

# サービスプラットフォーム パラメタ設定ガイド

---

# はじめに

本書は、サービスプラットフォームのシステム構築・設計において、これまでサービスプラットフォームが適用されたシステム事例から検討すべきパラメタ定義を抽出して解説しています。本書により、多くのお客様で利用される代表的なシステム構成でのパラメタ設計のポイントを理解することができます。なお、すべてのパラメタ設計について網羅しているものではありませんので、お客様個別のタイムアウトや性能などの要件がある場合は、マニュアルと合わせて使用するドキュメントとしてお使いください。

## 1. 対象とする読者

サービスプラットフォームを利用したシステムを構築する立場にあるシステム構築者を対象としています。本書によって、サービスプラットフォームの多くのお客様で利用される代表的なシステムのシステム構成定義書を作成できます。

## 2. 対象とする製品

### uCosminexus Service Platform 09-70

※本書では、Linux を基準に表記しています。Windows の場合、パスの表記は、指定のない限り以下のように読み替えてください。

Linux の表記	/opt/Cosminexus
Windows の表記	<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>

※本書では、SOAP、HTTP、FTP 連携の場合のパラメタ設定について記述しています。

※本書では、Reliable Messaging を使用したシステムは対象外としています。

### 適用 OS

- Red Hat Enterprise Linux 5.1 (AMD/Intel 64)以降,
- Red Hat Enterprise Linux 6.1 (AMD/Intel 64)以降
- Red Hat Enterprise Linux 7.1 (AMD/Intel 64)以降
- Windows Server 2008 x86
- Windows Server 2008 x64
- Windows Server 2008 R2

- Windows Server 2012
- Windows Server 2012 R2

#### ■商標類

- HITACHI、Cosminexus、HiRDB、JP1、uCosminexus は、株式会社 日立製作所の商標または登録商標です。
- AMD は、Advanced Micro Devices, Inc.の商標です。
- Intel は、アメリカ合衆国およびその他の国における Intel Corporation の商標です。
- Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。
- Red Hat は、米国およびその他の国で Red Hat, Inc. の登録商標もしくは商標です。
- Microsoft は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- Oracle と Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。
- Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- 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。

#### ■マイクロソフト製品の表記について

本書では、マイクロソフト製品の名称を次のように表記しています。

表記		製品名	
Windows	Windows(x86)	Windows Server 2008 x86	Microsoft® Windows Server® 2008 Standard 32-bit 日本語版
			Microsoft® Windows Server® 2008 Enterprise 32-bit 日本語版
	Windows(x64)	Windows Server 2008 x64	Microsoft® Windows Server® 2008 Standard 日本語版
			Microsoft® Windows Server® 2008 Enterprise 日本語版
		Windows Server 2008 R2	Microsoft® Windows Server® 2008 R2 Standard 日本語版
			Microsoft® Windows Server® 2008 R2 Enterprise 日本語版
			Microsoft® Windows Server® 2008 R2 Datacenter 日本語版
		Windows Server 2012	Microsoft® Windows Server® 2012 Standard 日本語版
	Microsoft® Windows Server® 2012 Datacenter 日本語版		

表記		製品名
	Windows Server 2012 R2	Microsoft® Windows Server® 2012 R2 Standard 日本語版
		Microsoft® Windows Server® 2012 R2 Datacenter 日本語版

#### ■製品名の表記

本書では、マイクロソフト製品以外の製品名を次のように表記しています。

表記		製品名
Oracle	Oracle 11g	Oracle® Database 11g
		Oracle® Database 11g Release 2
	Oracle 12c	Oracle® Database 12c
JP1/FTS/FTP		JP1/File Transmission Server/FTP

#### ■Java 関連用語の表記

本書では、Java 関連用語を次のように表記しています。

表記	Java 関連用語
Java	Java™
JavaVM	Java™ Virtual Machine
JDBC	Java™ Database Connectivity
	JDBC™

#### ■発行元

株式会社日立製作所 情報・通信システム社 IT プラットフォーム事業本部

All Rights Reserved. Copyright (C) 2016, Hitachi, Ltd.

# パラメタ設定ガイド

## 目次

<b>1 概要</b>	<b>1</b>
1.1 パラメタ種別と定義ファイル一覧	2
<b>2 サービスプラットフォームを使用するための必須パラメタの設定</b>	<b>5</b>
2.1 サービスプラットフォームのプロセス構成と各プロセスの設定	6
2.1.1 Management Server の設定	7
2.1.2 運用管理エージェントの設定	9
2.1.3 アプリケーションサーバの設定	10
2.1.4 ポート番号の設定	12
2.2 HCSC サーバの構築に関連するパラメタ	15
2.2.1 ユーザ情報の設定	16
2.2.2 受付のセットアップの設定	18
2.2.3 XML 解析モードの設定	20
<b>3 システム構成依存パラメタの設定</b>	<b>21</b>
3.1 データベースに関連するパラメタ	22
3.1.1 データベースを使用する場合の設定	22
3.2 クラスタ構成に関連するパラメタ	27
3.2.1 ロードバランスクラスタ構成時の設定	27

3.2.2 HA クラスタ構成時の設定 .....	30
3.3 1 台のマシンに複数の環境を構築する場合に関連するパラメタ .....	32
3.3.1 1 台のマシンに複数の環境を構築する流れ .....	33
3.3.2 CSCMNG_HOME 環境変数の設定 .....	34
3.3.3 設定に注意が必要なパラメタ .....	35
<b>4 受付・アダプタ単位のパラメタの設定</b> .....	<b>39</b>
4.1 作業フォルダまたは共通フォルダを利用する場合の設定 .....	40
4.2 FTP 連携で使用する受付・アダプタの設定 .....	41
4.2.1 コンポーネント構成 .....	41
4.2.2 ネットワーク設計 .....	44
4.2.3 ユーザ設計 .....	48
4.2.4 セキュリティ設計 .....	49
<b>5 性能改善パラメタの設定</b> .....	<b>53</b>
5.1 ログレベルの設定 .....	54
5.2 XML 処理性能の改善 .....	56
5.3 ファイルディスクリプタの設定 (※Linux 限定) .....	59
5.3.1 FTP 受付・FTP アダプタを使用する場合 .....	59
<b>6 チューニングパラメタの設定</b> .....	<b>60</b>
6.1 JavaVM メモリ .....	61
6.2 流量制御 .....	63
6.2.1 SOAP、HTTP 連携における流量制御 .....	64

6.2.2	ビジネスプロセス単位の流量制御.....	70
6.2.3	FTP 連携におけるの流量制御.....	74
6.3	タイムアウト.....	77
6.3.1	HCSC サーバ内のトランザクションタイムアウト .....	78
6.3.2	SOAP、HTTP 連携におけるタイムアウト .....	80
6.3.3	FTP 連携におけるタイムアウト .....	83
6.4	リトライ .....	95
6.4.1	FTP 連携におけるリトライ .....	95

# 1 概要

この章では、サービスプラットフォームのパラメタ種別と定義ファイル一覧を示します。

## 本章の構成

### 1.1 パラメタ種別と定義ファイル一覧



# 1.1 パラメタ種別と定義ファイル一覧

本書では、以下の 5 種類のパラメタについて説明します。

- サービスプラットフォームを使用するための必須パラメタ …2 章
- システム構成依存パラメタ …3 章
- 受付・アダプタ単位のパラメタ …4 章
- 性能改善パラメタ …5 章
- チューニングパラメタ …6 章

本書で使用する定義ファイルの一覧を示します。(各種受付、アダプタの定義ファイルは各設定箇所参照)

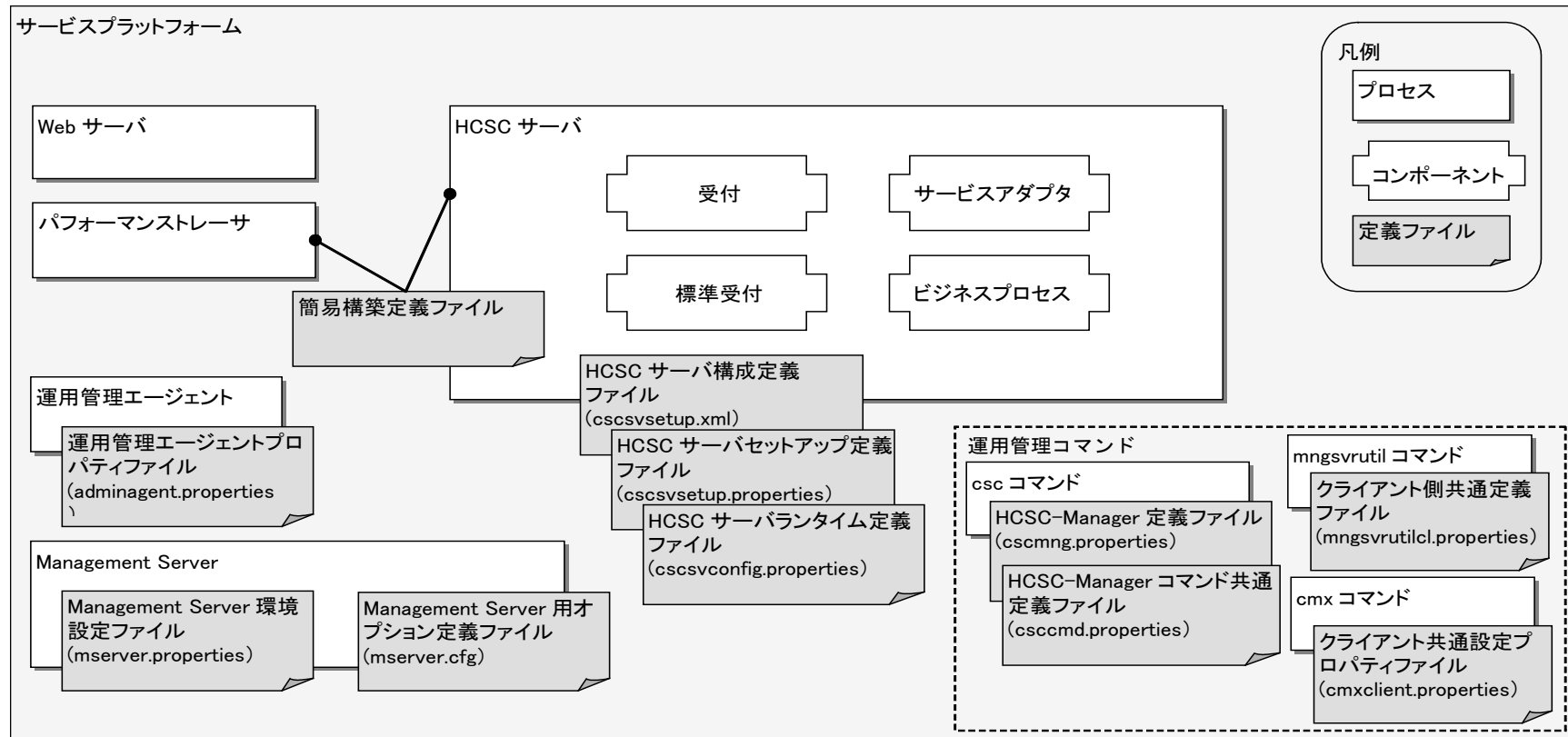


図 1.1-1 定義ファイル一覧

表 1.1-1 定義ファイル一覧

#	定義ファイル名	定義利用 コンポーネント	説明	格納先
1	運用管理エージェントプロパ ティファイル (adminagent.properties)	運用管理エージェント	運用管理エージェント本体に関する設定 をします。	/opt/Cosminexus/manager/config/adminagent.properties
2	Management Server 環境設 定ファイル (mserver.properties)	ManagementServer	Management Server が使用するポート 番号の設定や、障害検知時コマンドの動 作の設定をします。	/opt/Cosminexus/manager/config/mserver.properties
3	Management Server 用オプ ション定義ファイル (mserver.cfg)		Management Server を実行する JavaVM の起動オプションを指定しま す。	/opt/Cosminexus/manager/config/mserver.cfg
4	簡易構築定義ファイル	Web システム	簡易構築定義ファイルには、Web システ ムの属性、負荷分散機、物理ティア、サ ービスユニット、およびサービスユニッ トを構成するホストを定義します	任意 (次のテンプレートファイルをコピーして使用する。 /opt/Cosminexus/manager/config/templates/cmxdcombinedmodel.xml)
5	HCSC サーバ構成定義ファイ ル (cscsvsetup.xml)	HCSC サーバ	HCSC サーバの構成情報を設定します。	任意 (次のテンプレートファイルをコピーして使用する。 /opt/Cosminexus/CSC/config/manager/templates/cscsvsetup.xml) (cscsvsetup コマンドの-propfile オプションの引数に指定します。)
6	HCSC サーバセットアップ定 義ファイル (cscsvsetup.properties)		HCSC サーバのセットアップ時に設定で きる内容を設定します。	任意 (次のテンプレートファイルをコピーして使用する。 /opt/Cosminexus/CSC/config/manager/templates/cscsvsetup.properties) (#5 HCSC サーバ構成定義ファイルから参照される。)
7	HCSC サーバランタイム定義 ファイル		セットアップ後の HCSC サーバの定義 情報を設定します。	任意 (次のテンプレートファイルをコピーして使用する。

	(cscsvconfig.properties)			/opt/Cosminexus/CSC/config/manager/templates/cscsvconfig.properties)  (cscsvconfig コマンドの-propfile オプションの引数に指定します。)
8	HCSC-Manager 定義ファイル (cscmng.properties)	csc コマンド	HCSC-Manager の動作に必要となる情報を設定します。	/opt/Cosminexus/CSC/config/manager/cscmng.properties
9	HCSC-Manager コマンド共通定義ファイル (csccmd.properties)	(HCSC-Manager)	HCSC-Manager コマンドのオプションのデフォルト値を設定します。	/opt/Cosminexus/CSC/config/manager/csccmd.properties
10	mngsvrutil コマンドのクライアント側共通定義ファイル (mngsvrutilcl.properties)	mngsvrutil コマンド	mngsvrutil コマンドのオプションのデフォルト値を共通定義として設定できません。	/opt/Cosminexus/manager/config/mngsvrutilcl.properties
11	クライアント共通設定プロパティファイル (cmxclient.properties)	cmx コマンド	Smart Composer 機能で提供するコマンドの共通引数のデフォルト値を設定します。	/opt/Cosminexus/manager/config/cmxclient.properties

## 2 サービスプラットフォームを使用するための必須パラメタの設定

この章では、サービスプラットフォームを使用するための必須パラメタの設定について説明します。

### 本章の構成

- 2.1 サービスプラットフォームのプロセス構成と各プロセスの設定
- 2.2 HCSC サーバの構築に関連するパラメタ

## 2.1 サービスプラットフォームのプロセス構成と各プロセスの設定

サービスプラットフォームを使用するために最低限必要なプロセス構成を示します。

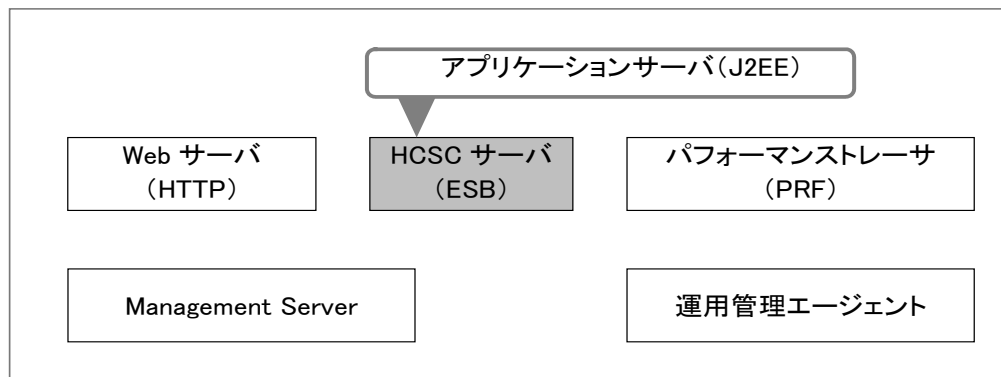


図 2.1-1 サービスプラットフォームのプロセス構成図

表 2.1-1 サービスプラットフォームのプロセス構成要素

#	構成要素	役割
1	HCSC サーバ(ESB)	ESB の実行プロセスで、実体はアプリケーションサーバ(J2EE)のプロセス
2	Web サーバ(HTTP)	サービスリクエストからの SOAP または HTTP 要求を HCSC サーバへ転送 (SOAP、HTTP 連携の場合のみ)
3	パフォーマンストレーサ(PRF)	HCSC サーバ内の処理のポイントで取得する PRF トレースを収集
4	運用管理エージェント	Web サーバ、HCSC サーバ、パフォーマンストレーサの運用管理
5	Management Server	

次から、サービスプラットフォームを使用するための各プロセスの必須パラメタについて説明します。

## 2.1.1 Management Server の設定

Management Server では、下記の 2 ファイルに定義項目を追加します。

### (1) Management Server 環境設定ファイル(mserver.properties)

表 2.1-2 パラメーター一覧

#	定義項目	説明	デフォルト値
1	<code>com.cosminexus.mngsvr.management.enabled=true</code>	Management Server リモート管理機能を有効にするかどうかを指定します。 サービスプラットフォームでは、「true」を設定します。	false
2	<code>com.cosminexus.mngsvr.management.connector.enabled=true</code>	Management Server リモート管理機能への外部接続を有効にするかどうかを指定します。 サービスプラットフォームでは、「true」を設定します。	false
3	<code>ejbserver.server.edition.settingforce=</code>	HCSC-Manager から Management Server を利用するかどうかを指定します。 サービスプラットフォームでは、「指定なし」を設定します。 指定なしとはキーだけを指定し、値を指定しないことです。 (指定方法) <code>ejbserver.server.edition.settingforce=</code>	Smart
4	<code>mngsvr.myhost.name=localhost</code>	論理サーバを配置するホスト名に、論理サーバを配置するホストからアクセス可能な IP アドレス、またはその IP アドレスを示すホスト名を指定します。 サービスプラットフォームでは、Management Server と各論理サーバを同一ホストに置くことを基本としているため、「localhost」を指定します。	なし

注：#3 は、値がないですが省略できないので、このとおりに設定してください。

(2) Management Server 用オプション定義ファイル(msserver.cfg)

表 2.1-3 パラメター一覧

#	定義項目	説明
1	add.jvm.arg=-XX:MetaspaceSize=128m	Management Server 起動時の Metaspace 領域の初期サイズを指定します。128MB 以上を指定してください。
2	add.jvm.arg=-XX:MaxMetaspaceSize=128m	Management Server 起動時の Metaspace 領域の最大サイズを指定します。128MB 以上を指定してください。

## 2.1.2 運用管理エージェントの設定

運用管理エージェントプロパティファイル(adminagent.properties)に以下の設定を追加します。

表 2.1-4 パラメータ一覧

#	定義項目	説明
1	adminagent.j2ee.watch.level=2	J2EE サーバの動作確認レベルを指定します。 サービスプラットフォームでは、「2 (プロセスの存在確認および J2EE サーバへのアクセスで J2EE サーバの動作を確認)」を指定してください。 「1 (プロセスの存在確認で論理サーバの動作を確認)」を指定すると、J2EE サーバの起動に失敗する場合があります。
2	adminagent.forcestop.threaddump=true	J2EE サーバの強制停止時にスレッドダンプを出力するかどうかを指定します。 トラブルシュート資料確保のため、「true」を設定してください。
3	adminagent.forcestop.threaddump.time out=45(初期値)	スレッドダンプが出力されるのに十分な時間を設定してください。スレッドダンプの出力が終わる時間よりも短い時間が設定されていると、スレッドダンプの出力内容が不完全になることがあります。 また、ワトソン博士を使用している場合、「メッセージボックスによる通知」を設定しないでください。ワトソン博士の詳細は Microsoft 社のホームページなどを参照してください。



## 2.1.3 アプリケーションサーバの設定

### (1) 簡易構築定義ファイル

構築で使用する簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の<configuration>タグ内には以下の設定を追加して下さい。

表 2.1-5 パラメータ一覧

#	定義項目 (設定値)	説明	省略値
1	<pre>&lt;param&gt; &lt;param-name&gt;ejbserver.rmi.localinvocation.scope&lt;/param-name&gt; &lt;param-value&gt;app&lt;/param-value&gt; &lt;/param&gt;</pre>	ローカル呼び出し最適化機能の適用範囲を指定します。 サービスプラットフォームでは、「app」を指定します。	app
2	<pre>&lt;param&gt; &lt;param-name&gt;ejbserver.server.j2ee.feature&lt;/param-name&gt; &lt;param-value&gt;1.4&lt;/param-value&gt; &lt;/param&gt;</pre>	Component Container の動作モードを指定します。 サービスプラットフォームでは、「1.4」を指定します。	1.4
3	<pre>&lt;param&gt; &lt;param-name&gt;ejbserver.jndi.global.enabled&lt;/param-name&gt; &lt;param-value&gt;false&lt;/param-value&gt; &lt;/param&gt;</pre>	アプリケーション開始時に、ネーミングサービスに対して Portable Global JNDI 名でオブジェクトを登録するかどうかを 指定します。 サービスプラットフォームでは、「false」を指定します。	true
4	<pre>&lt;param&gt; &lt;param-name&gt;ejbserver.DynamicStubLoading.Enabled&lt;/param-n ame&gt; &lt;param-value&gt;false&lt;/param-value&gt; &lt;/param&gt;</pre>	ダイナミッククラスローディングを使用するかどうかを指定し ます。 サービスプラットフォームでは、「false」を指定します。	false
5	<pre>&lt;param&gt; &lt;param-name&gt;add.jvm.arg&lt;/param-name&gt; &lt;param-value&gt;-XX:-HitachiUseExplicitMemory&lt;/param-value&gt; &lt;/param&gt;</pre>	明示管理ヒープ機能の有効、無効を指定します。 サービスプラットフォームでは、明示管理ヒープ機能は無効[-] に指定します。	有効[+]

## (2) コンテナ拡張ライブラリの設定

コンテナ拡張ライブラリとして、以下の設定を簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の<configuration>タグ内に設定してください。定義項目は、使用する SOAP モードに合わせて選択してください。両方の設定を追加することはできません。

表 2.1-6 パラメータ一覧

#	SOAP モード	定義項目
1	SOAP1.1 モード	<param-name>add.class.path</param-name> <param-value>/opt/Cosminexus/c4web/lib/hitsaaj.jar</param-value>
2	SOAP1.1/1.2 併用モード	<param-name>add.class.path</param-name> <param-value>/opt/Cosminexus/jaxws/lib/cjjaxws.jar</param-value>

## 2.1.4 ポート番号の設定

サービスプラットフォームで使用する、ポート番号の接続構成は下記のとおりです。必要に応じてデフォルト値から値を変更して設定してください。  
Web サーバは、SOAP、HTTP 連携の場合のみ使用します。

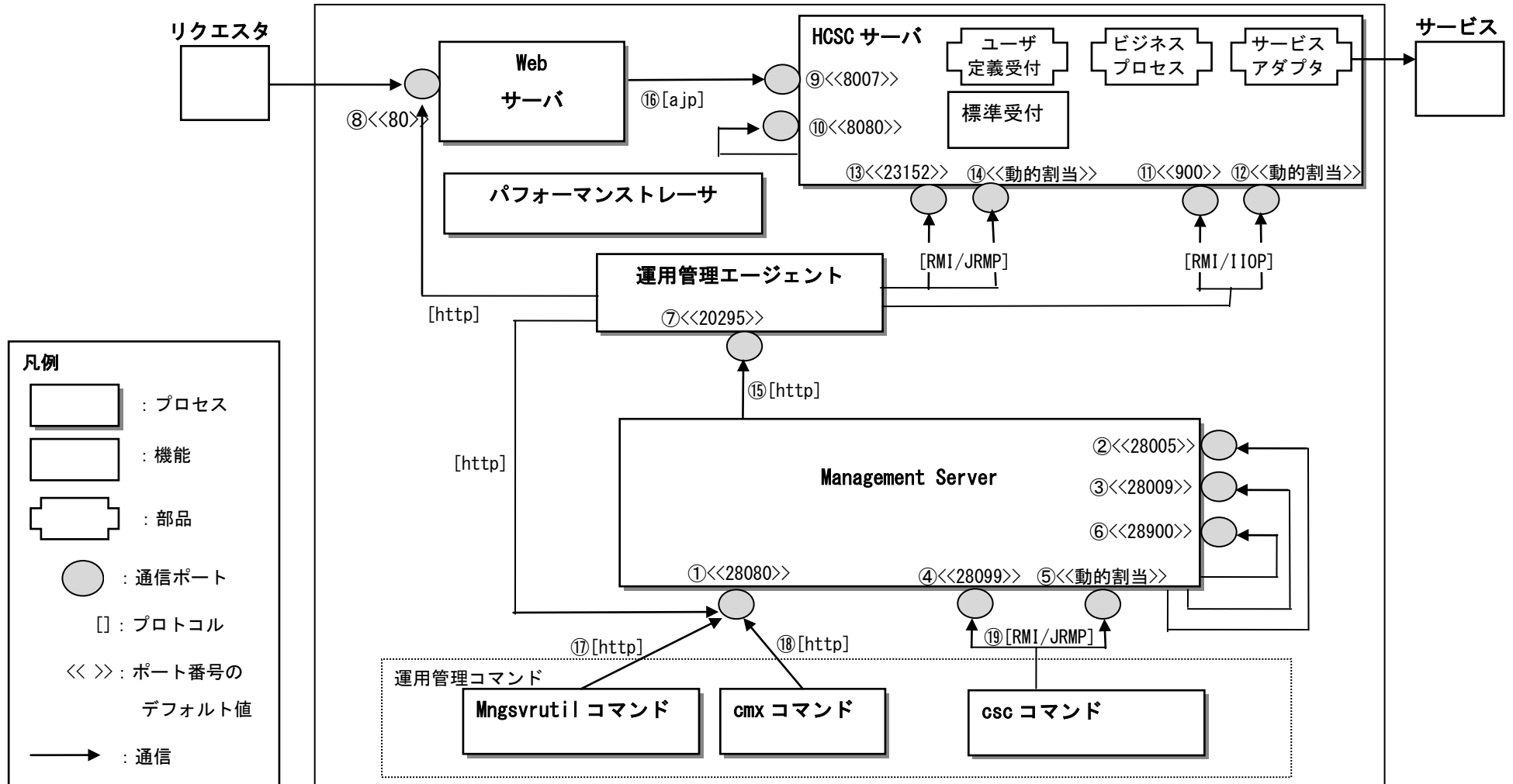


図 2.1-2 ポート番号一覧

表 2.1-7 ポート番号に関するパラメタの説明

#	定義ファイル	定義項目(初期値)	説明
①	Management Server環境設定 ファイル	webservice.connector.http.port=28080	運用管理コマンド(mngsvrutil コマンド、cmx コマンド)からの要求を受け付ける、Management Server プロセスのポート
②		webservice.shutdown.port=28005	内部通信用の Management Server プロセスのポート
③		webservice.connector.ajp13.port=28009	内部通信用の Management Server プロセスのポート
④		com.cosminexus.mngsvr.management.port=28099	運用管理コマンド(csc コマンド)からの要求を受け付ける、Management Server プロセスのポート
⑤		com.cosminexus.mngsvr.management.listen.port=0	運用管理コマンド(csc コマンド)からの要求を受け付ける、Management Server プロセスのポート。 初期値(0)は自動割当を表します。
⑥		ejbserver.naming.port=28900	内部通信用の Management Server プロセスのポート
⑦	運用管理エージェントプロパ ティファイル	adminagent.adapter.port=20295	Management Server からの要求を受付ける、運用管理エ ージェントプロセスのポート
⑧	簡易構築定義ファイル	<param-name>Listen</param-name> <param-value>80</param-value>※1	SOAP/HTTP クライアントからの要求を受け付ける、 Web サーバプロセスのポート
⑨		<param-name>webservice.connector.ajp13.port </param-name> <param-value>8007</param-value>※1	Web サーバからの要求を受け付ける、HCSC サーバプロ セスのポート
⑩		<param-name>ejbserver.http.port</param-name> <param-value>8080</param-value>	内部通信用の HCSC サーバプロセスのポート
⑪		<param-name>ejbserver.naming.port</param-name> <param-value>900</param-value>	運用管理エージェントからの要求を受け付ける、HCSC サーバプロセスのポート
⑫		<param-name>vbroker.se.iioptp.scmiioptp.listener.port</param- name> <param-value>0</param-value>	運用管理エージェントからの要求を受け付ける、HCSC サーバプロセスのポート 初期値(0)は自動割当を表します。

⑬		<param-name>ejbserver.rmi.naming.port</param-name> <param-value>23152</param-value>	運用管理エージェントからの要求を受け付ける、HCSC サーバプロセスのポート
⑭		<param-name>ejbserver.rmi.remote.listener.port</param-name> <param-value>0</param-value>	運用管理エージェントからの要求を受け付ける、HCSC サーバプロセスのポート 初期値(0)は自動割当を表します。
⑮		<host> <agent-port>20295</agent-port> </host>	⑦と一致させてください。 Management Server から運用管理エージェントへの通 信の接続先ポート
⑯		<param-name>worker.<ワーカー名>.port</param-name> <param-value>8007</param-value>※1	⑨と一致させてください。 Web サーバから HCSC サーバへの通信の接続先ポート
⑰	mngsvrutil コマンドのク ライアント側共通定義ファ イ ル (mngsvrutilcl.properties)	mngsvrutil.connect.host=localhost:28080	①と一致させてください。 運用管理コマンド(mngsvrutil コマンド)から Management Server への通信の接続先ポート (設定は任意です。)
⑱	クライアント共通設定プロ パ テ ィ ファ イ ル (cmxclient.properties)	cmx.connect.host=localhost:28080	①と一致させてください。 運用管理コマンド(cmx コマンド)から Management Server への通信の接続先ポート (設定は任意です。)
⑲	HCSC サーバ構成定義ファ イ ル	<cosminexus-manager> <port>28099</port> </cosminexus-manager>	④と一致させてください。 運用管理コマンド(csc コマンド)から Management Server への通信の接続先ポート

注 ※1 SOAP、HTTP 連携で Web サーバを使用する場合のみ設定します。

## 2.2 HCSC サーバの構築に関連するパラメタ

---

この章では、HCSC サーバの構築で使用する下記の 3 つのファイルにおいて、注意すべきパラメタについて説明します。

- (1) HCSC サーバ構成定義ファイル
- (2) HCSC サーバセットアップ定義ファイル
- (3) HCSC サーバランタイム定義ファイル

なお、クラスタ構成を組む場合は、3.2 クラスタ構成に関連するパラメタの内容も参照してください。



補足 データベースを使用しないシステムを構築する場合

(2) HCSC サーバセットアップ定義ファイルは、データベースに関連するパラメタが多いため、添付の「システム環境定義書」作成時には、「フィルタ DB なし」を「○」でフィルタリングしてパラメタを設定することを推奨します。

## 2.2.1 ユーザ情報の設定

Management Server、および HCSC サーバが使用するデータベースの管理ユーザ情報を設定します。

<前提> Management Server、データベースおよび HCSC-Manager のユーザ情報は設計・登録されていること。

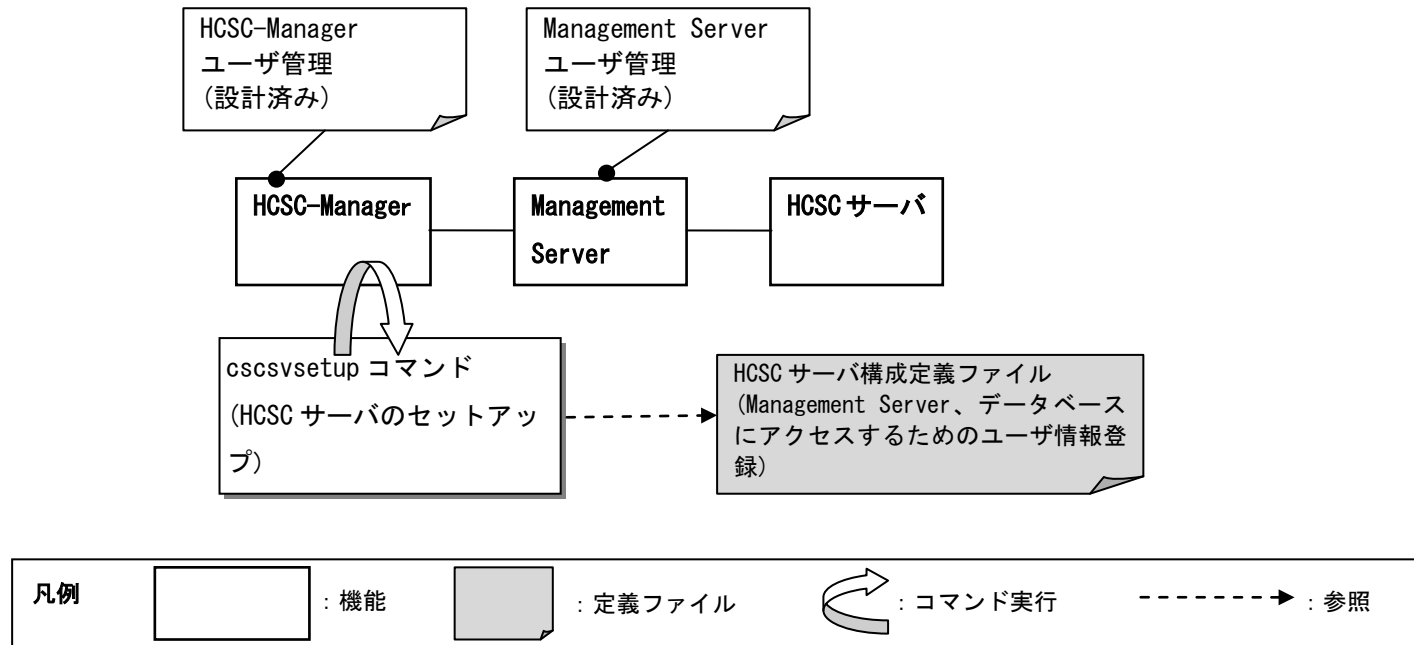


図 2.2-1 ユーザ情報の設定箇所

表 2.2-1 ユーザ情報の設定パラメータ一覧

#	設定項目	機能	設定方法	定義項目
1	Management Server 管理ユーザ ID	Management Server	HCSC サーバ 構成定義ファ イル	<cosminexus-manager> <id>Management Server の管理ユーザ ID</id> </cosminexus-manager>
2	Management Server 管理ユーザパ スワード			<cosminexus-manager> <password>Management Server の管理ユーザパスワード</password > </cosminexus-manager>
3	HCSC-Manager にアクセスするユ ーザ ID/パスワード	HCSC-Manager	cscpasswd コマンド	cscpasswd [-user <ログインユーザ ID>]※ [-pass <ログインパスワード>]※ -cscmnguser <変更後のユーザ ID> -cscmngpass <変更後のパスワード>

注※ HCSC-Manager インストール直後は、ユーザ ID : admin、パスワード : adamin で仮定されます。

上記で設定した Management Server の管理ユーザ ID とパスワードは、cmx コマンドまたは mngsvrutil コマンド実行時のオプションに必要になります。また、HCSC-Manager のユーザ ID とパスワードは csc コマンド実行時のオプションに必要になります。これらのオプションは、それぞれ下記のファイルにデフォルトとして登録することで、コマンド実行時に省略できます。(設定は任意)

表 2.2-2 コマンドのユーザ情報オプションデフォルト値設定パラメータ一覧

#	機能	設定項目	定義ファイル	定義項目
1	cmx コマンド	Management Server 管理ユーザ ID	クライアント共 通設定プロパテ ィファイル	cmx.connect.userid=(デフォルト値なし)
2		Management Server 管理ユーザパスワード		cmx.connect.passwd=(デフォルト値なし)
3	mngsvrutil コマン ド	Management Server 管理ユーザ ID	mngsvrutil コマ ンドのクライア ント側共通定義 ファイル	mngsvrutil.connect.userid=(デフォルト 値なし)
4		Management Server 管理ユーザパスワード		mngsvrutil.connect.password=(デフ ォルト値なし)
5	csc コマンド	HCSC-Manager のユーザ ID	HCSC-Manager	csc.all.user=(デフォルト値なし)



6	(HCSC-Manager)	HCSC-Manager のパスワード	コマンド共通定義ファイル	csc.all.pass=(デフォルト値なし)
---	----------------	---------------------	--------------	-------------------------

## 2.2.2 受付のセットアップの設定

受付のセットアップに関連するパラメタについて説明します。

セットアップする受付に合わせて、下記のパラメタを設定します。

表 2.2-3 セットアップする受付とパラメタの関係

#	セットアップする受付		HCSC サーバセットアップ定義ファイルの設定パラメタ	HCSC サーバ構成定義ファイルの設定パラメタ
1	標準の同期受付 (Web サービス)		request-soap=ON	<soap-reception> <ip-address></ip-address> <port></port> <protocol></protocol> </soap-reception>
2	標準の同期受付 (SessionBean)		request-ejb=ON	<ejb-reception> <ip-address></ip-address> <port></port> </ejb-reception>
3	ユーザ定義 受付	SOAP 受付	request-soap=ON※ request-userdef-soap=ON	<soap-reception> <ip-address></ip-address> <port></port> <protocol></protocol> </soap-reception>
4		SOAP 受付 以外	request-ejb=ON※ request-userdef-soap=OFF	<ejb-reception> <ip-address></ip-address> <port></port>

				</ejb-reception>
--	--	--	--	------------------

注※ユーザ定義受付のみを使用する場合でも、少なくとも一つの標準受付をセットアップする必要があります。

SOAP 受付以外のユーザ定義受付の場合は、標準の同期受付(SessionBean)を設定することを推奨します。

## 2.2.3 XML 解析モードの設定

XML の解析モードを選択します。

表 2.2-4 XML 解析モードの説明

#	解析モード	説明
1	standard	XML 全体を解析します。XML 解析処理で 09-00 以前との互換性を重視する場合は、「standard」を設定してください。
2	advanced	<p>次に示す処理を実行する場合に、メモリ使用量や処理性能が改善します。</p> <ul style="list-style-type: none"> <li>・ SOAP1.1/1.2 併用モードの SOAP 受付を使用する場合</li> <li>・ SOAP1.1/1.2 併用モードの SOAP アダプタを使用する場合</li> <li>・ DB アダプタを使用する場合</li> <li>・ データ変換を実行する場合</li> </ul> <p>&lt;注意事項&gt;</p> <ul style="list-style-type: none"> <li>・ XML 文章全体を解析しないため、不正な XML であってもエラーが発生しない場合があります。</li> <li>・ 次に示すコンポーネントは、XML 電文の解析に StAX を使用します。 <ul style="list-style-type: none"> <li>SOAP 受付 (SOAP1.1/1.2 併用モードの場合)</li> <li>SOAP アダプタ (SOAP1.1/1.2 併用モードの場合)</li> <li>DB アダプタ</li> <li>ビジネスプロセス (ヘッダ割当て処理)</li> </ul> </li> </ul> <p>そのため、次の点に注意してリクエストやサービスを作成してください。</p> <ul style="list-style-type: none"> <li>・ 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。</li> <li>・ 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。</li> <li>・ XML 文書に「:」で始まる要素名が含まれる場合、例外は発生しません。</li> <li>・ 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 ("]]&gt;") が単独で含まれる場合、例外は発生しません。</li> <li>・ 空要素タグは次のように、開始タグと終了タグで出力されます。 <p>(例) &lt;sample&gt;&lt;/sample&gt;</p> </li> </ul>

表 2.2-5 XML 解析モードの設定パラメタ

#	パラメタ	定義ファイル	デフォルト値	定義範囲
1	xmllanalyze-mode	HCSC サーバセットアップ定義ファイル	standard	standard   advanced

# 3 システム構成依存パラメタの設定

この章では、構築するシステムの構成によって設定が必要なパラメタについて説明します。

## 本章の構成

- 3.1 データベースに関連するパラメタ
- 3.2 クラスタ構成に関連するパラメタ
- 3.3 1台のマシンに複数の環境を構築する場合に関連するパラメタ

## 3.1 データベースに関連するパラメタ

データベースを使用する場合のパラメタについて説明します。

### 3.1.1 データベースを使用する場合の設定

データベースを使用する場合の設定パラメタについて次に示します。

#### (1) Management Server 用オプション定義ファイルの設定(msserver.cfg)

設定するパラメタを次に示します。使用するデータベースとデータベースに対応する JDBC ドライバに合わせて設定してください。

表 3.1-1 パラメタ一覧

#	データベース種別	定義項目
1	Oracle	add.class.path <sup>1</sup> = Oracle JDBC Thin Driver の JAR ファイルのパス
2	HiRDB	add.class.path <sup>1</sup> = HiRDB Type4 JDBC Driver の JAR ファイルのパス

また、プロセスインスタンスの操作で下記のコマンドを実行する場合、Management Server のメモリ使用量が増大する可能性があります。必要に応じて、初期値より大きい値を設定して下さい。

- cscpiselect (プロセスインスタンスの検索)
- cscpireexec (プロセスインスタンスの再実行)
- cscpidelete (プロセスインスタンスの実行履歴の削除)

表 3.1-2 パラメタ一覧

#	定義項目(初期値)
1	add.jvm.arg=-Xmx256m

<sup>1</sup> マニュアルでは「web.add.class.path」と記載されていますが、こちらは V8 互換のためのキーです。どちらを使用しても問題ありません。

## (2) 簡易構築定義ファイル

構築で使用する簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の<configuration>タグ内に設定するパラメタを次に示します。使用するデータベースに合わせて設定してください。

表 3.1-3 パラメター一覧

#	定義項目	説明	
1	<pre>&lt;param&gt; &lt;param-name&gt;ejbserver.distributedtx.XATransaction.enabled &lt;/param-name&gt; &lt;param-value&gt;%val%&lt;/param-value&gt; &lt;/param&gt;</pre>	<p>「true」または「false」を設定します。</p> <p>HCSC サーバが複数のリソース (DB サーバなど) へアクセスするかどうかを検討して、ライトトランザクション機能を無効にする場合は「true」、有効にする場合は「false」を設定します。</p>	
2	<pre>&lt;param&gt; &lt;param-name&gt;ex.properties&lt;/param-name&gt; &lt;param-value&gt;oracle.jdbc.autoCommitSpecCompliant=false&lt;/ param-value&gt; &lt;/param&gt;</pre>	Oracle 12c に接続する場合、「false」を指定します。	
3	<pre>&lt;param-name&gt;add.class.path&lt;/param-name&gt; &lt;param-value&gt;%val%&lt;/param-value &gt;</pre>	使用するデータベースに合わせて、%val%の値を設定します。	
		Oracle	Oracle JDBC Thin Driver の JAR ファイルのパス
		HiRDB	HiRDB Type4 JDBC Driver の JAR ファイルのパス

## (3) HCSC サーバ構成定義ファイルファイル

HCSC サーバが使用するデータベースのユーザ ID、パスワードを設定します。

表 3.1-4 パラメター一覧

#	定義項目	設定項目
1	<pre>&lt;csc-server&gt;   &lt;id&gt;HCSC サーバが使用するデータベースのユーザ ID&lt;/id&gt; &lt;/csc-server&gt;</pre>	HCSC サーバが使用するデータベースのユーザ ID
2	<pre>&lt;csc-server&gt;   &lt;password &gt;HCSC サーバが使用するデータベースのユーザパスワード&lt;/password &gt; &lt;/csc-server&gt;</pre>	HCSC サーバが使用するデータベースのパスワード

(4) **HCSC サーバセットアップ定義ファイルファイル**

データベースに関連する項目を設定します。システム環境定義書[CSC サーバセットアップ定義ファイル]タブの「フィルタ DB なし」のフィルタを外して設定してください。

表 3.1-5 パラメータ一覧

#	定義項目	設定項目
1	db-use	ON に設定してください。
2	DB コネクタ関連	使用する DB コネクタに合わせて設定してください。
3	データベース関連	使用するデータベースに合わせて設定してください。
4	データベースの RD エリア (表領域) 情報	

(5) **DB Connector 属性ファイルの設定**

使用する DB Connector に合わせて、設定を行ってください。また、DB Connector 属性ファイルのプロパティ定義については、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「4.1.10 DB Connector に設定する<config-property>タグに指定できるプロパティ」を参照してください。

表 3.1-6 インポートする RAR ファイル別の参照先

#	利用するデータベース	インポートする RAR ファイル	参照する表
1	HiRDB	・ライトトランザクション機能を利用する場合 DBConnector_HiRDB_Type4_CP.rar ・ライトトランザクション機能を利用しない場合 DBConnector_HiRDB_Type4_XA.rar	表 3.1-7
2		DBConnector_HiRDB_Type4_CP.rar	表 3.1-8
3	Oracle	・ライトトランザクション機能を利用する場合 DBConnector_Oracle_CP.rar ・ライトトランザクション機能を利用しない場合 DBConnector_Oracle_XA.rar	表 3.1-9
4		DBConnector_Oracle_CP.rar	表 3.1-10

表 3.1-7 dbcon-xadisplayname に設定する DB Connector 属性ファイルの設定内容(HiRDB Type4)

#	定義項目	設定値
1	<display-name>タグ	DB Connector 表示名
2	<config-property>タグに設定する maxBinarySize プロパティ	システムで利用する最大の電文長よりも大きな値※1
3	<transaction-support>	ライトトランザクション機能を利用しない場合
4		ライトトランザクション機能を利用する場合

注※1 LONGVARBINARY\_Access プロパティに REAL を指定した場合に指定してください。また、必ず 0 以外の値を指定してください。

表 3.1-8 dbcon-nodisplayname に設定する DB Connector 属性ファイルの設定内容(HiRDB Type4)

#	定義項目	設定値
1	<display-name>タグ	DB Connector 表示名
2	<config-property>タグに設定する maxBinarySiz プロパティ	システムで利用する最大の電文長よりも大きな値※1
3	<transaction-support>タグ	NoTransaction

注※1 LONGVARBINARY\_Access プロパティに REAL を指定した場合に指定してください。また、必ず 0 以外の値を指定してください。

表 3.1-9 dbcon-xadisplayname に設定する DB Connector 属性ファイルの設定内容(Oracle)

#	定義項目	設定値
1	<display-name>タグ	DB Connector 表示名
2	<config-property>タグに設定する databaseName プロパティ	データベース名※1
3	<transaction-support>	ライトトランザクション機能を利用しない場合
4		ライトトランザクション機能を利用する場合

注※1 Oracle サーバ上の特定のデータベース名 (SID) を指定します。詳細は、Oracle のマニュアルを参照してください。



表 3.1-10 dbcon-nodisplayname に設定する DB Connector 属性ファイルの設定内容(Oracle)

#	定義項目	設定値
1	<display-name>タグ	DB Connector 表示名
2	<config-property>タグに設定する databaseName プロパティ	データベース名※1
3	<transaction-support>	NoTransaction

注※1 Oracle サーバ上の特定のデータベース名 (SID) を指定します。詳細は、Oracle のマニュアルを参照してください。

## 3.2 クラスタ構成に関連するパラメタ

### 3.2.1 ロードバランスクラスタ構成時の設定

ロードバランスクラスタ構成時の設定での注意点について、次から示します。

#### (1) HCSC サーバ構成定義ファイル

HCSC サーバ構成定義ファイルに設定するパラメタのうち、次の表に示すパラメタは設定内容に留意してください。

表 3.2-1 パラメタ一覧

#	分類	使用するタグ名	要素	設定内容
1	cluster の定義	name	クラスタ名	ロードバランスクラスタ内のすべての HCSC サーバで同じ値を設定します。
2		type	クラスタ種別	「LB」を設定します。
3	manager の定義	name	Manager 名	HCSC-Manager で Manager 単位に一意になる識別名を設定します。
4		ip-address	Manager の IP アドレス	物理 IP アドレスを設定します。
5	csc-server の定義	name	HCSC サーバ名	データベースを使用する環境を構築する場合、ロードバランスクラスタ内で一意になる識別名を設定します。 データベースを使用しない環境を構築する場合は、HCSC-Manager 内で一意になる識別名を設定してください。
6	soap-reception の定義	ip-address	Web サーバの IP アドレス	ロードバランスクラスタ内のすべての HCSC サーバに、ロードバランサの IP アドレス値を設定します。
7	ejb-reception の定義	ip-address	NameServer の IP アドレス	ロードバランスクラスタ内のすべての HCSC サーバで、グローバル CORBA ネーミング Service の IP アドレスを設定します。 ただし、CTM を利用していない場合は、localhost を指定してください。

(2) HCSC サーバセットアップ定義ファイル

HCSC サーバセットアップ定義ファイルに設定するパラメタのうち、次の表に示すパラメタはロードバランスクラスタ内のすべての HCSC サーバで同じ値を設定する必要があります。

表 3.2-2 パラメタ一覧

#	分類	定義項目	内容
1	セットアップ構成関連	db-use	データベースの使用有無
2	標準受付関連	request-ejb	標準の同期受付 (SessionBean) の使用有無
3		request-soap	標準の同期受付 (Web サービス) の使用有無
4		request-jms	標準の非同期受付 (MDB (WS-R)) の使用有無
5		request-dbq	標準の非同期受付 (MDB (DB キュー)) の使用有無
6	ユーザ定義受付関連	request-userdef-soap	SOAP 受付の使用有無
7	DB コネクタ関連	dbcon-xadisplayname※1	XATransaction 用または LocalTransaction 用の DB Connector の表示名
8		dbcon-nodisplayname※1	NoTransaction 用 DB Connector の表示名
9	データベース関連	dbtype	使用するデータベース種別
10		jdbc-type	JDBC ドライバ種別
11		jdbc-url※2	JDBC 用データベース URL
12		jdbc-dbhostname	データベースのホスト
13		jdbc-dbconnectinfo	データベースの接続情報
14		db-character-sets	データベース文字コード種
15		db-tbl-split-key	日付データによるテーブルのレンジ分割機能の使用有無
16		db-tbl-split-corcheck-use	相関セットの重複チェック強化の有無

注※1 DB コネクタの設定もロードバランスクラスタ内のすべての HCSC サーバで同じにします。

注※2 指定するホスト名には、データベースを格納しているマシンを指定します。

### (3) HCSC サーバランタイム定義ファイル

HCSC サーバランタイム定義ファイルを設定する場合、ロードバランスクラスタ内のすべての HCSC サーバですべての定義項目を同じ値に設定する必要があります。

また、CTM を使用する場合、次の定義項目を追加してください。

- ・request-ejb.ctm=true

クラスタを構成する HCSC サーバで J2EE サーバ名が異なる場合、次の定義項目を追加してください。

- ・request-ejb.optional.name=<クラスタ内で共通の同期リクエストサービス (SessionBean) の別名>

### 3.2.2 HA クラスタ構成時の設定

HA クラスタ構成時の設定での注意点について次から示します。

#### (1) HCSC サーバ構成定義ファイル

HCSC サーバ構成定義ファイルに設定するパラメタのうち、次の表に示すパラメタは設定内容に留意してください。これ以外のパラメタに関しては、実行系と待機系で同じ値を設定してください。

表 3.2-3 パラメタ一覧

#	分類	使用するタグ名	要素	設定内容
1	cluster の定義	type	クラスタ種別	「HA」を設定します。
2	manager の定義	name	Manager 名	実行系と待機系で異なる値を設定します。
3		ip-address	Manager の IP アドレス	実行系と待機系それぞれの物理 IP アドレスを設定します。
4	csc-server の定義	name	HCSC サーバ名	データベースを使用する環境を構築する場合、ユニークな名称にしてください。
5	soap-reception の定義	ip-address	Web サーバの IP アドレス	論理 IP アドレスを設定します。
6	ejb-reception の定義	ip-address	NameServer の IP アドレス	論理 IP アドレスを設定します。

#### (2) HCSC サーバセットアップ定義ファイル

HCSC サーバセットアップ定義ファイルに設定する定義項目のうち、次の表に示す定義項目は実行系と待機系で同じ値を設定してください。

表 3.2-4 パラメタ一覧

#	分類	定義項目	内容
1	セットアップ構成関連	db-use	データベースの使用有無
2	標準受付関連	request-ejb	標準の同期受付 (SessionBean) の使用有無
3		request-soap	標準の同期受付 (Web サービス) の使用有無

4		request-jms	標準の非同期受付 (MDB (WS-R)) の使用有無
5		request-dbq	標準の非同期受付 (MDB (DB キュー)) の使用有無
6	ユーザ定義受付関連	request-userdef-soap	ユーザ定義受付の使用有無
7	DB コネクタ関連	dbcon-xadisplayname※1	XATransaction 用または LocalTransaction 用の DB Connector の表示名
8		dbcon-nodisplayname※1	NoTransaction 用 DB Connector の表示名
9	データベース関連	dbtype	使用するデータベース種別
10		jdbc-type	JDBC ドライバ種別
11		jdbc-url※2	JDBC 用データベース URL
12		jdbc-dbhostname	データベースのホスト
13		jdbc-dbconnectinfo	データベースの接続情報
14		db-character-sets	データベース文字コード種
15		db-tbl-split-key	日付データによるテーブルのレンジ分割機能の使用有無
16		db-tbl-split-corcheck-use	相関セットの重複チェック強化の有無
17	XML 解析	xmlanalyze-mode	XML の解析モードの設定

注※1 DB コネクタの設定も実行系と待機系で同じにします。

注※2 指定するホスト名には、データベースを格納しているマシンを指定します。

### (3) HCSC サーバランタイム定義ファイル

HCSC サーバランタイム定義ファイルを設定する場合、すべての実行系と待機系で同じ値を設定する必要があります。

また、クラスタを構成する HCSC サーバで J2EE サーバ名が異なる場合、次の定義項目を追加してください。

・request-ejb.optional.name=<クラスタ内で共通の同期リクエストサービス (SessionBean) の別名>

### 3.3 1 台のマシンに複数の環境を構築する場合に関連するパラメタ

使用する環境ごとに CSCMNG\_HOME 環境変数を割り当てることで、1 台のマシンに複数の運用・実行環境を構築することができます。1 台のマシンで複数の運用・実行環境を構築する例を次の図に示します。

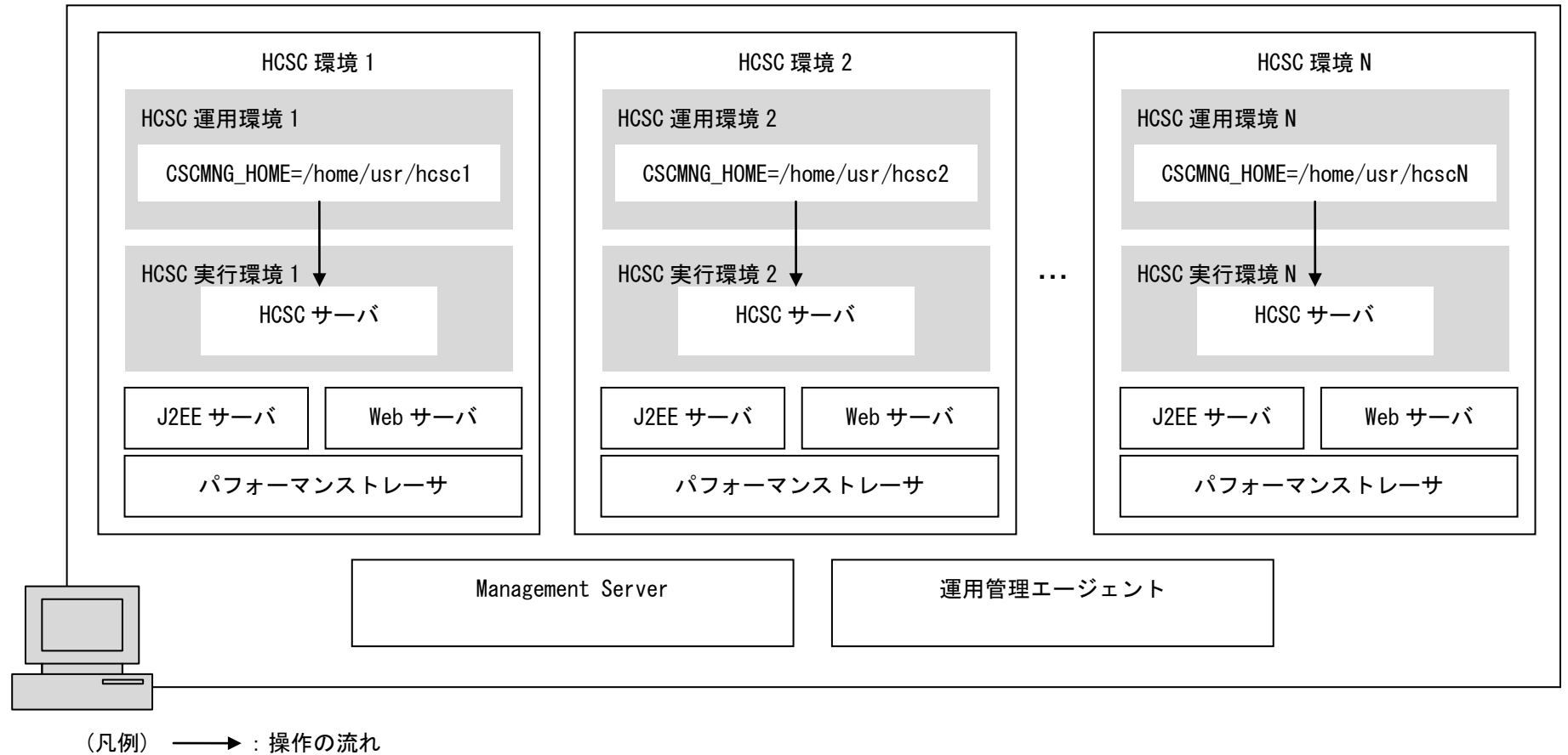
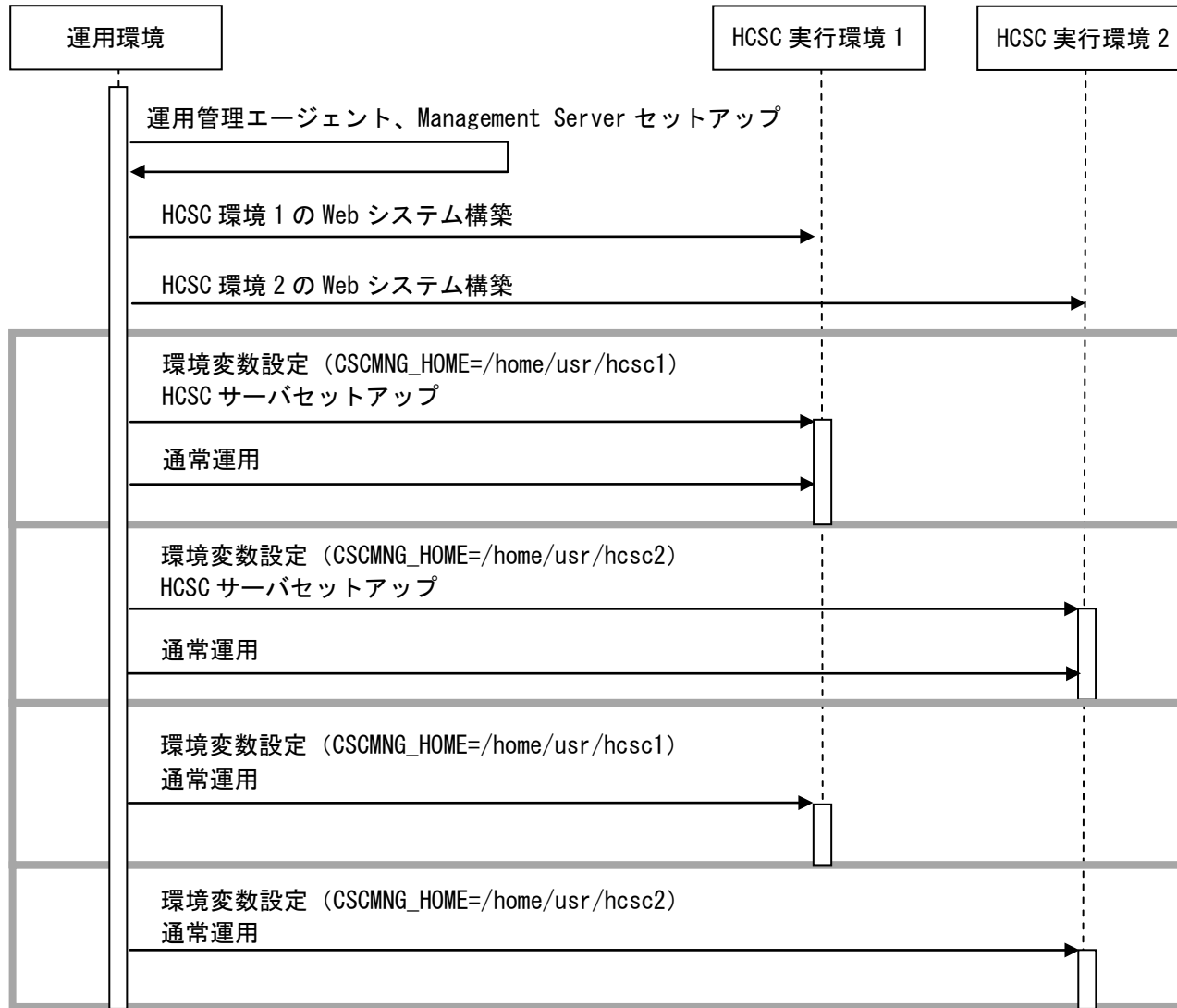


図 3.3-1 1 台のマシンに複数の運用・実行環境を構築する例

### 3.3.1 1 台のマシンに複数の環境を構築する流れ

1 台のマシンに 2 つの環境を構築する場合の流れを次の図に示します。



(凡例) —→ : 操作の流れ

図 3.3-2 1 台のマシンに 2 つの環境を構築する場合の流れ



### 3.3.2 CSCMNG\_HOME 環境変数の設定

1 台のマシンに複数の環境を構築する場合、構築する環境の数だけ CSCMNG\_HOME 環境変数を設定する必要があります。

CSCMNG\_HOME 環境変数には任意フォルダの絶対パスを設定してください。設定したパスの下にある定義ファイルが HCSC-Manager によって参照されます。

HCSC-Manager が参照する CSCMNG\_HOME 環境変数以下の定義ファイルを次の表に示します。

表 3.3-1 HCSC-Manager が参照する CSCMNG\_HOME 環境変数以下の定義ファイル

#	ディレクトリ名	説明
1	<code>\$CSCMNG_HOME/config</code>	運用環境で使用するコンフィグファイル格納ディレクトリ
2	<code>\$CSCMNG_HOME/log</code>	運用環境で使用するデフォルトログディレクトリ
3	<code>\$CSCMNG_HOME/repository</code>	運用環境で使用するデフォルトリポジトリディレクトリ
4	<code>\$CSCMNG_HOME/spool</code>	運用環境で使用する一時ディレクトリ

CSCMNG\_HOME 環境変数として「/home/usr/hcsc1」を設定する例を次に示します。

#### ● Bourne シェルの設定例

1. 次の内容を記述したファイルを任意のディレクトリに任意のファイル名で作成します。

```
export CSCMNG_HOME=/home/usr/hcsc1
```

作成したファイル名を「hcsc1\_env.sh」とします。

2. 次に示す形式で 1. で作成したファイルを実行します。

```
. hcsc1_env.sh
```

ファイルを実行すると、hcsc1\_env.sh ファイルを読み込み、CSCMNG\_HOME 環境変数が有効になります。

#### ● C シェルの設定例

1. 次の内容を記述したファイルを任意のディレクトリに任意のファイル名で作成します。

```
setenv CSCMNG_HOME=/home/usr/hcsc1
```

作成したファイル名を「hcsc1\_env.sh」とします。

2. 次に示す形式で 1.で作成したファイルを実行します。

```
source hcsc1_env.sh
```

ファイルを実行すると、hcsc1\_env.sh ファイルを読み込み、CSCMNG\_HOME 環境変数が有効になります。

### 3.3.3 設定に注意が必要なパラメタ

#### (1) HCSC-Manager 定義ファイル

初期値から変更する場合は、\$CSCMNG\_HOME/config/manager ディレクトリを作成し、テンプレートファイルをコピーして使用してください。

テンプレートファイル	/opt/Cosminexus/CSC/config/manager/templates/cscmng.properties
コピー先	\$CSCMNG_HOME/config/manager/cscmng.properties

また、次に示すパラメタのうち、省略値から変更したい場合は、複数の環境で異なる値になるよう設定してください。

表 3.3-2 CSCMNG\_HOME 環境変数が設定されるパラメタ

#	分類	定義項目(省略値)	内容
1	リポジトリ設定	cscmng.repository.root=\$CSCMNG_HOME/repository	リポジトリのルートディレクトリパス
2	ログ設定	cscmng.log.dir=\$CSCMNG_HOME/log/manager	ログ出力先ディレクトリパス
3	環境バックアップ	cscmng.envbackup.dir.envrestore=\$CSCMNG_HOME/spool/manager	リストア前の環境バックアップの出力先ディレクトリパス
4		cscmng.envbackup.dir.import=\$CSCMNG_HOME/spool/manager	インポート前の環境バックアップの出力先ディレクトリパス

#### (2) HCSC-Manager コマンド共通定義ファイル

初期値から変更する場合は、\$CSCMNG\_HOME/config/manager ディレクトリを作成し、テンプレートファイルをコピーして使用してください。

テンプレートファイル	/opt/Cosminexus/CSC/config/manager/templates/csccmd.properties
コピー先	\$CSCMNG_HOME/config/manager/csccmd.properties

### (3) 簡易構築定義ファイル

SmartComposer 機能を使用して、1 台のマシンに複数のサービスユニットを定義する方法は大きくわけて 2 種類あります。

- ① 1 つの Web システムに複数のサービスユニットを定義
- ② 1 つの Web システムに 1 つのサービスユニットを定義 × 複数環境分

それぞれの方法で、推奨する要件を以下に示します。構築するシステムの要件に合わせて、定義方法を選択してください。

#	①	②
要件	Web システムを一括運用したい、または各プロセスの定義を共通化したい場合に推奨します。 サービスユニット単位での運用、保守(設定変更)も可能です。	各 Web システムを独立させて、個別に運用したい場合に推奨します。

以下のパラメタは、設定に注意してください。

表 3.3-3 パラメタ一覧

#	機能	パラメタ	注意事項
1	ホスト	ホストの定義 <host></host>	②の方法で定義を作成する場合： ホストの定義<host></host>は 1 台目の定義のみに記述し、2 台目以降では定義しないでください。(すでにホストを定義しているため、構築時にエラーになります。)
2	Web システム	Web システム名 <web-system>/<name>	②の方法で定義を作成する場合： 複数の Web システムを構築するため、各 Web システム名は重複しないように設定する必要があります。
3	論理サーバ	各論理サーバ名 <logical-server-name>	各論理サーバ名は、重複しないように設定する必要があります。
4		各ポート番号	各ポート番号は、重複しないように設定する必要があります。ポート番号については、ポート番号の設定の章を参照してください。(運用管理エージェント、Management Server は除く。)

5	PRF	PRF 識別子(PRFID) <param-name>PRFID</param-name>	PRFID は省略すると、「PRF_ID」で構築されます。2 台目以降を構築する際は、省略すると同一の値となりエラーになるため、各 PRF サーバで異なる値を設定してください。
---	-----	--------------------------------------------------	------------------------------------------------------------------------------------------

・重複しないように設定する項目の詳細については、マニュアル「Cosminexus V9 アプリケーションサーバ システム構築・運用ガイド 4.8.2 簡易構築定義ファイルを設定するときの注意事項（その他の機能使用時）(4) 各機能で共通の注意事項」を参照してください。

#### (4) HCSC サーバ構成定義ファイル

以下のパラメタは、複数の環境ですべて異なる値を設定してください。

表 3.3-4 パラメタ一覧

#	分類	使用するタグ名	要素	設定内容
1	csc-server の定義	j2ee-server-name	HCSC サーバが使用する J2EE サーバ名	HCSC サーバをセットアップする J2EE サーバ名を指定します。
2		name	HCSC サーバ名	HCSC サーバ名を指定します。
3	soap-reception の定義	port	Web サーバのポート番号	Web サーバの受付ポート番号を指定します
4	ejb-reception の定義	port	NameServer のポート番号	ネーミングサービスの受付ポート番号を指定します。

#### (5) HCSC サーバセットアップ定義ファイル

以下のパラメタは、複数の環境ですべて異なる値を設定して下さい。

表 3.3-5 パラメタ一覧

#	分類	定義項目	内容
1	J2EE 関連	naming-service-port	ネーミングサービスのポート番号

#### (6) snapshot ログ収集対象定義ファイル

CSCMNG\_HOME を設定すると、ログ・リポジトリ・バックアップファイル・HCSC-Manager 定義ファイルの出力先が変わるため、snapshot ログの収集対象定義ファイルに以下を追加して下さい。下記の項目に対して複数環境分の定義を追加する必要があります。

また、(1)でログ・リポジトリ・バックアップファイルの出力先を変更している場合は、変更した値を設定して下さい。



# 4 受付・アダプタ単位のパラメタの設定

この章では、受付/アダプタ単位で設定するパラメタについて説明します。

## 本章の構成

- 4.1 作業フォルダまたは共通フォルダを利用する場合の設定
- 4.2 FTP 連携で使用する受付・アダプタの設定

## 4.1 作業フォルダまたは共通フォルダを利用する場合の設定

作業フォルダまたは共通フォルダを利用する場合の設定を次に示します。

### (1) HCSC サーバランタイム定義ファイル

作業フォルダ、共通フォルダをデフォルト値から変更して使用する場合は、次のパラメータに設定します。

表 4.1-1 パラメータ一覧

#	設定項目	デフォルト値
1	work-folder	/opt/Cosminexus/CSC/spool/ftp/work
2	common-folder-<共通フォルダ定義名>	/opt/Cosminexus/CSC/spool/ftp/com

### (2) Management Server 用オプション定義ファイル(msserver.cfg)

作業フォルダまたは共通フォルダを利用する場合、Management Server のメモリ使用量が増大する可能性があります。Java ヒープの最大サイズを見積もり、適切な値を設定します。Java ヒープの最大サイズを算出する式を以下に示します。

$200\text{MB} + (\text{cscfscls}, \text{cscfswls} \text{ コマンドで表示する最大ファイル数} \times 5000 \text{ バイト})$

※算出した値が 256MB 以下になる場合は、256MB で設定してください。

表 4.1-2 パラメータ一覧

#	定義項目
2	add.jvm.arg=-Xmx256m

## 4.2 FTP 連携で使用する受付・アダプタの設定

### 4.2.1 コンポーネント構成

FTP 連携のコンポーネント構成を以下に示します。

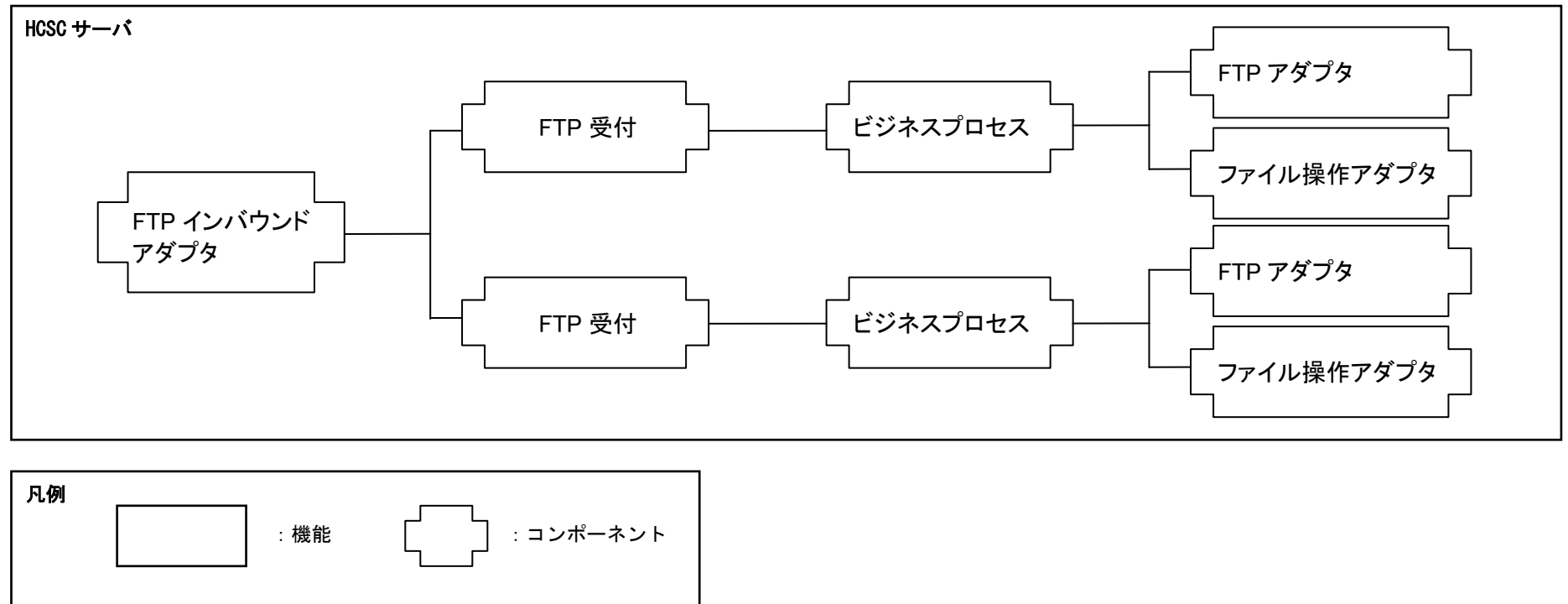


図 4.2-1 コンポーネント構成



FTP 連携で使用する定義一覧を以下に示します。

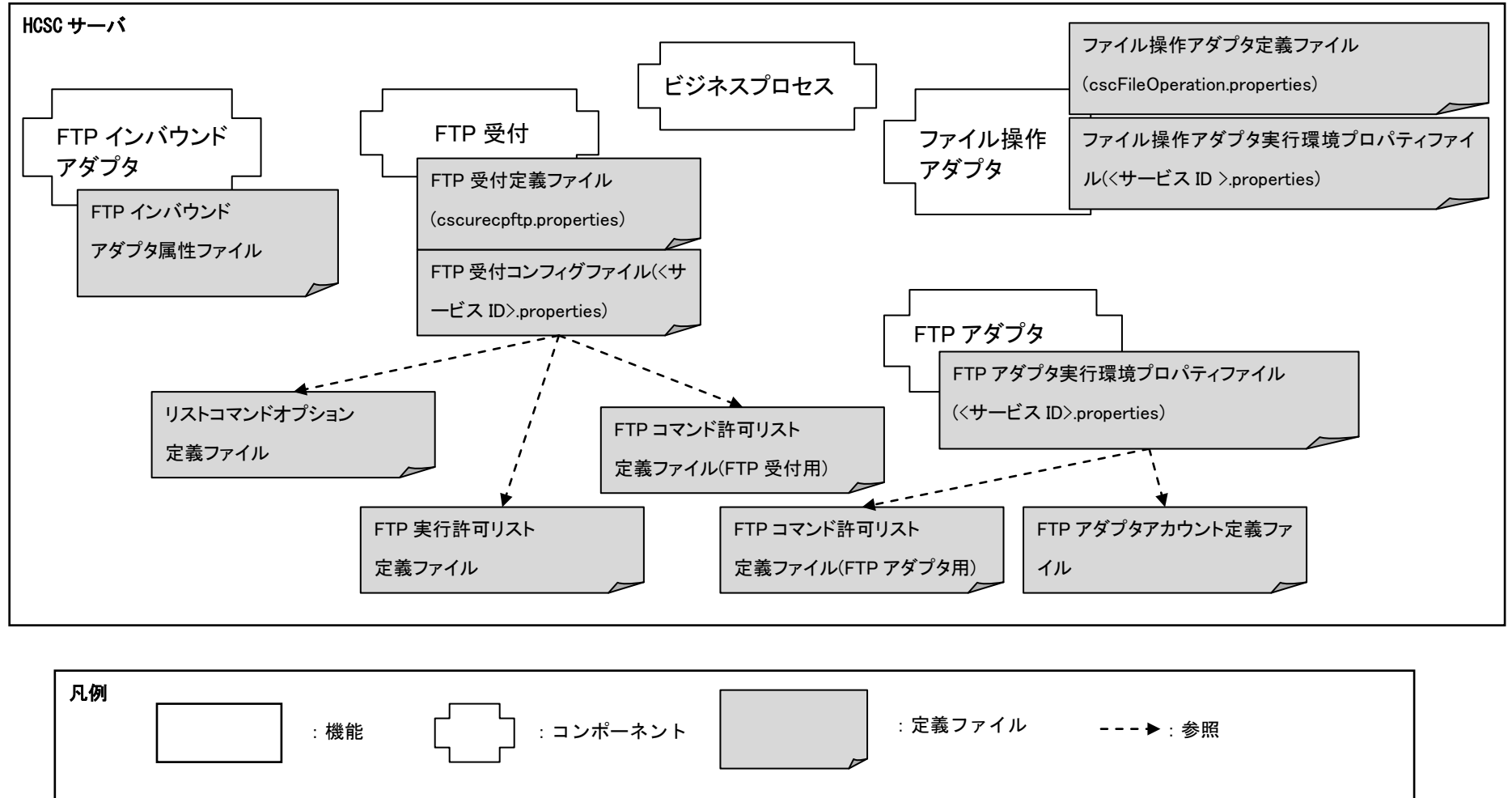


図 4.2-2 FTP 連携で使用する定義一覧

表 4.2-1 定義ファイル一覧

#	定義ファイル名	定義利用 コンポーネント	説明	格納先
1	FTP インバウンドアダプタ属性ファイル	FTP インバウンドアダプタ	FTP インバウンドアダプタの定義。	任意 (コマンド引数に設定して使用)
2	FTP 受付定義ファイル (cscurecpftp.properties)	FTP 受付	FTP 受付が参照する FTP インバウンドアダプタ、FTP 受付のインスタンス数、作業フォルダの自動削除有無を定義。	リポジトリ (FTP 受付開発時に開発環境の[ユーザ定義受付画面(詳細)]の [独自定義ファイル] で設定)
3	FTP 受付コンフィグファイル (<サービス ID>.properties)		#5 の FTP 実行許可リスト、#6 の FTP コマンド許可リスト定義ファイルの参照先を定義。	/opt/Cosminexus/CSC/config/ftprecp/<サービス ID>.properties
4	リストコマンドオプション定義ファイル		リストコマンドのオプション文字列の定義。	任意 (#3 FTP 受付コンフィグファイルから参照される)
5	FTP 実行許可リスト定義ファイル		FTP ユーザに対してビジネスプロセスの実行許可・拒否の定義。	任意 (#3 FTP 受付コンフィグファイルから参照される)
6	FTP コマンド許可リスト定義ファイル(FTP 受付用)		FTP コマンドの実行許可・拒否の定義。	任意 (#3 FTP 受付コンフィグファイルから参照される)
7	FTP コマンド許可リスト定義ファイル(FTP アダプタ用)	FTP アダプタ	FTP コマンドの実行許可・拒否の定義。	任意 (#8 FTP アダプタ実行環境プロパティファイルから参照される)
8	FTP アダプタ実行環境プロパティファイル (<サービス ID>.properties)	FTP アダプタ	FTP アダプタの定義。	/opt/Cosminexus/CSC/custom-adapter/FTP/config/<サービス ID>.properties
9	アカウント定義ファイル		FTP アダプタが使用するアカウントの定義。	任意 (#7 FTP アダプタ実行環境プロパティファイルから参照される)
10	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	ファイル操作アダプタの定義(排他のリトライ間隔・回数、性能チューニング)。

<code>&lt;サービス ID&gt;.properties</code>			
-----------------------------------------	--	--	--

#### 4.2.2 ネットワーク設計

FTP インバウンドアダプタ、FTP アダプタのバインドする IP アドレス、ポート番号のネットワーク設計を行います。  
FTP 連携の接続構成を示します。

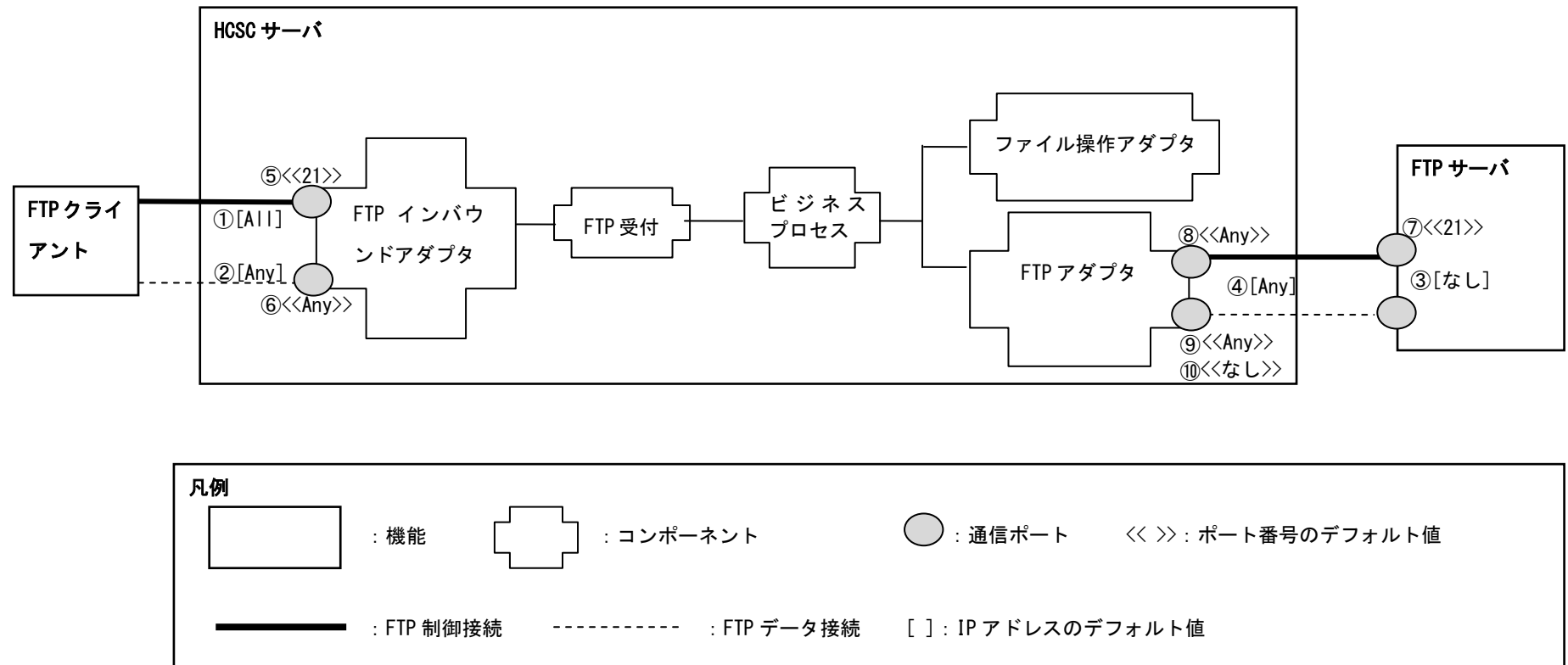


図 4.2-3 FTP 連携の接続構成

表 4.2-2 IP アドレスの設計項目

#	設計対象	設計項目	設計目的
1	FTP インバウンドアダプタ	①制御コネクションの IP アドレス	OS に複数 IP アドレスが存在する場合、FTP インバウンドアダプタが FTP クライアントに接続する IP アドレスを指定しないとコネクションを接続できない可能性があるため。
2		②データコネクションの IP アドレス	PORT モード使用時 OS に複数 IP アドレスが存在する場合、FTP インバウンドアダプタが FTP クライアントに接続する IP アドレスを指定しないとデータコネクションを接続できない可能性があるため。
3	FTP アダプタ	③接続する FTP サーバの IP アドレスまたはホスト名	FTP アダプタが接続する FTP サーバの IP アドレスまたはホスト名を指定します。
4		④制御コネクションの IP アドレス	FTP アダプタを実行するホストの OS に複数 IP アドレスが存在する場合、FTP アダプタが FTP サーバに接続する IP アドレスを指定しないとコネクションを接続できない可能性があるため。
5		④データコネクションの IP アドレス	

表 4.2-3 ポート番号の設計項目

#	設計対象	設計項目	設計目的
1	FTP インバウンドアダプタ	⑤制御コネクションのポート番号	初期値の「21」から変更する場合は、1-65535 の整数で空いているポート番号を設定します。
2		⑥データコネクションのポート番号	初期値の「Any available」を指定した場合、空いているポート番号が割り当てられます。ポート番号を指定する場合は 1-65535 の整数で、空いているポート番号を設定します。
3	FTP アダプタ	⑦接続する FTP サーバの制御コネクションのポート番号	FTP アダプタが接続する FTP サーバのポート番号を指定します。
4		⑧制御コネクションのポート番号の範囲(最小値、最大値)	制御コネクションのポート番号の範囲を指定できます。⑨・⑩と重複しない範囲を指定する必要があります。省略した場合は、任意のポート番号が設定されます。

5		⑨PASV モードの場合のデータコネクションのポート番号の範囲(最小値、最大値)	データコネクションが PASV モードの場合は、データコネクションを確立するポート番号の範囲を指定できます。⑧と重複しない範囲を指定する必要があります。省略した場合は任意のポート番号が設定されます。
6		⑩PORT モードの場合のデータコネクションのポート番号の範囲(最小値、最大値)	データコネクションが PORT モードの場合は、待ち受けるポート番号の範囲を指定する必要があります。⑧と重複しない範囲を指定する必要があります。

表 4.2-4 IP アドレスのパラメータ一覧

#	パラメタ	定義ファイル	定義項目(初期値)	定義範囲
①	FTP インバウンドアダプタの制御コネクションの IP アドレス	FTP インバウンドアダプタ属性定義ファイル	<code>&lt;config-property-name&gt;nioListener_localAddress&lt;/config-property-name&gt;</code> <code>&lt;config-property-type&gt;java.lang.String&lt;/config-property-type&gt;</code> <code>&lt;config-property-value&gt;All available&lt;/config-property-value&gt;</code>	0.0.0.0～ 255.255.255.255 0:0:0:0:0:0:0:0～ FFFF:FFFF:FF FF:FFFF:FFFF: FFFF:FFFF:FF FF
②	FTP インバウンドアダプタのデータコネクションの IP アドレス		<code>&lt;config-property-name&gt;nioListener_dataConnection_active_localAddress&lt;/config-property-name&gt;</code> <code>&lt;config-property-type&gt;java.lang.String&lt;/config-property-type&gt;</code> <code>&lt;config-property-value&gt;Any available&lt;/config-property-value&gt;</code>	0.0.0.0～ 255.255.255.255 0:0:0:0:0:0:0:0～ FFFF:FFFF:FF FF:FFFF:FFFF: FFFF:FFFF:FF FF
③	接続する FTP サーバの IP アドレスまたはホスト名	FTP アダプタ実行環境プロパティファイル	ftpadp.host-ipaddr=(デフォルト値なし)	実 IP アドレスまたはホスト名

④	FTP アダプタを実行するホストの IP アドレス		ftpadp.client-ipaddr=(デフォルト値なし)	0.0.0.0～ 255.255.255.255 0:0:0:0:0:0:0:0～ FFFF:FFFF:FF FF:FFFF:FFFF: FFFF:FFFF:FF FF
---	---------------------------	--	---------------------------------	--------------------------------------------------------------------------------------------------------

表 4.2-5 ポート番号のパラメータ一覧

#	パラメータ	定義ファイル	定義項目(初期値)	定義範囲	
⑤	FTP インバウンドアダプタの制御 コネクションのポート番号	FTP インバウンドアダ プタ属性ファイル	<config-property-name>nioListener_port</config-property-name> <config-property-type>java.lang.String</config-property-type> <config-property-value>21</config-property-value>	1-65535	
⑥	FTP インバウンドアダプタのデー タコネクション確立時のポート番号		<config-property-name>nioListener_dataConnection_active_localPo rt</config-property-name> <config-property-type>java.lang.String</config-property-type> <config-property-value>Any available</config-property-value>	1-65535	
⑦	FTP サーバの制御コネクションの ポート番号	FTP アダプタ実行環境 プロパティファイル	ftpadp.host-con-port=21	1-65535	
⑧	FTP アダプタの制御コネ クションのポート番号の 範囲		最小値	ftpadp.client-con-port.min=(デフォルト値なし)	1024-65535
			最大値	ftpadp.client-con-port.max=(デフォルト値なし)	1024-65535
⑨	PASV モードの場合の FTP アダプタがデータコ ネクションを待ち受ける ポート番号の範囲		最小値	ftpadp.client-pasv-port.min=(デフォルト値なし)	1024-65535
		最大値	ftpadp.client-pasv-port.max=(デフォルト値なし)	1024-65535	

⑩	PORT の場合の FTP アダプタがデータコネクションを待ち受けるポート番号の範囲	最小値	ftpadp.listen-port.min=(デフォルト値なし)	1024-65535
		最大値	ftpadp.listen-port.max=(デフォルト値なし)	1024-65535

### 4.2.3 ユーザ設計

FTP ユーザの設計をします。

<前提> FTP サーバに FTP ユーザが設定されていること。

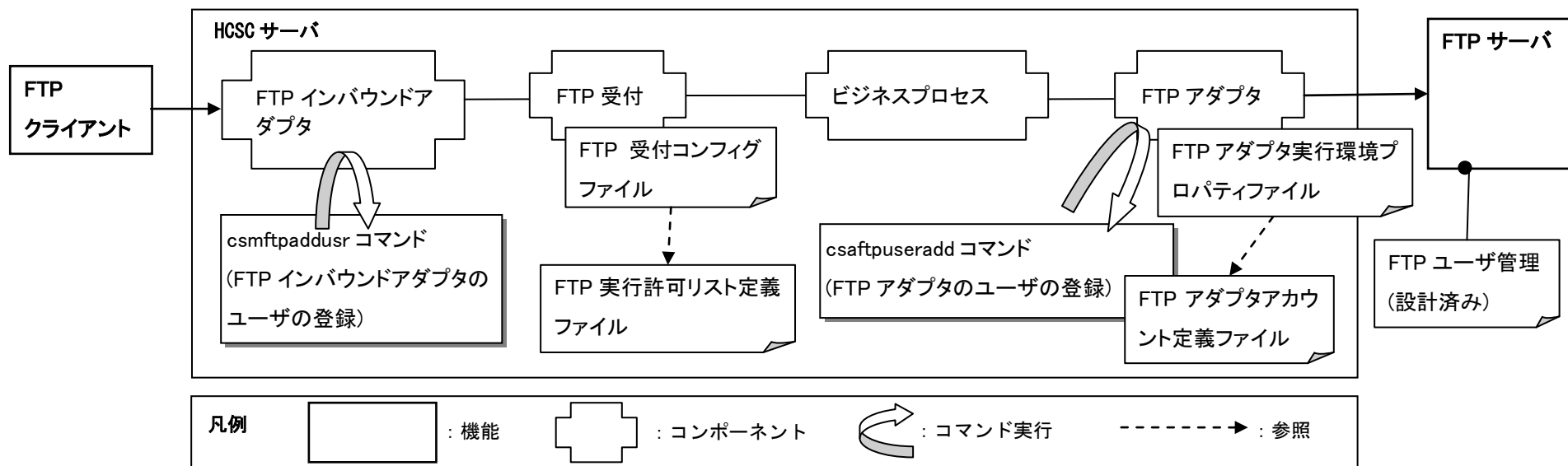


図 4.2-4 ユーザ設計の対象箇所

表 4.2-6 ユーザ設計項目一覧

#	設計対象	設計項目	設計内容
1	FTP インバウンドアダプタ	FTP ユーザ名※1	FTP クライアントからの接続に対応する FTP ユーザ名を設計します。
2		FTP ユーザ名のパスワード※1	FTP ユーザ名に対応するパスワードを設計します。
3	FTP アダプタ	FTP ユーザ名※2	接続する FTP サーバの FTP ユーザ名を設計します。
4		FTP ユーザ名のパスワード※2	FTP ユーザに対応するパスワードを設計します。
5		FTP ユーザ名に対応する課金情報※2	FTP ユーザに対応する課金情報を設計します。課金情報を設定しない・対応していない FTP サーバに接続する場合は設計不要です。
6		FTP アダプタアカウント定義ファイルの配置場所※2	任意のパスに任意のファイル名で格納できます。このファイルは、FTP アダプタ運用コマンドにより登録・参照されるもので、ユーザが定義ファイルを編集する必要はありません。また、このファイルの絶対パスを FTP アダプタ実行環境プロパティファイルの ftpadp.account-inf-filepath プロパティで指定する必要があります。

注※1 FTP インバウンドアダプタのユーザ登録は、`csmtftpaddusr` コマンド (FTP インバウンドアダプタのユーザの登録・更新) で実行します。

注※2 FTP アダプタのユーザ登録は、`csaftpuseradd` コマンド (FTP アダプタのユーザの登録・更新) で実行します。

#### 4.2.4 セキュリティ設計

セキュリティ設計の目的を次に示します。

- FTP クライアントからの不正な要求から、システムを守る。
- FTP サーバに対して、不正な要求を出さない。
- サービスプラットフォームを中継するデータを平文で流さない。
- 想定外のファイルが転送された場合に異常と判断し処理を打ち切りたい (他への影響を考慮)。



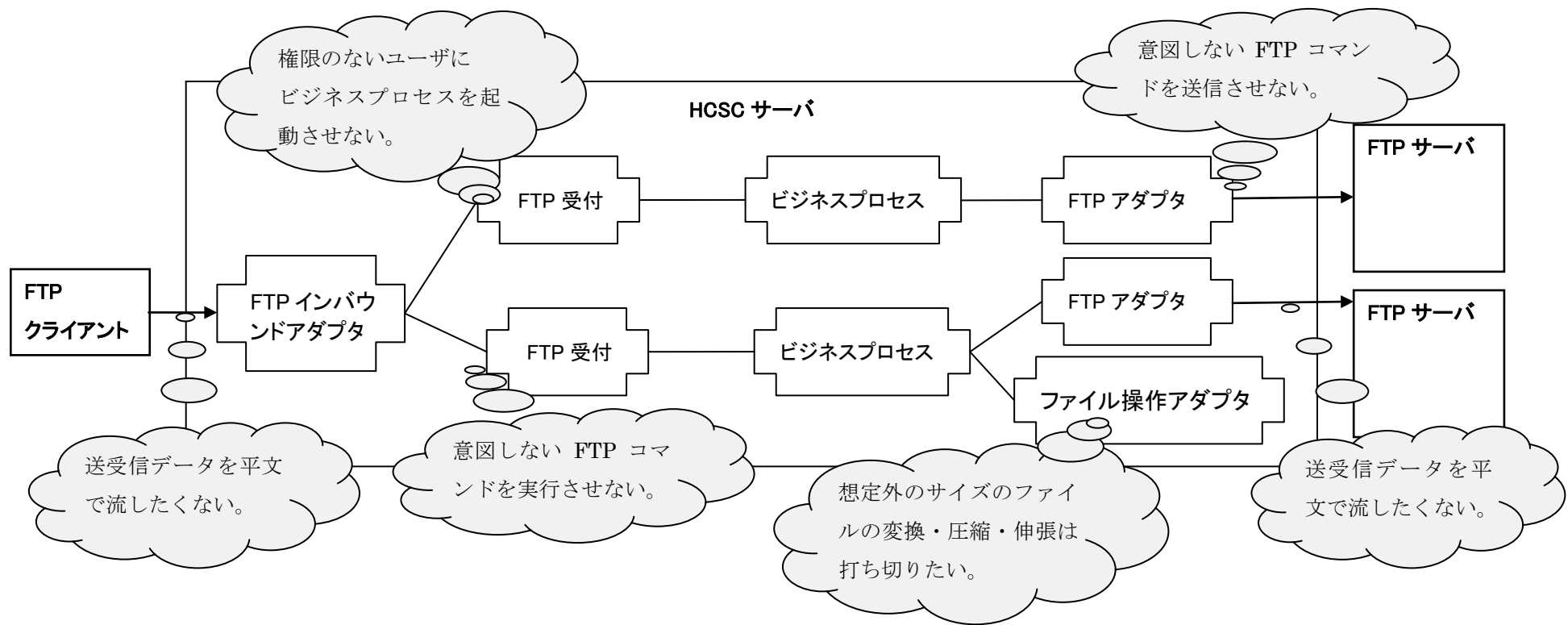


図 4.2-5 セキュリティ設計箇所

表 4.2-7 セキュリティ機能と役割

#	セキュリティ機能	役割	設定箇所
1	ユーザアクセス制御	許可しないユーザがビジネスプロセスを起動することを防ぎます。	FTP 受付
2	受信 FTP コマンドアクセス制御	FTP クライアントが送信した許可しない FTP コマンドを実行することを防ぎます。	FTP 受付
3	送信 FTP コマンドアクセス制御	FTP アダプタが許可しない FTP コマンドを送信することを防ぎます。	FTP アダプタ
4	FTPS	送受信データを SSL/TLS で暗号化しデータの機密性と整合性を保障します。	FTP インバウンドアダプタ FTP アダプタ

5	変換・圧縮・伸張打ち切り	想定外に大きいファイルが変換・圧縮・伸張の入力/出力ファイルとして与えられた場合に、異常と判断しシステムへの影響を考慮して変換処理を打ち切ります。	ファイル操作アダプタ
---	--------------	---------------------------------------------------------------------------	------------

表 4.2-8 セキュリティ設計の設定パラメータ一覧

#	分類	パラメータ	定義ファイル	定義項目(初期値)	定義範囲
1	ユーザ アクセ	ビジネスプロセスの実行を許可するユーザ	FTP 実行許可リスト定義ファイル	urecp-ftp.permission.allow=*	文字列または半角アスタリスク (*)
2	ス制御	ビジネスプロセスの実行を許可しないユーザ		urecp-ftp.permission.deny=(デフォルト値なし)	
3		FTP 実行許可リスト定義ファイルの配置場所		urecp-ftp.permission-def.filepath=(デフォルト値なし)	
4	受信 FTP コ マンド	実行を許可する FTP コマンド	(FTP 受付用) FTP コマンド許可リスト定義ファイル	ftp-command.allow=(デフォルト値なし)	文字列または半角アスタリスク (*)
5	アクセ ス制御	SITE コマンドで実行を許可するコマンド		ftp-command.site.allow=(デフォルト値なし)	文字列または半角アスタリスク (*)
6		(FTP 受付用)FTP コマンド許可リスト定義ファイル格納先		urecp-ftp.ftp-command-allow.filepath=(デフォルト値なし)	文字列 (2 バイト文字含む) および空白
7	送信 FTP コ マンド	実行を許可する FTP コマンド	(FTP アダプタ用) FTP コマンド許可リスト定義ファイル	ftp-command.allow=(デフォルト値なし)	文字列または半角アスタリスク (*)
8	アクセ ス制御	SITE コマンドで実行を許可するコマンド		ftp-command.site.allow=(デフォルト値なし)	文字列または半角アスタリスク (*)
9		(FTP アダプタ用)FTP コマンド許可リスト定義ファイル格納先		ftpadp.cscthrough-filepath=(デフォルト値なし)	文字列 (2 バイト文字含む) および空白
10	変換打 ち切り	ファイルの変換打ち切り入力ファイルサイズ	ファイル操作アダプタ 実行環境プロパティフ	transform.limitsize=62914560	0-2147483647

11	圧縮打ち切り	ファイルの圧縮を打ち切る入力ファイルのサイズ	ファイル	compress.inputfile.limitsize=4294967296	1-4294967296
12		ファイルの圧縮を打ち切る出力ファイルのサイズ		compress.outputfile.limitsize=4294967296	1-4294967296
13	伸張打ち切り	ファイルの伸張を打ち切る入力ファイルのサイズ		extract.inputfile.limitsize=4294967296	1-4294967296
14		ファイルの伸張を打ち切る出力ファイルのサイズ		extract.outputfile.limitsize=4294967296	1-4294967296

・FTP インバウンドアダプタの FTPS 関連設定パラメタに関しては、マニュアル「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 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) を参照してください。

・FTP アダプタの FTPS 関連設定パラメタに関しては、マニュアル「サービスプラットフォーム リファレンス 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。

# 5 性能改善パラメタの設定

この章では、性能が改善できるパラメタについて説明します。

## 本章の構成

- 5.1 ログレベルの設定
- 5.2 XML 処理性能の改善
- 5.3 ファイルディスクリプタの設定(※Linux 限定)

## 5.1 ログレベルの設定

性能を重視する場合に、ログレベルを下げることで性能が良くなる場合があります。ログレベルに関するパラメタを次に示します。

### (1) HCSC サーバランタイム定義ファイル

表 5.1-1 パラメタ一覧

#	定義項目(初期値)	設定値	説明	定義範囲
1	bptrace-level=10	0	ビジネスプロセス基盤のメソッドトレースの出力レベルを指定します。0を指定すると、トレースを出力しません。	0   10   20   30
2	requesttrace=ON	OFF	リクエストトレースを採取するかどうかを指定します。 OFFを指定すると、トレースを採取しません。	ON   OFF
3	methodtrace-level=3	1	メソッドトレースの出力レベルを指定します。 1を指定すると、入力ファイルおよび出力ファイルの情報(ファイルパス、ファイルの形式、ファイルサイズ、レコード数)のみ出力します。	1   2   3   4   5
4	telegramtrace=OFF	OFF	ユーザ電文トレースを採取するかどうかを指定します。 OFFを指定すると、トレースを採取しません。	ON   OFF

(#2): requesttrace=OFFにすると、利用者側でサービスプラットフォームが動作していることのログを確認することができません。トラブルシュート上、リクエストトレースは有効となるため、性能要件を満たしているのであれば、あえてOFFにする必要はありません。必要に応じてOFFにすることを検討ください。

### (2) ユーザ拡張性能解析トレース

(1)の設定にした場合、保守ログを出力しないのでエラーが発生しない障害の調査が難しくなる恐れがあります。(たとえば、リクエストのタイムアウトが発生していないが、処理が遅いなど)そこで、BP内の動作だけはPRFトレースとして追えるようにユーザ拡張性能解析トレースを設定します。

ユーザ拡張性能解析トレースの定義は、簡易構築定義ファイルの論理 J2EE サーバ (j2ee-server) の<configuration>タグ内に JavaVM 起動パラメタ (add.jvm.arg) で指定します。

●ユーザ拡張性能解析トレース設定ファイル

(uCSP システム構成定義書) [ユーザ拡張性能解析トレース設定ファイル]シートの内容を下記 userprf.cfg ファイルに設定してください。

```
/opt/Cosminexus/jdk/usrconf/userprf.cfg
```

ファイル格納先を変更した場合は、パラメータ一覧#2 の内容を合わせて変更してください。

表 5.1-2 パラメータ一覧

#	定義項目	説明
1	<pre>&lt;param&gt; &lt;param-name&gt;add.jvm.arg&lt;/param-name&gt; &lt;param-value&gt;-Djvm.userprf.Enable=true&lt;/param-value&gt; &lt;/param&gt;</pre>	ユーザ拡張性能解析トレースを使用する場合、有効(true)に設定します。
2	<pre>&lt;param&gt; &lt;param-name&gt;add.jvm.arg&lt;/param-name&gt; &lt;param-value&gt;-Djvm.userprf.File=/opt/Cosminexus/jdk/usrconf/userprf.cfg &lt;/param-value&gt; &lt;/param&gt;</pre>	ユーザ拡張性能解析トレースで使用するユーザ拡張性能解析トレース設定ファイルのファイルパスを指定します。
3	<pre>&lt;param&gt; &lt;param-name&gt;add.jvm.arg&lt;/param-name&gt; &lt;param-value&gt;-Djvm.userprf.LogLevel=class&lt;/param-value&gt; &lt;/param&gt;</pre>	ユーザ拡張性能解析トレースのトレース出力レベルを指定します。

また、ユーザ拡張性能解析トレースの設定をした場合、性能解析トレースの出力量が非常に多くなります。ディスク容量に余裕があるようでしたら、性能解析トレースの面数/サイズを可能な限り大きくすることを推奨します。設定内容については、Cosminexus V9 アプリケーションサーバ 機能解説 保守／移行編 「7.5.1 性能解析トレースを使用するための設定」を参照してください。

## 5.2 XML 処理性能の改善

XML 解析時のメモリ使用量や処理性能が改善するパラメタを次に示します。

表 5.2-1 XML 処理性能設計項目一覧

#	設計項目	設計内容
①	DOM パーサプール機能	<p>DOM パーサの初期割り当て数と最大プール数を設計します。XML パーサプールへ保存した XML パーサは、システム全体で共有し、再利用できるため、処理の効率向上が図れます。</p> <p>次の関係を満たすように設計してください。</p> <p>初期割り当て数 ≤ 最大プール数</p> <p>最大プール数 ≤ リクエストの最大同時実行数 + 応答アクティビティ後のアクティビティ実行に使用する最大スレッド数<sup>*1</sup></p>
②	SAX パーサプール機能	<p>SAX パーサの初期割り当て数と最大プール数を設計します。DOM パーサと同様の理由で処理の効率向上が図れます。</p> <p>次の処理を実行する場合、SAX パーサのプールオプションを指定します。</p> <ul style="list-style-type: none"> <li>・ビジネスプロセスの検証アクティビティ処理</li> <li>・データ検証機能 (HCSC サーバランタイム定義ファイルで telegram-validation プロパティに ON を指定)</li> </ul> <p>次の関係を満たすように設計してください。</p> <p>初期割り当て数 ≤ 最大プール数</p> <p>&lt;データ検証機能を使用する場合&gt;</p> <p>最大プール数 ≤ 各ビジネスプロセス・サービスアダプタの最大同時実行数(多重度)の総和 + 応答アクティビティ後のアクティビティ実行に使用する最大スレッド数<sup>*1</sup></p> <p>&lt;データ検証機能を使用しない場合&gt;</p> <p>最大プール数 ≤ 各ビジネスプロセスの検証アクティビティの最大同時実行数の総和 + 応答アクティビティ後の検証アクティビティ実行に使用する最大スレッド数<sup>*1</sup></p>
③	フォーマット定義キャッシュ数	<p>HCSC サーバで使用するフォーマット定義は、次に示す数の総和になります。なお、any 型は対象外です。</p> <ul style="list-style-type: none"> <li>・それぞれのビジネスプロセスで定義している全メッセージ型変数の数</li> <li>・それぞれのユーザ定義受付で定義している全要求電文フォーマットの数</li> <li>・それぞれのユーザ定義受付で定義している全応答電文フォーマットの数</li> <li>・それぞれのサービスアダプタで定義している全要求電文フォーマットの数</li> </ul>

		<ul style="list-style-type: none"> <li>それぞれのサービスアダプタで定義している全応答電文フォーマットの数</li> </ul>
④	データ変換定義キャッシュ数	<p>サーバ単位にデータ変換定義をキャッシュできる数を設計します。</p> <p>データ変換定義が再利用されるためには、定義したデータ変換の数以上の数を指定します。</p> <p>定義したデータ変換の数以上の値を指定した場合でも、必要以上のキャッシュを生成することはありません。</p>

注※1 HCSC サーバランタイム定義ファイルの「bp-reply-after-max-thread-pool-size」に設定した値です。

表 5.2-2 パラメータ一覧

#	パラメータ	定義ファイル	定義項目(初期値)	定義範囲
1	①DOM パーサプール機能の初期割り当て数(※1)	簡易構築定義ファイル J2EE サーバのコンフィグレーション定義	<param-name>ex.properties</param-name> <param-value>csc.dt.pool.DocumentBuilder.initialSize=0</param-value>	0-2147483647
2		HCSC サーバランタイム定義ファイル	xml-pooled-instance-minimum=0	0-2147483647
3	①DOM パーサプール機能の最大プール数(※1)	簡易構築定義ファイル J2EE サーバのコンフィグレーション定義	<param-name>ex.properties</param-name> <param-value>csc.dt.pool.DocumentBuilder.maxSize=64</param-value>	0-2147483647
4		HCSC サーバランタイム定義ファイル	xml-pooled-instance-maximum=64	0-2147483647
5	②SAX パーサプール機能の初期割り当て数	簡易構築定義ファイル J2EE サーバのコンフィグレーション定義	<param-name>ex.properties</param-name> <param-value>csc.dt.pool.XMLReader.initialSize=0</param-value>	0-2147483647
6	②SAX パーサプール機能の最大プール数		<param-name>ex.properties</param-name> <param-value>csc.dt.pool.XMLReader.maxSize=64</param-value>	0-2147483647
7	③フォーマット定義をキャッシュできる数	HCSC サーバランタイム定義ファイル	formatdef-maxcache-num=16	1-2147483647



8	④サーバ単位にデータ変換定義をキャッシュできる数		transformdef-maxcache-num=16	1-2147483647
---	--------------------------	--	------------------------------	--------------

注※1：2つの定義ファイルには、同じ値を設定してください。

## 5.3 ファイルディスクリプタの設定（※Linux 限定）

---

サービスプラットフォームで必要となるファイルディスクリプタ数を計算し、ファイルディスクリプタの上限として設定します。  
ここでは、サービスプラットフォームの最大ファイルディスクリプタ数の固定値を 46 とします。

### 5.3.1 FTP 受付・FTP アダプタを使用する場合

FTP 受付・FTP アダプタを使用する際のファイルディスクリプタ数の計算式を次に示します。

計算式から導いた値を、ファイルディスクリプタ数の上限として設定します。

$$\text{最大ファイルディスクリプタ数} = (\text{J2EE サーバの最大ファイルディスクリプタ数}^{\ast 1} + 46 + A + B + C + D) \times 1.25$$

(凡例)

A: FTP インバウンドアダプタを使用する場合は 10、使用しない場合は 0

B: FTP 受付への同時リクエスト数 × 4

C: ファイル操作アダプタの数

D: FTP アダプタの数 × 2

注※1 J2EE サーバの最大ファイルディスクリプタ数については、マニュアル「Cosminexus V9 アプリケーションサーバ システム設計ガイド」を参照してください。

# 6 チューニングパラメタの設定

この章では、性能チューニングしながら詳細を決めていくパラメタについて説明します。

## 本章の構成

- 6.1 JavaVM メモリ
- 6.2 流量制御
- 6.3 タイムアウト
- 6.4 リトライ

## 6.1 JavaVM メモリ

### (1) 設計指針

#### ●Java ヒープサイズ

Java ヒープは、HCSC サーバ及び HCSC サーバ上で動作するビジネスプロセスの各種オブジェクトを格納するメモリ領域です。Java ヒープサイズによって GC の発生間隔・GC にかかる時間が変動し、性能・信頼性に影響します。

Java ヒープサイズには、サイジング時に決定したマシン 1 台あたりの Java ヒープサイズを設定します。また、Java ヒープサイズには初期値と最大値に同じ値を設定し、HCSC サーバ起動時に必要なメモリ領域を確保するようにします。

#### ●Metaspace サイズ

Metaspace は、HCSC サーバ及び HCSC サーバ上で動作するビジネスプロセスのクラスやメソッドなどのメタデータを格納する領域です。そのサイズは、およそ JavaVM にロードされるクラスファイルの合計サイズになります。Metaspace が足りなくなると、OutOfMemoryError が発生します。なお、圧縮オブジェクトポイント機能が有効な場合は、Metaspace 内に Compressed Class Space という領域が作成されます。

ロードされるクラスファイルの合計サイズから Metaspace サイズを見積もることはできますが、多くの場合、すべてのクラスがロードされることはなく、初期値で足りません。そのため、検証のフェーズですべての業務を実施し、足りなければ増やす方法を取ります。また、Metaspace サイズには初期値と最大値に同じ値を設定し、HCSC サーバ起動時に必要なメモリ領域を確保するようにします。

Metaspace サイズが不足して FullGC が発生し、その後 OutOfMemoryError で HCSC サーバが異常終了した場合の拡張 GC ログの例を次に示します。

ログファイルパス (J2EE サーバの名称が J2EEServer の場合) : /opt/Cosminexus/CC/server/public/ejb/J2EEServer/logs /javalog[nn].log

```
[VGC]<Wed Jul 30 14:26:35.896 2014>[Full GC 99908K->81920K(1520448K), 0.7187500 secs][DefNew::Eden: 18270K->0K(419456K)][DefNew::Survivor: 0K->0K(52416K)][Tenured: 81637K->81920K(1048576K)][Metaspace: 262143K->262143K(262144K)][cause*1:MetaspaceAllocFail*2][User: 0.7187500 secs][Sys: 0.0000000 secs][IM: 61412K, 92064K, 0K][TC: 164][DOE: 1K, 5][CCI: 5872K, 49152K, 5952K]
[VGC]<Wed Jul 30 14:26:36.630 2014>[Full GC 81920K->81920K(1520448K), 0.7197266 secs][DefNew::Eden: 0K->0K(419456K)][DefNew::Survivor: 0K->0K(52416K)][Tenured: 81920K->81920K(1048576K)][Metaspace: 262143K->262143K(262144K)][cause*1:LastMetaspaceGC*3][User: 0.7031250 secs][Sys: 0.0000000 secs][IM: 61412K, 92064K, 0K][TC: 164][DOE: 1K, 5][CCI: 5872K, 49152K, 5952K]
[OOM][Thread: 0x00000000d051800]<Wed Jul 30 14:26:37.364 2014>[java.lang.OutOfMemoryError : requested 136 bytes. (Meta Space) : 164 threads exist]
```

注※1 VGC で始まる行の cause は、GC の要因を示しています。

注※2 MetaspaceAllocFail : Metaspace の割り付け失敗によって GC が発生したことを示します。

注※3 LastMetaspaceGC : Metaspace の OutOfMemory を出す前の最後の GC が発生したことを示します。

## (2) パラメータ一覧

JavaVM メモリのパラメータは、構築時に使用する簡易構築定義ファイルに設定します。また、簡易構築定義ファイルで設定値を変更できます。

パラメータ一覧を次に示します。

表 6.1-1 JavaVM メモリ パラメータ一覧

#	パラメータ	定義ファイル	定義項目 (初期値)	内容
1	Java ヒープの初期サイズ	簡易構築定義ファイル	<pre>&lt;param&gt; &lt;param-name&gt;add.jvm.arg&lt;/param-name&gt; &lt;param-value&gt;-Xms256m&lt;/param-value&gt; &lt;/param&gt;</pre>	256m は 256MB を表します。
2	Java ヒープの最大サイズ		<pre>&lt;param&gt; &lt;param-name&gt;add.jvm.arg&lt;/param-name&gt; &lt;param-value&gt;-Xmx512m&lt;/param-value&gt; &lt;/param&gt;</pre>	512m は 512MB を表します。
3	Metaspace の初期サイズ※1		<pre>&lt;param&gt; &lt;param-name&gt;add.jvm.arg&lt;/param-name&gt; &lt;param-value&gt;-XX:MetaspaceSize=128m&lt;/param-value&gt; &lt;/param&gt;</pre>	128m は 128MB を表します。
4	Metaspace の最大サイズ※1		<pre>&lt;param&gt; &lt;param-name&gt;add.jvm.arg&lt;/param-name&gt; &lt;param-value&gt;-XX:MaxMetaspaceSize=128m&lt;/param-value&gt; &lt;/param&gt;</pre>	128m は 128MB を表します。

注※1 データ変換定義のキャッシュ数は Metaspace のサイズに影響を及ぼします。1 キャッシュあたりの Metaspace 使用量は、データ変換定義によって変わるため見積もり式を提示することは難しいです。実測によりチューニングして設定する必要があります。

## 6.2 流量制御

---

システムは要件どおりリクエストを処理できるようにサイジングしています。しかし、実際の運用時には想定外のリクエストが来ることもあり、マシンの処理能力を超えると、安定したビジネスプロセス(サービス)を提供できなくなります。

流量制御パラメタはシステムで処理するリクエストの数を制御するパラメタです。このパラメタの設定を行い、想定外の負荷状態においてもシステムが安定稼働するようにします。

## 6.2.1 SOAP、HTTP 連携における流量制御

ユーザ定義受付(SOAP 受付、HTTP 受付)を前提として記載します。

### (1) 設計パラメタ

処理の流れと流量制御パラメタの一覧を以下に示します。図中の①～④は設計する順番です。

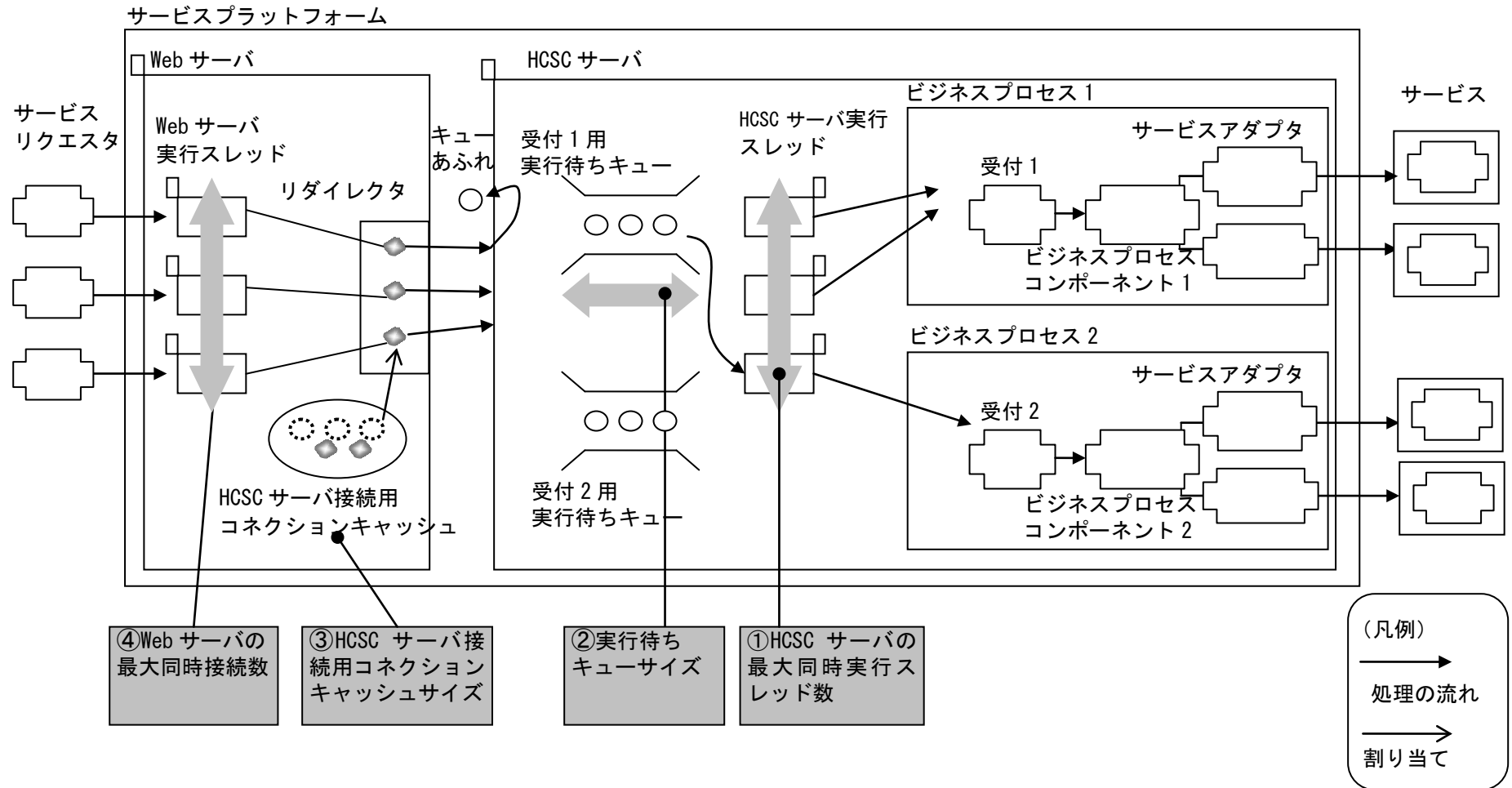


図 6.2-1 処理の流れと流量制御パラメタ一覧

HCSC サーバの最大同時実行スレッド数 (①) を設定することで、処理するリクエストを要件どおりに制限します。それを超える想定外のリクエストは、ある程度は実行待ちキュー (②) にいったん格納することで対処しますが、キューもあふれるような場合は、HTTP ステータスコード 503(Service Unavailable)を返しシステムが混雑していることをサービスリクエストに通知します。その他のパラメタ (③、④) は、HCSC サーバの最大同時実行スレッド数 (①)、実行待ちキューサイズ (②) 分のリクエストが滞りなく処理されるように設定します。

## (2) 設計指針

### ①HCSC サーバの最大同時実行スレッド数

HCSC サーバは、Web サーバから転送されたリクエストを受け取ると、1 リクエストに対し 1 スレッドを割り当てて処理を実行します。マシンの処理性能を超えたスレッド数で処理を行うと、動作が不安定になります。

HCSC サーバの最大同時実行スレッド数は、HCSC サーバで同時に動作するスレッド数の上限を設定するパラメタです。このパラメタに、サイジング時に想定した処理性能を設定することで、想定外の負荷状態においてもシステムが安定稼働するようにします。

サイジング時に想定した処理性能から HCSC サーバの最大同時実行スレッド数を算出する式を以下に示します。

$$\text{HCSC サーバの最大同時実行スレッド数} = \text{HCSC サーバの 1 秒あたりのリクエスト数(件/秒)} \times \text{内部保留時間(秒)}$$

$$\text{※HCSC サーバの 1 秒あたりのリクエスト数(件/秒)} \equiv \text{1 秒あたりのリクエスト数(件/秒)} / \text{マシン台数(台)}$$

上式の考え方を図に示します。

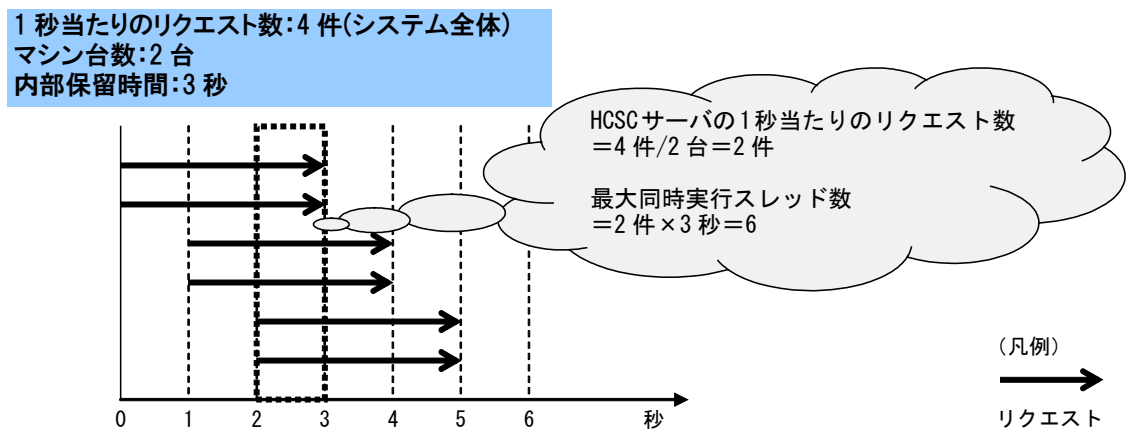


図 6.2-2 HCSC サーバの最大同時実行スレッド数の考え方



## ②実行待ちキューサイズ

要件どおりのリクエストであればエラー応答することなく処理できるように①で設計しましたが、実際の運用時には突発的なリクエスト<sup>2</sup>や、想定外のリクエスト<sup>3</sup>が来ることもあります。このような場合に、エラーが発生するのを防ぐため、リクエストを実行待ちキューで待たせることができます。

実行待ちキューに入ったリクエストは、レスポンスは遅くなりますが、エラーにしないで処理できます。

実行待ちキューが上限に達した後で到着したリクエストは、エラー応答し、混雑していることを通知できます。

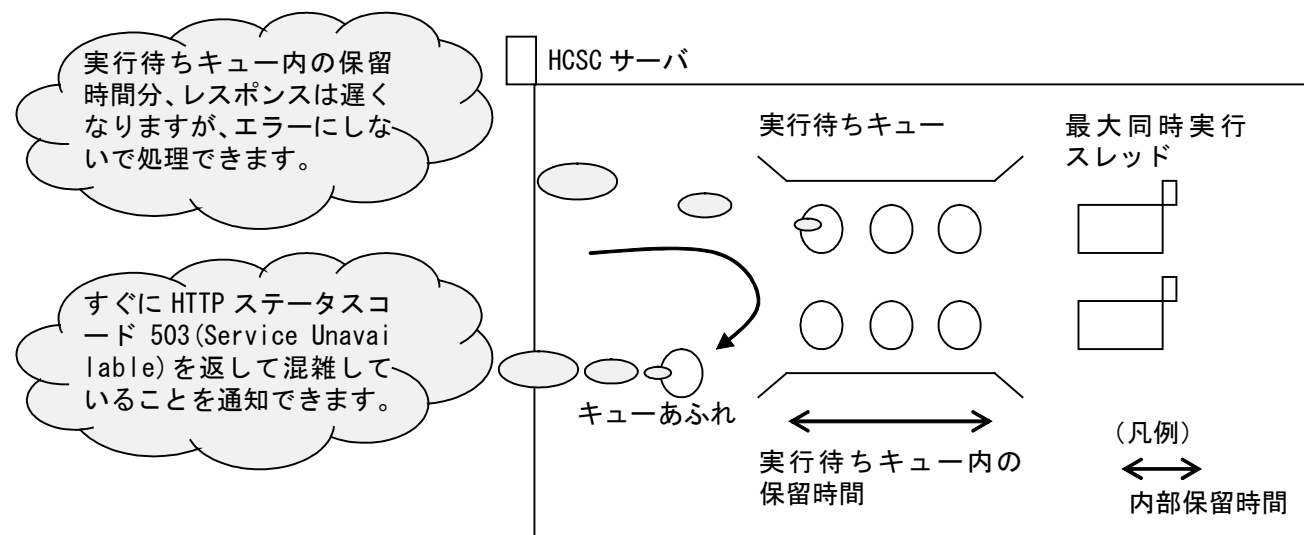


図 6.2-3 実行待ちキューサイズの解説

このため、突発的に来たリクエストに対して、正常に処理したいリクエストの数をキューサイズとして設計します。ただし、キューサイズを大きくしすぎると、エラー応答が遅くなるため、どこまでのリクエストを正常処理して、どこからをエラー応答とするのが、チューニングのポイントになります。

実行待ちキューサイズ=突発的に来たリクエストに対して正常に処理したいリクエストの数

<sup>2</sup> 突発的なリクエストとは、ごく短い時間にリクエストが集中した場合のリクエストのことを言います。全体的にみれば、リクエストの総量は変わりません。

<sup>3</sup> 想定以上のリクエストとは、長時間にわたりリクエストが集中した場合のリクエストのことを言います。全体的に、リクエストの総量が増加します。

### ③HCSC サーバ接続用コネクションキャッシュサイズ(Windows のみ)

リダイレクタはリクエスト転送時に生成した HCSC サーバとのコネクションをキャッシュすることで、性能の向上を図ります。

HCSC サーバ接続用コネクションキャッシュサイズは、キャッシュするコネクションの最大数を設定するパラメタです (リダイレクタが使用するコネクションの上限ではありません)。このパラメタには、HCSC サーバとの最大接続数を設定し、すべてのリクエストがすぐにコネクションを利用出来るようにします。

HCSC サーバ接続用コネクションキャッシュサイズを算出する式を以下に示します。

$$\text{HCSC サーバ接続用コネクションキャッシュサイズ} = \text{HCSC サーバ最大接続数} = \text{HCSC サーバの最大同時実行スレッド数} + \text{HCSC サーバ全体の実行待ちキューサイズ}$$

### ④Web サーバの最大同時接続数

Web サーバはサービスリクエスタからリクエストを受け取ると、1 リクエストに対し 1 スレッドを割り当てて処理を実行します。

Web サーバの最大同時接続数は、Web サーバで同時に動作するスレッド数の上限を設定するパラメタです。このパラメタには、HCSC サーバの最大同時実行スレッド数、HCSC サーバ全体の実行待ちキューサイズ分のリクエストが、滞りなく HCSC サーバに到達するのに必要な数を設定します。そうすることで、想定外のリクエストに対しては、実行待ちキューを使用した制御が可能となります。

Web サーバの最大同時接続数を算出する式を以下に示します。

Web サーバの最大同時接続数

= HCSC サーバの最大同時実行スレッド数

+ HCSC サーバ全体の実行待ちキューサイズ

+ 1

…キューあふれによる、HTTP ステータスコード 503(Service Unavailable)応答用スレッド

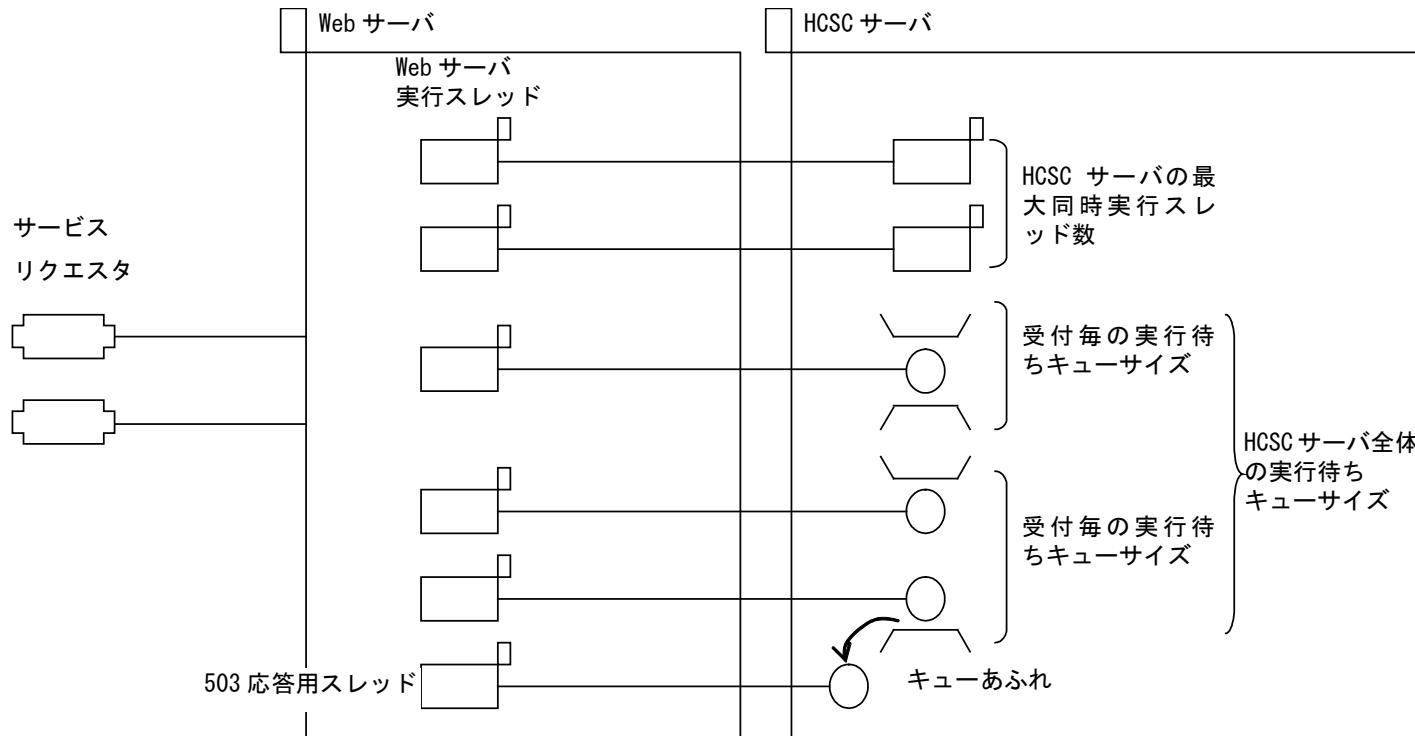


図 6.2-4 Web サーバの最大同時接続数の考え方

### (3) パラメータ一覧

SOAP 連携の場合、受付はユーザ定義受付になります。実行待ちキューサイズは#5 に設定してください。

HTTP 連携の場合、受付は HTTP 受付になります。実行待ちキューサイズは#6 に設定してください。

表 6.2-1 流量制御パラメータ一覧

#	パラメータ	定義ファイル	定義項目(初期値)	定義範囲
1	④ Web サーバの最大同時接続数	簡易構築定義ファイル	(UNIX の場合) <param> <param-name>MaxClients</param-name> <param-value>150</param-value> </param>	1-1024
2			(Windows の場合) <param> <param-name>ThreadsPerChild</param-name> <param-value>50</param-value> </param>	1-1024
3	① HCSC サーバの最大同時実行スレッド数	ユーザ定義受付ランタイム定義ファイル ※1(cscurecptnconfig.properties)	<param> <param-name>webserver.connector.ajp13.max_threads</param-name> <param-value>10</param-value> </param>	1-1024 指定されたリクエストの同時実行数分のスレッドがサーバ起動時に生成されます。
4	③ HCSC サーバ接続用コネクションキャッシュサイズ(Windows のみ)		<param> <param-name>worker.<ワーカー名>.cachesize</param-name> <param-value>64</param-value> </param>	1-2147483647
5	② 実行待ちキューのサイズ <ユーザ定義受付>	ユーザ定義受付ランタイム定義ファイル ※1(cscurecptnconfig.properties)	user-defined-reception-soap.queue-size=8192	0-2147483647
6	② 実行待ちキューのサイズ <HTTP 受付>	HTTP 受付定義ファイル ※2 (cscurecphttp.properties)	urecp-http.queue-size=8192	0-2147483647

注※1 ユーザ定義受付ランタイム定義ファイルにの格納先は任意で、csccompoconfig コマンドの引数に指定して使用します。

注※2 HTTP 受付定義ファイルは、HTTP 受付の開発時にユーザ定義受付定義画面(詳細)の[独自定義ファイル]で編集します。

## 6.2.2 ビジネスプロセス単位の流量制御

前項では、HCSC サーバの実行スレッド数および実行待ちキューを設定し、サービスプラットフォームの安定動作を図りました。しかし、ビジネスプロセスの特性によっては、さらに以下のような問題が発生します。

- 大量の CPU やメモリを使用するビジネスプロセスに想定以上のリクエストが集中した場合、CPU やメモリが不足し、安定したビジネスプロセス(サービス)を提供できない
- 時間を要するビジネスプロセスに想定以上のリクエストが集中した場合、そのビジネスプロセスがすべてのスレッドを占有し、他のビジネスプロセスをなかなか実行できない
- 他のビジネスプロセスに想定以上のリクエストが集中した場合、それらのビジネスプロセスがすべてのスレッドを占有し、重要なビジネスプロセスをなかなか実行できない

このような問題が発生する場合、システムでは以下の対処をおこなう必要があります。

- 大量のリソースを使用するビジネスプロセス、時間を要するビジネスプロセスの実行数を制限
- 重要なビジネスプロセスの実行数を確保

受付単位の流量制御パラメタによって、ビジネスプロセス単位で処理するリクエストの数を制御できます。このパラメタの設定を行い、想定以上の負荷状態においてもシステムが安定稼働するようにします。

## (1) 設計パラメタ

受付単位の流量制御パラメタには、受付単位の最大同時実行スレッド数、および受付単位の占有スレッド数があります。最大同時実行スレッド数は大量のリソースを使用するビジネスプロセス、時間を要するビジネスプロセスの実行数を制限するため、占有スレッド数は重要なビジネスプロセスの実行数を確保するために使用します。

受付単位の流量制御に関連するパラメタの一覧を以下に示します。

### 最大同時実行スレッド

●HCSC サーバの最大同時実行スレッド数  
HCSC サーバで同時に実行できるスレッドの上限。図は6を設定した場合。(前項で設定)

① 受付単位の最大同時実行スレッド数  
ある受付を同時に実行できるスレッドの上限。図は、受付1に3を設定した場合で、スレッド3やスレッド5が空いても、受付1は実行できず、受付1へのリクエストは、受付1用の実行待ちキューにたまる。

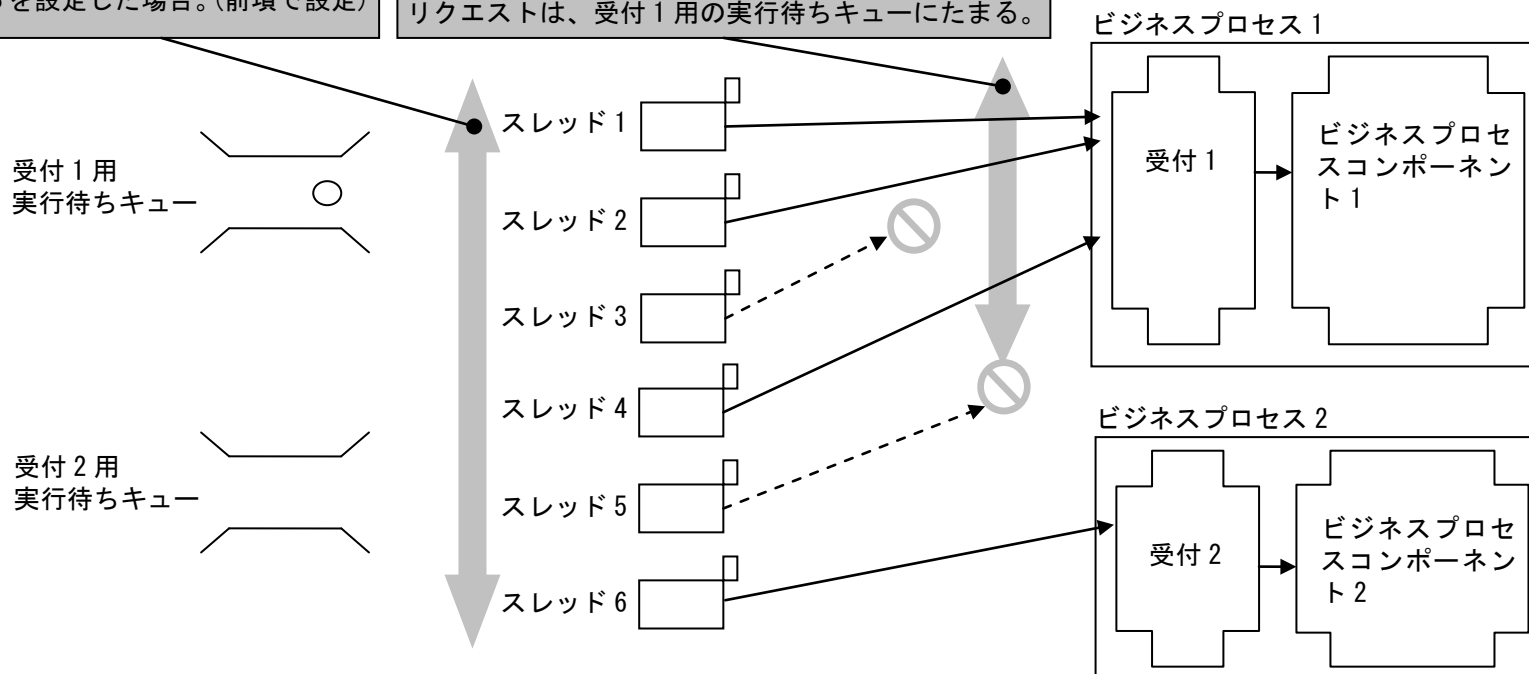


図 6.2-5 受付単位の最大同時実行スレッドの考え方

## 占有スレッド

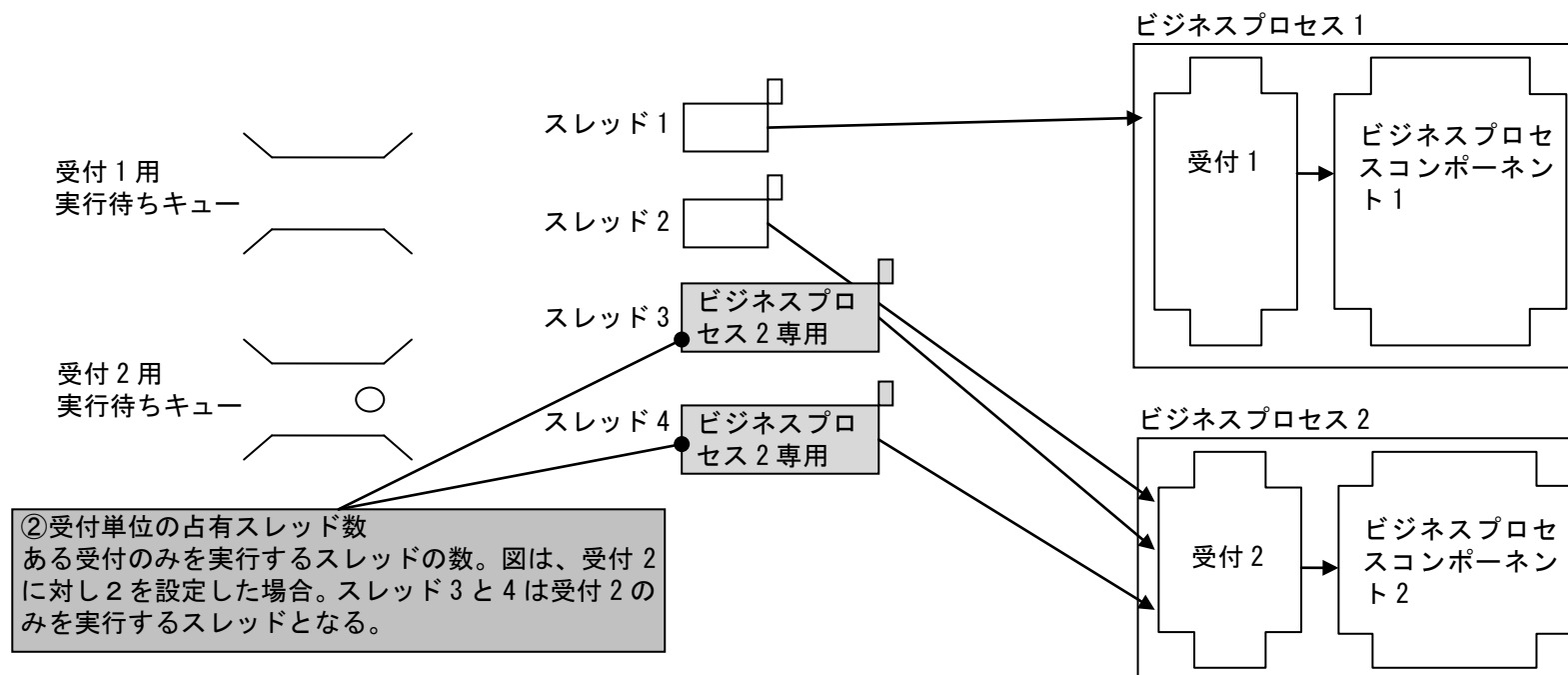


図 6.2-6 受付単位の占有スレッドの考え方

## (2) 設計指針

### ①受付単位の最大同時実行スレッド数

ビジネスプロセスの内部保留時間または、1 秒当たりのリクエスト数が同じビジネスプロセスごとに設計します。値は、制限を行うビジネスプロセスの要件を満たすだけの実行数で、「HCSC サーバの最大同時実行スレッド数」と考え方は同様です。

受付単位の最大同時実行スレッド数 = ビジネスプロセスの HCSC サーバの 1 秒あたりのリクエスト数(件/秒)

× ビジネスプロセスの内部保留時間(秒)

※ ビジネスプロセスの HCSC サーバの 1 秒あたりのリクエスト数(件/秒) ≡ ビジネスプロセスの 1 秒あたりのリクエスト数(件/秒) / マシン台数(台)

②受付単位の占有スレッド数

実行数を確保する必要がある、特に重要なビジネスプロセスがある場合に設定します。値は、必要最小限の実行数にします。

受付単位の占有スレッド数=必要最小限の実行数

(3) パラメター一覧

使用する受付に合わせて、最大同時実行スレッド数と占有スレッド数を指定してください。

表 6.2-2 流量制御パラメター一覧

#	受付種別	パラメタ	定義ファイル	定義項目(初期値)	定義範囲
1	ユーザ定義 受付	①最大同時実行数	ユーザ定義受付ランタイム定義ファイル (cscreptnconfig_< 受付 ID>.properties) ※1	user-defined-reception-soap.threads.m aximum=10	1-1024 占有スレッド数以上の値を指定する 必要があります。
2		②占有スレッド数		user-defined-reception-soap.exclusive.t hreads=0	0-1024 最大実行数を超えない範囲で値を 指定する必要があります。0 を指 定した場合、スレッドは占有され ません。
3	HTTP 受付	①最大同時実行スレッド 数	HTTP 受付定義ファイル※2 (cscrecphttp.properties)	urecp-http.max-threads=10	1-1024 占有スレッド数以上の値を指定す る必要があります。
4		②占有実行スレッド数		urecp-http.exclusive-threads=0	0-1024 最大実行数を超えない範囲で値を 指定する必要があります。0 を指 定した場合、スレッドは占有され ません。

注※1 ユーザ定義受付ランタイム定義ファイルにの格納先は任意で、csccompoconfig コマンドの引数に指定して使用します。

注※2 HTTP 受付定義ファイルは、HTTP 受付の開発時にユーザ定義受付定義画面（詳細）の [独自定義ファイル] で編集します。



## 6.2.3 FTP 連携におけるの流量制御

### (1) 設計パラメタ

処理の流れと流量制御パラメタの一覧を以下に示します。

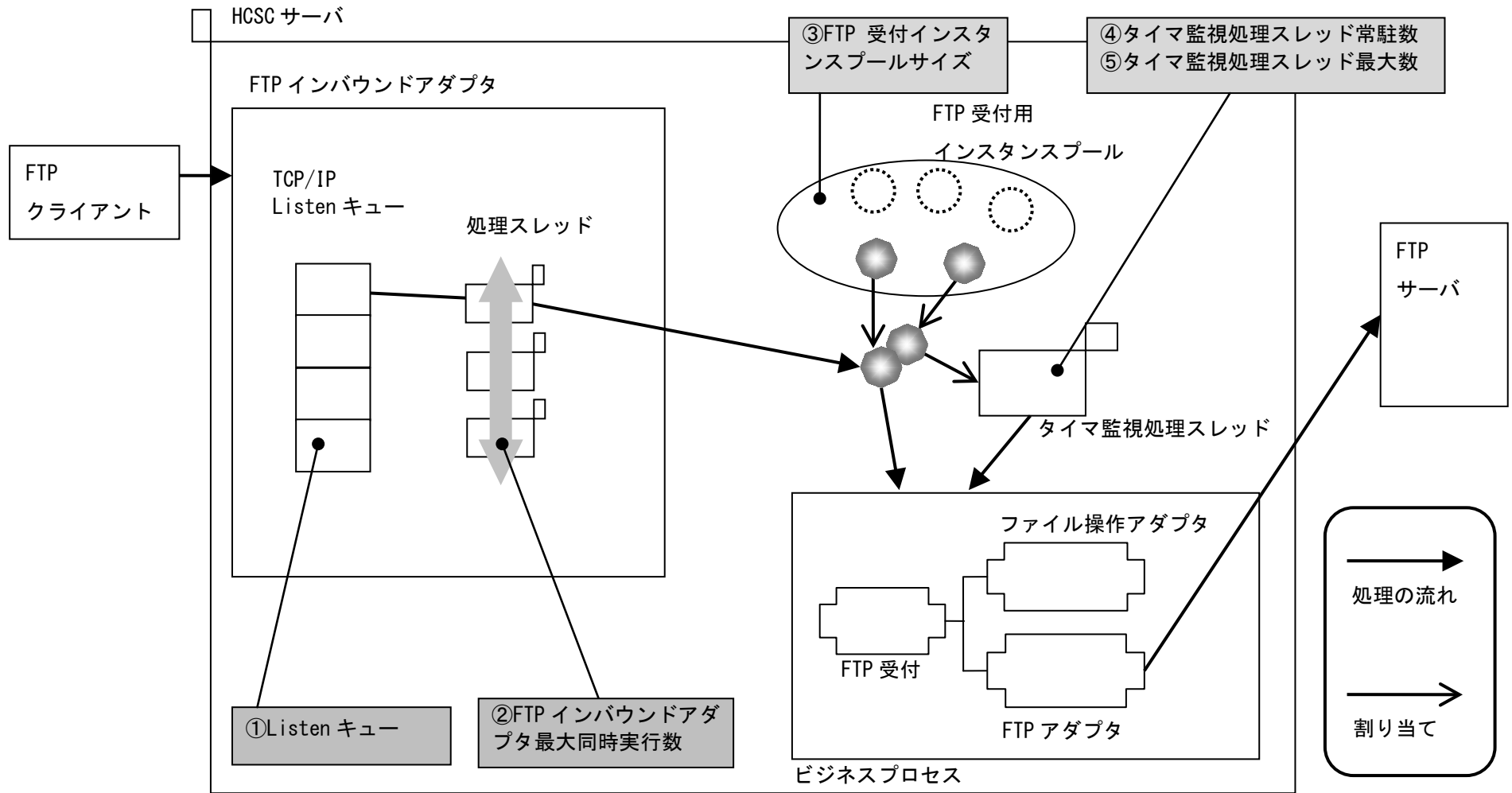


図 6.2-7 処理の流れと流量制御パラメタの関係

表 6.2-3 流量制御パラメタの説明

#	パラメタ	コンポーネント	説明
1	①Listen キュー	FTP インバウンドアダプタ	FTP クライアントからの接続要求の最大の待ち行列数を指定します。 この指定値はシステムコール <code>listen</code> のバックログ数として設定されます。 Listen キューに入れなかった接続は確立されず、キューを超えたリクエストは接続( <code>connect</code> )エラーとなります。
2	②FTP インバウンドアダプタ 最大同時実行数		FTP インバウンドアダプタとして起動するスレッド数を指定します。指定したスレッド数は FTP インバウンドアダプタの最大同時接続数を示します。最大同時接続数を越えたリクエストは、接続が切断されます。
3	③FTP 受付インスタンスプールサイズ	FTP 受付	FTP 受付のインスタンスプールの最小値、最大値を指定します。 インスタンスプールの最大値を超えた場合の、インスタンス生成処理はインスタンスが空くまで待たされます。
4	④FTP 受付タイマ監視処理スレッド常駐数 (※)		FTP 受付後にビジネスプロセスで処理する時間のタイマ監視を行う際に、常駐させるスレッド数を指定します。常駐スレッドは FTP 受付起動時に生成されます。
5	⑤FTP 受付タイマ監視処理スレッド最大数(※)		FTP 受付のタイマ監視機能で利用する最大スレッド数を指定します。

(※) FTP 受付のタイマ監視は FTP 受付コンフィグファイルの `urecp-ftp.request-timeout` の設定を 0 にすることで停止可能です。(デフォルトはタイマ監視しない)

## (2) 設計指針

### ●FTP インバウンドアダプタの最大同時実行数

HCSC サーバは、FTP インバウンドアダプタで設定した最大同時実行数分のリクエストを同時に処理することができます。マシンの処理性能を超えた同時実行数で処理を行うと、動作が不安定になります。

FTP インバウンドアダプタの最大同時実行数分に、サイジング時に想定した処理性能を設定することで、想定外の負荷状態においてもシステムが安定稼働するようにします。

サイジング時に想定した処理性能から FTP インバウンドアダプタの最大同時実行数を算出する式を以下に示します。

$$\text{FTP インバウンドアダプタの最大同時実行数} = 1 \text{ 秒あたりの FTP リクエスト数(件/秒)} \times \text{内部保留時間(秒)}$$

(3) パラメータ一覧

表 6.2-4 流量制御パラメータ一覧

#	パラメータ	定義ファイル※1	定義項目(初期値)	定義範囲
1	①Listen キュー	FTP インバウンドアダプタ 属性定義ファイル	<config-property-name>nioListener_listenBacklog</config-property-name> <config-property-type>java.lang.String</config-property-type> <config-property-value>50</config-property-value>	1- 2147483647
2	②FTP インバウンドアダプタ 最大同時実行数		<config-property-name>server_maxLogins</config-property-name> <config-property-type>java.lang.String</config-property-type> <config-property-value>10</config-property-value>	1-1024
3	③FTP 受付インスタンスプールの最大値	FTP 受付定義ファイル	urecp-ftp.pooled-instance.maximum=0※2	0- 2147483647
4	③FTP 受付インスタンスプールの最小値		urecp-ftp.pooled-instance.minimum=0	0- 2147483647
5	④FTP 受付タイマ監視用常駐スレッド数	FTP 受付コンフィグファイル	urecp-ftp.resident-thread.count=0	0- 2147483647
6	⑤FTP 受付タイマ監視用最大スレッド数		urecp-ftp.timer-thread.maximum= FTP 受付定義ファイルの urecp-ftp.pooled-instance.maximum の値	0- 2147483647

注※1 定義ファイルの詳細は、表 4.2-1 定義ファイル一覧参照。

注※2 0 を指定した場合、受付処理およびカスタム受付フレームワークのプール内のインスタンスの最大数には、1 が設定されます。

## 6.3 タイムアウト

---

HCSC サーバ及びサービスで、処理遅延、通信遅延の障害が発生した場合、以下のような問題が発生します。

- ビジネスプロセス処理が滞留し、そのビジネスプロセス処理がスレッドなどのリソースを占有し、他のビジネスプロセス処理に影響を与える
- サービスリクエストを不必要に待たせる

このような問題が発生する場合、システムでは以下の対処をおこなう必要があります。

- 遅延を検知
- ビジネスプロセス処理が占有しているリソースを解放
- サービスリクエストへのエラー応答

タイムアウトパラメタは、遅延を検知し、リソースの解放、サービスリクエストへのエラー応答をするためのパラメタです。このパラメタの設定を行い、処理遅延や通信遅延の障害時においてもシステムやサービスリクエストへの影響を軽減するようにします。

### 6.3.1 HCSC サーバ内のトランザクションタイムアウト

#### (1) 設計パラメタ

HCSC サーバのトランザクションタイムアウトパラメタの説明を以下に示します。

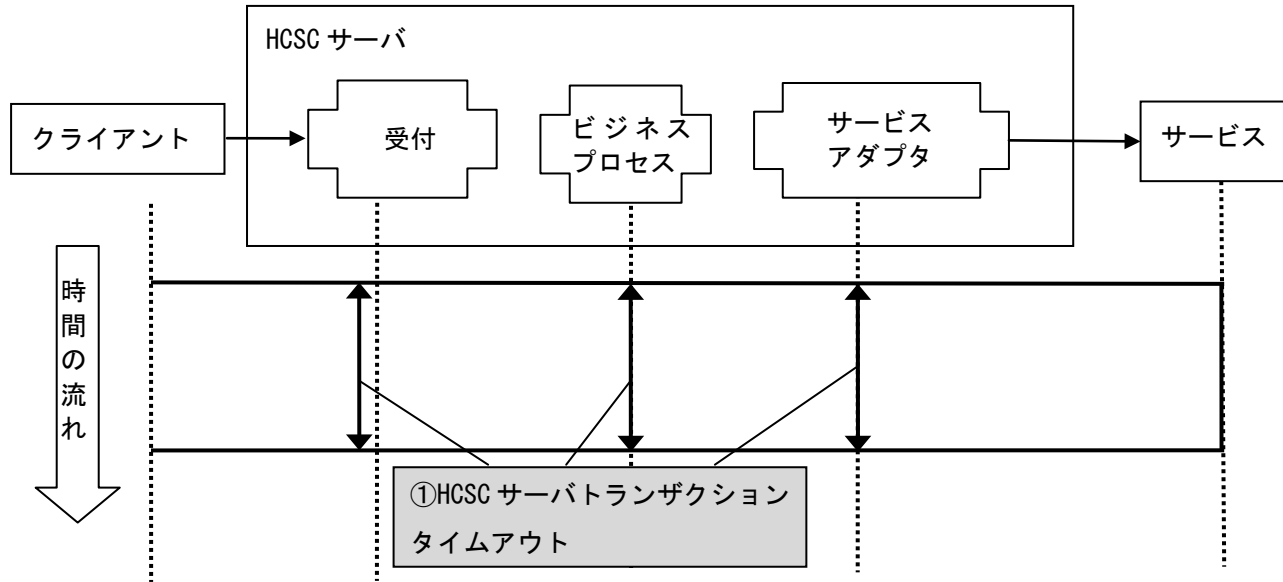


図 6.3-1 HCSC サーバトランザクションタイムアウト

表 6.3-1 トランザクションタイムアウトパラメタの説明

#	パラメタ	説明
1	①HCSC サーバトランザクションタイムアウト	HCSC サーバ稼働マシンのアプリケーションサーバの EJB コンテナでタイムアウトを設定します。

## (2) 設計指針

### ●HCSC サーバトランザクションタイムアウト①

HCSC サーバでは、受付、ビジネスプロセスのアクティビティの処理の間で、トランザクションタイムアウトを設定します。

HCSC サーバトランザクションタイムアウト = HCSC サーバ全体の処理において、遅延障害が発生していると判断する時間。

## (3) パラメータ一覧

表 6.3-2 パラメータ一覧

#	パラメータ	定義ファイル	定義項目(初期値)	定義範囲
1	①HCSC サーバトランザクションタイムアウト	簡易構築定義ファイル	<pre>&lt;param&gt; &lt;param-name&gt;ejbserver.jta.TransactionManager .defaultTimeout&lt;/param-name&gt; &lt;param-value&gt;180&lt;/param-value&gt; &lt;/param&gt;</pre>	1-2147483647(秒)

### 6.3.2 SOAP、HTTP 連携におけるタイムアウト

#### (1) 設計パラメタ

タイムアウトパラメタは2つあります。遅延の検知を複数箇所で行うことで、遅延発生箇所の絞込みが容易になります。以下に、各タイムアウトパラメタの説明と、それぞれのタイムアウト時の動作を示します。

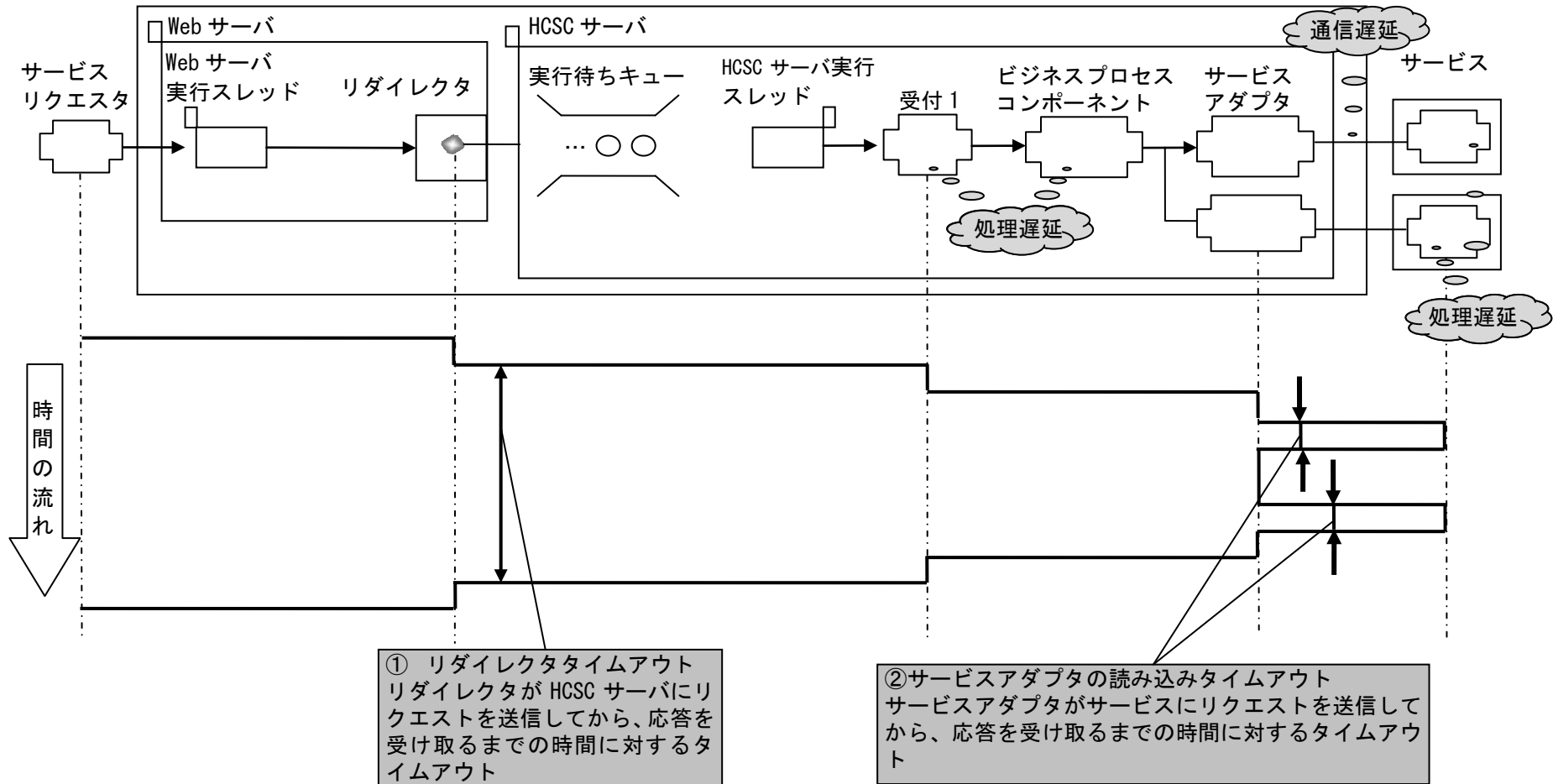


図 6.3-2 処理の流れとタイムアウトパラメタの関係

表 6.3-3 タイムアウト時の動作

#	パラメタ	サービスリクエストへのエラー応答	リソース解放
①	リダイレクタタイムアウト	HTTP ステータスコード 500(Internal Server Error)を返します。	Web サーバの実行スレッド、コネクションを解放します。(ただし、HCSC サーバ内の処理は継続します。)
②	SOAP アダプタのソケット読み込みタイムアウト	SOAP フォルト ( faultcode が ns1 * : CSCMsgServerFaultException) を返します。 ※ns1=http://ejb.reception.message.msg.csc.soft.hitachi.co.jp	HCSC サーバの実行スレッド、コネクションを解放します。(但し、サービス内の処理は継続します。)
	HTTP アダプタの読み込みタイムアウト	HTTP ステータスコード 500*(Internal Server Error)を返します。 ※HTTP 受付定義ファイルの「httprecp.system-exception.status-code」パラメタで 500 以外の応答に変更することができます。	HCSC サーバの実行スレッド、コネクションを解放します。(但し、サービス内の処理は継続します。)

## (2) 設計指針

### サービスリクエストへのエラー応答を目的としたパラメタ

#### ①リダイレクタタイムアウト

サービスリクエストをこれ以上待たせることが出来ないため、リダイレクタ以降(HCSC サーバ、サービス)の処理において、遅延障害が発生していると判断する時間を設定します。

リダイレクタタイムアウト=リダイレクタ以降の処理において遅延障害が発生していると判断する時間

また、実行待ちキューサイズ的设计時に考慮した、「許容できる内部保留時間」よりは大きい値を設定するようにしてください。小さい値を設定すると、キューに格納したリクエストの処理終了を待たずに、リダイレクタでタイムアウトする場合があります。

リダイレクタタイムアウト>許容できる内部保留時間=内部保留時間+実行待ちキュー内の保留時間

### リソースの解放を目的としたパラメタ

#### ②サービスアダプタの読み込みタイムアウト

サービスの処理時間を踏まえ、明らかにサービス実行中に異常な遅延が発生していると判断する時間を設定します。これによって、HCSC サーバのスレッド、コネクションを解放することが出来ます。



サービスアダプタの読み込みタイムアウト=サービス実行中に異常な遅延が発生していると判断する時間

### タイムアウトパラメタ間の関係

遅延発生部位の絞込みを容易にするため、またタイムアウトパラメタ間の矛盾がないように、バックエンドに近づくほどタイムを小さく設定します。

① リダイレクタタイムアウト > ②サービスアダプタの読み込みタイムアウト

### (3) パラメタ一覧

表 6.3-4 タイムアウトパラメタ一覧

#	パラメタ	定義ファイル	定義項目(初期値)	定義範囲
1	①リダイレクタタイムアウト	簡易構築定義ファイル	<code>&lt;param&gt;</code> <code>&lt;param-name&gt;worker.&lt;ワーカー名&gt;.receive_timeout&lt;/param-name&gt;</code> <code>&lt;param-value&gt;3600&lt;/param-value&gt;</code> <code>&lt;/param&gt;</code>	0-3600(秒)。 0を設定した場合は、タイムアウトしません。
2	②サービスアダプタの読み込みタイムアウト	SOAPアダプタ	(SOAP1.1 モードの場合)※2 <code>&lt;クラスタ名&gt;&lt;サービスID&gt;.WebService.c4web.application.socket_read_timeout=300</code>	0-86400 (秒) 0を指定した場合はタイムアウト監視をしません。
3			(SOAP1.1/1.2 併用モードの場合) ※2 <code>&lt;クラスタ名&gt;&lt;サービスID&gt;.WebService.com.cosminexus.csc.request.timeout=300</code>	0-86400 (秒) 0を指定した場合はタイムアウト監視をしません。
4		HTTPアダプタ	HTTPアダプタ定義ファイル	<code>adphttp.protocol.read-timeout=300000</code>

注※1 cscsvectl コマンドの-getfile オプションに指定したパスに格納されます。

※2 使用する SOAP モードに合わせてパラメタを設定してください。

#4 の定義ファイルは、開発環境の HTTP アダプタの開発時にサービスアダプタ定義画面(詳細)の[独自定義ファイル]で編集します。

### 6.3.3 FTP 連携におけるタイムアウト

以下の PP を FTP クライアント、FTP サーバの一例として記載します。

- FTP クライアントは、JP1/FTS/FTP クライアント
- FTP サーバは、JP1/FTS/FTP サーバ

#### (1) 設計パラメタ

タイムアウトパラメタの設計箇所を以下のパターンについて示します。

- (a) 同期連携パターン PUSH 型連携
- (b) 同期連携パターン PULL 型連携
- (c) FTP クライアントと FTP インバウンドアダプタのタイムアウト
- (d) FTP アダプタと FTP サーバのタイムアウト

詳細な設定に関する内容は、マニュアル「サービスプラットフォーム 解説」の「8.5.2 タイムアウトの監視」または「8.6.5 タイマ管理」も合わせてご確認ください。

(a) 同期連携パターン PUSH 型連携の場合

FTP クライアントから送信されたファイルを HCSC サーバで受信し、ファイル操作アダプタでファイル変換を行った後、FTP サーバに送信します。FTP サーバへの送信処理完了を待って、FTP クライアントに応答を返します。

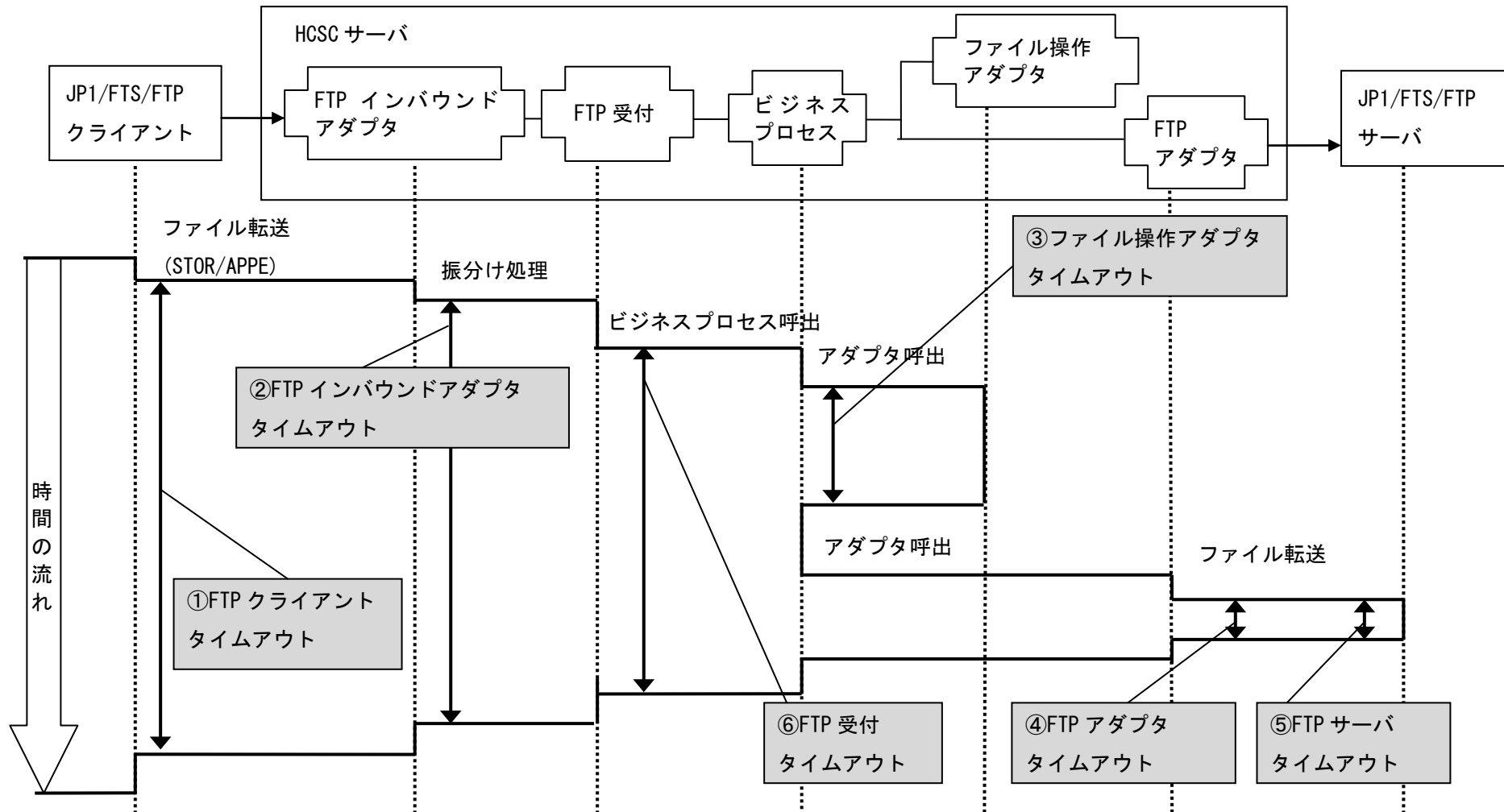


図 6.3-3 同期連携パターン PUSH 連携のタイムアウト

(b) 同期連携パターン PULL 型連携の場合

FTP クライアントからのファイル取得要求を HCSC サーバで受信し、FTP サーバからファイルを取得する。その後、ファイル操作アダプタでファイル変換を行い、FTP クライアントにファイルを送信する。

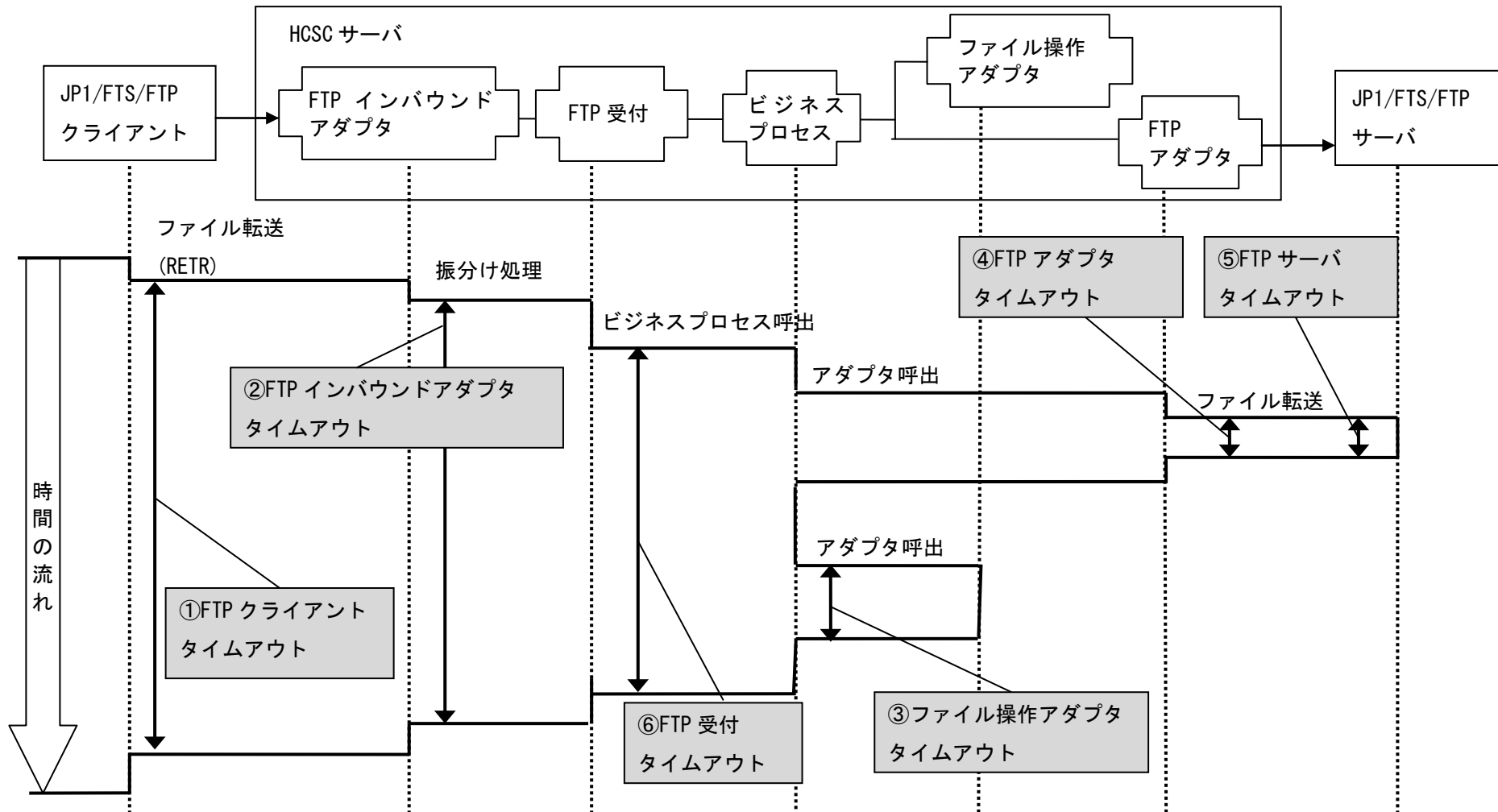


図 6.3-4 同期連携パターン PULL 連携のタイムアウト

表 6.3-5 同期連携パターン PUSH/PULL 連携タイムアウト時の動作

#	パラメタ	コンポーネント	説明	タイムアウト発生時の動作
1	①FTP クライアント タイムアウト	FTP クライアント	FTP クライアント、FTP インバウンドタイムアウトの詳細は、(c) FTP クライアントと FTP インバウンドアダプタのタイムアウト参照	
2	②FTP インバウンドアダプタ タイムアウト	FTP インバウンドアダプタ		
3	③ファイル操作アダプタ タイムアウト	ファイル操作アダプタ	ファイル操作アダプタのファイル操作処理時間に関するタイムアウト。	ファイル操作アダプタが異常終了する。
4	④FTP アダプタ タイムアウト	FTP アダプタ	FTP アダプタ、FTP サーバのタイムアウトの詳細は、(d) FTP アダプタと FTP サーバのタイムアウト参照	
5	⑤FTP サーバ タイムアウト	FTP サーバ		
6	⑥FTP 受付タイムアウト	FTP 受付	FTP 受付がビジネスプロセスに要求を渡した後、FTP 受付に応答が返るまでのタイムアウト。	FTP インバウンドアダプタにエラー応答する。タイムアウトが発生したあともスレッドの処理は継続される。

(c)FTP クライアントと FTP インバウンドアダプタのタイムアウト

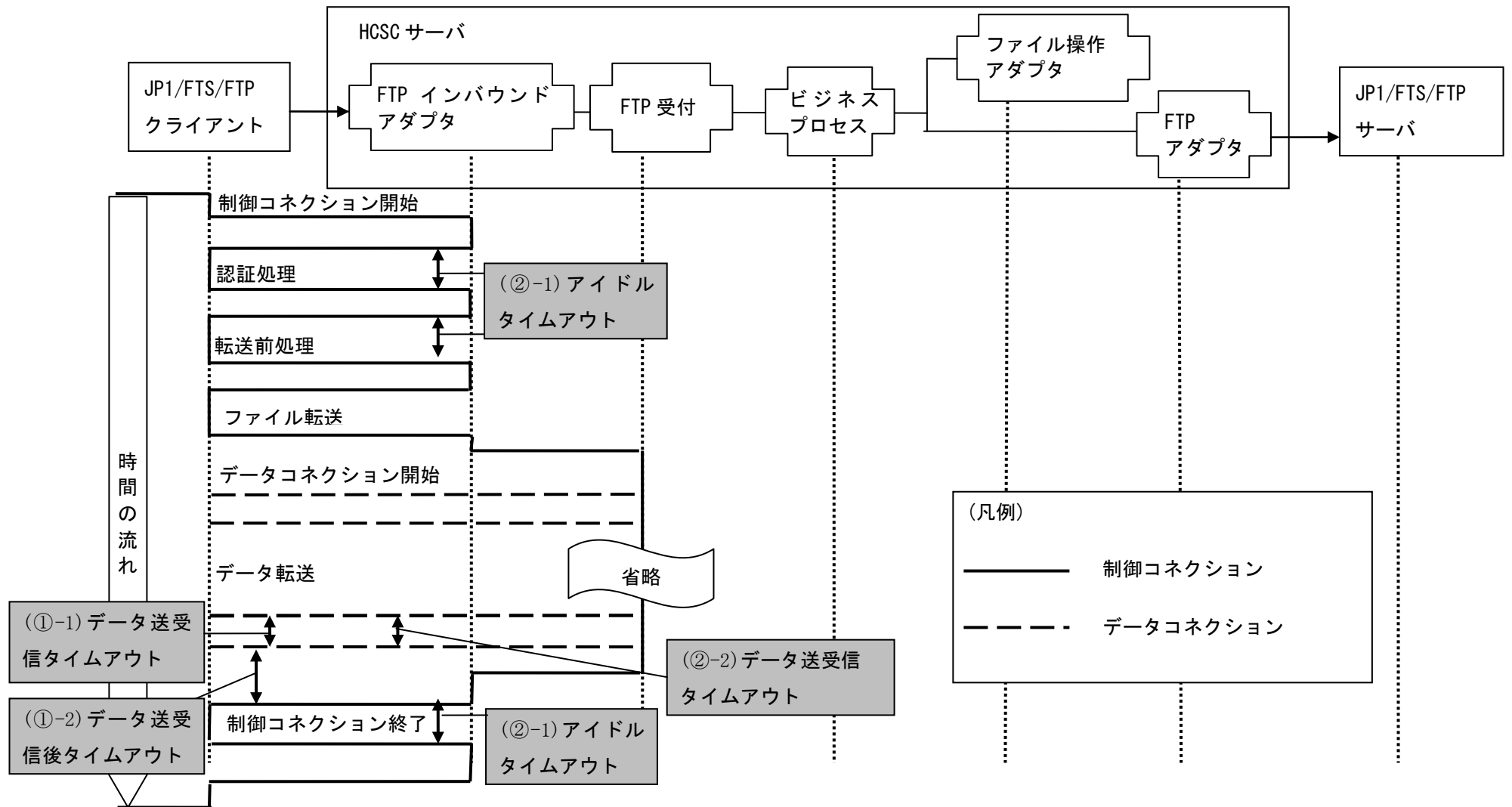


図 6.3-5 FTP クライアントと FTP インバウンドアダプタのタイムアウト

表 6.3-6 FTP クライアントと FTP インバウンドアダプタのタイムアウト時の動作

#	パラメタ	コンポーネント	説明	タイムアウト発生時の動作
1	(②-1)アイドル タイムアウト	FTP インバウンドアダプ タ	制御コネクション確立後、FTP クライアントから FTP コマンドの送信がない場合に、FTP インバウ ンドアダプタ側で FTP クライアントの制御コネ クションを解放する。	FTP クライアントに通信エラーが返る。 ビジネスプロセスは起動していない。
2	(①-1)データ送受信 タイムアウト	FTP クライアント	ファイル転送が STOR/APPE の場合、データ送信 タイムアウトが使用される。 ファイル転送が RETR の場合、データ受信タイム アウトが使用される。 データ送受信時に、パケットを送受信するまでの タイムアウト時間。	FTP クライアントが異常終了する。 ビジネスプロセスは、FTP 受付でエラーが 発生する。
3	(②-2)データ送受信 タイムアウト	FTP インバウンドアダプ タ	データ送受信時に、パケットを送受信するまでの タイムアウト時間。	FTP クライアントに通信エラーが返る。 ビジネスプロセスは、FTP 受付でエラーが 発生する。
4	(①-2)データ送受信 後タイムアウト	FTP クライアント	データ送受信後に、FTP インバウンドアダプタか らの応答メッセージを受信するまでのタイムアウ ト時間。	FTP クライアントに通信エラーが返る。 ビジネスプロセスは実行を続ける。

(d) FTP アダプタと FTP サーバのタイムアウト

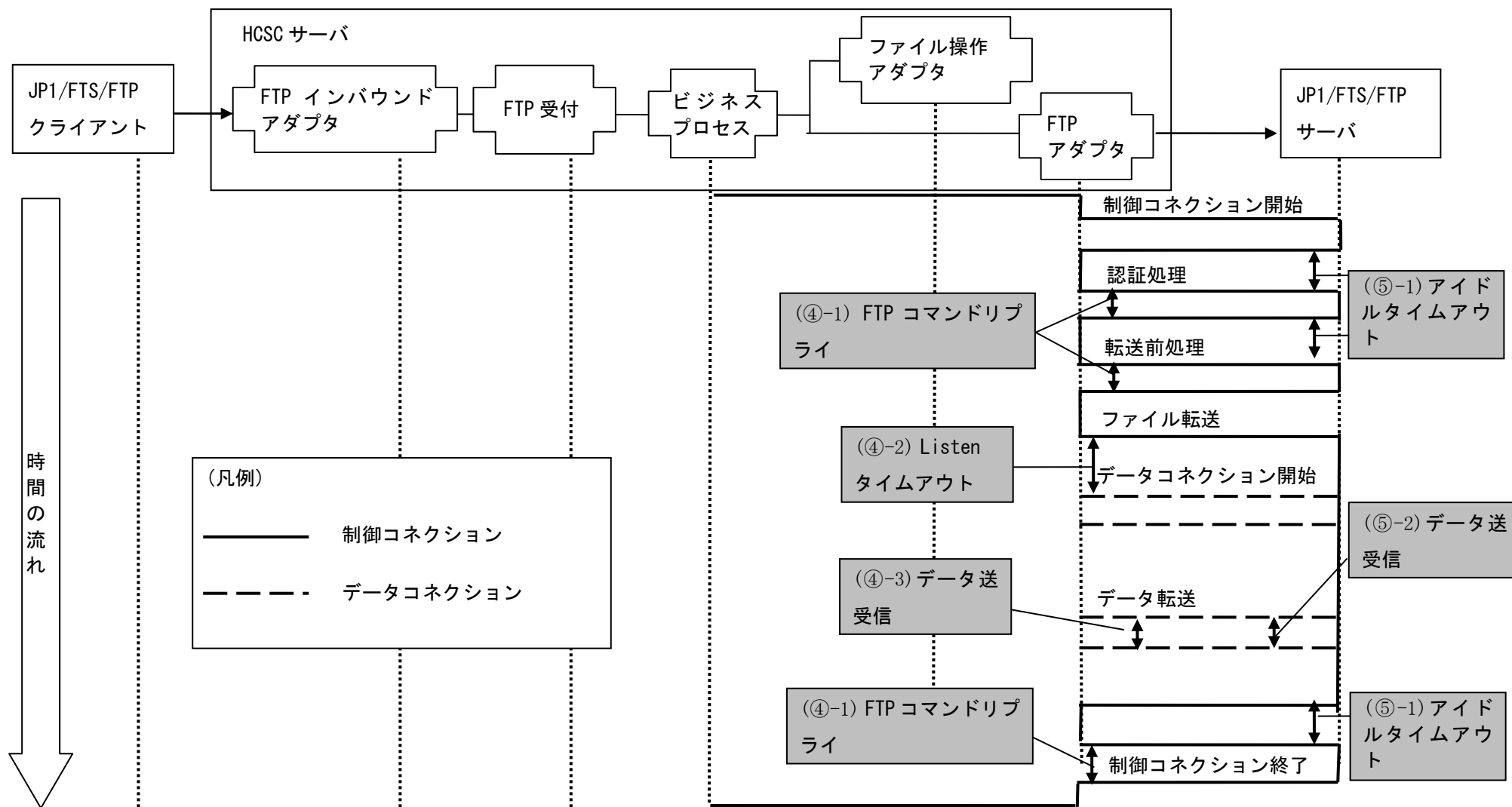


図 6.3-6 FTP アダプタと FTP サーバのタイムアウト



表 6.3-7 FTP アダプタと FTP サーバのタイムアウト時の動作

#	パラメタ	コンポーネント	説明	タイムアウト発生時の動作
1	(⑤-1)アイドル タイムアウト	FTP サーバ	制御コネクション確立後、FTP アダプタから FTP コマンドの送信がない場合に、FTP サーバ側で FTP アダプタの制御コネクションを解放する。	FTP アダプタに通信エラーが返り、異常終了する。
2	(④-1)FTP コマンドリプ ライタイムアウト	FTP アダプタ	FTP コマンド送信後、FTP サーバからの応答メッセージを受信するまでのタイムアウト時間。	FTP アダプタに通信エラーが返り、異常終了する。
3	(④-2)Listen タイムアウト		PORT コマンド発行時に、PORT コマンドの応答受信後から、データコネクションに接続するまでのタイムアウト時間。 Active モードの場合、ファイル受信時のタイムアウト時間。	FTP アダプタに通信エラーが返り、異常終了する。
4	(④-3)データ送受信 タイムアウト		ファイル転送が STOR/APPE の場合、データ送信タイムアウトが使用される。 ファイル転送が RETR の場合、データ受信タイムアウトが使用される。 データ送受信時に、パケットを送受信するまでのタイムアウト時間。	FTP アダプタが異常終了する。
5	(⑤-2)データ送受信 タイムアウト	FTP サーバ	ファイル転送が STOR/APPE の場合、データ受信タイムアウトが使用される。 ファイル転送が RETR の場合、データ送信タイムアウトが使用される。 データ送受信時に、パケットを送受信するまでのタイムアウト時間。	FTP アダプタに通信エラーが返り、異常終了する。

## (2) 設計指針

### FTP クライアントへのエラー応答を目的としたパラメタ

PUSH 型、PULL 型の場合で、以下の点に留意して、設計を行ってください。

PUSH 型の場合、ビジネスプロセス実行前に FTP クライアントから HCSC サーバへのデータ送信が完了します。

PULL 型の場合、ビジネスプロセス実行後に HCSC サーバから FTP クライアントへのデータ送信が開始します。

#### ●データ送受信後タイムアウト (FTP クライアント(①-2))

利用者をこれ以上待たせることができないため、FTP インバウンドアダプタへのデータ送受信後処理において、遅延障害が発生していると判断する時間を設定します。

##### <PUSH 型の場合>

データ送受信後タイムアウト=FTP クライアントから HCSC サーバへのデータ送信処理後のビジネスプロセス処理において遅延障害が発生していると判断する時間

タイムアウトパラメタ間の関係

データ送受信後タイムアウト > ビジネスプロセスの処理時間

##### <PULL 型の場合>

データ送受信後タイムアウト=ビジネスプロセス処理後、HCSC サーバから FTP クライアントへのデータ送信処理後において遅延障害が発生していると判断する時間

タイムアウトパラメタ間の関係

ビジネスプロセスの処理時間 > データ送受信後タイムアウト

#### ●データ送受信タイムアウト (FTP クライアント(①-1)、FTP インバウンドアダプタ(②-2))

利用者をこれ以上待たせることができないため、FTP クライアントと FTP インバウンドアダプタ間での入出力処理において、遅延障害が発生していると判断する時間を設定します。

データ送受信タイムアウト= FTP クライアントと FTP インバウンドアダプタ間での入出力処理で遅延障害が発生していると判断する時間

#### ●FTP 受付タイムアウト (FTP 受付⑥)

利用者をこれ以上待たせることができないため、FTP 受付がビジネスプロセスの処理において、遅延障害が発生していると判断する時間を設定します。

タイムアウト検知時、FTP 受付は FTP インバウンドアダプタにエラーを通知しますが、ビジネスプロセスの処理自体を中止するわけではないため、ビジネスプロセスがどのように決着したかは HCSC サーバ上のログで確認する必要があります。

PUSH 型、PULL 型いずれの場合でも、ビジネスプロセスの処理時間より長い時間を設定するようにします。

FTP 受付タイムアウト = ビジネスプロセス処理において遅延障害が発生していると判断する時間

タイムアウトパラメタ間の関係

FTP 受付タイムアウト > ビジネスプロセスの処理時間

リソースの開放を目的としたパラメタ

●データ送受信タイムアウト (FTP アダプタ(④-3)、FTP サーバ(⑤-2))

データ通信処理時間を踏まえ、明らかにファイル送受信中に異常な遅延が発生していると判断する時間を設定します。これによって、HCSC サーバのスレッド、コネクションを開放することが出来ます。

データ送受信タイムアウト = データ通信処理時間中に異常な遅延が発生していると判断する時間

●ファイル操作アダプタタイムアウト (ファイル操作アダプタ③)

ファイル操作処理時間を踏まえ、明らかにファイル操作中に異常な遅延が発生していると判断する時間を設定します。これによって、HCSC サーバのスレッド、コネクションを開放することが出来ます。

ファイル操作アダプタタイムアウト=ファイル操作処理時間中に異常な遅延が発生していると判断する時間

**(3) パラメタ一覧**

各定義ファイルの詳細は、表 4.2-1 定義ファイル一覧を参照してください。

**表 6.3-8 FTP クライアントのタイムアウトパラメタ**

#	パラメタ	設定方法	定義項目(初期値)	定義範囲
1	(①-1)データ送受信 タイムアウト	・ Linux の場合 ftsutil コマンドで、定義する。	・ Linux の場合 ftsutil -EO <データ送信待ちタイムアウト(60 秒)> ftsutil -EU <データ受信待ちタイムアウト(60 秒)>	1-3600(秒)

		<ul style="list-style-type: none"> <li>Windows の場合</li> </ul> JP1/FTS/FTP の[環境定義]ダイアログボックスで、定義する。	<ul style="list-style-type: none"> <li>Windows の場合</li> </ul> データ送信待ちタイムアウト(60 秒) データ受信待ちタイムアウト(60 秒)	
2	(①-2)データ送受信後 タイムアウト		データ送受信後タイムアウト値は、データ送信待ちタイムアウト、データ字 受信待ちタイムアウトの値が使用される。	-

・表 6.3-9 FTP インバウンドアダプタのタイムアウトパラメタ

#	パラメタ	定義ファイル	定義項目(初期値)	定義範囲
1	(②-1)アイドル タイムアウト	FTP インバウンドアダプタ 属性定義ファイル	<code>&lt;config-property-name&gt;nioListener_idleTimeout&lt;/config-property-name &gt;</code> <code>&lt;config-property-type&gt;java.lang.String&lt;/config-property-type&gt;</code> <code>&lt;config-property-value&gt;300&lt;/config-property-value&gt;</code>	0-7200(秒)
2	(②-2)データ送受信 タイムアウト		<code>&lt;config-property-name&gt;nioListener_dataConnection_idleTimeout&lt;/confi g-property-name&gt;</code> <code>&lt;config-property-type&gt;java.lang.String&lt;/config-property-type&gt;</code> <code>&lt;config-property-value&gt;10&lt;/config-property-value&gt;</code>	0-1800(秒)

・表 6.3-10 FTP アダプタのタイムアウトパラメタ

#	パラメタ	定義ファイル	定義項目(初期値)	定義範囲
1	(④-1)FTP コマンドリプライ タイムアウト	FTP アダプタ実行環境プロ パティファイル	<code>ftpadp.ftp-command-reply.timeout=3</code>	1-3600(秒)
2	(④-2)Listen タイムアウト		<code>ftpadp.listen.timeout=3</code>	1-3600(秒)
3	(④-3)データ送受信タイムア ウト		<ul style="list-style-type: none"> <li>データ送信タイムアウト</li> </ul> <code>ftpadp.data-con-put.timeout=60</code> <ul style="list-style-type: none"> <li>データ受信タイムアウト</li> </ul> <code>ftpadp.data-con-get.timeout=60</code>	1-3600(秒)

・表 6.3-11 FTP サーバのタイムアウトパラメタ

#	パラメタ	設定方法	定義項目(初期値)	定義範囲
1	(⑤-1)アイドル タイムアウト	<ul style="list-style-type: none"> <li>Linux の場合 ftputil コマンドで、定義する。</li> <li>Windows の場合</li> </ul>	<ul style="list-style-type: none"> <li>Linux の場合 ftputil -ET &lt;アイドルタイムアウト(900 秒)&gt;</li> <li>Windows の場合 アイドルタイムアウト(900 秒)</li> </ul>	30- 7200(秒)
2	(⑤-2)データ送受信 タイムアウト	JP1/FTS/FTP の[環境定義] ダイアログボックスで、定義する。	<ul style="list-style-type: none"> <li>Linux の場合 ftputil -EO &lt;データ送信待ちタイムアウト(60 秒)&gt; ftputil -EU &lt;データ受信待ちタイムアウト(60 秒)&gt;</li> <li>Windows の場合 データ送信待ちタイムアウト(60 秒) データ受信待ちタイムアウト(60 秒)</li> </ul>	1-3600(秒)

・表 6.3-12 ファイル操作アダプタのタイムアウトパラメタ

#	パラメタ	定義ファイル	定義項目(初期値)	定義範囲
1	③ファイル操作アダプタ タイムアウト	ファイル操作アダプタ実行 環境プロパティファイル	transform.timeout=60 (注: 分割処理方式の場合のみ有効となる。一括処理方式ではタイムアウト による打ち切りは行わない。)	0- 86400(秒)

表 6.3-13 FTP 受付のタイムアウトパラメタ

#	パラメタ	定義ファイル	定義項目(初期値)	定義範囲
1	⑥FTP 受付タイムアウト	FTP 受付コンフィグファイル	urecp-ftp.request-timeout=0	0- 2147483647 (秒)

## 6.4 リトライ

---

### 6.4.1 FTP 連携におけるリトライ

FTP アダプタが FTP サーバに接続する際に、FTP コネクションの最大同時接続数や Listen キューの上限に達して、接続エラーとなる場合があります。

リトライを行わない場合は、1 回の接続で、FTP アダプタで接続エラーが発生し、ビジネスプロセスの処理が中断してしまいます。リトライを行うことで、FTP アダプタが何度か FTP サーバに接続を実施し、FTP アダプタですぐに接続エラーとなり、ビジネスプロセスの処理が中断することを防ぎます。

また、FTP クライアントが FTP インバウンドアダプタに接続する際も上記と同様の問題が発生するため、リトライを設計します。

以下の PP を前提として記載します。

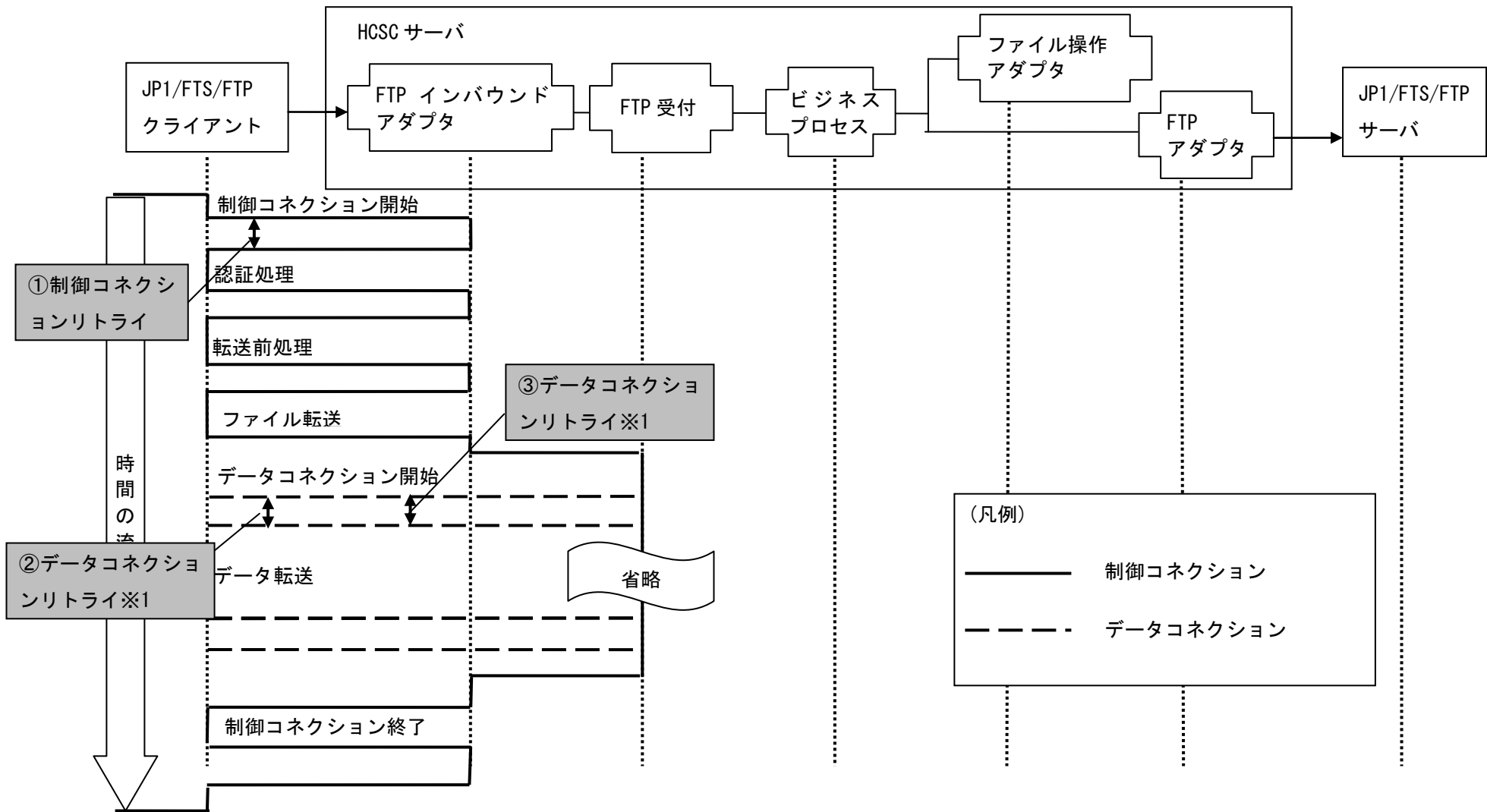
- FTP クライアントは JP1/FTS/FTP クライアント
- FTP サーバは JP1/FTS/FTP サーバ

#### (1) 設計パラメタ

リトライの設計箇所を以下のパターンについて示します。

- (a) FTP クライアントと FTP インバウンドアダプタのリトライ
- (b) FTP アダプタと FTP サーバのリトライ

(a) FTP クライアントと FTP インバウンドアダプタのリトライ



※1 : FTP のデータ転送モードが Passive モードの場合は、「②データコネクションリトライ」が使用され、Active モードの場合は、「③データコネクションリトライ」が使用されます。

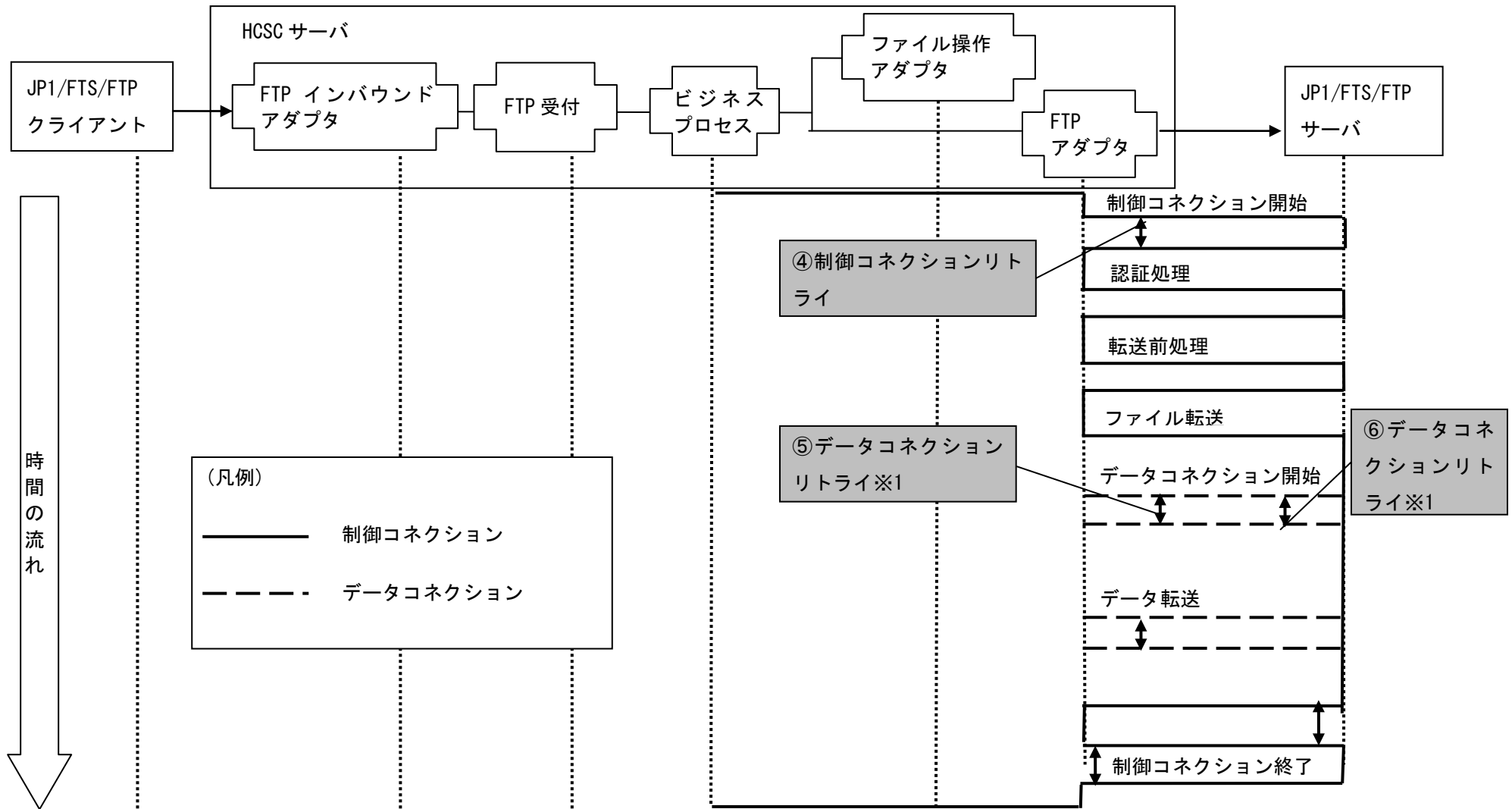
図 6.4-1 FTP クライアントと FTP インバウンドアダプタのリトライ

表 6.4-1 FTP クライアントと FTP インバウンドアダプタのリトライパラメータ一覧

#	パラメタ	コンポーネント	説明	タイムアウト発生時の動作
1	①制御コネクションリトライ	FTP クライアント	制御コネクションを確立できなかった場合、リトライを行います。リトライ回数とリトライ間隔を指定します。	FTP クライアントが異常終了します。ビジネスプロセスは実行していません。
2	②データコネクションリトライ	FTP クライアント	データ転送モードが <b>Passive</b> モードの場合のみ使用します。 データコネクションを確立できなかった場合、リトライを行います。リトライ回数とリトライ間隔を指定します。	FTP クライアントが異常終了します。ビジネスプロセスは実行していません。
3	③データコネクションリトライ	FTP インバウンドアダプタ	データ転送モードが <b>Active</b> モードの場合のみ使用します。 データコネクションを確立できなかった場合、リトライを行います。リトライ回数とリトライ間隔を指定します。	FTP クライアントに通信エラーが返ります。 ビジネスプロセスは実行していません。



(b) FTP アダプタと FTP サーバのリトライ



※1 : FTP のデータ転送モードが Passive モードの場合は、「⑤データコネクションリトライ」が使用され、Active モードの場合は、「⑥データコネクションリトライ」が使用される。

図 6.4-2 FTP アダプタと FTP サーバのリトライ

表 6.4-2 FTP アダプタと FTP サーバのリトライパラメータ一覧

#	パラメタ	コンポーネント	説明	タイムアウト発生時の動作
1	④制御コネクション リトライ	FTP アダプタ	制御コネクションを確立できなかった場合、リトライを行う。リトライ回数とリトライ間隔を指定します。	FTP アダプタが異常終了します。
2	⑤データコネクション リトライ	FTP アダプタ	データ転送モードが <b>Passive</b> モードの場合のみ使用します。 データコネクションを確立できなかった場合、リトライを行います。リトライ回数とリトライ間隔を指定します。	FTP アダプタが異常終了します。
3	⑥データコネクション リトライ	FTP サーバ	データ転送モードが <b>Active</b> モードの場合のみ使用します。 データコネクションを確立できなかった場合、リトライを行います。リトライ回数とリトライ間隔を指定します。	FTP アダプタに通信エラーが返り、異常終了します。

(2) 設計指針

●制御コネクションリトライ (FTP クライアント、FTP アダプタ)

制御コネクションのリトライ回数とリトライ間隔を長くすると、利用者を長時間待たせるため、適切な範囲でリトライ時間を設計します。

$$\text{リトライ時間} = \text{リトライ回数} \times \text{リトライ間隔}$$

●データコネクションリトライ (FTP クライアント、FTP インバウンドアダプタ、FTP アダプタ、FTP サーバ)

データコネクションのリトライ回数とリトライ間隔を長くすると、利用者を長時間待たせるため、適切な範囲でリトライ時間を設計します。

$$\text{リトライ時間} = \text{リトライ回数} \times \text{リトライ間隔}$$

(3) パラメータ一覧

表 6.4-3 FTP クライアントのリトライパラメタ

#	パラメタ	設定方法	定義項目(初期値)	定義範囲
1	①②リトライ回数※1	<ul style="list-style-type: none"> <li>Linux の場合 ftputil コマンドで、定義する。</li> <li>Windows の場合 JP1/FTS/FTP の[環境定義]</li> </ul>	<ul style="list-style-type: none"> <li>Linux の場合 ftputil -ER &lt;ホスト接続時のリトライ回数(5 回)&gt;</li> <li>Windows の場合 ホスト接続時のリトライ回数(5 回)</li> </ul>	0- 100000(回)
2	①②リトライ間隔※1	ダイアログボックスで、定義する。	<ul style="list-style-type: none"> <li>Linux の場合 ftputil -EY &lt;ホスト接続時のリトライ間隔(2 秒)&gt;</li> <li>Windows の場合 ホスト接続時のリトライ間隔(2 秒)</li> </ul>	0- 86400(秒)

※1：制御コネクション、データコネクションのリトライパラメタは、同一定義です。

・表 6.4-4 FTP インバウンドアダプタのリトライパラメタ

#	パラメタ	設定方法	定義項目(初期値)	定義範囲
1	③データコネクションのリトライ回数	FTP インバウンドアダプタ 属性定義ファイル	<config-property-name>nioListener_dataConnection_active_connectRetr yCount</config-property-name> <config-property-type>java.lang.String</config-property-type> <config-property-value>5</config-property-value>	0-50(回)
2	③データコネクションのリトライ間隔		<config-property-name>nioListener_dataConnection_active_connectRetr yInterval</config-property-name> <config-property-type>java.lang.String</config-property-type> <config-property-value>10</config-property-value>	0-1800(秒)

・表 6.4-5 FTP アダプタのリトライパラメタ

#	パラメタ	設定方法	定義項目(初期値)	定義範囲
1	④制御コネクションのリトライ回数	FTP アダプタ実行 環境プロパティフ ファイル	ftpadp.control-con.retry.count=5	0-10000(回)
2	④制御コネクションのリトライ間隔		ftpadp.control-con.retry.interval=2	0-86400(秒)
3	⑤データコネクションのリトライ回数		ftpadp.data-con.retry.count=5	0-10000(回)
4	⑤データコネクションのリトライ間隔		ftpadp.data-con.retry.interval=2	0-86400(秒)

・表 6.4-6 FTP サーバのリトライパラメタ

#	パラメタ	設定方法	定義項目(初期値)	定義範囲
1	⑥データコネクションのリトライ回数	<ul style="list-style-type: none"> <li>・ Linux の場合 ftputil コマンドで、定義する。</li> <li>・ Windows の場合 JP1/FPS/FTP の[環境定義]</li> </ul>	<ul style="list-style-type: none"> <li>・ Linux の場合 ftputil -ER &lt;ホスト接続時のリトライ回数(5 回)&gt;</li> <li>・ Windows の場合 ホスト接続時のリトライ回数(5 回)</li> </ul>	0-100000(回)
2	⑥データコネクションのリトライ間隔	ダイアログボックスで、定義する。	<ul style="list-style-type: none"> <li>・ Linux の場合 ftputil -EY &lt;ホスト接続時のリトライ間隔(2 秒)&gt;</li> <li>・ Windows の場合 ホスト接続時のリトライ間隔(2 秒)</li> </ul>	0-86400(秒)