ，FTP アダプタはコマンドのオプションの妥当性やコマンドの実行による影響について関知しません。

そのため，<ftp-commands-before>タグおよび<ftp-commands-after>タグに OPTS UTF {ON|OFF}コマンドを指定した場合は，FTP サーバとの通信に適用される文字セットを次のどちらかで指定しておく必要があります。

・要求電文の<file-name-charset>タグ

・FTP アダプタ実行環境プロパティファイルの ftpadp.file-name.charset プロパティ

●GET オペレーション

ビジネスプロセスが FTP アダプタに渡す GET オペレーションの要求電文のフォーマットを次に示します。ファイル名は「ftpadp_get_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/get_request」です。

表 3-66　要求電文フォーマット（GET オペレーション）

タグ名		出現回数※1	説明
<request>		1 回	－
	<host-ipaddr>	0 または 1 回	接続する FTP サーバの IP アドレスまたはホスト名です。 • タグがあって値がある場合 指定の文字列で connect が発行されます。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。 Windows を使用する場合，ループバック IP アドレス（127.0.0.1 または localhost）を指定すると，FTP 接続に失敗する場合があります。実 IP アドレスまたはホスト名を指定してください。

タグ名			出現回数※1	説明
	<host-con-port>		0 または 1 回	FTP サーバの制御コネクション用ポート番号です。 • タグがあって値がある場合 1 から 65535 までのポート番号を指定します。これ以外の値を指定した場合，システム例外を返します。 • タグがあって値がない場合 デフォルトの 21 が指定されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<file-name-charset>		0 または 1 回	FTP サーバとの制御コネクションで，情報（ファイル名など）を送受信するときに使用する文字セットの名称を示すタグです。指定値は次のどちらかです。 UTF-8：文字セットに「UTF-8」を使用します。 MS932：文字セットに「MS932」を使用します。 • タグがあって値が使用できる文字セット名の場合 情報（ファイル名など）を送受信するときの文字セットとして適用されます。 • タグがあって値が使用できない文字セット名の場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「UTF-8」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<ftps>		0 または 1 回	FTPS に関する設定を示すタグです。タグがない場合，FTPS に関するすべての設定は FTP アダプタ実行環境プロパティファイルの値が有効となります。
		<ftps-enable>	0 または 1 回	FTP サーバとの接続に FTPS を使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：FTP サーバとの接続に FTPS を使用します。 false：FTP サーバとの接続に通常の FTP を使用します。※2 FTPS を使用する場合，制御コネクションおよびデータコネクションの通信は暗号化されます。ただし，接続先の FTP サーバが FTPS による接続をサポートしている必要があります。「true」を指定した場合でも，データコネクションの通信の暗号化が設定されていないときは，データコネクションの通信も暗号化されません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
		<ftps-protocol-name>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，使用するセキュリティ通信用のプロトコルを示すタグです。指定値は次のどちらかです。

タグ名			出現回数※1	説明
		<ftps-protocol-name>	0 または 1 回	TLS：セキュリティ通信用のプロトコルに TLS を使用します。 SSL：セキュリティ通信用のプロトコルに SSL を使用します。 • タグがあって値が「TLS」または「SSL」の場合 指定した値が適用されます。 • タグがあって値が「TLS」または「SSL」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「TLS」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-implicit-mode>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，FTPS の Implicit モードを使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：Implicit モードを使用します。 false：Explicit モードを使用します。※2 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-data-con-secure>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，データコネクションの通信を暗号化するかどうかを指定するタグです。指定値は次のどちらかです。 true：データコネクションの通信を暗号化します。 false：データコネクションの通信を暗号化しません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「true」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-server-authentication>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，サーバ認証をするかどうかを指定するタグです。指定値は次のどちらかです。 true：サーバ認証をします。

タグ名			出現回数※1	説明
		<ftps-server-authentication>	0 または 1 回	false：サーバ認証をしません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。なお，「true」を指定する場合，キーストアに関する設定が必要です。キーストアの設定方法については，「付録 H.1　FTPS を使用したセキュアな接続(FTP アダプタ)」を参照してください。
	<ftp-user>※3		0 または 1 回	接続する FTP サーバのログインユーザ名を示すタグです。USER コマンドの引数として使用します。このタグに指定するログインユーザには，ファイル操作を実行できる権限が与えられている必要があります。 FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 • タグがあって正常な値がある場合 「USER　"値"」が発行されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値がない場合，または値に不正な文字が含まれている場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。
	<ftp-acct>※3		0 または 1 回	接続する FTP サーバの課金情報を示すタグです。FTP アダプタアカウント定義ファイルに重複して設定されている場合，要求電文の値が有効になります。 課金情報を必要とするユーザ 条件に応じて次の値が利用されます。 • タグがあって正常な値がある場合 タグの内容が ACCT コマンドに利用されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値に不正な文字が含まれている場合 システム例外を返します。 • タグがあって値がない場合 「ACCT　""」が発行されます。 • タグがない場合 FTP アダプタアカウント定義ファイルで指定されている ACCT の値が利用されます。FTP アダプタアカウント定義ファイルに ACCT がない場合は「ACCT　""」が発行されます。

タグ名		出現回数※1	説明
	<ftp-acct>※3	0 または 1 回	課金情報を必要としないユーザ 指定は無視されます。
	<ftp-type>	0 または 1 回	転送するデータタイプ（伝送モード）を示すタグです。TYPE コマンドの引数として使用します。FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 指定値は次のどちらかです。 A：ASCII 形式で転送します。 I：バイナリ形式で転送します。 • タグがあって値が「A」または「I」の場合 TYPE コマンドの引数として使用されます。 • タグがあって値が「A」または「I」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 TYPE コマンドは実行されません。※4 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルにも指定がない場合，TYPE コマンドは実行されません。※4
	<ftp-mode>	0 または 1 回	転送モード（圧縮モード）を示すタグです。MODE コマンドの引数として使用します。FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 指定値は次のどちらかです。 S：ファイルを圧縮しないで伝送します。 C：ファイルを圧縮して伝送します。 • タグがあって値が「S」または「C」の場合 MODE コマンドの引数として使用されます。 • タグがあって値が「S」または「C」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 MODE コマンドは実行されません。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効となります。FTP アダプタ実行環境プロパティファイルにも指定がない場合は，デフォルトの「S」で MODE コマンドが実行されます。
	<pasvport>	0 または 1 回	データコネクションの接続モードを示すタグです。 指定値は次のどちらかです。 PASV：PASV モードでデータコネクションを設定します。 PORT：PORT モードでデータコネクションを設定します。 • タグがあって値が「PASV」または「PORT」の場合 指定した値が適用されます。 • タグがあって値が「PASV」または「PORT」のどちらでもない場合

タグ名		出現回数※1	説明
	<pasvport>	0 または 1 回	システム例外を返します。 • タグがあって値がない場合 デフォルトの「PASV」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルにも指定がない場合は，デフォルトの「PASV」でデータコネクションを設定します。 要求電文からデータコネクションの接続モードを設定する場合，FTP アダプタ実行環境プロパティファイルで設定した値に対して，次のプロパティの値のチェック処理が実施されます。 • PASV ftpadp.client-pasv-port.min ftpadp.client-pasv-port.max • PORT ftpadp.listen-port.min ftpadp.listen-port.max チェック処理の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.8.3　FTP アダプタ実行環境プロパティファイル」を参照してください。
	<file-size-check>	0 または 1 回	送信側と受信側で，転送されたファイルのサイズが正しいかどうか確認する機能を，実行するかどうかを指定するタグです。 指定値は次のどちらかです。 true：ファイルサイズを確認します。 false：ファイルサイズを確認しません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<request-id>	0 または 1 回	受付で作成したリクエスト ID を示すタグです。 • タグがあって値がある場合 リクエスト ID として利用されます。 • タグがない，またはタグがあって値がない場合 作業フォルダを利用するときはシステム例外を返します。共通フォルダを利用するときは無視されます。
	<local-folder>	1 回	ローカル側フォルダを示すタグです。 • タグの属性に common="true"が指定されている場合

タグ名		出現回数※1	説明
	<local-folder>	1 回	ローカルフォルダとして共通フォルダを利用し，このタグには「共通フォルダ定義名」を指定します。 指定された共通フォルダ定義名から共通フォルダが取得できないときは，システム例外を返します。 • タグの属性に common="false"が指定されている場合 ローカルフォルダとして作業フォルダを利用します。 • common 属性がない，または値が"true"および"false"のどちらでもない場合 システム例外を返します。
	<local-file-name>	0 または 1 回	ローカル側ファイル名を示すタグです。 作業フォルダを利用する場合 • タグがあって値があるとき この設定は無視されて処理が続行されます。 • タグがない，またはタグがあって値がないとき 処理が続行されます。 共通フォルダを利用する場合 共通フォルダルート配下のファイルを指定してください。 指定したファイル名でファイルを作成します。 存在するファイル名を指定した場合，ファイルが上書きされます。 共通フォルダルートからの相対パスで指定したサブフォルダが存在しない場合は，サブフォルダを作成します。 また，シンボリックリンク以外のファイルを指定してください。 • タグがあって値が上記の制限に従っているとき ファイル名として使用されます。 • タグがあって値が上記の制限に違反するとき システム例外を返します。 • タグがない，またはタグがあって値がないとき システム例外を返します。
	<remote-path>※5	1 回	リモートパス名（FTP サーバ上にある取得対象パス名）を示すタグです。 • タグがあって値がある場合 リモートパス名として利用されます。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
	<ftp-commands-before>※6	0 または 1 回	ファイル転送前に実行する FTP コマンドとその引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します。（例：MKD transdir;CWD transdir） • タグがあって値がある場合 ファイル転送前に実行するコマンドとして利用されます。 • タグがない，またはタグがあって値がない場合 ファイル転送前に実行するコマンドはありません。

タグ名		出現回数※1	説明
	<ftp-commands-after>※6	0 または 1 回	ファイル転送後に実行する FTP コマンドとその引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します。（例：RNFR oldfile.txt;RNTO newfile.txt） • タグがあって値がある場合 ファイル転送後に実行するコマンドとして利用されます。 • タグがない，またはタグがあって値がない場合 ファイル転送後に実行するコマンドはありません。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

接続先 FTP サーバが FTPS の Implicit モードで動作している場合に<ftps-enable>タグ，または<ftps-implicit-mode>タグの値に false を指定すると，FTP サーバに接続したあとタイムアウトが発生するまで待ち状態となることがあります。

タイムアウトが発生するまでの時間は，次の値のうちどちらか小さい方です。

・FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値

・接続先 FTP サーバ側で設定されたタイムアウト値

FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値でタイムアウトとなった場合，FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.retry.count プロパティで指定した回数のリトライを行ったあと，KDEK30407-E のエラーが発生します。

接続先 FTP サーバ側で設定されたタイムアウト値でタイムアウトとなった場合，KDEK30428-E のエラーが発生します。

注※3

使用できる文字を次に示します。

半角英数字，半角空白，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

注※4

FTP プロトコルの仕様（RFC959）に従い，デフォルトでは ASCII 形式で転送されます。

注※5

FTP 受付の要求電文の<transfer-path>タグと同じ値を FTP アダプタの<remote-path>タグに設定すると，パスの先頭がスラッシュ（/）になります。

パスの先頭にスラッシュ（/）があると問題がある場合は，ビジネスプロセスのデータ変換アクティビティで部分文字列取得ファンクション（substr）を使用して FTP 受付の要求電文の<transfer-path>タグの値から 2 文字目以降の文字列を取り出して，FTP アダプタの<remote-path>タグに設定してください。パスの先頭にスラッシュ（/）があると問題がある場合とは，相対パスを指定する場合や Windows の絶対パス形式でパスを指定したい場合などです。

注※6

<ftp-commands-before>タグおよび<ftp-commands-after>タグに指定されたコマンドは，そのまま FTP サーバに送られ，FTP アダプタはコマンドのオプションの妥当性やコマンドの実行による影響について関知しません。

そのため，<ftp-commands-before>タグおよび<ftp-commands-after>タグに OPTS UTF {ON|OFF}コマンドを指定した場合は，FTP サーバとの通信に適用される文字セットを次のどちらかで指定しておく必要があります。

・要求電文の<file-name-charset>タグ
・FTP アダプタ実行環境プロパティファイルの ftpadp.file-name.charset プロパティ

●MGET オペレーション

ビジネスプロセスが FTP アダプタに渡す MGET オペレーションの要求電文のフォーマットを次に示します。ファイル名は「ftpadp_mget_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/mget_request」です。

表 3-67　要求電文フォーマット（MGET オペレーション）

タグ名		出現回数※1	説明
<request>		1 回	−
	<host-ipaddr>	0 または 1 回	接続する FTP サーバの IP アドレスまたはホスト名です。 • タグがあって値がある場合 指定の文字列で connect が発行されます。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。 Windows を使用する場合，ループバック IP アドレス（127.0.0.1 または localhost）を指定すると，FTP 接続に失敗する場合があります。実 IP アドレスまたはホスト名を指定してください。
	<host-con-port>	0 または 1 回	FTP サーバの制御コネクション用ポート番号です。 • タグがあって値がある場合 1 から 65535 までのポート番号を指定します。これ以外の値を指定した場合，システム例外を返します。 • タグがあって値がない場合 デフォルトの 21 が指定されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<file-name-charset>	0 または 1 回	FTP サーバとの制御コネクションで，情報（ファイル名など）を送受信するときに使用する文字セットの名称を示すタグです。指定値は次のどちらかです。 UTF-8：文字セットに「UTF-8」を使用します。 MS932：文字セットに「MS932」を使用します。 • タグがあって値が使用できる文字セット名の場合 情報（ファイル名など）を送受信するときの文字セットとして適用されます。 • タグがあって値が使用できない文字セット名の場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「UTF-8」が使用されます。 • タグがない場合

タグ名			出現回数※1	説明
	<file-name-charset>		0 または 1 回	FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<ftps>		0 または 1 回	FTPS に関する設定を示すタグです。タグがない場合，FTPS に関するすべての設定は FTP アダプタ実行環境プロパティファイルの値が有効となります。
		<ftps-enable>	0 または 1 回	FTP サーバとの接続に FTPS を使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：FTP サーバとの接続に FTPS を使用します。 false：FTP サーバとの接続に通常の FTP を使用します。※2 FTPS を使用する場合，制御コネクションおよびデータコネクションの通信は暗号化されます。ただし，接続先の FTP サーバが FTPS による接続をサポートしている必要があります。「true」を指定した場合でも，データコネクションの通信の暗号化が設定されていないときは，データコネクションの通信も暗号化されません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
		<ftps-protocol-name>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，使用するセキュリティ通信用のプロトコルを示すタグです。指定値は次のどちらかです。 TLS：セキュリティ通信用のプロトコルに TLS を使用します。 SSL：セキュリティ通信用のプロトコルに SSL を使用します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「TLS」または「SSL」の場合 指定した値が適用されます。 • タグがあって値が「TLS」または「SSL」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「TLS」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-implicit-mode>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，FTPS の Implicit モードを使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：Implicit モードを使用します。 false：Explicit モードを使用します。※2 また，タグの有無に応じて次のように動作します。

タグ名			出現回数※1	説明
		<ftps-implicit-mode>	0 または 1 回	• タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-data-consecure>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，データコネクションの通信を暗号化するかどうかを指定するタグです。指定値は次のどちらかです。 true：データコネクションの通信を暗号化します。 false：データコネクションの通信を暗号化しません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「true」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-server-authentication>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，サーバ認証をするかどうかを指定するタグです。指定値は次のどちらかです。 true：サーバ認証をします。 false：サーバ認証をしません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。なお，「true」を指定する場合，キーストアに関する設定が必要です。キーストアの設定方法については，「付録 H.1　FTPS を使用したセキュアな接続(FTP アダプタ)」を参照してください。

タグ名		出現回数※1	説明
	<ftp-user>※3	0 または 1 回	接続する FTP サーバのログインユーザ名を示すタグです。USER コマンドの引数として使用します。このタグに指定するログインユーザには，ファイル操作を実行できる権限が与えられている必要があります。 FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 また，タグの有無に応じて次のように動作します。 • タグがあって正常な値がある場合 「USER　"値"」が発行されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値がない場合，または値に不正な文字が含まれている場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。
	<ftp-acct>※3	0 または 1 回	接続する FTP サーバの課金情報を示すタグです。FTP アダプタアカウント定義ファイルに重複して設定されている場合，要求電文の値が有効になります。 課金情報を必要とするユーザ 条件に応じて次の値が利用されます。 • タグがあって正常な値がある場合 タグの内容が ACCT コマンドに利用されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値に不正な文字が含まれている場合 システム例外を返します。 • タグがあって値がない場合 「ACCT　""」が発行されます。 • タグがない場合 FTP アダプタアカウント定義ファイルで指定されている ACCT の値が利用されます。FTP アダプタアカウント定義ファイルに ACCT がない場合は「ACCT　""」が発行されます。 課金情報を必要としないユーザ 指定は無視されます。
	<ftp-type>	0 または 1 回	転送するデータタイプ（伝送モード）を示すタグです。TYPE コマンドの引数として使用します。FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 指定値は次のどちらかです。 A：ASCII 形式で転送します。 I：バイナリ形式で転送します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「A」または「I」の場合 TYPE コマンドの引数として使用されます。 • タグがあって値が「A」または「I」のどちらでもない場合

タグ名		出現回数※1	説明
	<ftp-type>	0 または 1 回	システム例外を返します。 • タグがあって値がない場合 TYPE コマンドは実行されません。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの ftpadp.ftp-type プロパティの値で TYPE コマンドが実行されます。このプロパティの指定がない場合，TYPE コマンドは実行されません。
	<ftp-mode>	0 または 1 回	転送モード（圧縮モード）を示すタグです。MODE コマンドの引数として使用します。FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 指定値は次のどちらかです。 S：ファイルを圧縮しないで伝送します。 C：ファイルを圧縮して伝送します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「S」または「C」の場合 MODE コマンドの引数として使用されます。 • タグがあって値が「S」または「C」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 MODE コマンドは実行されません。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの ftpadp.ftp-mode プロパティに指定された値で MODE コマンドが実行されます。このプロパティの指定がない場合，「MODE S」が実行されます。
	<pasvport>	0 または 1 回	データコネクションの接続モードを示すタグです。 指定値は次のどちらかです。 PASV：PASV モードでデータコネクションを設定します。 PORT：PORT モードでデータコネクションを設定します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「PASV」または「PORT」の場合 指定した値が適用されます。 • タグがあって値が「PASV」または「PORT」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「PASV」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルにも指定がない場合は，デフォルトの「PASV」でデータコネクションを設定します。
	<file-size-check>	0 または 1 回	送信側と受信側で，転送されたファイルのサイズが正しいかどうか確認する機能を，実行するかどうかを指定するタグです。 指定値は次のどちらかです。

タグ名			出現回数※1	説明
	<file-size-check>		0 または 1 回	true：ファイルサイズを確認します。 false：ファイルサイズを確認しません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルにも指定がない場合は，デフォルトの「false」が使用されます。
	<request-id>		0 または 1 回	受付で作成したリクエスト ID を示すタグです。 • タグがあって値がある場合 リクエスト ID として利用されます。 • タグがない，またはタグがあって値がない場合 作業フォルダを利用するときはシステム例外を返します。共通フォルダを利用するときは無視されます。
	<transfer-unit>		1 回以上	転送する個々のファイルに関する情報を指定します。
		<local-folder>	1 回	ローカル側フォルダを示すタグです。 • タグの属性に common="true"が指定されている場合 ローカルフォルダとして共通フォルダを利用し，このタグには「共通フォルダ定義名」を指定します。 指定された共通フォルダ定義名から共通フォルダが取得できないときは，システム例外を返します。 • タグの属性に common="false"が指定されている場合 ローカルフォルダとして作業フォルダを利用します。 • common 属性がない，または値が「true」および「false」のどちらでもない場合 システム例外を返します。
		<local-file-name>	0 または 1 回	ローカル側ファイル名を示すタグです。 作業フォルダを利用する場合 次の両方の条件を満たす必要があります。 • 作業フォルダ直下のファイルであること。 • 指定値（ファイル名）の途中に区切り文字（/や¥）を含まないこと。 指定値の先頭には「/」を指定できますが，「/」は無視されます。 共通フォルダを利用する場合 共通フォルダルート配下のファイルを指定してください。指定したファイル名でファイルを作成します。 また，タグの有無に応じて次のように動作します。

タグ名			出現回数※1	説明
		<local-file-name>	0 または 1 回	• タグがあって値が上記の制限に従っているとき 指定値がファイル名として使用されます。 • タグがあって値が上記の制限に違反するとき システム例外を返します。 • タグがない，またはタグがあって値がないとき システム例外を返します。
		<remote-path>※4	1 回	リモートパス名（FTP サーバ上にある取得対象パス名）を示すタグです。 • タグがあって値がある場合 リモートパス名として利用されます。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
	<ftp-commands-before>※5		0 または 1 回	ファイル転送前に実行する FTP コマンドとその引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します。（例：MKD transdir;CWD transdir） • タグがあって値がある場合 ファイル転送前に実行するコマンドとして利用されます。 • タグがない，またはタグがあって値がない場合 ファイル転送前に実行するコマンドはありません。
	<ftp-commands-after>※5		0 または 1 回	ファイル転送後に実行する FTP コマンドとその引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します。（例：RNFR oldfile.txt;RNTO newfile.txt） • タグがあって値がある場合 ファイル転送後に実行するコマンドとして利用されます。 • タグがない，またはタグがあって値がない場合 ファイル転送後に実行するコマンドはありません。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

接続先 FTP サーバが FTPS の Implicit モードで動作している場合に<ftps-enable>タグ，または<ftps-implicit-mode>タグの値に false を指定すると，FTP サーバに接続したあとタイムアウトが発生するまで待ち状態となることがあります。

タイムアウトが発生するまでの時間は，次の値のうちどちらか小さい方です。

・FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値

・接続先 FTP サーバ側で設定されたタイムアウト値

FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値でタイムアウトとなった場合，FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.retry.count プロパティで指定した回数のリトライを行ったあと，KDEK30407-E のエラーが発生します。

接続先 FTP サーバ側で設定されたタイムアウト値でタイムアウトとなった場合，KDEK30428-E のエラーが発生します。

注※3

使用できる文字を次に示します。

半角英数字，半角空白，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

注※4

FTP 受付の要求電文の<transfer-path>タグと同じ値を FTP アダプタの<remote-path>タグに設定すると，パスの先頭がスラッシュ（/）になります。

パスの先頭にスラッシュ（/）があると問題がある場合は，ビジネスプロセスのデータ変換アクティビティで部分文字列取得ファンクション（substr）を使用して FTP 受付の要求電文の<transfer-path>タグの値から 2 文字目以降の文字列を取り出して，FTP アダプタの<remote-path>タグに設定してください。パスの先頭にスラッシュ（/）があると問題がある場合とは，相対パスを指定する場合や Windows の絶対パス形式でパスを指定したい場合などです。

注※5

<ftp-commands-before>タグおよび<ftp-commands-after>タグに指定されたコマンドは，そのまま FTP サーバに送られ，FTP アダプタはコマンドのオプションの妥当性やコマンドの実行による影響について関知しません。

そのため，<ftp-commands-before>タグおよび<ftp-commands-after>タグに OPTS UTF {ON|OFF}コマンドを指定した場合は，FTP サーバとの通信に適用される文字セットを次のどちらかで指定しておく必要があります。

・要求電文の<file-name-charset>タグ

・FTP アダプタ実行環境プロパティファイルの ftpadp.file-name.charset プロパティ

●GETINFO オペレーション

ビジネスプロセスが FTP アダプタに渡す GETINFO オペレーションの要求電文のフォーマットを次に示します。ファイル名は「ftpadp_getinfo_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/getinfo_request」です。

表 3-68　要求電文フォーマット（GETINFO オペレーション）

タグ名		出現回数※1	説明
<request>		1 回	−
	<host-ipaddr>	0 または 1 回	接続する FTP サーバの IP アドレスまたはホスト名です。 • タグがあって値がある場合 指定の文字列で connect が発行されます。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。 Windows を使用する場合，ループバック IP アドレス（127.0.0.1 または localhost）を指定すると，FTP 接続に失敗する場合があります。実 IP アドレスまたはホスト名を指定してください。
	<host-con-port>	0 または 1 回	FTP サーバの制御コネクション用ポート番号です。 • タグがあって値がある場合

タグ名			出現回数※1	説明
	<host-con-port>		0 または 1 回	1 から 65535 までのポート番号を指定します。これ以外の値を指定した場合，システム例外を返します。 • タグがあって値がない場合 デフォルトの 21 が指定されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<file-name-charset>		0 または 1 回	FTP サーバとの間の制御コネクションで，次に示す情報を送受信するときに使用する文字セットの名称を示すタグです。 • 制御コネクションで送受信する情報（ファイル名など） • LIST コマンドまたは NLST コマンド実行時にデータコネクションを介して FTP サーバから受け取る情報（ファイルの一覧情報） 指定値は次のどちらかです。 UTF-8：文字セットに「UTF-8」を使用します。 MS932：文字セットに「MS932」を使用します。 • タグがあって値が使用できる文字セットの場合 情報（ファイル名など）を送受信するときの文字セットとして適用されます。 • タグがあって値が使用できない文字セット名の場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「UTF-8」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<ftps>		0 または 1 回	FTPS に関する設定を示すタグです。タグがない場合，FTPS に関するすべての設定は FTP アダプタ実行環境プロパティファイルの値が有効となります。
		<ftps-enable>	0 または 1 回	FTP サーバとの接続に FTPS を使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：FTP サーバとの接続に FTPS を使用します。 false：FTP サーバとの接続に通常の FTP を使用します。※2 FTPS を使用する場合，制御コネクションおよびデータコネクションの通信は暗号化されます。ただし，接続先の FTP サーバが FTPS による接続をサポートしている必要があります。「true」を指定した場合でも，データコネクションの通信の暗号化が設定されていないときは，データコネクションの通信も暗号化されません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。

タグ名			出現回数※1	説明
		<ftps-protocol-name>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，使用するセキュリティ通信用のプロトコルを示すタグです。指定値は次のどちらかです。 TLS：セキュリティ通信用のプロトコルに TLS を使用します。 SSL：セキュリティ通信用のプロトコルに SSL を使用します。 • タグがあって値が「TLS」または「SSL」の場合 指定した値が適用されます。 • タグがあって値が「TLS」または「SSL」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「TLS」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-implicit-mode>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，FTPS の Implicit モードを使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：Implicit モードを使用します。 false：Explicit モードを使用します。※2 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-data-connsecure>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，データコネクションの通信を暗号化するかどうかを指定するタグです。指定値は次のどちらかです。 true：データコネクションの通信を暗号化します。 false：データコネクションの通信を暗号化しません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「true」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-server-authentication>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，サーバ認証をするかどうかを指定するタグです。指定値は次のどちらかです。

タグ名			出現回数※1	説明
		<ftps-server-authentication>	0 または 1 回	true：サーバ認証をします。 false：サーバ認証をしません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。なお，「true」を指定する場合，キーストアに関する設定が必要です。キーストアの設定方法については，「付録 H.1　FTPS を使用したセキュアな接続(FTP アダプタ)」を参照してください。
	<ftp-user>※3		0 または 1 回	接続する FTP サーバのログインユーザ名を示すタグです。USER コマンドの引数として使用します。ログインユーザ名は 80 文字以内で指定してください。このタグに指定するログインユーザには，ファイル操作を実行できる権限が与えられている必要があります。 FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 • タグがあって正常な値がある場合 「USER　"値"」が発行されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値がない場合，または値に不正な文字が含まれている場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。
	<ftp-acct>※3		0 または 1 回	接続する FTP サーバの課金情報を示すタグです。FTP アダプタアカウント定義ファイルに重複して設定されている場合，要求電文の値が有効になります。 課金情報を必要とするユーザ 条件に応じて次の値が利用されます。 • タグがあって正常な値がある場合 タグの内容が ACCT コマンドに利用されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値に不正な文字が含まれている場合 システム例外を返します。 • タグがあって値がない場合 「ACCT　""」が発行されます。 • タグがない場合

タグ名		出現回数※1	説明
	<ftp-acct>※3	0 または 1 回	FTP アダプタアカウント定義ファイルで指定されている ACCT の値が利用されます。FTP アダプタアカウント定義ファイルに ACCT がない場合は「ACCT　""」が発行されます。 課金情報を必要としないユーザ 指定は無視されます。
	<pasvport>	0 または 1 回	データコネクションの接続モードを示すタグです。 指定値は次のどちらかです。 PASV：PASV モードでデータコネクションを設定します。 PORT：PORT モードでデータコネクションを設定します。 • タグがあって値が「PASV」または「PORT」の場合 指定した値が適用されます。 • タグがあって値が「PASV」または「PORT」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「PASV」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルにも指定がない場合は，デフォルトの「PASV」でデータコネクションを設定します。 要求電文からデータコネクションの接続モードを設定する場合，FTP アダプタ実行環境プロパティファイルで設定した値に対して，次のプロパティの値のチェック処理が実施されます。 • PASV ftpadp.client-pasv-port.min ftpadp.client-pasv-port.max • PORT ftpadp.listen-port.min ftpadp.listen-port.max チェック処理の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.8.3　FTP アダプタ実行環境プロパティファイル」を参照してください。
	<getinfo-type>	1 回	接続する FTP サーバから取得する情報の取得種別を示すタグです。 設定値は次のどちらかです。 LIST：LIST コマンドを実行し，ファイル情報の一覧を取得します。 NLST：NLST コマンドを実行し，ファイル名の一覧を取得します。 • タグがあって値が「LIST」または「NLST」の場合 指定した値を取得種別として利用します。 • タグがあって値が「LIST」または「NLST」のどちらでもない場合 システム例外を返します。 • タグがない，またはタグがあって値がない場合 システム例外を返します。

タグ名		出現回数※1	説明
	<getinfo-option>	0 または 1 回	接続する FTP サーバから取得する情報の取得オプションを示すタグです。<getinfo-type>で指定したコマンドを実行するときに指定するオプションです。 • タグがあって値がある場合 指定した値がオプションとして指定されて，<getinfo-type>で指定したコマンドが実行されます。 • タグがない，またはタグがあって値がない場合 オプションなしで，<getinfo-type>で指定したコマンドが実行されます。
	<getinfo-path>	0 または 1 回	接続する FTP サーバから取得する情報の取得パスを示すタグです。<getinfo-type>で指定したコマンドを実行するときに指定するパスです。 • タグがあって値がある場合 指定した値がパスとして指定されて，<getinfo-type>で指定したコマンドが実行されます。 • タグがない，またはタグがあって値がない場合 パスなしで，<getinfo-type>で指定したコマンドが実行されます。
	<request-id>	0 または 1 回	FTP 受付で作成したリクエスト ID を示すタグです。※5 • タグがあって値がある場合 ローカルフォルダとして作業フォルダが使用されます。指定されている値はリクエスト ID として利用されます。 <local-folder>の属性に common="true"以外が指定されているときだけ有効になります。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 <local-folder>タグを省略しているときは応答電文に出力されます。
	<local-folder>	0 または 1 回	ローカル側フォルダを示すタグです。※5 • タグの属性に common="true"が指定されている場合 ローカルフォルダとして共通フォルダを利用し，このタグには「共通フォルダ定義名」を指定します。 • タグの属性に common="false"が指定されている場合 ローカルフォルダとして作業フォルダを利用します。 • common 属性がない，または値が"true"および"false"のどちらでもない場合 システム例外を返します。 • タグがない場合 ローカルフォルダとして作業フォルダを利用します。ただし，<request-id>を省略したときは応答電文に出力されます。
	<local-file-name>	0 または 1 回	ローカル側ファイル名を示すタグです。※5 作業フォルダを利用する場合 タグの指定は無視されて処理が続行されます。

タグ名		出現回数※1	説明
	<local-file-name>	0 または 1 回	共通フォルダを利用する場合 共通フォルダルート配下のファイル名を指定してください。 指定したファイル名でファイルを作成します。 存在するファイル名を指定した場合，ファイルが上書きされます。 共通フォルダルートからの相対パスで指定したサブフォルダが存在しない場合は，サブフォルダを作成します。 シンボリックリンク以外のファイルを指定してください。 • タグがあって値が上記の制限に従っているとき ファイル名として使用されます。 • タグがあって値が上記の制限に違反するとき システム例外を返します。 • タグがない，またはタグがあって値がないとき システム例外を返します。 応答電文に出力する場合 タグの指定は無視されて処理が続行されます。
	<ftp-commands-before>※4	0 または 1 回	リストコマンド実行前に実行する FTP コマンドとその引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します（例：MKD transdir;CWD transdir）。 • タグがあって値がある場合 リストコマンド実行前に実行するコマンドとして利用されます。 • タグがない，またはタグがあって値がない場合 リストコマンド実行前に実行するコマンドはありません。
	<ftp-commands-after>※4	0 または 1 回	リストコマンド実行後に実行する FTP コマンドとその引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します（例：RNFR oldfile.txt;RNTO newfile.txt）。 • タグがあって値がある場合 リストコマンド実行後に実行するコマンドとして利用されます。 • タグがない，またはタグがあって値がない場合 リストコマンド実行後に実行するコマンドはありません。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

接続先 FTP サーバが FTPS の Implicit モードで動作している場合に<ftps-enable>タグ，または<ftps-implicit-mode>タグの値に false を指定すると，FTP サーバに接続したあとタイムアウトが発生するまで待ち状態となることがあります。

タイムアウトが発生するまでの時間は，次の値のうちどちらか小さい方です。

・FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値

・接続先 FTP サーバ側で設定されたタイムアウト値

FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値でタイムアウトとなった場合，FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.retry.count プロパティで指定した回数のリトライを行ったあと，KDEK30407-E のエラーが発生します。

接続先 FTP サーバ側で設定されたタイムアウト値でタイムアウトとなった場合，KDEK30428-E のエラーが発生します。

注※3

使用できる文字を次に示します。

半角英数字，半角空白，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

注※4

<ftp-commands-before>タグおよび<ftp-commands-after>タグに指定されたコマンドは，そのまま FTP サーバに送られ，FTP アダプタはコマンドのオプションの妥当性やコマンドの実行による影響について関知しません。

そのため，<ftp-commands-before>タグおよび<ftp-commands-after>タグに OPTS UTF {ON|OFF}コマンドを指定した場合は，FTP サーバとの通信に適用される文字セットを次のどちらかで指定しておく必要があります。

・要求電文の<file-name-charset>タグ

・FTP アダプタ実行環境プロパティファイルの ftpadp.file-name.charset プロパティ

注※5

GETINFO オペレーションの定義方法については，「(c) GETINFO オペレーションを使用する場合の要求電文および応答電文」を参照してください。

●LOGIN オペレーション

ビジネスプロセスが FTP アダプタに渡す LOGIN オペレーションの要求電文のフォーマットを次に示します。ファイル名は「ftpadp_login_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/login_request」です。

表 3-69　要求電文フォーマット（LOGIN オペレーション）

タグ名		出現回数※1	説明
<request>		1 回	−
	<host-ipaddr>	0 または 1 回	接続する FTP サーバの IP アドレスまたはホスト名です。 • タグがあって値がある場合 指定の文字列で connect が発行されます。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。 Windows を使用する場合，ループバック IP アドレス（127.0.0.1 または localhost）を指定すると，FTP 接続に失敗する場合があります。実 IP アドレスまたはホスト名を指定してください。

タグ名			出現回数※1	説明
	<host-con-port>		0 または 1 回	FTP サーバの制御コネクション用ポート番号です。 • タグがあって値がある場合 1 から 65535 までのポート番号を指定します。これ以外の値を指定した場合，システム例外を返します。 • タグがあって値がない場合 デフォルトの 21 が指定されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<ftps>		0 または 1 回	FTPS に関する設定を示すタグです。タグがない場合，FTPS に関するすべての設定は FTP アダプタ実行環境プロパティファイルの値が有効となります。
		<ftps-enable>	0 または 1 回	FTP サーバとの接続に FTPS を使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：FTP サーバとの接続に FTPS を使用します。 false：FTP サーバとの接続に通常の FTP を使用します。※2 FTPS を使用する場合，制御コネクションの通信は暗号化されます。ただし，接続先の FTP サーバが FTPS による接続をサポートしている必要があります。 データコネクションの通信を暗号化するには，<ftps-data-con-secure>を true に設定する必要があります。 通常の FTP を使用する場合，通信は暗号化されません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
		<ftps-protocol-name>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，使用するセキュリティ通信用のプロトコルを示すタグです。指定値は次のどちらかです。 TLS：セキュリティ通信用のプロトコルに TLS を使用します。 SSL：セキュリティ通信用のプロトコルに SSL を使用します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「TLS」または「SSL」の場合 指定した値が適用されます。 • タグがあって値が「TLS」または「SSL」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「TLS」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。

タグ名			出現回数※1	説明
		<ftps-protocol-name>	0 または 1 回	この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-implicit-mode>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，FTPS の Implicit モードを使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：Implicit モードを使用します。 false：Explicit モードを使用します。※2 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-data-con-secure>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，データコネクションの通信を暗号化するかどうかを指定するタグです。指定値は次のどちらかです。 true：データコネクションの通信を暗号化します。 false：データコネクションの通信を暗号化しません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「true」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-server-authentication>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，サーバ認証をするかどうかを指定するタグです。指定値は次のどちらかです。 true：サーバ認証をします。 false：サーバ認証をしません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。

タグ名			出現回数※1	説明
		<ftps-server-authentication>	0 または 1 回	• タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。なお，「true」を指定する場合，キーストアに関する設定が必要です。キーストアの設定方法については，「付録 H.1　FTPS を使用したセキュアな接続(FTP アダプタ)」を参照してください。
	<ftp-user>※3		0 または 1 回	接続する FTP サーバのログインユーザ名を示すタグです。USER コマンドの引数として使用します。ログインユーザ名は 80 文字以内で指定してください。このタグに指定するログインユーザには，ファイル操作を実行できる権限が与えられている必要があります。 FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 また，タグの有無に応じて次のように動作します。 • タグがあって正常な値がある場合 「USER　"値"」が発行されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値がない場合，または値に不正な文字が含まれている場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。
	<ftp-acct>※3		0 または 1 回	接続する FTP サーバの課金情報を示すタグです。FTP アダプタアカウント定義ファイルに重複して設定されている場合，要求電文の値が有効になります。 課金情報を必要とするユーザ 条件に応じて次の値が利用されます。 • タグがあって正常な値がある場合 タグの内容が ACCT コマンドに利用されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値に不正な文字が含まれている場合 システム例外を返します。 • タグがあって値がない場合 「ACCT　""」が発行されます。 • タグがない場合 FTP アダプタアカウント定義ファイルで指定されている ACCT の値が利用されます。FTP アダプタアカウント定義ファイルに ACCT がない場合は「ACCT　""」が発行されます。 課金情報を必要としないユーザ 指定は無視されます。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

接続先 FTP サーバが FTPS の Implicit モードで動作している場合に<ftps-enable>タグ，または<ftps-implicit-mode>タグの値に false を指定すると，FTP サーバに接続したあとタイムアウトが発生するまで待ち状態となることがあります。

タイムアウトが発生するまでの時間は，次の値のうちどちらか小さい方です。

・FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値

・接続先 FTP サーバ側で設定されたタイムアウト値

FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値でタイムアウトとなった場合，FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.retry.count プロパティで指定した回数のリトライを行ったあと，KDEK30407-E のエラーが発生します。

接続先 FTP サーバ側で設定されたタイムアウト値でタイムアウトとなった場合，KDEK30428-E のエラーが発生します。

注※3

使用できる文字を次に示します。

半角英数字，半角空白，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

●LOGOUT オペレーション

ビジネスプロセスが FTP アダプタに渡す LOGOUT オペレーションの要求電文のフォーマットを次に示します。ファイル名は「ftpadp_logout_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/logout_request」です。

表 3-70　要求電文フォーマット（LOGOUT オペレーション）

タグ名		出現回数※1	説明
<request>		1 回	−
	<host-ipaddr>	0 または 1 回	接続する FTP サーバの IP アドレスまたはホスト名です。 • タグがあって値がある場合 指定の文字列で connect が発行されます。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。 Windows を使用する場合，ループバック IP アドレス（127.0.0.1 または localhost）を指定すると，FTP 接続に失敗する場合があります。実 IP アドレスまたはホスト名を指定してください。
	<host-con-port>	0 または 1 回	FTP サーバの制御コネクション用ポート番号です。 • タグがあって値がある場合

タグ名			出現回数※1	説明
	<host-con-port>		0 または 1 回	1 から 65535 までのポート番号を指定します。これ以外の値を指定した場合，システム例外を返します。 • タグがあって値がない場合 デフォルトの 21 が指定されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<ftps>		0 または 1 回	FTPS に関する設定を示すタグです。タグがない場合，FTPS に関するすべての設定は FTP アダプタ実行環境プロパティファイルの値が有効となります。
		<ftps-enable>	0 または 1 回	FTP サーバとの接続に FTPS を使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：FTP サーバとの接続に FTPS を使用します。 false：FTP サーバとの接続に通常の FTP を使用します。 FTPS を使用する場合，制御コネクションの通信は暗号化されます。ただし，接続先の FTP サーバが FTPS による接続をサポートしている必要があります。 データコネクションの通信を暗号化するには，<ftps-data-con-secure>を true に設定する必要があります。 通常の FTP を使用する場合，通信は暗号化されません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
		<ftps-protocol-name>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，使用するセキュリティ通信用のプロトコルを示すタグです。指定値は次のどちらかです。 TLS：セキュリティ通信用のプロトコルに TLS を使用します。 SSL：セキュリティ通信用のプロトコルに SSL を使用します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「TLS」または「SSL」の場合 指定した値が適用されます。 • タグがあって値が「TLS」または「SSL」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「TLS」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。

タグ名			出現回数※1	説明
		<ftps-implicit-mode>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，FTPS の Implicit モードを使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：Implicit モードを使用します。 false：Explicit モードを使用します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-data-con-secure>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，データコネクションの通信を暗号化するかどうかを指定するタグです。指定値は次のどちらかです。 true：データコネクションの通信を暗号化します。 false：データコネクションの通信を暗号化しません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「true」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-server-authentication>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，サーバ認証をするかどうかを指定するタグです。指定値は次のどちらかです。 true：サーバ認証をします。 false：サーバ認証をしません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。

タグ名			出現回数※1	説明
		<ftps-server-authentication>	0 または 1 回	この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。なお，「true」を指定する場合，キーストアに関する設定が必要です。キーストアの設定方法については，「付録 H.1　FTPS を使用したセキュアな接続(FTP アダプタ)」を参照してください。
	<ftp-user>※2		0 または 1 回	接続する FTP サーバのログインユーザ名を示すタグです。USER コマンドの引数として使用します。ログインユーザ名は 80 文字以内で指定してください。このタグに指定するログインユーザには，ファイル操作を実行できる権限が与えられている必要があります。 FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 • タグがあって正常な値がある場合 「USER　"値"」が発行されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値がない場合，または値に不正な文字が含まれている場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。
	<ftp-acct>※2		0 または 1 回	接続する FTP サーバの課金情報を示すタグです。FTP アダプタアカウント定義ファイルに重複して設定されている場合，要求電文の値が有効になります。 課金情報を必要とするユーザ 条件に応じて次の値が利用されます。 • タグがあって正常な値がある場合 タグの内容が ACCT コマンドに利用されます。 • タグがあって値に不正な文字が含まれている場合 システム例外を返します。 • タグがあって値がない場合 「ACCT　""」が発行されます。 • タグがない場合 FTP アダプタアカウント定義ファイルで指定されている ACCT の値が利用されます。FTP アダプタアカウント定義ファイルに ACCT がない場合は「ACCT　""」が発行されます。 課金情報を必要としないユーザ 指定は無視されます。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

使用できる文字を次に示します。

半角英数字，半角空白，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

●RENAME オペレーション

ビジネスプロセスが FTP アダプタに渡す RENAME オペレーションの要求電文のフォーマットを次に示します。ファイル名は「ftpadp_rename_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/rename_request」です。

表 3-71　要求電文フォーマット（RENAME オペレーション）

タグ名		出現回数※1	説明
<request>		1 回	−
	<host-ipaddr>	0 または 1 回	接続する FTP サーバの IP アドレスまたはホスト名です。 • タグがあって値がある場合 指定の文字列で connect が発行されます。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。 Windows を使用する場合，ループバック IP アドレス（127.0.0.1 または localhost）を指定すると，FTP 接続に失敗する場合があります。実 IP アドレスまたはホスト名を指定してください。
	<host-con-port>	0 または 1 回	FTP サーバの制御コネクション用ポート番号です。 • タグがあって値がある場合 1 から 65535 までのポート番号を指定します。これ以外の値を指定した場合，システム例外を返します。 • タグがあって値がない場合 デフォルトの 21 が指定されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<file-name-charset>	0 または 1 回	FTP サーバとの間の制御コネクションで，次に示す情報を送受信するときに使用する文字セットの名称を示すタグです。 • 制御コネクションで送受信する情報（ファイル名など） 指定値は次のどちらかです。 UTF-8：文字セットに「UTF-8」を使用します。 MS932：文字セットに「MS932」を使用します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が使用できる文字セットの場合

タグ名			出現回数※1	説明
	<file-name-charset>		0 または 1 回	情報（ファイル名など）を送受信するときの文字セットとして適用されます。 • タグがあって値が使用できない文字セット名の場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「UTF-8」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
	<ftps>		0 または 1 回	FTPS に関する設定を示すタグです。タグがない場合，FTPS に関するすべての設定は FTP アダプタ実行環境プロパティファイルの値が有効となります。
		<ftps-enable>	0 または 1 回	FTP サーバとの接続に FTPS を使用するかどうかを指定するタグです。指定値は次のどちらかです。 true：FTP サーバとの接続に FTPS を使用します。 false：FTP サーバとの接続に通常の FTP を使用します。※2 FTPS を使用する場合，制御コネクションおよびデータコネクションの通信は暗号化されます。ただし，接続先の FTP サーバが FTPS による接続をサポートしている必要があります。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。
		<ftps-protocol-name>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，使用するセキュリティ通信用のプロトコルを示すタグです。指定値は次のどちらかです。 TLS：セキュリティ通信用のプロトコルに TLS を使用します。 SSL：セキュリティ通信用のプロトコルに SSL を使用します。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「TLS」または「SSL」の場合 指定した値が適用されます。 • タグがあって値が「TLS」または「SSL」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「TLS」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-implicit-mode>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，FTPS の Implicit モードを使用するかどうかを指定するタグです。指定値は次のどちらかです。

タグ名			出現回数※1	説明
		<ftps-implicit-mode>	0 または 1 回	true：Implicit モードを使用します。 false：Explicit モードを使用します。※2 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。
		<ftps-server-authentication>	0 または 1 回	FTP サーバとの接続に FTPS を使用する場合に，サーバ認証をするかどうかを指定するタグです。指定値は次のどちらかです。 true：サーバ認証をします。 false：サーバ認証をしません。 また，タグの有無に応じて次のように動作します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。 この設定は，FTP サーバとの接続に FTPS を使用する場合だけ有効となります。なお，「true」を指定する場合，キーストアに関する設定が必要です。キーストアの設定方法については，「付録 H.1　FTPS を使用したセキュアな接続(FTP アダプタ)」を参照してください。
	<ftp-user>※3		0 または 1 回	接続する FTP サーバのログインユーザ名を示すタグです。USER コマンドの引数として使用します。ログインユーザ名は 80 文字以内で指定してください。このタグに指定するログインユーザには，ファイル操作を実行できる権限が与えられている必要があります。 FTP アダプタ実行環境プロパティファイルに重複して設定されている場合，要求電文の値が有効になります。 また，タグの有無に応じて次のように動作します。 • タグがあって正常な値がある場合 「USER　"値"」が発行されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値がない場合，値に不正な文字が含まれている場合，または文字列長が不正な場合 システム例外を返します。

タグ名		出現回数※1	説明
	<ftp-user>※3	0 または 1 回	• タグがない場合 FTP アダプタ実行環境プロパティファイルの値が有効になります。FTP アダプタ実行環境プロパティファイルに指定がない場合，システム例外を返します。
	<ftp-acct>※3	0 または 1 回	接続する FTP サーバの課金情報を示すタグです。FTP アダプタアカウント定義ファイルに重複して設定されている場合，要求電文の値が有効になります。 課金情報を必要とするユーザ 条件に応じて次の値が利用されます。 • タグがあって正常な値がある場合 タグの内容が ACCT コマンドに利用されます。値に改行が含まれている場合，システム例外を返します。 • タグがあって値に不正な文字が含まれている場合 システム例外を返します。 • タグがあって値がない場合 「ACCT　""」が発行されます。 • タグがない場合 FTP アダプタアカウント定義ファイルで指定されている ACCT の値が利用されます。FTP アダプタアカウント定義ファイルに ACCT がない場合は「ACCT　""」が発行されます。 課金情報を必要としないユーザ 指定は無視されます。
	<rename-from-file-path>	1 回	リネーム対象のリモートファイルのパス名を示すタグです。 • タグがあって値がある場合 指定した値がリネーム対象のリモートファイルのパスとして利用されます。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
	<rename-to-file-path>	1 回	<rename-from-file-path>要素の値に指定したファイルの新しいパス名を示すタグです。 • タグがあって値がある場合 指定した値が，<rename-from-file-path>要素の値に指定したファイルの，新しいパス名として利用されます。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
	<ftp-commands-before>※4	0 または 1 回	ファイルのリネームコマンドの実行前に実行する FTP コマンドと，その引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します（例：MKD transdir;CWD transdir）。 • タグがあって値がある場合 ファイルのリネームコマンド実行前に実行するコマンドとして利用されます。 • タグがない，またはタグがあって値がない場合 ファイルのリネームコマンド実行前に実行するコマンドはありません。

タグ名		出現回数※1	説明
	<ftp-commands-after>※4	0 または 1 回	ファイルのリネームコマンドの実行後に実行する FTP コマンドと，その引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します（例：RNFR oldfile.txt;RNTO newfile.txt）。 • タグがあって値がある場合 ファイルのリネームコマンド実行後に実行するコマンドとして利用されます。 • タグがない，またはタグがあって値がない場合 ファイルのリネームコマンド実行後に実行するコマンドはありません。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

接続先 FTP サーバが FTPS の Implicit モードで動作している場合に false を指定すると，FTP サーバに接続したあとタイムアウトが発生するまで待ち状態となることがあります。

なお，タイムアウトが発生するまでの時間の設定方法は 2 とおりあり，値が小さい方が適用されます。タイムアウト値の設定方法の種類とタイムアウト発生時の動作を次に示します。

タイムアウト値の設定方法の種類	タイムアウト発生時の動作
FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.connect.timeout プロパティで指定した値	FTP アダプタ実行環境プロパティファイルの ftpadp.control-con.retry.count プロパティで指定した回数のリトライを行ったあと，KDEK30407-E のエラーが発生します。
接続先 FTP サーバ側で設定されたタイムアウト値	KDEK30428-E のエラーが発生します。

注※3

使用できる文字を次に示します。

半角英数字，半角空白，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

注※4

<ftp-commands-before>タグおよび<ftp-commands-after>タグに指定されたコマンドは，そのまま FTP サーバに送られ，FTP アダプタはコマンドのオプションの妥当性やコマンドの実行による影響について関知しません。

そのため，<ftp-commands-before>タグおよび<ftp-commands-after>タグに OPTS UTF {ON|OFF}コマンドを指定した場合は，FTP サーバとの通信に適用される文字セットを次のどちらかで指定しておく必要があります。

・要求電文の<file-name-charset>タグ

・FTP アダプタ実行環境プロパティファイルの ftpadp.file-name.charset プロパティ

(b) FTP アダプタの応答電文フォーマット

●PUT オペレーション

FTP アダプタが呼び出し元のビジネスプロセスに返す PUT オペレーションの応答電文のフォーマットを次に示します。ファイル名は「ftpadp_put_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/put_response」です。

表 3-72　応答電文フォーマット（PUT オペレーション）

タグ名			出現回数	説明
<response>			1 回	−
	<local-folder>		1 回	ローカル側フォルダを示すタグです。 作業フォルダを利用した場合，属性に「common="false"」が設定されます。値には空白が設定されます。 共通フォルダを利用した場合，属性に「common="true"」が設定されます。値には，要求電文の<local-folder>タグで指定された共通フォルダ定義名が設定されます。
	<local-file-name>		1 回	ローカル側ファイル名を示すタグです。 要求電文の<local-file-name>で指定した値が設定されます。
	<command-info>		1 回以上	−
		<command>	1 回	FTP アダプタが実行したコマンドを示すタグです。※1 実行したすべての FTP コマンドが出力されます。
		<command-message>	1 回以上	FTP アダプタが実行したコマンドの表示メッセージを示すタグです。※2 FTP コマンドのメッセージは FTP 応答コードとメッセージから構成されます。 メッセージが複数行ある場合，これらが同一の応答コードのときは，1 つのタグが生成されます。複数の種類の FTP 応答コードがあるときは，このタグも複数生成されます。

（凡例）

−：該当する項目はありません。

注※1

FTP アダプタが実行するコマンドの種類については，マニュアル「サービスプラットフォーム 解説」の「8.6.2　FTP コマンドの実行」を参照してください。

FTP アダプタが実行したコマンドの出力フォーマットと出力例については，「表 3-80」を参照してください。

注※2

FTP コマンドのメッセージフォーマットと出力例については，「表 3-81」を参照してください。

●MPUT オペレーション

FTP アダプタが呼び出し元のビジネスプロセスに返す MPUT オペレーションの応答電文のフォーマットを次に示します。ファイル名は「ftpadp_mput_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/mput_response」です。

表 3-73　応答電文フォーマット（MPUT オペレーション）

タグ名			出現回数	説明
<response>			1 回	−
	<path>		1 回以上	−
		<local-folder>	1 回	ローカル側フォルダを示すタグです。 作業フォルダを利用した場合，属性に「common="false"」が設定されます。値には空白が設定されます。 共通フォルダを利用した場合，属性に「common="true"」が設定されます。値には，要求電文の<local-folder>タグで指定された共通フォルダ定義名が設定されます。
		<local-file-name>	1 回	ローカル側ファイル名を示すタグです。 要求電文の<local-file-name>で指定した値が設定されます。
	<command-info>		1 回以上	−
		<command>	1 回	FTP アダプタが実行したコマンドを示すタグです。 実行したすべての FTP コマンドが出力されます。
		<command-message>	1 回以上	FTP アダプタが実行したコマンドの応答メッセージを示すタグです。 FTP コマンドのメッセージは FTP 応答コードとメッセージから構成されます。 メッセージが複数行ある場合，これらが同一の応答コードのときは，1 つのタグが生成されます。複数の種類の FTP 応答コードがあるときは，このタグも複数生成されます。

（凡例）

−：該当する項目はありません。

●GET オペレーション

FTP アダプタが呼び出し元のビジネスプロセスに返す GET オペレーションの応答電文のフォーマットを次に示します。ファイル名は「ftpadp_get_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/get_response」です。

表 3-74　応答電文フォーマット（GET オペレーション）

タグ名		出現回数	説明
<response>		1 回	−
	<local-folder>	1 回	ローカル側フォルダを示すタグです。 作業フォルダを利用した場合，属性に「common="false"」が設定されます。値には空白が設定されます。 共通フォルダを利用した場合，属性に「common="true"」が設定されます。値には，要求電文の<local-folder>タグで指定された共通フォルダ定義名が設定されます。
	<local-file-name>	1 回	ローカル側ファイル名を示すタグです。

タグ名			出現回数	説明
	<local-file-name>		1 回	作業フォルダを利用する場合 作業フォルダの下に格納したファイル名が設定されます。ファイル名は，「csc_<内部生成された文字>」の形式となります。 共通フォルダを利用する場合 要求電文の<local-file-name>で指定した値が設定されます。
	<file-size>		1 回	送受信したリモートファイルのサイズを示すタグです。 FTP サーバから取得したローカルファイルのサイズが設定されます。
	<command-info>		1 回以上	－
		<command>	1 回	FTP アダプタが実行した FTP コマンドを示すタグです。※1 実行したすべての FTP コマンドが出力されます。
		<command-message>	1 回以上	FTP アダプタが実行したコマンドの表示メッセージを示すタグです。※2 FTP コマンドのメッセージは FTP 応答コードとメッセージから構成されます。 メッセージが複数行ある場合，これらが同一の応答コードのときは，1 つのタグが生成されます。複数の種類の FTP 応答コードがあるときは，このタグも複数生成されます。

（凡例）

－：該当する項目はありません。

注※1

FTP アダプタが実行するコマンドの種類については，マニュアル「サービスプラットフォーム 解説」の「8.6.2　FTP コマンドの実行」を参照してください。

FTP アダプタが実行したコマンドの出力フォーマットと出力例については，「表 3-80」を参照してください。

注※2

FTP コマンドのメッセージフォーマットと出力例については，「表 3-81」を参照してください。

●MGET オペレーション

FTP アダプタが呼び出し元のビジネスプロセスに返す MGET オペレーションの応答電文のフォーマットを次に示します。ファイル名は「ftpadp_mget_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/mget_response」です。

表 3-75　応答電文フォーマット（MGET オペレーション）

タグ名			出現回数	説明
<response>			1 回	－
	<path>		1 回以上	－
		<local-folder>	1 回	ローカル側フォルダを示すタグです。 作業フォルダを利用した場合，属性に「common="false"」が設定されます。値には空白が設定されます。

タグ名			出現回数	説明
		<local-folder>	1 回	共通フォルダを利用した場合，属性に「common="true"」が設定されます。値には，要求電文の<local-folder>タグで指定された共通フォルダ定義名が設定されます。
		<local-file-name>	1 回	ローカル側ファイル名を示すタグです。 要求電文の<local-file-name>で指定した値が設定されます。
		<file-size>	1 回	受信したリモートファイルのサイズを示すタグです。
		<remote-path>	1 回	リモートパス名を示すタグです。 要求電文の<remote-path>で指定した値が設定されます。
	<command-info>		1 回以上	－
		<command>	1 回	FTP アダプタが実行した FTP コマンドを示すタグです。 実行したすべての FTP コマンドが出力されます。
		<command-message>	1 回以上	FTP アダプタが実行したコマンドの応答メッセージを示すタグです。 FTP コマンドのメッセージは FTP 応答コードとメッセージから構成されます。 メッセージが複数行ある場合，これらが同一の応答コードのときは，1 つのタグが生成されます。複数の種類の FTP 応答コードがあるときは，このタグも複数生成されます。

（凡例）

－：該当する項目はありません。

●GETINFO オペレーション

FTP アダプタが呼び出し元のビジネスプロセスに返す GETINFO オペレーションの応答電文のフォーマットを次に示します。ファイル名は「ftpadp_getinfo_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/getinfo_response」です。

表 3-76　応答電文フォーマット（GETINFO オペレーション）

タグ名		出現回数	説明
<response>		1 回	－
	<local-folder>	0 または 1 回	ローカル側フォルダを示すタグです。共通フォルダに出力する場合に出現します。作業フォルダまたは応答電文に出力する場合は出現しません。 共通フォルダを利用した場合 属性に「common="true"」が設定されます。値には，要求電文の<local-folder>タグで指定された共通フォルダ定義名が設定されます。
	<local-file-name>	0 または 1 回	ローカル側ファイル名を示すタグです。作業フォルダまたは共通フォルダに出力する場合に出現します。応答電文に出力する場合は出現しません。

タグ名			出現回数	説明
	<local-file-name>		0 または 1 回	作業フォルダを利用した場合 作業フォルダの下に格納したファイル名が設定されます。ファイル名は，「csc_<内部生成された文字>」の形式となります。 取得した情報は，「UTF-16」の文字セットで出力されます。また，改行コードは CRLF で出力されます。 共通フォルダを利用した場合 要求電文の<local-file-name>で指定した値が設定されます。 このファイル中には取得した情報が UTF-16 の文字セットで書き込まれます。また，改行コードは CRLF となります。
	<reply-code>		1 回	LIST コマンドまたは NLST コマンドのリプライコードを示すタグです。 LIST コマンドまたは NLST コマンドを実行したときに，FTP サーバから最後に送られてくるリプライコードとして，次のどれかが設定されます。 • 200～299 処理が正常に完了したことを示すコードです。 • 450 または 550 ファイル操作ができなかったことを示すコードです。 なお，LIST コマンドまたは NLST コマンドで指定したパスが存在しない場合，返されるリプライコードは接続先の FTP サーバによって異なります。
	<reply-message>		1 回	LIST コマンドまたは NLST コマンドのリプライメッセージを示すタグです。 LIST コマンドまたは NLST コマンドを実行したときに，FTP サーバから最後に送られてきたリプライメッセージが設定されます。
	<command-info>		1 回以上	－
		<command>	1 回	FTP アダプタが実行した FTP コマンドを示すタグです。※1 実行したすべての FTP コマンドが出力されます。
		<command-message>	1 回以上	FTP アダプタが実行したコマンドの表示メッセージを示すタグです。※2 FTP コマンドのメッセージは FTP 応答コードとメッセージから構成されます。 メッセージが複数行ある場合，これらが同一の応答コードのときは，1 つのタグが生成されます。複数の種類の FTP 応答コードがあるときは，このタグも複数生成されます。
	<list-info>		0 または 1 回	応答電文に出力する場合に出現します。※3
		<list>	0 回以上	応答電文に出力する場合に出現します。※3

（凡例）

－：該当する項目はありません。

注※1

FTP アダプタが実行するコマンドの種類については，マニュアル「サービスプラットフォーム 解説」の「8.6.2　FTP コマンドの実行」を参照してください。

FTP アダプタが実行したコマンドの出力フォーマットと出力例については，「表 3-80」を参照してください。

注※2

FTP コマンドのメッセージフォーマットと出力例については，「表 3-81」を参照してください。

注※3

GETINFO オペレーションの定義方法については，「(c) GETINFO オペレーションを使用する場合の要求電文および応答電文」を参照してください。

●LOGIN オペレーション

FTP アダプタが呼び出し元のビジネスプロセスに返す LOGIN オペレーションの応答電文のフォーマットを次に示します。ファイル名は「ftpadp_login_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/login_response」です。

表 3-77　応答電文フォーマット（LOGIN オペレーション）

タグ名			出現回数	説明
<response>			1 回	−
	<command-info>		0 回以上	−
		<command>	1 回	FTP アダプタが実行した FTP コマンドを示すタグです。 実行したすべての FTP コマンドが出力されます。
		<command-message>	1 回以上	FTP アダプタが実行したコマンドの応答メッセージを示すタグです。 FTP コマンドのメッセージは FTP 応答コードとメッセージから構成されます。 メッセージが複数行ある場合，これらが同一の応答コードのときは，1 つのタグが生成されます。複数の種類の FTP 応答コードがあるときは，このタグも複数生成されます。

（凡例）

−：該当する項目はありません。

●LOGOUT オペレーション

FTP アダプタが呼び出し元のビジネスプロセスに返す LOGOUT オペレーションの応答電文のフォーマットを次に示します。ファイル名は「ftpadp_logout_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/logout_response」です。

表 3-78　応答電文フォーマット（LOGOUT オペレーション）

タグ名			出現回数	説明
<response>			1 回	−
	<command-info>		0 回以上	−
		<command>	1 回	FTP アダプタが実行した FTP コマンドを示すタグです。 実行したすべての FTP コマンドが出力されます。
		<command-message>	1 回以上	FTP アダプタが実行したコマンドの応答メッセージを示すタグです。 FTP コマンドのメッセージは FTP 応答コードとメッセージから構成されます。

タグ名			出現回数	説明
		<command-message>	1 回以上	メッセージが複数行ある場合，これらが同一の応答コードのときは，1 つのタグが生成されます。複数の種類の FTP 応答コードがあるときは，このタグも複数生成されます。

（凡例）

－：該当する項目はありません。

●RENAME オペレーション

FTP アダプタが呼び出し元のビジネスプロセスに返す RENAME オペレーションの応答電文のフォーマットを次に示します。ファイル名は「ftpadp_rename_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csaftp/rename_response」です。

表 3-79　応答電文フォーマット（RENAME オペレーション）

タグ名			出現回数	説明
<response>			1 回	－
	<command-info>		1 回以上	－
		<command>	1 回	FTP アダプタが実行した FTP コマンドを示すタグです。※1 実行したすべての FTP コマンドが出力されます。
		<command-message>	1 回以上	FTP アダプタが実行したコマンドの表示メッセージを示すタグです。※2 FTP コマンドのメッセージは FTP 応答コードとメッセージから構成されます。 メッセージが複数行ある場合，これらが同一の応答コードのときは，1 つのタグが生成されます。複数の種類の FTP 応答コードがあるときは，このタグも複数生成されます。

（凡例）

－：該当する項目はありません。

注※1

FTP アダプタが実行するコマンドの種類については，マニュアル「サービスプラットフォーム 解説」の「8.6.2　FTP コマンドの実行」を参照してください。

FTP アダプタが実行したコマンドの出力フォーマットと出力例については，「表 3-80」を参照してください。

注※2

FTP コマンドのメッセージフォーマットと出力例については，「表 3-81」を参照してください。

●FTP アダプタが実行したコマンドの出力フォーマットと出力例

応答電文の<command>タグに出力される FTP コマンドの出力フォーマットと出力例を次の表に示します。

表 3-80　FTP アダプタが実行したコマンドの出力フォーマットと出力例

項番	FTP コマンド	出力フォーマット	出力例	備考
1	（接続処理）	Connected to <xxxxx>.	Connected to 10.208.180.254.	<xxxxx>は FTP サーバの IP アドレスまたはホスト名です。
2	USER	USER <ユーザ名>	USER user1	−
3	PASS	PASS ********	PASS ********	パスワードは 8 個の*で表示されます。
4	ACCT	ACCT <課金情報>	ACCT acct1	−
5	TYPE	TYPE <形式オプション>	TYPE A	<形式オプション>には「A」または「I」のどちらかが表示されます。
6	MODE	MODE <転送モードオプション>	MODE S	<転送モードオプション>には「S」または「C」のどちらかが表示されます。
7	PASV	PASV	PASV	−
8	PORT	PORT <aaa>,<bbb>,<ccc>,<ddd>,<mmm>,<nnn>	PORT 10,208,182,27,14,160	<aaa>,<bbb>,<ccc>,<ddd>はローカル側の IP アドレスです。 <mmm>,<nnn>はポート番号です。ローカル側のデータコネクションポート番号は 256*mmm+nnn で表現されます。
9	SITE	SITE FSIZE [<ファイルサイズ>]	SITE FSIZE 4096	GET オペレーションの場合，<ファイルサイズ>は出力されません。 また，PUT オペレーションの場合，<ファイルサイズ>に転送元ファイルの転送路上のファイルサイズが出力されます。
10	転送前コマンド，リスト前コマンド，またはリネーム前コマンド	<コマンド名> [<コマンド引数>]	CWD ./workdir	−
11	STOR	STOR <パス名>	STOR work/fileA.txt	−
12	APPE	APPE <パス名>	APPE work/fileA.txt	−
13	RETR	RETR <パス名>	RETR fileA.txt	−
14	LIST	LIST [<オプション>] [<パス名>]	LIST -l work	−
15	NLST	NLST [<オプション>] [<パス名>]	NLST -l work	−
16	転送後コマンド，リスト後	<コマンド名> [<コマンド引数>]	RNFR fileA.txt RNTO fileB.txt	−

項番	FTP コマンド	出力フォーマット	出力例	備考
16	コマンド，またはリネーム後コマンド	<コマンド名> [<コマンド引数>]	RNFR fileA.txt RNTO fileB.txt	－
17	RNFR	RNFR <リネーム対象のリモートファイルのパス名>	RNFR fileA.txt	－
18	RNTO	RNTO <ファイルの新しいパス名>	RNTO fileB.txt	－
19	QUIT	QUIT	QUIT	－

（凡例）

－：該当する説明はありません。

●FTP コマンドのメッセージフォーマットと出力例

応答電文の<command-message>タグに出力される FTP コマンドのメッセージフォーマット，および出力例を次の表に示します。

表 3-81　FTP コマンドのメッセージフォーマットと出力例

条件	メッセージフォーマット	出力例
単一応答コード	<FTP 応答コード> <メッセージ>	226 File receive OK.
	<FTP 応答コード>-<メッセージ 1> [<メッセージ 2>] ： <FTP 応答コード> <メッセージ n※>	214-The following commands are recognized. ACCT ALLO APPE CDUP ... MDTM MKD NLST NOOP OPTS PASS ... RMD RNFR 214 Help OK.
複数応答コード	<FTP 応答コード 1> <メッセージ 1> <FTP 応答コード 2> <メッセージ 2>	150 File status okay; about to open data connection. 226 Transfer complete.

注

応答コードおよびメッセージの文字列の内容については，使用している FTP サーバのドキュメントを参照してください。

注※

n は自然数です。

(c)　GETINFO オペレーションを使用する場合の要求電文および応答電文

LIST コマンドまたは NLIST コマンドの結果を応答電文，作業フォルダ，共通フォルダに出力する場合に要求電文に指定が必要なタグ，および応答電文に出現するタグを次の表に示します。

各タグの設定内容や表に記載がないタグについては，「(a) FTP アダプタの要求電文フォーマット」および「(b) FTP アダプタの応答電文フォーマット」を参照してください。

表 3-82　出力先と要求電文の関係

出力先	要求電文で指定するタグ		
	<request-id>	<local-folder>	<local-file-name>
応答電文	−	−	−
作業フォルダ	○	○	−
共通フォルダ	−	○	○

（凡例）

○：指定します。

−：指定しません。

表 3-83　出力先と応答電文の関係

出力先	応答電文で出現するタグ			
	<local-folder>	<local-file-name>	<list-info>	<list>
応答電文	−	−	○	○
作業フォルダ	−	○	−	−
共通フォルダ	○	○	−	−

（凡例）

○：出現します。

−：出現しません。

(2)　FTP アダプタの定義ファイルの作成

作成する定義ファイルの種類を次に示します。

- FTP アダプタコマンド定義ファイル

  FTP アダプタで使用するユーザ管理コマンドのメッセージログの出力先，出力レベルなどを定義するためのファイルです。

- FTP アダプタアカウント定義ファイル

  FTP 許可ユーザ，課金情報などのユーザアカウント情報を定義するためのファイルです。このファイルは FTP アダプタ実行環境プロパティファイルから参照されます。

- FTP コマンド許可リスト定義ファイル（FTP アダプタ用）

  転送前および転送後に実行する FTP コマンドの実行許可を定義するためのファイルです。このファイルは FTP 受付コンフィグファイルおよび FTP アダプタ実行環境プロパティファイルから参照されます。

- FTP アダプタ実行環境プロパティファイル

  FTP アダプタごとの構成情報を定義するためのファイルです。

各定義ファイルの作成手順を次に示します。定義ファイルは，FTP アダプタが提供しているテンプレートファイルを利用して作成できます。

(a) FTP アダプタコマンド定義ファイルの作成手順

1. テンプレートファイル（<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¥custom-adapter¥FTP¥config¥templates¥adpftpcommand.properties）をコピーして，次のディレクトリに格納します。

   <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>¥CSC¥custom-adapter¥FTP¥config

2. 定義内容を編集して保存します。

   FTP アダプタコマンド定義ファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.8.2　FTP アダプタコマンド定義ファイル」を参照してください。

(b) FTP アダプタアカウント定義ファイルの作成手順

1. csaftpuseradd コマンドを実行して FTP アダプタアカウント定義ファイルを作成します。

   作成した FTP アダプタアカウント定義ファイルのパスは，FTP アダプタ実行環境プロパティファイルの ftpadp.account-inf-filepath キーで指定する必要があります。csaftpuseradd コマンドの詳細については，マニュアル「サービスプラットフォーム リファレンス」の「csaftpuseradd（FTP アダプタのユーザの登録・更新）」を参照してください。

(c) FTP コマンド許可リスト定義ファイル（FTP アダプタ用）の作成手順

1. テンプレートファイル（<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¥custom-adapter¥FTP¥config¥templates¥cscthrough.properties）をコピーして，任意のディレクトリに格納します。

2. コピーしたテンプレートファイルのファイル名を任意の名称に変更します。

3. 定義内容を編集して保存します。

   FTP コマンド許可リスト定義ファイル（FTP アダプタ用）で編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.8.6　FTP コマンド許可リスト定義ファイル」を参照してください。

(d) FTP アダプタ実行環境プロパティファイルの作成手順

1. テンプレートファイル（<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¥custom-adapter¥FTP¥config¥templates¥adpftp.properties）をコピーして，次のディレクトリに格納します。

   全 HCSC サーバ共通の定義の場合

   <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>¥CSC¥custom-adapter¥FTP¥config

   特定の HCSC サーバ用の定義の場合

   サブディレクトリ<HCSC サーバ名>を作成して格納してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

<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¥custom-adapter¥FTP¥config¥<HCSC サーバ名>

2. コピーしたテンプレートファイルのファイル名を「<サービス ID※>.properties」に変更します。

3. 定義内容を編集して保存します。

FTP アダプタ実行環境プロパティファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.8.3　FTP アダプタ実行環境プロパティファイル」を参照してください。

注※

サービス ID は，FTP アダプタを新規に追加するときにサービスアダプタ定義画面で指定する任意の文字列です。

(e)　FTP アダプタ実行環境共通プロパティファイルの作成手順

1. テンプレートファイル（<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¥custom-adapter¥FTP¥config¥templates¥adpftp.properties）をコピーして，次のディレクトリに格納します。

全 HCSC サーバ共通の定義の場合

common ディレクトリはユーザが作成してください。

<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>¥CSC¥custom-adapter¥FTP¥config¥common

特定の HCSC サーバ用の定義の場合

サブディレクトリ<HCSC サーバ名>と「common」はユーザが作成してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

<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¥custom-adapter¥FTP¥config¥<HCSC サーバ名>¥common

FTP アダプタ実行環境共通プロパティファイルのファイル名は変更しないでください。

2. 定義内容を編集して保存します。

FTP アダプタ実行環境共通プロパティファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.8.4　FTP アダプタ実行環境共通プロパティファイル」を参照してください。

(3)　データ変換の定義

データ変換定義画面を利用して，変換元の電文フォーマット定義ファイルおよび変換先の電文フォーマット定義ファイルのデータ変換を定義します。

データ変換の定義方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.　データ変換の定義」を参照してください。

(4)　サービスアダプタ定義画面での操作

FTP アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。
   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. 定義情報を設定します。
   サービスアダプタ定義画面（基本）で設定が必要な項目については，「3.3.19(10)　FTP アダプタの場合」およびマニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

3. [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画面（詳細）が表示されます。

4. 定義情報を設定します。
   サービスアダプタ定義画面（詳細）で設定が必要な項目については，「3.3.19(10)　FTP アダプタの場合」およびマニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

3.3.11　ファイル操作アダプタを定義する

ファイル操作アダプタの定義方法について説明します。なお，ファイル変換オペレーションを使用する場合の定義例については，「付録 E　ファイル操作アダプタの定義例」を参照してください。

(1)　電文フォーマットの作成

ファイル操作アダプタの電文フォーマットの定義ファイルはサービスプラットフォームが提供するスキーマを使用します。そのため，電文フォーマットの定義ファイルの作成は不要です。

ファイル操作アダプタで使用する電文フォーマットの内容について説明します。

ファイルの格納場所は「<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¥schema¥adpfop」です。

(a)　ファイル操作アダプタの要求電文フォーマット

●共通仕様

ファイル操作アダプタの要求電文フォーマットの指定についての共通仕様を，次に説明します。

ファイル操作アダプタの要求電文フォーマットで指定する，ファイルやディレクトリのパス（ファイル名だけの指定も含む）には，次の制限があります。

- Windows の場合，次の制限があります。

・大文字と小文字は区別されません。
・UNC 形式のディレクトリおよびファイル名は指定できません。
・NTFS のストリーム名は指定できません。

- 絶対パスは指定できません。
- 末尾が「.」または「..」のパスは指定できません。
- 作業フォルダを指定する場合に，相対パスは指定できません。
- 作業フォルダを指定する場合に，ファイル名の先頭にだけスラッシュ（/）を使用できます。また，ファイル名の先頭に設定されたスラッシュ（/）は，無視されます。
- 共通フォルダを指定する場合に，共通フォルダの子孫ではないパスは指定できません。

作業フォルダを指定してファイルを出力する場合，java.io.File の createTempFile()を使用して作業フォルダ内で一意のファイルが作成されるため，指定は無視されます。

パスを指定する場合の注意事項について，次に説明します。

- デバイス名を含むパスは指定しないでください。
- シンボリックリンクを含むパスは指定しないでください。

ファイル操作アダプタの要求電文フォーマットを，オペレーションごとに説明します。

●ファイル変換オペレーション

ビジネスプロセスからファイル操作アダプタに渡す要求電文フォーマットを次に示します。ファイル名は「adpfop_transform_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/transform_request」です。

表 3-84　要求電文フォーマット（ファイル変換オペレーション）

タグ名		出現回数※	説明
<request>		1 回	−
	<request-id>	0 または 1 回	受付で作成したリクエスト ID です。作業フォルダのパス生成に使用します。入力フォルダ，および出力フォルダに作業フォルダを指定しない場合は，省略できます。
	<input-folder-name>	1 回	変換するファイルが格納されているディレクトリを指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を，値に共通フォルダ定義名を指定します。
	<input-file-name>	1 回	変換するファイルを指定します。
	<output-folder-name>	1 回	変換後のファイルを出力するディレクトリを指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を，値に空文字を指定します。 • 共通フォルダを指定する場合

タグ名		出現回数※	説明
	<output-folder-name>	1 回	タグの属性に common="true"を，値に共通フォルダ定義名を指定します。
	<output-file-name>	0 または 1 回	変換後のファイルを指定します。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル複製オペレーション

ビジネスプロセスからファイル操作アダプタに渡す要求電文フォーマットを次に示します。ファイル名は「adpfop_copy_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/copy_request」です。

表 3-85　要求電文フォーマット（ファイル複製オペレーション）

タグ名		出現回数※	説明
<request>		1 回	－
	<request-id>	0 または 1 回	受付で作成したリクエスト ID です。作業フォルダのパス生成に使用します。入力フォルダ，および出力フォルダに作業フォルダを指定しない場合は，省略できます。
	<input-folder-name>	1 回	複製するファイルが格納されているディレクトリを指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を，値に共通フォルダ定義名を指定します。
	<input-file-name>	1 回	複製するファイルを指定します。
	<output-folder-name>	1 回	複製後のファイルを出力するディレクトリを指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を，値に共通フォルダ定義名を指定します。
	<output-file-name>	0 または 1 回	複製後のファイルを指定します。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル移動オペレーション

ビジネスプロセスからファイル操作アダプタに渡す要求電文フォーマットを次に示します。ファイル名は「adpfop_move_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/move_request」です。

表 3-86　要求電文フォーマット（ファイル移動オペレーション）

タグ名		出現回数※	説明
<request>		1 回	−
	<request-id>	0 または 1 回	受付で作成したリクエスト ID です。作業フォルダのパス生成に使用します。入力フォルダ，および出力フォルダに作業フォルダを指定しない場合は，省略できます。
	<input-folder-name>	1 回	移動するファイルが格納されているディレクトリを指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を指定し，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を指定し，値に共通フォルダ定義名を指定します。
	<input-file-name>	1 回	移動するファイルを指定します。
	<output-folder-name>	1 回	移動先のディレクトリを指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を指定し，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を指定し，値に共通フォルダ定義名を指定します。
	<output-file-name>	0 または 1 回	移動先のファイルを指定します。 移動先のディレクトリが共通フォルダで当該要素が省略された場合，移動元ファイル名を仮定します。

（凡例）

−：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル・フォルダ削除オペレーション

ビジネスプロセスからファイル操作アダプタに渡す要求電文フォーマットを次に示します。ファイル名は「adpfop_delete_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/delete_request」です。

表 3-87　要求電文フォーマット（ファイル・フォルダ削除オペレーション）

タグ名	出現回数※1	説明
<request>	1 回	−

タグ名		出現回数※1	説明
	<request-id>	0 または 1 回	受付で作成したリクエスト ID です。作業フォルダのパス生成に使用します。入力フォルダ，および出力フォルダに作業フォルダを指定しない場合は，省略できます。
	<folder-name>	1 回	削除するファイルが格納されているディレクトリ，または削除するフォルダを指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を，値に共通フォルダ定義名を指定します。
	<file-name>	0 または 1 回	削除するファイルを指定します。※2
	<sub-folder-name>	0 または 1 回	削除するサブフォルダを指定します。※2

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

file-name タグと sub-folder-name タグについて，次に説明します。

・共通フォルダを指定する場合は，どちらか一方を指定する必要があります。また，両方を指定することはできません（どちらか一方のタグを省略する必要があります）。

・作業フォルダを指定する場合は，タグの指定は無視されます。

●ファイル圧縮オペレーション

ビジネスプロセスからファイル操作アダプタに渡す要求電文フォーマットを次に示します。ファイル名は「adpfop_compress_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/compress_request」です。

表 3-88　要求電文フォーマット（ファイル圧縮オペレーション）

タグ名		出現回数※1	説明
<request>		1 回	－
	<request-id>	0 または 1 回	受付で作成したリクエスト ID です。作業フォルダのパス生成に使用します。入力フォルダ，および出力フォルダに作業フォルダを指定しない場合は，省略できます。
	<input-folder-name>	1 回	圧縮するファイルが格納されているディレクトリを指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を，値に共通フォルダ定義名を指定します。

タグ名			出現回数※1	説明
<input-files>			1 回	−
	<input-file>		1 から 999 回	−
		<input-file-name>	1 回	圧縮するファイルを指定します。※2
		<entry-name>	1 回	同じ input-file タグ内の input-file-name タグで指定したファイルの圧縮後のエントリ名を指定します。空文字列を指定した場合，同じ input-file タグ内の input-file-name タグの値が指定されます。この場合，エントリ名は変更されません。 Windows の場合，次の制限があります。 • 大文字と小文字は区別されます。※3 • UNC 形式のエントリ名は指定できません。 • NTFS のストリーム名は指定できません。 • デバイス名を含むエントリ名は指定しないでください。
<output-folder-name>			1 回	圧縮後のファイルを出力するディレクトリを指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を，値に共通フォルダ定義名を指定します。
<output-file-name>			0 または 1 回	圧縮後のファイルを指定します。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

input-file-name タグでは，大文字と小文字が区別されないことに注意してファイル名を指定してください。同一のファイル名を複数指定した場合は動作を保証しません。

注※3

Windows で圧縮ファイルを伸張する場合，伸張後のファイル名は大文字と小文字が区別されないため，ファイル名が同一になるおそれがあります。この点に注意してエントリ名を指定してください。

●ファイル伸張オペレーション

ビジネスプロセスからファイル操作アダプタに渡す要求電文フォーマットを次に示します。ファイル名は「adpfop_extract_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/extract_request」です。

表 3-89　要求電文フォーマット（ファイル伸張オペレーション）

タグ名		出現回数※	説明
<request>		1 回	－
	<request-id>	0 または 1 回	受付で作成したリクエスト ID です。作業フォルダのパス生成に使用します。入力フォルダ，および出力フォルダに作業フォルダを指定しない場合は，省略できます。
	<input-folder-name>	1 回	伸張するファイルが格納されているディレクトリを指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を，値に共通フォルダ定義名を指定します。
	<input-file-name>	1 回	伸張するファイルを指定します。
	<output-folder-name>	1 回	伸張後のファイルを出力するディレクトリを指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を，値に共通フォルダ定義名を指定します。
	<output-sub-folder-name>	0 または 1 回	伸張後のファイルを出力するサブフォルダを指定します。指定を省略した場合，共通フォルダの直下にファイルを出力します。
	<output-file-prefix-name>	0 または 1 回	伸張後のファイル名の接頭辞を指定します。 • 変換後のファイルの出力先が共通フォルダの場合（output-folder-name タグの属性に common="true"を指定した場合） このタグの指定を使用してファイル名が自動で付与されます。 命名規則は<要求電文の output-file-prefix-name タグの要素> ＋ <_> ＋ <001 から始まる連番（3 桁）>です。 • 変換後のファイルの出力先が作業フォルダの場合（output-folder-name タグの属性に common="false"を指定した場合） java.io.File の createTempFile()を使用して作業フォルダ内で一意のファイルが作成されるため，指定は無視されます。 Windows の場合，次の制限があります。 • 大文字と小文字は区別されません。 • UNC 形式のファイル名は指定できません。 • NTFS のストリーム名は指定できません。 • デバイス名を含むファイル名は指定しないでください。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル一覧取得オペレーション

ビジネスプロセスからファイル操作アダプタに渡す要求電文フォーマットを次に示します。ファイル名は「adpfop_get_list_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/get_list_request」です。

表 3-90　要求電文フォーマット（ファイル一覧取得オペレーション）

タグ名	出現回数※1	説明
<request>	1 回	−
<input-folder-name>	1 回	ファイル一覧を取得する共通フォルダ名を指定します。
<input-file-name>	0 または 1 回	ファイル一覧を取得するファイルもしくはサブフォルダのパスを指定します。 • パスは共通フォルダからの相対パスで指定します。 • 当該要素が省略された場合，共通フォルダ自体を対象とします。
<options>	0 または 1 回	オプションを指定する場合に使用します。 当該要素が省略された場合はオプション項目をすべて省略時の動作とします。
<list-option>	0 または 1 回	出力するファイル一覧の出力形式を指定します。※2
<date-time-pattern>	0 または 1 回	最終更新日時の出力形式を指定します。 • 当該要素が省略もしくは空白の場合は，「yyyy-MM-dd'T'HH:mm:ss」が仮定されます。 • 指定できるパターン文字列は java.time.format.DateTimeFormatter#ofPattern(java.lang.String)の引数に指定できるパターン文字列です。 • ローカルタイムで使用できるパターンを指定できます。タイムゾーンで使用するパターンは指定できません。 • 次の文字はパターン文字の予約語として使用されるため，文字列として使用する場合はシングルクォーテーションで囲む必要があります。また，シングルクォーテーション（'）自身を指定する場合は「''」のように指定します。 （予約語）[a-zA-Z] （指定例）「T」と「'」を使用したい場合 yyyy-MM-dd'T'HH:mm:ss'' 出力例：「2015-03-04T11:22:33'」

（凡例）

−：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

指定可能な値については，「3.3.11(5)　要求電文 list-option 要素の指定値について」を参照してください。

●ファイル一覧書き込みオペレーション

ビジネスプロセスからファイル操作アダプタに渡す要求電文フォーマットを次に示します。ファイル名は「adpfop_write_list_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/write_list_request」です。

表 3-91　要求電文フォーマット（ファイル一覧書き込みオペレーション）

タグ名			出現回数※1	説明
<request>			1 回	−
	<request-id>		0 または 1 回	受付で作成したリクエスト ID です。作業フォルダのパス生成に使用します。出力フォルダに作業フォルダを指定しない場合は，省略できます。
	<input-folder-name>		1 回	ファイル一覧を取得する共通フォルダ名を指定します。
	<input-file-name>		0 または 1 回	ファイル一覧を取得するファイルもしくはサブフォルダのパスを指定します。パスは共通フォルダからの相対パスで指定します。 当該要素が省略された場合，共通フォルダ自体を対象とします。
	<output-folder-name>		1 回	取得したファイル一覧を出力するディレクトリを指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を指定し，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を指定し，値に共通フォルダ定義名を指定します。
	<output-file-name>		0 または 1 回	取得したファイル一覧を出力するファイルを指定します。
	<options>		0 または 1 回	オプションを指定する場合に使用します。 当該要素が省略された場合はオプション項目をすべて省略時の動作とします。
		<list-option>	0 または 1 回	出力するファイル一覧の出力形式を指定します。※2
		<date-time-pattern>	0 または 1 回	最終更新日時の出力形式を指定します。 • 当該要素が省略もしくは空白の場合は，「yyyyMMddHHmmss」が仮定されます。 • 指定できるパターン文字列は java.time.format.DateTimeFormatter#ofPattern(java.lang.String)の引数に指定できるパターン文字列です。 • ローカルタイムで使用できるパターンを指定できます。タイムゾーンで使用するパターンは指定できません。 • 次の文字はパターン文字の予約語として使用されるため，文字列として使用する場合はシングルクォーテーションで囲む必要があります。また，シングルクォーテーション（'）自身を指定する場合は「''」のように指定します。 （予約語）[a-zA-Z] （指定例）「T」と「'」を使用したい場合 yyyy-MM-dd'T'HH:mm:ss'' 出力例：「2015-03-04T11:22:33'」

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

指定可能な値については，「3.3.11(5)　要求電文 list-option 要素の指定値について」を参照してください。

●ファイル読み込みオペレーション

ビジネスプロセスからファイル操作アダプタに渡す要求電文フォーマットを次に示します。ファイル名は「adpfop_read_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/read_request」です。

表 3-92　要求電文フォーマット（ファイル読み込みオペレーション）

タグ名		出現回数※	説明
<request>		1 回	－
	<request-id>	0 または 1 回	受付で作成したリクエスト ID です。作業フォルダのパス生成に使用します。出力フォルダに作業フォルダを指定しない場合は，省略できます。
	<input-folder-name>	1 回	読み込むファイルが格納されているフォルダ名を指定します。common 属性に，true または false を必ず指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を指定し，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を指定し，値に共通フォルダ定義名を指定します。
	<input-file-name>	1 回	取得するファイルまたはサブフォルダのパスを指定します。パスは共通フォルダからの相対パスで指定します。 • 共通フォルダから入力する場合 共通フォルダからの相対パスを指定します。 • 作業フォルダから入力する場合 入力ファイル名を指定します。
	<input-file-validation>	0 または 1 回	入力ファイルの検証を実施するかどうかを ON または OFF で指定します。 • ON：検証を実施します。 • OFF：検証を実施しません。 指定を省略した場合は OFF が設定されます。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル書き込みオペレーション

ビジネスプロセスからファイル操作アダプタに渡す要求電文フォーマットを次に示します。ファイル名は「adpfop_write_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/write_request」です。

ファイル書き込みオペレーションの要求電文は，ヘッダとボディで指定します。ヘッダとボディの要求電文フォーマットを次の表に示します。

表 3-93　要求電文フォーマット（ファイル書き込みオペレーション）

要求電文	タグ名		出現回数※	説明
ヘッダ	<request>		1 回	−
		<request-id>	0 または 1 回	受付で作成したリクエスト ID です。作業フォルダのパス生成に使用します。出力フォルダに作業フォルダを指定しない場合は，省略できます。
		<output-folder-name>	1 回	ファイルを書き込むフォルダを指定します。common 属性に，true または false を必ず指定します。 • 作業フォルダを指定する場合 タグの属性に common="false"を指定し，値に空文字を指定します。 • 共通フォルダを指定する場合 タグの属性に common="true"を指定し，値に共通フォルダ定義名を指定します。
		<output-file-name>	0 または 1 回	書き込むファイルのパスを共通フォルダからの相対パスで指定します。 output-folder-name 要素で，common 属性に true を指定した場合は，必ず指定します。output-folder-name 要素で，common 属性に false を指定した場合は，指定しても無視されます。
		<output-mode>	0 または 1 回	ファイルの出力モードを new または append で指定します。 • new：ファイルを新規作成します。既存のファイルがある場合は，上書きされます。 • append：既存のファイルの末尾に出力ファイルの記述を追加して，書き込みます。 この要素を省略した場合は new が設定されます。 次の場合は，この要素の設定は無視され，new が設定されます。 • 出力ファイルの形式が XML（ボディの電文フォーマットが XML 形式）の場合 • 作業フォルダに出力する場合
		<output-file-validation>	0 または 1 回	出力ファイルの検証を実施するかどうかを ON または OFF で指定します。 • ON：検証を実施します。 • OFF：検証を実施しません。 指定を省略した場合は OFF が設定されます。
ボディ	ボディの内容が，出力データとしてファイルに書き込まれます。サービスプラットフォームではスキーマファイルを提供していません。スキーマファイルはユーザが作成してください。ボディが Any 型の場合は，スキーマファイルは不要です。			

（凡例）

−：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル操作アダプタが要求電文の取得に失敗した場合

エラーメッセージ（KDEC80002-E）を出力して処理を終了します。

(b) ファイル操作アダプタの応答電文フォーマット

ファイル操作アダプタの応答電文フォーマットを，オペレーションごとに説明します。

●ファイル変換オペレーション

ファイル操作アダプタが呼び出し元のビジネスプロセスへ返す応答電文のフォーマットを次に示します。ファイル名は「adpfop_transform_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/transform_response」です。

表 3-94 応答電文フォーマット（ファイル変換オペレーション）

タグ名		出現回数※	説明
<response>		1 回	−
	<output-folder-name>	1 回	変換後のファイルを出力するディレクトリが設定されます。 • 作業フォルダが設定されている場合 タグの属性に common="false"が，値に空文字が設定されます。 • 共通フォルダが設定されている場合 タグの属性に common="true"が，値に共通フォルダ定義名が設定されます。
	<output-file-name>	1 回	変換後のファイルのファイル名（ファイルパス）が設定されます。共通フォルダを操作した場合，要求電文の output-file-name の値がそのまま設定されます。
	<output-file-size>	1 回	変換後のファイルのサイズ（バイト）が設定されます。

（凡例）

−：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル複製オペレーション

ファイル操作アダプタが呼び出し元のビジネスプロセスへ返す応答電文のフォーマットを次に示します。ファイル名は「adpfop_copy_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/copy_response」です。

表 3-95 応答電文フォーマット（ファイル複製オペレーション）

タグ名	出現回数※	説明
<response>	1 回	−

タグ名		出現回数※	説明
	<output-folder-name>	1 回	複製後のファイルを出力するディレクトリが設定されます。 • 作業フォルダが設定されている場合 タグの属性に common="false"が，値に空文字が設定されます。 • 共通フォルダが設定されている場合 タグの属性に common="true"が，値に共通フォルダ定義名が設定されます。
	<output-file-name>	1 回	複製後のファイルのファイル名（ファイルパス）が設定されます。共通フォルダを操作した場合，要求電文の output-file-name の値がそのまま設定されます。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル移動オペレーション

ファイル操作アダプタが呼び出し元のビジネスプロセスへ返す応答電文のフォーマットを次に示します。ファイル名は「adpfop_move_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/move_response」です。

表 3-96　応答電文フォーマット（ファイル移動オペレーション）

タグ名		出現回数※	説明
<response>		1 回	－
	<output-folder-name>	1 回	移動後のファイルが格納されているディレクトリ名を設定されます。 • 作業フォルダを操作した場合 タグの属性に common="false"が設定され，値に空文字が設定されます。 • 共通フォルダを操作した場合 タグの属性に common="true"が設定され，値に共通フォルダ定義名が設定されます。
	<output-file-name>	1 回	移動後のファイルが格納されている共通フォルダもしくは作業フォルダからの相対パスを設定します。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル・フォルダ削除オペレーション

ファイル操作アダプタが呼び出し元のビジネスプロセスへ返す応答電文のフォーマットを次に示します。ファイル名は「adpfop_delete_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/delete_response」です。

表 3-97　応答電文フォーマット（ファイル・フォルダ削除オペレーション）

タグ名		出現回数※1	説明
<response>		1 回	－
	<folder-name>	1 回	削除されたファイルが格納されているディレクトリ名が設定されます。 • 作業フォルダが設定されている場合 タグの属性に common="false"が，値に空文字が設定されます。 • 共通フォルダが設定されている場合 タグの属性に common="true"が，値に共通フォルダ定義名が設定されます。
	<file-name>	0 または 1 回	削除されたファイルのファイル名（ファイルパス）が設定されます。 要求電文の file-name の値がそのまま設定されます。※2
	<sub-folder-name>	0 または 1 回	削除されたファイルのサブフォルダ名が設定されます。要求電文の sub-folder-name の値がそのまま設定されます。※2

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

file-name タグと sub-folder-name タグについて，次に説明します。

・共通フォルダを操作した場合は，どちらか一方だけ出力されます。ファイル削除の場合は file-name タグ，サブフォルダ削除の場合は sub-folder-name タグが出力されます。

・作業フォルダを指定する場合は，どちらも出力されません。

●ファイル圧縮オペレーション

ファイル操作アダプタが呼び出し元のビジネスプロセスへ返す応答電文のフォーマットを次に示します。ファイル名は「adpfop_compress_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/compress_response」です。

表 3-98　応答電文フォーマット（ファイル圧縮オペレーション）

タグ名		出現回数※	説明
<response>		1 回	－
	<output-folder-name>	1 回	圧縮されたファイルが格納されているディレクトリ名が設定されます。 • 作業フォルダが設定されている場合 タグの属性に common="false"が，値に空文字が設定されます。 • 共通フォルダが設定されている場合 タグの属性に common="true"が，値に共通フォルダ定義名が設定されます。
	<output-file-name>	1 回	圧縮されたファイルのファイル名（ファイルパス）が設定されます。共通フォルダを操作した場合，要求電文の output-file-name の値がそのまま設定されます。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル伸張オペレーション

ファイル操作アダプタが呼び出し元のビジネスプロセスへ返す応答電文のフォーマットを次に示します。ファイル名は「adpfop_extract_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/extract_response」です。

表 3-99　応答電文フォーマット（ファイル伸張オペレーション）

タグ名				出現回数※	説明
<response>				1 回	－
	<output-folder-name>			1 回	伸張されたファイルが格納されているディレクトリ名が設定されます。 • 作業フォルダが設定されている場合 タグの属性に common="false"が，値に空文字が設定されます。 • 共通フォルダが設定されている場合 タグの属性に common="true"が，値に共通フォルダ定義名が設定されます。
	<output-sub-folder-name>			0 または 1 回	伸張されたファイルが格納されているサブフォルダパスが設定されます。要求電文の output-sub-folder-name の値がそのまま設定されます。 ただし，次の場合は出力されません。 • ファイルの出力先が作業フォルダの場合 • 要求電文に output-sub-folder-name の指定がない場合
	<output-files>			1 回	－
		<output-file>		1 から 999 回	－
			<output-file-name>	1 回	伸張されたファイルのファイル名が設定されます。
			<entry-name>	1 回	output-file-name タグに設定されたファイルの伸張前のエントリ名が設定されます。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル一覧取得オペレーション

ファイル操作アダプタが呼び出し元のビジネスプロセスへ返す応答電文のフォーマットを次に示します。ファイル名は「adpfop_get_list_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/get_list_response」です。

表 3-100　応答電文フォーマット（ファイル一覧取得オペレーション）

タグ名				出現回数※	説明
<response>				1 回	−
	<file-list>			1 回	ファイルが 1 つも存在しない場合も生成します。
		<file>		0 から 999 回	一覧取得したファイル単位ごとに指定します。 要求電文の list-option で”l”が指定された場合，属性 type には，f か d が必ず設定されます。 • f：対象がファイルの場合。 • d：対象がディレクトリの場合。 ファイルが 1 つも存在しない場合には生成されません。
			<name>	1 回	ファイル名またはディレクトリ名を設定します。
			<size>	0 または 1 回	ファイルサイズをバイト単位で設定します。 要求電文の list-option 要素で”l”が指定された場合だけ出力します。 対象がディレクトリの場合には 0 を設定します。
			<last-modified-time>	0 または 1 回	ファイルの最終更新日時を出力します。 要求電文の list-option 要素で”l”が指定された場合だけ出力します。 出力形式は要求電文の date-time-pattern 要素の指定パターンで出力します。要求電文の date-time-pattern 要素が省略された場合は「yyyy-MM-dd'T'HH:mm:ss」の形式で出力します。

（凡例）

−：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル一覧書き込みオペレーション

ファイル操作アダプタが呼び出し元のビジネスプロセスへ返す応答電文のフォーマットを次に示します。ファイル名は「adpfop_write_list_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/write_list_response」です。

表 3-101　応答電文フォーマット（ファイル一覧書き込みオペレーション）

タグ名		出現回数※	説明
<response>		1 回	−
	<output-folder-name>	1 回	ファイル一覧情報が書き込まれたファイルが格納されているディレクトリ名を設定します。 • 作業フォルダを操作した場合 タグの属性に common="false"が設定され，値に空文字が設定されます。 • 共通フォルダを操作した場合 タグの属性に common="true"が設定され，値に共通フォルダ定義名が設定されます。

タグ名		出現回数※	説明
	<output-file-name>	1 回	ファイル一覧情報を書き込んだファイルが格納されている共通フォルダもしくは作業フォルダからの相対パスを設定します。
	<output-file-size>	1 回	ファイル一覧情報を書き込んだファイルのサイズを設定します。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル読み込みオペレーション

ファイル読み込みオペレーションの応答電文は，読み込んだファイルの内容となるため，サービスプラットフォームではスキーマファイルを提供していません。

●ファイル書き込みオペレーション

ファイル操作アダプタが呼び出し元のビジネスプロセスへ返す応答電文のフォーマットを次に示します。ファイル名は「adpfop_write_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/fileoperation/write_response」です。

表 3-102　応答電文フォーマット（ファイル書き込みオペレーション）

タグ名		出現回数※	説明
<response>		1 回	－
	<output-folder-name>	1 回	タグの属性 common に，true または false が設定されます。 • 作業フォルダに出力した場合 タグの属性に common="false"が設定され，値に空文字が設定されます。 • 共通フォルダに出力した場合 タグの属性に common="true"が設定され，値に共通フォルダ定義名が設定されます。
	<output-file-name>	1 回	書き込んだファイルのファイル名，またはファイルパスが設定されます。
	<output-file-size>	1 回	書き込んだファイルのサイズ（単位：バイト）が設定されます。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

●ファイル操作アダプタが応答電文の生成に失敗した場合

エラーメッセージ（KDEC80003-E）を出力して処理を終了します。

(2)　サービスアダプタ定義画面での操作

ファイル操作アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。
   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービスアダプタ定義画面（基本）およびサービスアダプタ定義画面（詳細）で定義情報を設定します。
   各オペレーションで設定する項目については，次の個所を参照してください。
   - ファイル変換オペレーションの場合
     3.3.19(11)(a)　ファイル変換オペレーション
   - ファイル複製オペレーションの場合
     3.3.19(11)(b)　ファイル複製オペレーション
   - ファイル移動オペレーションの場合
     3.3.19(11)(c)　ファイル移動オペレーション
   - ファイル・フォルダ削除オペレーションの場合
     3.3.19(11)(d)　ファイル・フォルダ削除オペレーション
   - ファイル圧縮オペレーションの場合
     3.3.19(11)(e)　ファイル圧縮オペレーション
   - ファイル伸張オペレーションの場合
     3.3.19(11)(f)　ファイル伸張オペレーション
   - ファイル一覧取得オペレーションの場合
     3.3.19(11)(g)　ファイル一覧取得オペレーション
   - ファイル一覧書き込みオペレーションの場合
     3.3.19(11)(h)　ファイル一覧書き込みオペレーション
   - ファイル読み込みオペレーションの場合
     3.3.19(11)(i)　ファイル読み込みオペレーション
   - ファイル書き込みオペレーションの場合
     3.3.19(11)(j)　ファイル書き込みオペレーション

   また，サービスアダプタ定義画面（基本）については，マニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

参考

作成したファイル操作アダプタは，必要に応じて保存，編集，および検証できます。

ファイル操作アダプタを編集する方法については，「3.5　サービスアダプタの編集」を参照してください。ファイル操作アダプタを検証する方法については，「3.6　サービスアダプタの検証」を参照してください。

(3)　ファイル操作アダプタの定義ファイルの作成および編集

作成および編集する定義ファイルの種類を次に示します。

- ファイル操作アダプタ実行環境プロパティファイル
  ファイル操作アダプタごとの構成情報を定義するためのファイルです。
- ファイル操作アダプタ定義ファイル
  ファイル操作アダプタの動作内容を定義するためのファイルです。

ファイル操作アダプタで使用する定義ファイルの作成手順および編集手順を次に示します。

(a)　ファイル操作アダプタ実行環境プロパティファイルの作成手順

ファイル操作アダプタ実行環境プロパティファイルの作成手順を示します。

1. テンプレートファイル（<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¥config¥adpfop¥templates¥adpfop.properties）をコピーして，次のディレクトリに格納します。
   <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>¥CSC¥config¥adpfop
2. コピーしたテンプレートファイルのファイル名を「<サービス ID>.properties」に変更します。
   サービス ID とは，ファイル操作アダプタを新規に追加するときに指定する任意の文字列です。
3. 定義内容を編集して保存します。
   ファイル操作アダプタ実行環境プロパティファイルで編集できる項目については，マニュアル「サービスプラットフォーム リファレンス」の「6.8.8　ファイル操作アダプタ実行環境プロパティファイル」を参照してください。

ファイル操作アダプタ実行環境プロパティファイルは，ファイル操作アダプタを開始する際に実行環境に反映されます。ファイル操作アダプタ実行環境プロパティファイルの内容を変更する場合は，ファイル操作アダプタを停止してから作業をします。ファイル操作アダプタを再開始すると，変更内容が実行環境に反映されます。

(b)　ファイル操作アダプタ定義ファイルの編集手順

ファイル操作アダプタ定義ファイルの編集手順を示します。

1. サービスアダプタ定義画面（詳細）の［独自定義ファイル］で「cscFileOperation.properties」を選択し，［編集］ボタンをクリックします。
   ファイル操作アダプタ定義ファイルを編集するためのエディタが起動します。

2. エディタ上でファイル操作アダプタ定義ファイルを編集します。

ファイル操作アダプタ定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.5.2　ファイル操作アダプタ定義ファイル」を参照してください。

3. Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

ファイル操作アダプタを開始すると，ファイル操作アダプタ定義ファイルに設定した内容が有効になります。

(4)　ファイル変換オペレーション使用時の注意事項

ファイル変換オペレーションで分割処理方式を選択して，入力ファイルに改行セパレータ形式のファイルを扱う場合，次のファイルの複合内容要素のセパレータには，LineFeed を設定しないでください。

- 入力側のヘッダレコード部のバイナリフォーマット定義ファイル
- 入力側のデータレコード部のバイナリフォーマット定義ファイル
- 入力側のトレーラレコード部のバイナリフォーマット定義ファイル

LineFeed を設定した場合，メッセージ（KDEC80043-E）が出力され，ファイル変換の処理が中止されます。

正しく設定したセパレータの設定画面例を次に示します。

図 3-21　セパレータの設定画面例（LineFeed 設定なしの場合）

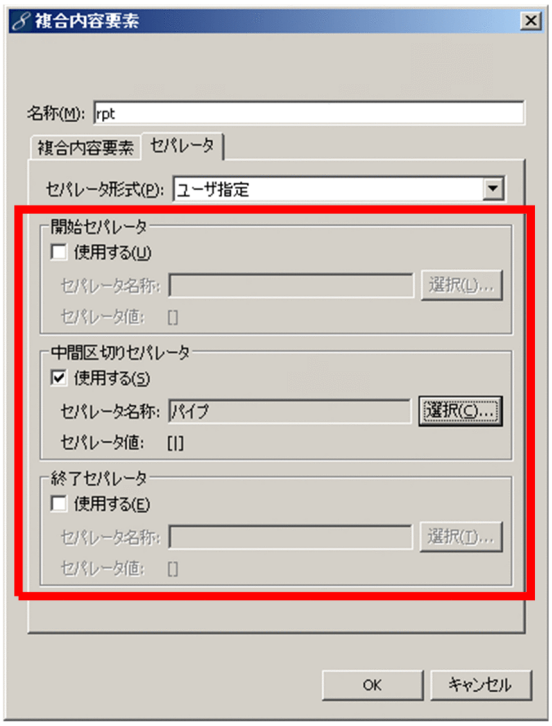
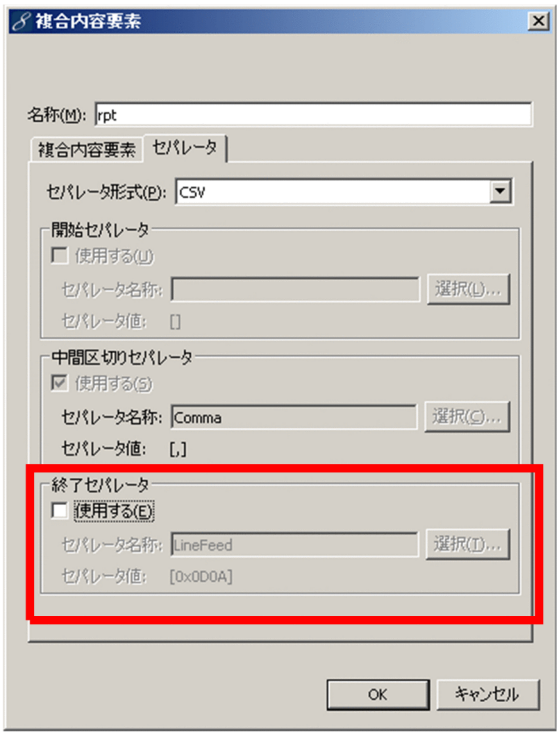


図 3-22　セパレータの設定画面例（CSV 形式の場合）



(5)　要求電文 list-option 要素の指定値について

ファイル一覧取得オペレーションとファイル一覧書き込みオペレーションの要求電文 list-option 要素について示します。

- ファイル一覧の出力形式を，ハイフン以降のオプション文字で指定できます。
  指定例：「-l」，「-rl」，「-l -r」
- 要素の値には次の形式の文字列だけ指定できます。

```
{空白}※1*[-][a-zA-Z0-9]+({セパレータ}※2+[-][a-zA-Z0-9]+)*{空白}*
```

  注※1
  {空白}：XML 解析上トリムされる前後空白文字

  注※2
  {セパレータ}：半角スペース，改行，タブ文字

- 有効なオプション文字は「l」（小文字エル），「r」，「t」だけです。そのほかの文字は無視します。大文字小文字は区別されます。
- GUN 形式（ハイフン 2 個）のオプション指定はサポートしません。

オプション文字の指定有無の動作を次に示します。

表 3-103　オプション文字指定有無の動作

オプション文字	指定有無	動作
l（エル）	指定なし	ファイル名だけを出力します。
l（エル）	指定あり	ファイル名，ファイル種別，サイズ，最終更新日時を出力します。
r	指定なし	出力する一覧のソート順を昇順にします。
r	指定あり	出力する一覧のソート順を降順にします。
t	指定なし	出力する一覧のソートキーをファイル名にします。※
t	指定あり	出力する一覧のソートキーを最終更新日時にします。※

注※

ソートはファイルとディレクトリを区別しません。

3.3.12　メールアダプタを定義する

メールアダプタの定義方法について説明します。

(1)　メールアダプタを使用するときの注意事項

メールアダプタを設定する前に知っておく必要がある注意事項について説明します。

- メールアダプタの宛先には，有効なアドレスを設定してください。
- メールアダプタに無効なアドレスを設定した場合，そのアドレス宛てのメールはメールサーバの機能により返送（バウンス）されます。
- FTP クライアントや FTP サーバから転送されたファイルを添付ファイルとして使用する場合は，メールアダプタのほかに FTP インバウンドアダプタ，FTP 受付，および FTP アダプタとの連携が必要です。FTP と連携する機能については，マニュアル「サービスプラットフォーム 解説」の「8.　FTP と連携してファイルを転送する機能（FTP 連携)」を参照してください。

(2)　電文フォーマットの作成

メールアダプタで使用する電文フォーマットの種類と形式を次に示します。

(a)　電文フォーマットの種類

メールアダプタで使用する電文フォーマットは次の 2 種類です。

- 要求電文フォーマット
- 応答電文フォーマット

要求電文フォーマットには，メールサーバの情報や，メール本文，アドレス，主題など，ユーザ入力情報を設定します。応答電文フォーマットには送信メールのメッセージ ID が設定されます。

メールアダプタでは，サービスプラットフォームが提供する XML スキーマを電文フォーマットに使用します。要求電文および応答電文に使用する XML スキーマは，それぞれ「adpmail_smtp_request.xsd」および「adpmail_smtp_response.xsd」です。

(b) 電文フォーマットの形式

ここでは，メールアダプタで使用する要求電文フォーマットおよび応答電文フォーマットの形式について説明します。

ファイルの格納場所は「<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¥schema¥mail」です。

●adpmail_smtp_request.xsd（要求電文フォーマットファイル）

メールアダプタがビジネスプロセスから呼び出されたときに，メールアダプタに渡される要求電文のフォーマットを次に示します。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/mail/send_request」です。

表 3-104 要求電文フォーマット

タグ名	種別	出現回数	説明
<request>	−	1 回	−
<smtp-host-name>	string	0 または 1 回	接続するメールサーバの IP アドレスまたはホスト名を示すタグです。 • タグがあって値がある場合 指定の文字列で接続します。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。
<request-id>	string	0 または 1 回	受付で作成したリクエスト ID を示すタグです。 • タグがあって値がある場合 リクエスト ID として利用されます。 • タグがない，またはタグがあって値がない場合 作業フォルダを利用するときはシステム例外を返します。共通フォルダを利用するときは無視されます。
<smtp-port>	long	0 または 1 回	接続するメールサーバのポート番号を示すタグです。 • タグがあって値がある場合 1 から 65535 までのポート番号を指定します。これ以外の値を指定した場合，システム例外を返します。 • タグがあって値がない場合 デフォルトの 25 が指定されます。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。 メールアダプタ実行環境プロパティファイルにも指定がない場合，デフォルトの 25 が指定されます。

タグ名	種別	出現回数	説明
<mail-user>	string	0 または 1 回	接続するメールサーバのログインユーザ名を示すタグです。 • タグがあって値がある場合 0x20～0x7E の範囲で指定された文字が有効になります。この範囲外の文字が指定された場合，システム例外を返します。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。
<mail-addr-to>※1	string	0 または 1 回	メールの送付先ユーザの電子メールアドレスを示すタグです。複数の電子メールアドレスを指定する場合，電子メールアドレスをコンマ（,）で区切ります。 • タグがあって値がある場合 「TO」フィールドのアドレスとして使用されます。アドレスは検証されません。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。
<mail-addr-from>	string	0 または 1 回	メールの発信元ユーザの電子メールアドレスを示すタグです。指定できる電子メールアドレスは 1 つだけです。 • タグがあって値がある場合 「FROM」フィールドのアドレスとして使用されます。アドレスは検証されません。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。
<mail-addr-cc>※1	string	0 または 1 回	「CC」フィールドに設定するユーザの電子メールアドレスを示すタグです。 複数の電子メールアドレスを指定する場合，電子メールアドレスをコンマ（,）で区切ります。 • タグがあって値がある場合 「CC」フィールドのアドレスとして使用されます。アドレスは検証されません。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。
<mail-addr-bcc>※1	string	0 または 1 回	「BCC」フィールドに設定するユーザの電子メールアドレスを示すタグです。 複数の電子メールアドレスを指定する場合，電子メールアドレスをコンマ（,）で区切ります。

タグ名	種別	出現回数	説明
<mail-addr-bcc>※1	string	0 または 1 回	• タグがあって値がある場合 「BCC」フィールドのアドレスとして使用されます。アドレスは検証されません。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。
<mail-subject-text>	string	0 または 1 回	メールの主題を示すタグです。 • タグがあって値がある場合 メールの主題として使用されます。ただし，主題の文字数がメールアダプタ実行環境プロパティファイルで指定したサイズを超えた場合，超えた文字が切り落とされます。 • タグがあって値がない場合 主題なしでメールが送信されます。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。メールアダプタ実行環境プロパティファイルにも指定がない場合，主題なしでメールが送信されます。
<mail-body>※2	−	0 または 1 回	メールの本文を示すタグです。メール本文は<message>および<file>で指定します。 • タグがあって値がある場合 メールの本文として使用されます。※3 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。メールアダプタ実行環境プロパティファイルにも指定がない場合，本文なしでメールが送信されます。
<message>	string	0 または 1 回	メール本文として使用する文字列を示すタグです。「<」，「>」などの文字を使う場合は，事前に文字列をサニタイズしてください。 • タグがあって値がある場合 メールの本文として使用されます。 • タグがあって値がない，または<file>の指定がない場合 本文なしでメールが送信されます。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。
<file>	−	0 または 1 回	内容をメール本文として使用するファイルのパスおよび文字セットを示すタグです。ファイルパスと文字セットは，<folder-name>，<mail-body-file-name>，および<file-charset>で指定します。 • タグがあって値がある場合 メール本文として使用されます。ファイルの内容をメール本文とする場合，文字列のサニタイズは不要です。

タグ名			種別	出現回数	説明
		<file>	−	0 または 1 回	• タグがあって値がない場合 システム例外を返します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。
			<folder-name>	string	1 回	ファイルを取得するフォルダを示すタグです。 • タグの属性に common="true"が指定されている場合 ファイルを取得するフォルダとして共通フォルダを利用し，このタグには「共通フォルダ定義名」を指定します。 指定された共通フォルダ定義名から共通フォルダが取得できないときは，システム例外を返します。 • タグの属性に common="false"が指定されている場合 ファイルを取得するフォルダとして作業フォルダを利用します。 • common 属性がない，common 属性の値が"true"および"false"のどちらでもない，またはタグがない場合 システム例外を返します。
			<mail-body-file-name>	string	1 回	取得するファイル名を示すタグです。 • 作業フォルダを利用する場合 作業フォルダ直下のファイルを指定してください。 ファイル名の先頭以外にスラッシュ（/）や円マーク（¥）などの区切り文字は使用できません。ファイル名の先頭にスラッシュ（/）を使用した場合，スラッシュ（/）は無視されます。 • 共通フォルダを利用する場合 共通フォルダルート配下のファイルを指定してください。 共通フォルダルートより上位のパスを指定した場合，エラーになります。 相対パスの先頭のファイル区切り文字（/）は省略できます。 サブフォルダを使用する場合は，共通フォルダルートからファイル名までの相対パスを指定してください。 複数のファイル名は指定できません。 • タグがあって値が上記の制限に従っている場合 ファイル名として使用されます。 • タグがあって値が上記の制限に違反する場合 システム例外を返します。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
			<file-charset>	string	0 または 1 回	メール本文の内容が記述されたファイルの文字セットを指定します。 • タグがあって値がある場合 文字セットとして使用されます。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合

タグ名	種別	出現回数	説明
<file-charset>	string	0 または 1 回	メールアダプタ実行環境プロパティファイルの値が有効になります。メールアダプタ実行環境プロパティファイルにも指定がない場合，UTF -8 が文字セットとして使用されます。
<attachment-file>	−	0 または 1 回	メールの添付ファイルを示すタグです。添付ファイルは<item>で指定します。複数の添付ファイルを指定できます。 • タグがあって値がある場合 メールの添付ファイルとして使用されます。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効になります。
<item>	−	1 回以上	メールの添付ファイルの詳細を示すタグです。添付ファイルの詳細は<attach-folder>および<attach-file-name>で指定します。 • タグがあって値がある場合 メールの添付ファイルとして使用されます。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
<attach-folder>	string	1 回	添付ファイルが格納されているフォルダを示すタグです。 • タグの属性に common="true"が指定されている場合 ファイルを取得するフォルダとして共通フォルダを利用し，このタグには「共通フォルダ定義名」を指定します。 指定された共通フォルダ定義名から共通フォルダが取得できないときは，システム例外を返します。 • タグの属性に common="false"が指定されている場合 ファイルを取得するフォルダとして作業フォルダを利用します。 • common 属性がない，common 属性の値が"true"および"false"のどちらでもない，またはタグがない場合 システム例外を返します。
<attach-file-name>	string	1 回	添付ファイル名を示すタグです。 • 作業フォルダを利用する場合 作業フォルダ直下のファイルを指定してください。 ファイル名の先頭以外にスラッシュ（/）や円マーク（¥）などの区切り文字は使用できません。ファイル名の先頭にスラッシュ（/）を使用した場合，スラッシュ（/）は無視されます。 • 共通フォルダを利用する場合 共通フォルダルート配下のファイルを指定してください。 共通フォルダルートより上位のパスを指定した場合，エラーになります。 相対パスの先頭のファイル区切り文字（/）は省略できます。 複数のファイル名は指定できません。 • タグがあって値が上記の制限に従っている場合

タグ名	種別	出現回数	説明
<attach-file-name>	string	1 回	ファイル名として使用されます。 • タグがあって値が上記の制限に違反する場合 システム例外を返します。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
<mail-format>	string	0 または 1 回	メールフォーマットを示すタグです。次のどれかを指定します。 text/plain：メール本文をプレーンテキストで作成します。 text/html：メール本文を html 形式で作成します。 text/xml：メール本文を xml 形式で作成します。※4 • タグがあって値が「text/plain」，「text/html」，または「text/xml」の場合 メールフォーマットとして使用されます。 • タグがあって値が「text/plain」，「text/html」，または「text/xml」のどれでもない場合 システム例外を返します。 • タグがあって値がない場合 JavaMail の設定に依存します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。 メールアダプタ実行環境プロパティファイルにも指定がない場合，「text/plain」でメール本文が作成されます。
<mail-encoding>	string	0 または 1 回	メールのエンコードを示すタグです。 • タグがあって値がある場合 メールのエンコードとして使用されます。 • タグがあって値がない場合 JavaMail の設定に依存します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。 メールアダプタ実行環境プロパティファイルにも指定がない場合，ISO-2022-JP が使用されます。
<mail-auth>	string	0 または 1 回	SMTP 認証の形式を示すタグです。次のどちらかを指定します。 PLAIN：認証しません。 LOGIN：ログイン時にユーザ名およびパスワードで認証します。 • タグがあって値が「PLAIN」または「LOGIN」の場合 SMTP 認証の形式として使用されます。 • タグがあって値が「PLAIN」または「LOGIN」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 JavaMail の設定に依存します。 • タグがない場合

タグ名	種別	出現回数	説明
<mail-auth>	string	0 または 1 回	メールアダプタ実行環境プロパティファイルの値が有効となります。 メールアダプタ実行環境プロパティファイルにも指定がない場合，「PLAIN」でメールが送信されます。
<mail-option>	−	0 回以上	ユーザ定義のメールヘッダを示すタグです。 ヘッダ名および値は，<option-key>および<option-value>を組み合わせて指定します。 • タグがあって値がある場合 ユーザ定義のメールヘッダとして使用されます。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。メールアダプタ実行環境プロパティファイルにも指定がない場合，ユーザ定義のメールヘッダなしでメールが送信されます。
<option-key>	string	1 回	ユーザ定義のメールヘッダのキー名を指定します。 • タグがあって値がある場合 メールヘッダのキー名として使用します。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
<option-value>	string	1 回	ユーザ定義のメールヘッダの値を指定します。 • タグがあって値がある場合 メールヘッダの値として使用します。 • タグがない，またはタグがあって値がない場合 システム例外を返します。
<smtps>	−	0 または 1 回	TLS に関する設定を示すタグです。 • タグがあって値がある場合 タグの値に従い，SMTP，または TLS を使用して SMTP サーバに接続します。 • タグがあって値がない場合 システム例外を返します。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。 メールアダプタ実行環境プロパティファイルに指定がない場合，SMTP を使用して SMTP サーバに接続します。
<smtps-enable>	boolean	0 または 1 回	SMTP サーバとの接続に TLS を使用するかどうかを指定するタグです。次のどちらかを指定します。 true：TLS を使用して SMTP サーバに接続します。 false：TLS を使用しません。SMTP で SMTP サーバに接続します。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合

タグ名	種別	出現回数	説明
<smtps-enable>	boolean	0 または 1 回	システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。 メールアダプタ実行環境プロパティファイルに指定がない場合，SMTP を使用して SMTP サーバに接続します。
<smtps-starttls-enable>	boolean	0 または 1 回	SMTP サーバとの接続に TLS を使用する場合に STARTTLS を使用するかどうかを指定するタグです。次のどちらかを指定します。 true：STARTTLS を使用します。通常の SMTP で通信を開始し，「STARTTLS」で双方が TLS に対応しているか確認したあとに TLS で再接続します。 false：STARTTLS を使用しません。SMTP 通信がすべて TLS によって暗号化されます。 この指定は，SMTP サーバとの接続に TLS を使用する場合だけ有効となります。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「true」が使用されます。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。 メールアダプタ実行環境プロパティファイルに指定がない場合，TLS によって暗号化されたメールが送信されます。
<smtps-starttls-required>	boolean	0 または 1 回	メールアダプタで STARTTLS を使用する設定にしている場合で，接続先の SMTP サーバで STARTTLS が処理できないときに，メールアダプタでエラーを発生させるかどうかを指定するタグです。 true：接続先の SMTP サーバが STARTTLS をサポートしていない場合，または STARTTLS の処理中にエラーが発生した場合，メールアダプタでエラーを発生させます。 false：接続先の SMTP サーバが STARTTLS をサポートしていない場合，または STARTTLS の処理中にエラーが発生した場合，メールアダプタでエラーを発生させません。SMTP を使用して SMTP サーバに接続します。 この指定は，SMTP サーバとの接続に TLS と STARTTLS を使用する場合だけ有効となります。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合

タグ名	種別	出現回数	説明
<smtps-starttls-required>	boolean	0 または 1 回	デフォルトの「false」が使用されます。 • タグがない場合 メールアダプタ実行環境プロパティファイルの値が有効となります。 メールアダプタ実行環境プロパティファイルに指定がない場合，STARTTLS を使用する設定に依存することなく，TLS によって暗号化されたメールが送信されます。

（凡例）

－：該当する項目はありません。

注※1

<mail-addr-to>，<mail-addr-cc>，および<mail-addr-bcc>にアドレス指定がない，かつメールアダプタ実行環境プロパティファイルにも指定がない場合，システム例外を返します。

注※2

要求電文の<mail-body>の設定と，メールアダプタ実行環境プロパティファイルの mailadp.mail.body-text キーおよび mailadp.mail.body-filepath キーの設定は背反となります。例えば，要求電文の<mail-body>で<message>だけを指定して<file>を指定しなかった場合，mailadp.mail.body-filepath キーでファイルを指定しても，<file>として本文に設定されることはありません。

注※3

<message>および<file>の両方を指定した場合，最初に<message>に指定された文字列がメール本文に設定されます。そのあと，<file>に指定されたファイルの内容がメール本文に設定されます。

注※4

text/xml を指定した場合，メールアダプタではプレーンテキスト形式で表示されます。

●adpmail_smtp_response.xsd（応答電文フォーマットファイル）

メールアダプタが呼び出し元のビジネスプロセスに返す応答電文のフォーマットを次に示します。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/mail/send_response」です。

表 3-105　応答電文フォーマット

タグ名	種別	出現回数	説明
<response>	－	1 回	－
<message-id>	string	1 回	送信されるメールメッセージのメッセージ ID を示すタグです。 メール送信時，内部で作成された MimeMessage がメールサーバに送信されます。この MimeMessage の ID がメッセージ ID となります。

（凡例）

－：該当する項目はありません。

(3)　メールアダプタの定義ファイルの作成

作成する定義ファイルの種類を次に示します。

- メールアダプタコマンド定義ファイル
  コマンドメッセージログの出力先，出力レベルなどを設定するためのファイルです。
- メールアダプタアカウント定義ファイル
  メールアダプタからメールサーバに接続するときの認証に必要なユーザ名およびパスワードを管理するためのファイルです。
- メールヘッダ定義ファイル
  メールの重要度，返信要求の有無など，ユーザが定義したメールヘッダのオプションを記述するためのファイルです。
- メールアダプタ実行環境プロパティファイル
  メールアダプタの構成情報を定義するためのファイルです。

各定義ファイルの作成手順を次に示します。

(a) メールアダプタコマンド定義ファイルの作成手順

1. テンプレートファイル（<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¥config¥mail¥templates¥adpmailcommand.properties）をコピーして，次のディレクトリに格納します。
   <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>¥CSC¥config¥mail
2. 定義内容を編集して保存します。

メールアダプタコマンド定義ファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.13.2　メールアダプタコマンド定義ファイル」を参照してください。

(b) メールアダプタアカウント定義ファイルの作成手順

1. csmmailaddusr コマンドを実行してメールアダプタアカウント定義ファイルを作成します。
   作成したメールアダプタアカウント定義ファイルのパスは，メールアダプタ実行環境プロパティファイルの mailadp.account.file.path キーで指定する必要があります。csmmailaddusr コマンドの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「csmmailaddusr（メールアダプタユーザ情報の登録・更新)」を参照してください。

(c) メールヘッダ定義ファイルの作成手順

1. テンプレートファイル（<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¥config¥mail¥templates¥mail-header.properties）をコピーして，任意のディレクトリに格納します。
2. コピーしたテンプレートファイルのファイル名を任意の名称に変更します。
3. 定義内容を編集して保存します。

メールヘッダ定義ファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.13.4　メールヘッダ定義ファイル」を参照してください。

(d)　メールアダプタ実行環境プロパティファイルの作成手順

1. テンプレートファイル（<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¥config¥mail¥templates¥adpmail.properties）をコピーして，次のディレクトリに格納します。

   全 HCSC サーバ共通の定義の場合

   <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>¥CSC¥config¥mail

   特定の HCSC サーバ用の定義の場合

   サブディレクトリ<HCSC サーバ名>を作成して格納してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

   <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¥config¥mail¥<HCSC サーバ名>

2. コピーしたテンプレートファイルのファイル名を「<サービス ID※>.properties」に変更します。

3. 定義内容を編集して保存します。

メールアダプタ実行環境プロパティファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.13.3　メールアダプタ実行環境プロパティファイル」を参照してください。

注※

サービス ID は，メールアダプタを新規に追加するときにサービスアダプタ定義画面で指定する任意の文字列です。

メールアダプタ実行環境プロパティファイルは，メールアダプタを開始する際に実行環境に反映されます。このため，メールアダプタ実行環境プロパティファイルの内容を変更する場合は，いったんメールアダプタを停止する必要があります。

(4)　サービスアダプタ定義画面での操作

メールアダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。

   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービスアダプタ定義画面（基本）で定義情報を設定します。

   サービスアダプタ定義画面（基本）で設定が必要な項目については，「3.3.19(12)　メールアダプタの場合」およびマニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

3. [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画面（詳細）が表示されます。

4. サービスアダプタ定義画面（詳細）で定義情報を設定します。

サービスアダプタ定義画面（詳細）で設定が必要な項目については，「3.3.19(12)　メールアダプタの場合」を参照してください。

(5)　メールアダプタのディレクトリ構成

サービスプラットフォームのインストール後のディレクトリ構成を次の図に示します。なお，ここではメールアダプタの定義を格納するディレクトリ，およびメールアダプタの電文スキーマを格納するディレクトリだけ記載しています。

図 3-23　　メールアダプタのディレクトリ構成

```
<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                  …HCSCのインストールディレクトリ
    ├config             …定義類を格納するディレクトリ
    │ └mail             …メールアダプタの定義を格納するディレクトリ
    │   └templates      …メールアダプタの定義のテンプレートを格納するディレクトリ
    ├schema             …スキーマ定義を格納するディレクトリ
    │ └mail             …メールアダプタの電文スキーマを格納するディレクトリ
    :                                   :
```

上記以外のディレクトリ構成については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「2.1.2　Service Platform のインストール」を参照してください。

(6)　メールアダプタの定義例

ここでは，標準受付に渡されたメッセージをメール本文に設定し，メールアダプタ実行環境プロパティファイルで設定した宛先にメール送信する場合を例に，メールアダプタおよびビジネスプロセスの設定例について説明します。

(a)　メールアダプタの設定例

開発環境でのメールアダプタの設定例，およびメールアダプタ実行環境プロパティファイルの設定例について説明します。

●開発環境でのメールアダプタの設定

開発環境でのサービスアダプタ定義画面の各項目は次のように設定します。なお，電文フォーマットおよび定義ファイルの格納先，開発環境での操作方法については，「3.2.12　メールアダプタを新規に追加する」を参照してください。

図 3-24　サービスアダプタ定義画面（基本）の設定例



表 3-106　サービスアダプタ定義画面（基本）の設定項目

分類	項目		設定内容
サービス部品制御情報	サービス名		mailadp
	サービス ID		mailadp
	サービス種別		メールアダプタ
	アドレス		－
	最大インスタンス数		0
	サービスクラス名		－
	オペレーション		SEND
オペレーション情報	オペレーション名		SEND
	通信モデル		同期
要求電文	any 型を使う（チェックボックス）		使わない（チェックしない）
	標準	使う（チェックボックス）	使わない（チェックしない）

分類	項目		設定内容
要求電文	標準	電文フォーマット	−
	サービス部品	電文フォーマット	adpmail_smtp_request.xsd
	データ変換定義		−
応答電文	any 型を使う（チェックボックス）		使わない（チェックしない）
	標準	使う（チェックボックス）	使わない（チェックしない）
		電文フォーマット	−
	サービス部品	電文フォーマット	adpmail_smtp_response.xsd
	データ変換定義		−
フォルト電文	−		−

（凡例）

−：設定しません。または該当する項目がありません。

図 3–25　サービスアダプタ定義画面（詳細）の設定例



表 3–107　サービスアダプタ定義画面（詳細）の設定項目

分類	項目	設定内容
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	cscmsg_adpejb.jar
	ユーティリティクラス（JAR ファイル）	adpmailpc.jar
	独自定義ファイル	−

（凡例）

−：設定しません。

●メールアダプタ実行環境プロパティファイルの作成

開発環境で定義したメールアダプタに対応するメールアダプタ実行環境プロパティファイルを作成します。ここでは，ファイル名を「mailadp.properties」としています。

なお，メールアダプタ実行環境プロパティファイルの作成方法については，「3.3.12(3)　メールアダプタの定義ファイルの作成」を参照してください。

メールアダプタ実行環境プロパティファイルの作成例を次に示します。斜体部分が変更個所です。

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

mailadp.smtp.host.name = mail-server
#mailadp.smtp.port = 25
#mailadp.user =
mailadp.addr.from = sampleuser2@XXXXXX.com
mailadp.addr.to = sampleuser@XXXXXX.com
#mailadp.addr.cc =
#mailadp.addr.bcc =
#mailadp.addr.count.max = 768
#mailadp.mail.body.size = 1024000
#mailadp.attach.path.1 =
#mailadp.attach.file.size = 5242880
#mailadp.max.attach.count.per.mail = 10
#mailadp.mail.format = text/plain
#mailadp.mail.body-text =
#mailadp.mail.body-filepath =
#mailadp.mail.body-filecharset = UTF-8
#mailadp.subject.size = 80
#mailadp.subject.text =
mailadp.smtp.encoding = ISO-2022-JP
#mailadp.smtp.auth = PLAIN
#mailadp.user.header.path =
#mailadp.account.file.path =
#mailadp.methodtrace.level = 3
#mailadp.methodtrace.filepath =
#mailadp.methodtrace.filenum =
#mailadp.methodtrace.filesize = 2097152
#mailadp.messagelog.level = 10
#mailadp.smtp.timeout = 180
#mailadp.smtp.connectiontimeout = 180
#mailadp.read-lock.retry.count = 0
#mailadp.read-lock.retry.interval = 1
```

作成したメールアダプタ実行環境プロパティファイルは，「<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¥config¥mail」フォルダに格納します。

(b)　ビジネスプロセスの設定例

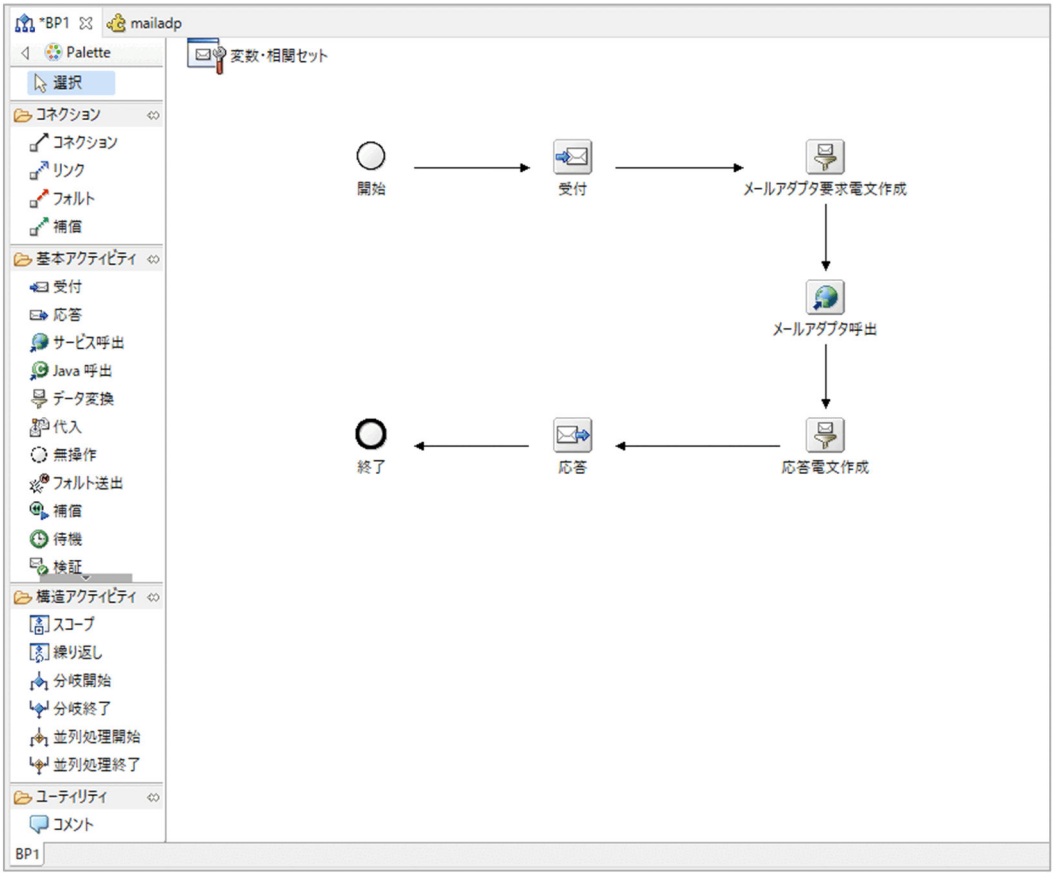
開発環境でのビジネスプロセスの設定例，およびデータ変換定義の設定例について説明します。

なお，開発環境で使用する画面の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「1. 画面（開発環境)」を参照してください。開発環境での操作方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5. ビジネスプロセスの定義」および「7. データ変換の定義」を参照してください。

●ビジネスプロセス定義画面の設定

開発環境のビジネスプロセス定義画面で次のようにアクティビティを定義します。

図 3-26　ビジネスプロセスの設定例



●変数の設定

ビジネスプロセス内の電文の受け渡しに必要なサービス部品電文用および標準電文用の変数を作成します。なお，この設定例では相関セットを使用しません。

- サービス部品電文用の変数の作成

  ビジネスプロセス定義画面の［変数・相関セット一覧］ダイアログで，次の変数を追加します。

表 3-108　サービス部品電文用の変数

変数名	説明
mail_input	メールアダプタの要求電文の変数
mail_output	メールアダプタの応答電文の変数

設定する種別はすべて［XML］です。また，それぞれの変数について［電文フォーマットの取込］ダイアログで次のとおりに設定します。

表 3-109 ［電文フォーマットの取込］ダイアログでの設定値

項目名	変数名	
	mail_input	mail_output
サービス/受付	サービス名	サービス名
サービス/受付の名称	mailadp	mailadp
オペレーション名	SEND	SEND
電文種別	要求電文（ボディ）	応答電文（ボディ）
電文フォーマット	mail_input	mail_output

- **標準電文用の変数の作成**

  次の名称および内容で標準電文の要求電文用と応答電文用の電文フォーマットを作成します。

  ・input.xsd

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="SampleSOAPinput">
    <xsd:complexType>
      <xsd:sequence>
       <xsd:element name="MAIL_BODY" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

  ・output.xsd

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="SampleSOAPoutput">
    <xsd:complexType>
      <xsd:sequence>
       <xsd:element name="MESSAGE_ID" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

  次に，ビジネスプロセス定義画面の［変数・相関セット一覧］ダイアログで，次の変数を追加します。

表 3-110 標準電文用の変数

変数名	説明
input	標準受付の要求電文の変数
output	標準受付の応答電文の変数

  設定する種別はすべて［XML］です。

  次に，それぞれの変数に対して［変数・相関セット一覧］ダイアログの［...］ボタンで，作成した電文フォーマットを次のとおりに指定します。

表 3-111　指定する電文フォーマット

変数名	指定する電文フォーマット
input	input.xsd
output	output.xsd

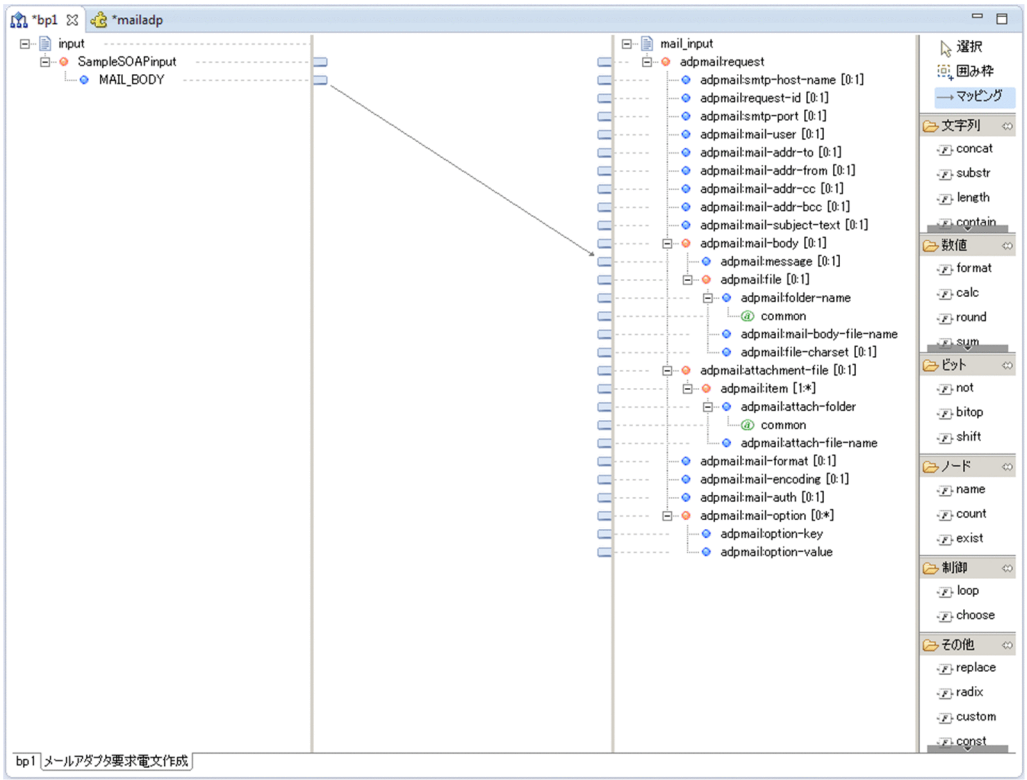
●データ変換定義の設定

データ変換アクティビティ（メールアダプタ要求電文作成）では，標準受付の要求電文からメールアダプタの要求電文を作成するマッピングをします。同様に，データ変換アクティビティ（応答電文作成）では，メールアダプタの応答電文から標準受付の応答電文を作成するマッピングをします。

まず，ビジネスプロセス定義画面でデータ変換アクティビティ（メールアダプタ要求電文作成）を右クリックし，［マッピング定義起動］を選択します。

標準受付の要求電文の MAIL_BODY 要素をメールアダプタの要求電文にある mail-body 要素の子要素の message 要素にマッピングします。

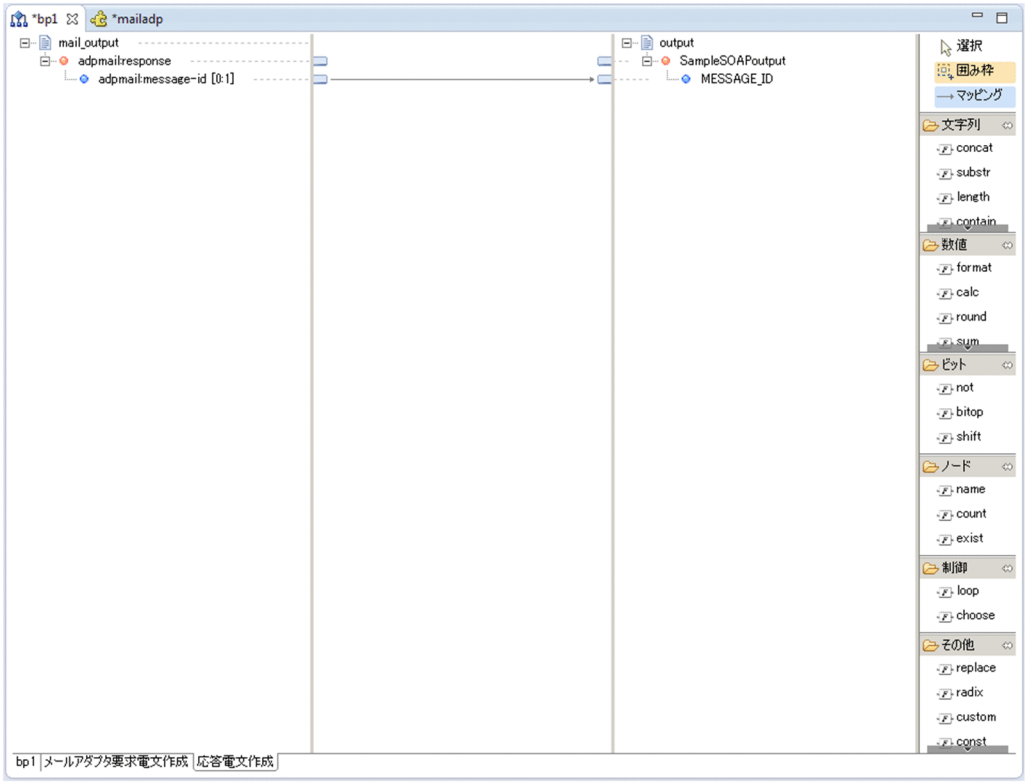
図 3-27　要求電文のマッピング



同様に，ビジネスプロセス定義画面でデータ変換アクティビティ（応答電文作成）を右クリックし，［マッピング定義起動］を選択します。

メールアダプタの応答電文にある message-id 要素を標準受付の応答電文の MESSAGE_ID 要素にマッピングします。

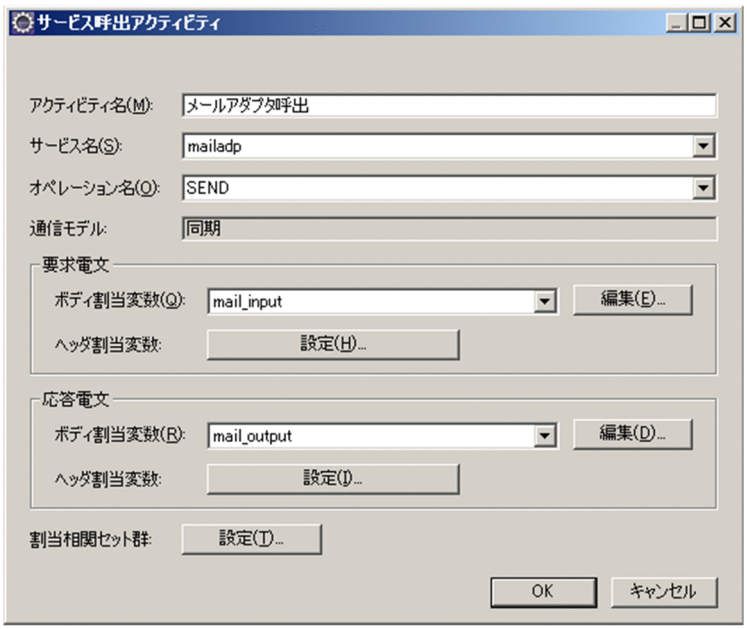
図 3-28　応答電文のマッピング



●サービス呼出アクティビティの設定

ビジネスプロセス定義画面のサービス呼出アクティビティ（メールアダプタ呼出）をダブルクリックし，表示される［サービス呼出アクティビティ］ダイアログで次のように設定します。

図 3-29　サービス呼出アクティビティの設定値



設定内容を次の表に示します。

表 3-112　サービス呼出アクティビティの設定内容

分類	項目	設定値
アクティビティ名		メールアダプタ呼出
サービス名		mailadp
オペレーション名		SEND
通信モデル		同期
要求電文	ボディ割当変数	mail_input
	ヘッダ割当変数	－
応答電文	ボディ割当変数	mail_output
	ヘッダ割当変数	－
割当相関セット群		－

（凡例）

－：設定しません。

これで，メールアダプタおよびビジネスプロセスの設定は完了です。

なお，この設定例でビジネスプロセスを実行するには，サービスリクエスタから標準受付に対して要求電文を指定してメッセージを送信する必要があります。サービスリクエスタの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「9.2　標準の同期受付（Web サービス）に要求を出すサービスリクエスタ（SOAP 通信基盤）」を参照してください。

そのあと，コンポーネントの検証とパッケージングをしてからテスト環境に配備定義して動作を確認してください。詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「8.　HCSC コンポーネントのパッケージングと配備定義」を参照してください。

3.3.13　HTTP アダプタを定義する

HTTP アダプタの定義方法について説明します。実際の設定例に基づいて HTTP アダプタの動作を確認したい場合は，「付録 G　HTTP 受付および HTTP アダプタを利用したビジネスプロセスの設定例」を参照してください。

(1)　J2EE サーバ用オプション定義ファイルの設定

J2EE サーバ用オプション定義ファイルの設定は，J2EE サーバの互換モードによって異なります。

- J2EE サーバの互換モードが推奨モードの場合

  実行環境で HTTP アダプタが動作するために，J2EE サーバのクラスパスに JAX-RS のライブラリ（jaxrs-impl.jar）を追加します。J2EE サーバ用オプション定義ファイル（usrconf.cfg）を編集し，次の内容を記述してください。

```
add.class.path=<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>¥CC¥javaee¥1100¥lib¥jaxrs-impl.jar
```

- J2EE サーバの互換モードが V9 互換モードの場合

  実行環境で HTTP アダプタが動作するために，J2EE サーバのクラスパスに JAX-RS のライブラリ（cjjaxrs.jar）を追加します。J2EE サーバ用オプション定義ファイル（usrconf.cfg）を編集し，次の内容を記述してください。

```
add.class.path=<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>¥jaxrs¥lib¥cjjaxrs.jar
```

J2EE サーバ用オプション定義ファイル（usrconf.cfg）の詳細は，マニュアル「アプリケーションサーバリファレンス 定義編(サーバ定義)」の「2.2.2　usrconf.cfg（J2EE サーバ用オプション定義ファイル)」を参照してください。

(2)　電文フォーマットの作成

ここでは，HTTP アダプタで使用する電文フォーマットの種類，形式，および作成方法について説明します。

なお，認証フローを使用してアクセストークンを取得する場合の要求電文および応答電文の設定については，マニュアル「サービスプラットフォーム 解説」の「2.16.17　OAuth 2.0 に対応したサービスへのアクセス」を参照してください。

(a)　電文フォーマットの種類

HTTP アダプタで使用する電文フォーマットの種類を次の表に示します。

表 3-113　電文フォーマットの種類

大分類	中分類	小分類	説明	編集可否
要求電文フォーマット	ヘッダ変数用（HTTP アダプタ用）	要求電文フォーマット（ヘッダ変数用）	HTTP リクエストヘッダに関する情報などを指定するヘッダ割当変数の電文フォーマットです。 サービスプラットフォームが提供する次のスキーマファイルを使用します。 • adphttp_header_request1.xsd Cookie 要素を一括設定する場合に使用します。 • adphttp_header_request2.xsd Cookie 要素を個別設定する場合に使用します。 どちらのスキーマファイルも adphttp_header_query_detail.xsd, および adphttp_header_http-header_detail.xsd を内包（include）します。	×
		クエリ用詳細電文フォーマット	クエリの情報を設定する電文フォーマットです。 サービスプラットフォームが提供するスキーマファイルは adphttp_header_query_detail.xsd です。 パラメタ設定部分を編集して使用します。	△

大分類	中分類	小分類	説明	編集可否
要求電文フォーマット	ヘッダ変数用（HTTPアダプタ用）	リクエストヘッダ用詳細電文フォーマット	リクエストヘッダの情報を設定する電文フォーマットです。 サービスプラットフォームが提供するスキーマファイルはadphttp_header_http-header_detail.xsd です。 パラメタ設定部分を編集して使用します。	△
	ヘッダ変数用（JSON-XML 変換用）	要求電文フォーマット（ヘッダ変数用）	JSON-XML 変換の定義を設定する電文フォーマットです。 サービスプラットフォームが提供する次のスキーマファイルを使用します。 • adp_custom_request.xsd	×
	ボディ変数用（HTTPアダプタ用）	ファイルデータ用電文フォーマット	ファイルデータだけ送信する場合など，HTTP リクエストのボディ部分を扱わない場合に設定する電文フォーマットです。 サービスプラットフォームが提供するスキーマファイルはadphttp_body_empty.fdx です。	×
		フォームデータ用電文フォーマット	HTTP リクエストボディにフォームデータを設定する場合に使用する電文フォーマットです。 サービスプラットフォームが提供するスキーマファイルはadphttp_body_form-data.xsd です。 OAuth 2.0 が対応する認証フローを使用して，アクセストークンを取得する場合は，サービスプラットフォームが提供する次のスキーマファイルを使用します。 • adphttp_body_form-data_jwt_assertion1.xsd OAuth 2.0 の認証フローとして JWT Bearer Token Flow を使用し，assertion を HTTP アダプタで作成する場合は，この電文フォーマットを使用します。 • adphttp_body_form-data_jwt_assertion2.xsd OAuth 2.0 の認証フローとして JWT Bearer Token Flow を使用し，HTTP アダプタ以外の発行者が作成した assertion を HTTP アダプタで使用する場合は，この電文フォーマットを使用します。 • adphttp_body_form-data_password.xsd OAuth 2.0 の認証フローとして Resource Owner Password Credentials Flow を選択する場合は，この電文フォーマットを使用します。 各スキーマファイルは，パラメタ設定部分を編集して使用します。	△
		パススルーモード用電文フォーマット	パススルーモードで任意の HTTP リクエストボディを送信したい場合に作成します。 サービスプラットフォームではスキーマファイルを提供していません。	○
	ボディ変数用（JSON-XML 変換用）	要求電文フォーマット（ボディ変数用）	JSON 変換ツールで作成した情報を設定した HTTP リクエストボディを送信したい場合に作成します。 サービスプラットフォームではスキーマファイルを提供していません。	○

大分類	中分類	小分類	説明	編集可否
応答電文フォーマット	ヘッダ変数用（HTTPアダプタ用）	応答電文フォーマット（ヘッダ変数用）	受信したHTTPレスポンスヘッダおよびステータスラインを割り当てるための電文フォーマットです。 サービスプラットフォームが提供する次のスキーマファイルを使用します。 • adphttp_header_response1.xsd Set-Cookie要素を一括設定する場合に使用します。 • adphttp_header_response2.xsd Set-Cookie要素を個別設定する場合に使用します。	×
	ボディ変数用（HTTPアダプタ用）	ファイルデータ用電文フォーマット	ファイルデータだけ受信する場合など，HTTPレスポンスのボディ部分を扱わない場合に設定する電文フォーマットです。 サービスプラットフォームが提供するスキーマファイルはadphttp_body_empty.fdxです。	×
		パススルーモード用電文フォーマット	パススルーモードで任意のHTTPレスポンスボディを受信したい場合に作成します。 サービスプラットフォームではスキーマファイルを提供していません。	○
	ボディ変数用（JSON-XML変換用）	応答電文フォーマット（ボディ変数用）	JSON変換ツールで作成した情報を設定したHTTPレスポンスボディを送信したい場合に作成します。 サービスプラットフォームではスキーマファイルを提供していません。	○
フォルト電文フォーマット			任意のステータスコードを受信したときにフォルト情報に割り当てるための電文フォーマットです。 サービスプラットフォームが提供するスキーマファイルはadphttp_fault.xsdです。	×

（凡例）

○：ユーザが作成します。

△：必要に応じて編集できます。

×：編集できません。編集した場合の動作は保証されません。

(b) 電文フォーマットの形式

ここでは，HTTPアダプタで使用する要求電文フォーマット，応答電文フォーマット，およびフォルト電文フォーマットの形式について説明します。

●要求電文フォーマット（ヘッダ変数用（HTTPアダプタ用））（要求電文フォーマット（ヘッダ変数用））

ビジネスプロセスからHTTPアダプタに渡すヘッダ変数の要求電文フォーマットを次に示します。この電文フォーマットのファイル名は「adphttp_header_request1.xsd」または「adphttp_header_request2.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/header_request」です。

表 3-114　要求電文フォーマット（ヘッダ変数用（HTTP アダプタ用））

タグ名	種別	出現回数	説明
<http-header-request>	−	1 回	−
<request-id>	string	0 または 1 回	受付で生成されたリクエスト ID を設定します。 作業フォルダ内のファイルデータを送信する場合や作業フォルダにファイルを受信する場合に必要です。
<method>※1	string	0 または 1 回	リクエストラインの HTTP メソッドを指定します。 指定できる値は次のどれかです。 • GET • HEAD • POST • OPTIONS • PUT • DELETE 上記以外の値を指定した場合の動作は保証されません。
<uri-scheme-authority>※1	string	0 または 1 回	リクエストラインの URI のスキームとオーソリティ（URI のポート番号まで）を指定します。値の URL エンコードは行われません。 URI を組み立てる際の最初の文字列となります。
<uri-path>※1	string	0 または 1 回	リクエストラインの URI のパス部分を指定します。 uri-scheme-authority 要素の直後に連結されるため，指定文字列の先頭は原則として「/」で始める必要があります。 ルートパス（"/"）の場合は指定を省略できます。
<uri-query>	string	※2	リクエストラインのクエリを指定します。
<http-header-Authorization>※1	string	0 または 1 回	HTTP リクエストヘッダの Authorization ヘッダに対応する認証情報を指定します。 type 属性には認証種別を指定します。 • 認証情報を送信する場合 type 属性に raw を指定し，この要素には Authorization ヘッダのヘッダフィールド値を指定します。 • 認証情報を送信しない場合 type 属性に none を指定します。この要素に値を指定しても無視されます。HTTP リクエストヘッダには Authorization ヘッダが設定されません。 • ベーシック認証の認証情報を送信する場合 type 属性に basic を指定し，この要素には HTTP アダプタアカウント定義ファイルに登録したユーザ名を指定します。 上記以外の値を type 属性に指定した場合はエラーになります。
<http-header-Content-Type>※1	string	0 または 1 回	HTTP リクエストヘッダの Content-Type ヘッダに対応するメディアタイプを指定します。 charset 属性には，Content-Type ヘッダの charset 属性に対応する文字コードを指定します。

タグ名		種別	出現回数	説明
	<http-header-Content-Type> ※1	string	0 または 1 回	HTTP ボディの型にマルチパート型（multipart/form-data）を指定した場合，Content-Type ヘッダには「multipart/form-data; boundary="<製品で生成した値>"」を指定します。
	<http-header-Cookies>	－(any)	0 または 1 回	HTTP リクエストの Cookie 情報を指定します。Cookie 要素を一括設定するか個別設定するかによって使用するスキーマファイルが異なります。 • Cookie 要素を一括設定する場合 スキーマファイルは adphttp_header_request1.xsd を使用します。any 型で Cookie 情報を指定します。 • Cookie 要素を個別設定する場合 スキーマファイルは adphttp_header_request2.xsd を使用します。Cookie 要素で個々の Cookie 情報を指定します。
	<Cookie>	string	0 回以上	HTTP リクエストの Cookie 情報を，Cookie 情報ごとに HTTP レスポンスから渡される Set-Cookie ヘッダの形式で指定します。 name 属性には，HTTP レスポンスの Set-Cookie ヘッダの Cookie 名を指定します。 host 属性には，HTTP レスポンスの Set-Cookie ヘッダの Cookie を送信するサーバのホスト（ドメイン）名を指定します。 path 属性には，HTTP レスポンスの Set-Cookie ヘッダの Cookie を送信するサーバのパスを指定します。 Cookie 要素の各属性値や要素の値は，引き継ぎ対象の Cookie 情報が格納された応答電文（ヘッダ）の Cookie 要素の値をそのまま使用してください。任意の値を設定した場合の動作は保証されません。
	<http-header-signed-aws-api-request>	string	0 または 1 回	署名付き AWS API リクエストで使用するアクセスキーペア ID を指定します。指定したアクセスキーペア ID をキーにしてアクセスキー ID，シークレットアクセスキーがアカウント定義ファイルから取得されます。 指定した文字列が，1 文字以上の半角英数字またはアンダースコア(_)からなる文字列でない場合は，エラーとなります。 指定できる属性を次に示します。 • type 属性 署名付き AWS API リクエストの署名情報を生成する場合は，「header」を指定します。「none」を指定した場合は，署名付き AWS API リクエストの署名情報は生成されません。「header」および「none」以外を指定した場合は，エラーとなります。 • version 属性 署名付き AWS API リクエストの署名バージョンに署名バージョン 4 を指定する場合は，「aws4_request」を指定

タグ名	種別	出現回数	説明
<http-header-signed-aws-api-request>	string	0 または 1 回	します。type 属性に「none」を指定した場合，この属性の指定は無視されます。 「aws4_request」以外を指定した場合は，エラーとなります。 • service 属性 接続先 AWS サービスを指定します。type 属性に「none」を指定した場合，この属性の指定は無視されます。存在しない AWS サービス名を指定した場合は，AWS サーバ上でエラーとなります。 Amazon S3 に接続する場合は，「s3」を指定します。 指定値が空の場合は，エラーとなります。 • region 属性※3 接続先 AWS リージョンを指定します。type 属性に「none」を指定した場合，この属性の指定は無視されます。存在しない接続先 AWS リージョンを指定した場合は，AWS サーバ上でエラーとなります。 指定値が空の場合は，エラーとなります。
<http-header>	any	※4	HTTP ヘッダ情報を指定します。
<http-body>	string	0 または 1 回	HTTP ボディの型を指定します。
<style>	string	0 または 1 回	HTTP ボディの型として次のどれかを指定します。 •「multipart/form-data」を指定した場合 マルチパート型の処理を行います。 •「single」または省略した場合 単一パートの送信処理を行います。 • 上記以外の値を指定した場合 システム例外が発生します。
<http-part>	−	0 または 1 回	パートのメタ情報を指定します。
<message>	−	0 または 1 回	要求電文（ボディ）に関する情報を指定します。
<binding>※1，※5	string	1 回	要求電文（ボディ）の処理方法として次のどれかを指定します。 • none 要求電文（ボディ）がない状態で HTTP リクエストを送信します。この値を指定した場合，HTTP リクエストの Content-Type ヘッダは設定されません。 • raw パススルーモードとして，要求電文（ボディ）の内容をそのまま送信します。HTTP リクエストの Content-Type ヘッダには，http-header-Content-Type 要素で指定したメディアタイプおよび文字コードが設定されます。 • form-data

タグ名					種別	出現回数	説明
			<binding>※1，※5		string	1 回	ノーマルモードとして，要求電文（ボディ）をフォームデータの形式に変換して送信します。HTTP リクエストの Content-Type ヘッダには，application/x-www-form-urlencoded が設定されます。文字コードには，http-header-Content-Type 要素で指定した文字コードが設定されます。 • 上記以外の場合 システム例外が発生します。
			<part-id>		string	0 または 1 回	要求電文（ボディ）をマルチパート型の 1 パートとして扱う場合，該当するパートに設定する Content-Disposition ヘッダの name 属性の値を指定します。
			<Content-Type>		string	0 または 1 回	要求電文（ボディ）をマルチパート型の 1 パートとして扱う場合，該当するパートに設定する Content-Type ヘッダのメディアタイプを指定します。 charset 属性には，Content-Type ヘッダの charset 属性に対応する文字コードを指定します。
		<files>			−	0 または 1 回	HTTP リクエストで送信するファイル情報を指定します。 この要素を指定した場合，要求電文（ボディ）が設定されていても，この要素での指定が優先されます。
			<file>		−	0 回以上	送信するファイルに関する情報を指定します。 この要素を 2 つ以上指定した場合は，最初に取得された要素 1 つが有効になります。
				<part-id>	string	0 または 1 回	送信対象ファイルをマルチパート型の 1 パートとして扱う場合，該当するパートに設定する Content-Disposition ヘッダの name 属性の値を指定します。
				<input-folder-name>※1	string	1 回	送信対象ファイルが格納されているフォルダとして，次のどちらかを指定します。 • 作業フォルダの場合 空文字を指定します（空文字列を指定してもエラーになりません）。 • 共通フォルダの場合 共通フォルダ定義名を指定します。 また，common 属性には，送信対象ファイルが存在するフォルダを識別するための次の値を指定します。 • 作業フォルダの場合 「common="false"」を指定します。 • 共通フォルダの場合 「common="true"」を指定します。
				<local-file-name>※1	string	1 回	送信対象のファイル名を指定します。 • input-folder-name 要素で作業フォルダを指定した場合 作業フォルダ直下の中間ファイル名を指定します。

タグ名					種別	出現回数	説明
				<local-file-name>※1	string	1 回	パス変更に関係する指定（/../など）が含まれる場合，エラーになります。 • input-folder-name 要素で共通フォルダを指定した場合 共通フォルダルート配下のファイル名を指定します。 共通フォルダルートより上位のパスを指定した場合，エラーになります。 相対パスの先頭のファイル区切り文字（/）は省略できます。 サブフォルダを使用する場合は，共通フォルダルートからファイル名までの相対パスを指定してください。 指定した名称のファイルが存在しない場合はシステム例外が発生します。
				<file-name>	string	0 または 1 回	該当するパートに設定する Content-Disposition ヘッダの filename 属性の値を指定します。
				<Content-Type>	string	0 または 1 回	送信対象ファイルをマルチパート型の 1 パートとして扱う場合，該当するパートに設定する Content-Type ヘッダのメディアタイプを指定します。 charset 属性には，Content-Type ヘッダの charset 属性に対応する文字コードを指定します。
	<output-folder-name>※1				string	0 または 1 回	HTTP レスポンス時にダウンロードファイルを出力するフォルダとして，次のどちらかを指定します。 • 作業フォルダの場合 空文字を指定します（空文字列を指定してもエラーになりません）。 • 共通フォルダの場合 共通フォルダ定義名を指定します。 また，common 属性には，送信対象ファイルが存在するフォルダを識別するための次の値を指定します。 • 作業フォルダの場合 「common="false"」を指定します。 • 共通フォルダの場合 「common="true"」を指定します。
	<output-sub-folder>				string	0 または 1 回	HTTP レスポンス時にダウンロードファイルを出力する共通フォルダのサブフォルダを設定します。 • 共通フォルダルートからサブフォルダまでの相対パスを指定します。 • 共通フォルダルートより上位のパスを指定した場合，実行時にエラーになります。 • 共通フォルダルートからの相対パスで指定したサブフォルダが存在しない場合，サブフォルダを作成します。 • 相対パスの先頭および末尾のファイル区切り文字（/）は省略できます。 • output-folder-name 要素で，common 属性に true 以外の値を指定した場合，この要素の値は無視されます。

（凡例）

－：該当する項目はありません。

注※1

要素の指定がない場合は，HTTP アダプタ実行環境プロパティファイルまたは HTTP アダプタ実行環境共通プロパティファイルの指定が有効になります。要求電文とプロパティファイルの対応関係については，マニュアル「サービスプラットフォーム 解説」の「2.16.14　要求電文（ヘッダ）と HTTP アダプタ実行環境（共通）プロパティファイルのパラメタの対応関係」を参照してください。

注※2

include 先のクエリ用詳細電文フォーマットの指定に従います。

注※3

region 属性に指定できる接続先 AWS リージョンを次に示します。

表 3–115　region 属性に指定できる接続先 AWS リージョン

指定値	接続先 AWS リージョン
us-east-2	米国東部（オハイオ）
us-east-1	米国東部（バージニア北部）
us-west-1	米国西部（北カリフォルニア）
us-west-2	米国西部（オレゴン）
af-south-1	アフリカ（ケープタウン）
ap-east-1	アジアパシフィック（香港）
ap-south-2	アジアパシフィック（ハイデラバード）
ap-southeast-3	アジアパシフィック（ジャカルタ）
ap-southeast-4	アジアパシフィック（メルボルン）
ap-south-1	アジアパシフィック（ムンバイ）
ap-northeast-3	アジアパシフィック（大阪）
ap-northeast-2	アジアパシフィック（ソウル）
ap-southeast-1	アジアパシフィック（シンガポール）
ap-southeast-2	アジアパシフィック（シドニー）
ap-northeast-1	アジアパシフィック（東京）
ca-central-1	カナダ（中部）
ca-west-1	カナダ西部（カルガリー）
eu-central-1	欧州（フランクフルト）
eu-west-1	欧州（アイルランド）
eu-west-2	欧州（ロンドン）
eu-south-1	欧州（ミラノ）
eu-west-3	欧州（パリ）

指定値	接続先 AWS リージョン
eu-south-2	欧州（スペイン）
eu-north-1	欧州（ストックホルム）
eu-central-2	欧州（チューリッヒ）
il-central-1	イスラエル（テルアビブ）
me-south-1	中東（バーレーン）
me-central-1	中東（アラブ首長国連邦）
sa-east-1	南米（サンパウロ）
us-gov-east-1	AWS GovCloud（米国東部）
us-gov-west-1	AWS GovCloud（米国西部）

注※4

include 先のリクエストヘッダ用詳細フォーマットの指定に従います。

注※5

JSON-XML 変換をする場合，HTTP アダプタ実行環境プロパティファイルの adphttp.request.part.message.binding プロパティで「raw」以外の値を設定しているときは，binding 要素の値に「raw」を指定してください。http-header-Content-type 要素の値については，「application/json」の設定を推奨します。

●要求電文フォーマット（ヘッダ変数用（HTTP アダプタ用））（クエリ用詳細電文フォーマット）

要求電文フォーマット（ヘッダ変数用）に内包（include）されるクエリ用詳細電文フォーマットを次に示します。この電文フォーマットのファイル名は「adphttp_header_query_detail.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/header_request」です。

表 3-116　クエリ用詳細電文フォーマット（ヘッダ変数用（HTTP アダプタ用））

タグ名		種別	出現回数	説明
<uri-query>		－	0 または 1 回	－
	任意の要素※	string	0 から 1,024 回	リクエストラインのクエリの詳細情報を設定します。

（凡例）

－：該当する項目はありません。

注※

要素の指定がない場合は，HTTP アダプタ実行環境プロパティファイルまたは HTTP アダプタ実行環境共通プロパティファイルの指定が有効になります。要求電文とプロパティファイルの対応関係については，マニュアル「サービスプラットフォーム 解説」の「2.16.14　要求電文（ヘッダ）と HTTP アダプタ実行環境（共通）プロパティファイルのパラメタの対応関係」を参照してください。

●要求電文フォーマット（ヘッダ変数用（HTTP アダプタ用））（リクエストヘッダ用詳細電文フォーマット）

要求電文フォーマット（ヘッダ変数用）に内包（include）されるリクエストヘッダ用詳細電文フォーマットを次に示します。この電文フォーマットのファイル名は「adphttp_header_http-

header_detail.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/header_request」です。

表 3-117　リクエストヘッダ用詳細電文フォーマット（ヘッダ変数用（HTTP アダプタ用））

タグ名		種別	出現回数	説明
<http-header>		−	0 または 1 回	−
	任意の要素※	string	0 から 1,024 回	リクエストヘッダ情報を設定します。 ヘッダフィールド名に要素名，ヘッダフィールド値に要素値を設定してください。階層構造や属性値は指定できません。 リクエストヘッダの詳細については，マニュアル「サービスプラットフォーム 解説」の「2.16.3(3)　HTTP リクエストヘッダ」を参照してください。

（凡例）

−：該当する項目はありません。

注※

要素の指定がない場合は，HTTP アダプタ実行環境プロパティファイルまたは HTTP アダプタ実行環境共通プロパティファイルの指定が有効になります。要求電文とプロパティファイルの対応関係については，マニュアル「サービスプラットフォーム 解説」の「2.16.14　要求電文（ヘッダ）と HTTP アダプタ実行環境（共通）プロパティファイルのパラメタの対応関係」を参照してください。

●要求電文フォーマット（ヘッダ変数用（JSON-XML 変換用））

JSON-XML 変換の定義を設定するヘッダ変数用の要求電文フォーマットを次に示します。この電文フォーマットのファイル名は，「adp_custom_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/custom/request」です。

表 3-118　要求電文フォーマット（ヘッダ変数用（JSON-XML 変換用））

タグ名		種別	出現回数※1	説明
<custom-adapter-request>		−	1 回	−
	<json-transfer>	boolean	0 または 1 回	HTTP アダプタのリクエスト処理時に，JSON-XML 変換するかどうかを指定するタグです。指定値は次のどちらかです。 true：JSON-XML 変換します。※2 false：JSON-XML 変換しません。 • タグがあって値が「true」または「false」の場合 指定した値が適用されます。 • タグがあって値が「true」または「false」のどちらでもない場合 システム例外を返します。 • タグがあって値がない場合 デフォルトの「false」が使用されます。

タグ名		種別	出現回数※1	説明
	<json-transfer>	boolean	0 または 1 回	• タグがない場合 HTTP アダプタ定義ファイルの adphttp.request.switchover.json-transfer.mode プロパティの値が有効になります。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

HTTP アダプタ定義ファイルの adphttp.request.switchover.json-transfer.mode プロパティの値に「true」が設定されていない場合，JSON-XML 変換をしません。

●要求電文フォーマット（ボディ変数用（HTTP アダプタ用））（ファイルデータ用電文フォーマット）

空の HTTP メッセージボディを表すファイルデータ用電文フォーマットの形式を次に示します。この電文フォーマットのファイル名は「adphttp_body_empty.fdx」です。

表 3-119　ファイルデータ用電文フォーマット（ボディ変数用（HTTP アダプタ用））

単純内容要素		種別	出現回数	説明
<http-body-common-empty>		－	1 回	－
	<data>	string	0 回以上	空のデータを渡す要素です。この要素は省略できます。

（凡例）

－：該当する項目はありません。

●要求電文フォーマット（ボディ変数用（HTTP アダプタ用））（フォームデータ用電文フォーマット）

ビジネスプロセスから HTTP アダプタに渡すフォームデータ用電文フォーマットを次に示します。この電文フォーマットのファイル名は「adphttp_body_form-data.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/body_form_data」です。

表 3-120　フォームデータ用電文フォーマット（ボディ変数用（HTTP アダプタ用））

タグ名		種別	出現回数	説明
<http-body-form-data>		－	1 回	－
	任意の要素	string	0 回以上	任意の要素を指定します。

（凡例）

－：該当する項目はありません。

●要求電文フォーマット（ボディ変数用（HTTP アダプタ用））（フォームデータ用電文フォーマット）（OAuth 2.0 の認証フローとして assertion を HTTP アダプタで作成する場合）

OAuth 2.0 の認証フローとして JWT Bearer Token Flow を使用し，assertion を HTTP アダプタで作成する場合の電文フォーマットを次に示します。この電文フォーマットのファイル名は

「adphttp_body_form-data_jwt_assertion1.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/body_form_data」です。

表 3-121　フォームデータ用電文フォーマット（ボディ変数用（HTTP アダプタ用））（OAuth 2.0 の認証フローとして assertion を HTTP アダプタで作成する場合）

タグ名	種別	出現回数	説明
<http-body-form-data>	−	1 回	−
<grant_type>	string	1 回	OAuth 2.0 の認証フロー（JWT Bearer Token Flow）を使用する場合に，グラントタイプに固定の文字列「urn:ietf:params:oauth:grant-type:jwt-bearer」を設定します。
任意の要素	string	0 または 1 回	任意の要素を指定します。
<auth-info>※1，※2	−	1 回	auth-info 配下のタグに認証情報を設定します。
<claim>	−	1 回	−
<iss>	string	0 または 1 回	JWT 発行者の識別子を設定します。
<aud>	string	0 または 1 回	JWT 利用者の識別子を設定します。
<sub>	string	0 または 1 回	JWT 発行者のユーザ識別子を設定します。
<exp>	string	0 または 1 回	JWT の有効期限を設定します。
<nbf>	string	0 または 1 回	JWT が有効になる日時を設定します。
<iat>	string	0 または 1 回	JWT を発行した時刻を設定します。
<jti>	string	0 または 1 回	JWT のための一意な識別子を設定します。
任意の要素	string	0 または 1 回	任意の要素を指定します。
<header>	−	1 回	−
<alg>	string	1 回	JWT クレームセットの暗号化に使用するアルゴリズム（JWA RFC7518 で定義された設定値）を設定します。
任意の要素	string	0 または 1 回	任意の要素を指定します。
<secretkey>	string	1 回	公開鍵認証情報登録・更新コマンドで登録した鍵 ID を設定します。※3 鍵 ID に対応する秘密鍵ファイルを署名時に使用します。

タグ名			種別	出現回数	説明
		<signature>	string	1 回	電子署名に使用する Java 暗号化アーキテクチャ標準アルゴリズム名（Signature アルゴリズム）を設定します。
		<keytype>	string	1 回	秘密鍵を変換する Java 暗号化アーキテクチャ標準アルゴリズム名（KeyFactory アルゴリズム）を設定します。 秘密鍵が対応するアルゴリズムを設定します。

（凡例）

－：該当する項目はありません。

注※1

設定が必要な項目および設定値は認可サーバによって異なるため，接続する認可サーバに合わせて適切な設定をする必要があります。

注※2

HTTP アダプタは，grant_type 要素の設定値が urn:ietf:params:oauth:grant-type:jwt-bearer で，かつ assertion 要素が設定されていない場合，設定された認証情報を基に，フォームデータとして assertion を作成します。

assertion は，次の形式で作成されます。

a.b.c

（凡例）

a：JSON 形式のヘッダ情報を Base64 変換した文字列

b：JSON 形式のクレーム情報を Base64 変換した文字列

c：a と b をピリオド（.）で連結した文字列に，電子署名で使用する暗号化キーのタイプ，アルゴリズム，および秘密鍵を使用して署名し，Base64 変換した文字列

作成された assertion は，ほかのフォームデータと同様に，フォームデータとして送信されます。

注※3

公開鍵認証情報登録・更新コマンドについては，マニュアル「サービスプラットフォーム リファレンス」の「csakeyadd（公開鍵認証の秘密鍵情報の登録・更新）」を参照してください。

●要求電文フォーマット（ボディ変数用（HTTP アダプタ用））（フォームデータ用電文フォーマット）（OAuth 2.0 の認証フローとして HTTP アダプタ以外の発行者が作成した assertion を HTTP アダプタで使用する場合）

OAuth 2.0 の認証フローとして JWT Bearer Token Flow を使用し，HTTP アダプタ以外の発行者が作成した assertion を HTTP アダプタで使用する場合の電文フォーマットを次に示します。この電文フォーマットのファイル名は「adphttp_body_form-data_jwt_assertion2.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/body_form_data」です。

表 3-122　フォームデータ用電文フォーマット（ボディ変数用（HTTP アダプタ用））（OAuth 2.0 の認証フローとして HTTP アダプタ以外の発行者が作成した assertion を HTTP アダプタで使用する場合）

タグ名		種別	出現回数	説明
<http-body-form-data>		－	1 回	－
	<grant_type>	string	1 回	OAuth 2.0 の認証フロー（JWT Bearer Token Flow）を使用する場合に，グラントタイプに固定の文字列

タグ名		種別	出現回数	説明
	<grant_type>	string	1 回	「urn:ietf:params:oauth:grant-type:jwt-bearer」を設定します。
	<assertion>	string	1 回	JSON Web Token を設定します。 assertion を HTTP アダプタで作成しない場合に，発行者が作成した JSON Web Token を assertion に設定します。
	任意の要素	string	0 または 1 回	任意の要素を指定します。

（凡例）

－：該当する項目はありません。

●要求電文フォーマット（ボディ変数用（HTTP アダプタ用））（フォームデータ用電文フォーマット）（OAuth 2.0 の認証フローとして Resource Owner Password Credentials Flow を選択する場合）

OAuth 2.0 の認証フローとして Resource Owner Password Credentials Flow を選択する場合の電文フォーマットを次に示します。この電文フォーマットのファイル名は「adphttp_body_form-data_password.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/body_form_data」です。

表 3-123　フォームデータ用電文フォーマット（ボディ変数用（HTTP アダプタ用））（OAuth 2.0 の認証フローとして Resource Owner Password Credentials Flow を選択する場合）

タグ名		種別	出現回数	説明
<http-body-form-data>		－	1 回	－
	<grant_type>	string	1 回	OAuth 2.0 の認証フローとして Resource Owner Password Credentials Flow を選択する場合に，グラントタイプに固定の文字列「password」を設定します。
	<username>	string	1 回	リソースオーナーのユーザ名を設定します。 パスワード認証情報登録・更新コマンドで登録したユーザ名を指定します。※
	任意の要素	string	0 または 1 回	任意の要素を指定します。

（凡例）

－：該当する項目はありません。

注※

パスワード認証情報登録・更新コマンドについては，マニュアル「サービスプラットフォーム リファレンス」の「csauseradd（パスワード認証のユーザ情報の登録・更新）」を参照してください。

●応答電文フォーマット（ヘッダ変数用（HTTP アダプタ用））（応答電文フォーマット（ヘッダ変数用））

HTTP アダプタが呼び出し元のビジネスプロセスに返すヘッダ変数の応答電文フォーマットを次に示します。この電文フォーマットのファイル名は「adphttp_header_response1.xsd」または「adphttp_header_response2.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/header_response」です。

表 3-124　応答電文フォーマット（ヘッダ変数用）（HTTP アダプタ用）

タグ名	種別	出現回数	説明
<http-header-response>	−	1 回	−
<request-id>	string	0 または 1 回	要求電文で指定したリクエスト ID が設定されます。 要求電文で指定しなかった場合，この要素は生成されません。
<status-code>	string	1 回	HTTP レスポンスのステータスラインのステータスコードが設定されます。
< reason-phrase>	string	1 回	ステータスラインの理由フレーズ部分が設定されます。
<http-header-Content-Type>	string	0 または 1 回	Content-Type ヘッダのメディアタイプが設定されます。
<http-header-Cookies>	− (any)	1 回	HTTP レスポンスの Set-Cookie ヘッダが設定されます。Cookie 要素を一括設定するか個別に設定するかによって使用するスキーマファイルが異なります。 • Cookie 要素を一括設定する場合 スキーマファイルは adphttp_header_response1.xsd を使用します。Cookie 情報は any 型で一括設定されます。 • Cookie 要素を個別設定する場合 スキーマファイルは adphttp_header_response2.xsd を使用します。Cookie 要素に個々の Cookie 情報が設定されます。
<Cookie>	string	0 回以上	HTTP レスポンスの Set-Cookie ヘッダが，Cookie 情報ごとに格納されます。 name 属性には，HTTP レスポンスの Set-Cookie ヘッダの Cookie 名が設定されます。 host 属性には，HTTP レスポンスの Set-Cookie ヘッダの Cookie を送信するサーバのホスト（ドメイン）名が設定されます。 path 属性には，HTTP レスポンスの Set-Cookie ヘッダの Cookie を送信するサーバのパスが設定されます。
<http-header>	−	1 回	HTTP レスポンスヘッダが格納されます。
任意の要素	any	0 回以上	HTTP ヘッダ情報が，要素名にヘッダフィールド名，要素値にヘッダフィールド値の形式で設定されます。
<http-body>	−	1 回	HTTP レスポンスボディの情報が設定されます。
<output-folder-name>	string	0 または 1 回	HTTP レスポンスでダウンロードしたファイルの出力先フォルダが設定されます。 出力先フォルダは要求電文フォーマット（ヘッダ変数用）の output-folder-name 要素で指定した値と同じです。 • 作業フォルダの場合 空文字が設定されます。 • 共通フォルダの場合 共通フォルダ定義名が指定されます。

タグ名	種別	出現回数	説明
<output-folder-name>	string	0 または 1 回	また，common 属性には，送信対象ファイルが存在するフォルダを識別するための値が設定されます。 • 作業フォルダの場合 「common="false"」が設定されます。 • 共通フォルダの場合 「common="true"」が設定されます。 ダウンロードするファイルがない場合，この要素は生成されません。
<http-part>	−	1 回	パートのメタ情報が設定されます。
<files>	−	0 または 1 回	HTTP レスポンスで受信するファイル情報が設定されます。
<file>	−	1 回以上	受信するファイルに関する情報が格納されます。
<part-id>	string	0 または 1 回	Content-Disposition ヘッダの name 属性に指定されている値が設定されます。 name 属性がない場合，この要素は生成されません。
<local-file-name>	string	1 回	共通フォルダまたは作業フォルダに出力されたファイル名が設定されます。 • 出力フォルダが共通フォルダの場合 HTTP アダプタ内で一意となる名称で出力されたファイル名が設定されます。 サブフォルダに出力する場合は，共通フォルダルートからの相対パスが設定されます。 • 出力フォルダが作業フォルダの場合 出力された中間ファイル名が設定されます。
<file-name>	string	0 または 1 回	Content-Disposition ヘッダの filename 属性に指定されている値が設定されます。 filename 属性がない場合，この要素は生成されません。
<Content-Type>	string	0 または 1 回	HTTP レスポンスでダウンロードしたファイルの MIME タイプを設定します。 省略した場合は「text/plain」を設定します。 charset 属性には，ダウンロードしたファイルの Content-Type ヘッダの charset 属性に対応する文字コードを指定します。
<context>	−	1 回	HTTP アダプタの実行情報が格納されます。
<adapter-name>	string	1 回	この HTTP レスポンスの基になった HTTP リクエストを送信したアダプタの名称が設定されます。
<operation-name>	string	1 回	この HTTP レスポンスの基になった HTTP リクエストを送信したオペレーションの名称が設定されます。

タグ名	種別	出現回数	説明
<request-method>	string	1 回	この HTTP レスポンスの基になった HTTP リクエストのメソッド種別が設定されます。
<request-uri>	string	1 回	この HTTP レスポンスの基になった HTTP リクエストの URI（Scheme および UserInfo を除く Authorization，Path）が設定されます。
<access-token>	string	0 または 1 回	OAuth 2.0 の認証フロー実行時に認可サーバから返されるアクセストークンが設定されます。

（凡例）

－：該当する項目はありません。

●フォルト電文フォーマット

HTTP アダプタが呼び出し元のビジネスプロセスに返すフォルト電文フォーマットを次に示します。この電文フォーマットのファイル名は「adphttp_fault.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/fault」です。

表 3-125　フォルト電文フォーマット

タグ名	種別	出現回数	説明
<http-fault>	－	1 回	－
<status-code>	string	1 回	HTTP レスポンスのステータスラインのステータスコードが設定されます。
< reason-phrase>	string	1 回	ステータスラインの理由フレーズ部分が設定されます。
<http-header>	－	1 回	HTTP レスポンスヘッダが格納されます。
任意の要素	string	0 回以上	HTTP ヘッダ情報が，要素名にヘッダフィールド名，要素値にヘッダフィールド値の形式で設定されます。
<context>	－	1 回	HTTP アダプタの実行情報が格納されます。
<adapter-name>	string	1 回	この HTTP レスポンスの基になった HTTP リクエストを送信したアダプタの名称が設定されます。
<operation-name>	string	1 回	この HTTP レスポンスの基になった HTTP リクエストを送信したオペレーションの名称が設定されます。
<request-method>	string	1 回	この HTTP レスポンスの基になった HTTP リクエストのメソッド種別が設定されます。
<request-uri>	string	1 回	この HTTP レスポンスの基になった HTTP リクエストの URI（Scheme および UserInfo を除く Authorization，Path）が設定されます。

（凡例）

－：該当する項目はありません。

(c)　電文フォーマットの作成方法

HTTP アダプタで使用する電文フォーマットの作成方法について説明します。

サービスプラットフォームが提供する電文フォーマットの XML スキーマは，「<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¥custom-adapter¥HTTP¥schema」に格納されています。編集する場合はフォルダごと任意の場所にコピーしてから編集してください。

次の電文フォーマットの作成方法を次に示します。

- クエリ用詳細電文フォーマット
- リクエストヘッダ用詳細電文フォーマット
- フォームデータ用電文フォーマット
- パススルーモード用電文フォーマット
- ボディ変数用（JSON-XML 変換用）電文フォーマット

●クエリ用詳細電文フォーマットの作成

1. XML スキーマファイル（adphttp_header_query_detail.xsd）を XML エディタなどで開き，次のように編集します。

●修正前のXMLスキーマファイル

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  targetNamespace="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/header_request"
  elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/header_request">
  <xsd:complexType name="http-header-type">
    <xsd:sequence>
      <xsd:any namespace="##any" processContents="skip" minOccurs="0"
       maxOccurs="1024" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

●修正後のXMLスキーマファイル（例）

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  targetNamespace="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/header_request"
  elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/header_request">
  <xsd:complexType name="http-header-type">
    <xsd:sequence>
      <xsd:element name="cache-control" type="xsd:string" minOccurs="0"/>
      <xsd:element name="host" type="xsd:string" minOccurs="0"/>
      <xsd:element name="accept-language" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

（凡例）

：編集個所

xsd:any 型の要素を，クエリ名をローカル名とする xsd:string 型の要素に変更します。出現回数は任意に指定できます。

注意事項

uri-query 要素直下に指定できるクエリ要素の上限は 1,024 個までです。1,025 個以上の要素を指定した場合の動作は保証されません。

2. XML スキーマファイルを上書き保存します。

●リクエストヘッダ用詳細電文フォーマットの作成

1. XML スキーマファイル（adphttp_header_http-header_detail.xsd）を XML エディタなどで開き，次のように編集します。

●修正前のXMLスキーマファイル

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
 targetNamespace="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/
body_form_data"
 elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/body_form_data">
 <xsd:element name="http-body-form-data" type="tns:http-body-form-data-type"></xsd:element>
  <xsd:complexType name="http-header-type">
   <xsd:sequence>
     <xsd:any namespace="##any" processContents="skip" minOccurs="0" maxOccurs="1024" />
   </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

●修正後のXMLスキーマファイル（例）

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
 targetNamespace="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/
body_form_data"
 elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/body_form_data">
 <xsd:element name="http-body-form-data" type="tns:http-body-form-data-type"></xsd:element>
  <xsd:complexType name="http-header-type">
   <xsd:sequence>
     <xsd:element name="cache-control" type="xsd:string" minOccurs="0"/>
     <xsd:element name="host" type="xsd:string" minOccurs="0"/>
     <xsd:element name="accept-language" type="xsd:string" minOccurs="0"/>
   </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

（凡例）

：編集個所

xsd:any 型の要素を，リクエストヘッダ名をローカル名とする xsd:string 型の要素に変更します。出現回数は任意に指定できます。

注意事項

http-header 要素直下に指定できるリクエストヘッダ要素の上限は 1,024 個までです。1,025 個以上の要素を指定した場合の動作は保証されません。

2. XML スキーマファイルを上書き保存します。

●フォームデータ用電文フォーマットの作成

1. XML スキーマファイル（adphttp_body_form-data.xsd）を XML エディタなどで開き，次のように編集します。

●修正前のXMLスキーマファイル

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
 targetNamespace="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/
body_form_data"
 elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/body_form_data">
 <xsd:element name="http-body-form-data" type="tns:http-body-form-data-type"></xsd:element>
  <xsd:complexType name="http-body-form-data-type">
   <xsd:sequence>
    <xsd:any namespace="##any" processContents="skip" minOccurs="0" maxOccurs="unbounded" />
   </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

●修正後のXMLスキーマファイル（例）

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
 targetNamespace="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/
body_form_data"
 elementFormDefault="qualified" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/body_form_data">
 <xsd:element name="http-body-form-data" type="tns:http-body-form-data-type"></xsd:element>
  <xsd:complexType name="http-body-form-data-type">
   <xsd:sequence>
    <xsd:element name="parameter1" type="xsd:string" minOccurs="0"/>
    <xsd:element name="parameter2" type="xsd:string" minOccurs="0"/>
   </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

（凡例）

：編集個所

xsd:any 型の要素を，フォームデータのキー名をローカル名とする xsd:string 型の要素に変更します。出現回数は任意に指定できます。

2. XML スキーマファイルを上書き保存します。

●パススルーモード用電文フォーマットの作成

1. エディタなどで任意の電文フォーマットを作成します。
2. 任意のファイル名で保存します。

注意事項

- 変数の型は，XML 型，non-XML 型，any 型のどれでも指定できます。ただし，XML 型の場合は XML フォーマット定義ファイル，non-XML 型の場合はバイナリフォーマット定義ファイルの電文フォーマットを作成する必要があります。
- HTTP リクエストボディの文字コードは Content-Type ヘッダの charset 属性で指定できますが，HTTP アダプタでは指定された文字コードに変換しません。このため，ボディ

データは要求電文（ボディ）に設定する前にあらかじめユーザ自身で文字コード変換をする必要があります。

●ボディ変数用（JSON-XML 変換用）電文フォーマット

ボディ変数用の電文フォーマットは，JSON 変換ツールで作成します。JSON 形式の要求電文，および応答電文のサンプルファイルを入力 JSON ファイルとして用意し，JSON 変換ツールで XSD ファイルを作成してください。HTTP アダプタで JSON-XML 変換の動作を変更する場合は，JSON-XML 変換定義ファイルを JSON 変換ツールの実行時に指定してください。

JSON 変換ツールの実行形式を次に示します。

```
cscjson2xsd -in 〈入力JSONファイルのパス〉 -out 〈出力XSDファイルのパス〉[-f 〈JSON-XML変換定義ファイルのパス〉]
```

JSON 変換ツールの詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「14. JSON 変換ツールを使用したデータ変換」を参照してください。

(3)　サービスアダプタ定義画面での操作

HTTP アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。

   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービスアダプタ定義画面（基本）で定義情報を設定します。

   サービスアダプタ定義画面（基本）で設定が必要な項目については，「3.3.19(13)　HTTP アダプタの場合」およびマニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

3. [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画面（詳細）が表示されます。

4. サービスアダプタ定義画面（詳細）で定義情報を設定します。

   サービスアダプタ定義画面（詳細）で設定が必要な項目については，「3.3.19(13)　HTTP アダプタの場合」を参照してください。

(4)　HTTP アダプタの定義ファイルの作成および編集

作成および編集する定義ファイルの種類を次に示します。

- HTTP アダプタ定義ファイル

  HTTP アダプタの動作情報を設定するためのファイルです。
- HTTP アダプタ実行環境プロパティファイル

  HTTP アダプタごとの構成情報を設定するためのファイルです。

- HTTP アダプタ実行環境共通プロパティファイル
  すべての HTTP アダプタに共通の構成情報を設定するためのファイルです。
- JSON-XML 変換定義ファイル
  JSON 形式データと XML 形式データとの間の変換に関する動作を設定するためのファイルです。

なお，これらの定義ファイルの作成および編集は任意です。省略した場合は各定義ファイルのデフォルト値が使用されます。

各定義ファイルの作成手順および編集手順を次に示します。

(a) HTTP アダプタ定義ファイル

HTTP アダプタ定義ファイルは，サービスプラットフォームが提供するテンプレートファイルを利用して編集します。

HTTP アダプタ定義ファイルの編集手順を示します。

1. サービスアダプタ定義画面（詳細）の［独自定義ファイル］で「cscadphttp.properties」を選択し，［編集］ボタンをクリックします。
   HTTP アダプタ定義ファイルを編集するためのエディタが起動します。
2. エディタ上で HTTP アダプタ定義ファイルを編集します。
   HTTP アダプタ定義ファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「3.10.1　HTTP アダプタ定義ファイル」を参照してください。
3. Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(b) HTTP アダプタ実行環境プロパティファイル

1. テンプレートファイル（<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¥custom-adapter¥HTTP¥config¥templates¥serviceid.properties）をコピーして，次のディレクトリに格納します。

   全 HCSC サーバ共通の定義の場合

   <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>¥CSC¥custom-adapter¥HTTP¥config

   特定の HCSC サーバ用の定義の場合

   サブディレクトリ<HCSC サーバ名>を作成して格納してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

   <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¥custom-adapter¥HTTP¥config¥<HCSC サーバ名>

2. コピーしたテンプレートファイルのファイル名を「<サービス ID※>.properties」に変更します。
3. 定義内容を編集して保存します。

HTTP アダプタ実行環境プロパティファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.14.2　HTTP アダプタ実行環境プロパティファイル」を参照してください。

注※

サービス ID は，HTTP アダプタを新規に追加するときにサービスアダプタ定義画面で指定する任意の文字列です。

HTTP アダプタ実行環境プロパティファイルは，HTTP アダプタを開始する際に実行環境に反映されます。このため，HTTP アダプタ実行環境プロパティファイルの内容を変更する場合は，いったん HTTP アダプタを停止する必要があります。

(c)　HTTP アダプタ実行環境共通プロパティファイル

1. テンプレートファイル（<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¥custom-adapter¥HTTP¥config¥templates¥adphttpcom.properties）をコピーして，次のディレクトリに格納します。

全 HCSC サーバ共通の定義の場合

common ディレクトリはユーザが作成してください。

<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>¥CSC¥custom-adapter¥HTTP¥config¥common

特定の HCSC サーバ用の定義の場合

サブディレクトリ<HCSC サーバ名>と「common」はユーザが作成してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

<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¥custom-adapter¥HTTP¥config¥<HCSC サーバ名>¥common

ファイル名は変更しないでください。

2. 定義内容を編集して保存します。

HTTP アダプタ実行環境共通プロパティファイルで編集できる定義内容は，HTTP アダプタ実行環境プロパティファイルと同じです。詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.14.3　HTTP アダプタ実行環境共通プロパティファイル」を参照してください。

HTTP アダプタ実行環境共通プロパティファイルは，HTTP アダプタを開始する際に実行環境に反映されます。このため，HTTP アダプタ実行環境共通プロパティファイルの内容を変更する場合は，いったん HTTP アダプタを停止する必要があります。

(d)　JSON-XML 変換定義ファイル

JSON-XML 変換定義ファイルは，サービスプラットフォームが提供するテンプレートファイルを独自定義ファイルとして追加し，編集します。ファイルの内容については，ボディ変数用の電文フォーマットの作成時に使用した JSON-XML 変換定義ファイルと合わせてください。

JSON-XML 変換定義ファイルの編集手順を示します。

1. サービスアダプタ定義画面（詳細）の［独自定義ファイル］の［追加］ボタンをクリックして，JSON-XML 変換定義ファイルを追加します。
   「<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¥config¥msg¥templates」に格納されている JSON-XML 変換定義ファイルのテンプレートファイル（csc_json_converter.properties）を選択してください。
2. ［独自定義ファイル］の［編集］ボタンをクリックします。
   JSON-XML 変換定義ファイルを編集するためのエディタが起動します。
3. エディタ上で JSON-XML 変換定義ファイルを編集します。
   JSON-XML 変換定義ファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「3.18.1　JSON-XML 変換定義ファイル」を参照してください。
4. Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(5)　J2EE サーバ用オプション定義ファイルの設定

プロキシサーバやコネクション継続などの設定が必要な場合は，J2EE サーバ用オプション定義ファイル（usrconf.cfg）で設定します。HTTP アダプタで設定するプロパティについては，マニュアル「サービスプラットフォーム 解説」の「2.16.6　プロキシサーバを経由した通信」およびマニュアル「サービスプラットフォーム 解説」の「2.16.7　コネクション継続（Keep-Alive）を使用した通信」を参照してください。

(6)　ビジネスプロセスの設定

HTTP アダプタを利用する場合，ビジネスプロセスのサービス呼出アクティビティ，および必要に応じてフォルト処理の設定が必要です。

サービス呼出アクティビティとフォルト処理の設定方法を次に示します。

(a)　サービス呼出アクティビティの設定

HTTP アダプタを呼び出すためのサービス呼出アクティビティを定義します。サービス呼出アクティビティの定義方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.6.4 サービス呼出アクティビティ」を参照してください。

［サービス呼出アクティビティ］ダイアログの設定内容を次に示します。

項目	設定値
アクティビティ名	任意の名称を設定します。
サービス名	HTTP アダプタの新規追加時に設定したサービス名を指定します。

項目		設定値
オペレーション名		「(3)　サービスアダプタ定義画面での操作」で設定したオペレーション名を指定します。
要求電文	ボディ割当変数	送信するデータ形式に応じて次のように指定します。 • ノーマルモードで送信する場合 (2)(c)で作成したフォームデータ送信用の電文フォーマットを割り当てた変数を指定します。 • パススルーモードで送信する場合 (2)(c)で作成したパススルーモードでのデータ送信用の電文フォーマットを割り当てた変数を指定します。 • ファイルデータを送信する場合 adphttp_body_empty.fdx を割り当てた変数を指定します。※1
	ヘッダ割当変数※2	Cookie の設定方法に応じて次のように指定します。 • Cookie を一括設定する場合 adphttp_header_request1.xsd を割り当てた変数を指定します。 • Cookie を個別設定する場合 adphttp_header_request2.xsd を割り当てた変数を指定します。
応答電文	ボディ割当変数	受信するデータ形式に応じて次のように指定します。 • パススルーモードで受信する場合 (2)(c)で作成したパススルーモードでのデータ受信用の電文フォーマットを割り当てた変数を指定します。 • ファイルデータを受信する場合 adphttp_body_empty.fdx を割り当てた変数を指定します。※3
	ヘッダ割当変数※2	Cookie の設定方法に応じて次のように指定します。 • Cookie を一括設定する場合 adphttp_header_response1.xsd を割り当てた変数を指定します。 • Cookie を個別設定する場合 adphttp_header_response2.xsd を割り当てた変数を指定します。
割当相関セット群		設定しません。

注※1

サービスを呼び出す前に変数を初期化しておく必要があります。

注※2

要求電文（ヘッダ）または応答電文（ヘッダ）を使用しない場合，設定は不要です。

注※3

サービス呼び出し後に空文字列が設定されます。

(b)　フォルト処理の設定

正常系として処理できないステータスコードやフォルトに変換したシステム例外をビジネスプロセスでエラーとして処理する場合は，サービス呼出アクティビティや上位のスコープにフォルト処理を定義します。フォルト処理の定義方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.4.4　フォルト処理を定義する」を参照してください。

［フォルト処理の割当］ダイアログの設定内容を次に示します。

項目	設定値
割当変数	• HTTP アダプタのフォルトをキャッチする場合 adphttp_fault.xsd を割り当てた任意の変数を指定します。 • システム例外のフォルトをキャッチする場合 cscfault.xsd を割り当てた任意の変数を指定します。
遷移先	フォルトが発生した場合に遷移する任意のアクティビティを指定します。

3.3.14　コマンドアダプタを定義する

コマンドアダプタの定義方法について説明します。

(1)　電文フォーマットの作成

コマンドアダプタでは，要求電文と応答電文の 2 種類の電文があります。それぞれの内容について説明します。

電文フォーマット作成手順の詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.　電文フォーマットの作成」を参照してください。

(a)　コマンドアダプタの要求電文フォーマット

サービス呼出アクティビティを使用して，コマンドアダプタに渡す要求電文の XML データフォーマットと指定値について説明します。

名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/cmd/request」です。

表 3-126　要求電文フォーマット

タグ名			種別	出現回数※	説明
<request>			−	1	−
	<command-line>		−	1	−
		<command>	string	1	外部コマンドのファイルパスを指定するタグです。
		<argument<*N*>>	string	0 または 1	外部コマンドのコマンドライン引数を指定するタグです。 argument<*N*>にそれぞれ 1 つずつ指定します。 <*N*>の部分は，1，2，…，100 のどれかを指定します。 <*N*> の部分は昇順に出現する必要があります。 出現する<*N*>が連続値である必要はありません。 要素値が空文字列の要素は，引数として採用しません。

タグ名				種別	出現回数※	説明
	<config>			−	0 または 1	外部コマンドプロセスに関する動作オプションを設定するタグです。 各子要素で値を指定した場合，コマンドアダプタ実行環境プロパティファイルの対応するプロパティ値は無効になります。 各子要素がない場合，コマンドアダプタ実行環境プロパティファイルの対応するプロパティ値が有効になります。 コマンドアダプタ実行環境プロパティファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.15.3 コマンドアダプタ実行環境プロパティファイル」を参照してください。
		<timeout>		−	0 または 1	外部コマンドプロセスのタイムアウトに関する情報を指定するタグです。
			<value>	int	0 または 1	タイムアウト時間（秒）を指定するタグです。 指定可能値は，0～86400 です。 0 を指定した場合，外部コマンドプロセスのタイムアウトを使用しません。 要素値が空要素の場合はエラーになります。 対応するコマンドアダプタ実行環境プロパティファイルのプロパティは adpcmd.config.timeout.value です。詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.15.3 コマンドアダプタ実行環境プロパティファイル」を参照してください。
			<enable-kill-process>	boolean	0 または 1	タイムアウト発生時の外部コマンドプロセス強制終了の実施可否を指定するタグです。 指定可能値は，true または false です（大文字可）。 true を指定した場合，タイムアウト発生時にプロセスの強制終了を試みます。 要素値が空要素の場合はエラーになります。 対応するコマンドアダプタ実行環境プロパティファイルのプロパティは adpcmd.config.timeout.enable-kill-process です。詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.15.3　コマンドアダプタ実行環境プロパティファイル」を参照してください。
		<process-environment>		−	0 または 1	外部コマンドプロセスの環境に関する情報を設定するタグです。
			<variable-file-path>	string	0 または 1	プロセス環境変数定義ファイルのファイルパスを指定するタグです。 絶対パスで指定してください。 HCSC サーバマシンのファイルを指定してください。 ファイルパスを指定した場合，コマンドアダプタ実行環境プロパティファイルで指定したプロセス環境変数定義ファイルの内容はすべて無視されます。 要素値が空要素の場合はエラーになります。

タグ名				種別	出現回数※	説明
			<variable-file-path>	string	0 または 1	対応するコマンドアダプタ実行環境プロパティファイルのプロパティは adpcmd.config.process-environment.variable-file-path です。詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.15.3　コマンドアダプタ実行環境プロパティファイル」を参照してください。
			<working-dir-path>	string	0 または 1	外部コマンドプロセスの作業ディレクトリを指定するタグです。 絶対パスで指定してください。 HCSC サーバマシンに存在するディレクトリを指定してください。 要素値が空要素の場合はエラーになります。 対応するコマンドアダプタ実行環境プロパティファイルのプロパティは adpcmd.config.process-environment.working-dir-path です。詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.15.3　コマンドアダプタ実行環境プロパティファイル」を参照してください。

（凡例）

－：該当する項目はありません。

注※

規定の出現回数を超える場合の動作は保証されません。

参考

サービスプラットフォームが提供している要求電文の XML スキーマファイルは，「<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/custom-adapter/Command/schema/adpcmd_request.xsd」に格納されています。

コマンドライン引数を 10 個以上使用したい場合は，フォルダごと任意の場所にコピーしてから，コマンドライン引数要素（argument<*N*>）を追加してください。

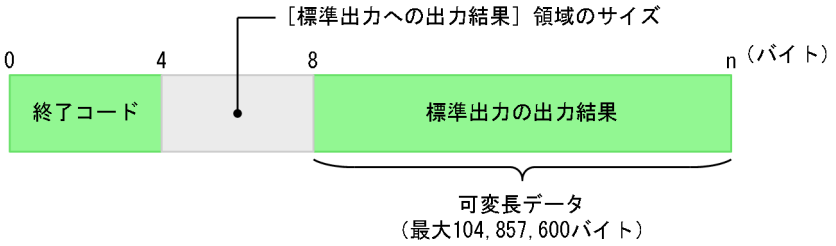
(b)　コマンドアダプタの応答電文フォーマット

サービス呼出アクティビティを使用して，コマンドアダプタから受け取る応答電文のバイナリデータフォーマットについて説明します。

●応答電文の内容

全体の構造を次の図で示します。

図 3-30　応答電文のバイナリフォーマットの構造



応答電文のバイナリデータフォーマットの詳細を次の表に示します。

表 3-127　応答電文のバイナリデータフォーマットの詳細

情報	サイズ（バイト）	説明
終了コード	4	外部コマンドプロセスの終了コードです。 符号付 2 進整数（リトルエンディアン）で表現した値が格納されています。
［標準出力への出力結果］領域のサイズ	4	標準出力への出力結果として応答電文に格納されているサイズ（バイト）です。 符号付 2 進整数（リトルエンディアン）で表現した値が格納されています。 サイズの最小値は 0，サイズの最大値はコマンドアダプタ実行環境プロパティファイルの adpcmd.config.stdout.size キーで指定した値です。
標準出力への出力結果	可変	外部コマンドプロセスが標準出力に出力した内容です。

サービスプラットフォームでは，標準出力への出力結果を 1 つの文字列として扱うバイナリフォーマットファイルを提供します。エンコード形式の設定が異なる 2 つのファイルを次の場所に提供します。

```
<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/custom-adapter/Command/schema/adpcmd_response_ms932.fdx
```

```
<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/custom-adapter/Command/schema/adpcmd_response_utf8.fdx
```

●応答電文フォーマットの作成

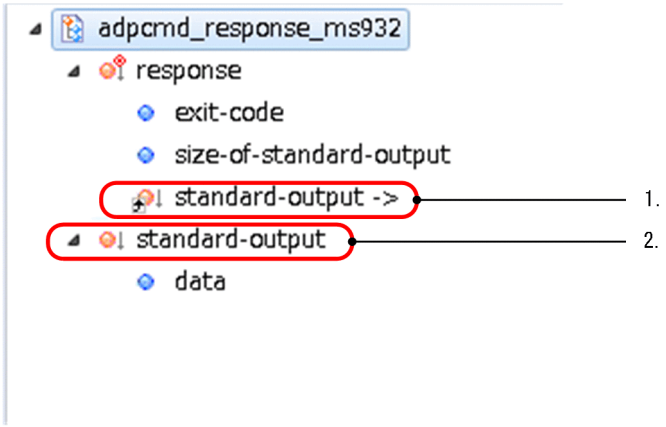
サービスプラットフォームが提供する応答電文のフォーマットファイルは，標準出力への出力結果を 1 つの文字列として扱う定義となっています。このファイルを利用する場合は，応答電文フォーマットを作成する必要はありません。

応答電文で標準出力への出力結果を構造データとして扱う場合は，バイナリフォーマットファイルを作成する必要があります。

バイナリフォーマットファイルの作成手順を次に示します。

1. サービスプラットフォームが提供する応答電文バイナリフォーマットファイルを，Eclipse のパッケージエクスプローラからアクセスできる場所にコピーします。
2. コピーしたファイルをバイナリフォーマットエディタで開きます。

3. 複合内容要素［standard-output］を削除します。
   削除する順序を次の図に示します。



4. 応答電文バイナリフォーマットの「標準出力への出力結果」領域に対応する定義を，外部コマンドの仕様に合わせて作成します。
   応答電文のバイナリデータフォーマットの詳細については，表 3-127 を参照してください。

作成したバイナリフォーマットファイルは，コマンドアダプタのサービスアダプタ定義画面で EXEC オペレーションの応答電文フォーマットに設定します。また，作成したバイナリフォーマットファイルでビジネスプロセスの変数を作成して，サービス呼出アクティビティの応答電文に指定します。

ポイント

コマンドアダプタは，要求電文で外部コマンドを指定できるため，リクエストごとに外部コマンドを変更できます。そのため，複数のビジネスプロセスでコマンドアダプタを共有しやすく，定義コンポーネント数を削減することができます。

定義コンポーネント数を削減するため，複数の外部コマンドに対して，外部コマンドの標準出力形式に合わせて作成した応答電文フォーマットを使用したい場合は，次のとおりビジネスプロセスとコマンドアダプタを設定してください。

- 外部コマンドに合わせて作成したバイナリフォーマット定義ファイルは，ビジネスプロセスでだけ使用してください。
  その際，サービス呼出アクティビティの応答電文変数は，コマンドアダプタが提供するバイナリフォーマット定義ファイルの変数とします。また，作成したバイナリフォーマット定義ファイルを基にした変数は，コマンドアダプタを呼び出した後に，Java 呼出アクティビティや代入アクティビティで使用してください。
- コマンドアダプタの EXEC オペレーションの応答電文フォーマットは，コマンドアダプタが提供するバイナリフォーマット定義ファイルを使用してください。

(2)　サービスアダプタ定義画面での操作

コマンドアダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。
   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービスアダプタ定義画面（基本）で定義情報を設定します。
   サービスアダプタ定義画面（基本）で設定が必要な項目については，「3.3.19(14)　コマンドアダプタの場合」およびマニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

3. [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画面（詳細）が表示されます。

4. サービスアダプタ定義画面（詳細）で定義情報を設定します。
   サービスアダプタ定義画面（詳細）で設定が必要な項目については，「3.3.19(14)　コマンドアダプタの場合」を参照してください。

(3)　コマンドアダプタの定義ファイルの作成および編集

作成および編集する定義ファイルの種類を次に示します。

- 実行許可コマンド定義ファイル
  コマンドアダプタで実行できる外部コマンドを定義するためのファイルです。
- プロセス環境変数定義ファイル
  外部コマンドプロセスだけで有効になる環境変数を定義するためのファイルです。
- コマンドアダプタ実行環境プロパティファイル
  コマンドアダプタごとの構成情報を設定するためのファイルです。
- コマンドアダプタ実行環境共通プロパティファイル
  すべてのコマンドアダプタに共通の構成情報を設定するためのファイルです。

各定義ファイルの作成手順および編集手順を次に示します。

(a)　実行許可コマンド定義ファイルの作成

1. コマンドアダプタで実行を許可するファイルパス，および指定を可能にするコマンドライン引数の仕様をまとめます。

2. チェック対象文字列を受理する正規表現を作成して，ファイルに定義します。

3. 作成したファイルを任意のパスに保存します。

注意事項

実行許可コマンド定義ファイルを有効にするには，コマンドアダプタ実行環境プロパティファイルの adpcmd.config.command.allow-file-path プロパティに保存先のファイルパスを指定してください。

(b)　プロセス環境変数定義ファイル

1. プロセス環境変数定義ファイルは，Java のプロパティファイル形式で作成します。
   環境変数名をキーとして，変数値を要素とします。

2. 作成したプロセス環境変数定義ファイルは，任意の場所に任意のファイル名で保存します。

注意事項

プロセス環境変数定義ファイルを有効にするには，コマンドアダプタ実行環境プロパティファイルの adpcmd.config.process-environment.variable-file-path プロパティに保存先のファイルパスを指定してください。

(c)　コマンドアダプタ実行環境プロパティファイル

1. テンプレートファイル（<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¥custom-adapter¥Command¥config¥templates¥serviceid.properties）をコピーして，次のディレクトリに格納します。

   全 HCSC サーバ共通の定義の場合

   <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>¥CSC¥custom-adapter¥Command¥config

   特定の HCSC サーバ用の定義の場合

   サブディレクトリ<HCSC サーバ名>を作成して格納してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

   <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¥custom-adapter¥Command¥config¥<HCSC サーバ名>

2. コピーしたテンプレートファイルのファイル名を「<サービス ID※>.properties」に変更します。

3. 定義内容を編集して保存します。
   コマンドアダプタ実行環境プロパティファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.15.3　コマンドアダプタ実行環境プロパティファイル」を参照してください。

注※

サービス ID は，開発環境のコマンドアダプタエディタで設定した値です。

コマンドアダプタ実行環境プロパティファイルは，コマンドアダプタを開始する際に実行環境に反映されます。このため，コマンドアダプタ実行環境プロパティファイルの内容を変更する場合は，いったんコマンドアダプタを停止する必要があります。

(d) コマンドアダプタ実行環境共通プロパティファイル

1. テンプレートファイル（<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¥custom-adapter¥Command¥config¥templates¥adpcmdcom.properties）をコピーして，次のディレクトリに格納します。

   全 HCSC サーバ共通の定義の場合

   common ディレクトリはユーザが作成してください。

   <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>¥CSC¥custom-adapter¥Command¥config¥common

   特定の HCSC サーバ用の定義の場合

   サブディレクトリ<HCSC サーバ名>と「common」はユーザが作成してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

   <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¥custom-adapter¥Command¥config¥<HCSC サーバ名>¥common

   ファイル名は変更しないでください。

2. 定義内容を編集して保存します。

コマンドアダプタ実行環境共通プロパティファイルで編集できる定義内容は，コマンドアダプタ実行環境プロパティファイルと同じです。

コマンドアダプタ実行環境共通プロパティファイルは，コマンドアダプタを開始する際に実行環境に反映されます。このため，コマンドアダプタ実行環境共通プロパティファイルの内容を変更する場合は，いったんコマンドアダプタを停止する必要があります。

3.3.15 SFTP アダプタを定義する

SFTP アダプタの定義方法について説明します。

(1) 電文フォーマットの作成

SFTP アダプタの電文フォーマットの定義ファイルはサービスプラットフォームが提供するスキーマを使用します。そのため，電文フォーマットの定義ファイルの作成は不要です。

SFTP アダプタで使用する電文フォーマットの内容について説明します。

ファイルの格納場所は「<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¥custom-adapter¥SFTP¥schema」です。

(a) SFTP アダプタの要求電文フォーマット

SFTP アダプタの要求電文フォーマットを，オペレーションごとに説明します。

●PUT オペレーション

ビジネスプロセスから SFTP アダプタに渡す PUT オペレーションの要求電文フォーマットを次に示します。ファイル名は「sftpadp_put_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csasftp/put_request」です。

表 3-128　要求電文フォーマット（PUT オペレーション）

タグ名	種別	出現回数※1	説明
<request>	−	1 回	−
<host-ipaddr>	string	0 または 1 回	接続する SFTP サーバの IP アドレスまたはホスト名です。 Windows を使用する場合，ループバック IP アドレス（127.0.0.1 または localhost）を指定すると，SFTP 接続に失敗する場合があります。実 IP アドレスまたはホスト名を指定してください。
<host-con-port>	long	0 または 1 回	接続する SFTP サーバのポート番号です。 1 から 65535 までのポート番号を指定します。
<sftp-user>※2	string	0 または 1 回	接続する SFTP サーバのユーザ名を示すタグです。 SFTP サーバが存在するマシンのユーザ名を指定します。
<data-compression>	boolean	0 または 1 回	転送データを圧縮するかどうかを指定するタグです。 true：転送データを圧縮します。 false：転送データを圧縮しません。
<host-auth>	−	0 または 1 回	ホスト認証に関する設定を示すタグです。
<strict-hostkey-checking>	boolean	0 または 1 回	厳格なホスト鍵チェックの実施の有無を指定するタグです。 true：厳格なホスト鍵チェックをします。 false：厳格なホスト鍵チェックをしません。
<overwrite-data>	boolean	0 または 1 回	転送先にファイルが存在する場合の動作を指定するタグです。 true：ファイルを上書きします。 false：ファイルを追記します。 要素が省略された場合は「true」が仮定されます。
<user-auth>	−	0 または 1 回	ユーザ認証に関する設定を示すタグです。
<user-auth-type>	string	0 または 1 回	ユーザ認証の形式を示すタグです。 password：パスワード認証です。 publickey：公開鍵認証です。
<user-auth-info>	string	0 または 1 回	ユーザ認証で使用する認証情報を指定するタグです。 • パスワード認証を実施する場合 運用コマンドで設定したユーザ名を指定します。

タグ名	種別	出現回数※1	説明
<user-auth-info>	string	0 または 1 回	• 公開鍵認証を実施する場合 運用コマンドで設定した鍵 ID を指定します。
<request-id>	string	0 または 1 回	受付で作成したリクエスト ID を示すタグです。 共通フォルダを利用するときは無視されます。
<local-folder>	string	1 回	ローカル側フォルダを示すタグです。 • タグの属性に common="true"が指定されている場合 ローカルフォルダとして共通フォルダを利用し，このタグには共通フォルダ定義名を指定します。 • タグの属性に common="false"が指定されている場合 ローカルフォルダとして作業フォルダを利用します。
<local-file-name>	string	1 回	ローカル側ファイル名を示すタグです。 • 作業フォルダを利用する場合，作業フォルダ直下のファイルを指定してください。ファイル名の先頭以外に区切り文字（スラッシュ（/），円マーク（¥）など）は使用できません。ファイル名の先頭にスラッシュ（/）を使用した場合，スラッシュ（/）は無視されます。 • 共通フォルダを利用する場合，共通フォルダルート配下のファイルを指定してください。シンボリックリンク以外のファイルを指定してください。
<remote-path>	string	1 回	リモートパス名（SFTP サーバに転送するときの転送先パス名）を示すタグです。
<sftp-commands-before>	string	0 または 1 回	ファイル転送前に実行する SFTP コマンドとその引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します。
<sftp-commands-after>	string	0 または 1 回	ファイル転送後に実行する SFTP コマンドとその引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

使用できる文字を次に示します。

半角英数字，半角空白，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

●GET オペレーション

ビジネスプロセスが SFTP アダプタに渡す GET オペレーションの要求電文のフォーマットを次に示します。ファイル名は「sftpadp_get_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csasftp/get_request」です。

表 3-129　要求電文フォーマット（GET オペレーション）

タグ名	種別	出現回数※1	説明
<request>	−	1 回	−
<host-ipaddr>	string	0 または 1 回	接続する SFTP サーバの IP アドレスまたはホスト名です。 Windows を使用する場合，ループバック IP アドレス（127.0.0.1 または localhost）を指定すると，SFTP 接続に失敗する場合があります。実 IP アドレスまたはホスト名を指定してください。
<host-con-port>	long	0 または 1 回	接続する SFTP サーバのポート番号です。 1 から 65535 までのポート番号を指定します。
<sftp-user>※2	string	0 または 1 回	接続する SFTP サーバのユーザ名を示すタグです。 SFTP サーバが存在するマシンのユーザ名を指定します。
<data-compression>	boolean	0 または 1 回	転送データを圧縮するかどうかを指定するタグです。 true：転送データを圧縮します。 false：転送データを圧縮しません。
<host-auth>	−	0 または 1 回	ホスト認証に関する設定を示すタグです。
<strict-hostkey-checking>	boolean	0 または 1 回	厳格なホスト鍵チェックの実施の有無を指定するタグです。 true：厳格なホスト鍵チェックをします。 false：厳格なホスト鍵チェックをしません。
<user-auth>	−	0 または 1 回	ユーザ認証に関する設定を示すタグです。
<user-auth-type>	string	0 または 1 回	ユーザ認証の形式を示すタグです。 password：パスワード認証です。 publickey：公開鍵認証です。
<user-auth-info>	string	0 または 1 回	ユーザ認証で使用する認証情報を指定するタグです。 • パスワード認証を実施する場合 運用コマンドで設定したユーザ名を指定します。 • 公開鍵認証を実施する場合 運用コマンドで設定した鍵 ID を指定します。
<request-id>	string	0 または 1 回	受付で作成したリクエスト ID を示すタグです。 共通フォルダを利用するときは無視されます。
<local-folder>	string	1 回	ローカル側フォルダを示すタグです。 • タグの属性に common="true"が指定されている場合

タグ名		種別	出現回数※1	説明
	<local-folder>	string	1 回	ローカルフォルダとして共通フォルダを利用し，このタグには「共通フォルダ定義名」を指定します。 • タグの属性に common="false"が指定されている場合 ローカルフォルダとして作業フォルダを利用します。
	<local-file-name>	string	0 または 1 回	ローカル側ファイル名を示すタグです。 • 作業フォルダを利用する場合 この設定は無視されます。 • 共通フォルダを利用する場合 共通フォルダルート配下のファイルを指定してください。 指定したファイル名でファイルが作成されます。 存在するファイル名を指定した場合，ファイルが上書きされます。 共通フォルダルートからの相対パスで指定したサブフォルダが存在しない場合は，サブフォルダを作成します。 また，シンボリックリンク以外のファイルを指定してください。
	<remote-path>	string	1 回	リモートパス名（SFTP サーバ上にある取得対象パス名）を示すタグです。
	<sftp-commands-before>	string	0 または 1 回	ファイル転送前に実行する SFTP コマンドとその引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します。
	<sftp-commands-after>	string	0 または 1 回	ファイル転送後に実行する SFTP コマンドとその引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

使用できる文字を次に示します。

半角英数字，半角空白，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

●GETINFO オペレーション

ビジネスプロセスが SFTP アダプタに渡す GETINFO オペレーションの要求電文のフォーマットを次に示します。ファイル名は「sftpadp_getinfo_request.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csasftp/getinfo_request」です。

表 3-130　要求電文フォーマット（GETINFO オペレーション）

タグ名	種別	出現回数※1	説明
<request>	−	1 回	−
<host-ipaddr>	string	0 または 1 回	接続する SFTP サーバの IP アドレスまたはホスト名です。 Windows を使用する場合，ループバック IP アドレス（127.0.0.1 または localhost）を指定すると，SFTP 接続に失敗する場合があります。実 IP アドレスまたはホスト名を指定してください。
<host-con-port>	long	0 または 1 回	接続する SFTP サーバのポート番号です。 1 から 65535 までのポート番号を指定します。
<sftp-user>※2	string	0 または 1 回	接続する SFTP サーバのユーザ名を示すタグです。 SFTP サーバが存在するマシンのユーザ名を指定します。
<data-compression>	boolean	0 または 1 回	転送データの圧縮の有無を指定するタグです。 true：転送データを圧縮します。 false：転送データを圧縮しません。
<host-auth>	−	0 または 1 回	ホスト認証に関する設定を示すタグです。
<strict-hostkey-checking>	boolean	0 または 1 回	厳格なホスト鍵チェックの実施の有無を指定するタグです。 true：厳格なホスト鍵チェックをします。 false：厳格なホスト鍵チェックをしません。
<user-auth>	−	0 または 1 回	ユーザ認証に関する設定を示すタグです。
<user-auth-type>	string	0 または 1 回	ユーザ認証の形式を示すタグです。 password：パスワード認証です。 publickey：公開鍵認証です。
<user-auth-info>	string	0 または 1 回	ユーザ認証で使用する認証情報を指定するタグです。 • パスワード認証を実施する場合 運用コマンドで設定したユーザ名を指定します。 • 公開鍵認証を実施する場合 運用コマンドで設定した鍵 ID を指定します。
<getinfo-type>	string	1 回	SFTP サーバから取得する情報の取得種別を指定するタグです。 LIST：ファイル情報の一覧を取得します。 NLST：ファイル名の一覧を取得します。
<getinfo-path>	string	0 または 1 回	接続する SFTP サーバから取得する情報の取得パスを示すタグです。<getinfo-type>で指定したコマンドを実行するときに指定するパスです。 タグがない場合，リストコマンド実行前に cd コマンドを実行したかどうかで，次のように異なります。 • cd コマンドを実行した場合 cd コマンドで変更したパスを利用します。

タグ名		種別	出現回数※1	説明
	<getinfo-path>	string	0 または 1 回	• cd コマンドを実行していない場合 リモートサーバのディレクトリのパスを利用します。
	<sftp-commands-before>	string	0 または 1 回	ファイル転送前に実行する SFTP コマンドとその引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します。
	<sftp-commands-after>	string	0 または 1 回	ファイル転送後に実行する SFTP コマンドとその引数を示すタグです。 複数のコマンドを実行する場合は，実行する順番にセミコロン（;）で区切って設定します。

（凡例）

－：該当する項目はありません。

注※1

規定の出現回数を超える場合の動作は保証されません。

注※2

使用できる文字を次に示します。

半角英数字，半角空白，感嘆符（!），ダブルクォーテーション（"），番号記号（#），ドル記号（$），パーセント（%），アンパサンド（&），アポストロフィ（'），始め小括弧（(），終わり小括弧（)），アスタリスク（*），正符号（+），コンマ（,），ハイフン（-），ピリオド（.），スラッシュ（/），コロン（:），セミコロン（;），左アングルブラケット（<），右アングルブラケット（>），等号（=），クエスチョンマーク（?），単価記号（@），始め角括弧（[），終わり角括弧（]），円マーク（¥），アクサンシルコンフレックス（^），アンダーバー（_），アクサングラーブ（`），始め波括弧（{），終わり波括弧（}），パイプライン（|），波ダッシュ（~）

(b) SFTP アダプタの応答電文フォーマット

SFTP アダプタの応答電文フォーマットを，オペレーションごとに説明します。

●PUT オペレーション

SFTP アダプタが呼び出し元のビジネスプロセスに返す PUT オペレーションの応答電文のフォーマットを次に示します。ファイル名は「sftpadp_put_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csasftp/put_response」です。

表 3-131　応答電文フォーマット（PUT オペレーション）

タグ名		種別	出現回数	説明
<response>		－	1 回	－
	<local-folder>	string	1 回	ローカル側フォルダを示すタグです。 作業フォルダを利用した場合，属性に「common="false"」が設定されます。値には空白が設定されます。 共通フォルダを利用した場合，属性に「common="true"」が設定されます。値には，要求電文の<local-folder>タグで指定された共通フォルダ定義名が設定されます。
	<local-file-name>	string	1 回	ローカル側ファイル名を示すタグです。

タグ名		種別	出現回数	説明
	<local-file-name>	string	1 回	要求電文の<local-file-name>で指定した値が設定されます。

（凡例）

－：該当する項目はありません。

●GET オペレーション

SFTP アダプタが呼び出し元のビジネスプロセスに返す GET オペレーションの応答電文のフォーマットを次に示します。ファイル名は「sftpadp_get_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csasftp/get_response」です。

表 3-132　応答電文フォーマット（GET オペレーション）

タグ名		種別	出現回数	説明
<response>		－	1 回	－
	<local-folder>	string	1 回	ローカル側フォルダを示すタグです。 作業フォルダを利用した場合，属性に「common="false"」が設定されます。値には空白が設定されます。 共通フォルダを利用した場合，属性に「common="true"」が設定されます。値には，要求電文の<local-folder>タグで指定された共通フォルダ定義名が設定されます。
	<local-file-name>	string	1 回	ローカル側ファイル名を示すタグです。 作業フォルダを利用する場合 作業フォルダの下に格納したファイル名が設定されます。ファイル名は，「csc_<内部生成された文字>」の形式となります。 共通フォルダを利用する場合 要求電文の<local-file-name>で指定した値が設定されます。
	<file-size>	long	1 回	SFTP サーバから取得したローカルファイルのサイズを示すタグです。

（凡例）

－：該当する項目はありません。

●GETINFO オペレーション

SFTP アダプタが呼び出し元のビジネスプロセスに返す GETINFO オペレーションの応答電文のフォーマットを次に示します。ファイル名は「sftpadp_getinfo_response.xsd」です。名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csasftp/getinfo_response」です。

表 3-133　応答電文フォーマット（GETINFO オペレーション）

タグ名		種別	出現回数	説明
<response>		－	1 回	－
	<list-info>	－	1 回	ファイル一覧情報を示すタグです。

タグ名			種別	出現回数	説明
		<list>	string	0 回以上	ファイル情報を示すタグです。 要求電文の getinfo-type 要素に「LIST」を指定した場合 ファイル情報の一覧を設定します。 要求電文の getinfo-type 要素に「NLST」を指定した場合 ファイル名の一覧を設定します。 リモートサーバから取得したファイル情報が何もない場合，この要素は出現しません。

（凡例）

－：該当する項目はありません。

(2) SFTP アダプタの定義ファイルの作成

作成する定義ファイルの種類を次に示します。

- SFTP アダプタ実行環境プロパティファイル
  SFTP アダプタごとの構成情報を設定するファイルです。
- SFTP アダプタ実行環境共通プロパティファイル
  すべての SFTP アダプタに共通する構成情報を設定するファイルです。

各定義ファイルの作成手順を次に示します。定義ファイルは，SFTP アダプタが提供しているテンプレートファイルを利用して作成できます。

(a) SFTP アダプタ実行環境プロパティファイル

1. テンプレートファイル（<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¥custom-adapter¥SFTP¥config¥templates¥adpsftp.properties）をコピーして，次のディレクトリに格納します。

   全 HCSC サーバ共通の定義の場合

   <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>¥CSC¥custom-adapter¥SFTP¥config

   特定の HCSC サーバ用の定義の場合

   サブディレクトリ<HCSC サーバ名>を作成して格納してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

   <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¥custom-adapter¥SFTP¥config¥<HCSC サーバ名>

2. コピーしたテンプレートファイルのファイル名を「<サービス ID※>.properties」に変更します。

3. 定義内容を編集して保存します。

SFTP アダプタ実行環境プロパティファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.16.1　SFTP アダプタ実行環境プロパティファイル」を参照してください。

注※

サービス ID は，SFTP アダプタを新規に追加するときにサービスアダプタ定義画面で指定する任意の文字列です。

(b)　SFTP アダプタ実行環境共通プロパティファイル

1. テンプレートファイル（<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¥custom-adapter¥SFTP¥config¥templates¥adpsftp.properties）をコピーして，次のディレクトリに格納します。

全 HCSC サーバ共通の定義の場合

common ディレクトリはユーザが作成してください。

<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>¥CSC¥custom-adapter¥SFTP¥config¥common

特定の HCSC サーバ用の定義の場合

サブディレクトリ<HCSC サーバ名>と「common」はユーザが作成してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

<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¥custom-adapter¥SFTP¥config¥<HCSC サーバ名>¥common

ファイル名は変更しないでください。

2. 定義内容を編集して保存します。

SFTP アダプタ実行環境プロパティファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.16.2　SFTP アダプタ実行環境共通プロパティファイル」を参照してください。

(3)　データ変換の定義

データ変換定義画面を利用して，変換元の電文フォーマット定義ファイルおよび変換先の電文フォーマット定義ファイルのデータ変換を定義します。

データ変換の定義方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.　データ変換の定義」を参照してください。

(4)　サービスアダプタ定義画面での操作

SFTP アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。

サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. 定義情報を設定します。

サービスアダプタ定義画面（基本）で設定が必要な項目については，「3.3.19(15)　SFTP アダプタの場合」およびマニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

3. [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画面（詳細）が表示されます。

4. 定義情報を設定します。

サービスアダプタ定義画面（詳細）で設定が必要な項目については，「3.3.19(15)　SFTP アダプタの場合」およびマニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

3.3.16　Kafka アダプタを定義する

Kafka アダプタの定義方法について説明します。

(1)　電文フォーマットの作成

Kafka アダプタでは，要求電文と応答電文の 2 種類の電文があります。それぞれの内容について説明します。

電文フォーマット作成手順の詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.　電文フォーマットの作成」を参照してください。

(a)　Kafka アダプタの要求電文フォーマット

Kafka アダプタの要求電文フォーマットについて説明します。

Kafka アダプタの要求電文フォーマットには，ヘッダ電文とボディ電文があります。

●ヘッダ要求電文

ヘッダ要求電文は，Kafka アダプタの要求電文フォーマットに従って作成します。

ファイルの格納場所は「<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¥custom-adapter¥Kafka¥schema¥adpkafka_header_send_request.xsd」です。

テキストデータは，W3C の XML の仕様に従って指定します。XML 文書で使用できない「<」，「>」，「&」などの文字を使用する場合，エスケープするか，CDATA セクションを利用する必要があります。

名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/kafka/header_send_request」です。

表 3-134　Kafka アダプタの要求電文フォーマット（ヘッダ）

タグ名	種別	出現回数	説明
<kafka-header-send-request>	−	1 回	−
<record>	−	1 回	−
<topic>	string	1 回	メッセージの送信先トピックを指定します。 指定できる文字数は 249 文字以下です。半角英数文字，ピリオド（.），アンダーバー（_），ハイフン（-）を使用できます。 範囲外の数値を指定した場合，および要素値が空要素の場合は送信時にエラーになります。
<headers>	−	0 または 1 回	−
<header>	−	0 回以上	−
<key>	string	1 回	メッセージヘッダのキーを指定します。任意の文字列を指定できます。 要素値が空要素の場合は，値に空文字を設定して，メッセージを送信します。
<value> ※1	string	0 または 1 回	メッセージヘッダの値を指定します。 任意の文字列を指定できます。 要素値を UTF-8 でバイト配列に変換した値をメッセージヘッダの値に設定してメッセージを送信します。 要素値が空要素の場合は，値に 0 バイトのバイト配列（長さ 0 の配列）を設定して，メッセージを送信します。 要素を省略した場合は，値に null を設定して，メッセージを送信します。
<key>※2	string	0 または 1 回	メッセージのキーを指定します。 任意の文字列を指定できます。 要素値が空要素の場合は，値に空文字を設定して，メッセージを送信します。 要素を省略した場合は，値に null を設定して，メッセージを送信します。
<partition>	int	0 または 1 回	メッセージの送信先のパーティション ID を指定します。0～2,147,483,647 までの数値を指定できます。範囲外の数値を指定した場合，および要素値が空要素の場合は，要求電文解析時にエラーになります。 要素を省略した場合は，値に null を設定して，メッセージを送信します。
<timestamp>	long	0 または 1 回	メッセージのタイムスタンプを指定します。0～9,223,372,036,854,775,807 までの数値を指定できます。 基点からミリ秒で時間を設定します。 要素を省略した場合は，トピックのデフォルトの設定では，メッセージの送信時刻（基点からのミリ秒単位の時間）がタイムスタン

タグ名			種別	出現回数	説明
		<timestamp>	long	0 または 1 回	プに付与されます。トピックの設定の詳細については，Apache Kafka の公式ドキュメントを参照してください。 範囲外の数値を指定した場合，および要素値が空要素の場合は，要求電文解析時にエラーになります。

（凡例）

－：該当する項目はありません。

注※1

value 要素に設定した文字列を，バイト配列に変換し，メッセージのヘッダの値に設定して Kafka に送信します。バイト配列を文字セット「UTF-8」を使用して変換します。

注※2

送信したい key のデータ型に応じて producer.key.serializer にシリアライズクラスを設定する必要があります。

例えば，送信データが文字列の場合に，producer.key.serializer に ByteArraySerializer を指定するなど，producer.key.serializer に送信データと対応していない値を指定したときの動作は保証しません。key 要素の送信データとシリアライズクラスとの対応については，表 3-135 を参照してください。

key 要素の送信データとシリアライズクラスとの対応を次に示します。

表 3-135　key 要素の送信データとシリアライズクラスとの対応

送信データ	producer.key.serializer の値
バイナリ形式（hexBinary，base64Binary）	ByteArraySerializer
文字列	StringSerializer

producer.key.serializer に ByteArraySerializer を指定した場合，ヘッダ要求電文のメッセージの key 要素に指定した値を adpkafka.request.record.binary.decoding に指定したデコード形式でデコードし，Kafka に送信するため，producer.key.serializer に指定する値は，adpkafka.request.record.binary.decoding に指定したデコード形式に合わせる必要があります。形式が異なる場合の動作は保証されません。

producer.key.serializer および adpkafka.request.record.binary.decoding の詳細については，マニュアル「サービスプラットフォーム リファレンス」の「3.13.1　Kafka アダプタ定義ファイル」を参照してください。

●ボディ要求電文

ボディ要求電文はユーザが任意に作成します。

ボディ要求電文で指定された内容は，Kafka の producer.value.serializer の指定に基づいてバイト配列化され，ヘッダ要求電文で指定した項目と共に Producer API を使用して Kafka に送信されます。

ボディ要求電文は，producer.value.serializer で指定したシリアライズクラスに対応した形式で作成してください。

例えば，要求電文形式が Binary の場合に，producer.key.serializer に StringSerializer を指定するなど，要求電文形式と対応していない値を指定したときは，送信要求実行時にエラーを出力し，処理が終了します。

要求電文形式とシリアライズクラスとの対応について次に示します。

表 3-136　要求電文形式とシリアライズクラスとの対応

要求電文形式※1	送信データ	producer.value.serializer の値	リクエスト電文の例
XML	XML 形式※2	StringSerializer	<userdata> 　<dataA>dataA</dataA> 　<dataB>dataB</dataB> </userdata>
	JSON 形式		
Binary	CSV 形式のデータ	ByteArraySerializer	data, data, data
	文字列		data
	バイナリデータ		0x64 0x61 0x74 0x61
Any	文字列	ByteArraySerializer	data
	バイナリデータ		0x64 0x61 0x74 0x61

注※1

要求電文形式が XML の場合は XML スキーマ，Binary の場合はバイナリフォーマット定義で作成します。

注※2

文字セットは「UTF-8」固定です。

producer.value.serializer の指定方法の詳細については，マニュアル「サービスプラットフォーム リファレンス」の「3.13.1　Kafka アダプタ定義ファイル」を参照してください。

(b)　Kafka アダプタの応答電文フォーマット

Kafka アダプタの応答電文フォーマットについて説明します。

ファイルの格納場所は「<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¥custom-adapter¥Kafka¥schema¥adpkafka_send_response.xsd」です。応答電文フォーマットの形式には XML 形式を使用してください。

名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/kafka/send_response」です。

表 3-137　要求電文フォーマット（Kafka アダプタ）

タグ名		種別	出現回数	説明
<send_response>		－	1 回	－
	<status>	string	1 回	送信が成功したことを示す結果「OK」が格納されます。

（凡例）

－：該当する項目はありません。

注

送信が失敗した場合，Kafka アダプタはカスタムアダプタ開発フレームワークに例外を通知し，応答電文は返さないため，結果が格納されるのは送信処理が成功した場合だけです。

(2) Kafka アダプタの定義ファイルの作成

作成する定義ファイルの種類を次に示します。

- Kafka アダプタ定義ファイル
  Kafka アダプタの動作情報を設定するためのファイルです。
- Kafka アダプタ実行環境プロパティファイル
  Kafka アダプタごとの構成情報を設定するファイルです。
- Kafka アダプタ実行環境共通プロパティファイル
  すべての Kafka アダプタに共通する構成情報を設定するファイルです。

各定義ファイルの作成手順を次に示します。定義ファイルは，サービスプラットフォームが提供しているテンプレートファイルを利用して作成できます。

(a) Kafka アダプタ定義ファイル

Kafka アダプタ定義ファイルは，サービスプラットフォームが提供するテンプレートファイルを利用して編集します。

Kafka アダプタ定義ファイルの編集手順を示します。

1. サービスアダプタ定義画面（詳細）の［独自定義ファイル］で「adpkafka.properties」を選択し，［編集］ボタンをクリックします。
   Kafka アダプタ定義ファイルを編集するためのエディタが起動します。

2. エディタ上で Kafka アダプタ定義ファイルを編集します。
   Kafka アダプタ定義ファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「3.13.1　Kafka アダプタ定義ファイル」を参照してください。

3. Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(b) Kafka アダプタ実行環境プロパティファイル

Kafka アダプタ実行環境プロパティファイルの設定方法を次に示します。

1. テンプレートファイル（<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¥custom-adapter¥Kafka¥config¥templates¥serviceid.properties）をコピーして，次のディレクトリに格納します。

   全 HCSC サーバ共通の定義の場合

   　<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>¥CSC¥custom-adapter¥Kafka¥config

   特定の HCSC サーバ用の定義の場合

   　サブディレクトリ<HCSC サーバ名>を作成して格納してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

<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¥custom-adapter¥Kafka¥config¥<HCSC サーバ名>

2. コピーしたテンプレートファイルのファイル名を「<サービス ID※>.properties」に変更します。

3. 定義内容を編集して保存します。

Kafka アダプタ実行環境プロパティファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.17.1　Kafka アダプタ実行環境プロパティファイル」を参照してください。

注※

サービス ID は，Kafka アダプタを新規に追加するときにサービスアダプタ定義画面で指定する任意の文字列です。

(c)　Kafka アダプタ実行環境共通プロパティファイル

Kafka アダプタ実行環境共通プロパティファイルの設定方法を次に示します。

1. テンプレートファイル（<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¥custom-adapter¥Kafka¥config¥templates¥adpkafkacom.properties）をコピーして，次のディレクトリに格納します。

全 HCSC サーバ共通の定義の場合

common ディレクトリはユーザが作成してください。

<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>¥CSC¥custom-adapter¥Kafka¥config¥common

特定の HCSC サーバ用の定義の場合

サブディレクトリ<HCSC サーバ名>と「common」はユーザが作成してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

<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¥custom-adapter¥Kafka¥config¥<HCSC サーバ名>¥common

ファイル名は変更しないでください。

2. 定義内容を編集して保存します。

Kafka アダプタ実行環境プロパティファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.17.2　Kafka アダプタ実行環境共通プロパティファイル」を参照してください。

(3)　データ変換の定義

データ変換定義画面を利用して，変換元の電文フォーマット定義ファイルおよび変換先の電文フォーマット定義ファイルのデータ変換を定義します。

データ変換の定義方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.　データ変換の定義」を参照してください。

(4)　サービスアダプタ定義画面での操作

Kafka アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。
   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. 定義情報を設定します。
   サービスアダプタ定義画面（基本）で設定が必要な項目については，「3.3.19(16)　Kafka アダプタの場合」およびマニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

3. [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画面（詳細）が表示されます。

4. 定義情報を設定します。
   サービスアダプタ定義画面（詳細）で設定が必要な項目については，「3.3.19(16)　Kafka アダプタの場合」およびマニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

3.3.17　gRPC アダプタを定義する

gRPC アダプタの定義方法について説明します。

(1)　電文フォーマットの作成

gRPC アダプタの電文フォーマットの定義ファイルはサービスプラットフォームが提供するスキーマを使用します。そのため，電文フォーマットの定義ファイルの作成は不要です。

gRPC アダプタで使用する電文フォーマットの内容について説明します。

ファイルの格納場所は「<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¥custom-adapter¥gRPC¥schema」です。

(a)　gRPC アダプタの要求電文フォーマット

gRPC アダプタの要求電文フォーマット（ヘッダ）について説明します。

ファイルの格納場所は「<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¥custom-adapter¥gRPC¥schema¥adpgrpc_metadata_request.xsd」です。

名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/grpc/header/metadata_request」です。

表 3-138　gRPC アダプタの要求電文フォーマット（ヘッダ）

タグ名			種別	出現回数	説明
<grpc-metadata-request>			−	1 回	−
	<metadata>		−	0 回以上	gRPC サービスに送信するメタデータを指定します。
		<key>	string	1 回	gRPC サービスに送信するメタデータのキーを指定します。キーに指定できる値は半角英数字，アンダーバー（_），ハイフン（-），ピリオド（.）です。キーの省略はできません。指定値の大文字・小文字は区別しません。 「grpc-」から始まる要素名は指定できません。「grpc-」から始まる要素名を指定した場合の動作は保証されません。 バイナリ形式で送信する場合，この要素値は「-bin（大文字・小文字を区別する）」で終了してください。ただし，「-bin」だけを指定することはできません。
		<value>	string	1 回	メタデータのキーに対する値を指定します。 テキスト形式の場合，0x20 および 0x21～0x7E の範囲の文字列が指定できます。ただし，空白（0x20）は，値の最初または最後には指定できません。 指定できない値を指定した場合の動作は保証されません。 バイナリ形式の場合，Base64 でエンコードした値を指定します。

(b)　gRPC アダプタの応答電文フォーマット

gRPC アダプタの応答電文フォーマット（ヘッダ）について説明します。

ファイルの格納場所は「<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¥custom-adapter¥gRPC¥schema¥adpgrpc_metadata_response.xsd」です。

名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/grpc/header/metadata_response」です。

表 3-139　gRPC アダプタの応答電文フォーマット（ヘッダ）

タグ名			種別	出現回数	説明
<grpc-metadata-response>			−	1 回	−
	<metadata>		−	0 回以上	gRPC サービスから返却されたメタデータを指定します。
		<key>	string	1 回	gRPC サービスから返却されたメタデータのキーを指定します。 バイナリ形式の場合，この要素値は「-bin」で終了させてください。
		<value>	string	1 回	メタデータのキーに対する値を指定します。 バイナリ形式の場合，Base64 でエンコードした値を指定します。

(c) gRPC アダプタのフォルト電文フォーマット

gRPC アダプタでは，gRPC のメッセージ内容※に沿ったフォルト電文フォーマットを提供しています。このフォルト電文フォーマット（XML スキーマファイル）は，gRPC アダプタを作成したときに，gRPC アダプタ定義画面（基本）のフォルト電文フォーマットに自動的に設定されます。

注※

gRPC を利用した通信では，google.rpc.Status のメッセージの処理方式によってエラー処理を行っています。google.rpc.Status のメッセージ内容，およびエラー処理については，Google 社の google.rpc.Status に関するサイトを参照してください。

gRPC アダプタを作成したあと，必要に応じてフォルト電文フォーマットの XML スキーマファイルを修正し，再登録できます。再登録することで，XML スキーマファイルの記述を簡略化できます。

XML スキーマファイルを修正する手順を次に示します。

1. 次の XML スキーマファイルを任意の場所にコピーします。

   <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¥custom-adapter¥gRPC¥schema¥adpgrpc_fault.xsd

2. コピーした XML スキーマファイルをテキストエディタで開きます。

3. XML スキーマファイルに定義されている，/fault/status/details 要素以下に定義されている次の要素のうち，使用しない要素を削除します。

   - retry-info
   - debug-info
   - quota-failure
   - error-info
   - precondition-failure
   - bad-request
   - request-info
   - resource-info
   - help
   - localized-message
   - additional-message

   処理に必要な要素を削除した場合の動作は保証されません。

   これらの要素は，Google 社が提供する google/rpc/error_details.proto の内容に対応しています。google/rpc/error_details.proto の内容については，Google 社の google/rpc/error_details.proto に関するサイトを参照してください。

なお，additional-message は，google/rpc/error_details.proto に定義されていない任意の型のメッセージを格納する場合に使用します。

4. 使用しない要素（子要素を含む）を削除した XML スキーマファイルを保存します。

5. 修正した XML スキーマファイルを，サービスアダプタ定義画面（基本）の［フォルト電文］に設定します。

   修正した XML スキーマファイル以外を設定した場合，動作は保証されません。

gRPC アダプタのフォルト電文フォーマットの形式について説明します。

名前空間名は「http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/grpc/fault」です。

表 3-140　gRPC アダプタのフォルト電文フォーマット

タグ名	種別	出現回数	説明
<fault>	−	1 回	−
<status>	−	1 回	ステータス情報を指定します。
<code>	文字列	1 回	ステータスコード※を指定します。
<message>	文字列	0～1 回	開発者向けのエラーメッセージを指定します。
<details>	−	0～1 回	詳細エラーのメッセージのリストを指定します。子要素として「error_details.proto」のメッセージ構造体に応じた要素が出現します。
<additional-message>	−	0 回以上	「error_details.proto」のメッセージ構造体に定義されていないメッセージの定義内容を指定します。Any 型のメッセージとして扱われます。
<type-url>	文字列	1 回	メッセージのタイプ（URL/resource name）を指定します。
<value>	文字列	1 回	メッセージの値を Base64 エンコードしたデータを指定します。
<bad-request>	−	0 回以上	メッセージ構造体「BadRequest」の定義内容を指定します。
<field-violation>	−	0 回以上	メッセージ構造体「FieldViolation」の定義内容を指定します。
<field>	文字列	0～1 回	引数「field」が設定されます。
<description>	文字列	0～1 回	引数「description」が設定されます。
<debug-info>	−	0 回以上	メッセージ構造体「DebugInfo」の定義内容を指定します。
<stack-entries>	文字列	0 回以上	引数「stack_entries」が設定されます。
<detail>	文字列	0～1 回	引数「detail」が設定されます。

タグ名	種別	出現回数	説明
<error-info>	－	0 回以上	メッセージ構造体「ErrorInfo」の定義内容を指定します。
<reason>	文字列	0～1 回	引数「reason」が設定されます。
<domain>	文字列	0～1 回	引数「domain」が設定されます。
<metadata>	－	0 回以上	引数「metadata」が設定されます。
<key>	文字列	1 回	メタデータのキーを指定します。
<value>	文字列	1 回	メタデータの値を指定します。
<help>	－	0 回以上	メッセージ構造体「Help」の定義内容を指定します。
<link>	－	0 回以上	メッセージ構造体「Link」の定義内容を指定します。
<description>	文字列	0～1 回	引数「description」が設定されます。
<url>	文字列	0～1 回	引数「url」が設定されます。
<localized-message>	－	0 回以上	メッセージ構造体「LocalizedMessage」の定義内容を指定します。
<locale>	文字列	0～1 回	引数「locale」が設定されます。
<message>	文字列	0～1 回	引数「message」が設定されます。
<precondition-failure>	－	0 回以上	メッセージ構造体「PreconditionFailure」の定義内容を指定します。
<violation>	－	0 回以上	メッセージ構造体「Violation」の定義内容を指定します。
<type>	文字列	0～1 回	引数「type」が設定されます。
<subject>	文字列	0～1 回	引数「subject」が設定されます。
<description>	文字列	0～1 回	引数「description」が設定されます。
<quota-failure>	－	0 回以上	メッセージ構造体「QuotaFailure」の定義内容を指定します。
<violation>	－	0 回以上	メッセージ構造体「Violation」の定義内容を指定します。
<subject>	文字列	0～1 回	引数「subject」が設定されます。
<description>	文字列	0～1 回	引数「description」が設定されます。
<request-info>	－	0 回以上	メッセージ構造体「RequestInfo」の定義内容を指定します。
<request-id>	文字列	0～1 回	引数「request_id」が設定されます。
<serving-data>	文字列	0～1 回	引数「serving_data」が設定されます。

タグ名						種別	出現回数	説明
			<resource-info>			－	0 回以上	メッセージ構造体「ResourceInfo」の定義内容を指定します。
				<resource-type>		文字列	0～1 回	引数「resource_type」が設定されます。
				<resource-name>		文字列	0～1 回	引数「resource_name」が設定されます。
				<owner>		文字列	0～1 回	引数「owner」が設定されます。
				<description>		文字列	0～1 回	引数「description」が設定されます。
			<retry-info>			－	0 回以上	メッセージ構造体「RetryInfo」の定義内容を指定します。
				<retry-delay>		文字列	0～1 回	引数「retry_delay」が設定されます。
					<seconds>	文字列	1 回	引数「retry_delay」の秒が設定されます。
					<nanos>	文字列	1 回	引数「retry_delay」のナノ秒が設定されます。
	<metadata>					－	0 回以上	トレーラ（gRPC サービスから返却されたメタデータ）を指定します。
		<key>				文字列	1 回	メタデータのキーを指定します。
		<value>				文字列	1 回	メタデータのキーに対する値を指定します。 バイナリ形式の場合，Base64 でエンコードした値を設定します。

注※

フォルト電文に設定できるステータスコードを次に示します。次に示すステータスコード以外の値が設定された場合，リクエストがエラーとなります。

- CANCELLED
- UNKNOWN
- INVALID_ARGUMENT
- DEADLINE_EXCEEDED
- NOT_FOUND
- ALREADY_EXISTS
- PERMISSION_DENIED
- RESOURCE_EXHAUSTED
- FAILED_PRECONDITION
- ABORTED
- OUT_OF_RANGE
- UNIMPLEMENTED
- INTERNAL

- UNAVAILABLE
- DATA_LOSS
- UNAUTHENTICATED

(2) gRPC アダプタの定義ファイルの作成

作成する定義ファイルの種類を次に示します。

- gRPC アダプタ独自定義ファイル
  gRPC アダプタの動作情報を設定するファイルです。
- gRPC アダプタ実行環境プロパティファイル
  gRPC アダプタごとの構成情報を設定するファイルです。
- gRPC アダプタ実行環境共通プロパティファイル
  すべての gRPC アダプタに共通する構成情報を設定するファイルです。

各定義ファイルの作成手順を次に示します。定義ファイルは，gRPC アダプタが提供しているテンプレートファイルを利用して作成できます。

(a) gRPC アダプタ独自定義ファイル

gRPC アダプタ独自定義ファイルは，サービスプラットフォームが提供するテンプレートファイルを利用して編集します。

gRPC アダプタ独自定義ファイルの編集手順を示します。

1. サービスアダプタ定義画面（詳細）の［独自定義ファイル］で「cscadpgrpc.properties」を選択し，［編集］ボタンをクリックします。
   gRPC アダプタ独自定義ファイルを編集するためのエディタが起動します。
2. エディタ上で gRPC アダプタ独自定義ファイルを編集します。
   gRPC アダプタ独自定義ファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「3.16.1 gRPC アダプタ独自定義ファイル」を参照してください。
3. Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

(b) gRPC アダプタ実行環境プロパティファイル

gRPC アダプタ実行環境プロパティファイルの設定方法を次に示します。

1. テンプレートファイル（<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¥custom-adapter¥gRPC¥config¥templates¥serviceid.properties）をコピーして，次のディレクトリに格納します。

   全 HCSC サーバ共通の定義の場合
   <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>¥CSC¥custom-adapter¥gRPC¥config

特定の HCSC サーバ用の定義の場合

サブディレクトリ<HCSC サーバ名>を作成して格納してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

<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¥custom-adapter¥gRPC¥config¥<HCSC サーバ名>

2. コピーしたテンプレートファイルのファイル名を「<サービス ID※>.properties」に変更します。

3. 定義内容を編集して保存します。

gRPC アダプタ実行環境プロパティファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.19.1　gRPC アダプタ実行環境プロパティファイル」を参照してください。

注※

サービス ID は，gRPC アダプタを新規に追加するときにサービスアダプタ定義画面で指定する任意の文字列です。

(c)　gRPC アダプタ実行環境共通プロパティファイル

gRPC アダプタ実行環境共通プロパティファイルの設定方法を次に示します。

1. テンプレートファイル（<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¥custom-adapter¥gRPC¥config¥templates¥adpgrpccom.properties）をコピーして，次のディレクトリに格納します。

全 HCSC サーバ共通の定義の場合

common ディレクトリはユーザが作成してください。

<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>¥CSC¥custom-adapter¥gRPC¥config¥common

特定の HCSC サーバ用の定義の場合

サブディレクトリ<HCSC サーバ名>と「common」はユーザが作成してください。Windows の場合，HCSC サーバ名の大文字・小文字は区別されます。

<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¥custom-adapter¥gRPC¥config¥<HCSC サーバ名>¥common

2. 定義内容を編集して保存します。

ファイル名は変更しないでください。

gRPC アダプタ実行環境プロパティファイルで編集できる定義内容については，マニュアル「サービスプラットフォーム リファレンス」の「6.19.2　gRPC アダプタ実行環境共通プロパティファイル」を参照してください。

(3)　サービスアダプタ定義画面での操作

gRPC アダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。
   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. 定義情報を設定します。
   サービスアダプタ定義画面（基本）で設定が必要な項目については，「3.3.19(17)　gRPC アダプタの場合」およびマニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

3. [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画面（詳細）が表示されます。

4. 定義情報を設定します。
   サービスアダプタ定義画面（詳細）で設定が必要な項目については，「3.3.19(17)　gRPC アダプタの場合」およびマニュアル「サービスプラットフォーム リファレンス」の「1.2.2　サービスアダプタ定義画面」を参照してください。

(4)　ビジネスプロセスの設定

gRPC アダプタを利用する場合，ビジネスプロセスで次の設定が必要です。

- 変数の設定
  - 要求電文（ヘッダ）の変数
  - 応答電文（ヘッダ）の変数
  - 要求電文（ボディ）の変数
  - 応答電文（ボディ）の変数
  - フォルト電文の変数
- サービス呼出アクティビティの設定
- フォルト処理の設定

(a)　要求電文（ヘッダ）および応答電文（ヘッダ）の変数の設定

要求電文（ヘッダ）および応答電文（ヘッダ）の変数の設定手順を次に示します。

1. サービスアダプタ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. [変数名] に任意の名称を指定します。

3. [種別] ドロップダウンリストから「XML」を選択します。

4. [電文フォーマット] の［参照］ボタンをクリックします。
   電文フォーマットを指定するダイアログが表示されます。

5. 変数に応じて次のファイルを指定します。
   - 要求電文（ヘッダ）の変数
     <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¥custom-adapter¥gRPC¥schema¥adpgrpc_metadata_request.xsd
   - 応答電文（ヘッダ）の変数
     <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¥custom-adapter¥gRPC¥schema¥adpgrpc_metadata_response.xsd
6. [変数・相関セット一覧] ダイアログで［追加］ボタンをクリックします。
   変数が追加されます。

(b) 要求電文（ボディ），応答電文（ボディ），およびフォルト電文の変数の設定

要求電文（ボディ），応答電文（ボディ），およびフォルト電文の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. [変数名] に任意の名称を指定します。
3. [種別] ドロップダウンリストから「XML」を選択します。
4. [電文フォーマット] の［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] をチェックし，ドロップダウンリストからサービスアダプタ定義画面で設定した gRPC アダプタの名称を選択します。
6. [オペレーション名] ドロップダウンリストから gRPC アダプタに設定したオペレーションを選択します。
7. 変数に応じて［電文種別］ドロップダウンリストで電文種別を選択します。
   - 要求電文（ボディ）の変数の場合
     ［要求電文（ボディ）］を選択します。
   - 応答電文（ボディ）の変数の場合
     ［応答電文（ボディ）］を選択します。
   - フォルト電文の変数の場合
     ［フォルト電文］を選択します。
8. [電文フォーマット] に任意の名称を入力します。
9. [OK] ボタンをクリックします。
10. [変数・相関セット一覧] ダイアログで［追加］ボタンをクリックします。

変数が追加されます。

(c)　サービス呼出アクティビティの設定

gRPC アダプタを呼び出すためのサービス呼出アクティビティを定義します。サービス呼出アクティビティの定義方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.6.4　サービス呼出アクティビティ」を参照してください。

［サービス呼出アクティビティ］ダイアログの設定内容を次に示します。

項目		設定値
アクティビティ名		任意の名称を設定します。
サービス名		gRPC アダプタの新規追加時に設定したサービス名を指定します。
オペレーション名		「(3)　サービスアダプタ定義画面での操作」で設定したオペレーション名を指定します。
要求電文	ボディ割当変数	「(b)　要求電文（ボディ），応答電文（ボディ），およびフォルト電文の変数の設定」で設定した要求電文の変数を指定します。
	ヘッダ割当変数※	「(a)　要求電文（ヘッダ）および応答電文（ヘッダ）の変数の設定」でヘッダ割当変数に設定した要求電文（ヘッダ）の変数を指定します。
応答電文	ボディ割当変数	「(b)　要求電文（ボディ），応答電文（ボディ），およびフォルト電文の変数の設定」で設定した応答電文の変数を指定します。
	ヘッダ割当変数※	「(a)　要求電文（ヘッダ）および応答電文（ヘッダ）の変数の設定」でヘッダ割当変数に設定した応答電文（ヘッダ）の変数を指定します。
割当相関セット群		設定しません。

注※

要求電文（ヘッダ）または応答電文（ヘッダ）を使用しない場合，設定は不要です。

(d)　フォルト処理の設定

正常系として処理できないステータスコードやフォルトに変換したシステム例外をビジネスプロセスでエラーとして処理する場合は，サービス呼出アクティビティや上位のスコープにフォルト処理を定義します。フォルト処理の定義方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.4.4　フォルト処理を定義する」を参照してください。

［フォルト処理の割当］ダイアログの設定内容を次に示します。

項目	設定値
割当変数	「(b)　要求電文（ボディ），応答電文（ボディ），およびフォルト電文の変数の設定」で設定したフォルト電文の変数を指定します。
遷移先	フォルトが発生した場合に遷移する任意のアクティビティを指定します。

3.3.18　汎用カスタムアダプタを定義する

汎用カスタムアダプタの定義方法について説明します。汎用カスタムアダプタは，サービスアダプタ定義画面で定義できます。

(1)　オペレーションの設計

1 つの汎用カスタムアダプタは複数のオペレーションを持つことができます。1 つのオペレーションごとにオペレーション名，通信モデル，要求電文，および応答電文の設計が必要です。

(a)　オペレーション名の決定

オペレーション名は，サービスリクエスタからのサービス部品呼び出しや，ビジネスプロセスのサービス呼出アクティビティからのサービス部品呼び出しで，オペレーションを区別するための識別名です。

255 バイト以内で，汎用カスタムアダプタ内で一意な名称を付け，同じ汎用カスタムアダプタ内のほかのオペレーション名と重複しないように指定してください。

指定できる文字列および記号は次のとおりです。

- 英数字
- アンダーバー（_）
- ピリオド（.）
- ハイフン（-）

汎用カスタムアダプタ内のプロトコルコンバータでは，汎用カスタムアダプタが呼び出されたときに，どのオペレーションが呼ばれたのかを識別するために使います。

(b)　通信モデルの決定

汎用カスタムアダプタではオペレーションごとに通信モデルを選択できます。呼び出すサービス部品に合わせて，同期または非同期のどちらかを選択してください。

(c)　要求電文・応答電文の設計

プロトコルコンバータが受け取る要求電文とプロトコルコンバータが応答する応答電文について，電文形式と電文フォーマットを設計します。ただし，通信モデルが非同期の場合，応答電文を返さないので，応答電文を定義する必要はありません。

電文形式は，XML 形式とバイナリ形式が選べます。

電文フォーマットは，サービス部品の実行に利用する電文で XML 形式のデータを扱うか，バイナリ形式（XML 形式以外）のデータを扱うかによって形式が異なります。呼び出すサービス部品に合わせて XML 形式とバイナリ形式のどちらかを選択し，呼び出すサービス部品の形式に合わせて，電文フォーマットを作成してください。

XML 形式とバイナリ形式の違いについては，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.2　電文フォーマットの種類」を参照してください。

XML 形式のデータを扱う場合の電文フォーマットの作成については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.3　電文フォーマット（XML フォーマット定義ファイル）の作成方法」を，バイナリ形式（XML 形式以外）のデータを扱う場合の電文フォーマットの作成については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.4　電文フォーマット（バイナリフォーマット定義ファイル）の作成方法」を参照してください。

(2)　データ変換

「3.3.18(1)　オペレーションの設計」で設計したプロトコルコンバータが受けたい電文フォーマットと，サービスリクエスタやビジネスプロセスから要求するときの電文フォーマットが異なる場合，データ変換が必要です。電文のフォーマットは，サービスアダプタ定義画面で設定できます。

電文のデータ変換を行う場合，変換元の電文フォーマット定義ファイルおよび変換先の電文フォーマット定義ファイルを設定し，データ変換定義画面でデータ変換を定義します。

データ変換の詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7. データ変換の定義」を参照してください。

(3)　独自定義ファイルの作成と設定

プロトコルコンバータやカスタムアダプタ開発フレームワークが使用する定義ファイルを作成します。

- プロトコルコンバータ用独自定義ファイル
- カスタムアダプタ開発フレームワーク動作定義ファイル
- カスタムアダプタプロパティファイル
- カスタムアダプタ用アプリケーション統合属性ファイル
- カスタムアダプタ定義ファイル

作成した定義ファイルは，開発環境のサービスアダプタ定義画面で設定します。

次に作成するファイルの内容を説明します。

(a)　プロトコルコンバータ用独自定義ファイル

プロトコルコンバータは，アダプタコンテキストを通して，汎用カスタムアダプタに設定されたリソース内容を読み込めます。

プロトコルコンバータの動作に必要なファイルがあれば作成してください。

(b)　カスタムアダプタ開発フレームワーク動作定義ファイル

カスタムアダプタ開発フレームワークが使用するプロトコルコンバータのクラス名（パッケージ名を含む）を指定します。ファイル名は必ず「framework_properties.xml」にしてください。

カスタムアダプタ開発フレームワーク動作定義ファイルの設定内容については，「付録 B.2(1)　カスタムアダプタ開発フレームワーク動作定義ファイル」を参照してください。

(c)　カスタムアダプタプロパティファイル

プロトコルコンバータに関する設定を外部ファイル化したい場合など，必要に応じて作成するファイルです。このファイルは独自定義ファイルの一種で，汎用カスタムアダプタの定義時に使用します。ファイル名は必ず「customadapter_properties.xml」にしてください。

カスタムアダプタプロパティファイルの設定内容については，「付録 B.2(2)　カスタムアダプタプロパティファイル」を参照してください。

(d)　カスタムアダプタ用アプリケーション統合属性ファイル

カスタムアダプタ用アプリケーション統合属性ファイルを用いることで，運用する環境に合わせて，汎用カスタムアダプタのパラメタを設定できます。カスタムアダプタ用アプリケーション統合属性ファイルを指定する場合，ファイル名は必ず「cscadapter_property.xml」にしてください。

アプリケーション統合属性ファイルの設定内容については，「付録 B.2(3)　カスタムアダプタ用アプリケーション統合属性ファイル」を参照してください。

アプリケーション統合属性ファイルを指定した場合，Session Bean 属性の<pooled-instance>はアプリケーション統合属性ファイルで記載した値となります。

(e)　カスタムアダプタ定義ファイル

カスタムアダプタ定義ファイルを用いることで，データ変換時に電文の構造変換をスキップする機能（構造変換スキップ機能）を利用できます。カスタムアダプタ定義ファイルを指定する場合，ファイル名は必ず「csccustomadapter.properties」にしてください。

カスタムアダプタ定義ファイルの設定内容については，「付録 B.2(4)　カスタムアダプタ定義ファイル」を参照してください。

構造変換スキップ機能の設定方法について説明します。

要求電文のデータ変換では，ビジネスプロセスから受け取った DOM ツリーから，直接，バイナリ形式の要求電文を生成し，汎用カスタムアダプタに渡したいときに，構造変換スキップ機能を有効にできます。

応答電文のデータ変換では，汎用カスタムアダプタから返ってきたバイナリ形式の応答電文から生成した DOM ツリーを，そのままビジネスプロセスに返したいときに，構造変換スキップ機能を有効にできます。

構造変換スキップ機能を設定するには，開発環境のサービスアダプタ定義（詳細）画面で，［独自定義ファイル］にカスタムアダプタ定義ファイルを設定する必要があります。

サービスアダプタ定義画面については，マニュアル「サービスプラットフォーム リファレンス」の「1.2.2 サービスアダプタ定義画面」を参照してください。

構造変換スキップ機能を使用するときの注意事項を次に示します。

- 開発環境のサービスアダプタ定義画面で，要求電文および応答電文の標準電文フォーマットに XML フォーマット定義ファイルを，要求電文および応答電文のサービス電文フォーマットにバイナリフォーマット定義ファイルを指定してください。
- 開発環境のサービスアダプタ定義画面で，要求電文および応答電文の標準電文フォーマットには，受付電文フォーマット（バイナリフォーマット定義ファイル）から cscfdx2xsd コマンドを使用して生成した XML フォーマット定義ファイルを指定してください。

(4)　クラスファイルの作成

プロトコルコンバータ（プロトコル変換処理）を実装するクラスファイルを作成します。作成するクラスファイルは，カスタムアダプタ開発フレームワークが提供する CSCMsgCustomProtocolConverter インターフェースを実装します。

プロトコルコンバータは，サービスプラットフォーム上で動く Stateless SessionBean の一部として動作します。そのため，クラスファイルを作成するときは，マニュアル「アプリケーションサーバ 機能解説 基本・開発編（EJB コンテナ）」の EJB や Stateless Session Bean に関する説明を参照してください。

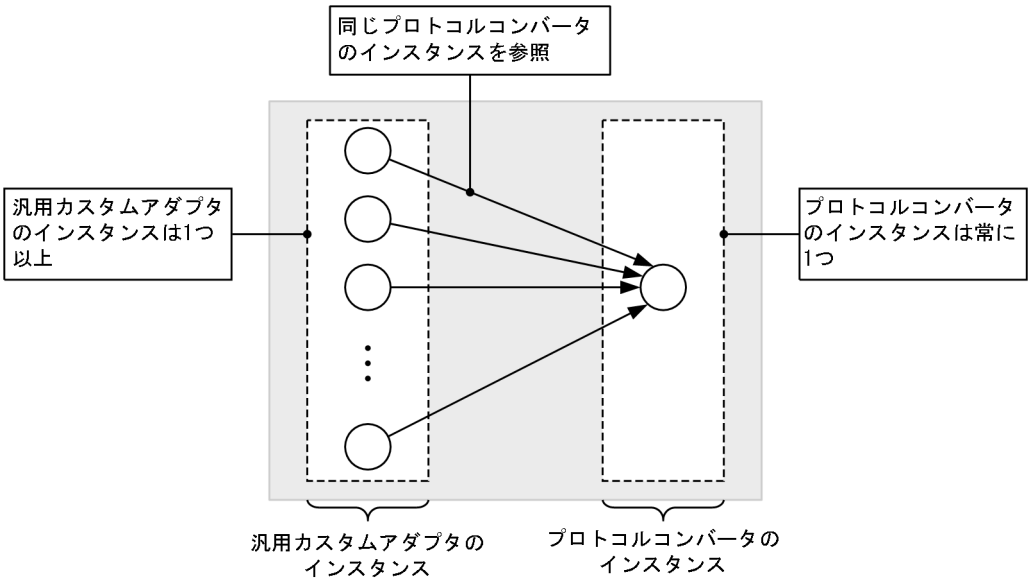
CSCMsgCustomProtocolConverter インターフェースの詳細は，「付録 B.1(1)(a) CSCMsgCustomProtocolConverter インターフェース」を参照してください。また，実装例については，「付録 B.3　カスタムアダプタ開発フレームワークのサンプルプログラム」を参照してください。

プロトコルコンバータを実装するときに考慮する内容を次に示します。

(a)　汎用カスタムアダプタのインスタンスとプロトコルコンバータのインスタンスの関係

1 つの汎用カスタムアダプタは複数のインスタンスを持つことがありますが，その場合でもプロトコルコンバータのインスタンスは常に 1 つです。そのため，プロトコルコンバータはマルチスレッドでアクセスされることに注意して実装してください。

図 3-31　汎用カスタムアダプタのインスタンスとプロトコルコンバータのインスタンスの関係



（凡例）　◯：インスタンス

(b)　リソースの確保方法

カスタムアダプタ開発フレームワークでは，プロトコルコンバータでのリソースの確保について，次の方法を推奨しています。

- 静的なリソースの取得：CSCMsgCustomProtocolConverter の start メソッド

  サービス部品の呼び出しに必要なプロトコルのヘッダ情報やサービス部品のアドレスなど，サービス部品を呼び出すときに参照だけが行われるリソースについては，start メソッドでインスタンスフィールドに確保することを推奨しています。

  これは，汎用カスタムアダプタ開始時に一度だけ取得すれば，以後同じリソースを使い続けることができるためです。

- 動的なリソース（スレッドセーフでないリソース）の取得：CSCMsgCustomProtocolConverter の invoke メソッド

  ストリームによるファイルアクセスやサービス部品に対するコネクションなど，スレッドセーフにアクセスできないリソースについては，invoke メソッドでサービス部品を呼び出すごとに取得することを推奨しています。

  スレッドセーフにアクセスできないリソースをインスタンスフィールドに確保してしまうと，スレッド間の排他制御が必要になります。

  サービス部品を呼び出すごとにリソースを取得すれば，スレッド間の排他制御による処理性能の低下を抑止できます。

注意事項

一度確保したリソースは，処理が終了しても保持し続けるため，システム全体に高い負荷がかかった場合に，Java ヒープ不足や Metaspace 領域不足などが原因で OutOfMemoryError が

発生することがあります。そのため，OutOfMemoryError が発生したときに適切にリソースを解放する処理や，ロールバックなどのエラー処理を実装しておく必要があります。

(c)　リソースファイルのファイル名指定方法

プロトコルコンバータでは，アダプタコンテキストを用いてアーカイブファイル内のリソースファイルにアクセスできます。アーカイブファイル内のリソースファイルにアクセスする場合，次の点に注意してください。

- 指定するファイル名は，アーカイブ内の相対パスを含むファイル名を指定してください。
- 独自定義ファイルのファイル名と，アーカイブファイル内のリソースファイル名が重なった場合，独自定義ファイルを指定したことになります。
- 複数のアーカイブファイル内で，同一名称のリソースファイルを配置しないでください。ファイル名が重複していると，どのアーカイブ内のファイルが使用されたのか特定できなくなります。

(d)　フォルト情報と例外の違い

CSCMsgCustomProtocolConverter の invoke メソッドの実行中にエラーが起きた場合，プロトコルコンバータは例外を生成して送出，またはフォルト情報を応答電文メッセージに格納します。

フォルト情報は，サービス部品呼び出し処理（CSCMsgCustomProtocolConverter の invoke メソッド）の実行中に発生した業務上の障害を，カスタムアダプタ開発フレームワークに伝達するために使用します。

例外は，サービス部品呼び出し処理中に発生した通信エラーなどのシステム障害を，カスタムアダプタ開発フレームワークに伝達するために使用します。

フォルトを使用した場合と例外を使用した場合とでは，障害情報を受け取るサービスアダプタやビジネスプロセスでは次のような違いがあります。

●フォルト情報

- サービスリクエスタへの応答

  フォルト情報の取得については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「9.2.8　エラー情報を取得する」または「9.4.7　エラー情報を取得する」を参照してください。
- ビジネスプロセスへの応答

  ビジネスプロセスでは，サービスプラットフォームが提供するほかのサービスアダプタのフォルトと同様に処理できます。ビジネスプロセスでのフォルト処理については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.4.4　フォルト処理を定義する」を参照してください。

参考

カスタムアダプタ開発フレームワークに伝達するフォルト情報は，SOAP の SOAP Fault 情報に対応する要素を持ちます。各要素の使用方法も SOAP の仕様に準拠しています。SOAP 通信基盤

がサポートする SOAP および SAAJ の仕様に従って記述してください。SOAP 通信基盤のサポートする SOAP および SAAJ の仕様については，マニュアル「アプリケーションサーバ SOAP アプリケーション開発の手引」の「12. 標準仕様との対応」を参照してください。

●例外

- サービスリクエスタへの応答
  例外情報の取得については，サービスリクエスタ作成でのエラー情報の取得に関する内容を参照してください。
- ビジネスプロセスへの応答
  ビジネスプロセスでは，フォルト処理はできないため，エラー（例外）として扱われます。

(e) コンパイル

カスタムアダプタ開発フレームワークで使用する次の JAR ファイルをクラスパスに指定してコンパイルする必要があります。

```
<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>¥CSC¥lib¥csc_adapter.jar
```

また，サービスプラットフォームの API などを使う場合は，別途必要な JAR ファイルを追加してコンパイルしてください。

(5) JAR ファイルの作成

汎用カスタムアダプタの開発で作成する JAR ファイルの総称を次のように分けています。

- プロトコルコンバータ JAR ファイル
  プロトコルコンバータクラスファイルおよびカスタムアダプタ開発フレームワーク動作定義ファイルを含む JAR ファイル
- ライブラリ JAR ファイル
  プロトコルコンバータ JAR ファイル以外の JAR ファイル

それぞれの JAR ファイルの作成方法について説明します。

(a) プロトコルコンバータ JAR ファイルの作成

jar コマンドを使用して，プロトコルコンバータクラスファイルとそれが使用するクラスファイルおよび，カスタムアダプタ開発フレームワーク動作定義ファイルを JAR ファイルにアーカイブします。

JAR ファイルにアーカイブする場合のディレクトリ構成，および jar コマンドの指定例を次に示します。

●JAR ファイルのディレクトリ構造

jar コマンドによってアーカイブするときの JAR ファイルのディレクトリ構成を次の表に示します。

表 3-141　jar コマンドによってアーカイブするときの JAR ファイルのディレクトリ構成

ディレクトリ名 またはファイル名	説明	指定
/	アーカイブ内のルートディレクトリです。	○
*.class	プロトコルコンバータおよびそれらが依存するクラス（インターフェース）のクラスファイル（*.class）をパッケージ名に従ったディレクトリ階層で格納します。	○
framework_properties.xml	カスタムアダプタ開発フレームワークが使用するプロトコルコンバータのクラス名（パッケージ名を含む）を指定します。	○※
*	プロトコルコンバータが使用する任意のリソースファイルです。 ディレクトリを作成して，その中にファイルを格納できます。	－
/META-INF/	管理情報を格納するディレクトリです。 jar コマンドによって自動的に作成されます。	－
MANIFEST.MF	マニフェストファイルです。 jar コマンドの m オプションを指定することで，アーカイブ内に自動的に作成されます。	－

（凡例）

○：アーカイブするときに指定が必須であることを示します。

－：アーカイブするときに指定が不要，または任意であることを示します。

注※

JAR ファイルが複数の場合は，プロトコルコンバータを格納した JAR ファイルにだけ格納してください。

●JAR ファイル作成時の jar コマンド指定例

JAR ファイルにアーカイブする場合の jar コマンドの指定例を次に示します。

- 前提

  カレントディレクトリおよびサブディレクトリに次のようにファイルが用意されているとします。

```
framework_properties.xml
protocolconverter/CustomProtocolConverter.class
lib/MyLib.class
data/table.dat
```

- 指定例

  「MyProtocolConverter.jar」という名称の JAR ファイルを作成する場合の jar コマンドの指定例を次に示します。

```
jar cf ..¥MyProtocolConverter.jar .¥
```

(b)　ライブラリ JAR ファイルの作成

ライブラリ JAR ファイルは一般の JAR ファイルと同じように作成してください。

ライブラリ JAR ファイルの作成時の注意事項を次に示します。

●JAR ファイルのファイル構造

複数の JAR ファイル間でファイル名は重複しないようにしてください。

ファイル名が重複していると，どの JAR ファイルのファイルが使用されたのか特定できなくなります。なお，ディレクトリだけが重複している場合は，問題ありません。

A.jar と B.jar という 2 つの JAR ファイルを使用し，アーカイブ内のディレクトリおよびファイル構造が重複した場合の例を次の表に示します。

表 3-142　アーカイブ内のディレクトリおよびファイル構造が重複した場合の例

A.jar の構造	B.jar の構造	説明
myfile.txt	－	A.jar のファイル「myfile.txt」が使用されます。
samedata.dat	samedata.dat	A.jar と B.jar のどちらのファイル「samedata.dat」が使用されたのか特定できません。ファイル名は重複させないでください。
dir/	dir/	ディレクトリの重複はかまいません。
－	dir/b.class	B.jar のファイル「dir/b.class」が使用されます。
dir/same.class	dir/same.class	A.jar と B.jar のどちらのファイル「def/same.class」が使用されたのか特定できません。ファイル名は重複させないでください。

（凡例）

－：存在しないことを示します。

●JAR ファイルの配置

ファイル名が「framework_properties.xml」であるファイルを JAR ファイルのルートディレクトリに配置しないでください。

「framework_properties.xml」と同じファイル名の JAR ファイルがルートディレクトリに存在すると，プロトコルコンバータ JAR ファイル内のファイル「framework_properties.xml」が使用されなくなるおそれがあります。

(6)　EAR ファイルの作成

汎用カスタムアダプタの新規登録に必要な EAR ファイルを作成します。

EAR ファイルの作成は，jar コマンドを使用し，サービスプラットフォームが提供している JAR ファイルを EAR ファイルにアーカイブします。

EAR ファイルにアーカイブする場合のディレクトリ構成，および jar コマンドの指定例を次に示します。

●EAR ファイルのディレクトリ構造

jar コマンドによってアーカイブするときの EAR ファイルのディレクトリ構成を次の表に示します。

表 3-143　jar コマンドによってアーカイブするときの EAR ファイルのディレクトリ構造

ディレクトリ名 またはファイル名	説明	指定
/	アーカイブ内のルートディレクトリです。	○
cscmsg_adpejb.jar	次に示すファイルのコピーを格納します。 <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>¥CSC¥lib¥cscmsg_adpejb.jar	○
/META-INF/	管理情報を格納するディレクトリです。 jar コマンドによって自動的に作成されます。	－
MANIFEST.MF	マニフェストファイルです。 jar コマンドの m オプションを指定することで，アーカイブ内に自動的に作成されます。	－

（凡例）

○：アーカイブするときに指定が必須であることを示します。

－：アーカイブするときに指定が不要，または任意であることを示します。

注意事項

上記以外のファイルは格納しないでください。複数の jar ファイルを格納した場合，汎用カスタムアダプタの登録ができなくなります。

●EAR ファイル作成時の jar コマンド指定例

EAR ファイルにアーカイブする場合の jar コマンドの指定例を次に示します。

- 前提

  カレントディレクトリに次に示すファイルのコピーを格納します。

  <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>¥CSC¥lib¥cscmsg_adpejb.jar

- 指定例

  「CustomAdapter.ear」という名称の EAR ファイルを作成する場合の jar コマンドの指定例を次に示します。

```
jar cf ..¥CustomAdapter.ear .¥
```

(7)　EAR ファイルの入れ替え

汎用カスタムアダプタの追加時に EAR ファイルを設定します。汎用カスタムアダプタを追加したあとに EAR ファイルを修正する必要がある場合，汎用カスタムアダプタのサービスアダプタ定義画面で，EAR ファイルを入れ替えます。

入れ替える EAR ファイルに含まれる EJB-JAR ファイル名は，汎用カスタムアダプタの追加時に設定した EAR ファイルの EJB-JAR 名と同じにしてください。EJB-JAR ファイル名が異なる場合は，EAR ファイルを指定したときにエラーが発生し，EAR ファイルを設定できません。

汎用カスタムアダプタのサービスアダプタ定義画面で，EAR ファイルを入れ替える手順を次に示します。

1. ツリービューのサービス定義一覧で，EAR ファイルを入れ替える汎用カスタムアダプタをダブルクリックします。
   サービスアダプタ定義画面が表示されます。

2. 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。

3. 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。

4. 入れ替える EAR ファイルを指定します。

(8)　サービスアダプタ定義画面での操作

汎用カスタムアダプタを定義する場合の手順を次に示します。

1. サービスアダプタ定義画面を表示します。
   サービスアダプタ定義画面の表示方法については，「3.3.1(4)　サービスアダプタ定義画面の表示」を参照してください。

2. サービス部品制御情報の［サービス名］，［サービス ID］，および［最大インスタンス数］を必要に応じて編集します。

3. サービス部品制御情報の［追加］ボタンをクリックして，オペレーションを追加します。
   「3.3.18(1)　オペレーションの設計」で設計したオペレーション名を指定してください。

4. サービス部品制御情報の［オペレーション］のドロップダウンリストから，編集したいオペレーションを選択します。

5. サービス部品制御情報の［システム例外をフォルトに変換する］チェックボックスを必要に応じてチェックします。

6. オペレーション情報の［通信モデル］のドロップダウンリストから，「同期」または「非同期」を選択します。
   「3.3.18(1)　オペレーションの設計」で設計した通信モデルを指定してください。

7. 次に示す操作をします。

   標準電文のフォーマットを指定する場合

   手順 8.～11.を実施して，手順 12.に進みます。

   標準電文のフォーマットを指定しない場合

   手順 12.に進みます。

8. 要求電文の［>>］ボタンをクリックします。
   標準電文およびデータ変換定義の情報欄が表示されます。

9. 標準電文の［使う］チェックボックスをチェックします。

10. 標準電文の［参照］ボタンをクリックして，［電文フォーマット］に標準電文のフォーマットを指定します。

［電文フォーマット］には電文種別は，XML 電文とバイナリ電文のどちらも指定できます。

11. 標準電文の［表示］ボタンをクリックします。

標準電文のフォーマットが表示されます。必要に応じて指定した標準電文のフォーマットを確認してください。

12. サービス部品電文の［参照］ボタンをクリックして，［電文フォーマット］にサービス部品電文のフォーマットを指定します。

「3.3.18(1)　オペレーションの設計」で設計した電文フォーマットを指定してください。

［電文フォーマット］には電文種別は，XML 電文とバイナリ電文のどちらも指定できます。

13. サービス部品電文の［表示］ボタンをクリックします。

サービス部品電文のフォーマットが表示されます。必要に応じて指定したサービス部品電文のフォーマットを確認してください。

14. 次に示す操作をします。

標準電文の［使う］チェックボックスをチェックした場合

手順 15.～17.を実施して，手順 18.に進みます。

標準電文の［使う］チェックボックスをチェックしない場合

手順 18.に進みます。

15. データ変換定義のファイル名を入力します。

16. ［編集］ボタンをクリックします。

データ変換定義画面が表示されます。

なお，初めて定義する場合は，［ルート要素選択］ダイアログが表示されます。

また，電文フォーマットを変更した場合は，電文フォーマットの変更を反映するかを確認するダイアログが表示されます。詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.3.2　電文フォーマットを変更した場合の定義手順」を参照してください。

17. 標準電文とサービス部品電文の内容をマッピングします。

18. 手順 6.で「同期」を選択した場合，応答電文についても同様に，手順 7.から手順 17.までを実施します。

19. サービスアダプタ定義（詳細）タブをクリックします。

20. サービスアダプタ（EJB-JAR ファイル）の名称を確認します。

名称が「cscmsg_adpejb.jar」であることを確認します。

21. ユーティリティクラス（JAR ファイル）の［追加］ボタンをクリックして，JAR ファイルを登録します。

追加するファイルは，「3.3.18(5)(a)　プロトコルコンバータ JAR ファイルの作成」で作成した JAR ファイルです。

必要に応じて，プロトコルコンバータで使用する JAR ファイル（ライブラリ JAR ファイルなど）も追加できます。

22. 独自定義ファイルの［追加］ボタンをクリックして，必要に応じて次のファイルを追加します。
    - プロトコルコンバータ用独自定義ファイル
    - カスタムアダプタプロパティファイル
    - カスタムアダプタ用アプリケーション統合属性ファイル
23. 必要に応じて，独自定義ファイルを選択して［編集］ボタンをクリックし，ファイルの内容を編集します。
24. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

3.3.19　サービスアダプタ定義での設定一覧

サービスアダプタの定義で使用するサービスアダプタ定義画面での設定について説明します。

注意事項

- サービスアダプタ定義画面で指定する電文フォーマット定義ファイルが，外部の XML スキーマを参照している電文フォーマットの場合，必ずルートスキーマに当たるファイルを設定してください。ルートスキーマから参照される外部の XML スキーマファイルは，自動的に取り込まれます。
- サービスアダプタ定義画面で指定する電文フォーマット定義ファイルは，一定の条件を満たしている必要があります。スキーマの条件の詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.6.5　XML スキーマの適用範囲」を参照してください。

(1)　SOAP アダプタの場合

SOAP アダプタを定義する場合の，サービスアダプタ定義画面での設定を次の表に示します。

表 3-144　SOAP アダプタのサービスアダプタ定義画面での設定

分類		項目	設定
基本画面	サービス部品制御情報	サービス名	◎
		サービス ID	◎
		サービス種別	×

分類			項目	設定
基本画面	サービス部品制御情報		アドレス	×
			最大インスタンス数	◎
			サービスクラス名	×
			オペレーション	◎
			システム例外をフォルトに変換する（チェックボックス）	○
	オペレーション情報		オペレーション名	×
			通信モデル	×
	要求電文	ヘッダ	ルート要素	○
			名前空間	×
			電文フォーマット	×
		ボディ	使う（チェックボックス）	○
			標準電文フォーマット	△※1
			サービス部品電文フォーマット	◎
			データ変換定義	△※1
	応答電文	ヘッダ	ルート要素	○
			名前空間	×
			電文フォーマット	×
		ボディ	使う（チェックボックス）	○
			標準電文フォーマット	△※2
			サービス部品電文フォーマット	◎
			データ変換定義	△※2
	フォルト電文		フォルト名	△※3
			フォルト電文フォーマット	△※3
詳細画面	Web サービス制御情報		クライアント定義ファイル	◎
			使う（チェックボックス）	○
			ユーザ名	△※4
			パスワード	△※4
			パスワード（確認）	△※4
	SessionBean 制御情報		クライアント定義ファイル	－
			ユーザ定義クラス（JAR ファイル）	－

分類		項目	設定
詳細画面	MDB 制御情報	JMS メッセージ種別	－
		リモート呼出し（ラジオボタン）	－
		ローカル呼出し（ラジオボタン）	－
		最大メッセージ数	－
		宛先 URL	－
		ベーシック認証（チェックボックス）	－
		ユーザ名	－
		パスワード	－
		パスワード（確認）	－
		転送キュー作成先 RD エリア名	－
		順序保証（チェックボックス）	－

（凡例）

◎：必ず設定します。

○：任意で設定します。

△：場合によっては必ず設定します。

×：表示されている内容を確認します。

－：該当しません。

注※1

要求電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注※2

応答電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注※3

サービス部品電文のフォルト電文があるときにだけ設定します。

注※4

ベーシック認証の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注意事項

サービスアダプタ定義画面では，入力項目の前後に半角スペースや全角スペースを指定できません。

(2)　SessionBean アダプタの場合

SessionBean アダプタを定義する場合の，サービスアダプタ定義画面での設定を次の表に示します。

表 3-145　SessionBean アダプタのサービスアダプタ定義画面での設定

分類			項目	設定
基本画面	サービス部品制御情報		サービス名	◎
			サービス ID	◎
			サービス種別	×
			アドレス	◎
			最大インスタンス数	◎
			サービスクラス名	×
			オペレーション	◎
			システム例外をフォルトに変換する（チェックボックス）	○
	オペレーション情報		オペレーション名	×
			通信モデル	×
	要求電文	ヘッダ	ルート要素	−
			名前空間	−
			電文フォーマット	−
		ボディ	使う（チェックボックス）	○
			標準電文フォーマット	△※1
			サービス部品電文フォーマット	◎
			データ変換定義	△※1
	応答電文	ヘッダ	ルート要素	−
			名前空間	−
			電文フォーマット	−
		ボディ	使う（チェックボックス）	○
			標準電文フォーマット	△※2
			サービス部品電文フォーマット	◎
			データ変換定義	△※2
	フォルト電文		フォルト名	−
			フォルト電文フォーマット	−
詳細画面	Web サービス制御情報		クライアント定義ファイル	−
			使う（チェックボックス）	−
			ユーザ名	−

分類		項目	設定
詳細画面	Web サービス制御情報	パスワード	－
		パスワード（確認）	－
	SessionBean 制御情報	クライアント定義ファイル	◎
		ユーザ定義クラス（JAR ファイル）	○
	MDB 制御情報	JMS メッセージ種別	－
		リモート呼出し（ラジオボタン）	－
		ローカル呼出し（ラジオボタン）	－
		最大メッセージ数	－
		宛先 URL	－
		ベーシック認証（チェックボックス）	－
		ユーザ名	－
		パスワード	－
		パスワード（確認）	－
		転送キュー作成先 RD エリア名	－
		順序保証（チェックボックス）	－

（凡例）

◎：必ず設定します。

○：任意で設定します。

△：場合によっては必ず設定します。

×：表示されている内容を確認します。

－：該当しません。

注※1

要求電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注※2

応答電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注意事項

サービスアダプタ定義画面では，入力項目の前後に半角スペースや全角スペースを指定できません。

(3) MDB（WS-R）アダプタの場合

MDB（WS-R）アダプタを定義する場合の，サービスアダプタ定義画面での設定を次の表に示します。

表 3-146　MDB（WS-R）アダプタのサービスアダプタ定義画面での設定

分類			項目	設定
基本画面	サービス部品制御情報		サービス名	◎
			サービス ID	◎
			サービス種別	×
			アドレス	◎
			最大インスタンス数	◎
			サービスクラス名	×
			オペレーション	◎
			システム例外をフォルトに変換する（チェックボックス）	○
	オペレーション情報		オペレーション名	×
			通信モデル	×
	要求電文	ヘッダ	ルート要素	−
			名前空間	−
			電文フォーマット	−
		ボディ	使う（チェックボックス）	○
			標準電文フォーマット	△※1
			サービス部品電文フォーマット	◎
			データ変換定義	△※1
	応答電文	ヘッダ	ルート要素	−
			名前空間	−
			電文フォーマット	−
		ボディ	使う（チェックボックス）	−
			標準電文フォーマット	−
			サービス部品電文フォーマット	−
			データ変換定義	−
	フォルト電文		フォルト名	−
			フォルト電文フォーマット	−
詳細画面	Web サービス制御情報		クライアント定義ファイル	−
			使う（チェックボックス）	−
			ユーザ名	−

分類		項目	設定
詳細画面	Web サービス制御情報	パスワード	−
		パスワード（確認）	−
	SessionBean 制御情報	クライアント定義ファイル	−
		ユーザ定義クラス（JAR ファイル）	−
	MDB 制御情報	JMS メッセージ種別	◎
		リモート呼出し（ラジオボタン）	◎※2
		ローカル呼出し（ラジオボタン）	◎※2
		最大メッセージ数	△※3
		宛先 URL	△※3
		ベーシック認証（チェックボックス）	○※4
		ユーザ名	△※5
		パスワード	△※5
		パスワード（確認）	△※5
		転送キュー作成先 RD エリア名	○※4
		順序保証（チェックボックス）	○※4

（凡例）

◎：必ず設定します。

○：任意で設定します。

△：場合によっては必ず設定します。

×：表示されている内容を確認します。

−：該当しません。

注※1

要求電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注※2

［リモート呼出し］または［ローカル呼出し］のどちらか一方のラジオボタンを必ず選択します。

注※3

［リモート呼出し］ラジオボタンを選択した場合，必ず設定します。

注※4

［リモート呼出し］ラジオボタンを選択した場合にだけ，任意で設定できます。

注※5

［リモート呼出し］（ラジオボタン）を選択し，ベーシック認証の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注意事項

サービスアダプタ定義画面では，入力項目の前後に半角スペースや全角スペースを指定できません。ただし，MDB（WS-R）の詳細画面の［転送キュー作成先 RD エリア名］には，末尾に半角スペースを指定できます。

(4) MDB（DB キュー）アダプタの場合

MDB（DB キュー）アダプタを定義する場合の，サービスアダプタ定義画面での設定を次の表に示します。

表 3-147 MDB（DB キュー）アダプタのサービスアダプタ定義画面での設定

分類			項目	設定
基本画面	サービス部品制御情報		サービス名	◎
			サービス ID	◎
			サービス種別	×
			アドレス	◎
			最大インスタンス数	◎
			サービスクラス名	×
			オペレーション	◎
			システム例外をフォルトに変換する（チェックボックス）	○
	オペレーション情報		オペレーション名	×
			通信モデル	×
	要求電文	ヘッダ	ルート要素	−
			名前空間	−
			電文フォーマット	−
		ボディ	使う（チェックボックス）	○
			標準電文フォーマット	△※
			サービス部品電文フォーマット	◎
			データ変換定義	△※
	応答電文	ヘッダ	ルート要素	−
			名前空間	−
			電文フォーマット	−
		ボディ	使う（チェックボックス）	−
			標準電文フォーマット	−

分類			項目	設定
基本画面	応答電文	ボディ	サービス部品電文フォーマット	－
			データ変換定義	－
	フォルト電文		フォルト名	－
			フォルト電文フォーマット	－
詳細画面	Web サービス制御情報		クライアント定義ファイル	－
			使う（チェックボックス）	－
			ユーザ名	－
			パスワード	－
			パスワード（確認）	－
	SessionBean 制御情報		クライアント定義ファイル	－
			ユーザ定義クラス（JAR ファイル）	－
	MDB 制御情報		JMS メッセージ種別	－
			リモート呼出し（ラジオボタン）	－
			ローカル呼出し（ラジオボタン）	－
			最大メッセージ数	－
			宛先 URL	－
			ベーシック認証（チェックボックス）	－
			ユーザ名	－
			パスワード	－
			パスワード（確認）	－
			転送キュー作成先 RD エリア名	－
			順序保証（チェックボックス）	－

（凡例）

◎：必ず設定します。

○：任意で設定します。

△：場合によっては必ず設定します。

×：表示されている内容を確認します。

－：該当しません。

注※

要求電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注意事項

サービスアダプタ定義画面では，入力項目の前後に半角スペースや全角スペースを指定できません。

(5) DB アダプタの場合

DB アダプタを定義する場合の，サービスアダプタ定義画面での設定を次の表に示します。

表 3-148 DB アダプタのサービスアダプタ定義画面での設定

分類		項目	設定	
			同期	非同期
基本画面	サービス部品制御情報	サービス名	◎	◎
		サービス ID	◎	◎
		サービス種別	×	×
		アドレス	×	×
		最大インスタンス数※1	◎	◎
		サービスクラス名	×	×
		オペレーション	◎	◎
		システム例外をフォルトに変換する（チェックボックス）	○	○
	オペレーション情報	オペレーション名	×	×
		通信モデル	◎※2	◎※3
	要求電文	any 型を使う（チェックボックス）	－	－
		使う（チェックボックス）	○	○
		標準電文フォーマット	△※4	△※4
		サービス部品電文フォーマット※6	◎	◎
		データ変換定義	△※4	△※4
	応答電文	any 型を使う（チェックボックス）	－	－
		使う（チェックボックス）	○	－
		標準電文フォーマット	△※5	－
		サービス部品電文フォーマット※6	◎	－
		データ変換定義	△※5	－
	フォルト電文	フォルト名	－	－
		サービス部品電文フォーマット	－	－

分類		項目	設定	
			同期	非同期
詳細画面	サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	×	×
		ユーティリティクラス（JAR ファイル）	−	−
		独自定義ファイル※7	◎	◎

（凡例）

◎：必ず設定します。

○：任意で設定します。

△：場合によっては必ず設定します。

×：表示されている内容を確認します。

−：該当しません。

注※1

サービスアダプタ定義画面で設定した値は適用されません。最大インスタンス数は，アプリケーション統合属性ファイルの要素「hitachi-application-all-property/ejb-jar/hitachi-session-bean-property/session-runtime/stateless/pooled-instance/maximum」で指定した値が設定されます。

注※2

「同期」を設定します。

注※3

「非同期」を設定します。

注※4

要求電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注※5

応答電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注※6

電文種別が XML 電文の電文フォーマットを設定します。

注※7

設定されている cscadapter_property.xml はテンプレートファイルです。cscadapter_property.xml を選択したあと［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。プロパティファイルの詳細は，「3.3.5(4)　アプリケーション統合属性ファイルの編集」を参照してください。

また，SQL オペレーション定義ファイルを作成して，独自定義ファイルに追加してください。SQL オペレーション定義ファイルの詳細は，「3.3.5(1)　SQL オペレーション定義ファイルの作成」を参照してください。

(6)　TP1 アダプタの場合

TP1 アダプタを定義する場合の，サービスアダプタ定義画面での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-149　TP1 アダプタのサービスアダプタ定義画面（基本）の設定項目

分類	項目		説明	設定
サービス部品制御情報	サービス名		サービスアダプタ定義追加（TP1 アダプタ）ウィザードで指定した HCSC コンポーネントの名称が表示されます。	◎
	サービス ID		HCSC コンポーネントの ID が表示されます。※1	◎
	サービス種別		サービス部品の種別として「TP1 アダプタ」が表示されます。	×
	アドレス		この項目は使用しません。	×
	最大インスタンス数		サービス部品の最大インスタンス数が表示されます。 同時に処理できる接続先システムへの RPC 要求数を指定します。ここで指定する値と TP1 アダプタの通信構成定義の con_pool_num 要素は同じ値を指定してください。	◎
	サービスクラス名		この項目は使用しません。	×
	オペレーション		オペレーション名が表示されます。サービス部品が持つオペレーション名をドロップダウンリストから選択します。	◎
オペレーション情報	−		サービス部品のオペレーションについての情報が表示されます。	−
	オペレーション名		サービス部品制御情報のオペレーション欄で選択したオペレーションの名称が表示されます。	×
	通信モデル		サービス部品制御情報のオペレーション欄で選択したオペレーションの通信モデルを選択します。ドロップダウンリストから次に示す「同期」または「非同期」のどちらかを選択します。 • 同期 応答電文の情報が表示されます。 • 非同期 応答電文の情報は表示されません。非同期のままアダプタ定義を保存すると応答電文のすべての項目は消えます。	◎
要求電文	−		サービスリクエスタから送信される要求電文の情報が表示されます。	−
	標準	使う（チェックボックス）	標準電文をサービス部品電文のフォーマットにデータ変換する場合にチェックします。	○
		電文フォーマット	サービスリクエスタから受け付けた標準電文のフォーマット名を入力します。	△※2
	サービス部品	電文フォーマット	サービス部品電文のフォーマット（TP1 アダプタ用の電文フォーマット定義ファイル）名を入力します。	◎

分類	項目		説明	設定
要求電文	データ変換定義		標準電文からサービス部品電文へデータ変換するための定義ファイル名を入力します。	△※2
応答電文	－		サービス部品から送信される応答電文の情報が表示されます。オペレーション情報の通信モデル欄で，「同期」を選択した場合だけ，設定できます。	－
	標準	使う（チェックボックス）	サービス部品電文を標準電文のフォーマットにデータ変換する場合にチェックします。	○
		電文フォーマット	サービス部品から受け付けた標準電文のフォーマット名を入力します。	△※3
	サービス部品	電文フォーマット	サービス部品の電文フォーマット（TP1 アダプタ用の電文フォーマット定義ファイル）名を入力します。	◎
	データ変換定義		サービス部品電文から標準電文へデータ変換するための定義ファイル名を入力します。	△※3
フォルト電文	－		サービス部品のフォルト電文についての情報が表示されます。	－
	フォルト名		サービス部品のフォルト名が表示されます。	◎
	電文フォーマット		次のフォルト電文フォーマットファイルが設定されています。 • adptp1_faultfmt.xsd※4	×

（凡例）

◎：必ず設定します。

○：任意で設定します。

△：場合によっては必ず設定します。

×：表示されている内容を確認します。

－：該当しません。

注※1

TP1 アダプタでは，サービス ID を基にトレースの出力パスが決められます。このため，サービス ID を編集する場合は，一度作成したことのある TP1 アダプタのサービス ID と同じサービス ID は指定しないでください。なお，サービス ID の大文字と小文字は区別されません。同じサービス ID を指定した場合は，トレースが過去に作成したトレース情報に出力されるため，障害発生時の原因究明が困難になる場合があります。

注※2

要求電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注※3

応答電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注※4

ファイルの格納先ディレクトリは「<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¥custom-adapter¥TP1¥lib」です。

TP1 アダプタを定義する場合の，サービスアダプタ定義（詳細）画面での設定項目を次の表に示します。

表 3-150　TP1 アダプタのサービスアダプタ定義（詳細）画面の設定項目

分類	項目	説明	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	TP1 アダプタ新規作成時に指定したサービス部品についての情報が表示されます。	×
	ユーティリティクラス（JAR ファイル）	TP1 アダプタの EJB-JAR ファイルの名称が表示されます。 TP1 アダプタでは，次のユーティリティクラスが設定されています。 • adptp1pc.jar（プロトコルコンバータアーカイブファイル）※1	×
	独自定義ファイル	TP1 アダプタの環境定義ファイル，および TP1 アダプタの通信構成定義ファイルの名称が表示されます。 TP1 アダプタでは，次の独自定義ファイルが設定されています。 • adptp1_config.xml（TP1 アダプタ通信構成定義ファイル）※2 • customadapter_properties.xml（TP1 アダプタ環境定義ファイル）※3	◎

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥custom-adapter¥TP1¥lib」です。

注※2

設定されている adptp1_config.xml はテンプレートファイルです。必要に応じて adptp1_config.xml を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。TP1 アダプタ通信構成定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.4.3　TP1 アダプタ通信構成定義ファイル」を参照してください。

注※3

設定されている customadapter_properties.xml はテンプレートファイルです。customadapter_properties.xml を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。TP1 アダプタ環境定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.4.2　TP1 アダプタ環境定義ファイル」を参照してください。

(7)　ファイルアダプタの場合

ファイルアダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-151　ファイルアダプタのサービスアダプタ定義画面（基本）の設定項目

分類	項目	説明	設定
サービス部品制御情報	－	ファイルアダプタ新規作成時に指定したサービス部品についての情報が表示されます。	－

分類	項目		説明	設定
サービス部品制御情報	サービス名		サービスアダプタ定義追加（ファイルアダプタ）ウィザードで指定した HCSC コンポーネントの名称が表示されます。	◎
	サービス ID		HCSC コンポーネントの ID が表示されます。※1	◎
	サービス種別		サービス部品の種別として「ファイルアダプタ」が表示されます。	×
	アドレス		この項目は使用しません。	×
	最大インスタンス数		サービス部品の最大インスタンス数が表示されます。	◎
	サービスクラス名		この項目は使用しません。	×
	オペレーション		オペレーション名が表示されます。サービス部品が持つオペレーション名をドロップダウンリストから選択します。	◎
オペレーション情報	−		サービス部品のオペレーションについての情報が表示されます。	−
	オペレーション名		サービス部品制御情報のオペレーション欄で選択したオペレーションの名称が表示されます。	×
	通信モデル		サービス部品制御情報のオペレーション欄で選択したオペレーションの通信モデルを選択します。ファイルアダプタでは「同期」を選択します。	◎
要求電文	−		ビジネスプロセスから送信される要求電文の情報が表示されます。	−
	標準	使う（チェックボックス）	標準電文をサービス部品電文のフォーマットにデータ変換する場合にチェックします。	○
		電文フォーマット	ビジネスプロセスから受け付けた標準電文のフォーマット名を入力します。	△※2
	サービス部品	電文フォーマット	サービス部品電文のフォーマット（ファイルアダプタ用の電文フォーマット定義ファイル）を入力します。 定義パターンが「XML 読み込み」または「バイナリ読み込み」の場合は adpff_read.xsd が設定されています。	△
	データ変換定義		標準電文からサービス部品電文へデータ変換するための定義ファイル名を入力します。	△※2
応答電文	−		サービス部品から送信される応答電文の情報が表示されます。オペレーション情報の通信モデル欄で，「同期」を選択した場合だけ，設定できます。ファイルアダプタでは「同期」を選択するため必ず設定します。	−
	標準	使う（チェックボックス）	サービス部品電文を標準電文のフォーマットにデータ変換する場合にチェックします。	○

分類	項目		説明	設定
応答電文	標準	電文フォーマット	サービス部品から受け付けた標準電文のフォーマット名を入力します。	△※3
	サービス部品	電文フォーマット	サービス部品の電文フォーマット（ファイルアダプタ用の電文フォーマット定義ファイル）名を入力します。 定義パターンが「XML 書き込み」または「バイナリ書き込み」の場合は adpff_result.xsd が設定されています。	△
	データ変換定義		サービス部品電文から標準電文へデータ変換するための定義ファイル名を入力します。	△※3

（凡例）

◎：必ず設定します。

○：任意で設定します。

△：場合によっては必ず設定します。

×：表示されている内容を確認します。

－：該当しません。

注※1

ファイルアダプタでは，サービス ID を基にトレースの出力パスが決められます。このため，サービス ID を編集する場合は，一度作成したことのあるファイルアダプタのサービス ID と同じサービス ID は指定しないでください。なお，サービス ID の大文字と小文字は区別されません。同じサービス ID を指定した場合は，トレースが過去に作成したトレース情報に出力されるため，障害発生時の原因究明が困難になる場合があります。

注※2

要求電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注※3

応答電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

ファイルアダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。

表 3-152　ファイルアダプタのサービスアダプタ定義画面（詳細）の設定項目

分類	項目	説明	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	ファイルアダプタ新規作成時に指定したサービス部品についての情報が表示されます。	×
	ユーティリティクラス（JAR ファイル）	ファイルアダプタの EJB-JAR ファイルの名称が表示されます。 ファイルアダプタでは，次のユーティリティクラスが設定されています。 • adpffpc.jar（プロトコルコンバータアーカイブファイル）※	×
	独自定義ファイル	ファイルアダプタでは独自定義ファイルは使用しません。	－

（凡例）

×：表示されている内容を確認します。

－：該当しません。

注※

ファイルの格納先ディレクトリは「<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¥custom-adapter¥File¥lib」です。

(8) Object Access アダプタの場合

Object Access アダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-153 Object Access アダプタのサービスアダプタ定義画面（基本）の設定項目

分類	項目		説明	設定
サービス部品制御情報	－		Object Access アダプタ新規作成時に指定したサービス部品についての情報が表示されます。	－
	サービス名		サービスアダプタ定義追加（Object Access アダプタ）ウィザードで指定した HCSC コンポーネントの名称が表示されます。	◎
	サービス ID		HCSC コンポーネントの ID が表示されます。※1	◎
	サービス種別		サービス部品の種別として「Object Access アダプタ」が表示されます。	×
	アドレス		この項目は使用しません。	×
	最大インスタンス数		サービス部品の最大インスタンス数が表示されます。	◎
	サービスクラス名		この項目は使用しません。	×
	オペレーション		オペレーション名が表示されます。サービス部品が持つオペレーション名をドロップダウンリストから選択します。 Object Access アダプタで指定できるオペレーション名は「exec2」だけです。	◎
オペレーション情報	－		サービス部品のオペレーションについての情報が表示されます。	－
	オペレーション名		サービス部品制御情報のオペレーション欄で選択したオペレーションの名称が表示されます。	×
	通信モデル		サービス部品制御情報のオペレーション欄で選択したオペレーションの通信モデルを選択します。Object Access アダプタでは「同期」を選択します。	◎
要求電文	－		ビジネスプロセスから送信される要求電文の情報が表示されます。	－
	標準	使う（チェックボックス）	標準電文をサービス部品電文のフォーマットにデータ変換する場合にチェックします。	○

分類	項目		説明	設定
要求電文	標準	電文フォーマット	ビジネスプロセスから受け付けた標準電文のフォーマット名を入力します。	△※2
	サービス部品	電文フォーマット	サービス部品電文のフォーマット（Object Access アダプタ用の電文フォーマット定義ファイル）です。 exec2_request.xsd※3 が設定されています。	×
	データ変換定義		標準電文からサービス部品電文へデータ変換するための定義ファイル名を入力します。	△※2
応答電文	－		サービス部品から送信される応答電文の情報が表示されます。オペレーション情報の通信モデル欄で，「同期」を選択した場合だけ，設定できます。Object Access アダプタでは「同期」を選択するため必ず設定します。	－
	標準	使う（チェックボックス）	サービス部品電文を標準電文のフォーマットにデータ変換する場合にチェックします。	○
		電文フォーマット	サービス部品から受け付けた標準電文のフォーマット名を入力します。	△※4
	サービス部品	電文フォーマット	サービス部品の電文フォーマット（Object Access アダプタ用の電文フォーマット定義ファイル）です。 exec2_response.xsd※3 が設定されています。	×
	データ変換定義		サービス部品電文から標準電文へデータ変換するための定義ファイル名を入力します。	△※4
フォルト電文	－		サービス部品のフォルト電文についての情報が表示されます。	－
	フォルト名		サービス部品のフォルト名が表示されます。	◎
	電文フォーマット		次のフォルト電文フォーマットファイルが設定されています。 • exec2_fault1.xsd※3	×

（凡例）

◎：必ず設定します。

○：任意で設定します。

△：場合によっては必ず設定します。

×：表示されている内容を確認します。

－：該当しません。

注※1

Object Access アダプタでは，サービス ID を基にトレースの出力パスが決められます。このため，サービス ID を編集する場合は，一度作成したことのある Object Access アダプタのサービス ID と同じサービス ID は指定しないでください。なお，サービス ID の大文字と小文字は区別されません。同じサービス ID を指定した場合は，トレースが過去に作成したトレース情報に出力されるため，障害発生時の原因究明が困難になる場合があります。

注※2

要求電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注※3

ファイルの格納先ディレクトリは「<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¥custom-adapter¥OA¥schema」です。

注※4

応答電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

Object Access アダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。

表 3-154　Object Access アダプタのサービスアダプタ定義画面（詳細）の設定項目

分類	項目	説明	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	Object Access アダプタ新規作成時に指定したサービス部品についての情報が表示されます。	×
	ユーティリティクラス（JAR ファイル）	Object Access アダプタの EJB-JAR ファイルの名称が表示されます。 次のユーティリティクラスが設定されています。 • adpoapc.jar（プロトコルコンバータアーカイブファイル）※ また，「3.3.8(4)　ユーザ定義ファイルの作成」で作成した次の jar ファイルをユーティリティクラスに追加します。 • adpoadef.jar（ユーザ定義ファイル）	◎
	独自定義ファイル	Object Access アダプタでは独自定義ファイルは使用しません。	－

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：該当しません。

注※

ファイルの格納先ディレクトリは「<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¥custom-adapter¥OA¥lib」です。

(9)　Message Queue アダプタの場合

Message Queue アダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-155　Message Queue アダプタのサービスアダプタ定義画面（基本）の設定項目

分類	項目	説明	設定
サービス部品制御情報	－	Message Queue アダプタ新規作成時に指定したサービス部品についての情報が表示されます。	－

分類	項目		説明	設定
サービス部品制御情報	サービス名		サービスアダプタ定義追加（Message Queue アダプタ）ウィザードで指定した HCSC コンポーネントの名称が表示されます。	◎
	サービス ID		HCSC コンポーネントの ID が表示されます。※1	◎
	サービス種別		サービス部品の種別として「Message Queue アダプタ」が表示されます。	×
	アドレス		この項目は使用しません。	×
	最大インスタンス数		サービス部品の最大インスタンス数が表示されます。	◎
	サービスクラス名		この項目は使用しません。	×
	オペレーション		オペレーション名が表示されます。サービス部品が持つオペレーション名をドロップダウンリストから選択します。	◎
オペレーション情報	−		サービス部品のオペレーションについての情報が表示されます。	−
	オペレーション名		サービス部品制御情報のオペレーション欄で選択したオペレーションの名称が表示されます。	×
	通信モデル		サービス部品制御情報のオペレーション欄で選択したオペレーションの通信モデルを選択します。Message Queue アダプタでは「同期」を選択します。	◎
要求電文	−		ビジネスプロセスから送信される要求電文の情報が表示されます。	−
	標準	使う（チェックボックス）	標準電文をサービス部品電文のフォーマットにデータ変換する場合にチェックします。	○
		電文フォーマット	ビジネスプロセスから受け付けた標準電文のフォーマット名を入力します。	△※2
	サービス部品	電文フォーマット	サービス部品電文のフォーマット（Message Queue アダプタ用の電文フォーマット定義ファイル）名を入力します。 定義パターンが「メッセージ受信」の場合は，adpmq_rcv_request.xsd※3 が設定されています。 定義パターンが「ブラウズ・メッセージ受信」の場合は，adpmq_browse_request.xsd※3 が設定されています。	△
	データ変換定義		標準電文からサービス部品電文へデータ変換するための定義ファイル名を入力します。	△※2
応答電文	−		サービス部品から送信される応答電文の情報が表示されます。オペレーション情報の通信モデル欄で，「同期」を選択した場合だけ，設定できます。Message Queue アダプタでは「同期」を選択するため必ず設定します。	−

分類	項目		説明	設定
応答電文	標準	使う（チェックボックス）	サービス部品電文を標準電文のフォーマットにデータ変換する場合にチェックします。	○
		電文フォーマット	サービス部品から受け付けた標準電文のフォーマット名を入力します。	△※4
	サービス部品	電文フォーマット	サービス部品の電文フォーマット（Message Queue アダプタ用の電文フォーマット定義ファイル）名を入力します。 定義パターンが「メッセージ送信」の場合は，adpmq_snd_response.xsd※3 が設定されています。	△
	データ変換定義		サービス部品電文から標準電文へデータ変換するための定義ファイル名を入力します。	△※4
フォルト電文	－		サービス部品のフォルト電文についての情報が表示されます。	－
	フォルト名		サービス部品のフォルト名が表示されます。	◎
	電文フォーマット		オペレーションの定義パターンが次の場合に「adpmq_fault_response.xsd」※3 が設定されています。 • メッセージ受信 • ブラウズ・メッセージ受信 • メッセージ送受信	×

（凡例）

◎：必ず設定します。

○：任意で設定します。

△：場合によっては必ず設定します。

×：表示されている内容を確認します。

－：該当しません。

注※1

Message Queue アダプタでは，サービス ID を基にトレースの出力パスが決められます。このため，サービス ID を編集する場合は，一度作成したことのある Message Queue アダプタのサービス ID と同じサービス ID は指定しないでください。なお，サービス ID の大文字と小文字は区別されません。同じサービス ID を指定した場合は，トレースが過去に作成したトレース情報に出力されるため，障害発生時の原因究明が困難になる場合があります。

注※2

要求電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注※3

ファイルの格納先ディレクトリは「<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¥custom-adapter¥MQ¥schema」です。

注※4

応答電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

Message Queue アダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。

表 3-156　Message Queue アダプタのサービスアダプタ定義画面（詳細）の設定項目

分類	項目	説明	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	Message Queue アダプタ新規作成時に指定したサービス部品についての情報が表示されます。	×
	ユーティリティクラス（JAR ファイル）	Message Queue アダプタの EJB-JAR ファイルの名称が表示されます。 次のユーティリティクラスが設定されています。 • adpmqpc.jar（プロトコルコンバータアーカイブファイル）※1	×
	独自定義ファイル	次の独自定義ファイルが設定されています。 • customadapter_properties.xml（Message Queue アダプタ環境定義ファイル）※2 • adpmq_communication.xml（Message Queue アダプタ通信構成定義ファイル）※3 • cscadapter_property.xml（アプリケーション統合属性ファイル）※4	◎

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥custom-adapter¥MQ¥lib」です。

注※2

設定されている customadapter_properties.xml（Message Queue アダプタ環境定義ファイル）はテンプレートファイルです。customadapter_properties.xml を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。Message Queue アダプタ環境定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.8.1　Message Queue アダプタ環境定義ファイル」を参照してください。

注※3

設定されている adpmq_communication.xml（Message Queue アダプタ通信構成定義ファイル）はテンプレートファイルです。必要に応じて adpmq_communication.xml を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。Message Queue アダプタ通信構成定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.8.2　Message Queue アダプタ通信構成定義ファイル」を参照してください。

注※4

設定されている cscadapter_property.xml（アプリケーション統合属性ファイル）はテンプレートファイルです。cscadapter_property.xml を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。アプリケーション統合属性ファイルの詳細は，「3.3.9(6)　アプリケーション統合属性ファイルの編集」を参照してください。

(10)　FTP アダプタの場合

FTP アダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-157　FTP アダプタのサービスアダプタ定義画面（基本）の設定項目

分類		項目	説明	設定
サービス部品制御情報		オペレーション	次の中から使用するオペレーションを追加します。 PUT，MPUT：PUT 系（STOR または APPE）処理 GET，MGET：GET 系（RETR）処理 GETINFO：GETINFO 系（LIST または NLST）処理 LOGIN：ログインセッションの確立処理 LOGOUT：ログインセッションの解放処理 RENAME：ファイルのリネーム処理	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	サービス部品	電文フォーマット	［オペレーション］の設定に応じて，次の電文フォーマット定義ファイルが設定されています。 • 定義パターンが「ファイル送信（PUT オペレーション）」の場合 ftpadp_put_request.xsd※1 • 定義パターンが「複数ファイル送信（MPUT オペレーション）」の場合 ftpadp_mput_request.xsd※1 • 定義パターンが「ファイル受信送信（GET オペレーション）」の場合 ftpadp_get_request.xsd※1 • 定義パターンが「複数ファイル受信送信（MGET オペレーション）」の場合 ftpadp_mget_request.xsd※1 • 定義パターンが「ファイル情報取得（GETINFO オペレーション）」の場合 ftpadp_getinfo_request.xsd※1※2 • 定義パターンが「ログインセッション確立（LOGIN オペレーション）」の場合 ftpadp_login_request.xsd※1 • 定義パターンが「ログインセッション解放（LOGOUT オペレーション）」の場合 ftpadp_logout_request.xsd※1 • 定義パターンが「ファイルのリネーム（RENAME オペレーション）」の場合 ftpadp_rename_request.xsd※1	×
応答電文	サービス部品	電文フォーマット	［オペレーション］の設定に応じて，次の電文フォーマット定義ファイルが設定されています。 • 定義パターンが「ファイル送信（PUT オペレーション）」の場合 ftpadp_put_response.xsd※1	×

分類		項目	説明	設定
応答電文	サービス部品	電文フォーマット	• 定義パターンが「複数ファイル送信（MPUT オペレーション）」の場合 ftpadp_mput_response.xsd※1 • 定義パターンが「ファイル受信送信（GET オペレーション）」の場合 ftpadp_get_response.xsd※1 • 定義パターンが「複数ファイル受信送信（MGET オペレーション）」の場合 ftpadp_mget_response.xsd※1 • 定義パターンが「ファイル情報取得（GETINFO オペレーション）」の場合 ftpadp_getinfo_response.xsd※1※2 • 定義パターンが「ログインセッション確立（LOGIN オペレーション）」の場合 ftpadp_login_response.xsd※1 • 定義パターンが「ログインセッション解放（LOGOUT オペレーション）」の場合 ftpadp_logout_response.xsd※1 • 定義パターンが「ファイルのリネーム（RENAME オペレーション）」の場合 ftpadp_rename_response.xsd※1	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥custom-adapter¥FTP¥schema」です。

注※2

Service Platform 09-50-01 以前のバージョンで定義した FTP アダプタを編集する場合，電文フォーマットを指定し直す必要があります。電文フォーマットの［参照］ボタンをクリックし，ftpadp_getinfo_request.xsd（または ftpadp_getinfo_response.xsd）を指定して電文フォーマットに再設定してください。

FTP アダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-158　FTP アダプタのサービスアダプタ定義画面（詳細）の設定項目

分類	項目	説明	設定
サービスアダプタ制御情報	ユーティリティクラス（JAR ファイル）	FTP アダプタでは，次のユーティリティクラスが設定されています。 • adpftppc.jar※	×
	独自定義ファイル	FTP アダプタでは独自定義ファイルは使用しません。	－

（凡例）

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥custom-adapter¥FTP¥lib」です。

(11) ファイル操作アダプタの場合

ファイル操作アダプタを定義する場合の，サービスアダプタ定義画面での設定項目について，オペレーションごとに説明します。

(a) ファイル変換オペレーション

ファイル変換オペレーションの場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-159 ファイル操作アダプタのサービスアダプタ定義画面（基本）の設定項目（ファイル変換オペレーション）

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「ファイル変換（TRANSFORM オペレーション）」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	標準	[使う] チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_transform_request.xsd※が設定されています。	×
応答電文	標準	[使う] チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_transform_response.xsd※が設定されています。	×
フォルト電文		電文フォーマット	設定しません（フォルト電文は使用しません）。	－

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥schema¥adpfop」です。

ファイル変換オペレーションの場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-160　ファイル操作アダプタのサービスアダプタ定義画面（詳細）の設定項目（ファイル変換オペレーション）

分類	項目	説明		設定
サービスアダプタ制御情報	ユーティリティクラス（JAR ファイル）	adpfoppc.jar※1 が設定されています。		×
	独自定義ファイル	一括処理方式および分割処理方式共通の設定	cscFileOperation.properties（ファイル操作アダプタ定義ファイル）※2 が設定されています。 また，文字コード変換 UOC が独自定義ファイルを参照する場合は，次のファイルを追加します。 • csc_owncodeconvert.properties • csc_owncodeconvert_in.properties（入力ファイル用） • csc_owncodeconvert_out.properties（出力ファイル用）	◎
		一括処理方式を選択した場合	ファイル操作アダプタ定義ファイルの次のキーに指定したファイルを追加します。 • csc.adapter.fileOperation.transform.all.inFormat • csc.adapter.fileOperation.transform.all.outFormat • csc.adapter.fileOperation.transform.all.styleSheet	◎
		分割処理方式を選択した場合	ファイル操作アダプタ定義ファイルの次のキーに指定したファイルを追加します。 • csc.adapter.fileOperation.transform.data.inFormat • csc.adapter.fileOperation.transform.data.outFormat • csc.adapter.fileOperation.transform.data.styleSheet ヘッダレコード部を使用する（csc.adapter.fileOperation.transform.headerRecord=ON）場合は，ファイル操作アダプタ定義ファイルの次のキーに指定したファイルを追加します。 • csc.adapter.fileOperation.transform.header.inFormat • csc.adapter.fileOperation.transform.header.outFormat • csc.adapter.fileOperation.transform.header.styleSheet トレーラレコード部を使用する（csc.adapter.fileOperation.transform.trailerRecord=ON）場合は，ファイル操作アダプタ定義	◎

分類	項目	説明		設定
サービスアダプタ制御情報	独自定義ファイル	分割処理方式を選択した場合	ファイルの次のキーに指定したファイルを追加します。 • csc.adapter.fileOperation.transform.trailer.inFormat • csc.adapter.fileOperation.transform.trailer.outFormat • csc.adapter.fileOperation.transform.trailer.styleSheet 出力ファイルが XML 形式（csc.adapter.fileOperation.transform.output=xml）の場合，csc.adapter.fileOperation.transform.outputTemplateXmlFile キーで指定した出力用テンプレート XML ファイルを追加します。	◎

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥lib」です。

注※2

設定されている cscFileOperation.properties（ファイル操作アダプタ定義ファイル）はテンプレートファイルです。必要に応じて cscFileOperation.properties を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。ファイル操作アダプタ定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.5.2　ファイル操作アダプタ定義ファイル」を参照してください。

(b)　ファイル複製オペレーション

ファイル複製オペレーションの場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-161　ファイル操作アダプタのサービスアダプタ定義画面（基本）の設定項目（ファイル複製オペレーション）

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「ファイル複製（COPY オペレーション）」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	標準	［使う］チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_copy_request.xsd※が設定されています。	×
応答電文	標準	［使う］チェックボックス	チェックしません。	－

分類		項目	説明	設定
応答電文	サービス部品	電文フォーマット	adpfop_copy_response.xsd※が設定されています。	×
フォルト電文		電文フォーマット	設定しません（フォルト電文は使用しません）。	－

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥schema¥adpfop」です。

ファイル複製オペレーションの場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-162　ファイル操作アダプタのサービスアダプタ定義画面（詳細）の設定項目（ファイル複製オペレーション）

分類	項目	説明	設定
サービスアダプタ制御情報	ユーティリティクラス（JAR ファイル）	adpfoppc.jar※1 が設定されています。	×
	独自定義ファイル	cscFileOperation.properties（ファイル操作アダプタ定義ファイル）※2 が設定されています。	○

（凡例）

○：任意で設定します。

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥lib」です。

注※2

設定されている cscFileOperation.properties（ファイル操作アダプタ定義ファイル）はテンプレートファイルです。必要に応じて cscFileOperation.properties を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。ファイル操作アダプタ定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.5.2　ファイル操作アダプタ定義ファイル」を参照してください。

(c)　ファイル移動オペレーション

ファイル移動オペレーションの場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-163　ファイル操作アダプタのサービスアダプタ定義画面（基本）の設定項目（ファイル移動オペレーション）

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「ファイル移動（MOVE オペレーション）」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	標準	[使う] チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_move_request.xsd※が設定されています。	×
応答電文	標準	[使う] チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_move_response.xsd※が設定されています。	×
フォルト電文		電文フォーマット	設定しません（フォルト電文は使用しません）。	－

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥schema¥adpfop」です。

ファイル移動オペレーションの場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-164　ファイル操作アダプタのサービスアダプタ定義画面（詳細）の設定項目（ファイル移動オペレーション）

分類	項目	説明	設定
サービスアダプタ制御情報	ユーティリティクラス（JAR ファイル）	adpfoppc.jar※1 が設定されています。	×
	独自定義ファイル	cscFileOperation.properties（ファイル操作アダプタ定義ファイル）※2 が設定されています。	○

（凡例）

○：任意で設定します。

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥lib」です。

注※2

設定されている cscFileOperation.properties（ファイル操作アダプタ定義ファイル）はテンプレートファイルです。必要に応じて cscFileOperation.properties を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。ファイル操作アダプタ定義ファイルの詳細については，マニュアル「サービスプラットフォーム リファレンス」の「3.5.2　ファイル操作アダプタ定義ファイル」を参照してください。

(d)　ファイル・フォルダ削除オペレーション

ファイル・フォルダ削除オペレーションの場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-165　ファイル操作アダプタのサービスアダプタ定義画面（基本）の設定項目（ファイル・フォルダ削除オペレーション）

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「ファイル削除（DELETE オペレーション）」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	標準	［使う］チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_delete_request.xsd※が設定されています。	×
応答電文	標準	［使う］チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_delete_response.xsd※が設定されています。	×
フォルト電文		電文フォーマット	設定しません（フォルト電文は使用しません）。	－

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥schema¥adpfop」です。

ファイル・フォルダ削除オペレーションの場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-166　ファイル操作アダプタのサービスアダプタ定義画面（詳細）の設定項目（ファイル・フォルダ削除オペレーション）

分類	項目	説明	設定
サービスアダプタ制御情報	ユーティリティクラス（JAR ファイル）	adpfoppc.jar※1 が設定されています。	×
	独自定義ファイル	cscFileOperation.properties（ファイル操作アダプタ定義ファイル）※2 が設定されています。	○

（凡例）

○：任意で設定します。

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥lib」です。

注※2

設定されている cscFileOperation.properties（ファイル操作アダプタ定義ファイル）はテンプレートファイルです。必要に応じて cscFileOperation.properties を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。ファイル操作アダプタ定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.5.2　ファイル操作アダプタ定義ファイル」を参照してください。

(e)　ファイル圧縮オペレーション

ファイル圧縮オペレーションの場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-167　ファイル操作アダプタのサービスアダプタ定義画面（基本）の設定項目（ファイル圧縮オペレーション）

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「ファイル圧縮（COMPRESS オペレーション）」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	標準	［使う］チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_compress_request.xsd※が設定されています。	×
応答電文	標準	［使う］チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_compress_response.xsd※が設定されています。	×
フォルト電文		電文フォーマット	設定しません（フォルト電文は使用しません）。	－

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥schema¥adpfop」です。

ファイル圧縮オペレーションの場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-168　ファイル操作アダプタのサービスアダプタ定義画面（詳細）の設定項目（ファイル圧縮オペレーション）

分類	項目	説明	設定
サービスアダプタ制御情報	ユーティリティクラス（JAR ファイル）	adpfoppc.jar※1 が設定されています。	×
	独自定義ファイル	cscFileOperation.properties（ファイル操作アダプタ定義ファイル）※2 が設定されています。	○

（凡例）

○：任意で設定します。

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥lib」です。

注※2

設定されている cscFileOperation.properties（ファイル操作アダプタ定義ファイル）はテンプレートファイルです。必要に応じて cscFileOperation.properties を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。ファイル操作アダプタ定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.5.2　ファイル操作アダプタ定義ファイル」を参照してください。

(f)　ファイル伸張オペレーション

ファイル伸張オペレーションの場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-169　ファイル操作アダプタのサービスアダプタ定義画面（基本）の設定項目（ファイル伸張オペレーション）

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「ファイル伸張（EXTRACT オペレーション）」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	標準	［使う］チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_extract_request.xsd※が設定されています。	×

分類		項目	説明	設定
応答電文	標準	[使う] チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_extract_response.xsd※が設定されています。	×
フォルト電文		電文フォーマット	設定しません（フォルト電文は使用しません）。	－

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥schema¥adpfop」です。

ファイル伸張オペレーションの場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-170　ファイル操作アダプタのサービスアダプタ定義画面（詳細）の設定項目（ファイル伸張オペレーション）

分類	項目	説明	設定
サービスアダプタ制御情報	ユーティリティクラス（JAR ファイル）	adpfoppc.jar※1 が設定されています。	×
	独自定義ファイル	cscFileOperation.properties（ファイル操作アダプタ定義ファイル）※2 が設定されています。	○

（凡例）

○：任意で設定します。

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥lib」です。

注※2

設定されている cscFileOperation.properties（ファイル操作アダプタ定義ファイル）はテンプレートファイルです。必要に応じて cscFileOperation.properties を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。ファイル操作アダプタ定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.5.2　ファイル操作アダプタ定義ファイル」を参照してください。

(g)　ファイル一覧取得オペレーション

ファイル一覧取得オペレーションの場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-171　ファイル操作アダプタのサービスアダプタ定義画面（基本）の設定項目（ファイル一覧取得オペレーション）

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「ファイル一覧取得（GETLIST オペレーション）」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	標準	［使う］チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_get_list_request.xsd※が設定されています。	×
応答電文	標準	［使う］チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_get_list_response.xsd※が設定されています。	×
フォルト電文		電文フォーマット	設定しません（フォルト電文は使用しません）。	－

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥schema¥adpfop」です。

ファイル一覧取得オペレーションの場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-172　ファイル操作アダプタのサービスアダプタ定義画面（詳細）の設定項目（ファイル一覧取得オペレーション）

分類	項目	説明	設定
サービスアダプタ制御情報	ユーティリティクラス（JAR ファイル）	adpfoppc.jar※1 が設定されています。	×
	独自定義ファイル	cscFileOperation.properties（ファイル操作アダプタ定義ファイル）※2 が設定されています。	○

（凡例）

○：任意で設定します。

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥lib」です。

注※2

設定されている cscFileOperation.properties（ファイル操作アダプタ定義ファイル）はテンプレートファイルです。必要に応じて cscFileOperation.properties を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。ファイル操作アダプタ定義ファイルの詳細については，マニュアル「サービスプラットフォーム リファレンス」の「3.5.2　ファイル操作アダプタ定義ファイル」を参照してください。

(h)　ファイル一覧書き込みオペレーション

ファイル一覧書き込みオペレーションの場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-173　ファイル操作アダプタのサービスアダプタ定義画面（基本）の設定項目（ファイル一覧書き込みオペレーション）

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「ファイル一覧書き込み（WRITELIST オペレーション）」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	標準	［使う］チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_write_list_request.xsd※が設定されています。	×
応答電文	標準	［使う］チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_write_list_response.xsd※が設定されています。	×
フォルト電文		電文フォーマット	設定しません（フォルト電文は使用しません）。	－

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥schema¥adpfop」です。

ファイル一覧書き込みオペレーションの場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-174　ファイル操作アダプタのサービスアダプタ定義画面（詳細）の設定項目（ファイル一覧書き込みオペレーション）

分類	項目	説明	設定
サービスアダプタ制御情報	ユーティリティクラス（JAR ファイル）	adpfoppc.jar※1 が設定されています。	×

分類	項目	説明	設定
サービスアダプタ制御情報	独自定義ファイル	cscFileOperation.properties（ファイル操作アダプタ定義ファイル）※2 が設定されています。	○

（凡例）

○：任意で設定します。

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥lib」です。

注※2

設定されている cscFileOperation.properties（ファイル操作アダプタ定義ファイル）はテンプレートファイルです。必要に応じて cscFileOperation.properties を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。ファイル操作アダプタ定義ファイルの詳細については，マニュアル「サービスプラットフォーム リファレンス」の「3.5.2　ファイル操作アダプタ定義ファイル」を参照してください。

(i)　ファイル読み込みオペレーション

ファイル読み込みオペレーションの場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-175　ファイル操作アダプタのサービスアダプタ定義画面（基本）の設定項目（ファイル読み込みオペレーション）

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「ファイル読み込み（READ オペレーション）」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	標準	［使う］チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	adpfop_read_request.xsd※が設定されています。	×
応答電文	標準	［使う］チェックボックス	チェックしません。	－
	サービス部品	電文フォーマット	入力ファイルに従って，任意のスキーマファイルが設定されています。［any 型を使う］をチェックした場合は，設定されません。	×
フォルト電文		電文フォーマット	設定しません（フォルト電文は使用しません）。	－

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥schema¥adpfop」です。

ファイル読み込みオペレーションの場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-176　ファイル操作アダプタのサービスアダプタ定義画面（詳細）の設定項目（ファイル読み込みオペレーション）

分類	項目	説明	設定
サービスアダプタ制御情報	ユーティリティクラス（JAR ファイル）	adpfoppc.jar※が設定されています。	×
	独自定義ファイル	何も設定しません。	×

（凡例）

×：表示されている内容を確認します。

注※

ファイルの格納先ディレクトリは「<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¥lib」です。

(j)　ファイル書き込みオペレーション

ファイル書き込みオペレーションの場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-177　ファイル操作アダプタのサービスアダプタ定義画面（基本）の設定項目（ファイル書き込みオペレーション）

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「ファイル書き込み（WRITE オペレーション）」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	標準	[使う] チェックボックス	チェックしません。	−
	サービス部品	電文フォーマット	出力ファイルに従って，任意のスキーマファイルが設定されています。[any 型を使う] をチェックした場合は，設定されません。	×
応答電文	標準	[使う] チェックボックス	チェックしません。	−
	サービス部品	電文フォーマット	adpfop_write_response.xsd※が設定されています。	×
フォルト電文		電文フォーマット	設定しません（フォルト電文は使用しません）。	−

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥schema¥adpfop」です。

ファイル書き込みオペレーションの場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-178　ファイル操作アダプタのサービスアダプタ定義画面（詳細）の設定項目（ファイル書き込みオペレーション）

分類	項目	説明	設定
サービスアダプタ制御情報	ユーティリティクラス（JARファイル）	adpfoppc.jar※が設定されています。	×
	独自定義ファイル	何も設定しません。	×

（凡例）

×：表示されている内容を確認します。

注※

ファイルの格納先ディレクトリは「<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¥lib」です。

(12)　メールアダプタの場合

メールアダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-179　メールアダプタのサービスアダプタ定義画面（基本）の設定項目

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「SEND」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	サービス部品	電文フォーマット	adpmail_smtp_request.xsd※が設定されています。	×
応答電文	サービス部品	電文フォーマット	adpmail_smtp_response.xsd※が設定されています。	×
フォルト電文			設定しません（フォルト電文は使用しません）。	－

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥schema¥mail」です。

メールアダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。

表 3-180　メールアダプタのサービスアダプタ定義画面（詳細）の設定項目

分類	項目	説明	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	cscmsg_adpejb.jar が設定されています。	×
	ユーティリティクラス（JAR ファイル）	次のユーティリティクラスが設定されています。 • adpmailpc.jar※	×
	独自定義ファイル	メールアダプタでは独自定義ファイルは使用しません。	−

（凡例）

×：表示されている内容を確認します。

−：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥lib」です。

追加したメールアダプタを編集するには，ツリービューのサービス定義一覧から，該当するメールアダプタを選択し，ダブルクリックします。

(13)　HTTP アダプタの場合

HTTP アダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要または任意の値を設定することを示します。

表 3-181　HTTP アダプタのサービスアダプタ定義画面（基本）の設定項目

分類		項目	説明	設定
サービス部品制御情報		システム例外をフォルトに変換する（チェックボックス）	通信エラーなどのシステム例外をフォルトとして処理する場合はチェックします。	○
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	ボディ	サービス部品電文フォーマット	送信するデータに応じて次の電文フォーマットを指定します。 • ノーマルモードの場合 次のファイルを指定します。 adphttp_body_form-data.xsd adphttp_body_form-data_jwt_assertion1.xsd adphttp_body_form-data_jwt_assertion2.xsd adphttp_body_form-data_password.xsd • パススルーモードの場合	◎

分類		項目	説明	設定
要求電文	ボディ	サービス部品電文フォーマット	「3.3.13(2)(c)　電文フォーマットの作成方法」で作成した任意のスキーマファイルを指定します。 • ファイルデータの場合 adphttp_body_empty.fdx を指定します。	◎
応答電文	ボディ	サービス部品電文フォーマット	受信するデータに応じて次の電文フォーマットを指定します。 • パススルーモードの場合 「3.3.13(2)(c)　電文フォーマットの作成方法」で作成した任意のスキーマファイルを指定します。 • ファイルデータの場合 adphttp_body_empty.fdx を指定します。	◎
フォルト電文		フォルト電文フォーマット	adphttp_fault.xsd※が設定されています。	×

（凡例）

◎：必ず設定します。

○：任意で設定します。

×：表示されている内容を確認します。

注※

ファイルの格納先ディレクトリは「<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¥custom-adapter¥HTTP¥schema」です。

HTTP アダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。

表 3-182　HTTP アダプタのサービスアダプタ定義画面（詳細）の設定項目

分類	項目	説明	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	cscmsg_adpejb.jar が設定されています。	×
	ユーティリティクラス（JAR ファイル）	adphttppc.jar※1 が設定されています。	×
	独自定義ファイル	cscadphttp.properties※2 が設定されています。	○
		必要に応じてカスタムアダプタ定義ファイル（csccustomadapter.properties）※3 を追加します。	○
		JSON-XML 変換をする場合は JSON-XML 変換定義ファイル（csc_json_converter.properties）※4 を追加します。	○

（凡例）

○：必要に応じて設定します。

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥custom-adapter¥HTTP¥lib」です。

注※2

設定されている cscadphttp.properties（HTTP アダプタ定義ファイル）はテンプレートファイルです。必要に応じて cscadphttp.properties を選択してから［編集］ボタンをクリックして内容を修正し，ファイルを保存してください。HTTP アダプタ定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.10.1　HTTP アダプタ定義ファイル」を参照してください。

注※3

カスタムアダプタ定義ファイル（csccustomadapter.properties）の詳細は，「付録 B.2(4)　カスタムアダプタ定義ファイル」を参照してください。

注※4

JSON-XML 変換定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.18.1　JSON-XML 変換定義ファイル」を参照してください。

追加した HTTP アダプタを編集するには，ツリービューのサービス定義一覧から，該当する HTTP アダプタを選択し，ダブルクリックします。

(14)　コマンドアダプタの場合

コマンドアダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要または任意の値を設定することを示します。

表 3-183　コマンドアダプタのサービスアダプタ定義画面（基本）の設定項目

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「EXEC」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	サービス部品	電文フォーマット	adpcmd_request.xsd※が設定されています。	×
応答電文	サービス部品	電文フォーマット	受信するデータに応じて次の電文フォーマット※を指定します。 • adpcmd_response_ms932.fdx • adpcmd_response_utf8.fdx	×
フォルト電文			設定しません（フォルト電文は使用しません）。	－

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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/custom-adapter/Command/schema/」です。

コマンドアダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。

表 3-184　コマンドアダプタのサービスアダプタ定義画面（詳細）の設定項目

分類	項目	説明	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	cscmsg_adpejb.jar が設定されています。	×
	ユーティリティクラス（JAR ファイル）	adpcmdpc.jar※が設定されています。	×
	独自定義ファイル	コマンドアダプタでは独自定義ファイルは使用しません。	－

（凡例）

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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/custom-adapter/Command/lib」です。

追加したコマンドアダプタを編集するには，ツリービューのサービス定義一覧から，該当するコマンドアダプタを選択し，ダブルクリックします。

(15)　SFTP アダプタの場合

SFTP アダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-185　SFTP アダプタのサービスアダプタ定義画面（基本）の設定項目

分類		項目	説明	設定
サービス部品制御情報		オペレーション	次の中から使用するオペレーションを追加します。 PUT：PUT 系処理 GET：GET 系処理 GETINFO：GETINFO 系処理	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	サービス部品	電文フォーマット	［オペレーション］の設定に応じて，次の電文フォーマット定義ファイルが設定されています。 • 定義パターンが「ファイル送信（PUT オペレーション）」の場合 sftpadp_put_request.xsd※ • 定義パターンが「ファイル受信送信（GET オペレーション）」の場合 sftpadp_get_request.xsd※	×

分類		項目	説明	設定
要求電文	サービス部品	電文フォーマット	• 定義パターンが「ファイル情報取得（GETINFO オペレーション）」の場合 sftpadp_getinfo_request.xsd※	×
応答電文	サービス部品	電文フォーマット	［オペレーション］の設定に応じて，次の電文フォーマット定義ファイルが設定されています。 • 定義パターンが「ファイル送信（PUT オペレーション）」の場合 sftpadp_put_response.xsd※ • 定義パターンが「ファイル受信送信（GET オペレーション）」の場合 sftpadp_get_response.xsd※ • 定義パターンが「ファイル情報取得（GETINFO オペレーション）」の場合 sftpadp_getinfo_response.xsd※	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※

ファイルの格納先ディレクトリは「<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¥custom-adapter¥SFTP¥schema」です。

SFTP アダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-186　SFTP アダプタのサービスアダプタ定義画面（詳細）の設定項目

分類	項目	説明	設定
サービスアダプタ制御情報	ユーティリティクラス（JAR ファイル）	SFTP アダプタでは，次のユーティリティクラスが設定されています。 • adpsftppc.jar※	×
	独自定義ファイル	SFTP アダプタでは独自定義ファイルは使用しません。	－

（凡例）

×：表示されている内容を確認します。

－：設定しません。

注※

ファイルの格納先ディレクトリは「<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¥custom-adapter¥SFTP¥lib」です。

(16)　Kafka アダプタの場合

Kafka アダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要または任意の値を設定することを示します。

表 3-187　Kafka アダプタのサービスアダプタ定義画面（基本）の設定項目

分類		項目	説明	設定
サービス部品制御情報		オペレーション	「SEND」を追加します。	◎
オペレーション情報		通信モデル	「同期」を設定します。	◎
要求電文	サービス部品	電文フォーマット	ユーザが必要に応じて作成した，ボディ要求電文の電文フォーマット定義ファイルを指定します。	◎
応答電文	サービス部品	電文フォーマット	電文フォーマット定義ファイル（adpkafka_send_response.xsd※）が設定されています。	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※

ファイルの格納先ディレクトリは「<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¥custom-adapter¥Kafka¥schema」です。

Kafka アダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-188　Kafka アダプタのサービスアダプタ定義画面（詳細）の設定項目

分類	項目	説明	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	cscmsg_adpejb.jar が設定されています。	×
	ユーティリティクラス（JAR ファイル）	Kafka アダプタでは，次のユーティリティクラスが設定されています。 adpkafkapc.jar※1	×
	独自定義ファイル	Kafka アダプタ定義ファイル（adpkafka.properties）が設定されています。 JSON-XML 変換を行う場合は，必要に応じて JSON-XML 変換定義ファイル（csc_json_converter.properties※2）を指定します。	×

（凡例）

×：表示されている内容を確認します。

注※1

ファイルの格納先ディレクトリは「<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¥custom-adapter¥Kafka¥lib」です。

注※2

JSON-XML 変換定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.18.1　JSON-XML 変換定義ファイル」を参照してください。

(17) gRPC アダプタの場合

gRPC アダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要または任意の値を設定することを示します。

表 3-189 gRPC アダプタのサービスアダプタ定義画面（基本）の設定項目

分類		項目	説明	設定
オペレーション情報		通信モデル	「同期」が設定されています。	×
要求電文	ボディ	電文フォーマット	XML スキーマファイルが設定されています。	×
応答電文	ボディ	電文フォーマット	XML スキーマファイルが設定されています。	×
フォルト電文		フォルト名	サービス部品のフォルト名が表示されます。	×
		電文フォーマット	フォルト電文フォーマット定義ファイル（adpgrpc_fault.xsd※）が設定されています。	◎

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※

ファイルの格納先ディレクトリは「<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¥custom-adapter¥gRPC¥schema」です。

gRPC アダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 3-190 gRPC アダプタのサービスアダプタ定義画面（詳細）の設定項目

分類	項目	説明	設定
サービスアダプタ制御情報	独自定義ファイル	gRPC アダプタ独自定義ファイル（cscadpgrpc.properties）が設定されています。	◎
	proto ファイル※	サービスアダプタ定義追加（gRPC アダプタ）ウィザードで設定した proto ファイルが設定されています。	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※

proto ファイルの詳細については，マニュアル「サービスプラットフォーム 解説」の「2.13.1(3)(b) proto ファイルの出力」を参照してください。

(18) 汎用カスタムアダプタの場合

汎用カスタムアダプタを定義する場合の，サービスアダプタ定義画面での設定を次の表に示します。

表 3-191　汎用カスタムアダプタのサービスアダプタ定義画面での設定

分類		項目	設定	
			同期	非同期
基本画面	サービス部品制御情報	サービス名	◎	◎
		サービス ID	◎	◎
		サービス種別	×	×
		アドレス	×	×
		最大インスタンス数※1	◎	◎
		サービスクラス名	×	×
		オペレーション	◎	◎
		システム例外をフォルトに変換する（チェックボックス）	○	○
	オペレーション情報	オペレーション名	◎	◎
		通信モデル	◎※2	◎※3
	要求電文	any 型を使う（チェックボックス）	○	○
		使う（チェックボックス）	○	○
		標準電文フォーマット	△※4	△※4
		サービス部品電文フォーマット	◎	◎
		データ変換定義	△※4	△※4
	応答電文	any 型を使う（チェックボックス）	○	－
		使う（チェックボックス）	○	－
		標準電文フォーマット	△※5	－
		サービス部品電文フォーマット	◎	－
		データ変換定義	△※5	－
	フォルト電文	フォルト名	－	－
		サービス部品電文フォーマット	－	－
詳細画面	サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	×	×
		ユーティリティクラス（JAR ファイル）	◎	◎
		独自定義ファイル	○	○

（凡例）

◎：必ず設定します。

○：任意で設定します。

△：場合によっては必ず設定します。

×：表示されている内容を確認します。

－：該当しません。

注※1

アプリケーション統合属性ファイルの要素を独自定義ファイルとして追加した場合，サービスアダプタ定義画面で設定した値は適用されません。最大インスタンス数は，アプリケーション統合属性ファイルの要素「hitachi-application-all-property/ejb-jar/hitachi-session-bean-property/session-runtime/stateless/pooled-instance/maximum」で指定した値が設定されます。

注※2

「同期」を設定します。

注※3

「非同期」を設定します。

注※4

要求電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

注※5

応答電文の［使う］（チェックボックス）にチェックした場合，必ず設定します。

3.4　サービスアダプタの保存

サービスアダプタ定義画面で編集中のサービスアダプタの内容を，編集途中や編集終了時に保存できます。サービスアダプタの内容はリポジトリに保存されます。

データ変換定義で生成したデータ変換定義ファイルは，サービスアダプタ定義画面と同時に保存されます。

サービスアダプタを保存する手順を次に示します。

方法 1

Eclipse のメニューから，［ファイル］－［保存］を選択します。

方法 2

Eclipse のメニューから，［ファイル］－［すべて保管］を選択します。

方法 3

サービスアダプタ定義画面で，［Ctrl］キー＋［S］キーを押します。

なお，サービスアダプタを定義する画面を閉じてサービスアダプタの定義を終了する場合に，サービスアダプタが保存されていないときは，リソース保管ダイアログが表示されます。リソース保管ダイアログでは，保存されていないサービスアダプタの定義を保存できます。

注意事項

不正なデータが入力されている場合，サービスアダプタを保存できないことがあります。その場合，表示されているメッセージに従って対処してください。

3.5　サービスアダプタの編集

保存したサービスアダプタの内容を変更できます。サービスアダプタの内容を変更するには，サービスアダプタ定義画面を表示して，内容を編集します。サービスアダプタ定義画面は，ツリービューのサービス定義一覧から，該当するサービスアダプタを選択し，ダブルクリックすると表示されます。

サービスアダプタの内容の定義方法については，「3.3　サービスアダプタの内容の定義」を参照してください。

また，電文フォーマットを変更した場合は，サービスアダプタの内容を編集してデータ変換を定義し直す必要があります。電文フォーマットを変更した場合の手順については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.3.2　電文フォーマットを変更した場合の定義手順」を参照してください。

3.6　サービスアダプタの検証

作成したサービスアダプタの内容が妥当かどうか検証できます。

必要なサービスアダプタ定義がない場合，または定義の関係が正しくない場合は正常に動作しません。そのため，実行環境で実行する前に，作成したサービスアダプタに必要な項目がそろっているか，それらの関係が正しいかどうかを検証できます。検証は，必要に応じて，任意のタイミングで検証できます。

3.6.1　検証方法

検証する手順を次に示します。

1. **ツリービューのサービス定義一覧で，右クリックします。**
   サービス一覧のポップアップメニューが表示されます。

2. **ポップアップメニューから「検証」を選択します。**
   検証結果がコンソールビューに表示されます。

また，パッケージングを実施した場合も，自動的に検証されます。

なお，検証する前にサービスアダプタが保存されていないときは，リソース保管ダイアログが表示されます。リソース保管ダイアログでは，保存されていないサービスアダプタの定義を保存できます。

3.6.2　検証内容の表示

検証結果として，メッセージがコンソールビューに表示されます。メッセージに従って，修正します。

表示されるメッセージの種類を次の表に示します。

表 3-192　メッセージの種類

種類	説明
エラー	次のどちらかの場合に表示されます。 • 定義の内容に誤りがある。 • 正しい構文で定義しているが，実行できない定義をしている。
警告	実行時にエラーになるおそれがある定義をしている場合に表示されます。
情報	追加情報が表示されます。

3.7　サービスアダプタの削除

作成したサービスアダプタを削除できます。

削除する手順を次に示します。次に示す 2 つの方法があります。

方法 1

1. ツリービューのサービス定義一覧で，サービスを選択します。
2. [Delete] キーを押します。
   削除を確認するダイアログボックスが表示されます。
3. [はい] ボタンをクリックします。
   指定したサービスアダプタが削除されます。

方法 2

1. ツリービューのサービス定義一覧で，サービスを選択して右クリックします。
   サービス一覧のポップアップメニューが表示されます。
2. ポップアップメニューから「削除」を選択します。
   削除を確認するダイアログボックスが表示されます。
3. [はい] ボタンをクリックします。
   指定したサービスアダプタが削除されます。

付録

付録 A　カスタム受付

ここでは，サービスリクエスタの仕様に合わせた受付処理を作成できるカスタム受付について説明します。

付録 A.1　カスタム受付の概要

サービスプラットフォームでは，ユーザが任意の受付を作成できるようにカスタム受付フレームワークを提供しています。このカスタム受付フレームワーク上でユーザが作成した J2EE アプリケーションを動作させることで，任意のプロトコルに対応した受付を開発できます。

(1)　カスタム受付の構成

カスタム受付は，開発者が作成する受付処理と，カスタム受付フレームワークから構成される J2EE アプリケーションです。

開発者が作成する受付処理の内容と，カスタム受付フレームワークで実行される処理の内容を次に示します。

開発者が作成する受付処理の内容

- サービスリクエスタからのサービス部品呼び出し要求を受け付ける処理
- カスタム受付フレームワークに要求電文を渡す処理
- カスタム受付フレームワークから返ってきた応答電文を，サービスリクエスタに返す処理

カスタム受付フレームワークで実行される処理

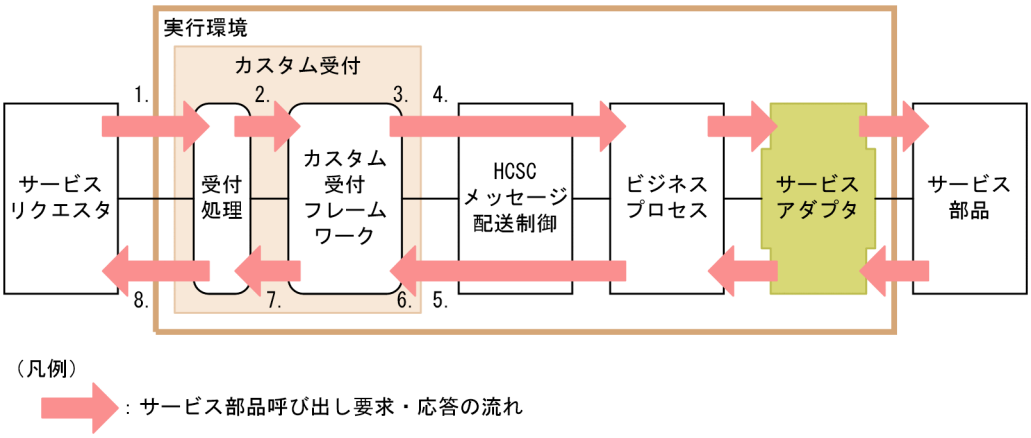
- 受付処理から渡された要求電文を，HCSC メッセージ配送制御に渡す処理
- HCSC メッセージ配送制御から返された応答電文を，受付処理に返す処理
- HCSC データ変換定義に従って，要求電文または応答電文のデータ変換を行う処理
- HTTP 受付ルーティング定義ファイルに設定された情報を基に，呼び出すサービスを選択する処理

(2)　カスタム受付フレームワークの動作の流れ

カスタム受付フレームワークは，カスタム受付を開発および動作させるためのフレームワークです。カスタム受付フレームワークでは，カスタム受付を作成するための API を提供しています。API の詳細は，「付録 A.3　カスタム受付フレームワークの API」を参照してください。

カスタム受付フレームワークは，開発者が作成したカスタム受付を，サービスプラットフォームの実行環境で受付として動作させるための仕組みを提供しています。カスタム受付がサービス部品呼び出し要求を受けるときの，カスタム受付フレームワークの動作の流れを次の図に示します。

図 A-1　カスタム受付フレームワークの動作の流れ



1. 受付処理は，サービスリクエスタからサービス部品呼び出し要求を受け付けます。
2. 受付処理は，サービスリクエスタから受け取った要求電文を，カスタム受付フレームワークに渡します。
3. カスタム受付フレームワークは，要求電文のデータ変換をします。
4. カスタム受付フレームワークは，HCSC メッセージ配送制御を呼び出します。
5. カスタム受付フレームワークは，HCSC メッセージ配送制御から，応答電文を受け取ります。
6. カスタム受付フレームワークは，応答電文のデータ変換をします。
7. 受付処理は，カスタム受付フレームワークから応答電文を受け取ります。
8. 受付処理は，サービスリクエスタに応答電文を返します。

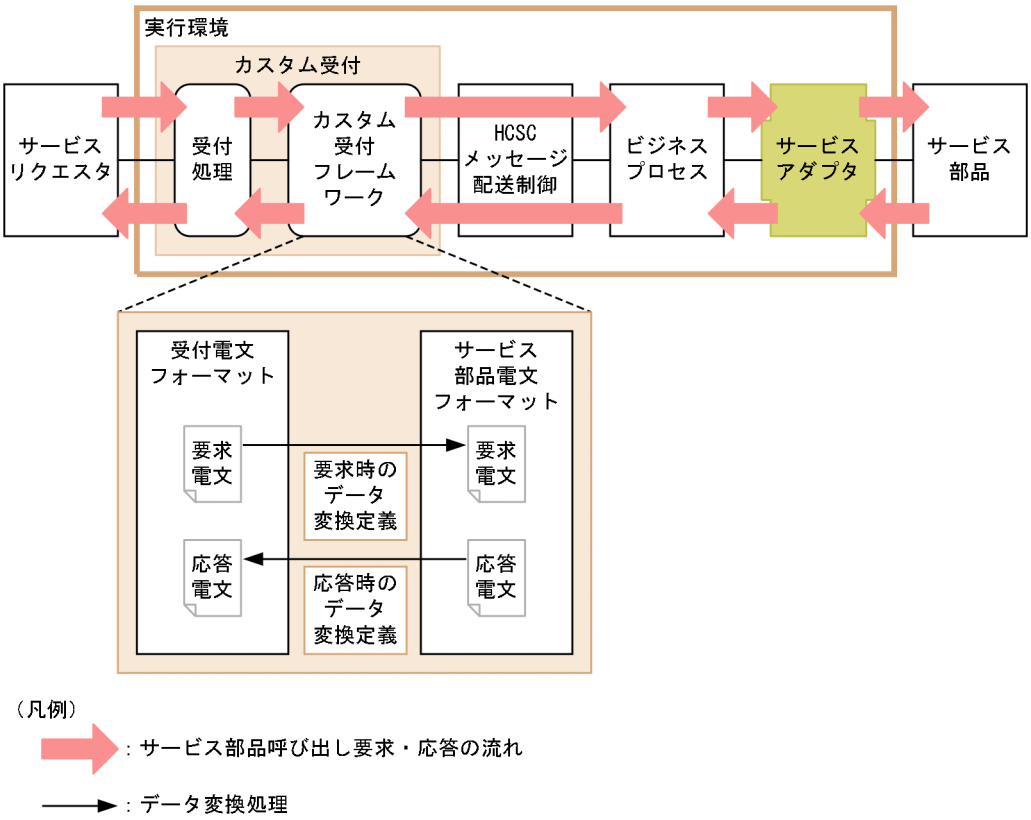
(3)　カスタム受付のデータ変換

カスタム受付では，カスタム受付フレームワークで要求電文および応答電文のデータ変換をします。ただし，通信モデルが非同期の場合は，応答電文のデータ変換をしません。この場合，受付処理に null を返します。

受付処理から受け取った要求電文は，HCSC メッセージ配送制御を呼び出す前に，カスタム受付フレームワークによってデータ変換されます。ビジネスプロセスには，データ変換によって変換された要求電文が，HCSC メッセージ配送制御を経由して渡されます。ビジネスプロセスから返ってきた応答電文は，受付処理に返る前に，カスタム受付フレームワークによってデータ変換されます。サービスリクエスタには，データ変換によって変換された応答電文が，受付処理を経由して返されます。

カスタム受付でのデータの流れとデータ変換の関係を次に示します。

図 A-2　カスタム受付でのデータの流れとデータ変換の関係



要求電文および応答電文のデータ変換時の形式の組み合わせを次に示します。

(a)　要求電文のデータ変換時の形式の組み合わせ

ビジネスプロセスを呼び出したときに，カスタム受付フレームワークが実行できる要求電文のデータ変換の形式を次の表に示します。

表 A-1　カスタム受付フレームワークが実行できる要求電文のデータ変換の形式

要求電文の形式		データ変換の可否
受付電文フォーマット	サービス部品電文フォーマット	
バイナリ	－（指定なし）	○
	バイナリ	○
	XML	○
XML	－（指定なし）	○
	バイナリ	○
	XML	○

（凡例）

○：変換できます。

(b)　応答電文のデータ変換時の形式の組み合わせ

ビジネスプロセスのオペレーションを呼び出したときに，カスタム受付フレームワークが実行できる応答電文のデータ変換の形式を次の表に示します。

表 A-2　カスタム受付フレームワークが実行できる応答電文のデータ変換の形式

応答電文の形式		データ変換の可否
サービス部品電文フォーマット	受付電文フォーマット	
バイナリ	－（指定なし）	○
	バイナリ	○
	XML	○
XML	－（指定なし）	○
	バイナリ	○
	XML	○

（凡例）

○：変換できます。

付録 A.2　カスタム受付の開発

ここでは，カスタム受付の開発について説明します。

カスタム受付は，開発環境で開発します。

(1)　開発するファイルの構成

開発環境で作成するカスタム受付 EAR ファイルの構成を次に示します。

```
├ META-INF
│ ├ application.xml          … 1.
│ ├ cosminexus.xml           … 2.
│ └ MANIFEST.MF              … 3.
└ 受付処理ファイル             … 4.
```

1. EAR ファイルのデプロイメント記述子
2. EAR ファイルのアプリケーション属性ファイル
3. EAR ファイルのマニフェストファイル
4. 作成した受付処理のアーカイブファイル

なお，カスタム受付 EAR ファイルの構成は変更しないでください。受付処理の動作に必要な定義情報などを使用したい場合は，カスタム受付ユーザファイルを作成してください。カスタム受付ユーザファイルについては，「付録 A.2(3)(d)　カスタム受付ユーザファイル」を参照してください。

(2)　受付処理ファイルを作成する

開発者は，受付処理の実装ファイルおよび必要な定義ファイルをアーカイブして，受付処理ファイルを作成します。

注意事項

受付処理の実行時に確保したリソースは，処理が終了しても保持し続けるため，システム全体に高い負荷が掛かった場合に，Java ヒープ不足や Metaspace 領域不足などが原因で OutOfMemoryError が発生することがあります。そのため，OutOfMemoryError が発生したときに適切にリソースを解放する処理や，ロールバックなどのエラー処理を実装しておく必要があります。

次に，受付処理ファイルの詳細を説明します。

(a)　受付処理ファイルの形式

受付処理ファイルの形式には，WAR ファイル形式または EJB-JAR ファイル形式を選択できます。受付処理がサポートするプロトコルに応じて，どちらかを選択してください。

(b)　受付処理の実装に使用するライブラリ

受付処理の実装には，カスタム受付フレームワークが提供する API のライブラリ JAR ファイルを使用します。受付処理の実装時には，次に示す JAR ファイルをクラスパスに指定する必要があります。

```
<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>¥CSC¥lib¥cscmsg.jar
<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¥lib¥cscmsg_urecp_custom.jar
```

(c)　受付処理の実装内容

受付処理で実装する内容は，次の 1.～3.です。

1. サービスリクエスタからのサービス部品呼び出し要求を受け付ける処理
2. カスタム受付フレームワークに要求電文を渡す処理
3. カスタム受付フレームワークから返ってきた応答電文を，サービスリクエスタに返す処理

なお，実装時に利用するカスタム受付フレームワークの API の詳細は，「付録 A.3　カスタム受付フレームワークの API」を参照してください。

(d)　デプロイメント記述子の設定

受付処理ファイルのデプロイメント記述子を設定します。

設定する内容は任意です。

(e)　受付処理ファイルの作成

受付処理の実装ファイルおよび必要な定義ファイルをアーカイブした受付処理ファイルを作成します。

作成手順は，受付処理ファイルの形式に依存します。

なお，作成する受付処理ファイルは，「csc」で始まる名称にはしないでください。「csc」で始まる名称にした場合，動作は保証されません。

(3)　定義ファイルを作成する

カスタム受付で設定する定義ファイルは次の４つです。

- デプロイメント記述子（application.xml）
- アプリケーション属性ファイル（cosminexus.xml）
- マニフェストファイル（MANIFEST.MF）
- カスタム受付ユーザファイル

次に，それぞれのファイルについて説明します。

(a)　デプロイメント記述子（application.xml）

カスタム受付 EAR ファイルのデプロイメント記述子を設定します。

デプロイメント記述子の設定は，サービスプラットフォームが提供するテンプレートを基に受付処理の設定を追加します。テンプレートの格納先を次に示します。

```
<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>¥CSC¥samples¥urecp¥application.xml
```

デプロイメント記述子のテンプレートファイルに記載されている要素および値の編集可否を次の表に示します。なお，デプロイメント記述子の DTD は，「application_1_3.dtd」です。

表 A-3　デプロイメント記述子のテンプレートファイルに記載されている要素

要素	編集可否
<?xml version="1.0" encoding="UTF-8"?>	×
<!DOCTYPE application PUBLIC "-//Sun Microsystems, Inc.//DTD J2EE Application 1.3//EN" "http://java.sun.com/dtd/application_1_3.dtd">	×
<!-- All Rights Reserved. Copyright (C) 2010, Hitachi, Ltd. -->	×

要素	編集可否
<application>	×
<display-name>CSCMsgCustomServiceDelivery</display-name>	×
<description></description>	○
<module>	○
<web>	○
<web-uri>web-uri</web-uri>	○
<context-root>context-root</context-root>	○
</web>	○
</module>	○
<module>	○
<ejb>ejb</ejb>	○
</module>	○
<module>	○
<java>java</java>	○
</module>	○
</application>	×

(凡例)

○：編集できます。

×：編集できません。

(b) アプリケーション属性ファイル（cosminexus.xml）

カスタム受付 EAR ファイルのアプリケーション属性ファイルを設定します。

アプリケーション属性ファイルの設定は，サービスプラットフォームが提供するテンプレートを基に受付処理の設定を追加します。テンプレートの格納先を次に示します。

<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>¥CSC¥samples¥urecp¥cosminexus.xml

アプリケーション属性ファイルのテンプレートファイルに記載されている要素および値の編集可否を次の表に示します。なお，アプリケーション属性ファイルの DTD は「cosminexus_8_0.dtd」です。

表 A-4 アプリケーション属性ファイルのテンプレートファイルに記載されている要素

要素	編集可否
<?xml version="1.0" encoding="UTF-8"?>	×

要素	編集可否
<!DOCTYPE cosminexus-app PUBLIC "-//Hitachi, Ltd.//DTD Cosminexus 8.0//EN" "file:///C:/Program%20Files/Hitachi/Cosminexus/CC/admin/dtds/cosminexus_8_0.dtd">	×
<!-- All Rights Reserved. Copyright (C) 2010, Hitachi, Ltd. -->	×
<cosminexus-app>	×
<start-order>180</start-order>	×
<ejb-jar>	×
<module-name>cscmsg_urecp_custom.jar</module-name>	×
<session>	×
<ejb-name>CSCMsgCustomFwInitializerEJB</ejb-name>	×
<maximum-sessions>0</maximum-sessions>	×
<stateless>	×
<pooled-instance>	×
<minimum>1</minimum>	×
<maximum>1</maximum>	×
</pooled-instance>	×
</stateless>	×
<start-order>10</start-order>	×
</session>	×
<session>	×
<ejb-name>CSCMsgCustomOperationCallerEJB</ejb-name>	×
<maximum-sessions>0</maximum-sessions>	×
<stateless>	×
<pooled-instance>	×
<minimum>0</minimum>	×
<maximum>0</maximum>	×
</pooled-instance>	×
</stateless>	×
<start-order>20</start-order>	×
<ejb-transaction-timeout>	×
<method>	×
<method-name>*</method-name>	×

要素	編集可否
</method>	×
<transaction-timeout>0</transaction-timeout>	×
</ejb-transaction-timeout>	×
</session>	×
</ejb-jar>	×
</cosminexus-app>	×

（凡例）

×：編集できません。

注 1

「csc」で始まる名称のモジュールの設定を追加（<module-name>要素に「csc」で始まる値を指定）しないでください。

注 2

メソッドタイムアウト時間（<method-observation-timeout>要素，および<ejb-method-observation-timeout>要素）を設定しないでください。

注 3

受付処理の J2EE アプリケーション開始時および停止時の順序（<start-order>要素）には，50 より大きい値を設定してください。

(c)　マニフェストファイル（MANIFEST.MF）

カスタム受付 EAR ファイルのマニフェストファイル（MANIFEST.MF）を記載します。

なお，マニフェストファイルを作成するかどうかは任意です。

(d)　カスタム受付ユーザファイル

受付処理に関する設定を外部ファイル化したい場合などに，カスタム受付ユーザファイルを作成します。カスタム受付ユーザファイルは，カスタム受付フレームワークの API を使用して，受付処理の中で読み込むことができます。カスタム受付ユーザファイルを読み込むための API の詳細は，「付録 A.3　カスタム受付フレームワークの API」を参照してください。

カスタム受付ユーザファイルは，開発環境でカスタム受付を定義するときに，独自定義ファイルとして登録します。カスタム受付の定義については，「付録 A.4　カスタム受付の定義」を参照してください。

(4)　カスタム受付 EAR ファイルを作成する

ant や jar コマンドなどを使用して，受付処理ファイルおよび定義ファイルをアーカイブしたカスタム受付 EAR ファイルを作成します。

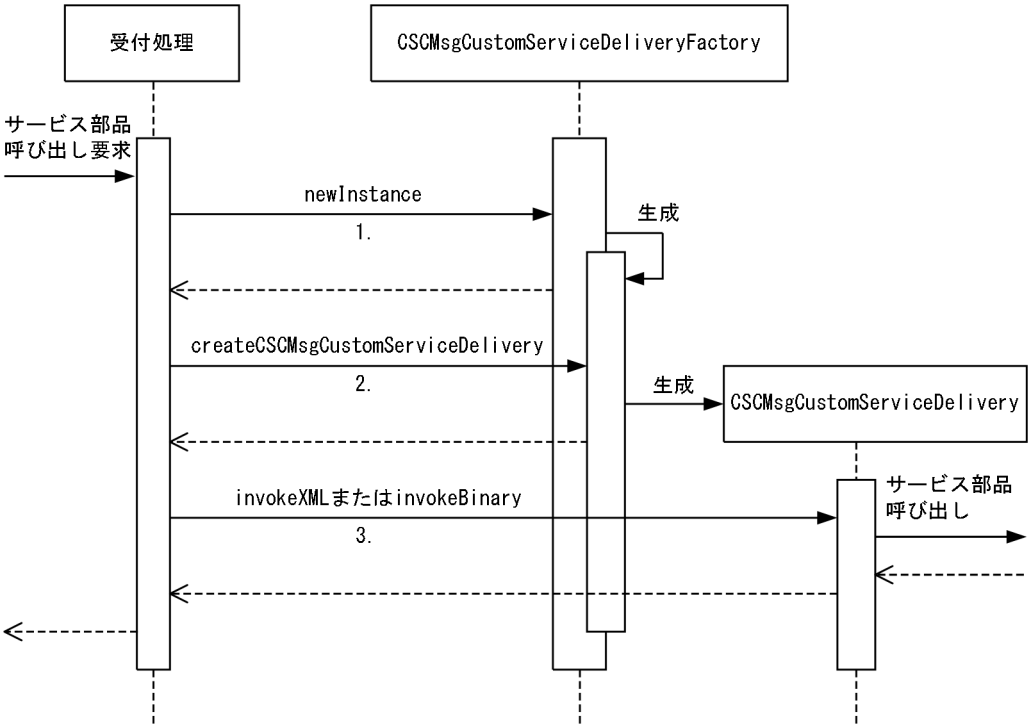
付録 A.3　カスタム受付フレームワークの API

ここでは，カスタム受付フレームワークが提供する API について説明します。

(1)　カスタム受付フレームワークの API の呼び出し順序

受付処理が，カスタム受付フレームワークの API を呼び出す順序を次の図に示します。

図 A-3　カスタム受付フレームワークの API の呼び出し順序



1. CSCMsgCustomServiceDeliveryFactory クラスの newInstance メソッドを呼び出して，CSCMsgCustomServiceDeliveryFactory オブジェクトを取得します。
2. 1.で取得した CSCMsgCustomServiceDeliveryFactory オブジェクトの createCSCMsgCustomServiceDelivery メソッドを呼び出して，CSCMsgCustomServiceDelivery オブジェクトを取得します。
3. 2.で取得した CSCMsgCustomServiceDelivery オブジェクトの invokeXML メソッドまたは invokeBinary メソッドを呼び出して，カスタム受付フレームワークに要求電文を渡します。

(2)　CSCMsgCustomServiceDeliveryFactory クラス

(a)　説明

CSCMsgCustomServiceDeliveryFactory クラスは，CSCMsgCustomServiceDeliveryFactory オブジェクトの生成，CSCMsgCustomServiceDelivery オブジェクトの生成などの機能を提供するファクトリクラスです。

(b) 形式

```
package jp.co.Hitachi.soft.csc.msg.message.reception.custom;
public abstract class CSCMsgCustomServiceDeliveryFactory {
  public static CSCMsgCustomServiceDeliveryFactory newInstance();
  public CSCMsgCustomServiceDelivery createCSCMsgCustomServiceDelivery();
}
```

(c) メソッド

CSCMsgCustomServiceDeliveryFactory クラスのメソッドを次の表に示します。

メソッド名	機能
newInstance メソッド	CSCMsgCustomServiceDeliveryFactory オブジェクトを生成します。
createCSCMsgCustomServiceDelivery メソッド	CSCMsgCustomServiceDelivery オブジェクトを生成します。

newInstance メソッド

説明

CSCMsgCustomServiceDeliveryFactory オブジェクトを生成します。

形式

```
public static CSCMsgCustomServiceDeliveryFactory newInstance();
```

パラメタ

なし。

例外

なし。

戻り値

- CSCMsgCustomServiceDeliveryFactory

  CSCMsgCustomServiceDeliveryFactory オブジェクトを返却します。

createCSCMsgCustomServiceDelivery メソッド

説明

CSCMsgCustomServiceDelivery オブジェクトを生成します。

形式

```
public CSCMsgCustomServiceDelivery createCSCMsgCustomServiceDelivery();
```

パラメタ

なし。

例外

なし。

戻り値

- CSCMsgCustomServiceDelivery

  CSCMsgCustomServiceDelivery オブジェクトを返却します。

(3) CSCMsgCustomServiceDelivery インターフェース

(a) 説明

カスタム受付の受付処理が，カスタム受付フレームワークと電文の受け渡しをするためのメソッドを提供します。また，開発環境で登録したカスタム受付ユーザファイルを取得するメソッドを提供します。

(b) 形式

```
package jp.co.Hitachi.soft.csc.msg.message.reception.custom;
public interface CSCMsgCustomServiceDelivery {
  public String invokeXML(String cscCorrelationID,
      String cscServiceOperationName, String msg)
      throws CSCMsgServerException;
  public byte[] invokeXML(String cscCorrelationID,
      String cscServiceOperationName, byte[] msg)
      throws CSCMsgServerException;
  public String invokeXML(String cscCorrelationID, String msg)
      throws CSCMsgServerException;
  public byte[] invokeXML(String cscCorrelationID, byte[] msg)
      throws CSCMsgServerException;

  public byte[] invokeBinary(String cscCorrelationID,
      String cscServiceOperationName,
      int requestMessageLength, byte[] msg)
      throws CSCMsgServerException;
  public byte[] invokeBinary(String cscCorrelationID,
      int requestMessageLength, byte[] msg)
      throws CSCMsgServerException;
  public InputStream getUserFile(String fileName);
  public InputStream getUserFileForExEnv(String fileName)
      throws CSCMsgServerException;
}
```

(c) メソッド

CSCMsgCustomServiceDelivery インターフェースのメソッドを次の表に示します。

メソッド名	機能
invokeXML メソッド（形式 1）	指定されたオペレーション名のサービスに対して XML 電文のメッセージを送信し，その応答を返します。

メソッド名	機能
invokeXML メソッド（形式 2）	指定されたオペレーション名のサービスに対してバイト配列型の XML 電文のメッセージを送信し，その応答を返します。応答も，バイト配列型になります。
invokeXML メソッド（形式 3）※1	デフォルトのオペレーション名のサービスに対して XML 電文のメッセージを送信し，その応答を返します。
invokeXML メソッド（形式 4）※1	デフォルトのオペレーション名のサービスに対してバイト配列型の XML 電文のメッセージを送信し，その応答を返します。応答も，バイト配列型になります。
invokeBinary メソッド（形式 1）	指定されたオペレーション名のサービスに対してバイナリ電文のメッセージを送信し，その応答を返します。
invokeBinary メソッド（形式 2）※2	デフォルトのオペレーション名のサービスに対してバイナリ電文のメッセージを送信し，その応答を返します。
getUserFile メソッド	カスタム受付ユーザファイルの InputStream を返します（非推奨メソッド）。
getUserFileForExEnv メソッド	カスタム受付ユーザファイルの InputStream を返します。

注※1

invokeXML メソッド（形式 1）のオペレーション名を省略した場合と同等のメソッドです。

注※2

invokeBinary メソッド（形式 1）のオペレーション名を省略した場合と同等のメソッドです。

invokeXML メソッド（形式 1）

説明

指定されたオペレーション名のサービスに対して XML 電文のメッセージを送信し，その応答を返します。

形式

```
public String invokeXML(String cscCorrelationID,
    String cscServiceOperationName, String msg)
    throws CSCMsgServerException;
```

パラメタ

- cscCorrelationID（クライアント相関 ID）

  クライアント相関 ID は，サービスリクエスタからの要求電文を一意に識別するための相関識別子です。

  クライアント相関 ID が次の制限に従っていない場合，CSCMsgServerException 例外をスローします。

  ・サイズ：255 文字以内

  ・使用できる文字：半角の英数字，アンダーバー（_），ピリオド（.）およびハイフン（-）

クライアント相関 ID は，サービスリクエスタからの要求電文と，HCSC サーバで管理しているログおよびトレースを対応づけるために使用します。したがって，HCSC サーバに送信した要求電文ごとに異なる ID を指定する必要があります。

クライアント相関 ID を省略する場合，null または空文字列（""）を指定します。

- cscServiceOperationName（オペレーション名）

  オペレーション名は，要求先のサービス名に対するオペレーション名です。

  オペレーション名は，開発環境で定義したサービス部品のオペレーション名を指定します。

  オペレーション名が次の制限に従っていない場合，CSCMsgServerException 例外をスローします。

  ・サイズ：255 バイト以内

  ・使用できる文字：XMLSchema の NCName 定義文字

  ユーザ定義受付定義画面のカスタム定義で定義していないオペレーション名を指定した場合，CSCMsgServerException 例外をスローします。

  オペレーション名を省略する（デフォルトのオペレーション名を使用する）場合，null または空文字列（""）を指定します。

- msg（ユーザ電文）

  ユーザ電文は，サービスリクエスタからの要求電文（XML 形式）です。

  要求電文がない場合，null または空文字列（""）を指定します。

  null または空文字（""）を指定してデータ変換などで変更しない場合，ビジネスプロセスに空の電文を渡します。

例外

- CSCMsgServerException

  次の場合にスローします。

  ・メソッドのパラメタに不正な値が指定された場合

  ・要求電文および応答電文のデータ変換でエラーが発生した場合

  ・要求先のサービス部品でエラーが発生した場合

戻り値

- String

  ・通信モデルが同期の場合

  サービス部品の実行要求に対する応答電文（XML 形式）を返却します。

  応答電文がない場合，null を返却します。

  ・通信モデルが非同期の場合

  null を返却します。

invokeXML メソッド（形式 2）

説明

指定されたオペレーション名のサービスに対して，バイト配列型の XML 電文のメッセージを送信し，その応答を返します。応答も，バイト配列型になります。

ユーザ電文をバイト配列型で扱うカスタム受付の場合，このメソッドを使用することで，String 型への変換処理を省くことができます。

形式

```
public byte[] invokeXML(String cscCorrelationID,
    String cscServiceOperationName, byte[] msg)
    throws CSCMsgServerException;
```

パラメタ

- cscCorrelationID（クライアント相関 ID）

  クライアント相関 ID は，サービスリクエスタからの要求電文を一意に識別するための相関識別子です。

  クライアント相関 ID が次の制限に従っていない場合，CSCMsgServerException 例外をスローします。

  ・サイズ：255 文字以内

  ・使用できる文字：半角の英数字，アンダーバー（_），ピリオド（.）およびハイフン（-）

  クライアント相関 ID は，サービスリクエスタからの要求電文と，HCSC サーバで管理しているログおよびトレースを対応づけるために使用します。したがって，HCSC サーバに送信した要求電文ごとに異なる ID を指定する必要があります。

  クライアント相関 ID を省略する場合，null または空文字列（""）を指定します。

- cscServiceOperationName（オペレーション名）

  オペレーション名は，要求先のサービス名に対するオペレーション名です。

  オペレーション名は，開発環境で定義したサービス部品のオペレーション名を指定します。

  オペレーション名が次の制限に従っていない場合，CSCMsgServerException 例外をスローします。

  ・サイズ：255 バイト以内

  ・使用できる文字：XMLSchema の NCName 定義文字

  カスタム受付定義画面で定義していないオペレーション名を指定した場合，CSCMsgServerException 例外をスローします。

  オペレーション名を省略する（デフォルトのオペレーション名を使用する）場合，null または空文字列（""）を指定します。

- msg（ユーザ電文）

  ユーザ電文は，サービスリクエスタからの要求電文（XML 形式）です。

  要求電文がない場合，null を指定します。

null または空文字（""）を指定してデータ変換などで変更しない場合，ビジネスプロセスに空の電文を渡します。

例外

- CSCMsgServerException

  次の場合にスローします。

  ・メソッドのパラメタに不正な値が指定された場合

  ・要求電文および応答電文のデータ変換でエラーが発生した場合

  ・要求先のサービス部品でエラーが発生した場合

戻り値

- byte[]

  ・通信モデルが同期の場合

  サービス部品の実行要求に対する応答電文（XML 形式）を返却します。

  応答電文がない場合，null を返却します。

  ・通信モデルが非同期の場合

  null を返却します。

invokeXML メソッド（形式 3）

説明

デフォルトのオペレーション名のサービスに対して XML 電文のメッセージを送信し，その応答を返します。

形式

```
public String invokeXML(String cscCorrelationID, String msg)
    throws CSCMsgServerException;
```

パラメタ

- cscCorrelationID（クライアント相関 ID）

  クライアント相関 ID は，サービスリクエスタからの要求電文を一意に識別するための相関識別子です。

  クライアント相関 ID が次の制限に従っていない場合，CSCMsgServerException 例外をスローします。

  ・サイズ：255 文字以内

  ・使用できる文字：半角の英数字，アンダーバー（_），ピリオド（.）およびハイフン（-）

  クライアント相関 ID は，サービスリクエスタからの要求電文と，HCSC サーバで管理しているログおよびトレースを対応づけるために使用します。したがって，HCSC サーバに送信した要求電文ごとに異なる ID を指定する必要があります。

  クライアント相関 ID を省略する場合，null または空文字列（""）を指定します。

- msg（ユーザ電文）

  ユーザ電文は，サービスリクエスタからの要求電文（XML 形式）です。

要求電文がない場合，null または空文字列（""）を指定します。

null または空文字（""）を指定してデータ変換などで変更しない場合，ビジネスプロセスに空の電文を渡します。

例外

- CSCMsgServerException

  次の場合にスローします。

  ・メソッドのパラメタに不正な値が指定された場合

  ・要求電文および応答電文のデータ変換でエラーが発生した場合

  ・要求先のサービス部品でエラーが発生した場合

戻り値

- String

  ・通信モデルが同期の場合

  サービス部品の実行要求に対する応答電文（XML 形式）を返却します。

  応答電文がない場合，null を返却します。

  ・通信モデルが非同期の場合

  null を返却します。

invokeXML メソッド（形式 4）

説明

デフォルトのオペレーション名のサービスに対して，バイト配列型の XML 電文のメッセージを送信し，その応答を返します。応答も，バイト配列型になります。

ユーザ電文をバイト配列型で扱うカスタム受付の場合，このメソッドを使用することで，String 型への変換処理を省くことができます。

形式

```
public byte[] invokeXML(String cscCorrelationID, byte[] msg)
    throws CSCMsgServerException;
```

パラメタ

- cscCorrelationID（クライアント相関 ID）

  クライアント相関 ID は，サービスリクエスタからの要求電文を一意に識別するための相関識別子です。

  クライアント相関 ID が次の制限に従っていない場合，CSCMsgServerException 例外をスローします。

  ・サイズ：255 文字以内

  ・使用できる文字：半角の英数字，アンダーバー（_），ピリオド（.）およびハイフン（-）

  クライアント相関 ID は，サービスリクエスタからの要求電文と，HCSC サーバで管理しているログおよびトレースを対応づけるために使用します。したがって，HCSC サーバに送信した要求電文ごとに異なる ID を指定する必要があります。

クライアント相関 ID を省略する場合，null または空文字列（""）を指定します。

- msg（ユーザ電文）

  ユーザ電文は，サービスリクエスタからの要求電文（XML 形式）です。

  要求電文がない場合，null を指定します。

  null または空文字（""）を指定してデータ変換などで変更しない場合，ビジネスプロセスに空の電文を渡します。

例外

- CSCMsgServerException

  次の場合にスローします。

  ・メソッドのパラメタに不正な値が指定された場合

  ・要求電文および応答電文のデータ変換でエラーが発生した場合

  ・要求先のサービス部品でエラーが発生した場合

戻り値

- byte[]

  ・通信モデルが同期の場合

  サービス部品の実行要求に対する応答電文（XML 形式）を返却します。

  応答電文がない場合，null を返却します。

  ・通信モデルが非同期の場合

  null を返却します。

invokeBinary メソッド（形式 1）

説明

指定されたオペレーション名のサービスに対してバイナリ電文のメッセージを送信し，その応答を返します。

形式

```
public byte[] invokeBinary(String cscCorrelationID,
    String cscServiceOperationName, int requestMessageLength, byte[] msg)
    throws CSCMsgServerException;
```

パラメタ

- cscCorrelationID（クライアント相関 ID）

  クライアント相関 ID は，サービスリクエスタからの要求電文を一意に識別するための相関識別子です。

  クライアント相関 ID が次の制限に従っていない場合，CSCMsgServerException 例外をスローします。

  ・サイズ：255 文字以内

  ・使用できる文字：半角の英数字，アンダーバー（_），ピリオド（.）およびハイフン（-）

クライアント相関 ID は，サービスリクエスタからの要求電文と，HCSC サーバで管理しているログおよびトレースを対応づけるために使用します。したがって，HCSC サーバに送信した要求電文ごとに異なる ID を指定する必要があります。

クライアント相関 ID を省略する場合，null または空文字列（""）を指定します。

- cscServiceOperationName（オペレーション名）

  オペレーション名は，要求先のサービス名に対するオペレーション名です。

  オペレーション名は，開発環境で定義したサービス部品のオペレーション名を指定します。

  オペレーション名が次の制限に従っていない場合，CSCMsgServerException 例外をスローします。

  ・サイズ：255 バイト以内

  ・使用できる文字：XMLSchema の NCName 定義文字

  ユーザ定義受付定義画面で定義していないオペレーション名を指定した場合，CSCMsgServerException 例外をスローします。

  オペレーション名を省略する（デフォルトのオペレーション名を使用する）場合，null または空文字列（""）を指定します。

- requestMessageLength（ユーザ電文長）

  ユーザ電文長は，要求電文の長さです。

  ユーザ電文長が次の制限に従っていない場合，CSCMsgServerException 例外をスローします。

  ・0 以上の整数値

  ・ユーザ電文（msg パラメタ）のバイト長以下の整数値

  要求電文がない場合，0 を指定します。

  また，ユーザ電文（msg パラメタ）のバイト長より小さい値を指定した場合，要求電文の長さとして，ユーザ電文長（requestMessageLength パラメタ）に指定した値を使用します。つまり，サービス部品に渡す要求電文には，ユーザ電文（msg パラメタ）のユーザ電文長（requestMessageLength パラメタ）＋ 1 バイト目以降は含まれません。

- msg（ユーザ電文）

  ユーザ電文は，サービスリクエスタからの要求電文（バイナリ形式）です。

  要求電文がない場合，null または 0 バイトのバイト配列を指定します。

  null または 0 バイトのバイト配列を指定してデータ変換などで変更しない場合，ビジネスプロセスに空の電文を渡します。

例外

- CSCMsgServerException

  次の場合にスローします。

  ・メソッドのパラメタに不正な値が指定された場合

  ・要求電文および応答電文のデータ変換でエラーが発生した場合

  ・要求先のサービス部品でエラーが発生した場合

戻り値

- byte[]

  ・通信モデルが同期の場合

  サービス部品の実行要求に対する応答電文（バイナリ形式）を返却します。

  応答電文がない場合，null を返却します。

  ・通信モデルが非同期の場合

  null を返却します。

invokeBinary メソッド（形式 2）

説明

デフォルトのオペレーション名のサービスに対してバイナリ電文のメッセージを送信し，その応答を返します。

形式

```
public byte[] invokeBinary(String cscCorrelationID,
    int requestMessageLength, byte[] msg)
    throws CSCMsgServerException;
```

パラメタ

- cscCorrelationID（クライアント相関 ID）

  クライアント相関 ID は，サービスリクエスタからの要求電文を一意に識別するための相関識別子です。

  クライアント相関 ID が次の制限に従っていない場合，CSCMsgServerException 例外をスローします。

  ・サイズ：255 文字以内

  ・使用できる文字：半角の英数字，アンダーバー（_），ピリオド（.）およびハイフン（-）

  クライアント相関 ID は，サービスリクエスタからの要求電文と，HCSC サーバで管理しているログおよびトレースを対応づけるために使用します。したがって，HCSC サーバに送信した要求電文ごとに異なる ID を指定する必要があります。

  クライアント相関 ID を省略する場合，null または空文字列（""）を指定します。

- requestMessageLength（ユーザ電文長）

  ユーザ電文長は，要求電文の長さです。

  ユーザ電文長が次の制限に従っていない場合，CSCMsgServerException 例外をスローします。

  ・0 以上の整数値

  ・ユーザ電文（msg パラメタ）のバイト長以下の整数値

  要求電文がない場合，0 を指定します。

  また，ユーザ電文（msg パラメタ）のバイト長より小さい値を指定した場合，要求電文の長さとして，ユーザ電文長（requestMessageLength パラメタ）に指定した値を使用します。つまり，サービス部品に渡す要求電文には，ユーザ電文（msg パラメタ）のユーザ電文長（requestMessageLength パラメタ）＋ 1 バイト目以降は含まれません。

- msg（ユーザ電文）

  ユーザ電文は，サービスリクエスタからの要求電文（バイナリ形式）です。

  要求電文がない場合，null または 0 バイトのバイト配列を指定します。

  null または 0 バイトのバイト配列を指定してデータ変換などで変更しない場合，ビジネスプロセスに空の電文を渡します。

例外

- CSCMsgServerException

  次の場合にスローします。

  ・メソッドのパラメタに不正な値が指定された場合

  ・要求電文および応答電文のデータ変換でエラーが発生した場合

  ・要求先のサービス部品でエラーが発生した場合

戻り値

- byte[]

  ・通信モデルが同期の場合

  サービス部品の実行要求に対する応答電文（バイナリ形式）を返却します。

  応答電文がない場合，null を返却します。

  ・通信モデルが非同期の場合

  null を返却します。

getUserFile メソッド

説明

このメソッドは非推奨メソッドです。代わりに getUserFileForExEnv メソッドを使用してください。

このメソッドでは，実行環境上で独自定義ファイルの設定機能を使用して配置された独自定義ファイルは取得できません。

パラメタに指定されたファイル名のカスタム受付ユーザファイルを読み込んで，ファイルの内容をストリームとして返します。

このメソッドは，カスタム受付ユーザファイルのファイル形式に関係なく，ストリームとしてファイルの内容をそのまま返します。このメソッドの呼び出し元で，ファイル形式に応じて処理してください（Properties クラスの load メソッドでプロパティファイルとして読み込むなど）。

このメソッドで取得したストリームは，使用後に，呼び出し元で閉じてください。

形式

```
public InputStream getUserFile(String fileName);
```

パラメタ

- fileName

  カスタム受付ユーザファイルのファイル名です。

  開発環境でカスタム受付ユーザファイルのファイル名を指定します。

使用できる文字は任意です。

例外

なし。

戻り値

- InputStream

  カスタム受付ユーザファイルのストリームを返却します。

  パラメタに指定したファイル名のファイルが見つからなかった場合は null を返却します。

getUserFileForExEnv メソッド

説明

パラメタに指定されたファイル名のカスタム受付ユーザファイルを次の優先順位で読み込んで，ファイルの内容をストリームとして返します。

表 A-5　読み込まれる独自定義ファイルの優先順位

優先順位	独自定義ファイルの種類
1	<HCSC サーバランタイム定義ファイルの definition-file-base-path プロパティで指定したディレクトリ>¥<HCSC サーバ名>¥<受付 ID>配下にあるカスタム受付ユーザファイル
2	csccmctl コマンド（-operation set）で運用環境に設定したカスタム受付ユーザファイル
3	開発環境で定義したカスタム受付ユーザファイル

このメソッドは，カスタム受付ユーザファイルのファイル形式に関係なく，ストリームとしてファイルの内容をそのまま返します。このメソッドの呼び出し元で，ファイル形式に応じて処理してください（Properties クラスの load メソッドでプロパティファイルとして読み込むなど）。

このメソッドで取得したストリームは，使用後に，呼び出し元で閉じてください。

形式

```
public InputStream getUserFileForExEnv (String fileName)
    throws CSCMsgServerException;
```

パラメタ

- fileName

  カスタム受付ユーザファイルのファイル名です。

  カスタム受付ユーザファイルのファイル名を指定します。

  使用できる文字は任意です。

例外

- CSCMsgServerException

  次の場合にスローします。

  ・実行環境に配置した独自定義ファイルの読み込みに失敗した場合

戻り値

- InputStream

  カスタム受付ユーザファイルのストリームを返却します。

  パラメタに指定したファイル名のファイルが見つからなかった場合は null を返却します。

(4) API の使用例

カスタム受付フレームワークの API を利用して，受付処理を EJB（Stateless Session Bean）形式で実装する場合の実装例を次に示します。

- MyReception.java

```
package com.example.sample.reception.ejb;

import javax.ejb.EJBObject;

/**
 * 受付処理(Stateless Session Bean)の実装例(Remoteインターフェース)
 */
public interface MyReception extends EJBObject {
  public String invoke(String cscCorrelationID, String cscServiceOperationName, String ms
g) throws RemoteException;
}
```

- MyReceptionHome.java

```
package com.example.sample.reception.ejb;

import java.rmi.RemoteException;

import javax.ejb.CreateException;
import javax.ejb.EJBHome;

/**
 * 受付処理(Stateless Session Bean)の実装例(Homeインターフェース)
 */
public interface MyReceptionHome extends EJBHome {
  public MyReception create() throws RemoteException, CreateException;
}
```

- MyReceptionBean.java

```
package com.example.sample.reception.ejb;

import java.rmi.RemoteException;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.SessionBean;
import javax.ejb.SessionContext;

import jp.co.Hitachi.soft.csc.msg.message.reception.CSCMsgServerException;
import jp.co.Hitachi.soft.csc.msg.message.reception.custom.CSCMsgCustomServiceDelivery;
```

```
import jp.co.Hitachi.soft.csc.msg.message.reception.custom.CSCMsgCustomServiceDeliveryFac
tory;

/**
 * 受付処理(Stateless Session Bean)の実装例(Beanクラス)
 */
public class MyReceptionBean implements SessionBean {
  private static final CSCMsgCustomServiceDeliveryFactory factory =
      CSCMsgCustomServiceDeliveryFactory.newInstance();
  private CSCMsgCustomServiceDelivery delivery;
  private SessionContext context;

  public MyReceptionBean() {
  }

  public void ejbActivate() throws EJBException, RemoteException {
  }

  public void ejbCreate() throws CreateException {
    // CSCMsgCustomServiceDeliveryオブジェクトを生成
    delivery = factory.createCSCMsgCustomServiceDelivery();
  }

  public void ejbPassivate() throws EJBException, RemoteException {
  }

  public void ejbRemove() throws EJBException, RemoteException {
    // CSCMsgCustomServiceDeliveryオブジェクトを破棄
    delivery = null;
  }

  public String invoke(final String cscCorrelationID, final String cscServiceOperationNam
e, final String msg) {
    try {
      // カスタム受付フレームワークを呼び出す
      final String response = delivery.invokeXML(cscCorrelationID, cscServiceOperationNam
e, msg);
      return response;
    } catch (CSCMsgServerException e) {
      // カスタム受付フレームワークの例外をキャッチ
      // 例外処理を実装すること
    }
  }

  public void setSessionContext(final SessionContext sc) throws EJBException {
    context = sc;
  }
}
```

(5) 注意事項

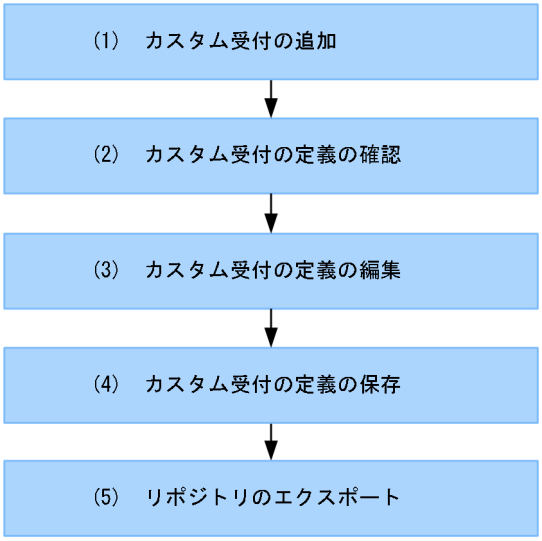
- カスタム受付フレームワークの API を呼び出す際は，カスタム受付が開始状態である必要があります。
- カスタム受付の開始操作を実行後，開始状態になるまでの間にカスタム受付フレームワークの API を呼び出した場合はエラーが発生します。

付録 A.4　カスタム受付の定義

ここでは，カスタム受付の定義について説明します。

カスタム受付の定義の流れを次の図に示します。

図 A–4　カスタム受付の定義の流れ



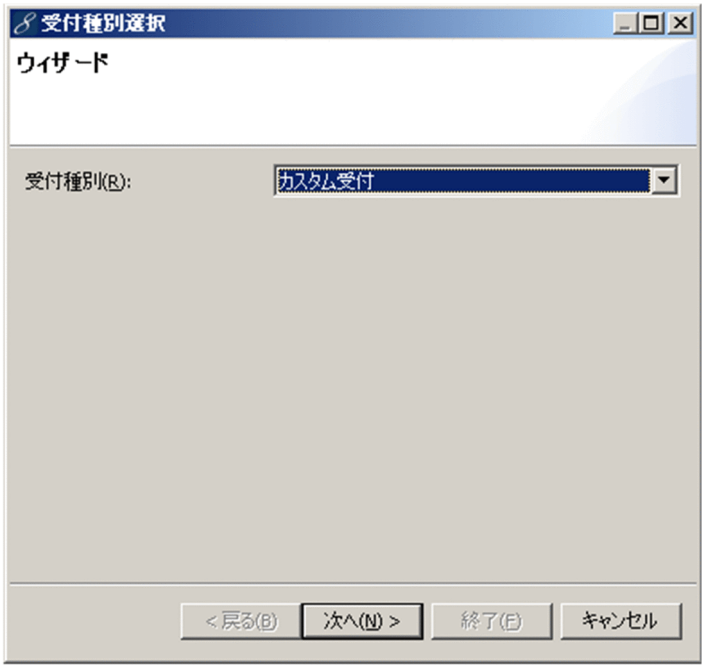
次に，カスタム受付の定義の流れに合わせてそれぞれの内容を説明します。

(1)　カスタム受付の追加

開発環境で作成したカスタム受付 EAR ファイルをインポートします。

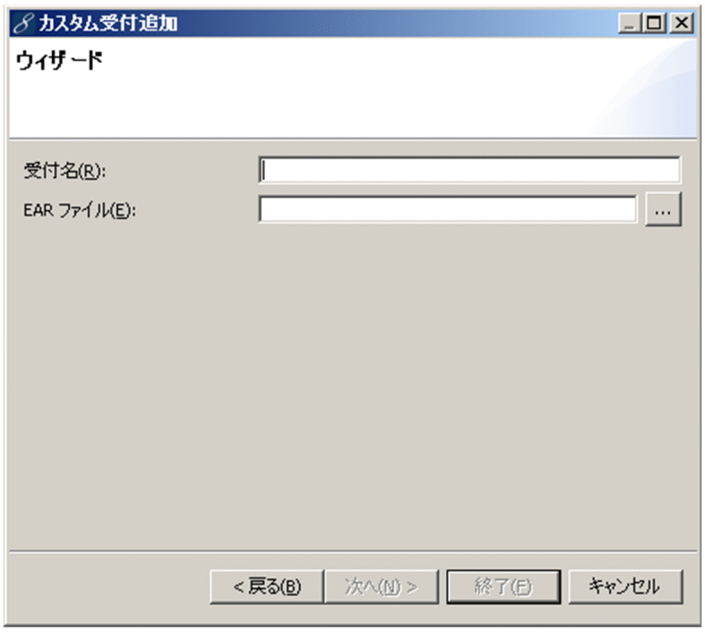
カスタム受付を新規に追加する手順を次に示します。

1. ツリービューのサービス定義一覧で，カスタム受付を追加するサービス（ビジネスプロセス）を選択して，右クリックします。
   サービス一覧のポップアップメニューが表示されます。
2. ポップアップメニューから［ユーザ定義受付（呼出先固定）追加］を選択します。
   受付種別選択ウィザードが表示されます。



3. [受付種別] のドロップダウンリストから，[カスタム受付] を選択します。

4. [次へ] ボタンをクリックします。
   カスタム受付追加ウィザードが表示されます。



5. [受付名] と [EAR ファイル] を指定します。

   - 受付名
     カスタム受付の名称を 1～40 バイトで指定します。
     受付名は，ビジネスプロセス内で一意である必要があります。
   - EAR ファイル

インポートする EAR ファイルを絶対パスで指定します。相対パスや UNC 形式で指定しないでください。

[...] ボタンをクリックして表示されるダイアログから EAR ファイルを選択することもできます。

6. [終了] ボタンをクリックします。

カスタム受付がビジネスプロセスに追加され，ユーザ定義受付定義画面が表示されます。

ユーザ定義受付定義画面の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「1.2.5 ユーザ定義受付定義画面」を参照してください。

(2) カスタム受付の定義の確認

カスタム受付をインポートしたあとに，定義された内容を確認してください。カスタム受付の内容は，ユーザ定義受付定義画面で確認できます。ユーザ定義受付定義画面は，カスタム受付追加ウィザードでカスタム受付を追加したあとに表示されます。

ユーザ定義受付定義画面を任意のタイミングで表示したい場合は，次の手順で表示させてください。

1. ツリービューのサービス定義一覧で，カスタム受付をダブルクリックします。

ユーザ定義受付定義画面が表示されます。

(3) カスタム受付の定義の編集

ユーザ定義受付定義画面で，受付名，受付 ID，オペレーション名などインポート済みのカスタム受付の定義を編集します。

ユーザ定義受付定義画面を任意のタイミングで表示させる手順については，「(2) カスタム受付の定義の確認」を参照してください。また，ユーザ定義受付定義画面および入力できる値については，マニュアル「サービスプラットフォーム リファレンス」の「1.2.5 ユーザ定義受付定義画面」を参照してください。

(4) カスタム受付の定義の保存

ユーザ定義受付定義画面で編集中のカスタム受付を編集途中や編集終了時に保存できます。カスタム受付の内容はリポジトリに保存されます。

ユーザ定義受付定義画面を閉じてカスタム受付の定義を終了する場合に，カスタム受付が保存されていないときは，リソース保管ダイアログが表示されます。リソース保管ダイアログでは，保存されていないカスタム受付の定義を保存できます。

カスタム受付を保存する方法を次に示します。

方法 1

Eclipse のメニューから，[ファイル] － [保存] を選択します。

方法 2

Eclipse のメニューから，[ファイル] － [すべて保管] を選択します。

方法 3

Eclipse のツールバーの，［保存］アイコンをクリックします。

方法 4

ユーザ定義受付定義画面で，［Ctrl］キー＋［S］キーを押します。

なお，不正なデータが入力されている場合，カスタム受付を保存できないことがあります。その場合，表示されているメッセージに従って対処してください。

(5) リポジトリのエクスポート

定義したカスタム受付を含むリポジトリは，1 つの ZIP ファイルにして，指定したディレクトリに保存することができます。

リポジトリをエクスポートする手順を次に示します。

1. Eclipse のメニューから，［HCSC-Definer］－［定義情報管理］－［全定義情報エクスポート］を選択します。

   ［リポジトリエクスポート］ダイアログが表示されます。

2. 保存するディレクトリを指定し，保存する ZIP ファイルのファイル名を入力します。

3. ［保存］ボタンをクリックします。

指定したディレクトリに ZIP ファイルが保存されます。

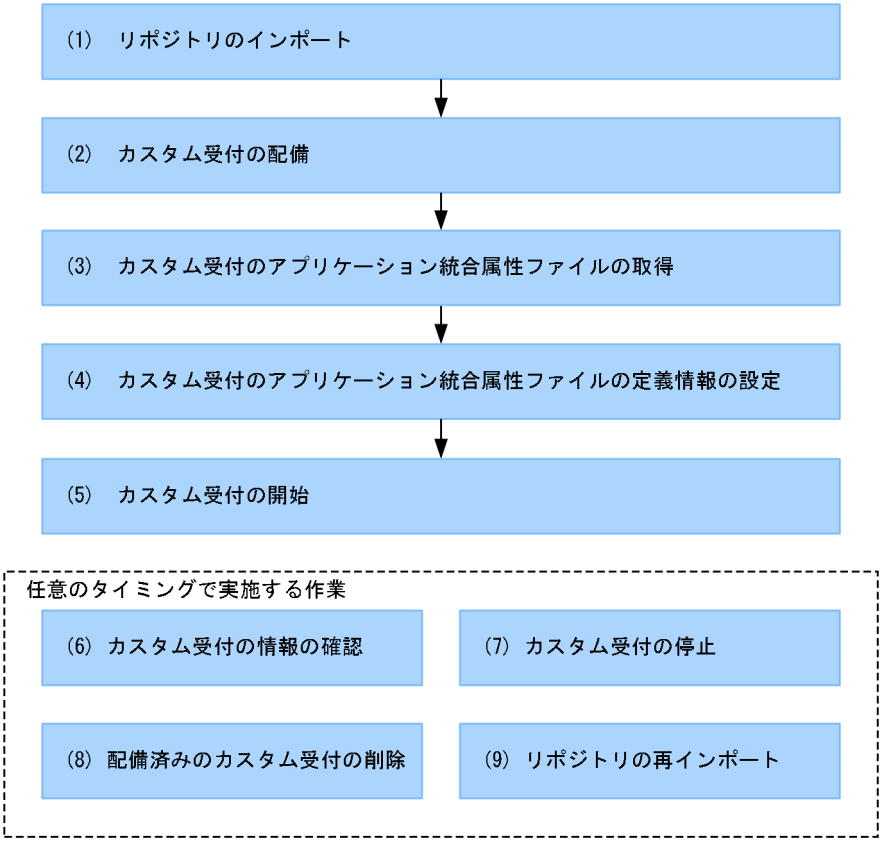
保存先のディレクトリに同じファイル名称のファイルがある場合，上書きを確認するダイアログが表示されます。上書きしてよいときは，［はい］ボタンをクリックして保存します。

付録 A.5 カスタム受付の運用

ここでは，カスタム受付の運用について説明します。

カスタム受付の運用の流れを次の図に示します。

図 A-5　カスタム受付の運用の流れ



次に，カスタム受付の運用の流れに合わせてそれぞれの内容を説明します。

(1)　リポジトリのインポート

開発環境で作成したカスタム受付を運用環境で使用する場合，ZIP ファイル形式で保存されたリポジトリ情報をリポジトリにインポートします。インポートする方法については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「4.3　リポジトリのインポート」を参照してください。

リポジトリにインポートするには，cscrepctl コマンドを実行します。詳細は，マニュアル「サービスプラットフォーム リファレンス」の「cscrepctl（インポート/エクスポート）」を参照してください。

(2)　カスタム受付の配備

カスタム受付を配備するには，運用環境で csccompodeploy コマンドを実行します。詳細は，マニュアル「サービスプラットフォーム リファレンス」の「csccompodeploy（HCSC コンポーネントの配備）」を参照してください。

カスタム受付を実行環境に配備する場合の前提条件，および配備方法は，ユーザ定義受付を配備する方法と同じです。詳細は，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「3.1.14 ユーザ定義受付を配備する」を参照してください。

(3)　カスタム受付のアプリケーション統合属性ファイルの取得

カスタム受付のアプリケーション統合属性ファイルを取得するときの前提条件，および取得方法について説明します。

(a)　前提条件

カスタム受付のアプリケーション統合属性ファイルを取得するには，HCSC サーバが起動状態である必要があります。

HCSC サーバの状態の確認方法については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「5.3.16　HCSC サーバの情報を確認する」を参照してください。

(b)　取得方法

カスタム受付のアプリケーション統合属性ファイルを取得するには，運用環境で csccompoconfig コマンドを実行します。

詳細は，マニュアル「サービスプラットフォーム リファレンス」の「csccompoconfig（HCSC コンポーネントの定義)」を参照してください。

csccompoconfig コマンドの実行形式を次に示します。

```
csccompoconfig -user 〈ログインユーザID〉 -pass 〈ログインパスワード〉 -operation get -csc 〈HCSCサーバ名〉 -name 〈カスタム受付の受付ID〉
```

(4)　カスタム受付のアプリケーション統合属性ファイルの定義情報の設定

カスタム受付のアプリケーション統合属性ファイルの定義情報を設定するときの前提条件，および設定方法について説明します。

(a)　前提条件

カスタム受付のアプリケーション統合属性ファイルの定義情報を設定するには，HCSC サーバが起動状態である必要があります。

HCSC サーバの状態の確認方法については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「5.3.16　HCSC サーバの情報を確認する」を参照してください。

(b)　設定方法

カスタム受付のアプリケーション統合属性ファイルの定義情報を設定するには，運用環境で csccompoconfig コマンドを実行します。

詳細は，マニュアル「サービスプラットフォーム リファレンス」の「csccompoconfig（HCSC コンポーネントの定義)」を参照してください。

csccompoconfig コマンドの実行形式を次に示します。

HCSC サーバに配備されているカスタム受付に定義情報を設定する場合

HCSC サーバ名，およびカスタム受付の受付 ID を指定して実行します。

指定した HCSC サーバ上のカスタム受付に，アプリケーション統合属性ファイルで指定した定義情報が設定されます。

```
csccompoconfig -user 〈ログインユーザID〉 -pass 〈ログインパスワード〉 -operation set -propfi
le 〈アプリケーション統合属性ファイル〉 -csc 〈HCSCサーバ名〉 -name 〈カスタム受付の受付ID〉
```

クラスタ内の HCSC サーバに配備されているカスタム受付に定義情報を設定する場合

クラスタ名，およびカスタム受付の受付 ID を指定して実行します。

指定したクラスタに配備されているカスタム受付に，アプリケーション統合属性ファイルで指定した定義情報が設定されます。

```
csccompoconfig -user 〈ログインユーザID〉 -pass 〈ログインパスワード〉 -operation set -propfi
le 〈アプリケーション統合属性ファイル〉 -cluster 〈クラスタ名〉 -name 〈カスタム受付の受付ID〉
```

(5)　カスタム受付の開始

カスタム受付を開始するときの前提条件，および開始方法について説明します。

(a)　前提条件

カスタム受付を開始するには，次の条件を満たしている必要があります。

- HCSC サーバが起動状態であること。
- カスタム受付が定義されているビジネスプロセスが開始状態であること。

HCSC サーバの状態の確認方法については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「5.3.16　HCSC サーバの情報を確認する」を参照してください。

(b)　開始方法（画面の場合）

運用環境の画面を利用してカスタム受付を開始する手順を次に示します。

1. ツリービュー上で，開始するユーザ定義受付（カスタム受付）をダブルクリックします。

   ダブルクリックしたカスタム受付の情報がエディタエリアの稼働運用ページに表示されます。

   カスタム受付の稼働運用ページの内容については，マニュアル「サービスプラットフォーム リファレンス」の「4.4　稼働運用ページ」のユーザ定義受付稼働運用に関する説明を参照してください。

2. 稼働運用ページに表示された内容を確認し，［開始］ボタンをクリックします。

   カスタム受付の開始を確認するダイアログが表示されます。

3. ダイアログに表示された内容を確認し，［OK］ボタンをクリックします。

   カスタム受付の開始処理が開始されます。開始処理中のログはコンソールビューに出力されます。

   ［キャンセル］ボタンをクリックした場合は，開始処理を実行しないで稼働運用ページに戻ります。

   カスタム受付の開始処理が完了すると，実行結果を示すダイアログが表示されます。

4. 実行結果を示すダイアログを確認します。

- 開始処理が成功した場合

  ［OK］ボタンをクリックして，稼働運用ページに戻ります。

  稼働運用ページでは，カスタム受付の稼働状態が「開始状態」になります。また，［開始］ボタンが非活性化し，［停止］ボタンが活性化します。

- 開始処理が失敗した場合

  ダイアログにエラーの内容が表示されます。詳細情報に表示されるメッセージ ID を参照して対処してください。

(c) 開始方法（コマンドの場合）

カスタム受付を開始するには，運用環境で csccompostart コマンドを実行します。

詳細は，マニュアル「サービスプラットフォーム リファレンス」の「csccompostart（HCSC コンポーネントの開始)」を参照してください。

csccompostart コマンドの実行形式を次に示します。

クラスタ内のすべての HCSC サーバに配備されているカスタム受付を一括して開始する場合（クラスタ単位）

クラスタ名と-all オプションを指定して実行します。

-cluster オプションで指定したクラスタ内のすべての HCSC サーバに配備されているカスタム受付が一括して開始されます（開始されていないサービスアダプタおよびビジネスプロセスも同時にすべて開始されます)。

```
csccompostart -user 〈ログインユーザID〉 -pass 〈ログインパスワード〉 -cluster 〈クラスタ名〉 -all
```

HCSC サーバに配備されているカスタム受付を一括して開始する場合（HCSC サーバ単位）

HCSC サーバ名と-all オプションを指定して実行します。

-csc オプションで指定した HCSC サーバに配備されているカスタム受付が開始されます（開始されていないサービスアダプタおよびビジネスプロセスも同時にすべて開始されます)。

```
csccompostart -user 〈ログインユーザID〉 -pass 〈ログインパスワード〉 -csc 〈HCSCサーバ名〉 -all
```

クラスタ内のすべての HCSC サーバに配備されている特定のカスタム受付を開始する場合

クラスタ名とカスタム受付の受付 ID を指定して実行します。

```
csccompostart -user 〈ログインユーザID〉 -pass 〈ログインパスワード〉 -cluster 〈クラスタ名〉 -name 〈カスタム受付の受付ID〉
```

HCSC サーバに配備されている特定のカスタム受付を開始する場合

HCSC サーバ名とカスタム受付の受付 ID を指定して実行します。

```
csccompostart -user〈ログインユーザID〉-pass〈ログインパスワード〉-csc〈HCSCサーバ名〉-name〈カスタム受付の受付ID〉
```

(6) カスタム受付の情報の確認

カスタム受付の情報を参照するには，cscsvcls コマンドを実行します。詳細は，マニュアル「サービスプラットフォーム リファレンス」の「cscsvcls（サービス情報の表示)」を参照してください。

cscsvcls コマンドを実行すると，開発環境または運用環境に設定されている定義ファイルのファイル名とその状態を確認できます。確認する方法については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「3.1.7　サービス情報を確認する」を参照してください。

(7) カスタム受付の停止

カスタム受付を停止するときの前提条件，および停止方法について説明します。

(a) 前提条件

カスタム受付を停止するには，次の条件を満たしている必要があります。

- HCSC サーバが起動状態であること

  HCSC サーバの状態の確認方法については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「5.3.16　HCSC サーバの情報を確認する」を参照してください。
- カスタム受付が開始状態であること

  カスタム受付の状態の確認方法は，ユーザ定義受付の状態を確認する方法と同じです。詳細は，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「5.3.21　ユーザ定義受付の情報を確認する」を参照してください。

(b) 停止方法

カスタム受付を停止するには，運用環境で csccompostop コマンドを実行します。

詳細は，マニュアル「サービスプラットフォーム リファレンス」の「csccompostop（HCSC コンポーネントの停止)」を参照してください。

csccompostop コマンドの実行形式を次に示します。

クラスタ内のすべての HCSC サーバに配備されているカスタム受付を一括して停止する場合（クラスタ単位)

クラスタ名と-all オプションを指定して実行します。

-cluster オプションで指定したクラスタ内のすべての HCSC サーバに配備されているカスタム受付が一括して停止されます（停止されていないサービスアダプタおよびビジネスプロセスも同時にすべて停止されます)。

```
csccompostop -user 〈ログインユーザID〉 -pass 〈ログインパスワード〉 -cluster 〈クラスタ名〉 -a
ll
```

HCSC サーバに配備されているカスタム受付を一括して停止する場合（HCSC サーバ単位）

HCSC サーバ名と-all オプションを指定して実行します。

-csc オプションで指定した HCSC サーバに配備されているカスタム受付が停止されます（停止されていないサービスアダプタおよびビジネスプロセスも同時にすべて停止されます）。

```
csccompostop -user 〈ログインユーザID〉 -pass 〈ログインパスワード〉 -csc 〈HCSCサーバ名〉 -all
```

クラスタ内のすべての HCSC サーバに配備されている特定のカスタム受付を停止する場合

クラスタ名とカスタム受付の受付 ID を指定して実行します。

```
csccompostop -user 〈ログインユーザID〉 -pass 〈ログインパスワード〉 -cluster 〈クラスタ名〉 -n
ame 〈カスタム受付の受付ID〉
```

HCSC サーバに配備されている特定のカスタム受付を停止する場合

HCSC サーバ名とカスタム受付の受付 ID を指定して実行します。

```
csccompostop -user 〈ログインユーザID〉 -pass 〈ログインパスワード〉 -csc 〈HCSCサーバ名〉 -nam
e 〈カスタム受付の受付ID〉
```

なお，csccompostop コマンドの停止処理では，CSC が保持するカスタム受付の状態を「停止状態」に更新してからカスタム受付のアプリケーションを停止します。カスタム受付の停止処理中にリクエストを受信した場合，CSC が保持するカスタム受付と，カスタム受付のアプリケーションが両方とも起動していれば，リクエストは正常に受け付けられます。

(8)　配備済みのカスタム受付の削除

実行環境に配備済みのカスタム受付を削除するには，運用環境で csccompoundeploy コマンドを実行します。詳細は，マニュアル「サービスプラットフォーム リファレンス」の「csccompoundeploy（配備済みの HCSC コンポーネントの削除)」を参照してください。

配備済みのカスタム受付を削除する場合の前提条件，および削除方法は，ユーザ定義受付を削除する方法と同じです。削除する方法については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「3.3.3　ユーザ定義受付を削除する」を参照してください。

(9)　リポジトリの再インポート

運用環境にリポジトリをインポートしたあとに，開発環境でカスタム受付の定義を編集した場合，次の手順でリポジトリを再インポートします。

1. リポジトリのエクスポート

編集後の定義を保存したリポジトリをエクスポートします。

リポジトリのエクスポートは開発環境で実行します。

リポジトリのエクスポートの詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「3.2.2　リポジトリをエクスポートする」を参照してください。

2. リポジトリのインポート

開発環境でエクスポートしたリポジトリを運用環境にインポートします。

リポジトリのインポートの詳細は，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「4.3　リポジトリのインポート」を参照してください。

付録 A.6　カスタム受付のチューニング

ここでは，カスタム受付の同時実行数，およびトランザクションタイムアウトの設定方法について説明します。

(1)　同時実行数の設定方法（カスタム受付）

同時実行数の設定方法を次に示します。

(a)　受付処理の場合

受付処理の同時実行数は，受付処理の処理内容に合わせて，カスタム受付の開発時および運用時に設定できます。

- **カスタム受付開発時の設定**

  受付処理の処理内容に合わせて，アプリケーション属性ファイル（cosminexus.xml）の要素に値を設定します。

- **カスタム受付運用時の設定**

  受付処理の処理内容に合わせて，アプリケーション統合属性ファイルの要素に値を設定します。

アプリケーション属性ファイル（cosminexus.xml）の詳細は，マニュアル「アプリケーションサーバ リファレンス 定義編(アプリケーション／リソース定義)」の「2.　アプリケーション属性ファイル(cosminexus.xml)」を参照してください。アプリケーション統合属性ファイルの詳細は，マニュアル「アプリケーションサーバ リファレンス 定義編(アプリケーション／リソース定義)」の「3.1　アプリケーション統合属性ファイル」を参照してください。

(b)　カスタム受付フレームワークの場合

カスタム受付フレームワークの同時実行数は，カスタム受付の開発時には設定できません。カスタム受付の運用時に設定できます。

アプリケーション統合属性ファイルの要素に値を設定します。

設定する要素については，次に示す XPath 式を参照してください。

- //hitachi-application-all-property/ejb-jar/hitachi-ejb-jar-property/display-name[.='cscmsg_urecp_custom']/../../hitachi-session-bean-property/display-name[.='CSCMsgCustomServiceDelivery']/../session-runtime/stateless/pooled-instance/minimum
- //hitachi-application-all-property/ejb-jar/hitachi-ejb-jar-property/display-name[.='cscmsg_urecp_custom']/../../hitachi-session-bean-property/display-name[.='CSCMsgCustomServiceDelivery']/../session-runtime/stateless/pooled-instance/maximum

(2) トランザクションタイムアウト時間の設定方法（カスタム受付）

トランザクションタイムアウト時間の設定方法を次に示します。

(a) 受付処理の場合

受付処理のトランザクションタイムアウトは，受付処理の実装形式に合わせて，カスタム受付の開発時および運用時に設定できます。

- **カスタム受付開発時の設定**
  受付処理の処理内容に合わせて，アプリケーション属性ファイル（cosminexus.xml）の要素に値を設定します。
- **カスタム受付運用時の設定**
  受付処理の処理内容に合わせて，アプリケーション統合属性ファイルの要素に値を設定します。

アプリケーション属性ファイル（cosminexus.xml）の詳細は，マニュアル「アプリケーションサーバ リファレンス 定義編(アプリケーション／リソース定義)」の「2. アプリケーション属性ファイル(cosminexus.xml)」を参照してください。アプリケーション統合属性ファイルの詳細は，マニュアル「アプリケーションサーバ リファレンス 定義編(アプリケーション／リソース定義)」の「3.1 アプリケーション統合属性ファイル」を参照してください。

(b) カスタム受付フレームワークの場合

カスタム受付フレームワークのトランザクションタイムアウト時間は，カスタム受付の開発時には設定できません。カスタム受付の運用時に設定できます。

アプリケーション統合属性ファイルの要素に値を設定します。

設定する要素については，次に示す XPath 式を参照してください。

- //hitachi-application-all-property/ejb-jar/hitachi-ejb-jar-property/display-name[.='cscmsg_urecp_custom']/../../hitachi-session-bean-property/display-name[.='CSCMsgCustomServiceDelivery']/../ejb-transaction-timeout/method/method-name[.='*']/../../transaction-timeout

付録 A.7　カスタム受付でのエラーの伝わり方

サービスリクエスタからカスタム受付を使用してサービス部品を呼び出すときに，障害が発生すると，サービスリクエスタにエラーがリターンします。ここでは，エラーが発生した個所からサービスリクエスタまでどのようにエラーが伝わるかについて説明します。

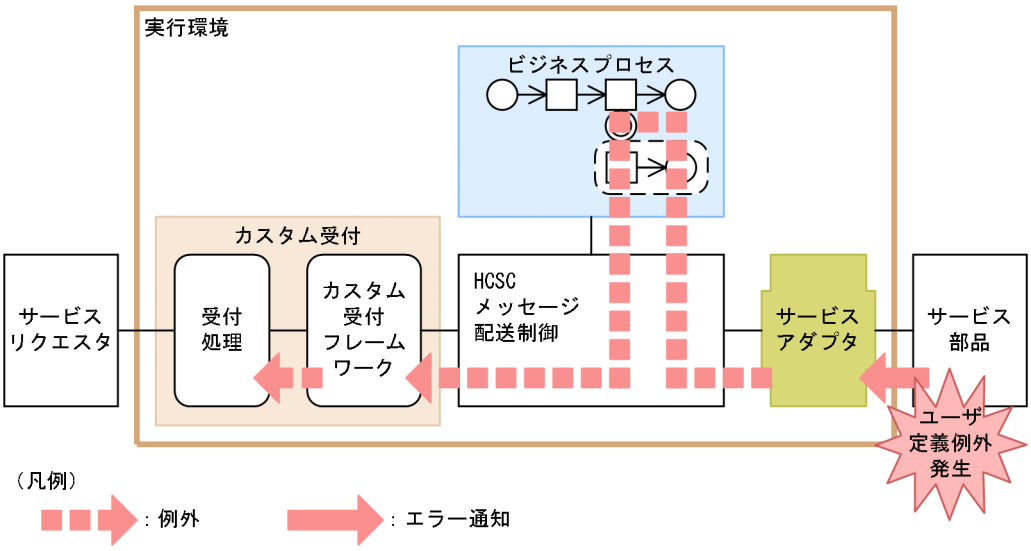
(1)　サービス部品からユーザ定義例外のエラーがリターンした場合（ビジネスプロセスを使用するとき）

ここでは，サービス部品からユーザ定義例外のエラーがリターンした場合（ビジネスプロセスを使用するとき）のエラーの伝わり方を説明します。

(a)　フォルト処理で障害情報をサービス電文にデータ変換しない場合

サービス部品からユーザ定義例外のエラーがリターンした場合（ビジネスプロセスを使用するとき）のカスタム受付でのエラーの伝わり方（フォルト処理なし）を次の図に示します。

図 A-6　サービス部品からユーザ定義例外のエラーがリターンした場合（ビジネスプロセスを使用するとき）のカスタム受付でのエラーの伝わり方（フォルト処理なし）

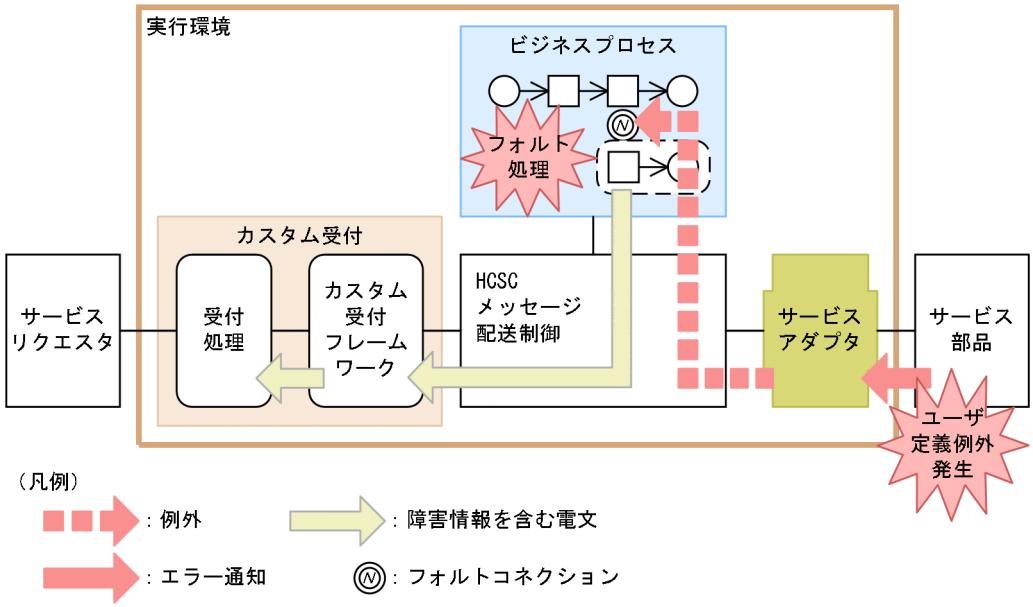


サービス部品で発生した例外は，そのまま例外としてカスタム受付フレームワークに伝わります。その例外をキャッチしたカスタム受付フレームワークは，CSCMsgServerException を受付処理に再スローします。

(b)　フォルト処理で障害情報をサービス電文にデータ変換する場合

サービス部品からユーザ定義例外のエラーがリターンした場合（ビジネスプロセスを使用するとき）のカスタム受付でのエラーの伝わり方（フォルト処理あり）を次の図に示します。

図 A-7　サービス部品からユーザ定義例外のエラーがリターンした場合（ビジネスプロセスを使用するとき）のカスタム受付でのエラーの伝わり方（フォルト処理あり）



サービス部品で発生した例外は，そのまま例外としてビジネスプロセスのフォルト処理に伝わります。そのあと，ビジネスプロセスのフォルト処理でその障害情報を応答電文にデータ変換し，障害情報を含んだ応答電文として，以降の処理に返します。この場合，HCSC メッセージ配送制御を経由して，カスタム受付フレームワークに応答電文が返ります。カスタム受付フレームワークは，通常の応答電文と同様，その応答電文を受付処理に返します。

(2)　サービス部品からユーザ定義例外以外のエラーがリターンした場合（ビジネスプロセスを使用するとき）

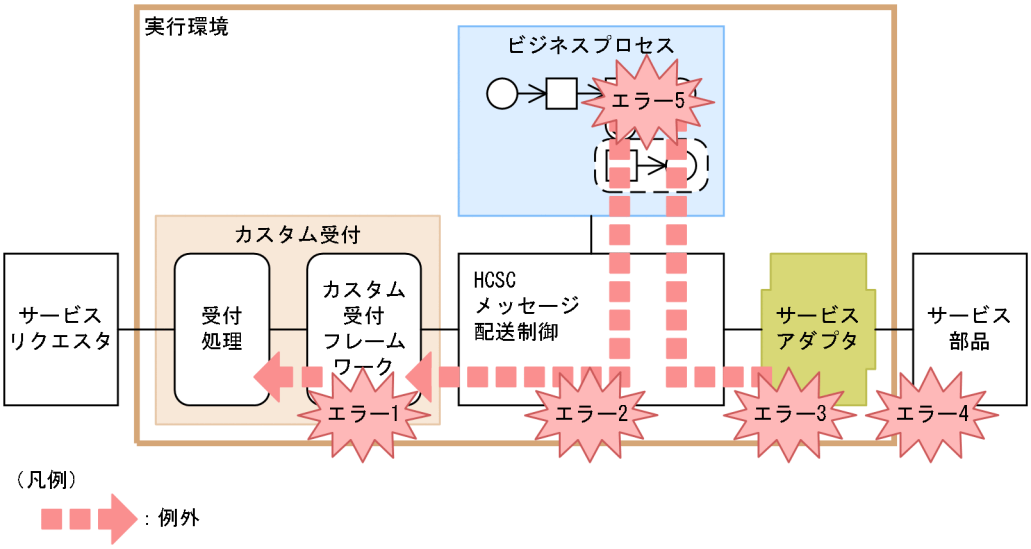
ここでは，サービス部品からユーザ定義例外以外のエラーがリターンした場合（ビジネスプロセスを使用するとき）のエラーの伝わり方を説明します。

(3)　HCSC サーバからエラーがリターンした場合（ビジネスプロセスを使用するとき）

ここでは，HCSC サーバからエラーがリターンした場合（ビジネスプロセスを使用するとき）のエラーの伝わり方を説明します。

HCSC サーバからエラーがリターンした場合（ビジネスプロセスを使用するとき）のカスタム受付でのエラーの伝わり方を次の図に示します。

図 A-8　HCSC サーバからエラーがリターンした場合（ビジネスプロセスを使用するとき）のカスタム受付でのエラーの伝わり方



図中の各エラーには，次に示すケースが該当します。

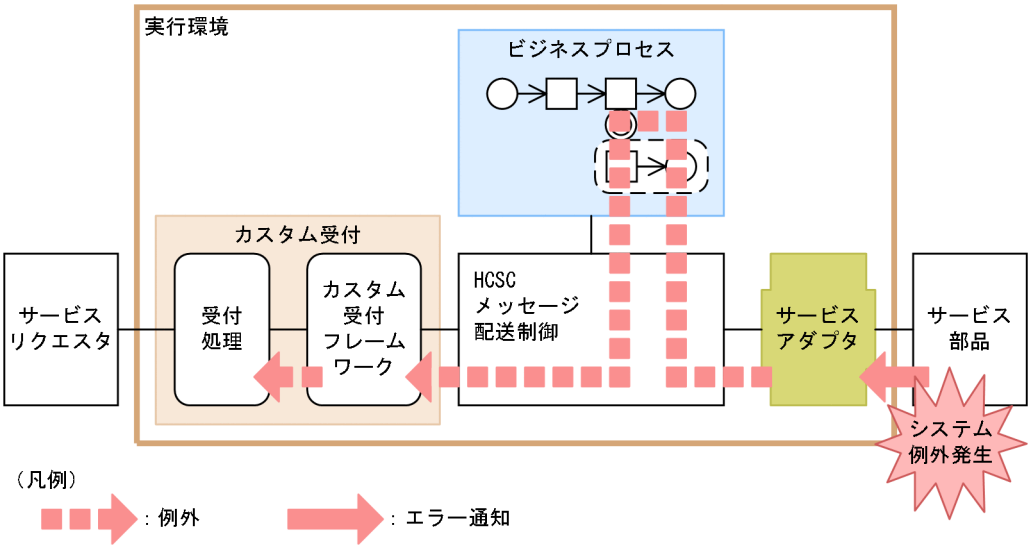
- エラー 1：要求パラメタ不正など
- エラー 2：宛先（ロケーション）が見つからない，サービスアダプタが停止しているなど
- エラー 3：データ変換に失敗したなど
- エラー 4：宛先不正，サービス部品が停止，通信障害など
- エラー 5：ビジネスプロセス処理上での例外エラーなど

HCSC サーバで図中のエラー 1～エラー 5 のどれかを検知した場合，カスタム受付フレームワークは，発生したエラーの情報を CSCMsgServerException で，受付処理にスローします。

(a)　フォルト処理で障害情報をサービス電文にデータ変換しない場合

サービス部品からユーザ定義例外以外のエラーがリターンした場合（ビジネスプロセスを使用するとき）のカスタム受付でのエラーの伝わり方（フォルト処理なし）を次の図に示します。

図 A-9　サービス部品からユーザ定義例外以外のエラーがリターンした場合（ビジネスプロセスを使用するとき）のカスタム受付でのエラーの伝わり方（フォルト処理なし）

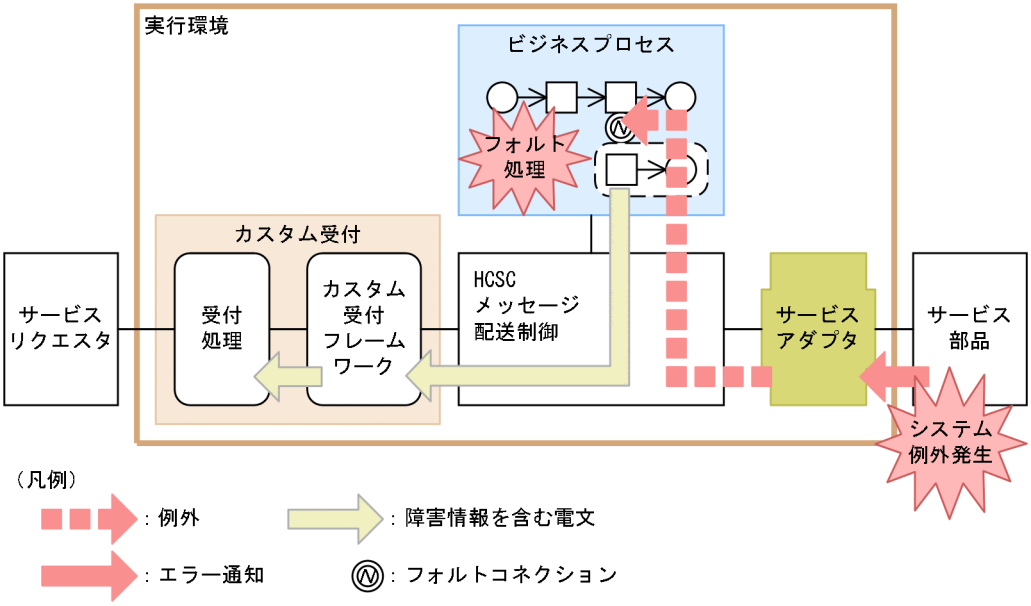


サービス部品で想定外の例外が発生した場合，RuntimeException（システム例外）としてカスタム受付フレームワークに伝わります。その例外をキャッチしたカスタム受付フレームワークは，キャッチしたRuntimeException（システム例外）を，そのまま受付処理に再スローします。

(b)　フォルト処理で障害情報をサービス電文にデータ変換する場合

サービス部品からユーザ定義例外以外のエラーがリターンした場合（ビジネスプロセスを使用するとき）のカスタム受付でのエラーの伝わり方（フォルト処理あり）を次の図に示します。

図 A-10　サービス部品からユーザ定義例外以外のエラーがリターンした場合（ビジネスプロセスを使用するとき）のカスタム受付でのエラーの伝わり方（フォルト処理あり）



サービス部品で発生した例外は，そのまま例外としてビジネスプロセスのフォルト処理に伝わります。そのあと，ビジネスプロセスのフォルト処理でその障害情報を応答電文にデータ変換し，障害情報を含んだ

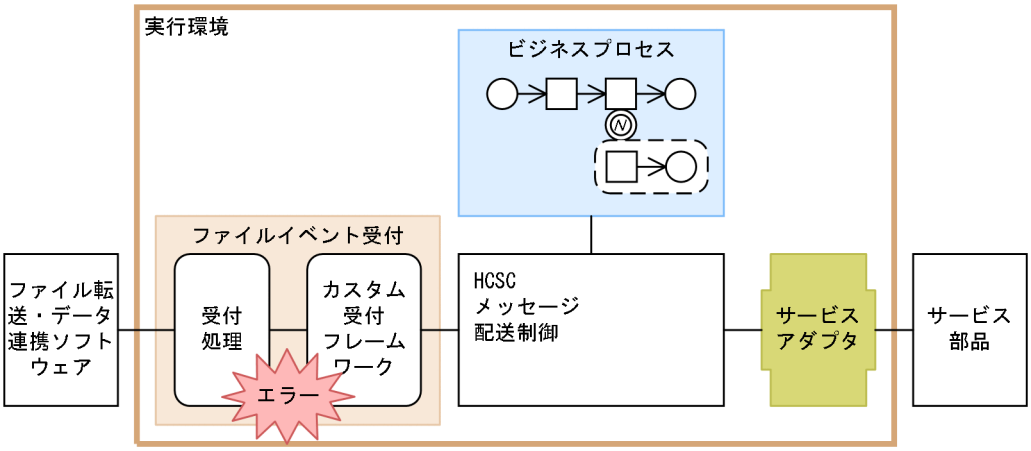
応答電文として，以降の処理に返します。この場合，HCSC メッセージ配送制御を経由して，カスタム受付フレームワークに応答電文が返ります。カスタム受付フレームワークは，通常の応答電文と同様，その応答電文を受付処理に返します。

(4)　ファイルイベント受付がエラーを検知した場合

ここでは，ファイル監視中にファイルイベント受付でエラーを検知した場合の要因の特定と対処方法について説明します。

ファイルイベント受付がエラーを検知する個所を次の図に示します。

図 A-11　ファイルイベント受付がエラーを検知する個所



ファイルイベント受付で検知するエラーは，監視対象のファイルによって異なります。

●更新完了通知ファイルを監視する場合

図中のエラーが発生する要因を次に示します。

- 取得対象ファイルが存在しない
- ロックファイルの作成失敗
- ロックファイルのロック取得失敗
- ロックファイルのロック解放失敗
- 更新完了通知ファイルのリネーム失敗

ファイルイベント受付は，障害の内容を基にメッセージテキストを作成し，メッセージログに出力します。ファイルイベントトレースを確認し，次の対策をしてください。

- **ファイルイベントトレースの取得ポイント「IN」にトレースが出力されている場合**
  ビジネスプロセス内での処理は正常終了しているため，回復処理としてメッセージログに出力されるエラーメッセージを参照し，対策をしてください。
- **ファイルイベントトレースの取得ポイント「IN」にトレースが出力されていない場合**
  メッセージログに出力されるエラーメッセージを参照し，対策をしてください。対策したあとで，更新完了通知ファイルがリネームされている場合は，再度ファイル転送・データ連携ソフトウェア

から更新完了通知ファイルを作成することで対処できます。リネームされていない場合は，次のファイル監視の間隔でエラーが検知されます。

●取得対象ファイルを監視する場合

図中のエラーが発生する要因を次に示します。

- 取得対象ファイルが存在しない
- ロックファイルの作成失敗
- ロックファイルのロック取得失敗
- ロックファイルのロック解放失敗
- 取得対象ファイルのリネーム失敗

ファイルイベント受付は，障害の内容を基にメッセージテキストを作成し，メッセージログに出力します。ファイルイベントトレースを確認し，次の対策をしてください。

- **ファイルイベントトレースの取得ポイント「IN」にトレースが出力されている場合**

  ビジネスプロセス内での処理は正常終了しているため，メッセージログに出力されるエラーメッセージを参照し，対策をしてください。

- **ファイルイベントトレースの取得ポイント「IN」にトレースが出力されていない場合**

  メッセージログに出力されるエラーメッセージを参照し，対策をしてください。対策した後で，取得対象ファイルがリネームされている場合は，再度ファイル転送・データ連携ソフトウェアから取得対象ファイルを作成することで対処できます。リネームされていない場合は，次のファイル監視の間隔でエラーが検知されます。

> **参考**
>
> リネーム後の取得対象ファイルが残っている場合は，ファイルイベントトレースを確認し，該当するファイルに関する処理が終了しているか確認します。
>
> 処理が終了している場合は，リネーム後の取得対象ファイルを削除できます。
>
> 再度ファイル転送・データ連携ソフトウェアから取得対象ファイルを作成する場合は，リネーム後の取得対象ファイルを削除してから行う必要があります。

付録 A.8　障害情報の取得（カスタム受付）

カスタム受付の運用時に障害が発生すると，障害対策に必要な次の情報が出力されます。

- メッセージログ
- リクエストトレース
- 性能解析トレース
- ユーザ電文トレース

ここでは，ログおよび各種トレースの取得方法について説明します。

(1)　メッセージログ（カスタム受付）

メッセージログは，次に示すログに出力されます。

- HCSC-Manager のログ
- 統合メッセージログ
- J2EE サーバの稼働ログ

メッセージログの取得方法および出力先については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「7.4.1　メッセージログ」に関する内容を参照してください。また，詳細は，マニュアル「サービスプラットフォーム メッセージ」の「2.　メッセージ一覧」を参照してください。

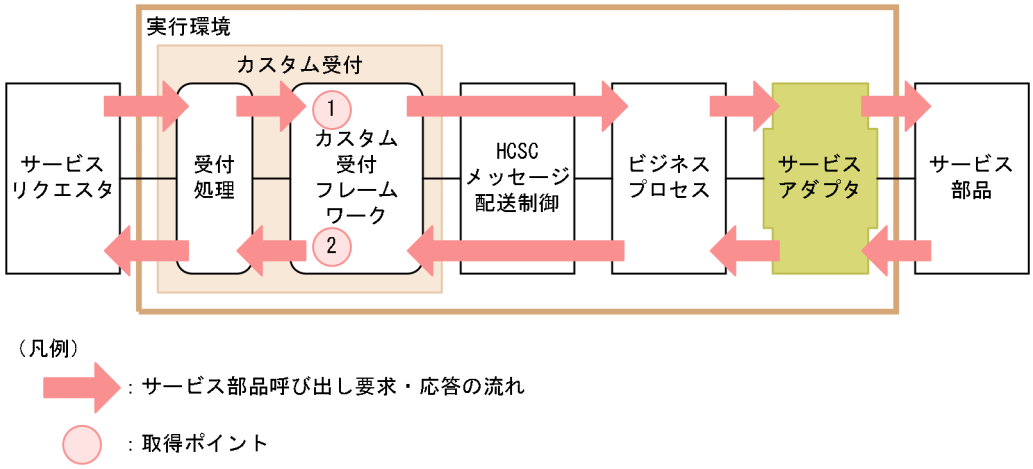
(2)　リクエストトレース（カスタム受付）

リクエストトレースは，リクエストの障害要因の解析に使用します。

(a)　リクエストトレースの取得ポイント

リクエストトレースの取得ポイントを次の図に示します。

図 A-12　リクエストトレースの取得ポイント



詳細位置，およびトレース取得ポイントを次の表に示します。なお，表の「図中の番号」は，図中の番号と対応しています。

表 A-6　リクエストトレースの取得ポイント

図中の番号	詳細位置	トレース取得ポイント
1	IN	カスタム受付フレームワークの入口
2	OUT	カスタム受付フレームワークの出口

(b) リクエストトレースの出力形式と出力内容

出力形式

リクエストトレースの出力形式については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「7.4.2　リクエストトレース」を参照してください。

出力内容

リクエストトレースに出力される内容を次の表に示します。

表 A-7　リクエストトレースに出力される内容

項目		内容
番号		トレースレコードの出力通番が出力されます。
日付		トレースレコードの取得日付（yyyy/mm/dd 形式）が出力されます。
時刻		トレースレコードの取得時刻（hh:mm:ss.SSS 形式）が，ローカル時刻でミリ秒単位に出力されます。
製品 ID		製品を識別するための ID が出力されます。 • CSCMSG：メッセージング基盤
pid		プロセスを識別するための ID が出力されます。
tid		スレッドを識別するための ID が出力されます。
ID		－
メッセージ共通 ID		リクエスト識別情報（親 ID）が出力されます。
サービスリクエスト ID		リクエスト識別情報（子 ID）が出力されます。
取得ポイント識別情報	取得位置	トレースの取得ポイント情報（取得位置）が出力されます。 • URCP：ユーザ定義受付
	プロトコル種別	トレースの取得ポイント情報（プロトコル種別）が出力されます。 • CUSTM：カスタム受付
	詳細位置	トレースの取得ポイント情報（詳細位置）が出力されます。 • IN：受付 • OUT：応答
	名称	－
	アダプタ種別	－
結果		応答結果種別が出力されます。ただし，詳細位置が IN の場合は，出力されません。 • NORMAL：正常終了 • ERROR：異常終了
付加情報		応答結果の付加情報が出力されます。 • 受付名 • 受付 ID

項目	内容
付加情報	• クライアント相関 ID • サービス名 • サービスオペレーション名
CRLF	レコード終端符号が出力されます。

（凡例）

－：何も出力されません。

(c)　リクエストトレースの取得方法と出力先

リクエストトレースの取得方法と出力先については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「7.4.2　リクエストトレース」を参照してください。

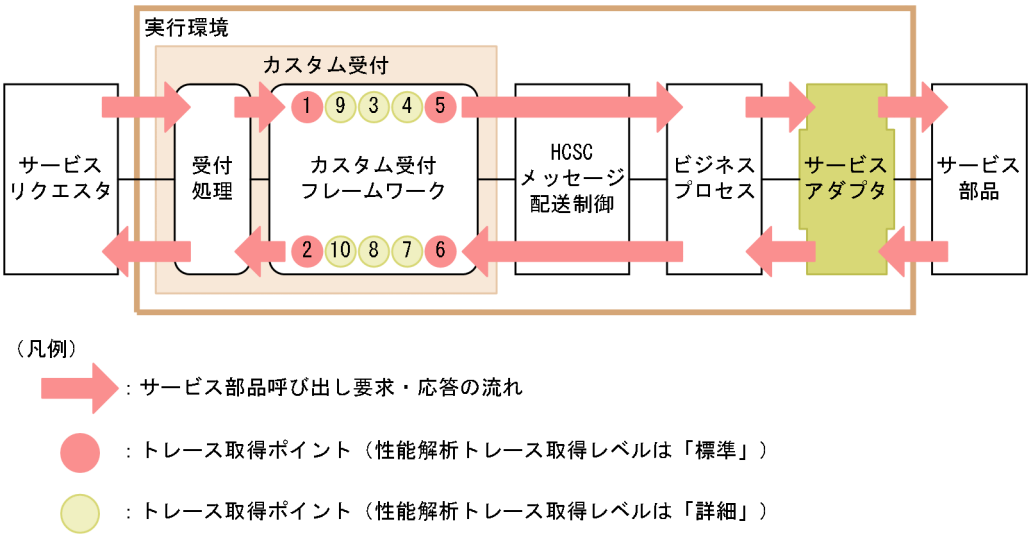
(3)　性能解析トレース（カスタム受付）

性能解析トレース（PRF トレース）は，サービスプラットフォームシステムの性能解析をするためのトレース情報で，それを CSV 形式で編集出力したテキストファイルが性能解析トレースファイルです。性能解析トレースは，J2EE アプリケーションを含めた，システム全体の性能ボトルネックを解析するための情報が出力されます。システムの性能ネックや性能トラブルシュートに使用します。性能解析トレース機能については，マニュアル「アプリケーションサーバ 機能解説 保守／移行編」の「7.　性能解析トレースを使用した性能解析」を参照してください。

(a)　性能解析トレースの取得ポイント

性能解析トレースの取得ポイントを次の図に示します。

図 A-13　性能解析トレースの取得ポイント



イベント ID，トレース取得ポイント，および性能解析トレース取得レベルを次の表に示します。なお，表の「図中の番号」は，図中の番号と対応しています。

表 A-8　性能解析トレースの取得ポイント

イベント ID	図中の番号	トレース取得ポイント	レベル
0x9860	1	カスタム受付フレームワークの入口	A
0x9861	2	カスタム受付フレームワークの出口	A
0x9862	3	データ変換（要求電文）の呼び出し口	B
0x9863	4	データ変換（要求電文）の応答受信口	B
0x9864	5	HCSC メッセージ配送制御の呼び出し口	A
0x9865	6	HCSC メッセージ配送制御の応答受信口	A
0x9866	7	データ変換（応答電文）の呼び出し口	B
0x9867	8	データ変換（応答電文）の応答受信口	B
0x986e※	9	カスタム受付フレームワークの JSON-XML 変換（要求電文）の呼び出し口	B
0x986f※	10	カスタム受付フレームワークの JSON-XML 変換（応答電文）の応答受信口	B

（凡例）

A：「標準」であることを示します。

B：「詳細」であることを示します。

注※

JSON-XML 変換をする場合だけ出力されます。

(b)　性能解析トレースの出力形式と出力内容

出力形式

性能解析トレースファイルに出力される形式は，J2EE サーバの性能解析トレースと同様です。性能解析トレースファイルの詳細は，マニュアル「アプリケーションサーバ 機能解説 保守／移行編」の「7.3 Management Server を利用した性能解析トレースファイルの収集」を参照してください。

出力内容

性能解析トレースファイルに出力される内容を次の表に示します。

表 A-9　性能解析トレースファイルに出力される内容

項目	内容
イベント ID	取得ポイントのイベント ID が出力されます。
リターンコード	取得ポイント種別が出力されます。 • 0：正常終了 • 1：異常終了
インターフェース名	クラス名が出力されます。 パッケージ名は表示されません。

項目	内容
オペレーション名	メソッド名が出力されます。
オプション情報	次のオプション情報が出力されます。 • リクエスト識別情報（親 ID） • リクエスト識別情報（子 ID） • 受付名 • 受付 ID • クライアント相関 ID • サービス名 • サービスオペレーション名 • 例外クラス名（異常終了時だけ）

(c)　性能解析トレースの取得方法と出力先

性能解析トレースファイルの取得方法および出力先は，アプリケーションサーバおよびサービスプラットフォーム全体で共通です。性能解析トレースファイルの取得方法および出力先については，マニュアル「アプリケーションサーバ 機能解説 保守／移行編」の「7.3　Management Server を利用した性能解析トレースファイルの収集」を参照してください。

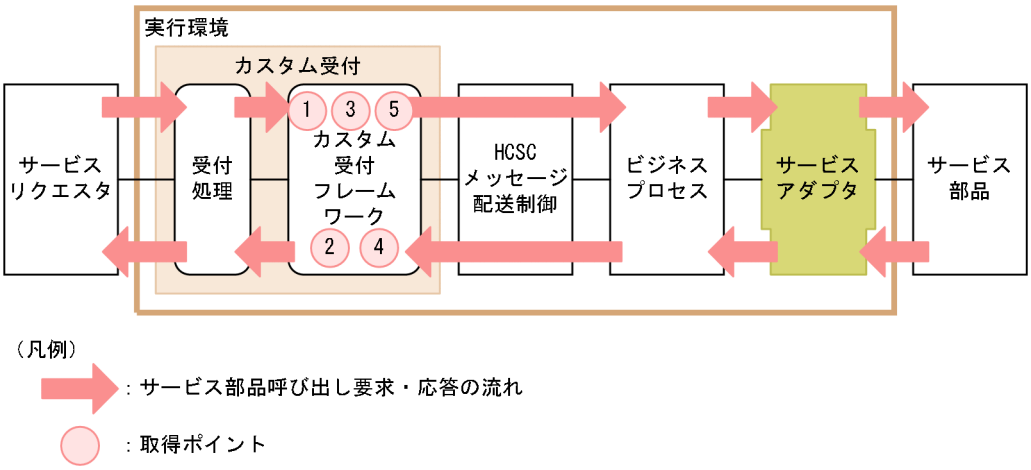
(4)　ユーザ電文トレース（カスタム受付）

ユーザ電文トレースは，電文の状態を確認するために使用します。

(a)　ユーザ電文トレースの取得ポイント

ユーザ電文トレースの取得ポイントを次の図に示します。

図 A-14　ユーザ電文トレースの取得ポイント



詳細位置，およびトレース取得ポイントを次の表に示します。なお，表の「図中の番号」は，図中の番号と対応しています。

表 A-10　ユーザ電文トレースの取得ポイント

図中の番号	詳細位置	トレース取得ポイント
1	IN	カスタム受付フレームワークの入口
2	OUT	カスタム受付フレームワークの出口
3	RQA※1	データ変換（要求電文）の応答口
4	RSB※1	データ変換（応答電文）の呼び出し口
5	ERR※2	カスタム受付フレームワークのサービス呼び出し

注※1

データ変換をする場合だけトレースを取得します。

注※2

サービス呼び出しでエラー（システム例外またはフォルト）を検出した場合だけ，要求電文のトレースを取得します。

(b)　ユーザ電文トレースの出力形式と出力内容

出力形式

ユーザ電文トレースの出力形式については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「7.4.4　ユーザ電文トレース」を参照してください。

出力内容

トレース開始時にユーザ電文トレースに出力される内容を次の表に示します。

表 A-11　ユーザ電文トレースに出力される内容（トレース開始）

項目		内容
番号		トレースレコードの出力通番が出力されます。
日付		トレースレコードの取得日付（yyyy/mm/dd 形式）が出力されます。
時刻		トレースレコードの取得時刻（hh:mm:ss.SSS 形式）が，ローカル時刻でミリ秒単位に出力されます。
製品 ID		製品を識別するための ID が出力されます。 • CSCMSG：メッセージング基盤
pid		プロセスを識別するための ID が出力されます。
tid		スレッドを識別するための ID が出力されます。
ID		−
電文トレース開始取得ポイント識別情報	電文トレース開始	ユーザ電文トレース開始を示す文字列「telegramtrace started」が出力されます。
	取得位置	トレースの取得ポイント情報（取得位置）が出力されます。 • URCP：ユーザ定義受付
	プロトコル種別	トレースの取得ポイント情報（プロトコル種別）が出力されます。

項目		内容
電文トレース開始取得ポイント識別情報	プロトコル種別	• CUSTM：カスタム受付
	詳細位置	トレースの取得ポイント情報（詳細位置）が出力されます。 • IN：受付 • OUT：応答 • RQA：カスタム受付での要求時のデータ変換後 • RSB：カスタム受付での応答時のデータ変換前
	メッセージ共通 ID	リクエスト識別情報（親 ID）が出力されます。
付加情報		次の付加情報が出力されます。 • リクエスト識別情報（子 ID）（Service Request ID） • 受付名（Reception Name） • 受付 ID（Reception ID） • クライアント相関 ID（Client ID） • サービス名（Service Name） • サービスオペレーション名（OP） • PRF ルートアプリケーション情報（RootApInfo）
CRLF		レコード終端符号が出力されます。

（凡例）

－：何も出力されません。

HCSC サーバランタイム定義ファイルに telegramtrace-format=FIXED を指定した場合のユーザ電文トレースデータに出力される内容を次の表に示します。

表 A-12　ユーザ電文トレースに出力される内容（telegramtrace-format=FIXED を指定した場合）

項目		内容
番号		トレースレコードの出力通番が出力されます。
日付		トレースレコードの取得日付（yyyy/mm/dd 形式）が出力されます。
時刻		トレースレコードの取得時刻（hh:mm:ss.SSS 形式）が，ローカル時刻でミリ秒単位に出力されます。
製品 ID		製品を識別するための ID が出力されます。 • CSCMSG：メッセージング基盤 • CSCBP：ビジネスプロセス
pid		プロセスを識別するための ID が出力されます。
tid		スレッドを識別するための ID が出力されます。
ID		－
電文トレースデータ情報	出力位置	ユーザ電文の先頭からのオフセット値（16 進数形式）が出力されます。
	データ （16 進数形式）	ユーザ電文の内容（16 進数形式）が出力されます。

項目		内容
電文トレースデータ情報	データ（ASCII 形式）	ユーザ電文の内容（ASCII 形式）が出力されます。 0x20～0x7E の範囲の場合に ASCII 文字が出力されます。 0x00～0x1F，0x7F～0xFF の範囲の場合は，ピリオド（.）が出力されます。
CRLF		レコード終端符号が出力されます。

（凡例）

－：何も出力されません。

注

SOAP メッセージの場合，ユーザ電文トレースデータにはユーザ電文に含まれる次の情報が出力されます。

・soap:Header 要素を含む複数の子要素

・soap:Body 要素の 1 つの子要素

なお，soap:Header 要素が存在しない場合は，soap:Body 要素の子要素だけが情報として出力されます。

HCSC サーバランタイム定義ファイルに telegramtrace-format=READABLE を指定した場合，16 進数形式データと UTF-8 形式データの出力内容が異なります。それぞれの出力形式を次の表に示します。

表 A-13　ユーザ電文トレースデータに出力される内容（telegramtrace-format=READABLE を指定した場合）（16 進数形式データ）

項目		内容
番号		トレースレコードの出力通番が表示されます。
日付		トレースレコードの取得日付（yyyy/mm/dd 形式）が出力されます。
時刻		トレースレコードの取得時刻（hh:mm:ss.SSS 形式）が，ローカル時刻でミリ秒単位に出力されます。
製品 ID		製品を特定するための識別子が出力されます。 • CSCMSG：メッセージング基盤 • CSCBP：ビジネスプロセス
pid		プロセスを識別するための ID が出力されます。
tid		スレッドを識別するための ID が出力されます。
ID		－
電文トレースデータ情報	出力位置（16 進数形式）	ユーザ電文の先頭からのオフセット値が 16 進数形式で出力されます。
	データ（16 進数形式）	ユーザ電文の内容が 16 進数形式で出力されます。
CRLF		レコード終端符号が出力されます。

（凡例）

－：何も出力されません。

注

SOAP メッセージの場合，ユーザ電文トレースデータにはユーザ電文に含まれる次の情報が出力されます。

・soap:Header 要素を含む複数の子要素

・soap:Body 要素の 1 つの子要素

soap:Header 要素が存在しない場合は，soap:Body 要素の子要素だけが情報として出力されます。

表 A-14　ユーザ電文トレースデータに出力される内容（telegramtrace-format=READABLE を指定した場合）（UTF-8 形式データ）

項目		内容
番号		トレースレコードの出力通番が表示されます。
日付		トレースレコードの取得日付（yyyy/mm/dd 形式）が出力されます。
時刻		トレースレコードの取得時刻（hh:mm:ss.SSS 形式）が，ローカル時刻でミリ秒単位に出力されます。
製品 ID		製品を特定するための識別子が出力されます。 • CSCMSG：メッセージング基盤 • CSCBP：ビジネスプロセス
pid		プロセスを識別するための ID が出力されます。
tid		スレッドを識別するための ID が出力されます。
ID		－
電文トレースデータ情報	データ（UTF-8 形式）	ユーザ電文を UTF-8 形式で出力します。 • 0x20～0x7E の範囲にある文字は，ASCII 文字として出力します。 • 次の範囲にある文字は，ピリオド（.）を出力します。 0x00～0x08 0x0B～0x0C 0x0E～0x1F 0x7F • 0x09 の場合は，水平タブを出力します。 • 改行コード（0x0a または 0x0d）は出力しません。 • 0xFE～0xFF の範囲にある文字は，クエスチョンマーク（?）を出力します。 • マルチバイト文字として正しく記述されている場合は，マルチバイト文字を出力します※。
CRLF		レコード終端符号が出力されます。

（凡例）

－：何も出力されません。

注 1

SOAP メッセージの場合，ユーザ電文トレースデータにはユーザ電文に含まれる次の情報が出力されます。

・soap:Header 要素を含む複数の子要素

・soap:Body 要素の 1 つの子要素

なお，soap:Header 要素が存在しない場合は，soap:Body 要素の子要素だけが情報として出力されます。

注 2

ユーザ電文トレースの 1 行の長さは最大 4,185 バイトです。それを超えた分は次の行に出力されます。4,185 バイト目のデータがマルチバイト文字の場合，マルチバイト文字は次の行に出力されます。

注※

マルチバイト文字の文字コードの割り当て範囲を次の表に示します。

表 A–15　UTF-8 のマルチバイト文字の割り当て範囲

区分	文字コードの範囲（16 進数）
2 バイト文字の開始バイト	0xC0～0xDF
3 バイト文字の開始バイト	0xE0～0xEF
4 バイト文字の開始バイト	0xF0～0xF7
5 バイト文字の開始バイト	0xF8～0xFB
6 バイト文字の開始バイト	0xFC～0xFD
マルチバイト文字の 2 バイト目以降	0x80～0xBF

なお，この範囲内でも出力するデータの並びが文字セットにとって不当な場合や，出力するデータの並びが正当でも有効な文字に対応付けできない場合は，クエスチョンマーク（?）を出力します。

トレース終了時にユーザ電文トレースに出力される内容を次の表に示します。

表 A–16　ユーザ電文トレースに出力される内容（トレース終了）

項目		内容
番号		トレースレコードの出力通番が出力されます。
日付		トレースレコードの取得日付（yyyy/mm/dd 形式）が出力されます。
時刻		トレースレコードの取得時刻（hh:mm:ss.SSS 形式）が，ローカル時刻でミリ秒単位に出力されます。
製品 ID		製品を識別するための ID が出力されます。 • CSCMSG：メッセージング基盤
pid		プロセスを識別するための ID が出力されます。
tid		スレッドを識別するための ID が出力されます。
ID		－
電文トレース終了取得ポイント識別情報	電文トレース終了	ユーザ電文トレース終了を示す文字列「telegramtrace ended」が出力されます。
	取得位置	トレースの取得ポイント情報（取得位置）が出力されます。 • URCP：ユーザ定義受付
	プロトコル種別	トレースの取得ポイント情報（プロトコル種別）が出力されます。 • CUSTM：カスタム受付
	詳細位置	トレースの取得ポイント情報（詳細位置）が出力されます。 • IN：受付 • OUT：応答 • RQA：カスタム受付での要求時のデータ変換後 • RSB：カスタム受付での応答時のデータ変換前

項目		内容
電文トレース終了取得ポイント識別情報	メッセージ共通 ID	リクエスト識別情報（親 ID）が出力されます。
付加情報		ユーザ電文の長さ（10 進数形式）が出力されます。ユーザ電文がない場合は「null」が出力されます。
CRLF		レコード終端符号が出力されます。

（凡例）

－：何も出力されません。

(c)　ユーザ電文トレースの取得方法と出力先

ユーザ電文トレースの取得方法と出力先については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「7.4.4　ユーザ電文トレース」を参照してください。

付録 B　カスタムアダプタ開発フレームワーク

カスタムアダプタ開発フレームワークを使用した汎用カスタムアダプタの開発について説明します。

付録 B.1　カスタムアダプタ開発フレームワークの API

ここでは，サービスプラットフォームが提供するカスタムアダプタ開発フレームワークの API について説明します。

カスタムアダプタ開発フレームワークで定義されているインターフェースおよび例外クラスは，次の場所に格納されています。

```
<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¥lib¥csc_adapter.jar
```

汎用カスタムアダプタをコンパイルするときは，この JAR ファイルをクラスパスに含めてください。

(1)　プロトコルコンバータインターフェース

(a)　CSCMsgCustomProtocolConverter インターフェース

説明

プロトコルコンバータを実装するためのインターフェースです。

CSCMsgCustomProtocolConverter のパッケージ名は，jp.co.Hitachi.soft.csc.msg.adapter.custom です。

形式

```
public interface CSCMsgCustomProtocolConverter
{
  public void start()
          throws CSCMsgCustomAdapterException;
  public void stop();
  public void setCustomAdapterContext(CSCMsgCustomAdapterContext adapterContext);
  public void invoke(CSCMsgRequestMessage requestMessage,
                     CSCMsgResponseMessage responseMessage)
          throws CSCMsgCustomAdapterException;
}
```

メソッド

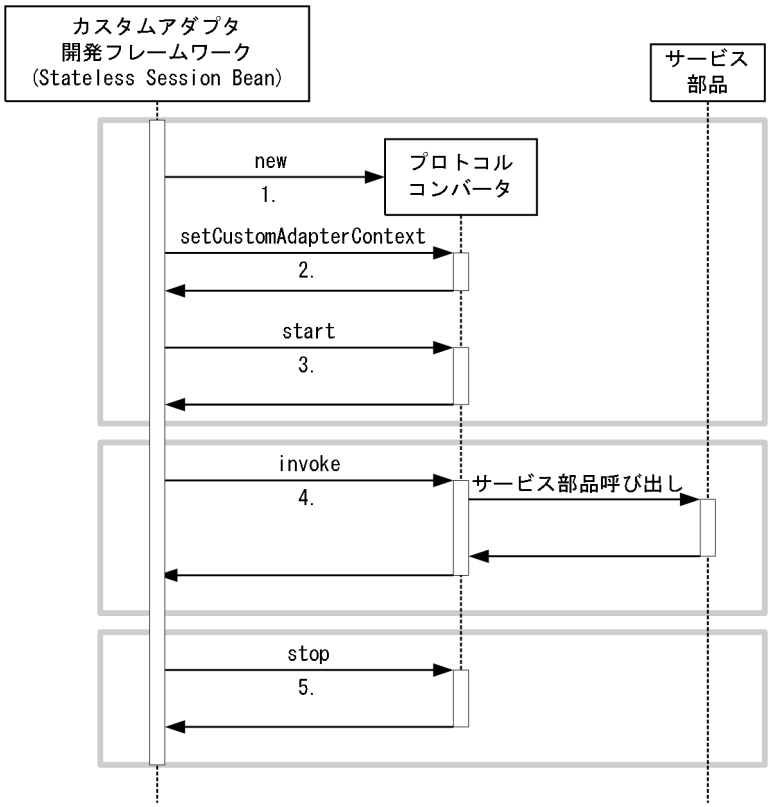
CSCMsgCustomProtocolConverter インターフェースのメソッドを次の表に示します。

メソッド名	説明
start メソッド	汎用カスタムアダプタが開始されるときに呼び出されるメソッドです。プロトコルコンバータ開始処理を実装します。 このメソッドは，一度だけ呼び出されます。

メソッド名	説明
stop メソッド	汎用カスタムアダプタが終了されるときに呼び出されるメソッドです。プロトコルコンバータ終了処理を実装します。 このメソッドは，一度だけ呼び出されます。
setCustomAdapterContext メソッド	アダプタコンテキストが設定されるときに呼び出されるメソッドです。与えられたアダプタコンテキストをプロトコルコンバータのインスタンスフィールドに格納する処理を記述します。 このメソッドは，一度だけ呼び出されます。
invoke メソッド	汎用カスタムアダプタを介してサービス部品を呼び出すときに呼び出されるメソッドです。サービス部品を呼び出す処理を記述します。 このメソッドは，サービスアダプタの呼び出し要求があるたびに呼び出されます。

カスタムアダプタ開発フレームワークからの各メソッドの呼び出し順序を次の図に示します。

図 B-1　カスタムアダプタ開発フレームワークからの各メソッドの呼び出し順序



1. コンストラクタ

プロトコルコンバータは，汎用カスタムアダプタ開始時に，カスタムアダプタ開発フレームワークから呼び出されたデフォルトコンストラクタによってインスタンス化されます。

2. setCustomAdapterContext メソッド

プロトコルコンバータクラスがインスタンス化された直後に，カスタムアダプタ開発フレームワークから呼ばれます。このメソッドは，汎用カスタムアダプタの開始時に一度だけ呼ばれます。引数として受け取ったアダプタコンテキストは，プロトコルコンバータのインスタンスフィールドに保持してください。

3. start メソッド

setServiceAdapterInfo メソッドの直後にカスタムアダプタ開発フレームワークから呼ばれます。このメソッドは，汎用カスタムアダプタの開始時に一度だけ呼ばれます。

4. invoke メソッド

サービス部品呼び出し処理時にカスタムアダプタ開発フレームワークから呼ばれます。このメソッドで，プロトコル変換とサービス呼び出し処理を行います。

5. stop メソッド

汎用カスタムアダプタが HCSC サーバから削除される直前にカスタムアダプタ開発フレームワークから呼ばれます。汎用カスタムアダプタの終了時処理を行います。このメソッドは，汎用カスタムアダプタの終了時に一度だけ呼ばれます。

注意事項

- public なデフォルトコンストラクタ

  カスタムアダプタ開発フレームワークは，このインターフェースを実装したクラスをインスタンス化するときに，デフォルトコンストラクタを使用します。そのため，このインターフェースを実装したクラスでは，public なデフォルトコンストラクタを実装する必要があります。

- マルチスレッド

  プロトコルコンバータは，複数の汎用カスタムアダプタのインスタンスから共有されるため，スレッドセーフなクラスを実装する必要があります。

- 例外の送出

  CSCMsgCustomProtocolConverter の各メソッドでは，指定された例外だけ送出できます。java.lang.RuntimeException およびその派生クラスは送出しないでください。送出した場合，動作は保証されません。

●start メソッド

説明

汎用カスタムアダプタが開始されるときに呼び出されるメソッドです。

汎用カスタムアダプタのライフサイクルを通じて使用するリソースは，このメソッドで確保することを推奨します。

形式

```
public void start()
  throws CSCMsgCustomAdapterException;
```

パラメタ

なし。

例外

jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgCustomAdapterException：

start メソッドの実行に失敗した場合に発生させます。

戻り値

なし。

●stop メソッド

説明

汎用カスタムアダプタ終了時に呼び出されるメソッドです。汎用カスタムアダプタの終了処理を記述します。

形式

```
public void stop();
```

パラメタ

なし。

例外

なし。

戻り値

なし。

●setCustomAdapterContext メソッド

説明

アダプタコンテキストが設定されるときに呼び出されるメソッドです。

このメソッドのあとに必ず start メソッドを呼び出します。

このメソッドで受け取ったアダプタコンテキストインターフェースは，汎用カスタムアダプタのライフサイクルを通じて保持する必要があります。

形式

```
public void setCustomAdapterContext(CSCMsgCustomAdapterContext adapterContext);
```

パラメタ

adapterContext：

CSCMsgCustomAdapterContext のリファレンスが渡されます。

例外

なし。

戻り値

なし。

●invoke メソッド

説明

汎用カスタムアダプタを経由してサービス部品を呼び出すときに，呼び出されるメソッドです。サービス部品を呼び出す処理を記述します。

requestMessage に格納されている要求電文を基にサービス部品を呼び出します。

通信モデルが同期の場合，応答電文は responseMessage を格納します。

通信モデルが非同期の場合，responseMessage には null が渡されます。

形式

```
public void invoke (CSCMsgRequestMessage requestMessage,
                    CSCMsgResponseMessage responseMessage)
  throws CSCMsgCustomAdapterException;
```

パラメタ

- requestMessage：

  要求電文が格納された CSCMsgRequestMessage のリファレンスが渡されます。

- responseMessage：

  通信モデルが同期の場合，応答電文を格納する CSCMsgResponseMessage のリファレンスが渡されます。通信モデルが非同期通信の場合，null が渡されます。

例外

jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgCustomAdapterException：

invoke メソッドの実行に失敗した場合に発生させます。

戻り値

なし。

(2) アダプタコンテキストインターフェース

(a) CSCMsgCustomAdapterContext インターフェース

説明

プロトコルコンバータがアダプタ情報を取得するためのインターフェースです。CSCMsgCustomAdapterContext のパッケージ名は，jp.co.Hitachi.soft.csc.msg.adapter.custom です。

アダプタコンテキストインターフェースは，汎用カスタムアダプタ開始処理の直前に，setCustomAdapterContext メソッドの引数として入力されます。プロトコルコンバータでは，この引数をインスタンスフィールドとして保持し，必要な情報を取得します。

形式

```
public interface CSCMsgCustomAdapterContext
{
  public String getAdapterName();
  public Properties getProperties();
  public byte[] getResourceAsBytes(String fileName)
    throws CSCMsgResourceAccessException;
  public java.io.InputStream getResourceAsStream(String fileName)
    throws CSCMsgResourceAccessException;
}
```

メソッド

CSCMsgCustomAdapterContext インターフェースのメソッドを次の表に示します。

メソッド名	説明
getAdapterName メソッド	汎用カスタムアダプタの名称を取得します。
getProperties メソッド	カスタムアダプタプロパティファイルの内容を Properties として取得します。
getResourceAsBytes メソッド	独自定義ファイルの内容をバイナリ形式で取得します。
getResourceAsStream メソッド	独自定義ファイルにアクセスするためのストリームを取得します。

●getAdapterName メソッド

説明

汎用カスタムアダプタの名称を取得します。

形式

```
public String getAdapterName();
```

パラメタ

なし。

例外

なし。

戻り値

汎用カスタムアダプタの名称を返却します。

●getProperties メソッド

説明

カスタムアダプタプロパティファイルの内容を Properties として取得します。

形式

```
public Properties getProperties();
```

パラメタ

なし。

例外

なし。

戻り値

プロパティファイルの内容を返却します。

●getResourceAsBytes メソッド

説明

次の優先順位によって，独自定義ファイルの内容をバイナリ形式で取得します。

表 B-1　独自定義ファイルの優先順位

優先順位	独自定義ファイルの種類
1	<HCSC サーバランタイム定義ファイルの definition-file-base-path プロパティで指定したディレクトリ>¥<HCSC サーバ名>¥<サービス ID>配下にある独自定義ファイル
2	csccmctl コマンド（-operation set）で運用環境に設定した独自定義ファイル
3	開発環境で定義した独自定義ファイル

形式

```
public byte[] getResourceAsBytes(String fileName)
  throws CSCMsgResourceAccessException;
```

パラメタ

fileName：

アクセスしたい独自定義ファイル名を指定します。

例外

CSCMsgResourceAccessException：

fileName で指定したリソースが見つからなかった，または java.io.IOException が発生しました。

戻り値

独自定義ファイルの内容をバイナリ形式で返却します。

●getResourceAsStream メソッド

説明

次の優先順位によって，独自定義ファイルにアクセスするためのストリームを取得します。

表 B-2　独自定義ファイルの優先順位

優先順位	独自定義ファイルの種類
1	<HCSC サーバランタイム定義ファイルの definition-file-base-path プロパティで指定したディレクトリ>¥<HCSC サーバ名>¥<サービス ID>配下にある独自定義ファイル
2	csccmctl コマンド（-operation set）で運用環境に設定した独自定義ファイル
3	開発環境で定義した独自定義ファイル

形式

```
public java.io.InputStream getResourceAsStream(String fileName)
  throws CSCMsgResourceAccessException;
```

パラメタ

fileName：

アクセスしたい独自定義ファイル名を指定します。

例外

CSCMsgResourceAccessException：

fileName で指定したリソースが見つからなかった，または java.io.IOException が発生しました。

戻り値

独自定義ファイルの内容をストリーム形式で返却します。

注意事項

このメソッドで取得したストリームを static なフィールドに割り当てないでください。メモリリークが発生するおそれがあります。

(3) メッセージインターフェース

HCSC サーバとプロトコルコンバータの間でメッセージを交換するためのインターフェースの一覧を次の表に示します。

表 B-3 メッセージ操作に使用する API の一覧

インターフェース名	説明
CSCMsgMessageConstant インターフェース	メッセージの操作に必要な定数を定義します。
CSCMsgRequestMessage インターフェース	サービス部品呼び出しに必要な情報（オペレーション名，電文など）を取得するためのメソッドを提供します。
CSCMsgResponseMessage インターフェース	サービス部品呼び出し結果（電文またはフォルト情報）を格納するためのメソッドを提供します。

(a) CSCMsgMessageConstant インターフェース

説明

プロトコルコンバータがメッセージの操作に必要な定数を定義するインターフェースです。

CSCMsgMessageConstant のパッケージ名は，jp.co.Hitachi.soft.csc.msg.adapter.custom です。

プロトコルコンバータで要求メッセージに格納されている電文の種類，または応答メッセージに格納する電文の種類を確認するためにこの定数を使用します。

形式

```
public interface CSCMsgMessageConstant
{
  public static final int MESSAGE_TYPE_NONE;
  public static final int MESSAGE_TYPE_XML;
  public static final int MESSAGE_TYPE_BINARY;
  public static final int MESSAGE_TYPE_ANY;
}
```

メンバ属性

MESSAGE_TYPE_NONE：

電文が格納されていないことを示します。

MESSAGE_TYPE_XML：

電文の形式が XML 形式(DOM 形式)であることを示します。

MESSAGE_TYPE_BINARY：

電文の形式がバイナリであることを示します。

MESSAGE_TYPE_ANY：

電文の形式が任意形式であることを示します。

(b) CSCMsgRequestMessage インターフェース

説明

サービス部品の呼び出しに必要な情報（オペレーション名，電文など）を取得するためのメソッドを提供するインターフェースです。

CSCMsgRequestMessage のパッケージ名は，jp.co.Hitachi.soft.csc.msg.adapter.custom です。

形式

```
public interface CSCMsgRequestMessage
{
  public byte[] getBytes()
          throws CSCMsgIllegalMessageTypeException,
                 CSCMsgInvalidMessageException;
  public Map getMessageContext();
  public int getMessageType();
  public String getOperationName();
  public org.w3c.dom.Document getXMLDocument()
          throws CSCMsgIllegalMessageTypeException,
                 CSCMsgInvalidMessageException;
  public byte[] getXMLBytes()
          throws CSCMsgIllegalMessageTypeException,
                 CSCMsgInvalidMessageException;
}
```

メソッド

CSCMsgRequestMessage インターフェースのメソッドを次の表に示します。

メソッド名	説明
getBytes メソッド	バイナリ形式の電文を取得します。
getMessageContext メソッド	ビジネスプロセス経由で汎用カスタムアダプタを呼び出した場合に経由したビジネスプロセスに関する情報を返却します。
getMessageType メソッド	要求メッセージに格納されている電文の種別を取得します。
getOperationName メソッド	サービス部品呼び出し時に呼び出すオペレーションの名称を取得します。
getXMLDocument メソッド	XML 形式の電文を取得します。

メソッド名	説明
getXMLBytes メソッド	XML 形式の電文をバイト配列で取得します。

●getBytes メソッド

説明

バイナリ形式の電文を取得します。

形式

```
public byte[] getBytes()
  throws CSCMsgIllegalMessageTypeException,
         CSCMsgInvalidMessageException;
```

パラメタ

なし。

例外

- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgIllegalMessageTypeException：
  電文の形式がバイナリではありません。または，電文が格納されている状態です。電文の種別が XML 形式の場合は，このメソッドを使用して電文を取得することはできません。
- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgInvalidMessageException：
  格納されている電文が不正です。

戻り値

設定されている電文の種別がバイナリ形式（MESSAGE_TYPE_BINARY）の場合，バイト配列で応答を返します。

電文が格納されていない（MESSAGE_TYPE_NONE）場合，0 バイトのバイト配列（長さ 0 の配列）を返します。

注意事項

- このメソッドで取得したバイナリ電文（バイト配列）については，変更しても問題ありません。
- このメソッドを実行したときに，オブジェクトのコピーを作成します。そのため，再度このメソッドを発行すると，変更前のオブジェクトを再取得することになるので注意してください。

●getMessageContext メソッド

説明

ビジネスプロセス経由で汎用カスタムアダプタを呼び出した場合は経由したビジネスプロセスに関する情報を返却します。

返却される情報の形式は java.util.Map です。

返却される Map は，Collections#unmodifiableMap()が返したオブジェクトです。

そのため，変更できません。

また，返却される Map は同期化されないため複数スレッドで共有する場合は排他処理が必要です。

なお，ビジネスプロセスを経由しない呼び出しの場合は次の表の HCSC サーバ名から J2EE サーバ名までの情報だけが返却される Map に含まれます。

表 B-4　取得情報一覧

情報種別	Map の内容			説明
	要素		値	
	要素名	要素の型	値の型	
ビジネスプロセス名	CSCMsgCustomAdapterConstant.ContextType.BP_DEFINITION_NAME	enum	java.lang.String	汎用カスタムアダプタの呼び出し元のビジネスプロセス名※1
ビジネスプロセスのバージョン	CSCMsgCustomAdapterConstant.ContextType.BP_VERSION	enum	java.lang.String	汎用カスタムアダプタの呼び出し元のビジネスプロセスのバージョン※1
サービス呼出アクティビティ名	CSCMsgCustomAdapterConstant.ContextType.BP_INVOKEACTIVITY_NAME	enum	java.lang.String	汎用カスタムアダプタの呼び出し元のビジネスプロセスのサービス呼出アクティビティ名※1
汎用カスタムアダプタのサービス名	CSCMsgCustomAdapterConstant.ContextType.BP_INVOKEACTIVITY_CUSTOMADAPTERNAME	enum	java.lang.String	サービス呼出アクティビティから呼び出される汎用カスタムアダプタのサービス名※1
オペレーション名	CSCMsgCustomAdapterConstant.ContextType.BP_INVOKEACTIVITY_OPERATIONNAME	enum	java.lang.String	サービス呼出アクティビティのオペレーション名※1※2
プロセスインスタンス ID	CSCMsgCustomAdapterConstant.ContextType.BP_PROCESSINSTANCEID	enum	java.lang.String	プロセスインスタンス ID※1
HCSC サーバ名	CSCMsgCustomAdapterConstant.ContextType.SERVERNAME_HCSC	enum	java.lang.String	HCSC サーバ名
クラスタ名	CSCMsgCustomAdapterConstant.ContextType.SERVERNAME_CLUSTER	enum	java.lang.String	クラスタ名
J2EE サーバ名	CSCMsgCustomAdapterConstant.ContextType.SERVERNAME_J2EE	enum	java.lang.String	J2EE サーバ名

注※1

ここで取得できる情報は該当の汎用カスタムアダプタを呼び出したビジネスプロセスの情報です。

BP1 → BP2 → 汎用カスタムアダプタのように，ビジネスプロセスからビジネスプロセスを呼び出す場合は BP1 の情報は取得できません。

注※2

サービス呼出アクティビティから呼び出される汎用カスタムアダプタに複数のオペレーションが定義されている場合は，そのリクエスト処理で呼び出されるオペレーションだけ取得できます。

形式

```
public Map getMessageContext();
```

パラメタ

なし。

例外

なし。

戻り値

java.util.Map

●getMessageType メソッド

説明

要求メッセージに格納されている電文の種別を取得します。

形式

```
public int getMessageType();
```

パラメタ

なし。

例外

なし。

戻り値

要求メッセージに格納されている電文の種別を返却します。返却される値については，「付録 B.1(3)(a)　CSCMsgMessageConstant インターフェース」を参照してください。ただし，任意形式の汎用カスタムアダプタを作成した場合は，電文の種別に関係なく，「MESSAGE_TYPE_ANY」を返却します。

●getOperationName メソッド

説明

サービス部品呼び出し時に呼び出すオペレーションの名称を取得します。

形式

```
public String getOperationName();
```

パラメタ

なし。

例外

なし。

戻り値

呼び出すオペレーションの名称を返却します。

●getXMLDocument メソッド

説明

XML 形式の電文を取得します。

形式

```
public org.w3c.dom.Document getXMLDocument()
  throws CSCMsgIllegalMessageTypeException,
         CSCMsgInvalidMessageException;
```

パラメタ

なし。

例外

- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgIllegalMessageTypeException：
  電文の形式が XML 形式ではありません。または，電文が格納されている状態です。電文の種別がバイナリ形式の場合は，このメソッドを使用して電文を取得することはできません。
- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgInvalidMessageException：
  格納されている電文が不正です。

戻り値

設定されている電文の種別が XML 形式（MESSAGE_TYPE_XML）の場合，DOM 形式のオブジェクトで応答を返します。

電文が格納されていない（MESSAGE_TYPE_NONE）場合，null を返します。

注意事項

- このメソッドで取得した XML 電文（DOM）については，変更しても問題ありません。
- このメソッドを実行したときに，オブジェクトのコピーを作成します。そのため，再度このメソッドを発行すると，変更前のオブジェクトを再取得することになるので注意してください。
- 任意形式の汎用カスタムアダプタを作成した場合，このメソッドを実行するとエラー（KDEC03016-E）になり，電文を取得できません。この場合は，getBytes メソッドを使用してください。

●getXMLBytes メソッド

説明

XML 形式の電文をバイト配列で取得します。

StAX などを利用して電文を読み込む場合，このメソッドを利用することでメモリ使用量を削減できます。

形式

```
public byte[] getXMLBytes()
  throws CSCMsgIllegalMessageTypeException,
         CSCMsgInvalidMessageException;
```

パラメタ

なし。

例外

- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgIllegalMessageTypeException：
  電文の形式が XML 形式ではありません。
  電文の種別がバイナリ形式の場合は，このメソッドを使用して電文を取得することはできません。
- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgInvalidMessageException：
  格納されている電文が不正です。

戻り値

設定されている電文の種別が XML 形式（MESSAGE_TYPE_XML）の場合，バイト配列形式のオブジェクトで応答を返します。

電文が格納されていない（MESSAGE_TYPE_NONE）場合，null を返します。

注意事項

- このメソッドで取得した XML 電文（バイト配列）については，変更しても問題ありません。
- このメソッドを実行したときに，オブジェクトのコピーを作成します。そのため，再度このメソッドを発行すると，変更前のオブジェクトを再取得することになるので注意してください。
- XML Processor の StAX を利用して XML 電文を読み込む場合，次の点に注意してください。
  ・XML バージョン 1.1 の電文を設定した場合，例外が発生します。
  ・ISO-10646-UCS-4 エンコーディングを使用した場合，例外が発生します。
  ・XML 文書に「:」で始まる要素名が含まれる場合，例外になりません。
  ・XML 文書中の文字データの中に，CDATA セクション終了区切り子（"]]>"）が単独で含まれる場合，例外になりません。
  ・空要素の場合，開始タグと終了タグだけが出力されます。
  （例）<sample></sample>

(c) CSCMsgResponseMessage インターフェース

説明

サービス部品呼び出し結果（電文またはフォルト情報）を格納するためのメソッドを提供するインターフェースです。

CSCMsgResponseMessage のパッケージ名は，jp.co.Hitachi.soft.csc.msg.adapter.custom です。

形式

```
public interface CSCMsgResponseMessage
{
  public int getMessageType();
  public void setBytes(byte[] message)
          throws CSCMsgIllegalMessageTypeException,
                 CSCMsgMultipleMessageInsertionException,
                 CSCMsgInvalidMessageException;
```

```
  public void setFault(String faultCode,
                       String faultString,
                       String faultActor,
                       byte[] faultDetail)
          throws CSCMsgMultipleMessageInsertionException;
  public void setFault(String faultCode,
                       String faultString,
                       String faultActor,
                       org.w3c.dom.Document faultDetail)
          throws CSCMsgMultipleMessageInsertionException,
                 CSCMsgInvalidMessageException;
  public void setXMLDocument(org.w3c.dom.Document dom)
          throws CSCMsgIllegalMessageTypeException,
                 CSCMsgMultipleMessageInsertionException,
                 CSCMsgInvalidMessageException;
  public void setXMLBytes(byte[] byte)
          throws CSCMsgIllegalMessageTypeException,
                 CSCMsgMultipleMessageInsertionException,
                 CSCMsgInvalidMessageException;
}
```

メソッド

CSCMsgResponseMessage インターフェースのメソッドを次の表に示します。

メソッド名	説明
getMessageType メソッド	応答メッセージに格納する電文の種別を取得します。
setBytes メソッド	応答メッセージにバイナリ形式の電文を格納します。
setFault メソッド（形式 1）	サービス部品を呼び出すときに発生したフォルト情報を格納します。
setFault メソッド（形式 2）	サービス部品を呼び出すときに発生したフォルト情報を格納します。
setXMLDocument メソッド	応答メッセージに XML 形式の電文を格納します。
setXMLBytes メソッド	応答メッセージに，XML 形式の電文をバイト配列で格納します。

●getMessageType メソッド

説明

応答メッセージに格納する電文の種別を取得します。

形式

```
public int getMessageType();
```

パラメタ

なし。

例外

なし。

戻り値

応答メッセージに格納する電文の種別を返却します。ただし，任意形式の汎用カスタムアダプタを作成した場合は，電文の種別に関係なく，「MESSAGE_TYPE_ANY」を返却します。

- MESSAGE_TYPE_NONE：
  電文を格納しません。
- MESSAGE_TYPE_XML：
  電文の種別が XML 形式の電文を格納します。
- MESSAGE_TYPE_BINARY：
  電文の種別がバイナリ形式の電文を格納します。

●setBytes メソッド

説明

応答メッセージにバイナリ形式の電文を格納します。

形式

```
public void setBytes(byte[] message)
        throws CSCMsgIllegalMessageTypeException,
               CSCMsgMultipleMessageInsertionException,
               CSCMsgInvalidMessageException;
```

パラメタ

message：

バイナリ形式の電文を指定します。

なお，バイナリ形式の電文を格納した場合は，電文の種別をバイナリ形式（MESSAGE_TYPE_BINARY）に設定し，長さ 0 のバイト列や null を格納した場合は，電文を格納しない種別（MESSAGE_TYPE_NONE）に設定します。

例外

- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgIllegalMessageTypeException：
  設定されている電文の種別がバイナリではありません。または電文が格納されていない状態ではありません。要求時の電文の形式が XML の場合に，応答時の電文の形式をバイナリに変えることはできません。
- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgMultipleMessageInsertionException：
  すでに電文またはフォルト情報が設定されています。
- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgInvalidMessageException：
  指定したバイナリ形式に誤りがあります。

戻り値

なし。

注意事項

任意形式の汎用カスタムアダプタを作成した場合は，電文の形式に関係なく，このメソッドを使用してください。格納した電文の種別を任意形式（MESSAGE_TYPE_ANY）に設定します。長さ 0 のバイト列や null を格納した場合でも，任意形式（MESSAGE_TYPE_ANY）に設定します。

●setFault メソッド（形式 1）

説明

サービス部品を呼び出したときに発生したフォルト情報を格納します。

faultDetail の内容をバイナリ形式で格納する場合に使用します。

なお，応答時の電文は存在しないため，電文を格納しない種別（MESSAGE_TYPE_NONE）を設定します。

形式

```
public void setFault(String faultCode,
                     String faultString,
                     String faultActor,
                     byte[] faultDetail)
        throws CSCMsgMultipleMessageInsertionException;
```

パラメタ

- faultCode：

  FaultCode 情報を指定します。

  null を指定した場合，"Server.ServiceExecutionError"が格納されます。

- faultString：

  FaultString 情報を指定します。

  null を指定した場合，"Service Excution Error at CustomAdapter"が格納されます。

- faultActor：

  FaultActor 情報を指定します。

  null を指定した場合，HCSC-Definer で設定したサービス ID が格納されます。

- faultDetail：

  XML 形式の Detail 情報をバイト配列で指定します。

  長さ 0 のバイト列を指定した場合，そのまま長さ 0 のバイト列が格納されます。null を指定した場合，そのまま null が格納されます。

例外

jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgMultipleMessageInsertionException：

すでに応答電文またはフォルト情報が設定されています。

戻り値

なし。

注意事項

- faultDetail は，XML 文書をバイナリ化したものです。必ず，先頭に XML 宣言をしてください。宣言をしなかった場合は，エラーにはなりませんが，正常に動作しないおそれがあります。
- XML 文書をバイト配列に変換する場合は，文字コードとして UTF-8 を使用してください。指定した文字コード以外では，正常に動作しないおそれがあります。

●setFault メソッド（形式 2）

説明

サービス部品を呼び出したときに発生したフォルト情報を格納します。

faultDetail の内容を XML 形式（DOM 形式）で格納する場合に使用します。

なお，応答時の電文は存在しないため，電文を格納しない種別（MESSAGE_TYPE_NONE）を設定します。

形式

```
public void setFault(String faultCode,
                     String faultString,
                     String faultActor,
                     org.w3c.dom.Document faultDetail)
        throws CSCMsgMultipleMessageInsertionException,
               CSCMsgInvalidMessageException;
```

パラメタ

- faultCode：

  FaultCode 情報を指定します。

  null を指定した場合，"Server.ServiceExecutionError"が格納されます。

- faultString：

  FaultString 情報を指定します。

  null を指定した場合，"Service Excution Error at CustomAdapter"が格納されます。

- faultActor：

  FaultActor 情報を指定します。

  null を指定した場合，HCSC-Definer で設定したサービス ID が格納されます。

- faultDetail：

  Detail 情報を DOM 形式で指定します。

  null を指定した場合，そのまま null が格納されます。

例外

- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgMultipleMessageInsertionException：

  すでに応答電文またはフォルト情報が設定されています。

- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgInvalidMessageException：

  指定した DOM 形式に誤りがあります。

戻り値

なし。

●setXMLDocument メソッド

説明

応答メッセージに XML 形式（DOM 形式）の電文を格納します。

形式

```
public void setXMLDocument(org.w3c.dom.Document dom)
        throws CSCMsgIllegalMessageTypeException,
               CSCMsgMultipleMessageInsertionException,
               CSCMsgInvalidMessageException;
```

パラメタ

dom：

DOM 形式の電文を指定します。

なお，XML 形式の電文を格納した場合は，電文の種別を XML（MESSAGE_TYPE_XML）に設定し，null を格納した場合は，電文を格納しない種別（MESSAGE_TYPE_NONE）に設定します。

例外

- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgIllegalMessageTypeException：

  設定されている電文の種別が XML ではありません。または電文が格納されていない状態ではありません。要求時の電文の形式が XML の場合に，応答時の電文の形式をバイナリに変えることはできません。

- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgMultipleMessageInsertionException：

  すでに電文またはフォルト情報が設定されています。

- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgInvalidMessageException：

  指定した DOM 形式に誤りがあります。

戻り値

なし。

注意事項

任意形式の汎用カスタムアダプタを作成した場合，このメソッドを実行するとエラー（KDEC03016-E）になり，電文を格納できません。この場合は，setBytes メソッドを使用してください。

●setXMLBytes メソッド

説明

応答メッセージに，XML 形式(バイト配列)の電文を格納します。

汎用カスタムアダプタで XML 形式の電文をバイト配列で扱っていた場合，DOM 形式に変換する必要がないため，このメソッドを利用することでメモリ使用量を削減できます。

形式

```
public void setXMLBytes(byte[] byte)
        throws CSCMsgIllegalMessageTypeException,
               CSCMsgMultipleMessageInsertionException,
               CSCMsgInvalidMessageException;
```

パラメタ

byte：

バイト配列の電文を指定します。

なお，byte が null ではない場合は，電文の種別を XML（MESSAGE_TYPE_XML）に設定し，null である場合は，電文を格納しない種別（MESSAGE_TYPE_NONE）に設定します。

例外

- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgIllegalMessageTypeException：
  要求電文の種別が XML ではありません。要求電文の種別がバイナリの場合に，応答電文の種別を XML に変えることはできません。
- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgMultipleMessageInsertionException：
  すでに電文またはフォルト情報が設定されています。
- jp.co.Hitachi.soft.csc.msg.adapter.custom.CSCMsgInvalidMessageException：
  指定した XML 形式に誤りがあります。

戻り値

なし。

注意事項

- 電文の作成時に StAX を使用する場合，次の点に注意してください。
  ・XML バージョン 1.1 の電文を設定した場合，例外が発生します。
  ・ISO-10646-UCS-4 エンコーディングを使用した場合，例外が発生します。
  ・XML 文書に「:」で始まる要素名が含まれる場合，例外になりません。
  ・XML 文書中の文字データの中に，CDATA セクション終了区切り子（"]]>"）が単独で含まれる場合，例外になりません。
  ・空要素の場合，開始タグと終了タグだけが出力されます。
  （例）<sample></sample>

(d) CSCMsgCustomAdapterConstant クラス

説明

CSCMsgRequestMessage インターフェースの getMessageContext メソッドで取得できる，ビジネスプロセス情報を格納した Map のキーを定義するインターフェースです。

CSCMsgCustomAdapterConstant のパッケージ名は，jp.co.Hitachi.soft.csc.msg.adapter.custom です。

形式

```
public interface CSCMsgCustomAdapterConstant {
  public enum ContextType {
            BP_DEFINITION_NAME,
            BP_VERSION,
            BP_INVOKEACTIVITY_NAME,
            BP_INVOKEACTIVITY_CUSTOMADAPTERNAME,
            BP_INVOKEACTIVITY_OPERATIONNAME,
            BP_PROCESSINSTANCEID,
            SERVERNAME_HCSC,
            SERVERNAME_CLUSTER,
```

```
            SERVERNAME_J2EE;
    }
}
```

Enum 定数

BP_DEFINITION_NAME：

汎用カスタムアダプタの呼び出し元のビジネスプロセス名

BP_VERSION：

汎用カスタムアダプタの呼び出し元のビジネスプロセスのバージョン

BP_INVOKEACTIVITY_NAME：

汎用カスタムアダプタの呼び出し元のビジネスプロセスのサービス呼出アクティビティ名

BP_INVOKEACTIVITY_CUSTOMADAPTERNAME：

サービス呼出アクティビティから呼び出される汎用カスタムアダプタのサービス名

BP_INVOKEACTIVITY_OPERATIONNAME：

サービス呼出アクティビティのオペレーション名

BP_PROCESSINSTANCEID：

プロセスインスタンス ID

SERVERNAME_HCSC：

HCSC サーバ名

SERVERNAME_CLUSTER：

クラスタ名

SERVERNAME_J2EE：

J2EE サーバ名

(4) 例外クラス

プロトコルコンバータの開発時に発生する例外クラスの一覧を次の表に示します。

表 B-5 プロトコルコンバータの例外クラス一覧

クラス名	説明
CSCMsgCustomAdapterException クラス	汎用カスタムアダプタの初期化処理やサービス部品を呼び出す処理で例外が発生した場合に送出する例外です。
CSCMsgIllegalMessageTypeException クラス	電文の取得または格納時に，電文タイプと異なるタイプの電文形式のメソッドが呼ばれた場合に送出する例外です。
CSCMsgMultipleMessageInsertionException クラス	すでに電文またはフォルト情報が格納されれた場合に送出する例外です。 電文またはフォルト情報を格納したあとに，再び電文またはフォルト情報を設定することはできません。

クラス名	説明
CSCMsgInvalidMessageException クラス	要求電文の場合，格納さてている電文の種別（形式）が不正なときに送出する例外です。 応答電文の場合，不正な形式の電文種別を指定しているときに送出する例外です。
CSCMsgResourceAccessException クラス	リソースアクセス処理中にエラーが発生した場合に送出する例外です。 次のどちらかの場合に発生します。 • 取得しようとしたリソースがない場合 • 入出力例外（java.io.IOException）が発生した場合

付録 B.2　カスタムアダプタ開発フレームワークの定義ファイル

ここでは，カスタムアダプタ開発フレームワークで汎用カスタムアダプタを開発するときに定義するファイルの内容について説明します。

(1)　カスタムアダプタ開発フレームワーク動作定義ファイル

カスタムアダプタ開発フレームワーク動作定義ファイルは，java.util.Properties クラスの loadFromXML メソッドで読み込むことができる XML 形式のプロパティファイルです。カスタムアダプタ開発フレームワーク動作定義ファイルのファイル名は必ず「framework_properties.xml」にしてください。

機能

カスタムアダプタ開発フレームワークが読み込むプロトコルコンバータのクラス名（パッケージ名を含む）を指定します。プロトコルコンバータのクラス名は，key 属性が classname である entry タグに記述します。

ファイルの格納先

プロトコルコンバータの JAR ファイルを作成するときに，JAR ファイルのルートディレクトリに格納します。プロトコルコンバータの JAR ファイルの作成方法については，「3.3.18(5)　JAR ファイルの作成」を参照してください。

注意事項

- ファイルが存在しない場合，または java.util.Properties クラスの loadFromXML メソッドで読み込めないファイル形式の場合は，汎用カスタムアダプタを起動できません。
- 複数の entry 要素で key 属性を重複して指定しないでください。

記述例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<entry key="classname">protocolconverter.MyProtocolConverter</entry>
</properties>
```

(2)　カスタムアダプタプロパティファイル

カスタムアダプタプロパティファイルは，java.util.Properties クラスの loadFromXML メソッドで読み込むことができる XML 形式のプロパティファイルです。カスタムアダプタプロパティファイルのファイル名は，必ず「customadapter_properties.xml」にしてください。

機能

プロトコルコンバータで使用するプロパティを設定します。

汎用カスタムアダプタの開始時に，カスタムアダプタ開発フレームワークからプロパティが読み込まれます。読み込まれたプロパティは，アダプタコンテキストから getProperty メソッドで取得する Property インスタンスとして取得できます。

ファイルの格納先

汎用カスタムアダプタの定義時に，独自定義ファイルとして汎用カスタムアダプタに登録します。汎用カスタムアダプタの定義方法については，「3.3.18　汎用カスタムアダプタを定義する」を参照してください。

注意事項

ファイルが存在する場合でも，java.util.Properties クラスの loadFromXML メソッドで読み込むことができないファイル形式のときは，汎用カスタムアダプタを起動できません。

記述例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<entry key="init">1</entry>
<entry key="min">0</entry>
<entry key="max">10</entry>
</properties>
```

(3)　カスタムアダプタ用アプリケーション統合属性ファイル

カスタムアダプタ用アプリケーション統合属性ファイルは，アプリケーション統合属性ファイルの一つで，次の場合に使用する XML 形式のファイルです。

- アプリケーション，EJB-JAR，および SessionBean の属性を一括して取得する場合
- アプリケーション，EJB-JAR，および SessionBean の属性を一括して編集する場合

カスタムアダプタ用アプリケーション統合属性ファイルのファイル名は，必ず「cscadapter_property.xml」にしてください。

アプリケーション統合属性ファイルの詳細については，次の個所を参照してください。

- マニュアル「アプリケーションサーバ アプリケーション設定操作ガイド」の「9.2　アプリケーション統合属性ファイルによるプロパティ設定」

- マニュアル「アプリケーションサーバ リファレンス 定義編(アプリケーション／リソース定義)」の「3.1　アプリケーション統合属性ファイル」

機能

アプリケーション統合属性ファイルを使用することで，実行環境に合わせて汎用カスタムアダプタのパラメタを設定できます。

ファイルの格納先

汎用カスタムアダプタの定義時に，独自定義ファイルとして汎用カスタムアダプタに登録します。汎用カスタムアダプタの定義方法については，「3.3.18　汎用カスタムアダプタを定義する」を参照してください。

指定内容

カスタムアダプタ用アプリケーション統合属性ファイルのテンプレートを次の表に示します。このテンプレートを基に，可変部分を変更または追加してください。

また，<hitachi-application-all-property>/<ejb-jar>/<hitachi-session-bean-property>以下の要素として，次の項目の追加および編集ができます。

- <ejb-ref>（リモートインターフェースを持つ Enterprise Bean への参照情報についての定義）
- <ejb-local-ref>（EJB 参照情報の設定）
- <env-entry>（環境エントリについての定義）
- <resource-ref>（リソース参照についての定義）
- <resource-env-ref>（リソース環境変数についての定義）
- <linked-queue>（queue 名称）

表 B-6　カスタムアダプタ用アプリケーション統合属性ファイルのテンプレート

タグ	説明	変更可否
<?xml version="1.0" encoding="MS932"?>	XML 宣言	○
<!DOCTYPE hitachi-application-all-property PUBLIC '-//Hitachi, Ltd.//DTD Application All Property 7.1//EN' 'http://localhost/hitachi-application-all-property_7_1.dtd'>	DOCTYPE 宣言	－
<hitachi-application-all-property>	ルートタグ	－
<hitachi-application-property>	アプリケーションに関する情報の定義開始タグ	－
<description></description>	アプリケーションの説明	○
<icon>	Deploy ツール上に表示される J2EE アプリケーションのアイコンの定義開始タグ	－
<small-icon></small-icon>	スモールアイコン（16×16）のファイル名	○

タグ	説明	変更可否
<large-icon></large-icon>	ラージアイコン（32×32）のファイル名	○
</icon>	Deploy ツール上に表示される J2EE アプリケーションのアイコンの定義終了タグ	−
<lookup-name>CTMADP</lookup-name>	クライアントから EJB をルックアップするときに使う名称	○※1
<security-prop>	Enterprise Bean のセキュリティの管理方法の定義開始タグ	−
<security-method>no_security_for_methods_without_roles</security-method>	セキュリティの管理方法	−
<default-security-role></default-security-role>	security-method に map_methods_without_roles を指定した場合のデフォルトセキュリティロール	−
</security-prop>	Enterprise Bean のセキュリティの管理方法の定義終了タグ	−
<start-order>120</start-order>	J2EE アプリケーション開始時および停止時の順番	△※2
<scheduling-unit>Application</scheduling-unit>	キューの配置モデルの指定	−
<scheduling>	CTM との連携についての定義開始タグ	−
<queue-name>CTMADP</queue-name>	スケジューリングをするキューの名称	○※1
<parallel-count>1</parallel-count>	CTM がアプリケーションを呼び出すために用意するスレッド数	−
<queue-length></queue-length>	スケジューリングをするキューの長さ	−
</scheduling>	CTM との連携についての定義終了タグ	−
<managed-by-ctm>false</managed-by-ctm>	CTM 連携をするかどうかの指定	−
<method-observation-recovery-mode></method-observation-recovery-mode>	J2EE アプリケーションの時間監視機能の障害回復モードの指定	−
</hitachi-application-property>	アプリケーションに関する情報の定義終了タグ	−
<ejb-jar>	EJB に関する情報の定義開始タグ	−

タグ	説明	変更可否
<hitachi-ejb-jar-property>	EJB-JAR の属性に関する情報の定義開始タグ	−
<description xml:lang="en"></description>	EJB-JAR の説明	○
<display-name xml:lang="en">CSCMsgServiceAdapter</display-name>	EJB-JAR の表示名	−
<icon xml:lang="en">	GUI ツール上に表示される EJB-JAR のアイコンの定義開始タグ	−
<small-icon></small-icon>	スモールアイコン（16×16）のファイル名	○
<large-icon></large-icon>	ラージアイコン（32×32）のファイル名	○
</icon>	GUI ツール上に表示される EJB-JAR のアイコンの定義終了タグ	−
<ejb-client-jar></ejb-client-jar>	クライアント JAR ファイルの名称	−
</hitachi-ejb-jar-property>	EJB-JAR の属性に関する情報の定義終了タグ	−
<hitachi-session-bean-property>	SessionBean の属性の定義の開始タグ	−
<description xml:lang="en"></description>	SessionBean の説明	○
<display-name xml:lang="en">CSCMsgServiceAdapterEJB</display-name>	SessionBean の表示名	−
<icon xml:lang="en">	GUI ツール上に表示される SessionBean のアイコンの定義開始タグ	−
<small-icon></small-icon>	スモールアイコン（16×16）のファイル名	○
<large-icon></large-icon>	ラージアイコン（32×32）のファイル名	○
</icon>	GUI ツール上に表示される SessionBean のアイコンの定義終了タグ	−
<session-type>Stateless</session-type>	SessionBean の種別	−
<transaction-type>Container</transaction-type>	トランザクション管理種別	−
<env-entry>	環境エントリについての定義開始タグ	△※3
<description xml:lang="en"></description>	env-entry の説明	

タグ	説明	変更可否
<env-entry-name>CscServerName</env-entry-name>	環境エントリの名称	△※3
<env-entry-type>java.lang.String</env-entry-type>	環境エントリのデータ型	
<env-entry-value>HCSC</env-entry-value>	環境エントリの値	
</env-entry>	環境エントリについての定義終了タグ	
<env-entry>	環境エントリについての定義開始タグ	
<description xml:lang="en"></description>	env-entry の説明	
<env-entry-name>CscClusterName</env-entry-name>	環境エントリの名称	
<env-entry-type>java.lang.String</env-entry-type>	環境エントリのデータ型	
<env-entry-value>Cluster</env-entry-value>	環境エントリの値	
</env-entry>	環境エントリについての定義終了タグ	
<env-entry>	環境エントリについての定義開始タグ	
<description xml:lang="en"></description>	env-entry の説明	
<env-entry-name>AdapterName</env-entry-name>	環境エントリの名称	
<env-entry-type>java.lang.String</env-entry-type>	環境エントリのデータ型	
<env-entry-value>CTMADP</env-entry-value>	環境エントリの値	
</env-entry>	環境エントリについての定義終了タグ	
<container-transaction>	コンテナのトランザクションについての定義開始タグ	－
<description xml:lang="en"></description>	container-transaction の説明	－
<method>	メソッドについての定義開始タグ	－
<description xml:lang="en"></description>	メソッドの説明	－
<method-intf></method-intf>	メソッドが属するインターフェース種別	－
<method-name>*</method-name>	メソッド名	－
</method>	メソッドについての定義終了タグ	－
<trans-attribute>Required</trans-attribute>	メソッドに割り当てるトランザクション属性	－
</container-transaction>	コンテナのトランザクションについての定義終了タグ	－

タグ	説明	変更可否
<session-runtime>	runtime の定義開始タグ	−
<lookup-name>CSCMsgServiceAdapterEJB</lookup-name>	クライアントから EJB をルックアップする場合に使う名前	−
<optional-name>CTMADP</optional-name>	リモートインターフェースの別名	△※3
<maximum-sessions>0</maximum-sessions>	セッションの最大数	○
<stateless>	stateless の定義開始タグ	−
<pooled-instance>	プール内のインスタンスについての定義開始タグ	−
<minimum>1</minimum>	プール内のインスタンスの最小数	○
<maximum>0</maximum>	プール内のインスタンスの最大数	○
</pooled-instance>	プール内のインスタンスについての定義終了タグ	−
<instance-timeout>0</instance-timeout>	インスタンス取得時のタイムアウト時間	○
</stateless>	stateless 定義の終了タグ	−
<enable-scheduling>false</enable-scheduling>	該当する Bean をスケジューリングの対象にするかどうか（スケジューラの Gate として使用するかどうか）の指定	−
<pass-by-reference>true</pass-by-reference>	Bean 単位でのデータの参照渡し（pass by reference）の指定	△※3
<scheduling>	CTM 連携関連の情報の定義開始タグ	−
<queue-name>CSCMsgServiceAdapterEJB</queue-name>	スケジューリングを行う queue 名称	−
<parallel-count>1</parallel-count>	CTM がアプリケーションを呼び出すために用意するスレッド数	−
<queue-length></queue-length>	スケジューリングをするキューの長さ	−
</scheduling>	CTM 連携関連の情報の定義終了タグ	−
<front-ejb>false</front-ejb>	フロント EJB（クライアントが直接呼び出す EJB）かどうかの指定	−
</session-runtime>	runtime の定義終了タグ	−
<start-order>10</start-order>	J2EE アプリケーション開始時および停止時の順番の指定	−

タグ	説明	変更可否
</hitachi-session-bean-property>	SessionBean の属性の定義の終了タグ	－
</ejb-jar>	EJB に関する情報の定義の終了タグ	－
</hitachi-application-all-property>	ルートタグ	－

（凡例）

○：変更できます。

△：自動的に変更されます。

－：変更できません。

注※1

サービス ID を指定してください。

注※2

汎用カスタムアダプタのデプロイ時に自動的に設定されます。

注※3

汎用カスタムアダプタのデプロイ時に自動的に追加または上書きされます。

注意事項

- 「表 B-6　カスタムアダプタ用アプリケーション統合属性ファイルのテンプレート」で示した変更できる項目以外は，テンプレートの内容を変更しないでください。
- ファイル形式がアプリケーション統合属性ファイルとして適切でない場合，汎用カスタムアダプタを配備できません。

(4)　カスタムアダプタ定義ファイル

カスタムアダプタ定義ファイルの機能，および設定できるプロパティについて説明します。

機能

汎用カスタムアダプタの情報を設定します。

ファイルの格納先

```
<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¥samples¥customadapter¥csccustomadapter.properties
```

設定できるプロパティ

custom-adapter.dt-skip={ true | false }

データ変換時に電文の構造変換をスキップするかどうかを指定します。キーまたは値を省略した場合は，構造変換をスキップしません。

- true

  構造変換をスキップします。

- false

  構造変換をスキップしません。

なお，値を指定する場合は，true または false のどちらかを指定してください。true または false 以外の値を指定した場合，コマンド実行時にエラーになります。

付録 B.3　カスタムアダプタ開発フレームワークのサンプルプログラム

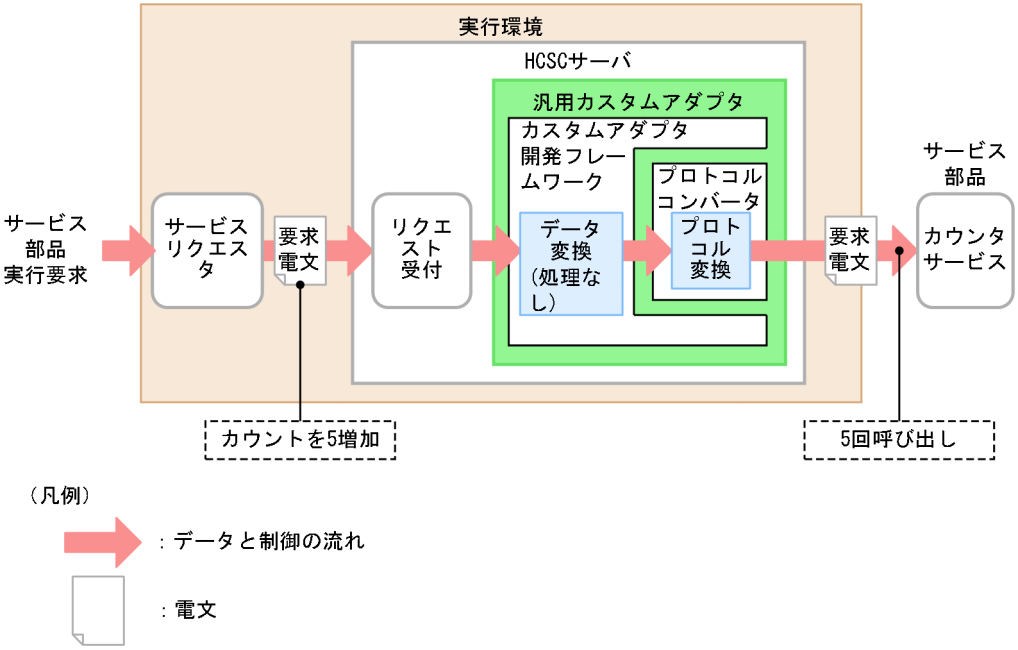
ここでは，カスタムアダプタ開発フレームワークを使用する汎用カスタムアダプタのサンプルプログラムについて説明します。

(1)　サンプルプログラムの概要

サンプルプログラムでは，カウント増加要求（increment），またはカウント減少要求（decrement）を受け付け，カウンタサービスを呼び出す汎用カスタムアダプタを作成します。

カウンタサービスを呼び出す汎用カスタムアダプタの概要を次の図に示します。

図 B-2　カウンタサービスを呼び出す汎用カスタムアダプタの概要



次にカウンタサービス，および汎用カスタムアダプタのプロトコルコンバータについて説明します。

(a)　カウンタサービス

カウンタサービスは，カウントを保持します。

サンプルプログラムのカウンタサービスは通常の Java クラスですが，サービスに見立てています。そのため，カウンタサービスの格納先は，プロトコルコンバータと同じ JAR になります。

カウンタサービスは次に示す 3 つのサービスを提供します。

- カウント増加サービス（increment）
  カウントを 1 つ増やします。
- カウント減少サービス（decrement）
  カウントを 1 つ減らします。ただし，カウントを 0 未満にしようとしたときには，例外が発生します。
- カウント値参照サービス（getCount）
  現在のカウント値を返します。

(b)　プロトコルコンバータ

プロトコルコンバータは，カウンタサービスを呼び出します。

プロトコルコンバータは，次に示す 2 つのオペレーションを持ちます。

- カウントの増加（increment オペレーション）
  カウンタサービスのカウント増加サービスを要求電文で指定された回数呼び出します。
  非同期通信で，要求電文としてバイナリ電文を受け付けます。
  要求電文で増加量を指定でき，指定された増加量の絶対値分だけ，カウンタサービスのカウント増加を実行します。増加量は 1 バイトの符号付の 2 進数で指定します。
  要求電文フォーマットについては，「付録 B.3(2)(a)　increment の要求電文フォーマット」を参照してください。
- カウントの減少（decrement オペレーション）
  カウンタサービスのカウント減少サービスを呼び，指定された数だけ減らします。
  同期通信で，要求電文として XML 電文を受け付け，応答電文として XML 電文を返します。
  要求電文で増加量を指定でき，指定された減少量の整数値分だけ，カウントを減らします。減少量は整数値を文字列で与えます。
  カウンタの減少要求の応答電文は，カウント減少後のカウント値を含みます。
  カウンタサービスのカウント減少サービスを実施した場合に例外が発生したときは，フォルトが発生します。また，要求電文で与えられたカウンタの減少量が Integer クラスの parseInt メソッドで処理できない形式だった場合は，例外を送出します。
  要求電文フォーマットについては「付録 B.3(2)(b)　decrement の要求電文フォーマット」を，応答電文フォーマットについては「付録 B.3(2)(c)　decrement の応答電文フォーマット」を参照してください。

(2)　電文フォーマット

サンプルプログラムで使用する電文形式を次に示します。

(a) increment の要求電文フォーマット

1 バイトの符号付の 2 進整数を 1 つ取るバイナリフォーマット定義ファイルを作成します。ファイル名は，「BinaryRequest.fdx」にしてください。

バイナリフォーマット定義ファイルの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.4　電文フォーマット（バイナリフォーマット定義ファイル）の作成方法」を参照してください。

(b) decrement の要求電文フォーマット

decrement の要求電文フォーマット（XML フォーマット定義ファイル）は，テキストエディタなどを使用して XML 文書形式で作成します。ファイル名は「XMLRequest.xsd」にしてください。

XML フォーマット定義ファイルの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.3　電文フォーマット（XML フォーマット定義ファイル）の作成方法」を参照してください。

作成する XML フォーマット定義ファイルの内容を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.example.org/Req
uestXML" xmlns:tns="http://www.example.org/RequestXML">
    <element name="request" type="string"></element>
</schema>
```

(c) decrement の応答電文フォーマット

decrement の応答電文フォーマット（XML フォーマット定義ファイル）は，テキストエディタなどを使用して XML 文書形式で作成します。ファイル名は「XMLResponse.xsd」にしてください。

XML フォーマット定義ファイルの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.3　電文フォーマット（XML フォーマット定義ファイル）の作成方法」を参照してください。

作成する XML フォーマット定義ファイルの内容を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.example.org/Req
uestXML" xmlns:tns="http://www.example.org/RequestXML">
    <element name="response" type="string"></element>
</schema>
```

(3) サンプルコード

サンプルプログラムで使用するカウンタサービス，およびプロトコルコンバータのサンプルコードについて説明します。

(a) カウンタサービス

カウンタサービスのサンプルコードを次に示します。クラス名は「Counter」，パッケージ名は「service」です。

```
/* All Rights Reserved. Copyright (C) 2008, Hitachi, Ltd. */
package service;

/**
 * カウント値の増減とカウント値の取得クラス(サービスに見立てる)
 * カウント値は常に0以上
 */
public class Counter {
    /**
     * カウント値（常に0以上）
     */
    protected int count;
    /**
     * コンストラクタ
     * @param init countの初期値
     * @throws IllegalArgumentException不正な0未満の値を指定しました。
     * @return void
     */
    public Counter(int init) {

        if (init < 0) {
              throw new IllegalArgumentException();
        }
              this.count = init;
    }
    /**
     * カウントを一つ増やす
     * @return void
     */
    public void increment() {
        ++this.count;
    }

    /**
     * カウントを一つ減らす
     * @throws IllegalStateException countが0以下の時に呼び出しました。
     * @return void
     */
    public void decrement() {
        if (this.count <= 0) {
              throw new IllegalStateException();
        }
        --this.count;
    }
    /**
     * カウント値を取得する
     * @return int カウント値
     */
    public int getCount() {
        return this.count;
```

```
    }
}
```

(b) プロトコルコンバータ

プロトコルコンバータのサンプルコードを次に示します。クラス名は「MyProtocolConverter」，パッケージ名は「protocolconverter」です。

```
/* All Rights Reserved. Copyright (C) 2008, Hitachi, Ltd. */
package protocolconverter;

import java.util.Properties;

import javax.xml.parsers.*;
import org.w3c.dom.*;
import jp.co.Hitachi.soft.csc.msg.adapter.custom.*;

import service.Counter;
/**
 * カスタムプロトコルコンバータインターフェースの実装クラス
 */
public class MyProtocolConverter implements CSCMsgCustomProtocolConverter {
    /**
     * アダプタコンテキスト
     */
    private CSCMsgCustomAdapterContext adapterContext;
    /**
     * カウンタクラス(サービスに見立てている)
     */
    private Counter counter;
    /**
     * DOM. XML電文を作成するために使用
     */
    private DocumentBuilderFactory docBuilderFactory;
    private DocumentBuilder docBuilder;
    /**
     * publicなデフォルトコンストラクタ（必須）
     */
    public MyProtocolConverter() { }
    /**
     * カスタムアダプタ開始時(startが呼ばれる前)に呼ばれます。
     * アダプタコンテキストを取得します。
     * @param adapterContext アダプタコンテキスト
     */
    public void setCustomAdapterContext(
             CSCMsgCustomAdapterContext adapterContext) {
        this.adapterContext = adapterContext;
    }
    /**
     * カスタムアダプタ開始時(setCustomAdapterContextが呼ばれた後)に呼ばれます。
     * 初期化処理を実装します。
     * @throws CSCMsgCustomAdapterException カスタムアダプタで例外が発生しました。
     */
    public void start() throws CSCMsgCustomAdapterException {
        try {
             log("start");
```

```
            Properties properties = this.adapterContext.getProperties();
            String init = properties.getProperty("init");
            if (init == null) {
                // init値の指定は必須
                throw new CSCMsgCustomAdapterException("init");
            }

            // リソースの確保（アダプタのライフサイクル中使い続けるリソース）
            this.counter = new Counter(Integer.parseInt(init));

            docBuilderFactory = DocumentBuilderFactory.newInstance();
            //docBuilderFactory.setNamespaceAware(true);
            docBuilder = docBuilderFactory.newDocumentBuilder();
        } catch (CSCMsgCustomAdapterException e) {
            // カスタムアダプタ側で障害解析に必要なログを出力する
            log(e);
            throw e;
        } catch (Exception e) { // RuntimeExceptionもキャッチする
            // カスタムアダプタ側で障害解析に必要なログを出力する
            log(e);
            // CSCMsgCustomAdapterExceptionでラップして再送出
            throw new CSCMsgCustomAdapterException("start failed", e);
        }
    }
    /**
     * サービス部品呼び出し処理を実装します。
     * @param request 要求メッセージ
     * @param response 応答メッセージ
     * @throws CSCMsgCustomAdapterException カスタムアダプタで例外が発生しました。
     */
    public void invoke(CSCMsgRequestMessage request,
            CSCMsgResponseMessage response)
            throws CSCMsgCustomAdapterException {

        try {
            log("invoke [" + request.getOperationName() + "]");

            // オペレーション名で振り分け
            if (request.getOperationName().equals("increment")) {
                // カウント増加処理
                execIncrement(request, response);
            } else if(request.getOperationName().equals("decrement")) {
                // カウント減少処理
                try {
                    execDecrement(request, response);
                } catch(IllegalStateException e) { // カウントの減少に失敗
                    // ログ出力
                    log(e);
                    // フォルト設定
                    response.setFault("invoke error", "decrement failed",
                            this.adapterContext.getAdapterName(), (Document)null);
                }
            } else {
                throw new CSCMsgCustomAdapterException("unknown operation");
            }
            // 現在のカウントを出力
            log("Count:" + counter.getCount());
```

```
        } catch (CSCMsgCustomAdapterException e) {
            // カスタムアダプタ側で障害解析に必要なログを出力する
            log(e);
            throw e;
        } catch (Exception e) { // RuntimeExceptionもキャッチする
            // カスタムアダプタ側で障害解析に必要なログを出力する
            log(e);
            // CSCMsgCustomAdapterExceptionでラップして再送出
            throw new CSCMsgCustomAdapterException("invoke failed", e);
        }
    }
    /**
     * カウント増加処理
     * @param request 要求メッセージ
     * @param response 応答メッセージ
     * @throws CSCMsgCustomAdapterException カスタムアダプタで例外が発生しました。
     * @throws CSCMsgIllegalMessageTypeException 電文形式がバイナリ形式ではありません。
     * @throws CSCMsgInvalidMessageException 不正な電文が指定されました。
     */
    private void execIncrement(
              CSCMsgRequestMessage request,
              CSCMsgResponseMessage response)
        throws CSCMsgCustomAdapterException,
              CSCMsgIllegalMessageTypeException,
              CSCMsgInvalidMessageException {
        // 非同期オペレーション：increment

        if (response != null) {
              // 通信モデルが同期
              throw new CSCMsgCustomAdapterException("不正な通信モデル");
        }

        if (request.getMessageType() != CSCMsgMessageConstant.MESSAGE_TYPE_BINARY) {
              // 電文フォーマットがバイナリ形式でない
              throw new CSCMsgCustomAdapterException("不正な電文フォーマット(要求)");
        }

        byte[] data = request.getBytes();
        if (data.length != 1) {
            throw new CSCMsgCustomAdapterException("不正なデータ");
        }

        int loop = Math.abs(data[0]);
        for (int i = 0; i < loop; ++i) {
             counter.increment();
        }
    }
    /**
     * カウント減少処理
     * @param request 要求メッセージ
     * @param response 応答メッセージ
     * @throws CSCMsgCustomAdapterException カスタムアダプタで例外が発生しました。
     * @throws CSCMsgIllegalMessageTypeException 電文形式がバイナリ形式ではありません。
     * @throws CSCMsgInvalidMessageException 不正な電文が指定されました。
     * @throws CSCMsgMultipleMessageInsertionException すでに電文またはフォルト情報が格納さ
れています。
     */
    private void execDecrement(
```

```
            CSCMsgRequestMessage request,
            CSCMsgResponseMessage response)
        throws CSCMsgCustomAdapterException,
            CSCMsgIllegalMessageTypeException,
            CSCMsgInvalidMessageException,
            CSCMsgMultipleMessageInsertionException  {
        // 同期オペレーション：decrement

        if (response == null) {
            // 通信モデルが非同期
            throw new CSCMsgCustomAdapterException("不正な通信モデル");
        }

        if (request.getMessageType() != CSCMsgMessageConstant.MESSAGE_TYPE_XML) {
            // 電文フォーマットがXML形式でない
            throw new CSCMsgCustomAdapterException("不正な電文フォーマット(要求)");
        }

        if (response.getMessageType() != CSCMsgMessageConstant.MESSAGE_TYPE_XML) {
            // 電文フォーマットがXML形式でない
            throw new CSCMsgCustomAdapterException("不正な電文フォーマット(応答)");
        }

        // 要求電文解析処理
        NodeList nodeList = request.getXMLDocument().getElementsByTagName("request");
        if (nodeList.getLength() != 1) {
            // ルートノードが見つからない
            throw new CSCMsgCustomAdapterException("不正な要求電文");
        }
        Text text = (Text)nodeList.item(0).getChildNodes().item(0);

        int loop = Math.abs(Integer.parseInt(text.getNodeValue()));
        for (int i = 0; i < loop; ++i) {
            this.counter.decrement();
        }

        // 応答電文の組み立て
        Document responseDoc = docBuilder.newDocument();
        Element rootElement = responseDoc.createElement("response");
        responseDoc.appendChild(rootElement);
        Text countTextNode = responseDoc.createTextNode(Integer.toString(counter.getCount())
);
        rootElement.appendChild(countTextNode);

        // 応答電文の格納
        response.setXMLDocument(responseDoc);
    }
    /**
     * 終了処理，リソースの解放などを行う
     */
    public void stop() {
        log("stop");

        // リソースの解放
        this.counter = null;
        this.adapterContext = null;
        this.docBuilderFactory = null;
        this.docBuilder = null;
```

```
    }
    /**
     * ログ出力
     * @param message 出力するメッセージ
     */
    public void log(String message) {
        // 標準出力への出力結果は，ユーザ出力ログに出力される
        System.out.printf("[%s] %s¥n",
                this.adapterContext.getAdapterName(), message);
                System.out.flush();
    }
    /**
     * ログ出力
     * @param message 出力する例外
     */
    public void log(Exception e) {
        // 標準エラー出力への出力結果は，ユーザエラーログに出力される
        e.printStackTrace();
        System.err.flush();
    }
}
```

(4) プロパティファイル

サンプルプログラムで使用するプロパティファイルについて説明します。

(a) カスタムアダプタ開発フレームワーク動作定義ファイル

カスタムアダプタ開発フレームワーク動作定義ファイルには，プロトコルコンバータのクラス名をパッケージ名付きで指定します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<entry key="classname">protocolconverter.MyProtocolConverter</entry>
</properties>
```

(b) カスタムアダプタプロパティファイル

カスタムアダプタプロパティファイルには，プロトコルコンバータで使用するプロパティを設定します。

カウンタサービスのカウントの初期値を 1 に設定するプロパティファイルを次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<entry key="init">1</entry>
</properties>
```

(5)　プロトコルコンバータ JAR ファイル

プロトコルコンバータクラスファイルとカスタムアダプタ開発フレームワーク動作定義ファイルを含む JAR ファイルを作成します。

プロトコルコンバータ JAR ファイルの作成は，jar コマンドを使用して，プロトコルコンバータクラスファイルとそれが使用するクラスファイル，およびカスタムアダプタ開発フレームワーク動作定義ファイルを JAR ファイルにアーカイブします。

jar コマンドによるアーカイブ時の JAR ファイルのディレクトリ構成については，「3.3.18(5)　JAR ファイルの作成」を参照してください。

(a)　事前準備

カレントディレクトリおよびサブディレクトリに，作成したファイルを次に示すように配置します。

```
framework_properties.xml
protocolconverter/CustomProtocolConverter.class
service/Counter.class
```

(b)　ファイルの作成

ここでは，作成する JAR ファイルを「MyProtocolConverter.jar」というファイル名でアーカイブします。

次のコマンドを実行すると，JAR ファイルが作成されます。

```
jar cf ..¥MyProtocolConverter.jar .¥
```

(6)　EAR ファイル

汎用カスタムアダプタの新規登録に必要な EAR ファイルを作成します。ここでは，作成する EAR ファイルを「CustomAdapter.ear」というファイル名でアーカイブします。

EAR ファイルの作成は jar コマンドを使用し，サービスプラットフォームが提供している JAR ファイルを EAR ファイルにアーカイブします。

jar コマンドによるアーカイブ時の EAR ファイルのディレクトリ構成については，「3.3.18(6)　EAR ファイルの作成」を参照してください。

(a)　事前準備

カレントディレクトリに次に示すファイルのコピーを格納します。

```
<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>¥CSC¥lib¥cscmsg_adpejb.jar
```

(b)　ファイルの作成

作成する EAR ファイルのファイル名は「CustomAdapter.ear」にしてください。

次のコマンドを実行すると，EAR ファイルが作成されます。

```
jar cf ..¥CustomAdapter.ear .¥
```

(c)　ファイルの登録

汎用カスタムアダプタとして使用するには，アダプタ追加ウィザードを利用し，作成した EAR ファイルを HCSC-Definer に登録する必要があります。登録方法については，「3.2.18　汎用カスタムアダプタを新規に追加する」を参照してください。

付録 B.4　汎用カスタムアダプタの定義画面

ここでは，汎用カスタムアダプタを定義するときのサービスアダプタ定義画面について説明します。

(1)　サービスアダプタ定義（基本）画面

サービスアダプタ定義（基本）画面の項目について説明します。

図 B-3　サービスアダプタ定義（基本）画面



(a) ［サービス部品制御情報］

汎用カスタムアダプタ新規作成時に指定したサービス部品についての情報が表示されます。

［サービス名］

サービスアダプタ定義追加ウィザードで設定した内容が表示されます。

［サービス ID］

サービス ID を入力します。サービス ID は，8 バイト以内の英数字およびアンダーバー（_）で指定します。

［サービス種別］

サービス部品の種別として「汎用カスタムアダプタ」が表示されます。

［アドレス］

この項目は使用しません。

［最大インスタンス数］

サービス部品の最大インスタンス数が表示されます。

［サービスクラス名］

この項目は使用しません。

［オペレーション］

オペレーション名が表示されます。サービス部品が持つオペレーション名をドロップダウンリストから選択できます。

［システム例外をフォルトに変換する］チェックボックス

サービス部品で発生した例外をフォルトに変換する場合にチェックします。

［追加］ボタン

オペレーションを追加するダイアログが表示されます。

［削除］ボタン

オペレーション欄で選択したオペレーションを削除します。

(b) ［オペレーション情報］

サービス部品のオペレーションについての情報が表示されます。

［オペレーション名］

サービス部品制御情報のオペレーション欄で選択したオペレーションの名称が表示されます。

［通信モデル］

サービス部品制御情報のオペレーション欄で選択したオペレーションの通信モデルが表示されます。同期または非同期をドロップダウンリストから選択できます。選択内容によって，次のように表示が変わります。

- 同期を選択した場合：応答電文とフォルト電文の情報が表示されます。

- 非同期を選択した場合：応答電文とフォルト電文の情報は表示されません。非同期のまま，アダプタ定義を保存すると，応答電文とフォルト電文の全項目は消えます。

(c) ［要求電文］

サービスリクエスタから送信される要求電文の情報が表示されます。

［any 型を使う］チェックボックス

要求電文のデータ型を指定しないで作成する場合にチェックします。

［標準］

［使う］チェックボックス

サービスリクエスタから受け付けた標準電文の内容を確認したり，標準電文を汎用カスタムアダプタ用の電文のフォーマットにデータ変換したりする場合にチェックします。

［電文フォーマット］

サービスリクエスタから受け付けた標準電文のフォーマット名が表示されます。

［参照］ボタン

標準電文フォーマットを設定するダイアログが表示されます。

［表示］ボタン

標準電文のフォーマットが表示されます。

［出力］ボタン

標準電文フォーマットの出力先を指定するダイアログが表示されます。

［サービス部品］

［電文フォーマット］

汎用カスタムアダプタのプロトコルコンバータが受け取る要求電文の電文フォーマットのフォーマット名が表示されます。

［参照］ボタン

汎用カスタムアダプタのプロトコルコンバータが受け取る要求電文の電文フォーマットを設定するダイアログが表示されます。

［表示］ボタン

汎用カスタムアダプタのプロトコルコンバータが受け取る要求電文の電文フォーマットが表示されます。

［出力］ボタン

汎用カスタムアダプタのプロトコルコンバータが受け取る要求電文の電文フォーマットの出力先を指定するダイアログが表示されます。

[データ変換定義]

標準電文から汎用カスタムアダプタのプロトコルコンバータが受け取る要求電文へデータ変換するための定義ファイル名を入力します。標準電文の［使う］チェックボックスにチェックを入れていない場合は，この項目は使用しません。

［編集］ボタン

データ変換定義画面が表示されます。データ変換定義欄で設定した定義ファイルの内容を編集できます。初めて定義する場合は，［ルート要素選択］ダイアログが表示されます。標準電文の［使う］チェックボックスにチェックを入れていない場合は，この項目は使用しません。

［削除］ボタン

［編集］ボタンで設定したデータ変換定義を削除します。標準電文の［使う］チェックボックスにチェックを入れていない場合は，この項目は使用しません。

(d) ［応答電文］

サービス部品から送信される応答電文の情報が表示されます。

オペレーション情報の通信モデル欄で，「同期」を選択した場合だけ，設定できます。

[any 型を使う］チェックボックス

応答電文のデータ型を指定しないで作成する場合にチェックします。

[標準]

［使う］チェックボックス

サービス部品から受け付けた汎用カスタムアダプタ用の電文の内容を確認したり，汎用カスタムアダプタ用の電文を標準電文のフォーマットにデータ変換したりする場合にチェックします。

［電文フォーマット］

サービスリクエスタへ応答する標準電文のフォーマット名が表示されます。

［参照］ボタン

標準電文フォーマットを設定するダイアログが表示されます。

［表示］ボタン

標準電文のフォーマットが表示されます。

［出力］ボタン

標準電文フォーマットの出力先を指定するダイアログが表示されます。

[サービス部品]

［電文フォーマット］

汎用カスタムアダプタのプロトコルコンバータが送信する応答電文の電文フォーマットのフォーマット名が表示されます。

［参照］ボタン

汎用カスタムアダプタのプロトコルコンバータが送信する応答電文の電文フォーマットを設定するダイアログが表示されます。

［表示］ボタン

汎用カスタムアダプタのプロトコルコンバータが送信する応答電文の電文フォーマットが表示されます。

［出力］ボタン

汎用カスタムアダプタのプロトコルコンバータが送信する応答電文の電文フォーマットの出力先を指定するダイアログが表示されます。

［データ変換定義］

汎用カスタムアダプタのプロトコルコンバータから送信する応答電文から標準電文へデータ変換するための定義ファイル名を入力します。標準電文の［使う］チェックボックスにチェックを入れていない場合は，この項目は使用しません。

［編集］ボタン

データ変換定義画面が表示されます。データ変換定義欄で設定した定義ファイルの内容を編集できます。初めて定義する場合は，［ルート要素選択］ダイアログが表示されます。標準電文の［使う］チェックボックスにチェックを入れていない場合は，この項目は使用しません。

［削除］ボタン

［編集］ボタンで設定したデータ変換定義を削除します。標準電文の［使う］チェックボックスにチェックを入れていない場合は，この項目は使用しません。

(e) ［フォルト電文］

サービス部品のフォルト電文についての情報が表示されます。

［追加］ボタン

フォルト名を追加するダイアログが表示されます。汎用カスタムアダプタの定義では，このボタンは使用しません。

［削除］ボタン

フォルト名欄で選択したフォルト名を削除します。汎用カスタムアダプタの定義では，このボタンは使用しません。

［フォルト名］

サービス部品のフォルトの名称が表示されます。汎用カスタムアダプタの定義では，この欄は使用しません。

［電文フォーマット］

サービス部品のフォルト電文のフォーマット名が表示されます。汎用カスタムアダプタの定義では，この欄は使用しません。

［参照］ボタン

フォルト電文のフォーマットを設定するダイアログが表示されます。汎用カスタムアダプタの定義では，このボタンは使用しません。

［表示］ボタン

フォルト電文のフォーマットが表示されます。汎用カスタムアダプタの定義では，このボタンは使用しません。

［出力］ボタン

フォルト電文のフォーマットの出力先を指定するダイアログが表示されます。汎用カスタムアダプタの定義では，このボタンは使用しません。

(2)　サービスアダプタ定義（詳細）画面

サービスアダプタ定義（詳細）画面の項目について説明します。

図 B–4　サービスアダプタ定義（詳細）画面



(a)　［サービスアダプタ制御情報］

汎用カスタムアダプタの新規作成時に指定したサービス部品についての情報が表示されます。

サービスアダプタ

［サービスアダプタ（EJB-JAR ファイル）］

汎用カスタムアダプタの EJB-JAR ファイルの名称が表示されます。

［参照］ボタン

汎用カスタムアダプタの EJB-JAR ファイル名を設定するダイアログが表示されます。

ユーティリティクラス

[ユーティリティクラス（JAR ファイル）]

［追加］ボタンで設定したユーティリティクラスが表示されます。

[追加] ボタン

ユーティリティクラスを追加するダイアログが表示されます。「3.3.18　汎用カスタムアダプタを定義する」で作成した JAR ファイルを追加します。

[出力] ボタン

ユーティリティクラスの出力先を指定するダイアログが表示されます。

[削除] ボタン

ユーティリティクラス欄で選択したファイルを削除します。

独自定義ファイル

[独自定義ファイル]

［追加］ボタンで設定したプロトコルコンバータ用独自定義ファイル，カスタムアダプタプロパティファイル，カスタムアダプタ用アプリケーション統合属性ファイル，およびカスタムアダプタ定義ファイルが表示されます。

[追加] ボタン

独自定義ファイル（プロトコルコンバータ用独自定義ファイル，カスタムアダプタプロパティファイル，カスタムアダプタ用アプリケーション統合属性ファイル，およびカスタムアダプタ定義ファイル）を追加するダイアログが表示されます。

[編集] ボタン

独自定義ファイル（プロトコルコンバータ用独自定義ファイル，カスタムアダプタプロパティファイル，カスタムアダプタ用アプリケーション統合属性ファイル，およびカスタムアダプタ定義ファイル）を編集するためのエディタが起動します。独自定義ファイルを複数選択してクリックした場合は，選択されたすべての独自定義ファイルに対してエディタが起動します。エディタ上で独自定義ファイルを保存したあと，汎用カスタムアダプタを保存することで，編集した内容がリポジトリに反映されます。独自定義ファイルの編集中に汎用カスタムアダプタを保存しようとすると，独自定義ファイルを保存してよいか確認するダイアログが表示されます。

[出力] ボタン

独自定義ファイル（プロトコルコンバータ用独自定義ファイル，カスタムアダプタプロパティファイル，カスタムアダプタ用アプリケーション統合属性ファイル，およびカスタムアダプタ定義ファイル）の出力先を指定するダイアログが表示されます。

[削除] ボタン

独自定義ファイル欄で選択したファイルを削除します。

付録 B.5　汎用カスタムアダプタを使用する際の環境設定

ここでは，汎用カスタムアダプタを使用するときに，HCSC サーバの実行環境に対して設定する必要がある内容について説明します。

(1)　保護区リストファイルの設定

HCSC サーバ上で汎用カスタムアダプタを使用するには，プロトコルコンバータ（プロトコル変換処理）を実装しているクラスを，メソッドキャンセルを禁止するクラスとして保護区リストファイルに設定します。

「保護区リストファイル」の要素として，汎用カスタムアダプタの実装クラス名またはプリフィックス名を設定してください。

設定した保護区リストファイルは，J2EE サーバの起動時に読み込まれ，サービスプラットフォームをインストールしたマシン上で動作するすべての J2EE サーバで有効になります。

「保護区リストファイル」については，マニュアル「アプリケーションサーバ リファレンス 定義編(サーバ定義)」の「2.2.5　criticalList.cfg（保護区リストファイル)」を参照してください。

付録 B.6　トラブルシューティング

ここでは，汎用カスタムアダプタの運用時に発生する障害と考えられる原因と対処方法を説明します。

(1)　汎用カスタムアダプタの配備に関するトラブルシューティング

汎用カスタムアダプタを配備できない場合の原因，原因の特定および対策を次に示します。

(a)　原因

汎用カスタムアダプタのデプロイに失敗する場合，主に次の原因が考えられます。

1. カスタムアダプタ用アプリケーション統合属性ファイルの形式または内容に誤りがある。
2. 「3.3.18　汎用カスタムアダプタを定義する」に示す正しい手順に従っていないため，誤りがある EAR ファイルを登録した。

(b)　原因の特定

デプロイ時に表示されるコンソールログを確認してください。

- 次のようなメッセージが表示されていた場合：

```
KEOS50007-E An exception occurred during the operation. (Cosminexus Manager
name = <マネージャ識別名>, object name = jp.co.Hitachi.soft.csc.msg:type=
CSCMsgAdapterBuilder, operation = setup, details = The application attributes
file is invalid. (<詳細情報>) ErrorCode=KDEC10031-E)
```

  原因の 1.または 2.のおそれがあります。

具体的な原因については，＜詳細情報＞を確認してください。

(c) 対策

次の対処をしてください。

- カスタムアダプタ用アプリケーション統合属性ファイルの形式または内容に誤りがないか確認してください。誤りがある場合は，ファイルを再作成してください。アプリケーション統合属性ファイルについては，「付録 B.2(3) カスタムアダプタ用アプリケーション統合属性ファイル」を参照してください。
- 使用している EAR ファイルの形式または内容に誤りがないか確認してください。誤りがある場合は，ファイルを再作成してください。EAR ファイルの作成方法については，「3.3.18(6) EAR ファイルの作成」を参照してください。

(2) 汎用カスタムアダプタの開始に関するトラブルシューティング

汎用カスタムアダプタを開始できない場合の原因，原因の特定および対策を次に示します。

(a) 原因

汎用カスタムアダプタの開始に失敗する場合，主に次の原因が考えられます。

1. 登録されているすべての JAR ファイルのルートフォルダに，カスタムアダプタ開発フレームワーク動作定義ファイルが配置されていない。
2. カスタムアダプタ開発フレームワーク動作定義ファイルの形式に誤りがある。
3. カスタムアダプタプロパティファイルの形式に誤りがある。
4. プロトコルコンバータ JAR ファイルにプロトコルコンバータのクラスファイルが含まれていない。
5. カスタムアダプタ開発フレームワーク動作定義ファイルで，プロトコルコンバータの指定に誤りがある，または存在しないプロトコルコンバータを指定している。
6. プロトコルコンバータに public なデフォルトコンストラクタがない。
7. プロトコルコンバータの start メソッドで例外が発生した。

(b) 原因の特定

メッセージログファイルを確認してください。

- メッセージログに次のようなメッセージが記録されていた場合：

```
KDEC02513-E An attempt to read the file has failed. (adapter name =
<アダプタ名>, file name =<ファイル名>, information =<詳細情報>)
```

埋め文字と原因の対応を次の表に示します。

表 B-7　KDEC02513-E 発生時の原因

ファイル名	詳細情報	原因
framework_properties.xml	InputStream is null	原因の 1.のおそれがあります。
	InvalidPropertiesFormatException	原因の 2.のおそれがあります。
defs/customadapter_properties.xml	InvalidPropertiesFormatException	原因の 3.のおそれがあります。

- メッセージログに次のようなメッセージが記録されていた場合：

```
KDEC03015-E An initialization error occurred in the protocol converter.
(adapter name = <アダプタ名>, information =<原因となった例外名>)
```

埋め文字と原因の対応を次の表に示します。

表 B-8　KDEC03015-E 発生時の原因

原因となった例外名	原因
ClassNotFoundException	原因の 4.または 5.のおそれがあります。
IllegalAccessException	原因の 6.のおそれがあります。
CSCMsgCustomAdapterException	原因の 7.のおそれがあります。

(c)　対策

原因が 1.～2.の場合，次の対処をしてください。

- カスタムアダプタ開発フレームワーク動作定義ファイルの形式または内容に誤りがないか確認してください。誤りがある場合は，ファイルを再作成してください。カスタムアダプタ開発フレームワーク動作定義ファイルの詳細は，「付録 B.2(1)　カスタムアダプタ開発フレームワーク動作定義ファイル」を参照してください。
- カスタムアダプタ開発フレームワーク動作定義ファイルをプロトコルコンバータ JAR ファイルの中の指定された場所に配置しているか確認してください。指定された場所に配置されていない場合は，正しい場所に配置し直してください。

原因が 3.の場合，次の対処をしてください。

- カスタムアダプタプロパティファイルの形式または内容に誤りがないか確認してください。誤りがある場合は，ファイルを再作成してください。カスタムアダプタプロパティファイルの詳細は，「付録 B.2(2)　カスタムアダプタプロパティファイル」を参照してください。

原因が 4.～5.の場合，次の対処をしてください。

- プロトコルコンバータ JAR ファイルにプロトコルコンバータのクラスファイルが含まれているか確認してください。含まれていない場合は，jar コマンドを使用して，プロトコルコンバータクラスファイルとそれが使用するクラスファイルおよび，カスタムアダプタ開発フレームワーク動作定義ファイルを JAR ファイルにアーカイブしてください。プロトコルコンバータ JAR ファイルの作成方法については，「3.3.18(5)(a)　プロトコルコンバータ JAR ファイルの作成」を参照してください。

- カスタムアダプタ開発フレームワーク動作定義ファイルで，プロトコルコンバータ名をパッケージ名付きで指定しているか確認してください。誤りがある場合は，プロトコルコンバータ名をパッケージ名付きで指定し直してください。
- カスタムアダプタ開発フレームワーク動作定義ファイルで，プロトコルコンバータ名を指定する entry 要素の key 属性の値は classname か確認してください。誤りがある場合は，key 属性が classname である entry タグに，プロトコルコンバータのクラス名を記述してください。

原因が 6.の場合，次の対処をしてください。

- プロトコルコンバータに public なデフォルトコンストラクタを定義しているか確認してください。定義されていない場合は，public なデフォルトコンストラクタを定義してください。

原因が 7.の場合，次の対処をしてください。

- プロトコルコンバータの start メソッドの処理を確認してください。処理内容に誤りがある場合は，実装し直してください。start メソッドの形式については，「付録 B.1(2)(a) CSCMsgCustomAdapterContext インターフェース」を参照してください。
- 使用している環境や設定がプロトコルコンバータの開始できる状態か確認してください。

(3) プロトコルコンバータからの操作に関するトラブルシューティング

プロトコルコンバータからファイルの操作をしようとすると java.security.AccessControlException 例外が発生する場合の原因，原因の特定および対策を次に示します。

(a) 原因

Java のセキュリティ機能によって，ファイルアクセスが禁止された可能性があります。

(b) 原因の特定

プロトコルコンバータの中で，ファイルアクセス時に java.security.AccessControlException が送出されることを確認してください。

(c) 対策

次の対策をしてください。

- コンテナ拡張ライブラリを使用する

  コンテナ拡張ライブラリの利用については，マニュアル「アプリケーションサーバ 機能解説 基本・開発編(コンテナ共通機能)」の「19.2 コンテナ拡張ライブラリの利用」を参照してください。コンテナ拡張ライブラリの実装については，マニュアル「アプリケーションサーバ 機能解説 基本・開発編(コンテナ共通機能)」の「19.3.2 コンテナ拡張ライブラリの作成と利用の流れ」を参照してください。
- セキュリティポリシーファイルの設定を書き換える

  セキュリティポリシーの設定については，マニュアル「アプリケーションサーバ 機能解説 拡張編」の「8.8.2 セキュリティポリシーの設定」を参照してください。
- ユーザログ機能を使用する（ファイルアクセスの目的がログ出力の場合）

ユーザログ機能については，マニュアル「アプリケーションサーバ 機能解説 拡張編」の「8.12　ユーザログ機能を使用する場合の注意事項」を参照してください。

付録 C　DB アダプタ定義支援機能を利用した DB アダプタの定義

ここでは，DB アダプタ定義支援機能を利用して，DB アダプタの定義を作成する方法について説明します。

付録 C.1　DB アダプタ定義支援機能の概要

DB アダプタ定義支援機能は，大きく分けて，次の 2 つの機能があります。

- DB アダプタ用各種定義ファイルの作成

  DB アダプタ定義時に必要なファイルを生成します。この機能を使用すると，次のファイルを生成できます。

  - SQL オペレーション定義ファイル
  - アプリケーション統合属性ファイル
  - 要求電文および応答電文の標準電文
  - 要求電文および応答電文のサービス部品電文

注意事項

サービスリクエスタからの 1 回のサービス部品の実行要求で，複数の SQL を実行する機能に対応した，要求電文の標準電文フォーマットについては生成できません。1 回の要求で複数の SQL を実行する要求電文の標準電文フォーマットについては，手動で作成してください。

- 引数要素のある SQL 文のデバッグ

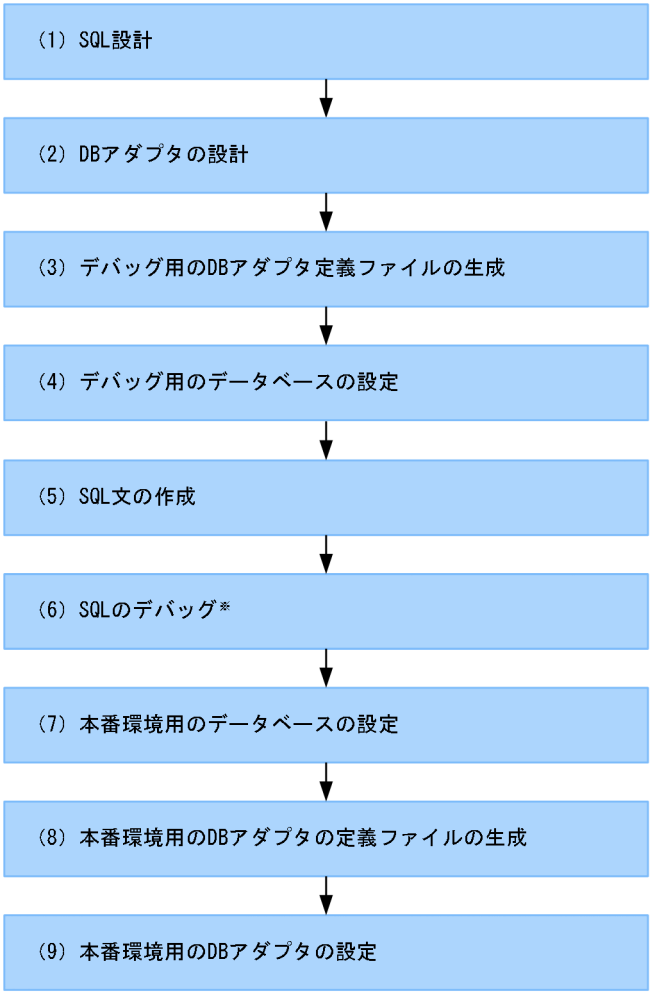
  DB アダプタ定義支援機能では，引数要素がある SQL 文をデバッグするために，次の 3 つの機能を提供します。

  - 引数要素を抽出する機能
  - 引数要素に値を代入したあとで SQL 文を実行する機能
  - SQL 実行結果を確認する機能

付録 C.2　DB アダプタの定義

開発環境上で DB アダプタを定義する場合の作業の流れを次の図に示します。

図 C-1　DB アダプタの定義の流れ



注※
本番環境用のデータベースとは別に，デバッグ用のデータベースを用意している場合にだけ実施します。

各作業の詳細は，以降で説明します。なお，ここでは，HiRDB を使用することを前提として説明します。Oracle を使用する場合は，各種定義を読み替えてください。

DB アダプタ定義支援機能では，HiRDB，Oracle 以外のデータベースはサポートしていません。

注意事項

Eclipse プロジェクトをクローズする場合は，DB アダプタ定義支援機能のエディタをすべて閉じてから実施してください。

(1)　SQL 設計

DB アダプタ定義支援機能を使って，DB アダプタ上で実行する SQL 文の内容を設計します。SQL 実行文の表や列の名称およびデータなどの可変部分については，引数要素として設計します。

ただし，DB アダプタ定義支援機能で指定できる SQL 命令は，デフォルトで次の 2 つです。

- SELECT
- INSERT

SQL オペレーション定義ファイルで sqloperation_chk 属性に N を指定することで，次の SQL を SQL 命令に指定できます。

- UPDATE
- DELETE

また，指定できるデータ型は，DB Connector で使用する JDBC ドライバがサポートしているデータ型だけです。

(2)　DB アダプタの設計

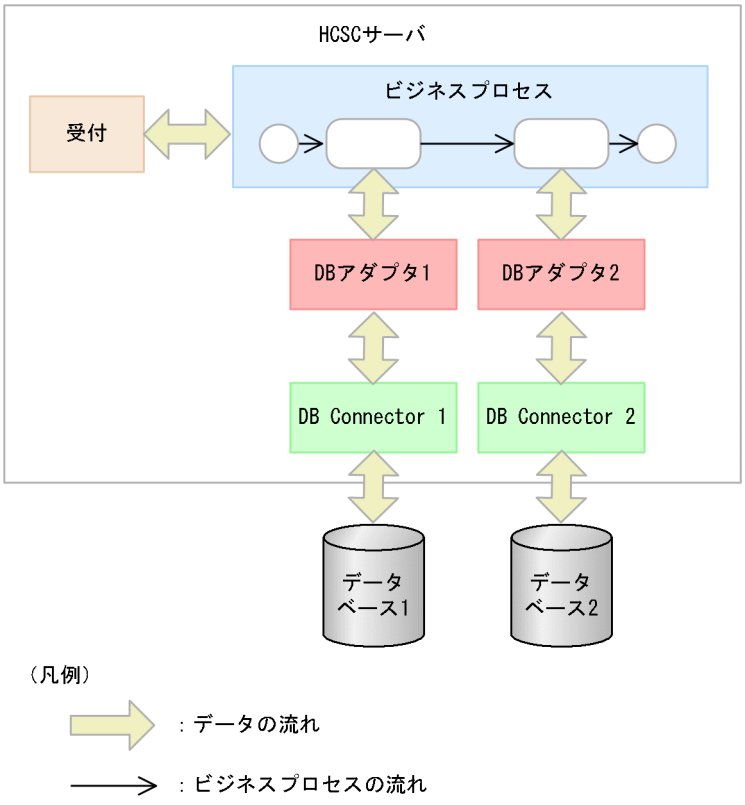
DB アダプタの内容を設計します。次の内容について設計してください。

(a)　DB アダプタの作成単位

DB アダプタは DB Connector 単位で必要なため，DB Connector の構成に合わせて DB アダプタを用意します。

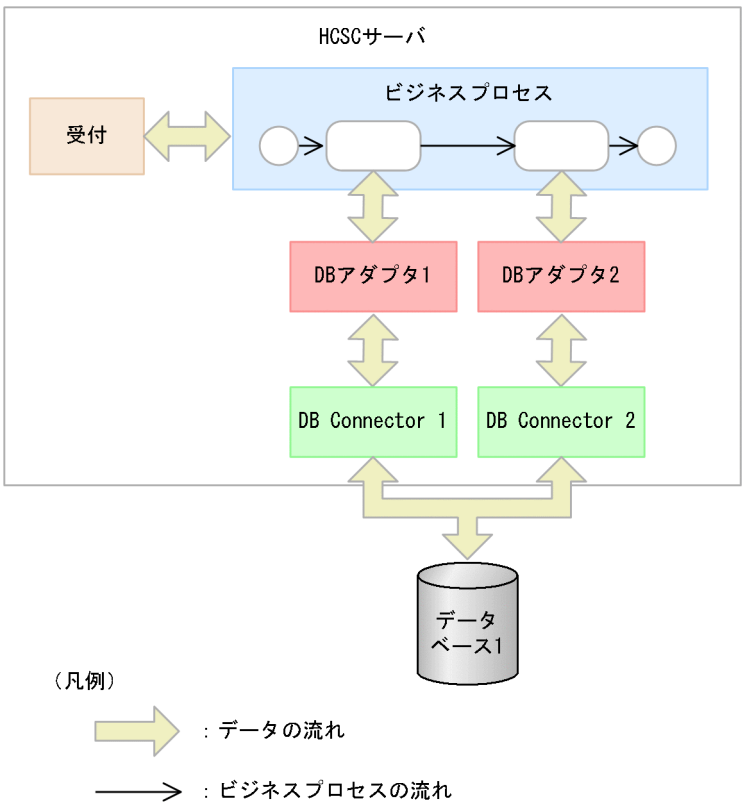
例えば，複数のデータベースと連携するシステムでは，データベースごとに DB Connector が存在します。このため，データベースの DB Connector ごとに DB アダプタを作成します。

図 C-2　データベースごとに DB Connector が存在する場合



また，コネクションプールの設定値を DB アダプタごとに設定する場合などは，同じデータベースに複数の DB Connector を用意します。この場合は，データベースの数に関係なく，DB Connector 単位で DB アダプタを作成します。

図 C-3　1 つのデータベースに複数の DB Connector が存在する場合



(b)　SQL 文の定義単位

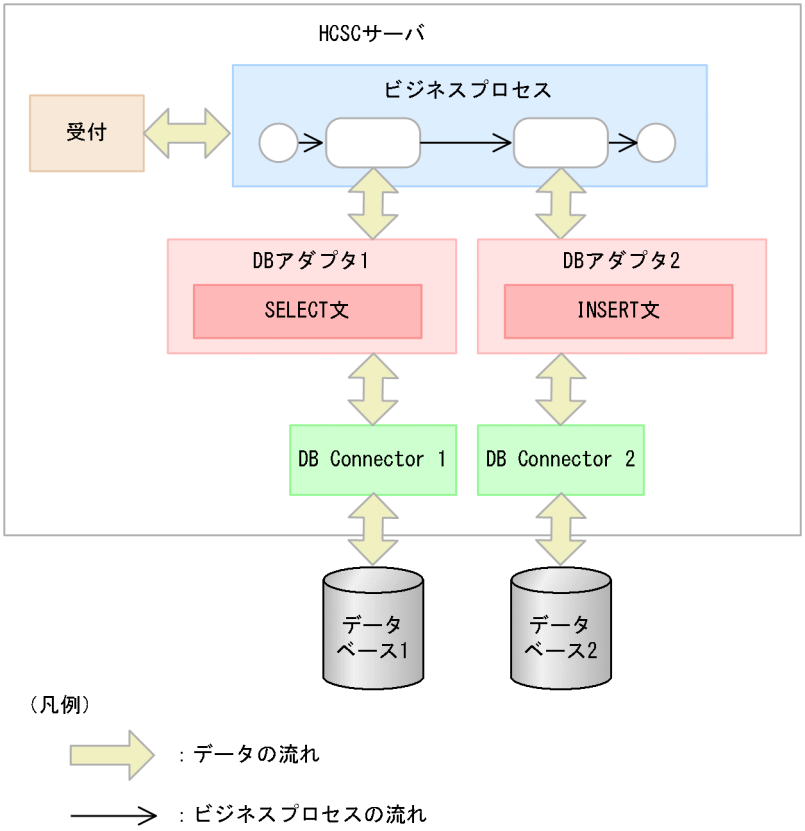
DB アダプタには，1 つ以上の SQL 文を定義できます。各 DB アダプタで定義する SQL 文の内容を設計します。SQL 文の設計例を次に示します。

- **データベースごとに異なる SQL 文を実行したい場合**

  データベースごとに異なる SQL 文を実行したい場合は，それぞれのデータベースに対応する DB アダプタ上に SQL 文を定義します。

  例えば，図中の「データベース 1」で検索した結果を「データベース 2」に登録したい場合は，「DB アダプタ 1」に SELECT を，「DB アダプタ 2」に INSERT を定義します。

図 C-4　データベースごとに異なる SQL 文を実行したい場合



（凡例）

：データの流れ

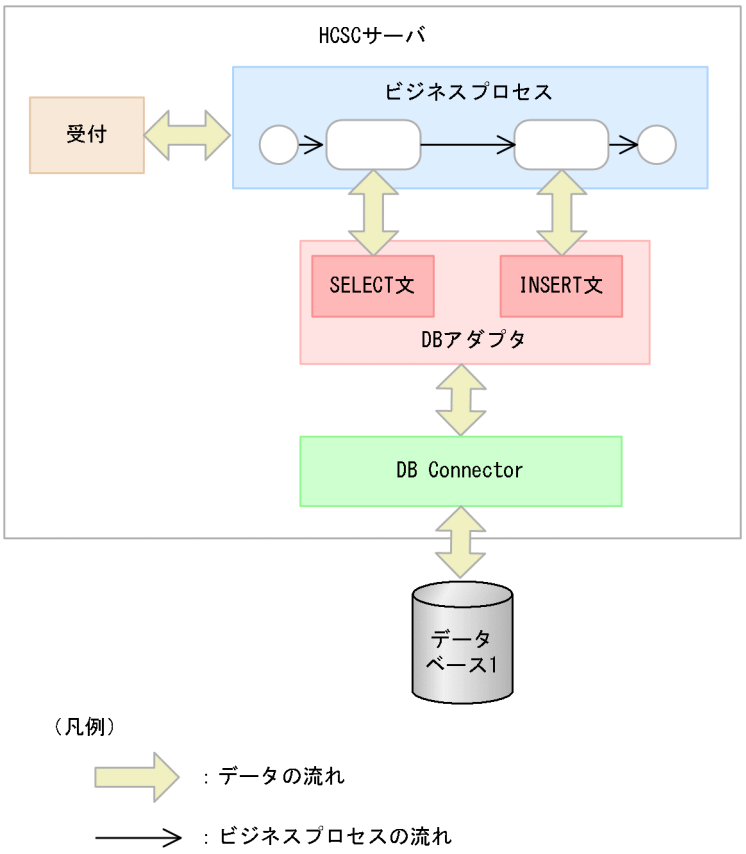
：ビジネスプロセスの流れ

- **同一データベースに対して，異なる SQL 文を実行したい場合**

  同じデータベースに対して，アクティビティごとに異なる SQL 文を実行したい場合は，DB アダプタ上に複数の SQL 文を定義します。

  例えば，最初のアクティビティで検索した結果を，次のアクティビティで登録したい場合は，同じ DB アダプタ上に SELECT と INSERT の両方を定義します。

図 C-5　同一のデータベースに対して異なる SQL 文を実行したい場合



DB アダプタ定義支援機能を使用して，1 つの DB アダプタ上に複数の SQL 文を定義する場合は，オペレーションを呼び出す SQL 文単位で定義します。

オペレーションの指定方法については，「(5)　SQL 文の作成」を参照してください。

(3)　デバッグ用の DB アダプタの定義ファイルの生成

デバッグに使用する DB アダプタの定義ファイルのひな形を作成します。DB アダプタの定義ファイルは，Eclipse 上で作成します。DB アダプタの定義ファイルを新規作成する方法を次に示します。

注意事項

- DB アダプタ定義ファイルの編集中に次のメッセージが表示された場合，［パッケージ・エクスプローラー］ビュー上で，プロジェクトを選択して右クリックし，［更新］メニューを選択してリソースとファイル・システムを同期させてください。

  リソース '<ファイルパス>' はファイル・システムと非同期です。
- DB アダプタ定義ファイルを削除する場合，DB アダプタ定義ファイル作成時に作られた出力先ディレクトリは削除されません。必要に応じて，削除してください。
- DB アダプタ定義ファイルを操作する場合は，HCSCTE-DBEditor パースペクティブ選択時に開かれるビューを使用してください。

1. Eclipse 上にプロジェクトを作成します。

プロジェクトの種類は，HCSCTE プロジェクトを推奨します。HCSCTE プロジェクトの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「3.1.1　プロジェクトを作成する」を参照してください。

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. [HCSC-DBEditor］を選択し，［開く］ボタンをクリックします。
［DB アダプタ開発支援］パースペクティブが表示されます。

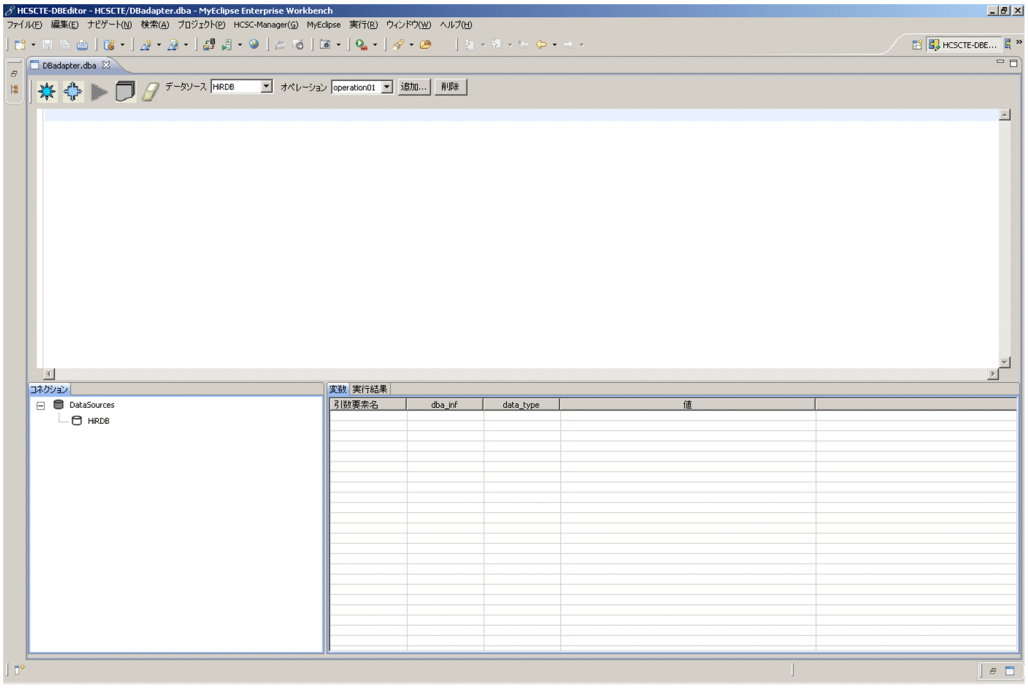
4. Eclipse のメニューから，［ファイル］－［新規］－［その他］を選択します。
ウィザードを選択するダイアログが表示されます。

5. [HCSCTE-DBEDITOR］の［DBEditor ファイル］を選択して，［次へ］ボタンをクリックします。
新規 DB アダプタ定義ファイルウィザードが表示されます。

6. [親フォルダを入力または選択］で，DB アダプタの定義ファイルを保存するフォルダを入力または選択します。

7. 保存する定義ファイルのファイル名を［ファイル名］に指定します。

8. [終了］ボタンをクリックします。
新しい定義ファイルが作成され，DB アダプタ開発支援画面が表示されます。



(4)　デバッグ用のデータベースの設定

DB アダプタの定義ファイルに，使用するデータベースの内容を設定します。本番環境用のデータベースとは別にデバッグ用のデータベースを使用するかどうかによって，設定方法が異なります。デバッグ用のデータベースを使用しない場合は，SQL のデバッグができません。

参考

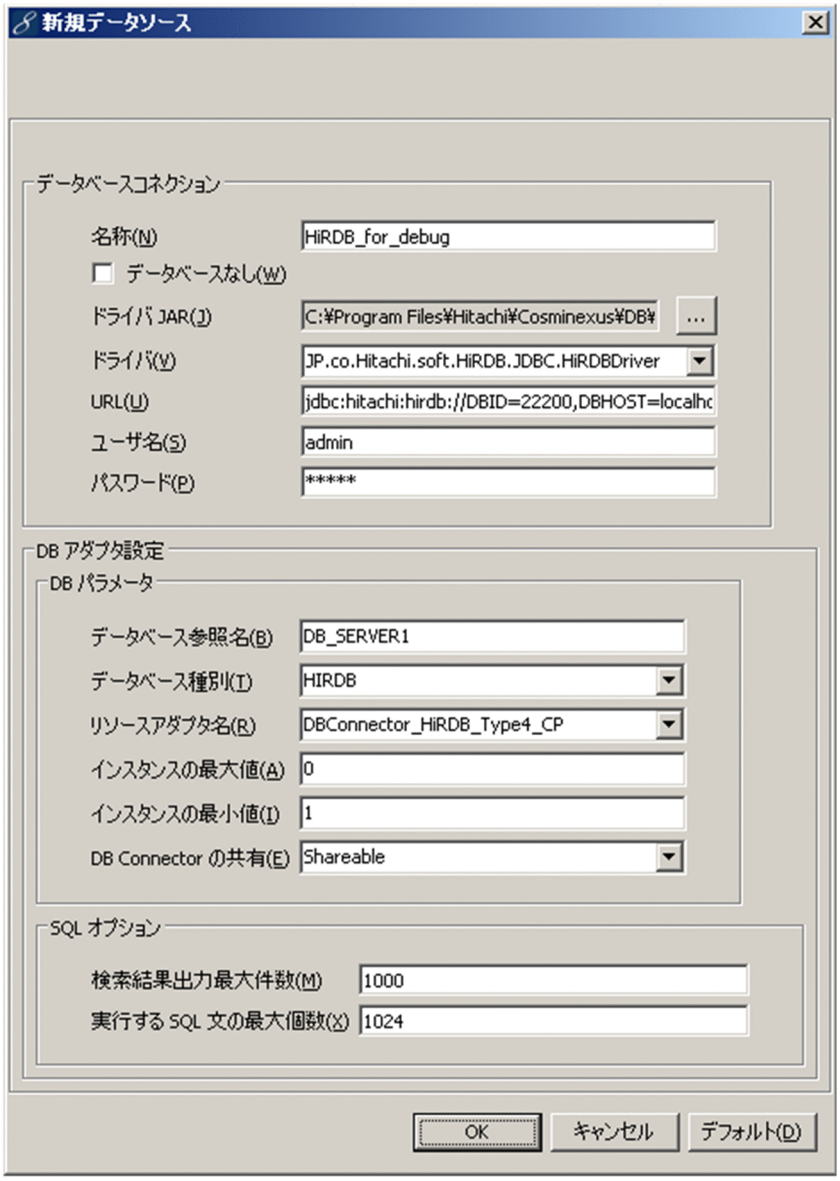
Service Architect に含まれる構成ソフトウェアの HiRDB/Single Server をインストールすると，デバッグ用のデータベースとして使用できます。インストールの詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.1.2　インストール」を参照してください。

(a)　本番環境用とは別にデバッグ用のデータベースを用意する場合

本番環境用のデータベースとは別に，デバッグ用のデータベースを用意する場合の設定方法を次に示します。

1. Eclipse 上で DB アダプタの定義ファイルを開きます。
   DB アダプタ開発支援画面が表示されます。

2. ［コネクション］タブにある［DataSources］を右クリックし，［新規データソース］を選択します。
   ［新規データソース］ダイアログが表示されます。



3. [データベースコネクション] の入力フィールドに値を入力します。

SQL をデバッグするための設定をします。

ここでは，JDBC ドライバを利用して接続するための設定をします。

表 C-1 [データベースコネクション] の入力フィールドに定義する内容（[新規データソース] ダイアログ）

表示項目	説明	デフォルト値※
名称	DB アダプタ開発支援画面の［コネクション］タブに表示する名称です。DB エディタ定義ファイル内で一意になるような任意の名称を指定します。	HiRDB
データベースなし	デバッグ用のデータベースがない場合にチェックします。	チェックなし
ドライバ JAR	JDBC ドライバの JAR ファイルを指定します。	<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>¥DB¥CLIENT¥UTL¥pdjdbc2.jar

表示項目	説明	デフォルト値※
ドライバ JAR	対応する JDBC ドライバは事前に入手しておいてください。	<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>¥DB¥CLIENT¥UTL¥pdjdbc2.jar
ドライバ	使用する JDBC ドライバのパッケージ名（完全修飾 Java クラス名）を指定します。	JP.co.Hitachi.soft.HiRDB.JDBC.HiRDBDriver
URL	接続するデータベースのポート番号とホスト名を指定した，データベース接続 URL を指定します。	jdbc:hitachi:hirdb://DBID=22200,DBHOST=localhost
ユーザ名	データベースへ接続するときに使用するログインユーザ名を指定します。	admin
パスワード	データベースへ接続するときに使用するパスワードを指定します。	admin（画面上の表示は「****」）

注

［データベースコネクション］の入力フィールドの値は，SQL 文をデバッグするときにだけに使用されます。DB アダプタ定義支援機能で生成する SQL オペレーション定義ファイルおよびアプリケーション統合属性ファイルには，使用されません。

注※

デフォルト値は，［新規データソース］ダイアログ上の［デフォルト］ボタンをクリックした場合の値です。

初期値は，HCSC 簡易セットアップ機能を初期値で実行した場合に構築された環境の値です。

4. ［DB アダプタ設定］の入力フィールドに値を入力します。

ここでは，DB アダプタ用の SQL オペレーション定義ファイル，およびアプリケーション統合属性ファイルを定義するための設定をします。使用する DB Connector の設定に沿って指定してください。

表 C-2 ［DB アダプタ設定］の入力フィールドに定義する内容（［新規データソース］ダイアログ）

表示項目	説明	デフォルト値※
データベース参照名	データベース参照名を指定します。	DB_SERVER1
データベース種別	データベース種別を指定します。	HIRDB
リソースアダプタ名	DB Connector のプロパティ定義（display-name）に指定したリソースアダプタの名称を指定します。 リソースアダプタ名がドロップダウンリストから選択できない場合は，直接値を入力してください。	DBConnector_HiRDB_Type4_CP
インスタンスの最大値	DB アダプタのインスタンスの最大数を指定します。 無制限の場合は，0 を指定します。	0
インスタンスの最小値	DB アダプタのインスタンスの最小数を指定します。	1
DB Connector の共有	DB アダプタが参照する DB Connector を共有するかどうかを指定します。	Shareable
検索結果出力最大件数	SELECT で検索した結果の最大出力件数を指定します。	1000

表示項目	説明	デフォルト値※
検索結果出力最大件数	検索結果が指定値より大きい場合は，指定値までの件数が表示されます。 （例） 「1000」を指定した場合は，最大 1,000 レコードまでの結果を表示します。	1000
実行する SQL 文の最大個数	1 回のサービス部品実行要求で複数の SQL 文を実行する場合，実行する SQL 文の最大個数を指定します。	1024

注※

デフォルト値は，［新規データソース］ダイアログ上の［デフォルト］ボタンをクリックした場合の値です。

初期値は，HCSC 簡易セットアップ機能を初期値で実行した場合に構築された環境の値です。

注意事項

DB アダプタ設定で，DBMS として HiRDB を使用する場合，データベース種別とリソースアダプタ名が次の組み合わせになるように設定してください。

- データベース種別として HIRDB を使用する場合，リソースアダプタ名には DBConnector_DABJ_CP または DBConnector_DABJ_XA を設定してください。
- データベース種別として HIRDB-TYPE4 を使用する場合，リソースアダプタ名には DBConnector_HiRDB_Type4_CP または DBConnector_HiRDB_Type4_XA を設定してください。

5. ［OK］ボタンをクリックします。

設定した内容が登録されます。

(b)　本番環境用のデータベースをデバッグに使用する場合

デバッグ用のデータベースを使用しない場合の設定方法を次に示します。

1. Eclipse 上で DB アダプタの定義ファイルを開きます。

DB アダプタ開発支援画面が表示されます。

2. ［コネクション］タブにある［DataSources］を右クリックし，［新規データソース］を選択します。

［新規データソース］ダイアログが表示されます。

3. ［データベースコネクション］の［データベースなし］チェックボックスをチェックします。

4. ［DB アダプタ設定］に本番環境用のデータベースの設定内容を入力します。

5. ［OK］ボタンをクリックします。

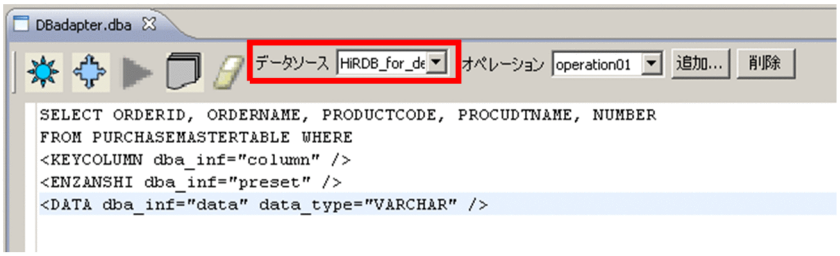
設定した内容が登録されます。

(5) SQL 文の作成

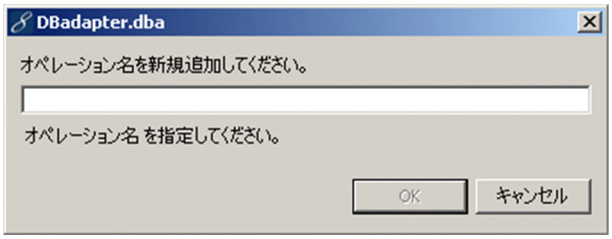
DB アダプタ上で実行する SQL 文を定義します。

1. Eclipse 上で DB アダプタの定義ファイルを開きます。
   DB アダプタ開発支援画面が表示されます。

2. [データソース] プルダウンリストから，使用するデータソースを選択します。



3. [オペレーション] プルダウンリストの横の [追加] ボタンをクリックします。
   追加するオペレーションの名称を定義するダイアログが表示されます。



4. 追加するオペレーション名を入力し，[OK] ボタンをクリックします。
   1 つの DB アダプタに複数の SQL を定義したい場合は，オペレーションを呼び出す SQL ごとに定義します。

5. SQL 文の入力フィールドに実行する SQL 文を定義します。
   引数要素の定義方法については，「●SQL オペレーション定義ファイルの形式」を参照してください。

注意事項

- 各オペレーションに記述できる SQL は 1 つだけです。
- ワイルドカード（*）は使用できません。SELECT で検索するカラム名は，すべて記述してください。
- SQL 文の最後にセミコロンは不要です。

(6) SQL のデバッグ

定義した SQL のデバッグを実施します。SQL 文中の引数要素については，値を代入して実行します。なお，SQL のデバッグは，本番環境用のデータベースとは別に，デバッグ用のデータベースを用意している場合にだけ実施できます。

次の手順で SQL のデバッグを実施してください。

1. デバッグ用のテーブルを作成します。

   事前にデバッグ用のデータベース上にデバッグで使用するテーブルを作成します。定義した SQL に沿ってテーブルを作成してください。

   作成例

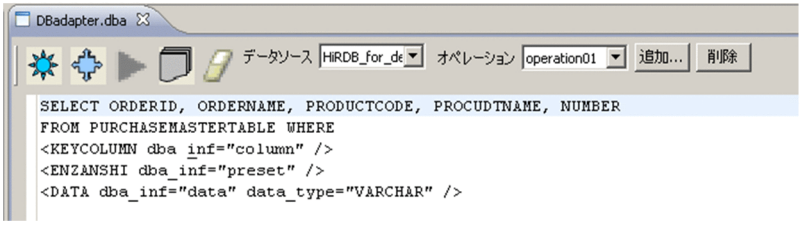
   テーブル名：PURCHASEMASTERTABLE

   テーブル構成：

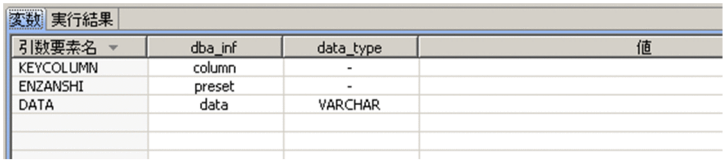
ORDERID (VARCHAR)	ORDERNAME (VARCHAR)	PRODUCTCODE (VARCHAR)	PRODUCTNAME (VARCHAR)	NUMBER (INTEGER)
01	AA001	0A1	NPW	4
02	AA002	0A1	NPW	3
03	AA003	0E1	NPW	3
04	BB001	0E1	NPW	4
05	BB002	0B1	NPW	5

2. Eclipse 上で DB アダプタの定義ファイルを開きます。

   DB アダプタ開発支援画面が表示されます。



3. 引数要素を抽出したい SQL 文を［オペレーション］プルダウンリストから選択した状態で，ツールバーの［変数の生成］アイコン（ ）［変数］タブに反映されます。

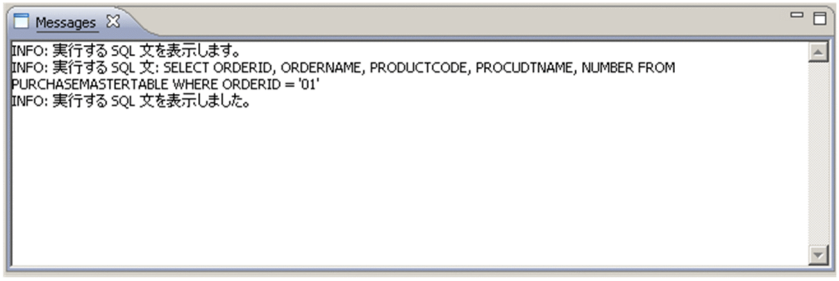


4. ［変数］タブの［値］列に，デバッグで使用したい値を入力します。

   設定例

引数要素名	DBa_inf	data_type	値
KEYCOLUMN	column	−	ORDERID
ENZANSHI	preset	−	=
DATA	data	VARCHAR	'01'

なお，デバッグ用のデータベースを用意していない場合は，以降の手順を実施できません。[実行する SQL 文の表示] アイコン（ ）をクリックすると，Messages ビューに引数要素に値を代入した状態の SQL が表示されるので，この画面で SQL 文が正しいかどうかを確認してください。



5. [コネクション] タブに表示されている，デバッグに使用するデータソースを右クリックし，[接続] を選択します。

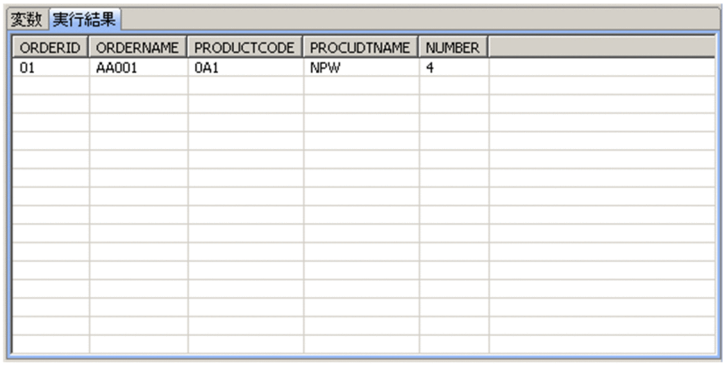
   設定した内容でデータベースに接続します。接続が成功すると，Messages ビューに，「INFO: データソース <データソース名> の接続に成功しました。」のメッセージが表示されます。

   なお，SQL 文が複数定義されている場合は，SQL 文ごとに手順 3～手順 5 を繰り返してください。

6. ツールバーにある [SQL の実行] アイコン（ ）をクリックします。

   SQL の実行に失敗した場合は，Messages ビューにエラーメッセージが表示されます。メッセージに従って SQL および引数要素名を修正してください。

   なお，SQL の実行結果は，[実行結果] タブに表示されます。



(7) 本番環境用のデータベースの設定

DB アダプタの定義ファイルのデータソースの設定を，本番環境用のデータベースの情報に変更します。

注意事項

本番環境用のデータベースの情報に変更する前に，デバッグで使用したデータリソースを切断しておいてください。データソースの切断方法は，次のとおりです。

1. デバッグで使用した DB アダプタの定義ファイルを Eclipse 上で開きます。

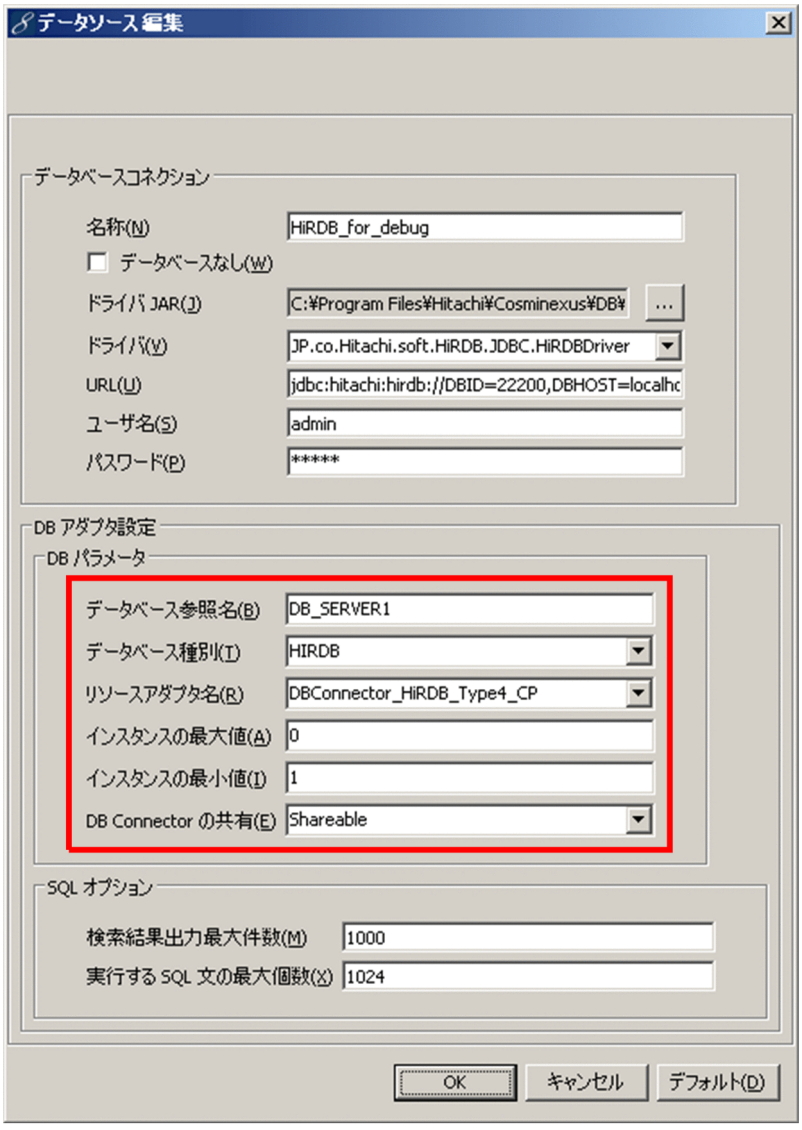
2. [コネクション] タブで，デバッグに使用したデータソースを右クリックし，[切断] を選択します。

次の手順でデータベースの情報を変更します。

1. デバッグで使用した DB アダプタの定義ファイルを Eclipse 上で開きます。
   DB アダプタ開発支援画面が表示されます。

2. [コネクション] タブに表示されているデータソースを右クリックして，[データソース編集] を選択します。
   [データソース編集] ダイアログが表示されます。

3. [DB アダプタ設定] の入力フィールドに本番環境用のデータベースの情報を入力します。



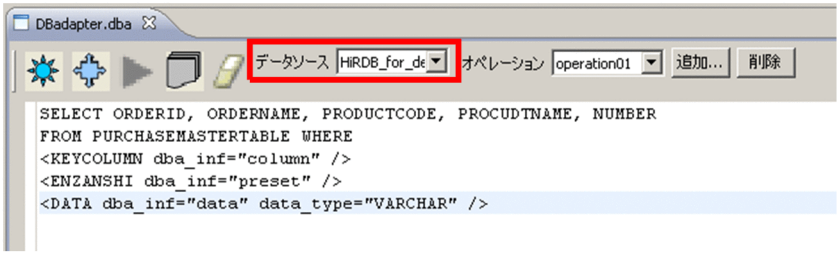
4. [OK] ボタンをクリックします。
   設定した内容が登録されます。

(8)　本番環境用の DB アダプタの定義ファイルの生成

本番環境で使用する DB アダプタの定義ファイルの生成方法を次に示します。

1. Eclipse 上で，デバッグ用の DB アダプタの定義ファイルを開きます。
   DB アダプタ開発支援画面が表示されます。

2. [データソース] プルダウンリストから，本番環境用の DB アダプタの定義ファイルに反映させたいデータソースを選択します。



3. ツールバーの［DB アダプタ定義生成］アイコン（  ）をクリックします。
   ［DB アダプタ定義生成］ダイアログが表示されます。



4. [出力先ディレクトリ] に，本番環境用の DB アダプタの定義ファイルを出力するディレクトリを指定します。
   [...] ボタンをクリックすると，［出力先ディレクトリ］に次のパスが選択されます。
   ＜ Eclipse のプロジェクトのパス＞¥＜ DB アダプタの定義ファイル名＞_Output

注意事項

- 出力先ディレクトリに Eclipse のプロジェクトのパスを指定した場合は，Eclipse の機能を使って，DB アダプタの定義ファイルおよび生成結果を一括でインポートおよびエクスポートできます。
- 出力先ディレクトリ名は，絶対パスを UTF-8 で変換したときに 155 バイト以内になるように指定してください。

5. 生成したいオペレーションのチェックボックスにチェックします。

1 つの DB アダプタに複数の SQL 文を定義した場合は，SQL の定義に必要なオペレーションをすべて選択してください。

6. [終了] ボタンをクリックします。

指定した出力先ディレクトリに，本番環境で使用する DB アダプタの定義ファイルが出力されます。

なお，出力先ディレクトリに Eclipse のプロジェクトのパスを指定している場合は，DB アダプタの定義ファイルを生成したあとに，出力先となる Eclipse のプロジェクトを更新してください。Eclipse のパッケージ・エクスプローラー上に，生成された DB アダプタの定義ファイルが表示されます。

生成されるファイルの一覧を次の表に示します。

表 C-3　生成されるファイルの一覧

ファイルの種別	ファイル名	数量	備考
SQL オペレーション定義ファイル	csa_sql_dbadapter.xml	1	－
アプリケーション統合属性ファイル	cscadapter_property.xml	1	－
要求電文の標準電文※1	＜オペレーション名＞_Input.xsd	0～N※2	＜オペレーション名＞は，指定したオペレーション名になります。
応答電文の標準電文※3	＜オペレーション名＞_Output.xsd	0～N※2	＜オペレーション名＞は，指定したオペレーション名になります。
要求電文および応答電文のサービス部品電文	ServiceMessageFile.xsd	1	要求電文および応答電文のサービス部品電文には，同じファイルを使用してください。

（凡例）

－：該当する説明はありません。

注※1

SQL 文中に引数要素がない場合，要求電文の標準電文は生成されません。サービス部品電文だけを使用してください。引数要素がない状態で DB アダプタを呼び出す場合は，Java 呼出アクティビティなどで要求電文を生成して，DB アダプタを呼び出します。

設定例

オペレーション名：select_op

実行する SQL 文：SELECT ORDERID FROM PURCHASEMASTERTABLE

Java 呼出アクティビティなどで生成する電文：

```
<?xml version="1.0" encoding="UTF-8"?>
<DBadapter>
    <DBA_MULTI_SQL>
        <select_op>
            <DBA_IN_DATA/>
        </select_op>
    </DBA_MULTI_SQL>
</DBadapter>
```

注※2

本番環境用の DB アダプタの定義ファイルを生成したときに指定したオペレーションの数になります。

注※3

応答電文の標準電文は，SQL 命令が SELECT の場合にだけ出力されます。INSERT の場合は，応答電文のサービス部品電文フォーマットを使用してください。

(9)　本番環境用の DB アダプタの設定

生成した本番環境用の DB アダプタの定義ファイルを使用して，DB アダプタを作成します。

(a)　DB アダプタの作成

Eclipse に組み込まれている HCSCTE を利用して，DB アダプタを新規に定義します。DB アダプタを新規に定義する方法については，「3.2.5　DB アダプタを新規に追加する」を参照してください。

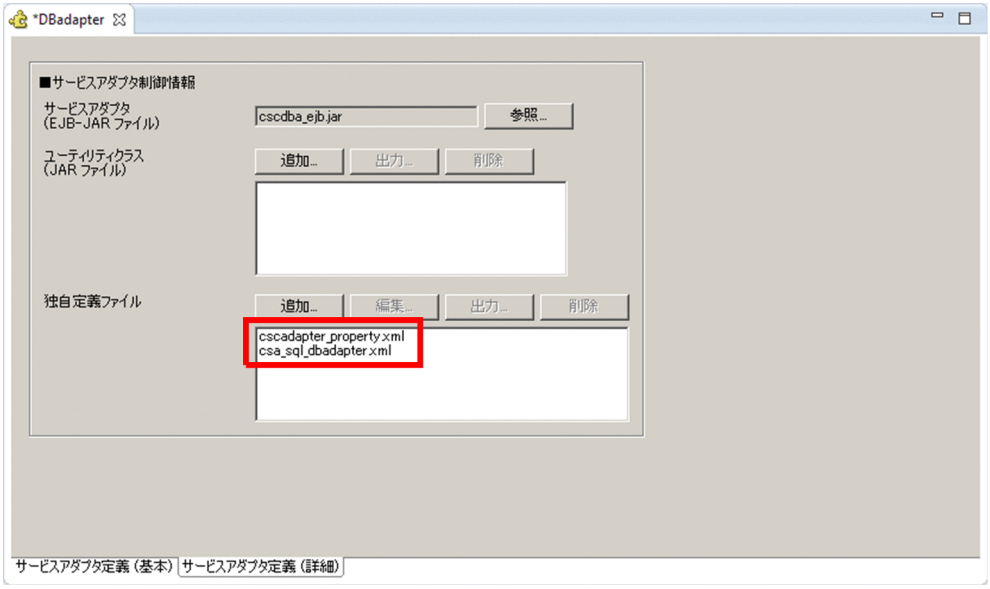
また，「(8)　本番環境用の DB アダプタの定義ファイルの生成」で指定したオペレーションを，作成した DB アダプタに追加してください。複数のオペレーションを指定した場合は，指定したオペレーションの数だけ追加してください。

(b)　SQL オペレーション定義ファイルとアプリケーション統合属性ファイルの設定

作成した DB アダプタのサービスアダプタ定義（詳細）画面を表示します。[独自定義ファイル] に次のファイルを追加してください。

- csa_sql_dbadapter.xml（生成した SQL オペレーション定義ファイル）
- cscadapter_property.xml（生成したアプリケーション統合属性ファイル）

図 C-6　DB アダプタに追加する独自定義ファイル



(c)　要求電文と応答電文の設定

作成した DB アダプタのサービスアダプタ定義（基本）画面を表示します。次の内容を設定してください。

オペレーション名の設定

- ［サービス部品制御情報］の［オペレーション］
  オペレーション名をドロップダウンリストから選択します。

オペレーションに対応する要求電文の設定

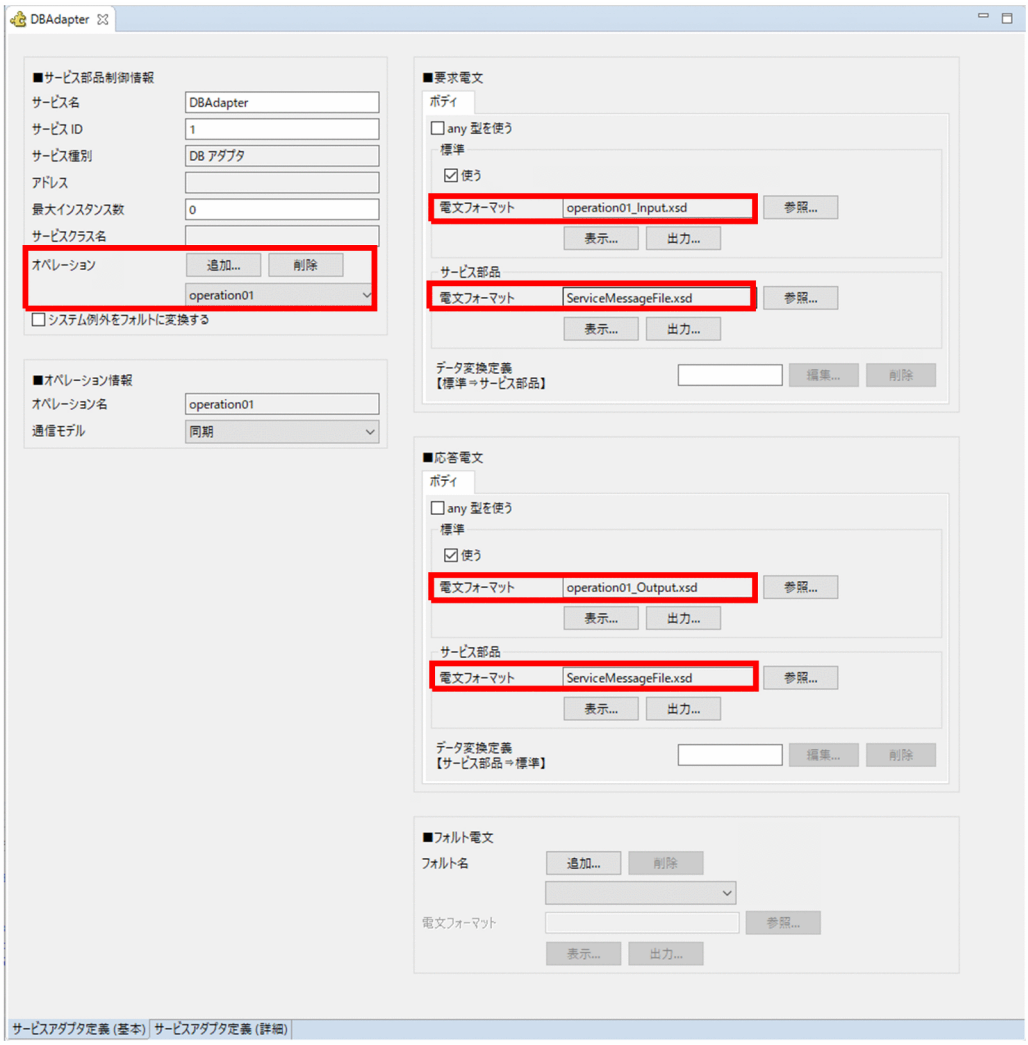
- ［要求電文］の［標準］の［電文フォーマット］
  「＜オペレーション名＞_Input.xsd」を指定します。
- ［要求電文］の［サービス部品］の［電文フォーマット］
  「ServiceMessageFile.xsd」を指定します。

オペレーションに対応する応答電文の設定

- ［応答電文］の［標準］の［電文フォーマット］
  「＜オペレーション名＞_Output.xsd」を指定します。ただし，指定したオペレーションに含まれる SQL 命令が SELECT の場合にだけ指定します。INSERT の場合は不要です。
- ［応答電文］の［サービス部品］の［電文フォーマット］
  「ServiceMessageFile.xsd」を指定します。

設定例を次の図に示します。

図 C-7　要求電文と応答電文の設定例



(d)　データ変換（マッピング）の定義

データ変換するときの変換元と変換先のマッピングを定義します。

- **要求電文のデータ変換（マッピング）の定義**

  要求電文の標準電文およびサービス部品電文のデータ変換には，自動マッピング機能を使用できます。自動マッピング機能の詳細については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.4　マッピングする」を参照してください。

- **応答電文のデータ変換（マッピング）の定義**

  応答電文の標準電文およびサービス部品電文のデータ変換は，手動でマッピングする必要があります。

  SELECT の検索結果を標準電文に変換する方法については，「3.3.5(5)　データ変換」を参照してください。

  INSERT の場合は，応答電文の標準電文フォーマットファイルが出力されないため，マッピングの必要はありません。INSERT の実行結果は，サービス部品側と同じ電文フォーマットの XML 電文に格納されます。

付録 C.3　分散開発時のエクスポート・インポート

複数の開発環境で DB アダプタを開発（分散開発）する場合は，マスタ開発環境以外の環境で開発した DB アダプタをエクスポートし，マスタ開発環境にインポートします。

DB アダプタ単位でエクスポートおよびインポートする方法と，DB アダプタの定義ファイル単位でエクスポートおよびインポートする方法について，次に説明します。

(1)　DB アダプタ単位のエクスポート・インポート

Eclipse に組み込まれている HCSCTE を利用して開発した場合，作成した DB アダプタはリポジトリに格納されます。このため，DB アダプタを分散開発した場合，一つのリポジトリとしてエクスポートおよびインポートすることができます。

エクスポート方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「3.2.2 リポジトリをエクスポートする」を参照してください。インポート方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「3.2.3　リポジトリをインポートする」を参照してください。

(2)　DB アダプタの定義ファイル単位のエクスポート・インポート

DB アダプタ用の各種定義ファイルの出力先に Eclipse のプロジェクトを指定している場合は，Eclipse の標準機能を使用して，DB アダプタの定義ファイルおよび生成結果を一括でエクスポートおよびインポートできます。

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

付録 D　ファイルアダプタの設定例

サンプルを利用してシステムを構築・運用する手順を説明します。

付録 D.1　サンプルについて

注意事項

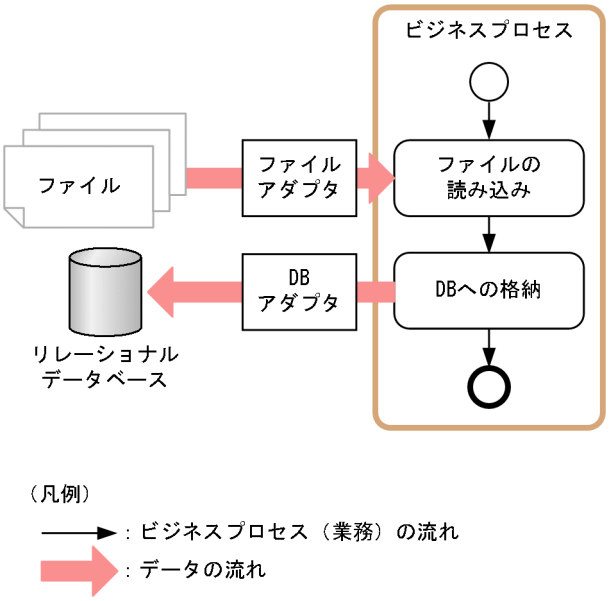
ここでは，開発環境内の運用・実行環境（テスト環境）でサンプルを実行することを前提として説明しています。

ここで使用するサンプルでは，ファイルアダプタと DB アダプタを利用して，ファイルから読み込んだデータの DB への格納，および DB 検索結果のファイルへの書き込みを行います。ファイルの読み込みおよび書き込みは，ファイル形式ごとに行います。

(1)　ファイルから読み込んだデータの DB への格納

処理の概要を次の図に示します。

図 D-1　サンプルの概要（ファイルから読み込んだデータの DB への格納）



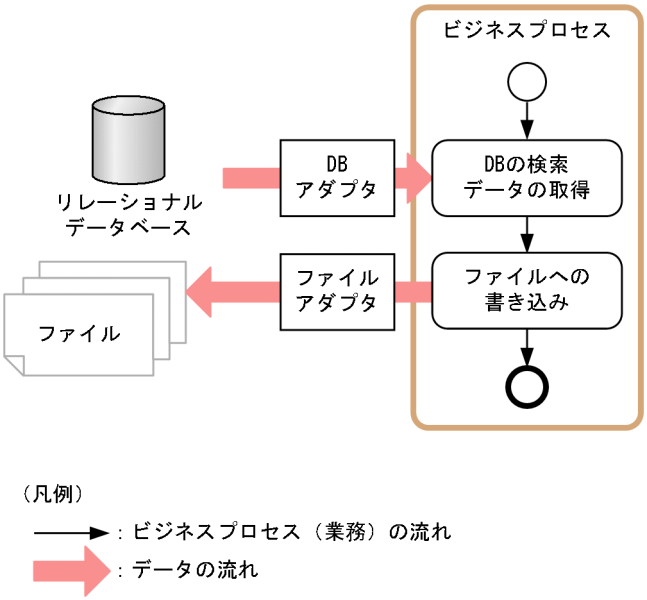
既存システムから FTP などで転送されたファイルから，ファイルアダプタによってデータを読み込み，DB アダプタを通じてデータベースに格納します。ビジネスプロセスはファイルの転送を契機に実行されるものとします。

サンプルは，ビジネスプロセス内の定義およびビジネスプロセスの実行（サービスリクエスタ）の範囲で提供されています。

(2)　DB 検索結果のファイルへの書き込み

処理の概要を次の図に示します。

図 D-2　サンプルの概要（DB 検索結果のファイルへの書き込み）



ビジネスプロセスを実行するサービスリクエスタからデータベースの検索条件を入力し，DB アダプタを通じてテーブルのデータを取得します。取得したデータはファイルアダプタを利用してファイルに書き込みます。

サンプルは，ビジネスプロセス内の定義およびビジネスプロセスの実行（サービスリクエスタ）の範囲で提供されています。

(3)　サンプルの一覧

提供するサンプルを次の表に示します。

表 D-1　サンプル一覧

サンプル	説明	ディレクトリ※1
XML データ読み込みサンプル	XML 形式のファイルから読み込んだ情報を DB へ格納します。	read_xml
CSV データ読み込みサンプル	CSV 形式のファイルから読み込んだ情報を DB へ格納します。	read_csv
固定長データ読み込みサンプル※2	固定長形式のファイルから読み込んだ情報を DB へ格納します。	read_fix
XML データ書き込みサンプル	DB 検索結果を XML 形式のファイルに書き込みます。	write_xml
CSV データ書き込みサンプル	DB 検索結果を CSV 形式のファイルに書き込みます。	write_csv

サンプル	説明	ディレクトリ※1
レングスタグデータ書き込みサンプル※3	DB 検索結果をレングスタグデータ形式のファイルに書き込みます。	write_len

注※1

サンプルで使用するファイルが格納されたディレクトリ名です。これらのディレクトリは，「<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¥custom-adapter¥File¥sample」の配下に格納されています。

注※2

CSV データの読み込みサンプルと同様の設定で使用できるため，このマニュアルでは説明していません。

注※3

CSV データの書き込みサンプルと同様の設定で使用できるため，このマニュアルでは説明していません。

(4)　サンプルの前提

サンプルを使用するための前提知識および前提環境を次に示します。

前提知識

次の内容を理解していることを前提とします。

- サービスプラットフォーム（Service Architect および Service Platform）を利用した，SOA に対応したシステム構築の知識
- SQL およびリレーショナルデータベースに関する基本的な知識

前提環境

次の製品のインストールおよび環境構築が完了していることを前提とします。

- Service Architect
- Service Platform
- HiRDB SQL Executer

サンプルでは，HCSC 簡易セットアップ機能で構築した環境に沿って説明しています。HCSC 簡易セットアップ機能の詳細は，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.4　テスト環境の簡易セットアップ」を参照してください。

なお，HCSC 簡易セットアップ機能を使用する場合，HCSC 簡易セットアップ画面（[サーバ名称] タブ）では［V7 互換名称］を選択してください。

付録 D.2　DB アダプタの設定

サンプルを使用するための DB アダプタの設定について説明します。

ここで示す設定は，1 回行えば，複数のサンプルで利用できます（サンプルごとに設定する必要はありません）。

(1) DB アダプタの定義

DB アダプタを定義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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   サービスアダプタ定義追加ウィザードが表示されます。

4. サービス部品の種別のドロップダウンリストから「DB アダプタ」を選択します。

5. ［次へ］ボタンをクリックします。
   サービスアダプタ定義追加（DB アダプタ）ウィザードが表示されます。

6. サービス名を入力します。
   サービス名には，「CSAFF_SAMPLE_DB」と入力します。

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

8. サービスアダプタ定義画面（基本）およびサービスアダプタ定義画面（詳細）を設定します。
   設定内容については，「DB アダプタの設定内容」を参照してください。

9. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

10. DB アダプタの設定を検証します。
    ツリービューのサービス定義一覧で右クリックして表示されるサービス一覧のポップアップメニューから，「検証」を選択すると，検証結果がコンソールビューに表示されます。

DB アダプタの設定内容

サンプルの DB アダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 D-2　サービスアダプタ定義画面（基本）の設定内容（DB アダプタ）

分類	項目	設定値	設定
サービス部品制御情報	サービス名	CSAFF_SAMPLE_DB	◎
	サービス ID	CSAFF_DB	◎
	サービス種別	DB アダプタ	×

分類	項目		設定値	設定
サービス部品制御情報	最大インスタンス数		0	◎
	オペレーション		CSAFF_SAMPLE_DB	◎
オペレーション情報	オペレーション名		CSAFF_SAMPLE_DB	×
	通信モデル		同期	◎
要求電文	標準	使う（チェックボックス）	使わない（チェックしない）	◎
	サービス部品	電文フォーマット	csaff_sample_sqlformat.xsd	◎
応答電文	標準	使う（チェックボックス）	使わない（チェックしない）	◎
	サービス部品	電文フォーマット	csaff_sample_sqlformat.xsd	◎

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

サンプルの DB アダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 D-3　サービスアダプタ定義画面（詳細）の設定内容（DB アダプタ）

分類	項目	設定値	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	cscdba_ejb.jar	×
	独自定義ファイル	csa_sql_csaff_sample.xml	◎
		cscadapter_property.xml	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

(2)　DB アダプタの配備定義

定義した DB アダプタをサーバに配備して開始します。

DB アダプタの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，［複数サービスをサーバに配備して開始］を選択します。

2. サービス選択ページで配備および開始するサービスを選択します。

3. [Finish] 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。

4. [OK] ボタンをクリックします。
   DB アダプタの配備と開始が完了します。

付録 D.3　XML データ読み込みサンプル

XML データ読み込みサンプルでは，XML 形式のファイルからファイルアダプタによってデータを読み込み，DB アダプタを通してデータベースに格納します。

入出力項目を次の表に示します。

表 D-4　入出力項目一覧（XML データ読み込みサンプル）

入出力項目	内容
ファイルアダプタの入力ファイル	<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¥custom-adapter¥File¥sample¥read_xml¥read_xml_data.xml
DB アダプタの出力テーブル	ADMIN.CSAFF_SAMPLE_RX※

注※
スキーマ名は変更しないでください。

(1)　ファイルアダプタの定義

(a)　ファイルアダプタの追加と検証

ファイルアダプタを追加設定し，検証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. [HCSC-Definer] － [HCSCTE ビュー] を選択し，[開く] ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   サービスアダプタ定義追加ウィザードが表示されます。

4. サービス部品の種別のドロップダウンリストから「ファイルアダプタ」を選択します。

5. [次へ] ボタンをクリックします。
   サービスアダプタ定義追加（ファイルアダプタ）ウィザードが表示されます。

6. サービス名を入力します。

サービス名には，「CSAFF_SAMPLE_READ_XML」と入力します。

7. ［終了］ボタンをクリックします。

必要なファイルが作成され，リポジトリに保存されます。サービスアダプタ定義画面（基本）が表示されます。

8. サービスアダプタ定義画面（基本）およびサービスアダプタ定義画面（詳細）を設定します。

設定内容については，「ファイルアダプタの設定内容（XML データ読み込みサンプル）」を参照してください。

9. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

10. ファイルアダプタの設定を検証します。

ツリービューのサービス定義一覧で右クリックして表示されるサービス一覧のポップアップメニューから，「検証」を選択すると，検証結果がコンソールビューに表示されます。

ファイルアダプタの設定内容（XML データ読み込みサンプル）

XML データ読み込みサンプルのファイルアダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 D-5　サービスアダプタ定義画面（基本）の設定内容（XML データ読み込みサンプル）

分類	項目		設定値	設定
サービス部品制御情報	サービス名		CSAFF_SAMPLE_READ_XML	◎
	サービス ID		CSAFF_RX	◎
	サービス種別		ファイルアダプタ	×
	最大インスタンス数		0	◎
	オペレーション		定義パターン：XML 読み込み オペレーション名：CSAFF_SAMPLE_READ_XML	◎
オペレーション情報	オペレーション名		CSAFF_SAMPLE_READ_XML	×
	通信モデル		同期	◎
要求電文	標準	使う（チェックボックス）	使わない（チェックしない）	◎
	サービス部品	電文フォーマット	adpff_read.xsd	×
応答電文	標準	使う（チェックボックス）	使わない（チェックしない）	◎
	サービス部品	電文フォーマット	csaff_sample_read_xml.xsd	◎

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

XML データ読み込みサンプルのファイルアダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 D-6　サービスアダプタ定義画面（詳細）の設定内容（XML データ読み込みサンプル）

分類	項目	設定値	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	cscmsg_adpejb.jar	×
	ユーティリティクラス（JAR ファイル）	adpffpc.jar	×

（凡例）

×：表示されている内容を確認します。

(b)　ファイルアダプタ実行環境プロパティファイルの作成

ファイルアダプタ実行環境プロパティファイルを作成する手順を次に示します。

1. サンプルで提供するファイルアダプタ実行環境プロパティファイルを，テキストエディタで開きます。

   サンプルで提供するファイルアダプタ実行環境プロパティファイルは，「<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¥custom-adapter¥File¥sample¥read_xml¥CSAFF_RX.properties」です。

2. 次の内容を記述します。

```
fileaccess.path1 = <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¥¥custom-adapter¥¥File¥¥sample¥¥read_xml
```

注意事項

ディレクトリの区切り文字は，「¥¥」で記述します。

3. ファイルを保存し，「<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¥custom-adapter¥File¥config」に格納します。

(2)　ビジネスプロセスの定義

ビジネスプロセスの定義は，ビジネスプロセスの追加，ビジネスプロセスの内容の定義，ビジネスプロセスの検証の順で行います。

(a)　ビジネスプロセスの追加

ビジネスプロセスを追加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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「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。

4. ビジネスプロセス名を入力し，ステータスを永続化するかどうか（ビジネスプロセスの実行状態をデータベースに保存するかどうか）を選択します。
   ビジネスプロセス名には，「CSAFF_SAMPLE_BP_READ_XML」と入力します。
   ステータスの永続化には，「yes」を選択します。

   参考

   BPEL ファイルは使用しません。

5. ［終了］ボタンをクリックします。
   必要なファイルが作成され，リポジトリに保存されます。ビジネスプロセス定義画面が表示されます。

(b)　変数の設定

次に示す変数を設定します。

- CSAFF_DB_RequestMessage
- CSAFF_DB_ResponseMessage
- CSAFF_RequestMessage
- CSAFF_ResponseMessage

参考

サンプルでは相関セットを使用しません。

変数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. 一覧から「変数一覧」を選択します。

3. 変数名を入力します。

4. ［種別］のドロップダウンリストから「XML」を選択します。

5. 次のどちらかの操作をします。

CSAFF_DB_ResponseMessage，CSAFF_RequestMessage または CSAFF_ResponseMessage の場合

［取込］ボタンをクリックして表示される［電文フォーマットの取込］ダイアログで，使用する電文フォーマットを指定します。設定内容については，「変数の設定内容（XML データ読み込みサンプル）」を参照してください。

CSAFF_DB_RequestMessage の場合

［...］ボタンをクリックして，「csaff_sample_read_xml_dt.xsd」を設定します。

6. ［追加］ボタンをクリックします。

変数一覧に，追加した変数が表示されます。

7. 変数ごとに手順 2 から 6 を繰り返します。

8. ［OK］ボタンをクリックします。

変数の設定値が保存されて，［変数・相関セット一覧］ダイアログが閉じます。

変数の設定内容（XML データ読み込みサンプル）

CSAFF_DB_ResponseMessage，CSAFF_RequestMessage および CSAFF_ResponseMessage の設定内容を次に示します。なお，表に記載されていない項目については，設定不要です。

表 D-7　CSAFF_DB_ResponseMessage の設定内容（XML データ読み込みサンプル）

分類	項目	設定値
サービス/受付	サービス名	CSAFF_SAMPLE_DB
	受付名	－
取込対象	オペレーション名	CSAFF_SAMPLE_DB
	電文種別	応答電文（ボディ）
－	電文フォーマット	csaff_sample_sqlformat

（凡例）

－：該当しません。

表 D-8　CSAFF_RequestMessage の設定内容（XML データ読み込みサンプル）

分類	項目	設定値
サービス/受付	サービス名	CSAFF_SAMPLE_READ_XML
	受付名	－
取込対象	オペレーション名	CSAFF_SAMPLE_READ_XML
	電文種別	要求電文（ボディ）
－	電文フォーマット	adpff_read

（凡例）

－：該当しません。

表 D-9　CSAFF_ResponseMessage の設定内容（XML データ読み込みサンプル）

分類	項目	設定値
サービス/受付	サービス名	CSAFF_SAMPLE_READ_XML
	受付名	－
取込対象	オペレーション名	CSAFF_SAMPLE_READ_XML
	電文種別	応答電文（ボディ）
－	電文フォーマット	csaff_sample_read_xml

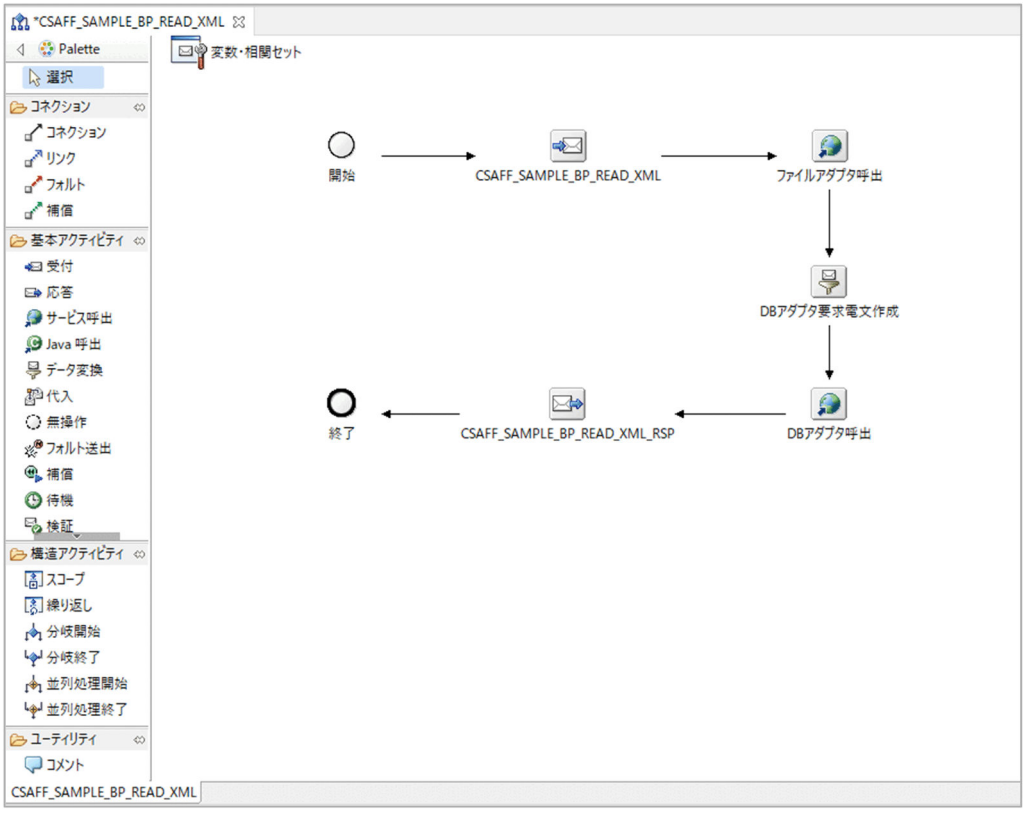
（凡例）

－：該当しません。

(c)　アクティビティの設定

ここでは，次の図に示すアクティビティの配置・定義，データ変換定義，およびアクティビティの連結について説明します。

図 D-3　ビジネスプロセス定義全体図（XML データ読み込みサンプル）



●アクティビティの配置・定義

XML データ読み込みサンプルでは，次の表に示すアクティビティを設定します。

表 D-10　アクティビティ一覧（XML データ読み込みサンプル）

アクティビティ	アクティビティ名	説明
受付	CSAFF_SAMPLE_BP_READ_XML	ビジネスプロセスの受付。
応答	CSAFF_SAMPLE_BP_READ_XML_RSP	ビジネスプロセスの応答。
サービス呼出	ファイルアダプタ呼出	ファイルアダプタの呼び出し。
サービス呼出	DB アダプタ呼出	DB アダプタの呼び出し。
データ変換	DB アダプタ要求電文作成	ファイルアダプタの応答電文を DB アダプタの要求電文に変換。

アクティビティを配置して設定する手順を次に示します。

1. パレットの基本アクティビティまたは構造アクティビティの中から，キャンバスに配置するアクティビティをクリックします。
   クリックしたアクティビティが選択された状態になります。
2. キャンバス上の適当な場所をクリックします。
   選択したアクティビティがキャンバス上に配置されます。配置されたアクティビティは，ドラッグ＆ドロップで場所を移動できます。
3. 配置されたアクティビティをダブルクリックします。
   ダブルクリックしたアクティビティのダイアログが表示されます。
4. ダイアログに必要な情報を入力します。
   設定内容の詳細は，「アクティビティの設定内容（XML データ読み込みサンプル）」を参照してください。
5. [OK] ボタンをクリックします。
   アクティビティのダイアログが閉じます。
6. アクティビティごとに手順 1 から 5 を繰り返します。

アクティビティの設定内容（XML データ読み込みサンプル）

XML データ読み込みサンプルで使用するビジネスプロセスのアクティビティの設定内容を次に示します。なお，表に記載されていない項目については，設定不要です。

表 D-11　CSAFF_SAMPLE_BP_READ_XML の設定内容

項目	設定値
アクティビティ名	CSAFF_SAMPLE_BP_READ_XML
オペレーション名	CSAFF_SAMPLE_BP_READ_XML
ボディ割当変数	CSAFF_RequestMessage
通信モデル	同期
インスタンス生成	Yes

表 D–12　CSAFF_SAMPLE_BP_READ_XML_RSP の設定内容

項目	設定値
アクティビティ名	CSAFF_SAMPLE_BP_READ_XML_RSP
オペレーション名	CSAFF_SAMPLE_BP_READ_XML
ボディ割当変数	CSAFF_DB_ResponseMessage

表 D–13　ファイルアダプタ呼出の設定内容（XML データ読み込みサンプル）

項目		設定値
アクティビティ名		ファイルアダプタ呼出
サービス名		CSAFF_SAMPLE_READ_XML
オペレーション名		CSAFF_SAMPLE_READ_XML
通信モデル		同期
要求電文	ボディ割当変数	CSAFF_RequestMessage
応答電文	ボディ割当変数	CSAFF_ResponseMessage

表 D–14　DB アダプタ呼出の設定内容（XML データ読み込みサンプル）

項目		設定値
アクティビティ名		DB アダプタ呼出
サービス名		CSAFF_SAMPLE_DB
オペレーション名		CSAFF_SAMPLE_DB
通信モデル		同期
要求電文	ボディ割当変数	CSAFF_DB_RequestMessage
応答電文	ボディ割当変数	CSAFF_DB_ResponseMessage

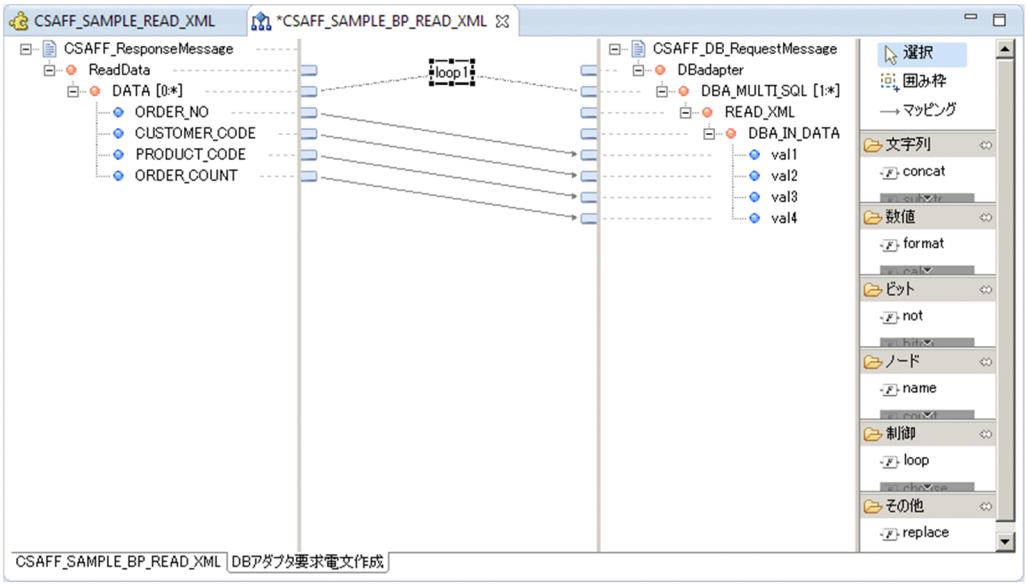
表 D–15　DB アダプタ要求電文作成の設定内容（XML データ読み込みサンプル）

項目	設定値
アクティビティ名	DB アダプタ要求電文作成
変換元変数	CSAFF_ResponseMessage
変換先変数	CSAFF_DB_RequestMessage

●DB アダプタ要求電文作成アクティビティのデータ変換定義の設定

DB アダプタ要求電文作成アクティビティのデータ変換定義を設定します。設定するデータ変換定義を次の図に示します。

図 D-4　DB アダプタ要求電文作成アクティビティデータ変換定義画面（XML データ読み込みサンプル）



DB アダプタ要求電文作成アクティビティのデータ変換定義の手順を次に示します。

1. DB アダプタ要求電文作成アクティビティからデータ変換（マッピング）定義画面を表示します。
   データ変換（マッピング）定義画面が表示されたあと，［ルート要素選択］ダイアログが表示されます。
2. すべてのスキーマ論理名のルート要素を選択します。
   ［OK］ボタンがクリックできる状態になります。
3. ［OK］ボタンをクリックします。
   変換元スキーマツリービューアおよび変換先スキーマツリービューアに，選択したルート要素のスキーマがツリー表示されます。
4. マッピングします。
   マッピングでは次の作業を行います。
   ・ファンクションの配置とファンクションの設定
   　ファンクションの設定内容を次の表に示します。

表 D-16　ファンクションの設定内容（XML データ読み込みサンプル）

ファンクション	ファンクション名	設定値	
loop	loop1	ノード選択	DATA
		ノード条件設定	設定しない

   ・ノード間のマッピングの設定
   　ノード間のマッピングの設定内容を次の表に示します。

表 D-17　マッピングの設定内容（XML データ読み込みサンプル）

変換元ノード	ファンクション	変換先ノード
−	loop1	DBA_MULTI_SQL
ORDER_NO	−	val1
CUSTOMER_CODE	−	val2
PRODUCT_CODE	−	val3
ORDER_COUNT	−	val4

（凡例）

−：該当しません。

5. データ変換（マッピング）定義画面の変換元スキーマツリービューア，マッピングビューア，または変換先スキーマツリービューアの適当な場所で右クリックして，［検証］を選択します。

   検証が実行されます。なお，警告は無視してください。

●アクティビティの連結

XML データ読み込みサンプルで使用するビジネスプロセスでは，次の表に示すようにアクティビティを連結します。

表 D-18　アクティビティの連結内容（XML データ読み込みサンプル）

連結元アクティビティ	連結先アクティビティ
開始	CSAFF_SAMPLE_BP_READ_XML
CSAFF_SAMPLE_BP_READ_XML	ファイルアダプタ呼出
ファイルアダプタ呼出	DB アダプタ要求電文作成
DB アダプタ要求電文作成	DB アダプタ呼出
DB アダプタ呼出	CSAFF_SAMPLE_BP_READ_XML_RSP
CSAFF_SAMPLE_BP_READ_XML_RSP	終了

アクティビティの連結の手順を次に示します。

1. パレットの［コネクション］をクリックします。

   コネクションが選択された状態になります。

   この状態で，アクティビティにマウスポインタを合わせると，そのアクティビティを連結元に設定できる場合，アクティビティの背景色が変わります。

2. キャンバスに配置したアクティビティのうち，連結元となるアクティビティをクリックします。

   連結元となるアクティビティが設定されます。

   この状態で，ほかのアクティビティにマウスポインタを合わせると，そのアクティビティを連結先に設定できる場合，アクティビティの背景色が変わります。

3. キャンバスに配置したアクティビティのうち，連結先となるアクティビティをクリックします。

   連結元となるアクティビティと連結先となるアクティビティが連結されます。

(d)　ビジネスプロセスの保存・検証

ビジネスプロセスを保存して検証する手順を次に示します。

1. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。
   ビジネスプロセスのサービス ID は，プロパティビューで「CSAFF_B1」を設定してください。

2. ビジネスプロセスの設定を検証します。
   ツリービューのサービス定義一覧で右クリックして表示されるサービス一覧のポップアップメニューから，「検証」を選択すると，検証結果がコンソールビューに表示されます。

(3)　ファイルアダプタとビジネスプロセスの配備定義

定義したファイルアダプタとビジネスプロセスをサーバに配備して開始します。

ファイルアダプタとビジネスプロセスの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，［複数サービスをサーバに配備して開始］を選択します。

2. サービス選択ページで配備および開始するサービスを選択します。

3. [Finish] 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。

4. [OK] ボタンをクリックします。
   ファイルアダプタとビジネスプロセスの配備と開始が完了します。

(4)　テーブルの作成

サンプルを実行するデータベースのテーブルを作成します。

HiRDB SQL Executer を起動して，次の SQL 文を実行してください。

```
CREATE TABLE CSAFF_SAMPLE_RX
(ORDER_NO INT,
 CUSTOMER_CODE CHAR(5),
 PRODUCT_CODE CHAR(4),
 ORDER_COUNT INT);
```

注意事項

ユーザ ID は「ADMIN」とします。

(5) 要求電文の作成

サンプルで提供するビジネスプロセス実行時の要求電文を編集して，要求電文を作成する手順を次に示します。

1. サンプルで提供する要求電文ファイルを，テキストエディタで開きます。
   サンプルで提供する要求電文ファイルは，「<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¥custom-adapter¥File¥sample¥read_xml¥read_xml_message.xml」です。

2. 次に示す斜体部分を記述します。

```
<?xml version="1.0" encoding="UTF-8"?>
<adpff:ReadRequest xmlns:adpff="http://FlatFiles.cstmadp.csc.soft.Hitachi.co.jp/ReadRequest">
  <fileheader>
    <filename><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¥custom-adapter¥File¥sample¥read_xml¥read_xml_data.xml</filename>
    <filemode>all</filemode>
  </fileheader>
</adpff:ReadRequest>
```

3. ファイルを上書き保存します。

(6) ビジネスプロセスの実行

サンプルを実行するには，サービスリクエスタからビジネスプロセスを起動します。

サービスリクエスタの処理の流れと概要を次に示します。

1. サービスリクエスタ実行
   ビジネスプロセスの要求電文が，そのままファイルアダプタの要求電文として使用されます。
2. ファイルアダプタ実行
   要求電文で指定されたファイルが読み込まれます。
3. DB アダプタ電文作成
   ファイルアダプタの応答電文が DB アダプタの要求電文の形式に変換されます。
4. DB アダプタ実行
   要求電文で指定されたファイルがデータベースに登録されます。
5. サービスリクエスタ終了
   DB アダプタの応答電文が，そのままビジネスプロセスの応答電文として使用されます。

サービスリクエスタの実行手順を次に示します。

1. カレントディレクトリを「<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¥custom-adapter¥File¥sample¥bin」に移動します。

2. request.bat を実行します。

次の形式で実行してください。

```
request read_xml <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¥custom-adapter¥File¥sample¥read_xml¥read_xml_message.xml
```

参考

HiRDB SQL Executer を起動して次の SQL 文を実行することで，テーブルの内容が確認できます（ユーザ ID は「ADMIN」とします）。

```
SELECT * FROM CSAFF_SAMPLE_RX;
```

付録 D.4　CSV データ読み込みサンプル

CSV データ読み込みサンプルでは，CSV 形式のファイルからファイルアダプタによってデータを読み込み，DB アダプタを通してデータベースに格納します。このとき，入力ファイルの製品コードが「1001」である製品だけを取得してデータベースに格納します。製品コードの検索はデータ変換機能で実現します。

入出力項目を次の表に示します。

表 D-19　入出力項目一覧（CSV データ読み込みサンプル）

入出力項目	内容
ファイルアダプタの入力ファイル	<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¥custom-adapter¥File¥sample¥read_csv¥read_csv_data.csv
DB アダプタの出力テーブル	ADMIN.CSAFF_SAMPLE_RC※

注※

スキーマ名は変更しないでください。

(1)　ファイルアダプタの定義

(a)　ファイルアダプタの追加と検証

ファイルアダプタを追加設定し，検証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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。
3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   サービスアダプタ定義追加ウィザードが表示されます。

4. サービス部品の種別のドロップダウンリストから「ファイルアダプタ」を選択します。

5. [次へ] ボタンをクリックします。
   サービスアダプタ定義追加（ファイルアダプタ）ウィザードが表示されます。

6. サービス名を入力します。
   サービス名には，「CSAFF_SAMPLE_READ_CSV」と入力します。

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

8. サービスアダプタ定義画面（基本）およびサービスアダプタ定義画面（詳細）を設定します。
   設定内容については，「ファイルアダプタの設定内容（CSV データ読み込みサンプル）」を参照してください。

9. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

10. ファイルアダプタの設定を検証します。
    ツリービューのサービス定義一覧で右クリックして表示されるサービス一覧のポップアップメニューから，「検証」を選択すると，検証結果がコンソールビューに表示されます。

ファイルアダプタの設定内容（CSV データ読み込みサンプル）

CSV データ読み込みサンプルのファイルアダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 D-20　サービスアダプタ定義画面（基本）の設定内容（CSV データ読み込みサンプル）

分類	項目		設定値	設定
サービス部品制御情報	サービス名		CSAFF_SAMPLE_READ_CSV	◎
	サービス ID		CSAFF_RC	◎
	サービス種別		ファイルアダプタ	×
	最大インスタンス数		0	◎
	オペレーション		定義パターン：バイナリ読み込み オペレーション名：CSAFF_SAMPLE_READ_CSV	◎
オペレーション情報	オペレーション名		CSAFF_SAMPLE_READ_CSV	×
	通信モデル		同期	◎
要求電文	標準	使う（チェックボックス）	使わない（チェックしない）	◎
	サービス部品	電文フォーマット	adpff_read.xsd	×

分類	項目		設定値	設定
応答電文	標準	使う（チェックボックス）	使う（チェックする）	◎
		電文フォーマット	csaff_sample_read_csv_standard.xsd	◎
	サービス部品	電文フォーマット	csaff_sample_read_csv.fdx	◎
	データ変換定義		SAMPLE_READ_CSV_DT	◎

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

CSV データ読み込みサンプルのファイルアダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 D-21　サービスアダプタ定義画面（詳細）の設定内容（CSV データ読み込みサンプル）

分類	項目	設定値	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	cscmsg_adpejb.jar	×
	ユーティリティクラス（JAR ファイル）	adpffpc.jar	×

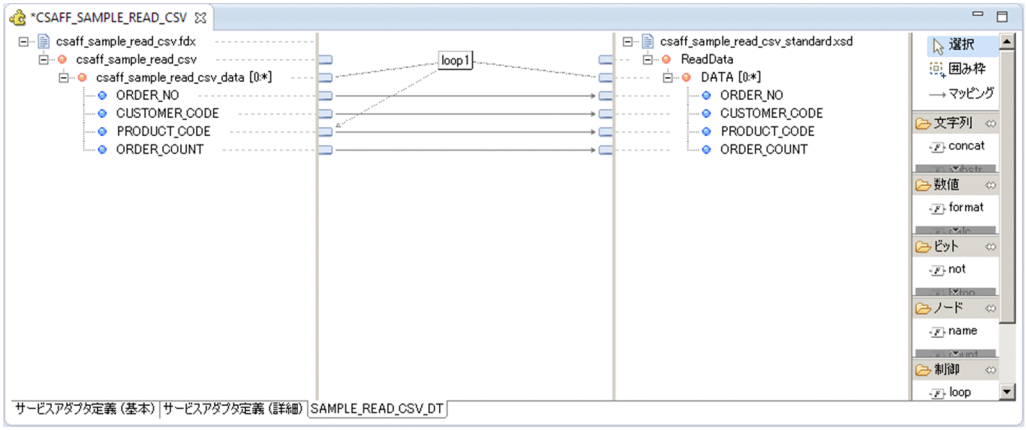
（凡例）

×：表示されている内容を確認します。

(b)　データ変換定義

データ変換定義を設定します。設定するデータ変換定義を次の図に示します。

図 D-5　ファイルアダプタのデータ変換定義画面（CSV データ読み込みサンプル）



データ変換定義の手順を次に示します。

1. アダプタ定義画面のデータ変換（マッピング）定義画面を表示します。

   データ変換（マッピング）定義画面が表示されたあと，［ルート要素選択］ダイアログが表示されます。

2. すべてのスキーマ論理名のルート要素を選択します。

［OK］ボタンがクリックできる状態になります。

3. [OK] ボタンをクリックします。

変換元スキーマツリービューアおよび変換先スキーマツリービューアに，選択したルート要素のスキーマがツリー表示されます。

4. マッピングします。

マッピングでは次の作業を行います。

- ファンクションの配置とファンクションの設定

  ファンクションの設定内容を次に示します。

表 D-22　ファンクションの設定内容（CSV データ読み込みサンプル-ファイルアダプタ）

ファンクション	ファンクション名	設定値	
loop	loop1	ノード選択	csaff_sample_read_csv_data
		ノード条件設定	下表参照

表 D-23　loop1 の設定内容

分類		設定値
ノード		csaff_sample_read_csv_data
条件	条件	条件式
	左辺	ノード・ファンクション（PRODUCT_CODE）
	等符号	＝
	右辺	値（1001）

- ノード間のマッピングの設定

  ノード間のマッピングの設定内容を次の表に示します。

表 D-24　マッピングの設定内容（CSV データ読み込みサンプル-ファイルアダプタ）

変換元ノード	ファンクション	変換先ノード
－	loop1	DATA
ORDER_NO	－	ORDER_NO
CUSTOMER_CODE	－	CUSTOMER_CODE
PRODUCT_CODE	－	PRODUCT_CODE
ORDER_COUNT	－	ORDER_COUNT

（凡例）

－：該当しません。

5. データ変換（マッピング）定義画面の変換元スキーマツリービューア，マッピングビューア，または変換先スキーマツリービューアの適当な場所で右クリックして，［検証］を選択します。
   検証が実行されます。

(c)　ファイルアダプタ実行環境プロパティファイルの作成

ファイルアダプタ実行環境プロパティファイルを作成する手順を次に示します。

1. サンプルで提供するファイルアダプタ実行環境プロパティファイルを，テキストエディタで開きます。
   サンプルで提供するファイルアダプタ実行環境プロパティファイルは，「<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¥custom-adapter¥File¥sample¥read_csv¥CSAFF_RC.properties」です。

2. 次の内容を記述します。

```
fileaccess.path1 = <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¥¥custom-adapter¥¥File¥¥sample¥¥read_csv
```

注意事項

ディレクトリの区切り文字は，「¥¥」で記述します。

3. ファイルを保存し，「<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¥custom-adapter¥File¥config」に格納します。

(2)　ビジネスプロセスの定義

ビジネスプロセスの定義は，ビジネスプロセスの追加，ビジネスプロセスの内容の定義，ビジネスプロセスの検証の順で行います。

(a)　ビジネスプロセスの追加

ビジネスプロセスを追加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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「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。

4. ビジネスプロセス名を入力し，ステータスを永続化するかどうか（ビジネスプロセスの実行状態をデータベースに保存するかどうか）を選択します。
   ビジネスプロセス名には，「CSAFF_SAMPLE_BP_READ_CSV」と入力します。

ステータスの永続化には，「yes」を選択します。

参考

BPEL ファイルは使用しません。

5. ［終了］ボタンをクリックします。
必要なファイルが作成され，リポジトリに保存されます。ビジネスプロセス定義画面が表示されます。

(b)　変数の設定

次に示す変数を設定します。

- CSAFF_DB_RequestMessage
- CSAFF_DB_ResponseMessage
- CSAFF_RequestMessage
- CSAFF_ResponseMessage

参考

サンプルでは相関セットを使用しません。

変数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. 一覧から「変数一覧」を選択します。

3. 変数名を入力します。

4. ［種別］のドロップダウンリストから「XML」を選択します。

5. 次のどちらかの操作をします。

CSAFF_DB_ResponseMessage，CSAFF_RequestMessage または CSAFF_ResponseMessage の場合

［取込］ボタンをクリックして表示される［電文フォーマットの取込］ダイアログで，使用する電文フォーマットを指定します。設定内容については，「変数の設定内容（CSV データ読み込みサンプル）」を参照してください。

CSAFF_DB_RequestMessage の場合

［...］ボタンをクリックして，「csaff_sample_read_csv_dt.xsd」を設定します。

6. ［追加］ボタンをクリックします。

変数一覧に，追加した変数が表示されます。

7. 変数ごとに手順 2 から 6 を繰り返します。

8. [OK] ボタンをクリックします。

変数の設定値が保存されて，［変数・相関セット一覧］ダイアログが閉じます。

変数の設定内容（CSV データ読み込みサンプル）

CSAFF_DB_ResponseMessage，CSAFF_RequestMessage および CSAFF_ResponseMessage の設定内容を次に示します。なお，表に記載されていない項目については，設定不要です。

表 D–25　CSAFF_DB_ResponseMessage の設定内容（CSV データ読み込みサンプル）

分類	項目	設定値
サービス/受付	サービス名	CSAFF_SAMPLE_DB
	受付名	－
取込対象	オペレーション名	CSAFF_SAMPLE_DB
	電文種別	応答電文（ボディ）
－	電文フォーマット	csaff_sample_sqlformat

（凡例）

－：該当しません。

表 D–26　CSAFF_RequestMessage の設定内容（CSV データ読み込みサンプル）

分類	項目	設定値
サービス/受付	サービス名	CSAFF_SAMPLE_READ_CSV
	受付名	－
取込対象	オペレーション名	CSAFF_SAMPLE_READ_CSV
	電文種別	要求電文（ボディ）
－	電文フォーマット	adpff_read

（凡例）

－：該当しません。

表 D–27　CSAFF_ResponseMessage の設定内容（CSV データ読み込みサンプル）

分類	項目	設定値
サービス/受付	サービス名	CSAFF_SAMPLE_READ_CSV
	受付名	－
取込対象	オペレーション名	CSAFF_SAMPLE_READ_CSV
	電文種別	応答電文（ボディ）
－	電文フォーマット	csaff_sample_standard

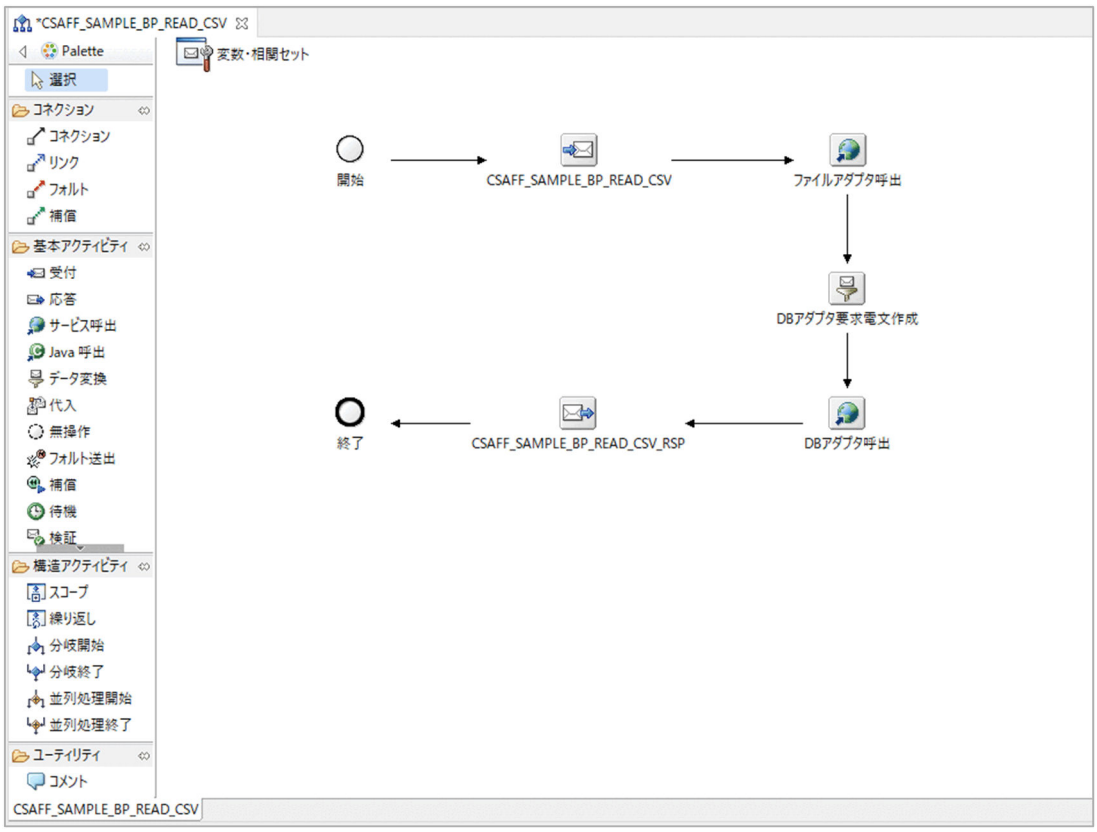
（凡例）

－：該当しません。

(c)　アクティビティの設定

ここでは，次の図に示すアクティビティの配置・定義，データ変換定義，およびアクティビティの連結について説明します。

図 D-6　ビジネスプロセス定義全体図（CSV データ読み込みサンプル）



●アクティビティの配置・定義

CSV データ読み込みサンプルでは，次の表に示すアクティビティを設定します。

表 D-28　アクティビティ一覧（CSV データ読み込みサンプル）

アクティビティ	アクティビティ名	説明
受付	CSAFF_SAMPLE_BP_READ_CSV	ビジネスプロセスの受付。
応答	CSAFF_SAMPLE_BP_READ_CSV_RSP	ビジネスプロセスの応答。
サービス呼出	ファイルアダプタ呼出	ファイルアダプタの呼び出し。
サービス呼出	DB アダプタ呼出	DB アダプタの呼び出し。
データ変換	DB アダプタ要求電文作成	ファイルアダプタの応答電文を DB アダプタの要求電文に変換。

アクティビティを配置して設定する手順を次に示します。

1. パレットの基本アクティビティまたは構造アクティビティの中から，キャンバスに配置するアクティビティをクリックします。

   クリックしたアクティビティが選択された状態になります。

2. キャンバス上の適当な場所をクリックします。

   選択したアクティビティがキャンバス上に配置されます。配置されたアクティビティは，ドラッグ＆ドロップで場所を移動できます。

3. 配置されたアクティビティをダブルクリックします。

   ダブルクリックしたアクティビティのダイアログが表示されます。

4. ダイアログに必要な情報を入力します。

   設定内容の詳細は，「アクティビティの設定内容（CSV データ読み込みサンプル）」を参照してください。

5. [OK] ボタンをクリックします。

   アクティビティのダイアログが閉じます。

6. アクティビティごとに手順 1 から 5 を繰り返します。

アクティビティの設定内容（CSV データ読み込みサンプル）

CSV データ読み込みサンプルで使用するビジネスプロセスのアクティビティの設定内容を次に示します。なお，表に記載されていない項目については，設定不要です。

表 D–29　CSAFF_SAMPLE_BP_READ_CSV の設定内容

項目	設定値
アクティビティ名	CSAFF_SAMPLE_BP_READ_CSV
オペレーション名	CSAFF_SAMPLE_BP_READ_CSV
ボディ割当変数	CSAFF_RequestMessage
通信モデル	同期
インスタンス生成	Yes

表 D–30　CSAFF_SAMPLE_BP_READ_CSV_RSP の設定内容

項目	設定値
アクティビティ名	CSAFF_SAMPLE_BP_READ_CSV_RSP
オペレーション名	CSAFF_SAMPLE_BP_READ_CSV
ボディ割当変数	CSAFF_DB_ResponseMessage

表 D–31　ファイルアダプタ呼出の設定内容（CSV データ読み込みサンプル）

項目	設定値
アクティビティ名	ファイルアダプタ呼出
サービス名	CSAFF_SAMPLE_READ_CSV

項目		設定値
オペレーション名		CSAFF_SAMPLE_READ_CSV
通信モデル		同期
要求電文	ボディ割当変数	CSAFF_RequestMessage
応答電文	ボディ割当変数	CSAFF_ResponseMessage

表 D-32　DB アダプタ呼出の設定内容（CSV データ読み込みサンプル）

項目		設定値
アクティビティ名		DB アダプタ呼出
サービス名		CSAFF_SAMPLE_DB
オペレーション名		CSAFF_SAMPLE_DB
通信モデル		同期
要求電文	ボディ割当変数	CSAFF_DB_RequestMessage
応答電文	ボディ割当変数	CSAFF_DB_ResponseMessage

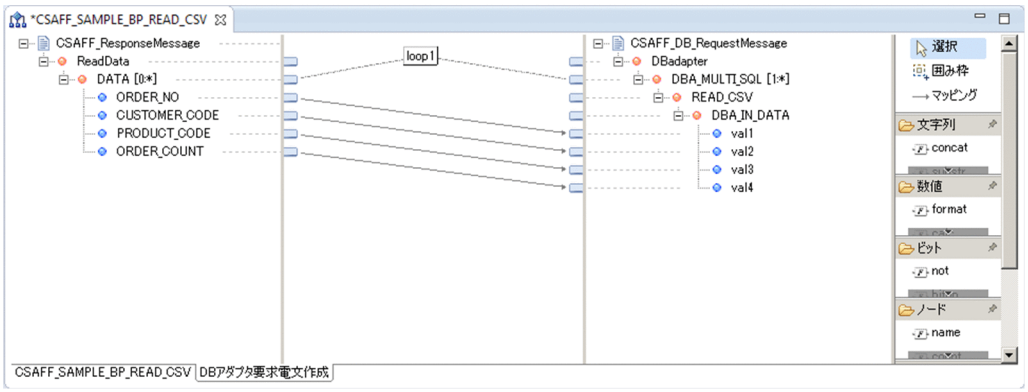
表 D-33　DB アダプタ要求電文作成の設定内容（CSV データ読み込みサンプル）

項目	設定値
アクティビティ名	DB アダプタ要求電文作成
変換元変数	CSAFF_ResponseMessage
変換先変数	CSAFF_DB_RequestMessage

●DB アダプタ要求電文作成アクティビティのデータ変換定義の設定

DB アダプタ要求電文作成アクティビティのデータ変換定義を設定します。設定するデータ変換定義を次の図に示します。

図 D-7　DB アダプタ要求電文作成アクティビティデータ変換定義画面（CSV データ読み込みサンプル）



DB アダプタ要求電文作成アクティビティのデータ変換定義の手順を次に示します。

1. DB アダプタ要求電文作成アクティビティからデータ変換（マッピング）定義画面を表示します。

データ変換（マッピング）定義画面が表示されたあと，［ルート要素選択］ダイアログが表示されます。

2. すべてのスキーマ論理名のルート要素を選択します。
［OK］ボタンがクリックできる状態になります。

3. ［OK］ボタンをクリックします。
変換元スキーマツリービューアおよび変換先スキーマツリービューアに，選択したルート要素のスキーマがツリー表示されます。

4. マッピングします。
マッピングでは次の作業を行います。
・ファンクションの配置とファンクションの設定
ファンクションの設定内容を次の表に示します。

表 D-34　ファンクションの設定内容（CSV データ読み込みサンプル-ビジネスプロセス）

ファンクション	ファンクション名	設定値	
loop	loop1	ノード選択	DATA
		ノード条件設定	設定しない

・ノード間のマッピングの設定
ノード間のマッピングの設定内容を次の表に示します。

表 D-35　マッピングの設定内容（CSV データ読み込みサンプル-ビジネスプロセス）

変換元ノード	ファンクション	変換先ノード
－	loop1	DBA_MULTI_SQL
ORDER_NO	－	val1
CUSTOMER_CODE	－	val2
PRODUCT_CODE	－	val3
ORDER_COUNT	－	val4

（凡例）
－：該当しません。

5. データ変換（マッピング）定義画面の変換元スキーマツリービューア，マッピングビューア，または変換先スキーマツリービューアの適当な場所で右クリックして，［検証］を選択します。
検証が実行されます。なお，警告は無視してください。

●アクティビティの連結

CSV データ読み込みサンプルで使用するビジネスプロセスでは，次の表に示すようにアクティビティを連結します。

表 D-36　アクティビティの連結内容（CSV データ読み込みサンプル）

連結元アクティビティ	連結先アクティビティ
開始	CSAFF_SAMPLE_BP_READ_CSV
CSAFF_SAMPLE_BP_READ_CSV	ファイルアダプタ呼出
ファイルアダプタ呼出	DB アダプタ要求電文作成
DB アダプタ要求電文作成	DB アダプタ呼出
DB アダプタ呼出	CSAFF_SAMPLE_BP_READ_CSV_RSP
CSAFF_SAMPLE_BP_READ_CSV_RSP	終了

アクティビティの連結の手順を次に示します。

1. パレットの［コネクション］をクリックします。
   コネクションが選択された状態になります。
   この状態で，アクティビティにマウスポインタを合わせると，そのアクティビティを連結元に設定できる場合，アクティビティの背景色が変わります。
2. キャンバスに配置したアクティビティのうち，連結元となるアクティビティをクリックします。
   連結元となるアクティビティが設定されます。
   この状態で，ほかのアクティビティにマウスポインタを合わせると，そのアクティビティを連結先に設定できる場合，アクティビティの背景色が変わります。
3. キャンバスに配置したアクティビティのうち，連結先となるアクティビティをクリックします。
   連結元となるアクティビティと連結先となるアクティビティが連結されます。

(d)　ビジネスプロセスの保存・検証

ビジネスプロセスを保存して検証する手順を次に示します。

1. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。
   ビジネスプロセスのサービス ID は，プロパティビューで「CSAFF_B2」を設定してください。
2. ビジネスプロセスの設定を検証します。
   ツリービューのサービス定義一覧で右クリックして表示されるサービス一覧のポップアップメニューから，「検証」を選択すると，検証結果がコンソールビューに表示されます。

(3)　ファイルアダプタとビジネスプロセスの配備定義

定義したファイルアダプタとビジネスプロセスをサーバに配備して開始します。

ファイルアダプタとビジネスプロセスの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，［複数サービスをサーバに配備して開始］を選択します。

2. サービス選択ページで配備および開始するサービスを選択します。

3. [Finish] 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。

4. [OK] ボタンをクリックします。
   ファイルアダプタとビジネスプロセスの配備と開始が完了します。

(4) テーブルの作成

サンプルを実行するデータベースのテーブルを作成します。

HiRDB SQL Executer を起動して，次の SQL 文を実行してください。

```
CREATE TABLE CSAFF_SAMPLE_RC
(ORDER_NO INT,
 CUSTOMER_CODE CHAR(5),
 PRODUCT_CODE CHAR(4),
 ORDER_COUNT INT);
```

注意事項

ユーザ ID は「ADMIN」とします。

(5) 要求電文の作成

サンプルで提供するビジネスプロセス実行時の要求電文を編集して，要求電文を作成する手順を次に示します。

1. サンプルで提供する要求電文ファイルを，テキストエディタで開きます。
   サンプルで提供する要求電文ファイルは，「<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¥custom-adapter¥File¥sample¥read_csv¥read_csv_message.xml」です。

2. 次に示す斜体部分を記述します。

```
<?xml version="1.0" encoding="UTF-8"?>
<adpff:ReadRequest xmlns:adpff="http://FlatFiles.cstmadp.csc.soft.Hitachi.co.jp/ReadRequest">
  <fileheader>
    <filename><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¥custom-adapter¥File¥sample¥read_csv¥read_csv_data.csv</filename>
    <filemode>all</filemode>
  </fileheader>
</adpff:ReadRequest>
```

3. ファイルを上書き保存します。

(6) ビジネスプロセスの実行

サンプルを実行するには，サービスリクエスタからビジネスプロセスを起動します。

サービスリクエスタの処理の流れと概要を次に示します。

1. サービスリクエスタ実行
   ビジネスプロセスの要求電文が，そのままファイルアダプタの要求電文として使用されます。
2. ファイルアダプタ実行
   要求電文で指定されたファイルが読み込まれます。
3. DB アダプタ電文作成
   ファイルアダプタの応答電文が DB アダプタの要求電文の形式に変換されます。
4. DB アダプタ実行
   要求電文で指定されたファイルがデータベースに登録されます。
5. サービスリクエスタ終了
   DB アダプタの応答電文が，そのままビジネスプロセスの応答電文として使用されます。

サービスリクエスタの実行手順を次に示します。

1. カレントディレクトリを「<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¥custom-adapter¥File¥sample¥bin」に移動します。
2. request.bat を実行します。
   次の形式で実行してください。

```
request read_csv <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¥custom-adapter¥File¥sample¥read_csv¥read_csv_message.xml
```

参考

HiRDB SQL Executer を起動して次の SQL 文を実行することで，テーブルの内容が確認できます（ユーザ ID は「ADMIN」とします）。

```
SELECT * FROM CSAFF_SAMPLE_RC;
```

付録 D.5 XML データ書き込みサンプル

XML データ書き込みサンプルでは，DB アダプタが取得したデータを，ファイルアダプタを通して XML 形式のファイルに書き込みます。

入出力項目を次の表に示します。

表 D-37　入出力項目一覧（XML データ書き込みサンプル）

入出力項目	内容
ファイルアダプタの出力ファイル	<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¥custom-adapter¥File¥sample¥write_xml¥write_xml_data.xml
DB アダプタの入力テーブル	ADMIN.CSAFF_SAMPLE_WX※

注※

スキーマ名は変更しないでください。

(1)　ファイルアダプタの定義

(a)　ファイルアダプタの追加と検証

ファイルアダプタを追加設定し，検証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. [HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。
3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   サービスアダプタ定義追加ウィザードが表示されます。
4. サービス部品の種別のドロップダウンリストから「ファイルアダプタ」を選択します。
5. [次へ］ボタンをクリックします。
   サービスアダプタ定義追加（ファイルアダプタ）ウィザードが表示されます。
6. サービス名を入力します。
   サービス名には，「CSAFF_SAMPLE_WRITE_XML」と入力します。
7. [終了］ボタンをクリックします。
   必要なファイルが作成され，リポジトリに保存されます。サービスアダプタ定義画面（基本）が表示されます。
8. サービスアダプタ定義画面（基本）およびサービスアダプタ定義画面（詳細）を設定します。
   設定内容については，「ファイルアダプタの設定内容（XML データ書き込みサンプル）」を参照してください。
9. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

10. ファイルアダプタの設定を検証します。

ツリービューのサービス定義一覧で右クリックして表示されるサービス一覧のポップアップメニューから，「検証」を選択すると，検証結果がコンソールビューに表示されます。

ファイルアダプタの設定内容（XML データ書き込みサンプル）

XML データ書き込みサンプルのファイルアダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 D-38　サービスアダプタ定義画面（基本）の設定内容（XML データ書き込みサンプル）

分類	項目		設定値	設定
サービス部品制御情報	サービス名		CSAFF_SAMPLE_WRITE_XML	◎
	サービス ID		CSAFF_WX	◎
	サービス種別		ファイルアダプタ	×
	最大インスタンス数		0	◎
	オペレーション		定義パターン：XML 書き込み オペレーション名： CSAFF_SAMPLE_WRITE_XML	◎
オペレーション情報	オペレーション名		CSAFF_SAMPLE_WRITE_XML	×
	通信モデル		同期	◎
要求電文	標準	使う（チェックボックス）	使わない（チェックしない）	◎
	サービス部品	電文フォーマット	csaff_sample_write_xml_dt.xsd	◎
応答電文	標準	使う（チェックボックス）	使わない（チェックしない）	◎
	サービス部品	電文フォーマット	adpff_result.xsd	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

XML データ書き込みサンプルのファイルアダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 D-39　サービスアダプタ定義画面（詳細）の設定内容（XML データ書き込みサンプル）

分類	項目	設定値	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	cscmsg_adpejb.jar	×
	ユーティリティクラス（JAR ファイル）	adpffpc.jar	×

（凡例）

×：表示されている内容を確認します。

(b)　ファイルアダプタ実行環境プロパティファイルの作成

ファイルアダプタ実行環境プロパティファイルを作成する手順を次に示します。

1. サンプルで提供するファイルアダプタ実行環境プロパティファイルを，テキストエディタで開きます。

   サンプルで提供するファイルアダプタ実行環境プロパティファイルは，「<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¥custom-adapter¥File¥sample¥write_xml¥CSAFF_WX.properties」です。

2. 次の内容を記述します。

```
fileaccess.path1 = <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¥¥custom-adapter¥¥File¥¥sample¥¥write_xml
```

注意事項

ディレクトリの区切り文字は，「¥¥」で記述します。

3. ファイルを保存し，「<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¥custom-adapter¥File¥config」に格納します。

(2)　ビジネスプロセスの定義

ビジネスプロセスの定義は，ビジネスプロセスの追加，ビジネスプロセスの内容の定義，ビジネスプロセスの検証の順で行います。

(a)　ビジネスプロセスの追加

ビジネスプロセスを追加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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。

   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「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。

4. ビジネスプロセス名を入力し，ステータスを永続化するかどうか（ビジネスプロセスの実行状態をデータベースに保存するかどうか）を選択します。

   ビジネスプロセス名には，「CSAFF_SAMPLE_BP_WRITE_XML」と入力します。

   ステータスの永続化には，「yes」を選択します。

参考

BPEL ファイルは使用しません。

5. [終了] ボタンをクリックします。
   必要なファイルが作成され，リポジトリに保存されます。ビジネスプロセス定義画面が表示されます。

(b) 変数の設定

次に示す変数を設定します。

- CSAFF_BP_RequestMessage
- CSAFF_DB_RequestMessage
- CSAFF_DB_ResponseMessage
- CSAFF_RequestMessage
- CSAFF_ResponseMessage

参考

サンプルでは相関セットを使用しません。

変数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. 一覧から「変数一覧」を選択します。
3. 変数名を入力します。
4. [種別] のドロップダウンリストから「XML」を選択します。
5. 次のどれかの操作をします。

   CSAFF_DB_ResponseMessage，CSAFF_RequestMessage または CSAFF_ResponseMessage の場合

   ［取込］ボタンをクリックして表示される［電文フォーマットの取込］ダイアログで，使用する電文フォーマットを指定します。設定内容については，「変数の設定内容（XML データ書き込みサンプル）」を参照してください。

   CSAFF_BP_RequestMessage の場合

   [...] ボタンをクリックして，「csaff_sample_write_xml.xsd」を設定します。

   CSAFF_DB_RequestMessage の場合

   [...] ボタンをクリックして，「csaff_sample_write_xml_db.xsd」を設定します。

6. [追加] ボタンをクリックします。

変数一覧に，追加した変数が表示されます。

7. 変数ごとに手順 2 から 6 を繰り返します。

8. [OK] ボタンをクリックします。

変数の設定値が保存されて，［変数・相関セット一覧］ダイアログが閉じます。

変数の設定内容（XML データ書き込みサンプル）

CSAFF_DB_ResponseMessage，CSAFF_RequestMessage および CSAFF_ResponseMessage の設定内容を次に示します。なお，表に記載されていない項目については，設定不要です。

表 D–40　CSAFF_DB_ResponseMessage の設定内容（XML データ書き込みサンプル）

分類	項目	設定値
サービス/受付	サービス名	CSAFF_SAMPLE_DB
	受付名	－
取込対象	オペレーション名	CSAFF_SAMPLE_DB
	電文種別	応答電文（ボディ）
－	電文フォーマット	csaff_sample_sqlformat

（凡例）

－：該当しません。

表 D–41　CSAFF_RequestMessage の設定内容（XML データ書き込みサンプル）

分類	項目	設定値
サービス/受付	サービス名	CSAFF_SAMPLE_WRITE_XML
	受付名	－
取込対象	オペレーション名	CSAFF_SAMPLE_WRITE_XML
	電文種別	要求電文（ボディ）
－	電文フォーマット	csaff_sample_write_xml_dt

（凡例）

－：該当しません。

表 D–42　CSAFF_ResponseMessage の設定内容（XML データ書き込みサンプル）

分類	項目	設定値
サービス/受付	サービス名	CSAFF_SAMPLE_WRITE_XML
	受付名	－
取込対象	オペレーション名	CSAFF_SAMPLE_WRITE_XML
	電文種別	応答電文（ボディ）

分類	項目	設定値
－	電文フォーマット	adpff_result

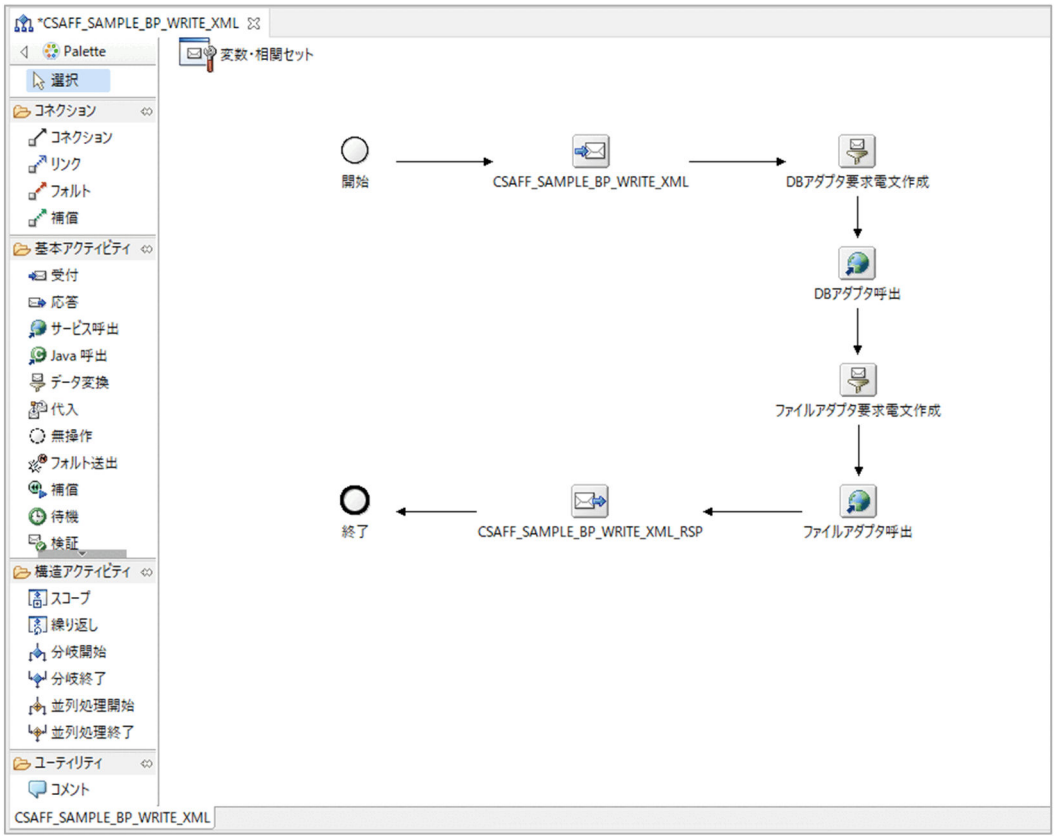
（凡例）

－：該当しません。

(c)　アクティビティの設定

ここでは，次の図に示すアクティビティの配置・定義，データ変換定義，およびアクティビティの連結について説明します。

図 D-8　ビジネスプロセス定義全体図（XML データ書き込みサンプル）



●アクティビティの配置・定義

XML データ書き込みサンプルでは，次の表に示すアクティビティを設定します。

表 D-43　アクティビティ一覧（XML データ書き込みサンプル）

アクティビティ	アクティビティ名	説明
受付	CSAFF_SAMPLE_BP_WRITE_XML	ビジネスプロセスの受付。
応答	CSAFF_SAMPLE_BP_WRITE_XML_RSP	ビジネスプロセスの応答。
サービス呼出	ファイルアダプタ呼出	ファイルアダプタの呼び出し。
サービス呼出	DB アダプタ呼出	DB アダプタの呼び出し。

アクティビティ	アクティビティ名	説明
データ変換	ファイルアダプタ要求電文作成	DB アダプタの応答電文をファイルアダプタの要求電文に変換。
データ変換	DB アダプタ要求電文作成	ビジネスプロセスの要求電文を DB アダプタの要求電文に変換。

アクティビティを配置して設定する手順を次に示します。

1. パレットの基本アクティビティまたは構造アクティビティの中から，キャンバスに配置するアクティビティをクリックします。

   クリックしたアクティビティが選択された状態になります。

2. キャンバス上の適当な場所をクリックします。

   選択したアクティビティがキャンバス上に配置されます。配置されたアクティビティは，ドラッグ＆ドロップで場所を移動できます。

3. 配置されたアクティビティをダブルクリックします。

   ダブルクリックしたアクティビティのダイアログが表示されます。

4. ダイアログに必要な情報を入力します。

   設定内容の詳細は，「アクティビティの設定内容（XML データ書き込みサンプル）」を参照してください。

5. [OK] ボタンをクリックします。

   アクティビティのダイアログが閉じます。

6. アクティビティごとに手順 1 から 5 を繰り返します。

アクティビティの設定内容（XML データ書き込みサンプル）

XML データ書き込みサンプルで使用するビジネスプロセスのアクティビティの設定内容を次に示します。なお，表に記載されていない項目については，設定不要です。

表 D-44　CSAFF_SAMPLE_BP_WRITE_XML の設定内容

項目	設定値
アクティビティ名	CSAFF_SAMPLE_BP_WRITE_XML
オペレーション名	CSAFF_SAMPLE_BP_WRITE_XML
ボディ割当変数	CSAFF_BP_RequestMessage
通信モデル	同期
インスタンス生成	Yes

表 D-45　CSAFF_SAMPLE_BP_WRITE_XML_RSP の設定内容

項目	設定値
アクティビティ名	CSAFF_SAMPLE_BP_WRITE_XML_RSP
オペレーション名	CSAFF_SAMPLE_BP_WRITE_XML

項目	設定値
ボディ割当変数	CSAFF_ResponseMessage

表 D-46　ファイルアダプタ呼出の設定内容（XML データ書き込みサンプル）

項目		設定値
アクティビティ名		ファイルアダプタ呼出
サービス名		CSAFF_SAMPLE_WRITE_XML
オペレーション名		CSAFF_SAMPLE_WRITE_XML
通信モデル		同期
要求電文	ボディ割当変数	CSAFF_RequestMessage
応答電文	ボディ割当変数	CSAFF_ResponseMessage

表 D-47　DB アダプタ呼出の設定内容（XML データ書き込みサンプル）

項目		設定値
アクティビティ名		DB アダプタ呼出
サービス名		CSAFF_SAMPLE_DB
オペレーション名		CSAFF_SAMPLE_DB
通信モデル		同期
要求電文	ボディ割当変数	CSAFF_DB_RequestMessage
応答電文	ボディ割当変数	CSAFF_DB_ResponseMessage

表 D-48　ファイルアダプタ要求電文作成の設定内容（XML データ書き込みサンプル）

項目	設定値
アクティビティ名	ファイルアダプタ要求電文作成
変換元変数	CSAFF_BP_RequestMessage
	CSAFF_DB_ResponseMessage
変換先変数	CSAFF_RequestMessage

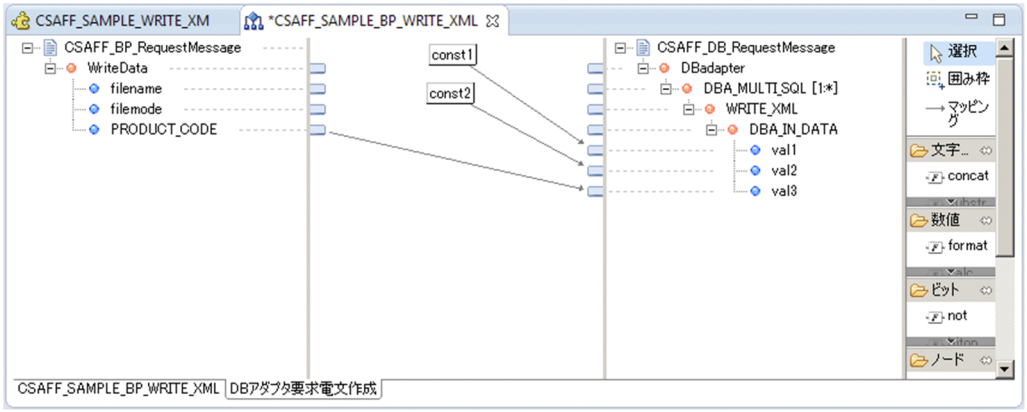
表 D-49　DB アダプタ要求電文作成の設定内容（XML データ書き込みサンプル）

項目	設定値
アクティビティ名	DB アダプタ要求電文作成
変換元変数	CSAFF_BP_RequestMessage
変換先変数	CSAFF_DB_RequestMessage

●DB アダプタ要求電文作成アクティビティのデータ変換定義の設定

DB アダプタ要求電文作成アクティビティのデータ変換定義を設定します。設定するデータ変換定義を次の図に示します。

図 D-9　DB アダプタ要求電文作成アクティビティデータ変換定義画面（XML データ書き込みサンプル）



DB アダプタ要求電文作成アクティビティのデータ変換定義の手順を次に示します。

1. DB アダプタ要求電文作成アクティビティからデータ変換（マッピング）定義画面を表示します。

   データ変換（マッピング）定義画面が表示されたあと，［ルート要素選択］ダイアログが表示されます。

2. すべてのスキーマ論理名のルート要素を選択します。

   ［OK］ボタンがクリックできる状態になります。

3. ［OK］ボタンをクリックします。

   変換元スキーマツリービューアおよび変換先スキーマツリービューアに，選択したルート要素のスキーマがツリー表示されます。

4. マッピングします。

   マッピングでは次の作業を行います。

   ・ファンクションの配置とファンクションの設定

   ファンクションの設定内容を次の表に示します。

表 D-50　ファンクションの設定内容（XML データ書き込みサンプル-DB アダプタ要求電文作成アクティビティ）

ファンクション	ファンクション名	設定値	
const	const1	定数種別	文字列
		値	PRODUCT_CODE
const	const2	定数種別	文字列
		値	=

・ノード間のマッピングの設定

ノード間のマッピングの設定内容を次の表に示します。

表 D-51　マッピングの設定内容（XML データ書き込みサンプル-DB アダプタ要求電文作成アクティビティ）

変換元ノード	ファンクション	変換先ノード
－	const1	val1
－	const2	val2
PRODUCT_CODE	－	val3

（凡例）

－：該当しません。

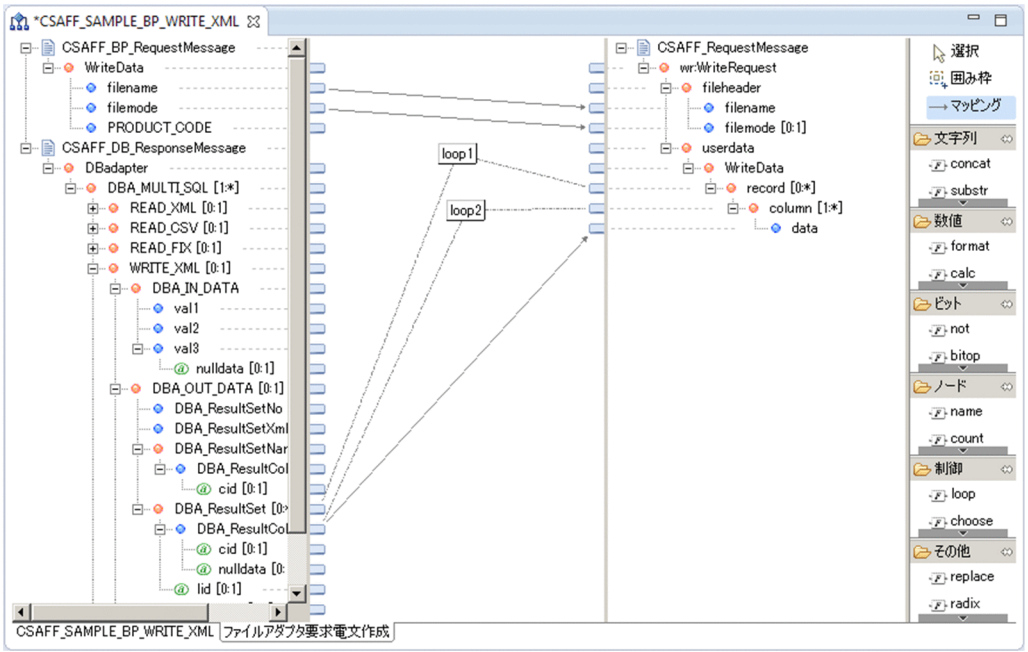
5. データ変換（マッピング）定義画面の変換元スキーマツリービューア，マッピングビューア，または変換先スキーマツリービューアの適当な場所で右クリックして，［検証］を選択します。

   検証が実行されます。

●ファイルアダプタ要求電文作成アクティビティのデータ変換定義の設定内容

ファイルアダプタ要求電文作成アクティビティのデータ変換定義を設定します。設定するデータ変換定義を次の図に示します。

図 D-10　ファイルアダプタ要求電文作成アクティビティデータ変換定義画面（XML データ書き込みサンプル）



ファイルアダプタ要求電文作成アクティビティのデータ変換定義の手順を次に示します。

1. ファイルアダプタ要求電文作成アクティビティからデータ変換（マッピング）定義画面を表示します。

   データ変換（マッピング）定義画面が表示されたあと，［ルート要素選択］ダイアログが表示されます。

2. すべてのスキーマ論理名のルート要素を選択します。

［OK］ボタンがクリックできる状態になります。

3. ［OK］ボタンをクリックします。

変換元スキーマツリービューアおよび変換先スキーマツリービューアに，選択したルート要素のスキーマがツリー表示されます。

4. マッピングします。

マッピングでは次の作業を行います。

・ファンクションの配置とファンクションの設定

ファンクションの設定内容を次の表に示します。

表 D-52　ファンクションの設定内容（XML データ書き込みサンプル-ファイルアダプタ要求電文作成アクティビティ）

ファンクション	ファンクション名	設定値	
loop	loop1	ノード選択	DBA_ResultSet※
		ノード条件設定	設定しない
loop	loop2	ノード選択	DBA_ResultColumn※
		ノード条件設定	設定しない

注※

DBA_ResultSet および DBA_ResultColumn は，WRITE_XML ノード下の情報を使用します。

・ノード間のマッピングの設定

ノード間のマッピングの設定内容を次の表に示します。

表 D-53　マッピングの設定内容（XML データ書き込みサンプル-ファイルアダプタ要求電文作成アクティビティ）

変換元ノード	ファンクション	変換先ノード
－	loop1	record
－	loop2	column
filename	－	filename
filemode	－	filemode
DBA_ResultColumn※	－	data

（凡例）

－：該当しません。

注※

DBA_ResultColumn は，WRITE_XML ノード下の情報を使用します。

5. データ変換（マッピング）定義画面の変換元スキーマツリービューア，マッピングビューア，または変換先スキーマツリービューアの適当な場所で右クリックして，［検証］を選択します。

検証が実行されます。なお，警告は無視してください。

●アクティビティの連結

XML データ書き込みサンプルで使用するビジネスプロセスでは，次の表に示すようにアクティビティを連結します。

表 D-54　アクティビティの連結内容（XML データ書き込みサンプル）

連結元アクティビティ	連結先アクティビティ
開始	CSAFF_SAMPLE_BP_WRITE_XML
CSAFF_SAMPLE_BP_WRITE_XML	DB アダプタ要求電文作成
DB アダプタ要求電文作成	DB アダプタ呼出
DB アダプタ呼出	ファイルアダプタ要求電文作成
ファイルアダプタ要求電文作成	ファイルアダプタ呼出
ファイルアダプタ呼出	CSAFF_SAMPLE_BP_WRITE_XML_RSP
CSAFF_SAMPLE_BP_WRITE_XML_RSP	終了

アクティビティの連結の手順を次に示します。

1. パレットの［コネクション］をクリックします。

   コネクションが選択された状態になります。

   この状態で，アクティビティにマウスポインタを合わせると，そのアクティビティを連結元に設定できる場合，アクティビティの背景色が変わります。

2. キャンバスに配置したアクティビティのうち，連結元となるアクティビティをクリックします。

   連結元となるアクティビティが設定されます。

   この状態で，ほかのアクティビティにマウスポインタを合わせると，そのアクティビティを連結先に設定できる場合，アクティビティの背景色が変わります。

3. キャンバスに配置したアクティビティのうち，連結先となるアクティビティをクリックします。

   連結元となるアクティビティと連結先となるアクティビティが連結されます。

(d)　ビジネスプロセスの保存・検証

ビジネスプロセスを保存して検証する手順を次に示します。

1. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

   ビジネスプロセスのサービス ID は，プロパティビューで「CSAFF_B4」を設定してください。

2. ビジネスプロセスの設定を検証します。

   ツリービューのサービス定義一覧で右クリックして表示されるサービス一覧のポップアップメニューから，「検証」を選択すると，検証結果がコンソールビューに表示されます。

(3)　ファイルアダプタとビジネスプロセスの配備定義

定義したファイルアダプタとビジネスプロセスをサーバに配備して開始します。

ファイルアダプタとビジネスプロセスの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，［複数サービスをサーバに配備して開始］を選択します。
2. サービス選択ページで配備および開始するサービスを選択します。
3. [Finish] 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。
4. [OK] ボタンをクリックします。
   ファイルアダプタとビジネスプロセスの配備と開始が完了します。

(4)　テーブルの作成

サンプルを実行するデータベースのテーブルを作成します。

HiRDB SQL Executer を起動して，次の SQL 文を実行してください。

```
CREATE TABLE CSAFF_SAMPLE_WX
(ORDER_NO INT,
 CUSTOMER_CODE CHAR(5),
 PRODUCT_CODE CHAR(4),
 ORDER_COUNT INT);
INSERT INTO CSAFF_SAMPLE_WX VALUES(1001,'AA001','1001',5);
INSERT INTO CSAFF_SAMPLE_WX VALUES(1002,'AB002','1001',1);
INSERT INTO CSAFF_SAMPLE_WX VALUES(1003,'AA001','1102',3);
INSERT INTO CSAFF_SAMPLE_WX VALUES(1004,'XA005','1103',1);
INSERT INTO CSAFF_SAMPLE_WX VALUES(1005,'AA001','1105',1);
```

注意事項

ユーザ ID は「ADMIN」とします。

(5)　要求電文の作成

サンプルで提供するビジネスプロセス実行時の要求電文を編集して，要求電文を作成する手順を次に示します。

1. サンプルで提供する要求電文ファイルを，テキストエディタで開きます。
   サンプルで提供する要求電文ファイルは，「<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¥custom-adapter¥File¥sample¥write_xml¥write_xml_message.xml」です。

2. 次に示す斜体部分を記述します。

```
<?xml version="1.0" encoding="UTF-8"?>
<WriteData>
  <filename><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¥custom-adapter¥File¥
sample¥write_xml¥write_xml_data.xml</filename>
  <filemode>new</filemode>
  <PRODUCT_CODE>1001</PRODUCT_CODE>
</WriteData>
```

3. ファイルを上書き保存します。

(6) ビジネスプロセスの実行

サンプルを実行するには，サービスリクエスタからビジネスプロセスを起動します。

サービスリクエスタの処理の流れと概要を次に示します。

1. サービスリクエスタ実行
   ビジネスプロセスの要求電文が，そのままファイルアダプタの要求電文として使用されます。
2. DB アダプタ実行
   要求電文で指定された条件に合致するデータがデータベースから取得されます。
3. ファイルアダプタ電文作成
   DB アダプタの応答電文がファイルアダプタの要求電文の形式に変換されます。
4. ファイルアダプタ実行
   要求電文で指定されたファイルが出力されます。
5. サービスリクエスタ終了
   ファイルアダプタの応答電文が，そのままビジネスプロセスの応答電文として使用されます。

サービスリクエスタの実行手順を次に示します。

1. カレントディレクトリを「<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¥custom-adapter¥File¥sample¥bin」に移動します。

2. request.bat を実行します。
   次の形式で実行してください。

```
request write_xml <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¥custom-adapter
¥File¥sample¥write_xml¥write_xml_message.xml
```

参考

出力先は，次のファイルです。

<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¥custom-adapter¥File¥sample¥write_xml¥write_xml_data.xml

付録 D.6　CSV データ書き込みサンプル

CSV データ書き込みサンプルでは，DB アダプタが取得したデータを，ファイルアダプタを通して CSV 形式のファイルに書き込みます。

入出力項目を次の表に示します。

表 D-55　入出力項目一覧（CSV データ書き込みサンプル）

入出力項目	内容
ファイルアダプタの出力ファイル	<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¥custom-adapter¥File¥sample¥write_csv¥write_csv_data.csv
DB アダプタの入力テーブル	ADMIN.CSAFF_SAMPLE_WC※

注※

スキーマ名は変更しないでください。

(1)　ファイルアダプタの定義

(a)　ファイルアダプタの追加と検証

ファイルアダプタを追加設定し，検証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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。
3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   サービスアダプタ定義追加ウィザードが表示されます。
4. サービス部品の種別のドロップダウンリストから「ファイルアダプタ」を選択します。
5. ［次へ］ボタンをクリックします。
   サービスアダプタ定義追加（ファイルアダプタ）ウィザードが表示されます。
6. サービス名を入力します。
   サービス名には，「CSAFF_SAMPLE_WRITE_CSV」と入力します。

7. [終了] ボタンをクリックします。

必要なファイルが作成され，リポジトリに保存されます。サービスアダプタ定義画面（基本）が表示されます。

8. サービスアダプタ定義画面（基本）およびサービスアダプタ定義画面（詳細）を設定します。

設定内容については，「ファイルアダプタの設定内容（CSV データ書き込みサンプル）」を参照してください。

9. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

10. ファイルアダプタの設定を検証します。

ツリービューのサービス定義一覧で右クリックして表示されるサービス一覧のポップアップメニューから，「検証」を選択すると，検証結果がコンソールビューに表示されます。

ファイルアダプタの設定内容（CSV データ書き込みサンプル）

CSV データ書き込みサンプルのファイルアダプタを定義する場合の，サービスアダプタ定義画面（基本）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 D-56　サービスアダプタ定義画面（基本）の設定内容（CSV データ書き込みサンプル）

分類	項目		設定値	設定
サービス部品制御情報	サービス名		CSAFF_SAMPLE_WRITE_CSV	◎
	サービス ID		CSAFF_WC	◎
	サービス種別		ファイルアダプタ	×
	アドレス		－	－
	最大インスタンス数		0	◎
	オペレーション		定義パターン：バイナリ書き込み オペレーション名：CSAFF_SAMPLE_WRITE_CSV	◎
オペレーション情報	オペレーション名		CSAFF_SAMPLE_WRITE_CSV	×
	通信モデル		同期	◎
要求電文	標準	使う（チェックボックス）	使う（チェックする）	◎
		電文フォーマット	csaff_sample_write_csv_standard.xsd	◎
	サービス部品	電文フォーマット	csaff_sample_write_csv.fdx	◎
	データ変換定義		SAMPLE_WRITE_CSV_DT	◎
応答電文	標準	使う（チェックボックス）	使わない（チェックしない）	◎

分類	項目		設定値	設定
応答電文	サービス部品	電文フォーマット	adpff_result.xsd	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

－：設定しません。

CSV データ書き込みサンプルのファイルアダプタを定義する場合の，サービスアダプタ定義画面（詳細）での設定項目を次の表に示します。なお，表に記載されていない項目については，設定不要です。

表 D-57　サービスアダプタ定義画面（詳細）の設定内容（CSV データ書き込みサンプル）

分類	項目	設定値	設定
サービスアダプタ制御情報	サービスアダプタ（EJB-JAR ファイル）	cscmsg_adpejb.jar	×
	ユーティリティクラス（JAR ファイル）	adpffpc.jar	×

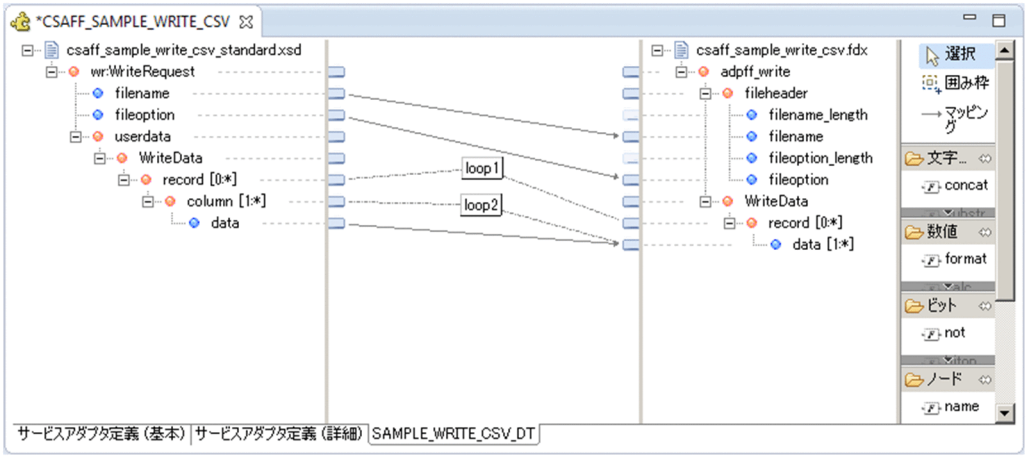
（凡例）

×：表示されている内容を確認します。

(b)　データ変換定義

データ変換定義を設定します。設定するデータ変換定義を次の図に示します。

図 D-11　ファイルアダプタのデータ変換定義画面（CSV データ書き込みサンプル）



データ変換定義の手順を次に示します。

1. アダプタ定義画面のデータ変換（マッピング）定義画面を表示します。

   データ変換（マッピング）定義画面が表示されたあと，［ルート要素選択］ダイアログが表示されます。

2. すべてのスキーマ論理名のルート要素を選択します。

   ［OK］ボタンがクリックできる状態になります。

3. [OK] ボタンをクリックします。

変換元スキーマツリービューアおよび変換先スキーマツリービューアに，選択したルート要素のスキーマがツリー表示されます。

4. マッピングします。

マッピングでは次の作業を行います。

- ファンクションの配置とファンクションの設定

  ファンクションの設定内容を次の表に示します。

表 D–58　ファンクションの設定内容（CSV データ書き込みサンプル-ファイルアダプタ）

ファンクション	ファンクション名	設定値	
loop	loop1	ノード選択	record
		ノード条件設定	設定しない
loop	loop2	ノード選択	column
		ノード条件設定	設定しない

- ノード間のマッピングの設定

  ノード間のマッピングの設定内容を次の表に示します。

表 D–59　マッピングの設定内容（CSV データ書き込みサンプル-ファイルアダプタ）

変換元ノード	ファンクション	変換先ノード
−	loop1	record
−	loop2	data
filename	−	filename
fileoption	−	fileoption
data	−	data

（凡例）

−：該当しません。

5. データ変換（マッピング）定義画面の変換元スキーマツリービューア，マッピングビューア，または変換先スキーマツリービューアの適当な場所で右クリックして，［検証］を選択します。

検証が実行されます。なお，警告は無視してください。

(c)　ファイルアダプタ実行環境プロパティファイルの作成

ファイルアダプタ実行環境プロパティファイルを作成する手順を次に示します。

1. サンプルで提供するファイルアダプタ実行環境プロパティファイルを，テキストエディタで開きます。

サンプルで提供するファイルアダプタ実行環境プロパティファイルは，「<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¥custom-adapter¥File¥sample¥write_csv¥CSAFF_WC.properties」です。

2. 次の内容を記述します。

```
fileaccess.path1 = <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¥¥custom-adapter¥¥File¥¥sample¥¥write_csv
character.code = MS932
codetable.path = <Code Converterの変換テーブル（ユーザマッピングファイル）格納パス>
```

ユーザマッピングファイルの格納パスを次に示します。

```
<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>¥¥CSC¥¥lib¥¥external¥¥table
```

3. ファイルを保存します。

ファイルの格納パスを次に示します。

```
<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>¥CSC¥custom-adapter¥File¥config
```

(2) ビジネスプロセスの定義

ビジネスプロセスの定義は，ビジネスプロセスの追加，ビジネスプロセスの内容の定義，ビジネスプロセスの検証の順で行います。

(a) ビジネスプロセスの追加

ビジネスプロセスを追加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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。

ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「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。

4. ビジネスプロセス名を入力し，ステータスを永続化するかどうか（ビジネスプロセスの実行状態をデータベースに保存するかどうか）を選択します。

ビジネスプロセス名には，「CSAFF_SAMPLE_BP_WRITE_CSV」と入力します。

ステータスの永続化には，「yes」を選択します。

参考

BPEL ファイルは使用しません。

5. [終了] ボタンをクリックします。
   必要なファイルが作成され，リポジトリに保存されます。ビジネスプロセス定義画面が表示されます。

(b) 変数の設定

次に示す変数を設定します。

- CSAFF_BP_RequestMessage
- CSAFF_DB_RequestMessage
- CSAFF_DB_ResponseMessage
- CSAFF_RequestMessage
- CSAFF_ResponseMessage

> **参考**
>
> サンプルでは相関セットを使用しません。

変数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. 一覧から「変数一覧」を選択します。
3. 変数名を入力します。
4. ［種別］のドロップダウンリストから「XML」を選択します。
5. 次のどれかの操作をします。

   CSAFF_DB_ResponseMessage，CSAFF_RequestMessage または CSAFF_ResponseMessage の場合

   ［取込］ボタンをクリックして表示される［電文フォーマットの取込］ダイアログで，使用する電文フォーマットを指定します。設定内容については，「変数の設定内容（CSV データ書き込みサンプル）」を参照してください。

   CSAFF_BP_RequestMessage の場合

   ［...］ボタンをクリックして，「csaff_sample_write_csv.xsd」を設定します。

   CSAFF_DB_RequestMessage の場合

   ［...］ボタンをクリックして，「csaff_sample_write_csv_db.xsd」を設定します。

6. ［追加］ボタンをクリックします。
   変数一覧に，追加した変数が表示されます。

7. 変数ごとに手順 2 から 6 を繰り返します。

8. [OK] ボタンをクリックします。

変数の設定値が保存されて，［変数・相関セット一覧］ダイアログが閉じます。

変数の設定内容（CSV データ書き込みサンプル）

CSAFF_DB_ResponseMessage，CSAFF_RequestMessage および CSAFF_ResponseMessage の設定内容を次に示します。なお，表に記載されていない項目については，設定不要です。

表 D-60　CSAFF_DB_ResponseMessage の設定内容（CSV データ書き込みサンプル）

分類	項目	設定値
サービス/受付	サービス名	CSAFF_SAMPLE_DB
	受付名	－
取込対象	オペレーション名	CSAFF_SAMPLE_DB
	電文種別	応答電文（ボディ）
－	電文フォーマット	csaff_sample_sqlformat

（凡例）

－：該当しません。

表 D-61　CSAFF_RequestMessage の設定内容（CSV データ書き込みサンプル）

分類	項目	設定値
サービス/受付	サービス名	CSAFF_SAMPLE_WRITE_CSV
	受付名	－
取込対象	オペレーション名	CSAFF_SAMPLE_WRITE_CSV
	電文種別	要求電文（ボディ）
－	電文フォーマット	csaff_sample_write_csv_standard

（凡例）

－：該当しません。

表 D-62　CSAFF_ResponseMessage の設定内容（CSV データ書き込みサンプル）

分類	項目	設定値
サービス/受付	サービス名	CSAFF_SAMPLE_WRITE_CSV
	受付名	－
取込対象	オペレーション名	CSAFF_SAMPLE_WRITE_CSV
	電文種別	応答電文（ボディ）
－	電文フォーマット	adpff_result

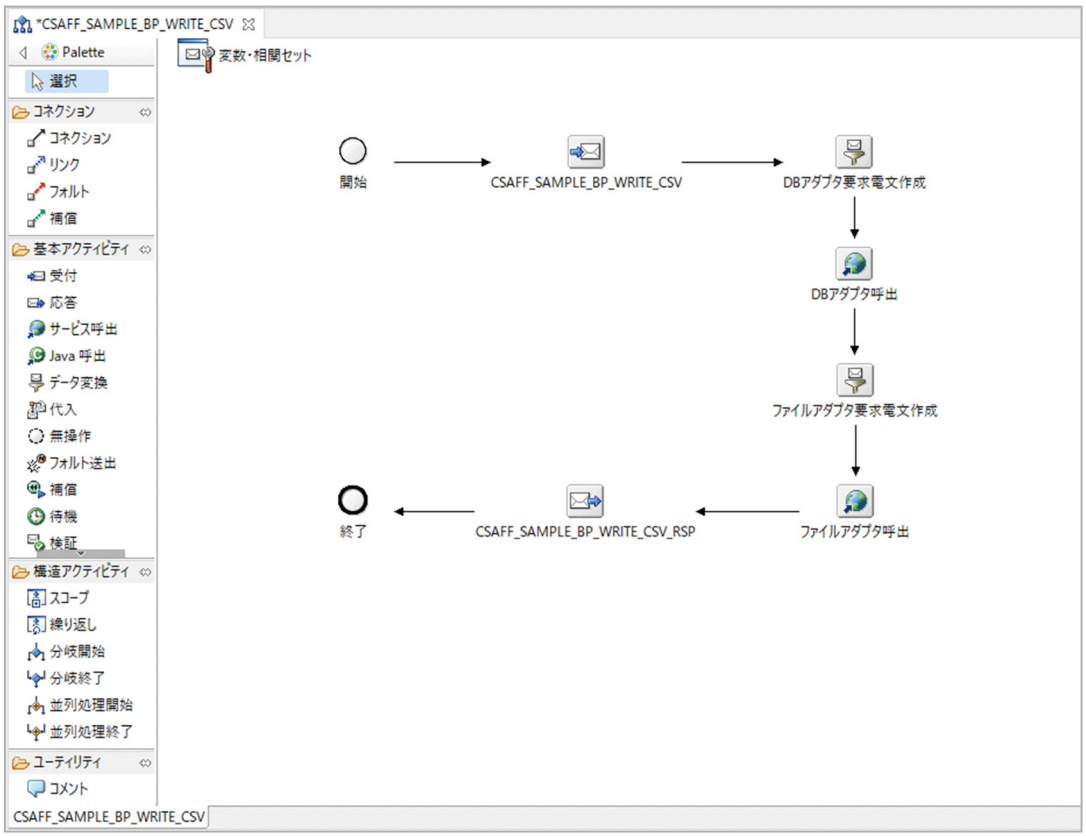
（凡例）

－：該当しません。

(c)　アクティビティの設定

ここでは，次の図に示すアクティビティの配置・定義，データ変換定義，およびアクティビティの連結について説明します。

図 D–12　ビジネスプロセス定義全体図（CSV データ書き込みサンプル）



●アクティビティの配置・定義

CSV データ書き込みサンプルでは，次の表に示すアクティビティを設定します。

表 D–63　アクティビティ一覧（CSV データ書き込みサンプル）

アクティビティ	アクティビティ名	説明
受付	CSAFF_SAMPLE_BP_WRITE_CSV	ビジネスプロセスの受付。
応答	CSAFF_SAMPLE_BP_WRITE_CSV_RSP	ビジネスプロセスの応答。
サービス呼出	ファイルアダプタ呼出	ファイルアダプタの呼び出し。
サービス呼出	DB アダプタ呼出	DB アダプタの呼び出し。
データ変換	ファイルアダプタ要求電文作成	DB アダプタの応答電文をファイルアダプタの要求電文に変換。
データ変換	DB アダプタ要求電文作成	ビジネスプロセスの要求電文を DB アダプタの要求電文に変換。

アクティビティを配置して設定する手順を次に示します。

1. パレットの基本アクティビティまたは構造アクティビティの中から，キャンバスに配置するアクティビティをクリックします。

   クリックしたアクティビティが選択された状態になります。

2. キャンバス上の適当な場所をクリックします。

   選択したアクティビティがキャンバス上に配置されます。配置されたアクティビティは，ドラッグ＆ドロップで場所を移動できます。

3. 配置されたアクティビティをダブルクリックします。

   ダブルクリックしたアクティビティのダイアログが表示されます。

4. ダイアログに必要な情報を入力します。

   設定内容の詳細は，「アクティビティの設定内容（CSV データ書き込みサンプル）」を参照してください。

5. [OK] ボタンをクリックします。

   アクティビティのダイアログが閉じます。

6. アクティビティごとに手順 1 から 5 を繰り返します。

アクティビティの設定内容（CSV データ書き込みサンプル）

CSV データ書き込みサンプルで使用するビジネスプロセスのアクティビティの設定内容を次に示します。なお，表に記載されていない項目については，設定不要です。

表 D-64　CSAFF_SAMPLE_BP_WRITE_CSV の設定内容

項目	設定値
アクティビティ名	CSAFF_SAMPLE_BP_WRITE_CSV
オペレーション名	CSAFF_SAMPLE_BP_WRITE_CSV
ボディ割当変数	CSAFF_BP_RequestMessage
通信モデル	同期
インスタンス生成	Yes

表 D-65　CSAFF_SAMPLE_BP_WRITE_CSV_RSP の設定内容

項目	設定値
アクティビティ名	CSAFF_SAMPLE_BP_WRITE_CSV_RSP
オペレーション名	CSAFF_SAMPLE_BP_WRITE_CSV
ボディ割当変数	CSAFF_ResponseMessage

表 D-66　ファイルアダプタ呼出の設定内容（CSV データ書き込みサンプル）

項目	設定値
アクティビティ名	ファイルアダプタ呼出

項目		設定値
サービス名		CSAFF_SAMPLE_WRITE_CSV
オペレーション名		CSAFF_SAMPLE_WRITE_CSV
通信モデル		同期
要求電文	ボディ割当変数	CSAFF_RequestMessage
応答電文	ボディ割当変数	CSAFF_ResponseMessage

表 D-67　DB アダプタ呼出の設定内容（CSV データ書き込みサンプル）

項目		設定値
アクティビティ名		DB アダプタ呼出
サービス名		CSAFF_SAMPLE_DB
オペレーション名		CSAFF_SAMPLE_DB
通信モデル		同期
要求電文	ボディ割当変数	CSAFF_DB_RequestMessage
応答電文	ボディ割当変数	CSAFF_DB_ResponseMessage

表 D-68　ファイルアダプタ要求電文作成の設定内容（CSV データ書き込みサンプル）

項目	設定値
アクティビティ名	ファイルアダプタ要求電文作成
変換元変数	CSAFF_BP_RequestMessage
	CSAFF_DB_ResponseMessage
変換先変数	CSAFF_RequestMessage

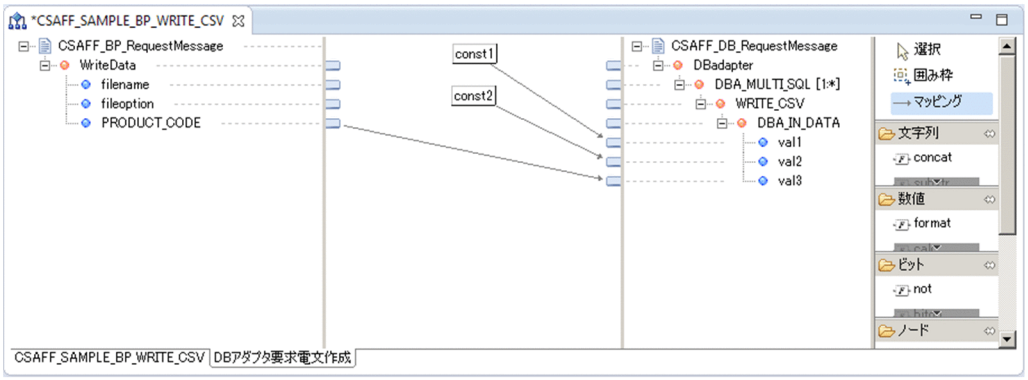
表 D-69　DB アダプタ要求電文作成の設定内容（CSV データ書き込みサンプル）

項目	設定値
アクティビティ名	DB アダプタ要求電文作成
変換元変数	CSAFF_BP_RequestMessage
変換先変数	CSAFF_DB_RequestMessage

●DB アダプタ要求電文作成アクティビティのデータ変換定義の設定

DB アダプタ要求電文作成アクティビティのデータ変換定義を設定します。設定するデータ変換定義を次の図に示します。

図 D–13　DB アダプタ要求電文作成アクティビティデータ変換定義画面（CSV データ書き込みサンプル）



DB アダプタ要求電文作成アクティビティのデータ変換定義の手順を次に示します。

1. DB アダプタ要求電文作成アクティビティからデータ変換（マッピング）定義画面を表示します。
   データ変換（マッピング）定義画面が表示されたあと，［ルート要素選択］ダイアログが表示されます。
2. すべてのスキーマ論理名のルート要素を選択します。
   ［OK］ボタンがクリックできる状態になります。
3. ［OK］ボタンをクリックします。
   変換元スキーマツリービューアおよび変換先スキーマツリービューアに，選択したルート要素のスキーマがツリー表示されます。
4. マッピングします。
   マッピングでは次の作業を行います。
   ・ファンクションの配置とファンクションの設定
   　ファンクションの設定内容を次の表に示します。

表 D–70　ファンクションの設定内容（CSV データ書き込みサンプル-DB アダプタ要求電文作成アクティビティ）

ファンクション	ファンクション名	設定値	
const	const1	定数種別	文字列
		値	PRODUCT_CODE
const	const2	定数種別	文字列
		値	=

   ・ノード間のマッピングの設定
   　ノード間のマッピングの設定内容を次の表に示します。

表 D-71　マッピングの設定内容（CSV データ書き込みサンプル-DB アダプタ要求電文作成アクティビティ）

変換元ノード	ファンクション	変換先ノード
−	const1	val1
−	const2	val2
PRODUCT_CODE	−	val3

（凡例）

−：該当しません。

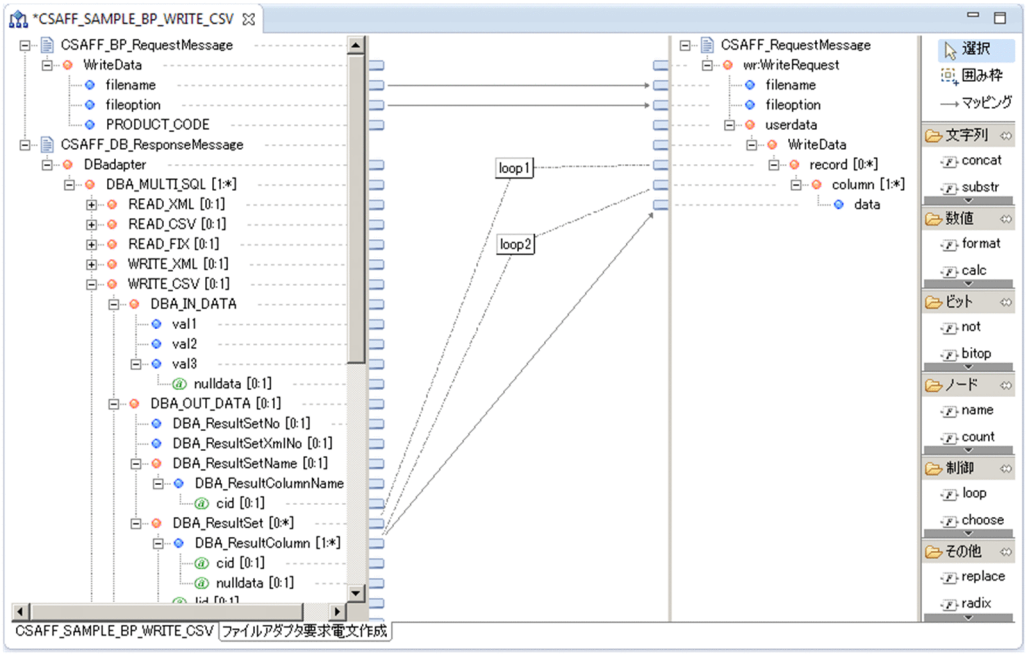
5. データ変換（マッピング）定義画面の変換元スキーマツリービューア，マッピングビューア，または変換先スキーマツリービューアの適当な場所で右クリックして，［検証］を選択します。

   検証が実行されます。

●ファイルアダプタ要求電文作成アクティビティのデータ変換定義の設定内容

ファイルアダプタ要求電文作成アクティビティのデータ変換定義を設定します。設定するデータ変換定義を次の図に示します。

図 D-14　ファイルアダプタ要求電文作成アクティビティデータ変換定義画面（CSV データ書き込みサンプル）



ファイルアダプタ要求電文作成アクティビティのデータ変換定義の手順を次に示します。

1. ファイルアダプタ要求電文作成アクティビティからデータ変換（マッピング）定義画面を表示します。

   データ変換（マッピング）定義画面が表示されたあと，［ルート要素選択］ダイアログが表示されます。

2. すべてのスキーマ論理名のルート要素を選択します。

   ［OK］ボタンがクリックできる状態になります。

3. [OK] ボタンをクリックします。

変換元スキーマツリービューアおよび変換先スキーマツリービューアに，選択したルート要素のスキーマがツリー表示されます。

4. マッピングします。

マッピングでは次の作業を行います。

・ファンクションの配置とファンクションの設定

ファンクションの設定内容を次の表に示します。

表 D-72　ファンクションの設定内容（CSV データ書き込みサンプル-ファイルアダプタ要求電文作成アクティビティ）

ファンクション	ファンクション名	設定値	
loop	loop1	ノード選択	DBA_ResultSet※
		ノード条件設定	設定しない
loop	loop2	ノード選択	DBA_ResultColumn※
		ノード条件設定	設定しない

注※

DBA_ResultSet および DBA_ResultColumn は，WRITE_CSV ノード下の情報を使用します。

・ノード間のマッピングの設定

ノード間のマッピングの設定内容を次の表に示します。

表 D-73　マッピングの設定内容（CSV データ書き込みサンプル-ファイルアダプタ要求電文作成アクティビティ）

変換元ノード	ファンクション	変換先ノード
－	loop1	record
－	loop2	column
filename	－	filename
fileoption	－	fileoption
DBA_ResultColumn※	－	data

（凡例）

－：該当しません。

注※

DBA_ResultColumn は，WRITE_CSV ノード下の情報を使用します。

5. データ変換（マッピング）定義画面の変換元スキーマツリービューア，マッピングビューア，または変換先スキーマツリービューアの適当な場所で右クリックして，［検証］を選択します。

検証が実行されます。

●アクティビティの連結

CSV データ書き込みサンプルで使用するビジネスプロセスでは，次の表に示すようにアクティビティを連結します。

表 D-74　アクティビティの連結内容（CSV データ書き込みサンプル）

連結元アクティビティ	連結先アクティビティ
開始	CSAFF_SAMPLE_BP_WRITE_CSV
CSAFF_SAMPLE_BP_WRITE_CSV	DB アダプタ要求電文作成
DB アダプタ要求電文作成	DB アダプタ呼出
DB アダプタ呼出	ファイルアダプタ要求電文作成
ファイルアダプタ要求電文作成	ファイルアダプタ呼出
ファイルアダプタ呼出	CSAFF_SAMPLE_BP_WRITE_CSV_RSP
CSAFF_SAMPLE_BP_WRITE_CSV_RSP	終了

アクティビティの連結の手順を次に示します。

1. パレットの［コネクション］をクリックします。

   コネクションが選択された状態になります。

   この状態で，アクティビティにマウスポインタを合わせると，そのアクティビティを連結元に設定できる場合，アクティビティの背景色が変わります。

2. キャンバスに配置したアクティビティのうち，連結元となるアクティビティをクリックします。

   連結元となるアクティビティが設定されます。

   この状態で，ほかのアクティビティにマウスポインタを合わせると，そのアクティビティを連結先に設定できる場合，アクティビティの背景色が変わります。

3. キャンバスに配置したアクティビティのうち，連結先となるアクティビティをクリックします。

   連結元となるアクティビティと連結先となるアクティビティが連結されます。

(d)　ビジネスプロセスの保存・検証

ビジネスプロセスを保存して検証する手順を次に示します。

1. 定義内容を確認して，Eclipse のメニューから［ファイル］－［保存］を選択し，定義内容を保存します。

   ビジネスプロセスのサービス ID は，プロパティビューで「CSAFF_B5」を設定してください。

2. ビジネスプロセスの設定を検証します。

   ツリービューのサービス定義一覧で右クリックして表示されるサービス一覧のポップアップメニューから，「検証」を選択すると，検証結果がコンソールビューに表示されます。

(3)　ファイルアダプタとビジネスプロセスの配備定義

定義したファイルアダプタとビジネスプロセスをサーバに配備して開始します。

ファイルアダプタとビジネスプロセスの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，［複数サービスをサーバに配備して開始］を選択します。
2. サービス選択ページで配備および開始するサービスを選択します。
3. [Finish] 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。
4. [OK] ボタンをクリックします。
   ファイルアダプタとビジネスプロセスの配備と開始が完了します。

(4)　テーブルの作成

サンプルを実行するデータベースのテーブルを作成します。

HiRDB SQL Executer を起動して，次の SQL 文を実行してください。

```
CREATE TABLE CSAFF_SAMPLE_WC
(ORDER_NO INT,
 CUSTOMER_CODE CHAR(5),
 PRODUCT_CODE CHAR(4),
 ORDER_COUNT INT);
INSERT INTO CSAFF_SAMPLE_WC VALUES(1001,'AA001','1001',5);
INSERT INTO CSAFF_SAMPLE_WC VALUES(1002,'AB002','1001',1);
INSERT INTO CSAFF_SAMPLE_WC VALUES(1003,'AA001','1102',3);
INSERT INTO CSAFF_SAMPLE_WC VALUES(1004,'XA005','1103',1);
INSERT INTO CSAFF_SAMPLE_WC VALUES(1005,'AA001','1105',1);
```

注意事項

ユーザ ID は「ADMIN」とします。

(5)　要求電文の作成

サンプルで提供するビジネスプロセス実行時の要求電文を編集して，要求電文を作成する手順を次に示します。

1. サンプルで提供する要求電文ファイルを，テキストエディタで開きます。
   サンプルで提供する要求電文ファイルは，「<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¥custom-adapter¥File¥sample¥write_csv¥write_csv_message.xml」です。

2. 次に示す斜体部分を記述します。

```
<?xml version="1.0" encoding="UTF-8"?>
<WriteData>
  <filename><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¥custom-adapter¥File¥
sample¥write_csv¥write_csv_data.csv</filename>
  <fileoption>mode=new</fileoption>
  <PRODUCT_CODE>1001</PRODUCT_CODE>
</WriteData>
```

3. ファイルを上書き保存します。

(6) ビジネスプロセスの実行

サンプルを実行するには，サービスリクエスタからビジネスプロセスを起動します。

サービスリクエスタの処理の流れと概要を次に示します。

1. サービスリクエスタ実行
   ビジネスプロセスの要求電文が，そのままファイルアダプタの要求電文として使用されます。
2. DB アダプタ実行
   要求電文で指定された条件に合致するデータがデータベースから取得されます。
3. ファイルアダプタ電文作成
   DB アダプタの応答電文がファイルアダプタの要求電文の形式に変換されます。
4. ファイルアダプタ実行
   要求電文で指定されたファイルが出力されます。
5. サービスリクエスタ終了
   ファイルアダプタの応答電文が，そのままビジネスプロセスの応答電文として使用されます。

サービスリクエスタの実行手順を次に示します。

1. カレントディレクトリを「<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¥custom-adapter¥File¥sample¥bin」に移動します。

2. request.bat を実行します。
   次の形式で実行してください。

```
request write_csv <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¥custom-adapter
¥File¥sample¥write_csv¥write_csv_message.xml
```

参考

出力先は，次のファイルです。

<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¥custom-adapter¥File¥sample¥write_csv¥write_csv_data.csv

付録 E　ファイル操作アダプタの定義例

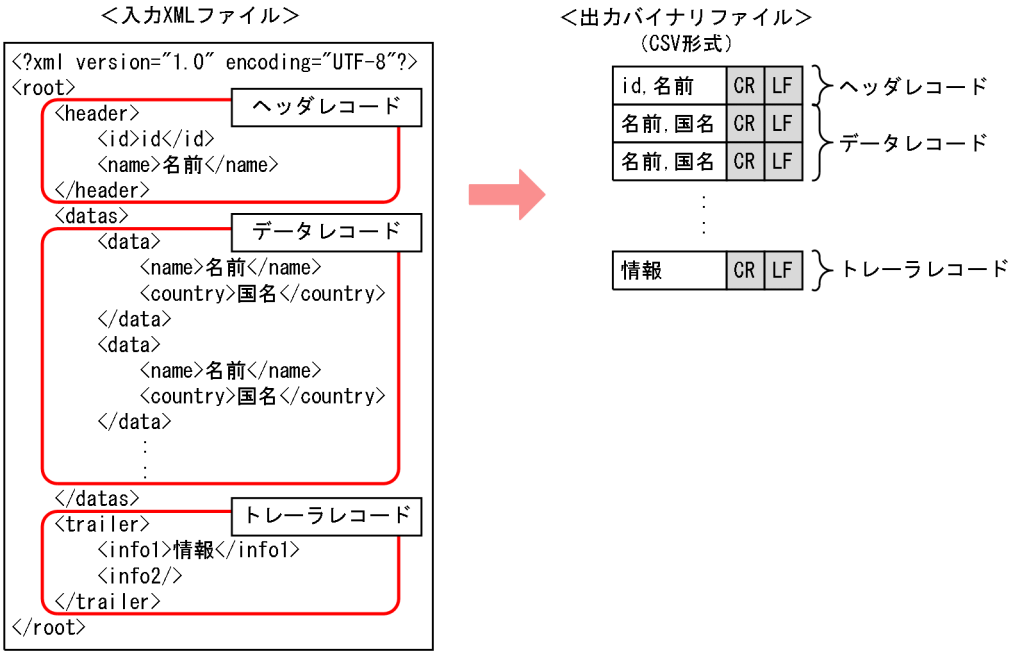
ここでは，ファイル操作アダプタでファイル変換オペレーションを使用する場合の電文フォーマットおよび独自定義ファイルの定義例について説明します。

付録 E.1　XML 形式からバイナリ形式に分割処理方式で変換する場合

XML 形式からバイナリ形式に分割処理方式でファイルを変換する場合の定義例について説明します。

この例で使用する入力 XML ファイルおよび出力バイナリファイルの例を次の図に示します。

図 E-1　入力 XML ファイルおよび出力バイナリファイルの例



この例では，ヘッダレコードとトレーラレコードを使用します。また，分割処理方式で変換するレコードは 1 レコードずつとします。

変換に必要なファイルの一覧を次の表に示します。

表 E-1　変換に必要なファイルの一覧（XML 形式からバイナリ形式に変換する場合）

#	ファイル種別	形式
1	入力 XML スキーマ	XSD
2	出力ヘッダレコード用バイナリフォーマット定義ファイル	FDX
3	出力データレコード用バイナリフォーマット定義ファイル	FDX
4	出力トレーラレコード用バイナリフォーマット定義ファイル	FDX
5	ヘッダレコード用データ変換定義ファイル	XSL

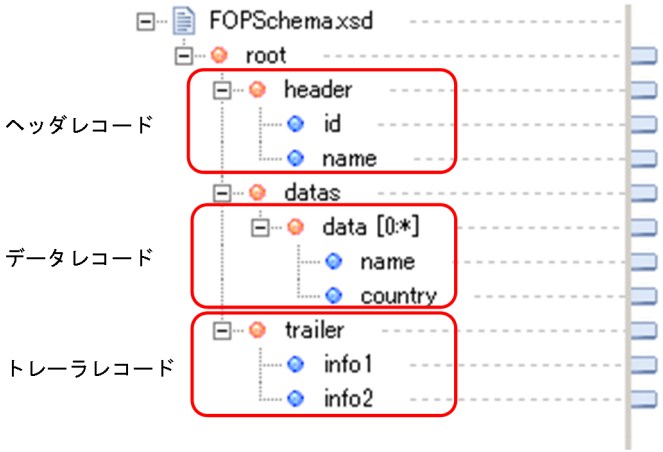
#	ファイル種別	形式
6	データレコード用データ変換定義ファイル	XSL
7	トレーラレコード用データ変換定義ファイル	XSL
8	ファイル操作アダプタ定義ファイル	プロパティファイル

バイナリ形式のファイルを出力する場合，各レコード種別に対応するバイナリフォーマット定義ファイル（#2，#3，#4），および各レコード種別に対応するデータ変換定義ファイル（#5，#6，#7）を作成します。

バイナリフォーマット定義ファイルの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.4　電文フォーマット（バイナリフォーマット定義ファイル）の作成方法」を参照してください。また，データ変換定義ファイルの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.3.1　データ変換の定義手順」の，Eclipse のメニューから表示する場合の説明を参照してください。

1. 入力 XML スキーマ（XSD ファイル）

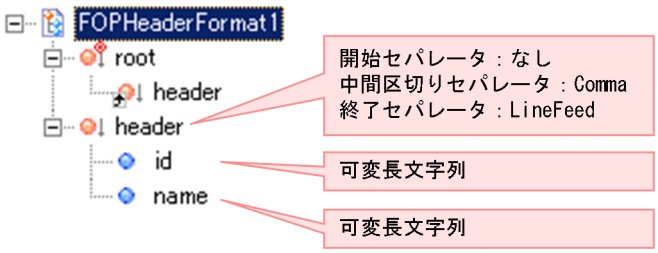
   Eclipse が提供する WST（Web Standard Tools）などを利用して，次の内容の入力 XML スキーマを作成します。入力 XML スキーマのファイル名は「FOPSchema.xsd」とします。



   図は，マッピング定義ファイル作成時に入力 XML スキーマを変換元スキーマとして読み込んだ状態です。

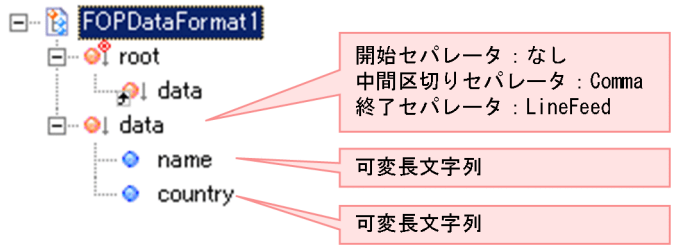
2. 出力ヘッダレコード用バイナリフォーマット定義ファイル（FDX ファイル）

   バイナリフォーマット定義ファイルを次のように作成します。バイナリフォーマット定義ファイルの名称は「FOPHeaderFormat1.fdx」とします。



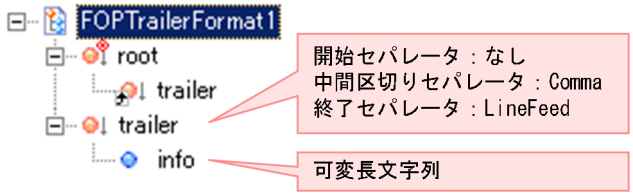
3. 出力データレコード用バイナリフォーマット定義ファイル（FDX ファイル）

バイナリフォーマット定義ファイルを次のように作成します。バイナリフォーマット定義ファイルの名称は「FOPDataFormat1.fdx」とします。



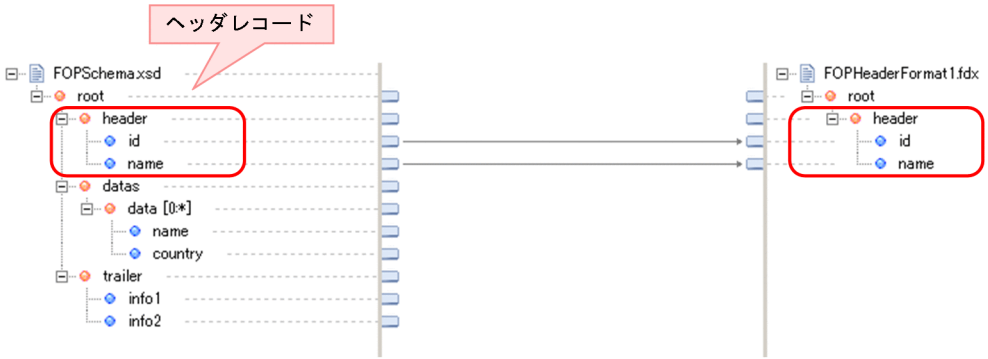
4. 出力トレーラレコード用バイナリフォーマット定義ファイル（FDX ファイル）

バイナリフォーマット定義ファイルを次のように作成します。バイナリフォーマット定義ファイルの名称は「FOPTrailerFormat1.fdx」とします。



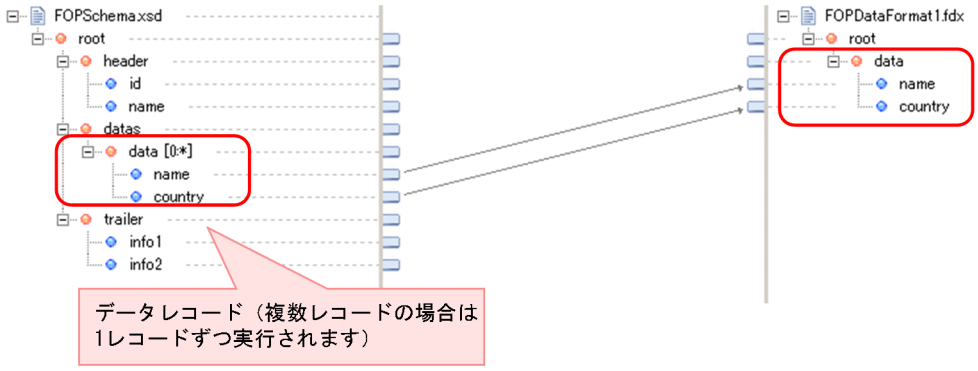
5. ヘッダレコード用データ変換定義ファイル（XSL ファイル）

データ変換定義ファイルを次のように作成します。XSL ファイルの名称は「FOPHeader1.xsl」とします。新規マッピング定義ファイルの作成時，変換元スキーマに「FOPSchema.xsd」を，変換先スキーマに「FOPHeaderFormat1.fdx」を指定します。



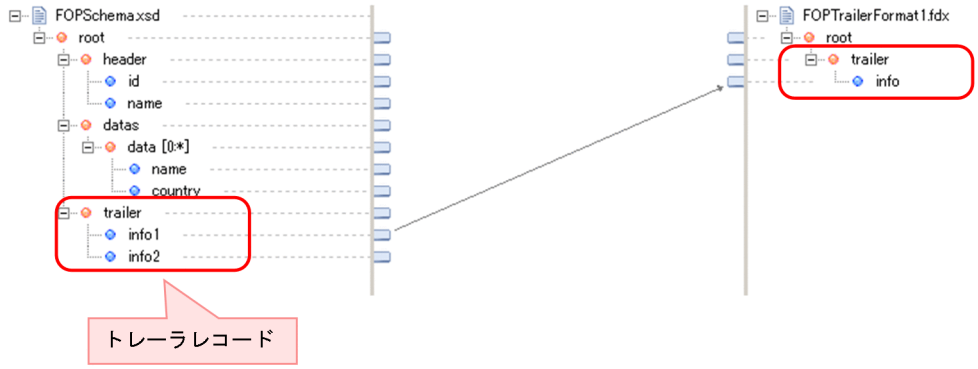
6. データレコード用データ変換定義ファイル（XSL ファイル）

データ変換定義ファイルを次のように作成します。XSL ファイルの名称は「FOPData1.xsl」とします。新規マッピング定義ファイルの作成時，変換元スキーマに「FOPSchema.xsd」を，変換先スキーマに「FOPDataFormat1.fdx」を指定します。



7. トレーラレコード用データ変換定義ファイル（XSL ファイル）

データ変換定義ファイルを次のように作成します。XSL ファイルの名称は「FOPTrailer1.xsl」とします。新規マッピング定義ファイルの作成時，変換元スキーマに「FOPSchema.xsd」を，変換先スキーマに「FOPTrailerFormat1.fdx」を指定します。



8. ファイル操作アダプタ定義ファイル

サービスアダプタ定義画面（詳細）で［独自定義ファイル］の cscFileOperation.properties を選択し，［編集］ボタンをクリックして次のように定義します。

```
#分割処理方式の指定
csc.adapter.fileOperation.transform.splitLoad = ON

#ヘッダレコード，トレーラレコードの有無を指定
csc.adapter.fileOperation.transform.headerRecord = ON
csc.adapter.fileOperation.transform.trailerRecord = ON

#入力XMLファイルの定義
csc.adapter.fileOperation.transform.input = xml
csc.adapter.fileOperation.transform.header.inElement = /root/header
csc.adapter.fileOperation.transform.data.inElement = /root/datas/data
csc.adapter.fileOperation.transform.trailer.inElement = /root/trailer

#出力バイナリファイルの定義
csc.adapter.fileOperation.transform.output = non-xml
csc.adapter.fileOperation.transform.header.outFormat = FOPHeaderFormat1.fdx
csc.adapter.fileOperation.transform.data.outFormat = FOPDataFormat1.fdx
csc.adapter.fileOperation.transform.trailer.outFormat = FOPTrailerFormat1.fdx

#各レコード種別に対応するXSLファイルの定義
csc.adapter.fileOperation.transform.header.styleSheet = FOPHeader1.xsl
csc.adapter.fileOperation.transform.data.styleSheet = FOPData1.xsl
csc.adapter.fileOperation.transform.trailer.styleSheet = FOPTrailer1.xsl
```

ファイル操作アダプタ定義ファイルの詳細はマニュアル「サービスプラットフォーム リファレンス」の「3.5.2　ファイル操作アダプタ定義ファイル」を参照してください。

9. サービスアダプタの定義

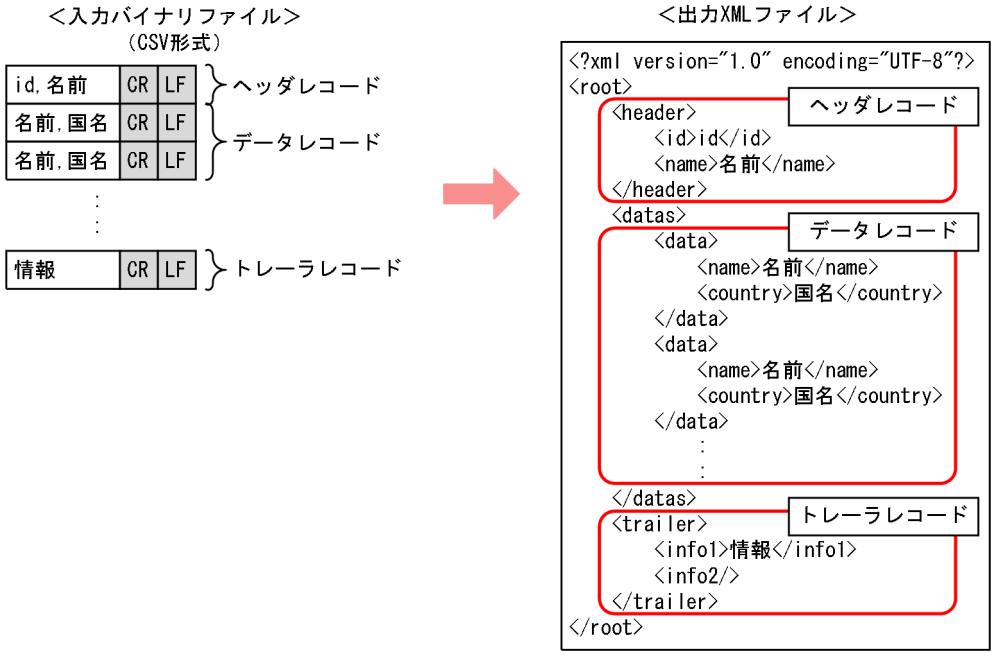
サービスアダプタ定義画面（詳細）で，作成した次のファイルを独自定義ファイルとして追加します。

- FOPHeaderFormat1.fdx
- FOPDataFormat1.fdx
- FOPTrailerFormat1.fdx
- FOPHeader1.xsl
- FOPData1.xsl
- FOPTrailer1.xsl

付録 E.2　バイナリ形式から XML 形式に分割処理方式で変換する場合

バイナリ形式から XML 形式に分割処理方式でファイルを変換する場合の定義例について説明します。

図 E-2　入力バイナリファイルおよび出力 XML ファイルの例



この例では，ヘッダレコードとトレーラレコードを使用します。また，分割処理で変換するレコードは 1 レコードずつとします。

変換に必要なファイルの一覧を次の表に示します。

表 E-2　変換に必要なファイルの一覧（バイナリ形式から XML 形式に変換する場合）

#	ファイル種別	形式
1	入力ヘッダレコード用バイナリフォーマット定義ファイル	FDX
2	入力データレコード用バイナリフォーマット定義ファイル	FDX

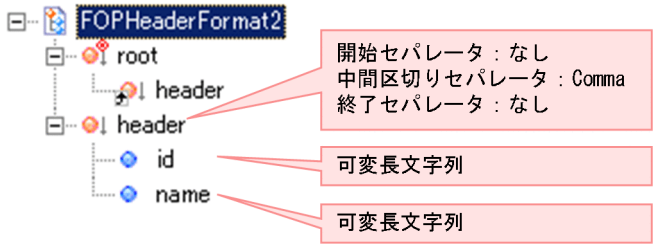
#	ファイル種別	形式
3	入力トレーラレコード用バイナリフォーマット定義ファイル	FDX
4	出力 XML スキーマ	XSD
5	ヘッダレコード用データ変換定義ファイル	XSL
6	データレコード用データ変換定義ファイル	XSL
7	トレーラレコード用データ変換定義ファイル	XSL
8	出力用テンプレート XML ファイル	XML
9	ファイル操作アダプタ定義ファイル	プロパティファイル

バイナリ形式のファイルを入力する場合，各レコード種別に対応するバイナリフォーマット定義ファイル（#1，#2，#3）が必要となります。ファイル入力時に行末の改行コードが削除されているため，終了セパレータの指定が不要となることに注意してください。また，各レコード種別に対応するデータ変換定義ファイル（#5，#6，#7）を作成します。

バイナリフォーマット定義ファイルの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「4.4　電文フォーマット（バイナリフォーマット定義ファイル）の作成方法」を参照してください。また，データ変換定義ファイルの作成方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.3.1　データ変換の定義手順」の，Eclipse のメニューから表示する場合の説明を参照してください。

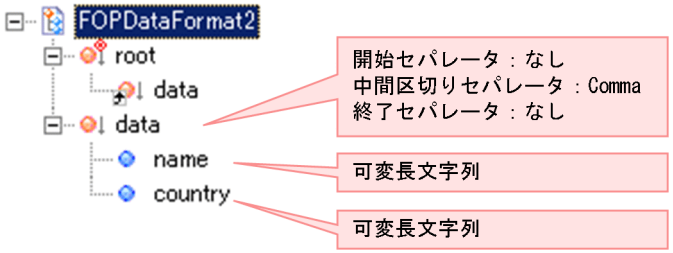
1. 入力ヘッダレコード用バイナリフォーマット定義ファイル（FDX ファイル）

   バイナリフォーマット定義ファイルを次のように作成します。バイナリフォーマット定義ファイルの名称は「FOPHeaderFormat2.fdx」とします。



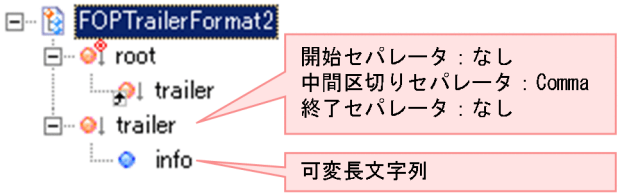
2. 入力データレコード用バイナリフォーマット定義ファイル（FDX ファイル）

   バイナリフォーマット定義ファイルを次のように作成します。バイナリフォーマット定義ファイルの名称は「FOPDataFormat2.fdx」とします。



3. 入力トレーラレコード用バイナリフォーマット定義ファイル（FDX ファイル）

バイナリフォーマット定義ファイルを次のように作成します。バイナリフォーマット定義ファイルの名称は「FOPTrailerFormat2.fdx」とします。

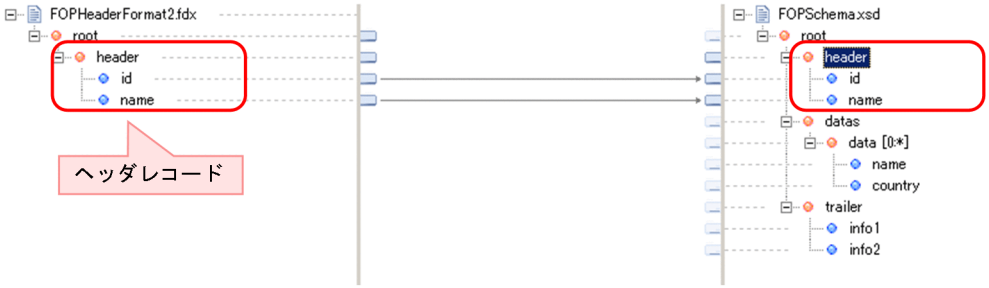


4. 出力 XML スキーマ

ファイル名および内容は「付録 E.1　XML 形式からバイナリ形式に分割処理方式で変換する場合」の入力 XML スキーマと同じです。

5. ヘッダレコード用データ変換定義ファイル（XSL ファイル）

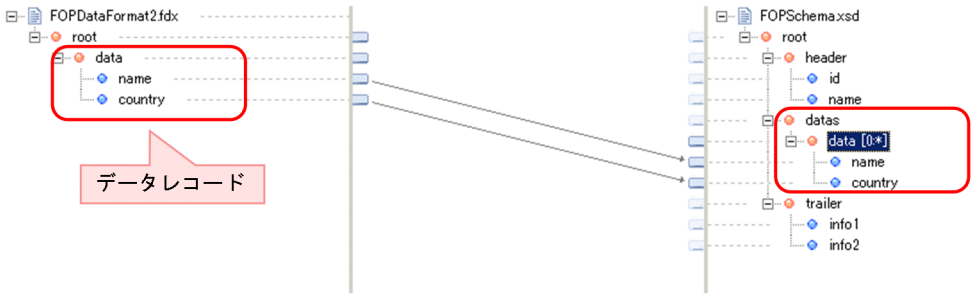
データ変換定義ファイルを次のように作成します。XSL ファイルの名称は「FOPHeader2.xsl」とします。変換元スキーマに「FOPHeaderFormat2.fdx」を，変換先スキーマに「FOPSchema.xsd」を指定します。



この例では，マッピング対象のノードを header 要素に限定しています。マッピング対象の範囲を限定する方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.4.8 マッピング対象の範囲を限定する」を参照してください。

6. データレコード用データ変換定義ファイル（XSL ファイル）

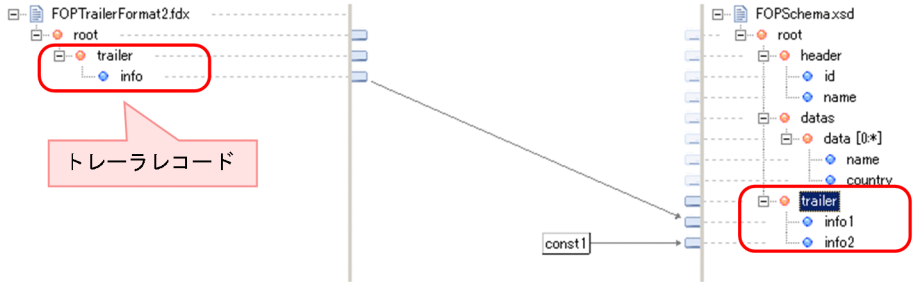
データ変換定義ファイルを次のように作成します。XSL ファイルの名称は「FOPData2.xsl」とします。変換元スキーマには「FOPDataFormat2.fdx」を，変換先スキーマには「FOPSchema.xsd」を指定します。



この例では，マッピング対象のノードを data 要素に限定しています。

7. トレーラレコード用データ変換定義ファイル（XSL ファイル）

データ変換定義ファイルを次のように作成します。XSL ファイルの名称は「FOPTrailer2.xsl」とします。変換元スキーマには「FOPTrailerFormat2.fdx」を，変換先スキーマには「FOPSchema.xsd」を指定します。



この例では，定数ファンクションを使用してマッピング対象のノードを info1 要素に限定しています。

8. 出力用テンプレート XML ファイル

Eclipse が提供する WST（Web Standard Tools）などを利用して，出力用テンプレート XML ファイルを作成します。ファイル名は「xml_output_template.xml」とします。

次の出力用テンプレート XML ファイル内の各レコードを，変換された各レコードに差し替えて出力します。出力テンプレート用 XML ファイルは，レコードの出力 XML スキーマに沿ったものを作成してください。出力 XML スキーマに沿っていない場合，変換されたレコードを出力できません。

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <header/>    ヘッダレコード
  <datas>
    <data/>    データレコード
  </datas>
  <trailer/>   トレーラレコード
</root>
```

9. ファイル操作アダプタ定義ファイル

サービスアダプタ定義画面（詳細）で［独自定義ファイル］の cscFileOperation.properties を選択し，［編集］ボタンをクリックして次のように定義します。

```
#分割処理方式の指定
csc.adapter.fileOperation.transform.splitLoad = ON

#改行セパレータ方式の指定
csc.adapter.fileOperation.transform.format = separate

#ヘッダレコード，トレーラレコードの有無の指定
csc.adapter.fileOperation.transform.headerRecord = ON
csc.adapter.fileOperation.transform.trailerRecord = ON

#入力バイナリファイルの定義
csc.adapter.fileOperation.transform.input = non-xml
csc.adapter.fileOperation.transform.header.inFormat = FOPHeaderFormat2.fdx
csc.adapter.fileOperation.transform.data.inFormat = FOPDataFormat2.fdx
csc.adapter.fileOperation.transform.trailer.inFormat = FOPTrailerFormat2.fdx

#出力XMLファイルの定義
csc.adapter.fileOperation.transform.output = xml
csc.adapter.fileOperation.transform.outputTemplateXmlFile = xml_output_template.xml
csc.adapter.fileOperation.transform.header.outElement = /root/header
csc.adapter.fileOperation.transform.data.outElement = /root/datas/data
csc.adapter.fileOperation.transform.trailer.outElement = /root/trailer

#各レコード種別に対応するXSLファイルの定義
csc.adapter.fileOperation.transform.header.styleSheet = FOPHeader2.xsl
csc.adapter.fileOperation.transform.data.styleSheet = FOPData2.xsl
csc.adapter.fileOperation.transform.trailer.styleSheet = FOPTrailer2.xsl
```

10. サービスアダプタの定義

サービスアダプタ定義画面（詳細）で，作成した次のファイルを独自定義ファイルとして追加します。

- FOPHeaderFormat2.fdx
- FOPDataFormat2.fdx
- FOPTrailerFormat2.fdx
- FOPHeader2.xsl
- FOPData2.xsl
- FOPTrailer2.xsl
- xml_output_template.xml

付録 F　HTTP 受付を利用したビジネスプロセスの設定例

ここでは，HTTP 受付に渡されたパラメタ（時速および時間）に対して，進んだ距離を応答として返すサンプルを使った HTTP 受付およびビジネスプロセスの設定例を紹介します。

次の HTTP 受付の設定例について，それぞれ説明します。

- 標準モードの設定例
- パススルーモードの設定例
- ルーティング機能の設定例

付録 F.1　HTTP 受付の設定例（標準モードの場合）

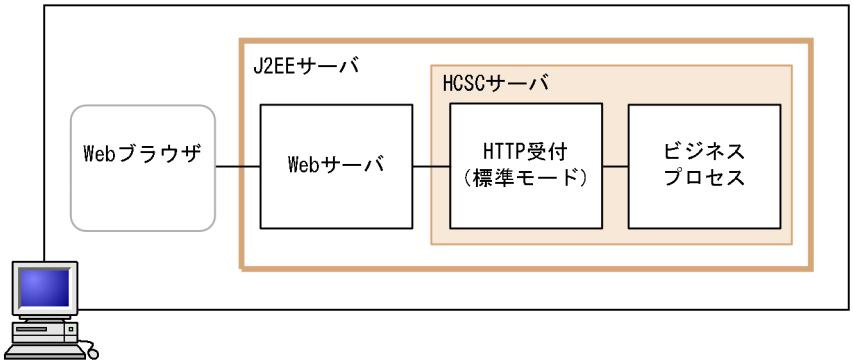
ビジネスプロセスを標準モードで実行する場合の設定例について説明します。

(1)　設定例の概要

この設定例で使用する構成と処理の流れについて説明します。

設定例の構成を次の図に示します。

図 F-1　設定例の構成（標準モードの場合）



この設定例では，HCSC サーバ環境を構築したマシンと同一マシン上の Web ブラウザを使用します。また，J2EE サーバ内の Web サーバを使用します。なお，HTTP 受付は標準モードを使用します。

設定例の処理の流れを次に示します。

1. Web ブラウザから GET メソッドで HTTP 受付を呼び出します。
   このとき，HTTP リクエストのクエリ文字列に時速と時間を設定します。
2. ビジネスプロセス内のデータ変換定義で距離を計算して，応答電文（ボディ）にセットします。
3. HTTP レスポンスヘッダ定義ファイルに指定されたヘッダを設定して，Web ブラウザに応答を返します。

なお，この設定例ではヘッダ割当変数を使用しません。以降で設定手順について説明します。

(2) HTTP 受付の電文フォーマットの作成

HTTP 受付で使用する電文フォーマットを作成します。電文フォーマットのテンプレートは次のフォルダに格納されています。

- テンプレートファイルの格納元

```
<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¥custom-reception¥http¥schema
```

フォルダごと任意の場所にコピーしてから編集してください。電文フォーマットを作成する場合，コピー先のファイルを使用します。

(a) 要求電文（ボディ）フォーマットの作成

受付時の HTTP 受付のボディ割当変数に設定する電文フォーマットを作成します。電文フォーマットはテンプレートを編集して作成します。

編集するファイルは「urecp_http_body_detail_request.xsd」です。

注　urecp_http_body_request.xsd は編集しないでください。

この設定例では，次のように時速を表す要素と時間を表す要素（斜体の個所）を追加します。

- urecp_http_body_detail_request.xsd

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema elementFormDefault="qualified"
  targetNamespace="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/reques
t"
  xmlns:hrc="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/request"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:complexType name="http-body-requestType">
      <xsd:sequence>
        <!-- User Customize -->
        <xsd:element name="kilometers_per_hour" type=" xsd:positiveInteger "/>
        <xsd:element name="hours" type=" xsd:positiveInteger "/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:schema>
```

なお，この例では HTTP 受付ヘッダ割当変数は使用しません。

(b) 応答電文（ボディ）フォーマットの作成

応答時の HTTP 受付のボディ割当変数に設定する電文フォーマットを作成します。

距離を返す処理を記述した次の電文フォーマットを作成し，コピーしたフォルダ内に保存してください。ファイル名は「urecp_http_body_response.xsd」です。なお，応答電文（ボディ）フォーマットのテンプレートファイルはありません。

- urecp_http_body_response.xsd

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="HTTPResponseBody ">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="distance" type=" xsd:positiveInteger "/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

(3) HTTP 受付の追加

呼び出しサービス選択機能を使用しないで，HTTP 受付を追加する手順を次に示します。

1. ツリービューのサービス定義一覧で右クリックして［ビジネスプロセス追加］を選択します。
   ビジネスプロセスが追加されます。ビジネスプロセスの内容はここでは定義しません。
2. ツリービューのサービス定義一覧で，追加したビジネスプロセスを選択して右クリックし，［ユーザ定義受付（呼出先固定）追加］を選択します。
   受付種別選択ウィザードが表示されます。
3. 受付種別で［HTTP 受付］を選択して［次へ］ボタンをクリックします。
4. ［受付名］に任意の名前を設定します。
5. ［終了］ボタンをクリックします。
   HTTP 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。

(4) HTTP 受付の定義

ユーザ定義受付定義画面で設定する内容を次に示します。なお，ユーザ定義受付定義画面の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「1.2.5　ユーザ定義受付定義画面」を参照してください。

(a) ユーザ定義受付定義画面（基本）の設定

設定する内容を次に示します。

表 F-1　ユーザ定義受付定義画面（基本）の設定内容

分類	項目	設定値
ユーザ定義受付情報	受付名	HTTPReception
	受付 ID	rcp1
	受付種別	HTTP 受付

分類	項目		設定値
ユーザ定義受付情報	デフォルトオペレーション名		calculateDistance
	オペレーション		calculateDistance
オペレーション情報	オペレーション名		calculateDistance
	通信モデル		同期
要求電文（ボディ）	any 型を使う（チェックボックス）		使わない（チェックしない）
	受付	電文フォーマット	urecp_http_body_request.xsd
	サービス部品	使う（チェックボックス）	使わない（チェックしない）
		電文フォーマット	−
	データ変換定義		−
応答電文(ボディ)	any 型を使う（チェックボックス）		使わない（チェックしない）
	受付	電文フォーマット	urecp_http_body_response.xsd（「(2) HTTP 受付の電文フォーマットの作成」で作成したファイル）
	サービス部品	使う（チェックボックス）	使わない（チェックしない）
		電文フォーマット	−
	データ変換定義		−

（凡例）

−：設定しません。

(b) HTTP レスポンスヘッダ定義ファイルの編集

HTTP レスポンスヘッダ定義ファイルの編集手順を次に示します。なお，この設定例ではビジネスプロセス中でヘッダ割当変数を使用しないため，プロパティの説明は省略します。

1. ユーザ定義受付定義画面（詳細）の［独自定義ファイル］で「cscurecphttp_header.properties」を選択し，［編集］ボタンをクリックします。

   独自定義ファイルを編集するためのエディタが起動します。

2. HTTP レスポンスヘッダ定義ファイルの内容を次のように修正します。

```
Connection=close
extension-header=extension-header1
```

   注 「extension-header」は，任意に指定できる拡張ヘッダを想定して設定しています。

3. Eclipse のメニューから［ファイル］−［保存］を選択します。

   HTTP レスポンスヘッダ定義ファイルが保存されます。

(c)　HTTP 受付定義ファイルの編集

HTTP 受付定義ファイルの編集手順を次に示します。

1. ユーザ定義受付定義画面（詳細）の［独自定義ファイル］で「cscurecphttp.properties」を選択し，［編集］ボタンをクリックします。

   独自定義ファイルを編集するためのエディタが起動します。

2. HTTP 受付定義ファイルの内容を次のように修正します。

```
#urecp-http.context-root=
#urecp-http.max-threads=10
#urecp-http.exclusive-threads=0
#urecp-http.queue-size=8192
#urecp-http.pooled-instance.minimum=0
#urecp-http.pooled-instance.maximum=0
#urecp-http.ejb-transaction-timeout=0
#httprecp.switchover.pass-through.mode=false
httprecp.http.charset=UTF-8
httprecp.response.header.filename=cscurecphttp_header.properties
#httprecp.system-exception.status-code=500
#httprecp.response.generate.content-length=true
```

   この例ではコンテキストルート（urecp-http.context-root）を省略しているため，HTTP 受付の受付 ID がデフォルト値として使用されます。

3. Eclipse のメニューから［ファイル］－［保存］を選択します。

   HTTP 受付定義ファイルが保存されます。

(5)　ビジネスプロセスの定義

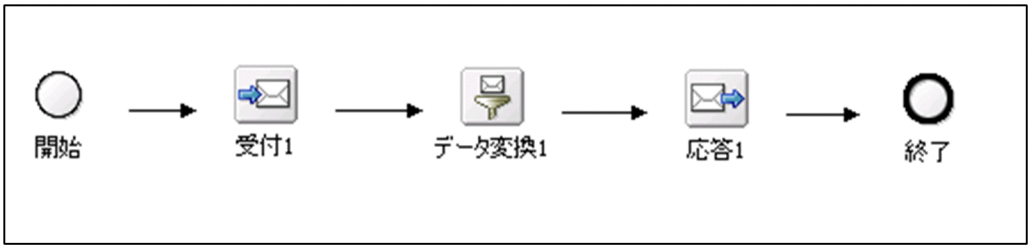
ビジネスプロセスの定義手順を次に示します。

(a)　ビジネスプロセスの作成

この例では，HTTP 受付に渡された要求電文を，データ変換アクティビティを使用してデータ変換し，応答電文を生成するビジネスプロセスを作成します。

次の図に示すようにアクティビティを配置・連結してください。

図 F-2　ビジネスプロセスの作成例



アクティビティを配置・連結する方法については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.4　アクティビティの配置と連結」を参照してください。

(b)　変数の定義

ビジネスプロセスで使用する変数を定義します。

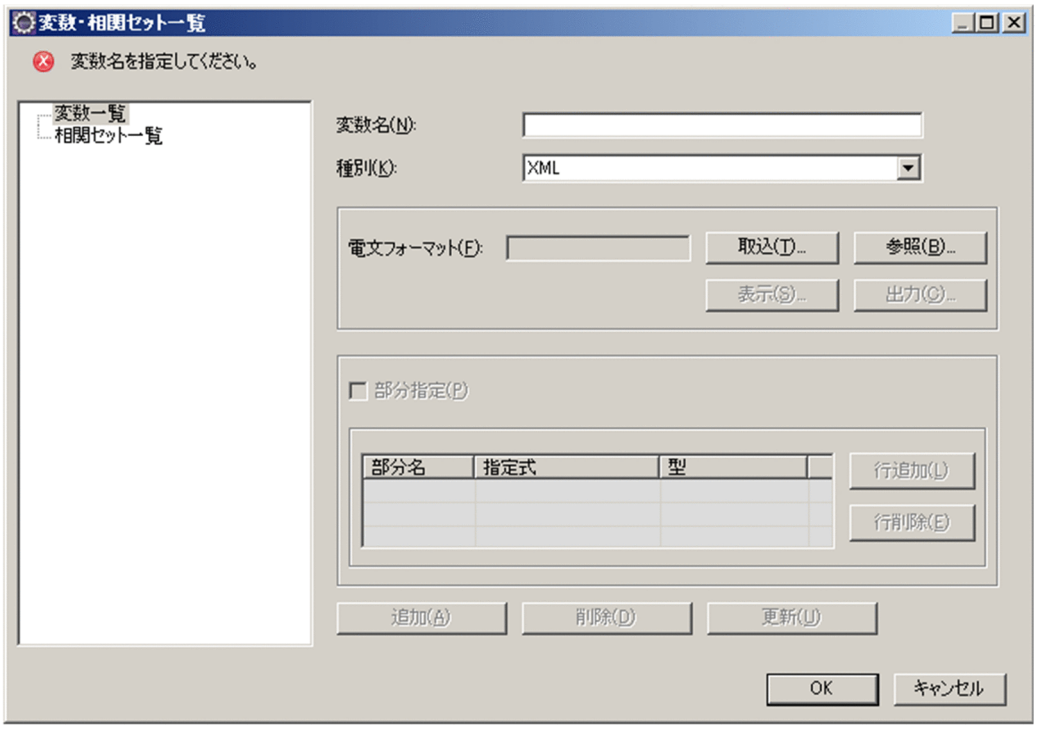
定義する変数を次に示します。

変数名	説明
request_body	HTTP 受付の要求電文（ボディ）の変数
response_body	HTTP 受付の応答電文（ボディ）の変数

なお，この設定例では相関セットを使用しません。

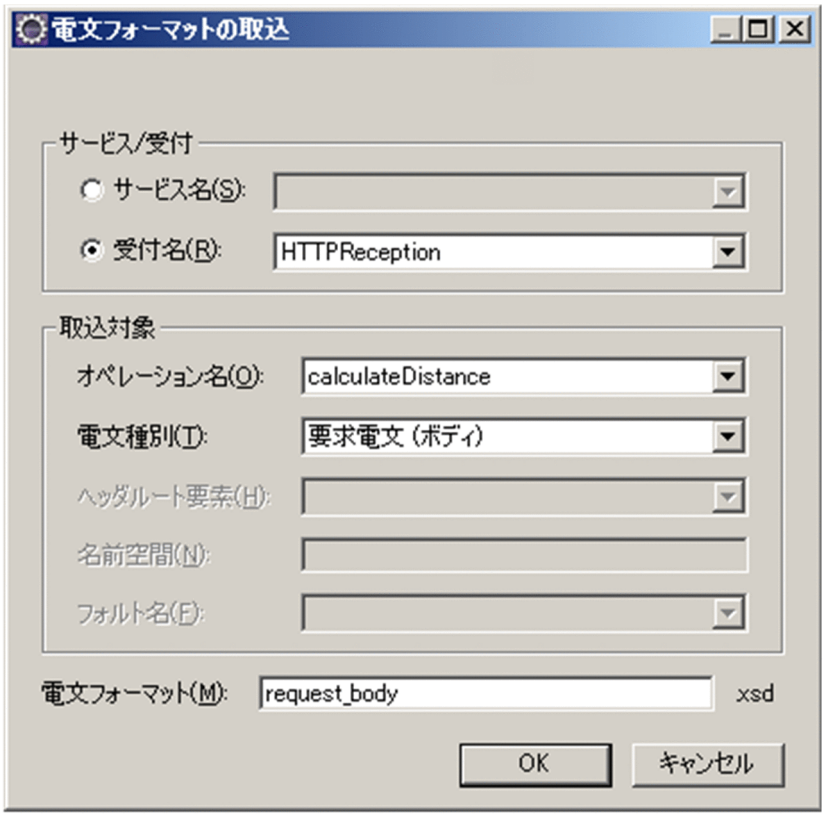
変数の定義は［電文フォーマットの取込］ダイアログで実施します。それぞれの変数の定義手順を次に示します。

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. [変数名] に「request_body」または「response_body」を入力します。
3. [種別] ドロップダウンリストから「XML」を選択します。
4. [電文フォーマット] の［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. 次に示すとおりに各変数を設定します。



- request_body

分類	項目	設定値
サービス/受付	サービス名	−
	受付名	HTTPReception
取込対象	オペレーション名	calculateDistance
	電文種別	要求電文(ボディ)
	フォルト名	−
−	電文フォーマット	request_body

- response_body

分類	項目	設定値
サービス/受付	サービス名	−
	受付名	HTTPReception
取込対象	オペレーション名	calculateDistance
	電文種別	応答電文(ボディ)
	フォルト名	−
−	電文フォーマット	response_body

（凡例）

－：設定しません。または，該当する分類はありません。

6. [OK] ボタンをクリックし，[変数・相関セット一覧] ダイアログで［追加］ボタンをクリックします。

定義した変数が追加されます。

(c) データ変換の定義

データ変換では，数値演算ファンクション「calc1」で要求電文に格納された 2 つのパラメタの積を求め，その値を応答電文にマッピングします。

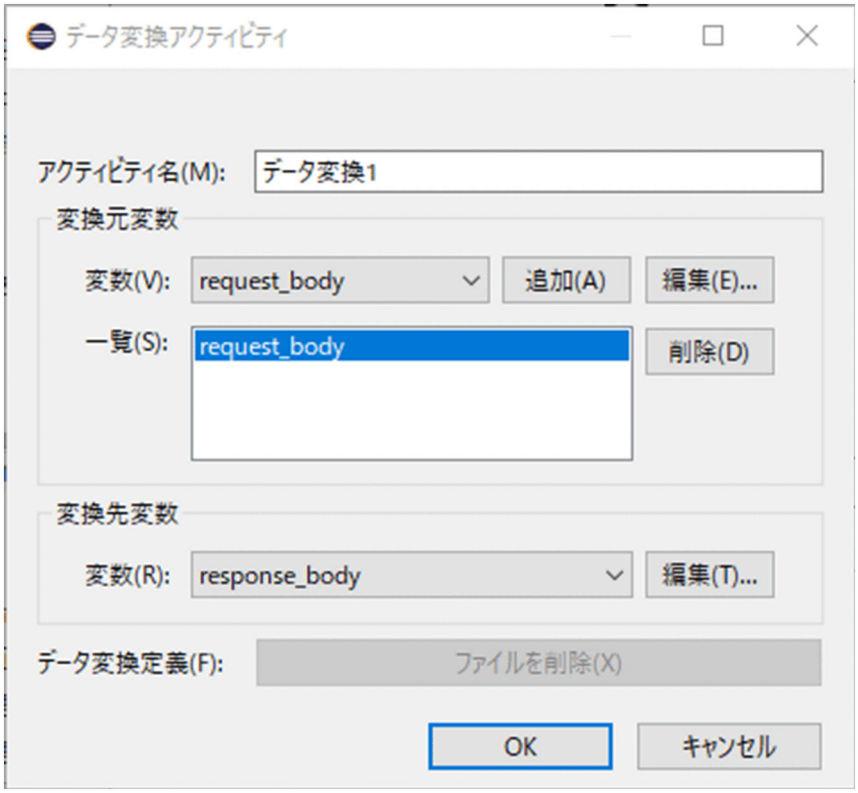
データ変換の定義手順を次に示します。

1. ビジネスプロセス定義画面上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. [データ変換アクティビティ] ダイアログで次のように指定します。

- ［変換元変数］に「request_body」を指定
- ［変換先変数］に「response_body」を指定



3. [OK] ボタンをクリックします。

4. データ変換アクティビティをダブルクリックします。

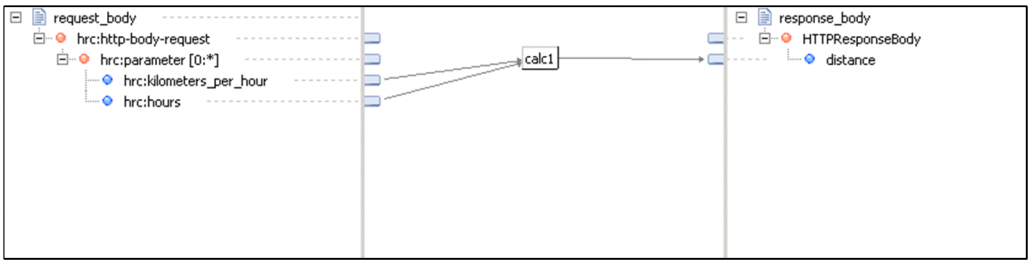
［ルート要素選択］ダイアログが表示されます。

5. [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。

6. パレットから数値演算ファンクション（calc）を選択して，マッピングビューアに配置します。

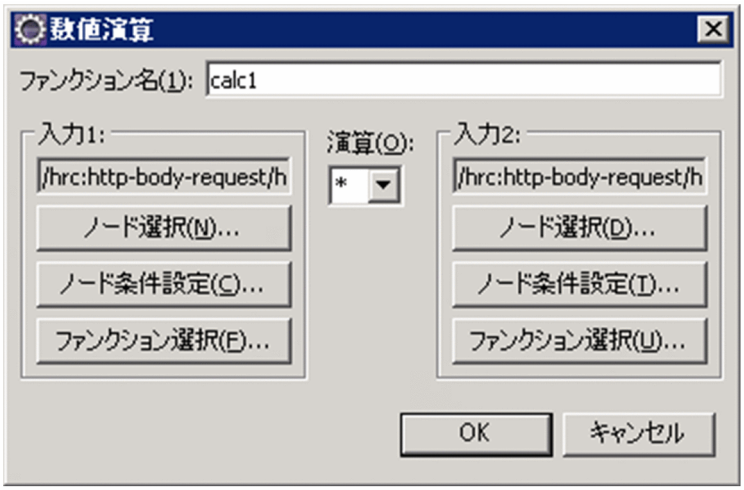
7. マッピング線を図のように設定します。



8. 数値演算ファンクションをダブルクリックします。

［数値演算］ダイアログが表示されます。

9. ［演算］で実行したい演算（この例では「*」）を選択します。



10. ［OK］ボタンをクリックします。

データ変換の定義が完了しました。

(d)　受付アクティビティ・応答アクティビティの設定

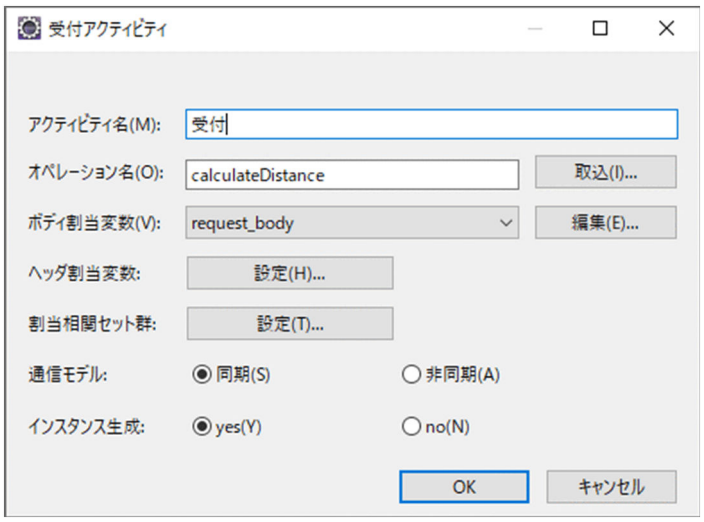
アクティビティの設定手順を次に示します。

●受付アクティビティの設定例

1. ビジネスプロセス定義画面上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. 次のように設定します。



項目	設定値
アクティビティ名	受付
オペレーション名	calculateDistance
ボディ割当変数	request_body
ヘッダ割当変数	－
割当相関セット群	－
通信モデル	同期
インスタンス生成	Yes

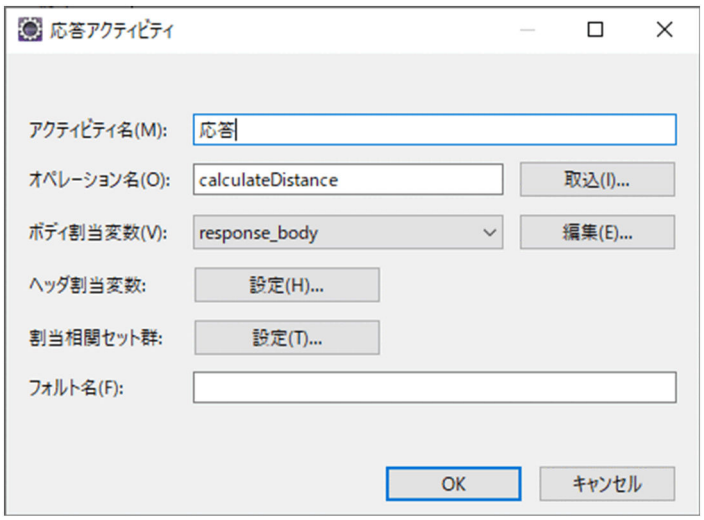
（凡例）

－：設定しません。

3. [OK] ボタンをクリックします。

●応答アクティビティの設定例

1. ビジネスプロセス定義画面上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. 次のように設定します。



項目	設定値
アクティビティ名	応答
オペレーション名	calculateDistance
ボディ割当変数	response_body
ヘッダ割当変数	－
割当相関セット群	－
フォルト名	－

（凡例）

－：設定しません。

3. [OK] ボタンをクリックします。

(6) HTTP 受付の呼び出し

作成したビジネスプロセスを実行するための手順を示します。

(a) 準備手順

1. Eclipse のメニューから，［ファイル］－［すべて保管］を選択し，定義した HCSC コンポーネントを保管します。
2. Windows の［スタート］メニューから，［Cosminexus］－［テストサーバ起動］を選択します。
3. ツリービューのサービス定義一覧で右クリックして，［複数サービスをサーバに配備して開始］を選択します。
4. サービス選択ページで配備および開始するサービスを選択します。
5. [Finish] 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。

6. [OK] ボタンをクリックします。

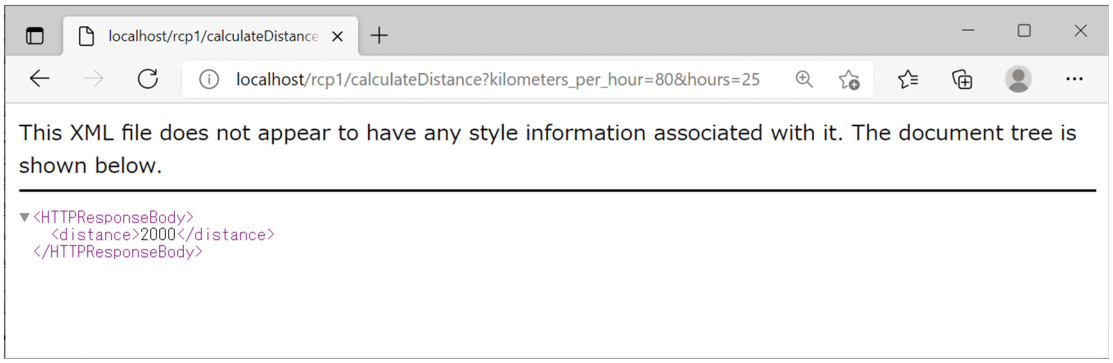
ビジネスプロセスと HTTP 受付を含むリポジトリが実行環境に配備されます。

(b) HTTP 受付の呼び出し

Web ブラウザを起動してアドレス指定部分に次の URL を指定してアクセスします。必要に応じてクエリストリングの指定値は変更してください。

```
http://localhost/rcp1/calculateDistance?kilometers_per_hour=80&hours=25
```

実行結果が次のように Web ブラウザに表示されます。



付録 F.2 HTTP 受付の設定例（パススルーモードの場合）

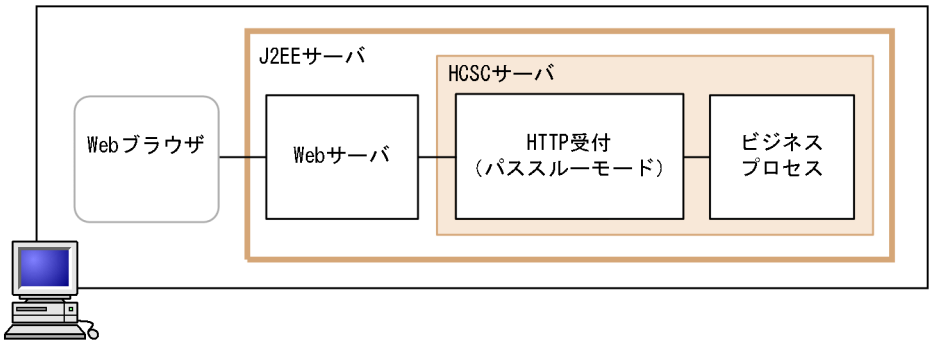
ビジネスプロセスをパススルーモードで実行する場合の設定例について説明します。

(1) 設定例の概要

この設定例で使用する構成と処理の流れについて説明します。

設定例の構成を次の図に示します。

図 F-3 設定するシステムの構成（パススルーモードの場合）



パススルーモードでは，HTTP 受付の要求電文（XML）を Web ブラウザで直接指定してビジネスプロセスを呼び出します。

(2)　HTTP 受付の電文フォーマットの作成

「付録 F.1(2)　HTTP 受付の電文フォーマットの作成」と同様の手順です。

(3)　HTTP 受付の追加

「付録 F.1(3)　HTTP 受付の追加」と同様の手順です。

(4)　HTTP 受付の定義

(a)　ユーザ定義受付定義画面（基本）の設定

「付録 F.1(4)(a)　ユーザ定義受付定義画面（基本）の設定」と同様の設定内容です。

(b)　HTTP レスポンスヘッダ定義ファイルの編集

「付録 F.1(4)(b)　HTTP レスポンスヘッダ定義ファイルの編集」と同様の手順です。

(c)　HTTP 受付定義ファイルの編集

「付録 F.1(4)(c)　HTTP 受付定義ファイルの編集」と同様の手順です。ただし，パススルーモードでは，HTTP 受付定義ファイルの httprecp.switchover.pass-through.mode プロパティに true を指定します。設定例を示します。

```
#urecp-http.context-root=
#urecp-http.max-threads=10
#urecp-http.exclusive-threads=0
#urecp-http.queue-size=8192
#urecp-http.pooled-instance.minimum=0
#urecp-http.pooled-instance.maximum=0
#urecp-http.ejb-transaction-timeout=0
httprecp.switchover.pass-through.mode=true
httprecp.http.charset=UTF-8
httprecp.response.header.filename=cscurecphttp_header.properties
#httprecp.system-exception.status-code=500
#httprecp.response.generate.content-length=true
```

(5)　ビジネスプロセスの定義

「付録 F.1(5)　ビジネスプロセスの定義」と同様の手順です。

(6)　HTTP 受付の呼び出し

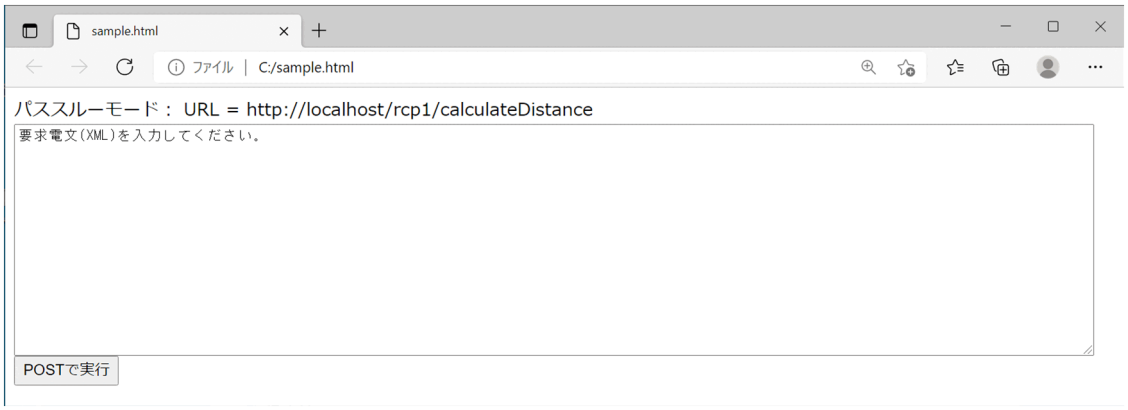
パススルーモードでは，HTTP 受付の要求電文（XML）を Web ブラウザで直接指定します。ここでは次の HTML（POST メソッド）を使用して呼び出します。

1. 次の HTML ファイルを作成して Web ブラウザで表示します。

   - sample.html

```
<html>
  <head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
  <body>
    <form method="POST" action="http://localhost/rcp1/calculateDistance">
      <label>パススルーモード： URL = http://localhost/rcp1/calculateDistance</label><
br>
      <textarea name=msg rows=15 cols=140 wrap=soft>要求電文(XML)を入力してください。<
/textarea><br>
      <input type="submit" value="POSTで実行" />
    </form>
  </body>
</html>
```

- Web ブラウザの表示結果



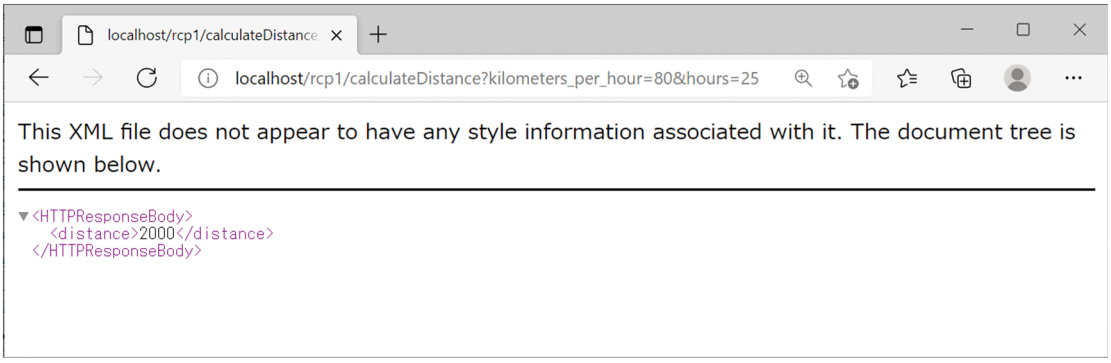
2. 入力エリアに HTTP 受付の要求電文を入力します。

ここでは，標準モードで使用したビジネスプロセスをそのまま使用するため，「付録 F.1(2)　HTTP 受付の電文フォーマットの作成」で定義した標準モード用の要求電文をそのまま指定します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<hrc:http-body-request
 xmlns:hrc="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/request"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/requ
est urecp_http_body_request.xsd">
  <hrc:parameter>
    <hrc:kilometers_per_hour>80</hrc:kilometers_per_hour>
    <hrc:hours>25</hrc:hours>
  </hrc:parameter>
</hrc:http-body-request>
```

3. [POST で実行] ボタンをクリックします。

Web ブラウザに実行結果が表示されます。実行結果を次に示します。



付録 F.3　HTTP 受付のルーティング機能の設定例

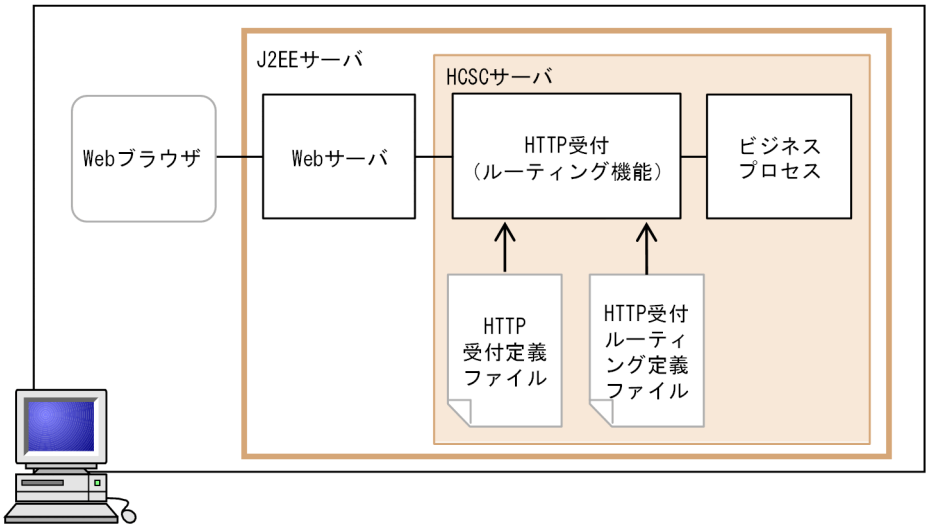
HTTP 受付のルーティング機能を使用して，呼び出すビジネスプロセス，パススルーモードを設定した場合の設定例について説明します。

(1)　設定例の概要

この設定例で使用する構成と処理の流れについて説明します。

設定例の構成を次の図に示します。

図 F-4　設定するシステムの構成（ルーティング機能を設定した場合）



(2)　HTTP 受付の電文フォーマットの作成

「付録 F.1(2)　HTTP 受付の電文フォーマットの作成」と同様の手順です。

(3)　HTTP 受付の追加

「付録 F.1(3)　HTTP 受付の追加」と同様の手順です。

(4)　HTTP 受付の定義

(a)　ユーザ定義受付定義画面（基本）の設定

「付録 F.1(4)(a)　ユーザ定義受付定義画面（基本）の設定」と同様の設定内容です。

(b)　HTTP レスポンスヘッダ定義ファイルの編集

「付録 F.1(4)(b)　HTTP レスポンスヘッダ定義ファイルの編集」と同様の手順です。

(c)　HTTP 受付定義ファイルの編集

「付録 F.1(4)(c)　HTTP 受付定義ファイルの編集」と同様の手順です。ただし，ルーティング機能を使用する場合は，HTTP 受付定義ファイルの httprecp.use-routing-definition プロパティに true を指定します。設定例を示します。

```
httprecp.use-routing-definition=true
```

(d)　HTTP 受付ルーティング定義ファイルの編集

HTTP 受付ルーティング定義ファイルの編集手順を次に示します。

1. ユーザ定義受付定義画面（詳細）の［独自定義ファイル］で「cscurecphttp_routing.xml」を選択し，［編集］ボタンをクリックします。

   独自定義ファイルを編集するためのエディタが起動します。

2. HTTP 受付ルーティング定義ファイルの内容を次のように修正します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<urecphttp-routing xmlns="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http
/rest"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/rest
 urecp_http_rest_definition.xsd">
  <path pattern="/api/books">
    <GET>
      <invoke>bp01</invoke>
      <property key="httprecp.switchover.pass-through.mode">true</property>
      <property key="httprecp.pass-through.parameter-use">true</property>
    </GET>
    <POST><invoke>bp02</invoke></POST>
  </path>
  <path pattern="/api/books/{book-id}">
    <GET><invoke>bp03</invoke></GET>
    <PUT><invoke>bp04</invoke></PUT>
    <DELETE><invoke>bp05</invoke></DELETE>
  </path>
  <path pattern="/api/books/{book-id}/authors/{author-idx}">
    <GET><invoke>bp06</invoke></GET>
    <PUT><invoke>bp07</invoke></PUT>
    <DELETE><invoke>bp08</invoke></DELETE>
```

```
  </path>
</urecphttp-routing>
```

3. Eclipse のメニューから［ファイル］－［保存］を選択します。

   HTTP 受付ルーティング定義ファイルが保存されます。

(5)　ビジネスプロセスの定義

「付録 F.1(5)　ビジネスプロセスの定義」と同様の手順です。

(6)　HTTP 受付の呼び出し

「付録 F.2(6)　HTTP 受付の呼び出し」と同様の手順です。

付録 G　HTTP 受付および HTTP アダプタを利用したビジネスプロセスの設定例

ここでは，次の 2 つの処理を行うシステムを例に，HTTP 受付および HTTP アダプタを使用したビジネスプロセスの設定手順について説明します。

- HTTP サーバから HTTP クライアントにファイルをダウンロードする
- HTTP アダプタの接続先をリクエストごとに動的に変更する

注意事項

この節では，開発環境内の実行環境（テスト環境）でビジネスプロセスを実行することを前提として説明します。

付録 G.1　設定例 1（HTTP サーバから HTTP クライアントにファイルをダウンロードする）

(1)　設定例の概要

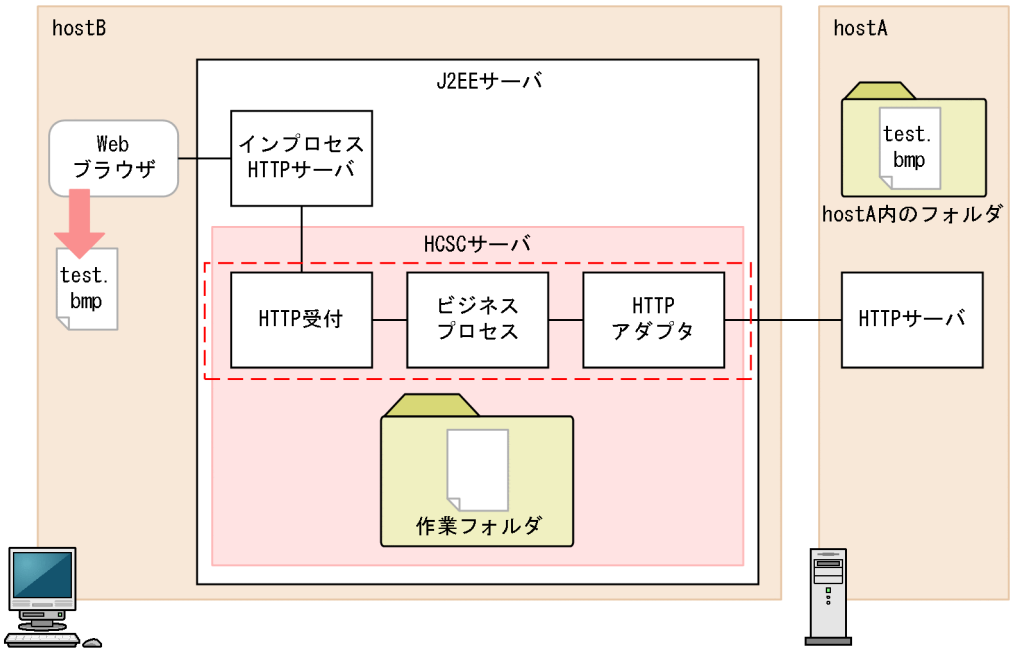
この設定例では，次のような処理を行うビジネスプロセスを定義します。

1. HTTP アダプタから hostA の HTTP サーバにアクセスして，ホスト内のファイル（test.bmp）を作業フォルダ内にダウンロードします。
2. HTTP 受付を経由して作業フォルダから HTTP クライアントに test.bmp をダウンロードします。

(a)　環境構成

この設定例で作成する環境の構成を次の図に示します。

図 G-1　設定例 1 で作成する環境の構成



この設定例では，図中の点線で囲まれた部分の定義手順について説明します。

HTTP クライアントには Web ブラウザを利用します。

ファイルを受信するフォルダには作業フォルダを利用します。

注意事項

この設定例で使用する HTTP サーバは，あらかじめ用意されていることを前提としています。http://hostA/test.bmp にアクセスすると test.bmp をダウンロードできるように HTTP サーバを構成しているものとします。

また，HTTP アダプタがファイルのデータをダウンロードしたことを認識するために，HTTP アダプタに返すレスポンスヘッダには Content-Disposition ヘッダを付与するように設定しておく必要があります。

(b)　設定の流れ

設定の流れを次に示します。

1. 実行環境（テスト環境）のセットアップ
2. 開発環境のセットアップ
3. 各種定義ファイルの設定
4. HTTP アダプタの定義
5. ビジネスプロセスの追加

6. HTTP 受付の追加
7. ビジネスプロセスの定義
8. データ変換の定義
9. ビジネスプロセスを実行するための準備
10. ビジネスプロセスの実行

それぞれの手順について次に説明します。

(2) 実行環境（テスト環境）のセットアップ

HCSC 簡易セットアップ機能を利用して J2EE サーバおよび HCSC サーバを構築します。

1. マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「2.4.2 HCSC 簡易セットアップ機能の実行」に従ってテスト環境を構築します。
   ［メイン］タブの［HCSC サーバ］では，必ず「SOAP1.1/1.2 併用モード」を選択してください。HTTP アダプタを利用するには，「SOAP1.1/1.2 併用モード」を選択していることが前提となります。
   ［メイン］タブの［DB あり/RM なしモデル］，［DB/RM なしモデル］および［DB/RM ありモデル］は，どれを選択しても実行できます。
2. 「3.3.13(1) J2EE サーバ用オプション定義ファイルの設定」に従って，J2EE サーバ用オプション定義ファイル（usrconf.cfg）を設定します。
3. ［スタート］メニューで［プログラム］－［Cosminexus］－［テストサーバ停止］を選択して，テスト環境の J2EE サーバおよび HCSC サーバを停止します。
4. ［スタート］メニューで［プログラム］－［Cosminexus］－［テストサーバ起動］を選択して，テスト環境の J2EE サーバおよび HCSC サーバを再起動します。

(3) 開発環境のセットアップ

開発環境を構築します。

1. マニュアル「サービスプラットフォーム ファーストステップガイド」の「3.5.2 HCSCTE プロジェクトの作成」および「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」に従って，HCSCTE プロジェクトを作成します。
2. システム構成定義をテスト環境からエクスポートして開発環境にインポートします。

(4) 各種定義ファイルの設定

テスト環境で HTTP 受付および HTTP アダプタが動作するために，各種定義ファイルを設定します。

(a) HTTP 受付の定義ファイル

HTTP 受付で使用する定義ファイルの種類と，この設定例での使用有無を次に示します。

定義ファイルの種類	概要	この設定例での使用有無
HTTP レスポンスヘッダ定義ファイル	HTTP 受付の応答電文フォーマット（ヘッダ変数用）を省略した場合のデフォルト値として使用する HTTP レスポンスヘッダ情報を設定します。	使用しません。
HTTP 受付定義ファイル	HTTP 受付の動作情報を設定します。	使用します。

この設定例では，作業フォルダを利用してファイルを受信するため，次に示す HTTP 受付定義ファイルのファイル設定用プロパティを設定します。

プロパティ名	設定値	説明
httprecp.switchover.file-transfer.mode	true	作業フォルダを生成するため，ここでは「true」を設定します。
httprecp.response.ignore-bodymsg	true	HTTP 受付のレスポンス内容に，応答電文（ボディ）ではなく作業フォルダ内のファイルを使用するため，ここでは「true」を設定します。

表に記載していないプロパティは設定不要です。

各プロパティの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.11.3　HTTP 受付定義ファイル」を参照してください。

HTTP 受付定義ファイルは「(6) ビジネスプロセスの追加」で編集します。

(b) HTTP アダプタの定義ファイル

HTTP アダプタで使用する定義ファイルの種類と，この設定例での使用有無を次に示します。

定義ファイルの種類	概要	この設定例での使用有無
HTTP アダプタ定義ファイル	HTTP アダプタの環境非依存の動作情報を設定します。	使用しません（すべてデフォルト値を使用します）。
HTTP アダプタ実行環境プロパティファイル	HTTP アダプタの環境依存の動作情報を設定します。	使用します。
HTTP アダプタ実行環境共通プロパティファイル	HTTP アダプタの環境依存の動作情報を設定します。複数のアダプタの定義を一括で変更する場合に使用します。	使用しません。

HTTP アダプタでは，これらの定義ファイル以外にシステムプロパティの設定が必要な場合がありますが，この設定例では使用しません。

HTTP アダプタ実行環境プロパティファイルで設定するプロパティを次の表に示します。

プロパティ名	設定値	概要
adphttp.request.method	GET	ファイルダウンロード要求のリクエストを GET メソッドで行うため，「GET」を設定します。
adphttp.request.uri-scheme-authority	環境に応じて設定します。	URI のスキームとオーソリティを指定します。事前に準備している接続先を指定します。この設定例では「http://hostA」とします。
adphttp.request.uri-path	環境に応じて設定します。	URI のパスを指定します。この設定例では test.bmp ファイルをダウンロードするパスとして「/test.bmp」を設定します。
adphttp.request.output-folder-name	－（設定しません）	この設定例ではダウンロードしたファイルを作業フォルダに出力するため，設定しません。

表に記載していないプロパティは設定不要です。

各プロパティの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.14.2　HTTP アダプタ実行環境プロパティファイル」を参照してください。

設定手順を次に示します。

1. テンプレートファイル（<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¥custom-adapter¥HTTP¥config¥templates¥serviceid.properties）をコピーして，次のフォルダに格納します。

   <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>¥CSC¥custom-adapter¥HTTP¥config

2. コピーしたテンプレートファイルをテキストエディタなどで開いて次のように編集します。

```
adphttp.request.method=GET
adphttp.request.uri-scheme-authority=http://hostA
adphttp.request.uri-path=/test.bmp
#adphttp.request.uri-query.<INDEX>=<QUERY-NAME>=<QUERY-VALUE>
#adphttp.request.header.authorization.type=none
#adphttp.request.header.authorization=
#adphttp.request.header.content-type.charset=
#adphttp.request.header.content-type=
#adphttp.request.header.userdef.<INDEX>=<FIELD-NAME>:<FIELD-VALUE>
#adphttp.request.part.message.binding=none
#adphttp.request.part.file.1.input-folder-name=
#adphttp.request.part.file.1.local-file-name=
#adphttp.request.output-folder-name=
#adphttp.config.trace.path=
#adphttp.config.messagelog.level=10
#adphttp.config.methodtrace.level=3
#adphttp.config.methodtrace.filenum=8
#adphttp.config.methodtrace.filesize=2097152
```

```
#adphttp.config.exptrace.filenum=8
#adphttp.config.exptrace.filesize=2097152
```

3. コピーしたファイルの名称を「adphttp.properties」に変更します。
   「adphttp」は HTTP アダプタのサービス ID です。このサービス ID は，開発環境で HTTP アダプタを定義するときに使用します。

(c) HCSC サーバランタイム定義ファイルの設定

この設定例では，デフォルトの作業フォルダルートを使用するため，HCSC サーバランタイム定義ファイルの設定は不要です。

作業フォルダの詳細は，マニュアル「サービスプラットフォーム 解説」の「2.15.6 HTTP 受付のファイル管理」を参照してください。

(5) HTTP アダプタの定義

開発環境で HTTP アダプタを定義します。操作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. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして「サービスアダプタ追加」を選択します。
   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。

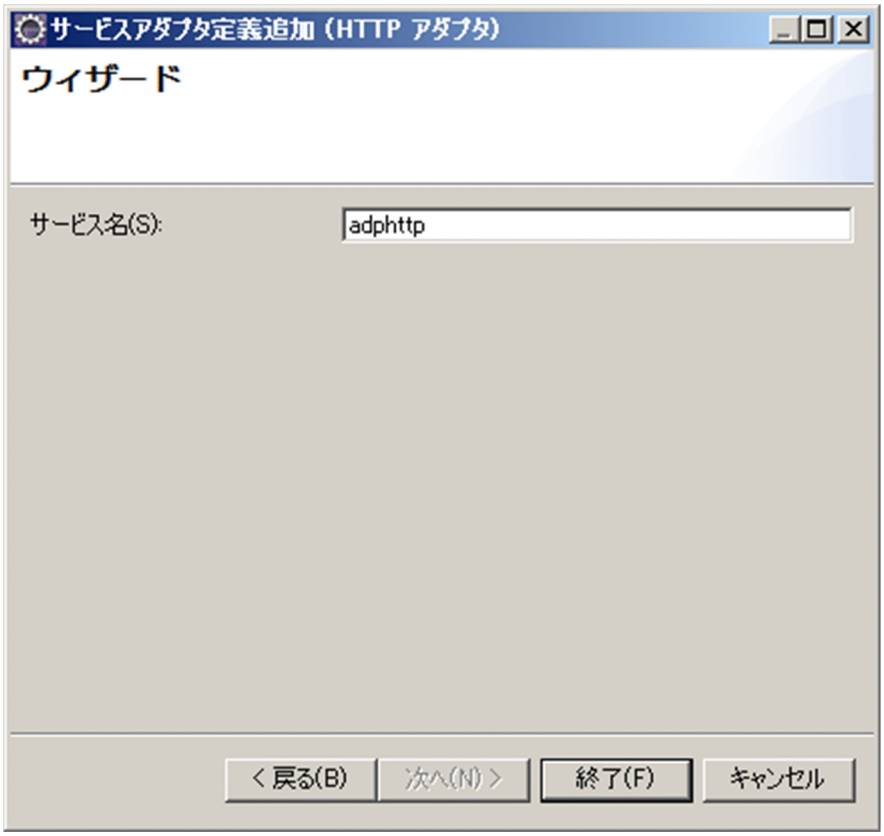


4. サービス部品の種別のドロップダウンリストから「HTTP アダプタ」を選択して［次へ］ボタンをクリックします。

   HTTP 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。

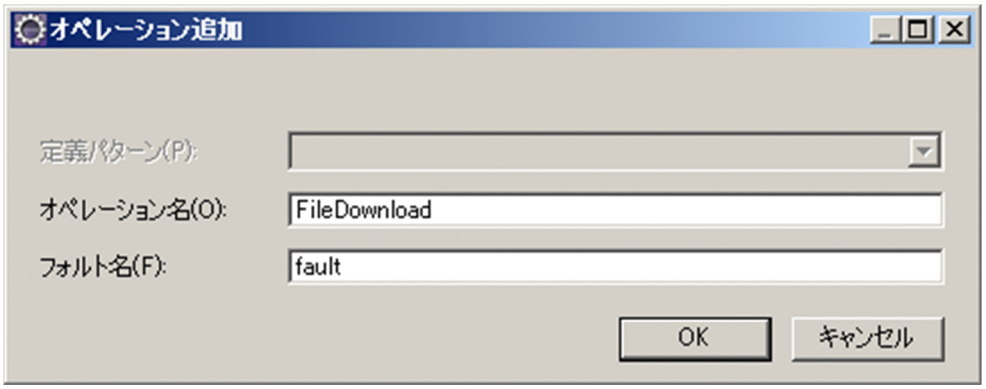
5. サービス名を入力します。

   ここでは「adphttp」とします。



6. [終了] ボタンをクリックします。
   サービスアダプタ定義画面（サービスアダプタ定義（基本）タブ）が表示されます。

7. オペレーションの［追加］ボタンをクリックして，オペレーション名とフォルト名を入力します。
   ここでは，オペレーション名は「FileDownload」，フォルト名は「fault」とします。



8. [オペレーション追加] ダイアログで［OK］ボタンをクリックします。

9. [要求電文] － ［サービス部品］ － ［電文フォーマット］の［参照］ボタンをクリックして，次の電文フォーマット定義ファイルを指定します。
   <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¥custom-adapter¥HTTP¥schema¥adphttp_body_empty.fdx

注

この設定例では HTTP アダプタでファイルの受信だけを行うため，要求および応答のボディ電文はサービスプラットフォームが提供する空の電文フォーマット定義ファイルを利用します。

10. [応答電文] − [サービス部品] − [電文フォーマット] の [参照] ボタンをクリックして，次の電文フォーマット定義ファイルを指定します。

<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¥custom-adapter¥HTTP¥schema¥adphttp_body_empty.fdx

11. [サービス部品定義情報] の [サービス ID] を「adphttp」に変更します。

設定後のサービスアダプタ定義（基本）タブの例を次に示します。



オペレーション情報の通信モデルは「同期」から変更しないでください。なお，この設定例ではサービスアダプタ定義（詳細）タブは設定しません。すべてデフォルトのまま使用します。

12. [ファイル] − [保存] を選択します。

HTTP アダプタの定義が保存されます。

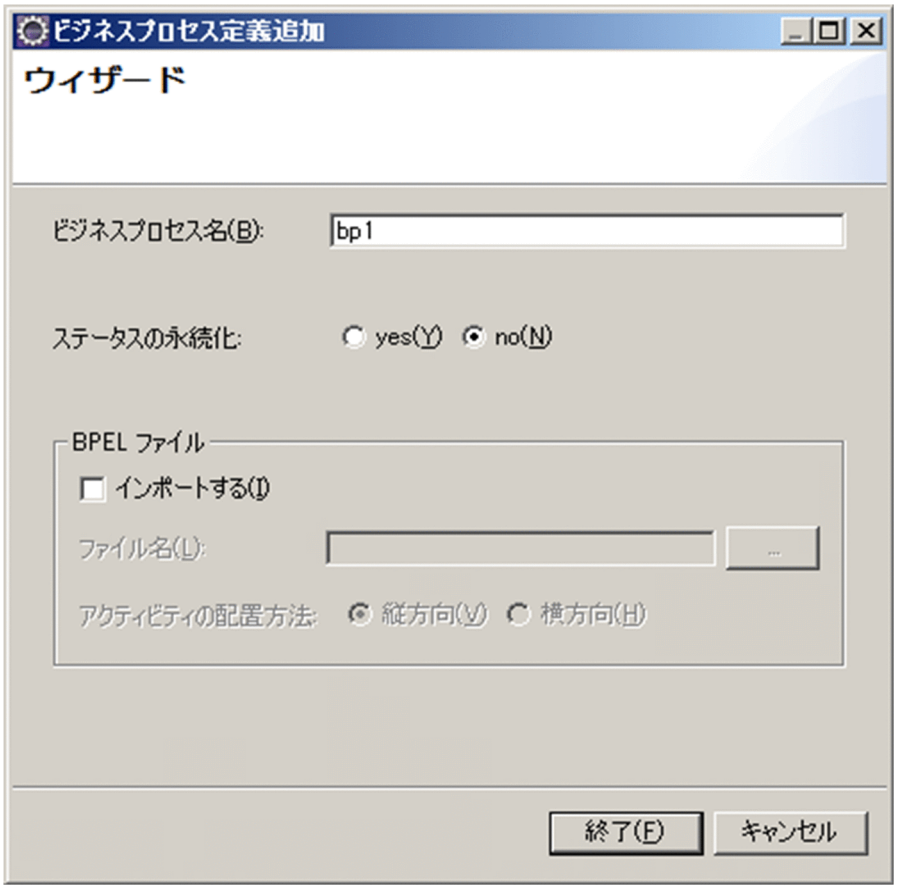
(6)　ビジネスプロセスの追加

開発環境でビジネスプロセスを追加します。操作手順を次に示します。

1. ツリービューのサービス定義一覧で右クリックして「ビジネスプロセス追加」を選択します。
   ビジネスプロセス定義を追加するためのダイアログが表示されます。

2. 次のように設定します。

項目	設定内容
ビジネスプロセス名	「bp1」を入力します。
ステータスの永続化	「no」を選択します。
BPEL ファイル	「インポートする」にチェックしません。



3. [終了] ボタンをクリックします。
   ビジネスプロセスが追加され，ビジネスプロセス定義画面が表示されます。

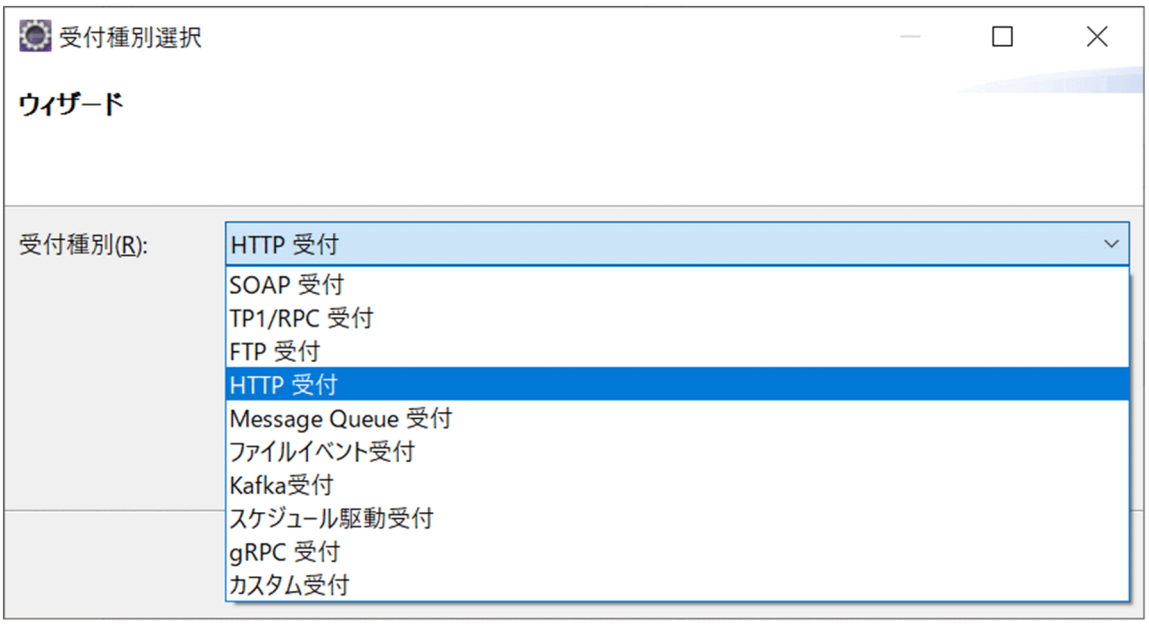
(7)　HTTP 受付の追加

呼び出しサービス選択機能を使用しないで，リクエストを受け付けるための HTTP 受付を追加します。操作手順を次に示します。

1. ツリービューのビジネスプロセス（bp1）を右クリックして［ユーザ定義受付（呼出先固定）追加］を選択します。

   受付種別選択ウィザードが表示されます。

2. 受付種別のドロップダウンリストから「HTTP 受付」を選択して［次へ］ボタンをクリックします。



   HTTP 受付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。

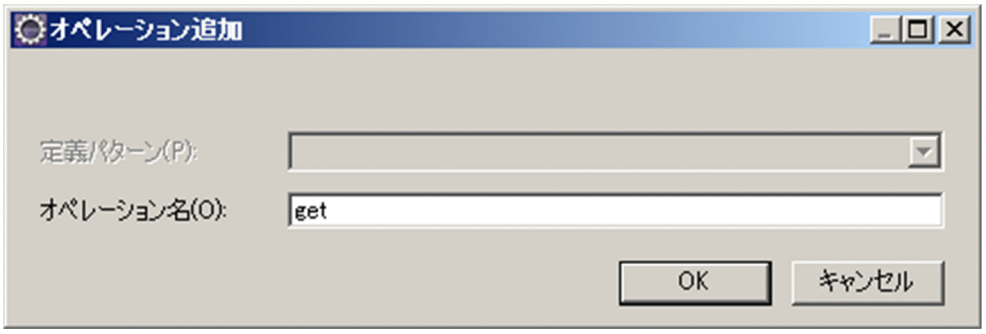
3. ［受付名］を指定します。

   ここでは「recphttp」とします。



4. [終了] ボタンをクリックします。
   ユーザ定義受付定義画面（ユーザ定義受付（基本）タブ）が表示されます。

5. オペレーションの［追加］ボタンをクリックして，オペレーション名を入力します。
   ここでは，オペレーション名は「get」とします。

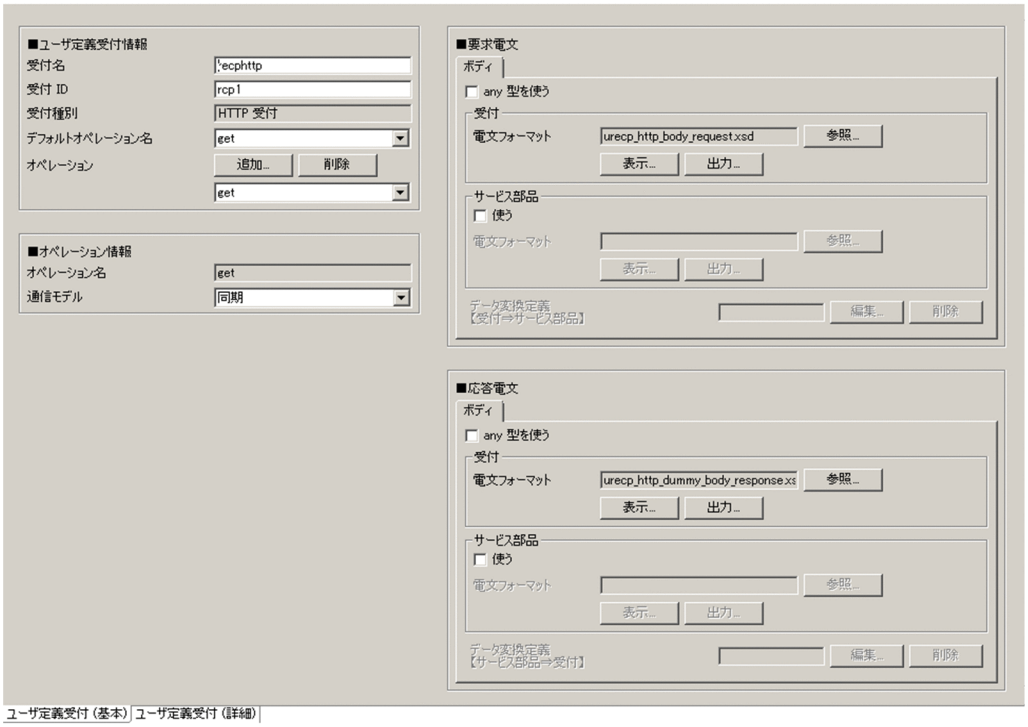


6. [オペレーション追加] ダイアログで［OK］ボタンをクリックします。

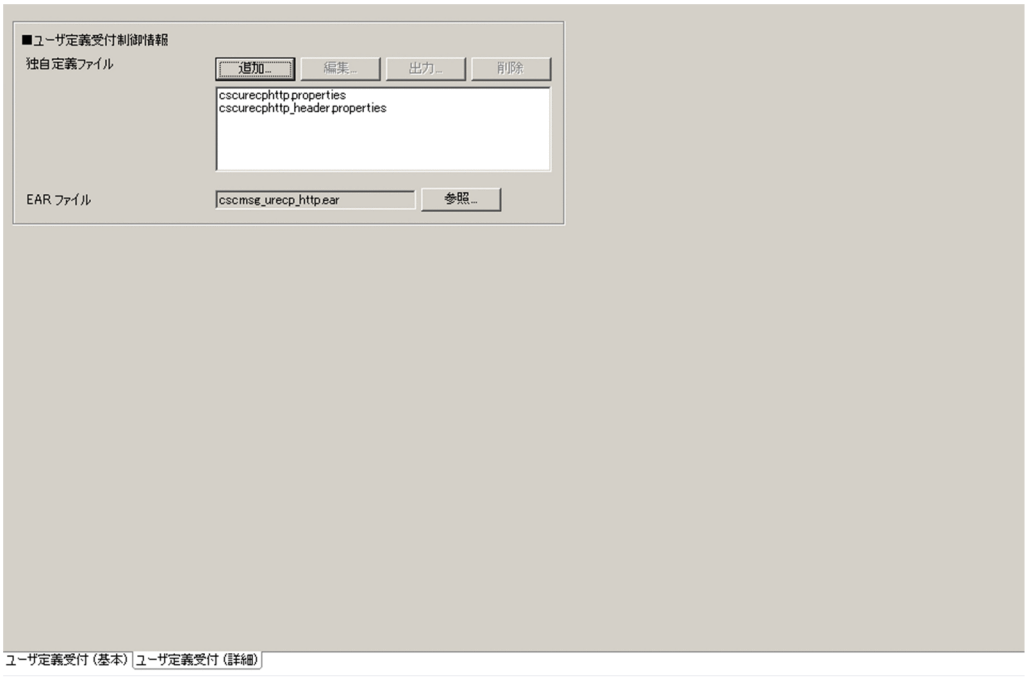
7. [要求電文] － ［受付］ － ［電文フォーマット］の［参照］ボタンをクリックして，次の電文フォーマット定義ファイルを指定します。
   <サービスプラットフォームのインストールディレクトリ>¥CSC¥custom-reception¥http¥schema¥urecp_http_body_request.xsd

8. [応答電文] － ［受付］ － ［電文フォーマット］の［参照］ボタンをクリックして，次の電文フォーマット定義ファイルを指定します。

<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¥custom-reception¥http¥schema¥urecp_http_dummy_body_response.xsd
オペレーション情報の通信モデルは「同期」から変更しないでください。
設定後のユーザ定義受付（基本）タブの例を次に示します。



9. [ユーザ定義受付（詳細）] タブをクリックします。
[ユーザ定義受付（詳細）] タブが表示されます。



10. [独自定義ファイル] で「cscurecphttp.properties」を選択し，[編集] ボタンをクリックします。

独自定義ファイルを編集するためのエディタが起動します。

11. HTTP 受付定義ファイルの内容を次のように修正します。

```
#urecp-http.context-root=
#urecp-http.max-threads=10
#urecp-http.exclusive-threads=0
#urecp-http.queue-size=8192
#urecp-http.pooled-instance.minimum=0
#urecp-http.pooled-instance.maximum=0
#urecp-http.ejb-transaction-timeout=0
#httprecp.switchover.pass-through.mode=false
httprecp.switchover.file-transfer.mode=true
#httprecp.http.charset=UTF-8
httprecp.response.ignore-bodymsg=true
#httprecp.response.header.filename=
#httprecp.system-exception.status-code=500
#httprecp.response.generate.content-length=true
#httprecp.file-trans.temp-file.partsize-threshold=0
#httprecp.file-trans.maxsize.part=2147483647
#httprecp.file-trans.maxsize.request=2147483647
#httprecp.work-dir.auto-delete=true
#httprecp.response.download.disposition-type=attachment
                        :
```

12. [ファイル] － [すべて保管] を選択します。

HTTP 受付の定義および HTTP 受付定義ファイルが保存されます。

(8) ビジネスプロセスの定義

ビジネスプロセスの処理フローを定義します。

(a) 変数の定義

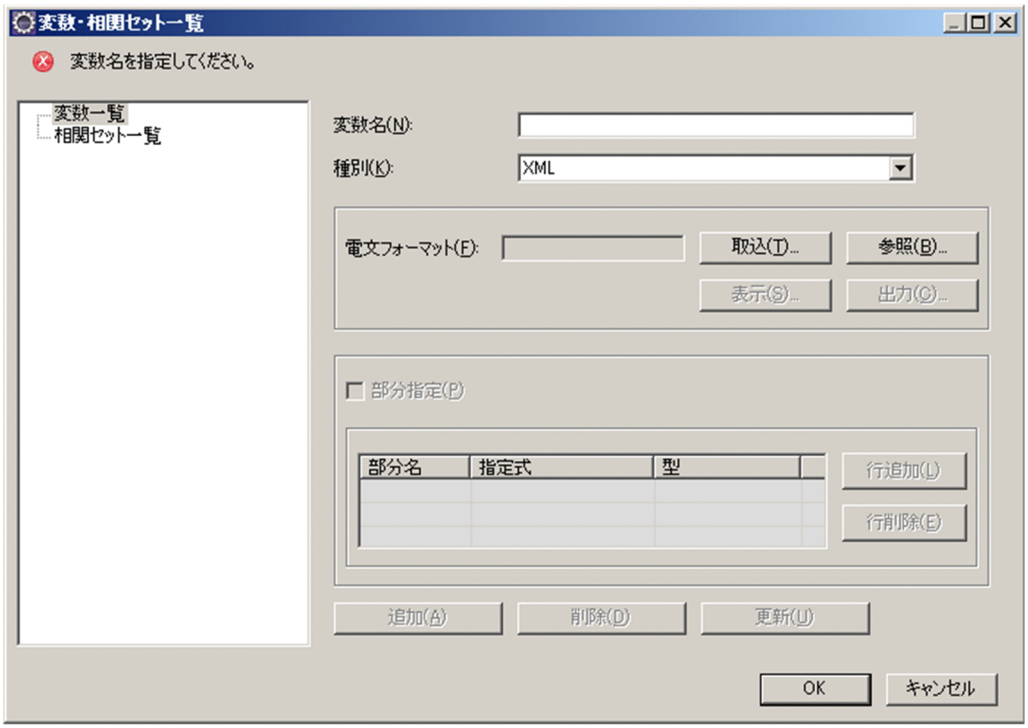
ビジネスプロセスで利用する変数を 1 つずつ定義します。この設定例で定義する変数の一覧を次に示します。

表 G-1 定義する変数一覧

#	変数名	種別	用途の説明
1	recphttp_head_request	XML	HTTP 受付の要求電文（ヘッダ）用の変数
2	recphttp_head_response	XML	HTTP 受付の応答電文（ヘッダ）用の変数
3	adphttp_head_request	XML	HTTP アダプタの要求電文（ヘッダ）用の変数
4	adphttp_head_response	XML	HTTP アダプタの応答電文（ヘッダ）用の変数
5	recphttp_body_request	XML	HTTP 受付の要求電文（ボディ）用の変数
6	recphttp_body_response	XML	HTTP 受付の応答電文（ボディ）用の変数
7	adphttp_body_request	non-XML	HTTP アダプタの要求電文（ボディ）用の変数
8	adphttp_body_response	non-XML	HTTP アダプタの応答電文（ボディ）用の変数

変数の定義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. 変数に応じて次の操作をします。

- #1～#4 の場合

  表 G-1 の#1～#4 の変数名および種別を指定して［参照］ボタンをクリックします。

  電文フォーマットに使用するファイルを指定するダイアログが表示されるため，次の表に示す参照フォーマット定義を指定します。

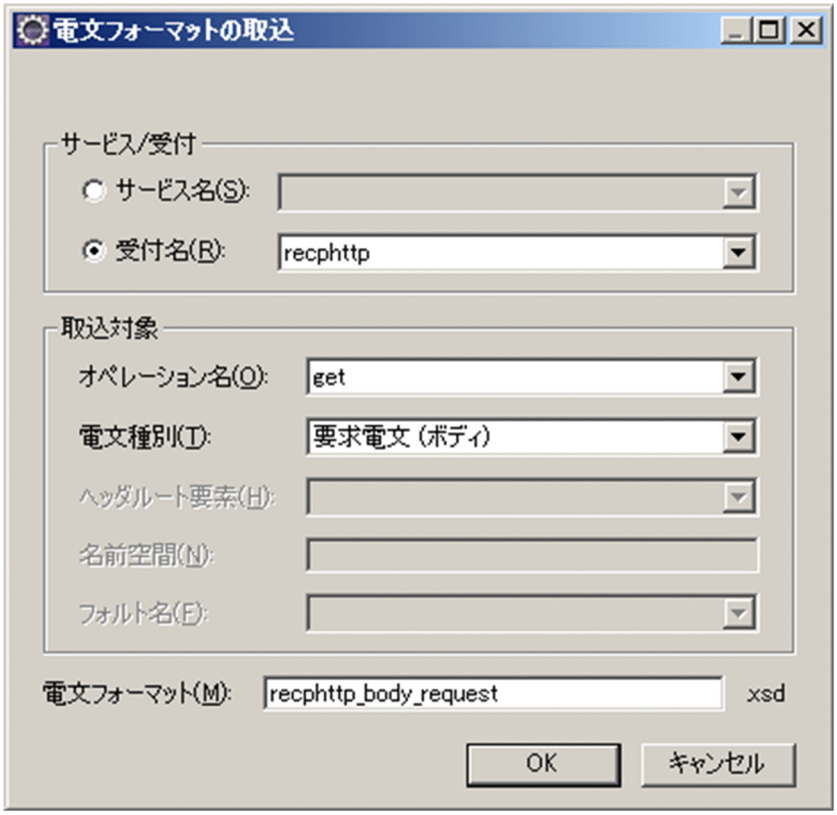
#	変数名	参照フォーマット定義
1	recphttp_head_request	<サービスプラットフォームのインストールディレクトリ>¥CSC¥custom-reception¥http¥schema¥urecp_http_header_request.xsd
2	recphttp_head_response	<サービスプラットフォームのインストールディレクトリ>¥CSC¥custom-reception¥http¥schema¥urecp_http_header_response.xsd
3	adphttp_head_request	<サービスプラットフォームのインストールディレクトリ>¥CSC¥custom-adapter¥HTTP¥schema¥adphttp_header_request1.xsd
4	adphttp_head_response	<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¥custom-adapter¥HTTP¥schema¥adphttp_header_response1.xsd

- #5～#8 の場合

  ［変数・相関セット一覧］ダイアログで，表 G-1 の#5～#8 の変数名および種別を指定して［取込］ボタンをクリックします。

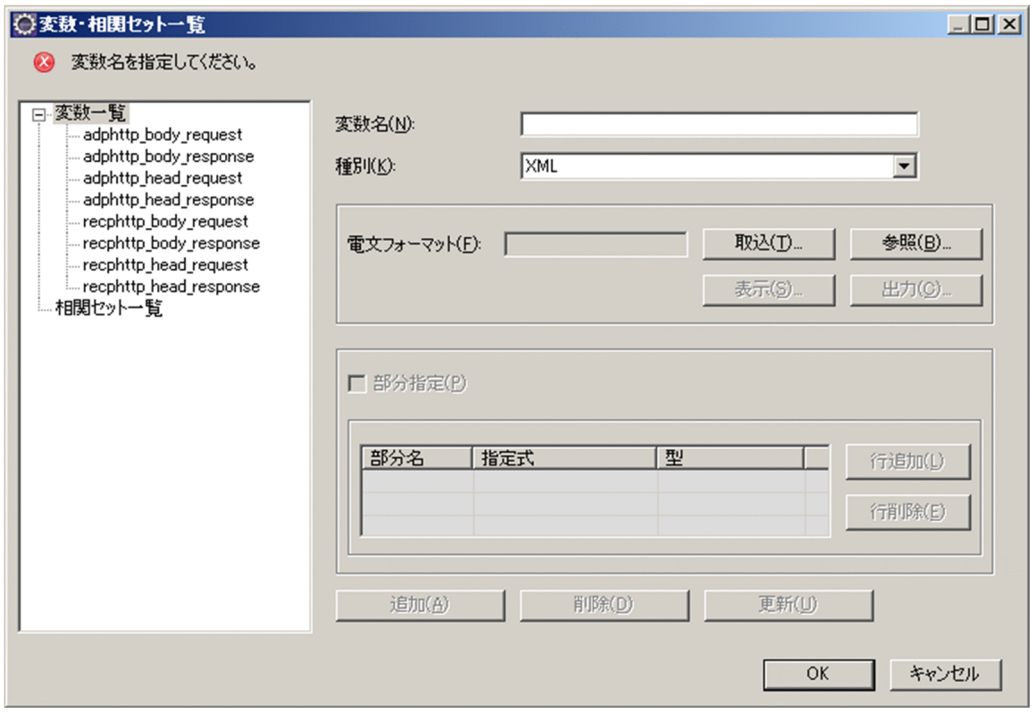
［電文フォーマットの取込］ダイアログが表示されるため，次のように指定して［OK］ボタンをクリックします。

#	変数名	サービス/受付		オペレーション名	電文種別	電文フォーマット
5	recphttp_body_request	受付名	recphttp	get	要求電文（ボディ）	recphttp_body_request.xsd
6	recphttp_body_response	受付名	recphttp	get	応答電文（ボディ）	recphttp_body_response.xsd
7	adphttp_body_request	サービス名	adphttp	FileDownload	要求電文（ボディ）	adphttp_body_request.fdx
8	adphttp_body_response	サービス名	adphttp	FileDownload	応答電文（ボディ）	adphttp_body_response.fdx



3. [変数・相関セット一覧］ダイアログで［追加］ボタンをクリックします。

変数一覧に，追加した変数が表示されます。#1～#8 の変数を同様に定義します。



4. [OK] ボタンをクリックします。

［変数・相関セット一覧］ダイアログが閉じます。

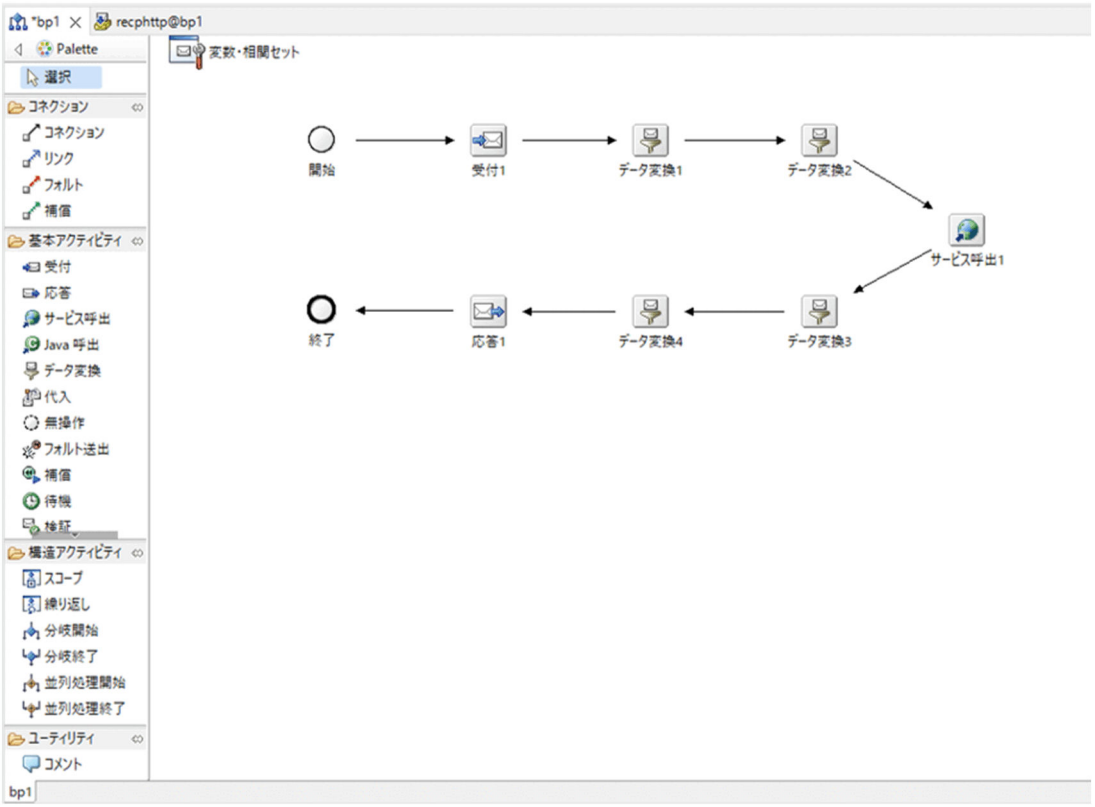
(b) アクティビティの配置

ビジネスプロセス定義画面で次の表に示す順番にアクティビティを配置して，コネクションで接続します。

順番	アクティビティ種別	初期アクティビティ名	説明
1	開始	開始	ビジネスプロセスの開始を表します。デフォルトで配置されています。
2	受付	受付 1	HTTP クライアントからの要求電文を受け付けるためのインターフェースを定義します。
3	データ変換	データ変換 1	データ変換処理を定義します。
4	データ変換	データ変換 2	データ変換処理を定義します。
5	サービス呼出	サービス呼出 1	定義済みの HCSC コンポーネントへの要求電文の送信を定義します。この設定例では HTTP アダプタが対象となります。
6	データ変換	データ変換 3	データ変換処理を定義します。
7	データ変換	データ変換 4	データ変換処理を定義します。
8	応答	応答 1	HTTP クライアントから受信した要求電文に対する応答を返すためのインターフェースを定義します。
9	終了	終了	ビジネスプロセスの終了を表します。デフォルトで配置されています。

アクティビティを配置する手順については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.4.1　アクティビティを配置する」を参照してください。

配置後のビジネスプロセス定義画面の状態を次に示します。



注　図のとおりに整列させる必要はありません。

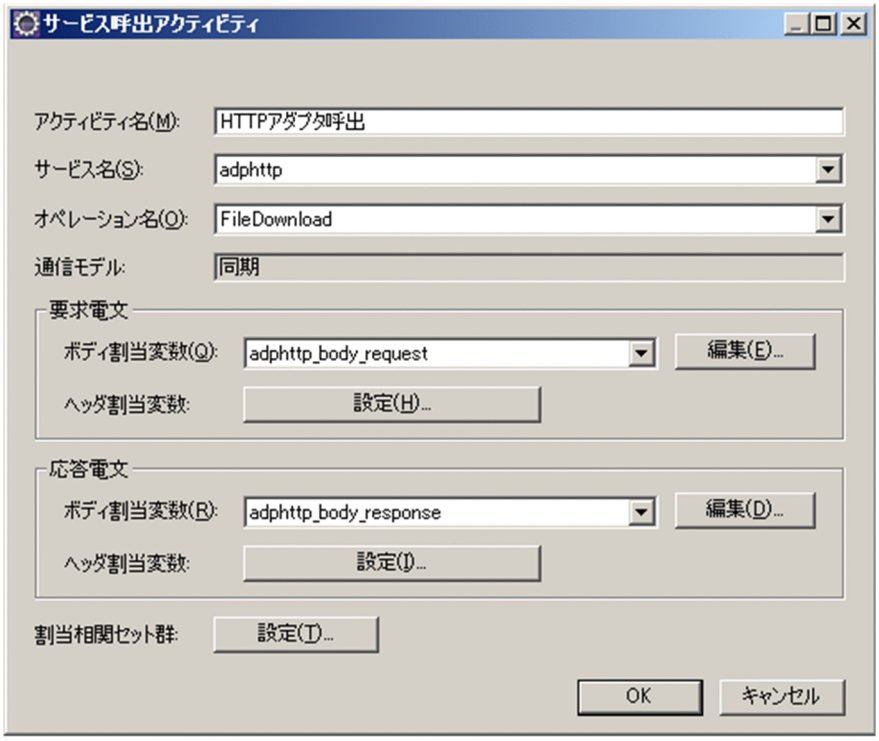
(c)　サービス呼出アクティビティの定義

HTTP アダプタを呼び出すためのサービス呼出アクティビティを定義します。

1. 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. 次のように設定します。

項目		設定内容
アクティビティ名		HTTP アダプタ呼出
サービス名		adphttp
オペレーション名		FileDownload
通信モデル		同期（編集できません）
要求電文	ボディ割当変数	adphttp_body_request
応答電文	ボディ割当変数	adphttp_body_response

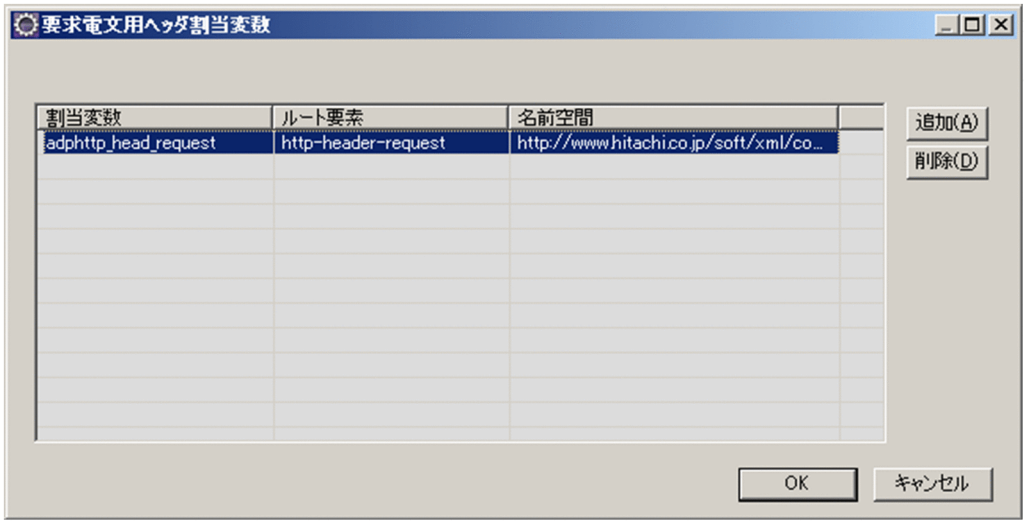


3. [要求電文] − [ヘッダ割当変数] の [設定] ボタンをクリックします。
   [要求電文用ヘッダ割当変数] ダイアログが表示されます。

4. [追加] ボタンをクリックします。

5. [要求電文用ヘッダ割当変数] ダイアログで次のように設定します。

項目	設定内容
割当変数	adphttp_head_request
ルート要素	http-header-request
名前空間	http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/header_request



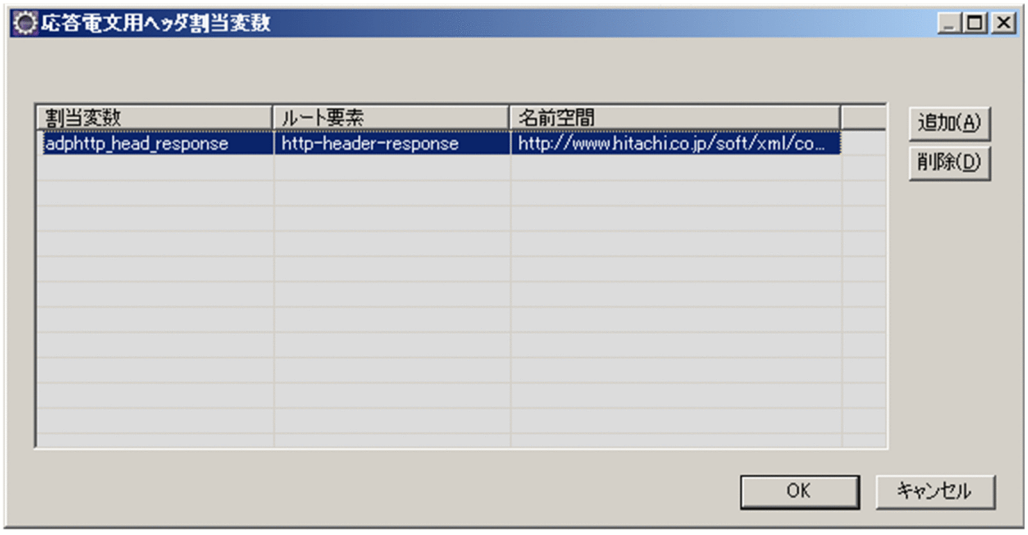
6. [要求電文用ヘッダ割当変数] ダイアログで［OK］ボタンをクリックします。
［要求電文用ヘッダ割当変数］ダイアログが閉じます。

7. [応答電文] －［ヘッダ割当変数］の［設定］ボタンをクリックします。
［応答電文用ヘッダ割当変数］ダイアログが表示されます。

8. [追加] ボタンをクリックします。

9. [応答電文用ヘッダ割当変数] ダイアログで次のように値を設定します。

項目	設定内容
割当変数	adphttp_head_response
ルート要素	http-header-response
名前空間	http://www.hitachi.co.jp/soft/xml/cosminexus/csc/adapter/http/header_response



10. [応答電文用ヘッダ割当変数] ダイアログで［OK］ボタンをクリックします。
［応答電文用ヘッダ割当変数］ダイアログが閉じます。

11. [サービス呼出アクティビティ] ダイアログで［OK］ボタンをクリックします。
設定内容が保存されて［サービス呼出アクティビティ］ダイアログが閉じます。

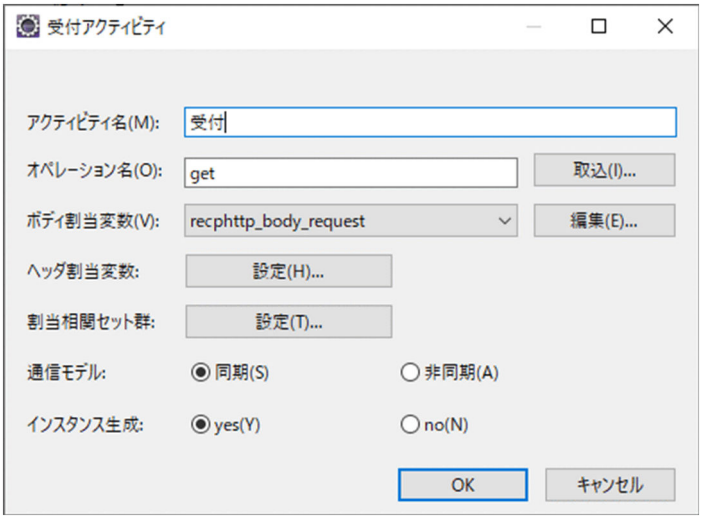
(d)　受付アクティビティの定義

受付アクティビティを定義します。

1. 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. 次のように設定します。

項目	設定内容
アクティビティ名	受付
オペレーション名	get
ボディ割当変数	recphttp_body_request
通信モデル	同期
インスタンス生成	yes

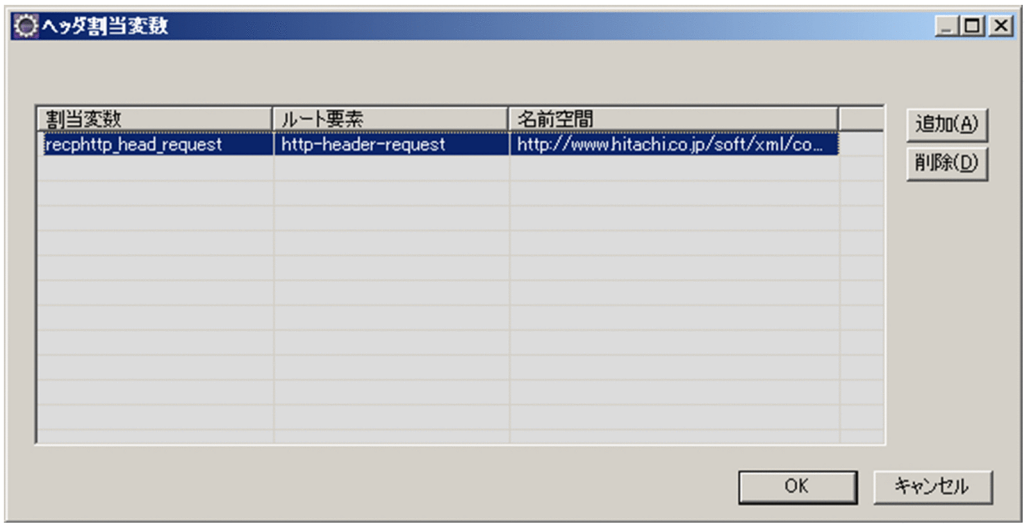


3. [ヘッダ割当変数] の [設定] ボタンをクリックします。
   [ヘッダ割当変数] ダイアログが表示されます。

4. [追加] ボタンをクリックします。

5. [ヘッダ割当変数] ダイアログで次のように設定します。

項目	設定内容
割当変数	recphttp_head_request
ルート要素	http-header-request
名前空間	http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/request



6. [ヘッダ割当変数] ダイアログで［OK］ボタンをクリックします。
   ［ヘッダ割当変数］ダイアログが閉じます。

7. [受付アクティビティ] ダイアログで［OK］ボタンをクリックします。
   設定内容が保存されて［受付アクティビティ］ダイアログが閉じます。

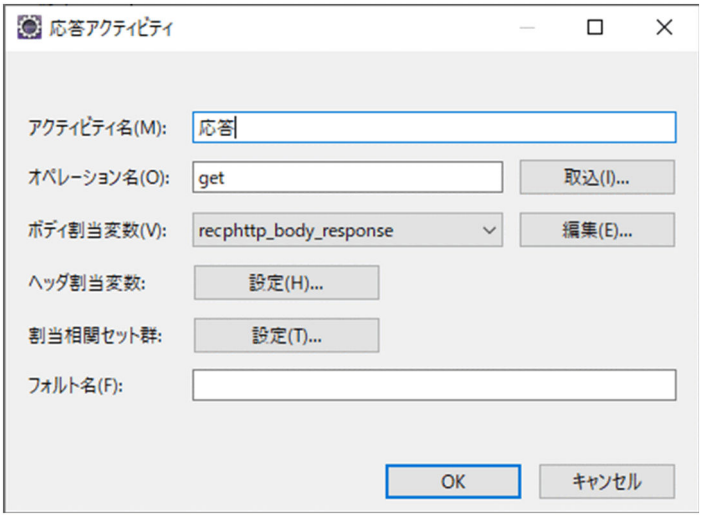
(e)　応答アクティビティの定義

応答アクティビティを定義します。

1. 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. 次のように設定します。

項目	設定内容
アクティビティ名	応答
オペレーション名	get
ボディ割当変数	recphttp_body_response
フォルト名	－（設定しません）

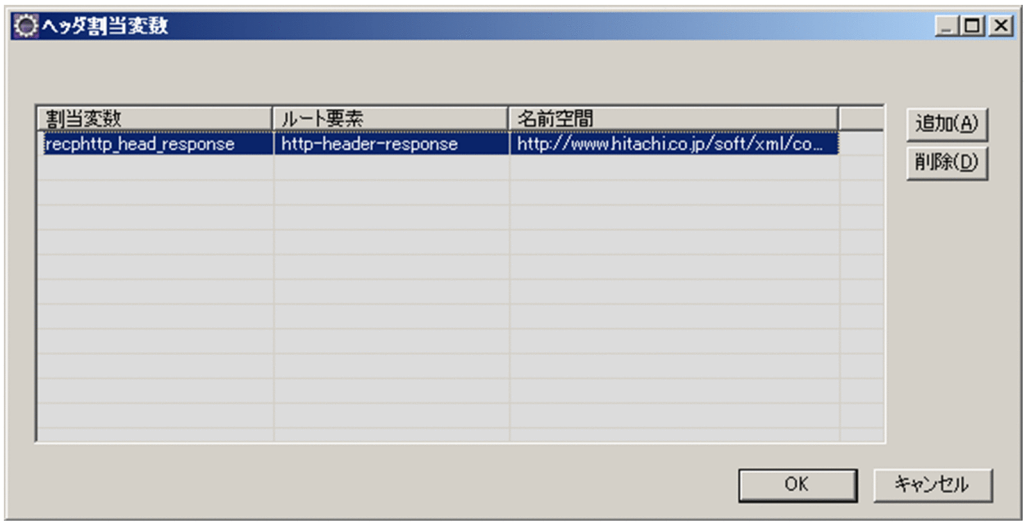


3. [ヘッダ割当変数] の［設定］ボタンをクリックします。
   ［ヘッダ割当変数］ダイアログが表示されます。

4. [追加] ボタンをクリックします。

5. [ヘッダ割当変数］ダイアログで次のように設定します。

項目	設定内容
割当変数	recphttp_head_response
ルート要素	http-header-response
名前空間	http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/response



6. [ヘッダ割当変数］ダイアログで［OK］ボタンをクリックします。
   ［ヘッダ割当変数］ダイアログが閉じます。

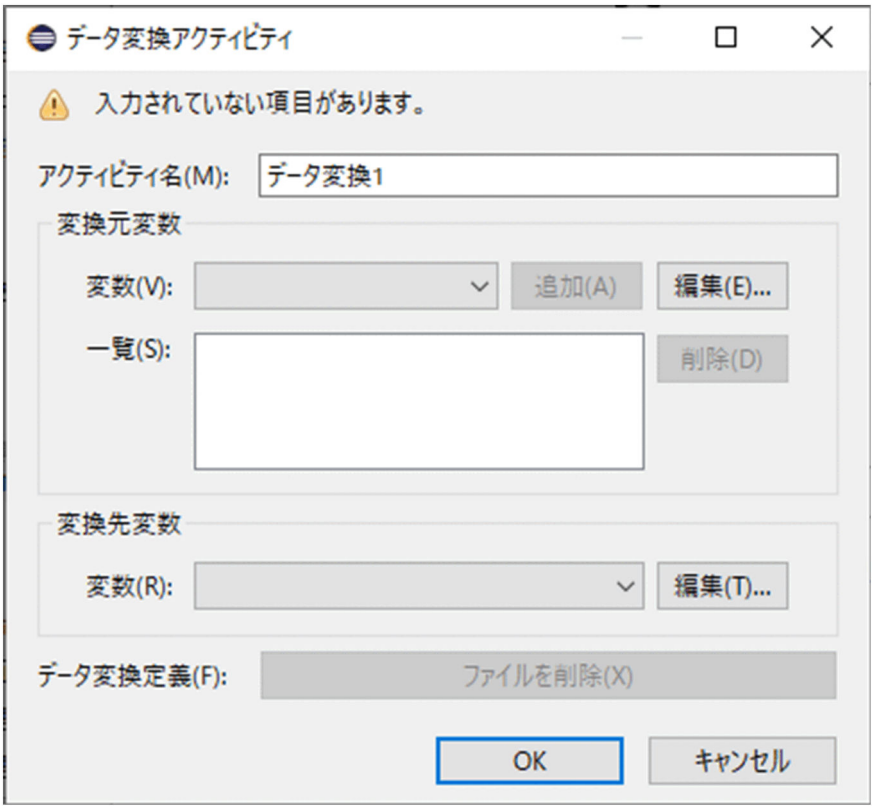
7. [応答アクティビティ］ダイアログで［OK］ボタンをクリックします。
   設定内容が保存されて［応答アクティビティ］ダイアログが閉じます。

(f)　データ変換アクティビティの定義

各データ変換アクティビティを定義します。

1. データ変換アクティビティ（データ変換 1～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. それぞれのデータ変換アクティビティで次のように設定します。

アクティビティ	項目		設定内容
データ変換 1	アクティビティ名		HTTP アダプタボディ要求電文生成
	変換元変数	一覧※	recphttp_body_request
	変換先変数	変数	adphttp_body_request
データ変換 2	アクティビティ名		HTTP アダプタヘッダ要求電文生成
	変換元変数	一覧※	recphttp_head_request
	変換先変数	変数	adphttp_head_request
データ変換 3	アクティビティ名		HTTP 受付ボディ応答電文生成
	変換元変数	一覧※	adphttp_body_response
	変換先変数	変数	recphttp_body_response
データ変換 4	アクティビティ名		HTTP 受付ヘッダ応答電文生成
	変換元変数	一覧※	adphttp_head_response

アクティビティ	項目		設定内容
データ変換 4	変換先変数	変数	recphttp_head_response

注※

［変換元変数］の［変数］ドロップダウンリストから該当する変数を選択して追加ボタンを押すと一覧に追加されます。

3. ［データ変換アクティビティ］ダイアログで［OK］ボタンをクリックします。

設定内容が保存されて［データ変換アクティビティ］ダイアログが閉じます。データ変換 1～データ変換 4 まで同様に定義します。

(9)　データ変換の定義

「(8)(f) データ変換アクティビティの定義」で定義したデータ変換アクティビティに対して，マッピング定義を生成します。

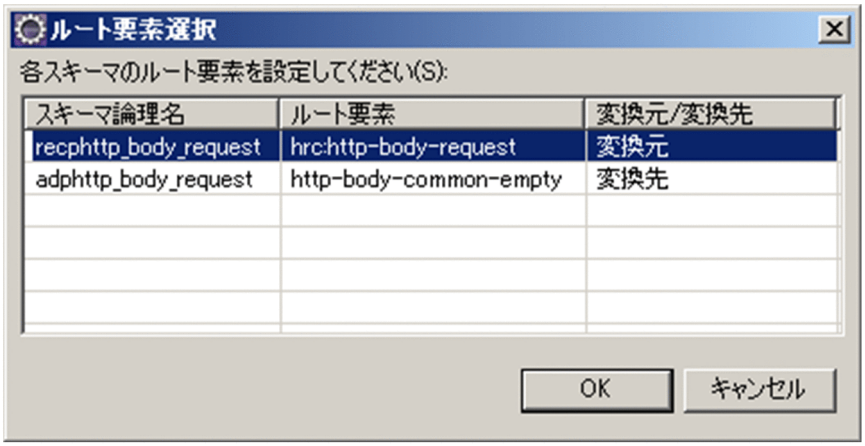
マッピング定義の操作方法についてはマニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「7.4　マッピングする」を参照してください。

各データ変換アクティビティのマッピング定義について示します。

(a)　HTTP アダプタボディ要求電文生成

1. ビジネスプロセス定義画面でデータ変換アクティビティ（HTTP アダプタボディ要求電文生成）をダブルクリックします。

［ルート要素選択］ダイアログが表示されます。



2. ［OK］ボタンをクリックします。

マッピングビューアが表示されます。



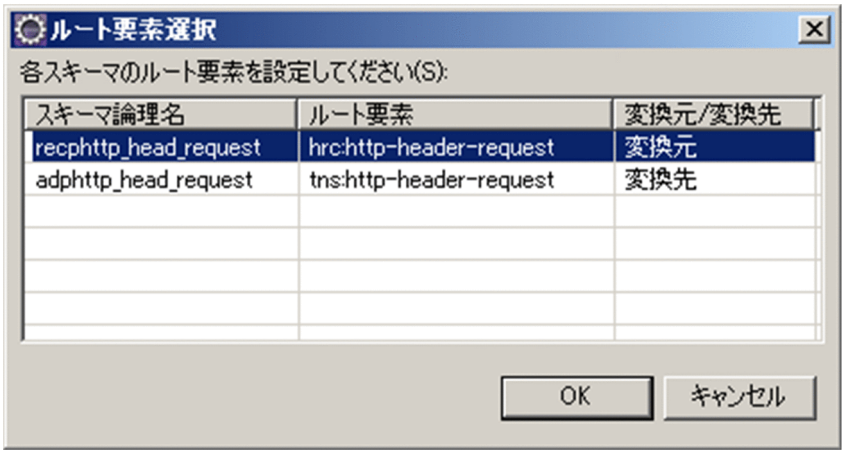
このアクティビティではマッピングは不要です。この設定例では，ファイルを送信する処理を実行するため，HTTP アダプタの要求電文（ボディ）は空の状態にします。

(b)　HTTP アダプタヘッダ要求電文生成

HTTP アダプタがダウンロードするファイルを作業フォルダに格納するようにマッピングします。リクエストのメソッドおよび URI は HTTP アダプタ実行環境プロパティファイルに定義しているため，要求電文の該当個所へのマッピングは不要です。

1. ビジネスプロセス定義画面でデータ変換アクティビティ（HTTP アダプタヘッダ要求電文生成）をダブルクリックします。

   ［ルート要素選択］ダイアログが表示されます。



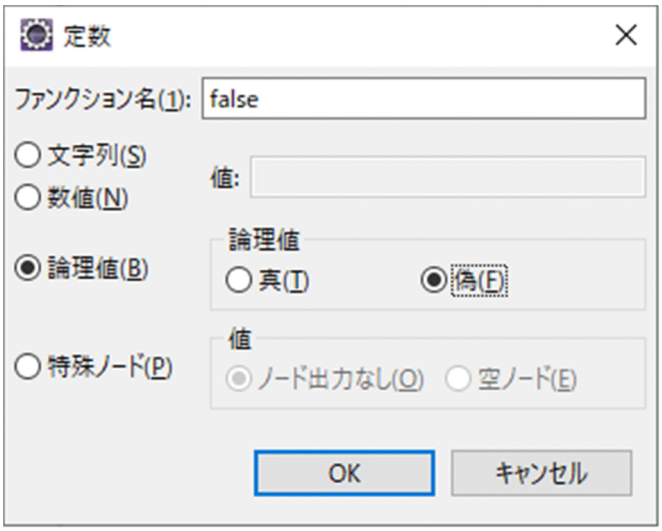
2. ［OK］ボタンをクリックします。

   マッピングビューアが表示されます。

3. const ファンクションを配置して，const ファンクションをダブルクリックします。

   ［定数］ダイアログが表示されます。

4. ファンクション名を「false」に変更して，「論理値」および「偽」を選択します。



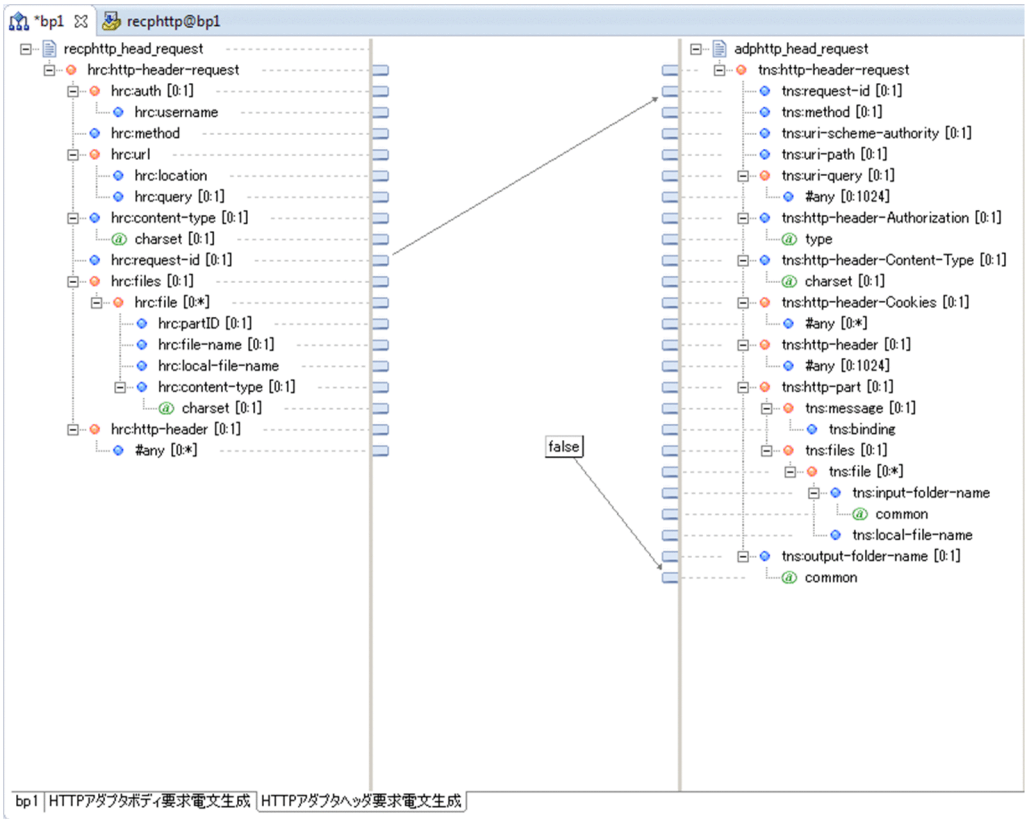
5. [OK] ボタンをクリックします。

［定数］ダイアログが閉じます。

6. 次の表を参考にマッピングします。

マッピング元およびマッピング先はルート要素からのパスで記載しています。

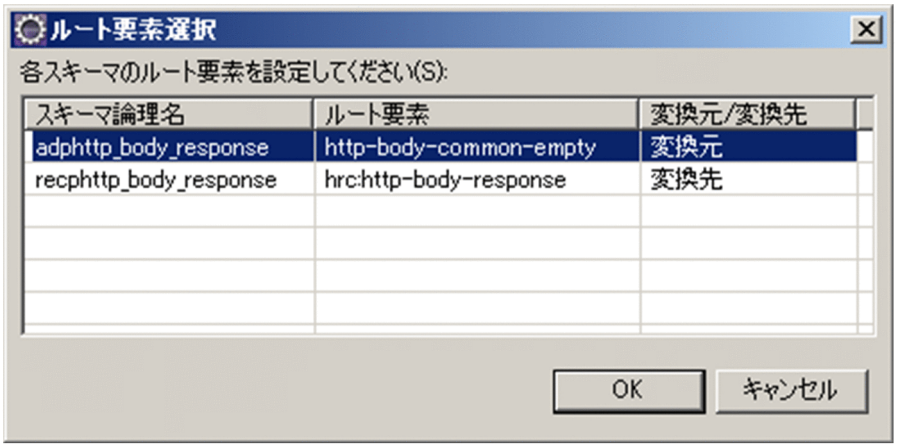
マッピング元	マッピング先
/hrc:http-header-request/hrc:request-id	/tns:http-header-request/tns:request-id
const ファンクション（false）	/tns:http-header-request/tns:output-folder-name/@common



(c) HTTP 受付ボディ応答電文生成

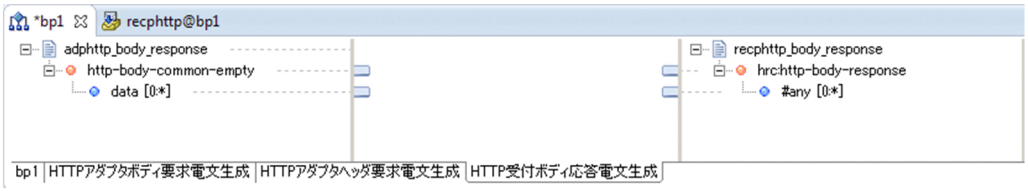
1. ビジネスプロセス定義画面でデータ変換アクティビティ（HTTP 受付ボディ応答電文生成）をダブルクリックします。

［ルート要素選択］ダイアログが表示されます。



2. [OK] ボタンをクリックします。

   マッピングビューアが表示されます。



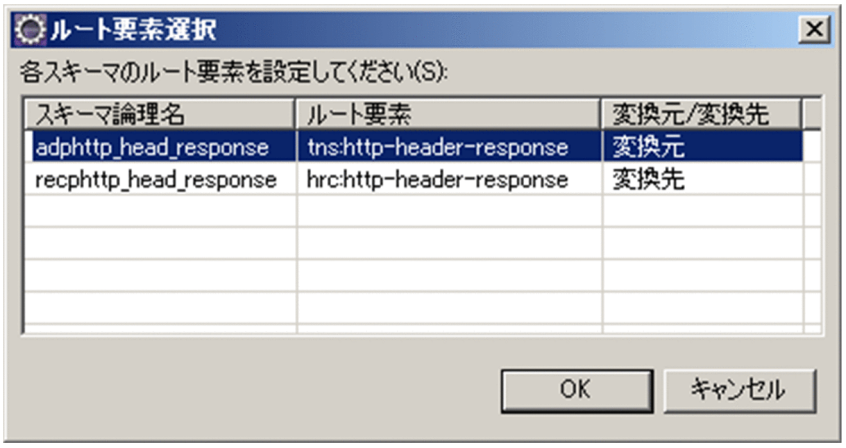
   このアクティビティではマッピングは不要です。この設定例では，HTTP 受付はレスポンス時にファイルをダウンロードする処理を実行するため，HTTP 受付の応答電文（ボディ）は空の状態にします。

(d) HTTP 受付ヘッダ応答電文生成

HTTP 受付がクライアントに返すファイル情報をマッピングします。

1. ビジネスプロセス定義画面でデータ変換アクティビティ（HTTP 受付ヘッダ応答電文生成）をダブルクリックします。

   [ルート要素選択] ダイアログが表示されます。



2. [OK] ボタンをクリックします。

   マッピングビューアが表示されます。

3. const ファンクションを配置して，const ファンクションをダブルクリックします。
［定数］ダイアログが表示されます。

4. ファンクション名を「file-name」に変更します。

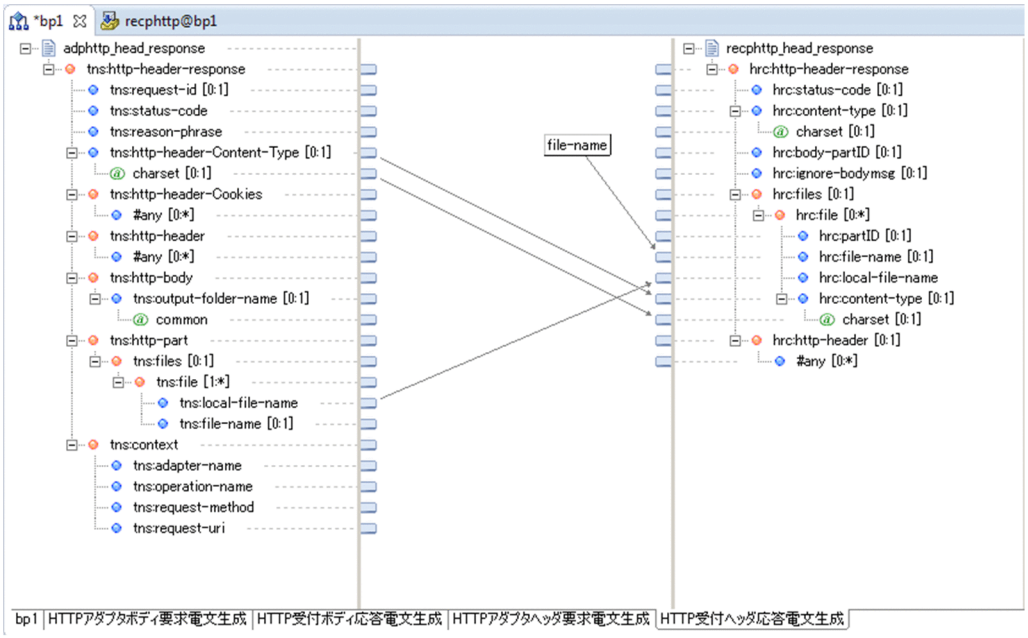
5.「文字列」を選択し，値にダウンロードするファイル名となる任意の文字列を指定します。
この設定例では「test.bmp」を指定します。



6. [OK] ボタンをクリックします。
［定数］ダイアログが閉じます。

7. 次の表を参考にマッピングします。
マッピング元およびマッピング先はルート要素からのパスで記載しています。

マッピング元	マッピング先
const ファンクション（file-name）	/hrc:http-header-response/hrc:files/hrc:file/hrc:file-name
/tns:http-header-response/tns:http-part/tns:files/tns:file/tns:local-file-name	/hrc:http-header-response/hrc:files/hrc:file/hrc:local-file-name
/tns:http-header-response/tns:http-header-Content-Type	/hrc:http-header-response/hrc:files/hrc:file/hrc:content-type
/tns:http-header-response/tns:http-header-Content-Type/@charset	/hrc:http-header-response/hrc:files/hrc:file/hrc:content-type/@charset

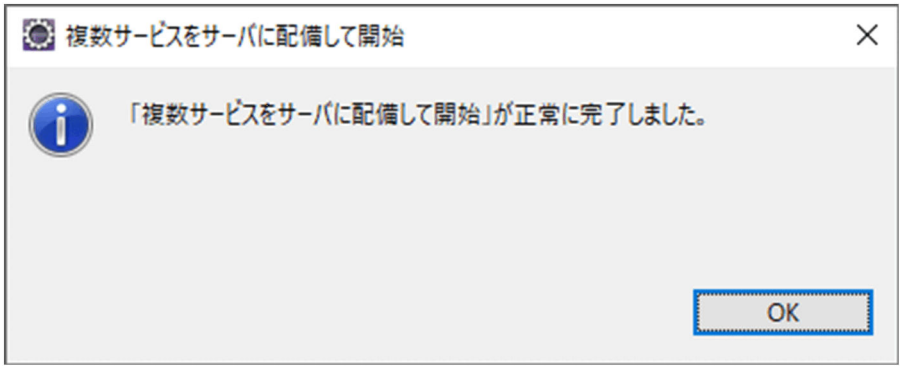


(10)　ビジネスプロセスを実行するための準備

定義した HCSC コンポーネントを保存して配備定義します。操作手順を次に示します。

1. [ファイル] − [すべて保管] を選択します。
   HTTP アダプタ，HTTP 受付，およびビジネスプロセスの定義が保存されます。
2. ツリービューの［サービス定義一覧］を右クリックして，［複数サービスをサーバに配備して開始］を選択します。
3. サービス選択ページで配備および開始するサービスを選択します。
4. [Finish] ボタンをクリックします。
5. [OK] ボタンをクリックします。

次のダイアログが表示されていれば処理が正常に完了しています。



失敗した場合は，エラー内容を基にここまで定義した内容に誤りがないか確認してください。

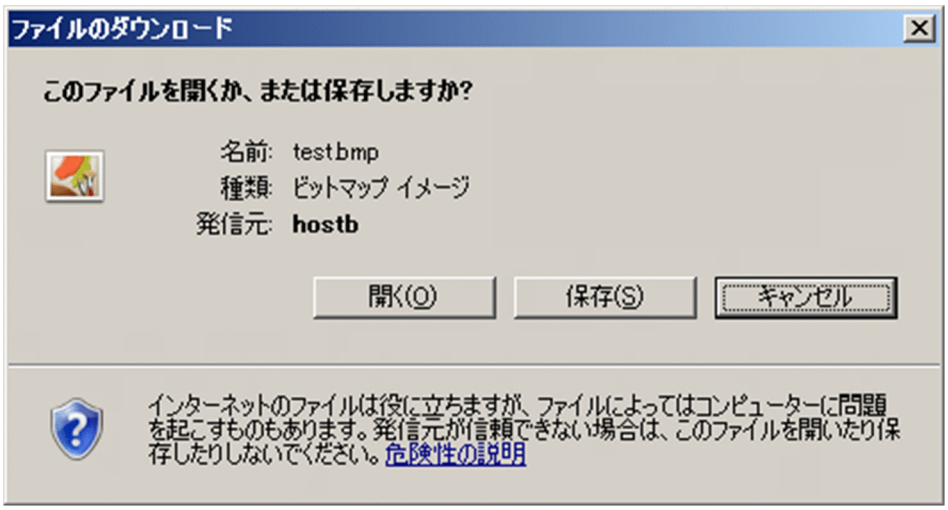
(11)　ビジネスプロセスの実行

1. Web ブラウザを起動して次の URL にアクセスします。

   http://hostB/rcp1/get

   URL 中の「hostB」は，hostB の IPv4 形式の IP アドレスに名前解決されるものとします。ホスト名以下のパスは「/受付 ID/受付のオペレーション名」です。

ビジネスプロセスが正しく実行されている場合は，次のポップアップが出力されて HTTP 受付からファイルをダウンロードできます。



ダウンロードするファイル名は，マッピング定義の const ファンクションで指定した名称となります。

付録 G.2　設定例 2（HTTP アダプタの接続先をリクエストごとに動的に変更する）

(1)　設定例の概要

この設定例では，「付録 G.1　設定例 1（HTTP サーバから HTTP クライアントにファイルをダウンロードする）」で作成した定義を変更して，HTTP アダプタの接続先パスをリクエストに応じて動的に変更する処理を行うビジネスプロセスを定義します。

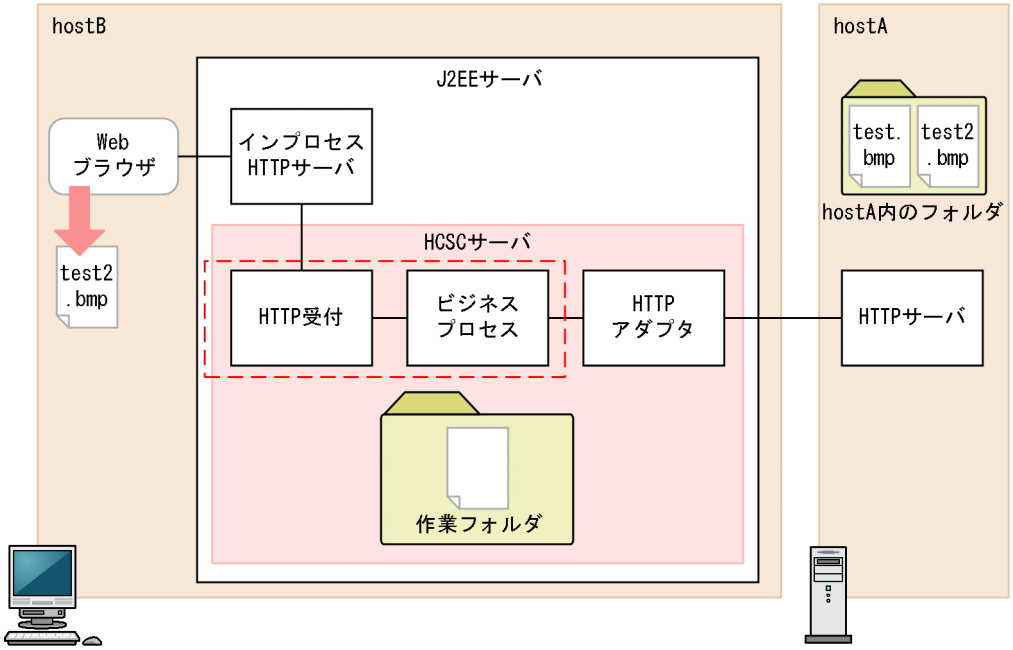
動的に変更するための入力データには，HTTP 受付に渡すクエリを利用します。

以降の説明は「付録 G.1　設定例 1（HTTP サーバから HTTP クライアントにファイルをダウンロードする）」を一とおり実施していることが前提となります。

(a)　環境構成

この設定例で作成する環境の構成を次の図に示します。

図 G-2　設定例 1 で作成する環境の構成



この設定例では，図中の点線で囲まれた部分の定義の変更手順について説明します。

設定例 1 と同様に，HTTP アダプタの接続先となる HTTP サーバについては事前に準備していることを前提とします。また，設定例 2 では，http://hostA:80/test2.bmp にアクセスすると test2.bmp をダウンロードできるように HTTP サーバを構成しているものとします。

(b)　設定の流れ

設定の流れを次に示します。

1. 電文フォーマットの作成
2. 変数定義の変更
3. データ変換定義の変更
4. ビジネスプロセスを実行するための準備
5. ビジネスプロセスの実行

それぞれの手順について次に説明します。

(2)　電文フォーマットの作成

HTTP 受付の要求電文（ボディ）として使用する電文フォーマットを作成します。作成した電文フォーマットに HTTP クライアントから渡されるクエリ情報が格納されます。

電文フォーマットはテンプレートを使用して作成します。作成手順を次に示します。

1. 電文フォーマットのテンプレートが格納されている次のフォルダを任意の場所にコピーします。

<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¥custom-reception¥http¥schema

2. コピーしたフォルダ内の「urecp_http_body_detail_request.xsd」をテキストエディタで開きます。

3. 次のようにファイル名を表す要素（斜体の個所）を追加します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema elementFormDefault="qualified"
 targetNamespace="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/request
"
 xmlns:hrc="http://www.hitachi.co.jp/soft/xml/cosminexus/csc/reception/http/request"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:complexType name="http-body-requestType">
  <xsd:sequence>
   <!-- User Customize -->
   <xsd:element name="filename" type="xsd:string"/>
  </xsd:sequence>
 </xsd:complexType>
</xsd:schema>
```

4. ファイルを上書き保存します。

注　urecp_http_body_request.xsd は編集しないでください。

(3)　変数定義の変更

(a)　HTTP 受付

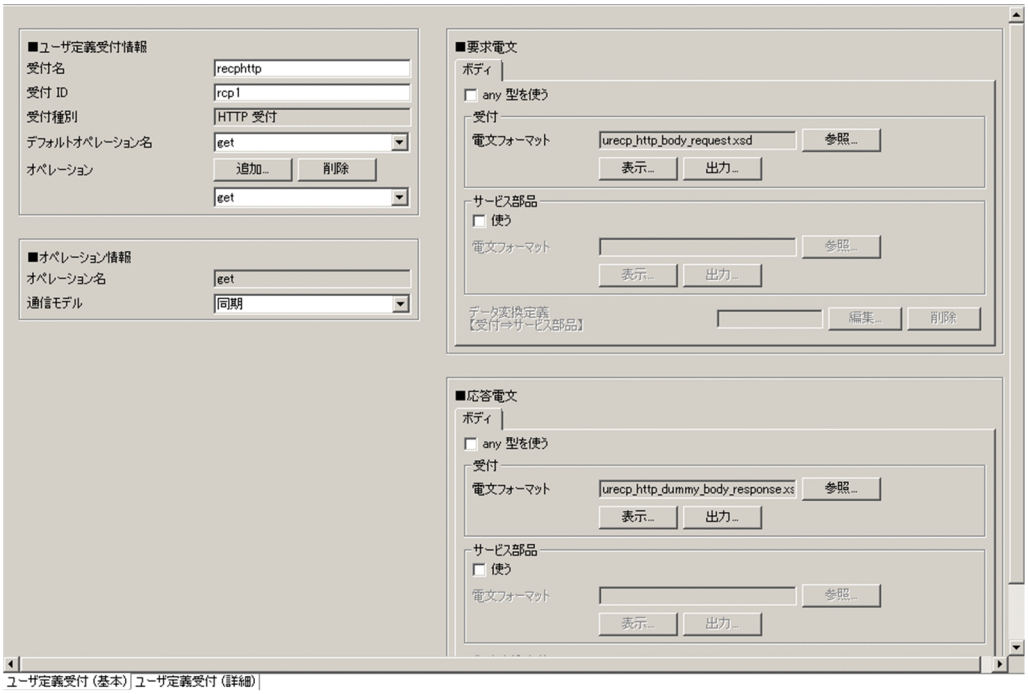
(2)で作成した電文フォーマットを HTTP 受付の定義に反映します。

1. サービス定義一覧のツリーで HTTP 受付（recphttp）をダブルクリックします。

HTTP 受付のユーザ定義受付定義画面が開きます。

2. [要求電文] − [受付] − [電文フォーマット] の [参照] ボタンをクリックして，次の電文フォーマット定義ファイルを指定します。

<(2)のコピー先フォルダ>¥schema¥urecp_http_body_request.xsd



3. ［ファイル］ − ［保存］を選択して HTTP 受付を保存します。
   保存時に次の確認ダイアログが表示された場合は［OK］ボタンをクリックします。



(b)　ビジネスプロセス

HTTP 受付の要求電文（ボディ）の割当変数である recphttp_body_request を，(2)で作成した電文フォーマットに更新します。

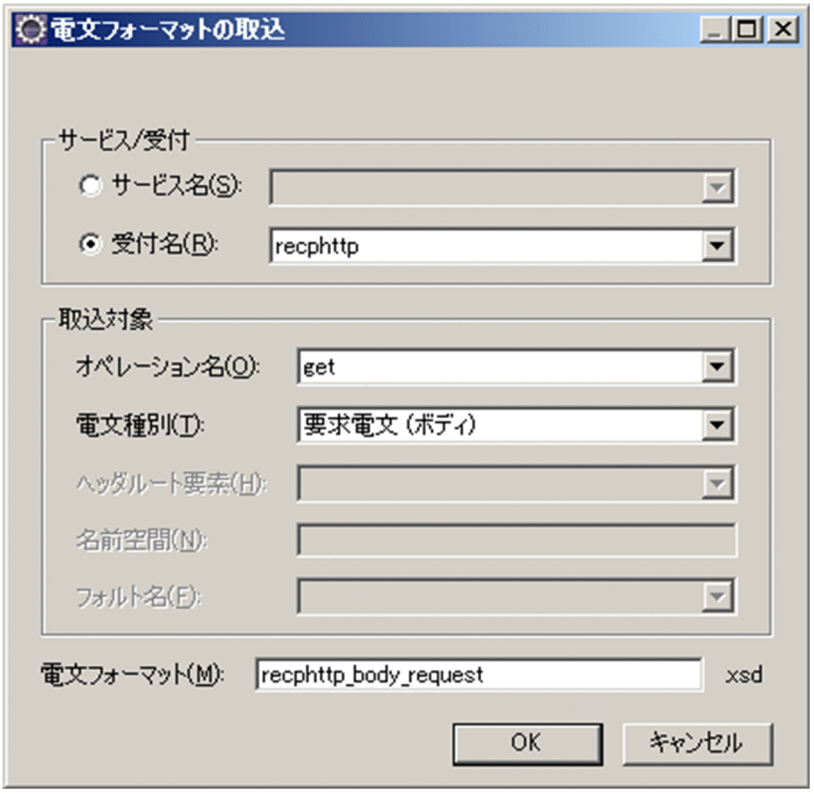
1. ツリービューのサービス定義一覧でビジネスプロセス（bp1）をダブルクリックします。
   ビジネスプロセス定義画面が表示されます。

2. ビジネスプロセス定義画面のキャンバス上の［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。

3. 変数一覧のツリーから recphttp_body_request を選択し，［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. 次のように指定して［OK］ボタンをクリックします。

変数名	サービス/受付		オペレーション名	電文種別	電文フォーマット
recphttp_body_request	受付名	recphttp	get	要求電文（ボディ）	recphttp_body_request.xsd



5. [変数・相関セット一覧] ダイアログで［更新］ボタンをクリックします。
   変数が更新されます。

6. [変数・相関セット一覧] ダイアログで［表示］ボタンをクリックします。
   ［変数の表示］ダイアログが表示されます。



変数が更新されていることを確認してダイアログを閉じます。

7. [変数・相関セット一覧] ダイアログで [OK] ボタンをクリックします。
   [変数・相関セット一覧] ダイアログが閉じます。

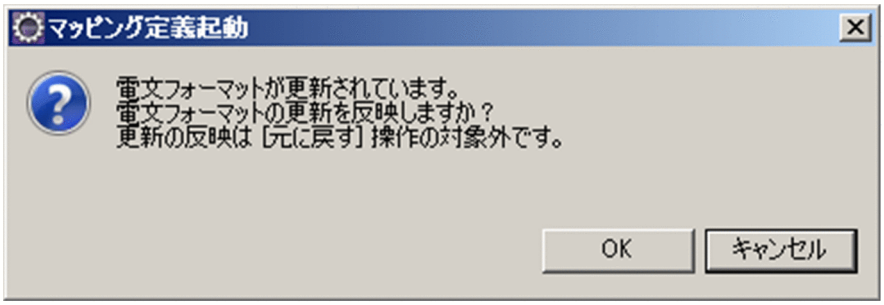
(4) データ変換定義の変更

変更した HTTP 受付の要求電文（ボディ）の変数を利用して，マッピング定義を変更します。

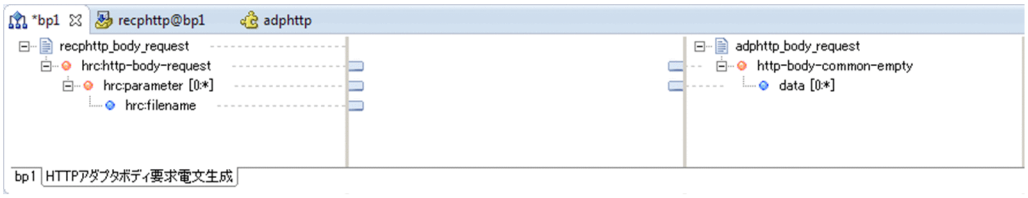
(a) HTTP アダプタボディ要求電文へのマッピング

1. ビジネスプロセス定義画面でデータ変換アクティビティ（HTTP アダプタボディ要求電文生成）をダブルクリックします。

2. マッピング定義起動を確認するダイアログが表示されます。



3. [OK] ボタンをクリックします。
   マッピング定義画面が表示されます。

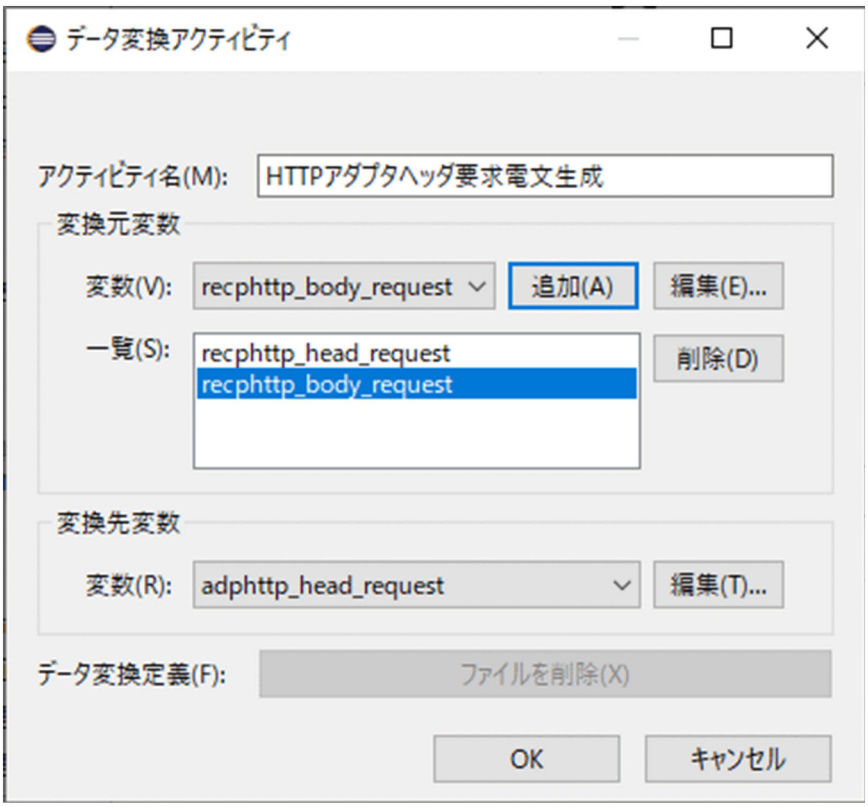


マッピング元に filename 要素が追加されています。設定例 1 と同様にマッピングは不要です。

(b) HTTP アダプタヘッダ要求電文へのマッピング

1. ビジネスプロセスの定義画面でデータ変換アクティビティ（HTTP アダプタヘッダ要求電文生成）を右クリックし，［設定］を選択します。
   ［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. 変換元変数の［変数］ドロップダウンリストから「recphttp_body_request」を選択して［追加］ボタンをクリックします。
   一覧に recphttp_body_request が追加されます。



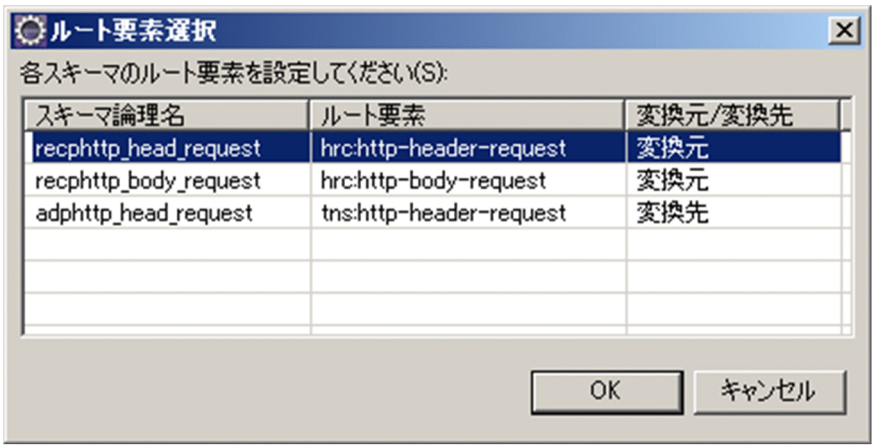
3. [OK] ボタンをクリックします。
   ［データ変換アクティビティ］ダイアログが閉じます。

4. データ変換アクティビティ（HTTP アダプタヘッダ要求電文生成）をダブルクリックします。

マッピング定義起動を確認する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。

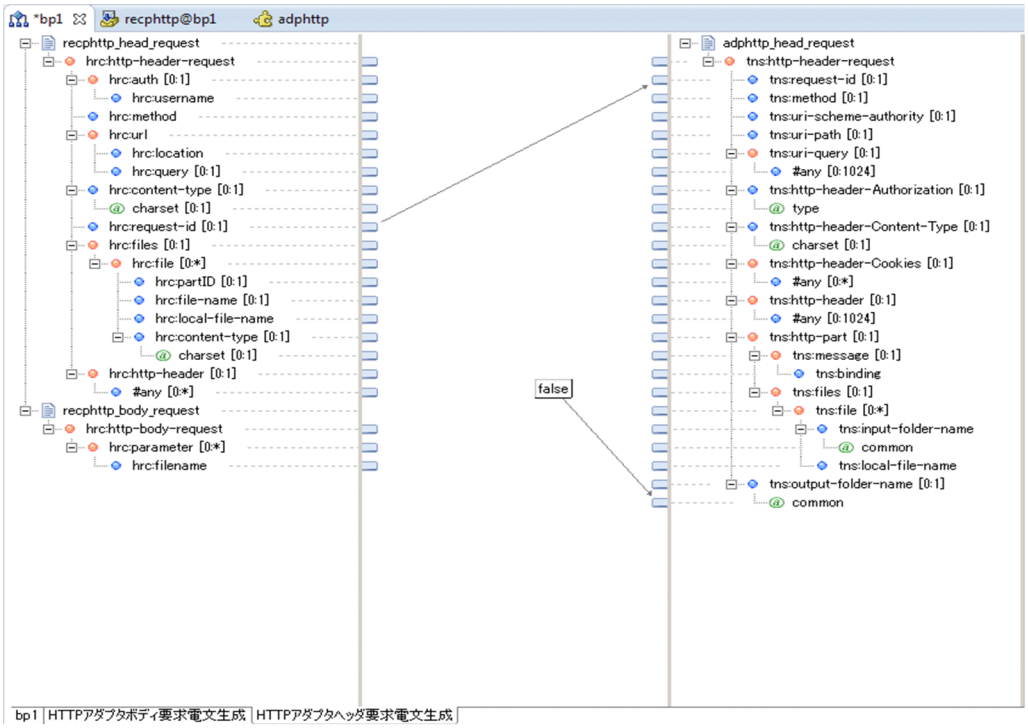
［ルート要素選択］ダイアログが表示されます。



6. [OK] ボタンをクリックします。

マッピング定義画面が表示されます。

マッピング元には recphttp_body_request 変数が追加されています。



このあと，recphttp_body_request の filename 要素を使って，HTTP アダプタが取得するファイル名を HTTP アダプタ実行環境プロパティファイルの設定値から変更するマッピングを定義します。

7. const ファンクションを配置して，const ファンクションをダブルクリックします。

［定数］ダイアログが表示されます。

8.「文字列」を選択して「値」に「/」を入力します。

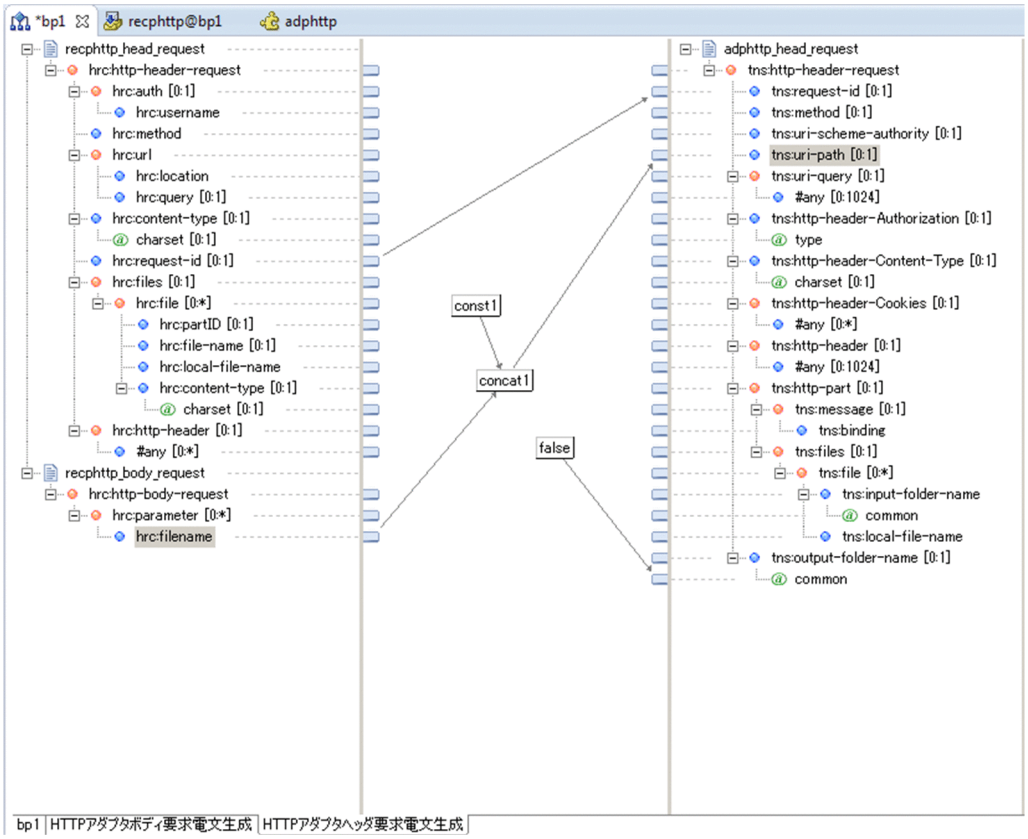


9. [OK] ボタンをクリックします。
   [定数] ダイアログが閉じます。

10. concat ファンクションを配置します。

11. 次の表を参考にマッピングします。
    マッピングする順番に注意してください。マッピング元およびマッピング先はルート要素からのパスで記載しています。

順番	マッピング元	マッピング先
1	const ファンクション	concat ファンクション
2	/hrc:http-body-request/hrc:parameter/hrc:filename	concat ファンクション
3	concat ファンクション	/tns:http-header-request/tns:uri-path

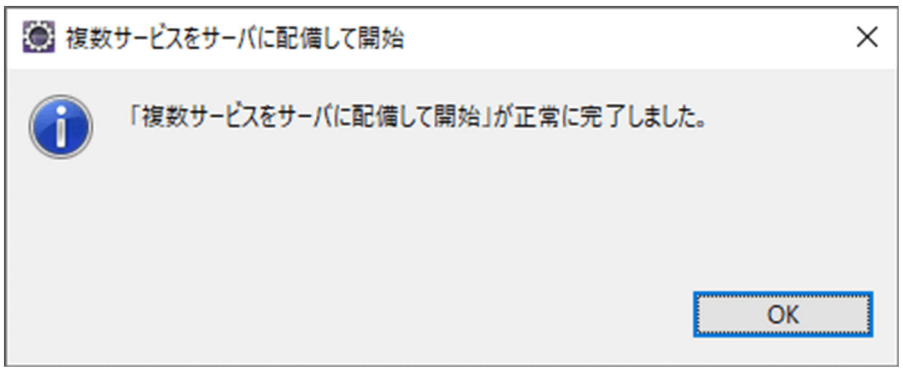


(5)　ビジネスプロセスを実行するための準備

定義した HCSC コンポーネントを配備定義します。操作手順を次に示します。

1. [ファイル] − [すべて保管] を選択します。
   HTTP アダプタ，HTTP 受付，およびビジネスプロセスの定義が保存されます。
2. ツリービューの［サービス定義一覧］を右クリックして，［複数サービスをサーバに配備して開始］を選択します。
3. サービス選択ページで配備および開始するサービスを選択します。
4. [Finish] 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. [OK] ボタンをクリックします。

次のダイアログが表示されていれば処理が正常に完了しています。



失敗した場合は，エラー内容を基にここまで定義した内容に誤りがないか確認してください。

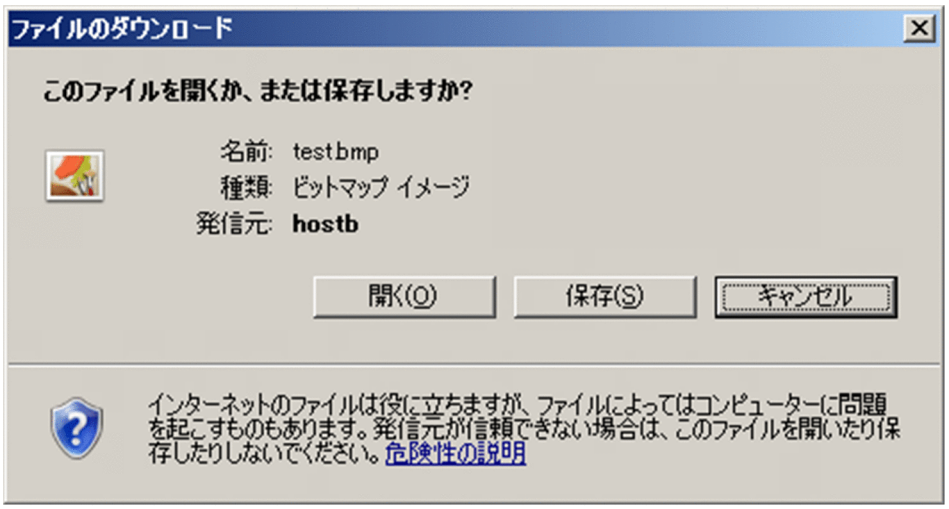
(6) ビジネスプロセスの実行

1. Web ブラウザを起動して次の URL にアクセスします。

   http://hostB/rcp1/get?filename=test2.bmp

   URL 中の「hostB」は，hostB の IPv4 形式の IP アドレスに名前解決されるものとします。ホスト名以下のパスは「/受付 ID/受付のオペレーション名」です。クエリには，定義した filename のパラメタを指定します。

ビジネスプロセスが正しく実行されている場合は，HTTP 受付からファイルをダウンロードできます。Web ブラウザでは次のポップアップが出力されます。



注

「付録 G.1(9)(c) HTTP 受付ボディ応答電文生成」で，ダウンロードするファイル名を静的に設定しているため，ファイル名は設定例 1 と同じ「test.bmp」となります。

設定例 2 の実行結果から，HTTP アダプタ実行環境プロパティファイルで設定した接続先のパスを HTTP アダプタの要求電文（ヘッダ）で上書きしたことを確認できます。

参考

ダウンロードするファイルの違いを確認したい場合は，次のように設定例 1 のファイルをダウンロードするようにパラメタ値を変更して URL にアクセスしてください。

http://hostB/rcp1/get?filename=test.bmp

付録 H　FTP 連携で必要なセキュリティの設定

FTP 連携で必要なセキュリティの設定について説明します。

付録 H.1　FTPS を使用したセキュアな接続（FTP アダプタ）

FTP アダプタでは，FTPS（File Transfer Protocol over SSL/TLS）による FTP サーバとの接続をサポートしています。FTPS による接続を利用することで，認証に必要なユーザ名やパスワードを暗号化し，第三者に盗聴される危険性を低下できます。

(1)　FTPS に関する設定

FTPS に関する設定について説明します。なお，FTP アダプタでは，要求電文または FTP アダプタ実行環境プロパティファイルで設定できます。要求電文と FTP アダプタ実行環境プロパティファイルの両方で設定された場合は，要求電文が優先されます。

表 H–1　FTP に関する設定

項目	説明
FTPS を使用する／使用しない	FTP サーバとの接続に，通常の FTP を使用するか FTPS を使用するか設定できます。デフォルトでは通常の FTP が使用されます。
使用するセキュリティ通信用のプロトコル	セキュリティ通信用のプロトコルとして，次のどちらかを選択できます。 • SSL • TLS
FTPS の Implicit モードを使用する／使用しない	次のどちらかを選択できます。 • Implicit モード FTP サーバに接続した直後から暗号化された通信を開始するモードです。 • Explicit モード FTP サーバに接続したあと，FTP クライアント側で AUTH コマンドを実行してから暗号化された通信を開始するモードです。 なお，FTP サーバ側で使用する制御コネクションのポート番号は，Explicit モードと Implicit モードで異なります。そのため，モードを変更する場合は接続先 FTP サーバの制御コネクションポート番号も変更する必要があります。
データコネクションの通信を暗号化する／しない	FTP サーバとの接続に FTPS を使用する場合，データコネクションの通信を暗号化するかどうかを設定できます。
サーバ認証をする／しない	次のどちらかを選択できます。 • サーバ認証をする 接続先 FTP サーバの公開鍵の証明書が信頼できるかどうかをチェックし，信頼できる場合だけ処理を実行します。 • サーバ認証をしない 接続先 FTP サーバの公開鍵の証明書が信頼できるかどうかをチェックしないで，処理を実行します。

要求電文で設定できる項目の詳細は，「3.3.10(1)(a)　FTP アダプタの要求電文フォーマット」を参照してください。FTP アダプタ実行環境プロパティファイルで設定できる項目の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「6.8.3　FTP アダプタ実行環境プロパティファイル」を参照してください。

注意事項

FTP アダプタでは，クライアント認証を必要とする FTPS サーバには接続できません。

(2)　キーストアに関する設定

FTPS を使用する場合で，かつ FTP アダプタでサーバ認証をするとき，信頼できる証明書を格納したキーストア（トラストストア）が必要になります。

トラストストアには，次の証明書のどれかが格納されている必要があります。

- 接続先 FTP サーバの公開鍵の証明書
- 接続先 FTP サーバの公開鍵の証明書が CA から発行された証明書の場合は，この証明書を発行した CA の証明書
- 上記の CA が中間 CA の場合は，中間 CA からルート CA までのどれかに該当する CA の証明書

キーストアへの証明書のインポート方法については，JDK のドキュメントの「keytool」に関する個所を参照してください。

キーストアのパスおよびパスワードに関する設定方法を次に示します。

キーストアのパス

サーバ認証をする場合，FTP アダプタは次の順にキーストアを探し，最初に発見したトラストストアを使用してサーバ認証を実施します。

1. usrconf.properties（J2EE サーバ用ユーザプロパティファイル）の javax.net.ssl.trustStore プロパティが指定されている場合，このシステムプロパティに指定されたパスに存在するキーストアを使用します。
2. 次のパスにファイルが存在する場合，このファイルをキーストアとして使用します。
   <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¥lib¥security¥jssecacerts
3. 次のパスにファイルが存在する場合，このファイルをキーストアとして使用します。
   <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¥lib¥security¥cacerts

上記のどれにも該当しない場合は，信頼できないサーバと判断されます。

キーストアのパスワード

usrconf.properties（J2EE サーバ用ユーザプロパティファイル）の javax.net.ssl.trustStorePassword プロパティでキーストアのパスワードが指定されている場合，キーストアを使用する前に，キーストアのデータが完全かどうかの検査が実施されます。

javax.net.ssl.trustStorePassword プロパティが指定されていない場合は，キーストアのデータの検査が実施されません。

usrconf.properties（J2EE サーバ用ユーザプロパティファイル）の詳細については，次に示すマニュアルを参照してください。

- J2EE サーバの互換モードが推奨モードの場合

  マニュアル「アプリケーションサーバ リファレンス 定義編(サーバ定義)」の「2.2.3　usrconf.properties（J2EE サーバ用ユーザプロパティファイル)」
- J2EE サーバの互換モードが V9 互換モードの場合

  マニュアル「アプリケーションサーバ 機能解説 互換編」の「10.1.1　usrconf.properties（J2EE サーバ用ユーザプロパティファイル)」

付録 H.2　セキュアプロトコルによる暗号化・認証（FTP インバウンドアダプタ）

FTP インバウンドアダプタでは，FTPS を使用してユーザ名やパスワードを含む送受信データを暗号化できます。なお，FTP インバウンドアダプタでは，サーバ認証だけをサポートしています。

ここでは，FTP インバウンドアダプタでサポートする FTPS 関連プロトコル，コマンド，および FTPS に関する設定について説明します。

(1)　サポートする FTPS 関連プロトコルおよびコマンド

FTP インバウンドアダプタでは，次の FTPS 関連プロトコルをサポートしています。

- SSL
- TLS

また，次の FTPS 関連コマンドをサポートしています。

- AUTH
- PBSZ
- PROT

(2)　FTP インバウンドアダプタの設定

FTPS に関する設定は，FTP インバウンドアダプタの属性ファイルで実施できます。

FTP インバウンドアダプタの属性ファイルの設定方法および定義項目の詳細は，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「3.2.3　FTP インバウンドアダプタをセットアップする」の，FTP インバウンドアダプタの設定に関する説明を参照してください。

(3) キーストアの設定

FTP インバウンドアダプタで使用するキーストアは，JKS 形式だけをサポートしています。JKS 形式のキーストアおよびキーは，JDK に付属する keytool コマンドで作成してください。keytool コマンドの詳細は，keytool のドキュメントを参照してください。

なお，複数のキーを格納したキーストアを使用する場合は，使用しないキーも含めて，すべてのキーのパスワードを同じにする必要があります。

また，keytool コマンドでは，alias オプション指定時の大文字が小文字として保存されます。このため，FTP インバウンドアダプタの属性ファイルの nioListener_ftps_keystore_keyAlias 属性には，大文字を使用しないで，keytool コマンドの list オプションの結果に表示されるエイリアス名を指定してください。

付録 H.3 セットアップ時の属性ファイルの定義項目（FTP インバウンドアダプタ）

FTP インバウンドアダプタのセットアップ時の属性ファイルの定義項目で，FTPS に関する設定について説明します。そのほかの定義項目については，マニュアル「サービスプラットフォーム システム構築・運用ガイド」の「3.2.3 FTP インバウンドアダプタをセットアップする」を参照してください。

表 H–2 属性ファイルの定義項目一覧（FTPS に関する設定）

定義項目	定義名	値の範囲	デフォルト値	補足
FTPS 有効化	server_ftps_enable	true または false	false	true を指定した場合は，FTPS に関する設定が有効になります。 false を指定した場合は，FTPS に関する設定がすべて無視されます。
制御コネクションの Implicit モード	nioListener_ftps_implicitMode	true または false	false	true を指定した場合は，Implicit モードで動作します。 false を指定した場合は，Explicit モードで動作します。
キーストアのファイルパス※	nioListener_ftps_keystore_file	任意の文字列	空文字列	絶対パスで記述してください。
キーストアのパスワード	nioListener_ftps_keystore_password	6 文字以上の任意の文字列	空文字列	−
キーのパスワード	nioListener_ftps_keystore_keyPassword	6 文字以上の任意の文字列	空文字列	この項目を指定しなかった場合は，キーストアのパスワードをキーのパスワードと見なします。
キーのエイリアス	nioListener_ftps_keystore_keyAlias	任意の文字列	空文字列	この項目を指定しなかった場合は，キーストアに含まれる最初のキー（キー証明書のフィンガープリントのバイト列を昇順にしたときの最初のキー）を利用します。

定義項目	定義名	値の範囲	デフォルト値	補足
データコネクションの暗号化	nioListener_ftps_dataConnection_secure	true または false	true	PROT コマンドを指定していない場合，この項目の指定が有効になります。PROT コマンドが成功した場合は，その指定が有効になります。 true を指定した場合は，データコネクションを暗号化します。 false を指定した場合は，暗号化しません。

（凡例）

－：該当する内容はありません。

注※

指定したパスに，「//」，「/./」，「/../」が含まれる場合は，エラーになります。なお，Windows の場合は，大文字と小文字は区別されません。

付録 I　HTTP アダプタで必要なセキュリティの設定

HTTP アダプタで必要なセキュリティの設定について説明します。

付録 I.1　HTTPS を使用したセキュアな接続

HTTP アダプタでは，HTTPS によるサーバとの接続をサポートしています。HTTPS による接続を利用することで，認証に必要なユーザ名やパスワードを暗号化し，第三者に盗聴される危険性を低下できます。

(1)　HTTPS に関する設定

HTTPS に関する情報として設定する項目を次に示します。

表 I-1　HTTPS に関する設定

項目	設定内容
HTTPS の使用有無	HTTPS を使用する場合は URI のスキーマに「https」を指定します。
HTTPS を使用する場合に使用するセキュリティ通信用プロトコル	TLS1.0，SSL3.0 など，使用できるプロトコルのうち，最適なものが自動で設定されます。
サーバ認証の有無	サーバ認証の有無は HTTP アダプタ定義ファイルの adphttp.protocol.https-server-authentication プロパティで設定できます。 • サーバ認証をする場合（デフォルト）： 接続先 HTTP サーバの公開鍵の証明書が信頼できるかどうかをチェックして，信頼できる場合だけ処理を実行します。 • サーバ認証をしない場合： 接続先 HTTP サーバの公開鍵の証明書が信頼できるかどうかをチェックしないで処理を実行します。
ホスト名検証の有無	ホスト名検証の有無は HTTP アダプタ定義ファイルの adphttp.protocol.https-hostname-verification プロパティで設定できます。 • ホスト名検証をする場合（デフォルト）： 証明書内のホスト名が接続先 URI のホスト名と一致するかをチェックして，処理を実行します。 • ホスト名検証をしない場合： 証明書内のホスト名が接続先 URI のホスト名と一致するかをチェックしないで，処理を実行します。

HTTP アダプタ定義ファイルの詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.10.1　HTTP アダプタ定義ファイル」を参照してください。

注意事項

HTTP アダプタでは，クライアント認証を必要とする HTTPS サーバには接続できません。

(2) キーストアに関する設定

HTTPS を使用する場合で，かつ HTTP アダプタでサーバ認証をするとき，信頼できる証明書を格納したキーストア（トラストストア）が必要になります。

トラストストアには，次の証明書のどれかが格納されている必要があります。

- 接続先 HTTP サーバの公開鍵の証明書
- 上記証明書が CA から発行された証明書の場合，上記証明書を発行した CA の証明書
- 上記の CA が中間 CA の場合は，中間 CA からルート CA までのどれかに該当する CA の証明書

キーストアへの証明書のインポート方法については，JDK のドキュメントの「keytool」に関する個所を参照してください。

キーストアのパスおよびパスワードに関する設定方法を次に示します。

キーストアのパス

サーバ認証をする場合，HTTP アダプタは次の順にキーストアを探し，最初に発見したトラストストアを使用してサーバ認証を実施します。

1. usrconf.properties（J2EE サーバ用ユーザプロパティファイル）の javax.net.ssl.trustStore プロパティが指定されている場合，このシステムプロパティに指定されたパスに存在するキーストアを使用します。
2. 次のパスにファイルが存在する場合，このファイルをキーストアとして使用します。
   <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¥lib¥security¥jssecacerts
3. 次のパスにファイルが存在する場合，このファイルをキーストアとして使用します。
   <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¥lib¥security¥cacerts

上記のどれにも該当しない場合は，信頼できないサーバと判断されます。

キーストアのパスワード

usrconf.properties（J2EE サーバ用ユーザプロパティファイル）の javax.net.ssl.trustStorePassword プロパティでキーストアのパスワードが指定されている場合，キーストアを使用する前に，キーストアのデータが完全かどうかの検査が実施されます。

javax.net.ssl.trustStorePassword プロパティが指定されていない場合は，キーストアのデータの検査が実施されません。

usrconf.properties（J2EE サーバ用ユーザプロパティファイル）の詳細については，次に示すマニュアルを参照してください。

- J2EE サーバの互換モードが推奨モードの場合
  マニュアル「アプリケーションサーバ リファレンス 定義編(サーバ定義)」の「2.2.3　usrconf.properties (J2EE サーバ用ユーザプロパティファイル)」
- J2EE サーバの互換モードが V9 互換モードの場合

マニュアル「アプリケーションサーバ 機能解説 互換編」の「10.1.1　usrconf.properties（J2EE サーバ用ユーザプロパティファイル）」

付録 J　メールアダプタで必要なセキュリティの設定

メールアダプタでは，TLS（SMTP over TLS）によるサーバとの接続をサポートしています。TLS による接続を利用することで，認証に必要なユーザ名やパスワードを暗号化し，第三者に盗聴される危険性を低下できます。

付録 J.1　TLS に関する設定

TLS に関する情報として設定する項目を次に示します。

表 J-1　TLS に関する設定

項目	設定内容
TLS の使用有無	次のどちらかを選択できます。 • TLS を使用する TLS を使用します。TLS を使用して SMTP サーバに接続します。 • TLS を使用しない（デフォルト） TLS を使用しません。SMTP で SMTP サーバに接続します。
TLS で使用するセキュリティ通信用プロトコル	使用できるプロトコルバージョンのうち，最適なものが自動で設定されます。
TLS で使用できる TLS 暗号スイート	TLS 接続で使用できる TLS 暗号スイートは，メールアダプタで制限しません。
STARTTLS の使用有無	SMTP サーバとの接続に TLS を使用する場合，次のどちらかを選択できます。 • STARTTLS を使用する（デフォルト） 通常の SMTP で通信を開始し，「STARTTLS」と呼ばれる手順で双方が TLS に対応しているか確認したあとに TLS で再接続します。 • STARTTLS を使用しない SMTP 通信がすべて TLS によって暗号化されます。
接続先の SMTP サーバで STARTTLS が処理できない場合，メールアダプタでエラーを発生させるかどうか	SMTP サーバとの接続に TLS と STARTTLS を使用する場合，次のどちらかを選択できます。 • エラーを発生させる 接続先の SMTP サーバが STARTTLS をサポートしていない場合，または STARTTLS の処理中にエラーが発生した場合，メールアダプタでエラーを発生させます。 • エラーを発生させない（デフォルト） 接続先の SMTP サーバが STARTTLS をサポートしていない場合，または STARTTLS の処理中にエラーが発生した場合，SMTP を使用して SMTP サーバに接続します。
TLS で使用するメール送信ポート番号	メール送信用のポート番号は，要求電文 smtp-port 要素，またはメールアダプタ実行環境プロパティファイルの mailadp.smtp.port プロパティで設定できます。 TLS と STARTTLS を使用する場合，一般的にポート番号に 25 番または 587 番を使用します。TLS を使用し，STARTTLS を使用しない場合，一般的にポート番号に 465 番を使用します。

TLS に関する情報は，要求電文またはメールアダプタ実行環境プロパティファイルで設定できます。要求電文で設定できる項目の詳細は，「3.3.12(2)　電文フォーマットの作成」を参照してください。メールアダプタ実行環境プロパティファイルで設定できる項目の詳細は，マニュアル「サービスプラットフォームリファレンス」の「6.13.3　メールアダプタ実行環境プロパティファイル」を参照してください。

付録 J.2　キーストアに関する設定

TLS を使用する場合で，かつメールアダプタでサーバ認証をするとき，信頼できる証明書を格納したキーストア（トラストストア）が必要になります。

トラストストアには，次の証明書のどれかが格納されている必要があります。

- 接続先 SMTP サーバの公開鍵の証明書
- 上記証明書が CA から発行された証明書の場合，上記証明書を発行した CA の証明書
- 上記の CA が中間 CA の場合は，中間 CA からルート CA までのどれかに該当する CA の証明書

キーストアへの証明書のインポート方法については，JDK のドキュメントの「keytool」に関する個所を参照してください。

キーストアのパスおよびパスワードに関する設定方法を次に示します。

キーストアのパス

サーバ認証をする場合，SMTP アダプタは次の順にキーストアを探し，最初に発見したトラストストアを使用してサーバ認証を実施します。

1. usrconf.properties（J2EE サーバ用ユーザプロパティファイル）の javax.net.ssl.trustStore プロパティが指定されている場合，このシステムプロパティに指定されたパスに存在するキーストアを使用します。
2. 次のパスにファイルが存在する場合，このファイルをキーストアとして使用します。
   <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¥lib¥security¥jssecacerts
3. 次のパスにファイルが存在する場合，このファイルをキーストアとして使用します。
   <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¥lib¥security¥cacerts

上記のどれにも該当しない場合は，信頼できないサーバと判断されます。

キーストアのパスワード

usrconf.properties（J2EE サーバ用ユーザプロパティファイル）の javax.net.ssl.trustStorePassword プロパティでキーストアのパスワードが指定されている場合，キーストアを使用する前に，キーストアのデータが完全かどうかの検査が実施されます。

javax.net.ssl.trustStorePassword プロパティが指定されていない場合は，キーストアのデータの検査が実施されません。

usrconf.properties（J2EE サーバ用ユーザプロパティファイル）の詳細については，次に示すマニュアルを参照してください。

- J2EE サーバの互換モードが推奨モードの場合

  マニュアル「アプリケーションサーバ リファレンス 定義編(サーバ定義)」の「2.2.3　usrconf.properties (J2EE サーバ用ユーザプロパティファイル)」

- J2EE サーバの互換モードが V9 互換モードの場合

  マニュアル「アプリケーションサーバ 機能解説 互換編」の「10.1.1　usrconf.properties（J2EE サーバ用ユーザプロパティファイル)」

付録 K　gRPC アダプタで必要なセキュリティの設定

gRPC アダプタで必要なセキュリティの設定について説明します。

付録 K.1　キーストアに関する設定

gRPC アダプタ実行環境プロパティファイルの adpgrpc.security.tls.certificate-chain でサーバのルート証明書を設定することができます。

この設定を省略した場合，信頼できる証明書を格納したキーストア（トラストストア）が必要になります。トラストストアに，次のどれかが格納されていることを確認してください。

- 接続先 gRPC サーバの公開鍵の証明書
- 上記証明書が CA から発行された証明書の場合，上記証明書を発行した CA の証明書
- 上記 CA が中間 CA の場合，中間 CA からルート CA までのどれかに該当する CA の証明書

キーストアへの証明書のインポート方法については，JDK のドキュメントの「keytool」に関する個所を参照してください。

トラストストアのパスおよびパスワードに関する設定方法を次に示します。

トラストストアのパス

サーバ認証をする場合，gRPC アダプタは次の順にキーストアを探し，最初に発見したトラストストアを使用してサーバ認証を実施します。

1. usrconf.properties（J2EE サーバ用ユーザプロパティファイル）の javax.net.ssl.trustStore プロパティが指定されている場合，このシステムプロパティに指定されたパスに存在するキーストアを使用します。

   システムプロパティを指定する場合は，J2EE サーバ用ユーザプロパティファイル（usrconf.properties ファイル）に次の形式で指定します。

   ```
   javax.net.ssl.trustStore =<トラストストアの絶対パス>
   ```

2. 次のパスにファイルが存在する場合，このファイルをトラストストアとして使用します。

   <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¥lib¥security¥jssecacerts

3. 次のパスにファイルが存在する場合，このファイルをトラストストアとして使用します。

   <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¥lib¥security¥cacerts

トラストストアのパスワード

usrconf.properties（J2EE サーバ用ユーザプロパティファイル）の javax.net.ssl.trustStorePassword プロパティでトラストストアのパスワードが指定されている場合，トラストストアを使用する前に，トラストストアのデータが完全かどうかの検査が実施されます。

システムプロパティを指定する場合は，J2EE サーバ用ユーザプロパティファイル（usrconf.properties ファイル）に次の形式で指定します。

javax.net.ssl.trustStorePassword =<トラストストアのパスワード>

javax.net.ssl.trustStorePassword プロパティが指定されていない場合は，トラストストアのデータの検査が実施されません。

付録 L　ユーザ定義受付（呼出先選択）の定義

ユーザ定義受付には，特定のビジネスプロセスを呼び出すユーザ定義受付（呼出先固定）と，呼び出すビジネスプロセスを選択できるユーザ定義受付（呼出先選択）があります。

ユーザ定義受付（呼出先選択）では，呼び出しサービス選択機能を使用して，HTTP 受付から任意のビジネスプロセスを呼び出します。

呼び出しサービス選択機能については，マニュアル「サービスプラットフォーム 解説」の「2.15.11 HTTP 受付の呼び出しサービス選択機能」を参照してください。

付録 L.1　ユーザ定義受付（呼出先選択）の追加

ここでは，ユーザ定義受付（呼出先選択）の HTTP 受付を追加する手順について説明します。

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

2. ［HCSC-Definer］－［HCSCTE ビュー］を選択し，［開く］ボタンをクリックします。
   ツリービューにサービス定義一覧が表示されます。

3. ツリービューのサービス定義一覧で右クリックして［ユーザ定義受付（呼出先選択）追加］を選択します。
   受付種別選択ウィザードが表示されます。

4. ［次へ］ボタンをクリックします。
   HTTP 受付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. ［受付名］を指定します。
   HTTP 受付の名称を 1～40 バイトで指定します。
   受付名は，サービス定義一覧上のほかのユーザ定義受付（呼出先選択）と重複しないように指定してください。

6. ［終了］ボタンをクリックします。
   ［終了］ボタンをクリックすると，必要なファイルが作成され，リポジトリに保存されます。ユーザ定義受付定義画面（基本）が表示されます。

7. HTTP 受付の定義情報を設定します。
   ユーザ定義受付（呼出先選択）では，［オペレーション］に HTTP リクエスト時に指定した HTTP メソッドを指定します。
   そのほかの HTTP 受付の設定項目については，「2.5.4　HTTP 受付を追加する」を参照してください。

また，HTTP 受付を利用した処理や構成例については，マニュアル「サービスプラットフォーム 解説」の「2.15　HTTP 通信を使用したクライアントとの接続」を参照してください。

8. ユーザ定義受付（詳細）タブをクリックします。

ユーザ定義受付定義画面（詳細）が表示されます。

9. HTTP 受付の定義情報を設定します。

HTTP 受付の設定項目について次の表に示します。

表 L–1　HTTP 受付のユーザ定義受付定義画面（詳細）の設定項目

分類	項目	設定内容	設定
ユーザ定義受付制御情報	独自定義ファイル	次の独自定義ファイルが表示されます。必要に応じて変更します。 • cscurecphttp.properties※1 • cscurecphttp_selection.properties※2 • cscurecphttp_header.properties※3 • cscurecphttp_routing.xml※4 ここに設定されているのはテンプレートファイルです。修正が必要な場合はファイルを選択して［編集］ボタンをクリックし，独自定義ファイルの内容を修正してください。	◎
	EAR ファイル	次の EAR ファイルが設定されていることを確認します。 • cscmsg_urecp_http.ear※5	×

（凡例）

◎：必ず設定します。

×：表示されている内容を確認します。

注※1

cscurecphttp.properties の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.11.3　HTTP 受付定義ファイル」を参照してください。

注※2

ユーザ定義受付（呼出先選択）では，HTTP 受付サービス選択定義ファイル（cscurecphttp_selection.properties）に，呼び出し先のビジネスプロセス名，HTTP メソッド（オペレーション名）など，ビジネスプロセスを呼び出すために必要な設定項目を必ず定義してください。定義しない場合は，受付起動時にエラーとなります。

cscurecphttp_selection.properties の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.11.2　HTTP 受付サービス選択定義ファイル」を参照してください。

注※3

cscurecphttp_header.properties の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.11.1　HTTP レスポンスヘッダ定義ファイル」を参照してください。

注※4

cscurecphttp_routing.xml の詳細は，マニュアル「サービスプラットフォーム リファレンス」の「3.11.4　HTTP 受付ルーティング定義ファイル」を参照してください。

注※5

ファイルの格納先ディレクトリは「<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¥custom-reception¥http¥lib」です。

付録 L.2　ユーザ定義受付（呼出先選択）の複製

ユーザ定義受付（呼出先選択）の複製については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「5.2.2　定義済みビジネスプロセスを利用してビジネスプロセスを追加する」を参照してください。

(1)　ユーザ定義受付（呼出先選択）の複製時の受付名の規則

複製先のユーザ定義受付（呼出先選択）の受付名は次のように設定されます。

```
CopyN_複製元のユーザ定義受付（呼出先選択）の受付名
```

N は，同一複製元の受付名に対して使用されている「Copy*N*_」の *N* の中での最小の整数です。ただし，複製後の受付名の長さが 40 バイトを超える場合，40 バイトを超えた文字は削除されます。複製先の受付名が既存の受付名と重複した場合は，「Copy*N*_」の *N* の値を変更します。

付録 L.3　ユーザ定義受付（呼出先選択）の検証

ユーザ定義受付（呼出先選択）の検証では，ユーザ定義受付の定義内容の検証，およびデータ変換定義の妥当性の検証を行います。ビジネスプロセスとユーザ定義受付のインターフェースの整合性の検証は行いません。

ビジネスプロセスとユーザ定義受付のインターフェースの整合性の検証については，「2.14　ユーザ定義受付の検証」を参照してください。

付録 L.4　ユーザ定義受付（呼出先選択）の削除

ユーザ定義受付（呼出先選択）の削除については，「2.16　ユーザ定義受付の削除」を参照してください。

付録 L.5　ユーザ定義受付（呼出先選択）のパッケージングと配備定義

ユーザ定義受付（呼出先選択）のパッケージングと配備定義については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「8.　HCSC コンポーネントのパッケージングと配備定義」を参照してください。

(1)　ユーザ定義受付（呼出先選択）の配備できる条件

ユーザ定義受付（呼出先選択）は，パッケージングでリポジトリに EAR ファイルが作成されているユーザ定義受付（呼出先選択）だけが配備できます。リポジトリに EAR ファイルが作成されていないユーザ定義受付（呼出先選択）は，［サービス追加］ダイアログ，および［複数サービス追加］ダイアログには表示されません。

(2) 複数クラスタ環境でユーザ定義受付（呼出先選択）を配備できる条件

複数クラスタ（HCSC サーバ）の環境では，1 つのクラスタにだけユーザ定義受付（呼出先選択）を配備できます。

例えば，クラスタ A にユーザ定義受付（呼出先選択）が配備されているとき，クラスタ B やクラスタ C を選択した［サービス追加］ダイアログにユーザ定義受付（呼出先選択）を配備しようとするとエラーとなります。

［複数サービス追加］ダイアログの場合は，クラスタ B やクラスタ C を選択したダイアログにユーザ定義受付（呼出先選択）が表示されますが，ユーザ定義受付（呼出先選択）は選択できません。

付録 L.6 ユーザ定義受付（呼出先選択）のリポジトリ管理

ユーザ定義受付（呼出先選択）のリポジトリ管理については，マニュアル「サービスプラットフォーム 開発ガイド 基本開発編」の「3.2 リポジトリの管理」を参照してください。

付録 M　用語解説

マニュアルで使用する用語について

マニュアル「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。

索引

C

CSCMsgCustomAdapterConstant クラス 574
CSCMsgCustomAdapterContext インターフェース 559
CSCMsgCustomProtocolConverter インターフェース 555
CSCMsgMessageConstant インターフェース 562
CSCMsgRequestMessage インターフェース 563
CSCMsgResponseMessage インターフェース 568
CSV データ書き込みサンプル 672
CSV データ読み込みサンプル 644

D

DB アダプタ定義支援機能の概要 606
DB アダプタ定義支援機能を利用した DB アダプタの定義 606
DB アダプタの設定 629
DB アダプタの定義 606
DB アダプタ用の XML フォーマット定義ファイルの作成 196
DB アダプタ用の XML フォーマット定義ファイルの作成例 197
DB アダプタを新規に追加する 152
DB アダプタを定義する 179
dcrapautoconnect オペランドの指定によるメリットおよびデメリット 221

E

EAR ファイルの入れ替え 204

F

FTPS に関する設定 757
FTPS を使用したセキュアな接続 757
FTP アダプタの応答電文フォーマット 324
FTP アダプタの定義ファイルの作成 333
FTP アダプタの要求電文フォーマット 274
FTP アダプタを新規に追加する 155
FTP アダプタを定義する 273
FTP 受付コンフィグファイル 48
FTP 受付定義ファイル 47
FTP 受付の応答電文フォーマット 42
FTP 受付の概要 20
FTP 受付の定義 36
FTP 受付の定義の流れ 36
FTP 受付の定義ファイルを作成する 46
FTP 受付の電文フォーマットを作成する 37
FTP 受付の要求電文フォーマット 37
FTP 受付を追加する 48
FTP コマンド許可リスト定義ファイル（FTP 受付用） 47
FTP 実行許可リスト定義ファイル 47

G

gRPC アダプタで必要なセキュリティの設定 768
gRPC アダプタの応答電文フォーマット 428
gRPC アダプタの定義ファイルの作成 433
gRPC アダプタのフォルト電文フォーマット 429
gRPC アダプタの要求電文フォーマット 427
gRPC アダプタを新規に追加する 160
gRPC アダプタを定義する 427
gRPC 受付の定義 98
gRPC 受付の定義の流れ 98
gRPC 受付を追加する 99

H

HTTP アダプタで必要なセキュリティの設定 762
HTTP アダプタの定義ファイルの作成および編集 399
HTTP アダプタを新規に追加する 157
HTTP アダプタを定義する 377
HTTP 受付および HTTP アダプタを利用したビジネスプロセスの設定例 715
HTTP 受付の概要 20
HTTP 受付の設定例（パススルーモードの場合） 709
HTTP 受付の設定例（標準モードの場合） 698
HTTP 受付の定義 51

HTTP 受付の定義の流れ 51
HTTP 受付の定義ファイルを作成する 63
HTTP 受付の電文フォーマットを作成する 53
HTTP 受付のルーティング機能の設定例 712
HTTP 受付を追加する 64
HTTP 受付を利用したビジネスプロセスの設定例 698

J

J2EE サーバ用オプション定義ファイルの設定（HTTP アダプタの場合） 377
JAR ファイルの作成〔プロトコルコンバータ JAR ファイル〕 444
JAR ファイルの作成〔ライブラリ JAR ファイル〕 445

K

Kafka アダプタの応答電文フォーマット 424
Kafka アダプタの要求電文フォーマット 421
Kafka アダプタを新規に追加する 159
Kafka アダプタを定義する 421
Kafka 受付の定義 85
Kafka 受付の定義の流れ 85
Kafka 受付の定義ファイルを作成する 88
Kafka 受付の電文フォーマットを作成する 86
Kafka 受付を追加する 88

M

MDB（DB キュー）アダプタを新規に追加する 151
MDB（DB キュー）アダプタを定義する 176
MDB（WS-R）アダプタを新規に追加する 151
MDB（WS-R）アダプタを定義する 173
Message Queue アダプタ環境定義ファイルの編集 269
Message Queue アダプタ実行環境プロパティファイルの作成 273
Message Queue アダプタ通信構成定義ファイルの編集 269
Message Queue アダプタの場合 468
Message Queue アダプタを新規に追加する 154
Message Queue アダプタを定義する 255
Message Queue 受付の概要 20
Message Queue 受付の定義 70
Message Queue 受付の定義の流れ 70
Message Queue 受付の定義ファイルを作成する 75
Message Queue 受付の電文フォーマットを作成する 72
Message Queue 受付を追加する 76
MHP を接続先とした TCP/IP 接続をする場合 223

O

Object Access アダプタ実行環境プロパティファイルの作成 254
Object Access アダプタ通信構成定義ファイルの作成 250
Object Access アダプタの場合 466
Object Access アダプタを新規に追加する 154
Object Access アダプタを定義する 242

S

SessionBean アダプタを新規に追加する 150
SessionBean アダプタを定義する 167
SFTP アダプタの応答電文フォーマット 417
SFTP アダプタの定義ファイルの作成 419
SFTP アダプタの要求電文フォーマット 412
SFTP アダプタを新規に追加する 159
SFTP アダプタを定義する 411
SOAP アダプタを新規に追加する 149
SOAP アダプタを定義する 162
SOAP 受付の WSDL 作成時の注意事項 135
SOAP 受付の概要 18
SOAP 受付の追加 28
SOAP 受付の定義 22
SOAP 受付の定義の流れ 22
SOAP 受付の定義を編集する 31
SQL オペレーション定義ファイル 179
SQL オペレーション定義ファイルの構造 179
SQL オペレーション定義ファイルの作成 179
SQL オペレーション定義ファイルの作成例 195

T

TP1/Client/J 環境定義の設定 219

TP1/RPC 受付の概要　19
TP1/RPC 受付の定義　32
TP1/RPC 受付の定義の流れ　32
TP1/RPC 受付の定義を編集する　35
TP1/RPC 受付を追加する　33
TP1 アダプタ環境定義ファイルの編集　218
TP1 アダプタ通信構成定義ファイルの編集　218
TP1 アダプタのサービスアダプタ定義画面（基本）の設定項目　461
TP1 アダプタの設定項目の反映　219
TP1 アダプタの通信構成定義および TP1/Client/J 環境定義の対応　220
TP1 アダプタを新規に追加する　153
TP1 アダプタを定義する　216
TPBroker 環境定義ファイルの作成　254

W

WSDL の作成　111
WSDL の作成例　113

X

XML データ書き込みサンプル　657
XML データ読み込みサンプル　632
XML フォーマット定義ファイルの作成　228

あ

アクティビティを設定する〔gRPC 受付〕　105
アクティビティを設定する〔Kafka 受付〕　90
アクティビティを設定する〔HTTP 受付〕　66
アダプタコンテキストインターフェース〔カスタムアダプタ開発フレームワークの API〕　559
アダプタのサービスアダプタ定義（詳細）画面の設定項目　463
アプリケーション統合属性ファイル　201
アプリケーション統合属性ファイルの編集（Message Queue アダプタの場合）　270

う

受付およびサービスアダプタの開発の概要　11

お

応答電文の形式〔DB アダプタ〕　211
応答電文フォーマット（ヘッダ変数用）（HTTP 受付用）　56
応答電文フォーマット（ボディ変数用）（HTTP 受付用）　60

か

カスタムアダプタ開発フレームワーク　555
カスタムアダプタ開発フレームワーク動作定義ファイル〔汎用カスタムアダプタ開発時の定義ファイル〕　576
カスタムアダプタ開発フレームワーク動作定義ファイル〔汎用カスタムアダプタの定義〕　440
カスタムアダプタ開発フレームワークの API　555
カスタムアダプタ開発フレームワークの定義ファイル　576
カスタムアダプタ定義ファイル〔汎用カスタムアダプタ開発時の定義ファイル〕　583
カスタムアダプタプロパティファイル〔汎用カスタムアダプタ開発時の定義ファイル〕　577
カスタムアダプタプロパティファイル〔汎用カスタムアダプタの定義〕　440
カスタムアダプタ用アプリケーション統合属性ファイル〔カスタムアダプタの定義〕　440
カスタムアダプタ用アプリケーション統合属性ファイル〔汎用カスタムアダプタ開発時の定義ファイル〕　577
カスタム受付　502
カスタム受付でのエラーの伝わり方　538
カスタム受付の運用　529
カスタム受付の開発　505
カスタム受付の概要　502
カスタム受付のチューニング　536
カスタム受付の定義　526
カスタム受付フレームワークの API　511

き

キーストアに関する設定　758

く

クライアント定義ファイルの設定（SessionBean） 171

クライアント定義ファイルの設定（Web サービス） 165

け

検証内容の表示〔サービスアダプタ〕 499

検証内容〔ユーザ定義受付〕 139

検証方法〔サービスアダプタ〕 499

検証方法〔ユーザ定義受付〕 140

こ

コネクション定義識別子 272

コマンドアダプタの定義ファイルの作成および編集 409

コマンドアダプタを新規に追加する 158

コマンドアダプタを定義する 404

さ

サービスアダプタ定義（基本）画面 594

サービスアダプタ定義（詳細）画面 599

サービスアダプタ定義画面での設定（DB アダプタの場合） 459

サービスアダプタ定義画面での設定（FTP アダプタの場合） 471

サービスアダプタ定義画面での設定（gRPC アダプタの場合） 494

サービスアダプタ定義画面での設定（HTTP アダプタの場合） 488

サービスアダプタ定義画面での設定（Kafka アダプタの場合） 492

サービスアダプタ定義画面での設定（MDB（DB キュー）アダプタの場合） 457

サービスアダプタ定義画面での設定（MDB（WS-R）アダプタの場合） 454

サービスアダプタ定義画面での設定（SessionBean アダプタの場合） 452

サービスアダプタ定義画面での設定（SFTP アダプタの場合） 491

サービスアダプタ定義画面での設定（SOAP アダプタの場合） 450

サービスアダプタ定義画面での設定（TP1 アダプタの場合） 460

サービスアダプタ定義画面での設定（コマンドアダプタの場合） 490

サービスアダプタ定義画面での設定（汎用カスタムアダプタの場合） 494

サービスアダプタ定義画面での設定（ファイルアダプタの場合） 463

サービスアダプタ定義画面での設定（ファイル操作アダプタの場合） 474

サービスアダプタ定義画面での設定（メールアダプタの場合） 487

サービスアダプタ定義画面での操作（FTP アダプタの場合） 336

サービスアダプタ定義画面での操作（gRPC アダプタの場合） 434

サービスアダプタ定義画面での操作（HTTP アダプタの場合） 399

サービスアダプタ定義画面での操作（Kafka アダプタの場合） 427

サービスアダプタ定義画面での操作（Message Queue アダプタの場合） 267

サービスアダプタ定義画面での操作（Object Access アダプタの場合） 252

サービスアダプタ定義画面での操作（SFTP アダプタの場合） 420

サービスアダプタ定義画面での操作（TP1 アダプタの場合） 216

サービスアダプタ定義画面での操作（コマンドアダプタの場合） 408

サービスアダプタ定義画面での操作（ファイルアダプタの場合） 240

サービスアダプタ定義画面での操作（ファイル操作アダプタの場合） 353

サービスアダプタ定義画面での操作（メールアダプタの場合） 368

サービスアダプタ定義画面での操作〔DB アダプタ〕 199

サービスアダプタ定義画面での操作〔MDB（DB キュー）〕 177

サービスアダプタ定義画面での操作〔MDB（WS-R)〕 174
サービスアダプタ定義画面での操作〔SessionBean〕 169
サービスアダプタ定義画面での操作〔Web サービス〕 163
サービスアダプタ定義画面での操作〔汎用カスタムアダプタ〕 448
サービスアダプタ定義での設定一覧 450
サービスアダプタの検証 499
サービスアダプタの削除 500
サービスアダプタの接続先情報を動的に変更する例 130
サービスアダプタの追加 149
サービスアダプタの定義 144
サービスアダプタの内容の定義 162
サービスアダプタの編集 498
サービスアダプタの保存 497
サービスリクエスタの作成（DB アダプタの場合） 205
サンプルについて 627
サンプルプログラム〔カスタムアダプタ開発フレームワーク〕 584

し

システム全体の中の受付およびサービスアダプタの位置づけ 12
障害情報の取得（カスタム受付） 543

す

スケジューラダイレクト機能使用時の TP1/Client/J 環境定義 222
スケジューラダイレクト機能を使用する場合 221
スケジュール駆動受付の定義 93
スケジュール駆動受付の定義の流れ 93
スケジュール駆動受付の定義ファイルを作成する 95
スケジュール駆動受付の電文フォーマットを作成する 94
スケジュール駆動受付を追加する 95

せ

セキュアプロトコルによる暗号化・認証 759
セットアップ時の属性ファイルの定義項目 760

つ

通信先を指定した接続をする場合 222

て

定義済みサービスアダプタを利用してサービスアダプタを追加する 161
定義の流れ〔サービスアダプタ〕 145
データ変換の定義〔TP1 アダプタ〕 216
データ変換の定義〔ファイルアダプタ〕 240
電文フォーマット定義ファイルの作成（Message Queue アダプタの場合） 255
電文フォーマットの作成（DB アダプタ電文の場合） 196
電文フォーマットの作成（FTP アダプタの場合） 273
電文フォーマットの作成（gRPC アダプタの場合） 427
電文フォーマットの作成（HTTP アダプタの場合） 378
電文フォーマットの作成（Kafka アダプタの場合） 421
電文フォーマットの作成（Object Access アダプタの場合） 242
電文フォーマットの作成（SFTP アダプタの場合） 411
電文フォーマットの作成（TP1 アダプタの場合） 216
電文フォーマットの作成（コマンドアダプタの場合） 404
電文フォーマットの作成（ファイルアダプタの場合） 223
電文フォーマットの作成（ファイル操作アダプタの場合） 336
電文フォーマットの作成（メールアダプタの場合） 357
電文フォーマットの種類（ファイルアダプタの場合） 227
電文例 236

と

トラブルシューティング 601

ね

ネームサービス機能使用時の TP1/Client/J 環境定義 221
ネームサービス機能を使用する場合 221

は

バイナリフォーマット定義ファイルの作成 232
汎用カスタムアダプタの開始に関するトラブルシューティング 602
汎用カスタムアダプタの定義画面 594
汎用カスタムアダプタの配備に関するトラブルシューティング 601
汎用カスタムアダプタを使用する際の環境設定 601
汎用カスタムアダプタを新規に追加する 161
汎用カスタムアダプタを定義する 438

ひ

ビジネスプロセスの設定（gRPC アダプタの場合）435
ビジネスプロセスの設定（HTTP アダプタの場合） 402
ビジネスプロセスの定義後に SOAP 受付を定義する26
ビジネスプロセスを定義する前に SOAP 受付を定義する 22

ふ

ファイルアダプタ実行環境プロパティファイルの作成 242
ファイルアダプタの設定例 627
ファイルアダプタを新規に追加する 153
ファイルアダプタを定義する 223
ファイルイベント受付の概要 20
ファイルイベント受付の定義 79
ファイルイベント受付の定義の流れ 79
ファイルイベント受付の定義ファイルを作成する 82
ファイルイベント受付の電文フォーマットを作成する 80
ファイルイベント受付を追加する 82
ファイル書き込み時の要求電文（XML 形式） 236
ファイル書き込み時の要求電文（バイナリ形式） 239
ファイル操作アダプタの応答電文フォーマット 347
ファイル操作アダプタの定義ファイルの作成および編集 354
ファイル操作アダプタの定義例 689
ファイル操作アダプタの要求電文フォーマット 336
ファイル操作アダプタを新規に追加する 156
ファイル操作アダプタを定義する 336
ファイル変換オペレーション使用時の注意事項 355
ファイル読み込み時の要求電文 236
フォルトが発生した場合にフォルト電文を返す例 111
プロトコルコンバータ JAR ファイルの作成 444
プロトコルコンバータインターフェース〔カスタムアダプタ開発フレームワークの API〕 555
プロトコルコンバータからの操作に関するトラブルシューティング 604
プロトコルコンバータ用独自定義ファイル 439
分散開発時のエクスポート・インポート 626

め

メールアダプタで必要なセキュリティの設定 765
メールアダプタの定義例 369
メールアダプタを使用するときの注意事項 357
メールアダプタを新規に追加する 157
メールアダプタを定義する 357
メッセージインターフェース〔カスタムアダプタ開発フレームワークの API〕 562

ゆ

ユーザ定義受付 18
ユーザ定義受付（呼出先選択）の定義 770
ユーザ定義受付の検証 139
ユーザ定義受付の削除 142
ユーザ定義受付の情報の変更 141
ユーザ定義受付の定義 17
ユーザ定義受付の内容の確認 137
ユーザ定義受付の保存 138
ユーザ定義ファイルの作成（Object Access アダプタの場合） 251

よ

要求電文 list-option 要素の指定値について 356

要求電文の形式〔DB アダプタ〕　205
要求電文フォーマット（ヘッダ変数用）（HTTP 受付用）　54
要求電文フォーマット（ボディ変数用）（HTTP 受付用）　56
用語解説　774

ら

ライブラリ JAR ファイルの作成　445

り

リストコマンドオプション定義ファイル　48
リモート API 機能使用時の TP1/Client/J 環境定義　220
リモート API 機能を使用する場合　220
利用できる受付　14
利用できるサービスアダプタ　15

れ

例外クラス〔カスタムアダプタ開発フレームワークの API〕　575