WebSphere® Application Server V4.0 for z/OS and OS/390

操作および管理

SA88-8653-00
(英文原典：SA22-7835-00)

WebSphere® Application Server V4.0 for z/OS and OS/390

操作および管理

SA88-8653-00
(英文原典：SA22-7835-00)

お願い

本書および本書で紹介する製品をご使用になる前に、143ページの『付録C. 特記事項』に記載する一般情報をお読みください。

本書は、WebSphere Application Server V4.0 for z/OS and OS/390 (5655-F31) に適用されます。また本書の改訂版などで特に断りのない限り、以降のすべてのリリース、修正レベルにも適用されます。

WebSphere Application Server V4.0 for z/OS and OS/390 に関連する資料の最新版は次の Web サイトにあります。
http://www.ibm.com/jp/software/websphere/appserv/

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

http://www.ibm.com/jp/manuals/main/mail.html

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

http://www.ibm.com/jp/manuals/ の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

原　典： SA22–7835–00
WebSphere® Application Server V4.0 for z/OS and OS/390
Operations and Administration

発　行： 日本アイ・ビー・エム株式会社

担　当： ナショナル・ランゲージ・サポート

第1刷 2001.6

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財) 日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

目次

図

表

本書について

本書は、WebSphere for z/OS の操作および管理の手順について述べたものです。

注: 完全な製品名は「WebSphere Application Server V4.0 for z/OS and OS/390」ですが、以後、本書では「WebSphere for z/OS」または「Application Server」と呼びます。

対象読者

本書は WebSphere for z/OS のシステム・オペレーターおよび管理者向けのマニュアルです。Application Server、OE、RRS、および WLM の実際の使用経験があることが望まれますが、必須ではありません。Application Server について知りたいオペレーターや管理者の方は、まず *WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を読んでください。このマニュアルでは、WebSphere for z/OS の管理アプリケーションと操作アプリケーションについて説明しています。また、WebSphere Application Server の Web サイト (`http://www.ibm.com/jp/software/websphere/appserv/`) では、関連情報や出版物にアクセスすることができます。

本書の構成

- 1ページの『第1章 はじめに』では、WebSphere for z/OS の操作および管理を概説します。
- 5ページの『第2章 WebSphere for z/OS 操作の実行場所の識別』では、一般的な操作タスクをリスト表示して、システム管理 GUI または MVS コンソールを使用してこれらのタスクを実行する時期について説明します。
- 11ページの『第3章 WebSphere for z/OS の操作』では、Application Server の基本的な操作タスクについて説明します。
- 33ページの『第4章 z/OS または OS/390 サブシステムの操作上の考慮事項』では、Application Server 環境で z/OS または OS/390 サブシステムを使用する際の操作上の考慮事項について説明します。
- 37ページの『第5章 WebSphere for z/OS のバックアップのガイドラインと手順』では、Application Server のバックアップのガイドラインと手順について説明します。

- 43ページの『第6章 WebSphere for z/OS および従属するサブシステムのモニタリングおよびリカバリー』では、Application Server とそれに従属するサブシステムのモニターおよびリカバリーのガイドラインについて説明します。
- 61ページの『第7章 WebSphere for z/OS の管理手順』では、Application Server の管理タスクについて説明します。
- 93ページの『第8章 WebSphere for z/OS のチューニングおよびパフォーマンス・モニター』では、Application Server のパフォーマンスのモニターに関するガイドラインについて説明します。
- 109ページの『第9章 システム管理機能 (SMF) による記録とモニター』では、システム管理機能 (SMF) による Application Server の記録とモニターについて説明します。
- 119ページの『付録A. SMF レコード・タイプ 120 (WebSphere for z/OS)』では、Application Server に対して使用されるシステム管理機能 (SMF) レコード・タイプ 120 について説明します。
- 139ページの『付録B. アプリケーション・サーバーのネーミング規則』では、アプリケーション・サーバーで安定したネーミング規則を確立する方法について説明します。
- 143ページの『付録C. 特記事項』には、プログラミング・インターフェース、本書で使用されている例、および商標に関する特記事項が記載されています。

関連情報の入手先

以下に、WebSphere for z/OS ライブラリーに含まれるマニュアルのリストを示します。これらのマニュアルは次の Web サイトで入手することができます。

`http://www.ibm.com/jp/software/websphere/appserv/`

- *WebSphere Application Server V4.0 for z/OS and OS/390: プログラム・ディレクトリー*, GI88-8549 - WebSphere for z/OS のエレメントを示し、WebSphere for z/OS のインストール方法について説明しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390: License Information*, LA22-7855 - WebSphere for z/OS のライセンス情報を記載しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 - WebSphere for z/OS の計画、インストール、およびカスタマイズのための各タスクとガイドラインについて説明しています。

- *WebSphere Application Server V4.0 for z/OS and OS/390: メッセージおよび診断*, GA88-8655 - 診断情報を示し、WebSphere for z/OS に関連するメッセージとコードについて説明しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390: 操作および管理*, SA88-8653 - システムの操作タスクと管理タスクについて説明しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390: J2EE アプリケーションのアセンブル*, SA88-8654 - J2EE アプリケーションの開発およびアセンブル方法と、WebSphere for z/OS J2EE サーバーへのインストール方法について説明しています。また、WebSphere Application Server for OS/390 の旧リリース、または他の WebSphere ファミリー・プラットフォームからのアプリケーションの移送に関する情報も収録しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390: CORBA アプリケーションのアセンブル*, SA88-8658 - CORBA アプリケーションの開発およびアセンブル方法と、WebSphere for z/OS (MOFW) サーバーへの展開方法について説明しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 - システム管理ユーザー・インターフェースで提供される、システム管理タスクと操作タスクについて説明しています。
- *WebSphere Application Server V4.0 for z/OS and OS/390: システム管理スクリプト API*, SA22-7839 - WebSphere for z/OS システム管理スクリプト API 製品の機能について説明しています。

z/OS または OS/390 の他のエレメントや製品に関する情報を参照したい場合は、次のインターネット・サイトにあるリンクからそれらのすべての情報にアクセスすることができます。

```
http://www.ibm.com/servers/eserver/zseries/zos/
http://www.ibm.com/servers/s390/os390/
```

以下の資料には、特に有用な情報を収録しています。

- *Getting Started with WebSphere Application Server,* SC09-4581 - WebSphere for z/OS の概要、および環境のセットアップ要件について説明しています。
- *WebSphere ビジネス構築のソリューション,* SD88-7362

第1章 はじめに

情報技術 (IT) を、IT サービスを効果的に提供するために管理することは、ビジネスにとってはかなりの難問です。ビジネス・リーダーを目指す企業にとって、システムの使用可能性を高めることがきわめて重要になります。

WebSphere for z/OS のような複合環境をうまく制御するための主なポイントの 1 つは、安定したモニタリングおよび操作の方式を確実に実装して、システムの使用可能性とパフォーマンスを最大限にすることです。各企業はそれぞれのビジネス・ニーズを慎重に検討したうえで、現在の技術で何ができるか、また自社のリソースが現在の技術の実装にどこまで役立つかを判別します。予定されたものであれ予定外のものであれ、長期間の障害に耐えられる企業はほとんどありません。企業には、高度な可用性が常に要求され、またその可用性を維持することにより企業は、ますます競争力を高めることができるのです。

組織が異なれば、可用性の意味も異なる場合があります。

高可用性

予定外の障害の影響を最小化または遮断するシステム特性。高可用性では、予定されたサービス時間中、アプリケーションの実行を継続するよう努力されます。高可用性には、コンポーネントに障害が発生した場合でも常にサービスが確保されるようにする、コンポーネント冗長度が含まれます。また、潜在的な問題が実稼働環境に影響を与える前に、確実にそれらを検出するための綿密なテストもここに含まれます。

連続稼働

予定された障害の影響を最小化または遮断するシステム特性。この特性は、障害 (予定されたものであってもそうでなくても) のない IT サービスを、カスタマーに提供しようとするものです。これを達成するのは難しくはありません。通信管理構成などの専用システムには、どんなタイプの障害も起こさずに何か月も稼働できる例がたくさんあります。ただし、これはシステムにほとんど、あるいはまったく変更を加えない場合の話で、実際のシステムでは現実的なシナリオとは言えません。

連続可用性

あらゆる障害の影響を最小化または遮断するシステム特性。高可用性と連続稼働を結合した結果です。つまり、アプリケーションが提供する IT サービスは、予定されたシステム障害でも予定外のシステム障害でも使用可能のままです。

WebSphere for z/OS の操作の概要

WebSphere for z/OS の操作アプリケーションでは、NT 上で稼働するシステム管理インターフェースを使用して WebSphere for z/OS サーバーおよびサーバー・インスタンスを管理することができます。すべてのサーバー・インスタンスの状況表示、アプリケーション・サーバーおよびサーバー・インスタンスの停止、アプリケーション・サーバーおよびサーバー・インスタンスの取り消し、サーバーおよびサーバー・インスタンスの取り消しと再始動、操作ウィンドウのフィルタリングを実行できます。WebSphere for z/OS 操作アプリケーションの使用方法については、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照してください。

本書「*WebSphere Application Server V4.0 for z/OS and OS/390: 操作および管理*, SA88-8653」では、WebSphere for z/OS の操作および管理のガイドラインと手順を示します。本書では、次の項目について説明します。

- z/OS または OS/390 コンソールからの操作タスクの実行
- サーバー管理のヒント
- 操作のガイドライン
- システム・パフォーマンスを向上させるための従属サブシステムの調整
- リカバリーのシナリオとガイドライン
- モニターとバックアップのガイドライン

WebSphere for z/OS の管理の概要

WebSphere for z/OS の管理アプリケーションでは、NT 上のシステム管理ユーザー・インターフェースを使用して、WebSphere for z/OS アプリケーションとそれらのアプリケーションの稼働環境を表示したり変更することができます。WebSphere for z/OS 管理アプリケーションの使用方法については、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照してください。付随する管理タスクおよびガイドラインは、61ページの『第7章 WebSphere for z/OS の管理手順』および 93ページの『第8章 WebSphere for z/OS のチューニングおよびパフォーマンス・モニター』にあります。

必須の WebSphere for z/OS エレメントとサブシステムの概要

WebSphere for z/OS エレメント

WebSphere for z/OS ホスト・システムの必須エレメントとして、次のものがあります。

- WebSphere for z/OS システム・サーバー・インスタンス
  - デーモン
  - システム管理サーバー (SMS)
  - ネーミング
  - インターフェース・リポジトリー (IR)
- WebSphere for z/OS アプリケーション・サーバー・インスタンス
  - 制御領域 (CR)
  - サーバー領域 (SR)

他の WebSphere Application Server ホスト (S/390 または分散ホスト) 上で稼働可能なオプション・エレメントとして、次のものがあります。

- WebSphere Application Server クライアント

詳しくは、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

z/OS または OS/390 必須サブシステム

サブシステムを始動および停止する順序と、サブシステムに障害が発生したときにシステムを回復する方法については、43ページの『第6章 WebSphere for z/OS および従属するサブシステムのモニタリングおよびリカバリー』を参照してください。

z/OS または OS/390 に必ず必要なサブシステムについては、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

第2章 WebSphere for z/OS 操作の実行場所の識別

この章では、WebSphere for z/OS の主な操作タスク、およびこれらのタスクを実行するために必要な情報を示します。Application Server の操作はすべて、z/OS または OS/390 MVS コンソールから実行することができます。NT では、一部のアクティビティーは、システム管理ユーザー・インターフェース (SM/EUI) からも実行できます。システム管理ユーザー・インターフェースに関しては、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照してください。

注: 下記で使用されている用語「アプリケーション・サーバー」は、デーモン、SM (システム管理)、ネーミング、または IR (インターフェース・リポジトリー) を指しています。

表 1. WebSphere for z/OS 操作タスク

タスク	MVS コンソール (z/OS または OS/390)	SM/EUI (NT)	TSO パネル	関連手順 (参照個所)
デーモンの取り消し	可	不可	不可	20ページの『デーモンの取り消し』を参照。
アプリケーション・サーバーまたはサーバー・インスタンスの取り消し	可	可	不可	19ページの『アプリケーション・サーバーおよびサーバー・インスタンスの取り消し』を参照。
ネーム・スペースの内容の検査	不可	不可	不可	22ページの『ネーム・スペースの内容の検査』を参照。
ARM 登録アドレス・スペース (WebSphere for z/OS サーバー) の状況の表示	可	不可	不可	22ページの『WebSphere for z/OS サーバーを含む、ARM 登録アドレス・スペースの状況の表示』を参照。

表 1. WebSphere for z/OS 操作タスク (続き)

タスク	MVS コンソール (z/OS または OS/390)	SM/EUI (NT)	TSO パネル	関連手順 (参照個所)
サーバーまたはサーバー・インスタンスの状況の表示	可	可	不可	11ページの『第3章 WebSphere for z/OS の操作』および *WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照。
DB2 の作業単位 (スレッド) の表示	可	不可	不可	24ページの『DB2 の作業単位 (スレッド) の表示』を参照。
DB2 の未確定作業単位 (スレッド) の表示	可	不可	不可	25ページの『DB2 の未確定の作業単位 (スレッド) の表示』を参照。
RRS の作業単位の表示	不可	不可	可	26ページの『RRS の作業単位の表示』を参照。RRS の作業単位を表示する方法については、*z/OS MVS プログラミング:リソース・リカバリー*, SA88-8582 を参照。
CICS の作業単位の表示	可	不可	可	25ページの『CICS の作業単位の表示』を参照。CICS の作業単位の表示に関する詳細については、*CICS Operations and Utilities Guide*, SC34-5717 を参照。
IMS の作業単位の表示	可	不可	不可	25ページの『IMS の作業単位 (トランザクション) の表示』を参照。*IMS/ESA Summary of Operator Commands*, SC26-8766 も参照。
Application Server のホット・スタート	可	不可	不可	20ページの『WebSphere for z/OS のホット・スタートとクイック・スタート』を参照。*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 も参照。

表 1. WebSphere for z/OS 操作タスク (続き)

タスク	MVS コンソール (z/OS または OS/390)	SM/EUI (NT)	TSO パネル	関連手順 (参照個所)
Application Server のクィック・スタート	可	不可	不可	20ページの『WebSphere for z/OS のホット・スタートとクイック・スタート』を参照。*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 も参照。
複数の異なるサーバーのエラー・ログ・ストリームのセットアップ	不可	SMUI から、ログ・ストリームをサーバーと関連付けることができます。	不可	22ページの『複数の異なるサーバーおよびサーバー・インスタンスのエラー・ログ・ストリームのセットアップ』を参照。 エラー・ログ・ストリームの設定に関する詳細については、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照。
SMF 記録のセットアップ	可	ここで使用可能にするが、コンソールから開始します。	不可	112ページの『SMF による記録のセットアップ』を参照。*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 も参照。
Application Server ホスト環境のシャットダウン	可	不可	不可	15ページの『WebSphere for z/OS ホスト環境のシャットダウン』を参照。
Application Server ホスト環境の始動	可	不可	不可	12ページの『WebSphere for z/OS ホスト環境の始動』を参照。
サーバーまたはサーバー・インスタンスの始動	可	アプリケーション・サーバーのみ	不可	16ページの『サーバーおよびサーバー・インスタンスの始動』および *WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照。

表 1. WebSphere for z/OS 操作タスク (続き)

タスク	MVS コンソール (z/OS または OS/390)	SM/EUI (NT)	TSO パネル	関連手順 (参照個所)
サーバーの停止	不可	アプリケーション・サーバーのみ	不可	*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照。
サーバー・インスタンスの停止	可	可	不可	19ページの『アプリケーション・サーバー・インスタンスの停止』および *WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照。
WebSphere for z/OS システム・サーバーのサービス停止	可	アプリケーション・サーバーのみ。WebSphere for z/OS システム・サーバーのサービスを SMUI (デーモン、IR、ネーミング、SM) から停止することはできません。	不可	20ページの『WebSphere for z/OS システム・サーバーのサービスの停止』を参照。

表 1. WebSphere for z/OS 操作タスク (続き)

タスク	MVS コンソール (z/OS または OS/390)	SM/EUI (NT)	TSO パネル	関連手順 (参照個所)
ワークロード管理				
ワークロード管理アプリケーション環境の検査および管理 (表示、停止 / 静止、再始動 / 再開)	可	不可	不可	26ページの『WLM アプリケーション環境の状況の表示』を参照。
停止状態から使用可能状態への復帰 (ワークロード管理)	可	不可	不可	30ページの『停止状態から使用可能状態への復帰』を参照。

システム管理ユーザー・インターフェースからの WebSphere for z/OS の操作

システム管理ユーザー・インターフェースの操作アプリケーションを使用すると、Application Server 環境を操作するための次のタスクを実行することができます。

- サーバーまたはサーバー・インスタンスの始動。
- サーバーまたはサーバー・インスタンスの停止。
- サーバーまたはサーバー・インスタンスの取り消し。
- サーバーまたはサーバー・インスタンスの取り消しおよび再始動。
- 「操作 (operations)」ウィンドウのフィルタリング。

詳しくは、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照してください。

MVS コンソールからの WebSphere for z/OS の操作

本書で説明しているように、WebSphere for z/OS は MVS コンソールから操作することができます。

Application Server 環境の自動化はすべて、MVS コンソールのインターフェースを介して実行されます。Netview などの製品で表示されるのは、MVS コンソ

ールに表示されるメッセージのコピーです。これらの自動化製品では、「仮想」MVS コンソールをソースに使用して、システムにコマンドを入力することもできます。

第3章 WebSphere for z/OS の操作

この章では、MVS コンソールから実行できる、WebSphere for z/OS の操作手順について説明します。

これらの操作タスクを実行する前に、以下の用語について理解しておく必要があります。

デーモン

Application Server ノード内の最初の接触点。他のサーバーやクライアントは、デーモンから発行されたネットワーク・アドレスを使用して、Application Server システムに要求を送ります。デーモンは要求を受け取ると、要求された機能がノード内のどのサーバーによって提供されるか判別し、要求を該当のサーバーに転送します。

z/OS または OS/390 システム

z/OS または OS/390 と Application Server が稼働しているコンピューターとその関連装置。

シスプレックス

特定のマルチシステム・ハードウェアとソフトウェア・サービスを使用して互いに通信し、協同で動作することによってカスタマーのワークロードを処理する z/OS または OS/390 システムのセット。シスプレックスは、単一イメージのシステム・コンプレックス (システム複合体) です。シスプレックスとは、複数の z/OS または OS/390 システムが統合され、単一のシステム複合体として機能しているものを指します (たとえば、1 つのハードウェアに 2 つの LPAR が存在する場合)。つまり、シスプレックスとは、複数のシステムから構成されていながら、単一のインスタンスのように動作し、反応するシステムのことです。

サーバー

サーバー・インスタンスの論理的なグループ。1 つのサーバー内のサーバー・インスタンスはすべて同じ構造を持ち、同じアプリケーションのセットを実行します。管理は通常、サーバー・レベルで行われます。また、管理の観点からは、サーバーはシスプレックス内の単一エンティティーです。サーバーは、ネットワークおよびオペレーターに対する、単一の制御インターフェースとなっています。

サーバー・インスタンス

Application Server アプリケーションが稼働する機能コンポーネント。サーバー・インスタンスは、サーバーのすべての機能を提供する複製サーバーのインスタンスです。サーバー内のすべてのサーバー・インスタンスは、構造は同じです。

個々のサーバー・インスタンスは、その固有の名前を通して、システム管理ユーザー・インターフェース操作アプリケーションまたは MVS コンソールから管理することができます。

サーバー・インスタンスには、制御領域 (1 つ) と サーバー領域 (1 つ以上) の 2 つの種類のアドレス・スペースがあります。アプリケーション・サーバー・コードはサーバー領域で実行されます。サーバー領域は、システムのワークロード要求に基づいて複製することができます。制御領域はサーバー領域へのメッセージをキューイングします。

WebSphere for z/OS ホスト環境の始動

ここでは、WebSphere for z/OS ホスト環境の始動方法について説明します。ホスト環境の始動前に準備しておく必要があるサブシステムについての詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

WebSphere for z/OS ホスト環境の始動ステップ

始める前に: デーモンを始動すると、デフォルトで SMS、ネーミング、および IR が使用可能になります。ただ、コンソールを使用するか、または自動化によって WebSphere for z/OS ホスト環境を始動する必要があります。

注: 以下の手順には、DB2 の起動ステップも含まれています。WebSphere for z/OS は DB2 を使用するため、シスプレックス構成において、Application Server を実行する各システムがデータ共用 DB2 インスタンスにアクセス可能であることが必要になります。

Application Server ホスト環境を始動するには、以下のステップを実行します。

1. 前提条件となっているすべてのサブシステムを始動します (*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照)。

---

2. 次の MVS start コマンドを使用して RRS を始動します。

```
start atrrrs,sub=master
```

注: RRS は、DB2 の始動前に始動する必要があります。

---

3. 各システムの DB2 を始動します。

   WebSphere for z/OS は、シスプレックス内の、Application Server が稼働するすべてのシステム上で共用 DB2 構成が稼働していることを必要とします。これは、Application Server の操作および管理データがこの共用 DB2 に保管されるためです。

   次の例は、シスプレックス内の各システム上で DB2 をどのように始動するかを示したものです。

   例: 1 つの DB2 がすべてのシステム間で共用されます。しかし、各サブシステムでは、固有の DB2 名を使用する必要があります。

   --DB1G start DB2
   この DB1G は、スリー・ウェイ・シスプレックス内のシステム 1 で稼働する DB2 サブシステムの名前です。

   --DB2G start DB2
   この DB2G は、スリー・ウェイ・シスプレックス内のシステム 2 で稼働する DB2 サブシステムの名前です。

   --DB3G start DB2
   この DB3G は、スリー・ウェイ・シスプレックス内のシステム 3 で稼働する DB2 サブシステムの名前です。

---

4. デーモンを始動します。

   デーモンは単一の障害地点となる可能性があるため、ツー・ウェイ・シスプレックスの場合は少なくとも 2 つのデーモンを使用するようお勧めします。シスプレックス内に複数のデーモンを配置すると、可用性が向上するという利点があります。デーモンが複数存在すれば、1 つのシステムがダウンした場合でも、作業を続行することができます。**Application Server ランタイム構成**全体は、次の要素で構成されます。

   - デーモン (DM)
   - システム管理 (SM) − (デーモンによって始動される)
   - ネーミング (NM) − (デーモンによって始動される)
   - インターフェース・リポジトリー (IR) − (デーモンによって始動される)

注:

a. WebSphere for z/OS サーバーは、デーモンが始動しているシステム上でのみ稼働可能です。

b. カスタマー・アプリケーション・サービスは、シスプレックス内の各システム上で始動するようお勧めします。

詳しくは、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

シスプレックス内の最初のシステム上のデーモンを始動するには、次のコマンドを入力します。

```
s bbodmn.daemon01,srvname='DAEMON01'
```

この daemon01 は、始動する制御領域のステップ名です (デーモンおよびサーバー名は各システムで異なります)。

注: 引用符で囲まれた制御領域の名前 (srvname='DAEMON01') では大文字と小文字が区別されるので、必ず**大文字**で入力する必要があります。

シスプレックス内の 2 番目のシステム上のデーモンを始動するには、次のコマンドを入力します。

```
s bbodmn.daemon02,srvname='DAEMON02'
```

この daemon02 は、始動する制御領域のステップ名です。

シスプレックス内のすべてのイメージまたはシステム上のデーモンを始動します。

例: 次の例は、コマンドのシステム・ログとデーモンの始動に対する応答を示したものです。

```
S BBODMN.DAEMON01
BBOU0007I CB SERIES DAEMON DAEMON01 IS STARTING.
START BBOSMS.SYSMGT01,SRVNAME='SYSMGT01',PARMS=''
BBOU0001I CB SERIES CONTROL REGION SYSMGT01 IS STARTING.
START BBONM.NAMING01,SRVNAME='NAMING01',PARMS=''
BBOU0001I CB SERIES CONTROL REGION NAMING01 IS STARTING.
START BBOIR.INTFRP01,SRVNAME='INTFRP01',PARMS=''
BBOU0001I CB SERIES CONTROL REGION INTFRP01 IS STARTING.
BBOU0016I INITIALIZATION COMPLETE FOR DAEMON DAEMON01.
BBOU0020I INITIALIZATION COMPLETE FOR CB SERIES CONTROL REGION SYSMGT01.
BBOU0020I INITIALIZATION COMPLETE FOR CB SERIES CONTROL REGION INTFRP01.
BBOU0020I INITIALIZATION COMPLETE FOR CB SERIES CONTROL REGION NAMING01.
```

注: 入力されたコマンドは、デーモンに対する start コマンドだけです。その他の start コマンドは、内部で生成されたコマンドです。

---

5. アプリケーション・サーバーを始動します (これらのサーバーが構成済みであると想定します)。

   開始するタスクに関連付けられている ID が start コマンドで指定されると、その ID は他の MVS コマンドによって使用されます。

   a. 最初のアプリケーション・サーバーを始動します。

      ```
      s bboasr1.bboasr1a,srvname='BBOASR1A'
      ```

      この BBOASR1A はアプリケーション・サーバー名です。

   b. 2 番目のアプリケーション・サーバーを始動します。

      ```
      s bboasr1.bboasr1b,srvname='BBOASR1B'
      ```

      この BBOASR1B はアプリケーション・サーバー名です。

   c. 3 番目のアプリケーション・サーバーを始動します。

      ```
      s bboasr1.bboasr1c,srvname='BBOASR1C'
      ```

      この BBOASR1C はアプリケーション・サーバー名です。

次のような初期化完了メッセージが表示されたら完了です。

```
BBOU0020I INITIALIZATION COMPLETE FOR CB SERIES CONTROL REGION BBOASR1C
```

WebSphere for z/OS ホスト環境のシャットダウン

シャットダウンは始動の逆の操作です。しかし、ここでは、Application Server ホスト環境をシャットダウンする前に実行する必要がある取り消し操作について説明します。

WebSphere for z/OS ランタイム環境のシャットダウン手順

始める前に: 1 つのシステム上のデーモンを停止しても、そのシステム以外のシステム上のサーバーは稼働を続けます。また、システム管理、ネーミング、および IR の各サーバーは、個別に停止する必要はありません。デーモンによってシステム管理が停止され、次にシステム管理によってネーミング、IR、およびその他すべての制御領域が停止されます。必要に応じて、これらのサーバーを直接停止することもできます。しかし、システム管理を停止すると、それ

によってデーモン以外のすべてのものが停止されます。制御領域はシステム管理が停止していると稼働できないためです。

WebSphere for z/OS ランタイム環境をシャットダウンするには、以下のステップを実行します。

1. すべてのアプリケーション制御領域を停止します。

   停止操作は、サーバーに対して SMUI から初期化することができます。シスプレックス内の、そのサーバーに関連するすべてのサーバー・インスタンスが停止します。また、**停止**操作は、MVS コンソールから個々のサーバー・インスタンスに対して発行することもできます。**停止**では、現在実行されているすべてのトランザクションが完了してからアプリケーションの制御領域が停止するのに対し、**取り消し**では、アプリケーションの制御領域は活動中のトランザクションの完了を待たずにただちに停止します。

   ---

2. シスプレックス内の各システム上のデーモンを停止します (または、時間があまりに長くかかる場合は、停止ではなく取り消し操作を実行します)。

   これは MVS コンソールから実行する必要があります。デーモンによってシステム管理が停止され、次にシステム管理によってネーミング、IR、およびその他のすべての制御領域が停止されます。

注: デーモンを停止すると、A03 異常終了コードとアドレス・スペースのダンプが生成されることがありますが、それによって停止操作が阻止されることはありません。

サーバーおよびサーバー・インスタンスの始動

ここでは、サーバーおよびサーバー・インスタンスの始動方法について説明します。

注: MVS コンソールからは、始動したい各サーバー・インスタンスを個別に始動する必要があります。しかし、SMEUI を使用する場合は、サーバー・インスタンスを個別に始動する方法のほかに、サーバー自体を始動することもできます。サーバーを始動すると、定義されているすべてのサーバー・インスタンスが自動的に始動されます。

サーバーおよびサーバー・インスタンスを始動するときには、システム・アドレス・スペースが稼働しているかどうかがわかると便利です。これを判別するには、次の 4 つの方法のいずれかを実行します。

- すべての アドレス・スペースのリストを表示する。

  ```
  d a,l
  ```

- *活動中*のすべての アドレス・スペースのリストを表示する。

  ```
  d a,a
  ```

- 特定のアドレス・スペースのみを表示する。

  ```
  d a,address-space-name
  ```

  (例: `D A,BBOASR1`)

  注: 最初の 2 つのコマンドでは実動システム上で非常に長いリストが生成されることになるため、最初の 2 つのコマンドよりも 3 番目のコマンドを使用するようお勧めします。ただし、表示したいアドレス・スペースの名前がわかっていなければなりません。

- 先頭に BBO が付く活動中のすべてのアドレス・スペースのリストを表示する。

  ```
  D A,BBO*
  ```

対象のアドレス・スペースが表示されたら、システムは稼働していることになります。

サーバーの始動ステップ

サーバーを始動するには、以下のステップを実行します。

1. 上記のコマンドのいずれかを発行して、該当のシステム・アドレス・スペースが稼働しているかどうかを判別します。稼働していない場合は、次のステップに進んでサーバーを始動します。

   ---

2. MVS コンソールからサーバーを始動するには、始動したい各サーバー・インスタンスを個別に始動する必要があります。『アプリケーション・サーバー・インスタンスの始動ステップ』を参照してください。

アプリケーション・サーバー・インスタンスの始動ステップ

アプリケーション・サーバー・インスタンスを始動するには、以下のステップを実行します。

1. アプリケーション・サーバー・インスタンスを始動するときには、事前に、アプリケーションで必要とするリソース・マネージャー (DB2、CICS など) が使用可能であることを確認する必要があります。詳しくは、関連資料を参照してください。

2. 上記のコマンドのいずれかを発行して、該当のシステム・アドレス・スペースが稼働しているかどうかを判別します。稼働していない場合は、次のステップに進んでサーバー・インスタンスを始動します。

3. アプリケーション・サーバー・インスタンスを始動するときには、事前に、アプリケーションで必要とするリソース・マネージャーが使用可能であることを確認する必要があります。

   サーバー・インスタンスを始動するには、次のコマンドを入力します。

   ```
   start controlregionprocname.serverinstance,srvname='serverinstance name',parms=''
   ```

   ここで、

   controlregionprocname
   サーバーの始動に使用する、proclib 内の JCL プロシージャー名。

   .serverinstance
   サーバー・インスタンスの名前 (proc の開始に使用されたステップ名)。この名前によって、SDSF パネルで実行中のアドレス・スペースを表示したときにそのアドレス・スペースを識別することができます。

   srvname
   特定のサーバー・インスタンスを指定したいときに使用するパラメーター。**このパラメーターでは大文字と小文字が区別されます。**

   注: このパラメーターは、始動対象が JCL proc のデフォルトのサーバー・インスタンス名である場合のみ省略可能です。それ以外の場合、このパラメーターの指定は必須です。

   'serverinstance name'
   始動する特定のサーバー・インスタンスの名前を指定します。

   parms JCL プロシージャーに渡すパラメーター情報を指定します。たとえば、コールド・スタートを指定する場合は、「'-ORBCBI COLD'」と入力します。

次のメッセージが表示された場合、サーバー・インスタンスは稼働しています。

```
BBOU0020I INITIALIZATION COMPLETE FOR CBSERIES CONTROL REGION server-instance...
```

アプリケーション・サーバー・インスタンスの停止

ここでは、アプリケーション・サーバー・インスタンスの停止方法について説明します。サーバー・インスタンスを停止すると、現在の処理が完了してからサーバーが停止します。サーバーを取り消すと、サーバーは現在の処理の完了を待たずに停止します。

アプリケーション・サーバー・インスタンスの停止ステップ

始める前に: この手順は、サーバー・インスタンスが次のいずれかの方法によって始動されていることを前提としています。

- ステップ名を指定した始動
- SMUI からの始動 - ステップ名が始動対象のサーバー・インスタンスを限定します。

サーバー・インスタンスを停止するには、以下のステップを実行します。

1. 次のコマンドを入力します。

   ```
   stop server-instance
   ```

   `server-instance` には、停止するサーバー・インスタンスの名前を指定します。

   注: 1 つのサーバー・インスタンスを停止しても、同じサーバーの他のインスタンスは影響を受けません。ただし、ワークロードは残りのサーバー・インスタンス間で平均化されます。

アプリケーション・サーバーおよびサーバー・インスタンスの取り消し

ここでは、サーバーを構成するアプリケーション・サーバー・インスタンスの取り消し方法について説明します。サーバー・インスタンスの取り消し操作では、サーバー・インスタンスはただちに停止します。それに対し、停止操作では、サーバー・インスタンスは現在の処理が完了してから停止します。

アプリケーション・サーバー・インスタンスの取り消しステップ

始める前に: MVS コンソールからは、サーバーの取り消し操作は実行できません。そのサーバーを構成する各サーバー・インスタンスに対して個別に取り消し操作を実行する必要があります。

サーバー・インスタンスを取り消すには、modify cancel コマンドを使用して以下のステップを実行します。

1. 次のコマンドを入力します。

```
modify server-instance,cancel
```

デーモンの取り消し

注: cancel コマンドは慎重に使用してください。

ここでは、デーモンの取り消し方法について説明します。デーモンに取り消し操作を実行すると、それによってシステム管理が停止し、さらにシステム操作の停止によってネーミング、IR、およびその他すべての制御領域が停止します。

デーモンの取り消しステップ

始める前に: デーモンに取り消し操作を実行すると、同じシステム上のすべての WebSphere for z/OS サーバーが取り消されます。

デーモンを取り消すには、以下のステップを実行します。

1. 次のコマンドのいずれかを入力します。

   ```
   cancel bbodmn.daemon01
   ```

   または

   ```
   cancel bbodmn.daemon01,norestart
   ```

   注: ARM が活動していて、それによりデーモンが再始動されるのを阻止したい場合は、「`norestart`」を指定する必要があります。

コールド・スタート WebSphere for z/OS

コールド・スタートの手順については、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

WebSphere for z/OS のホット・スタートとクイック・スタート

ホット・スタートとクイック・スタートの手順については、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

WebSphere for z/OS システム・サーバーのサービスの停止

ここでは、サーバーのサービスの停止方法について説明します。

サーバーのサービスの停止ステップ

始める前に: このタスクは、アプリケーションのサービスを停止したいときに実行します。通常、この場合のアプリケーションとはカスタマーが作成したアプリケーションを指します。システム・サーバーのサービスの停止は、インストール・システム全体を停止する場合以外は行いません。サーバーのサービスの停止操作には、サーバーを停止するほかに、オペレーターによって明示的なアクションがとられるまで、設定されている自動化によってサーバーが再始動されないようにすることが含まれます。

サーバーのサービスを停止するには、全システム上の、そのサーバーのすべてのサーバー・インスタンスを停止する必要があります。この操作は、SMUI 操作アプリケーションから実行するのが最も簡単です。詳しくは、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照してください。これはアプリケーション・サーバーのみに適用されます。

MVS コンソールからサーバーのサービスを停止するには、以下のステップを実行します。

1. 該当のサーバー・インスタンス (制御領域) を**停止**します。

   ---

2. 停止操作で停止しない場合は、サーバー・インスタンス (制御領域) を**取り消**します。

注: サーバー・インスタンス (制御領域) を停止または取り消すと、通常、サーバー領域が停止または取り消されます。サーバー領域が停止または取り消されない場合は、サーバー領域に対しても停止または取り消し操作を実行する必要があります。

ARM と再始動

ARM を使用してサーバーを再始動する場合は、次の点に注意します。

1. ARM が使用可能になっている状態でサーバーの取り消しまたは停止を実行すると、サーバーは同じ場所または別のシステム上で再始動します。
2. デーモンがすでに存在するシステム上でデーモンを始動すると、そのデーモンは終了します。
3. 他のすべてのサーバーは、構成に固定ポートが含まれていない場合、ダイナミック・ポート上に現れます。そのため、固定ポートはシスプレックス内で固有でなければなりません。

ネーム・スペースの内容の検査

ネーミング・ダンプ・ユーティリティーを使用して、ネーム・スペースの内容を検査することができます。このユーティリティーについては、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

複数の異なるサーバーおよびサーバー･インスタンスのエラー・ログ・ストリームのセットアップ

複数の異なるサーバーに対するエラー・ログ・ストリームのセットアップ方法については、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

WebSphere for z/OS サーバーを含む、ARM 登録アドレス・スペースの状況の表示

WebSphere for z/OS は、すべての制御領域が自動再始動管理 (ARM) 登録コマンドを発行する設定で出荷されます。ご使用のインストール・システムで ARM を使用可能にしている場合、以下の説明を参照してください。

ここでは、WebSphere for z/OS ランタイム環境内のすべての ARM 登録アドレス・スペース (サーバー・インスタンスのアドレス・スペースを含む) の状況を、ARM を使用して表示する方法について説明します。ARM に登録されているアドレス・スペースは、いずれも停止すると、ARM によって再始動されます。ただし、アドレス・スペースに対して取り消し操作が実行された場合は、再始動は行われません。

Application Server の各制御領域は ARM に登録されます。制御領域が異常終了したり、システムに障害が発生すると、ARM は障害の発生したアドレス・スペースを再始動しようとします。その際、ARM は、従属するアドレス・スペースがグループ化され、適切な順序で始動するようにします。通常、デフォルトの ARM ポリシーでは、Application Server は同じ場所で再始動されます。シスプレックスを使用している場合の、異なるシステム間の再始動を抑止するためのセットアップのガイドラインについては、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

ARM 登録アドレス・スペースの状況の表示ステップ

ARM の使用により、WebSphere for z/OS ランタイム環境内の、ARM に登録されたアドレス・スペース（サーバー・インスタンスのアドレス・スペースを含む) の状況を表示するには、以下のステップを実行します。

1. すべてのサーバーを初期化します。

---

2. 登録されているすべてのアドレス・スペース (サーバー・インスタンスのアドレス・スペースを含む) を表示するには、次のコマンドを入力します。

```
d xcf,armstatus,detail
```

個々のサーバーまたはサーバー・インスタンスの状況の表示

ここでは、WebSphere for z/OS ランタイム環境内の特定のサーバーまたはサーバー・インスタンスの状況を、ARM を使用して表示する方法について説明します。

個々のサーバーまたはサーバー・インスタンスの状況の表示ステップ

ARM の使用により、WebSphere for z/OS ランタイム環境内の特定のサーバーまたはサーバー・インスタンスの状況を表示するには、以下のステップを実行します。

自動再始動管理とランタイムに関する注記: ランタイムでは、次のような自動再始動管理の関与に注意してください。

1. サーバー・インスタンスが初期化されると、自動再始動管理に関連する各インスタンスの状況を表示することができます。登録されているすべてのアドレス・スペース (サーバー・インスタンスのアドレス・スペースを含む) を表示するには、次のコマンドを入力します。

   ```
   d xcf,armstatus,detail
   ```

   特定のサーバー・インスタンスの状況を表示するには、ジョブ名を指定して DISPLAY コマンドを発行します。たとえば、デーモン・サーバー・インスタンス (ジョブ BBODMN) の状況を表示するには、次のように入力します。

   ```
   d  xcf,armstatus,jobname=bbodmn,detail
   ```

2. サーバー・インスタンスに対して STOP、CANCEL、または MODIFY コマンドを発行する際には、WebSphere for z/OS サーバー・インスタンスに自動再始動管理がどのように関与するか注意する必要があります。

表 2. WebSphere for z/OS サーバー・インスタンスに関与する自動再始動管理の動作

発行コマンド	自動再始動管理の動作
STOP *address_space*	アドレス・スペースの再始動は行いません。

表 2. WebSphere for z/OS サーバー・インスタンスに関与する自動再始動管理の動作 (続き)

発行コマンド	自動再始動管理の動作
CANCEL *address_space*	アドレス・スペースの再始動は行いません。
CANCEL *address_space*, ARMRESTART	アドレス・スペースを再始動します。
MODIFY *address_space*,CANCEL	アドレス・スペースの再始動は行いません。
MODIFY *address_space*,CANCEL,ARMRESTART	アドレス・スペースを再始動します。

活動中のアドレス・スペースの表示

このコマンドは、活動中のアドレス・スペースを表示したい場合 (稼働している DB2 を知りたい場合など) に役立ちます。

活動中のアドレス・スペースの表示ステップ

活動中のすべてのアドレス・スペースを (リスト) 表示するには、次のステップを実行します。

1. 次のコマンドを入力します。

```
d a,l
```

活動中の応答の表示

MVS コンソールから活動中の応答を表示することによって、システムの活動を監視し、システムがオペレーターの応答を必要としているかどうかを判別することができます。

活動中の応答の表示

活動中のすべての応答を (リスト) 表示するには、次のステップを実行します。

1. 次のコマンドを入力します。

```
d r,r
```

DB2 の作業単位 (スレッド) の表示

ここでは、DB2 の作業単位 (スレッド) の表示方法について説明します。

DB2 の作業単位 (スレッド) の表示ステップ

DB2 の作業単位 (活動スレッド) を表示するには、次のステップを実行します。

1. 次のコマンドを入力します。

```
--db2 dis thread(*)
```

DB2 の未確定の作業単位 (スレッド) の表示

ここでは、DB2 の未確定の作業単位 (スレッド) の表示方法について説明します。

未確定の作業単位 (スレッド) とは、未確定状態にあるリカバリー単位 (UR)、つまり変更処理の実行 / 非実行が確定されていない 1 単位の変更セットのことです。RRS は、処理の調整を行うリソース・マネージャーから、コミットまたはバックアウトによる UR の解決を指示されるのを待ちます。詳しくは、*z/OS MVS プログラミング:リソース・リカバリー*, SA88-8582 を参照してください。

DB2 の未確定の作業単位 (スレッド) の表示ステップ

次のステップを実行して、DB2 の作業単位 (活動スレッド) を表示します。

1. 次のコマンドを入力します。

```
--db2 dis thread(*) type(indoubt)
```

次のメッセージが表示された場合、未解決のスレッドはありません。

```
DB2 No Indoubt Threads Found
```

CICS の作業単位の表示

詳しくは、*CICS Operations and Utilities Guide*, SC34-5717 を参照してください。

IMS の作業単位 (トランザクション) の表示

ここでは、IMS の作業単位 (トランザクション) の表示方法について説明します。

IMS の作業単位 (トランザクション) の表示ステップ

IMS の作業単位 (トランザクション) を表示するには、以下のステップを実行します。

1. 特定のトランザクションの状況を表示するには、次のコマンドを入力します。

   ```
   /dis tran trans-name
   ```

   ---

2. 特定のプログラムの状況を表示するには、次のコマンドを入力します。

   ```
   /dis prog program-name
   ```

   ---

3. 現在活動しているメッセージ処理領域 (MPR) の数を表示するには、次のコマンドを入力します。

   ```
   /DISPLAY ACTIVE REGION
   ```

   IMS コマンドについての詳細は、*IMS/ESA Summary of Operator Commands*, SC26-8766 を参照してください。

RRS の作業単位の表示

RRS の作業単位の表示方法については、*z/OS MVS プログラミング:リソース・リカバリー*, SA88-8582 を参照してください。

WebSphere for z/OS 操作に対するワークロード管理の使用

ここでは、WebSphere for z/OS の操作に使用できる一般的なワークロード管理 (WLM) タスクについて説明します。

注: WLM コマンドの有効範囲はシスプレックスです。したがって、アプリケーション環境を静止するコマンドを発行した場合、そのコマンドはシスプレックス内のすべてのシステムに作用します。WLM コマンドの使用時にはこの点に注意してください。

WLM アプリケーション環境の状況の表示

ここでは、アプリケーション環境の状況の表示方法について説明します。

注: WLM アプリケーション環境名は、個々のサーバー名と同じです。

アプリケーション環境の状況の表示ステップ

すべてのアプリケーション環境の状況を表示するには、次のステップを実行します。

1. 次のコマンドを入力します。

   ```
   d, wlm,applenv=*
   ```

   「*」と指定することによって、すべてのアプリケーション環境と状況が表示されます。

   ---

   display コマンドについての詳細は、*z/OS MVS システム・コマンドの要約*, SA88-8594 を参照してください。

例: 次に示すのは、display コマンドの使用例です。

```
- SY1     d wlm,applenv=*
   SY1     IWM029I  11.21.11  WLM DISPLAY 469
     APPLICATION ENVIRONMENT NAME     STATE      STATE DATA
     BBOABBOA                         AVAILABLE
     BBOASR1                          AVAILABLE
     BBOASR2                          AVAILABLE
     BBOASR3                          AVAILABLE
     BBOASR4                          AVAILABLE
     BBOASR5                          AVAILABLE
     BBOASR6                          AVAILABLE
     BBOASR7                          AVAILABLE
     BBOASR8                          AVAILABLE
     BBOASR9                          AVAILABLE
     CBINTFRP                         AVAILABLE
     CBNAMING                         AVAILABLE
     CBSYSMGT                         AVAILABLE
     PAAWYSV                          AVAILABLE
     PAAXFSV                          AVAILABLE
     PAAX1SV                          AVAILABLE
     PAAYYSV                          AVAILABLE
```

特定のアプリケーション環境の状況を表示するには、次のステップを実行します。

1. 次のコマンドを入力します。

   ```
   d wlm,applenv=bboasr1
   ```

   display コマンドについての詳細は、*z/OS MVS システム・コマンドの要約*, SA88-8594 を参照してください。

例: 次に示すのは、display コマンドの使用例です。

```
0- SY1     d wlm,applenv=bboasr1
   SY1     IWM029I  11.21.30  WLM DISPLAY 474
     APPLICATION ENVIRONMENT NAME      STATE     STATE DATA
     BBOASR1                       AVAILABLE
     ATTRIBUTES: PROC=BBOASR1S SUBSYSTEM TYPE: CB
```

上の例の PROC は、WLM がサーバー領域を始動するために使用する JCL PROC です。

display コマンドが発行されると、WebSphere for z/OS は以下の点について確認しようとします。

1. WLM アプリケーション環境名がサーバー名と一致しているかどうか
2. WLM アプリケーション環境の状況
3. アプリケーション環境に関連付けられている proc が、対応するサーバー領域用の proc かどうか

最も重要な情報はアプリケーション・サーバーの状態です。

表示される状態によって、次のことが示されます。

available
すべてが正常に動作していることを表します。アプリケーション・サーバーは使用可能です。

quiesced (q)
どのサーバー領域も始動しないことを表します。サーバーが quiesced (静止) 状態になるのは、(異常終了しているときなど) アプリケーションで問題が発生しているときだけです。静止状態の間、制御領域のサービスは停止されます。新しい要求は引き続き着信しますが、WLM は静止状態のサーバー領域は始動しません。

stopped
quiesced 状態と似ていますが、stopped 状態ではどのアプリケーション・サーバー領域も始動されません。制御領域に着信した要求は、処理されないまま制御領域にとどまります。サーバーが停止状態になるのは、サーバー領域に終了エラーが発生しているときです。10 分間に 3 つのアドレス・スペースが終了すると、サーバーは停止状態になります。その他の原因で、サーバーが停止することもあります。たとえば、オペレーターがサーバー領域に対して取り消し操作を行った場合です。

サーバー領域の取り消しは実行すべきではありません。サーバーのサービスを停止するには、サーバーを取り消すのではなく、制御領域を停止するようにします。

ワークロード管理とサーバー障害の処理

操作中、アプリケーションに繰り返しエラーが発生してアプリケーション・サーバー領域が終了すると、そのアプリケーションのアプリケーション環境はワークロード管理によって終了される場合があります。WebSphere for z/OS では、使用しようとしたアプリケーション環境に障害が発生していると、次のようなメッセージを発行します。

```
BBOU199E Unable to schedule work.  WLM application environment applenv has
         stopped.
```

アプリケーションで発生している問題を修正した後、VARY WLM コマンドの RESUME オプションを使用してアプリケーション環境を再始動する必要があります。

注: アプリケーション環境の有効範囲はシスプレックスです。したがって、WLM がアプリケーション環境を停止する場合、シスプレックス全体で停止されることになります。再開した場合は、シスプレックス内のすべてのシステム上でアプリケーション環境が再開されます。

WLM がアプリケーション環境をシャットダウンする場合、その理由はサーバー領域にエラーが発生しているためです。システムは原因の判別ができないため、環境をシャットダウンし、オペレーターのヘルプを要求します。シャットダウンしなかった場合、システムは*ストーム・ドレイン* と呼ばれるエラー・ループに入って続行します。トランザクションがすばやく終了するため、システムは正常に稼働しているように見えますが、実際にはエラーが発生しています。

ワークロード管理アプリケーション環境の検査および始動ステップ

ワークロード管理アプリケーション管理の検査と始動を行うには、以下のステップを実行します。

1. アプリケーション環境を表示するには、次のコマンドを入力します。

   ```
   d wlm,applenv=*
   ```

   ---

2. アプリケーション環境を始動するには、次のコマンドを入力します。

   ```
   v wlm,applenv=environment_name,resume
   ```

environment_name には、アプリケーション環境名を指定します。

アプリケーション環境が再開されたことを通知するメッセージがコンソールに発行されます。

停止状態から使用可能状態への復帰

シスプレックス内のあるサーバー領域がダウンして停止状態になった場合、アプリケーション環境の停止状態はシスプレックス全体に及ぶことに注意してください。この停止状態はインスタンス・レベルではなく、サーバー・レベルです。この状態になった場合、WLM は追加のサーバー領域を始動し、それによって進行中の作業を完了することはできません。WLM はこのエラーがランタイム・エラーか環境の問題かを判別できません。シスプレックス内のシステムはすべて同じであるため、1 つのシステムが停止した場合、他のシステムでも (現在は正常に稼働しているように見えても) 同じ障害が発生する可能性があります。

停止状態から使用可能状態への復帰ステップ

停止状態のシステムを使用可能状態に復帰させるには、以下のステップを実行します。

1. システムが停止状態になっている原因を判別します。次のいずれかが原因になっていると考えられます。
   - サーバー領域の proc の JCL エラー
   - WebSphere for z/OS ランタイムのバグ
   - アプリケーションのバグ
   - 他の環境に関する問題
   - 他の MVS 関連の問題

   問題の原因を判定できない場合は、弊社営業担当員にお問い合わせください。

   ---

2. 問題を解決します。

   ---

3. resume コマンドを発行して操作を再開します。

   WLM resume コマンドによって WLM が再開され、再開した WLM によってサーバー領域が始動されます。しかし、問題の修正を行わなかった場

合、アプリケーション環境はおそらくまた停止状態に戻ります。問題を修正してあれば、resume コマンドによってアプリケーション環境は使用可能状態に戻ります。

---

4. 次のコマンドを入力します。

   ```
   D WLM,APPLENV=applenv
   ```

   このコマンドによって、アプリケーション環境が活動していることを確認します。

第4章 z/OS または OS/390 サブシステムの操作上の考慮事項

この章では、WebSphere for z/OS に対して必要または推奨される z/OS または OS/390 サブシステムの操作上の考慮事項を説明します。以下の考慮事項はこれらのサブシステムを管理するシステム・プログラマーを対象としており、サブシステムのインストールや構成を指示するものではありません。

z/OS または OS/390 サブシステムの操作上の考慮事項

DB2 for z/OS または OS/390 の操作

DB2 操作のガイドライン: この節では、DB2 for z/OS または OS/390 の操作のためのガイドラインとヒントを提供します。Application Server による DB2 の使用法についての詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

- 会話を作成およびコミットすると、DB2 がデータをオフロードするときに多数の DB2 アクティビティーが発生します。場合によっては、別のログ・ボリュームを追加したり、ログをクリーンアップする必要が生じます。DB2 ログのサイズを検査してください。DB2 ログのスペースが使い尽くされるとプログラムが停止し、ログ・データ・セットを追加する必要が生じます。
- 構成の規模が拡大すると、デフォルト・バッファーを増加させる必要があります。バッファー・プールの増加に関する勧告については、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。DB2 用に 32K の一時ワークスペースがあるかどうか検査してください。ワークスペースは Application Server のインストール中に割り振っておく必要があります。ただし、ワークスペースが十分でない場合は、LDAP サーバー、システム管理サーバー、またはネーミング・サーバーの起動時に SQLCODE -904 戻りコードが戻される場合があります。
- 保守のために DB2 を停止させる必要がある場合は、WebSphere for z/OS も停止してください。Application Server では制御情報のために DB2 を使用します。したがって、Application Server ランタイム・サーバーを実行させるには DB2 が実行されている必要があります。
- DB2 スレッドを表示すると、相関 ID が要求側のクライアントの MVS ユーザー ID と等しくなっています。

- DB2 テーブルがボリュームを満杯にしたことを示すエラー・コードを受け取った場合、ソリューションとして DB2 テーブルをより大きいボリュームに移動するか、または可能であればボリュームにスペースを追加します。

CICS 操作

CICS 操作のガイドライン: この節では、WebSphere for z/OS と関連する場合の CICS 操作のガイドラインとヒントを提供します。

- **サンプル・アプリケーションを実行するための CICS 領域の構成:** CICS の構成に関する詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: J2EE アプリケーションのアセンブル*, SA88-8654 の *『サンプル・アプリケーション実行のための CICS 領域の構成』* の節を参照してください。RRMS 属性が「Yes」に設定されていること、また、NETNAME が CICS アダプターを実行する WebSphere for z/OS サーバーの名前であることを確認してください。

IMS 操作

この節では、Application Server と関連する場合の IMS 操作のガイドラインとヒントを提供します。WebSphere for z/OS と関連する場合の IMS 操作のガイドラインについては、*IMS/ESA Operations Guide*, SC26-8741 も参照してください。

- IMS を使用するときは、WebSphere for z/OS トランザクションで発行される可能性のある IMS トランザクションの合計数を処理できるだけの充分な数のメッセージ処理領域をセットアップする必要があります。1 つの Application Server トランザクションで 3 つまたはそれ以上のトランザクションを IMS に駆動することができます。これらのトランザクションを正常に処理するには、IMS に要求の処理に使用できる追加のメッセージ処理領域が必要となる場合があります。一般に、IMS は生成された IMS トランザクションの数と同数の開始メッセージ処理領域を必要とします。詳しくは、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652、および *WebSphere Application Server V4.0 for z/OS and OS/390: J2EE アプリケーションのアセンブル*, SA88-8654 を参照してください。
- また、以下の parlim がインストール時に設定されていない場合は設定する必要があります。

  ```
  assign parlim 0 tran tranname
  ```

  並列しきい値 「0」 を割り当てるということは、一度に実行できるトランザクションの数に限界がないことを示します。これは、IMS 生成時、構成中に TRANSACT ステートメントを使用して指定することもできます。この値を指定

するとトランザクションを多重的 (または並列的) にスケジュールすることができるため、多数のトランザクションを同時に実行できます。IMS を正常に動作させるには、複数のメッセージ処理領域をセットアップする以外にこの値も設定する必要があります。詳しくは、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

- **IMS OTMA サポート**: サーバーに論理リソース・マネージャーを定義するときは、IMS_OTMA_PAA を選択する必要があります。IMS 制御領域は、OTMA インターフェースをアクティブにした状態で開始する必要があります。Application Server が適切な IMS に接続を行うため、IMS プロシージャー・パラメーターは OTMA=YES を指定する必要があります。また、IBM によってサポートされる OTMA インターフェースに応じて XCF グループ名を定義する必要があります。IMS-OTMA 手続き型アプリケーション・アダプターのセットアップに関する詳細については、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 の説明を参照してください。
- **IMS OTMA サポート・ランタイムのヒント:** システムの再 IPL を実行する必要がある場合は、IMS OTMA SVC (監視プログラム呼び出し) を再定義する必要があります。これを行うには、実行ステートメント PGM=DFSYSVI0 を実行します。これにより必要な SVC が動的にインストールされます。SVC を再定義しない場合、IMS OTMA サポートを実行しようとすると、F92 異常終了を受け取り、サーバーがダウンします。

RRS 操作

RRS 操作のガイドラインについては、47ページの表6 および *z/OS MVS プログラミング:リソース・リカバリー*, SA88-8582 を参照してください。

RRS 操作のヒント:

- ログ・ストリームをカップリング・ファシリティーに構成した場合は、ログ・ストリームをモニターしてオフロードが発生していないことを確認してください。RRS はリカバリー・ログがオフロードしない場合の方がパフォーマンスが向上します。

  注: RRS ログを適切にサイジングすることが重要です。小さすぎるとロガーが頻繁にログをオフロードしすぎるため、スループットが減少します。大きすぎるとカップリング・ファシリティーがオーバーフローする可能性があります。

- メインおよび遅延 (アクティブ・データまたはライブ・データのみを含む) ログをカップリング・ファシリティーに保持してください。CF 定義がオーバーフローしないようにしてください。

  注: ログ・レコードが書き込まれるまでコミットは発生しません。
- ワークロードを安定化するまでは、アーカイブ・ログを使用することをお勧めします。アーカイブ・ログを構成すると、RRS が無条件にこれを使用します。ただし、アーカイブ・ログの使用に対してはパフォーマンス上のペナルティーがあります。

ワークロード管理 (WLM) 操作

WLM 操作のガイドラインについては、*z/OS MVS 計画:ワークロード管理* および *z/OS MVS プログラミング:ワークロード管理サービス*, SA88-8585 を参照してください。

第5章 WebSphere for z/OS のバックアップのガイドラインと手順

この章では、WebSphere for z/OS のバックアップのガイドラインと手順について説明します。

OS/390 ランタイム環境のバックアップ

WebSphere for z/OS システムのバックアップのガイドライン

WebSphere for z/OS システムの各部分のバックアップをとるには、次のガイドラインに従ってください。

1. 必ず、RRS の RMDATA ログのバックアップをとってください。このバックアップをとっておかないと、障害が発生したときに、RRS をコールド・スタートしなければならなくなります。
2. アーカイブ・ログの保存期間を 1 日にしておきます。
3. ユーザー独自のバックアップ手順に従って、ネーミングおよびインターフェース・リポジトリー・データを含む、LDAP データベースのバックアップをとります。

   注: LDAP データを復元する場合は、必ず次のものとの調整を行ってください。

   - 統合ネーミング・スペース内の他の WebSphere システム (他のシステムとの調整を行わないと、ネーミング・スペースの一貫性が失われます)。
   - システム管理データベース (LDAP テーブルと同じ時点で SM テーブルを復元する必要があります)。
4. 標準のバックアップ手順に以下を取り込みます。
   - WebSphere for z/OS proclib
   - WebSphere for z/OS loadlib
   - WebSphere for z/OS 環境ファイル
   - 管理アプリケーションがアプリケーションを書き込むディレクトリー (CBCONFIG 環境変数の値。デフォルトは `/WebSphere390/CB390`)。
5. 以下の DB2 for z/OS または OS/390 の表に含まれる参照コレクション・データのバックアップをとります。

- BBO.RCTABLE
- BBO.KRCTABLE
- BBO.RCHMTABLE

6. ユーザー所有のアプリケーションの実行可能プログラム、およびバインディングのバックアップをとります。
7. 会話を活動化すると、/*path*/controlinfo/envfile/*sysplex*/*server_instance*/backup/ に、各サーバー・インスタンスの現在の環境ファイルのバックアップがシステム管理によって自動的に作成されます。

   ここで、

   path
   CBCONFIG 環境変数の値です (デフォルトは /WebSphere390/CB390)。

   sysplex
   ユーザーのシスプレックスの名前です。

   server_instance
   はサーバー・インスタンスの名前です。

   バックアップ・ファイルの名前には、タイム・スタンプが含まれています。バックアップ・ディレクトリーが満杯になれば、古いバックアップ・ファイルを削除することができます。
8. コールド・スタートの準備をする場合、システム管理は、XML 形式で、/*path*/configuration/backup/ にある制御情報のバックアップをとります。

   ここで、

   path
   CBCONFIG 環境変数の値です (デフォルトは /WebSphere390/CB390)。

   バックアップ・ファイルの名前には、タイム・スタンプが含まれています。バックアップ・ディレクトリーが満杯になれば、古いバックアップ・ファイルを削除することができます。
9. 単一のサーバー・インスタンスのバックアップをとりたい場合は、管理アプリケーションのエクスポート / インポート機能を使用することができます。詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: J2EE アプリケーションのアセンブル*, SA88-8654 を参照してください。
10. システム管理データベースに関しては、次の表に従ってバックアップをとるものを決定します。

前提条件	バックアップをとるもの	
管理者を追加した場合	次のテーブル・スペース	
	BBOMDB01.BBOMS51	
	BBOMDB01.BBOMS54	
新しい会話を作成したか、会話をコミットした場合	次のテーブル・スペース	
	BBOMDB01.BBOMS00	BBOMDB01.BBOMS56
	BBOMDB01.BBOMS02	BBOMDB01.BBOMS58
	BBOMDB01.BBOMS04	BBOMDB01.BBOMS60
	BBOMDB01.BBOMS06	BBOMDB01.BBOMS62
	BBOMDB01.BBOMS10	BBOMDB01.BBOMS64
	BBOMDB01.BBOMS15	BBOMDB01.BBOMS66
	BBOMDB01.BBOMS19	BBOMDB01.BBOMS68
	BBOMDB01.BBOMS23	BBOMDB01.BBOMS70
	BBOMDB01.BBOMS25	BBOMDB01.BBOMS72
	BBOMDB01.BBOMS27	BBOMDB01.BBOMS74
	BBOMDB01.BBOMS29	BBOMDB01.BBOMS76
	BBOMDB01.BBOMS31	BBOMDB01.BBOMS80
	BBOMDB01.BBOMS33	BBOMDB01.BBOMS81
	BBOMDB01.BBOMS35	BBOMDB01.BBOMS82
	BBOMDB01.BBOMS37	BBOMDB01.BBOMS83
	BBOMDB01.BBOMS39	BBOMDB01.BBOMS84
	BBOMDB01.BBOMS41	BBOMDB01.BBOMS85
	BBOMDB01.BBOMS43	BBOMDB01.BBOMS86
	BBOMDB01.BBOMS45	BBOMDB01.BBOMS87
	BBOMDB01.BBOMS48	BBOMDB01.BBOMS90
	BBOMDB01.BBOMS52	

前提条件	バックアップをとるもの	
会話を活動化した場合	次のテーブル・スペースまたはデータベース	
	BBOMDB01.BBOMS00	BBOMDB01.BBOMS53
	BBOMDB01.BBOMS02	BBOMDB01.BBOMS55
	BBOMDB01.BBOMS04	BBOMDB01.BBOMS56
	BBOMDB01.BBOMS06	BBOMDB01.BBOMS58
	BBOMDB01.BBOMS10	BBOMDB01.BBOMS60
	BBOMDB01.BBOMS15	BBOMDB01.BBOMS62
	BBOMDB01.BBOMS19	BBOMDB01.BBOMS64
	BBOMDB01.BBOMS23	BBOMDB01.BBOMS66
	BBOMDB01.BBOMS25	BBOMDB01.BBOMS68
	BBOMDB01.BBOMS27	BBOMDB01.BBOMS70
	BBOMDB01.BBOMS29	BBOMDB01.BBOMS72
	BBOMDB01.BBOMS31	BBOMDB01.BBOMS74
	BBOMDB01.BBOMS33	BBOMDB01.BBOMS76
	BBOMDB01.BBOMS35	BBOMDB01.BBOMS80
	BBOMDB01.BBOMS37	BBOMDB01.BBOMS81
	BBOMDB01.BBOMS39	BBOMDB01.BBOMS82
	BBOMDB01.BBOMS41	BBOMDB01.BBOMS83
	BBOMDB01.BBOMS43	BBOMDB01.BBOMS84
	BBOMDB01.BBOMS45	BBOMDB01.BBOMS85
	BBOMDB01.BBOMS48	BBOMDB01.BBOMS86
	BBOMDB01.BBOMS52	BBOMDB01.BBOMS87
	LDAP データベース	BBOMDB01.BBOMS90
		BBOMDB01.BBOSLS01
		BBOMDB01.BBOSLS02

注:

a. WebSphere for z/OS のテーブル・スペースのバックアップと、他の WebSphere システム・マネージャー (Windows NT 上にあるものなど) との調整を行ってください。

b. 他のシステム (たとえば Windows NT) を使用してネーミング・ツリーを統合した場合は、LDAP データベースのバックアップを、Windows NT 上のバックアップと同期化しなければなりません。そうしないと、統合したネーミング・スペースの一貫性が保たれません。

第6章 WebSphere for z/OS および従属するサブシステムのモニタリングおよびリカバリー

この章では、WebSphere for z/OS およびこれに従属するサブシステムをモニターおよびリカバリーする方法について説明します。

WebSphere for z/OS および従属するサブシステムの始動順序

次の表は、WebSphere for z/OS に従属するサブシステムの始動順序を示しています。同一行に示されているサブシステムについては、同時または任意の順序で始動することができます。z/OS または OS/390 に従属するサブシステムの始動順序についての詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

始動順序	サブシステム	サブシステム	サブシステム
1	ワークロード管理 (WLM) が自動的に始動される		
2	RACF		
3	システム・ロガーが自動的に始動される		
4	RRS	USS (TCP/IP が始動されるまでは完了しないが、TCP/IP より前に始動することができる)	
5	VTAM	TCP/IP	
6	APPC	TSO	
7	DB2	CICS	IMS
8	LDAP	NFS (シスプレックスでのみ必要) **注:** 共用 HFS を使用する場合、NFS は不要。	

始動順序	サブシステム	サブシステム	サブシステム
9	WebSphere for z/OS • JVM: Application Server および WebServer 内で始動される • LE: Application Server および WebServer 内で始動される	WebServer (サーブレット)	

自動化およびリカバリーのシナリオとガイドライン

次の節では、WebSphere for z/OS および WebSphere for z/OS で使用するサブシステムのモニター方法およびリカバリー方法について説明します。始動、シャットダウン、およびリカバリーの手順、並びにこれらのシナリオを示します。また、サブシステムが稼働しているかどうかの判別方法、および詳細情報の入手方法についても説明します。

APPC の自動化およびリカバリーのシナリオ

表 3. APPC の自動化およびリカバリーのシナリオ

タスク	APPC の自動化およびリカバリーのシナリオ
始動	APPC は、WebSphere for z/OS より先に始動する必要があります。理論上は、Application Server を APPC より先に*始動*することができますが、この場合、IMS APPC LRMI と関連付けられているオブジェクトがコンテナー内でディスパッチされないことが条件になります。Application Server より先に APPC が始動されていないと、IMS との会話で APPC コネクターを使用する必要がある場合に、接続が見つかりません。CICS の場合、APPC/MVS を始動する必要はありません。APPC を VTAM の後に始動する必要はありません。
シャットダウン	始動手順と逆の順序です。Application Server、APPC、VTAM の順でシャットダウンします。

表 3. APPC の自動化およびリカバリーのシナリオ (続き)

タスク	APPC の自動化およびリカバリーのシナリオ
障害が発生した場合の未コミットまたは未確定のトランザクションの処理	APPC を使用して通信している場合、APPC で障害が発生したときは、次のことを実行してください。 1. APPC 接続を使用しているすべてのサーバーをシャットダウンする。 2. APPC を再始動する (APPC が完全に失敗した場合)。 3. WebSphere for z/OS サーバーを再始動する。 **注:** APPC は自分で再同期を行います。トランザクションが未確定の場合、APPC が再始動されるまで IMS はそのままの状態になります。IMS のリカバリーは RRS を介して行われます。RRS は、停止されるまでに通信していた各サブシステムとのハンドシェーキングにより、未確定作業を解決します。CICS を使用している場合は、CICS 独自の調整方式が使用されます。
APPC が稼働中かどうかの判別方法	`DISPLAY APPC,LU,ALL` コマンドを発行します。APPC が活動中でない場合は、その旨の応答があります。また、Application Server または IMS (あるいはその両方) が使用している LU の状況が活動中でないと、APPC の作業は失敗します。
APPC がダウンした場合に Application Server で発生する状況	IMS APPC PAA を使用しようとするオブジェクトは、すべて失敗します。コンテナーの代わりに実行されているサーバー領域が APPC を使用しようとすると、APPC ALLOCATE 要求が試行され、これが失敗したことを示す `C9C24C05` エラーを受け取る可能性があります。この領域に関連するログには、APPC エラーに関する追加の診断情報が含まれており、この情報は、APPC で発生した問題を特定するために使用できます。
詳細情報の入手方法	• *z/OS MVS 計画 : 操作*, SA88-8573 • *z/OS MVS 計画: APPC/MVS 管理*, SA88-8571 • *z/OS MVS プログラミング:リソース・リカバリー*, SA88-8582

WLM の自動化およびリカバリーのシナリオ

表 4. ワークロード管理 (WLM) の自動化およびリカバリーのシナリオ

タスク	WLM の自動化およびリカバリーのシナリオ
始動	システムの IPL を実行すると、z/OS または OS/390 によって WLM が自動的に始動されます。したがって、始動する必要はありません。

表 4. ワークロード管理 (WLM) の自動化およびリカバリーのシナリオ (続き)

タスク	WLM の自動化およびリカバリーのシナリオ
シャットダウン	WLM をシャットダウンすることはできません。
WebSphere for z/OS サーバー領域の破局的故障の処理方法	WebSphere for z/OS サーバー領域の破局的故障が発生した場合は、次に示す WLM resume コマンドを使用します。 `V WLM,APPLENV=XYZ,RESUME`
詳細情報の入手方法	• *z/OS MVS 計画:ワークロード管理*, SA88-8574 • *z/OS MVS プログラミング:ワークロード管理サービス*, SA88-8585

RACF の自動化およびリカバリーのシナリオ

表 5. RACF の自動化およびリカバリーのシナリオ

タスク	RACF の自動化およびリカバリーのシナリオ
始動	RACF がインストールされている場合、RACF は IPL の一環として始動されます。
シャットダウン	RACF をシャットダウンすることはできません。
RACF が稼働中かどうかの判別方法	RACF の状況を表示するには、`RACF SETROPTS` コマンドを使用します。
RACF がダウンした場合に Application Server で発生する状況	RACF は、フェイルセーフ・モードに入ります。つまり、アクセスする各リソースについて、オペレーターに確認を求めるプロンプトが表示されます。通常、RACF がダウンした場合、システムの IPL が実行されます。
RACF がダウンした場合にその他のサブシステムで発生する状況	サブシステムの種類および RACF での障害の状況によって異なります。
詳細情報の入手方法	• *z/OS SecureWay Security Server (RACF) システム・プログラマーのガイド*, SA88-8611 • *z/OS SecureWay Security Server (RACF) セキュリティー管理者のガイド*, SA88-8613

RRS の自動化およびリカバリーのシナリオ

表 6. RRS の自動化およびリカバリーのシナリオ

タスク	RRS の自動化およびリカバリーのシナリオ
始動	RRS より先にシステム・ロガーを始動する必要があります。 **注:** システム・ロガーを始動する前に RRS を始動しようとすると、システム・ロガーを先に始動する必要があることを示すエラー・メッセージが表示されます。 RRS は、WebSphere for z/OS より先に始動する必要があります。RRS は自動的には始動されません。RRS が自動的に始動されるのは、自動再始動マネージャー (ARM) に登録されていて、なおかつ ARM が稼働中の場合だけです。RRS を始動するには、次に示す `start` コマンドを発行します。 `start rrs` **注:** cancel コマンドを発行した場合、RRS は自動的には再始動されません。したがって、RRS を取り消した場合や ARM が稼働中でない場合は、RRS を手動で再始動する必要があります。
シャットダウン	RRS をシャットダウンするには、RRS を始動したときとは逆の順序で手順を実行します。Application Server、RRS、システム・ロガーの順でシャットダウンします。RRS の停止に使用できる制御された方法はありません。最善の方法を次に示します。 1. Application Server を静止する。 2. Application Server をシャットダウンする。 3. RRS を取り消す。 **注:** 必要な場合は、WebSphere for z/OS に使用している DB2 を停止してから、RRS を取り消してください。 RRS を取り消すには、次のコマンドを発行します。 `setrrs cancel`
障害が発生した場合の未コミットまたは未確定のトランザクションの処理	未コミット・トランザクションの表示および未確定トランザクションの解決には、RRS システム管理パネルを使用します。RRS の RM パネルを使用してリソース・マネージャーを表示し、すべてのリカバリー単位 (UR) を表示したり、UR をフィルターに掛けたりしてから、未確定作業を解決することができます。未コミット・トランザクションを解決することはできません。RRS で管理されているすべてのトランザクションを表示することができます。

表 6. RRS の自動化およびリカバリーのシナリオ (続き)

タスク	RRS の自動化およびリカバリーのシナリオ
RRS が稼働中かどうかの判別方法	display コマンドを次のように使用します。 `d a,atrrs` `atrrs` は、Application Server と共に出荷されるデフォルトの RRS proc の名前です。RRS を始動するときに使用した proc 名を指定してください。アドレス・スペースは proc から取得されます。
RRS がダウンした場合に Application Server で発生する状況	RRS がダウンすると、RRS が再始動されるまで Application Server はハングします。RRS は基本オペレーティング・システムの一部なので、RRS にはフォールト・トレランス機能が組み込まれており、弾力性があります。頻繁に障害が発生しても、オペレーターはあまり心配する必要がありません。
RRS がダウンした場合にその他のサブシステムで発生する状況	RRS は、z/OS または OS/390 トランザクション・モニターの役割を担います。RRS を取り消すと、RRS を使用している各サブシステム (WebSphere for z/OS、DB2、IMS など) で問題が発生します。RRS を取り消す場合は、それによってどのような状況が発生するかを理解しておく必要があります。
詳細情報の入手方法	• *z/OS MVS プログラミング:リソース・リカバリー*, SA88-8582

USS の自動化およびリカバリーのシナリオ

表 7. UNIX システム・サービス (USS) の自動化およびリカバリーのシナリオ

タスク	USS の自動化およびリカバリーのシナリオ
始動	USS は BCP の永続コンポーネントであり、IPL 時に自動的に始動されます。
シャットダウン	USS にはシャットダウン機能がないため、いつでも使用可能です。
障害が発生した場合の未コミットまたは未確定のトランザクションの処理	実際のところ、トランザクション・データと見なされる唯一のデータは、HFS に保管されるデータです。
USS が稼働中かどうかの判別方法	USS は常に使用可能です。

表 7. UNIX システム・サービス (USS) の自動化およびリカバリーのシナリオ (続き)

タスク	USS の自動化およびリカバリーのシナリオ
USS がダウンした場合に Application Server で発生する状況	USS で障害が発生した場合は、システムの再 IPL を実行する必要があります。Application Server はエラーを受け取って終了します。
USS がダウンした場合にその他のサブシステムで発生する状況	USS で障害が発生した場合は、システムの再 IPL を実行する必要があります。
詳細情報の入手方法	• *z/OS UNIX システム・サービス 計画*, GA88-8639

TCP/IP の自動化およびリカバリーのシナリオ

表 8. TCP/IP の自動化およびリカバリーのシナリオ

タスク	TCP/IP の自動化およびリカバリーのシナリオ
始動	TCP/IP は、WebSphere for z/OS より先に始動する必要があります。
シャットダウン	Application Server をシャットダウンしてから、TCP/IP をシャットダウンしてください。
障害が発生した場合の未コミットまたは未確定のトランザクションの処理	未コミット・メソッドでは、そのメソッドに対する応答の送信が失敗した場合、トランザクションがロールバックされます。その他のトランザクションは、タイムアウトになるまで待ちます。
TCP/IP が稼働中かどうかの判別方法	display コマンドを使用して、TCP/IP proc を探し出してください。
TCP/IP がダウンした場合に Application Server で発生する状況	TCP/IP がダウンした場合、そのシステム上の Application Server を再始動する必要があります。ソケット・レイヤーが壊れたので、SVC ダンプを受け取ります。
TCP/IP がダウンした場合にその他のサブシステムで発生する状況	TCP/IP がダウンした場合、セッションは中断され、トランザクションは前述の説明どおりに処理されます。 **注:** TCP/IP が再び活動化されても Application Server はこれを認識できないので、再始動する必要があります。

DB2 の自動化およびリカバリーのシナリオ

表 9. DB2 の自動化およびリカバリーのシナリオ

タスク	DB2 の自動化およびリカバリーのシナリオ
始動	DB2 は、RRS の始動後、LDAP、NFS、および WebSphere for z/OS より先に始動する必要があります。
シャットダウン	始動順序とは逆の順序です。

障害が発生した場合の未コミットまたは未確定のトランザクションの処理	RRS パネルを使用して解決します。*z/OS MVS プログラミング :リソース・リカバリー*, SA88-8582 を参照してください。DB2 の未確定作業を解決する場合は、RRS パネルを使用することを推奨します。このパネルでは、トランザクションにインタレストを持つすべてのリソース・マネージャーを表示することができるからです。DB2 を使って未確定作業を解決することもできます。次のコマンド、
	`DISPLAY THREAD(*) TYPE(INDOUBT)`
	を使用すると、未確定のスレッドに関する DB2 の情報を表示することができます (未確定のスレッドが多すぎる場合は、システム・ログで情報を参照することができます)。この場合、表示される情報には 「nid」 という DB2 ID が付きます。この nid をコピーして、次のコマンドに貼り付けます。
	`-RECOVER INDOUBT (RRSAF) ACTION(COMMIT) NID(B1D379D17ED6CF900000009401010000)`
	ここで、nid は display コマンドからコピーした ID です。トランザクションをロールバックするには、次のコマンドを発行します。
	`-RECOVER INDOUBT (RRSAF) ACTION(ABORT) NID(B1D379D17ED6CF900000009401010000)`

DB2 が稼働中かどうかの判別方法	DB2 アドレス・スペースを表示するには、display コマンドを使用します。
詳細情報の入手方法	• xiiページの『関連情報の入手先』に記載されている DB2 資料を参照してください。

CICS の自動化およびリカバリーのシナリオ

表 10. CICS の自動化およびリカバリーのシナリオ

タスク	CICS の自動化およびリカバリーのシナリオ
始動	CICS 対応の WebSphere for z/OS アプリケーション制御サーバー領域に対してワークフローを実行するには、CICS をインストールおよび初期化して、始動しておく必要があります。
シャットダウン	支援記憶装置として CICS を使用している WebSphere for z/OS アプリケーション制御領域をシャットダウンしてから、CICS サービスをシャットダウンします。
障害が発生した場合の未コミットまたは未確定のトランザクションの処理	処理中にエラーが発生した場合、CICS および Application Server はどちらも、基礎を形成する RRS サブシステムを介して、登録済みのインタレストに対するロールバック通知を処理します。未コミット・トランザクションの場合は、ロールバックが必要であることが RRS からすべての参加プログラムに対して通知され、登録されているプログラムごとに通常のロールバック処理が行われます。未確定トランザクションの場合は、WebSphere for z/OS アプリケーション制御 / サーバー領域をリサイクルして、CICS 内で保留中のトランザクションを解放する必要があります。
CICS が稼働中かどうかの判別方法	インストール・システムによって異なります。
Application Server がダウンした場合に CICS で発生する状況	Application Server がダウンすると、次のいずれかの状況が発生します。 1. 現在 Application Server と CICS が関与している作業単位がある場合、前述のとおりの RRS 処理が行われます。この場合、アプリケーション制御サーバー領域をリサイクルして、CICS 内で保留中のトランザクション作業を解放する必要があります。 2. 現在 Application Server と CICS が作業単位に関与していない場合、CICS には影響はありません。
CICS がダウンした場合にその他のサブシステムで発生する状況	ありません。
詳細情報の入手方法	• *CICS Operations and Utilities Guide*, SC34-5717

IMS の自動化およびリカバリーのシナリオ

表 11. IMS の自動化およびリカバリーのシナリオ

タスク	IMS の自動化およびリカバリーのシナリオ
始動	IMS 対応の WebSphere for z/OS アプリケーション制御サーバー領域に対してワークフローを実行するには、IMS をインストールおよび初期化して、始動しておく必要があります。
シャットダウン	支援記憶装置として IMS を使用している WebSphere for z/OS アプリケーション制御領域をシャットダウンしてから、IMS サービスをシャットダウンします。
障害が発生した場合の未コミットまたは未確定のトランザクションの処理	処理中にエラーが発生した場合、IMS および Application Server はどちらも、基礎を形成する RRS サブシステムに基づいて、登録済みのインタレストに対するロールバック通知を処理します。未コミット・トランザクションの場合は、ロールバックが必要であることが RRS からすべての参加プログラムに対して通知され、登録されているプログラムごとに通常のロールバック処理が行われます。未確定トランザクションの場合は、WebSphere for z/OS アプリケーション制御 / サーバー領域をリサイクルして、IMS MPR 内で保留中のトランザクションを解放する必要があります。
IMS が稼働中かどうかの判別方法	インストール・システムによって異なります。
Application Server がダウンした場合に IMS で発生する状況	Application Server がダウンすると、次のいずれかの状況が発生します。 1. 現在 Application Server と IMS が関与している作業単位がある場合、前述のとおりの RRS 処理が行われます。この場合、アプリケーション制御サーバー領域をリサイクルして、IMS MPR 内で保留中のトランザクション作業を解放する必要があります。 2. 現在 Application Server と IMS が作業単位に関与していない場合、IMS には影響はありません。
IMS がダウンした場合にその他のサブシステムで発生する状況	ありません。
詳細情報の入手方法	• *IMS/ESA オペレーター用解説書*, SD88-7127

LDAP の自動化およびリカバリーのシナリオ

表 12. LDAP の自動化およびリカバリーのシナリオ

タスク	LDAP の自動化およびリカバリーのシナリオ
始動	WebSphere for z/OS で使用される LDAP は、「ローカル・バックエンド」という Application Server のアドレス・スペース内で実行されます。このサポートでは、LDAP クライアント API をフロントエンド、およびデータベース実装をバックエンドとして使用し、これらを完全に Application Server ネーミング・サーバーおよびインターフェース・リポジトリー内で実行します。ネーミングおよびインターフェース・リポジトリーは、OMVS および DB2 を*活動化*してから、始動する必要があります。LDAP サーバーを実行するには、TCP/IP、OMVS、および DB2 を先に活動化しておく必要があります。 **注:** サポートされる LDAP モードは、次の 2 つです。 1. ローカル LDAP バックエンド。 2. リモート LDAP サーバー。 Application Server 環境を設定し、DB2、TCP/IP、およびリモート・サーバーが活動化された状態で、WebSphere for z/OS を始動する必要があります。
シャットダウン	ネーミングとインターフェース・リポジトリーをシャットダウンしてから、OMVS および DB2 をシャットダウンします。LDAP サーバーの場合は、LDAP サーバーをシャットダウンしてから、TCP/IP、DB2、OMVS の順でシャットダウンします。
障害が発生した場合の未コミットまたは未確定のトランザクションの処理	処理中に障害が発生した場合、ネーミングおよびインターフェース・リポジトリーは、RRS を介して、DB2 に対してロールバックを直接発行します。この結果、LDAP コードによって実行された作業は、このロールバック要求に基づいてロールバックされます。LDAP サーバーの場合は、AUTOCOMMIT を「NO」に設定してください。これにより、トランザクションでエラーが発生した場合、このトランザクションはロールバックされます。この結果、LDAP 操作の原子性特性が保証されます。

表 12. LDAP の自動化およびリカバリーのシナリオ (続き)

タスク	LDAP の自動化およびリカバリーのシナリオ
LDAP が稼働中かどうかの判別方法	WebSphere for z/OS の場合、ネーミングとインターフェース・リポジトリーが稼働しているときは、LDAP が稼働しています。LDAP サーバーの場合、開始済みのタスクを LDAP サーバー用に使用するときは、開始済みのタスクが実行中であることを SDSF を使って確認してください。開始済みタスクの出力ログを調べて、エラー・メッセージが表示されているか確認してください。あるいは、LDAPSRCH コマンド (TSO から) または LDAPSEARCH コマンド (USS シェルから) を使って簡易検索を実行することにより、LDAP サーバーが実行されているかどうかを確認することができます。
LDAP がダウンした場合に Application Server で発生する状況	• MOFW Application Server 領域では、LDAP は Application Server のアドレス・スペース内で実行されるため、問題ありません。サーバーがダウンすると、LDAP もダウンします。 • J2EE サーバー領域では、LDAP サーバーは Application Server 領域の外で実行される独立したサーバーになるので、LDAP サーバーが**活動化されている必要があります**。
LDAP がダウンした場合にその他のサブシステムで発生する状況	z/OS または OS/390 サブシステムの多くは LDAP に依存しませんが、これは将来変更される可能性があります。LDAP サーバーを介して LDAP にアクセスする場合、シスプレックス内に活動中の LDAP サーバーがあるときは、(シスプレックス対応 DNS を使用して) このサーバーに LDAP 要求が送られるように、LDAP サーバーを構成することができます。別の方法としては、LDAP を使用する必要があるサブシステムでは、1 次サーバーにアクセスできない場合の接続先として、バックアップ LDAP サーバーを構成することができます。この場合、アプリケーションでは、1 次サーバーと同じ複製メカニズムで処理された、完全に同一のデータをバックアップ・サーバーから取得できることを想定します。現在、LDAP サーバーはマスター / スレーブ複製メカニズムをサポートしていますが、DB2 データ共用を使用してシスプレックス・サーバーを複製することもできます。
詳細情報の入手方法	• ネーミングとインターフェース・リポジトリーについては、WebSphere for z/OS 資料を参照してください。 • LDAP サーバーについては、*z/OS SecureWay Security Server LDAP Server Administration and Use*, SC24-5923 を参照してください。

NFS の自動化およびリカバリーのシナリオ

注: NFS は、OS/390 R8 でファイル共用システムとして使用されています。共用 HFS は、OS/390 R9 以降で使用されています。次のコメントは、NFS のランタイム使用に関連しているものであり、アプリケーション開発時の使用を目的としていません。

表 13. NFS の自動化およびリカバリーのシナリオ

タスク	NFS の自動化およびリカバリーのシナリオ
始動	NFS クライアントは、USS ファイル・システムの初期化中に始動されて、NFS コロニー・アドレス・スペースで実行されます。NFS クライアント用の FILESYSTYPE parmlib ステートメントが SYS1.PARMLIB(BPXPRMxx) メンバーに含まれている必要があります。
シャットダウン	システム・オペレーターは、OS/390 NFS クライアントのアドレス・スペース名を指定して変更オペレーター・コマンド `STOP` を発行することにより、NFS クライアントを正常に停止させることができます。`STOP` コマンドが失敗して NFS クライアントを正常に停止させることができなかった場合、オペレーターは、`CANCEL` コマンドを発行することにより、異常終了を強制することができます。
NFS が稼働中かどうかの判別方法	ディレクトリー `/usr/lpp/NFS` で nfsstat ユーティリティーを実行します。
NFS がダウンした場合に Application Server で発生する状況	新たにサーバーを始動しようとする試みは失敗します。環境変数にアクセスしようとする試みは失敗します。
NFS がダウンした場合にその他のサブシステムで発生する状況	その他のサブシステムは、問題なく稼働し続けます。
詳細情報の入手方法	• *OS/390 NFS User's Guide*, SC26-7254 および *OS/390 NFS Customization and Operation*, SC26-7253 を参照してください。

WebSphere for z/OS (デーモン) の自動化およびリカバリーのシナリオ

表 14. WebSphere for z/OS の自動化およびリカバリーのシナリオ

タスク	WebSphere for z/OS (デーモン) の自動化およびリカバリーのシナリオ
始動	『WebSphere Application Server ランタイム環境の始動』を参照してください。
シャットダウン	『WebSphere Application Server ランタイム環境のシャットダウン』を参照してください。
障害が発生した場合の未コミットまたは未確定のトランザクションの処理	デーモンは、ロケーション・エージェントとして動作します。トランザクションの途中でデーモンに障害が発生すると、デーモンに対する位置指定要求は失敗します。このような要求失敗は、クライアント ORB によって表面化されます。シスプレックス内で実行されている WebSphere for z/OS クライアントの場合、同じシスプレックス内に使用可能な別のデーモンがあるときは、このデーモンに位置指定要求が転送されます。
Application Server が稼働中かどうかの判断方法	MVS display コマンドを使用します。
Application Server がダウンした場合にその他のサブシステムで発生する状況	その他のサブシステムは、問題なく稼働し続けます。 Application Server デーモンがダウンすると、終了したデーモンと同じシステムで始動されたすべての Application Server サーバーが終了します。一般に、デーモンがダウンした場合、シスプレックス内に別のデーモンがあるときは、クライアントに影響はありません。
詳細情報の入手方法	• xiiページの『関連情報の入手先』に記載されている WebSphere for z/OS 資料を参照してください。

ネーミングの自動化およびリカバリーのシナリオ

表 15. ネーミングの自動化およびリカバリーのシナリオ

タスク	ネーミングの自動化およびリカバリーのシナリオ
始動	(*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 で推奨されたとおりのセットアップを使用している場合) Application Server が始動されると、ネーミング制御領域が自動的かつ暗黙的に始動されます。(実際の作業を実行する) ネーミング・サーバー領域は、作業を実行する必要が生じたときにサーバーからの要求に従って始動されます。

表 15. ネーミングの自動化およびリカバリーのシナリオ (続き)

タスク	ネーミングの自動化およびリカバリーのシナリオ
シャットダウン	Application Server を停止すると、ネーミング制御領域および稼働中のすべてのサーバー領域が自動的に停止されます。
障害が発生した場合の未コミットまたは未確定のトランザクションの処理	処理中に障害が発生した場合、ネーミングは、RRS を介して、LDAP および DB2 に対してロールバックを直接発行します。この結果、この要求に従って各作業がロールバックされます。
ネーミングが稼働中かどうかの判断方法	Application Server が稼働中であれば、ネーミングも稼働しています。ネーミングがダウンすると、WebSphere for z/OS は使用不能になります。SDSF を使用してネーミング・タスクをモニターすることもできます。
Application Server がダウンした場合にその他のサブシステムで発生する状況	ネーミング・サーバー領域で障害が発生した場合は、WLM によってリカバリーが実行されます。WLM は、単に新しいサーバー領域を始動します。これにより、他のサブシステムが影響を受けることはありません。ネーミング制御領域がダウンしたにもかかわらず、ARM によって再始動されない場合、WebSphere for z/OS は使用不能になります。
詳細情報の入手方法	• xiiページの『関連情報の入手先』に記載されている WebSphere for z/OS 資料を参照してください。
注: シスプレックスでは、ネーミング・サーバーが 1 つだけ必要です。したがって、1 つのシステムでネーミングがダウンした場合でも、同じシスプレックス内で実行されているネーミングが 1 つあれば、実行を続けることができます。	

インターフェース・リポジトリーの自動化およびリカバリーのシナリオ

表 16. インターフェース・リポジトリー (IR) の自動化およびリカバリーのシナリオ

タスク	インターフェース・リポジトリーの自動化およびリカバリーのシナリオ
始動	(*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 で推奨されたとおりの設定を使用している場合) Application Server が始動されると、インターフェース・リポジトリー制御領域が自動的かつ暗黙的に始動されます。(実際の作業を実行する) インターフェース・リポジトリー・サーバー領域は、作業を実行する必要が生じたときにサーバーからの要求に従って始動されます。
シャットダウン	Application Server を停止すると、インターフェース・リポジトリー制御領域および稼働中のすべてのサーバー領域が自動的に停止されます。

表 16. インターフェース・リポジトリー (IR) の自動化およびリカバリーのシナリオ (続き)

タスク	インターフェース・リポジトリーの自動化およびリカバリーのシナリオ
障害が発生した場合の未コミットまたは未確定のトランザクションの処理	処理中に障害が発生した場合、インターフェース・リポジトリーは、RRS を介して、LDAP および DB2 に対してロールバックを直接発行します。この結果、このロールバック要求に基づいて作業がロールバックされます。
インターフェース・リポジトリーが稼働中かどうかの判別方法	インターフェース・リポジトリー・サーバーは、SDSF を使用してモニターすることができます。
Application Server がダウンした場合にその他のサブシステムで発生する状況	インターフェース・リポジトリー・サーバー領域で障害が発生した場合は、WLM によってリカバリーが実行されます。WLM は、単に新しいサーバー領域を始動します。これにより、他のサブシステムが影響を受けることはありません。
詳細情報の入手方法	• xiiページの『関連情報の入手先』に記載されている WebSphere for z/OS 資料を参照してください。
注: シスプレックスでは、インターフェース・リポジトリー・サーバーが 1 つだけ必要です。したがって、1 つのシステムでインターフェース・リポジトリーがダウンした場合でも、同じシスプレックス内で実行されているインターフェース・リポジトリーが 1 つあれば、実行を続けることができます。	

システム管理 (SM) の自動化およびリカバリーのシナリオ

表 17. システム管理 (SM) の自動化およびリカバリーのシナリオ

タスク	システム管理 (SM) の自動化およびリカバリーのシナリオ
始動	システム管理は、WebSphere for z/OS サーバーの一種であり、Application Server デーモンの始動時に自動的に始動されます。システム管理は、WebSphere for z/OS サーバーとして、Application Server インフラストラクチャー (DB2、 RRS、 OMVS、 LDAP、 WLM など) の前提条件となります。
シャットダウン	システム管理は、デーモンがシャットダウンされると、自動的にシャットダウンされます。

表 17. システム管理 (SM) の自動化およびリカバリーのシナリオ (続き)

タスク	システム管理 (SM) の自動化およびリカバリーのシナリオ
障害が発生した場合の未コミットまたは未確定のトランザクションの処理	システム管理では、ORB および OTS を使用してトランザクションを処理します。要求がシステム管理に送られると、トランザクションが暗黙的に開始されます (ORB/OTS によって処理されます)。問題が発生したときは、OTS から明示的にロールバックする必要がある場合と、ORB によって自動的にロールバックされる場合があります。ORB と OTS は、RRS を介してコミットおよびロールバックを管理します。
システム管理が稼働中かどうかの判別	システム管理制御領域が稼働中かどうかを確認してください。これには、SDSF を使用します。
Application Server がダウンした場合にその他のサブシステムで発生する状況	システム管理で障害が発生すると、Application Server は動作できなくなります。その他のサブシステムには影響はありません。
詳細情報の入手方法	• xiiページの『関連情報の入手先』に記載されている WebSphere for z/OS 資料 (特に次の資料) を参照してください。 – *WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 – *WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656

WebServer の自動化およびリカバリーのシナリオ

表 18. WebServer (サーブレット) の自動化およびリカバリーのシナリオ

タスク	WebServer の自動化およびリカバリーのシナリオ
始動	WebSphere for z/OS 製品および WebSphere for z/OS インフラストラクチャーでは、WebServer も Application Server も必須ではありません。WebServer および Application Server スタンダード版ランタイムでは、Application Server と関係している点は、WebSphere for z/OS 機能を使用することを指定して記述されたクライアント・アプリケーション・プログラムがサーブレットとして記述されるということだけです。始動順序の暗黙的指定はすべて、アプリケーションによって行われます。場合によっては、Application Server オブジェクト・サーバーを始動して作動可能にしてから、クライアント・アプリケーションのホストとして機能している Web サーバーを始動する必要があります。

表 18. WebServer (サーブレット) の自動化およびリカバリーのシナリオ (続き)

タスク	WebServer の自動化およびリカバリーのシナリオ
シャットダウン	したがって、製品コードからの依存関係はありません。多くのアプリケーションと同様、必要な場合は、クライアントを静止してから、ターゲットの WebSphere for z/OS サーバーを停止してください。
障害が発生した場合の未コミットまたは未確定のトランザクションの処理	これは、WebSphere for z/OS クライアントの ORB 機能に関する記述です。障害が発生した場合に、クライアントを再始動しなければならないという要件はありません。OTS では、接続のタイムアウトと切断、および異常終了の想定を通して、この処理を行います。
WebServer が稼働中かどうかの判断方法	display コマンド、SMF レコード、およびビューアー・ツール (SDDF) を使用して、Application Server をモニターしてください。
WebServer がダウンした場合に Application Server で発生する状況	ありません。アプリケーションで調整されます。
WebServer がダウンした場合にその他のサブシステムで発生する状況	ありません。
詳細情報の入手方法	• *Application Server 計画、インストールおよび使用の手引き*, GD88-7895 を参照してください。

第7章 WebSphere for z/OS の管理手順

この章では、WebSphere for z/OS の管理作業およびガイドラインについて説明します。

詳細情報:

- RACF および DCE のシステム・セキュリティーおよびユーザー ID の設定については、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。
- DNS 定義の更新、TCP/IP ネットワークの設定、およびシスプレックスの拡張に伴うホスト・ファイルの更新については、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。
- ユーザー ID の設定、並列シスプレックス環境でのアプリケーションの展開、およびシスプレックス規模での Application Server アプリケーションのインストールについては、*WebSphere Application Server V4.0 for z/OS and OS/390: J2EE アプリケーションのアセンブル*, SA88-8654 を参照してください。

SSL セキュリティーの管理

WebSphere for z/OS の SSL セキュリティーのセットアップ

ここでは、ユーザーが、SSL プロトコルおよび OS/390 上で Cryptographic Services System SSL がどのように機能するかを理解していることを前提としています。SSL プロトコルについては、以下の Web サイトを参照してください。

```
http://home.netscape.com/eng/ssl3/ssl-toc.html
```

Cryptographic Services System SSL についての詳細は、*z/OS System Secure Sockets Layer Programming*, SC24-5901 を参照してください。

保護が必要な通信への追加セキュリティーおよびネットワーク内でのユーザー認証が必要な場合は、Secure Sockets Layer (SSL) セキュリティーを使用することができます。WebSphere for z/OS の SSL サポートにはいくつかの目的があります。

- メッセージがネットワークを介して伝送されるときにメッセージのセキュリティーを保護するため、業界によって受諾された方法を提供する。これは通常トランスポート層セキュリティー と呼ばれます。トランスポート層セキュリティーは、2 つの通信アプリケーション間でのプライバシーとデータ保全性を提供する機能です。保護は基本トランスポート・プロトコルの上位のソフトウェア層 (TCP/IP の上位など) で行われます。

  SSL は暗号化テクノロジーを使用して通信リンクにおけるセキュリティーを提供し、ネットワークでのメッセージの保全性を確保します。通信を行う両者間で通信が暗号化されているため、第三者がメッセージの内容を変更することはできません。また、SSL は機密性 (メッセージの内容が読み取られないようにする)、再生検出、および順不同検出機能も提供します。

- 保護通信メディアを提供する。このメディアを介して各種認証プロトコルが動作します。単一の SSL セッションで複数の認証プロトコル、つまり通信者の ID を証明する方式を採用できます。

  SSL サポートは、サーバーが自身の ID を証明するメカニズムを常に提供します。WebSphere for z/OS の SSL サポートによってクライアントが ID を証明する方法は次の 3 つです。

  – 基本認証 (SSL タイプ 1 認証とも呼ばれる)。基本認証では、クライアントはターゲット・サーバーによって認識されているユーザー ID とパスワードを渡すことによってサーバーに ID を証明します。

    SSL 基本認証を使用すると、以下のことが可能です。

    - OS/390 または z/OS クライアントは、ユーザー ID とパスワードを使用することによって WebSphere for z/OS サーバーと安全に通信できる。
    - OS/390 または z/OS クライアントは、DCE プリンシパルとパスワードを使用することによって WebSphere 分散プラットフォーム上のサーバーと安全に通信できる。
    - 分散プラットフォーム・クライアントは、MVS ユーザー ID とパスワードを使用することによって WebSphere for z/OS サーバーと安全に通信できる。
    - パスワードは要求時に必ず必要になるため、単純なクライアント / サーバー間接続のみを確立できます。つまり、サーバーは要求に応答するためにクライアントのユーザー ID を別のサーバーに送信することはできません。この機能は*識別表明* またはトラステッド・アソシエーション と呼ばれます。詳細は後で説明します。

- クライアント認証サポート。クライアント認証では、サーバーとクライアントの両方が ID を証明するためのディジタル証明書を相互に提供します。

  Web アプリケーションでは何千ものクライアントが存在するため、クライアント認証の管理が管理上の大きな負担になっています。RACF *認証名フィルター操作* を使用して、WebSphere for z/OS の SSL サポートではクライアント証明書を保管せずに MVS ユーザー ID にマップできるようにします。認証名フィルター操作を使用すると、MVS ユーザー ID を作成してユーザーごとにクライアント証明書を管理するという管理オーバーヘッドなしに、一連のユーザーにサーバーへのアクセスを許可することができます。

- Kerberos セキュリティー。Kerberos セキュリティーでは、サーバーはクライアントに対してディジタル証明書を渡して、その ID を証明します。クライアントは Kerberos 認証を使用してサーバーに ID を証明します。

- アイデンティティー表明 (トラステッド・アソシエーション)。中間サーバーはそのクライアントの ID をターゲット・サーバーに安全でしかも効果的な方法で送信できます。このサポートでは、SSL セッションの所有者として中間サーバーを確立するためにクライアント証明書を使用します。RACF を使用することで、システムは中間サーバーの信頼性を検査することができます (特殊な SAF 許可が制御領域などのセキュア・システム・コードを実行するアドレス・スペースに与えられます)。この中間サーバーの信頼性が確立されると、クライアント・アイデンティティー (MVS ユーザー ID) をターゲット・サーバーによって個別に検査する必要はなくなります。これらのクライアント・アイデンティティーは単に表明されるだけで、認証の必要はありません。

- その他の製品 (以下を参照) と安全な方法で相互操作する方法
  - CICS トランザクション・サーバー (z/OS 版)
  - 分散プラットフォーム上の WebSphere
  - CORBA 準拠のオブジェクト・リクエスト・ブローカー

SSL サポートはオプションです。SSL を使用せずに WebSphere for z/OS を実行すると、通信を保護してクライアントとサーバーを認証する SSL 機能にのみ影響します。

以下で、SSL 接続の働きを説明します。

ステージ	説明
ネゴシエーション	クライアントがサーバーを見つけると、クライアントとサーバーは、通信のセキュリティー・タイプをネゴシエーションします。SSL を使用する場合、クライアントは特別の SSL ポートに接続するように指示されます。
ハンドシェーク	クライアントが SSL ポートに接続すると、SSL ハンドシェークが行われます。正常に実行されると、暗号化された通信が開始されます。クライアントは、サーバーのディジタル証明書を検査して、サーバーを認証します。 ハンドシェーク中にクライアント証明書が使用される場合、サーバーはクライアントのディジタル証明書を検査して、クライアントを認証します。
基本認証を使用する場合	SSL ハンドシェークが行われたあと、クライアントは SSL 暗号化パイプを介して ユーザー・アイデンティティーとパスワードを提供し、サーバーに対してクライアント・アイデンティティーを確立します。サーバーが OS/390 上にある場合、クライアントはユーザー ID とパスワードを提供します。サーバーがワークステーション上にある場合、クライアントは DCE プリンシパルとパスワードを提供します。
最初のクライアント要求	サーバーが最初のクライアント要求を受信すると、サーバーと RACF はそのクライアント証明書用に OS/390 ユーザー・アイデンティティーを確立して、そのクライアント・アイデンティティーのもとで要求を実行します。 RACF がユーザー ID を認証すると、サーバーは、クライアント・アイデンティティーに基づいて作業要求を実行します。クライアント認証が失敗すると、通信は停止します。
通信の進行中	SSL ハンドシェーク中に、クライアントとサーバーは暗号化通信で使用する暗号のスペックをネゴシエーションします。

規則:

- サーバーの制御領域と OS/390 クライアントのみが、Cryptographic Services System SSL へのアクセスを必要とします。ユーザー制御領域と OS/390 クライアントは、*hlq*.SGSKLOAD データ・セットへのアクセスを必要とします。SGSKLOAD を LPA の中に置いてください。詳細は、*z/OS System Secure Sockets Layer Programming*, SC24-5901 を参照してください。

- OS/390 上の Java または C++ クライアントは、WebSphere for z/OS またはワークステーション・サーバーと相互協調処理して SSL を使用することができます。
- ハンドシェーク中には、SSL がメッセージ保護に使用する暗号のスペックがネゴシエーションされます。システムにインストールされている Cryptographic Services のセキュリティー・レベルによって、WebSphere for z/OS で使用できる暗号のスペックおよびキー・サイズが決まります (詳細は、*z/OS System Secure Sockets Layer Programming*, SC24-5901 を参照してください)。
- ディジタル証明書およびキーの保管には、RACF またはそれと同等の機能を使用しなければなりません。HFS のキー・データベースにディジタル証明書およびキーを置くことはできません。
- デーモン・サーバーは SSL を使用しません。

アプリケーション・サーバーおよびクライアントでの SSL 基本認証セキュリティーの概説

SSL 基本認証セキュリティーを定義するには、最初にサーバーの署名証明書とサーバーの証明書に署名した認証局 (CA) からの認証局証明書を要求する必要があります。証明書を要求するプロセスについては、本書では扱いません。証明書の要求についての詳細は、*z/OS System Secure Sockets Layer Programming*, SC24-5901 を参照してください。

サーバーの署名証明書、および認証局からの CA (認証局) 証明書を受け取ったあとは、RACF を使用してディジタル証明書の使用を許可し、サーバーの証明書およびサーバーの鍵リングを RACF に保管して、管理アプリケーションを通じてサーバーの SSL セキュリティー特性を定義する必要があります。

クライアントについては、鍵リングを作成して、それにサーバーの証明書を発行した認証局からの CA (認証局) 証明書を添付します。OS/390 クライアントの場合は、RACF を使用してクライアント鍵リングを作成し、その鍵リングに CA (認証局) 証明書を添付します。

67ページの図1 は、SSL 基本認証における証明書の仕組みを示しています。

- **クライアントがサーバーを認証する場合は**、サーバー (実際には制御領域ユーザー ID) は認証局 (CA) によって作成された署名証明書を所有している必要があります。サーバーは署名証明書をクライアントに渡してアイデンティティーを証明します。クライアントはサーバーの証明書を発行した認証局と同じ認証局から発行された CA (認証局) 証明書を所有している必要があります。クライアントはその CA (認証局) 証明書を使用してサーバーの証明

書が正当であることを検査します。検査がすむと、クライアントはメッセージが他のサーバーではなく、間違いなくそのサーバーから送信されたものであることを信頼できます。

- **サーバーがクライアントを認証する場合は**、クライアントがサーバーに対してアイデンティティーを証明するために渡すクライアント証明書は存在しないので注意してください。SSL 基本認証方式では、サーバーはクライアントにユーザー ID とパスワードを要求することによってクライアントを認証します。

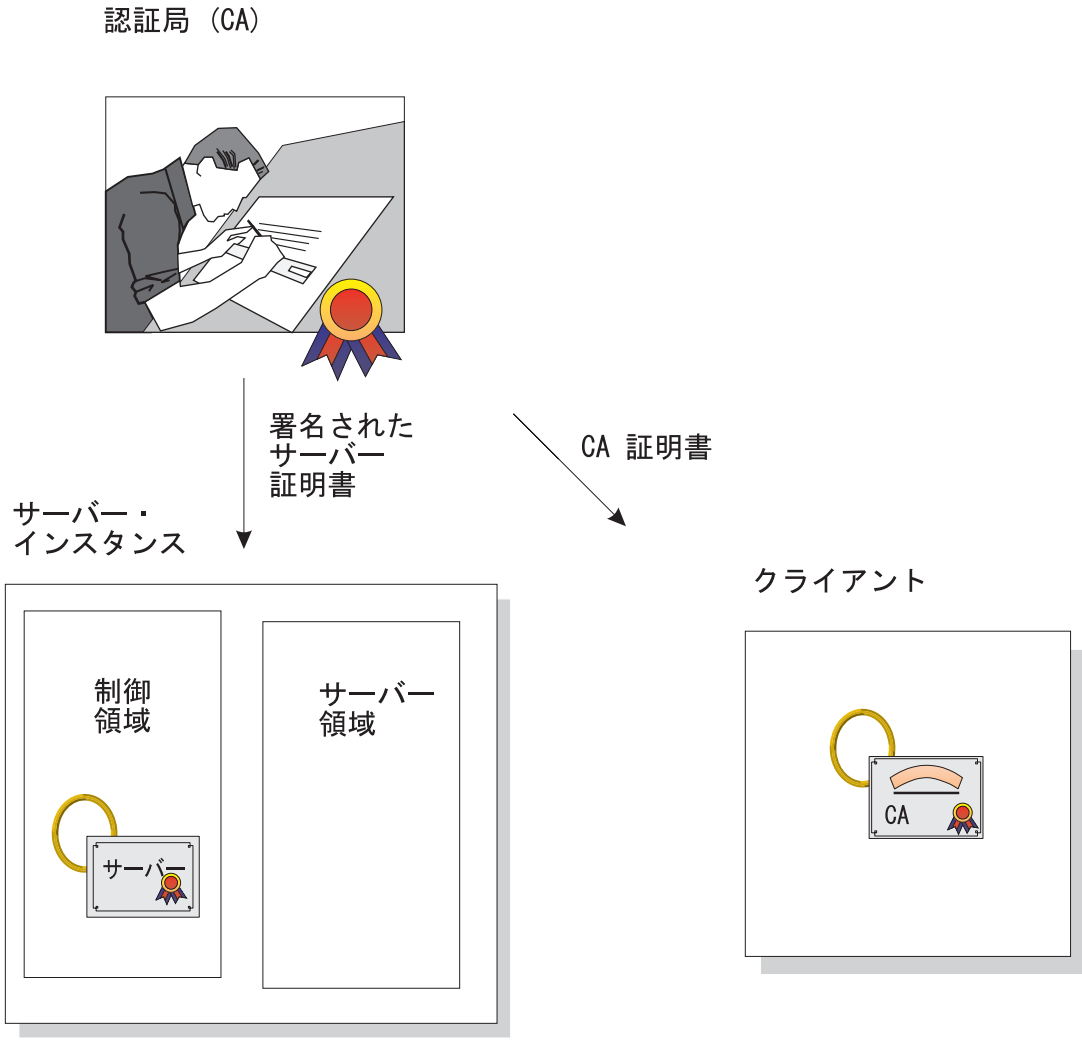

図 1. SSL 基本認証における証明書の仕組み

規則:

- OS/390 以外のプラットフォーム上にある Java クライアントの場合、WebSphere for z/OS サーバーと相互協調処理して SSL 基本認証を使用するために、そのプラットフォーム上に WebSphere Application Server エンタープライズ版 3.5 がなければなりません。他のプラットフォーム上にある C++ クライアントは、WebSphere for z/OS と相互協調処理する際に SSL 基本認証を使用できません。
- SSL 基本認証の場合、クライアントは次の方法で認証されます。

- リモート OS/390 サーバーと通信する OS/390 クライアントは、クライアントの環境ファイルにあるリモート・ユーザー ID およびパスワード (REM_USERID および REM_PASSWORD) 環境変数を使用して、クライアント・アイデンティティーを認証します。
- OS/390 クライアントが他のプラットフォームにある Component Broker サーバーで SSL を使用する場合、クライアントは、REM_DCEPRINCIPAL 環境変数および REM_DCEPASSWORD 環境変数を使用してサーバーに定義された DCE プリンシパルおよびパスワードを渡さなければなりません。
- OS/390 クライアントは、SSL_KEYRING 環境変数を使用して自身の鍵リングも示す必要があります。
- OS/390 サーバーと通信する WebSphere Application Server 分散プラットフォーム上のクライアントは、ユーザーがユーザー ID とパスワードを提供する ORB に用意されたユーザー・ダイアログを使用します。

以下の表には、SSL 基本認証セキュリティーの定義におけるサブタスクおよび関連手順を示します。

サブタスク	関連手順 (参照個所)
サーバー証明書および認証局 (CA) 証明書の要求	*z/OS System Secure Sockets Layer Programming*, SC24-5901
サーバーの SSL 基本認証セキュリティーのセットアップ	72ページの『RACF を使用してサーバーでのディジタル証明書の使用を許可するためのステップ』 74ページの『SSL セキュリティーのサーバー・セキュリティー特性を定義するステップ』
クライアントの SSL 基本認証セキュリティーのセットアップ	75ページの『クライアントでの SSL セキュリティーのセットアップ・ステップ』

アプリケーション・サーバーおよびクライアントでの SSL クライアント認証セキュリティーの概説

SSL クライアント認証セキュリティーを定義するには、最初にサーバーとクライアントの署名証明書と、それらの証明書に署名した認証局 (CA) からの認証局証明書を要求する必要があります。証明書を要求するプロセスについては、本書では扱いません。証明書の要求についての詳細は、*z/OS System Secure Sockets Layer Programming*, SC24-5901 を参照してください。

認証局から署名証明書および CA 証明書を受け取ったあとは、RACF を使用してディジタル証明書の使用を許可し、証明書および鍵リングを RACF に保管して、管理アプリケーションを通じてサーバーの SSL セキュリティー特性を定義する必要があります。

ディジタル証明書によって識別された各クライアントは、最後にターゲット WebSphere for z/OS サーバーによって MVS ユーザー ID に変換される必要があります。クライアントとサーバーが同じ RACF データベースを共用している場合は、このマッピングのために追加構成を行う必要はありません。クライアントとサーバーが同じ RACF データベースを共用していない場合は、次の方法でマッピングを構成できます。

- クライアント証明書をターゲット・サーバーの RACF データベースに追加する。この方法はあまり実用的ではありません。
- RACF 認証名フィルター操作を使用して、クライアント・グループを RACF アイデンティティーにマッピングする。
- この 2 つを組み合わせて使用する。

71ページの図2 は、SSL クライアント証明書認証における証明書の仕組みを示しています。

- **クライアントがサーバーを認証する場合は**、サーバー (実際には制御領域ユーザー ID) は認証局 (CA) によって作成された署名証明書を所有している必要があります。サーバーは署名証明書をクライアントに渡してアイデンティティーを証明します。クライアントはサーバーの証明書を発行した認証局と同じ認証局から発行された CA (認証局) 証明書を所有している必要があります。クライアントはその CA (認証局) 証明書を使用してサーバーの証明書が正当であることを検査します。検査がすむと、クライアントはメッセージが他のサーバーではなく、間違いなくそのサーバーから送信されたものであると信頼できます。
- **サーバーがクライアントを認証する場合は**、クライアントは認証局 (CA2) によって作成された署名証明書を所有している必要があります。(71ページの図2 では、明確にするために 2 つの異なる認証局を示しています。同じ認証

局がサーバーとクライアントの両方に署名証明書を提供する場合もあります。) サーバーはクライアントの証明書を発行した認証局と同じ認証局から発行された CA2 証明書を所有している必要があります。サーバーはその CA2 証明書を使用してクライアントの証明書が正当であることを検査します。検査が済むと、サーバーはメッセージが他のクライアントではなく、間違いなくそのクライアントから送信されたものであると信頼できます。

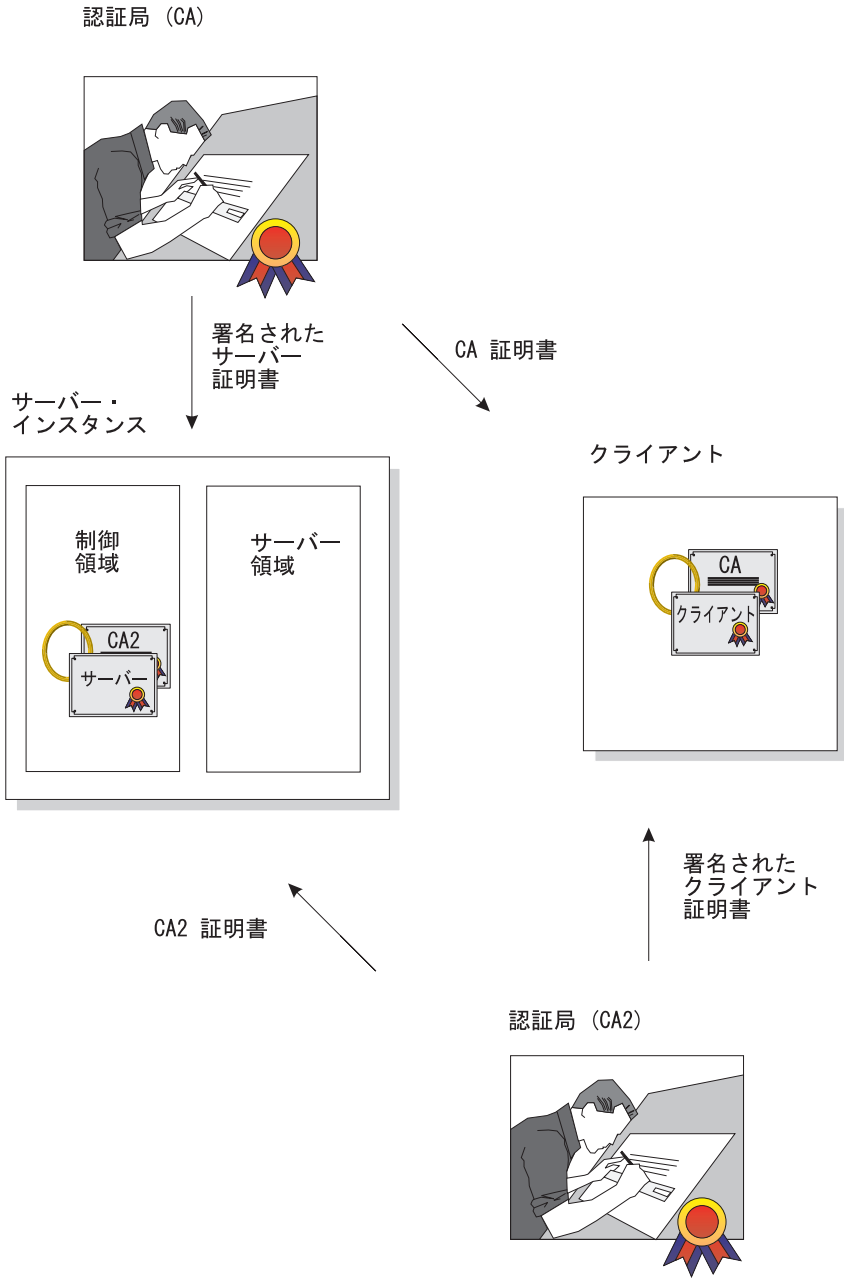

図 2. SSL クライアント認証セキュリティーにおける証明書の仕組み

以下の表には、SSL クライアント認証セキュリティーの定義におけるサブタスクおよび関連手順を示します。

サブタスク	関連手順 (参照個所)
サーバー証明書および認証局 (CA) 証明書の要求	*z/OS System Secure Sockets Layer Programming*, SC24-5901
サーバーの SSL クライアント認証セキュリティーのセットアップ	『RACF を使用してサーバーでのディジタル証明書の使用を許可するためのステップ』 74ページの『SSL セキュリティーのサーバー・セキュリティー特性を定義するステップ』
クライアントの SSL クライアント認証セキュリティーのセットアップ	75ページの『クライアントでの SSL セキュリティーのセットアップ・ステップ』
ユーザーのサーバー・システム上でのクライアント・ディジタル証明書の MVS ユーザー ID へのマッピング	77ページの『ユーザーのサーバー・システム上でクライアント・ディジタル証明書を MVS ユーザー ID にマッピングするためのステップ』

クライアントおよびサーバーでの SSL セキュリティーの定義

この節には、SSL ベースのすべての認証メカニズムをインプリメントするために従う必要のある手順が記載されています。

RACF を使用してサーバーでのディジタル証明書の使用を許可するためのステップ: SSL は、ディジタル証明書および公開鍵 / 秘密鍵を使用します。アプリケーション・サーバーで SSL を使用する場合、RACF を使用してサーバーの制御領域を実行する際のユーザー・アイデンティティーに対してディジタル証明書および公開鍵 / 秘密鍵を保管するする必要があります。

始める前に: 認証局 (CA) 証明書およびサーバーの署名証明書を要求する必要があります。

SSL クライアント認証サポートをインプリメントする場合は、クライアント証明書を検査する各認証局からの認証局 (CA) 証明書も必要になります。*z/OS System Secure Sockets Layer Programming*, SC24-5901 を参照してください。

ユーザーは、RACF で RACDCERT コマンドを使用する権限 (たとえば、SPECIAL 権限) を持つユーザー ID を持っていなければなりません。RACDCERT についての詳細は、*z/OS SecureWay Security Server (RACF) コマンド言語 解説書*, SA88-8617、および *z/OS SecureWay Security Server (RACF) セキュリティー管理者のガイド*, SA88-8613。

以下のステップを実行して、ディジタル証明書の使用を許可してください。

1. SSL を使用する各サーバーごとに、そのサーバーの制御領域ユーザー ID に対する鍵リングを作成します。

   例: ユーザーの制御領域は、CBACRU1 と呼ばれるユーザー ID に関連付けられています。以下を実行します。

   ```
   RACDCERT ADDRING(ACRRING) ID(CBACRU1)
   ```

---

2. 認証局からアプリケーション・サーバーの証明書を受け取ります。

   例: 証明書を要求すると、認証局は、ユーザーが CBACRU1.CA と呼ばれるファイルに保管した署名証明書をユーザーに戻します。以下を実行します。

   ```
   RACDCERT ID (CBACRU1) ADD('CBACRU1.CA') WITHLABEL('ACRCERT') PASSWORD('password')
   ```

---

3. 署名証明書を制御領域ユーザー ID の鍵リングに接続し、この証明書をデフォルトの証明書とします。

   例: ACRCERT というラベルの付いた証明書を、CBACRU1 が所有する ACRRING という鍵リングに接続します。以下を実行します。

   ```
   RACDCERT ID(CBACRU1) CONNECT (ID(CBACRU1) LABEL('ACRCERT') RING(ACRRING) DEFAULT)
   ```

---

4. サーバー認証クライアントが必要な場合 (SSL クライアント認証サポート)
   - ユーザーのクライアント証明書を検査する各認証局 (CA) 証明書を受け取る。各 CA 証明書に CERTAUTH 属性を付与する。

     例: ユーザー ID CLIENT1 によってクライアントを検査する CA (認証局) 証明書を受け取る。この証明書は USER.CLIENT1.CA というファイルにあります。以下を実行します。

     ```
     RACDCERT ADD('USER.CLIENT1.CA') WITHLABEL('CLIENT1 CA') CERTAUTH
     ```

   - 各クライアントの認証局 (CA) 証明書を制御領域ユーザー ID の鍵リングに接続する。

     例: CLIENT1 CA 証明書を、CBACRU1 が所有する ACRRING というリングに接続します。

     ```
     RACDCERT ID(CBACRU1) CONNECT(CERTAUTH LABEL('CLIENT1 CA') RING(ACRRING))
     ```

---

5. RACF FACILITY クラスの IRR.DIGTCERT.LIST および IRR.DIGTCERT.LISTRING に、制御領域ユーザー ID への読み取りアクセス権限を与えます。

例: ユーザーの制御領域ユーザー ID は CBACRU1 です。以下を実行します。

```
PERMIT IRR.DIGTCERT.LIST     CLASS(FACILITY) ID(CBACRU1) ACC(READ)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(CBACRU1) ACC(READ)
```

RACF コマンドが正常に終了すれば、RACF フェーズは完了です。『SSL セキュリティーのサーバー・セキュリティー特性を定義するステップ』に進んでください。

SSL セキュリティーのサーバー・セキュリティー特性を定義するステップ:
この手順では、サーバーが、管理アプリケーションを通して SSL クライアント証明書セキュリティーを使用するように指定する方法を説明します。

始める前に: 管理アプリケーションを開始し、ログオンして、新規の会話を作成する必要があります。詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照してください。

以下のステップを実行して、サーバーのセキュリティー特性を定義してください。

1. 「会話 (Conversations)」ツリーで「サーバー (Servers)」を展開します。

---

2. 新規サーバーを作成するか、既存のサーバーの名前をクリックします。

---

3. 特性フォームで以下のことを行います。
   - SSL 基本認証をインプリメントしている場合は、[SSL タイプ 1 (基本認証) (SSL Type 1 (basic authentication))] チェック・ボックスをクリックする。
   - SSL クライアント認証をインプリメントしている場合は、[SSL クライアント証明書 (SSL Client Certificates)] チェック・ボックスをクリックする。
   - Kerberos をインプリメントしている場合は、[Kerberos] チェック・ボックスをクリックする。
   - 表明アイデンティティーをインプリメントしている場合は、[表明アイデンティティー (Asserted identity)] チェック・ボックスをクリックする。[SSL クライアント証明書 (SSL client certificates)] チェック・ボックスもクリックしてください。

4. SSL RACF 鍵リングを指定します。これは、72ページの『RACF を使用してサーバーでのディジタル証明書の使用を許可するためのステップ』 のステップ 1 で定義した鍵リングです。

   注: 間違った RACF 鍵リングを指定すると、サーバー実行時にエラー・メッセージが出されます。

5. システムがセッション・キーを保持する時間の長さである、SSL V2 タイムアウト値を秒単位で指定します。範囲は 0 ～ 100 秒です。デフォルトは 100 秒です。

6. システムがセッション・キーを保持する時間の長さである、SSL V3 タイムアウト値を秒単位で指定します。範囲は 0 ～ 86400 (1 日) です。デフォルトは 600 秒です。

7. セキュリティーの優先リストを順序付けます。セキュリティーの優先リストについての詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

8. サーバーに関するそのほかのすべての指定を完了した後、妥当性検査を行ってコミットし、すべてのタスクを完了後、会話を活動化します。

会話が活動化されたことをシステムから通知されると完了です。

クライアントでの SSL セキュリティーのセットアップ・ステップ: すべてのクライアントは、SSL ハンドシェーク中にサーバーを認証できるように、サーバーの認証局 (CA) 証明書へのアクセス権限を持っている必要があります。SSL クライアント認証サポートをインプリメントする場合は、クライアントはさらに自身の鍵リングに対するデフォルト証明書として独自の証明書も持つ必要があります。

- クライアントがワークステーション上の WebSphere から WebSphere for z/OS に接続している場合、ワークステーション・システムに SSL 証明書をインポートする必要があります。詳細な説明については、*WebSphere Application Server エンタープライズ版 Component Broker システム管理の手引き*, SD88-7375 を参照してください。

- OS/390 では、クライアントは RACF で鍵リングに付加された証明書を持っている必要があります。

この手順では、証明書を OS/390 クライアントに付加する方法について説明します。

始める前に: SSL 基本認証の場合、CA (認証局) 証明書は、アプリケーション・サーバーの署名証明書を発行した認証局と同じ認証局に要求しなければなりません。SSL クライアント認証サポートをインプリメントする場合は、さらに認証局から発行されるクライアントの署名証明書も要求する必要があります。

ユーザーは、RACF で RACDCERT コマンドを使用する権限 (たとえば、SPECIAL 権限) を持つユーザー ID を持っていなければなりません。RACDCERT についての詳細は、*z/OS SecureWay Security Server (RACF) コマンド言語 解説書*, SA88-8617、および *z/OS SecureWay Security Server (RACF) セキュリティー管理者のガイド*, SA88-8613 を参照してください。

以下のステップを実行して、OS/390 クライアントによるディジタル証明書の使用を許可してください。

1. OS/390 クライアントの鍵リングを作成します。

   例: ユーザーのクライアント・ユーザー ID は CLIENT1 です。以下を実行します。

   ```
   RACDCERT ADDRING(C1RING) ID(CLIENT1)
   ```

   ---

2. サーバーの認証局 (CA) 証明書を受け取り、それに CERTAUTH 属性を付与します。

   例: CA (認証局) 証明書を要求すると、認証局は、ユーザーが USER.CBSERVER.CA と呼ばれるファイルに保管したその証明書をユーザーに戻します。以下のコマンドを実行します。

   ```
   RACDCERT ADD('USER.CBSERVER.CA') WITHLABEL('VERI CA') CERTAUTH
   ```

   ---

3. サーバーの CA 証明書をクライアント鍵リングに接続します。

   例: VERI CA (認証局) 証明書を CLIENT1 が所有する C1RING 鍵リングに接続します。

   ```
   RACDCERT ID(CLIENT1) CONNECT(CERTAUTH LABEL('VERI CA') RING(C1RING))
   ```

   ---

4. クライアントの環境ファイルに、クライアントの鍵リングに一致するように SSL_KEYRING 環境変数をコーディングします。

   環境変数についての詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

---

5. SSL クライアント認証サポートをインプリメントしている場合、次の操作を行います。
   - 認証局からクライアントの証明書を受け取る。

     例: 証明書を要求すると、認証局はユーザーが CLIENT1.SIGNED.CERT に保管した署名証明書を戻します。以下を実行します。

     ```
     RACDCERT ID (CLIENT1) ADD('CLIENT1.SIGNED.CERT') WITHLABEL('CLIENT1 CERT') PASSWORD('password')
     ```

   - クライアントの署名証明書をクライアント・ユーザー ID の鍵リングに接続し、この証明書をデフォルトの証明書とする。

     例: CLIENT1 というラベルの付いた証明書を、CLIENT1 が所有する C1RING という鍵リングに接続します。以下を実行します。

     ```
     RACDCERT ID(CLIENT1) CONNECT (ID(CLIENT1) LABEL('CLIENT1 CERT') RING(C1RING) DEFAULT)
     ```

RACF コマンドが正常に終了すると完了です。環境ファイルを保管してください。

ユーザーのサーバー・システム上でクライアント・ディジタル証明書を MVS ユーザー ID にマッピングするためのステップ: アイデンティティーを証明するためにディジタル証明書を提示した各 Component Broker クライアントは、ターゲット・サーバーのシステムまたはシスプレックス上の RACF に登録された個々の証明書を持っていない場合、有効な MVS ユーザー ID へのマッピングを持っている必要があります。このマッピングを作成するには、RACF 認証名フィルターを使用します。

RACF 認証名フィルターは、クライアントまたは証明書発行元の識別名に基づいて X.509 ディジタル証明書に含まれているとおりに作成できます。

始める前に: ディジタル証明書を提示する一連のクライアントをどのように編成するか、およびどのようなアクセスをそれらのクライアントが必要としているかを明確にする必要があります。

RACDCERT MAP コマンドを発行する権限を持つ必要があります。

以下のステップを実行して認証名フィルターをセットアップします。

1. 認証名フィルターに関連付けるそれぞれのユーザー ID ごとに MVS ユーザー ID を定義します。各ユーザー ID に PROTECTED 属性と RESTRICTED 属性を割り当てることを検討してください。PROTECTED 属性は、ユーザー ID がシステムへの直接的なログオンに使用されること、および誤ったパスワード入力によって取り消されることを防ぎます。RESTRICTED 属性は、ユーザー ID が明示的にアクセスを許可されていない保護リソースへのアクセスに使用されないようにします。**例:**

   ```
   ALTUSER WEBUSER NOPASSWORD RESTRICTED
   ```

   ---

2. 認証名フィルターを活動化します。**例:**

   ```
   SETROPTS CLASSACT(DIGTNMAP) RACLIST(DIGTNMAP)
   ```

   ---

3. 認証名フィルターを作成します。**例:** 以下のフィルターは、VeriSign Class 1 によって発行された証明書を提示する (システム上の RACF に登録された個々の証明書を持たない) 任意のユーザーに、ユーザー ID WEBUSER を 関連付けます。

   ```
   RACDCERT ID(WEBUSER) MAP WITHLABEL('INTERNET OTHERS') +
            IDNFILTER('OU=VeriSign Class 1 Individual Subscriber.O=VeriSign, Inc.L=Internet')
   ```

   このフィルターは発行元の名前に基づきます。サブジェクトの名前、または発行元とサブジェクトの名前を組み合わせた名前に基づいてその他のフィルターを作成できます。認証名フィルターの詳細については、*z/OS SecureWay Security Server (RACF) セキュリティー管理者のガイド*, SA88-8613 を参照してください。

   ---

4. DIGTNMAP クラスを最新表示します。**例:**

   ```
   SETROPTS RACLIST(DIGTNMAP) REFRESH
   ```

SETROPTS コマンドが終了すると完了です。

WebSphere for z/OS に対する Kerberos セキュリティーのセットアップ

WebSphere for z/OS では、Kerberos が SSL と共に機能することによって、完全な認証メカニズムが提供されます。

- SSL では、トランスポート層の保護によりメッセージの安全性が確保されます。また SSL では、クライアントによるサーバーの認証メカニズムも提供されます。
- Kerberos では、サーバーによるクライアントの認証メカニズムが提供されます。このメカニズムでは、サーバーは、クライアントから送られた Kerberos Generic Security Service Application Program Interface (GSS_API) トークンを使用してそのクライアントのアイデンティティーを認証します。
- サーバーはクライアントの要求を満たすために、GSS_API トークンを介して、そのクライアントのアイデンティティーを別のサーバーに渡すことができます。これを代行と言います。

以下に、SSL を介した Kerberos 接続がどのように動作するかを説明します。

ステージ	説明
ネゴシエーション	クライアントがサーバーを見つけると、クライアントとサーバーは、通信のセキュリティー・タイプをネゴシエーションします。Kerberos を使用する場合、クライアントは特別な SSL ポートに接続するように指示されます。
ハンドシェーク	クライアントが SSL ポートに接続すると、SSL ハンドシェークが行われます。ハンドシェークが正常に行われると、SSL メッセージの保護が開始されます。クライアントは、サーバーのディジタル証明書を検査して、サーバーを認証します。

ステージ	説明
クライアント認証	SSL ハンドシェークが行われた後、クライアントはその Kerberos のアイデンティティーを確立し、この Kerberos アイデンティティーとサーバーの Kerberos プリンシパルに基づいて Kerberos GSS_API トークンを取得します。クライアントはこのトークンと一緒に固有の SSL 接続 ID をサーバーに送ります。サーバーは GSS_API トークンを使用してクライアントを表す Kerberos プリンシパルを認証します。 クライアントが認証されると、システムは RACF を使用して、クライアントの Kerberos プリンシパルにマップされた z/OS ユーザー ID を取得します。この z/OS ユーザー・アイデンティティーは以後の許可検査で使用されます。 デフォルトで、クライアントは代行を可能にする GSS_API トークンを構成します。それによって、サーバーは要求時にクライアントの代理として振る舞うことができます。 z/OS ユーザー ID、Kerberos 代行証明書、および固有の SSL 接続 ID は、この SSL Kerberos 接続に対する以後の要求で使用するために保管されます。 Kerberos によるクライアントの認証または認証されたプリンシパルのマッピングに失敗すると、通信は停止します。
通信の進行	クライアントとサーバー間の通信では、SSL サービスを使用してメッセージが保護されます。各メッセージには固有の SSL 接続 ID が格納されます。この ID によって、サーバーは保管されている z/OS ユーザー ID と Kerberos 代行証明書に要求を一致させることができます。

このサポートでは、SSL セキュリティーをセットアップする必要があります。また、SSL 要件のほかに、Kerberos では次のものを OS/390 システムにインストールおよび構成するがことが必要になります。

- OS/390 SecureWay Security Server OS/390 用ネットワーク認証およびプライバシー・サービス。OS/390 V2R8 および V2R9 用には、このサポートは次の Web サイトで提供されます。

  `http://www.software.ibm.com`

  OS/390 V2R10 および z/OS では、このサポートは SecureWay Security Server に含まれています。

- ご使用の OS/390 システムの PTF。詳しくは、PSP のバケットを参照してください。
- このサポートが使用されるクライアントおよびサーバー・システム上で、Kerberos セキュリティー・サーバーが活動していること。
- Kerberos 認証に組み込むすべての OS/390 ユーザー ID (クライアントおよびサーバーの) にその Kerberos プリンシパルを定義する Kerberos RACF セグメントが存在すること。
- Kerberos サーバーはその Kerberos 秘密鍵を格納するファイルを持っている必要はありません。OS/390 上の Kerberos では、この要件は除外され、現在のシステム・アイデンティティーに関連する Kerberos プリンシパルを使用してサービス・チケットの暗号化を解除することができます。WebSphere for z/OS サーバーはこの機能を使用する必要があります。
- WebSphere for z/OS サーバーが RACF FACILITY クラスの IRR.RUSERMAP リソースに対する READ アクセス権を持っていること。
- Kerberos セキュリティーではセキュリティー参加システム間の時間調整が必要となります。Kerberos セキュリティーの管理者は、時刻提供システムを選択し、Kerberos セキュリティー参加システムにその時刻ソースに基づいて各自のシステム時刻を維持させます。

次の表には、Kerberos セキュリティーの定義におけるサブタスクおよび関連手順を示します。

サブタスク	関連手順 (参照個所)
基本認証用の SSL のセットアップ	61ページの『WebSphere for z/OS の SSL セキュリティーのセットアップ』
Kerberos サーバーの使用可能化	*z/OS SecureWay Security Server Network Authentication Service Administration*, SC24-5926
Kerberos プリンシパルへのサーバー・アイデンティティーの関連付け	82ページの『Kerberos プリンシパルへのサーバー・アイデンティティーの関連付けのステップ』
サーバー属性に対する Kerberos の定義	82ページの『サーバーのセキュリティー属性に対する Kerberos の定義ステップ』
Kerberos 使用のためのクライアントのセットアップ	83ページの『Kerberos 使用のためのクライアントのセットアップ・ステップ』

Kerberos プリンシパルへのサーバー・アイデンティティーの関連付けのステップ

始める前に: サーバーの制御領域に対して RACF ユーザー ID を確立しておきます。

サーバー・アイデンティティーを Kerberos プリンシパルに関連付けるには、以下のステップを実行します。

⇔ ALTUSER コマンドを発行してアソシエーションを作成します。**例:**

```
ALTUSER ctl_ID PASSWORD(new_password) NOEXPIRED
        KERB(KERBNAME(kerberos_principal))
```

ここで、

ctl_ID
STARTED クラスを通してサーバーの制御領域に割り当てられたユーザー ID。

new_password
OS/390 または z/OS および Kerberos の共通パスワード。

kerberos_principal
この OS/390 または z/OS ユーザー ID に関連する Kerberos プリンシパル名。

RACF コマンドが正常に終了すると完了です。

サーバーのセキュリティー属性に対する Kerberos の定義ステップ

この手順では、サーバーでの Kerberos セキュリティーの使用を管理アプリケーションを使用してどのように指定するかについて説明します。

始める前に: 管理アプリケーションを開始してログオンし、新しい会話を作成する必要があります。詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照してください。

以下のステップを実行して、サーバーのセキュリティー特性を定義してください。

1. 「会話 (Conversations)」ツリーで「サーバー (Servers)」を展開します。

---

2. 新規サーバーを作成するか、既存のサーバーの名前をクリックします。

---

3. 特性フォームで、「Kerberos 許可 (Kerberos allowed)」チェック・ボックスをクリックします。

---

4. SSL RACF 鍵リングを指定します。これは、72ページの『RACF を使用してサーバーでのディジタル証明書の使用を許可するためのステップ』 のステップ 1 で定義した鍵リングです。

   注: 間違った RACF 鍵リングを指定すると、サーバー実行時にエラー・メッセージが出されます。

---

5. システムがセッション・キーを保持する時間の長さである、SSL V2 タイムアウト値を秒単位で指定します。範囲は 0 ～ 100 秒です。デフォルトは 100 秒です。

---

6. システムがセッション・キーを保持する時間の長さである、SSL V3 タイムアウト値を秒単位で指定します。範囲は 0 ～ 86400 (1 日) です。デフォルトは 600 秒です。

---

7. セキュリティーの優先リストを順序付けます。セキュリティーの優先リストについての詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

---

8. サーバーに関するそのほかのすべての指定を完了した後、妥当性検査を行ってコミットし、すべての作業を完了後、会話を活動化します。

会話が活動化されたことをシステムから通知されると完了です。

Kerberos 使用のためのクライアントのセットアップ・ステップ

始める前に: SSL 基本認証がセットアップされていなければなりません。

OS/390 SecureWay Security Server OS/390 用ネットワーク認証およびプライバシー・サービス (Kerberos) をインストールして構成する必要があります。クライアントが Kerberos を使用する各 OS/390 または z/OS イメージ上で SecureWay Security Server (KDC) を使用可能にします。詳細は、*z/OS SecureWay Security Server Network Authentication Service Administration*, SC24-5926 を参照してください。

Kerberos を使用するようクライアントをセットアップするには、以下のステップを実行します。

1. RACF を使用して、Kerberos クライアントとして組み込む各 OS/390 または z/OS ユーザーをローカル・レルムの Kerberos プリンシパルにマップします。**例:**

   ```
   ALTUSER client_ID PASSWORD(CBIVP) NOEXPIRED KERB(KERBNAME(kerberos_principal))
   ```

   ここで、

   client_ID
   クライアントのユーザー ID。

   kerberos_principal
   この OS/390 または z/OS ユーザー ID に関連付ける Kerberos プリンシパル名。

   ヒント: セキュリティー管理者が OS/390 または z/OS RACF レジストリーを Kerberos にマイグレーションするときに使用できるユーティリティーがあります。このユーティリティーは次の Web サイトで入手できます。

   ```
   http://sandbox.s390.ibm.com/products/racf/kmigrate.html
   ```

---

2. RACF を使用して、ターゲット・サーバーが存在するレルムとクライアントが存在するレルムとの間にレルム間信頼関係をセットアップします。**例:** クライアントが Kerberos レルム CLIENTREALM にあり、サーバーが SERVERREALM にある場合は、次のように入力します。

   ```
   RDEFINE REALM /.../CLIENTREALM/krbtgt/SERVERREALM KERB(PASSWORD(password1))
   RDEFINE REALM /.../SERVERREALM/krbtgt/CLIENTREALM KERB(PASSWORD(password2))
   ```

   password1 と *password2* にはパスワードを指定します。各 RACF データベースに対してこれらの 2 つのコマンドを発行する必要があります。

---

3. RACF を使用して、サーバー・レルムで外部ユーザーのマッピングをセットアップします。**例:**

   a. 外部レルムのすべてのプリンシパルを単一のユーザー ID にマップするには、次のように入力します。

      ```
      RDEFINE KERBLINK /.../foreign_realm APPLDATA('user_ID')
      ```

   b. 外部レルムのプリンシパルを個別にユーザー ID にマップするには、次のように入力します。

      ```
      RDEFINE KERBLINK /.../foreign_realm/principal APPLDATA('user_ID')
      ```

ここで、

foreign_realm
外部レルム。

user_ID
MVS ユーザー ID。

principal
プリンシパル。

RACF コマンドが正常に終了すると完了です。

管理アプリケーションの新しい管理者の追加

管理アプリケーションのデフォルトの管理者は CBADMIN です。管理者を追加する場合は、次の作業を実行する必要があります。

サブタスク	関連手順 (参照個所)
MVS ユーザー ID の作成 (または現在のユーザー ID を使用する) **注:** 新しい管理者ユーザー ID に CBADMIN と同じ RACF 権限を与えます。	*z/OS TSO/E 管理*, SA88-8626 または *z/OS SecureWay Security Server (RACF)* セキュリティー管理者のガイド, SA88-8613
LDAP のアクセス・コントロール・リストの更新	『LDAP のアクセス・コントロール・リストの更新ステップ』
管理アプリケーションに対する新しい管理者の定義	*WebSphere Application Server V4.0 for z/OS and OS/390:* システム管理ユーザー・インターフェース, SA88-8656
新しい管理者ユーザー ID へのシステム管理データベース権限の付与	88ページの『新しい管理者へのデータベース権限の付与ステップ』

LDAP のアクセス・コントロール・リストの更新ステップ

管理アプリケーションの管理者を追加するには、その管理者を LDAP のアクセス・コントロール・リストに追加する必要があります。

始める前に: LDAP サーバーをセットアップする必要があります。WebSphere for z/OS の管理目的ですでに排他的な LDAP サーバーをセットアップ済みであるということを前提にしています。LDAP サーバーのセットアップについて

の詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

また、現在 LDAP サーバーが使用している bboslapd.conf ファイルも必要です。

以下のステップを実行して、LDAP のアクセス・コントロール・リストを変更してください。

1. bboslapd.conf ファイルを表示して、以下のものをメモに取ります。
   a. 管理者の識別名。**例:**

      ```
      adminDN         "cn=CBAdmin"
      ```

   b. 管理者のパスワード。**例:**

      ```
      adminPW         mypass
      ```

   c. WebSphere for z/OS ネーム・スペース構造のルート・ネーミング・コンテキスト (RDN)。**例:**

      ```
      suffix          "o=BOSS,c=US"
      ```

---

2. LDAP サーバーを開始します。

   ```
   S BBOLDAP
   ```

---

3. ldapcp コマンドを使用して現行のアクセス・コントロール・リストを取り出します。**例:**

   ```
   /u/myself-> ldapcp -p 1389
   GLD4005I Environment variable file not found.  Environment variables not set.
   GLD6009I No DN entered.  Enter DN now.
   ldapcp> cn=CBAdmin
   GLD6010I No password entered.  Enter password now.
   ldapcp>

   GLD6019I Communicating with server on port 1389.
   ldapcp> acl q ob "o=boss,c=us"
    object = o=boss,c=us
    aclSource = O=BOSS,C=US
    aclPropagate = TRUE


    acl = access-id:CBADMIN:object:ad:normal:rwsc

    acl = access-id:CBSYMCR1:object:ad:normal:rwsc

    acl = group:CN=ANYBODY:normal:rsc
   ```

```
 acl = access-id:CN=BOSSAdmin,O=BOSS,C=US:object:ad:normal:rwsc

ldapcp>quit
```

---

4. ホーム・ディレクトリーに新規ファイルを作成します (たとえば、acl_update.txt)。そのファイルに以下の行を追加します。

```
dn: o=boss, c=us
changetype:modify
replace:x
```

---

5. ファイルに追加した最初の 3 行に続けて、86ページの3 のステップで取り出したそれぞれの acl 行に aclentry ステートメントを追加します。USER1 に新しい aclentry ステートメントを追加します。

   注:

   a. 最後にダッシュ ('-') を追加することが重要です。

   b. ldapcp コマンドの出力形式は、入力 aclentry 行と同じではありません (たとえば、「acl=」 は 「aclentry:」 に変更しなければなりません)。

   c. 例において、USER1 の aclentry は、USER1 に CBADMIN と同じ権限を付与しています。

   例:

```
aclentry: access-id:cn=BOSSAdmin, o=boss, c=us:normal:rwsc:object:ad
aclentry: access-id:USER1:normal:rwsc:object:ad
aclentry: access-id:CBADMIN:normal:rwsc:object:ad
aclentry: access-id:CBSYMCR1:normal:rwsc:object:ad
aclentry: group:CN=ANYBODY:normal:rsc
-
```

---

6. 更新ファイルを保管して、以下の ldapmodify コマンドを実行します。

```
u/myself-> ldapmodify -v -p 1389 -D "cn=CBAdmin" -w mypass -f acl_update.txt
```

   結果: ldapmodify は以下のような応答を返します。

```
modifying entry o=BOSS, c=US
```

---

7. 86ページの3 のステップを繰り返して、アクセス・コントロール・リストに新規ユーザーが追加されたことを確認します。

アクセス・コントロール・リストに新規ユーザーがあれば完了です。

新しい管理者へのデータベース権限の付与ステップ

新しい管理者には、CBSYSMGT_PKG に対する実行権限、および管理者がシステム管理データベースに J2EE アプリケーションを展開するときに必要とされるテーブルに対する選択、更新、挿入、および削除権限を付与しなければなりません。

始める前に: DB2 for z/OS または OS/390 SYSADM 権限を持つユーザー ID を持っていることが必要です。

次のステップを実行して、新しい管理者にデータベース権限を付与します。

⇔ 次のコマンドを入力します。

```
GRANT EXECUTE ON PACKAGE CBSYSMGT_PKG.*    TO user_ID

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE
BBO.BBOMT80_J2EEAPP TO user_ID;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE
BBO.BBOMT81_MODULE TO user_ID;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE
BBO.BBOMT82_COMPONENT TO user_ID;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE
BBO.BBOMT83_METHOD TO user_ID;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE
BBO.BBOMT86_DATASI TO user_ID;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE
BBO.BBOMT87_COMP_DS TO user_ID;
```

user_ID には、定義した管理者ユーザー ID を指定します。

GRANT コマンドが正常に終了すると完了です。

Java サーバー・アプリケーションのメッセージおよびトレース・データのロギング

アプリケーション・メッセージおよびトレース・データのロギングを行う WebSphere for z/OS サポートを使用して、WebSphere for z/OS サーバーで実行される Java アプリケーションの信頼性、可用性、および保守容易性を高めることができます。このサポートを使用すると、MVS マスター・コンソー

ル、エラー・ログ・ストリーム、または WebSphere for z/OS のコンポーネント・トレース (CTRACE) データ・セットに Java アプリケーションのメッセージが出力されます。アプリケーションの複数のトレース・エントリーは、同じ CTRACE データ・セットに出力することができます。

メッセージ出力先の決定

基幹業務アプリケーションの重大エラー状態を報告するメッセージを MVS マスター・コンソールに送りたい場合があります。オペレーターはマスター・コンソールを通じてアプリケーションの状況を示すメッセージを受信し、必要に応じて対応処置を取ることができます。また、メッセージをマスター・コンソールに送ることによって、アプリケーションの処理に関連した特定の条件やイベントに対する処置を行う自動化パッケージを起動することができます。

アプリケーションがコンソールへ発行するメッセージは、そのメッセージ・タイプによって、エラー・ログ・ストリームか、WebSphere for z/OS の CTRACE データ・セットにも出力されます。これらのシステム・リソースにメッセージをロギングすることによって、アプリケーションの処理に関連したエラーをより簡単に診断することができます。同様に、CTRACE データ・セットへのトレース・データのログ要求を発行するという方法も、エラー状態を記録する、または診断目的でアプリケーション・データを収集するための 1 つの方法です。

メッセージおよびトレース・データのロギング時のシステム・パフォーマンス

収集するトレース・データの量とタイプを選択することで、パフォーマンスを優先させる場合には最小のトレースでアプリケーションを実行し、問題を再現して付加的な診断情報を収集する必要がある場合には詳細なトレースを指定してアプリケーションを実行することができます。

エラー・ログ・ストリーム、WebSphere for z/OS の CTRACE データ・セット、およびマスター・コンソールは、主に、システム・コンポーネントおよび基幹アプリケーションのモニターと診断データの記録を目的としています。ユーザーのインストール構成によって、これらのリソースへのアプリケーション・メッセージおよびデータの送信が、システム・パフォーマンスに悪影響を与えることがあります。たとえば、アプリケーション・データを CTRACE データ・セットに送信する場合、データ・セット内のトレース・エントリーがより速くラップする可能性があります。これは、ラップが行われると、システムは既存のエントリーに新しいエントリーを上書きするため、重大な診断データを失う可能性があるということです。このロギング・サポートは慎重に使用してください。

注: WebSphere for z/OS のメッセージおよびトレース・データのロギング・サポートは、Java アプリケーションのみに使用でき、Java アプレットには使用できません。Java サーバー・アプリケーションのメッセージおよびトレース・データのロギングについての詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: J2EE アプリケーションのアセンブル*, SA88-8654 を参照してください。

アプリケーション・メッセージの MVS マスター・コンソールへの発行

WebSphere for z/OS の Java での信頼性、可用性、および保守容易性サポート (JRAS) を使用すると、Java アプリケーションのメッセージを MVS マスター・コンソールに発行することができます。基幹業務アプリケーションの重大エラー状態を報告するメッセージや自動化パッケージを起動するメッセージを、マスター・コンソールに送りたい場合があります。

アプリケーションが発行するメッセージは、エラー・ログ・ストリームまたは WebSphere for z/OS が使用するコンポーネント・トレース (CTRACE) データ・セットへ出力することができます。

メッセージをロギングするという方法も、エラー状態を記録する、または診断目的でアプリケーション・データを収集するための 1 つの方法です。

メッセージ・ロガーの使用

WebSphere for z/OS は、アプリケーションのメッセージを処理するメッセージ・ロガーを作成および管理するコードを提供します。WebSphere for z/OS は、それぞれの固有の組織 / 製品 / コンポーネントにつき 1 つのメッセージ・ロガーのみを作成します。したがって、ユーザーは、特定のアプリケーションのエラー・ログ・ストリームまたは CTRACE データ・セットに記録されたメッセージをより簡単に確認することができます。メッセージ・ロガーは、Java アプリケーションが実行される Java Virtual Machine (JVM) for WebSphere for z/OS サーバーで実行されます。

メッセージ・ロガーを使用するには、Java アプリケーションで次のことを実行します。

1. メッセージ・ロガーを定義する。
2. メッセージ・ロガーの作成を WebSphere for z/OS に指示するメソッドを呼び出す。
3. アプリケーションの適切な位置にメッセージをコーディングする。

詳細情報:

- 一般情報については、*WebSphere Application Server V4.0 for z/OS and OS/390: J2EE アプリケーションのアセンブル*, SA88-8654 を参照してください。
- エラー・ログ・ストリームのセットアップ方法については、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

コールド・スタート WebSphere Application Server

詳細情報:

- コールド・スタートの準備については、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照してください。
- コールド・スタートの手順については、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652 を参照してください。

以前にもコールド・スタートを実行したことがない場合は、WebSphere for z/OS と共に提供された初期ファイルではなく、コールド・スタートで保管した現行の構成ファイルを使用してください。

第8章 WebSphere for z/OS のチューニングおよびパフォーマンス・モニター

この章では、EJB とサーブレット EJB 統合ランタイムについての一般的な WebSphere for z/OS のチューニング・ガイドラインとパフォーマンス・モニター手順について説明します。MOFW オブジェクト (EJB の先行オブジェクト) に特有のチューニング考慮事項は、この章の最後に記載されています。

注: この章では、スタンドアロン・サーブレット・ランタイム、またはスタンドアロン Web サーバーのチューニング考慮事項については説明していません。

WebSphere for z/OS プログラミング・モデルとランタイムの目標の 1 つは、アプリケーション開発者がアプリケーションを作成および展開するために必要な作業を大幅に単純化することです。WebSphere for z/OS によって、アプリケーション・プログラマーはアプリケーション開発に伴なう面倒な作業の大部分から解放されるとみなすことができます。

たとえば、WebSphere for z/OS のアプリケーション・コードは遠隔通信に直接かかわることはありません。その代わりに、オブジェクト (ローカルでもリモートでも可) を見つけてメソッドを駆動します。したがって、WebSphere for z/OS アプリケーションではソケット呼び出しや TCP/IP プログラミングを直接使用することはありません。

実行したい操作と、その操作を実行する場所を切り離すことは、アプリケーション・プログラマーを厄介な作業から解放する 1 つの方法です。その他の考慮事項は、一部のタイプの bean、場合によってはユーザー認証、およびスレッド化に対するデータ呼び出しを処理する必要がないということです。通常、ソケット、RACF 呼び出し、またはスレッド化管理に影響するアプリケーション・コードからの呼び出しはありません。アプリケーション・プログラマーをこの作業から解放するということは、この作業が行われないということではありません。このことは、それに代わり DBA、ネットワーク管理者、セキュリティー管理者、およびパフォーマンス分析者の作業が増える可能性があることを意味します。

WebSphere for z/OS のチューニングおよびパフォーマンス・モニター

この章では、WebSphere for z/OS のパフォーマンス・チューニングに焦点を当てて説明します。ランタイムの性質にはオペレーティング・システム (OS) とミドルウェアの多種多様なコンポーネントが関係するため、この解説は複雑なものになります。

WebSphere for z/OS のチューニング・ガイドラインの説明を読む前に注意しておかなければならないのは、ミドルウェアをいかにうまくチューニングしても、アプリケーションの設計やコードに不備があればそれを補うことはできないということです。アプリケーション・コードを重視することは、パフォーマンスの改善につながります。作成や設計が不完全なアプリケーション・コードを変更することで、パフォーマンス全体が大幅に改善されることがしばしばあります。

WebSphere for z/OS ランタイムのチューニング

診断

まず始めに、WebSphere for z/OS の構成を検討します。これを簡単に実行する方法の 1 つは、SDSF (システム表示 / 検索機能) でアプリケーション制御およびサーバー領域を調べることです。各サーバーが開始されると、ランタイムは現行の構成データをジョブ・ログに出力します。

注: すべての構成値が出力されるようにする、SHOW_SERVER_SETTINGS=YES という環境変数があります。

基本データから開始して、必要なデータ以上の診断データを収集しないようにする必要があります。WebSphere for z/OS トレース・オプションを検査して、TRACEALL=0 または 1、および TRACEBASIC および TRACEDETAIL が設定されていないことを確認してください。

注: TRACEALL=2 を設定すると、パフォーマンスが 10 倍 (TRACEALL=3 の場合はそれ以上) 低下する可能性があります。したがって、IBM サポート・チームとともに問題をデバッグしている場合以外は、TRACEALL を 1 より大きい値に設定しないことをお勧めします。

TRACEBASIC および TRACEDETAIL によって、WebSphere for z/OS ランタイムで TRACEALL=2 と 3 のそれぞれに対応するコンポーネント特有のトレース・レベルを設定することができます。これらを増分的に指定すると、トレースするコンポーネントの数が増加するにつれてオーバーヘッドも増大します。最も単純なレベルのトレースであっても、アプリケーションのパフォーマンスを 2 倍低下させます。

トレースのどのレベルを使用する場合でも (TRACEALL=1 も含む)、TRACEBUFFERLOC を CTRACEに設定してください。TRACEALL=1 は、ERROR ログだけでなく TRACE ログにも例外を書き込みます。CTRACE は SYSPRINT よりはるかに効果的なコレクションメカニズムであり、パフォーマンスを改善します。

メモリー所要量を削減するために、TRACEBUFFERNUMBER=4 および TRACEBUFFERSIZE=128 を設定できます。これによって 512KB のストレージがトレース・バッファー用に取得されます (最小許容量)。

JRAS トレースを使用不可にします。これを実行するには、トレース設定ファイルで以下の行を探します。

```
com.ibm.ejs.*=all=enable
com.ibm.ws390.orb=all=enable
```

次に、両方の行の「=enable」を「=disable」に変更するか、または この 2 行全体を削除します。詳しくは、*WebSphere Application Server V4.0 for z/OS and OS/390: メッセージおよび診断* の『第 4 章 Java サーバー・アプリケーションのトレース』を参照してください。

SM GUI サーバー指定で、各サーバーの DEBUG を使用不可にします。オンライン・トレース (OLT) または分散デバッガーを使用している場合以外は、DEBUG 許可フィールドを NO に設定してください。これによって、WebSphere for z/OS ランタイムがそれぞれのメソッドごとにオンライン・トレース / デバッグ・インターフェースを呼び出さないようにします。

プログラムの位置

構成について次に検討することは、プログラム・コードの位置です。IBM では、適切な量の WebSphere for z/OS コード自体を LPA (リンク・パック域) にインストールし、残りをリンク・リストにインストールすることをお勧めします。これにより、パフォーマンスに悪影響を与えるおそれのある不要な steplib を除去することができます。現時点では、IBM では HFS にランタイムを配置した場合のパフォーマンスを測定しておらず、USS システムおよびユーザー共用ライブラリーのパフォーマンスに関して解説することもできません。ただし、簡素化のため、C/C++ アプリケーション・コードはすべて HFS に置くことをお勧めします。ただし、これを行うにはパフォーマンス・コストがかかります。サーバー領域のみがアプリケーション・コードに対する可視性を持つようにしてください。制御領域は通常、steplib なしで実行されます。これは、必要なコードがすべてシステム位置にあるためです。制御領域およびサーバー領域 proc の STEPLIB DD が不要なものを指示しないようにしてください。

PATH ステートメントを検討して、必要なプログラムのみが PATH にあること、また、PATH の順序が、頻繁に参照されるプログラムが前に来るように配置されていることを確認してください。Java を使用している場合は、98ページの『JVM』を参照してください。

ストレージ

WebSphere for z/OS サーバーに適用された仮想記憶域の量が充分であることを確認してください。このサーバーでは、通常は z/OS または OS/390 上の従来のアプリケーション・サーバーよりもはるかに多くの仮想メモリーが使用されます。proc の JCL (ジョブ制御言語) の REGION の設定値は大きくする (実行するには最低でも 128MB) 必要があります。また、高いスループットが必要な場合はさらに大きい値に設定します。RMF またはその他のパフォーマンス・モニターを使用すると、仮想記憶域の使用状況がわかります。サーバー領域 proc に REGION=0M を指定して、これにより、オペレーティング・システムに使用可能なすべての領域 (約 2GB) を与えるように指示してもかまいません。

注: REGION=0M および IEFUSI について詳しくは、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652の『第 2 章 基本的な OS/390 または z/OS 環境の準備』(特に『メモリーの使用に関する推奨事項』の節) を参照してください。

プログラム位置の節で説明したとおりにランタイムの大部分を LPA に入れないと、ページング・サブシステムの処理量は負荷の増大につれて増加します。負荷が増加するにつれてさらに多くのサーバー領域が開始され、ますます多くの負荷をページング・サブシステムに追加します。

LE ヒープ

LE ヒープは次に考慮すべきレベルのストレージ管理です。サーバーについては、IBM では HEAP および HEAPPOOL のデフォルト値をサーバー・メインプログラムにコンパイルしました。これらのデフォルト値は単純なアプリケーションのための適切な開始点となります。JCL で PARM= の LE (層エンティティー) 機能 RPTSTG(ON) を使用して、アプリケーション・サーバーのストレージ使用率に関するレポートを取得することができます。また、proc の PARM= を使用してその値を変更することができます。RPTSTG はストレージ使用情報の収集に使用するとパフォーマンスを多少低下させるため、除去してください。z/OS または OS/390 上で稼働するクライアント・プログラムについては、クライアントの proc に少なくとも HEAPP(ON) を指定して、デフォルトの LE ヒープ・プールを取得することをお勧めします。

注: LE HEAPCHECK を使用する場合は、初期化されていないストレージがコードに組み込まれていないことが確認できたらこれをオフにしてください。HEAPCHECK は非常にコストが高くなる可能性があります。

ガーベッジ・コレクションおよび JVM_HEAPSIZE

充分な JVM_HEAPSIZE を指定することは、Java のパフォーマンスにとって重要です。JVM には JVM のストレージ管理に使用するためのしきい値があります。しきい値に達すると、ガーベッジ・コレクター (GC) が起動されて未使用のストレージを解放します。GC は Java のパフォーマンスを大きく低下させる場合があります。

GC の実行回数を減らすために、JVM に与えるメモリーの量を増やすことができます。これを実行するには、JVM_HEAPSIZE に指定する値を大きくします。デフォルトの 256M は開始点として適切な値であり、アプリケーションの規模の増大 (大きいほうがよい) に応じてさらに大きくする必要があります。デフォルトでは、サーバーは JVM_HEAPSIZE=256M および JVM_MINHEAPSIZE=256M で実行されます。

注:

1. JVM_MINHEAPSIZE は JVM_HEAPSIZE と等しい値に設定することをお勧めします。これは、GC によりストレージが解放される前に、割り振られたストレージを完全に充てんすることができるためです。等しい値に設定しない場合、GC は絶えず実行され、小単位のストレージを保守しようとするため、パフォーマンスが危機にさらされます。
2. 領域が、指定した JVM ヒープを保持するのに充分な大きさであることを確認してください。

ガーベッジ・コレクションによる影響を受けているかどうかを判別するために、JVM_ENABLE_VERBOSE_GC を指定できます。これを指定すると、ガーベッジ・コレクターが実行されるごとに出力ストリームにレポートが書き込まれます。これはヒューマン・フレンドリーなレポートではありませんが、Java GC によって何が実行されているかを知ることができます。

サーバー領域のガーベッジ・コレクション

WebSphere for z/OS にはサーバー領域ガーベッジ・コレクションと呼ばれる機能 (Java ガーベッジ・コレクションとの混同を避けるため) が備えられています。この機能により、インストール・システムに特定のサーバー領域で実行するトランザクションの数に対するしきい値を指定して、サーバー領域が浪費されることを防ぎます。これは、ストレージのリークがあるアプリケーションのパフォーマンス改善に大きく役立ちます。

サーバー領域ガーベッジ・コレクションに対するデフォルト指定は、50000 トランザクションです。これはつまり、50000 トランザクションを超えると、サーバー領域はそれ以上新規の作業を選出しないことを意味します。旧サーバー領域がすでに保持している作業を完了させている間に、新規のサーバー領域が開始され新規の作業を選出します。作業が完了すると旧サーバー領域は終了し、新規のサーバー領域でプロセスが継続されます。

アプリケーション・コードがストレージをリークしている場合は、それぞれのストレージ取得が次第に低速になる可能性があります。指定されたトランザクション数を超えると新規のサーバー領域を使用することができ、ストレージ取得は再び高速になります。この手順では、最初はパフォーマンスが良好であり、その後低下してサーバー領域がリサイクルされると再び改善されるというように、いくぶん鋸歯状のパフォーマンス曲線が生成されます。長期的な観点から見たアプリケーション・リークのソリューションは明らかにアプリケーションを修正することですが、即時にアプリケーションを修正できない場合にはサーバー領域ガーベッジ・コレクションが有用です。

JVM

Java を使用する場合は、以下のことを確実に行ってください。

- WebSphere for z/OS でサポートされる JVM の最新バージョンを取得する。本書の発行時点では、WebSphere Application Server V4.0 for z/OS and OS/390 の JVM レベルは 1.3.0 PTF 3 です。
- 最新の PTF を取得する。ほとんどすべての PTF レベルで JVM のパフォーマンスを改善しているためです。
- 充分な `JVM_HEAPSIZE` を確保する (前述の説明を参照)。
- JIT (Just In Time) コンパイラーをアクティブにして実行する。これを行うには、環境ファイルから「`JAVA_COMPILER=`」オプションを省略するか、または「`JAVA_COMPILER=JITC`」を設定します。

  注: 「`JITC`」以外の値を指定すると、JIT がオフになります。
- libpath に JVM `libjava_g` のデバッグ・バージョンを指定しない。デバッグ・バージョンは非デバッグ・バージョンと同様に実行されません。
- `CLASSPATH` が必要なクラスのみを指すようにする (最も頻繁に参照されるクラスは、可能であればパスの前の方に置いてください)。
- Java 構成の一部として `CLASSPATH` を検証する。

注: z/OS および OS/390 での JVM のパフォーマンスについて詳しくは、`http://www.s390.ibm.com/java/perform.html` を参照してください。

パフォーマンス情報およびアカウンティング

WebSphere for z/OS では、WLM サービスの使用に依存することで一部のアカウンティング・データとパフォーマンス・データを収集します。この情報は RMF および RMF 作成の SMF レコードを使用して再度インストール・システムに提供されます。さらに、WebSphere for z/OS には独自の SMF レコードがあり、これによって WebSphere for z/OS に関するその他のドメイン特有の情報を収集します。最初に、SMF レコードまたは RMF データが必要でない場合は、これらをオフにします。これらの SMF レコードの制御は、SMFPRMxx parmlib ステートメントで実行されます。SMF 情報が必要な場合は、SMF parmlib を検査して、必要なデータのみを収集していることを確認します (レコード・タイプと詳細の両方)。WebSphere for z/OS SMF レコードの詳細を制御することができます。最後に、CI (制御インターバル) サイズが定義されていることを検査して、SMF データ・セットが適切に割り振られていることを検証します。最良のスループットを実現するには約 26K が適切です。

注: SMF について詳しくは、109ページの『第9章 システム管理機能 (SMF) による記録とモニター』を参照してください。

ワークロード・マネージャーのゴールとフィルター基準を設定することは、おそらくこの節の範囲外です。ただし、ユーザー ID とサーバー名に基づいて作業をパフォーマンス・グループに分類できるということを認識する必要があります。制御領域を適度にハイ・パフォーマンスのシステム・タスクとして分類する必要があります。

トポロジー

単一サーバーか複数サーバーか

WebSphere for z/OS を使用すると、アプリケーションを単一サーバーにインストールすることも、複数のサーバーに分散させることもできます。アプリケーションを区分化する理由は多数あります。ただし、パフォーマンスという点では、どのような場合でもアプリケーションを区分化するよりもすべてのアプリケーションを同一サーバーに置いた方がより優れたパフォーマンスが得られます。アプリケーションを複数のサーバーに区分化する場合は、少なくともシスプレックス内の各システムにレプリカ・サーバーがあるとパフォーマンスが改善されます。Application Server ランタイムは、可能な場合はシステムにとってローカルな呼び出しを保持しようとします。これは、たとえばソケットではなくローカル・プロセス間呼び出しを使用します。

単一トランザクションか複数トランザクションか
サーバー領域の実行についても、サーバー領域ごとに 1 つのトランザクションを実行するか、サーバー領域ごとに複数のトランザクションを実行するかという分離ポリシーに関する選択を行うことができます。パフォーマンスの観点からは、どちらが優れていると定義することはできません。

ローカル・クライアントかリモート・クライアントか
クライアントおよび最適化された通信が同じシステムで実行されるローカル・クライアントと、クライアント・コストがプラットフォームにない代わりにソケットの追加通信オーバーヘッドがあるリモート・クライアントは、ほぼ同等のものです。待ち時間についてはリモート・クライアントの場合よりもローカル・クライアントの場合の方が優れています。つまり、ローカル・クライアントの方が応答時間が早くなります。

1 つのサーバー・コピーか、あるいは複数のレプリカか
サーバーの複数のコピーをシステムに定義することができます。これらのコピーはレプリカと呼ばれます。レプリカを複数定義して実行した場合、1 つのみ定義するよりもパフォーマンスがわずかに向上することが確認されています。いくつかの利点はありますが、IBM では現時点ではパフォーマンスの改善を唯一の目的として複製された制御領域を作成することはお勧めしません。ただし、単一障害点を除去し、システム停止なしでローリング・アップグレードを処理する目的で使用することをお勧めしています。

コンテナー構成

WebSphere for z/OS では、複数のタイプの EJB と複数のトランザクション・ポリシーがサポートされています。各タイプの選択はパフォーマンスと密接に関係しています。現時点では完全な解説は提供できませんが、いくつかの経験法則を以下に示します。

EJB
WebSphere for z/OS には、2 つの基本的な bean のタイプであるセッションとエンティティーがあります。

セッション bean: WebSphere for z/OS のセッション bean には、ステートレスなセッション bean とステートフルなセッション bean があります。

ステートレス・セッション bean
オーバーヘッドが最も低いタイプの bean です。安価に作成でき、自動的に実行される機能はほとんどありません。また、アプリケーションによってクリーンアップされない場合はサーバーの終了時に失われます。

ステートフル・セッション bean
ステートレス・セッション bean よりも多少オーバーヘッドが多くなります。

エンティティー bean: WebSphere Application Server V4.0 for z/OS and OS/390 では、エンティティー bean には 2 つのタイプがあります。bean 管理下のパーシスタンス (永続性) (BMP)、およびコンテナー管理のパーシスタンス (CMP) の 2 つです。

パーシスタンスの管理は BMP の bean が行うため、BMP が CMP より高速であるかどうかはロードおよび保管のインプリメント方法に大きく依存します。CMP bean はパーシスタンスを管理します。BMP bean は、適切にインプリメントされた場合おそらく通常の CMP bean より高速になります。ただし、CMP はまもなく高性能になり、一般のアプリケーション・プログラマーが容易に保守できるようになります。

トランザクション・ポリシー
WebSphere for z/OS には以下の 7 つのトランザクション・ポリシーがあります。

- TRANSACTION_REQUIRES
- TRANSACTION_REQUIRES_NEW
- TRANSACTION_SUPPORTS
- TRANSACTION_NOT_SUPPORTED
- TRANSACTION_BEAN_MANAGED
- TRANSACTION_NEVER
- TRANSACTION_MANDATORY

この仕様には、ローカル・トランザクションおよびグローバル・トランザクションも含まれています。通常、ローカル・トランザクションが最も高速です。(ピン・トランザクションおよびキャッシング・トランザクションについては検討が必要です。)

MOFW の考慮事項

MOFW オブジェクトの場合、いくつかの追加トランザクション・ポリシーがあります。HYBRID_GLOBAL および SUPPORTS_HYBRID_GLOBAL です。これらのポリシーは、EJB におけるローカル・トランザクションに似ています。これらは完全な 2 フェーズ OTS 媒介トランザクションを前提としていないため、トランザクションのオーバーヘッドが削減されました。このポリシーは標準ではな

く、アプリケーションの振る舞いに影響を及ぼしたり移植性を損ねたりする可能性があるため、充分注意して使用してください。

WebSphere for z/OS の一時オブジェクトを HYBRID_GLOBAL トランザクションで実行すると、永続オブジェクトの約 2 倍高速になります。これは主に、他のリソース・マネージャーとの対話がないためトランザクションを調整する必要がなくなることが原因です。また、ディスクへのロギングを行う必要もないため、トランザクションの待ち時間が改善されます。

可能な場合は、読み取り専用永続データはピン・ポリシーを持つコンテナーに構成できます。つまり、データはサーバーごとに 1 回ずつデータベースから読み取られるのであり、トランザクションごとに検索されるのではなくメモリー内に保存されます。

MOFW 照会のパフォーマンスを改善するため、サーバー構成で環境変数 SOMOOSQL=1 がデフォルトにより設定されます。この環境変数を設定すると、照会を容易に DB2 にプッシュダウンすることができます。これにより照会のパフォーマンスが大幅に改善されます。IBM では、この変数を on に設定することをお勧めします。on に設定すると、照会で一部の NLS サポートが使用できなくなります。

また、照会呼び出しが DB2 にプッシュダウンされるようにしてください。これを最も簡単な方法で判別するには、DB2PM レポートで DB2 に対する呼び出しの詳細を確認します。DB2 に対して発行された照会ステートメントを検討し、DB2 からの取り出しがどのくらい行われたかを確認することができます。詳しくは、105ページの『DB2 のチューニングのヒント』を参照してください。

最後に、各ホーム (コンテナー) の定義にメソッド・レベル・アクセス検査が必要かどうかを指定するオプションがあります。必要でない場合は、これをオフにすることでパフォーマンスを改善できます。

セキュリティー

通例として、セキュリティーを強化すると 2 つの事態が発生します。トランザクションごとのコスト増大と、スループットの減少です。

デフォルトでは、WebSphere for z/OS はセキュリティーをオンにして実行されます。ランタイムでは、セキュリティー証明書情報を収集し、ユーザーおよびサーバーに送達するために必ず多少の費用がかかります。すべてのセキュリティー許可検査は SAF (RACF またはそれと同等の機能) で実行されるため、セ

キュリティーを制御するために SAF クラスを使用可能にするか使用不可にするかを選択することができます。クラスを使用不可にすると、オーバーヘッドのコストはごくわずかになります。

クラスがアクティブであると、クラス内のプロファイルの数が検査のパフォーマンス全体に影響を及ぼします。これらのプロファイルを (RACLISTed) メモリー・テーブルに入れると、アクセス検査のパフォーマンスが向上します。アクセス検査に対する監査制御もパフォーマンスに影響します。通常、監査は成功ではなく失敗について行います。監査イベントは DASD に記録され、アクセス検査のオーバーヘッドを増加させます。

EJBROLE を使用している場合は、メソッドに多くの役割を指定すると、実行する必要のあるアクセス検査の数が増加して全体的なメソッド・ディスパッチの速度が低下します。EJBROLE を使用していない場合は、クラスをアクティブにしないでください。

認証

認証処理にはいくつかのオプションがあります。

- **ローカル認証:** ローカル認証は高度に最適化されているため、最も高速です。
- **ユーザー ID およびパスワード認証:** ユーザー ID とパスワードを使用する認証では、最初の呼び出しのコストが高く、後続の各呼び出しのコストは低くなります。
- **Kerberos セキュリティー認証:** kerberos セキュリティーのコストについては、まだ特性を十分に説明していません。
- **SSL セキュリティー認証:** SSL セキュリティーは、パフォーマンス・オーバーヘッドが高いことが業界で広く知られています。ただし、ハードウェアから使用可能な多数の補助メカニズムがあるため、z/OS でも使用できます。ここで特性について解説することはできませんが、まず始めに Web サーバーの SSL の構成オプションについて説明します。

注: セキュリティーについて詳しくは、*WebSphere Application Server V4.0 for z/OS and OS/390: インストールおよびカスタマイズ*, GA88-8652を参照してください。

サーブレット /EJB 統合ランタイム

統合ランタイム・パフォーマンスに関する特性について解説することはできませんが、いくつかの一般的な解説を示します。サーブレットのみを実行している場合は、統合ランタイムは最初はパフォーマンスの向上を示しません。ただし、サーブレットは EJB に対する呼び出しを実行しているときに統合ランタ

イムから大きな利益を得ます。本来、統合ランタイムはリモート・メソッド呼び出しをローカルのプロセス内 EJB 呼び出し (この方がはるかに高速である) に変換します。サーブレットは EJB に依存してパフォーマンスを促進します。

パフォーマンス診断情報の収集

WebSphere for z/OS のランタイムにパフォーマンス上の問題があると思われるときに実行していただきたいいくつかの一般的な事項があります。まず、アプリケーションを実行して 15 分間のアプリケーションのパフォーマンスを示すサンプルをとってください。サンプル・データは RMF モニター I を使用して収集してください。RMF コレクションに設定していただきたいパラメーターを以下に示します。

```
CPU
CHAN
CYCLE(1000)
DEVICE(NOCHRDR)
DEVICE(COMM)
DEVICE(DASD)
DEVICE(NOGRAPH)
DEVICE(NOTAPE)
DEVICE(NOUNITR)
ENQ(SUMMARY)
INTERVAL(15M)
IOQ(DASD)
IOQ(COMM)
NOVSTOR
OPTIONS
PAGING
PAGESP
RECORD
REPORT(REALTIME)
NOSTOP
SYSOUT(H)
WKLD(PERIOD,SYSTEM)
TRACE(CCVUTILP)
```

注: また、開始点を判別するため、どの WLM サービス・クラスが Application Server ワークロードを表しているかを把握する必要もあります。

たとえば、他の何らかのリアルタイム・パフォーマンス・データを見ることにより、特定のアドレス・スペースにスループット上の問題があると思われる場合は、1 つまたは複数のアドレス・スペースのダンプを確認する必要が生じる場合があります。これを実行するには以下のパラメーターを使用します。

```
JOBNAME=(<jobname list>)
SDATA=(LSQA,PSA,SQA,SUM,SWA,TRT,WLM,CSA,RGN)
```

z/OS または OS/390 のチューニングのヒント

- まず始めに、CTRACE 構成を検討します。すべてのコンポーネントが MIN または OFF のいずれかに設定されていることを確認してください。これによって、使用されていないトレース情報を収集するといった不要なオーバーヘッドが除去されます。デバッグ中に、CTRACE がコンポーネントに対してオンになっており、問題がデバッグされるときにオフになっていないことがよくあります。
- 最良のパフォーマンスを得るため、LE および C++ ランタイムが LPA にロードされるようにしてください。
- 1 秒に 200 トランザクションを超えるスループットを実現するには、ロガーの RRS ログを CF ログ・ストリームに移動します。通常、迅速に完了するトランザクションは DASD I/O を必要としません。アーカイブ・ログは余分な DASD I/O を導入する可能性があるため、必要でない場合は除去することをお勧めします。アーカイブ・ログには完了したトランザクションの結果が含まれています。通常、アーカイブ・ログは不要です。
- 必要以上の SMF データを収集しないようにしてください。また、最も効果的にデータ・セットへの SMF データの書き込みが実行できるように、SMF データ・セットの CI サイズを大きくしてください。

DB2 のチューニングのヒント

DB2 のパフォーマンス・チューニングは、通常、WebSphere for z/OS アプリケーションのパフォーマンス全体にとって重要です。DB2 は、多くの場合セッションまたは EJB にとって望ましいデータ・ストアです。DB2 のチューニングについては解説書が多数存在するため、本書で完全な説明を提供することはおそらく不可能でしょう。いくつかの基本的なパフォーマンス・ガイドラインを以下に示します。(ガイドライン) テンプレート)

- まず、DB2 ログが充分な大きさであること、およびユーザーの最速のボリュームに割り振られていることを確認します。DASD 高速書き込み機能がある場合は、使用可能になっていることを確認します。IBM による実行では、この機能を使用した場合の差は約 2 倍でした (35 ミリ秒 (msec) の I/O 差異に基づく)。また、RMF レポートには実際の I/O の差異は示されていないように見えます。レポートでは、変更の前後に 0.1 ミリ秒の差異が示されていますが、GTF I/O トレースでは明らかに問題が SSCH であること、および 2 つの I/O 割り込み (1 つは即時、もう 1 つはその後) が示されていました。RRS CTRACE では、存続期間がコミット中に幾分高いことが示されていました。これは、データベース・テーブルが複数のシリンダーで定義されるようにするために役立ちます。

- 次に、バッファー・プールを調整して、最も頻繁に読み取るデータが可能な限りメモリー内にあるようにしていることを確認します。バッファー・プール・サイズの設定では、すべてのデータが保持できるよう充分なメモリーを確保しながら 2G を超えないように定義することが重要です。
- すべてのオブジェクトの基本キーに索引を定義することをお勧めします。そうしないと、テーブル・スペースのスキャンのコストが高くなります。
- いったんテーブルに充分なデータを挿入したら、テーブルをコンパクトにするために再編成するようにしてください。Runstats を実行すると、テーブル、列サイズ、およびアクセスに関する DB2 カタログ統計が最新の情報に更新されるため、最良のアクセス・パターンが最適化プログラムによって選択されます。
- 頻繁に使用する予定のテーブルの事前フォーマットを検討したい場合があります。事前フォーマットにより、実行時のフォーマットを避けることができます。
- さらに多くの接続呼び出し先スレッドを DB2 に定義する必要が生じます。Application Server では多数のスレッドを使用します。これはしばしばスループット・ボトルネックの原因になる場合があります。これは、サーバーが作成スレッドでスレッドが使用可能になるまで待機するためです。
- bean 開発者は、JDBC または SQLJ のいずれかを選択できます。JDBC は動的 SQL を利用するのに対して、SQLJ は通常静的であり、事前に作成された計画を使用します。SQLJ は計画を作成およびバインドするために特別なステップを必要としますが、JDBC では必要ありません。概して、SQLJ は JDBC よりも高速です。
- JDBC を使用する場合は、DB2 で動的ステートメント・キャッシングを使用可能にすることをお勧めします。そのためには、次のようにします。

  1. ZPARMS を変更して `CACHEDYN(YES) MAXKEEPD(16K)` に設定する。
  2. JDBC 計画などの計画を、`KEEPDYNAMIC(YES)` を使用してバインドする。

  アプリケーションによっては、これによって DB2 のパフォーマンスが大幅に改善される場合があります。特に、JDBC、LDAP、および MOFW 照会において役立ちます。
- イテレーターをコーディングするときは、名前付きまたは位置指定のいずれかを選択できます。パフォーマンス上の理由から、位置指定イテレーターをお勧めします。
- JDBC および SQLJ では、行全体を検索する汎用呼び出しではなく、必要なデータのみを検索する固有の呼び出しを記述することをお勧めします。フィールドごとにかなりのコストがかかります。
- シリンダーにテーブルを割り振ります (720 の倍数)。

- ログの CI サイズを大きくします。

RACF のチューニングのヒント

- 常に言えることですが、必要な機能以外はオンにしないでください。一般に、セキュリティーのコストは高度に最適化されています。ただし、EJBROLE が必要でない場合は、このクラスを RACF で使用可能にしないでください。
- RACLIST コマンドを使用して、メモリー内にパフォーマンスを改善する項目を入れるようにしてください。特に、以下 (使用した場合) に対して RACLIST を必ず使用してください。
  - ACEE
  - GTS
  - UID/GID
  - CBIND
  - EJBROLE
- SSL などを使用するとコストが高くなります。SSL を頻繁に使用する場合は、ハンドシェーク・プロセスを高速化するため、PCI 暗号カードなどの適切なハードウェアがあることを確認してください。

システム・ロガーのチューニングのヒント

- 可能であれば、CF ログを使用してください。
- CF ログを使用できない場合は、高速書き込み DASD を使用して、ログに大サイズの CI が割り振られていることを確認してください。
- OS/390 R10 (DB2 6.1 を装備) 以前は、ロガーに 4 つのログ・レコードを書き込むことが可能でした。アーカイブ・ログは不要であったため、トランザクションごとに 3 つまでレコードを取得することができました。DB2 7.1 では、PTF を使用して、トランザクションごとに 2 つまでログ書き込みを取得することができ、OS/390 R10 (DB2 7.1 PTF を装備) では、トランザクションごとに 1 つのログ・レコードを取得できるようになりました。
- いずれの場合も、ロガーをモニターして CF に充分なサイズがあること、およびオフロードがスループット全体に影響を及ぼしていないことを確認する必要があります。トランザクション・ログは、メインラインの単独共用 I/O 集中リソースの 1 つであり、チューニングを誤るとスループットに多大な影響を及ぼす可能性があります。
- DASD ロガーは 1 秒につき 450 I/O に制限されています。これは G4 の RAMAC III 陳述によるものです。WebSphere for z/OS ではトランザクショ

ンごとに 3 つのログ・レコードがロガーに書き込まれるため (z/OS または OS/390 R7 の場合)、約 150 トランザクション / 秒に制限されます。

- CF ロガーは G4、RAMAC 3 で 1 秒につき約 2700 I/O でした。これは 900 トランザクション / 秒 / CEC を意味します。当然ながら、G6 ではこの数はさらに大きくなります。

TCP/IP のチューニングのヒント

TCP/IP は、何らかの重大なリモート・メソッド遅延の原因となる場合があります。

1. 最初に、システムに充分なソケットを定義していること、およびデフォルトのソケット・タイムアウト値である 180 秒が高すぎないことを確認してください。
2. 次に、TCPIP プロファイル・データ・セットのポートの指定を検査して、NODELAYACKS が以下のように指定されていることを確認します。

   ```
   PORT 8082 TCP NODELAYACKS
   ```

   この変更により、実行時にスループットが 50% 向上する場合があります。
3. DNS 構成が最適化され、頻繁に使用するサーバーとクライアントのルックアップがキャッシュに入れられていることを確認してください。これはネーム・サーバーの存続時間 (TTL) 値に関連する場合があります。一方では、TTL に高い値を設定すると、優れたキャッシュ・ヒットが保証されます。ただし、高い値に設定した場合、デーモンがダウンするとネットワーク内でそのことが認識されるまでに多少時間がかかります。

第9章 システム管理機能 (SMF) による記録とモニター

この章では、システム管理機能を使用可能にして使用することによって、WebSphere for z/OS システムにおけるシステムおよびジョブ関連の情報を収集し記録する方法について説明します。これらの説明は、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 および *z/OS MVS システム管理機能 (SMF)*, SA88-8596 にも記載されています。システムおよびジョブ関連の情報は、ユーザーへの課金、システムの信頼性のレポート生成、構成の分析、作業のスケジュール、システム・リソースの使用率の判別、および組織で必要とするその他のパフォーマンス関連タスクを実行するために活用できます。

SMF 記録は以下のものに対して使用可能に設定することができます。

- **キャパシティー・プランニング**

  次のことを判別できます。

  - 実行されたトランザクション数。
  - 各サーバーで実行されるメソッドの完了までの平均時間と最大時間。
  - 各サーバー・インスタンスに接続されているクライアント数。それらのクライアントのうち、活動中のクライアント数。
- **アプリケーション・プロファイル**
  - アプリケーションを分解し、そのコンポーネント・パーツを表示することができます。
  - アプリケーションのコンポーネント・パーツに関するタイミング情報を生成することができます。
- **エラー・レポート**
  - ソフト・エラー (例外により、またはパフォーマンスに関連して生成されるエラー) の検出と記録を実行できます。
  - このエラー情報を使用して、しきい値に到達したときに何らかのアクションを実行するイベントを起動することができます。

WebSphere for z/OS が生成する SMF レコードのうち該当のものを使用して、インストール・システムで上記の機能を実行することができます。

SMF レコード・タイプ

アクティビティー・レコード とインターバル・レコード の 2 つのタイプの SMF レコードを作成できます。

アクティビティー・レコード
サーバー内の各アクティビティーが完了すると収集されます。アクティビティーとはビジネス機能の論理ユニットです。アクティビティーは、サーバーまたはユーザーによって開始されるトランザクションです。

インターバル・レコード
インストール・システムで指定された間隔で収集されたデータから成り、キャパシティー・プランニングおよび信頼性に関する情報を提供します。

サーバー・アクティビティー・レコード、コンテナー・アクティビティー・レコード、サーバー・インターバル・レコード、コンテナー・インターバル・レコード の 4 つのレコードが生成されます。以下に、それぞれのレコードについて説明します。これらのレコードの活動化方法についての詳細は、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照してください。

サーバー・アクティビティー・レコード

サーバー・アクティビティー SMF レコードは、WebSphere for z/OS Application Server 内で実行されるアクティビティーを記録するレコードです。このレコードを使用して、基本的なチャージ・バック・アカウンティングを実行できるほか、アプリケーションのプロファイルを作成して WebSphere トランザクション・サーバー内部で何が起こっているかを詳細に判定することができます。

サーバーまたはサーバー・インスタンス内で実行される各アクティビティーごとに、1 つのレコードが作成されます。アクティビティーが複数のサーバーで実行される場合は、各サーバーごとにレコードが書き込まれます。

このレコードを活動化するには、システム管理ユーザー・インターフェースのサーバー定義で「サーバー・アクティビティーの書き込み SMF レコード (Write Server Activity SMF Records)」チェック・ボックスをオンにします。

コンテナー・アクティビティー・レコード

コンテナー・アクティビティー SMF レコードの使用目的は、WebSphere トランザクション・サーバーの内部にあるコンテナー内で実行されるアクティビテ

ィーの記録です。このレコードを使用して、基本的なチャージ・バック・アカウント、アプリケーションのプロファイル作成、問題判別、およびキャパシティー・プランニングを実行できます。

WebSphere トランザクション・サーバーの内部にあるコンテナーで実行される各アクティビティーごとに 1 つのレコードが作成されます。アクティビティーが複数のサーバーで実行される場合は、そのアクティビティーに関する複数のレコードが書き込まれます。

このレコードを活動化するには、システム管理ユーザー・インターフェースのサーバー定義で「コンテナー・アクティビティーの書き込み SMF レコード (Write Container Activity SMF Records)」チェック・ボックスをオンにします。

サーバー・インターバル・レコード

サーバー・インターバル SMF レコードの使用目的は、WebSphere for z/OS アプリケーション・サーバー内で実行されるアクティビティーの記録です。このレコードは一定間隔で生成され、その間隔においてサーバー内で実行された作業の集約を表します。

その間隔においてインターバル・レコードが活動化されている各サーバー・インスタンスごとに、1 つのレコードが作成されます。サーバーに複数のサーバー・インスタンスが存在する場合、各サーバー・インスタンスごとにレコードが書き込まれるため、サーバー内で実行された作業全体を確認するには、処理後それらのレコードを組み合わせる必要があります。

このレコードを活動化するには、システム管理ユーザー・インターフェースのサーバー定義で「サーバー・インターバルの書き込み SMF レコード (Write Server Interval SMF Records)」チェック・ボックスをオンにします。

コンテナー・インターバル・レコード

コンテナー・インターバル SMF レコードの使用目的は、WebSphere トランザクション・サーバーの内部にあるコンテナーで実行されるアクティビティーの記録です。このレコードは一定間隔で生成され、その間隔においてコンテナー内で実行されたアクティビティーの集約を表します。このレコードを使用して、アプリケーションのプロファイル作成、問題判別、およびキャパシティー・プランニングを実行できます。

記録対象として指定された間隔内で、WebSphere トランザクション・サーバー内に存在する活動中の各コンテナーごとに 1 つのレコードが作成されます。サーバーに複数のサーバー・インスタンスが関連付けられている場合、各サーバー・インスタンスごとにそれぞれのコンテナーのレコードが作成されます。そ

の間隔においてコンテナー内で実行された作業全体を確認するには、処理後に各レコードを組み合わせる必要があります。

このレコードを活動化するには、システム管理ユーザー・インターフェースのサーバー定義で「コンテナー・インターバルの書き込み SMF レコード (Write Container Interval SMF Records)」チェック・ボックスをオンにします。

SMF による記録のセットアップ

ここでは、WebSphere Application Server に対する SMF 記録を使用可能にする方法、出力データ・セットのフォーマットを設定する方法、および SMF 記録を使用不化にする方法について説明します。

SMF 記録の使用可能化ステップ

SMF 記録を使用可能にするには、以下のステップを実行します。

1. WebSphere Application Server に対する SMF 記録は、システム管理ユーザー・インターフェース管理アプリケーションのサーバー定義で使用可能にします (*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照)。以下の選択項目があります。

   サーバー・アクティビティー・レコード
   「サーバー・アクティビティーの書き込み SMF レコード (Write Server Activity SMF Records)」チェック・ボックスをオンにします。

   コンテナー・アクティビティー・レコード
   「コンテナー・アクティビティーの書き込み SMF レコード (Write Container Activity SMF Records)」チェック・ボックスをオンにします。

   サーバー・インターバル・レコード
   「サーバー・インターバルの書き込み SMF レコード (Write Server Interval SMF Records)」チェック・ボックスをオンにします。

   コンテナー・インターバル・レコード
   「コンテナー・インターバルの書き込み SMF レコード (Write Container Interval SMF Records)」チェック・ボックスをオンにします。

---

2. SMFPRMxx parmlib メンバーを編集します。

a. 「ACTIVE」ステートメントを挿入して SMF 記録を実行することを示します。*z/OS MVS 初期設定およびチューニング ガイド*, SA88-8563 を参照してください。

b. SYS ステートメントを挿入して、システムに作成させる SMF レコードのタイプを指定します。たとえば、WebSphere Application Server タイプ 120 のレコードのみを選択するには、SYS(TYPE(120:120)) と指定します。パフォーマンスへの影響を最小限に抑えるためには、選択レコード・タイプ数をなるべく少なくする必要があります。

サーバーおよびコンテナーのインターバル・レコードでは、次のいずれかが使用されます。

- SM ユーザー・インターフェースのサーバーまたはコンテナー定義に指定された値
- SMF 製品設定の SMF parmlib メンバーで指定された値 (長さ 0 が指定されている場合)

サーバーおよびコンテナーのインターバル・レコードを作成する間隔を SMFPRMxx parmlib メンバーで指定することができます (サーバーまたはコンテナー定義に対して SM EUI によって間隔を指定していない場合)。SMF 記録のデフォルト間隔は 30 分です。詳しくは、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照してください。

---

3. 次のコマンドを入力して、DASD へのレコードの書き込みを開始します。

```
t smf=xx
```

xx は SMF parmlib メンバー (SMFPRMxx) の接尾部です。詳しくは、*z/OS MVS システム管理機能 (SMF)*, SA88-8596 を参照してください。

DASD への書き込みを活動化すると、データがデータ・セット (SMFPRMxx で指定されたもの) に記録されます。

出力データ・セットのフォーマット設定ステップ

SMF 記録の出力データ・セットを、画面や他の出力装置に出力するために可読形式にフォーマットするには、以下のステップを実行します。

注: 詳しくは、*z/OS MVS システム管理機能 (SMF)*, SA88-8596 および SMF ダンプ・プログラムを参照してください。この情報を簡単に要約したものを、以下に示します。

1. MVS コンソールから「`i smf`」と入力して SMF データ・セットを切り替えます。

   _____

2. SMF ダンプ・プログラム (IFASMFDP) を実行して、生ダンプから順次データ・セットを作成します。*z/OS MVS システム管理機能 (SMF)*, SA88-8596 に JCL のサンプルが収録されています。

   _____

3. レコード・タイプ 120 を表示できるプログラムを使用してデータ・セットを表示します。

   _____

レコード・タイプ 120 用の SMF レコード・インタープリター

WebSphere for z/OS SMF レコード・インタープリターは、IBM z/OS ユーティリティー・プログラム IFASMFDP からの完全な出力データ・セットを解釈するためのツールです。このインタープリターでは、すべてのレコード・タイプのヘッダー行に続いて、レコード・タイプ 120 の詳細ダンプが作成されます。

注: IFASMFDP を使用してシステムの SMF データ・セットから SMF データの特定のサブセットを抽出し、順次データ・セットに書き出す方法については、113ページの『出力データ・セットのフォーマット設定ステップ』を参照してください。

WebSphere for z/OS SMF レコード・インタープリターは、WebSphere for z/OS 関連のすべてのデータを印刷可能な出力ファイルにダンプします。これは Java ユーティリティーであるため、z/OS または OS/390 UNIX 環境下で Java 仮想計算機 (JVM) を使用して解釈し実行する必要があります。

WebSphere for z/OS SMF レコード・インタープリターの印刷可能出力は、次のようなものです。

```
 SMF file analysis starts ...

1Rec-Num Type    RecLn SmfDate  SmfTime
1------- ------- ----- -------- --------

      1 2   18 2000.332 08:53:40

      2 120.001 (SERVER ACTIVITY)   368 2000.332 08:41:56
```

```
System ID: SY1  Subsystem ID:
Flag: 94
#Triplets: 3
Triplet:    offset= 64     length= 32     count= 1
Triplet:    offset= 96     length= 188     count= 1
Triplet:    offset= 284     length= 84     count= 1

Triplet #1
ProductSectionData
  Version= 1  Codeset= IBM-1047
  Endian= 1  TimeStampFormat= 1 (S390STCK64)
  IndexOfThisRecord= 1       Total#OfRecords= 1       Total#OfTriplets= 3

Triplet #2
ServerActivitySectionData
  HostName: PLEX1
  ServerName: BBOASR1    ServerInstanceName: BBOASR1A
  #OfServerRegions= 1
  ASID1= 54     ASID2= 0     ASID3= 0     ASID4= 0     ASID5= 0
  UserCredentials: CBIVP
  ActivityType= 1 (method request)
  ActivityID:         * b501f924 68c54d0b 000000d4 00000009 * §.9.E(....M.... *
                      * 0926306b xxxxxxxx xxxxxxxx xxxxxxxx * ...,............ *
  WlmEnclaveToken:    * 00000020 000001f0 xxxxxxxx xxxxxxxx * .......0........ *
  ActivityStartTime:  * b501f924 68c54d0b xxxxxxxx xxxxxxxx * §.9.E(......... *
  ActivityStopTime:   * b501f94d bec8f425 xxxxxxxx xxxxxxxx * §.9(´H4......... *
  #InputMethods= 1     #GlobalTransactions= 0     #LocalTransactions= 0

Triplet #3
CommSessionSectionData
CommSession-----
  CommSessionHandle:  * 252c3220 00000001 xxxxxxxx xxxxxxxx * ............... *
  CommSessionAddress: jobname=BBOIVP   asid=0037
  CommSessionOptimization= 1 (local optimization)
  DataReceived= 329
  DataTransferred= 721
.......
 SMF file analysis complete.
```

順次ファイルのデータはレコードごとに生成されます。各レコードにはいくつかのトリプレットが格納され、それらのトリプレットは最初にレコードのヘッダー・セクション (レコードの冒頭部分) に記述されます。次に、このツールによって提供されたトリプレットの内容の記述が、レコード内での出現順に出力されます。

注: 詳しくは、119ページの『付録A. SMF レコード・タイプ 120 (WebSphere for z/OS)』 を参照してください。

各トリプレットにはセクションのデータが格納されます。次のような、いつかのタイプのセクションが定義されています。

- ProductSection

- ActivitySection
- CommSessionSection
- その他

WebSphere for z/OS SMF レコード・インタープリターは各セクションをそれぞれのセクション固有の方法で解釈し、解釈後のデータを出力ファイルに書き出します。

注: 同じくトリプレットを使用して編成されたサブセクションを含むセクションもあります。

SMF 表示ツールのインストールと呼び出し

Java SMF レコード・インタープリターは、bbomsmfv.jar という名前の jar ファイルの形式で提供されます。z/OS または OS/390 UNIX 環境からこれを使用するには、次のようにします。

1. JAVA_HOME 環境変数が現在の java インストール・システム (JAVA_HOME=../usr/bin/java/J1.3) を指していることを確認します。

   注: インタープリターに必要なレコード・サポートを暗黙的に含む最初の Java はバージョン 1.3 であるため、ここでは、1.3 以上の Java が指定されなければなりません。

2. ファイル「bbomsmfv.jar」をツール・ディレクトリーにコピーします。
3. z/OS または OS/390 のカタログ作成された順次ファイル「USER.SMFDATA」(前ページの説明にあるように IFASMFDP ユーティリティーを使用して作成されたもの) の SMF データを解釈するため、次のコマンドを実行します。

   ```
   java -cp bbomsmfv.jar com.ibm.ws390.sm.smfview.Interpreter "USER.SMFDATA"
   ```

SMF 記録の使用不能化ステップ

SMF 記録を使用不可にするには、以下のステップを実行します。

1. システム管理ユーザー・インターフェース管理アプリケーションのサーバー定義で、WebSphere Application Server に対する SMF 記録を使用不可にします (*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 を参照)。

   ---

2. MVS システム全体の SMF 記録を使用不可にする場合は、SMFPRMxx parmlib メンバーを編集します。SMFPRMxx を「NOACTIVE」に設定して、DASD への SMF レコードの書き込みを使用不可にします。

SMF レコード・タイプ 120 (78) － WebSphere for z/OS

WebSphere for z/OS SMF レコード・タイプ 120 についての詳細は、119ページの『付録A. SMF レコード・タイプ 120 (WebSphere for z/OS)』を参照してください。*z/OS MVS システム管理機能 (SMF)*, SA88-8596 も参照してください。

付録A. SMF レコード・タイプ 120 (WebSphere for z/OS)

この付録では、WebSphere for z/OS によって作成された SMF レコードのレイアウトについて説明します。

SMF データ収集プロセスの結果収集された情報は、通常、SMF データ表示ツールを使用して表示されます。このレコード・フォーマットの説明は、ユーザーのツール・プロバイダーが SMF データ表示ツールを設計できるようにすることを目的としています。システム管理者は、SMF データ表示ツールを使用する際にツール・プロバイダーに提供された説明を参照します。SMF データ表示ツールを使用する場合は表示データの量を制限する適切な選択を行う必要があるためです。たとえば、特定の時間フレームや、特定のコンテナー、クラス、およびメソッドのみを表示したい場合があります。また、レコード記述を参照する必要が生じる場合もあります。

SMF レコードの使用に関する追加情報は、*z/OS MVS システム管理機能 (SMF)*, SA88-8596 を参照してください。

レコード・タイプ 120 (78) - WebSphere for z/OS パフォーマンス統計

以下の節では、SMF レコード・タイプ 120 (78) - WebSphere for z/OS パフォーマンス統計を定義します。WebSphere for z/OS はレコード・タイプ 120 を作成して WebSphere for z/OS パフォーマンス統計を収集します。SMF レコード・タイプについて詳しくは、*z/OS MVS システム管理機能 (SMF)*, SA88-8596 を参照してください。

レコード・タイプ 120 (78) - WebSphere for z/OS パフォーマンス統計

レコード・タイプ 120 のすべてのサブタイプの形式は以下のとおりです。

- 標準ヘッダー・セクション
- サブタイプ x の個々のヘッダー拡張
- 製品セクション
- サブタイプ特有のセクション (以下を参照)

レコード・タイプ 120 のサブタイプは以下のとおりです。

- **サブタイプ 1: サーバー・アクティビティー・レコード**

  - **サーバー・アクティビティー・セクション** (各レコードにつき 1 セクション)

    1 つのサーバー内で発生した各アクティビティーに関する情報が含まれます。
  - **通信セッション・セクション** (各レコードにつきゼロ、1、または複数のセクション)

    各通信セッションに関する情報が含まれます。
- **サブタイプ 2: コンテナー・アクティビティー・レコード**
  - **コンテナー・アクティビティー・セクション** (各レコードにつき 1 セクション)

    1 つのコンテナー内で発生した各アクティビティーに関する情報が含まれます。
  - **クラス・セクション** (各レコードにつき複数のセクション)

    このアクティビティーに関与したすべてのクラスに関する情報が含まれます。
  - **メソッド・セクション** (各クラス・セクションにつき複数のセクション)

    このアクティビティーに関与した、このクラスのすべてのメソッドに関する情報が含まれます。
- **サブタイプ 3: サーバー・インターバル・レコード**
  - **サーバー・インターバル・セクション** (各レコードにつき 1 セクション)

    指定されたサーバー・インターバル内に発生したすべてのアクティビティーに関する総合情報が含まれます。
- **サブタイプ 4: コンテナー・インターバル・レコード**
  - **コンテナー・インターバル・セクション** (各レコードにつき 1 セクション)

    指定されたインターバルに 1 つのコンテナー内で発生したすべてのアクティビティーに関する総合情報が含まれます。
  - **クラス・セクション** (各レコードにつき複数のセクション)

    指定されたインターバルにこのアクティビティーに関与したすべてのクラスに関する情報が含まれます。
  - **メソッド・セクション** (各クラス・セクションにつき複数のセクション)

    指定されたインターバルにこのアクティビティーに関与した、このクラスのすべてのメソッドに関する情報が含まれます。

レコード環境

このレコードの生成について以下の条件が存在します。

- **レコード環境**

  マクロ SMFWTM (レコード出口: IEFU83)

  モード タスク

  ストレージ常駐
  31 ビット

レコード・マッピング

ここには、ヘッダー / 自己定義セクション、および製品セクションが記載されています。

ヘッダー / 自己定義セクション

このセクションには、共通 SMF レコード・ヘッダー・フィールド、およびトリプレット・フィールド (オフセット / 長さ / 数値) が含まれています。適用可能な場合は、レコードでその他のセクションを見つけてください。トリプレットの説明については、134ページの『トリプレットおよび SMF レコードの分割』 および *z/OS MVS システム管理機能 (SMF)*, SA88-8596 を参照してください。

オフセット	オフセット	名前	長さ	フォーマット	記述
0	0	SM120LEN	2	バイナリー	レコード長。このフィールドおよび次のフィールド (4 つのバイトの合計) によって RDW が構成されます (レコード記述子ワード)。詳細記述は、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 の 『標準 SMF レコード・ヘッダー』を参照してください。
2	2	SM120SEG	2	バイナリー	セグメント記述子 (レコード長フィールドを参照)

4	4	SM120FLG	1	バイナリー	**ビットの意味 (設定時)** 0: 新規 SMF レコード・フォーマット 1: 使用されるサブタイプ 2: 予約済み 3-6: バージョン標識* 7: 予約済み *詳細記述は、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 の『標準 SMF レコード・ヘッダー』を参照してください。
5	5	SM120RTY	1	バイナリー	レコード・タイプ 120(X'78')
6	6	SM120TME	4	バイナリー	レコードが SMF バッファーに移動された真夜中以降の時刻 (100 分の 1 秒単位)
10	A	SM120DTE	4	圧縮	レコードが SMF バッファーに移動された日付。形式は 0*cyydddF*。詳細記述は、*WebSphere Application Server V4.0 for z/OS and OS/390: システム管理ユーザー・インターフェース*, SA88-8656 の『標準 SMF レコード・ヘッダー』を参照してください。
14	E	SM120SID	4	EBCDIC	システム ID (SMFPRMxx SID パラメーターから取得)
18	12	SM120SSI	4	EBCDIC	SUBSYS パラメーターから取得したサブシステム ID (SSID)

22	16	SM120STY	2	バイナリー	レコード・サブタイプ 1: サーバーのアクティビティー 2: コンテナーのアクティビティー 3: サーバーのインターバル 4: コンテナーのインターバル
24	18	SM120TRN	4	バイナリー	このレコードのトリプレットの数 回トリプレットは、レコードのセクションを定義する 3 つの SMF フィールド (オフセット / 長さ / 数値) の集合です。オフセットは RDW からのオフセットです。 サブタイプ 1: 値はセッション数 +2 に等しくなります。 2 および 4: 値はクラス数 +2 に等しくなります。
28	1C	SM120PRS	4	バイナリー	RDW から製品セクションへのオフセット
32	20	SM120PRL	4	バイナリー	製品セクションの長さ
36	24	SM120PRN	4	バイナリー	製品セクションの数

サブタイプ 1 の個々のヘッダー拡張					
40	28	SM120SAS	4	バイナリー	RDW からサーバー・アクティビティー・セクションへのオフセット
44	2C	SM120SAL	4	バイナリー	サーバー・アクティビティー・セクションの長さ
48	30	SM120SAN	4	バイナリー	サーバー・アクティビティー・セクションの数
52	34	SM120CSS	4	バイナリー	RDW から通信セッション・セクションへのオフセット
56	38	SM120CSL	4	バイナリー	通信セッション・セクションの長さ
60	3C	SM120CSN	4	バイナリー	通信セッション・セクションの数

サブタイプ 2 の個々のヘッダー拡張					
40	28	SM120CAS	4	バイナリー	RDW からコンテナー・アクティビティー・セクションへのオフセット
44	2C	SM120CAL	4	バイナリー	コンテナー・アクティビティー・セクションの長さ
48	30	SM120CAN	4	バイナリー	コンテナー・アクティビティー・セクションの数

以下のトリプレットは 0 ～ n 回 (クラス・セクションごとに 1 回) 表示されます。					
52	34	SM120CLS	4	バイナリー	RDW からクラス・セクションへのオフセット
56	38	SM120CLL	4	バイナリー	クラス・セクションの長さ
60	3C	SM120CLA	4	バイナリー	クラス・セクションの数

サブタイプ 3 の個々のヘッダー拡張					
40	28	SM120SIS	4	バイナリー	RDW からサーバー・インターバル・セクションへのオフセット
44	2C	SM120SIL	4	バイナリー	サーバー・インターバル・セクションの長さ
48	30	SM120SIN	4	バイナリー	サーバー・インターバル・セクションの数

サブタイプ 4 の個々のヘッダー拡張					
40	28	SM120CIS	4	バイナリー	RDW からコンテナー・インターバル・セクションへのオフセット
44	2C	SM120CIL	4	バイナリー	コンテナー・インターバル・セクションの長さ
48	30	SM120CIN	4	バイナリー	コンテナー・インターバル・セクションの数

以下のトリプレットは 0 ～ n 回 (クラス・セクションごとに 1 回) 表示されます。					
52	34	SM120CLS	4	バイナリー	RDW からクラス・セクションへのオフセット
56	38	SM120CLL	4	バイナリー	クラス・セクションの長さ
60	3C	SM120CLN	4	バイナリー	クラス・セクションの数

製品セクション

オフセット	オフセット	名前	長さ	フォーマット	記述
0	0	SM120MFV	4	バイナリー	CB SMF バージョン
4	4	SM120COD	8	EBCDIC	SMF レコードのストリングがエンコードされる文字コード・セット
12	C	SM120END	4	バイナリー	SMF レコード内の数のエンコード
16	10	SM120TSF	4	バイナリー	タイム・スタンプのエンコード 1: S390STCK64: 時刻値は 64 ビットの S/390 ストア・クロック形式でエンコードされます。

再組み立て情報					
20	14	SM120IXR	4	バイナリー	このレコードの索引
24	18	SM120NRC	4	バイナリー	レコードの合計数
28	1C	SM120NTR	4	バイナリー	トリプレットの合計数

サブタイプ 1: サーバー・アクティビティー・レコード

1. **サーバー・アクティビティー・セクション** (各レコードにつき 1 セクション)

   1 つのサーバー内で発生した各アクティビティーに関する情報が含まれます。

2. **通信セッション・セクション** (各レコードにつきゼロ、1、または複数のセクション)

   各通信セッションに関する情報が含まれます。

サーバー・アクティビティー・セクション

オフセット	オフセット	名前	長さ	フォーマット	記述
0	0	SM120HNM	64	EBCDIC	WebSphere トランザクション・サーバー・ホスト名
64	40	SM120SNM	8	EBCDIC	WebSphere トランザクション・サーバー名

72	48	SM120SIN	8	EBCDIC	WebSphere トランザクション・サーバー・インスタンス名
80	50	SM120SNM	4	バイナリー	このアクティビティーのプロセスに関与したサーバー領域の合計数。該当する場合、最初の 5 つまでのサーバー領域アドレス・スペース ID が次の 5 つのフィールドにリストされます。
84	54	SM120SR1	4	バイナリー	要求が実行される特定の WebSphere トランザクション・サーバー・インスタンス・サーバー領域
88	58	SM120SR2	4	バイナリー	要求が実行される特定の WebSphere トランザクション・サーバー・インスタンス・サーバー領域
92	5C	SM120SR3	4	バイナリー	要求が実行される特定の WebSphere トランザクション・サーバー・インスタンス・サーバー領域
96	60	SM120SR4	4	バイナリー	要求が実行される特定の WebSphere トランザクション・サーバー・インスタンス・サーバー領域
100	64	SM120SR5	4	バイナリー	要求が実行される特定の WebSphere トランザクション・サーバー・インスタンス・サーバー領域
104	68	SM120CRE	8	EBCDIC	その元でアクティビティーが開始されるユーザー証明書
112	70	SM120ATY	4	バイナリー	このレコードが参照するアクティビティーのタイプ 1: メソッド要求: このレコードは、グローバル・トランザクションの一部ではないメソッド要求を参照します。 2: トランザクション: このレコードはトランザクションを参照します。

116	74	SM120AID	20	EBCDIC	アクティビティーのアイデンティティー
136	88	SM120WLM	8	HEX	WLM エンクレーブ・トークン
144	90	SM120AST	16	S390STCK	アクティビティー開始時刻
160	A0	SM120AET	16	S390STCK	アクティビティー停止時刻
176	B0	SM120NIM	4	バイナリー	入力メソッドの数
180	B4	SM120NGT	4	バイナリー	サーバー領域で開始されたグローバル・トランザクションの数
184	B8	SM120NLT	4	バイナリー	サーバー領域で開始されたローカル・トランザクションの数

通信セッション・セクション

オフセット	オフセット	名前	長さ	フォーマット	記述
0	0	SM120CSH	8	HEX	通信セッション・ハンドル
8	8	SM120CSA	64	EBCDIC	通信セッション・アドレス
72	48	SM120CSO	4	バイナリー	通信セッション最適化 1: ローカル通信セッション: セッションはローカル OS/390 最適化通信セッションです。 2: リモート通信セッション: セッションはリモート通信セッションです。 3: リモート暗号化 (SSL) 4: シスプレックス内のリモート
76	4C	SM120SDR	4	バイナリー	受信データ・サーバーにより受信されたバイト数
80	50	SM120SDT	4	バイナリー	転送データ・サーバーからクライアントに伝送されたバイト数

サブタイプ 2: コンテナー・アクティビティー・レコード

1. **コンテナー・アクティビティー・セクション** (各レコードにつき 1 セクション)

1 つのコンテナー内で発生した各アクティビティーに関する情報が含まれます。

2. **クラス・セクション** (各レコードにつき複数のセクション)
   このアクティビティーに関与したすべてのクラスに関する情報が含まれます。
3. **メソッド・セクション** (各クラス・セクションにつき複数のセクション)
   このアクティビティーに関与したクラスのすべてのメソッドに関する情報が含まれます。

コンテナー・アクティビティー・セクション

オフセット	オフセット	名前	長さ	フォーマット	記述
0	0	SM120HNM	64	EBCDIC	WebSphere トランザクション・サーバー・ホスト名
64	40	SM120SNM	8	EBCDIC	WebSphere トランザクション・サーバー名
72	48	SM120SIN	8	EBCDIC	WebSphere トランザクション・サーバー・インスタンス名
80	50	SM120ASR	4	バイナリー	要求が実行される特定の WebSphere トランザクション・サーバー・インスタンス・サーバー領域
84	54	SM120CNM	256	EBCDIC	WebSphere コンテナー名
340	154	SM120CTP	4	バイナリー	コンテナー・トランザクション・ポリシー値 • 1: 必要なトランザクション • 2: 同一サーバーのハイブリッド・グローバル • 3: ハイブリッド・グローバル • 4: 同一サーバーのハイブリッド・グローバルをサポート

344	158	SM120CSP	4	バイナリー	コンテナー・セキュリティー・ポリシー • 00001: DCE • 00010: ユーザー ID パスワード • 00100: ユーザー ID パスチケット • 01000: SSL • 10000: 非認証クライアント
348	15C	SM120WLM	8	HEX	WLM エンクレーブ・トークン
356	165	SM120ATY	4	バイナリー	このレコードが参照するアクティビティーのタイプ 1: メソッド要求: このレコードは、グローバル・トランザクションの一部ではないメソッド要求を参照します。 2: トランザクション: このレコードはトランザクションを参照します。
360	168	SM120AID	20	HEX	アクティビティーのアイデンティティー

クラス・セクション

オフセット	オフセット	名前	長さ	フォーマット	記述
0	0	SM120CLN	256	EBCDIC	コンテナーによって活動化されたクラスの名前
256	100	SM120NIC	4	バイナリー	作成されたこのクラスのインスタンス数
260	104	SM120NIA	4	バイナリー	活動化されたクラスのインスタンス数
264	108	SM120NIR	4	バイナリー	除去 (削除) されたこのクラスのインスタンス数
268	10C	SM120NIP	4	バイナリー	非活性化されたこのクラスのインスタンス数
272	110	SM120RMR	4	バイナリー	予約済み

276	114	SM120RMW	4	バイナリー	予約済み
280	118	SM120MN	4	バイナリー	このクラス・セクションのメソッド・トリプレットの数

以下のトリプレットは 0 ～ n 回 (メソッド・セクションごとに 1 回) 表示されます。					
284	11C	SM120MS	4	バイナリー	このクラス・セクションの先頭からメソッド・セクションへのオフセット
288	120	SM120ML	4	バイナリー	メソッド・セクションの長さ
292	124	SM120MN	4	バイナリー	メソッド・セクションの数

メソッド・セクション

注: クライアントが WebSphere for z/OS サーバー領域のオブジェクト・インスタンスに対してメソッドを呼び出すと、このメソッドによって同じオブジェクト・インスタンス内の他のメソッドが呼び出された場合、クライアントによって呼び出された最初のメソッドのみが SMF に記録されます。クライアントが同じオブジェクト・インスタンスで呼び出した後続のメソッドは記録されません。

オフセット	オフセット	名前	長さ	フォーマット	記述
0	0	SM120MNM	256	EBCDIC	メソッドの名前 **注:** SMF 記録方式でメソッドを識別する唯一の方法はメソッド名です。SMF 記録方式では完全なメソッド・シグニチャーが評価されません。
256	100	SM120NMI	4	バイナリー	アクティビティー中にメソッドが起動された回数
260	104	SM120NEX	4	バイナリー	コンテナーによって検出された非フレームワーク例外の数

264	108	SM120ART	4	バイナリー	平均応答時間: 応答時間はマイクロ秒単位で測定されます。この値が 2**31 マイクロ秒を超えると、このフィールドは否定され、正確度は秒数に変更されます。肯定値はマイクロ秒単位の時刻です。否定値は秒単位の時刻です。
268	10C	SM120MRT	4	バイナリー	最大応答時間: 応答時間はマイクロ秒単位で測定されます。この値が 2**31 マイクロ秒を超えると、このフィールドは否定され、正確度は秒数に変更されます。肯定値はマイクロ秒単位の時刻です。否定値は秒単位の時刻です。

サブタイプ 3: サーバー・インターバル・レコード

1. **サーバー・インターバル・セクション** (各レコードにつき 1 セクション)

   指定されたサーバー・インターバル内に発生した各アクティビティーに関する情報が含まれます。

サーバー・インターバル・セクション

オフセット	オフセット	名前	長さ	フォーマット	記述
0	0	SM120HNM	64	EBCDIC	WebSphere トランザクション・サーバー・ホスト名
64	40	SM120SNM	8	EBCDIC	WebSphere トランザクション・サーバー名
72	48	SM120SIN	8	EBCDIC	WebSphere トランザクション・サーバー・インスタンス名
80	50	SM120SST	16	S390STCK	サンプルがサーバーで開始された時刻
96	60	SM120SET	16	S390STCK	サンプルが終了した時刻
112	70	SM120NGT	4	バイナリー	インターバル中に、サーバー・インスタンスを介して実行されたグローバル・トランザクションの数
116	74	SM120NLT	4	バイナリー	インターバル中にサーバー・インスタンスによって開始されたローカル・トランザクションの数

120	78	SM120NCS	4	バイナリー	インターバルの終了時に存在している通信セッションの数
124	7C	SM120NCA	4	バイナリー	インターバル中にアクティブであった通信セッションの数
128	80	SM120NLS	4	バイナリー	インターバルの終了時に存在しているローカル通信セッションの数
132	84	SM120NLA	4	バイナリー	インターバル中にサーバー・インスタンス内で付加され、アクティブであったアクティブ・ローカル通信セッションの数
136	88	SM120NRS	4	バイナリー	インターバルの終了時に存在している遠隔通信セッションの数
140	8C	SM120NRA	4	バイナリー	インターバル中にサーバー・インスタンス内で付加され、アクティブであったアクティブ遠隔通信セッションの数
144	90	SM120BTS	4	バイナリー	すべての接続クライアントからサーバーに転送されたバイト数
148	94	SM120BFS	4	バイナリー	サーバーからすべての接続クライアントに送信されたバイト数
152	98	SM120BTL	4	バイナリー	すべてのローカル接続クライアントからサーバーに転送されたバイト数
156	9C	SM120BFL	4	バイナリー	サーバーからすべてのローカル接続クライアントに転送されたバイト数
160	A0	SM120BTR	4	バイナリー	すべてのリモート接続クライアントからサーバーに転送されたバイト数
164	A4	SM120BFR	4	バイナリー	サーバーからすべてのリモート接続クライアントに転送されたバイト数

サブタイプ 4: コンテナー・インターバル・レコード

1. **コンテナー・インターバル・セクション** (各レコードにつき 1 セクション)
   指定されたインターバルに 1 つのコンテナー内で発生した各アクティビティーに関する情報が含まれます。
2. **クラス・セクション** (各レコードにつき複数のセクション)

指定されたインターバルにこのアクティビティーに関与したすべてのクラスに関する情報が含まれます。

3. **メソッド・セクション** (各クラス・セクションにつき複数のセクション)

   指定されたインターバルにこのアクティビティーに関与した、すべてのクラスのすべてのメソッドに関する情報が含まれます。

コンテナー・インターバル・セクション

オフセット	オフセット	名前	長さ	フォーマット	記述
0	0	SM120HNM	64	EBCDIC	WebSphere トランザクション・サーバー・ホスト名
64	40	SM120SNM	8	EBCDIC	WebSphere トランザクション・サーバー名
72	48	SM120SIN	8	EBCDIC	WebSphere トランザクション・サーバー・インスタンス名
80	50	SM120CNM	256	EBCDIC	WebSphere コンテナー名
336	150	SM120CTP	4	バイナリー	コンテナー・トランザクション・ポリシー。値は次のとおりです。 • 1: 必要なトランザクション • 2: 同一サーバーのハイブリッド・グローバル • 3: ハイブリッド・グローバル • 4: 同一サーバーのハイブリッド・グローバルをサポート

340	154	SM120CSP	4	バイナリー	コンテナー・セキュリティー・ポリシー • 00001: DCE • 00010: ユーザー ID パスワード • 00100: ユーザー ID パスチケット • 01000: SSL • 10000: 非認証クライアント
344	158	SM120SST	16	S390STCK	サンプルがサーバーで開始された時刻。
360	168	SM120SET	16	S390STCK	サンプルが終了した時刻。

クラス・セクション: (サブタイプ 2: クラス・セクションを参照)

メソッド・セクション: (サブタイプ 2: メソッド・セクションを参照)

トリプレットおよび SMF レコードの分割

トリプレット

トリプレットを使用して、各種のデータ・セクション、およびこれらの各セクションの可変数を含む自己記述 SMF レコードを作成できます。すべてのデータ・セクションは以下のものから成るトリプレットによって記述されます。

1. データの開始位置を指定するオフセット
2. セクションの長さを示す長さ
3. このレコードに含まれたセクションのインスタンス数を示すカウント

製品セクションおよび汎用レコード情報セクション (たとえば、コンテナー・アクティビティー・レコード内のコンテナー自体を記述するセクション) を記述する 2 つのトリプレットは、レコード内の固定位置にあります。これにより、レコード・ヘッダーを評価した直後にレコードの評価を開始することができます。

SMF レコードの分割

WebSphere Application Server SMF レコードの大部分は可変長データ構造の記述に使用されるため (たとえば、コンテナーごとのクラスやクラスごとのメソッドが多数存在する場合があります)、SMF レコードは SMF によってサポートされる最大レコード・サイズ (32KB) より大きくなる場合があります。この場合は、論理レコードをいくつかの物理レコードに分割する必要があります。

これらの各物理レコードは、自己記述および自己完結型であることが必要です。*自己記述* とは、すでにトリプレットに関する段落で説明した内容を指します。つまり、レコードの読み取りに役立つ完全な機械構造のことです。*自己完結型* とは、協調して元の論理レコードを記述する物理レコードのサブセットが 1 つのみの場合でも、これらのレコードを評価し、レコードに保管されている情報を結合し、「不完全」のフラグを立てる必要があることを示します。これを行う必要があるのは、論理レコードを物理レコードに分割し、それらを順次に SMF に書き込むときに、SMF が最初のいくつかの物理レコードのみが 1 次 SMF ダンプ・データ・セットに適合すると判断し、それによって残りの物理レコードが代替 SMF ダンプ・データ・セットに書き込まれる可能性があるためです。フォーマット済み SMF ダンプ・データ・セットが評価される時点では、1 つの論理レコードを構成するすべての物理レコードが存在していることを想定しない場合があります。たとえば、物理コンテナー・アクティビティー・レコードの自己完結性とは、レコードにコンテナーの記述が含まれていることを意味しますが、必ずしもそのすべてのクラスが含まれている必要はありません。

ここでは、現在 RMF 製品で使用されている分割メカニズムと同様のメカニズムを使用します。コンテナー・レコードの場合は (サブタイプ 2 および 4)、レコードがクラス境界で分割されることを想定することはできませんが、1 つのクラスに属するメソッドを複数の物理レコードに対しても分割する必要がある場合を考慮する必要があります。以下の図を参照してください。

注: 以下の図中のセクションの長さを示す数は、説明の目的でのみ使用されています。特に、32K 境界、またはレコードの全長を示す矢印は、恣意的に配置されたものです。この図によって指示された数よりも多くのクラスおよびメソッドを物理レコードに適合させることができます。

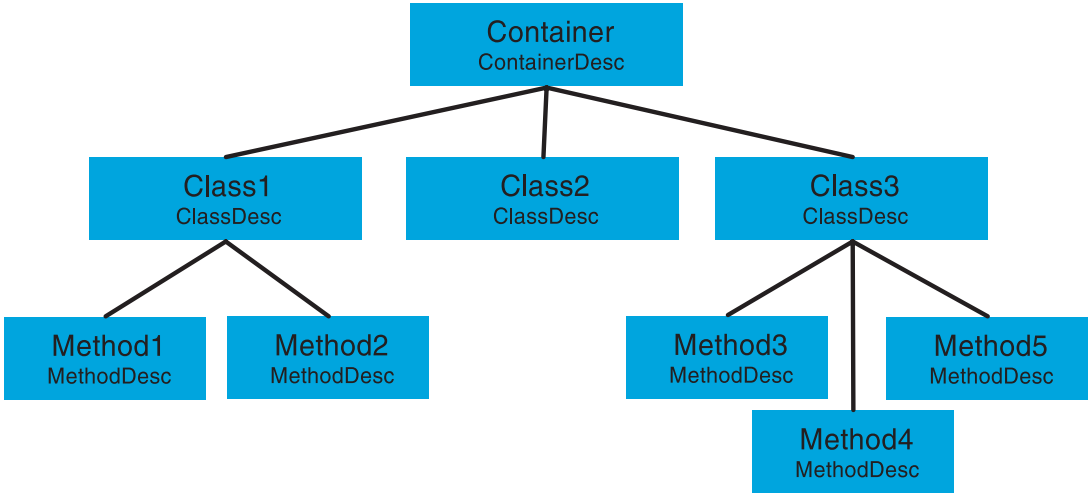

図 3. SMF レコード: 論理レコードおよび分割メカニズム

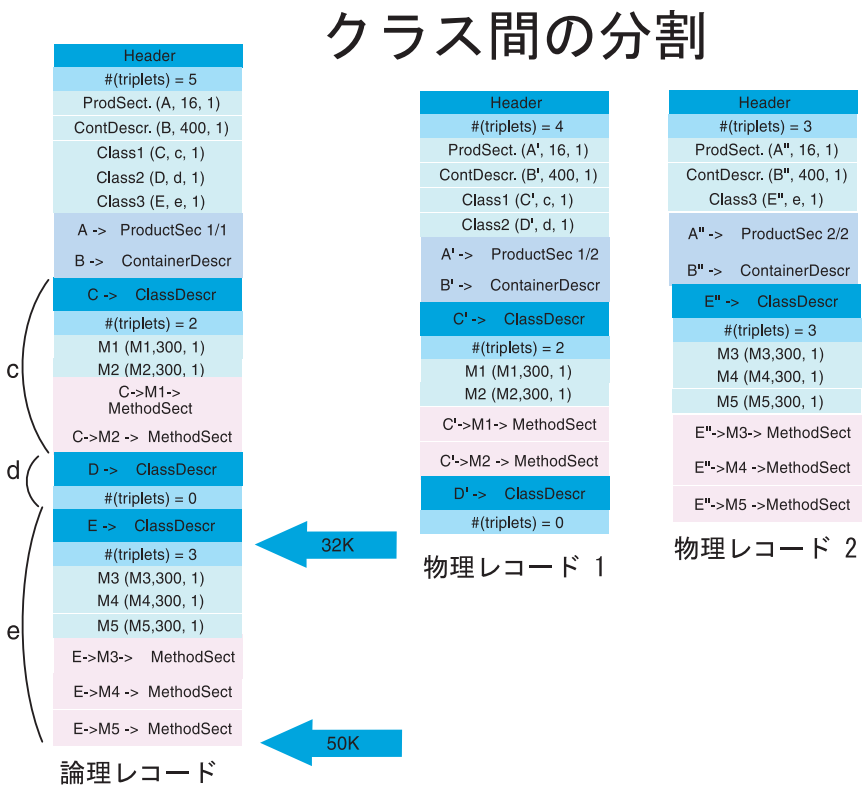

図 4. SMF レコード: クラス間の分割

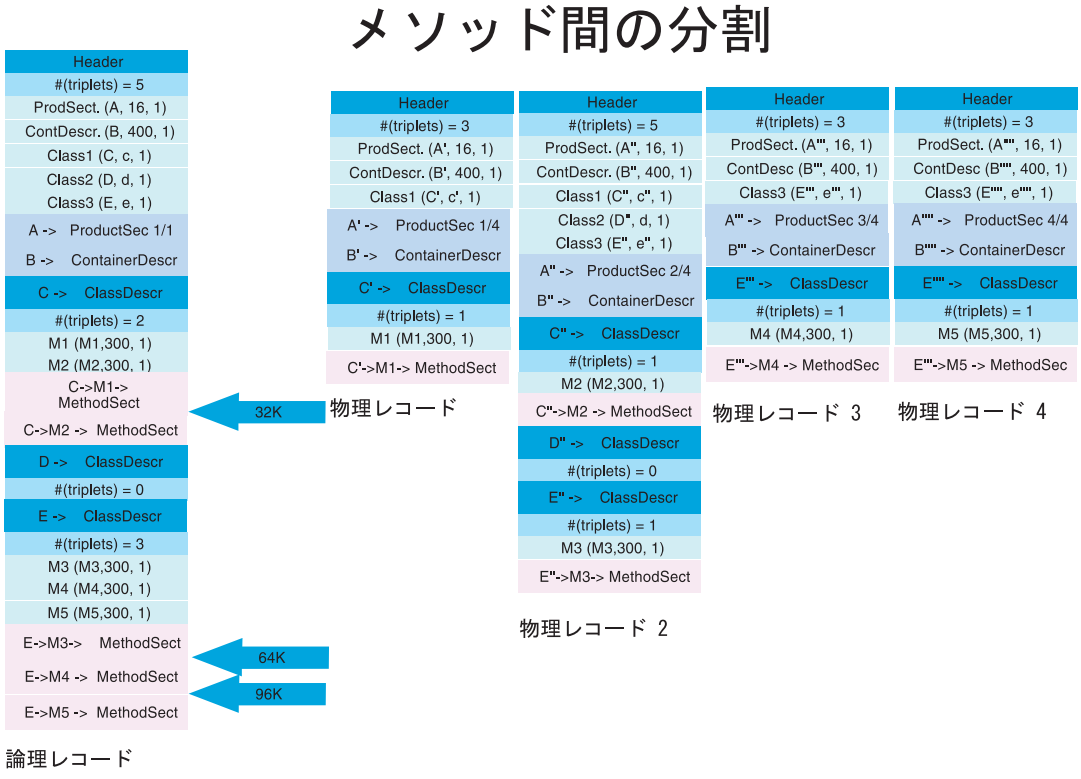

図 5. SMF レコード: メソッド間の分割

付録B. アプリケーション・サーバーのネーミング規則

本章では、アプリケーション・サーバーでの確かなネーミング規則を確立するためのガイドラインを示します。

アプリケーション・サーバーのネーミング規則が必要な理由

アプリケーション・サーバーでネーミング規則の確立を必要とする理由は以下のように数多くあります。

1. **WebSphere Application Server サーバーが IMS 領域または CICS 領域に類似しているため**
   - 制御領域およびサーバー領域のための調整済み手順がある。
   - サーバーの各インスタンスのための調整済み環境変数がある。
   - サーバーの各インスタンスのための環境変数がある。
   - 自己完結型の場合と、他のサーバーへの依存型の場合がある。
2. **セキュリティーのため**
   - 領域には、それに関連付けられたユーザー ID が割り当てられる。
   - ユーザーはサーバーおよびその中のオブジェクトへのアクセスが許可される。
3. **ワークロード・マネージャー (WLM) のため**
   - 領域の分類と領域内での作業。
   - アプリケーション環境。

WebSphere for z/OS アプリケーション・サーバーは、いくつかのアドレス・スペースで構成されます。これらのアドレス・スペースのために、インストール・システムは構成ファイル、セキュリティー・プロファイル、ワークロード種別構成などを管理する必要があります。アプリケーション・サーバーを作成、管理、および認識するためには、サーバーおよびサーバー・インスタンスをスタンプ・アウトするテンプレートが必要です。テンプレートは、以下のものに適用する必要があります。

- **サーバー名**
  - 制御領域の PROC 名
  - サーバー領域の PROC 名
  - アプリケーション環境名

  - インスタンス名
- **セキュリティー**
  - ユーザー / グループ /UID/GID
  - 制御領域
  - サーバー領域
  - インスタンス名
- **手順**
  - 環境ファイル
  - ライブラリー名
- **その他**
  - DB2 のコレクション名およびパッケージ名
  - ログ・ストリーム名

4 文字 (XXXX で示します) のアプリケーション・ネーミング方式に基づいてサーバーを作成するシステムがあります。1 つのサーバーの複数のインスタンスが WebSphere for z/OS 環境内の 1 つ以上のシステムに存在する場合があるため、サーバーを識別するための要件定義もあります。以下のようなシステムを使用することができます。

すべて 4 文字 (XXXX (および Y)) で判別されます。

Component Broker サーバー 名 = CBXXXX
- APPLENV 名 = CBXXXX
Component Broker サーバーのインスタンス名 = CBXXXXAY
- ORBsrvname のデフォルト値 = CBXXXXAY
制御領域のユーザー ID = CBXXXXC
- 制御領域の PROC = CBXXXXC
制御領域のグループ ID = CBXXXXG
サーバー領域のユーザー ID = CBXXXXS
- サーバー領域の PROC = CBXXXXS
サーバー領域のグループ ID = CBXXXX
デフォルト・リモート・ユーザー ID = CBXXXXI
デフォルト・ローカル・ユーザー ID = CBXXXXD
デフォルト ID のグループ ID = CBXXXXP

以下はユーザー ID です。必要に応じて変更してください。

CBXXXXC 0 - 変更しないでください。
CBXXXXS 1100
CBXXXXD 1101
CBXXXXI 1102

以下はグループ /GID です。必要に応じて変更してください。

CBXXXXG 1000
CBXXXXR 1001
CBXXXXP 1002

ネーミング規則は以下のものにも適用されます。

サーバー特定ログ・ストリーム = CBXXXX.ERROR.LOG
LRM、 = CBXXXX_LRM_DB2
LRMI、 = CBXXXXAY_LRMI_DB2
DB2 コレクション = CBXXXX_PK
HFS ファイル・システム名 = /WSCapps/CBXXXX/bin および
/WSapps/CBXXXX/lib
OS ファイル名 = hlq.CBXXXX.LOADLIB
hlq.CBXXXX.HFS
hlq.CBXXXXAY.PARMS
など

ネーミング方式の伏字の部分は、RACF ID に関連付けられた UID/GID が管理します。これらのエンティティーをユーザー ID に自動的に割り当てたり関連づけたりする平易なメカニズムはありません。

たとえば、以下に、アプリケーション・サーバー APP1 と関連付けられた制御領域およびサーバー領域の PROC を定義する 1 つの方法を示します。各サーバー・インスタンスには、環境設定を含むその固有のデータ・セットがあることに注意してください。この方式は、さまざまなメンバーを指定するシスプレックス全体に対して 1 つの PDS があるように簡単に変更することができます。重要な制限として、記号パラメーターの変更をサーバー領域に渡す最小限の機能であることは認識しておいてください。

また、データ・セット名は、そのデータ・セットがサーバーに固有のものか、あるいはシスプレックス内で共通のものかを示すことにも注意してください。このネーミング方式では、2 次レベル修飾子は、データ・セットが使用されるかどうかを示しています。

- シスプレックス規模

- 特定システムで実行されるサーバーのみ
- サーバー規模
- 指定のサーバー・インスタンスのみ

制御領域の PROC

```
//BBOASR1 PROC SRVNAME='BBOASR1A',
//       PARMS='',
//      CBCONFIG='/WebSphere390/CB390'
//* See instructions at the bottom of this file
//  SET BBOLIB='BBO'
//  SET LELIB='CEE'
//  SET DB2='DB2'
//  SET RELPATH='controlinfo/envfile'
//BBOASR1 EXEC PGM=BBOCTL,REGION=0M,
// PARM='/ -ORBsrvname &amp;SRVNAME &amp;PARMS'
//*STEPLIB  DD DSN=&BBOLIB..SBBOLD2,DISP=SHR
//*         DD DSN=&BBOLIB..SBBOLOAD,DISP=SHR
//*         DD DSN=&LELIB..SCEERUN,DISP=SHR
//*         DD DSN=&DB2..SDSNLOAD,DISP=SHR
//BBOENV    DD PATH='&CBCONFIG/&RELPATH/&SYSPLEX/&SRVNAME/current.env'
//CEEDUMP   DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSOUT    DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSPRINT  DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
```

サーバー領域の PROC

```
//BBOASR1S PROC IWMSSNM='BBOASR1A',PARMS='-ORBsrvname ',
//      CBCONFIG='/WebSphere390/CB390'
//* See instructions at the bottom of this file
//  SET BBOLIB='BBO'
//  SET LELIB='CEE'
//  SET DB2='DB2'
//  SET RELPATH='controlinfo/envfile'
//BBOASR1S EXEC PGM=BBOSR,REGION=0M,TIME=NOLIMIT,
// PARM='/ &PARMS &IWMSSNM'
//STEPLIB  DD DSN=&BBOLIB..SBBOULIB,DISP=SHR
//*         DD DSN=&BBOLIB..SBBOLD2,DISP=SHR
//*         DD DSN=&BBOLIB..SBBOLOAD,DISP=SHR
//*         DD DSN=&LELIB..SCEERUN,DISP=SHR
//*         DD DSN=&DB2..SDSNLOAD,DISP=SHR
//BBOENV   DD PATH='&CBCONFIG/&RELPATH/&SYSPLEX/&IWMSSNM/current.env'
//CEEDUMP  DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

付録C. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、米国以外の国においては本書で述べる製品、サービス、またはプログラムを提供しない場合があります。しかし、このことは、弊社がこのような IBM 製品、プログラミングまたはサービスを、日本で発表する意図があることを必ずしも示すものではありません。本書で IBM ライセンス・プログラムまたは他の IBM 製品に言及している部分があっても、このことは当該プログラムまたは製品のみが使用可能であることを意味するものではありません。これらのプログラムまたは製品に代えて、IBM の知的所有権を侵害することのない機能的に同等のプログラムまたは製品を使用することができます。ただし、IBM によって明示的に指定されたものを除き、これらのプログラムまたは製品に関連する操作の評価および検査はお客様の責任で行っていただきます。

IBM は、本書で解説されている主題について特許権 (特許出願を含む)、商標権、または著作権を所有している場合があります。本書の提供は、これらの特許権、商標権、および著作権について、本書で明示されている場合を除き、実施権、使用権等を許諾することを意味するものではありません。実施権、使用権等の許諾については、下記の宛先に、書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31
AP 事業所
IBM World Trade Asia Corporation
Intellectual Property Law & Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。
IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

本書に対して、周期的に変更が行われ、これらの変更は、文書の次版に組み込まれます。 IBM は、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するもので

はありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。また、IBM 以外の製品に関するパフォーマンスの正確性、互換性、またはその他の要求は確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があり、単に目標を示しているものです。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書で使用されている例

本書で使用される例は IBM Corporation が作成したサンプルに過ぎません。これらの例は、標準的な製品や IBM 製品の一部ではなく、単にユーザーのアプリケーション開発を支援する目的で提供されています。例は、「現状のまま」で提供されます。IBM は、これらの例の機能およびパフォーマンスに関して、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。IBM は、これらの例を使用することによって生じたいかなる損害に対しても、たとえ、そのような損害の可能性を通知している場合でも、法律上の責任は負いません。

これらの例は、上記の免責条項をそのまま適用することを条件として、配布し、コピーし、改変し、他のソフトウェアに取り込むことができます。

プログラミング・インターフェース情報

本書には、WebSphere for z/OS のプログラミング・インターフェースとして使用することを意図していない情報が含まれています。

商標

以下は、IBM Corporation の米国およびその他の国における商標または登録商標です。

CICS
DB2
IBM
IMS
IMS/ESA
MVS
OS/390
RACF
RAMAC
RMF
SecureWay
S/390
VTAM
WebSphere
z/OS

Lotus、Notes、Domino、および Lotus Go Webserver は、Lotus Development Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標および登録商標です。

Microsoft、ActiveX、Visual Basic、Visual C++、Windows、Windows NT、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標または登録商標です。

その他の会社名、製品名、サービス名は、各社の商標または登録商標です。

用語集

本書で使用している用語については、次の用語集を参照してください。

- *WebSphere Application Server エンタープライズ版 Component Broker 用語集*, SD88-7380。次の URL からご覧になれます。

  `http://www.ibm.com/jp/software/websphere/appserv/library.html`

- Sun Microsystems Glossary of Java Technology-Related Terms。次の URL からご覧になれます。

  `http://java.sun.com/docs/glossary.html`

探している用語が見つからない場合は、*IBM Glossary of Computing Terms* を調べてください。次の URL からご覧になれます。

`http://www.ibm.com/ibm/terminology/`

また、次の Sun Microsystems の Web サイトもご覧ください。

`http://www.sun.com/`

プログラム番号: 5655-F31

Printed in Japan

SA88-8653-00

日本アイ・ビー・エム株式会社
〒106-8711　東京都港区六本木3-2-12