

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

*CICS* の構成



## 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  
Configuring CICS

### 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

この PDF について.....	xi
<b>第 1 章 CICS トポロジー.....</b>	<b>1</b>
CICSplex 内の CICS 領域の役割.....	3
<b>第 2 章 CICS リソース.....</b>	<b>5</b>
CICS リソースの定義方法.....	5
リソースを管理するためのコマンド.....	10
相互通信のための共用リソース.....	12
リソース定義のセキュリティ.....	12
リソースの監査.....	14
<b>第 3 章 共用データ・セット、CSD、および SYSIN のセットアップ.....</b>	<b>21</b>
CSD 構成の計画.....	21
CSD ディスク・スペースの計算.....	22
CSD の初期化.....	23
さらに大きな CSD の作成.....	25
CSD 属性の定義.....	25
同じ CICS 領域からの共用ユーザー・アクセス.....	26
複数の CICS 領域からのユーザー・アクセスの分担.....	27
CICS 領域内の CSD の複数ユーザー (非 RLS).....	28
単一 MVS イメージ内の複数の CICS 領域による CSD の共用 (非 RLS).....	28
多重 MVS 環境内での CSD の共用 (非 RLS).....	29
CICS 領域またはバッチ領域 (非 RLS) をまたぐ 1 つの CSD の複数ユーザー.....	29
異なる CICS リリース間での CSD の共用.....	30
Db2 を使用する CICS 領域間での CSD の共用.....	30
CICS 提供の互換性グループ.....	30
CSD アクセスを制限する他の要因.....	31
RLS アクセスと非 RLS アクセスでの CSD 管理の違い.....	31
CSD の読み取り保全性の指定.....	32
CSD のファイル制御属性の指定.....	32
CSD バッチ・ユーティリティー DFHCSDUP に対する RLS の影響.....	32
バックアップとリカバリーの計画.....	33
緊急時再始動中のトランザクション・バックアウト.....	36
トランザクションの動的バックアウト.....	36
リカバリーに関するその他の考慮事項.....	36
CEDA コマンドの同期点の基準.....	36
オフライン・ユーティリティー・プログラム DFHCSDUP による CSD へのアクセス.....	36
RDO コマンドのロギング.....	37
SPI コマンドのロギング.....	38
CICS で CSD を有効にする方法.....	39
RDO トランザクションのインストール.....	40
日本語機能の定義のインストール.....	40
<b>第 4 章 CICS 領域のセットアップ.....</b>	<b>41</b>
データ・セットの定義.....	41
CICS データ・セットのセットアップ.....	41
一時記憶域データ・セットのセットアップ.....	48
一時データ用のデータ・セットのセットアップ.....	51
Setting up CICS log streams.....	55

カタログ式データ・セットの設定.....	66
補助トレース・データ・セットの設定.....	76
ダンプ・データ・セットの定義.....	78
ユーザー・ファイルの定義.....	82
CDBM GROUP コマンド・データ・セットの定義.....	94
CMAC メッセージ・データ・セットの定義.....	96
WS-AT データ・セットの定義.....	97
デバッグ・プロファイル・データ・セットのセットアップ.....	98
データ・セットの暗号化.....	102
CICS システム初期化パラメーターの指定.....	105
CICS 領域をセットアップするためのシステム初期設定パラメーター.....	106
DFHSIT マクロ・パラメーターの指定.....	112
デフォルトのシステム初期設定テーブル.....	114
SIT のアSEMBル.....	123
CICS プログラムとテーブルのバージョンの選択.....	124
システム初期設定パラメーターの処理.....	125
CICS に対するシステム初期設定パラメーターの提供.....	125
システム初期設定の制御キーワードの使用.....	126
始動と再始動の制御.....	130
システム始動時に使用されたシステム初期設定パラメーターに関する情報の取り出し.....	137
機能切り替えの指定.....	138
共通構成ファイルのセットアップ.....	139
領域レベル構成ファイルのセットアップ.....	140
CICS の始動.....	141
CICS 領域のアドレス・スペース・ストレージ制限の設定.....	142
サンプル始動ジョブ・ストリームの使用.....	143
サンプル CICS 始動プロシージャ.....	152
デバッグ・ツールを使用するための CICS の準備.....	153
デバッグのための CICS 領域の準備.....	153
CICS 領域の始動.....	154
始動前のシステム初期設定パラメーターの指定.....	155
バッチ・ジョブとしての CICS の開始.....	156
開始タスクとしての CICS の始動.....	156
始動時のシステム初期設定パラメーターのオーバーライド.....	158
CICS の始動に関するシステム・コンソール・メッセージ.....	159
CICS Transaction Server によるソフトウェアの価格設定のためのリソース使用状況の収集.....	163
<b>第 5 章 CICS データ共用サーバーのセットアップ.....</b>	<b>165</b>
AXM システム・サービスの定義と開始.....	165
一時記憶域サーバーのセットアップと実行.....	166
一時記憶域データ共用サーバーの概要.....	166
一時記憶域データ共用のための一時記憶域プールの定義.....	168
TS サーバー領域の定義.....	170
キュー・サーバー自動 ALTER 処理.....	176
共用 TS キュー・サーバーのコマンド.....	177
キュー・プールのアンロードと再ロード.....	179
カップリング・ファシリティ・データ・テーブル・サーバーのセットアップと実行.....	180
カップリング・ファシリティ・データ・テーブル・サーバーの概要.....	181
カップリング・ファシリティ・データ・テーブル.....	183
カップリング・ファシリティ・データ・テーブル・サーバー領域の定義と開始.....	187
カップリング・ファシリティ・データ・テーブル・サーバー領域の制御.....	197
カップリング・ファシリティ・データ・テーブル・プールを削除または空にする.....	202
カップリング・ファシリティ・データ・テーブル・プールのアンロードと再ロード.....	203
領域状況サーバーのセットアップと稼働.....	205
リスト構造の定義.....	206
領域状況サーバーの開始.....	208
領域状況サーバーの制御.....	210

領域状況サーバー・プールの削除.....	215
名前付きカウンター・サーバーのセットアップと実行.....	215
名前付きカウンター・サーバーの概要.....	216
名前付きカウンター・オプション・テーブルの定義.....	218
名前付きカウンター・サーバーのリスト構造の定義.....	221
名前付きカウンター・サーバー領域の定義と開始.....	222
名前付きカウンター・サーバー領域の制御.....	226
名前付きカウンター・プールを削除または空にする.....	229
名前付きカウンター・プールのサイズの変更.....	229
名前付きカウンター・プールのアンロードと再ロード.....	230
名前付きカウンター・プールのリスト構造のダンプ.....	231
カップリング・ファシリティ・サーバーの操作.....	231
カップリング・ファシリティ・サーバー・メッセージのモニター.....	231
カップリング・ファシリティのストレージ管理.....	232
プール構造の管理.....	234
サーバー接続管理.....	235
システム管理プロセスの CICS サーバー・サポート.....	237
システム管理リスト構造の再作成.....	237
システム管理リスト構造の二重化.....	238

## 第 6 章リソースの定義.....241

リソース定義バッチ・ユーティリティー DFHCSDUP.....	241
CICS Transaction Server for z/OS, バージョン 5 リリース 6 と旧リリースの間での CSD の共用...	242
バッチ・プログラムとして DFHCSDUP を起動するためのサンプル・ジョブ.....	242
内部エラーが検出された後の DFHCSDUP でのコマンド処理.....	244
自動インストール.....	245
自動インストール・モデル.....	245
自動インストール制御プログラム.....	245
z/OS Communications Server 端末の自動インストール.....	245
自動インストールする端末の決定.....	246
自動インストールおよび z/OS Communications Server.....	247
z/OS Communications Server の自動インストールの実装.....	251
自動インストールされた端末定義のリカバリーと再始動.....	253
MVS コンソールの自動インストール.....	258
MVS コンソールの自動インストールの実装.....	259
MVS コンソールの自動インストール制御プログラム.....	260
APPC 接続の自動インストール.....	261
APPC 接続の自動インストールの実装.....	262
接続の自動インストールのモデル定義.....	262
接続用自動インストール制御プログラム.....	263
接続の自動インストールにおけるリカバリーと再始動.....	263
IPIC 接続の自動インストール.....	264
自動インストール・プログラム、マップ・セット、および区分セット.....	264
プログラムの自動インストールの実装.....	264
プログラムの自動インストールのカタログ.....	266
プログラムの自動インストールのモデル定義.....	266
プログラムの自動インストール制御プログラム.....	266
プログラムの自動インストール、リカバリー、および再始動.....	267
モデル端末定義の自動インストール.....	268
ジャーナルの自動インストール.....	268
マクロ・リソース定義.....	268
CICS 管理テーブルとマクロの概要.....	268
CICS 管理テーブルでのリソースの定義.....	278
CICS バンドルの定義.....	283
バンドルにデプロイ可能な成果物.....	285
CICS バンドル内リソースの特性.....	290
バンドル内の zFS 成果物の参照.....	297

アプリケーション・バージョンの専用リソース.....	298
CICS バンドルのマニフェストの内容.....	304
バンドルのスコーピング.....	307
CICS リスタート後の OSGi バンドル・リカバリー.....	309
バンドルのセキュリティー.....	310
CICS プロジェクトでの変数.....	312
変数およびプロパティ・ファイルの定義.....	314
端末リソースの定義.....	315
z/OS Communications Server 端末の定義.....	316
順次 (BSAM) 装置.....	317
コンソール・デバイスの定義.....	320
z/OS Communications Server 持続セッション・サポートの定義.....	322
リソース定義のインストール.....	324
CICS が初期化されるときに発生すること.....	324
INSTALL コマンドを使用したときの動作.....	324
限られた数のデータ定義をインストールする方法.....	325
重複するリソース定義名.....	325
ATOMSERVICE リソース定義のインストール.....	326
接続定義のインストール.....	327
Db2 接続定義のインストール.....	327
Db2 接続リソースの定義の検査.....	328
Db2 エントリー定義のインストール.....	328
Db2 エントリー・リソースの定義の検査.....	329
Db2 トランザクション定義のインストール.....	329
Db2 トランザクション・リソースの定義の検査.....	329
エンキュー・モデル定義のインストール.....	330
ファイル定義のインストール.....	330
IPCONN 定義のインストール.....	330
CEDA を使用した LIBRARY リソース定義のインストール.....	331
パートナー定義のインストール.....	331
セッション定義のインストール.....	332
一時データ・キュー定義のインストール.....	332
既存の一時データ・キュー定義の置換.....	332
一時データ・キューの使用不可化.....	333
端末定義のインストール.....	333
端末定義の検査.....	334
URIMAP リソース定義のインストール.....	334
WEBSERVICE リソース定義のインストール.....	335

## 第 7 章 共用データ・テーブルの構成..... 337

データ・テーブルの使用計画.....	337
CICS 保守データ・テーブルのパフォーマンス.....	337
ユーザー保守データ・テーブルのパフォーマンス.....	337
共用データ・テーブル用のストレージの使用.....	337
共用データ・テーブルを使用する際の MVS JCL 要件.....	339
データ・テーブルとして使用するファイルの選択.....	339
データ・テーブル選択のために統計を使用する.....	340
データ・テーブルのセキュリティー検査.....	343
共用データ・テーブルのサポートの使用準備.....	344
データ・テーブルのリソース定義.....	346
CICS 保守データ・テーブルのリソース定義.....	346
ユーザー保守データ・テーブルのリソース定義.....	347
データ・テーブルに関する EXEC CICS コマンド.....	347
データ・テーブルに関する CEMT コマンド.....	348

## 第 8 章 プラットフォームのセットアップ..... 351

CICS プラットフォームの設計.....	352
-----------------------	-----

プラットフォーム用の zFS の準備.....	355
プラットフォームの作成.....	357
プラットフォームのデプロイ .....	358
<b>第 9 章 CMCI のセットアップ.....</b>	<b>361</b>
CICSplex SM での CMCI のセットアップ.....	361
WUI 領域の CMCI の構成.....	362
CMCI JVM サーバーを使用するための WUI 領域の構成.....	367
CICS バンドル・デプロイメント API 用の CMCI JVM サーバーの構成.....	371
CICSplex での複数の CMCI JVM サーバーのセットアップ.....	373
CICSplex SM WUI サーバーと CMCI JVM サーバーの間の構成パラメーターのマッピング.....	373
CMCI でのレコード・カウント警告.....	377
CMCI のストレージ要件の見積もり.....	381
CMCI JVM サーバーに対するクライアント・ホワइटリストの定義.....	382
スタンドアロン CICS 領域での CMCI のセットアップ.....	384
スタンドアロン CICS 領域での CMCI のセキュリティの構成.....	386
<b>第 10 章 イベント処理のセットアップ.....</b>	<b>389</b>
<b>第 11 章 Link3270 ブリッジの構成.....</b>	<b>391</b>
Link3270 システム初期設定パラメーターの定義.....	391
ブリッジ機能の定義.....	391
機能の定義.....	391
ブリッジ機能名の定義.....	392
特定のブリッジ機能名の定義.....	394
TCTUA の初期化.....	394
ブリッジ機能プロパティへのアクセス .....	394
<b>第 12 章 EXCI の構成.....</b>	<b>399</b>
EXCI の静的ルーティングのセットアップ.....	399
EXCI の動的ルーティングのセットアップ.....	399
CICS への接続の定義.....	400
EXCI の CONNECTION リソース定義.....	400
EXCI 接続の SESSIONS リソース定義.....	401
EXCI 接続の状態の照会.....	403
EXCI ユーザー置換可能モジュール.....	404
EXCI オプション・テーブル DFHXCOPT の使用.....	406
<b>第 13 章 CICS ONC RPC のセットアップ.....</b>	<b>413</b>
CICS ONC RPC セットアップ・タスク.....	414
CICS ONC RCP データ・セットの作成.....	414
ダンプ・フォーマット用の JCL エントリー.....	414
CICS バージョン間のマイグレーション.....	415
z/OS Communications Server データ・セットの変更.....	415
CICS への CICS ONC RPC リソースの定義.....	415
CICS ONC RPC トランザクション用のトランザクション定義.....	415
追加の別名トランザクション用のトランザクション定義.....	415
CICS ONC RPC プログラムのプログラム定義.....	416
ユーザー作成プログラムのプログラム定義.....	416
リモート CICS プログラムのプログラム定義.....	417
マップ・セット定義.....	417
一時データ定義.....	417
XLT 定義.....	417
<b>第 14 章 接続マネージャーを使用した CICS ONC RPC の構成.....</b>	<b>419</b>
接続マネージャーの開始.....	419

接続マネージャー BMS パネルの使用.....	420
CICS ONC RPC が無効な場合の接続マネージャーの開始.....	421
CICS ONC RPC が有効な場合の接続マネージャーの開始.....	421
CICS ONC RPC ステータスの更新.....	422
CICS ONC RPC ステータスの変更.....	422
CICS ONC RPC の有効化.....	423
オプションの設定および変更.....	423
オプションの検証、保管、およびアクティブ化.....	425
CICS ONC RPC が有効な場合.....	425
4 タブルの定義、保管、変更、および削除.....	425
4 タブルの属性の定義.....	426
新規の 4 タブル定義を保管する.....	429
既存の 4 タブル定義の変更.....	430
既存の 4 タブル定義の削除.....	430
4 タブルの登録.....	430
登録に関する制限.....	430
4 タブルの登録抹消.....	431
4 タブルを 1 つずつ登録抹消する.....	431
リストから 4 タブルを登録抹消する.....	432
CICS ONC RPC の無効化.....	433
CICS 通常シャットダウンの場合.....	433
CICS 即時シャットダウンの場合.....	434
CICS ONC RPC データ・セットの更新.....	434
CICS ONC RPC 定義レコードの更新.....	435
4 タブルのリストの処理.....	436
4 タブルの属性の変更.....	437
別名リストの処理.....	438
<b>第 15 章リカバリーと再始動のための構成.....</b>	<b>441</b>
ロギングおよびジャーナリング.....	441
MVS へのログ・ストリームの定義.....	442
レプリカ生成ログ・ストリームの定義.....	442
システム・ログ・ストリームの定義.....	442
順方向リカバリー・ログ・ストリームの定義.....	454
ログのログの定義.....	455
夏時間調整による変更の影響.....	457
CICS 管理リソースのリカバリーの構成.....	458
Liberty JVM サーバーでのリソースのリカバリー.....	458
トランザクションのリカバリー.....	459
ファイルのリカバリー.....	460
区画内一時データのリカバリー.....	466
区画外一時データのリカバリー.....	469
一時記憶域のリカバリー.....	470
Web サービスのリカバリー.....	471
プログラム・エラー・プログラム (PEP) の使用.....	474
<b>第 16 章 REXX の構成.....</b>	<b>477</b>
REXX サポートの構成.....	477
RFS ファイル・プールの作成.....	477
リソース定義の作成.....	477
LSRPOOL 定義の確認.....	478
CICSTART メンバーの更新.....	478
CICS 初期設定 JCL の変更.....	479
RFS ファイル・プールの形式設定.....	480
インストールの検証.....	481
ヘルプ・ファイルの作成.....	482
REXX Db2 インターフェースの構成.....	483



REXX/CICS システムの定義と管理.....	483
REXX/CICS 許可コマンドと許可コマンド・オプション.....	483
システム・プロファイル exec.....	483
MVS PDS REXX 許可ライブラリー.....	484
許可ユーザーの定義.....	484
システム・オプションの設定.....	484
REXX ファイル・システム (RFS) ファイル・プールの定義.....	484
CICSTART のための PLT エントリーの作成.....	484
セキュリティー出口.....	484
パフォーマンスの考慮事項.....	486
セキュリティー.....	486
REXX/CICS による複数トランザクション ID のサポート.....	487
REXX/CICS ファイル・セキュリティー.....	487
REXX/CICS コマンド・レベル・セキュリティー.....	487
REXX/CICS 許可コマンド・サポート.....	487
セキュリティー定義.....	488
<b>第 17 章 IBM Health Checker for z/OS による CICS 構成の検査.....</b>	<b>489</b>
CICS_CEDA_ACCESS.....	489
CICS_JOBSUB_SPOOL.....	490
CICS_JOBSUB_TDQINTRDR.....	490
<b>第 18 章 Parallel Sysplex への CICS のマイグレーション.....</b>	<b>491</b>
Parallel Sysplex で CICS を実装することの利点.....	491
CICSplex、CICSplex SM、および Parallel Sysplex.....	492
Parallel Sysplex の原理.....	493
Parallel Sysplex テクノロジーを直接利用する CICS 機能およびコンポーネント.....	497
CICS ログ・ストリーム.....	497
CICS カップリング・ファシリティー・データ・テーブル.....	499
CICSplex SM シスプレックス最適化ワークロード管理.....	499
MRO に関する XCF.....	500
一時記憶域プールの使用.....	500
名前付きカウンター.....	502
Parallel Sysplex テクノロジーを間接的に利用する CICS 機能およびコンポーネント.....	503
Db2 データ共用.....	503
IBM MQ 共用キュー.....	504
VSAM RLS.....	505
ネットワーキング.....	506
Parallel Sysplex を容易にするその他の CICS 機能およびコンポーネント.....	507
Parallel Sysplex への CICS のマイグレーションの計画.....	508
アプリケーションのアフィニティー.....	509
CICS ワークロード・ルーティングおよび管理.....	510
<b>特記事項.....</b>	<b>515</b>
<b>索引.....</b>	<b>521</b>



## この PDF について

---

この PDF では、CICS TS for z/OS のセットアップ方法について説明します。以下に示す他の PDF は、CICS の特定の領域の構成について説明するものであり、本 PDF とともにそれらも参照する必要がある場合があります。(IBM Knowledge Center では、この情報すべてが、「構成」という 1 つのセクションの下にあります。)「システム初期設定パラメーター・リファレンス」および「リソース・リファレンス」を、この PDF と併読する必要がある場合もあります。CICS TS V5.4 より前は、この PDF の情報は、「*System Definition Guide*」および「*Resource Definition Guide*」に収録されていました。

CICS の領域の構成については、以下の PDF を参照してください。

- SOAP および JSON については、「*Web サービス・ガイド*」を参照してください。
- ONC/RPC インターフェースについては、「*外部インターフェース・ガイド*」を参照してください。
- EXCI については、「*CICS での EXCI の使用*」を参照してください。
- Java および Liberty については、「*CICS での Java アプリケーション*」を参照してください。
- フロントエンド・プログラミング・インターフェースについては、「*フロントエンド・プログラミング・インターフェース・ユーザズ・ガイド*」を参照してください。
- Db2<sup>®</sup> は、*Db2 Guide* にあります。
- DBCTL については、「*IMS DB コントロール・ガイド*」を参照してください。
- 共用データ・テーブルについては、「*共用データ・テーブルの手引き*」を参照してください。
- CICSplex SM については、「*CICSplex SM Administration*」を参照してください。
- BTS については、「*ビジネス・トランザクション・サービス*」を参照してください。
- CICS システム間の接続については、「*相互通信ガイド*」を参照してください。

CICS システム初期設定で使用するパラメーターについての参照情報は、「システム初期設定パラメーター・リファレンス」にあります。

本書で使用する用語および表記の詳細については、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。

### この PDF の作成日

この PDF は、2020 年 5 月 28 日に作成されました。



---

# 第 1 章 CICS トポロジー

CICS アプリケーションおよびアプリケーションが使用するリソースは、相互接続された CICS 領域 間に分散させることができます。CICS 領域を CICS システム・グループおよび CICSplexes にグループ化し、1 つのシスプレックス 内の z/OS システム間に領域を分散させることが可能です。

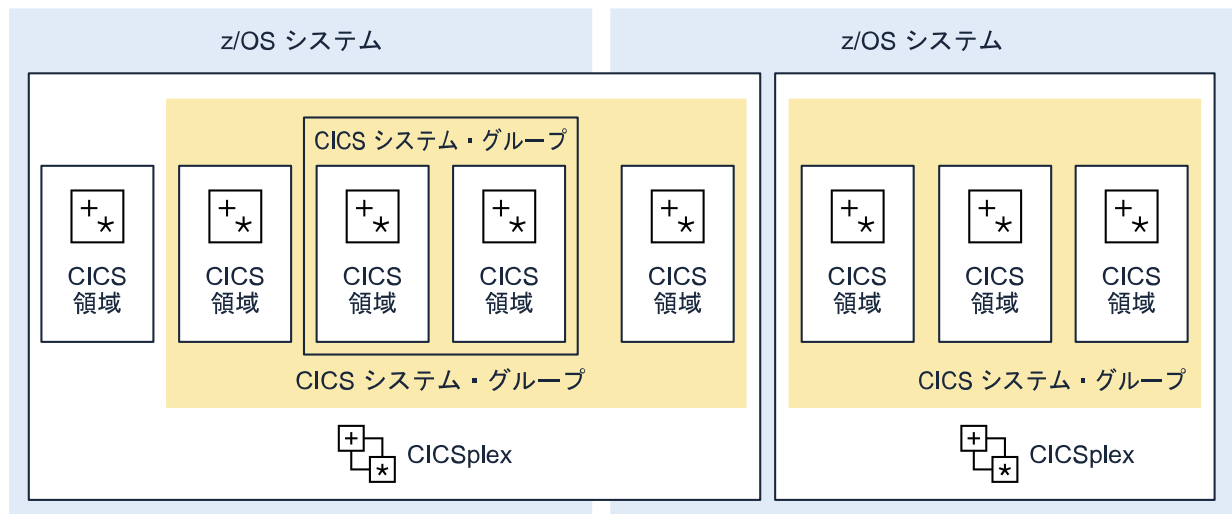


図 1. シスプレックス内で複数の CICS 領域を編成する方法

CICS トポロジを説明する際に、以下の用語が使用されます。

### **シスプレックス**

マルチシステム・ハードウェア・コンポーネントおよびソフトウェア・サービスを通して、お互いに通信し協力しあう z/OS システムの集合。

### **CICS 領域**

それ自体の z/OS アドレス・スペース内で実行される、CICS Transaction Server の名前付きインスタンス。CICS 領域は他の CICS 領域から独立して始動および停止できます。

### **CICSplex**

単一のエンティティとして管理される CICS 領域のグループ。各 CICS 領域は 1 つの CICSplex にのみ所属できます。CICSplex には、シスプレックス内の異なる z/OS システムで実行されている CICS 領域を含めることができます。

### **CICS システム・グループ**

単一のエンティティとして管理できる CICSplex 内の CICS 領域のグループ。システム・グループには、異なる z/OS システムで実行される CICS 領域を含めることができます。CICSplex 内で、各 CICS 領域は複数のシステム・グループに所属でき、システム・グループは他のシステム・グループ内に含めることができます。

## **CICSplex 内の CICS 領域の役割**

多数の CICS 領域が含まれている CICSplex では、領域がサポートしているアプリケーションに対して特定の役割を果たすように領域に割り当てると便利です。

以下に役割を示します。

### **トランスポート所有領域**

CICSplex を通信ネットワークに接続する CICS 領域です。トランスポート所有領域が受信するトランザクション要求は、処理のためにアプリケーション所有領域に渡されます。

### **端末所有領域**

端末や他のデバイス (プリンターも含む) を CICSplex に接続する CICS 領域です。端末所有領域が受信するトランザクション要求は、処理のためにアプリケーション所有領域に渡されます。

### **アプリケーション所有領域**

アプリケーション・プログラムを管理する CICS 領域です。データ要求はデータ所有領域に渡されます。

### **データ所有領域**

ファイルおよびデータベースへのアクセスを管理する CICS アドレス・スペースです。

### **ファイル所有領域**

ファイルへのアクセスを管理する CICS アドレス・スペースです。ファイル所有領域は、データ所有領域の特定のタイプです。

すべての CICS 領域がすべての役割を果たせます。1 つの領域で複数の役割を組み合わせるのが適切な場合もあります。





## 第 2 章 CICS リソース

CICS でプログラムを実行するには、プログラムやデータなどのソフトウェア・リソースを含むシステム・リソースに関する情報、および端末または通信リンクなどのハードウェア・リソースに関する情報を CICS に提供する必要があります。少なくとも、トランザクションを定義する必要があります。

すべてのリソースは属性のセットにより定義されます。属性はリソースのプロパティであり、例えば、ファイルを更新できるかどうか、トランザクションにどのセキュリティ・レベルを設定するか、または CICS が通信できるリモート・システムなどを CICS に通知します。

CSD に保持されているリソース定義は、グループとリストに編成されます。グループとは、CSD 上の関連するリソースの集合です。定義する各リソースはグループに属する必要があります。グループを指定せずにリソースを定義することはできません。リストには、初期スタートまたはコールド・スタート時に CICS がインストールするグループの名前が含まれています。初期スタートまたはコールド・スタート時にインストールする場合、またはグループをより効率的に管理するのに役立つ場合、リストにグループを追加できます。グループはリストに属する必要はなく、独立して定義できます。

### リソースが保持される場所

CICS システム定義 (CSD) ファイルは、CEDA、CICS Explorer®、または DFHCSDUP を介して CICS に定義されているリソースごとのリソース定義レコードが含まれている VSAM データ・セットです。

CSD を使用する稼働中の CICS 領域に影響を及ぼさずに、CSD の内容を変更できます。CICS 領域に定義をインストールすると、CICS はその情報を CSD からコピーし、独自のストレージに保持します。稼働中の領域の定義は、それらの定義を再インストールすることで変更できます。また、新しいリソースをインストールして、さらに定義を追加することもできます。CSD ファイルをリカバリー可能として定義できるため、異常終了が発生したときに未完了の変更はバックアウトされます。また、CSD ファイルとそのリソース定義を、異なる CICS 領域 (リリースが異なる領域も含む) と共用することもできます。CSD の定義について詳しくは、[CICS システム定義データ・セットのセットアップ](#)を参照してください。

CSD に定義できないリソースは、CICS 管理テーブルに保持されます。テーブルとそれらのリソース定義は、CICS テーブル・アセンブリ・マクロ命令を使用して作成されます。各リソースがテーブルに表示されるようにするために、アセンブラー言語マクロ・ステートメントをコーディングし、マクロ・ステートメントのセット全体をアセンブルして、出力をリンク・エディットしてロード・モジュールを作成し、DFHSIT にモジュール接尾部を指定する必要があります。[CICS 管理テーブルでのリソースの定義](#)を参照してください。

## CICS リソースの定義方法

CICS リソースは、CICS Explorer、CICS バンドル、CICSplex® SM ビジネス・アプリケーション・サービス、オンライン・リソース定義 (RDO)、CICS システム・プログラミング・コマンド、DFHCSDUP オフライン・ユーティリティ、自動インストール、またはマクロ・リソース定義を使用して定義できます。リソース定義に関連する方式を比較して、各 CICS リソースを定義する方法を選択します。

リソース定義は、以下のリポジトリに保管できます。

- CICS 領域の CSD ファイル
- CICSplex SM データ・リポジトリ
- zFS (z/OS UNIX ファイル・システム) (CICS バンドル用)
- プログラム・ライブラリーの管理テーブル (マクロ・リソース定義用)

特定のインターフェースを使用して、これらの各リポジトリでリソース定義を処理できます。一部のインターフェースおよびリポジトリでサポートされていない CICS リソース・タイプもあります。

以下のインターフェースを使用して CICS リソースを定義できます。

### CICS Explorer

CICS Explorer を使用して、リソースを定義、インストール、および管理できます。CICS Explorer が CICS システムに接続されている場合、定義は CICS システム定義 (CSD) ファイルに保管され、CSD フ

ファイルからアクティブな CICS システムにインストールされます。CICS Explorer が CICSplex SM に接続されている場合、定義は CICSplex SM データ・リポジトリに保管され、CICS の初期化中に自動的に、または動的に稼働中の CICS システムにインストールされます。

## バンドル

CICS Explorer または IBM Developer for Z を使用して、CICS バンドルにリソースを定義し、パッケージ化できます。CICS はバンドルがデプロイされるとリソースを動的に作成し、CICS システムで単一のユニットとして CICS バンドルを介してそれらのライフサイクルを管理します。CICS バンドルによって、バンドルがデプロイされる CICS 領域に必要なシステム・リソースを指定することもできます。CICS バンドルをスタンドアロン・バンドルとして CICS 領域に個別にデプロイすることも、プラットフォームにデプロイされるアプリケーション・バンドルの一部として使用することも、プラットフォームに直接デプロイすることもできます。CICS バンドルについて詳しくは、[Defining CICS bundles](#) を参照してください。

## CICSplex SM ビジネス・アプリケーション・サービス

CICSplex SM ビジネス・アプリケーション・サービス (BAS) を使用して、リソースの定義および管理ができます。定義は、CICSplex SM データ・リポジトリに保管され、CICS の初期化中に自動的に、または動的に稼働中の CICS システムにインストールされます。CICSplex SM BAS について詳しくは、[BAS の管理](#)を参照してください。

## オンライン・リソース定義 (RDO)

この方式では、提供されているオンライン・トランザクション CEDA、CEDB、および CEDC を使用します。これらを使用して、稼働中の CICS システムでリソースを定義、変更、およびインストールできます。定義は CSD ファイルに保管され、CSD ファイルからアクティブな CICS システムにインストールされます。この方式は、CSD ファイルのリソースを更新します。つまり、稼働中の CICS システムに対してリソースの定義、変更、およびインストールができます。CEDA トランザクションを使用する RDO について詳しくは、[リソース管理トランザクション CEDA コマンド](#)を参照してください。

## EXEC CICS SPI コマンドを使用したシステム・プログラミング

**EXEC CICS CREATE** コマンド、および FEPI リソース用の **EXEC CICS FEPI INSTALL** コマンドを使用して、CSD ファイルとは別個にリソースを作成できます。詳しくは、[リソース定義の作成](#)を参照してください。

## EXEC CICS CSD コマンドを使用したシステム・プログラミング

**EXEC CICS CSD** コマンドを使用して、ユーザー作成プログラムから CSD ファイル内のリソース定義を管理できます。**EXEC CICS CSD** コマンドでは、CEDA CHECK を除くすべての CEDA 機能を実行できます。

## DFHCSDUP オフライン・ユーティリティー

DFHCSDUP は、バッチ・ジョブを使用してリソースを定義、リスト、および変更できるオフライン・ユーティリティーです。DFHCSDUP はバッチ・プログラムとして起動、またはバッチ・モードや TSO 下で実行しているユーザー作成プログラムから起動できます。2 つ目の方式を使用することで、DFHCSDUP に最大 5 つのユーザー出口ルーチンを指定できます。DFHCSDUP ユーティリティーを使用して、CSD ファイル内の定義を変更できます。定義は CSD ファイルに保管されます。DFHCSDUP ユーティリティーについては、[システム定義ファイル・ユーティリティー・プログラム \(DFHCSDUP\)](#)を参照してください。

## 自動インストール (自動インストール)

自動インストールは、ユーザーが提供する "モデル" に基づき新しい定義を動的に作成することで、大量の定義を用意する必要性が最小限に抑えます。自動インストールは、VTAM® 端末、LU6.2 セッション、IPIC 接続、ジャーナル、プログラム、マップ・セット、および区分セットに適用されます。RDO または DFHCSDUP を使用して、"モデル" 定義をセットアップします。モデルを作成すると、CICS はモデルに基づきリソース用の新しい定義を動的に作成してインストールできます。

## マクロ定義

アセンブラー・マクロ・ソースを使用して、CSD に保管できないリソースを定義できます。定義はプログラム・ライブラリーのアセンブル済み制御テーブルに保管され、CICS の初期化中にこのテーブルからインストールされます。

マクロ命令を使用して、モニターおよびシステム・リカバリーを行うための非 VTAM ネットワークと端末、非 VSAM ファイル、データベース、およびリソースを定義する必要があります。CICS マクロについて詳しくは、[マクロ・リソース定義](#)を参照してください。

使用する方式は、定義するリソースによって異なります。使用する定義方式を決定する際に検討する必要がある事項については、7 ページの表 1 を参照してください。各リソースに使用できる方式については、8 ページの表 2 を参照してください。

表 1. リソース定義の方式		
メソッド	利点	欠点
CICS Explorer	<ul style="list-style-type: none"> <li>直感的で使いやすいインターフェース。</li> <li>その他の CICS ツールの統合ポイント。</li> <li>中心的な役割を担うリソース定義。</li> <li>論理的なスコープ設定。</li> <li>分散リソース・インストール。</li> <li>CICS バンドル、BAS、および CICS CSD を処理可能。</li> </ul>	FEPI リソースは、CICS Explorer では定義できません。
バンドル	<ul style="list-style-type: none"> <li>CICS バンドルにいくつかのリソース・タイプを定義して、バンドルがデプロイされると動的に作成されるようにすることができます。</li> <li>CICS 領域に存在する必要があるその他の必要なリソースを指定できます。</li> <li>単一リソースの操作により、アプリケーションをインストール、アンインストール、有効化、および無効化できます。</li> <li>バンドルによってバージョン管理が提供されるため、リソースとアプリケーションの更新を管理できます。</li> <li>リソースの中には、バンドルを使用しなければ定義およびデプロイできないものがあります。</li> <li>非 CICS リソースを、CICS リソースと共に CICS バンドルで作成および管理できます。</li> </ul>	<ul style="list-style-type: none"> <li>個別に定義されたリソースと同じ方法で、バンドルに定義されているリソースの状態を変更することはできません。</li> <li>すべてのアプリケーション・リソースがバンドルでサポートされているわけではありません。</li> </ul>
CICSplex SM BAS	<ul style="list-style-type: none"> <li>中心的な役割を担うリソース定義。</li> <li>論理的なスコープ設定。</li> <li>分散リソース・インストール。</li> </ul>	すべてのアプリケーション・リソースが BAS でサポートされているわけではありません。
RDO	RDO は CICS が稼動中に使用されるため、リソース定義に迅速にアクセスできます。	CEDA はアクティブな CICS システムで作動するため、実動システムで使用する場合には注意する必要があります。いくつかの監査形式を制御メカニズムとして使用します。
EXEC CICS SPI コマンド	単一の管理フォーカル・ポイントから多数の CICS 領域に対する CICS リソースの構成およびインストールができます。また、稼動中の CICS システムを管理するためのアプリケーションを作成することもできます。	CREATE コマンドは CSD ファイルの参照も、CSD ファイルへの記録も行いません。結果の定義は、コールド・スタート時に失われ、それらの定義を CEDA トランザクションで参照することはできません。

表 1. リソース定義の方式 (続き)		
メソッド	利点	欠点
EXEC CICS CSD システム・コマンド	<ul style="list-style-type: none"> <li>ユーザーは、CSD およびインストール済みリソースを管理できる、自分の環境用にカスタマイズしたアプリケーションを作成できます。</li> <li>この方式で更新されるリソースは、CEDA で参照可能です。</li> <li>CICS の以前のリリースと CSD を共用するための互換性モードがサポートされています。</li> </ul>	実装に必要な作業は、その他の一部方式より多くなります。
DFHCSDUP	<ul style="list-style-type: none"> <li>大量のリソースを 1 ジョブで変更または定義できます。</li> <li>リカバリー不能 CSD ファイルが、RLS アクセス・モードを使用して CICS 領域間で共有されている間は、そのファイルに対して DFHCSDUP を実行できます。</li> </ul>	<ul style="list-style-type: none"> <li>アクティブな CICS システムにリソースをインストールすることはできません。</li> <li>RLS モードでアクセスされているリカバリー可能 CSD ファイルを DFHCSDUP を介して更新することはできません。</li> </ul>
自動インストール	大量のリソースがある場合、定義するには多くの時間が必要になります。また、それらのリソースのすべてがその後使用されるわけではない場合、使用されない定義のためのストレージが無駄になります。自動インストールを使用すると、この時間とストレージの無駄を削減できます。	自動インストールの利点を活用するためには、自動インストールの初期セットアップに多少の時間を割く必要があります。
マクロ		<ul style="list-style-type: none"> <li>CICS が稼動中にテーブルに含まれている定義を変更することはできますが、変更したテーブルを使用する場合は、CICS を停止してから再始動する必要があります。</li> <li>マクロ・テーブルを生成するには、時間がかかるアセンブリーを実行する必要があります。</li> </ul>

表 2. リソースおよびリソースを稼動中の CICS システムに定義する方法							
リソース	CICS Explorer	CICSplex SM BAS	RDO、EXEC CICS SPI、および EXEC CICS CSD コマンド	バンドル	DFHCSDUP	自動インストール	マクロ
Atom 文書	はい	はい (ATOMDEF)	はい (ATOMSERVICE)	はい	はい	いいえ	いいえ
バンドル	はい	はい (BUNDLEDEF)	はい (BUNDLE)	該当せず	はい	いいえ	いいえ
接続	はい	はい (CONNDEF)	はい (CONNECTION)	いいえ	はい	LU 6.2 のみ	いいえ
Db2 接続	はい	はい (DB2CDEF)	はい (DB2CONN)	いいえ	はい	いいえ	いいえ
Db2 エントリ	はい	はい (DB2EDEF)	はい (DB2ENTRY)	いいえ	はい	いいえ	いいえ
Db2 トランザクション	はい	はい (DB2TDEF)	はい (DB2TRAN)	いいえ	はい	いいえ	いいえ
文書テンプレート	はい	はい (DOCDEF)	はい (DOCTEMPLATE)	いいえ	はい	いいえ	いいえ
エンキュー・モデル	はい	はい (ENQMDEF)	はい (ENQMODEL)	いいえ	はい	いいえ	いいえ

表 2. リソースおよびリソースを稼働中の CICS システムに定義する方法 (続き)							
リソース	CICS Explorer	CICSplex SM BAS	RDO、EXEC CICS SPI、および EXEC CICS CSD コマンド	バンドル	DFHCSDUP	自動インストール	マクロ
イベント・バイインディングおよびキャプチャー仕様	はい	いいえ	いいえ	はい	いいえ	いいえ	いいえ
イベント処理アダプター	はい	いいえ	いいえ	はい	いいえ	いいえ	いいえ
イベント処理アダプター・セット	はい	いいえ	いいえ	はい	いいえ	いいえ	いいえ
FEPI ノード・リスト	いいえ	はい (FENODEF)	はい (NODELIST)	いいえ	いいえ	いいえ	いいえ
FEPI プール定義	いいえ	はい (FEPODEF)	はい (POOL)	いいえ	いいえ	いいえ	いいえ
FEPI プロパティ・セット	いいえ	はい (FEPRODEF)	はい (PROPERTYSET)	いいえ	いいえ	いいえ	いいえ
FEPI ターゲット・リスト	いいえ	はい (FETRGDEF)	はい (TARGETLIST)	いいえ	いいえ	いいえ	いいえ
ファイル (BDAM)	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	はい (DFHFCT)
ファイル (VSAM)	はい	はい (FILEDEF)	はい (FILE)	はい	はい	いいえ	いいえ
IPIC 接続	はい	はい (IPCONDEF)	はい (IPCONN)	いいえ	はい	はい	いいえ
ジャーナル	はい	はい (JRNLEDEF)	いいえ	いいえ	いいえ	はい	いいえ
ジャーナル・モデル	はい	はい (JRNMDEF)	はい (JOURNALMODEL)	いいえ	はい	いいえ	いいえ
LIBRARY リソース	はい	はい (LIBDEF)	はい (LIBRARY)	はい	はい	いいえ	いいえ
ローカル共用リソース (LSR) プール	はい	はい (LSRDEF)	はい (LSRPOOL)	いいえ	はい	いいえ	いいえ
マップ・セット	はい	はい (MAPDEF)	はい (MAPSET)	いいえ	はい	はい	いいえ
Node.js アプリケーション	はい	いいえ	いいえ	はい	いいえ	いいえ	いいえ
OSGi バンドルおよびサービス	はい	いいえ	いいえ	はい	いいえ	いいえ	いいえ
パッケージ・セット	はい	いいえ	いいえ	はい	いいえ	いいえ	いいえ
区分セット	はい	はい (PRTNDEF)	はい (PARTITIONSET)	いいえ	はい	はい	いいえ
パートナー	はい	はい (PARTDEF)	はい (PARTNER)	いいえ	はい	いいえ	いいえ
パイプライン	はい	はい (PIPEDEF)	はい (PIPELINE)	いいえ	はい	いいえ	いいえ
ポリシー	はい	いいえ	いいえ	はい	いいえ	いいえ	いいえ
プロセス・タイプ	はい	はい (PROCDEF)	はい (PROCESSTYPE)	いいえ	はい	いいえ	いいえ
プロファイル	はい	はい (PROFDEF)	はい (PROFILE)	いいえ	はい	いいえ	いいえ
Programs (プログラム)	はい	はい (PROGDEF)	はい (PROGRAM)	はい	はい	はい	いいえ
リカバリー可能サービス・エレメント	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	はい (DFHRST)
セッション	はい	はい (SESSDEF)	はい (SESSIONS)	いいえ	はい	いいえ	いいえ
TCP/IP サービス	はい	はい (TCPDEF)	はい (TCPIPSERVICE)	いいえ	はい	いいえ	いいえ

表 2. リソースおよびリソースを稼働中の CICS システムに定義する方法 (続き)

リソース	CICS Explorer	CICSplex SM BAS	RDO、EXEC CICS SPI、および EXEC CICS CSD コマンド	バンドル	DFHCSDUP	自動インストール	マクロ
一時記憶域 (マクロで定義)	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	はい (DFHTST)
一時記憶域モデル (リソース定義)	はい	はい (TSMDEF)	はい (TSMODEL)	いいえ	はい	いいえ	いいえ
端末 (非 VTAM)	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	はい (DFHTCT)
端末 (VTAM)	はい	はい (TERMDEF)	はい (TERMINAL)	いいえ	はい	はい	いいえ
トランザクション	はい	はい (TRANDEF)	はい (TRANSACTION)	はい	はい	いいえ	いいえ
トランザクション・クラス	はい	はい (TRNCLDEF)	はい (TRANCLASS)	いいえ	はい	いいえ	いいえ
一時データ・キュー	はい	はい (TDQDEF)	はい (TDQUEUE)	いいえ	はい	いいえ	いいえ
Typeterms	はい	はい (TYPTMDEF)	はい (TYPETERM)	いいえ	はい	いいえ	いいえ
URI マップ	はい	はい	はい (URIMAP)	はい	はい	いいえ	いいえ
Web サービス	はい	はい	はい (WEBSERVICE)	いいえ	はい	はい	いいえ
IBM MQ 接続	はい	はい (MQCONDEF)	はい (MQCONN)	いいえ	はい	いいえ	いいえ
XML 変換	はい	いいえ	いいえ	はい	いいえ	いいえ	いいえ

## リソースを管理するためのコマンド

CEDA または DFHCSDUP の一部として提供されているコマンドを使用して、リソース定義を管理します。これらのコマンドを使用して、定義、削除、コピー、名前変更などを行い、リソースを処理できます。

コマンドを [10 ページの表 3](#) に示します。これらのコマンドの構文と、使用方法については、[システム定義ファイル・ユーティリティー・プログラム \(DFHCSDUP\)](#)を参照してください。

表 3. CEDA および DFHCSDUP コマンド

コマンド	機能	CEDA	DFHCSDUP
ADD	グループ名をリストに追加します。	<a href="#">CEDA ADD コマンド</a>	<a href="#">DFHCSDUP ADD コマンド</a>
ALTER	既存のリソース定義の属性を変更します。	<a href="#">CEDA ALTER コマンド</a>	<a href="#">DFHCSDUP ALTER コマンド</a>
APPEND	リストを、別のリストの末尾にコピーします。	<a href="#">CEDA APPEND コマンド</a>	
CHECK (CEDA のみ)	単一のグループ内、あるいは単一または複数のリスト内の複数のグループ内のリソース定義を、最大で 4 つのリストまでクロス・チェックします。	<a href="#">CEDA CHECK コマンド</a>	
COPY	あるグループから別のグループに 1 つ以上のリソース定義をコピーしたり、1 つのグループ内で 1 つのリソース定義をコピーしたりします。	<a href="#">CEDA COPY コマンド</a>	<a href="#">DFHCSDUP COPY コマンド</a>

表 3. CEDA および DFHCSDUP コマンド (続き)			
コマンド	機能	CEDA	DFHCSDUP
DEFINE	新規リソース定義を作成します。	<a href="#">CEDA DEFINE コマンド</a>	<a href="#">DFHCSDUP DEFINE コマンド</a>
DELETE	1 つ以上のリソース定義を削除します。	<a href="#">CEDA DELETE コマンド</a>	<a href="#">DFHCSDUP DELETE コマンド</a>
DISPLAY (CEDA のみ)	1 つ以上のグループ、リスト、またはグループ内のリソース定義の名前を表示します。	<a href="#">CEDA DISPLAY コマンド</a>	
EXPAND (CEDA のみ)	1 つ以上のグループまたはリスト内のリソース定義の名前を表示します。	<a href="#">CEDA EXPAND コマンド</a>	
EXTRACT (DFHCSDUP のみ)	CSD ファイルのグループまたはリストからリソース定義データを抽出して処理します。		<a href="#">DFHCSDUP EXTRACT コマンド</a>
INITIALIZE (DFHCSDUP のみ)	新たに定義されたデータ・セットを CSD ファイルとして使用できるように準備します。		<a href="#">DFHCSDUP INITIALIZE コマンド</a>
INSTALL (CEDA のみ)	アクティブな CICS システムに、リソース定義またはリソース定義のグループを動的に追加します。	<a href="#">CEDA INSTALL コマンド</a>	
LIST (DFHCSDUP のみ)	CSD ファイルの現在の状況のリストを作成します。		<a href="#">DFHCSDUP LIST コマンド</a>
LOCK (CEDA のみ)	他のオペレーターがグループ、またはリスト内のグループを更新または削除できないようにします。		
MOVE (CEDA のみ)	1 つ以上のリソース定義をあるグループから別のグループに移動します。	<a href="#">CEDA MOVE コマンド</a>	
PROCESS (DFHCSDUP のみ)	CSD ファイルに特定の APAR の保守を適用します。		<a href="#">DFHCSDUP PROCESS コマンド</a>
REMOVE	リストからグループ名を削除します。	<a href="#">CEDA REMOVE コマンド</a>	<a href="#">DFHCSDUP REMOVE コマンド</a>
RENAME (CEDA のみ)	1 つのグループ内でリソース定義を名前変更するか、リソース定義を別のグループに移動すると同時に名前変更します。	<a href="#">CEDA RENAME コマンド</a>	
SCAN (DFHCSDUP のみ)	あるリソースの提供グループおよびユーザー定義グループをスキャンします。提供されたグループ内で一致するリソースの定義は、ユーザー・グループ内で対応する一致したリソースの定義と比較されます。		<a href="#">DFHCSDUP SCAN コマンド</a>
SERVICE (DFHCSDUP のみ)	CSD ファイルに修理保守を適用します。		<a href="#">DFHCSDUP SERVICE コマンド</a>



表 3. CEDA および DFHCSDUP コマンド (続き)			
コマンド	機能	CEDA	DFHCSDUP
UNLOCK (CEDA のみ)	グループまたはリストのロックを解除します。		
UPGRADE (DFHCSDUP のみ)	CSD ファイル上の CICS 提供のリソース定義をアップグレードします (CICS の上位リリースにマイグレーションする場合など)。		<a href="#">DFHCSDUP UPGRADE コマンド</a>
USERDEFINE	独自のデフォルトで新規リソース定義を作成します。	<a href="#">CEDA USERDEFINE コマンド</a>	<a href="#">DFHCSDUP USERDEFINE コマンド</a>
VERIFY (DFHCSDUP のみ)	グループおよびリストの内部ロックを除去します。		<a href="#">DFHCSDUP VERIFY コマンド</a>
VIEW (CEDA のみ)	既存のリソース定義の属性を表示します。	<a href="#">CEDA VIEW コマンド</a>	

## 相互通信のための共用リソース

リモート・システムにあるが、ローカル CICS システムによってアクセスされるリソースは、リモートとローカルの両方のシステムで定義する必要があります。ローカルとリモート・システムにおいて、複数の CSD ファイルで定義が重複するのを避けるため、1つの CSD ファイルでリソース定義を作成し、そのファイルをローカルとリモート・システムで共用することができます。各共用リソースに必要な CSD ファイル・レコードは1つのみであるため、これにより、ディスク・ストレージと保守が削減されます。

二重目的のリソース定義を使用する場合、リソース定義グループ内のリソースを再編成することを検討できます。例えば、現在、CICS トランザクション所有領域 (TOR) のすべてのリソースが含まれているグループと、CICS アプリケーション所有領域 (AOR) のすべてのリソースを含むグループの2つのグループがあるとしてします。

共有リソース定義を使用する場合、3つのグループを使用できます。最初のグループには TOR 固有のリソースを入れ、2番目のグループには AOR 固有のリソースを入れ、3番目のグループには TOR と AOR の両方にインストールされるリソースを含めることができます。

これらのリソースは、ローカルとリモートの両方として定義する必要があります。TOR に定義がインストールされると、CICS は SYSIDNT 名と REMOTESYSTEM 名を比較します。これらが異なる場合、リモート・トランザクション定義が作成されます。定義が AOR にインストールされると、CICS は REMOTESYSTEM 名と SYSIDNT 名を比較します。これらが同じ場合、ローカル・トランザクション定義がインストールされます。

二重目的リソース定義は、以下のリソースで使用できます。

- ファイル
- Programs (プログラム)
- 一時記憶域モデル (TSMODEL)
- 端末
- 一時データ・キュー (TDQUEUE)
- トランザクション

## リソース定義のセキュリティ

CICS には、リソース定義を不正な使用から保護するのに役立つ多数の機能が用意されています。

リソース定義のセキュリティを検討する場合、以下を考慮します。

### CSD のリソース定義へのアクセス制限

CSD のリソース定義への読み取り/書き込みアクセス権限を少数のユーザーに限定する必要があります。以下はその方法です。



- CEDA コマンド LOCK を使用して、リソースのグループを保護する
- CEDA コマンド LOCK を使用して、システム 初期設定パラメーター GRPLIST に指定されているリソース・グループのリストを保護する
- CEDB トランザクションを使用してリソース定義を作成するが、インストールしない
- リソース定義に対する読み取り専用アクセスに対して CEDC トランザクションを使用する

CEDA LOCK および UNLOCK コマンドについて詳しくは、[リソース管理トランザクション CEDA コマンド](#)を参照してください。

## リソース・セキュリティ検査

リソース・セキュリティ検査により、端末オペレーターは権限を持っているリソースにしかアクセスできないようにすることができます。TRANSACTION 定義に対してリソース・セキュリティ検査 (RESSEC) を使用できます。

## 複数の CSD ファイル

異なる CICS システム用に、異なる CSD ファイルを使用できます。ある CICS のユーザーは、別の CICS の CSD ファイルにはアクセスできません。

RDO トランザクションを使用できるシステムではテスト CSD ファイルを使用し、RDO トランザクションを使用できないシステムでは実動 CSD ファイルを使用できます。これにより、権限のないユーザーが実動作業に必要なリソース定義を変更する可能性はなくなります。

## 同じ CSD ファイルに対する読み取り専用定義および更新定義

2つの CSD ファイルがあるということは、複数のシステムで共用されているリソースに対するリソース定義を複製しているということです。RDO の利点は、各リソースに必要な定義が1つのみであるということです。1つの CSD ファイルが複数の CICS システムで共用されるように定義し、1つのみが書き込み権限を持つようにすることができます。これを行うには、CSDACC 初期設定パラメーターを使用して、別のシステムとは異なるように1つの CSD ファイルを定義します。CSD ファイルを使用できるが、更新できないシステムの場合、次を指定します。

```
CSDACC=READONLY
```

CSD を更新する予定のシステムの場合、次を指定します。

```
CSDACC=READWRITE
```

定義をインストールするには、READONLY 権限が必要です。これにより、DISPLAY および VIEW コマンドも使用できます。ADD、APPEND、ALTER、COPY、MOVE、および RENAME コマンドを使用するには、READWRITE 権限が必要です。CSD ファイルの定義について詳しくは、[リソース管理トランザクション CEDA コマンド](#)を参照してください。

## グループまたはリストへのアクセスの制御 – LOCK および UNLOCK

RDO には、グループやリストへのアクセスを制御する機能も備わっているため、同じシステム内のユーザーに異なるタイプのアクセス権限を付与できます。これには LOCK コマンドおよび UNLOCK コマンドを使用します。

LOCK コマンドおよび UNLOCK コマンドを使用して、グループやリストへの更新アクセスを制御できるため、同じオペレーター ID を持つオペレーターしか変更を行えないようにすることができます。

ロックは CSD ファイルに保持され、CICS の再起動後も有効なままになります。ロックは、CICS 汎用アプリケーション ID (APPLID システム 初期設定パラメーターによって指定) と、ユーザーのオペレーター ID (OPIDENT) を組み合わせて特定されるユーザーが所有します。

OPIDENT は、該当するユーザーが RDO 用に使用される端末にサインオンするときに関連付けられるものです。OPIDENT について詳しくは、[CICS セグメント](#)を参照してください。

サインオンしていないユーザー、または異なる OPIDENT を持つユーザーは、ロックされたグループを変更することになる操作は実行できません。ただし、ロックされたグループに対する以下の操作は、どのユーザーでも行えます。

- COPY
- CHECK
- DISPLAY
- INSTALL
- VIEW

ロックは UNLOCK コマンドを使用して除去できますが、これを行えるのは、同じシステム上に存在する、同じオペレーター ID を持つユーザーのみです。

TYPETERM のグループと AUTINSTMODEL TERMINAL のグループにロックを掛けることをお勧めします。

## RDO トランザクションへのアクセスの制御

CEDA、CEDB、および CEDC トランザクションに対する推奨アクセスを以下に示します。

- CEDC は読み取り専用コマンドしか使用できないため、かなり広いアクセス権限を付与できます。
- CEDB は読み取り専用コマンドに加え、CSD ファイルの変更を行えるため、制限する必要があります。
- CEDA はさらに制限し、少数の人しかアクティブな CICS システムと CSD ファイルの両方を変更できないようにする必要があります。

## リソースのインストール

CEDA を使用する権限を持つユーザーは、CICS システムにあらゆるリソースをインストールできます。トランザクション自体を使用するためのユーザー権限をチェックする以外に、CICS は CEDA トランザクションでコマンドおよびリソースのセキュリティ検査は行いません。

これは、CREATE コマンドを使用してリソースをインストールするトランザクションには当てはまりません。そのような場合、CICS は以下のように処理します。

- コマンド・セキュリティを使用して、ユーザーが CREATE コマンドを使用する権限を持っていることを検査します。詳しくは、[CICS command security](#) を参照してください。
- リソース・セキュリティを使用して、ユーザーが対象のリソースを変更する権限を持っていることを検査します。詳しくは、[リソース定義のセキュリティ](#) を参照してください。

## リソースの監査

リソース・シグニチャーは定義とインストール・シグニチャーを組み合わせたもので、これを使用して、リソースの定義時、インストール時、および最終変更時に、その詳細情報を収集することでリソースの監査と管理を行います。

リソースが定義、インストール、および最終変更された時期に関する情報を表示できるため、問題の判別に役立ちます。詳細情報は、リソースの監査および追跡を向上させ、**EXEC CICS INQUIRE SPI** コマンドと **CEMT INQUIRE** コマンドを使用して、CICS Explorer ビュー、CICSplex SM ビュー、CEDA パネルに表示できます。以下のリソース・タイプでリソース・シグニチャーがサポートされています。

ATOMSERVICE  
BUNDLE  
CONNECTION  
DB2CONN  
DB2ENTRY  
DB2TRAN  
DOCTEMPLATE  
ENQMODEL  
EPADAPTER  
EPADAPTERSET

EVENTBINDING  
FILE  
IPCONN  
JOURNALMODEL  
JVMSEVER  
LIBRARY  
MQCONN  
MQINI  
NODEJSAPP  
OSGIBUNDLE  
PIPELINE  
PROFILE  
PROCESSTYPE  
PROGRAM  
TCPIPSERVICE  
TDQUEUE  
TRANCLASS  
TRANSACTION  
TSMODEL  
URIMAP  
WEBSERVICE  
XMLTRANSFORM

これらのリソースについては、定義とインストール・シグニチャーを比較して、発信元を特定できます。詳しくは、以下のトピックを参照してください。

### リソース定義の定義シグニチャー

定義シグニチャーは、CSD ファイルまたは CICSplex SM EYUDREP データ・リポジトリにおける各リソースの定義または変更の時期、方法、および実行者に関する詳細情報をキャプチャーします。定義シグニチャーは、リソースに対して変更が行われるたびに更新されます。こうした詳細情報を使用して、リソース変更を検出し、監査や問題の修正を行えます。

定義シグニチャーは、CICS Explorer のビュー、CEDA および CEMT パネル、CICSplex SM BAS ビュー、**EXEC CICS INQUIRE** コマンド、および DFHCSDUP レポートに表示されます。以下に、定義シグニチャーのフィールドを記します。

### DEFINESOURCE

リソース定義のソース。CHANGEAGENT に依存する DEFINESOURCE 値。

### DEFINETIME

リソース定義が **DEFINE**、**USERDEFINE**、**COPY**、**MOVE**、または **RENAME** コマンドを使用して作成された時刻。**ALTER** コマンドを使用して既存のリソースを変更すると、DEFINETIME で示された値は変更されません。CEDA パネルには、DATFORM システム 初期設定パラメーターで指定したフォーマットで日付が表示されます。

### CHANGEAGENT

リソースが定義または最終変更された方法。以下のいずれかの方式です。

#### 自動インストール

自動インストール

#### Csdapi

CEDA、DFHEDAP とのプログラマブル・インターフェース、または **EXEC CICS CSD** コマンド

#### Csdbatch

DFHCSDUP

#### Drepapi

CICSplex SM **BAS** API コマンド

## Dynamic (動的)

リソースは、以下によって生成されました。

PIPELINE スキャン (URIMAP または WEBSERVICE)  
DFHWBTL または DFHWBMS を使用した CICS Web テンプレート管理 (DOCTEMPLATE)  
トランザクション ID が指定された DB2ENTRY リソース定義のインストール (DB2TRAN)  
XSDBIND が指定された ATOMSERVICE リソース定義のインストール (XMLTRANSFORM)  
INITQNAME が指定された MQCONN リソース定義のインストール (MQINI)

## システム

CICS または CICSplex SM システム

## Table

テーブル定義

## CHANGEAGREL

リソース定義を定義する際に使用された、またはリソース定義に最後に変更を加えた際に使用された CICS システムのレベル。

## CHANGETIME

リソース定義が最後に変更された時刻。初めてリソースを定義する場合、CHANGETIME 値は DEFINETIME 値と等しくなります。CEDA パネルには、DATFORM システム 初期設定パラメーターで指定したフォーマットで日付が表示されます。

## CHANGEUSRID

リソース定義を定義、または最後に変更したユーザーの ID。

CEDA DISPLAY および EXPAND GROUP パネルで個々のリソースやリソース・グループの定義シグニチャーを表示するには、PF2 を押します。直前の CEDA コマンド・パネルに戻るには、もう一度 PF2 を押します。

指定したすべてのリソースの定義シグニチャーの要約を表示するには、**SIGSUMM** パラメーターを **DFHCSDUP LIST** コマンドに追加します。定義シグニチャーのフィールドは、このコマンドで **OBJECTS** オプションを使用するとリソース属性と一緒に表示されます。また **DFHCSDUP EXTRACT** コマンドを使用すると、CSD ファイルから定義シグニチャーのフィールドが取り出されます。

CICS TS 4.1 より前の CICS リリースで定義されたリソースの場合、今回の CICS リリース以降で変更しない限りは定義シグニチャーで表示される情報はありません。リソースを変更しても、DEFINETIME フィールドはブランクのままです。

## リソース定義のインストール・シグニチャー

インストール・シグニチャーには、各リソースがインストールされた時期、方法、およびインストール実行者が示されます。

インストール・シグニチャーは、リソースの **CEMT INQUIRE** コマンドの拡張ビュー・パネルの CICS Explorer ビュー、CICSplex SM 操作ビューに表示されます。**EXEC CICS INQUIRE** コマンドを使用することもできます。以下に、インストール・シグニチャーのフィールドを記します。

## INSTALLAGENT

リソースがインストールされた方法。以下のいずれかの方式です。

### 自動インストール

自動インストール

### バンドル

バンドル・デプロイメント

### Createspi

**EXEC CICS CREATE** コマンド

### Csdapi

CEDA、DFHEDAP とのプログラマブル・インターフェース、または **EXEC CICS CSD** コマンド

## Dynamic (動的)

インストール済みリソースは、以下によって生成されました。

PIPELINE スキャン (URIMAP または WEBSERVICE)

DFHWBTL または DFHWBMS を使用した CICS Web テンプレート管理 (DOCTEMPLATE)  
 トランザクション ID が指定された DB2ENTRY リソース定義のインストール (DB2TRAN)  
 XSDBIND が指定された ATOMSERVICE リソース定義のインストール (XMLTRANSFORM)  
 INITQNAME が指定された MQCONN リソース定義のインストール (MQINI)

#### Grplist

GRPLIST INSTALL

#### システム

CICS または CICSplex SM システム

#### Table

テーブル定義

#### INSTALLTIME

リソースがインストールされた時刻。

#### INSTALLUSRID

リソースをインストールしたユーザーの ID。

#### リソース・シグニチャー・フィールド値の要約

リソース・シグニチャー・フィールドには、稼働中の CICS システムにリソース定義をインストールするために使用されるすべての方式に関連する内容が含まれています。

表 4. リソース・シグニチャーの内容。第 1 部。				
リソース・シグニチャー・フィールド	GRPLIST INSTALL	CEDA INSTALL または EXEC CICS CSD INSTALL	EXEC CICS CREATE	自動インストール
DEFINESOURCE	CSD GROUP	CSD GROUP	EXEC CICS CREATE を発行したプログラム名	自動インストール・ユーザー・プログラム名
DEFINETIME	CSD レコード作成のタイム・スタンプ	CSD レコード作成のタイム・スタンプ	リソースが作成された時刻	自動インストールの時刻
CHANGEAGENT	CSDAPI または CSDBATCH	CSDAPI または CSDBATCH	CREATESPI	AUTOINSTALL
CHANGEAGREL	CHANGEAGENT システムの CICS リリース ("nnnn" というフォーマット) <sup>1</sup>	CHANGEAGENT システムの CICS リリース ("nnnn" というフォーマット) <sup>1</sup>	CHANGEAGENT システムの CICS リリース ("nnnn" というフォーマット) <sup>1</sup>	CHANGEAGENT システムの CICS リリース ("nnnn" というフォーマット) <sup>1</sup>
CHANGETIME	CSD レコード変更のタイム・スタンプ	CSD レコード変更のタイム・スタンプ	リソースが作成された時刻	自動インストールの時刻
CHANGEUSRID	CHANGEAGENT を実行したユーザー ID	CHANGEAGENT を実行したユーザー ID	EXEC CICS CREATE を実行したユーザー ID	自動インストールを実行したユーザー ID
INSTALLAGENT	GRPLIST	CSDAPI	CREATESPI	AUTOINSTALL
INSTALLTIME	最後にコールド・スタートした時刻	インストールの時刻	リソースが作成された時刻 (ウォーム・スタートした場合、以前の実行から)	自動インストールの時刻 (ウォーム・スタートした場合、以前の実行時から)
INSTALLUSRID	ジョブ・ステップ・ユーザー ID	インストールを実行したユーザー ID	EXEC CICS CREATE を実行したユーザー ID	自動インストールを実行したユーザー ID

表 5. リソース・シグニチャーの内容。第 2 部。				
リソース・シグニチャー・フィールド	テーブル定義	システムによって定義される	動的に作成される	BUNDLE によって作成される
DEFINESOURCE	テーブル名	SYSTEM	詳細については、 <a href="#">19 ページの表 7</a> を参照してください。	BUNDLE name (BUNDLE 名)
DEFINETIME	最後にコールド・スタートした時刻	CICS 始動の時刻	リソースが生成された時刻	BUNDLE から継承

表 5. リソース・シグニチャーの内容。第 2 部。(続き)				
リソース・シグニチャー・フィールド	テーブル定義	システムによって定義される	動的に作成される	BUNDLE によって作成される
CHANGEAGENT	TABLE	SYSTEM	DYNAMIC	BUNDLE から継承
CHANGEAGREL	テーブル・アセンブリからの CICS リリース ("nnnn" というフォーマット) <a href="#">1</a>	CHANGEAGENT システムの CICS リリース ("nnnn" というフォーマット) <a href="#">1</a>	CHANGEAGENT システムの CICS リリース ("nnnn" というフォーマット) <a href="#">1</a>	BUNDLE から継承
CHANGETIME	最後にコールド・スタートした時刻	CICS 始動の時刻	リソースが生成された時刻	BUNDLE から継承
CHANGEUSRID	ジョブ・ステップ・ユーザー ID	ジョブ・ステップ・ユーザー ID	リソースを生成したユーザー ID (詳しくは、 <a href="#">19 ページの表 7</a> を参照)	BUNDLE から継承
INSTALLAGENT	TABLE	SYSTEM	DYNAMIC	BUNDLE
INSTALLTIME	最後にコールド・スタートした時刻	CICS 始動の時刻	インストール済みリソースが生成された時刻	BUNDLE から継承
INSTALLUSRID	ジョブ・ステップ・ユーザー ID	ジョブ・ステップ・ユーザー ID	インストール済みリソースを生成したユーザー ID (詳しくは、 <a href="#">19 ページの表 7</a> を参照)	BUNDLE から継承

表 6. リソース・シグニチャーの内容。第 3 部。			
リソース・シグニチャー・フィールド	CICSplex SM (EXEC CICS CREATE)	プラットフォーム用に CICSplex SM により作成	CICSplex SM SYSLINK
DEFINESOURCE	CPSMVnn。nn は CICSplex SM BAS リソース定義のバージョン (VER 属性)	管理パーツ名	SYSLINK
DEFINETIME	EYUDREP の CREATETIME	リソースが作成された時刻	リソースがインストールされた時刻
CHANGEAGENT	DREPAPI または SYSTEM	CREATESPI	DREPAPI
CHANGEAGREL	CHANGEAGENT システムの CICS リリース ("nnnn" というフォーマット) <a href="#">1</a>	CHANGEAGENT システムの CICS リリース ("nnnn" というフォーマット) <a href="#">1</a>	CHANGEAGENT システムの CICS リリース ("nnnn" というフォーマット) <a href="#">1</a>
CHANGETIME	EYUDREP の CHANGETIME	リソースが作成された時刻	リソースがインストールされた時刻
CHANGEUSRID	EYUDREP の CHANGEUSRID <a href="#">2</a>	作成を実行したかユーザー ID か、CICSplex SM から渡されたユーザー ID	SYSLINK インストールを要求したユーザー ID <a href="#">2</a>
INSTALLAGENT	CREATESPI	CLOUD	CREATESPI
INSTALLTIME	リソースが作成された時刻 (ウォーム・スタートした場合、以前の実行から)	リソースが作成された時刻	リソースがインストールされた時刻
INSTALLUSRID	作成を実行したかユーザー ID か、CICSplex SM から渡されたユーザー ID	作成を実行したかユーザー ID か、CICSplex SM から渡されたユーザー ID	SYSLINK インストールを要求したユーザー ID

注:

1. CICS リリースの "nnnn" というフォーマットは 4 桁の固有 ID です。例えば、0660 は CICS TS 4.1 の ID です。
2. EYUDREP データ・リポジトリ内の CICSplex SM BAS リソース定義の場合、CICS セキュリティーがアクティブであるなら、CHANGEUSRID フィールドにはリソース定義に最後に変更を加えたユーザー ID が含まれます。CICS セキュリティーがアクティブでない場合、CHANGEUSRID フィールドはブランクとなります。

表 7. 動的リソースの <i>CHANGEUSRID</i> 、 <i>DEFINESOURCE</i> 、および <i>INSTALLUSRID</i> フィールドの内容				
動的リソース	生成者	<b>CHANGEUSRID</b>	<b>DEFINESOURCE</b>	<b>INSTALLUSRID</b>
DB2TRAN	トランザクション ID が指定された DB2ENTRY リソース定義のインストール	DB2ENTRY をインストールしたユーザー ID	DB2ENTRY 名	DB2ENTRY をインストールしたユーザー ID
DOCTEMPLATE	DFHWBTL または DFHWBBMS を使用した CICS Web テンプレート管理	DFHWBBMS または DFHWBTL を実行したユーザー ID	DFHWBBMS または DFHWBTL	DFHWBBMS または DFHWBTL を実行したユーザー ID
DUMPCODE	<b>SET SYSDUMPCODE ADD</b> コマンドまたは <b>SET TRANDUMPCODE ADD</b> コマンドによる DUMPCODE リソース定義のインストール	<b>SET SYSDUMPCODE ADD</b> または <b>SET TRANDUMPCODE ADD</b> コマンドを発行したユーザー ID	<b>SET SYSDUMPCODE ADD</b> または <b>SET TRANDUMPCODE ADD</b> コマンドを発行したプログラムの名前	<b>SET SYSDUMPCODE ADD</b> または <b>SET TRANDUMPCODE ADD</b> コマンドを発行したユーザー ID
MQINI	INITQNAME が指定された MQCONN リソース定義のインストール	MQCONN をインストールしたユーザー ID	MQCONN name (MQCONN 名)	MQCONN をインストールしたユーザー ID
PROGRAM	@CICSProgram アノテーションを含む Liberty アプリケーションのインストール	Liberty アプリケーションへのリンクをインストールしたユーザー ID	アプリケーションが CICS バンドルにインストールされていない場合の BUNDLE 名または \$WLPAPP	Liberty アプリケーションへのリンクをインストールしたユーザー ID
URIMAP	PIPELINE スキャン	PIPELINE スキャンを実行したユーザー ID	PIPELINE name (PIPELINE 名)	PIPELINE スキャンを実行したユーザー ID
WEBSERVICE	PIPELINE スキャン	PIPELINE スキャンを実行したユーザー ID	PIPELINE name (PIPELINE 名)	PIPELINE スキャンを実行したユーザー ID
XMLTRANSFORM	XSDBIND が指定された ATOMSERVICE リソース定義のインストール	ATOMSERVICE をインストールしたユーザー ID	ATOMSERVICE name (ATOMSERVICE 名)	ATOMSERVICE をインストールしたユーザー ID





## 第3章 共用データ・セット、CSD、および SYSIN のセットアップ

システム定義データ・セット (CSD) および SYSIN データ・セットは一般的に、複数の CICS 領域で共用されます。CSD にはリソース定義が格納され、SYSIN データ・セットには領域の開始情報 (例えば、システム初期設定パラメーター値) が入れられます。これらのデータ・セットを作成するには、用意されているジョブを使用してください。

### このタスクについて

CSD は、ほとんどのリソース定義に必要です。初めて CSD を作成する場合は、21 ページの『CSD 構成の計画』に示されている手順を実行してください。このセクションの残りの部分では、それらの手順について詳細に説明します。

以前のリリースの CICS で CSD を既に使用している場合は、CSD をアップグレードして、CICS Transaction Server for z/OS, バージョン 5 リリース 6 で新しくなった CICS リソース定義を組み込んでください。CSD のアップグレードについては、[システム定義ファイル・ユーティリティー・プログラム \(DFHCSDUP\)](#) を参照してください。

SYSIN データ・セットには、始動中に CICS を構成するための情報が含まれています。データ・セットを作成して、標準的な始動パラメーターのセットを複数の CICS 領域で共用することができます。また、必要に応じて、特定の CICS 領域の始動ジョブ・ストリームでそれらの値をオーバーライドすることもできます。CICS には、SYSIN データ・セットを作成するための DFHCOMDS ジョブが用意されています。詳しくは、『インストール』の『CICS データ・セットの作成』を参照してください。

### CSD 構成の計画

CSD を作成する前に、構成を計画する必要があります。

#### 手順

1. ディスク・スペースの必要量を決めます。
2. CSD を RLS モードで使用するか非 RLS モードで使用するかを決めます。

RLS モードで CSD を開いた場合は、複数の CICS 領域で同時に CSD を更新できます。ただし、CSD がリカバリー可能データ・セットとして定義されている状況で、バッチ・ユーティリティー DFHCSDUP によって CSD を更新する場合は、DFHCSDUP を実行する前に CICS 領域で CSD を静止する必要があります。

CSD で RLS を使用する場合は、システム初期設定パラメーターとして CSDRLS=YES を指定してください。85 ページの『VSAM レコード・レベル共用 (RLS)』を参照してください。
3. CSD で必要なバックアップ/リカバリー手順を決めます。CSD ではオープン時バックアップ (BWO) の機能を使用できるので、DFSMS コンポーネントは、データ・セットが更新モードで開かれている間に CSD をバックアップできます。

BWO を使用するには、DFSMS 1.2 以降の DFSMSHsm コンポーネントと DFSMSdss コンポーネントが有効になっていなければなりません。CSD に対応した ICF カタログ項目が必要であり、SMS 管理ストレージで CSD を定義しておかなければなりません。

  - RLS モードでアクセスする CSD のリカバリー・オプションを定義する場合:
    - ICF カタログで BWO(TYPECICS) を指定して、データ・セットをオープン時バックアップ機能に対応させます。
    - CSD をリカバリー可能データ・セットにするために、ICF カタログで該当する LOG パラメーターを指定します。
  - 非 RLS モードでアクセスする CSD のリカバリー・オプションを定義する場合:

- データ・セットをオープン時バックアップ機能に対応させるために、カタログで BWO(TYPECICS) を指定するか、**CSDBKUP** システム初期設定パラメーターを DYNAMIC に設定します。
- データ・セットをリカバリー可能にするために、ICF カタログ項目で該当する LOG パラメーターを設定するか、**CSDRECOV** システム初期設定パラメーターで該当するオプションを指定します。

デフォルトでは、FILE リソースの属性よりもカタログのリカバリー・オプションのほうが優先されます。カタログでリカバリー・オプションが設定されていない場合、CICS は FILE リソースの属性値を使用します。CICS でカタログではなく FILE リソースのリカバリー・オプションを常に使用する場合は、**NONRLSRECOV** システム初期設定パラメーターを FILEDEF に設定できます。

4. CSD を定義して初期化します。

5. CSD で指定する CICS ファイル処理属性を決めます。

CSD は CICS のファイル管理で制御するデータ・セットですが、CSD のファイル管理リソース定義は、CSDxxxxx システム初期設定パラメーターで指定します (25 ページの『CSD 属性の定義』を参照)。

6. RDO でコマンド・ログを使用するかどうかを決めます。

CICS が RDO のコマンド・ログで使用する CADL、CAIL、CRDI、CSDL、CSFL、CSKL、CSPL、CSRL の各宛先の詳細については、37 ページの『RDO コマンドのロギング』を参照してください。

7. CICS で CSD を有効にします。そのためには、動的割り振りを使用するか、必要な DD ステートメントを CICS 始動ジョブ・ストリームに組み込みます。

CSD の動的割り振りを使用する場合は、完全修飾データ・セット名と処理方法をシステム初期設定パラメーター CSDDSN と CSDDISP でそれぞれ指定します。

CICS の始動後に、RDO トランザクション CEDA、CEDB、CEDC をテストしてください。

## CSD ディスク・スペースの計算

CSD を作成する前に、CSD に定義レコードを組み込むために必要なスペースの量を計算する必要があります。

### 手順

1. ユーティリティ・プログラム DFHCSDUP で CSD を初期化する時に CSD にロードされる CICS 提供の各種リソース定義が 1800 個ほどあることを計算に入れる必要があります。そのための検討事項を以下にまとめます。

a) プログラム、トランザクション、端末などのリソース定義ごとに 1 つのレコードが必要です。

そのような定義レコードのサイズは以下のとおりです。

リソース	定義レコード・サイズ (最大)
ATOMSERVICE	720 バイト
BUNDLE	698 バイト
CONNECTION	260 バイト
CORBASERVER	1375 バイト
DB2CONN	308 バイト
DB2ENTRY	236 バイト
DB2TRAN	198 バイト
DJAR	445 バイト
DOCTEMPLATE	567 バイト
ENQMODEL	447 バイト
FILE	369 バイト
IPCONN	468 バイト
JOURNALMODEL	222 バイト

リソース	定義レコード・サイズ (最大)
JVMSERVER	208 バイト
LIBRARY	925 バイト
LSRPOOL	425 バイト
MAPSET	190 バイト
MQCONN	240 バイト
PARTITIONSET	190 バイト
PARTNER	408 バイト
PIPELINE	959 バイト
PROCESSTYPE	206 バイト
PROFILE	231 バイト
PROGRAM	499 バイト
REQUESTMODEL	1211 バイト
SESSION	296 バイト
TCIPSERVICE	571 バイト
TDQUEUE	331 バイト
TERMINAL	327 バイト
TRANCLASS	192 バイト
TRANSACTION	545 バイト
TSMODEL	308 バイト
TYPETERM	402 バイト
URIMAP	1443 バイト
WEBSERVICE	708 バイト

- b) それぞれのグループに 122 バイトのレコードが 2 つ必要です
  - c) それぞれのグループ・リストに 122 バイトのレコードが 2 つ必要です
  - d) リスト内のそれぞれのグループ名に 68 バイトのレコードが 1 つ必要です
2. 計算したサイズに予備のスペース (約 25%) を追加します。

#### 次のタスク

最終的に得られた計算値を、CSD のための VSAM クラスターを定義する時に使用します。(23 ページの『CSD の初期化』のサンプル・ジョブを参照してください。)

## CSD の初期化

**INITIALIZE** コマンドは、CICS に用意されているリソース定義を使用して CSD を初期設定します。初期設定の後、CICS 管理テーブルからリソース 定義を移行し、独自のリソースの定義を始めることができます。**INITIALIZE** コマンドを使用するのは、CSD の存続期間中に一度だけです。

#### このタスクについて

CSD を使用するには、その前にその CSD を VSAM KSDS データ・セットとして定義し、DFHCSDUP ユーティリティー・プログラムを使用して初期設定する必要があります。CSD を定義して初期設定するためのサンプル・ジョブを使用します。

## 手順

1. サンプル・ジョブに示されているように、**KEYS** パラメーターをコーディングします。  
キーの長さは 22 バイトです。
2. 次のように、必要な CSD ディスク・スペースを計算します。
  - **RECORDS** パラメーターは、CSD に割り振るレコード数を定義します。このパラメーターは、n1 および n2 という 2 つの値を取ります。n1 は 1 次エクステントです。n2 は 2 次エクステントで、n1 で定義した割り振るレコード数を超過した場合にのみ使用されます。このパラメーターの設定方法については、[22 ページの『CSD ディスク・スペースの計算』](#)を参照してください。
  - **RECORDSIZE** パラメーターは、平均レコード・サイズと最大レコード・サイズ (バイト単位) を定義します。用意されているリソース定義のみを含む CSD (**INITIALIZE** および **UPGRADE** コマンドで生成されます) の平均レコード・サイズは、200 バイトです。作成した端末リソース定義項目の比率が、最初の CSD で定義されたものより大きくなると、平均レコード・サイズは大きくなります。サンプル・ジョブに示されているように、最大レコード・サイズは 2000 です。
3. サンプル・ジョブに示されているように、**SHAREOPTIONS** パラメーターをコーディングします。
4. オプション: **CSD** 初期設定パラメーターを使用する代わりに、ICF カタログで CSD のリカバリー属性を指定できます。  
RLS モードで CSD を使用することにした場合は、ICF カタログでリカバリー属性を指定する必要があります。
  - LOG(NONE) (リカバリー不能データ・セット)
  - LOG(UNDO) (バックアウトのみ)
  - LOG(ALL) (バックアウトおよび順方向リカバリーの両方)LOG(ALL) を指定する場合は、LOGSTREAMID も指定して、順方向リカバリー・ログとして使用する MVS<sup>™</sup> ログ・ストリームの 26 文字の名前を定義する必要があります。ICF カタログでリカバリー属性を指定し、BWO も使用する場合は、LOG(ALL) と BWO(TYPECICS) を指定します。
5. CSD の DDNAME を DFHCSD として指定します。

## 例

```
//DEFINIT JOB accounting information
//DEFCSO EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//AMSDUMP DD SYSOUT=A
//SYSIN DD *
DEFINE CLUSTER -
  (NAME(CICSTS56.CICS.applid.DFHCSO) -
  VOLUMES(valid) -
  KEYS(22 0) - 1
  INDEXED -
  RECORDS(n1 n2) -
  RECORDSIZE(200 2000) - 2
  FREESPACE(10 10) -
  SHAREOPTIONS(2) - 3
  LOG(ALL) - 4
  LOGSTREAMID(CICSTS56.CICS.CSO.FWDRECOV) - 4
  BWO(NO)) - 4
  DATA -
  (NAME(CICSTS56.CICS.applid.DFHCSO.DATA) -
  CONTROLINTERVALSIZE(8192)) -
  INDEX -
  (NAME(CICSTS56.CICS.applid.DFHCSO.INDEX))
/*
//INIT EXEC PGM=DFHCSOUP,REGION=300K
//STEPLIB DD DSN=CICSTS56.CICS.SDFHLOAD,DISP=SHR
//DFHCSO DD DSN=CICSTS56.CICS.applid.DFHCSO,DISP=SHR 5
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN DD *
INITIALIZE
LIST ALL OBJECTS
/*
//
```

図 2. CSO を定義して初期化するサンプル・ジョブ

## 次のタスク

コマンド **LIST ALL OBJECTS** は、現在 CSO に存在する CICS 提供のリソースをリストします。

## さらに大きな CSO の作成

CICS の実行中に CSO がいっぱいになることを防ぐため、1 次スペース・パラメーターと 2 次スペース・パラメーターを指定してデータ・セットを定義しており、2 次エクステント用に十分な DASD スペースを使用できることを確認してください。

CEDA トランザクション (またはオフライン・ユーティリティー) の実行中に CSO がいっぱいになった場合、もっと大きなデータ・セットを定義し、REPRO などの AMS コマンドを使用して CSO のコンテンツをリカバリーします。CSO が動的に割り振られた場合、CSO を閉じてから削除し、もっと大きなデータ・セットとして再定義できます。CSO が動的に割り振られなかった場合、もっと大きなデータ・セットを作成するには、CICS をシャットダウンしなければなりません。

ファイルのコピーに使用できるコマンドの説明については、[z/OS DFSMS アクセス方式サービス・コマンド](#)を参照してください。

## CSO 属性の定義

CSO のファイル処理属性は、複数のシステム初期設定パラメーターに定義されます。

### このタスクについて

以下のシステム初期設定パラメーターに適切な定義を定義します。

### 手順

1. **CSOACC** パラメーターを使用して、許可されるアクセスのタイプを定義します。

2. **CSDBKUP** パラメーターを使用して、CSD が BWO に適格かどうかを定義します。

このパラメーターは、CSDRLS=YES を指定した場合は無視されます。CICS は ICF カタログ内の BWO パラメーターを代わりに使用します。デフォルトでは、ICF カタログ内の LOG パラメーターに UNDO または ALL が指定されている場合、CICS は非 RLS モードの CSD にも ICF カタログ内の BWO パラメーターを使用します。CICS が BWO 属性で常に **CSDBKUP** パラメーターを使用するようにする場合、**NONRLSRECOV** システム初期設定パラメーターを FILEDEF に設定できます。

3. **CSDBUFND** パラメーターを使用して、CSD データのバッファ数进行を定義します。

このパラメーターは、CSDRLS=YES を指定した場合は無視されます。

4. **CSDBUFNI** パラメーターを使用して、CSD 索引のバッファ数进行を定義します。

このパラメーターは、CSDRLS=YES を指定した場合は無視されます。

5. **CSDDISP** パラメーターを使用して、CSD データ・セットのファイル属性指定を定義します。

6. **CSDDSN** パラメーターを使用して、CSD の JCL データ・セット名 (DSNAME) を定義します。

7. **CSDFRLOG** パラメーターを使用して、順方向リカバリー・ジャーナル ID を定義します。

このパラメーターは、CSDRLS=YES を指定した場合、または、ICF カタログ内の **LOG** パラメーターにリカバリー属性が定義されている場合は無視されます (この場合、ICF カタログの LOGSTREAMID が代わりに使用されます)。CICS が LOGSTREAMID 属性で常に **CSDFRLOG** パラメーターを使用するようにする場合、**NONRLSRECOV** システム初期設定パラメーターを FILEDEF に設定できます。

8. **CSDINTEG** パラメーターを使用して、RLS モードでアクセスされる CSD の読み取り保全性のレベルを定義します。

9. **CSDJID** パラメーターを使用して、自動ジャーナリングの ID を定義します。

10. **CSDLSRNO** パラメーターを使用して、VSAM ローカル共用リソース・プールを定義します。

この値は、CSDRLS=YES を指定した場合は無視されます。

11. **CSDRECOV** パラメーターを使用して、CSD がリカバリー可能かどうか定義します。

このパラメーターは、CSDRLS=YES を指定していて、CICS が ICF カタログからの **LOG** パラメーターを代わりに使用する場合は無視されます。**LOG** パラメーターが「未定義」の場合、RLS モードで CSD を開く試みは失敗します。

CSDRLS=NO の場合、このパラメーターは、ICF カタログの **LOG** パラメーターが「未定義」の場合のみ使用されます。デフォルトでは、ICF カタログの **LOG** パラメーターに NONE、UNDO、または ALL が指定されている場合、**LOG** パラメーターにより **CSDRECOV** の値がオーバーライドされます。CICS が LOG 属性で常に **CSDRECOV** パラメーターを使用するようにする場合、**NONRLSRECOV** システム初期設定パラメーターを FILEDEF に設定できます。

12. **CSDRLS** パラメーターを使用して、CSD に RLS モードと非 RLS モードのどちらでアクセスするかを定義します。

13. **CSDSTRNO** パラメーターを使用して、同時要求のストリング数を定義します。

CSDRLS=YES を指定した場合は、この値は無視され、値 1024 が想定されます。

## 次のタスク

これらのパラメーターについては、[105 ページの『CICS システム初期化パラメーターの指定』](#)で詳しく説明しています。

## 同じ CICS 領域からの共用ユーザー・アクセス

CICS 領域内の複数のユーザーが同時に CSD にアクセスできます。

CSD の読み取り/書き込み権限を指定した場合は、CICS 領域内のすべての CEDA ユーザーが読み取り機能と書き込み機能を実行できます。CICS ファイル制御は、**CSDACC** システム初期設定パラメーターで指定されている属性を使用して、1 つの領域内の複数のユーザーの並行アクセスを管理します。

詳しくは、[28 ページの『CICS 領域内の CSD の複数ユーザー \(非 RLS\)』](#)を参照してください。



## 複数の CICS 領域からのユーザー・アクセスの分担

異なる CICS 領域の複数のユーザーが同時に CSD へアクセス可能です。

1. 1つの CICS 領域に、CSD への読み取り/書き込みアクセスを付与します (CSDACC=READWRITE システム初期設定パラメーター)。廃止された古いリリースのリソース属性を問題なく更新できるように、この CICS 領域は最新レベルである必要があります。その他の CICS 領域には、CSD への読み取りアクセスのみを付与します (CSDACC=READONLY システム初期設定パラメーター)。これにより、同じ MVS イメージまたは異なる MVS イメージにある複数の CICS 領域で CSD の整合性が保持されます。
2. 共用する CSD の更新は 1つの領域からのみ行い、それ以外の領域では必要な領域にインストールするためだけに CEDA を使用する場合、更新を行う領域に読み取り/書き込みアクセスを指定し、それ以外の領域には読み取り専用アクセスを指定します。
3. 複数の CICS 領域から CSD を更新する必要がある場合は、CICS トランザクション・ルーティング機能および MRO または ISC を使用すると、読み取り専用 CICS 領域で CSD を更新できます。手順は、以下のとおりです。
  - a. CSD を所有する 1つの領域 (CSD 所有領域) を選択し、その領域でのみ CSD の読み取り/書き込みアクセスを指定します。
  - b. その他の CICS 領域では、CSD を読み取り専用として定義します。
  - c. CSD 所有領域以外のすべての領域で、次のようにします。

- 1) CEDB トランザクションを、(CSD 所有領域で実行される) リモート・トランザクションとして再定義します。
- 2) 定義をインストールし、グループをそれらの領域のグループ・リストに追加します。

これにより、CEDB トランザクションを任意の領域から使用して CSD の内容を変更したり、CEDA を使用して呼び出し側の領域にインストールしたりできます。CSD を所有していない領域で CSD を変更するために CEDA を使用することはできません。

CSD 所有領域で障害が発生した場合は、CSD 所有領域の緊急リスタートが完了する (CSD のバックアウト処理が完了する) まで、CEDB トランザクションを介して CSD を使用することはできません。緊急リスタートの前に、バックアウト処理のターゲットである CSD の GROUP または LIST をインストールしようとする、その GROUP または LIST が内部的に他のユーザーに対してロックされていることが警告されます。CSD 所有領域で緊急リスタートが実行されると、内部ロックはバックアウト処理によって解除されるため、この状況でオフライン VERIFY を実行しないでください。

上記の方法を使用せずに、リカバリー可能リソースとして CSD を定義する場合は、CSD の整合性は保証されません。その場合、CSD は BWO に適していないため、CSDBKUP=DYNAMIC を指定できません。

4. CSD への読み取り/書き込みアクセスを持つ複数の CICS 領域を定義することができますが、この方法は、それらの CICS 領域が同じ MVS イメージで実行され、すべてが最新の CICS レベルである場合にのみ検討してください。
5. 25 ページの図 2 に示すように、複数の CICS 領域に同じ CSD への読み取り/書き込みアクセスを付与し、それらの領域が同じ MVS イメージにある場合、CSD の整合性は VSAM 定義の SHAREOPTIONS(2) オペランドによって保持されます。
6. 複数の CICS 領域に同じ CSD への読み取り/書き込みアクセスを付与し、それらの領域が異なる MVS イメージにある場合、それらの MVS イメージの VSAM は互いを認識しないため、VSAM SHAREOPTIONS(2) オペランドを使用しても CSD の整合性は保持されません。

### CICS 領域および DFHCSDUP からの共用アクセス

読み取り/書き込みモードで DFHCSDUP ユーティリティ・プログラムを使用して CSD を更新する場合、CEDA、CEDB、または CEDC トランザクションを使用している CICS ユーザーがいないことを確認する必要があります。

CSD へのアクセスを制限する可能性のあるその他の要因については、31 ページの『CSD アクセスを制限する他の要因』を参照してください。

CSD へのアクセスを制御するためのシステム初期設定パラメーターについては、25 ページの『CSD 属性の定義』を参照してください。

## CICS 領域内の CSD の複数ユーザー (非 RLS)

CSD の読み取り/書き込み権限を指定した場合は、CICS 領域内のすべての CEDA ユーザーが読み取り機能と書き込み機能を実行できます。CICS ファイル制御は、CSDACC システム 初期設定パラメーターで指定されている属性を使用して、1 つの領域内の複数のユーザーの並行アクセスを管理します。

CICS は、CSD に対する一連の内部ロックを使用して、個々のリソース定義を同時更新から保護します。CICS はこれらのロックをグループ・レベルで適用します。CICS は、グループ内の要素を更新するコマンドを実行する間、内部ロックを使用して、領域内のその他の RDO トランザクションが同じグループを更新できないようにします。更新コマンドの実行が完了すると、CICS はロック・レコードを解除します。この方法により、リストに対する操作も保護されます。

CSD に対する処理可能な同時要求の数は、CSDSTRNO システム 初期設定パラメーターで定義されます。CEDA (または CEDB、CEDC) のユーザーごとに 2 つの文字列が必要であるため、CSDSTRNO 値を計算するには、まず CSD への同時アクセスが必要なユーザー数を見積もり、その数を 2 倍します。

CSDSTRNO 値が小さすぎて、同時要求による CSD に対する瞬間的なニーズを満たせない場合に、CEDA は診断メッセージを発行します。競合が解消された場合、コマンドを再実行しようとする後続の試行は成功します。競合がまだ発生している場合は、CSDSTRNO 値を大きくしてください。

## 単一 MVS イメージ内の複数の CICS 領域による CSD の共用 (非 RLS)

同じ MVS イメージ内の複数の CICS 領域で CSD を共用することができます。

25 ページの図 2 に示すように VSAM 定義に SHAREOPTIONS(2) をコーディングすると、この状態で CSD の保全性を保持できます。特定の領域で参照する CSD の CICS 属性は、その領域のシステム 初期設定パラメーターで定義されます。

以下を定義することを検討する必要があります。

- CSD への読み取り/書き込みアクセス (CSDACC=READWRITE) を 1 つの CICS 領域に付与する。その領域では、CEDA、CEDB、および CEDC のすべての機能を使用できます。
- 他の CICS 領域には CSD への読み取りアクセスのみ (CSDACC=READONLY) を付与する。そのような CICS 領域では、CEDC トランザクション、および CSD への書き込みアクセスを必要としない CEDA と CEDB の機能を使用できます (例えば、INSTALL、EXPAND、VIEW は使用できますが、DEFINE は使用できません)。27 ページの『[複数の CICS 領域からのユーザー・アクセスの分担](#)』で説明している手順を使用して、そのような CICS 領域からの CSD の更新を可能にすることができます。

注：共用 CSD への読み取り専用アクセスを持つ CICS 領域では、読み取りの保全性は保証されません。例えば、完全な読み取り/書き込みアクセスを持つ 1 つの CICS 領域が、新しい定義または変更された定義を使用して共用 CSD を更新した場合に、読み取り専用アクセスを持つ別の CICS 領域がその更新された情報を取得していないことがあります。これは、(読み取り/書き込み領域が更新する前に) 読み取り専用領域が既に保持していた制御間隔 (CI) が、更新された定義をその読み取り専用領域が取得するために必要な CI と同じである場合に生じる可能性があります。この状況では、VSAM は CI を既に保持しているため、データ・セットを再読み取りしません。ただし、CSDLSRNO=NONE を指定し、CSDBUFNI と CSDBUFND を最小値に設定して、この VSAM に関する制限を最小限に抑えることができます。しかし、パフォーマンスは低下します。RLS モードでアクセスされるデータ・セットの読み取り保全性については、32 ページの『[CSD の読み取り保全性の指定](#)』を参照してください。

CSD への読み取り/書き込みアクセスを持つ複数の CICS 領域を定義する場合は、それらのすべての領域が最新のレベルである必要があります。VSAM SHAREOPTIONS(2) 定義によって他の領域は CSD を開くことができないため、読み取り/書き込みアクセスを持つ 1 つの CICS 領域だけが、CEDA、CEDB、または CEDC トランザクションを使用して CSD にアクセスできます。

リカバリー可能リソース (CSDRECOV=ALL) として定義された CSD を使用して CICS を実行する場合は、特別な考慮事項について、33 ページの『[バックアップとリカバリーの計画](#)』で確認してください。

CEMT を使用して CSD のファイル・アクセス属性を変更できます。また、アプリケーション・プログラムで **EXEC CICS SET FILE** コマンドを使用することもできます。ただし、その結果として設定される属性が、CSDACC=READWRITE または CSDACC=READONLY のいずれかによって定義される属性と同じでなければなりません。それらのパラメーターによって許可される CSD に対する操作は次のとおりです。



## READONLY

読み取りと参照。

## READWRITE

追加、削除、更新、読み取り、および参照。

## 多重 MVS 環境内での CSD の共用 (非 RLS)

異なる MVS イメージで実行される複数の CICS 領域間で CSD を共用する場合は、1 つの領域にのみ読み取り/書き込みアクセスを付与する必要があります。

多重 MVS 環境で実行される VSAM は互いを認識しないため、VSAM SHAREOPTIONS(2) を使用しても整合性が保持されません。

この制限のため、異なる MVS イメージで実行されるアクティブな代替 CICS 領域は、他の CICS 領域と CSD を共用してはいけません。ただし、ある種のグローバル・エンキュー (例えば、グローバル・リソースの逐次化 (GRS) など) を使用する場合は共用できます。

これらの多重 MVS の制限は、オフライン・ユーティリティ DFHCSDUP の実行にも適用されます。

## CICS 領域またはバッチ領域 (非 RLS) をまたぐ 1 つの CSD の複数ユーザー

CSD へのアクセスが必要なアクティビティには、4 つのタイプがあります。

CSD が使用される 4 つの状況に必要なアクセスのタイプを [29 ページの表 8](#) に示します。

表 8. CSD アクセス		
	アクティビティのタイプ	Access (アクセス)
1	初期設定を実行する CICS 領域 (コールド・スタートまたは初期始動)	読み取り専用
2	1 つ以上の CEDA、CEDB、または CEDC トランザクションを実行する CICS 領域	読み取り/書き込みまたは読み取り専用 (CSDACC パラメーターで指定)
3	ユーティリティ・プログラム DFHCSDUP を実行するバッチ領域	読み取り/書き込みまたは読み取り専用 (PARM パラメーターに依存)
4	緊急リスタートを実行する CICS 領域で、CSD ファイル・バックアウトが必要な場合	読み取り/書き込み

[29 ページの表 8](#) にリストしたアクティビティが同時に試行される場合は、次の制限事項に注意してください。

1. 同じ CSD を使用する CICS 領域が CEDA、CEDB、または CEDC トランザクションのいずれかを実行している場合、バッチ領域で読み取り/書き込みモードを使用して DFHCSDUP を実行することはできません (CSD にアクセスする CEDx トランザクションが、CSD が読み取り専用として定義された領域で実行されている場合は、例外です)。
2. 使用する CSD が読み取り/書き込みモードの DFHCSDUP ユーティリティ・プログラムによってアクセスされている場合、どの CEDx トランザクションも実行されません (CSD が読み取り専用として定義された領域でトランザクションが実行される場合、この制約事項は適用されません)。
3. RDO トランザクションが、読み取り/書き込みアクセスで定義された CSD が存在する別の CICS 領域で実行されている場合、読み取り/書き込みアクセスで CSD が定義された CICS 領域では、どの CEDx トランザクションも実行されません。

初期始動またはコールド・スタートで開始された CICS 領域は、CSDACC オペランドにかかわらず、初期設定時に読み取り専用アクセスで CSD を開きます。これにより、別の領域のユーザーまたは DFHCSDUP ユーティリティ・プログラムが同時に CSD を更新していても、CICS 領域を初期設定できます。グループ・リストがインストールされた後、CICS は CSD を閉じた状態のままにします。

CSDRECOV=NONE がシステム初期設定パラメーターとしてコーディングされている場合、ウォーム・スタートまたは緊急スタートの CICS 初期設定中に CSD が開かれることはありません。一方、CSDRECOV=ALL

がコーディングされていて、CSD に対するバックアウト処理が保留されている場合は、緊急スタートの CICS 初期設定中に CSD が開かれます。

## 異なる CICS リリース間での CSD の共用

共通のリソース定義を共用するために、CICS の異なるリリース間で CSD を共用することができます。リソース定義を更新する際、CSD を最高のリリース・レベルから更新する必要があります。さらに、廃止されたリソース属性に与える影響も考慮する必要があります。新しいリリースの CICS にとって適切でなくなったリソース属性は廃止されます。

リソース属性が使用されなくなったリリースなど、サービス中の CICS TS リリース間でのリソース定義の変更点については、[Changes to resource definitions](#) を参照してください。

CEDx パネルでも廃止されたリソース属性を見つけることができます。そのような属性は、保護されたフィールドとして CEDx パネルに表示されます。これは、そのリリースでサポートされていないことを示しています。

CEDx ALTER コマンドを使用してリソース定義を更新する場合、定義にある廃止された属性は維持されます。そのため、廃止された属性に影響を与えることなく、現在のリリースを使用してリソース定義を更新できます。

古いリリースの CICS 領域と CSD を共用する場合は、保護されたフィールドから保護を解除して、廃止された属性を ALTER モードで更新できます。これを実行するには、CEDA または CEDB 表示パネルで、互換性ファンクション・キー (COM) である F2 ファンクション・キーを使用します。F2 ファンクション・キーは、保護されたフィールドを、変更可能な保護されていないフィールドに変換します。この保護解除機能を使用し、異なるリリース・レベルの CICS 間で共通リソース定義を共用する場合、CSD の更新は最高のリリース・レベルから実行する必要があります。

互換モードで CEDA および CEDB ALTER コマンドを使用してリソース定義を更新する方法については、[リソース管理トランザクション CEDA コマンド](#) を参照してください。

CSD ユーティリティ・プログラム DFHCSDUP を使用して、廃止された属性を指定するリソースを更新することもできます。EXEC PGM=DFHCSDUP ステートメントの **PARM** パラメーターで、互換性オプション COMPAT または NOCOMPAT を指定する必要があります。デフォルトは NOCOMPAT で、これはプログラムが廃止された属性を更新しないことを意味します。詳しくは、[DFHCSDUP の入力パラメーター](#) を参照してください。

## Db2 を使用する CICS 領域間での CSD の共用

Db2 を使用するさまざまなリリースの CICS 間で CSD を共用する場合は、CICS のリリースごとに該当する Db2 リソース定義を使用しなければなりません。

このような CICS Db2 接続機能を提供する CICS のリリースの使用時には、DFHDB2 という CICS 提供のグループを使用する必要があります。このグループは、CICS 提供の始動リスト DFHLIST に組み込まれており、Db2 で提供されている接続機能とは違うプログラム名を指定しています。

## CICS 提供の互換性グループ

CICS Transaction Server for z/OS, バージョン 5 リリース 6 とそれより前のリリースの CICS との間で CSD を共用する場合は、**GRPLIST** システム初期設定パラメーターで指定するグループ・リストに CICS で必要なすべての標準定義を組み込まなければなりません。

CSD を CICS TS for z/OS, バージョン 5.6 レベルにアップグレードすると、グループ・リストで参照されている IBM グループの一部が削除され、コンテンツがいずれかの互換性グループ DFHCOMPx に移されます。その種のグループを旧リリースの CICS 領域で引き続き使用するには、そのような互換性グループを他のすべての CICS 提供定義の後に追加しなければなりません。

CSD のアップグレードや CICS TS for z/OS, バージョン 5.6 の互換性グループについては、『アップグレード』の『異なる CICS リリース間での CSD の互換性』を参照してください。

## CSD アクセスを制限する他の要因

CEDA、CEDB、または CEDC トランザクションが異常終了した後に CSD が開かれたままである場合も、CSD へのアクセスが制限されることがあります。

書き込みアクセスを使用して CSD が開かれたままの状態では、他のアドレス・スペースが書き込みアクセスで CSD を開くことができません。この状態を解消するには、CEMT を使用して CSD の状況を修正します。

CSD へのアクセスは、それを使用する RDO トランザクションが終了するまで解放されません。そのため、CEDA、CEDB、および CEDC のユーザーは、それらのトランザクションを実行する端末を放置しないようにする必要があります。必ず、できる限りすぐに PF3 を使用してトランザクションを終了してください。そうしないと、他の領域のユーザーが CSD を開くことができません。

時により、グループまたはリストに定義を作成できないことがあります。更新しようとしているグループまたはリストに内部ロック・レコードが存在する場合に、そのような状況になります。この状況が発生しているときに、DFHCSDUP ユーティリティー・プログラム (または CEDA トランザクション) を実行すると、CICS は、グループまたはリストがロックされていることを示すメッセージを発行します。[28 ページの『CICS 領域内の CSD の複数ユーザー \(非 RLS\)』](#)で説明しているように、これは通常、同じ領域内の別のユーザーが同じグループやリストを更新しているときに発生する一時的な状況です。ただし、障害が発生して CEDA トランザクションが正常に完了できなかった場合、CSDRECOV=NONE がコーディングされていると、内部ロックが解除されず、有効なまま残されます (CSDRECOV=ALL がコーディングされている場合、CSD はリカバリー可能であり、ファイル・バックアップが行われるので、ロックは解放されます)。例えば、CEDA トランザクションの実行中にシステム障害が発生すると、この状況が起こる可能性があります。また、CSD がフル状態になった場合にも起こる可能性があります。DFHCSDUP ユーティリティー・プログラムを VERIFY コマンドで実行すると、この状態を解消できます。

ただし、CSDRECOV=ALL をコーディングした場合は、オフライン VERIFY を実行する前に、CSD で保留されているバックアップ処理がないことを確認してください。CSDRECOV=ALL のコーディングが与える影響については、[33 ページの『バックアップとリカバリーの計画』](#)で詳しく説明しています。

## RLS アクセスと非 RLS アクセスでの CSD 管理の違い

RLS モードでアクセスされる CSD は VSAM RLS ロックで保護されますが、この操作は CICS ファイル制御レベルです。CEDA トランザクションや CEDB トランザクションが CSD グループの保全性を管理する方法は変わりません。

CEDx トランザクションは、RLS モードの CSD でも非 RLS モードの CSD でも同じ方法でリソース定義を保護します。このトランザクションは、CSD 上の一連の内部ロックによる同時更新に対して、個別リソース定義を保護します。RDO トランザクションは、これらのロックをグループ・レベルで適用します。RDO トランザクションは、グループ内のエレメントを更新するコマンドを実行している間、内部ロックを使用して、CICS 領域内のその他の RDO トランザクションが同じグループを更新できないようにします。更新コマンドの実行が完了した場合に限り、ロックが解放されます。リストに対する操作も同じ方法で保護されます。しかし、RLS 環境では、これらの内部ロックは、RLS モードで CSD を開いているすべての CICS 領域に影響します。非 RLS の場合、データ・セットを更新用に開いている CICS 領域 (このような領域は 1 つだけです) のみに適用されます。

SMSVSAM サーバーで単一バッファ・プールが使用されるので、取得したデータを非 RLS CSD と共有する際の問題の一部が回避されます。

他の注目すべき点には以下があります。

- CSDACC=READWRITE と CSDRLS=YES を指定して CSD が定義されている場合、複数の CICS 領域で CSD を同時に開くことができます。しかし、ファイル制御により、各 CEDx トランザクションの終了時に、RLS モードで開いている CSD が非 RLS CSD と同じ方法で閉じられます。RLS モードでアクセスされる CSD の場合、CSDACC=READONLY は不要です。
- RLS の場合、CSD に対して処理可能な同時要求の数は常に 1024 です。RLS モードの CSD の場合、CSDSTRNO 値に関する診断メッセージは示されません。
- CICS などのアプリケーションがデータ・セットを RLS モードで開くと、SMSVSAM では VSAM クラスター定義 SHAREOPTIONS パラメーターは無視されます。

- RLS モードでアクセスされる CSD では、非 RLS モードの CSD の場合には発生しない RDO トランザクション障害が発生することがあります。以下に例を示します。
  - RDO トランザクションはレコードの更新中に RLS 排他ロックを保持している場合があり、この場合には別の RDO トランザクションがタイムアウトになります。
  - CSD がリカバリー可能で CICS か MVS に障害が起こると、障害が起こった未完了 RDO トランザクションの更新ロックは保持ロックに変換されます。この場合、RDO トランザクションが VSAM から LOCKED 応答を受け取り、そのために RDO トランザクションに障害が起こることがあります。

## CSD の読み取り保全性の指定

CSD の読み取り保全性を RLS モードで開くように指定できます。

これにより、CEDx INSTALL コマンドが常に確実に最新のバージョンのリソース定義をインストールします。別の CEDx トランザクションによりレポートの更新が行われている場合、CEDA INSTALL コマンドは、インストールを実行しようとしているいずれかの CSD レコードのロックを待機する必要があります。インストールが完了するのは、更新タスクによるレコードの更新が完了し、排他ロックをリリースした後です。

CSDINTEG システム初期設定パラメーターは、整合読み取り保全性と反復可能読み取り保全性の両方をサポートしますが、整合読み取り保全性の方が RDO 操作に必要なすべてのメリットが揃っています。

## CSD のファイル制御属性の指定

CSDxxxxx システム初期設定パラメーターを使用して CSD に関するファイル制御属性を指定できますが、多少の例外もあります。

この場合の例外は以下のとおりです。

### **CSDBKUP**

ICF カタログ内で VSAM **BWO** パラメーターを使用して、CSD に関するオープン時バックアップ・サポートを指定します。

### **CSDBUFND**

無視されます。

### **CSDBUFNI**

無視されます。

### **CSDFRLOG**

ICF カタログ内で VSAM **LOGSTREAMID** パラメーターを使用して、CSD に関する順方向リカバリー・ログ・ストリームを指定します。

### **CSDINTEG**

このシステム初期設定パラメーターを使用して、RDO トランザクション (CEDx) の読み取り保全性を指定します。

### **CSDLSRNO**

無視されます。

### **CSDRECOV**

ICF カタログ内で VSAM **LOG** パラメーターを使用して、CSD に関するリカバリー属性を指定します。

**LOG** が「undefined」の場合は、RLS モードで CSD を開こうとすると失敗します。

### **CSDSTRNO**

RLS の場合、ストリングの数はデフォルトの 1024 になります。

## CSD バッチ・ユーティリティー DFHCSDUP に対する RLS の影響

DFHCSDUP を使用して、RLS モードのリカバリー不能 CSD を、CICS でも RLS モードでこの CSD を更新用に開いている場合に更新できます。

DFHCSDUP を使用して RLS モードの CSD を更新するには、DFHCSDUP JCL 内の CSD に関する DD ステートメント内で RLS=NRI または RLS=CR を指定します。一般的に、DFHCSDUP を非 RLS モードで実行すると、RLS モードの場合よりパフォーマンスが低くなります。



CSD がリカバリー可能と定義されている場合、CICS 領域で CSD が RLS モードで開かれているときには DFHCSDUP を実行できません。これは、DFHCSDUP などの非 CICS ジョブは、リカバリー可能データ・セットが既に RLS モードで開かれているときに、このデータ・セットを非 RLS モードで出力用に開くことができないことが理由です。したがって、DFHCSDUP を実行する場合は、その前に CEMT または EXEC CICS の SET DSNAME(...) QUIESCED コマンドを発行して、CSD を静止しなければなりません。

DFHCSDUP の実行中は、リカバリー可能 CSD はすべての CICS 領域で使用できません。ただし、CSD を静止解除すると、再度 RLS モードで使用できるようになります。DFHCSDUP の実行の終了時に CSD を静止解除するには、CEMT または EXEC CICS の DSNAME(...) UNQUIESCED コマンドを発行します。

リカバリー可能 CSD の場合、RLS を使用するかどうかを計画する際には、主要要素として DFHCSDUP の使用量と CEDx トランザクションの使用量の比較を考慮してください。頻繁に DFHCSDUP を使用して実動 CSD を更新する場合には、CSD を非 RLS モードで使用する方がよいと決定することもできます。他方、DFHCSDUP を頻繁には使用せず、CICS 領域から CSD をオンラインで更新する機能が必要な場合は、RLS を使用します。

## バックアップとリカバリーの計画

CSD に影響を与えるシステム障害に備えるために、一定間隔で CSD のバックアップを取得してください。それにより、何らかの理由で CSD が破損した場合に、最後にバックアップを取得したときの状態に CSD を復元できます。

### このタスクについて

可能な限り最新の CSD バックアップを取得するために、RDO トランザクションまたは DFHCSDUP のいずれかで更新アクティビティを行う場合には、そのアクティビティ 期間より前に CSD のイメージ・コピーを作成してください。

また、RDO 処理が実行されるときには必ず CSD が更新のために開かれるので、その CSD を BWO の対象にすることもできます。BWO の対象として CSD を指定すると、データ・セットが破損した場合に、DFSMSdss を使用して CSD の BWO イメージを復元し、順方向リカバリー・ユーティリティを使用して、破損した時点まで順方向リカバリーを実行することができます。

RLS モードで CSD を開く場合は、リカバリー属性を CSD の ICF カタログ・エントリーに定義する必要があります。この場合、CICS は、ICF カタログから取得した順方向リカバリーのログ・ストリーム名 (LSN) を使用します。

非 RLS モードで CSD を開く場合、リカバリー属性は、CSD の ICF カタログ・エントリーか、CSD システム初期設定パラメーターで定義することができます。順方向リカバリーのログ・ストリーム名 (LSN) は、CSDFRLOG または ICF カタログのいずれかから取得されます。LOG がカタログで定義されている場合は、カタログで指定された順方向リカバリー・ログ・ストリームが使用されます。LOG が定義されていない場合は、CSDFRLOG ジャーナル ID を使用してログ・ストリーム名が決定されます。**NONRLSRECOV** システム初期設定パラメーターが FILEDEF に設定されている場合は、必ず、CSDFRLOG ジャーナル ID を使用してログ・ストリーム名が決定されます。ICF カタログに指定されたリカバリー属性は無視されます。

非 RLS モードで CSD を開く場合は、システム初期設定パラメーター CSDBKUP=DYNAMIC|STATIC を使用して、CSD が BWO の対象かどうかを指定できます。BWO サポートの場合は CSDBKUP=DYNAMIC を指定し、「通常の」静止バックアップの場合は STATIC (デフォルト) を指定します。CSD に対して BWO サポートを指定した場合は、CSD を順方向リカバリー可能として定義する必要があります。BWO の詳細については、[46 ページの『VSAM ファイルのオープン中のバックアップ \(BWO\) の定義』](#)を参照してください。

RLS モードで CSD を開く場合、すべてのリカバリー属性 (バックアップを含む) を ICF カタログに指定する必要があります。BWO(TYPECICS) を使用して BWO バックアップの適格性を指定します。

CSD に対して順方向リカバリーを指定した場合は、CICS によって CSD に加えられた変更 (変更後イメージ) が順方向リカバリー・ログ・ストリームに記録されます。最新のバックアップと、順方向リカバリー・ログ・ストリームの変更後イメージを使用して、CICS VSAM 順方向リカバリー・ユーティリティなどのリカバリー・プログラムを実行することで、加えられたすべての変更をリカバリーできます。順方向リカバリーを実行した後、障害時に実行されていた CEDA トランザクションを再入力する必要があります。これらのトランザクションは順方向リカバリー処理によって効率的にバックアウトされているからです。CSDL 一時データ宛先 (すべての CEDA コマンドのコピーのログ) でこれらの詳細を確認できます。詳細については、[37 ページの『RDO コマンドのロギング』](#)を参照してください。

非 RLS モードで CSD にアクセスする場合は、リカバリーの可能性、順方向リカバリー・ログ・ストリーム名、および BWO 適性を ICF カタログにオプションで定義できます。一方、RLS モードで CSD にアクセスする場合は、これらを必ず ICF カタログに定義する必要があります。

CSDBKUP、CSDRECOV、および CSDFRLOG システム 初期設定パラメーターの相互作用は、各パラメーターの指定によって異なります。34 ページの表 9 と 34 ページの表 10 に、SIT がアセンブルされたとき、および CICS オーバーライドの処理中にこれらのパラメーターが与える影響をそれぞれまとめています。

表 9. SIT アセンブリー時の CSDBKUP および関連パラメーター (CSDRLS=NO)			
CSDRECOV	CSDFRLOG	CSDBKUP	結果
ALL	01 から 99 までの FRLOG	DYNAMIC または STATIC のいずれか	OK
ALL	NO	DYNAMIC または STATIC のいずれか	SIT アセンブリーは失敗し、CSDRECOV=ALL の場合は CSDFRLOG オプションを指定する必要があることを示す MNOTE が生成されます。
BACKOUTONLY または NONE	01 から 99 までの FRLOG	DYNAMIC	SIT アセンブリーは失敗し、CSDBKUP=DYNAMIC には CSDRECOV=ALL が必要で、CSDFRLOG には CSDRECOV=ALL が必要であることを示すアセンブラー MNOTE が生成されます。
BACKOUTONLY または NONE	NO	DYNAMIC	SIT アセンブリーは失敗し、CSDBKUP=DYNAMIC には CSDRECOV=ALL が必要であることを示すアセンブラー MNOTE が生成されます。
BACKOUTONLY または NONE	NO	STATIC	OK
BACKOUTONLY または NONE	01 から 99 までの FRLOG	STATIC	CSDRECOV=ALL でない限り CSDFRLOG は無視されることを示す SIT アセンブリー警告 MNOTE が生成されます。

注:

1. CSDBKUP=DYNAMIC の場合、CSD は BWO の対象になります。
2. RLS モードで CSD を開く場合 (CSDRLS=YES)、バックアップおよびリカバリー属性を ICF カタログに指定する必要があります。
3. 非 RLS モードで CSD を開く場合 (CSDRLS=NO)、バックアップおよびリカバリー属性を ICF カタログにオプションで指定できます。ただし、その場合も、上記の表で定義している整合性が取れた一連のパラメーターが必要です。

表 10. CICS オーバーライドの処理中の CSDBKUP および関連システム初期設定パラメーター (CSDRLS=NO)			
CSDRECOV	CSDFRLOG	CSDBKUP (「注」を参照)	結果
ALL	01 から 99 までの FRLOG	DYNAMIC または STATIC のいずれか	OK

表 10. CICS オーバーライドの処理中の CSDBKUP および関連システム初期設定パラメーター (CSDRLS=NO) (続き)			
CSDRECOV	CSDFRLOG	CSDBKUP (「注」を参照)	結果
ALL	NO	DYNAMIC または STATIC のいずれか	CSDRLS=NO の場合、CSDFRLOG なしでは CSDRECOV=ALL を指定できないことを示すメッセージ DFHPA1944 が発行されます。CICS の初期設定が終了されます。
BACKOUTONLY または NONE	01 から 99 までの FRLOG	DYNAMIC	処理は続行され、CSDBKUP がデフォルトの STATIC に設定されたことを示すメッセージ DFHPA1929、および CSDFRLOG が無視されたことを示す DFHPA1930 が発行されます。
BACKOUTONLY または NONE	NO	DYNAMIC	処理は続行され、CSDBKUP がデフォルトの STATIC に設定されたことを示す DFHPA1929 が発行されます。
BACKOUTONLY または NONE	NO	STATIC	OK
BACKOUTONLY または NONE	01 から 99 までの FRLOG	STATIC	処理は続行され、CSDFRLOG が無視されたことを示すメッセージ DFHPA1930 が発行されます。

**注：**

1. CSDBKUP=DYNAMIC の場合、CSD は BWO の対象になります。
2. RLS モードで CSD を開く場合 (CSDRLS=YES)、バックアップおよびリカバリー属性を ICF カタログに指定する必要があります。
3. 非 RLS モードで CSD を開く場合 (CSDRLS=NO)、バックアップおよびリカバリー属性を ICF カタログにオプションで指定できます。ただし、その場合も、上記の表で定義している整合性が取れた一連のパラメーターが必要です。

実動 CICS 領域の運用を開始する前に、CSD のバックアップとリカバリーの手順を作成してテストしてください。

CSD の更新が CICS の外部で行われた場合、CSD の順方向リカバリーは不可能です。CICS の外部で行われた更新のリカバリーを可能にするには、イメージ・コピーを使用する必要があります。CICS の外部から CSD を更新する場合は、イメージ・コピーが作成されるまで、CEDA を使用した CSD の更新は行わないでください。

## 緊急時再始動中のトランザクション・バックアウト

**CSDRECOV** システム初期設定パラメーターをコーディングして CSD をリカバリー可能リソースとして定義した場合、CSD には、他の CICS リカバリー可能リソースに対する規則と同じ規則が適用されます。

CSDRECOV=ALL (または BACKOUTONLY) をシステム初期設定パラメーターとしてコーディングし、障害が発生した後に緊急時再始動を実行する必要がある場合、CICS は障害発生時に未完了だったすべての不完全な RDO トランザクションをバックアウトします。

## トランザクションの動的バックアウト

CICS は、RDO トランザクションが異常終了すると動的トランザクション・バックアウトを実行します。

CSD 内のトランザクション定義で属性をコーディングして、動的トランザクション・バックアウトが必要かどうかを決めることはできません。CICS ではすべてのトランザクションについてバックアウトの必要があると想定します。(CSD をリカバリー不能として定義すると、バックアウトを避ける効果は得られませんが、推奨されていません。)

## リカバリーに関するその他の考慮事項

指定するリカバリー可能性オプションを決定する際には、多数の要素を考慮しなければなりません。

考慮の対象となる要素は以下のとおりです。

- CEDA コマンドの同期点の基準
- 別の CICS 領域との CSD の共用
- オフライン・ユーティリティー・プログラム DFHCSDUP による CSD へのアクセス

CEDA コマンドの同期点の基準については、[36 ページの『CEDA コマンドの同期点の基準』](#)を参照してください。CICS 領域間での CSD の共用については、[Sharing user access from several CICS regions](#) を参照してください。DFHCSDUP ユーティリティーを使用した CSD へのアクセスについては、[36 ページの『オフライン・ユーティリティー・プログラム DFHCSDUP による CSD へのアクセス』](#)を参照してください。

## CEDA コマンドの同期点の基準

CEDA コマンドは、コマンド・ラインで実行することもできれば、EXPAND パネルや DISPLAY パネルで一連のコマンドとして実行することもできます。

CSD の内容を変更するコマンドは、コマンド・レベルで変更をコミットしたりバックアウトしたりします。この規則の例外は、汎用の ALTER コマンドです。汎用の ALTER コマンドは、リソース・レベルでコミットされたりバックアウトされたりします。

INSTALL コマンドによって既存のリソース定義が置き換えられるのは、そのリソースが使用中でない場合に限られます。インストールするグループ内のいずれかのリソースが使用中になっていると、インストールが失敗します。

INSTALL コマンドによる以下のリソース定義の変更は、リソース・レベルでコミットされ、インストールが失敗した時にバックアウトされません。

- AUTOINSTALL MODEL、FILE、LSRPOOL、MAPSET、PARTITIONSET、PARTNER、PROFILE、PROGRAM、TDQUEUE、TRANSACTION

INSTALL コマンドによる以下のリソース定義の変更は、グループ・レベルでコミットされ、インストールが失敗した時にバックアウトされます。

- CONNECTION、SESSION、TERMINAL、TYPETERM

## オフライン・ユーティリティー・プログラム DFHCSDUP による CSD へのアクセス

オフライン・ユーティリティー・プログラム DFHCSDUP による CSD の変更はリカバリーできません。

このプログラムに用意されているコマンドで以下の操作を実行すると、さまざまな影響があります。障害が発生した CICS 領域の緊急リスタートを実行する前に、そのような影響について検討してください。

1. バックアウト処理のターゲットになっているリストやグループの内容の変更。
2. 内部ロックの解除 (VERIFY など)。



そうした状況は、複数の読み取り/書き込み領域を使用する場合に生じる問題とよく似ています。上記の説明を参照してください。

## RDO コマンドのロギング

RDO コマンドを記録する場合、サンプル・ジョブを使用して、区画外キュー CADL、CAIL、CRDI、CSDL、CSFL、CSKL、CSLB、CSPL、CSRL の定義を作成できます。

### このタスクについて

これらのキューは、以下のように使用されます。

#### CADL

アクティブな CICS 領域にインストールされた z/OS Communications Server のリソースをログに記録します。CICS は、インストールされた z/OS Communications Server リソース、削除された z/OS Communications Server リソース、動的にインストールされ、破棄された z/OS Communications Server リソースをすべてこのログに記録します。このログには、自動インストールされた端末定義、CEDA INSTALL コマンドによって明示的にインストールされた端末定義、およびシステム初期設定中にグループ・リストからインストールされた端末定義が含まれます。

#### CAIL

自動インストール端末モデルの動的インストールおよび削除をログに記録します。

#### CRDI

プログラム、トランザクション、マップ・セット、プロファイル、区画セット、ファイル、LSR プールなどのインストールされたリソース定義をログに記録します。

#### CSDL

CSD に影響を与える RDO コマンドをログに記録します。

#### CSFL

アクティブな CICS 領域にインストールされたファイル・リソースをログに記録します。つまり、インストールまたは削除されたファイル・リソース定義、動的にインストールされ、破棄されたファイル・リソース定義、データ・セットの動的割り振りに関するメッセージ、および CICS データ・テーブルのロードに関するメッセージのすべてが含まれます。

#### CSKL

アクティブな CICS 領域にインストールされたトランザクションおよびプロファイル・リソースをログに記録します。つまり、インストールまたは削除されたトランザクションおよびプロファイル・リソース定義、動的にインストールされ、破棄されたトランザクションおよびプロファイル・リソース定義のすべてが含まれます。

#### CSLB

アクティブな CICS 領域にインストールされた動的 LIBRARY リソースに加えられた変更をログに記録します。つまり、インストールおよび破棄に関する変更、および LIBRARY の有効化、ランキング、またはクリティカル状況への変更のすべてが含まれます。動的 LIBRARY リソースがインストールされていない場合、監査ロギングは発生しません。

#### CSPL

アクティブな CICS 領域にインストールされたプログラム・リソースをログに記録します。つまり、インストールまたは削除されたプログラム・リソース定義、動的にインストールされ、破棄されたプログラム・リソース定義のすべてが含まれます。

#### CSRL

アクティブな CICS 領域にインストールされた一連のパートナー・リソースに加えられた変更をログに記録します。つまり、パートナー・リソースをインストールまたは破棄したすべての操作が含まれます。

これらの RDO コマンド・ログを同じ宛先 (CSSL) にメッセージとして送信する場合、38 ページの図 3 に示す定義を使用できます。必要であれば、これらのログを他の一時データ・キューに送信したり、区画外データ・セットとして定義したりできます。

注：VTAM は現在 z/OS Communications Server になっています。

```
*
DEFINE TDQUEUE (CSSL)          GROUP(DFHDCTG)
  DESCRIPTION(USED FOR MESSAGES)
  TYPE(EXTRA)                  TYPEFILE(OUTPUT)
  RECORDSIZE(132)              BLOCKSIZE(136)
  RECORDFORMAT(VARIABLE)       BLOCKFORMAT(UNBLOCKED)
                                DDNAME(MSGUSR)

*
DEFINE TDQUEUE (CADL)          GROUP(DFHDCTG)
  DESCRIPTION(CEDA VTAM RESOURCE LOGGING)
  TYPE(INDIRECT)              INDIRECTNAME(CSSL)

*
DEFINE TDQUEUE (CAIL)          GROUP(DFHDCTG)
  DESCRIPTION(AITM MESSAGES)
  TYPE(INDIRECT)              INDIRECTNAME(CSSL)

*
DEFINE TDQUEUE (CRDI)          GROUP(DFHDCTG)
  DESCRIPTION(RDO INSTALL LOG)
  TYPE(INDIRECT)              INDIRECTNAME(CSSL)

*
DEFINE TDQUEUE (CSLB)          GROUP(DFHDCTG)
  DESCRIPTION(CICS LD Domain LIBRARY Audit Trail)
  TYPE(INDIRECT)              INDIRECTNAME(CSSL)

*
DEFINE TDQUEUE (CSDL)          GROUP(DFHDCTG)
  DESCRIPTION(CEDA COMMAND LOGGING)
  TYPE(INDIRECT)              INDIRECTNAME(CSSL)

*
DEFINE TDQUEUE (CSFL)          GROUP(DFHDCTG)
  DESCRIPTION(FILE ALLOCATION MESSAGES)
  TYPE(INDIRECT)              INDIRECTNAME(CSSL)

*
DEFINE TDQUEUE (CSKL)          GROUP(DFHDCTG)
  DESCRIPTION(TRANSACTION MANAGER MESSAGES)
  TYPE(INDIRECT)              INDIRECTNAME(CSSL)

*
DEFINE TDQUEUE (CSPL)          GROUP(DFHDCTG)
  DESCRIPTION(PROGRAM MANAGER MESSAGES)
  TYPE(INDIRECT)              INDIRECTNAME(CSSL)

*
DEFINE TDQUEUE (CSRL)          GROUP(DFHDCTG)
  DESCRIPTION(PARTNER RESOURCE MANAGER)
  TYPE(INDIRECT)              INDIRECTNAME(CSSL)
```

図 3. RDO コマンド・ログを CSSL に送信するための定義

## SPI コマンドのロギング

システム・プログラミング・インターフェース・コマンドは、システム管理や監査を支援する目的で監査されます。

### このタスクについて

システム・プログラミング・インターフェース・コマンドの監査によって、CADS 一時データ・キューに送信されるメッセージが生成されます。これらのメッセージは、人間が理解できる形式で書き込まれます。問題判別に役立つように別のキューにリダイレクトすることもできます。メッセージが大量になる場合にも、別のキューにリダイレクトすることができます。[SPI commands that can be audited](#) を参照してください。

CADS キューは、DFHLIST の一部である DFHDCTG グループに定義されています。CADS は、間接キューとして定義されていて、デフォルトでは CSSL キューにリダイレクトされます。定義を [39 ページの図 4](#) に示しています。ただし、特定の目的のために特別に定義した区画外キューに CADS をリダイレクトすることもできます。例えば、[39 ページの図 5](#) に示すように「SPIAUDIT」という名前の区画外キューを定義できます。

```

*
DEFINE TDQUEUE (CSSL)          GROUP(DFHDCTG)
      DESCRIPTION(USED FOR MESSAGES)
      TYPE(EXTRA)              TYPEFILE(OUTPUT)
      RECORDSIZE(132)          BLOCKSIZE(136)
      RECORDFORMAT(VARIABLE)   BLOCKFORMAT(UNBLOCKED)
                                DDNAME(MSGUSR)

*
DEFINE TDQUEUE (CADS)          GROUP(DFHDCTG)
      DESCRIPTION(SPI AUDIT MESSAGES)
      TYPE(INDIRECT)           INDIRECTNAME(CSSL)
*

```

図 4. SPI 監査メッセージを CSSL にリダイレクトするための定義

```

*
DEFINE TDQUEUE (SPIA)          GROUP(DFHDCTG)
      DESCRIPTION(USED FOR MESSAGES)
      TYPE(EXTRA)              TYPEFILE(OUTPUT)
      RECORDSIZE(132)          BLOCKSIZE(136)
      RECORDFORMAT(VARIABLE)   BLOCKFORMAT(UNBLOCKED)
                                DDNAME(SPIAUDIT)

*
DEFINE TDQUEUE (CADS)          GROUP(DFHDCTG)
      DESCRIPTION(SPI AUDIT MESSAGES)
      TYPE(INDIRECT)           INDIRECTNAME(SPIA)
*

```

図 5. SPI 監査メッセージを区画外キューにリダイレクトするための定義

## CICS で CSD を有効にする方法

CICS で CSD を有効にするには、DD ステートメントを CICS 始動ジョブに組み込むか、動的割り振りを使用します。

### このタスクについて

#### 手順

1. CICS 始動ジョブ・ストリームに以下の DD ステートメントを組み込みます。

```
//DFHCSD DD DSN=CICSTS56.CICS.applid.DFHCSD,DISP=SHR
```

通常は、DISP=SHR を組み込むための CSD DD ステートメントが必要です。[\(Sharing user access from several CICS regions\)](#) を参照してください。

CSD の DD ステートメントを CICS 始動ジョブに組み込むと、CICS ジョブ・ステップ開始時に CSD が割り振られ、CICS ジョブ・ステップの間はずっと割り振られたままになります。

2. CSD を動的に割り振る場合は、CSD のデータ・セット名 (DSNAME) と処理方法 (DISP) を以下のいずれかの方法で指定します。

- システム初期設定パラメーター **CSDDSN** と **CSDDISP**
- **CEMT SET FILE** コマンド
- **EXEC CICS SET FILE** コマンド

始動ジョブ・ストリームで DD ステートメントを指定しないでください。CSD の DD ステートメントがあると、動的割り振りの代わりにその DD ステートメントが使用されます。

CICS が OPEN 処理の一部として、完全なデータ・セット名 (DSNAME) を使用して CSD を割り振ります。関連する最後の項目が閉じられると、自動的に CSD の割り振りが解除されます。

## 次のタスク

OPEN 処理の詳細については、82 ページの『[ユーザー・ファイルの定義](#)』を参照してください。SIT でコーディングできる CSD 用のパラメーターについては、105 ページの『[CICS システム 初期化パラメーターの指定](#)』を参照してください。

## RDO トランザクションのインストール

RDO トランザクション (CEDA、CEDB、および CEDC) は、CICS 提供のグループ DFHSPI に定義されています。

このグループは、DFHLIST (CICS グループ・リスト) にも含まれています。必ず、CICS の始動に使用するグループ・リストに、DFHSPI のコピーを含めてください。グループ・リストは、GRPLIST 初期設定パラメーターで指定します。

CEDA、CEDB、および CEDC トランザクションについては、[CEDA - オンライン・リソース定義](#)を参照してください。

## 日本語機能の定義のインストール

日本語機能がある場合は、DFHCSDUP ユーティリティーを実行して、その機能の定義を CSD にインストールします。

DFHCSDUP ユーティリティーを実行する際に、次のオプションを指定します。

```
UPGRADE USING(DFHRDJPN)
```

サインオン用の言語固有マップ・セット (DFHSNLK) を以前使用していた場合、これは DFHRDJPN グループの一部としてサポートされなくなりました。その代わりに、サンプル・ライブラリー SDFHSAMP に用意されている CESL と CESN マップを変更することができます。詳細については、[CESL - サインオン長](#)および [CESN - サインオン](#)を参照してください。

カスタマイズされたマップ・セット DFHSNLK (前のリリースで使用していたもの) が既にある場合は、そのマップ・セットの名前を DFHSNLE に変更すれば、モジュールを再コーディングして再生成する必要はありません。

DFHCSDUP ユーティリティー・プログラムおよび使用可能なコマンドについては、[システム定義ファイル・ユーティリティー・プログラム \(DFHCSDUP\)](#)を参照してください。

## 第 4 章 CICS 領域のセットアップ

CICS 領域をセットアップするには、CICS 領域用のデータ・セットを作成し、JCL を編集して CICS 領域が使用できるサービスを構成し、CICS 領域を開始します。

### データ・セットの定義

CICS システム・データ・セットは、一時ストレージ、一時データ、トランザクション・ダンプ、トレースなど、複数の CICS プロセスで使用されます。これらのデータ・セットの中には、CICS を実行する上でオプションのものと、必須のものがあります。事前形式設定が必要なデータ・セットについては、その目的に使用できるジョブを示します。

### CICS データ・セットのセットアップ

CICS を実行するためには複数のデータ・セットを定義する必要があります。データ・セットの中には、必須のものもあれば、対応する機能を使用する場合にのみ必要になるものもあります。通常は、ユーザー・ファイル、DL/I データベース、および z/OS Communications Server 端末以外の端末のためのデータ・セット定義も指定する必要があります。

#### 手順

1. 必要な CICS データ・セットがどれかを計画します。
  - a) CICS 領域に組み込む CICS 機能およびそのデータ・セット要件について検討します。
  - b) データ・セットの命名規則を定義します。
  - c) データ・セットおよびデータ定義ステートメントを、実行する CICS 領域に定義するために割り振るスペースを計算します。
2. CICS で使用するデータ・セットを定義してカタログします。
3. 必要に応じて、CICS で使用するデータ・セットを初期設定または事前フォーマットします。
4. RACF® などの外部セキュリティ・マネージャーでデータ・セットを保護し、セキュリティ要件に合わせます。
5. 必要なデータ・セットの DD ステートメントを CICS 始動ジョブ・ストリームに定義します。

以下のデータ・セットについては、DD ステートメントを定義する必要はありません。

  - CICS 動的割り振り機能を使用するユーザー・ファイル。
  - CICS ファイル制御によって管理される、CICS 動的割り振り機能を使用する CICS システム・データ・セット。
  - CICS リモート DL/I サポートまたは DBCTL を介してアクセスする DL/I データベース。

ユーザー・ファイル定義について詳しくは、[82 ページの『ユーザー・ファイルの定義』](#)を参照してください。

#### 次のタスク

CICS データ・セットの定義が完了したら、CICS ユーティリティ・プログラムを使用して、そのデータ・セットの後処理を実行できます。これらのユーティリティについては、[CICS ユーティリティ・プログラム](#)で説明します。

#### CICS データ・セットの検討

必須およびオプションのデータ・セットのリストを参照し、CICS 領域に作成するデータ・セットを選択してください。

#### このタスクについて

定義する CICS データ・セットを計画する中で、z/OS データ・セット暗号化がサポートされるデータ・セットにデータ・セット暗号化を使用するかどうかも考慮してください。[103 ページの『データ・セット暗号化の計画』](#)を参照してください。

## 手順

1. 次の CICS データ・セットは、定義する必要があります。

データ・セット	DDNAME	ブロックまたは CI サイズ (バイト)	レコード形式	データ・セット編 成
補助トレース (補助トレース・データ・セットのセットアップと使用を参照)	DFHAUXT DFHBUXT	4096	F	順次
BTS ローカル要求キュー (ローカル要求キュー・データ・セットを参照)	DFHLRQ	1024 および 2560	VB	VSAM KSDS
カタログ (カタログ・データ・セットのセットアップと使用を参照)	DFHGCD DFHLCD	8192 & 2048	VB	VSAM KSDS
CSD (CICS システム定義データ・セットのセットアップを参照)	DFHCSD	8192	VB	VSAM KSDS
ダンプ (ダンプ・データ・セットの定義を参照)	DFHDMPA DFHDMPB	32760 (磁気テープ) または 1 トラック (DASD)	V	順次
メッセージおよびコード (CMAC メッセージ・データ・セットの定義を参照)	DFHCMACD	—	V	VSAM KSDS
一時記憶域 (一時記憶域データ・セットのセットアップを参照)	DFHTEMP	一時データ・セットの制御間隔サイズを参照	該当せず	VSAM ESDS
一時データ区画外 (54 ページの『区画外データ・セットの定義』を参照)	リソース定義の DDNAME オプションから	リソース定義の BLOCKSIZE オプションから	リソース定義の RECORD FORMAT オプションから	順次
一時データ区画内 (52 ページの『区画内データ・セットの定義』を参照)	DFHINTRA	53 ページの『区画内データ・セットのサイズ』を参照してください。	該当せず	VSAM ESDS

一時記憶域データ・セットおよび一時データ区画内データ・セットは、制御間隔 (CI) 処理を使用するため、レコード・フォーマットは該当しません。

2. 同等の機能を使用する場合は、次の CICS データ・セットを定義する必要があります。

データ・セット	DDNAME	詳細情報
CDBM グループ・コマンド	DFHDBFK	<a href="#">CDBM GROUP コマンド・データ・セットの定義</a>
CICSplex SM WUI サーバー HTML テンプレート・データ・セット	DFHHTML	<a href="#">WUI カスタマイズ可能ビューおよびメニュー・ヘルプ・データ・セットの指定</a>
デバッグ・プロファイルのデータ・セット	DFHDPFMB、 DFHDPFMP、 DFHDPFMX	<a href="#">デバッグ・プロファイルのデータ・セットのセットアップ</a>
Link3270 ブリッジ	DFHBRNSF	<a href="#">DFHBRNSF の定義</a>



データ・セット	DDNAME	詳細情報
サンプル・アプリケーション・プログラムのサンプル・ファイル FILEA	FILEA	<a href="#">The FILEA sample application programs</a>
WS-AT ディレクトリー・データ・セット	DFHPIDIR	<a href="#">WS-AT データ・セットの定義</a>

### データ・セット命名規則の定義

CICS データ・セット用に選択するデータ・セット名には、MVS の制約以外の制約事項はありません。この情報では、CICSTS56.CICS が高位修飾子として使用され、DD 名が最下位修飾子として使用されています。複数の CICS 領域を実行する場合は、第 2 レベルの修飾子として CICS アプリケーション ID を使用できます。

### このタスクについて

データ・セットがアクティブ CICS 領域と代替 CICS 領域の間で共用されている場合には、総称アプリケーション ID を使用してください。ただし、データ・セットがアクティブ CICS 領域または代替 CICS 領域のいずれかで固有の場合には、特定のアプリケーション ID を使用してください。

### 手順

- 4 文字の名前の場合は、CTGI という 4 文字のシンボルに基づいた CTGI 命名規則を使用します。各部分の意味は次のとおりです。
  - C は、CICSplex 全体を示します。
  - T は、領域のタイプを示します。
  - G は、領域のグループを示します。
  - I は、1 つのグループ内の領域の繰り返しを示します。
- CICS アプリケーション ID に関して、8 文字の名前では、最初の 4 文字 (特に実動領域の場合) に CICS という文字を使用します。  
例えば、CICSHTH1 がアプリケーション ID の場合、CSD のデータ・セット名は次のようになります。

```
DFHCSD DD DSN=CICSTS56.CICS.CICSHTH1.DFHCSD,DISP=SHR
```

### 複数のエクステントと複数のボリュームを使用したデータ・セットの定義

一時記憶データ・セットまたは一時データ宛先データ・セットは、単一ボリュームに定義された単一エクステントとして定義できます。そのデータ・セットは、すべてのデータを保持できる十分な大きさにする必要があります。一方、平均的なデータ・セット・サイズよりも大幅に大きいサイズが必要な例外的なケースに対応するために、複数のエクステントと複数のボリュームを使用してデータ・セットを定義することができます。

例えば、次のように定義できます。

- 1 つのボリュームに複数のエクステント
- 複数のボリュームのそれぞれに 1 つのエクステント
- 複数のボリュームに複数のエクステント

複数のエクステントを定義した場合、CICS は 1 次エクステントがいっぱいになった場合にのみ、追加のエクステントを使用します。1 次エクステントに平均的な要求を満たすのに十分な大きさを確保し、オーバーフローしたときのために、それよりも小さい 2 次エクステントを割り当てることができます。この方法により、必要になるまでスペースを節約できます。追加のエクステントがいっぱいになると、VSAM は次のエクステントを作成します。VSAM は、必要に応じて、最大 123 個まで追加のエクステントを作成し続けます。複数のボリュームを使用しても、この制限には影響しません。

同じボリュームで追加のエクステントを割り振る場合は、次のように RECORDS パラメーターに 2 次エクステント・オペランドをコーディングします。

```
RECORDS(primary,secondary)
```

複数ボリュームで単一エクステントを使用する場合は、次のようにコーディングします。

```
RECORDS(primary) -  
VOLUMES(volume1,volume2,volume3,.....)
```

複数のボリュームで複数のエクステントを割り振る場合は、次のように、1 次と 2 次の両方の RECORDS オペランドを複数の VOLUMES オペランドと組み合わせます。

```
RECORDS(primary,secondary) -  
VOLUMES(volume1,volume2,volume3,.....)
```

特定のボリュームがパフォーマンスのボトルネックになっている場合は、複数のボリュームで 1 つのエクステントを使用することを試してください。

VSAM の追加エクステント数の制限に達する前に、ボリュームの空き領域が使い果たされる可能性がある場合は、複数のボリュームで複数のエクステントを使用します。この状況が発生した場合、VSAM はリストにある次のボリュームに追加のエクステントを作成し続けます。

### CICS データ・セットの作成

CICS 提供のジョブを使用して、CICS データ・セットを作成します。

#### このタスクについて

CICS インストールの一環として DFHISTAR ジョブを実行すると、これらのジョブは実際の環境に合わせて調整され、DFHISTAR ジョブの **LIB** パラメーターで指定したライブラリーに格納されます。デフォルトは CICSTS56.XDFHINST です。

コピーしたいジョブを選択して DFHISTAR ジョブを再実行すれば、これらのジョブのコピーを複数生成することができます。これらのジョブの新しいコピーを生成するには、DFHISTAR ジョブを編集して、**DSINFO** および **SELECT** パラメーターに新しい値を指定します。**SELECT** パラメーターで名前が指定されたジョブだけが再生成されます。

#### 手順

1. DFHCOMDS ジョブを実行して CICS 領域定義データ・セット DFHCSD、および SYSIN データ・セットを作成します。
2. DFHDEFDS ジョブを実行して 1 つの CICS 領域でのみ使用するデータ・セットを作成します。  
このジョブの各コピーを別々に実行し、それぞれの CICS 領域用にデータ・セットを作成する必要があります。このジョブは、以下のデータ・セットを作成します。

表 11. DFHDEFDS によって作成されるデータ・セット	
データ・セット	説明
DFHAUXT	非 VSAM 補助トレース (A) データ・セット
DFHBRNSF	ブリッジ
DFHBUXT	非 VSAM 補助トレース (B) データ・セット
DFHDMPA	非 VSAM ダンプ (A) データ・セット
DFHDMPB	非 VSAM ダンプ (B) データ・セット
DFHDPFMB	デバッグ・プロファイルの基本データ・セット
DFHDPFMP	デバッグ・プロファイルのパス・データ・セット
DFHDPFMX	デバッグ・プロファイルのパス・データ・セット
DFHGCD	CICS グローバル・カタログ
DFHHTML	HTML テンプレート・データ・セット
DFHINTRA	区画内一時データ・セット



表 11. DFHDEFDS によって作成されるデータ・セット (続き)	
データ・セット	説明
DFHLCD	CICS ローカル・カタログ
DFHLRQ	BTS ローカル要求キュー
DFHPIDIR	WS-AT ディレクトリー・データ・セット
DFHTEMP	一時記憶データ・セット
FILEA	サンプル・プログラム・ファイル

- DFHCMACI ジョブを実行して CICS メッセージ・データ・セット DFHCMACD を作成し、それを CICSTS56.CICS.SDFHMSGs ターゲット・ライブラリー内の CICS 提供のファイル DFHCMACD のデータによってロードします。

### MVS システム・データ・セットのサイズの見積もり

CICS は、独自のシステム・データ・セットの他に、一部の MVS データ・セットも使用します。

### このタスクについて

それらのデータ・セットは、以下のとおりです。

データ・セット	以下によって所有または使用される	その他のコメント
SDUMP データ・セット	MVS SDUMP マクロ	MVS SDUMP マクロによるシステム・ダンプのために CICS によって使用されます。
SMF データ・セット	システム 管理機能	モニター・レコードおよび統計レコードのために CICS モニター・ドメインおよび統計ドメインによって使用されます。
GTF データ・セット	汎用トレース機能	CICS トレース・エントリーのために CICS トレース・ドメインによって使用されます。

### 手順

- CICS が生成するデータの増加量を考慮に入れつつ、これらのシステム・データ・セットのサイズを再計算してください。  
例えば、SDUMP データ・セットの場合、3380 デバイスの 25 シリンダー以上またはそれと同等のボリュームが必要になります。SDUMP データ・セットのサイズの計算については、[z/OS MVS 初期設定およびチューニングガイド](#)を参照してください。
- SDUMP データ・セットが不要な SDUMP によっていっぱいにならないようにするには、SDUMP を抑制します (79 ページの『ASRx 異常終了より先行するシステム・ダンプの抑止』を参照)。  
SDUMP は、DFHAP0001 メッセージの後で、ASRA、ASRB、ASRD 異常終了よりも先に発生します。
- CICS 間隔統計を頻繁に収集する場合、または 1 回の間隔で収集される統計量が多い場合は、SMF データ・セットのサイズの見積もり時に統計量を考慮に入れる必要があります。  
CICS はレコードを最大 32756 バイトの SMF に書き込むことができるため、SMF は複数の SMF データ・セットにまたがるレコードを書き込むことがあります。より効率的に DASD を使用するために、CICS で使用する SMF データ・セットを、16384 バイト (16 KB) または 8192 バイト (8 KB) のいずれかの制御間隔サイズで作成することを検討する必要があります。その他の制御間隔サイズを使用する場合は、DASD の効率的な使用と SMF データ・セットの I/O パフォーマンスのトレードオフについて、また、不十分な SMF バッファが原因でデータ損失が生じる可能性について検討する必要があります。
- CICS モニター・データを収集する場合は、モニター・クラスがアクティブなときに書き込まれるデータ量のサイズを見積もる必要があります。

SMF に関するバックグラウンド情報、およびその他の SMF データ・セットに関する考慮事項については、[z/OS MVS システム管理機能 \(SMF\)](#)を参照してください。

5. GTF トレースをオンにして CICS を実行する場合、GTF データ・セットの CICS トレース・エントリーを考慮に入れてください。

GTF のバックグラウンド情報については、[z/OS MVS 診断: ツールと保守援助プログラム](#)を参照してください。

### VSAM ファイルのオープン中のバックアップ (BWO) の定義

CICS では、DFSMSdss および DFSMSHsm から提供されるオープン中のバックアップ (BWO) 機能をサポートします。このサポートでは、CICS が VSAM データ・セットを更新している最中に、一部の VSAM データ・セット・タイプを DFSMSdss でバックアップできます。

それと同時に、CICS は、これらのデータ・セットに対する変更の順方向リカバリー・イメージを順方向リカバリー・ジャーナルに記録します。後から、DSMSHsm を使用してデータ・セットのバックアップを復元でき、CICS VSAM Recovery などの順方向リカバリー・ユーティリティを使用してログを適用することによって、整合点まで戻すことができます。

### 始める前に

BWO は、CICS ファイル制御からアクセスされるデータ・セットのみに使用できます。例えば、CICS システム定義 (CSD) データ・セットなどです。頻繁に更新される KSDS データ・セットに適用される制約事項の対象として (詳細については『[48 ページの『BWO に関する制約事項』](#)を参照)、BWO に適している VSAM ユーザー・データ・セットを判別する必要があります。

この機能を使用する VSAM データ・セットは、SMS 管理対象 DASD 上にある必要があり、ICF カタログ構造を持っている必要があります。VSAM ESDS、RRDS (固定と可変の両方)、および KSDS データ・セットのみがサポートされます。

BWO には、いくつかのストレージ管理機能が必要です。[47 ページの『BWO に必要なストレージ管理機能』](#)を参照してください。

### このタスクについて

DFSMS 1.3 の場合は、以下の 2 つの方法で BWO を定義できます。

### 手順

- パラメーター **BWO=TYPECICS** を指定してクラスターを定義します。  
この値は、クラスターが CICS 領域で BWO に適していることを表します。ファイル・リソース定義は、競合する場合でも無視されます。  
RLS モードでオープンされるデータ・セットが含まれたクラスターの場合は、そのクラスター定義に BWO が指定されている必要があります。
- BWO パラメーターが定義されていない場合は、デフォルトで UNDEFINED に設定されます。この場合は、CICS がファイル・リソース定義を参照します。  
CICS は、ファイルが RDO を使用して定義されているときに、BWO に適しているものとしてデータ・セットを定義します。VSAM ファイルに BACKUPTYPE=DYNAMIC が指定されている場合は、データ・セットのオープン時に、ファイルが BWO に適しているものとして定義されます。  
DFSMSdss が、BACKUPTYPE=STATIC とともに指定されたデータ・セットをバックアップする場合は、そのデータ・セットに対する更新のために現在オープンしているすべての CICS ファイルをクローズしてからでないと、バックアップを開始できません。

### タスクの結果

CICS は、VSAM 基本クラスター・データ・セットが BWO に適しているという事実をその基本クラスターのブロックに記録します。ウォーム・リスタート時、および緊急リスタート時これは、VSAM 基本クラスターに対してすべてのファイルがクローズされたとき、CICS のウォーム・リスタート時、および緊急リスタート時に参照されます。CICS のコールド・スタート時または初期始動時には参照されません。

## 次のタスク

適切なプロシーチャーを BWO および順方向リカバリーに適した場所に配置する必要があります。これらのプロシーチャーには、以下が組み込まれている必要があります。

- データ・セットを整合点まで戻すための、BWO バックアップの復元、および順方向リカバリー・ユーティリティの実行。復元するためには、リカバリー・プロセス中のファイルへのユーザー・アクセスを禁止する必要があります。
- CICS への割り振り中に損傷を受けた可能性のあるデータ・セットの復元および順方向リカバリー。この操作を実行するためには、CICS 緊急リスタートによって部分的にコミットされた作業単位をバックアウトしなければならない場合があります。

### BWO に必要なストレージ管理機能

オープン時バックアップ (BWO) 機能には、いくつかのストレージ管理機能が必要です。

BWO のストレージ管理機能は、次のとおりです。

**ストレージ管理サブシステム (SMS)。MVS/DFP バージョン 3 リリース 2 以降 (製品番号 5665-XA3) の一部です。**

SMS は、CICS がストレージ管理サブシステムを使用してデータの配置を決定する DASD ストレージ管理手法です。また、自動データ・マネージャーが、データ・バックアップ、移動、スペース、およびセキュリティを処理します。

これは、DFSMS と呼ばれることもあり、MVS/DFP やデータ機能プロダクト・ファミリーの他の個別の製品の機能を補完します。SMS について詳しくは、以下の資料を参照してください。

- [z/OS DFSMSdftp Storage Administration](#) では、ストレージ管理機能アプリケーションについて説明されています。
- [z/OS DFSMS 紹介](#) では、MVS/DFP の概要とその要件が記載され、SMS 管理ストレージの概念について説明されています。

**重要：**DFHSM を使用して VSAM データ・セットを管理する場合は、どの程度の期間が経過した後に、CICS VSAM データ・セットを 1 次ストレージまたは補助ストレージにマイグレーションするかについて注意深く検討する必要があります。

マイグレーションしたデータ・セットを CICS に再呼び出ししなければならない場合、1 次ストレージからの再呼び出しには数分、補助ストレージからの再呼び出しにはそれ以上の時間がかかる可能性があります。再呼び出しの実行中、ユーザーはロックアウトされ、データ・セットの再呼び出しが完了するまで、そのデータ・セットに対する他のオープンおよびクローズは実行できません。

マイグレーションされたデータ・セットの再呼び出しが必要になると、CICS はメッセージ DFHFC0989 をシステム・コンソールに発行し、再呼び出しが実行されることをユーザーに通知します。また、1 次ストレージと補助ストレージのどちらから実行されるかも示します。

### データ機能階層記憶プログラム (DFHSM)。製品番号 5665-329

DFHSM は、ボリュームとデータ・セットを管理する IBM ライセンス・プログラムです。DFHSM の詳細については、[z/OS DFSMS の Knowledge Center](#) を参照してください。

### データ機能データ・セット・サービス (DFDSS)。製品番号 5665-327

DFDSS は、データ・セットとボリュームのコピー、移動、ダンプ、および復元に使用する IBM ライセンス・プログラムです。DFDSS の詳細については、[z/OS DFSMSdss ストレージ管理](#) を参照してください。

### アクティビティ・キーポイント処理の無効化の影響

システム初期設定パラメーター **AKPFREQ=0** を指定して、CICS 領域でアクティビティ・キーポイント処理を無効にした場合、非 RLS アクティビティに対する BWO サポートに重大な影響があります。

アクティビティ・キーポイント処理を無効にすると、タイアップ・レコード (TUR) が順方向リカバリー・ログに書き込まれず、データ・セット・リカバリー・ポイントが更新されません。このため、更新のためにデータ・セットが最初に開かれた時点から BWO バックアップの順方向リカバリーを実行する必要があります。この場合、順方向リカバリーが行われるように、その時点からすべての順方向リカバリー・ログを保持することが必要になります。長さの変更を伴う多数の挿入やレコードがある場合、多くの順方向リカ

バリーが必要になることがあります。ただし、レコードの更新のみ行われ、長さが変更されない場合、CI の分割はありません。

TUR およびリカバリー・ポイントについて詳しくは、[BWO および並行コピー](#)を参照してください。

### **BWO に関する制約事項**

以下の制約事項は、VSAM KSDS タイプのデータ・セットだけに当てはまります。

BWO の進行中に VSAM の制御インターバル分割または制御域分割が発生すると、バックアップの信頼性が失われ、DFHSM と DFDSS でバックアップが破棄されます。そのような分割が発生すると、データ・セットの一部が重複したり、バックアップにまったく反映されなかったりする可能性があります。DFDSS ではコピー処理が順次実行されるからです。MVS/DFP 3.2 では、分割が発生したことを ICF カタログで確認できます。バックアップの終了時に、DFHSM と DFDSS は ICF カタログをチェックし、分割が発生していた場合や分割が今も進行中の場合は、バックアップを破棄します。そのため、更新頻度の高い VSAM KSDS データ・セットは、BWO には適さない可能性があります。あるいは、アクティビティー量の少ない時間帯 (夜間など) に限って使用できる場合もあります。DFHSM と DFDSS がデータ・セットのバックアップの作成に要する時間よりも、制御インターバル分割または制御域分割の標準的な発生間隔のほうが長ければ、BWO に適した KSDS データ・セットであると言えます。

### **VSAM データ・セットの別名として基本へのパスを使用する**

CICS VSAM データ・セットの場合、基本 DS 名 (データ・セット名) の別名 DS 名を指定する手段として、VSAM パスを使用できます。これは、基本クラスターへの直接のパスでなければならず、基本クラスターの代替索引へのパスであってはなりません。

### **このタスクについて**

CICS TS は、KSDS データ・セット DFHCSD、DFHGCD、および DFHLCD の PATH 別名、および ESDS データ・セット DFHINTRA および DFHTEMP の PATH 別名をサポートします。

### **例**

基本クラスター CICSTS.release.applid.DFHCSD に関連付けられた CICSTS.applid.DFHCSD の DS 名を使用してパスを定義できます。パスの DS 名を JCL で使用できます。この方法では、新しいリリースの CICS TS にマイグレーションするときに、JCL を変更する必要がありません。

## **一時記憶域データ・セットのセットアップ**

各 CICS 領域は、64 ビット・ストレージ (2 GB 境界より上) または 31 ビット・ストレージ (16 MB 境界より上) をメインの一時記憶域として CICS 領域で使用します。非索引付け VSAM データ・セットに、CICS 領域のための一部の補助一時記憶域を定義できます。CICS 領域のアプリケーションは、z/OS カップリング・ファシリティーにある共用一時記憶域プールも使用できます。

### **このタスクについて**

さまざまな一時記憶域の場所と各場所の一時記憶域キューで利用可能な機能の概要については、「パフォーマンスの改善」の「[CICS 一時記憶域: 概要](#)」を参照してください。

メインの一時記憶域に比べ、補助一時記憶域は、長い期間にわたってデータを保管するアプリケーション、または少ない頻度でデータにアクセスするアプリケーションに一般に使用されます。補助ストレージの一時記憶域キューはリカバリー可能として定義することもできます。これにより、CICS の実行が終了してから次の実行までの間、データを保持することができます。主ストレージの一時記憶域キューは、リカバリー可能として定義できません。

カップリング・ファシリティーの共用一時記憶域プールを使用するには、一時記憶域サーバーが必要です (通常は、シスプレックスの各 z/OS イメージ内に 1 台)。ただし、それらのプールに CICS 領域内のストレージは使用されません。共用一時記憶域プールの一時記憶域キューへのアクセスは、リモートの CICS 領域 (キューが所有する領域) によって保持されている一時記憶域キューへのアクセスより短時間で済みます。共用一時記憶域プールでは、トランザクション間の親和性が生じません。詳しくは、[168 ページの『一時記憶域データ共用のための一時記憶域プールの定義』](#)を参照してください。



## 補助一時記憶域データ・セットの定義

ここで説明するサンプル・ジョブを使用して、補助一時記憶域用の VSAM データ・セットを単一のエクステント・データ・セットとして単一ボリューム上に定義するか、CICS 提供のジョブ DFHDEFDS を使用できます。DFHDEFDS は、CICS 領域用のデータ・セットの 1 つとして DFHTEMP データ・セットを作成します。

## このタスクについて

一時記憶域データ・セットに追加の関連を定義することはできません。例えば、PATH は定義しないでください。PATH を定義すると、CICS の始動が失敗します。拡張アドレッシング機能を使用して、SMS データ・クラスから DFHTEMP データ・セットを割り振らないでください。このような割り振りは、CICS ではサポートされていません。

## 手順

1. データ・セット用に複数のエクステントおよび複数のボリュームを追加することを検討します。  
サンプル・ジョブでは、単一ボリュームに定義された単一エクステント・データ・セットが生成されますが、補助一時記憶域を過度に使用するとチャンネルとアームの競合が発生する場合があります。  
DASD スペースをより効率的に使用するために、通常アクティビティー用に十分なサイズの 1 次エクステント、および例外的な状況 (アクティビティーでの予期しないピークなど) 用の 2 次エクステントを使用して、DFHTEMP データ・セットを定義できます。  
さらに多くのエクステントまたはボリュームを定義する方法については、43 ページの『複数のエクステントと複数のボリュームを使用したデータ・セットの定義』を参照してください。
2. オプション: データ・セットを暗号化する場合は、102 ページの『データ・セットの暗号化』を参照してください。
3. **CONTROLINTERVALSIZE** より 7 バイト小さい **RECORDSIZE** 値を指定します。  
一時記憶域のスペース量は、次の 2 つの値で指定する必要があります。
  - a. 制御インターバル・サイズ。スペースの計算方法については、50 ページの『補助一時記憶域の制御インターバル・サイズ』を参照してください。
  - b. データ・セット内の制御インターバルの数。制御インターバルの適切な数の設定方法については、51 ページの『補助一時記憶域の制御間隔数』を参照してください。
4. DFHTEMP データ・セットのデータ定義ステートメントを始動ジョブ・ストリームに追加します。  
一時記憶域データ・セットは、アクティブな CICS 領域により所有され、パッシブに共用されるデータ・セットですが、アクティブな CICS 領域と代替 CICS 領域の両方に割り振られます。このデータ・セットは、代替 CICS 領域ではテークオーバー前に開かれませんが、ジョブ・ステップ開始時に割り振られます。そのため、DD ステートメントに DISP=SHR を指定して、代替 CICS 領域を開始できるようにする必要があります。

```
//DFHTEMP DD DSN=CICSTS56.CICS.applid.DFHTEMP,DISP=SHR
```

## 例

```
//DEFTS JOB accounting info,name
//AUXTEMP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
DEFINE CLUSTER(NAME(CICSTS56.CICS.CNTL.CICSqualifier.DFHTEMP)-
RECORDSIZE(4089,4089) -
RECORDS(200) -
NONINDEXED -
CONTROLINTERVALSIZE(4096) -
SHAREOPTIONS(2 3) -
VOLUMES(valid)) -
DATA(NAME(CICSTS56.CICS.CNTL.CICSqualifier.DFHTEMP.DATA) -
UNIQUE)
/*
```

図 6. 補助一時記憶域データ・セットを定義するジョブの例

## 次のタスク

**TS** システム 初期設定パラメーターを使用して、補助一時記憶域用の VSAM バッファおよびストリングの適切な数を指定します。CICS は、各 VSAM バッファを使用して、DFHTEMP からの制御インターバルを CICS ストレージでできるようにし、バッファと DFHTEMP 間の各 VSAM 入出力要求に VSAM ストリングを使用します。通常、3 つのバッファと 3 つのストリングのデフォルト設定で十分です。パフォーマンスの考慮事項については、[補助一時記憶域: モニターおよび調整](#)を参照してください。

### 補助一時記憶域の制御インターバル・サイズ

補助一時記憶域データ・セットの制御インターバル・サイズは、VSAM CLUSTER 定義の

**CONTROLINTERVALSIZE** パラメーターで指定します。制御インターバルには 1 つ以上の一時記憶域レコードが入るので、制御インターバル・サイズを選択する時には、一時記憶域レコード・サイズについてよく考える必要があります。

1 つの一時記憶域レコードは、一時記憶域キュー内の番号付きの単一項目であり、CICS またはアプリケーションによって書き込まれます。1 つの一時記憶域レコードには以下のスペースが必要です。

- 一時記憶域ヘッダーのために 36 バイト。
- 一時記憶域レコード (一時記憶域キュー内の項目) のデータ長。3270 サポート付きの BMS を使用している場合は、レコードのデータ長が 3270 のバッファ・サイズ以上になります。代替画面サイズ機能のある 3270 端末の場合は、データ長がその 2 つのサイズのいずれか大きいほうになります。大画面デバイスで使用するデータ長に合わせて十分なスペースを確保してください。

16 KB (16 384 バイト) 以下の制御インターバル・サイズの計算では、1 つの一時記憶域レコードに割り振るバイトの総数 (ヘッダー用の 36 バイトを含む) を 64 の倍数に丸めます。それより大きい制御インターバル・サイズの場合は、128 の倍数に丸めます。

CICS は、制御インターバル・サイズを超えた一時記憶域レコードでも処理できますが、その場合はパフォーマンスが落ちる可能性があります。通常発生する最大の一時記憶域レコードの少なくとも 1 つのインスタンス、および制御インターバルの VSAM 制御情報を十分に収容できる制御インターバル・サイズを選択してください。

通常、1 つの制御インターバルに複数の一時記憶域レコードが入ります。レコードのサイズは、それぞれ異なります。制御インターバル・サイズは、転送の効率に影響を与えます。一時記憶域へのアクセスがランダムな場合は、小さめのサイズのほうがパフォーマンスが高くなりますが、アプリケーションが一時記憶域内の項目を順番に使用する場合は、大きめのサイズのほうがパフォーマンスが高くなります。一般に、キューが大きければ大きいほど、読み取りに対する書き込みの比率が高ければ高いほど、項目を順番に使用する傾向が強くなります。

以下のルールに基づいて、正確な制御インターバル・サイズを選択してください。

- 制御インターバル・サイズの最大値は 32 KB です。
- 制御インターバル・サイズが 16 KB (16 384 バイト) 以下の場合は、一時記憶域レコードのためのスペースのほかに、64 バイトの VSAM 制御情報のためのスペースを組み込む必要があります。
- 制御インターバル・サイズが 16 KB (16 384 バイト) を超える場合は、一時記憶域レコードのためのスペースのほかに、128 バイトの VSAM 制御情報のためのスペースを組み込む必要があります。
- 制御インターバル・サイズが 8 KB 未満の場合は、512 バイトの倍数でなければなりません。
- 制御インターバル・サイズが 8 KB 以上の場合は、2 KB の倍数でなければなりません。

### 例

BMS によって 24 x 80 文字の画面を一時記憶域に書き込む場合は、書き込みデータで 1920 バイトが使用されます。CICS 一時記憶域ヘッダーのために 36 バイトが必要なので、合計が 1956 バイトになります。この値を 64 の倍数に切り上げると、1984 バイトです。最後に、VSAM 制御情報のための 64 バイトを加算すると、1 つのレコードを収容するための制御インターバル・サイズは 2048 バイトになります。サイズが異なる可能性のあるレコードを複数収容するために、2048 バイトよりも大きな制御インターバル・サイズを選択できます。

### 補助一時記憶域の制御間隔数

VSAM は RECORDS オペランドと RECORDSIZE オペランドを使用して、データ・セット用に十分なスペースを割り振り、指定されたサイズのレコード数を保持します。

RECORDSIZE パラメーターの 2 つのオペランド (レコード・サイズの平均値と最大値) に同じ値をコーディングしなければならず、この値は CONTROLINTERVALSIZE より 7 バイト小さくなければなりません。こうすると、一時記憶管理で利用できる制御間隔数と一致する VSAM レコード数を指定できます。つまり、間接的に一時記憶域データ・セット内の制御間隔数を指定することになります。(RECORDS パラメーターと RECORDSIZE パラメーターは、CICS 一時記憶域インターフェースで表示される一時記憶域レコード数に対応しないことに注意してください。)

割り振られる制御間隔数は、一時記憶域に関するユーザー要件やシステム要件に応じて異なり、最大で 65 535 まで割り振ることができます。

## 一時データ用のデータ・セットのセットアップ

一時データ・キュー用のデータ・セットは、区画内にするとも区画外にするともできます。一時データ区画内データ・セットは、CICS 領域内のメッセージやデータのキューイングに使用される VSAM 入力順データ・セット (ESDS) になります。一時データ区画外データ・セットは順次ファイルになります。通常これはディスクまたは磁気テープ上に存在します。このキューを使用して、CICS 領域外のデータを送受信できます。

### このタスクについて

メッセージまたはその他のデータは、CEDA トランザクションを使用して区画内または区画外として定義したシンボリック・キューにアドレス指定されます。キューを間接宛先として使用して、他のキューにメッセージまたはデータを転送することもできます。CICS が生成するシステム・メッセージは、通常、一時データ・キュー (区画内または区画外のいずれか) に送信されます。

### 手順

1. CICS が CICS 領域で使用するすべてのキューを定義します。  
キューを省略しても CICS 障害は発生しませんが、必要なキューに CICS がデータを書き込めないと、CICS 領域に関する重要な情報が失われます。
  - a) グループ DFHDCTG のサンプル定義を使用して、必要なキューを定義します。  
[TDQUEUE リソース](#) に要約しているので参照してください。
  - b) DFHDCTG に加えた変更のバックアップ・コピーを取ります。  
保守を適用すると、DFHDCTG グループが更新され、変更内容が上書きされる可能性があります。バックアップを取っておくことで、この問題を回避します。
2. ジョブ制御ステートメントを使用して、区画内データ・セットを定義します。  
この手順の実行方法について詳しくは、[52 ページの『区画内データ・セットの定義』](#)を参照してください。
3. ジョブ制御ステートメントを使用して、区画外データ・セットを定義します。  
この手順の実行方法について詳しくは、[54 ページの『区画外データ・セットの定義』](#)を参照してください。
4. CICS システム・メッセージが発行されたらローカル・プリンターで印刷するようにするには、一時データを端末に書き込むサンプル・プログラム DFH\$TDWT を使用します。  
このサンプル・プログラムは CICS に付属しており、CICSTS56.CICS.SDFHLOAD にあります。また、アセンブラー・ソースは CICSTS56.CICS.SDFHSAMP にあります。

## 区画内データ・セットの定義

ここで説明するサンプル・ジョブを使用して、一時データ区画内データ・セットを定義するか、CICS 提供のジョブ DFHDEFDS を使用できます。DFHDEFDS は、CICS 領域用のデータ・セットの 1 つとして DFHINTRA データ・セットを作成します。

## このタスクについて

区画内データ・セットは、区画内キューのデータをすべて保持できるよう十分な大きさをなければなりません。サンプル・ジョブを使用して、一時データ区画内データ・セットを定義する場合は、以下のステップを実行します。

## 手順

1. 1 つのボリューム上の単一のエクステント・データ・セットが適切かどうか判断します。  
1 つのボリュームに 1 つのエクステントを定義する場合、特殊なケースに対応するために必要な平均的な要件よりかなり大きなデータ・セットが必要になることがあります。複数のエクステントまたは複数のボリューム、またはその両方をデータ・セットに定義することができます。詳しくは、[53 ページの『複数のエクステントおよび複数のボリュームの使用』](#)を参照してください。
2. オプション: データ・セットを暗号化する場合は、[102 ページの『データ・セットの暗号化』](#)を参照してください。
3. CICS がその独自の目的のために必要とする 32 バイトに加えて、最長のデータ・レコードを保持するために十分な大きさの **CONTROLINTERVALSIZE** パラメーターを指定します。制御インターバル・サイズ of 最大値は 32 KB です。  
スペースは、制御インターバル (CI) 単位でキューに割り振られます。最初の CI は、CICS による使用のために予約済みです。残りの CI は、データの保持に使用できます。データ・レコードは、VSAM 標準に従って CI に保管されます。
- 4.トラック単位やシリンダー単位ではなく、レコード単位でスペースを割り振る場合、RECORDSIZE 値を指定する必要があります。  
この値は、**CONTROLINTERVALSIZE** より 7 バイト小さくする必要があります。
5. 区画内データ・セットのデータ定義ステートメントを CICS 始動ジョブ・ストリームに追加します。  
区画内データ・セットの DD 名は DFHINTRA です。また、DSN オペランドは VSAM 入力順データ・セットの名前でなければなりません。例えば、次のように指定できます。

```
//DFHINTRA DD DSN=CICSTS56.CICS.applid.DFHINTRA,DISP={OLD|SHR}
```

## 例

```
//DEFDS JOB accounting info,name,MSGCLASS=A
//TDINTRA EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
        DEFINE CLUSTER -
            ( NAME(CICSTS56.CICS.applid.DFHINTRA) -
              RECORDSIZE(1529,1529) -
              RECORDS(100) -
              NONINDEXED -
              CONTROLINTERVALSIZE(1536) -
              VOL(volid))
            DATA -
              ( NAME(CICSTS56.CICS.applid.DATA.DFHINTRA))
/*
//
```

図 7. 一時データ区画内データ・セットを定義するサンプル・ジョブ

## 次のタスク

**TD** システム初期設定パラメーターを使用して、一時データ区画内データ・セット用の VSAM バッファおよびストリングの適切な数を指定します。CICS は、データ・セットからの制御インターバルを CICS スト



レージで利用できるようにするためにバッファを使用し、バッファとデータ・セット間の VSAM 入出力要求にストリングを使用します。通常、3 つのバッファと 3 つのストリングのデフォルト設定で十分です。

#### 区画内データ・セットを開けない場合の影響

GRPLIST のインストール中にアクティブになる可能性のある区画内キュー定義の有無にかかわらず、CICS の初期設定中には DFHINTRA データ・セットが開かれます。

CICS の初期始動またはコールド・スタート時に DFHINTRA を開けなかった場合、そのことを通知するメッセージが発行され、CICS を続行するかキャンセルするかを尋ねられます。

前回の始動時に DFHINTRA が正常に開かれたのに、その次のウォーム・スタートまたは緊急リスタート時に開けなかった場合、CICS は終了されます。

CICS の初期設定に DFHINTRA データ・セットを使用していない場合に、その CICS の実行で区画内データの宛先をインストールしようとするすると失敗し、該当するエラー・メッセージが発行されます。

#### 複数のエクステントおよび複数のボリュームの使用

1 つのデータ・セットを定義する場合は例外のケースに対応するために平均的なニーズよりもはるかに大きくする必要があります。複数のエクステントや複数のボリュームを定義することができます。

52 ページの図 7 のジョブ制御ステートメントは、単一ボリュームに定義された単一のエクステント・データ・セットに対するものです。そのデータ・セットは、すべてのデータを保持できる十分な大きさにする必要があります。これらの定義方法について詳しくは、43 ページの『[複数のエクステントと複数のボリュームを使用したデータ・セットの定義](#)』を参照してください。

#### 区画内データ・セットのサイズ

すべての使用可能な制御間隔数が現時点でキューに割り振られている場合、さらに EXEC CICS WRITEQ TD 要求を行うと、NOSPACE 応答を受け取ります。READQ TD 要求または DELETEQ TD 要求で制御間隔を解放する必要があります。

区画内データ・セットでは、2 つ以上の制御間隔が保持されている必要があります。QZERO 条件が返されるまでの間に、論理的にリカバリー可能なキューが読み取られて、要求がコミットされる場合、CICS ではそのキューで最後に使用された CI が (最後のレコードの末尾と CI の末尾の間のスペースがなくなるまで) 保持されます。そのキューで最後に使用された CI が保持されるので、後続のそのキューへの書き込み要求は、CI 内の残りのスペース内に収まる場合はこのスペースに収容できます。そのため、QZERO 条件の後に初めてそのキューへの書き込み要求が行われる際に常に新しい CI を獲得する必要がなくなるので、後続の書き込み要求に関するパフォーマンス上の利点があります。しかし、CI がキューに割り振られたままになるので、データ・セットの使用量が増え、CICS により NOSPACE 条件が返される可能性があります。

#### 区画内データ・セットの制約事項

一時データ・キューとして使用される区画内データ・セットを、1 つの CICS 領域だけに関連付けなければなりません。

CICS は、一時データ・キューとして使用される区画内データ・セットに書き込まれるレコードの相対バイト・アドレス (RBA) を保管します。データ・セットに対する VSAM エクスポート操作やインポート操作の間、この RBA を保存するように注意しなければなりません。

CICS の開始時に誤った区画内データ・セット、つまり別の CICS 領域からのデータが含まれるデータ・セットがあると、データが破損したり失われたりすることがあります。

VSAM エクスポート・サービスやインポート・サービスを使用し、データ・セットを圧縮して使用可能なスペースを増やしたり、制御間隔のサイズを増やしたりすると、データが破損したり失われたりすることがあります。

区画内データ・セットに、拡張アドレッシング入力順データ・セット (ESDS) 形式を使用しないでください。

## 区画外データ・セットの定義

各区画外データ・セットは入力専用または出力専用として定義できますが、入出力の両方用に定義することはできません。

### このタスクについて

CICS メッセージのキューとして使用する一時データ区画外データ・セットの定義は、132 バイトのレコード長、および V または VB のレコード・フォーマットで行います。入力区画外項目に FREE=CLOSE パラメーターを使用する場合、CICS セッションで一度しかこのパラメーターを使用できないことに注意してください。キューが CLOSE されて再度 OPEN された後にそのキューの読み取りを試行すると、IOERR 条件が発生します。

区画外データ・セットに VBS を指定することを決定した場合は、JCL の定義に BFTEK=A を指定する必要があることに注意してください。可変長のスパン・レコードが拡張論理レコード・インターフェース (XLRI) を使用せずに処理され、論理レコード・インターフェース (LRI) のレコード域が提供されている場合 (BFTEK=A がデータ制御クロックに指定されている)、QSAM は最大レコード・サイズ (最大 32756 バイト) を収めるために十分な大きさの領域を提供します。このようなデータ・セット定義に BFTEK=A を指定しないと、ランタイム・エラー (アクセス方式から返される異常終了 002 など) が発生する可能性があります。

データ・セットを暗号化の場合は、[102 ページの『データ・セットの暗号化』](#)を参照してください。

### 例

```
//LOGUSR DD DSN=CICSTS56.CICS.applid.LOGUSR,DISP=(NEW,KEEP),  
//        DCB=(DSORG=PS,RECFM=V,BLKSIZE=136),  
//        VOL=SER=volid,UNIT=3380,SPACE=(CYL,2)  
//MSGUSR DD SYSOUT=A
```

図 8. 一時データ区画外データ・セットを定義するための JCL の例

## DFHCXRF データ・セット

ユーザーが定義できる区画外データ・セットの他に、CICS が動的に作成する特殊な区画外キューがあります。このキューは、CXRF という ID を持ち、初期設定プロセスの初期に CICS によって作成されます。

この区画外データ・セットの DD 名は DFHCXRF です。CXRF と DFHCXRF は定義することができません。DFHCXRF を DSCNAME としてコーディングしたり、CXRF を宛先 ID としてコーディングしたりすると、エラー・メッセージが出されます。CSD 内に CXRF の定義を作成しても、CICS はその定義をインストールしません。この CICS 項目はハードコーディングされており、削除も置換もできないためです。

CICS が完全に初期設定される前に CICS 定義の一時データ・キューへの書き込みが試行されると、メッセージが CXRF に書き込まれます。

初期スタートまたはコールド・スタート時に、GRPLIST の一部としてまだインストールされていないキューへのレコードの書き込み要求が受信されると、レコードは CXRF に書き込まれます。

ウォーム・キーポイントが取得された後に、区画内キューへの書き込みが試行されると、レコードは CXRF に書き込まれます。

## アクティブな CICS 領域

一部の CICS コンポーネントで一時データ・キューへの書き込みが必要になる場合があるため、CICS は CICS 初期設定時に CXRF キューを使用します。

初期設定時にキューが使用不可の場合、それらのキューへの書き込み要求は CXRF に転送されます。CXRF キュー定義がインストールされる前の CICS コンポーネントからの一時データへの書き込み要求は、QIDERR 条件で失敗します。

通常シャットダウン中にウォーム・キーポイントが取得された後の、区画内一時データ・キューへの書き込み要求は、CXRF にルーティングされます。

特殊な CXRF キューを利用する場合は、DFHCXRF の DD ステートメントを含める必要があります。DD ステートメントを省略すると、CXRF に転送された一時データの書き込み要求は、NOTOPEN 条件で失敗します。

### DFHCXRF DD ステートメント

DFHCXRF データ・セットを MVS に定義することができます。その場合、他の一時データ区画外データ・セットと同じ方法で、ディスク (またはテープ) 上の SYSOUT データ・セットまたは順次データ・セットに定義します。例えば、次の例に示すいずれかの DD ステートメントを CICS 領域の始動ジョブ・ストリームで使用できます。

```
//DFHCXRF DD SYSOUT=*
```

または

```
//DFHCXRF DD DSN=CICSTS56.CICS.applid.DFHCXRF,DISP=(NEW,KEEP),  
//                      DCB=(DSORG=PS,RECFM=V,BLKSIZE=136),  
//                      VOL=SER=valid,UNIT=3380,SPACE=(TRK,5)
```

図 9. DFHCXRF の DD ステートメントの例

## Setting up CICS log streams

CICS ログ・ストリームを作成して、MVS システム・ロガーを使用して、ジャーナリングおよびロギング情報を記録します。

### このタスクについて

CICS ログ・マネージャーのサポート対象は以下のとおりです。

- CICS システム・ログ。これは、動的トランザクション・バックアウトにも使用されます。
- ユーザー・ジャーナル、順方向リカバリー・ログ、および自動ジャーナル。これらは汎用ログです。

### CICS システム・ログの定義

各 CICS 領域には、それぞれのシステム・ログが必要です。システム・ログは、2 つの MVS システム・ロガー・ログ・ストリーム、すなわち、1 次および 2 次のログ・ストリームとして実装されますが、この 2 のログ・ストリームは単一の論理ログ・ストリームを共に形成します。

### このタスクについて

システム・ログはリカバリーの目的 (例えば、動的トランザクション・バックアウト、または緊急時再始動時) で使用され、他の目的のために使用されることはありません。

CICS は、初期設定時にそのシステム・ログに自動的に接続されます (システム・ログを DUMMY タイプとして定義するジャーナル・モデル定義を指定する場合は除く)。

作業単位の障害および CICS の障害が発生した場合にデータ保全性を維持するためには、システム・ログを定義する必要があります。CICS では、以下を実行するためにシステム・ログが必要になります。

- 障害が発生した作業単位によって変更されたリカバリー可能リソースのバックアウト
- CICS がリモート・パートナーとの会話状態データをリカバリーする必要がある場合のコールド・スタート。
- CICS が領域をシャットダウン前の状態に復元する必要がある場合のウォーム・リスタート。
- CICS が領域をシャットダウン前の状態に復元する必要がある場合や、シャットダウン時に未完了の作業単位によって変更されたリカバリー可能リソースのバックアウトを実行するためにトランザクションをリカバリーする必要がある場合の緊急時再始動。

CICS システム・ログ・ストリームの定義方法については、[カップリング・ファシリティ・ログ・ストリーム](#)および [DASD 専用ログ・ストリーム](#)を参照してください。

## CICS システム・ログ・ストリームの計画

CICS システム・ログ (2 つの物理ログ・ストリームから成る) は領域に固有なので、他の CICS 領域では使用しないでください。デフォルトのログ・ストリーム名 `region_userid.applid.DFHLOG` と `region_userid.applid.DFHSHUNT` は、名前が確実に固有になるように設計されています。

### JOURNALMODEL を使用したシステム・ログの定義

CICS 領域のユーザー ID が実行ごとに変わる場合 (例えば、1 つのテスト CICS 領域を複数のアプリケーション開発者が共用する場合) は、システム・ログにジャーナル・モデルを使用することができます。

ユーザーごとに別の高位修飾子を使用してログ・ストリームを作成すると無駄が多くなる場合があります。どのプログラマーが CICS 領域を開始するかにかかわらず、同じシステム・ログ・ストリームを使用するようにすると、ログ・ストリームの数を最小限に抑えることができます。次の例では、STREAMNAME に記号を指定し、特定の JOURNALNAME を使用して、1 次ログ・ストリームの明示的なモデルにしています。

```
DEFINE GROUP(TEST) DESC('System logs for test CICS regions')
  JOURNALMODEL(DFHLOG) JOURNALNAME(DFHLOG) TYPE(MVS)
  STREAMNAME(TESTCICS.&APPLID..&JNAME.)
```

JOURNALMODEL リソース定義を定義して DFHLOG および DFHSHUNT のログ・ストリーム名を定義する場合は、生成されるログ・ストリーム名が固有になるようにしてください。同じアプリケーション ID を使用する CICS 領域が複数存在する場合は、固有性を確保するためにログ・ストリーム名に他の修飾子を使用する必要があります。

システム・ログに JOURNALMODEL リソース定義を使用する場合は、CICS の初期始動前に、それらのリソース定義を定義して、適切なグループ・リストに (CSD ユーティリティー・プログラム DFHCSDUP を使用して) 追加する必要があります。

システム・ログには TYPE(SMF) を指定できません。

DFHLOG には TYPE(DUMMY) を指定できますが、CICS 領域を常に初期始動して、トランザクション・バックアウトを必要とするリカバリー可能リソースが存在しない場合に限られます。JOURNALMODEL 定義で TYPE(DUMMY) を指定した場合、CICS はコールド・スタートもウォーム・スタートも緊急リスタートも実行できません。

テスト領域や開発領域などでシステム・ログを使用したくない場合、次の例に示すように、タイプ DUMMY を使用して DFHLOG に対して JOURNALMODEL を定義します。

```
DEFINE JOURNALMODEL(DFHLOG) GROUP(CICSLOGS)
  JOURNALNAME(DFHLOG)
  TYPE(DUMMY)
```

システム・ログを使用せずに CICS 領域を開始するには、上記のような JOURNALMODEL 定義を始動グループ・リストに含める必要があります。DFHCSDUP バッチ・ユーティリティー・プログラムを使用して、必要な JOURNALMODEL を定義し、グループをグループ・リストに追加します。

DFHSHUNT には TYPE(DUMMY) を指定できます。ただし、これはシステム・ログを管理する CICS の機能を無効にするため、推奨されません。

### AKPFREQ パラメーターの効果

CICS 領域ごとに定義されている活動キーポイント頻度 (AKPFREQ) を確認してください。その値が大きいと、システム・ログでカップリング・ファシリティのスペースがそれだけ多く必要になりますが、AKPFREQ の値をあまりにも小さくして、活動キーポイントの間隔よりもトランザクションのほうが長く続くようなことにならないようにしてください。

## CICS 汎用ログの定義

汎用ログ・ストリームのジャーナルは、ユーザー・ジャーナル、順方向リカバリー・ログ、および自動ジャーナルから構成されます。

### ユーザー・ジャーナルと自動ジャーナルで使用するログ・ストリームの計画

一般ログは、MVS ログ・ストリーム名により一般ログとして識別され、末尾が DFHLOG や DFHSHUNT ではないのでシステム・ログ・ストリームと区別されます。



## JOURNALMODEL を使用した一般ログの定義

複製されたアプリケーション所有領域 (AOR) の複数のコピーを実行する場合は、ログ・データが共通するため、すべての AOR からのデータを同じログ・ストリームにマージすることが望ましい場合があります。

次の JOURNALMODEL リソース定義は、同じジャーナル ID の複数の CICS ジャーナルを 1 つの共用ログ・ストリームにマップします。

```
DEFINE GROUP(MERGE) DESC('Merge journals across cloned CICS regions')
  JOURNALMODEL(JRNLS) JOURNALNAME(DFHJ*) TYPE(MVS)
  STREAMNAME(&USERID..SHARED.&JNAME.)
```

この例では、領域ごとに固有のログ・ストリームが必要になるデフォルトの CICS アプリケーション ID の代わりに、リテラル SHARED が使用されています。

CICS 領域のユーザー ID が実行ごとに変わる場合に、JOURNALMODEL を使用してジャーナルをログ・ストリームにマップすることもできます。例えば、CICS テスト領域を複数の開発者グループで共用している場合などが当てはまります。ユーザーごとに別の高位修飾子を使用してログ・ストリームを作成すると無駄が多いので、どの開発者が CICS 領域を開始するかにかかわらず、同じログ・ストリームを使用したほうが良い場合があります。例えば、次の汎用 JOURNALMODEL 定義では、明示的な定義によって定義されていないすべてのジャーナルを同じログ・ストリームにマップしています。

```
DEFINE GROUP (TEST) DESC('Journals for test CICS regions')
  JOURNALMODEL(JRNLS) JOURNALNAME(*) TYPE(MVS)
  STREAMNAME(TESTRCICS.&APPLID..&JNAME.)
```

さまざまなジャーナル名を使用して CICS 領域が書き込んだデータを、単一のログ・ストリームにマージすることもできます。

```
DEFINE GROUP (TEST) DESC('Merging journals 10 to 19')
  JOURNALMODEL(J10T019) JOURNALNAME(DFHJ1*) TYPE(MVS)
  STREAMNAME(&USERID..MERGED.JNLS)
DEFINE GROUP (TEST) DESC('Merging journalnames JNLxxxxx')
  JOURNALMODEL(JNLXXXXX) JOURNALNAME(JNL*) TYPE(MVS)
  STREAMNAME(&USERID..MERGED.JNLS)
```

ディスパッチャーの待機では、ストリーム名の末尾の修飾子が CICS リソース名として使用されます。そのため、これが一目瞭然の分かりやすい名前であれば、モニタリング情報や CICS トレース・エントリーを解釈する際に役立ちます。

## 順方向リカバリー・ログで使用するログ・ストリームの計画

CICS は RLS データ・セットと非 RLS データ・セットに関するロギングを実行します。

複数のデータ・セット間で順方向リカバリー・ログ・ストリームを共用できます。つまり、順方向リカバリー可能データ・セットごとにログ・ストリームを定義する必要はありません。トランザクションのパフォーマンスのバランス、高速リカバリー、多数のログ・ストリームの管理に関する作業のみ決定します。

RLS モードで開いているデータ・セットの場合、MVS ロガーはさまざまな CICS システムからの順方向リカバリーログ・レコードをすべて共用順方向リカバリー・ログにマージします。

注意事項は、以下のとおりです。

- 1 つのトランザクションで使用されるすべてのデータ・セットは、同じログ・ストリームを使用する必要があります (同期点で書き込まれるログ・ストリーム数を減らすため)。
- 初めは、以前の CICS リリースで使用していた順方向リカバリー・ログ ID を使用することをお勧めします。
- 以下のようなデータ・セット間で順方向リカバリー・ログ・ストリームを共用します。
  - 同様のセキュリティ要件を持っている
  - 同様のバックアップ頻度である
  - 同時に全体を復元する必要性が出てくる可能性が高い
- ログ・ストリーム名は、データ・セットに関連している必要があります。例えば、PAYROLL.data\_sets は、PAYROLL.FWDRECOV.PAYLOG という名前の順方向リカバリー・ログにマップできます。

- ディスパッチャーの待機では、ストリーム名の末尾の修飾子が CICS リソース名として使用されます。そのため、これが一目瞭然の分かりやすい名前であれば、モニタリング情報や CICS トレース・エントリーを解釈する際に役立ちます。
- 更新頻度の高いデータ・セットと更新頻度の低いデータ・セットを混合しないようにします。混合すると、更新頻度の低いデータ・セットのリカバリー中に、不必要なログ・データが大量に読み取られます。
- すべての更新頻度の高いデータ・セットを単一のログ・ストリームに書き込まないようにして、ストリームのスループットの容量を超えないようにします。
- 単一ログ・ストリームに対して大量のデータ・セットを定義する場合、ログ・ストリームがデータ・フローに追いつかなくなると、頻繁に **structure-full** イベントが発生することがあります。
- 冗長データをログ・ストリームから定期的に削除する必要があります。OS/390® リリース 2 以前では、ユーザーが削除を担当します。システム・ロガー・インベントリー・エントリーがログ・ストリームあたり 168 個のデータ・セットの制限を超える前に、冗長データを削除しなければなりません。OS/390 リリース 3 以降では、**AUTODELETE(YES) RETPD(dddd)** を使用して定義すると、冗長データが一般ログ・ストリームから自動的に削除されるように指定できます。**AUTODELETE** パラメーターと **RETPD MVS** パラメーターについては、[VSAM RLS サポートに必要な定義](#)を参照してください。

通常、順方向リカバリー・ログでは、古いデータの削除は、データのバックアップ頻度に関係があります。例えば、4 世代のバックアップを保持しているとすると、冗長バックアップ世代を削除する場合、関連する冗長順方向リカバリー・ログ・レコードも同様に削除する必要があります。これらは、順方向リカバリーでは不要になっているため、冗長バックアップよりも古いレコードです。

IBM CICS VSAM Recovery については、[CICS VSAM Recovery for z/OS](#) を参照してください。

### ログのログ用に使用するログ・ストリームの計画 (DFHLGLOG)

ログのログは、CICS VSAM Recovery などの順方向リカバリー・プログラムに情報を提供するために、CICS によって書き込まれます。

ログのログは、一種のユーザー・ジャーナルであり、順方向リカバリー・ログに書き込まれた関係レコードのコピーを含んでいます。つまり、CICS がどのリカバリー可能 VSAM データ・セットを使用したか、いつ使用したか、どのログ・ストリームに順方向リカバリー・ログ・レコードが書き込まれたかという情報の要約を提供します。

ログのログを使用できる順方向リカバリー製品を所有している場合、ユーザーは、リカバリー可能データ・セットを共用するすべての CICS 領域が、ログのログ用の同じログ・ストリームに書き込むようにする必要があります。

```
DEFINE GROUP(JRNL) DESC('Merge log of logs')
  JOURNALMODEL(DFHLGLOG) JOURNALNAME(DFHLGLOG) TYPE(MVS)
  STREAMNAME(&USERID..CICSVR.DFHLGLOG)
```

注：この定義は、DFHLIST 内のグループ DFHLGMOD に用意されています。

ログのログを使用できる順方向リカバリー製品を所有していない場合、次のようにダミー・ログ・ストリームを使用できます。

```
DEFINE GROUP(JRNL) DESC('Dummy log of logs')
  JOURNALMODEL(DFHLGLOG) JOURNALNAME(DFHLGLOG) TYPE(DUMMY)
```

順方向リカバリーに加え、ログ・ストリーム・エラーの記録にもログのログは使用されます。ログ・ストリームで障害が発生した場合、DFHLGLOG が TYPE(DUMMY) と定義されていない限り、CICS はエラーおよび診断の情報を使用してログのログの更新を試みます。

テスト CICS 領域と実動 CICS 領域でログのログを共用しないでください。誤って使用されると、復元中に実動データ・セットの内容が失われる可能性があります。

### ジャーナルの命名

ジャーナルには、以下の命名規則があります。

### このタスクについて

## システム・ログ

DFHLOG と DFHSHUNT は、CICS システム・ログのジャーナル名です。

59 ページの表 12 に示されているように、初期設定中に DFHLOG と DFHSHUNT のジャーナル・テーブル・エントリーが CICS により自動的に作成されます。

表 12. CICS 1 次システム・ログのジャーナル名エントリー	
ジャーナル・テーブル・エントリー - CICS システム・ログ	システム初期設定時に作成されるとき状況
Name: DFHLOG	1 次ログの場合常に DFHLOG
Status: Enabled	ジャーナル・エントリーの作成時に設定
Type: MVS	デフォルト。ただし、JOURNALMODEL 定義で DUMMY として定義可能 (DUMMY = 出力なし)。
LSN: log_stream_name	デフォルトでは log_stream_name は &reguserid..&applid..DFHLOG に解決されるが、これは JOURNALMODEL 定義でユーザー定義にすることができる

## 順方向リカバリー・ログ

VSAM カタログでリカバリー属性が指定されていない非 RLS データ・セットでは、順方向リカバリー・ログ名は、DFHJnn という形式になります。ここで、nn には 1 から 99 までの範囲の数値が入ります。

FILE リソース定義に順方向リカバリー・ログの名前を順方向リカバリー・ログ ID (FWDRECOVLOG) で定義します。

ユーザー・アプリケーションでは、同じログ・ストリーム名にマップされたユーザー・ジャーナル名を介して順方向リカバリー・ログを使用できます。この場合、ユーザー・レコードは、順方向リカバリー・ログにマージされます。この例については、59 ページの表 13 を参照してください。

表 13. 非 RLS モードの順方向リカバリー・ログのためのジャーナル名の項目の例	
ジャーナル・テーブルの項目 - 順方向リカバリー・ログ	ファイル・オープン処理中に作成される項目
Name: DFHJ01	FWDRECOVLOG ID から導出された名前。例: FWDRECOVLOG(01) = DFHJ01 (FWDRECOVLOG(nn) = DFHJnn)
Status: Enabled	ジャーナル・エントリーの作成時に設定
Type: MVS	デフォルト。ただし、JOURNALMODEL 定義で DUMMY として定義可能 (DUMMY = 出力なし)。
LSN: log_stream_name	デフォルトでは、log_stream_name は &reguserid..&applid..DFHJ01 に解決される。ただし、JOURNALMODEL 定義でユーザーが定義可能

注: RLS ファイルの順方向リカバリー・ログでは、ジャーナル・テーブル項目はありません。リカバリー属性と LSN は VSAM カタログから直接取得され、LSN は CICS ファイル制御によって直接参照されます。そのため、ジャーナル名を介して間接的にマッピングする必要がありません。

VSAM カタログの非 RLS ファイルに対してリカバリー属性と LSN を指定することもできます。

## ユーザー・ジャーナル

CICS ユーザー・ジャーナルは、ジャーナル名 (または DFHJnn 名のケースでは番号) で識別され、MVS ログ・ストリームにマップされます。

ユーザー・ジャーナルの名前は、データ・セット修飾子名の規則に準拠する 1 から 8 文字で指定します。DFHJ の文字から始まり、2 文字の数字が続くユーザー・ジャーナル名は除き、DFH で始まる名前を使用するのは避けてください。形式が DFHJnn のユーザー・ジャーナル名は、以前の CICS リリースとの互換性のためにサポートされています。



8 文字のジャーナル名は以前のリリースでの DFHJnn 名の形式と比較するとかなりの柔軟性がありますが、多数のジャーナル (例えば、プログラム生成名の一部として端末名やのユーザー ID を使用するなど) を作成することはお勧めしません。

ジャーナル名 DFHLOG (EXEC CICS WRITE JOURNALNAME コマンド上) は、CICS システム・ログに書き込むことを示します。

FILE および PROFILE リソース定義で使用する場合、ジャーナル 番号 1 から 99 は、ジャーナル名 DFHJ01-99 にマップされます。これらのジャーナル名を特定の MVS ログ・ストリームにマップするには、JOURNALMODEL リソース定義を指定するか、それらをデフォルトにします。対応する JOURNALMODEL 定義を指定しない場合、デフォルトで、ユーザー・ジャーナルは *userid.applid.DFHJnn* の形式の LSN にマップされます。

60 ページの表 14 に、ユーザー・ジャーナル名テーブル・エントリー の例を示します。

表 14. MVS への出力用のユーザー・ジャーナル名エントリーの例	
ジャーナル・テーブル・エントリー - ユーザー・ジャーナル	最初の参照の際に作成されるエントリー
Name: JRNL001	API WRITE JOURNALNAME コマンドから派生した名前
Status: Enabled	ジャーナル・エントリーの作成時に設定
Type: MVS	このジャーナルは、JRNL001 名を参照する JOURNALMODEL によって MVS 出力用に定義 される
LSN: log_stream_name	デフォルトでは、log_stream_name は &reguserid..&applid..JRNL001 に解決 されるが、これは JOURNALMODEL 定義でユーザー定義にすることが できる

### システム・ログおよびジャーナル名のインストール

以前のリリースのジャーナル管理テーブルは廃止され、CICS によって動的に作成されるジャーナル名テーブルに置き換えられました。

CICS ログ・マネージャーには、CICS のシステム・ログまたは一般ログに対応するログ・ストリームの名前およびタイプ (MVS、SMF、またはダミー) が必要です。ICF カタログから直接取得される VSAM 順方向リカバリーのログ・ストリーム名を除き、CICS はこの情報をジャーナル名テーブルで保守します。このテーブルには、CICS がログ・ストリームに正常に接続したときに MVS システム・ロガーから返される、対応するログ・ストリームのトークンも含まれます。

### JOURNALMODEL の定義

CICS は、JOURNALMODEL 定義を使用して、ログ・ストリーム名を解決します。

### このタスクについて

CICS は、以下のタイミングでログ・ストリーム名を解決します。

#### システム・ログ

初期設定時、初期始動時のみ。

コールド・リスタート、ウォーム・リスタート、または緊急リスタート時に、CICS は CICS グローバル・カタログからログ・ストリーム名を取得します。

#### 一般ログ

CICS の始動後にジャーナル名が最初に参照されたとき、または、ログ・ストリームの切断された後、再びジャーナル名が最初に参照されたとき。ログ・ストリームが切断されると、一致する JOURNALMODEL リソース定義をさらに参照する必要が生じますが、この切断は次のように発生します。

#### ユーザー・ジャーナル

**DISCARD JOURNALNAME** コマンドを発行すると、直ちに発生します。

破棄されたジャーナル名が引き続き参照されると、CICS は一致する JOURNALMODEL リソース定義を探して、ログ・ストリーム名を再度解決する必要があります。変更された JOURNALMODEL 定義をインストールして、ユーザー・ジャーナルのログ・ストリーム名を変更できます。

#### ファイルの自動ジャーナル

自動ジャーナリング用のログ・ストリームを使用しているすべてのファイルが閉じられます。

#### 順方向リカバリー・ログ

順方向リカバリー・ロギング用のログ・ストリームを使用しているすべてのファイルが閉じられます。

JOURNALMODEL 定義では一般的に、総称ジャーナル名を指定します。そのため、同じ MVS ログ・ストリームに対して、総称名に一致するジャーナル名をマッピングします。JOURNALMODEL 定義は特定のモデルでも可能で、JOURNALMODEL 定義を使用して、多数のジャーナルまたは順方向リカバリー・ログを同じ MVS ログ・ストリームにマップしたり、それらを SMF に割り当てたりできます (61 ページの図 10 を参照)。

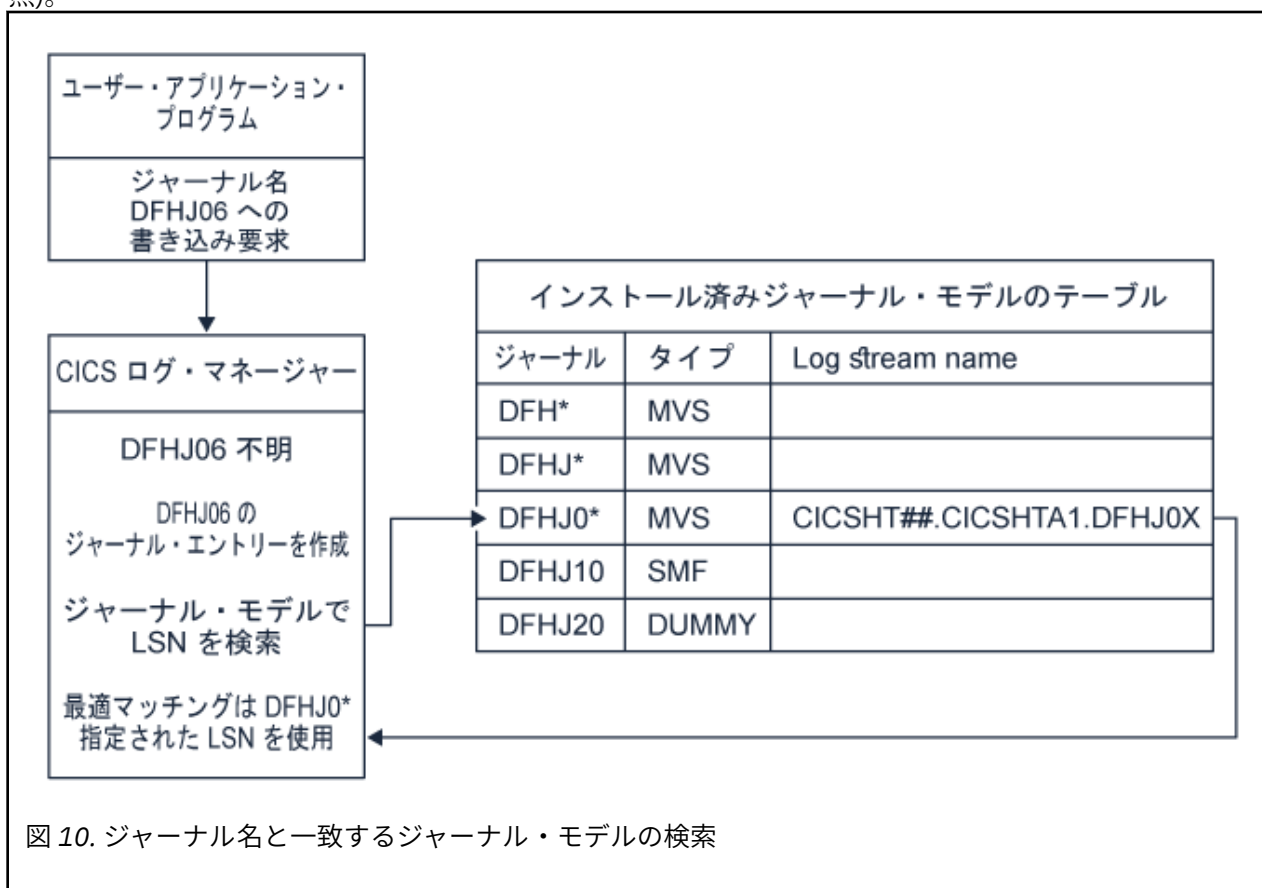


図 10. ジャーナル名と一致するジャーナル・モデルの検索

#### ログ・ストリームのマッピング

CICS は、VSAM カタログから直接取得される VSAM RLS 順方向リカバリー・ログ・ストリーム名以外のシステム・ログ名またはジャーナル名を、対応するログ・ストリーム名にマップします。

#### このタスクについて

CICS は、ユーザー定義の JOURNALMODEL リソース定義を使用するか (ある場合)、シンボル名を解決して作成されるデフォルト名を使用することによりこれを行います。

#### システム・ログ・ストリームのマッピング

コールド・スタート、ウォーム・スタート、緊急リスタートでは、CICS はシステム・ログ・ストリーム名を CICS グローバル・カタログから取得します。

システム・ログに対して JOURNALMODEL リソース定義を指定しない限り、CICS はデフォルトのログ・ストリーム名を使用します。

システム・ログに対する JOURNALMODEL 定義が存在する場合 (JOURNALNAME(DFHLOG) および JOURNALNAME(DFHSHUNT) が指定された JOURNALMODEL 定義を CICS が検出した場合)、それらの定義で指定されているシステム・ログ・ストリームへの接続を試みます。システム・ログ・ストリーム名は CICS 領域で固有である必要があります。

システム・ログに対する JOURNALMODEL リソース定義を定義する場合は、以下のことを確認してください。

- この定義で指定されたログ・ストリームが、MVS システム・ロガーに対して定義されている。
- または、適切なモデル・ログ・ストリームが定義され、ログ・ストリームが動的に作成できるようになっている。

適切な JOURNALMODEL 定義がない場合、CICS は、次のデフォルト名を使用してシステム・ログ・ストリームへの接続を試みます。

- userid.applid.DFHLOG
- userid.applid.DFHSHUNT

ここで、「userid」は CICS アドレス・スペースが実行される RACF ユーザー ID です。「applid」は領域の z/OS Communications Server の APPL 名です。CSD グループ DFHLGMOD に、デフォルトの DFHLOG と DFHSHUNT ログ・ストリーム用の CICS 提供の JOURNALMODEL 定義が含まれています。

これらのデフォルトのログ・ストリーム名を使用する前に、以下を行ってください。

- デフォルト・ログ・ストリームが、MVS システム・ロガーに対して明示的に定義されている。または
- または、適切なモデル・ログ・ストリームが定義され、ログ・ストリームが動的に作成できるようになっている。

これらのログ・ストリームが (MVS に定義されていないなどの理由で) 使用できない場合、または定義が (インストールされていないなどの理由で) 見つからない場合、CICS は `&sysname.LSN_last_qualifier.MODEL` という名前のモデル・ログ・ストリームを使用してシステム・ログ・ストリームを作成しようとします。

説明:

- `&sysname` は MVS イメージのシステム名に解決される MVS シンボルです。
- `LSN_last_qualifier` は、JOURNALMODEL リソース定義で指定された、ログ・ストリーム名の最終修飾子です。

DFHLOG および DFHSHUNT に対する JOURNALMODEL リソース定義を指定しない場合、またはグループ DFHLGMOD に用意されている CICS 定義を使用する場合、モデル名はデフォルトの `&sysname.DFHLOG.MODEL` および `&sysname.DFHSHUNT.MODEL` になります。これらのログ・ストリームが作成されると、CICS はそれらに接続します。

63 ページの図 11 に、初期始動時のシステム・ログのマッピング・プロセスを図示しています。

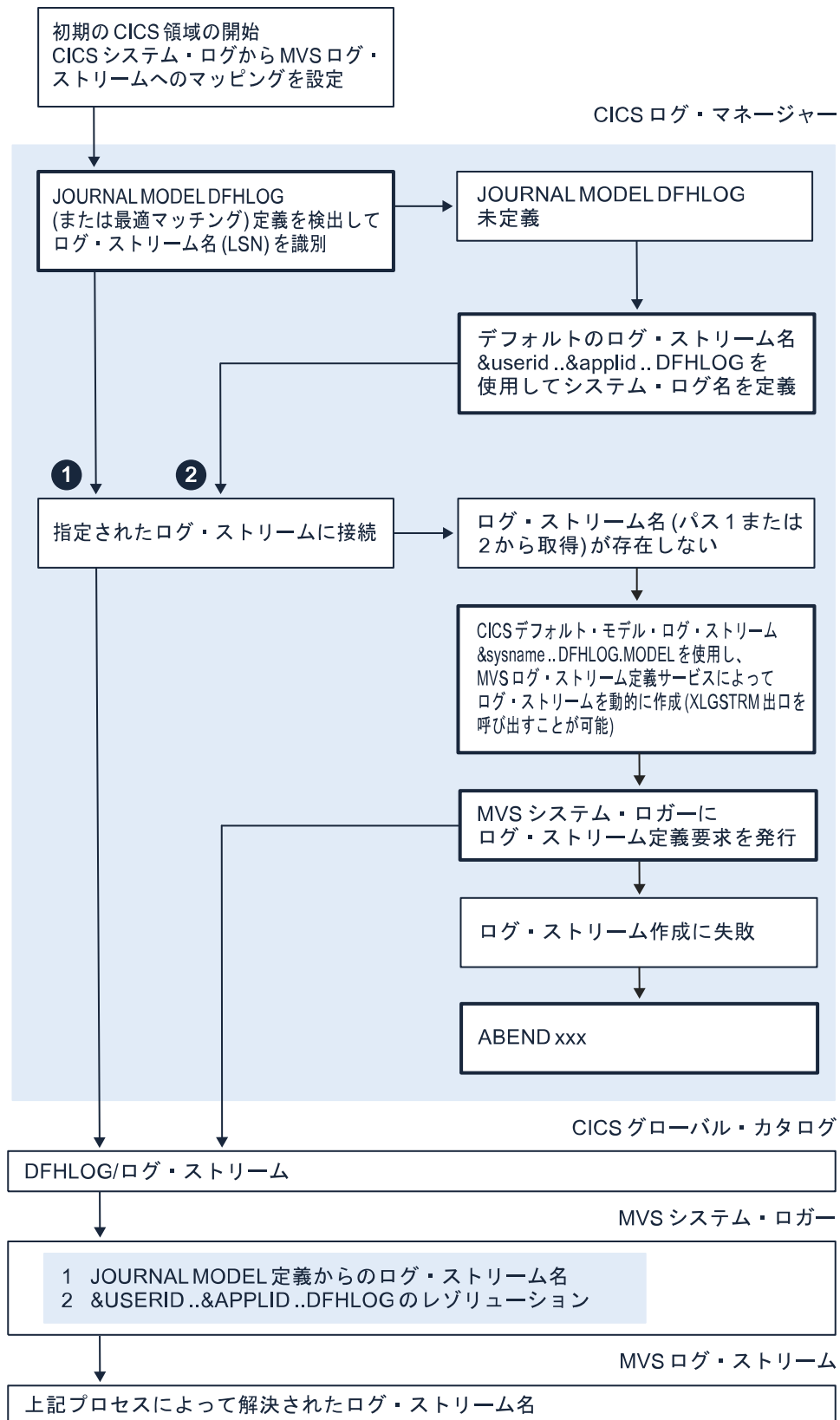


図 11. 初期始動時に CICS がシステム・ログ (DFHLOG) をログ・ストリーム名 (LSN) にマップする方法

### 一般ログ・ストリームのマッピング

CICS では、ユーザーがジャーナルまたはログに JOURNALMODEL リソースを定義しないと、デフォルトのログ・ストリーム名が使用されます。

ログに関する JOURNALMODEL 定義がある場合は、CICS ではこの定義で指定されているログ・ストリームへの接続が試行されます。

システム・ログに対する JOURNALMODEL リソース定義を定義する場合は、以下のことを確認してください。

- この定義で指定されたログ・ストリームが、MVS システム・ロガーに対して定義されている。
- または、適切なモデル・ログ・ストリームが定義され、ログ・ストリームが動的に作成できるようになっている。

JOURNALMODEL 定義内で名前が指定されているログ・ストリームに接続できない場合、CICS では以下のデフォルト名を使用してログ・ストリームへの接続が試行されます。

```
userid.applid.journalname
```

このデフォルト・ログ・ストリーム名を使用する前に、以下のことを確認してください。

- デフォルトのログ・ストリームが、MVS システム・ロガーに明示的に定義されている。
- または、適切なモデル・ログ・ストリームが定義され、動的に作成できるようになっている。

ログ・ストリームを使用できない(おそらく MVS に定義されていない)か、定義が見つからない(おそらくインストールされていない)場合は、CICS では以下のデフォルト名を使用してログ・ストリームの作成が試行されます。

```
LSN_QUALIFIER1.LSN_QUALIFIER2.MODEL
```

以下のように、修飾子フィールドは JOURNALMODEL 定義の streamname 属性に基づきます。

- 2つの名前 (*qualifier1.qualifier2*) のみから成る修飾名か、非修飾名のログ・ストリームが作成される場合、CICS ではモデル名を *qualifier1.MODEL* または *name.MODEL* として構成します。
- 3つ以上の名前 (*qualifier1.qualifier2....qualifier\_n*) から成る修飾名のログ・ストリームが作成される場合、CICS ではモデル名を *qualifier1.qualifier2.MODEL* として構成します。

ログ・ストリームが作成されると、CICS はそのログ・ストリームに接続します。

65 ページの図 12 は、一般ログのマッピング・プロセスのグラフィカル表現です。

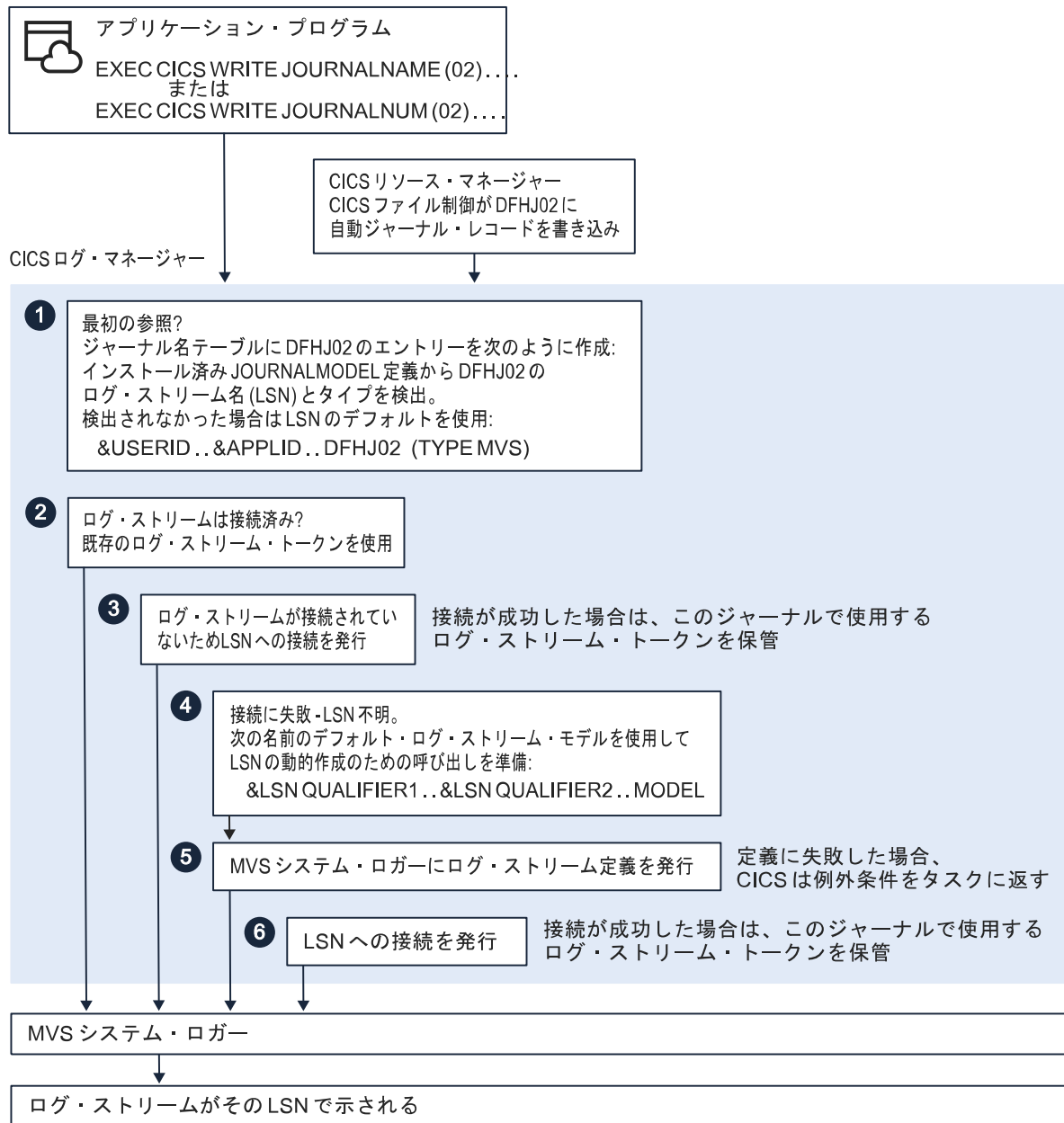


図 12. CICS ジャーナルをそのログ・ストリーム名 (LSN) にマップする方法

## MVS システム・ロガー・ログ・ストリームのデータの印刷またはコピー

CICS 提供のジャーナル・ユーティリティ・プログラム DFHJUP を使用して、MVS システム・ロガー・ログ・ストリームにあるデータを選択、印刷、またはコピーできます。あるいは、独自のユーティリティを使用して、これを行うこともできます。

### このタスクについて

ジャーナル・ユーティリティ・プログラム DFHJUP は、SUBSYS=(LOGR... 機能を使用します。独自のユーティリティを使用する場合は、そのユーティリティは SUBSYS=(LOGR... 機能を使用します。

DFHJUP の実行および SUBSYS=(LOGR.., 機能については、[バッチ・ジョブ \(DFHJUP\) の使用によるログ・ストリームの読み取り](#)を参照してください。

## カタログ式データ・セットの設定

CICS Transaction Server for z/OS, バージョン 5 リリース 6 用に、新規の CICS カタログを定義および初期設定する必要があります。

### このタスクについて

CICS が CICS システム情報をカタログするために必要な CICS グローバル・カタログ・データ・セットおよび CICS ローカル・カタログ・データ・セットを定義する必要があります。この資料の中では、これらのデータ・セットをグローバル・カタログおよびローカル・カタログと呼びます。CICS カタログ式データ・セットは、MVS システム・カタログには接続されず、CICS に固有のデータを含みます。

CICS でカタログがどのように使用されるか (開始と再開を含む) については、[130 ページの『CICS カタログの役割』](#)を参照してください。

グローバル・カタログに必要なスペース量を見積もるには、[68 ページの『スペース計算』](#)を参照してください。

カタログ・データ・セットをセットアップするには、次のようにします。

### 手順

1. グローバル・カタログを定義します。
2. ローカル・カタログを定義します。

### グローバル・カタログの定義

グローバル・カタログは、開始タイプの情報、CICS システム・ログの場所、リソース定義、端末管理情報、およびプロファイルを保管するために使用される VSAM キー順データ・セット (KSDS) です。

### このタスクについて

CICS はグローバル・カタログを使用して、以下のアクティビティを実行します。

- 可能な開始タイプ、および CICS システム・ログの場所を管理する情報を記録する。
- CICS がグループ・リストをインストールする際の初期設定時に、RDO **CEDA INSTALL** コマンドまたは **EXEC CICS CREATE** コマンドによってインストールされるリソース定義を、CICS の実行の間、保持する。
- 通常シャットダウン時に、端末管理情報およびプロファイルを記録する。その他のすべてのウォーム・キーポイント情報は CICS システム・ログに書き込まれます。

CICS ジョブの REGION パラメーターが、グローバル・カタログに使用されるバッファー・ストレージの増加に対応できるよう十分高い値であることを確認する必要があります。これは、このストレージは EDSA ではなく領域ストレージから取得されるためです。

CICS グローバル・カタログは 2 つの方法で定義および初期設定できます。以下で説明するサンプル・ジョブを使用するか、CICS 提供のジョブ DFHDEFDS を使用できます。

サンプル・ジョブを次のように編集します。



## 手順

1. CLUSTER 定義のデータ・セット名を、CICS 始動ジョブ・ストリーム内のグローバル・カタログの DD ステートメントの DSN パラメーターと同じ値に編集します。
2. 1 次エクステント・サイズおよび 2 次エクステント・サイズが *n1* シリンダーおよび *n2* シリンダーとして表示されます。インストール要件を満たすために必要なサイズを計算し、*n1* および *n2* を必要な値に置き換えます。  
グローバル・カタログ・データ・セットのスペース割り振りにどの **IDCAMS** パラメーターを使用する (CYLINDERS、TRACKS、または RECORDS) 場合でも、必ず 2 次エクステントを指定してください。グローバル・カタログ・データ・セットがいっぱいで、VSAM が 2 次エクステントを作成できない場合、CICS は異常終了します。レコード・サイズについて詳しくは、[69 ページの表 15](#) を参照してください。
3. **DEFINE CLUSTER** コマンドに REUSE オプションを指定します。  
このオプションにより、再使用可能クラスターとしてグローバル・カタログを繰り返し開けるようになります。また、DFHRMUTL ユーティリティの COLD\_COPY 入力パラメーターを使用する場合にも、REUSE を指定してください。
4. 必要に応じて、VSAM 定義の **CONTROLINTERVALSIZE** 値を編集します。  
このジョブは、最小または最大のバッファ・サイズを明示的に指定せず、VSAM により設定されたデフォルト値を受け入れます。特定のサイズのバッファを定義する場合は、明示的な値をコーディングできます。詳しくは、[71 ページの『バッファ・スペースのサイズ変更』](#) を参照してください。
5. リカバリー・マネージャー・ユーティリティ・プログラム DFHRMUTL を使用して、データ・セットを初期設定します。このユーティリティをジョブ・ステップ INITGCD に指定します。  
DFHRMUTL は、レコードをデータ・セットに書き込んで、このグローバル・カタログを使用した次の実行時に **START=AUTO** が指定されている場合に、CICS は初期始動を実行し、確認のためのプロンプトをオペレーターに出さないことを指定します。このレコードは、自動始動オーバーライド・レコードと呼ばれます。
6. DFHCCUTL ユーティリティを実行するためのジョブ・ステップを追加します。  
このステップを追加すると、グローバル・カタログとローカル・カタログがステップから外れることがなくなります。
7. CICS のデータ定義ステートメントを次のように定義します。

```
//DFHGCD DD DSN=CICSTS56.CICS.applid.DFHGCD,DISP=OLD
```

この例は、単一の CICS 領域で使用するグローバル・カタログの最低限の仕様を示しています。

8. 関連する **AMP** サブパラメーターを DD ステートメントに追加して、再始動時間とシャットダウン時間の改善に役立てます。  
**AMP** パラメーターについては、「[z/OS MVS JCL 解説書](#)」で説明しています。また、[141 ページの『CICS の始動』](#) の CICS 始動ジョブ・ストリームに例が示されています。

## 例

```
//GLOCAT JOB accounting info,,CLASS=A
//DEFGCD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    DEFINE CLUSTER -
        (NAME(CICSTS56.CICS.applid.DFHGCD) -
        INDEXED -
        CYLINDERS(n1 n2) -
        FREESPACE(10 10) -
        SHAREOPTIONS(2) -
        RECORDSIZE(4089 32760) -
        REUSE -
        VOLUMES(volid)) -
    DATA -
        (NAME(CICSTS56.CICS.applid.DFHGCD.DATA) -
        CONTROLINTERVALSIZE(32768) -
        KEYS(52 0)) -
    INDEX -
        (NAME(CICSTS56.CICS.applid.DFHGCD.INDEX) )
/*
//INITGCD EXEC PGM=DFHRMUTL,REGION=1M
//STEPLIB DD DSN=CICSTS56.CICS.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A
//DEFGCD DD DSN=CICSTS56.CICS.applid.DFHGCD,DISP=OLD
//SYSIN DD *
SET_AUTO_START=AUTOINIT
/*
```

図 13. グローバル・カタログを定義および初期設定するジョブの例

## スペース計算

インストールされたリソース定義、テーブル・エントリー、および制御ブロックにキーポイントを設定するため、グローバル・カタログ内に必要なスペースの量を見積もります。各グローバル・カタログ・レコードには 52 バイトのキーが含まれます。各エントリーは 1 つの VSAM レコードで、各タイプのテーブルのレコードには、それぞれ異なるキーが設定されています。

スペースの使用率を定期的に確認して、エクステントの実行がご使用の環境の上限に近過ぎないことを確認する必要があります。69 ページの表 15 で指定されたサイズを元に、グローバル・カタログ内に必要なスペースを見積もります。

DFHGCD などの VSAM KSDS のスペース所要量は、CICS コールド・スタートの方法に応じて異なります。VSAM KSDS に格納される予定の CICS 定義に変更が行われなかった場合、必要なスペースは異なります。これは、データ・セットが初期化されたばかりなのか、前回の CICS の実行時のデータがデータ・セットに残っているかによって、VSAM がデータ・セットで使用するスペースの量が異なるためです。CICS は VSAM を呼び出して 順次書き込みを行います。順次追加されるレコードのキーが、既存のキーの中で最大のものよりも値が高い場合、VSAM はデータ・セットの定義に指定された **freespace** 値を尊重します。ただし、データ・セットに含まれている既存のレコードのキーが、追加されるものより値が高い場合、**freespace** 値は CI 分割が行われた後にのみ尊重されます。

データ・セットの索引部分のサイズは、CI および CA 分割が行われた回数によって異なります。このサイズは、索引のシーケンス・セットに影響を与えます。

グローバル・カタログを初期化する際、**COLD\_COPY** パラメーターを

SET\_AUTO\_START=AUTOCOLD,COLD\_COPY のように使用できます。コールド・コピーは、作成するグローバル・カタログ・データ・セットのコピーの数を減らすことができるため、コールド・スタートのパフォーマンスが改善されます。最初のコールド・スタート以降、CI 分割は行われなくなり、データ・セットが追加のエクステントに拡張することはありません。代替手段として、時々データ・セットを再編成したり再初期化したりする方法もあります。

注：表 1 には、「エントリーあたりのバイト数」列に、キーの長さが 52 バイトの項目が含まれます。

表 15. グローバル・カタログにおけるエントリーのサイズ

インストール済みの定義、テーブル・エントリー、 または制御ブロック	エントリーあたりのバイト数	以前のリリースのサイズ (差異がある場合)
インストール済みの ATOMSERVICE 定義	660 バイト	
インストール済みの BUNDLE 定義 <b>1</b>	300 バイトから約 800 バイトまで	
インストール済みの CONNECTION 定義	528 バイト	440 バイト
インストール済みの CORBASERVER 定義	1304 バイト	1216 バイト
インストール済みの DB2CONN 定義	1548 バイト	1460 バイト
インストール済みの DB2ENTRY 定義	332 バイト	244 バイト
インストール済みの DB2TRAN 定義	160 バイト	62 バイト
インストール済みの DJAR 定義	432 バイト	344 バイト
インストール済みの DOCTEMPLATE 定義	284 バイト	196 バイト
インストール済みの ENQMODEL 定義	152 バイト	64 バイト
インストール済みの EVENTBINDING および CAPTURESPEC 定義 <b>2</b>	約 4000 バイト	
インストール済みの区画外キュー定義	392 バイト	296 バイト
インストール済みの FILE 定義	588 バイト	500 バイト
インストール済みの間接キュー定義	180 バイト	92 バイト
インストール済みの区画内キュー定義	328 バイト	240 バイト
インストール済みの IPCONN 定義	402 バイト	312 バイト
インストール済みの JOURNALMODEL 定義	168 バイト	80 バイト
インストール済みの JVM プログラム定義	168 バイトから 307 バイト まで	80 バイトから 219 バイト まで
インストール済みの JVMSERVER 定義	146 バイト	
インストール済みの LIBRARY 定義	852 バイト	764 バイト
インストール済みの MQCONN 定義	620 バイト	
インストール済みの MQINI 定義	212 バイト	
インストール済みの PARTNER 定義	148 バイト	124 バイト
インストール済みの PIPELINE 定義	1500 バイト	1412 バイト
インストール済みの PROCESSTYPE 定義	148 バイト	60 バイト
インストール済みの PROFILE 定義	158 バイト	70 バイト
インストール済みの PROGRAM 定義	168 バイト	44 バイト
インストール済みの REQUESTMODEL 定義	226 バイト	138 バイト
インストール済みのリモート・キュー定義	172 バイト	84 バイト
インストール済みの TCIPSERVICE 定義	924 バイト	580 バイト
インストール済みのモデル TERMINAL 定義 <b>3</b>	634 バイト	610 バイト
インストール済みの TRANCLASS 定義	124 バイト	36 バイト
インストール済みの TRANSACTION 定義 (TPNAME なし)	244 バイト	140 バイト

表 15. グローバル・カタログにおけるエントリーのサイズ (続き)

インストール済みの定義、テーブル・エントリー、 または制御ブロック	エントリーあたりのバイ ト数	以前のリリースのサイズ (差異がある場合)
インストール済みの TRANSACTION 定義 (TPNAME または XTPNAME あり)	388 バイト	204 バイト
インストール済みの TSMODEL 定義	236 バイト	148 バイト
インストール済みの TYPETERM 定義 <b>3</b>	634 バイト	610 バイト
インストール済みの VSAM ファイル (またはデー タ・テーブル) 定義	312 バイト	288 バイト
インストール済みの URIMAP 定義	1316 バイト	1220 バイト
インストール済みの WEBSERVICE 定義	1040 バイト	936 バイト
BDAM ファイル管理テーブル・エントリー (FCT)	284 バイト	146 バイト
BDAM データ制御ブロック	156 バイト	132 バイト
VSAM LSR 共有制御ブロック <b>4</b>	1224 バイト	1184 バイト
データ・セット名 (JCL または動的な割り振りによ る) <b>5</b>	96 バイト	72 バイト
データ・セット名ブロック	194 バイト	170 バイト
イベント処理の状況	89 バイト	
ファイル制御リカバリー・ブロック <b>6</b>	149 バイト	125 バイト
端末管理テーブル・エントリー (TCT)	約 1552 バイト	約 1500 バイト
ダンプ・テーブル・エントリー	100 バイト	76 バイト
インターバル制御エレメント (ICE)	120 バイト	96 バイト
自動開始記述子 (AID)	120 バイト	96 バイト
据え置き作業エレメント (DWE) <b>7</b>	132 バイト	108 バイト
インストール済みのジャーナル	111 バイト	88 バイト
リカバリー・マネージャーのリモート名	158 バイト	134 バイト
一時データ宛先レコード	70 バイト	46 バイト
一時データ宛先の補助レコード	58 バイト	34 バイト
ローダーのプログラム定義	68 バイト	44 バイト
セッション TCTTE	918 バイト	894 バイト
ログ・ストリーム	112 バイト	88 バイト
Uri 仮想ホスト	180 バイト	156 バイト

## 注

1. バンドル／カタログ・レコードのサイズは、バンドルに含まれるパーツの数、バンドル・ディレクトリー  
のサイズ、およびにスコープの長さによって異なります。スコープなしの、小さいディレクトリーの  
バンドルの最小サイズは、約 300 バイトです。それに対してスコープありの、大きなディレクトリー  
のバンドルは、バイト数が 800 にまで上ります。各バンドル・パーツについては、そのサイズは 300  
バイトから 800 バイトまでの範囲に及びます。
2. イベント・バインディングのカタログ・レコードのサイズは、イベント・バインディングのキャプチャ  
ー仕様と、各キャプチャー仕様にあるフィルターの数や、キャプチャー・データ項目の数に依存しま  
す。必要な場合は、各イベント・バインディングに複数のカタログ・レコードを持つこともできます。

3. TYPETERM およびモデル TERMINAL 定義は、自動インストールを使用している場合に存在します。これらは、CEDA トランザクションによって、またはグループ・リストを介してインストールされたグループのメンバーとして、定義のインストール時にグローバル・カタログに直接格納されます。例えば、CICS を始動パラメーター GRPLIST=DFHLIST で始動した場合、DFHTERM グループおよび DFHTYPE グループで定義されている CICS 提供の TYPETERM およびモデル端末定義はグローバル・カタログに記録されます。CICS 領域にインストールされているすべての自動インストールのリソースの計算に、スペースを割り振ります。
4. 各 LSR プールに 1 つ。つまり、8 つ。
5. VSAM パスを開く場合、2 つ。BDAM または VSAM ベースのデータ・セットの場合は 1 つ。
6. VSAM RLS SHCDS オプション NONRLSUPDATEPERMITTED を使用する場合は、これらが含まれます。この場合、NONRLSUPDATEPERMITTED を指定したデータ・セットごとに上限を指定できます。この上限は、データ・セットにアクセスするために使用する異なるファイル名の数に、データ・セットを更新するタスクの数を乗算した数です。通常は、これらの制御ブロックは、ある場合でも数個です。
7. 指定されている値は、LU6.1 セッションまたは APPC セッションからチェーニングする DWE 用です。

### バッファ・スペースのサイズ変更

バッファ・スペース・パラメーターを使用して、グローバル・カタログ・データ・セットの VSAM 定義で特定サイズのバッファを定義できます。

CICS ジョブの REGION パラメーターが、グローバル・カタログに使用されるバッファ・ストレージの増加に対応できるよう十分高い値であることを確認する必要があります。これは、このストレージは ED SA ではなく領域ストレージから取得されるためです。

VSAM はデフォルトで、データ・コンポーネントの制御インターバル・サイズとインデックスの制御インターバル・サイズの 2 倍に相当するバッファ・スペースを使用します。サンプル・ジョブでは、この計算に基づく 69632 バイトがデフォルト値になっています。最小バッファ・サイズとして大きめの値を設定すると、コールド・スタートとウォーム・リスタートの時間が短縮され、CICS のシャットダウン時間が大幅に削減される可能性があります。

バッファ・サイズを制御するために 2 つのパラメーターを使用できます。**BUFFERSPACE** パラメーターでは、最小バッファ・スペース・サイズを定義します。**BUFSP** パラメーターでは、最大バッファ・サイズを定義します。どちらのパラメーターも、サンプル・ジョブに追加してバッファ・サイズを設定できます。

パフォーマンス上の理由から、CICS では、STRNO (ストリング数) の値が 32 に定義されています。サンプル・ジョブに基づいて、BUFSP の最小値は以下のように計算されます。

```
BUFND = (STRNO + 1)
BUFNI = STRNO
BUFSP = 33 * 32768 (BUFND * CI size) + 32 * 1024 (BUFNI * CI size) =
1114112 bytes
```

CICS 始動ジョブ・ストリームの DD ステートメントで **AMP** パラメーターを使用することによって、グローバル・カタログ・データ・セットのバッファ・スペースを定義することも可能です。そのパラメーターを使用して、デフォルト値や定義値をオーバーライドできます。**AMP BUFSP** の指定するバイト数がアクセス方式サービス・プログラムの DEFINE コマンドの **BUFFERSPACE** パラメーターより少ない場合、**BUFFERSPACE** の数は **BUFSP** の数をオーバーライドします。

### グローバル・カタログの再利用によるコールド・スタートの実行

コールド・スタートを実行する必要がある場合は、グローバル・カタログ・データ・セットの削除および再定義を行わないでください。

グローバル・カタログを削除して再定義した場合は、CICS が初期 始動を実行するので、リモート・システムに関するリカバリー情報はすべて失われます。リモート・システムに再接続する際に、作業単位を再同期するのに必要な情報が失われたことの通知が CICS から出され、ローカル・システムとリモート・システムの両方で、メッセージが表示されてこの事実が記録されます。

代わりに、次の始動がコールド・スタートになるように指定するには、SET\_AUTO\_START=AUTOCOLD オプションを指定して DFHRMUTL ユーティリティーを使用します。これには、以下の利点があります。

- START システム 初期設定パラメーターを AUTO から COLD にリセットしたり元に戻したりする必要があります。
- グローバル・カタログとシステム・ログに十分な情報が保存されるので、CICS がリモート・システムに関する情報をログからリカバリーしたり、作業単位の再同期に使用できる方法でリモート・システムに回答したりできます。

DFHRMUTL COLD\_COPY オプションを使用して、コールド・スタートに必要なレコードだけを別のカタログ式データ・セットにコピーすると、コールド・スタートに要する時間を短縮できます。DFHRMUTL で設定された戻りコードに、コピーが正常に実行されたことが示されたら、後続のジョブ・ステップで、新しい(大規模で空の)カタログを元のカタログ式データ・セットにコピーし戻すことができます。始動時に CICS がすべての定義レコードをカタログから削除するのに要する時間が省かれるので、パフォーマンスが向上します。同じ理由で、この技法により初期始動に要する時間も短縮されます。この技法の例が [72 ページの図 14](#) に示されています。

**注:** COLD\_COPY を使用する前に、コールド・スタートと初期始動のどちらを実行するかを把握しておく必要があります。保護のために、元のグローバル・カタログのバックアップ・コピーを作成した後に、DFHRMUTL で新しいカタログ出力を上書きコピーしてください。CICS のコールド・スタート時のグローバル・カタログの使用について詳しくは、[130 ページの『始動と再始動の制御』](#)を参照してください。

```
//RMUTL      EXEC PGM=DFHRMUTL,REGION=1M
//STEPLIB DD DSNAME=CICSTS56.CICS.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A
//DFHGCD DD DSNAME=CICSTS56.CICS.applid.DFHGCD,DISP=OLD
//NEWGCD DD DSNAME=CICSTS56.CICS.applid.COPY.DFHGCD,DISP=OLD
//SYSIN DD *
SET_AUTO_START=AUTOCOLD,COLD_COPY
/*
//          IF (RMUTL.RC<16) THEN
//* Steps to be performed if RMUTL was a success
//COPY      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//DFHGCD DD DSNAME=CICSTS56.CICS.applid.DFHGCD,DISP=OLD
//NEWGCD DD DSNAME=CICSTS56.CICS.applid.COPY.DFHGCD,DISP=OLD
//SYSIN DD *
REPRO INFILE(NEWGCD) OUTFILE(DFHGCD) REUSE
/*
//* End of steps to be performed if RMUTL was a success
//          ENDIF
```

図 14. DFHRMUTL の例 – コールド・スタートに関するグローバル・カタログの設定

## ローカル・カタログの定義

ローカル・カタログは、VSAM キー順データ・セット (KSDS) です。ローカル・カタログは、他の CICS 領域と共有されません。

## 始める前に

グローバル・カタログは、インストールされたリソース定義の増加に対応するために十分なスペースを確保して定義する必要がありますが、これとは異なり、ローカル・カタログのサイズは比較的静的なものです。以下のセクションでは、ローカル・カタログに保持される情報について説明します。

CICS Transaction Server for z/OS は、ドメインと呼ばれる機能領域(つまり、コンポーネント)に分割されます。これらの各ドメインは、中央コンポーネントである CICS カーネルを介して通信し、その初期設定と終了は、ドメイン・マネージャーによって制御されます。すべてのドメインは個別のドメイン・パラメーター・レコードを必要とし、それらはローカル・カタログに保管されます。CICS ドメインはローカル・カタログを使用して、CICS の実行と実行の間で情報の一部を保管し、この情報をコールド・スタートが行われても保持します。ローカル・カタログに書き込まれる内容、および CICS が始動と再始動にローカル・カタログを使用する方法に関するガイダンス情報については、[130 ページの『始動と再始動の制御』](#)を参照してください。

## このタスクについて

CICS ローカル・カタログは 2 つの方法で定義および初期設定できます。以下で説明するサンプル・ジョブを使用するか、CICS 提供のジョブ DFHDEFDS を使用して、アクティブな CICS 領域用のローカル・カタログを作成できます。

サンプル・ジョブを次のように編集します。

## 手順

1. 複数の CICS 領域に対してローカル・カタログを定義する場合、各 CICS の特定のアプリケーション ID をデータ・セット修飾子の 1 つとすることにより、クラスターを一意に識別します。

例えば、アクティブ CICS 領域と代替 CICS 領域のクラスターに以下の名前を使用するとします。ここで、DBDCCIC1 と DBDCCIC2 は特定のアプリケーション ID です。

```
DEFINE CLUSTER -
  (NAME( CICSTS56.CICS.DBDCCIC1.DFHLCD))

DEFINE CLUSTER -
  (NAME( CICSTS56.CICS.DBDCCIC2.DFHLCD))
```

2. ローカル・カタログの場合、200 件のレコード用のスペースがおそらく適切なスペース量です。ただし、不測の事態に備えて 2 次エクステント用のスペースを指定してください。
3. ローカル・カタログのレコードは、グローバル・カタログと比べると小さいサイズになっています。示されたレコード・サイズ (指定されたレコード数が含まれ、データ・セットに十分なスペースが確保されます) を使用します。
4. ローカル・カタログのデータ定義ステートメントを次のように定義します。

```
//DFHLCD DD DSN=CICSTS56.CICS.applid.DFHLCD,DISP=OLD
```

## 例

```
//LOCAT JOB accounting info,,CLASS=A
//DEFLCD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
      DEFINE CLUSTER -
              (NAME( CICSTS56.CICS.applid.DFHLCD) - 1
              INDEXED -
              RECORDS( 200 10 ) - 2
              FREESPACE(10 10) -
              SHAREOPTIONS( 2 ) -
              REUSE -
              VOLUMES( volid )) -
      DATA
              (NAME( CICSTS56.CICS.applid.DFHLCD.DATA ) -
              KEYS( 52 0 ) -
              RECORDSIZE( 70 2041 ) - 3
              CONTROLINTERVALSIZE( 2048 )) -
      INDEX (NAME( CICSTS56.CICS.applid.DFHLCD.INDEX ) )
/*
//*****
//INITLCD EXEC PGM=DFHCCUTL
//*
//* INITIALIZE THE CICS LOCAL CATALOG
//*
//STEPLIB DD DSN=CICSTS56.CICS.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DFHLCD DD DSN=CICSTS56.CICS.applid.DFHLCD,DISP=SHR
//*
//
```

図 15. ローカル・カタログを定義および初期設定するサンプル・ジョブ



## ローカル・カタログの初期設定

ローカル・カタログを使用して CICS 領域を始動するためには、ドメイン・マネージャー・パラメーター・レコードと 3 つのローダー・ドメイン・パラメーター・レコードのデータを使用して、ローカル・カタログを初期設定する必要があります。

### このタスクについて

ドメイン・マネージャー・パラメーター・レコードには、CICS ドメインに関連する情報が含まれており、ドメイン・ネームによって識別されます。完全なリストについては、[74 ページの表 16](#)を参照してください。

3 つのローダー・ドメイン・パラメーター・レコードには、以下のリソースに関連する情報が含まれています。

- DFHDMP、CSD ファイル・マネージャー
- DFHEITSP、RDO 言語定義テーブル
- DFHPUP、CSD パラメーター・ユーティリティ・プログラム

### 手順

1. すべてのレコードを正しい順序で使用してローカル・カタログを適切に初期設定するために、VSAM データ・セットを定義した直後に、CICS 提供のユーティリティ DFHCCUTL を実行します。  
このユーティリティは、必要なデータをローカル・カタログから CICS ドメインに書き込みます。
2. ローダー・ドメインは、CICS の各中核モジュールのプログラム定義レコードも書き込みます。レコード数は、CICS 領域に含めた機能レベルに応じて異なります。少なくとも 75 個のローダー・ドメイン・レコードを見込んでおいてください。
3. ストレージ・マネージャー・ドメイン・サブプール用の CICS 自己調整メカニズムを有効にするために、CICS 提供のユーティリティ DFHSMUTL を使用してローカル・カタログにレコードを追加します。

### 例

表 16. ドメイン・ネームのリスト	
カタログのドメイン・ネーム	説明
DFHAP	アプリケーション・ドメイン
DFHAS	非同期サービス・ドメイン
DFHBA	ビジネス・アプリケーション・マネージャー
DFHCC	CICS ローカル・カタログ・ドメイン
DFHDD	ディレクトリー・マネージャー・ドメイン
DFHDDH	文書ハンドラー・ドメイン
DFHDM	ドメイン・マネージャー・ドメイン
DFHDP	デバッグ・プロファイル・ドメイン
DFHDS	ディスパッチャー・ドメイン
DFHDU	ダンプ・ドメイン
DFHEJ	Enterprise Java™ ドメイン
DFHEM	イベント・マネージャー・ドメイン
DFHEP	イベント処理ドメイン
DFHGC	CICS グローバル・カタログ・ドメイン
DFHIE	TCP/IP 上の ECI ドメイン

表 16. ドメイン・ネームのリスト (続き)

カタログのドメイン・ネーム	説明
DFHKE	カーネル・ドメイン
DFHLD	ローダー・ドメイン
DFHLG	ログ・マネージャー・ドメイン
DFHLM	ロック・マネージャー・ドメイン
DFHME	メッセージ・ドメイン
DFHML	XML サービス・ドメイン
DFHMN	モニター・ドメイン
DFHNQ	エンキュー・マネージャー・ドメイン
DFHOT	オブジェクト・トランザクション・ドメイン
DFHPA	システム初期設定パラメーター・ドメイン
DFHPG	プログラム・マネージャー・ドメイン
DFHPI	パイプライン・マネージャー・ドメイン
DFHPT	パートナー管理ドメイン
DFHRL	リソース・ライフサイクル・ドメイン
DFHRM	リカバリー・マネージャー・ドメイン
DFHRS	領域状況ドメイン
DFHRX	RRMS ドメイン
DFHRZ	要求ストリーム・ドメイン
DFHSH	スケジューラー・サービス・ドメイン
DFHSJ	JVM ドメイン
DFHSM	ストレージ・マネージャー・ドメイン
DFHSO	ソケット・ドメイン
DFHST	統計ドメイン
DFHTI	タイマー・ドメイン
DFHTR	トレース・ドメイン
DFHTS	一時記憶域ドメイン
DFHUS	ユーザー・ドメイン
DFHWB	Web ドメイン
DFHW2	Web 2.0 ドメイン
DFHXM	トランザクション・マネージャー・ドメイン
DFHXS	セキュリティ・ドメイン

## ローカル・カタログに書き込むドメイン

一部のドメインは、ウォーム・リスタートまたは緊急リスタートで使用するために、ローカル・カタログにドメイン状況レコードを書き込みます。例えば、ダンプ・ドメインの場合、前回の実行時に使用中だったトランザクション・ダンプ・データ・セットが状況レコードに示されます。

ローカル・カタログに書き込むドメインは以下のとおりです。

- ディスパッチャー・ドメイン
- ダンプ・ドメイン
- ローダー・ドメイン
- メッセージ・ドメイン
- パラメーター・マネージャー・ドメイン
- ストレージ・マネージャー・ドメイン
- 一時データ

## 補助トレース・データ・セットの設定

CICS トレース・エントリは、補助トレース・データ・セット (CICS 所有の BSAM データ・セット) に送信することができます。補助トレースを使用する場合、CICS の開始前にデータ・セットを作成する必要があります。CICS 実行中はデータ・セットの定義は行えません。

### このタスクについて

CICS トレース・ドメインで処理されるトレース・データの宛先は、複数の宛先から選択できます。以下の宛先の任意の組み合わせをいつでもアクティブにしておくことができます。

- CICS アドレス・スペース内の主記憶域における内部トレース・テーブル。
- ディスク上またはテープ上の BSAM データ・セットとして定義された補助トレース・データ・セット。
- z/OS 汎用トレース機能 (GTF) データ・セット
- z/OS Unix システム・サービス内の JVM サーバー・トレース・ファイル

補助トレースには次のような利点があります。

- 十分な大きさのトレース・データ・セットを最初に定義すれば、大量のトレース・データを収集できます。これは、例えば、長期間にわたってシステム・アクティビティをトレースする場合などに行います (おそらく、予測不能な記憶保護違反の問題を解決することが目的です)。
- 補助スイッチ (**AUXTRSW** システム初期設定パラメーター) を使用して、補助トレース・データが後続のトレース・データによって上書きされないように指定できます。内部トレース・テーブルでは、テーブルの終わりに達すると、後続の項目がテーブルの始まりの項目を上書きし、診断情報が失われます。
- 補助トレースは、始動時に CICS トレースを使用する場合に特に有用です。これは、CICS の初期化時に大量のトレース項目が書き込まれるためです。

補助トレースは、ディスクまたはテープ上の 1 つまたは 2 つの順次データ・セットに保持されます。補助トレース・データ・セットの DD 名は、CICS により DFHAUXT および DFHBUXT として定義されます。データ・セットを 1 つだけ定義する場合には、その DD 名は DFHAUXT としてください。

トレース項目は可変長ですが、補助トレース・データ・セットに書き込まれるデータの物理レコード長 (ブロック・サイズ) は 4096 バイトに固定されています。およその目安として、各ブロックには平均 40 件のトレース項目が含まれます。ただし、トレース項目の実際数は、実行中の処理によって異なります。

### 手順

1. 補助トレースに定義する順次データ・セットを 1 つにするか 2 つにするかを決定します。  
補助トレース・データ・セットに自動切り替えを指定する場合、2 つのデータ・セットを定義する必要があります。補助トレースに自動切り替えを指定する場合に、データ・セットを 1 つしか定義しないと、補助トレースが停止して、システム・ダンプが生成されます。
2. 補助トレース・データ・セットの場所を決定します。

補助トレース出力の記録にテープを使用する場合は、ラベルなしテープを使用します。標準ラベルのテープを使用すると、それが設置されている磁気テープ・ドライブが1つか2つかにかかわらず、CICS ステップが完了するまで、DFHTU730 ユーティリティを使用したあらゆるボリュームの内容の処理が停止します。標準ラベルのテープを使用する必要がある場合は、CICS の実行で生成されるすべての出力が、マウントされた1つ(または2つ)のボリュームに収まることを確認してください。

ラベルなしテープ上のデータ・セットはカタログできません。

3. ディスク上に補助トレース・データ・セットを定義する場合、CICS を開始する前に、補助トレース・データ・セットを割り振ってカタログします。以下のいずれかの方法を使用します。

- 77 ページの『補助トレース・データ・セットを割り振るためのサンプル・ジョブ』に示されているサンプル・ジョブを使用して、データ・セットを割り振ってカタログします。
- 用意されているジョブ DFHDEFDS を使用して、補助トレース・データ・セットを作成します。詳しくは、『インストール』の『CICS データ・セットの作成』を参照してください。

4. 補助データ・セットにテープを使用し、CICS の始動により補助トレースをアクティブにする場合は、CICS を始動する前に、テープ装置を割り当ててテープをマウントします。  
CICS の実行時にコマンドを入力して補助トレースを開始する予定の場合は、コマンドを入力する前にテープがマウントされていることを確認してください。
5. オプション: データ・セットを暗号化する場合は、102 ページの『データ・セットの暗号化』を参照してください。
6. 始動ジョブ・ストリームで補助トレース・データ・セットを CICS に定義します。

- a) ディスク上の補助トレース・データ・セットの場合は、次の DD ステートメントを使用します。

```
//DFHAUXT DD DSN=CICSTS56.CICS.applid.DFHAUXT,DCB=BUFNO=n,DISP=SHR
//DFHBUXT DD DSN=CICSTS56.CICS.applid.DFHBUXT,DCB=BUFNO=n,DISP=SHR
```

1 より大きい BUFNO を指定すると、補助トレース・レコードの書き込みに関連した I/O オーバーヘッドを減らすことができます。4 から 10 までの値を指定すると、補助トレースを使用して実行される際の I/O オーバーヘッドを大幅に減らせます。DISP=SHR により、他のデータ・セットへの切り替えが行われた後に、DFHTU730 オフライン・ユーティリティ・プログラムによるデータ・セットの同時処理が可能になります。

- b) ラベルなしテープ上の補助トレース・データ・セットの場合は、次のサンプル DD ステートメントを使用します。

```
//DFHAUXT DD DSN=CICSTS56.CICS.applid.DFHAUXT,UNIT=3400,VOL=SER=valid,
// DISP=(NEW,KEEP),LABEL=(,NL)
//DFHBUXT DD DSN=CICSTS56.CICS.applid.DFHBUXT,UNIT=3400,VOL=SER=valid,
// DISP=(NEW,KEEP),LABEL=(,NL)
```

### 補助トレース・データ・セットを割り振るためのサンプル・ジョブ

ディスクで補助トレース・データ・セットを定義する場合は、CICS を実行する前に、このサンプル・ジョブでデータ・セットを割り振ってカタログに登録できます。

#### 手順

1. このサンプル・ジョブに含まれている DCB サブパラメーターで、CICS 補助トレース・データ・セットに必要な DCB 属性を指定します。このジョブの代わりに、CICS 始動ジョブ・ストリームの DD ステートメントで (NEW,CATLG) を指定し、DCB パラメーターを省略して、CICS が同じデフォルト値でデータ・セットを開くようにする、という方法もあります。
2. インストール環境の要件に合わせて、このサンプル・ジョブ・ストリームでのスペース割り振りを変更してください。

## 例

```
//DEFTRCDS JOB (accounting information),
//          MSGCLASS=A,MSGLEVEL=(1,1),
//          CLASS=A,NOTIFY=userid
//*****
//*          Create auxiliary trace data sets
//*****
//ALLOCDSD EXEC PGM=IEFBR14
//DFHAUXT DD DSN=CICSTS56.CICS.applid.DFHAUXT,UNIT=3380,VOL=SER=valid,
//          DISP=(NEW,CATLG),DCB=(BLKSIZE=4096,RECFM=F,LRECL=4096), 1
//          SPACE=(CYL,(25)) 2
//DFHBUXT DD DSN=CICSTS56.CICS.applid.DFHBUXT,UNIT=3380,VOL=SER=valid,
//          DISP=(NEW,CATLG),DCB=(BLKSIZE=4096,RECFM=F,LRECL=4096), 1
//          SPACE=(CYL,(25)) 2
//
```

図 16. ディスクで補助トレース・データ・セットを定義するためのサンプル・ジョブ

## ダンプ・データ・セットの定義

CICS は、CICS の実行中に検出された障害の結果として発生したダンプを記録するため、または明示的な要求に応じて、ダンプ・データ・セットを使用します。

### このタスクについて

次の 2 つのタイプのダンプ・データ・セットを定義する必要があります。

1. MVS システム・ダンプ・データ・セット。これは、MVS SDUMP マクロを使用して CICS で要求されたシステム・ダンプを記録します。
2. CICS トランザクション・ダンプ・データ・セット。これは、トランザクション・ダンプを記録します。

### 手順

1. CICS の初期インストールでは、5 MB から 10 MB のダンプ・データ・セットを定義します。  
このダンプ・データ・セットは、通常操作の開始時に、各インストール済み環境の独自の要件に合わせて調整できます。
2. MVS **CHNGDUMP** コマンドを使用して SDUMP オプションを変更します。  
CICS 領域をセットアップする際、**CHNGDUMP** コマンドの使用について十分に検討する必要があります。
  - a) MERGE 関数を使用して、CICS によりダンプの対象として選択された領域が MVS ダンプ・データ・セット出力に含まれていることを確認します。
  - b) ADD オプションを使用して、SDUMP の発行時に CICS によって指定されたオプションを置き換えます。  
これにより、部分ダンプが MVS ダンプ・データ・セットに取得される可能性があります。
3. 81 ページの『[トランザクション・ダンプ・データ・セットの定義](#)』の説明に従って、CICS トランザクション・ダンプ・データ・セットを定義します。

### タスクの結果

CICS を初めて始動したときに、CICS は各ダンプ・テーブル・オプションに対してシステム・デフォルト値を使用し、ユーザーが変更するまでそのシステム・デフォルト値を使用し続けます。

CICS には、ダンプを制御できるようにするダンプ・テーブル機能があります。ダンプ・テーブルによって、以下を実行できます。

- CICS で記録するダンプ (複数可) のタイプを指定する。
- ダンプを完全に抑制する。
- CICS の実行中に取得するダンプの最大数を指定する。
- ダンプが発生する障害の結果として CICS を終了させるかどうかを制御する。

## System dumps (システム・ダンプ)

CICS は、MVS SDUMP マクロを使用してシステム・ダンプを生成します。

MVS SDUMP ダンプは、CICS が MVS SDUMP マクロを発行することで生成されます。システム・ダンプには、CICS アドレス・スペースのほぼ全体、つまり、MVS 中核やその他の共通域、さらに CICS 専用ストレージ域も含まれています。SDUMP ダンプは MVS ダンプ・データ・セットに書き込まれます。これは、対話式問題管理システム (IPCS) を使用して処理でき、TSO の下でオンラインで実行することも、バッチ・ジョブを実行依頼して印刷することもできます。IPCS については、[z/OS MVS 対話式問題管理システム \(IPCS\) ユーザーズ・ガイド](#)で説明しています。CICS IPCS ダンプ出口で使用する **IPCS VERBEXIT** パラメーターについて詳しくは、[IPCS を使用した CICS ダンプのフォーマットおよび分析: 概要](#)を参照してください。SDUMP マクロと、関連する MVS ダンプ・データ・セットについて詳しくは、[z/OS MVS 診断: ツールと保守援助プログラム](#)を参照してください。

CICS が発行する SDUMP マクロには通常、**QUIESCE=NO** パラメーターが含まれます。CICS SVC コードでの異常終了、または MRO 制御ブロックの変更により SDUMP が取得される場合は、そうでないことがあります。このパラメーターにより、MVS システムは SDUMP が取得されている間はディスパッチ可能な状態を保てるため、システムへの影響を抑えることができます。ただし、**QUIESCE=YES** が MVS システム・デフォルトとして指定されている場合、CICS によって指定されたデフォルトがオーバーライドされます。これらのデフォルトは、**MVS CHNGDUMP** コマンドを使用して変更できます。

### ASRx 異常終了より先行するシステム・ダンプの抑止

MVS システム・ダンプのデータ・セットは、ASRA、ASRB、および ASRD の各種異常終了より先行する (メッセージ DFHAP0001 または DFHSR0001 の後ろ) 不要な SDUMP でいっぱいになることがあります。

CICS ストレージ保護がアクティブの場合、アプリケーション・プログラムのエラーによって発生するシステム・ダンプ (メッセージ DFHSR0001 の後ろ) を抑制しながら、CICS コードのエラーによって発生するダンプ (メッセージ DFHAP0001 の後ろ) を保持することができます。これを行うには、**CEMT SET SYSDUMPCODE** コマンド、または **EXEC CICS SET SYSDUMPCODE** コマンドを使用して、システム・ダンプ・コード SR0001 のシステム・ダンプを抑制します。以下に例を示します。

```
CEMT SET SYSDUMPCODE(SR0001) ADD NOSYSDUMP
```

アプリケーション・プログラムが、プログラム・チェック時または MVS 異常終了時にユーザー・キーで実行していた場合、CICS はダンプ・コード SR0001 を使用します。これは、ストレージ保護がアクティブの場合にのみ使用可能です。プログラムが CICS キーで実行された場合、ダンプ・コード AP0001 が代わりに使用されます。

ストレージ保護がアクティブではない場合、システム・ダンプ・コード AP0001 のシステム・ダンプを抑制することができます。ただし、これにより、アプリケーションと CICS プログラムの両方のエラーのダンプが抑制されます。XDUREQ グローバル・ユーザー出口を使用すると、アプリケーションと CICS プログラムの AP0001 状態を区別することができます。

CICS で使用できるストレージ保護機能の詳細については、[ストレージ保護](#)を参照してください。

これらのトランザクション異常終了のいずれかの SDUMP のみ取得し、それ以外のトランザクション異常終了については取得しないようにするには、**CEMT TRDUMPCODE** または **EXEC CICS TRANDUMPCODE** コマンドを使用して、取得するいずれかを選択します。これにより、ダンプ・テーブルの項目で、ASRA、ASRB、または ASRD 異常終了の SDUMP が取得されることを指定します。例えば、ダンプ・テーブルに項目を追加し、ASRB 異常終了の SDUMP が取得されるようにするには、次のように指定します。

```
CEMT SET TRDUMPCODE(ASRB) ADD SYSDUMP
```

ただし、この場合では、普段システム・ダンプ・コード AP0001 および SR0001 で SDUMP が取得されるよりも後の時点で SDUMP が取得されます。

DFHAP0001 および DFHSR0001 メッセージについては、[CICS メッセージおよびプログラム・チェックが発生する場所の検索](#)を参照してください。

### CICS トランザクション・ダンプのデータ・セット

データ・セットを磁気テープやディスクにコピーする場合、オプションでダンプ・データ・セットに対して DCB パラメーターを指定することができます。CICS はダンプ・データ・セットを開くと、MVS DEVTYPE



マクロを実行します。このマクロは、直接アクセス装置のトラック・サイズ、または磁気テープの場合は 32760 を戻します。

トランザクション・ダンプに使用される最大ブロック・サイズは、DEVTYPE マクロから返される値と 4096 のうち小さいほうの値になります。その結果、通常、ブロック・サイズは 4096 になるため (デバイスのトラック・サイズは一般にこれより大きいため)、CICS はトラック 1 つにつき複数のブロックを書き込みます。各ブロックを書き込んだ後に、MVS は現在のトラックに残されているスペース量を返します。残りのスペースが 256 バイト以上ある場合、書き込まれる次のブロックのサイズは、MVS から返された値か 4096 のうち小さいほうの値になります。

残りのスペースが 256 バイトよりも少ない場合、次のブロックは、次のトラックに書き込まれます。

トランザクション・ダンプ・データ・セットで利用できるグローバル・ユーザー出口は、次の 4 つです。

1. XDUCLESE、ダンプ・ドメインがトランザクション・ダンプ・データ・セットを閉じた後
2. XDUREQ、ダンプ・ドメインがトランザクション・ダンプを取得する前
3. XDUREQC、ダンプ・ドメインがトランザクション・ダンプを取得した後
4. XDUOUT、ダンプ・ドメインがトランザクション・ダンプにレコードを書き込む前データ・セット

### トランザクション・ダンプのデータ・セットの印刷

データ・セットが 2 つある場合は、CICS を実行しながら、一方のデータ・セットからトランザクション・ダンプを印刷できます。

### このタスクについて

トランザクション・ダンプを明示的に印刷することも、自動切り替えを使用することもできます。

### 手順

- トランザクション・ダンプを明示的に印刷するには、次のようにします。
  - a) コマンド **CEMT SET DUMP SWITCH** を使用して、データ・セットを切り替えます。

CICS は、記録中のいずれかのトランザクション・ダンプが完了すると現行のデータ・セットを閉じて、他方のデータ・セットを開きます。
  - b) DFH DU730 ダンプのユーティリティ・プログラムを使用して、完了したデータ・セットを印刷します。

DFH DU730 ダンプのユーティリティ・プログラムについて詳しくは、[ダンプ・ユーティリティ \(DFH DU00nnn および DFH PDnnn\)](#) を参照してください。
- 自動切り替えを使用してトランザクション・ダンプを印刷するには、次のようにします。
  - a) コマンド **CEMT SET DUMP AUTO** を使用して、現行のデータ・セットがいっぱいになった際に自動切り替えが行われるようにします。

このコマンドで行える切り替えは **1 つ** のみです。

トランザクション・ダンプのデータ・セットがいっぱいになると、CICS はデータ・セットを閉じて、次のようなコンソール・メッセージを発行します。

```
DFH DU0303I applid Transaction Dump Data set data set closed.
DFH DU0304I applid Transaction Dump Data set data set opened.
DFH DU0305I applid Transaction Dump Data set switched to ddname.
```
  - b) ダンプ・データ・セットに対して DISP=SHR を指定した場合、DFH DU730 ユーティリティ・プログラムで完了したデータ・セットを印刷でき、その後 **CEMT SET DUMP AUTO** コマンドを再発行します。

これにより、現行のデータ・セットがいっぱいになると、データ・セットが再び自動的に切り替えられます (1 回のみ)。



## トランザクション・ダンプ・データ・セットの定義

CICS 始動ジョブ・ストリームに、DFHDMPA および DFHDMPB という DD 名でデータ・セットを定義する必要があります。データ・セットを 1 つだけ定義する場合には、その DD 名を DFHDMPA とする必要があります。

## このタスクについて

DFHDMPA と DFHDMPB を CICS を実行するたびに一時データ・セットとして定義することも、事前にデータ・セットを割り振ってカタログしておき、繰り返し再利用することもできます。

## 手順

1. **DUMPDS** システム 初期設定パラメーターをコーディングして、CICS の初期設定中に開くトランザクション・ダンプ・データ・セットを指定します。  
DUMPDS=AUTO を指定した場合、ウォーム・スタートまたは緊急スタート時に、CICS は最後に終了したときに使用されていなかったデータ・セットを開きます。これにより、異常終了した後に、終了時に使用されていたダンプ・データ・セットの出力を待たずに CICS を再始動できます。
2. オプション: データ・セットを暗号化する場合は、102 ページの『データ・セットの暗号化』を参照してください。
3. ダンプ・データ・セットを割り振ります。これは、次の 2 つの方法のいずれかで実行できます。

- サンプルのデータ定義ステートメントを使用して、ダンプ・データ・セットをディスク上に割り振ってカタログする。
- CICS 提供のジョブ DFHDEFDS を使用して、ダンプ・データ・セットを割り振ってカタログする。

サンプルのジョブを使用する場合は、以下のようにステートメントを編集してください。

- a) インストール済み環境の要件に合わせて、このサンプル・ジョブ・ストリームのスペース割り振りを変更します。
- b) XRF を使用して CICS を実行する場合は、代替用に別のデータ・セットを割り振ります。
- c) 磁気テープを使用してダンプ出力を記録する場合は、ラベルなしの磁気テープを使用します。

標準のラベル付きのテープでは、磁気テープ・ドライブが 1 台か 2 台にかかわらず、CICS ステップが完了するまで、DFHDU730 ユーティリティによるボリュームのコンテンツ 処理が停止されます。標準のラベル付き磁気テープを使用する場合は、CICS の実行で出力されるすべての出力が、マウントされている 1 つまたは 2 つのボリュームに収まるようにしてください。

ラベルなしの磁気テープに定義されたダンプ・データ・セットはカタログできません。データ・セット定義は、CICS の実行時に必ず CICS 始動ジョブ・ストリーム内に存在している必要があります。

```
//DFHDMPA DD DSN=CICSTS56.CICS.applid.DFHDMPA,DISP=(NEW,CATLG),
//          UNIT=3380,VOL=SER=valid,SPACE=(CYL,(25))
//DFHDMPB DD DSN=CICSTS56.CICS.applid.DFHDMPB,DISP=(NEW,CATLG),
//          UNIT=3380,VOL=SER=valid,SPACE=(CYL,(25))
```

図 17. ディスク・ダンプ・データ・セットを定義するためのサンプルのジョブ制御ステートメント

4. オプション: ダンプ・データ・セットを磁気テープまたはディスクにコピーするには、ダンプ・データ・セットを割り振ってカタログするときに、次のように DCB パラメーターを DD ステートメントに指定します。

```
//          DCB=(RECFM=VB,BLKSIZE=4096,LRECL=4092)
```

5. 以下の DD ステートメントを CICS 始動ジョブ・ストリームに含めます。

- トランザクション・ダンプ・データ・セットをカタログした場合、次の DD ステートメントを追加します。

```
//DFHDMPA DD DSN=CICSTS56.CICS.applid.DFHDMPA,DISP=SHR
//DFHDMPB DD DSN=CICSTS56.CICS.applid.DFHDMPB,DISP=SHR
```

DISP=SHR を指定することによって、他のデータ・セットに切り替えられた後でも、各データ・セット (ディスクで保持されている場合) を DFH DU730 オフライン・ユーティリティで処理することができます。

- ・ トランザクション・ダンプ・データ・セットをラベルなしの磁気テープに配置した場合は、次の DD ステートメントを追加します。

```
//DFHDMPA DD DSN=CICSTS56.CICS.applid.DFHDMPA,UNIT=3400,VOL=SER=volid1,  
// DISP=(NEW,KEEP),LABEL=(,NL)  
//DFHDMPB DD DSN=CICSTS56.CICS.applid.DFHDMPB,UNIT=3400,VOL=SER=volid2,  
// DISP=(NEW,KEEP),LABEL=(,NL)
```

## タスクの結果

CICS は、初期設定時には必ず少なくとも 1 つのトランザクション・ダンプ・データ・セットをオープンしようとしています。1 つ以上のトランザクション・ダンプ・データ・セットに対する DD ステートメントを CICS ジョブに含めていないと、DFH DU0306 メッセージがコンソールに送られた後も初期設定が続けられます。

CICS はダンプ・データ・セットを開くと、MVS DEVTYPE マクロを実行します。このマクロは、直接アクセス装置の場合はトラック・サイズを返し、磁気テープの場合は 32760 を返します。トランザクション・ダンプに使用される最大ブロック・サイズは、DEVTYPE マクロから返される値と 4096 のうち小さいほうの値になります。その結果、通常、ブロック・サイズは 4096 になるため (デバイスのトラック・サイズは一般にこれより大きい)、CICS はトラック 1 つにつき複数のブロックを書き込みます。各ブロックを書き込んだ後に、MVS は現在のトラックに残されているスペース量を返します。残りのスペースが 256 バイト以上ある場合、書き込まれる次のブロックのサイズは、MVS から返された値か 4096 のうち小さいほうの値になります。

残りのスペースが 256 バイトよりも少ない場合、次のブロックは次のトラックに書き込まれます。

トランザクション・ダンプ・データ・セットで利用できるグローバル・ユーザー出口は、次の 4 つです。

1. XDUCLE、ダンプ・ドメインがトランザクション・ダンプ・データ・セットを閉じた後
2. XDUREQ、ダンプ・ドメインがトランザクション・ダンプを取得する前
3. XDUREQC、ダンプ・ドメインがトランザクション・ダンプを取得した後
4. XDUOUT、ダンプ・ドメインがトランザクション・ダンプにレコードを書き込む前データ・セット

## ユーザー・ファイルの定義

このセクションでは、ユーザー・ファイルの定義方法と、VSAM データ・セット、BDAM データ・セット、データ・テーブル、およびカップリング・ファシリティ・データ・テーブルへのアクセス方法について説明します。

### このタスクについて

CICS アプリケーション・プログラムは、CICS にとって物理データ・セットまたはデータ・テーブルの論理ビューであるファイル进行处理します。データ・テーブルの場合、このファイルによって、データ・スペース・ストレージまたはカップリング・ファシリティ構造内にあるデータ・テーブルのビューが提供されます。基礎となる物理データ・セットがオプションであるカップリング・ファシリティ・データ・テーブルの場合を除き、データ・テーブルはテーブルのロード元であるソース・データ・セットにも関連付けられています。データ・テーブル・ファイル以外の場合、このファイルによってデータ・セットのビューが提供されます。

ファイルは、最大 8 文字の**ファイル名**により、CICS に対して識別されます。同じ物理データ・セットまたはデータ・テーブルを参照する CICS に対して多数のファイルを定義することもできます。その結果、ファイルで定義されるオブジェクトのタイプに応じて、以下の効果が生じます。

- ・ データ・テーブル・ファイルの場合を除き、複数のファイルが同じデータ・セットを参照する場合、各ファイルは同じ物理データを参照することになります。
- ・ ユーザーによって保守されるデータ・テーブルの場合は、複数のファイルが同じデータ・セットを参照する場合、各ファイルは固有のデータ・テーブルのビューを表すことになります。

- CICS によって保守されるデータ・テーブルの場合、複数のファイルが同じデータ・セットを参照する場合、1つのファイルだけを CMT として定義できます。その他のファイルは、CMT ファイル定義によって作成された CMT からデータをアクセスします。
- カップリング・ファシリティ・データ・テーブルの場合は、複数のファイルが同じデータ・セットを参照する場合、各ファイルは CFDT プール内の固有のカップリング・ファシリティ・データ・テーブルのビューを表すことになります。ただし、各ファイルが同じテーブル名とプール名を指定している場合は、それぞれが同じテーブルの独立したビューを提供することになります。

データ・セットは、ディスクで保持されているデータのコレクションであり、最大 44 文字のデータ・セット名 (DSNAME) で識別されます。CICS ファイル制御では、VSAM または BDAM データのみが処理されます。CICS ファイルによって参照されるすべてのデータ・セットは、作成された後、任意の CICS ジョブによって参照される前に MVS に認識させるよう、カタログする必要があります。また、データ・セットは通常、CICS トランザクションによって使用される前に、何らかのデータがプリロードされることによって初期化されます。

CICS またはユーザーによって保守されているデータ・テーブルを使用して、VSAM データ・セットを参照するファイルを使用する CICS 領域のパフォーマンスおよび機能を向上できます。データ・テーブルは、16 MB を超えるデータ・スペース・ストレージで保持されているテーブルに含まれているデータ・レコードの構成、保守、およびデータ・レコードへの高速アクセスを取得するための方式を提供します。各データ・テーブルは、そのソース・データ・セットとして認識される VSAM KSDS と関連付けられています。データ・テーブルの詳細については、43 ページの『[複数のエクステンツと複数のボリュームを使用したデータ・セットの定義](#)』を参照してください。

カップリング・ファシリティ・データ・テーブルを使用すると、CICS ファイル制御 API を使用してシスプレックス全体でデータを共有できます。ただし、キーの長さが 16 バイト以下であるなどの制限があります。

RLS アクセス・モードを使用すると、シスプレックスの全体を通じて、CICS アプリケーションが所有する領域間で VSAM データ・セットを共有できます。詳しくは、85 ページの『[VSAM レコード・レベル共用 \(RLS\)](#)』を参照してください。

上記の各方式の詳細については、以下のトピックを参照してください。

- 83 ページの『[VSAM データ・セット](#)』
- 88 ページの『[BDAM データ・セット](#)』
- 89 ページの『[CICS へのデータ・セットの定義](#)』
- 91 ページの『[VSAM または BDAM ファイルのオープン](#)』
- 92 ページの『[VSAM / BDAM ファイルのクローズ](#)』
- 93 ページの『[CICS データ・テーブル](#)』
- 183 ページの『[カップリング・ファシリティ・データ・テーブル](#)』。

## VSAM データ・セット

VSAM データ・セットを作成するには、Access Methods Services (AMS) ユーティリティ・プログラム IDCAMS をバッチ・ジョブで実行するか、TSO セッションで TSO DEFINE コマンドを使用します。

## このタスクについて

DEFINE コマンドは、VSAM および MVS に、データ・セットの VSAM 属性と特性を指定します。またこのコマンドを使用して、データ・セットを定義するカタログを識別できます。

必要に応じて、再度 IDCAMS を使用してデータ・セットにデータをロードすることもできます。既存のデータ・セットから新しく作成したデータ・セットにデータをコピーするには、AMS REPRO コマンドを使用します。

空の VSAM データ・セットを CICS トランザクションからロードすることもできます。これを行うには、データ・セットを CICS に定義し (データ・セットを CICS ファイルに割り当てることで)、その後、空の状態に関わらずデータ・セットにデータを書き込みます。84 ページの『[空の VSAM データ・セットのロード](#)』を参照してください。

データ・セットを作成する際に、最大で 44 文字のデータ・セット名を定義できます。名前を定義しないことを選択した場合は、VSAM によって名前が割り当てられます。データ・セット名 (または DSNAME) と呼ばれるこの名前は、データ・セットを MVS システム内で一意に識別する名前です。

ユーザー・ファイルによってアクセスされる VSAM データ・セットを CICS ファイル制御の下で定義して、CICS がこれらのデータ・セットを更新している最中にバックアップできるようにすることができます。更新のために開いている VSAM ファイルをバックアップする方法については、[46 ページの『VSAM ファイルのオープン中のバックアップ \(BWO\) の定義』](#)を参照してください。

### VSAM のベースとパス

データをレコード・レベルで参照するアプリケーション・プログラムを使用して、データ・セット内にデータを保管したり、データ・セットからデータを取得したりします。

データ・セットのタイプに応じて、キー (レコード内の事前定義済みフィールド内の固有値)、相対バイト・アドレス、または相対レコード番号で取得対象のレコードを識別できます。

これらの 1 次識別の方式によりレコードにアクセスすることを、ベースによるアクセスと呼びます。

2 次キーまたは代替キーでレコードを識別してアクセスする必要がある場合もあります。VSAM の使用時には、単一のベース・データ・セットに対して 1 つ以上の代替索引を作成できるので、同じ情報の複数のコピーを保持し、さまざまなアプリケーション用にさまざまな方法で編成する必要はありません。この方式の使用時には、**代替索引パス** (またはパス) を作成します。このパスは、代替索引 (または索引) とベースをリンクします。作成後、代替キーを使用し、アクセス対象のデータ・セットとしてパスを指定してレコードにアクセスできます。そのためには、パス情報を含んだデータ・セットを作成し、そのデータ・セットを CICS ファイルに割り振ります。

パスの作成時に、ベース・データ・セットと同じ方法で最大 44 文字の名前を付けます。CICS アプリケーション・プログラムは、パスとベースのどちらを使用してデータにアクセスするかを認識する必要はありません。ただし、プログラムがパスを介して (つまり、代替キーや索引を使用して) データにアクセスしている場合、非固有キーを使用するように代替索引を指定したときに、プログラムが重複キーを使用可能にしなければならないことがあります。

### 空の VSAM データ・セットのロード

空の VSAM データ・セットにデータをロードするには、2 つの方法があります。

- AMS ユーティリティー・プログラム IDCAMS を実行する
- CICS トランザクションを使用してデータ・セットにレコードを書き込む

最初のデータ・セット・ロード処理時には VSAM で課される制限がありますが、データ・セットがロード・モードに設定されていると、それらの制限は CICS トランザクションに影響しません。非 RLS モードで開かれたファイルの場合、CICS ファイル制御によって、ロード・モード処理がアプリケーション・プログラムから「隠蔽」されます。空のデータ・セットに対して RLS モードで開かれたファイルの場合、ロード・モード処理は VSAM によって CICS から隠蔽され、すべての VSAM 要求が許可されます。

### IDCAMS を使用して空の VSAM データ・セットにロードする

大量のデータを新しいデータ・セットにロードする場合は、バッチ・ジョブとして AMS ユーティリティー・プログラム IDCAMS を実行し、REPRO コマンドを使用してデータを既存のデータ・セットから空のデータ・セットにコピーします。

IDCAMS を使用してデータ・セットをロードし終えたら、CICS でそのデータ・セットを通常の方法で使用できます。

データ・セットが VSAM ロード・モードの場合は、アップグレード・セット内で代替索引を使用できません。代替索引のあるデータ・セットを作成してロードする場合は、AMS かその他の適切なバッチ・プログラムを使用してデータ・セットをロードし、BLDINDEX を呼び出して代替索引を作成しなければなりません。

### CICS アプリケーションを使用して空の VSAM データ・セットにロードする

ロードするデータが少量で、アップグレード・セットがない場合は、標準的な CICS ファイル WRITE 要求を使用して、空のデータ・セットにロードすることができます。



ファイルへの最初の書き込みや一連の書き込み (一括挿入) が完了すると、CICS はそのファイルを閉じ、閉じたまま使用可能な状態で保持するので、次の参照時に通常処理のために再度開くことができます。ロード・モードでファイルを読み取ろうとすると、CICS から NOTFOUND 条件が返されます。

#### **concurrency(required) プログラムを使用して空の VSAM データ・セットにロードする際の考慮事項**

concurrency(required) プログラムを使用して空の VSAM データ・セットにロードする場合は、プログラムが一連の WRITE MASSINSERT 要求を使用すると、オープン TCB から QR TCB に切り替えて再び元に戻す TCB 切り替えが過度に発生することに注意してください。

##### **TCB 切り替えが発生する理由**

ロード・モードでファイルを開くには、CICS はローカル共用リソース (LSR) を使用できないため、ファイルを非共用リソース (NSR) モードで開く必要があります。NSR では、オープン TCB ではなく QR TCB で処理が行われる必要があります。

CONCURRENCY(REQUIRED) として定義されているプログラムは常にオープン TCB で実行されます。concurrency(required) プログラムからファイルがロードされる場合は、ファイルがロード・モードである間、オープン TCB から QR TCB に切り替えて再び元に戻す TCB 切り替えが書き込み操作ごとに発生します。

##### **TCB 切り替えを回避する方法**

一連の WRITE MASSINSERT 要求を使用してファイルをロードすることはお勧めできません。一連の WRITE MASSINSERT 要求が完了するまでファイルはロード・モードのままになるためです。一括挿入なしの単一の WRITE を使用する必要があります。

一括挿入なしの単一の WRITE が完了すると、CICS はファイルを閉じて開き直すので、VSAM ロード・モードは終了します。ファイルがロード・モードでないときに LSR モードで正常にアクセスされると、CICS は LSR モードでファイルへのアクセスを再開します。その後ファイルに対してさらに書き込み操作が行われても、TCB 切り替えは回避されます。

#### **データ・セットの再利用**

AMS REUSE 属性を使用してデータ・セットを定義すると、CICS の実行中にデータを空にすることもできます。

空になったものを、作業ファイルとして使用できます。データ・セットを参照しているファイルの状況が CLOSED と DISABLED (または UNENABLED) の場合、SET EMPTY コマンドを使用できます。このコマンドは、EXEC CICS コマンド・レベル・インターフェースを使用してアプリケーション・プログラムから実行するか、マスター端末 CEMT コマンドを使用してマスター端末から実行できます。このコマンドは、インストール済みのファイル定義内で標識を設定し、次回ファイルを開く際に、VSAM 高使用相対バイト・アドレス (RBA) がゼロに設定され、データ・セットのコンテンツが効率的に消去されるようにします。

平均レコード長と最大レコード長を同じ値にしてデータ・セットを VSAM に定義し、固定長レコードを指定してファイルを CICS に定義してそのデータ・セットを参照する場合、そのデータ・セットに書き込むレコードのサイズは、定義したサイズでなければなりません。例えば、データ・セット内のレコードを更新のために読み取る際に、以下のようなことを行くと、レコードの再書き込み時にエラーを受け取ります。

- パラメーター RECORDSIZE(250 250) を指定して、レコード・サイズを 250 バイトとして VSAM に定義した
- パラメーター RECFORM=FIXED を使用してファイルを CICS に定義した
- 長さが 200 バイトしかないレコードをデータ・セットにロードした

#### **VSAM レコード・レベル共用 (RLS)**

レコード・レベル共有 (RLS) は、DFSMS 1.3 以降のリリースでサポートされる VSAM データ・セットのアクセス・モードです。RLS により、多数の CICS 領域で実行される多くのアプリケーション間で、完全な更新機能を使用して VSAM データを共用することができます。

RLS では、VSAM データ・セットを共有する CICS 領域は、MVS 並列シスプレックス内の 1 つ以上の MVS イメージに常駐可能です。CICSplex をサポートする VSAM RLS を使用する並列シスプレックスにおけるこの概念については、[86 ページの図 18](#) で説明されています。

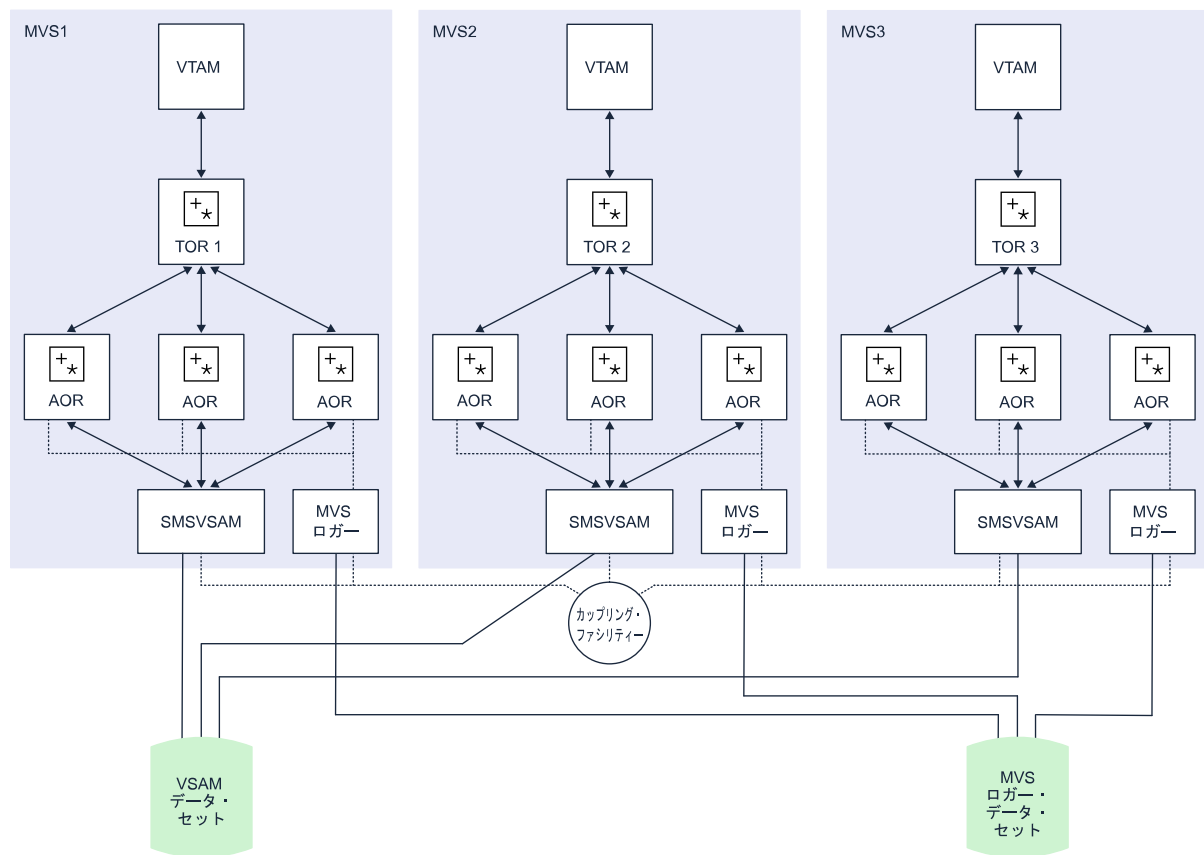


図 18. RLS を使用する Parallel Sysplex の図

RLS サポートがない場合 (RLS=NO システム 初期設定パラメーター)、非 RLS モード (LSR または NSR など) を使用して複数の CICS 領域が同じ VSAM データ・セットを同時に開くことはできません。これらのアクセス・モードは、CICS 領域間で VSAM データを共有するには、次のことを行う必要があることを意味します。

- 共有データ・テーブルを使用する。

または

- MRO、APPC、または IPIC 接続を使用して、1 つの CICS 領域、ファイル所有領域 (FOR)、およびアプリケーションから FOR への機能シップのファイル要求に、VSAM データ・セットを割り振る。

RLS のサポートがある場合、複数の CICS 領域が同じデータ・セットを並行して開くことができます。RLS を使用するには、次のようにします。

- RLS をサポートするレベルの DFSMS と、CICS システム 初期設定パラメーターとして RLS=YES が指定されていることが必要です
- CICS 領域はすべて、同じ Parallel Sysplex 内で実行されている必要があります
- 各 MVS イメージ内で 1 つの SMSVSAM サーバーが開始されている必要があります
- CICS ファイル・リソース定義で RLSACCESS(YES) を指定することで、複数の CICS 領域からアクセスされるデータ・セットの完全なアップデート機能を使用できるようにします。

CICS ファイル制御でサポートされているすべてのファイルに RLS アクセス権限を指定できます。ただし、以下の例外があります。

- キー範囲のデータ・セットはサポートされていません。
- IMBED 属性が定義された VSAM クラスタはサポートされていません。ただし、IMBED 属性をクラスタ定義から削除しても、機能は失われません。アクセス方式サービス・プログラム REPRO 機能を使用して、IMBED 属性が定義されていない新しいクラスタにデータを移動します。これで、新しいクラスタを参照するファイルで RLS アクセス・モードを使用できます。(IMBED は、最近のキャッシュ・ディスク・コントローラーでは通常は必要ないパフォーマンス・オプションです。)
- VSAM クラスタのコンポーネントを個別に開くことは、CICS ではどのモードのアクセスでもサポートされていません。
- 一時データ・セットはサポートされていません。
- 相対バイト・アドレス (RBA) モードのキー・シーケンス・データ・セット (KSDS) (OPTCDE=ADR) はサポートされていません。KSDS が開いた RLS モードのファイル制御 API コマンドに RBA キーワードを指定するアプリケーション・プログラムは、RESP2 51 例外条件の INVREQ を受け取ります。
- 代替索引データを直接開くことは、RLS アクセス・モードではサポートされていません。ただし、データへのパスを使ったアクセスはサポートされています。
- VSAM カタログおよび VVDS データ・セットはサポートされていません。

入力順データ・セット (ESDS) で RLS アクセスを指定することはできますが、レコードの追加時にパフォーマンスおよび可用性を低下させる可能性があるため、お勧めできません。詳しくは、[CICS VSAM およびファイル制御: パフォーマンスおよび調整](#)を参照してください。

VSAM RLS のサポートをセットアップするために必要な手順について詳しくは、[VSAM RLS サポートの設定](#)を参照してください。

#### VSAM データ・セットに関する混合モード操作

一般的に、複数の CICS 領域でどのデータ・セットを RLS モードで共用し、更新する必要があるかを選択します。選択し終わったら、選択したデータ・セットを常に RLS モードで更新することをお勧めします。

しかし、RLS サポートの使用時には、CICS 領域とバッチ・ジョブの間で、データ・セットを混合アクセス・モードで共用できます。混合アクセス・モードとは、複数のユーザーが 1 つのデータ・セットを同時に RLS モードと非 RLS モードで開くことを指します。

データ・セットはその都度異なるモードで開くことができますが、VSAM スフィア内のすべてのデータ・セットは、通常同じモードで開く必要があります。(スフィアは、ある VSAM 基本データ・セットに関連したすべてのコンポーネントの集合のことです - ベース、索引、代替索引、および代替索引パス) ただし VSAM では、CICS の制限を前提として、異なるアプリケーションによるスフィアでの混合モード操作が許可され



ています。以下の混合モード操作の説明で、データ・セットを参照している場合は、スフィアのコンポーネントを参照しています。

### 混合モードの SMSVSAM 操作

SMSVSAM は、シスプレックス内の複数のアプリケーションが 1 つのデータ・セットを複数のモードで同時に開くことができるようにします。

共用については、以下の規則と制限があります。

- 複数の CICS 領域に対して RLS モードで開かれるデータ・セットは、**読み取り専用**操作用に非 RLS モードでも開くことができます。
- 非 RLS アプリケーションの場合、読み取り安全性は保証されません。
- 同時に RLS モードと非 RLS モードで開かれるデータ・セットは、領域間 SHAREOPTIONS(2) を指定して定義しなければなりません。

### CICS の制約事項

ファイルを CICS 領域内で RLS モードまたは非 RLS モードで開くことができるのは、別のユーザー (CICS 領域かバッチ・ジョブ) が、参照されているデータ・セットを別のモードで既に関いている場合です。

しかし、上記の VSAM 規則に加えて、同じ CICS 領域内でデータ・セットを複数のモードで同時に開くことはできないという規則があります。この規則により、CICS では CICS 領域内のデータの整合性のある表示を維持できます。

この CICS の制約事項は以下のように運用されます。

- CICS 領域内でデータ・セットが RLS モードで開かれている場合は、同じ CICS 領域内で別のファイルを介してそのデータ・セットを非 RLS モードで開くことはできません。

CICS でデータ・セットが既に RLS モードで開かれている場合、メッセージ DFHFC0512 で非 RLS モードのファイル・オープン要求は失敗します。

- CICS 領域内でデータ・セットが非 RLS モードで開かれている場合は、同じ CICS 領域内で別のファイルを介してそのデータ・セットを RLS モードで開くことはできません。

CICS でデータ・セットが非 RLS モードで既に関かれていると、RLS モードのファイル・オープン要求はメッセージ DFHFC0511 を発行して失敗します。

- CICS 領域内にデータ・セットに関する未解決のリカバリー処理がある場合は、同じ CICS 領域内で別のファイルを介してそのデータ・セットを非 RLS モードで開くことはできません。

CICS 内にデータ・セットに関する未処理のリカバリー処理があると、非 RLS モードのファイル・オープン要求はメッセージ DFHFC0513 を発行して失敗します。

### BDAM データ・セット

CICS では、キー付き、およびキーなしの BDAM データ・セットへのアクセスをサポートしています。そのようなデータ・セットを作成して形式を設定する場合は、BDAM を使用します。

### このタスクについて

BDAM データ・セットを CICS の実行で使用するには、あらかじめデータを組み込んでおく必要があります。データ・セットのロードには、レコードを順次書き込んでいくバッチ・プログラムを使用します。その一例が 88 ページの図 19 です。

```
//BDAM EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=CICSTS56.bdam.user.file.init,DISP=SHR
//SYSUT2 DD DSN=CICSTS56.bdam.user.file,DISP=(,CATLG),
// SPACE=(TRK,(1,1)),UNIT=3380,VOL=SER=valid,
// DCB=(RECFM=F,LRECL=80,BLKSIZE=80,DSORG=DA)
```

図 19. BDAM データ・セットを作成してロードするためのサンプル JCL

注:

1. 入力データ・セット (この例では SYSUT1) は、物理的に順番に並んでいる順次データ・セットでなければならず、入力データ・セットの属性は、出力データ・セット (この例では SYSUT2) の DCB との間に互換性のある属性でなければなりません (注 3 を参照)。特に以下の点に注意してください。
  - 出力データ・セットで RECFM=F を指定する場合は、入力データ・セットで RECFM=F か RECFM=FB を指定する必要があります。また、両方のデータ・セットで LRECL に同じ値を指定しなければなりません。
  - 出力データ・セットで RECFM=V か RECFM=U を指定する場合は、入力データ・セットの LRECL の値を出力データ・セットの指定値以下にしなければなりません。
2. データ・セットを作成する時に、44 文字以下のデータ・セット名 (DSNAME) を定義します。MVS システムでデータ・セットを一意的に特定するためのデータ・セット名です。
3. 出力データ・セットの DCB パラメーターで以下の値を指定する必要があります。
  - DSORG=DA。データ・セットを BDAM データ・セットとして指定します。
  - BLKSIZE。関連するファイル管理テーブル (FCT) 項目の BLKSIZE の指定値と同じ値でなければなりません。
  - RECFM。F (固定)、V (可変)、U (未定義) のいずれかの値にします。関連する FCT 項目の RECFORM オペランドの最初のサブパラメーターに対応していなければなりません。

これらのオプションを DFHFCT TYPE=FILE 定義で指定します。DFHFCT TYPE=FILE のオプションを使用してファイルを定義する方法については、[File control table \(FCT\)](#)を参照してください。この例で作成するデータ・セットに [89 ページの図 20](#) のデータをロードする場合は、FCT 項目で以下の属性を指定します。

- BLKSIZE=80
- LRECL=40
- RECFORM=(FIXED BLOCKED)
- KEYLEN=8

```
RECORD 1 DATA FOR RECORD 1      RECORD 2 DATA FOR RECORD 2
RECORD 3 DATA FOR RECORD 3      RECORD 4 DATA FOR RECORD 4

RECORD98 DATA FOR RECORD 98     RECORD99 DATA FOR RECORD 99
1-----2-----3-----4-----5-----6-----7-----8
```

図 20. BDAM データ・セットをロードするためのサンプル・データ

### CICS へのデータ・セットの定義

CICS でデータ・セットを参照しているファイルを開くには、そのファイルのファイル定義が事前にインストールされている必要があります。この定義は、CSD またはファイル管理テーブル (FCT) からインストールすることができます。

CSD では VSAM ファイルのみ、FCT では BDAM ファイルのみ定義できます。FCT にある BDAM 以外の定義はインストールされませんが、FCT 内に残される場合があります。

ファイルは、そのファイルの定義時に指定したファイル名で識別されます。アプリケーション・プログラム、または CEMT コマンドを使用するマスター端末オペレーターが関連データ・セットを参照するときに、CICS でこの名前が使用されます。

また、各ファイルは、以下のいずれかの方法でそのデータ・セットと関連付けられている必要があります。

- 始動ジョブ・ストリームで JCL を使用する
- FILE リソース定義の DSNAME パラメーターと DISP パラメーターを使用する
- CEMT で動的割り振りを使用する
- アプリケーション・プログラムで動的割り振りを使用する

CICS アドレス・スペースに割り振ることができる VSAM ファイルの数は約 10000 個です。ただし、VSAM は開かれたファイルごとにテーブル項目を維持し、テーブル・スペースでは開かれるファイルの数が約 8189 個に制限されます。

注：CICS は、CICS によって最初に割り振られたファイルしか割り振り解除しません。サード・パーティー・アプリケーションによってファイルが割り振られた場合、CEMT CLOSE コマンドの使用によりファイルは閉じられますが、割り振り解除は行われません。

## JCL の使用

CICS 始動ジョブの JCL で DD ステートメント内にデータ・セットを定義できます。DD 名は、データ・セットを参照するファイル名と同じ名前であればなりません。

例えば、次の DD ステートメントは、ファイル名 VSAM1A および BDAMFILE のファイル定義に対応します。

```
//VSAM1A DD DSN=CICSTS56.CICS.vsam.user.file,DISP=OLD  
//BDAMFILE DD DSN=CICSTS56.CICS.bdam.user.file,DISP=SHR
```

この方法でデータ・セットを CICS に定義した場合、データ・セットは CICS の起動時に MVS により CICS に割り振られ、通常は、CICS の実行が終了するまで割り振られたままになります。また、物理データ・セットが、CICS の実行を通じてインストール済みファイル定義に関連付けられています。

JCL を使用してユーザー・データ・セットを CICS システムに定義する場合、DD ステートメントに FREE=CLOSE オペランドを含めることはできません。

DD ステートメントで RLS=CR オプションまたは RLS=NRI オプションを使用すると、そのオプションは無視されます。ファイルのアクセス・モード (RLS または非 RLS) およびすべての読み取り保全性オプションは、ファイル定義に指定する必要があります。

## DSNAME と DISP のファイル・リソース定義パラメーターの使用

ファイルの定義時に **DSNAME** パラメーターと **DISP** パラメーターを指定して、CICS にデータ・セットを定義できます。

これらのパラメーターを指定するには、ファイルに関する RDO を使用するか、DFHFCT マクロを使用します (BDAM ファイルのみ)。ファイル定義から DSNAME と DISP を使用する場合は、始動ジョブ・ストリーム内でデータ・セットに関する DD ステートメントを使用しないでください。その理由は、DD ステートメント内の属性により CICS リソース定義内の属性がオーバーライドされるからです。

ファイル定義上の DSNAME と DISP を使用する場合は、CICS はデータ・セットを、そのデータ・セットを参照する最初のファイが開かれる時点、つまりそのファイルが開かれる直前に動的に割り振ります。この段階で、CICS はファイル名とデータ・セットを関連付けます。

以後 CICS アプリケーションがそのデータ・セットを参照する際には、ファイル名を指定して参照します。この方法でデータ・セットを定義すると、ファイルを閉じる際に CICS によって自動的に割り振り解除されます。

**DSNAME** パラメーターと **DISP** パラメーターの使用法については、[システム・リカバリー・テーブル \(SRT\)](#) と [File control table \(FCT\)](#) を参照してください。

## CEMT を使用する動的割り振り

マスター端末 CEMT コマンドを使用して、インストール済みのファイル定義内でデータ・セット名を動的に設定できます。

```
CEMT SET FILE(filename) DSNAME(datasetname) SHARE|OLD
```

このコマンドを使用すると、CICS は OPEN 処理の一部として上記のようにデータ・セットを割り振ります。データ・セットに関連付けられている最後のファイル・エントリーを閉じると、そのデータ・セットは自動的に割り振り解除されます。CEMT コマンドを使用してファイルを動的に割り振る前に、ファイル状況を CLOSED かつ DISABLED か UNENABLED にしなければなりません。

この方式でデータ・セットを CICS に定義すると、ファイル定義をさまざまなデータ・セットにさまざまな時点で関連付けることができます。代わりに、ファイルを閉じ、データ・セットを割り振り解除してから、

再割り振りして、別の DISP 設定を使用して同じファイルを開くこともできます。例えば、この方式を使用して、物理データ・セットを、そのデータ・セットを読み取るバッチ・プログラムと共用できるようにすることもできます。

CEMT SET コマンドについて詳しくは、[CEMT SET コマンド](#) を参照してください。

### 他の形式の動的割り振り

前述のセクションで説明されている、CICS ファイル制御の一部である動的割り振り方式だけを使用することをお勧めします。

CICS 動的割り振りトランザクション ADYN を使用しないでください。ADYN は、VSAM と BDAM のユーザー・ファイルを動的に割り振るためにサンプル CICS ユーティリティー・プログラム DFH99 を呼び出します。ADYN トランザクションを使用すると、CICS ファイル制御内で使用される動的割り振り方式と競合する場合があります、予測不能な結果になることがあります。

ADYN トランザクションの使用対象は、補助トレースや CICS トランザクション・ダンプ・データ・セットなどの、CICS ファイル制御の管理下でないデータ・セットだけに制限してください。

### VSAM または BDAM ファイルのオープン

アプリケーション・プログラムでファイルにアクセスするためには、その前に、CICS が、プログラムで参照されているインストール済みファイル定義を使用してそのファイルを開いておく必要があります。

### このタスクについて

ファイルを開く処理の一環として、データ・セットに対する後続の処理に必要な制御ブロックおよびバッファを確保する必要があります。VSAM ローカル共用リソース (LSR) を使用するファイルを定義した場合、それらの制御ブロックおよびバッファはリソースのプールから割り振られます。ファイルを開くときに LSR プールが存在しない場合、CICS は所要量を計算し、ファイルを開く前にプールを作成します。非共用リソースを使用するファイルを定義した場合、必要な制御ブロックおよびバッファは OPEN 処理の一環として VSAM によって割り振られます。RLS アクセス・モードで開くようにファイルを定義した場合、制御ブロックおよびバッファは VSAM によって SMSVSAM アドレス・スペースおよび関連するデータ・セットに割り振られます。RLS モードでは、CF キャッシュ構造も使用されます。データ・セットを参照する最初のファイルが開かれたときに、この構造にデータ・セットがバインドされます。

単一の VSAM データ・セットに、ベースからアクセスしたり、さまざまなアクセス要求のための 1 つ以上のパスからアクセスしたりする場合があります。このような場合、CICS はアクセス経路ごとに別のファイル定義 (つまり、別のファイル) を使用します。各ファイル定義に、対応するデータ・セット名を関連付ける必要があります (パスにもデータ・セット名を割り当てます)。インストールされたファイル定義の属性を使用して CICS がファイルにアクセスするには、その前に各ファイルが開かれている必要があります。これは、異なる属性を持つ 2 つ以上のファイルとして論理的に定義されたデータ・セットの場合、そのデータ・セットのファイルを 1 つ開いても、すべてのアクセス経路でデータ・セットを使用できるわけではないからです。

CICS では、複数のファイル定義を同じ物理データ・セット名に関連付けることができます。例えば、同じデータ・セットを参照するファイルを、さまざまな処理属性を使用して定義しなければならない場合があります。

CICS は、ファイルの状態が ENABLED であるかどうかに応じて、ファイル内のデータへのアクセスを許可または拒否します。閉じられている **ENABLED** のファイルは、最初のアクセス要求が実行されたときに、CICS によって自動的に開かれます。明示的な CLOSE 要求か、CICS ジョブの終わりまで、そのファイルは開かれたままです。

次のいずれかのコマンドを使用して、ファイルを明示的に開くこともできます。

```
CEMT SET FILE(filename) OPEN
EXEC CICS SET FILE(filename) OPEN
```

これらのコマンドのいずれかを使用すると、状態が ENABLED または DISABLED のどちらであるかに関係なくファイルが開かれます。この方法を選択すると、ファイルにアクセスする最初のトランザクションで、ファイルを開くためのオーバーヘッドが生じることを防止できます。

RDO OPENTIME(STARTUP) 属性 (または DFHFCT マクロの FILSTAT=OPENED パラメーター) を指定して、初期設定の直後に CICS でファイルを開くように指定することもできます。始動後に CICS でファイルを開



くように指定した場合、ファイル状況が **ENABLED** または **DISABLED** であれば、CICS ファイル・ユーティリティー・トランザクション CSFU がそのファイルを開きます (CSFU は、**UNENABLED** として定義されているファイルを開きません。これらのファイルの状況は **CLOSED** または **UNENABLED** のままです)。CSFU は、CICS の初期設定が完了する直前に自動的に開始されます。CICS は、1 つの **OPEN** 要求で 1 つのファイルを開きます。CSFU の実行中にユーザー・トランザクションが開始された場合、CSFU がまだ開いていないファイルをユーザー・トランザクションで参照したり開いたりできます。CSFU が完了するのを待機する必要はありません。

### バッチ・ジョブからのファイルのオープンとクローズ

いくつかの方法で、z/OS バッチ・ジョブから CICS ファイルをオープンまたはクローズすることができます。

#### このタスクについて

- 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) を使用できます。この API で CICS システム 管理情報にアクセスできるので、外部プログラムから CICSplex SM サービスを呼び出すことができます。この API は、z/OS Batch で実行されているプログラムから呼び出すことができます。[Developing CICSplex SM applications](#) および [CICSplex SM API コマンド](#) を参照してください。
- Liberty batch-1.0 機能によって提供される Java Batch 1.0 API サポートを使用して、新しい Java バッチ・アプリケーションを実行できます。Java Batch 1.0 は、バッチ・アプリケーション用の API、およびバッチ・ジョブを実行および管理するためのランタイムを提供します。[Liberty フィーチャー](#) を参照してください。
- 外部 CICS インターフェース (EXCI) を使用することができます。MVS でプログラムを実行すると、EXEC CICS LINK PROGRAM コマンドを発行するか CALL インターフェースを使用して、CICS 領域で実行されている CICS アプリケーション・プログラムを呼び出すことができます。MVS プログラムは MVS ジョブの開始時と終了時に CICS リソースをオフラインにし、オンラインに戻すことができます (例えば CICS ファイルをオープンしてからクローズする場合など)。[EXCI の概念](#) および [EXCI サンプル・プログラム](#) を参照してください。

### VSAM / BDAM ファイルのクローズ

ファイルを閉じるには、**CLOSE** コマンドを使用します。FORCE オプションを付ける場合もあれば、付けない場合もあります。

#### このタスクについて

##### ファイルの通常のクローズ

**SET FILE(filename) CLOSED** コマンドを使用して、ファイルを明示的に閉じることができます。

要求時にファイルを使用しているトランザクションがないと、そのファイルは即時に閉じられます。クローズ操作の一部として、ファイルは使用不可にもされます。この使用不可化の形式は CEMT ディスプレイで **UNENABLED** として表示されます。この場合、以降の要求でファイルにアクセスして暗黙に再び開くことはできません。

VSAM 要求か BDAM 要求の実行中のトランザクションや、一連の接続された要求の実行中のトランザクションは、ファイルのユーザーと呼ばれます。例えば、以下の要求の実行中には、トランザクションはユーザーになります。

```
READ UPDATE ---- REWRITE
STARTBROWSE ---- READNEXT ... ---- ENDBROWSE
```

ファイルに対するリカバリー可能な変更を完了したものの、同期点やトランザクションの終わりにまだ達していないトランザクションも、ファイルのユーザーになります。

クローズ要求の時点でユーザーが存在する場合、ファイルは即時に閉じられません。CICS は、すべての現行ユーザーがファイルの使用を完了するまで待ちます。ファイルは **UNENABLING** 状態になり、新しいユーザーへのアクセスは拒否されますが、既存のユーザーがファイルの使用を完了することはできます。最後のユーザーがファイルを使用し終わると、ファイルは閉じられて **UNENABLED** になります。トランザクションがファイルに対してリカバリー可能な変更を加えてから、同期点中に障害が発生した場合、作業単位は中断され、その時点でファイルを閉じることができます。

## FORCE オプションを使用するファイルのクローズ

**SET FILE(filename)** コマンドで FORCE オプションを使用して、ファイルを閉じることができます。

ファイルの現行ユーザーであるトランザクションは異常終了し、必要に応じて変更をバックアウトできます。その後、ファイルは閉じられて UNENABLED になります。CLOSE 要求の結果として UNENABLED になったファイルは、その後明示的な OPEN 要求が行われる場合に再び使用可能にすることができます。

FORCE オプションを使用してファイルを閉じると、CICS タスクの FORCEPURGE メカニズムによって、ファイルの現行ユーザーのタスクは即時に終了します。このメカニズムでは、データ保水性は保証されません。極端な場合 (バックアウト処理中にエラーが発生するなど)、CICS は異常終了することがあります。この理由から、FORCE オプションを使用したファイルのクローズは例外的な状況だけに制限する必要があります。

## CICS データ・テーブル

**CEDA DEFINE FILE** コマンドを使用してデータ・テーブルを定義できます。

### このタスクについて

テーブルを開くと、CICS はそのテーブルを構築するために、対応するソース VSAM データ・セットからデータを抽出し、CICS データ・テーブル・サーバー領域の所有する MVS データ・スペースにロードして、16 MB 境界より上の CICS 仮想ストレージでインデックスを作成します。そのようなテーブルにアクセスする時に使用するコマンドは、CICS アプリケーション・プログラミング・インターフェース (API) のファイル管理コマンドです。

アプリケーション・プログラミング・インターフェースのファイル管理コマンドのプログラミング情報については、[CICS コマンド・サマリー](#)を参照してください。CICS は、次の2つのタイプのデータ・テーブルに対応しています。

- **CICS 保守データ・テーブル**。CICS がソース・データ・セットとの同期処理を継続的に実行します。
- **ユーザー保守データ・テーブル**。ロード後にソース・データ・セットから切り離されます。

どちらのタイプでも、グローバル・ユーザー出口を使用して、ソース・データ・セットのどのレコードをデータ・テーブルに組み込むかを選択できます。

グローバル・ユーザー出口のプログラミング・インターフェース 情報については、[グローバル・ユーザー出口プログラム](#)を参照してください。CICS データ・テーブルの詳細については、[共用データ・テーブルの概要](#)を参照してください。

### データ・テーブルのオープン

データ・テーブルのエントリーがアプリケーションによってアクセスされる 前に、そのデータ・テーブルを開かなければなりません。

データ・テーブルは、OPEN 要求を使用して明示的に開くか、最初の参照時に暗黙的に開くか、またはファイル定義内に OPENTIME(STARTUP) を指定した場合は始動直後に CSFU タスクにより開くことができます。データ・テーブルが開かれると、CICS は完全なソース・データ・セットを読み取り、レコードをデータ・スペース内にコピーして索引を作成します。

データ・テーブル内にコピーされたレコードごとにグローバル・ユーザー出口を呼び出すことができます。このコピー操作は、ユーザー作成出口の選択基準に従います。

データ・テーブルを開くのに使用するコマンドや、暗黙的に開いたり即時に開いたりすることに関係する規則やオプションは、[91 ページの『VSAM または BDAM ファイルのオープン』](#)で説明されているものと同じです。

### データ・テーブルのロード

データ・テーブルは、開く際に自動的に作成されます。

索引は、レコードに高速アクセスできるように構成されます。[を参照してください](#)。詳細については、[共用データ・テーブルの概要](#)を参照してください。

ユーザー保守データ・テーブルの場合、ソース・データ・セットの ACB はロード完了時に閉じられます。データ・セットは、最初に動的に割り振られており、他の ACB がまだそのデータ・セット用に開かれていない場合には割り振り解除されます。

### データ・テーブルのクローズ

CLOSE コマンドを使用してデータ・テーブルを閉じることができます。FORCE オプションは指定することもしないこともできます。

データ・テーブルを閉じると、レコードの保持に使用されたデータ・スペース・ストレージと、関連付けられている索引に使用されたアドレス・スペース・ストレージは、CLOSE 操作の一部として解放されます。

データ・テーブルを閉じるのに使用するコマンドや、データ・テーブルの現行ユーザーに関する規則は、[92 ページの『VSAM / BDAM ファイルのクローズ』](#)で説明されているものと同じです。

## CDBM GROUP コマンド・データ・セットの定義

CDBM GROUP コマンド・データ・セット DFHDBFK は、VSAM キー順データ・セット (KSDS) です。CDBM トランザクションはこのデータ・セットを使用して、DBCTL コマンドの保管されたグループのリポジトリを提供します。

### 手順

1. IDCAMS ジョブを実行して、DFHDBFK データ・セットを作成します。[95 ページの図 21](#) にその例が示されています。



```

//DBFKJOB JOB 'accounting information',name,MSGCLASS=A
//*
//DBFKDEF EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//AMSDUMP DD SYSOUT=*
//SYSIN DD *
DELETE CICSTS56.CICS.DFHDBFK
SET MAXCC=0
DEFINE CLUSTER (
    NAME( CICSTS56.CICS.DFHDBFK ) -
    INDEXED -
    RECORDS(100 20) -
    KEYS(22,0) -
    RECORDSIZE(1428 1428) -
) -
INDEX (
    NAME( CICSTS56.CICS.DFHDBFK.INDEX ) -
    CONTROLINTERVALSIZE(512) -
) -
DATA (
    NAME( CICSTS56.CICS.DFHDBFK.DATA ) -
    CONTROLINTERVALSIZE(2048) -
)

/*
/* The next two job steps are optional.
/*
//DBFKINID EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE CICSTS56.CICS.DBFKINIT
/*
//DBFKINIF EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSN=CICSTS56.CICS.DBFKINIT,DISP=(NEW,CATLG),
// UNIT=dbfkunit,VOL=SER=dbfkvol,SPACE=(TRK,(1,1)),
// DCB=(RECFM=FB,LRECL=40,BLKSIZE=6160)

```

```

/* Place the definitions you want to load after SYSUT1. For example:
//SYSUT1 DD *
SAMPLE DIS DB DI21PART
SAMPLE STA DB DI21PART
SAMPLE STO DB DI21PART
/*
//SYSIN DD *
GENERATE MAXFLDS=1
RECORD FIELD=(40)
/*
//DBFKLOAD EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//AMSDUMP DD SYSOUT=*
//SYS01 DD DSN=CICSTS56.CICS.DBFKINIT,DISP=SHR
//DFHDBFK DD DSN=CICSTS56.CICS.DFHDBKF,DISP=SHR
//SYSIN DD *
REPRO INFILE (SYS01) -
OUTFILE (DFHDBFK)
/*
//

```

*dbfkvol* は DFHDBFK データ・セットが作成されるボリュームで、*dbfkunit* はそのボリュームの装置タイプです。

図 21. DFHDBFK データ・セットを定義および初期化するサンプル・ジョブ

2. このジョブを使用して IMS コマンドをロードするか、CDBM トランザクション内の保守機能を使用します。

## CICS 実行のジョブ制御ステートメント

95 ページの図 21 に示されているサンプル JCL を使用して DFHDBFK データ・セットを定義した場合、CICS 実行のデータ定義ステートメントは次のようになります。

```
//DFHDBFK DD DSN=CICSTS56.CICS.DFHDBFK,DISP=SHR
```

また、動的ファイル割り振りを使用する場合、完全修飾データ・セット名を DFHDBFK ファイル・リソース定義に追加します。

## 関連情報

[CDBM GROUP コマンド・ファイル内のレコード・レイアウト](#)

## CMAC メッセージ・データ・セットの定義

CMAC メッセージ・データ・セットは、DFHMACD という名前の VSAM キー順データ・セット (KSDS) です。CMAC トランザクションは、DFHMACD を使用して CICS のメッセージおよびコードのオンライン記述を提供します。

### このタスクについて

CICS 領域で CICS メッセージ機能を使用するには、ユーザー独自の CSD グループ・リストを作成し、CICS メッセージ機能のための DFHMACD グループおよび CICS 領域に必要な他のリソース・グループを組み込まなければなりません。この新しいグループ・リストは、CICS 領域の始動時にシステム初期設定パラメーター **GRPLIST** に指定します。

CICS メッセージ機能用の DFHMACD リソース・グループは、その機能を使用する CICS 領域でのみ、指定してください。例えば、この機能は、端末所有領域では必要になりますが、データ所有領域では必要ないはずです。

### 手順

1. DFHMACD データ・セットを作成し、CICS 提供のメッセージおよびコード・データとともにそれをロードします。

次のいずれかの方法でこのデータ・セットを作成できます。

- DFHMACI ジョブを実行する DFHMACI ジョブの詳細については、『インストール』の『CICS データ・セットの作成』を参照してください。
  - 96 ページの『メッセージ・データ・セットを定義してロードするためのジョブ制御ステートメント』で説明されているように、提供されているサンプル・ジョブを使用します。
2. 提供されているサンプル・ジョブを使用してメッセージ・データ・セットを定義する場合は、CICS 始動 JCL に以下のデータ定義ステートメントを追加します。

```
//DFHMACD DD DSN=CICSTS56.CICS.DFHMACD,DISP=SHR
```

3. CMAC ファイル用の CICS リソース定義を作成します。

CMAC ファイルおよび CICS メッセージ機能に必要なその他のリソース用に提供されている定義は、CSD グループ DFHMACD に含まれています。CICS IVP には CMAC ファイル用の DD ステートメントが入っていますが、動的割り振りを行う場合には、CMAC ファイル用に提供されているリソース定義をコピーし、DSNAME オプションを追加してください。

4. 独自の CSD グループ・リストを作成し、提供されているグループ・リスト DFHLIST、CICS メッセージ機能用の DFHMACD グループ、および CICS 領域に必要な他のリソース・グループをそれに含めます。  
CICS 領域の始動時にシステム初期設定パラメーター **GRPLIST** でこのグループ・リストを指定します。

### タスクの結果

CICS メッセージは CMAC トランザクションから利用可能です。

### メッセージ・データ・セットを定義してロードするためのジョブ制御ステートメント

初めて使用する前に、DFHMACD データ・セットを VSAM キー順データ・セット (KSDS) として定義し、ロードする必要があります。以下のサンプル・ジョブは、これを実行する方法を示しています。

### このタスクについて

cmacvol が、DFHMACD データ・セットが作成されるボリュームであることに注意してください。

```

//CMACJOB JOB 'accounting information',name,MSGCLASS=A
//CMACDEF EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//AMSDUMP DD SYSOUT=*
//SYSIN DD *
DELETE CICSSTS56.CICS.DFHCMACD
SET MAXCC=0
DEFINE CLUSTER (
    NAME( CICSSTS56.CICS.DFHCMACD ) -
    CYL(2,1) -
    KEYS( 9 0 ) -
    INDEXED -
    VOLUME ( cmacvol) -
    RECORDSIZE( 8192 30646 ) -
    FREESPACE( 5 5 ) -
    SHAREOPTIONS( 2 ) -
    INDEX (
        NAME( CICSSTS56.CICS.DFHCMACD.INDEX ) -
    )
    DATA (
        NAME( CICSSTS56.CICS.DFHCMACD.DATA ) -
    )
)
/*
//CMACLOAD EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//AMSDUMP DD SYSOUT=*
//SYS01 DD DSN=CICSSTS56.CICS.SDFHMSGSGS(DFHCMACD),DISP=SHR
//DFHCMACD DD DSN=CICSSTS56.CICS.DFHCMACD,DISP=SHR
//SYSIN DD *
REPRO INFILE (SYS01) -
      OUTFILE (DFHCMACD)
/*
//

```

図 22. CMAC データ・セットを定義して初期設定するためのサンプル・ジョブ

## WS-AT データ・セットの定義

CICS Web Services Atomic Transaction (WS-AT) サポートの有効化に必要なデータ・セットは、WS-AT ディレクトリー・データ・セット DFHPIDIR です。WS-AT ディレクトリー・データ・セット DFHPIDIR は、コンテキストとタスク間のマッピングが含まれたファイルです。

WS-AT ディレクトリーおよびオブジェクト・ストア・データ・セット DFHPIDIR は、CSD 内のリソース定義を必要とする CICS ファイル制御管理対象データ・セットです。このデータ・セットは、WS-AT 対応の Web サービス・プロバイダーを共に提供する CICS 領域で共用されるため、以下のいずれかの形式でこのデータ・セットを CICS ファイル制御に定義できます。

- LSR モードまたは NSR モードのいずれかで開かれる通常の VSAM データ・セット。これは、プール番号を伴う LSRPOOLNUM 属性を使用するか、NONE として指定します。この場合、データ・セットは、他の領域がファイル要求を機能シッできる 1 つの CICS 領域 (ファイル所有領域) によって所有される必要があります。

データ・セットの定義は、CICS CSD グループ DFHPIVS で提供され、LSRPOOLNUM のデフォルト値 1 が使用されます。この CSD グループをコピーして名前変更し、ファイル定義を編集して、動的割り振り用のデータ・セット名、明示的な LSRPOOL リソース定義と一致させる特定の LSRPOOLNUM、リモート・システム属性などの詳細を追加します。

- RLS モードで開かれる VSAM データ・セット。このデータ・セットの定義は、RLSACCESS(YES) が指定された CICS CSD グループ DFHPIVR で提供されます。この CSD グループをコピーして名前変更し、ファイル定義を編集して、動的割り振り用のデータ・セット名などの属性を追加します。
- カップリング・ファシリティ・データ・テーブル。このデータ・セットの定義は、CICS CSD グループ DFHPICF で提供されます。この CSD グループをコピーして名前変更し、ファイル定義を編集して、プール名やテーブル名などの CFDT の詳細を変更します。

98 ページの図 23 は、DFHPIDIR の定義に使用されるステートメントを示しています。

```

DEFINE CLUSTER(NAME(@dsindex@.CICS@regname@.DFHPIDIR) -
INDEXED-
LOG(UNDO) -
CYL(2 1) -
VOLUME(@dsvol@) -
RECORDSIZE( 1017 1017 ) -
KEYS( 16 0 ) -
FREESPACE ( 10 10 ) -
SHAREOPTIONS( 2 3 )) -
DATA (NAME(@dsindex@.CICS@regname@.DFHPIDIR.DATA) -
CONTROLINTERVALSIZE(1024)) -
INDEX (NAME(@dsindex@.CICS@regname@.DFHPIDIR.INDEX))

```

図 23. WS-AT ディレクトリー・データ・セットを定義する JCL の例

1. ICF カタログにバックアウト・リカバリー属性を定義し、このジョブで定義されたデータ・セットを RLS モードまたは非 RLS モードで使用できるようにします。RLS モードで使用されるデータ・セットでは、ICF カタログにリカバリー属性を定義します。これらの属性は、CICS ファイル・リソース定義に指定されているあらゆる属性をオーバーライドします。PI ディレクトリー・データ・セットは、リカバリー可能として定義する必要があります。

2. VOLUME パラメーターにユーザー独自の値を指定するか、SMS 管理対象ストレージを使用している場合は、このパラメーターを完全に削除します。

3. デフォルトのレコード・サイズは 1 KB ですが、この値は変更できます。

**注:** 98 ページの図 23 に示された定義値はいずれも変更しません。DFHPIDIR には、1 つのレコードとその制御レコードのみ含まれています。ただし、ランタイム設定 (DD ステートメント上の AMP パラメーターのサブパラメーター BUFND、BUFNI、STRNO や、CICS ファイル・リソース定義内の同等の設定など) を必ず指定し、任意の時点でアクティブになる可能性がある要求の最大数に対応できるようにします。

## デバッグ・プロファイル・データ・セットのセットアップ

CICS で特定のデバッグ・ツールを使用するアプリケーション・プログラマーは、代替索引を持つ VSAM キー順データ・セットに保管されるデバッグ・プロファイルを作成します。

デバッグ・プロファイル・データ・セットをセットアップするには、次のようにします。

1. IDCAMS ユーティリティーを使用して、VSAM データ・セットを作成および初期化します。
2. データ・セットのファイル定義を作成します。複数の CICS 領域でデータ・セットを共用できます。また、データ・セットは次のものとして定義できます。

### VSAM RLS ファイル

同じシスプレックス内の複数の CICS 領域間でプロファイルを共用する場合は VSAM RLS ファイルを定義します。

### VSAM 非 RLS ファイル

同じシスプレックス内の領域間でプロファイルを共用する必要がある場合に、VSAM 非 RLS ファイルを定義します。

### リモート・ファイル

MRO または ISC を使用して接続された領域に保管されているプロファイルを使用する場合はリモート・ファイルを定義します。

## デバッグ・プロファイル・データ・セットの作成

デバッグ・プロファイル・データ・セットを作成するには、2 つの方法があります。IDCAMS ユーティリティーでデータ・セットを作成して初期化するという方法と、CICS 提供ジョブ DFHDEFDS を実行して CICS 領域用のデータ・セットを作成するという方法です。

IDCAMS ユーティリティーを使用して、以下の VSAM データ・セットを作成して初期化できます。

### DFHDPFMB

デバッグ・プロファイルの基本データ・セット。

## DFHDPFMP

デバッグ・プロファイルのパス・データ・セット。

## DFHDPFMX

デバッグ・プロファイルの代替索引データ・セット

99 ページの図 24 の JCL を使用してください。

```
//DPFM JOB 'accounting information',name,MSGCLASS=A
//DEFINE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
DELETE CICSSTS56.CICS.DFHDPFMB

DEFINE CLUSTER (RECORDS(1000) -
  NAME (CICSSTS56.CICS.DFHDPFMB) -
  SHAREOPTIONS(2 3) -
  LOG(NONE) -
  VOLUME (&DSVOL) -
  IXD) -
DATA -
  (RECSZ(2560,2560) -
  CONTROLINTERVALSIZE(3072) -
  NAME (CICSSTS56.CICS.DFHDPFMB.DATA) -
  KEYS(17 1) -
  FREESPACE(10 10) -
  BUFFERSPACE (8192)) -
INDEX -
  (NAME(CICSSTS56.CICS.DFHDPFMB.INDX))
//INITDP EXEC PGM=IDCAMS,REGION=512K
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
  REPRO INFILE ( SYS01 ) -
    OUTDATASET(CICSSTS56.CICS.DFHDPFMB)
//SYS01 DD *
DDUMMY RECORD          !! DO NOT ALTER !!
EEXAMPLE RECORD REMOVE THIS LINE IF SAMPLES NOT REQUIRED
/*
//DEFAULT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
  DEFINE ALTERNATEINDEX -
    ( NAME(CICSSTS56.CICS.DFHDPFMX ) -
    RECORDS(1000) -
    VOLUME(&DSVOL) -
    KEYS(12 20) -
    RELATE(CICSSTS56.CICS.DFHDPFMB) -
    RECORDSIZE(200 200) -
    SHAREOPTIONS(2 3) -
    UPGRADE ) -
  DATA -
    ( NAME(CICSSTS56.CICS.DFHDPFMX.DATA) ) -
  INDEX -
    ( NAME(CICSSTS56.CICS.DFHDPFMX.INDEX) )
  DEFINE PATH -
    ( NAME(CICSSTS56.CICS.DFHDPFMP) -
    PATHENTRY(CICSSTS56.CICS.DFHDPFMX) )
/*
//BLDDP EXEC PGM=IDCAMS
//BDSET1 DD DSN=CICSSTS56.CICS.DFHDPFMB,DISP=SHR
//ADSET1 DD DSN=CICSSTS56.CICS.DFHDPFMX,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
  BLDINDEX -
    INFILE(BDSET1) -
    OUTFILE(ADSET1)
/*
//*
```

図 24. デバッグ・プロファイル・データ・セットを作成するためのサンプル JCL

このサンプル JCL では、サンプル・デバッグ・プロファイルの入ったデータ・セットが作成されます。空のデータ・セットを作成する場合は、以下の行を削除してください。

```
EEXAMPLE RECORD REMOVE THIS LINE IF SAMPLES NOT REQUIRED
```

## VSAM RLS ファイルとしてのデバッグ・プロファイル・データ・セットの定義

同じシスプレックス内の複数の CICS 領域間でプロファイルを共有する場合は VSAM RLS ファイルを定義します。

CICS には、以下のサンプル定義が用意されています。

```
*-----*
*   Define base file for Debugging Profiles (RLS)   *
*-----*
DEFINE FILE(DFHDPFMB) GROUP(DFHDPVR)
DESCRIPTION(Debugging Profiles base file - VSAM RLS)
  RLSACCESS(YES)          TABLE(NO)
  LSRPOOLNUM(1)           DSNSHARING(ALLREQS)
  STRINGS(5)              STATUS(ENABLED)
  OPENTIME(FIRSTREF)      DISPOSITION(SHARE)
  DATABUFFERS(3)          INDEXBUFFERS(2)
  RECORDFORMAT(V)        READINTEG(REPEATABLE)
  ADD(YES)                BROWSE(NO)
  DELETE(YES)             READ(YES)
  UPDATE(YES)             JOURNAL(NO)
  JNLREAD(NONE)           JNLSYNCREAD(NO)
  JNLUPDATE(NO)          JNLADD(NONE)
  JNLSYNCWRITE(YES)       RECOVERY(NONE)
  FWDRECOVLOG(NO)        BACKUPTYPE(STATIC)
*-----*
*   Define path file for Debugging Profiles (RLS)   *
*-----*
DEFINE FILE(DFHDPFMP) GROUP(DFHDPVR)
DESCRIPTION(Debugging Profiles path file - VSAM RLS)
  RLSACCESS(YES)          TABLE(NO)
  LSRPOOLNUM(1)           DSNSHARING(ALLREQS)
  STRINGS(10)             STATUS(ENABLED)
  OPENTIME(FIRSTREF)      DISPOSITION(SHARE)
  DATABUFFERS(11)         INDEXBUFFERS(10)
  RECORDFORMAT(V)        READINTEG(REPEATABLE)
  ADD(YES)                BROWSE(NO)
  DELETE(YES)             READ(YES)
  UPDATE(YES)             JOURNAL(NO)
  JNLREAD(NONE)           JNLSYNCREAD(NO)
  JNLUPDATE(NO)          JNLADD(NONE)
  JNLSYNCWRITE(YES)       RECOVERY(NONE)
  FWDRECOVLOG(NO)        BACKUPTYPE(STATIC)
```

図 25. VSAM RLS ファイルとして定義されたデバッグ・プロファイル・データ・セットのリソース定義

1. DFHDPFMB および DFHDPFMP リソースのサンプルの FILE 定義を別のグループにコピーします。
2. DSNAME 属性を追加します。

- DFHDPFMB の場合は、デバッグ・プロファイルの基本データ・セットの名前を指定します。以下に例を示します。

```
DSNAME ==> CICSTS56.CICS.DFHDPFMB
```

- DFHDPFMP の場合は、デバッグ・プロファイルのパス・データ・セットの名前を指定します。以下に例を示します。

```
DSNAME ==> CICSTS56.CICS.DFHDPFMP
```

また、DSNAME 属性を省略し、DD カードを CICS 始動 JCL に含めることもできます。以下に例を示します。

```
//DFHDPFMB DD DSN=CICSTS56.CICS.DFHDPFMB,DISP=SHR
//DFHDPFMP DD DSN=CICSTS56.CICS.DFHDPFMP,DISP=SHR
```

3. FILE 定義をインストールします。

## VSAM 非 RLS ファイルとしてのデバッグ・プロファイル・データ・セットの定義

同じシスプレックス内の領域間でプロファイルを共有する必要がない場合は VSAM 非 RLS ファイルを定義します。

CICS には、以下のサンプル定義が用意されています。

```

*-----*
*   Define base file for Debugging Profiles (non-RLS)   *
*-----*
DEFINE FILE(DFHDPFMB) GROUP(DFHDPVSL)
DESCRIPTION(Debugging Profiles Base File)
    RLSACCESS(NO)          LSRPOOLNUM(1)
    READINTEG(UNCOMMITTED) DSNSHARING(ALLREQS)
    STRINGS(10)            STATUS(ENABLED)
    OPENTIME(FIRSTREF)     DISPOSITION(SHARE)
    DATABUFFERS(11)        INDEXBUFFERS(10)
    TABLE(NO)             RECORDFORMAT(V)
    ADD(YES)               BROWSE(YES)
    DELETE(YES)            READ(YES)
    UPDATE(YES)            JOURNAL(NO)
    JNLREAD(NONE)          JNLSYNCREAD(NO)
    JNLUPDATE(NO)          JNLADD(NONE)
    JNLSYNWRITE(NO)        RECOVERY(NONE)
    FWDRECOVLOG(NO)        BACKUPTYPE(STATIC)
*-----*
*   Define path file for Debugging Profiles (non-RLS)   *
*-----*
DEFINE FILE(DFHDPFMP) GROUP(DFHDPVSL)
DESCRIPTION(Debugging Profiles Path File)
    RLSACCESS(NO)          LSRPOOLNUM(1)
    READINTEG(UNCOMMITTED) DSNSHARING(ALLREQS)
    STRINGS(10)            STATUS(ENABLED)
    OPENTIME(FIRSTREF)     DISPOSITION(SHARE)
    DATABUFFERS(11)        INDEXBUFFERS(10)
    TABLE(NO)             RECORDFORMAT(V)
    ADD(YES)               BROWSE(YES)
    DELETE(YES)            READ(YES)
    UPDATE(YES)            JOURNAL(NO)
    JNLREAD(NONE)          JNLSYNCREAD(NO)
    JNLUPDATE(NO)          JNLADD(NONE)
    JNLSYNWRITE(NO)        RECOVERY(NONE)
    FWDRECOVLOG(NO)        BACKUPTYPE(STATIC)

```

図 26. VSAM 非 RLS ファイルとして定義されたデバッグ・プロファイル・データ・セットのリソース定義

1. DFHDPFMB および DFHDPFMP リソースのサンプルの FILE 定義を別のグループにコピーします。

2. DSNNAME 属性を追加します。

- DFHDPFMB の場合は、デバッグ・プロファイルの基本データ・セットの名前を指定します。以下に例を示します。

```
DSNAME ==> CICSTS56.CICS.DFHDPFMB
```

- DFHDPFMP の場合は、デバッグ・プロファイルのパス・データ・セットの名前を指定します。以下に例を示します。

```
DSNAME ==> CICSTS56.CICS.DFHDPFMP
```

また、DSNAME 属性を省略し、DD カードを CICS 始動 JCL に含めることもできます。以下に例を示します。

```
//DFHDPFMB DD DSN=CICSTS56.CICS.DFHDPFMB,DISP=SHR
//DFHDPFMP DD DSN=CICSTS56.CICS.DFHDPFMP,DISP=SHR
```

3. FILE 定義をインストールします。

### リモート・ファイルとしてのデバッグ・プロファイル・データ・セットの定義

MRO または ISC を使用して接続された領域に保管されているプロファイルを使用する場合はリモート・ファイルを定義します。

CICS には、以下のサンプル定義が用意されています。



```

*-----*
*   Define base file for Debugging Profiles (non-RLS remote)   *
*-----*
DEFINE FILE(DFHDPFMB) GROUP(DFHDPVSR)
DESCRIPTION(Debugging Profile Base File - VSAM Remote)
      REMOTESYSTEM(CICA)      REMOTENAME(DFHDPFMB)
*-----*
*   Define path file for Debugging Profiles (non-RLS remote)   *
*-----*
DEFINE FILE(DFHDPFMP) GROUP(DFHDPVSR)
DESCRIPTION(Debugging Profile Path File - VSAM Remote)
      REMOTESYSTEM(CICA)      REMOTENAME(DFHDPFMP)

```

図 27. リモート・ファイルとして定義されたデバッグ・プロファイル・データ・セットのリソース定義

1. DFHDPFMB および DFHDPFMP リソースのサンプルの FILE 定義を別のグループにコピーします。
2. DSNNAME 属性を追加します。

- DFHDPFMB の場合は、デバッグ・プロファイルの基本データ・セットの名前を指定します。以下に例を示します。

```
DSNAME ==> CICSTS56.CICS.DFHDPFMB
```

- DFHDPFMP の場合は、デバッグ・プロファイルのパス・データ・セットの名前を指定します。例:

```
DSNAME ==> CICSTS56.CICS.DFHDPFMP
```

また、DSNAME 属性を省略し、DD カードを CICS 始動 JCL に含めることもできます。以下に例を示します。

```
//DFHDPFMB DD DSN=CICSTS56.CICS.DFHDPFMB,DISP=SHR
//DFHDPFMP DD DSN=CICSTS56.CICS.DFHDPFMP,DISP=SHR
```

3. FILE 定義をインストールします。

リモート・ファイルを定義する場合、対応するファイル定義をリモート・システムにインストールする必要があります。

## データ・セットの暗号化

z/OS データ・セット暗号化がサポートされている、CICS で使用するあらゆるデータ・セットを暗号化できます。

### 始める前に

z/OS データ・セット暗号化がサポートされているデータ・セット・タイプ、および暗号化の対象となる CICS システム・データ・セットを確認します。[データ・セット暗号化の計画](#)を参照してください。

使用する鍵ラベルをセットアップしておきます。「[z/OS 暗号サービス ICSF 管理者ガイド](#)」の『[鍵生成プログラム・ユーティリティー・プログラムを使用した暗号鍵の管理](#)』を参照してください。

暗号化データ・セットを作成する権限の設定および鍵ラベルに対するアクセス権限の設定を行う RACF タスクを完了しておきます。これには、以下のタスクが含まれます。

- 暗号化データ・セットをユーザーが作成することを許可するために、STGADMIN.SMS.ALLOW.DATASET.ENCRYPT CL (FACILITY) に対する読み取り権限を付与します。
- 鍵ラベルに対する読み取り権限を付与します。ユーザーが暗号化されたデータ・セットにアクセスしようとすると、RACF は、CSFKEYS クラスおよび CSFSERV クラス内の関連するプロファイルへのアクセスを検査することによって、暗号化の前にユーザーが鍵ラベルを使用する権限を持っているかどうかを検査します。これらのクラスについて詳しくは、「[z/OS DFSMS データ・セットの使用法](#)」の『[データ・セット暗号化](#)』を参照してください。

### 手順

1. 暗号鍵および関連する鍵ラベルを作成するか、既存の鍵ラベルを使用します。

2. 拡張フォーマットとデータ・セット鍵ラベルを指定して、データ・セットの新規インスタンスを割り振ります。

a) 以下のいずれかの方法で、拡張フォーマットを指定します。

- SMS データ・クラスで、**DSNTYPE=EXT** パラメーターと **R** (必須) サブパラメーターまたは **P** (任意) サブパラメーターを使用する。拡張フォーマットでデータ・セットが割り振られるようにするには、**R** を指定します。
- JCL DD ステートメントで **DSNTYPE** パラメーターを使用し、値として **EXTREQ** (必須) または **EXTPREF** (任意) を指定する。拡張フォーマットでデータ・セットが割り振られるようにするには、**EXTREQ** を指定します。JCL DD ステートメントで指定した **DSNTYPE** は、データ・クラスのどの **DSNTYPE** セットよりも優先されます。

拡張フォーマットを指定するとき、データ・セットにもそのオプションが必要な場合は、拡張アドレッシング属性のみを指定するようにしてください。

b) 以下のいずれかの方法で、データ・セット鍵ラベルを指定します。

- RACF データ・セット・プロファイルで、**DATAKEY** キーワードを使用します。
- JCL で **DSKEYLBL** キーワードを使用する。
- 動的割り振りで **DALDKYL** テキスト・ユニットを使用する。
- TSO 割り振りで **DSKEYLBL** を使用する。
- IDCAMS DEFINE で **DEFINE CLUSTER** の **KEYLABEL** パラメーターを使用する。
- SMS データ・クラスで、**DEFINE/ALTER** パネルの「データ・セット鍵ラベル (Data Set Key Label)」フィールドを指定する。

複数のインターフェースでデータ・セット鍵ラベルが指定された場合の優先順位は、次のようになります。

- 1) RACF データ・セット・プロファイル
- 2) JCL、動的割り振り、TSO 割り振り、または IDCAMS DEFINE
- 3) SMS データ・クラス

鍵ラベルの管理のしやすさとデータを最大限保護することのバランスが取れるように、暗号化を計画しているデータ・セットのグループ全体で使用するさまざまな鍵ラベルの細分度を考慮に入れてください。これは、鍵ラベルの指定に使用する適切な手段を決定する際にも役立ちます。

3. 既存のデータ・セットから新しい暗号化データ・セットにデータをコピーします。

REPRO 関数、または EXPORT 関数と IMPORT 関数などの適切な DFSMS 関数を使用してください。一般的な手順は以下のとおりです。

- a. 新規データ・セットを作成します。
- b. 古いデータ・セットからデータをコピーします。
- c. 古いデータ・セットを削除します。
- d. 新規データ・セットの名前を変更して、古い名前に戻します。

### データ・セット暗号化の計画

z/OS データ・セット暗号化がサポートされている、CICS で使用するあらゆるデータ・セットを暗号化できます。そうしたデータには、CICS ファイル制御 API を介してアクセスされるユーザー・データ・セット、CICS 区画外一時データに使用される待機順次アクセス方式 (QSAM) データ・セット、CICS で使用される基本順次アクセス方式 (BSAM) データ・セット、暗号化の適切な候補となる CICS システム・データ・セットが含まれます。

CICS TS for z/OS の任意のサービス中リリースで、データ・セット暗号化を使用することができます。z/OS データ・セット暗号化は、z/OS 2.2 (APAR OA50569 の PTF 適用済み) 以降のリリースでサポートされます。

暗号化データ・セットは、ストレージ管理サブシステム (SMS) で管理する必要があり、拡張フォーマットでなければなりません。暗号化データ・セットを作成するには、新規データ・セットを割り振るときに、そのデータ・セットに鍵ラベルを割り当てます。その鍵ラベルは、データの暗号化または暗号化解除に使

用される、Integrated Cryptographic Service Facility (ICSF) 暗号鍵データ・セット (CKDS) の AES-256 ビット暗号鍵を指している必要があります。鍵ラベルは機密情報ではありませんが、鍵ラベルによって識別する暗号鍵は機密情報です。

## CICS ユーザー・データ・セットの暗号化サポート

CICS に定義されるユーザー・データ・セットにおける暗号化サポートには、基本 VSAM および VSAM レコード・レベル共用 (RLS) を介してアクセスされる、キー順データ・セット (KSDS)、入力順データ・セット (ESDS)、相対レコード・データ・セット (RRDS)、および可変相対レコード・データ・セット (VRRDS) が含まれます。また、共用データ・テーブルまたはカップリング・ファシリティ・データ・テーブルで使用されるバッキング VSAM キー順データ・セットでも、暗号化がサポートされます。

## CICS システム・データ・セットの暗号化サポート

暗号化がサポートされている、あらゆる CICS システム・データ・セットを暗号化できます。しかし、システム・データ・セットの中には暗号化に適しているものと、適していないものがあります。

- 機密データを含む可能性がある データ・セットは、暗号化に適しています。機密データを含む可能性があるのはどのデータ・セットか、またそれらを暗号化するかどうかを判断する必要があります。
- 機密データを含まないデータ・セットと、含む可能性が低いデータ・セットは、暗号化に適していません。

以下の表では、暗号化が可能な CICS システム・データ・セットをリストし、暗号化に適しているかどうかを示します。すべてを暗号化する方式を採用する場合は、この表にリストされているすべてのデータ・セット・タイプを暗号化することができます。

表 17. 暗号化に適しているシステム・データ・セット		
CICS システム・データ・セット	暗号化に適しているか?	特別な考慮事項
一時記憶域データ・セット (DFHTEMP)	はい。機密データを含む可能性があります。	DFHTEMP は拡張フォーマットをサポートしますが、拡張アドレッシングをサポートしません。
区画内一時データ (DFHINTRA)	はい。機密データを含む可能性があります。	DFHINTRA は拡張フォーマットをサポートしますが、拡張アドレッシングをサポートしません。
区画外一時データ	はい。機密データを含む可能性があります。	区分データ・セット (PDS) では、暗号化がサポートされません。
補助トレース・データ・セット (DFHAUXT および DFHBUXT)	はい。診断で機密データを含む可能性があります。別の方法として、 <u>CONFDATA システム初期設定パラメーター</u> を使用して、十分な保護が得られる可能性があります。	トレース・データが暗号化されていて、それを診断用に IBM に送信する必要がある場合は、CICS トレース・フォーマット設定か、または別の方法を使用して、暗号化解除されたデータを送信してください。
CICS ダンプ・データ・セット (DFHDMPA および DFHDMPB)	はい。診断で機密データを含む可能性があります。別の方法として、 <u>CONFDATA システム初期設定パラメーター</u> を使用して、十分な保護が得られる可能性があります。	<p>ダンプ・データが暗号化されていて、それを診断用に IBM に送信する必要がある場合は、CICS ダンプ・フォーマット設定か、または別の方法を使用して、暗号化解除されたデータを送信してください。</p> <p>CICS ダンプ・データ・セットは拡張フォーマットをサポートしますが、拡張アドレッシングをサポートしません。</p>

表 17. 暗号化に適しているシステム・データ・セット (続き)		
CICS システム・データ・セット	暗号化に適しているか?	特別な考慮事項
Doctemplate リソース	はい、機密データを含む可能性があります。	暗号化がサポートされるタイプのデータ・セットを使用していることが前提です。
静的配信に使用される URIMAP リソース	はい、機密データを含む可能性があります。	暗号化がサポートされるタイプのデータ・セットを使用していることが前提です。
BTS リポジトリ・データ・セットおよび BTS ローカル要求キュー (LRQ) データ・セット	いいえ。制御データのみが含まれます。	なし
グローバル・カタログ・データ・セットおよびローカル・カタログ・データ・セット (DFHGCDD および DFHLCD)	いいえ。構成データのみが含まれます。	CICS 構成データが機密情報であると考えられる場合のみ、検討します。
CICS システム定義データ・セット (DFHCSD)	いいえ。リソース構成に関する情報のみが含まれます。	リソース定義に機密情報が含まれると考えられる場合のみ、検討します。
CMAC メッセージ・データ・セット (DFHCMACD)	いいえ。メッセージの詳細のみが含まれます。	機密データを含むと考えられる、独自のメッセージを追加した場合のみ、検討します。
バンドル定義および Web サービスに使用する zFS ファイル	いいえ。構成情報のみが含まれます。	バンドル定義に機密情報が含まれると考えられる場合のみ、検討します。

## 関連タスク

102 ページの『データ・セットの暗号化』

z/OS データ・セット暗号化がサポートされている、CICS で使用するあらゆるデータ・セットを暗号化できます。

## CICS システム初期化パラメーターの指定

CICS システム 初期設定パラメーターを使用すると、CICS 領域の始動時に CICS システム属性を変更できます。CICS にパラメーターを定義するためのさまざまな方式があります。各システム 初期設定パラメーターの説明には、構文と詳細な説明が含まれています。

システム 初期設定パラメーターによって定義される情報は、以下の 3 つのカテゴリにグループ化できます。

1. CICS システム 機能の初期設定と制御に使用する情報 (例えば、動的ストレージ域限界および領域終了時間間隔などの情報)
2. 独自のバージョンの CICS 管理テーブルをロードする際に使用するモジュール接尾部 (例えば、DFHMCTxx)
3. 初期設定処理を制御するのに使用する特別な情報

システム 初期設定パラメーターを提供するための基本的な方法は、システム 初期設定テーブル (SIT) を使用するというものです。ロード・テーブルとしてアセンブルされる SIT のパラメーターによって、使用している固有の環境に合わせてシステムを初期設定するのに必要なほとんどの情報がシステム 初期設定プログラムに提供されます。複数の SIT を生成しておき、システム 初期設定時にニーズに合った適切なものを選択できます。

また、SIT ではコーディングできない、その他のシステム 初期設定パラメーターも指定できます。以下の 3 つの方法のいずれかにより、使用する SIT および他のシステム 初期設定パラメーター (例外がいくつかあります) を指定できます。

1. EXEC PGM=DFHSIP ステートメントの PARM パラメーターにより指定。
2. 始動ジョブ・ストリームに定義されている SYSIN データ・セットにより指定。
3. システム・オペレーターのコンソールを介して指定。

システム初期設定処理での入力にこうした方法を使用することにより、SIT にアセンブルされたシステム初期設定パラメーターのほとんどを指定変更することもできます。

システム初期設定パラメーターの中には SIT にコーディングできないものがある一方、他の方法では指定できないパラメーターもあります。最終的に、システム・オペレーターのコンソールから指定できないシステム初期設定パラメーターもあります。パラメーターの要約と、それらの指定方法については、[システム初期設定パラメーターの説明と要約](#)を参照してください。

## 新規の CICS リリースにアップグレードする際の作業

新規の CICS リリースにアップグレードする際に、既存のシステム初期設定テーブルがある場合、それに変更を加える必要があります。廃止されたすべてのパラメーターを削除するとともに、新規パラメーターまたは変更されたパラメーターをデフォルト値以外で実行する場合には、それらのために必要な値を指定します。必要な変更を加えたなら、CICS Transaction Server for z/OS, バージョン 5 リリース 6 マクロ・ライブラリーを使用してテーブルを再アセンブルします。

CICS 始動プロシージャで定義したシステム初期設定パラメーターがある場合には、それらも変更する必要があります。

各 CICS 領域に特定のシステム初期設定テーブルが生成されないようにするための簡単な解決策は、CICS が始動時にデフォルトの接尾部なしのテーブル (DFHSIT) をロードし、各領域のシステム初期設定パラメーターを SYSIN データ・セットに提供するようにすることです。デフォルトのシステム初期設定テーブルのソースについて詳しくは、[デフォルトのシステム初期設定テーブル](#)を参照してください。

## CICS 領域をセットアップするためのシステム初期設定パラメーター

CICS 領域をセットアップするために構成する必要がある、またはデフォルト値から変更する必要がある CICS システム初期設定パラメーターをリストしています。構成するかどうか検討できるシステム初期設定パラメーターもリストしています。

次のセクションにリストされていないパラメーターについては、一般に、使用環境に特別な要件がない限り、デフォルト値から変更する必要はありません。

システム初期設定パラメーターの完全なリストとそのデフォルト値については、[システム初期設定パラメーターの説明と要約](#)を参照してください。

### 構成する必要があるシステム初期設定パラメーター

次のシステム初期設定パラメーターを構成する必要があります。

#### APPLID

**APPLID** パラメーターは、この CICS 領域に z/OS Communications Server アプリケーション ID を指定します。[APPLID システム初期設定パラメーター](#)を参照してください。

#### CICSSVC

**CICSSVC** パラメーターは、CICS タイプ 3 SVC に割り当てた番号を指定します。[CICSSVC システム初期設定パラメーター](#)を参照してください。

#### GRPLIST

**GRPLIST** システム初期設定パラメーターは、CICS システム定義 (CSD) ファイルのリソース定義グループの最大 4 つのリストの名前 (それぞれ 1 文字から 8 文字) を指定します。[GRPLIST システム初期設定パラメーター](#)を参照してください。

#### ISC

**ISC** システム初期設定パラメーターは、複数領域操作 (MRO) および SNA 経由のシステム間通信に必要な CICS プログラムを組み込むかどうかを指定します。[ISC システム初期設定パラメーター](#)を参照してください。



## **MXT**

**MXT** システム 初期設定パラメーターは、CICS システムに同時に存在できるユーザー・タスクの最大数 (10 から 2000 までの範囲) を指定します。MXT 値には、CICS システム・タスクは含まれません。[MXT システム 初期設定パラメーター](#)を参照してください。

## **PLTPI**

**PLTPI** システム 初期設定パラメーターは、システム 初期設定の最終段階で実行されるプログラムのリストを入れたプログラム・リスト・テーブルの接尾部またはフルネームを指定します。[PLTPI システム 初期設定パラメーター](#)を参照してください。

## **PLTSD**

**PLTSD** システム 初期設定パラメーターは、システム 終了時に実行されるプログラムのリストを入れたプログラム・リスト・テーブルの接尾部またはフルネームを指定します。[PLTSD システム 初期設定パラメーター](#)を参照してください。

## **SIT**

**SIT** システム 初期設定パラメーターは、初期設定の開始時に CICS でロードするシステム 初期設定テーブルの接尾部を指定します (存在する場合)。[SIT システム 初期設定パラメーター](#)を参照してください。

## **START**

**START** システム 初期設定パラメーターは、システム 初期設定プログラムの開始のタイプを指定します。[START システム 初期設定パラメーター](#)を参照してください。

## **SUFFIX**

**SUFFIX** システム 初期設定パラメーターは、このシステム 初期設定テーブルの名前の最後の 2 文字を指定します。[SUFFIX システム 初期設定パラメーター](#)を参照してください。

## **SYSIDNT**

**SYSIDNT** システム 初期設定パラメーターは、CICS 領域でのみ認識される名前 (1 文字から 4 文字) を指定します。[SYSIDNT システム 初期設定パラメーター](#)を参照してください。

## **USSHOME**

**USSHOME** システム 初期設定パラメーターは、z/OS UNIX 上の CICS Transaction Server ファイルのルート・ディレクトリーの名前とパスを指定します。[USSHOME システム 初期設定パラメーター](#)を参照してください。

## **構成する必要がある可能性のあるシステム初期設定パラメーター**

次のシステム 初期設定パラメーターは、構成する必要があるかどうか検討する必要があります。

## **AICONS**

**AICONS** パラメーターは、コンソールの自動インストール・サポートを設定するかどうかを指定します。[AICONS システム 初期設定パラメーター](#)を参照してください。

## **AIRDELAY**

**AIRDELAY** パラメーターは、緊急時再始動が発生したときから、セッションにない自動インストール済み端末項目および APPC 接続項目が削除されるまでの遅延時間を指定します。[AIRDELAY システム 初期設定パラメーター](#)を参照してください。

## **CLINTCP**

**CLINTCP** パラメーターは、DFHCNV データ変換テーブルで使用されるデフォルトのクライアント・コード・ページを指定します (ただし、DFHCNV マクロの **CLINTCP** パラメーターが SYSDEF に設定される場合のみ)。[CLINTCP システム 初期設定](#)を参照してください。

## **CMDPROT**

**CMDPROT** パラメーターは、EXEC CICS コマンドの出力パラメーターで参照されているストレージの開始アドレスに対する CICS 検証を許可するか、それとも禁止するかを指定します。[CMDPROT システム 初期設定パラメーター](#)を参照してください。

## **CONFDATA**

**CONFDATA** パラメーターは、CICS トレース・エントリーまたはダンプ内に表示される可能性がある機密データを CICS で編集して表示されないようにするかどうかを指定します。[CONFDATA システム 初期設定パラメーター](#)を参照してください。



## CONFTEXT

**CONFTEXT** システム 初期設定パラメーターは、CICS が z/OS Communications Server によるユーザー・データのトレースを抑制するかどうかを指定します。[CONFTEXT システム 初期設定パラメーター](#)を参照してください。

## CPSMCONN

**CPSMCONN** パラメーターは、領域の初期設定中に、指定した CICSplex SM コンポーネントを CICS で起動するかどうかを指定します。[CPSMCONN システム 初期設定パラメーター](#)を参照してください。

## CRLPROFILE

**CRLPROFILE** パラメーターは、LDAP サーバーに保管されている認証取り消しリスト (CRL) へのアクセスを CICS に許可するために使用するプロファイルの名前を指定します。[CRLPROFILE システム 初期設定パラメーター](#)を参照してください。

## CSDDISP

**CSDDISP** パラメーターは、CSD に割り振られるデータ・セットの処理を指定します。[CSDDISP システム 初期設定パラメーター](#)を参照してください。

## CSDDSN

**CSDDSN** パラメーターは、CSD で使用される 1 文字から 44 文字の JCL データ・セット名 (DSNAME) を指定します。[CSDDSN システム 初期設定パラメーター](#)を参照してください。

## CSDFRLOG

**CSDFRLOG** パラメーターは、CICS で CSD の順方向リカバリー・ログ・ストリームの識別に使用するジャーナル名に対応する番号を指定します。[CSDFRLOG システム 初期設定パラメーター](#)を参照してください。

## CSDLSRNO

**CSDLSRNO** システム 初期設定パラメーターは、CSD をローカル共用リソース (LSR) プールと関連付けるかどうかを指定します。[CSDLSRNO システム 初期設定パラメーター](#)を参照してください。

## CSDRECOV

**CSDRECOV** システム 初期設定パラメーターは、CSD がリカバリー可能なファイルかどうかを指定します。[CSDRECOV システム 初期設定パラメーター](#)を参照してください。

## DATFORM

**DATFORM** システム 初期設定パラメーターは、CICS 日付データ表示で使用する外部日付表示標準を指定します。[DATFORM システム 初期設定パラメーター](#)を参照してください。

## DBCTLCON

**DBCTLCON** システム 初期設定パラメーターは、初期設定時に CICS で DBCTL 接続を自動的に開始するかどうかを指定します。[DBCTLCON システム 初期設定パラメーター](#)を参照してください。

## DB2CONN

**DB2CONN** システム 初期設定パラメーターは、初期設定時に CICS で Db2 接続を自動的に開始するかどうかを指定します。[DB2CONN システム 初期設定パラメーター](#)を参照してください。

## DFLTUSER

**DFLTUSER** システム 初期設定パラメーターは、デフォルト・ユーザーの RACF ユーザー ID を指定します。つまり、他に具体的なユーザー指定がない場合には、このデフォルト・ユーザーのセキュリティ属性を使用して CICS リソースを保護します。[DFLTUSER システム 初期設定パラメーター](#)を参照してください。

## DOCCODEPAGE

**DOCCODEPAGE** システム 初期設定パラメーターは、文書ドメインで使用するデフォルトのホスト・コード・ページを指定します。[DOCCODEPAGE システム 初期設定パラメーター](#)を参照してください。

## DSALIM

**DSALIM** システム 初期設定パラメーターは、CICS が 24 ビット・ストレージ (16 MB 境界より下) に常駐する個別の動的ストレージ域 (DSA) を割り振ることができる、ストレージの総量の上限を指定します。[DSALIM システム 初期設定パラメーター](#)を参照してください。

## DSRTPGM

**DSRTPGM** システム 初期設定パラメーターは、分散ルーティング・プログラムの名前を指定します。[DSRTPGM システム 初期設定パラメーター](#)を参照してください。

## DTRPGM

**DTRPGM** システム 初期設定パラメーターは、動的ルーティング・プログラムの名前を指定します。  
[DTRPGM システム 初期設定パラメーター](#)を参照してください。

## DUMPSW

**DUMPSW** システム 初期設定パラメーターは、CICS で最初のダンプ・データ・セットがいっぱいになったときに次のダンプ・データ・セットに自動的に切り替えるかどうかを指定します。[DUMPSW システム 初期設定パラメーター](#)を参照してください。

## EDSALIM

**EDSALIM** システム 初期設定パラメーターは、CICS が 31 ビット (16 MB 境界より上) の記憶域 (つまり、16 MB より上で 2 GB より下の記憶域) に存在する個別の拡張動的ストレージ域 (EDSA) を割り当てることのできる記憶域の合計量の上限を指定します。[EDSALIM システム 初期設定パラメーター](#)を参照してください。

## FCQRONLY

**FCQRONLY** システム 初期設定パラメーターは、すべてのファイル制御要求を CICS QR TCB の下で実行するように CICS が強制するかどうかを指定します。このパラメーターは、VSAM RLS ファイルおよびローカル VSAM LSR ファイルにアクセスするファイル制御要求に適用されます。他のすべてのファイル・タイプに対する要求は、常に QR TCB 上で実行されます。[FCQRONLY システム 初期設定パラメーター](#)を参照してください。

## GMTEXT

**GMTEXT** システム 初期設定パラメーターは、デフォルト・ログオン・メッセージ・テキスト (WELCOME TO CICS) または独自のメッセージ・テキストを画面に表示するかどうかを指定します。[GMTEXT システム 初期設定パラメーター](#)を参照してください。

## GMTRAN

**GMTRAN** システム 初期設定パラメーターは、トランザクションの ID を指定します。[GMTRAN システム 初期設定パラメーター](#)を参照してください。

## GRNAME

**GRNAME** システム 初期設定パラメーターは、CICSplex の CICS 端末専有領域グループを z/OS Communications Server に登録する際の z/OS Communications Server 総称リソース名 (1 文字から 8 文字) を指定します。[GRNAME システム 初期設定パラメーター](#)を参照してください。

## ICV

**ICV** システム 初期設定パラメーターは、領域終了時間間隔 (ミリ秒) を指定します。[ICV システム 初期設定パラメーター](#)を参照してください。

## ICVR

**ICVR** システム 初期設定パラメーターは、デフォルトのランナウェイ・タスク時間をミリ秒単位で 10 進数として指定します。[ICVR システム 初期設定パラメーター](#)を参照してください。

## INITPARM

**INITPARM** システム 初期設定パラメーターは、**ASSIGN INITPARM** コマンドを使用するアプリケーション・プログラムにパラメーターが渡されることを指定します。[INITPARM システム 初期設定パラメーター](#)を参照してください。

## IRCSTRT

**IRCSTRT** システム 初期設定パラメーターは、システムの初期設定時に IRC を開始するかどうかを指定します。[IRCSTRT システム 初期設定パラメーター](#)を参照してください。

## JVMPROFILEDIR

**JVMPROFILEDIR** システム 初期設定パラメーターは、CICS の JVM プロファイルを含んでいる z/OS UNIX ディレクトリーの名前 (最大 240 文字) を指定します。CICS は JVM を構成するために必要なプロファイルをこのディレクトリーで検索します。[JVMPROFILEDIR システム 初期設定パラメーター](#)を参照してください。

## KEYRING

**KEYRING** システム 初期設定パラメーターは、Secure Sockets Layer (SSL) および Web サービス・セキュリティで CICS サポートによって使用される鍵および X.509 証明書を含んでいる、外部セキュリティ・マネージャーのデータベース内の鍵リングの完全修飾名を指定します。これは CICS 領域 ID に

よって所有される必要があります。最初の鍵リングは、DFH\$RING exec を使用して [CICSTS56.CICS.SDFHSAMPKEYRING システム 初期設定パラメーター](#)を参照してください。

#### **LGNMSG**

**LGNMSG** システム 初期設定パラメーターは、z/OS Communications Server ログオン・データをアプリケーション・プログラムに提供するかどうかを指定します。[LGNMSG システム 初期設定パラメーター](#)を参照してください。

#### **LOCALCCSID**

**LOCALCCSID** システム 初期設定パラメーターは、ローカル領域のデフォルト CCSID を指定します。[LOCALCCSID システム 初期設定パラメーター](#)を参照してください。

#### **LPA**

**LPA** システム 初期設定パラメーターは、CICS およびユーザー・モジュールをリンク・パック域から使用できるようにするかどうかを指定します。[LPA システム 初期設定パラメーター](#)を参照してください。

#### **MINTLSLEVEL**

**MINTLSLEVEL** システム 初期設定パラメーターは、CICS がセキュア TCP/IP 接続に使用する最小 TLS プロトコルを指定します。[MINTLSLEVEL システム 初期設定パラメーター](#)を参照してください。

#### **MN**

**MN** システム 初期設定パラメーターは、初期設定時にモニターをオンまたはオフに切り替えるかどうかを指定します。[MN システム 初期設定パラメーター](#)を参照してください。

#### **MNPER**

**MNPER** システム 初期設定パラメーターは、CICS 初期設定中にモニター・パフォーマンス・クラスをアクティブにするかどうかを指定します。[MNPER システム 初期設定パラメーター](#)を参照してください。

#### **MQCONN**

システム 初期設定パラメーター **MQCONN** は、CICS から IBM MQ への接続を初期設定時に自動的に開始するかどうかを指定します。[MQCONN システム 初期設定パラメーター](#)を参照してください。

#### **MSGCASE**

**MSGCASE** システム 初期設定パラメーターは、メッセージ・ドメインでの大/小文字混合メッセージの表示方法を指定します。[MSGCASE システム 初期設定パラメーター](#)を参照してください。

#### **NATLANG**

**NATLANG** システム 初期設定パラメーターは、この CICS の実行でサポートされる言語の単一文字コードを指定します。[NATLANG システム 初期設定パラメーター](#)を参照してください。

#### **PGCHAIN**

**PGCHAIN** システム 初期設定パラメーターは、端末管理で BMS 端末ページ・チェーン・コマンドとして識別される文字ストリングを指定します。[PGCHAIN システム 初期設定パラメーター](#)を参照してください。

#### **PGCOPY**

**PGCOPY** システム 初期設定パラメーターは、端末管理で 1 つの端末から別の端末に出力をコピーする BMS コマンドとして識別される文字ストリングを指定します。[PGCOPY システム 初期設定パラメーター](#)を参照してください。

#### **PGPURGE**

**PGPURGE** システム 初期設定パラメーターは、端末管理で BMS 端末ページ除去コマンドとして識別される文字ストリングを指定します。[PGPURGE システム 初期設定パラメーター](#)を参照してください。

#### **PGRET**

**PGRET** システム 初期設定パラメーターは、端末管理で BMS 端末ページ検索コマンドとして認識される文字ストリングを指定します。[PGRET システム 初期設定パラメーター](#)を参照してください。

#### **PLTPISEC**

**PLTPISEC** システム 初期設定パラメーターは、CICS 初期設定時に CICS で PLT プログラムに対してコマンド・セキュリティーおよびリソース・セキュリティーの検査を実行するかどうかを指定します。[PLTPISEC システム 初期設定パラメーター](#)を参照してください。

## PLTPIUSR

**PLTPIUSR** システム 初期設定パラメーターは、CICS 初期設定中に実行される PLT プログラムに対するセキュリティ検査で CICS が使用するユーザー ID を指定します。[PLTPIUSR システム 初期設定パラメーター](#)を参照してください。

## PSBCHK

**PSBCHK** システム 初期設定パラメーターは、トランザクション・ルーティングを使ってこの CICS 領域のトランザクションを開始して接続済み IMS システムにアクセスするリモート 端末ユーザーに対して、CICS で PSB 許可検査を実行するかどうかを指定します。[PSBCHK システム 初期設定パラメーター](#)を参照してください。

## RLS

**RLS** システム 初期設定パラメーターは、CICS で VSAM レコード・レベル共用 (RLS) をサポートするかどうかを指定します。[RLS システム 初期設定パラメーター](#)を参照してください。

## RRMS

**RRMS** システム 初期設定パラメーターは、リカバリー可能リソース管理サービス (RRMS) を持つリソース・マネージャーとして CICS を登録するかどうかを指定します。[RRMS システム 初期設定パラメーター](#)を参照してください。

## SDTRAN

**SDTRAN** システム 初期設定パラメーターは、通常および即時シャットダウンの最初に開始されるシャットダウン・トランザクションの名前を指定します。[SDTRAN システム 初期設定パラメーター](#)を参照してください。

## SECPRF

**SECPRF** システム 初期設定パラメーターは、外部セキュリティ・マネージャーに対する許可要求に含まれるリソース名に、CICS で接頭部を付けるかどうかを指定します。[SECPRF システム 初期設定パラメーター](#)を参照してください。

## SPOOL

**SPOOL** システム 初期設定パラメーターは、システム・スプーリング・インターフェースが必要かどうかを指定します。[SPOOL システム 初期設定パラメーター](#)を参照してください。

## SRBSVC

**SRBSVC** システム 初期設定パラメーターは、CICS タイプ 6 SVC に割り当てた番号を指定します。[SRBSVC システム 初期設定パラメーター](#)を参照してください。

## SRVERCP

**SRVERCP** システム 初期設定パラメーターは、DFHCNV データ変換テーブルで使用するデフォルトのサーバー・コード・ページを指定します (ただし、DFHCNV マクロの SRVERCP パラメーターが SYSDEF に設定されている場合のみ)。[SRVERCP システム 初期設定パラメーター](#)を参照してください。

## STARTER

**STARTER** システム 初期設定パラメーターは、(接尾部が \$ および # の) スターター・システム・モジュールの生成を許可して、さまざまな MNOTE を抑止するかどうかを指定します。[STARTER システム 初期設定パラメーター](#)を参照してください。

## STATRCD

**STATRCD** システム 初期設定パラメーターは、CICS 初期設定時における間隔統計の記録の状況を指定します。[STATRCD システム 初期設定パラメーター](#)を参照してください。

## STGPROT

**STGPROT** システム 初期設定パラメーターは、CICS 領域でストレージ保護を行うかどうかを指定します。[STGPROT システム 初期設定パラメーター](#)を参照してください。

## TBEXITS

**TBEXITS** システム 初期設定パラメーターは、緊急時再始動のバックアウト処理の際に使用するバックアウト出口プログラムの名前を指定します。[TBEXITS システム 初期設定パラメーター](#)を参照してください。

## TCT

**TCT** システム 初期設定パラメーターは、ロードする端末管理テーブルを指定します (存在する場合)。[TCT システム 初期設定パラメーター](#)を参照してください。



## TD

**TD** システム 初期設定パラメーターは、区画内一時データ (TD) で使用される VSAM バッファーおよびストリングの数を指定します。[TD システム 初期設定パラメーター](#)を参照してください。

## TRANISO

**TRANISO** システム 初期設定パラメーターは **STGPROT** システム 初期設定パラメーターと共に使用されて、CICS 領域でトランザクション分離を行うかどうかを指定します。[TRANISO システム 初期設定パラメーター](#)を参照してください。

## TS

**TS** システム 初期設定パラメーターでは、一時記憶域のコールド・スタートを行うかどうかと、補助一時記憶域に使用する VSAM バッファーとストリングの数を指定します。[TS システム 初期設定パラメーター](#)を参照してください。

## WLMHEALTH

**WLMHEALTH** システム 初期設定パラメーターは、z/OS Workload Manager Health API (IWM4HLTH) 呼び出しで CICS が使用する時間間隔と正常性調整値を指定します。CICS は、この呼び出しを実行して z/OS WLM に CICS 領域の正常性状態を通知します。[WLMHEALTH](#) を参照してください。

## XAPPC

**XAPPC** システム 初期設定パラメーターは、APPC セッションを確立するときに RACF セッション・セキュリティを使用できるかどうかを指定します。[XAPPC システム 初期設定パラメーター](#)を参照してください。

## XDB2

**XDB2** システム 初期設定パラメーターは、CICS で DB2ENTRY セキュリティ検査を実行するかどうかを指定します。[XDB2 システム 初期設定パラメーター](#)を参照してください。

## DFHSIT マクロ・パラメーターの指定

DFHSIT マクロでほとんどのシステム 初期設定パラメーターを指定できます。

### 手順

- 最初に、次のマクロ・パラメーターをコーディングします。

**TYPE={CSECT|DSECT}**

このパラメーターは、生成する SIT のタイプを指定します。

#### CSECT

通常使用される正規の制御セクション。

#### DSECT

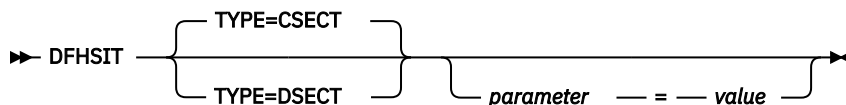
ダミー制御セクション。

以下に例を挙げます。

```
DFHSIT    TYPE=CSECT,      *
```

72 桁目のアスタリスク (\*) は任意の文字にすることができ、マクロが次の行に続くことを示します。

- システム 初期設定パラメーターをマクロに追加するには、次の構文を使用します。



ここで、*parameter* はコーディングするシステム 初期設定パラメーター、*value* は適切なパラメーター値です。パラメーターおよびキーワードは大文字でコーディングする必要があります (ただし、大/小文字が重要であるパラメーターは除きます)。例えば、z/OS UNIX ディレクトリーの名前を指定するパラメーターは、大文字小文字が区別されます。システム 初期設定パラメーターの詳細については、『[システム 初期設定パラメーターの説明と要約](#)』を参照してください。

以下に例を示します。

```
ADI=30, *  
AIBRIDGE=AUTO, *
```

```
AIEXIT=DFHZATDX, *  
AILDELAY=0, *
```

3. 次のステートメントを使用して、マクロ・パラメーターを終了します。

```
END DFHSITBA
```

## 例

提供されているサンプル・システム初期設定テーブルについて詳しくは、[CICS 管理テーブルでのリソースの定義](#)を参照してください。

## CICS リソース・テーブルおよびモジュールのキーワードの定義

### このタスクについて

以下の表では、次のような CICS リソースのシステム初期設定のキーワードを示します。

- 接尾部のオプションがある
- resource=NO をコーディングすると、ダミーのプログラムまたはテーブルがロードされる
- 個別にコールド・スタートを実行できる

表 18. 接尾部、ダミーのロード・モジュール、または COLD オプションに関するリソースの要約

DFHSIT キーワード 1	デフォルト 2	接尾部 3	ダミー 4	COLD スタート 5
BMS	FULL	-	-	COLD
CLT	-	xx	-	-
DIP	NO	-	プログラム	-
FCT	YES	xx	-	-
ICP	-	-	-	COLD
MCT	NO	xx	6	-
PDIR	NO	xx	-	-
PLTPI	NO	xx	-	-
PLTSD	NO	xx	-	-
RST	NO	xx	-	-
SRT	YES	xx	-	-
TCT	YES	xx	テーブル	-
TS	-	-	-	COLD
TST	NO	xx	-	-
XLT	NO	xx	-	-

### 注:

1. DFHSIT キーワードに 2 つ以上の値が指定されている場合、これらの値は括弧で囲まれている必要があります。例えば、BMS=(FULL,COLD) のようにします。
2. 「デフォルト」列は、DFHSIT マクロのキーワードのデフォルト値を示します。

接尾部のオプションを持つキーワードでは、SIT で YES とコーディングすると、接尾部のないバージョンのテーブルまたはプログラムがロードされます。例えば、TCT=YES と指定すると、DFHTCT というテ



ーブルがロードされます。**keyword=**, または **keyword=YES** を指定することで、CICS 始動時に接尾部のないモジュールやテーブルを選択することもできます。例えば、次のようにコーディングした場合、

```
FCT=, or FCT=YES
```

DFHFCT に空白が追加され、接尾部のないこれらの名前が初期設定時に使用されます。

PARM、SYSIN、または CONSOLE で **keyword=**, をシステム初期設定パラメーターとして指定した場合の結果は、必ずしも DFHSIT マクロの場合と一緒にではありません。例えば、DFHSIT マクロのコーディングで TST=, (または全体を省略) を指定すると TST=NO と解釈されますが、他の 3 つのメソッドでは TST=, は TST=YES と解釈されます。

3. 「接尾部」列は、接尾部をコーディングできるかどうかを示します。(xx は、接尾部をコーディングできることを表します。)

接尾部には任意の 1 文字または 2 文字を指定できますが、DY は指定できません。また、NO を接尾部として使用することもできません。

接尾部をコーディングすると、標準の名前にその接尾部が追加されたテーブルまたはプログラム がロードされます。

4. 「ダミー」列は、NO をコーディングした場合に、ダミー・バージョンのプログラムまたはテーブルがロードされるかどうかを表します。テーブルに関連付けられたオペランドに対して NO をコーディングすると、ダミーのプログラムがロードされるケースがあります。このオプションの影響について、詳しくは 124 ページの『CICS プログラムとテーブルのバージョンの選択』を参照してください。

5. 「COLD スタート」列は、リソースを強制的に COLD スタートさせることができるかどうかを表します。(COLD は、そのリソースを個別にコールド・スタートできることを表します。)TST と コールド・スタートは、TST に変更を加えた場合に、必ず一時記憶域またはシステム全体に対してコールド・スタートが実行されるようにします。

COLD がコーディングされている場合、これをオーバーライドできるのは、システム初期設定パラメーターとして START=(...,ALL) を コーディングした場合のみです。このオプションの詳細については、ALL のページを参照してください。

CICS テーブルおよびプログラムの選択について詳しくは、124 ページの『CICS プログラムとテーブルのバージョンの選択』を参照してください。

6. MCT=NO をコーディングすると、CICS モニター・ドメインは、デフォルトのモニター管理テーブルを動的に作成します。これにより、モニターが有効でモニター・クラスがアクティブになっている場合、デフォルトのモニター管理テーブルの項目が常に使用できるようになります。

## デフォルトのシステム初期設定テーブル

このデフォルトの SIT は、CICSTS56.CICS.SDFHAUTH にあり、DFHSIT\$\$ という名前のそのソースは CICSTS56.CICS.SDFHSAMP にあります。

デフォルトのシステム 初期設定テーブルのアセンブルに使用されるマクロ・ソース・ステートメントのパラメーターと値の一覧を 114 ページの表 19 に示します。

表 19. DFHSIT、事前生成されたデフォルトのシステム初期設定テーブル		
パラメーター	デフォルト値	説明
ADI	30	XRF(B) - 代替遅延間隔
AIBRIDGE	AUTO	ブリッジ自動インストール URM
AICONS	NO	MVS CONSOLES の自動インストールは行われません
AIEXIT	DFHZATDX	自動インストール・ユーザー・プログラム名
AILDELAY	0	AI 端末の削除遅延期間
AIQMAX	100	AI のキューに入れられた端末の最大数
AIRDELAY	700	AI 端末の再始動遅延期間

表 19. DFHSIT、事前生成されたデフォルトのシステム初期設定テーブル (続き)

パラメーター	デフォルト値	説明
AKPFREQ	4000	活動キーポイント頻度
APPLID	DBDCCICS	z/OS Communications Server の APPL ID
AUTCONN	0	自動接続の遅延
AUTODST	NO	Language Environment® の自動ストレージ調整
AUTORESETTIME	IMMEDIATE	時刻の同期
AUXTR	OFF	補助トレース・オプション
AUXTRSW	NO	補助トレース自動切り替え機能
BMS	FULL、UNALIGN、 DDS	基本マッピング・サポートのオプション
BRMAXKEEPTIME	86400	ブリッジの最大保持時間
CICSSVC	216	CICS SVC 番号
CILOCK	NO	読み取りの更新後は CI ロックを保持しません
CLINTCP	437	デフォルトのクライアント・コード・ページ
CLSDSTP	NOTIFY	ISSUE PASS コマンドの通知
CLT		コマンド・リスト・テーブルのオプションまたは接 尾部
CMDPROT	YES	Exec storage command checking (Exec ストレ ージ・コマンドの検証)
CMDSEC	ASIS	API コマンド・セキュリティー検査
CONFDATA	SHOW	ダンプとトレースに機密データを表示します
CONFTXT	NO	z/OS Communications Server によるユーザー・デー タのトレースを抑制しません
CPSMCONN	NO	CICSplex SM に接続しません
CRLPROFILE		証明書失効リストへのアクセスを CICS に許可する プロファイルの名前
CSDACC	READWRITE	CSD アクセス
CSDBKUP	STATIC	CSD (STATIC または DYNAMIC) のバックアップ・タ イプ
CSDBUFND		CSD のデータ・バッファ数
CSDBUFNI		CSD の索引バッファ数
CSDDISP		動的割り振り用の CSD ファイル属性指定
CSDDSN		動的割り振り用の CSD データ・セット名
CSDFRLOG	NO	CSD 順方向リカバリーのジャーナル ID
CSDINTEG	UNCOMMITTED	読み取り保全性 = アンコミット
CSDJID	NO	CSD 自動ジャーナリングのジャーナル ID
CSDLSRNO	1	CSD の VSAM LSR プール番号

表 19. DFHSIT、事前生成されたデフォルトのシステム初期設定テーブル (続き)

パラメーター	デフォルト値	説明
CSDRECOV	なし	CSD リカバリー可能ファイル・オプション
CSDRLS	NO	従来の VSAM を使用します
CSDSTRNO	6	CSD のストリング数
CWAKEY	USER	CWA ストレージ・キー
DAE	NO	SDUMPS は DAE によって抑制されません
DATFORM	MMDDYY	CSA 日付形式
DB2CONN	NO	CICS の始動時に Db2 に接続しません
DBCTLCON	NO	CICS の始動時に DBCTL に接続しません
DEBUGTOOL	NO	デバッグ・ツールにアクセスしません
DFLTUSER	CICSUSER	デフォルト・ユーザー
DIP	NO	バッチ・データ交換プログラム
DISMACP	YES	マクロ・プログラムを無効にします
DOCCODEPAGE	037	デフォルトのホスト・コード・ページ
DSALIM	5M	16 MB 境界より下の DSA の上限
DSHIPIDL	020000	Delete shipped idle time (シップの削除アイドル時間)
DSHIPINT	120000	Delete shipped interval (シップの削除インターバル)
DSRTPGM	なし	分散ルーティング・プログラム
DTRPGM	DFHDYP	動的ルーティング・プログラム
DTRTRAN	CRTX	デフォルトの動的トランザクション・ルーティングのトランザクション ID
DUMP	YES	ダンプ・オプション
DUMPDS	AUTO	CICS ダンプ・データ・セットのオープン・オプション
DUMPSW	NO	ダンプ・データ・セットの自動切り替えオプション
DURETRY	30	SDUMP 合計再試行時間 (秒単位)
EDSALIM	800M	31 ビット・ストレージ内の DSA の上限
EODI	E0	順次装置のデータ終了標識
ESMEXITS	NOINSTLN	外部セキュリティ・マネージャー出口
FCT	NO	ファイル管理テーブルのオプションまたは接尾部
FCQRONLY	YES	QR TCB でのスレッド・セーフ FC の実行
FEPI	NO	フロントエンド・プログラミング・インターフェース
FLDSEP	' ' (4 つのブランク)	フィールドの末尾の分離文字
FLDSTRT	' ' (1 つのブランク)	組み込み関数のフィールド開始文字

表 19. DFHSIT、事前生成されたデフォルトのシステム初期設定テーブル (続き)

パラメーター	デフォルト値	説明
FORCEQR	NO	QR でスレッド・セーフ・プログラムを強制しません
FSSTAFF	NO	機能シップ START アフィニティー・オプション
FTIMEOUT	30	30 秒のファイル・タイムアウト
GMTEXT	'WELCOME TO CICS'	「Good morning」メッセージ・テキスト
GMTRAN	CSGM	初期トランザクション
GNTRAN	NO	サインオフ・トランザクション
GRNAME		CICS TOR の汎用リソース名
GRPLIST	DFHLIST	始動時の CSD グループのリスト名
GTFTTR	OFF	GTF トレース・オプション
HPO	NO	z/OS Communications Server の高性能オプション (HPO)
HTTPSERVERHDR	YES	HTTP 応答のサーバー・ヘッダー内の値セット
HTTPUSRAGENTHDR	YES	HTTP 要求のユーザー・エージェント・ヘッダー内の値セット
ICP		間隔制御プログラム開始オプション
ICV	1000	領域終了間隔 (ミリ秒)
ICVR	2000	ランナウェイ・タスク間隔 (ミリ秒)
ICVTSD	0	端末スキャン遅延間隔
INITPARM		プログラムの初期設定パラメーター
INTTR	ON	CICS 内部トレース・オプション
IRCSTRT	NO	領域間通信の開始
ISC	NO	システム間連絡オプション
JESDI	30	XRF 代替の JES 遅延間隔
JVMCCSIZE	24M	共用クラス・キャッシュ・サイズ
JVMCCSTART	AUTO	必要な場合に共用クラス・キャッシュを開始します
JVMPROFILEDIR	/usr/lpp/cicsts /cicsts56/JVMProfiles	JVM プロファイル・ディレクトリー
KERBEROSUSER		Kerberos ユーザー ID = CICS 領域ユーザー ID
KEYRING		SSL サポートで使用する鍵リング
LGDFINT	5	ログ・マネージャーのログ延期間隔
LGNMSG	NO	z/OS Communications Server のログオン・データを取り出します
LLACOPY	YES	MVS LLACOPY サポートを使用します
LOCALCCSID	037	ローカル領域のデフォルト CCSID

表 19. DFHSIT、事前生成されたデフォルトのシステム初期設定テーブル (続き)

パラメーター	デフォルト値	説明
LPA	NO	CICS/ユーザー・モジュールの LPA 使用オプション
MAXOPENTCBS	算出値	CICS が L8 および L9 モードの TCB のプール内で作成できるオープン TCB の最大数
MAXSOCKETS	65535	IP ソケットの最大数
MAXSSLTCBS	32	SSL TCB 数の制限
MAXXPTCBS	算出値	CICS 領域内に同時に存在できる X8 および X9 のオープン TCB の最大数
MCT	NO	モニター管理テーブルのオプションまたは接尾部
MINTLSLEVEL	TLS12	TLS の暗号化のレベル
MN	OFF	CICS モニター領域
MNCONV	NO	対話記録のモニター・オプション
MNEXC	OFF	例外クラスのモニター・オプション
MNFREQ	0	頻度期間のモニター
MNIDN	OFF	ID クラスのモニター・オプション
MNPER	OFF	パフォーマンス・クラスのモニター・オプション
MNRES	OFF	リソース・クラスのモニター・オプション
MNSYNC	NO	同期点記録のモニター・オプション
MNTIME	GMT	タイム・スタンプのモニター (GMT または LOCAL)
MQCONN	NO	始動時に MQ に接続しません
MROBTCH	1	バッチ処理される MRO 要求数
MROFSE	NO	長期実行ミラーの存続期間を拡張します
MROLRM	NO	長期実行ミラー・タスク・オプション
MSGCASE	MIXED	大/小文字混合の CICS メッセージ
MSGLVL	1	システム・コンソールの MSG レベル・オプション
MXT	250	CICS 内の最大タスク数
NATLANG	E	各国語のリスト
NCPLDFT	DFHNC001	名前付きカウンターのデフォルト・プール名
NISTSP800131A	NOCHECK	CICS 領域が NIST 800-131A 標準への準拠性を検査するかどうか
NONRLSRECOV	VSAMCAT	非 RLS ファイルの復旧オプションの場所を選択します
NQRNL	NO	z/OS グローバル・リソースの逐次化による RNL 処理
OPERTIM	120	オペレーター宛メッセージのタイムアウト (秒)
OPNDLIM	10	OPNDST/CLSDST 要求限度

表 19. DFHSIT、事前生成されたデフォルトのシステム初期設定テーブル (続き)

パラメーター	デフォルト値	説明
PARMERR	INTERACT	システム 初期設定パラメーター・エラー・オプション
PDI	30	1 次遅延間隔 - XRF アクティブ
PDIR	NO	DL/I PSB ディレクトリーのオプションまたは 接尾部
PGAICTLG	MODIFY	PG 自動インストール・カタログ状態
PGAIEXIT	DFHPGADX	PG 自動インストール出口プログラム
PGAIPGM	INACTIVE	PG 自動インストール状態
PGCHAIN		BMS CHAIN コマンド
PGCOPY		BMS COPY コマンド
PGPURGE		BMS PURGE コマンド
PGRET		BMS RETURN コマンド
PLTPI	NO	プログラム・リスト・テーブルの PI オプションまたは接尾部
PLTPISEC	なし	PI プログラムでの PLT セキュリティー 検査なし
PLTPIUSR		PLT PI ユーザー ID = CICS 領域ユーザー ID
PLTSD	NO	プログラム・リスト・テーブル の SD オプションまたは接尾部
PRGDLAY	0	BMS パージ遅延間隔
PRINT	NO	プリント・キー・オプション
PRTYAGE	1000	ディスパッチャー 優先順位繰り上げ値
PSBCHK	NO	PSB リソース検査必須
PSDINT	0	持続セッション遅延間隔
PSTYPE	SNPS	z/OS Communications Server 単一ノード持続セッション
PVDELAY	30	LUIT テーブルのタイムアウト 値
QUIESTIM	240	静止要求のタイムアウト値
RACFSYNC	YES	タイプ 71 ENF イベントの listen
RAMAX	256	RECEIVE ANY の最大入出力域
RAPOOL	50	最大 RECEIVE ANY 要求パラメーター・リスト
RENTPGM	PROTECT	再入可能プログラム書き込み保護
RESP	FME	論理装置 (LU) の応答タイプ
RESSEC	ASIS	リソース・セキュリティ 検査
RLS	NO	RLS オプション
RLSTOLSR	NO	LSRPOOL ビルド時計算の RLS ファイル
RMTRAN	CSGM	XRF 代替リカバリー・トランザクション



表 19. DFHSIT、事前生成されたデフォルトのシステム初期設定テーブル (続き)

パラメーター	デフォルト値	説明
RRMS	NO	リカバリー可能リソース管理サービス
RST	NO	リカバリー・サービス・テーブル (XRF-DBCTL)
RSTSIGNOFF	NOFORCE	XRF - テークオーバー後の再サインオン
RSTSIGNTIME	500	XRF - サインオフ・タイムアウト値
RUWAPOL	NO	Language Environment のストレージ・プールの割り振り
SDTRAN	CESD	シャットダウン・トランザクション
SEC	YES	外部セキュリティー・マネージャー・オプション
SECPRFX	NO	セキュリティー接頭部
SKRPA1		SKR PA1 PAGE RETRIEVAL CMD
SKRPA2		SKR PA2 PAGE RETRIEVAL CMD
SKRPA3		SKR PA3 PAGE RETRIEVAL CMD
SKRPF1		SKR PF1 PAGE RETRIEVAL CMD
SKRPF2		SKR PF2 PAGE RETRIEVAL CMD
SKRPF3		SKR PF3 PAGE RETRIEVAL CMD
SKRPF4		SKR PF4 PAGE RETRIEVAL CMD
SKRPF5		SKR PF5 PAGE RETRIEVAL CMD
SKRPF6		SKR PF6 PAGE RETRIEVAL CMD
SKRPF7		SKR PF7 PAGE RETRIEVAL CMD
SKRPF8		SKR PF8 PAGE RETRIEVAL CMD
SKRPF9		SKR PF9 PAGE RETRIEVAL CMD
SKRPF10		SKR PF10 PAGE RETRIEVAL CMD
SKRPF11		SKR PF11 PAGE RETRIEVAL CMD
SKRPF12		SKR PF12 PAGE RETRIEVAL CMD
SKRPF13		SKR PF13 PAGE RETRIEVAL CMD
SKRPF14		SKR PF14 PAGE RETRIEVAL CMD
SKRPF15		SKR PF15 PAGE RETRIEVAL CMD
SKRPF16		SKR PF16 PAGE RETRIEVAL CMD
SKRPF17		SKR PF17 PAGE RETRIEVAL CMD
SKRPF18		SKR PF18 PAGE RETRIEVAL CMD
SKRPF19		SKR PF19 PAGE RETRIEVAL CMD
SKRPF20		SKR PF20 PAGE RETRIEVAL CMD
SKRPF21		SKR PF21 PAGE RETRIEVAL CMD
SKRPF22		SKR PF22 PAGE RETRIEVAL CMD
SKRPF23		SKR PF23 PAGE RETRIEVAL CMD

表 19. DFHSIT、事前生成されたデフォルトのシステム初期設定テーブル (続き)

パラメーター	デフォルト値	説明
SKRPF24		SKR PF24 PAGE RETRIEVAL CMD
SNPRESET	UNIQUE	事前設定ユーザー ID 端末がユーザー ID に関連付けられた単一 ACEE を共用するか、各端末が固有の ACEE を使用するかを指定
SNSCOPE	なし	ユーザー ID ごとの複数の CICS セッション
SOTUNING	YES	制約のないリソース要求から CICS を保護するために HTTP 接続のパフォーマンス・チューニングを行うかどうか。
SPCTR	(1,2)	必要な特殊トレースのレベル
SPOOL	NO	システム・スプーリング・インターフェース・オプション
SRBSVC	215	HPO タイプ 6 SVC 番号
SRT	1\$	システム・リカバリー・テーブルのオプションまたは接尾部
SRVERCP	037	DFHCNV データ変換テーブルで使用されるデフォルトのサーバー・コード・ページ (DFHCNV マクロの SRVERCP パラメーターが SYSDEF に設定されている場合のみ)
SSLCACHE	CICS	SSL セッション ID キャッシュ
SSLDELAY	600	SSL タイムアウト値
START	AUTO	CICS システム 初期設定オプション
STARTER	YES	スターター (\$ および #) 接尾部オプション 注: デフォルトは NO ですが、ここではパラメーターを YES に設定し、SIT で正しくアセンブルできるようにする必要があります。
STATEOD	0	統計終了時刻
STATINT	010000	統計間隔時間
STATRCD	OFF	統計記録状況
STGPROT	YES	ストレージ保護機能
STGRCVY	NO	ストレージ・リカバリー・オプション
STNTR	1	必要な標準トレースのレベル
SUBTSKS	0	並行モード TCB 数
SUFFIX	\$\$	この SIT の接尾部
SYDUMAX	999	取得する SYSDUMPS の数
SYSIDNT	CICS	ローカル・システム ID
SYSTR	ON	主システム・トレース・フラグ
TAKEOVR	MANUAL	XRF 代替テークオーバー・オプション
TBEXITS		バックアウト出口プログラム

表 19. DFHSIT、事前生成されたデフォルトのシステム初期設定テーブル (続き)

パラメーター	デフォルト値	説明
TCP	YES	端末管理プログラムのオプションまたは接尾部
TCPIP	YES	TCP/IP サポート
TCSACTN	なし	TC シャットダウン・アクション
TCSWAIT	4	TC シャットダウン待機
TCT	NO	端末管理テーブルのオプションまたは接尾部
TCTUAKEY	USER	TCT ユーザー域のストレージ・キー
TCTUALOC	ANY	TCT ユーザー域 ANY
TD	(3,3)	一時データ・バッファおよびストリング
TDINTRA	NOEMPTY	一時データ・キューの初期状態
TRANISO	NO	Transaction Isolation (トランザクション分離)
TRAP	OFF	F.E. グローバル・トラップ出口オプション
TRDUMAX	999	取得する TRANDUMPS の数
TRTABSZ	12288	内部トレース・テーブル・サイズ (1K バイト単位)
TRTRANSZ	1024	トランザクション・ダンプ・トレース・テーブル・サイズ
TRTRANTY	TRAN	トランザクション・ダンプ・トレース・オプション
TS	(3,3)	一時記憶域バッファおよびストリング
TSMMAINLIMIT	64M	TS メイン・キュー用のストレージの上限
TST	NO	一時記憶域テーブルのオプションまたは接尾部
UOWNETQL		NETUOWID の修飾子
USERTR	ON	マスター・ユーザー・トレース・フラグ
USRDELAY	30	ユーザー・ディレクトリー項目のタイムアウト値
USSCONFIG	/var/cicsts/dfhconfig	z/OS UNIX 上の CICS 構成ファイルのルート・ディレクトリーの名前とパス。
USSHOME	/usr/lpp/cicsts/cicsts56	z/OS UNIX 上の CICS ファイルのルート・ディレクトリーの名前とパス。
VTAM	YES	z/OS Communications Server アクセス方式オプション
VTPREFIX	\	クライアント仮想端末の接頭部
WEBDELAY	(5,60)	Web タイマーの値
WLMHEALTH	(20,25)	z/OS WLM に CICS 領域の正常性の状態を通知するために CICS が z/OS WLM Health API (IWM4HLTH) 呼び出しで使用するパラメーター
WRKAREA	512	共通作業域 (CWA) のサイズ (バイト単位)
XAPPC	NO	RACF クラス APPCLU が必須
XCFGROUP	DFHIR000	MRO 通信に使用する XCF グループ

表 19. DFHSIT、事前生成されたデフォルトのシステム初期設定テーブル (続き)

パラメーター	デフォルト値	説明
XCMD	YES	SPI で RACF 検査にデフォルト名を使用
XDB2	NO	DB2ENTRY リソースのセキュリティー検査
XDCT	YES	一時データ・キューのセキュリティー検査
XFCT	YES	ファイルのセキュリティー検査
XHFS	YES	z/OS UNIX ファイルのセキュリティー検査
XJCT	YES	ジャーナルのセキュリティー検査
XLT	NO	トランザクション・リスト・テーブルのオプションまたは接尾部
XPCT	YES	開始済みトランザクションのセキュリティー検査
XPPT	YES	プログラムのセキュリティー検査
XPSB	YES	DL/I PSB のセキュリティー検査
XPTKT	YES	パスチケットのセキュリティー検査
XRES	YES	XRES セキュリティーの対象となるリソースの検査では、RACF 検査のデフォルト名を使用します。 XRES セキュリティー検査の対象となるリソースのリストについては、 <a href="#">リソースおよびコマンドの検査の相互参照</a> を参照してください。
XRF	NO	拡張回復機能 (XRF) オプション
XTRAN	YES	トランザクション接続のセキュリティー検査
XTST	YES	一時記憶域キューのセキュリティー検査
XUSER	YES	実行する代理ユーザー検査

## SIT のアセンブル

DFHSIT マクロをコーディングした場合は、SIT をアセンブルする必要があります。

### このタスクについて

#### 手順

1. テーブルを APF 許可ライブラリー (CICSTS56.CICS.SDFHAUTH など) にアセンブルしてリンク・エディットします。
2. そのライブラリーを CICS 始動ジョブ・ストリームの STEPLIB 連結に組み込みます。

#### タスクの結果

SIT ソースでシステム 初期設定パラメーターをコーディングしたのに、そのパラメーターのキーワードを CICS 提供バージョンの DFHSIT マクロで定義しないと、アセンブリーから以下のような IEV017 警告メッセージが出されます。

```
IEV017  ** WARNING **  UNDEFINED KEYWORD PARAM. DEFAULT TO POSITIONAL, INCLUDING KW  --  OPENC/aaaaaaa
```

## 次のタスク

CICS 管理テーブルのアセンブルとリンク・エディットや、CICS マクロを記述するための構文表記法の説明については、[CICS 管理テーブルでのリソースの定義](#)を参照してください。

## CICS プログラムとテーブルのバージョンの選択

CICS プログラムは、通常、関連する CICS 機能モジュール (例えば、端末管理プログラム) のグループによって構成されます。

### このタスクについて

ほとんどの CICS プログラムは、CICS に用意されている 1 つのバージョンしか使用できません。しかし、一部の CICS プログラムは、複数のバージョン (例えば、サービス・レベルが異なるものなど) を作成できます。特定のバージョンのプログラムを選択するには、CICS 始動 JCL にそのバージョンを含むロード・ライブラリーを指定します。一方、基本マッピング・サポート (BMS) スイートでは、必要な機能レベルを明示的に選択して、さまざまなバージョンの中から選択することができます。

プログラムが必要でないことを指定することもできます (詳しくは、[124 ページの『不要なプログラムの除外』](#)を参照してください)。

システム初期設定パラメーターをコーディングして、このセクションと [124 ページの『不要なプログラムの除外』](#)で言及しているプログラムにのみ、これらの方法を使用することができます。

### 明示的な機能レベルを使用したプログラムの選択

明示的な機能レベルを使用して、BMS の一連のプログラムを選択できます。

システム初期設定パラメーター BMS で BMS 要件を指定するときには、3 つのバージョンの中からいずれかを選択できます。MINIMUM、STANDARD、または FULL のパラメーター・オプションを使用して、BMS の機能レベルを選択します。選択した機能レベルから、必要な一連のプログラムがシステム初期設定プログラムによってロードされます。

### 不要なプログラムの除外

不要なプログラムを除外する方法は、3 つあります。

以下を指定して、プログラムを除外できます。

1. programname=NO
2. tablename=NO
3. function=NO

#### **programname=NO の指定**

システム初期設定パラメーター *programname=NO* (例えば *DIP=NO*) をコーディングすると、指定した管理プログラムを CICS システム初期設定の際に除外できます。

*programname=NO* をコーディングして、以下のプログラムを除外することができます。

- バッチ・データ交換プログラム (DIP)
- 端末管理プログラム (TCP)

注: DIP のケースでは、配布テープに接尾部 **DY** 付きで含まれるダミー・バージョンの管理プログラムを取得します。

#### **プログラム管理テーブルでの tablename=NO の指定**

すべての CICS プログラムがその SIT に *programname* パラメーターを持つわけではありません。

システム初期設定パラメーターを使用する代わりに、関連するテーブルで **NO** をコーディングする方法があります。これは、*program name* パラメーターに対して **NO** をコーディングした場合と同じ効果があり、関連付けられた CICS プログラムは、ダミー・プログラムをロードするかその他の手法で、システム初期設定の実行時に除外されます。

システム・リカバリー・テーブル (SRT) をこの方法で使うことができ、関連付けられたシステム・リカバリー・プログラム (SRP) が除外されます。

### ダミー TCT、DFHTCTDY

tablename=NO を指定できる特殊なケースがあります。ただし、これによってダミー端末管理プログラムはロードされません。オンライン・リソース定義を使用していて、すべての端末リソース定義が CSD に含まれている場合、TCT=NO を指定します。

TCT=NO を指定すると、CICS は DFHTCTDY という名前のダミー TCT をロードします。この名前の事前生成 CICSTS56.CICS ダミー・テーブルが .SDFHLOAD に提供されており、DFHTCTDY のソース CICSTS56.CICS ステートメントが .SDFHSAMP に提供されています。TCT=NO を指定すると、生成されたこの名前のテーブルは、CICS を始動する際に DFHRPL 連結のライブラリーで使用可能になっている必要があります。

ダミー TCT は、z/OS Communications Server 端末を使用していて、端末定義の保管に CSD 使用している場合に、必要な CICS および z/OS Communications Server の制御ブロックのみを提供します。z/OS Communications Server 端末は、CSD バッチ・ユーティリティー・プログラム、DFHCSDUP の RDO トランザクションである CEDA、または DEFINE コマンドを使用して定義します。

### function=NO の指定

システム初期設定パラメーターとして *function=NO* をコーディングした場合、CICS システム初期化時に指定の機能に関連付けられた管理プログラムが除外されます。

この方法で、システム間連絡 (ISC)、3270 印刷要求機能、およびシステム・スプーリング・インターフェースを除外できます。

## システム初期設定パラメーターの処理

このセクションでは、CICS システム初期設定の処理について説明します。

### このタスクについて

以下が含まれています。

1. システム初期設定パラメーターを指定するプロセス、およびこのプロセスの CICS パラメーター管理者ドメインの役割についての簡単な紹介。
2. CICS が特別なシステム初期設定キーワードを使用する方法についての説明。
3. クラスの始動と再始動、およびそれらを制御する方法に関する説明。

## CICS に対するシステム初期設定パラメーターの提供

CICS パラメーター・マネージャー・ドメインは、初期設定プロセスの開始時にシステム初期設定テーブル (SIT) をロードします。

### このタスクについて

SIT= システム初期設定パラメーターで DFHSITxx ロード・モジュール (xx は接尾部) の接尾部をコーディングすることで、ニーズに適した CICS の特性を定義する SIT を指定します。SIT の接尾部の指定に失敗すると、CICS は接尾部のないモジュールをロードしようとします。

システム初期設定パラメーターを始動ジョブ・ストリームで指定するか、システム・コンソールから指定することで、CICS 初期設定の開始時に多くのシステム初期設定パラメーターを動的に変更できます。また、SIT にコーディングできないシステム初期設定パラメーターもいくつかあり、それらは起動時にのみ指定できます。システム初期設定パラメーターを起動時に指定するには、次の 3 つの方法があります。

1. EXEC PGM=DFHSIP ステートメントの PARM パラメーターにより指定。
2. 始動ジョブ・ストリームに定義されている SYSIN データ・セットにより指定。
3. システム・オペレーターのコンソールを介して指定。

これらの方法のいずれか 1 つを使用することも、2 つ、あるいは 3 つすべてを使用することもできます。ただし、CICS はこれらの 3 つの入力データのソースを、次のように厳密な順序で処理します。

1. PARM パラメーター



2. SYSIN データ・セット (ただし、SYSIN が PARM パラメーターでコーディングされている 場合のみ。  
SYSIN を参照)
3. コンソール (ただし、CONSOLE が PARM パラメーターまたは SYSIN データ・セットでコーディングさ  
れている場合のみ。126 ページの『CONSOLE (CN)』を参照)

注: 同じメディア、あるいは別のメディアからシステム 初期設定パラメーターを重複して指定した場合、CICS は最後に読み取った方を使用します。例えば、PARM パラメーターで MCT=1\$ を指定し、SYSIN データ・セットで MCT=2\$ を指定し、最後にコンソールを通じて MCT=3\$ を入力した場合、CICS は DFHMCT3\$ をロードします。

## システム初期設定の制御キーワードの使用

起動時に SYSIN、CONSOLE、および END の各制御キーワードを使用して、CICS がシステム 初期設定パラメーターを読み取る方法を制御することができます。

### このタスクについて

これら特殊なキーワードの目的と、それらをコーディングする 場所について、以下で説明します。

#### SYSIN (SI)

このキーワードは、CICS に SYSIN データ・セットから初期設定パラメーターを読み取るように指示します。

**SYSIN をコーディングする 場所:** SYSIN (または SI) は、EXEC PGM=DFHSIP ステートメントの PARM パラメーターでのみコーディングできます。キーワードを指定できるのは 1 回のみで、PARM パラメーターの末尾にある必要があります。CICS は、すべての PARM パラメーターのスキャンを終了するまで、または PARM パラメーターの末尾より前に .END にたどり着くまで、SYSIN を読み取りません。

例:

```
//stepname EXEC PGM=DFHSIP,PARM='SIT=6$,SYSIN,.END'  
//stepname EXEC PGM=DFHSIP,PARM='SIT=6$,DLI=YES,SYSIN,.END'
```

#### CONSOLE (CN)

このキーワードは、CICS にコンソールから初期設定パラメーターを読み取るように指示します。CICS がコンソールからパラメーターを読み取る準備ができると、メッセージ DFHPA1104 を出します。

**CONSOLE をコーディングする 場所:** CONSOLE (または CN) は、EXEC PGM=DFHSIP ステートメントの PARM パラメーター内、または SYSIN データ・セット内にコーディングできます。このキーワードは、PARM パラメーターの末尾、または SYSIN データ・セット内に指定できますが、いずれかの 1 か所でのみ指定できます。

CONSOLE (または CN) を PARM パラメーターにコーディングし、PARM にも SYSIN キーワードが含まれる場合、CICS は SYSIN データ・セットの読み取りと処理が完了するまで、コンソールからのパラメーターの読み取りを開始しません。同様に、CONSOLE キーワードを SYSIN データ・セットのどこに配置した場合でも、CICS は SYSIN データ・セットの読み取りと処理が完了するまで、コンソールからのパラメーターの読み取りを開始しません。

例:

```
//stepname EXEC PGM=DFHSIP,PARM='SIT6$,CONSOLE,.END'  
//stepname EXEC PGM=DFHSIP,PARM='CONSOLE,SYSIN,.END'  
//stepname EXEC PGM=DFHSIP,PARM='SIT=6$,CN,SI,.END'
```

SYSIN (または SI) と CONSOLE (または CN) の両方が PARM パラメーターのキーワードとして指定されている場合、それらがコーディングされている 順番は、後続に .END 以外のキーワードが指定されていない限り、意味はありません。

#### .END

このキーワードの意味は、状況によって異なります。

## PARM

.END キーワードの使用は、PARM パラメーターではオプションです。省略した場合、CICS はそれが PARM パラメーターの末尾にあるものと想定します。.END を PARM パラメーターにコーディングした場合、次の 2 つのうちいずれかの意味を持ちます。

1. 一緒に他の制御キーワードの 1 つまたは両方 (CONSOLE および/または SYSIN) をコーディングする場合、.END は PARM パラメーターの終了のみを意味します。

例えば、次のようになります。

```
//stepname EXEC PGM=DFHSSIP,PARM='SIT6$,SI,CN,.END'
```

2. .END を PARM パラメーター内で唯一の制御キーワードとしてコーディングする場合、これはすべてのシステム初期設定パラメーターの終了を意味し、CICS は初期設定プロセスを開始します。例えば、次のようになります。

```
//stepname EXEC PGM=DFHSSIP,PARM='SIT=6$,.END'
```

.END が PARM パラメーター内の最後の項目ではない場合、CICS は PARM パラメーターを切り捨て、.END キーワードの後に続くパラメーターは失われます。

## SYSIN

.END キーワードの使用は、SYSIN データ・セットではオプションです。省略した場合、CICS はそれが SYSIN の末尾にあるものと想定します。.END を SYSIN データ・セットにコーディングした場合、その意味は、次のように CONSOLE キーワードの使用方法によって異なります。

- CONSOLE 制御キーワードを PARM パラメーターまたは SYSIN データ・セットにコーディングした場合、.END は SYSIN データ・セットの終了のみを意味します。
- CONSOLE 制御キーワードを PARM パラメーターまたは SYSIN データ・セット内にコーディングしない場合、.END はすべての CICS システム初期設定パラメーターの終了を意味し、CICS は初期設定プロセスを開始します。

.END をコーディングし、それが SYSIN データ・セット内の最後の項目ではない場合、あるいは SYSIN レコードの末尾にない場合、.END の後に続く CICS 初期設定パラメーターは無視されます。初期設定パラメーターを不慮の損失から守るために、.END キーワードが SYSIN データ・セットの最後のレコードにあることと、同じ行に他の項目がないことを確認してください。(ただし、特定の CICS の実行で一部のシステム初期設定パラメーターを除外する場合、その実行の時のみ除外するパラメーターを .END ステートメントの後ろに配置することができます。)

SYSIN データ・セットにおける .END の使用の例を以下に示します。

```
//SYSIN DD *
* CICS system initialization parameters
SIT=6$,START=COLD,
PDIR=1$,          ( SUFFIX of PSB directory

.END
/*
```

## CONSOLE

コンソールを介した .END の意味は、新しいパラメーターを入力するのか、修正を入力するのかによって異なります。その 2 つの意味は次のとおりです。

1. ユーザーがメッセージ DFHPA1104 の応答として新しいパラメーターを指定している場合、.END はパラメーターの読み取りを終了し、CICS はロードした SIT に従って初期化を開始しますが、CICS がロードした SIT の値はユーザーが指定した値でオーバーライドされます。.END 制御キーワードを入力するまで、CICS はシステム初期設定パラメーターを指定するようにプロンプトを出し続けます。
2. PARMERR=INTERACT をコーディングしているときに、キーワードまたはキーワードに割り当てた値にパラメーター・エラーがあることを CICS が検出した場合、CICS はメッセージ DFHPA1912 または DFHPA1915 によって、エラーの修正を求めるプロンプトを出します。正しいキーワードまたは値を入力すると、初期設定は再開し、CICS は PARM パラメーター、または SYSIN データ・セットの処理を継続するか、CICS がエラーを検出した場所によっては、追加の

システム初期設定パラメーターをコンソールを介して指定するように求めます。エラーは修正できないものの、CICS が初期設定プロセスを継続するようにしたい場合、.END を入力してエラー修正フェーズを終了することができます。

### PARM パラメーターの処理

**PARM** パラメーターを EXEC PGM=DFHSIP ステートメントから省略した場合、CICS は SIT オーバーライドも他の初期設定パラメーターも存在しないと想定して、接尾部なしの DFHSIT というモジュールのロードを試みます。

通常、**PARM** パラメーターには、少なくとも SIT キーワードを使用してシステム初期設定テーブルの接尾部を指定する必要があります。代わりに、唯一の **PARM** パラメーターとして特別な SYSIN キーワードをコーディングし、SIT の接尾部およびその他のシステム初期設定パラメーターをその SYSIN データ・セットから渡すこともできます。

CICS は、PARM 文字列をスキャンして **SIT** パラメーター、特別な制御キーワード、またはシステム初期設定パラメーターを探し、次のように処理します。

- **SIT** パラメーターが検出されたが、SYSIN キーワードが検出されなかった場合、CICS は、**PARM** パラメーターのスキャンが終了したらすぐに SIT のロードを試行します。**PARM** パラメーターにも存在する CICS システム初期設定パラメーターの処理は、SIT がロードされた後に初めて実行されます。
- **SIT** パラメーターが検出され、SYSIN キーワードも検出された場合、CICS は、SYSIN データ・セットのスキャンも終了するまで、SIT のロードを試行しません。ここで、SIT のロードを延期する理由は、SYSIN データ・セットに、**PARM** パラメーターの指定をオーバーライドする他の **SIT** パラメーターがコーディングされている可能性があるからです。

**PARM** パラメーターにも存在するシステム初期設定パラメーターの処理は、SIT がロードされた後に初めて実行されます。

### PARM パラメーターのコーディング規則

**PARM** パラメーターのコーディングには、次のような規則が適用されます。

- コーディングできる最大文字数は 100 です。これに開始と終了の区切り文字は含まれません。区切り文字にはアポストロフィまたは括弧のいずれかを使用できます。
- すべての CICS システム初期設定パラメーターをコンマで区切る必要があります。区切り文字のコンマは 100 文字の制限に含まれます。この制限要素があるため、SYSIN 制御キーワードのみを指定するように **PARM** パラメーターの使用を制限することもできます。

EXEC ジョブ制御ステートメントの **PARM** パラメーターのコーディング規則について詳しくは、「[z/OS MVS JCL 解説書](#)」を参照してください。

### SYSIN データ・セットの処理

CICS は SYSIN データ・セットをスキャンして、SIT= パラメーターや特殊キーワード、さらにシステム初期設定パラメーターを探します。

CICS は、SYSIN に SIT= パラメーターを見つけると、PARM パラメーターで指定されているすべてをオーバーライドしてその SIT をロードしようとします。CICS が SYSIN に SIT= パラメーターを見つけないと、PARM パラメーターに指定されている任意の SIT をロードしようとします。

ただし、PARM パラメーターと SYSIN データ・セットをスキャンしても CICS が SIT= パラメーターを見つけられなかった場合、CICS は次のいずれかを行います。

1. PARM パラメーターまたは SYSIN データ・セットで CONSOLE を指定した場合、CICS は SIT 接尾部を最初のパラメーターとして入力するよう、次のメッセージをコンソールから出します。

```
II DFHPA1921 DBDCCICS PLEASE SPECIFY THE REQUIRED SIT SUFFIX, OR
                        SPECIFY 'NONE' (UNSUFFIXED).
```

2. CONSOLE を指定しないと、CICS は接尾部のない SIT モジュール (DFHSIT) を自動でロードしようとします。このロードが失敗すると、CICS はメッセージ DFHPA1106 を発行して、メッセージへの応答として SIT 接尾部を要求します。

注: CICS は、SIT がロードされるまで、PARM パラメーターおよび SYSIN データ・セットにコーディングされたもののシステム初期設定パラメーターも処理しません。

### **SYSIN データ・セットでの CICS システム初期設定パラメーターのコーディングに関するルール**

SYSIN データ・セットでの CICS システム初期設定パラメーターのコーディングに関するルールを以下にまとめます。

- 同じ行に複数のパラメーターを書くときは、コンマで区切る必要があります。
- SYSIN レコードの末尾にコンマを使用することもできます。
- 1 桁目にアスタリスクを入れて、コメントをコーディングしたり、CICS の特定の実行から初期設定パラメーターを一時的に削除したりできます。
- SYSIN 行のパラメーターの後にコメントを追加することもできますが、コメントの前に必ず 1 つ以上のブランク文字を入れてください。
- 1 桁目から 80 桁目までは、CICS によって入力データとして処理されます。
- オペランドが複数あるパラメーターの場合は、複数の SYSIN 行に分割してコーディングできます。ただしその場合は、コンマの直後で分割してください。CICS によって、ブランクが省略された形でオペランドが連結されます。
- 基本的に、1 つのオペランドを複数の行に分割することはできません。ただし、GMTEXT、USSCONFIG および USSCONFIG パラメーターの場合は、複数行にまたがる形で 1 つのオペランドを入力できます。これを行うには、最初の行で 80 桁目までコーディングした後に、次の行の 1 桁目から続行する必要があります。
- パラメーターのコーディングでアポストロフィや括弧やコンマを区切り文字として使用する場合は、注意が必要です。区切り文字を正しく組み込まないと、予期しない結果になりかねません。

### **コンソールの入力項目の処理**

一般的に CICS は、SIT をロードし、PARM パラメーターと SYSIN データ・セット内でコード化された初期設定パラメーターを処理するまで、コンソールからの読み取りを開始しません。CICS は、.END によって入力を終了するまで、コンソールからのシステム初期設定パラメーターを受け入れます。

SIT= パラメーターは、CICS が指定された SIT をロードしようとした時点で出されるメッセージ DFHPA1921 で要求されたときに、コンソールから最初のパラメーターとしてのみ指定できます。CICS が SIT をロードした後に SIT= パラメーターを指定しようとすると、エラーとして拒否されます。

### **コンソールでの CICS システム初期設定パラメーターのコーディング規則**

コンソールからパラメーターを読み取る準備ができると、CICS は DFHPA1104 メッセージを出します。

コンソールの単一行には初期設定パラメーターをいくつでも入力できますが、パラメーターを区切るのにコンマを使用する必要があります。.END 管理キーワードを入力してコンソールの入力を終了するまで、CICS はメッセージ DFHPA1105 を表示して、システム初期設定パラメーターを入力するよう引き続きプロンプトを出します。

### **コンソールからの初期設定パラメーターへの修正の入力**

PARMERR=INTERACT をコーディングしているときに、キーワードまたはキーワードに割り当てた値にパラメーター・エラーがあることを CICS が検出した場合、CICS はメッセージ DFHPA1912 または DFHPA1915 によって、エラーの修正を求めるプロンプトを出します。

CICS は、SIT のロード後、**PARM** パラメーターや SYSIN データ・セット内でエラーが検出されるたびに、各エラーに対する訂正を入力するようプロンプトを出します。つまり、**PARM** パラメーターや SYSIN データ・セット内でエラー状態のパラメーターの後に **APPLID** パラメーターがある場合、CICS がメッセージ DFHPA1912 および DFHPA1915 で表示するのは、SIT にコーディングされた APPLID であるということです。

## 始動と再始動の制御

CICS が実行する初期化のタイプは、START パラメーターだけによって決まるわけではありません。初期設定プロセスでは、ユーザーが指定したシステム初期設定パラメーターと共に、CICS のローカル・カタログおよびグローバル・カタログも重要な役割を果たします。

システム初期設定パラメーターは、SIT で指定することも、実行時に指定することも可能です。そのための 3 つの方法については、[126 ページの『システム初期設定の制御キーワードの使用』](#)を参照してください。

### CICS カタログの役割

CICS は、そのカタログを使用して、CICS のシャットダウンと次の再始動の間で情報を保管します。

#### グローバル・カタログ

CICS はグローバル・カタログを使用して、CICS がシャットダウンするとインストールされるすべてのリソース定義を保管します。

そのオプションは、以下のとおりです。

- BMS マップ・セット、および区分セット
- 接続およびセッション
- ファイル
- LIBRARY 連結
- Programs (プログラム)
- 端末および入力条件
- トランザクションおよびトランザクション・プロファイル
- 一時データ・キュー

CICS のシャットダウン時に CICS が保管するリソース定義は、コールド・スタート時にグループ・リスト・システム初期設定パラメーターで指定されたグループのリストからインストール済みであるか、あるいは CICS 領域の実行中に RDO を使用して動的にインストール済みである場合があります。

START=AUTO を指定して CICS を実行し、ウォーム・リスタートまたは緊急時再始動が行われた場合、CICS は、正常な CICS シャットダウン、またはシステム障害時と同じように、インストールされたすべてのリソース定義をリストアします。リストア中にエラーが続く場合、適切なメッセージが表示されます。一般のルールとして、START=COLD または START=INITIAL をコーディングしない限り、インストールされたリソース定義をリスタート時に変更することはできません。CICS 再始動タイプとグローバル・カタログの状態の可能な組み合わせの結果について詳しくは、[131 ページの『START システム初期設定パラメーターの指定』](#)を参照してください。

CICS ドメインもまた、グローバル・カタログを使用して、実行と実行の間のドメイン状態を保管します。この情報は、システム初期設定パラメーターを指定することで、再始動時にオーバーライドされるケースもあります。別のケースでは、カタログに保管されたドメイン情報は、再始動時に常に使用されます。あるいは、コールド・スタートを実行して、システム・デフォルトを強制することもできます。

**注：**なんらかの理由でグローバル・カタログを再初期化する必要がある場合、ローカル・カタログも再初期化する必要があります。

#### ローカル・カタログ

CICS ドメインは、ローカル・カタログを使用して、CICS の実行と実行の間で情報の一部を保管します。

ローカル・カタログを削除または再定義する場合、以下を行う必要があります。

1. ローカル・カタログを、ドメイン・レコードの初期設定で初期化する。
2. CICS 提供のユーティリティー・プログラム DFHSMUTL を使用して、レコードを再追加して、ストレージ・マネージャー・ドメイン・サブプール用に CICS 自己調整メカニズムを使用可能にする。この方法の詳細については、[ローカル・カタログ・ストレージ・プログラム \(DFHSMUTL\)](#)を参照してください。
3. グローバル・カタログを削除して再初期化する

ローカル・カタログの初期化について詳しくは、[72 ページの『ローカル・カタログの定義』](#)を参照してください。CICS トランザクション・ダンプ・データ・セット状況など、ローカル・カタログに保存されている一部の情報は、CICS システム初期設定時にシステム初期設定パラメーターでオーバーライドできます。



注：なんらかの理由でローカル・カタログを再初期化する必要がある場合、グローバル・カタログも再初期化を行う必要があります。

### START システム初期設定パラメーターの指定

以下のように **START** システム初期設定パラメーターを指定することで、CICS が実行する 始動のタイプを選択することができます。

#### START=AUTO

START オペランドとして AUTO をコーディングすると、CICS はグローバル・カタログ内に存在する可能性のある次の 2 つのレコードを確認することで、4 つのタイプの始動から実行するタイプを決定します。

- ・リカバリー・マネージャー制御レコード
- ・リカバリー・マネージャー自動始動オーバーライド・レコード

これらのレコードのいずれか、あるいは両方があるかどうかと、それらのコンテンツに基づいて、CICS は実行する始動のタイプを決定します。

#### 初期始動

CICS は、次のそれぞれのケースで初期始動を実行します。

- ・グローバル・カタログ内に AUTOINIT を指定するリカバリー・マネージャー自動始動オーバーライド・レコードがある。
- ・制御レコードが初期始動を指定している。(これは、前回の初期始動が失敗した場合に起こることがあります。)

注：特定のドメインはローカル・カタログに情報を格納するため、初期始動を実行するように CICS を設定する場合は CICS を 起動する前にローカル・カタログを再初期化する必要があります。ローカル・カタログへの書き込みを行うドメインについて詳しくは、[ローカル・カタログに書き込むドメイン](#)を参照してください。

#### コールド・スタート

CICS は、次のケースでコールド・スタートを実行します。

- ・リカバリー・マネージャー制御レコードがコールド・スタートを指定している。(これは、前回のコールド・スタートが完了しなかった場合に起こることがあります。)
- ・リカバリー・マネージャー制御レコード (初期始動以外を指定するもの) および AUTOCOLD を指定する自動始動オーバーライド・レコードの両方がある。

ローカル・リソースのログ・レコードは消去され、CSD または CICS 管理テーブルからリソース定義が再ビルドされます。START=COLD で説明されているように、他のシステム上の作業単位は、このシステムと再同期されます。

#### ウォーム・スタート

リカバリー・マネージャー制御レコードに、前回の CICS の実行が正常に終了したことが、成功のウォーム・キープポイントと共に示されている場合、CICS はウォーム・リスタートを実行します。ただし 自動始動オーバーライド・レコードが AUTOINIT または AUTOCOLD を指定している場合を除きます。その場合は、初期始動またはコールド・スタートが実行されます。

ウォーム・リスタートを成功させるには、前回の実行時に CICS ドメインによって保存された情報がローカル・カタログに含まれている必要があります。

ウォーム・スタートは、CICS を前回のシャットダウンの時点の状態にリストアします。

**NEWSIT** システム初期設定パラメーターをコーディングして、ウォーム・リスタートを変更できます。これは、SIT でコーディングされているシステム初期設定パラメーターの適用を強制する効果があり、前回の CICS シャットダウン時にカタログされた状況がすべてオーバーライドされます。

これに対する例外は、システム初期設定パラメーター FCT、CSDxxxxx グループ (例えば CSDACC)、および GRPLIST です。これらは NEWSIT=YES を指定してもウォーム・リスタートでは常に無視されます。NEWSIT=YES を指定すると、事実上、部分的なコールド・スタートが実行されます。



## 緊急始動

グローバル・カタログの制御レコードに、前回の CICS の実行時に即時シャットダウンまたは非制御のシャットダウンで終了したことが示されている場合、CICS は緊急時再始動を実行します。

通常モードの操作には、START=AUTO を使用してください。これにより、CICS によって自動的に始動のタイプが選択されます。です。オーバーライドを設定するには、リカバリー・マネージャー・ユーティリティー・プログラム DFHRMUTL を使用します。

## START=INITIAL

CICS は、システム初期設定パラメーターによって指定されたリソース定義を使用して、初期設定を行います。その際、グローバル・カタログ内のウォーム・キーポイントに保存されている、以前にインストールされたリソース定義は無視します。これには、**GRPLIST** システム初期設定パラメーターによって指定されたすべてのリソースのグループと、CICS 管理テーブルで指定されたリソースが含まれます。

**注:** グローバル・カタログおよびシステム・ログが初期設定されると、その中の情報はすべて失われます。リモート・システムのリカバリー情報は保存されないため、分散作業単位に破損が生じる可能性があります。

START=INITIAL の指定が必要になるケースはほとんどありません。CSD からローカル・リソースの定義を再インストールする場合は、代わりに START=COLD を使用します。

初期始動が必要な例として、次のような場合があります。

- 新しい CICS システムを初めて立ち上げる場合。
- 重大なソフトウェア障害が発生し、システム・ログが破損していた場合。
- グローバル・カタログがクリアされた、または再初期化された場合。
- ダミーのシステム・ログを使用して CICS を実行する場合。(システム・ログがダミーとして定義されている場合は無視されます。)

CICS の初期始動を実行する必要がある場合、次の 2 とおりの方法があります。

- START=INITIAL を指定する。
- リカバリー・マネージャー・ユーティリティー・プログラム DFHRMUTL を使用して、自動始動オーバーライド・レコードを AUTOINIT に設定し、START=AUTO を指定する。

**注:** 初期始動は、ローカル・カタログを自動的に再初期化しません。

## START=COLD

CICS は、システム初期設定パラメーターによって指定されたリソース定義を使用して、初期設定を行います。その際、グローバル・カタログ内のウォーム・キーポイントに保存されている、以前にインストールされたリソース定義は無視します。これには、**GRPLIST** システム初期設定パラメーターによって指定されたすべてのリソースのグループと、CICS 管理テーブルで指定されたリソースが含まれます。

リモート・システムや、RMI 接続のリソース・マネージャーに関連するリカバリー情報は保存されません。始動時に CICS ログがスキャンされ、リモート・システムや、RMI によって接続される非 CICS リソース・マネージャー (Db2 など) に対して作業単位が行う必要のある事柄に関する情報が保存されます。(つまり、ローカル UOW の結果に関する決定が保存されます。これによって、リモート・システムや RMI リソース・マネージャーはそれぞれのリソースを再同期することができます。)

コールド・スタートでは、次のものは保持されないことに注意してください。

- CICS の前回の実行では完全にはコミットまたはバックアウトされなかったローカル・リソースへの更新。具体的には、リモート・システムの作業単位を正常に再同期しても、それらの分散作業単位で更新されるローカル・リソースはロックされておらず、コミットまたはバックアウトのいずれの状態にもなる必要がありません。
- LU6.1 リンクによって接続されたリモート・システム、または MRO によって接続された以前のリリースの CICS システムのリカバリー情報。
- 動的に定義されたプログラムの LIBRARY 定義。静的な DFHRPL 連結のみが、始動時に指定されるグループ・リスト内の LIBRARY 定義、または始動時に BAS を介してインストールされる LIBRARY 定義とともに残されます。

START=COLD によって開始される始動は、同じグローバル・カタログを使用する前回の CICS システムの実行をまったく参照しないというわけではありません。前回の実行への参照を行わずに CICS の完全なコールド・スタートを実行したい場合、START=INITIAL をコーディングします。CSD からローカル・リソースの定義を再インストールするには、START=COLD を使用します。

CICS がグローバル・カタログに記録したシステム終了のタイプとは関係なく、CICS のコールド・スタートを実行する必要がある場合があります。これを行うには次の 2 とおりの方法があります。

- START=COLD を指定する。
- DFHRMUTL を使用して自動始動オーバーライド・レコードを AUTOCOLD に設定し、START=AUTO を指定する。

注：コールド・スタートは、ローカル・カタログを自動的に再初期化しません。

133 ページの表 20 では、START パラメーターの効果が、CICS グローバル・カタログおよびシステム・ログの状態によって異なることを示しています。

注：システム・ログがダミーとして定義されている場合は無視されます。

START パラメーター	グローバル・カタログの状態	システム・ログの状態	再始動時の結果
任意。	VSAM に対して定義されていません。	任意。	JCL エラー。
INITIAL	定義済み。	任意。	CICS は初期始動を実行します。グローバル・カタログおよびシステム・ログは初期設定されます。
COLD	定義済み。ただし、リカバリー・マネージャー制御レコードは含まれていません。	任意。	CICS は確認のプロンプトが出た後に初期始動を実行します。グローバル・カタログおよびシステム・ログは初期設定されます。
COLD	リカバリー・マネージャー・レコードを含みます。	未定義、ダミー、または空。	メッセージ DFHRM0401 が発行されます。始動は失敗します。
COLD	リカバリー・マネージャー・レコードを含みます。	前回の実行のレコードを含みます。	CICS はコールド・スタートを実行します。ローカル・リソースに対する変更に関連するシステム・ログのリカバリー・レコードは削除されます。
AUTO	定義済み。ただし、リカバリー・マネージャー制御レコードは含まれず、AUTOINIT 自動始動オーバーライドは行われません。	任意。	メッセージ DFHRM0137 が発行されます。始動は失敗します。
AUTO	リカバリー・マネージャー AUTOINIT 自動始動オーバーライドを含みます。	任意。	CICS はプロンプトなしで初期始動を実行します。グローバル・カタログおよびシステム・ログは初期設定されます。
AUTO	初期始動、および AUTOCOLD 自動始動オーバーライドを指示しないリカバリー・マネージャー制御レコードを含みます。	前回の実行のレコードを含みます。	CICS はコールド・スタートを実行します。ローカル・リソースに対する変更に関連するシステム・ログのリカバリー・レコードは削除されます。
AUTO	リカバリー・マネージャー・レコードを含むが、AUTOINIT オーバーライドは行われません。	未定義、ダミー、または空。	メッセージ DFHRM0401 が発行されます。始動は失敗します。

表 20. グローバル・カタログおよびシステム・ログと START= パラメーターの効果の関連 (続き)

START パラメーター	グローバル・カタログの状態	システム・ログの状態	再始動時の結果
AUTO	初期始動を指示するリカバリー・マネージャー制御レコードを含みます。	任意。	CICS は初期始動を実行します。グローバル・カタログおよびシステム・ログは初期設定されます。
AUTO	コールド・スタートを指示し、自動始動オーバーライドを行わないことを指示するリカバリー・マネージャー制御レコードを含みます。	前回の実行のレコードを含みます。	CICS はコールド・スタートを実行します。ローカル・リソースに対する変更に関連するシステム・ログのリカバリー・レコードは削除されます。
AUTO	緊急始動を指示し、自動始動オーバーライドを行わないことを指示するリカバリー・マネージャー制御レコードを含みます。	前回の実行のレコードを含みます。	CICS は緊急始動を実行します。
AUTO	ウォーム・スタートを指示し、自動始動オーバーライドを行わないことを指示するリカバリー・マネージャー制御レコードを含みます。	前回の実行のレコードを含みます。	CICS はウォーム・スタートを実行します。

注:

1. CICS グローバル・カタログとローカル・カタログの状態が揃っていることは重要です。CICS がウォーム・スタートまたは緊急始動を実行しようとしたが、ローカル・カタログが初期設定されていた場合、始動は失敗します。そのため、ローカル・カタログの初期設定を行うのは、グローバル・カタログの初期設定を行うときのみにする必要があります。
2. DFHRMUTL ユーティリティおよび DFHCCUTL ユーティリティを常に同じジョブで実行することを推奨します。まず DFHRMUTL を実行し、その戻りコードを確認してから DFHCCUTL を実行します。こうすることで、グローバル・カタログとローカル・カタログの状態が異なることはありません。

134 ページの表 21 では、CICS 始動の各タイプが CICS トレース、モニタリング、統計、およびダンプの各ドメインに与える影響を示しています。

表 21. 始動のタイプの CICS ドメインへの影響

ドメイン	CICS カタログの状態	ウォーム・スタートまたは緊急始動	初期始動またはコールド・スタート
トレース	関連なし。	ドメインはシステム 初期設定パラメーターに応じて初期設定します。	ドメインはシステム 初期設定パラメーターに応じて初期設定します。
モニター	グローバル・カタログは新たに初期設定されます。	ドメインはシステム 初期設定パラメーターに応じて初期設定します。	ドメインはシステム 初期設定パラメーターに応じて初期設定します。
モニター	グローバル・カタログに、前回の CICS シャットダウンのモニタリングの状況が含まれます。	ドメインはカタログのモニタリング状況を使用しますが、これはシステム 初期設定のオーバーライド・パラメーターがある場合は変更されます。	ドメインはシステム 初期設定パラメーターに応じて初期設定します。
統計	グローバル・カタログは新たに初期設定されます。	ドメインは CICS 定義のシステムのデフォルト値に応じて初期設定します。	ドメインは CICS 定義のシステムのデフォルト値に応じて初期設定します。

表 21. 始動のタイプの CICS ドメインへの影響 (続き)

ドメイン	CICS カタログの状態	ウォーム・スタートまたは緊急始動	初期始動またはコールド・スタート
統計	グローバル・カタログには、CICS シャットダウンの統計の状況が含まれます。	ドメインは、このカタログの統計状況を使用します。	ドメインは CICS 定義のシステムのデフォルト値に応じて初期設定します。
ダンプ	グローバル・カタログは新たに初期設定されます。	ドメインは CICS 定義のシステムのデフォルト値に応じてダンプ・テーブルを初期設定します。その他のダンプ属性は、システム初期設定パラメーターで設定されます。	ドメインは空のダンプ・テーブルを初期設定し、すべてのダンプ要求に対して CICS で定義されているデフォルトのアクションを実行します。その他のダンプ属性は、システム初期設定パラメーターで設定されます。
ダンプ	グローバル・カタログには、CICS シャットダウンのダンプ状況が含まれます。	ドメインは、カタログからダンプ・テーブルとダンプ状況を読み取ります。その他のダンプ属性は、任意のシステム初期設定パラメーターで変更されます。	ドメインは空のダンプ・テーブルを初期設定し、すべてのダンプ要求に対して CICS で定義されているデフォルトのアクションを実行します。その他のダンプ属性は、システム初期設定パラメーターで設定されます。

### CICS の始動と z/OS Communications Server セッション

SNA ネットワークでは、Communications Server が CICS より前に始動すると、CICS と z/OS Communications Server との間のセッションが自動的に開始されます。

CICS を始動したときに z/OS Communications Server がアクティブでない場合は、以下のメッセージが受信されます。

```
F vtamname,USERVAR,ID=generic-applid,VALUE=specific-applid
+DFHSI1589D 'applid' VTAM is not currently active.
+DFHSI1572 'applid' Unable to OPEN VTAM ACB - RC=xxxxxxx, ACB CODE=yy.
```

DFHSI1589D と DFHSI1572 のメッセージが生成された場合は、Communications Server の始動後に、サポート対象の MVS コンソールまたは非 Communications Server 端末から **CEMT SET VTAM OPEN** コマンドを使用して、CICS-z/OS Communications Server セッションを手動で開始できます。

z/OS Communications Server がアクティブになっていても、Communications Server が CICS APPLID を認識できないために、CICS が SNA ACB を開けない場合は、以下のメッセージが生成されます。

```
F vtamname,USERVAR,ID=generic-applid,VALUE=specific-applid
+DFHSI1592I 'applid' CICS applid not (yet) active to VTAM.
+DFHSI1572 'applid' Unable to OPEN VTAM ACB - RC=00000008, ACB CODE=5A.
```

原因としては、APPLID オペランドの値のエラーが考えられます。その場合は、エラーを修正して、CICS を再始動する必要があります。その他の原因と処置については、[CICS メッセージ](#)を参照してください。

### CICS パラメーター・マネージャー・ドメイン

パラメーター・マネージャー・ドメインは、初期化の開始時にシステム初期設定テーブルをロードしたり、PARM や SYSIN やシステム・コンソールから他のパラメーターを読み取ったりするほかに、SIT の管理も担当します。

パラメーター・マネージャー・ドメインは、要求に応じてシステム初期設定パラメーターを他の CICS ドメインに渡します。ただし、SIT を直接使用するアプリケーション・ドメイン (AP) は例外です。

ドメインの初期化プロセスは、以下のようになります。

## 始動タイプの照会

各ドメインは、始動のタイプ (初期、コールド、ウォーム) をパラメーター・マネージャーに尋ねます。ただし、トレース・ドメインは例外です。(この目的に関する限り、パラメーター・マネージャー・ドメインは、緊急リスタートをウォーム・スタートとして処理します。)

## 初期始動またはコールド始動の場合

初期始動またはコールド始動の場合、各ドメインは、カタログからドメイン・レコードを読み取りません。その代わりに、パラメーター・マネージャー・ドメインからシステム初期設定パラメーターを要求します。コールド・スタートの場合、パラメーター・マネージャー・ドメインは、SIT からすべてのシステム初期設定パラメーターを返します (オーバーライドがあれば、オーバーライドによって変更された値を返します)。

## ウォーム始動の場合

ウォーム始動の場合、各ドメインは、カタログからドメイン・レコードを読み取ることによってウォーム・スタートを実行しようとします。

- 各ドメインは、状況レコードの読み取りに成功すると、ウォーム・スタートを実行します。必要な場合には、パラメーター・マネージャー・ドメインからシステム初期設定パラメーターも要求します。ウォーム・スタートの場合、パラメーター・マネージャー・ドメインは、PARM や SYSIN やシステム・コンソールによってオーバーライドされているシステム初期設定パラメーターだけを返します。
- 各ドメインは、何かの理由で状況レコードの読み取りに失敗すると、コールド・スタートを実行します。そのために、パラメーター・マネージャー・ドメインからすべてのシステム初期設定パラメーターを要求するか、システム初期設定パラメーターがない場合はシステム・デフォルト値を使用します。

## NEWSIT または新しい接尾部

START=AUTO がウォーム・スタートに解決される状況でも、以下の場合は、パラメーター・マネージャーがほとんどのシステム初期設定パラメーターを適用します。

- PARM や SYSIN やコンソールでシステム初期設定パラメーターとして NEWSIT=YES を指定した場合。
- CICS の前回の実行時とは異なる SIT 接尾部を指定した場合。パラメーター・マネージャーは、実行ごとにその接尾部をグローバル・カタログに保管しますが、新しい接尾部が検出されると、NEWSIT=YES オプションを強制的に適用します。

NEWSIT=YES を指定した場合に無視されるパラメーターの詳細については、[NEWSIT システム初期設定パラメーター](#)にある NEWSIT パラメーターの説明を参照してください。

注：トレース・ドメインは上記のルールの例外で、常にコールド・スタートを実行します。トレース・ドメインは、他のドメインとは異なり、CICS のシャットダウン時に状況情報を保管しません。始動のタイプにかかわらず、パラメーター・マネージャー・ドメインからすべてのシステム初期設定パラメーターを要求します。

## CICS 始動の終了

実行された始動タイプにかかわらず、DFHSI1517 メッセージがオペレーティング・システム・コンソールに表示されたら、端末要求を処理するための CICS の準備は完了です。

始動プロセスが完了すると、ユーザーは CICS に接続されているすべての端末からトランザクションを入力することができます。CICS 提供のトランザクションについては、[CICS supplied transactions descriptions](#)を参照してください。

DFHSI1517 によってシステム初期設定の終了が通知されても、領域が作業を受け取る準備が完全にはできていない場合があります。その理由は、このメッセージが表示された後も初期設定が続く場合があるためです。例えば、バンドル定義リソースの中には、非同期的にインストールされるものがあります。このため、TCP/IP リスナーが開いていて作業を受け入れるようになっていても、一部のアプリケーションに必要な JVMSERVER などのリソースの初期設定がまだ完了していないことがあります。これが原因でトランザクションが異常終了する場合があります。Web サービスの場合、パイプライン・スキャンが完了していない可能性があるため、スキャンの完了前に Web サービスを使用すると失敗します。

**WLMHEALTH** システム初期設定パラメーターを設定すると、CICS でウォームアップ・プロセスを使用することができます。z/OS ワークロード・マネージャーのヘルス・サービスが領域でアクティブな場合は、メッセージ DFHSI1517 が返されたときに、ウォームアップ・プロセスが開始されます。次に、CICS は領域



の z/OS WLM ヘルス値を調整して、領域が作業を完全に処理できるようになるまで領域内への作業の流れを制御します。CICS ウォームアップ・プロセスについて詳しくは、[z/OS ワークロード・マネージャー 正常性サービスの使用による CICS のウォームアップおよび冷却](#)を参照してください。

## システム始動時に使用されたシステム初期設定パラメーターに関する情報の取り出し

CICSplex SM API を使用して、CICS システム 初期設定パラメーター、およびシステム 始動時に使用されたシステム 初期設定パラメーター・オーバーライドに関する情報を検出できます。システム 初期設定パラメーターの取り出しは、CICSplex SM コマンド・レベル・インターフェース、CICS 管理クライアント・インターフェース (CMCI)、および Web ユーザー・インターフェース (WUI) によりサポートされています。

注：CICSplex SM API は、システム 始動時に指定されていたシステム 初期設定パラメーターのみを取り出し、現在のアクティブ・システム値は表示しません。

パラメーターを取り出す際に、以下のオプションがあります。

- 指定されたオーバーライド値と組み合わせてシステム 初期設定パラメーターを取り出せます。
- システムの始動時に指定されていた元のシステム 初期設定パラメーター値を取り出せます。
- 単一のオーバーライド・ソースからの値を取り出せます。

多くの他の CICSplex SM 操作と同じように、コンテキストと有効範囲を指定してどの CICS 領域で取り出し操作を行うか制御できます。

システム 初期設定パラメーターの取り出しは、CICSplex SM のリソース SYSPARM を使用して実装されています。SYSPARM リソースには、GET 操作と関連して、2 つの必須パラメーター **PARMSRCE** および **PARMTYPE** があります。これらのパラメーターを使用して、どのパラメーターにそのソースに応じて取り出しを行わせるか指定します。

システム 初期設定パラメーターの検出は、以下の方法で実装できます。

- CICS Explorer **SM 操作ビュー** > 「領域ビュー (Regions view)」で、領域を右クリックし、「**SIT パラメーターの表示**」を選択します。
- SYSPARM オブジェクトで作動する **EXEC CPSM GET** コマンドを使用した API プログラム中。
- CICSSystemParameter 外部リソースで作動する **CMCI GET** メソッドの使用。
- CICS 領域ビュー・セットからリンクしている SYSPARM リソース・テーブルに基づいた WUI 操作ビューの使用。

**EXEC CPSM GET** コマンドまたは CMCI を使用する場合は、**PARMSRCE** および **PARMTYPE** を使用してパラメーター式を定義して、取り出すパラメーターを指定できます。WUI では、**PARMSRCE** および **PARMTYPE** をフィルターとして使用して、表示されるレコードを制御できます。これらのパラメーターは両方とも必須です。**PARMTYPE** には、システム 初期設定テーブルの値を指定する必要があります。**PARMSRCE** には、以下のオプションの 1 つを指定します。

### COMBINED

元のシステム 初期設定パラメーター定義と適用されたパラメーター・オーバーライドの組み合わせ

### CONSOLE

システム・コンソールで始動に指定されたとおりの指定変更パラメーター

### JCL

JCL EXEC PGM ステートメントにより提供される指定変更パラメーター

### SYSIN

SYSIN データ・セットに定義されている始動ジョブ・ストリームからの指定変更パラメーター

### TABLE

DFHSITxx ロード・モジュールから抽出される元のシステム 初期設定テーブル値

パラメーターおよび指定変更値は、ソース、例えば、SYSIN データ・セットにおける順序に関係なく、SYSPARM リソース・テーブルの KEYWORD 属性の順序で返されます。



システム・パラメーター値は 255 バイトを超えていることがあります。その場合、パラメーター値は、それぞれに最大 255 バイトのパラメーター値が格納された複数のレコードに分けられます。各レコードには、固有のセグメント番号 (レコードを正しい順序に並べるために使用)、合計セグメント数 (このパラメーター値のセグメント数)、値の全長 (すべてのセグメント長を合計した長さ) のフィールドが含まれています。

取り出すシステム 初期設定パラメーターが CICS 始動時に CICS 領域に対して有効であったことを確認する必要があります。取り出す操作は、無効なパラメーター値が始動時にコンソールから修正されていれば、矛盾することなく実行可能です。一部のパラメーターでは修正された値が表示されますが、元の値が表示されるものもあります。また、修正された値の場合は、システム・パラメーターのソース・ロケーションが、元のソース・ロケーションではなく、コンソールとして返されることがあります。

以下の CMCI 要求は、CICSplex MYPLEX における領域 REGION に対して設定されている制御テーブル接尾部の値を取り出します。

```
/CICSSystemManagement/CICSSystemParameter/<MYPLEX>/<REGION>?PARAMETER=PARMSRCE (COMBINED)
%20PARMTYPE (SIT)&CRITERIA=KEYWORD%3D++T%20AND%20NOT%20KEYWORD%3DMXT
```

## 機能切り替えの指定

CICS TS 継続的デリバリーでは、より短いリズムで新しい機能拡張が提供されるので、これらの新機能により迅速にアクセスできます。通常、これらの機能は CICS の基本機能ではなく、デフォルトで無効になっています。CICS 領域でそうした機能を有効にして使用するには、機能切り替え構成ファイルを 1 つ以上セットアップする必要があります。機能切り替え構成ファイルで機能切り替えを定義すれば、機能のオン/オフを切り替えられるようになります。一部の機能については、構成オプションを指定するための機能切り替えを定義することもできます。

**注:** 機能切り替えは軽量構成オプションです。そのため、検証は行われません。無効な機能切り替えが指定された場合は、それがシステム・ログのメッセージ DFHPA1956I の下に表示されますが、その値は無視され、代わりにデフォルト値が使用されます。

### 始める前に

CICS 領域にインストールして構成できる機能のリストについては、[トグル対応機能、リリース別のサポート](#)を参照してください。その他の機能が APAR に用意され、その説明が記されることもあります。どんな新機能を有効にできるかは、CICS システムに適用した APAR によって異なります。

個々の切り替え対応機能に関するトピックを見つけるには、[トグル対応機能、リリース別のサポート](#)にあるリンクを参照してください。各トピックでは、機能の概要を示しているほか、機能の構成オプションを有効にして設定するための機能切り替えと、CICS 始動時に機能切り替え構成ファイルが見つからなかった場合に使用されるデフォルト値について説明しています。

### このタスクについて

機能切り替え構成ファイルには、共通ファイルと領域レベル・ファイルの 2 つのタイプがあります。

共通ファイルを使用して、同じ USSCONFIG ディレクトリーを共用する CICS 領域のグループで有効にする機能を制御できます。共通ファイルを使用することで、1 つ以上の機能切り替え設定または構成オプションが領域レベル・ファイルによって設定されるのを防止できます。

領域レベル・ファイルを使用して単一領域で機能を試してから、実稼働環境でその機能を有効にすることができます。

どちらのファイルもオプションです。どちらも存在しなければ、デフォルト値が使用されます。

機能構成ファイルでは、機能を有効にしたり構成オプションを設定したりするための機能切り替えを指定します。

機能を有効にするために使用される機能切り替えは、次の形式になります。

```
[finalize] com.ibm.cics.component.feature = {true|false}
```

どの機能にも 1 つ以上の構成オプションを設定できます。構成オプションを設定するために使用される機能切り替えは、次の形式になります。

```
[finalize] com.ibm.cics.component.feature.config_property = value
```

値は、数値、ストリング、ブール値のいずれかです。

ヒント：

- 改行の位置で行継続文字 `¥` を使用できます。名前や値で `¥` 文字が必要な場合は、行継続をエスケープするために `¥` 文字を 2 つ続けて使用してください。
- コメント行を指定するために、行の先頭でハッシュ文字 `#` を使用できます。

グループ・レベル・ファイルでオプション・パラメーター **finalize** を使用すると、領域レベル・ファイルによって機能切り替えが変更されるのを防止できます。例えば、開発者に領域レベル・ファイルへのアクセスを認める場合に、特定の機能の使用を禁止できます。

## 手順

- 共通構成ファイルをセットアップするには、[139 ページの『共通構成ファイルのセットアップ』](#)の手順を実行します。
- 領域レベル構成ファイルをセットアップする場合は、[140 ページの『領域レベル構成ファイルのセットアップ』](#)の手順を実行します。
- CICS 領域を再始動します。

## タスクの結果

機能切り替え構成ファイルは、CICS の始動時にシステム 初期設定で SIT パラメーターが読み取られた直後に読み取られます。共通ファイルが先に読み取られ、その後に領域レベル・ファイルが読み取られます。パラメーター **finalize** が指定された共通設定を除き、領域レベル・ファイルでの設定は、共通ファイルでの設定をオーバーライドします。

CICS 領域の始動後に、CICS ログの DFHPA1956I メッセージでそれぞれの機能切り替えの値を確認できます。そのメッセージでは、機能切り替え構成ファイルから読み取られて格納された設定を確認できますが、機能切り替えが使用されるかどうかや使用方法は確認できません。

**制約事項：**CICSplex SM は機能切り替えを表示しません。

## 例

機能切り替え構成ファイルの例を以下に示します。

```
# Feature Toggle File
com.ibm.cics.bms.ids=true 1
com.ibm.cics.bms.ids.vtamignore=false 2
com.ibm.cics.bms.ids.action=ABEND 3
```

**1:** BMS 3270 侵入検出サービスを切り替えます。

**2** と **3:** 機能の構成オプションです。

## 次のタスク

SPI コマンド **INQUIRE FEATUREKEY** またはパラメーター・ドメイン関数 DFHPAIQX

INQUIRE\_FEATUREKEY を使用して、CICS 領域内の切り替え対応機能の使用可能性および構成設定を照会できます。詳しくは、[INQUIRE FEATUREKEY](#) および [パラメーター・ドメイン XPI 関数](#)を参照してください。

## 共通構成ファイルのセットアップ

共通ファイルを使用して、同じ USSCONFIG ディレクトリーを共用する領域のグループのために、機能切り替えをセットアップできます。

## このタスクについて

機能切り替え共通構成ファイルは、機能を有効にする CICS 領域の USSCONFIG ディレクトリーに配置する必要があります。

共通構成ファイルを領域のグループに適用する場合、それらの領域は同じ USSCONFIG ディレクトリーを共有する必要があります。

## 手順

1. USSCONFIG ディレクトリーで、`featuretoggle.properties` という名前のファイルを作成します。

デフォルトのパスは `/var/cicsts/dfhconfig` です。

**注:** CICS のインストール・プロセスで、`/pathprefix/usr/lpp/cicsts/ussdira/dfhconfig` に空の `featuretoggle.properties` ファイルが作成されます。そのファイルを必要な USSCONFIG ディレクトリーにコピーすることもできます。

**ヒント:** CICS 領域で使用しているパスが不明な場合は、システム・メッセージ・ログの DFHPA1951I メッセージでパスを確認できます。

2. そのファイルで、使用する機能ごとに、機能を有効にするための機能切り替えを指定し、必要に応じて、構成オプションを設定するための機能切り替えを指定します (138 ページの『機能切り替えの指定』を参照)。このファイルは、UNIX System Services 上に EBCDIC ファイル・エンコード方式で保管する必要があります。

**注:** 共通ファイルに指定された特定の機能切り替えが領域レベル・ファイルで設定されることを防止するために、次の例のように、切り替え設定で **finalize** パラメーターを指定できます。

```
finalize com.ibm.cics.bms.ids=true
finalize com.ibm.cics.bms.ids.vtamignore=false
```

## タスクの結果

これで、機能切り替え共通構成ファイル `featuretoggle.properties` が USSCONFIG ディレクトリーに作成されました。

## 領域レベル構成ファイルのセットアップ

領域レベル・ファイルを使用して、1つの領域の機能切り替えをセットアップできます。領域レベル・ファイルは、その領域の USSCONFIG ディレクトリーのサブディレクトリーにセットアップする必要があります。

## 手順

1. CICS® 領域の USSCONFIG ディレクトリー内に、1つのディレクトリーを作成します。このディレクトリーの名前は、領域のアプリケーション ID と同じにする必要があります、大文字でなければなりません。
2. このディレクトリーで、`featuretoggle.properties` という名前のファイルを作成します。
3. そのファイルで、使用する機能ごとに、機能を有効にするための機能切り替えを指定し、必要に応じて、構成オプションを設定するための機能切り替えを指定します (138 ページの『機能切り替えの指定』を参照)。このファイルは、UNIX System Services 上に EBCDIC ファイル・エンコード方式で保管する必要があります。

## タスクの結果

CICS 領域の機能切り替え構成ファイル `featuretoggle.properties` を作成できました。

## CICS の始動

システム環境に応じて、**START** コマンドを使ってプロシージャから CICS ジョブを開始することも、内部読み取りプログラムから CICS 始動ジョブ・ストリームを実行依頼することもできます。

### このタスクについて

内部読み取りプログラムから実行できるバッチ・ジョブの例については、[143 ページの『サンプル CICS 始動ジョブ』](#)を参照してください。CICS を開始タスクとして始動する時に使用できるカタログ式プロシージャの例については、[152 ページの『サンプル CICS 始動プロシージャ』](#)を参照してください。

始動ジョブを実行すると、**CICS システム初期化**というプロセスが開始されます。トランザクションを実行するのは、そのプロセスの完了後でなければなりません。CICS の初期化が完了すると、システム・コンソールに以下のメッセージが表示されます。

```
DFHSI1517 - applid: Control is being given to CICS.
```

CICS の初期化には多くのアクティビティーがかかわっています。その一部を以下にまとめます。

- CICS を実行するのに必要なストレージを CICS アドレス・スペースの専用領域から取得します (16MB 境界の上と下の両方)。
- システム初期設定パラメーターの指定値に基づいて、実行用の CICS システム・パラメーターをセットアップします。
- **START=** システム初期設定パラメーターで指定されている始動オプションに基づいて、CICS ドメインをロードして初期化します。
- CICS の中核と必要な CICS モジュールをロードします。
- CICS リソース定義をインストールするために、以下のようになります。
  - **GRPLIST=** システム初期設定パラメーターで指定されているリソースのグループを CSD からロードします。
  - システム初期設定パラメーターで指定されている管理テーブルをロードします。
- 初期化のために必要なデータ・セットを開きます。CICS 領域の前の実行が正常にシャットダウンしなかった場合は、バックアウトのために必要なデータ・セットも開きます。CICS のシャットダウン前にバックアウトが失敗していた場合は、正常なシャットダウンの後でも、バックアウトのために必要なデータ・セットを開きます。失敗したバックアウトが緊急リスタート時に再試行されるからです。
- 端末管理テーブル (TCT) で必要な BSAM 順次デバイスを開きます。

CICS リカバリー・オプションを指定して CICS を運用している場合は、バックアウト手順を使用して、リカバリー可能リソースを論理的な整合状態に復元できます。例えば、**START=**AUTO を指定して CICS を始動した時に、前回のシャットダウンが即時シャットダウンまたは非制御シャットダウンだったことを CICS が検出すると、バックアウトが発生する可能性があります。SDTRAN の場合は、即時シャットダウンの時でも、バックアウトの対象になる進行中の作業単位がいつでも残るとは限りません。進行中の UOW がない場合でも、バックアウト再試行の対象になるバックアウト失敗の UOW が存在する可能性もわずかながらあります。

バックアウト、リカバリー、再始動の背景情報については、[リカバリー処理のためのトラブルシューティング](#)を参照してください。

初期化の最終段階で、プログラム・リスト・テーブル (PLT) に指定した一連のプログラムを実行できます。SIT の PLTPI パラメーターで、PLT の絶対パス名または接尾部を指定してください。初期化時の PLT プログラムは、CICS キーにある CICS タスクの制御下で実行されます。ストレージも CICS キー・ストレージ内にあり、他のトランザクションによる上書きから保護されます。

PLTPI リソース・マネージャーは、システム初期設定テーブル内の指定値に基づいて以下の順序で開始されます。

1. CPSMCONN=CMAS
2. CPSMCONN=LMAS
3. CPSMCONN=WUI



4. DBCTLCON=YES

5. DB2CON=YES

6. MQCON=YES

この順序は、上記のシステム初期設定パラメーターだけに当てはまるものであり、PLTPI プログラム全般に当てはまるわけではありません。

PLT プログラムの実行に使用するユーザー ID について詳しくは、[PLT programs](#) を参照してください。

PLT プログラムの作成に関するプログラミング情報については、[Writing initialization and shutdown programs](#) を参照してください。

CICS と Db2 を一緒に実行している場合は、PLT 始動時に使用する Db2 サブシステム ID を **INITPARM** システム初期設定パラメーターで以下のように指定できます。

```
INITPARM=(DFHD2INI='yyyy')
```

yyyy は、4 文字の Db2 サブシステム ID です。特殊文字に関する MVS JCL ルールに準拠した値を指定する必要があります。**INITPARM** システム初期設定パラメーターで Db2 サブシステムのデータ共有グループの ID を使用することはできません。1 つの Db2 サブシステムの ID を指定する必要があります。**INITPARM** システム初期設定パラメーターを使用して Db2 サブシステムを指定する場合には、インストール済みの DB2CONN 定義の DB2GROUPID と DB2ID を両方ともブランクのままにしてください。DB2CONN 定義のこれらの属性のいずれかで指定された ID は、**INITPARM** システム初期設定パラメーターで指定した ID を指定変更します。

## CICS 領域のアドレス・スペース・ストレージ制限の設定

CICS ジョブを実行依頼する際、CICS が稼働するアドレス・スペースの仮想記憶の量を指定する z/OS パラメーターの **REGION** および **MEMLIMIT** を設定します。

### このタスクについて

CICS の実行中に、CICS 領域の **REGION** および **MEMLIMIT** の値を変更することはできません。新しい値は、CICS 領域の開始時にのみ指定できます。

**REGION** パラメーターは、CICS アドレス・スペースの 24 ビットと 31 ビット・ストレージ (2 GB 境界より下) の量の要求を指定します。最大で 2047 MB のストレージを要求できますが、24 ビット・ストレージのシステム領域や、ローカル・システム・キュー域 (LSQA) などの高専用域内の項目用には、境界の下に十分なストレージを残す必要があります。さまざまな方法で **REGION** パラメーターを指定して、特定の量のストレージを要求したり、使用可能なすべての 24 ビットか 31 ビットの専用ストレージを要求したりすることができます。2 GB 境界より下に得られる領域サイズは、予測不能なことがあります。

**MEMLIMIT** パラメーターでは、CICS 領域の 64 ビット (2 GB 境界より上) ストレージの制限を指定します。CICS 領域には、10 GB 以上の **MEMLIMIT** 値が必要です。z/OS の **MEMLIMIT** のデフォルト値は 2 GB です。

### 手順

1. **REGION** パラメーターで要求する 24 ビット (16 MB 境界より下) ストレージと 31 ビット (16 MB 境界より上) ストレージの量を計算します。

[REGION の見積もりおよび設定](#) を参照してください。

2. CICS 領域に適切な **REGION** パラメーターを指定します。

**REGION** は、次のように設定することができます。

- CICS JCL の JOB ステートメントで **REGION** パラメーターを指定できます。この場合、ジョブの各ステップは要求されたスペース量で実行されます。
- CICS の EXEC ステートメント (プログラム実行の行) で **REGION** パラメーターを指定できます。この場合、各ステップは独自のスペース量で実行されます。さまざまなステップで必要とされるスペース量が大きく異なる場合は、JOB ステートメントの代わりに EXEC ステートメントを使用してください。例えば、CICS のシャットダウン後に補助トレース・データ・セットを印刷するための追加のジョブ・

ステップを使用している場合 (DFHIVPOL インストール検査手順の場合のように)、EXEC ステートメントを使用します。

- z/OS インストール・システム 出口 IEFUSI は、指定する **REGION** 値を制限することがあります。IEFUSI については、[『z/OS MVS Installation Exits』](#) の『**IEFUSI - 手順開始出口**』を参照してください。

**REGION** パラメーターの指定方法について、および z/OS がユーザーの要求に応じて割り振るストレージの量について詳しくは、[『z/OS MVS JCL Reference』](#) の『**REGION パラメーター**』を参照してください。

3. **MEMLIMIT** パラメーターで指定する必要がある 64 ビット (2 GB 境界より上) ストレージの量を計算します。

この量は、使用する予定の機能によって異なります。[「パフォーマンスの改善」](#)の『**MEMLIMIT の見積もり、確認、および設定**』を参照してください。

4. 自身の CICS 領域に適した **MEMLIMIT** 値を割り出し、必要に応じて変更します。

**MEMLIMIT** は、次のように設定することができます。

- **MEMLIMIT** 値は、CICS JCL の JOB ステートメント、または CICS の EXEC ステートメント (プログラム実行の行) で指定できます。
- CICS 領域に固有の **MEMLIMIT** 値がない場合は、z/OS SMFPRMxx PARMLIB メンバーで設定された **MEMLIMIT** 値、またはシステム・デフォルトが適用されます。
- z/OS インストール・システム 出口 IEFUSI は、他の任意の **MEMLIMIT** 値をオーバーライドできます。

次の例は、プログラム実行の行で設定された **MEMLIMIT** 値を示しています。

```
//CICS EXEC PGM=DFHSIP,PARM='SI',REGION=0M,MEMLIMIT=10G
```

z/OS で **MEMLIMIT** 値を変更する手順や、セットアップしている CICS 領域に適した値を確認する手順については詳しくは、[『z/OS MVS P プログラミング: 拡張アドレッシング機能ガイド』](#) の『**専用メモリー・オブジェクトの使用制限**』を参照してください。

## サンプル始動ジョブ・ストリームの使用

サンプル・ジョブ制御ステートメントを使用して、CICS 領域の初期設定を行うことができます。

### このタスクについて

サンプル始動ジョブ・ストリームは、CICS 提供のサンプル・テーブル、DFHSIT6\$ に含まれるシステム初期設定パラメーターに基づいています。

CICS および IMS で必要なこのジョブ・ストリームの DD ステートメントについて詳しくは、[41 ページの『データ・セットの定義』](#)の該当するセクションを参照してください。

143 ページの『[サンプル CICS 始動ジョブ](#)』の JCL に類似した JCL は、サンプル始動プロシージャ、DFHSTART として、CICSTS56.CICS.SDFHINST ライブラリーに提供されています。DFHSTART プロシージャを、CICS 領域を開始するために使用することも、独自の始動プロシージャの基礎として使用することもできます。DFHSTART プロシージャについて詳しくは、[『インストール』](#)の『**CICS 始動プロシージャ DFHSTART**』を参照してください。

### サンプル CICS 始動ジョブ

サンプル CICS 始動ジョブを使用して、CICS 始動パラメーターに必要な値の決定および調整に役立てることができます。

EXEC ステートメントには、CICS 領域のサイズを定義する **REGION** パラメーターが含まれています。この例では、値を 240 に設定し、16 MB 境界の下にある 16 MB の専用ストレージのすべてと、240 MB の拡張領域サイズをジョブに割り振るように MVS に要求します。



```

/*
/** 1 The JOB statement
//CICSRUN JOB accounting info,name,CLASS=A,
// MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=userid
/*
/** 2 The JOBPARM statement
/*JOBPARM SYSAFF=sysid
/*
//*****
//***** EXECUTE CICS *****
//*****
/*
/** 3 The EXEC CICS=DFHSIP statement
//CICS EXEC PGM=DFHSIP,REGION=240M,
/** 4 System initialization parameters specified on PARM parameter
// PARM=('SIT=6$',
// 'DSALIM=6M,EDSALIM=120M',
// 'RENTPGM=PROTECT,STGPROT=YES',
// 'START=AUTO,SI')
/*
/** 5 System initialization parameters specified on the SYSIN data set
//SYSIN DD *
GRPLIST=(DFHLIST,userlist1,userlist2),
LPA=YES,
APPLID=CICSHTH1,
DFLTUSER=CICSUSER, The default user ID
MXT=30, Maximum number of user tasks is 30
INITPARM=(DFHDBCON='01',DFHD2INI=('MYDB')),
Pass DFSPZP01 suffix to DBCTL connect program
Connect to DB2 subsystem MYDB
ISC=YES, Include intersystem communication program
IRICSTR=YES, Start interregion communication
.END
/*
/** 6 The STEPLIB library
//STEPLIB DD DSN=CICSTS56.CPSM.SEYUAUTH,DISP=SHR
// DD DSN=CICSTS56.CICS.SDFHAUTH,DISP=SHR
// DD DSN=CICSTS56.CICS.SDFJAUTH,DISP=SHR
// DD DSN=CEE.SCEERUN2,DISP=SHR
// DD DSN=CEE.SCEERUN,DISP=SHR
/*
/** 7 The CICS library (DFHRPL) concatenation
//DFHRPL DD DSN=CICSTS56.CICS.SDFHLOAD,DISP=SHR
/** Your application library
// DD DSN=your.prog.library,DISP=SHR
/** Your CICS control tables library
// DD DSN=your.table.library,DISP=SHR
/** The Language Environment runtime data sets
// DD DSN=CEE.SCEECICS,DISP=SHR
// DD DSN=CEE.SCEERUN2,DISP=SHR
// DD DSN=CEE.SCEERUN,DISP=SHR
/** The Debug Tool runtime library
// DD DSN=EQA.SEQAMOD,DISP=SHR
/** 7a The DB2 load library
// DD DSN=SYS2.DB2.SDSNLOAD,DISP=SHR
/** 8 Auxiliary temporary storage data sets
//DFHTEMP DD DSN=CICSTS56.CICS.CNTL.CICSHTH1.DFHTEMP,DISP=SHR
/*
/** 9 Intrapartition data sets
//DFHINTRA DD DSN=CICSTS56.CICS.CNTL.CICSHTH1.DFHINTRA,DISP=SHR
/*
/** 10 The auxiliary trace data sets
//DFHAUXT DD DSN=CICSTS56.CICS.CICSHTH1.DFHAUXT,DISP=SHR
//DFHBUXT DD DSN=CICSTS56.CICS.CICSHTH1.DFHBUXT,DISP=SHR
/*

```

図 28. CICS 始動ジョブ・ストリーム

```

/* * 11      Extrapartition data sets
//DFHCXRF DD SYSOUT=*
//LOGUSR DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//MSGUSR DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=140)
//COUT DD SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=137)
//CEEMSG DD SYSOUT=A
//CEEOUT DD SYSOUT=A
/* *
/* * 12      The CICS local catalog data set
//DFHLCD DD DSN=CICSTS56.CICS.CICSHTH1.DFHLCD,DISP=SHR
/* * 13      The CICS global catalog data set
//DFHGCD DD DSN=CICSTS56.CICS.CICSHTH1.DFHGCD,DISP=SHR,
// AMP=('BUFND=33,BUFNI=32,BUFSP=1114112')
/* *
/* * 14      The transient data destination data set (CXRF)
//DFHCXRF DD SYSOUT=A
/* *
/* * 15      The CICS transaction dump data sets
//DFHDMPA DD DSN=CICSTS56.CICS.CICSHTH1.DFHDMPA,DISP=SHR
//DFHMPB DD DSN=CICSTS56.CICS.CICSHTH1.DFHMPB,DISP=SHR
/* *
/* * 16      MVS system dump data sets
//SYSABEND DD SYSOUT=*
/* *
/* * 16a     Default stdout file for C-language system services used by CICS
//SYSPRINT DD SYSOUT=*
/* * 17      The CICS system definition data set
//DFHCSD DD DSN=CICSTS56.CICS.DFHCSD,DISP=SHR
/* *
/* * 18      The CICS BTS local request queue data set
//DFHLRQ DD DSN=CICSTS56.CICS.DFHLRQ,DISP=SHR
/* *

/* * 19      The CMAC data file
//DFHCMACD DD DSN=CICSTS56.CICS.DFHCMACD,DISP=SHR
/* *
/* * 20      The CDBM group command file
//DFHDBFK DD DSN=CICSTS56.CICS.DFHDBFK,DISP=SHR
/* *
/* * 21      FILEA & other permanently allocated data sets
/* * (The FILEA DD statement below overrides the CSD definition in
/* * group DFHMROFD)
//FILEA DD DISP=SHR,
// DSN=CICSTS56.CICS.CICSHTH1.FILEA
/* *
//APPLICA DD DSN=CICSTS56.CICS.CICSHTH1.APPLICA,DISP=SHR
//APPLICB DD DSN=CICSTS56.CICS.CICSHTH1.APPLICB,DISP=SHR
/* *
/* *
/* * 22
//PRINTER DD SYSOUT=A,DCB=BLKSIZE=125,OUTLIM=0
/* * User transactions input from sequential terminal
//CARDIN DD *
CESF GOODNIGHT¥
/* *

```

1. この JOB ステートメントは、CICS のこの実行に使用するアカウント情報指定をします。例えば、次のようになります。

```

//CICSRUN JOB 24116475,userid,MSGCLASS=A,MSGLEVEL=(1,1),
// CLASS=A,NOTIFY=userid

```

2. この JES JOBPARM ステートメントは、この CICS (アクティブな CICS 領域) が実行される MVS イメージを識別するために組み込まれています。
3. DFHSIP は、CICS 初期設定を始動するプログラムです。CICS は、同一の MVS ジョブで複数の EXEC PGM=DFHSIP ジョブ・ステップはサポートしていません。

割り振られた専用ストレージのうち、CICS が動的ストレージ域に割り振る量と、オペレーティング・システム・ストレージの要求のために残す量を決定するには、**DSALIM** および **EDSALIM** システム初期設定パラメーターの値を設定します。**REGION** パラメーターによって定義された合計量から DSA で必要なスペース量を取得した後は、オペレーティング・システム・ストレージの要求を満たすために残りのストレージを使用できます。

このサンプル・ジョブ・ストリームでは、これらのシステム初期設定パラメーターを **PARM** パラメーターで指定しています。

**REGION** パラメーターおよび CICS ストレージの詳細については、[142 ページの『CICS 領域のアドレス・スペース・ストレージ制限の設定』](#)を参照してください。

RACF サポートを使用して CICS を実行する場合は、RACF に関連するパラメーターについて、[CICS TS セキュリティー](#)で確認してください。

4. 例に示されているように、EXEC ステートメントの **PARM** パラメーターを使用してシステム初期設定パラメーターを指定できます。

**PARM** パラメーターによって受け渡す情報は、100 文字までに制限されています。この制限にはすべてのコンマが含まれますが、**PARM** スtring を区切っているアポストロフィ、および **PARM** パラメーターを区切っている左括弧と右括弧は除外します。システム初期設定オペランドを囲む内部の括弧は含まれます。始動時に指定するシステム初期設定パラメーターが 100 文字では足りない場合は、**PARM** フィールドの最後に「SYSIN」または「CONSOLE」制御キーワード (または短縮形の「SI」か「CN」) を指定して、パラメーターが続くことを表します。SYSIN を指定するとシステム初期設定パラメーターが SYSIN データ・セットから読み込まれ、CONSOLE を指定するとコンソールからパラメーターを入力するよう CICS がプロンプトを出します。ただし、ランタイム・システム初期設定パラメーターすべてが **PARM** パラメーターにある場合には、PARM フィールドを制御キーワードなしで終わるか、END 制御キーワードで終了できます。

この例では、CICS は、DFHSIT6\$ システム初期設定テーブル内の値を使用しますが、それらは PARM フィールドおよび SYSIN データ・セットに指定したシステム初期設定パラメーターによって変更されます。この例では、以下のシステム初期設定パラメーターが **PARM** パラメーターで提供されています。

#### RENTPGM

読み取り専用 DSA、RDSA、および ERDSA のストレージは、**RENTPGM** システム初期設定パラメーターのデフォルトの PROTECT オプションを使用して、キー 0 の非取り出し保護ストレージから取得されます。開発 CICS 領域には RENTPGM=NOPROTECT を、実動 CICS 領域には RENTPGM=PROTECT をそれぞれ指定します。**RENTPGM** パラメーターの詳細については、[RENTPGM システム初期設定パラメーター](#)を参照してください。

#### STGPROT

**STGPROT** システム初期設定パラメーターで YES を指定するか、デフォルトの YES をそのまま受け入れると、CICS のこの実行に対するストレージ保護が得られます。**STGPROT** パラメーターについて詳しくは、[STGPROT システム初期設定パラメーター](#)を参照してください。

#### START

START=AUTO は実動 CICS システムに選択する標準的な始動タイプです。この場合、実行する始動のタイプを CICS が決定できます。

#### SI

PARM ステートメントを SI (SYSIN の短縮形) で終了し、SYSIN データ・セットからオーバーライドの読み取りを続けるように CICS に命令します。

- 5.

ジョブ・ストリームの一部として SYSIN データ・セットをインラインで含めることができます。SYSIN データ・セットに入力したシステム初期設定パラメーターによって、同じキーワードで **PARM** パラメーターに入力したものが置き換えられます。同じパラメーターを複数回含めている場合には、最後に読み込まれた値が初期設定に使用される値となります。ただし INITPARM は例外です INITPARM キーワードとそのパラメーターを複数回指定すると、それぞれが CICS によって受け入れられ、次のようになります。

```
* The following INITPARM parameters are for DBCTL and a user program
INITPARM=(DFHDBCON='XX,DBCON2',userprog='a,b,c')
* The following INITPARM parameter is for DB2
INITPARM=(DFHD2INI='DBA2')
```

システム初期設定の制御キーワード CONSOLE を明示的にコーディングしない場合には、CICS が SYSIN の最後または END 制御キーワードに達するとシステム初期設定パラメーターの読み取りを停止します。

サンプル・ジョブでは、CONSOLE は PARM または SYSIN のどちらにおいてもコーディングされていません。END 制御キーワードは SYSIN における最後のエントリーなので、CICS がさらにシステム初期設定パラメーターを求めてコンソールでプロンプトを出すことはありません。SYSIN データ・セットの読み取りが終わると、CICS は指定の SIT をロードし、PARM フィールドと SYSIN データ・セットで提供されているシステム初期設定パラメーターを適用して、初期設定処理を開始します。

この例の SYSIN データ・セットには、以下のいくつかのシステム初期設定パラメーターが含まれています。

#### GRPLIST

DFHSIT6\$ に定義されているグループ・リストは DFHLIST です。これは、DFHCSDUP INITIALIZE コマンドを使用して CSD を初期設定する際に生成されるリストです。DFHLIST には、CICS で必要な標準リソース定義のみが含まれています。GRPLIST システム初期設定パラメーターで指定したいいずれかのグループ・リストに、CICS で必要なこうしたリソース定義が含まれている必要があります。こうしたリソース定義を独自のグループ・リストの 1 つに含めるか、DFHLIST を明示的に指定することができます。独自のグループ・リスト (この例では userlist1 と userlist2) には、CICS アプリケーションのインストールで生成されるすべてのリソース定義が含まれていなければなりません。また、使用する IBM ライセンス・プログラム (COBOL や Db2 など) に必要な定義もグループ・リストに含まれている必要があります。

#### LPA

SIT では、リンク・パック域からモジュールを使用されないように指定します。SYSIN で LPA=YES を指定すると、この実行では LPA からモジュールが使用されます。

#### APPLID

この CICS 領域の APPLID は CICSHTH1 です。この CICS 領域で端末アクセスまたは ISC 通信に z/OS Communications Server を使用する場合には、この APPLID を z/OS Communications Server に定義する必要があります。

CICSID はこの CICS 領域の汎用 APPLID で、CICSHTH1 はアクティブ CICS 領域の固有 APPLID です。

固有なデータ・セット (例えば、ダンプ・データ・セット) に名前を付ける際に、固有 APPLID が役立ちます。

#### CICSSVC

245 は、LPA にインストールされ、SVCPARM ステートメントの MVS に定義されている CICS タイプ 3 SVC 番号です。詳細については、[CICSSVC システム初期設定パラメーター](#)を参照してください。

LPA での CICS SVC のインストール、およびその MVS への定義に関するガイド情報については、『インストール』の『CICS SVC のインストール』を参照してください。

#### DFLTUSER

CICSUSER は、RACF に指定されているデフォルトのユーザー ID です。始動時に、CICS はデフォルトのユーザー ID をサインオンしようとします。サインオンできない場合 (例えば、定義されていない場合) には、CICS はメッセージを出して、CICS 初期設定を停止します。有効なデフォルトのユーザー ID がサインオンされた後は、そのセキュリティ属性が、CESN トランザクションでサインオンしないすべての CICS 端末ユーザーに使用されます。デフォルト・ユーザー ID が CICS セグメントと一緒に RACF に定義される場合は、そのセグメントのオペレーター属性も、サインオンしないユーザーに使用されます。

#### MXT

この実行でのユーザー・タスクの最大数は、30 に制限されています。MXT パラメーターに含まれているタスクについては、[MXT システム初期設定パラメーター](#)を参照してください。

#### INITPARM

パラメーターをプログラムに渡します。この例では、DFSPZPxx の DBCTL suffix (01) が DFHDBCON に渡され、CICS は Db2 サブシステム MYDB に接続するよう通知を受けます。

INITPARM システム初期設定パラメーターで Db2 サブシステムのデータ共用グループの ID を使用することはできません。単一の Db2 サブシステムの ID を指定する必要があります。INITPARM システム初期設定パラメーターを使用して Db2 サブシステムを指定する場合には、インストール済みの DB2CONN 定義の DB2GROUPID と DB2ID を両方ともブランクのままにしてください。

DB2CONN 定義のこれらの属性のいずれかで指定された ID は、**INITPARM** システム 初期設定パラメーターで指定した ID を指定変更します。

## ISC

領域間通信 (IRC) を使用するためシステム間連絡プログラム DFHISP を組み込みます。

## IRCSTRT

この CICS は MRO を使用して実行中で、初期設定時に領域間通信が開始されます。

- STEPLIB は、オペレーティング・システムによってロードされるモジュールが含まれるライブラリーの DDNAME です。STEPLIB からロードされる DFHSIP は許可された状態で制御を受け取る必要があるため、STEPLIB に連結された個々の区分データ・セット (ライブラリー) がそれぞれ APF 許可されていなければなりません。このサンプル・ジョブ・ストリームでは、CICS 許可ライブラリーは CICSTS56.CICS.SDFHAUTH です。CICSTS56.CICS.SDFJAUTH も組み込まれています。これは、Java サポートのために必要で、APF 許可されていることも必要となります。Java サポートが不要な場合は、このライブラリーを組み込まないでください。連結には CICSTS56.CPSM.SEYUAUTH も組み込まれています。これは、領域が管理アドレス・スペースとして実行されない場合であっても、初期設定時に CICS に CICSplex SM メッセージ・ファイルをロードさせるためです。

ユーザー作成モジュールを APF 許可ライブラリーに配置すると、セキュリティおよび保全性の規則に違反する恐れがあります。CICSTS56.CICS.SDFHAUTH ライブラリー (および、使用している場合には CICSTS56.CICS.SDFJAUTH ライブラリー) は MVS リンク・リストに含めないでください。これらのライブラリーを保護し、承認されたユーザーのみが DFHSIP モジュールを実行できるようにするためです。STEPLIB DD ステートメントを定義するときには、CICSTS56.CICS.SDFHAUTH ライブラリーまたは CICSTS56.CICS.SDFJAUTH ライブラリーに連結された他のすべてのライブラリー (例えば、IMS.RESLIB) に APF 許可を与える必要があることを忘れないでください。APF 許可が必要な理由は、許可されていないライブラリーが STEPLIB 連結内に含まれていると、MVS がすべてのライブラリーを無許可と見なすからです。STEPLIB 連結内に無許可のライブラリーが存在すると、CICS 始動が失敗し、DFHSIP モジュールが APF 許可ライブラリー内にないことを示すメッセージ DFHKE0101 が発行されます。CICS データ・セットへのアクセスの許可については、[RACF データ・セット・プロファイル](#)を参照してください。

許可属性 (SETCODE AC(1)) を使用してリンク・エディットされた事前生成 DFHSIP モジュールは、CICSTS56.CICS.SDFHAUTH にあります。

Language Environment で COBOL、PL/I、C、C++、および、JVM で実行される Java プログラムを実行するには、Language Environment ランタイム・ライブラリー CEE.SCEERUN と CEE.SCEERUN2 を STEPLIB 連結または MVS リンク・リストに含める必要があります。SDFHAUTH 同様、SCEERUN と SCEERUN2 も APF 許可ライブラリーでなければなりません。

CICS Db2 接続機能では、Db2 プログラム要求処理ハンドラー DSNAPRH をロードする必要があります。Db2 ライブラリー DSNxxx.SDSNLOAD を MVS リンク・リストに含めるか、CICS ジョブの STEPLIB 連結に追加することができます。

- DFHRPL は、CICS がロードするモジュールが含まれるライブラリーの DD 名です。このライブラリーを構成する個々の区分データ・セットを保護して、その内容が承認のないまま、または誤って変更されることがないようにできます。DFHRPL 連結には、CICS アプリケーション・プログラムが含まれているライブラリー (この例では `your.prog.library` と示されています)、および CICS 管理テーブル (この例では `your.table.library` と示されています) を組み込む必要があります。

このサンプル・ジョブ・ストリームでは、提供ライブラリーは CICSTS56.CICS.SDFHLOAD で、CICS DFHRPL ライブラリー連結に組み込まなければなりません。CICSTS56.CICS.SDFHLOAD ライブラリーには、問題が発生した状態で実行するプログラムしか含まれていないので、このライブラリーは許可しないでください。

## Language Environment ランタイム・ライブラリーの要件

Language Environment のサービスを使用します。これは、アセンブラーおよび CICS がサポートする高水準言語 (HLL) (すなわち、COBOL、PL/I、C、および C++) の IBM インプリメンテーション用の共通ランタイム環境を提供します。

ご使用のすべてのプログラム言語をサポートするために Language Environment ランタイム・ライブラリーを使用するには、Language Environment ランタイム・ライブラリー SCEERUN2 と SCEERUN を DFHRPL DD 連結に追加します。COBOL プログラムを実行する場合、Language Environment ランタイ



ム・ライブラリー SCEECICS も DFHRPL に追加してください。SCEECICS ライブラリーは、SCEERUN ライブラリーの前に連結しなければなりません。DFHRPL DD 連結から、以前のバージョンの COBOL、PL/I、および C/C++ のランタイム・ルーチンが含まれるライブラリーを削除します。

## デバッグ・ツール・ライブラリー

CICS アプリケーションをデバッグするためにデバッグ・ツールを使用する場合、デバッグ・ツール・ライブラリー SEQAMOD を DFHRPL に組み込みます。CICS でのデバッグ・ツールの使用については、[CICS アプリケーションでのデバッグ・ツールの使用](#)を参照してください。

### 7a Db2 要件

一般に、DFHRPL DD ステートメントには Db2 ライブラリーを組み込む必要はありません。アプリケーションのために DFHRPL 連結に Db2 ライブラリーが必要な場合は、CICS ライブラリーの後に指定する必要があります。例えば、Db2 メッセージ処理モジュール DSNTIAR、またはその後の DSNTIA1 (両方とも SDSNLOAD に用意されています) の動的呼び出しを実行するアプリケーションをサポートするためには、DFHRPL に SDSNLOAD を追加します。DSNTIA1 は、プログラム DSNTIAC に **EXEC CICS LOAD** コマンドを発行する Db2 アプリケーション・スタブ DSNTIAC が含まれているアプリケーション・プログラムによってロードされます。

8. 後で使用するためにデータを保管する場合には、このデータ・セットを定義します。シンボル名によって識別される、使用中の一時記憶域キューは、明示的に削除するまで存続します。親タスクが削除された後でさえ、保管時に使用したシンボル名を参照することによって、他のタスクは一時データにアクセスできます。

こうしたデータ・セットの定義方法の詳細、およびスペース計算については、[48 ページの『一時記憶域データ・セットのセットアップ』](#)を参照してください。

一時記憶域のデータ共用を使用する場合は、必ず、CICS 領域に一時記憶域サーバーが必要になる前に、そのサーバーを始動しておいてください。

一時記憶域サーバーおよび一時記憶域のデータ共用について詳しくは、[166 ページの『一時記憶域サーバーのセットアップと実行』](#)を参照してください。

9. 一時データ区画内データ・セットは、CICS 領域内のメッセージとデータのキューイングに使用されます。こうしたデータ・セットの定義方法、およびスペース計算については、[52 ページの『区画内データ・セットの定義』](#)を参照してください。
10. 補助トレースを使用する場合には、こうした順次データ・セットの 1 つまたは両方を定義します。補助トレース・データ・セットの自動切り替えを定義する場合、両方のデータ・セットを定義します。データ・セットを 1 つだけ定義する場合には、その DD 名は DFHAUXT としてください。

こうしたデータ・セットの定義方法の詳細は、[76 ページの『補助トレース・データ・セットの設定』](#)を参照してください。

このジョブ・ストリーム内の補助トレース・データ・セットはアクティブ CICS 領域に固有のものであるため、この例では、アクティブ CICS 領域 (CICS1H1) の固有 APPLID を第 2 レベルの修飾子として使用して指定しています。

[78 ページの図 16](#) に示されているように、ディスク上に補助トレース・データ・セットを割り振ってカタログすると、以下の DD ステートメントを使用して始動ジョブ・ストリームでこれらのデータ・セットを CICS に定義できます。

```
//DFHAUXT DD DSN=CICSTS56.CICS.applid.DFHAUXT,DCB=BUFNO=n,DISP=SHR
//DFHBUXT DD DSN=CICSTS56.CICS.applid.DFHBUXT,DCB=BUFNO=n,DISP=SHR
```

1 より大きい BUFNO を指定すると、補助トレース・レコードの書き込みに関連する I/O を減らせます。4 から 10 までの範囲の値を指定すると、補助トレースを使用して実行する場合の I/O を大幅に減らせます。

11. 区画外一時データ・キューの例には、LOGA と CSSL があります。
  - LOGA は、CICS サンプル・プログラムが使用するユーザー・データ・セットを定義します。
  - CSSL は、いくつかの CICS サービスが使用するデータ・セットを定義します。
  - CCSO は、C/370 アプリケーション・プログラムが実行されている場合に限り出力キューとして使用されます。



- CESO は、Language Environment でアプリケーション・プログラムが実行されている場合に限ってエラー・キューとして使用されます。

CICS が使用するキューのサンプル定義が、グループ DFHDCTG で提供されています。DFHDCTG はロックを解除されているので、インストール前に定義を変更できます。

12. CICS ローカル・カタログは、CICS ドメインで使用され、CICS の実行と実行の間で情報の一部を保管し、この情報をコールド・スタートが行われても保持します。ローカル・カタログを、他の CICS システムが共用することはありません。CICS ローカル・カタログの作成および初期設定の方法について詳しくは、[66 ページの『カタログ式データ・セットの設定』](#)を参照してください。
13. グローバル・カタログは 1 つのみ存在します。

CICS グローバル・カタログの作成および初期設定の方法について詳しくは、[66 ページの『カタログ式データ・セットの設定』](#)を参照してください。

このサンプル・ジョブでは、DD ステートメント上の **AMP** パラメーターの使用法を例示しています。このパラメーターを、そのバッファー・サブパラメーターと一緒に指定すると、再始動時間とシャットダウン時間を短縮するのに役立ちます。この例では、[68 ページの図 13](#) に示している **DEFINE CLUSTER** ステートメントと、[66 ページの『グローバル・カタログの定義』](#) の 4 の下の関連する記載事項を使用しています。指定されている値は、これらのパラメーターの最小値として適した値であるため、減らさないでください。

14. この一時データ宛先は、CICS が区画内一時データの初期設定を完了する前に一時データ宛先に送信するメッセージのターゲットとして CICS によって使用されます。
15. CICS は、1 つの順次データ・セット、または対になった順次データ・セット、磁気テープ、ディスクにトランザクション・ダンプを記録します。データ・セットは DD 名 DFHDMPA と DFHDMPB で定義する必要がありますが、片方だけのデータ・セットを定義する場合にはその DD 名は DFHDMPA としてください。CICS は、初期設定時には必ず少なくとも 1 つのトランザクション・ダンプ・データ・セットをオープンしようとしています。

CICS トランザクション・ダンプ・データ・セットの定義と使用の方法について詳しくは、[78 ページの『ダンプ・データ・セットの定義』](#)を参照してください。

このジョブ・ストリーム内のトランザクション・ダンプ・データ・セットはアクティブ CICS 領域に固有のものであるため、アクティブ CICS 領域 (DBDCCIC1) の固有 APPLID を第 2 レベルの修飾子として使用して指定しています。

16. MVS にダンプを生成させるには、SYSABEND DD ステートメントを使用します。

ユーザー域とシステム域のダンプを生成します。このダンプには、SYSUDUMP でダンプされたすべての領域と (サブプール 229 と 230、および障害が発生したタスク用の入出力システム (IOS) 制御ブロックを含む) ローカル・システム・キュー域 (LSQA) が含まれます。ダンプは定様式なので、直接印刷できます。MVS は、以下のいずれかが真のときに要求されたダンプを生成します。

- CICS が異常終了した。
- CICS が異常終了を開始したが、システム・リカバリー手順によって CICS を正常終了できた。

ダンプ内容は、そのダンプに IBM 提供のデフォルトが使用される場合にのみ、説明されているようになります。こうしたダンプの内容は MVS システム初期設定時に設定でき、ABEND マクロ命令、CHNGDUMP コマンドにおいて、さらには SLIP コマンドを使用して個々のダンプ用に変更できます。詳しくは、[z/OS MVS 初期設定およびチューニングガイド](#)を参照してください。

ダンプはオプションですので、ダンプを生成する場合に限りダンプ DD ステートメントを使用してください。

MVS システム・ダンプ・データ・セットの定義方法、およびそれらからダンプを印刷する方法については、[z/OS MVS JCL 解説書](#)を参照してください。ダンプを解釈する方法については、[z/OS MVS 診断ツールと保守援助プログラム](#)を参照してください。

### 16a CICS で使用される C 言語システム・サービスのデフォルト stdout ファイル

SYSPRINT ステートメントは、CICS が使用する C 言語システム・サービス・ルーチン (システム SSL や JVM サーバーなど) によって stdout ファイルとしてオープンされます。このステートメントが省

略されると、代わりにオペレーティング・システムによって複数の sysout ファイルが動的に割り振られます。

17. システム定義ファイル (CSD) は、一部のリソース定義を保持するために CICS で必要とされます。

CSD にジョブ制御 DD ステートメントを提供することもできます。その場合、CSD データ・セットは CICS ジョブ・ステップ開始時に割り振られ、CICS ジョブ・ステップ期間中も割り振られたままになります。

CSD に対して動的割り振りを使用したほうがよい場合もあります。動的割り振りの場合、CSD に DD ステートメントを指定しないでください。SET FILE コマンドまたはシステム初期設定 (CSDDSN と CSDDISP パラメーターとして) のいずれかで、データ・セット名 (DSNAME) およびデータ・セット処理 (DISP) を指定します。CICS は、DSNAME と DISP を使用して OPEN 処理の一部としてファイルを割り振ります。

CSD の作成と初期設定については、21 ページの『第 3 章 共用データ・セット、CSD、および SYSIN のセットアップ』を参照してください。

18. DFHLRQ はファイル制御管理対象の VSAM キー順データ・セット (KSDS) で、CICS ビジネス・トランザクション・サービス (BTS) によって使用されます。DFHLRQ のために用意されている FILE リソース定義は、CSD を初期設定またはアップグレードするときに DFHLIST グループ・リストに自動的に入れられる CSD グループ DFHCBTS に含まれています。この FILE 定義には、初めて参照されたときに開くように指定されています。この参照が行われるのは、CICS 初期設定の最後に BTS によって開かれるときです。CICS は、データ・セットが見つからない場合には警告メッセージ DFHFC0951 と DFHSH0109 を発行します。CICS はこのデータ・セットがなくても実行を継続しますが、BTS を使用しない場合であっても DFHLRQ データ・セットを定義してください。DFHLRQ については、BTS の概要を参照してください。
19. DFHCMACD は、CICS メッセージおよびコードのオンライン記述を提供するために CMAC トランザクションによって使用される VSAM キー順データ・セット (KSDS) です。初めて使用される前に、KSDS データ・セットとして定義し、ロードする必要があります。DFHCMACD については、96 ページの『CMAC メッセージ・データ・セットの定義』で詳しく説明しています。
20. DFHDBFK は、Group コマンドを保管するために CDBM トランザクションが使用する VSAM キー順データ・セット (KSDS) です。初めて使用する前に、KSDS データ・セットとして定義することが必要です。DFHDBFK DD ステートメントは、CDBM トランザクションのコマンド・ストレージ機能を使用する予定の場合にのみ必要です。CDBM GROUP コマンド・データ・セットの定義には、DFHDBFK がより詳細に説明されています。
21. CSD で定義されているユーザー・ファイル (RDO を使用している場合) またはファイル制御テーブルで定義されているユーザー・ファイル (BDAM ファイルのみ) に、ジョブ制御 DD ステートメントを提供したい場合もあります。その場合、CICS ジョブ・ステップ開始時にデータ・セットが割り振られ、CICS ジョブ・ステップ期間中も割り振られたままになります。FILEA (サンプル・アプリケーション・プログラム用のデータが含まれている分散ライブラリー) を、ジョブ制御ステートメントによる直接割り振りの例として、この始動ジョブ・ストリームに含めています。

代わりに、CICS によるファイルの動的割り振りを利用することが好ましい場合もあります。動的割り振りの場合、CSD に DD ステートメントを指定しないでください。CICS は、FILE リソース定義の DSNAME パラメーターに指定されている完全なデータ・セット名 (最大 44 文字) を DISP パラメーターと共に使用して、OPEN 処理の一部としてファイルを割り振ります。この形式の動的割り振りは、明示的にオープンされるよう定義されるファイル、およびアプリケーションによって初めて参照される際にオープンされるファイルに同じように当てはまります。ファイル・オープンの詳細については、82 ページの『ユーザー・ファイルの定義』を参照してください。FILE リソースで定義可能な属性については、FILE リソースを参照してください。

このサンプル・ジョブ・ストリームに示しているカード・リーダー/ライン・プリンター (CRLP) のシミュレート端末は、サンプル TCT に定義されています (この始動ジョブでは使用されません)。そうした装置に必要なソース・ステートメントについては、CICSTS56.CICS.SDFHSAMP 内の COPY メンバー DFH\$TCTS を参照してください。

22. 順次装置の場合、入力ストリーム内の最終エンタリーには CESF GOODNIGHT¥ が可能で、論理クローズを提供して、装置を静止します。ただし、こうした方法で装置をクローズすると、CICS シャットダウン時のウォーム・キーポイントに受信専用状況が記録されます。それにより、その端末はそれ以降のウォーム・スタートでもまだ RECEIVE 状況であるため、CICS は入力ファイルを読み込みません。

サンプルにおける各行の最後にあるデータ終了文字(「¥」シンボル)に注目してください。

## サンプル CICS 始動プロシージャ

MVS 内部読み取りプログラムにバッチ・ジョブを実行依頼する代わりに、MVS **START** コマンドを使用して、CICS を開始タスクとして始動できます。この方法を使用する場合、プロシージャのコーディングに関する規則に従って始動ジョブ・ストリームをコーディングする必要があります。また、そのプロシージャを MVS プロシージャ・ライブラリーにインストールする必要があります。

### このタスクについて

**START** コマンドを使用して CICS を始動する場合、以下のいずれかが必要になります。

- MVS 開始タスク・プロシージャに、IEFSSNaa のサブシステム名とは異なる名前を付ける (デフォルトは「CICS」)。
- 該当するパラメーター SUB=JES2 または SUB=JES3 を指定して、開始コマンドを出す。

次の形式の MVS **START** コマンドを使用して、コンソールからジョブを開始できます。

```
S|START procname[.identifier][,SUB=subsystemname][,keyword=option  
[,keyword=option] . . .]
```

ここで、

#### procname

開始されるジョブを定義するカタログ式プロシージャの名前。

#### ID

タスクを識別するために選択した名前。

#### SUB=subsystemname

処理するジョブを選択するサブシステムの 名前。このパラメーターを省略した場合は、1 次ジョブ入力サブシステムが使用されます。

#### keyword=option

プロシージャ内の対応するパラメーターをオーバーライドするための適切なキーワード。このパラメーターを使用して、カタログ式プロシージャ内で定義されているシンボリック・パラメーターをオーバーライドすることができます。

START コマンドの完全な構文、および使用可能なすべてのキーワードとオプションに関するガイダンス情報については、「[z/OS MVS システム・コマンド](#)」を参照してください。

CICS を始動するために必要なコーディングは、**procname.identifier,keyword(s)=option** のみです。

例えば、次のようになります。

```
START DFHSTART.CICSA,SIP=T,REGNAME1=IDA,REGNAM2=IDA
```

この MVS **START** コマンドの例について、以下で説明します。

- DFHSTART は、CICS 提供のカタログ式始動プロシージャの名前です。
- CICS は、CICSA というタスク ID で開始されます。
- SIP は、この CICS 領域で使用する SYSIN データ・セット (CICSTS56.CICS.SYSIN) の DFH\$SIPx メンバーの接尾部です。
- REGNAM1 と REGNAM2 は、この CICS 領域でデータ・セットを一意に識別するために、プロシージャに指定した CICS システム・データ・セット (例えば、CICSTS56.CICS.CICSHTH1.DFHTEMP) に追加される修飾子です。REGNAM1 は、MRO 領域の REGNAM2 と同じ値に設定されます。

DFHSTART プロシージャについては、『インストール』の『[CICS 始動プロシージャ DFHSTART](#)』を参照してください。

RACF を使用して CICS を実行する場合、RACF テーブル ICHRIN03 で、適切な許可が与えられた RACF ユーザーとカタログ式プロシージャの名前を関連付ける必要があります。この関連付けについては、[RACF での CICS プロシージャの実行を許可する](#)を参照してください。

## デバッグ・ツールを使用するための CICS の準備

アプリケーション・プログラマーが特定のデバッグ・ツールを CICS で使用するためには、いくつかのシステム定義タスクを実行しておく必要があります。このセクションでは、それらのタスクについて説明します。

### デバッグのための CICS 領域の準備

アプリケーション・プログラマーが特定のデバッグ・ツールを CICS で使用するためには、そのための構成を CICS 領域に行っておく必要があります。

#### このタスクについて

以下のツールを使用して、デバッグ・アプリケーション・プログラムのために CICS 領域を準備できます。

- コンパイルされた言語アプリケーション・プログラム (COBOL、PL/I、C、C++ で作成されたプログラム) 用、および Language Environment 対応アセンブラー・サブルーチン用のデバッグ・ツール
- リモート・デバッグ・ツール (コンパイルされた言語アプリケーション・プログラム、Language Environment 対応アセンブラー・サブルーチン、および Java プログラム用)。コンパイルされた言語プログラムおよびアセンブラー・サブルーチンの場合、デバッグ・ツールはデバッグ・サーバーとして使用されます。

このトピックは、CICS 実行診断機能 (CEDF) などの他のデバッグ・ツールには適用されません。

領域を再始動する必要が生じるため、デバッグ・アプリケーションに使用する領域を計画しておくことをお勧めします。

- 実動領域でアプリケーション・プログラムをデバッグすることはあまりありません。アプリケーションが安定していて、信頼できることがわかっている場合は特にそうです。新しいアプリケーションの開発およびテストに使用される領域のほうが、デバッグが行われる可能性は高くなります。

デバッグのために CICS 領域を準備するには、次のようにします。

#### 手順

1. コンパイルされた言語プログラムのデバッグに領域を使用する場合は、CICS 始動 JCL の DFHRPL 連結にデバッグ・ツール・ライブラリー **SEQAMOD** を組み込みます。  
詳しくは、[143 ページの『サンプル始動ジョブ・ストリームの使用』](#)を参照してください。
2. デバッグ・プロファイル・データ・セットを作成します。  
詳しくは、[98 ページの『デバッグ・プロファイル・データ・セットのセットアップ』](#)を参照してください。
3. [GRPLIST システム初期設定パラメーター](#)で指定したリソース定義リストに、デバッグ・プロファイル・ファイルのリソース定義を含めます。
4. オプションで、[DEBUGTOOL システム初期設定パラメーター](#)に次の値を指定します。

```
DEBUGTOOL=YES
```

DEBUGTOOL=YES を指定しない場合、実行時に領域のデバッグを有効にできます。

- プログラムから領域のデバッグを有効にするには、**EXEC CICS SET SYSTEM DEBUGTOOL** コマンドを使用します。
- マスター端末トランザクションから領域のデバッグを有効にするには、**CEMT SET SYSTEM DEBUGTOOL** コマンドを使用します。

通常はデバッグに使用されない領域の場合は、実行時に領域のデバッグを有効にすることをお勧めします。デバッグが完了したら、同じコマンドを使用して領域のデバッグを無効にできます。



5. デバッグ・ツールのリソース定義を定義し、インストールします。それらは、デバッグ・ツールの SEQASAMP データ・セット内のメンバー EQACCSO にあります。詳しくは、[Debug Tool for z/OS](#) を参照してください。

## CICS 領域の始動

このセクションでは、CICS 領域の始動方法を説明しています。このセクションは、CICS のカスタマイズ、必要な追加サポートの生成、および必要なすべての CICS システム定義の定義が実行済みであることを想定しています。

### このタスクについて

CICS は以下の 2 つの方法で開始できます。

- MVS **START** コマンドを使用して、CICS を開始タスクとして開始します。
- CICS バッチ・ジョブを MVS 内部読み取りプログラムに実行依頼します。

いずれの方式でも、CICS 始動プロシージャで使用されるシステム初期設定パラメーターを指定することによって、CICS の開始方法と使用できる機能やリソースを決定します。通常は、CICS を開始する前にシステム初期設定パラメーターを指定します。しかし、CICS の初期設定の開始後に、始動前に指定したシステム初期設定パラメーターをオーバーライドできます。例えば、CICS の現時点の実行に特定の機能を使用できるようにオーバーライドできます。

### 手順

1. [始動前にシステム初期設定パラメーターを指定します。](#)
2. 以下のいずれかの方法で CICS を始動します。
  - [156 ページの『バッチ・ジョブとしての CICS の開始』](#)
  - [156 ページの『開始タスクとしての CICS の始動』](#)
3. オプション: CICS システム初期設定の開始後に、始動前に指定したシステム初期設定パラメーターをオーバーライドする場合は、[158 ページの『始動時のシステム初期設定パラメーターのオーバーライド』](#)の手順に従ってください。

### タスクの結果

システムの初期設定が完了すると、メッセージ DFHSD1517 が返され、制御権が CICS に付与されて CICS が端末要求を処理する準備できたことが示されます。

### ヒント:

#### **z/OS Workload Manager ヘルス・サービスを使用するシステム・ウォームアップ**

システムの初期設定の終了後に、CICS のウォームアップ・プロセスを許可することを考慮してください。

DFHSD1517 によってシステム初期設定の終了が通知されても、領域が作業を受け取る準備が完全にはできていない場合があります。その理由は、このメッセージが表示された後も初期設定が続く場合があります。例えば、バンドル定義リソースの中には、非同期的にインストールされるものがあります。このため、TCP/IP リスナーが開いていて作業を受け入れるようになっていても、一部のアプリケーションに必要な JVMSEVER などのリソースの初期設定がまだ完了していないことがあります。これが原因でトランザクションが異常終了する場合があります。Web サービスの場合、パイプライン・スキャンが完了していない可能性があるため、スキャンの完了前に Web サービスを使用すると失敗します。

**WLMHEALTH** システム初期設定パラメーターを設定すると、CICS でウォームアップ・プロセスを使用することができます。z/OS ワークロード・マネージャーのヘルス・サービスが領域でアクティブな場合は、メッセージ DFHSD1517 が返されたときに、ウォームアップ・プロセスが開始されます。CICS はその後断続的にその領域の z/OS WLM 正常性値を調整して、その領域が作業を完全に処理できるようになるまで、その領域に対する作業のフローを制御します。CICS ウォームアップ・プロセスについて詳しくは、[z/OS ワークロード・マネージャー正常性サービスの使用による CICS のウォームアップおよび冷却](#)を参照してください。

## CICS システム始動日時の照会

CICS 領域でコールド始動、緊急始動、初期始動、またはウォーム始動が最後に行われた日時を調べるには、**INQUIRE SYSTEM SPI** コマンドを使用できます。詳しくは、[INQUIRE SYSTEM](#) を参照してください。

## 始動前のシステム初期設定パラメーターの指定

普通は以下の方法で、CICS の開始前に、CICS が使用することになるシステム初期設定パラメーターを指定します。

### 手順

- CICS 始動プロシーチャーの STEPLIB 連結内のライブラリーからロードされたシステム 初期設定テーブル内で。
- CICS 始動プロシーチャーの EXEC PGM=DFHSIP ステートメントの **PARM** パラメーター内で。
- 始動プロシーチャーに定義されている SYSIN データ・セット内で (SYSIN が **PARM** パラメーター内でコード化されている場合のみ)。

### タスクの結果

上記に概説した順序でシステム 初期設定パラメーターが処理されます。後に指定されたシステム 初期設定パラメーターの値が前に指定された値をオーバーライドします。

### 例

例えば、CICS 始動プロシーチャーの指定が以下のようになっていますとします。

```
//INITCICS EXEC PGM=DFHSIP,REGION=&REG,  
//          PARM=('SYSIDNT=HTH1,SIT=6$,SYSIN,CN')  
//*  
//SYSIN DD DISP=SHR,DSN=&libpfx..CICSH###.SYSIN(CICS&CLONE)
```

CICS は以下のソースからシステム 初期設定パラメーターを使用します。後に指定されたシステム 初期設定パラメーターが前に指定されたパラメーターをオーバーライドします。

1. STEPLIB 連結からのシステム 初期設定テーブル DFHSIT6\$。
2. CICSTS56.CICS.CICSH###.SYSIN データ・セットのメンバー CICSH###。
3. システム・コンソール。

### 次のタスク

特に、**START** システム 初期設定パラメーターの新しい値を指定できます。

#### START システム初期設定パラメーター

**START** システム 初期設定パラメーターには、次のいずれかの値が含まれます。

##### START=AUTO

**START** システム 初期設定パラメーターにこの値を指定すると、CICS はグローバル・カタログ内の以下の 2 つのレコードを検査して、初期始動、コールド・スタート、ウォーム・スタート、緊急始動のどれを実行するか判別します。

- リカバリー・マネージャー制御レコード
- リカバリー・マネージャー自動始動オーバーライド・レコード

START=AUTO は、CICS によって自動的に始動が選択される、通常モードの操作です。

##### START=INITIAL

CICS の新しい実行には、前の実行に対する参照はありません。グローバル・カタログおよびシステム・ログが初期設定されると、その中の情報はすべて失われます。



## START=COLD

CICS の新しい実行には前の実行に対する制限付きの参照があり、同じグローバル・カタログとシステム・ログを使用します。特に、リモート・システムで作業単位を再同期するのに必要な再同期情報は保存されます。

## START=STANDBY

CICS は、アクティブな CICS 領域をモニターできるレベルに限定して初期設定し、XRF 代替 CICS 領域として始動します。アクティブな CICS 領域がシャットダウンされた方法に応じて、引き継ぎが必要な場合に代替 CICS 領域は以下のようにウォーム・スタートか緊急時再始動を完了します。

- **CEMT PERFORM SHUTDOWN TAKEOVER** コマンドが正常に完了してアクティブな CICS 領域がシャットダウンされた場合、代替 CICS 領域はウォーム・スタートを実行します。
- アクティブな CICS 領域が異常にシャットダウンされた場合、代替 CICS 領域は緊急時再始動を実行します。

注：XRF=YES システム 初期設定パラメーターも指定する必要があります。

## バッチ・ジョブとしての CICS の開始

### 手順

1. CICS を開始するジョブを作成します。このジョブで、次のいずれかを行うことができます。

- CICS 始動プロシーチャーをインラインで含めます。
- カタログ式の始動プロシーチャーを呼び出します。

後者の方式には、複数の CICS 始動ジョブ (CICS 領域が複数ある場合など) が同じプロシーチャーを使用し、始動パラメーターによりそのプロシーチャーを調整できるという利点があります。

例として、156 ページの図 29 は、カタログ式プロシーチャー CICSTASK を呼び出し、始動パラメーター SYSIDNT=HTH1 と CLONE=HT##1 を指定してコールド・スタートを実行する CICS 始動ジョブを示しています。SYSIDNT パラメーターと CLONE パラメーターを変更すると、同じジョブを使用し、同じプロシーチャーを使用して他の CICS 領域を始動することもできます。

```
//CIDCTOR JOB (accounting information),userid,MSGCLASS=A,MSGLEVEL=(1,1),
//          CLASS=C,NOTIFY=userid
//*****
//* THIS JOB CAN BE USED TO START UP A CICS REGION
//*****
//*
// CICS730 EXEC CICSTASK,
// START=COLD,
// SYSIDNT='HTH1',          SYSID OF CICS REGION
// CLONE='HT##'             CLONE CICS REGION TYPE
//*
```

図 29. CICS TOR の HTH1 を始動するジョブ

この MVS **START** コマンドの例について、以下で説明します。

- CICSTASK は、CICS 提供のサンプル始動プロシーチャーを調整した、カタログ式 CICS 始動プロシーチャーの名前です。
  - SYSIDNT は、各 CICS 領域に固有な CICS システム・データ・セットの識別に使用される修飾子です。
  - CLONE は、SYSIN データ・セット CICSTS56.CICS.SYSIN 内のメンバーの修飾子です。このデータ・セットには、CICS 領域のタイプごとに固有なシステム初期設定パラメーターがあります。
2. MVS 内部読み取りプログラムによりジョブを実行依頼します。  
そうすると、CICS がバッチ・ジョブとして開始します。

## 開始タスクとしての CICS の始動

### 手順

1. プロシーチャーを作成して MVS プロシーチャー・ライブラリー内にインストールします。

始動ジョブ・ストリームが、コーディング手順の規則に従ってコード化されていることを確認します。CICS 提供のサンプル始動プロシージャ DFHSTART を使用したり、必要に合わせて調整したりできます。CICS 提供の始動プロシージャの詳細については、『インストール』の『CICS 始動プロシージャ DFHSTART』を参照してください。

2. CICS を開始するために必要なのは、プロシージャ内で **procname.identifier,keyword(s)=option** をコード化することだけです。
3. 以下のいずれかを行う必要があります。
  - MVS 開始タスク・プロシージャには、IEFSSNaa のサブシステム名とは異なる名前を付けます。デフォルトは「CICS」です。
  - **SUB** パラメーターに JES2 か JES3 のうち該当する値を使用して、**START** コマンドを発行します。
4. オプション: RACF を使用して CICS を実行している場合、RACF テーブル ICHRIN03 により、カタログ式プロシージャ名を適切な許可 RACF ユーザーに関連付けます。  
この関連付けについて詳しくは、[RACF での CICS プロシージャの実行を許可する](#)を参照してください。
5. CICS を開始するには、MVS **START** コマンドを使用します。  
例えば、MVS コンソールから CICS を開始するには、以下のことを行います。

```
S|START procname[.identifier][,SUB=subsystemname][,keyword=option  
[,keyword=option] . . .]
```

ここで、

**procname**

開始される CICS ジョブを定義するカタログ式プロシージャの名前。

**ID**

CICS タスクを識別するために選択する名前。

**subsystemname**

処理するジョブを選択するサブシステムの 名前。このパラメーターを省略した場合は、1 次ジョブ入力サブシステムが使用されます。

**option**

プロシージャ内の対応するパラメーターをオーバーライドするための適切なキーワード。このパラメーターを使用して、カタログ式プロシージャ内で定義されているシンボリック・パラメーターをオーバーライドすることができます。

## 例

例えば、以下の開始コマンドを使用して、[157 ページの図 30](#) にリストされている CICS タスクを開始できます。

```
START CICS730  
  
//CICS730 PROC  
//*  
//DUMMY EXEC PGM=IEFBR14  
//*  
// START CICS730.CICS730,SYSDNT='CICS730',CLONE='HT#'#'  
//* START=COLD  
// START CICS730.CICS730,SYSDNT='CICS730',CLONE='HA#'#'  
//* START=COLD  
// START CICS730.CICS730,SYSDNT='CICS730',CLONE='HA#'#'  
//* START=COLD  
// START CICS730.CICS730,SYSDNT='CICS730',CLONE='HR#'#'  
//* START=COLD  
//*  
//* END OF CICS START PROCEDURE
```

図 30. CICS TOR、2 つの AOR、1 つの ROR を始動するプロシージャ

この MVS **START** コマンドの例について、以下で説明します。

- CICSTASK は、CICS 提供のサンプル始動プロシーチャーを調整した、カタログ式 CICS 始動プロシーチャーの名前です。
- 以下の CICS 領域が始動されます。
  - 端末専有領域 CICSHTH1
  - アプリケーション専有領域 CICS SHAH1
  - アプリケーション専有領域 CICS SHAH2
  - リソース専有領域 CICS HRH2
- SYSIDNT は、各 CICS 領域に固有な CICS システム・データ・セットの識別に使用される修飾子です。
- CLONE は、SYSIN データ・セット CICSTS56.CICS.SYSIN 内のメンバーの修飾子です。このデータ・セットには、CICS 領域のタイプごとに固有なシステム初期設定パラメーターがあります。

START コマンドの完全な構文と、使用できるすべてのキーワードやオプションについては、[z/OS MVS システム・コマンド](#)を参照してください。

## 始動時のシステム初期設定パラメーターのオーバーライド

CICS の初期設定の開始後に、SIT、PARM パラメーター、CICS 始動プロシーチャーの SYSIN データ・セットで指定したシステム初期設定パラメーターをオーバーライドすることもできます。そのためには、システム・コンソールでシステム初期設定パラメーターの新しい値を指定します。

### 始める前に

システム・コンソールでシステム初期設定パラメーターを指定できるのは、PARM パラメーターか SYSIN データ・セット内のいずれかに CONSOLE キーワードが指定されている場合に限られます。EXEC PGM=DFHSHIP ステートメントの PARM ステートメント内か、CICS 始動プロシーチャーの SYSIN データ・セット内に CONSOLE (または CN) キーワードを指定すると、システム・コンソールでシステム初期設定パラメーターを入力するよう CICS がプロンプトを出します。

### このタスクについて

一般的に CICS は、SIT をロードし、PARM パラメーターと SYSIN データ・セット内でコード化された初期設定パラメーターを処理するまで、コンソールからの読み取りを開始しません。

コンソールからパラメーターを読み取る準備ができると、CICS は以下のメッセージを表示します (nn は応答 ID)。

```
nn DFHPA1104 applid - SPECIFY ALTERNATIVE SIT PARAMETERS, IF ANY,
                        AND THEN TYPE '.END'.
```

### 手順

1. メッセージ DFHPA1921 で求められたら、SIT システム初期設定パラメーターを必ず最初のパラメーターとして指定します。  
この時点で、CICS は指定された SIT のロードを試行します。  
CICS が SIT をロードした後で SIT システム初期設定パラメーターを指定しようとする、エラーとして拒否されます。
2. コンソールの単一行には初期設定パラメーターをいくつでも入力できますが、パラメーターを区切るのにはコンマを使用する必要があります。  
CICS は、メッセージ DFHPA1105 を表示して、システム初期設定パラメーターを入力するよう引き続きプロンプトを出します。
3. 該当する初期設定パラメーターを変更し終わったら、END 制御キーワードを入力してコンソール入力を終了します。

## コンソールでのパラメーターに対する訂正の入力

### このタスクについて

PARMERR=INTERACT をコーディングしているときに、キーワードまたはキーワードに割り当てた値にパラメーター・エラーがあることを CICS が検出した場合、CICS はメッセージ DFHPA1912 または DFHPA1915 によって、エラーの修正を求めるプロンプトを出します。

```
DFHPA1912 'applid' SIT OVERRIDE 'keyword' IS NOT RECOGNIZED.
          SPECIFY CORRECT SIT OVERRIDE.
DFHPA1915 'applid' INVALID DATA HAS BEEN DETECTED FOR SIT OVERRIDE
          'keyword'. RESPECIFY THE OVERRIDE.
```

CICS は、SIT のロード後、PARM パラメーターや SYSIN データ・セット内でエラーが検出されるたびに、各エラーに対する訂正を入力するようプロンプトを出します。つまり、PARM パラメーターや SYSIN データ・セット内でエラー状態のパラメーターの後に APPLID パラメーターがある場合、CICS がメッセージ DFHPA1912 および DFHPA1915 で表示するのは、SIT にコーディングされた APPLID であるということです。

## CICS の始動に関するシステム・コンソール・メッセージ

以下の例は、代表的な始動プロシーチャーを使用して CICS 領域 CICSHT56 を始動する際に表示されるメッセージ・シーケンスを示しています。

CICS 提供のデフォルトのシステム 初期設定テーブル DFHSIT が使用され (SIT システム 初期設定パラメーターは指定されていない)、CICSHT56 領域に固有のシステム 初期設定パラメーターが永続的 SYSIN データ・セットから提供されます。SYSIN データはコンソールに表示されるので、例示したメッセージのこれらのオプションを参照できます。

例の後の注記で、対象になっている具体的なメッセージを識別しています。

```
J E S 2 J O B L O G -- S Y S T E M M V 2 6 -- N O D E W I N M V S 2 C
09.12.26 STC63069 $HASP373 CICSGBV1 STARTED
09.12.26 STC63069 IEF403I CICSGBV1 - STARTED
09.12.26 STC63069 -
09.12.26 STC63069 --JOBNAME STEPNAME PROCSTEP RC EXCP --TIMINGS (MINS.)--
09.12.26 STC63069 STEPNO CPU SRB CLOCK SERV PG ----PAGING COUNTS---
09.12.26 STC63069 -CICSGBV1 GENINPT 00 27 .00 .00 .00 202 0 0 0 0
0 1
09.12.26 STC63069 DFHPA1101 IYK2ZDV1 DFHSITVE IS BEING LOADED.
09.12.26 STC63069 DFHPA1108 IYK2ZDV1 DFHSITVE HAS BEEN LOADED. (GENERATED AT: MM/DD= 06/14 HH:MM= 13:42).
09.12.26 STC63069 DFHPA1100 IYK2ZDV1 OVERRIDE PARAMETERS FROM JCL EXEC STATEMENT: SI
09.12.26 STC63069 DFHPA1102 IYK2ZDV1 OVERRIDE PARAMETERS FROM SYSIN:
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 AICONS=AUTO, Autoinstall for consoles
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 CICSSVC=217,SRBSVC=215, CTS 5.4.0 The default CICS SVC number 1
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 CSDDSN=<USER>.CICSVEN.DFHCSO, CICS default userid 2
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 CSDDISP=SHR,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 DEBUGTOOL=NO,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 DSALIM=8M, 3
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 EDSALIM=800M, 3
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 JVMPROFILEDIR=/u/<user>/<version>/JVMProfiles, 4
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 MCT=D$,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 MN=ON,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 MNPER=ON,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 MNRES=ON,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 MXT=200,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 RLS=YES, Set maximum tasks to 200
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 SIT=VE, RLS support required
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 SLD=NO,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 SPCTR=OFF,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 STATINT=010000,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 USSHOME=/itbld/cics.ts.dev/Integrat/dist
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 DB2CONN=YES,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 APPLID=IYK2ZDV1,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 AUXTR=ON,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 AUXTRSW=NEXT,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 CHKSTSK=NONE,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 CHKSTRM=NONE,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 CPSMCONN=LMA5
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 DBCTLCON=YES,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 DEBUGTOOL=NO,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 DSALIM=8M,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 DTRPGM=EYU9XLOP, Use CPSM for routing
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 DSRTPGM=EYU9XLOP, Use CPSM for routing
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 FCT=NO,
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 GMTTEXT='**** <user_name> - <version_alias>(CICS TS <version>) System GMB1
****', 5
09.12.26 STC63069 DFHPA1927 IYK2ZDV1 GRPLIST=(DFHLIST,CICSGMB1,MYJAVA) JVM servers and Liberty Initialize with group
lists for TOR 6
```

09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	ICV=0100,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	ICVTS=0,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	INITPARM=(DFHDBCON='4D'),	DBCTL only
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	INITPARM=(DFHLETRU='USEQSAM'),	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	INITPARM=(DFHFCLJ1='01'),	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	LPA=NO,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	MN=ON,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	MNPER=ON,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	MNRES=ON,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	MQCONN=YES,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	MXT=200,	Set maximum tasks to 200
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	NCPLDFT=NCPPOOL4,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	NISTSP800131A=NOCHECK,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	PDIR=2\$,	All DFHSAM* requests go
to GMB2					
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	PLTPI=1B,	No tools,IP CICS sockets,PLT
for start-up programs					
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	PLTSD=4\$,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	PLTSD=4\$,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	RENTPGM=NOPROTECT,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	RRMS=YES,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	RLS=YES,	RLS support required
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	SLD=YES,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	STATRCD=ON,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	STATINT=010000,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	STNTR=(1,2),	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	SYDUMAX=999,	Limit the number of
system dumps per dump code					
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	SUBTSKS=0,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	STGPROT=YES,	Storage protection on
7					
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	SYSIDNT=GMB1,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	SYSSTR=ON,	Master trace flag
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	TRANISO=YES,	Transaction isolation on
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	TRTABSZ=1048576,	Internal trace table
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	TBEXITS=(GRCINIT,GRCINPT,GFCBFAIL,GFCCLDEL,GFCBOVER,GFCBOUT),	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	TCT=5\$,	Make sure TCP=YES is
specified					
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	GMTRAN=CSGM,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	SEC=NO,	Run without security
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	START=INITIAL,	
09.12.26	STC63069	DFHAPA1927	IYK2ZFFV1	.END	
09.12.26	STC63069	DFHAPA1103	IYK2ZFFV1	END OF FILE ON SYSIN.	
09.12.26	STC63069	DFHAPA1950I	IYK2ZFFV1	READING FEATURE TOGGLE FILE: featuretoggle.properties	
09.12.26	STC63069	DFHAPA1956I	IYK2ZFFV1	blah.blah.blah.joe.blogs=true	
09.12.26	STC63069	DFHAPA1950I	IYK2ZFFV1	READING FEATURE TOGGLE FILE: IYK2ZFFV1/featuretoggle.properties	
09.12.26	STC63069	DFHAPA1958I	IYK2ZFFV1	FILE DOES NOT CONTAIN FEATURE TOGGLES.	
09.12.29	STC63069	+DFHTR0103		TRACE TABLE SIZE IS 1048576K	
09.12.29	STC63069	+DFHSM0601I	IYK2ZFFV1	Limit of above the bar storage available is 10G from JCL.	
09.12.29	STC63069	+DFHSM0122I	IYK2ZFFV1	Limit of DSA storage below 16MB is 8,192K.	
09.12.29	STC63069	+DFHSM0123I	IYK2ZFFV1	Limit of DSA storage above 16MB is 800M.	
09.12.29	STC63069	+DFHSM0115I	IYK2ZFFV1	Storage protection is active.	
09.12.29	STC63069	+DFHSM0125I	IYK2ZFFV1	Transaction isolation is active.	
09.12.29	STC63069	+DFHSM0120I	IYK2ZFFV1	Reentrant programs will not be loaded into read-only storage.	
09.12.29	STC63069	+DFHDM0101I	IYK2ZFFV1	CICS is initializing.	
09.12.30	STC63069	+DFHFWB0109I	IYK2ZFFV1	Web domain initialization has started.	
09.12.30	STC63069	+DFHSO0100I	IYK2ZFFV1	Sockets domain initialization has started.	
09.12.30	STC63069	+DFHRX0100I	IYK2ZFFV1	RX domain initialization has started.	
09.12.30	STC63069	+DFHRX0104I	IYK2ZFFV1	The Resource Recovery Services (RRS) exit manager ATR.EXITMGR.IBM is now available.	
09.12.30	STC63069	+DFHRX0101I	IYK2ZFFV1	RX domain initialization has ended.	
09.12.30	STC63069	+DFHMP0100I	IYK2ZFFV1	Managed platform domain initialization started.	
09.12.30	STC63069	+DFHLG0101I	IYK2ZFFV1	Log manager domain initialization has started.	
09.12.30	STC63069	+DFHEP0101I	IYK2ZFFV1	Event Processing domain initialization has started.	
09.12.30	STC63069	+DFHDH0100I	IYK2ZFFV1	Document domain initialization has started.	
09.12.30	STC63069	+DFHAS0100I	IYK2ZFFV1	Asynchronous services domain initialization started.	
09.12.30	STC63069	+DFHW20100I	IYK2ZFFV1	Web2.0 domain initialization has started.	
09.12.30	STC63069	+DFHXS1100I	IYK2ZFFV1	Security initialization has started.	
09.12.30	STC63069	+DFHAS0101I	IYK2ZFFV1	Asynchronous services domain initialization has ended.	
09.12.30	STC63069	+DFHSI1500	IYK2ZFFV1	CICS startup is in progress for CICS Transaction Server	
09.12.30	STC63069	+DFHXS1102I	IYK2ZFFV1	Security is inactive.	
09.12.30	STC63069	+DFHSI1501I	IYK2ZFFV1	Loading CICS nucleus.	
09.12.30	STC63069	+DFHDM0304I	IYK2ZFFV1	Transaction Dump Data set DFHDMPA opened.	
09.12.30	STC63069	+DFHTR0113	IYK2ZFFV1	Auxiliary trace is being started on data set DFHAUT.	
09.12.30	STC63069	+DFHCQ0100I	IYK2ZFFV1	Console queue initialization has started.	
09.12.30	STC63069	+DFHCQ0101I	IYK2ZFFV1	Console queue initialization has ended.	
09.12.30	STC63069	+DFHCQ0103I	IYK2ZFFV1	MVS console queue is open.	
09.12.30	STC63069	+DFHCQ0200I	IYK2ZFFV1	CEKL transaction enabled.	
09.12.31	STC63069	+DFHMN0108I	IYK2ZFFV1	Using Monitoring Control Table suffix 'D\$'.	
09.12.31	STC63069	+DFHMN0109I	IYK2ZFFV1	CICS Monitoring is active.	
09.12.31	STC63069	+DFHMN0115I	IYK2ZFFV1	CICS Server z/OS WLM Health percentage is now 0.	
09.12.31	STC63069	+DFHFWB0110I	IYK2ZFFV1	Web domain initialization has ended.	
09.12.31	STC63069	+DFHSO0101I	IYK2ZFFV1	Sockets domain initialization has ended.	
09.12.31	STC63069	+DFHSI1502I	IYK2ZFFV1	CICS startup is Initial.	
09.12.31	STC63069	+DFHEM0100I	IYK2ZFFV1	Event Manager initialization has started.	
09.12.31	STC63069	+DFHEM0101I	IYK2ZFFV1	Event Manager initialization has ended.	
09.12.31	STC63069	+DFHEP0102I	IYK2ZFFV1	Event Processing domain initialization has ended.	
09.12.31	STC63069	+DFHMP0101I	IYK2ZFFV1	Managed platform domain initialization has ended.	
09.12.31	STC63069	+DFHSJ0101I	IYK2ZFFV1	946	
946				The JVM (SJ) domain for Java has started initializing. Java is a	
946				trademark of Oracle and/or its affiliates.	
09.12.31	STC63069	+DFHTS0100I	IYK2ZFFV1	Temporary Storage initialization has started.	
09.12.31	STC63069	+DFHLG0102I	IYK2ZFFV1	Log manager domain initialization has ended.	

```

09.12.31 STC63069 +DFHSI1503I IYK2ZVF1 Terminal data sets are being opened.
09.12.31 STC63069 +DFHTS0101I IYK2ZVF1 Temporary Storage initialization has ended.
09.12.31 STC63069 +DFHKE0406I IYK2ZVF1 967
967 CICS is about to wait for predecessors defined in the MVS automatic
967 restart management policy for this region.
09.12.31 STC63069 +DFHKE0412I IYK2ZVF1 CICS WAITPRED call to automatic restart manager has completed.
09.12.31 STC63069 +DFHZC4960 I IYK2ZVF1 969
969 10/16/2017 09:12:31 IYK2ZVF1 CICS has not been able to notify z/OS
969 Communications Server that it will handle detection of BMS 3270
969 intrusion detection.
09.12.31 STC63069 +DFHCP0101I IYK2ZVF1 CPI initialization has started.
09.12.31 STC63069 +DFHPR0104I IYK2ZVF1 Partner resource manager initialization has started.
09.12.31 STC63069 +DFHAI0101I IYK2ZVF1 AITM initialization has started.
09.12.31 STC63069 +DFHTD0100I IYK2ZVF1 Transient Data initialization has started.
09.12.31 STC63069 +DFHW20101I IYK2ZVF1 Web2.0 domain initialization has ended.
09.12.31 STC63069 +DFHFC0100I IYK2ZVF1 File Control initialization has started.
09.12.31 STC63069 +DFHTD0101I IYK2ZVF1 Transient Data initialization has ended.
09.12.31 STC63069 +DFHFC0562 IYK2ZVF1 The RLS control ACB has been successfully registered by CICS.
09.12.31 STC63069 +DFHFC0570 IYK2ZVF1 File control RLS access has been enabled.
09.12.32 STC63069 +DFHFC0101I IYK2ZVF1 File Control initialization has ended.
09.12.32 STC63069 +DFHCP0102I IYK2ZVF1 CPI initialization has ended.
09.12.32 STC63069 +DFHPR0105I IYK2ZVF1 Partner resource manager initialization has ended.
09.12.32 STC63069 +DFHAI0102I IYK2ZVF1 AITM initialization has ended.
09.12.32 STC63069 +DFHSI1511I IYK2ZVF1 Installing group list DFHLIST.
09.12.34 STC63069 +DFHSI1511I IYK2ZVF1 Installing group list CICSGBM1.
09.12.35 STC63069 +DFHDB2110I 10/16/2017 09:12:35 IYK2ZVF1 The total number of threads exceeds TCBLIMIT.
09.12.39 STC63069 +DFHSI1511I IYK2ZVF1 Installing group list MYJAVA.
09.12.39 STC63069 +DFHSJ0910 10/16/2017 09:12:39 IYK2ZVF1 CICSUSER JVMSEVER DFH$JVM has been created.
09.12.39 STC63069 +DFHSJ0910 10/16/2017 09:12:39 IYK2ZVF1 CICSUSER JVMSEVER GMBOSGI has been created.
09.12.40 STC63069 +DFHSJ0910 10/16/2017 09:12:40 IYK2ZVF1 CICSUSER JVMSEVER DFHWLP has been created.
09.12.40 STC63069 +DFHLOG0103I IYK2ZVF1 System log (DFHLOG) initialization has started.
09.12.41 STC63069 +DFHLOG0104I IYK2ZVF1 228
228 System log (DFHLOG) initialization has ended. Log stream
228 <USER>.IYK2ZVF1.DFHLOG is connected to structure LOG_GENERAL_001.
09.12.41 STC63069 +DFHLOG0103I IYK2ZVF1 System log (DFHSHUNT) initialization has started.
09.12.41 STC63069 +DFHLOG0104I IYK2ZVF1 230
230 System log (DFHSHUNT) initialization has ended. Log stream
230 <USER>.IYK2ZVF1.DFHSHUNT is connected to structure LOG_GENERAL_001.
09.12.41 STC63069 +DFHRX0106I IYK2ZVF1 Restart processing with Resource Recovery Services (RRS) is beginning.
09.12.41 STC63069 +DFHRX0107I IYK2ZVF1 Restart processing with Resource Recovery Services (RRS) has ended.
09.12.41 STC63069 +DFHSI1519I IYK2ZVF1 The interregion communication session was successfully started in XCF group
DFHIR000
09.12.41 STC63069 +DFHAP1203I IYK2ZVF1 Language Environment is being initialized.
09.12.42 STC63069 +DFHAP1211I IYK2ZVF1 Language Environment initialization completed.
09.12.42 STC63069 +DFHWP1007 IYK2ZVF1 Initializing CICS Web environment.
09.12.42 STC63069 +DFHWP1008 IYK2ZVF1 CICS Web environment initialization is complete.
09.12.42 STC63069 +DFHEC1006I IYK2ZVF1 Event processing status is STARTED.
09.12.42 STC63069 +DFHSI8440I IYK2ZVF1 Initiating connection to CICSplex SM.
09.12.42 STC63069 +EYUNX0001I IYK2ZVF1 LMAS initialization program starting
09.12.42 STC63069 +EYUXL0003I IYK2ZVF1 CPSM Version 550 LMAS startup in progress
09.12.42 STC63069 +EYUXL0119I IYK2ZVF1 CPSM Kernel loaded from EYU9XL01
09.12.42 STC63069 +EYUXL0022I IYK2ZVF1 LMAS Phase I initialization complete
09.12.42 STC63069 +EYUXL0211I IYK2ZVF1 CPSM Start Up Parameters.
09.12.42 STC63069 +EYUXL0212I IYK2ZVF1 NAME(IYK2ZVF1).
09.12.42 STC63069 +EYUXL0212I IYK2ZVF1 CICSplex(GBPLEX1).
09.12.42 STC63069 +EYUXL0212I IYK2ZVF1 CMASYSID(GMB3).
09.12.42 STC63069 +EYUXL0212I IYK2ZVF1 SEC(NO).
09.12.43 STC63069 +DFHSI8440I IYK2ZVF1 Initiating connection to DBCTL.
09.12.43 STC63069 +DFS4720I DRA open thread option is active
09.12.43 STC63069 +DFS4721I DRA monitoring of thread CPU usage is active
09.12.43 STC63069 +DFHSI8441I IYK2ZVF1 Connection to DBCTL IM4D successfully completed.
09.12.43 STC63069 +DFHSI8440I IYK2ZVF1 Initiating connection to MQ.
09.12.43 STC63069 +DFHMQ0337 I 10/16/2017 09:12:43 IYK2ZVF1 CONNECT received from DFHMQCOD.
09.12.43 STC63069 +DFHMQ0351 I 10/16/2017 09:12:43 IYK2ZVF1 Unable to LOAD API exit CSQCAPX. Program is disabled.
09.12.43 STC63069 +DFHMQ0307 I 10/16/2017 09:12:43 IYK2ZVF1 Successful connection to queue manager MQD5 release 0800
09.12.43 STC63069 +DFHSI8441I IYK2ZVF1 Connection to MQ MQD5 successfully completed.
09.12.43 STC63069 +DFHSI8440I IYK2ZVF1 Initiating connection to DB2.
09.12.43 STC63069 +DFHDB2211I IYK2ZVF1 281
281 10/16/2017 09:12:43 IYK2ZVF1 Maxopentchs value of 72 conflicts with
281 the tcblimit setting of 400 in the DB2CONN definition.
09.12.43 STC63069 +DFHDB2023I 10/16/2017 09:12:43 IYK2ZVF1 The CICS-DB2 attachment has connected to DB2 subsystem DK2C
group DKP2
09.12.43 STC63069 +DFHSI8441I IYK2ZVF1 Connection to DB2 DK2C successfully completed.
09.12.43 STC63069 +DFHSI8430I IYK2ZVF1 About to link to PLT programs during the third stage of initialization.
09.12.43 STC63069 +DFHSI8434I IYK2ZVF1 Control returned from PLT programs during the third stage of initialization.
09.12.43 STC63069 +DFHSI1517 IYK2ZVF1 Control is being given to CICS.
09.12.43 STC63069 +DFHSJ0102I IYK2ZVF1 SJ domain initialization has ended.
09.12.43 STC63069 +DFHDH0101I IYK2ZVF1 Document domain initialization has ended.
09.12.43 STC63069 +DFHXS1101I IYK2ZVF1 Security initialization has ended.
09.12.44 STC63069 +DFHMQ0323 I 10/16/2017 09:12:44 IYK2ZVF1 STOP received from TERMID=SAMA TRANID=CKSD USERID=CICSUSER.
09.12.44 STC63069 +DFHMQ0334 I 10/16/2017 09:12:44 IYK2ZVF1 Successful disconnection from queue manager MQD5 release
0800
09.12.44 STC63069 +DFHMQ0323 I 10/16/2017 09:12:44 IYK2ZVF1 CONNECT received from TERMID=SAMA TRANID=CKCN
USERID=CICSUSER.
09.12.44 STC63069 +DFHMQ0351 I 10/16/2017 09:12:44 IYK2ZVF1 Unable to LOAD API exit CSQCAPX. Program is disabled.
09.12.44 STC63069 +DFHMQ0307 I 10/16/2017 09:12:44 IYK2ZVF1 Successful connection to queue manager MQD5 release 0800
09.12.44 STC63069 +DFHMQ0323 I 10/16/2017 09:12:44 IYK2ZVF1 CONNECT received from TERMID=SAMA TRANID=CKCN
USERID=CICSUSER.
09.12.44 STC63069 +DFHMQ0386 I IYK2ZVF1 306
306 10/16/2017 09:12:44 IYK2ZVF1 STARTCKTI initiated from TERMID=SAMA
306 TRANID=CKSQ USERID=CICSUSER and is accepted.
09.12.44 STC63069 +DFHMQ0386 I IYK2ZVF1 311
311 10/16/2017 09:12:44 IYK2ZVF1 STARTCKTI initiated from TERMID=SAMA

```



```

311          TRANID=CKSQ USERID=CICSUSER and is accepted.
09.12.45 STC63069 +EYUXL0030I IYK2ZVF1 ESSS connection in progress to CICSplex(GBPLEX1) for SYSID(GMB3).
09.12.45 STC63069 +EYUXL0004I IYK2ZVF1 ESSS connection complete.
09.12.45 STC63069 +EYUCL0006I IYK2ZVF1 ESSS link to IYK2ZVF3 established.
09.12.45 STC63069 +EYUXL0007I IYK2ZVF1 LMAS Phase II initialization complete.
09.12.45 STC63069 +EYUNL0099I IYK2ZVF1 LMAS LRT initialization complete.
09.12.45 STC63069 +DFHJS0910 10/16/2017 09:12:45 IYK2ZVF1 CICSUSER JVMSEVER MYPLATJM has been created.
09.12.45 STC63069 +DFHFC0208I IYK2ZVF1 326
326          LSR pool 20 is being built dynamically by CICS because all of the
326          necessary parameters have not been supplied. Either there is no
326          LSRPOOL definition or it is incomplete. The following are not
326          defined: 'CI SIZE' 'STRINGS' 'MAXKEYLENGTH'. A delay is possible.
09.12.48 STC63069 +EYUBN0099I IYK2ZVF1 Resource creation processing complete.
09.12.49 STC63069 +CWWKE0001I: The server <server_name> has been launched.
09.12.51 STC63069 +EYUBN0099I IYK2ZVF1 Resource creation processing complete.
09.12.51 STC63069 +EYUBN0099I IYK2ZVF1 Resource creation processing complete.
09.12.51 STC63069 AXMSC0031I Connection to server DFHCF.DTP00L1 has been opened.
09.12.51 STC63069 +EYUBN0020I IYK2ZVF1 CICS application state recovery is in progress.
09.12.51 STC63069 +EYUWM0505I IYK2ZVF1 Target region (IYK2ZVF1), CICSplex(GBPLEX1) is 357
357          running in Sysplex Optimized WLM state. ?
09.12.51 STC63069 +EYUNL0160I IYK2ZVF1 Workload registration complete.
09.12.51 STC63069 +EYUWM0503I IYK2ZVF1 Routing region (IYK2ZVF1), CICSplex(GBPLEX1) is 359
359          running in Sysplex Optimized WLM state for workload(GMBSPEC).
09.12.51 STC63069 +EYUWI0020I IYK2ZVF1 WLM Routing initiated for workload(GMBSPEC) in 360
360          Routing Region(IYK2ZVF1), CICSplex(GBPLEX1). ?
09.12.52 STC63069 +EYUBN0021I IYK2ZVF1 CICS application state recovery is complete.
09.12.55 STC63069 +CWWKG0016I: Starting server configuration update.
09.12.55 STC63069 +CWWKG0017I: The server configuration was successfully updated in 365
365          0.375 seconds.
09.12.56 STC63069 +CWWKZ0001I: Application ExampleAppClientV855 started in 0.139 seconds.
09.12.56 STC63069 +CWWKZ0001I: Application ExampleAppWrapperClientV855 started in 0.184 367
367          seconds.
09.12.56 STC63069 +CWWKZ0001I: Application com.ibm.cics.server.examples.wlp.hello.war 368
368          started in 0.159 seconds.
09.12.56 STC63069 +CWWKZ0001I: Application com.ibm.cics.fv.wlp.web.db2.datasources 369
369          started in 0.241 seconds.
09.12.56 STC63069 +CWWKZ0001I: Application George started in 0.235 seconds.
09.12.56 STC63069 +CWWKZ0001I: Application ExampleAppDispatchOrderV855 started in 0.511 371
371          seconds.
09.12.56 STC63069 +CWWKZ0001I: Application JAXWSWebSample started in 0.367 seconds.
09.12.56 STC63069 +CWWKZ0001I: Application com.ibm.cics.server.examples.wlp.link started 373
373          in 0.547 seconds.
09.12.56 STC63069 +CWWKZ0001I: Application com.ibm.cics.wlp.link.mdb started in 0.752 374
374          seconds.
09.12.56 STC63069 +CWWKZ0001I: Application com.ibm.cics.server.examples.wlp.tsq.app 375
375          started in 0.975 seconds.
09.12.56 STC63069 +CWWKZ0001I: Application com.ibm.cics.server.examples.wlp.jdbc.app 376
376          started in 0.975 seconds.
09.12.57 STC63069 +CWWKF0011I: The server <server_name> is ready to run a smarter planet.
09.13.03 STC63069 +DFHMM0115I IYK2ZVF1 CICS Server z/OS WLM Health percentage is now 25.
09.13.03 STC63069 +DFHMQ0391 I 07/10/2019 09:13:03 IYK2ZVF1 Start requested for mqmonitor APPLID, transaction CKTI.
09.13.03 STC63069 +DFHMQ0391 I 07/10/2019 09:13:03 IYK2ZVF1 Start requested for mqmonitor CSQ4SAMP, transaction CKTI.
09.13.03 STC63069 +DFHMQ0391 I 07/10/2019 09:13:03 IYK2ZVF1 Start requested for mqmonitor DFHMQINI, transaction CKTI.
09.13.03 STC63069 +DFHMQ0391 I 07/10/2019 09:13:03 IYK2ZVF1 Start requested for mqmonitor EPD3BRID, transaction CKBR.
09.13.03 STC63069 +DFHMQ0391 I 07/10/2019 09:13:03 IYK2ZVF1 Start requested for mqmonitor MQGETTER, transaction MGTR.
09.13.03 STC63069 +DFHMQ0391 I 07/10/2019 09:13:03 IYK2ZVF1 Start requested for mqmonitor PIPE3, transaction CKTI.
09.13.03 STC63069 +DFHMQ0391 I 07/10/2019 09:13:03 IYK2ZVF1 Start requested for mqmonitor PIPE4, transaction CKTI.
09.13.03 STC63069 +DFHMQ0700 I 07/10/2019 09:13:03 IYK2ZVF1 CKBR 00119 CICS-MQ bridge initialization in progress.
09.13.03 STC63069 +DFHMQ0702 I 07/10/2019 09:13:03 IYK2ZVF1 CKBR 00119 CICS-MQ bridge monitor initialization complete.
09.13.03 STC63069 +DFHMQ0703 I 07/10/2019 09:13:03 IYK2ZVF1 CKBR 00119 WaitInterval=20,000, Auth=LOCAL
Q=SYSTEM.CICS.BRIDGE.QUEUE.
09.13.03 STC63069 +DFHMQ0783 I 07/10/2019 09:13:03 IYK2ZVF1 CKBR 00119 Msg=BOTH, PassTktA=IYK2ZVF1.
09.13.03 STC63069 +DFHMQ0792 I 07/10/2019 09:13:03 IYK2ZVF1 CKBR 00119 RouteMEM=N
09.13.03 STC63069 +DFHMQ0794 I 07/10/2019 09:13:03 IYK2ZVF1 CKBR 00119 SmfMqGet=100.
09.13.24 STC63069 +DFHMM0115I IYK2ZVF1 CICS Server z/OS WLM Health percentage is now 50.
09.13.44 STC63069 +DFHMM0115I IYK2ZVF1 CICS Server z/OS WLM Health percentage is now 75.
09.14.04 STC63069 +DFHMM0115I IYK2ZVF1 CICS Server z/OS WLM Health percentage is now 100.
09.14.04 STC63069 +DFHMQ0391 I 10/16/2017 09:14:04 IYK2ZVF1 Start requested for mqmonitor CSQ4SAMP, transaction CKTI.
09.16.19 STC63069 +DFHHC0101I IYK2ZVF1 CICS has registered successfully to the z/OS Health Checker.

```

## 注:

1. この SVC 値は単なる例です。SVC テーブル内で定義されている SVC 値を指定してください。
2. CICS 領域を起動しようとする前に、**DFLTUSER** システム初期設定パラメーターを使用してデフォルトの CICS ユーザー ID を定義しなければなりません。実動システムでは、不要などの CICS トランザクションにも、デフォルト・ユーザーはアクセスしてはなりません。デフォルト・ユーザーに付与するリソース・アクセス許可は、誰でも使用でき、したがって制限する必要のまったくないリソースだけに明確に限定しなければなりません。デフォルト・ユーザー ID の属性の定義については、[Defining the default CICS user ID to RACF](#) を参照してください。
3. DFHSM0122 メッセージは、16 MB より上の動的ストレージ域 (DSA) の制限を表示します。この制限は、**DSALIM** システム初期設定パラメーターによって設定します。DFHSM0123 メッセージは、16 MB より上で 2 GB より下の DSA の制限を示します。この制限は、**EDSALIM** システム初期設定パラメーター

によって設定します。これらのストレージ域については、[DSALIM システム 初期設定パラメーター](#)および [EDSALIM システム 初期設定パラメーター](#)を参照してください。

4. システム 初期設定パラメーターとして **MCT=NO** (デフォルト) を指定すると、CICS モニター・ドメインはデフォルトのモニター管理テーブルを作成します。これで、モニターがアクティブになっているときに、デフォルトのモニター管理テーブル・エントリーが常に使用可能になります。次のメッセージは、モニターが非アクティブになっていることを示します。
5. これは、端末が z/OS Communications Server を使用して CICS にログオンする場合は、CSGM (good morning) トランザクションにより画面に表示されるメッセージ・テキストを指定し、CESN トランザクションを使用して CICS にサインオンする場合は、このトランザクションにより画面に表示されるメッセージ・テキストを指定します。
6. CICS の始動時にロードされるリソース・グループ・リスト。これらのグループ・リスト内のリソース定義のグループは、初期始動時かコールド・スタート時のみインストールされます。ウォーム・リスタートの場合でも、リソース定義はシャットダウン時にインストールされ、CICS グローバル・カタログから復元されます。
7. システム 初期設定パラメーター **STGPROT=YES** が指定されているか、デフォルトの YES を使用できるので、ストレージ保護がアクティブになります。

## CICS Transaction Server によるソフトウェアの価格設定のためのリソース使用状況の収集

CICS TS は、ソフトウェアの使用状況の情報の収集をサポートしています。

サブキャパシティー・レポーティング・ツール (SCRT) の使用法については、[Using the Sub-Capacity Reporting Tool](#) を参照してください。

CICS は、各 CICS アドレス・スペースと、名前付きカウンター、一時記憶域、カップリング・ファシリティ・データ・テーブル・サーバーに関するアドレス・スペースの初期設定時に、IFAUSAGE マクロを使用して情報を登録します。領域状況サーバーは、カップリング・ファシリティ・データ・テーブルの変型で、同じく情報を登録します。登録される情報には、CICS のバージョンと製品 ID が含まれます。この情報は、SMF 89 レコードで収集されます。



## 第 5 章 CICS データ共用サーバーのセットアップ

CICS データ共用サーバーを初期化するには、AXM システム・サービスをセットアップし、適切なデータ共用サーバーを定義する必要があります。

CICS データ共用サーバーは、以下をサポートします。

- 一時記憶域データ共用
- カップリング・ファシリティ・データ・テーブル
- 名前付きカウンター

CICSplex SM シスプレックスの最適化ワークロード管理を実装するユーザーは、領域状況サーバーを構成する必要があります。このサーバーは、CICS 領域が汎用システム状況データをブロードキャストする特注タイプの CFDT サーバーです。CICSplex SM は、動的ルーティングを決定するためにそのデータを照会します。このサーバーによって管理されるテーブル構造を、ユーザー・アプリケーションで利用するために変更、調整、および再構成することはできません。

- 165 ページの『AXM システム・サービスの定義と開始』では、AXM システム・サービスを定義して開始する方法について説明します。
- 166 ページの『一時記憶域サーバーのセットアップと実行』では、CICS 一時記憶域データ共用サーバーをセットアップして実行する方法について説明します。
- 180 ページの『カップリング・ファシリティ・データ・テーブル・サーバーのセットアップと実行』では、CICS カップリング・ファシリティ・データ・テーブル・サーバーをセットアップして実行する方法について説明します。
- 231 ページの『カップリング・ファシリティ・サーバーの操作』では、3 種類のすべての CICS カップリング・ファシリティ・サーバーに共通する操作について概説します。
- 215 ページの『名前付きカウンター・サーバーのセットアップと実行』では、CICS 名前付きカウンター・サーバーをセットアップして実行する方法について説明します。

### AXM システム・サービスの定義と開始

許可仮想記憶間 (AXM) サーバー環境には、CICS データ共用サーバー領域のためのランタイム環境が用意されています。CICS データ共用サーバーを実行するには、AXM システム・サービスを定義して開始する必要があります。

#### このタスクについて

MVS イメージのための AXM 仮想記憶間接続を確立するには、AXM という MVS サブシステムを定義する必要があります。そのサブシステムは、静的に作成することも、動的に作成することも可能です。

#### 手順

- MVS サブシステムを静的に定義する場合は、必要なパラメーターを指定した項目を SYS1.PARMLIB の IEFSSNxx メンバーに追加します。

```
SUBSYS SUBNAME(AXM) INITRTN(AXMSI)
```

IEFSSNxx メンバーで AXM を定義すると、MVS で IPL を実行した時に AXM システム・サービスが自動的に有効になります。

AXM サブシステム初期化ルーチン AXMSI によって、該当する定義がマスター・スケジューラー領域からセットアップされます。AXM がそのサブシステム定義を使用するのは、マスター・スケジューラー・アドレス・スペースで AXM の初期化スケジュールを設定する手段として使用する場合に限られます。AXM のための MVS サブシステム・インターフェースが活動化されたり使用されたりすることはありません。

- MVS サブシステムを動的に定義する場合は、以下のコマンドを入力します。

```
SETSSI ADD,SUBNAME=AXM,INITRTN=AXMSI
```

コマンドにエラーがあったり、AXMSI がリンク・リスト・ライブラリーに含まれていなかったりすることなどが原因で、AXM サブシステムの初期化が失敗した場合は、MVS で再試行ができません。そのサブシステムはすでに定義されているからです。その場合は、別のサブシステム名 (AXM1 など) を使用してください。AXM は特定のサブシステム名に依存してはいません。初回に AXM を正常に開始できた場合は、再試行が無視されます。

### 次のタスク

AXM サブシステムを正常に開始できたら、データ共用サーバーを作成できます。

## 一時記憶域サーバーのセットアップと実行

AOR で実行される CICS トランザクションは、指定されたプールをサポートする一時記憶域サーバーを介して一時記憶域 (TS) プールにアクセスします。シスプレックス内の各 MVS イメージにおいて、カップリング・ファシリティで定義されているプールごとにそれぞれ 1 つの一時記憶域サーバーをセットアップする必要があります。

### 手順

1. カップリング・ファシリティで一時記憶域リスト構造として TS プールを定義します。
2. 共用 TS プールのための一時記憶域ジョブを定義して開始し、MVS バッチ領域で実行します。
3. 一時記憶域領域が実行されている間、コマンドを発行して、プール用のキュー・サーバーを制御できます。
4. オプションで、キュー・プールをアップロードしたり、再ロードしたりできます。

## 一時記憶域データ共用サーバーの概要

AOR で実行される CICS トランザクションは、指定されたプールをサポートする一時記憶域データ共用サーバーを介して一時記憶域プールにアクセスします。

シスプレックス内の各 z/OS イメージ内に、その z/OS イメージからアクセスできるカップリング・ファシリティに定義されているプールごとに 1 台の TS サーバーを用意する必要があります。すべての TS プールへのアクセスは、指定されたプール用の TS サーバーへの仮想記憶間呼び出しによって実行されます。

AOR は、複数の TS サーバーに同時にアクセスできます。各 TS サーバーがアクセスを提供する TS キューのプールは 1 つのみであるため、複数のプールを作成した場合は、このマルチサーバー・アクセスが必要になります。

CICS は、TSMODEL リソース定義の POOLNAME 属性を使用して、一時記憶域要求を TS サーバーにマップします。

167 ページの図 31 は、一時記憶域サーバーのアドレス・スペースに 3 つの CICS AOR がリンクされた並列シスプレックスを表しています。

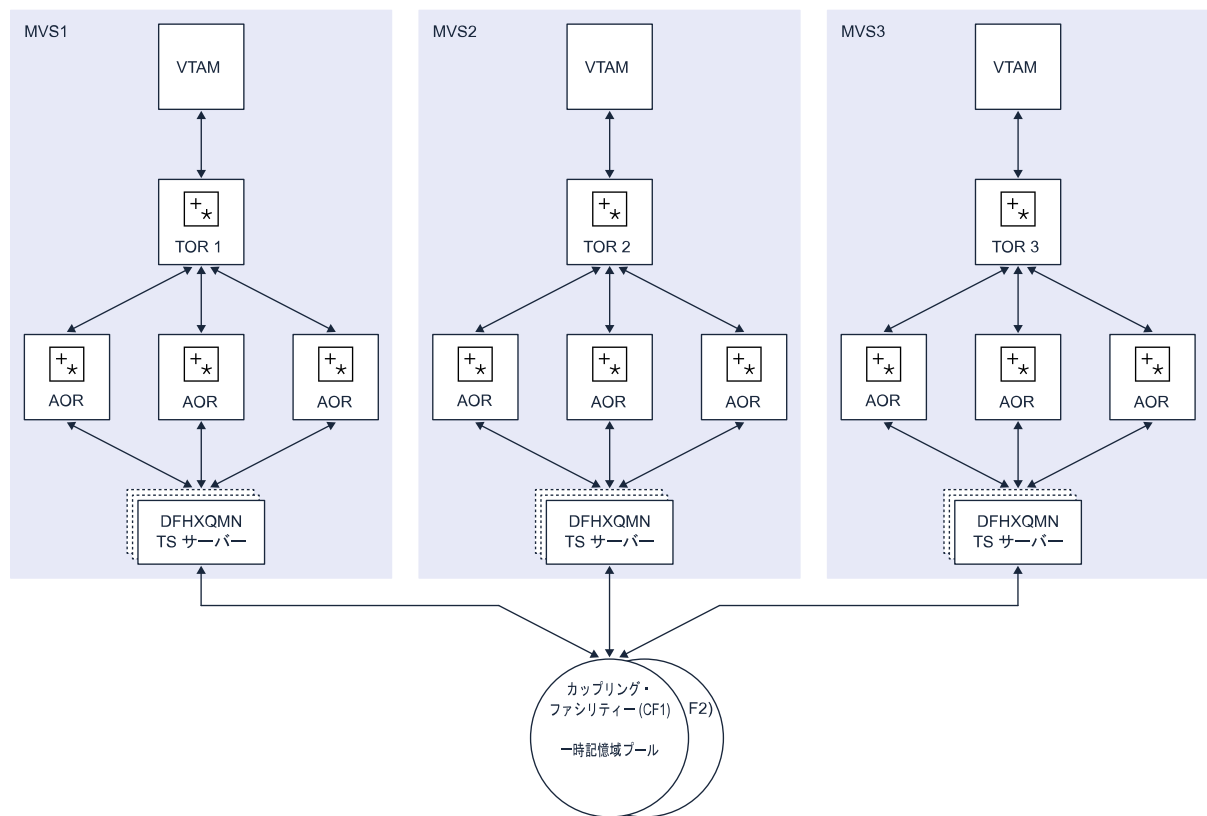


図 31. Parallel Sysplex の TS データ共用に関する概念ビュー



## セキュリティ

サーバーには、一時記憶域プールが定義されているカップリング・ファシリティのリスト構造へのアクセスを許可する必要があります。XES がこれを検査します。また、一時記憶域サーバーとして動作することも許可する必要があります。これは AXM が検査します。必要な許可の定義方法については、[一時記憶域サーバーへのアクセスを許可する](#)を参照してください。

## 一時記憶域データ共用のための一時記憶域プールの定義

カップリング・ファシリティ・リソース・マネージャー (CFRM) のポリシー・ステートメントを使用して、一時ストレージ (TS) プールを一時ストレージ・リスト構造としてカップリング・ファシリティに定義します。

### このタスクについて

TS データ共用を使用するには、TS キューの主ストレージまたは補助ストレージを 1 つ以上の TS プールに置き換えます。この場合、各 TS プールの範囲と機能はキュー所有領域 (QOR) と同様です。

各 TS プールは、MVS システム間拡張サービス (XES) を使用して、キー付きリスト構造としてカップリング・ファシリティに定義されます。このため、カップリング・ファシリティ・リソース・マネージャー (CFRM) のポリシー・ステートメントを使用して、プールを定義する必要があります。必要となるリスト構造のサイズ、およびカップリング・ファシリティ内のその位置を指定するには、CFRM ポリシー定義ユーティリティ `IXCMIAPU` を使用します。このユーティリティの例については、`SYS1.SAMPLIB` ライブラリーのメンバー `IXCCFRMP` を参照してください ([『z/OS MVS Setting Up a Sysplex』の『CFRM ポリシー・データ向け管理データ・ユーティリティ』](#)を参照)。[168 ページの図 32](#)に、定義ステートメントの例が示されています。

```
STRUCTURE NAME(DFHXQLS_PRODTSQ1)
  SIZE(8192)
  INITSIZE(10240)
  PREFLIST(FACIL01,FACIL02)
```

図 32. リスト構造の見積もりサイズの定義例

TS データ共用プールのリスト構造の名前は、TS プール名を接頭部 `DFHXQLS_` に付加して、`DFHXQLS_poolname` と指定して作成します。定義が完了したら、MVS オペレーター・コマンド `SETXCF START` を使用して CFRM ポリシーをアクティブにする必要があります。

リスト構造が割り振られると、CFRM ポリシーに初期サイズと最大サイズを指定できます。すべての構造サイズは、割り振り時にカップリング・ファシリティ・レベル (CFLEVEL) の次のストレージ増分値に切り上げられます。例えば、CFLEVEL 16 の場合、サイズは最も近い 1 MB 単位の値に丸められます。カップリング・ファシリティでスペースが使用可能な場合、リスト構造をその初期サイズから最大サイズまで動的に拡張したり、リスト構造のサイズを縮小して他の目的で使えるようカップリング・ファシリティのスペースを解放したりできます。初期構造の割り振りがいっぱいになると、割り振り済みの構造が指定された最大サイズより小さい場合でも、構造は自動的に拡張されません。割り振りがいっぱいになった場合にリスト構造を拡張するには、次の `SETXCF` コマンドを使用することにより、リスト構造を (最大サイズまで) 拡張できます。

```
SETXCF START,ALTER,STRNAME=DFHXQLS_poolname,SIZE=nnnn
```

リスト構造のための CFRM ポリシー・ステートメントを定義しても、リスト構造は作成されません。TS サーバーは、初期設定中にリスト構造を作成します。[166 ページの『一時記憶域サーバーのセットアップと実行』](#)を参照してください。

そのサーバー接続が失敗すると、カップリング・ファシリティ・データ・テーブルに影響が及びます。[235 ページの『サーバー接続管理』](#)および[名前付きカウンターのリカバリー](#)を参照してください。

リスト構造の定義について詳しくは、以下の [z/OS 情報を参照してください](#)。

- [z/OS MVS シスプレックスのセットアップ](#)
- [z/OS MVS プログラミング: シスプレックス・サービス ガイド](#)
- [z/OS MVS プログラミング: シスプレックス・サービス解説書](#)

## 一時記憶域データ共用に関するストレージの計算

z Systems® カップリング・ファシリティ構造サイズ決定支援ツール (CFSizer) を使用して、カップリング・ファシリティの一時記憶域リスト構造に関するストレージ所要量を計算できます。

カップリング・ファシリティの構造には、キー順データ・セットと同様に、保管データと、そのデータの管理とアクセスに必要な情報の両方が含まれます。カップリング・ファシリティ内のエントリーごとに、データは固定サイズ (普通は 256 バイト) のエレメントのチェーンとして保管されます。したがって、正しい長さの可変長データを個別に保管しなければなりません。そのために、CICS では保管データに長さ接頭部を組み込みます。したがって、スペース計算では接頭部の長さに各エントリーの正確な長さを加算します。内部制御情報の量は、現行のカップリング・ファシリティ・レベル (CFLEVEL) のカップリング・ファシリティ制御コードの機能とパフォーマンスのレベルに応じて異なります。CFLEVEL が高いほどストレージ要件が大きくなる可能性があります。詳細については、[232 ページの『カップリング・ファシリティのストレージ管理』](#)を参照してください。

CFSizer は、これらの要素を加味したうえで現行の CFLEVEL でカップリング・ファシリティと通信して、ストレージ所要量を計算する Web ベース・アプリケーションです。[CFSizer](#) を参照してください。

CFSizer を使用して、一時記憶域リスト構造のストレージ所要量を計算するには、以下の情報を入力します。

### 最大キュー数

CICS で構造リストを割り振る際に予約されるデータ・リストの最大数。この値により、構造内に保管できる大規模キューの最大数が決まります。この値は、**MAXQUEUES** サーバー・パラメーターに対応しています。[173 ページの『リスト構造パラメーター』](#)を参照してください。

データ項目の合計サイズが 32K を超えている場合、大規模キューと呼ばれます。複数の項目がある小規模キュー (32K 以下) の場合、すべてのキュー項目がキュー索引エントリーのデータ部分として保管されます。キューが 32K を超える場合、エントリーごとに 1 つの項目が構造内の別個のリスト内に保管されてキュー索引エントリーにより参照される形式に変換されます。

大規模キューを処理するのに十分な大きさを指定します。ただし、不使用の事前割り振りされたリスト・ヘッダーがカップリング・ファシリティ・ストレージの大部分を占めてしまうことにならないように、大き過ぎる値は指定しないでください。有効な範囲は 1 から 999999 までです。デフォルトは、1000 です。

### 切り上げられた項目の平均サイズ

TS キュー項目ごとに必要な平均ストレージ量。各項目には 2 バイトの長さの接頭部があり、1 つ以上の 256 バイトのエレメントとして保管されます。この値によって、必要な構造サイズを計算するために使用する「エントリー対エレメント」率が決まります。有効な範囲は 1 から 32768 までです。デフォルトは 256 です。

すべてのキュー項目がほぼ同サイズの場合、データ・サイズの平均を取り、2 を加算し、次の 256 の倍数に切り上げて、この値が計算されます。必要なエレメントのストレージの量は、データ項目のサイズに 2 バイトを加算した量になります。各項目に長さの接頭部があるからです。

キュー項目のサイズが異なる場合は、最初に各サイズを切り上げてから、その後に平均を取ります。例えば、項目のうち半分が 100 バイトで残りの半分が 300 バイトの場合、前者のサイズを 256 に、後者のサイズを 512 に切り上げてから、それらの平均を取ります。切り上げて平均した結果の項目のサイズは 384 になります。この値は、項目の平均サイズ 200 を使用して 256 に切り上げるより正確です。

### すべてのキュー内の項目の合計数

すべての TS キュー内のエントリーの合計数。

### ターゲット使用率 (%)

求められた合計数の項目が使用すると想定される、構造のスペースのパーセンテージ。1 から 100 の範囲内で数値を指定します。デフォルトは 75 です。この値は、次のことを保証します。

- ・一時的な拡張用のフリー・スペースが存在する。
- ・初期フリー・スペースでは足りない場合、警告メッセージに応答して構造を拡張する時点が存在する (通常 80% で開始されます)。
- ・エントリーとエレメントの比率を変更するアクティビティが減る。

## 最大拡張率 (%)

構造を拡張できるパーセンテージ。ゼロ以外の値を指定すると、構造の最大サイズは、初期構造サイズにこのパーセンテージ分だけデータの合計量を加算したサイズに増えます。例えば、値 200 を指定すると、指定された合計項目数を初期サイズで十分保管できる場合、最大サイズではこの 3 倍の項目数を十分保管できます。

## TS サーバー領域の定義

共用 TS プールは、仮想記憶間キュー・サーバー領域を介してアクセスされる XES リスト構造で構成されます。共用 TS プールのキュー・サーバー領域を開始して、MVS イメージ内でその共用 TS プールを開始します。

### このタスクについて

共用 TS プールの TS サーバー領域の開始には、バッチ・ジョブまたは開始タスクを使用できます。このジョブまたはタスクは、APF 許可ライブラリー内にあるキュー・サーバー領域プログラム DFHXQMN を起動する必要があります。

### 手順

1. JCL に定義された SYSIN データ・セット内、または EXEC ステートメントの **PARM** パラメーターに DFHXQMN プログラムを指定します。
2. DFHXQMN プログラムの必須およびオプションの始動パラメーターを指定します。
  - a) 印刷ファイルに SYSPRINT DD ステートメントを指定する必要があります。
  - b) サーバー・パラメーターに SYSIN DD ステートメントを指定する必要があります。
  - c) TS プール名を指定する必要があります。
  - d) ライセンスのアクティベーション・データ・セット (SDFHLIC ライブラリー) を STEPLIB DD ステートメントに連結する必要があります。
  - e) TIME=NOLIMIT を指定することをお勧めします。

サーバー・タスクは、ほとんどの通常処理の間、待機状態のままです。これは、サーバー処理がクライアント CICS 領域の TCB で実行されるためです。このパラメーターを省略すると、SYS1.PARMLIB の SMFPRMxx メンバーに指定した JWT 値によっては、サーバー・ジョブが異常終了 S522 (待機時間制限の超過) で失敗することがあります。
  - f) 必要に応じてパラメーターを追加指定します。

例えば、プールで対応可能なキューの最大数や、サーバーが割り振るバッファの数を制御することができます。

### TS サーバーの始動ジョブの例

170 ページの図 33 に、TS サーバーの始動に使用できる JCL の例を示します。

```
//PRODTSQ1 JOB ...
//TSSERVER EXEC PGM=DFHXQMN,REGION=64M,TIME=NOLIMIT Start TS data sharing server
//STEPLIB DD DSN=CICSTS56.CICS.SDFHAUTH,DISP=SHR Authorized library
// DD DSN=CICSTS56.CICS.SDFHLIC,DISP=SHR License activation data set
//SYSPRINT DD SYSOUT=* Messages and statistics
//SYSIN DD *
POOLNAME=PRODTSQ1 Pool name
MAXQUEUES=5000 Allow up to 5000 large queues
BUFFERS=1000 1000 buffers (32K each, 32M total)
/*
```

図 33. TS キュー・サーバーの始動ジョブの例

## キュー・サーバーの REGION パラメーター

キュー・サーバーの REGION パラメーター JCL では、少なくとも、指定したバッファ数およびキュー要求を処理するために使用されるストレージのために十分な仮想記憶域を指定する必要があります。

各バッファは 32K バイト強を占有し、接続された CICS 領域はそれぞれ一度に最大 10 個のキュー要求をアクティブにできます。それぞれのキュー要求で 5K から 10K バイトが使用されるため、念のため REGION サイズには、バッファごとに 32K、接続された CICS 領域ごとに 100K、そしてその他のストレージ域のためのマージンとして 10% を最低限確保する必要があります。

サーバーの初期設定中に、サーバーは REGION サイズに指定されたとおりに、16M 境界より上の使用可能なストレージをすべて確保し、そのうちの 5% をオペレーティング・システム・サービス用に解放します。また、24 ビットのアドレス可能ストレージが必要なルーチン (例えば、順次ファイル読み取り/書き込みルーチンなど) で使用するために、16M 境界より下の空きストレージの 5% も確保します。

サーバー初期設定の後には、サーバー領域ストレージを管理するために AXM ページ割り振りサービスが使用されます。サーバー統計は、割り振られたストレージの量、16M 境界の上下のストレージ域内の使用量を示します。これらは、統計では AXMPGANY および AXMPGLOW と呼ばれます。

サーバー領域のタスクまたは仮想記憶間要求がストレージを使い果たした場合、AXM が、異常終了のシミュレーションを使用して、そのタスクまたは要求を強制終了させ、GETMAIN 障害を示すシステム完了コード 80A を出す可能性があります。その場合、通常、サーバーは他の要求の処理を継続できますが、重要なルーチンでストレージを使い果たしてしまうと、サーバーが強制終了される可能性があります。そのため、十分な大きさの REGION サイズを割り振り、このリスクを完全に排除することをお勧めします。

## TS キュー・サーバーのパラメーター

パラメーターは、KEYWORD=value の形式で指定します。読みやすくするために、キーワードを大/小文字混合で指定することもできます。

PARM フィールドまたは同じ SYSIN 入力行で複数のパラメーターを指定する場合、パラメーターはコンマで区切る必要があります。1 つ以上のスペースに続くテキストは、すべて説明コメントと見なされます。アスタリスクまたはスペースで始まるパラメーター行はすべて、行全体がコメントと見なされます。

いくつかのパラメーター・キーワードを複数の形式 (例えば省略形式) で入力できます。各キーワードの標準の形式は、表示されている最初の単語の一番長い形式です。

使用される主なパラメーターは、始動中にサーバー・プリント・ファイルにリストされます。

以下のパラメーターはすべて (SYSIN ファイルまたは PARM フィールドの) 初期設定パラメーターとして有効で、SET コマンドで変更できるものもあります。

サーバーの DISPLAY コマンドを使用して、任意のパラメーターを表示することができます。すべてのパラメーターの値を表示するには、DISPLAY ALLPARMS を使用します。

以下のキーワードを指定して、情報の結合リストを作成します。

- PARMs、メインのパラメーター値
- STATS、使用可能なすべての統計
- INIT、初期設定が完了すると通常表示される値を持つパラメーターおよび統計を選択する

## 1 次パラメーター

以下のパラメーターは通常、すべてのサーバーで指定されています。

### POOLNAME=pool\_name

サーバー名と、カップリング・ファシリティ・リスト構造 DFHXQLS\_poolname を形成するキュー・プールの名前 (1 から 8 文字) を指定します。このパラメーターは初期化時のみ有効で、常に指定する必要があります。

このキーワードは、**POOL** としてコーディングすることもできます。

### BUFFERS={100|number}

サーバー・アドレス・スペースに割り振るキュー・バッファの数を指定します。

キュー索引バッファには、キュー索引入力と、最大で 32K までのキュー・データ (小さいキューの場合) が入ります。READ または WRITE 要求が完了すると、キュー索引情報はバッファ内に保存されます。これにより、バッファが再使用される前に同じキューが同じ MVS イメージから参照される場合



に、キュー索引を再読み取りする必要がなくなります。要求時にバッファが利用可能ではない場合、空きがでるまで要求は待機することになります。

バッファの数は、この MVS イメージ内のサーバーに接続できる CICS 領域ごとに 10 以上あることが理想です。これだけあれば、バッファ待ちが発生するリスクを低減できます。さらに、バッファを使用することで、最近使用したキュー索引入力がストレージに保存されるため、カップリング・ファシリティーのアクセスの数を減らすことができます。具体的には、キュー項目の読み取り時に、現在のバージョンのキュー索引入力がストレージに入っていると、要求によって必要なカップリング・ファシリティーのアクセス数は 2 つではなく、1 つで済みます。2 つ目または後続の項目が同じキューに書き込まれる際に現在のバージョンのキュー索引入力がストレージに入っている場合、要求によって必要なカップリング・ファシリティーのアクセス数は 3 つではなく 1 つで済みます。

MVS ページングが発生する程多くのバッファを定義することは有用ではありません。補助ストレージからバッファにページインするより、索引入力の再読み取りを行う方が効率的だからです。このパラメーターは、初期化時にのみ有効です。

有効な範囲は 1 から 999999 までです。

このキーワードは、**BUF** としてコーディングすることもできます。

#### **FUNCTION={SERVER|UNLOAD|RELOAD}**

このパラメーターに関する情報は、[179 ページの『キュー・プールのアンロードと再ロード』](#)に記載されています。

#### **STATSOPTIONS={NONE|SMF|PRINT|BOTH}**

間隔統計が作成されるかどうかと、統計が SMF または印刷ファイル、あるいはその両方に送信されるかどうかを決定する統計オプションを指定します。

このキーワードは、**STATSOPT** としてコーディングすることもできます。

#### **ENDOFDAY={00:00|hhmm}**

終業時統計を収集してリセットする時間を指定します。統計オプションで NONE が指定されると、終業時統計は印刷ファイルに書き出されます。有効な範囲は 00:00 から 24:00 までです。

このキーワードは、**EOD** としてコーディングすることもできます。

#### **STATSINTERVAL={3:00|hhmm}**

統計間隔を 1 分から 24 時間までの間で指定します。STATSOPTIONS=NONE に指定されている場合は無視されます。

有効な範囲は 00:01 から 24:00 です (秒単位で指定される場合もある)。

このキーワードは、**STATSINT** としてコーディングすることもできます。

#### **自動リスタート・マネージャー (ARM) のパラメーター**

サーバーの初期化中、サーバーは UNLOAD または RELOAD 機能のいずれかを使用してサーバー・プログラムが呼び出された場合を除き、ARM に無条件で登録します。登録が失敗している場合、サーバーは始動しません。

自動リスタート・マネージャーのデフォルトの処理をオーバーライドする場合は、以下のパラメーターを使用します。

#### **ARMELEMENTNAME=elementname**

自動再始動の目的で ARM に対してサーバーを識別するために ARM にポリシーで使用するために、自動リスタート・マネージャーのエレメント名を 16 文字までで指定します。エレメント名に使用できる文字は A から Z、0 から 9、\$ # @ およびアンダースコア記号 ( ) です。

デフォルトの識別子の形式は DFHXQnn\_poolname で、XQ はサーバー・タイプ、nn はシステムの &SYSCON 値 (1 文字または 2 文字のいずれか)、poolname はサーバーによってサービス提供されるプールの名前を表します。

このパラメーターは、サーバーの初期化時にのみ有効になります。

このキーワード ARMELEMENT または ARMELEMNAME と省略することができます。

## ARMELEMENTTYPE=elementtype

類似の要素を分類する手段として ARM ポリシーで使用するために、自動リスタート・マネージャーの要素・タイプを 8 文字までで指定します。要素・タイプに使用できる文字は A-Z 0-9 \$ # および @ です。

デフォルトの要素・タイプは SYSCICSS です。

このパラメーターは、サーバーの初期化時にのみ有効になります。

このキーワードは、ARMELEMENTTYPE と省略することができます。

## リスト構造パラメーター

リスト構造パラメーターは、TS プールの一時記憶域リスト構造の最初の割り振りにのみ使用される属性を指定します。最初の割り振りは、TS プールのサーバーが始めて始動した時にのみ発生します。

リスト構造パラメーターは、次のとおりです。

### MAXQUEUES={1000|number}

構造が割り振られる際に、予約できるデータ・リストの最大数を指定します。これは、構造に格納できる大きなキューの最大数を決定します。このパラメーターによって、構造の作成時に定義されるリスト・ヘッダーの数も決まります。

大規模キューを処理するのに十分な大きさを指定します。ただし、不使用の事前割り振りされたリスト・ヘッダーがカップリング・ファシリティー・ストレージの大部分を占めてしまうことにならないように、大き過ぎる値は指定しないでください。

構造再割り振りを行わずにこの数を変更することはできません。そのため、構造が最大サイズより小さく割り振られている場合は、この値を構造の初期サイズではなく最大サイズに基づいて決定する必要があります。

このパラメーターは、CFSizer ツールの「キューの最大数」の値に対応します (169 ページの『一時記憶域データ共用に関するストレージの計算』を参照)。

有効な範囲は 1 から 999999 までです。

このキーワードは、MAXQ としてコーディングすることもできます。

### POOLSIZE={0|number{K|M|G}}

リスト構造に割り振られるストレージの最大量を指定します。キロバイト (nK)、メガバイト (nM)、またはギガバイト (nG) で表します。

このパラメーターは、カップリング・ファシリティー・リソース管理 (CFRM) ポリシーでリスト構造に対して指定されている値よりも小さい値を指定してリスト構造が作成された場合に使用します。

デフォルト値 0 は、ここには最大値が適用されず、CFRM ポリシーで指定された最大値が適用されることを指定します。

0 以外の値は MVS によってカップリング・ファシリティー・レベル (CFLEVEL) の次のストレージ増分に切り上げられます。例えば、CFLEVEL 16 の場合、値は最も近い 1 MB 単位の値に丸められます。

有効な範囲は 0 から 16777215M です。ただし、z/OS 内の構造の最大サイズより小さい値を指定する必要があります。これを超えると、z/OS のエラーが発生します。例えば z/OS バージョン 1 リリース 12 では、構造の最大サイズは 1048576 MB (1 TB) です。CFRM パラメーターの詳細については、z/OS MVS シスプレックスのセットアップを参照してください。

一時記憶域リスト構造の定義について詳しくは、168 ページの『一時記憶域データ共用のための一時記憶域プールの定義』を参照してください。

## デバッグ・トレース・パラメーター

これらのパラメーターは、集中的なデバッグ・トレースにのみ使用されます。

これらのオプションを実稼働環境で使用すると、パフォーマンスに大きな影響を与え、印刷ファイルが急激に増えてスプール・スペースを使いすぎる原因になる点にご注意ください。

仮想記憶間の要求からのトレース・メッセージは、その生成がトレース印刷サブタスクが処理できるよりも早いと、失われる可能性があります。その場合、トレースには、失われたメッセージの数のみが示されます。



**TRACECF={OFF|ON}**

カップリング・ファシリティのインターフェースのデバッグ・トレース・オプションを、OFF または ON で指定します。このオプションは、カップリング・ファシリティの要求インターフェースへの主なパラメーターと IXLLIST マクロからの結果を示すトレース・メッセージを、印刷ファイルに作成します。

このキーワードは、**CFTR** または **CFTRACE** としてコーディングすることもできます。

**TRACERQ={OFF|ON}**

キュー要求のデバッグ・トレース・オプションを、OFF または ON で指定します。このオプションは、共用キュー要求または共用キュー照会インターフェースに入力する主なパラメーターと、出口における結果を示すトレース・メッセージを印刷ファイルに作成します。

このキーワードは、**RQTR** または **RQTRACE** としてコーディングすることもできます。

**チューニング・パラメーター**

これらのパラメーターはチューニングを行う目的で指定しますが、通常は省略可能で、TS サーバーではデフォルト値を使用するように指定できます。

**ELEMENTRATIO={1|number}**

構造が最初に割り振られる際の「項目対エレメント」率のエレメントの部分を指定する MVS カップリング・ファシリティ・リソース管理 (CFRM) パラメーター。この値は、データ・エレメント用に最初に設定される構造のスペースの比率を決定します。

「項目対エレメント」率の理想的な値は、各項目のデータの平均サイズをエレメントのサイズで割った値です。ただし、実際の項目やエレメントの使用率に応じて、サーバーが自動的に比率を調整します。

このパラメーターは、サーバーの初期化時にのみ有効になり、構造が最初に割り振られる際にのみ使用されます。有効な範囲は 1 から 255 までです。

このキーワードを **ELEMRTATIO** として省略することもできます。

カップリング・ファシリティ内でのこのパラメーターについて詳しくは、[z/OS MVS プログラミング: シスプレックス・サービス ガイド](#)を参照してください。

**ELEMENTSIZE={256|number}**

構造のスペースのエレメントのサイズを指定する MVS CFRM パラメーターで、2 の累乗である必要があります。このパラメーターは、カップリング・ファシリティのログ・ストリームにのみ使用されます。有効な範囲は 256 から 4096 です。現在のカップリング・ファシリティの実装では、このパラメーターをデフォルト値の 256 から変更する理由はありません。

このパラメーターは、サーバーの初期化時にのみ有効になり、構造が最初に割り振られる際にのみ使用されます。

このキーワードを **ELEMSIZE** として省略することもできます。

**ENTRYRATIO={1|number}**

構造が最初に割り振られる際の「項目対エレメント」率の項目の部分を指定する MVS CFRM パラメーター。この値は、リスト項目制御用に最初に設定される構造のスペースの比率を決定します。

実際の使用量によっては必要に応じてスペースの使用率を改善するためにサーバーが自動的に比率を調整するため、このパラメーターの指定は必須ではありません。

このパラメーターは、サーバーの初期化時にのみ有効になり、構造が最初に割り振られる際にのみ使用されます。有効な範囲は 1 から 255 までです。

カップリング・ファシリティ内でのこのパラメーターについて詳しくは、[z/OS MVS プログラミング: シスプレックス・サービス ガイド](#)を参照してください。

**LASTUSEDINTERVAL={00:10|hh:mm}**

大きなキューが前回使用された時間を更新する頻度を指定する CICS パラメーター。

小さいキューの場合、参照されるたびに前回使用された時間が更新されます。大きいキューの場合、前回使用された時間を更新するには追加のカップリング・ファシリティ・アクセスが必要になるため、この更新が発生するのは、現在の時間のこのインターバル内で、キューがまだアクセスされていない場合のみです。そのため、**INQUIRE** コマンドで戻される前回使用された時間間隔は、実際の値より、最

大でこのパラメーターの値の分だけ大きい場合があります。前回使用した時間は、主にキューが使用できないかどうかを判断するために使用します。そのため、**LASTUSEDINTERVAL** パラメーターの適切な値は、通常は数分です。

**INQUIRE TSQUEUE / TSQNAME** を参照してください。

有効な範囲は 00:00 から 24:00 までです。この値は、秒単位でも指定できます。

このキーワードを **LASTUSEDINT** として省略することもできます。

#### **SMALLQUEUEITEMS={9999|number}**

キュー索引入力のデータ領域に、小さいキュー形式で保管できる項目の最大数を指定する CICS パラメーター。このパラメーターは、小さい項目が多数ある場合に、大きいキュー形式に変換されるようにキューを強制することができます。小さいキューのデータ領域全体を毎回再書き込みするより、項目を個別に書き込む方が効率的な場合があります。

有効な値は 1 から 32767 までです。

#### **SMALLQUEUESIZE={32K|number}**

各データ項目に追加される 2 バイト長の接頭部を含む、小さいキューのデータの最大サイズを指定する CICS パラメーター。2 番目または後続の項目が書き込まれるとキューが最大サイズを超える場合、キューは大きいキュー形式に変換されます。

このパラメーターは、32K より小さいサイズの大きいキュー形式に変換されるようにキューを強制することができます。これにより、小さいキュー形式に大量のデータが書き込まれることを防ぐことができます。非同期のカップリング・ファシリティーの処理によりハードウェア・リソースの競合が発生するシステムでは、このパラメーターを使用するとパフォーマンスが改善されます。ただし、多くのシステムでは、最大サイズの 32K に達するまで変換を行わないほうが効率的です。

有効な範囲は 4096 から 32768 までです。

#### **警告パラメーター**

これらのパラメーターは、構造が満杯に近づくとき警告メッセージが出されて **ALTER** アクションが自動で実行される時点のしきい値を変更します。

#### **ELEMENTWARN={80|number}**

警告と自動 **ALTER** アクションが最初にトリガーされる時点の、使用中の要素のパーセンテージを指定します。

有効な範囲は 1 から 100 です。

このキーワードは、**ELEMWARN** としてコーディングすることもできます。

#### **ELEMENTWARNINC={5|number}**

次の警告が発行されるまでの使用中の要素のパーセンテージの増加 (または減少) (次の増加で 100 に達する場合は 1 に減らす) を指定します。追加のメッセージは、使用中の要素の数が変更されると発行されます。これらのメッセージは、使用中の要素が、最初の警告レベルよりこのパーセンテージ分以上減ると出されなくなります。

有効な範囲は 1 から 100 です。

このキーワードは、**ELEMWARNINC** としてコーディングすることもできます。

#### **ENTRYWARN={80|number}**

警告と自動 **ALTER** アクションが最初にトリガーされる時点の、使用中の項目のパーセンテージを指定します。

有効な範囲は 1 から 100 です。

#### **ENTRYWARNINC={5|number}**

次の警告が発行されるまでの使用中の項目のパーセンテージの増加 (または減少) (次の増加で 100 に達する場合は 1 に減らす) を指定します。追加のメッセージは、要素の数が変更されると発行されます。これらのメッセージは、使用中の項目が、最初の警告レベルよりこのパーセンテージ分以上減ると出されなくなります。

有効な範囲は 1 から 100 です。

### 自動 ALTER パラメーター

構造が満杯に近づくとサーバーが自動 ALTER アクションの実行を試行する際の条件を変更するには、以下のパラメーターを定義します。

キュー・サーバーの自動 ALTER プロセスの詳細については、[176 ページ](#)の『[キュー・サーバー 自動 ALTER 処理](#)』を参照してください。

#### **ALTERELEMMIN={100|number}**

過剰なエレメント数とその数を超えると自動 ALTER を発行して項目に変換する、最小数を指定します。

有効な範囲は 1 から 999999999 までです。

#### **ALTERELEMPC={1|number}**

過剰なエレメントがそのパーセンテージを超えると自動 ALTER を発行して項目の比率を増やす、最小パーセンテージを指定します。

有効な範囲は 0 から 100 までです。

#### **ALTERENTRYMIN={100|number}**

過剰な項目数とその数を超えると自動 ALTER を発行してそれらの項目をエレメントに変換する、最小数を指定します。

有効な範囲は 0 から 999999999 までです。

#### **ALTERENTRYPC={1|number}**

過剰なエントリーがそのパーセンテージを超えると自動 ALTER を発行してエレメントの比率を増やす、最小パーセンテージを指定します。

有効な範囲は 0 から 100 までです。

#### **ALTERMININTERVAL={00:10|hhmm}**

構造が満杯に近づいたとき (エレメントまたは項目の警告レベルを超えたとき) に行う自動 ALTER の試行と試行の間の間隔の最小時間を指定します。

有効な範囲は 00:00 から 24:00 までです。

このキーワードは、**ALTERMININT** としてコーディングすることもできます。

## キュー・サーバー 自動 ALTER 処理

キュー・サーバーは、要求のたびにカップリング・ファシリティから返される情報を使用して、構造で使用されているエレメントとエントリーの合計数をモニターします。

使用数が指定したしきい値を超えると、警告メッセージ DFHXQ0411 または DFHXQ0412 が発行され、使用数がしきい値を超えて増加するたびに警告が繰り返されます。

警告が発行されるたびに、サーバーはエントリー/エレメント比率に対する自動 ALTER を実行すべきかどうかテストします。このテストは、一方を完全に使い果たした場合に残るエレメントまたはエントリーの数を計算して実行されます。これは、現在使用されているエレメント数とエントリー数の比率に基づいて計算されます。

次の場合に、エントリー/エレメントの比率を、現在使用されている実際のエントリー数/エレメント数の比率に変更するために、IXLALTER 要求が発行されます。

- 余分なエレメント数またはエントリー数が、ALTERELEMMIN または ALTERENTRYMIN パラメーターに指定された数を超えた場合。
- 合計に対する割合で表される上記と同じ数が、ALTERELEMPC または ALTERENTRYPC パラメーターに指定された数を超えた場合。

1 つの構造に対してアクティブになる ALTER 要求は一度に 1 つのみです。あるサーバーで ALTER 処理が開始された場合、別のサーバーの ALTER は拒否されます。ただし、ALTER が完了すると、システムはすべてのサーバーに自動的に通知し、各サーバーがそれぞれの状況情報を更新できるようにエレメントとエントリーの新しい数を知らせます。

次の場合は、少なくとも ALTER の最小間隔 (ALTERMININTERVAL パラメーターで指定) が経過するまで、次の ALTER の試行が抑制されます。

- 何らかの形式の ALTER が (サーバーまたはオペレーターの SETXCF ALTER コマンドによって) 既に最近使用された。
- 前回の試行から構造のスペース使用量が警告レベルを超えたままである。

## 共用 TS キュー・サーバーのコマンド

サーバー領域のジョブ名または開始タスク名を指定した MVS MODIFY (F) コマンドを使用して、キュー・サーバーを制御するためのコマンドを発行できます。

キュー・サーバー・コマンドの一般的な形式は次のとおりです。

```
F server,cmd parameter,parameter...  comments
```

MVS STOP コマンドは、MVS MODIFY コマンドを使用してサーバー・コマンド STOP を発行するのと同じです。

キュー・サーバーは、以下のコマンドをサポートします。

### SET keyword=value

1 つ以上のサーバー・パラメーター値を変更します。これは、初期設定専用のパラメーターと示されているものを除くすべてのパラメーターに適用されます。MVS SET コマンドと同じく、このコマンドは T という省略形にすることもできます。

### DISPLAY keyword

コンソールに 1 つ以上のパラメーター値、または統計の要約情報を表示します。DISPLAY および PRINT の有効なパラメーター・キーワードについては、このセクションで後述します。MVS DISPLAY コマンドと同じく、このコマンドは D という省略形にすることもできます。

### PRINT keyword

DISPLAY と同じ出力を生成しますが、印刷ファイルにのみ出力します。

### STOP

サーバーを終了します。アクティブな接続が終了するのを待ってから、新たに接続されないようにします。

MVS STOP コマンドの場合と同じように、このコマンドは P として省略できます。

### CANCEL {RESTART={NO|YES}}

サーバーを即時に停止します。自動リスタートを要求する必要があるかどうかを指定します。

CANCEL RESTART については、[178 ページの『CANCEL コマンドのオプション』](#)を参照してください。

サーバーは、構造サイズを変更するオペレーター・コマンド SETXCF のような XES イベントにも応答します。サーバーは、カップリング・ファシリティにアクセスできなくなると、サーバー **CANCEL** コマンドを自動的に発行して、すぐにサーバー自体をクローズします。

### DISPLAY キーワードと PRINT キーワード

DISPLAY または PRINT は、すべてのシステム初期設定パラメーターの値を表示することができます。

これらのキーワードは、以下の追加情報も表示できます。

### ARMREGISTERED

ARM 登録が正常に完了したかどうかを示します (YES または NO)。

### CONNECTIONS

このサーバーに現在接続されている領域のジョブ名と APPLID のリスト。このキーワードは、**CONN** とコーディングすることもできます。

### 統計の要約

#### BUFSTATS

キュー索引バッファー・プールの統計。

このキーワードは、**BUFST** とコーディングすることもできます。

#### CFSTATS

カップリング・ファシリティのインターフェース I/O および応答の統計。

このキーワードは、**CFST** または **STATSCF** とコーディングすることもできます。

#### **POOLSTATS**

プール・リスト構造全体の使用状況の統計。

このキーワードは、**POOLST** とコーディングすることもできます。

#### **STORAGESTATS**

サーバー・アドレス・スペースの主記憶域割り振りの統計。

このキーワードは、**STGST**、**STGSTATS**、または **STORAGEST** とコーディングすることもできます。

#### 情報の結合リストを表すキーワード

##### **PARAMETERS**

メインとなるパラメーター値。

このキーワードは、**PARM**、**PARMS**、または **PARAM** とコーディングすることもできます。

##### **ALLPARAMETERS**

すべてのパラメーター値。

このキーワードは、**ALLPARMS** とコーディングすることもできます。

##### **STATISTICS**

使用可能なすべての統計。

このキーワードは、**STAT** または **STATS** とコーディングすることもできます。

##### **INITIALIZED**

初期設定が完了すると通常表示される値を持つ、選択されたパラメーターおよび統計。

このキーワードは、**INIT** とコーディングすることもできます。

##### **ARM**

すべての ARM 関連のパラメーター値を表示します。

- ARMELEMENTNAME
- ARMELEMENTTYPE
- ARMREGISTERED

このキーワードは、**ARMSTATUS** とコーディングすることもできます。

#### **CANCEL コマンドのオプション**

CANCEL コマンドを使用して、自動再始動を要求できます。

以下のパラメーターを指定します。

##### **RESTART={NOvYES}**

自動再始動を要求するかどうかを指定して、サーバーをすぐに強制終了します。デフォルト値は **RESTART=NO** です。

サーバーでカップリング・ファシリティの接続に関するリカバリー不能な問題が起きて、接続が失われたり、ストラクチャーで障害が発生したりすると、サーバーは、**CANCEL RESTART=YES** コマンドを使用して自身をキャンセルします。その結果、既存の接続が終了して、サーバーがシャットダウンします。その後、サーバー・ジョブの新しいインスタンスが開始されます。

サーバー・コマンド **CANCEL RESTART=YES** または **MVS** コマンド **CANCEL jobname,ARMRESTART** を使用して、サーバーを明示的に再始動することもできます。

**RESTART=YES** の代わりに **RESTART**、**RESTART=NO** の代わりに **NORESTART** を直接入力することも可能です。



## キュー・プールのアンロードと再ロード

キュー・プールのコンテンツを順次ファイルにアンロードして、その後、FUNCTION パラメーターを使用してサーバー・プログラムを実行することで再ロードすることができます。

### このタスクについて

これを使用して、計画されたカップリング・ファシリティのメンテナンス全体でキュー・プールを保存したり、キュー・プールを別のカップリング・ファシリティに移動したりすることができます。サーバーでは構造の自動 REBUILD をサポートしませんが、アンロードと再ロードのプロセスの方が、より柔軟です。再ロードされたキュー・プールは、元のプールと同じ名前にする必要はありません。

### FUNCTION={UNLOAD|RELOAD}

サーバーに特殊機能 UNLOAD または RELOAD の実行を要求します。アンロードまたは再ロードの処理が完了すると (正常終了または異常終了)、サーバー・プログラムは終了します。

このパラメーターを省略すると、サーバー・プログラムは仮想記憶間キュー・サーバー環境を初期化します。

UNLOAD または RELOAD が指定されると、サーバー・プログラムはリスト構造を排他使用する必要があります。構造が通常のサーバーによって現在使用されている場合、アンロードまたは再ロードの試行は拒否されます。同様に、アンロードまたは再ロード機能の進行中に通常のサーバーを始動しようとする、構造への共有アクセスが使用できないため、試行は拒否されます。

UNLOAD および RELOAD には、通常のサーバー・パラメーターをどれでも指定することはできますが、それらのうちの多くは (キュー・バッファの数など)、アンロードまたは再ロード処理には関係ないため、無視されます。

UNLOAD 機能には、キュー・プールの アンロード先の順次データ・セットを説明するファイル名 DFHXQUL の DD ステートメントが必要です。アンロードされたファイルの形式は、次のとおりです。

```
RECFM=F,LRECL=4096,BLKSIZE=4096.
```

サーバーによって作成されるプール使用量の統計から、データ・セットの合計サイズの上限 (バイト数) を見積もることができます。データ・サイズの合計 (バイト) は、使用中の要素の数に要素・サイズ (通常は 256) を乗算して、さらにキューごとにある制御情報 (通常はキューごとに 100 バイト未満) を足したものです。データ・要素内の未使用のスペースはアンロードされたファイルには含まれないため、実際のサイズは、通常はこれより小さくなります。UNLOAD JCL の例については、[179 ページの図 34](#) を参照してください。

```
//UNLDTSQ1 JOB    ...
//TSUNLOAD EXEC  PGM=DFHXQMN          CICS TS queue server program
//STEPLIB DD     DSN=CICSxxx.SDFHAUTH,DISP=SHR  Authorized library
//SYSPRINT DD    SYSOUT=*              Options, messages and statistics
//DFHXQUL DD     DSN=TSQ1.UNLOADED.QPOOL,  Unloaded queue pool
//              DISP=(NEW,CATLG),
//              SPACE=(4096,(10000,1000))    Estimated size in 4K blocks
//SYSIN DD      *
FUNCTION=UNLOAD          Function to be performed is UNLOAD
POOLNAME=PRODTSQ1       Pool name
/*
```

図 34. UNLOAD JCL の例

RELOAD 機能には、キュー・プールの再ロード元の順次データ・セットを説明するファイル名 DFHXQRL の DD ステートメントが必要です。必要な場合は、再ロード中に構造が割り振られ、その際に通常のサーバーの実行で使用する構造属性を制御するサーバー・パラメーターを使用できます。RELOAD 処理では、すでにキュー・プールにあるすべてのキューを迂回します。これは、例えば構造が小さすぎたため、ALTER を使用してサイズを大きくしてから再ロード・ジョブを再始動する必要があった場合などに起こります。

プールが満杯に近い (項目や要素の空きが 5% 未満である) 場合、まったく同じサイズの構造でアンロードや再ロードができる保証はありません。使用可能なスペースは、現在の「項目対要素」率の影響を受け、自動 ALTER プロセスによっておおまかにしか制御されません。



再ロード中に構造のスペースが警告レベルに達すると、自動 ALTER プロセスは「項目対エレメント」率の調整を試行し、ALTER の進行中に再ロードのスペースが足りなくなると、再ロード・プロセスは ALTER が完了するまで待機します。

スペース不足で RELOAD が失敗した場合、結果として表示されるメッセージには再ロードされたキューの数と、失敗した時点までに読み取られたブロックの数が含まれます。これらの値を、元の UNLOAD のメッセージと比較して、未ロードのキューとデータの数を出すことができます。RELOAD JCL の例については、180 ページの図 35 を参照してください。

```
//RELDTSQ1 JOB ...
//TSRELOAD EXEC PGM=DFHXQMN CICS TS queue server program
//STEPLIB DD DSN=CICSxxx.SDFHAUTH,DISP=SHR Authorized library
//SYSPRINT DD SYSOUT=* Options, messages and statistics
//DFHXQRL DD DSN=TSQ1.UNLOADED.QPOOL,DISP=OLD Unloaded queue pool
//SYSIN DD *
FUNCTION=RELOAD Function to be performed is RELOAD
POOLNAME=PRODTSQ1 Pool name
POOLSIZE=50M Increased pool size
MAXQUEUES=10000 Increased number of big queues
/*
```

図 35. RELOAD JCL の例

## カップリング・ファシリティ・データ・テーブル・サーバーのセットアップと実行

CICS カップリング・ファシリティ・データ・テーブルは、更新の保全性により、シスプレックス内で作業データを高速で共用できます。データ・テーブルは、カップリング・ファシリティ構造内に保持され、指定したサーバーを介してアクセスされます。MVS イメージ内のプールごとに 1 つのサーバーをセットアップする必要があります。

### 始める前に

指定したカップリング・ファシリティ・データ・テーブル・プール用のサーバーを始動する前に、そのプールで使用されているカップリング・ファシリティ構造を定義してください。カップリング・ファシリティ・リスト構造の定義については、185 ページの『カップリング・ファシリティ・データ・テーブル・プールの定義』を参照してください。

### このタスクについて

関連するカップリング・ファシリティ・データ・テーブルのグループごとに別のプールに配置できます。例えば、実動用のプールと、テスト用のプールを作成できます。プールは、カップリング・ファシリティ・リソース管理 (CFRM) ポリシーでリスト構造として定義されます。プール名と接頭部 DFHCF を使用してサーバー名が形成されます。プール名は、サーバーの始動 JCL で指定されます。

### 手順

1. カップリング・ファシリティ・データ・テーブル・サーバー・ジョブを定義して開始し、MVS バッチ領域で実行します。
2. サーバーが実行している間、さまざまな運用タスクを実行できます。
  - MVS コマンドを使用してサーバー領域を制御できます。
  - カップリング・ファシリティ・データ・テーブルのプールを削除したり、空にしたりできます。
  - 順次データ・セットとの間でプールの内容をアンロードしたり再ロードしたりできます。

## カップリング・ファシリティー・データ・テーブル・サーバーの概要

CICS カップリング・ファシリティー・データ・テーブルは、更新の整合性を保持してシスプレックス内で作業データを素早く共用するために設計されています。

データが保持されるテーブルは、ユーザー管理の共用データ・テーブルに多くの点で類似しています。また、データの保管および検索に使用する API は、ユーザー管理データ・テーブルに使用するファイル制御 API に基づいています。

[182 ページの図 36](#) は、カップリング・ファシリティー・データ・テーブル・サーバーに 3 つの CICS AOR がリンクされた Parallel Sysplex を表しています。

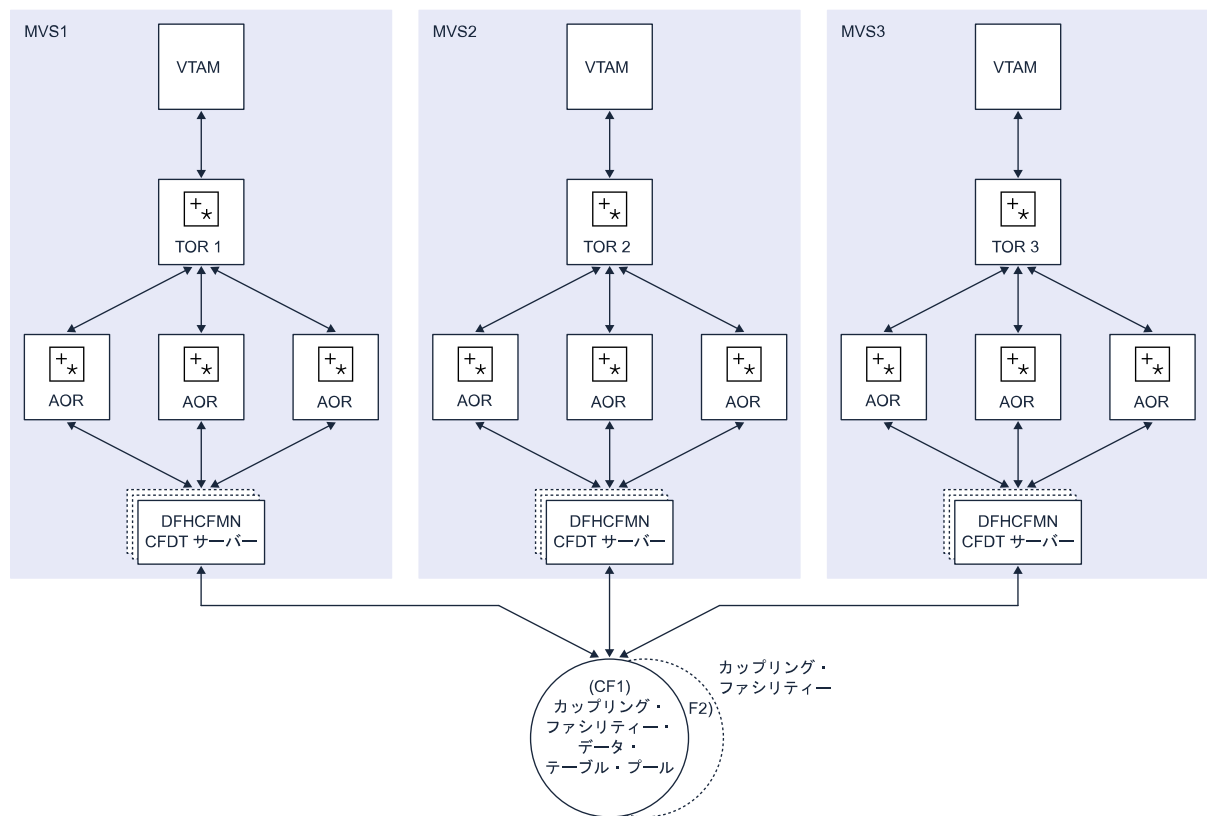


図 36. Parallel Sysplex のカップリング・ファシリティ・データ・テーブル・サーバーに関する概念ビュー

## カップリング・ファシリティ・データ・テーブル

カップリング・ファシリティ・データ・テーブル (CFDT) には、ファイル・データ共有の手段が用意されています。そのために使用するのは、CICS のファイル管理機能であり、ファイル所有領域は不要です。VSAM RLS サポートも不要です。CICS CFDT サポートによって、作業中のデータをシスプレックス内で手早く共有できます。更新時の整合性も確保できます。

カップリング・ファシリティのデータは、ユーザー保守の共有データ・テーブルとよく似たテーブルに保管されます。CFDT のために必要なリソースを MVS カップリング・ファシリティ・リソース管理 (CFRM) ポリシーで定義する必要があります。

この形式のテーブルは、読み取りアクセスと書き込みアクセスのパフォーマンスがほぼ同じなので、スクラッチパッド・データに適しています。代表的な使用法として、シスプレックス内の CICS 領域間のスクラッチパッド・データの共有や、変更を永久に保管する必要のないファイルの共有などが考えられます。スクラッチパッドについては多種多様な要件がありますが、そのほとんどは CFDT で実装できます。

CFDT は、さまざまなテーブルにデータをグループ分けして項目をキーで指定したり取得したりする場合にも適しています。以下に例を示します。

- CFDT のフィールドを使用して、注文処理アプリケーションで次に使用できる空き注文番号を管理できます。
- CFDT で紛失したクレジット・カードの番号のリストを管理できます。

アプリケーション側から見ると、CFDT は、シスプレックス全体を対象にしたユーザー保守のデータ・テーブルとよく似ています。その種のデータ・テーブルの場合と同じく、ファイル制御 API を使用してアクセスするからです。ただし、CFDT の場合は、16 バイトという最大キー長制限があります。

### カップリング・ファシリティ・データ・テーブルのモデル

CFDT では、競合モデルとロック・モデルのいずれかを使用できます。各テーブルで使用するモデルをそのテーブルのファイル・リソース定義で指定できるので、テーブルごとに別々のモデルを使用することも可能です。

- 競合モデルを使用するとパフォーマンスを最適化できますが、通常は、このモデルを活用するためのプログラムが必要です。特にこのモデルで使用するための CHANGED 条件コード (アプリケーション・プログラムが更新用の読み取り要求を実行した後にデータが変更されたことを示すコード) があるからです。競合モデル用に作成したわけではないプログラムでは、この条件を正しく処理できない場合があります。CHANGED 応答は、REWRITE コマンドまたは DELETE コマンドで起こります。競合モデルでは、REWRITE または DELETE で NOTFND 応答が返される状況もあります。

このモデルはリカバリー不能です。作業単位で障害が発生した時に、CFDT の更新はバックアウトされません。

- ロック・モデルの場合は、ファイル管理 API の UMT サブセットに準拠したプログラムとの間で API を共有できます (UMT サブセットは、フルセットのファイル管理 API とほぼ同じです)。このモデルには、以下のタイプがあります。

- リカバリー不能。

同期点までロックが続きません。作業単位で障害が発生した時に、CFDT の更新はバックアウトされません。

- リカバリー可能。

作業単位や CICS 領域で障害が発生した後に CFDT をリカバリーできます (CICS で障害が発生した時に進行中だった作業単位による更新がバックアウトされます)。

リカバリー可能なロック・モデルは、未確定状態での障害やバックアウト時の障害に対応しています。CFDT 更新のバックアウト時に作業単位で障害が発生したり、同期点処理時に未確定状態で障害が発生したりすると、ロックが保持ロックに変換され、作業単位が中断されます。

### サーバー接続

クライアント CICS 領域は、サーバーのプールを参照する要求を最初に実行する時に、そのサーバーへの仮想記憶間接続を確立します。そのサーバー接続が失敗すると、カップリング・ファシリティ・データ・

テーブルに影響が及びます。[235 ページの『サーバー接続管理』](#)および[名前付きカウンターのリカバリー](#)を参照してください。

### カップリング・ファシリティ・データ・テーブルの構造とサーバー

カップリング・ファシリティ・データ・テーブル (CFDT) は、カップリング・ファシリティ・リスト構造内に保持されます。CFDT へのアクセスは、名前付きの CFDT サーバーにより行われます。

CFDT サーバーは、共用データ・テーブル・ファイル専有領域と同様と考えることができます。共用データ・テーブルのサポートについては、[データ・テーブルの共用環境](#)を参照してください。CFDT サポートを使用すると、CFDT の関連グループを別々のプールに保管して分離できます。例えば、1 つのプールは実動用、別のプールはテスト用にすることができます。

CFDT プールは、カップリング・ファシリティ・リスト構造で、カップリング・ファシリティ・データ・テーブル・サーバーによりこのプールにアクセスできます。各 MVS イメージ内で、そのイメージ内の CICS 領域によりアクセスされる CFDT プールごとに 1 つずつ CFDT サーバーがなければなりません。

このサーバーの名前は、サーバー・アドレス・スペース名とプール名を組み合わせで形成されます。例えば、FACILITY クラス RACF プロファイルの場合、サーバー名は `DFHCF.poolname` になります。

カップリング・ファシリティ・リスト構造は、カップリング・ファシリティ・リソース管理 (CFRM) ポリシーで定義されます。CFDT の構造名は、`DFHCFLS_poolname` の形式になります。その後、CFDT プール名を CFDT サーバーに関する始動 JCL で指定します。

### ファイル制御 API を使用するアクセス

CFDT は、ファイル制御コマンドにより CICS からアクセスされます。

コマンドで指定されるファイル名は、テーブルの名とそのテーブルがあるプールの名前を示します。テーブル名は、ファイル定義内で指定されているか、ファイル定義と同じです。プール名は、ファイル定義内で指定されています。テーブルはプール名とテーブル名で一意に識別されるので、同じ名前の 2 つのテーブルが別々のプール内に存在することができ、エンティティは完全に異なります。

### CFDT プールに対する自動接続

特定のプール内の CFDT が初めて参照される際に、CICS はそのプールの CFDT サーバーに接続します。CFDT サーバーが失敗後に再始動する際にも、CICS はそのサーバーに自動的に再接続します。

CFDT サーバーは、そのサーバーを呼び出す CICS 領域による誤用に対して保護されるので、システム保全性が保証されます。呼び出し側の領域が機密パラメーターを許可関数に変更できないように保護されます。

同様に、CICS は CFDT サーバーの障害時の副作用から保護されます。CICS 領域から CFDT サーバーにファイル制御要求を発行して失敗すると、その結果の MVS 異常終了がトラップされ、アプリケーション・プログラムに `SYSIDERR` 条件として返されます。

### CFDT の作成と削除

初めての参照でカップリング・ファシリティ・データ・テーブル (CFDT) を開く必要がある場合、CFDT が CICS により自動的に作成されます。その後この CFDT は同じ領域で使用されるか、同じカップリング・ファシリティ・データ・テーブル名を指定して他のファイルのオープン要求を発行する他の CICS 領域で使用されます。

オプションで、CFDT が初めて開かれる際に、CICS によりソース VSAM (KSDS) データ・セットから CFDT を自動的にロードできます。ユーザーが保守するデータ・テーブルと違って、CFDT の使用時にはソース・データ・セットを関連付けずに指定できるので、空の CFDT を作成できます。

CFDT の作成直後にアプリケーション・プログラムからアクセスできます。ただし、データのロード中にアクセスできるキーに関する制約事項があります。

CFDT をロードすると、そのテーブルにアクセスする CICS 領域の動作やそのテーブルをロードした CICS 領域の動作から独立したエンティティになります。すべての CICS 領域が通常どおり終了したり異常終了したりしても、構造かカップリング・ファシリティを明示的に削除する処置を取らなければ、CFDT はカップリング・ファシリティ内に残ります。

CFDT のコンテンツや構造を削除するには、以下のコマンドを使用できます。

```
MODIFY cfdt_server, DELETE TABLE=name
```

## CFDT の管理

CICS で提供されているユーティリティー関数を使用して、カップリング・ファシリティ・データ・テーブル・サーバーから、プール内で定義されている CFDT に関する要約情報や定期的な統計を取得できます。[カップリング・ファシリティ・データ・テーブルの統計および『Reference』の『カップリング・ファシリティ・データ・テーブル・プール・レポート』](#)を参照してください。

カップリング・ファシリティ・データ・テーブル・プールを管理する際や、能力を評価するために、この情報を使用できます。詳しくは、[188 ページの『カップリング・ファシリティ・データ・テーブル・サーバーのパラメーター』](#)を参照してください。

## カップリング・ファシリティ・データ・テーブル・プールの定義

シスプレックス結合データ・セットの中で、カップリング・ファシリティ・リソース・マネージャー (CFRM) ポリシーのカップリング・ファシリティ・データ・テーブル (CFDT) のリスト構造を定義します。

CICS は、1 つ以上の CFDT を含めることができる CFDT プールに、仮想記憶間サービスを使用してサーバー領域を介してアクセスします。カップリング・ファシリティ・データ・テーブル領域の設定と開始に関する詳細は、[180 ページの『カップリング・ファシリティ・データ・テーブル・サーバーのセットアップと実行』](#)を参照してください。

アプリケーション側の観点からすると、プールとそのサーバーはファイルを所有する領域によく似ており、プールには各テーブル名が固有である限り、任意の数のテーブルを含めることができます。

CFDT サーバーがそのプールを使用するには、その前に、アクティブな CFRM ポリシーに、そのプールのために使用するリスト構造の定義が含まれていなければなりません。CFRM 構造定義は、リスト構造のサイズとそれを格納することが可能なカップリング・ファシリティの設定リストを指定します。選択された CFRM ポリシーにリスト構造を指定するステートメントを追加した後、そのポリシーをアクティブにする必要があります。

必要となるリスト構造のサイズ、およびカップリング・ファシリティ内のその位置を指定するには、CFRM ポリシー定義ユーティリティー IXCMIAPU を使用します。

CFDT プールのリスト構造の名前を作成するには、プール名を接頭部 DFHCFLS\_ に付加して DFHCFLS\_*poolname* とします。

## IXCMIAPU を使用したポリシーの更新

管理データ・ユーティリティー IXCMIAPU を使用して、CFRM 結合データ・セットの管理ポリシーを更新できます。このユーティリティーでは、管理ポリシーのポリシー・データのみ追加または変更され、アクティブな CFRM ポリシーのシステム・コピー内の情報は変更されません。

このユーティリティーを実行するジョブの例については、SYS1.SAMPLIB ライブラリーのメンバー IXCCFRMP を参照してください ([『z/OS MVS Setting Up a Sysplex』の『CFRM ポリシー・データ向け管理データ・ユーティリティー』](#)を参照)。

次の例は、カップリング・ファシリティ・データ・テーブル・プールのポリシー・ステートメントを示しています。

```
STRUCTURE NAME(DFHCFLS_PRODCT1)
  SIZE(79872)
  INITSIZE(32768)
  PREFLIST(FACIL01,FACIL02)
```

## CFRM ポリシーのアクティブ化

CFRM ポリシーを定義したら、そのポリシーをアクティブ化する必要があります。次の MVS コマンドを使用します。

```
SETXCF START,POLICY,POLNAME=policyname,TYPE=CFRM
```



リスト構造の定義を含む CFRM ポリシーをアクティブ化しても、構造は作成されません。この構造が作成されるのは、その構造に対する接続が最初に行われた時で、この接続の試行は、対応するプールを参照する最初のカップリング・ファシリティ・データ・テーブル (CFDT) サーバーが始動されたときに行われます。

CFDT サーバーがリスト構造を作成するときに、その構造に初期サイズが割り振られます。このサイズは、CFRM ポリシーの指定に従って最大サイズまで増加できます。すべての構造サイズは、割り振り時にカップリング・ファシリティ・レベル (CFLEVEL) の次のストレージ増分値に切り上げられます。例えば、CFLEVEL 16 の場合、サイズは最も近い 1 MB 単位の値に丸められます。

カップリング・ファシリティでスペースが使用可能な場合、リスト構造をその初期サイズから最大サイズまで動的に拡張したり、リスト構造のサイズを縮小して他の目的で使えるようカップリング・ファシリティのスペースを解放したりできます。

初期構造の割り振りがいっぱいになると、割り振り済みの構造が指定された最大サイズより小さい場合でも、構造は自動的に拡張されません。割り振りがいっぱいになった場合にリスト構造を拡張するには、次の **SETXCF** コマンドを使用することにより、リスト構造を (最大サイズまで) 拡張できます。

```
SETXCF START,ALTER,STRNAME=DFHCFLS_poolname,SIZE=nnnn
```

この方法でリスト構造のサイズを動的に大きくした場合には、その新しいサイズを反映するように CFRM **INITSIZE** パラメーターが更新され、後でその構造を再作成または再ロードした場合に、元のサイズに戻らなくなります。

### カップリング・ファシリティ・データ・テーブルのストレージの計算

z Systems カップリング・ファシリティ構造サイズ決定支援ツール (CFSizer) を使用して、カップリング・ファシリティのデータ・テーブル・リスト構造用のストレージ所要量を計算することができます。

カップリング・ファシリティの構造には、キー順データ・セットと同様に、保管データと、そのデータの管理とアクセスに必要な情報の両方が含まれます。カップリング・ファシリティ内のエントリーごとに、データは固定サイズ (普通は 256 バイト) のエレメントのチェーンとして保管されます。したがって、正しい長さの可変長データを個別に保管しなければなりません。そのために、CICS では保管データに長さ接頭部を組み込みます。したがって、スペース計算では接頭部の長さに各エントリーの正確な長さを加算します。内部制御情報の量は、現行のカップリング・ファシリティ・レベル (CFLEVEL) のカップリング・ファシリティ制御コードの機能とパフォーマンスのレベルに応じて異なります。CFLEVEL が高いほどストレージ要件が大きくなる可能性があります。詳細については、[232 ページの『カップリング・ファシリティのストレージ管理』](#)を参照してください。

CFSizer は、これらの要素を加味したうえで現行の CFLEVEL でカップリング・ファシリティと通信して、ストレージ所要量を計算する Web ベース・アプリケーションです。[CFSizer](#) を参照してください。

CFSizer を使用して、データ・テーブルリスト構造のストレージ所要量を計算するには、以下の情報を入力します。

#### テーブルの最大数

データ・テーブル構造に格納できるデータ・テーブルの最大数。この値は、MAXTABLES サーバー・パラメーターに対応します。[191 ページの『リスト構造パラメーター』](#)を参照してください。

すべてのデータ・テーブルを処理できる十分な大きさの値を指定します。ただし、事前に割り振られたリスト・ヘッダーが使用されずに、カップリング・ファシリティの大量のストレージを消費することがないように注意してください。有効な範囲は 1 から 999999 までです。デフォルトは、1000 です。

#### 丸められたレコードの平均サイズ

各データ・レコードに必要なストレージ量の平均。各レコードには、2 バイト長の接頭部があり、1 つ以上の 256 バイトのエレメントとして保管されます。この値によって、必要な構造サイズを計算するために使用する「エントリー対エレメント」率が決まります。有効な範囲は 1 から 32768 までです。デフォルトは 512 です。

すべてのレコードが同じようなサイズの場合、この値は、データ・サイズの平均値を取って 2 を足し、次の 256 の倍数に切り上げて計算します。必要なエレメントのストレージのサイズは、データ・レコードのサイズより 2 バイト大きいサイズです (各項目の接頭部の長さ分)。

各データ・レコードのサイズが異なる場合は、最初に各サイズを切り上げてから、その後に平均を取ります。例えば、半数のレコードが 100 バイトで、残りの半数が 300 バイトの場合、前者のサイズを 256 に、後者のサイズを 512 に切り上げてから、それらの平均を取ります。切り上げて平均した結果の項目のサイズは 384 になります。この値は、項目の平均サイズ 200 を使用して 256 に切り上げるより正確です。

最大で 63 バイトのレコード長を持つ競合モデルのデータ・テーブル内のレコードは、特殊なケースです。これらのレコードは、エントリーの付属領域に保管されるため、レコード長の平均値を求める目的においては、このデータ長は事実上 0 とみなされます。

### 合計レコード数

すべてのデータ・テーブルに格納されるレコード数の合計。

### ターゲット使用率 (%)

求められた合計数の項目が使用すると想定される、構造のスペースのパーセンテージ。1 から 100 の範囲内で数値を指定します。デフォルトは 75 です。この値は、次のことを保証します。

- ・一時的な拡張用のフリー・スペースが存在する。
- ・初期フリー・スペースでは足りない場合、警告メッセージに応答して構造を拡張する時点が存在する (通常 80% で開始されます)。
- ・エントリーとエレメントの比率を変更するアクティビティが減る。

### 最大拡張率 (%)

構造を拡張できるパーセンテージ。ゼロ以外の値を指定すると、構造の最大サイズは、初期構造サイズにこのパーセンテージ分だけデータの合計量を加算したサイズに増えます。例えば、値 200 を指定すると、指定された合計項目数を初期サイズで十分保管できる場合、最大サイズではこの 3 倍の項目数を十分保管できます。

これらの計算の結果を使用して、カップリング・ファシリティー・リソース・マネージャー (CFRM) の **INITSIZE** パラメーターおよび **SIZE** パラメーターを CFRM ポリシーで設定します。

カップリング・ファシリティー・データ・テーブル (CFDT) サーバーにおける、構造が満杯の状態を防ぐための CICS 予約スペース・パラメーターについては、195 ページの『予約スペースのパラメーター』および 196 ページの『ストラクチャー満杯状態の回避』を参照してください。

## カップリング・ファシリティー・データ・テーブル・サーバー領域の定義と開始

カップリング・ファシリティー・データ・テーブル・プールのカップリング・ファシリティー・データ・テーブル・サーバー領域を開始して、MVS イメージ内でそのプールをアクティブ化します。

### このタスクについて

サーバーは開始タスク、開始ジョブ、またはバッチ・ジョブとして始動できます。このジョブまたはタスクは、APF 許可ライブラリーからカップリング・ファシリティー・データ・テーブル・サーバー領域プログラム DFHCFMN を起動する必要があります。DFHCFMN は、CICS 許可ライブラリー CICSTS56.CICS.SDFHAUTH 内にあります。

### 手順

1. JCL に定義された SYSIN データ・セット内、または EXEC ステートメントの **PARM** パラメーターに DFHCFMN プログラムを指定します。
2. DFHCFMN プログラムの必須およびオプションの始動パラメーターを指定します。  
始動パラメーターを SYSIN データ・セットと **PARM** パラメーターの両方に指定した場合、**PARM** 値により SYSIN 値がオーバーライドされます。これは、**MVS START** コマンドにより **PARM** 値をオーバーライドできるためです。
  - a) 印刷ファイルに SYSPRINT DD ステートメントを指定する必要があります。
  - b) サーバー・パラメーターに SYSIN DD ステートメントを指定する必要があります。
  - c) TS プール名を指定する必要があります。
  - d) ライセンスのアクティベーション・データ・セット (SDFHLIC ライブラリー) を STEPLIB DD ステートメントに連結する必要があります。

e) **REGION** パラメーターを指定することをお勧めします。

このパラメーターにより、カップリング・ファシリティ・データ・テーブル・サーバー領域に、同時実行可能な最大数のデータ・テーブル要求を処理するための十分なストレージが確保されます。

f) **TIME=NOLIMIT** を指定することをお勧めします。

サーバー・タスクは、ほとんどの通常処理の間、待機状態のままです。これは、サーバー処理がクライアント CICS 領域の TCB で実行されるためです。このパラメーターを省略すると、SYS1.PARMLIB の SMFPRMxx メンバーに指定した JWT 値によっては、サーバー・ジョブが異常終了 S522 (待機時間制限の超過) で失敗することがあります。

g) 必要に応じてパラメーターを追加指定します。

例えば、プールで対応可能なキューの最大数や、サーバーが割り振るバッファの数を制御することができます。

## タスクの結果

**ヒント:** MVS イメージ全体ですべてのプール関連パラメーターの一貫性を確保する最も簡単な方法は、同じプールにアクセスするすべてのサーバーに対して同じ **SYSIN** パラメーター・データ・セットまたはその同一のコピーを使用すること、およびサーバー間で異なるあらゆるパラメーターを **PARM** フィールドに指定することです。

## カップリング・ファシリティ・データ・テーブル・サーバーの JCL の例

//PRODCFD1 JOB	...	
//CFSERVER EXEC	PGM=DFHCFMN,REGION=40M,TIME=NOLIMIT	CICS CFDT Server
//STEPLIB DD	DSN=CICSTS56.CICS.SDFHAUTH,DISP=SHR	Authorized library
// DD	DSN=CICSTS56.CICS.SDFHLIC,DISP=SHR	License activation data set
//SYSPRINT DD	SYSOUT=*	Messages and statistics
//SYSIN DD	*	
PPOOLNAME=PRODCFD1		Pool name
MAXTABLES=100		Allow up to 100 tables
/*		

図 37. CFDT サーバー・アドレス・スペースを開始する JCL の例

## カップリング・ファシリティ・データ・テーブル・サーバーのパラメーター

パラメーターは **KEYWORD=value** の形式で指定されます。読みやすくするために任意でキーワードを大/小文字混合で指定できます。

**PARM** フィールドまたは同じ **SYSIN** 入力行で複数のパラメーターを指定する場合、パラメーターはコンマで区切る必要があります。1 つ以上のスペースに続くテキストは、すべて説明コメントと見なされます。アスタリスクまたはスペースで始まるパラメーター行はすべて、行全体がコメントと見なされます。

いくつかのパラメーター・キーワードを複数の形式 (例えば省略形式や切り捨て形式) で入力できます。

主なパラメーターは、始動中にサーバー・プリント・ファイルにリストされます。

これらのパラメーターはすべて (**SYSIN** ファイルまたは **PARM** フィールド) の初期設定パラメーターとして有効で、そのうちのいくつかは **SET** コマンドで変更することもできます。

## REGION パラメーター

**JCL REGION** パラメーターは、カップリング・ファシリティ・データ・テーブルのサーバー領域に、並行して実行できるデータ・テーブル要求の最大数を処理できるだけの十分なストレージがあるようにします。

接続された各 CICS 領域でアクティブにできるカップリング・ファシリティ・データ・テーブル要求の数は、およそ 10 に制限されています。各要求には 40 KB ずつ必要なため、**REGION** サイズには、接続された CICS 領域ごとに 400 KB 以上と、他のストレージ域用におよそ 10% のマージンを指定する必要があります。したがって、最大で 5 つの CICS 領域をサポートするサーバーでは、**REGION=2200K** を指定します。

サーバーの初期化時に、サーバーは **REGION** パラメーターで指定されたとおりに 16 MB より上の使用可能なすべてのストレージを獲得し、そのうち 5% をオペレーティング・システム・サービスが使用するために開放します。また、24 ビットのアドレス可能ストレージを必要とするルーチン (例えば、順次ファイルの読み取りおよび書き込みのルーチン) に使用するために、16MB より下のフリー・ストレージの 5% も獲得します。

初期設定が終了すると、サーバーは AXM ページ割り振りサービスを使用して、そのストレージを管理します。サーバー統計は、どれだけのストレージが割り振られ、16MB より上と 16 MB より下のストレージ領域の中でどれだけ使用されているかを示します。これらは統計で AXMPGANY および AXMPGLOW と呼ばれます。

サーバー領域のタスクまたは仮想記憶間要求がストレージを使い果たした場合、AXM が、異常終了のシミュレーションを使用して、そのタスクまたは要求を強制終了させ、GETMAIN 障害を示すシステム完了コード 80A を出す可能性があります。サーバーは通常、このケースでは他の要求の処理を継続することはできませんが、クリティカルなルーチンでストレージが不足すると、サーバーが強制終了する可能性があります。そのため、このようなリスクを回避するために **REGION** のサイズを十分大きくしてください。

### プール名パラメーター

このパラメーター POOLNAME は必須です。

#### POOLNAME=name

カップリング・ファシリティ・データ・テーブル・プールの 8 文字の名前を指定します。サーバーは、これを接頭部 DFHCF に追加して独自のサーバー名を作成します (例: DFHCF.poolname)。また、接頭部 DFHCFLS\_ を追加してカップリング・ファシリティ・リスト構造の名前を作成します (DFHCFLS\_poolname)。

このパラメーターは、サーバーの初期設定時にのみ有効になります。指定は必須です。

このキーワードは、**POOL** と省略することができます。

### Security parameters

セキュリティ・パラメーターを使用すると、サーバーから提供されるオプションのセキュリティ・メカニズムを使用するかどうかを指定できます。このメカニズムでは、カップリング・ファシリティ・データ・テーブルをオープンする 権限があるかどうかについて CICS 領域を検査できます。また、これらのパラメーターを使用して、このオプション・セキュリティの標準処理をオーバーライドすることもできます。

#### SECURITY={YES|NO}

個々のカップリング・ファシリティ・データ・テーブルのセキュリティ 検査が必要かどうかを指定します。

##### YES

サーバーは、カップリング・ファシリティ・データ・テーブルを開こうとする各 CICS 領域に対してセキュリティ検査を実行します。**SECURITYCLASS** パラメーターで指定した一般リソース・クラスに定義されているプロファイルによってアクセスが制御されます。

この機能には、仮想記憶間モードの FASTAUTH 機能をサポートする、RACF などの外部セキュリティ・マネージャーが必要です。

##### NO

サーバーは、カップリング・ファシリティ・データ・テーブルを開こうとする各 CICS 領域に対してセキュリティ検査を実行しません。

これは、サーバーが実行する唯一のオプションのセキュリティ 検査です。その他のファイル・セキュリティ検査は、必ず、サーバーによって実行されます ([Security for coupling facility data tables](#) を参照)。

このパラメーターは、サーバーの初期化時にのみ有効になります。

このキーワードは、**SEC** と省略することができます。

#### SECURITYCLASS={FCICSFCT|class}

CICS 領域からのカップリング・ファシリティ・データ・テーブル・アクセスのセキュリティ 検査のためにサーバーで使用する RACF 一般リソース・クラスの名前を指定します。この名前は最大 8 文



字です。これは、CFDT リソース・プロファイルおよびそのアクセス・リストが定義されているクラスの名前です。

このパラメーターは、サーバーの初期化時にのみ有効になります。

このキーワードは、SECCLASS と省略することができます。

#### **SECURITYPREFIX={NO|YES}**

SECURITY=YES を指定した場合に、カップリング・ファシリティ・データ・テーブルのセキュリティ検査のために RACF に渡されるリソース名に、サーバー領域のユーザー ID を接頭部として付けるかどうかを指定します。

**注：**このセキュリティ検査で、サーバーによって使用されるリソース名は、CICS ファイル・リソース定義の TABLENAME 属性で指定された名前です。TABLENAME が指定されていない場合は FILE 名になります。

##### **YES**

サーバーは、リソース名にユーザー領域のユーザー ID (デフォルト) を接頭部として付けるか、**SECURITYPREFIX** パラメーターで指定された別の接頭部を付けます。

##### **NO**

サーバーは、接頭部を付けずに RACF に 8 文字のリソース名のみを渡します。

このパラメーターは、サーバーの初期化時にのみ有効になります。

このキーワードは、SECPREFIX または SECPRFX と省略することができます。

#### **SECURITYPREFIXID=identifier**

CICS 領域からのカップリング・ファシリティ・データ・テーブル・アクセスのセキュリティ検査のために、サーバー領域のユーザー ID の代わりにサーバーで使用する接頭部を指定します。接頭部は最大 8 文字です。CFDT のプロファイル名を RACF に定義するために使用された接頭部に対応している必要があります。このパラメーターは、SECURITYPREFIX=YES を指定した場合にのみ有効です。

このパラメーターは、サーバーの初期化時にのみ有効になります。

このキーワードは、SECPREFIXID と省略することができます。

#### **統計パラメーター**

以下のパラメーターを使用して、サーバー統計オプションを指定します。

#### **ENDOFDAY={00:00|hh:mm}**

サーバーが終業時統計を収集してリセットする時刻 (時間と分) を指定します。

**注：**STATSOPTIONS パラメーターが NONE に指定されている場合、サーバーは終業時統計を印刷ファイルに書き込みます。

有効な時間の範囲は 00:00 から 24:00 までです。

このキーワードは、EOD と省略することができます。

#### **STATSINTERVAL={03:00|hh:mm}**

統計収集間隔を 1 分から 24 時間までの間で指定します。このパラメーターは、STATSOPTIONS パラメーターが NONE に指定されている場合には無視されます。

時間間隔の範囲は 00:01 から 24:00 です。

このキーワードは、STATSINT と省略することができます。

#### **STATSOPTIONS={NONE|SMF|PRINT|BOTH}**

サーバーが間隔統計を作成するかどうかと、作成する統計の宛先を指定します。

##### **NONE**

サーバーは間隔統計を作成しません。

##### **SMF**

サーバーは間隔統計を作成し、それらを現行の SMF データ・セットにのみ書き込みます。

##### **PRINT**

サーバーは間隔統計を作成し、それらをサーバーの印刷ファイルにのみ書き込みます。

## BOTH

サーバーは間隔統計を作成し、それらを現行の SMF データ・セットと、サーバーの印刷ファイルに書き込みます。

このキーワードは、STATSOPT と省略することができます。

## 自動リスタート・マネージャー (ARM) のパラメーター

サーバーの初期化中、サーバーは UNLOAD または RELOAD 機能のいずれかを使用してサーバー・プログラムが呼び出された場合を除き、ARM に無条件で登録します。登録が失敗している場合、サーバーは始動しません。

自動リスタート・マネージャーのデフォルトの処理をオーバーライドする場合は、以下のパラメーターを使用します。

### ARMELEMENTNAME=elementname

自動再始動の目的で ARM に対してサーバーを識別するために ARM にポリシーで使用するために、自動リスタート・マネージャーのエレメント名を 16 文字までで指定します。エレメント名に使用できる文字は A から Z、0 から 9、\$ # @ およびアンダースコア記号 ( \_ ) です。

デフォルトの ID は、DFHCFnn\_poolname という形式になります (CF はサーバーのタイプ、nn はシステムの &SYSCONE 値 (1 文字か 2 文字)、poolname はサーバーで対応するプールの名前です)。

このパラメーターは、サーバーの初期化時にのみ有効になります。

このキーワード ARMELEMENT または ARMELEMENTNAME と省略することができます。

### ARMELEMENTTYPE=elementtype

類似のエレメントを分類する手段として ARM ポリシーで使用するために、自動リスタート・マネージャーのエレメント・タイプを 8 文字までで指定します。エレメント・タイプに使用できる文字は A-Z 0-9 \$ # @ および \_ です。

デフォルトのエレメント・タイプは SYSCICSS です。

このパラメーターは、サーバーの初期化時にのみ有効になります。

このキーワードは、ARMELEMENTTYPE と省略することができます。

## リスト構造パラメーター

リスト構造パラメーターに指定する属性は、カップリング・ファシリティ・データ・テーブル (CFDT) プールのためにデータ・テーブルのリスト構造が初めて割り振られるときにのみ使用されます。初期割り振りが実行されるのは、CFDT プールに対するサーバーが初めて開始されたときです。

リスト構造パラメーターは、次のとおりです。

### MAXTABLES={1000|number}

構造を割り振る際に保持するデータ・リストの最大数を指定します。これにより、構造で保管可能なテーブルの最大数が決まります。このパラメーターによって、構造の作成時に定義されるリスト・ヘッダーの数も決まります。

すべてのデータ・テーブルを処理できる十分な大きさの値を指定します。ただし、事前に割り振られたリスト・ヘッダーが使用されずに、カップリング・ファシリティの大量のストレージを消費することがないように注意してください。

構造を再割り振りせずに、この数値を変更することはできません。つまり、既存の構造をまず削除する必要があります (202 ページの『[カップリング・ファシリティ・データ・テーブル・プールを削除または空にする](#)』を参照)。そのため、構造が最大サイズより小さく割り振られている場合は、この値を構造の初期サイズではなく最大サイズに基づいて決定する必要があります。

このパラメーターは、CFSizer ツールの「テーブルの最大数」の値に対応します (186 ページの『[カップリング・ファシリティ・データ・テーブルのストレージの計算](#)』を参照)。

このパラメーターは、サーバーの初期化時にのみ有効になり、構造が最初に割り振られる際にのみ使用されます。有効な範囲は 1 から 999999 までです。

このキーワードは、MAXT と省略することができます。



注: **MAXTABLES** で定義されるデータ・リストに加えて、CICS で内部使用される制御リストも存在します。そのため、リストの合計数は、**MAXTABLES** に指定した数より多くなります。

#### **POOLSIZE={0|number{K|M|G}}**

プール・リスト構造に割り振られるカップリング・ファシリティ・ストレージの初期量を指定します。キロバイト (nK)、メガバイト (nM)、またはギガバイト (nG) で表します。

通常はこのパラメーターは省略可能で、カップリング・ファシリティ・リソース・マネージャー (CFRM) ポリシーの **INITSIZE** パラメーターを使用して構造サイズを指定できます。ただし、このパラメーターは、構造が再割り振りまたは再ロードされたが、CFRM ポリシーが必要なサイズを反映するために更新されていないような場合に便利です。

#### **0**

特殊値 0 は、CFRM ポリシーに指定されたパラメーターを使用して、サーバーが初期割り振りを取得することを意味します。CFRM ポリシーが構造に対して **INITSIZE** 値を指定する場合、これによって初期割り振りが決定します。そうでない場合、CFRM **SIZE** 値 (構造の最大サイズ) が割り振られます。

#### **number**

ゼロ以外の値は、割り振られるストレージの初期量を指定し、CFRM ポリシーの **INITSIZE** パラメーターをオーバーライドします。この値は MVS によってカップリング・ファシリティ・レベル (CFLEVEL) の次のストレージ増分に切り上げられます。例えば、CFLEVEL 16 の場合、値は最も近い 1 MB 単位の値に丸められます。

値は CFRM **SIZE** パラメーターより小さい必要があります。そうでない場合、**POOLSIZE** は無視され、初期割り振りでは CFRM ポリシーに指定されたパラメーターが使用されます。

有効な範囲は 0 から 16777215M です。ただし、z/OS 内の構造の最大サイズより小さい値を指定する必要があります。これを超えると、z/OS のエラーが発生します。例えば z/OS バージョン 1 リリース 12 では、構造の最大サイズは 1048576 MB (1 TB) です。CFRM パラメーターの詳細については、[z/OS MVS シスプレックスのセットアップ](#)を参照してください。

このパラメーターは、サーバーの初期化時にのみ有効になり、構造が最初に割り振られる際にのみ使用されます。

#### **デバッグ・トレース・パラメーター**

これらのパラメーターは、集中的なデバッグ・トレースのためにのみ提供されています。

これらのオプションを実稼働環境で使用すると、パフォーマンスに大きな影響を与え、印刷ファイルが急激に増えてスプール・スペースを使い果たす原因になる場合があります。

仮想記憶間の要求からのトレース・メッセージは、トレース印刷サブタスクで印刷できる速度より速く生成されると、失われる可能性があります。その場合、トレースには、失われたメッセージの数のみが示されます。

#### **CFTRACE={OFF|ON}**

カップリング・ファシリティ・インターフェース・デバッグ・トレース・オプションを指定します。

##### **OFF**

カップリング・ファシリティ・インターフェース・デバッグ・トレースが使用不可です。

##### **ON**

カップリング・ファシリティ・インターフェース・デバッグ・トレースによりトレース・メッセージが印刷ファイルに生成され、カップリング・ファシリティ要求インターフェースの主要パラメーターと、IXLLIST マクロからの結果が示されます。

このキーワードは、TRACECF と指定することもできます。

#### **RQTRACE={OFF|ON}**

テーブル要求デバッグ・トレース・オプションを指定します。

##### **OFF**

テーブル要求デバッグ・トレースが使用不可です。

## ON

テーブル要求デバッグ・トレースによりトレース・メッセージが印刷ファイルに生成され、各仮想記憶間要求のエントリーの主要パラメーターと、出口の結果が示されます。

このキーワードは、TRACERQ= と指定することもできます。

## チューニング・パラメーター

これらのパラメーターはチューニングを行う目的で指定しますが、通常は省略可能で、カップリング・ファシリティ・データ・テーブル (CFDT) サーバーではデフォルト値を使用するように指定できます。

### ELEMENTRATIO={1|number}

構造が最初に割り振られる際の「項目対エレメント」率のエレメントの部分指定する MVS カップリング・ファシリティ・リソース管理 (CFRM) パラメーター。この値は、データ・エレメント用に最初に設定される構造のスペースの比率を決定します。

「項目対エレメント」率の理想的な値は、各項目のデータの平均サイズをエレメントのサイズで割った値です。ただし、実際の項目やエレメントの使用率に応じて、サーバーが自動的に比率を調整します。

このパラメーターは、サーバーの初期化時にのみ有効になり、構造が最初に割り振られる際にのみ使用されます。有効な範囲は 1 から 255 までです。

このキーワードを **ELEMRATIO** として省略することもできます。

カップリング・ファシリティ内でのこのパラメーターについて詳しくは、[z/OS MVS プログラミング: シスプレックス・サービス ガイド](#)を参照してください。

### ELEMENTSIZE={256|size}

リスト構造のデータ・エレメントのサイズを指定する MVS CFRM パラメーターで、2 の累乗である必要があります。このパラメーターは、カップリング・ファシリティのログ・ストリームにのみ使用されます。現在のカップリング・ファシリティの実装では、デフォルトの 256 以外の値を使用する理由はありません。

このパラメーターは、サーバーの初期設定時にのみ有効になり、構造が最初に割り振られる際にのみ使用されます。有効な範囲は 256 から 4096 です。

このキーワードを **ELEMSIZE** として省略することもできます。

### ENTRYRATIO={1|number}

構造が最初に割り振られる際の「項目対エレメント」率の項目の部分指定する MVS CFRM パラメーター。この値は、リスト項目制御用に最初に設定される構造のスペースの比率を決定します。

実際の使用量によっては必要に応じてスペースの使用率を改善するためにサーバーが自動的に比率を調整するため、このパラメーターの指定は必須ではありません。

このパラメーターは、サーバーの初期化時にのみ有効になり、構造が最初に割り振られる際にのみ使用されます。有効な範囲は 1 から 255 までです。

カップリング・ファシリティ内でのこのパラメーターについて詳しくは、[z/OS MVS プログラミング: シスプレックス・サービス ガイド](#)を参照してください。

## ロック待機パラメーター

これらのパラメーターを使用して、サーバーのロック待機再試行メカニズムの時間間隔を変更できます。

システム障害や構造がフルになった状態が発生すると、まれに、ロック解放の通知メッセージが失われ、レコード・ロックを待機している要求が無期限に待機することがあります。これらのパラメーターは、主に、このような状況を防止するための「ウェイクアップ」メカニズムとして用意されています。また、このメカニズムによって、ロックを待機している要求があるのに CICS タスクがパージされた場合に、サーバーで待機している要求は、ロック待機再試行間隔の時間が経過したらすぐにウェイクアップされます。ウェイクアップされた要求プロセスは、CICS タスクがロックを待機していないことを検出します。そのため、ロックが解放されるまでサーバー・リソースを待機し続けることなく、終了します。

必要な時間は、スキャン時間間隔と待機時間の 2 つです。最初の要求が待機状態になると、サーバーはロック・スキャン間隔の時間計測を開始します。

このメカニズムは、通常の処理にはほとんど影響しません。また、デフォルトのロック待機再試行パラメーター値は、大多数のインストール済み環境に適合するように設計されています。

#### **LOCKSCANINTERVAL={5|number}**

ロック待機タイムアウトを検査するために、レコード・ロックを待機している要求をスキャンする時間間隔を指定します。

これはロック待機タイムアウトの全体の時間に影響します。あるスキャン間隔の途中でロックの待機を開始した要求は、そのスキャン間隔の始めから待機を開始したかのようにカウントされるからです。ロック・スキャン間隔はロック待機間隔より短くする必要があるため、ロック待機間隔をロック・スキャン間隔のちょうど倍数にするのが理想的です。

この値は秒数として指定することも、*hh:mm:ss* という時刻形式で指定することもできます。

有効な範囲は、1 (1 秒) から 24:00 (24 時間) までです。

このキーワードは、LOCKSCANINT と省略することができます。

#### **LOCKWAITINTERVAL={10|number}**

要求が再試行を実行するまでレコード・ロックを待機する最大時間を指定します。要求が待機する実際の時間は、スキャン間隔とどの程度の時間差で待機を開始したかによって異なります。例えば、デフォルトのスキャン間隔と待機間隔を使用するサーバーで、要求が待機を開始した時点で既に 4 秒経過していた場合、待機できる最大時間は 6 秒です。サーバーは、この要求のタイムアウト値を検査すると、要求がそのスキャン間隔の間中ずっと待機していたと見なします。したがって、デフォルト値の場合、5 秒から 10 秒の間待機した後、要求の再試行が実行されます。

この値は秒数として指定することも、*hh:mm:ss* という時間間隔形式で指定することもできます。

有効な範囲は、1 (1 秒) から 24:00 (24 時間) までです。

このキーワードは、LOCKWAITINT と省略することができます。

#### **警告パラメーター**

構造が満杯に近づくとき警告メッセージが出され、構造の自動変更が行われようにするしきい値を変更するには、これらのパラメーターを使用します。

#### **ELEMENTWARN={80|number}**

警告メッセージの発行と構造の自動変更が最初に行われる時点の、使用中のリスト構造エレメントのパーセンテージを指定します。

有効な範囲は 1 パーセントから 100 パーセントまでです。

このキーワードは、ELEMWARN と省略することができます。

#### **ELEMENTWARNINC={5|number}**

次の警告メッセージが発行されるまでの使用中のエレメントのパーセンテージの増加 (または減少) (次の増加で 100 に達する場合は 1 に減らす)。最初の警告の後に出されるメッセージは、エレメントの数が増加または減少すると発行されます。これらのメッセージは、使用中のエレメントが、最初の警告レベルよりこのパーセンテージ分以上減ると出されなくなります。

有効な範囲は 1 パーセントから 100 パーセントまでです。

このキーワードは、ELEMWARNINC と省略することができます。

#### **ENTRYWARN={80|number}**

警告メッセージの発行と構造の自動変更アクションが最初に行われる時点の、使用中のリスト構造項目のパーセンテージを指定します。

有効な範囲は 1 パーセントから 100 パーセントまでです。

#### **ENTRYWARNINC={5|number}**

次の警告メッセージが発行されるまでの使用中の項目のパーセンテージの増加 (または減少) (次の増加で 100 に達する場合は 1 に減らす) を指定します。最初の警告の後に出されるメッセージは、エレメントの数が増加または減少すると発行されます。これらのメッセージは、使用中の項目が、最初の警告レベルよりこのパーセンテージ分以上減ると出されなくなります。

有効な範囲は 1 パーセントから 100 パーセントまでです。

### 自動ストラクチャー変更パラメーター

このようなパラメーターを使用して、ストラクチャーがほぼ満杯になった時にサーバーが自動変更を試みる条件を変更できます。

ストラクチャー変更プロセスについては、[197 ページの『カップリング・ファシリティ・データ・テーブル・サーバーの自動ストラクチャー変更』](#)を参照してください。

#### **ALTERELEMMIN={100|number}**

超過エレメントの最小数を指定します。その数に達すると、サーバーが自動変更を実行して超過エレメントを項目に変換します。

有効な範囲は 0 から 9999999999 までです。

#### **ALTERELEMPC={5|number}**

超過エレメントの最小パーセント値を指定します。その値を超えると、サーバーが自動変更を実行して項目の比率を上げます。

有効な範囲は 0 パーセントから 100 パーセントまでです。

#### **ALTERENTRYMIN={100|number}**

超過項目の最小数を指定します。その数に達すると、サーバーが自動変更を実行して超過項目をエレメントに変換します。

有効な範囲は 0 から 9999999999 までです。

#### **ALTERENTRYPC={5|number}**

超過項目の最小パーセント値を指定します。その値を超えると、サーバーが自動変更を実行してエレメントの比率を上げます。

有効な範囲は 0 パーセントから 100 パーセントまでです。

#### **ALTERMININTERVAL={00:10|hh:mm}**

ストラクチャーがほぼ満杯になった時 (エレメントまたは項目の警告レベルを上回った時) の自動ストラクチャー変更の最小時間間隔を指定します。ストラクチャーがほぼ満杯になった時にスペースを最適化しようとする変更処理の回数が過剰にならないようにするための設定です。

有効な範囲は 00:00 から 24:00 までです。

このキーワードは、ALTERMININT と省略することができます。

### 予約スペースのパラメーター

これらのパラメーターを使用して、再書き込みおよびサーバーの内部処理 (作業単位の追跡、ロック解除時の他のサーバーへの通知など) のために予約される構造スペースの量を制御できます。

これらのパラメーターの影響については、[196 ページの『ストラクチャー満杯状態の回避』](#)を参照してください。

#### **ELEMENTRESERVEMIN={300|number}**

再書き込みおよびサーバーの内部処理のために予約するリスト構造データ・エレメントの最小数 (通常は 1 つあたり 256 バイト) を指定します。

有効な範囲は 0 から 9999999999 までです。

このキーワードは、ELEMRESERVEMIN と省略することができます。

#### **ELEMENTRESERVEPC={5|number}**

再書き込みおよびサーバーの内部使用のために予約するリスト構造データ・エレメントのパーセンテージを指定します。このパーセンテージが ELEMENTRESERVEMIN パラメーターに指定した最小数より少ない数に相当する場合は、最小数が使用されます。

有効な範囲は 0 から 100 までです。

このキーワードは、ELEMRESERVEPC または ELEMRESPC と省略することができます。

**ENTRYRESERVEMIN={100|number}**

再書き込みおよびサーバーの内部処理のために予約するリスト構造エントリーの最小数を指定します。

有効な範囲は 0 から 999999999 までです。

このキーワードは、ENTRYRESMIN と省略することができます。

**ENTRYRESERVEPC={5|number}**

再書き込みおよびサーバーの内部使用のために予約するリスト構造エレメントのパーセンテージを指定します。このパーセンテージが ENTRYRESERVEMIN パラメーターに指定した最小数より少ない数に相当する場合は、最小数が使用されます。

有効な範囲は 0 から 100 までです。

このキーワードは、ENTRYRESPC と省略することができます。

**ストラクチャー満杯状態の回避**

カップリング・ファシリティのデータ・テーブル・ストラクチャーが完全に満杯になることが許可されている場合は、新しいレコードやテーブルを追加できなくなるだけでなく、パフォーマンスとアプリケーションの機能に大きな影響が及ぶ可能性があります。

特に、新しいデータのサイズが元のデータ以下の場合でも再書き込み要求が拒否されたり、サーバーの内部操作が失敗したり、内部でタイムアウトと再試行が発生したりする可能性があります。

一定数の項目やエレメントを予約することでストラクチャーが完全に満杯になってしまうリスクを小さくするためのパラメーターとして、ELEMENTRESERVEMIN、ELEMENTRESERVEPC、ENTRYRESERVEMIN、ENTRYRESERVEPC が用意されています。ただし、これらのパラメーターを使用できるのは、基本的には余分のスペースが一時的に必要なだけの操作 (再書き込みや作業単位制御操作) に限られます。カップリング・ファシリティの各アクセス要求で返されてストラクチャーに残っている項目やエレメントの数が指定の予約レベル以下の場合に、新しいレコードの書き込み要求や新しいテーブルの作成要求をサーバーに送ると、その要求が拒否され、使用可能なスペースがないという趣旨のメッセージが生成されます。サーバーは、要求を拒否する前に、ダミーの読み取り要求を送信してストラクチャーの最新の使用状況レベルを確認し、さらに多くのスペースが直前に使用可能になっていないかどうかを確認めます。

予約スペース・パラメーターを使用すると、使用可能なスペースをかなり消費するテーブルをロードする操作などでストラクチャーが急速に満杯になっていく場合でも、既存のレコードの再書き込みを実行したりサーバー間の内部通信を正常に継続したりするために必要なスペースを十分に残しておかなければならなくなります。

このメカニズムを使用しても、ストラクチャーがいずれ完全に満杯になってしまう事態を回避することはできません。リカバリー可能な再書き込みでは予約スペースが一時的に使用され、データ長を増やす再書き込みでは予約エレメントが徐々に使用されていくからです。ストラクチャーが完全に満杯になる事態を回避するための処置を講じなければ、以下のような影響を受ける可能性があります。

- テーブルを閉じたりテーブルの状況を変更したりする操作を試みた時に、ストラクチャーが一時的に満杯になっているという状態になることがあります。その場合は、無限に再試行が続きます。テーブルの整合性を維持するには、その操作を完了することが必要だからです (唯一の代替処置はサーバーを強制終了することです)。通常、再試行プロセスはすぐに成功しますが、理論的には、別のサーバーが使用不能なりソースを一時的に解放するまでループが続く可能性があります。
- 競合更新モデルを使用してテーブルの再書き込みを実行する時に、データのサイズが元のデータと同じかそれより小さかったとしても、ストラクチャーの満杯状態が原因で最初の実行が失敗すると、再試行が無限に続きます。これは、アプリケーションがこの予期しない形式の障害を処理しなくてもよくするための動作です。この場合も、通常は、再試行プロセスがすぐに成功しますが、理論的には、しばらくのあいだループが続く可能性があります。
- ロック・モデルやリカバリー可能更新モデルを使用してテーブルの再書き込みを実行しようとする、データのサイズが増えなくても、ストラクチャー満杯状態で要求が拒否されることがあります。この場合、再試行はありません。
- サーバーがコミット処理のための作業単位制御項目を作成できないために、作業単位がバックアウトされることがあります。



- ロック解除メッセージを送信するための十分なストラクチャー・スペースがない場合があります。その場合は、ロック解除が検出されない限り、待ち状態のタスクがすぐにウェイクアップしないで、LOCKWAITINTERVAL パラメーターで指定されているタイムアウト間隔まで待機を続けます。

### カップリング・ファシリティ・データ・テーブル・サーバーの自動ストラクチャー変更

カップリング・ファシリティ・データ・テーブル・サーバーは、要求ごとにカップリング・ファシリティから返される情報に基づいて、ストラクチャー内で使用中になっているエレメントと項目の総数をモニターします。

使用中の数が指定の警告しきい値を超えると、サーバーから警告メッセージが出されます。その警告メッセージは、使用中の数がさらに増えて別のしきい値を超えるたびに繰り返されます。

サーバーから警告が出されるたびに、項目とエレメントの比率に関する自動ストラクチャー変更が必要かどうかを検査されます。いずれかの形の変更がすでに (サーバー自体またはオペレーター・コマンド SETXCF ALTER によって) 最近実行されたのに、ストラクチャーのスペース使用率がそれ以降も警告レベルを上回ったままであれば、ALTERMININTERVAL パラメーターで指定されている間隔の最小時間が経過するまでは、ストラクチャー変更が抑止されます。

自動ストラクチャーが必要かどうかサーバーによって検査される時には、現在使用中のエレメントと項目の数の比率に基づいて、過剰エレメントと過剰項目のいずれかが完全に使い果たされた時にもう一方がどれほど残っているかが計算されます。過剰エレメントまたは過剰項目の数が ALTERELEMMIN パラメーターまたは ALTERENTRYMIN パラメーターの指定数を超えていて、その同じ数を合計に対する比率で表した値が ALTERELEMPC パラメーターまたは ALTERENTRYPC パラメーターの指定値を超えている場合は、項目とエレメントの比率を現在使用中の項目とエレメントの数の実際の比率に変更するための IXLALTER 要求が実行されます。

1 つのストラクチャーで一度にアクティブにできる変更要求は 1 つだけです。したがって、別のサーバーがすでにストラクチャー変更プロセスを開始していた場合は、変更要求が拒否されます。ただし、そのストラクチャー変更が完了すると、システムによってすべてのサーバーに完了が自動的に通知され、エレメントと項目の新しい数が知られるので、それぞれのサーバーで状況情報を更新できるようになります。

## カップリング・ファシリティ・データ・テーブル・サーバー領域の制御

カップリング・ファシリティ・データ・テーブル・サーバーを制御するためのコマンドを実行できます。そのためには、**MVS MODIFY (F)** コマンドを使用し、サーバー領域のジョブ名または開始タスク名を指定し、その後にサーバー・コマンドを入力します。

### このタスクについて

簡易書式の F を使用した場合のカップリング・ファシリティ・データ・テーブル・サーバー・コマンドの一般的な形式は、以下のようになります。

```
F job_name,command parameters... comments
```

カップリング・ファシリティ・データ・テーブル・サーバー領域を制御するためのコマンドを以下にまとめます。

### 手順

- サーバー初期設定パラメーターを変更する場合は、**SET** コマンドを使用します。

```
SET keyword=operand[,keyword=operand,...]
```

**MVS SET** コマンドと同じく、このコマンドは **T** という省略形にすることもできます。詳しくは、[198 ページの『SET コマンドのオプション』](#)を参照してください。

- 1 つ以上のパラメーター値や統計の要約情報をコンソールに表示する場合は、**DISPLAY** コマンドを使用します。

```
DISPLAY keyword[=operand][,keyword[=operand],...]
```

**DISPLAY** の有効なキーワードとして、すべての初期設定パラメーターと、[199 ページの『DISPLAY および PRINT コマンド・オプション』](#)に記されている一連のパラメーターを使用できます。



**MVS DISPLAY** コマンドと同じく、このコマンドは **D** という省略形にすることもできます。

- **DISPLAY** コマンドで生成する情報をファイルに出力する場合は、**PRINT** コマンドを使用します。

```
PRINT keyword[=operand][,keyword[=operand,...]]
```

このコマンドでは、**DISPLAY** と同じキーワードを使用して、同じ出力を生成することができますが、出力先は印刷ファイルのみになります。

- テーブルを削除する場合は、**DELETE TABLE=name** コマンドを使用します。  
このコマンドを実行する時点でテーブルが使用中になっていると、成功しません。**DEL** という省略形のコマンドを使用することもできます。
- サーバーを正常に停止する場合は、**STOP** コマンドか **MVS STOP** コマンドを使用します。  
サーバーは、アクティブな接続がまず終了するのを待ちながら、新しい接続が確立されないようにします。**P** という省略形のコマンドを使用することもできます。例えば、**P jobname** のように入力します。
- 自動リスタートを要求する場合は、**CANCEL** コマンドを使用します。  
このコマンドを実行すると、サーバーがすぐに終了します。このコマンドで自動リスタートを実行するかどうかを指定できます。**CANCEL RESTART** については、[178 ページの『CANCEL コマンドのオプション』](#)を参照してください。
- サーバーは、構造サイズを変更するオペレーター・コマンド **SETXCF** のような XES イベントにも応答します。  
サーバーは、カップリング・ファシリティにアクセスできなくなると、サーバー **CANCEL** コマンドを自動的に発行して、すぐにサーバー自体をクローズします。

### SET コマンドのオプション

**SET** コマンドを使用して、サーバー初期設定パラメーターのグループを変更できます。

それらのシステム初期設定パラメーターのグループを次に示します。

- 統計パラメーター
- デバッグ・トレース・パラメーター
- ロック待機パラメーター
- 警告パラメーター
- 自動 ALTER パラメーター

以下の **SET** キーワードを使用すると、最後に終了したときに未解決の作業単位が存在した非アクティブな CICS 領域のサーバーのリカバリー状況を変更できます。

#### **RESTARTED=applid**

指定した **APPLID** のために一時的なリカバリー可能接続を確立します。これにより、領域が最後に終了されたときにコミット処理中またはバックアウト処理中だった作業単位が解決されます。また、未確定の作業単位が残っているかどうかを示されます。

このキーワード **RESTART** または **REST** と省略することができます。

#### **COMMITTED={applid|applid.uowid}**

指定した **APPLID** のために一時的なリカバリー可能接続を確立し、すべての未確定の作業単位をコミットします。あるいは、**uowid** も指定した場合は、その指定した作業単位をコミットします。

同じ作業単位で更新されたカップリング・ファシリティ・データ・テーブル・リソースと他の CICS リソースの間で不整合が生じる可能性があるため、このコマンドは、元の CICS 領域を再始動して作業を正常に解決することが不可能な場合にのみ使用してください。

このキーワード **COMMIT** または **COMM** と省略することができます。

#### **BACKEDOUT={applid|applid.uowid}**

指定した **APPLID** のために一時的なリカバリー可能接続を確立し、すべての未確定の作業単位をバックアウトします。あるいは、**uowid** も指定した場合は、その指定した作業単位をバックアウトします。

同じ作業単位で更新されたカップリング・ファシリティ・データ・テーブル・リソースと他の CICS リソースの間で不整合が生じる可能性があるため、このコマンドは、元の CICS 領域を再始動して作業を正常に解決することが不可能な場合にのみ使用してください。

このキーワード BACKOUT または BACK と省略することができます。

以下の SET パラメーターを使用すると、特定のテーブルに関連するオプションを変更できます。

#### **TABLE=name**

同じコマンドで、以下のテーブル関連パラメーターを適用するテーブルを指定します。このパラメーターは、テーブル関連パラメーターの前に指定する必要があります。

#### **MAXRECS=number**

この前の TABLE パラメーターで指定したテーブルに保管できるレコードの最大数を変更します。

現在のテーブルのレコード数より少ない値を最大数に設定した場合は、その新しい上限を下回るまでレコードが削除されないと、新しいレコードは保管できません。リカバリー可能テーブルの場合は、これはレコードの更新もできないことを意味します。リカバリー可能な更新プロセスでは、書き直し処理で新しいレコードが追加され、トランザクションが完了すると元のレコードが削除されるからです。

このキーワードは、MAXNUMRECS としても指定できます。

#### **AVAILABLE={YES|NO}**

この前の TABLE パラメーターで指定したテーブルを新しい OPEN 要求で使えるかどうかを指定します。テーブルを使用不可にすると、それより後にそのテーブルに対して OPEN 要求を発行した CICS 領域は、テーブルが使用できないことを示す応答を受け取ります。ただし、現在テーブルを開いている領域への影響はありません。テーブルに使用不可のマークが付いている場合であっても、サーバーは、再始動処理時にリカバリー可能な処理を解決するために CICS 領域の代わりに暗黙的にテーブルを開くことができます。

このキーワードは、AVAIL と省略することができます。

**SET コマンドの例:** 次の例は、統計オプションを変更します。

```
SET STATSOPT=BOTH,EOD=21:00,STATSINT=06:00
```

次の例は、指定したテーブルで許可されているレコードの最大数を変更します。

```
SET TABLE=PAYECFT1,MAXRECS=200000
```

#### **DISPLAY および PRINT コマンド・オプション**

DISPLAY (および PRINT) コマンドを使用して、初期設定パラメーターの値および追加情報を表示できます。

追加情報を提供するパラメーターの一部は、総称名をサポートしています。総称名を指定するには、以下のワイルドカード文字を使用します。

- \* (アスタリスク記号)。これはパラメーター値の任意の場所で使用でき、任意の値の 0 個から 8 個までの文字を表します。例えば、CICSH\* は、文字 H で識別される CICSplex 内のすべての CICS APPLID を表します。
- % (パーセント記号)。これはパラメーター値の任意の場所で使用でき、任意の値の 1 文字のみを表します。例えば、CICS%T\* は、すべての CICSplex 内のすべての TOR APPLID を表します。

DISPLAY コマンドおよび PRINT コマンドでサポートされるパラメーターは、次のとおりです。

#### **APPLIDS**

プールへのリカバリー可能接続を現在保持しているすべての CICS 領域の APPLID および MVS システム名を表示します。このコマンドは、MODIFY コマンドの発行先のサーバーに関する情報だけでなく、同じプールに接続されている他のすべてのサーバーに関する情報も返します。

このキーワードは、APPLID、APPLS、または APPL と省略することができます。

#### **APPLID={applid|generic}**

サーバーのプールへのリカバリー可能接続を現在保持し、その APPLID が *applid* または *generic* と一致する各領域の APPLID および MVS システム名を表示します。このコマンドは、MODIFY コマンドの発

行先のサーバーに関する情報だけでなく、同じプールに接続されている他のすべてのサーバーに関する情報も返します。

#### ***applid***

シスプレックス内の 1 つの領域にのみ一致する特定の APPLID に対してこの値を使用します。

#### ***generic***

複数の領域に関する情報を取得する場合は、適切な総称値を使用します。

*applid* も *generic* も指定されていない場合、サーバーではこれをコマンド DISPLAY APPLIDS と同等に扱います。

このキーワードは、APPLIDS、APPLS、または APPL と指定することもできます。

### **ARMREGISTERED**

ARM 登録が正常に完了したかどうかを示します (YES または NO)。

### **CONNECTIONS**

コマンドの発行先のサーバーに現在接続されている領域のジョブ名およびアプリケーション ID を表示します。

このキーワードは、CONN と省略することができます。

### **TABLES**

プールに現在割り振られているすべてのテーブルの名前を表示します。

#### **TABLE={name|generic\_name}**

特定のテーブル、または名前が総称名に一致するテーブルのセットの属性と状況に関する情報を表示します。

テーブル名が指定されていない場合、これは DISPLAY TABLES と同等に扱われます。

### **TABLEUSERS**

プールに現在定義されている各テーブルを使用中の領域の CICS APPLID を表示します。

このキーワードは、TABLEU と省略することができます。

#### **TABLEUSERS={name|generic\_name}**

指定されたテーブル、または名前が総称名と一致するテーブルの各セットを現在使用している領域の CICS APPLID を表示します。

テーブル名が指定されていない場合、これは DISPLAY TABLEUSERS と同等に扱われます。

このキーワードは、TABLEU と省略することができます。

### **UOWIDS**

未解決のリカバリー可能作業単位を現在保持しているすべての領域のアプリケーション ID と、現在未確定か、コミットまたはバックアウトの処理中である作業単位の数を表示します。表示される情報には、解決プロセスをまだ開始していない作業単位、つまり、依然として未完了の作業単位は含まれていません。

このキーワードは、UOWS と省略することができます。

#### **UOWIDS={applid|generic\_applid}v{applid.\*|generic\_applid.\*}**

指定された領域が未解決のリカバリー可能作業単位を現在保持している場合は、その領域のそれらの作業単位についての情報が表示されます。表示される情報には、解決プロセスをまだ開始していない作業単位、つまり、依然として未完了の作業単位は含まれていません。返される情報は、使用されるオペランドの形式によって異なります。

#### ***applid|generic\_applid***

この形式のオペランドは、現在未確定か、コミットまたはバックアウトの処理中である作業単位の数を表示します。

*applid* を指定すると、サーバーは、シスプレックス内の 1 つの領域にのみ対応する特定の APPLID についての UOW 情報を表示します。

*generic\_applid* を指定すると、サーバーは、指定された汎用 APPLID に一致するすべての APPLID についての UOW 情報を表示します。

### ***applid.\*|generic\_applid.\****

この形式のオペランドは、以下を表示します。

- 個々の作業単位の状態およびローカル UOWID。これに続いて、
- 現在未確定か、コミットまたはバックアウトの処理中である作業単位の数 요약。

*applid.\** を指定すると、サーバーは、シスプレックス内の 1 つの領域にのみ対応する特定の APPLID についての UOW 情報を表示します。

*generic\_applid.\** を指定すると、サーバーは、指定された汎用 APPLID に一致するすべての APPLID についての UOW 情報を表示します。

このキーワードは、UOWS と省略することができます。

### **UOWID=*applid.uowid***

アプリケーション ID およびローカル作業単位 ID (UOWID) で識別される、未解決の個々の作業単位の状態を表示します。16 桁の 16 進数字としてローカル UOWID を入力します。

このキーワードは、UOW と省略することができます。

### **統計の 요약の *DISPLAY* オプションおよび *PRINT* オプション**

以下のパラメーターを使用して、統計を表示または印刷します。

#### **CFSTATS**

カップリング・ファシリティ・インターフェースのアクセスやサーバーからの 応答の統計を表示します。

このキーワードは、**CFST** または **STATSCF** と指定することもできます。

#### **POOLSTATS**

プール・リスト構造全体の使用状況の統計を表示します。この統計は、カップリング・ファシリティ・アクセス要求によって戻される情報に基づいています。そのため、コマンドの発行先のサーバーを介して実行された最新の要求の時点の統計に過ぎません。

このキーワードは、**POOLST** と省略することができます。

#### **TABLESTATS**

コマンドの発行先のサーバーによって処理される、各テーブルの要求の統計と、作業単位制御などのテーブル固有ではない要求も含む、処理されたすべての要求の 요약が表示されます。

現在の統計インターバルの開始以降の、要求数がゼロ以外の表のみが表示されることにご注意ください。

このキーワードは、**TABLEST** として指定することもできます。

### **TABLESTATS={*name|generic\_name*}**

指定したテーブルの要求の統計を表示します。

#### ***name***

サーバーによってアクセスされたプール内の固有のテーブル名。このテーブルの統計のみを戻します。

#### ***generic\_name***

複数のテーブルに関する統計を取得するために使用できる総称名。総称名と一致するすべてのテーブル名の統計を戻します。

このキーワードは、**TABLEST** と省略することができます。

#### **STORAGESTATS**

サーバー・アドレス・スペースの主記憶域割り振りの統計を表示します。

このキーワード **STORAGEST** または **STGST** と省略することができます。

## 情報の結合リストでの **DISPLAY** および **PRINT** オプション

以下のキーワードは、情報の結合リストを表します。

### PARAMETERS

主要なパラメーター値を表示します。それらは、POOLNAME、SECURITY、SECURITYPREFIX、統計オプションとリスト構造オプションです。

このキーワードは、PARM または PARMS と省略することができます。

### ALLPARAMETERS

すべてのパラメーター値を表示します。

このキーワードは、ALLPARMS と省略することができます。

### STATISTICS

使用可能なすべての統計を表示します。

このキーワードは **STAT** または **STATS** という省略形にすることができます。

### INITIALIZED

初期設定が完了すると通常表示されるパラメーターおよび統計を表示します。これは、PARM、POOLSTATS、STGSTATS と同等です。

このキーワードは、INIT と省略することができます。

### ARM

すべての ARM 関連のパラメーター値を表示します。

- ARMELEMENTNAME
- ARMELEMENTTYPE
- ARMREGISTERED

このキーワードは、ARMSTATUS としてコーディングできます。

### CANCEL コマンドのオプション

CANCEL コマンドを使用して、自動再始動を要求できます。

以下のパラメーターを指定します。

#### RESTART={NOvYES}

自動再始動を要求するかどうかを指定して、サーバーをすぐに強制終了します。デフォルト値は RESTART=NO です。

サーバーでカップリング・ファシリティの接続に関するリカバリー不能な問題が起きて、接続が失われたり、ストラクチャーで障害が発生したりすると、サーバーは、CANCEL RESTART=YES コマンドを使用して自身をキャンセルします。その結果、既存の接続が終了して、サーバーがシャットダウンします。その後、サーバー・ジョブの新しいインスタンスが開始されます。

サーバー・コマンド CANCEL RESTART=YES または MVS コマンド CANCEL jobname,ARMRESTART を使用して、サーバーを明示的に再始動することもできます。

RESTART=YES の代わりに RESTART、RESTART=NO の代わりに NORESTART を直接入力することも可能です。

## カップリング・ファシリティ・データ・テーブル・プールを削除または空にする

カップリング・ファシリティ・リスト構造を削除する MVS SETXCF コマンドを使用して、カップリング・ファシリティ・データ・テーブル・プールを削除できます。

### このタスクについて

例えば、次のようになります。

```
SETXCF FORCE,STRUCTURE,STRNAME=DFHCFLS_poolname
```

プールに接続されているサーバーがない場合にのみ、構造を削除できます。そうでない場合、MVS はコマンドを拒否します。

削除されたプールに対してサーバーを始動(または、プールを再ロード)しようとする、その構造は新しい構造として割り振られます。新規に割り振られた構造は、現在アクティブな CFRM ポリシーにより指定されたサイズ属性と位置属性、およびサーバーの初期設定パラメーター (特に、MAXTABLES) で決定されたその他の値を使用します。

## カップリング・ファシリティ・データ・テーブル・プールのアンロードと再ロード

**FUNCTION** パラメーターに UNLOAD および RELOAD のオプションを指定して、カップリング・ファシリティ・データ・テーブル・プールの コンテンツ全体を順次データ・セットとの間でアンロードしたり、再ロードしたりすることができます。

### このタスクについて

この機能を使用して、次のようなことができます。

- 計画されたカップリング・ファシリティのメンテナンス時にカップリング・ファシリティ・データ・テーブル・プールを保存する
- プールを別のカップリング・ファシリティに移動する
- プールのリスト構造のサイズを増やす

元のプールで指定されたテーブルの数の最大数が小さすぎる、あるいはプールが最大サイズに達したため拡張する必要がある場合、プールをアンロードし、構造を削除することで、再ロードのプロセスでスペースを大きくして再割り振りできるようにします。

あるいは、システム管理再作成機能を使用して、同じシステムに接続された別のカップリング・ファシリティに構造を動的に移動する方法もあります。これにより、再作成の進行中、サーバーは一時的に処理を中断するものの、アクティブの状態を保つことができます。詳しくは、[237 ページの『システム管理リスト構造の再作成』](#)を参照してください。

### FUNCTION={UNLOAD|RELOAD}

サーバーの初期設定を行う対象の機能を指定します。

#### UNLOAD

順次データ・セットに対して POOLNAME パラメーターで指定されているカップリング・ファシリティ・データ・テーブル・プールのコンテンツ全体をアンロードします。アンロードの処理が完了すると (正常終了または異常終了)、サーバー・プログラムは終了します。

UNLOAD 機能には、テーブル・プールをアンロードする順次データ・セットを説明する DDNAME DFHCFUL の DD ステートメントが必要です。アンロード・データ・セットの形式は、次のとおりです。

```
RECFM=F  
LRECL=4096  
BLKSIZE=4096
```

サーバーによって作成されるプール使用量の統計から、データ・セットの合計サイズの上限 (バイト数) を見積もることができます。

- 統計を使用して、使用中の要素の数に要素のサイズ (通常は 256) を乗算してデータ・サイズの合計バイト数を算出します。データ・要素内の未使用のスペースはアンロードされないため、通常、データのアンロードに必要なスペースはこれよりかなり少量になります。
- レコード・キー用のスペースを追加します。このスペースは、2 バイトの接頭部と、各レコードのキーの長さ、テーブル制御情報用にテーブルごとに 100 バイト足したものです。したがって、キーと制御情報に必要な最大数は、次のようになります。

```
(18 bytes x number of entries) + (100 bytes x number of tables)
```



## RELOAD

直前にアンロードしたカップリング・ファシリティ・データ・テーブル・プールを、POOLNAME パラメーターで指定されたカップリング・ファシリティ・データ・テーブル・プールに再ロードします。

プールを、別の名前のプールに再ロードすることができます。元のプールと同じ名前である必要はありません。再ロードの処理が完了すると (正常終了または異常終了)、サーバー・プログラムは終了します。

RELOAD 機能には、テーブル・プールを再ロードする順次データ・セットを説明する DDNAME DFHCFRL の DD ステートメントが必要です。

必要な場合は、再ロード中に構造が割り振られ、その際に通常のサーバー始動で使用するのと同じ、構造属性を制御するサーバー・パラメーターを使用できます。再ロード処理では、すでにプールにあるテーブルや作業単位 (例えば、構造が小さすぎたため、構造サイズを大きくするために ALTER を使用した後に再ロード・ジョブを再始動する必要があった場合など) は迂回します。

**注:** アンロードしたプール構造が、最初の割り振り以降の任意の時点で (SETXCF コマンドを使用してサイズを増加し) 動的に変更された場合、再ロードしたプールに増加されたサイズが割り振られるようにしてください。現行の CFRM ポリシーの構造の INITSIZE パラメーターを、構造のサイズを変更するたびに更新し、更新したポリシーを SETXCF START ,POLICY コマンドを使用してアクティブにする方法が推奨されます。あるいは、再ロード JCL の POOLSIZE パラメーターに、必要なプール・サイズを指定することもできます。ただし、この方法では、CFRM INITSIZE パラメーターがプール・サイズの最大値と完全に一致する場合はオーバーライドされません。

**注:** FUNCTION パラメーターを省略すると、サーバー・プログラムはカップリング・ファシリティ・データ・テーブル・サーバー・アドレス・スペースを初期化します。

UNLOAD および RELOAD 関数では、サーバー・プログラムはリスト構造を排他使用する必要があります。構造が通常のサーバーによって現在使用されている場合、アンロードまたは再ロードの試行は拒否されます。同様に、アンロードまたは再ロードのジョブの進行中に通常のサーバーを始動しようとする、構造への共有アクセスが使用できないため、試行は失敗します。

アンロードまたは再ロード中に、通常のサーバーのすべてのパラメーターを指定することができますが、これらの一部 (セキュリティ関連のパラメーターなど) は、アンロードまたは再ロードの処理には適用されないため、無視されます。

プールが満杯に近い (項目やエレメントの空きが 5% 未満である) 場合、まったく同じサイズの構造でアンロードや再ロードができる保証はありません。これは、使用可能なスペースは現在の「項目対エレメント」率に影響され、自動 ALTER プロセスによっておおまかにしか制御されていないためです。再ロード中に構造が警告レベルに達した場合、自動 ALTER プロセスは「項目対エレメント」率の調整を試行します。ALTER の進行中に再ロードのスペースが足りなくなると、再ロード・プロセスは ALTER が完了するまで自動で待機します。

スペース不足で再ロードが失敗した場合、結果として表示されるメッセージには再ロードされたテーブルの数と、失敗した時点までに読み取られたブロックの数が含まれます。これらの値を、元のアンロード・ジョブのメッセージと比較して、未ロードのテーブルとデータの数を出すことができます。スペース不足になる前にテーブルが部分的に再ロードされた場合、それは削除されるので、再ロードを後に再試行する際にテーブル全体が再ロードされます。スペース不足以外の理由 (例えば MVS システム障害など) で再ロードが中断された場合、部分的に再ロードされた構造を使用して再ロードを引き続き再始動することはできません。そのため、構造を削除して (MVS SETXCF FORCE コマンドを使用)、新しく割り振られた構造で再ロードを再試行する方法を推奨します。

```
//UNLDCFD1 JOB    ...
//DTUNLOAD EXEC  PGM=DFHCFMN          CICS CF data table server program
//STEPLIB DD     DSN=CICSTS56.CICS.SDFHAUTH,DISP=SHR  Authorized library
//SYSPRINT DD    SYSOUT=*              Options, messages and statistics
//DFHCFUL DD     DSN=CFDT1.UNLOADED.POOL,  Unloaded data table pool
//              DISP=(NEW,CATLG),
//              SPACE=(4096,(10000,1000))  Estimated size in 4K blocks
//SYSIN DD      *
FUNCTION=UNLOAD          Function to be performed is UNLOAD
POOLNAME=PRODCFD1       Pool name
/*
```

図 38. アンロード JCL の例

```
//RELD1CFD1 JOB    ...
//DTRELOAD EXEC  PGM=DFHCFMN          CICS CF data table server program
//STEPLIB DD     DSN=CICSTS56.CICS.SDFHAUTH,DISP=SHR  Authorized library
//SYSPRINT DD    SYSOUT=*              Options, messages and statistics
//DFHCFRL DD     DSN=CFDT1.UNLOADED.POOL,DISP=OLD  Unloaded table pool
//SYSIN DD      *
FUNCTION=RELOAD          Function to be performed is RELOAD
POOLNAME=PRODCFD1       Pool name
POOLSIZ=50M             Increased pool size
MAXTABLES=500           Increased max number of tables
/*
```

図 39. 再ロード JCL の例

## 領域状況サーバーのセットアップと稼働

CICS 領域状況サーバーを使用して、シスプレックス内で CICS 領域状況データを素早く共用し、最適化されたワークロード管理をサポートすることができます。領域状況サーバーは、領域状況およびユーザー・アプリケーションの要求ではなく、領域状況の要求のみにサービス提供します。

### このタスクについて

CICS 領域状況データは、その領域をホストしている CICSplex の名前が付いたデータ・テーブルを使用して、シスプレックスにブロードキャストされます。CICSplex の各領域が、CICSplex データ・テーブルの 1 件のレコードで表されます。このデータ・テーブルはカップリング・ファシリティ構造内に保持され、データ・テーブルへのアクセスはカップリング・ファシリティ・データ・テーブル (CFDT) サーバーによって制御されます。MVS イメージ内に、プールごとに 1 台の CFDT サーバーをセットアップする必要があります。

関連する領域状況テーブルのグループごとに別のプールに配置することができます。例えば、実動用のプールと、テスト用のプールを作成できます。プールは、カップリング・ファシリティ・リソース管理 (CFRM) ポリシーでリスト構造として定義されます。プール名と接頭部 DFHCF を使用してサーバー名が形成されます。プール名は、サーバーの始動 JCL で指定されます。

ベスト・プラクティスとして、領域状況サーバーとそのプール名は、ユーザー・アプリケーションの目的で使用されるものとは別のものにする必要があります。領域状況サーバーとプールをユーザー・アプリケーションと共用しないでください。

### 領域状況サーバーのプール名に関する考慮事項

CICSplex SM で実装されるデフォルトのプール名は DFHRSTAT です。デフォルトの領域状況サーバー・プール名を使用することを強くお勧めします。以下のシナリオでは、デフォルト以外のプール名を使用できます。

- ・領域状況サーバーと同じシスプレックス上に重複する CICSplex 名が既にある。
- ・領域状況サーバーと同じシスプレックス上に重複する CICSplex 名を指定する計画がある。
- ・デフォルトのプール名の使用が禁止される厳格な命名規則ポリシーがある。

- 複数の環境 (例えばテスト環境と実稼働環境) が同じシスプレックス上で稼働している環境になっているため、分離するための付加的な手段が必要になっている。

デフォルト名の DFHRSTAT を使用しないときは、CICSplex 内で他の領域を開始する前に、名前を変更する必要があります。CICSplex SM は、CICSplex がアクティブである間のプール名の変更を禁止しません。CICSplex がアクティブである間に変更した場合は、CICSplex 内のすべての CMAS 領域と MAS 領域 (ルーターとターゲットの両方) をできるだけ早く再始動します。すべてのルーターとターゲットが新しい領域状況サーバー・プールに転送されるまでは、マイグレーションの完了した一部の領域について最適化済みであると表示され始めた場合であっても、WLM 最適化は非アクティブになっていて、ルーティング要求は最適化されません。そのため、CICSplex 内のすべての領域を再始動するまで、CICSplex SM WLM ビューには矛盾したデータが表示されます。

さらに、関連する許可が指定されていることを確認してください。 [Security for coupling facility data tables](#) を参照してください。

## 手順

CICS 領域を領域状況サーバーとしてセットアップするには、以下の手順を実行します。

1. 領域状況サーバー・プールのリスト構造が作成されていることを確認します。

注: デフォルトの領域状況サーバー・プール名を使用することを強くお勧めします。 [205 ページの『領域状況サーバーのプール名に関する考慮事項』](#) で説明されているシナリオでは、デフォルト以外のプール名を使用できます。

- 最良のパフォーマンスを得るには、領域状況サーバー・プールのために新規の専用リスト構造を定義してください。詳しい説明は、 [206 ページの『領域状況サーバーのリスト構造の定義』](#) を参照してください。
  - オプションで、既存の CFDT プールを使用して、CICSplex データ・テーブルを保管することができます。ただし、最適化されたワークロードのスループットが、指定したプール名に対するユーザー・アプリケーションのアクティビティによって低下する場合があります。また、そのプールのアプリケーション・スループットが、シスプレックスの最適化されたワークロードによって影響を受ける場合があります。
2. 領域状況サーバーの開始ジョブを定義し、そのジョブを MVS バッチ領域で実行します。  
詳しくは、 [208 ページの『領域状況サーバーの開始』](#) を参照してください。

## 次のタスク

### 領域状況サーバーの管理

領域状況サーバーが正常に始動した後に、MVS **MODIFY** コマンドを使用して領域状況サーバーを制御できます。詳しくは、 [210 ページの『領域状況サーバーの制御』](#) を参照してください。

### 領域状況サーバーの削除

必要に応じて (例えばサービス・アップグレードやクリーンなシスプレックス 再始動のためなど)、 [この指示に従って領域状況サーバーを削除できます。](#)

## 領域状況サーバーのリスト構造の定義

領域状況サーバー・プールは、カップリング・ファシリティ・データ・テーブルのリスト構造に定義されます。カップリング・ファシリティ・リソース・マネージャー (CFRM) ポリシーにこのリスト構造を定義します。

### このタスクについて

CICS 状況を保管するためのストレージをカップリング・ファシリティに割り振る必要があります。

CICS は、領域が属する CICSplex 名から名付けられたカップリング・ファシリティ・データ・テーブルに CICS 領域の状況を記録します。このテーブルは、その CICSplex の CICSplex 定義に指定された CFDT プールに属している必要があります。デフォルト名は DFHRSTAT です。各 z/OS イメージ内に、その CICSplex に属している CICS 領域にサービス提供する領域状況プールごとに領域状況サーバーが存在する必要があります。CICSplex データ・テーブルには、その CICSplex の領域ごとに 1 つの領域状況レコードが含まれています。

IXCMIAPU ユーティリティーを使用して、現在のカップリング・ファシリティ・リソース管理 (CFRM) ポリシーに構造を定義します。このユーティリティーの例については、SYS1.SAMPLIB ライブラリーのメンバー IXCCFRMP を参照してください。領域状況サーバー・プールのポリシー・ステートメントの例が [208 ページの図 41](#) に示されています。

リスト構造に対するサーバー・アクセスを許可する必要があります。詳しくは、[Authorizing server access to a list structure](#) を参照してください。

## 手順

### 1. リスト構造の名前を指定します。

この名前は、選択したプール名に接頭部 DFHCFLS\_ を追加して、DFHCFLS\_*poolname* という形式で指定します。

CICSplex SM で実装されるデフォルトのプール名は DFHRSTAT です。デフォルトの領域状況サーバー・プール名を使用することを強くお勧めします。以下のシナリオでは、デフォルト以外のプール名を使用できます。

- ・領域状況サーバーと同じシスプレックス上に重複する CICSplex 名が既にある。
- ・領域状況サーバーと同じシスプレックス上に重複する CICSplex 名を指定する計画がある。
- ・デフォルトのプール名の使用が禁止される厳格な命名規則ポリシーがある。
- ・複数の環境 (例えばテスト環境と実稼働環境) が同じシスプレックス上で稼働している環境になっているため、分離するための付加的な手段が必要になっている。

デフォルト名の DFHRSTAT を使用しないときは、CICSplex 内で他の領域を開始する前に、名前を変更する必要があります。CICSplex SM は、CICSplex がアクティブである間のプール名の変更を禁止しません。CICSplex がアクティブである間に変更した場合は、CICSplex 内のすべての CMAS 領域と MAS 領域 (ルーターとターゲットの両方) をできるだけ早く再始動します。すべてのルーターとターゲットが新しい領域状況サーバー・プールに転送されるまでは、マイグレーションの完了した一部の領域について最適化済みであると表示され始めた場合であっても、WLM 最適化は非アクティブになっていて、ルーティング要求は最適化されません。そのため、CICSplex 内のすべての領域を再始動するまで、CICSplex SM WLM ビューには矛盾したデータが表示されます。

EYUSTARTCPLEXDEF ビュー・セットを使用して、CICSplex を定義する操作や変更する操作を実行できます。CPLEXDEF 詳細ビューを使用して、領域状況 (RS) サーバーのカップリング・ファシリティ (CF) チューニング・パラメーターを変更できます。これらのパラメーターにより、シスプレックスの最適化ワークロード・ルーティングが提供されます。

### 2. リスト構造のサイズを指定します。

[208 ページの図 40](#) に示されたレコード・サイズと計算はユーザー自身の情報として役立ちますが、CFSizer ツールを使用して、**INITSIZE** パラメーターと **SIZE** パラメーターを計算する必要があります。CFSizer ツールは最小値を考慮しますが、[208 ページの図 40](#) の計算では考慮しません。CFSizer ツールを使用して有効なサイズ変更パラメーターを取得できない場合、CICS ランタイム障害と、DFHCF0403、DFHCF0409、および DFHCF0481 のメッセージが発生することがあります。

CFSizer ツールを使用するときは、「**CICS データ・テーブル・リスト構造 (CICS Data Tables list structure)**」を選択し、以下の値を指定します。

#### テーブルの最大数

CICSplex SM で定義済みの、シスプレックス最適化ワークロードを使用する CICSplex の数を指定します。これは通常、きわめて低い数値です。

#### 丸められたレコードの平均サイズ

40

#### 合計レコード数

すべての CICSplex に接続する CICS 領域の総数を指定します。

#### ターゲット使用率 (%)

デフォルト値を使用します。

#### 最大拡張率 (%)

デフォルト値を使用します。

ALLOWAUTOALT(YES) を指定することもできます。これにより、項目に対する要素の比率を自動的に変更でき、構造内のスペースをより効率的に使用できます。

領域状況レコードの長さは、約 40 バイトです。

PLEX1 に 100 領域、PLEX2 に 300 領域、および PLEX3 に 1000 領域がそれぞれ含まれている場合、必要な構造は次のとおりです。

- プール名 = DFHRSTAT、テーブル名 = PLEX1、100 領域 x 40 バイト = 合計 4 000 バイト
- プール名 = DFHRSTAT、テーブル名 = PLEX2、300 領域 x 40 バイト = 合計 12 000 バイト
- プール名 = DFHRSTAT、テーブル名 = PLEX3、1000 領域 x 40 バイト = 合計 40 000 バイト

図 40. 例: 必要な構造の計算

3. ポリシーを保管できるカップリング・ファシリティの設定リストを指定します。
4. 新しい構造定義で CFRM の新規ポリシーを更新したときに、次の MVS コマンドを使用してポリシーをアクティブ化します。

```
SETXCF START,POLICY,POLNAME=policyname,TYPE=CFRM
```

ここで、*policyname* は、開始される CFRM ポリシー (例えば、DFHCFLS\_DFHRSTAT) です。

リスト構造のための CFRM ポリシー・ステートメントを定義しても、リスト構造は作成されないことに注意してください。この構造が作成されるのは、その構造に対する接続が最初に行われた時で、この接続の試行は、対応するプールを参照する最初のカップリング・ファシリティ・データ・テーブル (CFDT) サーバーが始動されたときに行われます。

## 例

```
STRUCTURE NAME(DFHCFLS_DFHRSTAT)
  SIZE(7168)
  INITSIZE(6144)
  PREFLIST(FACIL01,FACIL02)
```

図 41. 領域状況サーバーのリスト構造の定義の例

## 領域状況サーバーの開始

領域状況サーバーを開始するには、領域状況サーバーの開始ジョブを定義し、そのジョブを MVS バッチ領域で実行する必要があります。

### 始める前に

領域状況サーバーを始動する際、そのサーバーの MVS イメージのプールをアクティブにします。そのため、領域状況サーバー領域を開始する前に、プールに使用する領域状況サーバー構造を定義する必要があります。詳しくは、[206 ページの『領域状況サーバーのリスト構造の定義』](#)を参照してください。

### このタスクについて

サーバーは開始タスク、開始ジョブ、またはバッチ・ジョブとして始動できます。このタスクでは、MVS バッチ領域で実行する領域状況サーバー・ジョブの開始方法について説明します。このジョブまたはタスクでは、CICS 許可ライブラリー CICSTS56.CICS.SDFHAUTH から領域状況サーバー・プログラム DFHCFMN を開始する必要があります。

### 手順

1. JCL に定義された SYSIN データ・セット内、または EXEC ステートメントの **PARM** パラメーターに DFHCFMN プログラムを指定します。
2. DFHCFMN プログラムの必須およびオプションの始動パラメーターを指定します。

始動パラメーターを SYSIN データ・セットと **PARM** パラメーターの両方に指定した場合、**PARM** 値により SYSIN 値がオーバーライドされます。これは、**MVS START** コマンドにより **PARM** 値をオーバーライドできるためです。

a) 印刷ファイルに SYSPRINT DD ステートメントを指定する必要があります。

b) サーバー・パラメーターに SYSIN DD ステートメントを指定する必要があります。

**ヒント:** MVS イメージ全体ですべてのプール関連パラメーターの一貫性を確保するには、同じプールにアクセスするすべてのサーバーに対して同じ SYSIN パラメーター・データ・セットまたはその同一のコピーを使用すること、およびサーバー間で異なるあらゆるパラメーターを PARM フィールドに指定する必要があります。

c) 領域状況プール名を指定する必要があります。

d) ライセンスのアクティベーション・データ・セット (SDFHLIC ライブラリー) を STEPLIB DD ステートメントに連結する必要があります。

e) **REGION** パラメーターを指定できます。

このパラメーターにより、カップリング・ファシリティ・データ・テーブル・サーバー領域に、同時実行可能な最大数のデータ・テーブル要求を処理するための十分なストレージが確保されます。

f) **TIME=NOLIMIT** を指定できます。

サーバー・タスクは、ほとんどの通常処理の間、待機状態のままです。これは、サーバー処理がクライアント CICS 領域の TCB で実行されるためです。このパラメーターを省略すると、SYS1.PARMLIB の SMFPRMxx メンバーに指定した JWT 値によっては、サーバー・ジョブが異常終了 S522 (待機時間制限の超過) で失敗することがあります。

g) 必要に応じてパラメーターを追加指定します。

例えば、プールでサポートする予定のキューの最大数や、サーバーが割り振るバッファの数を制御することができます。必要なセキュリティ権限の追加も必要になる場合があります。[Authorizing a CICS region to a coupling facility data table](#) を参照してください。

## タスクの結果

領域状況サーバーが稼働しており、領域状況データを受信し、接続先の CICS 領域にそのデータをブロードキャストする準備が整っています。CICS 領域は、CICSplex 定義に指定されているプール名を使用して接続されます。

## 領域状況サーバーの JCL の例

```
//PRODRSS1 JOB    ...
//RSSERVER EXEC   PGM=DFHCFMN,REGION=40M,TIME=NOLIMIT
//STEPLIB DD      DSN=CICSTS56.CICS.SDFHAUTH,DISP=SHR
//              DD DSN=CICSTS56.CICS.SDFHLIC,DISP=SHR
//SYSPRINT DD      SYSOUT=*
//SYSIN DD        *
POOLNAME=DFHRSTAT
MAXTABLES=100
/*
```

CICS CFDT Server for RS
Authorized library
License activation data set
Messages and statistics
Pool name
Allow up to 100 tables

図 42. 領域状況サーバー・アドレス・スペースを開始する JCL の例

セキュリティ・パラメーターの例については、[Authorizing a CICS region to a coupling facility data table](#) を参照してください。



## 領域状況サーバーの制御

コマンドを発行して、領域状況サーバーを制御することができます。その際、**MVS MODIFY (F)** コマンドを使用して、サーバー領域のジョブ名または開始タスク名を指定し、その後、サーバー・コマンドを指定します。

### このタスクについて

簡易書式の F を使用した場合の MVS MODIFY コマンドの一般的な形式は、以下のようになります。

```
F job_name,command parameters... comments
```

MODIFY コマンドを使用して、ジョブまたは開始タスクに情報を渡します。そのタスクで領域状況サーバーを制御するためのコマンドを以下にまとめます。

### 手順

- サーバー初期設定パラメーターを変更する場合は、**MVS SET** コマンドを使用します。

```
SET keyword=operand[,keyword=operand,...]
```

MVS **SET** コマンドと同じく、**SET** の代わりに **T** という省略形を使用することもできます。詳しくは、[198 ページの『SET コマンドのオプション』](#)を参照してください。

- 1 つ以上のパラメーター値や統計の要約情報をコンソールに表示する場合は、**DISPLAY** コマンドを使用します。

```
DISPLAY keyword[=operand][,keyword[=operand],...]
```

**DISPLAY** の有効なキーワードとして、すべての初期設定パラメーターと、[199 ページの『DISPLAY および PRINT コマンド・オプション』](#)に記されている一連のパラメーターを使用できます。

MVS **DISPLAY** コマンドと同じく、**DISPLAY** の代わりに **D** という省略形を使用することもできます。

- DISPLAY** コマンドで生成する情報をファイルに出力する場合は、**MVS PRINT** コマンドを使用します。

```
PRINT keyword[=operand][,keyword[=operand],...]
```

**PRINT** コマンドでは、**DISPLAY** と同じ出力を生成したり同じキーワードを使用したりできますが、出力先が印刷ファイルになります。

- テーブルを削除する場合は、**DELETE TABLE=name** コマンドを使用します。

このコマンドを実行する時点でテーブルが使用中になっていると、成功しません。**DEL** という省略形のコマンドを使用することもできます。

- サーバーを正常に停止する場合は、**STOP** コマンドを使用します。

サーバーは、アクティブな接続がまず終了するのを待ちながら、新しい接続が確立されないようにします。**P** という省略形を使用することもできます。さらに、**MVS STOP** コマンドを使用することも可能です。その場合は、**MVS MODIFY** コマンドでサーバー・コマンド **STOP** を実行するのと同じ結果になります。**STOP** コマンドの構文は以下のとおりです。

```
STOP|P [jobname.]identifier[,A=asid]
```

- サーバーをすぐに終了する場合は、**CANCEL** コマンドを使用します。

RESTART オプションによってサーバーの自動リスタートを実行するかどうかを指定することもできます。**CANCEL RESTART** については、[178 ページの『CANCEL コマンドのオプション』](#)を参照してください。

- サーバーは、構造サイズを変更するオペレーター・コマンド **SETXCF** などのシステム間拡張サービス (XES) イベントにも応答します。

サーバーは、カップリング・ファシリティにアクセスできなくなると、サーバー **CANCEL** コマンドを自動的に発行して、すぐにサーバー自体をクローズします。

## SET コマンドのオプション

SET コマンドを使用して、サーバー初期設定パラメーターのグループを変更できます。

それらのシステム初期設定パラメーターのグループを次に示します。

- 統計パラメーター
- デバッグ・トレース・パラメーター
- ロック待機パラメーター
- 警告パラメーター
- 自動 ALTER パラメーター

以下の SET キーワードを使用すると、最後に終了したときに未解決の作業単位が存在した非アクティブな CICS 領域のサーバーのリカバリー状況を変更できます。

### RESTARTED=*applid*

指定した APPLID のために一時的なリカバリー可能接続を確立します。これにより、領域が最後に終了されたときにコミット処理中またはバックアウト処理中だった作業単位が解決されます。また、未確定の作業単位が残っているかどうかを示されます。

このキーワード RESTART または REST と省略することができます。

### COMMITTED={*applid*|*applid.uowid*}

指定した APPLID のために一時的なリカバリー可能接続を確立し、すべての未確定の作業単位をコミットします。あるいは、*uowid* も指定した場合は、その指定した作業単位をコミットします。

同じ作業単位で更新されたカップリング・ファシリティ・データ・テーブル・リソースと他の CICS リソースの間で不整合が生じる可能性があるため、このコマンドは、元の CICS 領域を再始動して作業を正常に解決することが不可能な場合にのみ使用してください。

このキーワード COMMIT または COMM と省略することができます。

### BACKEDOUT={*applid*|*applid.uowid*}

指定した APPLID のために一時的なリカバリー可能接続を確立し、すべての未確定の作業単位をバックアウトします。あるいは、*uowid* も指定した場合は、その指定した作業単位をバックアウトします。

同じ作業単位で更新されたカップリング・ファシリティ・データ・テーブル・リソースと他の CICS リソースの間で不整合が生じる可能性があるため、このコマンドは、元の CICS 領域を再始動して作業を正常に解決することが不可能な場合にのみ使用してください。

このキーワード BACKOUT または BACK と省略することができます。

以下の SET パラメーターを使用すると、特定のテーブルに関連するオプションを変更できます。

### TABLE=*name*

同じコマンドで、以下のテーブル関連パラメーターを適用するテーブルを指定します。このパラメーターは、テーブル関連パラメーターの前に指定する必要があります。

### MAXRECS=*number*

この前の TABLE パラメーターで指定したテーブルに保管できるレコードの最大数を変更します。

現在のテーブルのレコード数より少ない値を最大数に設定した場合は、その新しい上限を下回るまでレコードが削除されないと、新しいレコードは保管できません。リカバリー可能テーブルの場合は、これはレコードの更新もできないことを意味します。リカバリー可能な更新プロセスでは、書き直し処理で新しいレコードが追加され、トランザクションが完了すると元のレコードが削除されるからです。

このキーワードは、MAXNUMRECS としても指定できます。

### AVAILABLE={YES|NO}

この前の TABLE パラメーターで指定したテーブルを新しい OPEN 要求で使用できるかどうかを指定します。テーブルを使用不可にすると、それより後にそのテーブルに対して OPEN 要求を発行した CICS 領域は、テーブルが使用できないことを示す応答を受け取ります。ただし、現在テーブルを開いている領域への影響はありません。テーブルに使用不可のマークが付いている場合であっても、サーバーは、再始動処理時にリカバリー可能な処理を解決するために CICS 領域の代わりに暗黙的にテーブルを開くことができます。

このキーワードは、AVAIL と省略することができます。

**SET コマンドの例:** 次の例は、統計オプションを変更します。

```
SET STATSOPT=BOTH,EOD=21:00,STATSINT=06:00
```

次の例は、指定したテーブルで許可されているレコードの最大数を変更します。

```
SET TABLE=PAYECFT1,MAXRECS=200000
```

### DISPLAY および PRINT コマンド・オプション

DISPLAY (および PRINT) コマンドを使用して、初期設定パラメーターの値および追加情報を表示できます。

追加情報を提供するパラメーターの一部は、総称名をサポートしています。総称名を指定するには、以下のワイルドカード文字を使用します。

- \* (アスタリスク記号)。これはパラメーター値の任意の場所で使用でき、任意の値の 0 個から 8 個までの文字を表します。例えば、CICSH\* は、文字 H で識別される CICSplex 内のすべての CICS APPLID を表します。
- % (パーセント記号)。これはパラメーター値の任意の場所で使用でき、任意の値の 1 文字のみを表します。例えば、CICS%T\* は、すべての CICSplex 内のすべての TOR APPLID を表します。

DISPLAY コマンドおよび PRINT コマンドでサポートされるパラメーターは、次のとおりです。

### APPLIDS

プールへのリカバリー可能接続を現在保持しているすべての CICS 領域の APPLID および MVS システム名を表示します。このコマンドは、MODIFY コマンドの発行先のサーバーに関する情報だけでなく、同じプールに接続されている他のすべてのサーバーに関する情報も返します。

このキーワードは、APPLID、APPLS、または APPL と省略することができます。

### APPLID={*applid*|*generic*}

サーバーのプールへのリカバリー可能接続を現在保持し、その APPLID が *applid* または *generic* と一致する各領域の APPLID および MVS システム名を表示します。このコマンドは、MODIFY コマンドの発行先のサーバーに関する情報だけでなく、同じプールに接続されている他のすべてのサーバーに関する情報も返します。

#### *applid*

シスプレックス内の 1 つの領域にのみ一致する特定の APPLID に対してこの値を使用します。

#### *generic*

複数の領域に関する情報を取得する場合は、適切な総称値を使用します。

*applid* も *generic* も指定されていない場合、サーバーではこれをコマンド DISPLAY APPLIDS と同等に扱います。

このキーワードは、APPLIDS、APPLS、または APPL と指定することもできます。

### ARMREGISTERED

ARM 登録が正常に完了したかどうかを示します (YES または NO)。

### CONNECTIONS

コマンドの発行先のサーバーに現在接続されている領域のジョブ名およびアプリケーション ID を表示します。

このキーワードは、CONN と省略することができます。

### TABLES

プールに現在割り振られているすべてのテーブルの名前を表示します。

### TABLE={*name*|*generic\_name*}

特定のテーブル、または名前が総称名に一致するテーブルのセットの属性と状況に関する情報を表示します。

テーブル名が指定されていない場合、これは DISPLAY TABLES と同等に扱われます。

## TABLEUSERS

プールに現在定義されている各テーブルを使用中の領域の CICS APPLID を表示します。

このキーワードは、TABLEU と省略することができます。

### TABLEUSERS={name|generic\_name}

指定されたテーブル、または名前が総称名と一致するテーブルの各セットを現在使用している領域の CICS APPLID を表示します。

テーブル名が指定されていない場合、これは DISPLAY TABLEUSERS と同等に扱われます。

このキーワードは、TABLEU と省略することができます。

## UOWIDS

未解決のリカバリー可能作業単位を現在保持しているすべての領域のアプリケーション ID と、現在未確定か、コミットまたはバックアウトの処理中である作業単位の数を表示します。表示される情報には、解決プロセスをまだ開始していない作業単位、つまり、依然として未完了の作業単位は含まれていません。

このキーワードは、UOWS と省略することができます。

### UOWIDS={applid|generic\_applid}v{applid.\*|generic\_applid.\*}

指定された領域が未解決のリカバリー可能作業単位を現在保持している場合は、その領域のそれらの作業単位についての情報が表示されます。表示される情報には、解決プロセスをまだ開始していない作業単位、つまり、依然として未完了の作業単位は含まれていません。返される情報は、使用されるオペランドの形式によって異なります。

#### applid|generic\_applid

この形式のオペランドは、現在未確定か、コミットまたはバックアウトの処理中である作業単位の数を表示します。

applid を指定すると、サーバーは、シスプレックス内の 1 つの領域にのみ対応する特定の APPLID についての UOW 情報を表示します。

generic\_applid を指定すると、サーバーは、指定された汎用 APPLID に一致するすべての APPLID についての UOW 情報を表示します。

#### applid.\*|generic\_applid.\*

この形式のオペランドは、以下を表示します。

- 個々の作業単位の状態およびローカル UOWID。これに続いて、
- 現在未確定か、コミットまたはバックアウトの処理中である作業単位の数の要約。

applid.\* を指定すると、サーバーは、シスプレックス内の 1 つの領域にのみ対応する特定の APPLID についての UOW 情報を表示します。

generic\_applid.\* を指定すると、サーバーは、指定された汎用 APPLID に一致するすべての APPLID についての UOW 情報を表示します。

このキーワードは、UOWS と省略することができます。

### UOWID=applid.uowid

アプリケーション ID およびローカル作業単位 ID (UOWID) で識別される、未解決の個々の作業単位の状態を表示します。16 桁の 16 進数字としてローカル UOWID を入力します。

このキーワードは、UOW と省略することができます。

## 統計の要約の DISPLAY オプションおよび PRINT オプション

以下のパラメーターを使用して、統計を表示または印刷します。

## CFSTATS

カップリング・ファシリティ・インターフェースのアクセスやサーバーからの 応答の統計を表示します。

このキーワードは、CFST または STATSCF と指定することもできます。

## POOLSTATS

プール・リスト構造全体の使用状況の統計を表示します。この統計は、カップリング・ファシリティ・アクセス要求によって戻される情報に基づいています。そのため、コマンドの発行先のサーバーを介して実行された最新の要求の時点の統計に過ぎません。

このキーワードは、POOLST と省略することができます。

## TABLESTATS

コマンドの発行先のサーバーによって処理される、各テーブルの要求の統計と、作業単位制御などのテーブル固有ではない要求も含む、処理されたすべての要求の要約が表示されます。

現在の統計インターバルの開始以降の、要求数がゼロ以外の表のみが表示されることにご注意ください。

このキーワードは、TABLEST として指定することもできます。

## TABLESTATS={name|generic\_name}

指定したテーブルの要求の統計を表示します。

### name

サーバーによってアクセスされたプール内の固有のテーブル名。このテーブルの統計のみを戻します。

### generic\_name

複数のテーブルに関する統計を取得するために使用できる総称名。総称名と一致するすべてのテーブル名の統計を戻します。

このキーワードは、TABLEST と省略することができます。

## STORAGESTATS

サーバー・アドレス・スペースの主記憶域割り振りの統計を表示します。

このキーワード STORAGEST または STGST と省略することができます。

## 情報の結合リストでの **DISPLAY** および **PRINT** オプション

以下のキーワードは、情報の結合リストを表します。

## PARAMETERS

主要なパラメーター値を表示します。それらは、POOLNAME、SECURITY、SECURITYPREFIX、統計オプションとリスト構造オプションです。

このキーワードは、PARM または PARMS と省略することができます。

## ALLPARAMETERS

すべてのパラメーター値を表示します。

このキーワードは、ALLPARMS と省略することができます。

## STATISTICS

使用可能なすべての統計を表示します。

このキーワードは **STAT** または **STATS** という省略形にすることができます。

## INITIALIZED

初期設定が完了すると通常表示されるパラメーターおよび統計を表示します。これは、PARM、POOLSTATS、STGSTATS と同等です。

このキーワードは、INIT と省略することができます。

## ARM

すべての ARM 関連のパラメーター値を表示します。

- ARMELEMENTNAME
- ARMELEMENTTYPE
- ARMREGISTERED

このキーワードは、ARMSTATUS としてコーディングできます。

## CANCEL コマンドのオプション

CANCEL コマンドを使用して、自動再始動を要求できます。

以下のパラメーターを指定します。

### RESTART={NOvYES}

自動再始動を要求するかどうかを指定して、サーバーをすぐに強制終了します。デフォルト値は RESTART=NO です。

サーバーでカップリング・ファシリティの接続に関するリカバリー不能な問題が起きて、接続が失われたり、ストラクチャーで障害が発生したりすると、サーバーは、CANCEL RESTART=YES コマンドを使用して自身をキャンセルします。その結果、既存の接続が終了して、サーバーがシャットダウンします。その後、サーバー・ジョブの新しいインスタンスが開始されます。

サーバー・コマンド CANCEL RESTART=YES または MVS コマンド CANCEL jobname,ARMRESTART を使用して、サーバーを明示的に再始動することもできます。

RESTART=YES の代わりに RESTART、RESTART=NO の代わりに NORESTART を直接入力することも可能です。

## 領域状況サーバー・プールの削除

カップリング・ファシリティ・リスト構造を削除することにより、領域状況サーバー・プールを削除することができます。サービス・アップグレードで、またはシスプレックスのクリーンな再始動が必要なときにこれを行う可能性があります。

### 始める前に

プールに接続されているサーバーがない場合にのみ構造を削除できます。そうでない場合、MVS はコマンドを拒否します。

### このタスクについて

例えば、次のようになります。

```
SETXCF FORCE,STRUCTURE,STRNAME=DFHCFLS_poolname
```

次に示す XCF コマンドを実行して、プールが正常に削除されたことを確認できます。

```
D XCF STRUCTURE,STRNAME=DFHCFLS_poolname
```

CICS の領域およびワークロードの実行中に領域状況サーバー構造を削除する場合、CICSplex SM WLM 最適化機能を無効にする必要があることに注意してください。

### 次のタスク

削除されたプールに対してサーバーを始動(または、プールを再ロード)しようとする、その構造は新しい構造として割り振られます。新規に割り振られた構造は、現在アクティブな CFRM ポリシーにより指定されたサイズ属性と位置属性、およびサーバーの初期設定パラメーター (特に、MAXTABLES) で決定されたその他の値を使用します。

## 名前付きカウンター・サーバーのセットアップと実行

CICS では、Parallel Sysplex 環境のアプリケーションで使用するために、固有のシーケンス番号を効率的に生成することができます。例えば、注文や送り状に固有の番号を割り振れます。名前付きカウンター・サーバーは、数字の各シーケンスを名前付きカウンターとして保持します。

### このタスクについて

以下の手順は、名前付きカウンター・サーバーをセットアップして管理するための大まかなステップを示しています。各ステップの詳細については、その後の各トピックを参照してください。



## 手順

1. 名前付きカウンターのオプション・テーブルを定義します。
2. リスト構造を定義します。
3. 名前付きカウンター・サーバー・ジョブを定義して開始し、MVS バッチ領域で実行します。
4. 名前付きカウンター・サーバー領域を管理します。
  - 名前付きカウンター・サーバーを制御するコマンドを発行します。
  - 名前付きカウンター・プールのサイズを変更します。
  - 名前付きカウンター・プールを削除するか空にします。
  - 名前付きカウンター・プールのアンロードと再ロードを実行します。
  - 名前付きカウンター・プールのリスト構造をダンプします。

## 名前付きカウンター・サーバーの概要

数字が割り当てられるたびに、対応する名前付きカウンターが自動的に増分され、次の要求が順番に次の数字を取得します。名前付きカウンターは、単一領域 CICS システムの Common System Area (CSA) の COUNTER に相当するシスプレックスの機能です。

名前付きカウンター・サーバーには、名前付きカウンターを定義および使用するためのすべての機能が用意されています。名前付きカウンターはそれぞれ、次の要素で構成されています。

- 16 バイトの名前
- 現行値
- 最小値
- 最大値

内部的には値は 8 バイト (ダブルワード) のバイナリー数値として保管されますが、ユーザー・インターフェースでは 1 バイトから 8 バイトまでの任意の長さ (通常は 4 バイトを使用) の値として扱えます。

名前付きカウンターは、名前付きカウンターのプールに保管されます。各プールは、小さいカップリング・ファシリティーのリスト構造であり、キーはありますが、データはありません。プール名はリスト構造名の一部を形成します。名前付きカウンターは、それぞれ、指定した名前をキーとして持つリスト構造エントリーとして保管されます。また、次の値を求める要求に必要なのは、単一のカップリング・ファシリティー・アクセスのみです。

**警告:** カップリング・ファシリティーに障害が発生すると、カウンターは失われます。

### 名前付きカウンターの構造およびサーバー

各 MVS イメージ内に、その MVS イメージの CICS 領域やバッチ・ジョブからアクセスする名前付きカウンター・プールごとに 1 台の名前付きカウンター・サーバーが存在する必要があります。

名前付きカウンター・プールは、カップリング・ファシリティー・リソース管理 (CFRM) ポリシーでリスト構造として定義されます。プール名 (プール名と接頭部 DFHNC を使用してサーバー名が形成されます) は、サーバーの始動 JCL で指定されます。

217 ページの図 43 は、名前付きカウンター・サーバーに 3 つの CICS AOR がリンクされた並列シスプレックスを示しています。

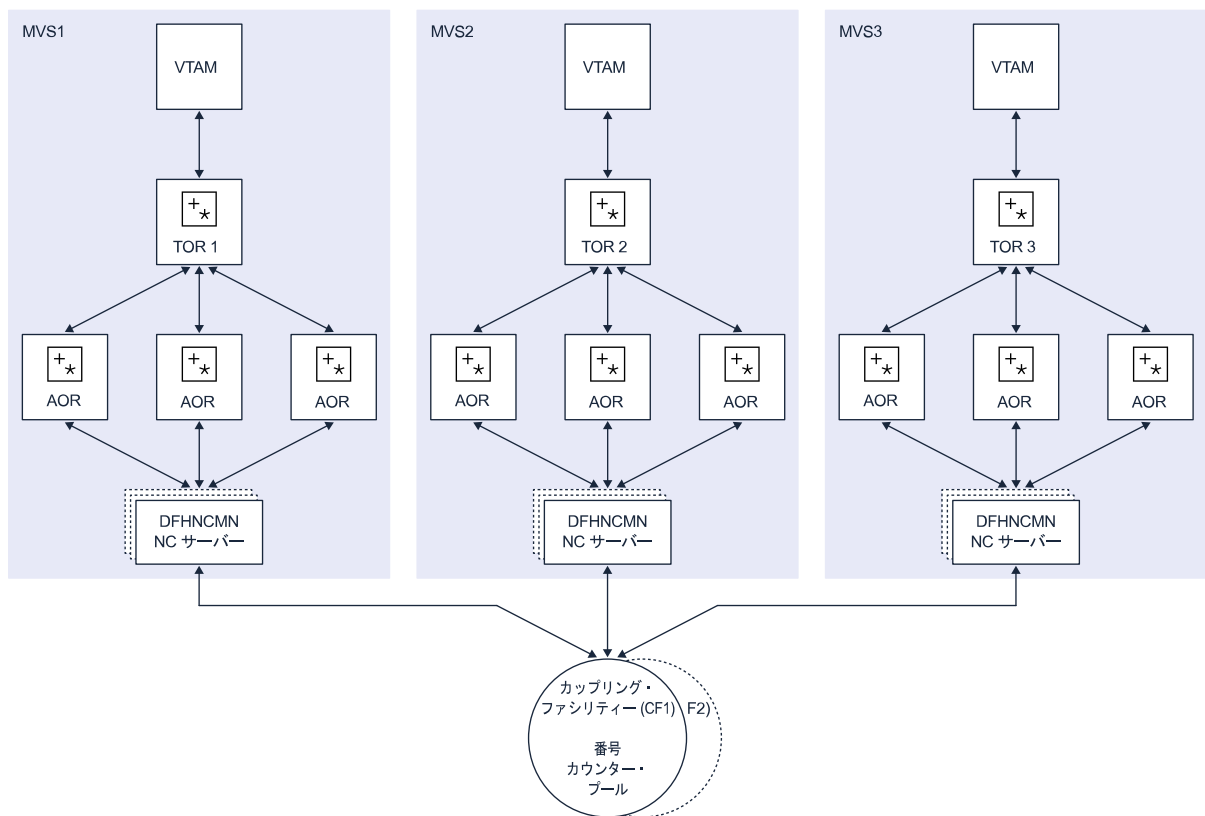


図 43. 並列シスプレックスの名前付きカウンターに関する概念ビュー

## 名前付きカウンターへのアプリケーション・プログラム・アクセス

CICS には、名前付きカウンター・ファシリティのためのコマンド・レベルの API が用意されています。名前付きカウンターを参照するために、アプリケーション・プログラムで名前付きカウンターが保管されているプールの実際の名前を指定することも、ダミー・プール選択パラメーターを指定することもできます。ダミー・プールの値は、オプション・テーブル DFHNCOPT で指定する **POOL** パラメーターによって実際のプール名にマップされます。

ダミー・プールを指定すると、アプリケーション・プログラムでプール選択パラメーターを変更する必要がないため、(例えば、テスト・プールを実動プールから切り離すために) 別のプールを使用することが簡単になります。

CICS 領域で使用するプールを変更するには、STEPLIB からオプション・テーブルの別のコピーをロードするか、プール名選択パラメーターのほかにジョブ名および CICS APPLID を条件としてプール名の選択を行う共通のオプション・テーブルを使用します。オプション・テーブルは、プール選択パラメーターで指定された適切なプールを選択するためのユーザー指定プログラムの呼び出しもサポートします。

## セキュリティ

サーバーには、名前付きカウンター・プールが定義されているカップリング・ファシリティのリスト構造へのアクセスを許可する必要があります。また、名前付きカウンター・サーバーとして動作することも許可する必要があります。

必要な許可の定義方法については、[名前付きカウンター・プールおよびサーバーへのアクセスの許可](#)を参照してください。

注：個々の名前付きカウンターへのアクセスは制御できません。

## 名前付きカウンター・オプション・テーブルの定義

名前付きカウンター呼び出し可能インターフェースは、DFHNCTR 呼び出しに対する応答として、DFHNCOPT オプション・テーブルを参照して実際のプール名を決定します。

### このタスクについて

プール・セクター値が最初に検出されたときに、プール名がオプション・テーブルを介して決定されます。続いて、名前が保管され、同じ TCB の同じプール・セクターに対する後続のすべての要求で使用されます。この動作は、TCB の存続期間中、または、そのプール・セクター値を指定する NC\_FINISH 関数が使用されるまで継続します。CICS は、デフォルトの DFHNCOPT をソース形式で提供します。ユーザーは DFHNCO マクロを使用して、その DFHNCOPT をカスタマイズおよび生成できます。オプション・テーブルの一般的な使用法は、アプリケーション・プログラムでプール名を変更することなく、実動領域およびテスト領域で異なるカウンター・プールを使用できるようにすることです。

複数のバージョンのオプション・テーブルを維持しなくて済むように、テーブル項目を使用して、DFHNCTR 呼び出しで指定されたプール選択パラメーターだけでなく、CICS 領域のジョブ名およびアプリケーション ID に基づいてプールを選択できます。また、プールを選択するために呼び出されるユーザー出口プログラムの名前も指定できます。

DFHNCO マクロの 1 つ以上の呼び出しを使用して、オプション・テーブルを定義します。各呼び出しにより、項目に指定された選択条件がアプリケーション・プログラム要求を満たすたびに使用されるプール名またはユーザー出口プログラムを定義するオプション・テーブル項目が生成されます。最初の項目は、CSECT ステートメントを含む、テーブル・ヘッダーを自動的に生成します。最後の項目の後に、テーブル・モジュール・エントリー・ポイント DFHNCOPT を指定する END ステートメントが続きます。

## オプション・テーブル・パラメーター

以下の図で、DFHNCOPT オプション・テーブルの各パラメーターについて説明します。

```
DFHNCO [POOLSEL={ (generic_values) | * },]  
        [JOBNAME={ (generic_values) | * },]  
        [APPLID={ (generic_values) | * },]  
        {POOL={YES|NO|name} | CALL=programname}  
  
        Terminate the last DFHNCO entry with the  
        following END statement:  
  
END      DFHNCOPT
```

図 44. DFHNCOPT オプション・テーブル

POOLSEL、JOBNAME、および APPLID の各パラメーターは、入力がある要求に適用されるかどうかを判断するオプションの選択条件を指定します。各オペランドを、以下のように指定できます。

- 単一の総称名
- 括弧に囲まれた名前のリスト。リストには、名前をコンマで区切った 2 つ以上の総称名が含まれます。

各名前は、適切なパラメーターに使用できる文字と、空白以外の 0 文字以上の文字列を表すワイルドカード文字 \* と、空白以外の単一文字を表す % で構成されます。複数の総称名が指定されている場合、選択条件は、いずれか 1 つが一致する場合に満たされます。空白のプール・セレクター値は、ヌルの POOLSEL オペランド、(例えば POOLSEL= または POOLSEL=() など) を使用して一致させることができます。

### POOLSEL={ (generic1,generic2,...)v\* }

このオプション・テーブル・エントリーが、アプリケーション・プログラムによって指定されるプール選択パラメーターと、このパラメーターによって指定されている総称名のいずれかが一致する場合にのみ適用されることを指定します。

POOLSEL=、または POOLSEL=() を指定することは、8 つの空白を指定することと同等です。

POOLSEL キーワードを省略すると、デフォルトの \* になります。

### JOBNAME={ (generic1,generic2,...)v\* }

このオプション・テーブル・エントリーが、呼び出し元のジョブ名と、このパラメーターによって指定された総称名のいずれかが一致する場合にのみ適用されることを指定します。

JOBNAME キーワードを省略すると、デフォルトの \* になります。

### APPLID={ (generic1,generic2,...)v\* }

このオプション・テーブル・エントリーが、呼び出し元の CICS APPLID と、このパラメーターによって指定された総称名のいずれかが一致する場合にのみ適用されることを指定します。

APPLID キーワードを省略すると、デフォルトの \* になります。

### POOL={YESvNOvname}

使用するプール名を指定します。このパラメーターは、CALL パラメーターと相互に排他的です。オプションは以下のとおりです。

#### YES

サーバーが、アプリケーション・プログラムによって指定されたプール選択パラメーターを、実際のプール名として使用するよう指定します。すべてが空のプール選択パラメーターは、サーバーがデフォルトのプール名を使用することを意味します。呼び出しインターフェースの場合、デフォルト名は DFHNC001 です。EXEC CICS API の場合、デフォルト名は NCPLDFT システム初期設定パラメーターによって指定されます。

#### NO

サーバーがいずれのプールも使用せず、エラーで要求を拒否することを指定します。

#### name

サーバーが使用する実際のプール名を指定します。name を省略する場合、デフォルトのプールが使用されることを示します。(CALL インターフェースの場合、デフォルトのプールは常に

DFHNC001 ですが、EXEC CICS インターフェースの場合、NCPLDFT システム 初期設定パラメーターを使用してデフォルトのプールを指定できます。)

### **CALL=programname**

使用する実際のプール名を判別するために呼び出すユーザー出口プログラムの名前を指定します。このパラメーターは、POOL パラメーターと相互に排他的です。

プログラム名は、オプション・テーブルによってリンク・エディットすることも (これにより、弱い外部参照 (WXTRN) が生成される)、最初の使用時に動的にロードすることもできます。プログラムは、標準の AMODE 31 の MVS リンケージを使用して呼び出されます。その際、標準の保存域と 4 つのフィールドを以下の順序で指すパラメーター・リストを使用します。

- 8 バイトの実際のプール名の結果フィールド
- 8 バイトのプール選択パラメーター
- 8 バイトのジョブ名
- CICS の下で実行する場合は 8 バイトの APPLID で、それ以外の場合はブランク

リストの終わりを示すビットは、最後のパラメーター・アドレスに設定されます。

プログラムは再入可能である必要があります、また RMODE ANY とリンクすることで、それ (およびプログラムとリンクされている場合はオプション・テーブル) が 16 MB 境界より上にロードできるようになっている必要があります。MVS GETMAIN および FREEMAIN を使用して、一時作業用ストレージを獲得してリリースすることができます。このプログラムは、新しいプール選択値が使用されたときにのみ呼び出されるため、GETMAIN および FREEMAIN を使用してもパフォーマンスに影響はありません。

出口プログラムでは、CICS サービスを使用できません。CICS 領域で使用されている場合、長時間待ちが発生する MVS サービスの使用は避ける必要があります。これは、通常は CICS 準再入可能 (QR) TCB の下で実行されるためです。

ユーザー出口プログラムは、以下のいずれかの戻りコードをレジスター 15 に設定することで結果を示します。

**0**

ユーザー出口プログラムによって、パラメーター・リストの最初のフィールドに正常に設定されたプール名を使用します。

**4**

この呼び出しでは、プログラムはプール名を判別できません。選択条件に合わないケースとして、次のエントリーでオプション・テーブル処理を継続してください。

**8**

要求を拒否します (POOL=NO が指定された場合と同様)。

CICSTS56.CICS.SDFHLINK に提供されているデフォルトのオプション・テーブルには、以下のエントリーが含まれます。

DFHNC0	POOLSEL=DFHNC*,POOL=YES
DFHNC0	POOL=
END	DFHNCOPT

デフォルトのオプション・テーブルが使用されている場合、DFHNC で始まる文字列を指定するプール選択パラメーターは、実際のプール名として使用され、テーブル・エントリーでは POOL=YES と示されます。すべて空白の値を含む、その他すべての値は、デフォルトのプール名が割り当てられ、POOLSEL パラメーターなしの POOL= table エントリーで示されます。

このデフォルトのテーブルのソースは、CICSTS56.CICS.SDFHSAMP で提供されています。

### **オプション・テーブルを CICS で使用可能にする**

CICS 領域に名前付きカウンター・オプション・テーブルをロードできるようにするには、リンク・エディットされたテーブルを STEPLIB の APF 許可を受けたライブラリーにインストールします。あるいは、テーブルを LINK リストの適切なライブラリーにインストールすることもできます。

## 名前付きカウンター・サーバーのリスト構造の定義

名前付きカウンター機能用の1つ以上のカップリング・ファシリティ・リスト構造を定義します。ここでは、各リスト構造が名前付きカウンターのプールを表します。それぞれの名前付きカウンター・プールは、仮想記憶間サーバー領域を介してアクセスされます。

### 始める前に

カップリング・ファシリティの構造には、キー順データ・セットと同様に、保管データと、そのデータの管理とアクセスに必要な情報の両方が含まれます。内部制御情報の量は、現在のカップリング・ファシリティ・レベル (CFLEVEL) のカップリング・ファシリティ制御コードの機能およびパフォーマンスのレベルに応じて異なり、CFLEVELが高いほど情報量は増加する可能性があります。必ず、ご自分のリスト構造のストレージ要件を検討してください。詳しくは、[221 ページの『名前付きカウンターのリスト構造ストレージ』](#)を参照してください。

### このタスクについて

ユーティリティ `IXCMIAPU` を使用して、現在のカップリング・ファシリティ・リソース管理 (CFRM) ポリシーに構造を定義します。このユーティリティの例については、`SYS1.SAMPLIB` ライブラリーのメンバー `IXCCFRMP` を参照してください ([『z/OS MVS Setting Up a Sysplex』の『CFRM ポリシー・データ向け管理データ・ユーティリティ』](#)を参照)。

### 手順

1. リスト構造の名前を指定します。  
この名前は、選択したプール名に接頭部 `DFHNCLS_` を追加して、`DFHNCLS_poolname` という形式で指定します。
2. リスト構造のサイズを指定します。  
CFRM ポリシー定義で `INITSIZE` パラメーターと `SIZE` パラメーターを使用して、初期サイズと最大サイズを割り振ることができます。ストレージ要件の正確な見積もりを行うには、`IBM CFSizer` ツールを使用します。[CFSizer](#) を参照してください。
3. ポリシーを保管できるカップリング・ファシリティの設定リストを指定します。  
名前付きカウンター用のカップリング・ファシリティ・リスト構造の定義の例を以下に示します。

```
STRUCTURE NAME(DFHNCLS_PRODNC1)
  SIZE(2048)
  INITSIZE(1024)
  PREFLIST(FACIL01,FACIL02)
```

4. 新しい構造定義で CFRM の新規ポリシーを更新したときに、次の MVS コマンドを使用してポリシーをアクティブ化します。

```
SETXCF START,POLICY,POLNAME=policyname,TYPE=CFRM.
```

### タスクの結果

リスト構造のための CFRM ポリシー・ステートメントを定義しました。このアクションではリスト構造は作成されません。リスト構造が作成されるのは、その構造に対する接続が最初に行われた時で、この接続の試行は、対応するプールを参照する最初の名前付きカウンター・サーバーが始動されたときに行われます。

名前付きカウンター・サーバーを始動する前に、許可された仮想記憶間 (AXM) サーバー環境を必ず定義および開始しておいてください ([165 ページの『AXM システム・サービスの定義と開始』](#)を参照)。

### 名前付きカウンターのリスト構造ストレージ

`z Systems` カップリング・ファシリティ構造サイズ決定支援 (`CFSizer`) ツールを使用して、カップリング・ファシリティの名前付きカウンターのリスト構造ストレージ所要量を計算できます。構造サイズを増やすまたは減らす場合は、その他のパラメーターを更新する必要があるかどうかを考慮してください。

カップリング・ファシリティの構造には、キー順データ・セットと同様に、保管データと、そのデータの管理とアクセスに必要な情報の両方が含まれます。内部制御情報の量は、現在のカップリング・ファシ



リティー・レベル (CFLEVEL) のカップリング・ファシリティー制御コードの機能およびパフォーマンスのレベルに応じて異なり、CFLEVEL が高いほど情報量は増加する可能性があります。詳細については、[232 ページの『カップリング・ファシリティーのストレージ管理』](#)を参照してください。

CFSizer は、これらの要素を加味したうえで現行の CFLEVEL でカップリング・ファシリティーと通信して、ストレージ所要量を計算する Web ベース・アプリケーションです。[CFSizer](#) を参照してください。

必要なカウンター数を入力すると、CFSizer ツールは、少なくともそのカウンター数を格納できる構造のサイズを計算します。しかし、実際の運用では、構造がいっぱいにならないように、またスペース不足に関する警告メッセージが表示されないように、空き領域をいくらか確保する必要があります。使用量を構造サイズの約 75% に抑えることをお勧めします。必要なカウンターの最大数を見積もり、空き領域を計算に入れるためにその数の 1/3 の分量を増やします。

名前付きカウンター・プールに必要なスペースは、必要な名前付きカウンターの種類数に応じて異なりますが、ほとんどの場合は、最小サイズで十分です。例えば、CFLEVEL 16 の場合、1 MB の構造で最大 1000 個の名前付きカウンターを保持できます。

すべての構造サイズは、割り振り時にカップリング・ファシリティー・レベル (CFLEVEL) の次のストレージ増分値に切り上げられます。例えば、CFLEVEL 16 の場合、サイズは最も近い 1 MB 単位の値に丸められます。

カップリング・ファシリティーに使用可能なスペースがある場合は、MVS **SETXCF** コマンドを使用して、構造サイズを初期サイズから最大サイズまでの間で動的に増やし、新しいスペースを現在アクティブなサーバーですぐに使用できるようにすることができます。割り振ったスペースが多すぎた場合は、構造サイズを減少させ、他の目的のためにカップリング・ファシリティー・ストレージを解放することができます。ただし、解放されるストレージの外部にカップリング・ファシリティーが既存のデータを移す必要がある場合は、時間がかかる可能性があります。このようにしてサイズを変更した場合は、次の再作成または再ロードで構造が元のサイズに戻らないように、カップリング・ファシリティー・リソース管理 (CFRM) ポリシーの **INITSIZE** パラメーターを更新して新しいサイズを反映させてください。

## 名前付きカウンター・サーバー領域の定義と開始

名前付きカウンター・プールの名前付きカウンター・サーバー領域を開始して、MVS イメージ内でそのプールをアクティブ化します。

### このタスクについて

サーバーは開始タスク、開始ジョブ、またはバッチ・ジョブとして始動できます。このジョブまたはタスクは、名前付きカウンター・サーバー領域プログラム DFHNCMN を起動し、APF 許可ライブラリーから実行する必要があります。DFHNCMN は、CICS 許可ライブラリー CICSTS56.CICS.SDFHAUTH 内にあります。

### 手順

1. JCL に定義された SYSIN データ・セット内、または EXEC ステートメントの **PARM** パラメーターに DFHNCMN プログラムを指定します。
2. DFHNCMN プログラムの必須およびオプションの始動パラメーターを指定します。  
始動パラメーターを SYSIN データ・セットと **PARM** パラメーターの両方に指定した場合、**PARM** 値により SYSIN 値がオーバーライドされます。これは、**MVS START** コマンドにより **PARM** 値をオーバーライドできるためです。
  - a) 印刷ファイルに SYSPRINT DD ステートメントを指定する必要があります。
  - b) サーバー・パラメーターに SYSIN DD ステートメントを指定する必要があります。
  - c) TS プール名を指定する必要があります。
  - d) ライセンスのアクティベーション・データ・セット (SDFHLIC ライブラリー) を STEPLIB DD ステートメントに連結する必要があります。
  - e) **REGION** パラメーターを指定することをお勧めします。

このパラメーターにより、カップリング・ファシリティー・データ・テーブル・サーバー領域に、同時実行可能な最大数のデータ・テーブル要求を処理するための十分なストレージが確保されます。

f) TIME=NOLIMIT を指定することをお勧めします。

サーバー・タスクは、ほとんどの通常処理の間、待機状態のままです。これは、サーバー処理がクライアント CICS 領域の TCB で実行されるためです。このパラメーターを省略すると、SYS1.PARMLIB の SMFPRMxx メンバーに指定した JWT 値によっては、サーバー・ジョブが異常終了 S522 (待機時間制限の超過) で失敗することがあります。

g) 必要に応じてパラメーターを追加指定します。

例えば、プールで対応可能なキューの最大数や、サーバーが割り振るバッファの数を制御することができます。

**ヒント:** MVS イメージ全体ですべてのプール関連パラメーターの一貫性を確保するための推奨される方法は、同じプールにアクセスするすべてのサーバーに対して同じ SYSIN パラメーター・データ・セットまたはその同一のコピーを使用すること、およびサーバー間で異なるあらゆるパラメーターを PARM フィールドに指定することです。

## 例

```
//MVSnnC1 JOB ...
//NCSERVER EXEC PGM=DFHNCMN,REGION=32M,TIME=NOLIMIT      named counter server
//STEPLIB DD DSN=CICSTS56.CICS.SDFHAUTH,DISP=SHR          Authorized library
// DD DSN=CICSTS56.CICS.SDFHLIC,DISP=SHR                  License activation data set
//SYSPRINT DD SYSOUT=*                                     Messages and statistics
//SYSIN DD *
POOLNAME=MVSnnC1                                          Pool name
/*
```

図 45. 名前付きカウンター・サーバー・アドレス・スペースを開始する JCL の例

## 名前付きカウンター・サーバーのパラメーター

パラメーターは KEYWORD=value の形式で指定されます。読みやすくするために任意でキーワードを大/小文字混合で指定できます。

PARM フィールドまたは同じ SYSIN 入力行で複数のパラメーターを指定する場合、パラメーターはコンマで区切る必要があります。1 つ以上のスペースに続くテキストは、すべて説明コメントと見なされます。アスタリスクまたはスペースで始まるパラメーター行はすべて、行全体がコメントと見なされます。

いくつかのパラメーター・キーワードを複数の形式 (例えば省略形式や切り捨て形式) で入力できます。

主なパラメーターは、始動中にサーバー・プリント・ファイルにリストされます。

## 名前付きカウンター・サーバーの REGION パラメーター

JCL REGION パラメーターを使用して、名前付きカウンター・サーバー領域に、並行して実行できる名前付きカウンター要求の最大数を処理できるだけの十分なストレージを確保します。

名前付きカウンター・サーバーは、通常は 16 MB より上で 2GB より下のストレージでは 1 MB 未満のストレージを使用し、16 MB より下では 20 KB 未満のストレージを使用します。

サーバーの初期化時に、サーバーは REGION パラメーターで指定されたとおりに、16 MB より上で 2 GB より下のすべての使用可能なストレージを獲得し、そのうち 5% をオペレーティング・システム・サービスが使用するために開放します。また、24 ビットのアドレス可能ストレージ-を必要とするルーチンに使用するために、16MB より下のフリー・ストレージの 5% も獲得します。

初期設定が終了すると、サーバーは AXM ページ割り振りサービスを使用して、そのストレージを管理します。サーバー統計は、どれだけのストレージが割り振られ、16MB より上と 16 MB より下のストレージ領域の中でどれだけ使用されているかを示します。これらは統計で AXMPGANY および AXMPGLOW と呼ばれます。

サーバー領域のタスクまたは仮想記憶間要求がストレージを使い果たした場合、AXM が、異常終了のシミュレーションを使用して、そのタスクまたは要求を強制終了させ、GETMAIN 障害を示すシステム完了コード 80A を出す可能性があります。サーバーは通常、このケースでは他の要求の処理を継続することはできませんが、クリティカルなルーチンでストレージが不足すると、サーバーが強制終了する可能性があります。そのため、このようなリスクを回避するために、REGION のサイズを十分大きくしてください。

## プール名パラメーター

このパラメーター POOLNAME は常に必須です。

### POOLNAME=name

名前付きカウンター・プールの 8 文字の名前を指定します。これがサーバーによって接頭部 DFHNC に追加され、DFHNC.poolname のようにそれ自身のサーバー名が作成されます。また 接頭部 DFHNCCLS\_ にも追加され、DFHNCCLS\_poolname のようにカップリング・ファシリティー・リスト構造の名前が作成されます。

このパラメーターは、サーバーの初期設定時にのみ有効になります。指定は必須です。

このキーワードは、**POOL** と省略することができます。

## 統計パラメーター

以下のパラメーターを使用して、サーバー統計オプションを指定します。

### ENDOFDAY={00:00|hh:mm}

サーバーが終業時統計を収集してリセットする時刻 (時間と分) を指定します。

注: STATSOPTIONS パラメーターが NONE に指定されている場合、サーバーは終業時統計を印刷ファイルに書き込みます。

有効な時間の範囲は 00:00 から 24:00 までです。

このキーワードは、EOD と省略することができます。

### STATSINTERVAL={03:00|hh:mm}

統計収集間隔を 1 分から 24 時間までの間で指定します。このパラメーターは、STATSOPTIONS パラメーターが NONE に指定されている場合には無視されます。

時間間隔の範囲は 00:01 から 24:00 です。

このキーワードは、STATSINT と省略することができます。

### STATSOPTIONS={NONE|SMF|PRINT|BOTH}

サーバーが間隔統計を作成するかどうかと、作成する統計の宛先を指定します。

#### なし

サーバーは間隔統計を作成しません。

#### SMF

サーバーは間隔統計を作成し、それらを現行の SMF データ・セットにのみ書き込みます。

#### PRINT

サーバーは間隔統計を作成し、それらをサーバーの印刷ファイルにのみ書き込みます。

#### BOTH

サーバーは間隔統計を作成し、それらを現行の SMF データ・セットと、サーバーの印刷ファイルに書き込みます。

このキーワードは、**STATSOPT** と省略することができます。

## 自動リスタート・マネージャー (ARM) のパラメーター

サーバーの初期化中、サーバーは UNLOAD または RELOAD 機能のいずれかを使用してサーバー・プログラムが呼び出された場合を除き、ARM に無条件で登録します。登録が失敗している場合、サーバーは始動しません。自動リスタート・マネージャーのデフォルトの処理をオーバーライドするには、ARMELEMENTNAME および ARMELEMENTTYPE の各パラメーターを使用します。

### ARMELEMENTNAME=elementname

自動再始動の目的で ARM に対してサーバーを識別するために ARM にポリシーで使用するために、自動リスタート・マネージャーの元素名を 16 文字までで指定します。元素名に使用できる文字は A から Z、0 から 9、\$ # @ およびアンダースコア記号 ( ) です。

デフォルトの識別子の形式は DFHNCnn\_poolname で、NC はサーバー・タイプ、nn はシステムの &SYSCONV 値 (1 文字または 2 文字のいずれか)、および poolname はサーバーによってサービス提供されるプールの名前を表します。

このパラメーターは、サーバーの初期化時にのみ有効になります。

このキーワード ARMELEMENT または ARMELEMENTNAME と省略することができます。

#### ARMELEMENTTYPE=elementtype

類似のエレメントを分類する手段として ARM ポリシーで使用するために、自動リスタート・マネージャーのエレメント・タイプを 8 文字までで指定します。エレメント・タイプに使用できる文字は A-Z 0-9 \$ # および @ です。

デフォルトのエレメント・タイプは SYSCICSS です。

このパラメーターは、サーバーの初期化時にのみ有効になります。

このキーワードは、ARMELEMENTTYPE と省略することができます。

#### リスト構造パラメーター

リスト構造パラメーターは、名前付きカウンター・プールに対してリスト構造が作成される際に、リソースの初期割り振りにのみ使用される属性を指定します。最初の割り振りは、名前付きカウンター・プールのサーバーの初回始動時にのみ発生します。

#### POOLSIZE={0|number{K|M|G}}

プール・リスト構造に割り振られるカップリング・ファシリティ・ストレージの初期量を指定します。キロバイト (nK)、メガバイト (nM)、またはギガバイト (nG) で表します。

通常はこのパラメーターは省略可能で、カップリング・ファシリティ・リソース・マネージャー (CFRM) ポリシーの **INITSIZE** パラメーターを使用して構造サイズを指定できます。ただし、このパラメーターは、構造が再割り振りまたは再ロードされたが、CFRM ポリシーが必要なサイズを反映するために更新されていないような場合に便利です。

#### 0

特殊値 0 は、CFRM ポリシーに指定されたパラメーターを使用して、サーバーが初期割り振りを取得することを意味します。CFRM ポリシーが構造に対して **INITSIZE** 値を指定する場合、これによって初期割り振りが決定します。そうでない場合、CFRM **SIZE** 値 (構造の最大サイズ) が割り振られます。

#### number

ゼロ以外の値は、割り振られるストレージの初期量を指定し、CFRM ポリシーの **INITSIZE** パラメーターをオーバーライドします。この値は MVS によってカップリング・ファシリティ・レベル (CFLEVEL) の次のストレージ増分に切り上げられます。例えば、CFLEVEL 16 の場合、値は最も近い 1 MB 単位の値に丸められます。

値は CFRM **SIZE** パラメーターより小さい必要があります。そうでない場合、**POOLSIZE** は無視され、初期割り振りでは CFRM ポリシーに指定されたパラメーターが使用されます。

有効な範囲は 0 から 16777215M です。ただし、z/OS 内の構造の最大サイズより小さい値を指定する必要があります。これを超えると、z/OS のエラーが発生します。例えば z/OS バージョン 1 リリース 12 では、構造の最大サイズは 1048576 MB (1 TB) です。CFRM パラメーターの詳細については、[z/OS MVS シスプレックスのセットアップ](#)を参照してください。

このパラメーターは、サーバーの初期化時にのみ有効になり、構造が最初に割り振られる際にのみ使用されます。

#### デバッグ・トレース・パラメーター

これらのパラメーターは、集中的なデバッグ・トレースのためにのみ提供されています。

これらのオプションを実稼働環境で使用すると、パフォーマンスに大きな影響を与え、印刷ファイルが急激に増えてスプール・スペースを使い果たす原因になる場合があります。

仮想記憶間の要求からのトレース・メッセージは、トレース印刷サブタスクで印刷できる速度より速く生成されると、失われる可能性があります。その場合、トレースには、失われたメッセージの数のみが示されます。

#### CFTRACE={OFF|ON}

カップリング・ファシリティ・インターフェース・デバッグ・トレース・オプションを指定します。

#### OFF

カップリング・ファシリティ・インターフェース・デバッグ・トレースが使用不可です。



## ON

カップリング・ファシリティ・インターフェース・デバッグ・トレースによりトレース・メッセージが印刷ファイルに生成され、カップリング・ファシリティ要求インターフェースの主要パラメーターと、IXLLIST マクロからの結果が示されます。

このキーワードは、TRACECF と指定することもできます。

## RQTRACE={OFF|ON}

要求デバッグ・トレース・オプションを指定します。

## OFF

要求デバッグ・トレースは無効です。

## ON

要求デバッグ・トレースは、各仮想記憶間要求への入力的主要パラメーターと、出口における結果を示す、トレース・メッセージを印刷ファイルに作成します。

このキーワードは、TRACERQ= と指定することもできます。

## 警告パラメーター

構造が満杯に近づくとき警告メッセージが出される時点のしきい値を変更するには、これらのパラメーターを使用します。

## ENTRYWARN={80|number}

警告メッセージの発行が最初に行われる時点の、使用中のリスト構造項目のパーセンテージを指定します。

有効な範囲は 1 パーセントから 100 パーセントまでです。

## ENTRYWARNINC={5|number}

次の警告メッセージが発行されるまでの使用中の項目のパーセンテージの増加 (または減少) (次の増加で 100 に達する場合は 1 に減らす) を指定します。最初の警告の後に発されるメッセージは、エレメントの数が増加または減少すると発行されます。これらのメッセージは、使用中の項目が、最初の警告レベルよりこのパーセンテージ分以上減ると出されなくなります。

有効な範囲は 1 パーセントから 100 パーセントまでです。

## 名前付きカウンター・サーバー領域の制御

名前付きカウンター・サーバーを制御するためのコマンドを実行できます。そのためには、**MVS MODIFY (F)** コマンドを使用し、サーバー領域のジョブ名または開始タスク名を指定し、その後にサーバー・コマンドを入力します。

## このタスクについて

簡易書式の F を使用した場合の名前付きカウンター・サーバー・コマンドの一般的な形式は、以下のようになります。

```
F server_job_name,command
parameters... comments
```

## SET keyword=operand[,keyword=operand,...]

1 つ以上のサーバーパラメーター値を変更します。

## 手順

- 1 つ以上のサーバーパラメーター値を変更する場合は、**SET** コマンドを使用します。  
**MVS SET** コマンドと同じく、**T** という省略形を使用することもできます。詳しくは、[227 ページの『SET コマンドのオプション』](#)を参照してください。
- 1 つ以上のパラメーター値や統計の要約情報をコンソールに表示する場合は、**DISPLAY** コマンドを使用します。  
構文は次のとおりです。

```
DISPLAY keyword[=operand][,keyword[=operand],...]
```

**DISPLAY** の有効なキーワードとして、すべての初期設定パラメーターと、[227 ページの『DISPLAY および PRINT コマンド・オプション』](#)に記されている一連のパラメーターを使用できます。**MVS DISPLAY** コマンドと同じく、**D** という省略形を使用することもできます。

- パラメーター値をファイルに出力する場合は、**PRINT** コマンドを使用します。  
このコマンドでは、**DISPLAY** コマンドと同じキーワードを使用して、同じ出力を生成することができますが、出力先は印刷ファイルのみになります。
- サーバーを正常に停止する場合は、**STOP** コマンドを使用します。  
サーバーは、アクティブな接続がまず終了するのを待ちながら、新しい接続が確立されないようにします。**P** という省略形を使用することもできます。さらに、**MVS STOP** コマンドを使用することも可能です。その場合は、**MVS MODIFY** コマンドでサーバー・コマンド **STOP** を実行するのと同じ結果になります。**STOP** コマンドの構文は以下のとおりです。

```
STOP|P [jobname.]identifier[,A=asid]
```

- サーバーをすぐに終了する場合は、**CANCEL** コマンドを使用します。  
**RESTART** オプションによってサーバーの自動リスタートを実行するかどうかを指定することもできます。**CANCEL RESTART** については、[178 ページの『CANCEL コマンドのオプション』](#)を参照してください。
- サーバーは、構造サイズを変更するオペレーター・コマンド **SETXCF** のような XES イベントにも応答します。  
サーバーは、カップリング・ファシリティにアクセスできなくなると、サーバー **CANCEL** コマンドを自動的に発行して、すぐにサーバー自体をクローズします。

### SET コマンドのオプション

**SET** コマンドを使用して、サーバー初期設定パラメーターのグループを変更できます。

サーバー初期設定パラメーターのグループは、以下のとおりです。

- 統計パラメーター
- デバッグ・トレース・パラメーター
- 警告パラメーター

これらのキーワードの詳細については、[223 ページの『名前付きカウンター・サーバーのパラメーター』](#)を参照してください。

**SET コマンドの例:** 次の例は、統計オプションを変更します。

```
SET STATSOPT=BOTH,EOD=21:00,STATSINT=06:00
```

### DISPLAY および PRINT コマンド・オプション

**DISPLAY** コマンドおよび **PRINT** コマンドを使用して、初期設定パラメーターの値および追加情報を表示できます。

**DISPLAY** コマンドおよび **PRINT** コマンドでサポートされるパラメーターは、次のとおりです。

#### ARMREGISTERED

ARM 登録が正常に完了したかどうかを示します (YES または NO)。

#### CONNECTIONS

コマンドの発行先のサーバーに現在接続されている領域のジョブ名および APPLID を表示します。

このキーワードは、**CONN** と省略することができます。

#### COUNTERS

プールに現在割り振られているすべての名前付きカウンターの名前を表示します。



**COUNTERS={name|generic\_name}**

特定の名前付きカウンター、または名前が総称名に一致する名前付きカウンターのセットについての詳細を表示します。総称名は、ワイルドカード文字 \* (アスタリスク記号) と % (パーセント記号) を使用して指定します。

名前付きカウンターが指定されていない場合、これは DISPLAY COUNTERS と同等に扱われます。

このキーワードは、**COUNTER** と省略することができます。

**統計の要約の DISPLAY オプションおよび PRINT オプション**

以下のパラメーターを使用して、統計を表示または印刷します。

**CFSTATS**

カップリング・ファシリティ・インターフェースのアクセスやサーバーからの 応答の統計を表示します。

このキーワードは、**CFST** または **STATSCF** と指定することもできます。

**POOLSTATS**

プール・リスト構造全体の使用状況の統計を表示します。この統計は、カップリング・ファシリティ・アクセス要求によって戻される情報に基づいています。そのため、コマンドの発行先のサーバーを介して実行された最新の要求の時点の統計に過ぎません。

このキーワードは、**POOLST** と省略することができます。

**STORAGESTATS**

サーバー・アドレス・スペースの主記憶域割り振りの統計を表示します。

このキーワード **STORAGEST** または **STGST** と省略することができます。

**情報の結合リストでの DISPLAY および PRINT オプション**

以下のキーワードは、情報の結合リストを表します。

**PARAMETERS**

以下の主要なパラメーター値を表示します。

- POOLNAME
- STATSOPT
- ENDOFDAY
- STATSINTERVAL
- POOLSIZE

このキーワードは、**PARM** または **PARMS** と省略することができます。

**ALLPARAMETERS**

PARAMETERS にリストされるすべてのパラメーター値に加えて、以下の値を表示します。

- CFTRACE
- RQTRACE
- ENTRYWARN
- ENTRYWARNINC

このキーワードは、**ALLPARMS** と省略することができます。

**STATISTICS**

使用可能なすべての統計を表示します。このキーワードは **STAT** または **STATS** という省略形にすることができます。

**INITIALIZED**

初期設定の完了時に通常表示されるパラメーターおよび統計 (PARAMETERS にリストされます) に加えて、以下の値を表示します。

- POOLSTATS
- STGSTATS

このキーワードは、**INIT** と省略することができます。

## ARM

すべての ARM 関連のパラメーター値を表示します。

- ARMELEMENTNAME
- ARMELEMENTTYPE
- ARMREGISTERED

このキーワードは、ARMSTATUS としてコーディングできます。

## CANCEL コマンドのオプション

CANCEL コマンドを使用して、自動再始動を要求できます。

以下のパラメーターを指定します。

### RESTART={NO|YES}

自動再始動を要求するかどうかを指定して、サーバーをすぐに強制終了します。デフォルト値は RESTART=NO です。

サーバーでカップリング・ファシリティの接続に関するリカバリー不能な問題が起きて、接続が失われたり、ストラクチャーで障害が発生したりすると、サーバーは、CANCEL RESTART=YES コマンドを使用して自身をキャンセルします。その結果、既存の接続が終了して、サーバーがシャットダウンします。その後、サーバー・ジョブの新しいインスタンスが開始されます。

サーバー・コマンド CANCEL RESTART=YES または MVS コマンド CANCEL jobname,ARMRESTART を使用して、サーバーを明示的に再始動することもできます。

RESTART=YES の代わりに RESTART、RESTART=NO の代わりに NORESTART を直接入力することも可能です。

## 名前付きカウンター・プールを削除または空にする

カップリング・ファシリティ・リスト構造を削除する MVS SETXCF コマンドを使用して、名前付きカウンター・プールを削除できます。

### このタスクについて

例えば、次のようになります。

```
SETXCF FORCE,STRUCTURE,STRNAME=DFHNCLS_poolname
```

プールに接続されているサーバーがない場合にのみ、構造を削除できます。そうでない場合、MVS はコマンドを拒否します。

削除されたプールに対してサーバーを始動(または、プールを再ロード)しようとする、その構造は新しい構造として割り振られます。新規に割り振られた構造は、現在アクティブな CFRM ポリシーにより指定されたサイズ属性と位置属性を使用します。

## 名前付きカウンター・プールのサイズの変更

ストラクチャーが満杯になりかけている場合に、現在のプール・サイズが最大値より小さければ、SETXCF START,ALTER コマンドを使用してプール・サイズを増やせます。

### このタスクについて

以下に例を示します。

```
SETXCF START,ALTER,STRNAME=DFHNCLS_poolname,SIZE=size
```

SIZE はキロバイト単位で指定します。

## 名前付きカウンター・プールのアンロードと再ロード

**FUNCTION** パラメーターに UNLOAD および RELOAD のオプションを指定して、名前付きカウンター・プールのコンテンツ全体を順次データ・セットからアンロードしたり、再ロードしたりすることができます。

### このタスクについて

この機能では、例えば以下のことを行えます。

- 計画されたカップリング・ファシリティのメンテナンス時に名前付きカウンター・プールを保存する
- プールを別のカップリング・ファシリティに移動する

### FUNCTION={UNLOAD|RELOAD}

サーバーの初期設定を行う対象の機能を指定します。

#### UNLOAD

**POOLNAME** パラメーターで指定されている名前付きカウンター・プールのコンテンツ全体を順次データ・セットにアンロードします。アンロードの処理が完了すると (正常終了または異常終了)、サーバー・プログラムは終了します。

UNLOAD 機能には、テーブル・プールのアンロード先の順次データ・セットを説明する DDNAME DFHNCUL の DD ステートメントが必要です。アンロード・データ・セットの形式は、次のとおりです。

```
RECFM=F  
LRECL=4096  
BLKSIZE=4096
```

#### RELOAD

直前にアンロードした名前付きカウンター・プールを、**POOLNAME** パラメーターで指定された名前付きカウンター・プールに再ロードします。

RELOAD 機能には、テーブル・プールの再ロード元の順次データ・セットを説明する DDNAME DFHNCRL の DD ステートメントが必要です。

必要な場合は、再ロード中に構造が割り振られ、その際に通常のサーバー始動で使用するのと同じ、構造属性を制御するサーバー・パラメーターを使用できます。再ロード処理では、すでにプールにある名前付きカウンター (例えば、構造が小さすぎたため、構造サイズを大きくするために ALTER を使用した後に再ロード・ジョブを再始動する必要があった場合など) は迂回します。

**注:** アンロードしたプール構造が、最初の割り振り以降の任意の時点で (SETXCF コマンドを使用してサイズを増加し) 動的に変更された場合、再ロードしたプールに増加されたサイズが割り振られるようにしてください。推奨される方法は、現行の CFRM ポリシーの構造の INITSIZE パラメーターを構造のサイズを変更するたびに更新し、更新したポリシーを、SETXCF START ,POLICY コマンドを使用してアクティブにします。あるいは、再ロード JCL の POOLSIZE パラメーターに、必要なプール・サイズを指定することもできます。

**注:** FUNCTION パラメーターを省略すると、サーバー・プログラム は名前付きカウンター・サーバー・アドレス・スペースを初期化します。

UNLOAD および RELOAD 関数では、サーバー・プログラムはリスト構造を排他使用する必要があります。構造が通常のサーバーによって現在使用されている場合、アンロードまたは再ロードの試行は拒否されます。同様に、アンロードまたは再ロードのジョブの進行中に通常のサーバーを始動しようすると、構造への共有アクセスが使用できないため、試行は失敗します。

アンロードまたは再ロードをする際、通常のサーバー・パラメーターをすべて指定することができますが、これらの一部 (統計関連のパラメーターなど) は、アンロードまたは再ロードの処理には適用されないため、無視されます。

スペース不足で再ロードが失敗した場合、結果として表示されるメッセージには再ロードされた名前付きカウンターの数と、失敗した時点までに読み取られたブロックの数が含まれます。これらの値を、元のアンロード・ジョブのメッセージと比較して、未ロードの名前付きカウンターの数を割り出すことができます。

## アンロード JCL の例

以下の例の JCL では、名前付きカウンター・プールをアンロードする方法を示しています。

```
//UNLDNCD1 JOB ...  
//NCUNLOAD EXEC PGM=DFHNCMN CICS named counter server program  
//STEPLIB DD DSN=CICSTS56.CICS.SDFHAUTH,DISP=SHR Authorized library  
//SYSPRINT DD SYSOUT=* Options, messages and statistics  
//DFHNCUL DD DSN=NC1.UNLOADED.POOL, Unloaded named counter pool  
// DISP=(NEW,CATLG),  
// SPACE=(4096,(10000,1000)) Estimated size in 4K blocks  
//SYSIN DD *  
FUNCTION=UNLOAD Function to be performed is UNLOAD  
POOLNAME=PRODNC1 Pool name  
/*
```

図 46. アンロード JCL の例

## 再ロード JCL の例

以下の JCL の例では、名前付きカウンター・プールを再ロードする方法を示しています。

```
//RELDNCD1 JOB ...  
//NCRELOAD EXEC PGM=DFHNCMN CICS named counter server program  
//STEPLIB DD DSN=CICSTS56.CICS.SDFHAUTH,DISP=SHR Authorized library  
//SYSPRINT DD SYSOUT=* Options, messages and statistics  
//DFHNCRL DD DSN=NC1.UNLOADED.POOL,DISP=OLD Unloaded pool  
//SYSIN DD *  
FUNCTION=RELOAD Function to be performed is RELOAD  
POOLNAME=PRODNC1 Pool name  
/*
```

図 47. 再ロード JCL の例

## 名前付きカウンター・プールのリスト構造のダンプ

MVS **DUMP** コマンドを使用して、名前付きカウンター・プールのカップリング・ファシリティー・リスト構造のダンプを取得できます。

### このタスクについて

リスト構造のダンプとフォーマットについては、[問題判別におけるダンプの使用](#)を参照してください。

## カップリング・ファシリティー・サーバーの操作

CICS カップリング・ファシリティー・サーバーには、一時記憶域用、カップリング・ファシリティー・データ・テーブル用、名前付きカウンター用という 3 つのタイプがあります。その 3 つのサーバーで実行できる操作はどれも基本的に同じです。以下の情報は、特に指定がない限り、その 3 種類すべてに当てはまります。

### カップリング・ファシリティー・サーバー・メッセージのモニター

サーバーは実行中にさまざまなメッセージを発行します。その中には、カップリング・ファシリティー構造がいっぱいになったなどの、重大な問題が進展していることを示すものが含まれている可能性があります。

サーバーから発行されるメッセージのタイプを知り、問題の可能性を示すシステム状況メッセージをモニターしていることを確認するのは重要です。

#### サーバー・メッセージ

サーバー・コード自体によって発行されるメッセージは、5 文字のサーバー接頭部 (DFHXQ、DFHCF または DFHNC) で始まります。

これらのメッセージは、以下のグループに分類されます。

- オペレーター・コンソール・システム状況メッセージ。これらは WTO (オペレーター宛メッセージ) によって発行され、宛先コード 2 (オペレーター情報) と 11 (プログラマー情報)、および記述子コード 4 (システム状況) が含まれます。

これらのメッセージには、状況の重要な変更、主にサーバー初期化の始まりと終わりにサーバー終了の始まりと終わり、そして問題についての情報が含まれます。

通常の実行時にこの方法で発行されたサーバー・メッセージは、潜在的に重大な問題を示しており、無視するべきではありません。このプロセスが自動化されている場合、通常の初期化と終了では特定の状況メッセージのリストは無視し、他のサーバー・メッセージを警告として扱うというのが単純なルールです。

これらのメッセージは、サーバー・アドレス・スペース、またはクライアント・アドレス・スペースのいずれかから発行されます。メッセージを要求するサーバー・コードが仮想記憶間モードで実行されている場合、メッセージは、クライアント・アドレス・スペースで WTO を 1 次モードで発行したルーチンに戻されます。これにより、仮想記憶間モードで発行される WTO メッセージに適用される制限を回避できます。例えば、仮想記憶間モードの WTO メッセージは、いずれのジョブ・ログにも表示されません。

- コマンド応答。WTO によって、宛先コード 2 と 11、および記述子コード 5 (即時コマンド応答) を指定して発行されます。

これらのメッセージには、統計の表示など、システムの MODIFY または STOP コマンドを介して発行されたサーバー・コマンドへの応答が含まれます。

- ジョブ・ログ・メッセージ:

通常これらのメッセージには、異常終了や試行されたセキュリティ違反の詳細などの診断情報が含まれます。それらは WTO によって宛先コード 11 (プログラマー情報) を指定して、ジョブ・ログに書き込まれます。

- トレースおよび統計メッセージ:

これらのメッセージは、サーバー SYSPRINT ファイルにのみ書き込まれます。

サーバー・コードによって発行されたすべてのメッセージは、それらがサーバー・アドレス・スペースで実行されたかクライアント・アドレス・スペースから仮想記憶間モードで発行されたかに関係なく、サーバー SYSPRINT ファイルにコピーされます。

## AXM メッセージ

AXM 環境コードは、AXM システム・サービス初期化時に、サーバー・アドレス・スペースおよびクライアント・アドレス・スペースとマスター・アドレス・スペースからオペレーター・メッセージを発行します。

このメッセージは、宛先コード 2 (オペレーター情報) および 11 (プログラマー情報。マスター・アドレス・スペースで実行中は使用されない)、および記述子コード 4 (システム状況) を指定した WTO を使用して発行されます。AXM メッセージ番号は "AXMxxnnnnn" の形式で表され、最初の 5 文字の "AXMxx" はメッセージを発行するモジュールの名前、"nnnn" はメッセージ番号の数値部分、"s" は接尾部文字です。メッセージがルーチンの通知メッセージである場合、接尾部文字は "I" で、メッセージにエラーが示されている場合はそれが省略されます。接尾部文字は、自動ツールがルーチンの通知メッセージとエラー状況を区別する際に使用されます。

サーバー環境から発行された AXM メッセージ (サーバー・ロード・モジュールにリンクされた AXM ランタイム環境ルーチン) は、サーバーの SYSPRINT ファイルにもコピーされます。AXM は、SYSPRINT ファイルに通知メッセージを書き込みます。これらには、ストレージ管理の初期化情報や閉止統計などの情報と、診断目的のサーバー・モジュールの主なプロシージャのエントリー・ポイントが含まれます。

## カップリング・ファシリティのストレージ管理

一時記憶域データ共用プール、カップリング・ファシリティ・データ・テーブル・プール、名前付きカウンター・プールのために CICS で使用するカップリング・ファシリティ・ストラクチャーのタイプは、キー付きのリスト・ストラクチャーです。

カップリング・ファシリティ・リスト・ストラクチャーには、16 バイトのキーが付いた項目を組み込んだ番号付きリストの配列が入ります。それぞれのリストはキー付きファイルのようなものです。各項目には固定の接頭部域があり、その中にキーと他の制御情報が入ります。また、プログラム用の 64 バイトの付

属域もあります。接頭部の後には、最大 128 個の 256 バイト・データ・エレメントのチェーンが続きます。最大データ・サイズは 32K バイトです。

名前付きカウンターの場合は、ストラクチャーに 1 つのリストだけが入ります。各項目は接頭部域だけを使用し、データ・エレメントはありません。

項目接頭部 (項目制御ともいう) とデータ・エレメントのためのストレージは、ストラクチャー内の 2 つのストレージ・プールから割り振られます。どちらのストレージ・プールもストラクチャー内のすべてのリストによって共用されます。

リスト・ストラクチャー内のストレージは、以下の 2 つのタイプに分類できます。

- 固定制御。

固定制御用のストレージは、ストラクチャーの存在している間ずっと同じ固定サイズで事前割り振りされます。このストレージには、ストラクチャーの制御情報が入ります。例えば、データ・バッファ・スペースやリスト・ヘッダーの配列などです (リスト・ヘッダーの個数は、ストラクチャーの作成時に定義したリストの最大数以下になります)。CICS の場合は、CICS の内部用のリストの数になるので、比較的小さな数になります。その値は、**MAXQUEUES** パラメーターか **MAXTABLES** パラメーターで指定します。

固定制御には、ストラクチャー内に存在し得る最大数 (ストラクチャーの最大サイズや項目とエレメントの有効な比率の範囲に基づく最大数) のエレメントと項目と、ストラクチャー内に存在し得る最大数のリストを処理するリスト・ヘッダーの配列を管理するための十分な内部制御域が含まれています。

ストラクチャーの再割り振りなしでリストの最大サイズや最大数を増やすことはできないので、ストラクチャーの最初の割り振り時に十分な大きさの値を慎重に指定する必要があります。ただし、リストの最大サイズや最大数として比較的大きな値を指定すると、固定制御のために大きめのストレージが事前割り振りされるので、所定の量のデータを格納するのに必要なストラクチャーの初期サイズが、拡張のための余地を少なめにした場合よりも、かなり大きくなります。

- 可変制御。

可変制御は、項目制御用のストレージ域とデータ・エレメント用のストレージ域に区分されています (項目制御は項目 1 つにつき 1 つ存在します)。その区分については、動的な調整が可能です。動的な調整では、項目とエレメントの比率が変更され、一方のタイプのストレージが他方のタイプのストレージに変換されます。一方のタイプの可変ストレージが尽きそうになっているのに、他方のタイプのストレージがまだ十分にあれば、その比率を変更する要求が CICS から自動的に実行されます。可変制御用のストレージの合計サイズについても動的な変更が可能です。その場合は、ストラクチャーのサイズが変更されます。そのサイズ変更は、アクティブなカップリング・ファシリティー・リソース管理 (CFRM) ポリシーで定義されているストラクチャー・サイズの範囲内になります (その値は、ストラクチャーの作成時に設定します)。そのストラクチャー・サイズは、システム・オペレーター・コマンド **SETXCF ALTER,SIZE** で変更できます。ストラクチャーの割り振り時に CFRM ポリシーで自動変更オプションを指定していれば、そのオプションに基づいて、ストラクチャー・サイズがオペレーティング・システムによって変更されます。

CICS のプール・ストラクチャーでは通常、キュー項目、データ・テーブル・レコード、名前付きカウンターが、ストラクチャー内でそれぞれ 1 つの項目になり、該当する数のデータ・エレメントがさらに組み込まれます。CICS では、そのほかに内部の制御用として追加の項目が使用されます。現在定義されているキューやデータ・テーブルのインデックスを保守したり、現在使用中のリストや、以前は使用されていたのに今は解放されて再利用が可能になっているリストを追跡管理したりするためです。プール内のすべてのキューやデータ・テーブルが削除された後でも、制御リスト内に使用中の項目が残ることもあります。

内部の制御情報のために必要なストレージの量は、現在のカップリング・ファシリティー・レベル (CFLEVEL) のカップリング・ファシリティー制御コードの機能やパフォーマンスのレベルによって決まります。CFLEVEL が高いほどストレージ要件が大きくなる可能性があります。

CICS のプール・ストラクチャーに必要なストレージ要件を計算する場合は、Web ベースの IBM CFSizer ツールを使用すると便利です。[CFSizer](#) を参照してください。このツールは、以下のような流れで処理を実行します。

1. ストラクチャー内に格納する情報の量を示すパラメーター値を入力するためのプロンプトを出します。
2. その情報をリストや項目やエレメントの数に変換します。



3. 現在の CFLEVEL のカップリング・ファシリティと通信して、その情報を格納するのに必要なストレージの量を計算します。指定量のフリー・スペースに拡張のための余地を加えた値になります。

すべてのストラクチャー・サイズは、割り振り時にカップリング・ファシリティ・レベル (CFLEVEL) の次のストレージ増分値に切り上げられます。例えば、CFLEVEL 16 の場合、サイズは最も近い 1 MB 単位の値に丸められます。異なる CFLEVEL に対するストレージ増加の詳細については、[z Systems Processor Resource/Systems Manager Planning Guide](#) のカップリング・ファシリティ制御コード・サポートについての情報を参照してください。

異なる CFLEVEL での CPC のサポートおよび各 CFLEVEL の機能についての詳細は、[CF levels](#) を参照してください。

ストレージ要件の計算の詳細については、以下のトピックを参照してください。

- 169 ページの『一時記憶域データ共用に関するストレージの計算』
- 186 ページの『カップリング・ファシリティ・データ・テーブルのストレージの計算』
- 221 ページの『名前付きカウンターのリスト構造ストレージ』

## プール構造の管理

プール構造がいっぱいになると、そのプールを使用するすべてのアプリケーションに深刻な影響を与える可能性があるため、その兆候に注意することは重要です。

### モニター・プール構造の使用レベル

プール構造の現在の使用レベルを表示するには、DISPLAY POOLSTAT サーバー・コマンドを使用します。

DISPLAY POOLSTAT コマンドはメッセージ DFHXQ0432I、DFHF0432I または DFHNC0432I を作成します。これらのメッセージの中で最も重要な情報は、現在の統計インターバルで使用されているリスト、項目、およびエレメントの最大パーセンテージです。名前付きカウンター・サーバーの場合、メッセージ DFHNC0432I には、エントリーの数のみが表示されます。ここには常に 1 つのリストのみが表示され、エレメントは表示されません。

### プール構造の使用状況に関するオペレーター・メッセージ・レポート

項目またはエレメントの使用数がしきい値レベルに達すると、オペレーターへのメッセージ (例えば DFHXQ0411I や DFHXQ0412I など) が発行されます。

プールが満杯になると、さらにメッセージが発行されます。必要に応じて、これらのメッセージを監視し、そのような状況になった場合にオペレーターに警告したり、修正アクションを実行したりするように自動操作プロセスをセットアップすることができます。

構造が CFRM ポリシーで定義されている最大値に達していない場合、SIZE オプションを増やして MVS オペレーター・コマンド SETXCF ALTER,START を使用することで、構造を拡張することができます。

### CFRM 自動 ALTER を使用したプール構造サイズの増大

カップリング・ファシリティ・リソース管理 (CFRM) ポリシーでは、キーワード ALLOWAUTOALT(YES) を指定できます。

これにより、オペレーティング・システムは構造が満杯に近づくとき、構造のサイズを増大させるか、エレメント対項目の比率を調整する ALTER コマンドを自動で発行することができます。これが行われるしきい値は、ポリシー内で FULLTHRESHOLD キーワードによって指定されます。デフォルトのしきい値は 80% で、これはサーバー自体が自動 ALTER コマンドを発行して項目対エレメントの比率を最適化するデフォルトのしきい値と同じです。サーバーの自動 ALTER プロセスの方が、現在の使用量だけではなく、現在の統計インターバル内での構造のピーク使用量を考慮するため、より高機能です。そのため、サーバーの ENTRYWARN および ELEMENTWARN パラメーターのパーセンテージの値を CFRM FULLTHRESHOLD キーワードのパーセンテージの値よりも 5% 以上小さくしておくことで、サーバーの自動 ALTER プロセスが先にトリガーされるようにすることが最良です。

## システム管理の再ビルドを使用したプール構造サイズの増大

構造が、残された項目やエレメントの数は少ないが、すでに最大サイズに達している場合、その構造を使用しているすべてのシステムがこの機能をサポートするレベルにあれば、システム管理の再ビルドを使用して、サーバーを閉じることなく動的に増大させることができます。

まず CFRM ポリシーを更新してサイズを必要な値まで増やし、次に SETXCF START,POLICY を使用して更新したポリシーをアクティブ化します。その後、構造を再ビルドできます。再ビルドすると、更新されたポリシーを使用する構造の新しいインスタンスが割り振られ、既存のデータがコピーされ、古いインスタンスは破棄されます。

## データ・リストの数の増大

MAXQUEUES または MAXTABLES サーバー・パラメーターによって指定されたデータ・リストの数が小さい場合、新しいデータ・リストを割り振ろうとすると、メッセージ DFHXQ0443 またはメッセージ DFHCF0443 で失敗します。

リストの数を増やすには、構造を削除して再作成するしか方法がなく、すべてのサーバーを一時的に閉じる必要があります。システム管理の再ビルドを使用すると、既存の構造からデータ・リストの数をコピーするため、この数を増やすことはできません。

サーバー・プログラムを使用して順次ファイルにアンロードし、その後 SETXCF FORCE を使用して既存の構造を削除すれば、既存のデータを保持することができます。その後、再びサーバー・プログラムを使用してデータを再ロードし、適切な MAXQUEUES または MAXTABLES パラメーターを使用して新しい構造を割り振ることができます。

## プール構造を削除または空にする

プール構造が不要になった、あるいは構造内のすべてのデータを破棄したい場合は、対象のプールのサーバーをすべて閉じてから SETXCF FORCE コマンドを実行して構造を削除することで、プールを削除できます。

その後サーバーが同じ構造名で再び始動された場合、アクティブ CFRM ポリシーおよびサーバー初期設定パラメーターの情報を使用して空の構造が作成されます。

## サーバー接続管理

クライアント CICS 領域がサーバーへの仮想記憶間接続を確立するのは、そのサーバーのプールが要求で初めて参照されたときです。プールでアクセスするテーブル、キュー、カウンターの数に関係なく、サーバーごとに 1 つの接続が確立されます。

カップリング・ファシリティ・データ・テーブル・サーバーおよび一時記憶域キュー・サーバーの場合は、マルチスレッドの非同期接続が確立されます。この接続では、固定の最大同時要求数まで要求をオーバーラップできます。この最大数を超える要求は、要求スレッドが使用可能になるまで、CICS 領域内のキューに入れられます。

名前付きカウンターの場合は、要求は同期的に処理されるため、特定のクライアント領域 TCB でアクティブにできる要求は一度に 1 つのみです。

カップリング・ファシリティ接続の障害の対処方法については、[z/OS MVS シスプレックスのセットアップ](#) のカップリング・ファシリティ接続の障害の対処に関するセクションを参照してください。

## サーバー接続の終了

各接続には、クライアント・サイドとサーバー・サイドがあります。いずれかのサイドが個別に終了した場合、他方のサイドがしばらく終了しないにも関わらず、その接続は使用可能ではなくなります。

QUASIRENT アプリケーションの場合、接続は CICS 準再入可能 (QR) TCB に関連付けられているため、接続は通常 QR TCB が終了して CICS がクローズするまでアクティブのままです。スレッド・セーフ・アプリケーションの場合、接続はオープン TCB に関連付けられているため、オープン TCB が終了するまでオープンのままです。接続は、TCB のリソース管理の終了処理の一部として自動的にクローズされます。リソース管理の終了中に、クライアント・サイドの接続終了ルーチンは、サーバーに対して仮想記憶間呼び出しを発行して、サーバー・サイドでも接続を終了します。

サーバーを使用しているすべての CICS 領域が終了した後にそのサーバーをクローズする場合 (例えばシステムのクローズの準備をしている場合など)、まずサーバーを静止してから、サーバーの STOP コマンド (または同等の MVS STOP コマンド) を使用して終了します。STOP コマンドを使用すると、新しい CICS 領

域がサーバーに接続するのを防ぐことができ、サーバーを使用しているすべての CICS 領域が終了するとすぐにサーバーを終了することができます。

CICS 領域がまだ接続された状態でサーバーをただちにクローズする必要がある場合、サーバーの CANCEL コマンドを使用します。通常のサーバーの STOP コマンド (または MVS の STOP コマンド) では、接続が終了するまで有効にならないためです。MVS の CANCEL コマンドを使用してサーバーを終了することもできますが、そうするとサーバーの通常の終了処理ができなくなります。

CICS は、クライアント・サイドの接続を自動で終了することはできない点にご注意ください。CICS から、サーバーを閉じる必要があるということを認識する方法はありません。接続により CICS からサーバーに仮想記憶間呼び出しを行うことはできるものの、サーバーの非同期イベントについてサーバーから CICS に通知する手段がないためです。現在 CICS には、接続を要求時に終了する方法はありませんが、名前付きカウンター・サーバー CALL インターフェースには、これを目的とした FINISH 機能があります。ただし、この機能は主にバッチ処理用です。

サーバーを CANCEL で終了すると、各接続のサーバー・サイドはただちに終了しますが、クライアント・サイドはその影響を受けません。元の接続を使用する CICS からの次の要求は失敗し、代わりに CICS は新しい接続の確立を試行します。これは、サーバーが再始動した場合にのみ成功します。ただし、CICS は古い接続を明示的にクローズすることはないため、最終的にクローズしたときに、アクティブな接続の終了に関するメッセージだけでなく、同じサーバーへの以前の接続の終了に関するメッセージも生成されます。

### 失敗したサーバー接続

CICS が準再入可能 TCB の通常のリソース管理終了処理を経ずに (例えば、FORCE コマンドやシステム完了コード 40D によって) 突然終了した場合、メモリーの終わりのリソース・マネージャー・ルーチンにより、接続のクライアント・サイドがクリーンアップされます。

このルーチンは CICS 領域自体では実行されないため、仮想記憶間接続を使用してサーバーに通知することはできません。そのため、CICS が準再入可能 TCB の通常のリソース・マネージャーの終了処理を実行せずに終了した場合、接続のサーバー・サイドはアクティブのままです。カップリング・ファシリティ・データ・テーブルの場合は、これにより問題が発生する可能性があります。これは、APPLID がすでに使用されている場合に、元の CICS 領域が再始動後に再同期することをサーバーが許可しないためです。

サーバーは各接続におけるクライアントの状況を随時確認し、クライアントがすでに終了していることが見つかったら、接続のサーバー・サイドもクリーンアップします。この確認は 1 分ごとに行われ、さらに新しい接続が確立されるたびにトリガーされます。そのため、失敗した接続があれば、通常は新しい接続の試行が再同期される前にクリーンアップされます。接続のサーバー・サイドを終了するクリーンアップ処理は非同期で、1 秒または 2 秒ほどかかることがあるため、クリーンアップ処理が元の CICS 領域からの再同期に間に合わず、すぐに成功しない可能性があります。再同期処理が 1 回目の試行で成功しない場合、次の試行では成功します。

### サーバーの再始動

3 種類の CICS データ共用サーバー (一時記憶域、カップリング・ファシリティ・データ・テーブル、および名前付きカウンター) はすべて、自動リスタート・マネージャー (ARM) のサービスを使用する自動リスタートをサポートします。

サーバーには、初期接続試行が失敗した場合に、イベント通知機能 (ENF) 出口を使用して、カップリング・ファシリティ構造が使用可能になるまで始動時に待機する機能もあります。

サーバーは通常、始動時に ARM に登録を行うため、失敗した場合であっても、インストール ARM ポリシー内のすべての規則に従って自動的に再始動します。ARM 登録が、ARM が使用不可であること以外の理由で失敗する場合、サーバーは始動できません。ARM が使用不可の場合、サーバーは正常に始動しますが、失敗した場合には手で再始動する必要があります。

サーバーは、ARM 結合データ・セットが現在の MVS システム用にフォーマット設定されていないことを示す ARM 戻りコードを認識します。これは、ARM が使用不可であることと同じことを意味します。

サーバーは、登録が戻りコード 8 以上で失敗した場合には始動しません。

サーバーが始動する際に、構造障害などの環境エラーにより構造に接続できない場合、イベント通知機能 (ENF) を使用して構造に関連するイベントを監視して、構造が使用可能になるまで自動で待機します。この待機は仮想記憶間インターフェースが使用可能になる前に発生するため、この時点ではサーバーはクラ

クライアント領域には表示されず、使用不可の状態になっています。待機の間にはサーバーが必要なくなった場合、MVS CANCEL コマンドを使用してそのサーバーをキャンセルできます。

サーバーが正常に実行中であっても、カップリング・ファシリティ・インターフェースが接続障害または構造障害を報告する場合、サーバーは即時に終了します。これによりサーバーはカップリング・ファシリティから切断され、クライアント領域からの現行の仮想記憶間接続のサーバー・サイドは終了します。サーバーは ARM によって正常に即時に再始動されますが、カップリング・ファシリティ構造が(場合によっては、新しい空の構造のインスタンスとして)再び使用可能になるまで、クライアント領域で引き続き使用することはできません。

電源障害などによるカップリング・ファシリティの突然の障害は、構造に障害が発生した場合であっても、接続が失われたと表示されることがあります。これは、このようなケースでは、オペレーティング・システムが構造の状態を判別できないためです。これにより、オペレーティング・システムが既存の構造の状況を判別できるまで、例えば、障害が発生したカップリング・ファシリティが正常に再始動するまで、新しい構造が割り振られるのを防ぐことができます。古い構造が失われたことが事実であっても、そのことをシステムがまだ認識していない場合、オペレーターは古い構造を削除する SETXCF FORCE コマンドを発行して時間を節約することができ、それによりシステムは先行して別のカップリング・ファシリティで同じ構造のインスタンスを作成することができます。

カップリング・ファシリティ・サーバーの自動再始動について詳しくは、[CICS データ共用サーバーの自動リスタート](#)を参照してください。

## システム管理プロセスの CICS サーバー・サポート

一時記憶域データ共用、カップリング・ファシリティ・データ・テーブル、および名前付きカウンター  
の3つのサーバーは、カップリング・ファシリティ・リスト構造のためのシステム管理プロセスをサポートします。

それらのシステム管理プロセスを次に示します。

- [237 ページの『システム管理リスト構造の再作成』](#)
- [238 ページの『システム管理リスト構造の二重化』](#)

## システム管理リスト構造の再作成

システム管理の再作成を使用すると、z/OS が共有一時記憶域、カップリング・ファシリティのデータ・テーブル、および名前付きカウンター・サーバー・プールなどに使用するカップリング・ファシリティ構造の移動を、それらを使用する CICS システムをリサイクルせずに管理できます。

以前は、構造を移動するには、サーバー機能を使用して順次データ・セットを UNLOAD し、データ・セットとは別の場所に RELOAD していました。しかし、この切り替えにより CICS にエラーが発生するため、再始動が推奨されていたものの、その間のシステム停止が許容できないような状況もありました。

システム管理再作成は、カップリング・ファシリティのリスト構造のコンテンツを新しい場所に作成します。再作成を行う際の影響は、再作成の処理に必要な数秒から数十秒程度の間、MVS 内での要求が一時的に中断することのみです。システム管理再作成処理は、構造を1つずつ再作成します。そのため、元の構造が失われたり損傷したような場合には適していません。これは、計画された保守に便利で、1つ以上のシステムにおいて構造への接続が失われたが、少なくとも1つのシステムは元の構造へのアクセスを保っている場合のリカバリーに使用されます。接続が失われた場合、システムは、ポリシーで指定した接続オプションに関わらずシステム管理再作成を自動的に開始しませんが、オペレーター・コマンドを使用して再作成を要求することができます。

システム管理再作成中は、すべての未処理要求が待機状態になります。システム管理の再作成の発生時にアプリケーションが認識する異常は、時間遅延のみです。システム管理再作成について詳しくは、[z/OS MVS プログラミング: シスプレックス・サービス ガイド](#)を参照してください。

CICS は MVS システム管理再作成の機能をサポートしますが、IXLCONN マクロの SUSPEND=FAIL オプションが有効である必要があります。CICS サーバーは、このサポートが使用可能かどうかを自動的に検出します。

- SUSPEND=FAIL のサポートが使用可能な場合、サーバーは IXLCONN マクロの ALLOWAUTO=YES オプションを指定して、そのリスト構造に接続します。

- SUSPEND=FAIL が使用可能ではない場合、サーバーは ALLOWAUTO=NO で接続し、システム管理再作成機能は使用されません。

サーバーがアクティブの場合、MVS オペレーター DISPLAY XCF,STR コマンドを使用して、接続の詳細をメッセージ IXC360I に表示できます。

例えば、構造 DFHXQLS\_PRODTSQ1 の接続の詳細を見るには、以下のコマンドを使用します。

```
Display XCF,STR,STRNAME=DFHXQLS_PRODTSQ1,CONNAME=ALL
```

カップリング・ファシリティのデータ・テーブル構造 DFHCFLS\_DTPPOOL1 の接続の詳細を見るには、以下のコマンドを使用します。

```
Display XCF,STR,STRNAME=DFHCFLS_DTPPOOL1,CONNAME=ALL
```

名前付きカウンター構造 DFHNCLS\_PRODNC1 の接続の詳細を見るには、以下のコマンドを使用します。

```
Display XCF,STR,STRNAME=DFHNCLS_PRODNC1,CONNAME=ALL
```

これらいずれの開始コマンドの場合も、結果として表示される IXC360I メッセージ出力には、その接続でシステム管理再作成が使用可能であることを示す、以下の行 (CONNECTION 情報のセクション) が含まれます。

```
ALLOW AUTO      : YES
SUSPEND         : FAIL
```

## TS データ共用サーバーと CFDT サーバー

一時記憶域データ共用とカップリング・ファシリティ・データ・テーブルの場合、システム管理の再作成中には、未処理要求が待機させられます。システム管理の再作成の発生時にアプリケーションが認識する異常は、時間遅延のみです。

## タイムアウトの考慮事項

システム管理の再作成時の待機時間は、数秒 (構造が小さい場合) から数十秒 (構造が大きい場合) までの幅があると予想されます。したがって、再作成中に待機している間に、DTIMOUT やオペレーター・コマンドによってトランザクションがパージされる可能性があります。

同期点処理中のタイムアウトにより不要な問題が発生するリスクを最小限にするには、CFDT 作業単位制御機能の CICS 待機出口で、待機がパージ不能であることを指定します。

## 名前付きカウンター・サーバー

CALL インターフェースを使用して名前付きカウンター・サーバーに対して発行された要求は、再作成中は待機させられませんが、代わりにこのサーバーは、NC\_ENVIRONMENT\_ERROR カテゴリー下の値 311 の新しい環境エラー戻りコードを返します。

再作成中の EXEC インターフェース 要求は、タイマー待機と再試行ループを使用して待機します。この待機はオペレーター・パージ・コマンドによって中断できますが、DTIMOUT は使用できません。この待機はどの形式のデッドロックによるものでもないが無期限に認識されるからです。

名前付きカウンターのクライアント領域のインターフェース・モジュール DFHNCF は、通常リンク・リスト・ライブラリー内にあり、z/OS イメージ内で使用中の最高レベルの CICS TS のものである必要があります。

## システム管理リスト構造の二重化

システム管理のカップリング・ファシリティの二重化は、ハードウェア補助による汎用のカップリング・ファシリティ構造データの二重化のメカニズムを提供します。

これは、単一構造またはカップリング・ファシリティの損失や接続の損失などといった障害から、二重化のペアの他方の構造に素早く切り替えることでリカバリーする堅固なメカニズムです。

1 つの構造から別の構造にのみ再ビルドでき、元の構造が失われているか破損している場合には利用できないシステム管理の再ビルドとは異なり、システム管理の二重化では、障害が発生する前に、構造の二重のコピーを作成して保守します。構造の大部分はユーザーに認識されることがなく、カップリング・ファシ



リティーまたは構造の計画および計画外の停止時に利用できます。二重化の前は、カップリング・ファシリティー・データ・テーブルおよび一時データ共有構造には、リカバリーでは重要ではないスクラッチパッドの情報ばかりが含まれていました。二重化では、カップリング・ファシリティーを使用して、より重要な情報を保管することができます。

二重化構造にアクセスするトランザクションには遅延が発生する可能性があります、次のような場合、DTIMOUT しきい値に到達する結果につながる可能性があります。

- 二重化が確立されている間に、構造が MVS によって 静止状態になっている。
- 二重化を停止または開始するオペレーター・コマンドの結果、構造が静止状態になっている。
- 構造が 2 次構造に切り替わった、または 1 次構造に戻った。

新しく開始されたデータ共有サーバーは、二重化の再ビルドのフェーズ中は、確立フェーズに達するまで構造に接続できません。

システム管理の二重化は、システム管理の再ビルド機能に基づいてビルドされています。また、構造の CFRM ポリシー情報で、DUPLEX オプションを DUPLEX(ENABLED) または DUPLEX(ALLOWED) として指定する必要があります。DUPLEX(ENABLED) を指定した場合、MVS はユーザー管理の構造の二重化操作を開始し、維持を試行します。DUPLEX(ALLOWED) を指定した場合、オペレーターは SETXCF START,REBUILD,DUPLEX コマンドを仕様して二重化を開始できます。また、二重化操作の一部であるカップリング・ファシリティーが、相互通信に「ピア・リンク」を使用していることを確認する必要があります。

システム管理の二重化について詳しくは、[z/OS MVS プログラミング: シスプレックス・サービス ガイド](#)および [z/OS MVS シスプレックスのセットアップ](#)を参照してください。





## 第 6 章 リソースの定義

CICS をインストールした後で、ユーザーのトランザクションを実行するために CICS 領域が必要とするすべてのリソースを定義してインストールする必要があります。

リソースのリストについては、[CICS リソース: リスト、構文、および属性](#)を参照してください。

CICS リソースの定義方法については、[Ways of defining CICS resources](#) を参照してください。

オンライン・リソース定義 (RDO) トランザクション CEDA を使用してリソースを CICS システムに定義する場合、詳細については [CEDA - オンライン・リソース定義](#)を参照してください。

### リソース定義バッチ・ユーティリティ DFHCSDUP

CICS システム定義ユーティリティ・プログラム DFHCSDUP は、オンライン・リソース定義 (RDO) のコンポーネントの 1 つです。DFHCSDUP オフライン・ユーティリティ・プログラムを使用して、CICS が実行中でも非アクティブでも、CICS システム定義 (CSD) ファイルとの間で読み取りおよび書き込みを行うことができます。

**制約事項:** DFHCSDUP ユーティリティは、JCL 上で RLS アクセスを要求する場合でも、CSD を非 RLS モードで開きます。したがって、RLS モードで CICS から CSD にアクセスする場合、DFHCSDUP の実行時に CSD を開くことはできません。この制限の理由は、RLS モードでリカバリー可能ファイルを開くのに必要な機能が DFHCSDUP ユーティリティにはないからです。しかし、CSD がリカバリー不能な場合でもこの制限は適用されます。

#### DFHCSDUP コマンドを使用して実行できる作業

DFHCSDUP を使用して、以下の作業を実行できます。

- CSD ファイル内の名前付きリストの末尾にグループを追加する (ADD コマンド)
- CSD 上の既存のリソース定義の属性を変更する (ALTER コマンド)
- ある CSD ファイルのグループ・リストを別のまたは同じ CSD ファイル内のグループ・リストに追加する (APPEND コマンド)
- あるグループ内のすべてのリソース定義を同じまたは別の CSD ファイル内の別のグループにコピーする (COPY コマンド)
- 単一のリソース定義のあるグループから別のグループにコピーする (COPY コマンド)
- CSD 上の単一リソースまたはリソース・グループを定義する (DEFINE コマンド)
- CSD から単一リソース定義、特定グループ内のすべてのリソース定義、または特定リスト内のすべてのグループ名を削除する (DELETE コマンド)
- CSD からデータを抽出して、処理のためにユーザー・プログラムに渡す (EXTRACT コマンド)
- 新しい CSD ファイルを初期化して、それに CICS 提供のリソース定義を追加する (INITIALIZE コマンド)
- 選択されたリソース定義、グループ、およびリストをリスト表示する (LIST コマンド)
- APAR を処理する。つまり、特定の APAR のメンテナンスを CSD に適用する (PROCESS コマンド)
- CSD ファイル上のリストから単一グループを除去する (REMOVE コマンド)
- すべての IBM 提供およびユーザー定義のグループをスキャンしてリソースを見つける (SCAN コマンド)
- 必要なときに CSD ファイルにサービスを提供する (SERVICE コマンド)
- 1 次 CSD ファイル内の CICS 提供リソース定義を新規リリースの CICS 用にアップグレードする (UPGRADE コマンド)
- ユーザー定義のデフォルト値のセットを使用してリソースを定義する (USERDEFINE コマンド)
- グループおよびリストに対する内部ロックを除去して CSD ファイルを検査する (VERIFY コマンド)

これらの各コマンドについて詳しくは、[リソース管理ユーティリティ DFHCSDUP コマンド](#)を参照してください。

## DFHCSDUP を呼び出す方法

以下の 2 つの方法で DFHCSDUP プログラムを呼び出すことができます。

- バッチ・プログラムとして

DFHCSDUP をバッチ・プログラムとして呼び出すため、[このサンプル・ジョブ](#)を編集して使用することができます。

- バッチ・モードまたは TSO 環境のいずれかで実行しているユーザー・プログラムから

詳しくは、[ユーザー・プログラムからの DFHCSDUP の呼び出し](#)を参照してください。

## DFHCSDUP のセキュリティ

DFHCSDUP ユーティリティは、CSD へのバッチ・アクセスを必要とします。許可ユーザーのみが CSD の更新を許可されていることを確認してください。CSD データ・セット・プロファイルのアクセス・リストを、CICS 領域ユーザー ID および他の許可ユーザーのみに制限します。詳しくは、「[z/OS Security Server RACF セキュリティ管理者のガイド](#)」の『データ・セットの保護』を参照してください。

## CICS Transaction Server for z/OS, バージョン 5 リリース 6 と旧リリースの間での CSD の共用

異なるリリース・レベルの CICS 領域間で CSD を共用して共通のリソース定義を共用できるようにするには、高レベルの領域である CICS Transaction Server for z/OS, バージョン 5 リリース 6 から CSD を更新する必要があります。

### このタスクについて

CICS Transaction Server for z/OS, バージョン 5 リリース 6 では、一部の属性が廃止され、CSD 定義から除去されています。

### 手順

- 廃止された属性を指定する定義で **ALTER** コマンドを使用します。  
これにより、CICS Transaction Server for z/OS, バージョン 5 リリース 6 でこれらの属性が失われることはないため、リソース定義を CICS TS for z/OS, バージョン 5.6 領域から安全に更新できます。
- CICS TS for z/OS, バージョン 5.6 領域と旧リリースの CICS の領域の間で CSD を共用する場合は、CICS TS for z/OS, バージョン 5.6 CSD ユーティリティ DFHCSDUP を使用して、廃止された属性を指定するリソースを更新できます。
  - a) 互換性オプションを使用するかどうかを指定します。EXEC PGM=DFHCSDUP ステートメントで、オプション COMPAT を使用して **PARM** パラメーターを使用します。  
デフォルトは NOCOMPAT です。つまり、廃止された属性は更新できません。(243 ページの図 48 を参照)。
  - b) COMPAT オプションを指定した場合、このリリースでは DFHCSDUP ユーティリティの **EXTRACT** コマンドを使用できません。

### 次のタスク

『アップグレード』の『異なる CICS リリース間での CSD の互換性』で、これらの廃止された属性、および旧リリースとの互換性について説明しています。

## バッチ・プログラムとして DFHCSDUP を起動するためのサンプル・ジョブ

DFHCSDUP をバッチ・プログラムとして呼び出すためのサンプル・ジョブが用意されています。このジョブを使用するには、手順のセクションの説明に従ってジョブを編集する必要があります。

### このタスクについて

243 ページの図 48 は、サンプル・ジョブ・ステートメントを示しています。

```

//CSDJOB JOB accounting info,name,MSGLEVEL=1
//STEP1 EXEC PGM=DFHCSDUP,REGION=0M,
// PARM='CSD(READWRITE),PAGESIZE(60),NOCOMPAT'
//STEPLIB DD DSN=CICSTS56.CICS.SDFHLOAD,DISP=SHR
//DFHCSD DD DISP=SHR,DSN=CICSTS56.CICS.DFHCSD
//SECNDCSD DD UNIT=SYSDA,DISP=SHR,DSN=CICSTS56.CICS.SECNDCSD
//indd DD UNIT=SYSDA,DISP=SHR,DSN=extract.input.dataset
//outdd DD UNIT=SYSDA,DISP=SHR,DSN=extract.output.dataset
//* or
//outdd DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
:
DFHCSDUP commands
/*
//

```

図 48. DFHCSDUP を実行するサンプル・ジョブ

## 手順

以下のように、ユース・ケースに基づいてサンプル・ジョブをカスタマイズします。

1. 適切な REGION サイズと **PARM** パラメーターを指定するように EXEC ステートメントを変更します。

**PARM** パラメーターを使用して、以下のいずれかのオプションを指定します。

### UPPERCASE

DFHCSDUP からの出力をすべて大文字にするように指定します。すべての出力を大/小文字混合 (デフォルト) にする場合、このオプションをコード化しないでください。

### CSD({READWRITE|READONLY})

このバッチ・ジョブから CSD へのアクセスを読み取りおよび書き込みにするか、読み取り専用にするかを指定します。デフォルト値は READWRITE です。

### PAGESIZE(nnnn)

出力リストのページ当たり行数を指定します。nnnn の値は 4 から 9999 です。デフォルト値は 60 です。

### NOCOMPAT または COMPAT

DFHCSDUP ユーティリティー・プログラムを互換モードで実行するかどうか (つまり、CICS Transaction Server for z/OS, バージョン 5 リリース 6 で廃止されている定義を更新できるかどうか) を指定します。デフォルトは NOCOMPAT です。つまり、廃止された属性は更新できません。このオプションについて詳しくは、[CICS Transaction Server for z/OS のバージョン間での CSD の共用](#)を参照してください。

2. APPEND、COPY、または SERVICE コマンドで **FROMCSD** パラメーターを指定する場合は、SECNDCSD DD ステートメントを追加して、2 次 CSD を指定します。  
この DD ステートメントの DD 名は、**FROMCSD** パラメーターで指定する名前です。2 次 CSD は、1 次とは異なるデータ・セットにする必要があります。同じデータ・セットを参照する 1 次および 2 次 DD ステートメントを定義することはできません。
3. EXTRACT コマンドを指定する場合は、以下のいずれかまたは両方のアクションの実行が必要になることがあります。

- a) STEPLIB と USERPROGRAM プログラムが含まれるライブラリーを連結します。
- b) ユーザー・プログラムで定義されている入力データ・セットの DD ステートメントを組み込みます。  
このステートメントは、図では DD 名 *indd* で示されています。  
例えば、提供されているユーザー・プログラム DFH\$CRFA には、DD 名が CRFINPT の DD ステートメントが必要です。

CRFINPT によって指定された入力ファイルは、ユーザー・プログラム DFH\$CRFx (アセンブラーの場合は x=A、PL/I の場合は x=P) と DFH0CRFC (COBOL の場合) で相互参照リストが必要なリソース・タイプまたは属性のリストを指定するために必要です。CEDA で認識されるリソース・タイプを行ごとに 1 タイプずつ (1 桁目から大文字で) 指定できます。例えば、CRFINPT ファイルには、相互参照される、以下のリソース・タイプを (行ごとに 1 つずつ) 含めることができます。

PROGRAM  
TRANSACTION  
TYPETERM  
XTPNAME  
DSNAME

プログラム DFH\$CRF<sub>x</sub> または DFH0CRFC (COBOL の場合) による CRFINPT ファイルの使用に関するプログラミング情報については、[サンプル EXTRACT プログラム](#)を参照してください。

4. EXTRACT コマンドを指定する場合は、抽出プログラムから出力を受け取るデータ・セットの DD ステートメントを組み込みます。図は、2つのオプション (データ・セットをディスクに書き込むか、SYSOUT に書き込むか) を示しています。

DD 名は、ユーザー・プログラムで定義した DD 名です。図では、DD 名として *outdd* を使用しています。CICS で提供されているサンプル・プログラムでは、以下の DD 名の DD ステートメントが必要です。

表 22. 提供されているサンプル・プログラムの DD ステートメント		
プログラム名	DD 名	サンプル DD ステートメント
DFH\$CRF <sub>x</sub> または DFH0CRFC (COBOL)	CRFOUT	//CRFOUT DD SYSOUT=A
DFH\$FOR <sub>x</sub> または DFH0FORC (COBOL)	FOROUT	//FOROUT DD SYSOUT=output.dataset
DFH0CBDC	CBDOUT SYSABOUT	//CBDOUT DD SYSOUT=A //SYSABOUT DD SYSOUT=A

5. これらの例の出力データ・セットは、SYSIN で EXTRACT コマンドが指定されるたびに開かれ、閉じられます。

- 出力を順次ディスク・データ・セットに書き込む場合は、DISP=MOD を指定して、データが連続する EXTRACT コマンドによって上書きされないようにします。
- また、DD ステートメントで SYSOUT を指定しない場合は、プログラムで OPEN ステートメントを (例えば、COBOL 版では OPEN EXTEND に) 変更できます。

CICS で提供されるユーザー・プログラムに関するプログラミング情報については、[サンプル EXTRACT プログラム](#)を参照してください。

6. SYSIN データ・セットでは、各コマンドの説明の構文ボックスで示されているように、省略形や大/小文字混合を使用してコマンドとキーワードをコード化できます。必要に応じて、データ・セットや区分データ・セット・メンバーを入力ストリームにコード化するのではなく、コマンドで使用できます。

行末 (72 桁目) で継続文字 (アスタリスク) を使用すると、キーワード値を 1 行より長くすることができます。後続の行は 1 桁目から始まります。

例えば、この機能を使用して、最大 128 の 16 進文字の XTPNAME 値を指定できます。

あいまいなコマンドまたはキーワードを入力すると、DFHCSDUP プログラムによってあいまいさを示すメッセージが発行されます。

## 内部エラーが検出された後の DFHCSDUP でのコマンド処理

DFHCSDUP プログラムに put-message-exit ルーチンを提供している場合、それはメッセージが発行される時に常に呼び出されます。DFHCSDUP プログラムがユーザー・プログラムから呼び出される時、DFHCSDUP 処理によって生成されるエラー・メッセージには、この出口を使用して応答できます。

DFHCSDUP プログラムがバッチ・プログラムとして実行している場合、put-message-exit ルーチンは使用されません。DFHCSDUP 出口に関するプログラミング情報については、[システム定義ユーティリティ・プログラム \(DFHCSDUP\) 用ユーザー・プログラム](#)を参照してください。

エラー (8 以上の戻りコードがある) に対する DFHCSDUP プログラムの反応は、エラーの性質と DFHCSDUP プログラムが呼び出された方法によって異なります。

DFHCSDUP プログラムがバッチ・プログラムとして実行されている際にエラーが検出されると、以下の 2 つの反応のいずれかが生じます。

1. CSD の接続中にエラーが発生した場合、後続のコマンドは完了しません。
2. その他の場所でエラーが発生した場合、後続のコマンドは LIST コマンド以外には実行されません。

DFHCSDUP プログラムが `get-command` 出口からコマンドを受け取る際にエラーが検出された場合、後続のすべてのコマンドは、可能であれば処理されます。

## 自動インストール

自動インストールとは、CICS リソースを必要に応じて動的に作成してインストールする方式のことをいいます。SNA LU、MVS コンソール、APPC 接続、IPIC 接続、プログラム、マップ・セット、区画セット、ジャーナルで自動インストールを使用できます。

自動インストールを使用すれば、使用する可能性があるすべてのリソースを定義してインストールする必要がなくなります。むしろ、リソースが要求された時点で、CICS によって定義が動的に作成され、インストールされます。CICS が新しい定義を作成するベースになるのは、ユーザーが指定するモデル 定義です。

自動インストール制御プログラムを使用して、CICS 領域での自動インストールの動作を制御できます。

自動インストールを使用すれば、すべてのリソースを定義してインストールする場合に必要な時間を節約できます。さらに、リソースをインストールすれば、リソースを使用するかどうかは別としてその分のストレージが必要になりますが、そのストレージも節約できます。

## 自動インストール・モデル

自動インストールされるタイプのリソースごとに、少なくとも 1 つのモデル・リソース定義を指定する必要があります。

インストール済みの定義を持たないリソースが要求された場合、CICS はモデルで指定した内容に基づいて定義を作成します。自動インストールされたリソースにどのプロパティを使用するかに応じて、モデルを複数持つことができます。例えば、500 の端末をすべて同じプロパティにして、別の 500 を別のプロパティ・セットにした場合、2 つのモデル端末定義 (2 つのプロパティ・セットのそれぞれに 1 つずつ) を持つことが可能です。

## 自動インストール制御プログラム

自動インストールは、ユーザー作成または CICS 提供の自動インストール 制御プログラムを使用して制御します。このプログラムは、CICS に 1 つ以上のモデル名を提供したり、端末での自動インストール用に z/OS Communications Server 情報を CICS に提供したりするなど、さまざまな処理を実行します。

CICS は 自動インストール制御プログラムをいくつか提供し、端末用に 1 つ、MVS コンソール用に 1 つ、接続用に 1 つ、プログラム、マップ・セット、区画セット用に 1 つ提供します。CICS 提供のプログラムを使用することもできますし、インストールに合わせてそれらをカスタマイズすることもできます。

## z/OS Communications Server 端末の自動インストール

z/OS Communications Server 端末および接続で自動インストールを使用する方法を説明します。

MVS コンソールの自動インストールについては、[258 ページの『MVS コンソールの自動インストール』](#)を参照してください。

接続の自動インストールと並列セッションについては、[261 ページの『APPC 接続の自動インストール』](#)を参照してください。

自動インストール制御プログラムに関するプログラミングの情報については、[LU の自動インストールを制御するプログラムの作成](#)を参照してください。



## 自動インストールする端末の決定

このセクションには、自動インストールする端末デバイスの決定をサポートする情報が記載されています。

### 自動トランザクション開始

TCT 項目があるがログオンしていない端末に BMS ROUTE メッセージが送信される場合、CICS は後で送信するためにメッセージを保管します。

さらに、TCT 項目で指定されている場合、CICS は端末を取得して、その目的でログオンしようとします。CICS は同じ方法で他の ATI 要求 (EXEC CICS START、または一時データ・トリガー・レベル) を満たそうとします。

プリンターのほとんどすべてのトランザクションは自動的に開始されるため、プリンターに対する自動インストールの利用は、厳しく制限されています。ただし、ログオンするよう調整している場合は、プリンターを自動インストールすることができます。プリンターの自動インストールは、コンソール・コマンドとして VARY NET,...,LOGON を入力すると実行されます。

自動インストールされた端末では、端末がログオフしている場合でも、TCT 項目を利用できる**可能性があります**。これは端末が CICS の実行の早い段階でログオンしている場合にのみ発生し、端末の最後のユーザーの TYPETERM 定義、システム初期設定パラメーター、および SNT 項目によって異なります。詳しくは、254 ページの『[自動サインオフ、ログオフ、および TCTTE の削除](#)』を参照してください。TCT 項目が存在する場合、自動インストールされた端末は、個別に定義された端末と同じように ATI 要求を受け入れることができます。

ATI 要求を受信する可能性がある端末の自動インストールを選択する場合、TYPETERM 定義の AUTOCONNECT 属性をモデルに対して利用します。AUTOCONNECT(YES) は、z/OS Communications Server SIMLOGON (シミュレート・ログオン) を要求する CICS によって、端末が緊急時再始動時に自動的に CICS にログオンすることを意味します (257 ページの図 52 を参照)。(これは TCT 項目を必要とするため、コールド・スタートまたはウォーム・スタートでは発生しません。)

特に多数の小さなプリンターに印刷を配信した場合などは、プリンターが所有する領域をセットアップすることが最善のアプローチであることがあります。

プリンターを自動インストールするかどうかにかかわらず、プリンターと代替プリンターをディスプレイ・デバイスに関連付けることができます。この関連付けは、ディスプレイ・デバイスが自動インストールされるときに行われます。これらのプリンターの定義は、ディスプレイ・デバイスが自動インストールされるときにインストールされている必要はありませんが、利用時には存在している必要があります。

### TCT ユーザー域 (TCTUA)

TCT ユーザー域は、TCT 項目に対するオプションの拡張です。

TCTUA は、アプリケーションで 사용할ことが可能です (TCT 項目のその他の部分は CICS に属します)。これは従来、以下の 2 つの目的で使用されてきました。

- 疑似会話型シーケンスの 1 つのトランザクションから次のトランザクションにデータを渡すため。
- 端末セッションの実行中にユーザー・プロファイル情報と統計を維持するため。(これは、z/OS Communications Server セッションである必要はありませんが、特定のアプリケーションに対する、そのアプリケーションによって定義されたアクセス可能な期間です。)

1 番目の使用法は、COMMAREA や他の CICS 機能に少しずつ置き換えられるようになっていますが、2 番目の使用法は現在でもかなり一般的です。アプリケーションは、出納合計などの統計を TCTUA に保管することができます。TCTUA は、実行の開始時に PLTPI プログラムによって初期化され、終了 (シャットダウン) 時に PLTSD プログラムによって取得されます。ログオフとログオンの間は、TCTUA が存在しないので、この情報を保管する機能は自動インストールによって提供されません。さらに、システムの初期化と終了のときにも、PLTPI および PLTSD プログラムは TCTUA を使用できません。ユーザーがログオンまたはログオフするときに TCTUA およびユーザー・データを初期化できるようにするためには、新しい技法を考案する必要があります。

以前に記載したとおり、自動インストール・プロセスが TCT 項目 (TCTUA を含む) を作成するのは、端末で最初のトランザクションが実行される前ではありますが、自動インストール制御プログラムが初期実行を完了した後です。このように、自動インストール制御プログラムでは TCTUA にアクセスできないので、後で TCTUA の初期設定を行う必要があります。これを行うために、「good morning」トランザクションを独自に作成することも、問題となるアプリケーションの最初のトランザクションを使用することもできます。

さらに、CICS は自動インストール制御プログラムを呼び出す前に TCT 項目 (TCTUA を含む) を削除するので、このプログラムはログオフの際にも TCTUA にアクセスできません。そのため、アプリケーションでそのような TCTUA からの統計や他の情報を取り込む必要がある場合、CICS が上記の処理を実行する前にアクセス権限を取得する必要があります。これは、端末エラー処理ルーチンのユーザー作成コンポーネントである **ノード・エラー・プログラム (NEP)** で行います。CICS は TCT 項目を削除する前に NEP 出口を駆動します。

### 端末リスト・テーブル (TLT)

端末リスト・テーブルは、4 文字の CICS 端末名または 3 文字の CICS オペレーター ID で定義された、端末のリストです。

これは基本的に、メッセージを複数の宛先にルーティングするため、さらに監視プログラムが特定の端末グループに対して行える操作制御を限定するために使用されます。これらの使用方法はどちらも、自動インストール環境では再考する必要があります。TLT の使用時にログオンしていないため TLT 項目を持たない端末が TCT にリストされている場合、それらの端末に対する監視操作は失敗します。例えば、存在しない TCT 項目のサービスを開始したり休止したりすることはできません。

同様に、個別に定義された端末と自動インストールされた端末とでは、メッセージ・ルーティングの方法が異なります。ログオンしていない個別定義の端末にメッセージが送られると、メッセージは一時記憶域に保管されて、端末がログオンしたときにメッセージが配信されます。自動インストールされた端末であるために定義されておらず、しかもログオンしていない端末にメッセージが送られると、CICS はルート失敗状態になります。これはその端末に関する情報がないことを示します。実際のところ、自動インストール環境などのように、端末名がランダムに生成されて割り当てられた場合、TLT メカニズム全体が崩壊します。

### トランザクション・ルーティング

ローカル・システムへの自動インストールの要求は、別のシステムからシッパされた同じ定義の自動インストールの要求をオーバーライドします。

トランザクション・ルーティングが 2 つの CICS システムの間で発生する可能性がある場合は、それらのシステムにログオンできる端末が、同一の方法で両方のシステムにインストールされている必要があります。その場合の端末の条件は次のとおりです。

- 両方のシステムに自動インストールされている、または
- 初期設定時に各システムに定義されている

### 自動インストールと出力専用デバイス

自動インストールの利点のほとんどは、プリンターよりもディスプレイ 装置に多く該当します。

CICS の開始トランザクションを表示します。通常は、トランザクション出力が入力端末に返されるため、CICS もアプリケーションもディスプレイの NETNAME (ログオン処理時にそれ自体を識別する) 以外は認識する必要がありません。

一方、出力が出力専用プリンターに送信される場合、CICS またはアプリケーションは使用する NETNAME を認識する必要があり、これはネットワークの情報の一部がどこかで維持されていることを意味します。この種の情報は、個別に定義された TCT 項目の基本/代替プリンターの名前で構成され、CICS によって維持されます。自動インストールされる端末では、必要な場合は該当する情報がテーブルまたはファイルで維持される必要があります、z/OS Communications Server の ASLTAB および ASLENT モデル端末サポート (MTS)によって提供、またはユーザーによって動的に提供されるアプリケーションに組み込まれています。

## 自動インストールおよび z/OS Communications Server

このトピックでは、自動インストールが z/OS Communications Server でどのように機能するのか説明します。自動インストール利用時に発生する処理の理解をサポートすることが目的です。

### 自動インストールを使用した CICS へのログオンのプロセス

(プロセスについては、249 ページの図 49 および 250 ページの図 50 で説明します。) CICS は、z/OS Communications Server のモデル端末サポート (MTS) 機能をサポートします。MTS を使用すると、z/OS Communications Server テーブルの各端末のモデル名、プリンター (PRINTER)、および代替プリンター (ALTPRINTER) を定義できます。CICS はこの情報をログオン時の自動インストール処理の一部としてキャプチャーし、端末の TCTTE を作成するために使用します。MTS を使用している場合は、CICS Transaction

Server for z/OS での使用に適したバージョンの DFHZATDX を使用する必要があります。ユーザーが置換可能な自動インストール・プログラムのプログラミング情報については、[LU の自動インストールを制御するプログラムの作成](#)を参照してください。

1. z/OS Communications Server が要求を受信して、CICS を使用するかどうかを判断し、CICS に要求を受け渡します。
2. CICS はログオン・データから端末の NETNAME 名を抽出します。CICS は同じ NETNAME を持つエントリーの TCT を検索します。
3. そのようなエントリーが検出されると、CICS は OPNDST を z/OS Communications Server に発行して、CICS と端末の間でセッションを確立します。これは、CICS の正常なログオン・プロセスです。
4. 一致するエントリーを検出できない場合、CICS は SIT で指定された、または CEMT を使用してリセットされたシステム初期設定パラメーターを確認し、自動インストールを許可できるかチェックします。
5. システム初期設定パラメーターが自動インストールを許可する場合、CICS は z/OS Communications Server によって受け渡される端末のデータを確認して、端末が自動インストールに適格かどうかを検査します。
6. 端末が適格である場合、CICS はバインド・イメージを調べ、十分な情報を持っているかどうかを判別します。
7. z/OS Communications Server のバインド・イメージのデータが十分であることがわかると、CICS は昇順にソートされた順序で自動インストール・モデル・テーブル (AMT) を検索して、次のいずれかの方法で端末を自動インストールします。
  - z/OS Communications Server が CICS に有効なモデル名を提供する場合、CICS はこの名前を**自動インストール制御プログラム**に受け渡します。(ログオン要求が z/OS Communications Server から CICS に来た場合、および z/OS Communications Server にモデル端末の名前を提供する場合、CICS はログオン・データからモデル端末の名前を取得できます。)
  - z/OS Communications Server が CICS に有効なモデル名を提供しない場合、CICS は AMT で適切な自動インストール・モデルを検索し、それらを z/OS Communications Server のログオン・データと共に**自動インストール制御プログラム**に受け渡します。z/OS Communications Server が CICS に無効なモデル名を提供した場合は、メッセージ DFHZC6936 が表示されます。
8. 自動インストール制御プログラム (CICS 提供のプログラム、またはユーザーが作成したプログラムのいずれか) はモデルの 1 つを選択し、端末の TCT 項目を完成させるために必要な残りの情報を提供します。
9. 自動インストール制御プログラムが制御を戻すとき、CICS は、端末に対する自動インストール・モデル、自動インストール制御プログラムによって戻されるデータ、および z/OS Communications Server のログオン・データを使用して、TCT 項目をビルドします。CICS は次に TCT に新しい項目を追加し、OPNDST を z/OS Communications Server に発行して、CICS と端末の間でセッションを確立します。
10. TYPETERM 定義で指定されている場合、CICS は QUERY 機能を使用して、端末の機能のいくつかを検索します。(これらの機能については、[TYPETERM リソース](#)に記載しています。)

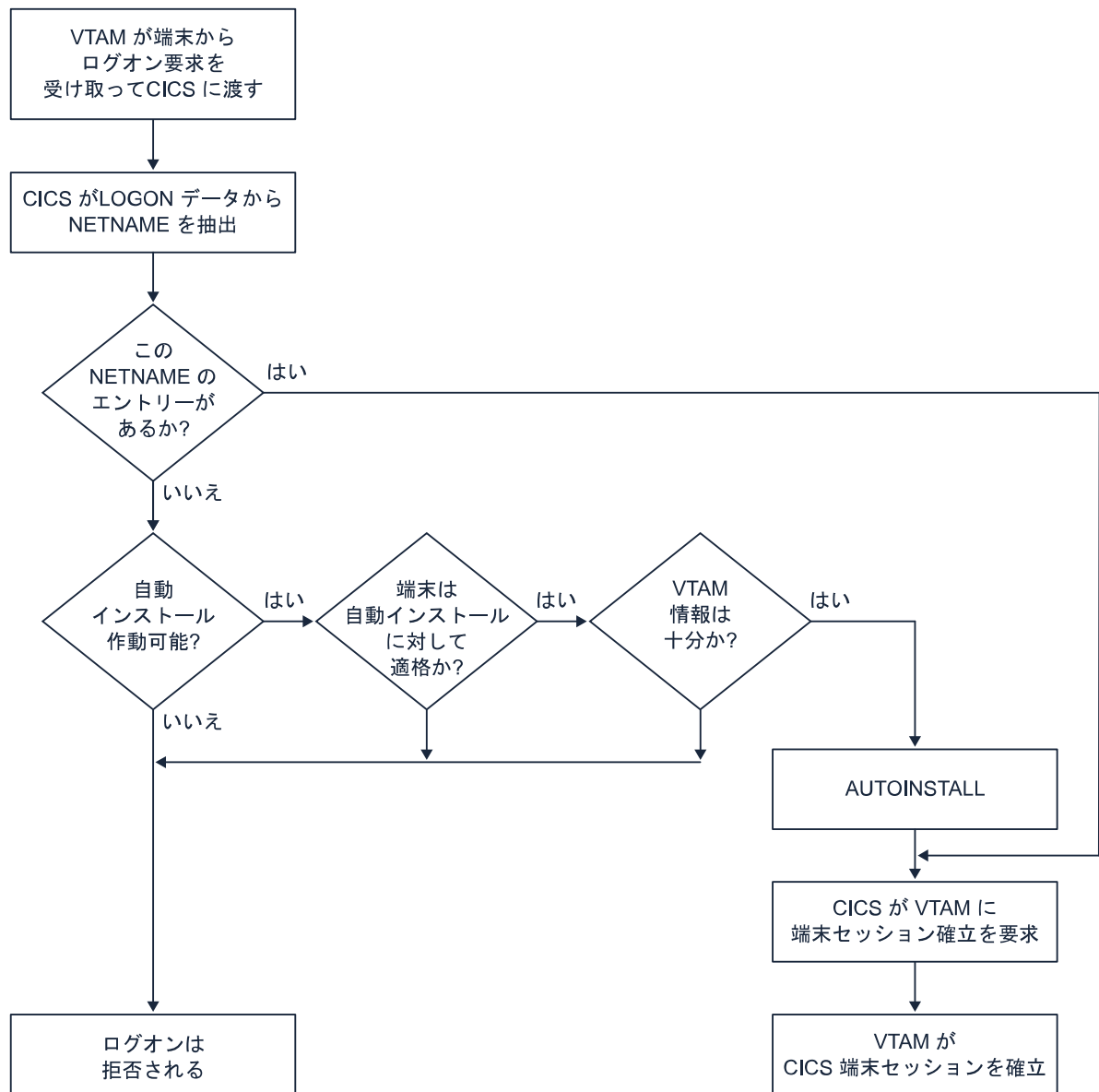


図 49. 自動インストールを使用した CICS へのログオンのプロセス

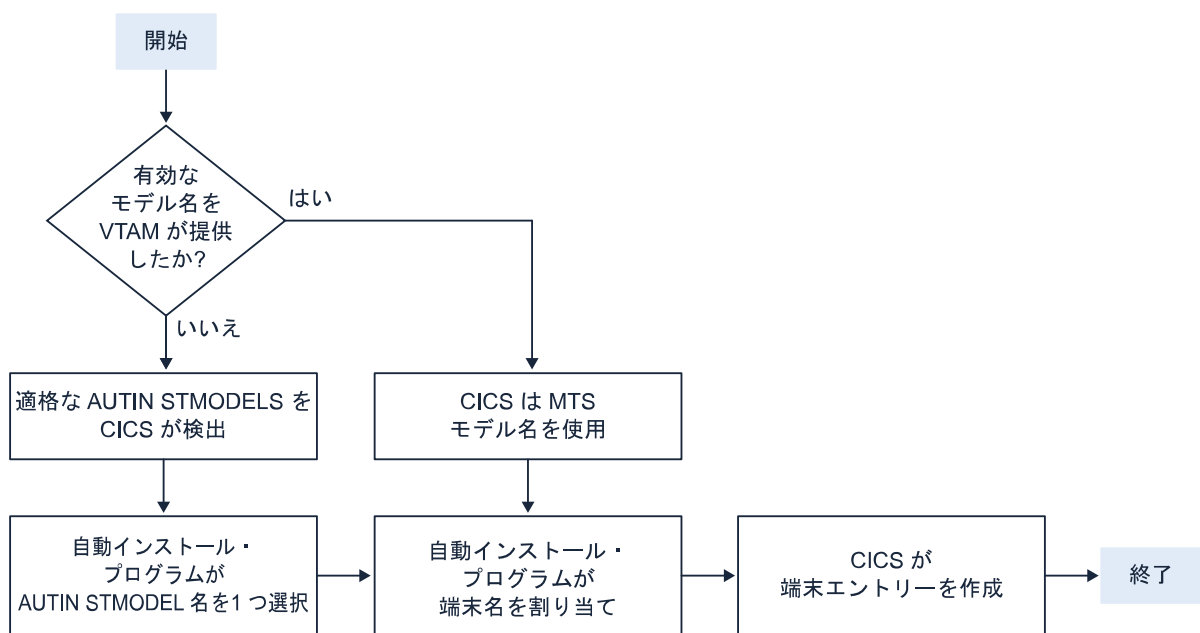


図 50. CICS が端末を自動インストールする方法

## ユーザーのログオフ時に発生すること

1. CICS は CLSDST を発行して z/OS Communications Server にセッションの終了を指示します。
2. CICS は [AILDELAY](#) システム初期設定パラメーターで指定された遅延の後で、TCT 項目を削除しようとします。

TCT 項目は、その時点で有効なロックがある場合は削除できません。例えば、**CEMT INQUIRE TERMINAL** や未解決の ATI 要求は TCT 項目をロックします。

3. CICS は、さらに行う処理がある場合に、自動インストール制御プログラムへのコントロールを提供します。例えば、端末に提供した TERMINAL 名を解放します。

端末の TYPETERM 定義が SIGNOFF(LOGOFF) を指定して、RACF セグメントがタイムアウトを指定する場合、タイムアウト期間の期限が切れると、端末がログオフされ、ユーザーがサインオフされます。この自動ログオフの後で、遅延期間が開始し、この期間の終了までに端末が再度ログオンしない限り、TCT 項目が削除されます。詳しくは、[254 ページの『自動サインオフ、ログオフ、および TCTTE の削除』](#)を参照してください。

## z/OS Communications Server の自動インストールの実装

z/OS Communications Server 端末の自動インストールをセットアップするには、いくつかの手順を実行する必要があります。

### 手順

1. 端末が自動インストールに適格かどうかを判断します。

次の端末を自動インストールできます。

- z/OS Communications Server のローカル接続された 3270 端末 (非 SNA)、ディスプレイとプリンターの両方
- z/OS Communications Server 論理装置タイプ 0 の端末
- z/OS Communications Server 論理装置タイプ 1 の端末、SCS プリンターを含む
- z/OS Communications Server 論理装置タイプ 2 の端末
- z/OS Communications Server 論理装置タイプ 3 の端末
- z/OS Communications Server 論理装置タイプ 4 の端末
- z/OS Communications Server 論理装置タイプ 6.2 の単一セッション端末
- NTO を使用した TLX または TWX 端末

次の端末は自動インストールできません。

- パイプライン端末
- 自動預金支払機 (ATM) (3614 および 3624)
- 非 z/OS Communications Server リソース
- z/OS Communications Server 論理装置タイプ 6.1 ISC および MRO のセッション

2. 自動インストールを利用する利点があるかどうかを判断します。

システムに以下のものがある場合、自動インストールによる利点があると考えられます。

- 大量の z/OS Communications Server 端末
- ネットワークの頻繁な変更
- ほとんどの時間ログオフしている多数の z/OS Communications Server 端末
- ほとんどの時間他のアプリケーションを使用している多数の z/OS Communications Server 端末
- 複数の接続されていない CICS システムにアクセスする必要がある多数の z/OS Communications Server 端末

以下のものがある場合、自動インストールの利点は少ないと考えられます。

- 小規模の静的ネットワーク



- 1つの CICS システムにほとんど永続的にログオンしている多数の端末
  - ログオンとログオフを頻繁に行う多数の端末
  - ログオンとログオフを同時に行う多数の端末
3. 自動インストールするデバイスを決定します。

この決定は、z/OS Communications Server 端末の利用方法に応じて異なります。例えば、常にログオンしている端末は自動インストールできますが、個別に定義することも選択できます。

自動インストールのログオンは、CICS に個別に定義された端末へのログオンよりも遅いため、アプリケーションを継続的に切り替え、CICS に頻繁にログオンする必要がある場合は、いくつかの端末に対して個別の定義が必要になる可能性があります。

また、インストール環境における自動トランザクション開始 (ATI) の利用、端末リスト・テーブル (TLT)、および使用中の相互通信方法についても考慮する必要があります。[246 ページの『自動インストールする端末の決定』](#)を参照してください。

4. TYPETERM とモデル TERMINAL の定義を作成します。

CICS はいくつかの TERMINAL および TYPETERM 定義を提供します。これらはグループ DFHTYPE 内の TYPETERM 定義およびグループ DFHTERM 内のモデル TERMINAL 定義にリストされています。適切であればこれらの定義を利用できます。適切でない場合は、CEDA または DFHCSDUP を使用して独自の定義を作成できます。

自動インストールされる種類の異なる端末ごとに自動インストール・モデルを定義します。定義の数を最小限にして、自動インストールの制御プログラムができるだけシンプルになるようにします。

定義を作成する場合は、QUERY 構造化フィールドを使用するかどうかを検討します (TYPETERM 属性を参照)。これは、自動インストールの制御プログラムが定義の基礎となるモデルを選択する際に役立ち、自動インストール・プロセスを高速化します。

5. DFHZCQ を再定義します。

自動インストールを使用するすべての領域で、DFHZCQ が RESIDENT(YES) になるように再定義します。(DFHZCQ は CICS 提供グループ DFHSPI です。) プログラムの常駐化を検討すべき理由については、[常駐、非常駐、または一時としてのプログラムの定義](#)を参照してください。

6. z/OS Communications Server の LOGMODE テーブルのエントリーが正しいことを確認します。

[247 ページの『自動インストールおよび z/OS Communications Server』](#)では、CICS の自動インストールと z/OS Communications Server の間のリレーションシップについて説明しています。z/OS Communications Server の LOGMODE テーブルのエントリーを含むプログラミング情報については、[Coding entries in the VTAM LOGON mode table](#) を参照してください。

7. 自動インストール制御プログラムを設計して作成します。

端末の自動インストール制御プログラムは、自動インストールされる TCT エントリーに対して有効な要求があるたび、および自動インストールされた TCT エントリーが削除されるたびに、CICS により起動されます。

自動インストール制御プログラムに関するプログラミングの情報については、[LU の自動インストールを制御するプログラムの作成](#)を参照してください。

プログラムを開始する前に、グループ DFHSPI 内の CICS 提供の自動インストール制御プログラム DFHZATDX を確認して、自動インストールを使用して実行したい内容に適しているかチェックします。

8. 端末の自動インストールを使用可能にします。

適切なシステム初期設定パラメーターを指定するか、EXEC CICS または CEMT SET および INQUIRE SYSTEM コマンドを使用して、端末の自動インストールを使用可能にできます。

5 つのシステム初期設定パラメーターが、端末の自動インストールに関連しています。

#### **AIEXIT**

使用される自動インストール・プログラムの名前を指定します。デフォルトで IBM が提供した自動インストール制御プログラムの名前である DFHZATDX に設定されます。

#### **AIQMAX**

自動インストールで同時にキューに入れることができる端末の最大数を指定します。この制限に達すると、自動インストールの要求がもう 1 つ CICS によって処理されるまで、LOGON および BIND

要求の CICS への受け渡しを停止するよう、CICS から z/OS Communications Server に対して要求します。

制限の目的は、その他のいくつかの異常イベントの結果として、自動インストール・プロセスがオペレーティング・システムのストレージを制御不能な状態で消費することからシステムを保護することです。通常、自動インストール・プロセスでは、CICS ストレージを主に消費するのは自動インストール・タスク (CATA) そのものです。正常な操作中に自動インストール・プロセスによって消費される CICS ストレージの量は、適切な TRANCLASS 定義を作成し、同時に存在できる自動インストール・タスクの数を制限することで制御できます。

#### **AILDELAY**

自動インストールの端末がログオフしてから TCTTE が削除されるまでに経過する時間間隔を時間、分、および秒 (hhmmss) 形式で指定します。デフォルト値は 0 で、セッション中の自動インストールの端末に対して CLSDST が発行されるため、ログオフ時およびウォーム・シャットダウン時に TCTTE が削除されることを示します。AILDELAY の間隔を指定すると、間隔の期限が切れる前に端末がもう一度ログオンした場合に、端末が TCTTE を再利用できます。

#### **AIRDELAY**

緊急時再始動の後で、端末がセッション中でない場合に端末入力削除されるまでに経過する時間間隔を時間、分、および秒 (hhmmss) 形式で指定します。デフォルト値は 700 で、再始動の遅延が 7 分であることを示します。

#### **GRPLIST**

作成される自動インストール・モデルのグループを含むリストを指定します。

これらのシステム初期設定パラメーターの指定方法については、[CICS システム初期設定パラメーターの指定を参照してください](#)。

次の 3 つのオプションは、INQUIRE および SET AUTOINSTALL コマンドの端末の自動インストールに関連します。

#### **CUR(value)**

現在処理中の自動インストールのログオン要求の数を指定します。

#### **MAXREQS(value)**

1 回でキューに入れることができる自動インストール要求の最大数を 0 から 999 の範囲で指定します。

この値を 0 に設定すると、それ以上の端末が自動インストールによってログオンされることを防げます。こうすることで、現在ログオンしている端末の自動インストールされたエントリーを、その端末がログオフするときに自動インストール・プログラムによって削除できます。

#### **PROGRAM(pgrmid)**

自動インストール・プロセスを制御しているユーザー・プログラムの名前を指定します。デフォルトは、CICS 提供プログラム DFHZATDX です。

### **自動インストールされた端末定義のリカバリーと再始動**

このトピックでは、ログオフとシステムの再始動時に自動インストールされた端末定義で発生することについて説明します。

#### **CICS の再始動で発生すること**

緊急時再始動で、**AIRDELAY** システム初期設定パラメーターで再始動の遅延期間にゼロを指定しない限り、自動インストールされた TCT 項目がリカバリーされます。

ユーザーは、緊急時再始動後に、自動インストール・プロセスを行わずにもう一度ログオンできます。AUTOCONNECT(YES) が TYPETERM 定義に指定されたこれらの端末は、再始動プロセス中に自動的にログオンされ、オペレーター介入の必要はありません。自動インストールされた TCT 項目のリカバリーは、CICS の再始動に続く多数の並行自動インストール要求のパフォーマンス上の影響を回避できます。再始動後にログオンする端末ユーザーは、前の CICS の実行中にその端末の NETNAME に対して作成された TCT 項目を使用します。これは、その端末が TCT にインストールされている個別の TERMINAL 定義を持っているのと似ています。

このことがセキュリティに対する脅威となる可能性があるため、CICS はリカバリー後にオペレーターのアクティビティを確認します。遅延後に、リカバリーされたが再びセッション中でない自動インストールされた TCT 項目がすべて削除されます。こうすることで、セキュリティの向上だけでなく、CICS ストレージが使用されない TCT 項目で浪費されなくなります。システム初期設定パラメーターを使用して、遅延の長さを指定できます。

z/OS Communications Server の持続セッション・サポートが CICS 領域に対して使用中であり、AIRDELAY が非ゼロの場合は、自動インストールされた TCT 項目が、その他の TCT 項目とまったく同じように処理されます。

ウォーム・スタートでは、AIRDELAY の時間の有効期限が切れる前にログオフして CICS がシャットダウンされない限り、前に自動インストールされた TCT 項目が失われます。

TCTTE が緊急時再始動中にリカバリーされている場合、AUTOCONNECT(YES) の仕様では、AIRDELAY によって TCTTE は削除されません。この状況で TCTTE ストレージを削除する場合は、AUTOCONNECT(NO) を指定します。

### 自動サインオフ、ログオフ、および TCTTE の削除

セッションがユーザーの TIMEOUT 期間の期限によって終了した場合、SIGNOFF(LOGOFF) がモデルの TYPETERM 定義で指定された場合にのみ、端末入力が前のセクションで説明したように削除されます。

254 ページの表 23 は自動インストールされた TCTTE の自動削除とリカバリーを要約します。

255 ページの図 51 では、セッションがタイムアウトした場合に、自動サインオフ、ログオフ、および TCTTE の削除がどのように発生するのかが示しています。257 ページの図 52 は、ウォーム・スタートまたは緊急時再始動に自動インストールされる端末に TCTTE が存在する場合、ログオンと TCTTE の削除がどのように発生するのかが示しています。256 ページの表 24 では、タイムアウト以外の理由でセッションが終了した場合に、TCTTE の自動削除がどのように発生するのかが示しています。

表 23. AUTOINSTALL – TCTTE のリカバリーと削除の要約	
<b>CICS RUNNING:</b>	
再始動の遅延 = 0	TCTTE 項目がカタログされないため、後続の実行でリカバリーできません。
再始動の遅延 > 0	TCTTE 項目がカタログされます。これらの項目がこの実行中またはシャットダウン時に削除されない場合、後続の緊急時再始動でリカバリーできます。
<b>SHUTDOWN:</b>	
ウォーム	シャットダウン前にログオフしたが、AIRDELAY の有効期限が切れていない端末を除き、端末がログオフして TCTTE が削除されます。
即時	TCTTE は削除されません。すべて後続の緊急時再始動でリカバリーできます。
異常終了	TCTTE は削除されません。すべて後続の緊急時再始動でリカバリーできます。
<b>STARTUP:</b>	
コールド	TCTTE はリカバリーされません。
ウォーム	TCTTE はリカバリーされません。
再始動の遅延 = 0 の場合の緊急時再始動	TCTTE はリカバリーされません (ただし 257 ページの図 52 の注を参照してください)。このセッションが持続する場合はアンバインドされています。
再始動の遅延 > 0 の場合の緊急時再始動	TCTTE はリカバリーされます。詳細については、257 ページの図 52 を参照してください。このセッションが持続する場合はリカバリーされます。

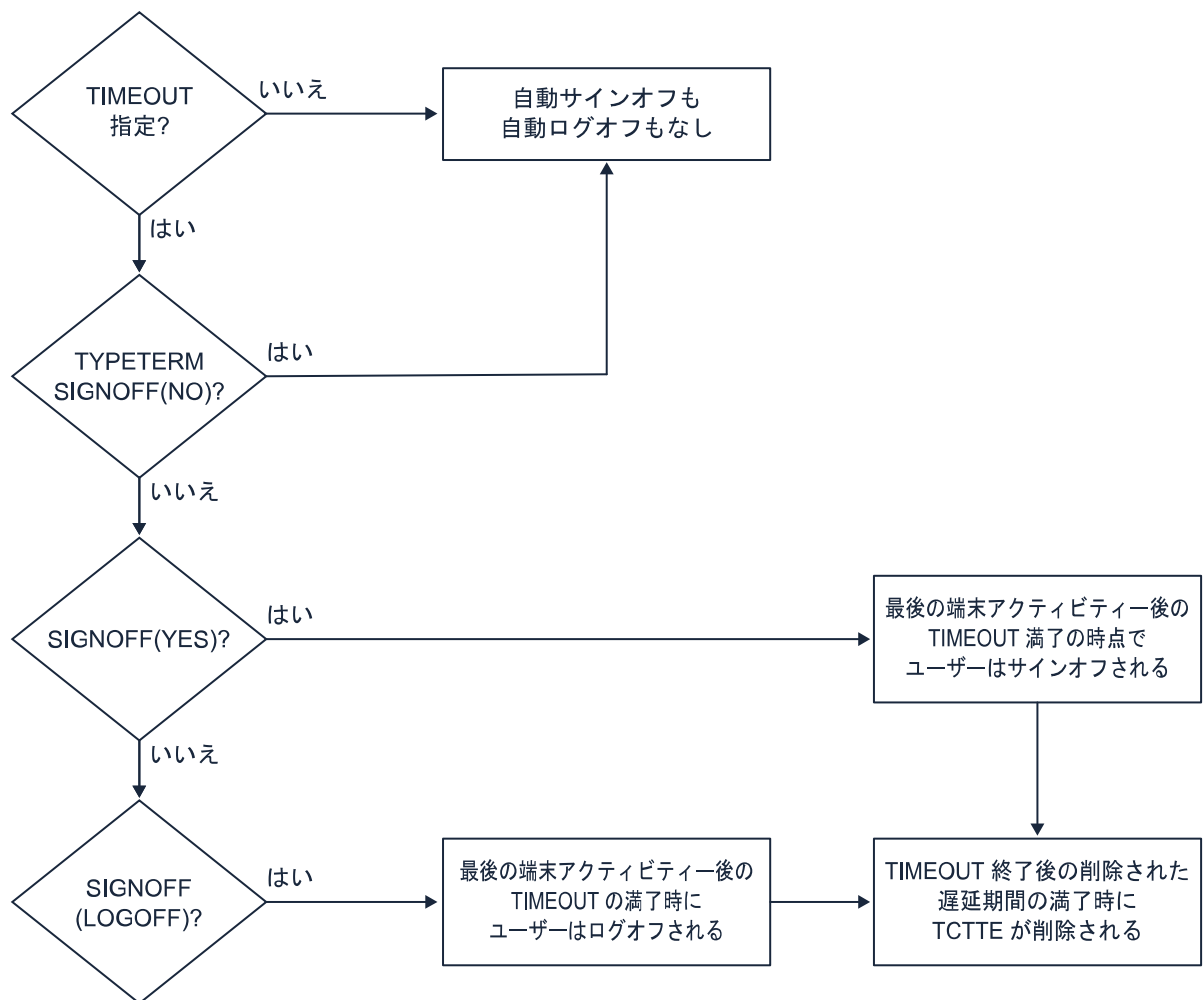


図 51. 自動インストールされた端末の自動サインオフ、ログオフ、および削除

表 24. AUTOINSTALL		
AUTOINSTALL – 非自動ログオフ後の TCTTE の自動削除		
非自動 LOGOFF: z/OS Communications Server は、CICS にセッションの障害、端末のログオフ、 またはトランザクションでの端末の切 断を通知します。	<-----遅延期間の削除-----> 端末はこの期間中、自動インストール・プロセス を繰り返さなくてもログオンできます。	TCTTE 項目が削除 されました、

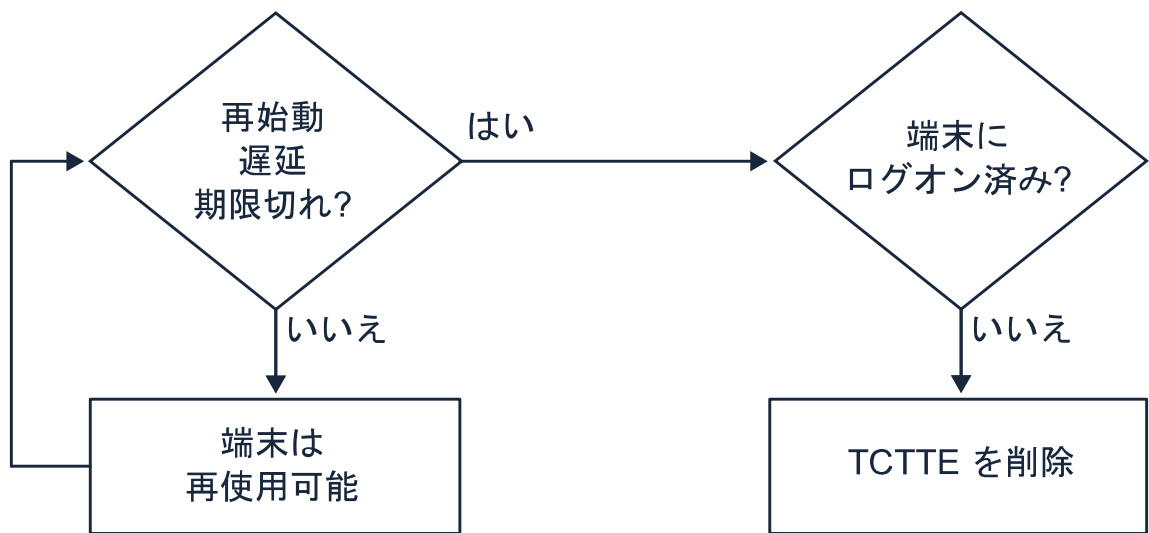


図 52. AUTOINSTALL – 緊急時再始動後の自動ログオフと TCTTE の削除



## MVS コンソールの自動インストール

MVS コンソール (またはジョブ・ストリーム) で発行されたコマンドは、MVS MODIFY コマンドを使用して、開始済みのタスクまたはジョブとして実行されている CICS 領域に送信できます。CICS で MVS コマンドを受け入れるためには、コマンドを発行するコンソールの項目が端末管理テーブル内に必要です。CICS 端末管理はこの項目を使用して応答を表示できます。

CICS は、MVS オペレーター・コンソールから受け取るコマンド (トランザクション呼び出し) を以下のように処理します。

### 事前インストールされたコンソール定義

MVS は、要求を受け取るとタスク名またはジョブ名から CICS 領域を特定して、要求を CICS に渡します。

CICS はコンソールの名前を MODIFY データから抽出し、コンソール名と一致する CONSNAME 項目を端末管理テーブル (TCT) から検索します。CICS は、一致する項目を見つけると、MODIFY コマンドで指定されたトランザクションを開始します。トランザクションは、端末管理テーブルに含まれるコンソールの項目の termid を使用して結果をコンソールに送信できます。

### 自動インストールされたコンソール定義

CICS は、一致する項目が見つからない場合、コンソールの自動インストール状況を調べて、そのコンソールの自動インストールを実行できるかどうかを判別します。

コンソールの自動インストールがアクティブであれば、CICS は自動インストール・モデル・テーブル (AMT) を検索して、そのコンソールの自動インストール処理を開始します。CICS は、以下のいずれかを行います。

- ・自動インストール・モデル定義のリストを、対象となるコンソールに関する情報と共に自動インストール制御プログラムに渡します。または、
- ・検出された最初のコンソール・モデル定義またはコンソールと同じ名前のコンソール・モデルを自動的に使用して、自動インストール制御プログラムを呼び出さずに、CICS により生成された termid を使用してコンソールを自動インストールします。

これらのオプションのどちらを CICS が採用するかは、コンソールの自動インストール状況によって決まります。コンソールの自動インストール状況は、AICONs システム 初期設定パラメーターによって起動時に設定されるか、または CEMT (あるいは EXEC CICS) の SET AUTOINSTALL CONSOLES コマンドによって動的に設定されます。

### 端末自動インストール制御プログラム

コンソールの自動インストールには、AIEXIT システム 初期設定パラメーターに制御プログラムの名前を指定して、z/OS Communications Server 端末および APPC 接続の場合と同じ自動インストール制御プログラムを使用します。

自動インストール制御プログラム (CICS 提供のプログラムまたは独自プログラムのいずれか) が呼び出されると、いずれかのモデルが選択されて、コンソールの TCT 端末項目を完成させるために必要な残りの情報が提供されます。自動インストール制御プログラムが制御を返すとき、CICS は、自動インストール制御プログラムから返された自動インストール・モデル、termid、およびその他のデータに加え、MVS コンソール・データを使用して、コンソールの端末管理テーブル端末項目 (TCTTE) を作成します。その後、CICS は新しい項目を TCT に追加して、MODIFY コマンドで指定されたトランザクションを開始します。

### 自動インストールされたコンソールの事前設定セキュリティ

モデル端末が USERID(\*FIRST) または USERID(\*EVERY) を指定する場合、CICS は、MODIFY コマンドで MVS によって渡されたユーザー ID を使用してコンソールにサインオンします。つまり、MVS によって渡されるユーザー ID が新しいコンソールの事前設定ユーザー ID として使用されます。

### 自動インストールされたコンソールの自動削除

CICS は、自動インストールされたコンソールが指定された遅延期間 (デフォルトは 60 分間) にわたって使用されない場合、そのコンソールを自動削除します。インストール機能の一部として、自動インストール制御プログラムはコンソールの「delete-delay」値を設定できます。delete-delay 期間は、自動インストールされたコンソールが未使用であることが許容される時間の長さ (分数) です。この期間を過ぎると、コンソールは CICS によって削除されます。この値を 0 に設定すると、自動削除が禁止されます。自動インストールされたコンソールはカタログに記録されないため、再始動のときにリカバ

リーされません。現在サインオンしているユーザーがいる場合でもコンソールが削除されることに注意してください。

## MVS コンソールの自動インストールの実装

MVS コンソールの自動インストールを実装するには、適切なリソース定義を定義して、システム初期設定パラメーターを構成します。

### このタスクについて

MVS コンソールを自動インストールするには、必要なモデル・リソース定義がインストールされていることを確認する必要があります。DFHLIST で提供される自動インストール・モデルには、コンソールの自動インストールに適したモデルが含まれていません。提供されているグループ DFHTERM C にはコンソールの自動インストール・モデル定義が含まれていますが、DFHLIST には含まれていません。

自動インストール・プロセスに対して追加の制御を行う場合は、オプションで自動インストール制御プログラムを作成できます。例えば、実稼働環境で自動インストールが可能となるコンソールを制限することができます。自動インストール・プログラムは、コンソールのインストール要求と削除要求を処理できなければなりません。コンソールの自動インストール・プログラムを作成する方法については、[コンソールの自動インストールを制御するプログラムの作成](#)を参照してください。

### 手順

1. 以下の属性を使用して、コンソール端末定義を定義します。

- AUTINSTMODEL(YES) または AUTINSTMODEL(ONLY) を指定します
- CONSNAME(name) を指定します
- DEVICE(CONSOLE) を指定する TYPETERM を参照します

**INITIALIZE** コマンドおよび **UPGRADE** コマンドによって CSD に追加される、グループ DFHTERM C に定義されているモデル・コンソール定義を使用できます。

注：コンソール自動インストール機能によってモデル端末定義が使用される場合、CICS は、モデル定義で指定されたコンソール名を無視します。

2. モデル・コンソール定義をインストールします。そのグループをグループ・リストに追加してコールド・スタートを実行するか、または CICS Explorer を使用してリソースをインストールすることができます。
3. MVS コンソールの自動インストールを制御するために独自のプログラムを提供する場合、[AICONS](#) システム初期設定パラメーターを YES に設定します。

この値は SIT で定義することも、**SET AUTOINSTALL CONSOLES(PROGAUTO)** コマンドを使用して動的にオーバーライドすることもできます。

4. CICS で提供される自動インストール制御プログラムを使用するには、[AICONS](#) システム初期設定パラメーターを AUTO に設定します。

この値は SIT で定義することも、**SET AUTOINSTALL CONSOLES(FULLAUTO)** コマンドを使用して動的にオーバーライドすることもできます。AUTO オプションを指定すると、CICS は TERMID を自動的に割り振ります。

5. 必要な自動インストール・プログラムとトランザクションがインストールされていることを確認します。これらのリソースは、自動インストール制御プログラム、トランザクション CATA と CATD、およびプログラム DFHZATD と DFHZATA です。

以下のリソースがインストールされている必要があります。

- 自動インストール制御プログラムのための PROGRAM リソース
- CATA と CATD のための TRANSACTION リソース
- DFHZATD と DFHZATA のための PROGRAM リソース。

提供されている自動インストール制御プログラム DFHZATDX または DFHZATDY は、コンソールの自動インストール・モデルが AMT にある場合、任意のコンソールからの要求を受け入れます。グループ DFHTERM C で提供されているモデル定義を使用するか、または代わりに独自の自動インストール・コン

ソール・モデルを作成できます (260 ページの『MVS コンソールの自動インストール制御プログラム』を参照してください)。

### コンソールのモデル TERMINAL 定義の定義

CICS が不明なコンソールからコマンドを受信して、コンソールの自動インストールがアクティブな場合、CICS はコンソールを記述する自動インストール・モデルの AMT を検索します。

これらのモデルは CONSNAME を指定して、DEVICE(CONSOLE) を指定する TYPETERM 定義を参照する必要があります。モデル定義のコンソール名は、自動インストールの場合はダミーの値であり、CICS はこの値を無視し、AUTINSTNAME 属性値のリストを自動インストール制御プログラムに渡します。受け取ったプログラムは、それらのリストの値を 1 つ選択できます。

### 自動事前設定セキュリティの指定

自動インストール制御プログラムがモデル定義で必要とする可能性がある 1 つの属性は、正しい事前設定セキュリティを提供する USERID です。

事前設定セキュリティが定義されたモデルを選択することで、オペレーターのセキュリティ権限が事前に決定されます。これにより、MODIFY コマンドを発行するために CESN サインオン・トランザクションを使用して、オペレーターが CICS にログオンする必要がなくなります。USERID パラメーターの 2 つの特殊なオペランドが、コンソールの自動サインオンを提供します。

- USERID(\*EVERY) は、MODIFY コマンドが受信されるたびに、MVS MODIFY コマンドで渡されるユーザー ID を CICS が使用することを意味します。コンソールは、自動インストールされているコンソールの事前設定のユーザー ID として、MVS ユーザー ID を使用してサインオンされます。コンソールが削除されるか、別の MODIFY コマンドが別のユーザー ID で受信されるまで、コンソールはこのユーザー ID でサインオンされたままです。MODIFY コマンドがユーザー ID なしで受信された場合、有効なユーザー ID を持つ MODIFY コマンドが受信されるまで、CICS はデフォルトの CICS ユーザー ID をサインオンします。コンソール以外の端末の場合、またはセキュリティが使用可能になっていない場合、この値は無視されます。
- USERID(\*FIRST) は、コンソールの自動インストールを要求する最初の MVS MODIFY コマンドで渡されるユーザー ID を CICS が使用することを意味します。コンソールは、MVS ユーザー ID を事前設定されたユーザー ID として使用してサインオンされます。コンソールが削除されるまで、このユーザー ID を使用してコンソールがサインオンされたままです。MODIFY コマンドがユーザー ID なしで受信された場合、CICS はデフォルトの CICS ユーザー ID をサインオンします。コンソール以外の端末の場合、またはセキュリティが使用可能になっていない場合、この値は無視されます。

## MVS コンソールの自動インストール制御プログラム

コンソール名およびコンソールに関する他の MVS データだけでは、コンソールの端末管理テーブル端末項目 (TCTTE) を作成する上で不十分です。自動インストール制御プログラムは、コンソールの CICS 端末 ID (termid) を作成し、CICS によって通信域で渡されるモデルのリストから適切なモデルを選択する必要があります。

IBM は、基本的な機能を実行する自動インストール制御プログラムの 2 種類のアセンブラー・バージョン (SDFHLOAD 内の DFHZATDX および DFHZATDY) を提供しますが、これらによって必要なすべての機能が実行されるとは限りません。例えば、端末名およびそれらとコンソール名との関係について、ユーザー独自の規則が設定されている場合があります。端末名の長さは最大 4 文字で、コンソール名の長さは最大 8 文字なので、多くの場合、一方から他方を派生させることはできません。

termid を提供することに加えて、コンソール定義に関連した他の機能を実行するようにプログラムをコーディングすることもできます。例えば、プログラムによって以下を行うことができます。

- セキュリティ検査の実行
- 現在ログオン中の自動インストールされたコンソール数のモニター

自動インストール制御プログラムは、トランザクション環境で、CICS 出口としてではなくユーザーによる置き換え可能なプログラムとして実行されます。つまり、端末名の判別に役立つようにファイルの読み取りや他の CICS コマンドの発行を行うことができます。ただし、コンソールの TCTTE 項目は、そのプログラムが呼び出されるどちらの場合にも存在しません。その理由は、(1) インストール機能のために呼び出すときにはまだ作成されておらず、(2) 削除機能のために呼び出すときには既に削除されているためです。そのため、このプログラムは transaction-without-terminal モードで実行します。

自動インストール制御プログラムを作成できる言語は、アセンブラー言語、C、COBOL、PL/I です。CICS 提供の自動インストール・プログラム・ソースは、4つの言語すべてで使用可能です。デフォルトで使用するアセンブラー・バージョンも、実行可能な形式で SDFHLOAD 内で提供されています。独自のプログラムを作成する場合は、IBM 提供のプログラムの 1つをパターンとして使用できます。

自動インストール制御プログラムの名前は AIEXIT システム 初期設定パラメーターで指定します。CICS は一度に 1つしか自動インストール制御プログラムをサポートしないので、コンソールと端末の自動インストールをサポートする必要がある場合は、それら両方の要求をプログラムで扱う必要があります。

自動インストール制御プログラムをテストすると、CICS が TCT 項目をインストールおよび削除するたびに、CICS がインストール・メッセージと削除メッセージを一時データ宛先 CADL に書き込むことが分かります。コンソールのための自動インストール・モデルがインストールされていない場合、CICS は自動インストール制御プログラムを呼び出さないで、自動インストール要求は失敗します。

CICS 提供の自動インストール制御プログラムを実装する方法や、独自のプログラムを設計して作成する方法について詳しくは、[コンソールの自動インストールを制御するプログラムの作成](#)を参照してください。

## APPC 接続の自動インストール

接続に自動インストールを使用するかどうかを決めるとき、いくつかの要素を考慮する必要があります。

### セキュリティ

接続に自動インストールを使用する前に、セキュリティ考慮事項が目的に適しているかどうかを考慮します。

自動インストールされた接続は、モデルで指定されたセキュリティ属性を継承します。CONNECTION 定義からのセキュリティ属性には、以下のものがあります。

- SECURITYNAME
- ATTACHSEC
- BINDSECURITY

互換モードを使用して CSD ファイルを CICS の以前のリリースと共用する場合、BINDPASSWORD 属性も継承されます。これらの属性については、[SESSIONS 属性](#)を参照してください。

自動インストールされた接続は、SESSIONS 定義から事前設定セキュリティ属性の USERID を継承します。非 EBCDIC ベースのシステムから attachsec 検証接続を自動インストールしようとする場合には、[LU6.2 でのバインド時のセキュリティ](#)を参照してください。

### モデル端末サポート (MTS)

接続の自動インストールでは MTS はサポートされません。したがって、モデル定義名を選択するルーチンを、ユーザーがコーディングする必要があります。z/OS Communications Server では、この作業を行うことはできません。MTS については、[247 ページの『自動インストールおよび z/OS Communications Server』](#)で説明しています。

### 自動インストールされた接続の削除

自動インストールされた端末定義とは異なり、自動インストールされた接続定義は、使用されなくなったときに削除されることはありません。つまり、それらはストレージを占有し続けます。

自動インストールされた APPC 接続のカタログ作成および削除の規則は、CICS Transaction Server for z/OS で変更されました。自動インストールされたすべての接続は、AIRDELAY システム 初期設定パラメーターに基づいてカタログ作成されるようになりました。AIRDELAY=0 の場合、同期レベル 1 接続はカタログ作成されません。

削除の規則は次のとおりです。

- 自動インストールされた同期レベル 1 接続は、解放されるときに削除されます。
- CICS が汎用リソースとして登録されていない場合、自動インストールされた同期レベル 2 接続は削除されません。
- CICS が汎用リソースとして登録されている場合、自動インストールされた同期レベル 2 接続および限定リソース接続は、親和性が終了するときに削除されます。



自動インストールが使用可能な場合に、一致する CICS CONNECTION 定義がない APPC サービス・マネージャ (SNASVCMG) セッションに対する APPC BIND 要求が受信されるか、単一セッションに対する BIND 要求が受信されると、新しい接続が自動的に作成されてインストールされます。

## APPC 接続の自動インストールの実装

すべて同じ特性を持つワークステーションが大量にある場合、接続の自動インストールから利点を得られる可能性が最も高くなります。

### 手順

1. 接続の自動インストールを使用するかどうかを判断します。

接続の自動インストールを使用する主な利点として、コールド・スタート、ウォーム・スタート、および緊急リスタートが高速になり、管理する CSD ファイル定義が少なくなり、使用されない定義によるストレージの占有が少なくなります。

ただし、261 ページの『[APPC 接続の自動インストール](#)』と 263 ページの『[接続の自動インストールにおけるリカバリーと再始動](#)』で考えられる制約事項がいくつか説明されています。

2. 自動インストールするセッションを決定します。

自動インストールは CICS 間の接続に対して使用できますが、主にワークステーションを想定しています。

3. モデル接続定義を作成します。

モデル定義は、同じプロパティですべての接続に使用できる 1 つの定義を CICS に提供します。262 ページの『[接続の自動インストールのモデル定義](#)』を参照してください。

4. 自動インストール制御プログラムを設計して作成します。

自動インストール制御プログラムは、自動インストール・モデル名など、CICS に自動インストール要求の実行に必要な追加情報を提供します。

自動インストール制御プログラムに関するプログラミングの情報については、LU の[自動インストールを制御するプログラムの作成](#)を参照してください。

5. 接続の自動インストールを有効にします。

端末の自動インストールを有効にすると接続の自動インストールが有効になります。使用されるシステム初期設定パラメーターと、CEMT および EXEC CICS INQUIRE および SET オプションの情報については、251 ページの『[z/OS Communications Server の自動インストールの実装](#)』を参照してください。

端末の自動インストールが有効になっているが、接続の自動インストールは行わない場合は、**SET CONNECTION(connection-name) OUTSERVICE** コマンドを使用して、モデル接続をサービス休止に設定します。詳細については、CEMT **SET CONNECTION** および **SET CONNECTION** を参照してください。モデル接続がサービス休止の場合、自動インストール制御プログラムはそのモデルにアクセスしてコピーできないため、接続の自動インストール機能が効率的に無効になります。

## 接続の自動インストールのモデル定義

接続の自動インストールのモデル定義は、モデルとして明示的に定義される必要がないという点で、端末の自動インストールと異なります。インストールされた接続定義は、自動インストールされた接続の「テンプレート」として使用できます。

パフォーマンス上の理由で、他では使用されないインストール済みの接続定義を使用します。定義は CICS がコピーしている間ロックされ、大量のセッションを自動インストールしている場合には遅延が目立つ可能性があります。

CEMT コマンド、または EXEC CICS SET CONNECTION OUTSERVICE コマンドを使用してモデル接続定義をサービス休止に設定できます。自動インストール制御プログラムがモデル定義にアクセスしてコピーできないため、そうすることで接続の自動インストールが効率的に無効になります。

CICS 間の接続の自動インストールでは、モデル接続の SESSIONS 定義内の MAXIMUM 属性は、モデルを使用した最大デバイスを格納できる十分な大きさが必要です。

モデル接続を作成する場合、セッションの MODENAME フィールドに指定されたユーザー・グループのモード名は、自動インストールされる接続のユーザー・グループのモード名に一致していることを必ず確認してください。

## 接続用自動インストール制御プログラム

自動インストール制御プログラムは、BIND によって開始される APPC の単一および並列セッションの接続に対してインストール時に起動されます。自動インストール制御プログラムは、CICS に自動インストール要求の実行に必要な追加情報を提供します。APPC 並列セッションでは、制御プログラムは新しい定義の SYSID を提供します。

APPC BIND 要求が CICS によって受信されると、CICS はパートナーの z/OS Communications Server NETNAME を受信し、自動インストール制御プログラムに受け渡します。制御プログラムは、パートナーの NETNAME と z/OS Communications Server BIND に含まれる情報を使用して、新しい接続の基本となる最適なモデルを選択します。最適なモデルの名前を CICS に戻すには、制御プログラムがすべてのモデルの NETNAME または SYSID を認識する必要があります。

CICS は接続用自動インストールのためにサンプルの制御プログラム DFHZATDY を提供します。以下の条件が両方とも該当する場合には、DFHZATDY を変更せずに使用できます。

- モデル接続が CCPs、CBPS、または CBSS と呼ばれている
- NETNAME の最後の 4 文字を SYSID または端末名として使用している

これらの条件が満たされない場合は、インストールに合うように DFHZATDY を変更する必要があります。ソースは CICSTS56 に提供されています。SDFHSAMP.DFHZATDY は以下のように定義されます。

```
DEFINE PROGRAM(DFHZATDY) GROUP(DFHAI62) LANGUAGE(ASSEMBLER)
      RELOAD(NO) RESIDENT(NO) STATUS(ENABLED) CEDF(NO)
      DATALOCATION(ANY) EXECKEY(CICS)
```

提供されるモデル接続とセッションの定義は以下のとおりです。

```
DEFINE CONNECTION(CBPS) GROUP(DFHAI62) NETNAME(TEMPLATE1)
      ACCESSMETHOD(VTAM) PROTOCOL(APPC) SINGLESESS(NO)
DEFINE SESSION(CBPS) GROUP(DFHAI62) CONNECTION(CBPS)
      MODENAME(LU62PS) PROTOCOL(APPC) MAXIMUM(10,5)
```

```
DEFINE CONNECTION(CBSS) GROUP(DFHAI62) NETNAME(TEMPLATE2)
      ACCESSMETHOD(VTAM) PROTOCOL(APPC) SINGLESESS(YES)
DEFINE SESSION(CBSS) GROUP(DFHAI62) CONNECTION(CBSS)
      MODENAME(LU62SS) PROTOCOL(APPC) MAXIMUM(1,0)
```

```
DEFINE CONNECTION(CCPs) GROUP(DFHAI62) NETNAME(TEMPLATE3)
      ACCESSMETHOD(VTAM) PROTOCOL(APPC) SINGLESESS(NO)
DEFINE SESSION(CCPs) GROUP(DFHAI62) CONNECTION(CCPs)
      MODENAME(LU62PS) PROTOCOL(APPC) MAXIMUM(10,5)
```

これらの定義を使用する場合は、グループ DFHAI62 をグループ・リストに追加する必要があります。端末の自動インストール出口 DFHZATDX を使用して接続を自動インストールしないでください。DFHZATDX によってインストールされるセッションは終了し、メッセージ DFHZC6921 が発行されます。

注：VTAM は現在 z/OS Communications Server です。

自動インストール制御プログラムのカスタマイズに関するプログラミング情報については、[LU の自動インストールを制御するプログラムの作成](#)を参照してください。

## 接続の自動インストールにおけるリカバリーと再始動

自動インストールされた接続に関する情報は、すべてではありませんが、再起動の状態の一部でリカバリーされます。

自動インストールされた接続はカタログされませんが、持続セッション・サポートから利点を得ることはできません。これは、セッションが永続的な場合に、それに関連する自動インストールされた接続がアンバインドされることを意味します。



自動インストールされた接続は、CICS のコールド・スタートでのみリカバリーされます。これらはウォーム・スタートおよび緊急時再始動ではリカバリーされません。

自動インストールされた接続のリカバリー単位ディスクリプター (URD) は、接続の SYSID が一貫している限り、コールド、ウォーム、および緊急時再始動の後でリカバリーされます。

自動インストールされた接続の SYSID は、リカバリー可能リソースを使用している場合は、一貫性がなければなりません。

## IPIC 接続の自動インストール

IPCONN の自動インストールの際には、他のリソースの自動インストールの場合とは異なり、モデル定義は必要ありません (ただしそれは推奨されています)。

IPCONN の自動インストールに使用される定義は、端末の自動インストールに使用されるモデル定義とは異なり、モデルとして明示的に定義されたものである必要はありません。代わりに、CICS は、以前にインストール済みの IPCONN 定義を新しい定義のテンプレートとして使うことができます。

IPCONN の自動インストールについて詳しくは、[IPIC 接続の自動インストール: 事前の考慮事項](#)を参照してください。

## 自動インストール・プログラム、マップ・セット、および区分セット

プログラムの自動インストールがアクティブであり、暗黙的または明示的なロード要求が以前には未定義だったプログラム、マップ・セット、または区画セットに対して発行された場合、CICS は動的に定義を作成し、インストールとカタログを適宜実行します。

暗黙的または明示的なロードが発生するのは以下の状態です。

- CICS がトランザクションを開始するとき。
- アプリケーション・プログラムが、以下のコマンドのいずれか 1 つを実行するとき。

EXEC CICS LINK (SYSID なし)

EXEC CICS XCTL

EXEC CICS LOAD

EXEC CICS ENABLE (グローバル・ユーザー出口プログラム、またはタスク関連のユーザー出口プログラムの場合)

- EXEC CICS HANDLE ABEND PROGRAM(...) の後になんらかの状態が発生し、CICS が指名されたプログラムに制御を移行させる。
- CICS はユーザーによる置き換え可能なプログラムを初めて呼び出すとき。
- アプリケーション・プログラムが、以下のコマンドのいずれか 1 つを実行するとき。

EXEC CICS RECEIVE または SEND MAP

EXEC CICS SEND PARTNSET

EXEC CICS RECEIVE PARTN

プラットフォームにデプロイされるアプリケーションは、プログラム、マップ・セット、または区画セットが自動インストールされる、暗黙的または明示的なロード要求を作成できます。プラットフォーム上にデプロイされたアプリケーションのタスクによって自動インストールされたプログラムは、そのアプリケーションのそのバージョン専用です。専用リソースは、プラットフォームにインストールされている他のアプリケーションやバージョンからは使用できず、CICS 領域内の他のタスクからも使用できません。アプリケーションの専用リソースについて詳しくは、[アプリケーション・バージョンの専用リソース](#)を参照してください。

## プログラムの自動インストールの実装

自動インストールできない唯一のプログラムは、プログラムの自動インストール制御プログラムです。

### 手順

1. プログラムが自動インストールに対して適格かどうかを判断します。

その他のプログラムは、すべて自動インストールできます (以下に例を示します)。

- その他のすべてのユーザー置換可能プログラム
- グローバル・ユーザー出口ルーチン (GLUE) およびタスク関連ユーザー出口ルーチン (TRUE)
- PLT programs

ユーザー置換可能プログラムと PLT プログラムは、最初の参照で自動インストールされます。GLUE と TRUE は使用可能な場合は自動インストールされます。

2. プログラムに自動インストールを使用するかどうかを判断します。

プログラムに自動インストールを使用すると、個別のプログラム、マップ・セット、および区画セットの定義に費やす時間を節約できます。また、自動インストールされるリソースの定義は、参照されるまでスペースを占有しないため、ストレージも節約できます。

プログラムに自動インストールを使用すると、コールド・スタートでインストールされる定義の数を削減できるため、所要時間も削減されます。

3. 自動インストールするプログラムを決定します。

プログラムに応じて、RDO と自動インストールを混合して使用することができます。プログラムの自動インストールを管理するために推奨される方法は、既存のプログラム定義、および関連するプログラムを含むグループのインストールに RDO を使用することです。新しいアプリケーションを開発してインストールする場合は自動インストールを使用します。テスト環境では、そうしないと CICS の起動時に大量のプログラムをインストールする可能性があります。

4. プログラムの自動インストールを有効にします。

システム初期設定パラメーターを指定するか、**SET SYSTEM** コマンドを使用して、プログラムの自動インストールを有効にできます。

次の 3 つのシステム初期設定パラメーターが、プログラムの自動インストールに関連しています。

**PGAICTLG**

PGAICTLG パラメーターは、自動インストールされたプログラム定義をカタログするかどうかを指定します。266 ページの『プログラムの自動インストールのカタログ』を参照してください。

**PGAIPGM**

PGAIPGM パラメーターは、プログラム自動インストール機能がアクティブか非アクティブかを指定します。

**PGAIEEXIT**

PGAIEEXIT パラメーターは、プログラム自動インストール出口の名前を指定します。

次の 3 つのオプションが、**SET SYSTEM** コマンドのプログラムの自動インストールに関連しています。

**PROGAUTOCTLG**

PROGAUTOCTLG オプションは、自動インストールされたプログラム定義をカタログするかどうかを指定します。266 ページの『プログラムの自動インストールのカタログ』を参照してください。

**PROGAUTOEXIT**

PROGAUTOEXIT オプションは、プログラム自動インストール出口の名前を指定します。

**PROGAUTOINST**

PROGAUTOINST オプションは、プログラム自動インストール機能がアクティブか非アクティブかを指定します。

EXEC CICS コマンドの指定方法のプログラミング情報については Introduction to System programming commands を参照し、CEMT の使用方法については CEMT - マスター端末 を参照してください。

5. モデル・プログラム定義の作成

モデル定義は、CICS に同じプロパティーですべてのプログラムに使用できる 1 つの定義を提供します。266 ページの『プログラムの自動インストールのモデル定義』を参照してください。

6. 自動インストール制御プログラムを設計して作成します。

自動インストール制御プログラムは、自動インストール・モデル名など、CICS に自動インストール要求の実行に必要な追加情報を提供します。CICS アプリケーション・プログラミング・インターフェース

への全アクセス権限を持ち、CICS がサポートするどの言語でも、自動インストール・プログラムを作成できます。266 ページの『プログラムの自動インストール制御プログラム』を参照してください。

## プログラムの自動インストールのカタログ

CEMT INQUIREvSET SYSTEM コマンドの PGAICTLG システム初期設定パラメーターまたは PROGAUTOCTLG オプションを使用して、自動インストールされたプログラム定義がカタログされるかどうか(つまり、定義がウォーム・スタートまたは緊急時始動 w を経ても保持されるかどうか)を指定できます。

指定できる値は、以下のとおりです。

### ALL

自動インストールされたプログラム定義は、自動インストール時にグローバル・カタログに書き込まれ、後続の変更に従います。

### MODIFY

自動インストールされたプログラム定義は、プログラム定義が自動インストールの後の SET PROGRAM コマンドによって変更された場合にのみカタログされます。

### なし

自動インストールされたプログラム定義はカタログされません。これを指定すると、CICS がグローバル・カタログから定義を再インストールしないため、MODIFY オプションあるいは ALL オプションを指定した場合と比較して、CICS の再始動 (ウォームおよび緊急時) が高速になります。定義は最初の参照で自動インストールされます。

自動インストールされたプログラムのカタログに指定する設定は、プラットフォームで導入されたアプリケーションのタスクによって自動インストールされるプログラムには、影響がありません。これらのプログラムがカタログされることはありません。

プログラムの自動インストールにカタログを指定すると、主にリカバリーと再始動に影響があります。267 ページの『プログラムの自動インストール、リカバリー、および再始動』を参照してください。

## プログラムの自動インストールのモデル定義

プログラムの自動インストールのモデル定義は、モデルとして明示的に定義される必要がないという点で、端末の自動インストールと異なります。インストールされたプログラム、マップ・セット、または区画セットの定義は、プログラムの自動インストールの「テンプレート」として使用できます。

独自の定義を使用しない場合は、CICS 提供のモデル定義を使用できます。そのオプションは、以下のとおりです。

- プログラムの DFHPGAPG
- マップ・セットの DFHPGAMP
- 区画セットの DFHPGAPT

これらは、提供されたリソース定義、グループ、およびリストにリストされています。

注: DFHPGAPG 定義はプログラム言語を指定しませんが、CICS 自動インストール・ルーチンは、ロードされているプログラムのプログラム言語を検出します。

## プログラムの自動インストール制御プログラム

PGAIXIT システム初期設定パラメーターで使用する制御プログラムの名前を指定するか、CEMT または EXEC CICS INQUIREvSET SYSTEM コマンドを使用します。

プログラムの自動インストール制御プログラムに関するプログラミングの情報については、LU の自動インストールを制御するプログラムの作成を参照してください。このトピックは、その情報を要約したものです。

### 自動インストール制御プログラムがプログラムの自動インストールに対して起動される場合

自動インストール・プログラムは、プログラム、マップ・セット、および区画セットの多数の異なる状態で呼び出されます。

プログラムでは、自動インストール制御プログラムは次の場合に呼び出されます。

- 次のコマンドのいずれかが、前に未定義のプログラムを参照する場合。
  - EXEC CICS LINK
  - EXEC CICS XCTL
  - EXEC CICS LOAD
- プログラムはトランザクションの最初のプログラムです。
- EXEC CICS ENABLE は GLUE または TRUE に対して実行されます。
- EXEC CICS HANDLE ABEND PROGRAM コマンドが発行された後で異常終了が発生し、CICS が指定されたプログラムを起動します。
- CICS はユーザーによる置き換え可能なプログラムを呼び出します。

マップ・セットでは、EXEC CICS SEND MAP または EXEC CICS RECEIVE MAP が前に未定義のマップ・セットを参照する場合に、自動インストール制御プログラムが起動されます。

区画セットでは、EXEC CICS SEND PARTNSET または EXEC CICS RECEIVE PARTN コマンドが前に未定義の区画セットを参照するときに、自動インストール制御プログラムが起動されます。

### サンプル・プログラム

CICS はプログラムの自動インストール出口に多数のサンプル・プログラムを提供します。

以下のサンプル・プログラムは、CICS によって提供されます。

- プログラムの自動インストール出口のための DFHPGADX アセンブラー・プログラム
- プログラムの自動インストール出口のための DFHPGAHX-C プログラム
- プログラムの自動インストール出口のための DFHPGALX-PL/I プログラム
- プログラムの自動インストール出口のための DFHPGAOX-COBOL プログラム

これらのプログラムのソースは、CICSTS56.SDFHSAMP サンプル・ライブラリーで提供されますが、アセンブラー・バージョンのみが CICSTS56.SDFHLOAD ロード・ライブラリーで実行可能形式で提供されます。

### プログラム自動インストール機能

プログラム自動インストール機能は、モデル定義と一緒にユーザーが置換できるプログラムを使用して、自動インストールする必要があるプログラム、マップ・セット、および区画セットの明示的な定義を作成します。

CICS は適切なモデル定義の名前を提供するパラメーター・リストを使用して、ユーザーが置換できるプログラムを呼び出します。プログラムからの戻り時に (戻りコードに応じて)、CICS はモデルの情報、およびプログラムが返すパラメーターから、リソース定義を作成します。

**注:** CICS は CICS のプログラム、マップ・セット、または区画セット (つまり、DFH という文字で始まるオブジェクト) に対して、ユーザーが置換できるプログラムを呼び出しません。

## プログラムの自動インストール、リカバリー、および再始動

カタログなしでプログラムの自動インストールを使用したウォーム・リスタートと緊急時再始動、およびカタログありで自動インストールを使用したウォーム・リスタートと緊急時再始動では、パフォーマンスに違いがあります。

カタログありで自動インストールを使用している場合、再始動時間はプログラムの自動インストールを使用していない CICS 領域の再始動と似ています。これはどちらの場合も、リソース定義が再始動中にカタログから再インストールされるためです。再始動後の定義は、システムが終了する前に存在したものです。

カタログなしで自動インストールを使用している場合、CICS は CICS グローバル・カタログから定義をインストールしないため、CICS の再始動時間が改善されます。代わりに、再始動後にプログラム、マップ・セット、および区画セットが参照されるときに、必要に応じて定義が自動インストールされます。

## モデル端末定義の自動インストール

自動インストールのモデル端末を定義する場合は、普通のリソース定義をインストールする場合のように、そのモデル端末が含まれているグループをインストールすることによって端末をインストールできます。

ただし、AUTINSTMODEL(ONLY) として指定されている端末定義は、この時に自動インストール・モデル・テーブル (AMT) にのみ保管されます。アクティブ CICS システムの TCTTE にはなりません。これらのモデル端末定義は、CICS が自動インストールを使用して z/OS Communications Server からログオンする実際の端末の定義を作成するためにそれを必要とするときまで AMT に保管されます。結果として作成される端末定義は、この時に「自動インストール」されます。

システム 初期化または INSTALL の使用によって TERMINAL 定義がインストールされると、次のことが行われます。

### AUTINSTMODEL(NO)

端末の TCT 項目が作成されます。

### AUTINSTMODEL(YES)

端末の TCT 項目が作成され、後で自動インストールのプロセスに使用するためにモデル定義が保管されます。

### AUTINSTMODEL(ONLY)

後で自動インストールのプロセスに使用するためにモデル定義が保管されます。

同じ AUTINSTNAME を持つモデル TERMINAL 定義を 2 つインストールした場合は、2 つ目にインストールされた方の定義によって最初の定義が置き換えられます。

自動インストールとモデル TERMINAL 定義について詳しくは、[245 ページの『z/OS Communications Server 端末の自動インストール』](#)を参照してください。

## ジャーナルの自動インストール

CICS は、ジャーナル・データを、MVS システム・ロガーによって管理されるログ・ストリーム上のジャーナルまたはログに、あるいは SMF に書き込みます。

ジャーナル名 (または ID) とログ・ストリーム (または SMF ログ) との間の接続を確立できるように、JOURNALMODEL を定義する必要があります。RDO、DFHCSDUP、または EXEC CICS CREATE JOURNALMODEL コマンドを使用して、journalmodel を定義できます。

CICS は、ユーザー定義の JOURNALMODEL を使用して (適切な JOURNALMODEL が存在しない状況では、シンボル名を解決することによって作成されたデフォルト名を使用して)、ログ・ストリームとジャーナル要求の間のリンクを解決します。この情報について詳しくは、[JOURNALMODEL の属性](#)を参照してください。

ジャーナルは、ユーザー定義が可能なリソースではありません。

## マクロ・リソース定義

CICS テーブルの作成に使用されるリソース定義マクロの参照情報。

### CICS 管理テーブルとマクロの概要

オンライン・リソース定義 (RDO) を使用してほとんどの CICS リソースを定義できますが、一部のリソースでは CICS マクロを使用する必要があります。また、マクロを使用して管理テーブルを作成します。

#### CICS マクロで定義する必要があるリソース

CICS マクロを使用して、次のリソースを定義します。

- 非 SNA ネットワーク
- 非 SNA LU
- 非 VSAM ファイル
- リソースのモニター



- システム・リカバリー・リソース

## RDO で定義する必要があるリソース

次のリソースについては、オンライン・リソース定義 (RDO) を使用する必要があります。

- VSAM ファイル
- Programs (プログラム)
- マップ・セット
- 区分セット
- キュー
- トランザクション
- プロファイル
- SNA LU
- MRO (複数領域操作) システムおよび ISC (システム間連絡) システムを使用したリンクとセッション

## 管理テーブルの定義方法

CICS はシステムの初期化中の操作で構成されます。システム初期設定テーブル (SIT) を選択すると、その選択に基づいて CICS が他の管理テーブルを選択します。各管理テーブルは個別に作成され、システムの初期化前であればいつでも再作成できます。適切なマクロをコーディングすることで、必要な管理テーブルを準備します。テーブルごとに、マクロが自動的に必要なリンケージ・エディターの制御ステートメントを生成します。

管理テーブルに関連する次の領域について確認する必要がある場合があります。

- システムの運用に必要なシステム初期設定テーブル (SIT)。その他のテーブルは、該当する CICS ファシリティーを使用している場合にのみ必要になります。SIT について詳しくは、[CICS システム初期設定パラメーターの指定](#)を参照してください。
- 管理テーブルに必要なジョブ制御言語 (JCL) と、指定するマクロのステートメントをリンク・エディットしてアSEMBルする方法 278 ページの『[CICS 管理テーブルでのリソースの定義](#)』を参照してください。
- CICS が 16 MB より上または 16 MB より下 (16 MB 境界より上または 16 MB 境界より下とも呼ばれる) のテーブルをロードするかどうか。CICS テーブルのマクロには、これを決定するリンケージ・エディターの制御ステートメントが含まれます。269 ページの表 25 は、特定のテーブルが 16MB 境界より上、または 16MB 境界より下でロードされるかどうかを示します。

マクロによって定義できる管理テーブルは、[269 ページの表 25](#) に示されています。

管理テーブル	内容	16 MB 境界より上?	Reference
コマンド・リスト・テーブル (CLT)	XRF テークオーバーのコマンドとメッセージのセット。コマンド・リスト・テーブル (CLT) が XRF (拡張リカバリー機構) で使用されます。XRF を使用している場合は、CLT が必要です。これは代替 CICS システムによってのみ使用されます。CLT には、実行のために JES または MVS に受け渡されるコマンドのリストが含まれます。また、アクティブな CICS システムをキャンセルするための権限も提供します。	はい	<a href="#">コマンド・リスト・テーブル (CLT)</a>



表 25. マクロによって定義できる管理テーブル. 3 番目の列は、テーブルが 16 MB 境界より上または下でロードされるかどうかを示します。(続き)

管理テーブル	内容	16 MB 境界 より上?	Reference
データ変換テーブル	データ変換テーブルは、EBCDIC を使用しないハードウェア・プラットフォームで動作する CICS ファミリーのメンバーと通信するために、CICS が ISC を使用している場合に必要になる可能性があります。変換テーブルは、データをワークステーションの ASCII フォーマットから CICS ホストの EBCDIC フォーマットに変更する方法を定義します。		テーブル作成に使用される DFHCNV マクロについては、 <a href="#">変換テーブルの定義</a> で説明されています。
DL/I ディレクトリー (PDIR)	データベースとプログラム仕様ブロック。CICS-IMS DBCTL (データベース制御) を排他的に使用して CICS システムの DL/I の利用を管理する場合は、CICS を使用して DL/I ディレクトリー (PDIR) を定義する必要はありません。  PDIR は、CICS システムがアクセスするすべてのリモート・プログラム仕様ブロック (PSB) のディレクトリーです。  要求をリモート・データベース・マネージャー (リモート DL/I) に機能シップする場合は、1 つのディレクトリー PDIR のみが必要です。	いいえ	<a href="#">PDIR: DL/I ディレクトリー</a>
File control table (FCT)	BDAM ファイル定義。ファイル管理テーブル (FCT) は、BDAM ファイルを定義できるように保持されます。	いいえ	<a href="#">File control table (FCT)</a>
Monitoring control table (MCT)	各ユーザー・イベント・モニター・ポイント (EMP) で行うモニター・アクション (データ収集)。各 EMP のモニタリング・クラスごとに、さまざまなアクションを指定できます。	はい	<a href="#">Monitoring control table (MCT)</a>
プログラム・リスト・テーブル (PLT)	関連するプログラムのリスト。いくつかの PLT を生成してプログラムのリストを指定します。これは、CICS の開始の初期設定プログラム・フェーズで実行されるプログラム、制御されたシャットダウンの最初、または 2 番目の静止ステージ、あるいは両方で実行されるプログラム、または CEMT ENABLE や DISABLE コマンドによってグループとして使用可能、または使用不可になるプログラムのリストです。	はい	<a href="#">プログラム・リスト・テーブル (PLT)</a>

表 25. マクロによって定義できる管理テーブル. 3 番目の列は、テーブルが 16 MB 境界より上または下でロードされるかどうかを示します。(続き)

管理テーブル	内容	16 MB 境界 より上?	Reference
Recoverable service table (RST)	リカバリー可能サービス・エレメントのセット。リカバリー可能サービス・テーブル (RST) は、IBM CICS IMS DBCTL (データベース制御) のサポートで使用されます。XRF と DBCTL を使用している場合は、RST が必要です。これはアクティブな CICS システムによって使用されます。RST には、DBCTL の構成を定義するリカバリー可能サービス・エレメントのリストが含まれます。これは、CICS が接続する DBCTL を定義します。	はい	<a href="#">Recoverable service table (RST)</a>
システム 初期設定テーブル (SIT)	システム 初期設定プロセスで使用するパラメーター。特に、SIT はユーザーが指定したバージョンの CICS システム制御プログラムと CICS テーブルがロードされるよう (接尾部文字によって) 定義します。		CICS システム 初期設定パラメーターの <a href="#">指定を参照してください</a> 。
システム・リカバリー・テーブル (SRT)	CICS が代行受信する異常終了のコードのリスト。		<a href="#">システム・リカバリー・テーブル (SRT)</a> 。
Temporary storage table (TST)	一時ストレージ・キューの総称 (または接頭部)。CICS は TSMODEL リソース定義と組み合わせて、または代わりとして TST を利用することを引き続きサポートしています。明示的な SYSID を EXEC CICS 一時記憶域コマンドで指定することによって、一時記憶域データ共用キューを参照するアプリケーション・プログラムを持っている場合、または SYSID が XTSEREQ グローバル・ユーザー出口コマンドによって追加される場合に、TST を使用する必要があります。また、TSAGE 属性を必要とする場合に、TST を使用する必要があります。これらの機能を必要としない一時記憶域キューでは、TSMODEL リソース定義を使用することができ、TST のその他すべての機能、およびいくつかの追加の機能を提供します。	はい	<a href="#">Temporary storage table (TST)</a>
Terminal control table (TCT)	非 SNA LU ネットワークを定義するために保持されます。	いいえ	<a href="#">Terminal control table (TCT)</a>

表 25. マクロによって定義できる管理テーブル。3 番目の列は、テーブルが 16 MB 境界より上または下でロードされるかどうかを示します。(続き)

管理テーブル	内容	16 MB 境界 より上?	Reference
Terminal list table (TLT)	関連する端末のセット。端末リスト・テーブル (TLT) では、端末またはオペレーターの ID、またはその両方を論理的にグループ化できます。TLT は、監視端末操作 (CEST) によって、操作の有効範囲を定義して制限するために必要とされます。また、監視またはマスター端末操作 (CEMT) によって、事前に決定された端末のグループに機能を適用するために使用することもできます。TLT は、単独または他の TLT と組み合わせて使用して、メッセージのスイッチングの事前定義された宛先を提供することができます。	いいえ	<a href="#">Terminal list table (TLT)</a>
トランザクション・リスト・テーブル (XLT)	論理的に関連するトランザクション ID のセット。システム終了の最初の静止ステージ中に端末から開始できる ID、またはマスター端末から使用不可または使用可能にできる ID のグループのリスト。	はい	<a href="#">トランザクション・リスト・テーブル (XLT)</a>

プログラム・リスト・テーブル (PLT) を除き、これらのテーブルは STEPLIB 連結に含まれるライブラリーにコンパイルされます。

PLT はコンパイルされません。CICS は PLT のソース・コードを処理します。必要な PLT のソース・コード (ソースで参照されるコピー・メンバーを含む) は、PARMLIB 連結または DFHTABLE 連結に存在する必要があります。

### **マクロのフォーマット**

CICS マクロはアセンブラー言語で記述し、すべてのアセンブラー言語命令と同様に標準形式で記述する必要があります。

名前	操作	オペランド	コメント
空白 または 記号	DFHxxxxx	コンマで区切られた 1つ以上のオペランド	

## TYPE=INITIAL (制御セクション)

ほとんどのテーブルは、TYPE=INITIAL マクロで開始する必要があります。

いくつかのテーブルでは、TYPE=INITIAL マクロでテーブル全体に適用される情報を指定できます。

TYPE=INITIAL マクロは、CICS システム・テーブル用の制御セクション (CSECT) を確立し、必要なリンケージ・エディター制御ステートメントを生成します。CICS は、それぞれの TYPE=INITIAL マクロから生成された DFHVM マクロによって、各テーブルのエントリー・ポイントのアドレスを自動的に生成します。テーブルのエントリー・ポイント・ラベルは DFHxxxBA です。END ステートメントだけを指定する必要があります。

### テーブルの名前指定と接尾部指定

テーブル名に接尾部を使用すると、1 つのテーブルに対して複数のバージョンを設定できます。各テーブルの名前は、固定した 6 文字と、それに続く 2 文字の接尾部からなります。

テーブルの名前は次のように指定されます。

表 26. 管理テーブルの名前	
Table	名前
コマンド・リスト・テーブル	DFHCLTxx
ファイル管理テーブル	DFHFCTxx
モニター管理テーブル	DFHMCTxx
プログラム・リスト・テーブル <small>注</small>	DFHPLTxx
リカバリー可能なサービス・テーブル	DFHRSTxx
システム・リカバリー・テーブル	DFHSRTxx
端末管理テーブル	DFHTCTxx
端末リスト・テーブル	DFHTLTxx
一時記憶テーブル	DFHTSTxx
トランザクション・リスト・テーブル	DFHXLtxx

### テーブルの接尾部の指定方法

各テーブルの名前の最初の 6 文字は固定されています。プログラム・リスト・テーブル DFHPLT を除き、SUFFIX オペランドを使用してテーブル名の最後の 2 文字を指定することができます。SUFFIX オペランドは、各テーブルの TYPE=INITIAL マクロで指定されます。

**注：**TYPE=INITIAL セクションは、プログラム・リスト・テーブル DFHPLT には関係ありません。DFHPLT に TYPE=INITIAL ステートメントを含めることができますが、そのステートメントは無効です。DFHPLT については、CICS が、PARMLIB または DFHTABLE メンバーの名前の中の DFHPLT に続く文字から、テーブルの接尾部を決定します。詳しくは、[プログラム・リスト・テーブル \(PLT\)](#) を参照してください。

接尾部は、1 つまたは 2 つの文字で構成されます。使用できる文字は A から Z、0 から 9、@ です。(NO や DY は使用できません。) システム初期化の際にシステムにロードされるテーブルのバージョンを選択するときには、システム初期設定パラメーターの適切なオペランドに接尾部を指定します。

例えば、次のようになります。

```
DFHSIT...,FCT=MY,...
```

**注：**TYPE=INITIAL マクロには、個別マクロの説明にはリストされていない STARTER オペランドがあります。STARTER=YES とコーディングすることにより、テーブルの接尾部で \$ および # 文字を使用できます。デフォルトは STARTER=NO です。このオペランドは、起動システム・モジュールでのみ使用してください。



### **TYPE=FINAL (テーブルの終了)**

ほとんどのテーブルは、TYPE=FINAL マクロで終了する必要があります。ここでも SIT が唯一の例外となります。

TYPE=FINAL マクロは、テーブルの終了を知らせるダミー項目を作成します。これはアセンブラーの END ステートメントの直前のステートメントでなければなりません。形式は常に以下ようになります。

	DFHxxT	TYPE=FINAL
--	--------	------------

## CICS 管理テーブルでのリソースの定義

一部の CICS リソースは CICS 管理テーブルで定義します。

これらのテーブルとそのリソース定義は、マクロを使用して作成します。マクロを使用して z/OS Communications Server 以外のネットワークと端末、VSAM 以外のファイル、データベース、およびリソースを定義し、それらを使ってモニターおよびシステム・リカバリーを行なう必要があります。VSAM ファイル、プログラム、マップ・セット、パーティション・セット、キュー、トランザクション、およびプロファイルのためには、RDO を使用する必要があります。

CICS 管理テーブルおよびマクロの概要 にリストされたそれぞれの CICS テーブルについて、以下の手順を実行します。

1. 必要なリソース定義をコーディングします。
2. CICS 提供のプロシージャ DFHAUPLE を使用してこれらの定義のアセンブルおよびリンク・エディットを行い、必要な CICS ロード・ライブラリー内にロード・モジュールを作成します。ロード・ライブラリーは、CICSTS56.SDFHLOAD または CICSTS56.SDFHAUTH のいずれかで、DFHAUPLE プロシージャの NAME パラメーターで指定する必要があります。CICS テーブルの作成に使用される CICS 提供のマクロによって、テーブルが 16MB ラインの上にロードされるかどうかが判定されます。TCT 以外のすべてのテーブルは、16MB ラインの上にロードされます。
3. システム初期設定パラメーターを使ってロード・モジュールの接尾部に名前を付けます。ほとんどの CICS テーブルでは、テーブルが必要ない場合、`tablename=NO` とコーディングできます。ただし、このルールには以下の例外があります。
  - CLT: CLT=NO を指定すると、CICS は DFHCLTNO のロードを試行します。CLT は、CICS を XRF と共に実行しているときに代替 CICS で常に必要となります。CLT はこのような場合のみ使用されます。
  - SIT: SIT=NO を指定すると、CICS は DFHSITNO のロードを試行します。SIT は常に必要であり、SIT システム初期設定パラメーターをコーディングすることにより接尾部を指定できます。
  - TCT: TCT=NO を指定すると、CICS はダミー TCT の DFHTCTDY をロードします。
  - TLT: 端末リスト・テーブルは、CSD のプログラム・エントリーによって指定され、システム初期設定パラメーターはありません。
  - MCT: MCT=NO を指定すると、CICS モニタリング・ドメインは、デフォルトのモニター管理テーブルを動的に作成します。これにより、モニタリングがオンで 1 つ以上のモニタリング・クラスがアクティブのとき、デフォルトのモニター管理テーブルのエントリーを常に使用できます。
4. CICS を XRF と共に実行している場合、アクティブ CICS 領域と代替 CICS 領域はテーブルの同じバージョンを共用します。ただし DASD 障害から保護するために、アクティブ CICS 領域と代替 CICS 領域をそれぞれ別のセットのロード・ライブラリーから実行することもできます。その場合には、管理テーブルを生成した後に別個のコピーを作成する必要があります。

278 ページの表 27 には、CICS ライブラリーでアセンブル、リンク・エディット、およびインストールが可能なすべての CICS テーブルがリストされています。

表 27. CICS ライブラリーでアセンブル、リンク・エディット、およびインストールが可能な CICS テーブル			
Table	モジュール名	省略語	必要なロード・ライブラリー
コマンド・リスト・テーブル	DFHCLTxx	CLT	SDFHAUTH
データ変換テーブル	DFHCNV	CNV	SDFHLOAD
ファイル管理テーブル	DFHFCTxx	FCT	SDFHLOAD
モニター管理テーブル	DFHMCTxx	MCT	SDFHLOAD
DL/I プログラム仕様ブロック	DFHPSBxx	PSB	SDFHLOAD
リカバリー可能なサービス・エレメント・テーブル	DFHRSTxx	RST	SDFHAUTH
システム初期設定テーブル	DFHSITxx	SIT	SDFHAUTH

表 27. CICS ライブラリーでアセンブル、リンク・エディット、およびインストールが可能な CICS テーブル (続き)

Table	モジュール名	省略語	必要なロード・ライブラリー
システム・リカバリー・テーブル	DFHSRTxx	SRT	SDFHLOAD
端末管理テーブル	DFHTCTxx	TCT	SDFHLOAD
端末リスト・テーブル	DFHTLTxx	TLT	SDFHLOAD
一時記憶テーブル	DFHTSTxx	TST	SDFHLOAD
トランザクション・リスト・テーブル	DFHXLTxx	XLT	SDFHLOAD

上記のテーブルのメンバーとして使用されるプログラム・リスト・テーブル (PLT)。ただし、PLT はアセンブルする必要がなくなりました。CICS は PLT のソース・コードを読み取り、それを使用して必要な PLT 処理を定義します。

テーブルを生成するマクロに SUFFIX=xx を指定することにより、それぞれの CICS 管理テーブルに対して複数のバージョンを作成できます。その後、この接尾部は、6 文字からなるデフォルトのロード・モジュール名に追加されます。

手軽に作業を開始できるように、CICS では、CICSTS56.SDFHSAMP ライブラリーの中に、[279 ページの表 28](#) にリストされているサンプルのテーブルが用意されています。

表 28. CICSTS56.SDFHSAMP ライブラリーにあるサンプルの CICS システム・テーブル

Table	接尾部	注
コマンド・リスト・テーブル (CLT)	1\$	XRF 領域のみ
モニター管理テーブル (MCT)	A\$	CICS AOR で使用
モニター管理テーブル (MCT)	F\$	CICS FOR で使用
モニター管理テーブル (MCT)	T\$	CICS TOR で使用
モニター管理テーブル (MCT)	2\$	
システム 初期設定テーブル (SIT)	\$	デフォルトのシステム 初期設定パラメーター
システム 初期設定テーブル (SIT)	6\$	
システム・リカバリー・テーブル (SRT)	1\$	
Terminal control table (TCT)	5\$	z/OS Communications Server 以外の端末のみ

CICS の実行中にテーブルを変更して再アセンブルすることは可能ですが、CICS をシャットダウンして再始動してからでなければ、新しいテーブルは CICS で使用できません。(コマンド・リスト・テーブル (CLT) は例外で、アクティブ CICS 領域または代替 CICS 領域のいずれかをシャットダウンしなくても、新しいテーブルを使用できます。)

### CICS に管理テーブルを定義する

テーブルの複数のバージョンをアセンブルおよびリンク・エディットし、(CNV 以外では) 接尾部を使用してそれらを区別できます。CICS で使用するバージョンを指定するには、始動時のオーバーライドとして、システム 初期設定パラメーターを `tablename=xx` の形式でコーディングします。

### このタスクについて

特別な要件のあるその他のテーブルは、プログラム・リスト・テーブル (PLT)、端末リスト・テーブル (TLT)、およびトランザクション・リスト・テーブル (XLT) です。TLT ごとに、プログラムの自動インストールをアクティブにする必要があります。あるいは、テーブル名 (接尾部を含む) をプログラム名として使用して CSD にプログラム・リソース定義を指定する必要があります。CSD 内にプログラム・リソース定義

がない場合、PLT または XLT が自動インストールされます。例えば、接尾部が AA の TLT (DFHTLTAA) を生成する CEDA コマンドは以下のようになります。

```
CEDA DEFINE PROGRAM(DFHTLTAA) GROUP(grpname) LANGUAGE(ASSEMBLER)
```

1 文字と 2 文字の接尾部について詳しくは、[275 ページの『テーブルの名前指定と接尾部指定』](#)を参照してください。

プログラム・リスト・テーブルと端末リスト・テーブルについて詳しくは、[プログラム・リスト・テーブル \(PLT\)](#)および [Terminal list table \(TLT\)](#)を参照してください。

z Systems 以外のプラットフォームで CICS と通信するためには、DFHCNV 変換テーブルが必要です。データ変換プロセスについて詳しくは、[型変換テーブルの定義](#)を参照してください。

### 管理テーブルのアセンブルおよびリンク・エディット: DFHAUPLE プロシージャ

テーブルのアセンブルおよびリンク・エディットを行うには、CICS 提供のサンプル・プロシージャ DFHAUPLE を呼び出すジョブを作成します。

#### このタスクについて

DFHAUPLE プロシージャでは、以下のライブラリーをオンラインにする必要があります。

表 29. DFHAUPLE プロシージャのライブラリー要件	
ライブラリー名	テーブルが必要となる対象
SYS1.MACLIB	すべて
CICSTS56.SDFHMAC	すべて
CICSTS56.SDFHLOAD	CLT、RST、および SIT を除くすべて
CICSTS56.SDFHAUTH	CLT、RST、および SIT
SMP/E グローバル・ゾーン	すべて
SMP/E ターゲット・ゾーン	すべて

テーブルのアセンブルとリンク・エディットを行った後、それらをシステム初期設定パラメーターで CICS に定義します。

#### DFHAUPLE プロシージャ内のステップ

DFHAUPLE プロシージャは、DFHISTAR ジョブを実行すると、ユーザーの CICS 環境に合わせて調整され、CICSTS56.XDFHINST ライブラリーに保管されます。

#### このタスクについて

DFHAUPLE プロシージャには、以下のステップが含まれます。

1. ASSEM: このステップでは、テーブル定義マクロを、ASM ステップおよび SMP ステップへの入力として使用する一時区分データ・セット (PDS) メンバーに配置します。以降の BLDMBR ステップで、その他のメンバーをこの一時 PDS に追加します。
2. ASM: このアセンブリー・ステップでは、SYSPUNCH 出力が一時順次データ・セットに格納されます。この出力は、IEBUPDTE 制御ステートメント、リンク・エディット制御ステートメント、SMP 制御ステートメント、およびオブジェクト・デックから構成されます。
3. BLDMBR: このステップでは、IEBUPDTE ユーティリティーが、その他のメンバーを ASSEM ステップで作成された一時 PDS に追加します。これらのメンバーには、アセンブリー・ステップからのリンク・エディット制御ステートメント、SMP 制御ステートメント、およびオブジェクト・デックが含まれます。
4. LNKEDT: リンク・エディット・ステップでは、PDS メンバー LNKCTL の内容を制御ステートメントとして使用します。ステップ 2 で生成されたオブジェクト・コードが一時 PDS から取得されます。出力は、プロシージャの NAME パラメーターで指定されたロード・ライブラリーに格納されます。CLT、RST、

および SIT に対しては NAME=SDFHAUTH を指定し、その他のすべてに対しては NAME=SDFHLOAD を指定する必要があります。

5. ZNAME: このステップでは、SMP/E JCLIN ジョブ・ステップに渡される一時データ・セットを作成します。それには、ターゲット・ゾーン名を定義する SET BDY コマンドが含まれます。これは更新するターゲット・ゾーンを SMP/E に知らせます。
6. SMP: SMP ステップでは、一時 PDS メンバーの MACROS、SMPCNTL、SMPJCL1、SMPJCL2、LNKCTL、SMPEOF、およびオブジェクト・デックを使用して、制御データ・セット (CDS) を更新します。
7. DELTEMP: この最終ステップでは、一時区分データ・セット &&TEMPPDS を削除します。

後でメンテナンスを適用するときに SMP で CICS テーブルを自動的に再アセンブルするには、このステップを正常に実行する必要があります。



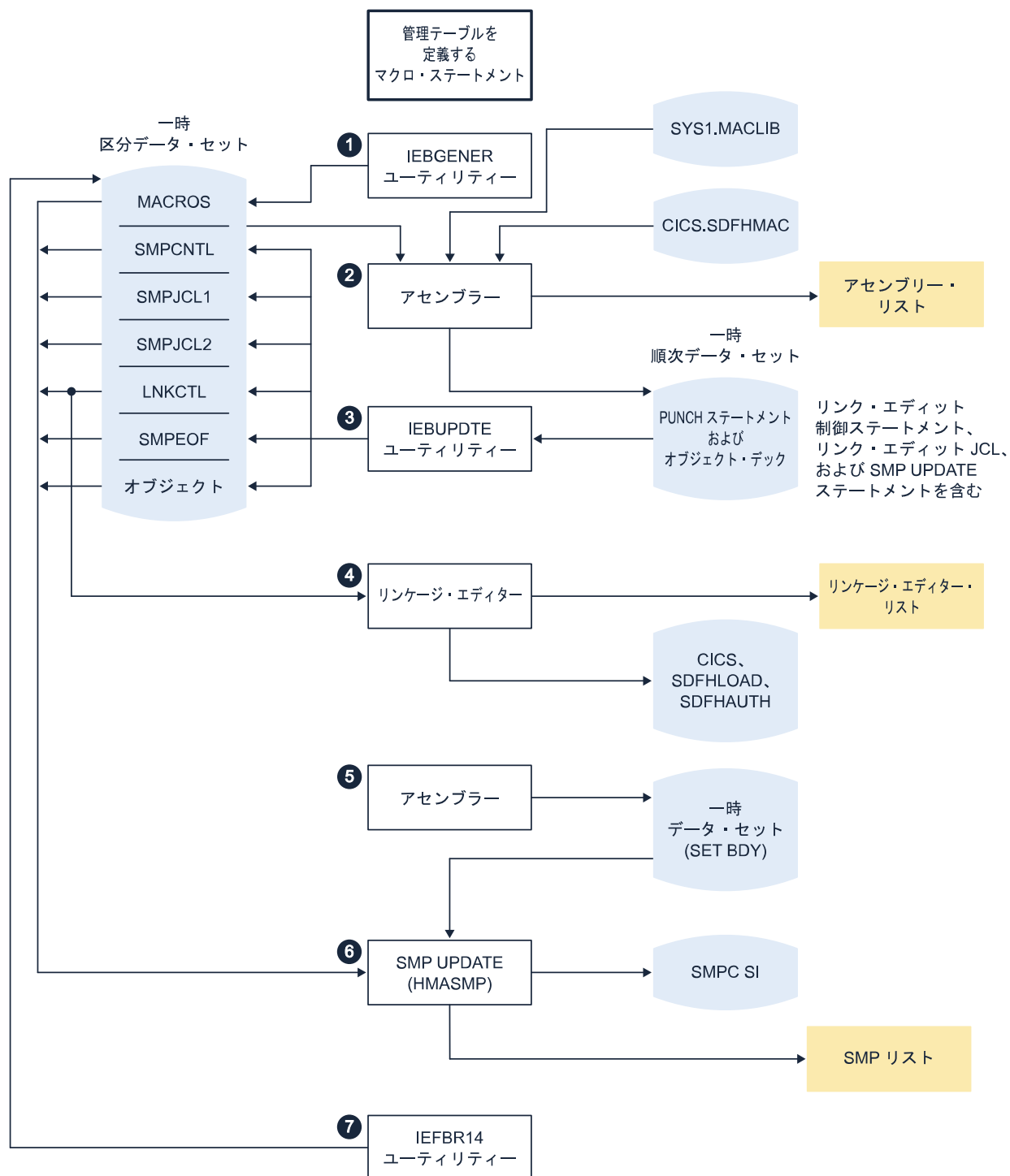


図 53. DFHAUPLE プロシーチャーを使用した管理テーブルのアセンブルおよびリンク・エディット

管理テーブルのためのサンプル・ジョブ・ストリーム

単一のジョブ・ストリームを使用して、CLT と RST を除くすべての CICS 管理テーブルをアセンブルおよびリンク・エディットすることができます。

以下のジョブ・ストリームを使用します。

```
//jobname      JOB      (accounting information),  
//             CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)  
//ASMTAB       EXEC     PROC=DFHAUPL[ ,NAME={SDFHLOAD|SDFHAUTH}]  
/*  
//ASSEM.SYSUT1 DD      *  
  
        Control table macro statements  
  
/*
```

図 54. 管理テーブルのためのジョブ・ストリーム

システム初期設定テーブルの場合のみ、このジョブに NAME=SDFHAUTH を指定します。他のすべてのテーブル (CLT と RST を除く) は、リンク・エディットして SDFHLOAD を作成します。

## CICS バンドルの定義

CICS バンドルとは、成果物や、バンドルとその依存関係を記述したマニフェストが含まれているディレクトリです。CICS バンドルは、関連したリソースをグループ化して管理する手段を提供します。

また、CICS バンドルは、リソース更新管理のためのバージョン管理方式を提供し、バンドル外部にある他のリソースへの依存関係を宣言できます。アプリケーション開発者は、アプリケーションのパッケージ化とデプロイメント、ビジネス・イベント、およびサービスのために CICS バンドルを使用できます。システム・プログラマーは、CICS ポリシーを定義するために CICS バンドルを使用できます。

### このタスクについて

CICS では、CICS バンドルを作成、パッケージ、およびデプロイするための 2 つの方法が用意されています。IBM CICS Explorer クライアントを使用してバンドル・プロジェクトを管理することができます。別の方法として、Maven または Gradle を使用してローカルでバンドルを作成してビルドし、API を使用してそれをデプロイすることもできます。

**CICS バンドルは、IBM CICS Explorer または IBM Developer for Z を使用して作成できます。**

バンドルは、バンドルのデプロイ時に CICS が動的に作成するリソースのセットを指定します。アプリケーションの前提条件システム・リソースをバンドルで指定することもできます。CICS は前提条件システム・リソースを動的に作成するわけではありませんが、それらが CICS 領域に存在するかどうかを検査できます。

CICS リソースは、CICS バンドルで定義され、バンドル・デプロイメントの一部として動的に作成されるため、その一部の特性が変更されます。このようにして作成された CICS リソースのライフサイクルを、BUNDLE リソースから切り離して管理することはできません。これらは、CICS バンドルを介して一つのまとまりとして管理する必要があります。バンドルとリソース・ライフサイクルの相互依存性のため、アプリケーションのどのリソース・セットのライフサイクルを CICS バンドルのライフサイクルに結び付けるかを、アプリケーション設計者は慎重に検討する必要があります。CICS バンドルでリソースを定義することの潜在的な影響についてより深く理解するには、[290 ページの『CICS バンドル内リソースの特性』](#)を参照してください。

個別 CICS バンドルを CICS 領域にデプロイしたり、プラットフォーム上にデプロイされたパッケージ化アプリケーションの一部として使用したりすることができます。あるいは、プラットフォーム上に直接デプロイすることもできます。CICS バンドルをアプリケーションまたはプラットフォームにデプロイする場合は、通常のプロセスで CICS バンドルを作成しますが、その後 CICS バンドルをアプリケーションの一部としてパッケージ化するか、プラットフォームに直接追加します。

プライベート・リソースとしてサポートされる CICS リソースを、アプリケーション用に設計されたその他のリソースやインポートされたリソースと組み合わせて使用する、プラットフォーム上にデプロイ

されるアプリケーションは、複数バージョンに適しています。以下のリソースは、複数バージョンに対応したアプリケーションの一部としてサポートされています。

- アプリケーションの一部である CICS バンドルで定義された PROGRAM リソース
- アプリケーションの一部である CICS バンドルで定義された LIBRARY リソース
- アプリケーションの一部である CICS バンドルで定義された PACKAGESET リソース
- アプリケーション・ユーザー・タスクを管理するためのタスク規則を定義するポリシー
- アプリケーション・エントリー・ポイントのステートメント
- アプリケーションに対して従属(またはインポート)として定義されたリソース

CICS バンドルは、マルチバージョン管理に適したリソースのみを含んでいれば、複数アプリケーションによって CICS 領域にインストールできます。

他のリソース・タイプを含んでいる CICS バンドルを、プラットフォーム内の同じ CICS 領域にインストールされた他のアプリケーションに組み込むことはできません。また、あるアプリケーションのために既にインストールされている CICS バンドルをプラットフォームに追加することもできません。マルチバージョン管理がサポートされず、プラットフォーム内のターゲット領域タイプにインストールされている CICS バンドルで既に使用されているリソースをアプリケーションが必要とする場合は、新しい CICS バンドル・プロジェクトを作成します。新しい CICS バンドルと依存関係を持つリソースを宣言するため、マニフェスト・ファイルにインポートを指定します。その後、新しい CICS バンドル・プロジェクトをアプリケーション・プロジェクトに組み込みます。

別の方法として、**CICS 提供の Maven または Gradle プラグイン**を使用して、**CICS バンドル**を作成、パッケージ、およびデプロイすることもできます。

この方法で作成した CICS バンドルは、IBM CICS Explorer または IBM Developer for Z を使用して作成されたものと似ていますが、以下の点が異なります。

- Maven または Gradle プラグインは、バンドル・パーツのサブセットのみをサポートしています (EAR、WAR、OSGi バンドルなど、Java 開発に最も関係したものを含む)。
- CICS バンドルが記述された Maven および Gradle のモジュールには、CICS Explorer でネイティブに作成された CICS バンドル・プロジェクトとは異なるメタデータと構造が含まれます。例えば、自動生成されるマニフェスト・ファイル (cics.xml) には、基本的な定義およびバージョンの情報が含まれます。そのため、CICS Explorer を使用してそれらを z/FS にエクスポートすることはできません。

## 手順

1. IBM CICS Explorer または IBM Developer for Z を使用して CICS バンドルを作成する場合:

- a) IBM CICS Explorer を使用して CICS バンドルを作成し、バンドルに組み込む成果物を定義します。  
[CICS Explorer 製品資料内の『Working with bundles』](#)の説明に従ってください。
- b) CICS バンドルを単独でデプロイする場合は、z/OS UNIX 上の適切なディレクトリーにデプロイします。  
そのディレクトリーに対する読み取り権限が CICS に必要です。  
CICS バンドルをアプリケーションまたはプラットフォームの一部としてデプロイする場合は、[CICS Explorer 製品資料内の『プラットフォームとアプリケーションの処理』](#)の説明に従って、CICS バンドルを組み込みます。
- c) CICS バンドルを単独でデプロイする場合は、BUNDLE リソースを定義して有効にし、CICS バンドルに対して使用可能にします。  
指定する属性の詳細については、[BUNDLE リソース](#)を参照してください。
- d) CICS バンドルをプラットフォームにデプロイする場合は、[CICS Explorer 製品資料内の『Working with bundles』](#)の説明に従って、以前のバージョンの CICS バンドルを除去した後、新しいバージョンの CICS バンドルをプラットフォームに追加します。
- e) CICS バンドルをアプリケーションの一部としてデプロイする場合は、[CICS Explorer 製品資料内の『Working with applications』](#)の説明に従って、アプリケーションをデプロイします。

2. Maven または Gradle プラグインを使用して CICS バンドルを作成してデプロイする場合は、[How it works: CICS bundle deployment API](#) を参照してください。

## タスクの結果

CICS はバンドル・ディレクトリー内のマニフェストを読み取り、CICS リソースを動的に作成します。また、CICS は、バンドル外の必須依存関係 (例えばプログラムやファイル) が CICS 領域に存在するかどうかを検査します。

## 次のタスク

CICS Explorer を使用して、インストールされた BUNDLE リソースとその関連リソースについて調べることができます。動的に作成された各リソースのリソース・シグニチャーは、それが BUNDLE リソースによって作成されたことを示します。

BUNDLE リソースによって動的に作成されたリソースの有効化、無効化、または破棄を直接行うことはできません。CICS バンドルを有効化、無効化、または破棄したり、使用可能または使用不可にしたりすると、それらのアクションはバンドル用に動的に作成されたリソースにも適用されます。CICS バンドルがプラットフォーム内のアプリケーションの一部である場合、そのプラットフォームまたはアプリケーションを有効化、無効化、または破棄したり、アプリケーションを使用可能または使用不可にしたりすると、それらのアクションは、CICS バンドルとそのバンドル用に動的に作成されたリソースに適用されます。

CICS Explorer のエディターを使用して、CICS バンドルで指定された成果物、およびバンドルのデプロイ時に CICS が動的に作成するリソースの定義を変更できます。変更を適用するには、変更した CICS バンドルを z/OS UNIX 上のディレクトリーにデプロイし、BUNDLE リソースを使用不可にしてから無効にして破棄した後、BUNDLE リソースを再度インストールして使用可能にします。

## バンドルにデプロイ可能な成果物

CICS バンドルに定義してデプロイできる成果物には、アプリケーションとシステムのイベント、Atom フィード、チャンネル・ベースのサービス、CICS ポリシー、CICS プログラム、OSGi バンドル、XML ベースのサービスおよびトランザクションが含まれます。これらの成果物はそれぞれ 1 つ以上の CICS リソースによって表され、それらのリソースはバンドル・デプロイメントの一部として動的に作成されます。

動的に作成されたリソースのリソース・シグニチャーは、それらがバンドル・デプロイメント時に作成されたことを示し、BUNDLE リソースの名前が含まれます。

CICS バンドルの中の CICS リソースを定義してデプロイすると、それらのリソースの管理とライフサイクルは、それらのリソースを動的に作成した CICS バンドルに委任されます。動的に作成されたリソースは、独立して存在するのではなく、実際にはバンドルまたはアプリケーションの一部です。

- 動的に作成されたリソースを照会できますが、それらを直接有効化、無効化、または破棄することはできません。
- CICS バンドルを有効化、無効化、または破棄すると、そのアクションはバンドル用に動的に作成されたリソースにも適用されます。
- CICS バンドルがプラットフォーム上のアプリケーションの一部である場合、そのプラットフォームまたはアプリケーションを有効化、無効化、または破棄すると、それらのアクションは、CICS バンドルとそのバンドル用に動的に作成されたリソースに適用されます。
- CICS Explorer のエディターを使用して、CICS バンドルで指定された成果物、およびバンドルに定義されているリソースの定義を変更します。

CICS バンドルのリソースを管理することの影響については、[290 ページの『CICS バンドル内リソースの特性』](#)を参照してください。

LIBRARY、PACKAGESET、または PROGRAM リソースの定義を CICS バンドル内に作成し、その CICS バンドルをアプリケーションの一部としてプラットフォームにデプロイすると、動的に作成された LIBRARY、PACKAGESET、または PROGRAM リソースは、そのアプリケーションのそのバージョン専用のものになります。専用リソースは、プラットフォームにインストールされている他のアプリケーションやバージョンからは使用できず、CICS 領域内の他のタスクからも使用できません。そのため、専用リソース名をインストール済み環境内で固有にする必要はありません。専用リソースとしてサポートされている CICS リソースを他のリソース (ポリシーやアプリケーション・エントリー・ポイントなど) と共に使用するアプリケーションは、複数バージョン管理に適格です。専用リソースについて詳しくは、[298 ページの『アプリケーション・バージョンの専用リソース』](#)を参照してください。

以下の成果物は、CICS バンドルを使用してデプロイできます。リソース・タイプは、バンドル・マニフェストに URI として定義されています。

## アプリケーション・イベントまたはシステム・イベント

CICS イベント処理については、CICS Explorer を使用して、イベント・バインディング、キャプチャー仕様、および EP アダプターを定義し、CICS バンドルに入れてデプロイできます。ビジネス・イベントのために BUNDLE リソースをインストールすると、CICS によって適切な EVENTBINDING、CAPTURESPEC、および EPADAPTER リソースが CICS 領域に動的に作成されます。CICS Explorer の CICS イベント・バインディング・エディターを使用してビジネス・イベントを定義し、それを CICS バンドルにデプロイする手順については、[Creating an event binding](#) を参照してください。

### URI:

<http://www.ibm.com/xmlns/prod/cics/bundle/EPADAPTER>  
<http://www.ibm.com/xmlns/prod/cics/bundle/EPADAPTERSET>  
<http://www.ibm.com/xmlns/prod/cics/bundle/EVENTBINDING>

## Atom フィード

CICS から Atom フィードを提供するには、CICS Explorer を使用して Atom 構成ファイルを作成し、それを CICS バンドルに入れてデプロイします。Atom フィードのために BUNDLE リソースをインストールすると、CICS によって適切な ATOMSERVICE、XMLTRANSFORM、および URIMAP リソースが CICS 領域に動的に作成されます。Atom 構成ウィザードおよび CICS Explorer の Atom 構成エディターを使用して Atom フィードを定義し、それを CICS バンドルにデプロイする手順については、[Atom フィード用の CICS 定義のセットアップ](#) を参照してください。

### URI:

<http://www.ibm.com/xmlns/prod/cics/bundle/ATOMSERVICE>

## チャンネル・ベースのサービス

チャンネル・ベースのサービスは、IBM Developer for Z 内の Service Component Architecture (SCA) ツールを使用して、コンポーネントとして記述され、一緒にアセンブルされる CICS アプリケーションです。SCA ツールは、複合アプリケーションを CICS に CICS バンドルとしてデプロイします。これらのサービスは、**INVOKE SERVICE** API コマンドを使用してコンテナ内のバイナリー・データをチャンネルで渡す、他の CICS アプリケーションにのみ使用できます。IBM Developer for Z を使用してチャンネル・ベースのサービスを作成する手順については、[チャンネル・ベース・サービスの作成](#) を参照してください。

### URI:

<http://www.ibm.com/xmlns/prod/cics/bundle/SCACOMPOSITE>

## File

CICS バンドルに FILE リソースの定義を作成できます。CICS バンドル内の定義では、次のファイル・タイプがサポートされます。

- VSAM ファイル (これには、CICS 保守、ユーザー保守、およびカップリング・ファシリティ・データ・テーブルを参照するファイルと、VSAM データ・セットを参照するファイルが含まれます)
- リモート VSAM ファイル
- リモート BDAM ファイル

ファイル定義を含む BUNDLE リソースをインストールすると、CICS によって FILE リソースが CICS 領域に動的に作成されます。

### URI:

<http://www.ibm.com/xmlns/prod/cics/bundle/FILE>

## JSON 変換

JSON 変換は、JSON を変換するためのリンク可能インターフェースを持つアプリケーションによって使用されます。JSON アシスタントは、言語構造または JSON スキーマを使用して JSON バインディングを生成します。また、バンドルも作成します。BUNDLE リソースをインストールするときに、CICS は JSON バインディングおよびスキーマの場所を定義する JSONTRANSFRM バンドル・パーツを動的に作成します。

### URI:

<http://www.ibm.com/xmlns/prod/cics/bundle/JSONTRANSFRM>

## JVM サーバー

CICS バンドルに JVMSERVER リソースの定義を作成できます。JVM サーバーの JVM プロファイルは、リソース定義と共に CICS バンドルにパッケージされるため、JVM プロファイルは CICS バンドルがイ



インストールされている任意の CICS 領域で使用できます。OSGi JVM サーバー、Axis2 JVM サーバー、Liberty JVM サーバー用のサンプル・テンプレートを使用して JVM プロファイルを作成するか、既存の JVM プロファイルをワークスペース内またはローカル・ファイル・システム内から CICS バンドルにインポートできます。JVM プロファイルは、システムの要件に合わせてカスタマイズできます。JVM サーバー定義を含む BUNDLE リソースをインストールすると、CICS によって JVMSERVER リソースが CICS 領域に動的に作成されます。

**URI:**

<http://www.ibm.com/xmlns/prod/cics/bundle/JVMSERVER>

**ライブラリー**

CICS バンドルに LIBRARY リソースの定義を作成できます。動的プログラム LIBRARY 連結の定義を含む BUNDLE リソースをインストールすると、CICS によって LIBRARY リソースが CICS 領域に動的に作成されます。

**URI:**

<http://www.ibm.com/xmlns/prod/cics/bundle/LIBRARY>

**Node.js アプリケーション**

Node.js アプリケーションは、BUNDLE リソース内でデプロイおよび管理されます。BUNDLEPART SPI を使用すると、これらが BUNDLEPARTS として表示されます。

**URI:**

<http://www.ibm.com/xmlns/prod/cics/bundle/NODEJSAPP>

**OSGi バンドル**

OSGi バンドルとしてパッケージされている Java アプリケーションは、CICS バンドルにデプロイして JVM サーバーで実行できます。OSGi バンドルおよびサービスを含む BUNDLE リソースをインストールすると、OSGi フレームワークの OSGi バンドルとサービスを表す OSGIBUNDLE および OSGISERVICE リソースが CICS によって動的に作成されます。CICS はこれらのリソースを使用して OSGi バンドルと OSGi サービスのライフサイクルを管理します。IBM CICS SDK for Java を使用して OSGi バンドルを含む CICS バンドルを作成する手順については、[IBM CICS SDK for Java を使用したアプリケーションの開発を参照してください](#)。

**URI:**

<http://www.ibm.com/xmlns/prod/cics/bundle/OSGIBUNDLE>

**パッケージ・セット**

CICS バンドルに CICS プラットフォーム、アプリケーション、またはアプリケーション・バインディング・プロジェクトがデプロイされる場合、PACKAGESET リソースの定義を作成できます。PACKAGESET リソースを使用して、CICS プラットフォームで実行されるアプリケーション・タスクによって行われる SQL 要求の修飾に使用する Db2 コレクションを定義します。

CICS アプリケーションまたはアプリケーション・バインディングと共に PACKAGESET リソースが含まれる BUNDLE リソースをインストールする場合、アプリケーション専用になります。CICS は、すべてのアプリケーション・タスクによって使用される Db2 コレクションを定義します。

CICS プラットフォームと共に PACKAGESET リソースが含まれる BUNDLE リソースをインストールする場合、CICS は、CICS アプリケーションに対して特定の Db2 コレクションが定義されないプラットフォームで実行されるすべてのアプリケーション・タスクによって使用される Db2 コレクションを定義します。

**URI:**

<http://www.ibm.com/xmlns/prod/cics/bundle/PACKAGESET>

**パイプライン**

CICS バンドルに PIPELINE リソースの定義を作成できます。パイプラインのパイプライン構成ファイルは、リソース定義と共に CICS バンドルにパッケージされます。パイプライン構成ファイルは、サービス・プロバイダーとサービス・リクエスターの CICS 提供のサンプル構成ファイルの 1 つを使用して作成するか、ローカル・ファイル・システムから既存の構成ファイルをインポートできます。パイプライン定義を含む BUNDLE リソースをインストールすると、CICS によって PIPELINE リソースが CICS 領域に動的に作成されます。CICS バンドル内で定義された PIPELINE リソースは、CICS バンドル内で定義されたか、パイプライン・スキャンによって動的に作成された WEBSERVICE リソースと一緒に使用する必要があります。



**URI:**

<http://www.ibm.com/xmlns/prod/cics/bundle/PIPELINE>

**ポリシー**

CICS Explorer を使用して、ポリシーを定義し、CICS バンドルに入れてデプロイできます。ポリシー・タスク規則をデプロイすることにより、CPU 消費量が多すぎる場合やストレージ割り振りが多すぎる場合など、個々の CICS ユーザー・タスクがしきい値を超える場合に実行するアクションを定義することができます。ポリシー・システム規則をデプロイすることにより、関係がある何らかの事象(リソース状態変更、しきい値の超過、通常と異なるシステム状態やアクション)が CICS システム内で発生した時点で実行するアクションを定義することができます。ポリシーのために BUNDLE リソースをインストールすると、CICS によって、CICS 領域の管理に使用されるルールセットにポリシー・ルールが動的に追加されます。

**URI:**

<http://www.ibm.com/xmlns/prod/cics/bundle/POLICY>

**プログラム**

高水準プログラミング言語用に CICS バンドルに PROGRAM リソースの定義を作成できます。プログラム定義を含む BUNDLE リソースをインストールすると、CICS によって PROGRAM リソースが CICS 領域に動的に作成されます。

**URI:**

<http://www.ibm.com/xmlns/prod/cics/bundle/PROGRAM>

**TCP/IP サービス**

CICS バンドルに TCPIPSERVICE リソースの定義を作成できます。TCP/IP サービス定義を含む BUNDLE リソースをインストールすると、CICS によって TCPIPSERVICE リソースが CICS 領域に動的に作成されます。

**URI:**

<http://www.ibm.com/xmlns/prod/cics/bundle/TCPIPSERVICE>

**トランザクション**

CICS バンドルに TRANSACTION リソースの定義を作成できます。トランザクション定義を含む BUNDLE リソースをインストールすると、CICS によって TRANSACTION リソースが CICS 領域に動的に作成されます。

**URI:**

<http://www.ibm.com/xmlns/prod/cics/bundle/TRANSACTION>

**URI マップ**

CICS バンドルに URIMAP リソースの定義を作成できます。URI マップ定義を含む BUNDLE リソースをインストールすると、CICS によって URIMAP リソースが CICS 領域に動的に作成されます。

**URI:**

<http://www.ibm.com/xmlns/prod/cics/bundle/URIMAP>

**Web アプリケーション**

WAR ファイル、EBA ファイル、または EAR ファイルでパッケージされている Web アプリケーションは、CICS バンドルにデプロイして Liberty JVM サーバーで実行できます。Web アプリケーションを含む BUNDLE リソースをインストールすると、CICS によって適切な Liberty アプリケーション定義が installedApps.xml ファイルに動的に作成されます。Liberty JVM サーバーはこれらの定義を使用して Web アプリケーションを動的に作成し、Web アプリケーションのライフサイクルは、BUNDLE リソースを使用して制御できます。IBM CICS SDK for Java を使用して Web アプリケーションを含む CICS バンドルを作成する手順については、[JVM サーバーへのアプリケーションのデプロイ](#)を参照してください。

**URI:**

<http://www.ibm.com/xmlns/prod/cics/bundle/WARBUNDLE>

<http://www.ibm.com/xmlns/prod/cics/bundle/EBABUNDLE>

<http://www.ibm.com/xmlns/prod/cics/bundle/EARBUNDLE>

## Web サービス

CICS バンドルに WEBSERVICE リソースの定義を作成できます。Web サービス定義を含む BUNDLE リソースをインストールすると、CICS によって WEBSERVICE リソースが CICS 領域に動的に作成されます。WEBSERVICE リソース定義と共に、Web サービス・バインディング・ファイルと WSDL 文書または WSDL アーカイブ・ファイルをインポートして、これらをバンドルにパッケージできます。URIMAP リソースと別名トランザクションを生成して Web サービスをサポートすることもできます。Web サービス・プロバイダーの場合は、PROGRAM リソース定義をバンドルに組み込むこともできます。PROGRAM リソースの定義は、CICS Explorer のリソース・エディターを使用して編集できます。

### URI:

<http://www.ibm.com/xmlns/prod/cics/bundle/WEBSERVICE>

## XML ベースのサービス

XML ベースのサービスは、通常、XML を使用して他のアプリケーションとインターフェースし、バインディングを使用してデータ変換を行う Web サービス・プロバイダーまたはリクエスターのアプリケーションです。XML ベース・サービスは、**INVOKE SERVICE** API コマンドを使用する CICS アプリケーションまたは外部ネットワーク上のビジネス・サービスで使用できます。IBM Developer for Z 内の SCA ツールを使用して Web サービスを作成すると、Web サービスを CICS バンドルとしてデプロイできます。Web サービスを含む BUNDLE リソースをインストールすると、CICS によって URIMAP リソースや WEBSERVICE リソースなどのさまざまなリソースが動的に作成されます。IBM Developer for Z を使用して XML ベースのサービスを作成する手順については、[XML ベース・サービスの作成](#)を参照してください。

XML ベース・サービスは、**TRANSFORM** API コマンドを使用して XML のアプリケーション・データをマップするアプリケーションになることもできます。XML アシスタントは言語構造または XML スキーマを使用して XML バインディングを生成します。バンドルを作成することもできます。BUNDLE リソースをインストールすると、CICS は XML バインディングおよびスキーマの場所を定義する XMLTRANSFORM リソースを動的に作成します。

### URI:

<http://www.ibm.com/xmlns/prod/cics/bundle/XMLTRANSFORM>

リソース・ライフ・サイクル・マネージャー・ドメインでコールバック・インターフェースを使用して、サポート対象のリソース・タイプのリストを拡張できます。このインターフェースを使用すると、ベンダーは新しいユーザー・リソース・タイプを作成し、BUNDLE リソースでそれらを管理できます。

## 関連概念

### [バンドル内の zFS 成果物の参照](#)

いくつかの CICS リソースが、外部 zFS 成果物を参照して詳しい構成情報を探します。例えば、JVMSERVER リソースでは JVM プロファイルが必要となり、PIPELINE リソースではパイプライン構成ファイルが必要となります。これらのリソースが CICS バンドルに定義されている場合、必要となる zFS 成果物も CICS バンドルに保管され、相対 zFS 名を使用して参照される必要があります。

### [CICS バンドルのマニフェストの内容](#)

それぞれの CICS バンドルには、バンドルの内容を記述するバンドル・マニフェストが含まれています。バンドル・マニフェストには、バンドルと、プロビジョン時に CICS 領域で設計または変更するリソース、さらに CICS バンドルを正常に使用可能にするために必要な依存関係が記述されます。

### [バンドルのスコーピング](#)

BUNDLE リソース定義には、関連する BUNDLE リソースに有効範囲を適用する方法として、BASESCOPE 属性があります。この属性は、プラットフォームにデプロイされているバンドルに使用したり、サービス・コンポーネント・アーキテクチャー (SCA) 複合アプリケーションが含まれているバンドルの SCA ドメインの設定に使用したりできます。

### [CICS リスタート後の OSGi バンドル・リカバリー](#)

OSGi バンドルを含む CICS 領域をリスタートすると、CICS は BUNDLE リソースをリカバリーし、OSGi バンドルを JVM サーバーのフレームワークにインストールします。

### [バンドルのセキュリティ](#)

定義から作成された BUNDLE リソースと、アプリケーションまたはプラットフォームをインストールするときに作成した BUNDLE リソースでは、適用されるセキュリティ・プロファイルとセキュリティ検査が異なります。

### [CICS プロジェクトでの変数](#)

変数を使用することで属性値を素早く簡単に変更することができます。各環境に固有のプロパティ・ファイルを使用して変数を解決できるので、複数環境へのデプロイメントが単純化されます。

## 関連資料

### 変数およびプロパティ・ファイルの定義

変数およびそれらに関連付けられたプロパティ・ファイルを定義する場合、変数の置換が正常に行われるように、これらのルールに従います。

## 関連情報

### CICS バンドル内リソースの特性

CICS リソースは、CICS バンドルで定義され、バンドル・デプロイメントの一部として動的に作成されるため、その一部の特性が変更されます。アプリケーションのアーキテクチャーを設計する、あるいは CICS バンドルで定義する既存のアプリケーションからリソースを選択する場合、これらの重要な考慮事項に注意してください。

### アプリケーション・バージョンの専用リソース

特定の CICS リソースを、プラットフォームにインストールするアプリケーションの一部として CICS バンドルで定義した場合、そのリソースは、そのアプリケーションのそのバージョン専用のものになります。そのため、これらのタイプのリソースは、同時に複数を同じ名前で同じプラットフォーム・インスタンス上にインストールできます。

## CICS バンドル内リソースの特性

CICS リソースは、CICS バンドルで定義され、バンドル・デプロイメントの一部として動的に作成されるため、その一部の特性が変更されます。アプリケーションのアーキテクチャーを設計する、あるいは CICS バンドルで定義する既存のアプリケーションからリソースを選択する場合、これらの重要な考慮事項に注意してください。

CICS バンドル内にリソース定義を作成できる CICS リソース・タイプの完全なリストは次のとおりです。

- FILE
- JVMSERVER
- LIBRARY
- PIPELINE
- PROGRAM
- TCPIPService
- TRANSACTION
- URIMAP
- WEBSERVICE

OSGi のバンドルとイベントなど、CICS バンドルに定義してデプロイできるすべての成果物のリストについては、[バンドルにデプロイ可能な成果物を参照してください](#)。

## バンドルとリソースの相互依存関係

CICS バンドルを使用する場合、インストールされたリソースのライフサイクルは、バンドルのライフサイクルに従います。CICS バンドルを選択可能/選択不可にしたり、使用可能/使用不可にしたり、または破棄したりすると、その CICS バンドルによって動的に作成されたすべてのリソースに対して同じアクションが適用されます。さらに、CICS バンドルがアプリケーション・バンドルに含まれている場合、CICS バンドルをアプリケーションの一部として管理することになります。

CICS バンドルがリソースを CICS 領域内に動的に作成してインストールする場合、そのリソースの状態は、そのリソースのインストールに使用された CICS バンドルを通じて管理されます。動的に作成されたリソースを使用可能/使用不可にするには、CICS バンドルに対してそのアクションを実行します。アプリケーションの一部として管理される CICS バンドル内のリソースに関しては、アプリケーション・エントリー・ポイントを宣言することによっても、リソースに対するユーザーのアクセスを制御します。PROGRAM、TRANSACTION、および URIMAP リソースをアプリケーション・エントリー・ポイントとして宣言し、それをユーザーに対して使用可能または使用不可に設定できます。アプリケーション・エントリー・ポイントを使用することにより、アプリケーションを (CICS バンドルとそのリソースを含めて) インストールし、

その後、それらを有効にしてインストールを検証できます。サービスをユーザーに提供する場合は、CICS バンドルを含んだアプリケーションを CICS Explorer を使用して使用可能にします。この操作により、アプリケーション・エントリー・ポイントと、CICS バンドル内のリソースが、呼び出し元に対して使用可能になります。アプリケーション・エントリー・ポイントを含んだ個々の CICS バンドルを使用可能にするには、**EXEC CICS SET BUNDLE** コマンドを使用する方法もあります。

一般に、CICS バンドルによって動的に作成されてインストールされたリソースの属性に直接変更を加えることはできません。必要な変更が加えられた新しいバージョンの CICS バンドルをインストールする必要があります。しかし、FILE、TCPIP SERVICE、JVM SERVER の各リソースのように、CICS バンドルによってインストールされた一部の長期間リソースについては、CICS システム・コマンドを使用することによって直接に変更を加えることができます。ただし、変更内容は、CICS をウォーム・リスタートした後に回復されません。

JVM SERVER リソースおよび TCPIP SERVICE リソースについては、CICS バンドルで定義されたリソースの状態を変更することに関する規則の別の例外も当てはまります。デフォルトでは、これらのリソースは、それらを使用している現行タスクがすべて完了してからでないと無効にされません。これらのリソースを即時に除去する必要があるシチュエーションでは、CICS 領域内に動的に作成されたリソースに対して SET JVM SERVER の PURGE、FORCEPURGE、または KILL コマンド、あるいは SET TCPIP SERVICE IMMCLOSE コマンドを使用することにより、リソースがまだ使用中であっても使用不可化プロセスを強制的に完了させることができます。動的に作成されたリソースに対してこれらのコマンドを発行できるのは、CICS バンドル、またはそれと一緒にデプロイされたアプリケーションを無効化する操作を試行した後のみです。

バンドルとリソース・ライフサイクルは相互に依存しているため、アプリケーション・アーキテクトは、アプリケーションのどのリソースのセットのライフサイクルを CICS バンドルのライフサイクルに結び付けるかを考慮する必要があります。次の 2 つのシナリオでは、バンドルされたリソースの編成により、アプリケーションを管理する方法が決まります。

- プログラムに関連付けられた URI マップを、プログラムを無効にしないで修正する場合。修正できるようにするには、URI マップとプログラム・リソースを異なる別個の CICS バンドルにパッケージ化することができます。
- CICS バンドルによって複数の CICS 領域にインストールされたプログラムを、CICSplex 全体にわたってプログラムを無効にする必要なしに再ロードする場合。再ロードするときは、CICS 領域ごとに、1 回に 1 つの領域に対してプログラムの変更を個別に適用できます。変更を適用している間、現在更新中の領域をバイパスするように作業を構成できます。

CICS 領域内の特定の CICS リソースが使用可能になることがアプリケーションまたは CICS バンドルによって求められる場合に、そのリソースのライフサイクルをその CICS バンドルのライフサイクルと結びけないようにするには、そのリソースをその CICS バンドルと依存関係にあるものとして追加します。この依存関係(またはインポート)は、バンドル・マニフェストの `<import>` エレメントの中で指定します。必要なインポート対象の 1 つが CICS 領域に存在しない場合に CICS がバンドルのインストール時に取る処置を指定できます。例えば、CICS バンドルを使用不可のままにする、使用可能にするが警告メッセージを出す、などです。依存関係について詳しくは、[304 ページの『CICS バンドルのマニフェストの内容』](#)を参照してください。

CICS バンドル内で定義されているインストール済みリソースは、リソース定義によって同じ名前では上書きできません。リソースが含まれているバンドル・パーツがインストールされている場合に別の CICS バンドルを使用するか、RDO の EXEC CICS CREATE コマンドまたは他のメソッドを使用して同じリソースをインストールしようとする、と、要求は拒否されます。CICS バンドル内で定義されている同じリソースでインストール済みリソースを上書きすることもできません。CICS バンドルをインストールしようとしたときに同じ名前とリソース・タイプのリソースが存在する場合、そのリソースを含むバンドルはインストールされず、CICS バンドルは使用不可状態になります。そのため、CICS バンドル内で定義されているリソースは、同じリソース名によって誤って上書きされることはありません。

## バンドル内の専用リソース

特定のリソース・タイプの定義を CICS バンドル内に作成し、その CICS バンドルをアプリケーションの一部としてプラットフォームにデプロイすると、動的に作成されたリソースは、そのアプリケーションのそのバージョン専用のものになります。専用リソースは、プラットフォームにインストールされている他のアプリケーションやバージョンからは使用できず、CICS 領域内の他のタスクからも使用できません。この方法でデプロイしたリソースには固有の名前を付ける必要がありません。また、以前のバージョンのリソ



ースを無効にする必要なしに、アプリケーション・バージョンの一部として複数のバージョンのリソースをデプロイできます。専用リソースの詳細とサポートされるリソース・タイプについては、[アプリケーションの専用リソース](#)を参照してください。

## プラットフォーム・バンドルと一緒にデプロイされるリソース

TCIPSERVICE リソース、JVM サーバー、およびパイプラインは、プラットフォーム内の複数の CICS 領域で必要になることの多いリソースで、複数のアプリケーションを実行するために使用されることがあります。良い習慣として、プラットフォームの拡張を容易にするために、TCIPSERVICE リソース、JVMSEVER リソース、および PIPELINE リソースの CICS バンドルを、アプリケーションの一部としてではなく、プラットフォーム・バンドルの一部としてデプロイすることをお勧めします。これらのリソースをインストールする、プラットフォーム内の CICS 領域を指定できます。新規のリソースについては、プラットフォームのインストール後に、稼働中のプラットフォームに対して新しい CICS バンドルを追加できます。

PIPELINE リソースおよび JVMSEVER リソースを含む特定の CICS リソースは、それらに依存する他のリソースをインストールする前にインストールして有効にする必要があります。これらのリソースを使用すると、プラットフォーム・アーキテクチャーを使用して依存関係を管理することができます。まず、必要なリソースの定義を含んだ CICS バンドルをプラットフォーム・バンドルの一部としてデプロイし、次に、依存するリソースを含んだ CICS バンドルを、プラットフォームにデプロイするアプリケーションの一部としてデプロイします。このアーキテクチャーにより、リソースを CICS 領域に最初にインストールする際や、CICS 領域を再始動する際に、必要なリソースと依存するリソースを確実に正しい順序でインストールして有効にすることができます。

## 大/小文字混合のリソース名

CICS で大/小文字混合の名前がサポートされる場合、CICS バンドルのリソースにも大/小文字混合の名前を使用できます。ただし、ご使用のファイル・システムによっては、大/小文字が異なる同じ名前のファイルが、同じ CICS バンドル・プロジェクトに存在できない場合もあります。大/小文字が異なる同じリソース名を使用する必要がある場合は、それらのリソースを別個の CICS バンドル・プロジェクトに定義してください。

## リソース名の中のエスケープ文字

CICS Explorer で CICS バンドルを保管すると、異なるファイル・システムでもリソースを使用できるように、リソース名の中の特定の文字がエスケープされます。CICS Explorer で CICS バンドルを展開して CICS バンドル内で定義されたリソースを表示すると、リソース名はエスケープ・シーケンスで表示されます。編集のためにリソースを開くと、リソース名は元のアンエスケープされた文字で表示されます。

CICS Explorer で CICS バンドルを保管するとき、リソース名の中でエスケープされる文字は次のとおりです。

表 30. CICS バンドル内のリソース名におけるエスケープ・シーケンス	
文字	エスケープ・シーケンス
.(ピリオド)	%2E
/(スラッシュ)	%2F
&(アンパーサンド)	%26
? (疑問符)	%3F
:(コロン)	%3A
(縦棒)	%7C
" (二重引用符)	%22
;(セミコロン)	%3B

表 30. CICS バンドル内のリソース名におけるエスケープ・シーケンス (続き)

文字	エスケープ・シーケンス
< (より小)	%3C
> (より大)	%3E
¬ (論理否定)	%AC

## FILE リソース

CICS バンドル内の定義では、次のファイル・タイプがサポートされます。

- VSAM ファイル (これには、CICS 保守、ユーザー保守、およびカップリング・ファシリティ・データ・テーブルを参照するファイルと、VSAM データ・セットを参照するファイルが含まれます)
- リモート VSAM ファイル
- リモート BDAM ファイル

CICS バンドルから動的に作成される FILE リソースの初期状況は、そのリソースを定義する CICS バンドルの初期状況から導出されます。そのため、FILE リソースの STATUS に UNENABLED を定義しても、アプリケーションがそれらのファイルを暗黙的に開くことを禁止することはできません。

動的に作成される FILE リソースでは、PASSWORD 属性はサポートされません。

CICS バンドル内で定義されているファイルをインストールすると、そのファイルはカタログに追加されます。CICS は、ウォーム・リスタートまたは緊急リスタート時に、バンドルによってインストールされたファイルをカタログからリカバリーします。これらのリカバリーされたファイルは、CICS リカバリーでのみ使用されます。CICS によるリカバリーが完了すると、これらのすべてのファイルは削除されます。バンドルが再作成されるときに、CICS はバンドル・ディレクトリーから定義を取得して新しいファイルを作成します。カタログからリカバリーされたファイルに保持ロックがある場合、そのファイルはリカバリーの完了時に削除できません。その結果、ファイルは孤立し (どのバンドルにも属さない)、CICS のリスタート後に手動で削除する必要があります。

## JVMSERVER リソース

CICS バンドルにパッケージされる JVM サーバーの場合、JVM プロファイルはリソース定義と一緒に CICS バンドルにパッケージされます。したがって、CICS 領域に JVM サーバーをインストールするとき、CICS 領域のローカル JVM プロファイル・ディレクトリーに JVM プロファイルをセットアップする必要はありません。

JVM サーバーを作成する際は、OSGi JVM サーバー、Axis2 JVM サーバー、Liberty JVM サーバー用のサンプル・テンプレートを使用して JVM プロファイルを作成するか、既存の JVM プロファイルをワークスペース内またはローカル・ファイル・システム内から CICS バンドルにインポートすることができます。

JVMSERVER リソースを CICS バンドル内で定義する場合、JVM プロファイル名に使用できる文字はより限定的になります。詳しくは、JVMSERVER 属性を参照してください。JVM プロファイルが保管されているディレクトリーのディレクトリー名は 240 文字に制限されています。これは、CICS バンドル内で定義されていない JVM プロファイルに適用されるのと同じ制限です。

CICS バンドル内で定義した JVM サーバーは、その中で実行する OSGi バンドルまたは Java アプリケーションの他のアプリケーション成果物をインストールする前に、インストールして有効にする必要があります。良い習慣として、まず JVMSERVER リソースの定義を含んだ CICS バンドルをプラットフォーム・バンドルの一部としてデプロイし、次に OSGi バンドルまたは他の Java アプリケーション成果物を含んだ CICS バンドルを、プラットフォームにデプロイするアプリケーションの一部としてデプロイすることをお勧めします。このアーキテクチャーにより、リソースを CICS 領域に最初にインストールする際や、CICS 領域を再始動する際に、JVM サーバーと Java アプリケーション・リソースを確実に正しい順序でインストールして有効にすることができます。



## LIBRARY リソース

LIBRARY リソースを CICS バンドルに含める場合、そのリソースはその CICS バンドルによって使用されるモジュールをロードするためだけに使用してください。この要件は、スタンドアロン CICS バンドルでは強制されませんが推奨されています。アプリケーションの一部としてパッケージしてプラットフォームにインストールする CICS バンドルでは強制されます。

LIBRARY リソースに関連付けられた BUNDLEPART リソースは、LIBRARY 連結とそれからロードされたすべてのプログラムの使用回数がゼロにならないと、DISABLED 状況を報告しません。バンドル無効化プロセスを完了させるには、CICS から作業をパージしなければならない場合があります。

CICS バンドルによって定義されてインストールされた LIBRARY 連結からプログラムがロードされた場合、そのプログラムが独立して定義されたものであったとしても、そのプログラムは CICS バンドル内で定義されたもののように取り扱われます。これらのプログラムのライフサイクルは、LIBRARY リソースが定義された CICS バンドルのライフサイクルに準拠します。LIBRARY リソースを含んだ CICS バンドルを無効にすると、その LIBRARY 連結からロードされたすべてのプログラムが無効になります。

## PIPELINE リソース

CICS バンドルにパッケージされるパイプラインの場合、パイプライン構成ファイルはリソース定義と一緒に CICS バンドルにパッケージされます。パイプライン構成ファイルは、CICS 提供のサンプル・パイプライン構成ファイルの 1 つを使用して作成するか、ローカル・ファイル・システムから既存のパイプライン構成ファイルをインポートすることができます。

PIPELINE リソースが CICS バンドル内にパッケージされている場合、それは、CICS バンドルを使用して定義される WEBSERVICE リソースをホスティングするため、プラットフォームの一部としてデプロイする必要があります。CICS バンドル内で定義された PIPELINE リソースは、CICS バンドル内で定義されたか、パイプライン・スキャンによって動的に作成された WEBSERVICE リソースと一緒に使用する必要があります。CICS CSD または BAS を使用して定義された WEBSERVICE リソースは、CICS バンドル内で定義された PIPELINE リソースと互換性がありません。

CICS バンドルにパッケージされた PIPELINE リソースは、それを必要とする WEBSERVICE リソースをインストールする前にインストールして有効にする必要があります。良い習慣として、まず PIPELINE リソースの定義を含んだ CICS バンドルをプラットフォーム・バンドルの一部としてデプロイし、次に WEBSERVICE リソースを含んだ CICS バンドルを、プラットフォームにデプロイするアプリケーションの一部としてデプロイすることをお勧めします。このアーキテクチャーにより、リソースを CICS 領域に最初にインストールする際や、CICS 領域を再始動する際に、PIPELINE リソースと WEBSERVICE リソースを確実に正しい順序でインストールして有効にすることができます。

CICS バンドル内で定義されている PIPELINE リソースで WSDIR 属性を指定することは可能ですが、これはお勧めできません。SHELF 属性は、CICS バンドル内で定義されている PIPELINE リソースには使用されません。

## PROGRAM リソース

プログラムが使用で、そのインストール元の CICS バンドルが無効である場合、CICS はプログラムを無効にします。そのため、そのプログラムによって新規の作業を開始することはできません。ただし、関連付けられた BUNDLEPART リソースは、プログラムの使用回数がゼロになるまで、有効のままになります。その結果として、その期間のあいだは BUNDLEPART リソースとプログラムの両方から報告される状態が異なることになります。道理にかなった間隔を空けてもプログラムの使用回数がゼロにならない場合は、バンドル無効化プロセスを完了させるために CICS から作業をパージしなければならない場合があります。

バンドルの状態が DISABLING から DISABLED に変わったら、CICS バンドルを破棄できます。ただし、CICS バンドルを破棄すると、そのバンドルによってインストールされたプログラムは破棄され、CICS バンドルが再インストールされるまで使用不可になります。

CICS バンドルからインストールされたプログラムについては、PROGRAM NEWCOPY プロセスおよび PROGRAM PHASEIN プロセスを使用できません。CICS バンドルで定義されたプログラムについて実行できるプロセスは、新しいバージョンの CICS バンドルのインストール、またはその CICS バンドルでデプロイされたアプリケーションの新しいバージョンのインストールです。スタンドアロン CICS バンドル内で定義されている PROGRAM リソースの場合、新しいバージョンをインストールする前に、CICS バンドルの旧バージョンをアンインストールする必要があります。プラットフォーム上のアプリケーションの一部と

して CICS バンドルで定義する PROGRAM リソースをデプロイする際、PROGRAM リソースの前のバージョンをアンインストールせずに新規バージョンをインストールすることができ、ユーザーに新規バージョンを使用可能にするかどうかを管理できます。アプリケーションの複数バージョンの管理について詳しくは、[プラットフォームにデプロイされたアプリケーションの複数バージョン管理](#)を参照してください。

CICS バンドルによってインストールされたプログラムは、PLT からの呼び出しには適しません。

PROGRAM リソースはアプリケーション・エントリー・ポイントとして宣言できます。その PROGRAM リソースを CICS バンドル内でアプリケーション・エントリー・ポイントとして宣言するように定義する必要はありません。アプリケーション・エントリー・ポイントとして宣言されるプログラムには、使用環境内で固有の PROGRAM リソース名が必要です。アプリケーション・エントリー・ポイントとしてのプログラムの詳細については、[アプリケーション・エントリー・ポイント](#)を参照してください。

## TRANSACTION リソース

トランザクションには、EXEC CICS START コマンドによってスケジュールされた時間に開始される作業を割り当てることができます。その作業が CICS バンドルの無効化後に開始するようにスケジュールされている場合、スケジュールされている作業は取り消され、そのトランザクションで新しい作業を開始できません。CICS バンドルが再度有効になっても、取り消された作業のスケジュールは変更されず、その作業は取り消されたままになります。

TRANSACTION リソースはアプリケーション・エントリー・ポイントとして宣言できます。その TRANSACTION リソースを CICS バンドル内でアプリケーション・エントリー・ポイントとして宣言するように定義する必要はありません。アプリケーション・エントリー・ポイントとして宣言される TRANSACTION には、使用環境内で固有の TRANSACTION リソース名が必要です。アプリケーション・エントリー・ポイントとしての TRANSACTION について詳しくは、[アプリケーション・エントリー・ポイント](#)を参照してください。

## URIMAP リソース

WEBSERVICE リソースは、通常、少なくとも 1 つの URIMAP リソースに関連付けられます。URIMAP リソースを WEBSERVICE リソースと同じバンドルにパッケージすることもできますが、WEBSERVICE リソースのデプロイ先のプラットフォームごとにカスタマイズできるように別個のバンドルにパッケージするほうが良い場合があります。例えば、品質保証プラットフォームで使用する URI は実動プラットフォームとは異なるものを使用する場合があります。また、実動プラットフォームでは WEBSERVICE リソースのミラー・トランザクションを定義して、単体テスト・プラットフォームでは定義しない場合もあります。

CICS アプリケーション・バンドルの中で URIMAP リソースを定義する場合は、アプリケーション・エントリー・ポイント宣言を使用して、URIMAP リソースが提供するサービスへのユーザーのアクセスを制御することができます。この場合、アプリケーションをインストールして有効にした時点では、URIMAP リソースによって提供されるサービス呼び出し元が使用することはまだできません。サービスをユーザーに提供する場合は、CICS Explorer を使用して、CICS バンドルを含んだアプリケーションを使用可能にします。これにより、アプリケーション・エントリー・ポイントが呼び出し元に対して使用可能になるため、URIMAP リソースによって提供されるサービスが呼び出し元に対して使用可能になります。

アプリケーションの外側に定義された URIMAP リソースを、アプリケーション・エントリー・ポイントとして宣言することもできます。この場合、URIMAP リソースをインストールおよび使用可能にするとすぐに、ユーザーはサービスを使用できるようになります。

URIMAP リソースを使用して CICS から静的応答を提供することにより HTTP サーバーとして機能させている場合は、応答の本体を形成する z/OS UNIX System Services zFS ファイルの名前を HFSFILE 属性で指定します。CICS バンドルで定義される URIMAP リソースについては、zFS ファイルを URIMAP リソースと一緒に CICS バンドルにパッケージする必要があります。HFSFILE 属性には、CICS バンドルのルート・ディレクトリを基準とする相対ファイル・パスを指定してください。相対ファイル・パスの場合、パスの先頭にスラッシュを使用しません。詳しくは、297 ページの『[バンドル内の zFS 成果物の参照](#)』を参照してください。バンドルの外側の zFS ロケーションを絶対参照することも可能ですが、この方法は使用しないことを強くお勧めします。

## WEBSERVICE リソース

CICS バンドルにパッケージする Web サービスについては、Web サービス・バイnding・ファイルと、WSDL 文書または WSDL アーカイブ・ファイルをインポートして、リソース定義と一緒にバンドルにパッケージすることができます。Web サービスをサポートするには、CICS バンドルにパッケージされた WEBSERVICE 定義を使用して URIMAP 定義を別個のバンドルに生成できます。また、その URI マップに対する別名トランザクションと、WSDL ディスカバリー用のオプションの URIMAP 定義を作成することもできます。

CICS バンドルにパッケージされた Web サービスには、DISABLING および DISABLED という追加の状態がありますが、これは他の方法を使用して作成された Web サービスには当てはまりません。CICS バンドルの無効化が進行中のとき、バンドル内で定義された WEBSERVICE リソースは DISABLING 状態に入ります。現在実行中の作業は完了できますが、Web サービスは新規の作業を受け入れません。Web サービスが使用中でなくなった時点で、WEBSERVICE リソースは DISABLED 状態に入ります。DISABLING または DISABLED 状態の Web サービスへの要求は拒否され、SOAP 障害「Web service is not in service」が送信されます。CICS が Web サービス・リクエスターである場合は、INVOKE SERVICE コマンドが、INVREQ の RESP コードと 8 の RESP2 値を戻します。

CICS バンドルが再び有効になると、WEBSERVICE リソースは INSERVICE 状態に戻ります。それ以外の場合、CICS バンドルを破棄することにより、WEBSERVICE リソースを破棄することができます。WEBSERVICE リソースの状態を照会するには、EXEC CICS または CEMT **INQUIRE WEBSERVICE** コマンド、CICSplex SM Web ユーザー・インターフェース、または CICS Explorer を使用できます。ただし、状態を手動で設定することはできません。

### 関連概念

#### バンドルにデプロイ可能な成果物

CICS バンドルに定義してデプロイできる成果物には、アプリケーションとシステムのイベント、Atom フィード、チャンネル・ベースのサービス、CICS ポリシー、CICS プログラム、OSGi バンドル、XML ベースのサービスおよびトランザクションが含まれます。これらの成果物はそれぞれ 1 つ以上の CICS リソースによって表され、それらのリソースはバンドル・デプロイメントの一部として動的に作成されます。

#### バンドル内の zFS 成果物の参照

いくつかの CICS リソースが、外部 zFS 成果物を参照して詳しい構成情報を探します。例えば、JVMSERVER リソースでは JVM プロファイルが必要となり、PIPELINE リソースではパイプライン構成ファイルが必要となります。これらのリソースが CICS バンドルに定義されている場合、必要となる zFS 成果物も CICS バンドルに保管され、相対 zFS 名を使用して参照される必要があります。

#### CICS バンドルのマニフェストの内容

それぞれの CICS バンドルには、バンドルの内容を記述するバンドル・マニフェストが含まれています。バンドル・マニフェストには、バンドルと、プロビジョン時に CICS 領域で設計または変更するリソース、さらに CICS バンドルを正常に使用可能にするために必要な依存関係が記述されます。

#### バンドルのスコーピング

BUNDLE リソース定義には、関連する BUNDLE リソースに有効範囲を適用する方法として、BASESCOPE 属性があります。この属性は、プラットフォームにデプロイされているバンドルに使用したり、サービス・コンポーネント・アーキテクチャー (SCA) 複合アプリケーションが含まれているバンドルの SCA ドメインの設定に使用したりできます。

#### CICS リスタート後の OSGi バンドル・リカバリー

OSGi バンドルを含む CICS 領域をリスタートすると、CICS は BUNDLE リソースをリカバリーし、OSGi バンドルを JVM サーバーのフレームワークにインストールします。

#### バンドルのセキュリティ

定義から作成された BUNDLE リソースと、アプリケーションまたはプラットフォームをインストールするときに作成した BUNDLE リソースでは、適用されるセキュリティ・プロファイルとセキュリティ検査が異なります。

#### CICS プロジェクトでの変数

変数を使用することで属性値を素早く簡単に変更することができます。各環境に固有のプロパティ・ファイルを使用して変数を解決できるので、複数環境へのデプロイメントが単純化されます。

### 関連資料

#### 変数およびプロパティ・ファイルの定義

変数およびそれらに関連付けられたプロパティ・ファイルを定義する場合、変数の置換が正常に行われるように、これらのルールに従います。

## 関連情報

### アプリケーション・バージョンの専用リソース

特定の CICS リソースを、プラットフォームにインストールするアプリケーションの一部として CICS バンドルで定義した場合、そのリソースは、そのアプリケーションのそのバージョン専用のものになります。そのため、これらのタイプのリソースは、同時に複数を同じ名前で同じプラットフォーム・インスタンス上にインストールできます。

## バンドル内の zFS 成果物の参照

いくつかの CICS リソースが、外部 zFS 成果物を参照して詳しい構成情報を探します。例えば、JVMSERVER リソースでは JVM プロファイルが必要となり、PIPELINE リソースではパイプライン構成ファイルが必要となります。これらのリソースが CICS バンドルに定義されている場合、必要となる zFS 成果物も CICS バンドルに保管され、相対 zFS 名を使用して参照される必要があります。

CICS バンドルに zFS 構成ファイルを保管すると、リソース定義とその構成ファイルがアトミック・ユニットとしてデプロイされます。デプロイメントの後に構成ファイルを更新する場合、CICS Explorer の CICS バンドル・プロジェクトで変更を加え、そのバンドルを再デプロイする必要があります。ほとんどの場合、CICS Explorer には、構成ファイルを扱うための多機能エディターが含まれています。

リソース定義を含んだ CICS バンドルがインストールされると、CICS は、CICS バンドルのルート・ディレクトリーを基準とした相対位置で zFS 構成ファイルへの参照を解決します。構成ファイルは、CICS バンドル・プロジェクト内のルート・ディレクトリーまたはサブディレクトリーに保管できます。

JVM サーバーにデプロイされたアプリケーションの場合、**&CONFIGROOT;** シンボルを、JVM プロファイル内で使用できます。これは、実行時に CICS バンドルのルート・ディレクトリーに置換されます。アプリケーションは **com.ibm.cics.jvmserver.configroot** システム・プロパティの値を使用して、JVM サーバーの構成ファイルを見つけることができます。

## 関連概念

### バンドルにデプロイ可能な成果物

CICS バンドルに定義してデプロイできる成果物には、アプリケーションとシステムのイベント、Atom フィード、チャネル・ベースのサービス、CICS ポリシー、CICS プログラム、OSGi バンドル、XML ベースのサービスおよびトランザクションが含まれます。これらの成果物はそれぞれ 1 つ以上の CICS リソースによって表され、それらのリソースはバンドル・デプロイメントの一部として動的に作成されます。

### CICS バンドルのマニフェストの内容

それぞれの CICS バンドルには、バンドルの内容を記述するバンドル・マニフェストが含まれています。バンドル・マニフェストには、バンドルと、プロビジョン時に CICS 領域で設計または変更するリソース、さらに CICS バンドルを正常に使用可能にするために必要な依存関係が記述されます。

### バンドルのスコーピング

BUNDLE リソース定義には、関連する BUNDLE リソースに有効範囲を適用する方法として、BASESCOPE 属性があります。この属性は、プラットフォームにデプロイされているバンドルに使用したり、サービス・コンポーネント・アーキテクチャー (SCA) 複合アプリケーションが含まれているバンドルの SCA ドメインの設定に使用したりできます。

### CICS リスタート後の OSGi バンドル・リカバリー

OSGi バンドルを含む CICS 領域をリスタートすると、CICS は BUNDLE リソースをリカバリーし、OSGi バンドルを JVM サーバーのフレームワークにインストールします。

### バンドルのセキュリティ

定義から作成された BUNDLE リソースと、アプリケーションまたはプラットフォームをインストールするときに作成した BUNDLE リソースでは、適用されるセキュリティ・プロファイルとセキュリティ検査が異なります。

### CICS プロジェクトでの変数

変数を使用することで属性値を素早く簡単に変更することができます。各環境に固有のプロパティ・ファイルを使用して変数を解決できるので、複数環境へのデプロイメントが単純化されます。

## 関連資料

### 変数およびプロパティ・ファイルの定義



変数およびそれらに関連付けられたプロパティ・ファイルを定義する場合、変数の置換が正常に行われるように、これらのルールに従います。

## 関連情報

### CICS バンドル内リソースの特性

CICS リソースは、CICS バンドルで定義され、バンドル・デプロイメントの一部として動的に作成されるため、その一部の特性が変更されます。アプリケーションのアーキテクチャーを設計する、あるいは CICS バンドルで定義する既存のアプリケーションからリソースを選択する場合、これらの重要な考慮事項に注意してください。

### アプリケーション・バージョンの専用リソース

特定の CICS リソースを、プラットフォームにインストールするアプリケーションの一部として CICS バンドルで定義した場合、そのリソースは、そのアプリケーションのそのバージョン専用のものになります。そのため、これらのタイプのリソースは、同時に複数を同じ名前で同じプラットフォーム・インスタンス上にインストールできます。

## アプリケーション・バージョンの専用リソース

特定の CICS リソースを、プラットフォームにインストールするアプリケーションの一部として CICS バンドルで定義した場合、そのリソースは、そのアプリケーションのそのバージョン専用のものになります。そのため、これらのタイプのリソースは、同時に複数を同じ名前で同じプラットフォーム・インスタンス上にインストールできます。

サポートされるリソース・タイプにおいては、CICS リソースが、アプリケーションの一部として (つまり、アプリケーション・バンドルの一部として、またはアプリケーション・バインディングの一部として) パッケージされてインストールされる CICS バンドルで定義される場合、専用リソースになります。この方法で CICS リソースを作成した場合、そのリソースは、プラットフォームにインストールされている他のアプリケーションやバージョンからは使用できず、CICS 領域内の他のアプリケーションからも使用できません。そのリソースが定義されたバージョンのアプリケーションからのみ使用できます。このようなリソースは、専用リソースと呼ばれます。

アプリケーションの専用リソースとしてサポートされる CICS リソースは次のとおりです。

- **LIBRARY** リソース。これは、動的プログラム LIBRARY 連結と呼ばれる 1 つ以上のデータ・セットを表しており、このデータ・セットからプログラムのロード・モジュールをロードできます。
- **PACKAGESET** リソース。これは、Db2 コレクションを表し、非修飾 EXEC SQL 要求が参照する Db2 データベース内の表を修飾するために使用されます。
- **POLICY** リソース。これは、CICS 領域のユーザー・タスクの動作を管理する 1 つ以上の規則を表しています。
- **PROGRAM** リソース。アプリケーション・プログラムを表します。プラットフォームにデプロイされるアプリケーションのためにタスクによって自動インストールされるプログラムは、そのバージョンのアプリケーションに専用のものでもあります。

アプリケーションの一部として定義されるその他のリソース・タイプの CICS リソースと、他のいずれかの方法で定義される CICS リソースは、公用になり、すべてのタスクから使用可能になります。このようなリソースは、公用リソースと呼ばれます。1 つのバージョンしか存在しないアプリケーションの場合、アプリケーション・エントリー・ポイントとして宣言された専用リソースは、アプリケーション・エントリー・ポイントが使用可能になった時点で公用リソースになります。複数のバージョンがあるアプリケーションの場合、そのアプリケーションが使用可能な最も高いバージョンであるなら、アプリケーション・エントリー・ポイントとして宣言されているプログラムが公用になります。同じアプリケーションのそれ以外のバージョンについて、アプリケーション・エントリー・ポイントとして宣言されているプログラムは、専用になります。

サポートされているリソース・タイプのリソースを専用にしない場合は、リソース定義をアプリケーションの一部としてパッケージしないでください。代わりに、CICS リソースを定義するために、スタンドアロンの CICS バンドル、プラットフォームのレベルでインストールされる CICS バンドル、CICS CSD、または CICSplex SM データ・リポジトリを使用します。CICS 領域内の CICS リソースが使用可能になることがアプリケーションによって求められる場合、バンドル・マニフェストの <import> エレメントの中に、そのリソースをアプリケーションまたはアプリケーション・バインディングと依存関係にあるものとして追加します。

POLICY および PACKAGESET リソースは CICS バンドル内でのみ定義でき、アプリケーションの一部としてデプロイされる CICS バンドルの中で定義すれば、専用リソースと同様の機能がサポートされます。したがって、異なるアプリケーションおよび異なるアプリケーション・バージョンの中で同じ名前の POLICY および PACKAGESET リソースを使用できます。

専用リソースは、特定のアプリケーションに関連しているだけでなく、特定のバージョンのアプリケーションにも関連しています。したがって、複数のバージョンのアプリケーションを同じプラットフォーム・インスタンス上に同時にインストールできます。アプリケーションの複数バージョンの管理について詳しくは、[プラットフォームにデプロイされたアプリケーションの複数バージョン管理](#)を参照してください。

## アプリケーションの専用リソースの管理

専用リソースが、アプリケーションの一部としてパッケージされてインストールされる CICS バンドルで定義される場合、インストール済み環境内でリソース名を固有にする必要はありません。この機能を使用すると、別々に開発されたアプリケーション間で同じリソース名を使用しているためにリソース名が競合することを回避できます。サポートされる CICS リソースについてリソース名を固有にするという要件は、プラットフォームにデプロイされるアプリケーションの一部としてリソースを管理することにより除去されます。このプロセスは、サーバーの統合を支援するために使用できます。

アプリケーションを変更する場合は、CICS バンドルにパッケージされたアプリケーションの関連エレメントを CICS Explorer を使用して変更し、変更を示すためにこれらの CICS バンドルに新しいバージョン番号を適用した後、アプリケーションのリバージョンと再インストールを行います。同じ ID および同じバージョン番号で CICS バンドルを複数インストールするプロセスは CICS が管理するので、未変更の CICS バンドルは前と同じバージョン番号のままにしてください。

未変更の CICS バンドルの再インストールを使用できるのは、プラットフォームにデプロイされるアプリケーションの一部としてインストールされる CICS バンドルについてのみです。スタンドアロンの CICS バンドルは、既に同じ ID およびバージョンでインストールされている場合、または CICS TS バージョン 5.1 より前のリリースで作成された CICS バンドルでは ID およびバージョンを指定せずにインストールされた場合に、再インストールできません。ただし、同じ CICS バンドルを CICS 領域にスタンドアロン CICS バンドルとしてインストールすることは可能で、プラットフォームにデプロイされる 1 つ以上のアプリケーションの一部としてインストールまたは再インストールすることも可能です。

専用リソースに対してインストールや破棄などの操作が実行されると、特別な CICS メッセージが出されます。そのメッセージでは、そのタイプの公用リソースに対して対応する操作が実行されたときと同じ情報が提供されますが、それに加えて、専用リソースの適用先のプラットフォーム、アプリケーション、およびアプリケーション・バージョンについても記載されます。そのため、関係するコンテキストでそれらの操作の監査やトラブルシューティングを実行できます。

インストールされたアプリケーションの各バージョンについて専用リソースを表示するには、CICS Explorer を使用します。アプリケーション記述子エディターでは、アプリケーションの専用リソースとアプリケーション・エントリ・ポイントをリソース・タイプ別に表示できます。また、それらの情報を CICS 領域または CICS バンドルでフィルタリングして特定のリソースを見つけることもできます。さらに、LIBRARY 連結に対して z/OS が生成したデータ・セットの DD 名を専用 LIBRARY リソースごとに表示することもできます。

専用リソースを照会またはブラウズするために、リソース・タイプに対する **EXEC CICS INQUIRE** システム・プログラミング・コマンドを使用できます。デフォルトでは、CICS は、**EXEC CICS INQUIRE** コマンドが発行されたプログラムで使用可能な専用リソースを検索します。指定したアプリケーションの専用リソースをブラウズすることも選択できます。

- 公用プログラムから **EXEC CICS INQUIRE** コマンドを発行すると、指定した公用リソースに関する情報が返されます。そのリソースが公用リソースとしては使用できない場合、「見つかりません (not found)」という応答が返されます。
- プラットフォーム上にデプロイされているアプリケーションのためのタスク下で実行されているプログラムから **EXEC CICS INQUIRE** コマンドを発行すると、そのアプリケーションの、指定した専用リソース (存在する場合) に関する情報が返されます。そのアプリケーションにその名前の専用リソースが存在しない場合には、指定された名前の公用リソースに関する情報が返されます。そのリソースが、そのアプリケーションの専用リソースとしても、公用リソースとしても使用できない場合、「見つかりません (not found)」という応答が返されます。



- **EXEC CICS INQUIRE** コマンドを公用プログラムからブラウズ・モードで使用した場合、他に入力パラメーターを指定しないと、指定したタイプの公用リソースのセットが返されます。同じブラウズ・コマンドを、プラットフォーム上にデプロイされているアプリケーションのためのタスク下で実行されているプログラムから発行した場合は、そのアプリケーションの指定したタイプの専用リソースすべて、および指定したタイプの公用リソースすべてで構成されるリソースのセットが返されます。
- 公用プログラムまたは専用プログラムのいずれかからアプリケーションの専用リソースをブラウズするには、**START** オプションを付けて **EXEC CICS INQUIRE** コマンドを発行し、プラットフォーム、アプリケーション、およびアプリケーション・バージョンで構成されるアプリケーション・コンテキストを入力として指定します。このブラウズ・コマンドからは、アプリケーションの指定したタイプの専用リソースのみで構成されるリソースのセットが返されます。指定したアプリケーション・コンテキストを持つアプリケーションが見つからない場合は、**APPNOTFOUND** 状態が返されます。

コマンドを発行したアプリケーションから別のアプリケーションのリソースをブラウズする例など、専用リソースのブラウズについて詳しくは、[リソース定義のブラウズ](#)を参照してください。

CICSplex SM リアルタイム分析 (RTA) 機能で専用リソースをサポートするには、そのリソースの評価定義 (EVALDEF) でアプリケーション・コンテキスト・パラメーター **PLATFORM**、**APPLICATION**、**APPLMAJORVER**、**APPLMINORVER**、および **APPLMICROVER** を指定する必要があります。ユーザー定義の CICSplex SM WUI ビューとリソース・オブジェクトを使用してリアルタイム分析定義を作成および保守する手順については、[リアルタイム分析](#)を参照してください。

CICS は、専用リソースに対して別個の統計レコードを作成します。専用リソースのための統計レコードには、そのリソースが定義された先のアプリケーションに関する情報が含まれます。公用プログラムおよびライブラリー・リソースに関する統計レポート **DSECT** および **DFHSTUP** には、専用リソースに関する対応する **DSECT** および **DFHSTUP** レポートがあります。アプリケーション・エントリー・ポイントとして宣言されたプログラムは、公用統計と専用統計の両方で識別されて報告されます。これは、エントリー・ポイントは公用としてアクセス可能であると同時に、アプリケーションの一部でもあるからです。

## 従属の専用リソース

プラットフォームにデプロイされるアプリケーションの一部としてパッケージする CICS バンドルの中には、公用リソースまたは専用リソース (アプリケーションによって自動インストールされる専用プログラムを含む) を従属として定義できます。CICS は、まず、指定されたタイプおよび名前の専用リソースが、バンドル・マニフェスト・ファイルに **<import>** エレメントが含まれている CICS バンドルと同じアプリケーションに含まれているかどうかを検査します。複数のバージョンのアプリケーションがインストールされている場合は、現在のアプリケーション・コンテキストにより、インポートされる専用リソースのバージョンが決まります。そのリソースがアプリケーションの専用リソースとして見つからない場合は、CICS は CICS 領域にある公用リソースの中に指定されたタイプおよび名前に一致するリソースがあるかどうか検査し、そのリソースをインポートします。他のアプリケーションから専用リソースをインポートすることはできません。

## 専用 LIBRARY リソース

LIBRARY リソースは、動的プログラム LIBRARY 連結と呼ばれる 1 つ以上のデータ・セットを表しており、このデータ・セットからプログラムのロード・モジュールをロードできます。LIBRARY リソースは、アプリケーション・バージョンの専用リソースとしてサポートされます。アプリケーションの各バージョンには、そのアプリケーションのロード・モジュールの入ったバージョン固有のデータ・セットを表す専用 LIBRARY リソースを少なくとも 1 つ組み込む必要があります。

- プラットフォーム上にデプロイされているアプリケーションのためのタスクでプログラムのロードが必要な場合、正しいバージョンのプログラムがロードされるようにするため、まず、そのバージョンのアプリケーションに対して定義されている専用 LIBRARY 連結が検索されます。同じアプリケーション・バージョンに対して複数の LIBRARY 連結が定義されている場合は、ランキングの順序に従って検索されます。
- アプリケーション内に定義された LIBRARY 連結がない場合、またはどの専用 LIBRARY 連結にもプログラムが見つからない場合は、CICS 領域全体で定義されている公用 LIBRARY 連結 (**DFHRPL** を含む) が検索の対象になります。
- プラットフォーム上にデプロイされているアプリケーションに関連しないタスクの場合は、公用 LIBRARY 連結のみが検索の対象になるため、専用 LIBRARY 連結の中にあるプログラム・ロード・モジュールはすべて使用できません。

1つのアプリケーションの専用 LIBRARY 連結のいずれかにおいて、関連する LIBRARY リソースを定義している CICS バンドルが無効になったために無効になった場合、CICS は他の専用 LIBRARY 連結を検索せず、CICS 領域全体に対して定義された公用 LIBRARY 連結も検索しません。このため、そのアプリケーションによる後続のプログラム・ロードは、その LIBRARY リソースを定義する CICS バンドルが有効になるまで、すべて失敗します。

アプリケーション・バンドルまたはアプリケーション・バインディング・バンドルの一部としてパッケージされてインストールされる CICS バンドルの中で定義されている専用 LIBRARY リソースの場合、LIBRARY リソースの名前は、データ・セットの LIBRARY 連結に対する DD 名としては使用されません。その代わりに、CICS は、アプリケーションがプラットフォームにインストールされる時点で、データ・セットの LIBRARY 連結に対する固有の DD 名を要求します。したがって、リソース名は、インストール済み環境内のどこかで使用されている LIBRARY 名と同じであっても、または異なるバージョンのアプリケーションで使用されている名前と同じであってもかまいません。CICS は、LIBRARY 連結に対して z/OS が生成した DD 名を通知するメッセージ DFHLD0518 を発行します。インストールされたアプリケーションのデータ・セット名は、CICS Explorer で表示することもできます。

## 専用 PACKAGESET リソース

専用リソース PACKAGESET は Db2 コレクションを表します。PACKAGESET リソースには name という 1つの属性があり、これは Db2 コレクション識別子の名前で、長さは最大 128 バイトです。アプリケーションの代わりに CICS で EXEC SQL SET CURRENT PACKAGESET コマンドを発行できるようにすることで、PACKAGESET を使用してさまざまな環境にわたるさまざまなコレクション識別子を指定します。このリソースは、クラウド環境でのデータの処理をより簡単かつ柔軟にします。

アプリケーションには、最大 2 つの PACKAGESET リソースを関連付けることができます。複数の PACKAGESET リソースがインストールされている場合、関連付けられるリソースは、以下の順序で選択されます。

1. アプリケーションまたはアプリケーション・バインディングの一部としてインストールされている PACKAGESET リソース。PACKAGESET リソースをアプリケーション・バインディングの一部としてインストールすると、アプリケーション・バンドルを変更せずに、さまざまな環境でさまざまなコレクション識別子を使用できます。例えば、テスト、品質保証、および実稼働環境は、それぞれアプリケーションを異なる PACKAGESET リソース、したがって異なるコレクション識別子に関連付けることができます。PACKAGESET リソースをアプリケーション・バインディングではなくアプリケーションの一部としてインストールすると、すべての環境で同じ PACKAGESET リソース、したがって同じコレクション識別子が使用されます。
2. プラットフォームの一部としてインストールされている PACKAGESET リソース。この PACKAGESET リソースを関連付けると、プラットフォームで実行されているすべてのアプリケーションによって同じ PACKAGESET リソース、したがって同じコレクション識別子が使用されます。

CICS-Db2 接続機能は、作業単位の最初の SQL 要求を処理するときに、タスクに適用される PACKAGESET リソースが存在するかどうかを検査します。まず、タスクのアプリケーション・コンテキストで有効範囲がプラットフォーム、アプリケーション、およびバージョンと一致する PACKAGESET リソースが定義されているかどうかを検査します。定義されていない場合、この機能はプラットフォーム上のすべてのアプリケーションについて、PACKAGESET リソースが定義されているかどうかを検査します。PACKAGESET リソースが存在する場合、CICS-Db2 接続機能は CICS 提供プログラム DFHD2SPS にリンクして、アプリケーションからの最初の SQL 要求を処理する前に **EXEC SQL SET CURRENT PACKAGESET** コマンドを発行します。

後続の SQL 要求については、アプリケーション・コンテキストが変更されない限り、追加処理は必要ありません。アプリケーション・コンテキストが変更された場合、CICS-Db2 接続機能は DFHD2SPS にリンクして、新しいパッケージ・セット名を設定するか、新しいコンテキストに PACKAGESET リソースが定義されていない場合はブランクのパッケージ・セット名を設定します。アプリケーション・コンテキストが変更される可能性がある場合の例として、新しいアプリケーションに **EXEC CICS LINK** コマンドが発行された場合、**EXEC CICS LINK** 呼び出しが呼び出し元のアプリケーションに戻る場合があります。

同期点では、CICS-Db2 接続機能は DFHD2SPS にリンクして PACKAGESET リソースをブランクにリセットします。

DFHD2SPS は、L8 オープン TCB 上で CONCURRENCY(REQUIRED)、API(CICSAPI)、EXECKEY(CICS) プログラムとして実行されます。通常は L8 TCB 上で実行されている CICS-Db2 TRUE DFHD2EX1 からリンク

されるため、TCB 切り替えのオーバーヘッドは発生しません。ただし、Java プログラムと XPLINK C プログラムでは、DFHD2EX1 はそれぞれ T8 または X8 TCB 上で実行されるため、L8 TCB への TCB 切り替えが実行され、DFHD2SPS の起動時に再び戻ります。

PACKAGESET リソースはオプションであり、さまざまな環境にわたってさまざまなコレクションを管理するための既存のメカニズム (複数の計画、動的計画出口、またはアプリケーションでのパッケージ・セットの設定など) は引き続き使用可能です。



**重要: SET CURRENT PACKAGESET** コマンドを使用する既存のプログラムは、PACKAGESET リソースを使用するアプリケーションを呼び出した後、パッケージ・セットを再確立する必要があります。パッケージ・セットを再確立するには、別の **SET CURRENT PACKAGESET** コマンドを発行します。CICS は、PACKAGESET リソースで定義されていないパッケージ・セットを認識できません。

バンドル定義の PACKAGESET リソースは、通常はアプリケーション・バインディングを介して、アプリケーションの一部としてのみ使用します。アプリケーションまたはプラットフォームの一部ではない PACKAGESET リソースが含まれているスタンドアロン・バンドルをインストールしようとすると失敗します。

## 専用 POLICY リソース

POLICY リソースは、CICS 領域でのユーザー・タスクの動作を管理する 1 つ以上のルールを表します。ポリシーは、しきい値およびアクションが関連付けられた 1 つ以上の規則タイプが含まれる XML 定義です。ポリシー・ルールは、1 つ以上のタスクに適用できる制御やアクションを記述します。条件とアクションのペアにより、ポリシー・ルールが構成され、ポリシー内に 1 つ以上のポリシー・ルールを定義できます。ポリシーは CICS バンドル内で定義され、CICS バンドルは 1 つ以上のポリシーで構成できます。

実行中のタスクがリソース使用量に関する定義済みのしきい値を超過したときに、CICS は、ポリシーで指定されたアクションを実行します。ポリシーをデプロイすると、ユーザー・タスクによるリソース使用状況をモニターし、リソース使用量が定義したしきい値を超えた場合に自動的に応答することができます。

**EXEC CICS** コマンドを発行してタスクが制御を CICS に解放している限り、リソースの過剰使用状況や、ループ・トランザクション、ランナウェイ・トランザクションを検出し、適切に対処できます。

ポリシーは、常に特定の有効範囲にデプロイされる CICS バンドルで常に定義されます。有効範囲としては、アプリケーション、アプリケーション内の操作、またはプラットフォームが可能です。

ポリシーでは、アプリケーションまたはシステムの側面の動作に関する契約を定義します。ポリシーはユーザー・タスクのインスタンスに適用され、タスクのさまざまなグループで共通のポリシー・セットを共用できるため、ワークロードのセクションをさまざまな方法で制御できます。アプリケーションまたはプラットフォームで実行されているタスクに関連付けられたポリシー・ルールのコレクションによって、タスクを安全に実行できる境界が定義されます。これにより、システム管理者はユーザー・アプリケーションの動作に対して制限を事前に設定できます。このように、ポリシーを使用すると、適用されるポリシーに定義されているルールに違反するタスクの動作を CICS で実行時に測定、対応、および強制できます。

ポリシーについて詳しくは、[CICS ポリシー](#)を参照してください。

## 専用 PROGRAM リソース

PROGRAM リソースは、プログラム・ライブラリーに格納されたプログラムのロード・モジュールを表します。PROGRAM リソースは、アプリケーション・バージョンの専用リソースとしてサポートされます。プラットフォームにデプロイされるアプリケーションのためにタスクによって自動インストールされるプログラムは、そのバージョンのアプリケーションに専用のものでもあります。

各プログラムの各バージョンの 1 つのコピーのみが CICS ストレージにロードされます。CICS は、専用プログラムをロードする前に、PROGRAM リソース定義が一致する同じデータ・セット (PDS または PDSE) からそのバージョンのプログラムが既にロードされているかどうか検査します。そうであれば、CICS は既存のコピーを使用します。したがって、PROGRAM リソース名の再利用に関して、次のような規則が適用されます。

- 同じ PDS または PDSE からロードする同じプログラムを共用する予定のある複数のアプリケーションは、PROGRAM リソース定義で同じ属性を使用する必要があります。
- 複数のアプリケーションが異なるプログラム・リソースに対して同じ名前を使用する場合は、各アプリケーションは異なるデータ・セット (PDS または PDSE) からプログラムをロードする必要があります。

専用プログラムの PROGRAM リソース定義の中で **RELOAD=YES** を指定すると、プログラムのロードの動作は公用プログラムと同じ動作をするように変更されます。プログラム制御のリンク、ロード、または XCTL 要求により、プログラムの新しいコピーがストレージに取り込まれます。RELOAD(YES) のプログラムは再利用できず、複数のアプリケーションで共用することもできません。プログラムの各コピーが不要になった場合は、タスクが終了する前に、ストレージ管理 FREEMAIN 要求を使用して各コピーをストレージから明示的に除去する必要があります。

アプリケーションが必要とするプログラムがそのアプリケーションの専用プログラム・ディレクトリー内に見つからない場合、CICS は公用プログラム・ディレクトリーを検索します。

プラットフォームにデプロイされているアプリケーションのためのタスク下で実行されているプログラムから **EXEC CICS LINK**、**XCTL**、**LOAD**、および **RELEASE** コマンドが発行された場合、CICS は、指定されたプログラムをまずそのアプリケーションの専用プログラム・ディレクトリーから検索します。指定されたプログラムがそこで見つからない場合、CICS は公用プログラム・ディレクトリーを検索します。

アプリケーション・エントリー・ポイントとして宣言されるプログラムには、使用環境内で固有の PROGRAM リソース名が必要です。これらのプログラムをアプリケーション外から呼び出せるようにするためには、それらが公用リソースでなければなりません。専用 PROGRAM リソースに対するアプリケーション・エントリー・ポイントを含んだアプリケーションを使用可能にすると、アプリケーション・エントリー・ポイントとして指定されている PROGRAM リソースが専用リソースから公用リソースに変更されます。特定の名称で CICS 領域内に存在できる公用リソースは 1 つのインスタンスのみです。そのため、PROGRAM リソースは、CICS 領域にインストールされている公用プログラムと同じ名前や、インストールされている別のアプリケーションがアプリケーション・エントリー・ポイントとして定義している公用プログラムと同じ名前にすることができません。ただし、アプリケーション・エントリー・ポイントとして定義された同じ PROGRAM リソースの複数のバージョンは、同じアプリケーションの複数のバージョンに対してインストールできます。これは、アプリケーションの各バージョンにおける、PROGRAM リソースの公用状態へのプロモーションを CICS が管理しているためです。

## 関連概念

### バンドルにデプロイ可能な成果物

CICS バンドルに定義してデプロイできる成果物には、アプリケーションとシステムのイベント、Atom フィード、チャンネル・ベースのサービス、CICS ポリシー、CICS プログラム、OSGi バンドル、XML ベースのサービスおよびトランザクションが含まれます。これらの成果物はそれぞれ 1 つ以上の CICS リソースによって表され、それらのリソースはバンドル・デプロイメントの一部として動的に作成されます。

### バンドル内の zFS 成果物の参照

いくつかの CICS リソースが、外部 zFS 成果物を参照して詳しい構成情報を探します。例えば、JVMSEVER リソースでは JVM プロファイルが必要となり、PIPELINE リソースではパイプライン構成ファイルが必要となります。これらのリソースが CICS バンドルに定義されている場合、必要となる zFS 成果物も CICS バンドルに保管され、相対 zFS 名を使用して参照される必要があります。

### CICS バンドルのマニフェストの内容

それぞれの CICS バンドルには、バンドルの内容を記述するバンドル・マニフェストが含まれています。バンドル・マニフェストには、バンドルと、プロビジョン時に CICS 領域で設計または変更するリソース、さらに CICS バンドルを正常に使用可能にするために必要な依存関係が記述されます。

### バンドルのスコーピング

BUNDLE リソース定義には、関連する BUNDLE リソースに有効範囲を適用する方法として、BASESCOPE 属性があります。この属性は、プラットフォームにデプロイされているバンドルに使用したり、サービス・コンポーネント・アーキテクチャー (SCA) 複合アプリケーションが含まれているバンドルの SCA ドメインの設定に使用したりできます。

### CICS リスタート後の OSGi バンドル・リカバリー

OSGi バンドルを含む CICS 領域をリスタートすると、CICS は BUNDLE リソースをリカバリーし、OSGi バンドルを JVM サーバーのフレームワークにインストールします。

### バンドルのセキュリティ

定義から作成された BUNDLE リソースと、アプリケーションまたはプラットフォームをインストールするときに作成した BUNDLE リソースでは、適用されるセキュリティ・プロファイルとセキュリティ検査が異なります。

### CICS プロジェクトでの変数



変数を使用することで属性値を素早く簡単に変更することができます。各環境に固有のプロパティ・ファイルを使用して変数を解決できるので、複数環境へのデプロイメントが単純化されます。

## 関連資料

### 変数およびプロパティ・ファイルの定義

変数およびそれらに関連付けられたプロパティ・ファイルを定義する場合、変数の置換が正常に行われるように、これらのルールに従います。

## 関連情報

### CICS バンドル内リソースの特性

CICS リソースは、CICS バンドルで定義され、バンドル・デプロイメントの一部として動的に作成されるため、その一部の特性が変更されます。アプリケーションのアーキテクチャーを設計する、あるいは CICS バンドルで定義する既存のアプリケーションからリソースを選択する場合、これらの重要な考慮事項に注意してください。

## CICS バンドルのマニフェストの内容

それぞれの CICS バンドルには、バンドルの内容を記述するバンドル・マニフェストが含まれています。バンドル・マニフェストには、バンドルと、プロビジョン時に CICS 領域で設計または変更するリソース、さらに CICS バンドルを正常に使用可能にするために必要な依存関係が記述されます。

CICS バンドルのマニフェストは `cics.xml` と呼ばれ、META-INF サブディレクトリーにあります。バンドル・マニフェストは XML で記述され、スキーマに準拠します。詳しくは、[バンダー・インターフェース](#)を参照してください。バンドル・マニフェストは UTF-8 でエンコードされ、バンドルそのものに関する情報に加え、バンドルパーツ (バンドルによって提供される定義、エクスポート、インポート、アプリケーション・エントリー・ポイント、およびポリシーの有効範囲) に関する情報が含まれます。

### バンドル情報

マニフェストには、バンドルの作成時のタイム・スタンプなど、バンドルに関する情報を格納する `<meta-directives>` エレメントがオプションで含まれています。

マニフェストには、CICS バンドルのバージョン情報に関する詳細も含まれています。

### 定義

`<define>` エレメントは、CICS バンドルによって管理される CICS リソースを定義します。各リソース定義には以下のものが含まれます。

- 成果物の名前。
- URI として定義されるリソース・タイプ。
- バンドル内のファイルの相対パス位置。このファイルにはリソースのメタデータが格納されます。

いくつかの定義では、CICS が 1 つ以上のリソースを動的に作成することになります。例えば、`http://www.ibm.com/xmlns/prod/cics/bundle/EVENTBINDING` リソース・タイプは、EVENTBINDING リソースと 1 つ以上の CAPTURESPEC リソースを作成します。その他の定義 (`http://www.ibm.com/xmlns/prod/cics/bundle/POLICY` リソース・タイプなど) は、ランタイム処理でのみ CICS によって使用され、同等の CICS リソース定義はありません。

CICS は、CICS で提供されるリソースと一緒に、非 CICS リソースを管理するユーザーのための登録およびコールバック用インターフェースを提供しています。詳しくは、[コールバック・インターフェースで使用されるコンテナ](#)および[バンドルにデプロイ可能な成果物](#)を参照してください。

### エクスポート

`<export>` エレメントは、バンドルが提供可能なリソースまたはサービスに関する追加情報を提供します。CICS は、バンドルの処理にエクスポート・ステートメントを使用しません。指定されたエクスポートは、プロビジョンされる CICS バンドル内で、SPI、CMCI、および CICS Explorer などのツールを使用して表示できます。

### インポート

`<import>` エレメントは、バンドルが必要とする他のリソースへの依存関係を定義します。それぞれの依存関係には名前とタイプがありますが、パス属性はありません。依存関係には、必要なインポート対象の 1 つが CICS 領域に存在しない場合に、CICS が BUNDLE リソースのインストールを扱う方法を記述する属性も含まれています。BUNDLE リソースは、無効のままの状態、有効にして警告メッセージ

が出た状態、または有効にして警告が出ない状態のいずれかになります。デフォルトの動作では、すべてのインポートが必須です。BUNDLE リソースはインストールできますが、有効にはなりません。

プラットフォームにデプロイされるアプリケーションの一部としてパッケージする CICS バンドルの中では、公用リソースまたは専用リソース (アプリケーションによって自動インストールされる専用プログラムを含む) を従属として定義できます。CICS は、まず、指定されたタイプおよび名前の専用リソースが、バンドル・マニフェスト・ファイルに `<import>` エlementが含まれている CICS バンドルと同じアプリケーションに含まれているかどうかを検査します。複数のバージョンのアプリケーションがインストールされている場合は、現在のアプリケーション・コンテキストにより、インポートされる専用リソースのバージョンが決まります。そのリソースがアプリケーションの専用リソースとして見つからない場合は、CICS は CICS 領域にある公用リソースの中に指定されたタイプおよび名前に一致するリソースがあるかどうか検査し、そのリソースをインポートします。他のアプリケーションから専用リソースをインポートすることはできません。

ユーザーは `<import>` リストを拡張して、独自のユーザー・リソース・タイプを含めることができます。詳しくは、[バンドルにデプロイ可能な成果物を参照してください](#)。

### アプリケーション・エントリー・ポイント

CICS バンドルが CICS アプリケーションの一部である場合は、バンドル・マニフェスト内に 1 つ以上のアプリケーション・エントリー・ポイントを定義できます。アプリケーション・エントリー・ポイントは、アプリケーションへのアクセス・ポイントとなるリソースを識別します。アプリケーション・エントリー・ポイントは、ユーザーが、プラットフォームで実装されたアプリケーションの別のバージョンへアクセスするのを制御するのに使用されます。アプリケーション・エントリー・ポイントはまた、アプリケーション・コンテキストを作成して、アプリケーションのリソース使用をモニターし、実行中のアプリケーションを識別するためにも使用されます。

`<modify>` Element 内の `<entrypoint>` Element は、アプリケーション・エントリー・ポイントを定義します。バンドル・マニフェストには、アプリケーション・エントリー・ポイントのために 0 個以上の `<modify>` Element を含めることができます。`<entrypoint>` Element は、CICS リソースの名前、リソースのタイプ、およびアプリケーション操作を指定します。その CICS バンドルによって管理されないリソースのアプリケーション・エントリー・ポイントを指定することができます。

### ポリシー・スコープ

ポリシー・スコープは、ポリシーを特定のアプリケーション操作に限定するために使用します。アプリケーション・スコープを使用してポリシーがデプロイされると、そのポリシーは、そのアプリケーション・コンテキストでプラットフォーム、アプリケーション、およびアプリケーション・バージョン情報が一致するすべてのユーザー・タスクに適用されます。ポリシー・スコープは、さらにポリシーを、特定の操作とも一致するユーザー・タスクだけに限定するために使用されます。

`<modify>` Element 内の `<policyscope>` Element は、ポリシーのポリシー・スコープを定義します。バンドル・マニフェストには、ポリシー・スコープのために 0 個以上の `<modify>` Element を含めることができます。`<policyscope>` Element は、アプリケーション操作の名前、およびその操作のためにユーザー・タスクに適用するポリシーの名前を指定します。操作の名前は別個の `<entrypoint>` Element で定義する必要があり、ポリシーは、同じバンドル内、またはポリシー・スコープを定義するバンドルと共にデプロイされた別個のバンドル内の、`<define>` Element で定義する必要があります。

### マニフェストの例

以下の例は、プログラム定義 (アプリケーション・エントリー・ポイントでもある) を、そのアプリケーション・エントリー・ポイントのユーザー・タスクのポリシーおよびポリシー・スコープと共に格納する、CICS バンドルのマニフェストを示しています。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<manifest xmlns="http://www.ibm.com/xmlns/prod/cics/bundle" bundleVersion="1"
  bundleRelease="0" build="IntB-201705232112">
  id="CustomerRecordsStore"
  bundleMajorVer="1"
  bundleMinorVer="200"
  bundleMicroVer="4"
  <meta_directives>
    <timestamp>2018-01-25T14:59:32.092Z</timestamp>
  </meta_directives>
  <import
```



```

    name="ADDC"
    type=http://www.ibm.com/xmlns/prod/cics/bundle/TRANSACTION
    optional="false" warn="false"/> 3
  <define
    name="ADDCUST"
    type=http://www.ibm.com/xmlns/prod/cics/bundle/PROGRAM
    path="ADDCUST.program"/> 4
  <define
    name="CUSTLIB"
    type=http://www.ibm.com/xmlns/prod/cics/bundle/LIBRARY
    path="CUSTLIB.library"/> 5
  <define
    name="NewCustomer"
    type=http://www.ibm.com/xmlns/prod/cics/bundle/POLICY
    path="NewCustomer.policy"/> 6
  <modify>
    <entrypoint
      name="ADDCUST"
      type=http://www.ibm.com/xmlns/prod/cics/bundle/PROGRAM
      operation="add_customer"/> 7
    </modify>
  <modify>
    <polycscope
      name="NewCustomer"
      type=http://www.ibm.com/xmlns/prod/cics/bundle/POLICY
      operation="add_customer"/> 8
    </modify>
  </manifest>

```

1. バンドル ID およびバンドル・バージョンを含むバンドル・プロジェクトに関する情報を格納します。
2. meta directives エレメントには、バンドルが作成された日時が入ります。
3. バンドルは、トランザクション・リソースをインポートします。この CICS バンドルを使用可能にするために、プロビジョンされるシステムにこのリソースが存在して使用可能になっている必要があります。この CICS バンドルはトランザクション・リソースを宣言または管理することはありません。
4. バンドルは、リソースの名前、およびリソース属性を定義する XML ファイルへのパスを含む PROGRAM リソースを定義します。
5. バンドルは LIBRARY リソースを定義して、定義済みの PROGRAM (リソースの名前、およびそのリソース属性を定義する XML ファイルのパスを含む) を動的にロードします。
6. バンドルは、ポリシーの名前、およびポリシー属性を定義する XML ファイルへのパスを含む POLICY リソースを定義します。
7. バンドルは、特定の操作のためのアプリケーション・エン트리・ポイントとして PROGRAM リソースを宣言します。
8. バンドルは、POLICY が、指定操作と一致するタスクにのみ適用されることを宣言します。

## 関連概念

### バンドルにデプロイ可能な成果物

CICS バンドルに定義してデプロイできる成果物には、アプリケーションとシステムのイベント、Atom フィールド、チャンネル・ベースのサービス、CICS ポリシー、CICS プログラム、OSGi バンドル、XML ベースのサービスおよびトランザクションが含まれます。これらの成果物はそれぞれ 1 つ以上の CICS リソースによって表され、それらのリソースはバンドル・デプロイメントの一部として動的に作成されます。

### バンドル内の zFS 成果物の参照

いくつかの CICS リソースが、外部 zFS 成果物を参照して詳しい構成情報を探します。例えば、JVMSEVER リソースでは JVM プロファイルが必要となり、PIPELINE リソースではパイプライン構成ファイルが必要となります。これらのリソースが CICS バンドルに定義されている場合、必要となる zFS 成果物も CICS バンドルに保管され、相対 zFS 名を使用して参照される必要があります。

### バンドルのスコーピング

BUNDLE リソース定義には、関連する BUNDLE リソースに有効範囲を適用する方法として、BASESCOPE 属性があります。この属性は、プラットフォームにデプロイされているバンドルに使用したり、サービス・コンポーネント・アーキテクチャー (SCA) 複合アプリケーションが含まれているバンドルの SCA ドメインの設定に使用したりできます。

### CICS リスタート後の OSGi バンドル・リカバリー

OSGi バンドルを含む CICS 領域をリスタートすると、CICS は BUNDLE リソースをリカバリーし、OSGi バンドルを JVM サーバーのフレームワークにインストールします。

#### バンドルのセキュリティ

定義から作成された BUNDLE リソースと、アプリケーションまたはプラットフォームをインストールするときに作成した BUNDLE リソースでは、適用されるセキュリティ・プロファイルとセキュリティ検査が異なります。

#### CICS プロジェクトでの変数

変数を使用することで属性値を素早く簡単に変更することができます。各環境に固有のプロパティ・ファイルを使用して変数を解決できるので、複数環境へのデプロイメントが単純化されます。

#### **関連資料**

##### 変数およびプロパティ・ファイルの定義

変数およびそれらに関連付けられたプロパティ・ファイルを定義する場合、変数の置換が正常に行われるように、これらのルールに従います。

#### **関連情報**

##### CICS バンドル内リソースの特性

CICS リソースは、CICS バンドルで定義され、バンドル・デプロイメントの一部として動的に作成されるため、その一部の特性が変更されます。アプリケーションのアーキテクチャーを設計する、あるいは CICS バンドルで定義する既存のアプリケーションからリソースを選択する場合、これらの重要な考慮事項に注意してください。

##### アプリケーション・バージョンの専用リソース

特定の CICS リソースを、プラットフォームにインストールするアプリケーションの一部として CICS バンドルで定義した場合、そのリソースは、そのアプリケーションのそのバージョン専用のものになります。そのため、これらのタイプのリソースは、同時に複数を同じ名前で同じプラットフォーム・インスタンス上にインストールできます。

## **バンドルのスコーピング**

BUNDLE リソース定義には、関連する BUNDLE リソースに有効範囲を適用する方法として、BASESCOPE 属性があります。この属性は、プラットフォームにデプロイされているバンドルに使用したり、サービス・コンポーネント・アーキテクチャー (SCA) 複合アプリケーションが含まれているバンドルの SCA ドメインの設定に使用したりできます。

BASESCOPE は、さまざまな方法で使用できる、BUNDLE リソース定義のオプション属性です。IBM CICS Explorer を使用して、CICS 領域に定義されているすべての BUNDLE リソースを表示したり、BASESCOPE 属性の値の順に並べたりすることができます。

#### **プラットフォームのバンドルのスコーピング**

プラットフォームにデプロイされている CICS バンドルには、アプリケーション・デプロイメント中に BUNDLE リソースが作成されるときに BASESCOPE が自動的に適用されます。アプリケーション・バンドルまたはアプリケーション・バインディング・バンドルの一部としてプラットフォームにデプロイされている BUNDLE リソースには、BASESCOPE 属性に以下の形式の URI が設定されます。

```
cicsapplication://Platform/ApplicationID/MajorVersion/MinorVersion/MicroVersion
```

*Platform* はプラットフォームの名前、*ApplicationID* はアプリケーション・バンドルの ID、その後にアプリケーションのバージョン (メジャー、マイナー、マイクロ) が続きます。

CICS は、BASESCOPE 属性を使用して関連するアプリケーションとバージョンを識別するため、同じアプリケーションの複数のバージョンをプラットフォームにインストールできます。BASESCOPE 属性を使用すると、リソースをアプリケーションの適切なバージョンに制限できます。例えば、アプリケーション・バンドルまたはアプリケーション・バインディング・バンドルの一部としてパッケージされてインストールされる CICS バンドルの中で定義されている、特定のサポートされているリソース・タイプは、そのアプリケーションのそのバージョン専用のものになります。アプリケーションの専用リソースについて詳しくは、アプリケーション・バージョンの専用リソースを参照してください。

同じ cicsapplication 形式の basescope は、プラットフォーム環境の外部にある CICS バンドルに追加できます。このことは、プラットフォームやアプリケーションがないスタンドアロン CICS 領域 (SMSS) で有用

です。SMSS の場合、basescope を正しく指定することはユーザーの責任となりますが、プラットフォームでは、CICS によって正しい basescope 属性が自動的に生成されます。

## ポリシーのバンドルのスコーピング

ポリシーは CICS バンドルに定義されます。プラットフォームでは、ポリシーは特定の有効範囲にデプロイされます。有効範囲としては、アプリケーション、(アプリケーション内の) 操作、またはプラットフォームが可能です。詳細については、[ポリシー・スコープ](#)を参照してください。

## SCA ドメインへのバンドルのスコーピング

SCA 複合アプリケーションが含まれているバンドルでは、BASESCOPE 属性に特定の用途があります。複合アプリケーションは、SCA ドメインにデプロイされます。通常、SCA ドメインは単一の組織によって制御されるビジネス機能の領域を提供するサービスのセットを表します。例えば、企業の会計部門の SCA ドメインは、すべての財務関連機能をカバーし、会計の特定の領域を扱う一連の複合アプリケーションを含むことがあります。

CICS 領域には、デフォルトで、1 つの SCA ドメインがあります。CICS 領域にデプロイされたすべてのバンドルで、デフォルトの SCA ドメインは同じになりますが、値は空です。BUNDLE リソース定義の BASESCOPE 属性を使用して、SCA ドメインの値を設定できます。

また、異なる SCA ドメインを BASESCOPE 属性に指定することで、同じバンドルを CICS 領域に何度もデプロイすることができます。CICS は、ランタイム処理中に SCA ドメインとコンポジットをともに使用してサービスを特定します。要求を処理中のタスクは、サービスの有効範囲を使用できます。

BASESCOPE 属性値には固有の URI を使用します。例えば、`http://mycompany/HR` や `http://mycompany/warehouse` を使用します。CICS では、BASESCOPE 属性の値を拡張することによって、サービス、複合、および参照の名前を作成します。例えば、ローカル名が `location/taxService` のサービスを HR SCA ドメインにインストールすると、`http://mycompany/HR/location/taxService` という、有効範囲が設定された名前が作成されます。

同じ URI を拡張して新しい SCA ドメインを作成しないでください。同じ URI を拡張すると、サービス名または参照名の予期しない競合が発生する可能性があります。例えば、異なる SCA ドメインとして `http://mycompany/HR` および `http://mycompany/HR/location` を使用し、ローカル名が `location/taxService` のサービスと `taxService` という別のサービスがある場合、これらのサービスを `http://mycompany/HR` と `http://mycompany/HR/location` の両方の SCA ドメインにインストールすると、サービス名の競合が発生します。これらの値を使用して BUNDLE リソースを正常にインストールして有効にすることができますが、他のアプリケーションによってサービスが呼び出されたとき、予期しない結果とエラーが発生する可能性があります。

## 関連概念

### [バンドルにデプロイ可能な成果物](#)

CICS バンドルに定義してデプロイできる成果物には、アプリケーションとシステムのイベント、Atom フィード、チャネル・ベースのサービス、CICS ポリシー、CICS プログラム、OSGi バンドル、XML ベースのサービスおよびトランザクションが含まれます。これらの成果物はそれぞれ 1 つ以上の CICS リソースによって表され、それらのリソースはバンドル・デプロイメントの一部として動的に作成されます。

### [バンドル内の zFS 成果物の参照](#)

いくつかの CICS リソースが、外部 zFS 成果物を参照して詳しい構成情報を探します。例えば、JVM SERVER リソースでは JVM プロファイルが必要となり、PIPELINE リソースではパイプライン構成ファイルが必要となります。これらのリソースが CICS バンドルに定義されている場合、必要となる zFS 成果物も CICS バンドルに保管され、相対 zFS 名を使用して参照される必要があります。

### [CICS バンドルのマニフェストの内容](#)

それぞれの CICS バンドルには、バンドルの内容を記述するバンドル・マニフェストが含まれています。バンドル・マニフェストには、バンドルと、プロビジョン時に CICS 領域で設計または変更するリソース、さらに CICS バンドルを正常に使用可能にするために必要な依存関係が記述されます。

### [CICS リスタート後の OSGi バンドル・リカバリー](#)

OSGi バンドルを含む CICS 領域をリスタートすると、CICS は BUNDLE リソースをリカバリーし、OSGi バンドルを JVM サーバーのフレームワークにインストールします。

### [バンドルのセキュリティ](#)

定義から作成された BUNDLE リソースと、アプリケーションまたはプラットフォームをインストールするときに作成した BUNDLE リソースでは、適用されるセキュリティー・プロファイルとセキュリティー検査が異なります。

#### CICS プロジェクトでの変数

変数を使用することで属性値を素早く簡単に変更することができます。各環境に固有のプロパティー・ファイルを使用して変数を解決できるので、複数環境へのデプロイメントが単純化されます。

#### **関連資料**

##### 変数およびプロパティー・ファイルの定義

変数およびそれらに関連付けられたプロパティー・ファイルを定義する場合、変数の置換が正常に行われるように、これらのルールに従います。

#### **関連情報**

##### CICS バンドル内リソースの特性

CICS リソースは、CICS バンドルで定義され、バンドル・デプロイメントの一部として動的に作成されるため、その一部の特性が変更されます。アプリケーションのアーキテクチャーを設計する、あるいは CICS バンドルで定義する既存のアプリケーションからリソースを選択する場合、これらの重要な考慮事項に注意してください。

##### アプリケーション・バージョンの専用リソース

特定の CICS リソースを、プラットフォームにインストールするアプリケーションの一部として CICS バンドルで定義した場合、そのリソースは、そのアプリケーションのそのバージョン専用のものになります。そのため、これらのタイプのリソースは、同時に複数を同じ名前で同じプラットフォーム・インスタンス上にインストールできます。

## **CICS リスタート後の OSGi バンドル・リカバリー**

OSGi バンドルを含む CICS 領域をリスタートすると、CICS は BUNDLE リソースをリカバリーし、OSGi バンドルを JVM サーバーのフレームワークにインストールします。

CICS バンドルにパッケージされる OSGi バンドルは、CSD に保管されません。BUNDLE リソース自体はカタログに保管されるため、CICS 領域のリスタート後、BUNDLE リソースが復元されると OSGi バンドルは動的に再作成されます。

CICS のコールド・リスタート、ウォーム・リスタート、または緊急リスタート後、JVM サーバーは、BUNDLE リソースのリカバリーとは非同期に始動されます。CICS リスタートで OSGi バンドルを正常に復元するには、JVM サーバーが完全に使用可能でなければなりません。したがって、BUNDLE リソースは CICS 開始の最後の段階時にリカバリーされますが、OSGi バンドルがインストールされるのは、JVM サーバーがその開始を完了したときのみです。

BUNDLE リソースとそれに含まれている OSGi バンドルは正しい順にインストールされ、CICS バンドルと OSGi バンドルの両方の間の依存関係がフレームワークで確実に解決されるようになります。CICS が OSGi バンドルをインストールできない場合、BUNDLE リソースは Disabled 状態でインストールされます。IBM CICS Explorer を使用すると、BUNDLE リソース、OSGi バンドル、および OSGi サービスの状態を表示できます。

#### **関連概念**

##### バンドルにデプロイ可能な成果物

CICS バンドルに定義してデプロイできる成果物には、アプリケーションとシステムのイベント、Atom フィード、チャンネル・ベースのサービス、CICS ポリシー、CICS プログラム、OSGi バンドル、XML ベースのサービスおよびトランザクションが含まれます。これらの成果物はそれぞれ 1 つ以上の CICS リソースによって表され、それらのリソースはバンドル・デプロイメントの一部として動的に作成されます。

##### バンドル内の zFS 成果物の参照

いくつかの CICS リソースが、外部 zFS 成果物を参照して詳しい構成情報を探します。例えば、JVMSERVER リソースでは JVM プロファイルが必要となり、PIPELINE リソースではパイプライン構成ファイルが必要となります。これらのリソースが CICS バンドルに定義されている場合、必要となる zFS 成果物も CICS バンドルに保管され、相対 zFS 名を使用して参照される必要があります。

##### CICS バンドルのマニフェストの内容



それぞれの CICS バンドルには、バンドルの内容を記述するバンドル・マニフェストが含まれています。バンドル・マニフェストには、バンドルと、プロビジョン時に CICS 領域で設計または変更するリソース、さらに CICS バンドルを正常に使用可能にするために必要な依存関係が記述されます。

#### バンドルのスコーピング

BUNDLE リソース定義には、関連する BUNDLE リソースに有効範囲を適用する方法として、BASESCOPE 属性があります。この属性は、プラットフォームにデプロイされているバンドルに使用したり、サービス・コンポーネント・アーキテクチャー (SCA) 複合アプリケーションが含まれているバンドルの SCA ドメインの設定に使用したりできます。

#### バンドルのセキュリティ

定義から作成された BUNDLE リソースと、アプリケーションまたはプラットフォームをインストールするときに作成した BUNDLE リソースでは、適用されるセキュリティ・プロファイルとセキュリティ検査が異なります。

#### CICS プロジェクトでの変数

変数を使用することで属性値を素早く簡単に変更することができます。各環境に固有のプロパティ・ファイルを使用して変数を解決できるので、複数環境へのデプロイメントが単純化されます。

#### **関連資料**

##### 変数およびプロパティ・ファイルの定義

変数およびそれらに関連付けられたプロパティ・ファイルを定義する場合、変数の置換が正常に行われるように、これらのルールに従います。

#### **関連情報**

##### CICS バンドル内リソースの特性

CICS リソースは、CICS バンドルで定義され、バンドル・デプロイメントの一部として動的に作成されるため、その一部の特性が変更されます。アプリケーションのアーキテクチャーを設計する、あるいは CICS バンドルで定義する既存のアプリケーションからリソースを選択する場合、これらの重要な考慮事項に注意してください。

##### アプリケーション・バージョンの専用リソース

特定の CICS リソースを、プラットフォームにインストールするアプリケーションの一部として CICS バンドルで定義した場合、そのリソースは、そのアプリケーションのそのバージョン専用のものになります。そのため、これらのタイプのリソースは、同時に複数と同じ名前と同じプラットフォーム・インスタンス上にインストールできます。

## **バンドルのセキュリティ**

定義から作成された BUNDLE リソースと、アプリケーションまたはプラットフォームをインストールするときに作成した BUNDLE リソースでは、適用されるセキュリティ・プロファイルとセキュリティ検査が異なります。

#### **BUNDLE 定義のセキュリティ**

個々の CICS 領域に定義されている CICS バンドルについては、他の CICS リソースの場合と同様に、BUNDLE リソースに対してアクションを実行すると、CICS コマンドおよびリソース・セキュリティ検査が適用されます。CICS 領域の XRES システム初期設定パラメーターでは、特に BUNDLE リソース・タイプに対してセキュリティ検査を実行するかどうかを指定します。バンドルのリソース・セキュリティは、BUNDLE セキュリティ・プロファイルによって制御されます。BUNDLE セキュリティ・プロファイルは、BUNDLEPART、OSGIBUNDLE、および OSGISERVICE リソースにも適用されます。

スタンドアロンの CICS バンドルについては、アプリケーション・エントリー・ポイントを含む場合に限り、使用可能または使用不可にする必要があります。スタンドアロン CICS バンドルのセキュリティ・プロファイルは BUNDLE リソース・タイプと BUNDLE リソースの名前を指定します。このプロファイルで UPDATE アクセス権限を付与されたオペレーターはリソースを使用可能または使用不可に設定できます。

CICS バンドルによって動的に作成されたリソースについては、リソースがバンドルのインストール時に動的に作成されたか、CICS バンドルに変更を加えてリソースを操作した場合、そのリソース・タイプに対して、追加の CICS コマンド・セキュリティ検査およびリソース・セキュリティ検査は行われません。ただし、それらのリソース・タイプに対する CICS コマンド・セキュリティおよびリソース・セキュリティは、動的に作成されたリソースを照会するとき、または動的に作成されたリソースを直接操作する場合に適用されます。

## アプリケーションとプラットフォームによって生成された BUNDLE リソースのセキュリティ

アプリケーションまたはプラットフォームが CICSplex にインストールされている場合、デプロイメントの一部である CICS バンドルが CICSplex SM によって適切な CICS 領域に動的に作成されます。各 BUNDLE リソースは動的に作成され、\$ 文字で始まる固有の名前が生成されてそれぞれに与えられます。

ユーザーにプラットフォームまたはアプリケーションをインストールする 権限を付与するには、CICSplex SM で CLOUD.DEF、CLOUD.PLATFORM、および CLOUD.APPLICATION セキュリティー・プロファイルに対する適切なアクセス権限をユーザーに付与します。ユーザーにこの権限を付与すると、CICS 領域の動的に作成された BUNDLE リソースをインストールする権限もユーザーに付与されます。CICS バンドルがアプリケーションまたはプラットフォームの一部としてインストールされている場合、CICS コマンド・セキュリティ検査およびリソース・セキュリティ検査は実行されません。CICS バンドルがアプリケーションまたはプラットフォームの一部としてインストールされている場合、CICSplex SM のシミュレートされた CICS セキュリティー検査も実行されません。

アプリケーションまたはプラットフォームのインストール時に動的に作成された BUNDLE リソースを使用可能/使用不可にしたり、有効/無効にしたり、照会したりすると、CICS コマンドおよびリソース・セキュリティ検査と CICSplex SM のシミュレートされた CICS セキュリティー検査は、個々の CICS バンドルに対して直接アクションを実行した場合にのみ適用されます。CICS バンドルが使用可能/使用不可にされたり、有効/無効にされたり、照会されたりすると、アプリケーションまたはプラットフォームに対して実行したアクションによって、アプリケーションまたはプラットフォームのセキュリティ検査が代わりに適用されます。アプリケーションまたはプラットフォームのインストール時に作成された個々の CICS バンドルを直接破棄することはできません。

**ヒント:** アプリケーションまたはプラットフォームのインストール時に動的に作成された個々の CICS バンドルに対するアクションのセキュリティを設定するには、BUNDLE リソース・タイプおよびリソース名 \$\* を指定してセキュリティ・プロファイルをセットアップします。BUNDLE.\$\* に対する UPDATE アクセス権限を持つユーザーは、プラットフォームやアプリケーションのために作成された BUNDLE リソースを使用可能/使用不可にしたり、有効/無効にしたりできます。また、READ アクセス権限を持つユーザーは、それらの BUNDLE リソースを照会することができます。

アプリケーションまたはプラットフォームに対してインストールされた各 CICS バンドルの内部に定義されているリソースは、動的に作成された BUNDLE リソースのインストール中に、CICS 領域に動的に作成されます。個別のリソース・タイプに対する CICS コマンド・セキュリティ検査およびリソース・セキュリティ検査は、これらのリソースが CICS 領域で動的に作成された場合は実行されません。ただし、個別のリソース・タイプに対する CICS コマンド・セキュリティおよびリソース・セキュリティは、動的に作成されたリソースを照会するときに適用されます。CICS 領域で動的に作成されたリソースの有効化、無効化、または破棄を直接行うことはできません。

さまざまなプラットフォームにデプロイされるアプリケーションについて、セキュリティ対策を個々の PROGRAM リソースに適用する場合、アプリケーション・エントリー・ポイントとして宣言されているプログラムは保護しますが、そのアプリケーション内の他のプログラムは保護しないようにしてください。あるプラットフォーム上にデプロイされるアプリケーションの一部を構成するプログラムについて指定するセキュリティ設定値は、パブリックなプログラムとプライベートなプログラムの両方に適用されます。アプリケーションのバージョンは考慮されません。アプリケーション・エントリー・ポイントとして宣言されるプログラムには、使用環境内で固有の PROGRAM リソース名が必要です。しかし、そのアプリケーションの下位レベルで実行されるプログラムを保護すると、別のアプリケーションで同じ名前のプログラムが実行されている可能性があり、その場合、予期しない結果になる可能性があります。この場合、アプリケーション・エントリー・ポイントとして宣言されているプログラムについてはアクセス権限がユーザーに付与されている一方、そのプログラム名の別のインスタンスによるセキュリティ設定が有効になっているために、アプリケーションの下位レベルで実行されるプログラムにはアクセスする許可がないということがあり得ます。アプリケーション・エントリー・ポイント・プログラムとして宣言されているプログラムに適用するセキュリティ対策としては、アプリケーション全体に適用されるものを考慮してください。

以前の CICS リリースで CICS バンドルを使用していた場合は、これらのバンドルに関してユーザーに与えたセキュリティ権限を確認してください。CICS バンドルのセキュリティのセットアップ方法によっては、個々の CICS バンドルに対するアクションの実行権限を持つユーザーが、バンドル・インストールの過程で動的に作成されるリソースに対してアクションを実行できるようになっている可能性があります。BUNDLE リソースに対する権限のレベルが引き続き適切であることを確認してください。



アプリケーションとプラットフォームのセキュリティ、および CLOUD.DEF、CLOUD.PLATFORM、CLOUD.APPLICATION の各セキュリティ・プロファイルについて詳しくは、[プラットフォームおよびアプリケーションのセキュリティ](#) を参照してください。

## 関連概念

### [バンドルにデプロイ可能な成果物](#)

CICS バンドルに定義してデプロイできる成果物には、アプリケーションとシステムのイベント、Atom フィード、チャンネル・ベースのサービス、CICS ポリシー、CICS プログラム、OSGi バンドル、XML ベースのサービスおよびトランザクションが含まれます。これらの成果物はそれぞれ 1 つ以上の CICS リソースによって表され、それらのリソースはバンドル・デプロイメントの一部として動的に作成されます。

### [バンドル内の zFS 成果物の参照](#)

いくつかの CICS リソースが、外部 zFS 成果物を参照して詳しい構成情報を探します。例えば、JVMSERVER リソースでは JVM プロファイルが必要となり、PIPELINE リソースではパイプライン構成ファイルが必要となります。これらのリソースが CICS バンドルに定義されている場合、必要となる zFS 成果物も CICS バンドルに保管され、相対 zFS 名を使用して参照される必要があります。

### [CICS バンドルのマニフェストの内容](#)

それぞれの CICS バンドルには、バンドルの内容を記述するバンドル・マニフェストが含まれています。バンドル・マニフェストには、バンドルと、プロビジョン時に CICS 領域で設計または変更するリソース、さらに CICS バンドルを正常に使用可能にするために必要な依存関係が記述されます。

### [バンドルのスコーピング](#)

BUNDLE リソース定義には、関連する BUNDLE リソースに有効範囲を適用する方法として、BASESCOPE 属性があります。この属性は、プラットフォームにデプロイされているバンドルに使用したり、サービス・コンポーネント・アーキテクチャー (SCA) 複合アプリケーションが含まれているバンドルの SCA ドメインの設定に使用したりできます。

### [CICS リスタート後の OSGi バンドル・リカバリー](#)

OSGi バンドルを含む CICS 領域をリスタートすると、CICS は BUNDLE リソースをリカバリーし、OSGi バンドルを JVM サーバーのフレームワークにインストールします。

### [CICS プロジェクトでの変数](#)

変数を使用することで属性値を素早く簡単に変更することができます。各環境に固有のプロパティ・ファイルを使用して変数を解決できるので、複数環境へのデプロイメントが単純化されます。

## 関連資料

### [変数およびプロパティ・ファイルの定義](#)

変数およびそれらに関連付けられたプロパティ・ファイルを定義する場合、変数の置換が正常に行われるように、これらのルールに従います。

## 関連情報

### [CICS バンドル内リソースの特性](#)

CICS リソースは、CICS バンドルで定義され、バンドル・デプロイメントの一部として動的に作成されるため、その一部の特性が変更されます。アプリケーションのアーキテクチャーを設計する、あるいは CICS バンドルで定義する既存のアプリケーションからリソースを選択する場合、これらの重要な考慮事項に注意してください。

### [アプリケーション・バージョンの専用リソース](#)

特定の CICS リソースを、プラットフォームにインストールするアプリケーションの一部として CICS バンドルで定義した場合、そのリソースは、そのアプリケーションのそのバージョン専用のものになります。そのため、これらのタイプのリソースは、同時に複数を同じ名前と同じプラットフォーム・インスタンス上にインストールできます。

## CICS プロジェクトでの変数

変数を使用することで属性値を素早く簡単に変更することができます。各環境に固有のプロパティ・ファイルを使用して変数を解決できるので、複数環境へのデプロイメントが単純化されます。

### 変数の使用

一般的には、リソース定義の属性値は、複数の異なる環境にインストールする前に変更する必要があります。例えば、データ・セットは、開発環境、テスト環境、および実稼働環境ごとに異なる高位修飾子を持

つ可能性があります。変数を使用すると、デプロイ先の環境に応じて属性値の一部を変更することができます。その際、各環境に固有の `variables.properties` ファイルを使用します。

リソースを CICS にインストールする前に、`--resolve` オプションを指定して CICS Build Toolkit を実行することにより、デプロイメントの際に変数が解決されます。変数の解決に使用されるプロパティ・ファイルは、変数がスタンドアロン・バンドルにあるか、アプリケーションの一部として組み込まれているかに応じて異なります。

変数を作成するための最も確実な方法は、CICS Explorer の「**変数の挿入**」ウィザードまたは「**変数に値を抽出します**」ウィザードを使用する方法です。詳しくは、[CICS Explorer 製品資料内の『Creating variables』](#)を参照してください。

## 変数の使用例

以下の例は、変数を効果的に使用するためのいくつかのシナリオです。

- Java アプリケーションの CICS バンドル・パーツ `.osgibundle`、`.warbundle`、`.ebabundle`、および `.earbundle` に含まれている JVM サーバー名 (JVM サーバー名が環境ごとに異なる場合)。
- FILE および LIBRARY 定義のデータ・セット名 (データ・セット名には環境の修飾子が含まれます)。
- PROGRAM 定義での CEDF の指定 (デバッグは開発環境に適していますが、実動には適していません)。
- VSAM ファイルのストリング (開発環境で必要なストリングはほんの少しですが、実動ではアクセス時間を改善するためにより多くのストリングが必要になります)。

## 関連概念

### バンドルにデプロイ可能な成果物

CICS バンドルに定義してデプロイできる成果物には、アプリケーションとシステムのイベント、Atom フィード、チャンネル・ベースのサービス、CICS ポリシー、CICS プログラム、OSGi バンドル、XML ベースのサービスおよびトランザクションが含まれます。これらの成果物はそれぞれ 1 つ以上の CICS リソースによって表され、それらのリソースはバンドル・デプロイメントの一部として動的に作成されます。

### バンドル内の zFS 成果物の参照

いくつかの CICS リソースが、外部 zFS 成果物を参照して詳しい構成情報を探します。例えば、JVMSERVER リソースでは JVM プロファイルが必要となり、PIPELINE リソースではパイプライン構成ファイルが必要となります。これらのリソースが CICS バンドルに定義されている場合、必要となる zFS 成果物も CICS バンドルに保管され、相対 zFS 名を使用して参照される必要があります。

### CICS バンドルのマニフェストの内容

それぞれの CICS バンドルには、バンドルの内容を記述するバンドル・マニフェストが含まれています。バンドル・マニフェストには、バンドルと、プロビジョン時に CICS 領域で設計または変更するリソース、さらに CICS バンドルを正常に使用可能にするために必要な依存関係が記述されます。

### バンドルのスコーピング

BUNDLE リソース定義には、関連する BUNDLE リソースに有効範囲を適用する方法として、BASESCOPE 属性があります。この属性は、プラットフォームにデプロイされているバンドルに使用したり、サービス・コンポーネント・アーキテクチャー (SCA) 複合アプリケーションが含まれているバンドルの SCA ドメインの設定に使用したりできます。

### CICS リスタート後の OSGi バンドル・リカバリー

OSGi バンドルを含む CICS 領域をリスタートすると、CICS は BUNDLE リソースをリカバリーし、OSGi バンドルを JVM サーバーのフレームワークにインストールします。

### バンドルのセキュリティ

定義から作成された BUNDLE リソースと、アプリケーションまたはプラットフォームをインストールするときに作成した BUNDLE リソースでは、適用されるセキュリティ・プロファイルとセキュリティ検査が異なります。

## 関連資料

[変数およびプロパティ・ファイルの定義](#)

変数およびそれらに関連付けられたプロパティ・ファイルを定義する場合、変数の置換が正常に行われるように、これらのルールに従います。

## 関連情報

### CICS バンドル内リソースの特性

CICS リソースは、CICS バンドルで定義され、バンドル・デプロイメントの一部として動的に作成されるため、その一部の特性が変更されます。アプリケーションのアーキテクチャーを設計する、あるいは CICS バンドルで定義する既存のアプリケーションからリソースを選択する場合、これらの重要な考慮事項に注意してください。

### アプリケーション・バージョンの専用リソース

特定の CICS リソースを、プラットフォームにインストールするアプリケーションの一部として CICS バンドルで定義した場合、そのリソースは、そのアプリケーションのそのバージョン専用のものになります。そのため、これらのタイプのリソースは、同時に複数を同じ名前で同じプラットフォーム・インスタンス上にインストールできます。

## 変数およびプロパティ・ファイルの定義

変数およびそれらに関連付けられたプロパティ・ファイルを定義する場合、変数の置換が正常に行われるように、これらのルールに従います。

### **プロパティ・ファイルでの変数の定義**

プロパティ・ファイルでの変数の定義には、以下のルールが適用されます。

- 変数名に有効な文字は、大文字、小文字、0 から 9 までの数字、ピリオド、下線、およびハイフンのみです。変数名は 255 文字までに制限されています。
- プロパティ・ファイルは、CICS バンドルまたはアプリケーション・バインディングに配置できます。これらのファイルは、プロジェクトのルートになければならず、`variables.properties` という名前であればなりません。
- スタンドアロン・バンドルを解決する際に個別のプロパティ・ファイルを指定できますが、このオプションはアプリケーションではサポートされていません。スタンドアロン・バンドルを使用している場合、プロパティ・ファイルの名前に関する制約事項はありません。
- プロパティ・ファイルは、Java プロパティ・ファイルの標準形式に準拠し、ISO-8859-1 でエンコードされている必要があります。詳細については、[Java プラットフォーム、Standard Edition 7 API 仕様の Java プロパティ・ファイル](#)を参照してください。
- バンドルで使用される変数は、そのバンドルのプロパティ・ファイルで定義されていなければなりません。リソース定義に置き換える場合、バンドルで指定された値はその属性の有効値になる必要があります。
- 変数の値は、他の変数を参照することはできません。例えば、`hlq=${prod.hlq}` は無効です。
- プロパティ・ファイルに変数が複数回定義されている場合は、最後のインスタンスが使用されます。

### **バンドル・パーツでの変数の参照**

バンドル・パーツでの変数の参照には、以下のルールが適用されます。

- 変数名は `${ }` の文字によって区切られます。
- 変数は、バンドル・パーツの属性または要素内を含め、すべての CICS バンドル・パーツで使用できます。
- 1 つの属性またはタグに配置できる変数の数に制限はありません。
- 変数をネストすることはできません。また、CICS リソース定義名の一部として使用することはできません。

## 関連概念

### バンドルにデプロイ可能な成果物

CICS バンドルに定義してデプロイできる成果物には、アプリケーションとシステムのイベント、Atom フィード、チャネル・ベースのサービス、CICS ポリシー、CICS プログラム、OSGi バンドル、XML ベースのサービスおよびトランザクションが含まれます。これらの成果物はそれぞれ 1 つ以上の CICS リソースによって表され、それらのリソースはバンドル・デプロイメントの一部として動的に作成されます。

### バンドル内の zFS 成果物の参照

いくつかの CICS リソースが、外部 zFS 成果物を参照して詳しい構成情報を探します。例えば、JVMSERVER リソースでは JVM プロファイルが必要となり、PIPELINE リソースではパイプライン構成ファイルが必要となります。これらのリソースが CICS バンドルに定義されている場合、必要となる zFS 成果物も CICS バンドルに保管され、相対 zFS 名を使用して参照される必要があります。

### CICS バンドルのマニフェストの内容

それぞれの CICS バンドルには、バンドルの内容を記述するバンドル・マニフェストが含まれています。バンドル・マニフェストには、バンドルと、プロビジョン時に CICS 領域で設計または変更するリソース、さらに CICS バンドルを正常に使用可能にするために必要な依存関係が記述されます。

### バンドルのスコーピング

BUNDLE リソース定義には、関連する BUNDLE リソースに有効範囲を適用する方法として、BASESCOPE 属性があります。この属性は、プラットフォームにデプロイされているバンドルに使用したり、サービス・コンポーネント・アーキテクチャー (SCA) 複合アプリケーションが含まれているバンドルの SCA ドメインの設定に使用したりできます。

### CICS リスタート後の OSGi バンドル・リカバリー

OSGi バンドルを含む CICS 領域をリスタートすると、CICS は BUNDLE リソースをリカバリーし、OSGi バンドルを JVM サーバーのフレームワークにインストールします。

### バンドルのセキュリティ

定義から作成された BUNDLE リソースと、アプリケーションまたはプラットフォームをインストールするときに作成した BUNDLE リソースでは、適用されるセキュリティ・プロファイルとセキュリティ検査が異なります。

### CICS プロジェクトでの変数

変数を使用することで属性値を素早く簡単に変更することができます。各環境に固有のプロパティ・ファイルを使用して変数を解決できるので、複数環境へのデプロイメントが単純化されます。

## **関連情報**

### CICS バンドル内リソースの特性

CICS リソースは、CICS バンドルで定義され、バンドル・デプロイメントの一部として動的に作成されるため、その一部の特性が変更されます。アプリケーションのアーキテクチャーを設計する、あるいは CICS バンドルで定義する既存のアプリケーションからリソースを選択する場合、これらの重要な考慮事項に注意してください。

### アプリケーション・バージョンの専用リソース

特定の CICS リソースを、プラットフォームにインストールするアプリケーションの一部として CICS バンドルで定義した場合、そのリソースは、そのアプリケーションのそのバージョン専用のものになります。そのため、これらのタイプのリソースは、同時に複数を同じ名前で同じプラットフォーム・インスタンス上にインストールできます。

## **端末リソースの定義**

使用する端末のアクセス方法のタイプに応じて、2つの異なる方法で端末リソースを定義できます。

1. ACF/SNA LU は、CSD に明示的に定義されるか、CICS 自動インストール機能 (*autoinstall*) を使用している場合にはモデル端末定義を使用して CSD に定義されます。自動インストールを使用して、ログオン時に CICS が端末リソース定義を動的にインストールするようにします。CICS は CSD に記録されている TERMINAL と TYPETERM 定義から端末入力を作成するのに必要な情報を取得します。このプロセスの詳細な説明については、[245 ページの『自動インストール』](#)を参照してください。

z/OS Communications Server 定義を、オフラインで CICS ユーティリティ・プログラム DFHCSDUP の DEFINE コマンドを使用して、またはオンラインで CEDA DEFINE コマンドを使用して追加できます。端末定義を CICS の初期化中にインストールする場合は、コールド・スタート中に使用されるグループ・リストに、定義を含むグループの名前を追加する必要があります。それ以外の場合は、CEDA INSTALL GROUP(groupname) コマンドをオンラインで使用して、定義のグループをインストールできます。詳しくは、[GRPLIST システム初期設定パラメーター](#)を参照してください。

また、各 LU は、z/OS Communications Server の定義ステートメントの ACF/SNA に定義する必要があります。

2. 非 SNA LU は、DFHTCT マクロを使用して端末管理テーブル (TCT) に定義されます。



CICS の初期化中、CICS は TCT システム 初期設定パラメーターによって指定された TCT をロードし、TCT で定義されたこれらの LU は、CICS のリソースとしてインストールされます。また、これらの LU をオペレーティング・システムに認識させ、DD ステートメントを各 LU の CICS 始動ジョブ・ストリームに含める必要があります。

## z/OS Communications Server 端末の定義

CICS 領域は、z/OS Communications Server サービスを使用して、端末やその他の領域と通信できます。

z/OS Communications Server サービスを使用するには、以下を行う必要があります。

1. SYS1.VTAMLST の APPL ステートメントを使用して、CICS を ACF/Communications Server に定義します。CICS の APPL ステートメントの定義の詳細については、[CICS 領域をアプリケーションとして SNA に定義する](#)を参照してください。
2. CICS が使用する端末のリソースを z/OS Communications Server に定義します。
3. 使用する端末のリソースを CICS に定義します。

### CICS 端末リソースの z/OS Communications Server への定義

SNA 端末の場合に CICS が使用する各端末または各論理装置 (LU) は、z/OS Communications Server に定義する必要があります。

端末はローカルまたはリモートとして定義することができます。

#### ローカルの z/OS Communications Server 端末

チャネル接続クラスター・コントローラーに接続された SNA 端末、またはローカル制御装置から接続された非 SNA 3270 端末の可能性あります。

#### リモート z/OS Communications Server 端末

チャネル接続通信コントローラーを使用して SDLC ラインから接続された SNA クラスター・コントローラーに接続します。通信コントローラーも、2 進データ同期 (BSC) 行によりリモート端末を接続できるようにするコードとともにロードすることができます。

z/OS Communications Server のテーブルで、端末、コントローラー、ラインを定義します。z/OS Communications Server には、通信する端末のネットワークを記述するテーブルがあります。z/OS Communications Server は、これらのテーブルを使用して CICS と端末との間のデータのフローを管理します。ネットワークのノードとしてです。SNA 端末の場合、各端末、または各論理装置 (LU) は、z/OS Communications Server ドメイン全体で一意である z/OS Communications Server のノード名を使用して、z/OS Communications Server のテーブルで定義する必要があります。

z/OS Communications Server 3.3 以降を使用している場合は、z/OS Communications Server の MDLTAB および ASLTAB マクロを使用して、AUTINSTMODEL の名前、プリンター、および代替プリンターを z/OS Communications Server に定義できます。これらの定義は自動インストールのモデルとプリンターを選択するために CICS に受け渡されます。

リソースの z/OS Communications Server への定義について詳しくは、[z/OS Communications Server: SNA リソース定義解説書](#)を参照してください。

### CICS への端末リソースの定義

特定の z/OS Communications Server 端末または論理装置を、CICS システム定義ファイル (CSD) で明示的に定義することができます。端末が CSD に明示的な定義を持っていない場合、CICS は CICS 自動インストール機能を使用して、端末のログオン時に端末の定義を動的に作成してインストールできます。

## このタスクについて

### CICS CSD での端末リソースの明示的定義

z/OS Communications Server 端末または論理装置が CSD で明示的に定義されている場合は、端末が TERMINAL 名と NETNAME を持ちますが、これは z/OS Communications Server のノード名と同じです。この方法で定義された端末には、CICS の起動時にインストールされた端末入力があります。

### CICS の自動インストール機能を使用して、端末の定義を動的に作成してインストールします。

CICS は AUTINSTMODEL および AUTINSTNAME 属性を使用して作成された TYPETERM およびモデル TERMINAL 定義を参照して端末を自動インストールできます。TYPETERM および TERMINAL 定義に

ついて詳しくは、268 ページの『モデル端末定義の自動インストール』およびグループ DFHTERM 内のモデル TERMINAL 定義を参照してください。

自動インストールを使用する場合は、CICS リソース定義が z/OS Communications Server のリソース定義に正確に一致していることを確認する必要があります。z/OS Communications Server のログモード定義とそれらが一致する CICS 自動インストールのモデル定義のプログラミング情報については、[Coding entries in the VTAM LOGON mode table](#) を参照してください。

システム初期設定パラメーター TCTUALLOC=ANY を指定する場合、CICS は可能であれば 16 MB 境界より上の z/OS Communications Server 端末の端末管理テーブル・ユーザー領域 (TCTUA) を保管します。

### 端末のシャットダウンの時間制限の定義

CICS がシャットダウンしているときに、時間制限を指定し、その時間内に CICS が使用するすべての SNA LU をシャットダウンすることができます。

(これは、中断状態の端末が CICS のシャットダウンを停止するのを防ぐためです。) この時間制限は TCSWAIT システム初期設定パラメーターで指定します。また、時間制限を超過した場合に、CICS が実行するアクションも指定できます。アクションを TCSACTN システム初期設定パラメーターで指定します。TCSWAIT と TCSACTN の適切な値の選択についての詳細は、次のトピックで説明します。

### TCSWAIT の適切な値の選択

**TCSWAIT** システム初期設定パラメーターで指定する値は、通常的环境ですべての SNA LU と接続が適切な方法でシャットダウンするのに十分な大きさにする必要があります。

この値を選択する指針として、次の 2 つの CICS の端末管理シャットダウン・メッセージの間の経過時間よりも、少し大きな値を使用することを検討してください。

```
DFHZC2305 Termination of VTAM sessions beginning
DFHZC2316 VTAM ACB is closed
```

時間制限を行わない場合 (つまりすべての端末がハングしないと想定する場合)、**TCSWAIT** システム初期設定パラメーターで NO を指定します。このパラメーターについて詳しくは、[TCSWAIT システム初期設定パラメーター](#)を参照してください。

注: VTAM は、現在は z/OS Communications Server for SNA と呼ばれています。

### CICS が中断状態の端末のみを報告するよう指定

中断状態の端末を報告し、その端末の強制終了を試行しない場合は、TCSWAIT=mm (適切な時間間隔を指定) および TCSACTN=NONE システム初期設定パラメーターを指定します。

### CICS がすべての中断状態の端末を強制終了するよう指定

すべての中断状態の端末の強制終了を試行するには、TCSWAIT=mm (適切な時間間隔を指定) および TCSACTN=UNBIND システム初期設定パラメーターを指定します。

## 順次 (BSAM) 装置

入出力順次データ・セットのペアを使用して、端末を CICS にシミュレートできます。例えば、想定している端末が利用できるようになる前に、これを行ってアプリケーション・プログラムをテストできます。

そのためには、次の DFHTCT TYPE= マクロをコーディングします。

```
DFHTCT TYPE=INITIAL,
        ACCMETH=(NONVTAM)    defining the access
                               method
```

次のマクロ命令を連続して定義します。

```
DFHTCT TYPE=SDSCI,
        DSCNAME=isadscn,      defining the input
        DDNAME=indd, ...      data set
DFHTCT TYPE=SDSCI,
        DSCNAME=osadscn,      defining the output
        DDNAME=outdd, ...     data set
DFHTCT TYPE=LINE,
        ISADSCN=isadscn,
        OSADSCN=osadscn, ...
```



```
DFHTCT TYPE=TERMINAL,  
TRMIDNT=name, ...
```

DFHTCT TYPE=SDSCI マクロによって定義される 2 つのデータ・セットは、DFHTCT TYPE=TERMINAL マクロの TRMIDNT オペランドで指定されている名前によって認識される CICS 端末をシミュレートします。入力データ・セットと出力データ・セットの DSCNAME は、それぞれ DFHTCT TYPE=LINE マクロの ISADSCN オペランドと OSADSCN オペランドで指定する必要があります。

SDSCI マクロによって定義されている各順次データ・セットの DD ステートメントをコーディングする必要があります。DD ステートメントの DD 名は、SDSCI マクロの DDNAME パラメーター (またはデフォルトの DSCNAME パラメーター) でコーディングされた名前と同じにする必要があります。例えば、順次入出力に対して次の DD ステートメントを使用することができます。

```
//CARDIN DD *,DCB=BLKSIZE=80  
.  
Statements containing valid transactions  
.  
/*  
//PRINTER DD SYSOUT=A,DCB=BLKSIZE=132
```

I/O の組み合わせのこの例は、端末を CICS アプリケーション・プログラムにシミュレートします。コピーブック DFH\$TCTS のこの CARDIN/PRINTER の組み合わせをサポートする SDSCI ステートメントの例があります。それは、サンプル TCT、DFHTCT5\$ で定義されています。DFH\$TCTS は CICSTS56.CICS.SDFHSAMP で提供されています。アプリケーション・プログラムへの入力は、入力ストリーム (CARDIN) を通じて実行依頼され、端末への出力は出力ストリーム (PRINTER) に送信されます。BLKSIZE パラメーターがデータ・セットの TCT で定義されている場合、このパラメーターを JCL から省略できます。ただし、このパラメーターが TCT で定義されていない場合、ブロック・サイズはデフォルトで 0 に設定されます。その場合に、データ・セットの DD ステートメントからも省略すると、メッセージ IEC141I 013-34 が表示されます。それ以外にも、いくつかの CICS 提供のインストール検査手順の I/O 順次データ・セットに関する DD ステートメントの例があります。それらの例は、インストール後に CICSTS56.CICS.SDFHINST で見つけることができます。

また、2 つの DASD データ・セットを使用して、端末をシミュレートすることもできます。SDSCI マクロによって定義されている各データ・セットの DD ステートメントをコーディングする必要があり、DD ステートメントの DD 名を SDSCI マクロの DDNAME (または DSCNAME) パラメーターでコーディングされた名前にする必要があります。

例えば、次のようにコーディングすることができます。

```
//DISKIN1 DD DSN=SIMULATD.TERMINAL.IN,  
// UNIT=3380,DISP=OLD,VOL=SER=valid  
//DISKOT1 DD DSN=SIMULATD.TERMINAL.OUT,  
// UNIT=3380,DISP=OLD,VOL=SER=valid
```

このシミュレートされた端末からの入力は、DISKIN1 データ・セットから読み取られます。端末への出力は、DISKOT1 データ・セットに書き込まれます。

入力ファイルの各ステートメント (前に使用した例の CARDIN または DISKIN1) は、X'E0' を表す文字で終了する必要があります。このデータの終わりの 16 進値の標準的な EBCDIC 記号は円記号 (¥) であり、これが事前生成システムの CICS に定義された文字です。EODI システム初期設定パラメーターで、インストールに対してこの値を再定義できます。詳しくは、[CICS のシステム初期設定パラメーターの指定](#)を参照してください。

## 終了

ファイルの終わりで順次入力終了されることはありません。**CESF GOODNIGHT** を最後のトランザクションとして使用し、装置を終了して装置からの読み取りを停止します。

そうでない場合、CICS は端末エラー・プログラム (DFHTEP) を起動して、順次装置のファイルの終わりで [319 ページの表 31](#) のメッセージを発行します。

表 31. 順次端末が終了していない場合の警告メッセージ	
メッセージ	宛先
DFHTC2507 <i>date time applid</i> Input event rejected return code zz {on line w/term at term}termid {, trans}trandid{, rel line=}rr,time	CSMT
DFHTC2500 <i>date time applid</i> {Line CU Terminal} out of service {Term W/Term} termid	CSMT

**CESF GOODNIGHT** を使用すると、順次装置を RECEIVE 状況にして、装置からの読み取りを終了します。ただし、こうした方法で入力装置をクローズすると、CICS シャットダウン時のウォーム・キーポイントに受信専用状況が記録されます。つまり、その端末は後続のウォーム・スタートにおいても依然として RECEIVE 状況のままで、CICS は入力ファイルを読み込みません。

また、**CESF LOGOFF** を使用して装置を終了し、装置からの読み取りを終了できますが、CICS は引き続き DFHTEP を起動して ファイルの終わりにメッセージ DFHTC2507 および DFHTC2500 を発行します。ただし、装置は TTI 状況のままで、CICS をウォーム・スタートで再始動するときに利用できます。

ウォーム・スタート中、またはウォーム・スタートの後で CICS を順次入力データ・セットから読み取る場合は、次の方法のいずれかを選択できます。

- **CESF LOGOFF** で入力を閉じ、その結果表示されるメッセージを無視します。これにより端末は TTI 状況のままで、CICS は次の起動時に入力を自動的に読み取ります。
- 入力を閉じずに、その結果表示されるメッセージを無視します。これにより端末は TRANSCEIVE 状況のままで、CICS は次の起動時に入力を自動的に読み取ります。
- **CESF GOODNIGHT** を使用して入力を閉じます。これにより、順次端末を RECEIVE 状況にして、端末からの読み取りを終了します。この場合は、PLT プログラムをコーディングして、端末の状況を TRANSCEIVE に変更することをお勧めします。
- プログラム・リスト・テーブル (PLT) から起動されるユーザー・プログラムをコーディングして、入力の処理に必要とされる順次装置ごとに適切な **EXEC CICS INQUIRE** コマンドと **EXEC CICS SET** コマンドを発行します。例えば、次のステートメントを使用して、順次端末の状況を設定します。

```
EXEC CICS INQUIRE TERMINAL(termid) SERVSTATUS(cvda) TTISTATUS(cvda)
```

SERVSTATUS が DFHVALUE(INSERVICE) を返し、TTISTATUS が DFHVALUE(NOTTI) を返す場合の各端末に対して、次のステートメントを使用して端末を TRANSCEIVE に設定します。

```
EXEC CICS SET TERMINAL(termid) TTI
```

**EXEC CICS INQUIRE** コマンドと **EXEC CICS SET** コマンドの使用に関するプログラミング情報については、システム・コマンドを参照してください。初期化段階後のプログラムの作成に関するプログラミング情報については、[初期設定プログラムの作成](#)を参照してください。

BSAM 装置をテスト目的で使用する場合、CICS を閉じる最終トランザクションは **CEMT PERFORM SHUT** になる可能性があります。CICS のシャットダウンで受信専用状況がウォーム・キーポイントに記録されます。つまり、CICS の後続のウォーム・スタートで、その端末は依然として RECEIVE 状況のままで、CICS は入力ファイルを読み込みません。

**CEMT PERFORM SHUT** を使用する場合は、CICS 領域のコールド・スタートまたは初期始動を実行して、入力ファイルを読み取ります。これは、リソースのリカバリー可能性がユーザーにとって重要でないことを想定しています。

## コンソール・デバイスの定義

CICS をコンソール・デバイスから操作できます。コンソール・デバイスには、ローカル接続のシステム・コンソール、コンソールとして定義された TSO ユーザー、または NetView® などの自動化プロセスなどがあります。

端末はシステム・コンソール、および CICS 端末の両方として使用できます。これを使用可能にするには、端末を CSD でコンソールとして定義する必要があります。(コンソールを TCT で定義することはできません。)

適切な権限がある TSO ユーザーは、MODIFY コマンドを TSO に接続された端末から入力できます。これを使用可能にするには、TSO ユーザーを CSD でコンソール・デバイスとして定義します。

各コンソール・デバイスを通常のオペレーティング・システムの機能に対して使用し、CICS トランザクションを起動することができます。特に、CICS マスター端末機能にコンソール装置を使用すれば、CICS 端末を制御したり、複数領域操作と併用していくつかの CICS 領域を制御することができます。結果として、いくつかの CICS 領域のマスター端末オペレーターになることができます。

### CICS へのコンソール装置の定義

DFHCSDUP ユーティリティの **DEFINE TERMINAL** コマンドまたは RDO を使用した **CEDA DEFINE TERMINAL** コマンドのいずれかを使って CICS に対してコンソール装置を定義できます。

コンソールはそれぞれ明示的に定義できますが、自動インストール・モデル定義を定義して、コンソールの CICS 端末の自動インストール機能を利用し、コンソールを自動的にインストールすることもできます。

コンソールの自動インストール機能を使用する場合は、AICONS=YES|AUTO をシステム初期設定パラメーターとして指定し、AUTINSTMODEL(YES) および AUTINSTNAME 属性を指定する TERMINAL モデル定義を定義します。

### システム・コンソール

システム・コンソールは、CONSOLnn メンバーの SYS1.PARMLIB ライブラリーにある MVS に定義され、NAME、UNIT、SYSTEM などの属性を定義します。

名前は、CICS がコンソール識別のために使用するもので、最も重要な属性となります。名前は MVS MODIFY コマンドで CICS に受け渡されます。コンソールも数字の ID を持つため、これは IPL 中に MVS によって動的に割り振られ、この利用はコンソールを CICS に定義するためには推奨されていません。

コンソール・デバイスを MVS に定義する方法については、[z/OS MVS 初期設定およびチューニング解説書](#)を参照してください。

MVS コンソールを CICS に定義する方法の詳細については、[321 ページの『CICS に MVS コンソールを定義する』](#)を参照してください。

### コンソールとしての TSO ユーザー

コマンドを CICS に発行する TSO ユーザーは、TSO **CONSOLE** コマンドまたは SDSF のいずれかを使用して、CONSOLnn メンバーでコンソールとして MVS 定義を要求しません。MVS はコンソール名としてユーザーの TSO/E ユーザー ID を使用してコンソールを自動的にアクティブにします。

**CONSOLE** コマンドを発行する TSO ユーザーは、NAME オプションを使用して、TSO ユーザー ID とは異なるコンソール名を指定できます。

CICS 領域を TSO または SDSF から通信するには、TSO ユーザー ID (またはコンソール・コマンドで指定された名前) をコンソール名として指定する CICS コンソール定義をインストールする必要があります。

TSO **CONSOLE** コマンドについて詳しくは、[z/OS TSO/E システム・プログラミング コマンド解説書](#)を参照してください。

TSO ユーザーを CICS に定義する方法について詳しくは、[321 ページの『TSO ユーザーをコンソール・デバイスとして定義』](#)を参照してください。

### CICS に MVS コンソールを定義する

MVS コンソールを CICS のマスター端末として使用するには、CSD の端末定義項目によって明示的に MVS コンソールを CICS に定義するか、CICS コンソール自動インストール機能を使用します。

定義する各コンソールは、TERMINAL 定義の上で CONSNAME(*name*) 属性によって識別されます。CICS は CONSOLE(*number*) 属性をサポートしなくなりました。MVS に接続されるコンソール・デバイスの sysplex 内での識別は、CONSNAME 属性を使用して、コンソールの名前によって行われます。

コンソールを定義するのに必要な DEFINE コマンドの例については、[321 ページの図 55](#) を参照してください。

```
//DEFTERM JOB (accounting information),MSGCLASS=A,
// MSGLEVEL=(1,1),CLASS=A,NOTIFY=userid
//CONSDEF EXEC PGM=DFHCSDUP
//STEPLIB DD DSN=CICSTS56.CICS.SDFHLOAD,DISP=SHR
//DFHCSD DD DSN=CICSTS56.CICS.DFHCSD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
*
* Define a console for CICS
DEFINE TERMINAL(trmidnt) GROUP(grpname) TYPETERM(DFHCONS)
        CONSNAME(consname) DESCRIPTION(MVS CONSOLE consname)
*
* Define a TSO user as a console device for CICS
DEFINE TERMINAL(trmidnt) GROUP(grpname) TYPETERM(DFHCONS)
        CONSNAME(tsouser) DESCRIPTION(TSO USER tsouser)
        USERID(tsouser)
*
* Define an AUTOINSTALL model definition for a console device
DEFINE TERMINAL(autc) GROUP(grpname) TYPETERM(DFHCONS)
        CONSNAME(console) DESCRIPTION(Autoinstall model for a console)
        USERID(*FIRST) AUTINSTNAME(name) AUTINSTMODEL(YES)
*
ADD GROUP(grpname) LIST(yourlist)
*
LIST LIST(yourlist) OBJECTS
/*
//
```

図 55. DFHCSDUP を使用して CSD でコンソールおよび TSO ユーザーを定義する

### TSO ユーザーをコンソール・デバイスとして定義

TSO ユーザーごとに明示的に定義された TERMINAL 定義を使用するか、コンソールの自動インストール機能を使用して、TSO ユーザーをコンソール・デバイスとして CICS に定義できます。

TSO ユーザーをコンソールとして定義するには、DEFINE TERMINAL コマンドの CONSNAME 属性で TSO ユーザーが使用するコンソール名を指定します。デフォルトで、コンソール名はユーザーの TSO ユーザー ID です。USERID オペランドを使用して、事前設定セキュリティがある CICS にコンソールを定義し、TSO ユーザーが CESN トランザクションを使用してサインオンしなくても済むようにすることを推奨します。そうしない場合は、CESN トランザクションでの入力で、TSO ユーザーの CICS サインオン・パスワードが表示されます。

TSO ユーザーを定義するために必要な DEFINE コマンドの例は、[321 ページの図 55](#) を参照してください。

事前設定セキュリティがあるコンソールの定義について詳しくは、[Security facilities in CICS](#) を参照してください。

注：321 ページの図 55 に示された DEFTERM ジョブの中でイタリックで表されているオペランドについては、ユーザー独自の値に置き換える必要があります。

#### AUTINSTMODEL(YES)

この TERMINAL 定義を自動インストールの目的でモデルとして使用できるかどうかを指定します。

#### AUTINSTNAME(*name*)

この自動インストール・モデル定義に割り当てられた名前、およびモデルが自動インストール制御プログラムに認識される名前。

#### CONSNAME(*consname*)

固有の 8 文字のコンソール名。コンソールを定義する、または TSO ユーザーが使用するコンソール名に一致する CONSOLnn PARMLIB メンバーの NAME パラメーターに対応します。



自動インストール・モデルに対しても CONSNAME を定義する必要があります。

#### **GROUP(*grpname*)**

コンソール・リソース定義が属するグループの固有の名前。

#### **LIST(*yourlist*)**

コンソール定義を定義したグループを含む始動グループ・リスト。新しいグループ・リストに、ユーザー独自のリソースだけでなく、必要な CICS 提供リソースも含まれていない場合には、CICS 始動ジョブの GRPLIST システム初期設定パラメーターに、DFHLIST と *yourlist* を指定しなければなりません。

#### **TERMINAL(*trmidnt/autc*)**

CICS が TCT 端末項目 (TCTTE) のコンソールを識別する名前としての、固有の 4 文字の端末 ID *trmidnt*、または自動インストール・モデル定義の場合はダミー名 *autc*。

#### **USERID(*tsouser*)**

このコンソール・デバイスをサインオンするために使用される CICS の事前設定セキュリティのユーザー ID。

CSD で CONSNAME (INTERNAL) としてコンソール・デバイスを定義している場合は、それを利用して MVS ジョブ制御言語を使用し、コマンドを発行することができます。また、MGCR マクロを使用する許可されたプログラムにより、MVS コマンドを発行するためにも使用されます。

CSD でコンソール・デバイスを定義して、リソース定義が実行中の CICS 領域にインストールされていることを確認します。次のように、2 つの方法のいずれかで定義をインストールできます。

1. CICS の始動ジョブの **GRPLIST** システム初期設定パラメーターのリソース定義を含むグループ・リストが含まれます。
2. CICS の実行中に、RDO コマンドの CEDA INSTALL GROUP(*groupname*) を使用してコンソール・デバイス・グループをインストールします。ここで、*groupname* はコンソール・デバイス定義を含むリソース・グループの名前です。

DFHLIST。DFHCSDUP INITIALIZE コマンドを使用して CSD を初期化する場合に作成される CICS 定義のグループ・リスト。コンソール・デバイスのリソース定義は含まれません。ただし、CSD はコンソール定義を含む 2 つのグループで初期化されます。

#### **DFH\$CNLSL**

このグループには、3 つのコンソールの定義が含まれます。グループはインストール検査手順と CICS 提供のサンプル・プログラムで使用するよう想定されています。これを独自のグループ・リストに追加して、独自のコンソール・デバイスを定義するよう定義を変更することができます。

#### **DFHTERM**

このグループには、MVS コンソールの自動インストール・モデル定義の単一の定義が含まれます。

コンソール・デバイスに新しい端末定義を作成することを決定した場合は、CICS 提供の TYPETERM 定義の DFHCONS を TYPETERM(*name*) パラメーターで指定できます。コンソール・デバイスのこの TYPETERM 定義は、CSD を初期化するときにグループ DFHTYPE で生成されます。

## **z/OS Communications Server 持続セッション・サポートの定義**

z/OS Communications Server 持続セッション・サポートを使用して CICS 領域を実行するよう選択した場合、CICS または z/OS Communications Server に障害が発生したときに、端末のユーザーが、セッションをリカバリーして作業を続行することができます。

### **このタスクについて**

CICS システム初期設定パラメーター **PSTYPE** および **PSDINT** を使用して、CICS 領域の持続セッション・サポートをセットアップし、CONNECTION、TYPETERM、および SESSIONS リソース定義のオプションを使用して、リカバリー時の端末ユーザーのユーザー・エクスペリエンスをカスタマイズします。

**PSTYPE** および **PSDINT** システム初期設定パラメーターのデフォルト設定は、CICS 領域で持続セッション・サポートが利用可能であるが、活用されていないことを意味します。z/OS Communications Server 持続セッションを使用したリカバリーでは、持続セッション・サポートを利用したときに起こること、および持続セッション・サポートなしで CICS 領域を実行した方がよい理由について説明しています。

TYPETERM 定義の **RECOVOPTION** および **RECOVNOTIFY** のデフォルト値は、それぞれ **SYSDEFAULT** と **NONE** ですが、持続セッションでこの組み合わせで使用すると、状況によってはリカバリーされた端末が使用できなくなります。この状態は、**EXEC CICS START TRANSID(xxx) TERMID(yyy)** を使用するなど、端末でタスクが自動的に開始されるまで続くことがあります。RECOVNOTIFY(NONE) の利用は、プリンターなどのデバイスでは適していますが、ユーザーの端末では別の値を検討する必要があります。

既存の CICS 領域に対する持続セッション・サポートのタイプを変更する場合は、領域をシャットダウンして新しい設定でコールド・スタートを実行する必要があります。CICS の実行中は、単一ノードの持続セッション・サポート、マルチノードの持続セッション・サポート、および持続セッション・サポートなしの間で変更することはできません。セッションが作成されたときにバインドされたセッションの持続性の特性が確立されるため、アンバインドを強制してセッションを再バインドし、CICS によって要求される持続セッション・サポートの新しいレベルに対して正しい持続性の特性を取得する必要があります。これを達成する最も単純な方法は、CICS 領域のコールド・スタートまたは初期始動を実行することです。

CICS の実行中に持続セッション遅延間隔を変更できますが、変更された間隔は CICS グローバル・カタログに格納されません。このため、緊急時再始動時に復元されません。

持続セッション・サポートを CICS 領域に対して使用する場合は、以下のステップに従ってください。

## 手順

1. **PSTYPE** システム初期設定パラメーターを使用して、持続セッション・サポートの適切なレベルを指定します。
  - a) カップリング・ファシリティを使用した Parallel Sysplex に z/OS Communications Server V4R4 以降がある場合は、マルチノード持続セッション・サポートに対して **MNPS** を指定します。  
このレベルのサポートにより、z/OS Communications Server、z/OS、または CICS で障害が発生した場合にリカバリーが可能になります。
  - b) マルチノード持続セッション・サポートに適切なファシリティがない場合は、単一ノード持続セッション・サポートに対して **SNPS** を指定します。  
このレベルのサポートにより、CICS に障害が発生した場合にリカバリーが可能になりますが、z/OS Communications Server または z/OS に障害が発生した場合にはリカバリーできません。
2. **PSDINT** システム初期設定パラメーターを使用して、適切な持続セッション遅延間隔を指定します。  
この値は、z/OS Communications Server がセッションをリカバリー・ペンディング状態に保つ長さを決定します。指定する間隔は、CICS に障害が発生してから、それに続く緊急時再始動中に z/OS Communications Server ACB が CICS により開かれるまでの時間をカバーする必要があります。
3. CICS 領域で使用される **SESSIONS** および TYPETERM リソース定義の **RECOVOPTION** オプションに対して、適切な値を選択して設定します。  
通常、この値はデフォルトの **SYSDEFAULT** にすることができ、これにより CICS がセッションのリカバリーで最適なプロシージャを選択できます。この設定は、リカバリーのときに端末のユーザーが画面をクリアし、CICS の transid の入力を続行できることを意味します。
4. CICS 領域で使用される TYPETERM リソース定義の **RECOVNOTIFY** オプションに対して適切な値を選択して設定します。
  - a) 正常なリカバリーが端末のユーザーに対して透過的に行われるようにするには、**NONE** を指定します。
  - b) システムがリカバリーしたことを示すメッセージを画面に表示する場合は、**MESSAGE** を指定します。
  - c) 端末で good-morning トランザクションなどのトランザクションを開始する場合は、**TRANSACTION** を指定します。
5. 既存の CICS 領域で作業している場合は、CICS 領域のコールド・スタートを実行して、持続セッション・サポートへの変更を実装します。



## リソース定義のインストール

リソース定義がインストールされている場合、リソースの情報は、CICS アドレス・スペース内のリソースを表すデータ構造を構成するのに使用されます。

### CICS が初期化されるときに発生すること

CICS を初期化するとき、リソース定義で発生することは、開始のタイプに応じて異なります。これは START システム 初期設定パラメーターで定義され、初期始動の場合は START=INITIAL、コールド・スタートの場合は START=COLD、ウォーム・スタートまたは緊急時再始動の場合は START=AUTO になります。

#### 初期始動またはコールド・スタート

初期始動またはコールド・スタート中に、CICS は GRPLIST システム 初期設定パラメーターによって指定されたリストに記載されたグループをインストールすることで、システム・テーブルを作成します。

前の CICS 実行中に INSTALL コマンドを使用してグループをインストールした場合、コールド・スタート中にインストールする場合は、グループの名前をリストに追加する必要があります。

通常、START=COLD を CICS の初期化時に使用している場合は、リストを使用してインストールするのが、リソース定義を CICS に対して使用可能にする標準的な方法です。INSTALL コマンドの使用は補足的な方法であり、システムをテストするときや、CICS の実行中にリソースの予期しないニーズが発生した場合に非常に有用になる可能性があります。

セキュリティやパフォーマンス上の理由から、実動システムでは RDO トランザクションを使用しないでください。この場合は、CSD ファイルが両方のシステムによって共用されますが、実動システムでは読み取り専用です。開発システムを使用して実動システムのリソースをすべて定義して、コールド・スタートを行うときに、実動 CICS システムにインストールします。

#### ウォーム・スタートまたは緊急始動

ウォーム・スタートまたは緊急始動中に、CICS はシステム・ログとグローバル・カタログに保管されるリソース定義からテーブルを再作成します。

CSD ファイルへの参照は行われず、GRPLIST 名も使用されません。このため、以前の CICS 実行の終わりにインストールされたすべてのグループは、ウォーム・リスタートまたは緊急時再始動で自動的に再インストールされます。このため、RDO を使用して導入された CICS システム 修正変更は持続します。自動インストールされたリソースについては、以下を参照してください。

- [253 ページの『CICS の再始動で発生すること』](#)
- [263 ページの『接続の自動インストールにおけるリカバリーと再始動』](#)
- [267 ページの『プログラムの自動インストール、リカバリー、および再始動』](#)

GRPLIST オペランドで別のリストを指定した場合、または最後のシステムの初期化の後で新しいグループを追加した場合、CICS は新しいリストを参照しないため、ウォーム・リスタートまたは緊急時再始動中に新しいリストのグループをインストールしません。

通常、START=AUTO を CICS の初期化時に使用している場合は、INSTALL コマンドを使用してリソース定義を CICS に対して使用可能にするのが標準的な方法です。初期始動またはコールド・スタートを実行する必要がある場合にのみ、リストを使用してシステムを定義します。INSTALL コマンドを使用してインストールされた各グループをリストに追加することで、リストを最新の状態に保つことができます。

### INSTALL コマンドを使用したときの動作

ほとんどのリソース定義は、グループとしてまたは個別にインストールすることが可能で、それらは個別リソース・レベルでコミットされます。ただし、一部の SNA LU 制御リソース定義は、グループとしてインストールする必要があり、インストール可能セットでコミットされます。

CSD グループ内にある一部の SNA LU 制御リソース定義は、インストール可能セット・レベルでコミットされます。インストール可能セットは、CONNECTION やそれに関連した SESSIONS などのリソースで構成され、ある意味で依存関係にあります。インストール可能セットがバックアウトしても、グループ全体がバックアウトすることはありません。以下のタイプのリソース定義が、インストール可能セットでインストールされます。

- CONNECTION 定義および関連した SESSIONS 定義

- パイプライン端末 - 同じ POOL 名を共有するすべての端末定義

あるインストール可能グループに属する 1 つのメンバーがインストールに失敗した場合、CICS はメッセージ DFHZC6216 を発行して、そのセットのインストールが失敗した原因となったメンバーを特定します。

他のすべてのリソース・タイプは、インストール可能セットとしてではなく個別にコミットされます。これらのリソースで、グループ INSTALL が一部だけ成功した場合、追加されたリソースはコミット済みの状態になります。

インストール可能セットと個別の定義をインストールする場合、以下の 2 つの理由のいずれかにより、INSTALL が成功しないことがあります。

1. リソース定義が現在使用中でインストールできなかった。
2. インストール中にシステム障害が発生した。

CEDA INSTALL コマンドを使用して、使用されたものと同じグループをコールド・スタートまたは初期スタート時に再インストールできます。これにより、元の定義のいくつかが使用中でインストールに失敗した場合でも、新しい定義が正常にインストールされます。

## 限られた数のデータ定義をインストールする方法

少数の新しいまたは変更された定義だけをインストールする場合は、単一リソースをインストールしてください。単一リソース INSTALL を使用すると、障害により部分的 INSTALL が生じる問題をなくすことができます。

一部の CONNECTION および SESSIONS では、単一リソース INSTALL を実行できないことに注意してください。

しかし、多数の定義を変更または追加する場合には、新しいグループをインストールすることもできます。その場合には、以下の考慮事項があります。

- 既存のリソース定義を更新したグループをインストールするとき、その時点でそのリソースが使用中であれば、インストールは失敗します。グループのインストールを試行する前に、グループ内に使用中のリソースがないことを確認してください。
- インストールは、2 つのステージからなるプロセスです。定義をインストールするには、その前にそのリソースの既存の定義をシステム・テーブルから削除する必要があります。「削除」が失敗してインストールが失敗した場合、複数のメッセージが出されることがあります。
- 同じ CSD ファイルを共有する複数の CICS システムがある場合、間違ったシステム内にリソースのグループをインストールしないように注意する必要があります。

## 重複するリソース定義名

TDQUEUE および FILE リソースを除くすべてのリソース・タイプでは、リストの 2 つのグループに名前とリソース・タイプが同じリソース定義が含まれる場合、CICS はリストの後にある方のグループの定義を使用します。

- TDQUEUE リソース定義では、リストの最初の定義が使用されます。
- ローカル・ファイルの FILE リソース定義では、ファイルが ENABLED として定義される場合、重複する後の方のインストールは失敗します。ただし、ファイルが DISABLED として定義される場合、重複する後の方のインストールが成功します。ENABLED または DISABLED と定義されていないリモート・ファイルの FILE リソース定義では、重複しているインストールが常に成功します。

CICS がアクティブな間にグループをインストールする場合、グループのリソース定義はすでにインストールされている同じタイプと名前をオーバーライドします。既存のリソース定義がこの方法で置き換えられる場合、古いリソース定義に関連付けられた統計は、新しい定義に転送されます。PROGRAM 定義が置き換えられる場合、新しい定義が最初に参照されるときに、プログラムがライブラリーに再配置されてロードされます。実際には、新しい定義は NEWCOPY 操作を意味します。同じルールがマップ・セットと区画セットに適用されます。

リソース・タイプが異なっても、名前が同じリソース定義が CSD ファイルに複数あることは推奨されません。PROGRAM、MAPSET、および PARTITIONSET の名前は固有にする必要があります。例えば、同じ名前の PROGRAM と MAPSET がある場合は、CICS で利用できるのはそれらの 1 つだけです。名前に関

する限り、インストール後、これらの定義は同じリソース・タイプとして処理されます。実リソースが同じで属性が異なる代替定義が必要な場合は、これらのリソース定義を別のグループにする必要があります。

RDO 定義の定義は、同じ名前のマクロ定義の定義をオーバーライドします。例えば、非 SNA LU と同じ名前を持つ SNA LU の定義をインストールしようとする、SNA LU 項目は非 SNA LU 項目を上書きします。

CICS バンドル内で定義されているインストール済みリソースは、リソース定義によって同じ名前では上書きできません。リソースが含まれているバンドル・パーツがインストールされている場合に別の CICS バンドルを使用するか、RDO の EXEC CICS CREATE コマンドまたは他のメソッドを使用して同じリソースをインストールしようとする、要求は拒否されます。CICS バンドル内で定義されている同じリソースでインストール済みリソースを上書きすることもできません。CICS バンドルをインストールしようとしたときに同じ名前とリソース・タイプのリソースが存在する場合、そのリソースを含むバンドルはインストールされず、CICS バンドルは使用不可状態になります。そのため、CICS バンドル内で定義されているリソースは、同じリソース名によって誤って上書きされることはありません。

専用リソースが、アプリケーションの一部としてパッケージされてインストールされる CICS バンドルで定義される場合、インストール済み環境内でリソース名を固有にする必要はありません。この機能を使用すると、別々に開発されたアプリケーション間で同じリソース名を使用しているためにリソース名が競合することを回避できます。サポートされる CICS リソースについてリソース名を固有にするという要件は、プラットフォームにデプロイされるアプリケーションの一部としてリソースを管理することにより除去されます。このプロセスは、サーバーの統合を支援するために使用できます。

アプリケーションの専用リソースについて詳しくは、[アプリケーション・バージョンの専用リソース](#)を参照してください。

リソース・シグニチャーを使用して、既存のインストール済みリソースを置き換えた、重複するリソース定義のソースを識別します。リソース・シグニチャーには、各リソースが定義、変更、およびインストールされた時期、方法、およびそれらの実行者が示されます。リソース・シグニチャーは、CEDA および CEMT パネル、CICSplex SM BAS および操作ビュー、EXEC CICS INQUIRE コマンド、および DFHCSDUP の CICS Explorer ビューに表示されます。

## ATOMSERVICE リソース定義のインストール

このプロシージャでは、CEMT トランザクションと CEDA トランザクションを使用して、ATOMSERVICE リソース定義をインストールします。ATOMSERVICE リソースが既に存在する場合は、再インストールの前にそのリソースを無効にする必要があります。

### 手順

1. ATOMSERVICE リソースが既に存在する場合は、そのリソースが無効になっていることを確認してください。

以下のコマンドを使用します。

```
CEMT SET ATOMSERVICE(name) DISABLED
```

ATOMSERVICE リソースが無効になっていると、Web クライアントがそのリソースを必要とする HTTP 要求を送信したときに、CICS から Web クライアントに HTTP 503 応答 (サービス無効) が返されます。

2. ATOMSERVICE 定義をインストールします。

以下のコマンドを使用します。

```
CEDA INSTALL GROUP(groupname) ATOMSERVICE(name)
```

3. オプション: ATOMSERVICE 定義のインストールが成功したら、CEMT を使用してそのリソースを有効にします。

この手順を実行するのは、ATOMSERVICE リソースがまだ ENABLED として定義されていない状態でそのリソースを Web クライアントに対して有効する場合に限られます。以下のコマンドを使用します。

```
CEMT SET ATOMSERVICE(name) ENABLED
```

## 接続定義のインストール

新規の CONNECTION 定義をインストールするには、既にインストール済みの CONNECTION 定義を含まない専用のグループに新しい定義を入れてから、CEDA INSTALL を使用してグループ全体をインストールします。単一の CONNECTION 定義をインストールすることはできません。

### このタスクについて

既存の MRO CONNECTION 定義を変更および再インストールする場合、あるいは新規および既存の MRO CONNECTION 定義を同じグループに入れる場合、その定義を使用できるようにするには、事前にすべての領域間通信 (IRC) を閉じてから開き直す必要があります。IRC を閉じてから開き直すことをせずに、新しい接続でその領域を立ち上げようとすると、メッセージ DFHIR3788 を受け取ります。

### 手順

1. IRC を閉じます。  
以下のコマンドを使用します。  
`CEMT SET IRC CLOSED`
2. リソース定義をインストールします。  
以下のコマンドを使用します。  
`CEDA INSTALL GROUP(groupname)`
3. 定義が含まれるグループのインストールが正常に完了したら、IRC を再度開きます。  
以下のコマンドを使用します。  
`CEMT SET IRC OPEN`

## Db2 接続定義のインストール

このセクションでは、DB2CONN 定義のインストールと削除についての指針、および活動の一部での割り込みの影響について説明します。

### このタスクについて

- CICS システムにインストールできる DB2CONN は、一度に 1 つのみです。2 つ目の DB2CONN をインストールすると、インストールを進める前に既存の DB2CONN とそれに関連した DB2ENTRY および DB2TRAN が暗黙的に削除されます（同じ名前の DB2CONN を再インストールする場合は除く）。
- DB2CONN は、DB2ENTRY または DB2TRAN 定義の前にインストールする必要があります。DB2ENTRY および DB2TRAN 定義は単独で存在することはできず、すでにインストール済みの DB2CONN に関連付けることができます。また、DB2CONN を削除すると、それに関連した DB2ENTRY と DB2TRAN の資源定義も一緒に削除されます。自身が属する DB2CONN を明示的に指定する DB2ENTRY または DB2TRAN には属性がないことに注意してください。これにより、DB2CONN 定義が DB2ENTRY および DB2TRAN 定義を変更なしに共用することができます。

**注：**DB2CONN、DB2ENTRY、および DB2TRAN が同じグループで定義されている場合、CICS は自動的に DB2CONN を最初にインストールします。Db2 定義を複数のグループから（リストまたは複数の INSTALL GROUP コマンドを使用して）インストールする場合、インストールする最初のグループに DB2CONN 定義が含まれている必要があります。その後のグループには、DB2CONN 定義が含まれないようにします。DB2CONN がインストールされていないときに、DB2ENTRY や DB2TRAN をインストールすると、CICS がエラー・メッセージを出します。複数の DB2CONN 定義がインストールされた場合、最後の DB2CONN 定義以前にインストールされた Db2 定義はすべて削除されます。CICS は、すべての削除に対してメッセージを出します。

- DB2CONN は、CICS Db2 接続を開始する前にインストールする必要があります。DB2CONN には、プール・スレッドとコマンド・スレッド、およびグローバル・タイプ情報が含まれるため、CICS Db2 接続を開始するために必要な最低条件を表します。項目スレッドは存在せず、すべてのトランザクションがプールを使用します。CICS Db2 接続がアクティブになった後、DB2ENTRY および DB2TRAN を追加することができます。

- DB2CONN を再インストールすることができるのは、CICS Db2 接続機能が Db2 に接続されていないために非アクティブである場合のみです。
- DB2CONN を破棄できるのは、CICS Db2 接続機能が Db2 に接続されていない場合のみです。
- DB2CONN を削除すると、インストール済みのすべての DB2ENTRY と DB2TRAN が暗黙的に削除されます。

注: DB2CONN、DB2ENTRY、および DB2TRAN のグループ・コミットおよびグループ破棄はありません。ただし、DB2CONN が削除されると、基本となる制御ブロックに削除が進行中であることを示すマークが付きます。DB2ENTRY および DB2TRAN は、DB2CONN の前に削除されます。削除が半ば完了した時点で障害が起こった場合、DB2CONN の制御ブロックが生成され、削除が進行中であることを示すメッセージが出されます。一部破棄された Db2 リソース定義を使用している場合、CICS Db2 接続機能の開始は失敗し、以下のメッセージが発行されます。

```
DFHDB2074 CICS DB2 ATTACHMENT FACILITY STARTUP
          CANNOT PROCEED AS THE CURRENTLY
          INSTALLED DB2CONN IS NOT USABLE
```

削除を再発行する必要があります。削除時の障害の後に CICS システムを再始動したとき、CICS システムは削除が行われたことを認識しています。CICS は、カタログからブロックを回復しないため、削除が効率的に完了します。（同様に、定義もカタログから除去されることに注意してください。）

インストール時に、グループのインストール、またはリストのインストールの一部が失敗する場合がありますが、どの資源に障害があるかを示すメッセージが表示されます。これが発生すると、CICS Db2 接続機能の開始に進むことができます。

## Db2 接続リソースの定義の検査

DB2CONN を定義すると、CICS は、同じグループあるいはリスト内の他のリソース定義との整合性を検査します。

DB2CONN オブジェクトでは、以下の検査が行われます。

- グループまたはリストに DB2CONN が 1 つだけ定義されていることの確認。複数の定義がある場合（名前が異なっていたとしても）警告メッセージが出されます。一度にインストールできる DB2CONN は 1 つのみです。
- PLANEXITNAME が指定されている場合、PLANEXITNAME がグループまたはリスト内のプログラム定義として存在すること、およびプログラム自動インストールがアクティブでないことの確認。

## Db2 エントリー定義のインストール

このセクションでは、DB2ENTRY 定義のインストールと削除についての指針、および活動の一部への割り込みの影響について説明します。

### このタスクについて

- すでに DB2CONN をインストールしている場合にのみ、DB2ENTRY をインストールすることができます。
- CICS Db2 アダプターが Db2 に接続されていても、いつでも新規の DB2ENTRY をインストールできます。
- DB2ENTRY が使用不可の状態、これを使用するトランザクションがない場合にのみ、再インストール（既存の DB2ENTRY を置き換えることによって）できます。活動を静止したり、エントリーを使用不可にしたりするには、**SET DB2ENTRY DISABLED** コマンドを使用します。DB2ENTRY が使用不可の場合、DB2ENTRY を使用しようとする新規トランザクションは、DISABLEDACT キーワードの設定に応じて、プールに送られるか、異常終了するか、または SQLCODE が戻されます。
- DB2ENTRY が使用不可の場合にのみ、DB2ENTRY を削除できます。活動を静止したり、エントリーを使用不可にしたりするには、**SET DB2ENTRY DISABLED** コマンドを使用します。DB2ENTRY が使用不可の場合、DB2ENTRY を使用しようとする新規トランザクションは、DISABLEDACT キーワードの設定に応じて、プールに送られるか、異常終了するか、または SQLCODE が戻されます。

DB2ENTRY を削除する場合、対応する DB2TRAN を「オーファン」にすることができます。その後、トランザクションを実行すると、CDB2 一時データ宛先にメッセージが送信され、要求がプールに転送されます。

## Db2 エントリー・リソースの定義の検査

DB2ENTRY を定義すると、CICS は、同じグループあるいはリスト内の他のリソース定義との整合性を検査します。

DB2ENTRY オブジェクトでは、以下の検査が行われます。

- 1つのリストに同じ名前の2つの DB2ENTRY が表示されていないかどうか。表示されている場合、警告メッセージが出されます。
- グループまたはリストに DB2CONN が存在するかどうか。存在しない場合、警告メッセージが出されます。DB2CONN は別のグループの他の場所に存在できるため、エラーでない場合がありますが、DB2ENTRY をインストールするためには、DB2CONN がインストールされていることが必要です。
- DB2ENTRY に TRANSID が指定されている場合は、グループまたはリスト内の TRANSID に対して、トランザクション資源定義が存在するかどうか。存在しない場合、警告メッセージが出されます。
- PLANEXITNAME が指定され、プログラム自動インストールがアクティブでない場合、グループまたはリストにプログラム定義として PLANEXITNAME が存在するかどうか。

## Db2 トランザクション定義のインストール

このセクションでは、DB2TRAN 定義のインストールと破棄のガイドライン、および部分活動での中断の影響について説明します。

### このタスクについて

- DB2TRAN 定義で完全なトランザクション ID を指定した場合、同一のトランザクション用に複数の DB2TRAN をインストールできません。ワイルドカード文字を使用した汎用トランザクション ID を使用していた場合、そのトランザクションに潜在的に一致する複数の DB2TRAN をインストールできますが、CICS は、最も近い一致のみを使用します ([トランザクション ID のワイルドカード文字を参照](#))。
- 存在しない DB2ENTRY を参照する DB2TRAN はインストールできません。DB2ENTRY を先にインストールする必要があります。
- DB2ENTRY 定義と DB2TRAN 定義が同じグループで定義されている場合、CICS はインストール時に DB2ENTRY を先にインストールします。
- CICS Db2 アダプターが Db2 に接続されていても、いつでも新規の DB2TRAN をインストールできます。
- DB2TRAN はいつでも再インストールでき、いつでも破棄できます。

## Db2 トランザクション・リソースの定義の検査

DB2TRAN を定義すると、CICS は、同じグループあるいはリスト内の他のリソース定義との整合性を検査します。

DB2TRAN オブジェクトに対して、以下の検査が行われます。

- 同じ名前の DB2TRAN が同じリストに含まれていないこと。含まれている場合、警告メッセージが出されます。
- DB2CONN がグループまたはリストに存在するかどうか。存在しない場合、警告メッセージが出されます。DB2CONN が別のグループに存在する可能性があるため、これはエラーではありませんが、DB2TRAN をインストールするには、先に DB2CONN をインストールする必要があります。
- DB2TRAN で指定された DB2ENTRY がグループまたはリストに存在するかどうか。存在しない場合、警告メッセージが出されます。
- DB2TRAN で指定された TRANSID がグループまたはリストに存在すること。存在しない場合、警告メッセージが出されます。



## エンキュー・モデル定義のインストール

通常、ENQMODEL のデフォルト状況は *ENABLED* であるため、ENQMODEL のインストール順序は、ENABLING ENQMODEL の規則に従う必要があります。

### このタスクについて

ネストされた汎用エンキュー名を形成する ENQMODEL を有効にする場合、最も具体性の高いものから具体性の低いものへと順番に行う必要があります。

例えば、ABCD\*、ABC\*、および AB\* というエンキュー名を持つ ENQMODELS がある場合、以下の順序で有効にしてください。

1. ABCD\* (具体性が最も高い)
2. ABC\*
3. AB\* (具体性が最も低い)

## ファイル定義のインストール

この手順では、CEMT トランザクションおよび CEDA トランザクションを使用して、FILE 定義をインストールします。CEMT の代わりに、ユーザー作成トランザクションで **EXEC CICS SET FILE** コマンドを使用して、ファイルを使用可能/使用不可にすることができます。

### このタスクについて

#### 手順

1. ファイルが既に存在する場合は、そのファイルが閉じられ、使用不可になっていることを確認してください。

以下のコマンドを使用します。

```
CEMT SET FILE(filename) CLOSED DISABLED
```

2. ファイル定義をインストールします。

以下のコマンドを使用します。

```
CEDA INSTALL GROUP(groupname) FILE(filename)
```

3. オプション: ファイルが含まれるグループのインストールが正常に完了したら、CEMT を使用してファイルを開き、使用可能にします。

この手順は、ファイルが *ENABLED* として定義されておらず、ファイルを明示的に開きたい場合にのみ実行してください。以下のコマンドを使用します。

```
CEMT SET FILE(filename) OPEN ENABLED
```

## IPCONN 定義のインストール

新しい IPCONN 定義をインストールするには、既にインストール済みの IPCONN 定義を含まない専用のグループに新しい IPCONN 定義を入れてから、CEDA INSTALL を使用してグループをインストールします。IPCONN 定義は、個別にインストールすることもできます。

### このタスクについて

IPCONN 定義のインストール時に接続を実現するために、以下のリソース要件を考慮してください。

1. この IPCONN 定義の TCPIP SERVICE 属性で指定された TCPIP SERVICE 定義が、この領域にもインストールされており、PROTOCOL(IPIC) を指定している必要があります。
2. 対応する IPCONN 定義と TCPIP SERVICE 定義は、リモート領域にインストールする必要があります。  
ここで、"対応する" とは、次のような意味です。
  - a. リモート領域の IPCONN 定義の HOST 属性が、この領域を指定している必要があります。

- b. リモート領域の IPCONN 定義の PORT 属性が、この IPCONN で指定されたローカル TCPIP SERVICE 定義の PORTNUMBER 属性で指定されているのと同じポート番号を指定している必要があります。
- c. リモート領域の IPCONN 定義で指定された、リモート領域の TCPIP SERVICE 定義が PROTOCOL(IPIC) を指定していること、およびその PORTNUMBER 属性で、この IPCONN の PORT 属性で指定されているのと同じポート番号が指定されている必要があります。

## CEDA を使用した LIBRARY リソース定義のインストール

新規または更新された LIBRARY リソース定義をインストールし、LIBRARY リソースが CICS 領域でアクティブになっていることを確認します。この手順では、CICS CEDA トランザクションを使用して、LIBRARY リソース定義をインストールする方法を説明します。

### このタスクについて

CEDA を使用してリソース定義をインストールする際に、ENABLED または DISABLED のいずれかの初期状況を設定でき、リソースはその状況でインストールされます。

以下の手順では、グループ *groupname* 内で定義されている LIBRARY リソース定義 *libname* をインストールし、初期リソース状況を ENABLED に設定します。

### 手順

以下のコマンドを使用して LIBRARY 定義をインストールします。

```
CEDA INSTALL LIBRARY(libname) GROUP(groupname) STATUS(ENABLED)
```

### 次のタスク

#### CICS システムに対するデータ・セットの定義

LIBRARY リソース定義のインストールが正常に完了し、LIBRARY リソース内のアプリケーションを構成するプログラム、マップ・セット、その他のエレメントもインストールしたら、CICS システムにデータ・セットを定義することができます (まだ定義していない場合)。

#### LIBRARY リソース定義の使用可能化

LIBRARY リソースが DISABLED として定義されてインストールされた場合は、**EXEC CICS SET LIBRARY** コマンドまたは **CEMT SET LIBRARY** コマンドを使用して、状況を ENABLED に変更できます。

## パートナー定義のインストール

PARTNER 定義をインストールすると、CICS は、CONNECTION 定義および PROFILE 定義に対する参照の解決を試行します。

### このタスクについて

その後、CICS は、PARTNER 定義のエントリをパートナー・リソース・テーブル (PRT) に作成します。詳しくは、[DTP のリモート・リソースの定義](#)を参照してください。

プログラムが SAA 通信インターフェース SET 呼び出しを使用して PARTNER 名および TPNAME 名を変更することが可能であるため、CICS は、PARTNER リソースのインストール時に CONNECTION 定義が存在するかを確認しません。

PROFILE 属性を省略すると、CICS はデフォルト・プロファイル DFHCICSA を使用します。SAA 通信インターフェース・プログラムが別のプロファイルを設定することはできないため、PROFILE は、PARTNER リソースの前にインストールする必要があります。ただし、PROFILE が欠落していても PARTNER のインストールは失敗しませんが、パートナー通信の試行時にランタイム・エラーが発生します。

SAA 通信インターフェース・プログラムは、MODENAME に異なる値を設定できるため、PROFILE に指定する MODENAME には、対応する SESSIONS 定義を指定する必要はありません。

SET 呼び出しを使用しない CICS プログラムおよび SAA 通信 インターフェース・プログラムでは、すべての関連定義がインストールされている必要があります。CEDA CHECK を使用して、すべての必要な定義がグループ内に存在していることを確認できます。

## セッション定義のインストール

定義がすでにインストールされている場合に INSTALL コマンドを使用して MRO の新規 SESSIONS 定義をインストールする場合、その定義を使用するには、まず領域間通信(IRC)をすべて閉じてから 再び開く必要があります。

### このタスクについて

#### 手順

1. IRC を閉じます。  
以下のコマンドを使用します。  
`CEMT SET IRC CLOSED`
2. リソース定義をインストールします。  
以下のコマンドを使用します。  
`CEDA INSTALL GROUP(groupname)`
3. 定義が含まれるグループのインストールが正常に完了したら、IRC を再度開きます。  
以下のコマンドを使用します。  
`CEMT SET IRC OPEN`

## 一時データ・キュー定義のインストール

CICS の初期設定後、CEDA トランザクションまたは EXEC CICS CREATE コマンドを使用して、追加のリソースをインストールすることができます。

### このタスクについて

一時データ・キューは、常に使用可能状態でインストールされます。CICS システムの終了時に使用不可だったキューは、ウォーム・スタートまたは緊急再始動でシステムを復元したときには使用可能になっています。

## 既存の一時データ・キュー定義の置換

特定の条件を満たした場合に、既存の一時データ・キューを置き換えることができます。

### このタスクについて

既存の一時データ・キューを置き換えるには、以下のすべての条件が満たされている必要があります。

- CICS 初期設定が完了している
- キュー TYPE が既存の定義と同じである
- 区分内キューが使用不可になっている
- 区分外キューが使用不可で閉じられている

CICS のコールド・スタート時に既存の定義を置き換えることは**できません**。複数のグループを使用している場合は、重複する名前が存在しないことを確認してください。重複する名前があると、エラー・メッセージが発行されます。重複する定義は、既存の定義の置換に使用されません。

次のトランザクションとコマンドを使用して、一時データ定義のインストール後にそれらを照会、設定、破棄することができます。

- CEMT INQUIRE TDQUEUE (または EXEC CICS INQUIRE TDQUEUE)

- CEMT SET TDQUEUE（または EXEC CICS SET TDQUEUE）
- CEMT DISCARD TDQUEUE（または EXEC CICS DISCARD TDQUEUE）
- CECI トランザクション

## 一時データ・キューの使用不可化

一時データ・キューは、使用中あるいは他のタスクが使用を待機中には使用不可にすることができません。

### このタスクについて

区画内キューおよび区画外キューの場合:

- タスクが区画外キュー、物理的にリカバリー可能なキュー、またはリカバリー不能な区画内キューの使用を待機している場合、そのキューは「使用不可保留中」状態になります。キューを最後に使用したタスクはキューを完全に使用不可にします。
- 作業単位がエンキューされている状態の論理的にリカバリー可能な区画内一時データ・キューを使用不可にしようとした場合、そのキューは「使用不可保留中」状態になります。エンキューを取得した最後の作業単位は、区分内キューを完全に使用不可にします。
- 作業単位が「使用不可保留中」状態のキューのエンキューを所有している場合は、更新を続行することができます。
- キューが「使用不可保留中」状態になると、新規タスクがキューの状態や内容を変更することはできません。CICS は、「使用不可保留中」状態のキューに対して READQ TD, WRITEQ TD 要求あるいは DELETEQ TD 要求が発行されると、使用不可応答を返します。

## 端末定義のインストール

新規または変更された TERMINAL リソース定義を CICS 領域内でアクティブにするには、そのリソース定義をインストールする必要があります。

### このタスクについて

TERMINAL 定義および TYPETERM 定義は、インストール時に解決されて TCT エントリー (TCT 端末エントリーや TCTTE と呼ばれます) になります。各定義には、TCT エントリーの作成に必要な情報の一部のみが含まれます。

各 TYPETERM 定義は、それを参照する TERMINAL 定義より前あるいは同時にインストールする必要があります。CEDA を使用してグループをインストールする際に、TYPETERM 定義が TERMINAL 定義と別のグループ内にある場合は、TERMINAL グループの前に TYPETERM グループをインストールする必要があります。長期間の使用は推奨されませんが、テストのために TYPETERM 定義と TERMINAL 定義を同じグループに含めることは可能です。

TERMINAL 定義が、関連付けられた TYPETERM 定義と同じグループ内にある必要はないため、TYPETERM 定義の保管にはグローバル・カタログが使用されます。

TYPETERM 定義を変更してから再インストールしても、既にインストールされている端末エントリーには影響しません。これは、その定義を作成するために使用した TERMINAL 定義が TYPETERM を参照している場合でも同じです。端末エントリーを変更するには、TYPETERM 定義と TERMINAL 定義の両方を再インストールする必要があります。

TERMINAL 定義のグループの場合は、以下の手順を使用します。

### 手順

1. インストールする端末を誰も使用していないことを確認します。**CEMT INQUIRE TERMINAL** コマンドを使用すると TCT エントリーがロックされ、その端末が含まれるグループのインストールを妨げるので注意してください。
2. 端末に未解決の ATI がないことを確認します。
3. リソース定義をインストールします。以下のコマンドを使用します。

```
CEDA INSTALL GROUP(groupname)
```

4. CEMT を使用して、端末を再度使用可能にします。以下のコマンドを使用します。

```
CEMT SET TERMINAL(trmid) ACQUIRED INSERVICE ATI
```

## 端末定義の検査

CHECK コマンドを使用して TERMINAL 定義のグループを検査することができます (例えば、このコマンドではディスプレイ 装置からプリンターへの参照が解決されます)、TYPETERM 参照が別のグループ内にある場合は、TYPETERM がないことを示す不要なメッセージが大量に生成されるため、そのような参照を解決するにはあまり有用ではありません。

### このタスクについて

端末装置の新規クラスターを追加する場合、それらのクラスターの新規グループを作成してから、ご使用の TYPETERM 定義グループと TERMINAL 定義の新規グループが含まれるリストを作成します。その後、CHECK コマンドを使用して、リスト全体を検査します。これには、時間はかかりません。このリストは、検査中にのみ必要で、GRPLIST として指定されることはありません。

TERMINAL 名の重複を避けるために、TERMINAL 定義が含まれるすべてのグループのリストを保持することができます。CHECK LIST を使用して、すべての新規 TERMINAL 名が固有であることを確認することができます。この処理に長時間かかる場合、例えば以下のように、類似した文字から始まる TERMINAL 名が別のグループに保持されていれば、この問題を回避することができます。

```
TERMINAL (AZ01) GROUP (AZTERMS)
TERMINAL (AZ02) GROUP (AZTERMS)
TERMINAL (AZ03) GROUP (AZTERMS)
      :
      :
      :
TERMINAL (AZnn) GROUP (AZTERMS)

TERMINAL (BJ01) GROUP (BJTERMS)
TERMINAL (BJ02) GROUP (BJTERMS)
TERMINAL (BJ03) GROUP (BJTERMS)
      :
      :
      :
TERMINAL (BJnn) GROUP (BJTERMS)
```

**要確認:** 端末の自動インストールを使用する場合、自動インストール・モデル定義をインストールする前に TYPETERM 定義をインストールして、確実にモデルが作成されるようにする必要があります。CHECK コマンドでは、そのような定義の順序は検査しません。

## URIMAP リソース定義のインストール

この手順では、CEMT トランザクションおよび CEDA トランザクションを使用して、URIMAP リソース定義をインストールします。URIMAP リソースが既に存在する場合は、再インストールする前に無効にする必要があります。

### 手順

1. URIMAP リソースが既に存在する場合は、無効にされている必要があります。

以下のコマンドを使用します。

```
CEMT SET URIMAP(name) DISABLED
```

URIMAP リソースが無効にされている場合、Web クライアントがそのリソースを必要とする HTTP 要求を行うと、CICS はエラー・メッセージ DFHWB0763 を発行し、Web エラー・プログラムを介して Web クライアントに HTTP 503 応答 (Service Unavailable) を返します。Web エラー・プログラムを変更することにより、この応答を調整できます。

2. URIMAP 定義をインストールします。

以下のコマンドを使用します。

CEDA INSTALL GROUP(*groupname*) URIMAP(*name*)

URIMAP 定義をインストールすると、CICS は以下のセキュリティー検査を実行します。

- URIMAP 定義で SCHEME(HTTPS) が指定されている場合、CICS は、インストール時に SSL が CICS 領域内でアクティブであるかを検査します。これは、KEYRING システム 初期設定パラメーターを使用して CICS 領域で使用される鍵リングを指定することで指示されます。CICS 領域内で SSL がアクティブにされていない場合、CICS はメッセージ DFHAM4905 を発行し、URIMAP 定義はインストールされません。
- URIMAP 定義で CIPHERS 属性が指定されている場合、CICS は稼働中のシステムでサポートされている暗号に対して、暗号リストの妥当性検査を行います。リスト内で有効な暗号が見つからない場合、CICS はメッセージ DFHAM4918 を発行し、URIMAP 定義はインストールされません。ただし、リスト内の一部の暗号のみがサポートされている場合、CICS はメッセージ DFHAM4917 を発行し、URIMAP は暗号コードのセットを削減してインストールされます。
- URIMAP 定義で CERTIFICATE 属性が指定されている場合、CICS は鍵リングで指定された証明書の妥当性検査を行います。指定された証明書が有効ではない場合、CICS はメッセージ DFHAM4889 および DFHAM4928 を発行し、URIMAP 定義はインストールされません。

**ヒント:** CICS は、外部セキュリティー・マネージャーのデータベース内の CICS 領域の鍵リングに保持されている情報に対して、証明書の妥当性検査を行います。CICS 領域は、SSL ハンドシェイクを実行する場合に、CICS 領域の SSL 環境内の証明書のキャッシュからの情報を使用します。これは、z/OS システム SSL によって管理されます。この証明書を鍵リングに追加した場合、あるいは CICS 領域の SSL 環境を最後に構築または再構築して以降にその証明書を更新した場合は、CICS 領域に対して PERFORM SSL REBUILD コマンドを発行します。このコマンドは、証明書のキャッシュをリフレッシュし、この証明書の正しい情報がキャッシュ内に存在するようにします。

3. オプション: URIMAP 定義のインストールが正常に完了したら、CEMT を使用してリソースを使用可能にします。

このステップは、URIMAP リソースがまだ ENABLED として定義されておらず、Web クライアントあるいは Web サービスでこのリソースを使用できるようにしたい場合にのみ実行します。以下のコマンドを使用します。

CEMT SET URIMAP(*name*) ENABLED

## WEBSERVICE リソース定義のインストール

CICS は、WEBSERVICE リソースを作成し、それらをご使用の CICS 領域にインストールするための通常のリソース定義メカニズムを提供しますが、使用可能な代替戦略もあります。スキャン・メカニズムを使用して、稼働中の CICS システムに WEBSERVICE リソースをインストールすることができます。

### このタスクについて

WEBSERVICE リソース定義をインストールするには、3 つの方法があります。

#### 手順

- **PERFORM PIPELINE SCAN** コマンドを使用して (CEMT トランザクションまたはシステム・プログラミング・インターフェースを使用)、PIPELINE リソースのピックアップ・ディレクトリーのスキャンを開始します。

この方法は、既にインストールされている PIPELINE のピックアップ・ディレクトリー内に、追加または更新された Web サービス・バインディング・ファイルがある場合に使用します。CICS は、ピックアップ・ディレクトリーをスキャンし、そこで見つかった各 Web サービス・バインディング・ファイルを使用して、WEBSERVICE リソースを動的にインストールします。

- PIPELINE リソースをインストールします。

この方法は、PIPELINE リソースのピックアップ・ディレクトリーに PIPELINE に関連付けたい Web サービス・バインディング・ファイルが含まれており、PIPELINE がまだインストールされていない場合に使用します。PIPELINE をインストールすると、CICS は、ピックアップ・ディレクトリーをスキャンし、そこで見つかった各 Web サービス・バインディング・ファイルを使用して、WEBSERVICE リソースをインストールします。



- CSD から WEBSERVICE リソースをインストールします。

この方法は、他の 2 つの方法がどちらも適切でない場合に使用します。例えば、PIPELINE 定義を PIPELINE のピックアップ・ディレクトリー内にはない Web サービス・バインディング・ファイルと一緒に使用したい場合などです。

#### **次のタスク**

CSD からインストールされた WEBSERVICE 定義によって参照されている Web サービス・バインディング・ファイルを更新した場合、その WEBSERVICE を破棄してから再インストールし、新規の Web サービス・バインディング・ファイルを取り込む必要があります。

## 第7章 共用データ・テーブルの構成

仮想記憶間サービスを使用してファイルを共用するように、共用データ・テーブルを構成できます。共用データ・テーブル機能は、CICS ファイル管理サービスの拡張機能で、機能シップを使用するアプリケーションのパフォーマンスを改善できます。

### データ・テーブルの使用計画

データ・テーブルを使用する主な理由は、パフォーマンス上の利点を活用することです。

このセクションでは、診断、変更、またはチューニングについて説明します。

### CICS 保守データ・テーブルのパフォーマンス

ファイルのすべてのデータ・レコードと索引レコードが完全に LSR プールに含まれている場合は、そのファイルを CICS 保守データ・テーブルとして定義しても DASD I/O アクティビティは減りません。しかし、CPU 消費量が減少する可能性が大いにあります。また、LSR プール内のバッファ数減らすこともできます。

ファイルが LSR プール内に完全に含まれていない場合、CICS 保守データ・テーブルを使用すると、DASD I/O アクティビティと CPU 消費量の両方とも減少することになります。

ローカル共用リソース (LSR) プール内にある VSAM KSDS と比較して、CICS 保守データ・テーブルの CPU 消費量をどの程度節約できるかは、アプリケーションの使用状況に応じて異なります。

### ユーザー保守データ・テーブルのパフォーマンス

ユーザー保守データ・テーブルのロード後、すべてのデータ・テーブル操作から DASD I/O アクティビティが除外されるので、LSR プール内にある VSAM KSDS と比較して CPU 消費量を大幅に節約できます。

### 共用データ・テーブル用のストレージの使用

共用データ・テーブルにより、メモリー内のデータを効率的に使用できるようになります。つまり、ストレージの使用量が少し増えるというコストだけで、パフォーマンス上の大きな利点が生じます。

このストレージ使用の概説では、実記憶域と仮想記憶域や、アドレス・スペース・ストレージとデータ・スペース・ストレージなどの、さまざまなタイプのストレージの違いについて理解していることが前提になっています。ほとんどの場合、データ・スペース・ストレージがストレージとして使用されます。このストレージは、アドレス・スペース仮想記憶域から分離された仮想記憶域です。

共用データ・テーブルは仮想記憶域を以下のように使用します。

- レコード・データは DFHDT003、DFHDT004、DFHDT005 と続くデータ・スペースに保管され、必要に応じて新しいデータ・スペースが割り振られます。ロード時の合計レコード・データ・ストレージは、基本的にすべてのレコードの合計サイズ (キーはテーブル・エントリー・ストレージに保管されるので除きます) に少量の制御情報を加算したサイズになります。データ・スペース・ストレージは、16 MB 単位で獲得され、個別のテーブルに割り振られる際に 128 KB ずつ増加します。その後ストレージは、ページ位置合わせされたフレームに 2 次割り振りされます。このフレームは、テーブルの最大レコード長を十分組み込むことができる大きさです。データ・テーブル・フレームは、大まかに VSAM 制御間隔と同等で、通常は同じようなキーがあるレコードのセットを保持します。可能な場合、新しいレコードはそれぞれ、近接するキーのうち小さな方の値を持つ既存のレコードと同じフレームに保管されます。

ロード後に多くのレコードの長さが増加したり、新しいレコードがファイルの大部分にランダムに追加されたりすると、ストレージの量は増え、最大で元のサイズの 2 倍になることがあります。

- テーブル・エントリー記述子ストレージは、データ・スペース DFHDT001 から割り振られます。割り振られる際に 32 KB ずつ増加します。

テーブル内のレコードごとにエントリー記述子が 1 つずつあり、さらにキー・シーケンス内のギャップ (CICS 保守データ・テーブルから 1 つ以上のレコードが省かれている場所) ごとにエントリー記述子が 1 つずつあります。各エントリーのサイズは、キー長に 9 バイトを加算し、次の 8 KB の倍数に切り上げた値です。

- 索引ノード・ストレージは、データ・スペース DFHDT002 から割り振られます。割り振られる際に 32 KB ずつ増加します。

この領域のサイズは、[338 ページの表 32](#) で示されているキー値の分散とフォーマット、さらに実際のレコード数によって決まります。

表 32. キーの分散とフォーマット		
キーの分散	キー・フォーマット	レコード当たりのバイト数
高密度 (すべてのキーが連続している)	バイナリー 10 進数 英字	5.1 8.5 19
低密度 (連続しているキーがない)	10 進数 英字	44 51
最悪の事例	-	76

- ECSA ストレージは、データ・テーブルを共用するすべての領域がアクセスする必要がある、いくつかの小さな制御ブロックに使用されます。

ファイルを共用データ・テーブルに変換すると実記憶域の使用量が増える可能性があります。更新がほとんどない場合は VSAM LSR バッファ用の実記憶域の使用量が減る可能性があります。また、各 CICS 領域内で読み取り専用テーブルのレプリカを生成して現在高性能を実現しているアプリケーションも、各テーブルの単一コピーを共用してストレージを大幅に節約できる可能性があります。

特定のデータ・テーブルに割り振られたストレージは、そのテーブルが閉じられるまで割り振られたままの状態になります。例えば、データ・テーブルが 1 GB に拡大した後、そのテーブルからすべてのレコードが削除されても、引き続きそのテーブルは 1 GB のデータ・スペース・ストレージを所有しています。所有しているデータ・テーブルを閉じるまで、他のデータ・テーブルはそのストレージを使用できません。

データ・テーブル内のフリー・スペースは追跡され、該当する場合には再利用されます。例えば、テーブル・エントリー記述子や索引ノードが不要になると、同じテーブル内で再利用できるようにフリー・チェーンに追加されます。同様に、レコード・データ・フレーム内のすべてのレコードが削除されると、空のフレームがフリー・チェーンに戻されます。フレーム内の一部のレコードのみ削除された場合に、そのスペースが再利用されるのは、新しいレコードのキーが、偶然同じフレーム (または、そのフレーム内にスペースがない場合は直前のフレーム) 内の別の既存のレコードの直後のキーだった場合だけです。VSAM 制御間隔とは違って、フレーム内のレコードは、記述子によって間接的に配置されるので、必ずしもキー・シーケンスにはなりません。また、統合フリー・スペースでは同時読み取りできないので、レコードを統合フリー・スペースに移動できません。

継続的に増えてほぼ同じシーケンス内で削除されるキーがレコードに割り振られると、通常、スペースは非常に効率的に再利用されます。その理由は、通常新しいレコードでフレームがいっぱいになってから新しいフレームに移行するので、結局古いフレームは完全に空になり、再利用できるようになるからです。また、複数の別個の範囲が、フレーム全体を解放するのに十分な大きさである場合も、スペースが同様に効率的に再利用されます。このような状態では、データ・テーブル・ファイルに割り振られるストレージの量は、使用中のストレージの量に近くなります。

新しいデータ・テーブル・アプリケーションを導入すると、データ・テーブルごとに割り振られるストレージと使用中のストレージをモニターし、現在と将来の使用を十分にサポートできるだけのオペレーティング・システム・リソースが使用可能であることを確認するのに役立つことがあります。割り振られたストレージを読み取る際には、各データ・テーブルで所有されるストレージが示されます。このストレージはデータ・テーブルが削除されるまで解放されません。使用中のストレージを読み取る際には、使用中の割り振られたストレージの量が表示されます。この情報は CICS サンプル統計プログラム DFHOSTAT により提供されます。DFHOSTAT については、[サンプルの統計プログラム DFHOSTAT](#) に説明があります。

共用データ・テーブルにより記述子、索引エントリー、またはデータ用のスペースが使い尽くされることがあります。ロード時だけでなく、通常の実行中にレコードが追加される際や、更新される際であっても、スペースが使い尽くされることがあります。現在 CICS は共用データ・テーブルをサポートするために複数のデータ・スペースを使用しているので、全 3 種類のストレージに関する限度は非常に大きくなり、CICS アドレス・スペース内のエントリーがなくなったなどの他の考慮事項とは無関係になります。ただし、使

用可能なストレージは依然として有限です。例として、非常に多数の比較的小さなレコードがあり、特にそのほとんどがキー・データで構成されている場合、キー長や他の要因によっては、レコード・データ自体のためのストレージより前にエントリー記述子や索引ノード用のスペースが使い尽くされることがあります。エントリー記述子や索引ノード用のスペースが不足している場合は、データ・テーブルを別個の FOR など、異なる CICS 領域に分割することを考慮してください。単一のデータ・テーブルが独自のスペースを使い尽くすと、そのデータ・テーブル用のスペースの限度に達するので、複数の別個のテーブルに分割する必要があるか考慮しなければなりません。

## 共用データ・テーブルを使用する際の MVS JCL 要件

共用データ・テーブルを使用する前に、ご使用の JCL ステートメントへの変更、操作手順の変更、または MAXUSER MVS 初期化パラメーターの値の増加が必要になる場合があります。

これは、ジョブの複数のステップが共用データ・テーブル・サーバーとして動作することを MVS が許可しないからです。2 番目のジョブ・ステップが共用データ・テーブル・サーバーとして動作しようすると、CICS はメッセージ DFHFC0405 を発行します。さらに、サーバー・ステップの後に続くジョブ・ステップも MRO で仮想記憶間サービスを使用することができないので、サーバー・ステップの後に続くジョブ・ステップがどれも CICS の別の実行にならないようにすることをお勧めします。

共用データ・テーブル・サーバー・ステップを含むジョブが、このサーバーに接続したすべての要求側ジョブ・ステップが終了する前に終わる場合、サーバーのアドレス・スペースが MVS によって終了させられます。共用データ・テーブル・サーバーが、開始タスクではなくバッチ・イニシエーターの制御下で実行されている場合は、前述の状況が発生したときに、新しいイニシエーターを開始しなければなりません。

MVS はメッセージ IEF355A INITIATOR TERMINATED, RESTART INITIATOR を出してバッチ・イニシエーターを終了します。これは整合性の理由からであり、これを行わなければ MVS はそのイニシエーターの下で実行される次のジョブで使用される可能性がある機能を制限しなければならなくなって、そのジョブが失敗する原因になる可能性があるからです。MVS は、そのサーバーに接続したすべての要求側ジョブ・ステップが終了するまで、共用データ・テーブル・サーバーの ASID が再使用されることを許可しません。

## データ・テーブルとして使用するファイルの選択

ファイルを共用データ・テーブルに変換することで利点が得られるかどうかに関して、正確な規則を定めることはできません。このトピックのチェックリストに、一般的な説明が記載されています。

多数の考慮事項があるので、各種アプリケーションがファイルをどのように使用しているかや、CICS 領域の構成について把握している人が、共用データ・テーブル・サポートの潜在的な使用について分析する必要があります。

ファイルの選択に役立つ追加の情報源には以下のものがあります。

- ファイル統計。選択作業に対する入力データの 1 つとして統計情報を使用する方法については、[340 ページの『データ・テーブル選択のために統計を使用する』](#)を参照してください。
- LSR プール統計。
- トレース・エントリー。
- モニター・データ。

しかし、選択プロセスに対する最も役立つ入力、アプリケーションとそのアプリケーションがファイルを使用する方法について十分に理解していることです。

インストール済み環境で初めてデータ・テーブルを使用する場合は、以下のチェックリストの一般原則を、データ・テーブルとして定義するファイルの選択に役立ててください。

- CICS 保守データ・テーブルが実装しやすいので、最初にこのデータ・テーブルの使用を考慮する必要があります。CICS 保守データ・テーブルを使用する場合、アプリケーションに対する変更は不要です。ユーザー保守データ・テーブルを使用する場合、変更が必要になることがあります。
- CICS リスタートの間データ・テーブルの保全性を保証する必要がある場合は、CICS 保守データ・テーブルを使用します。
- 更新のジャーナリングが必要な場合は、CICS 保守データ・テーブルを使用します。すべてのアクセス要求のジャーナリングが必要な場合、ファイルをデータ・テーブルとして使用することは適切ではありません。



- 仮想記憶間サービスでデータ・テーブルへのアクセス要求が満たされる場合、EXEC インターフェースのユーザー出口である XEIIIN と XEIOU、ファイル制御のユーザー出口である XFCREQ と XFCREQC は、file-owning 領域で呼び出されません。ファイルを選択する際には、アプリケーションの正常な操作が、これらのユーザー出口で実行されるアクティビティーに依存しないことを確認する必要があります。
- 343 ページの『データ・テーブルのセキュリティ 検査』で説明されている、データ・テーブルを共用する際のセキュリティへの影響に注意する必要があります。
- 別の領域からファイルに頻繁にアクセスする場合、多数の他の領域からアクセスする場合、またはアクセスが主に読み取り要求である場合は、そのファイルをデータ・テーブルにする利点は非常に大きくなります。ローカル・ファイルよりリモート・ファイルの方がパフォーマンスの向上が大きいことに留意してください。
- CICS 保守データ・テーブルの場合、データ・テーブルのみにアクセスする要求の比率が適度に高いファイルを選択します (データ・テーブルへのアクセス用の開発を参照)。最大のパフォーマンスを得るには、その中から、このような要求の使用率が最も高いファイルを選択します。  
ファイルの使用状況に関する情報は、DFHSTUP レポートのファイル制御統計にあります。データ・テーブルを活用できない読み取り要求もあるので、後で CICS 統計レポート内のデータ・テーブル情報を調べて、そのデータ・テーブルが効率的に使用されていることを確認する必要があります。詳細については、データ・テーブルのモニターを参照してください。
- ユーザー保守データ・テーブルの場合、更新アクティビティーの比率が多く、CICS リスタート中にその更新のリカバリーの必要がないファイルを選択します (347 ページの『データの保全性』を参照)。
- パフォーマンス測定を使用して、将来の使用予測に留意し、おおよその CPU 使用率の削減を見積もります。
- 最善の見積もりのファイルを 1 つまたは 2 つ選択します。削減の見積もりが同じである場合は、大きなファイルより小さなファイルを優先します。小さなファイルの方が実記憶域の使用量が少ない可能性が高いからです。
- 実記憶使用量をモニターしてください。既にシステムに実記憶の制約がある場合は、大きいデータ・テーブルを使用するとページイン率が増す可能性があります。これにより、CICS システム・パフォーマンスに悪影響を与えます。RMF (バージョン 5) など、標準のパフォーマンス・ツールを使用して、実記憶の使用量とページング率を監視してください。
- LSR プール内のバッファ数を減らすことを考慮してください。データ・テーブルを使用すると、LSR プールを使用する回数が減る可能性があるからです。
- ユーザー出口 XDTRD を使用して、データ・テーブル内に含まれるレコードを選択してください。さらに、ユーザー保守データ・テーブルの場合、ユーザー出口 XDTRD を使用してレコードを変更することもできます。つまり、必要なデータのみデータ・テーブルに保管して、仮想記憶域や実記憶域の使用を最適化できます。
- 非常に大きなデータ・テーブルの場合、MVS IEFUSI 出口で設定される普通の領域制限より多くのデータ・スペース・ストレージが必要になることがあります。この場合、IEFUSI 出口を変更して制限を増やすか、CICS XDTRD グローバル・ユーザー出口プログラムを使用して一部のレコードを抑制することができます。IEFUSI 出口については、「z/OS MVS 導入システム 出口」マニュアル (SA88-8565) を参照してください。

## データ・テーブル選択のために統計を使用する

共用が単一 MVS イメージに限定されている場合、共用データ・テーブルを活用するアクセス・パターンをどのファイルが持つかを検討する必要があります。

データを複数の MVS イメージで共用する必要がある場合、RLS モードを使用したファイルの共用について調べるべきです。

341 ページの図 56、341 ページの図 57、および 342 ページの図 58 に、非 RLS モードでアクセスされるファイルの架空のファイル統計セットからの抜粋を示します。これは、後述する CICS 統計をどのように選択処理に役立てることができるかを示す例で使います。

統計は、CICS オフライン・フォーマット・ユーティリティーによって報告された場合と同様に表示します。要求されたファイル統計を示していますが、間隔統計または終業時統計も同様に適しています。ファイル・セクションの「パフォーマンス 情報」統計は VSAM ストリングとバッファの使用量を報告しますが、ここには表示していません。

図に示す数値は純粋に図示を目的としています。ご使用のシステムの統計がこれに似ているとは期待しないでください。また、CICS 領域とファイルの構成は、特定のポイントを強調する目的で選択しました。これが典型的な、あるいは望ましい構成だというわけではありません。

データ・テーブルのモニターで、データ・テーブルとして定義されたファイルに関して報告される統計について説明しています。これは、得られる利点を評価するために使用できます。

Requested Statistics Report		Collection Date-Time 12/25/99-11:51:51		Last Reset 09:00:00		Applid CICFOR		Jobname SDTGSTF1			
FILES - Resource Information											
File Name	Data Set Name Base Data Set Name (If Applicable)	Data Set Type	RLS File	DT Indicator	Time Opened	Time Closed	Remote Name	Remote Sysid	Lsrpool ID		
APPLE	CIC01.CICOWN.APPLES	K	NO		07:44:12	OPEN			1		
BANANA	CIC01.CICOWN.BANANAS	K	NO		09:45:08	OPEN			1		
ORANGE	CIC01.CICOWN.CITRUS	K	NO		10:51:10	OPEN			2		
PEAR	CIC01.CICOWN.PEARS	K	NO		07:30:14	OPEN			3		
Requested Statistics Report		Collection Date-Time 12/25/99-11:51:51		Last Reset 09:00:00		Applid CICFOR		Jobname SDTGSTF1			
FILES - Requests Information											
File Name	Get Requests	Get Upd Requests	Browse Requests	Update Requests	Add Requests	Delete Requests	Brws Upd Requests	VSAM EXCP Data	Requests Index	RLS req Timeouts	
APPLE	2317265	1020	0	1019	21	1	0	11503	310	0	
BANANA	536452	1674	20344	1674	908	0	0	2651	70	0	
ORANGE	2069454	98560	17831	98327	4543	2563	0	8511	481	0	
PEAR	45871	65493	6512	65493	30109	362	0	3773	231	0	
*TOTALS*		4969042	166747	44687	166513	35581	2926	0	0	0	
Requested Statistics Report		Collection Date-Time 12/25/99-11:51:51		Last Reset 09:00:00		Applid CICFOR		Jobname SDTGSTF1			
FILES - Data Table Requests Information											
File Name	Close Type	Read Requests	Recs in Table	Adds from Reads	Add Requests	Adds rejected - Exit	Adds rejected - Table Full	Rewrite Requests	Delete Requests	Highest Table Size	Storage Alloc(K)
DFHST0223 I There are no data table statistics to report.											

図 56. CICFOR の要求されたファイル統計

Requested Statistics Report		Collection Date-Time 12/25/99-11:51:38			Last Reset 09:00:00		Applid CICAOR1		Jobname SDTGST1		
FILES - Resource Information											
File Name	Data Set Name Base Data Set Name (If Applicable)	Data Set Type	RLS File	DT Indicator	Time Opened	Time Closed	Remote Name	Remote Sysid	Lsrpool ID		
APPLE	REMOTE				CLOSED	CLOSED	APPLE	CIF1	N		
BANANA	REMOTE				CLOSED	CLOSED	BANANA	CIF1	N		
ORANGE	REMOTE				CLOSED	CLOSED	ORANGE	CIF1	N		
ZUCCHINI	REMOTE				CLOSED	CLOSED	COURGETT	CIA2	N		
Requested Statistics Report		Collection Date-Time 12/25/99-11:51:38			Last Reset 09:00:00		Applid CICAOR1		Jobname SDTGST1		
FILES - Requests Information											
File Name	Get Requests	Get Upd Requests	Browse Requests	Update Requests	Add Requests	Delete Requests	Brws Upd Requests	VSAM EXCP Data	Requests Index	RLS req Timeouts	
APPLE	1158701	532	0	531	11	1	0	0	0	0	
BANANA	305641	0	19067	0	0	0	0	0	0	0	
ORANGE	58709	32854	4265	32621	1018	1001	0	0	0	0	
ZUCCHINI	78914	0	14765	0	0	0	0	0	0	0	
*TOTALS*		1601965	33386	38097	33152	1029	1002	0		0	
Requested Statistics Report		Collection Date-Time 12/25/99-11:51:38			Last Reset 09:00:00		Applid CICAOR1		Jobname SDTGST1		
FILES - Data Table Requests Information											
File Name	Close Type	Read Requests	Recs in Table	Adds from Reads	Add Requests	Adds rejected - Exit	Adds rejected - Table Full	Rewrite Requests	Delete Requests	Highest Table Size	Storage Alloc(K)
DFHST0223 I There are no data table statistics to report.											

図 57. CICAOR1 の要求されたファイル統計



Requested Statistics Report			Collection Date-Time 12/25/99-11:49:31		Last Reset 09:00:00		Applid CICAOR2	Jobname SDTGSTA2			
FILES - Resource Information											
File Name	Data Set Name Base Data Set Name (If Applicable)		Data Set Type	RLS File	DT Indicator	Time Opened	Time Closed	Remote Name	Remote Sysid	Lsripool ID	
COURGETT LEMON	CIC02.CICOWN.COURGETT REMOTE		K	NO NO		08:22:15 CLOSED	OPEN CLOSED	ORANGE	CIF1	1 N	
Requested Statistics Report			Collection Date-Time 12/25/99-11:49:31		Last Reset 09:00:00		Applid CICAOR2	Jobname SDTGSTA2			
FILES - Requests Information											
File Name	Get Requests	Get Upd Requests	Browse Requests	Update Requests	Add Requests	Delete Requests	Brws Upd Requests	VSAM EXCP Data	Requests Index	RLS req Timeouts	
COURGETT LEMON	78914 2010745	27469 65706	14765 13566	27469 65706	336472 3525	0 1562	0 0	8212 0	481 0	0 0	
*TOTALS*		2089659	93175	28331	93175	339997	1562	0	0		
Requested Statistics Report			Collection Date-Time 12/25/99-11:49:31		Last Reset 09:00:00		Applid CICAOR2	Jobname SDTGSTA2			
FILES - Data Table Requests Information											
File Name	Close Type	Read Requests	Recs in Table	Adds from Reads	Add Requests	Adds rejected - Exit	Adds rejected - Table Full	Rewrite Requests	Delete Requests	Highest Table Size	Storage Alloc(K)
DFHST0223 I There are no data table statistics to report.											

図 58. CICAOR2 の要求されたファイル統計

このサンプルでは、3つの CICS 領域からなる架空の構成を使用しています。CICS アプリケーションによって使用されるファイルの大部分はファイル所有領域 CICFOR によって所有されており、アプリケーションの大部分はアプリケーション所有領域 CICAOR1 および CICAOR2 で実行されます。この説明では、統計レポートに示された各データ・セットは VSAM ベース KSDS (Data Set Type の K で示されている) なので、これらのいずれもデータ・テーブルとして定義することができます。

このセクションは、CICS 保守データ・テーブルとして定義する候補を識別することを中心としています。なぜなら、ユーザー保守データ・テーブルの定義は、一般に、ファイルのパフォーマンスの検討からではなく、特定のアプリケーションの考慮事項から決定されることが多いからです。この点が中心となっているため、示す統計はどれも RLS モードでアクセスされるファイルのものではありません。RLS モードのデータ・セットは CICS 保守データ・テーブルのソースになることができないからです。

また、ここで示す統計では1つの領域のどのファイル名が別の領域のファイル名にアクセスするように定義されているかを示します。*Remote Sysid* (リモート・システム識別名) は、2つの領域間の接続で指定された名前です。このサンプルでは、CICFOR の SYSID は CIF2 であり、CICAOR2 の SYSID は CIA2 です。

### 更新に対する読み取りの比率が高いファイル

ファイル APPLE は、アプリケーション所有領域 CICAOR1 上で実行しているアプリケーションで使用されます。このファイルは CICAOR1 内でリモート・ファイルとして定義され、このファイル定義は CICFOR に所有されるファイル APPLE を指します。

このファイルを CICFOR 内で CICS 保守データ・テーブルとして再定義すると利点があります。それは、[341 ページの図 57](#) で示されているように、リモート更新 (追加 11、削除 1、更新 531) に対するリモート読み取り (レポート対象の期間内に 1158701 の Get Requests) の比率が高いためです。

統計レポートの「FILES - Requests Information」セクションの意味については、[DFHSTUP レポートのファイル制御統計の説明](#)を参照してください。

### リモート読み取りの比率が高いファイル

ファイル BANANA は CICFOR で更新されたり読み取られたりしますが、CICAOR1 でもアクセスされます。

リモート・アクセスはすべて更新ではなく読み取りとブラウズなので、BANANA がデータ・テーブルとして定義済みの場合は CICAOR1 内で実行しているアプリケーションには大きな利点がある可能性が高く、ローカル・データ・テーブルから読み取る CICFOR 上のアプリケーションにも利点がある可能性があります。

### 複数の領域で共用されるファイル

ORANGE が特に適切なデータ・テーブル候補でないように見える場合があります。

[341 ページの図 57](#) の統計は、CICAOR1 からのリモート取り出しの数 (Get Requests は 58709、Browse Requests は 4265) が比較的少ないことを示しています。しかし、CICAOR2 内のリモート・ファイル LEMON も CICFOR 内の ORANGE を指しているため、CICFOR 内で ORANGE を共用 CICS 保守データ・テーブルとして定義すると、両方の AOR 内のアプリケーションのパフォーマンスにとって利点がある可能性が高くなります。

### 適した UMT 候補

CICAOR2 で所有されるファイル COURGETT は、CICAOR1 内でファイル名 ZUCCHINI によりアクセスされます。

CICAOR1 はこのファイルの読み取りとブラウズのみ行います。更新は所有領域で発行されます。また、これらの更新は当日の CICS の実行だけに関係することがわかっているため、永続的に保存する必要はありません（実際、シャットダウン時に削除されます）。したがって、このファイルはユーザー保守データ・テーブルとして定義するのに適した候補です。その後、VSAM I/O アクティビティーなしですべての更新をデータ・テーブルに対して行うことができ、機能シップなしですべてのリモート取り出しを行うことができます。

### あまり適していない候補

ファイル PEAR はおそらく共用データ・テーブル・サポートの利点をあまり得られません。リモートにアクセスされず、多数の更新要求やブラウズ要求があるからです。

ローカル・ブラウズからは、ローカル読み取りやいずれの形式のリモート取り出しよりも利点を得られません。VSAM ブラウズ (STARTBR コマンドの処理を除く) が非常に効率的だからです。もちろんこの分析は、さまざまなファイル・アクセスの相対的な重要度を考慮に入れていません。読み取りは重要なアプリケーションによって行われる一方で、更新に費やされる時間は重要でない場合もあります。

### その他の可能な候補

前述の例は、共用データ・テーブル・サポートの利点が得られる可能性があるファイルの構成と使用法のごく小規模なサンプルを図示しています。

各領域内でファイルやデータ・テーブルを重複しなくてもよいように、共用データ・テーブル・サポートを使用することもできます。また、既存のファイルに目を向けるだけではなく、AOR から FOR にファイルを移動することも検討できます。AOR から FOR へのファイルの移動は、共有データ・テーブルのサポートがなかったときには、機能シップ使用時のファイル・アクセスのコストのために実用的ではありませんでした。

## データ・テーブルのセキュリティ検査

共用データ・テーブルは、LOGON 時や CONNECT 時にセキュリティ検査を実行して、仮想記憶間サービスの使用時にセキュリティを提供します。データ・テーブルに関連付けられているファイルを共用する前に、セキュリティ検査の影響を考慮する必要があります。

共用データ・テーブルで、以下を確認しなければなりません。

- FOR は偽名を使用できません。LOGON 時に、CICS 領域の指定済みの総称 *applid* を使用して FOR にログオンできるか検査し、偽名の使用を防止します。
- AOR は認識していないと思われるデータにアクセスできません。CONNECT 時に、AOR が FOR にアクセスできるか検査し、さらにファイル・セキュリティが実施されている場合は AOR が要求されたファイルにアクセスできるか検査してこのアクセスを防止します。

これらのセキュリティ検査は、System Authorization Facility (SAF) を使用して、リソース・アクセス管理機能 (RACF) か同等のセキュリティ・マネージャーを呼び出して実行します。

**注：**領域に共用データ・テーブル・サーバーとして動作する権限がない場合でも、その領域でローカルにデータ・テーブルを使用できます。

共用データ・テーブル・サポートには、領域レベルで作動する機能シップ・セキュリティの主な特性が再現されていますが、以下の違いに注意する必要があります。

- 共用データ・テーブル・サポートは、FOR がトランザクション・レベルでセキュリティ検査を実行するメカニズム (ATTACHSEC(IDENTIFY) や ATTACHSEC(VERIFY) と同等) を提供しません。したがって、AOR によって実行されるトランザクション・レベルの検査が一部のファイルにとって不適切と見なされる場合は、それらのファイルが FOR 内のデータ・テーブルと関連付けられていないことを確認しなければなりません。
- 共用データ・テーブル・サポートは、事前設定セキュリティをサポートしていません。
- 共用データ・テーブル・サポートは、インストール・パラメーター・リスト (INSTLN) 情報をセキュリティ・ユーザー出口に渡しません。

共用データ・テーブル・セキュリティの実装に必要なステップの説明については、[システム・リソースを保護するための RACF クラス](#)を参照してください。

### LOGON セキュリティ検査

LOGON 処理には、指定されたアプリケーション名の FOR がサーバーとして動作するよう許可されているか検証するセキュリティ検査が含まれます。

この検査により、アプリケーション所有領域 (AOR) が、事実上偽のファイル所有領域 (FOR) から偽のデータ・レコードを受け入れるリスクを最小限にすることができます。システム初期化時に SEC=NO を指定した場合であっても、この検査をバイパスすることはできません。

### CONNECT セキュリティ検査

CONNECT 時に実行されるセキュリティ検査により、2 つのレベルのセキュリティが得られます。

#### バインド・セキュリティ

CICS ファイル・セキュリティがない状態で実行している FOR が、共用アクセスを選択済みの AOR に制限できます。(ファイル・セキュリティなしで実行すると、実行時のオーバーヘッドとセキュリティ定義の数が最小になります。)

#### ファイル・セキュリティ

よりきめ細かいセキュリティ検査が必要な場合に FOR 内で活動化できます。その後、共用データ・テーブル・サポートによりこの種の検査が実装され、全体として AOR に適用されます。

ATTACHSEC(IDENTIFY) または ATTACHSEC(VERIFY) と機能シップを併用する場合、共用データ・テーブル・サポートでは、FOR がトランザクション・レベルで行うセキュリティ検査を実装する方法は提供されません。

## 共用データ・テーブルのサポートの使用準備

共用データ・テーブルのサポートを使用するには、以下の作業を実行する必要があります。一部の作業は、機能シップやデータ・テーブルを現在使用しているインストールの際に既に実行済みです。

### このタスクについて

- 以下のモジュールが MVS システムの LNKLIST 内の許可システム・ライブラリー内にあることを確認するか、これらのモジュールを LPALST 連結内のライブラリー内に移動します。
  - DFHDTSCV および DFHDTTCV (共用データ・テーブルを使用するすべての領域は、同レベルの SVC コードを使用しなければならないため)
  - DFHMRMS、つまり RESMGR 出口スタブ (CICS JOBLIB/STEPLIB データ・セットがメモリーの終わりで使用できないため)
- 以下のモジュールは、CICS のインストール時にターゲット・ライブラリー SDFHLINK に入れられます。このライブラリーは通常 LNKLIST 連結に含まれています。
  - SDFHLINK が LNKLIST 連結内にある場合は、モジュールを使用可能にするには、オペレーター・コマンド MODIFY LLA,REFRESH を発行し、確認メッセージ CSV210I LIBRARY LOOKASIDE REFRESHED を待つ必要があります。
  - SDFHLINK が LNKLIST 連結内にない場合は、モジュールをその連結に含まれる適切なライブラリーにコピーし、LLA リフレッシュを発行するか、LPALST 連結内のライブラリーにコピーし、CLPA を指定して MVS システムを再 IPL する必要があります。
- AOR 内のファイルが共用を使用する予定の場合は、CICS が MVS サブシステムとして定義されていることを確認してください。
- 必要なセキュリティのレベルに応じて、FOR が共用データ・テーブル・サーバーとして動作し、サーバーが所有するファイルに AOR がアクセスできるように、セキュリティ許可を定義します。単一の MVS イメージで、以下のことが可能です。
  - 任意の数の FOR が共用データ・テーブル・サーバーとして動作できます。
  - 単一の AOR で任意の数の FOR を使用できます。
  - 単一の FOR が任意の数の AOR にサービスを提供できます。

- 1つの領域が、あるデータ・テーブルにとっては AOR として、別のデータ・テーブルにとっては FOR として機能することが可能です。
- 2つの FOR に同じ APPLID がある場合、特定の時間には一方の FOR のみが共用データ・テーブル・サーバーとして使用されます。しかし、一方の FOR が共用データ・テーブル・サーバーとして動作し、同じ APPLID の他方の FOR が機能シッパされた要求に使用されることは妨げられません。共用データ・テーブル・サービスを使用するデータ・テーブル要求が機能シッパを使用する要求と同じ領域に送信されないリスクがあるので、操作手順でこの状況が許可されていないことを確認する必要があります。
- FOR 内のデータ・テーブルにしているファイルを、CICS 保守データ・テーブルまたはユーザー保守データ・テーブルとして定義します。
- 必要に応じて、AOR 内で追加のリモート・ファイル定義を作成します。既存のリモート・ファイル定義に対する変更は不要です。
- データ・テーブルを共用する予定の AOR の場合、システム初期設定パラメーターとして ISC=YES を指定し、関連する FOR への MRO リンクか ISC リンクを定義します。IP 相互接続性 (IPIC) 接続の場合、同等のシステム初期設定パラメーター TCP/IP=YES を指定し、関連する FOR への IPIC リンクを定義します。
- 共用データ・テーブルを使用する前に、ご使用の JCL ステートメントへの変更、操作手順の変更、または MAXUSER MVS 初期化パラメーターの値の増加が必要になる場合があります。詳しくは、[339 ページ](#)の『共用データ・テーブルを使用する際の MVS JCL 要件』を参照してください。

## ロード・モジュール

共用データ・テーブルを使用するには、以下のロード・モジュールを CICS 領域にインストールしなければなりません。

表 33. 共用データ・テーブル・サポートによって使用されるロード・モジュール			
ロード・モジュール	ロード・ライブラリー	ロード方法	説明
DFHDTINS	SDFHLOAD	16 MB 境界より上の CICS ロード	初期設定
DFHDT SVC	SDFHLINK	リンク・リストからの 16 MB 境界より上の MVS LOAD	MVS 許可が必要なすべての機能を実行します。
DFHDT FOR	SDFHAUTH	16 MB 境界より上の MVS LOAD	データ・テーブル FOR モジュール
DFHDT AM	SDFHAUTH	16 MB 境界より上のサブプール 252 ストレージへの MVS LOAD	データ・テーブル・アクセス・マネージャー。AOR から仮想記憶間モードで実行されるコードが含まれます。
DFHDT AOR	SDFHAUTH	16 MB 境界より上の MVS LOAD	データ・テーブル AOR モジュール
DFHDT CV	SDFHLINK	リンク・リストから ECSA への MVS LOAD	接続の妥当性検査 (AOR)
DFHDT XS	SDFHAUTH	ECSA への MVS LOAD	接続のセキュリティー検査 (FOR)
DFHMVRMS	SDFHLINK	リンク・リストからの 16 MB 境界より上の MVS LOAD	リソース・マネージャー EOT/EOM インターフェース・コード

## ストレージ占有

16 MB 境界より上のストレージを占めるモジュールの合計サイズは、約 41KB です。ECSA ストレージ内のモジュールの場合、ログオンされる FOR ごとに約 1.5KB と AOR ごとに約 0.5KB 必要です。

これらはすべてリンク・パック域 (LPA) に組み込むのに適格なモジュールですが、DFHDT FOR、DFHDT AM、DFHDT AOR、また場合によっては DFHDT CV はかなり頻繁に使用されるのでリンク・パック域へ入れることを考慮してください。

## データ・テーブルのリソース定義

データ・テーブルを定義する方法は、CICS ファイルを定義する方法と同じですが、使用するデータ・テーブルのタイプと、データ・テーブル内に保持できるレコードの最大数も指定する必要があることが違います。

VSAM KSDS 定義で、最大レコード長とキー長が提供されます。

[#unique 431](#) で説明されているように、CEDA DEFINE FILE コマンドを使用してファイルをデータ・テーブルとして定義できます。

さらに、既存のファイルのデータ・テーブル属性を変更したり検査したりするには、以下のコマンドを使用できます。

- EXEC CICS SET FILE コマンドと INQUIRE FILE コマンド ([347 ページの『データ・テーブルに関する EXEC CICS コマンド』](#)を参照)。
- CEMT SET FILE コマンドと INQUIRE FILE コマンド ([348 ページの『データ・テーブルに関する CEMT コマンド』](#)を参照)。

## CICS 保守データ・テーブルのリソース定義

CICS 保守データ・テーブルには、固定長レコードか可変長レコードのどちらかの形式を指定できます。

共用データ・テーブル・サポートでサポートされる最大レコード長は 32KB です。この長さは CICS ファイル管理でサポートされる長さより長いので、実際には制限があります。[CICS コマンドに渡される区域の長さを参照してください](#)。サポートされるレコードの最大数は 16,777,215 です。

基本 VSAM クラスターのみが、そのクラスターに基づく CICS 保守データ・テーブルを持つことができます。代替索引パスを介する読み取り要求はデータ・テーブルを使用しませんが、代替索引パスを介するソース・データ・セットへの変更はデータ・テーブルに反映されます。

CICS 保守データ・テーブルのソース・データ・セットを RLS アクセス・モードで開くことはできないので注意してください。したがって、ファイル定義で RLSACCESS(NO) を指定しなければならないので、その他のファイルをすべて同じ基本データ・セットに関連付ける必要があります。

CICS 保守データ・テーブルとして定義されたファイルが開かれた後、その定義内の同じソース・データ・セットの名前を指定しているその他の非 UMT ファイルは (CMT として定義されているかどうかにかかわらず)、自動的に同じデータ・テーブルを使用します。これらの他のファイルのいずれかが CMT として定義されている場合、そのファイルが開かれる際にメッセージ DFHFC0937 がコンソールに対して発行されます。これはエラー状態ではありません。ファイルが開かれ、可能な場合は必ず既存のデータ・テーブルが使用されます。

### VSAM SHAREOPTION

DISP=SHR を使用してソース・データ・セットを割り振ると、FOR 以外の領域により更新される可能性があるというリスクが生じます。更新された場合は、データ・テーブルはソース・データ・セットと一致しなくなります。このリスクを最小限にするには、VSAM 領域間 SHAREOPTION を 1 または 2 に設定する必要があります。

- 1 に設定すると、1つの領域がデータ・セットへの更新アクセス権を持つこと、または多数の領域が読み取り専用アクセス権を持つことができます。
- 2 に設定すると、1つの領域がデータ・セットへの更新アクセス権を持ち、同時に多数の領域が読み取り専用アクセス権を持つことができます。

DISP の設定にかかわらず、領域間 SHAREOPTION が 3 または 4 の場合や、2 で CICS 保守データ・テーブルに読み取り専用アクセス権がある (つまり別の領域がデータ・セットを更新できない) 場合は、警告メッセージが発行されます。

### データの保全性

CICS 保守データ・テーブルを使用するファイルを、リカバリー可能なリソースとして定義できます。システムやトランザクションの障害後に、ソース・データ・セットが通常の方法でリカバリーされます。

- システム障害後、ファイルの再オープン時にデータ・テーブルはリカバリーされたソース・データ・セットから再ロードされます。



- ・トランザクション障害後、動的トランザクション・バックアウトによりソース・データ・セットに加えられた変更が、データ・テーブルにも加えられます。

ソース・データ・セットにアクセスするファイル操作に関する自動ジャーナリングが(その他のファイルと同じ方法で)サポートされています。ソース・データ・セットにアクセスしないファイル操作はジャーナル処理されません。

## ユーザー保守データ・テーブルのリソース定義

ユーザー保守データ・テーブルには、可変長レコード形式を指定しなければなりません。

共用データ・テーブル・サポートでサポートされる最大レコード長は 32KB です。この長さは CICS ファイル管理でサポートされる長さより長いため、実際には制限があります。[CICS コマンドに渡される区域の長さを参照してください](#)。サポートされるレコードの最大数は 16 777 215 です。

ユーザー保守データ・テーブルのソース・データ・セットを RLS アクセス・モードで開くことができます。RLS モードのデータ・セットにアクセスするその他のファイル定義がある場合や、このデータ・セットがその他の CICS 領域で更新される場合は、このデータ・セットをユーザー保守データ・テーブルのソースにすることもできます。

別々のコマンドかマクロを使用して各データ・テーブルを定義し、すべての定義が同じソース・データ・セットを参照するようにして、そのデータ・セットから複数のユーザー保守データ・テーブルをロードできます。

データ・テーブルは VSAM KSDS からロードしなければなりません、その後アプリケーションにより、CICS アドレス・スペースからアクセスできるデータ・ソースから、ユーザー保守データ・テーブルにレコードをコピーできます。このデータ・ソースは、IMS ファイルか Db2 ファイルです。データ・テーブルのソース・データ・セットとして使用する KSDS は空でもかまいません。最大レコード長と、キーの長さ位置を定義する必要のみあります。

### データの保全性

ユーザー保守データ・テーブルを、リカバリー可能なリソースとして定義できます。データ・テーブルへの変更は、システム・ログに記録されませんが、CICS メモリー内に内部的に保持されます。したがって、トランザクション障害後に(動的バックアウトにより)データ・テーブルをリカバリーできますが、システム障害後はリカバリーできません。

これは、CICS 共有データ・テーブル機能では独自のリカバリーを管理し、ログ・マネージャーやリカバリー・マネージャーのサービスを使用しないためです。ただし、分散作業単位の一部としてリカバリー可能なデータ・テーブルに変更を加える場合は例外です。この場合、その他のリカバリー可能なリソースと同様に、リンクのレコードが 2 フェーズ・コミット・プロセスの一部としてシステム・ログに書き込まれます。しかし、変更自体はシステム・ログに記録されません。

システム障害後、ファイルの再オープン時にデータ・テーブルはソース・データ・セットから再ロードされます。以下のことを確認していない場合は、障害発生時に、ソース・データ・セットのコンテンツとデータ・テーブルのコンテンツが同じでない可能性に留意してください。

- ・どちらにも変更が加えられていない。
- ・または、両方に変更が加えられている。

ロード時にソース・データ・セットにアクセスする要求の場合のみ、自動ジャーナリングがサポートされます。ロード・プロセスでアクセスされるレコードはユーザー出口 XDTRD の前にジャーナル処理され、アプリケーション要求のためにアクセスされるレコードはユーザー出口 XDTRD の後にジャーナル処理されます。

## データ・テーブルに関する EXEC CICS コマンド

**EXEC CICS SET FILE** コマンドを使用して既存のファイルの定義を変更したり、**EXEC CICS INQUIRE FILE** コマンドを使用して既存のファイルの定義を検査したりできます。

これらのコマンドや、ここで説明されているパラメーターの使用法の詳細などのプログラミング情報については、[SET FILE](#) を参照してください。データ・テーブルに関係するパラメーターについて、以下で説明されています。

このセクションには、汎用プログラミング・インターフェースと関連ガイダンス情報が含まれています。



## SET FILE

以下のパラメーターはデータ・テーブルに関係しています。これらのパラメーターは、ファイルが閉じていて使用不可になっている場合に限り使用できます。

CICS 値データ域 (cvda) 内のファイルのデータ・テーブル属性を指定できます。

### TABLE(cvda)

ファイルを CICS 保守データ・テーブルとして定義するには、**CICSTABLE** の CVDA 値を指定します。

ファイルをユーザー保守データ・テーブルとして定義するには、**USERTABLE** の CVDA 値を指定します。

ファイルがデータ・テーブルでないことを示すには、**NOTTABLE** の CVDA 値を指定します。

注：CFTABLE を指定してカップリング・ファシリティ・データ・テーブルを示すこともできます。

### MAXNUMRECS(value)

データ・テーブルに格納できる最大レコード数を指定します。範囲は 1 から 99999999 です。値ゼロは限度がないことを意味します。

## INQUIRE FILE

データ・テーブルに関係するパラメーターは以下のとおりです。

以下のように指定して、ファイルの各データ・テーブル属性が CICS 値データ域 (cvda) 内に返されるように要求できます。

### TABLE(cvda)

値 **CICSTABLE** が返された場合、ファイルは CICS 保守データ・テーブルとして定義されています。

値 **USERTABLE** が返された場合、ファイルはユーザー保守データ・テーブルとして定義されています。

値 **CFTABLE** が返された場合、ファイルはカップリング・ファシリティ・データ・テーブルとして定義されています。

値 **NOTTABLE** が返された場合、現在ファイルはデータ・テーブルとして定義されていません。

値 **NOTAPPLIC** が返された場合、ファイルがリモート・ファイルなのでオプションを適用できません。

### MAXNUMRECS(cvda)

返される値は、データ・テーブルに格納できる最大レコード数を指定します。値ゼロは限度がないことを意味します。

## データ・テーブルに関する CEMT コマンド

CEMT SET FILE コマンドを使用して既存のファイルの定義を変更したり、CEMT INQUIRE FILE コマンドを使用して既存のファイルの定義を検査したりできます。

ここで説明されているパラメーターを含む、これらのコマンドの使用法の詳細については、[INQUIRE FILE](#) を参照してください。データ・テーブルに関係するパラメーターについて、以下で説明されています。

## SET FILE

以下のパラメーターはデータ・テーブルに関係しています。これらのパラメーターは、ファイルが閉じていて使用不可になっている場合に限り使用できます。

### {CICSTABLE|USERTABLE|CFTABLE|NOTTABLE}

ファイルを CICS 保守データ・テーブルとして定義するには、**CICSTABLE** を指定します。

ファイルをユーザー保守データ・テーブルとして定義するには、**USERTABLE** を指定します。

注：CFTABLE を指定してカップリング・ファシリティ・データ・テーブルを示すこともできます。

ファイルがデータ・テーブルでないことを示すには、**NOTTABLE** を指定します。

**MAXNUMRECS(value)**

データ・テーブルに格納できる最大レコード数を指定します。範囲は、1 から 99999999 です。値ゼロは限度がないことを意味します。

**INQUIRE FILE**

データ・テーブルに関するパラメーターは以下のとおりです。

**Data table**

値 **CICSTABLE** が返された場合、ファイルは CICS 保守データ・テーブルとして定義されています。

値 **USERTABLE** が返された場合、ファイルはユーザー保守データ・テーブルとして定義されています。

値 **CFTABLE** が返された場合、ファイルはカップリング・ファシリティ・データ・テーブルとして定義されています。

値 **NOTTABLE** が返された場合、現在ファイルはデータ・テーブルとして定義されていません。

**MAXNUMRECS(value)**

返される値は、データ・テーブルに格納できる最大レコード数を指定します。値ゼロは、最大数の限度がないことを意味します。



## 第 8 章 プラットフォームのセットアップ

プラットフォームは、クラウド対応の CICS Transaction Server for z/OS の主要機能です。CICS 領域を迅速なアプリケーション・デプロイメントのプラットフォームとしてグループ化できるので、基礎となるトポロジーからアプリケーションが分離され、柔軟性が高まります。

### 始める前に

CICSplex SM は、CICS 領域の単一システム・イメージを提供するための前提条件です。ビジネス・アプリケーション・サービス (BAS) やワークロード管理など、CICSplex SM の他の機能は不要です。

プラットフォームの概要については、[仕組み: プラットフォーム](#)を参照してください。プラットフォーム・リソースの保護については、[プラットフォームおよびアプリケーションのセキュリティ](#)を参照してください。

### このタスクについて

プラットフォームのセットアップの大部分の作業は、CICS Explorerで行います。CICS クラウド・パースペクティブは、プラットフォームのライフサイクルを管理するためのビューを提供します。CICS Explorerでの作業に加えて、必要なディレクトリー構造を zFS でセットアップします。

下記のステップは、プラットフォームのセットアップ・プロセスの概要を示しています。各ステップに、さらに詳細な説明へのリンクがあります。

### 手順

1. プラットフォームを設計します。  
どのようなアプリケーション、ポリシー、リソースをプラットフォームにデプロイするかを検討すると共に、新しい CICS 領域と領域タイプを作成するのか、それとも既存の CICS 領域とシステム・グループ定義を取り入れるのかを検討します。詳しくは、[352 ページの『CICS プラットフォームの設計』](#)を参照してください。
2. zFS で、プラットフォーム・ホーム・ディレクトリーを構成します。  
詳しくは、[プラットフォーム用の zFS の準備](#)を参照してください。
3. CICS Explorer で、プラットフォーム・プロジェクトを作成します。  
このプロジェクトに、領域タイプを追加し、デプロイする CICS バンドルとデプロイ先の領域タイプを指定します。
4. CICS Explorer から、プラットフォーム・プロジェクトを zFS にエクスポートします。  
エクスポート処理では、CICS プラットフォーム・プロジェクトで参照される CICS バンドルがパッケージされ、プラットフォーム・バンドルと CICS バンドルのすべてのファイルが zFS のプラットフォーム・ホーム・ディレクトリーにエクスポートされます。詳しくは、[358 ページの『プラットフォームのデプロイ』](#)を参照してください。
5. CICS Explorer で、プラットフォーム定義を作成します。  
プラットフォーム定義は CICSplex SM PLATDEF リソース定義です。この定義で、zFS のプラットフォーム・ホーム・ディレクトリー内のプラットフォーム・バンドルを指し示し、プラットフォームのターゲット CICSplex を指定します。詳しくは、[CICS プラットフォームのデプロイ](#)を参照してください。
6. プラットフォーム・プロジェクト内の領域タイプで作成した CICS 領域定義ごとに、実際の CICS 領域をセットアップします。  
詳しくは、[358 ページの『プラットフォームのデプロイ』](#)を参照してください。
7. CICS Explorer で、プラットフォームを実行する CICSplex にプラットフォーム定義をインストールします。  
CICSplex SM は、プラットフォーム・バンドル内の情報を使用してプラットフォームをターゲット CICSplex にインストールします。プラットフォームと一緒にインストールされる CICS バンドルもインストールされます。詳しくは、[358 ページの『プラットフォームのデプロイ』](#)を参照してください。
8. CICS 領域を開始します。

9. プラットフォームと一緒に CICS バンドルがデプロイされている場合は、CICS Explorer で使用可能にして、プラットフォームで使えるようにします。

## タスクの結果

CICSplex 内で稼働するプラットフォームができました。

## 次のタスク

CICSplex にプラットフォームをセットアップした後、パッケージされたアプリケーションをそこにデプロイできます。詳しくは、[プラットフォームへのアプリケーションのデプロイ](#)を参照してください。環境を制御するためのポリシーをデプロイすることにより、サービス品質をさらに加えることもできます。これらのポリシーは、プラットフォーム内のすべての領域または特定のアプリケーションに適用できます。詳しくは、[CICS ポリシー](#)を参照してください。

プラットフォームの実行について詳しくは、[プラットフォームおよびアプリケーションの管理](#)を参照してください。

## CICS プラットフォームの設計

プラットフォームを作成する前に、既存の CICS 領域をプラットフォームにマップする作業をどのように計画するかや、プラットフォームで使用するためにどのような新しい CICS 領域を特に作成する可能性があるかを明確にしてください。また、プラットフォームにデプロイされるアプリケーションが使用できるようになるサービスについても、検討してください。

ご使用のプラットフォームに関する設計について、以下の項目を考慮してください。

- プラットフォームの一部として採用する既存の CICS 領域とシステム・グループ定義 (CSYSGRP)。
- プラットフォーム内で作成する新しい CICS 領域と領域タイプ。
- プラットフォーム上にデプロイするアプリケーション。
- プラットフォーム上にデプロイするポリシー。
- プラットフォーム上に CICS バンドルとしてデプロイする個別リソース。

プラットフォームを設計してインストールし終えた後に容量要件が増減する場合には、CICS Explorer を使用して、アクティブなプラットフォームに CICS 領域をさらに追加したり、アクティブなプラットフォームから CICS 領域を削除したりできます。

## 領域タイプの設計

領域タイプとは、共通の特性を共有する複数の CICS 領域を一緒に収集し、プラットフォーム内で 1 つの単位として管理できるようにする論理グループ化のことです。グループとして管理する共通の CICS 領域の特性を検索し、それらの特性を使用してプラットフォーム内の CICS 領域を適切な領域タイプに分割します。以下の例のように、領域タイプを使用して、機能的、地理的、または法的要件を満たすように CICS 領域をグループ化することもできます。

- 端末専有領域 (TOR)、ファイル専有領域 (FOR)、アプリケーション専有領域 (AOR): CICS 機能タイプ
- 実動、テスト、開発: ビジネス機能タイプ
- 給与計算、人事、会計: ビジネス処理機能タイプ
- 英国、アジア、ヨーロッパ: 大陸、国、都道府県別の区域

プラットフォーム・アーキテクチャーの例については、[プラットフォームの例](#)を参照してください。

プラットフォーム内に新しい領域タイプを設計して作成することができます。あるいは、プラットフォーム内の領域タイプとして、既存のシステム・グループ定義 (CSYSGRP) を採用できます。単一のプラットフォームは、作成された領域タイプと採用された領域タイプの両方を含むことができます。作成された領域タイプの使用時には、プラットフォーム別にこれらの CICS 領域に関する CICSplex SM トポロジーを作成し、プラットフォームのライフサイクルと関連付けます。採用された領域タイプの使用時には、プラットフォームのライフサイクルとは無関係にこれらの CICS 領域に関する CICSplex SM トポロジーをセットアップして保守します。

プラットフォーム内の1つのCICS領域に複数の目的がある場合は、その領域を複数の領域タイプに共用領域として組み込むことができます。CICS領域を、他のプラットフォーム内の領域タイプと共用することもできます。領域タイプ間でのCICS領域の共用がプラットフォーム設計に含まれている場合は、作成された領域タイプには、作成された領域のみを含めることができ、採用された領域タイプには、採用された領域のみを含めることができることに注意してください。領域の共用について詳しくは、[355 ページの『プラットフォーム内の領域タイプ間でのCICS領域の共用』](#)を参照してください。

CICS TS バージョン 5.1 領域が MAS として CICS TS バージョン 5.1、5.2、または 5.3 CMAS に接続している場合、その領域は、MAS として CICS TS バージョン 5.2 または CICS TS バージョン 5.3 に接続している CICS TS バージョン 5.2 領域や CICS TS バージョン 5.3 領域と一緒にプラットフォームの一部にすることができます。CICS TS バージョン 5.2 または バージョン 5.3 を使用して作成されたアプリケーションを、CICS TS バージョン 5.1 領域が含まれているプラットフォーム上にインストールできます。しかし、CICS TS バージョン 5.2 または バージョン 5.3 アプリケーションをプラットフォーム内の CICS TS バージョン 5.1 領域にインストールすることには、以下の制限があります。

- 専用 PROGRAM リソースや専用 LIBRARY リソースなどの、アプリケーションの専用リソースは CICS TS バージョン 5.1 領域ではサポートされないため、これらの領域内には作成されません。プラットフォーム上に複数のバージョンのアプリケーションをインストールする場合、そのために CICS TS バージョン 5.1 領域でリソース名が競合することがあります。この場合、重複するリソースの CICS TS バージョン 5.1 領域へのインストールは失敗してメッセージ DFHAM4950 または DFHAM4834 が発行され、これらの領域で新しいバージョンのアプリケーションに関する CICS バンドルは使用できなくなります。
- 同じ ID とバージョンの複数の CICS バンドルは、CICS TS バージョン 5.1 領域ではサポートされません。同じ ID とバージョンの CICS バンドルが含まれる複数のバージョンのアプリケーションをプラットフォーム上にインストールする場合、CICS バンドルの CICS TS バージョン 5.1 領域へのインストールは失敗し、この領域からメッセージ DFHAM4952 が発行されます。この場合、アプリケーションは INCOMPLETE 状況になり、CICS TS バージョン 5.1 領域で使用できなくなります。

これらの制限のために、CICS TS バージョン 5.2 または バージョン 5.3 を使用して作成された複数のバージョンのアプリケーションを CICS TS バージョン 5.1 領域が含まれているプラットフォーム上にインストールできても、新しい方のバージョンのアプリケーションの CICS TS バージョン 5.1 領域内へのインストールは失敗する可能性が高いと考えられます。インストール・エラーが発生しないようにするには、複数のバージョンがあるアプリケーションをインストールしようとしている領域タイプ内に CICS TS バージョン 5.2 または バージョン 5.3 領域だけを組み込んでください。

## プラットフォームでの新しい CICS 領域の設計

プラットフォームの設計時に、プラットフォーム上にデプロイしようとしているアプリケーションの要件を正確に満たすために、作成された領域タイプ内に新しい CICS 領域を組み込むことができます。作成された領域を使用して、プラットフォームの一部として採用する計画の既存の CICS 領域を補足する追加の機能を提供できます。または、作成された領域だけで構成されるようにプラットフォームを設計できます。

プラットフォーム設計では、作成された領域タイプごとの CICS 領域に必要な特性を考慮してください。作成された領域タイプでは、それに含まれる CICS 領域のプロパティを指定できます。1つの領域タイプに含まれるすべての CICS 領域の特定の領域属性値は、それらの属性を領域タイプ・レベルで指定することによって複製できます。必須設定値を受け入れることのできる CICS 領域だけを、その領域タイプに含めることができます。CICS 領域内の設定を領域タイプの設定と同じにすることも、CICS 領域内で設定を行わないことも可能です。後者の場合は、領域タイプの設定から提供されます。しかし、CICS 領域内の設定と領域タイプの設定は競合できません。共用される領域は、それらの領域が含まれているすべての領域タイプからの設定を受け入れることができなければなりません。

以下の領域属性値を領域タイプ・レベルで指定することができます。

### ルーティング領域として適格 (WLMSTATUS 属性)

この CICS 領域を、CICS 領域始動時にルーティング領域として関連するワークロードに関連させるかどうかを示します。

### ターゲット領域として適格 (DYNROUTE 属性)

CICS の始動時に、この CICS 領域がターゲット領域としてアクティブになり、ワークロードのターゲットとして作業を受け入れるかどうかを示します。



### **BAS インストールが使用可能 (AUTOINST 属性)**

リソース記述により CICS 領域に関連付けられたリソースを、MAS から CMAS への接続時に自動的にインストールするかどうかを示します。

### **BAS インストール失敗アクション (AINSFAIL 属性)**

BAS インストールに失敗した場合に行うアクションを示します。

使用しているプラットフォームのアーキテクチャーにおいて、ある領域タイプのすべての CICS 領域がそれらのエリアで特定の機能または制限を持つ必要がある場合には、領域タイプのセットアップ時に該当する値を領域タイプ・レベルで指定します。作成された領域タイプに、属性に関する要件が特にならない場合は、CICS 領域ですべての設定が許可されるようにするため、属性に値を指定しないでください。領域属性値を領域タイプ・レベルで指定すると、その属性値はロックされ、その後はその領域タイプに属する CICS 領域で変更できなくなります。

### **既存の CICS 領域のプラットフォームへのマッピング**

既存のシステムを評価して可能な候補をプラットフォームにするには、複数のシステム・グループ定義 (CSYSGRP) と複数の CICS 領域が含まれる最上位グループを検索します。最上位グループはプラットフォームとして再実装するのに適した候補になる可能性があり、CSYSGRP は領域タイプの採用に適した候補になる可能性があります。既存のトポロジーの一部をプラットフォームとしてパッケージ化すると、もっと簡単にリソースやアプリケーションを管理したり CICS 領域にデプロイしたりできます。

プラットフォームの一部として組み込む各 CICS システム・グループ (CSYSGRP) は、以下の要件を満たしている必要があります。

- そのグループは、既にインストールされたプラットフォームによってまだ採用されていない。そのグループが既にプラットフォームに関連付けられている場合、それを領域タイプとして採用することはできません。
- そのグループは、サブグループを含んでいない。
- そのグループの修正が必要にならない (例えば WLM または RTA に関するグループ)。プラットフォームは、領域タイプとして使用されるグループのロックを必要とします。
- グループ内のすべての CICS 領域で CICSplex SM システム・パラメーター MASPLTWAIT(YES) が指定されている。MASPLTWAIT(YES) はビジネス・アプリケーション・サービスにも必要です。このパラメーターは、CICS 領域が初期化される場合、アプリケーションまたはオペレーティング・システムのリソースを自動的にインストールするために必須です。

CSYSGRP 内にこれらの要件を満たしていない CICS 領域がある場合、特にその CICS 領域をプラットフォームの一部として使用する必要があるときは、それらの領域のシステム定義 (CSYSDEF) をそのプラットフォーム専用で作成した新しい CSYSGRP に追加してください。

### **対象となるプラットフォームにおいて CICSplex SM データ・リポジトリ (EYUDREP) が十分大きいことの確認**

大規模なデプロイメントを計画している場合、CICSplex SM データ・リポジトリのサイズが十分であることを確認してください。データ・リポジトリは、CICSplex SM のシステム構成や定義データが保管される VSAM データ・セットです。各 CMAS には独自のデータ・リポジトリがあります。

プラットフォームやアプリケーションのリソースは、CICS CSD からではなく、データ・リポジトリから管理されます。そのため、データ・リポジトリには、プラットフォームの定義とそこにデプロイされるアプリケーションやバンドルのための十分なスペースが必要です。データ・リポジトリの現行サイズは、IDCAMS ユーティリティの LISTCAT 機能を使用することによって判別できます。

データ・リポジトリを拡張する場合、IDCAMS ユーティリティの REPRO 機能を使用します。この JCL の例は、CICSTS51.CPSM.SEYUSAMP ライブラリーの EYUJXDRP メンバーにあります。JCL 内の RECORDS(xx,yy) ステートメント上で、自分の環境にとって適切な 1 次 (xx) および 2 次 (yy) の値を指定します。初期値は 500 と 3000 です。

## プラットフォーム内の領域タイプ間での CICS 領域の共用

領域タイプ間で領域を共用すると、構成の価値を上げることができ、プラットフォーム内のセットアップが簡単になります。それぞれを分離したままにする場合より考慮事項は増えますが、計画の負荷を軽減できます。

CICS 領域を共用する理由はさまざまです。例えば、開発 CICSplex 上のプラットフォームで、すべての領域タイプで共用される単一の CICS 領域を使用することもできます。すべての領域タイプで共用される単一の CICS 領域を使用すると、複数の CICS 領域の構成、管理、保守が不要になるので、新規開発が簡素化されます。複数のプラットフォームが TOR の単一セットか 1 つの FOR を共用している実稼働環境で、CICS 領域を共用することもできます。

プラットフォームのセットアップ中か、インストール済みのアクティブなプラットフォームで、CICS 領域を領域タイプ間で共用できます。作成された領域タイプには、作成された CICS 領域のみを含めることができ、採用された領域タイプには、採用された CICS 領域のみを含めることができます。

作成された領域タイプ内で、すべての CICS 領域の特定の領域属性値を、プラットフォーム・プロジェクト内でそれらの属性を領域タイプ・レベルで指定することによって複製できます。共用される CICS 領域は、それらの領域が含まれているすべての領域タイプからの設定を受け入れることができなければなりません。例えば、CICS 領域がワークロード管理に関するルーティング領域として適格でなければならないと指定している領域タイプを作成することがあります。この領域タイプは、CICS 領域がルーティング領域として不適格である必要がある領域タイプと、CICS 領域を共用できません。しかし、同様に CICS 領域がルーティング領域として適格である必要がある領域タイプや、領域タイプ・レベルで設定が指定されていない領域タイプとは、CICS 領域を共用できます。

1 つのプラットフォーム内と複数のプラットフォーム間のどちらで CICS 領域を共用していても、特定の CICS リソース間で競合が存在することのないように注意しなければなりません。プラットフォームやアプリケーションのバンドル・インストールでは、CICS バンドル・レベルでの競合のみ検査されます。2 種類のバンドルを同じ領域にインストールし、両方とも同じリソースを作成する場合、バンドルの重複部分は使用できなくなります。

複数のプラットフォームが CICS 領域の 1 つのグループを共用しており、両方のプラットフォームで同じリソースをインストールする必要がある場合、一方のプラットフォームのみこれらのバンドルのインストールを試行する必要があります。最初にこのプラットフォームのインストールが行われてから、破棄される必要があります。両方のプラットフォームとも同じバンドルのセットをインストールできますが、この方法の結果は不確実です。プラットフォームのインストール時に、2 つ目のプラットフォームに関するバンドルがインストールされなかったことが示されます。2 つ目のプラットフォームでは、プラットフォームの ENABLESTATUS が INCOMPLETE になり、バンドル・インストールが失敗したことが共用領域に記録されます。この 2 番目の方法を採用する場合は、プラットフォームの破棄に注意してください。一方のプラットフォームにしかバンドルがインストールされないの、もう一方のプラットフォームに、機能に必要なバンドルがインストールされていることを確認する必要があります。この技法の使用時の最悪のシナリオは、両方のプラットフォームが同時にインストールされることです。この場合、各プラットフォームでバンドルのサブセットがインストールされ、両方のプラットフォームともプラットフォームの ENABLESTATUS が INCOMPLETE になります。しかし、このシナリオでも、これらのプラットフォームは破棄時に残っているすべてのバンドルを正しくクリーンアップします。

## プラットフォーム用の zFS の準備

プラットフォームを作成してデプロイする前に、zFS でプラットフォーム・ホーム・ディレクトリーを構成する必要があります。専用ファイル・システムの作成、ファイル・システム・セキュリティのセットアップ、CICS Explorer からアクセスするための FTP セキュリティのセットアップを行います。

### 始める前に

プラットフォームの概要については、**仕組み: プラットフォーム**を参照してください。このタスクは、プラットフォームをどのような構造にするかを決定していることを前提としています。(まだの場合は、を参照してください。)

## このタスクについて

ディレクトリーの概要については、[z/OS UNIX のプラットフォーム・ディレクトリーの構造](#)を参照してください。プラットフォーム・ホーム・ディレクトリーでは、さまざまなプラットフォーム・リソースやアプリケーション・リソースがいくつかのサブディレクトリーで保持されます。これらのサブディレクトリーは、CICS Explorer からプラットフォームがエクスポートされるときに作成されます。

## 手順

1. zFS プラットフォーム・ホーム・ディレクトリーとして使用する z/OS UNIX ファイル・システム・データ・セットを作成します。

これは、プラットフォーム内ですべての CICS 領域によって使用される専用ファイル・システムです。デフォルトのプラットフォーム・ホーム・ディレクトリーは `/var/cicsts/CICSplex/platform1/` です (CICSplex はプラットフォームがインストールされる CICSplex の名前、platform1 はプラットフォームの名前)。

ベスト・プラクティスは、このデフォルトを保持することです。別のディレクトリーをプラットフォーム・ホーム・ディレクトリーとして使用する場合は、CICS プラットフォーム・プロジェクトの作成後に、プラットフォーム・バンドルを変更して代替のディレクトリー名を指定する必要があります。これは、CICS Explorer プラットフォーム記述子エディターで行います。

- a) 非共有 zFS を使用する場合は、読み取り/書き込みファイル・システムとして、データ・セットを `/var` に `/var/cicsts` としてマウントします。
- b) 多重システム (LPAR) 環境で共有ファイル・システムを使用する場合、データ・セットをルート・ファイル・システム (`/`) に `/cicsts` としてマウントします。また、アクセスが必要なシステムごとに、`/var/cicsts` から `/cicsts` 共有ディレクトリーへのシンボリック・リンクを作成します。
- c) 多重システム環境かシスプレックス間環境で、すべてのシステム間でファイル・システムを共有できない場合、プラットフォーム・ホーム・ディレクトリー用にセットアップした構造を各 zFS ファイル・システム内で複製します。プラットフォーム、アプリケーション、または CICS バンドルをエクスポートする際には、必ずプラットフォーム・ホーム・ディレクトリーのコンテンツが各 zFS ファイル・システムに複製されていることを確認します。CICS Explorer でエクスポート・プロセスを繰り返したり、個々のファイル・システムごとに該当する z/OS 接続を選択したりできます。
- d) ディレクトリーが既存でない場合は、`/var/cicsts/CICSplex` と `/var/cicsts/CICSplex/platform1` のサブディレクトリーを作成します。

CICS Explorer を使用する場合、これらのディレクトリーが自動的に作成されます。

2. ファイル・システム・セキュリティをセットアップします。

このファイル・システム・セキュリティによって、CICSplex SM CMAS 領域を含む、プラットフォーム内のすべての CICS 領域が、プラットフォーム・ホーム・ディレクトリー内のバンドル・ファイルを読み取れるようになります。

- a) `/var/cicsts` のディレクトリー所有者を、バンドル・ファイルの作成に使用されるユーザー ID に変更します。
- b) `/var/cicsts` のディレクトリーのグループ所有権を、プラットフォーム内のすべての CICS 領域が属するグループに変更します。
- c) ディレクトリーの所有者に読み取り、書き込み、実行の許可を付与します。また、前述のグループに読み取りと実行の許可を付与します。  
例えば、`rwxr-x---` とします。
- d) オプション: 複数の管理者ユーザー ID が書き込み権限を必要とする場合、または異なる複数のグループが読み取り権限を必要とする場合、UNIX システム・サービス (USS) のアクセス制御リスト (ACL) エントリーを使用すると、グループ許可または所有者許可を追加できます。これは、**FSSEC** リソース・クラスをアクティブにし、**setfacl** コマンドを使用することによって、実現できます。

3. FTP セキュリティをセットアップします。

このレベルのセキュリティによって、CICS Explorer からエクスポートされたバンドルが zFS 上のプラットフォーム・ホーム・ディレクトリーに書き込まれ、またプラットフォーム内のすべての CICS 領域から読み取られるようになります。

- a) z/OS FTP デーモンのファイル・モード作成マスクを設定し、所有者が書き込み許可を持ち、グループが読み取り許可を持つようにします。  
この構成を行うには、FTP.DATA 構成ファイル内で **UMASK** ステートメントを使用します。
- b) オプション: ACL エントリーも使用してセキュリティを制御している場合は、デフォルト ACL が zFS プラットフォーム・ホーム・ディレクトリー (/var/cicsts/CICSplex/platform1 など。  
CICSplex は CICSplex の名前、platform1 はプラットフォームの名前) から継承されるようにします。

### タスクの結果

zFS 環境が正しいディレクトリーと許可で構成されました。デプロイメント・プロセスの一部として CICS Explorer から zFS にプラットフォームをエクスポートするときに、追加のディレクトリーが作成されます。

### 次のタスク

357 ページの『プラットフォームの作成』の手順に従ってプラットフォーム・バンドルを作成できるようになりました。

## プラットフォームの作成

CICS Explorer でプラットフォームを作成します。CICS プラットフォーム・プロジェクトを作成して、プラットフォーム・バンドルを定義します。プラットフォーム・バンドルは、プラットフォームについて記述する一種の管理バンドルです。プラットフォーム・バンドルは、プラットフォームの領域タイプを指定します。

### 始める前に

プラットフォームの概要については、[仕組み: プラットフォーム](#)を参照してください。このタスクは、以下の項目が既に完了していることを前提としています。

1. プラットフォームをどのような構造にするかを決定した。まだの場合は、[プラットフォームの構造](#)を参照してください。
2. zFS にプラットフォーム・ホーム・ディレクトリーをセットアップした。まだの場合は、[プラットフォーム用の zFS の準備](#)を参照してください。

ご使用のプラットフォームのレベルで CICS バンドルをデプロイする場合は、CICS プラットフォーム・プロジェクトの作成時に追加する準備が整った CICS バンドルを作成できます。例えば、プラットフォーム内のすべての CICS 領域で必要とされるリソースや、プラットフォームにデプロイされる複数のアプリケーションに適用されるポリシーを入れた CICS バンドルをデプロイできます。プラットフォーム用のアプリケーション、リソース、またはポリシーがまだセットアップされていない場合、プラットフォームの作成後またはデプロイ後の任意の時点で CICS バンドルをプラットフォームに追加できます。CICS バンドルの作成については、[CICS バンドルの定義](#)を参照してください。

### このタスクについて

CICS Explorer でプラットフォームを作成します。以下のステップは、手順の概要を示しています。詳細なステップについては、[CICS Explorer 製品資料内の『プラットフォームとアプリケーションの処理』](#)を参照してください。

### 手順

1. CICS プラットフォーム・プロジェクトを作成します。プロジェクトの名前と、プラットフォーム自体の名前と説明を入力します。  
プロジェクトの場所は、ローカル・ワークスペース内のどこに CICS プラットフォーム・プロジェクトが保管されるかを指定します。
2. プラットフォームの 1 つ以上の領域タイプを追加します。各領域タイプに名前を付けてから、新しいシステム・グループを使用する作成された領域タイプか、既存のシステム・グループを使用する採用された領域タイプかを指定します。作成された領域タイプの場合、領域タイプ用に作成される CICS システム・グループ (CSYSGRP) の名前を入力します。  
採用された領域タイプとして既存の CICS システム・グループを追加するには、CICSplex SM に対する接続がなければなりません。



3. オプション: プラットフォームと共にデプロイする CICS バンドルを指定してから、デプロイ先の領域タイプを選択します。
- プラットフォームと共にデプロイする準備のできた CICS バンドルがない場合は、この段階をスキップしてください。
4. CICS Explorer のプラットフォーム記述子エディターを使用して CICS プラットフォーム・プロジェクトを編集し、プラットフォーム・バンドルの指定を確認して完成させます。
- プラットフォーム記述子エディターは、プラットフォーム・プロジェクトの作成後に自動的に開きます。後でプラットフォーム記述子エディターを開くときには、プラットフォーム・バンドルのいずれかの .xml ファイル (ただし manifest.xml ファイルを除く) をダブルクリックしてください。
- a) デフォルトのプラットフォーム・ホーム・ディレクトリーの代わりに別のディレクトリーを使用する必要がある場合は、既にセットアップ済みのホーム・ディレクトリーを参照して、それをプラットフォーム・ホーム・ディレクトリーとして指定します。
  - b) 領域タイプを確認し、必要に応じて、作成された領域タイプや採用された領域タイプを追加したり削除したりします。
- プラットフォームがインストールされてアクティブになった後、領域タイプ内で個々の CICS 領域を追加したり削除したりできます。しかし、インストールされたプラットフォーム内で領域タイプを変更することはできないため、プラットフォームのインストール前に領域タイプを確定してください。
- c) プラットフォーム内の作成された領域タイプごとに、その領域タイプ内のすべての CICS 領域内で適用しなければならない設定を指定します。
- 設定値を受け入れることのできる CICS 領域だけを、その領域タイプに含めることができます。作成された領域タイプに、属性に関する要件が特にならない場合は、CICS 領域ですべての設定が許可されるようにするため、属性に値を指定しないでください。
- d) プラットフォーム内の作成された領域タイプごとに、その領域タイプ内の一部になる CICS 領域に関する 1 つ以上の CICS 領域定義を追加します。デフォルトの CICS 領域定義が指定されています。これを作成された領域タイプ内の CICS 領域に関する実際の定義に置き換える必要があります。
- CICS 領域ごとの基本プロパティーと、その領域の組み込み先の作成された領域タイプを指定します。
- e) プラットフォームと共にデプロイされるすべての CICS バンドルを確認します。必要に応じてバンドルを追加したり削除したりし、各 CICS バンドルのデプロイ先の領域タイプを確認したり変更したりします。
- プラットフォームをインストールした後、アプリケーションやポリシーを開発していく時点で、プラットフォーム・レベルで他の CICS バンドルをデプロイすることができます。

## 次のタスク

CICS プラットフォーム・プロジェクトを zFS 上のプラットフォーム・ホーム・ディレクトリーにエクスポートし、プラットフォーム内の領域タイプで作成したそれぞれの CICS 領域定義に一致するように CICS 領域をセットアップしてから、CICSplex でプラットフォームをインストールして使用可能にします。説明は、[プラットフォームのデプロイ](#)を参照してください。

## プラットフォームのデプロイ

プラットフォームをデプロイするには、CICS Explorer から zFS 内のプラットフォーム・ホーム・ディレクトリーに CICS プラットフォーム・プロジェクトをエクスポートし、プラットフォーム内で作成したそれぞれの CICS 領域定義に一致するよう CICS 領域をセットアップし、CICSplex SM でプラットフォーム定義 (PLATDEF) を作成してインストールします。

### 始める前に

プラットフォームの概要については、[仕組み: プラットフォーム](#)を参照してください。このタスクは、以下の項目が既に完了していることを前提としています。

- プラットフォームをどのような構造にするかを決定した。まだの場合は、[プラットフォームのインストール](#)を参照してください。
- zFS にプラットフォーム・ホーム・ディレクトリーをセットアップした。まだの場合は、[プラットフォーム用の zFS の準備](#)を参照してください。

- CICS Explorer で CICS プラットフォーム・プロジェクトを作成した。まだの場合は、[CICS プラットフォームの作成を参照してください](#)。

また、CICSplex SM データ・リポジトリのサイズが十分であることも確認してください。[354 ページの『対象となるプラットフォームにおいて CICSplex SM データ・リポジトリ \(EYUDREP\) が十分大きいことの確認』](#)を参照してください。

## このタスクについて

CICS Explorer でプラットフォームをデプロイしますが、CICS Explorer の外部でいくつかの追加ステップを実行することによって、CICS 領域をセットアップして開始し、CICSplex SM に認識させる必要があります。以下のステップは、手順の概要を示しています。詳細なステップについては、[CICS Explorer 製品資料内の『プラットフォームとアプリケーションの処理』](#)を参照してください。

## 手順

1. CICS Explorer の CICS クラウド・パースペクティブを使用し、CICS Explorer から zFS 内のプラットフォーム・ホーム・ディレクトリーに CICS プラットフォーム・プロジェクトをエクスポートします。  
エクスポート処理では、CICS プラットフォーム・プロジェクトで参照される CICS バンドルをパッケージ化し、プラットフォーム・バンドルのすべてのファイルと CICS バンドルを zFS のプラットフォーム・ホーム・ディレクトリーにエクスポートします。エクスポート中に、ウィザードはプラットフォーム・ホーム・ディレクトリーのサブディレクトリーの有無を検査し、存在しない場合はそれを作成します。プラットフォーム・ディレクトリーの構造について詳しくは、[z/OS UNIX のプラットフォーム・ディレクトリーの構造を参照してください](#)。
2. CICS Explorer を使用してプラットフォーム定義 (PLATDEF) を作成します。この定義で、zFS 内のプラットフォーム・ホーム・ディレクトリーを指し示し、プラットフォームのターゲット CICSplex を指定します。  
プラットフォーム・エクスポート・ウィザード内のボックスをチェックすると、プラットフォーム・プロジェクトをエクスポートした後すぐにプラットフォーム定義を作成する選択が可能です。別の時点でプラットフォーム定義を作成するには、CICS Explorer の「新規プラットフォーム定義 (New Platform Definition)」ウィザードを使用します。  
CICSplex SMCMA のデータ・リポジトリ内にプラットフォーム定義が作成されます。
3. プラットフォーム・プロジェクト内の領域タイプで作成した CICS 領域定義ごとに、実際の CICS 領域をセットアップします。新しい CICS 領域をセットアップする代わりに、まだ CICSplex SM によって管理されていない既存の CICS 領域を使用することもできます。
  - a) 作成したシステム定義に一致する APPLID、SYSID、その他の属性を使用して CICS 領域を作成します。ベスト・プラクティスとして、1 つの領域タイプ内の各 CICS 領域が互いの複製になるようにするのが適切です。これを行う手順については、[CICS 領域のセットアップを参照してください](#)。
  - b) CICS 領域に対して一連の手順を実行し、これを管理対象アプリケーション・システム (MAS) として CICSplex SM に認識させます。これを行う方法については、[Setting up a CICSplex SM managed application system \(MAS\)](#) を参照してください。

この時点では、まだ CICS 領域を開始しないでください。
4. CICS Explorer を使用して、プラットフォーム定義をインストールします。これにより、ターゲット CICSplex 内にプラットフォームが作成されます。  
プラットフォーム定義をインストールすると、CICSplex SM は、そのプラットフォームを表す PLATFORM リソースを CICSplex 内に作成します。さらに、CICSplex SM は、そのプラットフォームに関するレコードをデータ・リポジトリ内に作成します。このレコードは、プラットフォームのバンドルのリカバリー処理で使用されます。
5. 作成された領域タイプの CICS 領域の場合、この時点で通常の方式を使用して領域を始動してから、CICS Explorer の「クラウド・エクスプローラー」ビューをリフレッシュして、プラットフォームの状況が現在 ACTIVE になっていることを確認します。
6. プラットフォームと一緒にデプロイする CICS バンドルがある場合は、プラットフォームを右クリックし、「使用可能」をクリックします。



## タスクの結果

プラットフォームがインストールされ、使用可能になりました。

## 次のタスク

パッケージされたアプリケーションをプラットフォームにデプロイできます。詳しくは、[プラットフォームへのアプリケーションのデプロイ](#)を参照してください。環境を制御するためのポリシーをデプロイすることにより、サービス品質をさらに加えることもできます。詳しくは、[CICS ポリシー](#)を参照してください。

プラットフォームの実行について詳しくは、[プラットフォームおよびアプリケーションの管理](#)を参照してください。

## 第9章 CMCI のセットアップ

CICS Explorer などの HTTP システム 管理クライアントから CICS 領域を構成して管理するには、CICS 環境に CICS 管理クライアント・インターフェース (CMCI) をセットアップする必要があります。

### このタスクについて

CICSplex SM 環境またはスタンドアロン CICS 領域 (SMSS) で CMCI をセットアップできます。

CICSplex SM で CMCI を使用する場合、ユーザーは CICS Explorer などの HTTP クライアントで、CICSplex SM によって管理されているすべての CICS 領域内の定義リソース、操作リソース、および CSD リソースを管理できます。

スタンドアロン CICS 領域 (SMSS) で CMCI を使用する場合、ユーザーは CICS Explorer などの HTTP クライアントで、その領域に関連付けられた操作リソースと CSD リソースのみ管理できます。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。

## CICSplex SM での CMCI のセットアップ

CICS 管理クライアント・インターフェース (CMCI) を CICSplex SM 環境にインストールするには、WUI 領域に CMCI を構成する必要があります。

### CMCI セットアップの計画

1. 高可用性のために、複数の CICSplex SM WUI 領域にセットアップできます。
2. CMCI で多要素認証 (MFA) や CMCI GraphQL API 機能などの拡張クライアント認証を使用するには、CMCI で CMCI JVM サーバーを使用する必要があります。CICS バンドル・デプロイメント API を介したバンドルのデプロイメントを有効にするには、[371 ページの『CICS バンドル・デプロイメント API 用の CMCI JVM サーバーの構成』](#)の説明に従って追加の構成を実行する必要があります。
3. ご使用の環境に基づいて、[CMCI のストレージ要件を見積もってください](#)。

注: CMCI インターフェースは、デフォルトで CMCI JVM サーバーを使用します。この CMCI 構成手順は、CMCI が CMCI JVM サーバーを使用することを前提としています。CMCI で CMCI JVM サーバーを使用しない場合は、機能切り替え `com.ibm.cics.cmci.jvmserver=false` を設定して CMCI JVM サーバーをオフに切り替え、[CICS TS 5.4 製品情報内の『CICSplex SM での CMCI のセットアップ』](#)の指示に従って CMCI JVM サーバーを使用しないように CMCI をセットアップできます。ただし、この機能切り替えは CICS TS の将来のリリースでは除去されるので、できるだけ早く CMCI JVM サーバーにアップグレードすることを強くお勧めします。

### 始める前に

1 つ以上の CICSplex SM WUI 領域が構成されている必要があります。CICSplex SM WUI 領域をセットアップするには、[CICSplex SM Web ユーザー・インターフェース・サーバーの設定](#)に従ってください。

### セットアップ・ガイド

**シナリオ 1: CMCI が構成されていない WUI 領域に CMCI をセットアップする**

[362 ページの『WUI 領域の CMCI の構成』](#)の説明に従ってください。

**シナリオ 2: WUI 領域に CMCI が既にセットアップされているが、CMCI JVM サーバーは使用されていない。CMCI JVM サーバーを使用するように WUI 領域をアップグレードすることを検討している。**

詳しくは、[367 ページの『CMCI JVM サーバーを使用するための WUI 領域の構成』](#)を参照してください。

**シナリオ 3: CICSplex 内で複数の CMCI JVM サーバーを実行する**

CICSplex 内で複数の CMCI JVM サーバーを実行する場合は、Liberty でシングル・サインオン (SSO) サポートを構成すると、1 つの CMCI JVM サーバーで HTTP クライアント・ユーザーの認証を 1 回行うだ

けで、同じ CICSplex 内の他の CMCI JVM サーバーに再認証なしでアクセスできるようになります。詳しくは、[373 ページの『CICSplex での複数の CMCI JVM サーバーのセットアップ』](#)を参照してください。

**シナリオ 4: CMCI JVM サーバーを使用するようにセットアップされた WUI 領域があり、CICS バンドル・デプロイメント API を使用する。**

CMCI JVM サーバーで API を使用するように追加構成を実行する必要があります。詳しくは、[371 ページの『CICS バンドル・デプロイメント API 用の CMCI JVM サーバーの構成』](#)を参照してください。

## WUI 領域の CMCI の構成

WUI 領域に CMCI がない場合は、その WUI 領域に CMCI JVM サーバー を使用して CMCI をセットアップできます。CMCI JVM サーバーは、拡張クライアント認証のサポート、GraphQL API のサポート、および CICS バンドル・デプロイメント API のサポートを提供する Liberty サーバーです。

### 始める前に

#### CMCI JVM サーバーのシステム要件

1. 必要な Java コンポーネントがすべてインストールされていることを確認してください。[Java コンポーネントのチェックリスト](#)に従ってください。
2. 領域に Java サポートがセットアップされていることを確認してください。詳しくは、[Setting up Java support](#) を参照してください。

#### 接続で多要素認証 (MFA) 資格情報を使用できるようにするための追加要件

- 多要素認証をサポートするには、IBM Multi-Factor Authentication for z/OS、または RACF が構成された同等の製品が必要です。代わりに外部セキュリティ・マネージャー (ESM) を使用する場合、詳細についてベンダーに問い合わせてください。
- CMCI JVM サーバーの実行場所となる領域と、このサーバーの接続先 CMAS が同じ CICS レベルであることを確認します。CICSplex SM トポロジーのセットアップに関する CICS レベルの考慮事項については、[Designing your CICSplex SM environment](#) を参照してください。

### 手順

1. 領域内の 64 ビット (2 GB 境界より上) ストレージと補助記憶域の適切なレベルを以下のように指定します。
  - 領域の 64 ビット・ストレージに制限を設定するには、z/OS **MEMLIMIT** パラメーターを使用します。
  - CICSplex SM システム・パラメーターの **MAXAUXCPSM** と **MAXAUXTOTL** を使用して、CMAS の補助記憶域を設定します。

発生する可能性のあるストレージの問題を回避するためのこれらの値の設定については、[381 ページの『CMCI のストレージ要件の見積もり』](#)を参照してください。

2. CSD 上の CICS リソース (例えば、プログラム、ファイル、トランザクションなど) を CICS Explorer が照会できるように、CICS システム初期設定パラメーター **CSDSTRNO** を少なくとも 4 にしておきます (CSDSTRNO=4)。

**CSDSTRNO** が 4 より小さい場合、要求は CNX0591E RESP=CSDERR RESP2=5 (十分でない VSAM スtring) で失敗することがあります。

3. **KEYRING**、**NISTSP800131A**、および **SEC** システム初期設定パラメーターの値を確認します。

これらのシステム初期設定パラメーターは、**APPLID** および **DFLTUSER** と一緒に、CMCI JVM サーバー構成パラメーターにマップされます。場合によっては CMCI JVM サーバーで追加構成が必要になる可能性があるということに注意してください。詳しくは、[373 ページの『CICSplex SM WUI サーバーと CMCI JVM サーバーの間の構成パラメーターのマッピング』](#)を参照してください。

4. WUI サーバー初期設定パラメーターを指定して、CICSplex SM で CMCI を使用できるようにします。これらのパラメーターとしては、CMCI オプション、TCP/IP オプション、環境オプションなどがあります。

WUI サーバー初期設定パラメーターを指定するときは、以下のことを考慮してください。

- **CMCI**PORT パラメーターに固有値を指定する必要があります。このパラメーターは CMCI に TCP/IP ポート番号を割り振ります。**CMCI**PORT パラメーターに値を設定することで、CMCI が WUI 領域に確実にインストールされます。CMCI では、Web ユーザー・インターフェースに対して別のポートを使用する必要があります。
- デフォルトでは、URIMAP と TCPIP SERVICE のリソース定義は、**SEC CICS** システム初期設定パラメーターと **TCPIPSSL** WUI サーバーの初期化パラメーターから派生するセキュリティー設定とともに自動インストールされます。オプションの **CMCIAUTH** および **CMCISSL** パラメーターを使用してデフォルトの CMCI TCPIP SERVICE 設定をオーバーライドすることにより、セキュリティー強化のために SSL 認証を使用可能にすることができます。WUI のセキュリティーのセットアップについて詳しくは、[CICSplex SM Web User Interface security access overview](#) を参照してください。
- WUI サーバーの初期化パラメーター **DEFAULTWARNCNT** にゼロ以外の値を設定することを検討してください。このパラメーターに適切な値を設定することで、受け入れられない量のデータが取り出されることはなくなり、また CICS リソースに対して要求を出すときに長時間待ちおよび潜在的なストレージ問題を回避できます。CMCI での警告カウント・メカニズムについては、[377 ページの『CMCI でのレコード・カウント警告』](#)を参照してください。

詳しい説明は、[Web ユーザー・インターフェース・サーバーの初期設定パラメーター](#)を参照してください。

注：CMCI オプションや TCP/IP オプションなどの WUI サーバー初期設定パラメーターが CMCI JVM サーバー構成パラメーターにどのようにマップされるかを知ることは役に立ちます。CMCI JVM サーバーと互換性がない値もあることに注意してください。詳しくは、[373 ページの『CICSplex SM WUI サーバーと CMCI JVM サーバーの間の構成パラメーターのマッピング』](#)を参照してください。

5. リソース定義グループ DFHWU がインストールされていることを確認します。

グループ DFHWU には、CMCI 環境にとって必要なリソース定義が含まれています。

#### CMCI JVM サーバー の構成

6. 以下のシステム初期設定パラメーターを領域に追加します。

- **JVMPROFILEDIR**
- **START=INITIAL**
- **EDSALIM**

7. CMCI JVM サーバーの JVM プロファイルを作成します。

a) /usr/lpp/cics56/JVMProfiles にある EYUCMCIJ.jvmprofile を、**JVMPROFILEDIR** システム初期設定パラメーターで指定された場所にコピーします。

b) JVM プロファイルの **JAVA\_HOME** および **WORK\_DIR** の値を検証または更新します。

詳しくは、[JVM プロファイルで使用されるシンボル](#)を参照してください。

c) 以下の行を JVM プロファイルに追加することで、WUI 領域の SAF プロファイル接頭部を指定します。

SAF プロファイル接頭部は各 WUI 領域で使用される RACF ルールを決定して、同じ SAF プロファイル接頭部を持つ複数の WUI 領域で同じセキュリティー構成が共用されるようにします。例えば、CICS バンドル・デプロイメント API のユーザーを認証する場合に、同じ SAF プロファイル接頭部を持つ複数の WUI 領域内の CMCI JVM サーバーでは、同じユーザーが API にアクセスすることが許可されます。

```
-Dcom.ibm.cics.jvmserver.wlp.saf.profilePrefix=${MYPREFIX}
```

ここで、**\${MYPREFIX}** は、セキュリティー構成を共用する必要がある WUI 領域の SAF プロファイル接頭部です。

8. CMCI JVM サーバー のストレージ要件を設定します。

付属の EYUCMCIJ.jvmprofile ファイルでは共用ライブラリー領域の使用が無効になっており、必要とされる非 CICS 31 ビット・ストレージの量が削減されます。デフォルトの場合、CMCI JVM サーバ

一用に自動的に作成される JVMSERVER リソースでは、**THREADLIMIT** 属性の値が 15 になっています。したがって、ストレージ要件の初期見積もりとしては以下の値を使用できます。

- 24 ビット・ストレージ: 512 KB
- 31 ビット・ストレージ: 100 MB

引き続き、[JVM サーバーのストレージ要件の計算](#)の説明に従ってストレージ要件をモニターし、確認します。

#### 9. CMCI JVM サーバー用の Liberty エンジェル・プロセス開始タスクを構成します。

- a) Liberty エンジェル・プロセスが実行されていない場合は、[Liberty サーバー・エンジェル・プロセス](#)のステップに従ってそれを作成します。
- b) 既に Liberty エンジェル・プロセスが実行されている場合は、エンジェル JCL の ROOT シンボリック・パラメーターで指定された Liberty のバージョンが、CICS で提供される Liberty のバージョン以上のレベルであることを確認します。

##### 例: 開始タスクのシステム・ログから Liberty のバージョンを確認する

Liberty エンジェル・プロセスが Liberty 18.0.0.2 以上で実行されている場合、開始タスクのシステム・ログには、Liberty のバージョンを示す次のようなメッセージが含まれています。

```
CWWKB0079I THE ANGEL BUILD LEVEL IS 18.0.0.2 20180619-0654 2018.7.0.0 20180619-0654
```

##### 例: メッセージ DFHSJ1405I から Liberty のバージョンを確認する

CICS JVM サーバーで実行されている Liberty のバージョンが、次のメッセージで示されます。

```
DFHSJ1405I 08/22/2018 17:04:39 IYK3ZDRI JVMSERVER EYUCMCIJ is running WebSphere Application Server  
Version 18.0.0.2 Liberty - (18.0.0.2-cl180220180619-0403) process ID  
67174497.
```

##### 例: スクリプトを実行して Liberty のバージョンを確認する

エンジェル JCL が次の ROOT パラメーターを指定しているとします。

```
// SET ROOT='/usr/lpp/zosmf/wlp'
```

Liberty のバージョンを調べるには、次のスクリプトを実行します。

```
/usr/lpp/zosmf/wlp/bin/productInfo version --verbose
```

CICS の場合は、次のスクリプトを実行します。

```
/usr/lpp/cicsts56/wlp/bin/productInfo version --verbose
```

```

WebSphereApplicationServer.properties:
com.ibm.websphere.productId=com.ibm.websphere.appserver
com.ibm.websphere.productOwner=IBM
com.ibm.websphere.productVersion=16.0.0.3
com.ibm.websphere.productName=WebSphere Application Server
com.ibm.websphere.productInstallType=Archive
com.ibm.websphere.productEdition=zOS
com.ibm.websphere.productLicenseType=IPLA

WebSphereApplicationServerZOS.properties:
com.ibm.websphere.productId=com.ibm.websphere.appserver.zos
com.ibm.websphere.productOwner=IBM CORP
com.ibm.websphere.productVersion=16.0.0.3 <== Liberty Version
com.ibm.websphere.productName=WAS FOR Z/OS
com.ibm.websphere.productPID=5655-WAS
com.ibm.websphere.productQualifier=WAS Z/OS
com.ibm.websphere.productReplaces=com.ibm.websphere.appserver
com.ibm.websphere.productEdition=
com.ibm.websphere.gssp=true

zOSMF.properties:
com.ibm.websphere.productId=com.ibm.zosmf
com.ibm.websphere.productOwner=IBM
com.ibm.websphere.productVersion=2.2.0
com.ibm.websphere.productName=z/OSMF
com.ibm.websphere.productPID=5650-ZOS
com.ibm.websphere.productQualifier=z/OSMF
com.ibm.websphere.productReplaces=com.ibm.websphere.appserver.zos
com.ibm.websphere.productEdition=N/A

```

図 59. 出力例

c) Liberty のバージョンが CICS で提供される Liberty のバージョンより低いレベルの場合は、CICS Liberty インストールを使用して、指定されたエンジェル・プロセスを構成します。

1) CMCI JVM サーバーの JVM プロファイル EYUCMCIJ.jvmprofile 内の次の行のコメントを外します。

```
#-Dcom.ibm.ws.zos.core.angelName=<named_angel>
```

2) -Dcom.ibm.ws.zos.core.angelName プロパティにエンジェル名を指定します。

このプロパティは、指定されたエンジェル・プロセスに CMCI JVM サーバーが接続できるようにします。詳しくは、[Liberty サーバー・エンジェル・プロセス](#)を参照してください。

d) 領域を開始する前に、Liberty エンジェル・プロセスの準備ができていることを確認します。

10. Liberty エンジェル・プロセスを使用するように WUI 領域のセキュリティを構成します。

RACF を使用している場合は、以下のように、SEYUSAMP に含まれるサンプル CLIST EYU\$ANGL を使用することにより、WUI 領域で Liberty エンジェル・プロセスを使用するための RACF 定義を作成することができます。

a) SEYUSAMP 内の CLIST EYU\$ANGL のコピーをとります。

b) 以下の変数を指定することにより、コピーを更新します。

#### WUI\_REGION\_USERID

どのユーザー ID で WUI 領域が実行されるかを指定します。

#### ANGEL\_NAME

名前付きエンジェル・プロセスを使用している場合、値は ANGEL.name です (name は -Dcom.ibm.ws.zos.core.angelName プロパティの値)。

名前エンジェル・プロセスを使用していない (-Dcom.ibm.ws.zos.core.angelName が指定されていない) 場合、値は ANGEL です。

c) CLIST を実行します。

RACF 以外の外部セキュリティ・マネージャーを使用している場合の手順については、外部製品の資料を参照してください。



11. CMCI JVM サーバーから発生したタスクは、デフォルトで CJSU トランザクションで実行されます。WUI 領域でトランザクションのセキュリティがアクティブの場合は、CJSU トランザクションに CICS デフォルト・ユーザー・アクセスを付与します。

あるいは、CICS のデフォルト・ユーザーに基づいて、CJSU トランザクションに対する追加のアクセス権限が付与された新規ユーザー ID を作成することもできます。そのユーザー ID を、`com.ibm.cics.jvmserver.unclassified.userid` プロパティの中で指定する必要があります。

また、JVM サーバーで実行される未分類の作業について、CJSU の重複トランザクションを使用することもできます。この場合、`com.ibm.cics.jvmserver.unclassified.tranid` プロパティでトランザクション ID を指定し、このトランザクションに必要なアクセス権を付与する必要があります。

JVM システム・プロパティについては、[JVM システム・プロパティ](#)を参照してください。

Liberty JVM サーバーのセキュリティ構成について詳しくは、[Liberty JVM サーバーに関するセキュリティの構成](#)を参照してください。

12. 領域 JCL を更新して、CMCI 診断用の新しい DD ステートメントを組み込みます。

```
//JVMOUT DD SYSOUT=*,LRECL=1024
//JVMERR DD SYSOUT=*,LRECL=1024
//JVMTRACE DD SYSOUT=*,LRECL=1024
//MSGLOG DD SYSOUT=*,LRECL=1024
```

13. 機能切り替え構成ファイルを参照して、CMCI JVM サーバーの機能切り替えが `true` に設定されているか、デフォルトのままになっていることを確認します。

```
com.ibm.cics.cmci.jvmserver=true
```

機能切り替え構成については、[機能切り替えの指定](#)を参照してください。

14. CMCI JVM サーバーでユーザーが認証できるようにします。

CMCI JVM サーバーで認証するためのアクセス権限 (CMCI を使用する権限を含む) をユーザーに付与する必要があります。

RACF を使用している場合は、以下のように、CMCI JVM サーバーを介してユーザーを認証するように SEYUSAMP 内のサンプル CLIST EYU\$CMCI を更新できます。

a) SEYUSAMP 内の CLIST EYU\$CMCI のコピーをとります。

b) 以下の変数を指定することにより、コピーを更新します。

**WUI\_REGION\_USERID**

どのユーザー ID で WUI 領域が実行されるかを指定します。

**WUI\_APPLID**

WUI 領域の APPLID を指定します。

**CMCIUSER\_ACCESS\_LIST**

CICS Explorer 経由で CMCI にアクセスするユーザーまたはユーザー・グループのリストを指定します。

**profile\_prefix**

JVM プロファイルで定義されている、WUI 領域の SAF プロファイル接頭部を指定します。

c) CLIST を実行します。

RACF 以外の外部セキュリティ・マネージャーを使用している場合の手順については、外部製品の資料を参照してください。

## 次のタスク

CMCI JVM サーバーに接続できるクライアントを制限するには、CMCI JVM サーバーに対してクライアント・ホワイトリストを定義します。詳しくは、[382 ページの『CMCI JVM サーバーに対するクライアント・ホワイトリストの定義』](#)を参照してください。

CICS バンドル・デプロイメント API を使用して CICS バンドルをデプロイする場合は、CMCI JVM サーバーの追加構成を実行する必要があります。詳しくは、[371 ページの『CICS バンドル・デプロイメント API 用の CMCI JVM サーバーの構成』](#)を参照してください。

## CMCI JVM サーバーを使用するための WUI 領域の構成

WUI 領域に CMCI が構成されていても、CMCI JVM サーバーが無効になっている場合は、WUI 領域をアップグレードすることで、CMCI JVM サーバーを使用できるようになります。CMCI JVM サーバーは、拡張クライアント認証のサポート、GraphQL API のサポート、および CICS バンドル・デプロイメント API のサポートを提供する Liberty サーバーです。

### 始める前に

#### CMCI JVM サーバーのシステム要件

1. 必要な Java コンポーネントがすべてインストールされていることを確認してください。[Java コンポーネントのチェックリスト](#)に従ってください。
2. 領域に Java サポートがセットアップされていることを確認してください。詳しくは、[Setting up Java support](#) を参照してください。

#### 接続で多要素認証 (MFA) 資格情報を使用できるようにするための追加要件

- 多要素認証をサポートするには、IBM Multi-Factor Authentication for z/OS、または RACF が構成された同等の製品が必要です。代わりに外部セキュリティ・マネージャー (ESM) を使用する場合、詳細についてベンダーに問い合わせてください。
- CMCI JVM サーバーの実行場所となる領域と、このサーバーの接続先 CMAS が同じ CICS レベルであることを確認します。CICSplex SM トポロジーのセットアップに関する CICS レベルの考慮事項については、[Designing your CICSplex SM environment](#) を参照してください。

### このタスクについて

WUI 領域での CMCI JVM サーバーの使用を構成して使用可能にする必要があります。

### 手順

1. 以下のシステム 初期設定パラメーターを領域に追加します。
  - **JVMPROFILEDIR**
  - **START=INITIAL**
  - **EDSALIM**
2. **KEYRING**、**NISTSP800131A**、および **SEC** システム 初期設定パラメーターの値を確認します。  
これらのシステム 初期設定パラメーターは、**APPLID** および **DFLTUSER** と一緒に、CMCI JVM サーバー構成パラメーターにマップされます。場合によっては CMCI JVM サーバーで追加構成が必要になる可能性があるということに注意してください。詳しくは、[373 ページの『CICSplex SM WUI サーバーと CMCI JVM サーバーの間の構成パラメーターのマッピング』](#)を参照してください。
3. CMCI オプションや TCP/IP オプションなどの WUI サーバー 初期設定パラメーターを確認します。  
CMCI JVM サーバーと互換性のない値をいくつか変更しなければならない場合があります。詳しくは、[373 ページの『CICSplex SM WUI サーバーと CMCI JVM サーバーの間の構成パラメーターのマッピング』](#)を参照してください。
4. CMCI JVM サーバーの JVM プロファイルを作成します。
  - a) /usr/lpp/cics56/JVMProfiles にある EYUCMCIJ.jvmprofile を、**JVMPROFILEDIR** システム 初期設定パラメーターで指定された場所にコピーします。
  - b) JVM プロファイルの **JAVA\_HOME** および **WORK\_DIR** の値を検証または更新します。  
詳しくは、[JVM プロファイルで使用されるシンボル](#)を参照してください。
  - c) 以下の行を JVM プロファイルに追加することで、WUI 領域の SAF プロファイル接頭部を指定します。

SAF プロファイル接頭部は各 WUI 領域で使用される RACF ルールを決定して、同じ SAF プロファイル接頭部を持つ複数の WUI 領域で同じセキュリティ構成が共用されるようにします。例えば、CICS バンドル・デプロイメント API のユーザーを認証する場合に、同じ SAF プロファイル接頭部を持つ複数の WUI 領域内の CMCI JVM サーバーでは、同じユーザーが API にアクセスすることが許可されます。

```
-Dcom.ibm.cics.jvmserver.wlp.saf.profilePrefix=${MYPREFIX}
```

ここで、`${MYPREFIX}` は、セキュリティ構成を共用する必要がある WUI 領域の SAF プロファイル接頭部です。

## 5. CMCI JVM サーバー のストレージ要件を設定します。

付属の `EYUCMCIJ.jvmprofile` ファイルでは共用ライブラリー領域の使用が無効になっており、必要とされる非 CICS 31 ビット・ストレージの量が削減されます。デフォルトの場合、CMCI JVM サーバー用に自動的に作成される JVMSERVER リソースでは、**THREADLIMIT** 属性の値が 15 になっています。したがって、ストレージ要件の初期見積もりとしては以下の値を使用できます。

- 24 ビット・ストレージ: 512 KB
- 31 ビット・ストレージ: 100 MB

引き続き、[JVM サーバーのストレージ要件の計算](#)の説明に従ってストレージ要件をモニターし、確認します。

## 6. CMCI JVM サーバー用の Liberty エンジェル・プロセス開始タスクを構成します。

- a) Liberty エンジェル・プロセスが実行されていない場合は、[Liberty サーバー・エンジェル・プロセス](#)のステップに従ってそれを作成します。
- b) 既に Liberty エンジェル・プロセスが実行されている場合は、エンジェル JCL の ROOT シンボリック・パラメーターで指定された Liberty のバージョンが、CICS で提供される Liberty のバージョン以上のレベルであることを確認します。

### 例: 開始タスクのシステム・ログから Liberty のバージョンを確認する

Liberty エンジェル・プロセスが Liberty 18.0.0.2 以上で実行されている場合、開始タスクのシステム・ログには、Liberty のバージョンを示す次のようなメッセージが含まれています。

```
CWWKB0079I THE ANGEL BUILD LEVEL IS 18.0.0.2 20180619-0654 2018.7.0.0 20180619-0654
```

### 例: メッセージ DFHSJ1405 から Liberty のバージョンを確認する

CICS JVM サーバーで実行されている Liberty のバージョンが、次のメッセージで示されます。

```
DFHSJ1405I 08/22/2018 17:04:39 IYK3ZDRI JVMSERVER EYUCMCIJ is running WebSphere Application Server
Version 18.0.0.2 Liberty - (18.0.0.2-cl180220180619-0403) process ID
67174497.
```

### 例: スクリプトを実行して Liberty のバージョンを確認する

エンジェル JCL が次の ROOT パラメーターを指定しているとします。

```
// SET ROOT='/usr/lpp/zosmf/wlp'
```

Liberty のバージョンを調べるには、次のスクリプトを実行します。

```
/usr/lpp/zosmf/wlp/bin/productInfo version --verbose
```

CICS の場合は、次のスクリプトを実行します。

```
/usr/lpp/cicsts56/wlp/bin/productInfo version --verbose
```

```

WebSphereApplicationServer.properties:
com.ibm.websphere.productId=com.ibm.websphere.appserver
com.ibm.websphere.productOwner=IBM
com.ibm.websphere.productVersion=16.0.0.3
com.ibm.websphere.productName=WebSphere Application Server
com.ibm.websphere.productInstallType=Archive
com.ibm.websphere.productEdition=zOS
com.ibm.websphere.productLicenseType=IPLA

WebSphereApplicationServerZOS.properties:
com.ibm.websphere.productId=com.ibm.websphere.appserver.zos
com.ibm.websphere.productOwner=IBM CORP
com.ibm.websphere.productVersion=16.0.0.3 <== Liberty Version
com.ibm.websphere.productName=WAS FOR Z/OS
com.ibm.websphere.productPID=5655-WAS
com.ibm.websphere.productQualifier=WAS Z/OS
com.ibm.websphere.productReplaces=com.ibm.websphere.appserver
com.ibm.websphere.productEdition=
com.ibm.websphere.gssp=true

zOSMF.properties:
com.ibm.websphere.productId=com.ibm.zosmf
com.ibm.websphere.productOwner=IBM
com.ibm.websphere.productVersion=2.2.0
com.ibm.websphere.productName=z/OSMF
com.ibm.websphere.productPID=5650-ZOS
com.ibm.websphere.productQualifier=z/OSMF
com.ibm.websphere.productReplaces=com.ibm.websphere.appserver.zos
com.ibm.websphere.productEdition=N/A

```

図 60. 出力例

c) Liberty のバージョンが CICS で提供される Liberty のバージョンより低いレベルの場合は、CICS Liberty インストールを使用して、指定されたエンジェル・プロセスを構成します。

1) CMCI JVM サーバーの JVM プロファイル EYUCMCIJ.jvmprofile 内の次の行のコメントを外します。

```
#-Dcom.ibm.ws.zos.core.angelName=<named_angel>
```

2) -Dcom.ibm.ws.zos.core.angelName プロパティにエンジェル名を指定します。

このプロパティは、指定されたエンジェル・プロセスに CMCI JVM サーバーが接続できるようにします。詳しくは、[Liberty サーバー・エンジェル・プロセス](#)を参照してください。

d) 領域を開始する前に、Liberty エンジェル・プロセスの準備ができていることを確認します。

7. Liberty エンジェル・プロセスを使用するように WUI 領域のセキュリティを構成します。

RACF を使用している場合は、以下のように、SEYUSAMP に含まれるサンプル CLIST EYU\$ANGL を使用することにより、WUI 領域で Liberty エンジェル・プロセスを使用するための RACF 定義を作成することができます。

a) SEYUSAMP 内の CLIST EYU\$ANGL のコピーをとります。

b) 以下の変数を指定することにより、コピーを更新します。

#### WUI\_REGION\_USERID

どのユーザー ID で WUI 領域が実行されるかを指定します。

#### ANGEL\_NAME

名前付きエンジェル・プロセスを使用している場合、値は ANGEL.name です (name は -Dcom.ibm.ws.zos.core.angelName プロパティの値)。

名前エンジェル・プロセスを使用していない (-Dcom.ibm.ws.zos.core.angelName が指定されていない) 場合、値は ANGEL です。

c) CLIST を実行します。

RACF 以外の外部セキュリティ・マネージャーを使用している場合の手順については、外部製品の資料を参照してください。

8. CMCI JVM サーバーから発生したタスクは、デフォルトで CJSU トランザクションで実行されます。WUI 領域でトランザクションのセキュリティがアクティブの場合は、CJSU トランザクションに CICS デフォルト・ユーザー・アクセスを付与します。

あるいは、CICS のデフォルト・ユーザーに基づいて、CJSU トランザクションに対する追加のアクセス権限が付与された新規ユーザー ID を作成することもできます。そのユーザー ID を、`com.ibm.cics.jvmserver.unclassified.userid` プロパティの中で指定する必要があります。

また、JVM サーバーで実行される未分類の作業について、CJSU の重複トランザクションを使用することもできます。この場合、`com.ibm.cics.jvmserver.unclassified.tranid` プロパティでトランザクション ID を指定し、このトランザクションに必要なアクセス権を付与する必要があります。

JVM システム・プロパティについて詳しくは、[JVM システム・プロパティ](#)を参照してください。

Liberty JVM サーバーのセキュリティ構成について詳しくは、[Liberty JVM サーバーに関するセキュリティの構成](#)を参照してください。

9. 領域 JCL を更新して、CMCI 診断用の新しい DD ステートメントを組み込みます。

```
//JVMOUT DD SYSOUT=*,LRECL=1024
//JVMERR DD SYSOUT=*,LRECL=1024
//JVMTRACE DD SYSOUT=*,LRECL=1024
//MSGLOG DD SYSOUT=*,LRECL=1024
```

10. 機能切り替え構成ファイルを参照して、CMCI JVM サーバーの機能切り替えが true に設定されているか、デフォルトのままになっていることを確認します。

```
com.ibm.cics.cmci.jvmserver=true
```

機能切り替え構成については、[機能切り替えの指定](#)を参照してください。

11. CMCI JVM サーバーでユーザーが認証できるようにします。

CMCI JVM サーバーで認証するためのアクセス権限 (CMCI を使用する権限を含む) をユーザーに付与する必要があります。

RACF を使用している場合は、以下のように、CMCI JVM サーバーを介してユーザーを認証するように SEYUSAMP 内のサンプル CLIST EYU\$CMCI を更新できます。

a) SEYUSAMP 内の CLIST EYU\$CMCI のコピーをとります。

b) 以下の変数を指定することにより、コピーを更新します。

#### **WUI\_REGION\_USERID**

どのユーザー ID で WUI 領域が実行されるかを指定します。

#### **WUI\_APPLID**

WUI 領域の APPLID を指定します。

#### **CMCIUSER\_ACCESS\_LIST**

CICS Explorer 経由で CMCI にアクセスするユーザーまたはユーザー・グループのリストを指定します。

#### **profile\_prefix**

JVM プロファイルで定義されている、WUI 領域の SAF プロファイル接頭部を指定します。

c) CLIST を実行します。

RACF 以外の外部セキュリティ・マネージャーを使用している場合の手順については、外部製品の資料を参照してください。

## 次のタスク

CMCI JVM サーバーに接続できるクライアントを制限するには、CMCI JVM サーバーに対してクライアント・ホワイトリストを定義します。詳しくは、[382 ページの『CMCI JVM サーバーに対するクライアント・ホワイトリストの定義』](#)を参照してください。



CICS バンドル・デプロイメント API を使用して CICS バンドルをデプロイする場合は、CMCI JVM サーバーの追加構成を実行する必要があります。詳しくは、[371 ページの『CICS バンドル・デプロイメント API 用の CMCI JVM サーバーの構成』](#)を参照してください。

## CICS バンドル・デプロイメント API 用の CMCI JVM サーバーの構成

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 バンドルを領域にデプロイすることをサポートします。この API は、Maven プラグインや Gradle プラグインなどの HTTP クライアントで使用できます。CICS バンドル・デプロイメント API を使用するには、システム・プログラマーは CMCI JVM サーバーの追加構成を行う必要があります。

### 始める前に

この API 向けの構成を行う前に、API がどのように機能するのかを理解することを強くお勧めします。  
[How it works: CICS bundle deployment API](#) を参照してください。

### ソフトウェア要件

CICS バンドル・デプロイメント API は、WUI 領域内でセットアップされる必要がある CMCI JVM サーバーによってサポートされます。次のことを確認します。

- CICS TS V5.6 以降である CICS 領域がある。
- この領域は、デプロイメント・ターゲット領域を含む CICSplex の WUI 領域となるように構成されます。[CICSplex SM Web ユーザー・インターフェース・サーバーをセットアップする server](#) を参照してください。
- この WUI 領域は、CMCI JVM サーバーを使用するように構成される必要があります。
  - この WUI 領域で CMCI をセットアップしていない場合は、CMCI JVM サーバーを直接セットアップできます。[362 ページの『WUI 領域の CMCI の構成』](#)を参照してください。
  - この WUI 領域内で CMCI をセットアップ済みであるが、WUI 領域で CMCI JVM サーバーが使用されない場合は、CMCI JVM サーバーを使用するように WUI 領域をアップグレードできます。[367 ページの『CMCI JVM サーバーを使用するための WUI 領域の構成』](#)を参照してください。

システム初期設定パラメーターを [SEC=YES](#) に設定します。

### 手順

1. CMCI JVM サーバー用の JVM プロファイルである EYUCMCIJ.jvmprofile 内でバンドルのディレクトリ (**-Dcom.ibm.cics.jvmserver.cmci.bundles.dir**) を指定することで、CICS バンドル・デプロイメント API を使用可能にします。

以下の行を追加します。

```
-Dcom.ibm.cics.jvmserver.cmci.bundles.dir=<bundles_directory>
```

バンドルの管理に API を使用するための専用ディレクトリを zFS 上に作成することをお勧めします。CICS バンドル・デプロイメント 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に保管され、CICS ターゲット領域によってアクセスされます。

2. オプション: JVM プロファイル内で、API を渡すことが許可されるデータに関する仕様を追加することもできます。

#### **-Dcom.ibm.cics.jvmserver.cmci.deploy.timeout={120000|timeout\_limit}**

CICS バンドルをデプロイするためのタイムアウト制限をミリ秒単位で指定します。これには、すべてのバンドル・ライフサイクル・アクション (無効化、破棄、インストール、使用可能化を含む) の時間が含まれます。

デフォルト値は 120000 (120 秒) です。

#### **-Dcom.ibm.cics.jvmserver.cmci.max.file.size={52428800|max\_file\_size}**

アップロードされる CICS バンドルに許可される最大サイズをバイト単位で指定します。デフォルト・サイズは 52428800 (50 MB) です。



アップロードされたバンドルのサイズがこのサイズを超える場合は、400 Bad Request メッセージ「SRVE8021E: アップロードするファイルが大きすぎます (The file being uploaded is too large)」が発行されます。

**-Dcom.ibm.cics.jvmserver.cmci.max.request.size={104857600|max\_request\_size}**

multipart 要求または form-data 要求に許可される最大サイズをバイト単位で指定します。デフォルト・サイズは 104857600 (100 MB) です。

アップロードされたすべてのファイルの合計サイズがこのしきい値を超える場合は、Web コンテナは例外をスローします。

3. CLIST でセキュリティ・レベルを構成します。

RACF を使用している場合は、SEYUSAMP 内のサンプル CLIST EYU\$BUND をコピーして更新できます。以下の必須オプションを指定します。

**deploy\_userid**

バンドル・デプロイメントを実行する権限を持つユーザー ID を指定します。この目的用の機能 ID (例: DPLYUSR) を作成することをお勧めします。または、ご使用の領域ユーザー ID に十分な権限があることを確認して、機能 ID の代わりにこの領域ユーザー ID 指定してください。

**mngdbndl\_access\_list**

CICS バンドル・デプロイメント API へのアクセス権限を持つユーザー・グループ (例: DEVELOPER) を指定します。

**profile\_prefix**

JVM プロファイルで定義されている、WUI 領域の SAF プロファイル接頭部を指定します。

**wui\_region\_userid**

CMCI JVM サーバーを使用するように構成した CICSplex SM WUI 領域のユーザー ID を指定します (例: CICSRRGN)。

以下のオプションの指定は任意です。指定されない場合、デフォルト値が使用されます。

**classname**

RACF トランザクション・グループ化クラスの名前を指定します。デフォルト値は GCICSTRN です。

**cicsplex**

デプロイメント機能 ID (deploy\_userid) が照会してバンドルをデプロイすることを許可されている CICSplex。

デフォルト値の \* を使用した場合は、deploy\_userid は任意の CICSplex への照会とバンドルのデプロイを行うことができます。

**cmas**

デプロイメント機能 ID (deploy\_userid) が照会することを許可されている CMAS。

デフォルト値の \* を使用した場合は、deploy\_userid は指定された CICSplex 内の任意の CMAS に照会できます。

**notify**

アクセス違反の場合に通知を受ける TSO ユーザー ID を指定します。デフォルト値は IBMUSER です。

**owner**

プロファイル所有者の TSO ユーザー ID を指定します。デフォルト値は IBMUSER です。

**領域**

デプロイメント機能 ID (deploy\_userid) が照会してバンドルをデプロイすることを許可されているターゲット CICS 領域を指定します。

デフォルト値の \* を使用した場合は、deploy\_userid は指定された CICSplex 内の任意の CICS 領域への照会とバンドルのデプロイを行うことができます。

次に CLIST を実行します。

4. UNIX コマンド **chmod** を使用するかアクセス制御リスト (ACL) を適用することで、zFS 上のバンドルのディレクトリーへのアクセス権限をユーザー ID に付与します。詳しくは、[Implementing security for](#)

[z/OS UNIX files](#) および [CICS 領域に対する z/OS UNIX ディレクトリーおよびファイルへのアクセス権限の付与](#)を参照してください。

- デプロイメント機能 ID (この例では DPLYUSR) には、バンドルのディレクトリーへの READ、WRITE、および EXECUTE 権限が必要です。
- ターゲット CICS 領域の領域ユーザー ID には、ディレクトリーに対する READ および EXECUTE 権限が必要です。

### 次のタスク

CICS バンドル・デプロイメント API 向けの構成を行った後に、Java 開発者は、CICS 側で提供される Maven プラグインや Gradle プラグインなどのクライアントを使用して CICS バンドルをデプロイできます。説明は、[How it works: CICS bundle deployment API](#) を参照してください。

## CICSplex での複数の CMCI JVM サーバーのセットアップ

CICSplex で複数の CMCI JVM サーバーを実行できます。ただし、1 つの CMCI JVM サーバーで HTTP クライアント・ユーザーの認証を 1 回行うだけで同じ CICSplex 内の他の CMCI JVM サーバーに再認証なしでアクセスできるようにするには、Liberty でシングル・サインオン (SSO) サポートを構成する必要があります。

SSO を使用すると、ユーザーは 1 つの CMCI JVM サーバーで認証するだけで、プロンプトが再度出されることなく、CICSplex 内の他の CMCI JVM サーバーにアクセスできるようになります。ユーザーが 1 つの CMCI JVM サーバーで認証されると、認証プロセスでそのユーザー用に作成された SSO トークンが Cookie でクライアントに転送されます。この Cookie は、同じ CICSplex 内の他の CMCI JVM サーバーに認証情報を伝搬するために使用されます。

認証のみが共有されます。CMCI キャッシュ内の結果セットは共有されません。

LTPA および SSO に対する Liberty サポートの概要については、[認証](#)を参照してください。

### このタスクについて

CMCI JVM サーバー間で SSO を使用するには、CICSplex で実行される CMCI JVM サーバーが同じ LTPA 鍵を使用し、同じユーザー・レジストリーを共有する必要があります。

複数の異なるドメインにある CMCI JVM サーバーをサポートするように SSO を構成できます。

### 手順

- Liberty で LTPA を構成します。  
[Liberty での LTPA の構成の説明](#)に従ってください。Liberty サーバー構成で設定できる LTPA プロパティーについては、[LTPA トークン \(ltpa\)](#)を参照してください。
- Liberty で LTPA Cookie を使用するよう、SSO 構成サポートをカスタマイズします。  
[Liberty での LTPA Cookie を使用した SSO 構成のカスタマイズ](#)の説明に従ってください。

## CICSplex SM WUI サーバーと CMCI JVM サーバーの間の構成パラメーターのマッピング

セットアップ時に、CMCI JVM サーバー は CICSplex SM WUI 構成プロパティーを読み取り、それに応じてその JVM 構成プロパティーを構成します。WUI サーバー初期設定パラメーター (CMCI オプションや TCP/IP オプションなど) および WUI 領域 SIT パラメーターから、CMCI JVM サーバー 構成パラメーターへのマッピングがリストされます。パラメーターによっては、CMCI JVM サーバーで追加構成が必要なものもあります。

374 ページの表 34 は、CICSplex SM WUI サーバー初期設定パラメーターが CMCI JVM サーバー構成パラメーターにどのようにマップされるかを示すとともに、CMCI JVM サーバーで必要な追加構成を示しています。

表 34. CICSplex SM WUI サーバー初期設定パラメーターと CMCI JVM サーバー構成パラメーターの間のマッピング

CICSplex SM WUI パラメーター名	WUI パラメーター値	CMCI への影響	CMCI JVM サーバー構成要素
CMCIAUTH	AUTOMATIC	CMCI は、基本認証とクライアント証明書認証の両方をサポートします。 <b>1</b>	ssl 要素内の clientAuthenticationSupported="true"  webAppSecurity 要素内の allowFailOverToBasicAuth="true"  基本認証とクライアント認証がサポートされています。
	AUTOREGISTER	サポートなし。AUTOMATIC として構成されます。	サポートなし。AUTOMATIC として構成されます。
	BASIC	CMCI は基本認証を必要とします。 <b>SEC=YES</b> になっているときはこれがデフォルトです。 <b>1</b>	ssl 要素内の clientAuthentication="false" および clientAuthenticationSupported="false"  webAppSecurity 要素内の allowFailOverToBasicAuth="false"  基本認証のみがサポートされています。
	CERTIFICATE	CMCI はクライアント証明書認証を必要とします。 <b>1 2</b>	true に設定された ssl 要素内の clientAuthentication="true"  webAppSecurity 要素内の allowFailOverToBasicAuth="false"  クライアント認証のみがサポートされています。
	NO	CMCI はクライアント認証を必要としません。 <b>SEC=NO</b> になっているときはこれがデフォルトです。	セキュリティは構成されません。
CMCIPORT (必須)	value	CMCI の HTTP または HTTP ポートを設定します。	httpEndPoint 要素内の httpPort または httpsPort

表 34. CICSplex SM WUI サーバー初期設定パラメーターと CMCI JVM サーバー構成パラメーターの間のマッピング (続き)			
CICSplex SM WUI パラメーター名	WUI パラメーター 値	CMCI への影響	CMCI JVM サーバー構成要素
CMCISSL (TCIPSSL をオーバーライド)	ATTLISAWARE	CMCI への 非 HTTPS 接続が使用されます。	SSL を使用するように CMCI JVM サーバーを構成せずに、httpEndPoint 要素内の httpsPort を無効にしてください。
	CLIENTAUTH	CMCI への HTTPS 接続が使用されます。CMCI はクライアント証明書認証を必要とします。	ssl 要素内の clientAuthentication="true" SSL を使用するように CMCI JVM サーバーを構成し、httpEndPoint 要素内の httpPort を無効にしてください。
	NO	CMCI への 非 HTTPS 接続が使用されます。	SSL を使用するように CMCI JVM サーバーを構成せずに、httpEndPoint 要素内の httpsPort を無効にしてください。
	YES	CMCI への HTTPS 接続が使用されます。	SSL を使用するように CMCI JVM サーバーを構成し、httpEndPoint 要素内の httpPort を無効にしてください。
TCPIPADDRESS (TCPIPHOSTNAME をオーバーライド)	name	指定された TCP/IP アドレスのみを使用した CMCI へのクライアント接続を許可します。	httpEndPoint 要素内の host
	INADDR_ANY	LPAR に関連付けられているいずれかの TCP/IP アドレスを使用した CMCI へのクライアント接続を許可します。	httpEndPoint 要素内の host="*"
TCPIPHOSTNAME (必須)	name	LPAR に関連付けられているいずれかのホスト名を使用したクライアント接続を許可します。	httpEndPoint 要素内の host="*"
TCPIPHTTPHOST	YES NO	CMCI は使用しない。	適用外
TCIPSSL (CMCISSL によりオーバーライドされることがある)	YES	CMCI への HTTPS 接続が使用されます。	SSL を使用するように CMCI JVM サーバーを構成し、httpEndPoint 要素内の httpPort を無効にしてください。
	NO	CMCI への 非 HTTPS 接続が使用されます。これはデフォルトです。	SSL を使用するように CMCI JVM サーバーを構成せずに、httpEndPoint 要素内の httpsPort を無効にしてください。
TCIPSSLCERT	name	TCIPSSL=YES が有効な場合、または CMCISSL が NO 以外の値に設定されている場合に有効になります。	使用されません。 Liberty はデフォルトの証明書を使用します。 <b>3</b>
TCIPSSLCIPHERS	cipher_list	TCIPSSL=YES が有効な場合、または CMCISSL が NO 以外の値に設定されている場合に有効になります。 CMCI で使用可能な暗号のリストを指定します。 <b>4</b>	ssl 要素内の enabledCiphers SSL がアクティブのときだけ使用されます。 <b>4</b>

注:

1. **SEC=YES** になっているときに有効です。
2. **KEYRING** システム 初期設定パラメーターが有効になっていなければなりません。
3. デフォルトの SSL 証明書が Liberty で構成されていることを確認してください。詳しくは、[Liberty での SSL のデフォルト](#)を参照してください。
4. 無効な暗号がリストに含まれている場合、CICS は無効な暗号を除去し、有効な暗号が少なくとも 1 つ残っている限り続行します。有効な暗号がなければ、CMCI へのアクセスは拒否されます。Liberty は、無効な暗号を構成することを許可しますが、接続を拒否し、Liberty ログに **Unsupported ciphersuite** というメッセージを記録します。このような場合は、問題の原因を特定するうえで以下のメッセージが役立ちます。

DFHSO0145W は、無効な暗号が指定されたことを示します。

DFHSO0146I では、CICS によって除去された無効な暗号がリストされます。

#### Liberty のリファレンス :

ssl 要素の属性については、[Liberty での SSL 構成属性](#)を参照してください。

webAppSecurity 要素の属性については、[Liberty での Web コンテナ・アプリケーション・セキュリティー \(webAppSecurity\)](#)を参照してください。

httpEndPoint 要素の属性については、[Liberty での HTTP エンドポイント \(httpEndpoint\)](#)を参照してください。

376 ページの表 35 は、WUI 領域システム 初期設定パラメーターが CMCI JVM サーバー 構成パラメーターにどのようにマップされるかを示すとともに、CMCI JVM サーバーでどのような追加構成が必要かを示しています。

表 35. CICSplex SM WUI 領域システム初期設定パラメーターと CMCI JVM サーバー構成パラメーターの間のマッピング			
CICSplex SM WUI 領域 SIT パラメーター名	WUI 領域 SIT パラメーター値	CMCI への影響	CMCI JVM サーバー構成要素
APPLID	applid (アプリケーション ID)	CMCI セキュリティー・プロファイルの接頭部を設定します。	safCredentials 要素内の profilePrefix
DFLTUSER	userid	CMCI 非認証デフォルト・ユーザーを設定します。	safCredentials 要素内の unauthenticatedUser  セキュリティーがアクティブのときに使用されます。
KEYRING	keyring-name	HTTPS またはクライアント証明書認証に使用される鍵リングの名前を指定します。	keyStore 要素内の location  SSL またはクライアント認証がアクティブのときに使用されます。
MINTLSLEVEL		CMCI は使用しない。	ssl 要素内の sslProtocol を構成することで、SSL プロトコルを設定できます。
NISTSP800131A	NOCHECK CHECK	NIST SP800-131A 標準に準拠しているかどうかを検査するよう CMCI に指示します。	SP800-131a で実行されるように <a href="#">Liberty</a> を構成してください。



表 35. CICSplex SM WUI 領域システム初期設定パラメーターと CMCI JVM サーバー構成パラメーターの間のマッピング (続き)			
CICSplex SM WUI 領域 SIT パラメーター名	WUI 領域 SIT パラメーター値	CMCI への影響	CMCI JVM サーバー構成要素
SEC	NO	認証を無効にします。	Liberty セキュリティーを無効にします。
	YES	CMCIAUTH によってオーバーライドされない限り、CMCI での基本認証を有効にします。	Liberty セキュリティーを有効にします。

#### Liberty のリファレンス：

safCredentials 要素の属性については、[Liberty](#) でのインターフェース [SAFCredential](#) を参照してください。

ssl 要素の属性については、[Liberty](#) での [SSL 構成属性](#) を参照してください。

### CMCI でのレコード・カウント警告

レコード・カウント警告をセットアップすると、CICS 管理クライアント・インターフェース (CMCI) 要求は、許容できないほど大量のデータの取り出しが要求の結果として起きそうな場合には失敗します。レコード・カウント警告のセットアップは、CMCI のセットアップ時の WUI サーバー構成中に、ゼロ以外の値を **DEFAULTWARNCNT** 初期化パラメーターに指定することによって行います。

**DEFAULTWARNCNT** には、0 から 99999999 の派煮の整数値を指定することができます。デフォルト値は 0 で、警告が出されないことを意味します。

警告メカニズムにより、返されるレコードの数が警告カウント値より多いと判断されると、要求は HTTP 403 応答コードで失敗します。ユーザーが何らかの結果を取り出すには、**SCOPE**、**CRITERIA**、そして場合によっては **PARAMETER** オプションの値を変えて新しい要求を発行する必要があります。返される値が警告カウント値より小か等しい場合には、要求は通常の方法で処理されます。

注：CMCI 要求では、**count**、**index**、**SUMMONLY** オプションを使用して、レコードの表示を制限するか、レコードが表示されないようにすることができます。ただし、これらのオプションが、要求されるレコードの数に影響することはない、またそれがレコード・カウント警告メカニズムに影響を及ぼすことはありません。

#### ユーザーがレコード・カウント警告をバイパスできるようにする方法

**DEFAULTWARNCNT** がゼロ以外の値に設定されていると、警告カウント限度が有効になります。ユーザーは、要求内で CMCI URI オプション **OVERRIDEWARNINGCOUNT** を使用して、警告カウント限度をバイパスできます。CMCI URI オプション **OVERRIDEWARNINGCOUNT** を有効にするには、CMCI をセットアップする際に、WUI サーバーの構成中に **RESOURCELIMIT** 初期設定パラメーターを **WARNING** に設定しなければなりません。ユーザーが警告カウント限度をバイパスできないようにするには、**RESOURCELIMIT** を **FAIL** に設定します。

**RESOURCELIMIT(WARNING)** が有効の場合に、要求の結果として警告カウント限度よりも多い量のデータが取り出されることになると、403 HTTP 応答が返され、**errorInfo** 属性 **override\_warning\_count\_allowed** が **yes** に設定されていることが示されます。その後ユーザーは、要求内で CMCI URI オプション **OVERRIDEWARNINGCOUNT** を使用して、警告カウント限度をバイパスできます。

しかし、**RESOURCELIMIT(FAIL)** が有効の場合は、上記のような要求は拒否され、403 HTTP 応答で **errorInfo** 属性 **override\_warning\_count\_allowed** が **no** に設定されていることが示されます。

**OVERRIDEWARNINGCOUNT** オプションを指定して要求を発行しても、結果は同じ 403 HTTP 応答になります。



## 適用できる CMCI リソース

この機能はすべてのリソースには適用されません。CMCI では、レコード・カウント警告は、以下のリソースに関連した CMCI 要求にのみ適用されます。

CMCI リソース
AIMODEL
ATOMSERV
BRFACIL
BUNDLE
BUNDPART
CFDTPOOL
CICSDSA
CICSPAGP
CICSRGN
CICSSTOR
CLCACHE
CMDT
CONNECT
DBCTLSS
DB2CONN
DB2ENTRY
DB2TRN
DOCTEMP
DOMSPOOL
DSNAME
DSPGBL
DSPMODE
DSPPOOL
EJCOBEAN
EJCOSE
EJDJAR
EJDJBAN
ENQUEUE
ENQMODEL
EPADAPT
EPADSET
EPAINSET
EVCSDATA

<b>CMCI リソース</b>
EVCSINFO
EVCSOPT
EVCSPEC
EVNTBIND
EVNTGBL
EXCI
EXITGLUE
EXITTRUE
EXTRATDQ
FEPICONN
FEPINODE
FEPIPOOL
FEPIPROP
FEPITRGT
HOST
HTASK
INDTDQ
INTRATDQ
IPCONN
IPFACIL
JRNLMODL
JRNLNAME
JVM
JVMPPOOL
JVMPROF
JVMSERV
LIBDSN
LIBRARY
LOADACT
LOADER
LOCFILE
LOCTRAN
LSRPBUF
LSRPOOL
MASHIST

<b>CMCI リソース</b>
MODENAME
MONITOR
MQCON
MQCONN
MQINI
MVSESTG
MVSTCB
MVSTCBGL
MVSWLM
OSGIBUND
OSGISERV
PARTNER
PIPELINE
PROCTYP
PROFILE
PROGRAM
RECOVERY
REMFIL
REMTDQ
REMTAN
REQID
RPLIST
RQMODEL
RULE
STREAMNM
SYSDUMP
SYSPARM
TASK
TASKASSC
TASKESTG
TASKFILE
TASKRMI
TASKTSQ
TCPIPGBL
TCPIPS

<b>CMCI リソース</b>
TDQGBL
TERMNL
TRANCLAS
TRANDUMP
TSKSPOLS
TSKSPool
TSMODEL
TSPOOL
TSQGBL
TSQNAME
TSQSHR
UOW
UOWDSNF
UOWENQ
UOWLINK
URIMAP
URIMPGBL
WEBSERV
WORKREQ
XMLTRANS

## CMCI のストレージ要件の見積もり

CICS 管理クライアント・インターフェースを使用するときに潜在的なストレージ問題が発生するのを回避するためには、適切なレベルの 64 ビット (2 GB 境界より上) のストレージを CICS 領域に指定し、さらに補助ストレージを指定する必要があります。WUI 領域または CICS 領域の 64 ビット・ストレージに制限を設定するには、z/OS パラメーター **MEMLIMIT** を使用し、CMAS の補助ストレージを設定するには、CICSplex SM システム・パラメーター **MAXAUXCPsm** および **MAXAUXTOTL** を使用します。

### このタスクについて

大量のワークロードで CMCI を実行すると、WUI サーバーでストレージ不足の状況が発生し、補助ストレージを使い果たすことによって CMAS がシャットダウンすることがあります。

CMCI は、WUI サーバー用に保持されている結果セットを、サブプール WU\_64 の GCDSA にある、CICS 領域の 64 ビット・ストレージに格納します。

CMCI 要求の間、CMAS は要求されたリソース・レコードを収集および保管します。それらのレコードは補助ストレージにバックアップされます。CMCI を介して要求を同時実行すると、各要求で CMAS によって保持されるレコードの数が乗算されます。

WUI 領域で CMCI と共に CMCI JVM サーバー を使用している場合、JVM サーバーは保持されている結果セットに加えてストレージを使用します。見積もりの中には、CMCI JVM サーバー のストレージ要件を含めるようにしてください。

## 手順

1. 典型的な要求のストレージ要件を計算します。  
要求に対して大量のレコードを生成する可能性があるリソースを選択して、レコードの数に各レコードのサイズを掛けます。レコード・サイズを決定するには、[CICSplex SM リソース・テーブル](#)の適切な表を参照してください。
2. 保持されている結果セットの 64 ビット・ストレージの合計所要量を計算します。  
単一の要求に対して保持されている結果レコードの予想最大数を見積もり、その数値に各要求の予想ストレージ数を掛けます。例えば、単一の要求で 100,000 個の CICS 端末を持てる場合、この数値にステップ 1 で決定したリソース・レコードのサイズを掛けます。
3. CICSplex SM 環境に CMCI をセットアップする場合は、CMAS の補助ストレージ要件を見積もります。  
予想される同時要求の最大数を見積もり、その数値にステップ 2 で計算した各要求の予想ストレージ数を掛けます。同時要求の数は、任意の時間に CMCI を使用することが予想されるユーザーの総数、およびそれらのユーザーが発行する可能性がある同時要求の数から導き出すことができます。
4. より多くのストレージをメタデータ用に追加することを検討します。  
CMCI は、各要求で要求されたすべてのリソース・レコードを新しい保持されている結果のために保管します。新しいリソース・タイプの最初の要求では、例えば、CICS プログラムに対する最初の要求の場合、少量の属性メタデータも保管されます。大きな要求の場合、属性メタデータのサイズおよび要求を行うために使用されるその他のストレージ量は、レコード自体に必要なストレージと比べたらごくわずかなものです。WUI サーバーの CMCI で内部的に使用される追加メタデータをカバーするために、最終的な予想値に 2% 追加することを検討します。この追加されたメタデータは、CMAS 計算に必須のものではありません。

### CMCI JVM サーバーのストレージ要件

5. CMCI JVM サーバー は、WUI 領域で実行されている Liberty JVM サーバーです。JVM サーバーのストレージ要件を計算するには、[JVM サーバーのストレージ要件の計算](#)を参照してください。  
CMCI JVM サーバー の最大ヒープ・サイズとして推奨される値は、少なくとも 2 GB です。

## 次のタスク

- CICSplex SM 環境に CMCI をインストールしている場合は、補助ストレージの見積もりを使用して、WUI サーバーに関連付けられている CMAS の **MAXAUXCP** および **MAXAUXTOTL** パラメーターの値を設定します。
- 保持されている結果セットに必要な 64 ビット・ストレージに、最大ヒープ・サイズを加えた見積もり値を使用して、WUI 領域 (シングル・サーバー・バージョンの CMCI を実行している場合は CICS 領域) の **z/OS MEMLIMIT** 値を決定します。64 ビット・ストレージを使用する他の CICS 機能を考慮する必要があります。

CICS の **MEMLIMIT** 値、および CICS 領域に現在適用されている **MEMLIMIT** の値を確認する方法については、「[パフォーマンスの改善](#)」の『**MEMLIMIT** の見積もり、確認、および設定』を参照してください。  
**z/OS** での **MEMLIMIT** について詳しくは、「[z/OS MVS P プログラミング: 拡張アドレッシング機能ガイド](#)」の『**専用メモリー・オブジェクトの使用制限**』を参照してください。

## CMCI JVM サーバーに対するクライアント・ホワイトリストの定義

クライアント・ホワイトリストを使用して、CMCI JVM サーバーに接続できるクライアントを制限できます。例えば、どのレベルの CICS Explorer またはブラウザーが CMCI JVM サーバーに接続できるかを制限することができます。

注: この機能により、CMCI に接続できるクライアントを管理できますが、これで CMCI が保護されると期待することはできません。

## 始める前に

ご使用の CMCI 構成で CMCI JVM サーバー がセットアップされ、稼働状態になっている必要があります。CMCI JVM サーバー をセットアップするには、362 ページの『[WUI 領域の CMCI の構成](#)』または 367 ページの『[CMCI JVM サーバーを使用するための WUI 領域の構成](#)』の指示に従ってください。

## このタスクについて

クライアント・ホワイトリストは、CICS Explorer またはブラウザーなどのクライアントによって送信される有効なユーザー・エージェント HTTP ヘッダーのリストを格納する、ASCII ファイルです。

**com.ibm.cics.jvmserver.cmci.user.agent.white.list** JVM プロパティを使用して、ホワイトリスト・ファイルの場所を指定します。プロパティが CMCI JVM サーバー の JVM プロファイルに定義されていない場合は、すべてのクライアントが受け入れられます。

ユーザー・エージェントがファイル内にない場合、要求は HTTP コード 403 で拒否されて、メッセージ DFHSJ1412 が発行されます。

**com.ibm.cics.jvmserver.cmci.user.agent.white.list.reject.text** JVM プロパティを使用して、ユーザーに返す代替応答テキストを指定できます。

これらの JVM システム・プロパティについては、[JVM システム・プロパティ](#)を参照してください。

注：CICS Explorer ユーザー・エージェントは、複数の CICS Explorer コンポーネントの各バージョンをエンコードするので、コンポーネントが更新されると変更されることがあります。IBM CICS Explorer for Aqua V3.1<sup>1</sup> (フィックスパック 5.4.0.5) 以降では、「ヘルプ」>「バージョン情報」>「インストール詳細」>「構成」をクリックして、実行中の CICS Explorer インストール済み環境で表示されるユーザー・エージェントを検出できます。CMCI ユーザー・エージェント・ストリングは、「インストール詳細」ダイアログ・ボックスの「構成」タブにリストされます。

## 手順

1. クライアント・ホワイトリスト・ファイルを定義します。

ファイルでは、行の先頭で番号記号 (#) を使用して、コメントを指定できます。また、項目の最後の文字として、アスタリスク (\*) をワイルドカードとして使用することもできます。ファイルは ASCII ファイル・エンコード方式で保管する必要があります。

```
# CICS Explorer User-Agent header
IBM_CICS_Explorer/5.5.6.201912070533 IBM_zOS_Explorer/3.2.9.201912061644 JRE/1.8.0_211 (Windows 7)

IBM_CICS_Explorer/5.5.6.201912070533 IBM_zOS_Explorer/3.2.9.201912061644 JRE/1.8.0_211 (Windows 8)

IBM_CICS_Explorer/5.5.6.201912070533 IBM_zOS_Explorer/3.2.9.201912061644 JRE/1.8.0_211 (Mac OS X)
```

図 61. CICS Explorer がクライアントであるクライアント・ホワイトリスト・ファイルの例

```
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:65.0) Gecko/20100101 Firefox/65.0
```

```
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36
```

```
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_2) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.0.2 Safari/605.1.15
```

図 62. ブラウザーがクライアントであるクライアント・ホワイトリスト・ファイルの例

2. 以下のように、ファイルの場所を指定します。

CMCI JVM サーバーの JVM プロファイル EYUCMCIJ.jvmprofile 内の次の行のコメントを外し、ファイルの場所を指定します。

```
-Dcom.ibm.cics.jvmserver.cmci.user.agent.white.list=/var/userAgentWhiteList
```

ホワイトリストに含まれていないクライアントからユーザーが CMCI JVM サーバーに接続しようとする、要求は HTTP コード 403 で拒否されます。カスタム応答テキストをユーザーに返す場合は、[ステップ 384 ページの『3』](#)から続行します。

<sup>1</sup> Aqua は IBM Explorer for z/OS Aqua を指します。



3. オプション: CMCI JVM サーバーの JVM プロファイル EYUCMCIJ.jvmprofile に次の行を追加します。ここで、*response\_text* はユーザーに対するメッセージです。

```
-Dcom.ibm.cics.jvmserver.cmci.user.agent.white.list.reject.text=response_text
```

## タスクの結果

ホワイトリストで処理される有効なユーザー・エージェントごとに、メッセージ DFHSJ1410I が発行されます。ホワイトリストに定義されているユーザー・エージェントのみが CMCI に接続できます。

**トラブルシューティング:** CMCI に対するアクセスを許可されたユーザー・エージェントがない場合、クライアントのホワイトリスト構成に問題がある可能性があります。例えば、**com.ibm.cics.jvmserver.cmci.user.agent.white.list** JVM プロパティに指定されたファイルが見つからないか、そのファイルに無効な値が含まれていることがあります。そのような場合、CSMT に対してメッセージ DFHSJ1408 が出力され、CMCI への接続を試みるユーザーにはメッセージ DFHWU4303 が出力されます。

## 次のタスク

### ホワイトリストの更新

このファイル内のホワイトリスト値はキャッシュに保持され、デフォルトでは Liberty キャッシュ・ファイル・モニタリングによって更新されます。Liberty キャッシュ・ファイル・モニタリングは、デフォルトでは 10 秒ごとに、ファイルが変更されたかどうかを検査します。

リストを更新する必要がある場合は、CMCI JVM サーバーの JVM プロファイルで次の Java システム・プロパティを設定することによって、デフォルトをオーバーライドすることができます。

```
-Dcom.ibm.cics.jvmserver.cmci.user.agent.white.list.monitor.interval=20s
```

## スタンドアロン CICS 領域での CMCI のセットアップ

スタンドアロン領域 (SMSS) で CICS 管理クライアント・インターフェース (CMCI) をセットアップすることができます。この構成では CMCI JVM サーバーはサポートされていません。

### このタスクについて

SMSS では CMCI をセットアップするためのサンプル・リソース定義が提供されています。サンプルは、グループ DFH\$WU の CICS システム定義ファイル (CSD) に組み込まれています。

- DFH\$WUUR はサンプル URIMAP 定義です。
- DFH\$WUTC はサンプル TCPIP SERVICE 定義です。

これらのリソースはそのままインストールできます。通常は、コピーして変更を加え、自分の環境に合わせます。ただし、URIMAP パスは常に CICSSystemManagement/\* として指定する必要があります。

デフォルトで提供される手順では、インターフェースのセキュリティはアクティブになりません。セキュリティがアクティブでない場合、検査システムのプログラミング・インターフェース・コマンドによって作成されるメッセージには、領域のデフォルトのユーザー ID が含まれます。

SMSS で CMCI のセキュリティをセットアップするために、386 ページの『[スタンドアロン CICS 領域での CMCI のセキュリティの構成](#)』に示されているような手順で、提供されているサンプルを調整することができます。

提供されるサンプルを使用して、SMSS で CMCI をセットアップするには、以下の手順を実行します。

### 手順

1. CICS 始動 JCL (ジョブ制御言語) を変更します。
  - a) *hlq*.CPSM.SEYUAUTH ライブラリーを STEPLIB 連結に追加します。ここで、*hlq* は、CICSTS56 などの高位修飾子です。

b) *hlq*.CPSM.SEYULOAD ライブラリーを DFHRPL 連結に追加します。ここで、*hlq* は、CICSTS56 などの高位修飾子です。

これらのライブラリーは CICS のライブラリーと同じ CICS TS レベル、つまり、STEPLIB 連結での CICS *hlq*.CICS.SDFHAUTH および CICS *hlq*.CICS.SDFHLOAD ライブラリーと同じでなければなりません。

2. CICS 領域に関するシステム初期設定パラメーター CPSMCONN が NO に設定されていることを確認します。  
CICS Explorer が CMCI スタンドアロン領域に接続するには、CPSMCONN を NO に設定しなければなりません。
3. CICS の始動時にグループ・リスト (GRPLIST) に組み込まれたリストに RDO グループ DFHFEPI がインストールされていることを確認します。DFHFEPI グループは、デフォルトの CICS 始動リスト DFHLIST に組み込まれています。このグループには、CMCI にとって必要なリソースが含まれています。
4. CICS 領域を始動します。
5. サンプル URIMAP 定義の DFH\$WUUR をインストールします。  
このサンプル URI マップは、トランザクション CWWU を使用し、プログラム DFHWBA を呼び出して、CICS Web 要求を分析します。DFH\$WUUR には、以下の属性値が含まれます。

属性名	属性値
説明	サンプルのシステム 管理インターフェース URI マップ
グループ	DFH\$WU
Host	*
パス	CICSSystemManagement/*
Port	No
Program	DFHWUIPG
スキーム	HTTP
Status	有効
TCP/IP サービス名	DFH\$WUTC
トランザクション	CWWU
URI マップ	DFH\$WUUR
使用法	サーバー

サンプル定義のこれ以外の属性は、デフォルト値を保持します。

注：TCP/IP サービス名は TCPIPSERVICE 定義に一致しなければなりません。DFH\$WUTC 以外の名前を持つ TCPIPSERVICE 定義を使用する場合には、それに応じて TCP/IP サービス名を URIMAP 定義で確実にリネームしてください。

6. サンプルの TCPIPSERVICE 定義である DFH\$WUTC をコピーしてリネームし、ポート番号を固有値に変更します。  
DFH\$WUTC には、以下の属性値が含まれます。

属性名	属性値
認証レベル	No
CICS トランザクション ID	CWXN
説明	サンプルのシステム 管理インターフェース TCPIP サービス
グループ	DFH\$WU
Host	任意

属性名	属性値
Port	1490
Protocol	HTTP
キュー・バックログ限度	0
SSL	No
Status	Open (オープン)
TCP/IP サービス名	DFH\$WUTC
ソケット・クローズに対するタイムアウト	No
ユーザー置き換え可能モジュール名	DFHWBAAX

サンプル定義のこれ以外の属性は、デフォルト値を保持します。

#### 7. TCPIPService 定義をインストールします。

##### ヒント:

- CODB、COD0、COD1、COD2、COSH、COLU、CWWU、CWXN のトランザクションを RACF に対して、または等価の外部セキュリティ・マネージャーに対して定義し、CMCI ユーザーがそれらのトランザクションにアクセスすることを確実に許可されるようにしてください。
- CICS が使用する DFHCNV テーブルに DFHWBUD エントリが確実に含まれるようにしてください。
- 独自のバージョンの DFHCNV ソース・モジュールを使用する場合は、新規マクロを使用してそれをアセンブルおよびリンク・エディットするか、または必要なコード・ページを含めるようにしてください。

### 次のタスク

CICS Explorer またはブラウザーを使用して、セットアップが正しく機能するか検査します。

必要に応じて、以下の検査を使用できます。

1. スタンドアロン領域で TCPIPService が OPEN になっており URIMAP が ENABLED になっているか検査します。ポートを確認し、URIMAP が TCPIPService をコピー済みの古い DFH\$WUTC リソースではなく新しい名前で参照していることを確認します。
2. ブラウザーで、以下の要求を使用して URL にアクセスします。ブラウザーから結果が返された場合は、セットアップは想定どおりに機能しています。CMCI URIMAP は、先頭が以下の URL に関する要求のみに応答します。

```
http://hostname:port/CICSSystemManagement/CICSTask/
```

3. 以下のような、領域ジョブ・ログ上の SMSS 関連メッセージを検査します。

```
EYUXL0022I xxxxxxxx SMSS Phase I initialization complete
EYUXL0007I xxxxxPhase II initialization complete
EYUNL0099I xxxxxxxx SMSS LRT initialization complete
```

4. ジョブ・ログと MSGUSR ログの両方にあるインストール・エラーを検査します。
5. CICS Explorer の CMCI ポートへの接続を試行して、この接続が機能するかどうか検査します。

## スタンドアロン CICS 領域での CMCI のセキュリティの構成

スタンドアロン CICS 領域 (SMSS) で CMCI のセキュリティを構成するには、サンプル定義の設定を変更する必要があります。HTTP 基本認証を使用するか、高水準のセキュリティ用の Secure Sockets Layer (SSL) 認証を使用するかを選択できます。

## このタスクについて

インターフェースへのアクセスを保護するために使用できるセキュリティ手段について詳しくは、[CICS Web サポートのセキュリティ](#)を参照してください。

SMSS で CMCI のセキュリティをセットアップするには、以下に示されている手順のようにして、CICS 提供のサンプル定義 DFH\$WUTC および DFH\$WUUR を調整することができます。

## 手順

1. サンプルの TCPIP SERVICE 定義、DFH\$WUTC、およびサンプルの URIMAP 定義、DFH\$WUUR をコピーしてリネームします。  
これらのサンプルは、グループ DFH\$WU の CICS システム定義ファイル (CSD) に組み込まれています。
2. TCPIP SERVICE 定義を変更して、使用するセキュリティ機能を取り込みます。  
Web クライアントのセキュリティを含む TCPIP SERVICE 定義を作成する方法については、[CICS Web サポートの TCPIP SERVICE リソース定義の作成](#)を参照してください。
3. URIMAP 定義内の TCPIP SERVICE 属性を変更して、リネームした TCPIP SERVICE を参照するようにします。  
SCHEME 属性を HTTP から HTTPS に変更することもできますが、この変更は、関連付けられている TCPIP SERVICE でセキュリティが有効になると自動的にインストール済みの URIMAP に加えられるので必須ではありません。
4. 定義を CICS 領域にインストールします。  
これらのリソース定義をインストールするために使用できるメソッドについての説明は、[Ways of defining CICS resources](#) を参照してください。  
**注:** CWWU と CWXN のトランザクションを RACF に対して、または等価の外部セキュリティ・マネージャーに対して定義し、CMCI ユーザーがそれらのトランザクションにアクセスすることを確実に許可されるようにしてください。

## タスクの結果

セキュリティがアクティブの場合、検査システムのプログラミング・インターフェース・コマンドによって作成されるメッセージには、CICS Explorer にログオンするのに使用するユーザー ID が含まれます。



## 第 10 章 イベント処理のセットアップ

イベント処理を使用するには、その前に CICS システムを正しく構成しておく必要があります。さまざまな方法を使って CICS でイベント処理を停止/開始することができます (例えば、IBM CICS Explorer を使用できます)。

### このタスクについて

アップグレードまたはシステム・メンテナンスのためにイベント処理を停止して、その後、再びイベント処理を開始することができます。このタスクでは、CICS Explorer を使用するプロセスを説明します。

LOCALCCSID システム初期設定パラメーターを使用して、ローカル CICS 領域のコード化文字セット ID を指定します。これはアプリケーション・プログラムのデフォルト・コード・ページです。LOCALCCSID のデフォルト設定は、EBCDIC コード・ページ 037 です。CICS は、イベント・キャプチャー・フィルター操作に使用する前に、イベント・バインディング・ファイルのデータ内容をこのコード・ページに変換します。キャプチャーされる文字データはすべてコード・ページであることを想定します。

CICS システムの始動時に START=INITIAL または START=COLD パラメーターが使用される場合、デフォルトでイベント処理が有効になります。

START=WARM または START=EMERGENCY パラメーターが使用される場合、前回の CICS 実行時の設定が使用されます。

**注:** 同期的な複数のトランザクション・イベントをキャプチャーする作業単位が進行中である間は、イベント処理の状況の変更 (つまり開始、ドレーン、または停止の設定) を行わないでください。それを行った場合、イベントがバックアウトされてトランザクションが異常終了する可能性があります。

### 手順

このタスクを使用して、イベント処理を停止または開始します。

1. CICS Explorer 内の CICSplex Explorer ビューを使用して、イベント処理を停止する CICSplex または CICS 領域をクリックして、その CICS 領域を選択します。
2. CICS Explorer ツールバーを使用して、「操作」 > 「イベント処理 (Event Processing)」の順にクリックします。

イベント処理の状況が表示されます。

3. イベント処理を停止する領域を右クリックして選択し、「停止」をクリックします。次に「OK」をクリックします。

**注:** 最新表示アイコンをクリックすると状況が更新され、「イベント処理状況」列に STOPPED 状況が表示されます。これはイベント処理が停止していることを示します。

4. イベント処理を再始動するには、イベント処理の開始対象となる領域を右クリックして選択し、「開始」をクリックします。次に「OK」をクリックします。

**注:** 最新表示アイコンをクリックすると状況が更新され、イベント処理が開始していることを示す STARTED 状況が表示されます。

### タスクの結果

イベント処理が STOPPED になった場合、イベント・キャプチャーは直ちに停止します。ディスパッチャー・キューにあるすべてのイベントが削除されます。

イベント処理が STARTED になった場合、未完了トランザクションに関しては、非トランザクション・イベントのキャプチャーは直ちに開始され、トランザクション・イベントのキャプチャーは次の同期点で開始されます。

### 次のタスク

イベント・コンシューマーで処理されるイベントを発行できるようにするには、その前に、イベント・バインディングおよび EP アダプターを有効にしておく必要があります。詳しくは、[CICS Explorer 製品資料](#)



内の『[Enabling an event binding](#)』および CICS Explorer 製品資料内の『[Enabling an EP adapter](#)』を参照してください。

WebSphere® MQ EP アダプターを使用してイベントのフォーマットと処理を行っている場合は、CICS 領域と z/OS 上の IBM MQ との接続を提供するための CICS-WebSphere MQ アダプターの設定が必要です。これを行う手順については、[CICS-MQ アダプターのセットアップ](#)を参照してください。

## 第 11 章 Link3270 ブリッジの構成

3270 ブリッジは、3270 端末を使用せずに 3270 ベースの CICS トランザクションを実行できるインターフェースを提供します。このセクションでは、Link3270 ブリッジの構成手順について説明します。

### Link3270 システム初期設定パラメーターの定義

システム 初期設定パラメーターをオプションで定義して、Link3270 ブリッジの管理に役立てることができます。

#### このタスクについて

通常、CICS はすべてのブリッジ機能を自動的に定義します。しかし必要なら、ユーザー置換可能モジュールを作成し、ブリッジ機能の自動インストールを制御することができます。

#### 手順

1. ブリッジ・ルーター領域で、**AIBRIDGE** システム 初期設定パラメーターを定義し、ブリッジ機能を作成および削除する際に、自動インストールのユーザー置換可能モジュール (URM) を呼び出すかどうかを指定します。
2. ブリッジ (BR) ドメインやパートナー (PT) ドメインの標準的および特殊なトレースが必要な場合、**SPCTR** および **STNTR** システム 初期設定パラメーターを定義します。

#### タスクの結果

URM を使用してブリッジ機能および追加のトレースをインストールするために、CICS を構成しました。

### ブリッジ機能の定義

ブリッジ機能は、エミュレート 3270 端末です。これは、ブリッジ機能を割り振る単一トランザクション・モード要求またはセッション・モード要求を受け取ると、DFHL3270 によって作成される仮想端末です。

ブリッジ機能の **TERMINAL** リソース定義を提供できませんが、テンプレートとして使用される 3270 **TERMINAL** リソース定義を提供することによって、使用される端末プロパティを制御できます。この **TERMINAL** 定義は、**facilitylike** と呼ばれます。

#### 機能の定義

**facilitylike** の値は、ブリッジ機能の一部のプロパティのテンプレートとして使用する実際の端末リソース定義の名前です。

使用する **facilitylike** 定義の名前は、次の 3 つの方法のいずれかによって CICS に渡すことができます (検出された最初の非ブランク値が使用されます)。

- Link3270 呼び出しで **BRIH-FACILITYLIKE** パラメーターから渡す。
- ユーザー・トランザクションの **PROFILE** リソース定義から渡す。
- デフォルトは、ブリッジをサポートするために CICS に用意されている定義 **CBRF** です。

ブリッジ機能が定義された後に、その **facilitylike** テンプレートを変更することはできません。このため、ブリッジ機能がセッション・モードで再利用される場合、CICS は、2 回目以降の呼び出しで渡された **facilitylike** 値を無視します。

注：VTAM=NO システム 初期設定テーブル・パラメーターを使用して開始された CICS システムで 実行する場合は、**facilitylike** で指定するリソース定義を **REMOTE** として定義する必要があります。**REMOTE** として定義されたデフォルトの定義 **CBRF** が、グループ **DFHTERM** に用意されています。

注：VTAM は現在 z/OS Communications Server です。

**PROFILE** リソース定義については、[PROFILE リソース](#)を参照してください。

## ブリッジ機能名の定義

CICS は、ブリッジ機能を作成するときに、その機能を識別するための 8 バイト・トークン (*facilitytoken*) と 4 文字の端末 ID (TERMID と NETNAME の両方として使用されます) の両方を作成します。

*facilitytoken* は Link3270 割り振り呼び出しで返されます。ユーザーは後続のすべてのセッション・モード呼び出しでこれを指定する必要があります ([Link3270 セッション・モードの使用](#)を参照)。

Link320 ブリッジで作成されるブリッジ機能の場合、トークンと名前はその CICSplex において固有であり、TERMID および NETNAME の形式は AAA} です。命名は、セッション・モードの場合は **allocate** コマンドを処理する際に、単一トランザクション・モードの場合は内部割り振りステップの際にルーティング領域で行われます。セッション・モードおよび単一トランザクション・モードの詳細については、[Link3270 ブリッジの使用](#)を参照してください。

Link3270 ブリッジ機能の名前空間割り振りについての情報は、名前の固有性を保つために共用ファイル DFHBRNSF に記録されます。ファイル DFHBRNSF とその関連した AOR を共用する複数のルーター領域によって、ブリッジ CICSplex が作成されます。大規模な CICSplex の中に複数のブリッジ CICSplexes をセットアップし、ブリッジ CICSplex 内の各ルーター領域で同じ DFHBRNSF ファイルを共用することができます。1 つのブリッジ CICSplex の AOR 領域は、1 つのブリッジ CICSplex のルーター領域とのみ関連付けることができます。

パフォーマンスを向上させるため、Link3270 ブリッジ名前空間は複数の割り振り範囲に分割されています。このため、名前の「チャンク」が各ルーター領域に割り振られ、DFHBRNSF ファイルは、名前空間範囲が割り振られるか、解放されたときにのみ、アクセスされます。保持期間が終了したら、割り振られたチャンク内の名前を再使用できます。また、他の領域でもチャンクを再使用することができます。このため、異なる領域に同じ名前が表示される場合がありますが、これらの名前は常に 1 つの領域でのみアクティブになります。

DFHBRNSF の名前の 90% が割り振られると、メッセージ DFHBR0505 が発行され、割り振られた名前のパーセンテージが 1 ポイント増加するたびに発行されます。メッセージ DFHBR0506 は、割り振られた名前のパーセンテージが 1 ポイント減少するたびに (90% 未満になるまで) 発行されます。使用可能な名前がなくなると、メッセージ DFHBR0507 が発行され、クライアント・アプリケーションの新規の割り振り (または 1 回限りの) 要求は戻りコード BRIHRC\_NO\_FREE\_NAME を受け取ります。

### DFHBRNSF ファイル・タイプ

ブリッジ機能の名前空間割り振りファイル (DFHBRNSF) には、ローカル・ユーザー・データ・テーブル、ローカル VSAM ファイル、カップリング・ファシリティ・データ・テーブル (CFDT)、リモート VSAM ファイル、または VSAM RLS ファイルを指定できます。

ルーター領域を 1 つのみ使用する場合は、ユーザー保守データ・テーブルまたはローカル VSAM バージョンの DFHBRNSF をお勧めします。

複数のルーター領域を使用する場合は、DFHBRNSF ファイルをリモートのファイル所有領域 (FOR) のローカル VSAM ファイルとして、また、ルーター領域のリモート VSAM ファイルとして定義できます。すべてのルーター領域の VSAM RLS として定義することも、すべてのルーター領域のカップリング・ファシリティ・データ・テーブルとして定義することもできます。

DFHBRNSF のユーザー保守データ・テーブル・バージョンを使用する場合、CICS ファイル定義で指定されている VSAM データ・セットは空でなければならず、ファイルからデータ・テーブルにレコードがロードされてはなりません。ルーター領域が実行されているときにデータ・テーブルを閉じないでください。閉じると、すべてのブリッジ機能の名前空間データが失われ、ブリッジ機能の範囲の割り振り/解放を求める次の要求が失敗します。そのため、ユーザー保守データ・テーブルを実稼働環境で使用することはお勧めしません。

### DFHBRNSF の定義

VSAM ファイルおよびデータ・テーブルの場合、IDCAMS を使用してファイル DFHBRNSF を作成する必要があります。

393 ページの図 63 にいくつかのサンプル IDCAMS ステートメントを示しています。これらを変更して DFHBRNSF ファイルを作成できます。使用環境の基準に準拠させるために、クラスター名およびボリューム値を変更してください。

```
//DEFDS      JOB   accounting info,name,MSGCLASS=A
//TDINTRA    EXEC  PGM=IDCAMS
//SYSPRINT   DD    SYSOUT=A
//SYSIN      DD    *
              DEFINE CLUSTER(NAME(CICSTS22.CICS.DFHBRNSF) -
                              INDEXED-
                              TRK(1 1)-
                              RECORDSIZE(384 384)-
                              KEYS(13 20)-
                              FREESPACE(0 50)-
                              SHAREOPTIONS(2 3)-
                              LOG(NONE)-
                              VOLUME(DISK01)-
                              CISZ(512)) -
              DATA  (NAME(CICSTS22.CICS.DFHBRNSF.DATA)-
                      CISZ(512)) -
              INDEX  (NAME(CICSTS22.CICS.DFHBRNSF.INDEX)-
                      CISZ(512))

/*
//
```

図 63. DFHBRNSF ファイルを作成するためのサンプル IDCAMS ジョブ

カップリング・ファシリティ・データ・テーブルの作成方法については、[カップリング・ファシリティ・データ・テーブル](#)を参照してください。

ファイル DFHBRNSF には、制御レコード 2 つと、ファイルにアクセスするルーター領域ごとに 1 つのレコードが含まれます。DFHBRNSF に書き込み可能なレコードの最大数は 731 です (2 つの制御レコードもこれに含まれます)。

ファイル DFHBRNSF を CICS の Link3270 ルーター領域に定義する必要があります。すべてのバージョンのファイル用のリソース定義が用意されています。適切なグループを始動グループ・リストに含めるか、選択した定義をグループ・リスト内のグループに含める必要があります。IDCAMS ステートメントと一致するように定義を編集する必要があります。CFDT バージョンのファイルを使用する場合を除き、ファイルを作成する IDCAMS で使用したクラスター名と一致するように DSN フィールドを変更してください。CFDT 定義を使用する場合は、CFDTPOOL 値をファイル定義で定義したテーブルを含むプールの名前に変更します。この表は、DFHBRNSF 定義を含むグループとして用意されているものを示しています。

表 36. DFHBRNSF 用に用意されているリソース定義

グループ	タイプ
DFHBRVR	VSAM RLS
DFHBRVSL	ローカル VSM、非 RLS
DFHBRVSR	リモート VSAM、非 RLS
DFHBRCF	カップリング・ファシリティ・データ・テーブル
DFHBRUT	ユーザー保守のデータ・テーブル

注：

1. DFHBRNSF が使用不可になった場合は、ブリッジ機能の名前空間範囲の割り振り/解放を行わない Link3270 要求のみが正常に完了します。
2. DFHBRNSF ファイルを何らかの理由で再定義する必要がある場合は、ファイルを再定義する前にファイルにアクセスするすべてのルーター領域をシャットダウンし、ファイルを再定義した後に再始動する必要があります。

## CICS 終了時の DFHBRNSF

CICS ルーター領域が正常終了する間、新しい Link3270 要求は拒否され、既存の Link3270 機能が解放されます。すべての機能が解放されると、ルーター領域に関連付けられた DFHBRNSF 名前範囲が解放されます。

DFHBRNSF 名前範囲の解放は失敗することがあります。例えば、ファイル DFHBRNSF がリモートにある場合、シャットダウン・トランザクション CESD は、名前領域が解放される前にファイル所有領域への接続を切断します。これが発生すると、エラー・メッセージが発行されます。CICS 領域が再始動し、Link3270 トランザクションが領域で実行している場合は、名前範囲のみ解放されます。

CICS が即時終了する場合は、Link3270 要求は戻りコード BRIH-CICS-TERMINATION で拒否されます。ただし、Link3270 機能および機能に関連した DFHBRNSF 名前範囲は解放されません。CICS 領域が再始動し、Link3270 トランザクションが領域で実行している場合は、名前範囲のみ解放されます。

## 特定のブリッジ機能名の定義

3270 端末の名前またはネット名が 3270 アプリケーションのロジックにとって重要である場合は、Link3270 呼び出しの BRIH-TERMINAL または BRIH-NETNAME パラメーターで特定の名前を渡すことができます。また、オプションでブリッジ機能の割り振り時に自動インストール・ユーザー置換可能モジュール (URM) を呼び出すように要求することもできます。

CICS 始動時にシステム初期設定パラメーター AIBRIDGE=YES を指定すると、自動インストール URM が呼び出されます。また、SET AUTOINSTALL を使用して、後でこのオプションをアクティブにすることもできます。

AUTOINSTALL URM は、指定または生成された端末名およびネット名を許可、拒否、または変更できます。自動インストール URM について詳しくは、[ブリッジ・ファシリティ 仮想端末はどのように自動インストールされるか](#)を参照してください。

## TCTUA の初期化

ブリッジ機能に、EXEC CICS ADDRESS TCTUA を使用して通常の方法でアクセスできる端末管理テーブル・ユーザー域 (TCTUA) を備えることができます。

ブリッジ機能の作成時に、TCTUA はヌルに初期化されます。ブリッジ機能が作成および破棄されるときに、XFAINTU というグローバル・ユーザー出口 (GLUE) が呼び出されます。XFAINTU は TCTUA のアドレスに渡されるため、この出口を使用して TCTUA を初期化できます。

## ブリッジ機能プロパティへのアクセス

基本機能 (ブリッジ機能) に関する情報をユーザー・トランザクションで EIB から取得することができます。また、基本機能が実際の 3270 である場合に通常実行されるのと同じように、INQUIRE および ASSIGN コマンドを使用して取得することもできます。

例えば、TERMINID は、EIBTERMINID、ASSIGN FACILITY コマンド、INQUIRE TASK FACILITY コマンド、または INQUIRE NETNAME コマンドから取得できます。NETNAME は ASSIGN NETNAME または INQUIRE TERMINAL を使用して取得できます。

INQUIRE BRFACILITY コマンドを使用すると、機能トークンで識別される任意のブリッジ機能に関する情報を取得できますが、他のすべての INQUIRE コマンドは、そのコマンドを発行したトランザクションの基本機能であるブリッジ機能に関する情報のみを返します。他のトランザクションにとって、ブリッジ化された環境で実行されているトランザクションは非端末トランザクションのように見えるので、別のトランザクションによって発行されたブリッジ機能の TERMINID に対して INQUIRE TERMINID を実行すると、TERMINIDERR になります。INQUIRE NETNAME および INQUIRE TASK も同じように動作します。

INQUIRE TERMINAL による参照への応答に、ブリッジ機能は現れません。

ASSIGN および INQUIRE のすべてのキーワードがサポートされます。それらは、FACILITYLIKE 端末定義からブリッジ機能に設定された値、またはトランザクションの実行時に設定された値を返します。

一部のキーワードは、ブリッジ環境では、CICS によって固定された値を返します。以下に示します。

表 37. INQUIRE TERMINAL 値	
キーワード	戻り値
ACQSTATUS	ACQUIRED
ACCESSMETHOD	VTAM
CORRELID	ブランク
EXITTRACING	NOTAPPLIC
LINKSYSTEM	ブランク
MODENAME	ブランク
REMOTENAME	ブランク
REMOTESYSTEM	ブランク
REMOTESYSNET	ブランク
SERVSTATUS	INSERVICE
TCAMCONTROL	X'FF'
TERMSTATUS	ACQUIRED
TTISTATUS	YES
ZCPTRACING	NOZCPTRACE

注：VTAM は現在 z/OS Communications Server です。

表 38. INQUIRE TASK 値	
キーワード	戻り値
FACILITY	ブリッジ機能名
FACILITYTYPE	TERM または TASK
STARTCODE	S、SD、TO、TP

## QUERY

リストされているキーワードは、実際のデバイスのログオン時に 3270 Query 関数によって設定できる端末属性を表します。

ALTSCRNHT

ALTSCRNWD

APLKYBDST

APLTEXTST



BACKTRANSST	COLORST	EXTENDEDSSST	GCHARS
GCODES	HIGHLIGHTST	MSRCONTROLST	OUTLINEST
PARTITIONSST	PROGSYMBOLST	SOSIST	VALIDATIONST

ブリッジ機能の作成時に、実際の FACILITYLIKE 端末でログオンすると、QUERY が実行され、戻り値がブリッジ機能に適用されます。

実際の FACILITYLIKE 端末がブリッジ機能の作成時にログオンしなかった場合、QUERY は実行されず、FACILITYLIKE リソース定義からの値を使用してブリッジ機能が作成されます。

### SET TERMINAL/NETNAME

以下の表は、ブリッジ機能についてユーザー・トランザクションから SET TERMINAL/NETNAME キーワードを実行した場合の影響をキーワードごとに示しています。特に記載がない限り、応答は DFHRESP(NORMAL) です。

キーワード	効果
ACQSTATUS	無視されます。
ALTTPRINTER	値は SET で、INQUIRE で戻されますが、CICS では使用されません。
ALTPRTCOPIST	値は SET で、INQUIRE で戻されますが、CICS では使用されません。
ATISTATUS	通常の 3270 と同じように動作します。
CANCEL	無視
CREATESESS	無視されます。
DISCREQST	値は SET で、INQUIRE で戻されますが、CICS では使用されません。
EXITTRACING	無視されます。
FORCE	無視されます。
MAPNAME	通常の 3270 と同じように動作します。
MAPSETNAME	通常の 3270 と同じように動作します。
NEXTTRANSID	通常の 3270 と同じように動作します。
OBFORMATST	通常の 3270 と同じように動作します。
PAGESTATUS	無視されます。
PRINTER	値は SET で、INQUIRE で戻されますが、CICS では使用されません。
PRTCOPIST	値は SET で、INQUIRE で戻されますが、CICS では使用されません。
PURGE	無視されます。
PURGETYPE	無視されます。
RELREQST	値は SET で、INQUIRE で戻されますが、CICS では使用されません。
SERVSTATUS	通常の 3270 と同じように動作します。
TCAMCONTROL	通常の 3270 と同じように INVREQ を戻します。
TERMPRIORITY	値は SET で、INQUIRE で戻されますが、CICS では使用されません。
TERMSTATUS	無視されます。
TRACING	値は SET で、INQUIRE で戻されますが、CICS では使用されません。

キーワード	効果
TTISTATUS	無視されます。
UCTRANST	通常の 3270 と同じように動作します。
ZCPTRACING	無視されます。



## 第 12 章 EXCI の構成

外部 CICS インターフェース (EXCI) は、MVS で実行される CICS 以外のプログラム (クライアント・プログラム) が、CICS 領域で実行されるプログラム (サーバー・プログラム) を呼び出し、通信域を使用したりチャネルとコンテナのセットを使用したりしてデータを送受信できるようにするアプリケーション・プログラミング・インターフェースです。このセクションでは、EXCI の構成手順について説明します。

### EXCI の静的ルーティングのセットアップ

EXCI を使用するアプリケーションからの要求を CICS プログラムに静的に転送することができます。

#### 始める前に

始めに、MVS パラメーター **Maxmember** が高い値に設定されていることを確認します。このパラメーターは、DFHIRP00 リソースに対して確立できる接続の数を制御します。

#### 手順

1. RDO グループ EXCIXXXX を CICS 領域のグループリストに追加します。EXCIXXXX がない場合は、用意されている DFH\$EXCI RDO グループからコピーを作成します。

このグループは、EXCI 機能に必要なすべての接続を含むものであり、バッチ要求用に最大 100 の接続をサポートできます。

2. アプリケーションの RDO グループを CICS 領域のグループリストに追加します。
3. DFHXCOPT を SDFHEXCI ロード・ライブラリーにアセンブルします。DFHXCOPT に SURROGATE=YES が含まれていることを確認します。
4. アプリケーション・プログラムをアセンブルし、コンパイルします。  
アセンブラで作成したアプリケーション・プログラムの場合は、リンケージ・エディター・パラメーター AMODE(31) および RMODE(ANY) を使用してください。そのプログラムをアプリケーション・ロード・ライブラリーにリンクします。
5. アプリケーション・プログラムを実行するためのバッチ JCL を構成します。

- a) JCL を編集して、バッチ・プログラムが EXCI 要求を送信する CICS 領域を指定します。

```
//step0010 EXEC PGM=program,PARM='applid,userid'
```

*applid* は CICS 領域であり、*userid* は RACF ユーザー ID です。

- b) ロード・ライブラリーを次のように連結します。

```
//STELIB DD Disp=shr,Dsn=SYS5C.CICn.CICS730.SDFHEXCI  
//DD Disp=shr,Dsn=Your.application.loadlib
```

6. バッチ・プログラムを実行し、予期した結果になることを確認します。

### EXCI の動的ルーティングのセットアップ

CICSplex SM を使用して、EXCI を使用するアプリケーションからの要求を CICS プログラムに動的に転送することができます。

#### このタスクについて

#### 手順

1. CICS 領域で、次のシステム初期設定パラメーターを指定します。
  - DSRTPGM=EYU9XLOP
  - DTRPGM=EYU9XLOP
2. RDO グループを以下のように更新します。

- a) EXCI サーバー・プログラム DFHMIRS のための RDO エントリーを追加します。

必要に応じて、EXCI の代わりに別のトランザクションを使用することもできますが、そのトランザクションではプログラム DFHMIRS を指し、プロファイル DFHCICSA を使用する必要があります。グループ DFH\$EXCI の EXCI トランザクションの後にそれをモデル化します。このトランザクションは、DFHMIRS プログラムを指します。

注：リモートとして定義されたユーザー・トランザクションは、APPC 構成内でのみ動作します。EXCI または MRO 構成内では、REMOTE 属性を指定せずに DYNAMIC(NO) 属性を指定して、ミラー・トランザクションがローカルの CICS 領域で実行されるようにする必要があります。ミラー・トランザクションを別の CICS 領域にルーティングすると、パフォーマンスに影響が及んだり、問題判別が難しくなったりする可能性があります。

- b) RDO グループを端末専有領域 (TOR) LIST に追加します。

TOR 領域のプログラムの自動インストールが無効になっていることを確認します。

3. CICSplex SM にログインし、このトランザクションに、CPSM グループ (TXNGRP、WLMDEF、WLMGRP、WLMSPEC) へのルーティング領域を定義します。
4. アプリケーション所有領域 (AOR) ごとに、サンプル定義 DFH\$EXCI と同じ方法で RDO グループを作成します。

このグループには、接続、セッション、およびアプリケーション・プログラム、または同等のトランザクションのみが含まれている必要があります。EXCI の代わりに別のトランザクションを使用した場合は、そのトランザクションをこのグループに指定する必要があります。

### タスクの結果

アプリケーションが分散プログラム・リンクを発行した場合、CICSplex SM は着信トランザクションがその制御下にあるかどうかを検査します。EXCI またはこれと同等のものがトランザクション・グループで有効になっていることを検出すると、CICSplex SM はそのトランザクションを候補 AOR の 1 つに転送して処理します。

この方法の代わりに、プログラムまたはトランザクションの RDO 定義に DYNAMIC=Yes を指定することもできます。CICSplex SM は選択した領域にその要求を転送します。

## CICS への接続の定義

EXCI クライアント・プログラムと CICS 領域の間の接続には、CICS 領域での接続定義が必要です。それらを定義するには、CICS が備える CONNECTION および SESSIONS リソース定義機能を使用します。

外部 CICS インターフェースのために 特別に以下のオプションが用意されています。

- CONNECTION リソース定義の CONNTYPE
- CONNECTION および SESSIONS リソース定義の PROTOCOL 属性の EXCI

### EXCI の CONNECTION リソース定義

CONNECTION リソース定義の PROTOCOL 属性には、外部 CICS インターフェースを使用する MVS プログラムのための接続であることを示す EXCI オプションが用意されています。

CONNECTION リソース定義には CONNTYPE 属性が用意されています。EXCI 接続では、これを使用して、接続が汎用接続と固有接続のどちらであることを示します。これは、外部 CICS インターフェース以外のプロトコルには使用されません。

EXCI に関連するパラメーターは以下のとおりです。

#### CONNTYPE({SPECIFIC|GENERIC})

外部 CICS インターフェース 接続の場合に、その接続の性質を表します。

#### **SPECIFIC**

この接続は非 CICS クライアント・プログラムから CICS 領域への通信用で、かつ、固有です。固有接続とは、クライアント・プログラムで単一のユーザーに占有される 1 つ以上のセッションが含まれる MRO リンクのことをいいます。

注: ユーザーとは、TCB ごとに固有の名前で Initialize\_User 要求を発行したプログラム (または Initialize\_User 要求を発行する対象になったプログラム) のことです。例えば、次のとおりです。

- MVS の下で実行される単純なクライアント・プログラムが、外部 CICS インターフェースの単一ユーザーになることが可能です。
- MVS の下で実行される 1 つのクライアント・プログラムで複数のパイプを開き、それらを使用して、さまざまなベンダー・パッケージのために外部 CICS インターフェース呼び出しを順次発行することができます。その場合、クライアント・プログラムの観点から見ると、各パッケージが、固有のユーザー名で識別されるユーザーに相当します。つまり、1 つのクライアント・プログラムが複数のユーザーの代理として機能できます。
- MVS の下で実行される 1 つのプログラムが複数の TCB を接続し、各 TCB でベンダー・パッケージが自身のために外部 CICS インターフェース呼び出しを発行することができます。各パッケージは、独立したクライアント・プログラムであり、独自の TCB で実行されます。各パッケージは、固有のユーザー名を持つユーザーでもあります。

固有接続では、NETNAME が必須です。

### GENERIC

接続は非 CICS クライアント・プログラムから CICS システムへの通信用で、汎用です。汎用接続とは、複数の EXCI ユーザーが共用する複数のセッションが含まれる MRO リンクのことをいいます。汎用接続では NETNAME 属性を指定できません。

注: 1 つの CICS 領域にインストールする汎用 EXCI 接続は 1 つのみにする必要があります。

### NETNAME

外部 CICS インターフェース 接続の場合、NETNAME は、INITIALISE\_USER 呼び出しの *user\_name* パラメーターで指定された、固有のパイプのユーザーの名前に対応します。

外部 CICS インターフェースの固有パイプの場合、NETNAME を指定する必要があります。

外部 CICS インターフェースの汎用パイプの場合、NETNAME をブランクのままにする必要があります。

### PROTOCOL({APPC|LU61|EXCI|blank})

リンクで使用するプロトコルのタイプ。

#### ブランク

CICS 領域間で MRO を使用する場合。MRO の場合は PROTOCOL をブランクのままにしておく必要があります。また、SESSIONS 定義では、PROTOCOL として LU6.1 を指定する必要があります。

#### APPC (LUTYPE6.2 プロトコル)

拡張プログラム間通信 (APPC) プロトコル。これは、ACCESSMETHOD(VTAM) のデフォルト値です。CICS-CICS ISC にはこれを指定します。

注: VTAM は現在 z/OS Communications Server です。

#### LU61

LUTYPE6.1 プロトコル。これは、CICS-CICS ISC または CICS-IMS ISC について指定しますが、MRO には指定しません。

#### EXCI

外部 CICS インターフェース。これを指定して、この接続が外部 CICS インターフェースを使用する非 CICS クライアント・プログラム用であることを示します。

PROTOCOL(EXCI) を指定する場合は、ACCESSMETHOD(IRC) も指定しなければなりません。EXCI は、CICS 領域間通信プログラム DFHIRP を使用して MRO で実装されるため、MVS 仮想記憶間サービス (XM) を使用する MRO リンクは使用できません。EXCI は、同じく DFHIRP を介して動作する XCF MRO リンクも使用することができます。

## EXCI 接続の SESSIONS リソース定義

SESSIONS リソース定義の PROTOCOL 属性で、MRO 接続に割り振られたセッションが外部の CICS インターフェースで使用されるかどうかを示します。

SESSIONS リソース定義の詳細については、[SESSIONS リソース](#)を参照してください。EXCI に関連するパラメーターは以下のとおりです。



## **PROTOCOL({APPC|LU61|EXCI})**

相互通信リンク (ISC または MRO) で使用するプロトコルのタイプを 指定します。

### **APPC (LUTYPE6.2)**

拡張プログラム間通信機能 (APPC) プロトコル。CICS-CICS ISC にはこれを指定します。

### **LU61**

LUTYPE6.1 プロトコル。CICS-CICS ISC、CICS-IMS、または MRO の場合に 指定します。

### **EXCI**

外部 CICS インターフェース。これを指定すると、セッションが外部 CICS インターフェースを使用する非 CICS クライアント・プログラムによって使用されることが示されます。EXCI を指定する場合、SEND COUNT はブランクのままにしなければなりません。

## **RECEIVECOUNT({blank|number})**

通常、送信の前に受信する MRO、LUTYPE6.1、または EXCI セッションの数。

MRO の場合、受信セッションで受信できるのは送信前のみです。

### **blank**

これらのセッションは送信専用です。受信セッションはありません。

### **number**

CONNECTION 定義のプロトコル・パラメーターでブランク、LU61、または EXCI を指定した接続での受信セッションの数を指定します。CICS は、その数を使用してセッション名の最後の 2 文字または 3 文字を生成します (詳しくは、RECEIVEPFX を参照してください)。

デフォルトの受信接頭部 (<) または独自の 1 文字の接頭部を使用する場合は、1 から 999 までの範囲の数を指定してください。

2 文字の接頭部を指定する場合、数値の範囲は 1 から 99 までに制限されます。

外部 CICS インターフェース (EXCI) 接続を除き、このシステムの RECEIVECOUNT は、他のシステムの SEND COUNT と同じでなければなりません。

## **RECEIVEPFX(<|prefix)**

CICS が受信セッション名 (そのセッションの端末管理テーブルの端末入力 (TCTTE) の名前) の最初の 1 文字または 2 文字に使用する、1 文字または 2 文字の接頭部を指定します。

既存の接続名や端末名と競合するような接頭部は 指定できません。

### **< (MRO および EXCI セッション)**

MRO セッションでは、独自の受信接頭部を指定しないと、CICS はデフォルトの接頭部 (より小記号 (<)) を使用します。この記号に受信数を付加したものが、受信セッション名になります。

CICS は、セッション名の最後の 3 文字を、英数字 (A から Z、1 から 9) から作成します。この 3 文字の ID は文字 AAA から始まり、セッション項目数が RECEIVECOUNT 値で設定した限界に達するまで、昇順で続きます。受信セッション名が生成されるのは送信セッションの後であり、送信セッション名に続いて 同じ順序で生成されます。

例えば、送信セッションで最後に生成されるセッション名が、デフォルトの接頭部 (<) を使用した <AAJ だとすると、CICS は受信セッション名を <AAK、<AAL、<AAM のように生成します (セッション ID のこの生成方式は、最初の接頭部記号を除いては、APPC セッションでも同じです)。

注: 独自の接頭部を指定すると、CICS は以前のリリースの LUTYPE6.1 セッションの場合と同じ方法でセッション名を生成します。

### **prefix (LUTYPE6.1 セッション)**

セッションが LUTYPE6.1 ISC 接続上にある場合は、1 文字または 2 文字の接頭部を 指定する必要があります。デフォルトの < 記号は LUTYPE6.1 セッションには使用できません。

LUTYPE6.1 セッションでは (独自の 1 文字または 2 文字の接頭部を指定した場合は MRO でも)、CICS はその接頭部に番号 (1 から 99、または 1 から 999) を付加してセッション名を生成します。番号は 1 から始まり、指定した RECEIVECOUNT に達するまで 1 ずつ増えていきます。

## **SEND COUNT(blank|number)**

通常、受信する前に送信する MRO または LUTYPE6.1 セッションの数。

MRO の場合、送信セッションを送信しないと受信できるようになりません。

### **blank**

これらのセッションは受信専用です。送信セッションはありません。

セッションが外部の CICS インターフェース (EXCI) 接続に存在する場合は、このフィールドをブランクにしておく必要があります。

### **number**

CONNECTION 定義のプロトコル・パラメーターでブランクまたは LU61 を指定した接続での送信セッションの数を指定します。CICS は、その数を使用してセッション名の最後の 2 文字または 3 文字を生成します (詳しくは、SENDPFX を参照してください)。

デフォルトの送信接頭部 (>) または独自の 1 文字の接頭部を使用する場合は、1 から 999 までの範囲の数を指定してください。

2 文字の接頭部を指定する場合、数値の範囲は 1 から 99 までに制限されます。

外部 CICS インターフェース (EXCI) 接続を除き、送信システムの SENDCOUNT は、受信システムの RECEIVEDCOUNT と同じでなければなりません。

### **SENDPFX(>|prefix)**

CICS が送信セッション名 (そのセッションの端末管理テーブルの端末入力 (TCTTE) の名前) の最初の 1 文字または 2 文字に使用する、1 文字または 2 文字の接頭部を指定します。

既存の接続名や端末名と競合するような接頭部は指定できません。

#### **> (MRO セッション)**

MRO セッションでは、独自の送信接頭部を指定しないと、CICS はデフォルトの接頭部 (より大記号 >)) を使用します。この記号に送信数を付加したものが、送信セッション名になります。

CICS は、セッション名の最後の 3 文字を、英数字 (A から Z、1 から 9) から作成します。この 3 文字の ID は文字 AAA から始まり、セッション項目数が SENDCOUNT 値で設定した限界に達するまで、昇順で続きます。

例えば、デフォルト接頭部 (>) を使用した場合、CICS が生成するセッション名は >AAA、>AAB、>AAC のようになります。(セッション ID のこの生成方式は、最初の記号を除いては、APPC セッションの場合と同じです。)

注: 独自の接頭部を指定すると、CICS は以前のリリースの LUTYPE6.1 セッションの場合と同じ方法でセッション名を生成します。

#### **prefix (LUTYPE6.1 セッションの場合)**

セッションが LUTYPE6.1 ISC 接続上にある場合は、1 文字または 2 文字の接頭部を指定する必要があります。デフォルトの > 記号は LUTYPE6.1 セッションには使用できません。

LUTYPE6.1 セッションでは (独自の 1 文字または 2 文字の接頭部を指定した場合は MRO でも)、CICS はその接頭部に番号 (1 から 99、または 1 から 999) を付加してセッション名を生成します。番号は 1 から始まり、指定した SENDCOUNT に達するまで 1 ずつ増えています。

### **USERID(userid)**

リンクのセキュリティ検査に使用する事前設定のユーザー ID。

リンク・セキュリティの事前設定ユーザー ID を指定しない場合、CICS は、リモート・ユーザーから渡されたユーザー ID をリンク・セキュリティのユーザー ID として使用します。外部 CICS インターフェース・リンクの場合、これはクライアントのユーザー ID です。

## **EXCI 接続の状態の照会**

CICS 端末から CICS サーバー領域にアクセスできる場合は、クライアント・アプリケーション・プログラムを実行しているバッチ・ジョブについて照会し、CICS 内のサーバー・プログラムにリンクするために外部 CICS インターフェースを使用しているのがどのバッチ・ジョブかを照会できます。

## このタスクについて

MRO を介して CICS にリンクされているバッチ・ジョブに関するこの情報を取得するには、CEMT INQUIRE EXCI コマンドを使用します。このコマンドを使用すると、領域間通信 (IRC) 機能を介して現在 CICS に接続されている外部 CICS インターフェース・バッチ・ジョブの名前を調べることができます。

CICS は、ジョブ ID を次の形式で返します。

```
jobname.stepname.procname - mvsid
```

stepname、または procname、あるいはその両方が存在しない (ピリオド (.) が連続していることでわかります) 場合があります。

mvsid は、ジョブが実行されている MVS システムを指します。XCF/MRO を使用する場合、ジョブは CICS が実行されているイメージとは別の MVS イメージに存在できます。

外部 CICS インターフェースを使用するジョブに関する情報は、ジョブが DPL 要求を 1 つ以上実行した場合にのみ取得可能です。ゼロ以外のタスク番号は、DPL 要求が現在アクティブであることを示しています。ゼロのタスク番号は、現在アクティブな DPL 要求が存在しないにもかかわらず、外部 CICS インターフェース・セッションがまだそのジョブに対してオープンされている (接続されている) ことを示しています。

CEMT コマンドについて詳しくは、[CEMT - マスター端末](#)を参照してください。

## EXCI ユーザー置換可能モジュール

外部 CICS インターフェースには、ユーザー置換可能モジュール DFHXCURM が用意されています。

ロード・モジュールは CICSTS56.CICS.SDFHEXCI 内にあり、ソースは CICSTS56.CICS.SDFHSAMP 内にあります。ユーザー置換可能プログラムのアセンブルおよびリンク・エディットについては、[ユーザー置換可能プログラムのアセンブルとリンク・エディット](#)を参照してください。

DFHXCURM は、**allocate\_pipe** コマンドの処理中と、再試行可能エラーが発生した後に、非 CICS 領域で外部 CICS インターフェースによって開始されます。

再試行可能応答は以下のとおりです。

- ターゲット CICS 領域が使用不可である
- ターゲット CICS 領域に使用可能なパイプがない
- MVS IPL 以降に IRC アクティビティーがなかった

再試行可能エラー発生後に再試行するには、EXCI 呼び出しを再発行します。

サンプル・ロジックをバイパスし、外部 CICS インターフェース呼び出し元に制御を返すブランチ命令があるため、提供されたままの DFHXCURM は実際のところダミー・プログラムです。サンプル・ロジックを使用するには、ブランチ命令を削除し、モジュールをアセンブルしてリンク・エディットしてください。以下のアクションを実行するように DFHXCURM をカスタマイズすることができます。

- **allocate\_pipe** の処理中に、指定された CICS APPLID を変更して、要求を別の CICS システムに送ることができます。
- **allocate\_pipe** 処理中に、要求を別の XCF グループへ送ることができます。
- 再試行可能なエラーの後に DFHXCURM が開始される場合、CICS の可用性に関する情報を保管できます。その後、**allocate\_pipe** 処理のために DFHXCURM を次に呼び出す時に、どの CICS システムに要求を送付するかを決定するのにこの情報を使用できます。

DFHXCURM は、標準の MVS レジスター規則 (レジスター 1 にパラメーター・リストのアドレス、レジスター 14 に呼び出し元の戻りアドレスを格納する) を使用して呼び出されます。レジスター 1 でアドレス指定されたパラメーターは、DFHXCPLD コピーブック内にある EXCI\_URM\_PARMS DSECT でマップされます。DFHXCURM に渡されるパラメーターは、以下のとおりです。

### URMINV

DFHXCURM の呼び出しの理由が格納されるフルワードのアドレス。以下のように定義されます。

URM_ALLOCATE	EQU 1	This invocation is for an Allocate_Pipe
URM_NO_CICS	EQU 2	The target CICS region is not available

```
URM_NO_PIPE      EQU 3  There are no pipes available
URM_NO_CICS_IRC   EQU 4  There has been no IRC activity since the MVS IPL
```

## URMCICS

ターゲット CICS システムの APPLID を格納する 8 バイト領域のアドレス。この ID は、**Allocate\_Pipe** コマンドの **CICS\_applid** パラメーター、または **EXEC CICS LINK** コマンドの **APPLID** パラメーターで指定されます。

これらのコマンドの 1 つで指定された場合、APPLID は別のターゲット CICS 領域のものに変更できます。また、これらのコマンドの 1 つで指定された APPLID が有効な固有のアプリ ID でない場合は、その APPLID を有効な固有のアプリ ID に変更する必要があります。

allocate\_pipe 要求で **CICS\_applid** パラメーターが省略された場合、または **EXEC CICS LINK** コマンドで APPLID が省略された場合、このパラメーターでアドレス指定されるフィールドには 8 個のリンクが格納されます。この場合は、呼び出し元に制御を返す前に、DFHXCURM に APPLID を指定する必要があります。

## URMAPPL

Initialize\_User コマンドの *my\_name* パラメーターで指定されたクライアント・プログラムのユーザー名を格納する 8 バイト領域のアドレス。EXEC CICS LINK コマンドで DFHXCURM が開始される場合、この名前は必ず DFHXCEIP に設定されます。

## URMPROG

ターゲット・プログラムが使用可能な場合に、その名前を格納する 8 バイト領域のアドレス。この名前が使用可能なのは、DFHXCURM が EXEC CICS LINK コマンドで開始された場合のみです。外部 CICS インターフェースの allocate\_pipe コマンドの場合は、DPL 呼び出しが発行されるまで、プログラム名は分かりません。

## URMOPTS

*allocate\_opts* パラメーターに指定されている割り振りオプション (X'00' または X'80') が格納される 1 バイト領域のアドレス。このアドレスが有効なのは、Allocate\_Pipe 要求の場合のみです。

## URMANCH

DFHXCURM 専用として提供される 4 バイト領域のアドレス。これは一般的に、複数回の呼び出しに渡って DFHXCURM の情報を保管するために使用される領域のグローバル・アンカー・アドレスを保管する目的で使用されます。例えば、必要なストレージを GETMAIN で取得し、このパラメーターでアドレス指定される 4 バイト領域にそのアドレスを保管できます。4 バイト領域の初期値はゼロに設定されます。

EXCI を使用するアドレス・スペースでは、TCB ごとに URMANCH パラメーターが 1 つずつあります。

## URMXCFG

DFHXCOPT テーブルの **XCFGROUP** パラメーターに指定された XCF グループ名を格納する 8 バイト領域のアドレス。EXCI Allocate\_Pipe コマンドの処理中に DFHXCURM が呼び出されたときには、このパラメーターを使用して XCF グループ名を変更します。DFHXCURM の呼び出しに失敗したが、再試行できる場合は、パイプを以前割り振ったときに使用された値がこの領域に格納されます。値の変更による影響はありません。

グループ名の長さは 8 文字でなければならず、必要に応じて右側に空白が埋め込まれます。有効な文字は A-Z 0-9 \$ # @ です。文字 A、B、C、E、F、G、H、I、"SYS" でグループ名を始めないください。これらの名前は、IBM がその XCF グループに使用しています。また、「UNDESIG」という名前も使用しないでください。この名前は、インストール済み環境でのシステム・プログラマーによる使用のために予約されています。

文字 "DFHIR" で始まるグループ名を使用することをお勧めします。

# EXCI オプション・テーブル DFHXCOPT の使用

DFHXCOPT マクロによって生成される EXCI オプション・テーブルを使用すると、外部 CICS インターフェースに必要な複数のパラメーターを指定できます。

## DFHXCOPT オプション・テーブル: 新しい形式と古い形式

DFHXCOPT オプション・テーブルは、最初に導入されて以降変更が加えられ、将来の拡張に備えて、より柔軟性を持たせるため、テーブルにバージョン番号が含まれるようになりました。例えば、CICS の以前のリリースから、カスタマイズ済みの DFHXCOPT テーブルをマイグレーションすることを計画している場合、この変更に注意する必要があります。

新旧の形式を区別するために、新しい形式のテーブルは、DFHXCOPE という別名によってリンク・エディットされています。以下の手順を使用してオプション・テーブルがロードされます。

1. CICS は、DFHXCOPE という別名を使用して、DFHXCOPT テーブルをロードしようとします。CICS は、DFHXCOPE という名前のロード・モジュールを検出し、そのロードに成功した場合、そのテーブルを新しい形式のものとして想定します。
2. CICS が DFHXCOPE という名前のロード・モジュールを検出しない(または、検出されたものの、ロードに失敗した) 場合、DFHXCOPT というベース名を使用してテーブルのロードを試行します。この場合、CICS はテーブルが古い形式であると想定します。

CICS にはデフォルトの DFHXCOPT テーブルが用意されていて、そのデフォルト・テーブルのソース・コードも CICSTS56.CICS.SDFHSAMP ライブラリーにあります。デフォルトの DFHXCOPT テーブルのロード・モジュールと、その別名 DFHXCOPE は、CICSTS56.CICS.SDFHEXCI ライブラリーに用意されています。

## カスタマイズされた DFHXCOPT テーブルの作成

CICS 提供のデフォルトの DFHXCOPT テーブルのソース・コードを、ユーザーの要件に合わせて調整できます。

カスタマイズした DFHXCOPT テーブルをアセンブルし、MVS クライアント・プログラムを実行するジョブの STEPLIB 連結内の適切なライブラリーにリンク・エディットする必要があります。

**重要: 独自にカスタマイズした DFHXCOPT テーブルを作成する場合は、必ず DFHXCOPE 別名を使用してリンク・エディットするようにしてください。** 標準 DFHAUPLE プロシージャを使用して、そのようになっていることを確認します。別名を使用せずにテーブルを再アセンブルし、リンク・エディットすると、CICS は、カスタマイズされたテーブルではなく、デフォルトのテーブル (DFHXCOPE 別名を使用して検出) をロードします。

このために、独自のバージョンの CICS DFHAUPLE プロシージャを使用することができます。DFHAUPLE プロシージャは、CICSTS56.CICS.SDFHINST に用意されています。

## DFHXCOPT マクロ: 形式およびパラメーター

CICS 領域に指定するテーブルとは違って、DFHXCOPT テーブルには接尾部を付けることはできません。

次の表に、DFHXCOPT マクロとそのパラメーターの形式を示します。

DFHXCO	TYPE={ <b>CSECT</b>  DSECT} [,ABENDBKOUT={ <b>NO</b>  YES}] [,CICSSVC={ <b>216</b>  number}] [,CONFDATA={ <b>HIDE</b>  SHOW}] [,DURETRY={ <b>30</b>  number-of-seconds}] [,GTF={ <b>OFF</b>  ON}] [,LOCALCCSID={ <b>037</b>  CCSID}] [,MSGCASE={ <b>MIXED</b>  UPPER}]
--------	---

END	<p>除去</p> <p>[,SURROGCHK={<b>YES</b> NO}]</p> <p>[,TIMEOUT={<b>0</b> number}]</p> <p>[,TRACE={<b>OFF</b> 1 2 3}]</p> <p>[,TRACESIZE={<b>16</b> number-of-kilobytes}]</p> <p>[,TRAP={<b>OFF</b> ON}]</p> <p>[,XCFGROUP={<b>DFHIR000</b> name}]</p> <p>次の END ステートメントを使用してパラメーターを終了する必要があります。</p> <p>DFHXCOPT</p>
-----	---

#### TYPE={CSECT|DSECT}

生成されるテーブルのタイプを示します。

##### **CSECT**

通常使用される正規の制御セクション。

##### **DSECT**

ダミー制御セクション。

#### ABENDBKOUT={NO|YES}

タスクが CICS サーバー内で異常終了した場合にグローバル作業単位の自動ロールバックを起動するかどうかを指定します。グローバル作業単位は、EXCI クライアント・プログラムが MVS RRS を通じてリソース・リカバリーを制御している場合 (つまり、DPL 要求上で SYNCONRETURN が指定されていない場合) に存在します。この場合、CICS サーバー・プログラムが異常終了する場合にロールバックするよう、グローバル作業単位にマークを付けると良いかもしれません。

注：DPL 要求上で SYNCONRETURN が指定された場合、ABENDBKOUT は無効です。

##### **NO**

グローバル作業単位はロールバックされるようにマークを付けられません。

##### **YES**

サーバー・プログラムの異常終了を処理しているとき、CICS ミラー・プログラムは、バックアウトされるようにグローバル作業単位にマークを付けます。

どちらの場合でも、EXCI クライアント・プログラムは、EXCI DPL 要求上で、戻りコード 422、SERVER\_ABENDED を受け取ります。

#### CICSSVC={216|number}

MRO 通信用に使用されている CICS タイプ 3 SVC の番号を指定します。デフォルトは 216 です。

外部 CICS インターフェースは、クライアント・プログラムが稼働中の MVS イメージ内にある CICS MRO 領域で使用されているのと同じ SVC 番号を使用する必要があります。

##### **0**

外部 CICS インターフェースが、by means of an MVS VERIFY コマンドを使用して MVS から CICS SVC 番号を取得することを示すには、0 を指定します。

0 を指定するのは、少なくとも 1 つの CICS 領域が、MVS IPL の存続期間中に DFHIRP にログオンしたことが確かな場合だけにしてください。

##### **number**

CICS 領域間通信用に使用中の CICS SVC 番号を 200 から 255 の範囲で指定します。これは、クライアント・プログラムが稼働中の MVS イメージ (ローカル MVS) にインストールされた SVC 番号である必要があります。



IPL の存続時間中にローカル MVS の DFHIRP にログオンした MRO CICS 領域がない場合、ゼロ以外の SVC 番号を指定する必要があります。ゼロを指定して、外部 CICS インターフェース 要求が MVS から SVC を要求した場合、DFHIRP にログオンした CICS 領域がなければ、この要求は失敗します。

CICS 領域をまったく実行しておらず、クライアント・プログラムが、別の MVS に常駐するサーバー CICS 領域に対して DPL 要求を発行する MVS イメージでは、ゼロ以外の値が必要です。このような環境では、クライアント・プログラムは、ローカルに定義された SVC を使用してローカル DFHIRP にログオンし、XCF/MRO を使用してリモート CICS 領域と通信します。

**注：**同一の MVS イメージ内で MRO を使用するすべての CICS 領域は、DFHIRP と CICS SVC、DFHCSVC の両方とも最高レベルのものを使用する必要があります。MRO CICSplex がさまざまなリリース・レベルの CICS 領域で構成されている場合、LPA にインストールされた DFHIRP と DFHCSVC は、CICSplex 内の最新のリリース・レベルの CICS からのものである必要があります。

### **CONFDATA={HIDE|SHOW}**

このパラメーターは、外部 CICS インターフェースが、GTF への EXCI トレース項目出力や EXCI ダンプに通常なら表示されるユーザー・データを抑止する (非表示にする) かどうかを指示するためにコーディングします。このオプションは、EXCI クライアント・プログラムと CICS サーバー・プログラムとの間でフローする COMMAREA データまたは CONTAINER データのトレースに適用されます。

#### **HIDE**

EXCI はトレース項目からユーザーの COMMAREA データまたは CONTAINER データを「非表示」にします。代わりに、トレース項目には、データが抑止されたことを示す文字ストリングが設定されます。

#### **SHOW**

データ抑止は有効ではありません。ユーザー・データはトレースされます。

### **DURETRY={30|number-of-seconds|0}**

外部 CICS インターフェースが、SDUMP マクロを使用して MVS システム・ダンプを取得しようと試行を続ける合計時間を秒単位で指定します。

DURETRY は、外部 CICS インターフェースが SDUMP 要求を出す際に、同じ MVS システムの別のアドレス・スペースが既に SDUMP を取っている場合に、外部 CICS インターフェースが SDUMP を再発行するかどうか、および再発行する場合はその期間の長さを制御できます。

SDUMP が失敗した場合は、外部 CICS インターフェースは、次のように対応します。

- MVS が既に別のアドレス・スペースの SDUMP を取っており、DURETRY パラメーターがゼロでない場合、外部 CICS インターフェースは、SDUMP マクロを再試行する前に MVS STIMERM マクロを発行して、5 秒間待機します。外部 CICS インターフェースは、DURETRY 制限時間まで 5 秒ごとに SDUMP を再試行することを示すメッセージを発行します。
- SDUMP が次のような理由で失敗した場合、外部 CICS インターフェースは、SDUMP が失敗したことを通知するメッセージを理由とともに出力します。
  - 使用可能な SYS1.DUMP データ・セットがない。
  - ダンプの完了を妨げる入出力エラーがある。
  - SDUMP の再試行中に DURETRY の制限時間が満了した。

#### **30**

30 秒を指定すると、外部 CICS インターフェースは、最大 6 回まで (5 秒ごとに 1 回) 再試行できます。

#### **number-of-seconds**

外部 CICS インターフェースに SDUMP マクロの再試行を継続させる秒数の合計 (最大 32767 秒) をコーディングします。外部 CICS インターフェースは、成功するか、DURETRY 値以上の期間が経過するまで、5 秒ごとに 1 回 SDUMP を再試行します。

#### **0**

CICS に SDUMP を再試行させたくない場合には、ゼロの値をコーディングします。

**GTF={OFF|ON}**

外部 CICS インターフェース・トレース・テーブルに通常書き込まれるすべてのトレース項目が、(GTF トレースがアクティブの場合に) MVS 汎用トレース機能 (GTF) データ・セットにも書き込まれるかどうかを指定します。

**OFF**

トレース項目を GTF に書き込まない場合にコーディングします。

**ON**

トレース項目を GTF に書き込む場合にコーディングします。

**LOCALCCSID={037|CCSID}**

EXCI ジョブのデフォルト CCSID を指定します。CCSID は 8 文字までの値です。CCSID 値を指定しない場合は、デフォルトの LOCALCCSID が 037 に設定されます。有効な CCSID のリストについて詳しくは、以下の情報を参照してください。

- [CICS がサポートされている変換](#)
- [z/OS Unicode Services ユーザーズ・ガイドおよび解説書の関連付録](#)

**037**

LOCALCCSID のデフォルト値です。

**CCSID**

その他の有効な EBCDIC CCSID 値を表します。

**MSGCASE={MIXED|UPPER}**

DFHEXxxxx メッセージを大/小文字混合または大文字のどちらで 発行するかを指定します。

**MIXED**

メッセージを大/小文字混合で出力する場合にコーディングします。

**UPPER**

メッセージを大文字で出力する場合にコーディングします。

**SURROGCHK={YES|NO}**

**除去:** SURROGCHK パラメーターが除去されました。代理検査は常に行われます。SURROGCHK=NO というオプションを希望する場合は、IBM サポートに usermod を要求する必要があります。

クライアント・ジョブ・ユーザー ID (EXCI ジョブを実行中のユーザー ID) に対して、外部 CICS インターフェースが、代理ユーザー検査を実行するかどうかを指定します。

**YES**

外部 CICS インターフェースは、代理ユーザー検査を実行して、EXCI クライアント・ジョブを実行中のユーザー ID が、DPL 呼び出しで指定したユーザー ID に対して、代理として認可されているかどうかを検証します。この検査は、クライアント・ユーザー ID が、DPL 呼び出しで指定したユーザー ID と違う場合のみ実行されます。

クライアント・ユーザー ID が、SURROGAT 一般リソース・クラスの適切なプロファイルに対して認可されている必要があります。そのためには、クライアント・ユーザー ID に、SURROGAT 一般リソース・クラスの userid.DFHEXCI という名前のプロファイルに対する READ 権限を与えます。userid は、DPL 呼び出し上で指定されたユーザー ID です。

バッチ領域のユーザー ID を DPL ユーザー ID の代理ユーザーとして認可する方法の例については、[代理ユーザー検査](#)を参照してください。

**NO**

代理ユーザー検査は実行されません。

**TIMEOUT={0|number}**

外部 CICS インターフェースが、DPL コマンドの完了を待機する時間間隔を 100 分の 1 秒単位で指定します。

DPL コマンドは、1 つのチャンネルとコンテナ・セットを CICS に渡すことができます。この情報を渡すコマンドには、コンテナ・データを構成するために、CICS までの複数のデータ・フローを含めることができます。タイムアウト間隔は、データの最後のフローが CICS に送信された時に開始します。このときにデータは完了し、CICS サーバー・プログラムの呼び出しが開始します。

## **0**

時間制限を適用せず、外部 CICS インターフェースが、DPL コマンドの完了を無期限に待機することを指定します。

### **number**

外部 CICS インターフェースが、DPL コマンドの完了を待機する時間間隔を 100 分の 1 秒単位で指定します。number は、100 分の 1 秒単位の時間を 1 から最大 2 147 483 647 までの数で表現します。以下に例を示します。

#### **6000**

1 分のタイムアウト値を表します。

#### **30000**

5 分のタイムアウト値を表します。

#### **60000**

10 分のタイムアウト値を表します。

### **TRACE={OFF|1|2|3}**

外部 CICS インターフェースの内部トレースを行うかどうか、およびどのレベルにするかを指定します。

#### **OFF**

外部 CICS インターフェースの内部トレースは不要です。しかし、通常のトレースをオフに切り替えても、例外トレース項目は常に CICS 領域内の外部 CICS インターフェース・トレース・テーブルに書き込まれます。

#### **1**

例外、およびレベル 1 のトレース項目が外部 CICS インターフェース・トレース・テーブルに書き込まれます。

#### **2**

例外、レベル 1、およびレベル 2 のトレース項目が外部 CICS インターフェース・トレース・テーブルに書き込まれます。

#### **3**

例外、レベル 1、レベル 2、レベル 3 のトレース項目が外部 CICS インターフェース・トレース・テーブルに書き込まれます。

### **TRACESIZE={16|number-of-kilobytes}**

外部 CICS インターフェースによって使用されるトレース・テーブルのサイズをキロバイト単位で指定します。このトレース・テーブルは、CICS 領域内の 31 ビットのストレージ (16 MB 境界より上) に割り振られます。

#### **16**

トレース・テーブルのデフォルト・サイズであり、同時に最小サイズでもあります。

#### **number-of-kilobytes**

トレース・テーブル用に割り振られるストレージの、16 KB から 1 048 576 KB までの範囲のキロバイト数。トレース・テーブル・ストレージ用にサブプール 1 が使用されます。これは、ジョブ・ステップ TCB の期間だけ存在します。テーブルは、ページ単位で位置合わせされ、整数のページを占めます。指定された値がページ・サイズ (4 KB) の倍数でない場合は、4 KB の次の倍数に切り上げられます。

### **TRAP={OFF|ON}**

サービス・トラップ・モジュール DFHXCTRA を使用するかどうかを指定します。DFHXCTRA は、ユーザー置き換え可能モジュールとして提供されます。ここには、IBM サービス技術員が、エラーをトラップするコードを追加できます。

#### **OFF**

DFHXCTRA を使用しない場合にコーディングします。

#### **ON**

DFHXCTRA が必要な場合にコーディングします。

### **XCFGROUP={DFHIR000|name}**

システム間カップリング・ファシリティ (XCF) グループの名前をこのクライアント・プログラムによって結合するように指定します。

**注:** XCF グループを使用すれば、同じシスプレックス内の異なる MVS イメージの CICS 領域が、複数領域操作 (MRO) 接続を通じて相互に通信できます。XCF/MRO の概要と、XCF グループのセットアップ方法については、[システム間複数領域操作 \(XCF/MRO\)](#)を参照してください。

各クライアント・プログラムは、最大で 1 つの XCF グループを結合できます。

#### **DFHIR000**

デフォルトの XCF グループ名。

##### **name**

グループ名の長さは 8 文字でなければならず、必要に応じて右側にブランクが埋め込まれます。有効な文字は、A-Z、0-9、および国別文字、\$、# および @ です。IBM が自社の XCF グループ用に使用する名前の使用を回避するために、A から C、E から I の文字、または文字ストリング "SYS" で始まるグループ名を使用しないでください。また、「UNDESIG」という名前も使用しないでください。この名前は、インストール済み環境でシステム・プログラマーが使用するために予約されています。

「DFHIR」という文字で始まるグループ名を使用することをお勧めします。



## 第 13 章 CICS ONC RPC のセットアップ

CICS ONC RPC を使用すると、クライアント・アプリケーションは、ONC RPC 形式を使用して CICS プログラムをリモート・プロシージャーとして呼び出して利用することができます。CICS ONC RPC をセットアップするには、以下の情報に従ってください。

### クライアント

クライアントは、TCP/IP ネットワーク経由で CICS ONC RPC のサーバーにアクセスする必要があります。

クライアント・システムでは、TCP/IP for MVS (バージョン 2.2.1 および 3.1) でサポートされる ONC RPC バージョン 3.9 のライブラリーと互換性のあるライブラリーを使用する必要があります。TCP/IP ネットワーク経由で通信するために、適切なハードウェアおよびソフトウェアを用意する必要があります。

### MVS

次の項目が前提条件です。つまり、CICS ONC RPC を実行するには、これらが MVS システムにインストールされている必要があります。

- TCP/IP for MVS バージョン 2.2.1 以降。TCP/IP for MVS ポートを、関係する CICS 領域で使用できるようにしておく必要があります。
- Language Environment。これに、CICS ONC RPC を実行するための前提条件である C ランタイム・ライブラリーが含まれています。
- RPCGEN を使用している場合、または独自の XDR ルーチンを作成している場合は、RPCGEN 出力および XDR ルーチンをコンパイルするために C コンパイラーが必要です。

### CICS

CICS に、Language Environment サポートをセットアップする必要があります。

注：TCP/IP for MVS CICS Sockets は、CICS ONC RPC の前提条件ではありません。

### TCP/IP for MVS

ある CICS 領域から、ある TCP/IP for MVS 領域への接続に、CICS ONC RPC と TCP/IP for MVS CICS Sockets バージョン 2.2.1 の両方を使用することはできません。CICS Sockets と CICS ONC RPC は、別々の CICS 領域で実行することをお勧めします。

TCP/IP for MVS バージョン 3.1 のユーザーには、この問題はありません。CICS Sockets および CICS ONC RPC の両方を同じ CICS 領域から実行することができます。

### TCP/IP for MVS 2.2.1

CICS ONC RPC を実行するための前提条件はありません。

注：ある CICS 領域から、ある TCP/IP for MVS 領域への接続に、CICS ONC RPC と TCP/IP for MVS CICS Sockets バージョン 2.2.1 の両方を使用することはできません。CICS ソケットおよび CICS ONC RPC を異なる CICS 領域で実行することをお勧めします。

### TCP/IP for MVS 3.1

以下の PTF が、CICS ONC RPC を実行するための前提条件です。

- `xdr_text_char` XDR ライブラリー関数の使用に関連する PTF、番号は UN79963。

注：ある CICS 領域から、ある TCP/IP for MVS 領域への接続に、CICS ONC RPC と TCP/IP for MVS CICS Sockets バージョン 2.2.1 の両方を使用することはできません。CICS ソケットおよび CICS ONC RPC を異なる CICS 領域で実行することをお勧めします。

### ストレージ要件



特に記載がない限り、CICS ONC RPC で使用されるストレージは、CICS サブプールから取得されます。  
CICS ONC RPC を有効にする場合のストレージ所要量は以下のとおりです。

- 40 KB の基本ストレージ
- 登録済みの 4 タプルごとに 100 バイト。

処理されるクライアント要求ごとに、以下のストレージが必要です。

- インバウンド XDR ルーチンが内部データ構造用に使用する MVS 制御のストレージ。
- インバウンド XDR ルーチンが **Decode** 関数のために作成するデータ構造用に使用するストレージ。
- CICS プログラム通信域用のストレージ。
- CICS プログラムの実行中に別名トランザクションが使用するストレージ。
- **Encode** 関数がアウトバウンド XDR ルーチンのためにデータ構造を作成するのに使用するストレージ。
- アウトバウンド XDR ルーチンが使用する MVS 制御のストレージ。

## CICS ONC RPC セットアップ・タスク

CICS ONC RPC データ・セット、ダンプ・フォーマット、およびマイグレーションに関する警告に関連するタスクがあります。

### CICS ONC RCP データ・セットの作成

CICS ONC RPC データ・セットを作成するための JCL が、DFHCOMDS ジョブに用意されています。

このデータ・セットは、次のような DEFINE CLUSTER コマンドによって VSAM キー順データ・セットとして定義されます。

```
DEFINE CLUSTER (           -  
  NAME( xxxxxxxx.CICSONC.RESOURCE ) -  
  CYL ( 2 1 )              -  
  KEYS( 19 0 )             -  
  INDEXED                  -  
  VOLUME ( vvvvvv )       -  
  RECORDSIZE( 150 150 )   -  
  FREESPACE( 5 5 )        -  
  SHAREOPTIONS( 1 )       -  
)
```

このデータ・セットを定義するジョブを、接続マネージャーを初めて開始する前に実行しておく必要があります。

### ダンプ・フォーマット用の JCL エントリー

CICS ONC RPC (および実行中のすべての機能) に対してダンプ・フォーマットをオンに切り替えるには、IPCS VERBEXIT 制御ステートメントを変更します。

```
IPCS VERBEXIT DFHPD730  
FT=2
```

VERBEXIT は、CICS ONC RPC 制御ブロックのフォーマット済みダンプを提供します。

## CICS バージョン間のマイグレーション

CICS ONC RPC は CICS Transaction Server ベースの一部です。IBM 提供の CICS ONC RPC 用のプログラムを、以前のリリースから CICS Transaction Server に移行することはできません。

## z/OS Communications Server データ・セットの変更

CICS Transaction Server 領域を TCPIP.PROFILE データ・セットで z/OS Communications Server に定義し、ONC RPC アプリケーション用の特定のポートを予約することができます。

これについては、[z/OS Communications Server IP 構成ガイド](#)で説明しています。

## CICS への CICS ONC RPC リソースの定義

CICS ONC RPC には、CICSONC RPC で使用する CICS リソースを定義する 2 つの RDO グループ、DFHRP および DFHRPF が用意されています。

## CICS ONC RPC トランザクション用のトランザクション定義

これらの CICS ONC RPC トランザクションは、ロックされたグループ DFHRP に定義されています。

### CRPA

別名

### CRPC

接続マネージャー

### CRPM

サーバー・コントローラー

これらの定義は変更できません。

## 追加の別名トランザクション用のトランザクション定義

さまざまな理由で、他の別名トランザクション名を使用しなければならない場合があります。

- 監査目的
- リソースおよびコマンドの検査
- 開始優先順位の割り振り
- データベースのプラン選択の割り振り
- 各種 CICS プログラムの各種ランナウェイ値の割り当て

これを行う場合は、別名トランザクションを CICS に定義して、CRPA から定義をコピーし、必要に応じて修正する必要があります。CRPA 定義は以下のとおりです。

DEFINE	TRANSACTION(CRPA)	GROUP(DFHRP)
	PROGRAM(DFHRPAS)	TWASIZE(0)
	PROFILE(DFHCICST)	STATUS(ENABLED)
	TASKDATALOC(BELOW)	TASKDATAKEY(USER)
	RUNAWAY(SYSTEM)	SHUTDOWN(ENABLED)
	PRIORITY(1)	TRANCLASS(DFHTCL00)
	DTIMOUT(NO)	INDOUBT(BACKOUT)
	SPURGE(YES)	TPURGE(NO)
	RESSEC(NO)	CMDSEC(NO)

CICS プログラムを CRPA 以外の名前の別名で実行する場合、[426 ページの『4 タブルの属性の定義』](#)で説明されているように、CICS プログラムと関連付けられた 4 タブルの属性を定義するときに、その別名を接続マネージャーに入力できます。別名の名前は、[別名と CICS プログラムの変更](#)で説明されているように、**Decode** 関数でも変更できます。

### **CMDSEC 値と RESSEC 値の変更**

クライアント要求を処理する CICS プログラムを含め、別名トランザクションで実行されるプログラムに対してセキュリティ検査を適用するために、CMDSEC(YES) または RESSEC(YES) を指定して新規の別名トランザクションを定義しなければならない場合があります。

別名で使用する IBM 提供のプログラムは、システム・プログラマー・インターフェース (SPI) コマンドを使用しないため、CMDSEC を変更する必要はありません。一方、CICS プログラム、リソース・チェッカー、またはコンバーターの **Encode** 関数による SPI コマンドの使用を監視するには、CMDSEC(YES) が必要です。

## **CICS ONC RPC プログラムのプログラム定義**

すべての CICS ONC RPC プログラムは、ロックされたグループ DFHRP に定義されています。

## **ユーザー作成プログラムのプログラム定義**

CICS プログラム、コンバーター、ユーザー作成 XDR ルーチン、およびリソース・チェッカーの定義を作成する必要があります。

### **LANGUAGE オプション**

ユーザー作成の XDR ルーチンは、LANGUAGE(C) で定義する必要があります。コンバーターおよび CICS プログラムは、適切な LANGUAGE で定義する必要があります。

### **CEDF オプション**

デバッグに EDF が必要な場合は、CICS プログラムのプログラム定義に CEDF(YES) を含める必要があります。

EDF を使用する場合、[426 ページ](#)の『[4 タブルの属性の定義](#)』で説明されているように、CICS プログラムと関連付けられた 4 タブルの属性を定義するときに、端末 ID を接続マネージャーに入力する必要があります。

### **EXECKEY オプション**

CICS プログラムは、CICS システムでそれらを CICS キーとして定義する特定の理由がない限り、EXECKEY(USER) として定義する必要があります。プログラムを EXECKEY(USER) として定義すると、プログラムによる CICS の上書きを防止できます。

ストレージを定義する場合は、コンバーターおよびリソース・チェッカーをアプリケーション・プログラムとして見なさないでください。それらを EXECKEY(CICS) として定義することをお勧めします。そうすると、それらは CICS キー・ストレージを変更できるようになります。

**Decode** および **Encode** 関数が変換後のデータを保持するためにストレージを割り振る場合、そのストレージは CICS キーとして割り振られる必要があります。

ユーザー作成 XDR ルーチンは、EXECKEY(CICS) として定義する必要があります。

CICS プログラムに EXECKEY(USER) を指定した場合は、その別名に TASKDATAKEY(USER) を指定してください。USER は、用意されているグループ DFHRP の別名定義に含まれているデフォルトの TASKDATAKEY 設定です。

EXECKEY(CICS) で指定する必要がある CICS プログラムが存在する場合は、それらを実行する別名に対して TASKDATAKEY(CICS) を指定することをお勧めします。

CICS は、システム初期設定パラメーター STGPROT が YES に設定されている場合、またはデフォルトで YES に設定された場合、ストレージ保護機能を使用して実行されます。

### **RELOAD オプション**

すべてのユーザー作成 XDR ルーチンに RELOAD(YES) を指定して、CICS ONC RPC の無効化処理のエラーを防止する必要があります。

## リモート CICS プログラムのプログラム定義

リモート・プロシージャ・コールを提供する CICS プログラムが、CICS ONC RPC によって別の CICS システムで実行される場合は、ローカル・システムとリモート・システムの両方にプログラム定義が必要です。

プログラムはリモート・システム上に存在するので、リモート・システムでのプログラム定義は単純です。ローカル・システムでのプログラム定義は次のようになります。

- プログラムが存在するシステムを指定する REMOTESYSTEM パラメーターが含まれている必要があります。
- ローカル・システムおよびリモート・システムで別の名前を使用する場合は、オプションで REMOTENAME パラメーターを含めることができます。
- オプションで TRANSID パラメーターを含めることができます。
  - TRANSID を指定しない場合、CICS プログラムは、リモート CICS システム上の CICS ミラー・トランザクションの下で実行されます。
  - TRANSID を指定した場合、リモート CICS システムのプログラムは、指定したトランザクション名で実行されます。別の名前を指定したほうが良い状況については、[415 ページの『追加の別名トランザクション用のトランザクション定義』](#)を参照してください。

リモート・トランザクション ID を指定する場合は、一致するトランザクション定義をリモート CICS システムに用意する必要があります。その定義で、リモート・システム用の適切なミラー・プログラムを指定する必要があります (CICS for MVS/ESA および CICS Transaction Server for z/OS システムの場合は DFHMIRS)。

CICS プログラムが CICS for MVS/ESA および CICS Transaction Server for z/OS を除く CICS プラットフォームで実行されている場合、同様の考慮事項が適用されますが、そのプラットフォームの DPL の詳細を参照する必要があります。

## マップ・セット定義

接続マネージャーのマップ・セット用のマップ・セット定義が、グループ DFHRP に用意されています。それらの定義は変更できません。

## 一時データ定義

CICS には、CICS ONC RPC のメッセージ一時データ・キュー CRPO のためのリソース定義が用意されています。このリソース定義は、DFHLIST の一部であるグループ DFHDCTG に含まれています。

グループ DFHDCTG はロックで保護されていないので、中に含まれている定義を必要に応じて変更することができます。CRPO は区画外キューとして定義されていますが、必要に応じて宛先を区画内または間接に設定できます。

CRPO の定義を区画外キューのままにする場合は、その区画外キューに対する適切な DD ステートメントを CICS JCL に追加する必要があります。例えば、以下のようになります。

```
//CRPO DD SYSOUT=A
```

## XLT 定義

XLT システム 初期設定パラメーターとその関連トランザクション・リストで、接続マネージャー CRPC が通常の CICS シャットダウン中に開始されることを許可する必要があります。CICS ONC RPC によってシャットダウンが遅延している場合には、接続マネージャーを使用して CICS ONC RPC の即時無効化を強制できます。



## 第 14 章 接続マネージャーを使用した CICS ONC RPC の構成

接続マネージャーには、4 つの主要な機能があります。

- CICS ONC RPC の有効化
- CICS ONC RPC の無効化
- CICS ONC RPC データ・セットに保管される動作オプションおよび 4 タプル情報の制御
- CICS ONC RPC データ・セットが有効な場合には、現在使用中の動作オプションおよび 4 タプル情報の制御

### 接続マネージャーの開始

接続マネージャーはさまざまな方法で開始できます。

- BMS マップをサポートする端末から。このセクションで説明する接続マネージャー・パネルで操作できます。
- CICS コンソールから。
- EXEC CICS START コマンドを使用する。
- 順次端末から。

接続マネージャーを開始すると何が行われるかは、以下に応じて異なります。

- CICS ONC RPC が有効か無効か
- BMS の使用を許可する端末から接続マネージャーを開始するかどうか
- トランザクション名を含む追加データを入力するかどうか
- CICS ONC RPC 定義レコードの Automatic Enable オプションが YES に設定されているかどうか

CICS ONC RPC が無効な場合に、BMS をサポートする端末でトランザクション名 (およびオプションの追加データ) を入力すると、次が行われます。

#### CRPC

- Automatic Enable が YES の場合は、自動有効化処理が行われます。
- Automatic Enable が NO の場合は、BMS パネル (DFHRP01) が表示されます。
- CICS ONC RPC 定義レコードがまだ存在しない場合は、BMS パネル (DFHRP01) が表示されます。

#### CRPC E A(N)

- BMS パネル (DFHRP01) が表示されます。

#### CRPC E A(Y)

- 自動有効化処理が行われます。CICS ONC RPC 定義レコードが存在しない場合は、オプションのデフォルト値を使用してレコードが 1 つ作成されますが、4 タプルは登録されません。

パネルが表示されない方法 (EXEC CICS START や非 BMS 端末など) で接続マネージャーを開始した場合、パネルを表示する操作を行うと、エラー・メッセージ DFHRP1505 が生成されます。

CICS ONC RPC が有効な場合に、トランザクション名 (およびオプションの追加データ) を入力すると、次が行われます。

- CRPC ではパネル DFHRP04 が表示されます。パネルを表示できない場合はエラー・メッセージ DFHRP1505 が生成されます。
- CRPC D(N) では通常の無効化処理が行われます。
- CRPC D(I) では即時の無効化処理が行われます。

CRPC E A(N)、CRPC E A(Y)、CRPC D(N)、および CRPC D(I) の形式は高速パス・コマンドと呼ばれます。



接続マネージャーで CICS ONC RPC を有効にする前に、z/OS Communications Server を開始しておく必要があります。そうしないと、4 タプルを登録できないので、z/OS Communications Server を開始してから CICS ONC RPC をもう一度有効化する必要があります。

```
CRPC                                CICS ONC RPC for MVS/ESA                                DFHRP01

Select one of the following. Then press Enter.

- 1. Enable CICS ONC RPC
  2. View or modify the CICS ONC RPC data set

Current Status: Disabled

PF1=Help  PF3=Exit  PF9=Messages  PF12=Return  SYSID= CI41  APPLID= IYK1ZF11
```

図 64. パネル DFHRP01

```
CRPC                                CICS ONC RPC for MVS/ESA                                DFHRP04

Select one of the following. Then press Enter.

- 1. Disable CICS ONC RPC
  2. View or modify the CICS ONC RPC data set
  3. View or modify CICS ONC RPC status

Current Status: Enabled

PF1=Help  PF3=Exit  PF9=Messages  SYSID= CI41  APPLID= IYK1ZF11
```

図 65. パネル DFHRP04

## 接続マネージャー BMS パネルの使用

BMS 入力では、前後のブランクはすべて無視されます。

すべてのパネルは上部の右隅にパネル ID (DFHRP02 など) が表示され、左隅に CRPC が表示されます。

すべてのパネルは下部の下から 4 番目の行に CICS ONC RPC の状況が表示され、下から 3 番目の行にプロンプト行が表示されます。一番下の行には使用可能な PF キーがリストされます。これには以下が含まれます。

### PF1

ヘルプ情報 (すべてのパネル)

**PF2**

CICS ONC RPC データ・セットから定義を削除します (表示されている場合のみ)

**PF3**

CRPC を終了します (PF3 を再度使用すると、確認のプロンプトが出されます)

**PF4**

フィールドを CICS ONC RPC データ・セットに書き込みます (表示されている場合のみ)

**PF7**

スクロールアップ (表示されている場合のみ)

**PF8**

スクロールダウン (表示されている場合のみ)

**PF9**

現在の入力に関連したメッセージを表示する

**PF12**

このパネルをキャンセルし、前のパネルに戻る

**接続マネージャーのエラー・メッセージの出力**

接続マネージャー・メッセージの宛先は、メッセージの性質によって異なります。

- ・オペレーターによる介入が必要な重大エラーはコンソールに送られます。その他のメッセージはコンソールに送られません。
- ・パネルの無効な入力に関連したメッセージは、PF9 を押して表示できます。
- ・内部エラーを報告するメッセージは CRPO に送信され、ほとんどの場合、PF9 を押すことで端末でも表示できます。

**PF9 を使用したメッセージの表示**

接続マネージャーの操作時に、エラー・メッセージが発行されることがあります。

これらはすぐには画面に表示されませんが、表示されるのを待機しているメッセージがあることを示すプロンプトがプロンプト行に表示されます。メッセージを表示するには、PF9 を押します。メッセージの番号とテキストが表示されます。

メッセージを読み終わったら、Enter、PF3、または PF12 を押して入力パネルに戻ることができます。

**CICS ONC RPC が無効な場合の接続マネージャーの開始**

CICS ONC RPC が無効な場合は、パネル DFHRP01 が表示されます。

(420 ページの図 64 を参照)。

オプションを選択して、Enter を押します。

**オプション**

詳細については、以下を参照してください。

**1**

[423 ページの『CICS ONC RPC の有効化』](#)

**2**

[434 ページの『CICS ONC RPC データ・セットの更新』](#)

**CICS ONC RPC が有効な場合の接続マネージャーの開始**

CICS ONC RPC が有効な場合は、パネル DFHRP04 が表示されます。

(420 ページの図 65 を参照)。

オプションを選択して、Enter を押します。

**オプション**

詳細については、以下を参照してください。

**1**

[433 ページの『CICS ONC RPC の無効化』](#)

2

[434 ページの『CICS ONC RPC データ・セットの更新』](#)

3

[422 ページの『CICS ONC RPC ステータスの更新』](#)

## CICS ONC RPC ステータスの更新

パネル DFHRP04 でオプション 3 を選択すると、パネル DFHRP10 が表示されます。

```
CRPC                                CICS ONC RPC for MVS/ESA Update Status          DFHRP10

Select one of the following. Then press Enter.

- 1. Change CICS ONC RPC settings
  2. Register procedure(s)
  3. Unregister procedure(s)
  4. View or modify alias list

Current Status: Enabled

PF1=Help   PF3=Exit   PF9=Messages   PF12=Return   SYSID= CI41   APPLID= IYK1ZF11
```

図 66. パネル DFHRP10

オプションを選択して、Enter を押します。

### オプション

詳細については、以下を参照してください。

1

[422 ページの『CICS ONC RPC ステータスの変更』](#)

2

[425 ページの『4 タブルの定義、保管、変更、および削除』](#)

3

[431 ページの『4 タブルの登録抹消』](#)

4

[438 ページの『別名リストの処理』](#)

## CICS ONC RPC ステータスの変更

パネル DFHRP10 でオプション 1 を選択すると、パネル DFHRP16 が表示されます。

CRPM ユーザー ID 以外の項目は、上書きして、CICS ONC RPC で現在使用されている値を変更できます。CRPM ユーザー ID は情報を提供するためだけに表示されています。CRPM ユーザー ID を変更するには、まず CICS ONC RPC を無効にする必要があります。

CRPC

CICS ONC RPC for MVS/ESA Status

DFHRP16

Trace( STARTED )  
Resource Checker( NO )

Trace Level( 1 )  
CRPM Userid( CICSUSER )

Current Status: Enabled

PF1=Help PF3=Exit PF9=Messages PF12=Return      SYSID= CI41 APPLID= IYK1ZF11

図 67. パネル DFHRP16

## CICS ONC RPC の有効化

CICS ONC RPC を有効化する方法には、オペレーターの介入によって有効化する方法と自動で有効化する方法の 2 つがあります。

CICS ONC RPC が無効な場合は、接続マネージャーを使用して以下を行うことができます。

- CICS ONC RPC 定義レコードをデータ・セット内に作成する、またはそれを更新する
- データ・セット内の 4 タプル・レコードを追加、削除、および変更する
- CICS ONC RPC を有効にする

接続マネージャーを使用して、次の 2 つの方法で CICS ONC RPC を有効化できます。

- オペレーター介入による有効化 - CICS ONC RPC を有効にする前に、以下を実行できます。
  - 一部またはすべてのオプションを変更する
  - 登録する 4 タプルを選択する
  - 登録前に 4 タプルの属性を変更する

CICS ONC RPC を有効にすると、動作を制御するオプションが有効になり、4 タプルを登録することができます。

オペレーター介入による有効化で行った変更は一時的なものであり、次に CICS ONC RPC を無効にしたときに失われます。それらを CICS ONC RPC データ・セットに保管し、次に CICS ONC RPC を有効にしたときにそれらを使用することもできます。

- 自動有効化 - CICS ONC RPC 定義レコードの内容によって、次に無効にされるまでの CICS ONC RPC の動作を制御するオプションが決まります。4 タプル定義の属性に応じて、一部の 4 タプルが登録される場合があります。

CICS ONC RPC データ・セットは、動作環境の情報を保管するものです。これには CICS ONC RPC 定義レコードと 4 タプル・レコードという 2 種類のレコードが含まれます。CICS ONC RPC 定義レコードには動作オプションが含まれ、4 タプル・レコードには 4 タプル情報が含まれます。

## オプションの設定および変更

CICS ONC RPC が無効になっている場合に接続マネージャーを開始し、パネル DFHRP01 でオプション 1 を選択すると、パネル DFHRP02 が表示されます。

CRPC	CICS ONC RPC for MVS/ESA Enable		DFHRP02
Overtyping to Modify			
		Choice	Possible Options
Trace	==>	STARTED	STArtd   STOpped
Trace Level	==>	1	1   2
Resource Checker	==>	NO	Yes   No
CRPM Userid	==>	CICSUSER	
Automatic Enable	==>	NO	Yes   No
Current Status: Disabled			
		SYSID= CI41 APPLID= IYK1ZF11	
PF1=Help	PF3=Exit	PF4=Save	PF9=Messages PF12=Return

図 68. パネル DFHRP02

「Choice」列に表示されている値は、CICS ONC RPC データ・セットに格納されているものです。このデータ・セットは、CRPM ユーザー ID として表示されている値が、CICS ONC RPC を実行する CICS システムのデフォルトの CICS ユーザー ID である場合を除き、424 ページの図 68 に示す値で初期化されています。

以下のフィールドに入力できます。小文字でも大文字でも入力可能です。フィールドへの入力が (例えば、YES か NO などに) 制限されている場合は、完全なオプション (YES) または最小のオプション (Y) を入力できます。パネルの「Possible Options」列に最小の入力内容が大文字で表示されています。このマニュアルの参照資料では、最小の入力内容を完全な入力内容の後に括弧で囲んで表記しています。

#### トレース

CICS ONC RPC トレースがアクティブかどうかを指定します。STARTED (STA) はアクティブであることを意味し、STOPPED (STO) はそうでないことを意味します。デフォルト値は STARTED です。

このオプションの設定が何であっても、CICS ONC RPC 例外トレース・エントリーは常に CICS 内部トレースに書き込まれます。例外以外のトレース・エントリーが書き込まれるようにするには、CICS トレースが開始され、このオプションが STARTED に設定されている必要があります。

#### トレース・レベル

CICS ONC RPC のトレース・レベルを指定します。値 1 はレベル 1 のトレース・ポイントをトレースすることを意味し、値 2 はレベル 1 と 2 の両方をトレースすることを意味します。デフォルトは 1 です。

#### リソース・チェッカー

YES (Y) は、CICS ONC RPC がすべての着信 RPC 要求の受信時にユーザー作成のリソース・チェック・モジュールを呼び出すことを意味します。NO (N) は、リソース・チェッカーを呼び出さないことを意味します。デフォルトは NO です。

#### CRPM ユーザー ID

サーバー・コントローラーを実行する CICS ユーザー ID を指定します。デフォルトは CICS ONC RPC が作動している CICS システムのデフォルト・ユーザー ID です。

#### Automatic Enable

YES (Y) または NO (N) を入力します。CICS ONC RPC データ・セットに YES が保管されている場合は、CRPC と入力するだけで CICS ONC RPC を有効化できます。すべての値がデフォルトで CICS ONC RPC データ・セットから取得されるので、追加のユーザー入力なしで CICS ONC RPC が有効になります。また、Register from Data Set オプションが YES に設定されているすべての 4 タプルが登録されます。デフォルト値は NO です。

このフィールドの設定は、CICS ONC RPC を使用可能にした場合にのみ有効になります。PF4 を使用して値を CICS ONC RPC データ・セットに保管した場合、その値が適用されるのは、次に有効化されたときです。ただし、オーバーライドした場合は別です。CICS ONC RPC データ・セットのこのフィールドの YES は、高速パス・コマンド CRPC E A(N) を使用してオーバーライドできます。

# オプションの検証、保管、およびアクティブ化

パネル DFHRP02 で変更を行ったら、Enter キーを押してそれらを接続マネージャーで検証する必要があります。

CICS ONC RPC データ・セットに新しい値を保管する場合は、PF4 を押します。

Enter を 2 度押すと、425 ページの『4 タプルの定義、保管、変更、および削除』で説明しているように、CICS ONC RPC が有効になり、パネル DFHRP03 が表示されます。

## CICS ONC RPC が有効な場合

CICS ONC RPC が有効な場合は、接続マネージャーを使用して以下を行うことができます。

- データ・セット内の CICS ONC RPC 定義レコードを更新する
- データ・セット内の 4 タプル・レコードを追加、削除、および変更する
- CICS ONC RPC の動作制御に使用するオプションを変更する
- データ・セットから 4 タプル定義を登録する
- 一時的な 4 タプル定義を作成し、登録する
- 4 タプル定義を登録抹消する
- CICS ONC RPC を無効化する

CICS ONC RPC を無効化するには、通常は無効化と即時無効化という 2 つの方法があります。無効化処理の影響については、433 ページの『CICS ONC RPC の無効化』で説明しています。

# 4 タプルの定義、保管、変更、および削除

4 タプルを定義、保管、変更、および削除するための最初のパネルは DFHRP03 です。

4 タプルを定義、保管、変更、および削除するための最初のパネルは DFHRP03 です (425 ページの図 69 を参照)。このパネルは、CICS ONC RPC を有効にするとすぐに表示されます。また、パネル DFHRP10 でオプション 2 を選択した場合にも表示されます。

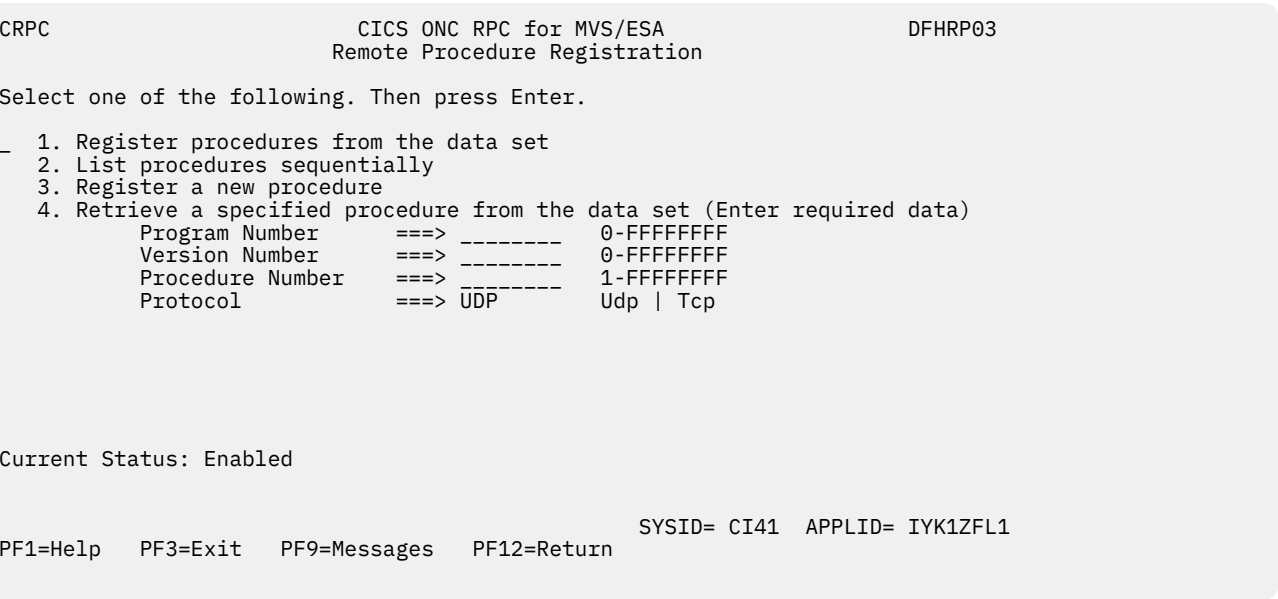


図 69. パネル DFHRP03

## オプション

詳細については、以下を参照してください。



**1** このセクションで後述する情報を参照してください。

**2** [426 ページの『4 タブルの属性の定義』](#)

**3** [431 ページの『4 タブルの登録抹消』](#)

**4** このセクションで後述する情報を参照してください。

**オプション 1** を選択した場合、Register from Data Set 属性が YES に設定されている CICS ONC RPC データ・セット内の 4 タブルがすべて登録されます。

CICS ONC RPC データ・セットに定義が存在しない 4 タブルを指定した場合は、Enter キーを押すとメッセージが発行されます。画面はパネル DFHRP03 のままです。

## 4 タブルの属性の定義

パネル DFHRP03 でオプション 3 またはオプション 4 を選択すると、パネル DFHRP5 が表示されます。オプション 3 を選択すると、一部のフィールドが空になりますが、オプション 4 を選択すると、選択された 4 タブルの詳細が表示されます。パネル DFHRP5B でさらに情報を指定する必要があります。

```

CRPC      CICS ONC RPC for MVS/ESA Remote Procedure Registration      DFHRP5

Overtyping to Modify. Then press Enter to Validate

ONC RPC ATTRIBUTES
ONC RPC Program Number  ===> 0-FFFFFFFF
ONC RPC Version Number  ===> 0-FFFFFFFF
ONC RPC Procedure Number ===> 1-FFFFFFFF
Protocol                ===> UDP      Udp | Tcp
RPC Call Type           ===> BLOCKING Blocking | Nonblocking
Inbound XDR Routine     ===>
Outbound XDR Routine    ===>
CICS ATTRIBUTES
ALIAS Transaction ID    ===> CRPA
EDF Terminal ID        ===>
+ Program Name          ===>

Current Status: Enabled

PF1=Help  PF3=Exit  PF4=Save  PF8=Forward  SYSID= CI41  APPLID= IYK1ZFL1
PF9=Messages  PF12=Return

```

```

CRPC      CICS ONC RPC for MVS/ESA Remote Procedure Registration      DFHRP5B

Overtyping to Modify. Then press Enter to Validate

+ CICS ONC RPC ATTRIBUTES
Converter Program Name  ===>
Encode                  ===> NO      Yes | No
Decode                  ===> YES     Yes | No
Getlengths              ===> YES     Yes | No
  Server Input Length   ===> 0 - 32767 Bytes
  Server Output Length  ===> 0 - 32767 Bytes
  Server Data Format     ===> CONTIGUOUS Contiguous | Overlaid
Register from Data set  ===> YES     Yes | No

Current Status: Enabled

PF1=Help  PF3=Exit  PF4=Save  PF7=Back  SYSID= CI41  APPLID= IYK1ZFL1
PF9=Messages  PF12=Return

```

図 70. パネル DFHRP5 および DFHRP5B

パネル DFHRP5 に変更を加えたら、PF8 を押してパネル DFHRP5B に移動する必要があります。パネル DFHRP5B からパネル DFHRP5 に戻るには、PF7 を押します。これらのパネルに変更を加えたら、Enter を押して、すべての変更を検証します。

4 タブルの属性は、以下の 3 つのカテゴリーに分けられます。

- ONC RPC 属性
- CICS 属性
- CICS ONC RPC 属性

### ONC RPC 属性

最初の 4 つのオプションにより、属性を定義する 4 タブルを設定します。

#### ONC RPC Program Number

4 タブルのプログラム番号を 1 文字から 8 文字の 16 進数ストリングで指定します。0 から 1FFFFFFF までの範囲の数値は、公共のネットワーク・サービス用に予約されていて、Sun Microsystems によって割り振られているため、使用しないことをお勧めします。

### ONC RPC Version Number

4 タブルのバージョン番号を 1 文字から 8 文字の 16 進数ストリングで指定します。

### ONC RPC Procedure Number

4 タブルのプロシージャ番号を 1 文字から 8 文字の 16 進数ストリングで指定します。プロシージャ 0 は、パラメーターも処理も持たない、空の応答を返すプロシージャ用に z/OS Communications Server によって予約されています。

### Protocol

4 タブルのプロトコルを指定します。UDP の場合は UDP (U)、TCP の場合は TCP (T) です。

残りのオプションでは、4 タブルの属性を指定します。

### RPC Call Type

CICS ONC RPC でクライアントからの呼び出しを BLOCKING (B) と見なすか、それとも NONBLOCKING (N) と見なすかを指定します。NONBLOCKING を指定した場合、アウトバウンド XDR ルーチンを指定することはできず、クライアントに応答は送信されません。デフォルトは BLOCKING です。

### インバウンド XDR ルーチン

インバウンド XDR ルーチンの名前を指定します。XDR ライブラリー関数が使用されている場合、そのフルネームが指定されます。ここで指定できるライブラリー・ルーチンを検索するには、[ステップ 3: XDR ルーチンの書き込み](#)を参照してください。ユーザー定義ルーチンが使用されている場合、その名前 (最長 8 文字) が指定されます。

### アウトバウンド XDR ルーチン

RPC Call Type が BLOCKING の場合に、アウトバウンド XDR ルーチンの名前を指定します。XDR ライブラリー関数が使用されている場合、そのフルネームが指定されます。ここで指定できるライブラリー・ルーチンを検索するには、[ステップ 3: XDR ルーチンの書き込み](#)を参照してください。ユーザー定義ルーチンが使用されている場合、その名前 (最長 8 文字) が指定されます。空の入力は、RPC Call Type が NONBLOCKING である場合にのみ有効です。

### CICS 属性

別名のトランザクション ID、EDF 端末 ID、およびプログラム名は、CICS に指定する必要がある属性です。

### 別名のトランザクション ID (ALIAS Transaction ID)

別名に使用するトランザクション ID を指定します。これを省略した場合、**Decode** 関数でも指定されていなければ、別名トランザクション ID は CRPA になります。CRPA とは別の名前を指定したほうが良い状況については、[415 ページの『追加の別名トランザクション用のトランザクション定義』](#)を参照してください。

### EDF 端末 ID (EDF Terminal ID)

別名に使用する端末 ID を指定します。端末 ID が必要になるのは、実行診断機能 (EDF) を使用してリソース・チェッカー、CICS プログラム、またはコンバーターの **Encode** 関数をデバッグする場合のみです。空になっている場合、EDF を使用できないことを意味します。EDF のセットアップについては、[EDF の使用](#)で説明しています。

### Program Name (プログラム名)

この 4 タブルの要求を処理するために呼び出される CICS プログラムの名前を指定します。

### CICS ONC RPC 属性

#### Converter Program Name

コンバーター・プログラムの名前を指定します。この名前の指定は必須です。

#### Encode

YES (Y) は、この 4 タブルのクライアント要求を処理する場合に CICS ONC RPC がコンバーターの **Encode** 関数を呼び出す必要があることを意味します。NO (N) はその必要がないことを意味します。デフォルトは NO です。

#### Decode

YES (Y) は、この 4 タブルのクライアント要求を処理する場合に CICS ONC RPC がコンバーターの **Decode** 関数を呼び出す必要があることを意味します。NO (N) はその必要がないことを意味します。デフォルトは YES です。

## Getlengths

YES (Y) は、この 4 タプルを登録する前に接続マネージャーがコンバーターの **Getlengths** 関数を呼び出す必要があることを意味します。NO (N) は、その必要がないことを意味します。ここで、YES を指定した場合、次の 2 つの属性を無視する必要がありますが、Server Data Format は設定できます。ここで NO を指定した場合、次の 3 つの属性を指定する必要があります。デフォルトは YES です。

## Server Input Length

このオプションの使用法については、サーバー・データ形式の説明を参照してください。

Getlengths オプションに YES を指定した場合、このフィールドはブランクのままにします。

## Server Output Length

このオプションの使用法については、サーバー・データ形式の説明を参照してください。

Getlengths オプションに YES を指定した場合、このフィールドはブランクのままにします。

## Server Data Format

以下を制御する値。

- **Encode** の入力データ・ポインターをセットアップする方法
- 接続マネージャーによって検査される通信域の長さを計算する方法

指定可能な値は次のとおりです。

## CONTIGUOUS

**Encode** に渡されるデータ・ポインターの値、または、この 4 タプルで **Encode** を使用しない場合はアウトバウンド XDR ルーチンに渡されるデータ・ポインターの値は、CICS プログラム通信域のアドレスに Server Input Length の値を加算したものです。ただし、このオフセットは **Decode** で変更することができます。

接続マネージャーは、Server Input Length と Server Output Length の値を加算して、通信域の長さを計算します。この長さが 32,767 b バイトを超えると、メッセージ DFHRP1965 が発行されます。この長さが **Decode** から CICS プログラムに渡された通信域の実際の長さとは違う場合は、クライアント要求の処理中にエラーが発生する可能性があります。

## OVERLAID

**Encode** に渡されるデータ・ポインターの値、またはこの 4 タプルで **Encode** を使用していない場合は、アウトバウンド XDR ルーチンに渡されるデータ・ポインターの値は、CICS プログラム通信域のアドレスです。

接続マネージャーは、Server Input Length と Server Output Length の出力値のうち大きいほうを使用して、通信域の長さを計算します。この長さが CICS プログラムに渡される通信域の実際の長さとは違う場合は、クライアント要求の処理中にエラーが発生する可能性があります。

Getlengths オプションに YES を指定した場合、このフィールドの値はコンバーターの **Getlengths** 関数への入力として使用されます。

## Register from Data Set

YES (Y) は次の場合に 4 タプルを登録することを意味します。

- 自動有効化処理の実行時
- 430 ページの『[4 タプルの登録](#)』で説明しているように、パネル DFHRP03 でオプション 1 が選択された場合

NO (N) は、4 タプルを登録しないことを意味します。デフォルトは YES です。NO として指定したエントリーを、CICS ONC RPC データ・セットに保管し、CICS ONC RPC が有効にされた場合にいつでもそれらを登録することができます。

## 新規の 4 タプル定義を保管する

新規の 4 タプル定義を保管する方法は 5 つあります。

- パネル DFHRP03 でオプション 3 を選択します。パネル DFHRP5 および DFHRP5B に入力し、入力内容を検証します (426 ページの『[4 タプルの属性の定義](#)』を参照)。CICS ONC RPC データ・セットに定義を保管するには PF4 を押します。

- パネル DFHRP03 でオプション 4 を選択します。パネル DFHRP5 および DFHRP5B を変更し、入力内容を検証します (426 ページの『4 タブルの属性の定義』を参照)。CICS ONC RPC データ・セットに定義を保管するには PF4 を押します。
- パネル DFHRP20 でオプション 3 を選択します。パネル DFHRP21 および DFHRP2B に入力し、入力内容を検証します (437 ページの『4 タブルの属性の変更』を参照)。CICS ONC RPC データ・セットに定義を保管するには Enter を押します。
- パネル DFHRP20 でオプション 4 を選択します。パネル DFHRP21 および DFHRP2B を変更し、入力内容を検証します (437 ページの『4 タブルの属性の変更』を参照)。CICS ONC RPC データ・セットに定義を保管するには Enter を押します。
- パネル DFHRP03 でオプション 2 を選択します。パネル DFHRP14 で 4 タブルに対してコマンド **M** を入力します。パネル DFHRP21 および DFHRP2B を変更し、入力内容を検証します (437 ページの『4 タブルの属性の変更』を参照)。CICS ONC RPC データ・セットに定義を保管するには Enter を押します。

## 既存の 4 タブル定義の変更

CICS ONC RPC データ・セットに既に定義が存在する 4 タブルの属性の一部を変更するには、パネル DFHRP03 またはパネル DFHRP20 でオプション 4 を選択します。

## 既存の 4 タブル定義の削除

CICS ONC RPC データ・セットから既存の 4 タブル定義を削除するには、2 つの方法があります。

- パネル DFHRP03 でオプション 2 を選択します。次に、パネル DFHRP14 でリスト内の 4 タブルに対して **D** を入力します。Enter を押すと、選択したものがデータ・セットから削除されます。
- パネル DFHRP21 でキー PF2 を使用します (437 ページの『4 タブルの属性の変更』を参照)。

## 4 タブルの登録

以下のいずれかの方法で 4 タブルを登録できます。

- CICS ONC RPC データ・セットに含まれている、Register from Data Set に YES を指定して定義されたすべての 4 タブルを登録できます。これを行うには、パネル DFHRP03 でオプション 1 を選択し、Enter キーを押します。これらの 4 タブルの登録後もパネル DFHRP03 はそのまま表示されるため、他の選択を行うことができます。
- 4 タブル定義は一度に 1 つずつ登録できます。これを行うには、パネル DFHRP03 でオプション 3 またはオプション 4 を使用します。必要に応じて、426 ページの『4 タブルの属性の定義』で説明しているように、パネル DFHRP5 および DFHRP5B を変更し、それらを検証します。定義を登録するには、Enter キーを押します。
- リストから 4 タブルを登録できます。436 ページの『4 タブルのリストの処理』を参照してください。
- CICS ONC RPC が無効な場合は、自動有効化処理を開始することで、Register from Data Set 属性が YES に設定されている CICS ONC RPC データ・セット内のすべての 4 タブルを登録できます。

4 タブルを登録するときには、次の 2 つが行われます。

- プログラム-バージョン-プロトコルの 3 タブルが TCP/IP for MVS にまだ登録されていない場合は、登録されます。ポートマッパーがその組み合わせにポート番号を割り当てます。このポート番号が、この 4 タブルで表されるサービスをクライアントが要求するときに使用する番号になります。このプログラム、バージョン、およびプロトコルに対するプロシージャ 0 が呼び出し元で使用できるようになります。
- 4 タブルと関連付けられたリソースが、サービス・クライアント要求で使用できるようになります。クライアント要求が CICS ONC RPC に到着した場合、その要求の処理に使用されるリソースは、プログラム、バージョン、およびプロシージャ番号が要求のものと同じで、さらに、プロトコルがクライアントからサーバーまで要求を伝送するのに使用されたプロトコルと一致する 4 タブルのリソースになります。

## 登録に関する制限

CICS ONC RPC では、合計 252 のソケットを使用可能にします。プログラム/バージョン/プロトコルの各 3 タブルは、そのプログラム、バージョン、およびプロトコルの最初の 4 タブルが登録された時点から、1 個のソケットを使用します。このソケットは、そのプログラムとバージョンの最後の 4 タブルが登録抹消さ

れるまで使用中のままになります。各 TCP 呼び出しは、その呼び出しの期間中、1 個のソケットを使用します。

過剰な数の 4 タプルを登録すると、CICS ONC RPC が着信クライアント要求に提供できるサービスが削減されることになります。252 個を超えるプログラム/バージョン/プロトコルの 3 タプルを z/OS Communications Server に登録しようとすると、結果が予測不能になります。

## 4 タプルの登録抹消

CICS ONC RPC が既に有効になっている場合にのみ、前に CICS ONC RPC に登録された 4 タプルの登録を抹消できます。

パネル DFHRP10 でオプション 3 を選択すると、DFHRP11 が表示されます ([431 ページの図 71](#) を参照)。

```
CRPC                                CICS ONC RPC for MVS/ESA          DFHRP11
                                Remote Procedure Unregister

Select one of the following. Then press Enter.

- 1. Unregister procedures from a list
  2. Unregister a specified procedure (Enter required data)
      Program Number    ==> ----- 0-FFFFFFFF
      Version Number    ==> ----- 0-FFFFFFFF
      Procedure Number   ==> ----- 1-FFFFFFFF
      Protocol          ==> UDP----- Udp | Tcp

Current Status: Enabled

PF1=Help  PF3=Exit  PF9=Messages  PF12=Return  SYSID= CI41  APPLID= IYK1ZF11
```

図 71. パネル DFHRP11

オプションを選択して、Enter を押します。

### オプション

詳細については、以下を参照してください。

1

[432 ページの『リストから 4 タプルを登録抹消する』](#)

2

[431 ページの『4 タプルを 1 つずつ登録抹消する』](#)

## 4 タプルを 1 つずつ登録抹消する

パネル DFHRP11 でオプション 2 を選択する前に、プログラム番号、バージョン番号、プロシージャ番号、およびプロトコルを指定する必要があります。

### プログラム番号

登録抹消する 4 タプルのプログラム番号。

### バージョン番号

登録抹消する 4 タプルのバージョン番号。

### プロシージャ番号

登録抹消する 4 タプルのプロシージャ番号。

### Protocol

登録抹消する 4 タプルのプロトコル。



登録されている 4 タプルを指定した場合、Enter キーを押すと登録が抹消されます。画面はパネル DFHRP11 のままです。

登録されていない 4 タプルを指定した場合、Enter キーを押すとメッセージが発行されます。画面はパネル DFHRP11 のままです。

## リストから 4 タプルを登録抹消する

パネル DFHRP11 でオプション 1 を選択すると、パネル DFHRP12 が表示されます。

このパネルには、CICS ONC RPC に現在登録されている 4 タプルのリストが表示されます。リスト内の 4 タプルに対して **U** を入力した場合、Enter キーを押すとそれらが登録抹消されます。4 タプルの属性を表示するには、4 タプルに対して **?** を入力し、Enter キーを押します。パネル DFHRP13 が表示されます ([432 ページの図 73](#) を参照)。

```
CRPC                                CICS ONC RPC for MVS/ESA          DFHRP12
                                Registered Procedures List

Enter 'U' to Unregister, or '?' to display details of a procedure
-   Prog( 20000002 ) Vers( 00000001 ) Proc( 00000006 ) Prot( UDP )
-   Prog( 20000002 ) Vers( 00000001 ) Proc( 00000007 ) Prot( TCP )
-   Prog( 20000002 ) Vers( 00000001 ) Proc( 00000007 ) Prot( UDP )
-   Prog( 20000002 ) Vers( 00000001 ) Proc( 00000008 ) Prot( TCP )
-   Prog( 20000002 ) Vers( 00000001 ) Proc( 00000009 ) Prot( UDP )
-   Prog( 20000002 ) Vers( 00000001 ) Proc( 0000000A ) Prot( TCP )
-   Prog( 20000002 ) Vers( 00000001 ) Proc( 0000000B ) Prot( TCP )
-   Prog( 20000002 ) Vers( 00000001 ) Proc( 0000000B ) Prot( UDP )
-   Prog( 20000002 ) Vers( 00000001 ) Proc( 0000000C ) Prot( TCP )
-   Prog( 20000002 ) Vers( 00000001 ) Proc( 0000000C ) Prot( UDP )
-   Prog( ----- ) Vers( ----- ) Proc( ----- ) Prot( --- )
-   Prog( ----- ) Vers( ----- ) Proc( ----- ) Prot( --- )
-   Prog( ----- ) Vers( ----- ) Proc( ----- ) Prot( --- )
-   Prog( ----- ) Vers( ----- ) Proc( ----- ) Prot( --- )
-   Prog( ----- ) Vers( ----- ) Proc( ----- ) Prot( --- )
-   Prog( ----- ) Vers( ----- ) Proc( ----- ) Prot( --- )
Current Status: Enabled

                                SYSID= CI41  APPLID= IYK1ZFL1
PF1=Help PF2=Refresh PF3=Exit PF7=Back PF8=Forward PF9=Messages PF12=Return
```

図 72. パネル DFHRP12

```
CRPC                                CICS ONC RPC for MVS/ESA          DFHRP13
                                Display Registered Procedure

Program Number( 20000002 )          Version Number( 00000001 )
Procedure Number( 00000006 )        Protocol( UDP )
RPC Call Type( Blocking )           Inbound XDR( XDR_WRAPSTRING )
Outbound XDR( XDR_WRAPSTRING )      Alias Transid( CRPA )
Alias Termid( )                     Server Program Name( STRING6 )
Converter Program Name( RINGCVNY )   Getlengths( NO )
Decode( YES )                       Encode( NO )
Server Input Length( 00001 )         Server Output Length( 00001 )
Server Data Format( CONTIGUOUS )

Current Status: Enabled

                                SYSID= CI41  APPLID= IYK1ZFL1
PF1=Help PF3=Exit PF12=Return
```

図 73. パネル DFHRP13

## CICS ONC RPC の無効化

パネル DFHRP04 からオプション 1 を選択すると、パネル DFHRP06 が表示されます。

```
CRPC                      CICS ONC RPC for MVS/ESA Disable          DFHRP06
Select the type of disable required. Then press Enter.

Type of Disable   ==>  _____ Normal | Immediate

Current Status: Enabled

PF1=Help   PF3=Exit   PF9=Messages   PF12=Return   SYSID= CI41  APPLID=  IYK1ZFL1
```

図 74. パネル DFHRP06

このパネルには、入力するフィールドが 1 つしかありません。

### 無効化のタイプ

#### NORMAL (N)

通常の無効化処理が開始されます。

- プログラムとバージョンのすべてのペアが z/OS Communications Server から登録抹消されます。
- 既に CICS ONC RPC に入れている処理はすべて最後まで実行され、対応するクライアントに応答が送信されます。

#### IMMEDIATE (I)

即時の無効化処理が開始されます。

- まだ開始されていない別名が開始されることはありません。
- 別名で実行されている CICS プログラムは最後まで実行され、その後、別名は異常終了します。DPL を使用して呼び出された CICS プログラムが通常どおり終了した場合、リカバリー可能リソースに対して行われた変更はコミットされます。CICS プログラムがローカル・プログラムである場合、リカバリー可能リソースに対して行われた変更はバックアウトされます。ただし、その CICS プログラムが EXEC CICS SYNCPOINT を使用して同期点を取った場合は別です。
- プログラムとバージョンのすべてのペアが z/OS Communications Server から登録抹消されます。
- 応答がクライアントに送信されないため、クライアントは CICS プログラムが実行されたかどうかを判別できません。

Enter キーを押すと、入力内容が検証されます。Enter キーを再度押すと、無効化処理が開始されます。無効化処理の進行状況に応じて、Current Status が Disabling または Disabled に変わります。無効化処理の完了後に Enter キーを押すと、Current Status が Disabled に変わります。

パネルは PF3 または PF12 を使用するまで表示されます。

## CICS 通常シャットダウンの場合

CICS の通常シャットダウンは、CICS ONC RPC の通常無効化処理を開始します。

## CICS 即時シャットダウンの場合

CICS 即時シャットダウンでは、すべてのトランザクションが終了させられます。クライアントは、シャットダウンまたはその影響について通知されません。z/OS Communications Server に登録されているプログラム/バージョン/プロトコルの 3 タプルは、登録されたままになる可能性があります。

## CICS ONC RPC データ・セットの更新

パネル DFHRP01 でオプション 2 を選択するか、パネル DFHRP04 でオプション 2 を選択すると、パネル DFHRP20 が表示されます。

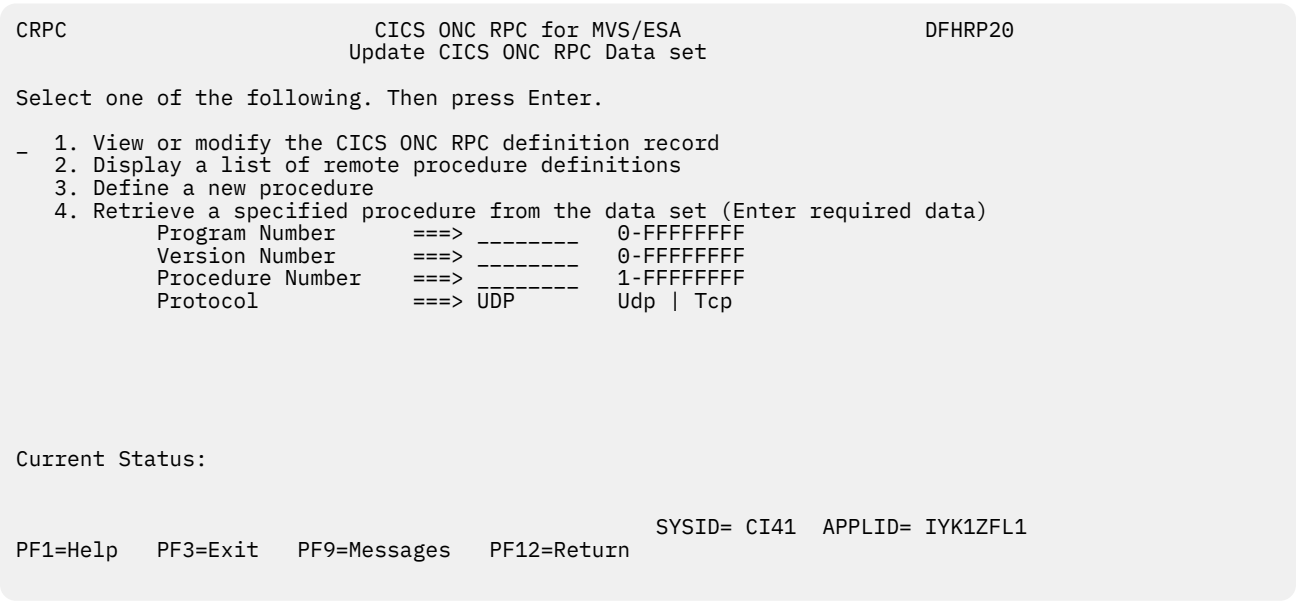


図 75. パネル DFHRP20

このパネルの Current Status フィールドには、元のパネルに応じて Enabled または Disabled が表示されます。

オプション 4 を選択する前に、以下の情報を指定する必要があります。

### プログラム番号

定義を取得する 4 タプルのプログラム番号。

### バージョン番号

定義を取得する 4 タプルのバージョン番号。

### プロシージャ番号

定義を取得する 4 タプルのプロシージャ番号。

### Protocol

定義を取得する 4 タプルのプロトコル。

オプションを選択して、Enter を押します。

### オプション

詳細については、以下を参照してください。

- 1 435 ページの『CICS ONC RPC 定義レコードの更新』
- 2 436 ページの『4 タプルのリストの処理』
- 3 437 ページの『4 タプルの属性の変更』

CICS ONC RPC データ・セットで定義されていない 4 タブルを指定した場合、Enter キーを押すとメッセージが発行されます。画面はパネル DFHRP20 のままです。

## CICS ONC RPC 定義レコードの更新

パネル DFHRP20 でオプション 1 を選択すると、パネル DFHRP22 が表示されます。

CRPC	Update	CICS ONC RPC for MVS/ESA CICS ONC RPC Definition Record	DFHRP22
Overttype to Modify		Choice	Possible Options
Trace	==>	STARTED	STArtd   STOppd
Trace Level	==>	1	1   2
Resource Checker	==>	NO	Yes   No
CRPM Userid	==>	CICSUSER	
Automatic Enable	==>	NO	Yes   No
Current Status:			
SYSID= CI41 APPLID= IYK1ZF11			
PF1=Help PF3=Exit PF9=Messages PF12=Return			

図 76. パネル DFHRP22

「Choice」列に表示されている値は、CICS ONC RPC データ・セットに格納されているものです。

変更を行った後、Enter キーを押してそれらを検証する必要があります。Enter キーを再度押すと、指定した値で CICS ONC RPC データ・セットを更新できます。接続マネージャーを次に開始するときには、保管されたオプションがパネル DFHRP02 のセットアップに使用されます。

### Trace

CICS ONC RPC トレースがアクティブかどうかを指定します。STARTED (STA) はアクティブであることを意味し、STOPPED (STO) はそうでないことを意味します。デフォルト値は STARTED です。

このオプションの設定が何であっても、CICS ONC RPC 例外トレース・エントリーは常に CICS 内部トレースに書き込まれます。例外以外のトレース・エントリーが書き込まれるようにするには、CICS トレースが開始され、このオプションが STARTED に設定されている必要があります。

### Trace Level

CICS ONC RPC のトレース・レベルを指定します。値 1 はレベル 1 のトレース・ポイントをトレースすることを意味し、値 2 はレベル 1 と 2 の両方をトレースすることを意味します。デフォルトは 1 です。

### Resource Checker

YES (Y) は、CICS ONC RPC がすべての着信 RPC 要求の受信時にユーザー作成のリソース・チェック・モジュールを呼び出すことを意味します。NO (N) は、リソース・チェッカーを呼び出さないことを意味します。デフォルトは NO です。

### CRPM Userid

サーバー・コントローラーを実行する CICS ユーザー ID を指定します。デフォルトは CICS ONC RPC が作動している CICS システムのデフォルト・ユーザー ID です。

### Automatic Enable

YES (Y) または NO (N) を入力します。CICS ONC RPC データ・セットに YES が保管されている場合は、CRPC と入力するだけで CICS ONC RPC を有効化できます。すべての値がデフォルトで CICS ONC RPC

このフィールドの設定は、CICS ONC RPC を使用可能にした場合にのみ有効になります。値を CICS ONC RPC データ・セットに保管した場合、その値が適用されるのは、次に有効化したときです。ただし、オーバーライドした場合は別です。CICS ONC RPC データ・セットのこのフィールドの値は、高速パス・コマンド CRPC E A(N) を使用してオーバーライドできます。

パネル DFHRP03 でオプション 2 を選択するか、パネル DFHRP20 でオプション 2 を選択すると、パネル DFHRP14 が表示されます。

図 77. パネル DFHRP14

**D** データ・セットから定義を削除します。

**M** パネル DFHRP21 を表示します。詳しくは、[437 ページの『4 タプルの属性の変更』](#)を参照してください。

## 436 CICS TS for z/OS: CICS の構成

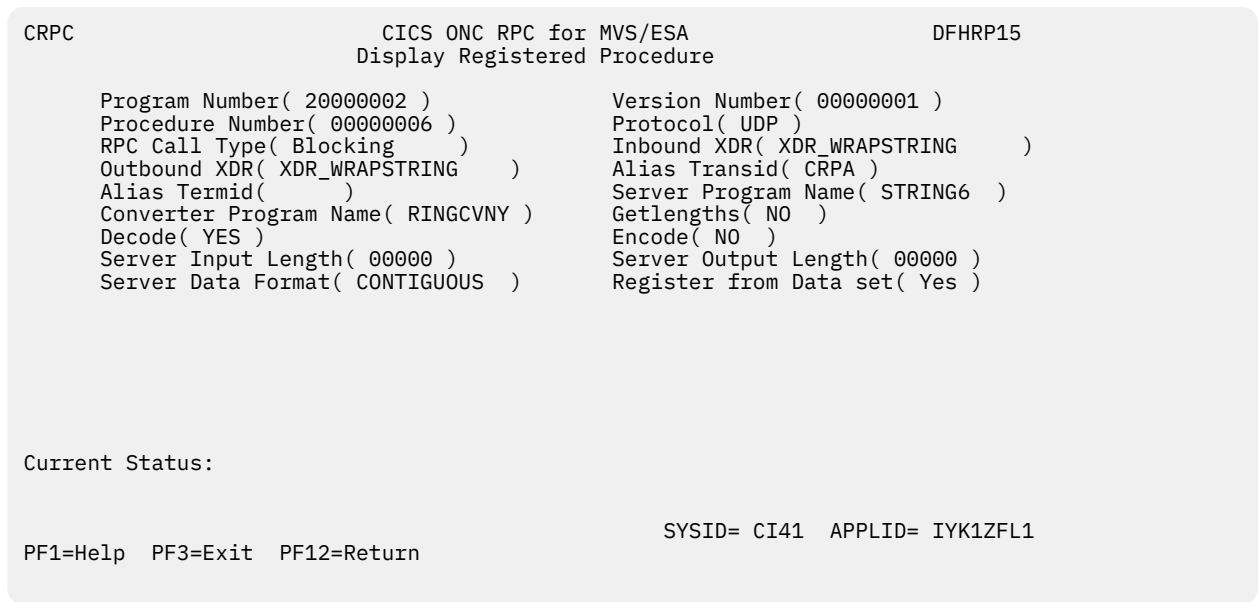


図 78. パネル DFHRP15

## 4 タブルの属性の変更

パネル DFHRP20 でオプション 3 または 4 を選択するか、パネル DFHRP14 で **M** コマンドを入力すると、パネル DFHRP21 が表示されます。

4 タブルの属性は、以下の 3 つのカテゴリーに分けられます。

- ONC RPC 属性 - [427 ページの『ONC RPC 属性』](#)を参照してください。
- CICS 属性 - [EDF 端末 ID](#) を参照してください。
- CICS ONC RPC 属性 - [428 ページの『CICS ONC RPC 属性』](#)を参照してください。



```

CRPC          CICS ONC RPC for MVS/ESA Remote Procedure Definition          DFHRP21

Overtyping to Modify. Then press Enter to Validate

ONC RPC ATTRIBUTES
ONC RPC Program Number  ===> 0-FFFFFFFF
ONC RPC Version Number  ===> 0-FFFFFFFF
ONC RPC Procedure Number ===> 1-FFFFFFFF
Protocol                ===> UDP      Udp | Tcp
RPC Call Type           ===> BLOCKING Blocking | Nonblocking
Inbound XDR Routine     ===>
Outbound XDR Routine    ===>
CICS ATTRIBUTES
ALIAS Transaction ID    ===> CRPA
EDF Terminal ID        ===>
+ Program Name          ===>

Current Status:

PF1=Help  PF2=Delete  PF3=Exit  PF8=Forward  SYSID= CI41  APPLID= IYK1ZFL1
PF9=Messages  PF12=Return

```

```

CRPC          CICS ONC RPC for MVS/ESA Remote Procedure Registration        DFHRP2B

Overtyping to Modify. Then press Enter to Validate

+ CICS ONC RPC ATTRIBUTES
Converter Program Name  ===>
Encode                 ===> NO      Yes | No
Decode                 ===> YES     Yes | No
Getlengths             ===> YES     Yes | No
  Server Input Length   ===> 0 - 32767 Bytes
  Server Output Length  ===> 0 - 32767 Bytes
  Server Data Format     ===> CONTIGUOUS Contiguous | Overlaid
Register from Data set ===> YES     Yes | No

Current Status:

PF1=Help  PF2=Delete  PF3=Exit  PF7=Back  PF9=Messages  PF12=Return  SYSID= CI41  APPLID= IYK1ZFL1

```

図 79. パネル DFHRP21 および DFHRP2B

これらのパネルで PF2 を押すと、4 タブルの定義を CICS ONC RPC データ・セットから削除することができます。

4 タブルの定義を変更する場合は、まずパネル DFHRP21 を変更してから、PF8 を押してパネル DFHRP2B に移動する必要があります。パネル DFHRP2B からパネル DFHRP21 に戻るには、PF7 を押します。これらのパネルに変更を加えたら、Enter キーを押してすべての変更を検証してから、再度 Enter キーを押して定義を変更する必要があります。

## 別名リストの処理

パネル DFHRP10 でオプション 4 を選択すると、パネル DFHRP17 が表示されます。

このパネルには、サーバー・コントローラーによって開始された別名、またはスケジュールされてまだ終了していない別名のリストが表示されます。このパネルでは、各別名が 2 行で表示されます。

- 最初の行は、クライアント要求の 4 タブルを示しています。
- 2 番目の行は、クライアント要求を処理している別名の CICS タスク番号を示しています。

スケジュールに入れられていてまだ開始されていない別名の場合、タスク番号はブランクです。別名が開始されると、タスク番号が付与され、その行が強調表示されます。

別名に対して以下のコマンドを入力できます。

**P**

別名をページします。

**?**

パネル DFHRP18 が表示され、別名および関連するクライアント 要求の詳細が表示されます (439 ページの図 81 を参照)。

スケジュールに入れられていてまだ開始されていない別名の場合、タスク番号および開始時刻はブランクです。別名が開始されると、タスク番号および開始時刻が表示されます。

```
CRPC                                CICS ONC RPC for MVS/ESA                DFHRP17
                                Alias List

Enter 'P' to Purge, or '?' to display details of an alias task
-   Prog( 00000103 ) Vers( 00000114 ) Proc( 00000001 ) Prot( UDP )
    Task Number( 00000033 )
-   Prog(          ) Vers(          ) Proc(          ) Prot(    )
    Task Number(          )
-   Prog(          ) Vers(          ) Proc(          ) Prot(    )
    Task Number(          )
-   Prog(          ) Vers(          ) Proc(          ) Prot(    )
    Task Number(          )
-   Prog(          ) Vers(          ) Proc(          ) Prot(    )
    Task Number(          )
-   Prog(          ) Vers(          ) Proc(          ) Prot(    )
    Task Number(          )
-   Prog(          ) Vers(          ) Proc(          ) Prot(    )
    Task Number(          )
-   Prog(          ) Vers(          ) Proc(          ) Prot(    )
    Task Number(          )
-   Prog(          ) Vers(          ) Proc(          ) Prot(    )
    Task Number(          )
Current Status: Enabled

                                SYSID= CI41  APPLID= IYK1ZFL1
PF1=Help PF2=Refresh PF3=Exit PF7=Back PF8=Forward PF9=Messages PF12=Return
```

図 80. パネル DFHRP17

```
CRPC                                CICS ONC RPC for MVS/ESA                DFHRP18
                                Display Alias Task Details

Program Number( 00000103 )          Version Number( 00000114 )
Procedure Number( 00000001 )        Protocol( UDP )
Task Number( 00000033 )             Client IP Addr( 9.20.2.19 )
CICS Program Name( RPROC103 )       Transid( CRPA )
Port Number( 000007BC )             Socket Descriptor( 00000003 )
Task Start Time( 14:38:19 )         Termid(          )

Current Status: Enabled

                                SYSID= CI41  APPLID= IYK1ZFL1
PF1=Help PF3=Exit PF12=Return
```

図 81. パネル DFHRP18



## 第 15 章 リカバリーと再始動のための構成

### ロギングおよびジャーナリング

すべての CICS システム・ロギングおよびジャーナリングは、CICS ログ・マネージャーによって制御されます。このマネージャーは、MVS システム・ロガー・ログ・ストリームを使用してその出力を保管します。

#### このタスクについて

CICS ロギングおよびジャーナリングは、大きく分けて 4 種類のアクティビティーに分けられます。

#### システム・ロギング

CICS では、リカバリー可能リソース用のトランザクション・バックアウトをサポートするためのシステム・ログを維持しています。CICS は、自動的にシステム・ロギングを実施しますが、この機能を禁止するためにログ・ストリームを DUMMY として定義することができます。ただし、システム・ログに TYPE(DUMMY) を指定すると、トランザクション・バックアウト機能も再始動機能もないまま実行することになり、CICS を START=INITIAL でしか始動できません。

CICS はシステム・ログに独自のデータを書き込むためのプログラミング・インターフェースもサポートしていますが、**ユーザー作成レコードはリカバリー目的でしか使用できません。**

442 ページの『[システム・ログ・ストリームの定義](#)』を参照してください。

#### 順方向リカバリー・ロギング

CICS は、VSAM データ・セットの順方向リカバリー・ログをサポートしています。

順方向リカバリー・ロギングは自動的にには行われません。この機能を自分のファイルに対して実行することを指定する必要がある、順方向リカバリー・ログ・ストリームも定義する必要があります。

454 ページの『[順方向リカバリー・ログ・ストリームの定義](#)』を参照してください。

#### 自動ジャーナリング

CICS は、ファイル制御データおよび端末管理メッセージの自動ジャーナリングをサポートしています。自動ジャーナリングは、一般に監査証跡に使用されます。

自動ジャーナリングは自動的にには行われません。この機能を自分のファイルおよび端末メッセージに対して実行することを指定する必要がある、使用する汎用ログ・ストリームも定義する必要があります。

FILE および PROFILE リソース定義での自動ジャーナリングの指定方法については、[FILE 属性](#)および [PROFILE 属性](#)で説明されている JOURNAL 属性を参照してください。

#### ユーザー・ジャーナリング

CICS は、CICS アプリケーションがユーザー定義のレコードをユーザー・ジャーナルに書き込むためのプログラミング・インターフェースをサポートしています。このジャーナルは、汎用ログ・ストリームに保持されます。

ユーザー・ジャーナルの汎用ログ・ストリームを定義する方法については、56 ページの『[CICS 汎用ログの定義](#)』を参照してください。

自動ジャーナリングおよびユーザー・ジャーナルは CICS リカバリーには影響しませんので、ここでは扱いません。

CICS ログ・マネージャーはこれらロギングおよびジャーナリングの各アクティビティーに関連するデータを 2 種類の MVS ログ・ストリームに書き込みます。

#### システム・ログ・ストリーム

これらは作業単位をリカバリーする目的に限って、CICS ログ・マネージャーおよび CICS リカバリー・マネージャーが使用します。それぞれのシステム・ログは CICS 領域に対して固有であり、他のシステム・ログとマージしてはなりません。

## 汎用ログ・ストリーム

これらは他のあらゆる種類のログギングおよびジャーナリングのために CICS ログ・マネージャーが使用します。順方向リカバリー・レコード、自動ジャーナル・レコード、およびユーザー・ジャーナル・レコードは、同一あるいは別の CICS 領域の同じ汎用ログ・ストリームにマージすることができます。

CICS ログ・マネージャーによって検出されたさまざまなエラー条件を CICS がどのように処理するかについては、[ログ・マネージャーの待機](#)を参照してください。

## MVS へのログ・ストリームの定義

ログ・ストリームは、カップリング・ファシリティ (使用されている場合) およびデータ・セット内に存在する MVS リソースです。

### このタスクについて

CICS が必要とするすべてのログ・ストリームは、CICS の使用前に MVS システム・ロガーに定義されている必要があります。ユーザーは、ログ・ストリームを明示的に定義するか、または最初を使用するときに CICS に動的に作成させるように定義することができます。CICS を使用可能にしてログ・ストリームを動的に作成するには、最初にモデル・ログ・ストリームを MVS システム・ロガーに定義します。明示的なログ・ストリームおよびモデル・ログ・ストリームを定義するには、MVS IXCMIAPU ユーティリティを使用します。

カップリング・ファシリティ・ログ・ストリームおよび DASD-only ログ・ストリームの定義について詳しくは、[カップリング・ファシリティ・ログ・ストリーム](#)を参照してください。カップリング・ファシリティおよび一般的なログ構造の定義について詳しくは、[z/OS MVS シスプレックスのセットアップ](#)を参照してください。

## レプリカ生成ログ・ストリームの定義

**LOGREPLICATE** と定義されている VSAM データ・セットのレプリカ生成ログを定義する必要があります。

### 手順

1. 統合カタログ機能 (ICF) で、レプリカ生成を含むデータ・セットのリカバリー属性を定義します。
2. レプリカ生成ログ・ストリームが定義されていない場合は、レプリカ生成データ用の汎用ログ・ストリームを定義する必要があります。汎用ログ・ストリームを定義しない場合、CICS ではログ・ストリームを動的に作成することを試みます。

## システム・ログ・ストリームの定義

作業単位の障害および CICS の障害が発生した場合にデータ保全性を維持するためには、システム・ログを定義する必要があります。

### このタスクについて

システム・ログは、以下のプロセスで必須です。

- 障害が発生した作業単位によって変更されたりリカバリー可能リソースのバックアウト
- コールド・スタート。CICS がシャットダウン時に中断された作業単位を復元できるようにします。
- ウォーム・リスタート。CICS が領域を (シャットダウン時に中断された作業単位も含めて) 直前のシャットダウン時の状態に復元できるようにします。
- 緊急リスタート。CICS が以下のことを実行できるようにします。
  - 領域を直前のシャットダウン時の状態に復元する。これには、シャットダウン時に中断された作業単位が含まれます。
  - CICS の障害発生時に未完了だった作業単位をリカバリーし、障害の前に更新されたりリカバリー可能リソースのバックアウトを実行する。

CICS ログ・マネージャーは、CICS JOURNALMODEL リソース定義で TYPE(DUMMY) として定義されていない限り、システムの初期設定時に、そのログ・ストリームに自動的に接続します。

CICS システム・ログは論理的には単一の論理ログ・ストリームですが、1 次および 2 次の 2 つの物理ログ・ストリームに書き込まれます。通常、これらを区別する必要はなく、ほとんどの参照の対象はシステム・ログ・ストリームとなります。単一のシステム・ログの 1 次および 2 次ログ・ストリームを使用することで、CICS は、長期実行 UOW を 1 次ログ・ストリームから 2 次ログ・ストリームに移動するプロセスを通じてログ・ストリーム・ストレージの使用を最適化できます。

### JOURNALMODEL リソース定義の指定

コールド・スタート、ウォーム・リスタート、および緊急リスタート時に、CICS はグローバル・カタログからシステム・ログ・ストリームの名前を取得して、前回の実行時に使用したものと同一ログ・ストリームに再接続していることを確認します。

初期始動時には、JOURNALMODEL リソース定義で特に指定されていない限り、CICS はデフォルトのログ・ストリーム名を使用します。システム・ログ・ストリームの選択および接続プロセスは、以下のとおりです。

### JOURNALMODEL 定義がない場合

CICS では、DFHLOG および DFHSHUNT の JOURNALMODEL リソース定義を検出できない場合、MVS システム・ロガーへの呼び出しを発行して、デフォルト名を使用してシステム・ログ・ストリームに接続します。デフォルト名は以下のとおりです。

```
region_userid.applid.DFHLOG  
region_userid.applid.DFHSHUNT
```

ここで、*region\_userid* は CICS アドレス・スペースが実行されている RACF ユーザー ID であり、*applid* は (APPLID システム初期設定パラメーターから取得される) 領域の z/OS Communications Server APPL 名です。デフォルトの DFHLOG および DFHSHUNT ログ・ストリームの CICS 提供 JOURNALMODEL 定義は CSD グループ DFHLGMOD に含まれ、このグループは DFHLIST に含まれています。

これらのデフォルト・ログ・ストリーム名を使用する場合は、以下のことを確認してください。

- ・ デフォルト・ログ・ストリームが、MVS システム・ロガーに対して明示的に定義されている。または
- ・ 適切なモデル・ログ・ストリームが動的作成用に定義されている。

### JOURNALMODEL 定義がある場合

CICS では、JOURNALNAME(DFHLOG) および JOURNALNAME(DFHSHUNT) を持つ JOURNALMODEL 定義を検出すると、MVS システム・ロガーへの呼び出しを発行して、定義で指定されているシステム・ログ・ストリームに接続します。システム・ログ・ストリームの名前が、CICS 領域に対して一意であることを確認してください。

システム・ログに対する JOURNALMODEL リソース定義を定義する場合は、以下のことを確認してください。

- ・ JOURNALMODEL で指定されたログ・ストリームが MVS システム・ロガーに対して定義されている。または
- ・ 適切なモデル・ログ・ストリームが、動的に作成できるように定義されている。

(例えば、CICS テスト領域で) システム・ログを使用しない場合は、TYPE(DUMMY) を指定して DFHLOG および DFHSHUNT の JOURNALMODEL リソース定義を指定します。TYPE(DUMMY) と定義されているシステム・ログを使用して CICS を実行すると、初期始動を実行するように強制され、CICS で動的トランザクション・バックアウトがサポートされないことに注意してください。

### CICS システム・ログに対するモデル・ログ・ストリーム

システム・ログ・ストリームが未定義のために CICS がそこへの接続に失敗した場合は、CICS はモデル・ログ・ストリームを使用してシステム・ログ・ストリームを動的に作成しようとします。

ログ・ストリームを動的に作成するためには、CICS が MVS システム・ロガーに、新しいログ・ストリームに必要なすべてのログ・ストリーム属性を指定する必要があります。さもないと不明になってしまうこれらの属性を判別するため、CICS は MVS システム・ロガーに対して、既存のモデル・ログ・ストリーム定義の属性を使用してログ・ストリームを作成することを要求します。ログ・ストリームの動的な作成を CICS に許可することにした場合は、必要なモデル・ログ・ストリーム定義を必ず作成して、動的な作成が正常に行われるようにしなければなりません。

一般に、モデル・ログ・ストリームをセットアップする価値があるのは、以下のような場合のみです。



- それぞれのモデル・ログ・ストリームが、複数のログ・ストリームの作成に使用される場合
- モデル・ログ・ストリームを使用して作成される各ログ・ストリームが類似した特性を持っており、同一のカップリング・ファシリティー構造を使用できる場合
- 所定の MVS イメージで実行しようとする CICS 領域がいくつかあるのか、事前にわからない場合 (たとえば開発環境やテスト環境の場合)

上記以外の場合は、IXCMIAPU を使用してログ・ストリームを明示的に定義するほうが、恐らく作業量は少なく済みます。一般に、モデル・ログ・ストリームを使用して動的に作成する方法はテストや開発目的に最も適しており、実動領域では明示的に定義する方法が適しています。

CICS がシステム・ログ・ストリームの動的な作成に使用するモデル・ログ・ストリームの名前は `&sysname.LSN_last_qualifier.MODEL` です。ここで、

- `&sysname` は MVS イメージのシステム名に解決される MVS シンボルです。
- `LSN_last_qualifier` は、JOURNALMODEL リソース定義で指定された、ログ・ストリーム名の最終修飾子です。

DFHLOG および DFHSHUNT に JOURNALMODEL リソース定義を指定しない場合、またはグループ DFHLGMOD で CICS により指定された定義を使用する場合、モデル名はデフォルトで `&sysname.DFHLOG.MODEL` および `&sysname.DFHSHUNT.MODEL` になります。

**例:** CICS 領域がその 1 次システム・ログに対するログ・ストリームの作成要求を発行し、CICS が MV10 というシステム ID の MVS イメージで実行されていて、デフォルトの JOURNALMODEL 定義を使用している場合、システム・ロガーは、MV10.DFHLOG.MODEL という名前のモデル・ログ・ストリームを見つけようとしています。

CICS はログ・ストリームを作成するため、MVS システム・ロガーを呼び出す直前に XLGSTRM グローバル・ユーザー出口を呼び出します。ログ・ストリームの作成に CICS デフォルト値を使用しない場合は、XLGSTRM グローバル・ユーザー出口プログラムを作成して要求の詳細を変更できます。そのような詳細にはモデル・ログ・ストリーム名も含まれます (パラメーター UEPMLSN に指定)。

### リカバリーに関する考慮事項

カップリング・ファシリティー・ログ・ストリームを使用している場合は、MVS イメージ間で構造を共用することにより、リカバリー上の利点が生じます。ある MVS イメージまたはロガー・アドレス・スペースに障害が起こったとき、残存する別の MVS イメージが同じログ・ストリーム構造 (同じログ・ストリームである必要はありません) を使用していると、残存するイメージには障害が通知されるため、障害の発生した MVS で使用されているログ・ストリームの即時ログ・ストリーム・リカバリーを開始することができます。そうでない場合、次にシステムが影響を受けた構造内のログ・ストリームに接続するまで、または障害のあるシステムのロガー・アドレス・スペースが再始動されるまで、リカバリーは延期されます。

ただし、CICS デフォルト名で定義されたモデル・ログ・ストリームは、常に、MVS イメージ内の同じ構造に割り当てられます。CICS デフォルト名を使用して定義されたモデル・ログ・ストリームを使用する場合、複数のカップリング・ファシリティーをまたいで定義された構造を使用していると、リカバリーの考慮事項の観点からは最適な割り振りとならないことがあります。

例えば、2 つのカップリング・ファシリティーを使用する両方向シスプレックスについて考えます。このカップリング・ファシリティーはそれぞれ、CICS システム・ログで使用するよう定義された 1 つのログ構造、LOG\_DFHLOG\_001 および LOG\_DFHLOG\_002 を持ちます。この場合、各 MVS の CICS 領域のリカバリーの観点からは、両方のログ構造間でのログ・ストリームの割り振りを均衡化すると、より適切です。このシナリオを 445 ページの図 82 で説明しています。ここでは、MVSA および MVSB で構成され、各 MVS に 4 つの CICS 領域を持つ両方向シスプレックスが示されています。CICSA1 から CICSA4 は MVSA 内に存在し、CICSB1 から CICSB4 は MVSB 内に存在します。各 MVS が 2 つのカップリング・ファシリティー CF1 および CF2 を使用しており、このそれぞれがモデル・ログ・ストリームにより参照される 1 つのログ構造を持ちます。各 MVS 内の最初と 2 番目の CICS 領域は最初のログ構造で定義されたログ・ストリームを使用し、各 MVS 内の 3 番目と 4 番目の CICS 領域は 2 番目のログ構造で定義されたログ・ストリームを使用します。このシナリオでは、各 MVS イメージは、MVS 障害が発生した場合にそのログ・ストリームをリカバリーするためのパートナーを持つことになります。

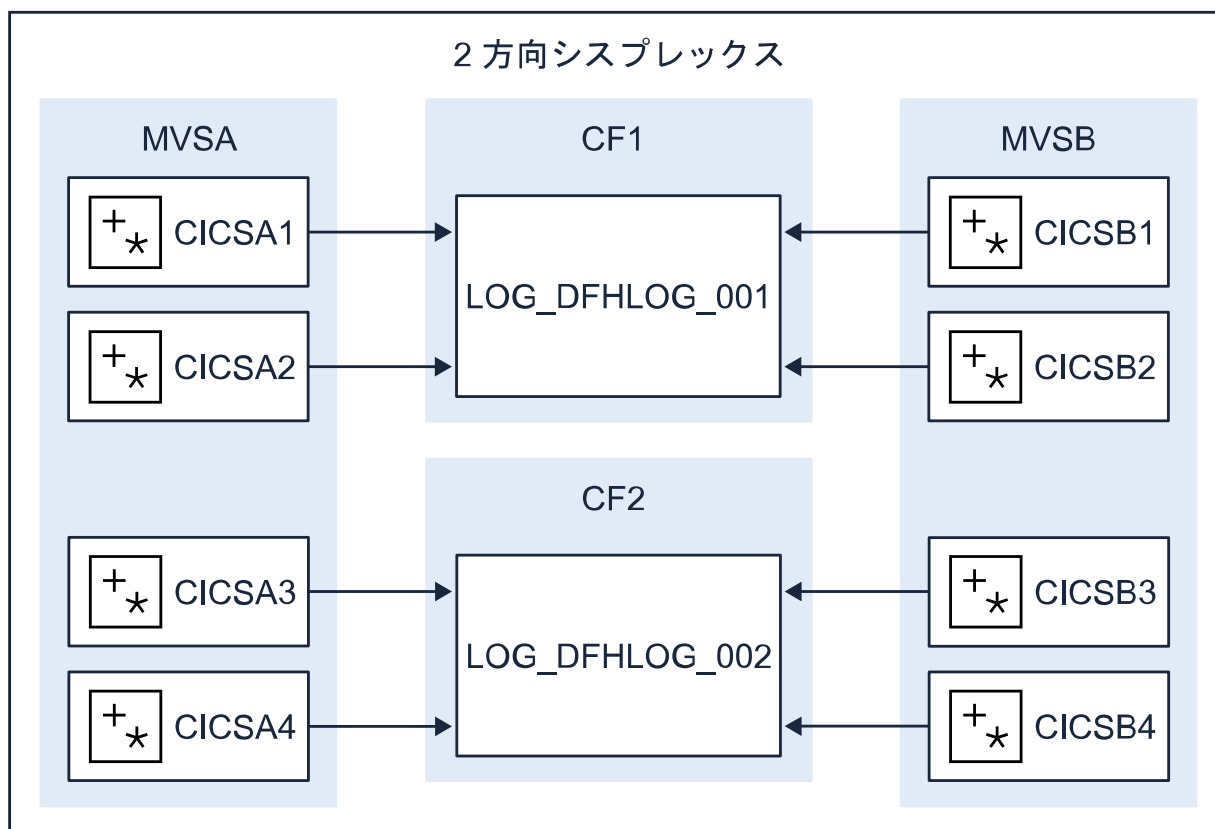


図 82. 2 つの MVS イメージ間でのシステム・ロガー構造の共用

447 ページの図 83 に示すもう 1 つの例は、MVSA、MVSb、MVSC、および MVSD で構成される 4 方向シスプレックスに基づきます。この場合は、MVSA と MVSb で通常実行される CICS 領域では構造 LOG\_DFHLOG\_001 が使用され、MVSC と MVSD で実行される領域では構造 LOG\_DFHLOG\_002 が使用されるようにします。こうすると、各 MVS イメージは、MVS 障害が発生した場合にそのログ・ストリームをリカバリーするためのパートナーを持つことになります。構造に障害が発生しても、他方の構造を使用する 2 つの MVS イメージがワークロードを引き継ぐことができます。

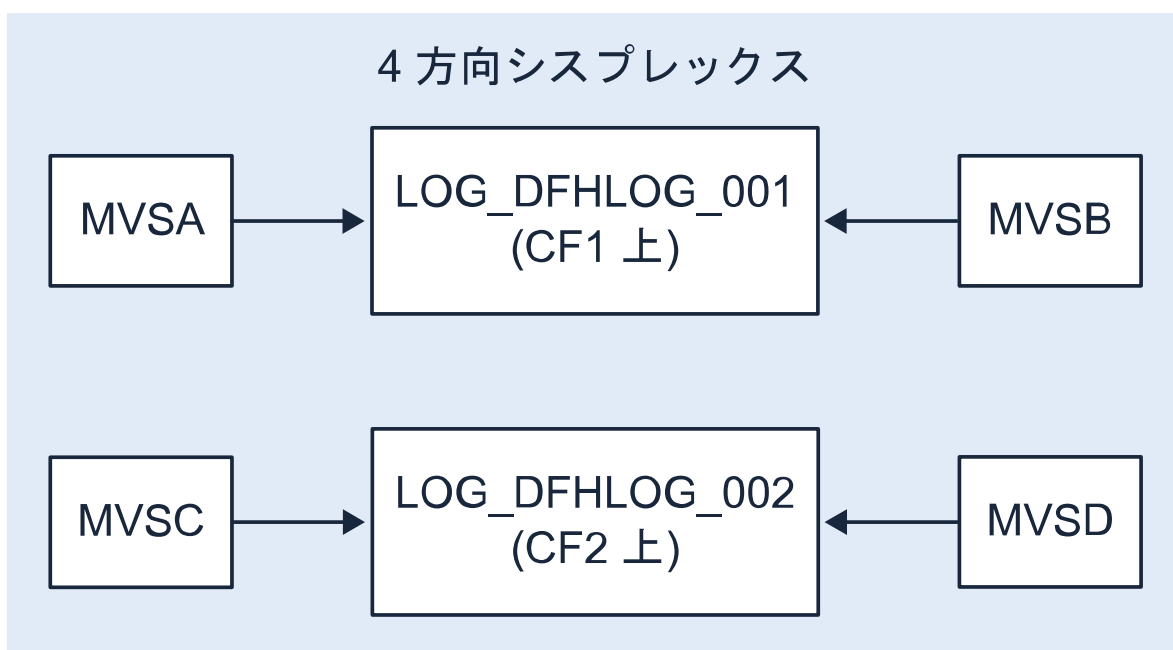


図 83. 4 つの MVS イメージ間でのシステム・ロガー構造の共用

## モデル・ログ・ストリーム名の多様化

複数のログ構造間でログ・ストリームを均衡化する場合、モデル・ログ・ストリームを使用することはモデル・ログ・ストリーム名をカスタマイズすることを意味します。CICS のデフォルト・モデル名を使用した場合は、このシナリオで示すログ・ストリームの分散を実現できません。

### このタスクについて

XLGSTRM グローバル・ユーザー出口プログラムを使用すると、いくつかの方法で、システム・ロガーに渡されるモデル・ログ・ストリームの名前を多様化できます。そのような方法の 1 つは、出口のグローバル作業域に適切な値を保管することです。例えば、**INITPARM** システム初期設定パラメーターを使用して、出口で使用するパラメーター・ストリングを指定できます。これは、XLGSTRM グローバル・ユーザー出口プログラムを使用可能にするための第 1 フェーズの PLT プログラムで、**EXEC CICS ASSIGN INITPARM** コマンドを使用して取得できます。INITPARM コマンドから関連するモデル・ログ・ストリーム情報を取得したら、後で XLGSTRM グローバル出口プログラムで使用するために、この情報をグローバル作業域に保管できます。このようにモデル・ログ・ストリームの詳細を多様化することで、カップリング・ファシリティー内の異なるログ構造をまたいでログ・ストリームを均衡化することができます。

XLGSTRM グローバル・ユーザー出口プログラムの作成方法、および PLT 初期設定プログラムについて詳しくは、[ログ・マネージャー・ドメイン出口 XLGSTRM](#) を参照してください。

### アクティビティ・キーポイント処理

CICS ログ・マネージャーは、再始動中にログを逆方向にスキャンして、作業単位情報を回復します。

ログの逆方向スキャンは、以下のように行われます。

1. 最初に、CICS は、終了前に書き込まれた最後の完全なアクティビティ・キーポイントまでさかのぼって、すべてのレコードを順次読み取ります。このスキャンにかかる時間を短縮するためには、アクティビティ・キーポイントの頻度をゼロに指定しないことが重要です。CICS 領域に見合ったアクティビティ・キーポイントの頻度の設定については、[活動キーポイント頻度 \(AKPFREQ\)](#) を参照してください。
2. CICS が最後に書き込まれた完全アクティビティ・キーポイントに到達すると、進行中の作業単位に関する情報が、未確定の作業単位も含めてすべて抽出されます。CICS はこの情報を使用して逆方向読み取りを続行しますが、今度はアクティビティ・キーポイントの中で識別された作業単位についてのレコードのみを読み取ります。この読み取り作業は、CICS がアクティビティ・キーポイントで識別された作業単位についてのすべてのレコードを読み取り終えるまで続行されます。

この処理は、中断してバックアウトに失敗した作業単位やコミットに失敗した作業単位も含めて、完了した作業単位がログ・スキャンのこの部分では無視されることを意味しています。このことは、ユーザー作成ログ・レコードの検索にとって重要な点です。ユーザー作成レコードが、CICS がログ・スキャン・プロセスのこの部分でスキップした作業単位の一部である場合、それらは XRCINPT グローバル・ユーザー出口プログラムでは提示されません ([451 ページの『ユーザー・リカバリー・データの書き込み』](#) を参照してください)。

[449 ページの図 84](#) に、CICS 再始動中の CICS によるログ処理の実行方法を示します。

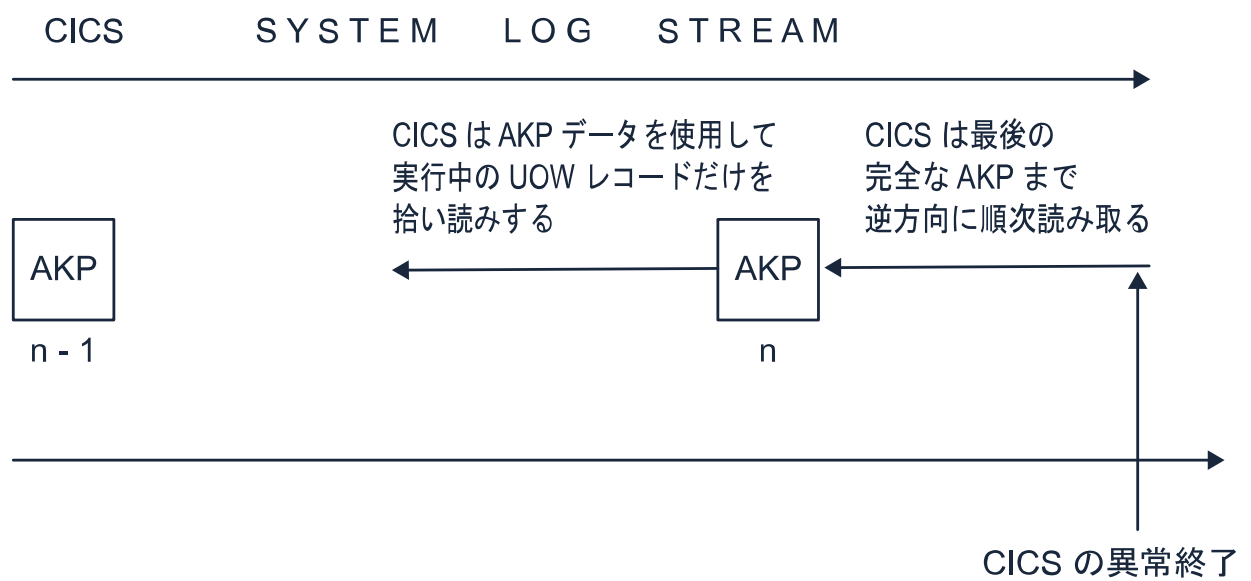


図 84. 再始動中のシステム・ログ・スキャン



以下に、システム・ログ・ストリームのサイズおよび再始動時間を最小に抑えるための手順を示します。

- 読み取りが必要なデータ量は最低にとどめます。このことは、アクティビティ・キーポイントの頻度として、以下の条件を満たすゼロ以外の値を指定することを意味します。

- 余分なキーポイント処理を避けるために十分な長さがあること
- キーポイント間のデータが大容量にならないような長さに抑えること

システム・ログ・ストリーム構造のサイズを計算する方法については、[システム・ログで使用する構造サイズ](#)を参照してください。

- 緊急再始動中に必要な自分独自のリカバリー・レコードを除き、システム・ログに自分独自のデータ (監査証跡データなど) を書き込まないでください。
- 特に、保持する必要のある情報はシステム・ログに書き込まないでください。(452 ページの『システム・ログでの保持期間の回避』を参照してください)。
- リカバリー可能リソースを更新する長時間実行のトランザクションを書き込む場合は、定期的な間隔で同期点をとるようにしてください。

### システム・ログ・データの最小サイズの保持

CICS ログ・マネージャーは、完了した最も古い作業単位レコードを定期的に削除することで (ログ・テールの削除)、システム・ログ・ストリームのサイズを制御します。

### このタスクについて

この操作は、活動キーポイントに関連付けられています。そのため、正しい活動キーポイント頻度 (AKPFREQ) を選択することが重要です。つまり、CICS がシステム・ログを適切なサイズに維持でき、かつ、カップリング・ファシリティーが使用されている場合には、カップリング・ファシリティー内にこのログを保持できる頻度です。

- システム・ログ・ストリームが割り振られている 1 次ストレージ・スペースを超えた場合は、補助ストレージに予備的に格納されます。(1 次ストレージおよび補助ストレージの定義については、[CICS ログ・マネージャー用の環境のセットアップ](#)を参照してください。)その結果得られる入出力は、システム・パフォーマンスに悪影響を与える可能性があります。
- 活動キーポイント間の間隔が長い場合、データの量が再始動時間に影響を与える可能性があります。一般的に、活動キーポイントの間隔は、ほとんどのトランザクション (長期実行トランザクションのカテゴリに含まれるものは除く) の経過時間よりも長い必要があります。

注 : AKPFREQ=0 は指定しないでください。これは、活動キーポイントがない場合、CICS では、シャットダウンするまでログ・テールの削除を実行できず、その時点までシステム・ログが補助ストレージに予備的に格納されるためです。

### ログ・テールの削除

ログ末尾とは、最古のログの末尾のことであり、各アクティビティ・キーポイントでログ・マネージャーによって削除されます。

各活動キーポイントで、CICS リカバリー・マネージャーはログ・マネージャーに、そのポイントより前にある古いデータ・ブロックをすべて削除できるポイントをシステム・ログに設定して、システム・ログのテールを削除するように要求します。したがって、最も古い「ライブ」作業単位がデータ・ブロック  $x$  内にある場合、CICS ログ・マネージャーはシステム・ロガーに、 $x$  より古い ( $x-1$  およびそれより古い) データ・ブロックをすべて削除するよう要求します。

システム・ログへの書き込みを定期的に開始する長期実行作業単位があると、CICS が多数の活動キーポイントにわたって完了した作業単位を削除できなくなる可能性があります。[452 ページの『長期実行トランザクション』](#)を参照してください。

## 2 次ログへの作業単位の移動

システムの活動キーポイント頻度の構成の効果を最適化するには、ログ・テールの削除プロセスによって、1 次ログ・ストリームを 1 次ストレージ・スペースの割り振り内に保持できるように、すべての作業単位の実行時間を十分に短くします。

### このタスクについて

ただし、ほとんどのシステムでは、一部の作業単位の実行時間が、より長くなります。これは、アプリケーションの設計が原因であることも、作業単位が未確定障害(コミット障害またはバックアウト障害)の影響を受けていることが原因であることもあります。このような少数の作業単位が領域全体に影響を与えないようにするために、CICS では、作業単位が長期実行になったことを検出して、2 次ログ・ストリームにそのログ・レコードのコピーを作成します。これにより、1 次ログ・ストリームをより多くトリムできるようになるため、システム・ログを 1 次スペースの割り振り内に保持できる可能性が高まります。CICS では、ある作業単位が 2 つの完了した活動キーポイント間隔にわたってシステム・ログへの書き込みを行わない場合、その作業単位が長期実行であると判断します。

作業単位のすべてのログ・レコードは、最初に 1 次ログ・ストリームに書き込まれます。その後、ログ・レコードはログ・テールの削除メカニズムによって削除されるか、2 次ログ・ストリームにコピーされます。コピーは、活動キーポイント処理中に行われます。

作業単位のログ・レコードは、2 次ログ・ストリームに移動した後、その作業単位が完了するまでそこに置かれます。その作業単位による後続のシステム・ログは、1 次ログに書き込まれることに注意してください。

## ユーザー・リカバリー・データの書き込み

### このタスクについて

システム・ログ・ストリームには、リカバリーに関するレコードのみを書き込む必要があります。これを行うには、アプリケーション・プログラミング・インターフェース (API) または出口プログラミング・インターフェース (XPI) で提供されるコマンドを使用します。ユーザー・リカバリー・レコードは XRCINPT 出口点で有効になっているグローバル・ユーザー出口プログラムに対して提示されるため、このようにすることが重要です。

ユーザー作成のリカバリー・レコードは、リカバリー・マネージャー・レコードと同じ方法で管理され、これによって CICS がログ・テール削除を実行できなくなることはありません。

- ユーザー・トランザクションによって書き込まれたユーザー作成リカバリー・レコードは、それらが書き込まれた作業単位の一部として処理されます。つまり、それらのレコードは UOW チェーンの一部を形成します。作業単位が何らかの理由で失敗した場合、CICS は作業単位の一部としてユーザー作成リカバリー・レコードを中断します。失敗の理由には、以下のようなものがあります。

- 緊急リスタート中に未完了の作業単位のリカバリーによってリカバリーされた
- ウォーム・リスタート中および緊急リスタート中の両方で、中断された作業単位のリカバリーによってリカバリーされた

- XAKUSER グローバル・ユーザー出口プログラムによって書き込まれたユーザー・リカバリー・レコードは、活動キーポイントの一部として書き込まれます。

### システム・ログからのユーザー・レコードの取得

ウォーム・リスタートおよび緊急リスタート中、CICS ではシステム・ログを逆方向にスキャンします。この逆方向スキャン中に CICS でアクティブなユーザー・レコードが検出された場合、それらのレコードは XRCINPT 出口点で有効になっているグローバル・ユーザー出口プログラムに対して提示されます。

### このタスクについて

アクティブなユーザー・ログ・レコードとは、以下のようなレコードです。

- 最後に完了した活動キーポイント中に XAKUSER グローバル・ユーザー出口プログラムによって書き込まれたレコード。
- API コマンドを使用してユーザー・トランザクションにより書き込まれたレコード。

作業単位チェーンの一部を形成するこれらのユーザー・レコードは、以下の場合にのみアクティブであると見なされます。

- JTYPEID フィールドの高位ビットが 1 に設定されている場合。

これらのレコードは、作業単位がリスタート中のシステムの逆方向スキンの有効範囲内にある場合は、作業単位の状態に関係なく、XRCINPT グローバル・ユーザー出口プログラムに提示されます。このカテゴリには、バックアウト失敗またはコミット失敗状態の作業単位も含まれます。

- コマンドが発行された作業単位が、前回実行された CICS の終了時に未完了であったか、未確定で失敗した場合。

これらのユーザー・レコードは、JTYPEID フィールドの高位ビットの状態に関係なく、常に、XRCINPT グローバル・ユーザー出口プログラムに提示されます。

CICS でシステム・ログのスキニングが完了し、アクティブなユーザー・レコードが検出されない場合は、メッセージ DFHER5731 が発行されます。

### システム・ログでの保持期間の回避

MVS へのシステム・ログを定義するときに保持期間 (RETPD=dddd) を指定した場合、CICS ログ・テール削除、つまり、完了したログ・レコードが、完了した 2 つの活動キーポイントに対してのみ保持されるようにする機能が禁止されます。

### このタスクについて

dddd 値は、データがログに保持される最小日数を指定します。

保持する必要があるレコードには、システム・ログを使用しないことを強くお勧めします。保持する必要があるすべてのログおよびジャーナル・データは、汎用ログ・ストリームに書き込む必要があります。を参照してください。汎用ログ・ストリーム・データ・セットの作成方法のアドバイスについては、[Setting up CICS log streams](#) を参照してください。

### 長期実行トランザクション

長期実行トランザクションの単一の作業単位が、複数の活動キーポイントをまたいで、システム・ログ・ストリームに頻繁に書き込むような設計は避けてください。

長期実行トランザクションが定期的な同期点を取らず、かつ、システム・ログへの書き込みを定期的に開始する場合、CICS では長期実行タスクが開始した後に完了した作業単位をページできなくなります。

453 ページの図 85 では、長期実行トランザクションがリカバリー可能リソースを更新し、これにより、システム・ログへの書き込みが発生しています。これは明示的な同期点を取らないため、すべての更新が同じ作業単位内にあります。最も古い削除点が活動キーポイント番号 5 よりも前にあるため、CICS では活動キーポイント 5 と 8 の間にある完了した作業単位をいずれも削除できません。この例では、長期実行トランザクションによって、最終的に、システム・ログ・ストリームが 2 次ストレージに予備的に格納されます。

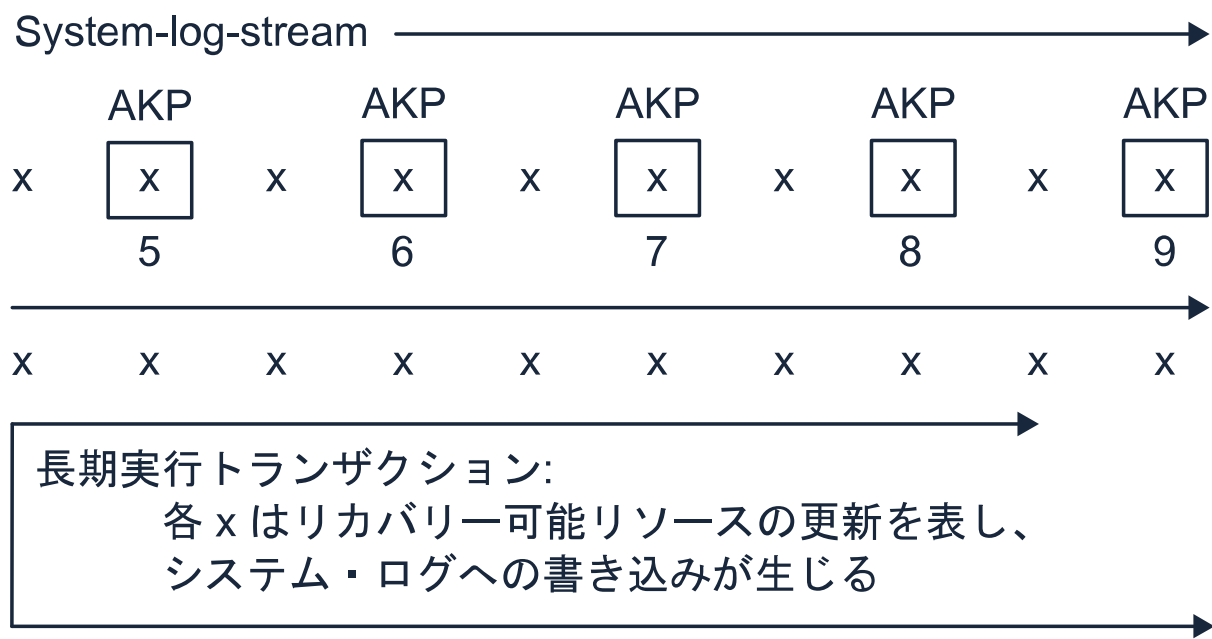


図 85. 「問題のある」長期実行トランザクションの影響

## 順方向リカバリー・ログ・ストリームの定義

リカバリー可能ファイルとして定義されている VSAM データ・セットの順方向リカバリー・ログを定義する必要があります。リカバリー不能データ・セットに対しては、CICS も VSAM も、順方向リカバリー・ロギングを一切サポートしません。

### このタスクについて

#### 手順

1. 順方向リカバリーを含むデータ・セットのリカバリー属性を、統合カタログ機能 (ICF) のカタログ (DFSMS 1.3 以降を使用している場合) または CICS ファイル・リソース定義のいずれかで定義します。  
詳しくは、[462 ページの『リカバリー可能リソースとしてのファイルの定義』](#)を参照してください。
2. 順方向リカバリー・データの汎用ログ・ストリームを定義します。  
汎用ログ・ストリームを定義しない場合、CICS ではログ・ストリームを動的に作成することを試みます。詳しくは、[454 ページの『CICS 汎用ログのモデル・ログ・ストリーム』](#)を参照してください。
3. 異なる CICS 領域からの順方向リカバリー・データを 1 つ以上のログ・ストリームにマージする方法を決定します。  
詳しくは、[455 ページの『共用汎用ログ・ストリームでのデータのマージ』](#)を参照してください。

#### 次のタスク

物理的な障害や破損が発生した場合は、最新のバックアップを復元し、CICS VSAM Recovery などの順方向リカバリー・ユーティリティを使用して更新を再適用します。

ストレージの障害からリカバリーする場合は、まず最新のバックアップを新規データ・セットに復元します。その後、CICS VSAM Recovery (CICS VR) などの順方向リカバリー・ユーティリティを使用して、バックアップ日付より後に順方向リカバリー・ログ・ストリームに書き込まれたすべての更新を適用します。

#### CICS 汎用ログのモデル・ログ・ストリーム

汎用ログ・ストリームが定義されていないために CICS がそのログ・ストリームへの接続に失敗した場合、CICS ではそれを動的に作成しようとします。

ログ・ストリームを動的に作成するためには、CICS が MVS システム・ロガーに、新しいログ・ストリームに必要なすべてのログ・ストリーム属性を指定する必要があります。さもないと不明になってしまうこれらの属性を判別するため、CICS は MVS システム・ロガーに対して、既存のモデル・ログ・ストリーム定義の属性を使用してログ・ストリームを作成することを要求します。ログ・ストリームの動的な作成を CICS に許可することにした場合は、必要なモデル・ログ・ストリーム定義を必ず作成して、動的な作成が正常に行われるようにしなければなりません。

一般に、モデル・ログ・ストリームをセットアップする価値があるのは、以下のような場合のみです。

- それぞれのモデル・ログ・ストリームが、複数のログ・ストリームの作成に使用される場合
- モデル・ログ・ストリームを使用して作成される各ログ・ストリームが類似した特性を持っており、同一のカップリング・ファシリティ構造を使用できる場合
- CICS アプリケーションによって使用されるジャーナル名が事前に分かっていない場合

上記以外の場合は、IXCMIAPU を使用してログ・ストリームを明示的に定義するほうが、恐らく作業量は少なく済みます。

CICS が汎用ログ・ストリームの動的な作成に使用するデフォルトのモデル・ログ・ストリームの名前は、`LSN_qualifier1.LSN_qualifier2.MODEL` という形式になります。ここで、最初および 2 番目の修飾子はそれぞれ、CICS 領域のユーザー ID および CICS APPLID です。

**例:** ユーザー ID に CICS SHA##、APPLID に CICS HAA1 を使用して実行されている CICS 領域で、JOURNALMODEL リソース定義で定義されていない汎用ログ・ストリームを作成する要求が発行された場合、CICS ではモデル・ログ・ストリーム名として `CICS SHA##.CICS HAA1.MODEL` を指定します。

XLGSTRM グローバル・ユーザー出口プログラムを使用してログ・ストリームの作成要求を変更する方法について詳しくは、[443 ページの『CICS システム・ログに対するモデル・ログ・ストリーム』](#)を参照してください。



## 共用汎用ログ・ストリームでのデータのマージ

1つの CICS 領域に対して一意であるシステム・ログ・ストリームとは異なり、汎用ログ・ストリームは多数の CICS 領域で共用することができます。これは、複数の CICS 領域からの順方向リカバリー・データを同じ順方向リカバリー・ログ・ストリームにマージできることを意味します。

### このタスクについて

- 複数のデータ・セットに対し、同じ順方向リカバリー・ログ・ストリームを使用できます。順方向リカバリー可能なデータ・セットごとにログ・ストリームを定義する必要はありません。
- 複数の CICS 領域間で順方向リカバリー・ログ・ストリームを共用できます。領域が RLS モードでデータ・セットにアクセスしている場合は、同じ順方向リカバリー・ログ・ストリームの使用が強制されます。

順方向リカバリー・ログ・ストリームをどのように定義し、割り振るかは、トランザクションのパフォーマンス、リカバリーの速さ、および管理するログ・ストリームの数が多数になることのトレードオフによって決定します。

考慮すべき点を以下に示します。

- 同じまたは異なる MVS イメージで実行されている複数の CICS 領域間で、カップリング・ファシリティ・ログ・ストリームを共用できます。DASD-only ログ・ストリームは、同じ MVS イメージで実行されている CICS 領域間でのみ共用できます。
- 1つのトランザクションで使用されるすべてのデータ・セットは、同じログ・ストリームを使用する必要があります(同期点で書き込まれるログ・ストリーム数を減らすため)。
- 以下のようなデータ・セット間で順方向リカバリー・ログ・ストリームを共用します。
  - 同様のセキュリティ要件を持っている
  - 同様のバックアップ頻度である
  - 同時に全体を復元する必要性が出てくる可能性が高い
- ディスパッチャーの待機では、ストリーム名の末尾の修飾子が CICS リソース名として使用されます。このため、分かりやすい修飾子を使用すると、モニター情報および CICS トレース・エントリーを解釈するときに役立ちます。
- 同じ順方向リカバリー・ログで、更新頻度の高いデータ・セットと更新頻度の低いデータ・セットを混合しないようにします。混合すると、更新頻度の低いデータ・セットのリカバリー中に、望ましくない大量のログ・データが読み取られます。
- すべての更新頻度の高いデータ・セットを単一のログ・ストリームに書き込まないようにして、ログ・ストリームのスループットの容量を超えないようにします。
- 単一ログ・ストリームに対して大量のデータ・セットを定義する場合、ログ・ストリームがデータ・フローに追いつかなくなると、頻繁に **structure-full** イベントが発生することがあります。
- 補助ストレージを保持し、システム・ロガーがログ・ストリーム当たりのデータ・セットの制限を超えないようにするために、冗長データをログ・ストリームから定期的に削除します。(上限は数百万であるため、通常環境ではこの上限を超えることはあまりありません。) ログ・ストリーム・データ・セットの管理について詳しくは、[補助ストレージの管理](#)を参照してください。
- ログ・ストリーム名をデータ・セットと関連性のあるものにします。例えば、PAYROLL.data\_sets は、PAYROLL.FWDRECOV.PAYLOG という名前の順方向リカバリー・ログにマップできます。
- カップリング・ファシリティ・ログ構造の制限があるため、順方向リカバリー・ログ・ストリームを多くしすぎないようにします。

## ログのログの定義

ログのログ・ログ・ストリームを定義し、そのログ・ストリームを参照する JOURNALMODEL リソース定義を定義します。

### このタスクについて

CICS 提供グループ DFHLGMOD には、DFHLGLOG と呼ばれるログのログの JOURNALMODEL が含まれ、これには &USERID.CICSVR.DFHLGLOG というログ・ストリーム名が付いています。&USERID は CICS 領域



のユーザー ID に解決され、ご使用の CICS 領域が異なる RACF ユーザー ID で実行されている場合は、DFHLGLOG 定義が領域ごとに固有のログのログ・ログ・ストリーム名に解決されることに注意してください。ただし、CICS VSAM Recovery などのリカバリー製品で個別のログのログを使用することは推奨されません。また、同じ CICSplex 内の CICS 領域間で共用されるログ・ストリームを定義することを検討する必要があります。例えば、ログのログの有効範囲を表す高位修飾子を選択して、同じ実動 CICSplex のメンバーであるすべての CICS 領域に対して 1 つのログ・ストリームを定義し、テスト領域に対して別のログ・ストリームを定義するとします。VSAM RLS を使用しており、複数の CICS アプリケーション所有領域が同じデータ・セットに直接アクセスする場合、ログのログに対して CICSplex 全体のログ・ストリームを使用することが特に重要です。

ログのログには、ファイルがオープンまたはクローズされるたびに書き込まれるレコードが含まれます。ファイル・オープン時には、以下を識別するタイアップ・レコードが書き込まれます。

- ファイルの名前
- VSAM データ・セットの名前
- 順方向リカバリー・ログ・ストリームの名前
- ファイルのオープンを実行した CICS 領域の名前。

ファイル・クローズ時には、以下を識別するタイアップ・レコードが書き込まれます。

- ファイルの名前
- ファイルのクローズを実行した CICS 領域の名前。

ログのログは、順方向リカバリー・ユーティリティがログ・データ・セットの索引を保持するのに役立ちます。

順方向リカバリーに加え、ログ・ストリーム・エラーの記録にもログのログは使用されます。ログ・ストリームで障害が発生した場合、DFHLGLOG が TYPE(DUMMY) と定義されていない限り、CICS はエラーおよび診断の情報を使用してログのログの更新を試みます。

### ログのログの障害

ログのログで障害が発生した場合、CICS ではメッセージを発行しますが、エラーは無視します。

それ以降のログのログへの書き込み試行は無視されます。パフォーマンスを最適化するためにログのログを使用する順方向リカバリー・プログラムでは、障害の発生後にログのログを使用できません。ログのログの障害の最も一般的な原因は定義エラーで、この場合、障害が実動データ・セットに影響を及ぼす可能性は低くなります。

### オフラインでのログ・ストリームの読み取り

MVS システム・ロガーのログ・ストリーム・データへのアクセスは、LOGR サブシステムのサブシステム・インターフェース (SSI) を通して提供されます。

### このタスクについて

SSI を使用してログ・ストリーム・データへのアクセスを取得するには、ログ・ストリーム DD ステートメントの SUBSYS パラメーターで LOGR を指定します。SUBSYS パラメーターで LOGR サブシステム名を指定することで、LOGR が SSI でデータ・セットのオープン要求および読み取り要求をインターセプトし、それらをログ・ストリーム・アクセスに変換することができるようになります。

汎用ログ・ストリーム・レコードは、SUBSYS パラメーターで指定されたオプションに応じて、以下のいずれかの形式で提示されます。

- ジャーナル制御プログラム DFHJUP を使用してロギングおよびジャーナリングを行う CICS のリリース向けに作成されたユーティリティ・プログラムと互換性のあるレコード形式
- ログ・マネージャーを使用してロギングおよびジャーナリングを行う CICS のバージョン向けに作成されたユーティリティ・プログラムと互換性のある形式

LOGR SSI を使用してログ・ストリーム・データにアクセスする方法の詳細およびサンプル JCL については、[バッチ・ジョブ \(DFHJUP\) の使用によるログ・ストリームの読み取り](#)を参照してください。

## 夏時間調整による変更の影響

多くの国で、夏の始まりと終わりに時計を 1 時間調整する、一般的に夏時間調整と呼ばれる制度が実施されています。

これらの時間変更は、コンピューターで保持されている現地時間にも適用されます。通常、ほとんどのハードウェア (TOD) クロックは、グリニッジ標準時 (GMT) に現地時間を示すオフセット値を反映した時刻に設定されています。夏時間調整を行う場合はこのオフセット値を調整し、ハードウェア・クロックは変更しません。

### 現地時間の自動調整

システム初期設定パラメーター **AUTORESETTIME=IMMEDIATE** を指定して、CICS 領域の実行中に MVS TOD クロックでシステムの日付または時刻が変更された場合は、必ず、CICS 時刻を z/OS 時刻と即時に同期するようにします。**AUTORESETTIME=IMMEDIATE** を指定すると、次のタスク接続時に CICS の時刻がシステム時刻と異なる場合に、CICS が **PERFORM RESET** コマンドを発行して CICS の時刻をシステム時刻と同期するようになります。**AUTORESETTIME** のデフォルト設定は IMMEDIATE です。これとは異なる設定を指定している場合は、MVS TOD クロックの変更直後に必ず **CEMT PERFORM RESET** または **EXEC CICS PERFORM RESETTIME** コマンドを手動で発行するプロセスを設定していることを確認してください。

### 現地時間の手動調整

CICS 領域の実行中に、夏時間および冬時間の調整のためにクロックを 1 時間進めるか戻して設定した場合は、**CEMT PERFORM RESET** または **EXEC CICS PERFORM RESETTIME** コマンドを使用して、CICS がその現地時間を MVS TOD クロックの現地時間と即時に再同期するようにしてください。

CICS は、始動時および **CEMT PERFORM RESET** コマンドの実行時に現地時間オフセットを取得して保管します。CICS の実行中にシステムの日付または時刻を変更した場合は、必ず、即時に **CEMT PERFORM RESET** コマンドを使用して、API を含むすべての CICS 機能で正しい現地時間を使用されるようにしてください。アプリケーション・プログラムによって **EXEC CICS ASKTIME** コマンドが発行されるたびに、CICS では MVS TOD クロックから現在時刻を取得して、保管されている現地時間との差でこれを変更します。次に、CICS は、現地時間を使用して EIBTIME フィールドを更新します。

### ログおよびジャーナル・レコードへのタイム・スタンプの設定

現地時間が前後に変更されても、CICS のロギングまたはジャーナリング、または CICS のリスタートへの影響はありませんが、DFHJUP などのユーティリティ・プログラムの動作に影響を及ぼす可能性があります。

CICS では、以下のようにして、システム・ログに書き込むデータにタイム・スタンプを設定します。

- システム・ログ・ブロック・ヘッダーには、マシン・クロック (STCK) 値および現地時間の両方を使用してタイム・スタンプが設定されます。
- システム・ログ・レコードには、マシン・クロック (STCK) 値のみを使用してタイム・スタンプが設定されます。

汎用ログの場合、CICS ではシステム・ログと同様のタイム・スタンプの設定に加えて、ジャーナル・レコードに現地時間も含まれます。

リスタート中、システム・リカバリーの目的で、CICS は 1 次ログ・ストリームから最新の (最後に書き込まれた) レコードを読み取ります。その後、CICS ではブロック ID を使用した直接読み取りのみを使用し、タイム・スタンプは利用しません。CICS は、トランザクションのバックアウトの目的でログ・データを取得するためにもブロック ID による直接読み取りを使用し、この場合も、タイム・スタンプに依存しません。

システム・ログ・データのタイム・スタンプに依存しないリカバリー・プロセスを実行することで、CICS では、現地時間が戻された直後に障害が発生した場合でも、異常終了の後に確実に正常にリスタートできます。

## オフライン・ユーティリティ・プログラム、DFHJUP

現地時間を進めるように変更しても、CICS ユーティリティ・プログラム、DFHJUP によるシステム・ログ・ストリームまたは汎用ログ・ストリームの処理に影響はありません。

現地時間を戻すように変更しても、DFHJUP JCL でログ・ストリーム DD ステートメントの SUBSYS パラメーターで GMT オプションを指定している場合には、DFHJUP の操作に影響はありません。

ただし、DFHJUP による処理用のログ・ストリームのパーティション化を指定するために SUBSYS パラメーターで現地時間を使用している場合は、クロックを戻すように調整したとき、タイム・スタンプの日時順を確保するための手順を実行する必要があります。このことを行うには、新しい現地時間が、変更が加えられた時点の古い時間を過ぎるまで、CICS 領域を停止します。

ユーザーまたはベンダー作成のジャーナル・ユーティリティおよび DFHJUP 出口プログラムも、現地時間の変更の影響を受ける場合があります。これらについては、時間を戻す変更による問題がないかどうかを確認する必要があります。

順方向リカバリー・ユーティリティ（ただし CICS VSAM Recovery for z/OS 2.3 を除く）も、順方向リカバリー・ログ・データの時系列の影響を受ける場合があります。CICS VSAM Recovery を使用していない場合は、使用している順方向リカバリー・ユーティリティでログ・レコードの不連続性を処理できることを確認してください。

CICS VSAM Recovery を使用している場合は、以下の状況でのみ、順方向リカバリーで問題が発生します。

- (現地) 時間を変更してから 1 時間以内にバックアップを取得する場合 (例えば、時間を 02.00 から 01.00 に設定し直した後、01.30 に取得する場合)
- 時間変更をまたぐログ・レコードを含むバックアップを使用して、順方向リカバリーを実行する場合
- GMT ではなく現地時間を指定した場合

新しい現地時間より前に取得されたバックアップを使用する場合、または GMT を指定している場合は、CICS VSAM Recovery によって順方向リカバリーが正常に処理されます。

## CICS 管理リソースのリカバリーの構成

このセクションでは、CICS がアプリケーション・プログラムのために制御しているリソースを、確実にリカバリーできるようにするために行うことについて説明します。

### Liberty JVM サーバーでのリソースのリカバリー

Java Transaction API (JTA) を使用して、トランザクションの更新を調整できます。ローカルおよびリモート・トランザクションに対し、限定的な CICS リカバリー可能性オプションが適用されます。

Java Web アプリケーションで JTA を使用して、CICS リソースおよびその他のサード・パーティー製 XA リソース・マネージャーに対するトランザクション更新を調整できます。[Java Transaction API \(JTA\)](#)を参照してください。

このシナリオでは、Liberty トランザクション・マネージャーがトランザクション・コーディネーターとなり、zFS ファイリング・システムの JTA トランザクション・ログにリカバリー情報を保管します。緊急時再始動が発生すると、JVM サーバーが自動的に再インストールされ、Liberty トランザクション・マネージャーは JVM サーバーの初期化が完了するとすぐに XA トランザクション・リカバリーを開始します。このプロセスによって、CICS の未確定の作業単位およびサード・パーティー製リソース・マネージャーのトランザクションがすべてリカバリーされることが確実にあります。

リカバリー可能な更新が未完了の状態で Liberty JVM サーバーで障害が発生した場合に、CICS が正常にリカバリーを実行できるようにするには、CICS システム・ログおよび Liberty トランザクション・ログ両方の健全性が重要です。、Liberty JVM サーバーの構成方法、および Liberty トランザクション・ログの保管に使用する zFS ディレクトリーの場所の制御方法について詳しくは、[server.xml の手動調整](#)を参照してください。CICS システム・ログのセットアップについて詳しくは、[CICS システム・ログ](#)を参照してください。

## トランザクションのリカバリー

ローカルおよびリモート・トランザクションに対し、限定的な CICS リカバリー可能性オプションが適用されます。

ローカル・トランザクションには、以下のオプションが適用されます。

- トランザクションの異常終了時の自動リスタート
- トランザクションが、指定された時間間隔よりも長い時間リソースを待機している場合のタイムアウト。このことは、通常、そのトランザクションと別のトランザクションでの、同じリソースに対するデッドロックによって発生します。
- 他の方法では解決できない理由でハングしているトランザクションのページの許可

リモート・トランザクションの場合、CICS では、ローカル・トランザクションのオプションに加えて未確定障害が発生した場合のリカバリー・アクションを指定するための未確定オプションを提供します。

### トランザクション・リカバリー・オプション

ローカル・リソースを更新するユーザー・トランザクションを定義する場合、リカバリーのために TRANSACTION リソース定義で以下のオプションを指定できます。

#### RESTART({NO|YES})

このオプションは、特定の状況で、トランザクションが DFHREST ユーザー置き換え可能モジュールの制御下で CICS による自動リスタートの対象となるかどうかを定義します。

デフォルトは RESTART(NO) です。

#### DTIMOUT({NO|1-6800})

指定された間隔で、タスクの中断が継続した場合は、要求がタイムアウトします。CICS WEB API コマンドが使用された場合、TIMEDOUT 応答がアプリケーションに戻されます。その他の状況では、SPURGE(YES) が指定されている場合、CICS によってタスクの異常終了が開始されます。

この値は *mmss* 形式で指定します。ここで *mm* は分数、*ss* は秒数を表します。例えば、3420 という値は 34 分 20 秒を意味します。最大値は 6800 (つまり 68 分) です。

デフォルトは DTIMOUT(NO) です。

#### SPURGE({YES|NO})

このオプションは、トランザクションが初期設定でシステムによるページ可能かどうか、つまり、CICS が以下の結果としてトランザクションをページできるかどうかを指定します。

- デッドロック・タイムアウト (DTIMOUT) 遅延間隔の経過
- CEMT または EXEC CICS、SET TASK(id) PURGE コマンド

デフォルトは SPURGE(YES) です。

#### TPURGE({YES|NO})

このオプションは、(非 z/OS Communications Server) 端末エラーが発生した場合に、システムによるトランザクションのページが可能かどうかを指定します。

デフォルトは TPURGE(YES) です。

### 分散トランザクションの未確定オプション

リモート・リソースを更新するユーザー・トランザクションを定義する場合、リモート・リカバリーのために以下のオプションを指定できます。

#### ACTION({BACKOUT|COMMIT})

作業単位が未確定期間に入った後の 2 フェーズ・コミット処理中に障害が起こった場合に実行するアクションを指定します。

WAIT(YES) を指定した場合、リカバリーが行われる前に WAITTIME で指定された間隔が経過しない限り、アクションは実行されません。

BACKOUT を指定するか COMMIT を指定するかは、通常、リモート・システムのリソースに、どのような種類の変更が加えられたかによって異なります。

#### **WAIT({YES|NO})**

障害が発生した未確定の作業単位が、障害のリカバリー処理を保留にしながら未確定状態の解決を待機するか、または CICS が ACTION オプションで指定されたアクションを即時に実行するかを指定します。

注：WAIT(YES) オプションは、区画内一時データ・キューで定義された WAIT(NO) によりオーバーライドできます。トランザクションがアクセスする TD キューで WAIT(NO) が指定され、かつトランザクションで WAIT(YES) が指定されている場合は、TD キュー定義によって、トランザクション・リソース定義の ACTION 属性の指定に従い、トランザクションのバックアウトまたはコミットが強制されます。

#### **WAITTIME({00,00,00|dd,hh,mm})**

WAIT(YES) の場合、トランザクションがどのくらい待ってから、ACTION で指定されたアクションを実行するかを指定します。

WAIT と WAITTIME を使用すると、通常のリカバリーと再同期が行われる機会を与えると同時に、トランザクションによって適切な時間内にコミットまたはバックアウトが確実に実行されるようにすることができます。分散トランザクションの待ち時間の定義について詳しくは、「[トラブルシューティング](#)」の『[トランザクション定義の未確定属性](#)』を参照してください。

#### **さらに学習したい方に**

分散作業単位のリカバリーについて詳しくは、[システム間の問題のトラブルシューティング](#)を参照してください。

TRANSACTION 定義のオプションについて詳しくは、[TRANSACTION 属性](#)を参照してください。

### **ファイルのリカバリー**

CICS ファイルは、CICS に対して、ファイル・リソース定義で 8 文字のファイル名を使用して定義される物理データ・セットの論理ビューです。

CICS ファイルは、以下のいずれかによって、VSAM または BDAM データ・セットに関連付けられています。

- 動的割り振り。この場合、データ・セット名はファイル定義の DSNAME パラメーターで事前定義されています (または、CEMT、または EXEC CICS SET FILE DSNAME(name) コマンドにより設定されます)。
- ジョブ・ステップ開始時の JES による割り振り。この場合、データ・セット名は DD ステートメントで定義されます。

複数のファイルが同じデータ・セットを参照できます。

データ・セットは、DASD 上に存在する物理オブジェクトとして定義されます。これは、44 文字の DSNAME を持ちます。例えば、VSAM データ・セットは、VSAM アクセス方式サービス (AMS) を使用して定義されます。

アプリケーションを設計するときは、必ず、データの破損を引き起こす可能性のあるトランザクション障害からアプリケーション・データを保護し、ストレージ・デバイスが不測の損傷を受けた場合にリカバリーできるようにしてください。

アクセス方式を決定するときには、各方式によって提供されるリカバリーおよびリスタート機能を検討してください。それらの考慮事項について、以下のトピックで説明しています。

#### **VSAM ファイルのリカバリー**

CICS ファイル制御は、ローカル共用リソース (LSR)、非共用リソース (NSR)、およびレコード・レベル共用 (RLS) の 3 つの VSAM アクセス・モードをサポートしています。

#### **バッチ・ジョブとのデータ・セットの共用**

非 RLS アクセス・モードでは、VSAM 共用オプション (使用可能な場合) またはジョブ制御共用を使用して、オンライン CICS 更新トランザクションとバッチ更新プログラムとの間でデータ・セットを直接共用することは推奨されません。共用すると、アプリケーション・プログラムが正常に機能しているように見えるも



の、誤ったデータが使用されることで、データの保全性に対するリスクが発生します。そのようなデータ保全性の欠如は、例えば、CICS 作業単位によってレコードが更新され、その後、その CICS 作業単位がまだ実行されている間に、そのレコードが非 CICS ジョブによって更新された場合に発生する可能性があります。CICS 作業単位が異常終了すると、CICS は CICS 作業単位の開始時に設定されていた値にレコードをバックアウトするため、非 CICS ジョブからの更新が破棄されます。

## ファイル所有領域および RLS アクセス

複数の CICS 領域間でデータ・セットを共用するには、RLS アクセス・モードを使用するか、アプリケーション所有領域 (AOR) からの機能シップと共にファイル所有領域 (FOR) を使用します。リカバリーの観点から、RLS は AOR と FOR の間で行われる場合、分散作業単位を作成しません。これは、RLS モードでアクセスされるファイルはすべて、各 CICS 領域に対してローカルであると見なされるためです。SMSVSAM サーバーが FOR の代わりになります。ただし、RLS モードでアクセスされるデータ・セットには、SMSVSAM の役割およびそのロック管理について特別なリカバリーの考慮事項があります。

## 順方向リカバリー

VSAM ファイルでは、データ・セットに対する物理的な損傷の結果としてオンライン・バックアウト処理が失敗した場合に、CICS VSAM Recovery for z/OS などの順方向リカバリー・ユーティリティを使用できます。順方向リカバリーでは、以下の操作を実行します。

- データ・セットのバックアップ・コピーを作成します。
- 順方向リカバリー・ストリームでのファイル変更後イメージを記録します。ファイルに対する順方向リカバリー・サポートが必要であると指定した場合、CICS はこれを自動的に実行します。
- 順方向リカバリー・ユーティリティを実行するジョブを作成し、入力として必要な可能性があるバックアップ・データ・セットおよびログ・ストリームを制御します。CICS VSAM Recovery for z/OS は、ISPF ダイアログ・インターフェースを使用して、順方向リカバリー・ジョブを自動的に作成します。

## 逆方向リカバリー

VSAM ファイルの逆方向リカバリーを確実に実行できるようにするには、以下の点を考慮する必要があります。

- キー順データ・セット (KSDS) および固定長および可変長両方の相対レコード・データ・セット (RRDS):
  - KSDS または RRDS データ・セットを参照するファイルが、LOG(ALL) の指定によってリカバリー可能に指定されている場合、CICS は中断された作業単位によって行われたすべての更新、追加、および削除をバックアウトできます。
  - バックアウトの障害について詳しくは、[バックアウト障害からのリカバリー](#)を参照してください。
- 入力順データ・セット (VSAM-ESDS):
  - 新しいレコードは VSAM-ESDS の末尾に追加されます。追加されたレコードは、物理的に削除できなくなります。論理削除は、レコードでデータを変更することによってのみ実行できます。例えば、XFCLDEL グローバル・ユーザー出口プログラムを使用して、レコードに「論理削除」フラグを設定します。

詳細については、[トランザクション・バックアウト](#)を参照してください。

## レプリカ生成ロギング

データ・セットに **LOGREPLICATE** が定義されている場合、VSAM ファイルはレプリカ生成に関与できます。**LOGREPLICATE** を定義すると、レプリカ生成機能による 2 番目のサイトでのファイル更新の複製が可能になります。

VSAM ファイルを使用してレプリケーション・ロギングを実装するときには、システム・トランザクション CFCT およびそれと関連付けられたプログラム DFHFCLJ1 を使用して、VSAM ファイルのタイアップ・レコードを複製ログに指定間隔で提供できます。

CFCT を使用可能にするには、**INITPARM** システム初期設定パラメーターを **INITPARM=(DFHFCLJ1='nn')** というフォーマットで指定します。ここで、**nn** は分単位の間隔を定義す



る 01 から 60 の範囲の 2 桁の数値です。この範囲外の間隔値を指定すると、メッセージ DFHFC6045 が発行され、デフォルト値の 30 分が代わりに使用されます。

INITPARM=(DFHFCLJ1='nn') を使用してシステムを始動した後に、LOGREPLICATE 属性で定義された VSAM ファイルが開かれたことを CICS が検出すると、DFHFCLJ1 は、DFHFCLJ、ファイル制御ロギング、およびジャーナリング・プログラムを指定された間隔で呼び出します。詳しくは、[DFHFCLJ1 \(ファイル制御タイアップ・レコード・レプリケーター\)](#)を参照してください。

### 基本直接アクセス方式 (BDAM)

CICS では、BDAM ファイルの順方向リカバリーをサポートしていません。自動ジャーナリング・オプションを使用して、独自の順方向リカバリー・サポートを実装できます。

BDAM データ・セットのバックアウトは、データ・セットからレコードを削除できないという点で、ESDS データ・セットのバックアウトと同じです。

### リカバリー可能リソースとしてのファイルの定義

このセクションでは、CICS ファイル制御が管理するファイルのリカバリー属性の定義方法を説明します。

### このタスクについて

順方向リカバリーを含むリカバリー・オプションは、統合カタログ機能 (ICF) のカタログ (DFSMS 1.3 以降を使用している場合) または CICS ファイル・リソース定義のいずれかで、以下のように指定します。

- VSAM データ・セットが CICS から RLS モードでアクセスされる場合は、ICF カタログでリカバリー属性を定義する必要があります。
- VSAM データ・セットが CICS から非 RLS モードでアクセスされる場合は、FILE リソースまたは ICF カタログのいずれかでリカバリー属性を定義できます。ICF カタログを使用して非 RLS モードでアクセスされるデータ・セットの属性を定義する場合、CICS では、FILE リソースの代わりに ICF カタログ・エントリー・リカバリー属性を使用します。カタログではなく FILE リソース属性を使用することを CICS に強制する場合は、**NONRLSRECOV** システム初期設定パラメーターを FILEDEF に設定します。
- BDAM ファイルのリカバリー属性は、ファイル管理テーブル (FCT) のファイル・エントリーで定義します。

### 非 RLS モードでの VSAM ファイルへのアクセス

RECOVERY および FWDRECOVLOG オプションを使用して、VSAM ファイルの順方向リカバリーおよび逆方向リカバリー両方のサポートを指定できます。**BACKUPTYPE** パラメーターを使用して、必要なデータ・セット・バックアップのタイプを定義します。

### RECOVERY(ALL)

RECOVERY(ALL) を指定すると、CICS はファイルに順方向およびバックアウト両方のリカバリーを提供します。CICS リカバリー・マネージャーおよびログ・マネージャーのサービスを使用して、CICS ファイル制御は以下の書き込みを実行します。

- システム・ログ・ストリームへの更新レコードの変更前イメージの書き込み。これらは、トランザクションの異常終了後、または CICS の異常終了後の緊急リスタート時に、ファイルの変更をバックアウトするために使用されます。
- FWDRECOVLOG オプションにより参照される汎用ログ・ストリームへの変更後イメージの書き込み。変更後イメージは、以下のもので構成されます。

- 新規レコードが追加されたときの write\_add\_complete レコード
- レコードが削除されたときの write\_delete レコード
- レコードが更新されたときの write\_update レコード

ファイル制御では、順方向リカバリー・ログに以下のレコードも書き込みます。

- FILE\_OPEN タイアップ・レコード
- FILE\_CLOSE タイアップ・レコード
- TAKE\_KEYPOINT タイアップ・レコード

#### – データ・セット BACKUP タイアップ・レコード

CICS 順方向リカバリー・サポートは、自動ジャーナリングとは完全に独立しています。ただし、ファイル・リソース定義の JOURNAL および関連する JNLxxx オプションにより制御される自動ジャーナル・レコードは、順方向リカバリー・データと同じ汎用ログ・ストリームに書き込むことができます。これを行うには、同じ順方向リカバリー・ログ・ストリームにマップされるジャーナル ID を JOURNAL オプションで指定します。ジャーナル・モデル・リソース定義を使用したログ・ストリームへの CICS ジャーナル ID のマッピングについては、[55 ページの『Setting up CICS log streams』](#)を参照してください。

**注:** 順方向リカバリー目的での自動ジャーナル・レコードの使用は、VSAM ファイルでは推奨されません。

### RECOVERY(BACKOUTONLY)

RECOVERY(BACKOUTONLY) を指定すると、CICS はシステム・ログに変更前イメージのみを書き込むことで、バックアウト・リカバリーのみを提供します。

### BACKUPTYPE(DYNAMIC)

BACKUPTYPE(DYNAMIC) を指定した場合、CICS は、データ・セットがオープンしているときにバックアップできるようにする DFSMS backup-while-open (BWO) 機能をサポートします。STATIC を指定した場合、データ・セットに対してオープンしているすべての CICS ファイルを、バックアップする前にクローズする必要があります。backup-while-open (BWO) 機能については、[Back-up-while-open \(BWO\)](#)を参照してください。

### RLS モードでの VSAM ファイルへのアクセス

RLS モードでデータ・セットをオープンするファイル定義を指定する場合は、ICF カタログにリカバリー・オプションを指定します。

ファイル定義で RLS アクセスが指定されている場合、CICS ファイル・リソース定義のリカバリー・オプション (RECOVERY、FWDRECOVLOG、および BACKUPTYPE) は無視されます。

アクセス方式サービスの DEFINE CLUSTER および ALTER コマンドで指定される VSAM パラメーター LOG および LOGSTREAMID は、範囲全体のリカバリー可能性を決定します。これらのリカバリー・パラメーターを ICF カタログ内で指定すると、所定の範囲に対してオープンしているすべてのファイルについて、シスプレックス内のすべての CICS 領域で同一のオプションの使用が強制されます。

### LOG({NONE|UNDO|ALL})

VSAM 範囲で必要なリカバリーのタイプを指定します。RLS モードで CICS が使用するためのデータ・セットの LOG パラメーターを指定します。

#### なし

範囲はリカバリー不能です。

#### UNDO

範囲はリカバリー可能です。CICS はバックアウトのためにシステム・ログ・レコードを保持する必要があります。

#### ALL

範囲は、バックアウトおよび順方向リカバリーの両方でリカバリー可能です。CICS は、システム・ログ・レコード (UNDO の場合と同様) および順方向リカバリー・ログ・レコードを保持する必要があります。LOG(ALL) を指定する場合は、LOGSTREAMID も指定して順方向リカバリー・ログ名を示します。

**注:** 順方向リカバリー・サポートは、リカバリー可能ファイルに対してのみ使用可能です。バックアウト・リカバリーなしで順方向リカバリーを行うことはできません。

### LOGSTREAMID(log\_stream\_name)

LOG(ALL) が定義されている場合に、順方向リカバリー・ログ・レコードに使用するログ・ストリームの名前を指定します。IDCAMS は、DEFINE 処理中に LOGSTREAMID が存在するかどうかの検査を行わないことに注意してください。CICS は、LOG(ALL) で定義されているデータ・セットのオープンを試行するときに、ログ・ストリーム名を検査します。ログ・ストリームが定義されておらず、動的に作成できない場合、オープンが失敗します。

VSAM データ・セットの定義の際に LOG パラメーターを省略すると、リカバリーは UNDEFINED であると想定され、RLS モードでデータ・セットをオープンできません。NULLIFY(LOG) を指定することで UNDEFINED 状況を明示的に設定することもできます。

アクセス方式サービスの DEFINE および ALTER コマンドについて詳しくは、[z/OS DFSMS アクセス方式サービス・コマンド](#)および [z/OS DFSMS データ・セットの使用法](#)を参照してください。

リカバリー属性についての照会

CEMT または EXEC CICS INQUIRE FILE および INQUIRE DSNAME コマンドを使用して、ファイルおよびデータ・セットに指定されたリカバリー・オプションを決定できます。

INQUIRE FILE コマンドは、データ・セットの最初のファイルがオープンされるまでは、CICS ファイル定義のオプションを表示します。最初のファイルのオープン時に ICF カタログからオプションが取得される場合は、その ICF カタログ値が戻されます。INQUIRE DSNAME コマンドでは、VSAM 基本クラスター・ブロック (BCB) からの値が戻されます。ただし、最初のファイルがオープンされるまでは基本クラスター・ブロック (BCB) のリカバリー値は設定されないため、最初のファイル・オープンの前に INQUIRE DSNAME コマンドを発行すると、CICS は RECOVSTATUS に NOTAPPLIC を戻します。

### BDAM ファイル

DFHFCT TYPE=FILE マクロで LOG パラメーターを使用して、BDAM ファイルの逆方向リカバリーに対する CICS サポートを指定できます。FCT マクロの JREQ および JID パラメーターを使用して、さまざまなタイプのファイル・アクセスの自動ジャーナリングが必要であることを指定することもできます。

#### LOG=YES

LOG=YES を指定すると、CICS からファイルのバックアウト・リカバリーが提供されます。CICS リカバリー・マネージャーおよびログ・マネージャーのサービスを使用して、CICS ファイル制御はシステム・ログ・ストリームに更新レコードの変更前イメージを書き込みます。

#### JREQ=request-type(s) および JID=nn

CICS で、JID パラメーターで指定されたジャーナル ID によりマップされているログ・ストリームへの自動ジャーナリングを行う必要があるというファイル要求を指定します。

CICS では BDAF ファイルの順方向リカバリー・サポートを提供しませんが、ユーザーは自動ジャーナル・レコードを使用して独自の機能を提供できます。JREQ=(WU,WN) は、CSD ファイル定義のパラメーター JNLUPDATE(YES) および JNLADD(BEFORE) を組み合わせて指定した場合に相当し、JID=nn で指定されたジャーナルへの順方向リカバリーに必要なイメージを提供します。

BDAM ファイル・リソース定義について詳しくは、[FILE リソース](#)を参照してください。

### CSD データ・セット

CICS システム定義 (CSD) VSAM データ・セットは特別なデータ・セットで、CICS ファイル制御が管理する CICS システム・データ・セットです。このため、CSD を CSD で定義することはできず、代わりにシステム初期設定パラメーターにより定義されます。

RECOVERY、FWDRECOVLOG、および BACKUPTYPE に相当するリカバリー・オプションは、それぞれ CSDRECOV、CSDFRLOG、および CSDBKUP です。CSD の定義について詳しくは、[21 ページの『第 3 章 共用データ・セット、CSD、および SYSIN のセットアップ』](#)を参照してください。

### ファイル・リカバリー属性の整合性チェック (非 RLS)

非 RLS モードでアクセスされるデータ・セットでは、同じ基本データ・セット・クラスターを参照するファイル間またはそのパス間で、リカバリー属性に整合性があることを確認します。

基本データ・セットに対して最初にオープンするファイルによって、基本データ・セットのリカバリー属性が決定され、これらが基本クラスター・ブロック (BCB) に保管されます。後続のファイル・オープン要求で、ファイル定義のリカバリー属性と BCB に保管されている属性との間の不整合が CICS によって検出された場合、オープン要求は失敗します。

ファイルおよびデータ・セットのリカバリー属性の確認について詳しくは、[464 ページの『リカバリー属性についての照会』](#)を参照してください。

## XFCNREC グローバル・ユーザー出口でのオープンの失敗のオーバーライド

CICS では、グローバル・ユーザー出口点 XFCNREC を提供して、同じデータ・セットに関連付けられているファイルのバックアウト設定に不整合があっても、ユーザーが処理を続行できるようにします。

### このタスクについて

XFCNREC を使用して、バックアウト設定の不整合が原因で発生したオープンの失敗を抑制している場合、CICS から、データ・セットの保全性を保証できなくなったことを警告するメッセージが発行されます。

この時点以降に発行されるすべての **INQUIRE DSNAME RECOVSTATUS** コマンドでは、CICS が基本クラスターにこれまで設定していたリカバリー属性に関係なく、NOTRECOVERABLE を戻します。この状態は、(**SET DSNAME REMOVE** コマンドを指定して) データ・セットの基本制御ブロックを削除するか、コールド・スタートを実行するまで続きます。

同じ基本データ・セットに対するファイルのオープン順序によって、XFCNREC を使用してオープンの失敗を抑制している場合に受信するメッセージの内容が決まります。基本クラスター・ブロックがリカバリー不能であると設定され、かつ不整合が許可されている場合、データ・セットが完全にリカバリーされる前に、リカバリー不能ファイルを介したデータ・セットへのアクセスが許可されることがあります。

XFCNREC のプログラミング情報については、[サンプル NEP](#) を参照してください。

### ファイル・オープン要求への CICS 応答

CICS ファイル制御では、ファイル定義のバックアウト設定を使用して、ファイル要求の変更前イメージをログに記録するかどうかを決定します。

CICS では、非 RLS モードで更新処理のためにファイルをオープンする際、つまり、ファイル定義で RLSACCESS(NO) が指定されており、操作パラメーターで ADD(YES)、DELETE(YES)、または UPDATE(YES) が指定されている場合に、以下のリストに示すアクションを実行します。READ(YES) のみ、BROWSE(YES) のみ、またはこの両方のみを設定している場合、CICS ではこれらの整合性チェックを行いません。これらのチェックは、リソース定義でまたはインストール時には行われません。

- ファイル定義が代替索引パスを参照し、かつ、RECOVERY が ALL または BACKOUTONLY である場合、代替索引が基本アップグレード・セットに含まれている必要があります。これは、基本データ・セットに対して行われたすべての変更が、代替索引にも反映されることを意味します。代替索引がアップグレード・セットに含まれていない場合、この代替索引パスに対する ACB のオープン試行が失敗します。
- 最後のコールド・スタート後に、基本クラスターに対してファイルを初めてオープンする場合、ファイル定義のリカバリー・オプションが基本クラスター・ブロックにコピーされます。
- 最後のコールド・スタート後に、基本クラスターに対して更新のためにファイルをオープンするのが初めてではない場合は、ファイル定義のリカバリー・オプションが、最初のファイルのオープンにより基本クラスター・ブロックにコピーされたオプションに対してチェックされます。以下のような場合が考えられます。
  - 基本クラスターに RECOVERY(NONE) がある場合:
    - ファイルに RECOVERY(NONE) が定義されている場合: オープンが続行されます。
    - ファイルに RECOVERY(BACKOUTONLY) が定義されている場合: 同じ基本データ・セットに関連付けられているファイルのバックアウト設定の不整合を許可できる XFCNREC グローバル・ユーザー出口プログラムによりオーバーライドされていない限り、ファイルのオープン試行が失敗します。
    - ファイルに RECOVERY(ALL) が定義されている場合: オープンが失敗します。
  - 基本クラスターに RECOVERY(BACKOUTONLY) がある場合:
    - ファイルに RECOVERY(NONE) が定義されている場合: 同じ基本データ・セットに関連付けられているファイルのバックアウト設定の不整合を許可する XFCNREC グローバル・ユーザー出口プログラムによってオーバーライドされていない限り、ファイルのオープン試行が失敗します。
    - ファイルに RECOVERY(BACKOUTONLY) が定義されている場合: オープンが続行されます。
    - ファイルに RECOVERY(ALL) が定義されている場合: オープンが失敗します。
  - 基本クラスターに RECOVERY(ALL) がある場合:
    - ファイルに RECOVERY(NONE) が定義されている場合: オープンが失敗します。



- ファイルに RECOVERY(BACKOUTONLY) が定義されている場合: オープンが失敗します。
- ファイルに RECOVERY(ALL) が定義されている場合: FWDRECOVLOG で基本クラスターとは異なるジャーナル ID が指定されている場合を除き、オープンが続行されます。指定されている場合は、オープンが失敗します。

データ・セットに対するファイルのオープンが失敗した場合は必ず、コンソールにメッセージが表示されます。必要に応じて、リカバリー・オプションを変更する必要があります。(基本クラスター・ブロックに保持されている) VSAM データ・セットのリカバリー属性を変更するには、CEMT または EXEC CICS の SET DSNAME REMOVE コマンドを使用します。これにより、基本クラスター・ブロックが削除されるため、VSAM データ・セット用の、前のリカバリー設定値のレコードが、すべて、CICS から失われます。このデータ・セットに対して次にオープンされるファイルによって、新しい基本クラスター・ブロックが構築され、そのファイルが更新のためにオープンされた場合、データ・セットにはこのファイルのリカバリー属性が引き継がれます。

基本クラスター・ブロックとそのリカバリー属性、および XFCNREC を使用している場合に設定されている不整合条件は、ブロックに関連するすべてのファイルがクローズしている場合、およびウォーム・リスタートおよび緊急リスタートの後であっても保持されます。

### ユーザー作成のユーティリティによる 順方向リカバリーの実装

独自の順方向リカバリー・プログラムを使用する場合は、プログラムで順方向リカバリー・ログ・ストリームからの変更後イメージが使用されていることを確認する必要があります。汎用ログ・ストリームに書き込まれた自動ジャーナル・レコードの使用、VSAM 順方向リカバリーでは推奨されません。

### このタスクについて

順方向リカバリーの実行手順について詳しくは、[順方向リカバリー手順](#)を参照してください。

### CICS VSAM Recovery による 順方向リカバリーの実装

CICS VSAM Recovery for z/OS を使用して、失われたまたは損傷を受けた VSAM データ・セットをリカバリーできます。

### このタスクについて

詳しくは、[CICS VSAM Recovery for z/OS](#) を参照してください。

## 区画内一時データのリカバリー

このセクションでは、区画内一時データの逆方向と順方向の両方のリカバリーについて説明します。

### 逆方向リカバリー

CICS では、区画内一時データのみをリカバリーできます。区画内データ・セットは、DFHINTRA というファイル名の VSAM ESDS データ・セットです。

割り振りおよびスペース所要量について詳しくは、[52 ページの『区画内データ・セットの定義』](#)を参照してください。[区画外一時データに関する考慮事項](#)については、[469 ページの『区画外一時データのリカバリー』](#)を参照してください。

キュー定義でリカバリー可能にする必要があるすべての区画内一時データ・キューの名前を指定する必要があります。区画内一時データ・キューに指定できるリカバリー属性は以下のとおりです。

- 論理
- 物理
- なし

### 論理リカバリー

区画内キュー定義で論理リカバリーを要求すると、中断された作業単位による一時データ・キューの変更がバックアウトされます。バックアウトは、タスクが異常終了した場合は動的に行われ、CICS に障害が発生した場合は CICS の緊急リスタート時に行われます。

原則として、論理リカバリー可能性を要求する必要があります。例えば、区画内一時データを含むリソースのセットに関連する変更を加え、すべての変更をコミット (またはバックアウト) する場合は、論理リカバリーを要求します。

### 物理リカバリー

物理リカバリー可能性は一時データに固有であり、ウォーム・リスタート時および緊急リスタート時の両方で有効です。区画内キュー定義で物理リカバリーを要求することで、1つの例外を除き、キューへの変更が即時にコミットされ、バックアウトされないようにすることができます。

その例外とは、作業単位が失敗する前の、物理的にリカバリー可能なキューからの最後の読み取りです。CICS は、常に、物理的にリカバリー可能な一時データ・キューからの最後の読み取りをバックアウトします。CICS が TD キューに保持する読み取りおよび書き込みポインターの場合、これは読み取りポインターがリセットされ、書き込みポインターは変更されないことを意味します。これを [468 ページの図 86](#) の図に示します。この例での一連の TD アクションおよびその後のリカバリーは、以下のように行われます。

- 作業単位は項目 1 および 2 を読み取り、読み取りポインターは項目 3 に残っています。
- 作業単位は項目 4 を書き込み、書き込みポインターは次の項目を書き込むことができる状態になっています。
- CICS が異常終了し、緊急リスタートによりリスタートします。
- 一時データ・リカバリーの結果、読み取りポインターは項目 2 にリセットされ、キュー項目 2、3、および 4 はまだ使用可能であり、書き込みポインターが復元されます。



物理的にリカバリー可能な TD キュー (障害前)



物理的にリカバリー可能な TD キューの状態 (緊急時再始動後)



図 86. 物理的にリカバリー可能な TD キューのリカバリーの図

区画内 TD を物理的にリカバリー可能にすると、一部の CICS キューで役立つ場合があります。例えば、CICS での障害発生後、CICS をできる限り迅速にリスタートし、その後、障害の原因を調べることを選択することがあります。CSMT などのキューを区画内および物理リカバリー可能として指定することで、障害の直前に作成されたメッセージをリカバリーすることができ、これを問題の診断に役立てることができま

### リカバリーされない

区画内一時データ定義のリカバリー属性で NO を指定した場合、リカバリーは実行されません。

### 順方向リカバリー

CICS では、一時データの順方向リカバリー・サポートを提供しません。

区画内一時データの順方向リカバリーが必要な場合は、CICS の実行中に区画内一時データ・キューの内容に加えられた変更を記録するアプリケーション・プログラムを指定します。変更はユーザー・ジャーナルに記録されます。ジャーナル処理される情報には、以下が含まれている必要があります。

- 書き込まれるデータを含む各 WRITEQ
- 各 READQ
- キューの各 DELETEQ
- 論理的にリカバリー可能なキューごとの各バックアウト、同期点、または同期点ロールバック

ジャーナル処理された情報を読み取り、その情報を一時データ・キューに適用して、データを再構築するアプリケーション・プログラムを指定する必要があります。アプリケーション・プログラムは、プログラム・リスト・テーブル (PLT) フェーズ中または緊急リスタート後に実行できます。データ・セットが完全にリカバリーされるまでは、キューに書き込まないでください。これは、そのようにすると、データが誤った順序になることや、読み取りで有効なデータ (またはすべてのデータ) が提供されなくなることがあるためです。これらの理由から、リカバリー・プログラムをリスタートの後に実行するよりも、PLT フェーズ中に実行するほうが良い場合があります。

このようなリカバリー戦略なしに、壊れた区画内データ・セットでコールド・スタートを実行した場合は、区画内データ・セットの内容が失われます。TDINTRA=EMPTY をシステム初期設定パラメーターとして指定した場合も、区画内データ・セットの内容が失われます。

### 区画外一時データのリカバリー

CICS では、区画外データ・セットのリカバリーは行いません。区画外データに依存している場合は、CICS の制御または非制御いずれかのシャットダウンの後にリスタートするとき、実行が継続されるようにするためのデータのリカバリー手順を作成する必要があります。

区画外データ・セットのリカバリーでは、以下の 2 つの領域を考慮します。

- 入力区画外データ・セット
- 出力区画外データ・セット

#### 入力区画外データ・セット

リスタート時に必要となる主な情報は、システムの終了時まで処理されたレコードの数です。これは、以下の段落の説明に従い、CICS ジャーナリングを使用して、処理中に記録できます。

区画外入力キューからレコードを読み取る各アプリケーション・プログラムは、最初にこれらのキューへの排他アクセスをエンキューする必要があります。これにより、同時実行されている他のタスクによる同じキューへのインターリーブ・アクセスが防止されます。

次に、アプリケーション・プログラムは、READQ TD コマンドを発行して区画外入力レコードを読み取って処理します。この方法で、各キューの実行中に読み取られ、処理された入力レコードの合計数が累積されます。READQ 操作の合計数は、関連する宛先 ID と共にジャーナル・データ・セットに書き込まれます。このジャーナリングは、RETURN または SYNCPOINT コマンドの直前に実行する必要があります。

ジャーナル・レコードの出力後、各アプリケーション・プログラムは、それ自体を区画外入力キューからデキューして、他のアプリケーション・プログラムがそれらのキューにアクセスできるようにします。

このジャーナリングの前に非制御のシャットダウンが発生した場合、その作業単位のジャーナル・データ・セットにはレコードが含まれません。このため、この未完了タスクの結果は、緊急リスタート時に自動的にバックアウトされます。ただし、非制御のシャットダウン前にジャーナル・レコードが書き込まれた場合、この完了している入力データ・セット処理は緊急リスタート時に認識されます。

非制御のシャットダウン後の緊急リスタート時、または制御されたシャットダウン後のウォーム・スタート時に、前回の CICS 操作中のレコードの入力および処理を反映するように区画外入力データ・セットの位置を変更する以下の手順を使用します。

初期設定フェーズで実行する区画外入力リカバリー・プログラムを PLT で指定できます。このプログラムは、ジャーナル・データ・セットを順方向に読み取ります。ジャーナル処理された各レコードは、アプリケーション・プログラムの前回の実行中に関連する区画外入力データ・セットに対して実行された READQ 操作の数を示します。リカバリー・プログラムによって、同じ数の READQ TD コマンドが、以前に参照されたものと同じ入力キューに対して再度発行されます。

ジャーナル・データ・セットの末尾に達すると、区画外入力データ・セットは、非制御のシャットダウン時に未完了だったタスクの開始前に到達していたポイントと同じポイントに配置されます。その結果、未完了タスク・アクティビティーを含むこれらの入力データ・セットの論理リカバリーがバックアウトされます。

### 出力区画外データ・セット

出力区画外データ・セットのリカバリーは、入力データ・セットのリカバリーとは若干異なります。

テープ出力データ・セットでは、リスタート時に新しい出力テープを使用します。このようにすると、終了前に記録された情報をリカバリーする必要がある場合に、以前の出力テープを使用できます。

終了時にテープ出力バッファのデータが失われないようにするために、非ブロック化レコードを作成できます。または、(ウォーム・スタートまたは緊急リスタート時に CICS によりリカバリーされた) 区画内ディスク宛先にデータを書き込み、定期的に、自動開始タスクを使用して、このデータを区画外テープ宛先にコピーします。終了時には、データはリスタート時に再コピーできる状態のままになります。

CICS の制御されたシャットダウンが発生した場合、以前の出力テープは正常にクローズし、テープ・マークを書き込みます。ただし、電力障害やマシン・チェックなどの非制御のシャットダウンでは、テープの終わりを示すテープ・マークは書き込まれません。

ライン・プリンターの出力データ・セットの場合は、単純に、システムが停止したときに印刷が停止された場所から続行することを選択できます。ただし、ページの先頭など、定義されたポイントから出力を続行するには、ジャーナル・データ・セットを使用することが必要になる場合があります。各ページが通常の CICS 操作中に完了するたびに、ジャーナル・データ・セットにレコードが書き込まれます。

リスタート時に、障害の発生時に処理されていたページをジャーナル・データ・セットから特定でき、そのページを再処理して同じ出力を再作成できます。または、(前に説明した) テープ出力バッファの中間区画内宛先を使用します。

### 事後初期設定 (PLTPI) プログラムの使用

区画外一時データをリカバリーするため、またはリカバリー中に必要な出口を使用可能にするために必要な処理の一環として、初期設定 (PLTPI) プログラムを使用できます。

### このタスクについて

PLT フェーズには 2 つあります。最初のフェーズはシステム 初期設定タスクが接続される前に行われ、初期設定が完了していないため、CICS リソースは使用できません。最初のフェーズの目的は、リカバリー処理中に必要な出口を使用可能にするためのみです。2 番目のフェーズは CICS 初期設定の完了後に行われ、この時点で、PLT プログラムを使用して環境をカスタマイズできます。

PLT のコーディング方法については、を参照してください。PLT プログラムに適用される特殊な条件に関するプログラミング情報については、を参照してください。

## 一時記憶域のリカバリー

このセクションでは、一時記憶域の逆方向と順方向の両方のリカバリーについて説明します。

## 逆方向リカバリー

CICS でリカバリー可能にする一時記憶域キューは、補助一時記憶域上に存在する必要があります。

以下の定義ステートメントの例に示す一時記憶域モデル・リソース定義を使用して、一時記憶域キューをリカバリー可能として定義します。

```
CEDA DEFINE DESCRIPTION(Recoverable TS queues for START requests) TSMODEL(RECOV1)
          GROUP(TSRECOV) PREFIX(DF) LOCATION(AUXILIARY) RECOVERY(YES)

CEDA DEFINE DESCRIPTION(Recoverable TS queues for BMS) TSMODEL(RECOV2)
          GROUP(TSRECOV) PREFIX(**) LOCATION(AUXILIARY) RECOVERY(YES)

CEDA DEFINE DESCRIPTION(Recoverable TS queues for BMS) TSMODEL(RECOV3)
          GROUP(TSRECOV) PREFIX($$) LOCATION(AUXILIARY) RECOVERY(YES)
```

BMS については、この例は、デフォルトの一時記憶域キュー接頭部に必要なリソース定義を示しています。アプリケーション・プログラマーは、以下のオペランドを指定して、BMS のデフォルトの一時記憶域キュー接頭部をオーバーライドできます。

- **SEND MAP** コマンドの REQID オペランド
- **SEND TEXT** コマンドの REQID オペランド
- **ROUTE** コマンドの REQID オペランド
- **CMSG** トランザクションの PROTECT オペランド

アプリケーション・プログラマーが使用するすべての代替一時記憶域キューの接頭部をリカバリー可能として定義する必要があります。

割り振りおよびスペース所要量について詳しくは、[一時記憶域データ・セットのセットアップ](#)を参照してください。

## 順方向リカバリー

一時記憶域の順方向リカバリーが必要な場合は、現在の CICS 実行中に一時記憶域に加えられた変更を記録するアプリケーション・プログラムを指定する必要があります。

1. 緊急リスタート時に、アプリケーション・プログラムでは (PLTPI などを使用して) 緊急リスタートを遅延させ、再度アプリケーション・プログラムを使用し、前回読み取られたレコードを使用して、可能な限り多くの一時記憶域データを再構築することができます。
2. 再度緊急リスタートを実行しますが、今回は、一時記憶域のコールド・スタートを実行するように変更された (TS=(COLD)) システム初期設定パラメーターを使用します。ただし、このことによって一時記憶域データ・セット全体の内容が失われることに注意してください。

## Web サービスのリカバリー

Web サービスによって WebSphere MQ トランSPORTおよび永続メッセージが使用される場合は、BTS を使用して、CICS システム障害が発生した場合にメッセージが確実にリカバリーされるようにすることができます。

### 永続メッセージをサポートするための CICS の構成

CICS は、IBM MQ トランSPORT・プロトコルを使用した永続メッセージの、CICS 領域に配置された Web サービス・プロバイダー・アプリケーションへの送信をサポートしています。

### このタスクについて

CICS は、ビジネス・トランザクション・サービス (BTS) を使用して、CICS システム障害の際に永続メッセージが確実に回復されるようにします。これを正しく機能させるためには、以下のステップに従います。

### 手順

1. IDCAMS (データ操作ユーティリティー) を使用して、MVS へのローカルの要求キューとリポジトリ・ファイルを定義します。  
ファイル定義のために、STRINGS に適切な値を指定する必要があります。デフォルト値 1 では十分ではないと思われるため、10 を使用することをお勧めします。

2. CICS に対するローカルの要求キューとリポジトリ・ファイルを定義します。

CICS に対するローカルの要求キューを定義する方法の詳細は、472 ページの『サービス・プロバイダーでのローカル・キューの定義』で説明しています。ファイル定義に、STRINGS に適切な値を指定する必要があります。デフォルト値 1 では十分ではないと思われるため、10 を使用することをお勧めします。

3. リポジトリ・ファイル名を FILE オプションの値として使用して、DFHMQSOA という名前の PROCESSTYPE リソースを定義します。
4. 永続メッセージの処理中に、最初の暗黙的な同期点が要求される前にプログラムが **EXEC CICS SYNCPOINT** コマンドを必ず発行することを確認してください。例えば **EXEC CICS CREATE TDQUEUE** のような SPI コマンドを使用すると、暗黙的に同期点が取られます。

**EXEC CICS SYNCPOINT** コマンドを発行すると、永続メッセージが正常に処理されたことを確認できます。プログラムが暗黙的に同期点を取るを試みる前に明示的に同期点を要求しない場合、ASP7 異常終了が発行されます。

## 次のタスク

片方向の要求メッセージの場合は、Web サービスが異常終了またはバックアウトすると、トランザクションまたはプログラムが障害が発生している要求を再試行したり障害を適切に報告したりするための十分な情報が保持されます。このようなりカバリー・トランザクションまたはプログラムを提供する必要があります。詳しくは、473 ページの『永続メッセージの処理』を参照してください。

## サービス・プロバイダーでのローカル・キューの定義

サービス・プロバイダーで IBM MQ トランスポートを使用する場合は、要求メッセージを処理するまで要求メッセージを保管する 1 つ以上のローカル・キューと、要求メッセージを処理する CICS トランザクションを指定する 1 つのトリガー・プロセスを定義する必要があります。

## 手順

1. 開始キューを定義します。  
以下のコマンドを使用します。

```
DEFINE
QLOCAL('initiation_queue')
DESCR('description')
```

ここで、*initiation\_queue* は、CICS 領域のインストール済み MQMONITOR リソース定義の QNAME 属性に指定されている値と同じか、インストール済み MQCONN リソース定義の INITQNAME 属性に指定されている値と同じです。

2. ローカル要求キューごとに、QLOCAL オブジェクトを定義します。  
以下のコマンドを使用します。

```
DEFINE
QLOCAL('queueename')
DESCR('description')
PROCESS(processname)
INITQ('initiation_queue')
TRIGGER
TRIGTYPE(FIRST)
TRIGDATA('default_target_service')
BOTHRESH(nnn)
BOQNAME('requeueename')
```

ここで、

- *queueename* は、ローカル・キュー名です。

- *processname* は、トリガー・イベント発生時にキュー・マネージャーによって開始されるアプリケーションを示すプロセス・インスタンスの名前です。各 QLOCAL オブジェクトにも、同じ名前を指定します。
  - *initiation\_queue* は、使用される開始キューの名前です。例えば、CICS 領域のインストール済み MQMONITOR リソース定義の QNAME 属性で指定されている開始キューです。
  - *default\_target\_service* は、要求にサービスが指定されていない場合、使用されるデフォルトのターゲット・サービスです。ターゲット・サービスの形式は「/string」で、これは URIMAP 定義のパスと突き合わせるために使用されます (例えば /SOAP/test/test1)。先頭文字は必ず「/」にする必要があります。
  - *nnn* は、行われる再試行の回数です。
  - *requeuename* は、障害発生メッセージの送信先キューの名前です。
3. トリガー・プロセスを指定する PROCESS オブジェクトを定義します。  
以下のコマンドを使用します。

```
DEFINE
PROCESS(processname)
APPLTYPE(CICS)
APPLICID(CPIL)
```

ここで、

*processname* は、プロセスの名前で、要求キューの定義時に使用される名前と同じにする必要があります。

### 永続メッセージの処理

Web サービス要求が IBM MQ 永続メッセージで受信されると、CICS は、プロセス・タイプが DFHMQSOA である固有の BTS プロセスを作成します。インバウンド要求に関連するデータは、プロセスに関連付けられた BTS データ・コンテナ内に取り込まれます。

プロセスは、非同期で実行されるようにスケジュールに入れられます。Web サービスが正常に完了してコミットすると、CICS は BTS プロセスを削除します。これには、SOAP 障害が Web サービス・リクエスターに生成され戻される場合が含まれます。

### エラー処理

必要な BTS プロセスの作成時にエラーが発生すると、Web サービス・トランザクションは異常終了し、インバウンド Web サービス要求は処理されません。BTS が使用不可の場合は、メッセージ DFHPI0117 が発行され、CICS は、既存のチャンネル・ベースのコンテナ・メカニズムを使用して、BTS なしで続行します。

Web サービスが開始または処理を完了する前に CICS 障害が発生すると、BTS リカバリーにより、CICS の再起動時にプロセスのスケジュールが変更されます。

Web サービスが異常終了してバックアウトすると、BTS プロセスには、ABENDED 状態で完了したというマークが付きます。応答を必要とする要求メッセージの場合は、SOAP 障害が Web サービス・リクエスターに戻されます。BTS プロセスは取り消され、CICS は、失敗した要求に関する情報を保持しません。CICS は、一時データ・キュー CSBA ではメッセージ DFHBA0104 を発行し、一時データ・キュー CPIO ではメッセージ DFHPI0117 を発行します。

片方向メッセージの場合、障害に関する情報をリクエスターに戻す方法はないため、BTS プロセスは COMPLETE ABENDED 状態を保ちます。CICS は、一時データ・キュー CSBA ではメッセージ DFHBA0104 を発行し、一時データ・キュー CPIO では DFHPI0116 を発行します。

CBAM トランザクションを使用して COMPLETE ABENDED プロセスを表示することができます。または、リカバリー・トランザクションを指定して、DFHMQSOA の COMPLETE ABENDED プロセスをチェックし、適切な処置をとることができます。

例えば、リカバリー・トランザクションで以下のことが可能です。

1. **RESET ACQPROCESS** コマンドを使用して BTS プロセスをリセットする。



2. **RUN ASYNC** コマンドを発行して、障害がある Web サービスを再試行する。プロセスでの別のデータ・コンテナに再試行カウントを保持して、障害が繰り返されるのを回避することができます。
3. 関連する以下のデータ・コンテナ内の情報を使用して、問題を報告する。

DFHMQORIGINALMSG データ・コンテナには、IBM MQ から受信したメッセージが含まれ、これには RFH2 ヘッダーが含まれている場合があります。

DFHMQMSG データ・コンテナには、RFH2 ヘッダーが除去された IBM MQ メッセージが含まれます。

DFHMQDLQ データ・コンテナには、元のメッセージに関連付けられた送達不能キューの名前が含まれます。

DFHMQCONT データ・コンテナには、元のメッセージの **MQ GET** に関連する IBM MQ MQMD 制御ブロックが含まれます。

## プログラム・エラー・プログラム (PEP) の使用

プログラム・エラー・プログラム (PEP) は、すべてのプログラム・レベルの ABEND 出口コードの実行後、および動的トランザクション・バックアウトの実行後に、制御を取得します。

### このタスクについて

すべての領域に対し、プログラム・エラー・プログラムは 1 つのみです。

### 手順

1. CICS 提供のプログラム・エラー・プログラムである DFHPEP を使用するか、自分で作成するかを決定します。  
CICS 提供の DFHPEP プログラムでは何の機能も実行されませんが、トランザクションの異常終了後にインストール・レベルのアクションを実行するための独自のコードをこのプログラムに含めることができます (474 ページの『[CICS 提供の PEP](#)』を参照)。
2. 独自の PEP を作成する場合は、CICS 提供のバージョンを変更します。  
デフォルトの DFHPEP は CICSTS56.CICS.SDFHLOAD にあり、CSD の CICS DFHMISC グループ で定義されています。  
PEP の作成方法について詳しくは、[プログラム・エラー・プログラムの作成](#)に記載のガイドをお読みください。
3. DFHLIST グループ・リストを含めるように **GRPLIST** システム 初期設定パラメーターを更新します。  
これには、DFHMISC グループの定義が含まれます。

### CICS 提供の PEP

CICS 提供の PEP は処理を行いません。その唯一の効果は、CICS からリンクされた場合に、そうでない場合には出される DFHAC2259 メッセージが出されないようにすることです。

CICS の機能はすべて DFHPEP プログラムで使用できます。例えば、以下の作業を実行できます。

- 端末へのメッセージの送信
- マスター端末へのメッセージの送信
- 異常終了に関する情報または統計の記録
- このタスクに関連付けられているトランザクション・エントリーの使用不可化要求

ただし、DFHPEP プログラムには以下の処理が適用されます。

1. タスクの異常終了が CICS による処理の一部である場合は、システムの停止を避けるために、DFHPEP に制御は与えられません。
2. 異常終了したトランザクションが DFHREST によって再始動されることをトランザクション・マネージャーが検出した場合は、DFHPEP に制御は与えられません。
3. DFHPEP 処理は、トランザクション・ダンプの実行後に行われます。DFHPEP はダンプが発生しないようにすることはできません。
4. 同期点の処理中にトランザクション障害が発生した場合、DFHPEP に制御は与えられません。

5. タスクの終了を引き起こした条件が CICS の異常条件プログラム (ACP) によって処理される場合、DFHPEP に制御は与えられません。ACP によって処理される条件は、何らかの接続障害です。例えば、トランザクションが存在しない場合や、セキュリティ違反が検出された場合などです。
6. タスクが異常終了し、CICS がストレージ不足の場合は、DFHPEP に制御は与えられません。
7. CICS トランザクション障害プログラム DFHTFP は、トランザクション・バックアウトが実行される前に DFHPEP にリンクします。これは、異常終了トランザクションによって使用されるリソースが解放されていない可能性があることを意味します。DFHPEP はこのことを認識している必要があります。また、まだロックされているリソースを処理するためのロジックが必要になる可能性があります。
8. 基本機能が APPC リンクである分散トランザクションの再始動機能は使用しないでください。一部のエラー状態では、CICS が APPC 会話状態を解決できず、トランザクションはコード AZCP で異常終了します。

### 独自の PEP

CICS の運用の早期のフェーズで、異常終了の原因を調査および修正している間、異常終了トランザクションを使用不可状態にすることができます。

プログラムでこのプロセスを処理する必要がある場合や、関連するプログラムまたはトランザクションも使用不可にする必要がある場合は、これらのアクションを独自の PEP に組み込むことをお勧めします。

プログラム・エラー・プログラムはコマンド・レベルのプログラムで、CICS でサポートされる任意の言語で作成できます。CICS 異常条件プログラムは、異常終了に関する情報を含むプログラム・エラー・プログラムに通信域 (COMMAREA) を渡します。

プログラム・エラー・プログラムには、以下のようなプログラムを組み込むことを検討できます。

- 特定のトランザクション ID を (他のユーザーが使用できないように) 使用不可にして、診断および修正アクションを保留にする。手操作による介入や、それ以降に複数の異常終了が立て続けに発生するリスクを回避できます。
- 特定のプログラムの正常な動作に依存する他のトランザクションまたはプログラムを使用不可にする。
- 機能タイプ (トランザクションまたはファイル) 別にエラーの数を計測する。
- トランザクションの異常終了後に CICS を異常終了させる。このことが適する条件は以下のとおりです。
  - 異常終了したトランザクションが重要なファイルを処理していた場合
  - 異常終了したトランザクションがシステムの動作にとって重要であった場合
  - 異常終了の原因が、アプリケーションの使用を継続する意味がないか、継続するとデータ安全性が脅かされる可能性があったためである場合
  - 特定の機能タイプ (トランザクションまたはファイル) のエラー件数が事前定義されたレベルに達した場合
- 機能を使用不可にする。これにより、CICS を異常終了させた場合よりも長くシステムの実行を継続できます。

タスクが異常終了した場合、プログラム・レベルの出口または PEP でコーディングすることで、インストールされているトランザクション定義の該当するトランザクション・コード・エントリーに使用不可のフラグを設定できます。CICS は、そのトランザクション・コードが再度使用可能になるまで、それ以降の端末またはプログラムによるそのコードの使用の試みを拒否します。その結果、問題のあるトランザクション・コードが使用されるたびにプログラム・チェックが実行されることがなくなり、プログラム・チェックの影響を最小化できます。最初のプログラム・チェックのみが処理されます。インストールされているトランザクション定義が使用不可にされることが PEP に示された場合、CICS ではそれ以降、そのトランザクション・コードの使用を受け入れなくなります。

エラーを修正したら、関連するインストール済みのトランザクション定義を再度使用可能にして、端末でできるようにすることができます。アプリケーション依存の理由でトランザクションを受け入れないようにする場合、トランザクション ID を使用不可にして、後で再度使用可能にすることもできます。

DFHPEP 内のロジックによって CICS の実行を継続することが安全でないと判断された場合は、オペレーティング・システムの ABEND マクロを発行して、CICS を強制的に異常終了させることができます。DFHPEP が異常終了した場合 (トランザクションの異常終了)、CICS によってメッセージ DFHAC2263 が生成されます。

### **PEP の省略**

CICS 提供の PEP は、CICSTS56.CICS.SDFHLOAD ライブラリーにあります。ただし、CICS の異常条件プログラムがここにリンクされるのは、DFHPEP のプログラム・リソース定義がインストールされている場合のみとなります。CICS では、このことや別の理由で DFHPEP にリンクできない場合は、DFHAC2259 メッセージを CSMT に送信します。

### **このタスクについて**

## 第 16 章 REXX の構成

このセクションでは、REXX を構成して管理する方法の概要を示します。

### REXX サポートの構成

REXX プログラムを実行するには、その前に REXX サポートを構成する必要があります。

以下のステップを使用します。

1. RFS ファイル・プールを作成します。
2. リソース定義を作成します。
3. LSRPOOL 定義を確認します。
4. CICSTART メンバーを更新します。
5. CICS 初期設定 JCL を変更します。
6. インストールを検証します。
7. RFS ファイル・プールを形式設定します。
8. ヘルプ・ファイルを作成します。
9. REXX Db2 インターフェースを構成します。

### RFS ファイル・プールの作成

REXX ファイルリング・システム (RFS) では、データを保管するために複数のファイル・プールが使用されます。

これらは VSAM クラスターのセットとして実装されます。CICSTS56.REXX.SCICJCL 内で提供されているメンバー CICVSAM を使って、2 つの RFS ファイル・プールのための VSAM データ・セットを作成します。必要に応じて、お客様独自の環境に合うようにこのジョブを変更します。例えば、ご使用のインストール済み環境の標準指針に合わせてデータ・セット名を変更することができます。変更が完了したら、ジョブを実行してファイル・プールを定義します。

リソース定義でファイル・プール名を変更したりファイル・プール・コンポーネントを追加したりする場合は、CICSTART メンバー内のファイル・プール定義に、対応する変更を加える必要があります。477 ページの『リソース定義の作成』および 478 ページの『CICSTART メンバーの更新』を参照してください。

以下の定義が整合していることを確認してください。

- CICSTART 内のファイル・プール定義
- CICRDOD (Development System の場合) または CICRDOR (Runtime Facility の場合) 中のリソース定義
- CICVSAM 内の VSAM クラスター定義

### リソース定義の作成

REXX のプロファイル、プログラム、トランザクション、およびファイル・リソースの定義は、提供された CSD またはアップグレードされた CSD 内のグループ CICREXX に含まれています。

CICSTS56.REXX.SCICJCL データ・セット内の CICRDOR ジョブ (Runtime Facility の場合) または CICRDOD ジョブ (Development System の場合) により、REXX/CICS プロファイル、VSAM ファイル、プログラム、トランザクション、および一時データ・キューを含む、製品で必要とされるエントリーを追加します。

一時データ・キューは、REXX/CICS IMPORT および EXPORT コマンドで使用されます。このジョブには、Db2 プランに対してトランザクションを許可する、REXX/CICS SQL インターフェースのための定義も含まれています。

JCL を編集し (JCL の最初のコメント内で説明されているように、エントリーのコメント解除してください)、ジョブを実行します。

## IMPORT および EXPORT コマンドの TD キューの変更

REXX/CICS 環境では、区分データ・セットからメンバーを IMPORT する際、または区分データ・セットに RFS ファイルを EXPORT する際に、動的割り振りを使用します。

CICSTS56.REXX.SCICJCL データ・セット内のメンバー CICRDOD または CICRDOR を使って、IMPORT の入力として使用する一時データのエントリーを3つと、EXPORT の出力として使用する一時データのエントリーを3つ定義します。これにより、区分データ・セットに対して、3 ユーザーによる同時の IMPORT と、3 ユーザーによる同時の EXPORT を可能にします。

ユーザーの要件に合うように TDQ エントリーの数を変更します。しかし、入力用エントリーと出力用エントリーが少なくとも1つずつは必要です。TDQUEUE NAME は REX で始まらなければならず、その接尾部は有効な文字でなければなりません。REX で始まる TDQUEUE 名を使用する他のアプリケーションを使用しないでください。これは、IMPORT および EXPORT がこの名前を使用するため、ファイルが破壊されてしまう場合があるためです。

## LSRPOOL 定義の確認

RFS ファイルの最大キー長は 252 であり、提供される VSAM 定義では制御インターバル・サイズとして 18K が使用されるため、RFS に使用される LSRPOOL がこれらの値をサポートしていることを確認することが重要です。

これを実行しないと、システム・コンソールに OPEN エラーが表示される可能性があります。詳しくは、[LSRPOOL リソースおよびローカル共用リソース \(LSR\) または非共用リソース \(NSR\)](#)を参照してください。

## CICSTART メンバーの更新

hlq.CICSTS56.REXX.SCICEXEC データ・セット内のメンバー CICSTART には、REXX/CICS 環境のデフォルト定義が含まれています。CICSTART は、CICS システムが始動した後、CICS/REXX プログラムを使用する最初のトランザクションが出される時に実行されます。

CICSTART は、環境の永続的な構成を保持します。提供されている CICSTART メンバーを編集するか、またはそのメンバーを独自の CICSTART exec のモデルとして使用してください。

- 初期設定

疑似会話型モードをオフに設定する (PSEUDO OFF) 初期設定ステートメントと、RLS ディレクトリーを検査する (RLS CKDIR) 初期設定ステートメントを、提供されている CICSTART メンバーの開始時に示されるとおりに使用します。

- RFS ファイル・プールを定義する

提供されている CICSTART メンバーは、2つのファイル・プールを定義します。

```
'FILEPOOL DEFINE POOL1 RFSDIR1 RFSP00L1 (USER'  
  IF RC = 0 THEN EXIT RC  
'FILEPOOL DEFINE POOL2 RFSDIR2 RFSP00L2 (USER'  
  IF RC = 0 THEN EXIT RC
```

477 ページの『[RFS ファイル・プールの作成](#)』のステップでファイル・プール名を変更したり、ファイル・プールを追加したりした場合は、CICSTART exec 内のファイル・プール定義に、対応する変更を加える必要があります。

- ユーザー ID に権限を付与する

提供されている CICSTART メンバーは、ユーザー ID RCUSER に権限を付与します。その権限を必要とする組織内のユーザーに権限を付与するコマンドが CICSTART exec に含まれていることを確認してください。

ファイル・プールをフォーマット設定するための権限を必要とするすべてのユーザーを含めてください。そうしないと、FILEPOOL FORMAT コマンドが戻りコード -4 で失敗します。

CICS セキュリティーがオフになっている領域 (つまり、SIT パラメーター SEC=NO) については、CICS デフォルト・ユーザー ID のために AUTHUSER コマンドを CICSTART に組み込みます。例えば、領域のデフォルト・ユーザー ID が SYSA である場合、次のコマンドを組み込みます。

```
'AUTHUSER SYSA'
```

- 必要に応じて CICS トランザクション ID を REXX exec に関連付ける

CICSTART には、CICS DEFINE TRANSACTION コマンドによって定義された CICS トランザクション ID を呼び出し対象の REXX exec に関連付けるために、DEFTRNID ステートメントを含める必要があります。提供されている CICSTART メンバーは、REXX、EDIT、および FLST の各トランザクションをそれぞれ、CICRXTRY exec、CICEDIT exec、および CICFLST exec に関連付けます。これらの CICS トランザクションは、CICRDOR および CICRDOD の JCL ファイルで定義されています。[477 ページの『リソース定義の作成』](#)を参照してください。

ご使用の CICSTART exec には、REXX/CICS トランザクションを実行するように定義するすべての CICS トランザクションのために、DEFTRNID ステートメントを含める必要があります。

- REXX exec を実行するためのデフォルトを設定する

ご使用の環境に必要な SETSYS コマンドを組み込みます。例えば、インストール済み環境で REXX/CICS exec を実行するための言語、検索 PF キー、および疑似会話型設定を設定できます。

デフォルトでは、疑似会話型モードはオンです (ただし、CICSTART exec 自体は会話型モードで実行する必要があります)。疑似会話型モードについて詳しくは、[PSEUDO](#) および [ユーザーに表示するダイアログの処理](#)を参照してください。

- ヘルプ・パスを定義する

提供されている CICSTART メンバーでは、現行のヘルプ・ユーティリティのヘルプ・パス HELPPTH2 と、互換性のために以前のヘルプ機能のヘルプ・パス HELPPTH が定義されています。

```
HELPPATH = 'POOL2:\BOOK'
'RLS VARPUR HELPPATH \SYSTEM\DEFAULTS'
IF RC = 0 THEN EXIT
HELPPTH2 = 'POOL2:\HELP'
'RLS VARPUR HELPPTH2 \SYSTEM\DEFAULTS'
IF RC = 0 THEN EXIT RC
```

以前のヘルプ機能がインストールされていない新しいインストール済み環境の場合、HELPPTH を定義する 3 つのステートメントを削除できます。

- 必要に応じて exec をプリロードするための EXECLOAD を使用する

exec をプリロードするための EXECLOAD ステートメントを組み込むことができます。exec をプリロードすると、ユーザーごとに exec をロードする必要がないためパフォーマンスが向上します。また、すべての同時ユーザーが同じコピーを共用できるため、必要なストレージが少なくなります。

提供されている CICSTART メンバーで EXECLOAD コマンドのリストを使用するか、必要な exec のために EXECLOAD コマンドを組み込むことができます。

提供されている CICSTART メンバーには、CICEDIT exec、CICESVR exec、および CICEPROF exec 用の EXECLOAD コマンドが組み込まれています。これらの exec は、REXX/CICS テキスト・エディターのコンポーネントです。このエディターは REXX Development System のみで提供され、独自の REXX プログラムを開発するときに使用することができます。ご使用のインストール済み環境で REXX/CICS テキスト・エディターを使用している場合、これらの EXECLOAD コマンドを含めることができます。

詳しくは、[EXECLOAD](#) を参照してください。

## CICS 初期設定 JCL の変更

DD ステートメントを CICS 始動ジョブおよび DFHRPL 連結に追加します。

CICS 始動ジョブに、次の DD ステートメントを追加します。

```
//CICAUTH DD DSN=CICSTS56.REXX.SCICMDS,DISP=SHR
//CICEXEC DD DSN=CICSTS56.REXX.SCICEXEC,DISP=SHR
//CICUSER DD DSN=CICSTS56.REXX.SCICUSER,DISP=SHR
```

REXX データ・セットの DD ステートメントを DFHRPL 連結に追加します。

```
//DFHRPL DD DSN=CICSTS56.REXX.SCICLOAD,DISP=SHR
```



REXX/CICS 環境では、CICS でリソース定義を持たない、CICCMDS、CICEXEC、および CICUSER という 3 つのデータ・セット連結の DD 名を使用します。これらのデータ・セットは区分データ・セットで、MVS 機能を使用してアクセスします。

### CICCMDS

CICCMDS の DD 名連結は、CICSTS56.REXX.SCICCMDS データ・セットの参照で始まります。このデータ・セットには、REXX/CICS 環境の許可コマンドをインプリメントする exec が含まれています。許可ユーザー、または許可コマンドの使用を許可された exec のみが、これらの exec にアクセスできます。独自の許可コマンドを使用して REXX/CICS 環境を拡張する場合は、データ・セットをこの DD 名連結に連結します。

### CICEXEC

CICEXEC の DD 名連結は、CICSTS56.REXX.SCICEXEC データ・セットの参照で始まります。このデータ・セットには、REXX/CICS 環境によって提供されている、許可コマンドを使用する exec が含まれています。許可コマンドを使用する独自の exec を作成して REXX/CICS 環境を拡張する場合は、その exec が入っているデータ・セットをこの DD 名に連結します。

### CICUSER

CICUSER の DD 名連結は、CICSTS56.REXX.SCICUSER データ・セットの参照で始まります。このデータ・セットには、REXX/CICS 環境によって提供されている、許可コマンドを使用しない exec が含まれています。許可コマンドを使用しない独自の exec を作成して REXX/CICS 環境を拡張する場合は、その exec が入っているデータ・セットをこの DD 名に連結します。

これらのデータ・セット連結にアクセスするために使用する機能では、CICS 領域が待ち状態になるのを避けるために CICS WAIT EXTERNAL の機能を使用します。

## RFS ファイル・プールの形式設定

CICSTART で定義されているすべての REXX ファイリング・システム (RFS) ファイル・プールの形式設定をする必要があります。

1. 必要なすべての構成タスクが完了していることを確認し、必要に応じて CICS を再始動します。
2. CICSTART で許可ユーザーとして定義されたユーザー ID を使用してサインオンします。
3. REXX と入力します (これは、CICRXTY exec に関連したデフォルト・トランザクション ID です)。

このアクションによって、REXX および REXX/CICS コマンドを対話式に実行できる対話環境を提供する REXXTY ユーティリティが呼び出されます。画面上部に以下の行が表示され、右下隅には READ が表示されます。カーソルは左下隅にあります。

Enter a REXX command or EXIT to quit

4. CICSTART で定義されたすべてのファイル・プールに以下のコマンドを入力して、ファイル・プールを使用できるように準備します。

'FILEPOOL FORMAT *poolname*'

*poolname* は、CICSTART exec で指定したファイル・プール名です (デフォルトは POOL1 および POOL2)。

REXX コマンドを区切るために単一引用符または二重引用符を使用することをお勧めします。

対話環境では、画面上で次に使用可能な行に各コマンドがエコー出力され、要求された出力も表示されます。

**FILEPOOL FORMAT** コマンドは、正常に完了したことを示す情報を表示しませんが、画面の右下隅に READ の語が表示されます。

5. **FILEPOOL FORMAT** コマンドが正常に終了したかどうかを調べるには、SAY RC (アポストロフィなし) を入力します。使用可能な次の行に 0 が表示されると、コマンドが成功したことを示しています。

ファイル・プールを再形式設定しようとする、以下のメッセージが表示されます。

Subcommand return code = 1836

6. すべての RFS ファイル・プールが形式設定されるまで、この処理を続けます。ファイル・プールを形式設定するのは、新しいファイル・プールを定義した場合、または既存のファイル・プールのクラスターを削除して再定義した場合のみです。

ファイル・プールを形式設定する際に、または REXX や REXX/CICS のコマンドや命令を対話式に実行している間に、画面がいっぱいになったら、右下隅に MORE 標識が表示されます。画面をクリアするには、Enter キーを押します。いつでも Clear キーを押してデータの画面をクリアできます。対話環境を終了するには、PF3 キーを押します。このアクションは EXIT 命令の入力をシミュレートします。ユーザー自身で EXIT 命令を (アポストロフィなしで) 入力することもできます。

対話環境で RETRIEVE キーを押すと、以前に入力したコマンドが再呼び出しされます。このキーのデフォルトのシステム設定は PF12 です。RETRIEVE キーを押すと、直前に入力した行が入力位置に再表示されます。その後、この領域を変更し、Enter を押して命令を再実行できます。RETRIEVE キーを複数回押すと、前に入力したコマンドが新しい順に入力域に表示されます。

## インストールの検証

インストールが成功したことを検証するために、exec が提供されています。

1. REXX/CICS REXXTRY ユーティリティーで、CALL CICIVP1 を入力して、インストール検査プログラムを実行します。exec からの出力は以下のようになります (画面に MORE と表示されたら、必ず Enter キーを押してください)。

```
EXEC CICIVP1
***-----***
*** This is a test REXX program running under CICS-TS ***
*** It was loaded from CICUSER-CICIVP1 ***
***-----***

What is your name?
```

2. 名前を入力し、Enter キーを押します。出力は以下のようになります。

```
Welcome to REXX/CICS for CICS-TS , xxxx

Invoking nested exec CICIVP2 (which has tracing on)
  20 ** say 'You entered CICIVP2 exec'
    >>> "You entered CICIVP2 exec"
You entered CICIVP2 exec
  21 ** call CICIVP3
You entered CICIVP3 exec which has tracing off
  22 ** exit
Back to CICIVP1 exec

      This is fullscreen output to terminal xxxx
      Now input some data and press ENTER or a PF key

The AID key that was pressed = ENTER
The cursor was at (Row Col): 24 7
The data that was entered (Row Col Data): 24 1 <DATA>
Example of more than one screen
1000 assignment statements have been executed
2000 assignment statements have been executed
3000 assignment statements have been executed
4000 assignment statements have been executed
5000 assignment statements have been executed
6000 assignment statements have been executed
7000 assignment statements have been executed
8000 assignment statements have been executed
9000 assignment statements have been executed
10000 assignment statements have been executed
11000 assignment statements have been executed
12000 assignment statements have been executed
13000 assignment statements have been executed
14000 assignment statements have been executed
15000 assignment statements have been executed
16000 assignment statements have been executed
17000 assignment statements have been executed
18000 assignment statements have been executed
19000 assignment statements have been executed
20000 assignment statements have been executed
Today's date is dd mmm yyyy
The time is hh:mm:ss
REXX/CICS CICIVP1 is now finished
```

## ヘルプ・ファイルの作成

REXX/CICS に含まれているオンライン・ヘルプ・ユーティリティを使用して、REXX/CICS に付属している製品資料を検索して表示できます。

### このタスクについて

インストールの一部として、CICS TS 製品のデリバリー・データ・セットから REXX/CICS ファイル・システムに情報をロードする必要があります。

### 手順

1. REXX ファイリング・システム (RFS) のファイル・プールを定義し、ヘルプ・ユーティリティのヘルプ・パスを設定します。

*hlq.CICSTS56.REXX.SCICEXEC* データ・セットで提供される *CICSTART* メンバーには、RFS ファイル・プールを定義するステートメントと、REXX ヘルプのヘルプ・パスを設定するためのステートメントが含まれています。それらのステートメントを確認し、必要に応じてお客様の要件に合うように変更してください。詳しくは、[478 ページの『CICSTART メンバーの更新』](#)を参照してください。

提供されている *CICSTART* メンバーでは、現行のヘルプ・ユーティリティのヘルプ・パス *HELPPTH2* が定義されています。互換性のために、以前のヘルプ機能のヘルプ・パス *HELPPTH* も定義されています。以前のヘルプ機能がインストールされており、対象となるインストール手順が完了するまで使用する必要がある場合を除き、*HELPPTH* は必要ありません。

2. ステップ [482 ページの『3』](#) でヘルプ情報をロードするために使用するユーザー ID に、関係するデータ・セットを CICS TS 製品デリバリー・ライブラリーから REXX ファイリング・システムにインポートする権限が付与されていることを確認します。

*hlq.CICSTS56.REXX.SCICEXEC* データ・セット内の *CICHLOAD exec* は、提供されている *hlq.CICSTS56.REXX.SCICDOC* データ・セットと *hlq.CICSTS56.REXX.SCICPNL* データ・セットからヘルプ情報をインポートします。その情報を使用して、REXX ヘルプで使用される RFS にファイルが作成されます。

この権限を確認する 1 つの方法は、関係するユーザー ID を高位修飾子として使用して、*hlq.CICSTS56.REXX.SCICDOC* および *hlq.CICSTS56.REXX.SCICPNL* のコピーを作成することができます。

3. REXX/CICS を開始し、EXEC *CICHLOAD* コマンドを発行します。

*CICHLOAD* は、MVS ライブラリーからヘルプ情報とパネル定義をインポートします。以下のメッセージが表示されたら、前のステップで設定したデータ・セット名を指定します。

Please enter the MVS dataset name of the help package library  
Example: *hlq.CICSTS55.REXX.SCICDOC*

Please enter the MVS dataset name of the panels library  
Example: *hlq.CICSTS55.REXX.SCICPNL*

提供されている *CICSTS56.REXX.SCICDOC* データ・セットには、オンライン・ヘルプ情報を入れた *CICRXHLP* メンバーが入っています。*CICHLOAD* プログラムはこれを処理して、オンライン・ヘルプ・ユーティリティで使用されるファイルを作成します。ファイルは、指定したヘルプ・パス *HELPPTH2* に作成されます。

このデータ・セットにある *CICR3270* または *CICR3820* メンバーは、*CICRXHLP* メンバーと、[CICS Transaction Server for z/OS 製品情報](#)に付属の PDF に置き換えられます。

[CICS Transaction Server for z/OS 製品情報の最新リリース](#)に付属している REXX for CICS Transaction Server 製品資料を参照して、オンライン製品資料より新しい更新情報が含まれていないか確認することをお勧めします。

## REXX Db2 インターフェースの構成

このステップが必要となるのは、REXX EXECSQL コマンド環境を Db2 サポートのために有効にしている場合だけです。このステップを実行するには、その前に Db2 を完全にインストールしておく必要があります。

CICSTS56.REXX.SCICJCL データ・セットのメンバー CICRDOD または CICRDOR は、REXX トランザクションでの Db2 プランの使用を許可します。

提供されたトランザクションを REXX/CICS 環境用に変更する場合、または Db2 インターフェース・コードを使用する新規トランザクションを実装する場合は、Db2 エントリー定義も変更または追加する必要があります。

### Db2 プランへの CICSQL プログラムの BIND

CICSTS56.REXX.SCICJCL データ・セット内の CICBIND ジョブを使って、CICSQL を正しい Db2 パッケージにバインドします。ジョブを編集し、実行してください。

使用している Db2 のレベルによっては、このジョブで条件コード 4 を受け取る場合があります。

## REXX/CICS システムの定義と管理

REXX/CICS のシステム定義、カスタマイズ、管理について説明します。

### REXX/CICS 許可コマンドと許可コマンド・オプション

いくつかの REXX/CICS コマンドやコマンド・オプションは、許可されたものと見なされています。

[REXX/CICS コマンド](#)を参照してください。REXX/CICS 許可コマンドは、以下の場合にのみ実行できます。

- exec を発行するユーザー ID が REXX/CICS 許可ユーザーである場合。許可ユーザーは AUTHUSER コマンドで定義されます。
- exec が REXX/CICS 許可サブライブラリーからロードされた場合。許可ライブラリーは、(CICS 始動プロシージャー/JCL で) DD 名 CICAUTH または CICEXEC に割り振られた MVS 区分データ・セットです。

これらの規則は、コマンドを試行する exec が許可 exec によって発行されたかどうかに関係なく、適用されます。

### システム・プロファイル exec

CICS システム再始動の後、最初のユーザー exec が実行される前に、CICSTART という名前のシステム・プロファイル exec が発行されます。

通常、システム・プロファイル exec には、DD 名 CICAUTH または CICEXEC に割り振られる MVS PDS REXX 許可ライブラリー内に存在する必要があるシステム・カスタマイズ・コマンド、許可サブライブラリー定義、許可ユーザー定義、および許可コマンド定義が含まれています。

CICS システムの再始動後にユーザーが初めて REXX/CICS に入ったときに、CICSPROF という名前のシステム・ユーザー・プロファイル exec が発行されます。この exec には、すべてのユーザーが実行する必要があるセットアップ手順が含まれています。CICSPROF は、ユーザー・プロファイルも呼び出します。

ユーザー・プロファイルはユーザーが作成し、保守する exec です。これによってユーザーは REXX/CICS 環境をカスタマイズできます (パスの設定、RETRIEVE キーの変更、他の exec の呼び出しなど)。このプロファイルはユーザー個人の RFS ディレクトリーに配置する必要があります。

### 関連資料

478 ページの『[CICSTART メンバーの更新](#)』

hlq.CICSTS56.REXX.SCICEXEC データ・セット内のメンバー CICSTART には、REXX/CICS 環境のデフォルト定義が含まれています。CICSTART は、CICS システムが始動した後、CICS/REXX プログラムを使用する最初のトランザクションが出される時に実行されます。

## MVS PDS REXX 許可ライブラリー

DD 名 CICAUTH および CICEXEC に割り振られる MVS 区分データ・セットは、すべて REXX/CICS 許可ライブラリーと見なされます。

複数のデータ・セットが連結されている場合、それらは連結の順序で検索されます。CICSTART exec は CICEXEC にあります。

- ユーザーは、REXX/CICS PATH コマンドを使用して、独自の非許可ライブラリーの動的割り振りを行うことができます。
- すべてのユーザーは CICEXEC データ・セットから exec を実行できます。また、CICEXEC からロードされたすべての exec は REXX/CICS 許可コマンドを使用できます。
- CICEXEC JCL DD ステートメントにあるデータ・セットは、可変ブロック化形式でなければなりません。

## 許可ユーザーの定義

AUTHUSER コマンドを使用して、ユーザーを許可ユーザーとして指定できます。

すべての AUTHUSER コマンドを CICSTART exec 内、または CICSTART exec から発行される exec 内に入れることをお勧めします。478 ページの『[CICSTART メンバーの更新](#)』を参照してください。

## システム・オプションの設定

システム・オプションは、REXX/CICS SETSYS コマンドを使用して指定できます。

システム全体にわたる SETSYS コマンドを CICSTART exec 内に入れることをお勧めします。478 ページの『[CICSTART メンバーの更新](#)』を参照してください。

## REXX ファイル・システム (RFS) ファイル・プールの定義

RFS ファイル・プールを定義したり、初期化したり、ファイルを追加したりするには、FILEPOOL のさまざまなコマンドを使用します。RFS ファイル共用許可には、RFS AUTH コマンドを使用します。

- FILEPOOL DEFINE コマンドを使用して、RFS ファイル・プールを定義します。
- FILEPOOL FORMAT コマンドを使用して、各ファイル・プール内の最初のファイルを初期化します。
- FILEPOOL ADD コマンドを使用して、VSAM ファイルを RFS ファイル・プールに追加します。

### RFS ファイル共用の許可

RFS AUTH コマンドを使用して、ファイル共用の許可を指定します。

通常は、自分が所有するリソースの共用を許可することができます。REXX/CICS 許可ユーザーは、自分が作成したすべての RFS ディレクトリーの共用の許可を指定できます。

## CICSTART のための PLT エントリーの作成

CICREXD または CICREXR プログラムを呼び出す CICS プログラム・ロード・テーブル (PLT) エントリーを作成することにより、CICS システム初期設定の直後に CICSTART exec を発行することができます。

そのようにしない場合、領域を始動した後の最初の REXX/CICS ユーザーにより CICSTART exec が実行されます。

## セキュリティー出口

置き換え可能なセキュリティー出口 CICSECX1 および CICSECX2 について説明します。IBM は、カスタマイズや置き換えが可能なサンプルのアセンブラー出口を提供しています。

このセクションには、プロダクト・センシティブ・プログラミング・インターフェース情報が含まれています。

注：これらの出口は、REXX/CICS と同じ領域に存在する必要があります (例えば、分散プログラム・リンクの使用は許可されません)。

## CICSECX1

CICSECX1 は、MVS データ・セットのアクセス・セキュリティ出口です。この出口は EXEC CICS LINK によって呼び出され、パラメーターは COMMAREA で渡されます。

### パラメーター



**重要:** このトピックには、プロダクト・センシティブ・プログラミング・インターフェースと関連ガイダンス情報が含まれています。

COMMAREA では、出口への入力に以下が入ります。

パラメーター	バイト数	データ・タイプ	説明
1	4	フルワード	戻りコード
2	8	文字	CICS サインオン ID
3	4	文字	要求される機能
4	3		IBM で使用するために予約済み
5	44	文字	MVS データ・セット

### 戻りコード

0

許可される機能要求

4

リジェクトされた機能要求

### 関数 ID

A

ALLOCATE REQUEST

E

EXPORT 要求

F

FREE REQUEST

I

IMPORT 要求

## CICSECX2

CICSECX2 は、REXX File System アクセス・セキュリティ出口です。

### パラメーター



**重要:** このトピックには、プロダクト・センシティブ・プログラミング・インターフェースと関連ガイダンス情報が含まれています。

COMMAREA では、出口への入力に以下が入ります。

パラメーター	バイト数	データ・タイプ	説明
1	4	フルワード	戻りコード
2	8	文字	CICS サインオン ID
3	1	文字	要求される機能



パラメーター	バイト数	データ・タイプ	説明
4	3		IBM で使用するために予約済み
5	4	フルワード	完全修飾 RFS ファイル ID スtring のアドレス
6	4	フルワード	RFS ファイル ID スtring のディレクトリー・パスの長さ

## 戻りコード

0

許可される機能要求

ゼロ以外

許可されません

## 関数 ID

A

変更

R

Read (読み取り)

U

更新

## パフォーマンスの考慮事項

クライアント/サーバーのサポートは、パフォーマンス上の利点をかなり見込める REXX 機能です。この機能により、アプリケーション機能を提供するために、ネストされた REXX ではなくサーバー REXX exec がしばしば使用されます。そのようなサーバーのパフォーマンス上の特性は、管理の容易さの面で優れています。ネストされた exec と比べたときのサーバー exec の利点として、サーバー exec を開始すると、終了する前に複数のクライアント要求を処理できることです。そのため、パスの長さが短くなり、応答時間が向上して、多くの場合にシステム・リソースの使用量が少なくなります。詳しくは、[ハイレベルで、最適な透過的 REXX クライアント・インターフェース](#) を参照してください。

典型的な CICS アプリケーション・プログラミングとは異なり、EXEC CICS SUSPEND コマンドを発行する必要はありません。それは、REXX/CICS が一定の間隔でプログラムを自動的に一時停止することによって処理されます。

REXX/CICS exec は VSAM ベースの REXX ファイル・システムのファイル内、または MVS 区分データ・セットに常駐させることができます。CICS は、起動時に特定の MVS 区分データ・セットを CICS 領域に割り振ります。REXX PATH (データ・セット名が指定された場合のみ)、IMPORT、EXPORT、および ALLOC コマンドで、SVC 99 を使用することにより、MVS 区分データ・セットを動的に割り振ることができます。SVC 99 を過度に使用すると、CICS 領域のパフォーマンスが低下することがあります。

## セキュリティ

REXX トランザクションは、CICS トランザクション・セキュリティを使用して制御できます。REXX トランザクションは、それが実行される CICS 領域の特質により、広く使用できるようにすることも、何人かの個人に限定することもできます。

REXX/CICS は、CICS 提供のコマンド・レベル・インタープリター・トランザクション (CECI) のより洗練されたバージョンと見なすことができます。REXX トランザクション (REXX exec の発行に使用) は CECI トランザクションとよく似ており、CICS トランザクション・セキュリティを使用して制御できます。

注: 既存の REXX exec を実行する際には REXX トランザクションは必要ありませんが、ユーザーまたはプログラマーが REXX exec を作成または変更した後にテストするためには必要です。

## REXX/CICS による複数トランザクション ID のサポート

REXX/CICS は、REXX 以外のトランザクション ID (TRANID) を REXX/CICS サポート・プログラムに関連付ける機能をサポートしています。

このケースでは、発行される REXX exec の名前は直前の DEFTRNID コマンドによって決まります。これにより、REXX で、トランザクション・セキュリティを exec ごとに引き続き使用することが可能になります。

## REXX/CICS ファイル・セキュリティ

RFS ディレクトリー・レベルでのアクセスは、RFS AUTH コマンドおよび RFS 置き換え可能セキュリティ出口によって制御されます。

CICAUTH DD 名によってアクセスされる許可 exec へのアクセスは、DEFCMD コマンドおよび DEFSCMD コマンドの AUTH パラメーターによって制御されます。

動的に割り振られたデータ・セットへのアクセスは、データ・セット置き換え可能セキュリティ出口によって制御されます。

## REXX/CICS コマンド・レベル・セキュリティ

コマンド・レベル・セキュリティを使用して、CICS (および他の製品やシステム) の機能へのアクセスを制御できます。

現在のソフトウェア・プラクティスによっては、CICS リソース・セキュリティに依存するだけでは、その有効性に限界がある場合もあります。セキュリティ管理を追加する目的で、REXX/CICS の設計にはコマンド・レベル・セキュリティという概念が取り入れられています。REXX/CICS におけるほとんどの機能はコマンドとしてアクセスされるので、コマンド・レベル・セキュリティを使用して、CICS (および他の製品やシステム) の機能へのアクセスを制御できます。例えば、VSAM ファイル・アクセスは、READ、WRITE、および REWRITE コマンドを使用して行われます。

REXX/CICS コマンド・レベル・セキュリティは、DEFSCMD および DEFCMD AUTH パラメーターの使用と、許可 REXX/CICS ライブラリー・サポートのプロビジョンによって制御されます。

コマンド実行セキュリティは、特定の REXX/CICS コマンドまたはコマンド・キーワードの使用を制御します。この制御は一般に、特定のコマンド (またはコマンド・オプション) を許可済みとして指定することによって行います。このようなコマンド指定は、DEFCMD および DEFSCMD コマンドで行います。許可コマンドを正しく実行するためには、以下のいずれかの条件に該当する必要があります。

1. このコマンドは、CICS 始動 JCL プロシーチャーの DD 名 CICAUTH または CICEXEC に割り当てられた MVS PDS からロードされた exec から実行する必要があります。
2. このコマンドは許可ユーザーによって実行される必要があります。AUTHUSER コマンドでユーザーに権限を付与できます。

## REXX/CICS 許可コマンド・サポート

どの REXX/CICS コマンドも、REXX/CICS システム管理者が許可対象として指定できます。許可コマンドは、許可 REXX/CICS ユーザーが発行した exec、または許可 REXX/CICS ライブラリーからロードされた exec でのみ正常に実行できます。

許可 exec ライブラリー CICAUTH にアクセスできるのは、REXX/CICS 許可ユーザーのみです。すべてのユーザーは、CICEXEC 許可ライブラリーにある exec を実行できます。すべてのユーザーは、CICUSER 無許可ライブラリーにある exec を実行できます。許可ユーザーは、既存の許可ユーザーが定義することも、許可 exec で定義することもできます。REXX/CICS 初期設定時 (CICS 再始動後の最初の REXX/CICS トランザクション時) に呼び出される REXX/CICS CICSTART exec は、自動的に許可されます。これは、許可ユーザーおよびライブラリーを定義するための論理プレースです。

REXX/CICS ライブラリーへのアクセスは容易に制御できるので、これは CICS 実動プログラム・ライブラリーへのアクセスの制御に対応する論理側の制御になります。サイトで重要と思われるコマンド (READ、WRITE、DELETE など) を、実動領域で許可対象と定義することもできます。これは、許可コマンドを発行する exec を作成でき、許可コマンドを含むこれらの exec をすべてのユーザーが呼び出せるか他の許可ユーザーのみが呼び出せるかを決定できるのは、許可ユーザーのみであることを意味します。

注：CICS START、LINK、および XCTL コマンドを REXX/CICS 許可コマンドと再定義することにより、外部 API への REXX/CICS exec のアクセス許可を制御できます。

## セキュリティ定義

一般ユーザー、許可ユーザー、許可コマンド、許可 exec、システム・ライブラリーなどの REXX/CICS のセキュリティ定義について説明します。

### REXX/CICS 一般ユーザー

AUTHUSER コマンドで許可済みと定義されていない REXX/CICS ユーザーは、REXX/CICS 許可コマンドを使用できません。ただし、これらのユーザーは、ユーザー・コマンド (DEFCMD コマンドを使用して定義) および exec を定義、作成、変更、使用することができます。ユーザーは、CICEXEC ライブラリーにある REXX/CICS 許可 exec を使用することもできます (ただし、定義、作成、変更することはできません)。

個々のユーザーの情報は、ユーザー ID を指定することによって REXX/CICS 環境で保守されます。各ユーザーは一意的に識別される必要があり、各ユーザーは REXX/CICS 環境に 1 度だけサインオンする必要があります。同じユーザー ID を持つ 2 人のユーザーが同時に操作を行うと、異常な結果を招くおそれがあります。

### REXX/CICS 許可ユーザー

許可ユーザーは AUTHUSER コマンドで定義され、許可 REXX/CICS コマンド (AUTH オプションを指定した DEFCMD または DEFSCMD コマンドを使用して定義されたコマンド) を使用することが許可されます。

### REXX/CICS 許可コマンド

許可コマンドとは、許可ユーザーのみが使用できる REXX/CICS コマンド、または許可 exec からのみ使用できる REXX/CICS コマンドのことです。許可コマンドは、AUTH オプションを指定した DEFCMD または DEFSCMD コマンドを使用して定義されます。

### REXX/CICS 許可 exec

許可 exec は、DD 名 CICEXEC または CICAUTH からロードされたプログラム (exec) であり、許可済みと見なされます。つまり、これらのプログラムは許可 REXX/CICS コマンドを使用することが許可されます。すべての REXX/CICS ユーザーは CICEXEC 許可プログラムにアクセスできますが、CICAUTH 許可プログラムにアクセスできるのは許可ユーザーだけです。

### REXX/CICS システム・ライブラリー

REXX 言語で書かれたすべての許可コマンドは、DD 名 CICAUTH に連結された MVS 区分データ・セットからロードする必要があります。これらは、IBM が提供する場合と、お客様 (またはベンダー) が提供する場合があります。

すべての許可 exec は、DD 名 CICEXEC または CICAUTH に連結された MVS 区分データ・セットからロードする必要があります。これらは、IBM が提供する場合と、お客様 (またはベンダー) が提供する場合があります。

許可されていないものの、すべての REXX/CICS ユーザーが共用するユーザー exec は、DD 名 CICUSER に割り振られた、または連結された MVS 区分データ・セットに置くことができます。

注：

1. DEFCMD または DEFSCMD の AUTH オプションは、それ自体が許可コマンド・オプションです。つまり、そのコマンドを発行するユーザーが許可ユーザーであるか、許可ライブラリーからロードされた exec からそのコマンドが発行された場合のみ、AUTH を使用できます。
2. EXECLOAD および EXECDROP コマンドは許可コマンドです。したがって、許可サブライブラリーからの exec に対して EXECLOAD を実行できるのは、許可ユーザーまたは許可 exec のみです。

## 第 17 章 IBM Health Checker for z/OS による CICS 構成の検査

IBM Health Checker for z/OS は、潜在的な構成上の問題の検出を単純化かつ自動化して、可用性への影響や障害の発生を未然に防ぐ z/OS コンポーネントです。CICS TS は、CICS システム構成のベスト・プラクティスを定義したヘルス・チェッカー規則をサポートします。

### 始める前に

z/OS オペレーティング・システムに以下の PTF がインストールされていることを確認してください。

- z/OS V2.1: UA91584
- z/OS V2.2: UA91583

### このタスクについて

各 CICS 領域が、CICS システム・トランザクション CHCK を 30 分間隔で実行し、ベスト・プラクティスに準拠しているかどうかを調べて報告します。

CICS ヘルス・チェッカー規則が、IBM Health Checker for z/OS アドレス・スペース内で 30 分間隔で実行され、IBM Health Checker for z/OS が、直前の 30 分間に実行されていたすべての CICS 領域について報告します。いずれかのベスト・プラクティスに準拠していない LPAR の領域が検出された場合、IBM Health Checker for z/OS は、ユーザーが修正処置を取れるように、準拠していない領域に関する詳細情報を含む警告メッセージを発行します。

IBM Health Checker for z/OS について詳しくは、[IBM Health Checker for z/OS ユーザーズ・ガイド](#)を参照してください。

CICS TS は、以下のヘルス・チェッカー規則をサポートします。

### セキュリティ・ルール

以下のルールは、CICS TS セキュリティーのためのベスト・プラクティスを定義します。

#### **CICS\_CEDA\_ACCESS**

非認証ユーザーが CEDA にアクセス可能かどうかを検査します。アクセス可能であると、LPAR のセキュリティを損なう可能性があります。

#### **CICS\_JOBSUB\_SPOOL**

非認証ユーザーがシステム・スプール (有効な場合) にアクセス可能かどうかを検査します。アクセス可能であると、LPAR のセキュリティを損なう可能性があります。

#### **CICS\_JOBSUB\_TDQINTRDR**

内部読み取りプログラムに書き込む一時データ・キューに非認証ユーザーがアクセス可能かどうかを検査します。アクセス可能であると、LPAR のセキュリティを損なう可能性があります。

## CICS\_CEDA\_ACCESS

この規則は、IBM 提供のトランザクション CEDA をデフォルト・ユーザーが使用できるかどうか、またはこの領域で CICS セキュリティーがオフになっている (つまり SIT パラメーター **SEC=NO**) かどうかを検査します。

いずれかが検出された場合は、その領域の IP アドレスとポート番号に接続できるすべてのユーザーが、CICS の構成を変更して、この LPAR のセキュリティを侵害できることを意味します。

### ベスト・プラクティス・アドバイス

一般的に、領域は **SEC=YES** を使用して定義する必要があり、CEDA は許可されたサインオン済みユーザーのみがシステム構成に対する変更を行えるように保護されている必要があります。

必ず **SEC=NO** を使用して CICS を実行する必要がある場合、この領域の CEDA アクセスを無効にするか、領域を構成して分離された LPAR で実行してください。

## CICS\_JOB SUB\_SPOOL

この規則は、システム・スプーリング・インターフェースが有効 (つまり、SIT パラメーター **SPOOL=YES**) かどうか、また、有効である場合には、IBM 提供のトランザクション CECI をデフォルト・ユーザーが使用できるかどうか、またはこの領域で CICS セキュリティーがオフになっている (つまり SIT パラメーター **SEC=NO**) かどうかを検査します。

システム・スプーリング・インターフェースが有効な状況で、CECI をデフォルト・ユーザーが使用できる場合、または CICS セキュリティーがオフになっている場合には、CICS 領域の IP アドレスおよびポート番号に接続できるすべてのユーザーが、認証なしで領域ユーザー ID を使用してリモートから z/OS システムにジョブを実行依頼して実行することができます。

### ベスト・プラクティス・アドバイス

API コマンド **SPOOLWRITE** を使用してスプールに書き込む必要があるアプリケーションが存在する領域でのみ、SIT パラメーター **SPOOL=YES** を使用してください。

ロード・モジュール・スキャナー DFHEISUP を使用して、どのプログラムが SPOOLWRITE コマンドを使用するかを調べることができます。

一般的に、領域は **SEC=YES** を使用して定義する必要があり、CECI は許可されたサインオン済みユーザーのみが CECI トランザクションを使用してコマンドを発行できるように保護されている必要があります。

必ず **SEC=NO** を使用して CICS を実行する必要がある場合、この領域の CECI アクセスを無効にするか、領域を構成して分離された LPAR で実行してください。

## CICS\_JOB SUB\_TDQINTRDR

この規則は、内部読み取りプログラムに書き込む一時データ・キューに対して、デフォルト・ユーザーによる書き込みが可能かどうか、または CICS セキュリティーがオフ (つまり、SIT パラメーター **SEC=NO**) になっている領域でその一時データ・キューを使用可能であるかどうかを検査します。

一時データ・キューが内部読み取りプログラムに書き込むように定義されているとき (つまり、一時データ・キューが区画外として定義され、INTRDR を参照する SYSOUT を持つ DD 名を指定しているとき)、以下のいずれかの状況が検出される場合、CICS 領域の IP アドレスおよびポート番号に接続できるすべてのユーザーは、ジョブを認証なしで領域ユーザー ID を使用してリモートから実行依頼して、そのジョブを z/OS システム上で実行することができます。

- 内部読み取りプログラムへの書き込みを行う一時データ・キューに、デフォルト・ユーザーが書き込むことができる。
- IBM 提供のトランザクション CECI を、デフォルト・ユーザーが使用できる。
- CICS セキュリティーがオフになっている。

### ベスト・プラクティス・アドバイス

内部読み取りプログラムへの書き込みを行う一時データ・キューを保護して、許可されたユーザーのみがサインオンしたうえで書き込めるようにする必要があります。

一般的に、領域は **SEC=YES** を使用して定義する必要があり、CECI は許可されたサインオン済みユーザーのみが CECI トランザクションを使用してコマンドを発行できるように保護されている必要があります。

必ず **SEC=NO** を使用して CICS を実行する必要がある場合、この領域の CECI アクセスを無効にするか、領域を構成して分離された LPAR で実行してください。

# 第 18 章 Parallel Sysplex への CICS のマイグレーション

非 z/OS 環境から、カップリング・ファシリティを使用するシスプレックス (Parallel Sysplex) への CICS システムおよびアプリケーションのマイグレーション

## シスプレックスと Parallel Sysplex

シスプレックスは、特定の複数システム・ハードウェア・コンポーネントおよびソフトウェア・サービスを介して互いに通信を行う一連の z/OS システムです。シスプレックスでは、通信システム間でのリソース共用が可能です (テープ、コンソール、カタログ、およびその他のリソース)。シスプレックスでは、協働可能な処理装置の数および z/OS オペレーティング・システムの数が増加するので、処理できる作業量もまた増加します。

Parallel Sysplex は、1 つ以上のカップリング・ファシリティ (CF) を使用するシスプレックスです。カップリング・ファシリティとは、シスプレックスで高速のキャッシング、リスト処理、およびロッキング機能を提供する特殊な論理区画です。Parallel Sysplex では、シスプレックス内の任意のシステムがワークロードを実行できるようデータを共用することができます。

注：この資料ではこれ以降、カップリング・ファシリティを持たないシスプレックスとカップリング・ファシリティを持つシスプレックスを区別しません。カップリング・ファシリティが存在し、Parallel Sysplex にマイグレーションしていることを前提とします。シスプレックスという用語が出現する場合、それは Parallel Sysplex を意味しているものとご理解ください。

## Parallel Sysplex で CICS を実装することの利点

Parallel Sysplex で CICS を実装することによって得られる利点に関する情報。

従来、企業は複数の CICS 領域を実行し、それぞれの領域を特定の目的に特化して使用していました (端末専有領域、アプリケーション所有領域、キュー所有領域など)。次のステップは必然的に、CICS を拡張して Parallel Sysplex での可用性を最大化することになります。

Parallel Sysplex が CICS 環境にもたらす利点として、次の点が考えられます。

- 高可用性と連続稼働。

CICS の機能を提供する領域が複数存在する場合、それらの領域のいずれかが停止しても、少なくとも領域の 1 つが使用可能であれば、ユーザーに対してサービスを継続して提供することができます。CICS 領域で計画停止または計画外の停止が発生した場合、本来領域で処理される作業はすべて、他の領域の 1 つに自動的にルーティングされます。そうすることには、以下のような利点があります。

- 計画外の停止の数と影響を最小限に抑える。
- 計画停止の数と影響を最小限に抑える。
- CICS アプリケーションの可用性を最大化する。
- CICS アプリケーションがビジネスの SLA 要件を満たすか、それを上回るようにする。

- スケーラビリティ。

Parallel Sysplex の概念は、あらゆる作業をシスプレックスのどのメンバーでも実行できるようにすることです。CICS においてこのことは、CICS で容量がさらに必要な場合に、別の CICS 領域を素早く簡単に複製し、シスプレックス内のいずれかのシステムに追加でき、作業がその領域に自動的に流れ始める必要があることを意味します。

- 柔軟性。

CICS 領域の数を減らしたい場合には、領域を停止することができ、その領域で実行されていたすべての作業は残りの領域の 1 つにルーティングされる必要があります。

シスプレックス内の 1 つのシステムから別のシステムに CICS 領域を移動することもできます (例えば、容量を増やすために 1 つの中央演算処理複合システム (CPC) をアップグレードする場合)。簡単に領域



を移動できるようにすることで、アップグレードした CPC の追加容量を利用し、領域の移動元 CPC の容量を解放することができます。

- パフォーマンス。

CICSplex がシスプレックス内の複数のシステムにまたがるかどうかに関係なく、ユーザーに最短の応答時間を提供できる CICS 領域に CICS トランザクションをルーティングすることにより、パフォーマンスを最適化することができます。

- システム管理の容易化と単一のシステム・ビュー。

CICSplex SM は、単一の z/OS イメージだけでなく、Parallel Sysplex 全体に渡る複数の CICS 領域を管理できます。

これらの利点を最大限活用するために、アプリケーションを変更する必要がある可能性があります。アプリケーションを Parallel Sysplex で使用できるようにする方法については、[508 ページの『Parallel Sysplex への CICS のマイグレーションの計画』](#)を参照してください。

## CICSplex、CICSplex SM、および Parallel Sysplex

CICSplex、CICSplex SM、および Parallel Sysplex の間にはかなりの協同があります。それらを組み合わせて使用することで、できるだけ高い可用性と連続稼働を CICS アプリケーションに提供することができます。

### CICSplex

複数の CICS 領域間で相互通信が可能であり、インバウンド作業要求を協力して処理することができます。この特殊なクラスター・タイプは、CICSplex と呼ばれます。CICS 環境で CICSplex を使用することには多くの利点があります。例えば、スケラビリティが向上したり、単一システム・イメージを使用したりすることができます。CICSplex は、単一エンティティによって管理されます。これは、CICS ワークロードでの高可用性や連続稼働を実現するための重要なテクノロジーです。ただし、これだけでは他のサブシステムまたはデータの可用性に関して何も行えません。場合によっては、CICS アプリケーションの業務要件 (例えば、Db2 データへのアクセスなど) を満たす必要があります。

### CICSplex SM

CICS TS の CICSplex SM エLEMENT は、複数のイメージに渡る複数の CICS システム (1 つの CICSplex) を単一の制御点から管理できるシステム管理インフラストラクチャーです。CICSplex SM を必要とする可能性のある企業は、数台のみの CICS システムを稼働するものから、数百またはそれ以上の CICS 領域を稼働するものまであります。大規模な z/OS シスプレックス環境では多くの場合、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必要な CICS システムが多数必要になる可能性があります。ワークロード管理、リアルタイム分析、およびモニタリング・サービスは、CICSplex SM 構成や CICSplex 環境の管理、および統計情報の収集のために使用されます。

CICSplex SM は、トランザクション処理環境の各 CICSplex 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・イメージを提供します。これは、CICSplex 内のリソースを定義およびデプロイするための単一制御点となります。

CICSplex SM ワークロード・マネージャーは動的にトランザクションをルーティングし、ターゲット・スコープ内の CICS システムに渡るすべての動的トランザクションおよびプログラム・リンクに対するワークロード管理を提供します。その際、領域の可用性を考慮に入れ、ワークロードのアフィニティやワークロードの分離が必要であればそれをすべて受け入れます。

### Parallel Sysplex

システム・コンポーネント、ハードウェア、ソフトウェア、またはデータを単一コピーとして持っている場合には、コンポーネントの障害や、オフラインにする必要があるコンポーネントの計画変更のために、システム障害の危険にさらされることは必至です。

Parallel Sysplex の 1 つの目標は、スケジュールされた停止によってアプリケーションの可用性に影響が及ばないようにし、シスプレックス内の他のシステムで作業が引き続き実行されるようにすることで、スケジュールされていない停止の影響を最小限に抑えることです。これを達成するうえで特に必要なことは、Parallel Sysplex 内のシステムが冗長になるように設計することです。アプリケーションは複数のシステ

ムで実行できなければならず、少なくとも 2 つのシステムからのデータ・アクセスが可能である必要があります。少なくともリソースの 2 つのインスタンスが存在していれば、リソースの 1 つで障害が発生してもアプリケーションは実行を継続できます。

**注 :** Parallel Sysplex は、継続的可用性のための機能のみを提供します。Parallel Sysplex だけでは、スケジュールされたものであれ、スケジュールされていないものであれ、アプリケーションの停止を解決できません。アプリケーション自体にも継続的可用性が設計されている必要があります。データの共用は、この設計の一部に過ぎません。

Parallel Sysplex は、シスプレックス内で実行されているサブシステムやアプリケーションで利用できる機能を提供します。その機能を使用することで、可用性を向上させ、システム管理を簡単に行えるようにし、スケーラビリティを改善することができます。以下に例を示します。

- XCF サービスを使用すると、あるシステムの許可プログラムが、同一システムまたは他のシステムのプログラムと通信できるようになります。システムに障害が起きた場合、XCF サービスは、バッチ・ジョブおよび開始タスクを、シスプレックス内の別の適格なシステムで再始動するための機能も提供します。
- カップリング・ファシリティを使用すると、シスプレックス内で実行される許可プログラムの並列処理が可能になり、データ共用が改善されます。

例えば、CICS 環境では、以下のことができます。

- CICS は、カップリング・ファシリティ経由でリカバリー不能な一時ストレージ・キューを共用することで、Parallel Sysplex を直接利用できます。
- CICS は、VSAM RLS シスプレックス全体に渡るレコード・レベルのロックを使用することで、Parallel Sysplex を間接的に利用できます (この例では、DFSMS VSAM は直接的な利用と見なされます)。
- CICSplex SM シスプレックス最適化ワークロード管理は、カップリング・ファシリティから領域状況情報を取得することで Parallel Sysplex を直接利用できます。

### **CICSplex、CICSplex SM、および Parallel Sysplex の組み合わせ**

CICS 環境の CICSplex と CICSplex SM は、単一システム・イメージ、ワークロード管理、およびある程度のデータ共用機能を提供します。ただし、これは CICS の場合だけです。Parallel Sysplex は、CICS だけでなく、適切に構成されているすべてのワークロードに対して単一システム・イメージとデータ共用機能を提供します。Parallel Sysplex を適切に設計および構成すると、例えばサブシステム、または z/OS システム全体が使用不可になったというだけの理由で CICS アプリケーションが停止してしまうのを避けられます。

## **Parallel Sysplex の原理**

Parallel Sysplex は、特定のハードウェア・コンポーネントおよびソフトウェア・コンポーネントを共用して高可用性ワークロード処理環境を実現する z/OS システムの集合です。

Parallel Sysplex は、接続されて単一の論理コンピューティング・プラットフォームとして動作する最大 32 個の z/OS システムのセットです。

基本シスプレックスは 1990 年に導入されました。

1994 年には、カップリング・ファシリティが加えられた Parallel Sysplex が導入されました。

基礎となる構造は事実上、ユーザー、ネットワーク、アプリケーション、および操作に対して透過的です。

Parallel Sysplex の背後にある主要な原則は、データ共用機能を提供することです。これには、次の利点があります。

- サーバー内、LPAR 内、およびサブシステム内の単一障害点の減少/除外
- アプリケーション可用性の向上
- 単一システム・イメージ
- 動的セッション・バランシング
- 動的トランザクション・ルーティング
- スケーラブルなキャパシティー

## Parallel Sysplex のコンポーネント

Parallel Sysplex は、すべてが 1 つ以上の同じカップリング・ファシリティにアクセスできるシステムのセットです。基本シスプレックスが、定義された名前 (シスプレックス名) を持つ実際のエンティティであるのに対し、Parallel Sysplex はより概念的で、Parallel Sysplex の名前やそれに含まれるシステムのリストを保持する単一の場所はありません。Parallel Sysplex には、いくつかの構成要素があります。

### カップリング・ファシリティ

どの Parallel Sysplex においても重要な構成要素は、カップリング・ファシリティ (CF) インフラストラクチャーです。Parallel Sysplex では、中央演算処理装置複合システム (CPC) が CF 経由で接続されます。CF は、ハードウェアと、シスプレックス内のシステムにサービスを提供する特殊マイクロコード (制御コード) とで構成されます。CF は独自の LPAR で実行されます。CF ストレージの領域は、CF 利用者による特定の目的の使用のために割り振られます。これらの領域を構造と呼びます。構造には 3 つのタイプがあります。

- ロック:

細分度の高いデータの直列化用。例えば、ロックは IMS DB や IBM MQ データベースに対する IRLM や、DFSMS VSAM RLS を使用する際の CICS で使用されます。

- キャッシュ:

データの保管およびローカル・バッファ・プールのコヒレンシー情報の維持用。例えば、キャッシュは、カタログ共用のための DFSMS、RACF データベース、IBM MQ データベース、IMS 用の VSAM データベースと OSAM データベース、および DFSMS VSAM RLS を使用する際の CICS で使用されます。キャッシュには、ディレクトリー項目とオプションのデータ項目の両方が入ります。

- リスト

共用キューおよび共用状況情報用。CICS で使用されるリスト構造としては、カップリング・ファシリティ・データ・テーブル (CFDT)、名前付きカウンター、CICS 共用一時ストレージ、および CICSplex SM 領域状況があります。

CF のストレージには、ダンプ・スペースと呼ばれる領域もあります。これは、直列化された構造の制御情報を素早くキャプチャーするために SVC ダンプ・ルーチンで使用されます。ダンプの後、ダンプ・スペースは z/OS CPC ストレージにコピーされ、その後、SVC ダンプ・データ・セットにコピーされます。ダンプ・スペースには、複数の構造のデータを含めることができます。CF 構造の定義は、カップリング・ファシリティ・リソース管理 (CFRM) ポリシーに保持されます。

z/OS のシステム間拡張サービス (XES) コンポーネントを使用すると、アプリケーションおよびサブシステムが、このカップリング・ファシリティを利用できるようになります。

Parallel Sysplex の高可用性の特性は、中断することなく構造を 1 つの CF から別の CF に移動できること、つまり CF を使用するシステムに影響を与えることなくサービスに対して 1 つの CF をオフラインにすることができる機能に依存しています。少なくとも 2 つの CF がすべての Parallel Sysplex にあり、シスプレックス内のすべてのメンバーがその CF にアクセス可能である必要があります。

### 結合データ・セット

z/OS では、Parallel Sysplex 内のすべてのシステムでデータ・セット (加えて、可用性のために代替データ・セットが推奨されます) を共用する必要があります。z/OS では、Parallel Sysplex、システム、XCF グループ、およびそれらのメンバーに関連した情報が、シスプレックス結合データ・セットに保管されます。

### XCF

システム間カップリング・ファシリティ (XCF) サービスは、あるシステム上の権限のあるプログラムが、同じシステム上の、または他のシステム上のプログラムと通信することを可能にします。システムに障害が起きた場合、XCF サービスは、バッチ・ジョブおよび開始タスクを、シスプレックス内の別の適格なシステムで再始動するための機能も提供します。

アプリケーション・コンポーネントは XCF に対して定義され、アプリケーションをサポートするその他のコンポーネントの存在を認識します。1 つのコンポーネントが失敗すると、XCF は自動的に他のコンポーネントに通知します。XCF の詳細については、「[z/OS MVS プログラミング: シスプレックス・サービスガイド](#)」の『XCF の概念』を参照してください。

シスプレックスのノードは XCF 通信サービス (XCF) を使用して、シスプレックス TCP/IP スタック間の調整を実行したり、新規 TCP/IP スタックが開始されるタイミングを検出したり、TCP/IP スタックが停止されるタイミングまたは (障害が発生した後に) XCF グループから出るタイミングを学習したりします。この情報は、アプリケーションの移動を自動化したり、シスプレックス・ディストリビューターがインテリジェント・ワークロード管理に関する決定を行えるようにしたりするために必要不可欠です。XCF 通信メッセージのトランスポートは、カップリング・ファシリティーを使用して行うか、IBM エンタープライズ・システム接続 (ESCON) または IBM ファイバー・チャンネル接続 (FICON®) を使用して直接行うことができます。

## システム・ロガー

z/OS System Logger は、ログ・レコードの保管および再呼び出しを行うための汎用ロギング機能を提供します。シスプレックス内のすべてのシステムからの全情報を結合した単一の syslog を提供します。

System Logger を利用するものとして、シスプレックス全体にわたる syslog 用の OPERLOG、シスプレックス全体にわたるエラー情報用の LOGREC、および DFHLOG ログ・レコードと DFHSHUNT ログ・レコードを Logger に書き込む CICS があります。

## z/OS ワークロード・マネージャー

z/OS のコンポーネントの 1 つであり、単一の z/OS イメージまたは複数のイメージで同時に複数のワークロードを実行するための機能を提供します。

## Parallel Sysplex のネットワークング

高可用性を持つ Parallel Sysplex 環境を設計する際の全体的な目標は、いずれか 1 つのコンポーネントが失われてもアプリケーションの可用性に影響しない環境を作ることです。高可用性を実現し、Parallel Sysplex 内のシステムおよびサブシステムの障害が自動的に解決されるようにするには、アプリケーションを単一のネットワーク・アドレスに固定しようとしなくて済む必要があります。これを行ういくつかの方法があります。

### VIPA (仮想 IP アドレス)

これまで、IP アドレスは物理リンクと関連付けられており、ネットワーク内の中間リンクの障害を転送することはできませんでしたが、エンドポイントが障害点となっていました。z/OS システム・スタックに TCP/IP へのフォールト・トレラント・ネットワーク接続を提供する VIPA が導入されました。これにより、ハードウェア・コンポーネントに関連付けられていないものの常に使用可能な仮想インターフェースの定義が使用できるようになります。ルーティング・ネットワークには、VIPA は z/OS に間接的に接続されているホスト・アドレスのように見えます。ネーム・サーバーは、物理インターフェースではなく、TCP/IP スタックの VIPA を返すよう構成されています。物理インターフェースで障害が発生する場合、代替パスを使用するよう IP ルーティング・テーブルを更新するための動的ルートの更新が送信されます。IP 接続は切断されませんが、残りの物理インターフェースから中断なしでリカバリーされます。

VIPA には、次の 2 つのバージョンがあります。

- 静的 VIPA
- 動的 VIPA (DVIPA)

静的 VIPA は、特定の TCP/IP スタックと関連付けられる IP アドレスです。静的 VIPA では、ARP のテーブルまたは動的ルーティング・プロトコル (例えば OSPF) のいずれかを使用して、ネットワーク・インターフェースの障害によってメインフレーム・アプリケーション通信が途絶えないようにすることができます。単一ネットワーク・インターフェースがホストで動作している間、ホスト上のアプリケーションとの通信は持続します。静的 VIPA では TCP/IP スタック間の調整が必要ないため、シスプレックス (XCF 通信) が不要です。

動的 VIPA (DVIPA) は、複数のスタックで定義することができ、シスプレックス内の 1 つの TCP/IP スタックから別の TCP/IP スタックに自動的に移動することができます。1 つのスタックが 1 次スタックまたは所有スタックとして定義され、その他のスタックはバックアップ・スタックとして定義されます。IP ネットワークに認識されるのは 1 次スタックだけです。

シスプレックス内の TCP/IP スタックは DVIPA やその存在、現在の位置に関する情報を交換します。スタックは、パートナー・スタックが機能しているかどうかを引き続き認識します。例えば、ある種の障害が発生したために所有スタックが XCF グループを出る場合、バックアップ・スタックの 1 つが自動的にその代わりとなり、DVIPA の所有権を引き継ぎます。ネットワークでは、ルーティング・テーブルでの変更や、ARP 要求に応答するアダプターの変更のみが認識されます。これらの DVIPA と関連付けられるアプリケ

ーションはバックアップ・システムでアクティブとなり、ホット・スタンバイと高可用性をサービスに提供します。DVIPA アドレスは、サーバー・アプリケーションが実行されるシスプレックス内のイメージに関係なくアプリケーションを識別し、シスプレックス内のイメージ間で移動するときにアプリケーションがその ID を保持できるようにします。

### シスプレックス・ディストリビューター

シスプレックス・ディストリビューターは、シスプレックス内の接続ワークロード管理を提供します。これは、それぞれがデータへのアクセスを共用する複数の同時アプリケーション・インスタンスを実装するシステム間のワークロードとログオンのバランスを取ります。シスプレックス・ディストリビューターによってシスプレックス内の複数のサーバー・インスタンスに IP ワークロードを分散させることができます。その際、クライアントやネットワークング・ハードウェアに対して変更を加える必要はなく、接続のセットアップで遅延が生じることもありません。シスプレックス・ディストリビューターを使用すると、同じシスプレックス・クラスターに属するホストのセットに対して使用される単一のネットワーク可視 IP アドレスとして動的 VIPA を実装できます。IP ネットワーク上のクライアントはシスプレックス・クラスターを、クラスター内のホストの数に関係なく 1 つの IP アドレスとして認識します。

z/OS 環境で内部ワークロード管理を使用すると、特定の分散 DVIPA アドレスで TCP 接続要求を受け取る際に、アプリケーションのどのインスタンスがその特定の要求を実行するかに関する決定が、ルーティング・スタックになるよう構成されている TCP/IP スタックで実行されるシスプレックス・ディストリビューターによって行われます。シスプレックス・ディストリビューターでは(シスプレックス・ワークロード・マネージャーから)キャパシティーに関するリアルタイム情報を入手でき、サービス・ポリシー・エージェントからの QoS 情報を使用できます。そのため、内部ワークロード管理に特別な外部ハードウェアは必要ありません。ただし、分散 DVIPA へのインバウンド・メッセージはすべて、適切なアプリケーション・インターフェースに転送される前にルーティング・スタックを通過する必要があります。

### ポートの共用

ポート共用は、z/OS LPAR 内の IP アプリケーションのワークロードを分散する方法の 1 つです。TCP/IP では、複数のリスナーが、同じ組み合わせのポートとインターフェースを `listen` できます。このアプリケーション宛てに送られるワークロードは、同じポートで `listen` するサーバーのグループ間に分散することができます。ポート共用は、アクティブなシスプレックス・ディストリビューターの実装に依存しません。シスプレックス・ディストリビューターがなくても機能します。ただし、シスプレックス・ディストリビューターの運用に加えてポート共用を使用することもできます。

z/OS では、ポート共用の 2 つのモードをサポートしています。

- SHAREPORT
- SHAREPORTWLM

SHAREPORT を PORT ステートメントで指定すると、このポートおよびインターフェースの着信クライアント接続が TCP/IP スタックによってリスナー全体に分散されます。その際、ポートを共用するリスナーのサーバー許容効率 (SEF) に基づく重み付けラウンドロビン分散方式が使用されます。SEF は、新規接続要求を受け入れてそのバックログ・キューを管理する際のサーバー・アプリケーションの効率の指標です。これはおよそ 1 分間隔で計算されます。

SHAREPORTWLM は SHAREPORT の代わりに使用することができます。SHAREPORT と同様、SHAREPORTWLM も TCP リスナー・セット間での着信接続の分散を発生させます。ただし、SHAREPORT とは異なり、リスナーの選択は WLM サーバー固有の推奨に基づきます。この推奨は、各リスナーの SEF 値によって変更されます。これらの推奨は、WLM から約 1 分間隔で取得され、リスナーが追加作業を処理するキャパシティーを反映します。

### z/OS Communications Server 総称リソース

z/OS Communications Server 総称リソースは、シスプレックス内のどこで実行されるかに関係なく、アプリケーションの単一システム・イメージを提供します。ユーザーはアプリケーションの汎用リソース名を使用してアプリケーションにアクセスし、z/OS Communications Server は、パフォーマンスおよびワークロードの基準に基づいて実際のアプリケーション・インスタンスを判別します。これにより、ユーザーに影響を与えずにアプリケーション・インスタンスを追加、削除、および移動することができます。

CICS での z/OS Communications Server 汎用リソースの使用について詳しくは、『[Configuring](#)』の『[Configuring z/OS Communications Server generic resources](#)』を参照してください。



## Parallel Sysplex テクノロジーを直接利用する CICS 機能およびコンポーネント

Parallel Sysplex テクノロジーを直接利用する CICS 機能およびコンポーネントがいくつかあります。

CICS のこれらの要素は、例えば独自の z/OS カップリング・ファシリティ構造を実装することにより、Parallel Sysplex を直接利用します。

### CICS ログ・ストリーム

CICS ログ・マネージャは、z/OS System Logger によって提供されているサービスを使用して、Parallel Sysplex で z/OS カップリング・ファシリティにログ・データを書き込むことができます。

DFHLOG と DFHSHUNT は CICS のシステム・ログです。CICS 領域は、アプリケーション、製品混合、および使用されている関数とツールに応じて、さらに別のログ・ストリームを使用することもできます。

CICS システム・ログ、順方向リカバリー・ログ、自動ジャーナル、およびユーザー・ジャーナルは、特定のログ・ストリームにマップされます。ログ・ストリームは、カップリング・ファシリティ内で、または DASD 専用のログ・ストリームとして構造内で定義されます。

DASD 専用ロギングは、単一イメージ環境と非 Parallel Sysplex 環境で使用できます。DASD 専用ログ・ストリームは、カップリング・ファシリティのストレージを使用しません。DASD 専用ログ・ストリームには、単一のシステム有効範囲があります。1 つの DASD 専用ログ・ストリームに、同時に複数のシステムを接続することはできません。z/OS System Logger プログラミング・インターフェースを使用すると、Parallel Sysplex に渡るさまざまなシステムなど、アプリケーションの複数インスタンスからデータをマージすることができます (ただし、CICS のシステム・ログ・データはマージされません)。DASD 専用ロギングは Parallel Sysplex で使用できますが、マージがサポートされるのは 1 つの z/OS イメージ内のみです。

カップリング・ファシリティ・ログ・ストリームは、一時的な保管のためのカップリング・ファシリティと、より永続的な保管のための DASD ログ・データ・セットという 2 つのレベルのストレージにまたがります。ログ・ストリーム用のカップリング・ファシリティのスペースが満杯になると、データは DASD ログ・データ・セットにオフロードされます。カップリング・ファシリティ・ログ・ストリームには、複数のシステムからのデータを含めることができます。これにより、システム・ロガー・アプリケーションは Parallel Sysplex 全体のシステムからデータをマージできるようになります (ただし、CICS のシステム・ログ・データはマージされません)。カップリング・ファシリティ・ログ・ストリームと DASD 専用ログ・ストリームの主な違いは、一時ログ・データを保持するためにシステム・ロガーが使用するストレージ・メディアです。カップリング・ファシリティ・ログ・ストリームでは、ログ・データ用の一時ストレージはカップリング・ファシリティのリスト構造内に含まれています。DASD 専用ログ・ストリームでは、ログ・データ用の一時ストレージはシステムのローカル・ストレージ・バッファ内に含まれています。ローカル・ストレージ・バッファとは、システム・ロガーのアドレス・スペースに関連付けられたデータ・スペース域のことです。カップリング・ファシリティ・ログ・ストリームまたは DASD 専用ログ・ストリームを使用することも、両タイプのログ・ストリームを組み合わせることもできます。

z/OS System Logger の一般的な理解とその仕組みについては、IBM Redbooks® 資料 [Systems Programmer's Guide to: z/OS System Logger \(SG24-6898\)](#) を参照してください。CICS と z/OS System Logger の間の対話については、IBM Redbooks 資料 [Performance Considerations and Measurements for CICS and System Logger \(REDP-3768\)](#) を参照してください。CICS ログ・ストリームについて詳しくは、[ロギングおよびジャーナリング](#)を参照してください。

以下のガイドラインは特に DFHLOG ログ・ストリームと DFHSHUNT ログ・ストリームのものであり、多くの場合、CICS に関連したログ・ストリームの中でも使用頻度と重要性が最も高いものです。

- CICS DFHLOG ログ・ストリームと DFHSHUNT ログ・ストリームには単一の CICS 領域からしかアクセスできないため、ログ・ストリームを CF ログ・ストリームまたは DASDONLY ログ・ストリームとして定義するという選択肢もあります。
- CF ログ・ストリームの場合:
  - ご使用の環境で使用する CF 構造ごとのログ・ストリームの数は、注意深く計画する必要があります。各ログ・ストリームは、単一カップリング・ファシリティ構造のリソースの一部を使用するからです。CICS でのロガー環境の定義について詳しくは、[カップリング・ファシリティ構造を定義する](#)を参照してください。



- CICS 領域の 1 つが異常終了した場合にリカバリーできるようにするために、複数の CPC の CICS 領域によって接続されているログ・ストリームが各構造に含まれていることを確認します。
- 同じ構造に DFHLOG ログ・ストリームと DFHSHUNT ログ・ストリームを混在させないようにします。
- 同程度のアクティビティー・レベルのログ・ストリームを同じ構造に配置します。例えば、使用頻度の高いログ・ストリームを 1 つの構造に置き、それより使用頻度の低いログ・ストリームを別の構造に配置できます。
- すべてのログ・ストリームについて:
  - ステージング・データ・セットまたは z/OS System Logger 構造 (あるいはその両方) について、ログ・レコードが 1 次ストレージからオフロード・データ・セットに移動されないようなサイズにします。
  - 妥当な LS\_SIZE 値を指定します。デフォルト値である 2 つのトラックだけでは小さすぎます。
  - CI サイズが 24 KB になるオフロード・データ・セットのデータ・クラスを使用します。ただし、ステージング・データ・セットの CI サイズは 4 KB でなければなりません。
  - DFHLOG ログ・ストリームと DFHSHUNT ログ・ストリームに OFFLOADRECALL(NO) を指定します。
  - 新規 CICS 領域を開始するたびに新規ログ・ストリームの定義を明示的に指定しなくて済むよう、モデル・ログ・ストリーム定義を使用します。
  - 災害復旧のために DASD ミラーリングを使用し、災害時に CICS 領域の INITIAL スタートを行わない場合、ステージング・データ・セットを使用するようすべての CICS ログ・ストリームを定義します。
  - HIGHOFFLOAD を 80% から 85% に設定してください。これは、CICS システム・ログ・ストリーム DFHLOG と DFHSHUNT の、実験に基づいた適切な範囲です。LOWOFFLOAD パラメーター値の最も適した範囲は 40% から 60% です。値が小さすぎると、z/OS System Logger オフロード・プロセスがオフロード処理中に不要なログ・データの物理的な削除を完了した後に、ログ・データが 1 次ストレージから補助ストレージに物理的にオフロードされることがあります。逆に、値が大きすぎると、オフロード処理中に 1 次ストレージから解放されるスペースが少なくなるため、以降のオフロード処理の頻度が高まることがあります。
  - 順方向リカバリー・ログおよびユーザー・ジャーナルなどの汎用ログ・ストリームに関するガイドラインは、システム・ログに関するガイドラインと異なります。カップリング・ファシリティ構造にログ・データを保存するための要件はありません。このようなデータを通常使用する場合に必要なのは、データを迅速に DASD にオフロードするための小さな構造のみです。その場合、HIGHOFFLOAD を 80% から 85%、LOWOFFLOAD を 0 に設定してください。
  - 大きな構造を使用するとオフロードがまれにしか行われず、しかも長時間実行されます。それよりも、構造をできるだけ小さくし、小さなオフロードを頻繁に行うことをお勧めします。
  - z/OS System Logger 構造の自動変更を有効にしないでください。
  - テスト CICS システム (および通常使用以外の他のシステム) 用のログ・ストリームは、実動 CICS システムのログ・ストリームを入れる構造とは別の構造に入れます。
  - 端末専有領域 (TOR) 用のログ・ストリームを、アプリケーション所有領域 (AOR) 用のログ・ストリームを入れる構造とは別の構造に入れます。さらに、ファイル所有領域 (FOR) 用のログ・ストリームも、TOR 用および AOR 用のログ・ストリームを入れる構造とは別の構造に入れるようにします。
  - MAXBUFSIZE を 64 K よりやや小さく設定します (例えば、64000)。

## ログ延期インターバル (LGDFINT) 値

LGDFINT CICS SIT パラメーターは、ログ延期インターバルを定義します。インターバルは、強制書き込み要求のために CICS ログ・マネージャーが z/OS System Logger を呼び出すまでに待機する時間の長さです。これはミリ秒単位で指定されます。選択する値は、トランザクションの応答時間に影響を与える可能性があります。

大きな LGDFINT 値を指定すると、CICS がログ・ブロックに配置できるログ・レコードの数が増える可能性があります。z/OS System Logger を使用するコストも減ります (z/OS System Logger が呼び出される頻度が低くなるため)。しかし、より新しいプロセッサの速度があれば、ログ・ブロックをより頻繁に z/OS System Logger に送信することをお勧めします。そのため、デフォルトの LGDFINT 値は 5 ミリ秒に減られます。これよりも小さな値を使用することにはいくつかの利点があるかもしれませんが、5 より大きい値を使用することはお勧めしません。

## 活動キーポイント頻度 (AKPFREQ) 値

AKPFREQ キーワードは、CICS でアクティビティ・キーポイントへの書き込みを行う前に必要な、CICS システム・ログ・ストリーム出力バッファーへの書き込み要求の数を指定します。活動チェックポイントを取ることを結果の 1 つとして、バックアウトとリカバリーで不要になったログ・レコードがログ・ストリームから削除され、一時ストレージのスペースが解放されます。AKPFREQ 値を大きくすればするほど、ログ・レコードが一時ストレージに保存される時間が長くなります。

その結果、ログ・ブロックをオフロード・データ・セットに移動するか (これはお勧めしません)、一時ストレージのサイズを大きくする必要が生じ、CF と z/OS System Logger のデータ・スペースのストレージがさらに必要になります。別の影響として、AKPFREQ の値が大きければ大きいほど、再始動にかかる時間が長くなる可能性があります。

AKPFREQ 値を小さくすると、CICS はログの末端をより頻繁に切り取るようになります。末端をより頻繁にトリミングすることで、データをオフロード・データ・セットに移動するための入出力を発行するオーバーヘッドが発生することなく、オフロード・プロセスが LOWOFFLOAD ポイントに到達できるようになります。

## CICS カップリング・ファシリティ・データ・テーブル

CICS ファイルは、DASD でのデータ・セットの表現です。ファイルでデータ・テーブル・サービスを使用することを指定する場合、CICS は、ファイルが開かれたときにデータ・セットの内容をメモリー内の共用ストレージにコピーし、可能な時には常にそのコピーを使用します。共用データ・テーブル (CICS 保守データ・テーブルまたはユーザー保守データ・テーブル) では、z/OS イメージ全体で共用される z/OS データ・スペースに内容がコピーされます。カップリング・ファシリティ・データ・テーブルでは、内容はカップリング・ファシリティ構造にコピーされ、シスプレックス全体で共用されます。また、カップリング・ファシリティ・データ・テーブルでは、事前にデータ・セットからデータをロードせずに、実行時にデータをテーブルに取り込むこともできます。

カップリング・ファシリティ・データ・テーブル (CFDT) には、共用データ・テーブル (SDT) のパフォーマンス上のいくつかの利点があり、かつシスプレックス内のどこからでもデータ・テーブルにアクセスできるという柔軟性もあります。CFDT は、多くの点で共用ユーザー保守データ・テーブル (UMT) に似ています。そこで使用される CFDT および CF 構造は、CFDT サーバーによって管理されます。

CFDT の一般的な使用ではデータは長時間持続する、または永久的なものである必要はありませんが、トランザクション障害発生時やシステム障害発生時に CF データ・テーブルをリカバリーできるように定義することは可能です。データに対する更新は自動的にソース・データ・セットにハード化されず、CF または CF 構造での障害発生後に CFDT の内容は維持されません。必要に応じてデータ・テーブルに対する更新を DASD のソース・データ・セットにコピーすることはユーザーの責任です。CFDT 構造を CF の障害から保護するために、システム管理の二重化を使用することができます。

CFDT (そのデータは CF 構造内にある) は、共用データ・テーブル (そのデータはデータ・スペースにある) ほど高いパフォーマンスは発揮しませんが、データ・セットからのデータ・アクセスや CICS 領域間の機能シッパ要求よりは高いパフォーマンスを発揮します。

**注:** カップリング・ファシリティ・データ・テーブルになるようユーザー保守データ・テーブルを変更する場合、アプリケーションを変更する必要はありません。(z/OS データ・スペースまたは CF 構造での) データ・テーブルの場所のマッピングは CICS によって制御され、アプリケーション・プログラムはデータ・テーブルの場所を認識する必要はありません。特に CFDT に関連したアプリケーションの考慮事項については、[カップリング・ファシリティ・データ・テーブル](#)を参照してください。

カップリング・ファシリティ・データ・テーブル・サーバーのセットアップと実行について詳しくは、[カップリング・ファシリティ・データ・テーブル・サーバーのセットアップと実行](#)を参照してください。FILE リソースと CFDT について詳しくは、[カップリング・ファシリティ・データ・テーブル](#)を参照してください。

## CICSplex SM シスプレックス最適化ワークロード管理

CICSplex SM シスプレックス最適化ワークロード管理は、Parallel Sysplex 環境での CICS ワークロード・スループットを最大化するために CICS TS 4.1 で導入されました。

CICSplex SM シスプレックス最適化ワークロード管理は、ルーティング領域とターゲット領域が異なる CMAS によって管理されている分散ワークロードで最大限の効率性を発揮します。シスプレックスの最適

化ワークロード・ルーティングは、領域状況 (RS) サーバーごとの z/OS カップリング・ファシリティ・レベルで使用可能になります。各ターゲット領域の状況は CICS によって RS サーバーに送られ、そこでカップリング・ファシリティ・データ・テーブルに配置されます。各ルーティング領域は、シスプレックス内のすべてのターゲット領域の、トランザクション負荷と領域の正常性に関するリアルタイムに近い情報にアクセスすることができます。これによって、より効率的に WLM ルーティングを決定できるようになります。シスプレックスを使用した動的ワークロードの分散をモニターできるよう、追加の CICSplex SM WUI ビューが提供されています。シスプレックス最適化ワークロード管理は、単一のシスプレックス内に含まれる複数のワークロードに最適です。

CICSplex SM シスプレックス最適化ワークロード管理について詳しくは、[最適化された動的ワークロード・ルーティングの実装およびシスプレックス最適化ワークロード管理の学習パス](#)を参照してください。

## MRO に関する XCF

シスプレックスの異なる z/OS イメージにある CICS 領域の間での MRO リンクには XCF が必要です。

複数領域操作 (MRO) のための CICS 領域内通信 (IRC) では、同じシスプレックス内の領域内通信で XCF を使用できるようにすることで、この通信に VTAM を使用しなくても済むようにします。各 CICS 領域は、他の z/OS イメージ内の領域に現在接続されていない場合でも、IRC にログオンした時点で XCF グループに割り当てられます。別の z/OS イメージにある CICS XCF グループのメンバーが通信するとき、CICS は接続リソース定義に指定されたアクセス方式をオーバーライドして、この XCF を動的に選択します。

XCF グループの名前を XCFGROUP システム初期設定パラメーターで指定できます。XCFGROUP が指定されない場合、領域はデフォルトの CICS XCF グループ DFHIR000 のメンバーになります。同じシスプレックス内で異なるタイプの CICS 領域を実行する場合 (例えばテストと実動、または開発と実動など)、領域の集合に対してそれぞれ異なる XCF グループを使用することもできます。XCF グループで 2047 というメンバー数の制限に近づいており、相互通信が不要な領域のグループに CICS の領域を分割する場合、複数のグループを使用することもできます。

複数の CICS XCF グループを設ける必要がある場合は、以下の指針に従ってください。

- 実動領域を開発およびテスト領域とは異なる XCF グループに入れること。
- 必要以上に XCF グループを作成しないこと。
- XCF グループ間で領域を移動しないこと。

CICS 領域は XCF を使用して、同じ XCF グループ内の別の CICS 領域とのみ通信することができます。異なる XCF グループのメンバーは、同じ z/OS イメージ内にある場合でも、MRO を使用して通信することはできません。

XCF/MRO には、以下の利点があります。

- z/OS イメージ間の通信オーバーヘッドが少ない。このため、ISC リンクを使って z/OS システム間で通信するよりもパフォーマンスがかなり良い。
- VTAM テーブルを更新する必要がないため、ISC リンクの場合よりも接続リソース定義が容易である。
- 障害のある z/OS システムや CICS 領域のワークロードを代替のプロセッサとシステムによって継続できるようにすることで、高可用性を実現できる。
- z/OS システム間で CICS 領域を容易に移動できる。MRO の接続リソース定義が簡単で、かつ VTAM テーブルを更新する必要がないため、CICS 領域をある z/OS システムから別の z/OS システムへ移動することが容易になります。接続定義を CICS MRO から CICS ISC に変更する必要がなくなります (この変更は、新しい z/OS での CICS の始動がウォーム・スタートかコールド・スタートの場合にのみ可能です)。

XCF/MRO について詳しくは、[システム間複数領域操作 \(XCF/MRO\)](#) および [MVS システム間 MRO 定義](#)を参照してください。

## 一時記憶域プールの使用

一時記憶域は、複数のトランザクションに利用可能にする必要があるデータを格納するための、CICS の基本的な機能です。

一時記憶域のデータ項目はキューに保持され、キューの名前は、データを格納するプログラムによって割り当てられます。複数の項目が入っている一時記憶域キューは、小さいデータ・セットと考えることがで

き、そのレコードは、項目番号によって順次または直接のいずれかでアドレッシングすることができます。キューに単一項目しか入っていない場合には、名前付きのスクラッチパッド域と考えることができます。

一時記憶域は、CICS 領域内の主記憶域、VSAM データ・セット内の補助記憶域、または z/OS カップリング・ファシリティ内の共用一時記憶域プールに配置することができます。非並列シスプレックス環境では、一時記憶域キューは、リカバリー不能またはリカバリー可能として定義されます。リカバリー不能キューは通常、CICS 領域の仮想記憶域内に存在しています。これにより、I/O を DASD に行う場合よりもパフォーマンスが改善されます。しかし、CICS 領域が非アクティブになると、キューのデータが失われます。リカバリー可能な一時記憶域キューは、VSAM データ・セット内に配置する必要があります。これによって、アクセスの速度は低下します。また、キューへの更新はすべてログに記録する必要があるため、リカバリー可能キューの使用によるオーバーヘッドが高くなります。

MRO シナリオで一時記憶域データを 1 つの領域のタスクから別の領域のタスクに渡す必要がある場合、専用 CICS 領域 (キュー所有領域または QOR) を定義し、その領域に置かれているキューを使用する各 CICS アプリケーション所有領域 (AOR) に指定することができます。QOR は一時記憶域キューでデータを共有している 2 つのトランザクション間のアフィニティを除去する一方、トランザクションと同じ AOR 内に保持されているキューと比べてパフォーマンスが低下します。AOR と QOR の間の通信と関連付けられている機能シップによって余分のオーバーヘッドが発生し、QOR は単一障害点を構成します。QOR で障害が発生すると、そこに含まれるキュー内のデータはすべて失われます。

リカバリー不能な一時記憶域キューを共有する別の方法として、そのキューをカップリング・ファシリティ (CF) 構造に配置することをお勧めします。キューの場所は、DFHTST マクロ、または RDO TSMModel 定義でシステム・プログラマーによって定義されます。アプリケーションは常に同じ名前でキューにアクセスしますが、CICS は、キューの配置場所とそのアクセス方法を動的に判別します。

**注:** DFHTST マクロも CICS Transaction Server for z/OS によってサポートされますが、リソース定義オンライン (RDO) を使用して、そのマクロに相当する TSMODEL 定義を定義します。

キューは CF 構造に保持されるので、キューにあるデータは CICS 領域の障害から保護されます。すべての CICS 領域が停止した場合でも、一時記憶域構造のキューに書き込まれていたデータは、領域を再始動するとアクセスが可能になります。また、CF アクセス時間は通常マイクロ秒単位で測定されるので、構造内のデータへのアクセスは QOR を使用する場合と比べて格段に高速化されます。

**注:** 一時記憶域構造に配置できるキューはリカバリー不能キューだけなので、構造またはその構造が置かれている CF で障害が発生したときに構造の内容を障害点に復元する独自のメカニズムは CICS に備えられていません。構造内のデータの可用性がアプリケーションにとって重要な場合、システム管理の二重化を使用して構造を二重化することができます。一時記憶域構造に配置できるキューはリカバリー不能キューだけなので、定義により、そのタイプのキューを使用するアプリケーションがそのデータの消失に対処できるようにする必要があります。

AOR で実行されている CICS トランザクションは、一時記憶域キューの指定されたプールをサポートする一時記憶域サーバー・アドレス・スペースを介して一時記憶域構造のデータにアクセスします。シスプレックス内の各 z/OS イメージでは、CF で定義されるプールごとに 1 つの一時記憶域サーバー・アドレス・スペースをセットアップする必要があります。一時記憶域プールへのアクセスは常に、指定されたプールの一時記憶域サーバーに対するクロスメモリーの呼び出しによって実行されます。サーバーでサポートされる一時記憶域プールの名前は、一時記憶域サーバー・アドレス・スペース JCL の **POOLNAME** パラメーターで指定されます。サーバー・アドレス・スペースに割り振るバッファの数も指定する必要があります。バッファの最小数を、サーバーに接続できる CICS 領域ごとのデフォルト・バッファ数である 10 から増やすことで、バッファ待機のリスクを回避し、CF アクセス数を減らすことができます。適当なバッファ数を指定すると、最近使用されたキュー索引項目が記憶域に保持されます。READ または WRITE 要求の完了後、キュー索引情報はバッファに保持されます。キュー項目が読み取られるときにキュー索引項目の現行バージョンが記憶域にあれば、要求が CF に 2 回アクセスする必要はなく、1 回で済みます。

一時記憶域について詳しくは、一時記憶域を参照してください。一時記憶域の使用と、トランザクションのアフィニティ回避に関する考慮事項については、[プログラミング手法と類縁性を参照してください。](#) / xref> 共有一時記憶域の構造の定義について詳しくは、[一時記憶域データ共用のための一時記憶域プールの定義を参照してください。](#)



## 名前付きカウンター

名前付きカウンター機能は、アプリケーション・プログラムが Parallel Sysplex 環境で使用する、固有のシーケンス番号を生成するために使用されます。

名前付きカウンターは、名前付きカウンター・サーバーによって制御されます。このサーバーは、数字の各シーケンスを名前付きカウンターとして維持します。シーケンス番号が割り当てられるたびに、対応する名前付きカウンターが自動的に増分されます。この機能には、文書 (例えば注文書、送り状、発送メモ) に固有の番号を付けたり、カスタマー・ファイルでレコード番号を割り振るのに 1 ブロックの数字を確保するなど、いろいろな使い方があります。

単独の CICS 領域では、さまざまな方式を使用して固有番号の割り振りを制御することができます。例えば、CICS 共通作業域 (CWA) を使用すれば、アプリケーション・プログラムごとに更新される数値を保管することができます。ただし、CWA は CICS アドレス・スペースに対して一意的であるため、同じアプリケーションを実行している他の領域で共用することができません。CICS 共用データ・テーブルを使用すれば、このサービスを提供することができますが、CICS 領域はすべて、同じ z/OS イメージで実行される必要があります。名前付きカウンター機能は、名前付きカウンターをカップリング・ファシリティーで保守し、シスプレックス内の各 z/OS イメージで稼働している名前付きカウンター・サーバー経由でアクセスできるようにすることによって、他の方式が提示する共用上の問題をすべて解決します。これにより、Parallel Sysplex による CICS 領域はすべて、同一の名前付きカウンターにアクセスできるようになります。

名前付きカウンター・サーバーを使用する場合は、(次のカウンター値に割り当てる) 通常の各要求は、単一のカップリング・ファシリティー・アクセスのみを必要とします。これにより、同じ目的でファイルを使用する場合に比較すると、パフォーマンスが大幅に向上します。この観点からすると、名前付きカウンター・サーバーのパフォーマンスは、カップリング・ファシリティー・データ・テーブルより優れています。これは、カップリング・ファシリティー・データ・テーブルを更新するには少なくとも 2 つのカップリング・ファシリティー・アクセスが必要とされるためです。

**注:** 名前付きカウンター機能は、CICS プログラムを越えて拡張されます。名前付きカウンター・ファシリティーへのインターフェースを提供するプログラムには、バッチ・プログラムでも使用できる呼び出し可能インターフェースがあります。これは、シスプレックス内のどこでもバッチ・ジョブを実行でき、固有値を生成するためのハイパフォーマンス・メカニズムがあることを意味します。そのため、この機能を提供するために独自のメカニズムを構築する必要はありません。

名前付きカウンター・ファシリティーのセットアップと管理について詳しくは、[名前付きカウンター・サーバーのセットアップと実行](#)を参照してください。

### 名前付きカウンターのリカバリー

名前付きのカウンターは、CF の構造にのみ保管され、カウンターに対する更新は記録されません。したがって、名前付きカウンターを使用するアプリケーションは、場合によっては、名前付きカウンター構造の損失に対処するためのリカバリー・ロジックを実装する必要があります。

名前付きカウンター構造の損失から自身を保護するための最も簡単な方法は、それらの構造に対してシステム管理の二重化を使用することです。一般的に、構造へのアクセス比率は高くないため、特に構造が失われた場合の影響を考慮すると、要求を二重化するためのコストは通常、許容できます。また、名前付きカウンター構造が失われ、二重化されていない場合、その構造を使用する名前付きカウンター・サーバー・アドレス・スペースはすべて異常終了します。

名前付きカウンター・サーバー・アドレス・スペースの損失につながるシナリオはいくつかあります。

#### • 名前付きカウンター CF 構造の損失

名前付きカウンター構造が失われた場合、接続されているすべての名前付きカウンター・サーバー・アドレス・スペースが異常終了します。障害が発生する前に稼働していたシステムと同じシステムでサーバー・アドレス・スペースを、自動リスタート・マネージャー (ARM) を使用して自動的に再始動することを検討してください。最初に開始するサーバー・アドレス・スペースで、その構造が割り振られていないことが検出され、代替 CF で空の新規インスタンスが割り振られます。アプリケーションでサービスを再び使用する前に、構造の内容を復元するプロセスを用意しておく必要があります。

#### • 名前付きカウンター・サーバー・アドレス・スペースの異常終了

名前付きカウンター・サーバー・アドレス・スペースでの障害は、サービスの使用を試みるそのシステム上のすべてのアプリケーションに影響を与えます。ただし、名前付きカウンター構造の内容には影響を与えません。その場合、障害が発生したサーバー・アドレス・スペースを、障害が発生する前に稼働し

ていたシステムと同じシステムで、ARM を使用して自動的に再始動することを検討してください。追加のリカバリーは必要ありません。

- 名前付きカウンター・サーバー・アドレス・スペースが実行されているシステムの障害

名前付きカウンター・サーバー・アドレス・スペースが実行されているシステムで障害が発生した場合、IPL が実行された後と同じように、通常のシステムの開始プロセスでアドレス・スペースが開始されるようにしてください。他のシステムでアドレス・スペースを開始することに利点はありません。

カウンターの内容を再作成する場合は、テスト済みのメカニズムを使用することをお勧めします。例えば、構造を二重化していないときに、その構造が含まれる CF で障害が発生すると、その内容は失われます。構造を二重化した場合でも、電源障害のようなイベントでは、カウンターの内容がすべて失われる可能性があります。名前付きカウンターが、構造障害発生中に保存する必要のない一時情報 (例えば、構造障害時には保持されない可能性のある、アクティブ・プロセスの固有 ID など) にのみ使用されている場合、リカバリー技法は最小限で済みます。例えば、単にカウンターを標準初期値で再作成する場合などです。ただし、永続情報 (オーダー番号など) に名前付きカウンターが使用されている場合、カウンターの再作成には、カウンター値を再構成するための特有のアプリケーション・ロジックが必要となる場合があります。

名前付きカウンター構造が失われた後にカウンター値をリカバリーするために使用できる技法について詳しくは、[名前付きカウンターのリカバリー](#)を参照してください。

## Parallel Sysplex テクノロジーを間接的に利用する CICS 機能およびコンポーネント

Parallel Sysplex テクノロジーを間接的に利用する CICS 機能およびコンポーネントがいくつかあります。

CICS のこれらの要素は、他のソフトウェアによって実装される Parallel Sysplex テクノロジーを利用します。

### Db2 データ共用

Db2 データ共用は、CICS アプリケーション専有領域で実行されている CICS トランザクションが、同じ共用 Db2 データにアクセスできるようにします。

DB2CONN 定義には、Db2 サブシステムの名前、または CICS が接続する Db2 データ共用グループのグループ接続名が含まれます。特定のサブシステム名ではなくグループ接続名を使用することで、シスプレックス内のすべてのシステム上の CICS 領域が、データ共用グループ内のどの Db2 にも接続できるようにします。これは、z/OS カップリング・ファシリティーで定義されている Db2 構造を活用します。Db2 データ共用グループは、すべてが同じ表にアクセスできる複数の Db2 データ・マネージャーで構成されます。データ・マネージャーは、同じ z/OS 論理区画で実行することも、別個の z/OS 論理区画で実行することもできます。この実行方法により、CICSplex のさまざまな領域で同じ Db2 表のデータを共用できるようになります。

データ共用グループには、可用性の利点があります。グループ内の 1 つのデータ・マネージャーで障害が発生した場合に、グループ内の他のデータ・マネージャーが引き続きデータにアクセスできるからです。

Db2 グループ接続と呼ばれるファシリティーには、その他にも可用性に関する利点があります。

- Db2 グループ接続を使用しない場合、指定された Db2 データ・マネージャーを使用するよう各 CICS 領域が構成されます。
- Db2 グループ接続を使用する場合、データ共用グループ内のいずれかのデータ・マネージャーを使用するよう各 CICS 領域が構成されます。ただし、Db2 INDOUBT の解決を考慮する必要があります。

Db2 データ共用グループのサポートにより、さまざまな方法で CICSplex を構成できます。

- CICS Db2 グループ接続を使用しない、各論理区画に 1 つのデータ・マネージャー。  
区画内のすべての CICS 領域が、指定された Db2 データ・マネージャーに接続します。その Db2 マネージャーは、同じ論理区画で実行されている必要があります。
- CICS Db2 グループ接続を使用した、各論理区画に 1 つのデータ・マネージャー。

区画内のすべての CICS 領域は、同じ論理区画内で実行されている任意の Db2 データ・マネージャーに接続します。この構成は、前述の構成と似ていますが、1 区画内の CICS 領域で障害が発生した後に、デー



タ・マネージャーが同じデータ共用グループ内にある区画で再始動できるという点が異なります。これはデータ共用グループ内のデータ・マネージャーを使用して Db2 データにアクセスできます。

- CICS Db2 グループ接続を使用した、各論理区画に複数のデータ・マネージャー。

区画内の CICS 領域はそれぞれ、同じ論理区画内で実行されている Db2 データ・マネージャーの 1 つに接続します。この構成は、前述の構成と似ています。ただし、次の状況では異なります。

- ある区画の CICS 領域で障害が発生した後に同じ領域で再始動すると、別の Db2 データ・マネージャーに接続する可能性があります。このステップによって未確定リカバリー単位の解決が遅れる可能性があります。
- ある区画の Db2 データ・マネージャーで障害が発生すると、その区画の CICS 領域は同じ区画の別のデータ・マネージャーに接続して引き続き Db2 データにアクセスできます。

Db2 への接続を CICS が失ったときに、その Db2 に関連した進行中の INDOUBT 作業単位がある場合、Db2 への再接続が試行されます。しかし、初回の再接続試行が成功しなかった場合でも、その Db2 に関する未解決の作業単位があることを覚えています。CICS 領域を後で再始動し (あるいは Db2 接続が停止した後再始動し)、その Db2 が使用可能な場合、CICS はその Db2 に自動的に再接続します (CICS が Db2 サブシステムと同じ z/OS システムにある場合)。そのシステムに同じデータ共用グループの複数の Db2 サブシステムがある場合も同様です。

Db2 との初回の接続が失われたときの CICS の反応は、DB2CONN 定義の RESYNCMEMBER キーワードと STANDBYMODE キーワードによって制御されます。CICS は必ず再接続を 1 度試行します。しかし、その試行が失敗した場合、CICS はその Db2 が再びアクティブになるのを待つか、CICS と同じ z/OS システム上に別のメンバーが存在していれば、データ共用グループのそのメンバーへの接続を試行します。

同じ z/OS システムの同じデータ共用グループに複数の Db2 サブシステムがある場合は、RESYNCMEMBER キーワードを NO に設定することを検討してください。これは、CICS 領域が別の Db2 に接続して Db2 トランザクションの処理を続けられることを意味します。INDOUBT 作業単位は未解決のままとなり、CICS が元の Db2 に再接続するまで解決されません。ただし、適切なときに CICS 領域をドレーンし、Db2 接続を停止して再始動することもできます。その時点で INDOUBT 作業単位が解決されます。

Db2 への接続を初めて失った後のこの領域への Db2 トランザクションのフローを停止するには、[513 ページの『CICSplex SM トランザクションの異常終了のしきい値』](#)の説明に従って CICSplex SM の ABENDTHRESH 機能と ABENDCRIT 機能を使用してください。

CICS が Db2 に接続する方法について詳しくは、[概要: Db2 に CICS を接続する方法](#)を参照してください。詳しくは、[Db2 グループ接続機能の使用](#)を参照してください。

## IBM MQ 共用キュー

IBM MQ キュー共用グループ接続のサポートが CICS TS 4.1 以降のバージョンで提供されています。それよりも前のバージョンでは、特定の IBM MQ サブシステムの名前を指定する必要がありました。

Parallel Sysplex では、複数の IBM MQ キュー・マネージャーをキュー共用グループ (QSG) として構成できます。QSG のキュー・マネージャーは、共用キューと専用キューの 2 種類のローカル・キューをサポートしています。QSG のキュー・マネージャーは、1 つ以上の z/OS カップリング・ファシリティーで協力して共用キューを維持してアクセスします。こうして、すべてのキュー・マネージャーが共用キューを所有します。QSG 内のさまざまなキュー・マネージャーを同じ LPAR 内で実行することも、個別の LPAR 内で実行することもできます。QSG 内のいずれかのキュー・マネージャーに接続されている CICS 領域で実行されているアプリケーションは、QSG が所有する共用キューとの間で入出力を行うことができます。

CICS-MQ 接続に IBM MQ キュー共用グループ名を指定し、CICS が IBM MQ に再接続する際に特定のキュー・マネージャーを待機するのではなく、グループ内の任意の適格なキュー・マネージャーを使用するようにします。キュー共用グループを使用すると、IBM MQ に再接続する際の信頼性が高まり、CICS 領域と z/OS イメージにおけるこの面の CICS セットアップを標準化するのに役立ちます。

INITPARM システム初期設定パラメーターの DFHMQRPM オペランドで CICS-MQ 接続のデフォルト設定を定義するのではなく、MQCONN リソース定義を使用します。MQCONN リソース定義を使用すると、キュー共用グループ名を指定することも、特定のキュー・マネージャーの名前を指定することもできます。

接続にキュー共用グループ名を指定する場合、MQCONN リソース定義の RESYNCMEMBER 属性を使用して、CICS に適合する再同期アクションを選択できます。再同期は、Db2 のグループ接続機能の場合と同じように機能します。再同期が行われるのは、IBM MQ への接続が失われ、CICS が最後のキュー・マネー

ャーに対して未解決の作業単位を保持している場合です。CICS に同じキュー・マネージャーへの再接続を待機させるかどうか、あるいは CICS に同じキュー・マネージャーへの再接続を一度試行させるかどうかを選択できます。ただし、再接続の試行が失敗した場合は、グループ内の別の使用可能なキュー・マネージャーへの接続が行われます。キュー・マネージャーは、CICS 領域と同じ LPAR 上でアクティブであれば、CICS 領域に接続できます。

**EXEC CICS** コマンドと **CEMT** コマンド、または CICSplex SM を使用して、CICS-MQ 接続の開始と停止、および接続のすべての属性の変更を行うことができます。あるいは、CICS-MQ アダプターの操作に関して以前の方法を引き続き使用し、CICS と IBM MQ 間の接続の開始と管理を行うこともできます。

CICS-MQ アダプター制御パネルから CKQC トランザクションを使用するか、CICS コマンド行または CICS アプリケーションからこのトランザクションを呼び出すことができます。

CICS-MQ アダプターについて詳しくは、[仕組み: CICS-MQ アダプター](#)を参照してください。MQCONN リソースのセットアップについては、[MQCONN リソースの定義およびインストール](#)を参照してください。

## VSAM RLS

レコード・レベル共用 (RLS) は、多くの CICS 領域で実行中の多くのアプリケーション間で完全な更新能力を備えた VSAM データの共用を可能にする VSAM 機能です。

完全なデータ保全性を持つ別個の CICS 領域からの VSAM ファイルへの同時アクセスは、ファイル所有領域 (FOR) を実装することによって行えました。VSAM データ・セットへのアクセスはすべて、FOR から行われていました。CICS トランザクションはアプリケーション所有領域 (AOR) で実行され、VSAM ファイル要求は FOR へ機能シップされます。この実装には、次のようないくつかの問題があります。

- FOR は単一障害点である。
- FOR は CPU およびスループットのボトルネックになる可能性がある。
- AOR と FOR の間の機能シップは追加のオーバーヘッドである。

VSAM RLS は完全な更新の保全性を備え、Parallel Sysplex 内の 1 つ以上の z/OS イメージで実行されるアプリケーション間で、VSAM データを共用できるよう設計されています。CICS に FOR がなくても共用可能です。VSAM RLS は、別の LPAR で実行されている領域を含む、複数の CICS 領域へのトランザクション・インターフェースを提供します。これは、可用性のために重要です。複数の LPAR シスプレックスによって提供される回復力は、単一の LPAR より高いからです。VSAM RLS は、各 LPAR で SMSVSAM アドレス・スペースを使用します。領域内の AOR は、FOR の代わりに SMSVSAM アドレス・スペースに接続します。

### VSAM RLS のコンポーネント

DFSMS VSAM は、以下を提供します。

- N Way データ共用
- シスプレックス全体にわたるレコード・レベルのロック
- 共用バッファ・プールを持つ、各 z/OS システム内の SMSVSAM アドレス・スペース
- カタログ項目のリカバリー属性 - **LOG(NONE | UNDO | ALL)** と、**LOG(ALL)** の場合の **LOGSTREAMID**。リカバリーは、ファイル・プロパティではなく、データ・セットプロパティになります。
- 読み取り保全性オプション
- 順方向リカバリー・ログ・ストリームへのログ・レコードのマージ

CICS は、以下を提供します。

- バックアウトとリカバリーののための VSAM データ・セット変更のロギング。
- トランザクションの異常終了時や AOR の再始動時などにアンコミットされた変更のバックアウト。
- 順方向リカバリー・ロギングおよびファイル自動ジャーナリング。

CICS VSAM Recovery for z/OS (CICS VR) (または同等の順方向リカバリー製品) は、以下を提供します。

- イメージ・コピーと CICS ログのインベントリー。
- 自動化されたデータ・セット・リストアおよび順方向リカバリー。
- 必須 RLS コマンドの使用による、未解決のロックなどのアンバインドと再バインド。

## 非 RLS アクセス・モードから RLS アクセス・モードへのマイグレーションに関する考慮事項

非 RLS アクセス・モードから RLS アクセス・モードにマイグレーションするときに、ファイル自体のデッドロックを引き起こすおそれのある一連のファイル制御要求がトランザクションによって発行されると、AFCG 異常終了が発生する可能性があります。マイグレーションの円滑な実行に役立つ、以下の機能切り替えを有効にして、AFCG 異常終了を回避してください。

```
com.ibm.cics.rls.delete.ridfld=true
```

### RLS マイグレーションの後に AFCG 異常終了が発生する原因

非 RLS モードでファイルにアクセスしている場合に、トランザクションによって同じ制御インターバル (CI) に対して競合する要求が行われると、それが原因で AFCG 異常終了が発生します。例えば、LSR モードでファイルにアクセスしている場合に、トランザクションによって CI 内のレコードに対する **DELETE** 要求が発行され、その CI が同じトランザクションによって発行された **READ UPDATE** 要求の対象にもなっているなら、自己デッドロックが発生する可能性があります。ただし、**READ UPDATE** コマンドの後に非 RLS VSAM によって CI ロックが保持されないようにシステム初期設定パラメーター **CILOCK=NO** が有効になっている場合は、**DELETE** 要求時に自己デッドロック状態が発生しないため、AFCG 異常終了は返されません。しかし、RLS の場合は **CILOCK** が関係ないため、ファイルが RLS アクセス・モードにマイグレーションされた後は、上記の理由で AFCG 異常終了が発生する可能性があります。

### 解決策

マイグレーションの円滑な実行に役立つ、機能切り替え

`com.ibm.cics.rls.delete.ridfld=true` を有効にすると、RLS ファイルに対してローカル VSAM の **CILOCK=NO** の動作を実現できます。これにより、前述のような状況で **DELETE** 要求を正常に実行できるようになります。

機能切り替えを指定する方法については、[機能切り替えの指定](#)を参照してください。

### さらに学習したい方に

CICS での VSAM RLS サポートの設定について詳しくは、[VSAM RLS サポートに必要な定義](#)を参照してください。

CICS VR について詳しくは、[CICS VSAM Recovery for z/OS](#) を参照してください。

## ネットワーキング

CICS は、間接的に Parallel Sysplex ネットワーキングの技法と機能のいくつかを利用します。

### z/OS Communications Server 総称リソース

CICS システムは Communications Server 総称リソースとして登録できます。その後、その固有の APPLID または総称リソース名で認識できるようになります。これはいくつかの CICS 領域で共用され、すべての領域が同じ総称リソースに登録されています。

### TCP/IP ポートの共用

TCP/IP ポートの共用により、複数の CICS 領域で同じ IP スタックから同じポートを listen できるようになります。この IP スタックのポートにアドレッシングする要求は、各 CICS 領域のソケットの数と正常性に基いてアクティブ領域間で均等に分配されます。

### DVIPA およびシスプレックス・ディストリビューター

CICS 領域が別々の LPAR で実行される場合や、同じ LPAR 内の別の IP スタックに接続する場合には、DVIPA を使用できます。

シスプレックス・ディストリビューターによってシスプレックス内の複数のサーバー・インスタンスに IP ワークロードを分散させることができます。その際、クライアントやネットワーキング・ハードウェアに対して変更を加える必要はなく、接続のセットアップの際に遅延が生じることもありません。

シスプレックス・ディストリビューターと一緒にポートの共用を使用することで、同じ LPAR の CICS 領域や別々の LPAR の 1 つのシスプレックスに対して、結合された高可用性アプローチを提供することができます。

## Parallel Sysplex を容易にするその他の CICS 機能およびコンポーネント

Parallel Sysplex 内の CICS の実装を容易にする CICS 機能およびコンポーネントは他にもいくつかあります。

### CICSplex SM

CICSplex SM により、複数のイメージに渡る複数の CICS システムを単一システム・イメージ (SSI) として単一の制御点から管理できます。CICSplex SM は、複数の CICS 領域を単一の z/OS イメージで管理するのと同じように、1 つ以上の Parallel Sysplex に渡る CICS 領域を管理することができます。単一システム・イメージのプロビジョンにより、物理レベルではなく論理レベルで操作できるようになります。その際に、CICS リソースのスケールや場所を考慮する必要はありません。

CICSplex SM ワークロード管理は、その時点で最適な CICS 領域にトランザクションやプログラムを動的に経路指定します。その際、存在しているトランザクションのアフィニティーおよびワークロード分離が考慮されます。

CICSplex SM 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 相互通信

CICS 相互通信とは、ローカル CICS システムと リモート・システムの間の通信を指します。このリモート・システムは別の CICS システムである場合も、そうでない場合もあります。相互通信には 2 つのタイプがあります。複数領域操作とシステム間通信です。

#### 複数領域操作

CICS 複数領域操作 (MRO) を使用すると、同じ z/OS イメージまたは同じ z/OS シスプレックスで実行されている複数の CICS システムが相互に通信を行うことができます。MRO は、CICS システムと非 CICS システム (例えば、IMS) の間の通信はサポートしていません。MRO には、ネットワーク機能は必要ありません。

#### システム間連絡

CICS は、外部システムとの通信を可能にするために、SNA を利用したシステム間連絡 (ISC over SNA) および IP 相互接続 (IPIC) に相互通信機能を提供しています。

- CICS は、TCP/IP ネットワーク経由のシステム間連絡を提供しています。この形式の通信は、IP 相互通信または IPIC と呼ばれます。IPIC によって、ISC over SNA が提供するのと同様の機能とサービス品質が提供されます。
- SNA 経由のシステム間通信 (ISC over SNA) を使用することにより、CICS と非 CICS システム間、または同じ z/OS イメージやシスプレックス内にない CICS システム間で通信することができます。同じ z/OS イメージまたはシスプレックス内にある CICS 領域間でこれらの相互通信機能を使用することもできます。

CICS 提供のミラー・プログラム DFHMIRS は、スレッド・セーフ・プログラムとして定義されています。サポート対象の CICS 機能の場合、IPIC 接続を介してのみ、リモートの CICS 領域はスレッド・セーフ・ミラー・トランザクションを使用し、可能な場合はいつでも L8 オープン TCB で要求を実行します。IPIC 接続を使用してリモート CICS システム上の機能に対するコマンドを発行するスレッド・セーフ・アプリケーションの場合、TCB の交換が減少するため、他の相互通信方式に比べてアプリケーションのパフォーマンスが向上します。オープン TCB を使用することで、CICS 領域間のスループットが大幅に改善する可能性もあります。

CICS 相互通信について詳しくは、[相互通信入門](#)および [CICS 相互通信の紹介](#)を参照してください。

### IPIC 高可用性フィーチャー

IPIC 高可用性機能は、TCP/IP ネットワーク経由での CICS TS 領域のクラスターへの単一アクセス・ポイントを提供します。これにより、クラスター内の個々の領域で計画停止が発生した場合にも、計画外の停止



が発生した場合にも、全体としてクラスターへのアクセスを確実に回復することができます。詳しくは、[IPIC 高可用性フィチャー](#) を参照してください。

### 動的トランザクション・ルーティング

CICS にトランザクションを定義する際、リモート またはローカル としてそれらを記述できます。ローカル・トランザクションは常に、端末専有領域で実行されます。リモート・トランザクションは、IPIC、MRO、または APPC (LUTYPE6.2) ISC リンクによって端末専有領域に接続されている他の領域にルーティングすることができます。CICSplex 全体で動的トランザクション・ルーティングを使用すると、静的ルーティングと比べてパフォーマンスと可用性が改善されます。

動的トランザクション・ルーティングについて詳しくは、[動的トランザクション・ルーティング](#) を参照してください。動的ルーティング・プログラムの作成について詳しくは、[動的ルーティング・プログラムの作成](#) を参照してください。

CICSplex SM には、ワークロード・ルーティングとワークロードの分離を両方ともサポートする動的ルーティング・プログラムがあります。

### 分散プログラム・リンク

CICS 分散プログラム・リンクを使用すると、CICS アプリケーション・プログラムは、プログラム制御 LINK 要求をシップすることにより、他の CICS 領域にあるプログラムを実行することができます。これにより、要求されたプログラムの位置が不明な場合でも、アプリケーションを作成することが可能になります。詳しくは、[分散プログラム・リンク \(DPL\)](#) を参照してください。

### CICS Interdependency Analyzer for z/OS (CICS IA)

CICS Interdependency Analyzer (CICS IA<sup>®</sup>) は、CICS リソースをディスカバーして分析し、それらのリソースの間にある関係を識別するために使用する生産性向上ツールです。

CICS IA は、CICS システム内の主要なリソース間のランタイム関係を動的に検出します。これは、API コマンドと SPI コマンドのアプリケーションをオプションの Db2、IMS、IBM MQ、および COBOL 呼び出しと一緒にモニターすることによって行われます。これによって、アプリケーションの全体像と、対話や参照されるリソースがその相互関係と共に把握できます。CICS IA は、実行効率全体を向上させるためのスレッド・セーフの考慮事項について、アプリケーションを分析するうえで役立ちます。CICS IA は、アフィニティーについてアプリケーションを分析するうえで役立ちます。これは、Parallel Sysplex への CICS アプリケーションのロールアウトに影響を与える可能性があります。

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

スレッド・セーフ・プログラムについて詳しくは、[スレッド・セーフ・プログラム](#) を参照してください。

アフィニティーについて詳しくは、[類縁性および 509 ページの『アプリケーションのアフィニティー』](#) を参照してください。

### CICS Deployment Assistant for z/OS (CICS DA)

CICS Deployment Assistant (CICS DA) は、実行中の CICS 領域、および Db2 や IBM MQ などの他のサブシステムへの接続をディスカバーするために使用できます。また、CICSplex インフラストラクチャーの構築やワークロード管理インフラストラクチャーのセットアップの支援も行います。

CICS DA について詳しくは、[CICS Deployment Assistant for z/OS](#) を参照してください。

## Parallel Sysplex への CICS のマイグレーションの計画

Parallel Sysplex 環境から十分恩恵を受けられるようにするために、CICS アプリケーションの実行方法に変更を加える必要が生じる場合があります。

Parallel Sysplex の恩恵を最大限受けられるようにするために重点を置くべき分野が主に 2 つあります。

- アフィニティー

CICS トランザクションでは、1 つの場所から別の場所へデータを渡すために、多くの手法を使用します。技法によっては、データを交換するトランザクションを同じ CICS 領域で実行しなければならないため、

トランザクションの動的ルーティングが制約されます。複数のトランザクションが、そのような制限が課される方法でデータを交換する場合、それらの間にはアフィニティーがあると言います。アフィニティーが存在すると、トランザクションの動的ルーティングやワークロード・バランシングに制限が加えられ、アプリケーションの可用性に悪影響を及ぼすおそれがあります。可能な場合には、アフィニティーを避けるべきです。

- ワークロード・ルーティングおよび管理

Parallel Sysplex におけるワークロード・ルーティングおよび管理の目標は、アプリケーションの応答時間や可用性の目標を達成しつつ、プロセッサ・キャパシティーを最適化することです。これを達成するには、システムおよびサブシステムの可用性と負荷に関する最新の情報に基づいて、最も適切な場所にワークロードを動的にルーティングする必要があります。

## アプリケーションのアフィニティー

CICS 用の Parallel Sysplex の利点を最大限活用するには、アフィニティーをできるだけ除去する必要があります。トランザクションが単一の CICS 領域の可用性と結び付けられることを望まないからです。

可用性の目標を達成するには、CICS 領域で計画的停止が発生した場合にも計画外の停止が発生した場合にもアプリケーションの可用性を保守できなければなりません。すべてのアフィニティーを除去し、複数の領域に同じ機能を提供することで、どの CICS 領域がアクティブか非アクティブかに関係なくすべての CICS トランザクションにサービスを提供することができます。

アフィニティーには 3 つのカテゴリーがあります。

- トランザクション間のアフィニティー

トランザクション間のアフィニティーは、複数の CICS トランザクションの間のアフィニティーです。このアフィニティーは、お互いに情報をやりとりするための手法を使用するトランザクション間、またはお互いのアクティビティーを同期化するための手法を使用するトランザクション間で生じます。その場合トランザクションは、同一の CICS 領域で実行される必要があります。

トランザクション間のアフィニティーは、トランザクションの動的ルーティングに制限を加えるもので、次のような状況で生じる可能性があります。

- 1 つのトランザクションが終了し、そのトランザクションと同じ CICS 領域で実行した場合にのみ 2 番目のトランザクションがアクセスできる場所に状態データが残る場合。
- 最初のトランザクションが、その最初のトランザクションが実行されている間に 2 番目のトランザクションがアクセスするデータを作成する場合。この作業を安全に実行するために、最初のトランザクションは通常、イベントを待ちます。そのイベントは、最初のトランザクションが作成したデータを 2 番目のトランザクションが読み取ったときに、2 番目のトランザクションによって通知されます。この同期手法を使用する場合、両方のトランザクションが同一の CICS 領域で実行されている必要があります。

- トランザクションとシステム間のアフィニティー

トランザクションとシステムの間のアフィニティーは、あるトランザクションと特定の CICS 領域の間のアフィニティーです。CICS 領域のプロパティーの問い合わせまたは変更を行うトランザクションによって生じます。別のトランザクションではなく、特定の CICS 領域に対してアフィニティーのあるトランザクションは、動的トランザクション・ルーティングに適格ではありません。通常、それらのトランザクションは、**EXEC CICS INQUIRE** または **SET** などの CICS SPI コマンドを使用するか、グローバル・ユーザー出口プログラムに依存しています。

- 作業単位

作業単位のアフィニティーは、共用リソースを更新する同じトランザクション (UOW) の複数の要求が存在する場所です。例えば、Db2 行や VSAM レコードなどです。

トランザクションのアフィニティーによって生じる動的ルーティングの制限は、アフィニティーの存続時間と有効範囲によって異なります。理想的な状態は、トランザクションのアフィニティーがまったく存在せず、利用可能なターゲット領域の選択に制限がないことです。しかし、トランザクションのアフィニティーが存在する場合でも、次の要因によって決定されるこれらのアフィニティーの範囲には限界があります。



- アフィニティーの関係: 最も重要なのは、グローバルな LU 名、およびユーザー ID です。トランザクションのインスタンスのターゲット領域を動的ルーティング・プログラムが選択する方法はこれらの要素によって決まるからです。
- アフィニティー 存続時間: アフィニティーの終了時間が決まります。

アフィニティーをできるだけシステムから除去してください。アフィニティーによって、複数システムでアプリケーションのインスタンスを実行できなくなる可能性があります。一部の必要なリソースが使用できなくなるためです。CICSplex 環境からすべてのアフィニティーを除去すると、CICSplex の他の領域でアプリケーションを実行し続けることが可能になり、ユーザーに停止を意識させないようにすることができます。存在しているアフィニティーを調べるときは、単一のアプリケーションを分離して見るのではなく、すべてのアプリケーションを考慮することが重要です。1つのアプリケーションによって他のアプリケーションに制限が課される可能性があるからです。

CICS には、既存のアフィニティーに対処するうえで役立つ代替策がいくつかあります。その多くは、アプリケーションに変更を加えることなく使用できます。

CICSplex のアフィニティーの影響について詳しくは、IBM Redbooks の資料 [Parallel Sysplex Application Considerations \(SG24-6523\)](#) を参照してください。CICS Interdependency Analyzer を使用してアフィニティーを識別する方法について詳しくは、IBM Redbooks 資料 [IBM CICS Interdependency Analyzer \(SG24-6458\)](#) を参照してください。CICS Interdependency Analyzer について詳しくは、[CICS Interdependency Analyzer for z/OS](#) を参照してください。アフィニティーの詳細については、[類縁性](#)を参照してください。

CICS ENQ 機能を使用することによってアフィニティーが導入されることもあります。

### CICS ENQ

CICS アプリケーションは CICS ENQ/DEQ 機能を使用して、指定されたリソースへのアクセスを直列化することができます。

当初、直列化の有効範囲は単一の CICS 領域内でした。トランザクションでリソースに対する排他的アクセスが必要な場合には、ENQ コマンドを発行し、必要な変更を加え (誰もそのリソースに対して処理を行えないことが分かっている)、その後 DEQ を発行して直列化を解放していました。そのリソースに関心があるすべてのトランザクションが同じ CICS 領域で実行されている間は、この処置で問題ありません。しかし、この処置によって、それらすべてのトランザクションと単一の CICS 領域の間にアフィニティーが生まれます。そうすると、別の CICS 領域のトランザクションがそのリソースを変更しようとする場合に、元の領域のトランザクションによって保持されている直列化が認識されなくなってしまうます。

このアフィニティーに対する取り組みとして、CICS では、選択したリソースが GRS に外部化され、シスプレックス全体のリソースとして扱われるよう、システム・プログラマーがその直列化要求を指定することができます。これらのリソースは、ENQMODEL 定義を使用して定義されます。シスプレックス全体のリソースとなるよう ENQMODEL を使用してリソースが定義されていると、そのリソースに関心があるトランザクションを多くの CICS 領域で実行できるようになります。そして、ENQ 要求を発行するプログラムから発生するすべての直列化が認識されるようになります。重要なのは、アプリケーションの変更が必要ないという点です。リソースがローカル・リソースとして扱われるか、シスプレックスのリソースとして扱われるかは、CICS レベルで制御されるからです。

グローバル ENQ および DEQ の使用について詳しくは、[ENQMODEL リソース定義による ENQ および DEQ コマンドの使用](#)を参照してください。ENQMODEL 定義のセットアップについて詳しくは、[ENQMODEL リソース](#)を参照してください。

## CICS ワークロード・ルーティングおよび管理

CICS には、Parallel Sysplex 内でトランザクションを自動かつ効率的に分散するための機能がいくつか備えられています。

作業が CICS に届くと、いくつかのことが生じる可能性があります。

- 要求全体を単一の CICS 領域で実行できます。
- しかし、通常、高可用性の Parallel Sysplex では、一部あるいは全部の要求がルーティングされます。

CICS アプリケーションからアフィニティーをすべて除去すると、どの使用可能な CICS 領域でもトランザクションを実行できるようになります。ただし、可用性を維持するには、最も適切な CICS 領域にトランザクションをできるだけ素早く自動でルーティングする必要があります。

Parallel Sysplex での CICS ワークロードの管理には、主に 3 つのコンポーネントが関係しています。

- CICSplex SM

CICSplex でワークロード管理とワークロード分離の両方をサポートするために必要な CICS 固有のワークロードの指定と定義を CICSplex SM に提供します。これらのワークロードの指定と定義により、CICSplex ワークロードを構成するトランザクションと、作業の処理に使用できるアプリケーション所有領域に関する必要なすべての情報が CICSplex SM に提供されます。

- CICS 動的ルーティング・プログラム

CICSplex SM のサービスを使用して適切なアプリケーション所有領域にトランザクションをルーティングできる動的トランザクション・ルーティング・プログラムが必要です CICSplex SM を実装する場合、それによって提供される動的トランザクション・ルーティング・プログラムを使用できます。

- z/OS ワークロード・マネージャー

ワークロード・マネージャーにサービス定義を提供します。このサービス定義は、最終的にユーザー・トランザクションを処理する CICS 領域および関連サブシステムで必要なリソースを使用できるようにするために z/OS で必要になります。z/OS サービス定義は、シスプレックス全体のワークロードをカバーする必要があります。そうすることで、オンライン、バッチ、システム関連などすべてのタイプの作業に対して z/OS でリソースを割り振れるようになります。

サービス定義は主に、目標指向のワークロード管理に必要です。

### CICS トポロジーとセッション・バランシング

Parallel Sysplex の可用性を最大化するには、初期セッション接続要求を受け入れることができる複数の CICS 領域が必要です。

複数の端末専有領域 (TOR) を提供し、1 つの TOR を停止する必要がある場合にユーザーが継続してログオンできるようにします。1 つの CPC がダウンした場合に、ユーザーが他の CPC の TOR を経由して CICS に引き続きアクセスできるよう、可能な場合には CICSplex 内のすべての CPC (中央演算処理複合システム) に TOR を設置します。理想的なのは、各システムに複数の TOR を設置して、1 つの TOR がダウンした場合にシスプレックスの他のメンバー、および TOR がダウンしたシステムの CICS セッションがいくつか存続するようにすることです。

複数のセッション所有領域を持つだけでは不十分です。使用可能な領域全体で新しいログオン要求のバランスが自動的に取られるメカニズムも必要です。TCP/IP 接続の場合は、シスプレックス・ディストリビューターを使用します。シスプレックス内での TCP/IP ネットワーキングについては、[「z/OS Communications Server IP 構成ガイド」](#)の『[シスプレックス内の TCP/IP](#)』を参照してください。SNA の場合は、VTAM 総称リソースを使用します。

### 動的トランザクション・ルーティング

Parallel Sysplex を最大限活用するには、最も適切な領域にトランザクションを動的にルーティングする必要があります。

従来の CICS MRO モデルは、CICS 領域の 1 つのセットがユーザーとのセッションを所有し、ユーザーによって開始されるトランザクションが実行のために別の領域に送信される、というものでした。どのターゲット領域を選択するかに関する決定は、動的トランザクション・ルーティング出口によって制御されます。この役割を果たすために独自のプログラムを作成することも、CICSplex SM によって提供される出口を使用することもできます。

動的トランザクション・ルーティングについて詳しくは、[動的トランザクション・ルーティング](#)を参照してください。また、独自の動的ルーティング・プログラムの作成について詳しくは、[動的ルーティング・プログラムの作成](#)を参照してください。CICSplex SM 動的ルーティングについては、[CICSplex SM ワークロード管理](#)で説明されています。

### CICSplex SM ワークロード管理

CICSplex SM ワークロード管理は、その時点で最適な CICS 領域にトランザクションやプログラムを動的に経路指定します。その際、存在しているトランザクションのアフィニティーおよびワークロード分離が考慮されます。

CICSplex SM ワークロード管理は、ワークロードのターゲットを選択するときに、3 つの主な基準を考慮に入れます。

- ルーティング

以下の想定のもとに、作業を配布するための最も適切なターゲット領域を決定するプロセス。

- 作業は、ターゲット領域のいずれかに送信されます。
- 作業によって配布に対して制約が課されることはありません。
- z/OS ワークロード・マネージャーとの協力における、応答時間目標を達成する可能性が最も高い領域。

- ワークロード分離

指定された作業を特定のターゲット領域のセットに配布するプロセス。

- 例えば、アプリケーション・タイプによる作業の分離。

- アフィニティー管理

既存の依存関係が宣言されているワークロードのルーティング。

- 例えば、単一の領域に維持されている状態データ。

可能であれば、アフィニティーはすべて 除去します。

ワークロード管理では、使用可能な CICS システムを最大限に活用し、スループット、パフォーマンス、および領域の正常性を改善するために、一群のターゲット領域の中から、各ターゲット領域の可用性、パフォーマンス、および負荷に基づいてトランザクションやプログラムのルーティング先を決定します。

CICSplex SM ワークロード管理機能は、EYU9XLOP というユーザー置き換え可能な動的ルーティング・プログラムを使用して、要求側領域で発生する作業要求を実行するターゲット領域を識別します。決定が行われる領域は、ルーティング領域と呼ばれます。要求側領域とルーティング領域の両方が同じ領域になる可能性があります。CICSplex SM による動的ルーティング、および動的ルーティング・プログラム EYU9XLOP について詳しくは、[CICSplex SM を使用した動的ルーティング](#)を参照してください。

ターゲット領域は、次の 4 つのアルゴリズムのうちの 1 つを使用して選択できます。

- キュー・アルゴリズムは、作業要求を処理するキューが最短で、ストレージ不足などの状態による影響が最も少なく、トランザクションが異常終了する可能性が最も低いターゲット領域にルーティングします。
- ゴール・アルゴリズムは、z/OS ワークロード・マネージャーで事前定義されているトランザクションの応答時間目標に到達する可能性が最も高いターゲット領域にトランザクションを経路指定します。CICS TS 4.1 より前のバージョンでは、CICSplex SM は、平均応答時間目標のみをサポートしていました。しかし、CICS TS 4.1 以降では、パーセンタイルの応答時間目標も使用できるようになりました。
- リンクに依存しないキュー (LNQUEUE) アルゴリズムは、ルーティングとターゲット領域間の接続のタイプが考慮されない点を除き、キュー・アルゴリズムと同等です。
- リンクに依存しないゴール (LNGOAL) アルゴリズムは、ルーティングとターゲット領域間の接続のタイプが考慮されない点を除き、ゴール・アルゴリズムと同等です。

CICSplex SM 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。CICSplex SM ワークロード管理の構成について詳しくは、[ワークロード管理の構成](#)を参照してください。

### CICS ワークロード・ルーティングの最適化

Parallel Sysplex では、カップリング・ファシリティーを使用して領域の状況に関する情報を保管できます。これにより、ルーティングに関する決定を最新の情報に基づいて行えるようになります。

CICS TS 4.1 より前のバージョンでは、各システムの CICSplex SM は、そのシステムで管理していた領域に関するリアルタイム情報を保持していました。しかしながら、他のシステムとの情報の交換は、一定の間隔でのみ行われていました。これは、トランザクションのルーティング先の決定に使用されていた情報が古くなっている可能性があることを意味しています。CICS TS 4.1 以降、ワークロード管理機能がより効率的になったためにワークロード・スループットが向上しています。CICS 動的ワークロード管理で、カップリング・ファシリティーを利用して、CICS から直接通知された現在の領域状況情報が格納されるようになりました。これにより、ルーティング領域とターゲット領域が同じ z/OS システムにあるかどうかに関係なく、最新の情報に基づいてルーティングの決定を行えるようになります。

この機能を有効にするには、CICSplex 内のすべての領域を CICS TS 4.1 以降で実行する必要があります。各システムに領域サーバー・アドレス・スペースが必要です。CICSplex SM CSYSDEF ビューを使用し、



WLM 最適化の使用可能化を ENABLED に指定することで、最適化モードで実行中として領域を定義する必要があります。このビュー内の属性、およびこのリソースのすべての属性とアクションについては、を参照してください。

CICSplex SM シスプレックス最適化ワークロード管理について詳しくは、[最適化動的ワークロード・ルーティングの実装およびシスプレックス最適化ワークロード管理の学習パス](#)を参照してください。

### CICSplex SM トランザクションの異常終了のしきい値

CICSplex SM を使用して、トランザクションの異常終了率が高い領域を避けて作業を自動的にルーティングすることができます。

複数の領域がそれぞれのトランザクションを実行できる CICSplex では、CICSplex SM を使用して、トランザクションの異常終了率が高い領域を避けて作業を自動的にルーティングすることができます。例えば、Db2 サブシステムが異常終了した AOR があるとします。いずれかの Db2 トランザクションがその AOR にルーティングされると、DB2® が再び使用可能になるまで異常終了します。

このような状況では、CICSplex SM **ABENDTHRESH** パラメーターと **ABENDCRIT** パラメーターを使用して、障害が発生する可能性がある領域を避けてトランザクションをルーティングすることができます。これらのパラメーターについて詳しくは、[CICSplex SM の管理](#)を参照してください。

### z/OS ワークロード・マネージャー: 領域とトランザクションの目標

z/OS ワークロード・マネージャーは、領域の目標またはトランザクションの応答時間の目標に向けて CICS の作業を管理することができます。使用する目標を選択することができます。

領域の目標に向けて管理することを選択する場合、z/OS ワークロード・マネージャーは、JES または STC 分類規則の下で CICS アドレス・スペースに割り当てられるサービス・クラスの目標を使用します。トランザクションの目標に向けて管理することを選択する場合、z/OS ワークロード・マネージャーは、CICS サブシステムの分類規則の下でトランザクションまたはトランザクションのグループに割り当てられるサービス・クラスの目標を使用します。JES または STC 分類規則を使用して作業を行う際、使用するモードを選択することができます。

トランザクションの目標に向けて管理する場合、z/OS ワークロード・マネージャーは、トランザクションのリソース割り振りを直接的には管理しません。その代わりに、トランザクションを実行できる CICS 領域に適切なリソースを割り当てるための計算を行います。領域でトランザクションの目標と応答時間の目標が異種混合している場合には、この作業効率が低下することがあります。この場合、領域の目標に向けて管理を行うことで作業効率が改善される可能性があります。

場合によっては、単一作業要求の処理のために CICS 領域で複数のトランザクションが必要になります。例えば、CICS プロバイダー領域でインバウンド SOAP 要求を処理するために、トランザクション ID が異なる最大 4 つのトランザクションが必要になる可能性があります。トランザクションの目標を使用するか、領域の目標を使用するかを決める際には、この点を考慮に入れてください。

オプションで、RMF レポートを使用してトランザクションの応答時間を報告することができます。ただし、引き続き領域レベルで CICS の作業を管理します。これについては、IBM Redbooks 資料 [System Programmer's Guide to: Workload Manager](#) (SG24-6472) の『11.5 Reporting』というタイトルのセクションで説明されています。

### CICS と自動リスタート・マネージャー

z/OS 自動リスタート・マネージャー (ARM) は、アドレス・スペース障害またはシステム障害が発生した後に、アドレス・スペースを自動的に再始動することができます。領域障害が発生した後、ARM を使用して CICS 領域を再始動することができます。

システム障害発生後に代替システムで CICS を再始動するために ARM を使用するかどうかは、接続されているサブシステム (例えば Db2) を別のシステムで再始動するために ARM を使用するかどうかによって異なります。シスプレックスのあらゆるメンバーに各 CICS 領域タイプのインスタンスがいくつかある場合、トランザクション処理機能を提供するという観点から CICS 領域を別のイメージで開始する必要はないはずです。ただし、接続済みサブシステムを別のシステムで再始動する場合、そのサブシステムに接続されていた CICS 領域があれば、INDOUBT 作業単位を解決するために、それも再始動する必要が生じる可能性があります。これを行わないと、その作業単位に関連付けられたロックは解放されず、そのリソースの使用を試みる他のトランザクションに影響を与える可能性があります。

ARM では、システム 障害発生時に同じシステムですべてを再始動する必要があるアドレス・スペースのグループを指定できます。ARM を使用して Db2 をどこか別の場所で再始動する場合、その Db2 と関連付けられている CICS 領域を同じグループに組み込みます。

## 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涉外

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](http://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]

- アプリケーション
  - バンドル [285](#)
- アプリケーション所有領域 [3](#)
- アプリケーション専有領域 (AOR) [352](#)
- アプリケーション・バンドル [283](#)
- 暗号化
  - データ・セット [103](#)
- 一時記憶
  - 一時記憶域サーバーの開始 [166](#)
  - 順方向リカバリ [471](#)
  - バックアウト (backout) [471](#)
  - リカバリ可能性 [471](#)
  - TS データ共用サーバー [166](#)
- 一時記憶域サーバー
  - 始動ジョブの例 [170](#)
- 一時記憶域データ共用
  - TS データ共用のための TS プールの定義 [168](#)
- 一時記憶域データ共用サーバーの概要 [166](#)
- 一時データ (区画外) データ・セット
  - 定義 [54](#)
- 一時データ (区画内) データ・セット
  - オープンの失敗 [53](#)
  - その他の考慮事項 [53](#)
  - 定義 [52](#)
  - 複数のエクステンントおよびボリューム [53](#)
- 一時データ、区画外
  - 回復 [469](#)
- 一時データ、区画内
  - 順方向リカバリ [469](#)
  - バックアウト (backout) [466](#)
  - リカバリ可能性 [466](#)
- 一時データ・キュー
  - 既存の定義の置換 [332](#)
  - 無効にする [333](#)
- 一時データ・キュー定義
  - インストール [332](#)
- 一時データ定義 [417](#)
- 一時データを端末に書き込むサンプル・プログラム (DFH \$TDWT) [51](#)
- イベント処理 [285](#)
- インストール
  - パラメーター・リスト [343](#)
  - MVS の考慮事項 [344](#)
- インストールの検証 [481](#)
- ウォーム・リスタート
  - 自動インストールされた項目の非リカバリ [253](#)
  - テーブルの再作成 [324](#)
- 永続メッセージ [471](#)
- 永続メッセージのサポート [473](#)
- オンライン・リソース定義 (RDO)
  - コマンド・ログ [37](#)
  - CICS システム定義データ・セット (CSD) (CICS system definition data set (CSD)) [21](#)

## [カ行]

- カード読取装置 [318](#)
- カード読取装置およびライン・プリンター [318](#)
- 開始タスク、CICS [152](#), [156](#)
- 外部 CICS インターフェース (EXCI)
  - 接続の状態の照会 [403](#)
  - 接続の定義 [400](#)
  - ユーザー置換可能モジュール (DFHXCURM) [404](#)
- 回復
  - 逆方向 [460](#), [466](#), [471](#)
  - 物理 [467](#)
  - リカバリ不要 [469](#)
  - 論理 [466](#)
- カタログ
  - プログラム自動インストール [266](#)
- カタログ式プロシージャ
  - 開始タスクとしての CICS の始動 [152](#), [156](#)
  - バッチ・ジョブとしての CICS の開始 [156](#)
- カップリング・ファシリティー
  - ストレージの管理 [232](#)
- カップリング・ファシリティー・データ・テーブル
  - サーバーの開始 [180](#)
  - リソースの定義 [183](#)
- カップリング・ファシリティー・データ・テーブル・サーバーの概要 [181](#)
- カップリング・ファシリティー・リスト・ストラクチャー
  - カップリング・ファシリティー・データ・テーブル [183](#)
- 管理テーブル
  - 名前指定と接尾部指定 [275](#)
- 管理テーブルの接尾部指定 [275](#)
- 管理テーブルの名前指定 [275](#)
- 機動的なサービス・デリバリー
  - 作成 [357](#)
- 機能
  - 一時記憶 [48](#)
  - 汎用トレース機能 (GTF) [45](#)
- 逆方向リカバリ (backward recovery) [460](#), [466](#), [471](#)
- ギャップ [337](#)
- 許可ライブラリー [148](#)
- 緊急時再始動
  - 自動インストールされた項目の一時的なリカバリ [253](#)
  - テーブルの再作成 [324](#)
- 区画外一時データ
  - CSSL、および CICS が使用する他の宛先 [149](#)
- 区画外一時データ・キュー [149](#)
- 区画外データ・セットのリカバリ
  - 出力データ・セット [470](#)
  - 入力データ・セット [469](#)
- 区画内一時データ
  - 順方向リカバリ [469](#)
  - バックアウト (backout) [466](#)
  - リカバリ可能性 [466](#)
- 区画内一時データ・キュー
  - 区画内データ・セットの定義 [52](#)
- 区分セット
  - 自動インストール [264](#)



クライアント/サーバー (client/server) [486](#)  
グローバル・カタログ  
  インストール時 [324](#)  
  再始動で使用 [130](#)  
  指定用の AMP パラメーター [150](#)  
  自動インストールあり [248](#)  
  リソース定義の [130](#)  
グローバル・カタログ・データ・セット (GCD)  
  スペース計算 [68](#)  
  説明 [66](#)  
  定義および初期設定のためのジョブ制御ステートメント [66](#)  
  バッファ・スペース [71](#)  
  CICS 実行のジョブ制御ステートメント [67](#)  
結果セット警告のカウント [377](#)  
現地時間の変更 [457](#)  
構成  
  機動的なサービス・デリバリー [351](#)  
  プラットフォーム [351](#)  
  REXX [477](#)  
高速パス・コマンド [419](#)  
コールド・スタート (cold start)  
  自動インストールされた TCT 項目の非リカバリー [253](#)  
  テーブル・エントリを除去するための使用 [324](#)  
  テーブルを再作成するための GRPLIST の使用 [324](#)  
コピーブック  
  DFH\$TCTS [318](#)  
コマンド  
  ロックされたグループに有効 [13](#)  
  CEDA DEFINE [315](#)  
  CEDA INSTALL GROUP(groupname) [315](#)  
  CEDA コマンドの同期点の基準 [36](#)  
  DEFINE TERMINAL CONSNAME(name) [321](#)  
  DEFINE TERMINAL CONSOLE(number) [322](#)  
  DFHCSDUP INITIALIZE [322](#)  
  LIST ALL OBJECTS [25](#)  
  MODIFY コマンド [320](#)  
  RDO CEDA INSTALL [66](#)  
  REPRO、IDCAMS を実行する場合 [84](#)  
コマンド・セキュリティ (command security) [416](#)  
コマンド・リスト・テーブル (CLT) (command list table (CLT))  
  CLT (コマンド・リスト・テーブル) [269](#)  
コマンド・レベル・インタープリター・トランザクション (CECI) [486](#)  
コマンド・ログ、RDO [37](#)  
コミット・プロセス [324](#)  
固有接続  
  定義 [400](#)  
コンソール  
  システム 初期設定パラメーターの入力 [129](#), [158](#)  
  自動インストール [258](#)  
  自動インストールに不適格 [251](#)  
  デバイス [320](#)  
  CICS に対する定義 [320](#)  
  CICS の始動に関するコンソール・メッセージ [158](#)  
  CICS マスター端末としての z/OS コンソール [320](#)  
  DEFINE TERMINAL CONSNAME(name) コマンド [321](#)  
  DEFINE TERMINAL CONSOLE(number) コマンド [322](#)  
  TSO ユーザー、コンソール・デバイスとして定義 [320](#)  
  TSO ユーザーの定義 [321](#)  
コンバーター (CICS ONC RPC)  
  提供された関数の定義 [428](#)

## [サ行]

サーバー・コントローラー  
  ユーザー ID [424](#), [435](#)  
サーバー・コントローラー (CICS ONC RPC)  
  定義 [415](#)  
再開  
  ウォーム  
    自動インストールされた項目の非リカバリー [253](#)  
    テーブルの再作成 [324](#)  
  緊急時  
    自動インストールされた項目の一時的なリカバリー [253](#)  
    テーブルの再作成 [324](#)  
サインオン・テーブル (SNT) (sign-on table (SNT))  
  OPIDENT オペランド [13](#)  
削除  
  インストール時の既存のエントリ [325](#)  
  システム・テーブルからのリソース 定義  
    自動インストールのログオフ時 [247](#)  
  ログオフ時の TCT 項目 (自動インストール) [247](#)  
参照する zFS 成果物 [297](#)  
サンプル  
  ディスクで補助データ・セットを定義するためのサンプル・ジョブ [77](#)  
  BDAM データ・セットを作成してロードするための JCL [88](#)  
  BDAM データ・セットをロードするためのデータ [89](#)  
  DFH\$TDWT (一時データを端末に書き込むプログラム) [51](#)  
  DFHSTART、サンプル始動プロシージャ [143](#)  
  DFHTCT5\$、サンプル TCT [318](#)  
サンプル・ジョブ・ストリーム  
  CICS startup [143](#)  
シーケンス番号  
  名前付きカウンターの参照 [215](#)  
時間変更 [457](#)  
事後初期設定 (PLT) プログラム  
  (初期設定プログラム)  
  使用 [470](#)  
システム管理機能  
  CICS 統計用 [45](#)  
システム管理再作成 [237](#)  
システム管理者 [487](#)  
システム管理の二重化 [238](#)  
システム管理プロセスの CICS サーバー・サポート [237](#)  
システム・グループ [1](#)  
システム・コンソール [320](#)  
システム 初期設定  
  代替 CICS の (XRF=YES)  
    START=STANDBY [156](#)  
  CICS が始動のタイプを決定する方法 [130](#)  
  START=AUTO [131](#), [155](#), [156](#)  
  START=COLD [132](#)  
  START=INITIAL [132](#), [155](#)  
システム 初期設定テーブル (SIT)  
  CICS に対するシステム 初期設定パラメーターの提供 [125](#)  
システム 初期設定テーブル (SIT) (system initialization table (SIT))  
  DFHSIT キーワードおよびオペランド [112](#)  
システム 初期設定パラメーター  
  オペレーターのコンソールから [126](#), [129](#), [159](#)  
  コンソールでの入力 [129](#), [158](#)

システム 初期設定パラメーター (続き)  
   指定方法 [105](#)  
   PARM ステートメントにおける指定 [146](#)  
   PARM パラメーターでの [126, 128](#)  
   PGAICTLG [266](#)  
   SYSIN データ・セットから [146](#)  
   SYSIN データ・セット内に [126, 128](#)  
 システム 初期設定パラメーターのオーバーライド  
   コンソールから [129, 158](#)  
   SYSIN データ・セットから [128](#)  
 システム・タスク [152, 156](#)  
 システム・ダンプ [79](#)  
 システム・データ・セット [41](#)  
 システム・テーブル、CICS  
   既存のエントリーの削除 (リソース定義の再インストール)  
   [325](#)  
 システム・テーブルからのリソース 定義の除去  
   コールド・スタート時 [324](#)  
   自動インストールのログオフ時 [247](#)  
   初期始動時 [324](#)  
 システムの始動  
   始動ジョブ・ストリーム (startup job stream) [143](#)  
 システム・プログラミング  
   EXEC CICS CREATE コマンド [6](#)  
   EXEC CICS CSD コマンド [6](#)  
 システム・ログ  
   ログ・テーブルの削除 [450](#)  
 システム・ログ・ストリームの定義  
   アクティビティー・キーポイント [448](#)  
   データ保全性の維持 [442](#)  
   モデル・ログ・ストリーム (カップリング・ファシリティ)  
   [443](#)  
   JOURNALMODEL [443](#)  
   MVS ログ・ストリーム [442](#)  
 シスプレックス [1](#)  
 事前設定セキュリティ [261](#)  
 持続セッション [322](#)  
 自動 TCTTE 削除 [254](#)  
 自動インストール  
   ジャーナル [268](#)  
   プログラム、マップ・セット、および区分セット [264](#)  
   モデル [245](#)  
   APPC 接続 [261](#)  
   IPIC 接続 [264](#)  
   MVS コンソール [258](#)  
   z/OS Communications Server 端末 [245](#)  
 自動インストール (autoinstall)  
   および Communications Server [247](#)  
   およびセキュリティ [261](#)  
   自動 TCTTE 削除  
   制御プログラム [245](#)  
   端末定義 [315](#)  
   端末リソース定義のインストール [315](#)  
   適切なデバイス [251](#)  
   トランザクション・ルーティングの制限 [247](#)  
   不適格なデバイス [251](#)  
   モデル [247](#)  
   リカバリーと再始動 [253](#)  
   ログオン・プロセス [247](#)  
   ATI への影響 [246](#)  
   QUERY 機能 [247](#)  
   TCTUA への影響 [246](#)  
   TLT への影響 [247](#)  
 自動インストール (自動インストール) [6](#)  
 自動インストール制御プログラム (autoinstall control  
   program)  
   接続の自動インストール [263](#)  
   プログラムの自動インストール [266](#)  
 自動インストール・モデル端末定義  
   インストール [268](#)  
 自動インストール・モデル・テーブル (AMT) [248](#)  
 自動サインオフ [254](#)  
 自動始動 [131, 155](#)  
 自動始動オーバーライド・レコード [67](#)  
 自動ジャーナリング [347](#)  
 自動ジャーナル処理 [346](#)  
 始動ジョブ・ストリーム  
   端末 [315](#)  
   自動トランザクション開始  
   削除を阻止する TCT 項目のロック [251](#)  
   自動インストール端末 [246](#)  
 自動取引装置 (3614 および 3624)  
   自動インストールに不適格 [251](#)  
 始動プロシージャ、DFHSTART [143](#)  
 自動預金支払機 (ATM) (3614 および 3624)  
   自動インストールに不適格 [251](#)  
 自動ログオフ [254](#)  
 ジャーナリング  
   BWO [46](#)  
 ジャーナル  
   区画外一時データ・セットのリカバリー用 [469](#)  
   自動インストール [268](#)  
   読み取り用のオフライン・プログラム [456](#)  
 出力データ・セット [470](#)  
 順次端末装置  
   静止のための論理クローズ [151](#)  
   入力データの終了 [318](#)  
   入力のコーディング [318](#)  
   ファイルの終わり [318](#)  
   DFHCT2500、端末終了警告メッセージ [318](#)  
   DFHCT2507、端末終了警告メッセージ [318](#)  
   (静止中の) 装置のクローズ [318](#)  
 順方向リカバリー  
   一時記憶 [471](#)  
   区画内一時データ [469](#)  
   VSAM ファイルの定義 [462](#)  
 使用ストレージ  
   説明 [337](#)  
 初期始動  
   テーブル・エントリーを除去するための使用 [324](#)  
   テーブルを再作成するための GRPLIST の使用 [324](#)  
 初期設定 (PLT) プログラム  
   使用 [470](#)  
 初期設定パラメーターに関するオペレーター通信 [159](#)  
 初期設定パラメーターのオペレーター通信 [129](#)  
 ジョブ  
   DFHISTAR [44](#)  
   ジョブ・ストリーム  
   CICS startup [143](#)  
   ジョブ制御言語 (JCL)  
   開始タスクとしての CICS の [152, 156](#)  
   バッチ・ジョブとしての CICS [143, 156](#)  
 スタート、ウォーム  
   自動インストールされた TCT 項目の非リカバリー [253](#)  
 スタート、緊急時  
   自動インストールされた TCT 項目の一時的なリカバリー  
   [253](#)  
 スタート、コールド

スタート、コールド (続き)  
自動インストールされた TCT 項目の非リカバリ [253](#)  
ストレージ管理サブシステム (SMS)  
オープン中のバックアップ機能 [46](#)  
リリース情報 [47](#)  
ストレージ所要量 (CICS ONC RPC) [413](#)  
ストレージの計算 [169](#)  
ストレージ要件  
CICS 管理クライアント・インターフェース 用 [381](#)  
スペース計算  
グローバル・カタログ [68](#)  
ダンプ・データ・セット [78](#)  
ディスク・スペース [22](#)  
データ・セットの定義 [41](#)  
CSD [22](#)  
制御間隔 (CI) [50](#)  
静的ルーティング  
EXCI [399](#)  
セキュリティ  
および自動インストール [261](#)  
事前設定 [261](#)  
実動システム用の読み取り専用 CSD ファイル [13](#)  
バンドル・リソース [310](#)  
非 MRO CICS システム用の異なる CSD ファイル [13](#)  
リソース・セキュリティ検査 (RESSEC) [13](#)  
CICS 管理クライアント・インターフェース 用 [386](#)  
LOCK および UNLOCK コマンド  
グループへのアクセスの制御 [13](#)  
リストへのアクセスの制御 [13](#)  
セキュリティ検査  
機能シップとの比較 [343](#)  
データ・テーブルの [343](#)  
AOR 接続時 [344](#)  
FOR ログオン時 [344](#)  
RACF の考慮事項 [343](#)  
SAF の使用 [343](#)  
セキュリティ出口 [484](#)  
接続の自動インストール  
およびリカバリと再始動 [263](#)  
自動インストール制御プログラム [263](#)  
接続マネージャー (CICS ONC RPC)  
定義 [415](#)  
パネル・フォーマット [420](#)  
接続マネージャーの開始 [419](#)  
接続マネージャー・パネル  
DFHRP03 [425](#)  
DFHRP22 [435](#)  
相互通信、リソース定義 [12](#)  
相対バイト・アドレス (RBA) [53](#)  
ソース・データ・セット  
ファイルが複数の [346](#)

## [タ行]

代替索引 [346](#)  
代替索引 (alternate index) [465](#)  
単一セッション端末  
APPC  
自動インストールに適合 [251](#)  
単一ノードの持続セッション [322](#)  
ダンプ  
ダンプ・テーブルによる制御 [78](#)  
START= パラメーターの効果 [135](#)  
ダンプ・データ・セット

ダンプ・データ・セット (続き)  
スペース計算 [78](#)  
ダンプ・テーブル機能 [78](#)  
割り振るためのジョブ制御ステートメント [81](#)  
ダンプ・フォーマット用の JCL  
CICS ONC RPC [414](#)  
端末エラー・プログラム、DFHTEP [318](#)  
端末管理テーブル (TCT)  
ダミー・管理テーブル、DFHTCTDY [124](#)  
TCT 項目、自動インストール済み [246-248](#)  
端末所有領域 [3](#)  
端末専有領域 (TOR) [352](#)  
端末定義 [315](#)  
端末定義の検査 [334](#)  
端末リスト・テーブル (TLT) [278](#)  
端末リスト・テーブル (TLT) (terminal list table (TLT))  
自動インストールの効果 [247](#)  
チャネル・ベースのサービス [285](#)  
重複するリソース定義名 [325](#)  
定義  
ENQMODEL [330](#)  
LIBRARY [331](#)  
データ機能階層記憶プログラム (DFHSM)  
オープン時バックアップ [46](#)  
データ所有領域 [3](#)  
データ・スペース  
データ・テーブルによる使用 [337](#)  
データ・セット  
暗号化 [103](#)  
一時データ (区画外) [51](#)  
一時データ (区画内) [51](#)  
カタログ式データ・セット [66, 72](#)  
ダンプ [78](#)  
定義、一時データ (区画外) [54](#)  
定義、一時データ (区画内) [52](#)  
デバッグ・プロファイル  
作成 [98](#)  
定義 [98](#)  
補助一時記憶域 [48](#)  
補助トレース [76](#)  
メッセージ・データ・セット [96](#)  
ユーザー・データ・セット  
オープン [91, 92](#)  
クローズ [92](#)  
BDAM [88](#)  
CICS に対する定義 [89](#)  
VSAM [83](#)  
VSAM データ・セットのロード [84](#)  
ユーザー・ファイルの定義 [89](#)  
BDAM [88](#)  
CDBM SUPPORT データ・セット [94](#)  
CEMT を使用する動的割り振り [90](#)  
CICS によって使用される MVS システム・データ・セット  
[45](#)  
CICS によって使用される MVS システム・データ・セッ  
ト [45](#)  
CSD [21](#)  
DFHCMACI ジョブによる作成 [45](#)  
DFHCOMDS ジョブによる作成 [44](#)  
DFHDEFDS ジョブによる作成 [44](#)  
GTF データ・セット [46](#)  
SDUMP データ・セット [45](#)  
SMF データ・セット [45](#)  
VSAM のベースとパス [84](#)

- データ・セット、区画外
  - 出力 [470](#)
  - 入力 [469](#)
- データ定義
  - 限られた数のインストール [325](#)
- データ・テーブル
  - オープン [93](#)
  - 概要 [93](#)
  - クローズ [94](#)
  - タイプ [93](#)
  - ロード [93](#)
- データ・テーブルの計画 [337](#), [344](#)
- データ・テーブルのサイズ
  - INQUIRE コマンドによる検索 [348](#), [349](#)
  - SET コマンドによる定義 [348](#)
- データ・テーブルのタイプ
  - INQUIRE コマンドによる検索 [348](#), [349](#)
  - SET コマンドによる定義 [348](#)
- データ・フォーマット [429](#)
- データ保全性
  - CMT の [346](#)
  - UMT の [347](#)
- 出口
  - XDUCLSE、ダンプ・グローバル・ユーザー出口 [80](#), [82](#)
  - XDUOUT、ダンプ・グローバル・ユーザー出口 [80](#), [82](#)
  - XDUREQ、ダンプ・グローバル・ユーザー出口 [80](#), [82](#)
  - XDUREQC、ダンプ・グローバル・ユーザー出口 [80](#), [82](#)
- デッドロック、トランザクション
  - DTIMOUT の影響 [459](#)
- デバッグ
  - 準備 [153](#)
- デバッグ・プロファイル・データ・セットの定義
  - リモート・ファイルとして [101](#)
  - VSAM RLS ファイルとして [100](#)
  - VSAM 非 RLS ファイルとして [100](#)
- デバッグ・プロファイルのデータ・セット
  - 作成 [98](#)
  - 定義
    - リモート・ファイルとして [101](#)
    - VSAM RLS ファイルとして [100](#)
    - VSAM 非 RLS ファイルとして [100](#)
- 統計
  - データ・テーブル選択のため [340](#)
- 動的トランザクション・バックアウト [347](#)
- 動的ルーティング
  - EXCI [399](#)
- 動的割り振り
  - ADYN、動的割り振りトランザクション [91](#)
  - DFH99、サンプル DYNALLOC ユーティリティ・プログラム [91](#)
- 登録
  - CICS ONC RPC に [430](#)
  - TCP/IP for MVS に [430](#)
- トポロジー [1](#)
- ドメイン
  - カーネル [72](#)
  - パラメーター [72](#)
- トラップ、DFHXCTRA
  - TRAP、DFHXCOPT のパラメーター [410](#)
- トランザクション
  - ADYN、動的割り振りトランザクション [91](#)
  - CESF GOODNIGHT [318](#)
  - CESF LOGOFF [319](#)
  - CESN [321](#)

- トランザクション (続き)
  - CSFU、CICS ファイル・ユーティリティ・トランザクション [92](#)
- トランザクション ID [487](#)
- トランザクションの異常終了の処理
  - プログラム・エラー・プログラム (PEP) ユーザーによるコーディング [475](#)
- トランザクションのリカバリー
  - 長時間待ちのタイムアウト [459](#)
  - ページ [459](#)
  - DFHREST を使用した自動リスタート [459](#)
- トランザクション・リスト・テーブル (XLT) [272](#)
- トランザクション・ルーティング
  - 自動インストールの制限 [247](#)
- トランスポート所有領域 [3](#)
- トレース
  - ディスクで補助データ・セットを定義するためのサンプル・ジョブ [77](#)
  - 補助トレース・データ・セットの定義 [76](#)
  - 補助トレース・データ・セットを割り振るためのジョブ制御ステートメント [77](#)
  - DFHAUXT 補助トレース・データ・セット [76](#)
  - DFHBUXT 補助トレース・データ・セット [76](#)
  - DFHXCPT の TRACE パラメーター [410](#)
  - DFHXCPT の TRACESZE パラメーター [410](#)
- トレース (CICS ONC RPC)
  - トレース・オプションの設定 [424](#), [435](#)
  - トレース・レベルの設定 [424](#), [435](#)
- トレース内のユーザー・データの抑止
  - CONFDATA オプション [408](#)

## [ナ行]

- 夏時間
  - クロックの変更 [457](#)
  - CICS への影響 [457](#)
- 名前付きカウンター
  - オプション・テーブル
    - 定義 [218](#)
    - パラメーター [219](#)
    - CICS で使用可能にする [220](#)
  - オプション・テーブルの定義 [218](#)
  - 警告パラメーター [226](#)
  - サーバーの開始 [215](#)
  - サーバーの概要 [216](#)
  - サーバーの定義および始動 [222](#)
  - サーバー・パラメーター [223](#)
  - サーバー領域の制御 [226](#)
  - 自動リスタート・マネージャ・パラメーター [224](#)
  - デバッグ・トレース・パラメーター [225](#)
  - パラメーター、サーバー [223](#)
  - プールのアンロードと再ロード [230](#)
  - プールのリスト構造のダンプ [231](#)
  - プールを削除または空にする [229](#)
  - リスト構造、定義 [221](#)
  - リスト構造パラメーター [225](#)
  - SET コマンド [227](#)
  - XES [227](#)
- 二重の目的を持つリソース定義 [12](#)
- 日本語機能
  - CSD への定義のインストール [40](#)
- 入力データ・セット [469](#)

## [ハ行]

パイプ  
  ALLOCATE\_PIPE 中の DFHXCURM の呼び出し [404](#)  
パイプライン端末  
  自動インストールに不適格 [251](#)  
バインド・セキュリティ [344](#)  
バックアウト (backout)  
  一時記憶域の [471](#)  
  一時データの [466](#)  
発見 [137](#)  
バッチ・ジョブ、状態の照会 [404](#)  
パフォーマンス  
  データ・テーブルの利点 [337](#)  
  CMT の [337](#)  
  UMT の [337](#)  
バンドル  
  アプリケーションのタイプ [285](#)  
  スコーピング (scoping) [307](#)  
  定義 [283](#)  
  プラットフォーム [357](#)  
  プラットフォーム内 [307](#)  
バンドルされた LIBRARY リソース [290](#)  
バンドルされた PROGRAM リソース [290](#)  
バンドルされた TRANSACTION リソース [290](#)  
バンドルされた zFS 成果物 [297](#)  
バンドルされたリソースの特性 [290](#)  
バンドル・セキュリティ [310](#)  
バンドルのスコーピング [307](#)  
バンドル・リカバリー [309](#)  
バンドル・リソースの相互依存関係 [290](#)  
汎用接続  
  定義 [401](#)  
汎用トレース機能 (GTF) [45](#)  
汎用ログ・ストリームの定義 [441](#)  
ビジネス・アプリケーション・サービス、CPSM [6](#)  
非ブロッキング呼び出しタイプ  
  指定 [428](#)  
ファイル  
  外部設計上の考慮事項  
    アプリケーション・データの使用 [460](#)  
  データ・テーブルとして使用 [339](#)  
  複数のパスの更新 [465](#)  
  リカバリー属性の定義 [460](#), [462](#)  
ファイル所有領域 [3](#)  
ファイル・セキュリティ [344](#)  
ファイル専有領域 (FOR) [352](#)  
ファイル定義  
  リカバリー属性の整合性チェック  
  オープン失敗のオーバーライド [465](#)  
ファイルの選択  
  データ・テーブルとして使用する [339](#)  
ファイルのバックアウト  
  BDAM [462](#)  
複数の CSD ファイル [13](#)  
複数のファイル  
  ソース・データ・セットが同じ [346](#)  
物理リカバリー [467](#)  
プラットフォーム  
  作成 [357](#)  
  デプロイ [358](#)  
プラットフォーム設計 [352](#)  
プラットフォームのデプロイ [358](#)  
プラットフォーム・プロジェクト [357](#), [358](#)

ブリッジ (3270)  
  システム初期設定パラメーター [391](#)  
  ブリッジ機能定義 [391](#)  
  ブリッジ機能プロパティ [394](#)  
  FACILITYLIKE [391](#)  
プリンター  
  自動インストールに適格 [251](#)  
プログラム  
  構成 [477](#)  
  自動インストール [264](#)  
プログラム・エラー・プログラム (PEP)  
  編集 [475](#)  
  ユーザー提供 DFHPEP [475](#)  
  CICS 提供の DFHPEP [474](#)  
  DFHPEP の省略 [476](#)  
プログラム自動インストール  
  およびリカバリーと再始動 [267](#)  
  カタログ [266](#)  
  自動インストール制御プログラム [266](#)  
  モデル定義 [266](#)  
プログラム自動インストール機能 [267](#)  
プログラムの自動インストール出口  
  DFHPGADX アセンブラー・プログラム [267](#)  
  DFHPGAHX-C プログラム [267](#)  
プログラムの自動インストール出口のための DFHPGADX ア  
  センブラー・プログラム [267](#)  
プログラムの自動インストール出口のための DFHPGAHX-C  
  プログラム [267](#)  
プログラムの自動インストール出口のための DFHPGALX-  
  PL/I プログラム [267](#)  
プログラムの自動インストール出口のための DFHPGAOX-  
  COBOL プログラム [267](#)  
プログラム・リスト・テーブル (PLT) [141](#), [270](#)  
プログラム・ロード・テーブル (PLT) [484](#)  
別名 (CICS ONC RPC)  
  指定 [428](#)  
  定義 [415](#)  
  トランザクション定義 [415](#)  
  EDF 端末 ID の指定 [428](#)  
ヘルプ・ファイルの作成 [482](#)  
変数 [312](#), [314](#)  
補助一時記憶域データ・セット  
  スペースに関する考慮事項 [49](#)  
  制御間隔サイズ [50](#)  
  制御間隔数 [51](#)  
  定義するジョブ制御ステートメント [49](#)  
  CICS 実行のジョブ制御ステートメント [49](#)  
補助トレース・データ・セット  
  割り振るためのジョブ制御ステートメント [77](#)  
  CICS 実行のジョブ制御ステートメント [76](#)  
保全性  
  CMT データの [346](#)  
  UMT データの [347](#)

## [マ行]

マクロ  
  ASLTAB (z/OS Communications Server のマクロ) [316](#)  
  DEVTYPE (MVS マクロ) [79](#)  
  MDLTAB (z/OS Communications Server のマクロ) [316](#)  
  MGCR (MVS コマンドを発行するため) [322](#)  
  MVS SDUMP [45](#)  
マクロ定義  
  フォーマット [273](#)



マクロのフォーマット [273](#)  
マップ・セット  
    自動インストール [264](#)  
マップ・セット定義 [417](#)  
マルチノードの持続セッション [322](#)  
無効化処理  
    タイプ [433](#)  
メッセージ  
    CICS の始動に関するコンソール・メッセージ [158](#)  
メッセージ・データ・セット  
    定義してロードするためのジョブ制御ステートメント  
    [96](#)  
モジュール・ロード・ライブラリー連結、DFHRPL [148](#)  
モデル端末サポート (MTS) [261](#)  
モニター [134](#)  
モニター管理テーブル (MCT) [270](#), [278](#)

## [ヤ行]

ユーザー異常終了出口の作成 [474](#)  
ユーザー置換可能モジュール  
    DFHXCURM [404](#)  
ユーザー・ファイル  
    カップリング・ファシリティ・データ・テーブル・サ  
    ーバー [181](#)  
ユーザー・ファイル定義 [82](#)  
ユーザー保守データ・テーブル  
    ジャーナリング [347](#)  
    データ保全性 [347](#)  
    パフォーマンス [337](#)  
    リソース定義 [346](#)  
ユーティリティ・プログラム  
    DFHCCUTL、ローカル・カタログ初期化ユーティリティ  
    ・プログラム [74](#)  
    DFHJUP、CICS ジャーナル・ユーティリティ・プログ  
    ラム [66](#)  
    IDCAMS、AMS ユーティリティ・プログラム [84](#)  
ユーティリティ・プログラム、オフライン  
    CICS システム定義ファイルの初期化、DFHCSDUP [241](#)  
読み取り、16 MB 境界より上 [269](#)  
読み取り専用 CSD ファイル、実動システム用 [13](#)

## [ラ行]

ライブラリー  
    SCEERUN, Language Environment runtime library [144](#)  
    SCEERUN2, Language Environment ランタイム・ライブ  
    ラリー [144](#)  
ライン・プリンター [318](#)  
リカバリー、OSGi バンドル [309](#)  
リカバリー可能サービス・テーブル (RST) [270](#)  
リカバリー属性の定義  
    ファイル [462](#)  
リカバリーと再始動  
    および接続の自動インストール [263](#)  
    およびプログラムの自動インストール [267](#)  
    自動インストール (autoinstall) [253](#)  
リスト構造  
    ストレージに関する考慮事項 [221](#)  
リスト構造、定義 [206](#), [221](#)  
リソース  
    定義方法 [8-10](#)  
リソース・セキュリティー [416](#)

リソース・セキュリティー検査 (RESSEC) [13](#)  
リソース・チェッカー (CICS ONC RPC)  
    オプションの指定 [424](#), [435](#)  
リソース定義  
    インストール [324](#)  
    概要 [5](#)  
    コミット [324](#)  
    システム初期設定時  
        ウォーム・スタート (warm start) [324](#)  
        緊急時再始動 [324](#)  
        コールド・スタート (cold start) [324](#)  
        初期始動 [324](#)  
    システム・テーブルからの除去  
        コールド・スタート時 [324](#)  
        自動インストールのログオフ時 [247](#)  
        初期始動時 [324](#)  
    説明 [346](#)  
    相互通信用 [12](#)  
    属性 [5](#)  
    二重の目的 [12](#)  
    方式 [7](#), [8](#)  
    CICS システムでのインストール  
        INSTALL コマンドの使用 [325](#)  
    CMT の概要 [346](#)  
    CONNECTION 定義 [400](#)  
    SESSIONS 定義 [401](#)  
    UMT の概要 [347](#)  
リソース定義のインストール  
    一時データ・キュー [332](#)  
    ウォーム・スタート (warm start) [324](#)  
    および既存のエントリーの削除 [325](#)  
    緊急時再始動 [324](#)  
    コールド・スタート (cold start) [324](#)  
    最初にリソースを静止する [325](#)  
    初期始動 [324](#)  
リソース定義のグループ  
    インストール  
        CICS の実行中 [325](#)  
        ロックされた、有効なコマンド [13](#)  
リソース定義の方式 [7](#), [8](#)  
利点  
    データ・テーブルの [337](#)  
領域 [1](#)  
領域状況サーバー  
    リスト構造、定義 [206](#)  
領域タイプ [352](#)  
領域の状況  
    サーバーの開始 [205](#)  
リンク・エディット  
    DFHAUPLE の使用 [406](#)  
    DFHXCOPト オプション・テーブル [406](#)  
レコード・カウント警告 [377](#)  
レコード・レベル共用 (RLS)  
    VSAM データ共有 [85](#)  
レプリカ生成ログ・ストリームの定義 [442](#)  
ローカル・カタログ・データ・セット (LCD)  
    再始動での使用 [130](#)  
説明 [72](#)  
    定義および初期設定のためのジョブ制御ステートメント  
    [72](#)  
    CICS 実行のジョブ制御ステートメント [73](#)  
ロード・モジュール  
    データ・テーブルで必要 [345](#)  
ロギング



ロギング (続き)  
一般ログのための CICS ジャーナルの定義  
順方向リカバリー・ログ [59](#)  
ユーザー・ジャーナル [59](#)  
ダミー・ログの定義 [56](#)  
ログの自動インストール [60](#)  
CICS システム・ログの定義 [55](#)  
CICS ログの定義 [55](#)  
JOURNALMODEL 定義 [60](#)  
ロギング・マネージャー [55](#)  
ログオフ (自動インストール)、TCT 項目の削除 [247](#)  
ログオン・プロセス、自動インストール使用 [247](#)  
ログ・ストリーム  
システム・ログとジャーナル名のマッピング [61](#)  
ログのログ  
障害 [456](#)  
ロック、内部  
CEMT INQUIRE TERMINAL による実行 [247](#)  
ロックされたグループ、有効なコマンド [13](#)  
論理リカバリー [466](#)

## [数字]

16 MB 境界より上、読み取り [269](#)  
3270 端末 (非 SNA)  
自動インストールに適格 [251](#)  
3270 ブリッジ  
ブリッジ機能プロパティ [394](#)  
3614 および 3624 のデバイス  
自動インストールに不適格 [251](#)  
4 タブルの登録 [430](#)  
4 タブル・レコード [423](#)

## A

ADYN、動的割り振りトランザクション [91](#)  
AIBRIDGE [391](#)  
ALLOCATE\_PIPE コマンド  
処理中の DFHXCURM の呼び出し [404](#)  
AMT (自動インストール・モデル・テーブル) [248](#)  
ANYNET ソフトウェア [413](#)  
AOR [352](#)  
AOR (アプリケーション所有領域) [3](#)  
API 改善点 [137](#)  
APPLID オペランド  
DFHSIT  
グループおよびリストへのアクセスを制御するため  
[13](#)  
ARMREGISTERED、名前付きカウンター・サーバー [227](#)  
ASLTAB (z/OS Communications Server のマクロ) [316](#)  
Atom フィールド [285](#)  
AUTHUSER コマンド [483](#), [484](#), [487](#)  
AUTOCONNECT 属性  
TYPETERM [246](#), [253](#), [256](#)  
Automatic Enable オプション [424](#), [435](#)

## B

BASESCOPE [307](#)  
BDAM データ・セット  
オープンとクローズ [91](#), [92](#)  
作成とロード [88](#)  
BDAM ファイル

BDAM ファイル (続き)  
バックアウト [462](#)  
BMS (基本マッピング・サポート)  
BMS のバージョンの選択 [124](#)  
BMS ROUTE 要求  
自動インストール端末 [246](#)  
BRMAXKEEPTIME [391](#)  
BSAM 装置  
DD ステートメント [145](#), [318](#)  
BTS データ・セット、DFHLRQ [151](#)  
BWO (オープン時バックアップ)  
アクティビティ・キーポイント処理の無効化 [47](#)  
概要 [46](#)  
ストレージ管理機能 [47](#)  
制約事項 [48](#)

## C

CADS [38](#)  
CCSO 一時データ宛先 [149](#)  
CEDA コマンド [10-12](#)  
CEDA トランザクション  
コンソール装置の定義 [320](#)  
複数の CICS 領域での CSD の共用 [21](#)  
リカバリーとバックアップ [33](#)  
SNA LU 端末定義のインストール [315](#)  
CEMT  
INQUIRE コマンド [348](#), [349](#)  
SET コマンド [348](#)  
CEMT INQUIRE EXCI コマンド [404](#)  
CEMT INQUIRE TERMINAL  
TCT 項目のロック [247](#)  
CEMT INQUIRE コマンドの EXCI [404](#)  
CEMT マスター端末トランザクション [80](#)  
CESF GOODNIGHT トランザクション [318](#)  
CESF LOGOFF トランザクション [319](#)  
CESN トランザクション [321](#)  
CESO 一時データ宛先 [150](#)  
CI (制御間隔) [50](#)  
CICREX プログラム [484](#)  
CICS ONC RPC データ・セット [423](#)  
CICS ONC RPC オプション [423](#)  
CICS ONC RPC 定義レコード [423](#)  
CICS ONC RPC によってサポートされるクライアント [413](#)  
CICS ONC RPC の前提条件 [413](#)  
CICS ONC RPC の無効化 [433](#)  
CICS ONC RPC の有効化 [425](#)  
CICS ONC RPC 用のプログラム  
CICS での定義 [416](#)  
CICS TCP/IP ソケット・インターフェース [413](#)  
CICS 管理クライアント・インターフェース  
ストレージ要件 [381](#)  
セキュリティのセットアップ [386](#)  
セットアップ [361](#), [384](#), [386](#)  
CICS 領域のセットアップ [384](#)  
DFH\$WUTC サンプル [384](#)  
DFH\$WUUR サンプル URI マップ [384](#)  
CICS 管理テーブルの接尾部 [279](#)  
CICS グローバル・カタログ  
自動インストール・モデル [247](#)  
CICS システム・グループ [1](#)  
CICS システム定義ファイル (CSD)  
オフライン・ユーティリティ・プログラム、DFHCSDUP  
[241](#)

CICS ジャーナル・ユーティリティー・プログラム (DFHJUP)

[66](#)

CICS 初期化

ウォーム・スタート (warm start) [324](#)

緊急時再始動 [324](#)

コールド・スタート (cold start) [324](#)

初期始動 [324](#)

CICS 初期設定 JCL の更新 [479](#)

CICS 提供の TYPETERM 定義

AUTOCONNECT 属性 [246](#), [253](#)

CICS トポロジー [1](#)

CICS 内のリソース定義 [415](#)

CICS バージョン間のマイグレーション [415](#)

CICS 保守データ・テーブル

ジャーナリング [346](#)

データ保全性 [346](#)

パフォーマンス [337](#)

リソース定義 [346](#)

CICS 領域 [1](#)

CICS 領域の構成 [41](#)

CICS 領域の始動

開始タスクとしての [152](#), [156](#)

サンプル・ジョブ・ストリーム [143](#)

始動のタイプの指定 [130](#)

バッチ・ジョブとして [156](#)

MVS START コマンド [152](#), [156](#)

START=AUTO [131](#), [155](#), [156](#)

START=COLD [132](#)

START=INITIAL [132](#), [155](#)

START=STANDBY、XRF 代替 CICS の [156](#)

CICS 領域のストレージ所要量 [142](#)

CICS 領域のセットアップ [41](#)

CICSECX1 セキュリティー 出口 [485](#)

CICSECX2 セキュリティー 出口 [485](#)

CICSplex [1](#), [352](#)

CICSplex SM [137](#)

CICSPROF exec [487](#)

CICSSVC、DFHXCOPT のパラメーター [407](#)

CICSTART exec [483](#), [484](#)

CICSTART.PROC の更新 [478](#)

CLSDST、ログオン時に CICS により発行 [247](#)

CLT (コマンド・リスト・テーブル)

CLT (コマンド・リスト・テーブル) [269](#)

CMAC 機能用のメッセージ・データ・セット [96](#)

CMAC サポート、メッセージ・データ・セット [96](#)

CMCI

ストレージ要件 [381](#)

セットアップ [361](#)

レコード・カウンタ警告の設定 [377](#)

DEFAULTWARNCNT [377](#)

CMDSEC [416](#)

CONFDATA、DFHXCOPT のパラメーター [408](#)

CONNECT

セキュリティ検査 [344](#)

connection manager panels

DFHRP01 [419](#)

DFHRP02 [423](#)

DFHRP04 [419](#)

DFHRP06 [433](#)

DFHRP10 [422](#)

DFHRP11 [431](#)

DFHRP12 [432](#)

DFHRP13 [432](#)

DFHRP14 [436](#)

connection manager panels (続き)

DFHRP15 [436](#)

DFHRP16 [422](#)

DFHRP17 [438](#)

DFHRP18 [438](#)

DFHRP20 [434](#)

DFHRP21 [437](#)

DFHRP2B [438](#)

DFHRP5 [426](#)

DFHRP5B [427](#)

CONNECTION 定義

CONNTYPE 属性 [400](#)

PROTOCOL 属性 [401](#)

CONNTYPE 属性、CONNECTION 定義 [400](#)

CPSM ビジネス・アプリケーション・サービス [6](#)

CRPC トランザクション [419](#)

CRPO 一時データ・キュー [417](#), [421](#)

CSD

リリース間での共用 [30](#)

CSD (CICS システム 定義) ファイル

再始動時に使用されない [324](#)

実動システムに読み取り専用ファイルを使用 [13](#)

定義 [464](#)

非 MRO CICS システムに複数のファイルを使用 [13](#)

CSD (CICS システム 定義ファイル)

オフライン・ユーティリティー・プログラム、DFHCSDUP

[241](#)

緊急時再始動およびバックアウト [36](#)

定義 [21](#)

定義および初期設定のためのジョブ [23](#)

データ・セットのスペース [22](#)

動的割り振り [39](#)

日本語機能の定義 [40](#)

リカバリーとバックアップ [33](#)

CICS 実行のジョブ制御ステートメント [39](#)

CICS 始動ジョブの DD ステートメント [39](#)

CICS で CSD を有効にする方法 [39](#)

CSD の共用

内部ロックによる保護 [28](#)

内部ロックの解放 [31](#)

CICS 領域間での共用 [28](#)

CSD の動的割り振り [39](#)

CSD ファイル

インストール

CICS の初期化 [324](#)

複数の CSD ファイル [13](#)

CSFU、CICS ファイル・ユーティリティー・トランザクショ

ン [92](#), [93](#)

CSSL 一時データ宛先 [149](#)

CSYSGRP [352](#)

CXRF 一時データ・キュー [54](#)

CXRF キュー [54](#)

## D

Db2 エントリー・リソースの定義の検査 [329](#)

Db2 接続リソースの定義の検査 [328](#)

Db2 トランザクション・リソースの定義の検査 [329](#)

DB2 ロード・ライブラリー

DSNTIAR と DSNTIA1 の要件 [149](#)

DB2CONN

インストールおよび破棄 [327](#)

DEFCMD コマンド [487](#)

DEVTYPE マクロ (MVS) [79](#)

DFH\$TCTS、コピーブック [318](#)  
DFH\$TDWT (一時データを端末に書き込むサンプル・プログラム) [51](#)  
DFH\$WUTC サンプル TCP/IP サービス [384](#)  
DFH\$WUUR サンプル URI マップ [384](#)  
DFH99、サンプル DYNALLOC ユーティリティ・プログラム [91](#)  
DFHAUPLD プロシージャ [406](#)  
DFHAUXT 補助トレース・データ・セット [76](#)  
DFHBUXT 補助トレース・データ・セット [76](#)  
DFHCCUTL、ローカル・カタログ初期化ユーティリティ・プログラム [74](#)  
DFHCNV マクロ [269](#)  
DFHCOMP1、CSD リソース定義グループ [30](#)  
DFHCOMP2、CSD リソース定義グループ [30](#)  
DFHCSDUP  
    日本語機能の定義 [40](#)  
DFHCSDUP オフライン・ユーティリティ [6](#)  
DFHCSDUP コマンド [10-12](#)  
DFHCSDUP システム定義ユーティリティ・プログラム  
    コマンド処理に関する考慮事項 [244](#)  
    システム定義ファイルの処理 [241](#)  
    バッチ・プログラムとして起動 [242](#)  
DFHCSVC、CICS タイプ 3 SVC [162](#)  
DFHCXRF データ・セット、一時データ区画外  
    アクティブな CICS 領域内 [54](#)  
    DD ステートメント [55](#)  
DFHDBFK データ・セット  
    定義してロードするためのジョブ制御ステートメント [95](#)  
DFHDCTG、一時データ定義のグループ [37, 38](#)  
DFHDPFMB  
    定義  
        リモート・ファイルとして [101](#)  
        VSAM RLS ファイルとして [100](#)  
        VSAM 非 RLS ファイルとして [100](#)  
        デバッグ・プロファイルのデータ・セット  
        作成 [98](#)  
DFHDPFMP  
    定義  
        リモート・ファイルとして [101](#)  
        VSAM RLS ファイルとして [100](#)  
        VSAM 非 RLS ファイルとして [100](#)  
        デバッグ・プロファイルのデータ・セット  
        作成 [98](#)  
DFHDPFMX  
    デバッグ・プロファイルのデータ・セット  
    作成 [98](#)  
DFHDTCV [344](#)  
DFHDTVCV [344](#)  
DFHFCT マクロ  
    SERVREQ オペランド [13](#)  
DFHGCD、グローバル・カタログ・データ・セット [67](#)  
DFHISTAR ジョブ [44, 280, 283](#)  
DFHJUP、CICS ジャーナル・ユーティリティ・プログラム [66](#)  
DFHLCD、ローカル・カタログ・データ・セット [73](#)  
DFHLIST 始動リスト  
    DFHMISC [474](#)  
DFHLRQ、BTS データ・セット [151](#)  
DFHMRMS [344](#)  
DFHNCMN、名前付きカウンター・サーバー領域プログラム [222](#)  
DFHNCO マクロ [218](#)

DFHNCOPT、名前付きカウンター・オプション・テーブル [218](#)  
DFHPEP  
    DFHMISC での定義 [474](#)  
DFHPGALX-PL/I プログラム [267](#)  
DFHPGAOX-COBOL 定義 [267](#)  
DFHREST、ユーザー置き換え可能モジュール [459](#)  
DFHRPL、モジュール・ロード・ライブラリー [148](#)  
DFHRSTAT [206](#)  
DFHSIT キーワードおよびオペランド  
    未定義のキーワード・エラー・メッセージ [123](#)  
DFHSIT マクロ  
    APPLID オペランド [13](#)  
    GRPLIST オペランド [324](#)  
    START オペランド [324](#)  
DFHSM (データ機能階層記憶プログラム)  
    オープン時バックアップ [46](#)  
DFHSNT マクロ  
    OPIDENT オペランド [13](#)  
DFHSTART、サンプル始動プロシージャ [143](#)  
DFHTC2500、端末終了警告メッセージ [318](#)  
DFHTC2507、端末終了警告メッセージ [318](#)  
DFHTCT5\$, サンプル TCT [318](#)  
DFHTCTDY、ダミー TCT [124](#)  
DFHTEP、端末エラー・プログラム [318](#)  
DFHXCUM ユーザー置換可能モジュール [404](#)  
DFHXQMN システム初期設定パラメーター  
    プール・リスト構造の作成のため [170](#)  
DFHXQMN、TS サーバー・プログラム [170](#)  
DSA (動的ストレージ域)  
    RENTPGM、読み取り専用 DSA のストレージ [146](#)  
DSNTIA1 [149](#)  
DSNTIAC [149](#)  
DSNTIAR [149](#)  
DTIMOUT オプション (DEFINE TRANSACTION) [459](#)  
DURETRY、DFHXCUM のパラメーター [408](#)

## E

ENQMODEL 定義  
    定義のインストール [330](#)  
EXCI  
    静的ルーティング [399](#)  
    動的ルーティング [399](#)  
EXEC CICS CREATE コマンド [6](#)  
EXEC CICS CSD コマンド [6](#)  
EXEC CICS START コマンド  
    自動インストール端末 [246](#)

## F

FACILITYLIKE [391](#)  
FILEPOOL コマンド [484](#)  
FILSTAT オペランド [93](#)  
FINAL、TYPE= マクロ [276](#)  
FOR [352](#)  
FOR (ファイル所有領域) [3](#)

## G

GCD (グローバル・カタログ・セット)  
    スペース計算 [68](#)  
    説明 [66](#)

GCD (グローバル・カタログ・セット) (続き)  
定義および初期設定のためのジョブ制御ステートメント [66](#)  
バッファ・スペース [71](#)  
CICS 実行のジョブ制御ステートメント [67](#)  
GCD の指定用 AMP パラメーター [150](#)  
GRPLIST  
再始動時に使用されない [324](#)  
DFHSIT オペランド [324](#)  
GTF (汎用トレース機能) [45](#)  
GTF、DFHXCOPT のパラメーター [409](#)

## I

IDCAMS、AMS ユーティリティ・プログラム [84](#)  
IEV017 エラー・メッセージ [123](#)  
INITIAL、TYPE= マクロ [275](#)  
INQUIRE FILE コマンド  
説明 [348, 349](#)  
MAXNUMRECS パラメーター [348, 349](#)  
TABLE パラメーター [348, 349](#)  
INSTALL コマンド  
リソース定義のグループをインストールする [325](#)  
INSTLN パラメーター [343](#)  
IPCONN 定義  
インストール [330](#)  
IPCS VERBEXIT [414](#)

## J

Java アプリケーション [285](#)  
JCL (ジョブ制御言語)  
CICS startup  
開始タスクとしての [152](#)  
バッチ・ジョブとして [143](#)  
JOURNALMODEL 定義 [60](#)

## K

KSDS (キー順データ・セット)  
UMT を使用 [347](#)

## L

Language Environment [413](#)  
Language Environment ランタイム・ライブラリー、SCEERUN [144](#)  
Language Environment ランタイム・ライブラリー、SCEERUN2 [144](#)  
LCD (ローカル・カタログ・データ・セット)  
再始動での使用 [130](#)  
説明 [72](#)  
定義および初期設定のためのジョブ制御ステートメント [72](#)  
CICS 実行のジョブ制御ステートメント [73](#)  
LIBRARY 定義  
定義のインストール [331](#)  
LOCK GROUP コマンド  
グループへのアクセスの制御 [13](#)  
LOCK LIST コマンド  
リストへのアクセスの制御 [13](#)  
LOGA 一時データ宛先 [149](#)  
LOGON

LOGON (続き)  
セキュリティ検査 [344](#)  
LSRPOOL 定義の更新 [478](#)  
LU [315](#)  
LU、定義 [315](#)  
LUTYPE 0 端末  
自動インストールに適合 [251](#)  
LUTYPE 1 端末  
自動インストールに適合 [251](#)  
LUTYPE 2 端末  
自動インストールに適合 [251](#)  
LUTYPE 3 端末  
自動インストールに適合 [251](#)  
LUTYPE 4 端末  
自動インストールに適合 [251](#)  
LUTYPE6.1  
CICS-CICS ISC リンクおよびセッション  
自動インストールに不適合 [251](#)  
CICS-IMS リンクおよびセッション  
自動インストールに不適合 [251](#)  
MRO セッション  
自動インストールに不適合 [251](#)

## M

MCT (モニター管理テーブル) [270, 278](#)  
MDLTAB (z/OS Communications Server のマクロ) [316](#)  
MEMLIMIT パラメーター [142](#)  
MGCR マクロ、MVS コマンドを発行するため [322](#)  
MNPS [322](#)  
MODIFY コマンド [320](#)  
MSGCASE、DFHXCOPT のパラメーター [409](#)  
MTS (モデル端末サポート) [261](#)  
MVS SDUMP マクロ [79](#)  
MVS START コマンド、CICS を開始する [152, 156](#)  
MVS コンソール  
自動インストール [258](#)  
MVS の考慮事項 [339, 344](#)

## N

NETNAME 属性  
自動インストールを使用したログオンでの役割 [247](#)  
TCT でのエントリーなし [247](#)  
NEWSIT、システム 初期設定パラメーター  
ウォーム・スタートへの影響 [131](#)  
NOPS [322](#)  
NTO (ネットワーク端末オプション) [251](#)  
NTO を使用した TLX 端末  
自動インストールに適合 [251](#)  
NTO を使用した TWX 端末  
自動インストールに適合 [251](#)

## O

OPIDENT オペランド  
DFHSNT  
グループおよびリストへのアクセスを制御するため [13](#)  
OPNDST、ログオン時に CICS により発行 [247](#)  
OSGi バンドル [285](#)  
OSGi リカバリー [309](#)

## P

PARM 始動パラメーター  
システム 初期設定パラメーター [146](#)  
PARMSRCE パラメーター [137](#)  
PARMTYPE パラメーター [137](#)  
PLATDEF [352, 358](#)  
PLATFORM [358](#)  
PLT (プログラム・リスト・テーブル) [270](#)  
PROTOCOL 属性  
CONNECTION 定義 [401](#)  
SESSIONS 定義 [402](#)  
PSDINT [322](#)  
PSTYPE [322](#)

## Q

QUERY 機能  
自動インストールあり [247](#)

## R

RACF  
セキュリティ・マネージャーとして使用 [343](#)  
RBA (相対バイト・アドレス) [53](#)  
RDO (オンライン・リソース定義)  
CICS システム定義データ・セット (CSD) (CICS system definition data set (CSD)) [21](#)  
RECEIVECOUNT 属性、SESSIONS 定義 [402](#)  
RECEIVEPFX 属性、SESSIONS 定義 [402](#)  
RECOVNOTIFY [322](#)  
RECOVOPTION [322](#)  
REGION パラメーター [142](#)  
Register from Data Set オプション [429](#)  
RELOAD、名前付きカウンター・プール [230](#)  
REMOTENAME パラメーター [417](#)  
REMOTESYSTEM パラメーター [417](#)  
RENTPGM、読み取り専用 DSA のストレージ [146](#)  
RESSEC [416](#)  
RESSEC (リソース・セキュリティ検査) [13](#)  
RESSEC 属性  
TRANSACTION [13](#)  
RESTART オプション (DEFINE TRANSACTION) [459](#)  
REXX Db2 インターフェースの構成 [483](#)  
RFS ファイル・プールの形式設定 [480](#)  
RFS ファイル・プールの作成 [477](#)  
RLS、レコード・レベル共有  
VSAM データ共有 [85](#)  
RST (リカバリー可能サービス・テーブル) [270](#)

## S

SAF、System Authorization Facility  
セキュリティ検査のための使用 [343](#)  
SCS プリンター  
自動インストールに適格 [251](#)  
SDLC 3767 デバイス  
自動インストールに適格 [251](#)  
SDUMP データ・セット [45](#)  
SDUMP マクロ [79](#)  
SENDcount 属性、SESSIONS 定義 [402](#)  
SENDPFX 属性、SESSIONS 定義 [403](#)  
SESSIONS 定義

SESSIONS 定義 (続き)  
PROTOCOL 属性 [402](#)  
RECEIVECOUNT 属性 [402](#)  
RECEIVEPFX 属性 [402](#)  
SENDcount 属性 [402](#)  
SENDPFX 属性 [403](#)  
SET FILE コマンド  
説明 [348](#)  
MAXNUMRECS パラメーター [348](#)  
TABLE パラメーター [348](#)  
SET コマンド、名前付きカウンター [227](#)  
SETSYS コマンド [484](#)  
SETXCF、名前付きカウンター・プールの削除 [229](#)  
SHAREOPTION、VSAM [346](#)  
SIT (システム 初期設定テーブル)  
APPLID (CICS システム名) [13](#)  
CICS に対するシステム 初期設定パラメーターの提供  
[125](#)  
DFHSIT TYPE=CSECT [112](#)  
DFHSIT TYPE=DSECT [112](#)  
DFHSIT キーワードおよびオペランド [112](#)  
SIT パラメーター  
情報の取り出し [137](#)  
SMS (ストレージ管理サブシステム) [46, 47](#)  
SMSS  
セキュリティのセットアップ [386](#)  
セットアップ [384](#)  
SNPS [322](#)  
SPURGE オプション (DEFINE TRANSACTION) [459](#)  
START オペランド、DFHSIT [324](#)  
START コマンド、MVS [152, 156](#)  
START、システム 初期設定パラメーター  
START=AUTO [131, 155](#)  
START=COLD [132, 156](#)  
START=INITIAL [132, 155](#)  
START=STANDBY [156](#)  
SVC  
始動ジョブ内での DFHCSVC の指定 [162](#)  
SYSIN データ・ストリーム [146](#)  
SYSPARM リソース・テーブル [137](#)  
System Authorization Facility [343](#)

## T

TCP/IP サービス  
DFH\$WUTC サンプル [384](#)  
TCT (端末管理テーブル)  
TCT 項目、自動インストール済み [246-248](#)  
TCT 項目  
自動 TCTTE 削除 [256](#)  
自動削除、自動インストールを使用 [256](#)  
ログオフ時に削除、自動インストール使用 [247](#)  
TCT ユーザー域 (TCTUA)  
自動インストールの効果 [246](#)  
TCTUA (TCT ユーザー域)  
自動インストールの効果 [246](#)  
TDQ [38](#)  
TDQUEUE [38](#)  
TDQUEUE 定義  
インストール [332](#)  
TERMINAL 定義  
インストール [333](#)  
サービス休止時のインストール [325](#)  
TERMINAL [251](#)



TERMINAL 定義 (続き)  
  TERMINAL 属性 [251](#)  
TIMEOUT、DFHXCOPT のパラメーター [409](#)  
TLT (端末リスト・テーブル)  
  自動インストールの効果 [247](#)  
TOR [352](#)  
TOR (端末所有領域) [3](#)  
TPURGE オプション (DEFINE TRANSACTION) [459](#)  
tranlog  
  JTA [458](#)  
TRANSACTION 定義  
  RESSEC 属性 [13](#)  
TSO ユーザー [321](#)  
TYPE、DFHXCOPT のパラメーター [407](#)  
TYPE=CSECT, DFHSIT [112](#)  
TYPE=DSECT, DFHSIT [112](#)  
TYPE=FINAL マクロ [276](#)  
TYPE=INITIAL マクロ [275](#)  
TYPETERM 定義 [248](#)

## U

UNLOAD、名前付きカウンター・プール [230](#)  
UNLOCK GROUP コマンド  
  グループへのアクセスの制御 [13](#)  
UNLOCK LIST コマンド  
  リストへのアクセスの制御 [13](#)  
URI マップ  
  DFH\$WUUR サンプル [384](#)

## V

VERBEXIT、IPCS パラメーター [79](#)  
VSAM  
  基本クラスター (base cluster) [346](#)  
  代替索引 [346](#)  
  SHAREOPTION [346](#)  
VSAM RLS  
  データ共用のため [340](#)  
  CMT に不適切 [346](#)  
  UMT に適切 [347](#)  
VSAM 区画内データ・セット [51](#)  
VSAM データ・セット  
  オープンとクローズ [91](#), [92](#)  
  空の VSAM データ・セットのロード [84](#)  
  ベースとパス [84](#)  
VSAM ファイル  
  順方向リカバリーに関する考慮事項 [460](#)

## W

Web サービス [471](#)  
WebSphere MQ 永続メッセージ [471](#)

## X

XCFGROUP、DFHXCOPT のパラメーター [410](#)  
XDR ルーチン  
  指定 [428](#)  
XDUCLSE、ダンプ・グローバル・ユーザー出口 [80](#), [82](#)  
XDUOUT、ダンプ・グローバル・ユーザー出口 [80](#), [82](#)  
XDUREQ、ダンプ・グローバル・ユーザー出口 [80](#), [82](#)  
XDUREQC、ダンプ・グローバル・ユーザー出口 [80](#), [82](#)

XES、名前付きカウンター・サーバーの応答 [227](#)  
XFCNREC  
  オープンの失敗のオーバーライド [465](#)  
XLT (トランザクション・リスト・テーブル) [272](#)  
XLT 定義 [417](#)  
XML ベースのサービス [285](#)  
XRES システム 初期設定パラメーター [310](#)

## Z

z/OS Communications Server  
  持続セッションのサポート [322](#)  
  セッション終了時の自動インストールによる TCT 項目の  
  削除 [247](#)  
  端末、ステートメント [316](#)  
  CICS 始動時の ACB [135](#)  
  CICS へのログオン [247](#)  
  NETNAME [247](#), [253](#)  
z/OS Communications Server 端末  
  自動インストール [245](#)  
z/OS コンソール、CICS に定義 [320](#)  
z/OZ Communications ServerSNA  
  および自動インストール [247](#)

## [特殊文字]

TCP/IP for MVS  
  CICS TCP/IP ソケット・インターフェース [413](#)  
相互通信のためのリソース定義 [12](#)  
ブリッジ  
  システム 初期設定パラメーター [391](#)  
  ブリッジ機能定義 [391](#)  
.STEPLIB、CICS ロード・ライブラリー [148](#)







