

CICS Transaction Server for z/
OSバージョン 5 リリース 6

CICSplex SM 概念および計画



注記

本書および本書で紹介する製品をご使用になる前に、[製品の特記事項](#)に記載されている情報をお読みください。

本書は、IBM® CICS® Transaction Server for z/OS®, バージョン 5 リリース 6 (製品番号 5655-Y305655-BTA)、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典：

CICS Transaction Server for z/OS
Version 5 Release 5
CICSplex SM Concepts and Planning

発行：

日本アイ・ビー・エム株式会社

担当：

トランスレーション・サービス・センター

© Copyright International Business Machines Corporation 1974, 2020.

目次

本書について.....	v
第 1 章 CICSplex SM の概要.....	1
CICSplex SM の機能.....	1
CICSplex SM 環境.....	4
CICSplex.....	6
管理下アプリケーション・システム (MAS).....	7
CICSplex SM アドレス・スペース (CMAS).....	8
環境サービス・システム・サービス (ESSS).....	8
CICSplex SM オブジェクト.....	9
データ・リポジトリ.....	10
CICS 管理クライアント・インターフェース (CMCI).....	10
第 2 章 CICSplex SM 環境の設計.....	25
CICSplex の設計.....	26
CICS システムおよび領域の特定.....	26
企業システムに存在する CICSplex の数の決定.....	28
システム・グループの指定.....	31
CMAS の配置.....	32
CMAS をインストールする場所.....	32
CMAS 間リンク.....	33
保守ポイント CMAS の配置.....	37
WUI サーバーの場所についての計画.....	37
CICSplex SM エンティティの命名.....	39
段階的な実装.....	41
第 3 章 CICSplex SM の計画のセットアップ.....	43
CICSplex SM のセキュリティ計画.....	43
時間帯の定義.....	44
CICSplex SM オブジェクト定義の再利用.....	45
第 4 章 CICSplex SM 構成とトポロジーの定義.....	47
CMAS 構成の定義.....	47
CMAS 構成の管理.....	47
CICSplex トポロジーの定義.....	48
CICSplex 定義の準備.....	48
CICS システム定義の計画.....	48
次に行うこと.....	51
第 5 章 CICSplex SM によるリソース管理.....	53
ビジネス・アプリケーション・サービス (BAS) によるリソースの管理.....	53
ビジネス・アプリケーション・サービス (BAS) とは.....	53
BAS の管理.....	58
リソース定義のマイグレーション.....	62
リソースの定義.....	63
マイグレーション形式 BAS の使用.....	64
全機能形式 BAS の使用.....	64
アプリケーションとリソースのインストール.....	65
セキュリティに関する考慮事項.....	65
BAS の計画.....	66

BAS の実装.....	67
次に行うこと.....	68
第 6 章リアルタイム分析 (RTA) によるモニター.....	69
CICSplex SM リアルタイム分析とは.....	69
RTA 外部通知.....	70
システム使用可能性モニター (SAM).....	70
MAS リソース・モニター (MRM).....	71
分析点モニター (APM).....	72
RTA の管理.....	73
RTA の計画.....	76
RTA の実装.....	76
システム使用可能性モニター (SAM) の実装.....	76
MAS リソース・モニター (MRM) の実装.....	77
分析点モニター (APM) の実装.....	77
推奨されている方法.....	77
第 7 章 CICSplex SM モニターによる統計の収集.....	79
CICSplex SM モニターとは.....	79
モニター機能の管理.....	80
CICSplex SM モニターの計画.....	81
CICSplex SM モニターの実装.....	83
リソース状況機能の活動化.....	83
CICSplex SM での CICS モニター機能 (CMF) の使用.....	84
推奨されている方法.....	84
次に行うこと.....	84
特記事項.....	85
索引.....	91

本書について

本書では、CICSplex SM の一般的な概要を紹介しています。企業で CICSplex SM の導入準備を担当するシステム設計者、計画担当者、およびプログラマーを対象としています。

以下のことについて説明します。

- CICSplex SM の動作の概要と、いくつかの新しい概念を紹介
- CICS システム管理ツールとしての CICSplex SM の主な機能と利点
- CICSplex SM 構成の設計のアプローチ方法
- CICSplex SM 環境の定義の始め方
- CICSplex SM のビジネス・アプリケーション・サービス、ワークロード管理、リアルタイム分析、およびモニター機能の使用に関する計画の方法
- CICSplex SM のインストールの準備方法

これらのタスクは、プログラム・ディレクトリー と関連して行われます。

使用されている用語や表記について詳しくは、IBM Knowledge Center の [CICS 資料で使用されている表記規則および用語](#)を参照してください。

本書の日付

本書は、2020 年 5 月 28 日に作成されました。

第 1 章 CICSplex SM の概要

CICS Transaction Server for z/OS の CICSplex® System Manager エLEMENT は、複数の CICS システムを単一の制御点から管理できるようにするためのシステム管理ツールです。

企業において CICSplex SM が必要になる可能性があるのは、10 ないしは 15 から 200 ないしは 300 (またはそれ以上) の範囲の CICS 領域を実行する場合です。最近の z/OS シスプレックス環境では、トランザクション処理ワークロードをサポートするためにそのくらい多くの CICS 領域があるのはしだいに一般的になっています。

CICSplex SM には、次のような主な特徴があります。

CICSplex SM は、システム管理において IBM ストラテジーに準拠します。

- システム管理データの作成と管理の一貫性が保たれます。
- 企業全体で使用されるシステム管理アプリケーションです。
- システム管理タスクをさらに自動化できます。
- マネージャーとエージェントのコンポーネントに構造化されており、全社的情報ベース、調整情報ベースがあり、データ・モデルに基づいています。

CICSplex SM には、多くの信頼性、可用性、および保守容易性 (RAS) 機能があります。

以下に例を挙げます。

- ユーザー・アプリケーションの CICS 領域は、離散的制御領域 (関連端末がないか、あるいはアプリケーション・ELEMENT またはアプリケーション・コンポーネントを使用する) により管理されます。この管理領域には複数のインスタンスが存在可能です。
- CICSplex SM 機能へのユーザー・アクセスは、別個のアドレス・スペースを使用して提供されます。アドレス・スペースにも、1 つの企業において複数のインスタンスが存在可能です。
- システム管理データは、管理側と被管理側のいずれかの CICS システムでの操作によって影響を受けないように、データ・スペース・キャッシュで別個に保持されます。

CICSplex SM には機密保護機能があります。

CICSplex SM は、企業のすべての CICS リソースに単一セッションからアクセスすることを許可する可能性がありますとはいえ、セキュリティが危険にさらされることはありません。SAF 準拠の外部セキュリティ・マネージャーを使用して、次の場所への無許可アクセスを防止できます。

- CICSplex SM プロシージャおよびデータ
- CICSplex SM システム管理機能およびそれらが管理する CICS リソース

通常は CICS によって実行されるセキュリティ検査 (つまり CICS コマンド検査および CICS リソース検査) は、代わりに CICSplex SM によって実行されます。

CICSplex SM の機能

CICSplex SM は、リアルタイムの単一システム・イメージ、単一制御ポイント、ビジネス・アプリケーション管理、CICSplex 全体の操作などの多くのシステム管理機能を組み込んでいます。

• リアルタイムの単一システム・イメージ (SSI)

CICSplex SM は、企業のトランザクション処理環境を構成するすべての CICS 領域およびリソースについてのリアルタイムの単一システム・イメージ (SSI) を提供します。CICSplex SM は、CICS システムとリソースのインベントリを作成し、それらの関係をマップします。このプロセスは、CICSplex SM トポロジーと呼ばれます。オペレーターは、リソースの場所が分からなくてもリソースを処理できるようになりました。SSI は、CICSplex SM 操作機能によって提供され、操作、モニター、リアルタイム分析 (RTA)、およびワークロード管理 (WLM) 用のすべての CICSplex SM アプリケーションに適用されます。

• 単一制御ポイント

CICSplex SM オペレーターは、企業の CICS システムおよびそれらのリソースすべてを、単一セッションから管理できます。つまり、CICSplex SM によって企業に単一制御ポイントを提供できるため、

CICSplex SM オペレーターは、大規模で複雑な CICS リソースの構成を管理できます。さらに、複数の単一制御ポイントを指定することもできます。これにより、複数の場所の複数の CICS オペレーターがそれぞれ企業の CICS システムおよびリソースの全体像を表示することもできますし、個々の要件に合わせてそれらのシステムおよびリソースの一部を表示することもできます。最後の点として、これらの制御ポイントの物理的位置は無関係であるため、制御場所の指定は非常に柔軟に行えます。

• ビジネス・アプリケーションの管理

CICSplex SM を使用すると、CICS システムの観点からではなく、ご使用のビジネス・アプリケーションの観点で企業を管理できるようになります。**ビジネス・アプリケーション・サービス (BAS)** を使用すると、以下の機能が提供されるため、リソースをアプリケーション・レベルで管理できます。

- リソースの単一定義ポイント。CICSplex のすべてのリソース、およびそれらの間の関係は、1つの場所である CICSplex SM データ・リポジトリに保持され、CICSplex の単一システム・イメージが得られます。CICSplex SM は、単一の定義からリソースのローカル・インスタンスおよびリモート・インスタンスを生成します。
- 論理スコープ。リソースを、CICS システムにおける物理的位置によってではなく、リソースのビジネス相互関係に従ってリンクおよび管理できます。
- これらのリソースのインストール。CICS システム開始時に自動的に行うこともできますし、必要に応じて実行中の CICSplex に動的に行うこともできます。

BAS は、オンライン・リソース定義 (RDO) の代わりになります。RDO は定義が単一のグループに結合される構造になっていて、グループはグループ・リストに従って順番に処理されます。BAS では、唯一のグループとの関連というリソース定義の制限がなくなります。リソース定義を再利用して、必要に応じて複数のグループに関連付けることができます。BAS では、リソースを、企業内での使用方法に応じて関連付けることができます。グループではなく、個々のリソースの管理が可能です。例えば、給与計算システムに関連付けられたすべてのファイルを、CICSplex 内の他のファイルに影響を与えることなく使用不可に設定できます。

[ビジネス・アプリケーション・サービス \(BAS\) によるリソースの管理を参照してください。](#)

• CICSplex 全体の操作

任意の制御ポイントから、オペレーターは SSI を利用して CICSplex 全体または選択したサブセット全体に対する操作を実行できます。企業内にある CICS リソースの 1 つ以上のインスタンスの状況について情報を表示することができます。さらに、単一のエントリーでリソースの状況を変更することもできます。

Web ユーザー・インターフェースのアクション・ボタンを使用すると、表示されたリソースの状況に影響があります。応答は、操作ビュー というパネルに表示されます。ここでは、関連した情報を要約でき、リンクを使用して他の関連情報にアクセスできます。操作ビューには、現在で CICS システムに提供されている機能が反映されます。そのためオペレーターは、普段システム・アクティビティーを行うときの基本的なアプローチを変更せずに、今までと基本的に同じ方法で作業できます。

操作ビューに関する詳細については、『[CICS 操作ビュー](#)』および『[CICSplex SM Web ユーザー・インターフェース](#)』を参照してください。

• ワークロードの管理

CICSplex SM ワークロード管理 (WLM) では、動的ルーティング・プログラム EYU9XLOP を使用して、適格となる作業要求を、要求領域から、トランザクション開始時に選択された適切なターゲット領域にルーティングします。

EYU9XLOP は、次の動的ルーティングを処理します。

- 端末で開始されたトランザクション
- 端末に関連付けられている適格な EXEC CICS START 要求
- 端末に関連付けられていない適格な EXEC CICS START 要求
- 以下を使用して受信する動的プログラム・リンク (DPL) 要求
 - CICS Web インターフェース
 - CICS Gateway for Java™
 - 外部 CICS インターフェース (EXCI) クライアント・プログラム

- 外部呼び出しインターフェース (ECI) を使用する CICS クライアント・ワークステーション製品
- オープン・ネットワーク・コンピューティング (ONC) RPC
- EXEC CICS LINK PROGRAM 要求を発行する関数
- CICS Business Transaction Services (BTS) のアクティビティーに関連付けられたトランザクション

CICSplex SM は、BTS セット全体で BTS 作業を動的にルーティングすることにより、さらに分散環境におけるデータの管理をモニターすることにより、BTS の管理サポートを行います。詳細については、[BTS の概要](#)を参照してください。

BTS アクティビティーのルーティングを行うために CICSplex SM ワークロード管理を使用する必要はありませんが、CICSplex SM を使用すると次のような多くの利点があります。

- 分散データの管理
- ワークロード管理のワークロード分離機能とワークロード・ルーティング機能
- ユーザーが置き換え可能な動的ルーティング・プログラム EYU9XLOP
- BAS を併用する場合のリンク定義の縮約
- ビジネス環境の管理における BAS と BTS の間の協働

自分でプログラムを作成して EYU9XLOP を置換し、それをユーザーが置き換え可能な CICS 提供プログラムである DFHDYP および DFHDSRP と一緒に使用することによって、動的ルーティング要件を処理できます。

動的ルーティングおよびワークロード管理について詳しくは、[CICSplex SM によるワークロードの管理](#)を参照してください。

• CICS リソースの自動化された例外報告

CICSplex SM リアルタイム分析 (RTA) 機能により、対象とした条件に関する外部通知が自動的に提供されます。通知は、コンソール・メッセージの形式と NetView® への総称アラートの形式の一方または両方で行われます。リアルタイム分析は、一般的に認識されているエラー状態のみを対象としているものではありません。リソースの状況のどの特徴についても通知を受けるようにすることができます。リアルタイム分析を使用すると、別個の自動化製品を使用しなくてもアクションを実行することができます。

CICSplex SM の RTA 機能については、[リアルタイム分析 \(RTA\) によるモニター](#)でさらに詳しく説明しています。詳細については、『モニター』の『リアルタイム分析』を参照してください。

• CICS リソースで統計データを収集するときのモニター機能

CICSplex SM モニター 機能により、一連の CICS システム内の指定のリソース・インスタンスに関するパフォーマンス関連データを、ユーザー定義の間隔で収集できます。

CICSplex SM のモニター機能については、[CICSplex SM モニター](#)を使用する統計の収集でさらに詳しく説明しています。詳細については、『モニター』の『リアルタイム分析』を参照してください。

• アプリケーション・プログラミング・インターフェース (API)

CICSplex SM は、アプリケーションで以下の処理を可能にするアプリケーション・プログラミング・インターフェース (API) を備えています。

- CICS リソースおよび CICSplex SM リソースに関する情報へのアクセス。
- CICSplex SM のサービスの呼び出し。

以下の言語で作成されたプログラムでは、コマンド・レベル・インターフェースを使用できます。

- アセンブリ
- PL/I
- COBOL
- C

さらに、REXX ランタイム・インターフェースを使用できます。

CICSplex SM API を使用すると、CICSplex SM と CICS リソース定義の管理を自動化する外部プログラムを作成できます。そのようなプログラムを使用すると、CICSplex SM のシステム管理機能を企業全体の

変更管理プロセスに統合できます。例えば、リソース定義の変更を、データベースまたはファイルの更新情報を使って、またはアプリケーションの標準的なライフ・サイクルを使って調整する API プログラムを作成できます。API 全体の説明については、「[CICSplex SM コマンドの概要](#)」を参照してください。

- **CICSplex SM 環境の管理**

CICSplex SM 環境の管理は、以下を使用して行います。

- **CICSplex SM オブジェクト**

CICS システムの構成を CICSplex SM に定義するには (さらに BAS、WLM、RTA、およびモニター要件を定義するには)、CICSplex SM オブジェクト を作成し、それらを互に関連付けます。オブジェクトごと、およびそれらの間の関連またはリンクごとに、CICSplex SM データ・リポジトリ内にレコードが作成されます。CICSplex SM オブジェクトについては、[9 ページの『CICSplex SM オブジェクト』](#)で説明しています。

- **データ・リポジトリ**

データ・リポジトリには、CICSplex SM コンポーネント、リソース、システム管理要件、およびそれらの間の関係を定義するオブジェクトが入っています。これらの定義は、Web ユーザー・インターフェース、CICSplex SM API、またはバッチ処理されるリポジトリ更新機能を使用して作成できます。

- **バッチ処理されるリポジトリ更新機能**

バッチ処理されるリポジトリ更新機能を使用すると、他の定義のテンプレートとして使用される 1 つのコマンドを実行依頼することにより、多数の CICSplex SM および CICS リソース定義を作成したり更新したりすることができます。バッチ処理されるリポジトリ更新機能は、あるプラットフォームから別のプラットフォームに定義をマイグレーションする場合、およびデータ・リポジトリをバックアップする場合にも使用されます。詳しくは、[CICSplex SM の管理](#)を参照してください。

- **時間依存アクティビティの管理**

CICSplex SM のアクティビティの多くは時間に依存しています。例えば、定義をアクティブにする時間は、RTA およびモニター定義で指定できます。また、同じ CICSplex 内であっても異なる時間帯で稼働する CICS システムを、同じ時間帯にあるかのように稼働させたい場合もあります。その場合、以下を制御する時間枠定義を作成できます。

- ローカル時間帯とは無関係の、企業システムの任意の部分が操作可能になる厳密な時刻
 - 特定のシステム管理機能を操作可能にする時刻

グリニッジ標準時 (GMT) に基づく、時間帯における国際標準が使用されます。ユーザーは自分の CICSplex を実行する時間帯を選択します。その後、GMT と同じ時間を導入している場所について、60 分の倍数で調整したり、夏時間調整を行ったりできます。

時間枠定義の詳細については、「[CICSplex SM の管理](#)」に記載されています。

CICSplex SM 環境

CICSplex SM 環境の定義を始める前に、CICSplex SM のコンポーネントを理解する必要があります。その理由は、それらのコンポーネントを定義する必要があるため、またはそれらが環境の設計に影響があるためです。

CICSplex とは

CICSplex とは、複数の CICS システムを単一のエンティティであるかのように管理および操作するためにグループ化したものです。つまり、CICSplex SM は、単一システム・イメージ (SSI) を設定するこれら複数の CICS システムで構成された、管理ドメインです。

[続きを読む...](#)

MAS とは

CICSplex SM によって管理される各 CICS 領域は、管理下アプリケーション・システム (MAS) と呼ばれます。MAS は、CICSplex の一部として定義および管理します。CICSplex 内のそれぞれの MAS は、CICSplex SM アドレス・スペース (CMAS) によって管理されます。複数の CMAS で CICSplex 内の MAS を管理できますが、1 つの MAS は所定の時刻において 1 つの CMAS にしか接続できません。

[続きを読む...](#)

CMAS とは

それぞれの CICSplex には、保守ポイント として定義された 1 つの CICSplex SM アドレス・スペース (CMAS) があります。保守ポイント CMAS には、すべてのデータ・リポジトリに入っているオブジェクトのデータ保全性を維持する役割があります。これは、そのデータ・リポジトリを他の CMAS のデータ・リポジトリと同期化することによって行います。CMAS 保守ポイントでのこの同期は、CMAS 間リンクを使用して実行されます。このリンクは通常、CMAS 間での管理コマンドおよびデータのルーティングで使用されます。

[続きを読む...](#)

CMCI とは

CICS 管理クライアント・インターフェース (CMCI) は、システム管理アプリケーション・プログラミング・インターフェースで、IBM CICS Explorer などの HTTP クライアント・アプリケーションで使用するためのものです。このインターフェースを使用して、CICSplex の CICS 領域内にあるインストール済みおよび定義 CICS リソースおよび CICSplex SM リソースや、スタンドアロン CICS 領域にある CICS リソースを管理する HTTP クライアント・アプリケーションを開発できます。

[続きを読む...](#)

CICSplex SM の管理に使用できるツール

すべての CICSplex SM コンポーネント、リソース、システム管理要件、およびそれらの間の関係は、オブジェクト としてデータ・リポジトリ に保持されます。これらのオブジェクトは、提供されている 1 つ以上のインターフェースを使用して管理できます。

• CICS Explorer®

CICS Explorer は Eclipse ベースのツールで、データ・リポジトリでオブジェクトを作成、インストール、および管理するために使用できます。CICS 領域に接続するには、[CMCI または SM データ接続](#)が必要です。

[続きを読む...](#)

• CICSplex SM Web ユーザー・インターフェース (WUI)

CICSplex SM WUI は、データ・リポジトリでオブジェクトを作成、インストール、および管理するためのカスタマイズ可能な、プラットフォームに依存しない Web インターフェースを提供します。WUI を使用すれば、標準 Web ブラウザー・ソフトウェアを使用して CICSplex SM にアクセスし、CICS システムのリソースのモニターや制御、CICSplex SM 構成の定義および保守、BAS、WLM、RTA、およびモニター要件の指定などの運用タスクや管理タスクを実行できます。Web ブラウザー・クライアントは、CICS Web インターフェースを介して、HTTP 要求により WUI サーバーに接続します。

WUI サーバーは、CICSplex SM ローカル MAS として実行される専用 CICS 領域で、接続先の CMAS を介して管理対象リソースと通信します。複数の WUI サーバーをアクティブにすることができます。例えば、さまざまな言語を使用する必要がある場合や、さまざまなサーバーに対してさまざまなシステムを使用可能にする必要がある場合などがあります。

注: Web ユーザー・インターフェース・サーバーのコードは、これが実行される CICS 領域およびこれが接続される CMAS と同じリリース・レベルでなければなりません。

すべてのメニューおよびビュー定義は、サーバー・リポジトリに保管されます。WUI サーバーごとに 1 つのリポジトリがあります。メニューおよびビュー定義は、バックアップのため、他のサーバーに定義を配布するため、および製品の新規リリースにアップグレードする際にメニューおよびビューを転送するために、エクスポートすることができます。

[続きを読む...](#)

• バッチ処理されるリポジトリ更新機能

バッチ処理されるリポジトリ更新機能は、CICSplex SM リソース定義オブジェクトを作成するためのバッチ・ジョブを提供します。

バッチ処理されるリポジトリ更新機能は、CICS 定義の CICSplex SM へのマイグレーションに役立ちます。CICS DFHCSDUP ユーティリティの EXTRACT コマンドを使用して、CSD レコードを読み取ることができます。CICSplex SM は、バッチ処理されるリポジトリ更新機能への入力となる同等なリソース定義を DFHCSDUP 出力から生成する、出口ルーチン EYU9BCSD を提供します。

バッチ処理されるリポジトリ更新機能は、あるプラットフォームから別のプラットフォームに定義をマイグレーションする場合にも便利です。それを使用して既存のリソース定義を CICSplex SM データ・リポジトリから取り出してから、その定義に必要な変更を行った後、変更内容を、別のバッチ処理されるリポジトリ更新機能を実行して新しいプラットフォーム上にリソース定義を作成するときの入力とすることができます。

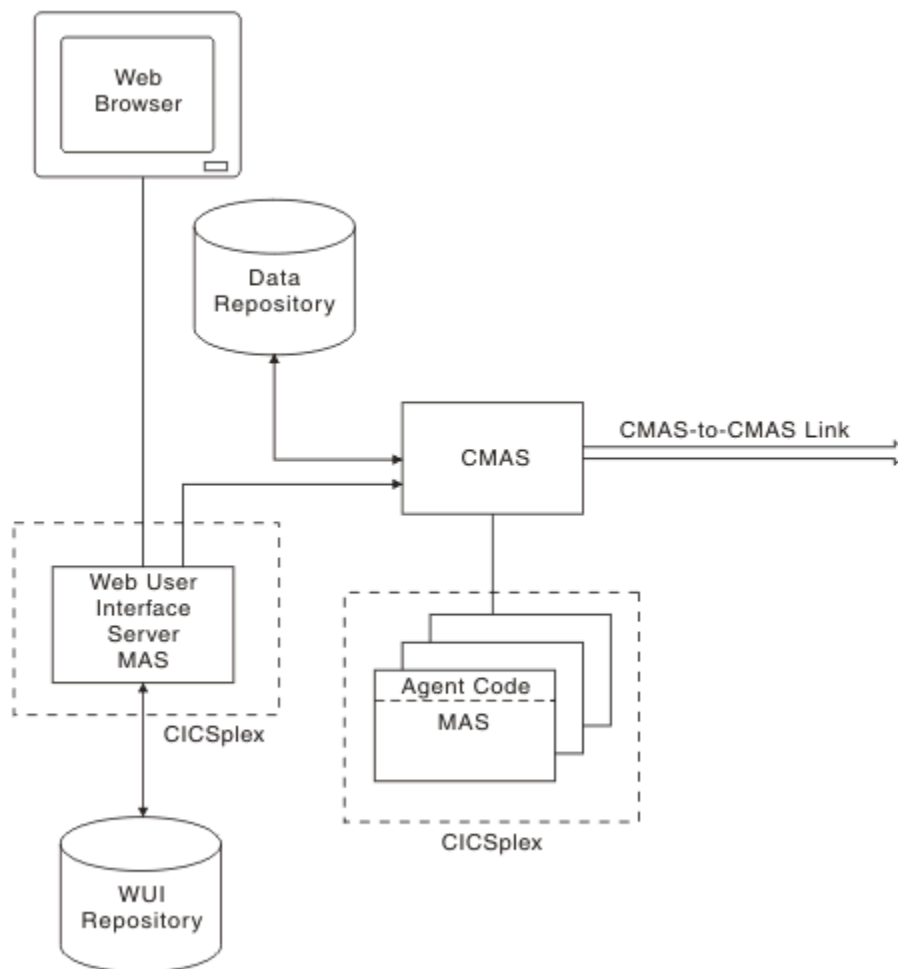


図 1. CICSplex SM 構成のキー・コンポーネント

関連情報

[CICSplex SM のセットアップ](#)

CICSplex

CICSplex SM で管理される CICSplex には、企業内のすべての CICS システムを含めることができます。あるいは、複数の CICSplex を定義し、それぞれに CICS システムの論理グループを含めることもできます。

例えば、CICSplex は、特定の MVS™ イメージ上のすべての CICS システムで構成されるようにもできますし、ユーザーのサブセットがアクセス可能なすべての CICS システム、または特定の地理的区域にサービスを提供するすべての CICS システムで構成されるようにもできます。さらに、CICSplex の構成は、基礎となる CICS システムの機能に影響を与えることなく変更できます。CICSplex SM で管理される単一の CICSplex に含まれる CICS システムは、管理のために明示的に互いに接続する必要はありません。

CICSplex SM によって管理される CICSplex に関する最も重要な点は、以下の点です。

- CICSplex は、処理できる最も大きい単位です。つまり、CICSplex はグループ化できず、そのようなグループを単一エンティティとして操作することはできません。
- CICSplex SM データをある CICSplex から別の CICSplex にコピーすることはできません。システム管理の理由で、CICSplex は他の CICSplex に対して「密封」されています。
- CICSplex は相互に排他的であるため、CICS システムが複数の CICSplex に属することはできません。

CICSplex SM を使用すると、CICS システム・グループと呼ばれる CICSplex のサブセットを定義できます。CICS システム・グループは相互に排他的ではなく、同じ CICS システムを参照できます。そのため、企業内のすべての CICS システムを単一の CICSplex に含める場合、その CICSplex 内の CICS システムのグループを単一システム・イメージとして管理するための仕組みがあります。

CICS システムおよび CICS システム・グループは、数の制限なく既存の 1 つの CICSplex に割り当てることができます。

1 つの CICS システムは 1 つの CICSplex にしか定義できませんが、1 つの CICS システムをその CICSplex 内の複数の CICS システム・グループに割り当てすることもできます。その CICS システム・グループを任意の数の他の CICS システム・グループに割り当てすることもできます。

管理下アプリケーション・システム (MAS)

CICSplex SM によって管理されている稼働中の各 CICS システムは、管理下アプリケーション・システム (MAS) と呼ばれます。

CICSplex SM が管理できる、サポートされている CICS リリースの詳細については、[アップグレードの CICSplex SM のアップグレードに関する情報を参照してください](#)。

ある CICSplex 内のすべての MAS は、同じ *CICSplex SM* アドレス・スペース (CMAS) または同じグループの CMAS によって管理されます。

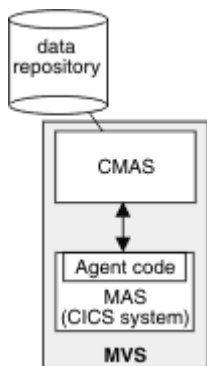


図 2. MAS およびそれらの CMAS

それぞれの MAS には、それを管理する CMAS 用の CICSplex SM 機能 (データ収集など) をインプリメントする CICSplex SM エージェント・コードがあります。例えば、特定の MAS においてリソース・モニターがアクティブである場合、MAS 内のエージェント・コードは選択されたリソースをモニターし、結果データを CMAS に送ります。

単一の CICSplex 内にある MAS は、CICSplex SM 固有の通信においては明示的に互いに接続する必要はありません。ただし、トランザクション・ルーティングと機能シップのサポートで必要な CICS 接続は必要です。

MAS は、個々の CICS システムとして定義および管理することもできますし、CICSplex 内のシステム・グループにグループ化することもできます。各システム・グループは、CICSplex の単位サブセットとして管理する 1 つ以上の CICS システムです。システム・グループについては、[システム・グループの指定](#)で説明します。

すべての MAS はローカルです。つまり、それらは、それらを管理する CMAS と同じ MVS イメージ上で実行されます。CMAS およびローカル MAS の間の明示的リンクを定義する必要はありません。ローカル MAS の場合、システム管理データは (6 ページの図 1 に示されているように) データ・スペース・キャッシュに累積され、環境サービス・システム・サービス (ESSS) アドレス・スペース (これについては、[環境サービス・システム・サービス \(ESSS\)](#) で説明します) を介して CMAS に送られます。

MAS は CICSplex SM Web ユーザー・インターフェース・サーバーとしてセットアップできます。この場合、MAS および CICSplex SM Web ユーザー・インターフェース・サーバーの CICS リリース・レベルは同じである必要があります。

CICSplex SM アドレス・スペース (CMAS)

CICSplex SM アドレス・スペース (CMAS) は、すべての CICSplex SM 構成のハブといえます。なぜならこれは CICS システムおよびそれらのリソースの管理およびレポート作成に関係するほとんどの作業を担当するからです。各 CICSplex は、少なくとも 1 つの CMAS によって管理されます。CMAS は、単一システム・イメージ (SSI) を担当します。SSI を使用すると、ある CICSplex に属すると定義された CICS システムの数に関係なく、さらにそれらの物理的位置に関係なく、オペレーターはその CICSplex を単一の CICS システムであるかのようにして管理できます。

CMAS は、CICSplex SM の BAS、WLM、RTA、およびモニター機能をインプリメントし、それが管理している CICSplex に関する構成情報を保守します。さらに CMAS は、それが所有する他の CMAS とのリンクに関する情報も保持します。この情報は、データ・リポジトリに保管されます ([7 ページの図 2](#) を参照してください)。

CMAS は、CICS Transaction Server for z/OS システムです。ほとんどの CMAS コンポーネントは CICS タスクとして実行され、他のコンポーネントへの CMAS 接続は、CICS 相互通信方式を使用してインプリメントされます。

注: CMAS はユーザー・アプリケーションや端末はサポートせず、CMAS 以外のタスク (CICSplex SM の一部として提供されるツール以外のあらゆるモニター・ツールおよびパフォーマンス・ツールの使用を無制限に含む) では、CMAS のリソースは使用可能であると考えるべきではありません。

CMAS は、その MAS より低いリリースの CICS を実行することはできません。また、CMAS と MAS の両方は、同じリリースの CICSplex SM を実行する必要があります。

CMAS は、どの CICSplex の一部でもありません。単一の CMAS が複数の CICSplex の管理に関与する場合がありますが、そのいずれにも属することはありません。

1 つの CICSplex が複数の CMAS によって管理される場合には、以下のようになります。

- CMAS は、CICSplex SM に定義された CMAS 間リンクによって互いに接続されます。これらのリンクによって、各 CMAS は他の CMAS が保持しているデータにアクセスでき、単一システム・イメージをオペレーターに示すことができます。
- CMAS の 1 つは、保守ポイント CMAS として指定されます。その CMAS は、CICSplex に関連するすべての CICSplex SM 定義の保守を担当し、すべてのデータ・リポジトリの同期を保ちます。

CICSplex SM は、CICSplex SM が処理している NetView インスタンスと同じ MVS イメージに CMAS がインストールされている場合、SNA 総称アラートを NetView に発行できます。

環境サービス・システム・サービス (ESSS)

環境サービス・システム・サービス (ESSS) は、z/OS サービスを CICSplex SM コンポーネントに提供する制限された機能である、MVS システム・アドレス・スペースです。特に、ESSS は z/OS イメージ上のすべての MVS/ESA データ・スペースを所有し、それらが CMAS および MAS とは独立して存在していてもその両方からアクセス可能になるようにします。

この配置の利点は、データ・スペース内に累積される CICSplex SM データは、MAS および CMAS コンポーネントでのイベントによって影響を受けることがないという点です。ESSS は、ある CMAS と同じ z/OS イメージ上にある NetView インスタンスとその CMAS との間の通信における一部の局面において、特定の役割を果たします。

どの z/OS イメージにおいても、その CMAS のバージョンまたはレベルごとに、ESSS の 1 つのインスタンスがあります。

CICSplex SM オブジェクト

CICS システムの構成を CICSplex SM に定義し、さらに BAS、WLM、RTA、およびモニター要件を定義するには、*CICSplex SM* オブジェクト を作成し、それらを互いに関連付けます。

オブジェクトごと、およびそれらの間の関連またはリンクごとに、CMAS データ・リポジトリ内にレコードが作成されます。9 ページの図 3 は、CICSplex SM オブジェクトの相互関係を示しています。

この図には、この表示フォーマットではサポートされない高解像度グラフィックが含まれています。このグラフィックを表示するには、CICS インフォメーション・センターを使用してください。

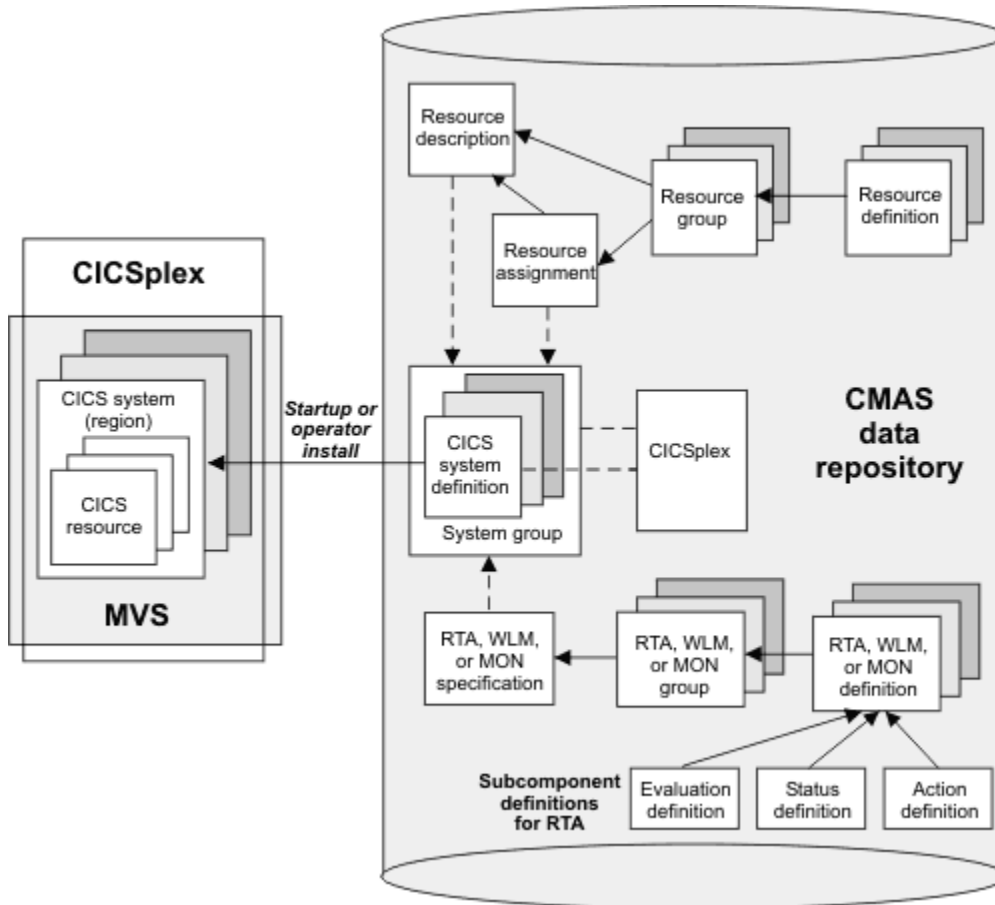


図 3. CICSplex SM オブジェクト・モデル

これらのオブジェクトは、次の 3 つのカテゴリーに分割できます。

1. *CICSplex* オブジェクトおよび CICS システム・オブジェクト。*CICSplex SM* によって管理される *CICSplex* および CICS システムを定義するために使用されます。これには、CICS システム間のリンクを定義することと CICS システム・グループを作成することが含まれます。これらのオブジェクトについては、[Designing your CICSplex SM environment](#) で説明します。これらのオブジェクトの定義についての情報は、[CICSplex SM 構成とトポロジーの定義](#)にあります。
2. *BAS* オブジェクト。CICS リソース定義およびインストール・プロセスの論理スコープ設定および管理に使用されます。このカテゴリーのオブジェクトは、CICS システムが使用するリソースを定義するものと見なすことができます。これらの *CICSplex SM* オブジェクトについては、[複数バージョンのリソース定義](#)で説明しています。*BAS* オブジェクトの定義に関する情報は、[全機能形式 BAS の使用](#)にあります。
3. 操作 オブジェクト。実行中の CICS システムに存在する CICS リソースを操作するために使用されるオブジェクトです。このカテゴリーのオブジェクトは、自動化ワークロード管理、自動化例外報告、および統計データの収集において、実行時にモニターするリソースを特定するものと見なすことができます。それらの *CICSplex SM* オブジェクトは、[モニター機能の管理](#)、[ワークロード管理リソース](#)、および [RTA の管理](#)で説明されています。

データ・リポジトリ

1つのCICSシステムまたは複数のCICSシステムのグループの単一システム・イメージは、CICSplex SMと、データ・リポジトリに保持されているリソース定義を使用してCMASによって提供されます。

各CMASには独自のデータ・リポジトリがあります。データ・リポジトリは、CICSplex SMのインストール後ジョブを使用して作成されるVSAM KSDSです。すべてのデータ・セットと同様、ご使用の環境に含まれる各データ・リポジトリのバックアップを定期的に取りする必要があります。

データ・リポジトリに保持されているCICSplex SM定義およびCICSリソース定義は、以下の方法で管理できます。

- CICS Explorerの使用。[CICS Explorer 製品資料内の『Working with resources』](#)を参照してください。
- WUIビューの使用。[CICSplex SM Web ユーザー・インターフェース](#)を参照してください。
- EYU9XDBT CICSplex SM 定義ユーティリティの使用。
- バッチ処理されるリポジトリ更新機能 (BATCHREP) の使用。[バッチ処理されるリポジトリ更新機能](#)を参照してください。
- APIの使用

データ・リポジトリにある定義の視覚マップを生成できます。マップは、ビジネス・アプリケーション・サービス、リソース・モニター、リアルタイム分析、またはワークロード・モニターについて生成できます。WUIの詳細または表形式ビューから、マップの開始点 (CICSシステム・グループ、個々のリソースなど) を選択します。CICSplex SMには、開始点と、その定義を参照するかその定義によって参照されるすべての定義が表示されます。これについては、[CICSplex SM 定義をマップする方法](#)を参照してください。

CICS 管理クライアント・インターフェース (CMCI)

CICS 管理クライアント・インターフェース (CMCI) は、IBM CICS Explorer などの HTTP クライアント・アプリケーションによって使用されるシステム管理アプリケーション・プログラミング・インターフェースです。CMCIは、インストール済みシステム・リソースおよび定義システム・リソースを管理するCICSシステム管理クライアント用のCMCI REST API および CMCI GraphQL API を提供します。また、クライアント認証のサポートも提供します。

CMCIは、CICSplex SM 環境にも、またスタンドアロン CICS 領域 (SMSS) にもインストールできます。

CICSplex SM 環境での CMCI

CMCIをCICSplex SMと共に使用する場合は、CICSplex SMで管理されている全CICS領域の定義リソース、操作リソース、およびCSDリソースを管理できます。

SMSS 環境での CMCI

スタンドアロンのCICS領域でCMCIを使用する場合、その領域に関連付けられている操作リソースおよびCSDリソースのみを管理でき、コンテキストはそのCICS領域のアプリケーションIDとして指定されます。

CMCI JVM サーバーとは何か

CMCI JVM サーバーは、Liberty サーバーです。これはオプションですが、CMCI 要求のサポートを強化する点で強くお勧めするCMCIのコンポーネントです。このコンポーネントは、多要素認証 (MFA) のサポートを含むクライアント認証を実行し、CMCI GraphQL APIをサポートします。

10 ページの表 1 は、CMCI JVM サーバーが構成されたCMCIで利用できる機能と、(CMCI JVM サーバーがない) 基本的なCMCIで利用できる機能とを比較したものです。

表 1. CMCI での機能の使用可能性		
機能	CMCI JVM サーバー による CMCI	CMCI (基本)
認証サポート		
ユーザー ID/パスワード	✓	✓
Certificate	✓	✓
パスチケット	✓	

表 1. CMCI での機能の使用可能性 (続き)		
機能	CMCI JVM サーバー による CMCI	CMCI (基本)
MFA	✓	
API サポート		
REST API	✓	✓
GraphQL API	✓	
CICS バンドル・デプロイメント API	✓ (追加の構成が必要)	

CMCI JVM サーバーの使用可能化は、機能切り替え `com.ibm.cics.cmci.jvmserver` によって制御されます。

注：スタンドアロン CICS 領域 (SMSS) では、CMCI JVM サーバーはサポートされません。

CMCI REST API と CMCI GraphQL API: 説明と両者の違い

CMCI REST API および CMCI GraphQL API は、いずれも HTTP ベースのアプリケーション・プログラミング・インターフェースです。これらは、CICSPlex SM によって管理されている CICS 領域上のインストール済みおよび定義 CICS および CICSPlex SM リソースを管理する HTTP クライアント・アプリケーションを開発するために使用できます。

CMCI REST API は、Representational State Transfer (RESTful) 原則に基づいて設計されているため、固定データ構造を使用して複数のエンドポイントからデータを取得する必要があります。一方、GraphQL API は単一のエンドポイントのみを公開し、より高い柔軟性を備えています。そのため、クライアント、単一の照会要求で、複数の CICSPlex をまたいで多くのタイプの CICS リソースを照会でき、また、明示的に示された固有の関係において必要なデータを具体的に指定することができます。

例えば、[11 ページの図 4](#) の GraphQL 照会は、接続されているすべての CICSPlex 内のすべての領域におけるローカル・トランザクションと関連プログラムに関するデータ (使用回数を含む) を取得します。CMCI REST API で同じ結果を得るには、まず、使用可能なローカル・トランザクションのリストを返すエンドポイントにアクセスした後、すべてのプログラムを返すエンドポイントにアクセスするという方法があります。その場合は、これらの結果を後処理してローカル・トランザクションとプログラムをマッチングさせるためのクライアント・コードを作成する必要があります。GraphQL の場合、CMCI REST API よりも CMCI GraphQL API を使用した方が、照会されるリソース内の関係もより明示的に示すことができます。

```
{
  cicsplexes {
    cicsResources {
      loctrans {
        records {
          name
          to_program {
            name
            useCount
          }
        }
      }
    }
  }
}
```

図 4. ローカル・トランザクションに関連するプログラムを要求する CMCI GraphQL API 照会

CICS バンドル・デプロイメント API: 内容

CMCI は、CICS バンドル・デプロイメント API を介して、単一の CICS 領域への CICS バンドルのデプロイをサポートします。

この REST API は、HTTP を介して CICS バンドルを zip ファイルとして受け取ります。バンドルは、適切な CICS 領域で自動的に unzip、インストール、および使用可能化されます。同じ名前の CICS バンドルが既に

存在する場合、その CICS バンドルは使用不可になり破棄されてから、新しいバンドルがインストールされます。

CICS バンドル・デプロイメント API は、Java 開発者がアプリケーションに加えた変更内容を実行中の CICS 領域内で数秒以内に反映可能にすることで、Java 開発者の生産性を高めることができます。開発者は、API を利用する CICS 提供の Maven プラグインまたは Gradle プラグイン (cics-bundle-maven-plugin または `com.ibm.cics.bundle`) を使用して、CICS バンドルのビルドとデプロイメントをツールチェーンに統合することもできます。

この API を使用すると、システム・プログラマーが制御権を保持している状態で、Java 開発者がバンドルをデプロイすることも可能になります。機能 ID または十分なアクセス権限を持つ別のユーザー ID が、開発者の代わりにバンドル・ライフサイクルを処理して zFS とやり取りします。

API について詳しくは、[17 ページの『仕組み: CICS バンドル・デプロイメント API』](#)を参照してください。

CICS を API 向けに構成するには、[CMCI JVM サーバーを CICS バンドル・デプロイメント API 用に構成する](#)を参照してください。

仕組み: CMCI REST API

CICS 管理クライアント・インターフェース (CMCI) は、IBM CICS Explorer などのシステム管理クライアント用の REST アプリケーション・プログラミング・インターフェース (API) を提供します。

CMCI REST API は、HTTP を介してサポートされます。クライアントは、CMCI に対して HTTP 要求を開始します。このインターフェースは、要求が有効であることを判別すると、CICSplex SM API コマンドを構成します。スタンドアロン CICS 領域の場合には、CICS システム・コマンドを構成します。コマンドを実行した後、CMCI は HTTP 応答を作成します。要求が成功すると、応答は HTTP 200 (OK) 応答コードと結果セットが入った XML フィードの形式で、クライアントに戻されます。要求が失敗した場合、OK 以外の HTTP 応答コードと失敗の詳細が応答に含まれます。

CMCI の HTTP 要求と応答のフォーマットは、HTTP/1.1 プロトコルに基づきます。このプロトコルの詳細については、[HTTP プロトコル](#)を参照してください。

CMCI HTTP 要求の作成方法

CMCI 要求の形式は、HTTP ヘッダー、その後に URI (Universal Resource Identifier)、そして CICS または CICSplex SM リソースに加えられたすべての変更の詳細が入った XML 本体 (必要な場合) となります。

ヘッダーには、以下の HTTP メソッドのいずれかが取り込まれます。

DELETE

CICSplex SM データ・リポジトリからリソースを除去するか、CSD からリソースを除去するか、またはインストール済みリソースを破棄します。

GET

CICSplex SM データ・リポジトリにあるリソースに関する情報を取り出すか、CSD にあるリソースに関する情報を取り出すか、またはインストール済みリソースに関する情報を取り出します。

POST

CICSplex SM データ・リポジトリにリソースを作成するか、CSD にリソースを作成します。

PUT

CICSplex SM データ・リポジトリにある既存のリソースを更新するか、CSD の既存のリソースを更新するか、または属性を設定してインストール済みリソースに対してアクションを実行します。また、CICSplex SM リソースおよび CSD リソースに対してもアクションを実行します。

URI には、CICS または CICSplex SM リソースの名前が含まれます。これを使用して、指定されたリソースの 1 つ以上のインスタンスを識別するために照会のスコープと性質を詳細化する一連のパラメーターを指定します。GET 要求の URI では、API が結果のセットを保持するかまたは破棄するかも指定します。API が結果を保持する場合、新しい要求は、取り出し操作を繰り返さずに、保持された結果に対して作用できます。後続の要求を使用して、1 度に 1 つ以上のレコードを選択しながら、保持された結果のページを送ってゆくこともできます。

POST および PUT 要求には、XML 本体も組み込まれます。PUT 要求の場合、本体に収容される内容は、リソース属性に加えられる変更の詳細、またはターゲット・リソースに対して実行されるアクションです。POST 要求の場合、本体に取り込まれる内容は、新規リソース・インスタンスに設定する属性値です。

GET 要求および DELETE 要求では XML 本体は必要ありません。DELETE 要求に追加のパラメーターが必要な場合、それらのパラメーターは URI に含められる必要があり、オプションで XML 本体に追加することもできます。

その他トピック

CMCI [RESTful API プログラミング・リファレンス](#)に、DELETE、GET、POST、および PUT メソッド、CMCI リソース名、CMCI XML 本体要素、診断エイドなどに関する詳細が示されています。

仕組み: CMCI GraphQL API

CICS 管理クライアント・インターフェース (CMCI) は、IBM CICS Explorer などのシステム管理クライアント用の GraphQL アプリケーション・プログラミング・インターフェース (API) を提供します。CMCI GraphQL API は、HTTP によりサポートされています。GraphQL API を使用すると、クライアントは、単一の要求で、複数の CICSplex をまたいで多くのタイプの CICS リソースを照会できます。単一の照会要求では、クライアントは、CICS リソース間の本来の関係が明示的に示された複数の CICS リソースについて必要なデータを正確に指定できます。

CICS Explorer CICS TS 5.5 の集約関数には CMCI GraphQL API が必要です。CMCI GraphQL API を使用するには、CMCI で CMCI JVM サーバーを使用する必要があります。GraphQL について詳しくは、[GraphQL の概要](#)を参照してください。

GraphQL 照会とは

単純な GraphQL 照会要求は次のようになります。

```
{
  cicsplexes {
    name
  }
}
```

図 5. CICSplex 名を要求する単純な照会

照会のルートに cicsplexes フィールドがあるため、WUI サーバーが接続されているすべての CICSplex が検索されます。cicsplexes フィールドにネストされている name フィールドによって、各 CICSplex の名前が要求されます。

照会応答は JSON オブジェクトとして返され、要求されたデータは data フィールドの値で囲まれます。応答の構造は、照会の構造に従います。

```
{
  "data": {
    "cicsplexes": [
      {
        "name": "CICSPLX01"
      },
      {
        "name": "CICSPLX02"
      }
    ]
  }
}
```

図 6. CICSplex 名に関する単純な照会に対する応答

照会フィールド (ネストされたものを含め) を追加することで、より多くの情報を取得できます。これについては、[14 ページの『照会の例』](#)を参照してください。

GraphQL API 要求の作成方法

GraphQL API エンドポイントは次の場所にあります。

```
https://host:port/graphql
```

ここで、*host* および *port* は、CMCI JVM サーバー のホスト名とポート番号です。

GraphQL API は、GET 要求と POST 要求を受け入れます。

GET 要求の場合:

Content-Type: application/json ヘッダーを送信する必要があります。query 照会パラメーターで照会を指定します。オプションの operationName 照会パラメーターで操作を指定します。

例えば、[13 ページの図 5](#) の単純な照会は、以下の URL を使用して送信できます。

```
https://host:port/graphql?query={cicsplexes{name}}
```

POST 要求の場合:

Content-Type: application/json ヘッダーを送信する必要があります。要求の本体は、JSON でエンコードされたオブジェクトである必要があります。

```
{
  "query": "query_body",
  "operationName": "operation_name"
}
```

ここで、query フィールドのみ必須です。

または、POST 要求で Content-Type: application/graphql ヘッダーを送信することもできます。この場合、要求の本体は GraphQL 照会自体である必要があり、操作名を指定することはできません。

GraphQL 照会のサンプル・コードについては、[14 ページの『照会の例』](#)を参照してください。

照会の例

GraphiQL またはオンラインの GraphQL 可視化エディターを使用して、GraphQL 照会またはサンプルをテストできます。GraphiQL の URL は以下のとおりです。

```
https://host:port/graphiql
```

ここで、*host* および *port* は、CMCI JVM サーバーのホスト名とポート番号です。

GraphiQL のヒント :

- GraphiQL では、オートコンプリート機能と、GraphQL スキーマ参照用の組み込みドキュメンテーション・エクスプローラーを使用できます。Ctrl + スペースを押して、使用可能なフィールド名を表示できます。
- GraphiQL ヒストリー内の照会を容易に区別するには、照会の接頭部に query *QueryName* を付けて固有の照会名を指定します。

以下の例では、接続されているすべての CICSplex 内のすべての領域のローカル・ファイルのカウントおよび各 CICSplex と領域の名前を照会します。また、照会名として LocalFilesInRegionsInCICSplexes が設定されています。

```
query LocalFilesInRegionsInCICSplexes {
  cicsplexes {
    name
    regions {
      name
      cicsResources {
        locfile {
          count
        }
      }
    }
  }
}
```

照会の対象となる属性をさらに追加できます。この例では、すべての CICSplex および各 CICSplex 内のすべての領域を照会します。各領域内で、すべてのローカル・トランザクションの **name**、**useCount**、および **status** フィールドが取得されます。

```
{
  cicsplexes {
    name
    regions {
      name
      cicsResources {
        loctran {
          records {
            name
            useCount
            status
          }
        }
      }
    }
  }
}
```

照会の対象となる CICSplex または CICS 領域を指定できます。この例では、CICSplex PLEX1 内の領域 AORRGN を照会し、その領域内のすべてのローカル・トランザクションの **name** フィールド、**useCount** フィールド、および **status** フィールドを取得します。

```
{
  cicsplex(name: "PLEX1") {
    name
    region(name: "AORRGN") {
      name
      cicsResources {
        loctran {
          records {
            name
            useCount
            status
          }
        }
      }
    }
  }
}
```

CICSplex と領域の指定を削除したこの例では、接続されているすべての CICSplex と、それらの CICSplex 内のすべてのローカル・トランザクションの **name**、**useCount**、および **status** フィールドを照会します。

```
{
  cicsplexes {
    name
    cicsResources {
      loctran {
        records {
          name
          useCount
          status
        }
      }
    }
  }
}
```

この例は、CED で始まるトランザクションのみを取得するフィルターを使用する点を除き、前の例と類似しています。

```
{
  cicsplexes {
    name
    cicsResources {
      loctran(filter: {name: {value: "CED*"}}) {
        records {
          name
          useCount
        }
      }
    }
  }
}
```

```

    status
  }
}
}
}
}

```

CICS 定義を照会することもできます。この要求は、CICSplex データ・リポジトリ内のすべてのファイル定義の **name** 属性および **update** 属性を照会します。

```

{
  cicsplex(name: "PLEX1") {
    drep {
      cicsDefinitions {
        filedef {
          records {
            name
            update
          }
        }
      }
    }
  }
}

```

同様に、この要求は、CICSplex PLEX1 内の領域 AORRGN での CSD におけるすべてのパイプライン定義の名前を照会します。

```

{
  cicsplex(name: "PLEX1") {
    region(name: "AORRGN") {
      csd {
        cicsDefinitions {
          pipedef {
            records {
              name
            }
          }
        }
      }
    }
  }
}

```

この照会は、各 CICSplex 内のすべてのローカル・ファイルの集約を実行し、**name** 属性の共通値でグループ化し、各集約グループ内の集約レコードのカウント、各グループの名前、および各グループ内の平均、最小、最大の **readCount** を取得します。

```

{
  cicsplexes {
    name
    cicsResources {
      locfile {
        groupBy(attribute: "name") {
          count
          aggregateRecord {
            name {
              value
            }
          }
          readCount {
            average
            min
            max
          }
        }
      }
    }
  }
}

```

仕組み: CICS バンドル・デプロイメント API

CICS 管理クライアント・インターフェース (CMCI) は、CICS バンドル・デプロイメント API を介して CICS バンドルを単一の領域にデプロイすることをサポートします。この API を使用するには、CMCI JVM サーバーの追加構成を使用可能にする必要があります。

目次

[17 ページの『概要』](#)

[19 ページの『API の仕組み』](#)

[21 ページの『API のセキュリティー・モデル』](#)

[21 ページの『CICS バンドル・デプロイメント API 要求の作成方法』](#)

[22 ページの『次のステップ』](#)

概要

Java 開発者は開発時にこの API を使用することで、アプリケーションに加えた変更内容を実行中の CICS 領域内で数秒以内に反映させることができます。この API と組み合わせて使用するために、CICS バンドルのビルドとデプロイメントを開発者のツールチェーンに統合するために使用できる Maven プラグインと Gradle プラグインも CICS 側で提供されています。CICS での Maven および Gradle のサポートについて詳しくは、[CICS における Java サポート](#)を参照してください。

CICS バンドル・デプロイメント API を使用すると、システム・プログラマーが制御権を保持している状態で、Java 開発者はバンドルをデプロイできます。このことは、開発者が FTP を使用して zFS にバンドルを書き込む必要や、CSD からバンドルをインストールする必要をなくすことで実現されます。これらの操作は、機能 ID または十分なアクセス権限を持つ別のユーザー ID を使用して API 側で実行されます。

次の図は、複数のアプリケーション開発者が、WUI 領域内で構成されている API にバンドルをプッシュする一般的な状況を示しています。

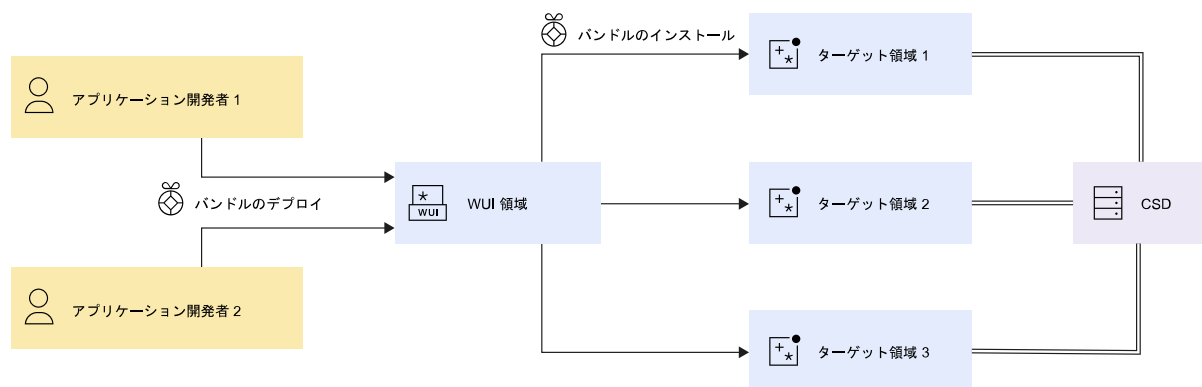


図 7. 複数の開発者が API にバンドルをプッシュ

この REST API は、開発者のアプリケーションが含まれた CICS バンドルを HTTP 経由で zip ファイルとして受け取ります。バンドルは、適切な CICS 領域で自動的に unzip、インストール、および使用可能化されます。同じ名前の CICS バンドルが既に存在する場合、その CICS バンドルは使用不可になり破棄されてから、新しいバンドルがインストールされます。

通常、システム・プログラマーは、必要なパラメーター (CICSplex、CICS 領域、CSD グループ、および BUNDLE 定義の名前) を Java 開発者に渡します。システム・プログラマーは、開発中のアプリケーションの正しいバンドル・ディレクトリー属性を使用して構成された BUNDLE 定義を作成します。CICS は、CSD から単一領域へのバンドル・インストールをサポートしています。

API の仕組み

前述のように、API はバンドルのライフサイクルを管理し、バンドルの受信後に開発者の代わりに zFS と相互動作します。通常、API は、CICS 側で提供される `cics-bundle-maven-plugin` や `com.ibm.cics.bundle` などのクライアントによって駆動されます。

API はバンドルのディレクトリーを使用して構成されます。これは、バンドルの管理に API を使用するための専用にする必要がある zFS 上の場所です。API にプッシュされたバンドルはバンドルのディレクトリー内に unzip されて、CICS ターゲット領域によってアクセスされます。このことは開発者によって意識されることなく、システム・プログラマーによって管理されます。この API を使用する開発者は、zFS 上のバンドルを操作する必要はありません。

開発者が取り組んでいるアプリケーションごとに、システム・プログラマーは 1 つの **BUNDLE 定義を CSD 内に作成**します。この BUNDLE 定義の BUNDLEDIR 属性には、バンドルのディレクトリーにアップロードされた正しいバージョンのアプリケーションへのパス (例: `/u/path/to/bundles/dir/ApplicationName_1.0.0`) を設定する必要があります。システム・プログラマーは、開発者が API を駆動するための BUNDLE 定義名と CSD グループを指定します。この API を使用する開発者は、CICS BUNDLE リソースのインストール、無効化、および破棄を行う必要はありません。

次の図は、アプリケーション開発者によって公表された時点から CICS 領域にインストールされるまでの **バンドルのライフサイクル** を示しています。これは、ユーザーの介入なしに API によって自動的に処理されます。

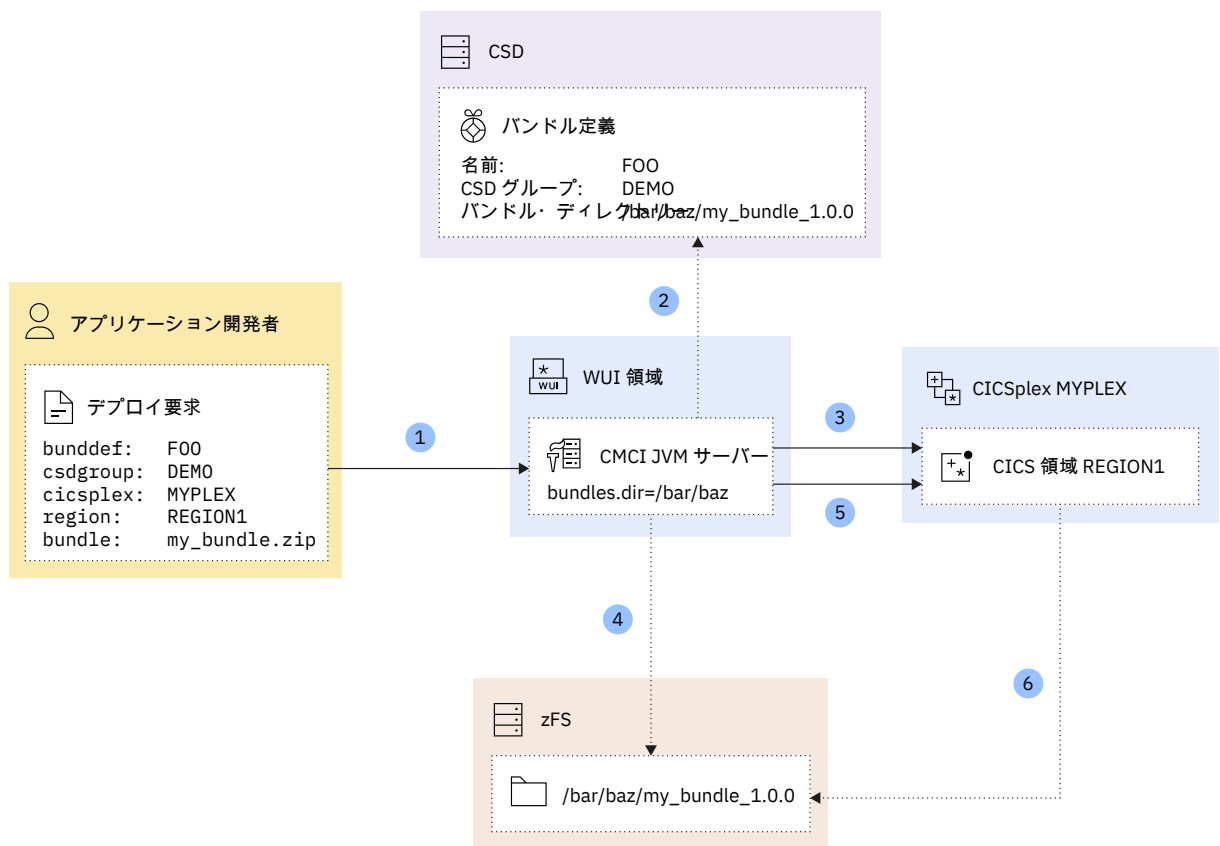


図 8. API におけるバンドルのライフサイクル

- 1 アプリケーション開発者は、CICS バンドル・デプロイメント API を介してアプリケーション・バンドルを公表します。

検証

- 2 CMCI JVM サーバーはターゲット領域の CSD 内で BUNDLE 定義を検出して、BUNDLE 定義のバンドル・ディレクトリー (BUNDLEDIR) 属性値が API の構成済みバンドルのディレクトリー内であることを確認します。

アンインストール

- 3 CMCI JVM サーバーは、指定された BUNDLE 定義と同じ名前を持つインストール済みバンドルがターゲット領域内に存在するかどうかを確認します。このようなバンドルは使用不可になり、必要に応じて破棄されます。
- 4 CMCI JVM サーバーは、名前とバージョンが同じである以前のバンドルがある場合は、それらのバンドルを zFS 上のバンドルのディレクトリーから削除します。次に、このサーバーは公表されたバンドルをバンドルのディレクトリー内にアンパックします。

Install

- 5 CMCI JVM サーバー は BUNDLE 定義の CSD インストールを開始します。
- 6 ターゲット領域によって zFS からこのバンドルが読み取られてインストールされます。

API のセキュリティ・モデル

制御されたアクセスを実現するために、機能 ID が導入されました。この結果として、アプリケーション開発者は、バンドルを操作するための一般アクセス権なしで、CICS バンドル・デプロイメント API に対する特定のアクセス権を通じてバンドルをデプロイできるようになりました。API のワークフローでは、以下のようさまざまなユーザー ID が使用されます。

- アプリケーション開発者は通常どおりに認証されて、アプリケーション開発者のユーザー ID が API に対するアクセス権を持っていることが確認されます。
- タスクがアプリケーション開発者のユーザー ID から構成済みのデプロイメント機能 ID に切り替わります。
- このデプロイメント機能 ID によってバンドルが zFS に書き込まれて、ターゲット領域へのバンドルの CSD インストールが開始されます。
- ターゲット領域の領域ユーザー ID によって、インストール時に zFS からバンドルが読み取られます。

API が機能するためには、API のセキュリティ・メカニズムで追加の構成が必要です。

- システム・プログラマーは SAF を正しく構成して、API を使用したバンドル・デプロイメントを可能にする必要があります。
- API は、SEC=YES として構成された領域内のみで機能します。

詳しくは、[CMCI JVM サーバーを CICS バンドル・デプロイメント API 用に構成する](#)を参照してください。

CICS バンドル・デプロイメント API 要求の作成方法

CICS では、この API を使用してバンドルを CICS に公表する Maven プラグインと Gradle プラグインが提供されます。Java 開発者は、開発時にこのプラグインをバンドル・デプロイメントのために使用できます。詳しくは、[cics-bundle-maven](#) を参照してください。または、独自のツールを記述することを希望する場合は、任意の標準 REST クライアントを使用して API を呼び出すことができます。

API 要求の形式は HTTP /1.1 に基づいています。この API では POST 要求のみが使用され、マルチパート・フォーム・データが受け入れられます。Content-Type: multipart/form-data ヘッダーを送信する必要があります。ストリング・パラメーターおよび zip 形式で圧縮されたバンドル・ファイルは、ペイロード内でフォーム・データ・パーツとして受け入れられます。

API は、HTTP 応答コードと、要求の結果の JSON 記述を返します。応答コード 4xx は、ユーザーが正しくない形で API をアドレス指定したことを示します。応答コード 5xx は、システムが正しくセットアップされていないことを示します。

API エンドポイントは、標準 CMCI インターフェースと一緒に提供されます。例えば、CMCI インターフェースをポート 4444 上の mycicshost.com で使用可能な場合、API エンドポイントの URL は `https://mycicshost.com:4444/managedcicsbundles` になります。

以下のパラメーターを指定して API を呼び出します。

cicsplex

ターゲット領域が属している CICSplex の名前。

領域

バンドルのインストール先にする必要がある領域の名前。

csdgroup

ソース BUNDLE 定義が存在する CSD グループの名前。

bunddef

BUNDLE 定義の名前。

バンドル

バンドルの内容を含む zip ファイル (content-type application/zip という形式)。META-INF ディレクトリーは、この zip ファイルのルートになければなりません。

例えば、コマンド・ライン・ツール cURL を使用してローカル・ワークステーションから API を呼び出す場合、コマンド・ラインは次のようになります。

```
curl -i -X POST -u MYUSER -F cicsplex=MYPLEX -F region=MYREGION -F csdgroup=CSDGRP -F bunddef=MYBUND -F bundle=@c:/path/to/bundle_0.0.1.zip https://mycicshost.com:4444/managedcicsbundles
```

CICS バンドル・デプロイメント API は、コード・ページ変換を実行せずに、アップロードされた zip ファイルの内容を unzip します。バンドル内の各ファイルは、zip 圧縮される前に正しいコード・ページでエンコードする必要があります。

次のステップ

この API を使用してバンドルをデプロイするように CMCI JVM サーバーを構成します。[CMCI JVM サーバーを CICS バンドル・デプロイメント API 用に構成する](#)を参照してください

関連概念

[Maven または Gradle を使用したアプリケーションの開発](#)

CMCI セキュリティー機能: CMCI によるクライアントに認証方法

CICS Explorer などの HTTP システム管理クライアントがサインオンしようとするとき、CMCI はユーザー資格情報を検査します。ユーザー資格情報には、ユーザー ID とパスワード、パスチケット、MFA トークン、または証明書を使用できます。CMCI JVM サーバーを有効にすると、認証プロセスが処理されます。パスチケットまたは MFA トークンによる認証は、CMCI JVM サーバーでのみ使用可能です。

CMCI JVM サーバーがクライアントを認証する方法

22 ページの図 9 は、CICS Explorer に基づくクライアント認証のワークフローを示しています。

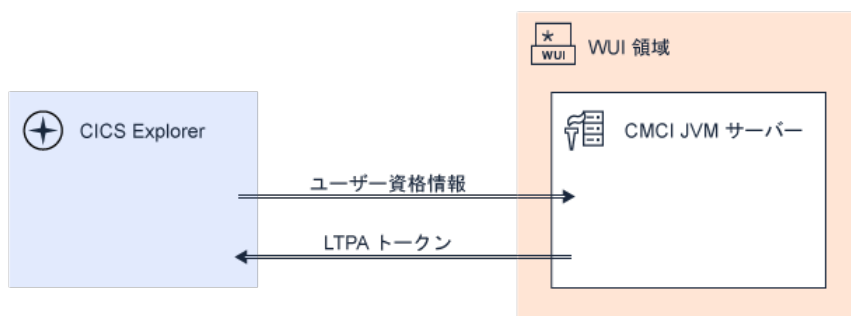


図 9. CMCI HTTP クライアント認証のワークフロー

1. ユーザーが CICS Explorer からログオンすると、CICS Explorer はユーザー資格情報を CMCI JVM サーバーに渡します。ユーザー資格情報には、ユーザー ID とパスワード、パスチケット、MFA トークン、または証明書を使用できます。
2. CMCI JVM サーバーは、外部セキュリティ・マネージャー (ESM) への SAF インターフェースを使用してユーザーの資格情報を検証し、LTPA トークンを生成します。
3. CMCI JVM サーバーは、応答および LTPA トークンを使用して CICS Explorer に応答します。

以降の要求では、CICS Explorer は LTPA トークンを使用してユーザーを認証します。

注：

- LTPA トークンは Cookie です。したがって、HTTP クライアントは Cookie を受け入れる必要があります。
- CMCI のトランスポートおよび認証に JVM サーバーが使用されますが、ほとんどの処理は CICS コアで行われます。したがって、専用エンジンによるオフロードの向上を CMCI JVM サーバーに期待することはできません。

LTPA タイムアウト

LTPA トークンの存続時間は固定されています。セッションでユーザーがアクティブであっても、延長したり更新したりすることはできません。タイムアウトになるとユーザーはログアウトされ、新しいトークンを取得するためにはログイン資格情報を再度入力する必要があります。LTPA トークンの有効期限時間は構成可能です。詳しくは、[Liberty](#) での [LTPA の構成](#) を参照してください。

LTPA トークンの共用

Liberty ではシングル・サインオン (SSO) 構成がサポートされているため、複数の領域間で LTPA トークンを共用できるように Liberty をセットアップできます。HTTP クライアント・ユーザーは 1 回認証でき、同じ LTPA 鍵を共用する他の領域にアクセスできます。詳しくは、[Liberty](#) を参照してください。

CMCI JVM サーバーなしの CMCI がクライアントを認証する方法

CMCI で CMCI JVM サーバーが使用されない場合、ユーザーの認証は、証明書または HTTP ヘッダー内の基本認証子を使って行われます。

1 回使用の (ワンタイム) トークン (MFA トークンやパスチケットなど) はサポートされていません。

その他トピック

では、Liberty での認証プロセスの概要を示し、LTPA および SSO について詳しく説明しています。

[CMCI のセットアップ](#) で、構成手順を示しています。

第 2 章 CICSplex SM 環境の設計

システム管理要件を満たす CICSplex SM 環境を設計するために企業システムのマップを行います。設計には、トポロジー情報 (必要な CICSplex、CMAS、WUI サーバー領域を指定すること、従う命名規則などのその他の設計上の決定を行うことなど) を含める必要があります。

始める前に

CICSplex SM 環境を設計するには、CICSplex SM の概念とコンポーネントを十分理解する必要があります。

CICSplex SM のセットアップや保守に関する問題を回避するために、以下の考慮事項にご注意ください。

- CICSplex SM にサービスを適用する場合、ESSS に適用される PTF は、同じリリースの以前の保守レベルとの下位互換が想定されていません。つまり、すべての CMAS、MAS、WUI サーバー領域および API プログラムは、ESSS と同じリリースの保守レベルで実行する必要があります。そうしない場合、常終了やデータ破損につながったり、予期しない結果になる可能性があります。PTF を CICSplex SM に適用する場合、SMP/E 保守に関連付けられたすべての ++HOLD ACTION 項目は、注意深くフォローされていることが必要です。
- さらに、CICSplex SM トポロジーを設計する際には、以下のガイドラインを考慮してください。
 1. 実稼働と非実稼働 (例えば、テスト、開発、または QA) の各領域を同じ LPAR 内で実行することは避けてください。同じ CICSplex SM リリースを実行するすべての領域は、ESSS の 1 つのコピーを使用 (つまり、ESS を共用) します。その結果、この共用 ESSS に PTF を適用する場合は、実稼働領域と非実稼働領域の両方を停止する必要があります。
 2. WUI サーバーを保守ポイント CMAS (MP CMAS) に直接接続して、それら両方が最新の CICSplex SM リリース・レベルになるようにします。この構成により、WUI サーバーは最新のリリース・テーブルと WUI ビューを使用するので、アップグレード手順が簡単になります。

リリース環境が混在した状態で実行する場合、複数の WUI サーバーが必要になることがあります。その場合でも、このアドバイスは引き続き適用されます。1 つの WUI サーバーを最新のリリース・レベルにして最新リリースの CMAS に直接接続し、必要な場合には、他の以前のリリースの WUI サーバーを適切なリリースの CMAS に接続します。
 3. MP CMAS は、別の CMAS か WUI サーバーにのみ接続します。MAS が MP CMAS に直接接続されている場合、この構成は、障害が発生した場合に MP CMAS が容易に別の LPAR に移動してしまうことを防止します。
 4. 単一の MP CMAS を使用することで、アップグレード時の複雑さを回避します。例えばこの方法により、アップグレード時に CMAS が CICSplex の管理から分離されることにつながる CICSplex SM リリース・レベルの競合の可能性を回避できます。

手順

1. CICS 領域をシステム・グループにグループ化する方法を決定して、どの CICSplex が必要であるかを指定します。

CICSplex は、CICSplex SM 構成で管理できる最大の単位です。
2. CICSplex ごとに必要な CMAS の数と、それらをリンクする方法を決定します。

CMAS は、CICS 領域およびそれらのリソースの管理およびレポート作成に関係するほとんどの作業を担当する CICSplex SM トポロジーのコンポーネントです。各 CICSplex には、少なくとも 1 つの CMAS が必要です。
3. 必要な WUI サーバーの数、およびそれらを適切な CMAS にリンクする方法を決定します。
4. CICSplex SM コンポーネントの命名規則を決定します。

命名規則は、有意義で拡張可能なものにする必要があります。
5. インプリメンテーション・ストラテジーを決定します。

タスクの結果

これらのステップに従うと、企業システムの詳細なトポロジー・マップが作成されます。

次のタスク

企業システムのマップを完成させると、CICSplex SM のインストールを計画して、計画している構成を作成する準備ができます。トポロジーにシステム・グループまたは領域を追加したかトポロジーを変更したときには必ずマップを最新にしてください。なぜなら、マップが正確であれば、CICSplex SM 構成およびトポロジー・データの保守が容易になるからです。

CICSplex の設計

CICSplex は、CICSplex SM 構成で操作できる最大の単位です。CICSplex は、CICS システムと CICS システム・グループの関連で構成されます。このセクションでは、CICS システムをシステム・グループにグループ化する方法、およびグループ化した後に CICSplex を指定する方法を決定する上でのガイドを記載します。

CICS システムおよび領域の特定

CICSplex SM をインストールする計画を立てる際の最初のアクションは、企業内にある CICS システムまたは領域を特定することです。企業内にどのようなシステムがあって、それらがどこにインストールされているかについての全体像を既にはっきりと理解しているかもしれません。しかし、CICS システムの数が何百もあるような大規模な企業では、1 人の個人がこの全体像を完全に把握することは不可能です。

目標は、現在の CICS システムの配置をグラフィカル形式で文書化することです。作成する「マップ」は、CICS システムの物理的表記ではなく、論理的表記にする必要があるため、特定のプロセッサが存在する場所などの詳細を記録する必要はありません。グラフィックス・ツールを使用するなどしてマップを記録する場合、この演習の作業を進めるにあたり、CICSplex SM コンポーネントをマップに追加して更新できるように、必ず十分のスペースを残しておいてください。

企業 CICS システムの初期マップには、CICS がインストールされているすべての稼働環境を含めます。さらに以下の情報も示します。

- 使用中のオペレーティング・システムの現行バージョンとリリース
- 各環境で稼働する CICS システム、およびそれぞれの CICS プラットフォーム、バージョン、およびリリース
- 各 CICS システムのリソース・マネージャーの役割 (該当する場合)

この詳細すべてをマップに収めることができない場合は、それを CICS システムのグラフィカル表現とは別に記録してください。

27 ページの図 10 は、目指すべきマップのタイプの例を示しています。

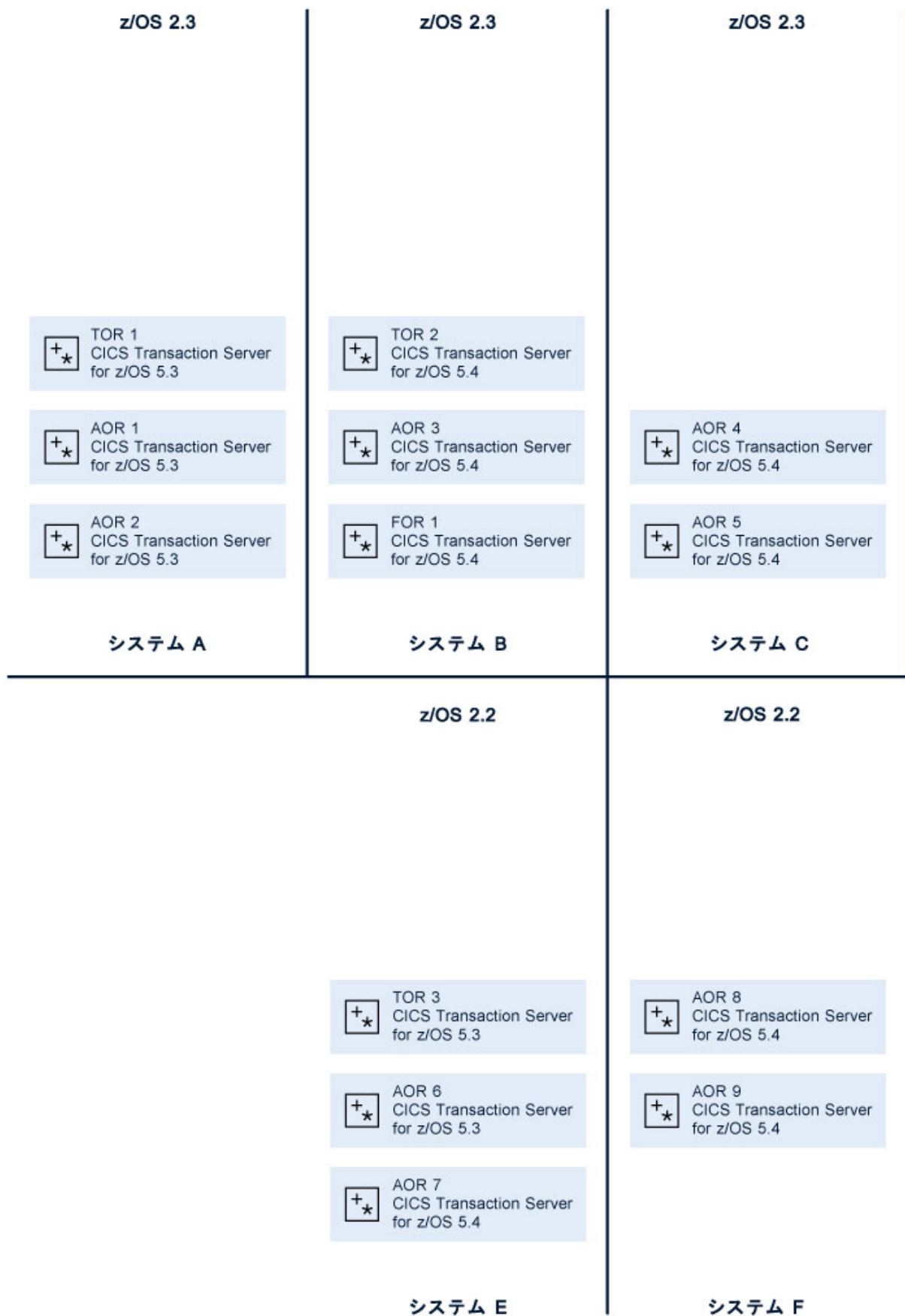


図 10. 企業 CICS システムのマップ

CICSplex SM は、サポートされるすべての CICS リリースを管理できます。CICS システムが管理対象アプリケーション・システム (MAS) になります。

企業システムに存在する CICSplex の数の決定

企業システム内で CICSplex SM で管理できる CICS システムまたは領域を特定したら、次の作業として、CICSplex SM に対して定義する CICSplex の数と、各 CICSplex にどの CICS システムが属するかを決定します。

任意の数の CICSplex を定義できます。例えば、以下のそれぞれに対して CICSplex を 1 つ定義できます。

- 企業全体
- 各地理的場所
- 企業内の各事業単位
- 既存の各 TOR-AOR-FOR 構成
- プロセッサ

ワークロード管理機能を使用する予定がない場合には、CICSplex を形成するために CICS システムと CICS システム・グループを結合させる方法に制限はありません。例えば、CICS システムの関連付けは以下の基準で分類できます。

- CICSplex 内の地理的領域
- 機能。アプリケーション専有領域 (AOR)、ファイル専有領域 (FOR)、または端末専有領域 (TOR) のいずれかであるすべての CICS システムなど
- アプリケーション。特定のアプリケーションまたはアプリケーション・グループによって使用される、AOR、FOR、および TOR として機能する CICS システムなど
- 時間枠。日中または夜間の特定の時間帯に通常アクティブになるすべての CICS システムなど

ワークロード管理機能を使用する計画である場合、以下の点を守る必要があります。

- 特定のワークロード処理のターゲットとして作動する各 CICS システムは、ルーティング領域として作動する CICS システムと同じ CICSplex に配置する必要があります。(CICSplex に関連付けられるルーティング領域とターゲット領域の配置場所は、同じ MVS イメージ上であっても異なる MVS イメージ上であっても構いません。)
- ルーティング領域として作動する CICS システムは、CICS TS システムでなければなりません。

BAS の論理スコープ、リソース管理、またはインストール機能を使用する計画である場合、ビジネス・アプリケーションを 1 つの CICSplex の中に保持してください。

CICS BTS を使用する計画である場合、1 つの BTS セットを 1 つの CICSplex と一緒に保持する必要があります。

問題は、行う作業をどのように決定するかです。定義する CICSplex の数を決定する明確な規則はありませんが、企業にとって最適な構成を選択する上で役立ついくつかの指針があります。これらの指針は、次の 3 段階のプロセスの形式で示されています。

1. 企業全体に単一の CICSplex を定義することを決定する。
2. 単一の CICSplex ではふさわしくない理由を探す。
3. 決定を確認する。

ステップ 1: 企業全体に単一の CICSplex を定義する

大多数の企業では、単一の CICSplex を定義することが自分たちのシステム管理目標に最適であると判断しています。

CICSplex は、CICSplex SM が管理できる最大の単一エンティティです。CICSplex SM の定義または仕様のいずれも、CICSplex 境界をまたがる内容にすることはできません。さらに、CICSplex 同士は相互に排他的です。CICS システムが複数の CICSplex に属することはできません。そのため、企業が単一の CICSplex を定義することにはいくつかの利点があります。例:

- BAS、WLM、RTA、モニター仕様、およびモニター定義を共用および再利用する機会が最大になります。

- CICSplex SM ワークロード・ルーティング機能を使用するときの CICS ワークロード管理の柔軟性が最も高まります。
- 企業全体の単一システム・イメージ (SSI) が得られます。さらに CICS オペレーターは、単一の CICSplex SM ウィンドウから企業 CICS リソースの全体像を知ることができます。(どの CICSplex SM ウィンドウにおいても、単一のコンテキストからのデータ (つまり単一の CICSplex) しか表示できないため、CICSplex が複数ある環境では複数のウィンドウを表示する必要があります。)

要約すると、CICSplex が 1 つであるということは、企業 CICS システムのあるグループと別のグループの間のシステム管理上のバリアがないことを意味します。

ステップ 2: 単一の CICSplex ではふさわしくない理由を探す

単一の CICSplex ソリューションは、必ずしもすべての企業で最適な手法とはなりません。実装が不可能な場合、またはシステム管理上の他の目標と調和しない場合が、その理由として挙げられます。

最初の点として、企業の編成が単一 CICSplex に適しているかどうかを考慮する必要があります。例えば、プロセッサが地理的に異なる場所にある場合、それらのプロセッサ間に接続があるかどうか、またはそれらがそれぞれ独自のワークロードを持つ別個のエンティティとして管理されているかなどを考慮します。このような別個の単位が企業にある場合には、複数の CICSplex を定義して、企業 CICS システムが複数の企業に属している場合のようにして管理する必要があります。

同様に、企業が複数の別個の業務単位に編成され運営されているかについても考慮します。例えば、さまざまなお客様にコンピューター・サービスを提供する事務局を運営している場合、たとえ単一のプロセッサ内であっても、ある領域の集合を別の集合と完全に分離することで、セキュリティ管理、お客様への課金、ワークロード管理などの他の処理が簡素化される可能性があります。同様の理由で一部の領域の管理を他の領域の管理から完全に分離しておく場合、1 つではなく複数の CICSplex を定義することを検討してください。

複数の CICSplex を定義する必要があると決定した場合、おそらく、既にそれぞれの CICSplex に属する CICS システムが明確になっているはずです。明確になっていない場合は、複数の CICSplex を定義するという決定を再検討してください。複数の CICSplex を定義することは人工的なバリアを築こうとすることを示唆しているためです。最終確認として、領域の分離方法が原因でシステム管理上の他の目標の達成が妨害されることがないようにしてください。例えば、CICSplex SM WLM 機能を使用する場合、ルーティング領域と、それらによるトランザクションのルーティング先のターゲット領域の両方が同じ CICSplex に属する必要があります (ただし提供されている動的ルーティング・プログラムをカスタマイズする計画がある場合を除く)。

ステップ 3: 決定を確認する

CICSplex を 1 つにするか複数にするかの決定をしたときには、次のような他の考慮事項に照らして決定事項を確認してください。

- CICSplex の編成は企業の組織を反映していますか? 企業が複数の独立した単位として構成されている場合、複数の CICSplex を定義した方が良いでしょう。企業が単一の事業体として構成されている場合は、単一 CICSplex ソリューションの方が適していると考えられます。
- 決定事項は、企業の業務または情報システムのいずれかにおいて、企業計画と矛盾しませんか? 例えば、現在企業が複数の別個の事業体として運営されている場合、それらを統合する計画はありますか?
- 提案された構成は、できるかぎり単純なものですか? 例えば、CICSplex が 2 つあればシステム管理の目標を達成できるのに 4 つを定義しようとしていますか?
- 複数の CICSplex を計画している場合、目標を達成するために CICS システム・グループを使用できるかどうかを考慮しましたか? 複数の CICSplex とは異なり、CICS システム・グループは相互に排他的ではありません。これは、複数の CICSplex を定義する理由によっては利点ともいえますし欠点ともいえます。

決定は変更可能であることを覚えておってください。ただし、最初の試行時に最良の構成が見つけれることが理想的です。しかし、しばらくして別の CICSplex 構成の方が良いと判断した場合、必要な変更を行えます。

30 ページの図 11 で、サンプル企業は 2 つの離散的単位で運営されています。最初の 3 つの MVS イメージは、残りの 2 つとは異なるワークロードを処理します。これらの 2 つのグループの間ではリソースを共用しません。したがって、企業 CICS システムは 2 つの CICSplex の間で分割されます。

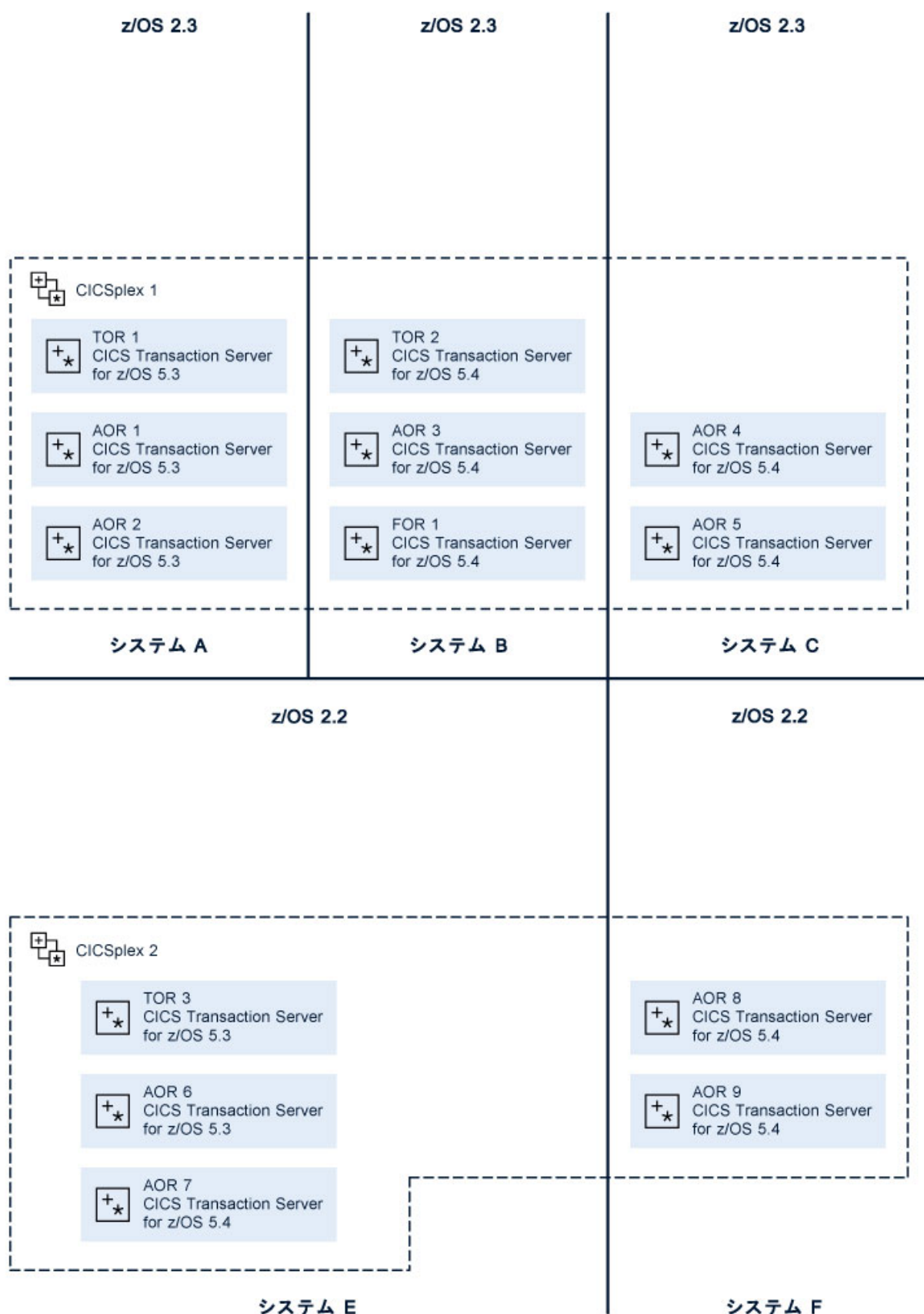


図 11. サンプル企業での CICSplex の識別

システム・グループの指定

CICSplex 内に含まれる CICS システムの 1 つ以上のサブセットを CICS システム・グループとして指定できます。これは、単一エンティティーとして、CICSplex 内の他の部分とは独立して操作できます。

例えば、TOR、AOR、および FOR の CICS システムで構成される CICSplex を定義する場合に、AOR は CICS システム・グループとして定義して、そのカテゴリに含まれるすべての CICS システムへの変更またはそこからのデータ要求を、単一の CICSplex SM コマンドを使って行えるようにすることもできます。

あるいは、以下のいずれかのために単一グループを定義することもできます。

- とりわけ負荷が大きい CICS システム
- CICSplex に含まれる CICS システムの中で、他の CICS システムとはセキュリティ要件が異なるもの
- 特定のアプリケーションが実行される CICS システム

複数の CICSplex とは異なり、CICS システム・グループは相互に排他的である必要はありません。CICS システムは、CICSplex 内に含まれる任意の数のグループに属することができます。ただし、CICS システム・グループは CICSplex のサブセットであるため、システム・グループは CICSplex の境界をまたぐことはできません。

30 ページの図 11 の構成例では、推奨される CICS システム・グループとして、以下の例があります。

- CICSplex 1
 - グループ 1: TOR 1 および TOR 2
 - グループ 2: AOR 1 から AOR 5
 - グループ 3: FOR 1
 - グループ 4: システム A 上のすべての CICS システム
 - グループ 5: システム B 上のすべての CICS システム
 - グループ 6: システム C 上のすべての CICS システム
 - グループ 7: グループ 4 およびグループ 5
 - グループ 8: CICSplex 1 に含まれるすべての CICS システム
- CICSplex 2
 - グループ 1: TOR 3
 - グループ 2: AOR 6 から AOR 9
 - グループ 4: システム E 上のすべての CICS システム
 - グループ 5: システム F 上のすべての CICS システム
 - グループ 6: グループ 4 およびグループ 5
 - グループ 7: CICSplex 2 に含まれるすべての CICS システム

CICSplex 1 のグループ 7 および CICSplex 2 のグループ 6 は、他のグループで構成されている点に注目してください。グループ内にグループを定義すると、この作業を行うユーザーにとっても (労力が少なくなるため)、CICSplex SMにとっても非常に効率的になります。

CICSplex 1 のグループ 8 と CICSplex 2 のグループ 7 は、それが属する CICSplex と同じ一連の CICS システムで構成されます。通常これらのグループを定義すると使いやすくなります。なぜなら、スコープ値 (例えばモニター仕様などのために指定される) は、CICS システムまたは CICS システム・グループの名前しか使用できず、CICSplex の名前にはできないためです。

これは、システム・グループの単なる初期リストです。BAS、WLM、RTA、およびモニター要件が指定されるときにこれに追加または変更される可能性があります。

グループ内部のグループ

CICS システム・グループは他のグループから作成できます。例えば、CICSplex 内のすべての AOR およびすべての TOR で構成される単一グループを必要とする場合、そのメンバーを以下のように定義できます。

- すべての AOR で構成される CICS システム・グループ

- すべての TOR で構成される CICS システム・グループ

このようにして発生する CICS システム名の重複 (例えば、ある特定の CICS システムが、構成要素となる複数のグループに属する場合) は、CICSplex SM によって調整されます。ある CICS システム・グループが CICSplex SM コマンドのターゲットである場合、そのグループに複数回出現する CICS システムは 1 回しか使用されません。

CMAS の配置

CMAS は、システムおよびそれらのリソースの管理およびレポート作成に関係するほとんどの作業を担当する CICSplex SM トポロジーのコンポーネントです。

SSI をオペレーターに示す作業を担当するのは CMAS です。各 CICSplex は、少なくとも 1 つの CMAS によって管理されます。このセクションでは、CMAS を置く場所と定義する数を決定する上でのガイドを記載します。

CMAS をインストールする場所

どの CICS システムを CICSplex SM で管理するか、およびそれらをどのように CICSplex に編成するかを決めた後で、CMAS が必要となる場所について考える必要があります。

CMAS をインストールできる場所およびインストールする必要がある場所を左右する規則および推奨事項は、以下のとおりです。

- 各 CICSplex は、少なくとも 1 つの CMAS で管理する必要があります。
- 各 CICSplex には保守ポイント CMAS を定義する必要があります。
- CMAS は複数の CICSplex の管理に関与できます。
- CMAS 内の SDFH* ライブラリーおよび SEYU* ライブラリーは同じレベルである必要があり、CMAS は、実行している CICSplex SM のレベルに適したものでなければなりません。[CICSplex SM のアップグレード](#)を参照してください。
- CICSplex SM で SNA 総称アラートを NetView インスタンスに発行する場合、NetView インスタンスと同じ MVS イメージに CMAS をインストールする必要があります。
- 管理対象 CICS システムが稼働している MVS イメージごとに CMAS を 1 つずつインストールします。
- 1 つの MVS イメージに複数の実動 CMAS を定義することは可能ですが、必要になることはほとんどありません。

さらに、決定事項のパフォーマンスへの影響も考慮する必要があります。各 CMAS には独自のスペース所要量と独自のデータ・リポジトリがあります。

これらの規則および推奨事項をサンプル企業に適用する場合、次の点を確認できます。

- 企業に少なくとも 1 つの CMAS を定義する必要がある。
- NetView がシステム C にインストールされている。SNA アラートをその NetView インスタンスに送信するには、同じ MVS イメージ上に CMAS をインストールする必要があります。

33 ページの図 12 は、これらの CMAS 規則と推奨事項の適用を示すように更新されたマップの例を示しています。MVS イメージごとに 1 つの CMAS がインストールされます。CMAS 4 は、同じ MVS イメージ上にある NetView インスタンスに SNA 総称アラートを発行できます。CMAS 4 は CICSplex 1 の保守ポイント CMAS (MP CMAS) で、CMAS 7 は CICSplex 2 の MP CMAS です (つまり、CMAS 4 は CICSplex 1 を定義するときにコンテキスト CMAS になり、CMAS 7 は CICSplex 2 を CICSplex SM に定義するときにコンテキスト CMAS になります)。

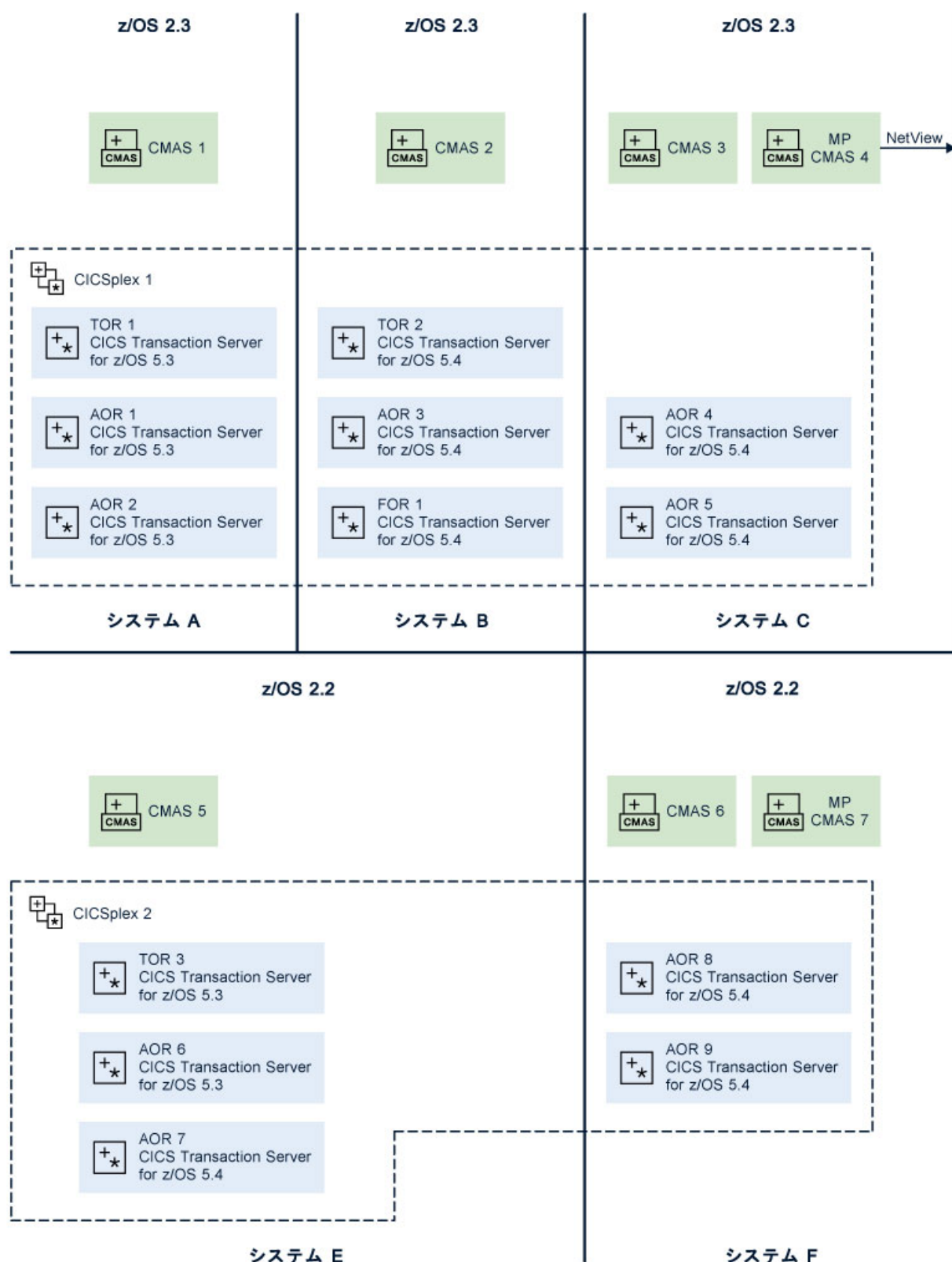


図 12. エンタープライズ・マップの例への CMAS の追加

CMAS 間リンク

CICSplex が複数の MVS イメージをまたぐ場合、複数の CMAS がその CICSplex の管理に関係している可能性があります。それらの CMAS は (CICS 相互通信方式を使用して) 互いに接続する必要があります。その

ように接続すると、その CICSplex に属する CICS システムに関連するデータは関係するすべての CMAS でアクセスできるようになり、単一システム・イメージ (SSI) を CICS オペレーターに表示できるようになります。

最小必要要件は、同じ CICSplex を管理している CMAS がリンクされて CMAS の開いたチェーンを形成し、間接的にのみであっても各 CMAS が他のすべての CMAS にリンクされていることです。この要件を満たすことにより、各 CMAS から出て、同じ CICSplex の管理に関係する他のすべての CMAS に到達するパスが少なくとも必ず 1 つ存在することになります。最良のパフォーマンスおよび可用性を達成するためには、各 CMAS を他のすべての CMAS に直接リンクします。

同じ CICSplex の管理に関係する複数の CMAS 間のリンクは、その CICSplex に属する CICS システムまたはその他のシステムの SSI を担当します。ただし定義するリンクの数に関しては、この作業を行うユーザーにある程度の選択の自由があります。最初に、CMAS 間リンクに関する規則および推奨事項の覚え書を以下に記します。

- CICSplex の SSI を確立するには、その CICSplex を管理している CMAS が、少なくとも開いたチェーンを形成するようにリンクされている必要があります。
- リンクを増やせば増やすほど、パフォーマンスは良くなります。すべての CMAS が、単一の CICSplex の管理に関係する他のすべての CMAS に直接接続されているときにパフォーマンスは最大になります。
- CMAS 間リンクの多重化も可用性の向上につながります。例えば CMAS 1 がチェーンの末端にあり、CMAS 2 にしか接続されていない場合、CMAS 2 で障害が発生するとチェーンが切断され、CMAS 1 への接続はなくなります。

34 ページの図 13 は、直接または間接の CMAS 間リンクの概念を例示しています。この CICSplex は 6 つの CICS システムから構成されます。3 つのシステムは CMAS 1 によって管理され、3 つのシステムは CMAS 5 によって管理されます。これら 2 つの CMAS の間に直接リンクがなくても、CICSplex SM はその「ネットワーク」内を動的にナビゲートして、CICSplex に関連した情報を収集できます。これは CMAS 4 の中を進むことができますし、CMAS 4 を使用できない場合は CMAS 2 および CMAS 3 の中を進むことができます。しかし最適なパフォーマンスを得るためには、CMAS 1 と CMAS 5 の間に直接リンクを追加する必要があります。

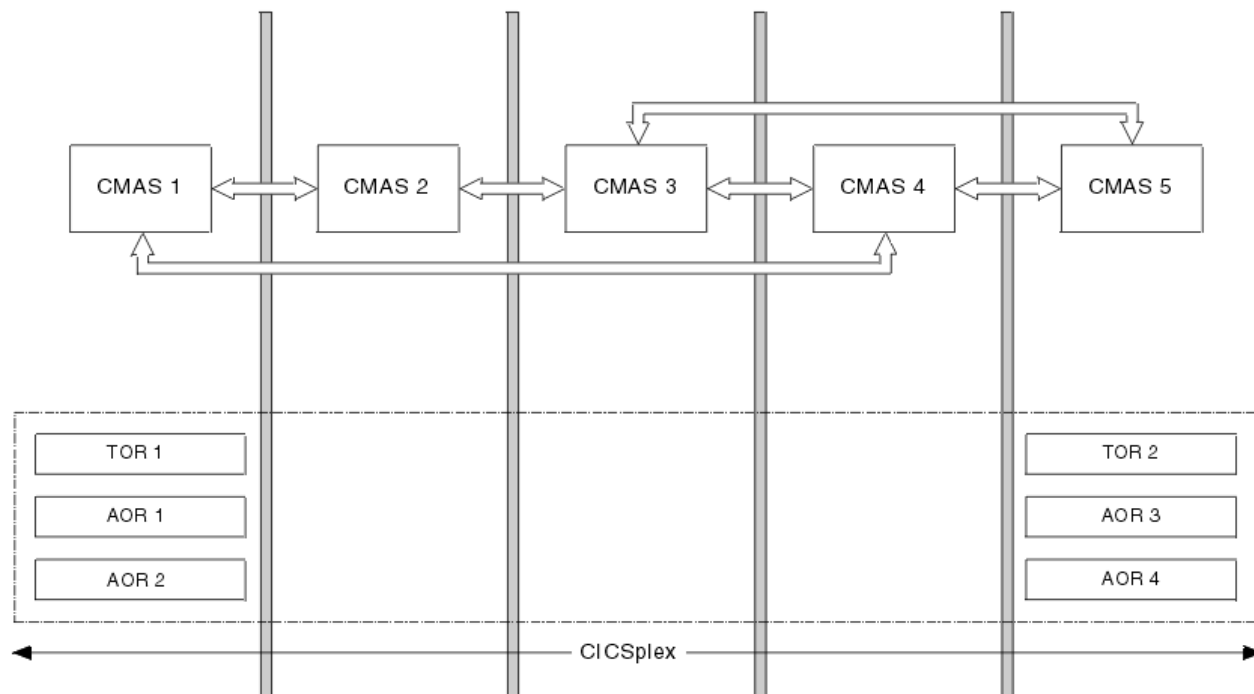


図 13. CICSplex での CMAS 間リンク

36 ページの図 14 のエンタープライズ構成の例では、2 つの CICSplex のそれぞれで、CMAS 間の完全な接続性が確立されます。つまり、CICSplex 1 の管理用に、CMAS 1、2、3 および 4 が互いに直接接続されています。CICSplex 2 の管理用に CMAS 5、6、および 7 が互いに直接接続されています。そのため、次の CMAS 間リンクを定義する必要があります。

CICSplex 1

CMAS1 と CMAS2 の間
CMAS1 と CMAS3 の間
CMAS1 と CMAS4 の間
CMAS2 と CMAS1 の間
CMAS2 と CMAS3 の間
CMAS2 と CMAS4 の間
CMAS3 と CMAS1 の間
CMAS3 と CMAS2 の間
CMAS3 と CMAS4 の間
CMAS4 と CMAS1 の間
CMAS4 と CMAS2 の間
CMAS4 と CMAS3 の間

CICSplex 2

CMAS5 と CMAS6 の間
CMAS5 と CMAS7 の間
CMAS6 と CMAS5 の間
CMAS6 と CMAS7 の間
CMAS7 と CMAS5 の間
CMAS7 と CMAS6 の間

CMAS 1 を CMAS 3 に直接リンクしないことに決めた場合、この配置は (CMAS の開いたチェーンがあるので) CMAS 間リンクの最小要件を満たし、定義する必要のあるリンクの数は 2 つ少なくなります。一部の情報は間接的に取得されます。例えば、システム C から要求された、システム A の CICS リソースに関する情報は、隣接する CMAS である CMAS 2 から間接的に取得されます。CMAS のグループの中で完全な接続性が確立されなかった場所では、CICSplex SM は要求データへの最短経路を動的に確立します。

間接的に情報を取得する場合、パフォーマンスへの影響は少し大きくなりますが、これは、CMAS 間リンクのセットアップと保守に必要なオーバーヘッドとのバランスを考慮する必要があります。実際のところ、エンタープライズ構成の例で 2 つではなく 1 つの CICSplex を設定した場合でも、CMAS 間の完全な接続性は実現可能となります。つまり、42 の CMAS 間リンクが必要になります。なぜなら、接続する CMAS の数を n とすると、必要なリンクの数は $n^2 - n$ となるからです。ただし、単一の CICSplex を管理する 10 個の CMAS がある場合に必要なリンクの数は 90 個になり、15 個の CMAS がある場合には 210 個のリンクになります。そのため、いくつかの直接リンクを定義することができるか、およびどこで間接リンクを受け入れられるかは、最小要件が CMAS 間リンクの開いたチェーンの確立であることを常に思いに留めながら決定する必要があります。

36 ページの図 14 は、エンタープライズ構成の例における CMAS 間リンクを示しています。18 個のリンクが定義されていて、同じ CICSplex を管理する CMAS 間に完全な接続性があります。

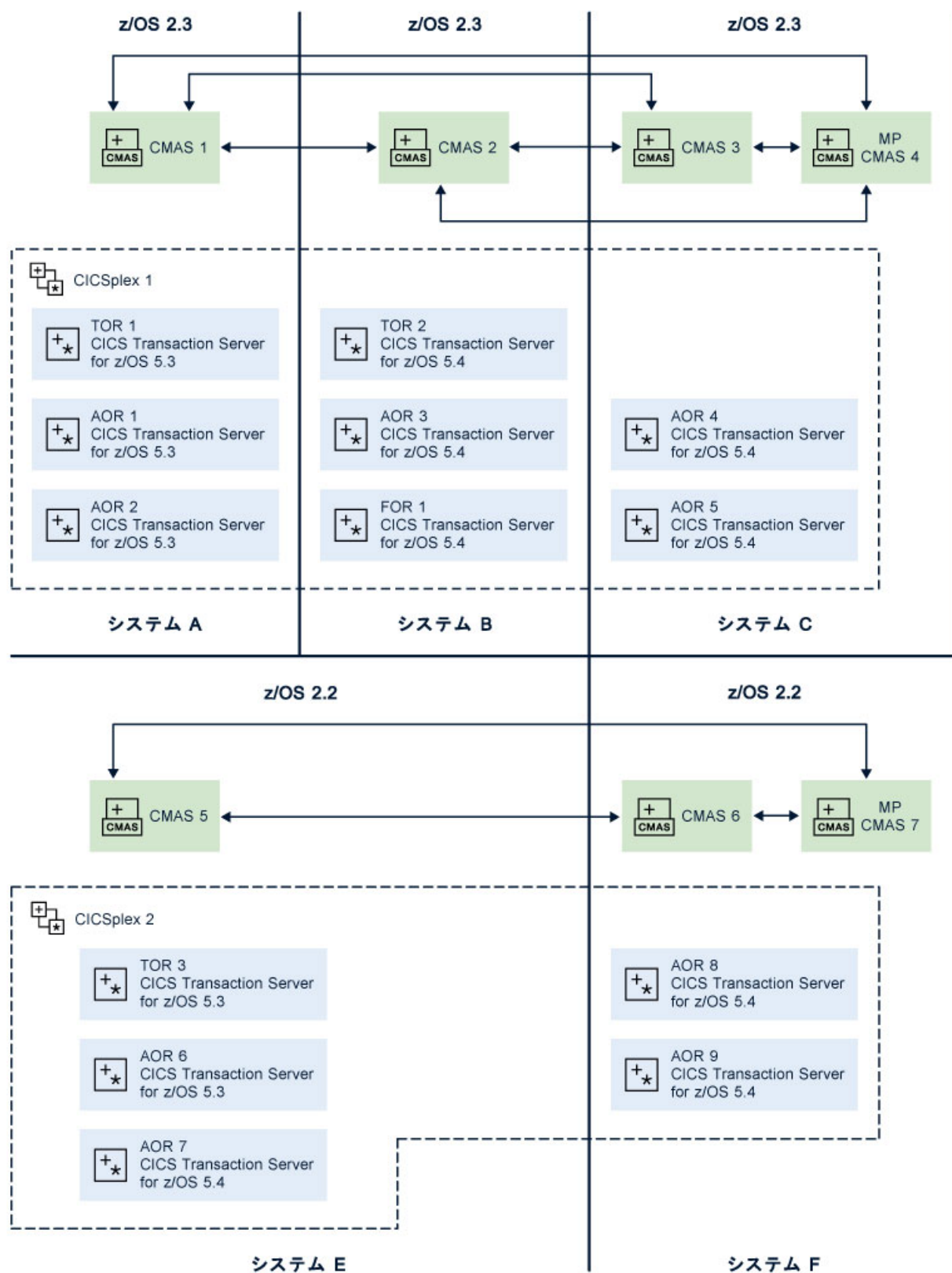


図 14. エンタープライズ・マップの例への CMAS 間リンクの追加

同じ MVS システムに、同じ CICSplex を管理する複数の CMAS がある場合、CICSPLEX(name) CICSplex SM システム・パラメーターのみを指定し、CMASSYSID(name) CICSplex SM システム・パラメーターは指定

しないローカル MAS は、特定のリリースの CICSplex SM 用の指定された CICSplex を管理する MVS イメージ上の、最後に初期化する CMAS に接続されます。

複数の CICSplex における CMAS 間リンク

エンタープライズ構成の例には 2 つの CICSplex があります。CICSplex SM オペレーターがシステム B から CICSplex 2 のデータにアクセスできるようにするため、例えば、CICSplex 1 の管理に関係する 1 つ以上の CMAS と、CICSplex 2 の管理に関係する 1 つ以上の CMAS の間のリンクを確立することができます。例えば、CMAS 1 から CMAS 5 へのリンクを確立することによって企業の CMAS すべてのチェーンを形成することができます。

保守ポイント CMAS の配置

単一の CICSplex が複数の CMAS によって管理されている場合、そのいずれかが保守ポイント CMAS (MP CMAS) に指定されます。

MP CMAS は、CICSplex に関連するすべての CICSplex SM 定義の保守と、同じ CICSplex の管理に関するその他の CMAS への最新情報の配布 (データ・リポジトリが互いに一致するようにするため) を担当します。

MP CMAS は、CICSplex が CICSplex SM に定義されるときにコンテキスト値である CMAS です。各 CICSplex に MP CMAS は 1 つしか設定できませんが、1 つの CMAS が複数の CICSplex の MP CMAS になることは可能です。MP CMAS を常に使用できない場合、どの CICSplex SM 定義も変更できません。

MP CMAS は、他の CICSplex の保守ポイントではない CMAS としても機能できます。

注：CICSplex SM に単一制御ポイントがあるということは、CICSplex SM WUI を使用可能な任意のシステムから MP CMAS の変更を行えることを意味します。

MP CMAS は、他の CMAS や Web ユーザー・インターフェース (WUI) サーバーにのみ接続することが理想的です。MP CMAS を他の CICS 領域に接続しないでください。この構成により、MP CMAS は、MVS イメージ間で簡単に移動することや他の領域に影響を与えずに保守を適用することが可能になります。

WUI サーバーの場所についての計画

社内での WUI サーバー領域の場所と数は、使用可能性要件とグローバリゼーション・サポートによって異なります。

始める前に

WUI サーバー領域のインストール場所について計画を立てる前に、CMAS のインストール場所とそれら CMAS のリンク方法について計画を立てる必要があります。

このタスクについて

WUI サーバーは CICSplex SM アプリケーションとして作動する CICS 領域で、API を使用して CMAS のデータ・リポジトリ内のオブジェクトを表示および管理します。

手順

1. すべての MVS イメージで WUI サーバーを使用可能にするかどうかを決定します。
MVS イメージごとに 1 つの WUI サーバーがある場合、それぞれの MVS イメージで CMAS に直接 WUI サーバーを接続する必要はありません。
ヒント：WUI サーバーに対して別の CICSplex を定義すると、社内での CMAS 保守ポイント数を最小に保てます。またそれにより、CICS アプリケーションを実行している CICSplex から生成される統計と、WUI サーバーを分離することにもなります。
2. WUI がサポートする必要のある各国語を決定します。
複数の言語で WUI を表示する場合、言語ごとに WUI サーバーが必要です。
3. 各 WUI サーバー領域から CMAS に必要な接続を決定します。

- WUI サーバーの接続先の CMAS は、WUI サーバーがアクセスを必要とする CICSplex すべてを管理しなければなりません。ただし、WUI が接続する先の CMAS では、これらの CICSplex 内の MAS を管理する必要はありません。
 - 接続する CMAS と WUI サーバーは、同じリリース・レベルの CICS Transaction Server になければなりません。
4. トポロジー・マップを更新し、WUI サーバー領域を含めます。

例

システム F では、WUI サーバーが保守ポイント CMAS 7 に接続していますが、CICSplex 4 という別の CICSplex に配置されています。この WUI サーバーと CMAS 7 はどちらも同じレベルの CICS Transaction Server です。CMAS 7 は CICSplex 2 と CICSplex 4 の両方の保守ポイントです。

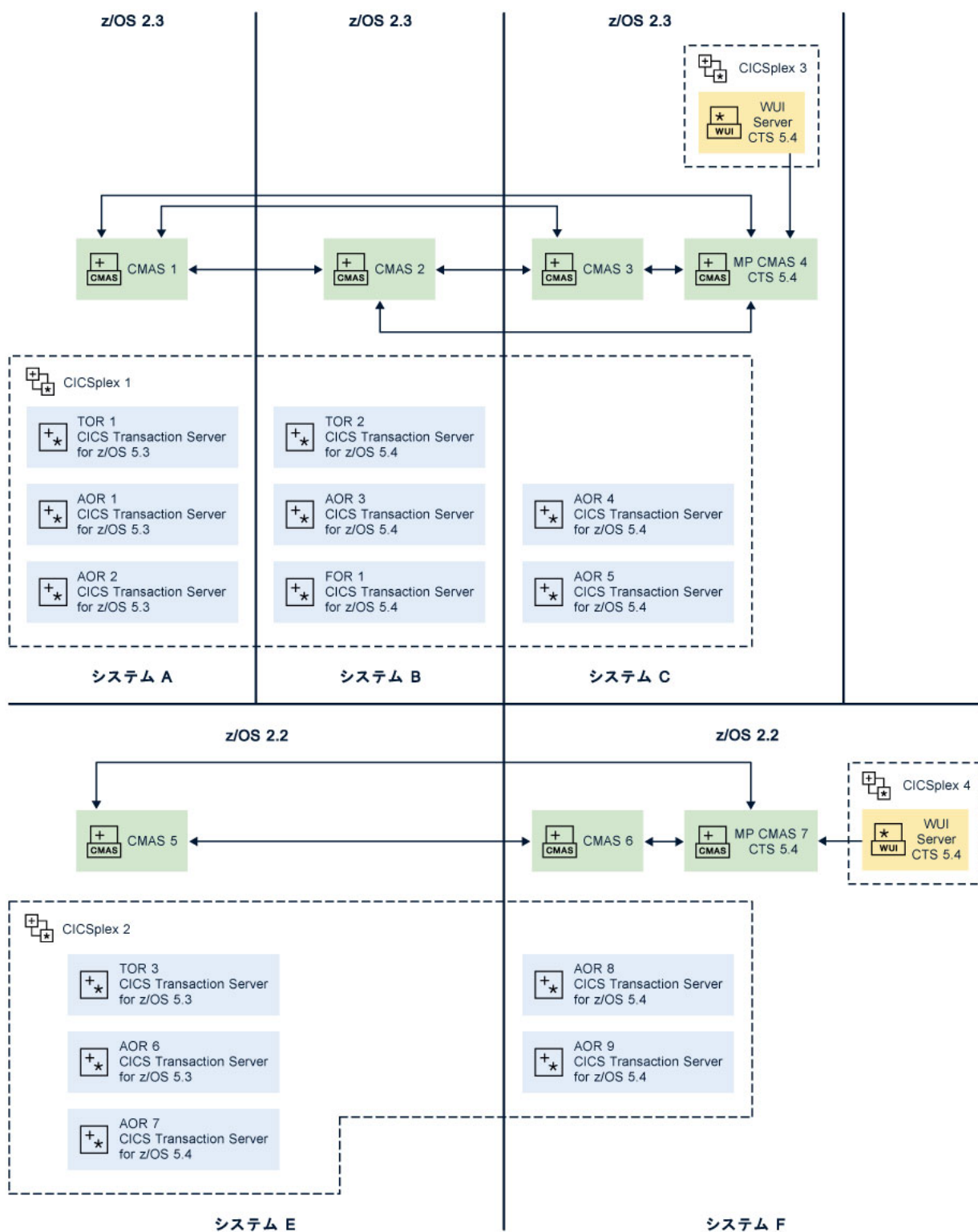


図 15. WUI サーバーのマップ例への追加

CICSPlex SM エンティティの命名

CICSPlex SM マップに入力するすべてのエンティティに名前を指定する必要があります。これらのエンティティに命名するための意味のある拡張可能な規則を考案する必要があります。

CICSPlex SM エレメントの命名に適用される規則には、以下のものがあります。

- 各エレメント名の長さは 8 文字以下です。
- エレメント名は英字 (または国別文字) で始まる必要があります、埋め込みブランクは許可されません。後続文字は、英字または数字です。
- CICSplex と CMAS の名前は、企業内で固有でなければなりません。
- CICS システムと CICS システム・グループの名前は、CICSplex 内で固有でなければなりません。
- その他のすべての名前は、CICSplex 内でタイプごとに固有でなければなりません。

名前が必要なインスタンスごとのエンティティ・タイプは、以下のとおりです。

- CICSplex SM コンポーネントおよび CICSplex エンティティ
 - CMAS、CICSplex、MAS (CICS システムまたは WUI サーバー)、CICS システム・グループ、および時間枠定義。
- BAS エンティティ
 - リソース割り当て、リソース記述、リソース定義、およびリソース・グループ。
アプリケーション・リソース定義には以下が含まれます。
 - Db2® トランザクション、ファイル、マップ・セット、区画セット、プログラム、一時データ・キュー、トランザクション、CICS BTS プロセス・タイプ、文書テンプレート、FEPI ノード、FEPI プール、FEPI プロパティ・セット、FEPI ターゲット、ファイルおよび鍵ファイルのセグメント定義、LIBRARY 定義、マップ・セット、シスプレックス ENQ モデル、TCP/IP サービス、一時記憶域モデル

領域リソース定義には以下が含まれます。

- Db2 エントリ、エンタープライズ Bean、ジャーナル、ジャーナル・モデル、ローカル共用リソース (LSR) プール、プロファイル、一時記憶域キュー・モデル、トランザクション・クラス、端末、入力条件

システム間接続リソース定義には以下が含まれます。

- 接続、Db2 接続、IPIC 接続、パートナー、セッション

- モニター・エンティティ
 - モニター定義、モニター・グループ、およびモニター仕様
- 分析エンティティ
 - 分析定義、分析グループ、分析ポイント仕様、分析仕様、アクション定義、評価定義、および状況定義
- ワークロード管理エンティティ
 - トランザクション・グループ、ワークロード定義、ワークロード・グループ、およびワークロード仕様。

選択する規則がこれらすべてのエンティティ・タイプに対応できるものとなるようにします。

総称名

CICSplex SM は、そのコマンドの多くにおいて総称名の使用をサポートします。例えば、名前が文字「DNW」で始まるすべてのトランザクションが 300 秒ごとに一度モニターされるように指定できます。各トランザクションを個別に指定する必要はありません。

総称名の使用に適用される規則は、以下のとおりです。

- 1 つ以上の文字の代わりにアスタリスク文字 (*) を使用できます。この文字は、名前の中で一度だけ使用でき、部分的なストリングの末尾で使用する必要があります。例えばストリング「DNW*」は、文字 DNW で始まるすべての名前という意味です。
- 任意の単一文字を正符号 (+) に置き換えることができ、名前の任意の位置で使用できます。例えば「DNW+++L」は、文字 DNW で始まり文字 L で終わる 7 文字の任意の名前という意味です。

段階的な実装

CICS システムが非常に多数ある場合、または複数の CICSplex を作成することにした場合、複数の段階に分けて CICSplex SM の実装を行うこともできます。

システムのサブセットに CICSplex SM をインストールし、それを使用して 1 つの CICSplex を管理するか、または自社用に 1 つの CICSplex を定義した場合にはそれを使用して特定のシステム・グループを管理できます。その構成に別の CICS システムを徐々に追加できます。

漸進的な方法を選ぶ場合、社内の CICS システムの最終的なマップを再び参照し、どのシステムから開始する予定かを識別する必要があります。例えば、段階的な実装をマップ例で採用した場合、CICSplex 2 のみの実装から開始できます。CMAS 間のリンク数は大いに減りますが (CMAS 間のリンクは 18 ではなく 6 になる)、CICSplex 2 に関して行われた他の決定は依然として有効です。つまり、同じ CICS システム・グループを定義し、CMAS インストールが引き続き必要です。

第 3 章 CICSplex SM の計画のセットアップ

CICSplex SM は、製品の一部としてインストールされています。CICSplex SM をセットアップするには、必要なセキュリティと、CICSplex で使用するタイム・ゾーンを計画する必要があります。

現在の環境で CICSplex SM を実装するには、CICS 起動時に提供されたサンプルを使用することができません。

CICSplex SM のセキュリティ計画

CICSplex SM は SAF 準拠の外部セキュリティ・マネージャー (RACF® など) を使用して、CICSplex SM の機能や CICS リソースへの無許可アクセスを防止し、CICS コマンド検査および CICS リソース検査のシミュレーションを制御します。

いずれの場合においても、セキュリティ検査は、リソースにアクセスするための要求のターゲットである CICS システムを管理する CMAS によって処理されます。例えば、ある CICSplex が 2 つの CMAS によって管理され、その CICSplex に属するすべての CICS システムのリソースにアクセスする要求が出された場合、両方の CMAS でセキュリティ検査が実行されます。

セキュリティ検査をアクティブにするには、CMAS またはその管理対象 CICS システムを開始するために使用する JCL を変更する必要があります。CICS システムでセキュリティ検査がオフになっている場合、CMAS の設定値に関係なく検査は実行されません。ただし、セキュリティ検査が CMAS ではオフで CICS システムではオンになっている場合、CICS システムは CMAS に接続できません。

初めに、どの程度のセキュリティ検査が必要であるかを決定します。特に、CICSplex SM へのアクセス権限が必要なユーザーを特定し、CMAS がインストールされているすべてのシステムにおいて、個々のユーザーが必ず同じユーザー ID を使用するようにします。セキュリティ検査の実行対象となるユーザー ID は、CICSplex SM へのサインオンに使用された RACF ID です。実施するセキュリティ検査のタイプも考慮してください。

CICSplex SM のセキュリティのセットアップ方法について詳しくは、[Implementing CICSplex SM security](#) を参照してください。

CICSplex SM 機能および CICS リソースへのアクセスの保護

無許可アクセスを防ぐには、保護対象となる CICS リソースと CICSplex SM 機能の組み合わせに対するセキュリティ・プロファイルを作成します。ほとんどの場合、CICSplex SM セキュリティ・プロファイルによって提供されるセキュリティで十分です。

CICSplex SM のライブラリー、プロシージャー、および Web ユーザー・インターフェースのリソースを保護するために、外部セキュリティ・マネージャーも使用されます。CICSplex SM のライブラリーおよびプロシージャーを保護する方法についての詳細は、[Implementing CICSplex SM security](#) に記載されています。Web ユーザー・インターフェースのビュー、メニュー、ヘルプ情報、およびビュー・エディターを保護するため、FACILITY クラス内に適切なプロファイルを作成する必要があります。詳細については、[Web ユーザー・インターフェース・リソースへのアクセス制御](#)を参照してください。

BAS についての特別な考慮事項

BAS ビューの保護については特別な注意を払う必要があることを意識して、無許可ユーザーはリソースを作成および管理できないようにします。RDO 用語では、CSD を無保護にしておくことがこれに相当します。

また、**EXEC CICS CREATE** コマンドを使用して新規リソースを作成する場合には注意してください。コンテキストとして CICSplex で作成されるすべての定義は、CICSplex 内のすべての CMAS に自動的に配布されます。そのため、ユーザーに BAS オブジェクトを作成する権限を付与することは、その CICSplex 内の任意の CICS システムにリソースをインストールする権限を付与することに相当します。CICS システムが開始するとき、だれがシステムにリソースをインストールしたかについての検査はありません。

CICS コマンドおよびリソースの検査

CICS コマンドおよびリソースの検査は、要求が送信される CMAS 内で CICSplex SM によってシミュレートされます。これにより、外部セキュリティ・マネージャーをサポートしない CICS システムを保護することができます。さらに、セキュリティ検査をあるレベルで統合することもできます。

CICS リソースおよびコマンドの検査が有効である場所を判別し、それを他の CICSplex SM セキュリティ検査と一緒に保持する必要があるかどうかを決定します。

時間帯の定義

CICSplex SM のアクティビティの多くは、時間に依存しています。

例えば、モニター定義または分析定義は特定の時間枠にアクティブになるように指定できます。CICSplex SM では、単一の CICSplex 内のすべての MAS が同じ時間帯で実行される必要はないため、エンティティ間の時間帯の相違を調整できなければなりません。そのため、以下のようにする必要があります。

- 時間枠定義を作成 (CICSplex SM の「**時間枠定義**」ビューを使用) するときには常に、定義内で時間帯を指定する必要があります。例えば、0800 から 1159 の東部標準時について、「MORNING」という時間枠定義を作成することができます。
- CMAS のデータ・リポジトリ初期化ジョブ、EYU9XDUT で、CMAS ごとに時間帯を指定する必要があります。CMAS が実行中であるとしても、CICSplex SM Web ユーザー・インターフェースを使用して CMAS 時間帯値に永続的な変更を行うことができます。
- 管理対象 CICS システムごとに時間帯を設定する必要があります。CICS システムを CICSplex SM に対して定義するとき、システムが実行される時間帯を指定できます。あるいは、CICS システム定義で時間帯を指定しない場合、CICS システムは、その接続先の CMAS のデフォルトの時間帯で実行されていると想定されます。管理対象 CICS システムの時間帯のデフォルトが、その CMAS の時間帯になるようにすることをお勧めします。管理対象 CICS システムの時間帯は、後で CICS システムの実行中に変更できます。このようにして行った変更は、CICS システムの存続期間の間、または次に変更が行われるまでのいずれか早い時点まで維持されます。
- 時間帯は、CICSplex ごとに、それを最初に定義するときに指定する必要があります。この時間帯は CICSplex SM モニター機能で使用され、CICSplex のモニター間隔が満了する実際の時刻が判別されます。CICSplex の時間帯は、CICSplex SM エンド・ユーザー・インターフェースを使用して変更できます。

時間帯は、B から Z までの範囲の 1 文字のコードを使用して指定されます。例えば、コード「S」はアメリカ山岳標準時を表し、コード「T」は中央標準時を表し、コード「C」は東ヨーロッパ時間を表します。コードの詳細なリストについては、[期間定義の時間帯設定](#)を参照してください。CICSplex SM では、標準時間帯では実行されていない領域を受け入れるために、オフセット（「時間帯調整」と呼ばれる）を 0 から 59 分の範囲で指定できます。さらに、夏時間調整時刻も指定できます。

複数の CICSplex SM エンティティで時間帯を指定する必要があるため、「矛盾する」時間帯が指定される可能性があることは明らかです。例えば、同じ CICSplex 内の CMAS と MAS で時間帯が異なることは十分考えられます。CICSplex SM では、常に MAS の時間帯が優先されます。例えば次の状況を考えます。

- 時間枠定義の時間帯が S であり、

かつ

- CMAS の時間帯が B であり、

かつ

- MAS の時間帯が C である

MAS によって時間帯 C が使用されるため、CMAS は時間帯 B、C、および S の間で必要な調整を行ってその時間帯が優先されるようにします。

CICSplex SM オブジェクト定義の再利用

CICS は、そのプラットフォームまたはリリースすべてで同一のリソースおよび機能をサポートするわけではありません。

例えば、CICS TS 環境でサポートされる FEPI などのリソースは、他のプラットフォームではサポートされない場合があります。同様に、あるリソースで使用可能なデータの量またはタイプは、CICS プラットフォームおよびリリースごとに異なる場合があります。これらのサポート上の相違は、複数の多様な CICS システムの SSI を提供する CICSplex SM にとっては特別な挑戦となります。

CICS TS システムを含む CICSplex 全体において、CICSplex SM がリソースおよび機能のサポート上の相違を処理する方法を示すいくつかの例を挙げます。

• 例 1: 一時データ・キューのモニター

一時データ・キューのモニター定義を作成し、それをモニター・グループに追加してから、モニター・グループをモニター仕様に関連付けます。モニター仕様のスコープは、その CICSplex 内のすべての CICS システムです。各 CICS システムが開始すると、CICSplex SM はそのシステムにモニター定義をインストールできるかどうかを判別します。できない場合、CICSplex SM は、モニター定義をインストールできなかったことを通知するメッセージを送出します。モニター機能と一時データ・キュー・リソースの両方が CICS TS の下でサポートされるため、CICSplex SM はそれらの環境にモニター定義をインストールし、モニターが開始されます。

• 例 2: モニター・プログラムの RTA

評価定義を作成し、MPROGRAM (モニター・プログラム) を、CICSplex SM RTA 機能によって評価されるリソースとして指定します。評価定義は、分析グループを介して分析仕様にリンクされる分析定義で指定されます。分析仕様のスコープは、その CICSplex 内のすべての CICS システムです。分析定義は CICS TS システムにインストールされます。分析定義で複数の評価定義が指定されている場合、CICSplex SM はターゲット環境でサポートされる定義をインストールします。

要約すると、特定の CICS 環境で使用できない機能を要求した場合、CICSplex SM は警告メッセージを送出します。そのメッセージには応答する必要はありませんし、さらに重要な点として、CICSplex SM オブジェクト定義を作成するときにサポート上の相違を気にする必要はありません。CICSplex SM オブジェクト定義が再使用可能になるように常に考慮する必要があります。

第 4 章 CICSplex SM 構成とトポロジーの定義

CICSplex SM をインストールし、ご使用の CICSplex SM 環境のマップがある場合、構成とトポロジーを定義できます。

CMAS 構成の定義

定義済みの CMAS ごとに、CICSplex SM に特定の情報を提供する必要があります。

以下の情報を提供する必要があります。

- 管理するそれぞれの CICSplex の名前 (保守ポイント CMAS の場合)
- 接続先の各 CMAS のリンク定義

この情報を入力するには、適切な「**CMAS 構成管理**」ビュー (「**CMAS 構成管理**」メニューからアクセスします)、またはバッチ・リポジトリ更新機能を使用します。

CMAS 構成の管理

以下の WUI ビュー・セットおよび関連オブジェクトを使用して CMAS 構成を管理します。

表 2. CMAS 構成を管理するためのビュー・セット		
WUI ビュー・セット	オブジェクト名	説明
バッチ・リポジトリ更新要求 (Batched repository-update requests)	BATCHREP	ローカル CMAS に関連付けられたデータ・リポジトリから 1 つ以上の定義を作成、更新、除去、リスト、またはダンプします。
CMAS から CMAS へのリンク定義	CMTCMDEF	ローカル CMAS と他の CMAS との間の直接 LU6.2 および MRO 通信リンクに関する情報を表示します。
CICSplex の定義 (CICSplex definitions)	CPLEXDEF	ローカル CMAS に CICSplex を定義し、ローカル CMAS の CICSplex 定義を管理します。
CICSplex 定義中の CMAS	CPLXCMAS	CICSplex に関連したすべての CMAS (ローカル CMAS はその CICSplex の保守ポイント) に関する情報を管理します。

CICSplex SM は、関連付けられた CMAS がアクティブであるときに CMAS 構成定義を管理するために使用できる、以下の WUI ビュー・セットおよびリソース・オブジェクトも提供します。

表 3. アクティブ CMAS を管理するためのビュー・セット		
WUI ビュー・セット	オブジェクト名	説明
CICSplex 管理の CMAS	CICSPLEX	ローカル CMAS が認識している CICSplex に関連付けられている CMAS に関する情報を表示します。
ローカル CMAS 認知の CMAS	CMAS	ローカル CMAS が認識している CMAS に関する情報の表示、CMAS のシャットダウン、および CMAS コンポーネント・トレース設定の変更を行います。

表 3. アクティブ CMAS を管理するためのビュー・セット (続き)		
WUI ビュー・セット	オブジェクト名	説明
CMAS 管理の CICSplex	CMASPLEX	ローカル CMAS によって管理される CICSplex に関する情報を表示し、MPSTATE が INVALID または NOTCONNECTED のときは管理側の CICSplex から CMAS を除去します。
CMAS から CMAS へのリンク	CMTCLNKL	ローカル CMAS にリンクされているすべてまたは特定の CMAS に関する情報を表示し、CMAS 間リンクを廃棄します。
CMAS から MAS へのリンク	CMTPLNKL	ローカル CMAS にリンクされているすべてまたは特定の MAS に関する情報を表示し、CMAS から MAS へのリンクを廃棄します。

CICSplex トポロジーの定義

ご使用の CICSplex および MAS を識別した場合、それらを CICSplex SM に対して識別させる定義オブジェクトを作成する必要があります。この段階では、必要なすべての情報が揃っていないことに注意してください。CICSplex SM システムの設計を続行してゆくうちに多くの情報が明らかになります。

CICSplex の設計について詳しくは、[26 ページの『CICSplex の設計』](#)に記されています。

CICSplex 定義の準備

各 CICSplex について、社内で固有の名前で CICSplex SM に定義する必要があります。

命名規則については、[39 ページの『CICSplex SM エンティティの命名』](#)に記されています。また、CICSplex 定義には他の情報も必要となることに注意してください。こうした情報は、CICSplex SM システムの設計を続けるうちに明らかになります。以下の情報が含まれます。

- CICSplex の定義先となる CMAS の名前。(これは、CICSplex の保守ポイント CMAS です。)
- CICSplex の管理に関係するその他の CMAS の名前。
- CICSplex SM モニター機能の時間帯。
- CICS コマンド検査をシミュレートするかどうか。
- CICS リソース検査をシミュレートするかどうか。

CICSplex 定義の管理

「**CICSplex の定義 (CICSplex definitions)**」ビュー (CPLEXDEF オブジェクト) を使用して、CICSplex SM に対して定義する CICSplex を識別します。

.

この情報を入力するには、「**CMAS 構成管理 (CMAS configuration administration)**」メニューからアクセスできる、該当する **CMAS 構成管理**ビューか、バッチ処理されるリポジトリ更新機能のいずれかを使用できます。

CICSplex 定義を作成した後、CICS システム (MAS) および CICS システム・グループを CICSplex に関連付けることができます。[48 ページの『CICS システム定義の計画』](#)を参照してください。

CICS システム定義の計画

CICSplex SM を使用して CICS 領域を管理するには、CICS 領域を、CICSplex SM に定義されている CICSplex に関連付ける必要があります。この関連付けを設定する定義には、CICS システムにおける CICSplex SM の BAS、WLM、RTA、およびモニター・コンポーネントの使用法についての情報も含まれます。

このタスクについて

CICSplex SM によって管理するすべての CICS 領域について、以下の情報を収集する必要があります。

手順

1. 定義を作成するための CICS 領域に関する基本情報を収集します。
 - a) この CICS 領域を CICSplex SM が認識するための名前を決定します。
この名前の長さは最大で 8 文字まで可能です。
 - b) APPLID、つまり CICS 領域の z/OS Communications Server アプリケーション ID を検索します。
 - c) SYSID、つまり CICS 領域の CICS システム ID を検索します。
データ・リポジトリに定義を追加する場合、この情報で十分です。任意の CICS 領域に関して特別な BAS、WLM、RTA、またはモニター要件がある場合、定義を追加する前にこの情報を収集するか、後ほど定義を更新できます。
2. オプション: ご使用の CICS 領域の定義に関する追加情報を収集します。
 - a) システム使用可能性モニター (SAM) を実装している場合、1 次 CMAS の名前および CICS 領域の操作時間を検出します。
1 次 CMAS は、CICS 領域が通常接続する CMAS です。
 - b) CICS コマンド検査をシミュレートするかどうかを決定します。
 - c) CICS リソース検査をシミュレートするかどうかを決定します。
 - d) 他の CICS 領域に対して必要な接続のタイプ、および各接続で許可するセッション数を決定します。
IP 相互接続性 (IPIC) 接続を作成する場合、CICS 領域のポート番号、ホスト名、およびネットワーク ID も決定しなければなりません。
 - e) CMAS と同じ時間帯を使用しない場合には、CICS 領域で使用する時間帯を決定します。

次のタスク

システム定義を作成するために必要な情報を収集し終えたなら、Web ユーザー・インターフェースで CICS Explorer または「**トポロジー管理 (Topology administration)**」ビューを使用して、データ・リポジトリ内に定義を作成できます。

CICS システム・グループ定義の準備

CICSplex SM に対して定義する CICS システム・グループごとに、グループの名前のみを指定する必要があります。CICSplex 内で固有な名前にできます。

CICS システム・グループに CICS システムを追加するには、その前にそのシステム・グループを定義する必要があります。CICS システム・グループを定義するには、「**トポロジー管理**」(ADMTOPOL) ビューを使用します。

CICS システム・グループを定義するというこのプロセスは、おそらく設計ステージの他の部分だけでなく何度も繰り返して行うプロセスです。CICSplex 内の CICS システムはその性質上既にグループ化されていて、そうした各グループを 1 つのエンティティとして操作したいということがあることは明らかです。このような場合、グループ化に関する資料にまず目を通してください。各グループは相互に排他的である必要はありませんし、1 つの CICS システムが任意の数のグループに属することができるので、新しい要件が生じるたびに CICS システム・グループ定義を追加できます。

CICS システム・グループを作成するには、「**システム・グループ定義**」ビュー (CSYSGRP オブジェクト) を使用します。

現行コンテキストとして識別される CICSplex に認識されている CICS システム・グループ、およびそれらに関連付けられている CICS システムに関する情報を、「**システム・グループ定義**」ビュー (CSYSGRP オブジェクト) を使用して表示できます。

CICS システム定義ビュー

Web ユーザー・インターフェースには、CICS システム定義を定義、インストール、および管理するのに役立つ数多くのビューがあります。また CICS Explorer を使用しても、同じタスクを実行できます。

これらのビューについては、[CPSM 管理ビュー](#)に詳しい説明があります。

表 4. CICS システム定義を管理するビュー・セット		
WUI ビュー・セット	オブジェクト名	説明
ISC/MRO 接続定義	CONNDEF	CICS 領域がシステム間連絡 (ISC) または複数領域操作 (MRO) を使用して通信するリモート・システムを識別する接続定義を作成および管理します。
CICS システム定義	CSYSDEF	CICS システムを CICSplex に関連付ける定義を作成および管理します。この定義には、CICS システムが WLM、リアルタイム分析、リソース・モニター、および CICSplex SM のビジネス・アプリケーション・サービス・コンポーネントを使用する方法についての情報も含まれます。
システム・グループ定義	CSYSGRP	CICS 領域の CICS システム・グループへの追加を含め、CICS システム・グループを CICSplex 内に作成して管理します。
IPIC 接続定義	IPCONDEF	IP 相互接続性 (IPIC) 接続を使用して通信する CICS 領域間の接続のアウトバウンド属性について記述する TCP/IP 接続定義を作成および管理します。
時間枠定義	PERIODEF	リソース・モニターおよびリアルタイム分析で使用する時分の特定の範囲を識別する時間枠定義を作成および管理します。
セッション定義	SESSDEF	システム間連絡 (ISC) または複数領域操作 (MRO) を使用して通信する領域間の論理的接続の性質について記述するセッション定義を作成および管理します。
CICS システム・リンク	SYSLINK	CICSplex 内の CICS 領域間のリンクを作成および管理します。
TCP/IP サービス定義	TCPIPS	CICS 領域間の接続のインバウンド属性について記述する TCP/IP 接続定義を作成および管理します。

CICSplex SM には「ランタイム MAS 表示 (Runtime MAS display)」(MAS) ビュー・セットが備えられています。このビュー・セットを使用すると、関連する CICS 領域がアクティブになる際に構成定義を管理できます。

表 5. アクティブな CICS 領域で構成定義を管理するためのビュー・セット		
WUI ビュー・セット	オブジェクト名	説明
ランタイム MAS 表示	MAS	<p>CICSplex が既知の、あるいは、CICSplex の管理に関係する指定の CMAS に接続されている、アクティブな CICS 領域についての情報を管理します。</p> <ul style="list-style-type: none"> • アクティブな CICS 領域についての情報を表示します。 • アクティブな CICS 領域に関する一般情報、WLM、リアルタイム分析、リソース・モニター、および MAS トレース属性を参照または一時的に更新します。 • アクティブな CICS 領域内の MAS エージェント・コードを停止します。

注：定義を更新するために「**CICS システム定義**」ビュー (CSYSDEF オブジェクト) および「**時間枠定義**」ビュー (PERIODEF オブジェクト) を使用すると、現在実行中のシステムとデータ・リポジトリ内の定義

の両方に影響を及ぼします。ただし、「**ランタイム MAS 表示 (Runtime MAS display)**」ビュー (MAS オブジェクト) を使用した更新は、データ・リポジトリには組み込まれません。

次に行うこと

構成とトポロジーが定義され、エンティティー間のリンクが定義された作業 CICSplex SM 環境が整っています。

エンティティーは互いに通信することができ、トランザクションを実行できます。しかし、企業システムの運用を最適化し、CICSplex SM で提供される機能を十分に活用するには、BAS、WLM、RTA、およびモニター要件をセットアップする必要があります。これらのガイドについては、必要に応じて[ビジネス・アプリケーション・サービス \(BAS\) によるリソースの管理](#)、[CICSplex SM によるワークロードの管理](#)、[リアルタイム分析 \(RTA\) によるモニター](#)、または [CICSplex SM モニターを使用する統計の収集](#)をご覧ください。

第 5 章 CICSplex SM によるリソース管理

CICS リソースおよび CICSplex SM リソースはすべて、CICS Explorer、CICS 管理クライアント・インターフェース (CMCI)、WUI ビュー、または CICSplex SM API の各インターフェースのうち、任意のものを使用して管理できます。

リソース定義は以下の方法で管理します。

- CMCI を使用します。この場合、ご使用のアプリケーションからの HTTP 要求を受け入れます。
- CICS Explorer を使用します。この場合、CMCI を利用します。
- CICSplex SM Web ユーザー・インターフェース (WUI) を使用します。
- アプリケーションに直接 CICSplex SM API コマンドを追加します。

リソースの管理およびインストール

CICSplex SM データ・リポジトリに保管されている CICS リソースを管理およびインストールするには、ビジネス・アプリケーション・サービス (BAS) を使用します。

CICS システム定義 (CSD) リポジトリに保管されている CICS リソースを管理およびインストールするには、CSD を使用します。

CSD リソースの CSDGROUP を定義し、BAS リソースの DEFVER を定義する必要があります。

ビジネス・アプリケーション・サービス (BAS) によるリソースの管理

ビジネス・アプリケーション・サービスは CICSplex SM のコンポーネントで、社内のビジネス・アプリケーション用の CICS リソース定義とインストール・プロセスを管理するために使用します。

ビジネス・アプリケーション・サービス (BAS) とは

ビジネス・アプリケーション・サービス (BAS) はオンライン・リソース定義 (RDO) に取って代わり、CICSplex の物理的な場所ではなく、所属するビジネス・アプリケーションに関してその CICS リソースを管理するために使用できます。

社内における有意義なエンティティを表すリソースの集合が、ビジネス・アプリケーションとなり得ます。このアプリケーション内では、特定のタイプのリソースを一緒にグループ化し、そのグループの実行時の特性を定義できます。実行時には、インストールされるリソースの選択対象をフィルター式を使用して絞り込み、オーバーライド式を使用してリソース属性を変更することができます。

BAS 機能

BAS を使用すると、リソースの物理的な場所とは別に、ビジネスにおけるリソースの使用法という観点からリソースの表示と管理を行えます。これは、以下の機能を提供します。

中心的な役割を担うリソース定義

BAS により、CICSplex 全体にわたってリソース定義と関連を実装できます。リソース定義は、CICS システム定義 (CSD) ファイルに保持されます。CICSplex SM データ・リポジトリ (EYUDREP) は、CICSplex 内のすべての CICS システムによってアクセスされる、CICS リソース定義の中央リポジトリとして機能できます。CICSplex SM は、以下のようにして、CICSplex に必要なリソース定義の数を最小化します。

- サポート対象のすべてのプラットフォームにおける CICS リソースの定義方法として、単一システム・イメージを用いた方法が備えられています。
- リソースのローカル・インスタンスとリモート・インスタンスの両方を、単一定義の属性から生成する。
- 複数のバージョンのリソース定義を管理します。例えば、アプリケーションのテスト・フェーズを重ねて最終的に実働に移す際に、アプリケーションの異なるバージョンのリソースを持つことができます。

同一のリソースに同じ名前で新しい定義を作成するたびに、BAS は新しいバージョン番号を割り振ります。リソースを更新した場合には、新しいバージョン番号は作成されないことに注意してください。バージョン番号で、リソース定義の特定のバージョンを指定できます。

バージョン・サポートを使用すると、ビジネス・アプリケーションを開発する際にリソース定義を作成できます。その後、例えば複数のグループ内に単一バージョンのリソースを所有したり、CICSplex 全体で複数バージョンのリソースを所有したりできます。1つの CICS システムで一度にインストールできるリソースのバージョンは 1 つだけであることに注意してください。

詳しくは、[複数バージョンのリソース定義](#)を参照してください。

- 接続とセッション定義の単一の集合から、複数の CICS 通信リンクを生成できます。

BAS リソース・オブジェクトのセットを作成して接続定義を定義し、それらをシステム・リンク (SYSLINK) オブジェクトを使用して多くの CICS 領域で再利用できます。

システム・リンク定義には、接続のタイプ、およびペアの CICS 領域間で接続を作成するのに必要な接続定義を記述します。こうした接続定義を、同じ特性を共用する数多くのシステム・リンクを作成するためのモデルとして使用できます。

SYSLINK オブジェクトを使用すると、以下のように異なるタイプの接続を定義できます。

- MRO 接続または ISC 接続では、CONNECTION リソースと SESSION リソースを記述するための CONNDEF リソース・オブジェクトと SESSDEF リソース・オブジェクトが必要です。
- IPIC 接続では、IPCONN リソースと TCPIP SERVICE リソースを記述するための IPCONDEF リソース・オブジェクトと TCPDEF リソース・オブジェクトが必要です。

こうしたモデル定義を使用する SYSLINK をインストールすると、リソース定義は CICS 領域に自動的にインストールされて、接続が作成されます。

詳しくは、[CICS システム・リンク定義](#)を参照してください。

論理的スコープ

CICS リソースが CICSplex SM に定義されている時、CICSplex 内での物理的な位置ではなく、指定されたビジネス・アプリケーションへの参加という観点から、それらのリソースをモニターおよび制御することができます。論理的に関連したリソースは、ある特定の時点にそれらがどこにあるかに関係なく、1つのセットとして識別および参照することができます。

定義セットを再利用して、システム構成ではなく自分のビジネスのニーズを反映した、任意の数の他のリソースの論理的な関連付けと関連付けることができます。

スコープをアプリケーションに設定した場合、操作またはモニター・ビューにはすべて、選択基準を満たすリソースのみが表示されます。それによって、それらのリソースの管理方法を仔細に制御できるようになります。

分散リソース・インストール

リソースは、CICSplex SM に定義されていても、CICS または CICSplex SM によって、適切なシステムにインストールされる必要があります。BAS を使用して、CICS の初期設定時にリソースを自動的にインストールすることも、領域の実行中に動的にインストールすることもできます。1つのリソースを、ローカル側またはリモート側のいずれか適切な側で、複数の CICS 領域にインストールできます。

サポートされるリソース

ビジネス・アプリケーション・サービスは、以下の CICS リソースをサポートしています。

アプリケーション・リソース

これらは企業でのビジネス・アプリケーションをサポートするリソースです。これらは、アプリケーションが以下を実行するために必要なリソースです。

- CICS BTS プロセス・タイプ
- Db2 の接続およびトランザクション
- 文書テンプレート
- FEPI ノード、プール、プロパティ・セット、およびターゲット
- ファイルおよびファイル・キー・セグメント定義

- IPIC 接続
- LIBRARY リソース
- マップ・セット
- 区分セット
- パイプライン
- Programs (プログラム)
- Sysplex エンキュー・モデル
- TCP/IP サービス
- 一時記憶域モデル
- トランザクション
- 一時データ・キュー
- URI マップ
- Web サービス

領域プロパティ・リソース

以下の領域プロパティ・リソースは、CICS 領域の実行をサポートするグローバル・リソースです。

- ジャーナル
- ジャーナル・モデル
- ローカル共用リソース (LSR) プール
- プロファイル
- トランザクション・クラス
- 端末
- Typeterms

接続性リソース

以下のリソースは、CICS 領域と他のシステムとの間の接続の作成をサポートします。

- 接続
- パートナー
- セッション
- TCP/IP サービス
- IP 相互接続 (IPIC) 接続

BAS の 2 つの形式

BAS は、マイグレーション形式または全機能形式の 2 つの方法で使用できます。状況または必要とする精度に応じて、これらの形式のいずれか一方または両方を使用できます。

- マイグレーション形式では、リソース・グループはリソース記述に関連付けられています。この形式の場合、リソース記述はグループ・リストに似ています。[56 ページの図 16](#) および [64 ページの『マイグレーション形式 BAS の使用』](#)を参照してください。
- 全機能形式では、リソース割り当てを使用するのは、リソース・グループおよびリソース記述の内容の修飾や、CICS システムに対するリソースの割り当ての制御を行うときです。[57 ページの図 17](#) および [64 ページの『全機能形式 BAS の使用』](#)を参照してください。

最も簡単な形式は、リソース記述を使用したマイグレーション形成です。リソース定義とリソース・グループを作成し、それを 1 つ以上のリソース記述に関連付けます。リソース記述は、リソースの論理的スコープを定義します。リソースは特定の CICS システムに割り当てられるので、1 つのシステムに対してはローカルで別のシステムに対してはリモートであるリソースに関しては個別の定義が必要となります。BAS に対するこの方法は、リソース記述がグループ・リストに類似しているという点で RDO を使用する際と似ています。実際には、CICSplex SM データ・リポジトリ上にご使用のバージョンの CSD を作成します。このようにすることは CICSplex SM リソース環境を設定する上で適してはいますが、BAS によって提供される機能すべてを十分に活用することはできません。

場所ではなく、ビジネス・アプリケーションの観点から考えると、リソースを管理するために BAS によって提供される機能を活用するには、リソース割り当てを使用する必要があります。リソース割り当てはグループから特定のタイプのリソースを選択し、それらを適切な CICS システムに割り当てます。リソース割り当ては、リソース記述に関連付けられています。その後、リソース記述はグループ・リストのように機能しなくなりますが、アプリケーションなどのユーザー定義の論理的なリソースの集合となります。

リソース割り当てを使用すると、個々のリソースを管理して、システムごとにその属性を変更したり、特殊な環境に適合させたりできます。以下のことを行うことができます。

- 特定のグループ内の指定のタイプのリソースを制御します。
- リソースをローカルまたはリモートとして識別し、それらを単一のリソース定義を使用して種々の CICS システムに割り当てます。
- フィルター式を指定して、グループから選択したリソースを処理します。
- オーバーライド式を指定して、特定の使用方法をするためにリソース属性を変更します。

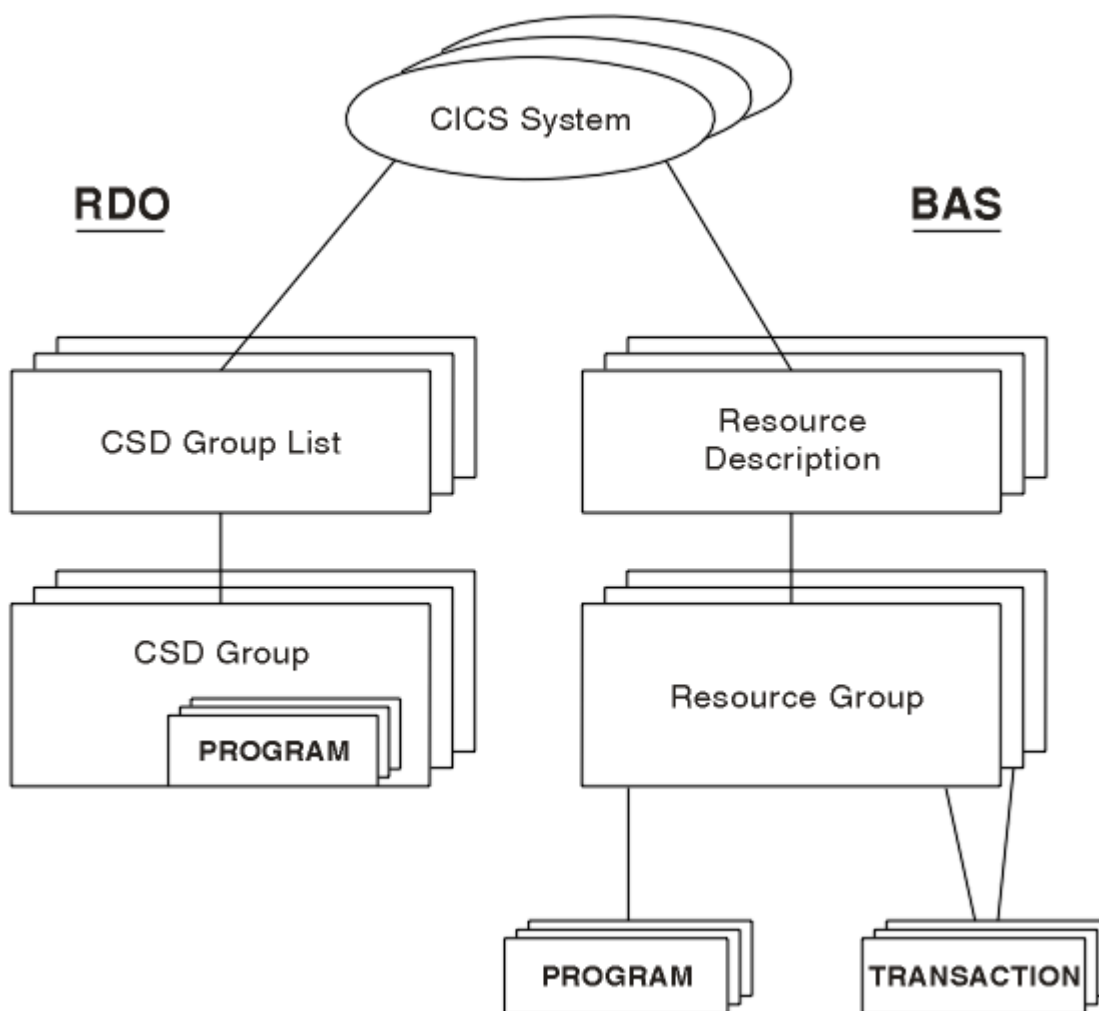


図 16. BAS からのマイグレーション

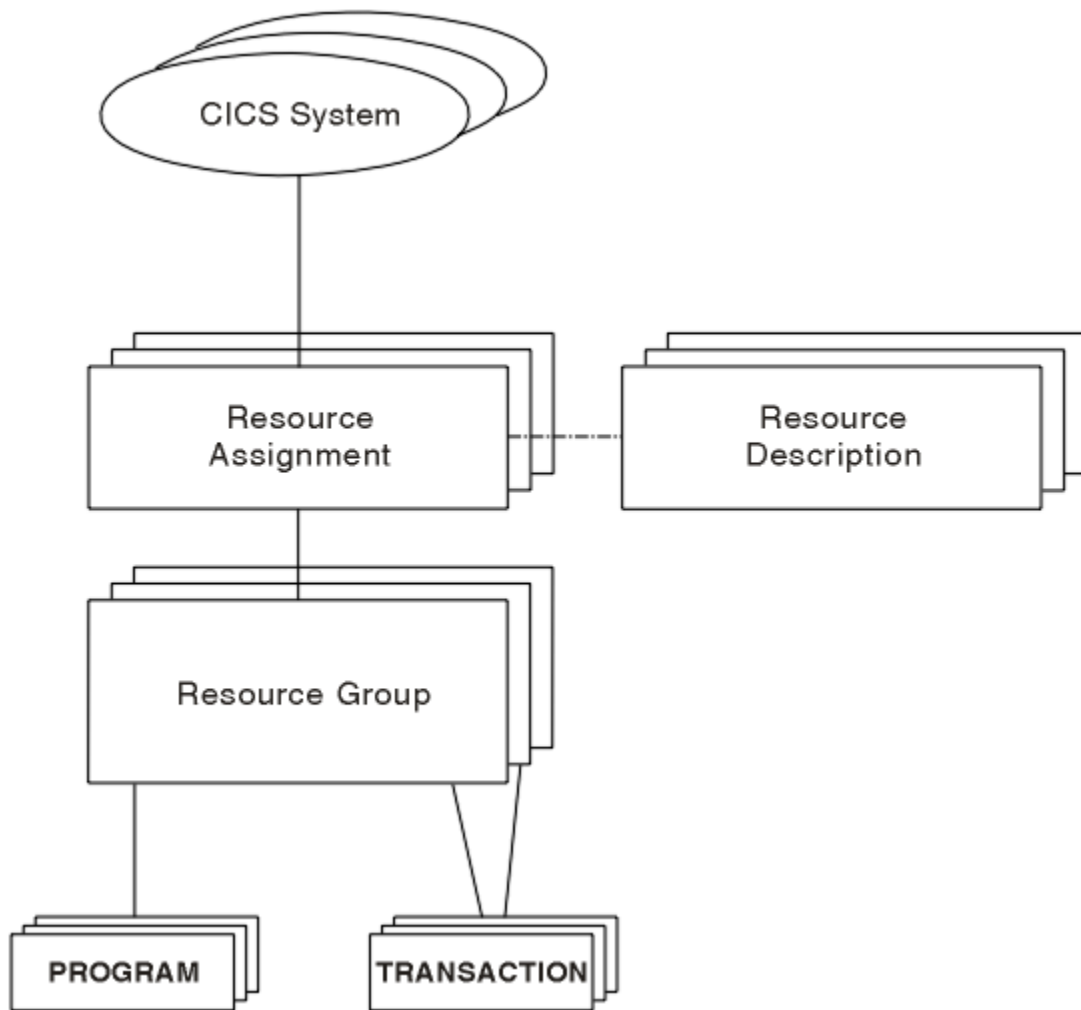


図 17. 全機能 BAS

RDO で BAS を使用する利点

BAS を使用してリソースの管理をすると、RDO を使用する場合に比べていくつかの利点があります。

RDO の制約

RDO を使用するとリソース定義は単一のグループおよび複数のグループに結合され、グループはグループ・リストに従って順番に処理されます。処理中に後で重複した定義が見つかった場合、先に出現した定義がオーバーライドされます。また、RDO で必要になる定義の特徴そのものと、それらの定義が複数の CICS システムで使用可能になる程度には、限界がある場合があります。CICSplex SM 環境では、RDO には次のようないくつかの欠点があります。

- リソースは、指定された CICS システムに対して割り当てられたグループに対して割り当てられます。
- リソース定義は、CSD が共用されているのでないかぎり、CICS システム全体で重複している必要があります。
- 各通信リンクの両端を明示的に定義する必要があります。

BAS の利点

BAS には次の利点があります。

- 使い慣れた、RDO のような定義プロセスで、インターフェース (WUI、バッチ、または API) を選択できる。
- 論理スコープ。これを使用すると、リソースを場所ではなく、ビジネス・アプリケーションの観点で扱えます。

- サポートされるすべての CICS プラットフォームにおいて、CICSplex 内のすべてのリソースの定義リポジトリが共通化される。
- 必要な定義の数の削減。必要に応じて定義を再利用し、個々の属性をオーバーライドできます。
- 一貫性のあるリソース定義と、定義プロセスの多くに対する制御。
- アプリケーションが実行される場所を定義するグループにリソースを追加することによって、リソースを追加領域に追加できる。
- アプリケーションが CICSplex SM に認識されているため、コマンドを、任意の領域グループにではなく、そのアプリケーションと一致するスコープに送信できる。

BAS の管理

BAS 環境は、以下の WUI ビュー・セットおよびリソース管理オブジェクトを使用してセットアップされます。

表 6. WUI ビュー・セットおよびリソース管理オブジェクト		
WUI ビュー・セット	オブジェクト名	説明
リソース割り当て定義	RASGNDEF	リソース割り当ては、選択したリソース定義タイプの特性と、これらのリソースが CICS システムに割り当てられる方法を記述します。割り当てられるリソースはすべて 1 つのタイプ (ファイルなど) でなければならず、1 つのリソース・グループに属する必要があります。リソースは、1 つ以上の CICS システムのローカルおよびリモートの両方として割り当てることができます。リソース割り当てを使用する前に、リソース記述と関連付ける必要があります。「 リソース記述定義 (Resource description definitions) 」(RESDESC) ビューを参照してください。このオブジェクトは、割り当てられるリソースを選択する上で使用される属性、およびリソースが割り当てられるときに変更される属性値を指定するためにも使用されます。
リソース記述内のリソース割り当て	RASINDSC	このビューには、リソース記述、およびそれに関連付けられたリソース割り当てに関する情報が表示されます。この情報の中で、該当するリソース・グループと、CICS システム・グループに割り当てられた CICS システムが、リソースごとに示されます。
リソース割り当てで選択されたリソース	RASPROC	このビューには、指定されたリソース割り当ての処理時に処理されるリソースが表示されます。表示されるリソースは、関連付けられたリソース・グループに含まれるものの中から、提供された選択基準を使用して選択されます。
リソース記述で選択されたリソース (Resource selected by resource description)	RDSCPROC	このビューには、指定されたリソース記述の処理時に選択されるリソースが表示されます。リソースは、リソース記述に直接関連したリソース・グループから (マイグレーション形式 BAS の場合と同様)、およびリソース割り当てから、現在有効な選択基準を使用して選択できます。
リソース記述定義	RESDESC	リソース記述は、リソース・グループのセットを指定します。それによってリソース定義が指定されます。このオブジェクトは、このリソース記述の論理スコープを使用するかどうか、および関連付けられたリソース・グループの CICS システムを指定するために使用します。
リソース・グループ定義	RESGROUP	このオブジェクトは、1 つ以上の関連リソース定義を関連付けるために使用されます。リソース定義は、すべて同じタイプである場合もありますし、タイプが異なる場合もあります。

表 6. WUI ビュー・セットおよびリソース管理オブジェクト (続き)		
WUI ビュー・セット	オブジェクト名	説明
記述内のリソース・グループ (Resource groups in descriptions)	RESINDSC	このビューには、既存のリソース記述、およびそれに関連付けられたリソース・グループに関する情報が表示されます。
リソース・グループ内のリソース定義 (Resource definitions in resource groups)	RESINGRP	このビューには、既存のリソース・グループ、およびそれに関連付けられたリソース定義に関する情報が表示されます。
CICS システム・リンク定義	SYSLINK	このビューには、CICSplex 内の CICS システム間に存在するリンクに関する情報が表示されます。この情報の中で、CICS システムの名前と、リンクを定義するために使用される接続定義とセッション定義の名前が示されます。
CICS システム・リソース (CICS system resources)	SYSRES	このビューには、指定された CICS システムに割り当てられるリソースが表示されます。リソースは、現在 CICS システムに関連しているリソース記述に基づいて選択されます。

表 6. WUI ビュー・セットおよびリソース管理オブジェクト (続き)		
WUI ビュー・セット	オブジェクト名	説明
リソース定義 (Resource definitions)	resDEF ここで <i>res</i> は、定義されているリソースを示します。	<p>リソース定義タイプごとに、その定義の属性を定義するリソース定義オブジェクトがあります。使用可能なリソース・タイプ (括弧内に表示) および CICSplex SM 用にそれらを定義するために使用される WUI ビューは以下のとおりです。</p> <ul style="list-style-type: none"> 「Atomservice 定義 (Atomservice definitions)」ビュー (ATOMDEF オブジェクト) 「バンドル定義 (BUNDLE definitions)」ビュー (BUNDEF オブジェクト) 「CICS-配置 JAR ファイル定義 (CICS-deployed jar file definitions)」ビュー (EJDJDEF オブジェクト) 「DB2® 接続定義 (DB2 connection definitions)」ビュー (DB2CDEF オブジェクト) 「DB2 エントリー定義 (DB2 entry definitions)」ビュー (DB2EDEF オブジェクト) 「DB2 トランザクション定義 (DB2 transaction definitions)」ビュー (DB2TDEF オブジェクト) 「配置済みエンタープライズ Java アーカイブ定義 (Deployed enterprise java archive definitions)」ビュー (EJCODEF オブジェクト) 「文書テンプレート定義 (Document template definitions)」ビュー (DOCDEF オブジェクト) 「FEPI ノード・リスト定義 (FEPI node list definitions)」ビュー (FENODEF オブジェクト) 「FEPI プール定義 (FEPI pool definitions)」ビュー (FEPODEF オブジェクト) 「FEPI プロパティ定義 (FEPI property definitions)」ビュー (FEPRODEF オブジェクト) 「FEPI ターゲット・リスト定義 (FEPI target list definitions)」ビュー (FETRGDEF オブジェクト) 「ファイル定義 (File definitions)」ビュー (FILEDEF オブジェクト) 「ファイル・セグメント (File segment)」ビュー (FSEGDEF オブジェクト) 「グローバル・エンキュー (Global enqueues)」ビュー (ENQMDEF オブジェクト) 「IPIC 接続定義 (IPIC connection definitions)」 (IPCONDEF オブジェクト) 「ISC/MRO 接続定義 (ISC/MRO connection definitions)」ビュー (CONNDEF オブジェクト) 「ジャーナル定義 (Journal definitions)」ビュー (JRNDEF オブジェクト) 「ジャーナル・モデル定義 (Journal model definitions)」ビュー (JRNMDEF オブジェクト) 「LIBRARY 定義 (LIBRARY definitions)」ビュー (LIBDEF オブジェクト) 「LSR プール定義 (LSR pool definitions)」ビュー (LSRDEF オブジェクト) 「マップ・セット定義 (Map set definitions)」ビュー (MAPDEF オブジェクト) 「パートナー定義 (Partner definitions)」ビュー (PARTDEF オブジェクト)

これらのオブジェクト間の関係を示すオブジェクト・モデルは、61 ページの図 18 および 62 ページの図 19 に示されています。ビュー名称の後の括弧内にリソース名が示されています。

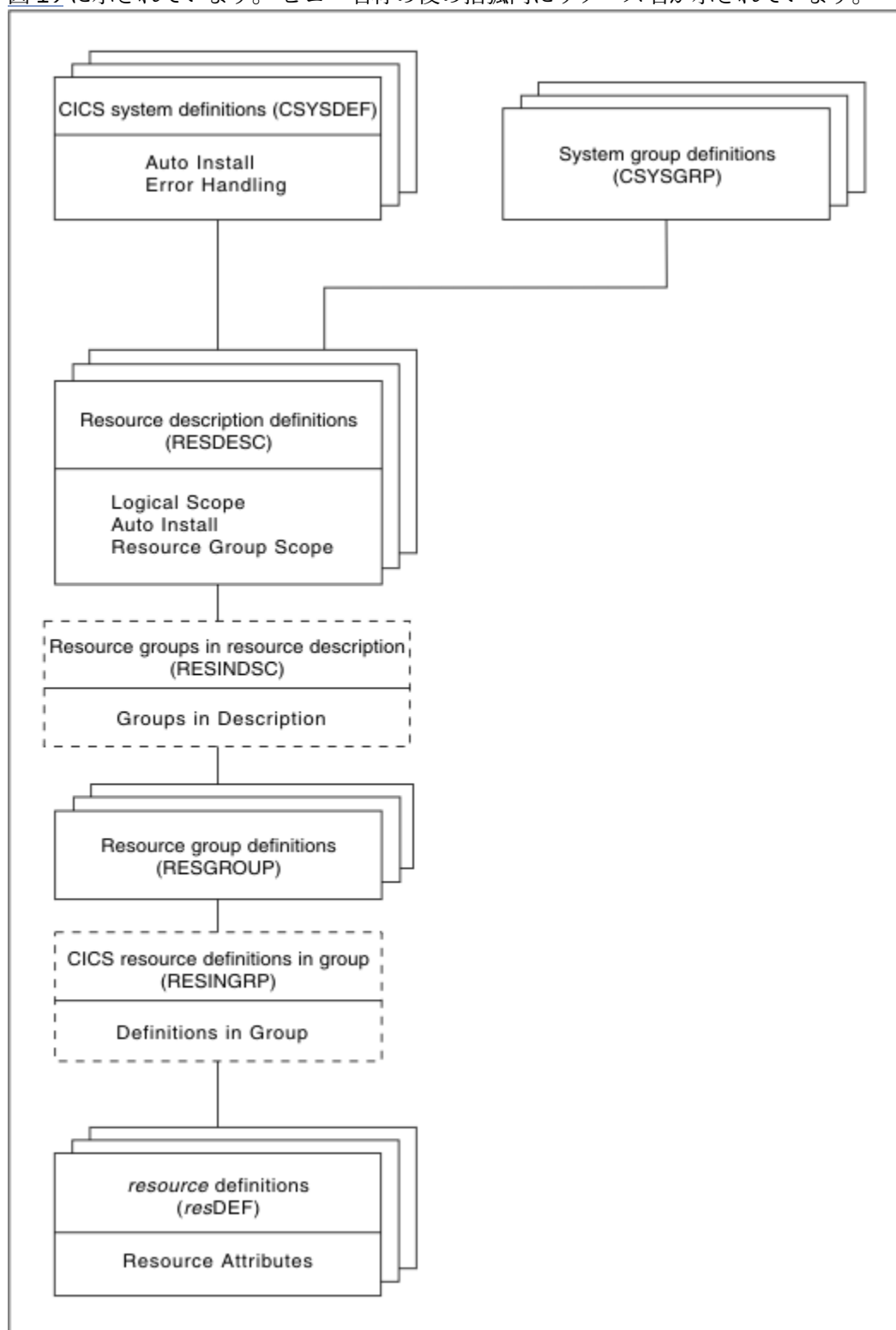


図 18. マイグレーション形式 BAS オブジェクト・モデル

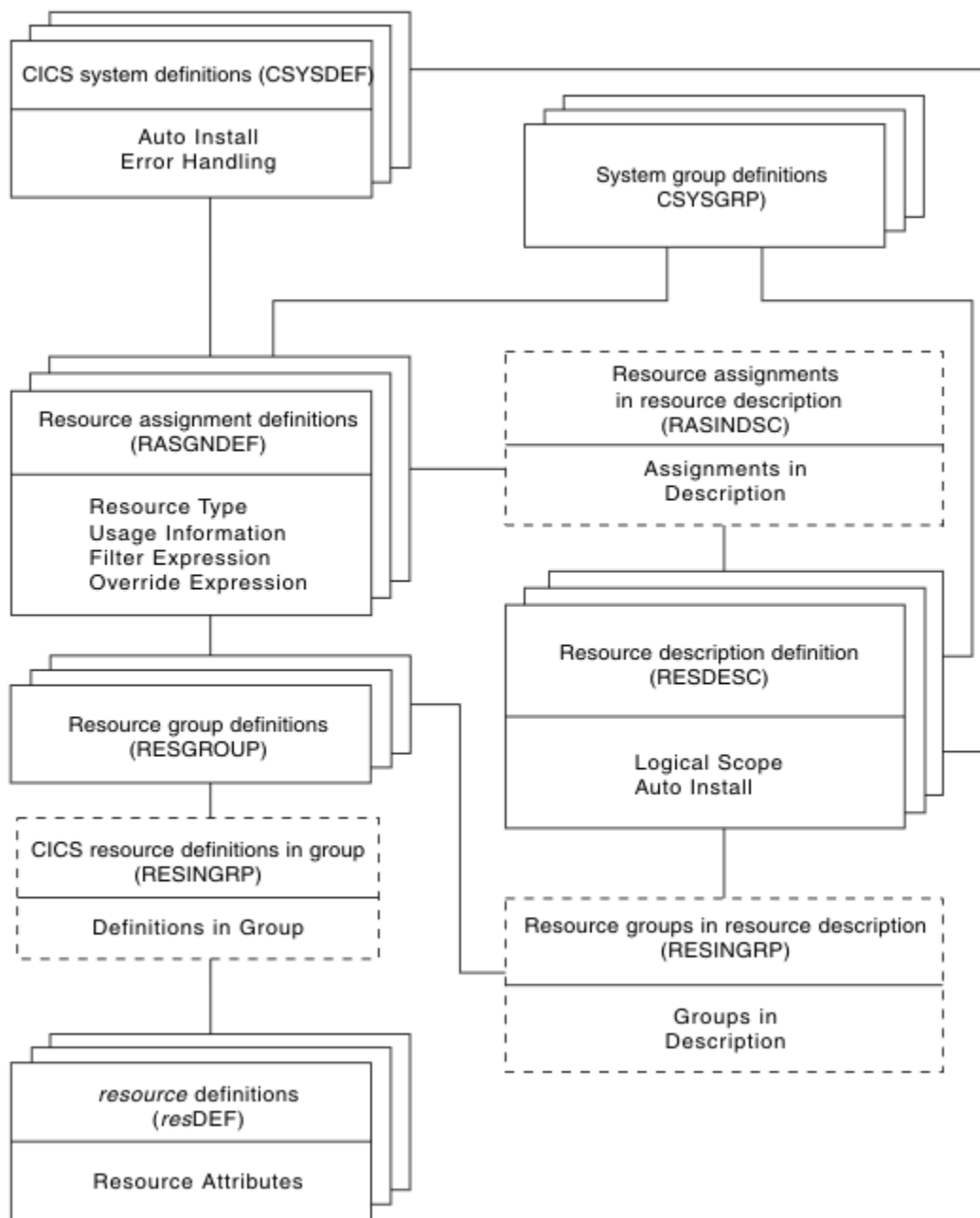


図 19. 全機能 BAS オブジェクト・モデル

リソース定義のマイグレーション

BAS でリソース定義とリソース・グループを再作成する必要はありません。既存の CSD 構造をマイグレーションできます。

このタスクについて

リソース定義とリソース・グループを CSD からデータ・リポジトリに移動するには、以下の手順を実行します。このプロセスでは、リソースとグループ間の関係が維持されます。どの時点においても、CSD のすべてまたは一部をマイグレーションしたり、複数の CSD をマイグレーションしたりすることができます。

手順

1. DFHCSDUP ユーティリティー・ルーチンの EXTRACT コマンドを使用して、CSD レコードを読み取ります。

2. CICSplex SM 抽出ルーチン EYU9BCSD を使用して、バッチ・リポジトリ更新機能コマンドを作成します。
3. これらのコマンドをバッチ処理されるリポジトリ更新機能に入力して、リソース定義とリソース・グループ、およびそれらをリンクするオブジェクトを作成します。

リソースの定義

BAS を使用して CICSplex 全体のリソース定義を定義し、保守することができます。データ・リポジトリ内にリソース定義オブジェクトを作成することにより、大量のリソース定義を作成するためのテンプレートとして、これらのオブジェクトを使用できます。

CICSplex SM のリソースの定義は、CICS RDO に似ています。リソースを定義するには、リソース定義オブジェクトを作成します。定義内でリソースの属性について記述すると、すべてのリソースが出現するごとに定義する必要がなくなります。大量のリソースを作成するためのテンプレートとして、少量のリソース定義を活用できます。リソース定義は、CICSplex のデータ・リポジトリに保管されます。

CICSplex SM リソース定義オブジェクトと CICS リソース定義の相違点は、以下のとおりです。

- 同じリソース定義は、すべてのサポート対象のプラットフォームにおいて CICSplex 内のすべての CICS 領域で使用できます。
- CICSplex SM に対して、各リソースのすべての属性 (ローカル値とリモート値の両方を含む) を定義できます。CICSplex SM は、リソース定義が CICS 領域に割り当てる際に使用する適正な属性のサブセットを判別します。
- ご使用のリソースのインスタンスすべてを初めから定義する必要はありません。テンプレートとなるリソース定義を作成し、それを類似した属性値、あるいはまったく同じ属性値を持つ数多くのリソースに使用できます。CICSplex SM に対してオーバーライドと呼ばれる一時的または永続的な変更を指定し、異なる一群の値を持つリソースを作成できます。
- 同一の名前が付けられたリソース定義の複数のバージョンを作成できます。各バージョンは事実上異なるリソース定義で、異なる CICS システムで使用したり、異なるシステム要件に対して使用したりできます。例えば、システムの開発とテスト用に、種々のリソース要件がある場合があります。[63 ページの『リソースの妥当性検査』](#)を参照してください。

以下の 3 つの方法でリソース定義オブジェクトを作成できます。

- CICS Explorer の管理ビューを使用するか、Web ユーザー・インターフェースの BAS 管理ビューと定義ビューを使用する。
- バッチ・リポジトリ更新機能を使用する ([バッチ処理されるリポジトリ更新機能を参照してください](#))。
- CICSplex SM API を使用する ([Developing CICSplex SM applications](#) を参照してください)。

リソースの妥当性検査

BAS は、RDO と同様に多くの同一リソース定義の検査を行います。

それぞれのリソースが定義およびインストールされるたびに、BAS は以下の検査を行います。

- リソースの個別の属性
- 相互に依存しているリソース属性
- リリース固有のリソース属性

また BAS は、それぞれのリソースのセットの整合性を検査します。リソース・セットに変更を加えるたびに、BAS は追加または更新されているリソースに CICS システムに既にあるリソースと競合していないかどうかを検査します。例えば、同じリソースの異なるバージョンを同じ CICS システムに割り当てようとしたら、ローカルおよびリモートの両方で 1 つのリソースを割り当てようとしたらすると、セット不整合エラーが出ます。

また BAS には MAP 機能が備えられていて、この機能を使用すると、リソース関連が希望通りかどうかを検査できます。この機能を使用して、階層のどの点からでも、リソース関連の構造を表示できます。

リソース・グループの定義

リソース・グループは、1つの単位として管理する関連したリソース定義セットです。リソース・グループは、「リソース・グループ (Resource group)」ビュー (RESGROUP オブジェクト) を使用して定義されます。1つのグループに含まれるリソース定義は、タイプが同じであっても異なっても構いませんが、通常は何らかの共通点があります。それらは、特定のアプリケーションまたは通信ネットワークでの使用法によって論理的に関連している場合もありますし、特定のサイトでの使用法によって地理的に関連している場合もあります。

グループを構成できるリソース定義の数または組み合わせには特に制限はありません。ただし、どのリソースにおいても、一度に1つのリソース・グループに含めることができるのは1つのバージョンのみです。リソース定義の各バージョンは、異なるリソース・グループで保守する必要があります。

マイグレーション形式 BAS の使用

マイグレーション形式 BAS を使用すると、リソース・グループはリソース記述に直接関連付けられます。

リソース記述は、論理スコープを使用しているかどうか、およびそのリソース記述に関連付けるリソース・グループの CICS システムを指定します。61 ページの図 18 を参照してください。

リソースとリソース・グループをマイグレーションまたは定義した場合、アプリケーションを定義するためにリソース記述を作成する必要があります。リソース記述は、RESDESC オブジェクトを使用して定義します。リソース記述は、エンティティとして管理される1つ以上のリソース・グループに直接関連付けられます。マイグレーション形式 BAS モデルでは、この関係は論理スコープを表すと見なすことができますが、すべてのリソースが特定の CICS システムまたは CICS システム・グループと結び付けられるという点において、リソース記述は実際には RDO グループ・リストに類似しています。

全機能形式 BAS の使用

全機能形式 BAS を使用すると、リソース・グループはリソース記述に直接関連付けられることはありません。

追加オブジェクトであるリソース割り当て (RASGNDEF) は、リソース・グループから選択したリソース・タイプの特性と使用法を定義します。リソース割り当てはリソース記述に関連付けられ、リソース記述はビジネス・アプリケーションを表すのに使用できます。62 ページの図 19 を参照してください。

これらのオブジェクトは、ビジネス・アプリケーションに関してリソースを定義するのに使用されます。

リソース割り当ての使用

リソース管理の Power® と柔軟性は、リソース割り当て (RASGNDEF) オブジェクトによって提供されます。

注: マイグレーション形式の BAS を使用している場合、リソース割り当てを使用しないでください。

各リソース割り当ては、1つのリソース・グループ内の1つのリソース・タイプに関連しています。これを論理スコープで使用する場合またはこれが自動的にインストールされる場合には、リソース記述に関連付ける必要があります。各リソース・グループは、複数のリソース割り当てに含めることができます。

フィルター式を使用することによって、リソース割り当てによって指定されたタイプの中でリソースの選択をさらに詳細化できます。さらに、オーバーライド式を使用するとリソース属性を変更できます。属性値は、論理演算子 AND、OR、および NOT を使用して数の制限なく式の中で結合できます。リソース記述がスコープとして指定された場合、CICSplex SM は指定された選択基準を満たすリソースのみを処理します。これにより、リソースの管理の多くを制御できます。

各リソース割り当てがインストールされて論理スコープで使用される場合、それをリソース記述に追加する必要があります。リソース割り当てを作成してそれらをリソース記述に追加すると、リソース・セットを、多数の CICS システムをまたぐことが可能な論理スコープで管理できます。この場合、リソース記述は実質上、リソースのユーザー定義論理セット (アプリケーションなど) になります。特定の CICS システムに複数の異なるリソース記述を関連付け、リソース記述ごとに異なるリソース・セットを示すことができます。

例えば、リソース割り当てを使用すると、以下のことが可能になります。

- リソース・グループから特定のリソースを選択する。

- リソースのローカル・インスタンスおよびリモート・インスタンスを割り当てる必要がある CICS システムを識別する。
- 特定のリソース属性の値を一時的にオーバーライドする。

リソース割り当てによって選択されたリソースは、単独では管理できません。リソースはリソース・グループのメンバーでなければならず、リソース割り当ては少なくとも 1 つのリソース記述に関連付ける必要があります。

リソース記述の使用

全機能 BAS では、リソース記述は、アプリケーション、つまり複数の CICS システムをまたぐことができる論理的に関連付けられたリソース・セットを表します。そのためリソース記述は、アプリケーションの論理スコープを指定します。

アプリケーションの一部として使用される各リソース・グループをリソース記述に追加する必要があります。リソース・グループ全体をリソース記述に関連付けると、より大きなリソース・セット (CSD グループ・リストのようなもの) を作成できます。このようにすると、リソースをさらに効率的に管理できます。リソース記述で指定されるリソース・セットには、以下の特徴があります。

- 以降の CICSplex SM 要求で使用する論理スコープ (アプリケーションなど) として指定できます。
- CICS システムが自身を CICSplex SM に MAS として指定するとき自動的にインストールすることもでき、CICS システムの稼働中に動的にインストールすることもできます。

これらのリソース関連の作成および管理で使用する BAS オブジェクトは、[61 ページの図 18](#) に示されています。これらの機能については [58 ページの表 6](#) で要約されています。

アプリケーションとリソースのインストール

CSD からリソースをインストールするための代替りのオプションとして、BAS を使用して、データ・リポジトリからリソースをインストールできます。

BAS を使用すると、リソース記述または 1 つのリソース・グループをインストールすることにより、アプリケーションに関連付けられたすべてのリソースをインストールできます。BAS は、アプリケーション・リソース記述またはリソース・グループに関連付けられたリソース割り当てを適用し、それらの基準に基づいてインストールする実際のリソースを選択します。別の方法としては、個別のリソースを 1 つ以上の CICS システムにローカルまたはリモートのいずれか適切な方法でインストールできます。

BAS はリソースをシステム初期設定時に自動的にインストールすることもでき、アクティブな CICS システムに動的にインストールすることもできます。リソースをインストールする前に、CICSplex SM はそのリソースが CICS システムに既に存在するかどうかを判別する検査を実行します。通常の状態では、リソースが存在すると、新しいリソースは重複であると見なされ、インストールされません。ただし、リソースを動的にインストールする場合、BAS では重複リソースの検査を迂回して、リソースの無条件インストールを施行するというオプションが提供されます。

自動インストール

CICS システムが初期化され、CMAS に認識されると、BAS はそのシステムに関連付けられているリソースを検討し、インストールするリソースの集合を決定します。

動的インストール

アクティブな CICS システムにリソースを動的にインストールできます。そのためには、該当するリソース定義ビュー、または以下のいずれかのビューから「**インストール (install)**」ボタンをクリックします。

- 「リソース・グループ定義」ビュー。
- 「リソース記述定義」ビュー。
- 「システム・リンク定義」ビュー。

セキュリティに関する考慮事項

CICSplex SM 環境に対するリソース定義の重要性を考えると、特定のタイプのリソースまたは特定の機能に対するユーザー・アクセスの許可を実装する際には十分に注意を払う必要があります。

ユーザーにすべてのリソース定義ビューとその関連する管理ビューへのアクセスを許可することもできますし、特定のリソース・タイプの定義ビューにしかアクセスできないように制限することもできます。ま

たとえば、あるユーザーには「**ISC/MRO 接続定義**」ビュー (CONNDEF オブジェクト) と「**セッション定義**」ビュー (SESSDEF オブジェクト) への読み取り専用アクセスと、その他すべてのリソース・タイプには更新アクセス (リソースの作成と管理が許可される) を与えるなど、ユーザーが持つアクセス権のタイプを定義できます。BAS ビューが適切に保護されていて、許可されていないユーザーがリソースの作成と管理を行えないようにする必要があります。

また CICS TS を実行していて、EXEC CICS CREATE コマンドを使用して新しいリソースを作成する場合にも注意が必要です。コンテキストとして CICSplex で作成されるすべての定義は、CICSplex 内のすべての CMAS に自動的に配布されます。そのため、BAS オブジェクトを作成する権限をユーザーに与えるということは、CICSplex 内の任意の CICS システムにリソースをインストールする権限を付与することと同じ意味を持つことになりかねません。CICS システムが開始するとき、だれがシステムにリソースをインストールしたかについての検査はありません。

企業での CICSplex SM のセキュリティのセットアップについて詳しくは、[BAS のセキュリティの考慮事項](#)を参照してください。

BAS の計画

BAS を使用すると、リソースを定義する CICSplex SM オブジェクトを作成し、アプリケーションの実行を厳密に制御するような仕方でそれらのオブジェクトをグループ化することができます。

すべての CSD リソースを CICSplex SM に同時に転送したり、最初に使用方法を決定したりする必要はありません。1 つ以上の CSD からリソース定義レコードを抽出したり、1 つの CSD からレコードのサブセットのみを抽出したりできます。全機能 BAS を使用するよう直接移行することもできますし、BAS からマイグレーションを使用して、徐々に全機能 BAS に移行することもできます。同じ CICSplex で、RDO を使用することもできますし、BAS からマイグレーションすることも可能で、さらには全機能 BAS を使用することもできます。

このセクションでは、社内で BAS の実装計画を開始する際に考慮するであろう以下のような幾つかの質問を取り上げます。

- 抽出機能を使用する予定の場合:
 - CSD の抽出順序は?
 - 対象とするのは CSD の全レコードそれともサブセットですか?
 - 同時に複数の CSD を抽出しますか?
- マイグレーション形式の方法を使用しますか?
- 全機能 BAS を実装しますか? その場合、ビジネス・アプリケーションにおけるリソースの使用法と作成する必要がある割り当てについて考慮する必要があります。
- 自動的にインストールするリソースと、動的にインストールするリソースはどれですか?
- PLT 処理の前に必要な定義はどれですか? PLT フェーズ 2 処理の前に必要な定義が CSD になければなりません。
- リソースのインストール場所はどこですか?
- CSD で定義されていないリソースを手動で定義する必要がありますか?
- 実装する必要があるセキュリティ手段は何ですか? (65 ページの『[セキュリティに関する考慮事項](#)』を参照してください)。

CICS Interdependency Analyzer は、アプリケーション・リソース・フローについて計画し理解するのに役立ちます。以下について表示されます。

- トランザクション・リソースの依存関係; つまり、正常に実行されるために、個別の CICS トランザクションで使用され、トランザクションが依存しているリソースの集合です。
- トランザクションの親和性; つまり、相互に親和性を持っているトランザクションのグループ。この場合、グループ全体が同じ領域にインストールされるか、特定の領域内になければなりません。

CICS Interdependency Analyzer について詳しくは、[CICS Interdependency Analyzer for z/OS の概要](#)を参照してください。

BAS の実装

このセクションは、CICSplex に BAS を実装するために行える経路指定について取り上げています。

BAS のマイグレーションの作成

BAS を使用できるように移行するためのプロセスの最初のステージは、CICS から CICSplex SM にリソースをマイグレーションすることです。

- リソース定義、リソース・グループ、およびそれらの間の関係を、CSD から CICSplex SM データ・リポジトリに移動します。[62 ページの『リソース定義のマイグレーション』](#)で取り上げられているこのプロセスにより、中間のリソース階層が提供されます。これは、各リソース定義がリソース・グループ内にあるという点において CEDA に非常によく似ています。
- CSD 上に保留されていないリソースの定義とグループを作成します。
- 各リソース・グループをリソース記述に関連付けて、アプリケーションを作成します。

これで、CICSplex SM システムを完全に申し分なく実行できるようになりましたが、リソース割り当てがないので、BAS によって提供される特別な機能を活用することはできません。

CEDA 定義を抽出すると、[67 ページの『リソース割り当てを使用するまでのマイグレーション』](#)で説明されているように、リソース定義を完全な BAS 機能にマイグレーションすることができます。

リソース割り当てを使用するまでのマイグレーション

同時にすべてのリソースをリソース割り当てにマイグレーションする必要はありません。

「リソース割り当て定義」ビュー (RASGNDEF オブジェクト) を使用して、対象となる特定のリソースを識別し、それらからリソース定義との関連付けを削除した後に、リソース割り当てを作成します。リソース割り当てがうまく行えたなら、次に別の一群のリソース定義を定義できます。

リソース定義のプロセスに移行し十分に機能するようになってきたなら、リソース・グループをリソース記述との直接的な関連付けから削除し、代わりに 1 つ以上のリソース割り当てに指定します。

注: 同じリソース記述を使用して、リソース・グループ全体と、リソース割り当て内で識別される選択済みリソースの両方を管理できます。リソース割り当て機能を活用し始めるには、基礎となるリソース定義のいくつかを更新する必要がある場合があります。例えば、以前に CICS システムにデフォルトのまま関連付けられていたリソース定義の場合、別の CICS システムにローカルまたはリモートのリソースとして割り当てするにはその前に属性を追加しなければならないこともあります。

推奨されている方法

社内で BAS 機能を実装することにする場合、まず該当するオブジェクトを定義する必要があります。

このセクションでは、これまでのセクションを要約しています。以下を実行する必要があります。

- [62 ページの『リソース定義のマイグレーション』](#)で説明されているように、CSD からリソース定義とリソース・グループを抽出します。以下の方法を使用することもできます。
 - 適切なリソース定義ビューを使用して、リソース定義を作成します。
 - 「リソース・グループ定義」ビューを使用して、リソース・グループ (RESGROUP オブジェクト) を作成します。
- 「リソース割り当て定義」ビューを使用して、リソース割り当て (RASGNDEF オブジェクト) を作成します。
- 「リソース記述定義」ビューを使用して、リソース記述 (RESDESC オブジェクト) を作成し、先ほど作成した定義と割り当てに関連付けます。

その後、これらのオブジェクト間の関連を作成するには以下のようにします。

- 適切なリソース定義ビューを使用して、リソース定義をリソース・グループ (RESINGRP オブジェクト) に追加します。
- 「リソース・グループ定義」(RESGROUP) ビューを使用して、リソース・グループをリソース記述 (RESINDESC オブジェクト) に追加します。
- 「リソース割り当て定義」ビュー (RASGNDEF オブジェクト) を使用して、リソース割り当てをリソース記述 (RASINDSC オブジェクト) に関連付けます。

- 「**リソース割り当て定義**」ビュー (RASGNDEF オブジェクト) を使用して、リソース割り当てを CICS システムまたは CICS システム・グループに関連付けます。

「**マップ (Map)**」アクション・ボタンを使用すると、データ・リポジトリ内の BAS 定義のビジュアル・マップを表示できます。定義済みのオブジェクトと関連のリストを表示するには、「**リソース・グループ内のリソース定義**」ビュー (RESINGRP オブジェクト) を使用します。

次に行うこと

BAS 要件を識別したなら、リソースとその関連をセットアップする必要があります。

この説明は、[Creating resources with BAS](#) にあります。

第 6 章 リアルタイム分析 (RTA) によるモニター

リアルタイム分析は CICSplex SM のコンポーネントで、社内での例外条件の通知の定義および管理に使用します。

RTA 要件を確認したら、[リアルタイム分析管理ビューの操作](#)の説明のように CICSplex SM に対してそれらを定義してください。

CICSplex SM リアルタイム分析とは

CICSplex SM リアルタイム分析 (RTA) 機能により、対象とした条件に関する外部通知が自動的に提供されます。

リアルタイム分析は、認識される一般的なエラー状態のみを対象としているものではありません。リソースの状況に関する任意の特徴について通知されるようにすることができます。

RTA 機能は以下のとおりです。

- 70 ページの『システム使用可能性モニター (SAM)』
- RTA リソース・モニター。これには、以下の機能が含まれています。
 - 71 ページの『MAS リソース・モニター (MRM)』
 - 72 ページの『分析点モニター (APM)』

例えば、CICSplex SM に対して CICS システム AORPAY1 が使用するすべてのファイルが ENABLED 状況になければならないと指示すると、こうしたいずれかのファイルの状況が ENABLED 以外になると CICSplex SM は警告メッセージを送出します。ただし、CICSplex SM は問題が生じた後にのみ通知するように制限されるわけではありません。CICSplex SM の強力な機能により、CICS リソースの潜在的な問題が検出されます。実際、CICSplex SM は CICS リソースの状況のわずかな変更であっても検出して警告する機能において類を見ません。例えば、CICSplex SM はリソース競合の度合いが強くなる場合、動的ストレージ域 (DSA) の空きスペースが少なくなってきた場合、特定のトランザクションのユーザー数が多い場合に通知できます。これらは、将来の問題の指標とすべてなり得るもので、予防的なアクションを実行することによって回避できます。

通知は、コンソール・メッセージの形式と NetView への総称アラートの形式の一方または両方で行われます。CICSplex SM アプリケーション・プログラミング・インターフェース (API) を使用すると、CICSplex SM が MVS メッセージおよび SNA 総称アラートを送出するのを待機するのではなく、自動化製品が CICSplex SM からの状況情報を積極的に収集することが可能になります。

自動化する理由

CICSplex SM オペレーターは、RTA メッセージをモニターして、適切なアクションを実行できます。

例えば、CICSplex SM オペレーターは、CICSplex SM ローカル・ファイル・ビュー・セット (LOCFILE オブジェクト) を使用してファイルの状況を DISABLED から ENABLED に変更できます。ただし、この方法は CICS 可用性とパフォーマンスの要件が増大するにつれ、十分に対応できなくなります。多くの企業では、クローン作成するなどして多くの CICS システムを使用し、増大するワークロードを処理しています。この要因 1 つだけを取り上げても、CICSplex SM オペレーターのタスクは困難なものとなり、エラーが生じやすくなります。しかし、同時にサービス・レベルの合意によりパフォーマンスや可用性の目標が設定され、以前より多くが要求されるようになります。従って、CICSplex SM オペレーターには最も例外的な事情の問題だけを表示するのが理想的な環境と言えます。CICSplex SM によって検出される実際の問題および潜在的問題のほとんどは、以下に示す自動化に対応した製品を賢明に使用することによって修正したり回避したりすることができるので、前述のような環境が実現可能になりました。

- NetView
- 自動オペレーション管理プログラム/MVS (AOC/MVS)
- 自動オペレーション管理プログラム/MVS CICS Automation (AOC/MVS CICS Automation)

- CICSplex SM 自体。(CICSplex SM は CICS リソース状況に関する詳細な情報を他の自動化製品に適時提供するだけでなく、検出した多くの問題を修正することもできます。)

RTA の利点

- アクションを CICS 内で実行できるので、外部の自動化製品を使用する必要がなくなります。
- 潜在的な問題を検出し、重大なエラーとなる前に修正できます。
- 応答時間に影響を及ぼしかねないボトルネックを早めに検出し、エンド・ユーザーのパフォーマンスを向上させます。
- 不要なメッセージをフィルター処理して、オペレーターに表示する情報量を減らせます。

RTA 外部通知

RTA 外部通知は、CICS リソースの状況が想定外の場合に生成されます。この通知は、外部メッセージ または SNA 総称アラート のいずれか、またはその両方の形式で行うことができます。

外部メッセージ

外部メッセージは、デフォルトで MVS コンソールに書き込まれるオペレーター・メッセージです。また外部メッセージは、条件がアクティブな間は CICSplex SM 「**RTA 未解決のイベント**」ビュー (EVENT オブジェクト) で表示できます。

条件が解決されると、「**RTA 未解決のイベント**」ビュー (EVENT オブジェクト) には表示されなくなります。(ただし、「**RTA 未解決のイベント**」ビュー (EVENT オブジェクト) に書き込まれた外部メッセージは CICSplex SM EYULOG にも書き込まれ、「**RTA 未解決のイベント**」ビュー (EVENT オブジェクト) の内容のレコードが提供されます。)**「RTA 未解決のイベント」**ビュー (EVENT オブジェクト) では、外部メッセージは緊急度に応じてソートされ、最も緊急度の高い条件に関するメッセージがリストの先頭に表示されます。条件の緊急度はユーザーが定義します。

外部メッセージはペアになって発行されます。1つのメッセージは条件の開始時に発行され、もう1つのメッセージは条件が終了すると発行されます。すべてのメッセージに接頭部「EYUPN」が付きます。外部メッセージは、NetView、AOC/MVS、および AOC/MVS CICS Automation などの自動化製品によって代行受信される可能性があります。

SNA 総称アラート

CICSplex SM は SNA 総称アラートを構成して、それら进行处理するために NetView プログラム間インターフェース (PPI) を介して NetView に送信できます。

NetView は自動化テーブルで SNA 総称アラートを捕捉できます。自動化テーブルでは、問題を解決するためのアクションを実行したり、コマンド・リスト (CLIST) またはコマンド・プロセッサから一部の自動化コードを開始したりすることができます。

システム使用可能性モニター (SAM)

CICSplex SM は、CICS システムの可用性を予定の使用可能な時間中モニターするシステム使用可能性モニター (SAM) 機能を備えています。

CICS システムがモニターされている間に、事前定義の一連の条件のいずれかが起きた場合、その条件の開始時点とその解決時点に外部通知が CICSplex SM から送信されます。条件は以下のとおりです。

SAMMAX

CICS の最大タスク数に達しました。

SAMOPS

CICS システムが使用不可です。

SAMNRM

CMAS が、その管理対象であるアクティブな MAS との接続を失いました。

CMAS からの照会に対して MAS が 60 秒間応答しなかったため、SAMNRM イベントが出されました。他の SAM 条件とは異なり、このイベントは CMAS が MAS と通信しているときのみ検出可能です。この条件は、CMAS が MAS と通信していない場合は検出できません。

SAMSDM

CICS システム・ダンプが進行中です。

SAMSOS

CICS 動的ストレージ域 (DSA) がストレージ不足です。

SAMSTL

CICS システム内の 1 つ以上のタスクが停止しました。DBCTL、DL/I、一時データ、ファイル、ジャーナル、および一時記憶域などのリソースの競合が原因です。

SAMTDM

CICS トランザクション・ダンプが進行中です。

例えば、CICS システム AORPAY2 が時刻 0900 から時刻 1200 までアクティブでなければならず、CICSplex SM が時刻 0930 に非アクティブであることを検出すると、CICSplex SM はデフォルトで外部メッセージを発行します。CICSplex SM オペレーターから、または自動化製品からの標準的な応答では、CICS システムを再開しようとするか、別のシステムを開始しようとしています。

SAM 外部通知のカスタマイズ

いずれかの SAM 条件が発生すると、CICSplex SM はデフォルトで外部メッセージを発行します。

各 SAM 条件にはデフォルトの重大度と優先順位の値があり、それにより、EVENT ビュー内の他の外部メッセージに対してその外部メッセージの相対的な位置が決定されます。例えば、SAMTDM 条件の重大度は HW (高位の警告) で優先順位が 128、SAMSOS 条件の重大度は HS (高位の重大度) で優先順位は 255 (最高の優先順位) なので、SAMSOS は「**RTA 未解決のイベント**」ビュー (EVENT オブジェクト) では SAMTDM より先に位置します。

CICSplex SM アクション定義を作成すると、任意の CICS システムのデフォルト SAM 通知をカスタマイズできます。アクション定義を使用すると、以下を変更できます。

- 特定の SAM 条件に対して発行される通知のタイプ。例えば、特定の CICS システムで SAMSOS 条件が生じると SNA 総称アラートが NetView に送信され (そのため NetView では修正アクションを実行できるようになります)、外部メッセージは「**RTA 未解決のイベント**」ビュー (EVENT オブジェクト) には表示されないように指定できます。
- SAM 条件の重大度値と優先順位の値。例えば、特定の CICS システムに対して、SAMTDM 条件の重大度が LW (低位の警告)、SAMSOS 条件の優先順位が 128 でなければならぬと指定できます。

またアクション定義を使用して、ある条件によって影響を受ける CICS TS システムを収集し、MVS 自動リスタート・マネージャー (ARM) を使用して再開するように要求できます。これは、一例として SAMOPS 条件などでは適した自動化アクションとなり得ます。

自動化と SAM

AOC/MVS または NetView などの自動化製品は SAM 通知を代行受信し、CICS システムの開始またはシャットダウン、トランザクションの使用不可化、またはダンプの要求など修正アクションを実行します。

例えば、CICS が使用不可になる場合、いずれかの自動化製品がシステムの再始動を試行したり、別のシステムを開始したりすることができます。

ただし、ほとんどの SAM 通知はユーザーに既に影響を及ぼしている実際の問題と関連しているので、予防のためにとるアクションの有効範囲は小さくなります。従って、通常ほとんどの SAM 条件に関する最善の自動化オプションは、ダンプを要求することです。これにより、問題とその原因に関する詳細情報が提供可能です。

MAS リソース・モニター (MRM)

CICSplex SM MAS リソース・モニター (MRM) 機能を使用すると、特定または一般的な CICS リソースの状況をモニターし、その状態が指定規範から逸脱した場合には通知を受け取るようにすることができます。

(対照的に SAM は、CICS システム自体の『デフォルト』リソースのモニターと評価です。SAM から受信する通知のタイプを調整することは可能ですが、通知を起動する条件は CICSplex SM によって定義されます。) MRM を使用すれば、関心の対象となるリソース状況と、それによって生成される外部通知のタイプを選択することができます。以下は、MRM を使用できる方法例の一部です。

- MRM を使用して、リソースの絶対状況の変更が通知されるようにできます。例えば、LU6.2 接続の状況が ACQUIRED でなければならぬと指定されていたもの RELEASED である場合、外部通知を発行し、

CICSplex SM オペレーターまたは自動化製品が接続の再獲得を試行できるようにします。MRM は、多くのリソースの絶対状況の変更を警告できます。例えば、ジャーナルが CLOSED の場合、一時データ・キューが DISABLED の場合、FEPI ノードまたは端末が OUTSERVICE の場合、トランザクションが DISABLED の場合などに通知されるようになります。

- CICSplex SM は、CICS リソースの絶対状況値のモニターに加え、CICS システムにおけるトランザクションの応答時間の低下、プログラムのユーザー数の増加、または Db2 スレッド数の変更など、リソース状況のわずかな変更に関する情報も提供できます。これらすべては、問題の発端の指標となり得るリソース動作における傾向の一例です。
- MRM を使用すると、複合条件を指定できます。例えば、特定のトランザクションのユーザー数が指定のレベルに達し、なおかつ 動的ストレージ域 (DSA) の空きサイズが低下した場合に、外部通知が発行されるようになります。両方の条件が真の場合にのみ、通知が発行されます。条件はどれほど複雑でも構いません。
- MRM は Db2 などの CICS 以外のリソースや、CICS/400 などの CICS ファミリーの他のメンバーであっても、状況プローブと呼ばれるユーザー作成プログラムを呼び出すことによりモニターをサポートします。

その名の通り MAS リソース・モニターは、CICS システム・レベルで動作します。つまり、リソースの状況が 2 つの領域で変更されると、外部通知がそれぞれ 1 つの領域に対して 1 つずつ、合計 2 セットが発行されます。

自動化と MRM

MRM は、実際の問題と潜在的な問題の両方についてのアラートを出せるので、広範囲に渡る CICS リソース管理を自動化できます。

実際の問題についてアラートが出る場合、MRM は改善 アクションを実行する機会を提供し、潜在的な問題について警告が出る場合、MRM は予防 アクションを実行する機会を差し伸べます。また、すべての MRM 検出問題については自動応答が可能です。

リソースの絶対状況が変更されると (例えば、ENABLED でなければならないファイルが DISABLED になったり、INSERVICE でなければならない接続が OUTSERVICE になったりする場合)、改善アクションが通常必要になります。CICSplex SM 自体もそのようなリソースの状況を自動的に修正できます。NetView または AOC/MVS CICS Automation などの自動化製品と同様です。

リソースの状況に関してごくわずかな変更が生じる場合、NetView などの自動化製品は問題の重大度に応じて応答を調整できます。例えば、CICSplex SM は CICS システム内の DSA の空きサイズを 5 秒ごとにモニターできます。CICSplex SM が DSA の空きサイズに関して、以下のようにモニターするとします。

- 連続して 4 回とも 86KB から 100KB の範囲であると、NetView に対して重大度 LW (低位の警告) で SNA 総称アラートを発行します。
- 連続して 3 回、71KB から 85KB の範囲であると、NetView に対して重大度 LS (低位の重大度) で SNA 総称アラートを発行します。
- 連続して 2 回 70KB 以下であれば、重大度 VLS (非常に低位の重大度) で SNA 総称アラートを発行します。

こうしたアラートの応答として、NetView 自動化テーブルは MVS 変更コマンドを関係する領域に発行し、DSA サイズを動的に増やすことによって、ストレージ不足条件 (つまり SAMSOS 条件) が生じるのを防ぎます。それぞれの場合の DSA の増加量は、CICSplex SM が SNA 総称アラートに付加する重大度によってまったく異なります。

分析点モニター (APM)

CICSplex SM の分析点モニター (APM) 機能は MRM 機能と同じです。ただし、複数の CICS システム (CICS システム・グループまたは CICSplex 全体) でのリソースのモニター時には、同じ問題が起きた CICS システムがいくつあっても、複数ではなく単数の外部通知が行われる点を除きます。

APM が特に便利なのは、複製された AOR を使用する環境の場合です。そこでは、各領域は同じであるため、一般問題について警告するための通知は 1 つで足りるからです。APM では、CICS 以外のリソースのモニターでの状況プローブの使用はサポートされません。ただし、それ以外のどの面でも、潜在的な問題の早期検出や、問題の発生時の自動応答といった点で APM は MRM と同じ働きをします。

RTA の管理

RTA は、WUI ビュー・セットとリソース・オブジェクトを使用してセットアップおよび管理します。

この説明は、[73 ページの表 7](#)にあります。

表 7. リアルタイム分析オブジェクト		
WUI ビュー・セット	オブジェクト名	説明
分析点仕様に関連付けられたインストール済みの分析定義	APACTV	このオブジェクトには、RTA 点仕様に関連付けられた RTA 定義についての情報が表示されます。
アクション定義	ACTION	アクション定義は、RTA 定義中で識別される 1 つ以上の条件が真の場合に生じる外部通知のタイプを指定します。
RTA 分析点仕様	APSPEC	このオブジェクトには、RTA 点仕様についての情報が表示されます。RTA 分析点仕様は、コンテキストとして識別される CICSplex 内の CICS システムの分析を担う 1 つ以上の CMAS を識別します。
1 次 CMAS 内の RTA 仕様	CMDMPAPS	このオブジェクトには、RTA 仕様と 1 次 CMAS との間の関連が表示されます。
2 次 CMAS 内の RTA 仕様	CMDMSAPS	このオブジェクトには、RTA 仕様と 2 次 CMAS との間の関連が表示されます。
評価定義	EVALDEF	このオブジェクトには、評価定義に関する情報が表示されます。評価定義は、サンプリングされて評価される、1 つ以上の CICS システム内にあるリソースを識別します。評価の結果が真の場合、関連する RTA 定義を使用して通知すべき条件が発生したかどうかを判別します。
リアルタイム分析 (RTA) 未解決のイベント	EVENT	このオブジェクトには、CICSplex またはその CICS システムの状況に関する未解決の変更についての情報が表示されます。
仕様とシステム・グループとの間のリンク	LNKSRSCG	このオブジェクトには、分析仕様に関連付けられる CICS システム・グループが表示されます。
RTA 仕様と CICS システムとの間のリンク	LNKSRSCS	このオブジェクトには、分析仕様に関連付けられる CICS システムが表示されます。
リアルタイム分析 (RTA) インストール済みの分析と状況定義	RTAACTV	このオブジェクトには、現行コンテキストとして識別される CICSplex に認識される CICS システムにインストールされた RTA と状況定義に関する情報が表示されます。
RTA 定義	RTADEF	RTA 定義は、定期的に行われる評価、通知する必要がある条件が発生した場合にとるアクションを識別します。
RTA グループ	RTAGROUP	RTA グループは、1 つ以上の関連した RTA 定義、状況定義、あるいはその両方を関連付けるために使用します。
分析点仕様内の RTA グループ	RTAINAPS	このオブジェクトには、RTA 点仕様に関連付けられる RTA グループの名前が表示されます。
RTA グループ内の RTA 定義	RTAINGRP	このオブジェクトには、RTA グループの名前、およびそれに関連付けられる RTA 状況定義が表示されます。

表 7. リアルタイム分析オブジェクト (続き)		
WUI ビュー・セット	オブジェクト名	説明
RTA 仕様内の RTA グループ	RTAINSPC	このオブジェクトには、RTA 仕様の名前、およびそれに関連付けられる RTA グループが表示されます。
RTA 仕様	RTASPEC	RTA 仕様は、システム使用可能性モニターで使用されるデフォルト制御属性を指定し、CICS システムに関連付けられたすべての RTA 定義および状況定義のアンカーとなります。
RTA グループ内の状況定義	STAINGRP	このオブジェクトには、RTA グループ内の状況定義が表示されます。
状況プローブ定義	STATDEF	状況プローブ定義は、特定の間隔で CICSplex SM により呼び出されるユーザー・プログラムを指定します。

RTA オブジェクト・モデルは、[75 ページの図 20](#) に示されています。この図には、WUI ビューの名称と共にリソースの名前が括弧内に示されています。

RTA の計画

RTA の実装を計画する際、以下の点について考慮できます。

- SAM は簡単に実装できるだけでなく (76 ページの『システム使用可能性モニター (SAM) の実装』を参照してください)、CICS システムをアクティブにすべきときにアクティブになっていないことを通知される必要があるため、SAM を使用したいと思われるに違いありません。そのためには以下を行う必要があります。
 - いずれかのデフォルト条件をカスタマイズするかどうかを決定します。
 - 各 CICS システムの 1 次 CMAS を識別します。
- MRM および APM を使用するには、以下を識別する必要があります。
 - 関心のあるリソース
 - どの時点の状況が関心の対象となるか
 - 関心のある状況について通知する方法
 - リソース評価を必要とする間隔
 - データ分析を必要とする間隔
 - 1 次 CMAS

注：APM の場合のみ、1 つ以上の 2 次 CMAS も指定できます。

RTA の実装

リアルタイム分析は、結果として生成される情報に対して徐々に実装する必要がありますし、既に生じている必要に応じて実装する必要があります。特に、例外条件に対する応答を自動化することを予定していない場合にはそう言えます。

まず SAM を実装し、問題ログなどの分析対象レコードが存在するようになってから MRM を実装することが推奨されています。そうしたレコードによって、MRM が有用になる箇所について示唆が得られます。

システム使用可能性モニター (SAM) の実装

SAM を以下のように実装することが推奨されています。

1. 「時間枠定義」ビューを使用して、CICSplex SM によって管理される CICS システムごとに予想される可用性時間についての時間枠定義 (PERIODEF オブジェクト) を作成します。例えば、7 つの CICS システムを時刻 0800 から時刻 1900 までアクティブにする必要がある場合、1 日のその部分に関する時間枠定義を 1 つ作成します。
2. それぞれの CICS システムを CICSplex SM に定義する際、システムが使用可能な時刻を定義した時間枠定義の名前を指定します。

SAM をアクティブにして、通知する必要がある条件が生じた場合にデフォルトの外部通知が生成されるようにするにはこの 2 つのアクションで十分です。ご存じのとおり、CICSplex SM を社内でインストールするとすぐに SAM を実装でき、その際に余分な努力はほとんど必要ありません。すべての SAM 条件のデフォルト通知を少なくとも一度は参照するまでは、SAM をデフォルトの通知のまま使用することをお勧めします。いずれかの通知をカスタマイズする場合、1 つの SAM 条件のアクション定義をまず作成し、それを単一の CICS システムに適用して、影響をテストしてください。変更内容が希望どおりであれば、それを他の SAM 条件と他の CICS システムに必要なに応じて適用します。

SAM に関連するオーバーヘッドは大きくないので、CICSplex SM またはそれにより管理されている CICS システムの一般的なパフォーマンスに影響を与えることなく、社内の CICSplex 全体で SAM をアクティブにできます。

MAS リソース・モニター (MRM) の実装

システムごとに MRM を実装することから開始してください。1 つの CICS システムで接続またはファイルなどの特定のリソース・タイプを 1 つ選択してから、以下の CICSplex SM オブジェクトを作成します。

- 評価定義。関心のあるリソースのインスタンス、CICSplex SM がリソース状況を評価する頻度、検出対象を CICSplex SM が解釈する方法、および条件が真の場合に割り当てる重大度を指定します。
- 分析定義。CICSplex SM が結果を分析する頻度、外部通知を発行する必要があるポイントを指定します。
- アクション定義。発行される外部通知のタイプを指定します。

リソース評価の間隔 (評価定義で指定されている)、およびデータが分析される間隔 (分析定義で指定されている) はほぼ一致している (理想的には同一である) 必要があります。特に、データの分析頻度を評価の頻度より多くしてはなりません。また、リソース評価に関しては依然として有用なデータが提供されている間は可能な限り実行頻度を少なくしてください。頻繁にデータを収集すると、収集コストもそれだけ大きくなります。

初めに、アクティブな CICS システムに分析定義を手動でインストールします。MAS リソース・モニターが予想通りに十分に作動している場合には、分析定義を自動インストールすると役立つかどうかを考慮してください。そうであれば、分析定義を分析グループに追加し、分析グループを分析仕様に関連付けます。この時点で、同じ CICS システム内の別のリソースをモニターするかどうかを決めます。MAS リソース・モニターを他のリソースにも拡張するには、追加の評価定義が必要となり、別の分析定義も必要とされる場合がありますが、元のアクション定義をおそらく再利用できます。

分析点モニター (APM) の実装

MAS リソース・モニターの結果に満足できる場合、次に CICS システムの論理的グループ化に適用される分析点モニター (APM) に進みます。分析点モニター要件によって影響を受ける CICSplex を考察し、それらの CICSplex の管理に係る CMAS を識別します。

MAS リソース・モニターで必要とされるほとんどの CICSplex SM 定義は分析点モニターに再利用できるので、この方法で行う実装ステージのコストは高くありません。例えば、1 つのアクション定義を、複数の分析定義で使用することもできます。ただし、分析点モニターの場合、少なくとも 1 つの分析点仕様を作成しなければなりません。MAS リソース・モニターの場合と同様、簡単な定義の分析点モニターから開始して、徐々により複雑な定義を使用するように移行してください。

推奨されている方法

社内で RTA 機能を実装することにする場合、まず該当するオブジェクトを定義する必要があります。

以下を実行する必要があります。

- 「**RTA 定義**」ビューを使用して、分析定義 (RTADEF オブジェクト) を作成します。
- 「**評価定義**」ビューと「**アクション定義**」ビューを使用して分析定義に必要なサブコンポーネント (EVALDEF オブジェクトと ACTION オブジェクト) を作成します。
- 「**RTA グループ**」ビューを使用して分析グループ (RTAGROUP オブジェクト) を作成します。
- 「**RTA 仕様**」ビューを使用して分析仕様 (RTASPEC オブジェクト) を作成します。

その後、これらのオブジェクト間の関連を作成するには以下のようにします。

- 「**RTA 定義**」ビューを使用して、分析定義を分析グループ (RTAINGRP オブジェクト) に追加します。
- 「**RTA グループ**」ビューを使用して、分析グループを分析仕様 (RTAINAPS オブジェクト) に追加します。
- 「**RTA 仕様と CICS システムとの間のリンク**」ビュー (LNKSRSCS オブジェクト) または「**RTA 仕様と CICS システム・グループとの間のリンク**」ビュー (LNKSRSCG オブジェクト) を使用して、CICS システムまたは CICS システム・グループに分析仕様を関連付けます。LNKSxSCG パラメーターについて詳しくは、[属性およびリソース・テーブルに関する特別な考慮事項: LNKSxSCG レコード \(LNKSMSCG、LNKSRSCG、LNKSWSCG\)](#) を参照してください。

「**マップ (Map)**」アクション・ボタンを使用すると、データ・リポジトリ内のリアルタイム分析定義のビジュアル・マップを表示できます。定義済みのオブジェクトと関連のリストを表示するには、「**RTA グループ**

「**RTA 定義**」ビュー (RTAINGRP オブジェクト)、「**RTA 仕様内の RTA グループ**」ビュー (RTAINSPC オブジェクト)、「**RTA 仕様と CICS システムとの間のリンク**」ビュー (LNKSRSCS オブジェクト)、および「**RTA 仕様と CICS システム・グループとの間のリンク**」ビュー (LNKSRSCG オブジェクト) を使用します。

第 7 章 CICSplex SM モニターによる統計の収集

モニターは CICSplex SM のコンポーネントで、社内でのパフォーマンス関連データの収集を定義および管理するために使用します。

CICSplex SM モニターとは

CICSplex SM モニター は、一群の CICS システム内にある指定のリソース・インスタンスのパフォーマンス関連データを、ユーザー定義の間隔で収集できます。

CICSplex SM モニターには、以下の利点があります。

- リソースに関する要約と詳細情報をすぐに利用できることで、ヘルプ・デスク・サービスを改善できます。
- オペレーターが効率的に問題を調査してボトルネックを検出できるようになるので、システム可用性とパフォーマンスが向上します。

ユーザー定義の間隔で、モニターは選択したリソースが CICSplex 内に存在し、制御可能な状態になるといつでもその状況情報と統計を収集します。CICSplex SM は、標準の CICS インターフェース (**EXEC CICS INQUIRE**、**EXEC CICS COLLECT STATISTICS**) を使用して情報を収集するので、CICS モニター機能 (CMF) の代わりに使用できます。CMF が実行されていると、それにより提供される情報を CICSplex SM は活用します。リアルタイム分析 (76 ページの『RTA の実装』を参照してください) の監視対象となっているリソースは、既に別個にモニターされているのでモニターを指定する必要はありません。

CICSplex SM モニター関数

CICSplex SM モニターは、以下の機能を提供します。

リソースのモニター

CICSplex SM は、CICS リソースのパフォーマンスを定期的にモニターできる機能を持っています。

このモニター活動によって収集されるデータは、CICS コマンド **EXEC CICS COLLECT STATISTICS** によって収集されるデータの種類の、またはそのようなデータから導き出されたデータのいずれかになります。後者については、CICSplex SM によって未加工の統計データから速度、合計、平均、および割合が計算されます。ローカル MAS の場合に限り、CICSplex SM は CICS モニター機能 (CMF) を介して詳細なモニター・データも取得できます。

モニター間隔

特定のリソースのアクティビティをモニターするとき、通常は、特定の期間におけるパフォーマンスに関心があります。そのため、CICSplex を CICSplex SM に定義するとき、CICSplex のモニター間隔を指定します。

「**CICSplex の定義 (CICSplex definitions)**」ビュー (CPLEXDEF オブジェクト) を使用して、CICSplex を定義します。モニター間隔は時間の長さで、その終了時に導き出されたモニター・データを保持するカウンターはゼロにリセットされます。例えば、モニター間隔を 60 分に設定した場合、カウンターは 60 分間は増え続け、その後ゼロにリセットされてモニター・データの累算が再度開始します。この手順は、モニターがアクティブである間は繰り返されます。間隔は分単位 (15 から 1440) で指定され、デフォルトは 480 分 (8 時間) です。モニター間隔中に CICSplex SM によって導き出された値には、関連したビューで「MI」の接頭文字が付きます。

サンプリング間隔

モニターするリソースのタイプ (ファイル、接続など) ごとに、サンプリング間隔を指定します。サンプリング間隔は、そのリソース・タイプの選択インスタンスに関するパフォーマンス関連データを CICSplex SM が収集する頻度を決定します。

例えば、CICS システムの特定のグループで使用されるファイルに、300 秒のサンプリング間隔を指定できます。最後のサンプリング間隔の際に CICSplex SM によって導き出された値には、関連したビューで「CS」の接頭文字が付きます。したがって、MI 値は長時間のトレンドを識別する上で役立つ一方、CS 値は変化の速度が速いものを識別できます。

CICSplex SM モニター機能の管理

WUI ビュー・セットとリソース・オブジェクトを使用して、モニター機能をセットアップして管理します。

これについては、[80 ページの表 8](#) を参照してください。

表 8. モニター WUI ビュー・セットとリソース・オブジェクト		
WUI ビュー・セット	オブジェクト名	説明
モニター定義	MONDEF	モニター定義では、モニターの対象または非対象とするリソースの出現を示すことにより、モニター仕様で指定されるリソース・タイプを限定します。モニター定義は、アクティブになった際にどのリソースがリソース状況ファシリティーに報告されるかも指定します。
モニター・グループ	MONGROUP	モニター・グループは、1 つ以上の関連したモニター定義を関連付けるために使用します。
グループ内のモニター定義	MONINGRP	このオブジェクトには、モニター・グループの名前、それに関連付けられるモニター定義、およびモニター定義がアクティブになっている期間が表示されます。
モニター仕様内のモニター・グループ	MONINSPC	このオブジェクトには、モニター仕様の名前、およびそれに関連付けられるモニター・グループが表示されます。
モニター仕様とシステム・グループとの間のリンク	LNKSMSCG	このオブジェクトには、モニター仕様に関連付けられる CICS システム・グループが表示されます。
モニター仕様と CICS システムとの間のリンク	LNKSMSCS	このオブジェクトには、モニター仕様に関連付けられる CICS システムが表示されます。
アクティブ・モニターの仕様	POLMON	このオブジェクトには、現行コンテキストとして識別される CICSplex に認識される CICS システムにインストールされたモニター定義に関する情報が表示されます。

モニター・オブジェクト・モデルは [81 ページの図 21](#) に示されています。WUI ビューの名称の後に、リソース名が括弧内に記されています。

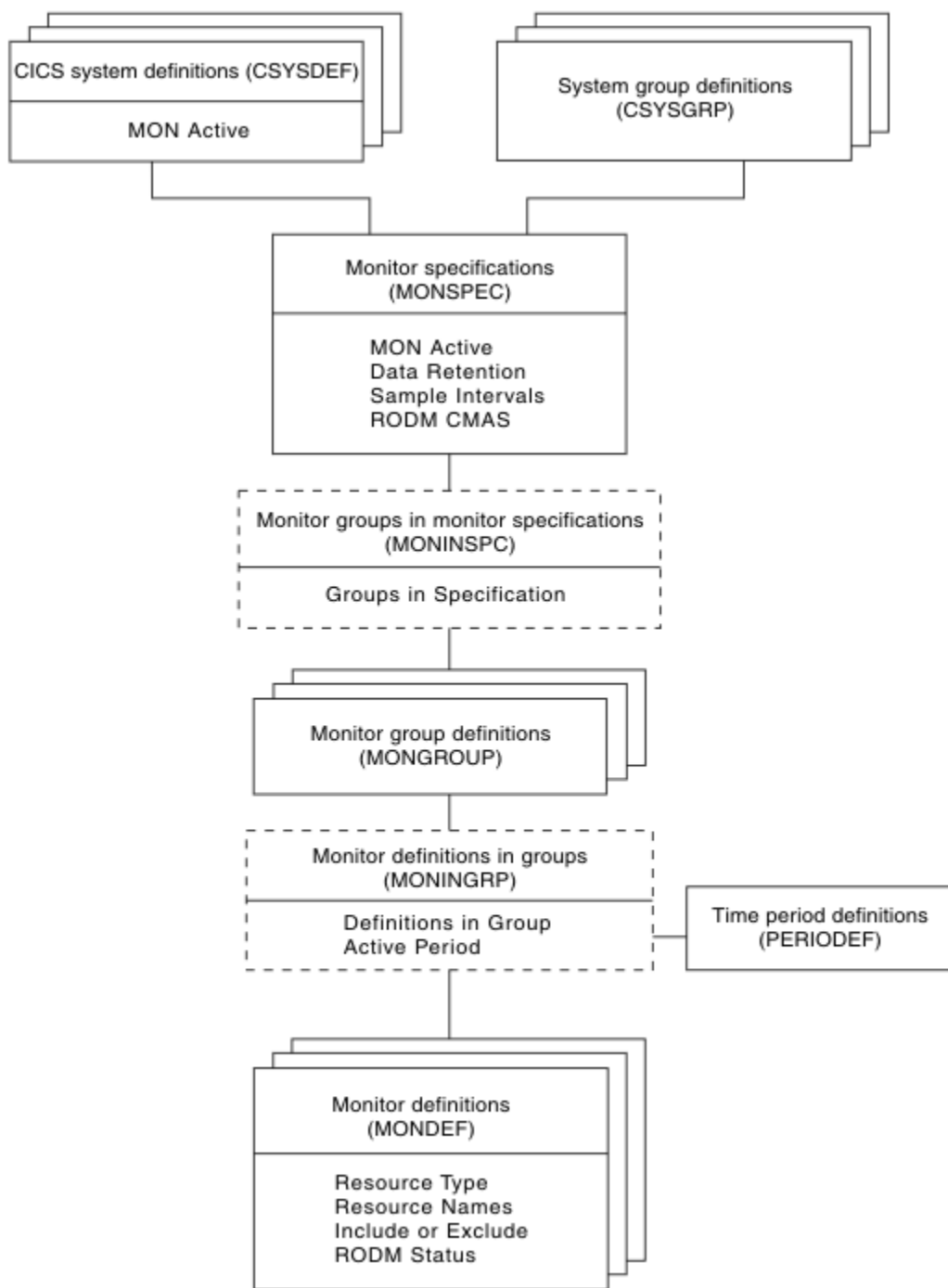


図 21. リソース・モニター・オブジェクト・モデル

CICSplex SM モニターの計画

CICSplex SM を使用して社内でのモニター対象について計画を立てる場合、使用可能なモニター機能を最適な方法で使用するのに以下の推奨事項が役立ちます。

要求するモニター・データの使用目的を必ず理解しておく

サンプリング間隔がゼロより大きい各リソース・クラスの MAS (つまり、管理対象の各 CICS システム) には、モニターが必要な CICS タスクがあります。そのため、例えば CICSplex 内のすべての CICS システム内のトランザクションのモニターをアクティブにすると、それぞれの MAS では指定の間隔でトランザクションをモニターする CICS タスクをサポートします。また、モニター・データのキャッシュを維持するためには CMAS 処理も必要です。一般的で制限のないモニターを行うと、容認できないほどの

オーバーヘッドがすぐに生じるようになります。ですから、特定のデータを収集する理由を把握することなく、CICS システムまたは CICS リソース・クラスに対するモニターをアクティブにはしないでください。

可能な限り具体的にリソースを指定する

CICSplex SM モニターは、詳細な使用法とパフォーマンス・データが提供される特殊機能です。通常これは、今後問題となり得る、または実際に問題となっている短期間における傾向を分析するのに最も役立ちます。例:

- 限られた数のリソースの使用レベルが高い場合に、モニター対象の候補となります。モニター・データによって、リソースの数を増やすかどうかを決定するのに役立つからです。
- 競合のあるリソースも十分に候補となります。そうしたデータは、スループットにおける競合の影響を評価するのに役立ちます。

しかし、最も関心を払うべきリソースはどれかを理解しておく必要があります。例えば、CICS 提供リソースには本当に関心がありますか? リソース名としてアスタリスク (*) を使って指定したくなるのがよくありますが、これは実際には、指定のリソースのすべてのインスタンスを意味します。すべての出現に対して関心を持っているわけではない場合、汎用の名前パターンを使用します。例えば、"PAY" で始まるすべてのトランザクションにモニターを指示するには、リソース名として "PAY*" と指定することができます。同様に、リソース状況機能に報告する状況のリソースについて、可能な限り十分に識別する必要があります。

できるだけ長いサンプリング間隔を指定する

リソース・クラスに指定するサンプリング間隔が短ければ短いほど、モニター・データを収集するコストはそれだけ大きくなります。例えば、サンプリング間隔を各リソースにつき 1 秒に設定すると、他の CICSplex SM 処理はまったく行われなくなってしまう。このため、有用なデータを取得できる範囲で最長のサンプリング間隔を必ず指定してください。この間隔をどれほどにすべきかを正確に決定するには、特定のリソース・インスタンスに関して既に把握している事柄を活用すべきです。通常、サンプリング間隔はリソースの使用に関して実際に起こりうるレベルを反映していなければなりません。リソースのサンプリングを継続的に行っても重要な変更を示していないことが分かるなら、サンプリング間隔を長くする必要があります。またサンプリング間隔は、CICSplex SM または NetView オペレーターがモニター・データを調べる頻度とも何らかの関係があるはずです。

できるだけ長いモニター間隔を指定する

モニター間隔は、累積されたモニター・データを保持するカウンターがゼロにリセットされる頻度を決定します。この間隔を社内で意味のある値に設定することにより、モニターから得られる情報も意味のあるものとなるようにしてください。例えば、この間隔を勤務時間にしたり、特定のワークロードが処理中のその他の期間にしたりできます。

この間隔を短くし過ぎないように推奨されています。モニター間隔が短くなると CICSplex SM オーバーヘッドが大きくなるだけでなく、間隔を短くし過ぎると有効な統計を得られる可能性が低くなってしまふからです。

実際に必要とする以上のモニター仕様を定義しない

モニター仕様が多くなり過ぎないようにしてください。それらの仕様の作成と CICS システムへの関連付けに処理が余分に必要となるだけでなく、1 つの CICS システムが同時に属することのできるモニター仕様は 1 つだけだからです。より多くのモニター仕様を定義すればするほど、汎用のモニター・ポリシーを設立できる可能性が低くなるからです。

適切な保存期間を指定する

保存期間を指定する必要はありません。ただし、保存期間をデフォルト値にゼロ分にする、と、任意の CICS システムでモニターが停止するとすぐにそのシステムのモニター・データが失われます。何らかの予期しないイベントが生じる場合、このデータはとても貴重になり得るので、保存期間をゼロより大きく設定するのが当を得ています。ただし、CICS システムが停止すると、指定した保存期間に関わらず、システムの再始動時にモニター・データは失われることに注意してください。この場合、CICS システムを再始動する前にモニター・データを調べることが大切になります。

保存期間は、1 分から 1440 分 (24 時間) の間の値に設定できます。モニター仕様で保存期間を指定できます。この場合、この期間はそのモニター仕様に関連付けられているすべての CICS システムに適用されます。あるいは、CICSplex SM CICS システム定義で保存期間を指定することもできます。

モニター定義をアクティブにする時期を決定する

モニター定義をモニター・グループに追加する際、一日のうちいつアクティブにするかを指定できます。十分に考慮したうえでこの機能を使用するようにしてください。例えば、夜間はモニターをアクティブにする必要はおそらくないかもしれませんが、全体的に使用量が少なくデータを表示するユーザーがだれもない期間にはモニター定義が効力を持たないようにしてください。また、競合があるリソースをモニターしている場合、その競合がいつ生じるかを考慮してください。絶えず発生しますか？それとも時刻 0900 から 1100 までだけ生じますか？後者の場合には、モニターをその期間にだけ限定してください。

CICSplex SM モニターの実装

CICSplex SM モニターを実装するおそらく最適な方法は当初まったくモニターを行わず、特定の CICS システムと CICS リソース・インスタンスに対して徐々に実装していくという方法です。

具体的には、以下のとおりです。

1. モニター仕様を作成し、それを CICSplex 内のすべての CICS システムと関連付けます。ただし、サンプリング間隔すべてをデフォルト値のゼロのままにし、モニター状況を OFF に設定したままにしてください。これは、何らかのモニター・データを収集する前にすべき事柄の 1 つにモニター仕様があるからです。こうした説明を実行するとモニターは使用可能になりますが、まだアクティブにはなっていません。
2. 特定のリソース・インスタンスをモニターすることにした場合、CICSplex 内の CICS システムで使用する場合と同様に、以下を行う必要があります。
 - a. CICS システム定義を更新してモニターをオンに切り替え、リソースのサンプリング間隔を指定します。トポロジー「ランタイム MAS 表示」ビュー (MAS オブジェクト) を使用して、この変更をアクティブな CICS システムに加えると、現在実行されている CICS システムにのみ変更内容が適用されます。
 - b. モニター定義を作成してリソース・インスタンスに名前を付けてから、CICS システムにそれを手動でインストールします。
3. 通常の要件がはっきりとするまで、モニターに関するこの漸進的な方法を繰り返します。この時点で、モニター定義の自動的なインストール (および CICS システムすべての実行) を考慮します。自動インストールする場合には、モニター定義をモニター・グループに追加し、そのモニター・グループをモニター仕様に関連付けます。サンプリング間隔とモニター状況を恒久的なものにできます。そのための 1 つの方法は、モニター仕様でそれらを設定する方法で、この場合、このモニター仕様に関連付けられているすべての CICS システム、および該当するモニター定義が提供されている CICS システムに適用されます。もう 1 つの方法は、CICS システム定義で設定する方法で、この場合、他の CICS システムにはまったく影響を及ぼしません。

モニターを徐々に構築するための方法を洗練できます。そのためには、CICSplex 用の追加のモニター仕様を作成します。例えば、指定の期間に自動的にインストールされる時限式モニター定義を使用できます。以下を覚えておいてください。

- モニター仕様は使用中に変更することはできますが、こうした変更内容は CICS システムを次に開始するまでは有効になりません。ただし、MAS ビューを使用して、アクティブな CICS システムのモニター値を変更できます。
- CICS システムを関連付けることができるのは、一度に 1 つのモニター仕様だけです。
- モニター仕様を、既に実行している CICS システムに関連付けることができますが、CICS システムを次に開始するまではこの新しい関連の影響は有効になりません。

指定するモニター仕様とモニター定義は通常の状況に対応し、CICS システムのオーバーライドは例外に使用されることを目的としています。

リソース状況機能の活動化

リソース状況機能を活動化するには、以下を行う必要があります。

- リソース状況機能にリソースを報告することになっている CICSplex の CICSplex SM 定義を更新します。
- モニター定義を更新または作成して、リソース状況機能の「状況 (Status)」フラグを「YES」に設定し、リソース・インスタンスを識別します。

- ・モニター仕様を更新または作成して、リソース・タイプごとのサンプリング間隔を設定します。

これらすべてのタスクに関する説明は、[CICSplex SM の管理およびリソース・モニター](#)に記載されています。

リソース状況機能について詳しくは、[CICSplex SM システム・パラメーター](#) の **RESSTATUS** パラメーターに関する説明を参照してください。

CICSplex SM での CICS モニター機能 (CMF) の使用

CICSplex SM のモニター・データの一部のクラス、特に一部のトランザクション・モニター・ビューに表示される詳細なタスク関連データに関しては、CICS パフォーマンス・クラス・データ・モニターがモニター対象の CICS システムで活動化されていない場合は収集できません。

CICSplex SM 操作機能、CICS モニター機能トランザクション CEMN、または **SET MONITOR** コマンドを使用して、アクティブな CICS 領域でのパフォーマンス・クラス・データ・モニターを活動化できます。または、CICS システム初期設定パラメーターを使用してパフォーマンス・クラス・データ・モニターを恒久的に活動化することもできます。

CICS モニター・データは、SMF データ・セットに通常書き込まれます。ただし、CICSplex SM が SMF データ・セットに書き込まないで、使用するモニター・データを収集することも望むのであれば可能です。そのためには、CICSplex SM システム・パラメーター **SUPPRESSCMF** を指定します。このパラメーターについて詳しくは、[CICSplex SM システム・パラメーター](#) を参照してください。

モニター・データの様々なクラスについて、および CICS モニター機能の制御方法については、[CICS モニタリングの概要](#)をご覧ください。

推奨されている方法

社内にモニター機能を実装することにした場合は、まず該当するオブジェクトを定義します。

1. 以下のオブジェクトを作成します。

- ・「**モニター定義**」ビューを使用して、モニター定義 (MONDEF オブジェクト) を作成します。
- ・「**モニター・グループ**」ビューを使用して、モニター・グループ (MONGROUP オブジェクト) を作成します。
- ・「**モニター仕様**」ビューを使用して、モニター仕様 (MONSPEC オブジェクト) を作成します。

2. これらのオブジェクト間の関連を作成します。

- ・「**モニター定義**」ビューを使用して、モニター定義を分析グループ (MONINGRP オブジェクト) に追加します。
- ・「**モニター・グループ**」ビューを使用して、モニター・グループを分析仕様 (MONINSPC オブジェクト) に追加します。
- ・「**モニター仕様と CICS システムとの間のリンク**」ビュー (LNKSMSCS オブジェクト) および「**モニター仕様と CICS システム・グループとの間のリンク**」ビュー (LNKSMSCG オブジェクト) を使用して、モニター仕様を CICS システムまたは CICS システム・グループに関連付けます。LNKSxSCG パラメーターについて詳しくは、属性およびリソース・テーブルに関する特別な考慮事項: [LNKSxSCG レコード \(LNKSMSCG、LNKSRSCG、LNKSWSCG\)](#) を参照してください。

「**マップ (Map)**」ボタンを使用すると、データ・リポジトリ内のモニター定義のビジュアル・マップを表示できます。定義済みオブジェクトと関連のリストを表示するには、「**グループ内のモニター定義**」ビュー (MONINGRP オブジェクト)、「**モニター仕様内のモニター・グループ**」ビュー (MONINSPC オブジェクト)、「**モニター仕様と CICS システムとの間のリンク**」ビュー (LNKSMSCS オブジェクト)、および「**モニター仕様と CICS システム・グループとの間のリンク**」ビュー (LNKSMSCG オブジェクト) を使用します。

次に行うこと

モニター要件を識別したなら、それらを CICSplex SM に定義しなければなりません。

これについては、[リソース・モニター](#)で説明しています。

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す。この資料の他の言語版を IBM から入手できる場合があります。ただし、これを入手するには、本製品または当該言語版製品を所有している必要がある場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービス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。IBM 製品、プログラムまたはサービスに代えて、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せん。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

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を提供し、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。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、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、お客様自身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 Director of Licensing

IBM Corporation

North Castle Drive, MD-NC119 Armonk,

NY 10504-1785

United States of America

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。

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。したがって、他社製品に関す

る実行性、互換性、またはその他の要求については確認できません。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をお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、類似する個人や企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

プログラミング・インターフェース情報

CICS には、プログラミング・インターフェースと見なすことのできる資料と、プログラミング・インターフェースと見なすことのできない資料があります。

オンライン製品資料の以下のセクションには、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のサービスを取得するプログラムをお客様が作成するためのプログラミング・インターフェースが含まれています。

- [アプリケーションの開発](#)
- [システム・プログラムの開発](#)
- [CICS TS セキュリティー](#)
- [外部インターフェースに向けた開発](#)
- [アプリケーション開発のリファレンス](#)
- [リファレンス: システム・プログラミング](#)
- [リファレンス: 接続](#)

オンライン製品資料の以下のセクションには、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のプログラミング・インターフェースとして意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が含まれています。

- [トラブルシューティングおよびサポート](#)
- [CICS TS 診断参照](#)

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のサービスを取得するプログラムをお客様が作成するためのプログラミング・インターフェースが以下のマニュアルに含まれています。

- [アプリケーション・プログラミング・ガイドおよびアプリケーション・プログラミング・リファレンス](#)
- [Business Transaction Services](#)
- [Customization Guide](#)
- [C++ OO Class Libraries](#)
- [Debugging Tools Interfaces Reference](#)
- [Distributed Transaction Programming Guide](#)
- [External Interfaces Guide](#)
- [Front End Programming Interface Guide](#)

- IMS Database Control Guide
- インストール・ガイド
- セキュリティー・ガイド
- Supplied Transactions
- CICSplex SM Managing Workloads
- CICSplex SM Managing Resource Usage
- CICSplex SM アプリケーション・プログラミング・ガイドおよび CICSplex SM アプリケーション・プログラミング・リファレンス
- CICS における Java アプリケーション

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のプログラミング・インターフェースとして意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が以下のマニュアルに含まれています。

- Data Areas
- Diagnosis Reference
- Problem Determination Guide
- CICSplex SM Problem Determination Guide

商標

IBM、IBM ロゴおよび ibm.com[®] は、世界の多くの国で登録された International Business Machines Corporation の商標または登録商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社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。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社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。

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

Microsoft、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です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

製品資料に関するご使用条件

これらの資料は、以下のご使用条件に同意していただける場合に限りご使用いただけます。

適用範囲

IBM Web サイトの「ご利用条件」に加えて、以下のご使用条件が適用されます。

個人使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

商用使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

権利

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM これらの資料の内容 についていかなる保証もしません。これらの資料は、特定物として現存するままの状態 で提供され、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なしで提供されます。

IBM オンラインでのプライバシー・ステートメント

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品 (ソフトウェア・オファリング) では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項をご確認ください。

CICSplex SM Web ユーザー・インターフェース (メイン・インターフェース) の場合:

このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、認証、お客様の利便性の向上、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名、およびその他の個人情報を、セッションごとの Cookie および持続的な Cookie を使用して収集する場合があります。これらの Cookie を無効にすることはできません。

CICSplex SM Web ユーザー・インターフェース (データ・インターフェース) の場合:

このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、認証、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名またはその他の個人情報を、セッションごとの Cookie を使用して収集する場合があります。これらの Cookie を無効にすることはできません。

CICSplex SM Web ユーザー・インターフェース (「Hello World」ページ) の場合:

このソフトウェア・オファリングは、展開される構成に応じて、個人情報を収集しないセッションごとの Cookie を使用する場合があります。これらの Cookie を無効にすることはできません。

CICS Explorer の場合:

このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、お客様の利便性の向上、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名、およびその他の個人情報を、セッションごとの設定および持続的な設定を使用して収集する場合があります。これらの設定を無効にすることはできませんが、ユーザー・パスワードの暗号化形式でのディスクへの保管は、サインオン中にチェック・ボックスにチェック・マークを付けることによるユーザーの明示的な操作によってのみ有効化することができます。

この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。

このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、『IBM オンラインでのプライバシー・ステートメント』 (<http://www.ibm.com/privacy/details/jp/ja/>) の『クッキー、ウェブ・ビー

コン、その他のテクノロジー』および『IBM Software Products and Software-as-a-Service Privacy Statement』 (<http://www.ibm.com/software/info/product-privacy>) を参照してください。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。
なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アクション定義ビュー [73](#)
アクション・ボタン [2](#)
アクティブ・モニターの仕様 [80](#)
アプリケーション・プログラミング・インターフェース (API)
[3](#)
一時記憶域モデル定義ビュー [60](#)
一時データ・キュー定義ビュー [60](#)
インストール計画 [43](#)
エージェント・コード、MAS [7](#)
エンティティ・タイプ
 CICSplex SM [39](#)
オブジェクト
 TERMDEF [60](#)
 ACTDEF [73](#)
 APACTV [73](#)
 APSPEC [73](#)
 ATOMDEF [60](#)
 BATCHREP [47](#)
 BUNDDEF [60](#)
 CICSplex [48](#)
 CMAS [48](#)
 CMASD [48](#)
 CMASPLEX [48](#)
 CMDMPAPS [73](#)
 CMDMSAPS [73](#)
 CMTCMDEF [47](#)
 CMTCMLND [48](#)
 CMTCMLNK [48](#)
 CMTPMLND [48](#)
 CMTPMLNK [48](#)
 CONNDEF [60](#)
 CPLEXDEF [47, 48](#)
 CPLXCMAS [47](#)
 CSYSDEF [49](#)
 CSYSDEF オブジェクト [49](#)
 CSYSGRP [49](#)
 CSYSGRP オブジェクト [49](#)
 DB2CDEF [60](#)
 DB2EDEF [60](#)
 DB2TDEF [60](#)
 DOCDEF [60](#)
 EJCODEF [60](#)
 EJDJDEF [60](#)
 ENQMDEF [60](#)
 EVALDEF [73](#)
 FENODDEF [60](#)
 FEPODEF [60](#)
 FEPRODEF [60](#)
 FETRGDEF [60](#)
 FILEDEF [60](#)
 FSEGDEF [60](#)
 IPCONDEF [60](#)
 JRNLEDEF [60](#)

オブジェクト (続き)

JRNMDEF [60](#)
LIBDEF [60](#)
LNKSMSCG [80](#)
LNKSMSCS [80](#)
LNKSRSCG [73](#)
LNKSRSCS [73](#)
LSRDEF [60](#)
MAPDEF [60](#)
MAS [50](#)
MONDEF [80](#)
MONGROUP [80](#)
MONINGRP [80](#)
MONINSPC [80](#)
MQCONDEF [60](#)
PARTDEF [60](#)
PERIODEF [44, 49, 76](#)
PIPELINE [60](#)
POLMON [80](#)
PROCDEF [60](#)
PROFDEF [60](#)
PROGDEF [60](#)
PRTNDEF [60](#)
RASGNDEF [58](#)
RASINDSC [58](#)
RASPROC [58](#)
RDSCPROC [58](#)
RESDESC [58](#)
RESGROUP [58, 64](#)
RESINDSC [59](#)
RESINGRP [59](#)
RQMDEF [60](#)
RTAACTV [73](#)
RTADEF [73](#)
RTAGROUP [73](#)
RTAINAPS [73](#)
RTAINGRP [73](#)
RTAINSPC [74](#)
RTASPEC [74](#)
SESSDEF [60](#)
STAINGRP [74](#)
STATDEF [74](#)
SYSLINK [49, 59](#)
SYSRES [59](#)
TCPDEF [60](#)
TDQDEF [60](#)
TRANDEF [60](#)
TRNCLDEF [60](#)
TSMDEF [60](#)
TYPTMDEF [60](#)

[カ行]

外部通知、RTA [70](#)
外部メッセージ、RTA [70](#)
概要 [1](#)
環境サービス・システム・サービス (ESSS) [8](#)

管理下アプリケーション・システム (MAS)
定義 [7](#)
記述内のリソース・グループ・ビュー [59](#)
区画セット定義ビュー [60](#)
グループ、CICS システム [31](#), [49](#), [70](#)
グループ・ビュー内のモニター定義 [80](#)
グローバルゼーション・サポート [37](#)
グローバル・エンキュー・ビュー [60](#)
言語サポート [37](#)
構成
CMAS [47](#)
構成管理ビュー [48](#)

[サ行]

サンプリング間隔
モニター [82](#)
時間帯 [4](#), [44](#)
「時間枠定義 (time period definitions)」ビュー [44](#), [49](#), [76](#)
システム・グループ [31](#)
システム・グループ定義ビュー [49](#)
システム使用可能性モニター (SAM)
外部通知のカスタマイズ [71](#)
実装 [76](#)
自動化 [71](#)
説明 [70](#)
システム・リンク・ビュー [59](#)
自動インストール
アクティブな分析定義の [77](#)
モニター定義の [83](#)
リソースの [65](#)
ジャーナル定義ビュー [60](#)
ジャーナル・モデル定義ビュー [60](#)
集約
グループ [13](#)
仕様とシステム・グループとの間のリンク・ビュー [73](#)
仕様ビュー [74](#)
セキュリティ
計画 [43](#)
BAS [65](#)
セッション定義ビュー [60](#)
セット不整合 [63](#)
操作オブジェクト [9](#)
総称アラート、SNA
CMAS の考慮事項 [32](#)
総称名
CICSplex SM [39](#)
その機能 [1](#)

[タ行]

単一システム・イメージ (SSI)
それを提供する上での CMAS の役割 [8](#)
定義 [1](#)
単一制御ポイント
定義 [1](#)
段階的な実装 [41](#)
端末定義ビュー [60](#)
通知、外部、RTA [70](#)
データ・リポジトリ [4](#), [10](#)
データ・リポジトリ初期化ジョブ [44](#)
動的ルーティング [2](#)
トポロジー

トポロジー (続き)
管理 [48](#)
トポロジー管理メニュー [48](#)
トランザクション・クラス定義ビュー [60](#)
トランザクション定義 (TRANDEF) ビュー [60](#)

[ナ行]

入力条件定義ビュー [60](#)

[ハ行]

パートナー定義ビュー [60](#)
配置済みエンタープライズ Java アーカイブ定義ビュー [60](#)
パイプライン定義ビュー [60](#)
バッチ処理されるリポジトリ更新機能
説明 [4](#)
EXTRACT ルーチン [67](#)
「バッチ・リポジトリ更新ジョブ (batched repository-update job)」ビュー [47](#)
パフォーマンスの考慮事項
モニターに関する [81](#)
CMAS [32](#)
CMAS 間リンク [33](#)
ビジネス・アプリケーション・サービス (BAS)
オブジェクト [9](#)
機能 [53](#)
説明 [53](#)
定義 [2](#)
リソース・インストール [53](#)
リソース定義の作成 [63](#)
EXTRACT ルーチン [67](#)
ビュー
時間枠定義 [44](#)
CMAS 構成管理 [47](#)
評価定義ビュー [73](#)
ファイル・セグメント定義ビュー [60](#)
ファイル定義ビュー [60](#)
プログラム定義ビュー [60](#)
プロセス・タイプ定義ビュー [60](#)
プロファイル定義ビュー [60](#)
文書テンプレート定義ビュー [60](#)
分析グループ [73](#)
分析仕様 [74](#)
分析定義 [73](#)
分析点仕様内の RTA グループ・ビュー [73](#)
分析点仕様に関連付けられたインストール済みの分析定義ビュー [73](#)
分析点モニター (APM)
実装 [77](#)
自動化 [69](#)
説明 [72](#)
保守ポイント CMAS [37](#)
保存期間
モニター [82](#)

[マ行]

マップ・セット定義ビュー [60](#)
命名規則
エンティティ [39](#)
スターター・セット [39](#)
メニュー

メニュー (続き)
トポロジー管理 [48](#)
ADMCNFIG [48](#)
CMAS 構成管理メニュー [47](#)
モニター
概要 [79](#)
サンプリング間隔 [82](#)
定義 [3](#)
モニター間隔 [82](#)
CICS モニター機能 (CMF) [84](#)
モニター間隔 [82](#)
モニター・グループ [80](#)
モニター・グループ・ビュー [80](#)
モニター仕様と CICS システムとの間のリンク [80](#)
モニター仕様とシステム・グループとの間のリンク [80](#)
モニター仕様内のモニター・グループ・ビュー [80](#)
モニター定義 [80](#)
モニター定義ビュー [80](#)

[ヤ行]

要求モデル定義ビュー [60](#)

[ラ行]

ランタイム MAS 表示 (MAS) ビュー [50](#)
リアルタイム分析 (RTA)
外部メッセージ [70](#)
概要 [69](#)
計画 [76](#)
システム使用可能性モニター (SAM) [70](#), [76](#)
実装 [76](#)
定義 [3](#)
分析点モニター (APM) [72](#), [77](#)
AOC CICS Automation [70](#)
MAS リソース・モニター (MRM) [71](#), [77](#)
SNA 総称アラート [70](#)
リソース
自動インストール [65](#)
動的インストール [65](#)
リソース・オブジェクト・データ・マネージャー (RODM)
インターフェースの活動化 [83](#)
リソース記述処理ビュー [58](#)
「リソース記述内のリソース割り当て (Resource assignments in resource description)」ビュー [58](#)
リソース記述ビュー [58](#)
「リソース・グループ (Resource group)」(RESGROUP) ビュー [64](#)
リソース・グループ内のリソース定義ビュー [59](#)
リソース・グループ・ビュー [58](#)
リソース検査 [63](#)
リソース状況機能
活動化 [83](#)
リソース定義の作成 [63](#)
リソースのインストール
自動 [65](#)
動的 [65](#)
リソースの妥当性検査 [63](#)
リソースの動的インストール [65](#)
リソース割り当て
使用するまでのマイグレーション [67](#)
リソース割り当て処理ビュー [58](#)
リソース割り当てビュー [58](#)

リンク、通信
CMAS-CMAS [33](#)
「ローカル CMAS 認知の CMAS (CMASs known to local CMAS)」ビュー [48](#)
ローカル MAS [7](#)

[ワ行]

ワークロード管理 (WLM)
定義 [2](#)

[数字]

1 次 CMAS 内の RTA 仕様ビュー [73](#)
2 次 CMAS 内の RTA 仕様ビュー [73](#)

A

ADMCNFIG メニュー [48](#)
AOC CICS Automation [70](#)
ATOMDEF オブジェクト [60](#)
ATOMSERVICE 定義ビュー [60](#)

B

BATCHREP オブジェクト [47](#)
BUNDDDEF オブジェクト [60](#)
BUNDLE 定義ビュー [60](#)

C

CICS システム [26](#)
CICS システム・オブジェクト [26](#)
CICS システム・グループ
識別 [49](#)
定義 [49](#)
複数の CICSplex の代わりに使用 [29](#)
CICS システム定義ビュー [49](#)
CICS システムと CICS システム・グループとの間のリンク・ビュー [49](#)
CICS システムのマップ、生成 [26](#)
CICS システム・リソース・ビュー [59](#)
CICS モニター機能 (CMF) [84](#)
CICS-配置 JAR ファイル定義ビュー [60](#)
CICSplex
識別 [28](#)
定義 [6](#), [48](#)
複数定義する場合 [29](#)
CICSplex SM アドレス・スペース (CMAS)
リリース・レベル [8](#)
「CICSplex 定義中の CMAS (CMAS in CICSplex definitions)」ビュー [47](#)
CICSplex トポロジー
管理 [48](#)
定義 [48](#)
「CICSplex の定義 (CICSplex definitions)」ビュー [47](#), [48](#)
CICSplex のトポロジーの管理 [48](#)
CICSplex のトポロジーの定義 [48](#)
CICSplex) ビュー・オブジェクト [48](#)
CMAS [8](#)
CMAS オブジェクト [48](#)
「CMAS から CMAS へのリンク (CMAS to CMAS link)」ビュー [48](#)

「CMAS から CMAS へのリンク定義 (CMAS to CMAS link definitions)」ビュー [47](#)
「CMAS から CMAS へのリンクの詳細 (CMAS to CMAS link detail)」ビュー [48](#)
「CMAS から MAS へのリンク (CMAS to MAS link)」ビュー [48](#)
「CMAS から MAS へのリンクの詳細 (CMAS to MAS link detail)」ビュー [48](#)
「CMAS 管理の CICSplex (CICSplexes managed by CMAS)」ビュー [48](#)
「CMAS 管理の CICSplex (CMASs managing CICSplex)」ビュー [48](#)
CMAS 構成管理ビュー [47](#)
CMAS 構成の管理 [47](#)
CMAS 構成の定義 [47](#)
CMAS 詳細ビュー [48](#)
CMASPLEX オブジェクト [48](#)
CMTCMDEF オブジェクト [47](#)
CMTCMLND) オブジェクト [48](#)
CMTCMLNK オブジェクト [48](#)
CMTPMLND オブジェクト [48](#)
CMTPMLNK オブジェクト [48](#)
CONNDEF オブジェクト [60](#)
CPLEXDEF オブジェクト [47, 48](#)
CPLXCMAS オブジェクト [47](#)
CSYSGRP オブジェクト [49](#)

D

DB2 エントリー定義ビュー [60](#)
DB2 接続定義ビュー [60](#)
DB2 トランザクション定義ビュー [60](#)
DB2CDEF オブジェクト [60](#)
DB2EDEF オブジェクト [60](#)
DB2TDEF オブジェクト [60](#)
DOCDEF オブジェクト [60](#)

E

EJCODEF オブジェクト [60](#)
EJDJDEF オブジェクト [60](#)
ENQMODE オブジェクト [60](#)
EXTRACT ルーチン [67](#)
EYU9DXDUT [44](#)

F

FENODDEF オブジェクト [60](#)
FEPI ターゲット・リスト定義ビュー [60](#)
FEPI ノード・リスト定義ビュー [60](#)
FEPI プール定義ビュー [60](#)
FEPI プロパティ定義ビュー [60](#)
FEPODEF オブジェクト [60](#)
FEPRODEF オブジェクト [60](#)
FETRGDEF オブジェクト [60](#)
FILEDEF オブジェクト [60](#)
FSEGDEF オブジェクト [60](#)

G

graphql
api [13](#)

I

IBM CICS Explorer [4](#)
IPCONDEF オブジェクト [60](#)
IPIC 接続定義ビュー [60](#)
ISC/MRO 接続定義ビュー [60](#)

J

JRNLDDEF オブジェクト [60](#)
JRNMDDEF オブジェクト [60](#)

L

LIBRARY 定義ビュー [60](#)
LNKSMSCG オブジェクト [80](#)
LNKSMSCS オブジェクト [80](#)
LSR プール定義ビュー [60](#)
LSRDEF オブジェクト [60](#)

M

MAPDEF オブジェクト [60](#)
MAS エージェント・コード [7](#)
MAS リソース・モニター (MRM)
実装 [77](#)
自動化 [72](#)
説明 [71](#)
MONDEF オブジェクト [80](#)
MONGROUP オブジェクト [80](#)
MONINGRP オブジェクト [80](#)
MONINSPC オブジェクト [80](#)
MQCONDEF オブジェクト [60](#)

N

NetView [8, 32](#)

P

PARTDEF オブジェクト [60](#)
PERIODEF オブジェクト [44, 49](#)
PIPELINE オブジェクト [60](#)
POLMON オブジェクト [80](#)
PROCDEF オブジェクト [60](#)
PROFDEF オブジェクト [60](#)
PROGDEF オブジェクト [60](#)
PRTNDEF オブジェクト [60](#)

R

RASGNDEF オブジェクト [58](#)
RASINDSC オブジェクト [58](#)
RASPROC オブジェクト [58](#)
RDSCPROC オブジェクト [58](#)
RESDESC オブジェクト [58](#)
RESGROUP オブジェクト [58](#)
RESINDSC [59](#)
RESINGRP オブジェクト [59](#)
RQMDEF オブジェクト [60](#)
RTA インストール済みの分析と状況定義ビュー [73](#)
RTA グループ内の RTA 定義ビュー [73](#)

RTA グループ内の状況定義ビュー [74](#)
RTA グループ・ビュー [73](#)
RTA 仕様と CICS システムとの間のリンク・ビュー [73](#)
RTA 仕様内の RTA グループ・ビュー [74](#)
RTA 定義ビュー [73](#)
RTA 分析点仕様ビュー [73](#)

S

SESSDEF オブジェクト [60](#)
SNA 総称アラート
 CMAS の考慮事項 [8, 32](#)
STATDEF ビュー [74](#)
SYSLINK オブジェクト [49, 59](#)
SYSLINK ビュー [49](#)
SYSRES オブジェクト [59](#)

T

TCPIP オブジェクト [60](#)
TCPIP サービス定義ビュー [60](#)
TERMDEF オブジェクト [60](#)
TRANDEF オブジェクト [60](#)
TRNCLDEF オブジェクト [60](#)
TSMDEF オブジェクト [60](#)
TYPTMDEF オブジェクト [60](#)

W

Web ユーザー・インターフェース [4](#)
Web ユーザー・インターフェース (WUI)
 グローバリゼーション・サポート [37](#)
 配置する場所 [37](#)
WebSphere MQ 接続定義ビュー [60](#)
WUI サーバーについての計画 [37](#)

[特殊文字]

アドレス・スペース (CMAS)
 インストールする場所 [32](#)
 管理 [47](#)
 構成 [47](#)
 他の CMAS へのリンク [33](#)
 定義 [47](#)
 パフォーマンスの考慮事項 [32](#)
 保守ポイント [37](#)
 ESSS の使用 [8](#)
 NetView [8](#)
オブジェクト・モデル、推奨される使用法
 ビジネス・アプリケーション・サービス [67](#)
 モニター [84](#)
 リアルタイム分析 [77](#)
操作ビュー [2](#)
(CMASD オブジェクト [48](#))

