CICS Transaction Server for z/OS
バージョン 4 リリース 1

相互通信ガイド

SC88-5857-00
(英文原典：SC34-7018-00)

CICS Transaction Server for z/OS
バージョン 4 リリース 1

相互通信ガイド

SC88-5857-00
(英文原典：SC34-7018-00)

注

本書および本書で紹介する製品をご使用になる前に、 447 ページの『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、CICS Transaction Server for z/OS のバージョン 4 リリース 1 (プロダクト番号 5697-E93)、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： SC34-7018-00
CICS Transaction Server for z/OS
Version 4 Release 1
Intercommunication Guide

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

第1刷 2009.7

© Copyright International Business Machines Corporation 1977, 2009.

目次

前書き ix
本書について ix
本書で取り扱わない内容 ix
本書の対象読者 x
本書を理解するために必要な知識 x
本書の利用方法 x
本書の構成 x
| 用語 xi

CICS Transaction Server for z/OS バージョン 4 リリース 1の変更内容 . . . xiii

第 1 部 相互通信の概念と機能 1

第 1 章 CICS 相互通信の紹介 3
相互通信方式 3
システム間の通信. 3
複数領域操作 4
相互通信機能 4
機能シップ 5
非同期処理 6
トランザクション・ルーティング 6
分散プログラム・リンク (DPL) 7
分散トランザクション処理 (DTP) 7
CICS 相互通信機能の使用 7
地域本部の接続 10
組織内の各部門の接続 11

第 2 章 ISC および IPIC 相互通信機能 13
| IP interconnectivityを使用した相互通信 13
| IPIC を使用した相互通信機能 14
| 関連データと発信元データ 15
SNA を介したシステム間通信 20
| ISC を使用した相互通信機能 21
サブシステム間の接続. 21
システム間セッション. 23
システム間セッションの確立 26

第 3 章 複数領域操作 27
MRO を使用した相互通信機能 27
システム間複数領域操作 (XCF/MRO) 28
XCF/MRO の利点 31
複数領域操作の応用 32
プログラム開発 32
タイム・シェアリング. 32
信頼できるデータベース・アクセス 33
部門の分離 33
マルチプロセッサーのパフォーマンス 33
シスプレックスにおけるワークロード・バランシング. 34
仮想記憶域制約解放 34
単一領域システムからの変換 35

第 4 章 CICS 機能シップ 37
機能シップの概要 37
機能シップの設計上の考慮事項 38
ファイル制御 39
DL/I 39
一時記憶域 40
一時データ 40
システム間のキューイング 41
ミラー・トランザクションと変換プログラム . . . 42
ISC 機能シップ 42
MRO 機能シップ 44
ミラー・トランザクションのエラーと障害の処理 45
機能シップの例 46

第 5 章 非同期処理 51
非同期処理の概要 51
非同期処理方式 52
START と RETRIEVE コマンドを使用する非同期処理 53
リモート・トランザクションの開始と取り消し . 53
START コマンドによって渡される情報 54
システム間 START 要求のパフォーマンスの向上 55
作業単位への開始要求送達の組み込み 56
NOCHECK オプションを指定した START 要求の ISC リンクへの据え置き伝送 57
システム間のキューイング 58
開始されたトランザクションによるデータ検索 . 59
リモートで開始された CICS トランザクションによる端末の獲得 60
システム・プログラミングに関する考慮事項 . . . 60
非同期処理の例 60

第 6 章 CICS 動的ルーティングの紹介 65
動的ルーティングとは何か 65
2 つのルーティング・モデル 66
「ハブ」モデル 67
分散モデル 68
2 つのルーティング・プログラム 70

第 7 章 CICS トランザクション・ルーティング 73
トランザクション・ルーティングの概要. 73
トランザクション・ルーティングの開始. . . . 74
端末開始トランザクション・ルーティング 75
静的トランザクション・ルーティング 75
動的トランザクション・ルーティング 75
ATI によって開始されたトランザクションの従来のルーティング. 79

© Copyright IBM Corp. 1977, 2009

自動トランザクション開始用端末のシップ . . . 80
ATI および総称リソース 87
START コマンドで呼び出されたトランザクションのルーティング 87
拡張方法の利点 87
端末関連 START コマンドによって開始されたトランザクションのルーティング方法 88
非端末関連の START コマンド. 94
リモート APPC 接続の割り振り 97
APPC デバイスを使用したトランザクション・ルーティング 97
代替機能の割り振り 98
端末としてのシステム 98
中継プログラム. 100
基本マッピング・サポート (BMS) 101
リモート端末とオペレーターへの BMS メッセージ・ルーティング 101
ルーティング・トランザクション (CRTE) の使用 102
トランザクション・ルーティングのためのシステム・プログラミング 103
システム間のキューイング 103

第 8 章 CICS 分散プログラム・リンク 105
DPL の概要 105
DPL 要求の静的ルーティング 106
ミラー・トランザクションの使用. 107
グローバル・ユーザー出口による DPL 要求の宛先変更. 109
DPL 要求の動的ルーティング 109
動的にルーティングできる要求 110
動的ルーティング・プログラムを呼び出す場合 111
要求のルーティングのための CICSPlex SM の使用 111
DPL 要求の「デイジー・チェーン」. 112
DPL サーバー・プログラムでの制約事項 113
システム間のキューイング 113
DPL の例. 114

第 9 章 分散トランザクション処理 . . 117
DTP の概要 117
機能シップやトランザクション・ルーティングに対する利点 117
なぜ分散トランザクション処理なのか 119
会話とは何か、なぜ必要なのか 119
会話の開始とトランザクション階層 120
2 つのトランザクション間でのダイアログ. . . 121
制御フローと制御ブラケット 121
会話の状態とエラーの検出 122
同期 122
DTP では MRO か APPC か 124
APPC マップ式会話か、基本会話か 125
EXEC CICS か CPI コミュニケーションか . . . 126

第 2 部 相互通信サポートのインストールおよび構成 129

第 10 章 システム間通信の構成 131
TCP/IP ネットワークを介した通信のサポートの構成 131
SNA を介した ISC のサポートの構成 132

第 11 章 複数領域操作の構成 135
MRO サポートの提供 135
MRO 構成後の手順 135

第 12 章 VTAM 総称リソースの構成 137
VTAM 総称リソースの前提条件 137
VTAM 総称リソースを使用するための CICSplex の計画. 138
CICS 領域の命名 139
総称リソース環境における接続の定義 139
接続の定義 140
VTAM 総称リソース・サポートの生成. 141
総称リソースへの TOR のマイグレーション . . . 142
推奨される方法. 142
TOR を総称リソースから除去する 144
TOR を別の総称リソースへ移動する 145
総称リソース間でのシスプレックス間通信の設定 145
CICS TS for z/OS 総称リソース間での接続の確立 145
類縁性の終了 150
アフィニティーを終了すべき場合. 151
アフィニティーを終わらせるバッチ・プログラムの作成. 151
ATI での総称リソースの使用 154
ISSUE PASS コマンドの使用 157
規則のチェックリスト 158
特殊な事例の対処 159
非自動インストール端末と接続 159
アウトバウンドの LU6 接続 160

第 3 部 相互通信リソースの定義 163

第 13 章 リモート・システムへの接続の定義方法. 165
接続定義の紹介. 165
ローカル CICS 領域名 166
リモート・システムの識別 168
IP 相互接続 (IPIC) の定義 169
APPC 接続と MRO 接続を IPIC にマイグレーションする 171
複数領域操作のリンクの定義 181
MRO リンクの定義 181
MRO のアクセス方式の選択 183
互換 MRO ノードの定義 184
外部 CICS インターフェースが使用するリンクの定義 184
MRO と EXCI リンク定義のインストール . . 186
APPC 接続の定義 186
リモート APPC システムの定義 187
APPC セッション・グループの定義 188

互換 CICS APPC ノードの定義 189
APPC リンクの自動インストール. 190
単一セッション APPC 端末の定義 191
AUTOCONNECT オプション 192
APPC リンクでの VTAM 持続セッションの使用 194
論理装置タイプ 6.1 リンクの定義 195
CICS-IMS 間 LUTYPE6.1 リンクの定義 196
互換の CICS ノードと IMS ノードの定義. . . 197
IMS システムへの複数リンクの定義. 200
トランザクション・ルーティングのための間接リンクの定義 202
CICS Transaction Server for z/OS での間接リンクの定義 204
間接リンクを使用するトランザクション・ルーティングのためのリソース定義 205
XRF のための総称 applid と特定 applid 208

第 14 章 TCP/IP 管理および制御 . . . 209

第 15 章 APPC 接続の管理 213
APPC リンク管理に関する一般情報 213
接続の獲得 214
獲得プロセス中の接続状況 214
AUTOCONNECT オプションの効果 214
MAXIMUM オプションの効果. 216
SET MODENAME コマンドによるセッションの制御 216
コマンドの有効範囲と制約事項 217
接続の解放 218
解放プロセス中の接続状況 218
限定リソースの効果 219
接続を使用不能にする 219
APPC リンク管理の要約. 221
コマンドの有効範囲と制約事項 222

第 16 章 リモート・リソースの定義 223
どのリモート・リソースを定義しなければならないか 223
「デイジー・チェーン」に関する注記 224
リソースのローカル名とリモート名 224
機能シップのためのリモート・リソースの定義 . . 225
リモート・ファイルの定義 226
リモート DL/I PSB の定義. 227
リモート一時データ宛先の定義 228
リモート一時記憶域キューの定義. 228
DPL のリモート・リソースの定義 229
リモート・サーバー・プログラムの定義 . . . 229
リモート・サーバー・プログラムの定義が必要ない場合. 230
非同期処理のためのリモート・リソースの定義 . . 232
リモート・トランザクションの定義 232
トランザクション・ルーティングのためのリモート・リソースの定義 233
トランザクション・ルーティングにおける端末の定義 234
トランザクション・ルーティングのトランザクションの定義 244
DTP のリモート・リソースの定義 251

第 17 章 ローカル・リソースの定義 253
通信プロファイルの定義. 253
基本機能の通信プロファイル 254
デフォルト・プロファイル 254
デフォルト・プロファイルの修正. 255
体系化プロセス. 256
プロセス名 257
体系化プロセス定義の修正 257
インストールが必要なリソース定義の選択. . . . 258
区画内一時データ・キューの定義. 259
トランザクション 260
基本機能 260
DPL のローカル・リソースの定義 262
ミラー・トランザクション 262
サーバー・プログラム 262

第 4 部 システム間環境におけるアプリケーション・プログラミング . . 263

第 18 章 アプリケーション・プログラミングの概要. 265
用語 265
問題判別 266

第 19 章 CICS 機能シップのアプリケーション・プログラミング 267
機能シップのためのプログラミングの紹介. . . . 267
ファイル制御 268
DL/I 268
一時記憶域 268
一時データ 269
機能シップの例外条件 269
リモート・システム使用不能 269
無効な要求 269
ミラー・トランザクションの異常終了 269

第 20 章 CICS DPL のアプリケーション・プログラミング 271
DPL プログラミングの紹介. 271
クライアント・プログラム 271
サーバー・プログラムの障害 272
サーバー・プログラム 272
許可されているコマンド. 272
同期点. 272
DPL の例外条件 273
リモート・システム使用不能 273
サーバー作業のバックアウト 273
同じサーバー領域に対する複数のリンク . . . 273
ミラー・トランザクションの異常終了 274
同じ分散 UOW による 1 つのリカバリー可能リソースに対する複数の更新 274

第 21 章 非同期処理のアプリケーション・プログラミング 277
リモート・システムでのトランザクションの開始 277
START コマンドの例外条件 277
リモートで出された開始要求に関連したデータの検索 278

第 22 章 CICS トランザクション・ルーティングのアプリケーション・プログラミング 279
アプリケーション・プログラミングの制限. . . . 279
基本マッピング・サポート 279
疑似会話型トランザクション 280
アプリケーション所有領域で EXEC CICS ASSIGN コマンドによって返された値を確認する 280

第 23 章 CICS-IMS 間アプリケーション 283
CICS-IMS 間 ISC アプリケーションの設計 . . . 283
データ形式 283
IMS とのシステム間通信の形式 285
CICS-IMS 間アプリケーション - 非同期処理 . . . 285
START と RETRIEVE インターフェース . . . 285
非同期の SEND と RECEIVE インターフェース 290
CICS-IMS 間アプリケーション - DTP 291
CICS-IMS 間セッションに対する CICS コマンド 291
フロントエンド・トランザクションに関する考慮事項 292
リモート・トランザクションの接続 293
バックエンド・トランザクションに関する考慮事項 297
会話 298
セッションの解放 299
EXEC インターフェース・ブロック (EIB). . . 299
CICS-IMS 間セッションのコマンド順序 . . . 301
状態遷移 302

第 5 部 システム間環境におけるパフォーマンス 305

第 24 章 システム間のセッション・キューの管理. 307
セッション・キュー管理の概要 307
割り振りキューの管理 307
リソース定義を使用してキューを管理する. . . 307
NOQUEUE オプションの使用 308
XISQUE および XZIQUE グローバル・ユーザー出口の使用 308

第 25 章 シップされた端末定義の効率的な削除. 311
シップされた端末の削除方法の概要 311
選択的削除 311
タイムアウト削除メカニズム 312
タイムアウト削除の実装. 312
タイムアウト削除のパフォーマンスのチューニング 313
DSHIPIDL 313
DSHIPINT 314

第 6 部 システム間環境におけるリカバリーおよび再始動 315

第 26 章 相互接続されたシステムにおけるリカバリーと再始動 317
用語 317
同期点交換 318
同期点フロー 319
リカバリーの機能とインターフェース 321
リカバリー機能. 322
リカバリー・インターフェース 322
初期始動とコールド・スタート 326
コールド・スタートがいつ可能かの決定 . . . 327
ログ名交換プロセス 328
接続定義の管理. 329
CICS TS for z/OS システムへの MRO 接続と IPIC 接続. 330
CICS TS for z/OS システムへの APPC 並列セッション接続 330
VTAM 総称リソースとの間の APPC 接続. . . 330
棚上げを完全にはサポートしない接続 331
LU6.1 接続 331
非 CICS TS for z/OS システムへの APPC 接続 333
APPC 単一セッション接続 333
APPC 接続静止処理 334
問題判別 334
CICS のリカバリー・アクションを知らせるメッセージ. 334
問題判別の例 338

第 27 章 相互通信と XRF 349
MRO セッション 349
LUTYPE6.1 セッション 349
単一セッション APPC デバイス 350
並列 APPC セッション 350
アプリケーション・プログラムへの影響 350

第 28 章 相互通信と VTAM 持続セッション 351
持続セッションのサポートと XRF との比較 . . . 351
相互接続された CICS の環境、リカバリー、および再始動. 352

第 7 部 システム間環境におけるデータ変換 355

第 29 章 データが変換される場所 . . . 357
機能シップと DPL. 357

分散トランザクション処理 358
トランザクション・ルーティング. 358

第 30 章 データ変換の回避. 359

第 31 章 変換の種類 361

第 32 章 文字データ 363

第 33 章 バイナリー・データ. 365

第 34 章 CICS がサポートされている変換 367
アラビア語 368
Baltic Rim 369
キリル文字 369
デーバナーガリー 370
ペルシア語 370
ギリシャ語 371
ヘブライ語 371
日本語. 372
韓国語. 373
ラオ語. 374
Latin 1 と Latin 9. 374
Latin 2 376
Latin 5 377
中国語 (簡体字) 377
タイ語. 378
中国語 (繁体字) 378
ウルドゥー語 379
ベトナム語 379
Unicode データ. 380

第 35 章 変換処理 381
コンポーネント. 381
処理 381
標準の変換と非標準の変換 382
CICS による変換 382
ユーザー/CICS 変換 382
ユーザーによる変換 383
変換処理の順序. 383

第 36 章 データ変換を可能にするためのリソース定義 387

第 37 章 変換テーブルの定義. 389
DFHCNV マクロのタイプ 389
変換テンプレートとキー・テンプレート . . . 390
クライアント・コード・ページおよびサーバー・コード・ページのデフォルト 390
初期プログラム検査 (IVP) 用の変換テーブル 391
DFHCNV TYPE=INITIAL 392
DFHCNV TYPE=ENTRY. 394
DFHCNV TYPE=KEY. 397
DFHCNV TYPE=SELECT 398
DFHCNV TYPE=FIELD 399
DFHCNV TYPE=FINAL 400
マクロのコード化に関するヒント. 401

第 38 章 ユーザー定義の変換テーブル 403
無効な、または未定義の DBCS 文字 406

第 39 章 マクロの例 407

第 40 章 変換プログラムのアセンブルとリンク・エディット 411

第 41 章 ユーザー置き換え可能変換プログラム. 413
ユーザー指定の変換プログラム 413
DFHUCNV への入力 413
パラメーター・リスト (DFHUVNDS) 413
変換テンプレートとキー・テンプレート . . . 417
フィールド変換レコード. 418
提供されているユーザー置き換え可能変換プログラム 420
ユーザー置き換え可能変換プログラム 421

第 8 部 付録 427

付録 A. 相互通信規則と制約事項のチェックリスト. 429
トランザクション・ルーティング. 429
DPL 要求の動的ルーティング 431
自動トランザクション開始 432
基本マッピング・サポート 432
LUTYPE6.1 セッションの獲得. 432
同期点処理 432
ローカル名とリモート名. 432
マスター端末トランザクション 433
インストールと操作 433
リソース定義 433
カスタマイズ 433
MRO 異常終了コード 434

付録 B. CICS の APPC アーキテクチャーへのマッピング 435
サポートされるオプション・セット 435
制御オペレーター verb の CICS による実装 . . . 437
制御オペレーターの verb 437
制御オペレーター verb の戻りコード 444
APPC アーキテクチャーからの逸脱 445
APPC トランザクション・ルーティングの APPC アーキテクチャーからの逸脱 445

特記事項. 447
商標 448

参考文献. 449
CICS Transaction Server for z/OS の CICS の資料 449

CICS Transaction Server for z/OS の SCICSPlex SM の資料 450
その他の CICS の資料 450
その他の IBM の資料 451

アクセシビリティー 453

索引 455

前書き

本書について

この資料には、プログラムを作成するユーザーが バージョン 4 リリース 1 のサービスを使用するためのプログラム・インターフェースが記述されています。

この資料の内容は、以下のとおりです。

- 複数領域操作 (MRO): IBM® システム・ネットワーク体系 (SNA) のネットワーキング機能を使用しない、同じオペレーティング・システム内、または同じ MVS™ シスプレックス内の CICS® 領域間通信。[1]
- SNA を介したシステム間通信 (SNA を介した ISC): IBM CICS Transaction Server for z/OS® 領域と、SNA の論理装置タイプ 6.2 または 論理装置タイプ 6.1 のプロトコルをサポートする他の (CICS および非 CICS) システムまたは端末との間の通信。論理装置タイプ 6.2 のプロトコルは、拡張プログラム間通信機能 (APPC) とも呼ばれます。リモート・システムは、CICS と同じ MVS シスプレックス内にあってもなくても構いません。
- IP interconnectivity (IPIC): IBM CICS Transaction Server for z/OS 領域と、Transport Control Protocol/Internet Protocol (TCP/IP) をサポートする他の (CICS および非 CICS) システムまたは端末との間の通信。リモート・システムは、CICS と同じ MVS シスプレックス内にあってもなくても構いません。

本書で取り扱わない内容

本書に記載された情報は、一部を除いて、その大部分が CICS Transaction Server for z/OS バージョン 4 リリース 1 と、他の System/390® CICS システムまたはIMS™ システムとの間の通信に関するものです。CICS TS for z/OS バージョン 4.1 と非 System/390 CICS システムの間の通信に関する補足情報については、「*CICS Family: Communicating from CICS on System/390*」を参照してください。

注: 本書では、*System/390* という句を、System/370™、System/390、および zSeries® ファミリーのコンピューターを表す総称用語として使用しています。

他の CICS 製品で提供されている相互通信機能についての概説は、「*CICS Family: Interproduct Communication*」を参照してください。

CICS のプログラムおよびトランザクションに対するインターネットからのアクセスについては、「*CICS インターネット・ガイド*」を参照してください。CICS のプログラムおよびトランザクションに対する他の非 CICS 環境からのアクセスについては、「*CICS External Interfaces Guide*」を参照してください。

CICS クライアント・ワークステーション製品の CICS サポートについては、「*CICS Family: Communicating from CICS on System/390*」を参照してください。

1. CICS 外部インターフェース (EXCI) では、特殊な形式の MRO リンクを使用して、MVS バッチ・プログラムと CICS 間の通信、および CICS プログラムに対する DCE リモート・プロシージャー・コールをサポートします。

© Copyright IBM Corp. 1977, 2009

CICS Business Transaction Services (BTS) での相互通信に関する情報は、「*CICS Business Transaction Services*」を参照してください。

CICS Front End Programming Interface については、「*CICS Front End Programming Interface User's Guide*」を参照してください。

分散トランザクション・プログラミングについては、「*CICS Distributed Transaction Programming Guide*」を参照してください。

本書の対象読者

本書は、CICS SNA を介したシステム間通信 (SNA を介した ISC)、IP interconnectivity (IPIC)、または複数領域操作 (MRO) の計画および実装に携わるお客様を対象にしています。

本書を理解するために必要な知識

本書は、お客様に単一の CICS システムの使用経験があるという前提の下に作成されています。本書には、特に複数システム環境の場合に当てはまる情報が記載されており、単一の CICS システムの概念や機能についての知識は既に持っているということが基本前提になっています。

また、SNA に関する概念および用語を理解していることも前提となっています。IPIC ネットワークを作成する予定の場合は、TCP/IP の知識が必要になります。

注: 本書で使用される用語「MVS」は、z/OS の基本制御プログラム (BCP) が備えているサービスおよび機能を表します。BCP は z/OS の基本エレメントです。

本書の利用方法

まず、本書のパート 1 を読み、CICS の複数領域操作およびシステム間通信の概念を理解することをお勧めします。

その後、所要の作業についてのガイダンスおよび参照資料として、本書の個々のパートにお進みください。

本書の構成

本書の構成は、以下のとおりです。

『**相互通信の概念と機能**』では、CICS 相互通信について紹介し、使用可能な機能について説明しています。評価および計画を目的としたパートです。

『**相互通信サポートのインストール**』では、CICS のインストールに関して、特に相互通信に当てはまる内容について説明しています。また、IMS システム定義に関するいくつかの注意点も記載しています。このパートは、「*CICS Transaction Server for z/OS インストール・ガイド*」および「*CICS System Definition Guide*」と併せてお読みください。

『**相互通信リソースの定義**』には、リソース定義のガイダンスを記載しています。リモート・システムへのリンクの定義方法、リモート・リソースの定義方法、およ

び相互通信環境で必要となるローカル・リソースの定義方法が説明されています。このパートは、「*CICS Resource Definition Guide*」と併せてお読みください。

『システム間環境におけるアプリケーション・プログラミング』では、CICS 相互通信機能を使用するアプリケーション・プログラムの作成方法について説明しています。このパートは、「*CICS アプリケーション・プログラミング・ガイド*」および「*CICS アプリケーション・プログラミング・リファレンス*」と併せてお読みください。

『システム間環境におけるパフォーマンス』では、パフォーマンスに関して、特に相互通信環境に当てはまる内容について説明しています。このパートは、「*CICS パフォーマンス・ガイド*」と併せてお読みください。

『システム間環境におけるリカバリーおよび再始動』では、リカバリーおよび再始動に関して、特に相互通信環境に当てはまる内容について説明しています。このパートは、「*CICS Recovery and Restart Guide*」と併せてお読みください。

用語

特に言及されていない限り、本書では以下の用語を使用します。

1. 用語「*CICS*」は、CICS Transaction Server for z/OS バージョン 4 リリース 1 を意味します。他の CICS 製品を指している場合には、それぞれの個所でそのことが明示されます。
2. 用語「システム*間通信*」および「*ISC*」は、SNA を介したシステム間通信 (SNA を介した ISC) および IP interconnectivity (IPIC) を意味する総称名です。SNA を介した ISC と IPIC のいずれかを指している場合には、そのことが明示されます。

   2 つのタイプの ISC についての説明は、3 ページの『システム間の通信』を参照してください。
3. 用語「*IP 接続*」は、IP interconnectivityを意味します。
4. 用語「*MVS*」は、z/OS の基本制御プログラム (BCP) が備えているサービスおよび機能を表します。BCP は z/OS の基本エレメントです。

CICS Transaction Server for z/OS バージョン 4 リリース 1の変更内容

このリリースに加えられた変更内容については、インフォメーション・センターか、以下の資料の*新機能 (What's New)*を参照してください。

- *CICS Transaction Server for z/OS リリース・ガイド*
- *CICS Transaction Server for z/OS V3.2 からのアップグレード*
- *CICS Transaction Server for z/OS V3.1 からのアップグレード*
- *CICS Transaction Server for z/OS V2.3 からのアップグレード*

© Copyright IBM Corp. 1977, 2009

第 1 部 相互通信の概念と機能

このセクションでは、CICS 相互通信の基本概念と提供される各種機能について説明します。

3 ページの『第 1 章 CICS 相互通信の紹介』では、CICS **相互間通信**を定義し、**複数領域操作**と**システム間通信**について説明します。さらに CICS の提供する基本的な相互通信機能について説明します。これらの機能には、次のものがあります。

- 機能シップ
- 非同期処理
- トランザクション・ルーティング
- 分散プログラム・リンク (DPL)
- 分散トランザクション処理 (DTP)

以下のセクションでは、これらの各概念について詳しく説明します。

- 27 ページの『第 3 章 複数領域操作』
- 13 ページの『第 2 章 ISC および IPIC 相互通信機能』
- 37 ページの『第 4 章 CICS 機能シップ』
- 51 ページの『第 5 章 非同期処理』
- 65 ページの『第 6 章 CICS 動的ルーティングの紹介』
- 73 ページの『第 7 章 CICS トランザクション・ルーティング』
- 105 ページの『第 8 章 CICS 分散プログラム・リンク』
- 117 ページの『第 9 章 分散トランザクション処理』.

© Copyright IBM Corp. 1977, 2009

第 1 章 CICS 相互通信の紹介

本書は、読者が単独システムとしての CICS の使用、関連データ・リソース、および端末ネットワークをよく理解していることを前提としています。この情報は、複数システム環境における CICS の役割に関係しています。この環境において、CICS は類似の通信機能を備えた他のシステムと通信を行うことができます。この種の通信を *CICS 相互通信*と呼びます。

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。CICS Transaction Server for z/OS の CICS クライアント・ワークステーション・プロダクトのサポートについては、「*CICS Family: Communicating from CICS on zSeries*」を参照してください。

インターネットからの CICS プログラムおよびトランザクションへのアクセスについては、 を参照してください。 *CICS インターネット・ガイド*. 非 CICS 環境からの CICS プログラムおよびトランザクションへのアクセスについては、 「*CICS External Interfaces Guide*」を参照してください。

このセクションには、以下のトピックが含まれています。

- 『相互通信方式』
- 4 ページの『相互通信機能』
- 7 ページの『CICS 相互通信機能の使用』.

相互通信方式

CICS では、複数領域操作 (MRO) を使用して、同じオペレーティング・システムまたはシスプレックス内にある他のシステムと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ない他の CICS システムまたは非 CICS システムと通信するために、CICS では TCP/IP (IPIC) または SNA (ISC over SNA) プロトコルを使用して接続します。

システム間の通信

CICS システムと非 CICS システムとの通信、または同じオペレーティング・システムか同じ z/OS シスプレックスにない CICS システム間の通信には、通常、必要な通信プロトコルを提供するために、ネットワーク・アクセス方式が必要です。

CICS TS for z/OS バージョン 4.1 は、以下のような 2 つの*相互通信機能*をサポートしています。

1. Transport Control Protocol/Internet Protocol (TCP/IP)
2. IBM システム・ネットワーク体系 (SNA) を実装する ACF/VTAM

TCP/IP 経由のシステム間の通信は、*IP 相互接続 (IPIC)* と呼ばれます。SNA 経由のシステム間通信の一般名は、*システム間通信* または *ISC over SNA* です。

© Copyright IBM Corp. 1977, 2009

IPIC および ISC は CICS システムと非 CICS システム、または同じ z/OS イメージまたはシスプレックス内にない複数の CICS システムを接続する際に使用されます。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。例えば、同じシスプレックス内の 2 つの CICS 領域間に 2 つの接続が必要で、MRO 接続が既に存在する場合に、その 2 つの CICS 領域間で ISC 接続を作成するときなどです。

関連概念

14 ページの『IPIC を使用した相互通信機能』
IP 相互通信 (IPIC) を使用することにより、TCP/IP ネットワークを使用している CICS システム間で通信することができます。

21 ページの『ISC を使用した相互通信機能』
SNA を介したシステム間通信 (SNA を介した ISC) 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。

13 ページの『第 2 章 ISC および IPIC 相互通信機能』
CICS は、外部システムとの通信を可能にするために、SNA を介したシステム間通信 (SNA を介した ISC) および IP interconnectivity (IPIC) に相互接続機能を提供しています。

複数領域操作

CICS には、CICS-CICS 間通信のために、ACF/VTAM または TCP/IP などのネットワーク・アクセス方式の使用を必要としない、**領域間通信**機能が用意されています。

この形式の通信を**複数領域操作** (MRO) と呼びます。MRO は、次の CICS 領域同士の間で使用することができます。

- 同じ z/OS イメージ
- 同じ z/OS 複合システム (**シスプレックス**)

CICS Transaction Server for z/OS は、MRO を使用して以下のシステムと通信することができます。

- その他の CICS Transaction Server for z/OS システム
- CICS Transaction Server for OS/390® システム

注: 外部 CICS インターフェース (EXCI) では、次の機能をサポートするために特殊形式の MRO リンクが使用されます。

- MVS バッチ・プログラムと CICS との間の通信
- CICS プログラムへの DCE リモート手続き呼び出し

相互通信機能

複数システム環境において、各参加システムは、独自のローカル端末とデータベースを持つことができ、そのローカル・アプリケーション・プログラムを、ネットワーク内の他のシステムとは無関係に実行することができます。

また、参加システムでは、他のシステムへのリンクを確立することによって、リモート・リソースにアクセスすることができます。このメカニズムにより、参加システム間でリソースを分散して、共用することができます。

他の CICS、IMS、または他のシステムと通信するために、CICS には以下の機能が備わっています。

- 機能シップ
- 非同期処理
- トランザクション・ルーティング
- 分散プログラム・リンク (DPL)
- 分散トランザクション処理 (DTP)

多数の相互通信機能。この機能は、非 CICS 環境からの CICS プログラムおよびトランザクションへのアクセスをサポートしています。これについては、「*CICS External Interfaces Guide*」のCICS トランザクションおよびプログラムへのインターフェース、および「*CICS インターネット・ガイド*」のCICS および HTTP で説明されています。

これらの通信機能のすべてが、あらゆる形式の相互通信で使用できるわけではありません。これらの機能を使用できる条件は、表 1 のとおりです。

表 1. CICS 基本相互通信機能のサポート (他の CICS、IMS、APPC、または TCP/IP システムと通信する場合)

	IRC 領域間通信	**システム間通信 SNA 経由 (ACF/VTAM を使用)**				**システム間通信 TCP/IP 経由**	
	MRO	**LUTYPE6.2 (APPC)**		**LUTYPE6.1**		**IPIC**	
機能	**CICS**	**CICS**	**非 CICS (例、CTG)**	**CICS**	**IMS**	**CICS**	**非 CICS (例、CTG)**
機能シップ	あり	あり	なし	あり	なし	なし	なし
非同期処理	あり	あり	なし	あり	あり	なし	なし
トランザクション・ルーティング	あり	あり	なし	なし	なし	あり	あり
分散プログラム・リンク	あり	あり	なし	なし	なし	あり	あり
分散トランザクション処理	あり	あり	あり	あり	あり	なし	なし

機能シップ

CICS で機能シップを使用すると、アプリケーション・プログラムで、他の CICS システムが所有するか、またはアクセスできるリソースに、アクセスすることができます。読み取りと書き込みの両方のアクセスが可能で、排他制御とリカバリー / 再始動を行う機能があります。

アクセスできるリモート・リソースは次のとおりです。

- ファイル
- DL/I データベース

- 一時データ・キュー
- 一時記憶域キュー

リモート・リソースにアクセスするアプリケーション・プログラムは、トランザクションの実行システムがそれらのリソースを所有するかのように設計して、コード化することができます。実行中、CICS は該当するシステムに対して要求をシップします。

機能シップは、MRO、または ISC over SNA リンクによって接続された CICS システム間でサポートされています。IPIC は、非端末 EXEC CICS START、START CHANNEL、および CANCEL の各コマンドの機能シップ (CICS TS 4.1 領域間) をサポートしています。

非同期処理

非同期処理を使用すると、CICS トランザクションから、リモート・システムのトランザクションを開始して、そのトランザクションにデータを渡すことができます。さらに、このリモート・トランザクションから、ローカル・システムでトランザクションを開始して、応答を受け取ることができます。

応答は、リモート・トランザクションを開始した**タスク**に必ず返されるわけではないため、要求と応答を直接結び付けることはできません (データ内のユーザー定義フィールドによる場合を除く)。そのため、この処理は**非同期**処理と呼ばれます。

非同期処理は、MRO、または ISC over SNA リンクによって接続された CICS システム間でサポートされています。IPIC は、非同期に行われる非端末 EXEC CICS START、START CHANNEL、および CANCEL の各コマンドの機能シップ (CICS TS 4.1 領域間) に制限される処理をサポートしています。

トランザクション・ルーティング

トランザクション・ルーティングを使用すると、異なる複数の CICS システムが、1 つのトランザクションとそれに関連する端末を所有することができます。

トランザクション・ルーティングでは、次のことが可能になります。

- ある CICS システムが所有する端末で、他の CICS システムが所有するトランザクションを実行することができます。
- 自動トランザクション開始 (ATI) によって開始されたトランザクションが、他の CICS システムの所有する端末を獲得することができます。
- ある CICS システムで実行中のトランザクションが、他の CICS システムの所有する APPC デバイスに対して、セッションを割り振ることができます。

トランザクション・ルーティングは、MRO、または ISC over SNA リンクによって接続された CICS システム間でサポートされています。IPIC は、3270 端末の従来のトランザクション・ルーティング (この端末では、端末専有領域 (TOR) は APPLID によって CICS TS 4.1 領域間で固有に識別されます) をサポートしています。

分散プログラム・リンク (DPL)

CICS 分散プログラム・リンクを使用すれば、CICS プログラム (クライアント・プログラム) から、リモートの CICS 領域にある他の CICS プログラム (サーバー・プログラム) を呼び出すことができます。

CICS 分散プログラム・リンクを使用すれば、CICS プログラム (クライアント・プログラム) から、リモートの CICS 領域にある他の CICS プログラム (サーバー・プログラム) を呼び出すことができます。次に、DPL を使用する形のアプリケーションを設計する理由のいくつかを示します。

- エンド・ユーザー・インターフェース (例えば、BMS 画面の処理) を、データのアクセスおよび処理などのアプリケーション・ビジネス論理から分離することによって、ホストからワークステーションへのアプリケーションの部分的な移植を容易にするため。
- アクセスするリソースの近くでプログラムを実行することにより、機能シップ要求が必要になる頻度を減らして、パフォーマンスを向上させるため。
- 多くの場合、DPL は、分散トランザクション処理 (DTP) アプリケーションの作成に代わる簡単な方法であるため。

DPL は、MRO、または ISC over SNA リンクによって接続された CICS システム間でサポートされています。IPIC は、以下の DPL 呼び出しをサポートしています。

- CICS TS 3.2 以降の領域との間の分散プログラム・リンク (DPL) 呼び出し
- CICS TS および TXSeries バージョン 7.1 との間の分散プログラム・リンク (DPL) 呼び出し

.

分散トランザクション処理 (DTP)

トランザクションの機能をネットワーク内のいくつかのトランザクション・プログラムに分散するこの技法を、**分散トランザクション処理 (DTP)** と言います。DTP を使用すると、CICS トランザクションは、他のシステムで実行されているトランザクションと通信を行うことができます。トランザクションは、相互に通信を行うために、システム間リンクを最も効率的に使用するように設計され、コード化されます。

DTP における通信は、CICS からみれば**同期処理**です。つまりこの通信は、CICS トランザクションの 1 回の呼び出し中に起こり、2 つのトランザクション間の要求と応答を直接対応付けることができます。これは、前に説明した非同期処理とは対照的です。

DTP は、MRO、または ISC over SNA リンクによって接続された CICS システム間でサポートされています。

CICS 相互通信機能の使用

CICS 相互通信機能を使用すると、多数の異なるタイプの分散トランザクション処理を実行することができます。ここでは、代表的な応用例をいくつか示します。

複数領域操作を行うと、2 つの CICS 領域が、選択されたシステム・リソースを共用する一方で、端末オペレーターに対しては「単一システム」としての外観を示すことができます。また同時に、各領域を相互に無関係な状態で実行して、一方の領域で起こったエラーからもう一方の領域を保護することができます。MRO の各種の応用方法については、 27 ページの『第 3 章 複数領域操作』で説明します。

ACF/VTAM アクセス方式と ACF/NCP/VS ネットワーク制御を使用する SNA を介した ISC では、異なる複数のシステム間でリソースを分散および共用することができます。これらのシステムが物理的に同じ場所にあるかどうかは関係ありません。

IPIC 接続を使用すると、システム間の相互通信に TCP/IP ネットワークを用いることができます。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。

9 ページの図 1 に、いくつかの代表的な形態を示します。

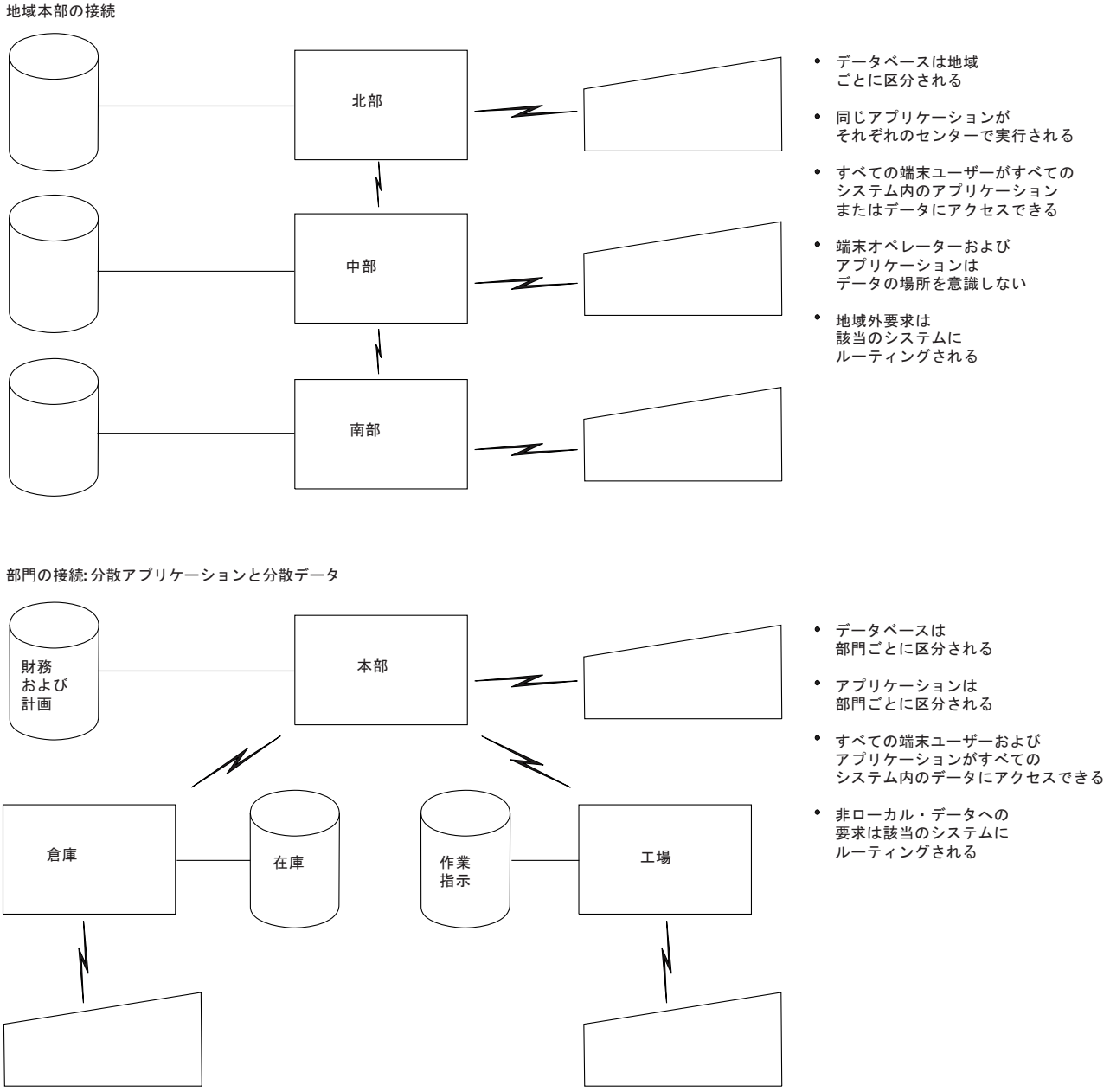
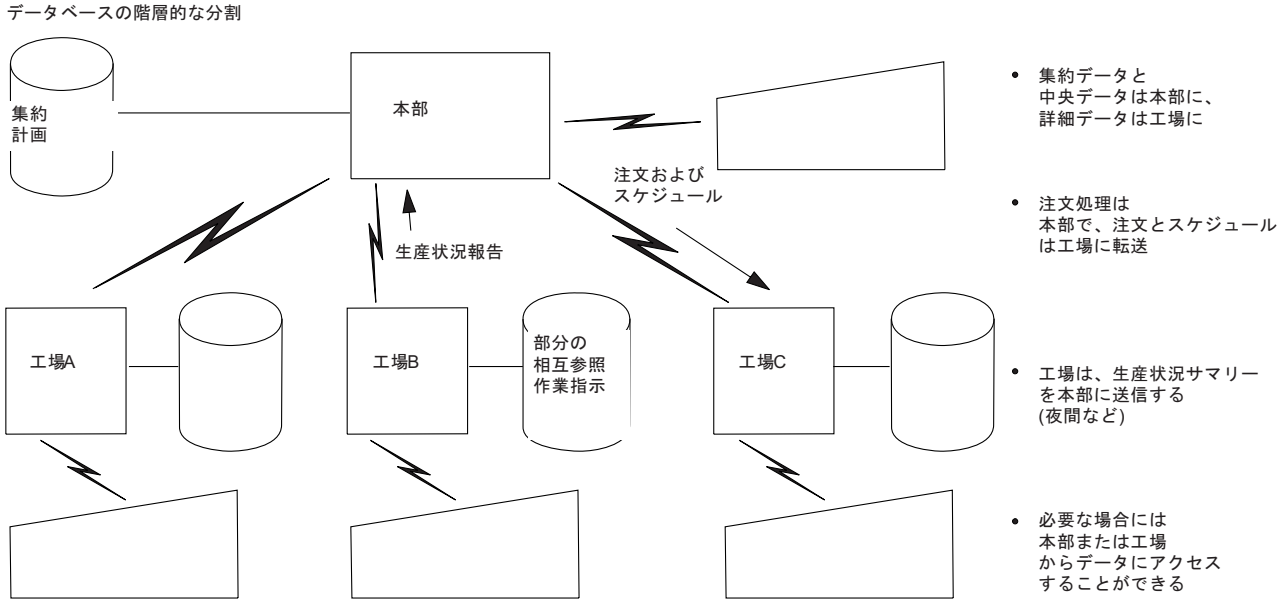


図 1. 分散リソースの例 (パート 1)



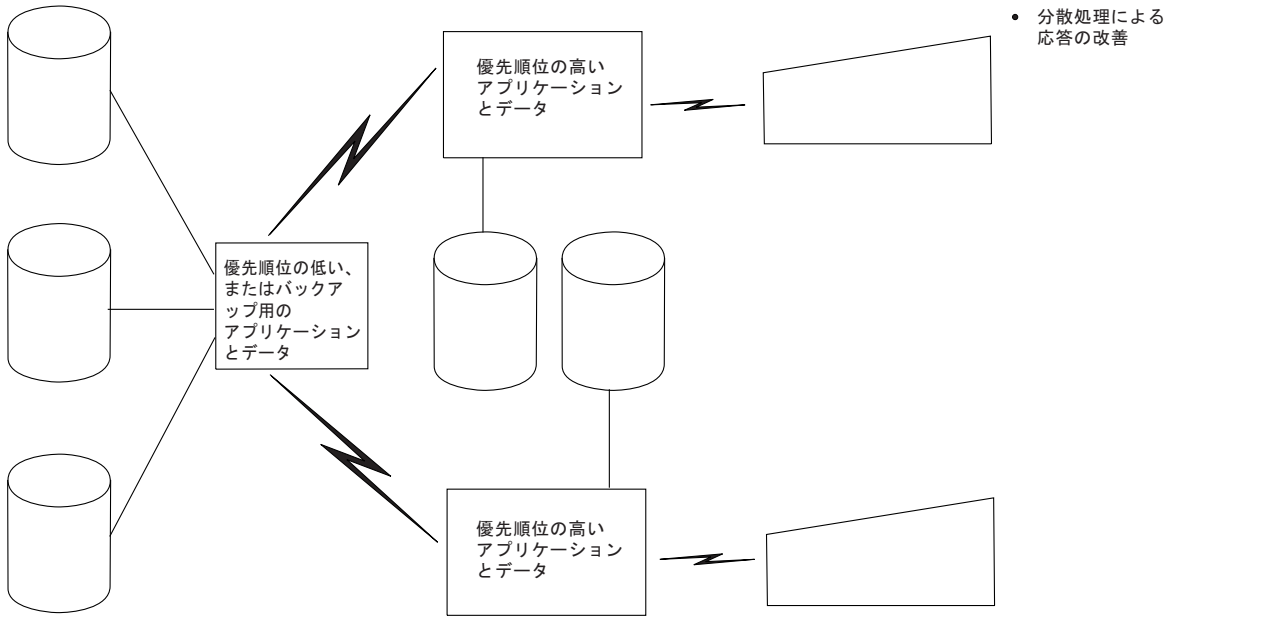


図 2. 分散リソースの例 (パート 2)

地域本部の接続

ユーザーの多くは、その主要な活動地域ごとに、コンピューター操作を設定しています。

各システムには、その地域の活動に合わせて編成されたデータベースと、その地域のデータベースの照会または更新を行うことができる独自の端末ネットワークがあります。ある地域の要求が他の地域のデータを必要とする場合、システム間通信機

能がないと、手作業手順を使用してこのような要求を処理しなければなりません。システム間通信機能を使用すると、該当する地域のデータベースへのファイル・アクセスを行うことにより、これらの「地域外」要求を自動的に処理することができます。

CICS 機能シップを使用すると、データの実際の位置に関係なくアプリケーション・プログラムを作成して、それをどの地域本部でも実行することができます。このタイプのアプリケーションの例として、顧客アカウントに対する貸方の検証があります。

組織内の各部門の接続

ユーザーによっては部門別に編成され、部門 (技術開発、生産、倉庫など) ごとに、独立したシステム、端末、およびデータベースを備えている場合があります。これらの部門を相互に、および本社に接続すると、プログラムやデータへのアクセスが容易になるため、企業内の協調性を高めることができます。

アプリケーションおよびデータは階層構造にして、本社側に要約データと中央データ、および工場側に詳細データを持たせることができます。あるいは、アプリケーションとデータを部門間で分散させて、本社側に計画や財務に関するデータとアプリケーション、工場側に製造に関するデータとアプリケーション、および倉庫側に在庫に関するデータとアプリケーションを持たせることもできます。いずれの場合も、ある部門のアプリケーションが、他の部門のデータに必要に応じてアクセスしたり、あるいは必要なデータを含むリモート部門でのアプリケーションの実行を求めて、準備できたときに要求側の部門へ応答を返すように要求したりすることができます。

第 2 章 ISC および IPIC 相互通信機能

CICS は、外部システムとの通信を可能にするために、SNA を介したシステム間通信 (SNA を介した ISC) および IP interconnectivity (IPIC) に相互接続機能を提供しています。

この章には、以下のトピックが含まれています。

- 20 ページの『SNA を介したシステム間通信』
- 『IP interconnectivityを使用した相互通信』

IP interconnectivityを使用した相互通信

CICS では、Transmission Control Protocol/Internet Protocol (TCP/IP) ネットワークを介してシステム間連絡を行うことができます。この形式の通信は IP interconnectivity または IPIC と呼ばれます。

IPIC 接続の要件

IPIC 接続を作成する前に、接続する各 CICS 領域で TCP/IP サービスをアクティブにする必要があります。

IPIC 接続は、IPCONN 定義および TCPIPSERVICE 定義という 2 つの補完リソースで構成されています。これらは、接続する各 CICS 領域にインストールする必要があります。IPCONN 定義は、アウトバウンド TCP/IP 通信リンクを示す CICS リソースです。IPCONN という用語は、一般に、IPIC 接続を指す言葉として使用されます。接続のインバウンド属性は、TCPIPSERVICE 定義で指定されます。TCPIPSERVICE リソースの名前は、IPCONN 定義の TCPIPSERVICE オプションで指定されます。

14 ページの図 3 は、IPCONN 定義と TCPIPSERVICE 定義との関係を示しています。

© Copyright IBM Corp. 1977, 2009

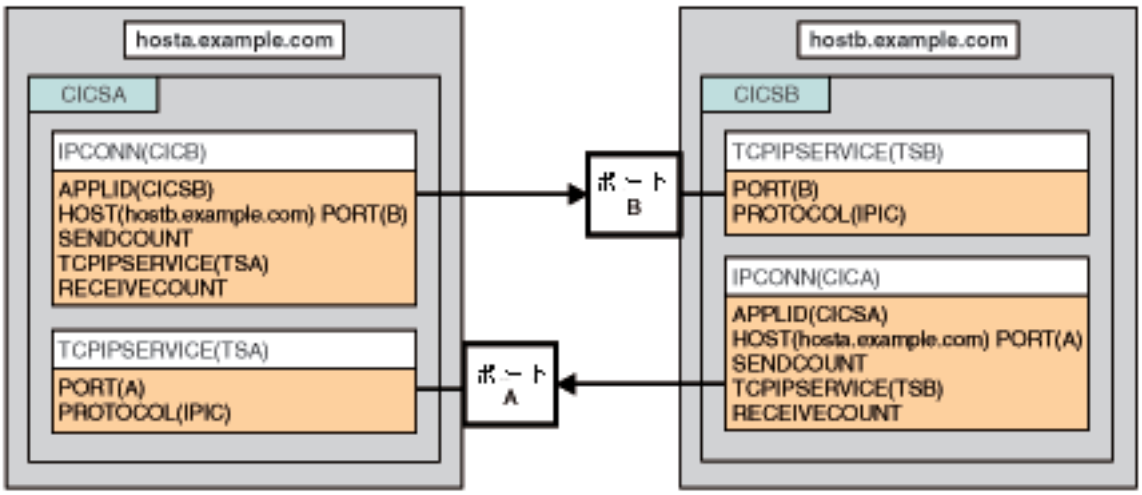


図 3. 関連した IPCONN 定義と TCPIPSERVICE 定義

同期レベル

IPIC による接続は、同期レベル 2 をサポートします。つまり、ロールバックを含む CICS の完全な同期点処理をサポートします。

ソケットの能力

CICS TS 4.1 システムから CICS TS 4.1 システムへの接続では、IPIC 通信に最大で 2 つのソケットを使用できます。CICS TS 4.1 システムから CICS TS 3.2 システムへの接続では、IPIC 通信に 1 つのソケットを使用できます。ネットワーク・エラーなどが原因で、IPCONN が使用していたソケットが 1 つ以上無効になった場合は、すべてのソケットが無効になり、IPCONN 接続が解放されます。

IPIC を使用した TCP/IP 接続バランシング (TCP/IP ポート共用など) を実行することはサポートされていないため、IPIC 接続の獲得の試行中に予期しない結果が生じる可能性があります。

関連タスク

169 ページの『IP 相互接続 (IPIC) の定義』
TCP/IP 接続を定義することにより、IP 相互接続 (IPIC) を使用してリモート CICS 領域と通信することができます。

IPIC を使用した相互通信機能

IP 相互通信 (IPIC) を使用することにより、TCP/IP ネットワークを使用している CICS システム間で通信することができます。

- IPIC では、以下の相互通信機能とリリースがサポートされています。
  - CICS TS 3.2 以降の領域との間の分散プログラム・リンク (DPL) 呼び出し
  - CICS TS および TXSeries バージョン 7.1 との間の分散プログラム・リンク (DPL) 呼び出し
  - 非同期に行われる非端末 EXEC CICS START、START CHANNEL、および CANCEL の各コマンドの機能シップ (CICS TS 4.1 領域間) に制限される処理

– 3270 端末の従来のトランザクション・ルーティング (この端末では、端末専有領域 (TOR) は APPLID によって CICS TS 4.1 領域間で固有に識別されます)
– CICS Transaction Gateway バージョン 7.1 以降からの ECI 要求

関連概念

4 ページの『相互通信機能』
複数システム環境において、各参加システムは、独自のローカル端末とデータベースを持つことができ、そのローカル・アプリケーション・プログラムを、ネットワーク内の他のシステムとは無関係に実行することができます。

関連データと発信元データ

関連データは、非システム・タスクが実行される環境、および領域内での非システム・タスクへの接続方法を示す一連の情報です。発信元データは、外部要求が CICSPlex に到着した際に開始される非システム・タスクによって作成されます。

関連データの特性

一部の関連データは、タスクに固有のものです。例えば、タスク ID、タスクに関連するユーザー ID、タスクの基本機能などがあります。他の関連データは、タスクの起点に関する情報を提供します。

CICS Explorer、WUI、INQUIRE ASSOCIATION、および INQUIRE ASSOCIATION LIST の各コマンドを使用して、関連データを表示することができます。INQUIRE ASSOCIATION LIST コマンドは、関連データ内に含まれる相関情報のうち、一致する情報を持つタスクのリストを、ローカル領域に返します。

関連データを使用して、TCP/IP 接続と CICS 領域およびその領域を使用するトランザクションを相互に関連付けることができます。関連データでは、タスクを開始する要求を受け取るソケットに、ソケット・アプリケーション・データ (ApplData) が使用されます。TCP/IP では、Netstat の ALL/-A、ALLConn/-a、および COnn/-c レポートで ApplData 情報を取得することができ、APPLD/-G フィルターで検索することができます。Netstat での ApplData の使用に関する追加情報については、「*System Administrator's Commands*」を参照してください。ApplData 情報は、SMF 119 TCP Connection Termination レコードで取得することができます。追加情報については、「*IP Configuration Reference*」を参照してください。ApplData 情報は、ネットワーク管理インターフェースを介して取得できます。詳しくは、「*IP Programmer's Guide and Reference*」を参照してください。

発信元データの特性

発信元記述子レコード (ODR) は、発信元データ情報を保持する関連データの一部です。発信元データは、関連データの別のセクションに格納され、タスクが開始された場所 (起点) を示します。

発信元データは、VTAM® 端末、ブラウザー、または他の外部装置からスケジュールに入れられるトランザクション ID の結果である可能性があります。CICS が接続するタスクは、新しい起点にあります。また、CICS によって、タスクの ODR のフィールドに、そのタスクに関連する情報が入力されます。タスクの下で実行中のアプリケーション・プログラムが原因で同じ領域の他のタスクに接続される場合、発信元データは新しいタスクから継承されます。IPIC 接続を介して新しいタスクに

リモートに接続される場合、そのリモート・タスクは同じ発信元データを継承します。発信元データは、APPC 接続または MRO 接続を介して伝搬されるわけではありません。また、いずれかの接続を介して接続されるタスクは、新しい起点にあるものと見なされます。

CICS Transaction Gateway を使用している場合、起点は CICS 以外の場所 (CICS TG) にすることができます。起点の情報は、タスクが CICSPlex の境界で開始される際に、ODR に入力されます。例えば、CICS TG は、JCA リソース・アダプターの起点に関するコンテキスト情報を記録します。また、この情報は、起点データの一部として CICS に渡されます。

関連データの起点データ・フィールドには、すべて 『OD』 から始まる名前が付けられています。ユーザー関係子データ・フィールドである USERCORRDATA を除くすべてフィールドには CICS によって入力されます。USERCORRDATA は 64 バイトの領域で、XAPADMGR グローバル・ユーザー出口を使用して入力することができます。出口は、CICSplex 内の起点で実行されているタスクからのみ呼び出すことができます。発信元データを使用することにより、IPIC 接続を使用して作業を共有する領域間において、分散トランザクションを追跡することができます。WUI を使用して、共通の発信元データのセットを共有する CICSplex でアクティブなすべてのタスクを検索したり、フィールドのサブセットで検索したりすることができます。

発信元データを使用してトランザクション・グループ ID (TRNGRPID) を提供し、複雑なシステムを追跡および監査することができます。TRNGRPID は、起点データを示す固有キーです。トランザクションが同じ作業単位を共有しておらず (START コマンドを使用する場合など)、トランザクションのどの部分に共通ソースが含まれているかが示されていない場合でも、TRNGRPID を使用して、トランザクションが作成された場所を追跡することができます。CICS は、情報のターゲット位置ではなく情報のソースを確認します。また、発信元データを使用して、独自の識別トークンを処理要求に付加することができます。

発信元データはモニター・レコードに書き込まれ、オフライン分析に備えて CICSPlex® SM ヒストリー・レコードに格納されます。発信元データはリカバリー不能な情報です。つまり、発信元データは、トランザクション再始動が原因で接続されたタスク、または領域を再始動したときにシステム・ログから再作成されるタスクでは表示されません。

CICS タスクとコンポーネント間での関連データおよび発信元データのフロー

18 ページの図 4 は、CICS において関連データと発信データがどのように収集されて、使用されるかを示しています。

- 新しいタスクを接続すると、関連データが作成されます。TCP/IP ネットワークを介して到達するメッセージに対する応答としてこのタスクが作成される場合、CICS がインターネット・プロトコル・スタックから入手した追加情報 **1** も格納されます。
- 新規タスクの発信元データは、関連データの別のセクションに格納され **2**、タスクが開始された場所 (起点) を示します。

- グローバル・ユーザー出口がこのタスクによって呼び出されると 3 、出口は XPI を使用して他のソースの情報を入手でき 4 、タスクに戻します。その情報は、発信元データに組み込まれます 5 。
- タスクがリモート領域に DPL 要求を発行すると、発信元データは、TCP/IP を介してリモート CICS 領域に送信される DPL 要求に追加されます。リモート領域に DPL 要求が到達すると、別の新規タスクがその要求の処理を開始します。CICS はこのタスクの固有関連データを作成しますが、発信元データを検出し、DPL 要求を処理するために接続される際にミラー・タスクにその発信元データを渡します 6 。
- タスク接続処理の際、発信元データは新規タスクの関連データの一部として格納され 7 、グローバル・ユーザー出口は呼び出されません。
- モニターが有効な場合、タスクのモニター・レコードに発信元データが書き込まれ 8 、CICSPlex SM が構成されている場合には、データがヒストリー・レコードに格納されます 9 。
- CICSPlex SM WUI を使用すると、実行中のタスクの関連データに格納されている情報を取り出すことができます 10 。例えば、発信元データが一致する CICSplex 内のタスクを検出する検索を作成できます。
- また CICSPlex SM を使用すると、ヒストリー・レコードに格納されている発信元データ情報のオフラインでの分析を実行できます 11 。例えば、TCP/IP ネットワークが分散トランザクションでどのように使用されているかを把握するために実行できます。

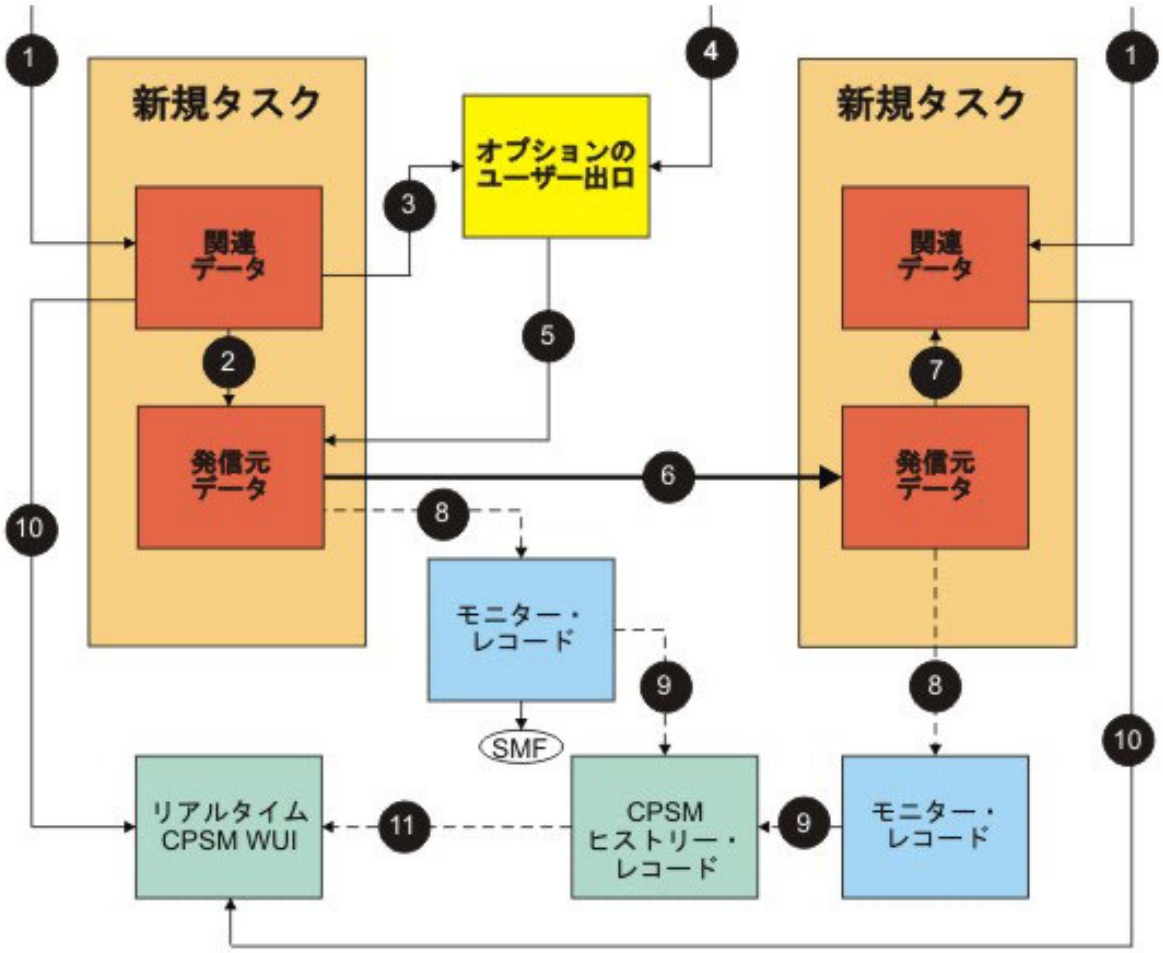


図 4. CICS 内での関連データおよび発信元データのフロー

発信元データの作成例

VTAM 端末の例と Web の例を見ることによって、発信元データがどのように格納され、他のタスクに渡されるかを理解することができます。

VTAM 端末の例

VTAM 端末でトランザクション ID を入力すると、タスクが領域で開始されます。発信元データが起点に格納され、同じ領域で初期タスクの結果として開始される他のタスクに渡されます。

1. タスクは CICSPlex の境界および起点にあります。CICS は、タスクに接続された際に、関連データの他のフィールドから発信元データ (VTAM 端末情報) を入力します。
2. タスクが、IPIC 接続を使用して他の領域で処理される DPL 要求を発行した場合、発信元データは DPL 要求を使用して渡されます。
3. メッセージを受信するリモート領域は、発信元データを抽出し、DPL 要求を処理する際に接続されるミラー・トランザクションにデータを渡します。

この例では、ミラー・トランザクションに関連データの次の情報が格納されます。

- ミラー・トランザクションについて説明する値。例えば、タスク ID、IPIC 接続の基本機能などです。
- DPL をスケジュールに入れた端末タスクが作成し、独自の関連データに格納した発信元データ。

この例では、関連データ出口である XAPADMGR は、端末タスクに接続された際に実行することができます。ただし、ミラー・タスクが初期化される際には、出口は呼び出されません。

Web の例

20 ページの図 5 は、TCP/IP ネットワークを介して渡され、CICS による処理のために到着した HTTP 要求を示しています。発信元データが起点に格納され、同じ領域で初期タスクの結果として開始される他のタスクに渡されます。この例では、発信元データは次の 2 つのタスクから入力されます。

1. HTTP 要求は、CSOL システム・タスクによって CICS に渡されます。
2. 要求は CWXN タスクによって処理されます。CWXN は起点にあります。CICS は、CWXN タスクに接続された際に、関連データの他のフィールドから発信元データ (HTTP 要求情報) を入力します。
3. 新しい CWBA タスクに接続されます。CWBA は、CWXN から ODR を継承します。あるいは、XAPADMGR グローバル・ユーザー出口は CWBA から呼び出され、この出口が発信元データを提供します。CWBA と CWXN は、別のユーザー ID で実行されることがあります。ただし、監査が目的である場合は、CWBA タスクによって使用されるユーザー ID (userid2) のほうが便利です。このため、CWBA タスクによって使用されるユーザー ID は CWBA の発信元データに格納されます。
4. CWBA タスクの制御下で実行されているアプリケーション・プログラムは、IPIC 接続を介して処理される DPL 要求を発行します。発信元データは、変更されないまま、DPL メッセージとともに CISR システム・タスクに渡されます。
5. DPL メッセージを受信するリモート領域は、発信元データを抽出し、ミラー・トランザクション (CSMI) に渡します。その後、ミラー・トランザクションに接続され、DPL 要求が提供されます。
6. ミラー・トランザクションで実行されるプログラムが START コマンドを発行します。発信元データは、START 要求を提供するために接続されるタスク (USER) に継承されます。

20 ページの図 5 には、CICS が HTTP 要求を処理する際に発信元データがどのように作成されるか、また発信元データが、要求を満たすために接続される他のタスクにどのように継承されるかが示されています。

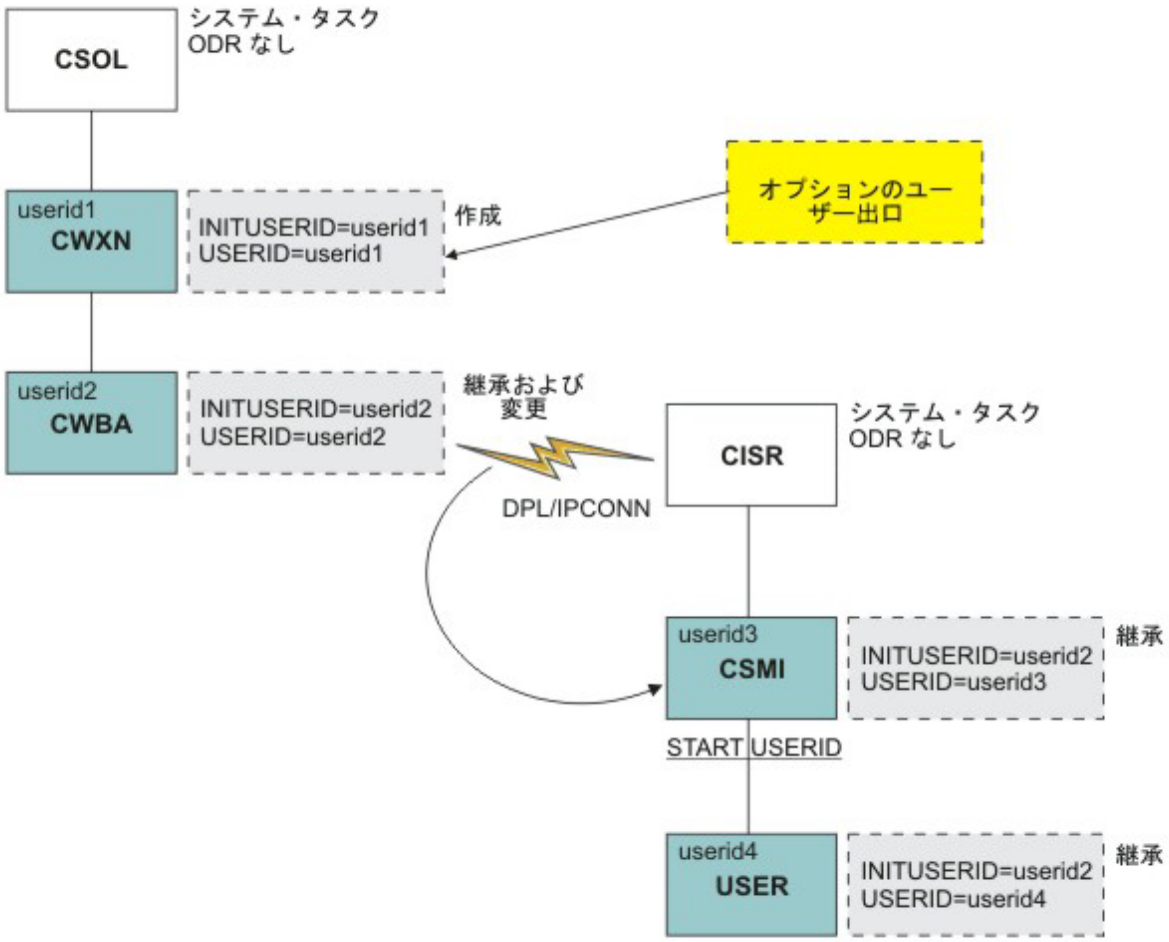


図 5. HTTP 要求が処理される際の発信元データの作成と移動

SNA を介したシステム間通信

CICS は、SNA を介したシステム間通信 (SNA を介した ISC) に相互接続機能を提供しています。SNA を介した ISC は、IBM システム・ネットワーク体系 (SNA) を実装しています。SNA は、複数システム環境でシステム間のデータ・フォーマットと通信プロトコルを定義しています。SNA は、CICS と、APPC または LUTYPE6.1 通信をサポートする他のシステムとの間で使用することができます。SNA は、すべての基本 CICS 相互通信機能をサポートしています。

以下のトピックに進む前に、SNA の全般的な概念と用語の知識が必要です。

この章には以下のトピックが含まれています。

- 21 ページの『サブシステム間の接続』
- 23 ページの『システム間セッション』
- 26 ページの『システム間セッションの確立』.

ISC を使用した相互通信機能

SNA を介したシステム間通信 (SNA を介した ISC) 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。

これらの機能は、ISC を使用した次のような相互通信で使用可能です。

- 機能シップ
- 非同期処理
- トランザクション・ルーティング
- 分散プログラム・リンク
- 分散トランザクション処理

ISC は、VTAM 拡張プログラム間通信 (APPC) または論理装置タイプ 6.1 (LUTYPE6.1) の通信をサポートする CICS およびその他のシステム間で使用することができます。例えば、SNA を介した ISC 接続が可能なのは、異なる z/OS シスプレックスまたは異なるオペレーティング・システム・プラットフォームで稼働する CICS 領域間、CICS とすべての APPC デバイス間、および CICS と IMS 間です。

CICS Transaction Server for z/OS は SNA を介した ISC を使用して以下のシステムと通信することができます。

- その他の CICS Transaction Server for z/OS システム
- CICS Transaction Server for VSE
- CICS Transaction Server for iSeries®
- IMS バージョン 9.1 以降
- 拡張プログラム間通信 (APPC) プロトコル (LU6.2) をサポートするすべてのシステム

サブシステム間の接続

システム間連絡でサブシステムに接続するには、次の 3 つの基本形式を使用します。

- 単一ホスト・オペレーティング・システム内の ISC
- 物理的に隣接するオペレーティング・システム間の ISC
- 物理的にリモートにあるオペレーティング・システム間の ISC

22 ページの図 6 に可能な構成を示します。

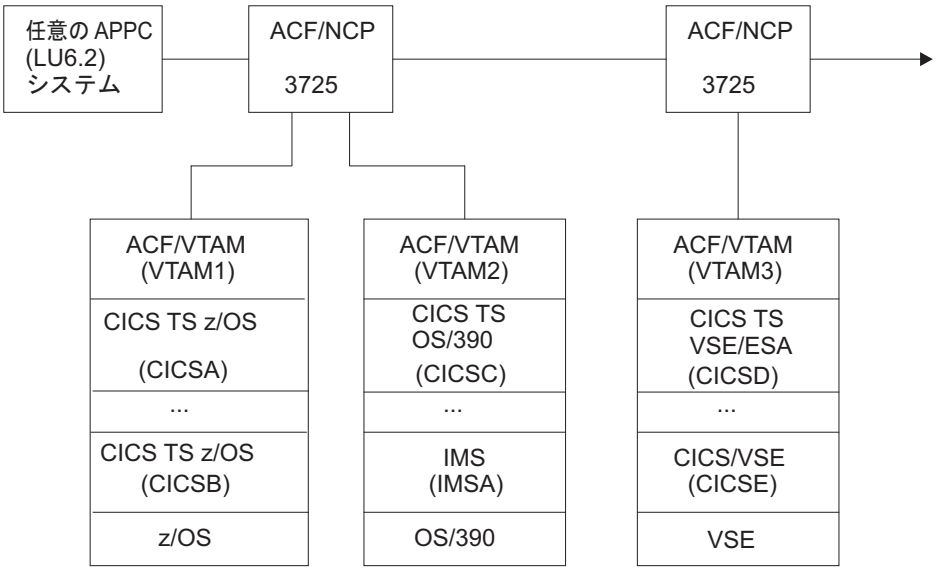


図 6. 相互通信するシステムの可能な構成

単一のオペレーティング・システム

単一オペレーティング・システム内の ISC (ホスト内 ISC) は、ACF/VTAM のアプリケーション間機能を使用することで実行可能になります。図 6 では、これらの機能を使用すれば、CICSA と CICSB の間、CICSC と IMSA の間、および CICSD と CICSE の間で通信を行うことができます。

MVS システムの場合、ホスト内 ISC は、2 つ以上の CICS システム間での通信 (ただし、MRO の方が効率的です) や、CICS システムと IMS システム間などでの通信に使用することができます。

CICS から見ると、ホスト内 ISC は、異なる VTAM ドメインにあるシステム間での ISC と変わりません。

物理的に隣接するオペレーティング・システム

IBM 3725 を多重チャネル・アダプターの構成にすると、単一の ACF/NCP/VS によって 2 つの VTAM ドメイン (例えば、図 6 の VTAM1 と VTAM2) を接続することができます。この構成は、以下の通信の際に便利です。

- 実動システムとローカルの別個のテスト・システム間の通信
- 異なる特性または要件を持つ 2 つの実動システム間の通信

ACF/VTAM がインストールされた複数のシステム間では、直接チャネル間通信が可能です。

リモート・オペレーティング・システム

システム間連絡で最も一般的な構成は、リモート・オペレーティング・システム間の接続です。例えば、図 6 では、CICSD と CICSE を、CICSA、CICSB、および CICSC に対してこの方法で接続することができます。各参加システムは、MVS か拡張仮想記憶域 (VSE) の CICS または IMS と、ACF/VTAM などの ACF アクセス方式の 1 つを使用して、その特定の場所に応じて適宜構成されます。

CICS Transaction Server for z/OS が ISC を使って接続できる CICS および 非 CICS のシステムのリストについては、3 ページの『システム間の通信』を参照してください。ISC を使用して CICS Transaction Server for z/OS を他の CICS 製品に接続する方法についての詳細は、 「*CICS Family: Communicating from CICS on zSeries*」を参照してください。

システム間セッション

CICS は、ACF/VTAM を使用して、リモート・システムとの論理装置間 (LU-LU) セッションを確立 (またはバインド) します。これは論理的な接続ですので、LU-LU セッションは、2 つのシステム間における物理的な経路からは独立しています。単一の論理接続によって、複数の独立したセッションが可能です。このようなセッションを、*並列*セッションと呼びます。

CICS は、2 つのタイプのセッションをサポートしていますが、いずれも IBM システム・ネットワーク体系によって定義されています。

- LUTYPE6.1 セッション
- LUTYPE6.2 通常呼び出される APPC セッション

LUTYPE6 セッションの特性については、システム・ネットワーク体系の資料「*Sessions Between Logical Units*」を参照してください。

1 つの LU-LU ペア間に、同時に複数の APPC 接続をインストールすることはできません。LU-LU ペアの間で APPC 接続と LUTYPE6.1 接続を同時にインストールすることはできません。

LUTYPE6.1

LUTYPE6.1 は、LUTYPE6.2 (APPC) の前にあった機能です。

LUTYPE6.1 セッションは CICS と IMS の両方でサポートされますので、これを CICS-IMS 間通信で使用することができます。(CICS-CICS 間通信には、LUTYPE6.2 が望ましいプロトコルです。)

LUTYPE6.2 (APPC)

LUTYPE6.2 プロトコルは一般に、拡張プログラム間通信 (APPC) と呼んでいます。APPC 体系は、トランザクション処理システム間のデータ通信を可能にするだけでなく、デバイス・レベルの製品 (APPC 端末) がホスト・レベルの製品と通信したり、相互に通信したりできるようにするサブセットも定義しています。したがって、APPC セッションは、CICS-CICS 間通信、および CICS と他の APPC システムや端末との通信にも使用することができます。

次に、APPC 体系の基本的な特性の概要をいくつか示します。

プロトコル境界

APPC プロトコル境界は、トランザクションと SNA ネットワーク間の汎用インターフェースです。これは、動詞 (*verb*) と呼ばれる形式設定された関数と、verb を使用するためのプロトコルによって定義されます。この SNA プロトコル境界の詳細については、システム・ネットワーク体系の「*Transaction Programmer's Reference Manual for LU Type 6.2*」を参照してください。

CICS には、プロトコル境界にマップすることによって、APPC 会話をもつアプリケーション・プログラムをユーザーが作成できるようにするコマンド・レベル言語があります。あるいは、この代わりに Systems Application Architecture® (システム・アプリケーション体系、SAA) 環境の*共通プログラミング・インターフェース・コミュニケーション (CPI コミュニケーション)* を使用することもできます。

次の 2 つのタイプの APPC 会話が定義されています。

マップ式
マップ式会話では、APPC アプリケーション・プログラム・インターフェースとの間でやりとりされるデータはユーザー・データです。ユーザーは、この体系で必要とされる内部データ形式については関知しません。

基本 基本会話では、APPC アプリケーション・プログラム・インターフェースとの間でやりとりされるデータに、GDS ヘッダーというヘッダーが接頭部として付けられます。ユーザーは、このヘッダーを作成し解釈する必要があります。基本会話は、主に、マップ式会話をサポートしておらず、ユーザーにオープンされたアプリケーション・プログラミング・インターフェースを持たないデバイス・レベルの製品との通信に使用されます。

同期レベル

APPC 体系には、3 つのレベルの同期があります。CICS では、これらのレベルは、レベル 0、1、および 2 として知られています。SNA の用語では、これらのレベルは次のように、NONE、CONFIRM、および SYNCPOINT に対応します。

レベル 0 (NONE)
このレベルは、同期点をサポートしないシステムまたはデバイスと通信する場合、または同期が不要な場合に使用されます。

レベル 1 (CONFIRM)
このレベルを使用すると、会話トランザクションで、プライベートな同期要求を交換することができます。CICS 組み込み同期は、このレベルでは起こりません。

レベル 2 (SYNCPOINT)
このレベルは、ロールバックを含む完全な CICS 同期点処理に相当します。レベル 1 同期要求を使用することもできます。

EXEC CICS コマンドおよび CPI コミュニケーションは、3 つのレベルすべてをサポートしています。

プログラム初期設定パラメーターのデータ

トランザクションは、APPC セッションによって接続されたリモート・トランザクションを開始する際に、その接続されたトランザクションが受信するデータを送信することができます。このデータは、プログラム初期設定パラメーター (PIP) と呼ばれ、SNA 体系の規則に従って、1 つまたは複数の可変長サブフィールドに形式設定されます。CPI コミュニケーションは PIP をサポートしません。

LU サービス・マネージャー

マルチセッション APPC 接続では、LU サービス管理を使用します。これは、セッション・バインドの交渉、セッションの活動化と非活動化、再同期、およびエラー

処理を行うソフトウェア・コンポーネントです。これは、リモート LU との 2 つの特殊セッションを必要とします。これらのセッションのことを *SNASVCMG* セッションと言います。これらのセッションがバインドされると、この接続がユーザーの「割り振りに使用できない」場合でも、LU-LU 接続の両側が相互に通信することができます。

単一セッションの APPC 接続には、SNASVCMG セッションがありません。このため、その機能は制限されます。例えば、この接続はレベル 2 の同期をサポートしません。

サービス・クラス

CICS の APPC 実装には、「サービス・クラス」の選択についてのサポートが組み込まれています。

サービス・クラス (COS) は、ACF/VTAM 機能の 1 つで、論理装置ペア間のセッションに異なる特性を持たせることができます。

- 代替ルーティング - 指定 COS の仮想経路を、異なる物理パス (明示経路) に割り当てることができます。
- 各種トラフィック - 異なる種類のトラフィックを同じ仮想経路に割り当てることができます。また、適切な伝送優先順位を選択することによって、セッションへの不当な干渉を防止することができます。
- トランキング機能 - 明示経路は特定ノード間の並列リンクを使用することができます。

特に、セッションは、異なる仮想経路をとることができます。このため、異なる物理リンクを使用できます。また、セッションの優先順位は、それらが伝送するトラフィックに合わせて高くすることも低くすることもできます。

CICS では、APPC セッションは *モードセット* というグループで指定され、それぞれに *モード名* が割り当てられます。このモード名は、VTAM LOGMODE 項目 (*モードグループ* とも呼ばれます) の名前でなければなりません。これによって、そのセッション・グループに必要なサービス・クラスを指定することができます。詳細については、CICS の ACF/VTAM LOGMODE テーブル項目を参照してください。

限定リソース

ネットワーク・リソースのいくつか (例えば、交換回線など) を効率的に使用するために、SNA では、このようなリソースをネットワークにおいて*限定リソース*として定義することができます。セッションがバインドされると、VTAM は CICS に対して、バインドが限定リソースを介して行われたかどうかを示します。限定リソースを介したセッションを使用するタスクがそのセッションを解放した場合、他のタスクが必要としていなければ、CICS はそのセッションをアンバインドします。

単一セッション接続でもマルチセッション接続でも、限定リソースを使用することができます。複数セッション接続の場合、CICS は、その接続におけるすべてのモードグループが最初の『セッション数変更』(CNOS) の交換を完了するまで、LU サービス管理セッションをアンバインドしません。CICS が、あるセッションをアンバインドすると、CICS は競合勝者と敗者のバランスをとろうとします。このバランシン

グにより、CICS は、アンバインドされたセッションが勝者にも敗者にもならないようにリセットする場合があります。

システム間セッションの確立

通信がシステム間セッションで行われるには、セッションが確立されているか、またはバインドされていなければなりません。

CICS は、システム間セッションにおいて 1 次 (BIND 送信側) になることも 2 次 (BIND 受信側) になることもあり、 競合勝者になる場合も競合敗者になる場合もあります。LU-LU セッションにおける競合勝者とは、 いつでも会話を開始できる許可を与えられた LU のことを言います。競合敗者とは、会話の開始許可を要求するために、SNA BID コマンド (LUTYPE6.1) または LUSTATUS コマンド (APPC) を使用しなければならない LU のことをいいます。

特定のリモート・システムへのリンクに必要な競合勝者セッションと競合敗者セッションの数を指定することができます。

LUTYPE6.1 セッションでは、CICS は常に競合敗者としてバインドします。

APPC リンクの場合、競合勝者セッションの数は、リンクの定義時に指定されます。 186 ページの『APPC 接続の定義』を参照してください。競合勝者セッションは、通常 CICS によってバインドされますが、CICS は、これらのセッションに対するリモート・システムからのバインド要求も受け入れます。

一般に、競合敗者セッションは、リモート・システムによってバインドされます。ただし CICS は、リモート・システムがバインド要求を送信できない場合に、競合敗者セッションをバインドすることもできます。

APPC 端末に対する単一セッションは通常競合勝者として定義され、CICS によってバインドされます。しかし CICS は、競合勝者が敗者に変更される折衝バインドを受け入れることもできます。

セッションは、次のいずれかの方法で開始することができます。

- AUTOCONNECT (YES) か AUTOCONNECT (ALL) が指定されたセッションの CICS 初期設定中に CICS によって。 165 ページの『第 13 章 リモート・システムへの接続の定義方法』を参照してください。
- CICS マスター端末オペレーターからの要求によって。
- CICS の通信相手であるリモート・システムによって。
- アプリケーションがシステム間セッションの使用を明示的に、または暗黙指定で要求し、その要求を、以前にアンバインドされたセッションをバインドすることによってしか満たすことができない場合、CICS によって。

第 3 章 複数領域操作

CICS 複数領域操作 (MRO) を使用すると、同じ MVS イメージまたは同じ MVS 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。

この章には以下のトピックが含まれています。

- 『MRO を使用した相互通信機能』
- 28 ページの『システム間複数領域操作 (XCF/MRO)』
- 32 ページの『複数領域操作の応用』
- 35 ページの『単一領域システムからの変換』.

MRO を使用した相互通信機能

複数領域操作 (MRO) を使用すると、同じ MVS イメージまたは同じ MVS 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 は以下のような相互通信機能を提供しています。

- 機能シップ
- 非同期処理
- トランザクション・ルーティング
- 分散プログラム・リンク
- 分散トランザクション処理

MRO には、分散トランザクション処理について一定の制限があります。外部 CICS インターフェース (EXCI) では、次の種類の通信をサポートするために、特殊形式の MRO リンクが使用されます。

- MVS バッチ・プログラムと CICS との間の通信
- CICS プログラムへの DCE リモート手続き呼び出し

MRO には、ACF/VTAM および SNA ネットワーク機能は必要ありません。領域間通信のための CICS サポートは、*領域間通信* (*IRC*) と呼ばれます。IRC は、次の 3 つの方法で実装できます。

- CICS 端末管理モジュールによるサポートを通して、および、MVS リンク・パック域 (LPA) にロードされた CICS 提供の領域間プログラム (DFHIRP) の使用によって。DFHIRP は、タイプ 3 の監視プログラム呼び出し (SVC) によって開始されます。便宜上、この複数領域操作の実装は MRO(IRC) と呼ばれます。これは、この実装の選択時には、CONNECTION 定義で ACCESSMETHOD(IRC) を指定するためです。
- MVS 仮想記憶間 (XM) サービスによって。CICS のタイプ 3 SVC メカニズムの代わりに、この方法を選択することもできます。この場合、DFHIRP が領域間リンクのオープンとクローズのためだけに使用されます。

© Copyright IBM Corp. 1977, 2009

- IBM MVS/ESA™ のシステム間カップリング・ファシリティー (XCF) によって。MVS シスプレックスの異なる MVS イメージにある CICS 領域と CICS 領域の間での MRO リンクには XCF が必要です。このようなリンクの場合には、XCF が使用可能であれば、それが CICS によって動的に選択されます。

MRO によってリンクされる CICS 領域は、異なるリリース・レベルであっても構いません。MVS イメージが異なるリリースの CICS を含んでおり、そのすべてが互いに通信する際に MRO を使用し、シスプレックス内の他のイメージの領域と通信する際に SCF/MRO を使用する場合、MVS LPA 内の DFHIRP モジュールは、そのイメージ内の最新の CICS リリースまたはそれよりも新しいリリースからのものである必要があります。

システム間複数領域操作 (XCF/MRO)

システム間カップリング・ファシリティー (XCF) は MVS/ESA 基本制御プログラムの一部です。これにより、シスプレックス (*sys*tems com*plex*) において、チャネル間リンク、チャネル、またはカップリング・ファシリティー・リンクによってリンクされた MVS イメージ間でハイパフォーマンスの通信リンクが可能になります。

IRC によって提供される XCF アクセス方式を使用することにより、同じ MVS シスプレックスにある MVS イメージの間で通信する場合に、VTAM が不要になります。

各 CICS 領域は、他の MVS 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。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します。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。

異なる MVS 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指定変更して、この XCF アクセス方式を動的に選択します。MVS システム間カップリング・ファシリティーを使用すると、MRO は、シスプレックス環境の複数の MVS イメージの*間で* 機能するようになります。このため、通常の MRO 操作がすべてサポートされます。

XCF/MRO は、MVS イメージ間の共用データ・テーブルへのアクセスをサポートしません。複数の CICS 領域にわたって、1 つのデータ・テーブルに共用アクセスするには、それらの領域が同じ MVS イメージになければなりません。異なる MVS イメージにあるデータ・テーブルにアクセスするには、機能シップを使用します。

各 CICS 領域は、IRC にログオンした時点で参加する 1 つの XCF グループだけのメンバーになることができます。XCF グループの最大サイズは MVS MAXMEMBER パラメーターにより制限され、メンバーは最大 2047 に制限されています。この制限が問題となる場合、例えばシスプレックスに含めることができる CICS 領域の数を制限する場合は、それぞれが異なる一連の領域を含む複数の XCF グループを作成することができます。例えば、稼働領域に XCF グループを 1 つ、また開発領域とテスト領域に別のグループをそれぞれ含めることができます。複数の XCF グループを含める場合は、以下の推奨事項に従ってください。

- 稼働領域を、開発領域とテスト領域とは異なる XCF グループに置く

- 必要以上の XCF グループを作成しない。説明とは異なりますが、2 つあれば十分な場合があります。
- XCF グループ間の領域を移動しないようにする
- 既存の XCF グループに領域を追加したり、既存の XCF グループから領域を除去しないようにする

CICS 領域は MRO または XCF/MRO を使用して、同じ XCF グループ内の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 MVS イメージ内にある場合でも、MRO または XCF/MRO を使用して通信することはできません。

XCF/MRO によってリンクされる CICS 領域は、異なるリリース・レベルであっても構いません。これについては、4 ページの『複数領域操作』を参照してください。XCF/MRO に参加する MVS イメージにインストールされている CICS のバージョンによって、MVS イメージのリンク・パック域にインストールされる DFHIRP のバージョンは異なって構いません。単一の MVS イメージが異なるリリースの CICS を含んでおり、そのすべてがシスプレックス内の他のイメージの領域と通信するために XCF/MRO を使用する場合、MVS LPA 内の DFHIRP モジュールは、そのイメージ内の最新の CICS リリースまたはそれよりも新しいリリースからのものである必要があります。ただし、CICS TS for z/OS バージョン 4.1 バージョンの DFHIRP (複数の XCF グループへのサポートが必要) を z/OS バージョン 1.7 またはそれ以降のバージョンでのみ使用することができます。XCF/MRO のソフトウェアとハードウェアの要件については、XCF/MRO のインストール要件を参照してください。

30 ページの図 7 は、シスプレックス環境における XCF/MRO の使用例です。この例では、CICS XCF グループ DFHIR000 が 1 つだけあります。DFHIR000 のメンバーは 2 つの MVS イメージ間を XCF/MRO リンク経由で通信することができます。

CICS1 と CICS2 間、および CICS3 と CICS4 間の MRO リンクには、そのリンクの定義に従って IRC か XM のアクセス方式が使用されます。MVS1 の CICS 領域と MVS2 の CICS 領域の間の MRO リンクには XCF 方式が使用されますが、これは CICS によって動的に選択されます。

各 MVS では、LPA の DFHIRP モジュールは、イメージ内の最高リリース・レベルの CICS TS for z/OS になっている必要があります。

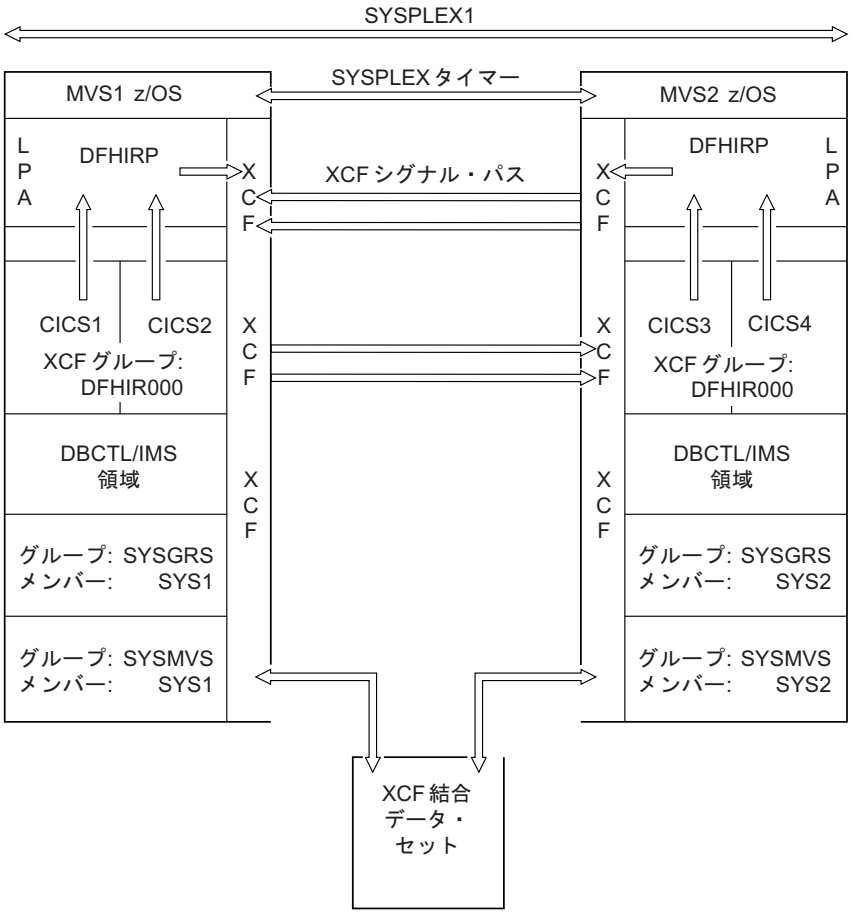


図 7. 単一の CICS XCF グループを含むシスプレックス (SYSPLEX1):

31 ページの図 8 では、もう少し複雑な例を示しています。この例には、2 つの CICS XCF グループ、DFHIR000 と DFHIR001 があります。各 XCF グループのメンバーは、XCF/MRO リンクにより MVS イメージ間で通信することがきます。

複数の CICS XCF グループに対応するには、両方の MVS イメージが z/OS バージョン 1.7 またはそれ以降のバージョンであり、CICS TS for z/OS バージョン 3.2 またはそれ以降のバージョンの DFHIRP を使用しなければなりません。z/OS では、バージョン 1.6 以降、複数の XCF グループをサポートしていますが、CICS TS for z/OS バージョン 3.2 (DFHIR000 以外の XCF グループの結合に必要) には z/OS バージョン 1.7 またはそれ以降のバージョンが必要です。

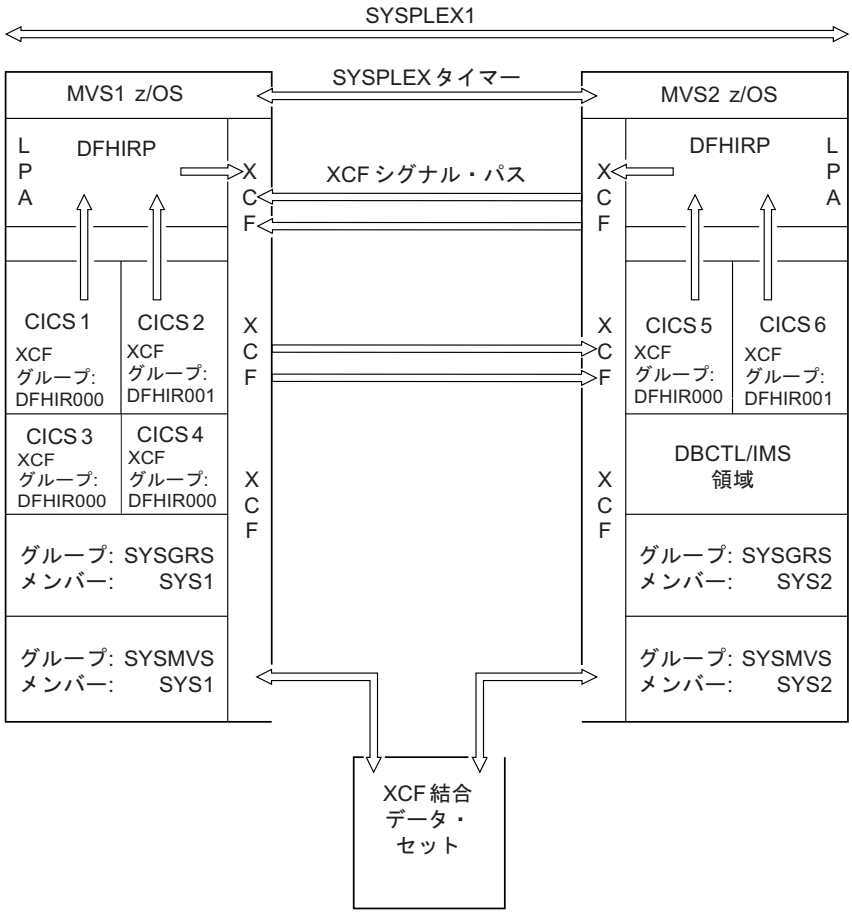


図 8. 2 つの CICS XCF グループを含むシスプレックス (SYSPLEX1)

注:

- MVS1 (CICS 1、CICS 3、CICS 4) 内の DFHIR000 XCF グループのメンバーは、MVS2 (CICS 5) 内の DFHIR000 XCF グループのメンバーと通信できるように CICS により動的に選択された XCF/MRO を使用します。同様に、MVS1 内の CICS 2 は XCF/MRO を使用して MVS 2 内の CICS 6 と通信します。これら 2 つの CICS はどちらも DFHIR001 グループのメンバーです。
- CICS 1、CICS 3、CICS 4 は、CICS 6 が別の XCF グループにあるため、CICS 6 との通信に XCF/MRO を使用することができません。同様に、CICS 2 は CICS 5 との通信に XCF/MRO を使用することができません。
- これらの CICS は同じ MVS イメージ内および同じ XCF グループ内にあるため、CICS 1、CICS 3、CICS 4 は、リンクに定義された MRO(IRC) または MRO(XM) のいずれかのアクセス方式により相互に通信することができます。
- CICS 5 は、CICS が同じ MVS イメージ内にあっても別の XCF グループに属しているため、CICS 6 との通信にどの形式の MRO も使用することができません。同様に、CICS 2 は、CICS 1、CICS 3、CICS 4 との通信にどの形式の MRO も使用することができません。

XCF/MRO の利点

XCF リンクを使用したシステム間 MRO には多くの利点があります。

- MVS イメージと MVS イメージの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 MVS システム間で通信するよりもパフォーマンスがかなりよい。したがって、XCF/MRO を使用すると、同じシスプレックスにおけるトランザクション・ルーティング、機能シップ、非同期処理、分散プログラム・リンクの効率がよくなります。さらに、LUTYPE6.1 プロトコルが該当する場合は、分散トランザクション処理にも XCF/MRO を使用することができます。
- 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 MVS や CICS のワークロードを代替のプロセッサーとシステムによって継続できるようにすることで、使用可能度が高まる。
- MVS イメージ同士の間で CICS システム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 MVS から別の MVS 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 MVS 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)。
- HPCS 環境において、低コストで、ラック・マウント方式の、かつ空冷式のプロセッサーを結合することによる価格とパフォーマンスの改善。
- システムを少しずつ拡張することができる。
- 組織としての利点。異なる XCF グループにある領域は MRO または XCF/MRO を介して通信することができないため、その領域の各グループは事実上、他のものから分離されます。この分離は、例えば、開発領域またはテスト領域から稼働領域への、偶然起こり得るアクセスを避けたい場合などに便利です。

複数領域操作の応用

複数領域操作には、標準的な複数の応用に対応できる環境が用意されています。

プログラム開発

新規作成プログラムのテストを実動作業から分離するには、別の CICS 領域をテスト用に実行します。このように分離することにより、テスト・システムが異常終了した場合でも実動システムは稼働しているため、新しいアプリケーションの開発中も、実動システムの信頼性と可用性を維持することができます。

テスト・システムは、実動作業を中断することなく、必要に応じて開始したり停止したりすることができます。実動システムに新しいプログラムをカットオーバーする間、端末オペレーターは、正規の実動端末から、テスト・システム内のトランザクションを実行することができます。また、新しいプログラムは、実動システムのリソースすべてにアクセスすることができます。

タイム・シェアリング

ある CICS システムが、正規の DB/DC 作業だけでなく、APL や ICCF などの計算作業を実行する場合、DB/DC ユーザーの応答時間が非常に長くなる可能性があります。優先順位の低いアドレス・スペースで計算用アプリケーションを実行し、DB/DC アプリケーションを別のアドレス・スペースで実行することによって、応答時間を短縮することができます。

トランザクション・ルーティングを使用すると、オペレーターに 2 つの異なるシステムの存在を意識させることなく、端末からどちらの CICS システムにもアクセスすることができます。

信頼できるデータベース・アクセス

CICS のストレージ保護とトランザクション分離を使用して、信頼性の低いアプリケーションのためにシステムが停止したり、他のアプリケーションが使用不能になることを避けることができます。

しかし、MRO を使用すれば、この保護レベルが強化されます。

例えば、CICS 領域を 2 つ定義し、一方で信頼性が低いと考えられるアプリケーションを、他方で信頼性の高いアプリケーションとそのデータベースを所有するとします。データベース所有領域で小数のアプリケーションを実行することにより、信頼性が高い領域で作業することができます。ただし、領域間トラフィックは増加するため、パフォーマンスが低下する可能性があります。したがって、パフォーマンスと信頼性のバランスをとる必要があります。

MRO のこの応用方法を最大限に活用した場合は、データベース所有領域にはユーザー・アプリケーションを置かない形になります。オンラインの性能低下は、非常に大規模なデータベースを所有する CICS 領域を再始動するために必要な経過時間を考えれば、許容できるかもしれません。

部門の分離

MRO を使用すれば、*CICSplex* を作成することによって、そこに、組織のさまざまな部門ごとに独自の CICS システムをもつことができます。

各部門は、必要に応じて各自のシステムを開始し、終了することができます。さらに、各部門は他の部門のデータにアクセスすることができます。このアクセスは、システム・プログラマーによって管理されます。1 つの部門は、別の部門のシステムでトランザクションを実行することができます。これも、システム・プログラマーの管理下におかれます。トランザクション・ルーティングを使用した場合、どの端末からでも任意のシステムでトランザクションを実行できるため、端末を部門に割り振る必要はありません。

マルチプロセッサーのパフォーマンス

MRO を使用すれば、いくつかの CICS システムを 1 つの CICSplex にリンクし、それらのシステムのトランザクションやデータ・リソースにどの端末からでもアクセスできるようにすることによって、マルチプロセッサーの利点を生かすことができます。

システム・プログラマーは、トランザクションとデータ・リソースを任意の接続システムに割り当てることによって、パフォーマンスを最適化することができます。トランザクション・ルーティングでは、端末ユーザーに対して、単一システム・イメージが示されます。したがって、ユーザーは、複数の CICS システムが存在することを意識する必要がありません。

トランザクション・ルーティングについては、 73 ページの『第 7 章 CICS トランザクション・ルーティング』を参照してください。

シスプレックスにおけるワークロード・バランシング

シスプレックスでは、MRO リンクと XCF/MRO リンクを使用して、機能的に同等の端末専有領域 (TOR) とアプリケーション所有領域 (AOR) のセットから成る CICSplex を作成することができます。

これらの製品と機能を使用して、ワークロード・バランシングを実行することができます。

- VTAM 総称リソース機能
- 動的トランザクション・ルーティング
- DPL 要求の動的ルーティング
- CICSPlex System Manager (CICSPlex SM)
- MVS ワークロード・マネージャー

CICS などの VTAM アプリケーション・プログラムが VTAM で認識される名前は、その VTAM APPL 定義ステートメントに定義された特定のネットワーク名はもちろん、総称リソース名でも構いません。いくつかの CICS 領域で同じ総称リソース名を使用することができます。

いくつかの端末専有領域がある CICSplex とのセッションを開始する場合、端末ユーザーは、ログオン要求で総称リソース名を使用します。総称リソース名を使用することにより、VTAM は、CICS TOR の 1 つをそのセッションのターゲットとして選択することができます。この方法が機能するためには、TOR がすべて同じ総称リソース名のもとに VTAM に登録されていなければなりません。VTAM は、使用可能なすべての端末専有領域にわたって端末セッションのワークロード・バランシングを実行することができます。

次に端末専有領域は、動的トランザクション・ルーティングを使用して、ワークロード・バランシングを実行することができます。アプリケーション所有領域は、DPL 要求を動的にルーティングすることができます。CICSPlex SM 製品を使用することにより、CICSplex における動的ルーティングを管理することができます。

VTAM 総称リソースについての詳細は、「*VTAM バージョン 4 リリース 2 リリース・ガイド*」を参照してください。

- 109 ページの『DPL 要求の動的ルーティング』
- 75 ページの『動的トランザクション・ルーティング』
- *CICSPlex System Manager Managing Workloads.*
- *CICS パフォーマンス・ガイド*

仮想記憶域制約解放

大規模な CICS システムの中には、使用可能な仮想記憶域の量が制約要因になるものがあります。

このような場合、システムを、共用リソースを持つ複数の独立したシステムに分割することによって、仮想記憶域の問題を解決できることがあります。MRO のすべての機能を使用して、ユーザーの単一システムのイメージを維持することができます。

DL/I データベースを使用している場合、システムを分割して仮想記憶域の制約を避けるためには、CICS 機能シップではなく DBCTL を使用して、CICS のアドレス・スペースの間でそのデータベースを共用することを検討してください。

単一領域システムからの変換

通常は、既存の単一領域の CICS システムを、少々のプログラミングで、またはプログラミングを行うことなく、複数領域の CICS システムに変換することができます。

CICS 機能シップを使用すると、既存のコマンド・レベル・アプリケーションが所有する端末のオペレーターは、アプリケーションまたはリソースのいずれかが別の CICS 領域に転送された後でも、既存のデータ・リソースに引き続きアクセスすることができます。機能シップを使用するアプリケーションは、267 ページの『第 19 章 CICS 機能シップのアプリケーション・プログラミング』に示す規則に従っていなければなりません。これらの規則に準拠するために、単一領域 CICS システム用に作成されたプログラムを修正する必要がある場合があります。

CICS トランザクション・ルーティングを使用すると、ある CICS 領域に所有されている端末のオペレーターは、接続されている CICS 領域でトランザクションを実行することができます。この機能の使用法の 1 つは、CICS の現行リリースではサポートされなくなった機能をアプリケーションから引き続き使用できるようにすることです。このような共存に関する考慮事項については、*CICS Transaction Server for z/OS V3.2 からのアップグレード*を参照してください。さらに、この場合の制約事項については、279 ページの『第 22 章 CICS トランザクション・ルーティングのアプリケーション・プログラミング』で説明します。

2 つの領域間の MRO リンクを定義して、共用リソースのローカル定義およびリモート定義を行う必要があります。

第 4 章 CICS 機能シップ

CICS 機能シップを使用して、CICS アプリケーション・プログラムを、要求されたリソースの位置を考慮することなく作成することができます。これらのアプリケーション・プログラムでは、ファイル制御コマンドや一時記憶コマンドなどの機能がまったく同じ方法で使用されます。

この章には以下のトピックが含まれています。

- 『機能シップの概要』
- 38 ページの『機能シップの設計上の考慮事項』
- 42 ページの『ミラー・トランザクションと変換プログラム』
- 46 ページの『機能シップの例』.

機能シップの概要

CICS 機能シップを使用して、CICS アプリケーション・プログラムで次のタスクを実行することができます。

- ファイル制御要求をシップすることによって、他の CICS システムが所有している CICS ファイルにアクセスする。
- DL/I 機能に対する要求をシップすることによって、他の CICS システムが管理するか、アクセスできる DL/I データベースにアクセスする。
- 一時データや一時記憶域の機能に対する要求をシップすることによって、他の CICS システム内にある一時データや一時記憶域のキューにデータを転送する、またはそこからデータを転送する。
- インターバル制御機能の START 要求をシップすることによって、他の CICS システムや、SNA LU タイプ 6 プロトコルを実装する他の非 CICS システム (IMS など) でトランザクションを開始する。この形式の通信については、 51 ページの『第 5 章 非同期処理』で説明します。

アプリケーションは、要求されたリソースの位置に関係なく作成することができます。これらのアプリケーションでは、ファイル制御コマンドや一時記憶コマンドなどの機能がまったく同じ方法で使用されます。CICS リソース定義テーブル中の項目によって、システム・プログラマーは、指定のリソースがローカル (つまり要求側の) システムではなく、リモート (つまり所有側の) システムにあるものとして指定することができます。

シップされたファイル制御要求を、 38 ページの図 9 に示します。この図では、CICA で実行されているトランザクションが、NAMES というファイルに対しファイル制御コマンドの READ を出します。ファイル管理テーブルには、このファイルが CICB というリモート CICS システムによって所有されていることが示されています。CICS は、READ 要求を適切な伝送形式に変更してから、それを CICB にシップして実行します。

© Copyright IBM Corp. 1977, 2009

CICB では、その要求は、ミラー・トランザクションという特殊なトランザクションに渡されます。ミラー・トランザクションは、元の要求を再作成し、CICB でそれを出し、入手したデータを CICA に戻します。

CICS のリカバリーと再始動によって、リモート・システムのリソースの更新が可能になります。さらに、要求側のアプリケーション・プログラムが同期点に達すると、保護リソースを更新しているミラー・トランザクションも同期点をとります。こうすることによって、リモート・システムとローカル・システムの保護リソースに加えられた変更の一貫性が維持されます。CICS マスター端末オペレーターは、この処理で障害が発生すると通知を受け取るため、適切な修正処置をとることができます。この処置は、手作業でも、ユーザー作成コードでも実行することができます。

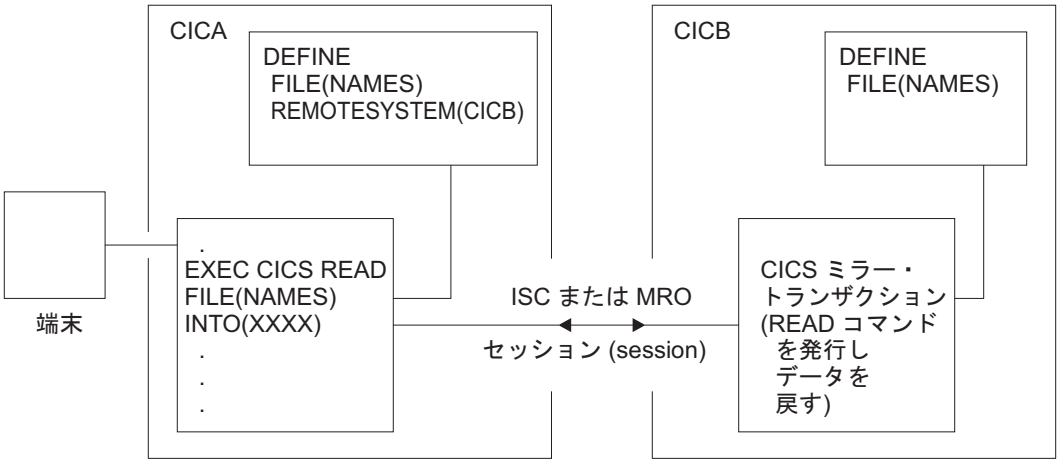


図 9. 機能シップ

機能シップの設計上の考慮事項

ユーザー・アプリケーション・プログラムは、CICS 相互通信環境で実行でき、アクセスするファイルまたはその他のリソースの位置を意識することなく、相互通信機能を使用することができます。リソースの位置は、リソース定義に指定されています。

リモート・リソースを識別および定義する方法については、 223 ページの『第 16 章 リモート・リソースの定義』を参照してください。

リソース定義には、リモート・システムで認識されているリソースの名前が、ローカルで認識されている名前と異なる場合、それを指定することもできます。リソースがそのローカル名で要求されると、CICS は、リモート名に置き換えてからその要求を送信します。リモート名を置き換える機能は、複数のシステムに同じ名前のリソースが存在し、それぞれが存在するシステムに特有のデータが各リソースに入っている場合に役立ちます。

この機能を使用すると、プログラムの独立性を損なうおそれがあります。アプリケーション・プログラムで SYSID オプションを使用することによって、機能シップ

するコマンドにリモート・システムの名前を明示的に指定することもできます。このオプションを指定すると、要求は指定されたシステムに直接ルーティングされるため、ローカル・システムにあるリソース定義テーブルは使用されません。SYSID オプションにローカル・システムを指定することもできます。こうすると、アクセス先 (ローカル・リソースまたはリモート・リソース) を実行時に決定することができます。

ファイル制御

機能シップを使用すると、リモート CICS システムにある VSAM ファイルや BDAM ファイルにアクセスすることができます。

次の点に注意してください。

- INQUIRE FILE、INQUIRE DSNAME、SET FILE、および SET DSNAME はサポートされません。
- 読み取り専用要求と更新要求の両方が可能です。ファイルは、そのシステムにおいて保護リソースとして定義することができます。
- リモート保護ファイルの更新は、アプリケーション・プログラムが同期点要求を出すか正常終了するまでコミットされません。
- リモート・ファイルが、接続された複数の CICS システムに存在する場合でも、ローカル・ファイルとリモート・ファイルの更新をリンクさせて同じ作業単位内で実行することができます。

重要:

VSAM RBA、BDAM、またはキーがレコードに組み込まれていないファイルなど、物理レコード ID の値を使用するリモート・ファイル要求の場合は、そのシステムの設計に注意が必要です。リモート・システム内のアプリケーション・プログラムすべてが、レコードの追加またはこれらのタイプのファイルの再編成を行った後で、正しい値にアクセスすることを確認してください。

DL/I

機能シップを使用すると、CICS トランザクションで、リモートの CICS システムに関連した IMS Database Manager (IMS DM) データベース、あるいはリモートの CICS Transaction Server for VSE システムに関連した DL/I データベースにアクセスすることができます。

CICS Transaction Server for z/OS バージョン 4 リリース 1 の通信相手となるシステムのリストについては、3 ページの『第 1 章 CICS 相互通信の紹介』を参照してください。

リモートの CICS Transaction Server for z/OS システムに関連した IMS データベースは、リモート・システムによって所有されるローカル・データベースであっても、IMS データベース制御 (DBCTL) を使ってアクセスされるデータベースであっても構いません。機能シップを行う CICS システム側から見ると、このデータベースはリモートです。

ファイル制御と同じように、リモート DL/I データベースの更新は、アプリケーションが同期点に到達するまでコミットされません。IMS DM では、各作業単位に対

し複数のプログラム仕様ブロック (PSB) をスケジュールすることはそれらの PSB が別々のリモート・システムにあるものとして定義されていてもできません。したがって、異なるシステムの DL/I 更新を単一の作業単位でリンクさせることはできません。

PSB がリモート・システムにあることを定義するには、PSB ディレクトリー・リスト (PDIR) を使用します。リモート・システムは、データベースと関連のプログラム連絡ブロック (PCB) 定義を所有します。

一時記憶域

機能シップを使用すると、アプリケーション・プログラムにおいて、リモート・システムにある一時記憶域キューとの間でデータをやりとりすることができます。

一時記憶域キューは、ローカルの一時記憶域テーブル (TST) 内の項目によって、リモートとして指定されます。キューを保護する場合は、リモート・システムの TST において、そのキュー名 (またはリモート名) もリカバリー可能として定義する必要があります。

一時データ

アプリケーション・プログラムは、リモート・システムにある区画内または区画外の一時データ・キューにアクセスすることができます。

要求側システムのそのキューの定義では、キューはリモート・システムにあるものとして定義されます。キューのリカバリー可能属性と、キューにトリガー・レベルと関連端末があるかどうかは、リモート・システムのそのキューの定義に指定されます。区画外キューは、所有システムにおいて、固定長または可変長のレコードを含むように定義することができます。

単一の CICS システム内にある一時データ・キューおよび一時記憶域キューの現在の使用法の多くは、相互接続されたプロセッサー・システム環境でも使用することができます。例えば、システムで夜間処理を行うためのレコードのキューを作成することができます。キューを使えば、他の要求のために端末を解放する一方で、他のシステムからの要求を処理することができます。応答は、端末が作動可能になるとその端末に返され、トランザクションの入力に小休止ができるとオペレーターに送達されます。

一時データ・キューにトリガー・レベルのトランザクションが関連付けられている場合、指定されたトランザクションは、そのキューを所有するシステムで実行されるように定義する必要があります。これをリモートとして定義することはできません。端末がトランザクションに関連付けられている場合、その端末を、別の CICS システムに接続して、CICS のトランザクション・ルーティング機能を介して使用することができます。

リモート命名機能を使用することにより、プログラムは、ローカル・システムとリモート・システムの両方の CICS サービス宛先 (CSMT など) にデータを送信することができます。

システム間のキューイング

空きセッションを待つ機能シップ要求が要求側の領域でキューに入れられると、パフォーマンス上の問題が起こる可能性があります。

バインドされたすべてのコンテンション勝者セッションが使用中のために、すぐに使用できるセッションがない場合、リソース所有領域へ機能シップされる要求はキューに入れられます。リソース所有領域からの応答が悪いと、キューが長くなり、要求側領域のパフォーマンスが著しく損なわれます。さらに、要求側の領域がアプリケーション所有領域である場合は、パフォーマンスの悪化は端末専有領域に及びます。

注: 「競合勝者」は、APPC 接続で使用される用語です。MRO と LUTYPE6.1 の接続では、SEND セッション (セッション定義で定義される) は、ALLOCATE 要求に対して使用されます。すべての SEND セッションが使用中になると、キューイングが始まります。

パフォーマンスの低下の症状は、次のように現れます。

- 多くのタスクの要求がキューに入れられるため、システムがその最大トランザクション (MXT) 限度に達する。
- システムがストレージ不足になる。

どちらの場合も、CICS で新しい作業を開始することはできません。

CICS には、これらの問題を防ぐための方法が 2 つあります。

- CONNECTION 定義の QUEUELIMIT と MAXQTIME オプション。これらのオプションを使用することにより、特定のリモート領域に対してキューイングできる要求の数や、応答の悪い接続において要求がセッションを待つ時間を制限することができます。
- 2 つのグローバル・ユーザー出口 XZIQUE と XISCONA。コンテンション勝者セッションをすぐに使用できない場合、XZIQUE 出口プログラムまたは XISCONA 出口プログラムが呼び出されます。この出口プログラムでは、要求をキューに入れるか、アプリケーション・プログラムに SYSIDERR を戻すかを CICS に指定することができます。どちらにするかは、ユーザー出口プログラムのパラメーター・リストからアクセス可能な統計に基づいて決めることができます。XZIQUE および XISCONA 出口プログラムを作成する場合のプログラミング情報については、「*CICS Customization Guide*」のシステム間通信プログラムの出口 XISCONA および XISLCLQ (Intersystem communication program exits XISCONA and XISLCLQ) を参照してください。出口プログラムに渡す統計レコードについては、「*CICS パフォーマンス・ガイド*」の CICS 統計の概要 (Introduction to CICS statistics) を参照してください。

  注: XISCONA 出口ではなく、XZIQUE 出口を使用することをお勧めします。XISCONA よりも XZIQUE の方が機能的に優れており、一般的な使用に適しています。この出口は、機能シップだけでなく、DPL、トランザクション・ルーティング、および分散トランザクション処理の各要求に対しても呼び出されますが、XISCONA は機能シップと DPL に対してしか呼び出されません。両方の出口を

使用可能にすると、機能シップおよび DPL 要求に対して XZIQUE と XISCONA の両方が呼び出される可能性があるので、両方の出口を使用可能にすることはお勧めしません。

XISCONA 出口プログラムが既にある場合は、これを修正することによって XZIQUE 出口点で使用できることがあります。

システム間キューの制御については、さらに 307 ページの『第 24 章 システム間のセッション・キューの管理』を参照してください。

ミラー・トランザクションと変換プログラム

CICS には、多数のミラー・トランザクションがあります。その中には、「体系化プロセス」に対応するものがあります。

提供されるミラー・トランザクションの詳細については、253 ページの『第 17 章 ローカル・リソースの定義』で説明します。ここでは、通常、これらをミラー・トランザクションと呼び、トランザクション ID「CSM*」を付けて示します。

ミラー・トランザクションおよび変換プログラムに関する以下の説明は、一般に、ISC および MRO の両方の機能シップに適用することができます。ただし、MRO でのミラー・トランザクションの動作方法には若干の違いがあり、使用される変換プログラムが異なります。これらの違いについては、44 ページの『MRO 機能シップ』で説明します。

ISC 機能シップ

ミラー・トランザクションは、通常の CICS トランザクションとして実行され、CICS 端末管理プログラムを使用して、要求側システムとの通信を行います。

要求側システム (43 ページの図 10 の CICA) では、要求されたリソースが別のシステム (例では CICB) にあることが、コマンド・レベルの EXEC インターフェース・プログラム (DL/I 要求を除くすべての要求の場合) によって判別されます。したがって、このプログラムは、機能シップ変換プログラムを呼び出して、その要求を伝送に適した形式に変換します。例では、線 (2) がこの呼び出しを示しています。次に、EXEC インターフェース・プログラムは、相互通信コンポーネントに、変換した要求を接続された該当システムへ送信するように要求します (3)。DL/I 要求の場合、この機能の一部は、CICS DL/I インターフェース・モジュールによって処理されます。DL/I 要求の処理ガイダンスについては、「 *CICS IMS Database Control Guide*」の IMS データベース管理 (DBCTL)を参照してください。

相互通信コンポーネントは、CICS 端末管理プログラムを使用して、ミラー・トランザクションに要求を送信します。トランザクションに代わって特定のリモート・システムに最初に要求を行うと、ローカル・システムにある通信コンポーネントが、形式設定された要求の前に適切なミラー・トランザクション ID を付けて、リモート・システム上のこのトランザクションに接続できるようにします。その後で、ミラー・トランザクションが停止したかどうかが追跡され、必要であればそれを再び呼び出します。

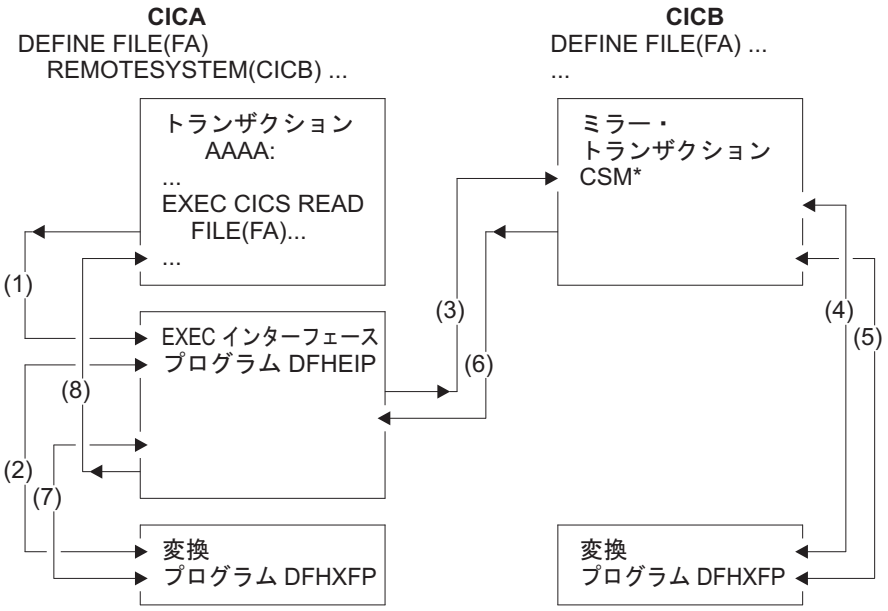


図 10. 機能シップにおける変換プログラムとミラー

図 10 のイベント順序は次のとおりです。

- ミラー・トランザクションは、機能シップ変換プログラム DFHXFP を使って、形式設定された要求をデコードします (図 10 の (4))。次に、ミラーが、対応するコマンドを実行します。このコマンドが完了すると、ミラー・トランザクションは、変換プログラムを使用して、応答を形式設定します (5)。ミラー・トランザクションは、形式設定した応答を要求側のシステム CICA へ戻します (6)。CICA では、その応答は再度変換プログラムを使用してデコードされ (7)、アプリケーション・プログラムが出した元の要求を完了するために使用されます (8)。
- ミラー・トランザクションが保護リソースを更新する必要がなく、それより前の要求でもシステム内の保護リソースを更新していない場合、ミラー・トランザクションは、応答の送信後に停止します。しかし、要求によって、ミラー・トランザクションが保護リソースを変更または更新する場合や、要求がいずれかの DL/I プログラム仕様ブロック (PSB) に対するものである場合、ミラー・トランザクションは、要求側アプリケーション・プログラムが同期点要求を出すか、正常に終了するまで停止しません。ブラウズが実行されている場合、ミラー・トランザクションは、そのブラウズが終了するまで終了しません。
- アプリケーション・プログラムが同期点要求を出すか正常に終了すると、相互通信コンポーネントは、ミラー・トランザクションにメッセージを送り、同期点要求を出して停止するように指示します。ミラー・トランザクションによる正常な同期点が、要求側システムに戻される応答に示されると、要求側システムは、同期点処理を終了します。これによって、保護リソースに対する変更すべてがコミットされます。DL/I 要求を他のシステムから受け取ると、CICS は、アプリケーション・プログラムによって出され、ミラー・トランザクションによって実行された同期点要求による処理の一部として、DL/I TERM 要求を出します。
- アプリケーション・プログラムは、保護リソースや無保護リソースに任意の順序でアクセスできるので、保護リソースの位置による影響は受けません。例えば、保護リソースすべてがリモート・システムにある可能性もあります。アプリケーション・プログラムが、複数のリモート・システムにあるリソースにアクセスす

ると、相互通信コンポーネントは、各システムでミラー・トランザクションを呼び出して、アプリケーション・プログラムの要求を実行します。各ミラー・トランザクションは、上記の規則に従って終了します。アプリケーション・プログラムが同期点に到達すると、相互通信コンポーネントは、終了していないミラー・トランザクションと同期点メッセージを交換します。この状態を多重ミラー といいます。

- ミラー・トランザクションは、CICS コマンド・レベル・インターフェースを使用して CICS 要求を実行し、DL/I CALL または EXEC DL/I インターフェースを使用して DL/I 要求を実行します。したがって、要求はその他のトランザクションの場合と同じように処理され、要求されたリソースを見つけるために、適切なリソース・テーブルが検索されます。その項目にリソースがリモートとして定義されていると、ミラー・トランザクションの要求は伝送用に形式設定され、指定システム内のさらに別のミラー・トランザクションに送信されます。この状態をチェーン・ミラー といいます。セッションの失敗によってデータの保全性が損なわれることのないように、チェーン・ミラー要求が起こるような接続システムはできるだけ定義しないでください。ただし、そのときの要求が保護リソースにアクセスしない場合や、照会のみの要求の場合は除きます。

MRO 機能シップ

MRO 機能シップのミラー・トランザクションの動作は、ISC 機能シップで説明した動作と若干異なります。

長期実行ミラー・タスク

通常、MRO ミラー・タスクは、ISC ミラーの場合と同じように、できるだけ速やかに停止されます。これは、アクティブ・タスクの数を最小にするとともに、そのセッションを長期にわたって保持することを避けるためです。

しかし、一部のアプリケーションでは、データ保全性という点では不要であっても、次の同期点までミラー・タスクとセッションの両方を保存した方が効率的な場合があります。例えば、多数の READ FILE 要求をリモート・システムに出すトランザクションは、各要求ごとのミラー・タスクではなく、単一のミラー・タスクによって処理した方が効率的です。このように、送信側におけるセッション割り振り、および受信側におけるミラー・タスクの接続といったオーバーヘッドを減らすことができます。

次の同期点を待っているミラー・タスクは、論理的には長時間の実行を必要としなくても、*長期実行*ミラー と呼ばれます。これらは、MRO リンクにのみ適用可能であり、システム初期設定パラメーターに MROLRM=YES とコード化することによって、ミラーが*実行*されるシステムに 指定されます。長期実行ミラーは、送信側の次の同期点 (または RETURN) によって停止されます。

アプリケーションによっては、長期実行ミラーを使用することによって著しいパフォーマンスの向上が得られる場合があります。

46 ページの『機能シップの例』の 47 ページの図 12と 48 ページの図 13 に、MROLRM=NO と MROLRM=YES それぞれに対しミラーがどのように作用するかを示します。

フロントエンド領域で指定される、追加のシステム初期設定パラメーター MROFSE=YES では、ミラー・タスクとセッションの保存が、次の同期点からタスクの終了まで拡張されます。最大限に利用するため、MROFSE=YES は、バックエンド領域上の MROLRM=YES と一緒に使用してください。ただし、ミラー・トランザクションがそのインバウンド・セッションを保持するようなタイプの要求の場合には、MROFSE=YES は、バックエンド領域で MROLRM=NO が指定されている場合でも適用されます。

概念上、MROLRM はバックエンド領域で指定し、MROFSE はフロントエンド領域で指定します。ただし、「バックエンド」と「フロントエンド」の区別がはっきりしていない場合は、必要に応じて各領域で両方のパラメーターをコード化しておけば安全です。

MROFSE=YES を指定したことによりパフォーマンスが向上するのは、フロントエンド領域から開始された大半のアプリケーションが複数の同期点を持っており、機能シップ要求が各同期点間で発行される場合だけです。

機能シップ要求に長期実行タスクが使用される可能性がある場合は、フロントエンド領域に MROFSE=YES を指定しないでください。これは、未使用時には、SEND セッションを他のタスクへの割り振りに使用できないためです。MROFSE=YES を指定すると、バックエンド領域との接点が失われた場合に、タスクが終了するか、機能シップ要求が発行されるまで、接続を解放できなくなる可能性があります。

短パス変換プログラム

CICS は、MRO リンクでの機能シップに、特殊な変換プログラム (DFHXFX) を使用します。

この*短パス変換プログラム* は、機能シップの MRO セッションで送信される端末入出力域 (TIOA) の作成に関してパス長さを最適化します。これは、変換された要求に対して、SNA によって定義された体系化形式ではなく、プライベート CICS 形式を使用してパス長さを最適化します。

CICS は、ファイル制御、一時データ、一時記憶域、およびインターバル制御 (非同期処理) の各要求をシップする際に DFHXFX を使用します。これは、DL/I 要求には使用されません。シップされた要求は常に、CICS ミラー・トランザクションの CSMI を指定します。体系化プロセス名は使用されません。

ミラー・トランザクションのエラーと障害の処理

リモート領域でミラー・タスクにエラーが発生したり、ミラー・タスクが異常終了した場合、ミラー・プログラムでそのエラーや異常終了に対応できるときには、機能シップ要求を出したアプリケーション・プログラムにエラーまたは異常終了が戻されます。

このタスクについて

リモート・ミラー (サーバー) タスクと、要求を出したプログラムを実行するタスク (クライアント・タスク) は、要求が以下のいずれかに該当する場合を除いて、共通のトランザクション・スコープを共用します。

- 機能シップされた EXEC CICS START NOCHECK コマンド

- SYNCONRETURN を持つ分散プログラム・リンク (DPL) 要求
- 非更新要求 (読み取り専用のファイル制御など)

サーバー・タスクがこのような共通のトランザクション・スコープの一部としてリカバリー可能作業を実行する場合、エラーや異常終了が発生した場合でも、その作業はクライアント・タスクの同期点処理の制御下でコミットまたはバックアウトされます。デフォルト・アクションでは、エラーまたは異常終了が発生すると、クライアント・タスクが異常終了し、クライアント・プログラムとサーバー・プログラムの両方で行われたリカバリー可能な更新がすべてバックアウトされます。

ただし、ローカルで実行された場合と同様に、つまり機能シップや分散プログラム・リンクを使用しなかった場合と同様に、機能シップされた要求を出したアプリケーション・プログラムは、エラーや異常終了に対応しようとします。その後、処理ロジックによって、EXEC CICS SYNCPOINT、SYNCPOINT ROLLBACK、RETURN、または ABEND コマンドが出されます。エラーまたは異常終了が通知される一方で、SYNCPOINT ROLLBACK や ABEND ではなく SYNCPOINT や RETURN が試行されることにより、クライアント・プログラムのローカル・リソースの更新、およびエラーや異常終了が発生する前にサーバー・トランザクションによって実行された更新のコミットが試行されます。

ミラー・トランザクションで発生したエラーや異常終了にミラー・プログラムで対応できないことが原因で、クライアント・アプリケーションに応答を送信することなくミラー・トランザクションが停止およびバックアウトする場合は、CICS によってクライアント・プログラムのトランザクションが強制的にバックアウトされます。明示的な同期点の試行は失敗し、ローカル更新はバックアウトします。クライアント・タスクとサーバー・タスク間の通信リンクで問題が発生した場合も、同様の処理が行われます。

クライアント・タスクとサーバー・タスクで共通のトランザクション・スコープを共用していない場合は、前述のように、エラーや異常終了によってサーバー・タスクが停止したとき、および通信リンクで問題が発生したときでも、クライアントのトランザクションは強制的にバックアウトされません。

機能シップの例

以下の例は、ミラー・トランザクションの存続時間と、アプリケーションとそのミラー (CSM*) 間の情報の流れを示しています。

これらの例では、ミラー・トランザクションがアプリケーション・プログラムの代わりにリソースにアクセスするときのアクションを、リソースが保護か無保護か、リンクが MRO か ISC であるか、あるいは MRO 長期ミラー・タスクを使用するかしないかで対比させて示しています。

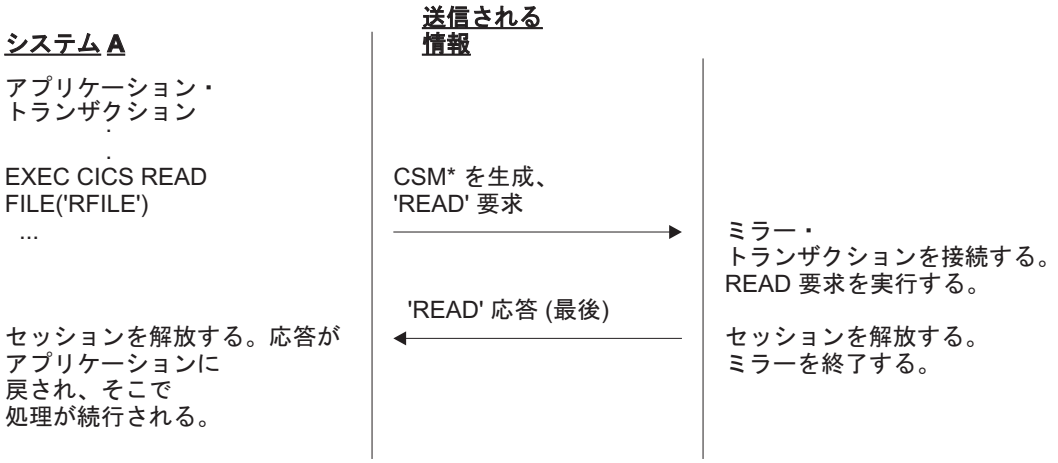


図 11. ISC 機能シップ - 単純照会: ここでは、リソースは変更されません。セッションは解放され、ミラー・タスクは即座に停止します。

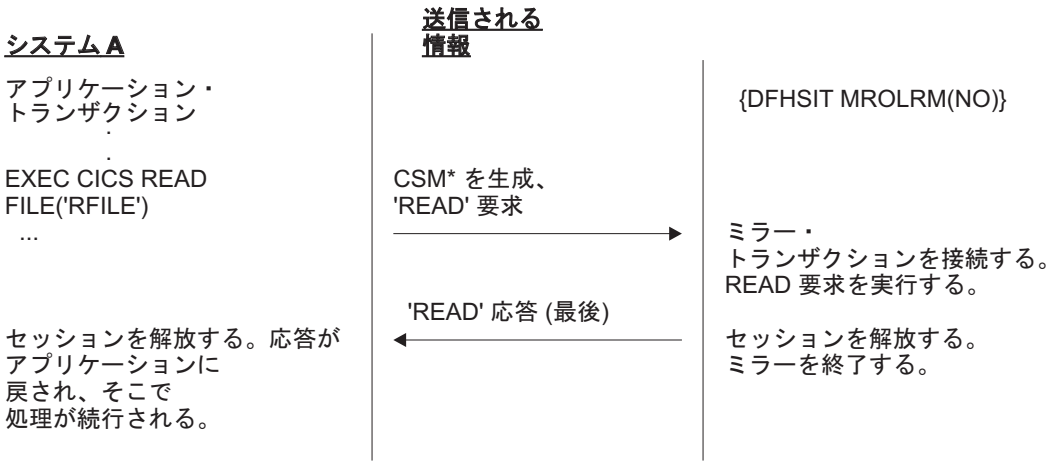


図 12. MRO 機能シップ - 単純照会: この例では、リソースは変更されません。長期実行ミラー・タスクは指定されていないため、セッションはシステム B によって解放され、ミラー・タスクはすぐに停止します。

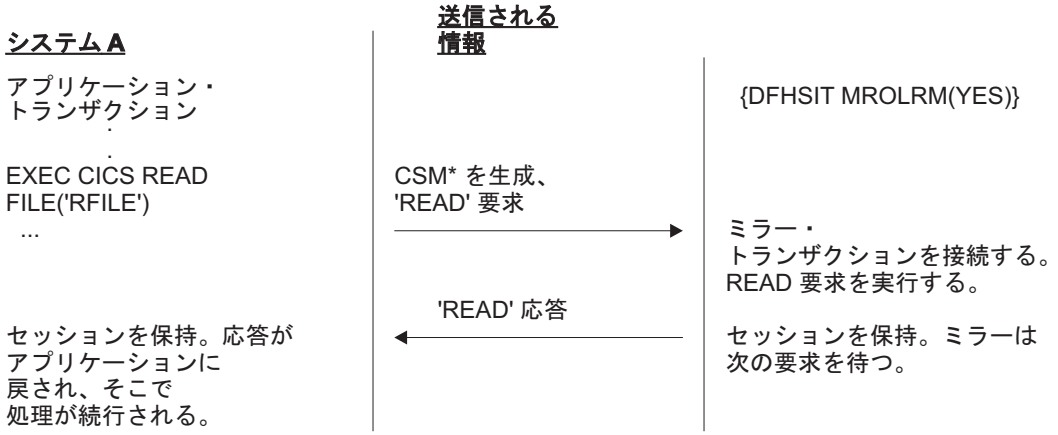


図 13. MRO 機能シップ - 単純照会： この例では、リソースは変更されません。ただし、長期実行ミラー・タスクが指定されているため、セッションはシステム B によって保持されて、ミラー・タスクは次の要求まで待機します。

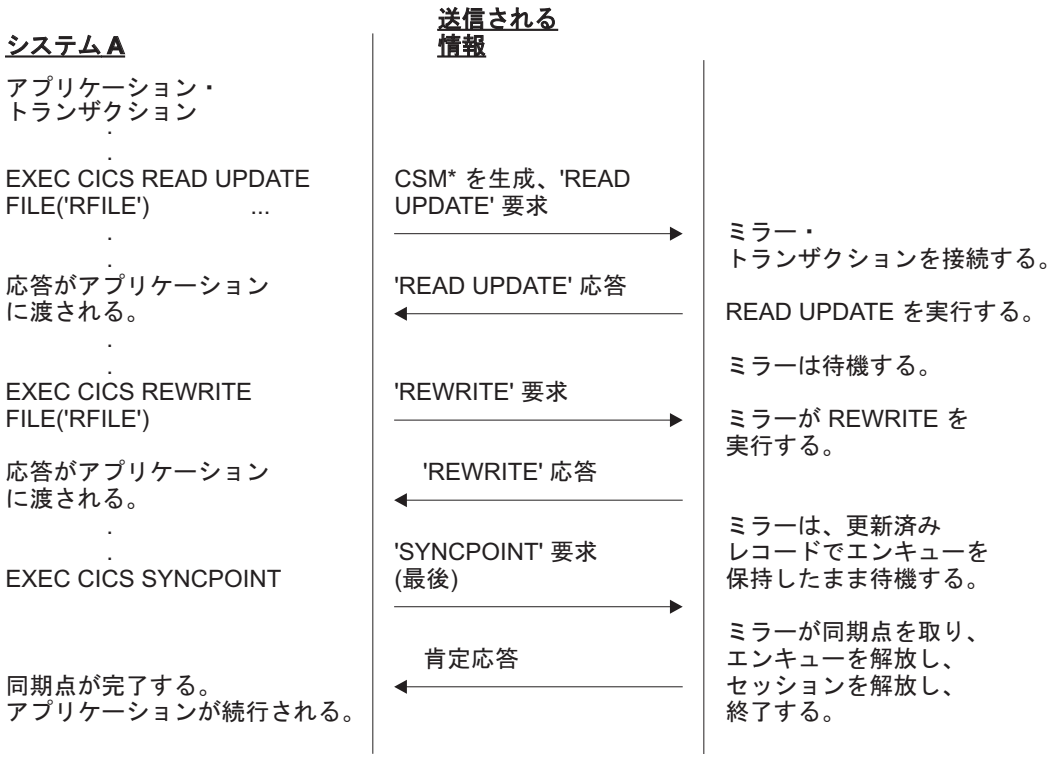


図 14. ISC または MRO 機能シップ - 更新： ミラーは、REWRITE まで待機しなければならないため長期実行タスクとなり、SYNCPOINT を受け取るまで終了しません。ファイルがリカバリー可能でない限り、更新されたレコードでのエンキューは、REWRITE コマンドが終わった後は保持されないことに注意してください。

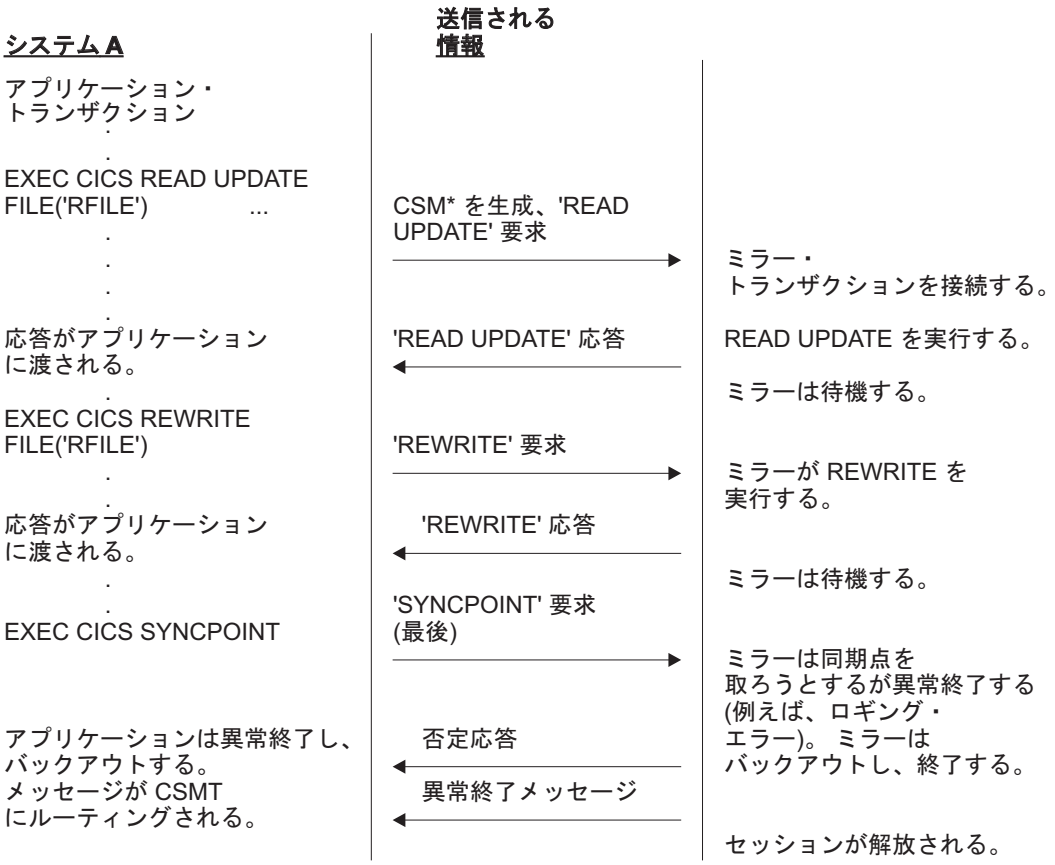


図 15. ISC または MRO 機能シップ - 更新 (ABEND)

図 15 は 48 ページの図 14 に似ていますが、アベンドが同期点処理中に発生するという点が異なります。

第 5 章 非同期処理

非同期処理によって、アプリケーションに必要な処理が相互通信システム間で配信されます。処理は、要求が送信されるセッションや応答が受信されるセッションから独立しています。

この章には以下のトピックが含まれています。

- 『非同期処理の概要』
- 52 ページの『非同期処理方式』
- 53 ページの『START と RETRIEVE コマンドを使用する非同期処理』
- 60 ページの『システム・プログラミングに関する考慮事項』
- 60 ページの『非同期処理の例』.

非同期処理の概要

非同期処理を行うと、相互通信環境内のシステム間でアプリケーションによって必要とされる処理を分散させることができます。ただし、分散トランザクション処理とは違って、この処理は**非同期**です。

分散トランザクション処理では、セッションは 2 つのトランザクションによって、それらの間の「会話」期間中保持され、要求と応答を直接対応させることができます。

非同期処理では、処理は、要求が送信されるセッションや応答が受信されるセッションから独立しています。要求と応答の間に直接の相関関係はなく、応答のタイミングを想定することはできません。これらの違いを図 16 に示します。

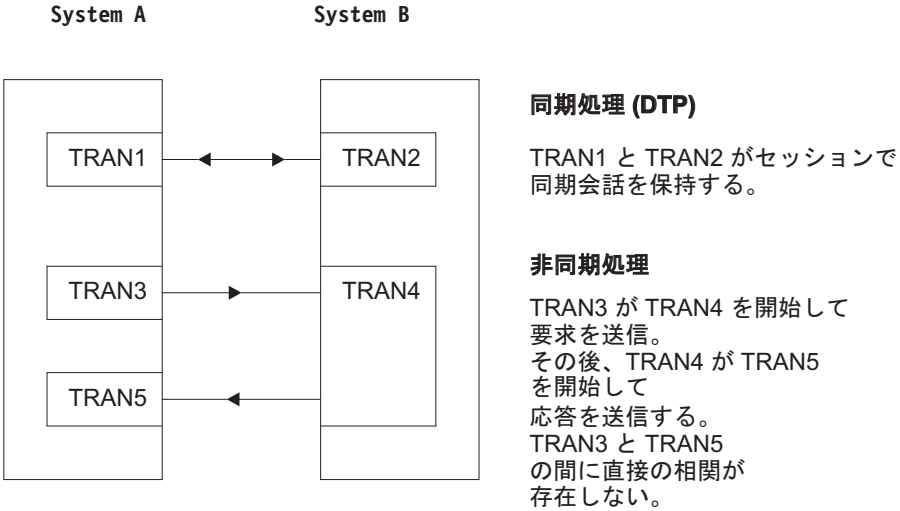


図 16. 同期処理と非同期処理の比較

非同期処理の代表的なアプリケーション領域として、リモート・データベースでのオンライン照会があります。例えば、信用格付けをチェックするアプリケーション

© Copyright IBM Corp. 1977, 2009

などです。端末オペレーターは、ローカル・トランザクションを使用して、各照会に対する応答があるまで待機することなく、一連の照会を続けて入力することができます。ローカル・トランザクションは、各照会ごとにリモート・トランザクションを開始して要求を処理するため、リモート・トランザクションの多数のコピーを同時に実行することができます。リモート・トランザクションは、ローカル・トランザクション (おそらく同じトランザクション) を開始して、オペレーター端末 (トランザクションを開始した端末) に出力を送達することによって、応答を送信します。応答は、照会が出されたときと同じ順序で受信されるとは限りません。照会と応答の相関関係は、ユーザー・データ内のフィールドによって確立する必要があります。

一般的に、非同期処理は、リモート要求を処理する際に、ローカル・リソースを結び付けておく必要がないか結び付けておきたくない状況に適しています。

非同期処理は、ローカル・リソースとリモート・リソースを同期的に変更しなければならないアプリケーションには適していません。例えば、2 つのシステム間で分割されているデータを同時にリンクさせて更新を処理するために使用することはできません。

非同期処理方式

CICS では非同期処理を次の 2 つの方法のどちらかで実行することができます。

1. インターバル制御コマンドの START と RETRIEVE を使用する。

   START コマンドを使用すれば、単一の CICS システムで行うのと同じように、リモート・システムのトランザクションをスケジュールすることができます。このタイプの非同期処理は、本質的には、一種の CICS 機能シップです。このため、アプリケーションが関知する必要はありません。システム・プログラマーは、接続されたトランザクションがローカルかリモートかを判別します。

   非同期処理に START コマンドを使用すると、機能シップに必要な特殊プロトコルをサポートするシステム (つまり、CICS 自体および IMS) のみと通信を行うことができます。

   リモート側で出された開始要求によって開始される CICS トランザクションは、RETRIEVE コマンドを出して、その要求に関連するすべてのデータを検索することができます。データ転送は、開始する側のトランザクションから開始されるトランザクションに渡される単一のレコードに限定されます。
2. 分散トランザクション処理 (DTP) を使用する。

   これは、システム間方式であり、単一システムにおける同等の方式はありません。この方法を使用すると、DTP プロトコルのいずれかをサポートするリモート・システムでトランザクションを開始することができます。

   DTP を使用してリモート・トランザクションに接続すると、セッションが同時に割り振られ、会話が開始されます。そのため、データを直接送信し、必要なら、リモート・トランザクションからデータを受け取ることもできます。トランザクションの設計によって、交換するデータの形式と量が決まります。例えば、繰り返し SEND コマンドを使用すれば、複数レコード・ファイルを渡すことができます。

データ交換を終了すると、会話が終了し、ローカル・トランザクションも終了しますが、リモート・トランザクションの処理は続行されます。

2 つのトランザクションがともに作動している間に、これらが従うプロシージャーは、使用中のプロトコルのアプリケーション・プログラミング・インターフェース (API) によって決まります。望ましいのは APPC ですが IMS と通信する必要がある場合は、LUTYPE6.1 を使用しなければなりません。この方式を MRO リンクにも適用すると、柔軟なデータ交換機能を利用することができます。

どのプロトコルを使用する場合でも、その規則に従う必要があります。会話がどんなに短くても、その会話の進行中は同期処理となります。コマンド順序、エラー・リカバリー、同期点機能の点から見ると、これは完全な DTP です。

どちらの形式の非同期処理でも (そして、同期処理でも)、CICS トランザクションは、EXEC CICS ASSIGN STARTCODE コマンドを使って、それ自身がどのようにして開始されたのかを知ることができます。

CICS-IMS 間通信には、前述した DTP 方式の特殊なケースが含まれます。この方式では、データ通信が、単一の RECEIVE で応答する 1 つの SEND LAST コマンドに限定されるため、本書ではこの方式を SEND/RECEIVE インターフェースと呼びます。これらの機能を使用できる条件については、283 ページの『第 23 章 CICS-IMS 間アプリケーション』で説明します。

この章の後半では、START および RETRIEVE コマンドを使用する非同期処理について説明します。分散トランザクション処理については、117 ページの『第 9 章 分散トランザクション処理』で説明します。

START と RETRIEVE コマンドを使用する非同期処理

次のインターバル制御機能コマンドを非同期処理に使用することができます。

- START
- CANCEL
- RETRIEVE

CICS インターバル制御機能のプログラミング情報については、「*CICS アプリケーション・プログラミング・ガイド*」のインターバル制御を参照してください。

リモート・トランザクションの開始と取り消し

インターバル制御機能の START コマンドは、リモートの CICS システムと IMS システムでトランザクションを非同期的にスケジュールに入れるために使用されます。このコマンドは機能シップされます。リモート・システムが CICS の場合は、ミラー・トランザクションがリモート・システムで呼び出されて、そのシステムで START コマンドが出されます。

このタスクについて

注:

動的に転送される START コマンドの取り消しについては、96 ページの『インターバル制御要求の取り消し』を参照してください。

CICS-CICS 間通信では、INTERVAL オプションか TIME オプションを使って、シップされる START コマンドに通常の方法で時間制御情報を含めることができます。TIME の指定は、CICS によってローカル・クロックに対応する時間間隔に変換され、そのあとコマンドがシップされます。システム間リンクの各終端は時間帯が異なる可能性があるため、システム間通信では通常、絶対時刻よりも時間間隔の方が適しています。

START コマンドに指定した時間間隔によって指定される時刻は、要求がリモート・システムにシップされる時刻ではなく、リモート・トランザクションが開始される時刻であることに特に注意してください。

リモート CICS システムにシップされた START コマンドは、そのシステムに CANCEL コマンドをシップすることによって、満了時間までの間であればいつでも取り消すことができます。特定の START コマンドは固有の ID (REQID) を持ちます。これは、START コマンドとそれに対応する CANCEL コマンドに指定することができます。CANCEL コマンドは、この ID を「知る」すべてのタスクが発行できます。

時間制御は、IMS システムに送られる START コマンドに指定することはできません。INTERVAL(0) を指定するか、あるいはデフォルトをとるようにする必要があります。したがって、IMS トランザクションに対する開始要求を出した後に、それを取り消すことはできません。

START コマンドによって渡される情報

START コマンドには、リモート・トランザクションがその開始時に情報を使用できるようにするためのオプションがいくつかあります。リモート・トランザクションが CICS システムにある場合は、そのトランザクションは RETRIEVE コマンドを使用して情報を獲得します。

このタスクについて

指定できる情報を下記のリストにまとめます。

- ユーザー・データ。FROM オプションに指定します。

  これは、リモート・トランザクションに情報を渡すための基本的な方法です。

  CICS-CICS 間通信では、QUEUE オプションに指定された一時データまたは一時記憶域のキューにおいて、追加データを使用できるようにすることができます。キューは、リモート・トランザクションが実行されるシステムがアクセス可能な CICS システムに置くことができます。

  QUEUE オプションを CICS-IMS 間通信に使用することはできません。
- 応答に使用されるトランザクション名と端末名。RTRANSID と RTERMID のオプションに指定します。

  これらのオプション (その値はローカル・トランザクションによって設定される) は、リモート・トランザクションが応答をローカル・システムに渡す手段となり

ます (つまり、応答時にリモート・トランザクションが指定する TRANSID と TERMID は、最初の要求時にローカル・トランザクションが指定した RTRANID と RTERMID です)。

- 端末名。TERMID オプションに指定します。

  CICS-CICS 間通信では、これは、リモート・トランザクションの開始時にそれに対応付けられる端末の名前になります。端末は、そのリモート・トランザクションを所有する領域に定義されているが、その領域には所有されていない場合があります。その場合、端末は、トランザクション・ルーティングの自動トランザクション開始 (ATI) 機能によって獲得されます。これについては、79 ページの『ATI によって開始されたトランザクションの従来のルーティング』を参照してください。

  グローバル・ユーザー出口の XICTENF と XALTENF をコーディングすることで、シップ*可能な* 端末が、アプリケーション所有領域に定義されていないケースに対応することができます。これについては、80 ページの『自動トランザクション開始用端末のシップ』を参照してください。

  CICS-IMS 間通信の場合、これはトランザクション・コードか LTERM 名です。

START コマンドによる sysid と applid の受け渡し

いくつかの異なるシステムから開始できるトランザクションがあり、それを開始したシステムに START コマンドを出す必要がある場合は、呼び出し側トランザクションすべてに、そのローカル・システムの SYSID または APPLID を START コマンドのユーザー・データの一部として送信させるようにすることができます。

このタスクについて

開始されたトランザクションは、ASSIGN SYSID コマンドによってそのローカル sysid を、あるいは ASSIGN APPLID コマンドによってその applid を知ることができます。

リモート・システムへの接続の名前が、リモート・システムの SYSIDNT システム初期設定パラメーターと一致する場合 (MRO の場合は、標準)、開始されたトランザクションは、渡された sysid を START コマンドに指定して応答することができます。

リモート・システムに対する APPC または LUTYPE6.1 接続の名前が、リモート・システムの SYSIDNT システム初期設定パラメーターに一致しない場合でも、開始されたトランザクションは、応答を戻す相手の sysid を判別することができます。そのためには、渡された applid を NETNAME オプションに指定して EXTRACT TCT コマンドを出します。

システム間 START 要求のパフォーマンスの向上

多数の照会専用アプリケーションでは、複雑なエラー検査およびリカバリー・プロシージャーは認められていません。トランザクションが照会だけを行う場合、端末オペレーターは、特定の時間内に応答を受け取らなければ、操作を再試行すること

ができます。この場合、START コマンドの NOCHECK オプションを使用すると、リモート・システムとの間でやりとりされるメッセージの数を大幅に減らすことができます。

このタスクについて

2 つのシステムが VTAM を介して接続されているときにこのようにすると、パフォーマンスは大幅に向上します。パフォーマンスが向上する一方で、CICS は、START コマンドにおいていくつかのタイプのエラーを検出できなくなります。

一般に、START NOCHECK コマンドは、この章の冒頭で説明したリモート照会アプリケーションで使用します。

端末オペレーターの照会によって接続されたトランザクションは、NOCHECK オプションを指定した適切な START コマンドを出します。これによって、1 つのメッセージが該当するリモート・システムに送信され、その照会を行うトランザクションが非同期で開始されます。このコマンドには、オペレーターの端末 ID を指定しなければなりません。これにより、オペレーターの端末に接続されたトランザクションは終了し、端末は応答を受け取るか、あるいは別の要求を開始することができます。

リモート・システムは、そのローカル・データベースについて要求された照会を実行してから、発信元のシステムに開始要求を出します。このコマンドは、要求されたデータとオペレーターの端末 ID を戻します。ここでも、2 つのシステム間で渡されるメッセージは 1 つだけです。発信元のシステムで次に開始されるトランザクションは、データを形式設定して、オペレーターの端末にそれを表示しなければなりません。

システムやセッションが失敗した場合には、端末オペレーターは照会を再度入力する必要があります。その場合、重複した応答が受信されます。オペレーターがこれを判別できるようにするには、各要求に相関フィールドを含めるか、すべての応答が自明のものでなければなりません。

NOCHECK オプションを使用した相互通信の例を、63 ページの図 18に示します。

NOCHECK オプションは、リモート・システムとのリンクの確立が未処理で START コマンドのシップがキューイングされる場合 (58 ページの『START コマンドのローカル・キューイング』を参照) や、要求が IMS にシップされる場合に常に必要です。

作業単位への開始要求送達の組み込み

START コマンドに PROTECT オプションを指定することにより、リモート・システムへの開始要求の送達を作業単位の一部とすることができます。

このタスクについて

PROTECT オプションは、ローカル・トランザクションが同期点を正常に完了するまでリモート・トランザクションをスケジュールしてはならないことを示します。(同期点は、SYNCPOINT コマンドを出すか、正常に終了することによってとられます。)

同期点が正常に終了すれば、開始要求はリモート・システムに確実に送達されています。ただし、これは、リモート・トランザクションが完了したことはもちろん、リモート・トランザクションが開始されたかどうかさえ保証するものではありません。

リモート・システムが IMS の場合は、START コマンドと同期点の間に、リンクを介してメッセージを送信することは一切できません。すべての IMS リカバリー可能トランザクションには、PROTECT と NOCHECK の両方を指定する必要があります。

NOCHECK オプションを指定した START 要求の ISC リンクへの据え置き伝送

START コマンドに NOCHECK オプションを指定した場合、PROTECT が指定されているかどうかに関係なく、CICS は、ISC リンクのリモート・システムに対する要求の伝送を据え置くことがあります。MRO リンクおよび IP 相互接続 (IPIC) の場合、NOCHECK を指定した START 要求は据え置かれません。

ISC リンクの場合、NOCHECK を指定した START 要求は、次のいずれかのイベントが起こるまで据え置かれます。

- トランザクションが、同じシステムに対してさらに START コマンド (または機能シップ要求) を出す。
- トランザクションが SYNCPOINT コマンドを出す。
- トランザクションが暗黙同期点で停止する。

トランザクションからリモート・システムに最初に (または唯一) 伝送された開始要求は、ブラケット開始標識を送ります。最後の (または唯一の) 開始要求は、ブラケット終了標識を送ります。さらに、トランザクションによって出された開始要求のどれかに PROTECT が指定されていると、その作業単位 (UOW) 内の最後の要求が同期点要求標識を送ります。据え置き送信によって据え置きデータに標識が追加されるので、必要な伝送の回数が減ります。

プロトコル、接続、または受信システムによる制限がある場合、START 要求は別に処理されます。

- APPC および LUTYPE6.1 の両方のプロトコルで、NOCHECK を指定した最初の START に、同じく NOCHECK コマンドを指定した 2 番目の START が続く場合、CICS は最初の START を伝送して、2 番目の START を据え置きます。
- LUTYPE6.1 プロトコルおよび 6.2 プロトコルでは、一連の要求は 1 つの SNA ブラケット内で伝送されます。リモート・システムが CICS の場合、すべての要求は同じミラー・タスクによって処理されます。
- MRO および IPIC 接続で、NOCHECK を指定した最初の START に、同じく NOCHECK コマンドを指定した 2 番目の START が続く場合、CICS は両方のコマンドを伝送します。
- IMS では、START 要求とその後に続く同期点の間で、リンクを介してメッセージを送信することはできません。したがって、複数の START NOCHECK PROTECT 要求を IMS に送信することはできません。各要求に続けて SYNCPOINT コマンドを出すか、またはトランザクションを終了する必要があります。IP 相互接続 (IPIC) では、IMS に要求を送信できません。

システム間のキューイング

リモート領域へのリンクが確立されても、使用できる空きセッションがないと、リモート・トランザクションをスケジュールするために機能シップされた EXEC CICS START 要求は、それを出した領域でキューに入れられることがあります。

キューが長くなりすぎると、パフォーマンスが問題になる場合があります。この問題については、41 ページの『システム間のキューイング』を参照してください。

システム間キューを制御するためのガイダンスについては、307 ページの『第 24 章 システム間のセッション・キューの管理』を参照してください。

START コマンドのローカル・キューイング

リモート・システムが、アクティブでないか、接続が確立できないために使用可能でない場合、それに対して START 要求を機能シップすると、通常は SYSIDERR 条件がアプリケーションに戻されます。

リモート・システムがこの CICS システムに直接接続されており、START コマンドに NOCHECK オプションが指定されている場合は、その要求をローカルでキューに入れ、必要なリンクがサービス可能になったときに伝送することができます。

要求がローカル・キューにある間に、START 要求を取り消すことはできません。要求を取り消すことができるのは、必要なリンクがサービス中の状態に戻り、要求がターゲット領域に送信された一方で、その要求が実行されていないときのみです。

SYSIDERR 条件は、リモート・システムへの接続が確立されている 一方で、使用できるセッションがなく、要求を出した領域でその要求がキューに入れられることになっていない場合にも戻されます。ローカル・キューイングは、次の 2 つの方法で指定できます。

1. リモート・トランザクションのローカル定義に LOCALQ(YES) を指定する。
   LOCALQ オプションを指定すると、特定のリモート・トランザクションに対するローカル・システムのすべての要求には、必要な場合、ローカル・キューが使用されます。

   LOCALQ オプションについての詳細は、「*CICS Resource Definition Guide*」を参照してください。
2. XISLCLQ グローバル・ユーザー出口プログラムまたは XISQLCL グローバル・ユーザー出口プログラムを使用する。

   XISLCLQ は、次の条件が適用される場合に、非 IPIC 接続にスケジュールされている、機能シップされた EXEC CICS START NOCHECK コマンドに対してのみ呼び出されます。
   - そのリモート・システムは使用不可である。または
   - そのリモート・システムへの接続は確立されているが、使用できるセッションがなく、かつ、そのコマンドを出した領域で現在キューイングされている要求の数が、CONNECTION 定義の QUEUELIMIT オプションに指定された最大数に達したか、XZIQUE または XISCONA グローバル・ユーザー出口プログラムの指定によって、その要求が、そのコマンドを出した領域にキューイングされないことになっているかのどちらか である。

XISQLCL は、次の条件が適用される場合に、IPIC 接続にスケジュールされている EXEC CICS START NOCHECK コマンドに対して呼び出されます。

- IPIC 接続が獲得されなかった場合。
- セッションが使用できないため、CICS で新しいセッションの要求をキューに入れることができなかった場合。

接続リソースが廃棄された場合、ローカル・キューに追加された要求が失われます。

ユーザー出口プログラムは、要求ごとに、それをローカルでキューイングするかどうかを決めることができます。

XISCONA、XISLCLQ、および XISQLCL の各グローバル・ユーザー出口のプログラミング情報については、「*CICS Customization Guide*」を参照してください。

開始されたトランザクションによるデータ検索

開始要求によって開始される CICS トランザクションは、RETRIEVE コマンドを使用して、要求に関連したユーザー・データやその他の情報を入手することができます。

トランザクションに対する開始要求で、ユーザー・データと端末 ID の両方を送る場合、そのトランザクションが既にアクティブで同じ端末に関連付けられていると、CICS インターバル制御についての通常の規則に従って、開始要求はキューイングされます。この待機中に、キューイングされた要求に関連するデータをアクセスするには、アクティブのトランザクションからさらに RETRIEVE コマンドを使用します。これは、キューイングされた開始要求を取り消す効果をもちます。

したがって、複数の開始要求に関連するデータを処理できるトランザクションを設計することが可能です。一般的な例としては、同じ端末から複数の照会を受け付け、開始要求をリモート・システムにシップする長期実行のローカル・トランザクションを設計することができます。それらのトランザクションは、時々 RETRIEVE コマンドを出して応答を受信します。応答がこれ以上存在しないと、ENDDATA 状態になります。

RETRIEVE コマンドの WAIT オプションを使用すると、リモート・システムから次の開始要求が来るまでの間、そのトランザクションを待ち状態にすることができます。APPC デバイスに接続されたタスクでこのオプションを使用すると、CICS では、データが存在しない場合、そのタスクを延期せずに、ENDDATA 条件が起こります。しかし、APPC デバイス以外の装置に接続されたタスクの場合には、新しい開始要求がないとき、トランザクションが永続待ち状態に入ることのないようにしなければなりません。

重要:

開始済みトランザクションが複数の RETRIEVE コマンドを発行したり、または RETRIEVE コマンドの WAIT オプションを使用したりする場合は、*START コマンドが発行される領域内で、トランザクション定義の ROUTABLE オプションをデフォルトで ROUTABLE(NO) に設定*できるようにしてください。トランザクションを

ROUTABLE(YES) として定義すると、複数の RETRIEVE または RETRIEVE WAIT コマンドが予想外の働きをすることがあります。

START コマンドの ROUTABLE オプションについては、87 ページの『START コマンドで呼び出されたトランザクションのルーティング』を参照してください。

リモートで開始された CICS トランザクションによる端末の獲得

端末 (TERMID) を指定した開始要求によって CICS トランザクションが開始された場合、CICS は、トランザクションがその端末を基本機能として使用できるようにします。

開始要求は、ローカル CICS システムにおいてユーザー・トランザクションが出しても、ミラー・トランザクションがリモート・システムから受け取って出しても同じです。

ISC または MRO セッションによるトランザクションの開始

START コマンドの TERMID オプションには、端末ではなくシステムを指定することができます。

このタスクについて

CICS は、ローカルまたはリモートで出された開始要求に指定された「端末」がシステムであることを認識すると、そのシステムで使用可能なセッションを選択し、それを開始されたトランザクションの基本機能にします (265 ページの『用語』を参照)。使用可能なセッションがないと、要求は、それが見つかるまでキューイングされます。

システムへのリンクが APPC リンクの場合、CICS は、トランザクション定義に関連するモード名を使用して、セッションのサービス・クラスを選択します。

システム・プログラミングに関する考慮事項

このセクションでは、非同期処理を行うために定義する必要がある CICS リソースについて説明します。

- リモート・システムへのリンクを定義する必要があります。
- 開始要求によって開始されるリモート・トランザクションは、ローカル CICS システムにリモート・リソースとして定義する必要があります。ただし、SYSID オプションに明示的にリモート・システムを指定した START コマンドだけで開始されるトランザクションの場合は、この定義は必要ありません。
- QUEUE オプションを使用する場合は、指名したキューを開始要求のシップ先のシステムに定義しなければなりません。キューは、そのシステムのローカル・リソースとリモート・リソースのどちらにでもすることができます。
- START 要求が「応答」トランザクションを指定する場合は、そのトランザクションを、開始要求のシップ先システムに定義する必要があります。

非同期処理の例

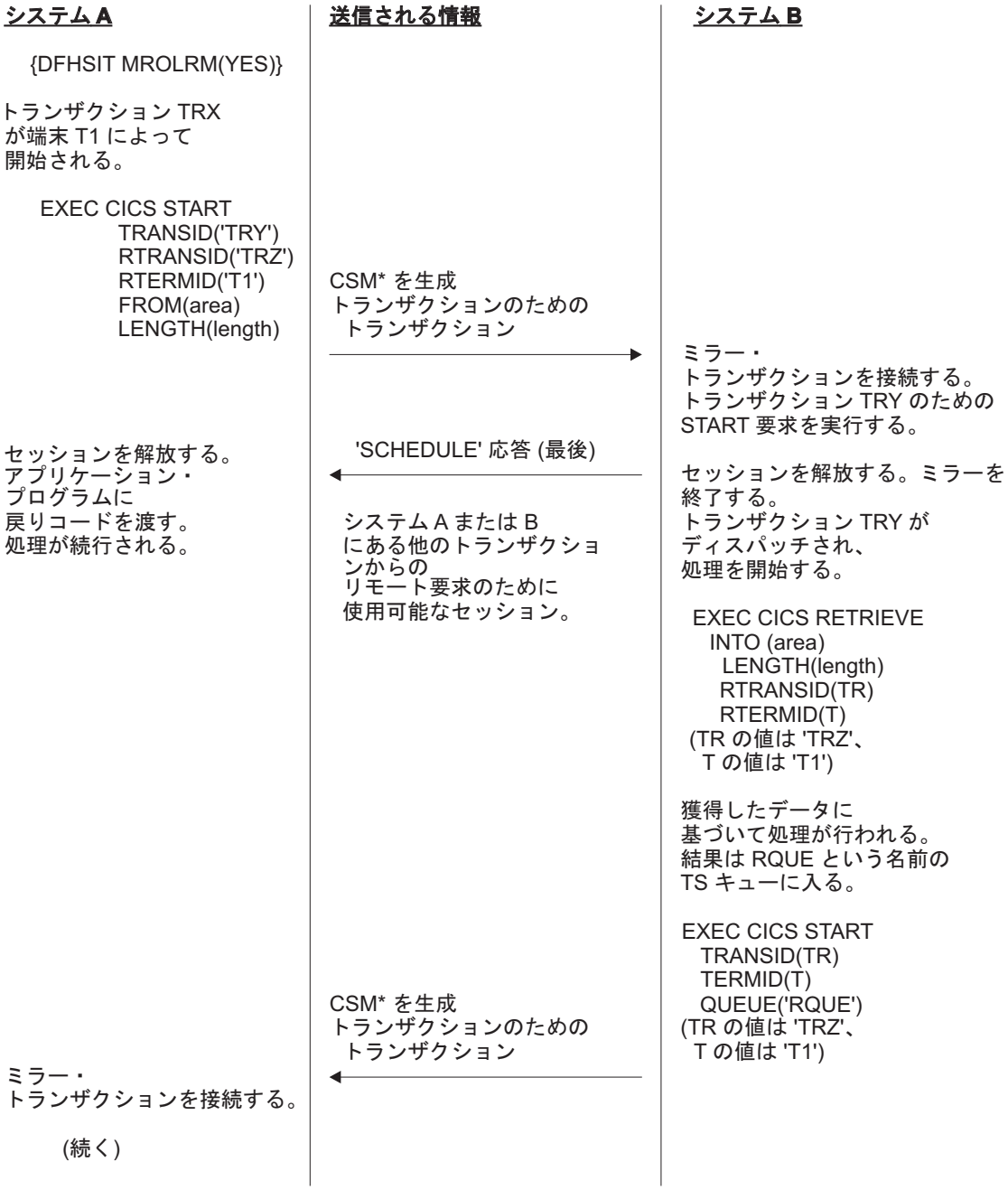
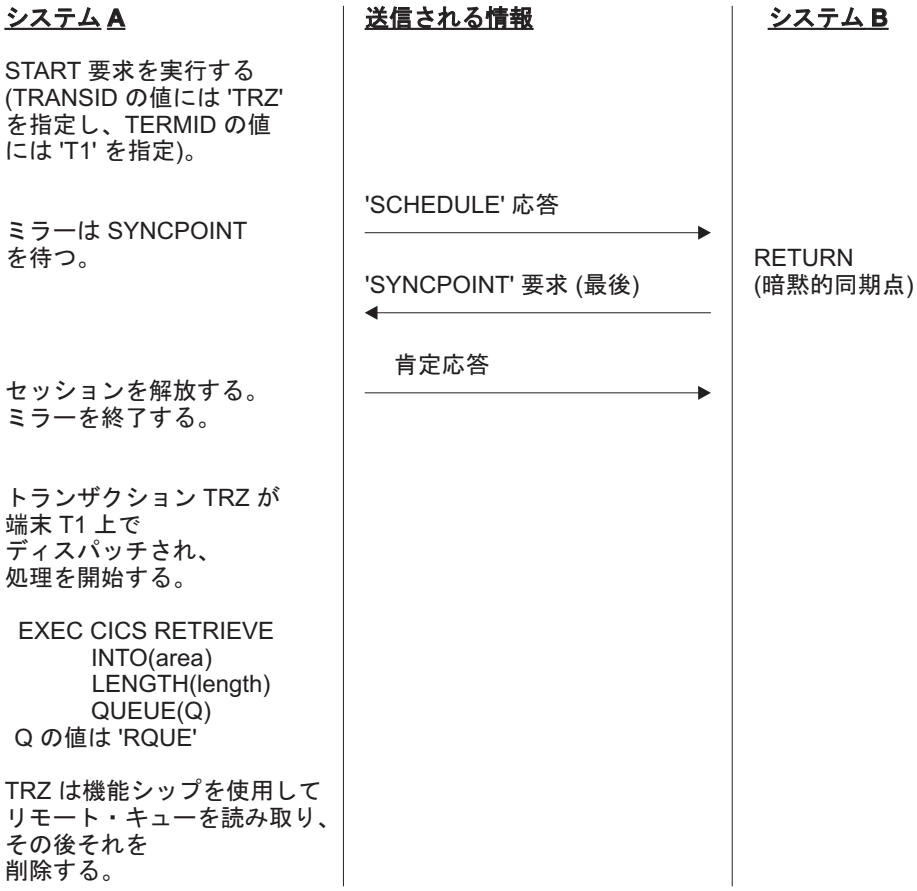


図 17. 非同期処理 - リモート・トランザクションの開始： この例では、長期実行ミラー (MROLRM) がシステム A に対しては指定されているが、システム B に対しては指定されていない MRO 接続を示します。 2 つのシステムでのミラー・トランザクションのアクションの違いに注意してください。


システム A
送信される情報
システム B
START 要求を実行する
(TRANSID の値には 'TRZ'
を指定し、TERMID の値
には 'T1' を指定)。
'SCHEDULE' 応答
ミラーは SYNCPOINT
を待つ。
RETURN
(暗黙的同期点)
'SYNCPOINT' 要求 (最後)
肯定応答
セッションを解放する。
ミラーを終了する。
トランザクション TRZ が
端末 T1 上で
ディスパッチされ、
処理を開始する。
EXEC CICS RETRIEVE
INTO(area)
LENGTH(length)
QUEUE(Q)
Q の値は 'RQUE'
TRZ は機能シップを使用して
リモート・キューを読み取り、
その後それを
削除する。

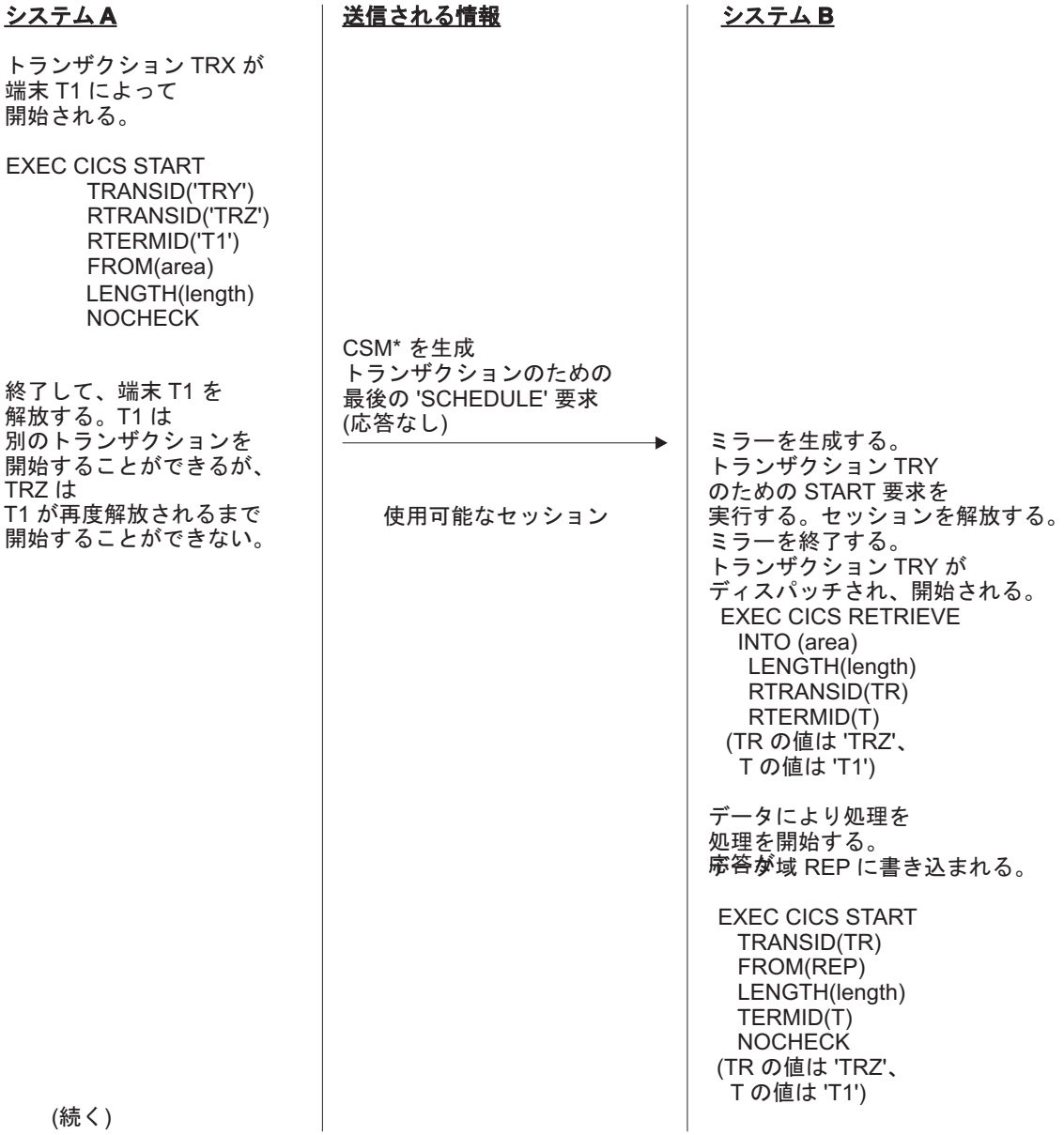
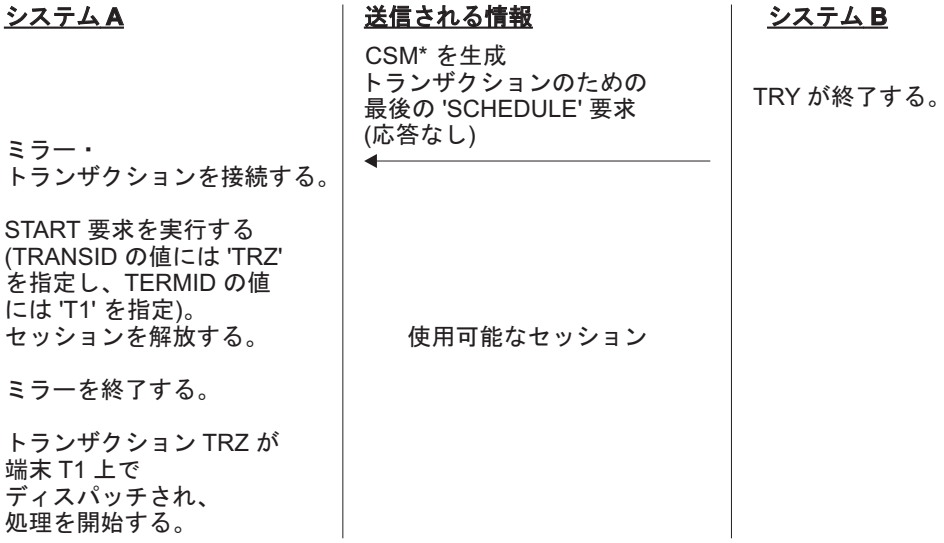


図 18. 非同期処理 - NOCHECK を使用したリモート・トランザクションの開始： この例は、ISC 接続、または長期実行ミラーのない MRO 接続を示しています。



63 ページの図 18 は、ISC 接続、または長期実行ミラーのない MRO 接続を示しています。

第 6 章 CICS 動的ルーティングの紹介

この章では、CICS の動的ルーティング・インターフェースの概要を紹介します。

ここに記載する情報は、73 ページの『第 7 章 CICS トランザクション・ルーティング』と 105 ページの『第 8 章 CICS 分散プログラム・リンク』の両方に関係しています。

動的ルーティングとは何か

CICSplex では、ある領域が必要とするリソース (例えば、トランザクションやプログラム) が、別の領域 (リソース所有領域) によって所有されていることがあります。例えば、あるアプリケーション所有領域が所有しているトランザクションへのアクセスを必要とする端末専有領域です。

静的ルーティング
リモート・リソースの場所を設計時に指定することを意味します。特定のリソースの要求は、常に同じ領域に送られます。一般に、静的ルーティングを使用する場合は、リソースの場所をインストール済みリソースの定義で指定します。

動的ルーティング
リモート・リソースの場所を実行時に指定することを意味します。この指定は、CICS 提供の、ユーザーが置き換えることのできる**ルーティング・プログラム**が行います。ルーティング・プログラムは、特定のリソースの要求を、要求ごとに別々の領域に送ることができます。つまり、例えば複数の複製されたアプリケーション所有領域がある場合には、ルーティング・プログラムはワークロードを領域全体で動的に分散させることができます。

動的にルーティングできるものは次のとおりです。

- 端末から開始されるトランザクション。
- EXEC CICS START コマンドのサブセットによって呼び出されるトランザクション。
- CICS-CICS 間分散プログラム・リンク (DPL) 要求。
- CICS の外側から受け取るプログラム・リンク要求 (例えば、CICS クライアントから受け取る外部呼び出しインターフェース (ECI) 呼び出し)。
- CICS ビジネス・トランザクション・サービス (BTS) のプロセスおよび活動 (BTS については、「*CICS Business Transaction Services*」の BTS の概説に説明があります。)
- Enterprise Beans および CORBA ステートレス・オブジェクトに対するメソッド要求 (エンタープライズ Bean については、「*Java™ Applications in CICS*」を参照)。
- ブリッジ 3270 トランザクション

そのほかに、次の定義が必要です。

© Copyright IBM Corp. 1977, 2009

要求側領域

トランザクションまたはその他のルーティング可能要求が出された領域。次に、「要求側領域」の例を示します。

- 端末から開始されたトランザクションの場合は、端末専有領域 (TOR) です。
- EXEC CICS START コマンドで開始されたトランザクションの場合は、START コマンドが発行された領域です。
- 「従来の」CICS-CICS 間 DPL 呼び出しの場合は、EXEC CICS LINK PROGRAM コマンドが発行された領域です。
- CICS の外側から受け取ったプログラム・リンク呼び出しの場合は、呼び出しを受け取った CICS 領域です。
- BTS プロセスおよび活動の場合は、EXEC CICS RUN ACTIVITY ASYNCHRONOUS コマンドが発行された領域です。
- Enterprise Beans または CORBA ステートレス・オブジェクトのメソッド要求の場合は、
  - メソッド呼び出しが CICS の外部 (リモート (非 CICS) IIOP クライアントなど) で発行される場合、要求側領域は、呼び出しを受け取るリスナー領域です。
  - メソッド呼び出しが CICS 内部 (別のエンタープライズ Bean のメソッドを呼び出すエンタープライズ Bean オブジェクトなど) で発行される場合、要求側領域は、呼び出しが発行される領域です。

ルーティング領域

経路選択にルーティング・プログラムが起動された領域。要求側領域とルーティング領域は、常に同じ領域となりますが、例外が 2 つあります。例外は、次のとおりです。

1. 端末関連 START コマンド:
   - 端末関連 START コマンドは常に端末専有領域で実行されるので、要求側領域とルーティング領域は同じである場合と、同じでない場合があります (詳細については、87 ページの『START コマンドで呼び出されたトランザクションのルーティング』を参照してください)。
   - ルーティング領域は常に TOR です。
2. CICS 内部から出された Enterprise Beans または CORBA ステートレス・オブジェクトに対するメソッド要求:
   - ローカル EJB/CORBA サーバー上の Enterprise Bean、プログラム、またはオブジェクトは、リモート EJB/CORBA サーバー上のオブジェクトのメソッドを呼び出します。要求側領域は、メソッドの呼び出しが行われるローカル領域です。ルーティング領域は、リモート EJB/CORBA サーバー上のリスナー領域です。

ターゲット領域

ルーティングされたトランザクションまたは要求が実行される領域。

2 つのルーティング・モデル

動的ルーティング・モデルには、次の 2 つがあります。

「ハブ」モデル

「ハブ」は、従来は CICS 動的トランザクション・ルーティングで使用されていたモデルです。

TOR で実行されているルーティング・プログラムは、複数の AOR 間でトランザクションをルーティングします。通常、AOR は (AOR/TOR でない限り) 動的ルーティングを行いません。図 19 は「ハブ」ルーティング・モデルです。

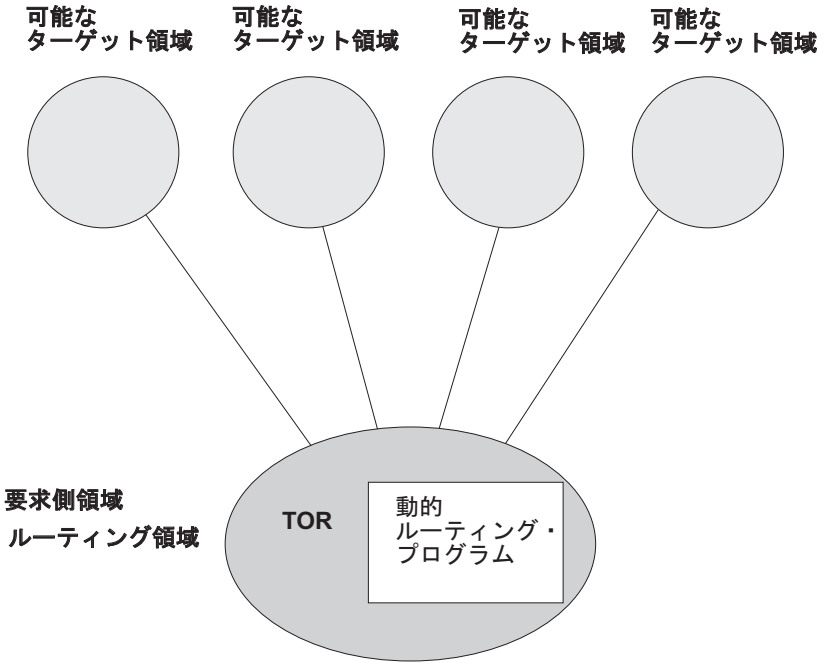


図 19. 「ハブ」ルーティング・モデルを使用した動的ルーティング: 1 つのルーティング領域 (TOR) が、複数のターゲット領域の間で選択します。

「ハブ」モデルは、次のルーティングに適用されます。

- 端末から開始されるトランザクション。
- 端末関連の START コマンドで開始されるトランザクション。
- CICS の外側から受け取ったプログラム・リンク要求 (受信側領域は、要求をバックエンド・サーバー領域のセットの間でルーティングするので、「ハブ」または「TOR」として動作します)。
- ブリッジ 3270 要求

「ハブ」モデルは*階層型* のシステムです。つまり、ルーティングは 1 つの領域 (TOR) で制御され、通常、ルーティング・プログラムは TOR 内でのみ実行されます。

「ハブ」モデルの利点

比較的簡単に実装できるモデルです。例えば、分散モデルと比べ、維持するリージョン間接続は少なくなります。

「ハブ」モデルの欠点

- 1 つの「ハブ」を使用して AOR 間でトランザクションおよびプログラム・リンク要求を使用すると、「ハブ」TOR は 1 つの障害点となります。
- 複数の「ハブ」を使用して同じ AOR セット間でトランザクションおよびプログラム・リンク要求をルーティングすると、分散データで問題が発生することがあります。例えば、ルーティング・プログラムがロード・バランシングを取る目的でルーティング・トランザクションのカウントを保持している場合、各「ハブ」TOR はこのデータにアクセスする必要があります。

分散モデル

分散モデルでは、各領域がルーティング領域とターゲット領域の両方になることがあります。

ルーティング・プログラムは、各領域で実行されます。 69 ページの図 20 は、分散ルーティング・モデルを示しています。

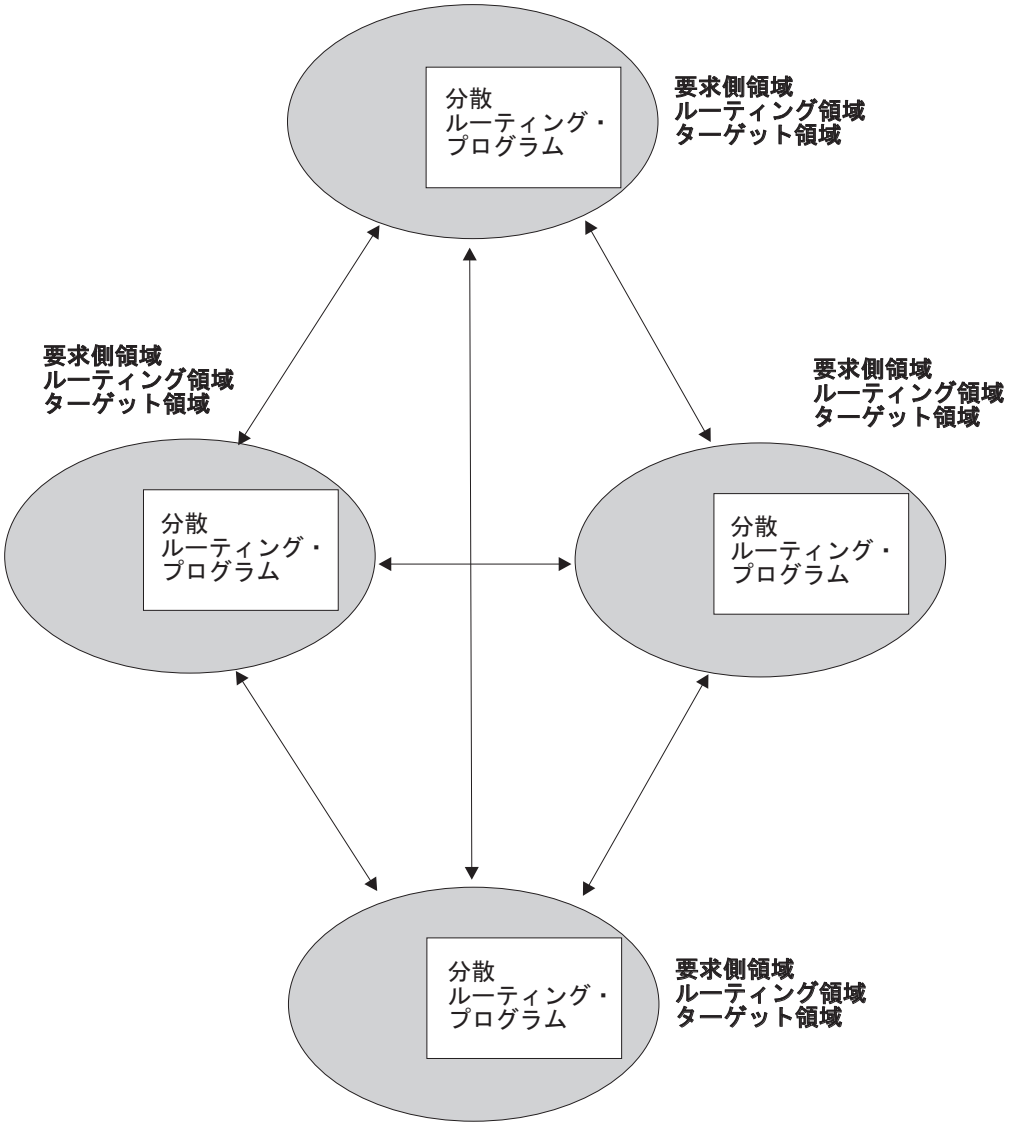


図 20. 分散ルーティング・モデルを使用した動的ルーティング: 各領域がルーティング領域とターゲット領域の両方になることがあります。

分散モデルは、次のルーティングに適用されます。

- CICS ビジネス・トランザクション・サービスのプロセスおよび活動
- Enterprise Beans および CORBA ステートレス・オブジェクトに対するメソッド要求
- 非端末関連の START 要求
- CICS-CICS 間 DPL 要求

分散モデルはピアツーピア・システムです。つまり、関係のある各 CICS 領域は、ルーティング領域とターゲット領域の両方になることがあります。ルーティング・プログラムは、各領域で実行されます。

分散モデルの利点

分散モデルが 1 つの障害点になることはありません。

分散モデルの欠点

- 「ハブ」モデルと比べて、保持するリージョン間接続がかなり多くなります。
- 分散データで問題が発生することがあります。例えば、ルーティングを決定する場合に使用するデータは、すべての領域で利用可能でなければなりません (CICSPlex SM では、データ・スペースを使用することでこの問題を解決しています)。

2 つのルーティング・プログラム

CICS には、ユーザーが置き換えることのできる、動的ルーティング用のプログラムが 2 つ用意されています。

動的ルーティング・プログラム、DFHDYP
次の動的ルーティングに使用できます。

- 端末から開始されるトランザクション
- 端末関連の START コマンドで開始されるトランザクション
- CICS-CICS 間 DPL 要求
- CICS の外側から受け取ったプログラム・リンク要求
- ブリッジ 3270 要求

分散ルーティング・プログラム、DFHDSRP
次の動的ルーティングに使用できます。

- CICS ビジネス・トランザクション・サービスのプロセスおよび活動
- Enterprise Beans および CORBA ステートレス・オブジェクトに対するメソッド要求
- 非端末関連の START 要求

2 つのルーティング・プログラムは、

1. 別個のシステム初期設定パラメーターで指定されます。動的ルーティング・プログラムの名前は、DTRPGM システム初期設定パラメーターで指定します。分散ルーティング・プログラムの名前は、DSRTPGM システム初期設定パラメーターで指定します。
2. 同じ通信領域が渡されます (一方のプログラムに意味のあるフィールドが、もう一方のプログラムにも意味があるとは限りません)。
3. 同じ時点で呼び出されます。例えば、経路選択の場合は、経路選択エラーが発生したとき、および (必要に応じて) ルーティングされたトランザクションまたはプログラム・リンク要求の終了時に呼び出されます。

これらの 3 つの要因を合わせると、幅広い柔軟性が提供されます。例えば、次のことを行えます。

- 動的ルーティングと分散ルーティングに別々のユーザー作成プログラムを使用する。
- 動的ルーティングと分散ルーティングに同じユーザー作成プログラムを使用する。
- 動的ルーティングにユーザー作成プログラムを使用し、分散ルーティングに CICSPlex SM ルーティング・プログラムを使用する (またはその反対)。

動的ルーティング・プログラムと分散ルーティング・プログラムには、次の 2 つの重要な相違点があることに注意してください。

1. 動的ルーティング・プログラムは、リソース (トランザクションまたはプログラム) が DYNAMIC(YES) と定義されている場合しか呼び出されません。一方、分散ルーティング・プログラムは、関連するトランザクションが DYNAMIC(NO) と定義されている場合でも (非端末関連 START 要求、BTS 活動、およびエンタープライズ Bean や CORBA ステートレス・オブジェクトのメソッド要求が適切であれば) 呼び出されます。ただし、要求をルーティングすることはできません。つまり、分散ルーティング・プログラムは、静的ルーティング要求の影響をターゲット領域の相対ワークロードでモニターすることができます。
2. 動的ルーティング・プログラムは階層型の「ハブ」ルーティング・モデルを使用する (1 つのルーティング・プログラムで、複数のターゲット領域上のリソースへのアクセスを制御する) ので、*ルーティング要求の終了時に呼び出されるルーティング・プログラムは、経路選択の際に呼び出したものと同じプログラムになります。*

   一方、分散ルーティング・プログラムは、ピアツーピア・システムである、分散モデルを使用し、ルーティング・プログラムそのものが分散されます。*ルーティング・トランザクションの開始時または終了時に呼び出されるルーティング・プログラムは、経路選択の際に呼び出したプログラムと同じではありません。*呼び出されるのは、ターゲット領域のルーティング・プログラムです。

重要:

CICS Transaction Server for z/OS バージョン 4 リリース 1 から CICS Transaction Server for OS/390 バージョン 1 リリース 3 領域へルーティングする場合 (あるいはその逆の場合) は、CICS APAR PQ 75814 に対する PTF が CICS Transaction Server for OS/390 バージョン 1 リリース 3 に適用されていることを確認する必要があります。

CICSPlex SM をルーティングに使用する場合、以下のCICSPlex SM APAR のそれぞれに対する PTF が、CICSPlex SM の該当する各リリースに適用されている必要があります。

CICSPlex SM バージョン 1 リリース 4
PQ80891

CICSPlex SM バージョン 2 リリース 2
PQ80893

CICSPlex SM バージョン 2 リリース 3
PQ81235

第 7 章 CICS トランザクション・ルーティング

CICS トランザクション・ルーティングを使用すると、CICS システムに接続されている端末で、別の CICS システム内のトランザクションを実行することができます。

この章には以下のトピックが含まれています。

- 『トランザクション・ルーティングの概要』
- 75 ページの『端末開始トランザクション・ルーティング』
- 79 ページの『ATI によって開始されたトランザクションの従来のルーティング』
- 87 ページの『START コマンドで呼び出されたトランザクションのルーティング』
- 97 ページの『リモート APPC 接続の割り振り』
- 100 ページの『中継プログラム』
- 101 ページの『基本マッピング・サポート (BMS)』
- 102 ページの『ルーティング・トランザクション (CRTE) の使用』
- 103 ページの『トランザクション・ルーティングのためのシステム・プログラミング』.

トランザクション・ルーティングの概要

CICS トランザクション・ルーティングを使用すると、ある CICS システムに接続されている端末で、接続されている別の CICS システム内のトランザクションを実行することができます。CICS システム全体に端末とトランザクションを分散させても、任意の端末で任意のトランザクションを実行することができます。

図 21 は、ある CICS システムに接続されている端末が、別の CICS システムにあるユーザー・トランザクションを実行する様子を示したものです。端末とユーザー・トランザクションの通信は、*中継トランザクション*と呼ばれる、CICS 提供のトランザクションによって処理されます。

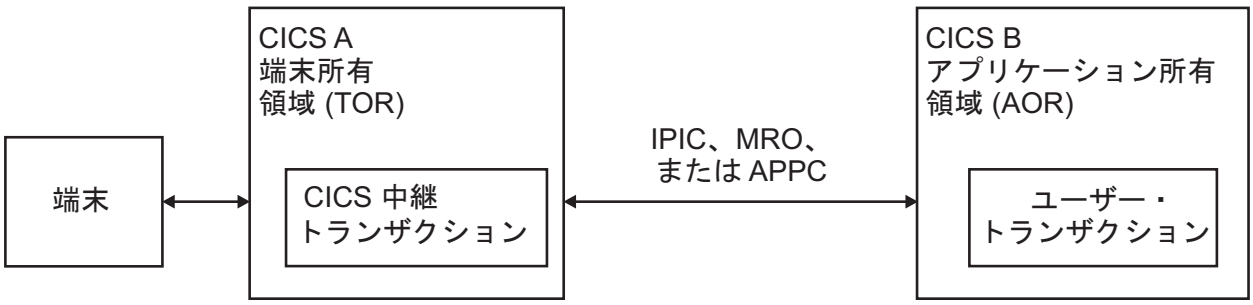


図 21. トランザクション・ルーティングのエレメント

© Copyright IBM Corp. 1977, 2009

端末を所有する CICS システムは、*端末専有領域* または *TOR* と呼ばれ、トランザクションを所有する CICS システムは、*アプリケーション所有領域* または *AOR* と呼ばれます。これらの用語は、あるシステムがすべての端末を所有し、もう一方のシステムがすべてのトランザクションを所有することを意味するものではありません。ただし、このような構成は可能です。

端末専有領域とアプリケーション所有領域は、IPIC、MRO、または APPC リンクで接続する必要があります。LUTYPE6.1 リンクを介したトランザクション・ルーティングはサポートされません。

トランザクション・ルーティングにおける*端末* という用語は、一般に、IBM 3270、単一セッション APPC デバイス、または別の CICS システムとの APPC セッションという意味で使用されます。CICS によってサポートされるすべての端末とセッションのタイプは、トランザクション・ルーティングでの使用に適しています。ただし、次にリストするものは除きます。

- LUTYPE6.1 の接続およびセッション
- MRO の接続とセッション
- EXCI の接続およびセッション
- IBM 7770 または 2260 端末
- プールされた 3600 または 3650 パイプライン論理装置
- MVS システム・コンソール

ユーザー・トランザクションは、CICS の端末管理、BMS、またはバッチ・データ交換の各機能を使用して、端末またはセッション・タイプに応じて適宜、端末と通信を行います。マッピング機能とデータ交換機能は、アプリケーション所有領域で実行されます。BMS ページング操作は、端末専有領域で実行されます

疑似会話型トランザクションがサポートされます。ただし、その「端末」が APPC セッションの場合は除きます。疑似会話型トランザクションを構成する各種のトランザクションは、異なる複数のシステムに置くことができます。

トランザクション・ルーティングの開始

トランザクション・ルーティングは、下記の 3 つの方法で開始することができます。

1. トランザクションの開始要求は、TOR に接続された端末から行うことができます。そのトランザクションのインストール済みリソース定義と、ユーザー作成の動的ルーティング・プログラムによって行われる決定に基づいて、その要求は適切な AOR に送られ、トランザクションは、端末が同じ領域に接続されているかのようにして実行されます。
2. トランザクションは、自動トランザクション開始 (ATI) によって開始することができます。そのトランザクションは、別の CICS システムが所有している端末を獲得することができます。ATI によってルーティング・トランザクションを開始する 2 つの方法については、以下を参照してください。
   - 79 ページの『ATI によって開始されたトランザクションの従来のルーティング』
   - 87 ページの『START コマンドで呼び出されたトランザクションのルーティング』.

3. トランザクションは、ALLOCATE コマンドを出して、別のシステムが所有する APPC 端末や接続とのセッションを獲得することができます。

以上の方法のほかに、CICS には、他のシステムでトランザクションをときおり呼び出すための特別なトランザクション (CRTE) があります。 102 ページの『ルーティング・トランザクション (CRTE) の使用』を参照してください。

端末開始トランザクション・ルーティング

トランザクションの開始要求が CICS TOR に到着した場合、その TOR は、そのトランザクションをどのシステムで実行するかを判別しなければなりません。

この判別は、インストールされているトランザクション定義、とりわけ DYNAMIC オプションと REMOTESYSTEM オプションの値を調べることによって行われます。 244 ページの『トランザクション・ルーティングのトランザクションの定義』を参照してください。

トランザクション・ルーティングは、DYNAMIC オプションの値に応じて、**静的**にも**動的**にもなります。

静的トランザクション・ルーティング

トランザクション定義に DYNAMIC(NO) が指定されていると、静的トランザクション・ルーティングが起こります。

この場合、要求は、REMOTESYSTEM オプションに指定されたシステムにルーティングされます。(REMOTESYSTEM が指定されていないか、そこにローカル CICS システムが指定されていると、そのトランザクションはローカル・トランザクションであるため、トランザクション・ルーティングは行われません。)

動的トランザクション・ルーティング

動的ルーティング・モデル:

端末開始トランザクションの動的ルーティングでは、「ハブ」ルーティング・モデルを使用します (67 ページの『「ハブ」モデル』を参照)。

DYNAMIC(YES) を指定すると、定義されたトランザクションが呼び出されたときに、端末データを代替トランザクションにルーティングすることができます。これを可能にするために CICS は、**動的トランザクション・ルーティング・プログラム**と呼ばれるユーザーで置き換え可能なプログラムによって端末入力データを代行受信し、それを任意のトランザクションやシステムに宛先変更できるようにします。CICS で提供されるデフォルトの動的トランザクション・ルーティング・プログラムは DFHDYP です。提供されるプログラムは、修正することも、独自のもので置き換えることもできます。さらに、独自プログラムの名前を DFHDYP 以外のものにしたければ、DTRPGM システム初期設定パラメーターを使って、動的ルーティングのために呼び出されるプログラムの名前を指定することができます。ユーザーが置き換え可能なプログラムの概要と、DFHDYP の詳細については、「*CICS Customization Guide*」の動的ルーティング・プログラムの作成 (Writing a dynamic

routing program)を参照してください。システム初期設定パラメーターについては、「*CICS System Definition Guide*」の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。

ルーティング・プログラムを呼び出す場合

CICS は、次のような場合に動的ルーティング・プログラムを呼び出します。

- DYNAMIC(YES) として定義されたトランザクションが開始されたとき。

  注:

  1. トランザクション定義が見つからない場合は、CICS は DTRTRAN システム初期設定パラメーターで指定された共通のトランザクション定義を使用します。 249 ページの『TOR での単一トランザクション定義の使用』を参照してください。
  2. トランザクションがターゲット領域およびルーティング領域 (TOR) で DYNAMIC(YES) と定義されている場合には、ルーティングを行うために、ターゲット領域と TOR で動的ルーティング・プログラムが呼び出されます。したがって、ルーティングされた要求を 1 つの領域から別の領域に関連付けることが可能です。誤ってデイジー・チェーンが行われることがないように、注意してください。

  トランザクションが端末から開始された場合、動的ルーティング・プログラムは要求をルーティングすることができます。

  トランザクションが EXEC CICS START コマンドによって開始された場合、ルーティング・プログラムで、要求をルーティングできる場合とできない場合があります。 87 ページの『START コマンドで呼び出されたトランザクションのルーティング』を参照してください。
- 経路の選択でエラーが発生した場合。
- 最初の呼び出しが終了時に再呼び出しを要求した場合、ルーティングされたトランザクションの終わりで。
- 最初の呼び出しが終了時に再呼び出しを要求した場合、ルーティングされたトランザクションが異常終了したとき。
- DPL 要求のルーティングの場合は、 109 ページの『DPL 要求の動的ルーティング』に示されているすべての点で。

ルーティング・プログラムに渡される情報

パラメーターは、CICS と動的ルーティング・プログラムの間で連絡域を使って渡されます。

プログラムは、このパラメーターのうちのいくつかを変更して、後続の CICS のアクションに影響を与えることがあります。これらのパラメーターは次のとおりです。

- 現在の呼び出しの理由。
- エラー情報。
- ターゲット・システムの sysid。初期設定では、インストールされているトランザクション定義の REMOTESYSTEM オプションに指定されている sysid。sysid が指定されていない場合、ローカル・システムの sysid が渡されます。

動的にルーティングされるすべてのリモート・トランザクションに、単一の共通定義を使用します。 249 ページの『TOR での単一トランザクション定義の使用』を参照してください。

- ターゲット・トランザクションの名前。初期設定では、インストールされているトランザクション定義の REMOTENAME オプションに指定されている名前。名前が指定されていない場合は、ローカル名が渡されます。
- 端末入出力域 (TIOA) のデータのコピーをもつバッファーのアドレス。
- ターゲット・システムのネット名。初期設定では、インストールされているトランザクション定義の REMOTESYSTEM オプションに指定されている sysid に対応するネット名です。
- ターゲット・トランザクションの連絡域のアドレス。チャネルとコンテナーを使用している場合で、DFHROUTE コンテナーを定義しているときには、DFHROUTE がアドレスに使用されます。
- ユーザー域。

動的ルーティング・プログラムの使用

動的トランザクション・ルーティングを使用すると、トランザクションへの入力、使用可能な CICS システム、使用可能なシステムの相対負荷などの要因に基づいてトランザクション・ルーティングにおける判断を行うことができます。しかし、ルーティング・プログラムでは、トランザクション要求の再ルーティング以外にもいくつかの機能を行うことができます。

動的ルーティング・プログラムは、次の目的で使用することができます。

- ワークロード・バランシングを実行する。例えば、CICSplex において、このプログラムを使用すれば、並列関係にある AOR の同等のトランザクションからいろいろな条件を考慮して 1 つを選択することができます。
- リモート・システムに使用できるセッションがない場合に、要求をキューに入れるかどうかを指定する。システム間キューの長さの制御については、 307 ページの『第 24 章 システム間のセッション・キューの管理』を参照してください。
- MRO および IPIC リンクの場合、AOR で接続されたトランザクションの優先順位を設定する。
- トランザクションをルーティングできない場合、またはルーティング先のトランザクションが異常終了した場合、ユーザー定義プログラムを実行する。例えば、すべてのリモート CICS 領域が使用不能なため、トランザクションをルーティングできない場合、ローカル端末専有領域でプログラムを実行して、適切なメッセージをユーザーに送信したい場合があります。
- 特定のシステムへルーティングされた要求の数をモニターする。

動的ルーティング・プログラムから EXEC CICS コマンドを出すことはできますが、EXEC CICS RECEIVE コマンドを出しても、ルーティング先のトランザクションは初期端末データを入手できません。

動的トランザクション・ルーティング・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、「*CICS Customization Guide*」の 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。

CICS Interdependency Analyzer

CICS トランザクションは、いろいろな技法を使って情報を相互に受け渡し、相互の間のアクティビティーを同期化します。

技法によっては、データを交換するトランザクションは同じ CICS 領域で実行され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。ワークロード・バランシングの目的で動的トランザクション・ルーティングを使用する場合 (この場合、同等のトランザクションが複数のシステムにある)、ルーティング・プログラムは、相互に依存する (つまり、アフィニティー を持つ) トランザクションを認識していなければなりません。そうすれば、それらのトランザクションを整合性をもってルーティングすることができます。

動的トランザクション・ルーティング環境を作成するとき、その環境に CICS Transaction Server for z/OS バージョン 4 リリース 1 とそれ以前のシステムが混在している場合、CICS Interdependency Analyzer が役立つ場合があります。このユーティリティーを使用すれば、CICS Transaction Server for z/OS および CICS Transaction Server for OS/390 領域において、トランザクション間類縁性の原因を知ることができます。

このユーティリティーについての詳細は、「*CICS Interdependency Analyzer for z/OS ユーザーズ・ガイドおよびリファレンス*」を参照してください。

トランザクションのアフィニティーについての詳細は、「*CICS アプリケーション・プログラミング・ガイド*」のアフィニティーを参照してください。

CICSPlex SM の使用

通常、動的トランザクション・ルーティングを利用するには、動的トランザクション・ルーティング・プログラムを作成する必要があります。

ただし、CICSPlex System Manager (CICSPlex SM) 製品を使用して CICSPlex を管理する場合には、その必要はありません。CICSPlex SM には、ワークロード・バランシ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。しなければならないことは、CICSPlex SM にそのユーザー・インターフェースを使って CICSplex 内のどの TOR と AOR が動的トランザクション・ルーティングで使用できるかを指定し、特定のトランザクションをルーティングする先の AOR を管理する類縁性を定義するだけです。CICS Interdependency Analyzer からの出力は、CICSPlex SM で直接使用することができます。

CICSPlex SM を使用すれば、トランザクションと DPL 要求のワークロード・バランシングを統合することができます。

CICSPlex SM の入門情報については、「*CICSPlex SM Concepts and Planning*」を参照してください。

ATI によって開始されたトランザクションの従来のルーティング

自動トランザクション開始 (ATI) で開始されたトランザクションのルーティングに「従来」の方法を使用するのは、拡張された方法が使用できない場合だけにしてください。

重要:

可能な限り、 87 ページの『START コマンドで呼び出されたトランザクションのルーティング』で紹介する拡張方法を使用するようにしてください。ただし、次をルーティングする場合は、拡張方法は使用できません。

- 一時データ・キュー上でトリガー・レベルで呼び出されたトランザクション。
- IPIC 接続を介して送信されるトランザクション
- EXEC CICS START コマンドによって呼び出された一部のトランザクション。

これらの場合は、「従来」の方法を使用してください。

自動トランザクション開始 (ATI) の処理では、CICS システムやシステム・ネットワークでトランザクション要求が内部的に作成されると、そのトランザクションがスケジュールされます。ATI 要求が発生するのは次の場合です。

EXEC CICS START コマンド
START コマンドでは、指定された時間 (ゼロでもよい) が経過すると、CICS インターバル制御によりトランザクションが開始されます。

一時データ・キュー
キュー上のレコード数が指定されたレベルに達したときにトランザクションが自動的に開始されるように、一時データ・キューを定義することができます。

CICS トランザクション・ルーティングを使用すれば、特定の CICS システムが所有するトランザクションに対する ATI 要求によって、別の接続されたシステムが所有する端末を指定することができます。例えば、 80 ページの図 22 において、AOR1 のアプリケーションが、トランザクション TRAA を端末 PRT1 に接続する START 要求を出すとします。

元の ATI 要求は AOR で起こりますが、その要求は実行のために TOR に送信されます。したがって、この例では、AOR1 が、実行のために START 要求を TOR1 へ送信します。TOR では、ATI 要求の結果、指定された端末 (この例では PRT1) と関連させて、中継プログラムが開始されます。

次に、アプリケーション所有領域のユーザー・トランザクションが、端末開始トランザクション・ルーティングで説明した方法によってアクセスされます。その要求には、リモート・トランザクションの名前 (TRAA) とリモート・システムの名前 (AOR1) を指定する自動開始記述子 (AID) が関連付けられます。

静的トランザクション・ルーティングの場合は、端末専有領域 (TOR1) が、REMOTESYSTEM(AOR1) と REMOTENAME(TRAA) を指定するトランザクション定義を見つけなければなりません。正しい定義が見つからないと、要求は失敗します。

従来の方法を使用した動的トランザクション・ルーティングでは、DYNAMIC(YES) がトランザクション定義に指定されていれば、動的ルーティング・プログラムが呼び出されますが、リモート・システム名は AID から取られるため、その要求を転送することはできません。トランザクション定義の ROUTABLE オプションを使用して拡張ルーティングを指定する方法については、87 ページの『START コマンドで呼び出されたトランザクションのルーティング』を参照してください。

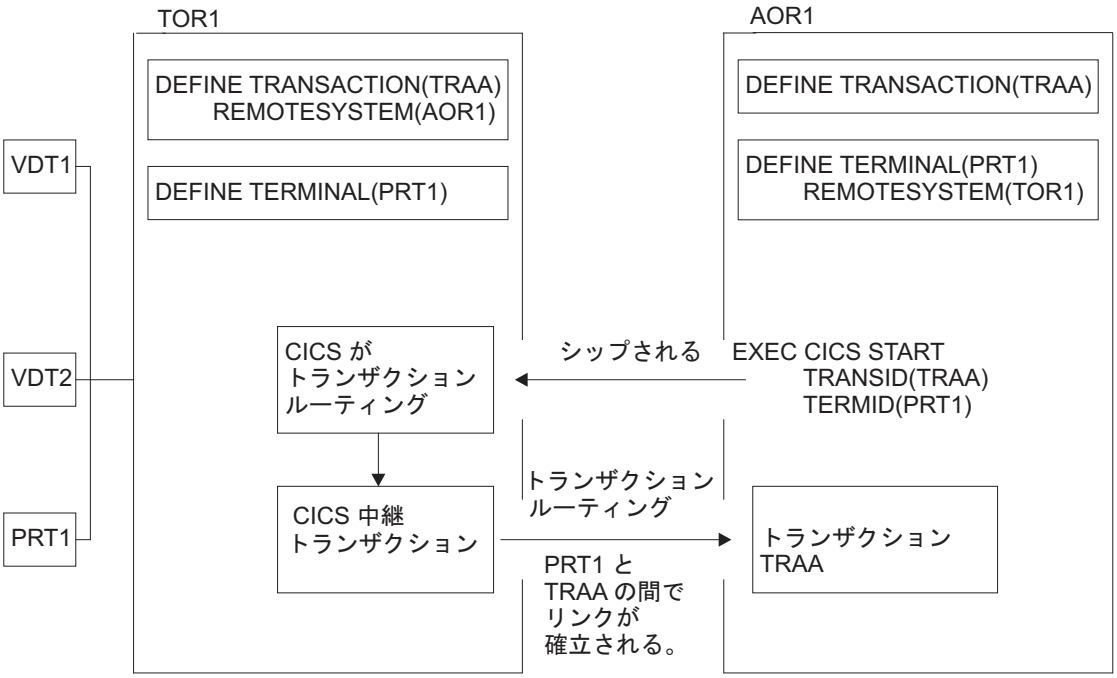


図 22. ATI によって開始されたトランザクション・ルーティング

端末専有領域へのリンクが使用できないと、ATI 要求は、アプリケーション所有領域でキューイングされます。また、そのあとで端末が使用できないと、その要求は端末専有領域でキューイングされます。

アプリケーション所有領域に関する限り、これは、全体的に言えば、ATI の「単一システム」ビューを作成する効果があります。つまり、端末がリモート側にあっても、ATI の表面上の動作には影響ありません。

アプリケーション所有領域では、通常の ATI 規則が適用されます。つまり、トリガー・レベルに達するか、インターバル制御開始要求が満了すると、トランザクションが一時データ・キューから開始されます。特に、一時データによる開始の場合は、一時データ・キューがトランザクションと同じシステムになければならないことに注意してください。トランザクション・ルーティングを使用しても、一時データ・キュー項目から、リモート・トランザクションを開始することはできません。

自動トランザクション開始用端末のシップ

CICS システムの CICA は、ATI 要求を、別の CICS システムである CICB で、次の方法で実行することができます。

以下に例を挙げます。

1. CICA が CICB に START 要求を機能シップする。

2. CICA が、CICB の所有する一時データ・キューに対する WRITEQ 要求を機能シップして、最終的にそのキュー・トリガーが起こる。
3. CICA が CICB 内のトランザクションへのルーティングを起こさせ、そのあと、そのトランザクションが START を出すか、一時データ・キューに書き込む。

ATI 要求に対応する端末がある場合、CICB は、そのリソースを検索して、その端末の定義を探します。端末がリモートであることがわかると、CICB は、端末定義の REMOTESYSTEM オプションに指定されたシステムに ATI 要求を送信します。端末関連の ATI 要求は、TOR で実行されることを忘れないでください。

端末未認識状態

次の例のような場合には、端末関連の START コマンドが端末専有領域で発行され、アプリケーション所有領域に対して機能シップされたのに、端末がまだ定義されていないので、「端末未認識状態」が発生することがよくあります。

重要:

87 ページの『START コマンドで呼び出されたトランザクションのルーティング』で説明する拡張ルーティング方法では、TOR で発行した START コマンドを AOR に対して機能シップしません。したがって、「端末未認識」状態は発生しません。

領域間 ATI を正しく機能させるには、端末を使用する必要があるネットワーク上のすべてのシステムにそれらの端末を定義します。ただし、*自動インストール* を使用している場合は、この処理を行うことができません。自動インストールの使用方法については、「*CICS Resource Definition Guide*」の自動インストールを参照してください。自動インストールされた端末はログオンされるまでシステムに認識されず、また、端末定義を必要とするすべてのシステムにそれらの定義をシップするのは CICS に任されています (236 ページの『端末定義と接続定義のシップ』を参照してください。) この処理は、端末からリモート・システムへのルーティングで行われますが、関連する端末の位置を通知されていないために、システムが ATI 要求を処理できない場合があります。

82 ページの図 23 の例は、この状態を示しています。

1. 端末 T1 のオペレーターが、CICA 上のメニュー・トランザクション M1 を選択します。
2. メニュー・トランザクション M1 が実行され、オペレーターが CICB のトランザクション X1 によって実行される機能を選択します。
3. トランザクション M1 が次のコマンドを出します。

   ```
   EXEC CICS START
        TRANSID(X1)
        TERMID(T1)
   ```

   そして終了します。
4. X1 は CICB によって所有されるリモート・トランザクションとして定義されているので、CICA は START コマンドを CICB に機能シップします。
5. CICB は START コマンドを処理しますが、その際、T1 を所有する領域を見つけようとします。なぜなら、この領域が、START コマンドによって起こる ATI 要求を実行する領域だからです。

6. 以前にルーティングされたトランザクションによって T1 定義が存在する場合のみ、CICB は ATI 要求の送信先を判別することができます。このような定義が存在しないと、インターバル制御プログラムは、START 要求を拒否して、TERMIDERR を出します。

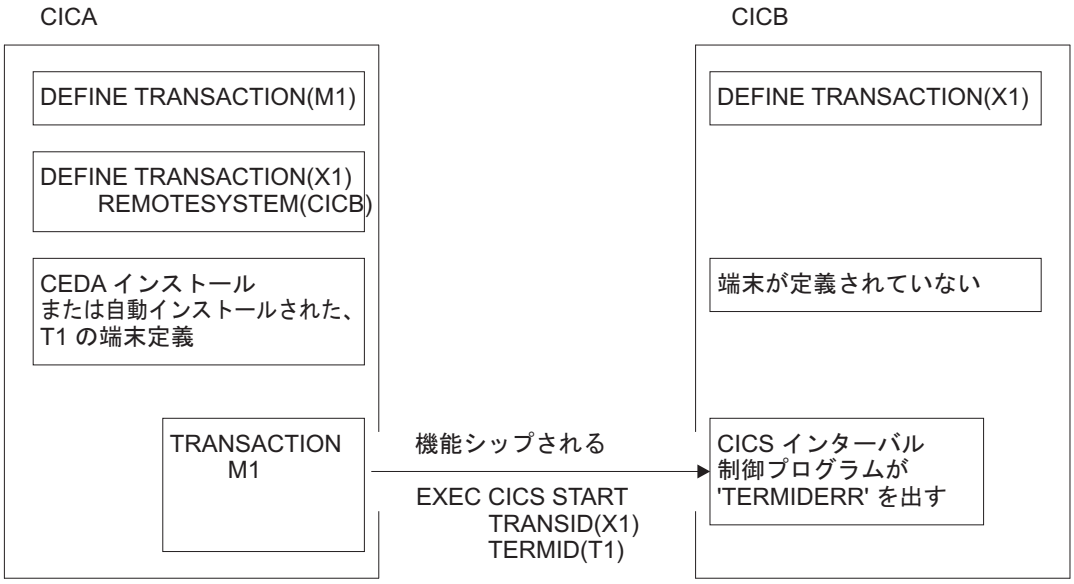


図 23. 端末 ID が認識されていないシステムでの ATI 要求の失敗

グローバル・ユーザー出口 XICTENF および XALTENF:

システムのユーザーは、このルーティングの問題を解決する方法を知っており、CICS には、その解決方法をシステムに伝える機能が用意されています。2 つのグローバル・ユーザー出口 XICTENF および XALTENF がこれにあたります。

XICTENF は、インターバル制御が START コマンドを処理するときに、関連する端末 ID がシステムに未定義であることが分かると実行されます。XALTENF は、やはり 端末 ID が未定義のときに端末割り振りプログラムによって実行されます。

端末割り振りプログラムは、START コマンドの最終的な実行、および一時データ・キュー・トリガー・メカニズムの両方から起こった要求をスケジュールします。つまり、START コマンドが出されると、両方の出口が呼び出される可能性があります。

これらのグローバル・ユーザー出口のいずれか、または両方にサービスを提供するユーザー・プログラムは、次の情報を含むパラメーター・リストにアクセスします。

- ATI 要求が、データをともなう START コマンド、データなしの START コマンド、または一時データ・キュー・トリガーのどれから起こったか。
- START コマンドが、トランザクション・ルーティングの対象であったトランザクションによって出されたかどうか。
- START コマンドが、別の領域から機能シップされたかどうか。
- 実行されるトランザクションの ID。

- トランザクションの実行に使用される端末の ID。
- START コマンドを出したトランザクションがルーティングされたトランザクションの場合は、それに関連する端末の ID。あるいはコマンドが機能シップされた場合は、セッションの ID。それ以外の場合は、ブランクが返されます。
- START 要求を最後にシップしたシステムのネット名。または START がローカルに出された場合は、トランザクション・ルーティングを最後に行ったシステムのネット名。リモート・システムが関与していない場合は、ブランクが返されます。
- 返されたネット名に対応する SYSID。

プログラムを終了する際、CICS に対して、端末が存在するかどうかを通知する必要があります。端末が存在する場合は、TOR のネット名または sysid のいずれかを指定します。CICS は、指定された領域に ATI 要求を送信します。この結果、端末定義が TOR から AOR にシップされ、トランザクション・ルーティングが正常に行われます。

したがって、 82 ページの図 23 の問題は、解決することができます。つまり、小さな出口プログラムを作成し、それによって、CICS 提供のパラメーターを変更せずに戻し、「ネット名が返されたネット名」の戻りコードを設定するだけです。

84 ページの図 24 には、下記のイベントが示されています。

1. インターバル制御プログラムは、START コマンドを受け入れて、必要であれば、発行側システムに対してその受け入れを通知します。
2. 指定の間隔が満了になった後、あるいは間隔が指定されていない場合はただちに、端末割り振りプログラムは、ATI 要求をスケジュールしようとします。定義された端末が見つからないと、出口 XALTENF を実行して、必要なネット名を再提供します。
3. ATI 要求は CICA にシップされます。CICA は中継トランザクションを割り振って、CICB のトランザクション X1 へのトランザクション・ルーティング・リンクを確立し、T1 の端末定義のコピーを CICB にシップします。

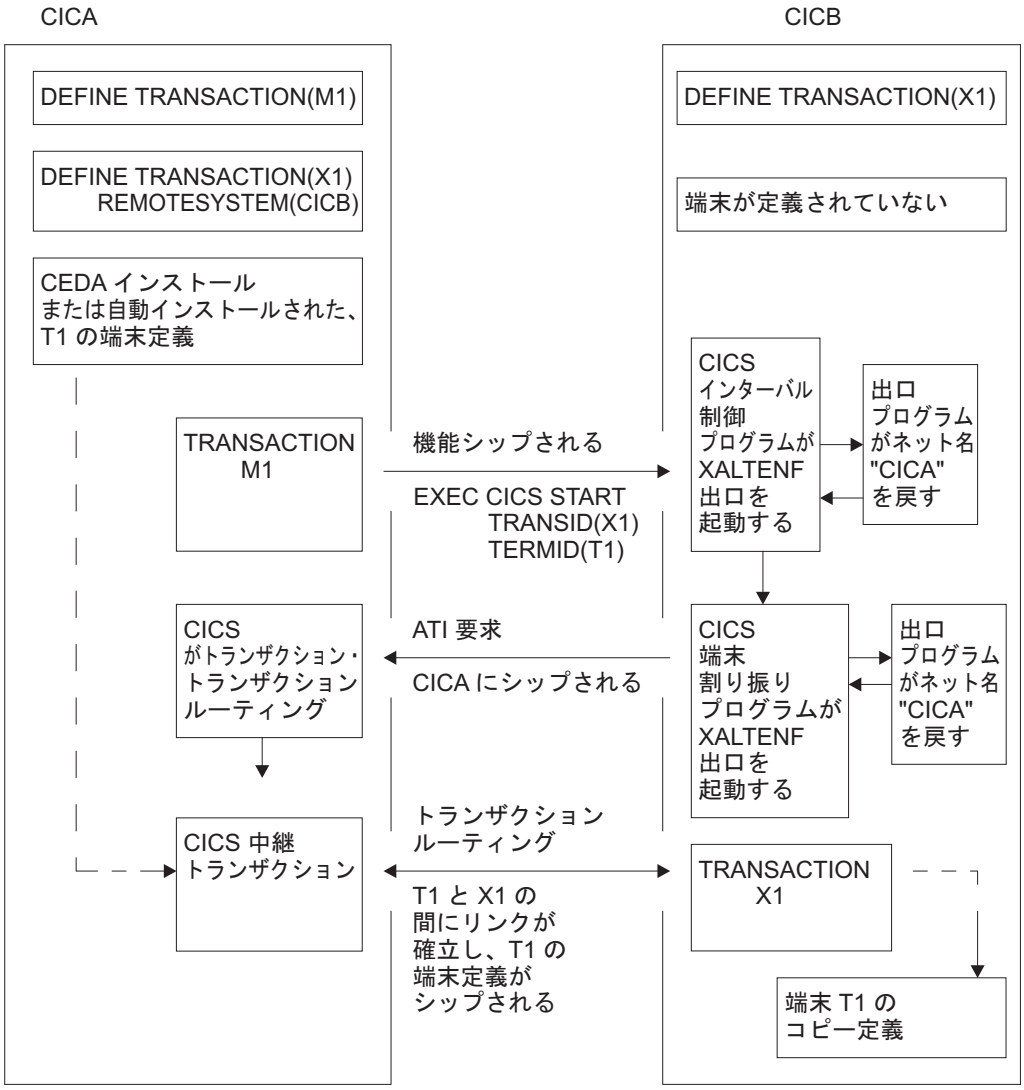


図 24. START 要求における「端末未認識」状態の解決

図 24 の例は可能な構成のうちの 1 つを示していますが、この基本的な例をみれば、複数領域ネットワークで発生する可能性のある、より複雑な状態に対する解決方法を理解することができます。

リソース定義:

出口 XICTENF と XALTENF の使用は、自動インストールされた端末を使用する場合だけではありません。この技法は、CEDA インストール端末が SHIPPABLE(YES) を指定して定義されていれば、これらの端末でも機能します。

ネットワーク操作を行うより前にすべての端末定義が完了している必要はありませんが、システム間のすべてのリンクが完全に定義され、リモート・トランザクションが、それらを使用するシステムに認識されていなければなりません。

注: 再始動の際、グローバル・ユーザー出口プログラムが使用可能にされる前に、CICS 端末割り振りモジュールで「端末未認識」状態が発生することがあります。ここでも介入したい場合には、最初のフェーズの PLTPI プログラムで XALTENF 出

口プログラムを使用可能にする必要があります (PLTPI プログラムに関するプログラミング情報は、
「*CICS Customization Guide*」の 初期化プログラムおよびシャットダウン・プログラムの作成を参照してください。) これは、ウォーム・スタートと緊急スタート両方に適用されます。

重要:

XICTENF と XALTENF 出口は、AOR と TOR の間に直接リンクがある場合のみ使用できます。つまり、出口プログラムから CICS に返す sysid またはネット名は、間接的に接続されたシステムを示すものであってはなりません。

XICTENF と XALTENF 出口の出口プログラム:

出口プログラムが、CICS 提供のパラメーターから TOR を識別する方法は、システム設計を参照しないとわかりません。

最も単純な場合には、CICS に対して、START 要求を出した元のシステムのネット名を返します。これよりも複雑な場合には、各端末に対して、それが常駐するシステムを示す名前を与えることができます。

出口プログラムのプログラミング情報については、「*CICS Customization Guide*」の「端末未認識」状態の出口 XALTENF および XICTENFを参照してください。また、サンプル・プログラムがライブラリー CICSTS41.CICS.SDFHSAMP の DFHXTENF メンバーに用意されています。

複数 TOR の ATI 用端末のシップ

次のネットワークが設定されているとします。

1. 2 つ以上の端末専有領域 (TOR) に接続されたアプリケーション所有領域があり、それらの端末専有領域は同じか類似した端末 ID 群を使用します。
2. それらの TOR の 1 つまたはそれ以上から、AOR のトランザクションに対する EXEC CICS START 要求が出されます。
3. それらの START 要求はそれぞれ端末に対応付けられます。
4. AOR では、リモート端末を静的に定義する代わりに、シップ可能な端末が使用されます。

次のシナリオを考えてみてください。

端末専有領域 TORB がトランザクション TRANB (領域 AOR1 によって所有される) に対する EXEC CICS START 要求を出します。これは、端末 T1 に対して実行されます。一方、領域 TORA の端末 T1 が既に AOR1 にトランザクション・ルーティングしており、T1 の定義が既に TORA から AOR1 へシップされています。その START 要求は、AOR1 に到達すると、端末 T1 からのトランザクション・ルーティングに対し、TORB ではなく TORA へシップされます。

86 ページの図 25 にこの様子を示します。

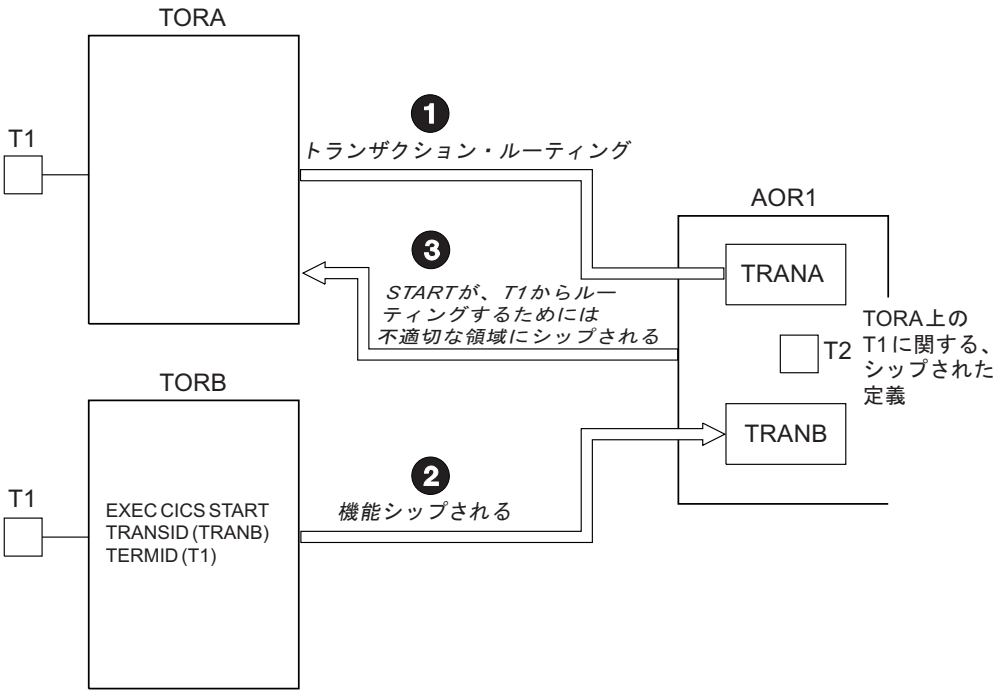


図 25. 正しくない端末に対して開始される機能シップされた START 要求: 端末 T1 (TORA によって所有される) のシップされた定義は AOR1 にインストールされていますので、TORB から受信された START 要求は、ルーティングに対し、TORB ではなく TORA へシップされます。

この状態を避ける方法は 2 つあります。

1. **こちらの方法を使用することをお勧めします。**

   87 ページの『START コマンドで呼び出されたトランザクションのルーティング』に記載されている拡張ルーティング方法を使用します。端末専有領域で発行された端末関連の START コマンドは、AOR に機能シップ されません。したがって、間違った TOR にシップされることはありません。代わりに、START は TOR で直接実行され、トランザクションは端末から開始されたかのようにルーティングされます。

   端末の定義は AOR へシップされ、自動インストール・ユーザー・プログラムが呼び出されます。自動インストール・ユーザー・プログラムは、AOR で*別名* 端末 ID を割り振ることにより、既にインストールされているリモート定義との矛盾を避けることができます。端末の別名については、 243 ページの『端末の別名』を参照してください。シップされた定義のインストールを制御する自動インストール・プログラムの作成については、「*CICS Customization Guide*」を参照してください。

2. 拡張ルーティング方法を使用できない場合には、こちらの方法を使用してください。

   AOR で FSSTAFF システム初期設定パラメーターに YES を指定します。こうすれば、START 要求を端末専有領域から受信したときに、その要求に指定されている端末のシップされた定義が AOR に既にインストールされていると、その要求は、リモート端末定義で参照されている TOR が何であれ、ルーティングに対し、*その要求が受信されたリンクを使って* 必ずその TOR へ戻されます。(この場合の唯一の例外は、START 要求が TOR_NETNAME を提供し、正しい

TOR_NETNAME が指定されているリモート端末が見つかった場合です。その場合には、要求は適切な TOR にシップされます。)

その START 要求が戻された TOR が、インストール済みのリモート端末定義で参照されるものと**異なる**場合には、その端末の定義が AOR へシップされ、自動インストール・ユーザー・プログラムが呼び出されます。自動インストール・ユーザー・プログラムは、AOR で別名端末 ID を割り振ることにより、既にインストールされているリモート定義との矛盾を避けることができます。

FSSTAFF システム初期設定パラメーターの詳細については、「*CICS System Definition Guide*」を参照してください。

ATI および総称リソース

AOR は、VTAM 総称リソースが所有する端末に対して EXEC CICS START 要求を出すことができます。その場合、その端末が現在ログオンされている総称リソース・グループのメンバーを知っている必要はありません。

ATI を総称リソースに対して使用する詳しい方法については、154 ページの『ATI での総称リソースの使用』を参照してください。

START コマンドで呼び出されたトランザクションのルーティング

EXEC CICS START コマンドによって呼び出されたトランザクションのルーティングに「拡張」方法を使用するよう、トランザクションを要求領域 (START コマンドが発行される領域) で ROUTABLE(YES) と定義します。

「拡張」方法は、79 ページの『ATI によって開始されたトランザクションの従来のルーティング』で記述されている「従来の」方法に優先して使用されます。ただし、拡張方法では、次に挙げるいくつかのトランザクションをルーティングできないので、注意してください。

- 一時データ・キュー上でトリガー・レベルで呼び出されたトランザクション。
- IPIC 接続を介して送信されるトランザクション
- EXEC CICS START コマンドによって呼び出された一部のトランザクション。

これらの場合は、「従来の方法」を使用してください。

拡張方法の利点

次に、拡張方法が「従来」の方法よりも優れている点をいくつか挙げます。

動的ルーティング

「従来」の方法では、開始されたトランザクションを動的にルーティングすることができません例えば、端末専有領域において、トランザクションが端末関連の START コマンドで DYNAMIC(YES) と定義されている場合、動的ルーティング・プログラムは通知の場合にのみ呼び出されます。このトランザクションをルーティングすることはできません。

拡張方法を使用すれば、開始されたトランザクションを動的にルーティングすることができます。

効率

「従来」の方法では、TOR で発行された端末関連の START コマンドは、トランザクションを所有する AOR に機能シップされます。要求は、TOR からルーティングするために、再度シップされます。

拡張方法では、AOR へのシップと TOR への再シップの 2 つのホップが省かれます。TOR で発行された START コマンドは TOR で直接実行され、トランザクションは遅延なくルーティングされます。

単純さ

「従来」の方法では、TOR で発行された端末関連の START コマンドがトランザクションを所有する AOR に機能シップされたときに、端末が AOR で定義されていないと「端末未認識」状態が発生することがあります。

拡張方法では、TOR で発行された START コマンドは AOR に機能シップされないので、「端末未認識」状態は発生しません。START は TOR で直接実行され、トランザクションは端末から開始されたかのようにルーティングされます。端末が AOR で定義されていない場合は、TOR から定義がシップされます。

端末関連 START コマンドによって開始されたトランザクションのルーティング方法

端末関連 START コマンドには複数のオプションを設定することができます。このオプション設定は、トランザクションのルーティング先である一連の領域に影響を与えることがあります。

IPIC 接続での拡張ルーティングはサポートされていません。IPIC 接続で拡張ルーティングを使用しようとしても、トランザクションのルーティングには従来のルーティング方法が使用されます。

端末関連 START コマンドによって開始されたトランザクションで拡張ルーティング方式を使用するには、次の条件をすべて満たす必要があります。

- START コマンドが、適格な START コマンドのサブセットのメンバーである。すなわち、次の条件をすべて満たしている。
  - START コマンドの TERMID オプションで、現在のタスクに関連している端末を指定している。
  - START コマンドを発行するタスクの基本機能が端末である。例えば、START コマンドを発行するプログラムに DPL リンクがある場合、基本機能は端末ではありません。この場合の基本機能はシステム間のセッションになります。
  - START コマンドを発行するタスクの基本機能が、代理クライアントの仮想端末ではない。
  - START コマンドの SYSID オプションで、リモート領域の名前が指定されていない。つまり、トランザクションが開始されるリモート領域が明示的に指定されていない。
  - 開始されるトランザクションの TRANSACTION 定義では、REMOTESYSTEM オプションは指定されない。トランザクションが開始されるリモート領域が明示的に指定されていない。

要求側領域と TOR は同じ領域であっても構いません。

- 要求側領域と TOR が異なる場合、これらが次のいずれかのリンクで接続されている。
  - MRO リンク
  - APPC 並列セッション・リンク
- TOR とターゲット領域が次のいずれかのリンクで接続されている。
  - MRO リンク
  - APPC 単一または並列セッション・リンク。APPC リンクを使用している場合は、少なくとも次のうちの 1 つを満たしていなければならない。
    1. 端末開始トランザクションのルーティングが、リンクを介して既に行われている
    2. CICSPlex SM がルーティングで使用されている。
- *要求側* 領域のトランザクション定義では、ROUTABLE(YES) が指定されている。
- トランザクションを動的にルーティングする場合は、TOR のトランザクション定義で DYNAMIC(YES) が指定されている。

  重要: 動的ルーティングの際に候補とする START 開始トランザクションを考慮する場合には、START コマンドで次のオプションが指定されているかどうかに特に注意してください。
  - AT、AFTER、INTERVAL、または TIME。つまり、Start の実行前に遅延がある。
  - QUEUE
  - REQID
  - RTERMID
  - RTRANID

AOR で発行された START コマンド

端末関連の START コマンドが AOR で発行された場合、そのコマンドは、TERMID オプションで指定された端末を所有する TOR にシップされます。START は TOR で実行されます。

AOR で出されたコマンドの静的ルーティング:

静的ルーティングは、アプリケーション所有領域 (AOR) のトランザクション定義で ROUTABLE(YES) が指定されており、端末専有領域 (TOR) のトランザクション定義で DYNAMIC(NO) が指定されている場合に実行されます。このため、動的ルーティング・プログラムは呼び出されません。

トランザクションが拡張ルーティング可能なものである場合、そのトランザクションは、TOR のトランザクション定義の REMOTESYSTEM オプションで指定された AOR にルーティングされます。REMOTESYSTEM が指定されていない場合、トランザクションは TOR 内でローカルに実行されます。

トランザクションが拡張ルーティング可能なものでない場合、トランザクションは通常の方法で処理されます (79 ページの『ATI によって開始されたトランザクションの従来のルーティング』を参照)。つまり、CICS は、要求元の AOR で実行するよう、その AOR にトランザクションをルーティングし直します。TOR におけるト

ランザクション定義の REMOTESYSTEM オプションで、要求元の AOR 以外の領域が指定されている場合、要求は失敗します。

図 26 に、拡張方法を使用して、AOR で発行された端末関連の START コマンドで開始されたトランザクションを静的にルーティングする場合の要件を示します。

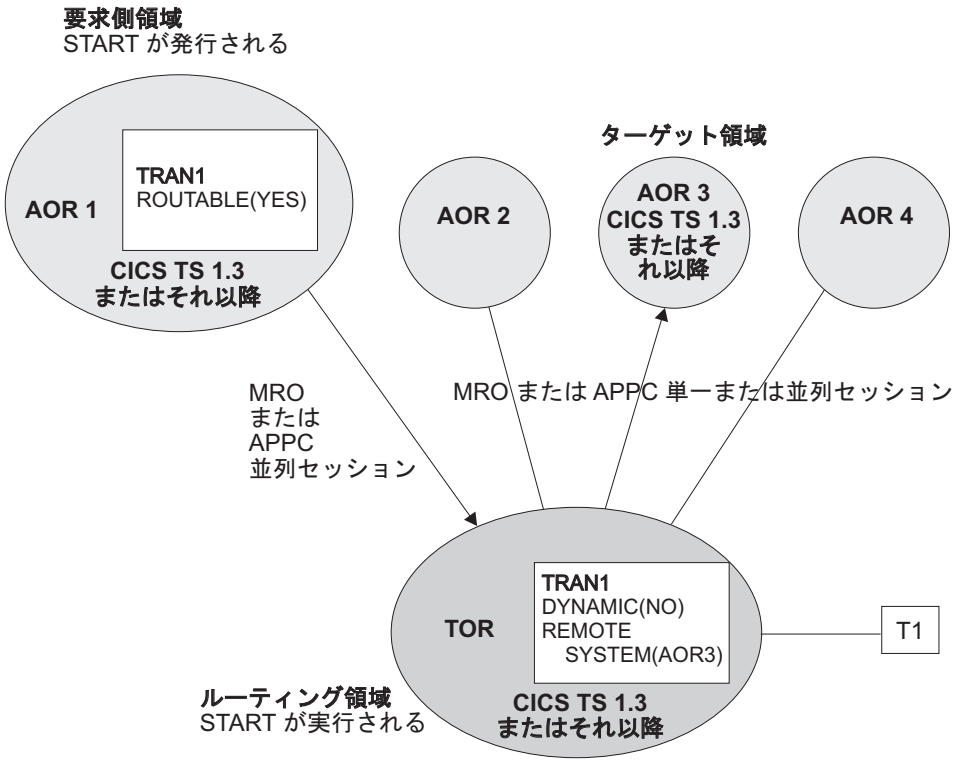


図 26. 拡張方法を使用した、AOR で発行された端末関連 START コマンドの静的ルーティング

要求側領域と TOR は、MRO または APPC 並列セッション・リンクで接続されています。 TOR とターゲット領域は、MRO または APPC (単一または並列セッション) リンクで接続されています。要求側領域のトランザクション定義では、ROUTABLE(YES) が指定されています。TOR のトランザクション定義では、DYNAMIC(NO) が指定されています。 REMOTESYSTEM オプションでは、トランザクションのルーティング先の AOR が指定されています。

AOR で出されたコマンドの動的ルーティング:

動的ルーティングは、アプリケーション所有領域 (AOR) のトランザクション定義で ROUTABLE(YES) が指定されており、端末専有領域 (TOR) のトランザクション定義で DYNAMIC(YES) が指定されている場合に実行されます。このため、動的ルーティング・プログラムは TOR で呼び出されます。

端末関連 START コマンドで呼び出されたトランザクションの動的ルーティングでは、「ハブ」ルーティング・モデルを使用します (67 ページの『「ハブ」モデル』を参照)。

トランザクションが拡張ルーティング可能である場合、ルーティング・プログラムは、そのトランザクションを代替 AOR、すなわち、START が発行された AOR 以外の AOR に転送することができます。

トランザクションが拡張ルーティング可能でない場合、動的ルーティング・プログラムは通知の場合にのみ呼び出されます。トランザクションを転送することはできません。トランザクションは通常の方法で処理されます。つまり、トランザクションは要求元の AOR にルーティングされ実行されます。

図 27 に、AOR で発行された端末関連 START コマンドで開始されたトランザクションを動的ルーティングする場合の要件を示します。

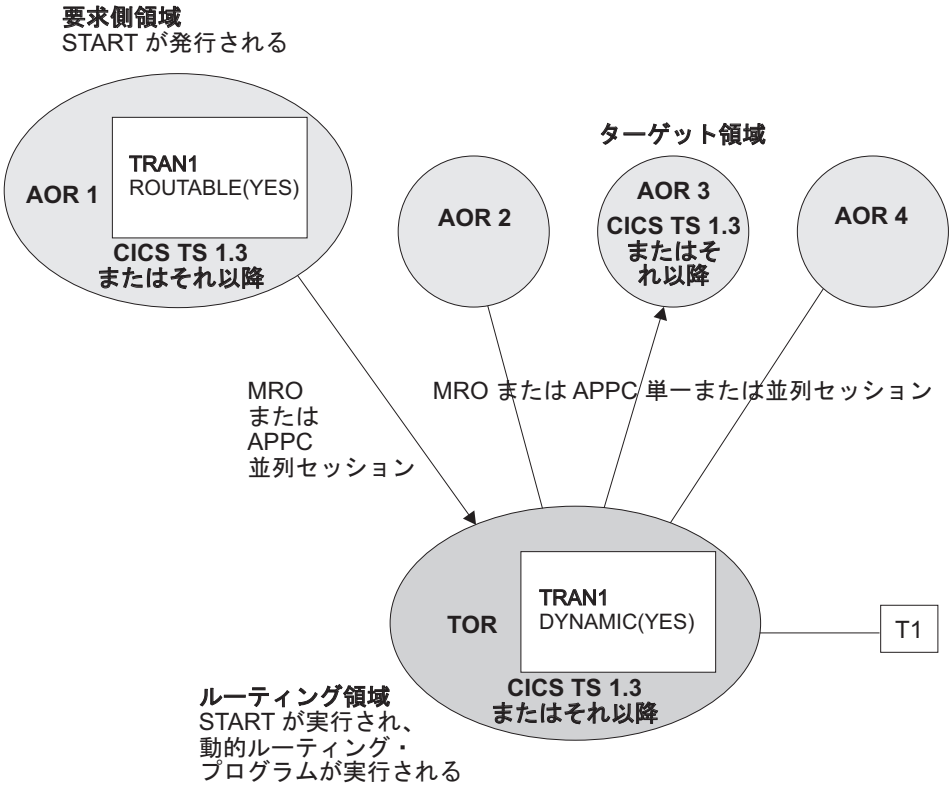


図 27. AOR で発行された端末関連 START コマンドの動的ルーティング

要求側領域と TOR は、MRO または APPC 並列セッション・リンクで接続されています。 TOR とターゲット領域は、MRO または APPC (単一または並列セッション) リンクで接続されています。要求側領域のトランザクション定義では、ROUTABLE(YES) が指定されています。TOR のトランザクション定義では、DYNAMIC(YES) が指定されています。

TOR で発行された START コマンド

TOR で出された端末関連 START コマンドは、静的または動的にルーティングされます。

端末関連 START コマンドの静的ルーティング:

端末専有領域で、静的にルーティングされたトランザクションのトランザクション定義で ROUTABLE(YES) と DYNAMIC(NO) が指定されているため、動的ルーティング・プログラムは呼び出されません。

トランザクションが拡張ルーティング可能な場合には、次のステップが実行されます。

1. START コマンドが TOR で実行されます。
2. トランザクションは、トランザクション定義の REMOTESYSTEM オプションで指定されている AOR にルーティングされます。REMOTESYSTEM が指定されていない場合は、トランザクションは TOR 内でローカルに実行されます。

トランザクションが拡張ルーティング可能でない場合、START 要求は通常の方法で処理されます (79 ページの『ATI によって開始されたトランザクションの従来のルーティング』を参照)。つまり、トランザクション定義の REMOTESYSTEM オプションで指定された AOR に機能シップされます。REMOTESYSTEM が指定されていない場合、START 要求は TOR 内でローカルに実行されます。

図 28 に、拡張方法を使用して、TOR で発行された端末関連 START コマンドによって開始されたトランザクションを静的にルーティングする場合の要件を示します。

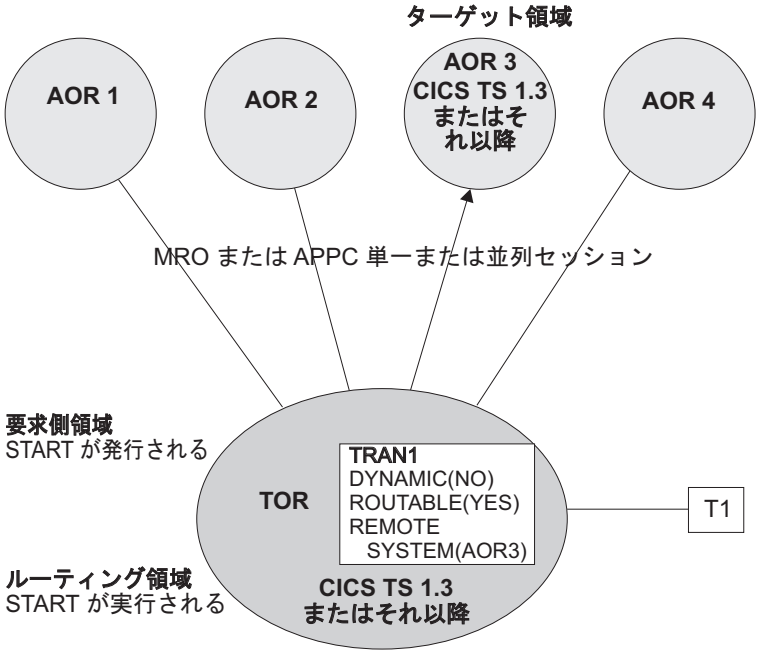


図 28. 拡張方法による、TOR で発行された端末関連 START コマンドの静的ルーティング

TOR とターゲット領域は、MRO または APPC (単一または並列セッション) リンクで接続されています。TOR のトランザクション定義では、DYNAMIC(NO) と ROUTABLE(YES) が指定されています。REMOTESYSTEM オプションでは、トランザクションのルーティング先の AOR が指定されています。

関連概念

88 ページの『端末関連 START コマンドによって開始されたトランザクションのルーティング方法』

端末関連 START コマンドには複数のオプションを設定することができます。このオプション設定は、トランザクションのルーティング先である一連の領域に影響を与えることがあります。

端末関連 START コマンドの動的ルーティング:

端末専有領域で、動的にルーティングされたトランザクションのトランザクション定義で ROUTABLE(YES) と DYNAMIC(YES) が指定されているため、動的ルーティング・プログラムは呼び出されません。

端末関連 START コマンドで開始されたトランザクションの動的ルーティングでは、ハブ・ルーティング・モデルを使用します。

トランザクションが拡張ルーティング可能な場合には、次のステップが実行されます。

1. START コマンドが TOR で実行されます。
2. ルーティング・プログラムでトランザクションをルーティングすることができます。

トランザクションが拡張ルーティング可能でない場合、動的ルーティング・プログラムは、トランザクションをルーティングできないため、通知の場合にのみ開始されます。START 要求は通常の方法で処理されます。つまり、TOR におけるトランザクション定義の REMOTESYSTEM オプションで指定された AOR に機能シップされます。REMOTESYSTEM が指定されていない場合、START 要求は TOR 内でローカルに実行されます。

図 29 に、TOR で発行された端末関連 START コマンドで開始されたトランザクションを動的にルーティングする場合の要件を示します。

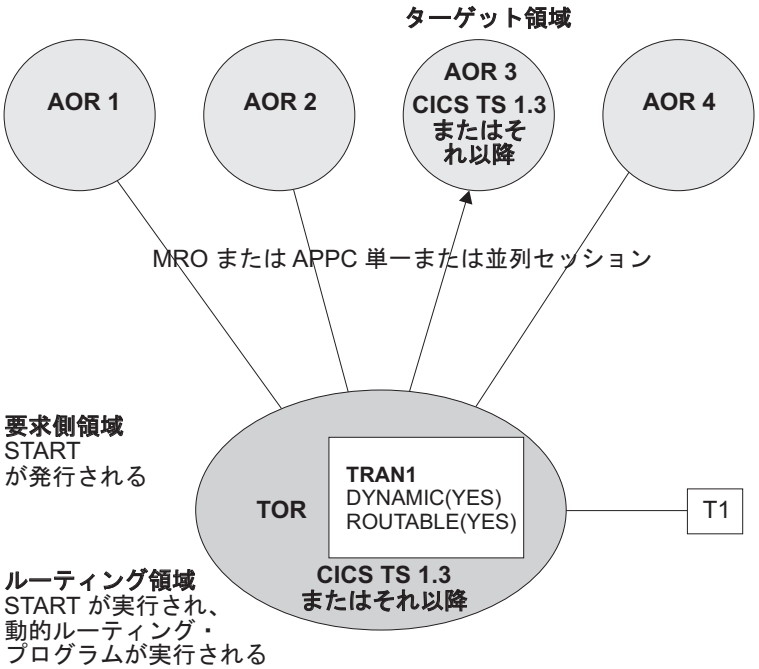


図 29. TOR で発行された端末関連 START コマンドの動的ルーティング

TOR とターゲット領域は、MRO または APPC (単一または並列セッション) リンクで接続されています。TOR のトランザクション定義では、ROUTABLE(YES) と DYNAMIC(YES) が指定されています。

関連概念

67 ページの『「ハブ」モデル』
「ハブ」は、従来は CICS 動的トランザクション・ルーティングで使用されていたモデルです。

非端末関連の START コマンド

非端末関連 START 要求が拡張ルーティング可能であるためには、次の条件をすべて 満たしていなければなりません。

- 要求側領域とターゲット領域が次のいずれかで接続されている。
  - MRO リンク
  - APPC 単一または並列セッション・リンク。APPC リンクを使用しており、分散ルーティング・プログラムをターゲット領域で起動する場合は、ルーティングに CICSPlex SM を使用する必要がある。
- 要求側領域のトランザクション定義では、ROUTABLE(YES) が指定されている。

さらに、要求を動的にルーティングする場合には、次の条件が適用されます。

- 要求側領域のトランザクション定義で、DYNAMIC(YES) を指定しなければならない。
- START コマンドの SYSID オプションで、リモート領域の名前を指定してはならない(つまり、トランザクションを開始するリモート領域が明示的に指定されていてはならない)。

注: 動的ルーティングの際に候補とする START 開始要求を考慮する場合には、START コマンドで次のオプションが指定されているかどうかに特に注意してください。

- AT、AFTER、INTERVAL (非ゼロ)、または TIME。つまり、START の実行前に遅延がある。

  遅延が発生した場合には、START 要求で作成されたインターバル制御エレメント (ICE) が、CDFS のトランザクション ID と共に要求側領域に保持されます。CDFS トランザクションは、ユーザーが指定したデータを取得して、インターバルなしで START 要求を再発行します。ICE が満了すると、要求は、その時点でのトランザクション定義の状態とシスプレックスに基づいて、ルーティングされます。
- QUEUE
- REQID
- RTERMID
- RTRANID

これらの各オプションをどのように使用しているかを認識しておく必要があります。例えば、要求のルーティング先となる領域のセットに影響するかどうかなどです。

静的ルーティング

要求側領域のトランザクション定義では、ROUTABLE(YES) と DYNAMIC(NO) が指定されています。START 要求が拡張ルーティング可能である場合、分散ルーティング・プログラム (DSRTPGM システム初期設定パラメーターで指定されたプログラム) は、静的ルーティングされた要求の通知の場合に呼び出されます。

注:

1. 分散ルーティング・プログラムは、呼び出し方法の点で動的ルーティング・プログラムと異なります。分散ルーティング・プログラムは、トランザクションが DYNAMIC(NO) と定義されている場合でも、トランザクションが ROUTABLE(YES) と定義されている非端末関連 START 要求が適格であれば、呼び出されます。一方、動的ルーティング・プログラムは、トランザクションが DYNAMIC(NO) と定義されている場合には絶対に呼び出されません。つまり、分散ルーティング・プログラムを使用すれば、ワークロード全体に対する静的ルーティングされた要求の影響を評価することができます。
2. 要求が拡張ルーティング可能でない場合には、分散ルーティング・プログラムは呼び出されません。

動的ルーティング

動的ルーティング・モデル:

非端末関連 START 要求の動的ルーティングでは、分散ルーティング・モデル (68 ページの『分散モデル』を参照) を使用します。

要求側領域のトランザクション定義では、ROUTABLE(YES) と DYNAMIC(YES) が指定されています。要求が拡張ルーティング可能な場合には、分散ルーティング・プログラムは呼び出されます。START 要求は、ルーティング・プログラムから戻されたターゲット領域に機能シップされます。

注:

1. 要求が拡張ルーティング可能でない場合には、分散ルーティング・プログラムは呼び出されません。SYSID オプションでリモート領域を明示的に指定していない限り、START 要求は、要求側領域におけるトランザクション定義の REMOTESYSTEM オプションで指定された AOR に機能シップされます。REMOTESYSTEM が指定されていない場合は、START は、要求側領域内で、ローカルに実行されます。
2. 要求が拡張ルーティング可能である一方で、START コマンドの SYSID オプションでリモート領域が指定されている場合、分散ルーティング・プログラムは通知のときにのみ呼び出されます。要求をルーティングすることはできません。START は、SYSID オプションで指定されたリモート領域で実行されます。
3. CICS Transaction Server for z/OS バージョン 4 リリース 1 から CICS Transaction Server for OS/390 バージョン 1 リリース 3 領域へルーティングする場合 (あるいはその逆の場合) は、CICS APAR PQ 75814 に対する PTF が CICS Transaction Server for OS/390 バージョン 1 リリース 3 に適用されていることを確認する必要があります。

CICSPlex SM をルーティングに使用する場合、以下のCICSPlex SM APAR のそれぞれに対する PTF が、CICSPlex SM の該当する各リリースに適用されている必要があります。

CICSPlex SM バージョン 1 リリース 4
PQ80891

CICSPlex SM バージョン 2 リリース 2
PQ80893

CICSPlex SM バージョン 2 リリース 3
PQ81235

インターバル制御要求の取り消し:

前に発行した START、DELAY、または POST インターバル制御要求を取り消すには、CANCEL コマンドを使用します。

このタスクについて

REQID オプションで、取り消したい要求の ID を指定します。要求がリモート領域上での実行によるものである場合には、SYSID オプションを使用して、CANCEL コマンドをその領域にシップすることを指定することができます。

START 要求と DELAY 要求を取り消すことができるのは、要求で指定されたインターバルが満了するまでの間です。START 要求は、動的にルーティングされた場合には、インターバルが満了するまでローカル領域に保持されます。したがって、SYSID オプションが必要ないローカル発行の CANCEL コマンドによって取り消すことができます。ただし、分散ルーティング環境 (各領域を要求側領域とターゲット領域の両方に設定できる環境) では、どの領域が CANCEL コマンドを送信しているのか分からないことがあります。例えば、有効な領域のセットのうちの 1 つで発行された DELAY 要求を取り消したいとします。このような状況を解決するには、次のようにします。

1. REQID オプションで取り消したい要求の ID を指定し、SYSID オプションを指定していない CANCEL コマンドを発行します。コマンドはローカルで実行されます。
2. CICS 提供のサンプル・プログラム DFH$ICCN に基づいて、XICEREQ グローバル・ユーザー出口プログラムを使用します。出口プログラムは、CANCEL コマンドの実行前に呼び出されます。DFH$ICCN:
   a. 次の点をチェックします。
      1) CANCEL コマンドに対して呼び出されたこと。
      2) このコマンドで SYSID オプションが指定されていないこと。
      3) 取り消したい要求の ID の先頭が「DF」でないこと (「DF」は、CICS 内部で発行された要求を示します)。
      4) CANCEL コマンドを発行したトランザクションの名前の先頭が「C」でないこと。つまり、そのトランザクションが CICS の内部トランザクションでも、CICS 提供トランザクション (CECI など) でもないこと。

      これらの条件の 1 つ以上を満たしていない場合 (例えば、RETRIEVE コマンドに対して呼び出した場合など) には、DFH$ICCN は何も行わずに戻ります。

b. CICSPlex SM に対し、次のことを行うよう指示します。

1) 各 CICS 領域で、CANCEL コマンドで指定した ID (REQID) が付いているインターバル制御要求に関する情報を検索する。
2) 各領域で、探し出した最初の要求 (指定された ID を持つもの) を取り消す。次の点に注意してください。
   - 要求は、複数の領域で取り消される場合があります。
   - 特定の領域に、指定された ID を持つ要求が複数入っている場合、取り消されるのは、CICSPlex SM が探し出した最初の要求だけです。
   - CICSPlex SM が、CANCEL 要求と関連付けられたトランザクションのトランザクション ID に UPDATE アクセスできることを確認してください。

注: DFH$ICCN の処理については、サンプル・プログラムのコメントを参照してください。

CANCEL コマンドの詳細については、「*CICS アプリケーション・プログラミング・リファレンス*」の CANCELを参照してください。XICEREQ グローバル・ユーザー出口プログラムの作成方法の概要については、「*CICS Customization Guide*」のインターバル制御機能の EXEC インターフェース・プログラムの出口を参照してください。

リモート APPC 接続の割り振り

アプリケーション所有領域で実行されるトランザクションは、ALLOCATE コマンドを出すことによって、別のシステムが所有する APPC 端末または接続に対するセッションを得ることができます。

トランザクションとリモート APPC システムまたは端末との間で要求をやり取りするために、中継プログラムが端末専有領域で開始されます。

APPC デバイスを使用したトランザクション・ルーティング

APPC デバイスは、APPC アーキテクチャーの実装である CICS にデータ・インターフェースを提供します。このデバイスをトランザクションにリンクする APPC セッションは、デバイス自体というよりトランザクションの基本機能を表すものです。トランザクションは、このリンクを介して、デバイス内のトランザクション・プログラムと会話します。このデバイスは、ハードコーディングされた端末デバイス、プログラム式システム、または他の CICS システムのいずれでも可能です。

APPC デバイスによるトランザクション・ルーティングと、他の端末によるトランザクション・ルーティングの間に実質的な違いはありません。ただし、次の点に注意する必要があります。

- APPC デバイスには、独自の「知能」があります。この知能は、オペレーターの入力データ、または CICS から受信したデータを、設計者が選択した方法で解釈することができます。
- CICS からのエラー・メッセージはありません。APPC デバイスは CICS から指示を受信しますが、これは、オペレーターのためにテキストに変換することができます。

- CICS は APPC デバイスでの疑似会話型操作を直接にはサポートしませんが、デバイスそのものを同じ効果をもつようにプログラミングできる場合があります。
- 基本マッピング・サポート (BMS) は、APPC デバイスでは無意味です。
- APPC デバイスは、複数のセッションによって、ホスト・システムにリンクすることができます。
- APPC 単一セッション端末の場合は TCTUA がその接続でシップされますが、その基本機能が APPC 並列セッションの場合はシップされません。

APPC デバイスと通信を行うには、APPC アプリケーション・プログラム・インターフェースを使用します。関連する入門情報については、 117 ページの『第 9 章 分散トランザクション処理』を参照してください。

代替機能の割り振り

トランザクション・ルーティングにおける設計基準の 1 つは、単一 CICS 環境で実行されるトランザクションが、リンクされた代替システムに転送された場合、そのトランザクションを元の端末にルーティングすることが必要になったときに機能の欠落があってはならないということです。

APPC デバイスでは複数のセッションをもつことができるので、単一 CICS において、トランザクションが ALLOCATE コマンドによって同じデバイス (ただし異なるタスクへの) に対するセッションをさらに獲得することが可能です。このようにして獲得された各セッションは、そのトランザクションにとって**代替機能**となります。さらに、セッションは、他の端末やシステムに対して確立することもできます。

同じように、トランザクション・ルーティングを使用することによって、その APPC デバイスと AOR の間に中間システムがあっても、トランザクションは ALLOCATE を使用して APPC デバイスの代替機能を獲得することができます。このためには、TOR にインストールされている APPC リンク定義のリモート・バージョンが AOR に必要です。おそらく、これは、トランザクション・ルーティングの操作によって AOR にシップされているはずです。そうでない場合は、このバージョンを明示的にインストールする必要があります。ユーザー出口の XICTENF と XALTENF を使用して、代替機能をルーティングすることはできません。

端末としてのシステム

APPC デバイスのリソース定義は、CONNECTION および SESSIONS 形式をとる場合があるため、これらをシステム間リンクの定義と混同することがよくあります。

システム間リンクの定義は**直接**か**間接**のどちらかであるのに対し、APPC デバイスの定義は、TOR では**直接**であり、AOR と中間システムでは**リモート**であることに注意してください。また、リモート CONNECTION 定義は、対応する SESSIONS 定義を必要としません。

99 ページの図 30 は、チェーニングされた 3 つの CICS システムによるネットワークを示しています。最初のシステムは APPC 端末にリンクされています。

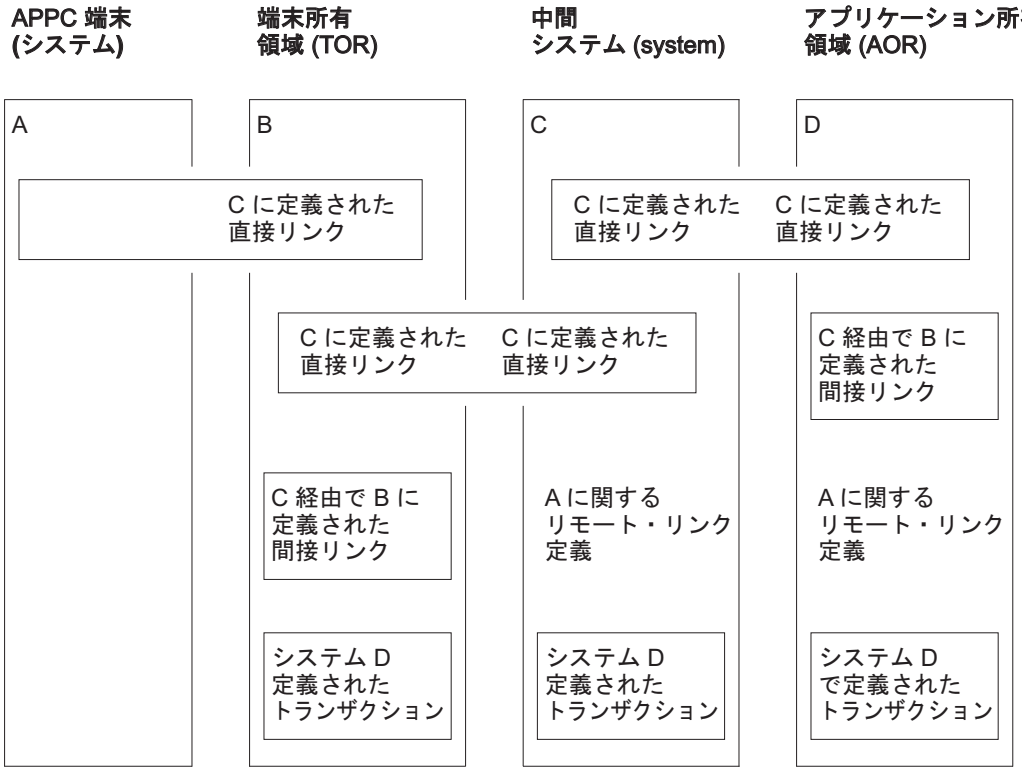


図 30. デイジー・チェーンされたシステムを介した APPC 端末へのトランザクション・ルーティング

注:

1. A のリモート・リンク定義は、ユーザーが定義することも、トランザクション・ルーティングでシステム B からシップすることもできます。
2. この例では間接リンクは必要ありませんが、可能性のあるリンクの組み合わせすべてを示すために含まれています。 202 ページの『トランザクション・ルーティングのための間接リンクの定義』を参照してください。
3. B と C および C と D のリンクは、MRO か APPC のどちらかです。

システム A (または 4 つのシステムのどれか) が端末の役割を担うことができます。これは、対になったトランザクションが、中間システムを介して会話できるようにするための技法です。次のイベント順序について考えてみます。

1. A で稼働するトランザクションが B へのリンク上にセッションを割り振り、特定のトランザクションの接続要求を出します。
2. B は、トランザクションが C にあることを判別し、A へのリンク定義によって表される基本機能に関連させて中継プログラムを開始します。
3. 接続要求が端末の詳細 (つまり A に対する B のリンク) とともに C に送信されます。C は、その端末のリモート定義を作成して、トランザクションの接続に進みます。
4. C は、さらに、そのトランザクションが**リモート**であり、D が所有するものとして定義されていることを検出します。C が中継プログラムを開始すると、そのプログラムは D のトランザクションに接続しようとします。
5. D もまた A に対する B のリンクのリモート定義を作成し、そのローカル・トランザクションに接続します。

6. これで、初めに接続要求を出した A のトランザクションは、トランザクション・ルーティング・メカニズムを通して、ターゲット・トランザクションと通信を行うことができます。

次の点に注意してください。

- APPC 端末は常にシップ可能です。したがって、これらの端末をシップ可能と定義する必要はありません。
- A と B のリンクの他のセッションでの接続要求は、別のシステムへルーティングすることができます。
- トランザクション・ルーティングによって可能になった会話のパートナーはどちらも、もう一方がどこにあるかを知りません。ただし、ルーティングされた先のトランザクションは、EXEC CICS ASSIGN PRINSYSID コマンドを使用すれば TERMINAL/CONNECTION 名を知ることができます。この名前を使用すれば、A に戻る 1 つまたは複数の追加セッションを割り振ることができます。
- D のトランザクションは、EXEC CICS (GDS) EXTRACT PROCESS コマンドから始めることもできますが、普通は、EXEC CICS (GDS) RECEIVE コマンドから始めます。

中継プログラム

端末オペレーターがリモート・システムにあるトランザクションのトランザクション・コードを入力すると、**中継プログラム**と呼ばれる CICS 提供プログラムを実行する TOR でトランザクションが生成されます。このプログラムは、端末とリモート・トランザクションの間の通信メカニズムになります。

このトランザクションに関連付けるプログラムは CICS が決めますが、その属性は、リモート・トランザクションに対するユーザーの定義によって決まります。これらは通常、リモート・システムにある「実際の」トランザクションのものです。

このトランザクションは、中継プログラムを実行するため**中継トランザクション**と呼ばれます。

中継トランザクションは、接続されると領域間セッションかシステム間セッションを獲得し、要求をリモート・システムに送って、「実際の」ユーザー・トランザクションが開始されるようにします。アプリケーション所有領域では、端末は**サロゲート** TCTTE と呼ばれる制御ブロックによって表されます。この TCTTE はトランザクションの基本機能となりますが、これは、トランザクションにとって「実際の」端末項目と区別がつきません。ただし、トランザクションがその基本機能に対して要求を出すと、その要求は CICS 端末管理プログラムによって代行受信され、領域間またはシステム間セッションを介して、中継トランザクションに返されます。中継トランザクションは、次に、その要求または出力を端末に出します。同じようにして、端末状況と入力が中継トランザクションを介してユーザー・トランザクションにシップされます。

自動トランザクション開始 (ATI) も同じような方法で処理されます。ATI によって開始されたトランザクションが、別のシステムに接続された端末が必要になると、中継トランザクションを開始するための要求が端末専有領域に送信されます。 端末が解放されていれば、中継トランザクションはその端末に接続されます。

中継トランザクションは、ユーザー・トランザクションが存続する間ずっと存在し、この間、リモート・システムへのセッションを排他的に使用します。ユーザーのトランザクションが終了すると、そのことが中継トランザクションに通知され、中継トランザクションも終了して端末を解放します。

基本マッピング・サポート (BMS)

BMS のマッピング操作は、ユーザーのトランザクションが実行されるシステム、つまりアプリケーション所有領域で実行されます。マップされた情報は、端末管理操作の場合と同じように、中継トランザクションを介して端末とこのトランザクションの間でルーティングされます。

BMS ページ作成要求とルーティング要求の場合、ページはアプリケーション所有領域に作成され、保管されます。論理メッセージが完結すると、それらのページは端末専有領域 (それらがルーティング要求によって生成された場合は複数の領域) にシップされ、アプリケーション所有領域から削除されます。ページ検索要求は、端末が接続されているシステムで実行される BMS プログラムによって処理されます。

リモート端末とオペレーターへの BMS メッセージ・ルーティング

BMS ROUTE コマンドを使用すると、メッセージをリモート端末にルーティングすることができます。

BMS ROUTE コマンドのプログラミング情報については、「*CICS アプリケーション・プログラミング・リファレンス*」の ROUTEを参照してください。ただし、メッセージが送達される端末を指定しないと、そのメッセージを、指定のリモート・オペレーターまたはオペレーター・クラスにルーティングすることはできません。

いずれの場合も、ROUTE コマンドを出すシステムには、リモート端末が定義されていなければなりません (または、シップされた端末定義が既に使用可能でなければなりません。 236 ページの『端末定義と接続定義のシップ』 を参照してください)。 80 ページの『自動トランザクション開始用端末のシップ』 の説明にある機能は、ROUTE コマンドによってアドレッシングされた端末には適用されませんので注意してください。

表 2. リモート端末とオペレーターへの BMS メッセージ・ルーティング

LIST 項目	OPCLASS	結果
指定なし	指定なし	メッセージは、元のシステムに定義されているすべてのリモート端末にルーティングされる。
オペレーターではなく端末を指定する項目	指定なし	メッセージは、指定のリモート端末にルーティングされる。
オペレーターではなく端末を指定する項目	指定あり	メッセージは、指定の OPCLASS をもつオペレーターがサインオンすると、指定のリモート端末に送達される。

表 2. リモート端末とオペレーターへの *BMS* メッセージ・ルーティング *(続き)*

LIST 項目	OPCLASS	結果
指定なし	指定あり	メッセージはどのリモート・オペレーターにも送達されない。
端末ではなくオペレーターを指定する項目	(無視)	メッセージはリモート・オペレーターに送達されない。
端末とオペレーターの両方を指定する項目	(無視)	メッセージは、指定のオペレーターがサインオンすると、指定のリモート端末に送達される。

ルーティング・トランザクション (CRTE) の使用

ルーティング・トランザクション (CRTE) は、CICS 提供のトランザクションの 1 つです。端末オペレーターは、このトランザクションを使用して、接続された CICS システムが所有するトランザクションを呼び出します。

CRTE は、任意の 3270 表示装置から使用することができます。

CRTE を使用するには、端末オペレーターは次のように入力します。

```
CRTE SYSID=xxxx  [TRPROF={DFHCICSS|profile_name}]
```

説明:

- *xxxx* は CONNECTION の名前か、リモート・システムへの接続を定義する IPCONN リソースの最初の 4 文字です。
- *profile_name* は、リモート・システムとのセッションに使用されるプロファイルの名前です。

プロファイル定義の詳細については、253 ページの『通信プロファイルの定義』を参照してください。 次に、トランザクションが、ルーティング・セッションが確立されたことを示しますので、ユーザーは次のような形式の入力データを入力します。

```
yyyyzzzzzz...
```

ここで、*yyyy* は、必要なリモート・トランザクションがリモート・システムで認識されている名前であり、*zzzzzz...* は、そのトランザクションへの最初の入力です。その後、リモート・トランザクションは、ローカルで定義して通常の方法で呼び出したときと同じように使用することができます。その後の入力はすべて、オペレーターが CANCEL を入力してルーティング・セッションを終了するまで、リモート・システムに送られます。

保護システムの場合、オペレーターは、通常、サインオンしないとトランザクションを開始できません。したがって、ルーティング・セッションで最初に呼び出されるトランザクションは、通常はサインオン・トランザクション CESN です。つまり、オペレーターがリモート・システムにサインオンします。

ルーティング・トランザクションは疑似会話型トランザクションとして実装されますが、それを呼び出した端末は、ルーティング・セッションが終了するまで CICS によって保持されます。したがって、端末を指定した ATI 要求は、CANCEL コマンドが出されるまでキューイングされます。

CRTE 機能は、特定のリモート・システムでマスター端末トランザクション CEMT を開始する際に特に役立ちます。この機能を使用すると、ローカル・システムにリモート CEMT の定義をインストールする必要がなくなります。CRTE は、最終インストールの前にリモート・トランザクションをテストする際にも役立ちます。

トランザクション・ルーティングのためのシステム・プログラミング

ご使用のシステムでトランザクション・ルーティングを実装するには、次の操作を実行する必要があります。

このタスクについて

1. MRO サポートまたは ISC サポート、あるいはその両方をインストールします。
2. 接続されるシステム間の MRO リンクまたは ISC リンクを 165 ページの『第 13 章 リモート・システムへの接続の定義方法』の説明に従って定義します。
3. トランザクション・ルーティングに参加する端末とトランザクションを 223 ページの『第 16 章 リモート・リソースの定義』の説明に従って定義します。
4. トランザクション・ルーティングに必要なローカル通信プロファイル、トランザクション、およびプログラムが、253 ページの『第 17 章 ローカル・リソースの定義』の説明どおりに、ローカル・システムに定義されてインストールされていることを確認します。
5. 動的トランザクション・ルーティングを使用する場合には、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 DFHDYP をカスタマイズするか、独自に作成します。 これを実行する方法についてのプログラミング情報は、「*CICS Customization Guide*」を参照してください。
6. シップ可能端末が「認識されていない」領域からそれらの端末へルーティングする場合は、グローバル・ユーザー出口 XICTENF および XALTENF をコーディングし、使用可能にする必要があります。 これらの出口のコーディングについてのプログラミング情報は、「*CICS Customization Guide*」を参照してください。

システム間のキューイング

リモート領域へのリンクが確立しても、使用可能な空きセッションがないと、トランザクション・ルーティング要求は、要求側の領域でキューイングされます。キューが長くなりすぎると、パフォーマンスが問題になる場合があります。

システム間キューを制御するためのガイダンスについては、307 ページの『第 24 章 システム間のセッション・キューの管理』を参照してください。

第 8 章 CICS 分散プログラム・リンク

この章では、CICS 分散プログラム・リンク (DPL) について説明します。

内容は次のとおりです。

- 『DPL の概要』
- 106 ページの『DPL 要求の静的ルーティング』
- 109 ページの『DPL 要求の動的ルーティング』
- 113 ページの『DPL サーバー・プログラムでの制約事項』
- 113 ページの『システム間のキューイング』
- 114 ページの『DPL の例』.

DPL の概要

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 領域に配置されているプログラムを実行することができます。

DPL の利点は、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できるということです。アプリケーションは、プログラム制御 LINK コマンドを通常の方法で使用します。通常、CICS プログラム定義テーブル内の項目では、指定のプログラムをローカル領域 (*クライアント領域*) ではなくリモート領域 (*サーバー領域*と呼ばれる) に入れることを指定します。

DPL 要求の図を 106 ページの図 31 に示します。この図では、CICA で実行されるプログラム (*クライアント・プログラム*) が、プログラム制御 LINK コマンドを、PGA と呼ばれるプログラム (*サーバー・プログラム*) に対して出しています。CICS は、インストール済みのプログラム定義から、このプログラムが CICB というリモート CICS システムによって所有されていることを確認します。CICS は、LINK 要求を適切な伝送形式に変更してから、それを CICB にシップして実行します。

CICB では、ミラー・トランザクション (37 ページの『第 4 章 CICS 機能シップ』を参照) に接続します。ミラー・プログラムは、元の要求を再作成して、それを CICB 上で出します。次に、ミラー・トランザクションは、サーバー・プログラムが実行されて終了すると、連絡域データをすべて CICA に返します。

© Copyright IBM Corp. 1977, 2009

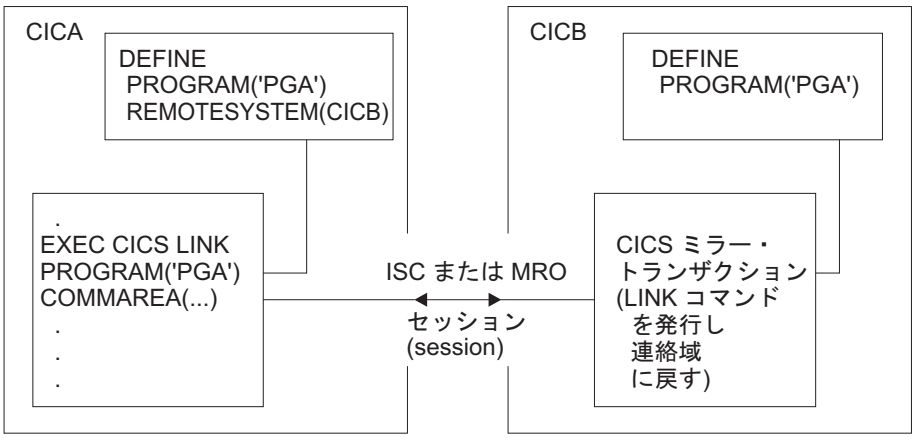


図 31. 分散プログラム・リンク

CICS リカバリー機能と再始動機能を使用すると、リモート領域のリソースの更新が可能になると同時に、クライアント・プログラムが同期点に達した時点で、保護リソースを更新中のミラー・トランザクションも同期点をとります。このため、リモート・システムとローカル・システムの保護リソースに加えられた変更の一貫性が維持されます。CSMT 一時データ・キューには、このプロセスで起こるすべての障害が通知されるため、手作業またはユーザー作成コードのいずれかによって、適切な修正処置をとることができます。

クライアント・プログラムは、CICS 相互通信環境において実行することができ、サーバー・プログラムの位置が不明な場合でも、DPL を使用することができます。サーバー・プログラムの位置は、2 つの方法のいずれかで CICS に通知されます。DPL 要求は、*静的* または*動的* のいずれかで、サーバー領域にルーティングすることができます。

クライアント領域とサーバー領域がどちらも CICS TS for z/OS バージョン 4.1 以降であれば、DPL は、IPICによる接続だけでなく、MRO および SNA を介した ISC 接続でもサポートされます。DPL over TCP/IP のサポートは、DPL over MRO および DPL over SNA のサポートと同等です。例えば、2 フェーズ・コミットとコンテナーの両方がサポートされます。

2 つの CICS 領域間に IPIC による接続と SNA を介した ISC 接続の両方が存在し、両方に同じ名前が付けられている場合には、IPIC による接続が優先されます。つまり、リモート領域 CICB が IPCONN 定義と CONNECTION 定義の両方で定義されている場合、CICS は IPCONN 定義を使用します。ただし、IPCONN が取得されない一方で、それがサービス中の場合には、SNA を介した ISC 接続が使用されます。

DPL 要求の静的ルーティング

静的ルーティングとは、サーバー・プログラムの位置が実行時ではなく設計時に指定されていることを意味します。特定のリモート・プログラムの DPL 要求は、常に同じサーバー領域にルーティングされます。一般に、静的ルーティングを使用する場合は、サーバー・プログラムの位置をインストール済みのリソースの定義で指定します (詳細は、 229 ページの『DPL 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・リソース定義では、リモート・システムで知られているサーバー・プログラムの名前が、ローカルで知られている名前と異なる場合、そのリモートでの名前を指定することもできます。サーバー・プログラムがローカル名で要求されると、CICS はその要求を、リモート名に置き換えてから送信します。この機能は、同じ名前のサーバー・プログラムが複数のシステムに存在し、その機能がプログラムがあるシステムによって異なる場合、特に便利です。例えば、ローカル・システム CICA と 2 つのリモート・システム CICB および CICC があるとしましょう。PG1 という名前のプログラムが CICB と CICC の両方に存在するとします。これらの 2 つのプログラムは CICA に定義されていますが、同じ名前を持っています。2 つの定義が必要ですので、ローカル別名と REMOTENAME を少なくともどちらかのプログラムに定義する必要があります。CICA の定義は次のようになります。

```
DEFINE PROGRAM(PG1) REMOTESYSTEM(CICB) ...
DEFINE PROGRAM(PG99) REMOTENAME(PG1) REMOTESYSTEM(CICC) ...
```

注: クライアント・プログラムの独立性が制限される可能性がありますが、クライアント・プログラムは、LINK コマンドに SYSID オプションを使用することによって、リモート・システムを明示的に指定することもできます。このオプションでリモート・システムを指定した場合、CICS は、無条件でそのシステムに要求をルーティングします。SYSID オプションの値が「ハードコーディング」されている場合 (つまり、ある可能性の範囲から推測されない場合) には、この方法も別の形の静的ルーティングであると言えます。

SYSID オプションでは、ローカル・システムも指定することができます。つまり、リモートのサーバー・プログラムにリンクするか、ローカルのサーバー・プログラムにリンクするかの決定を実行時に取ることができることを意味します。このアプローチは、単純な形の**動的ルーティング**です。

クライアント領域 (108 ページの図 32 の CICA) では、コマンド・レベル EXEC インターフェース・プログラムが、要求されたサーバー・プログラムが他のシステム (この例では CICB) にあることを判別します。したがって、このコマンドは、変換プログラムを呼び出して、この要求を伝送に適した形式に変換します (この例では、(2) がこの処理を示しています)。この例の (3) で示すように、EXEC インターフェース・プログラムは次に、変換された要求を該当する接続システムに送信するように、相互通信コンポーネントに要求します。

ミラー・トランザクションの使用

相互通信コンポーネントは、CICS 端末管理機能を使用して、ミラー・トランザクションに要求を送信します。特定のサーバー領域に対する要求があると、クライアント領域の通信コンポーネントは、サーバー・システムで接続する適切なミラー・トランザクションの ID をその形式設定した要求の前に付けます。

リソースへのアクセス制御、システム使用のアカウンティング、パフォーマンスの調整、および監査証跡の確立はすべて、指定の DPL 要求によって開始されるミラー・トランザクションにユーザー指定の名前を使用すると、さらに容易になります。このトランザクション名は、ミラー・プログラム DFHMIRS を呼び出すトランザクションとして、サーバー領域に定義する必要があります。ミラー・プログラムを呼び出すユーザー・トランザクションを定義すると、トランザクション・リソース定義の他のオプションすべてに、適切な値を自由に指定できることに留意してく

ださい。ユーザー定義ミラー・トランザクションを開始するために、クライアント・プログラムは、LINK 要求にトランザクション名を指定します。あるいは、トランザクション名は、プログラム・リソース定義の TRANSID オプションに指定することもできます。

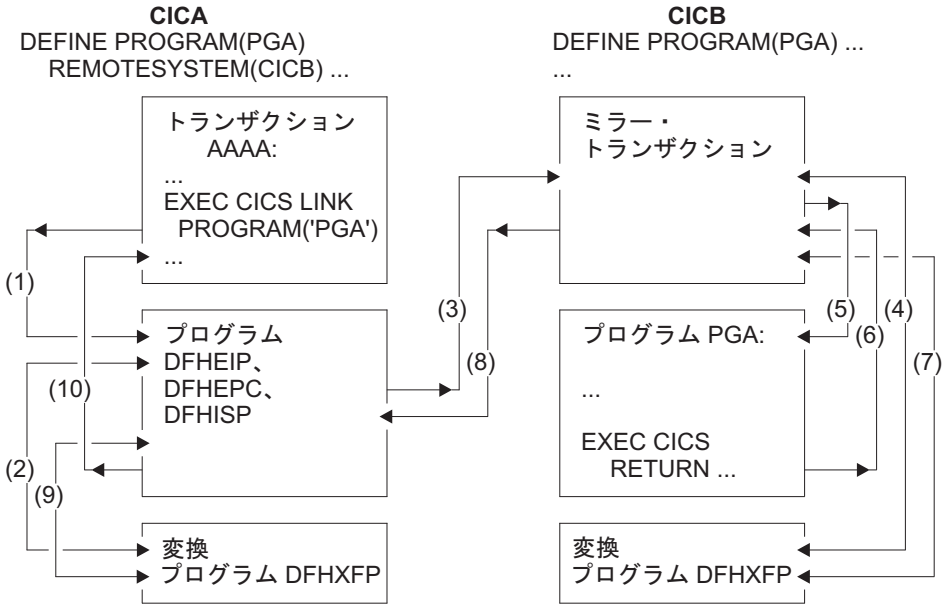


図 32. DPL における変換プログラムとミラー

図 32 の (4) が示すように、ミラー・トランザクションは、変換プログラム DFHXFP を使用して、形式設定されたリンク要求をデコードします。次にミラーは、対応するコマンドを実行し、それによってサーバー・プログラム PGA にリンクします (5)。サーバー・プログラムが RETURN コマンドを出すと (6)、ミラー・トランザクションは変換プログラムを使用して、形式設定された応答を作成します (7)。ミラー・トランザクションは、形式設定した応答をクライアント領域へ戻します (8)。その領域 (例の CICA) では、その応答が変換プログラムによって再びデコードされ (9)、その応答を使って、クライアント・プログラムから出された元の要求が完了します (10)。

ミラー・トランザクション (DPL では常に長期実行となる) は、その連絡域を送信すると延期されます。ミラー・トランザクションは、クライアント・プログラムが同期点要求を出すか、または正常に終了するまで終了しません。

クライアント・プログラムが同期点要求を出すか、または正常に終了すると、相互通信コンポーネントは、ミラー・トランザクションにメッセージを送信することによって、ミラー・トランザクションにも同期点要求を出させて終了させます。ミラー・トランザクションによる正常な同期点がクライアント領域に返された応答に示されると、同期点処理が終了して、保護リソースに対する変更すべてがコミットされます。

クライアント・プログラムは、サーバー・プログラムの位置に関係なく、複数のサーバー・プログラムに任意の順序でリンクすることができます (例えば、これらはすべて異なるサーバー領域にある可能性もあります)。クライアント・プログラムが、複数のサーバー領域内にあるサーバー・プログラムにリンクすると、相互通信

コンポーネントは、各サーバー領域でミラー・トランザクションを呼び出して、クライアント・プログラムのリンク要求を実行します。各ミラー・トランザクションは、上記の規則に従って終了します。アプリケーション・プログラムが同期点に到達すると、相互通信コンポーネントは、終了していないミラー・トランザクションがあれば、それと同期点メッセージを交換します。

グローバル・ユーザー出口による DPL 要求の宛先変更

DPL 処理の際に、2 つのグローバル・ユーザー出口を呼び出すことができます。

このタスクについて

- XPCREQ は使用可能になっていれば、CICS プログラム制御プログラムの入り口で、リンク要求が処理される**前に**呼び出されます。DPL 要求の場合、この出口は、リンクの両側、つまりクライアント領域とサーバー領域の両方で呼び出されます。
- XPCREQC は使用可能になっていれば、リンク要求が完了した**後**に呼び出されます。DPL 要求の場合、この出口はクライアント領域でのみ呼び出されます。

XPCREQ と XPCREQC は、さまざまな目的で使用できます。例えば、これらの出口を使って DPL 要求をいろいろな CICS 領域へルーティングすれば、簡単なロード・バランシングのメカニズムとして使用できます。ただし、ロード・バランシングを実行する場合は、CICS 動的ルーティング・プログラムを使用することをお勧めします。『DPL 要求の動的ルーティング』を参照してください。

XPCREQ および XPCREQC の各グローバル・ユーザー出口プログラムのプログラミング情報については、「*CICS Customization Guide*」の プログラム制御プログラムの出口を参照してください。

DPL 要求の動的ルーティング

動的ルーティングは、サーバー・プログラムの位置が、設計時ではなく、実行時に決定されることを意味します。特定のリモート・プログラムの DPL 要求は、各種のサーバー領域にルーティングすることができます。例えば、アプリケーション所有領域の複製がいくつかある場合には、動的ルーティングを使用して、その領域全体でワークロードの平衡を取ることができます。

動的ルーティング・モデル:

CICS の外側から受け取った DPL 要求の動的ルーティングでは、「ハブ」ルーティング・モデルを使用します (67 ページの『「ハブ」モデル』を参照)。

CICS-CICS 間の DPL 要求の動的ルーティングでは、分散ルーティング・モデルを使用します (68 ページの『分散モデル』を参照)。ただし、CICS-CICS 間の DPL 要求のルーティングで呼び出されるのは、*動的* ルーティング・プログラムであって、分散ルーティング・プログラムではないことに注意してください。

適格な DPL 要求であれば、**動的ルーティング・プログラム**というユーザーが置き換えできるプログラムが呼び出されます (これは、DYNAMIC として定義されたトランザクションで呼び出される動的ルーティング・プログラムと同じです。

75 ページの『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)。ルーティング・プログラムは、プログラム・リンク要求のシップ先のサーバー領域を選択します。

CICS で提供されるデフォルトの動的トランザクション・ルーティング・プログラムは DFHDYP です。提供されるプログラムは、修正することも、独自のもので置き換えることもできます。さらに、独自プログラムの名前を DFHDYP 以外のものにしたければ、DTRPGM システム初期設定パラメーターを使って、動的ルーティングのために呼び出されるプログラムの名前を指定することができます。ユーザーが置き換え可能なプログラムの概要と、動的ルーティング・プログラムの詳細については、「*CICS Customization Guide*」の 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。

プログラム・リンク要求のシップ先のサーバー領域では、静的ルーティングの場合と同じようにミラー・トランザクションが呼び出されます。

動的にルーティングできる要求

プログラム・リンク要求が動的ルーティングに対して適格である場合、リモート・プログラムをローカル・システムに対して DYNAMIC(YES) と定義するか、リモート・プログラムをローカル・システムに対して定義しないようにします。

注: EXEC CICS LINK コマンドで指定されたプログラムが現時点で定義されていない場合、次に発生することは、プログラム自動インストールがアクティブであるかどうかによって異なります。

- プログラム自動インストールが非アクティブの場合には、動的ルーティング・プログラムが呼び出されます。
- プログラム自動インストールがアクティブの場合には、自動インストール・ユーザー・プログラムが呼び出されます。そして、動的ルーティング・プログラムは、次のような場合のみ呼び出されます。
  - 自動インストール・プログラムにより、DYNAMIC(YES) を指定するプログラム定義がインストールされた場合。
  - 自動インストール・プログラムにより、プログラム定義がインストールされなかった場合。

  EXEC CICS LINK コマンドによって呼び出される自動インストール・プログラムの詳細については、 230 ページの『リモート・サーバー・プログラムの定義が必要ない場合』を参照してください。

EXEC CICS LINK PROGRAM コマンドによって実行される「従来」の CICS-CICS 間 DPL 呼び出しと同様に、CICS の外側から受け取ったプログラム・リンク要求も動的にルーティングすることができます。例えば、次のタイプのプログラム・リンク要求はすべて動的にルーティングすることができます。

- 以下から受け取った呼び出し:
  - CICS Web インターフェース
  - CICS Gateway for Java
- 外部 CICS インターフェース (EXCI) クライアント・プログラムからの呼び出し

- CICS クライアント・ワークステーション製品からの外部呼び出しインターフェース (ECI) 呼び出し
- 分散コンピューティング環境 (DCE) リモート手続き呼び出し (RPC)
- ONC/RPC 呼び出し

CICS の外側から受け取ったプログラム・リンク要求を動的にルーティングするには、次のようにします。

- CICS Transaction Server for z/OS に、プログラムを DYNAMIC(YES) と定義する
- 要求をルーティングするための動的ルーティング・プログラムをコーディングする

動的ルーティング・プログラムを呼び出す場合

プログラム・リンク要求は、「従来の」CIC-CIC 間の DPL 呼び出し、および CICS の外側から受け取った要求の両方です。プログラム・リンク要求が適格な場合、動的ルーティング・プログラムは次の時点で呼び出されます。

- リンク先プログラムが実行される前は、次のいずれかが行われます。
  - リンクのルーティング先の領域の SYSID を取得します。

    注: 呼び出し側の連絡域 (COMMAREA) のアドレスがルーティング・プログラムに渡されるので、COMMAREA の内容 (適切な場合) によって要求をルーティングすることができます。
  - ルーティング・プログラムに要求が静的ルーティングであることを通知します。この処理が行われるのは、プログラムが DYNAMIC(YES) と定義されている場合、またはプログラムが定義されていない一方で、呼び出し側が LINK コマンドの SYSID オプションでリモート領域の名前を指定している場合です。

    この場合には、ターゲット領域の明示的な指定が、動的ルーティング・プログラムから戻された SYSID に優先します。
- 代替 SYSID を指定する際に、ルーティング選択でエラーが発生した場合。例えば、動的ルーティング・プログラムから返された SYSID を利用できないか、それが認識されない場合。あるいは指定されたターゲット領域でリンクに失敗した場合です。このプロセスは、プログラム・リンクが成功するまで、または動的ルーティング・プログラムからの戻りコードがゼロ以外になるまで、繰り返されます。
- リンク要求が完了した後で、ルーティング・プログラムにより再呼び出しが要求された場合。
- リンク要求を指定のリモート・システムにシップした後で異常終了が検出された場合、ルーティング・プログラムにより再呼び出しが要求された場合。

要求のルーティングのための CICSPlex SM の使用

CICSPlex System Manager (CICSPlex SM) 製品を使用して CICSplex を管理すると、独自に動的ルーティング・プログラムを作成する必要がなくなる可能性があります。

CICSPlex SM には、ワークロード・バランシ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。必要なことは、ユーザ

ー・インターフェースを介して、CICSPlex SM に対し CICSplex 内のどの領域を動的ルーティングに使用できるかを指定することだけです。

CICSPlex SM を使用すれば、プログラム・リンク要求のワークロード・バランシングと、端末開始トランザクションのワークロード・バランシングを統合することができます。

CICSPlex SM の入門情報については、「*CICSPlex SM Concepts and Planning*」を参照してください。

CICS がトランザクション ID を取得する方法

トランザクション ID は、常に各動的プログラム・リンク要求に関連付けられます。CICS は、次の順序でトランザクション ID を取得します。

1. LINK オプションの TRANSID オプションから
2. プログラム定義の TRANSID オプションから
3. 「CSMI」、一般ミラー・トランザクション。TRANSID オプションのいずれも指定されていない場合は、これがデフォルトです。

DFHDYP などに基づいて、独自に動的ルーティング・プログラムを作成する場合には、要求に関連付けられたトランザクション ID は重要でないことがあります。例えば、プログラム名と利用可能な AOR に基づいて要求をルーティングするプログラムをコード化することができます。

ただし、CICSPlex SM を使用してプログラム・リンク要求をルーティングする場合には、トランザクション ID はさらに重要となります。これは、CICSPlex SM のルーティング論理はトランザクション・ベースだからです。CICSPlex SM は、対応付けられたトランザクションに対して指定されたルールに従って、各 DPL 要求をルーティングします。

注: CICSPlex SM システム・プログラマーは、EYU9WRAM というユーザーが置き換え可能なモジュールを使用して、DPL 要求に関連付けられたトランザクション ID を変更することができます。

DPL 要求の「デイジー・チェーン」

静的ルーティングの DPL 要求は、領域間で「デイジー・チェーン」させることができます。

例えば、A、B、C の 3 つの CICS 領域があるとします。領域 A では、プログラム P は属性 REMOTESYSTEM(B) で定義されています。領域 B では、P は属性 REMOTESYSTEM(C) で定義されています。領域 A で発行された EXEC CICS LINK PROGRAM(P) コマンドは、領域 B にシップされて実行され、領域 B から領域 C にシップされます。

動的にルーティングされた DPL 要求は、領域間でデイジー・チェーンさせることができません。2 つの CICS 領域 A および B があり、プログラム P が両方の領域で DYNAMIC(YES) として定義されている (あるいは定義されていない) とします。領域 A で EXEC CICS LINK PROGRAM(P) コマンドが発行されると、動的ルーティング・プログラムが領域 A で呼び出され、領域 B に要求をルーティングし

ます。領域 B では、プログラム P が DYNAMIC(YES) として定義されている場合でも、動的ルーティング・プログラムは*呼び出されません*。P は、領域 B ではローカルで実行されます。

DPL サーバー・プログラムでの制約事項

DPL サーバー・プログラムでは、次の種類のコマンドを出すことはできません。

- その基本機能に及ぶ端末管理コマンド
- 端末属性を設定または照会するコマンド
- BMS コマンド
- サインオン・コマンドおよびサインオフ・コマンド
- バッチ・データ交換コマンド
- TCTUA をアドレッシングするコマンド
- 同期点コマンド (クライアント・プログラムが LINK 要求に SYNCONRETURN オプションを指定する場合を除く)

クライアントが SYNCONRETURN を指定すると、

- サーバー・プログラムは同期点要求を出すことができます。
- ミラー・トランザクションは、サーバー・プログラムの処理が完了すると、同期点を要求します。

重要: これらの同期点はどちらも、サーバー・プログラムによって行われた作業だけをコミットします。クライアント・プログラムとサーバー・プログラムの両方がリカバリー可能リソースを更新するアプリケーションでは、クライアント・プログラムが LINK 要求を出した後で障害を起こした場合、この種の同期点によって、データ保全性の問題が生じるおそれがあります。

DPL のアプリケーション・プログラミングの詳細については、 271 ページの『第 20 章 CICS DPL のアプリケーション・プログラミング』を参照してください。

システム間のキューイング

リモート領域へのリンクが確立しても、空きセッションがないと、分散プログラム・リンク要求は、要求側の領域でキューイングされることがあります。キューが長くなりすぎると、パフォーマンスが問題になる場合があります。

システム間キューを制御するためのガイダンスについては、 307 ページの『第 24 章 システム間のセッション・キューの管理』を参照してください。

DPL の例

この項では、ミラー・トランザクションの存続時間と、クライアント・プログラムとそのミラー・トランザクションとの間の情報の流れを示す例をいくつか示します。

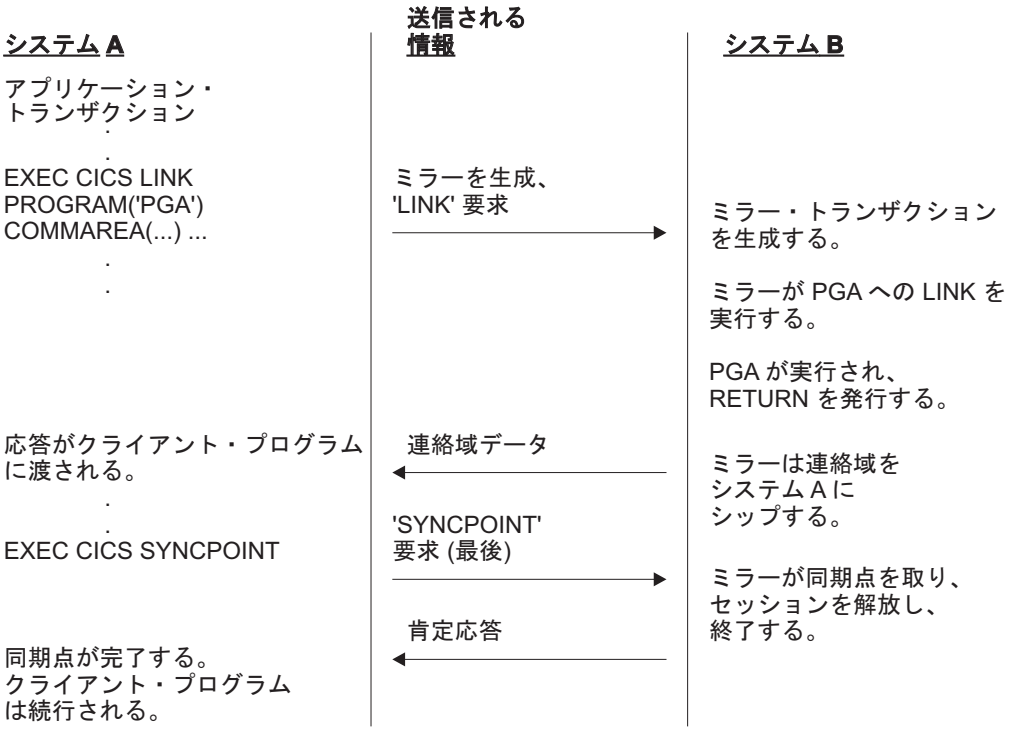


図 33. DPL でクライアント・トランザクションが同期点を出す

図 33 は、クライアント・トランザクションが同期点を発行する DPL 要求を示しています。ミラーは常に長期実行となるため、SYNCPOINT を受け取る前に終了することはありません。

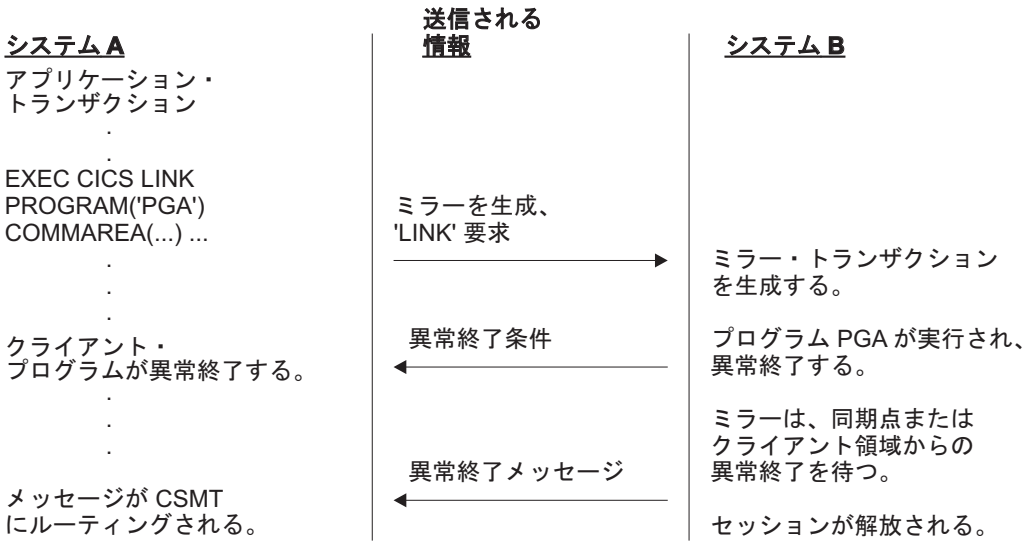


図 34. DPL でサーバー・プログラムが異常終了

図 34 は、サーバー・プログラムが異常終了する DPL 要求を示しています。

第 9 章 分散トランザクション処理

トランザクションの機能をネットワーク内のいくつかのトランザクション・プログラムに分散するこの技法を、**分散トランザクション処理 (DTP)** と言います。

この章には以下のトピックが含まれています。

- 『DTP の概要』
- 『機能シップやトランザクション・ルーティングに対する利点』
- 119 ページの『なぜ分散トランザクション処理なのか』
- 119 ページの『会話とは何か、なぜ必要なのか』
- 124 ページの『DTP では MRO か APPC か』
- 125 ページの『APPC マップ式会話か、基本会話か』
- 126 ページの『EXEC CICS か CPI コミュニケーションか』.

DTP の概要

CICS は、機能シップや、分散プログラム・リンク (DPL)、非同期トランザクション処理、トランザクション・ルーティングなどを準備する際、リモート・システムへの論理データ・リンクを確立します。

次に、2 つのシステムの間でデータ交換が行われます。このデータ交換は、CICS 提供のプログラムにより、APPC、LUTYPE6.1、または MRO のいずれかのプロトコルを使用して制御されます。 CICS 提供のプログラムは、コマンドを出して会話を割り振り、システム間でデータを送受信します。これらと同等のコマンドがアプリケーション・プログラムでも使用できますので、これらのコマンドを使用すればアプリケーションの間で会話を行うことができます。トランザクションの機能をネットワーク内のいくつかのトランザクション・プログラムに分散するこの技法を、**分散トランザクション処理 (DTP)** と言います。

5 つの相互通信機能のうち、DTP は最も柔軟性があり、最も強力な機能ですが、同時に最も複雑な機能でもあります。この章では、この機能の基本的な概念について紹介します。

DTP アプリケーションの開発については、「*CICS Distributed Transaction Programming Guide*」を参照してください。

機能シップやトランザクション・ルーティングに対する利点

機能シップを使用すれば、リモート・リソースにアクセスでき、トランザクション・ルーティングを使用すれば、端末からリモート・トランザクションと通信することができます。

一見、これら 2 つの機能があれば、すべての相互通信の要件が満たされるように見えるかもしれません。確かに、機能の点からいえば、これらの機能でおそらく十分でしょう。しかし、純粋に機能的な側面からだけでは満足できない設計上の基準が

© Copyright IBM Corp. 1977, 2009

常にあります。例えば、トランザクションの設計に影響する要素としては、計算機の負荷、応答時間、サービスの連続性、リソースの経済的な使用などがあります。

次の例を考えてみます。

> *あるスーパーマーケット・チェーンに多数の支店があり、それらの支店はいくつかの流通センターから品物を仕入れています。それぞれの流通センターには別々の種類の商品が在庫されています。支店の在庫記録は、POS 端末によってオンラインで更新されます。また、それぞれの流通センターごとに販売情報をソートし、再注文と配達ができるようにそれを支店に伝送する必要があります。*

アナリストは、機能シップを使って、再注文が必要になるたびにそれをリモート・ファイルに書き込む方法をとりたくなります。この方法は、簡潔ではありますが、いくつかの理由で採用すべきではありません。

- データは、小さなパケット単位で不定期にリモート・システムに伝送されます。つまり、リンクの使用効率がよくありません。
- POS 装置に関連するトランザクションが、リモート・システムとのセッションを求めて競合します。つまり、遅延のため POS が実用的な使用に耐えない可能性があります。
- リンクに障害が発生すると、支店での操作ができなくなります。
- 相互通信のアクティビティーが集中すると (ピーク時など)、POS 端末のパフォーマンスが低下します。

次に、それぞれの売上トランザクションごとに、その再注文レコードを一時データ・キューに書き込む方法を考えてみましょう。この場合、データはすみやかに処理され、トランザクションが端末との会話を続けます。

仕入れの要求が緊急の場合はまれですので、ピーク時が過ぎるのを待って、データのソートと送信を行うことができるでしょう。あるいは、データ量が事前に定義したレベルに達したときに送信側トランザクションを起動するために、一時データ・キューを設定することができます。どちらの方法であっても、送信側トランザクションが行う仕事は同じです。

再注文レコードを送信するのには、また機能シップを使いたくなります。ソート処理が終わったあとで、各レコードを該当するリモート・システムのリモート・ファイルに書き込むこともできます。しかし、これも理想的な方法とはいえません。送信側トランザクションは、各レコードを書き込んだあと、正しい応答を受け取るまで待たなければなりません。この場合、リンクの使用効率がよくないことの他に、レコードとレコードの間で待機することにより、処理全体の速度が非常に遅くなります。この章では、分散トランザクション処理を使ってこの問題を解決する方法やその他の点について説明します。

状況によっては、DTP の柔軟性を利用すれば、機能シップよりもよいパフォーマンスが得られます。例えば、リモート・ファイルをブラウズして、ある基準を満たすレコードを選択する例を考えてみましょう。機能シップを使用すると、CICS は、そのリンクで GETNEXT 要求をシップし、ミラーにその操作を実行させ、レコードを要求側に戻します。

このとき、多くのアクティビティーが実行されます。つまり、ネットワークには 2 つのフローが必要になり、データ・フローも非常に大きくなります。大きなファイルをブラウズする場合には、オーバーヘッドが非常に高くなる可能性があります。これに代わる方法としては、選択基準をシップし、選択したレコードのキーと該当フィールドだけを戻す DTP 会話を作成する方法があります。こうすれば、リンクにおける流れの数と送信されるデータ量の両方が減るため、機能シップの場合のオーバーヘッドを減らすことができます。

なぜ分散トランザクション処理なのか

複数システム環境では、エンド・ユーザーがリモート・リソースにアクセスするためにシステム間のデータ転送が必要になります。

これらのリモート・リソースを管理する際には、ネットワーク・リソースを使用しますが、ネットワークを使いすぎると、パフォーマンスが低下します。したがって、あるリソースに関する処理をそのリソースの所有領域で行うようにアプリケーションを設計すれば、パフォーマンスが向上します。

DTP では、データが発生するところでそれを処理します。したがって、中央の処理でデータを集めるために、ネットワーク・リソースを頻繁に使用する必要はありません。

DTP を使用する理由はもちろん他にもあります。DTP には次の機能があります。

- ある程度の並列処理により応答時間を短縮することができる。
- いくつかの異なるトランザクションによって接続されるトランザクションに対し共通のインターフェースとなる。
- 他のシステム (特に非 CICS システム) で稼働するアプリケーションと通信することができる。
- セキュリティーが問題となるファイルまたはデータベースとアプリケーションの間のバッファーを提供する。そのため、アプリケーションはファイル・レコードの形式を知る必要がありません。
- リモート・システムへ送信されるデータが緊急でない場合、それらをバッチで処理できるようにする。

会話とは何か、なぜ必要なのか

DTP では、トランザクションがデータを相互に直接渡します。一方が送信しているときは、他方が受信します。2 つのトランザクションの間で行われるデータ交換を**会話**といいます。

単一の分散処理でいくつかのトランザクションがかかわることがありますが、それらの間の通信は、対のトランザクションごとの自己完結型のいくつかの会話に分かれます。それぞれの会話では、**セッション**という CICS リソースを使用します。

会話の開始とトランザクション階層

トランザクションは、リモート・システムへのセッションの使用を要求して、会話を開始します。そのセッションを確立すると、トランザクションは、接続要求が他のシステムに送信されるようにして、会話のパートナーとなるトランザクションを活動化します。

トランザクションは、他のトランザクション、したがって会話をいくつでも開始することができます。複雑な処理では、最上部に端末開始トランザクションがくる個別の階層ができます。図 35 に構成の例を示します。トランザクション TRAA が端末セッションを介して接続されます。トランザクション TRAA はトランザクション TRBB と接続します。このトランザクションは、次にトランザクション TRCC および TRDD と接続します。これらの 2 つのトランザクションは、システム CICSE にある同じトランザクション SUBR と接続します。これによって、SUBR の 2 つの異なるタスクが発生します。

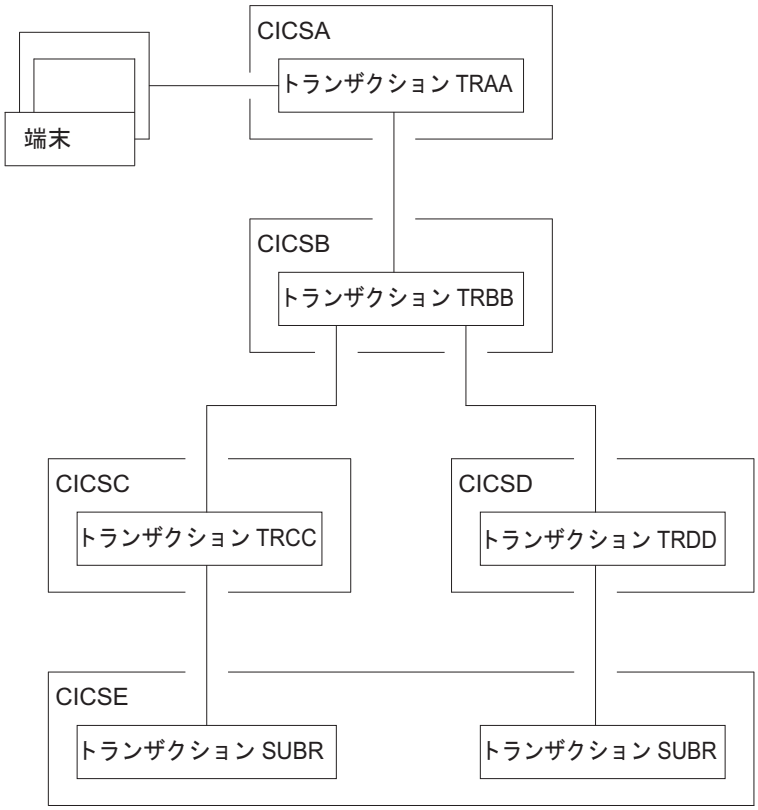


図 35. 複数システム構成の DTP

分散処理の構造はプログラムによって動的に決まります。これを前もって定義することはできません。各トランザクションでは、インバウンドの接続要求は 1 つだけですが、アウトバウンドの接続要求はいくつでも構いません。トランザクションを活動化するセッションのことを、その**基本機能**といいます。別のトランザクションを活動化するためにトランザクションによって割り振られるセッションを、その**代替機能**といいます。したがって、トランザクションには、基本機能は 1 つしかありませんが、代替機能はいくつもある場合があります。

会話を開始するトランザクションは、その会話の**フロントエンド**です。その会話のパートナーは、同じ会話の**バックエンド**です。 (マニュアルの中には、フロントエンドを開始側、バックエンドを受信側と呼んでいるものもあります)。上位にあたるフロントエンドが会話の方法を管理し、判別するのが普通です。必要ならバックエンドがフロントエンドの代わりをすることが可能ですが、複合処理では、このようにすると、必要以上に複雑になるおそれがあります。以上の内容については、この章の同期化の項でさらに説明します。

2 つのトランザクション間でのダイアログ

データは、会話によってトランザクションからトランザクションへ伝送されます。

これが正しく機能するためには、各トランザクションは、もう一方のトランザクションが意図していることを知っていなければなりません。例えば、バックエンドが週次販売報告書を印刷しようとしているときに、フロントエンドがデータを送っても意味がありません。したがって、フロントエンドとバックエンドを 1 つのソフトウェア単位として設計、コーディング、およびテストする必要があります。これは、会話とトランザクション・プログラムが複数になっても同様です。新しい会話を追加すれば、全体的な設計の複雑さが増します。

117 ページの『機能シップやトランザクション・ルーティングに対する利点』の例の場合、DTP の解決方法は、一時データ・キューの内容をフロントエンドからバックエンドへ送信するというものです。フロントエンドは、キューから取り出すレコードごとに SEND コマンドを出します。バックエンドは、伝送の終了を示す通知を受け取るまで RECEIVE コマンドを出し続けます。

実際には、ほとんどの会話では、1 つのデータ・ファイルがトランザクションから別のトランザクションに伝送されます。さらに複雑なのは、バックエンドがフロントエンドにデータ (なんらかの処理の結果) を戻さなければならないということです。そのためフロントエンドは、適切な箇所で会話の送受反転を要求するようにプログラミングされます。

制御フローと制御ブラケット

会話の間、データはリンクを介して両方向に渡されます。

1 つの伝送のことを流れ (**フロー**) といいます。SEND コマンドを出しても、常に流れが発生するとは限りません。これは、ユーザー・データの伝送が据え置かれる場合があるからです。つまり、伝送は、なんらかのイベントが起こるまでバッファーに保持されることがあります。データの形式とパック方法は、APPC 体系によって定義されます。これらのことは、CICS がユーザーの代わりに処理するので、これらを知る必要があるのは、デバッグのために流れをトレースする場合だけです。

APPC 体系では、伝送ごとにデータ・ヘッダーを定義します。このヘッダーには、それに続くデータの目的と構造についての情報が保持されます。さらに、このヘッダーには、もう一方の側に制御情報を送るためのビット標識も入っています。例えば、一方が送信を開始できることを他方に伝えたいとき、CICS はそのヘッダー内のビットを設定して、会話の方向の変更を知らせます。

流れをできるだけ少なくするために、緊急でない制御標識は累積され、ユーザー・データの送信が必要になったときに、そのヘッダーに追加されます。

APPC が使用するヘッダーと制御標識の形式については、「*SNA Formats*」を参照してください。

同期点の設定などの複雑なプロシージャーでは、送信できるユーザー・データがないときに制御標識を送信しなければならない場合があります。このことを**制御フロー**といいます。

会話の始め (つまり、トランザクションが接続されたとき) は、BEGIN_BRACKET によってマークされます。会話の終わりは、CONDITIONAL_END_BRACKET です。会話は状況によっては再度オープンされることがありますので、ブラケット終了は条件付きです。会話は、アクティブにある場合**ブラケット内**です。

MRO は、その内部編成において APPC とまったく異なるわけではありません。これは、LUTYPE6.1 (これもまた SNA 定義体系の 1 つ) に基づいているからです。

会話の状態とエラーの検出

会話は、進行につれて、会話している両方のトランザクションにおいてある状態から別の状態へ変わります。

出すことのできるコマンドは、会話の状態によって決まります。例えば、フロントエンドとバックエンドをリンクするセッションがなければ、データの送信や受信をしようとしても意味がありません。同じように、バックエンドが会話の終わりを知らせたら、フロントエンドは、この会話でこれ以上のデータを受信することはできません。

会話のどちらの側も状態を変更することができます。このためには、通常、特定の状態から特定のコマンドを出します。CICS は、この変更を追跡することによって、トランザクションが不適切な状態で不適切なコマンドを出せないようにします。

同期

トランザクションの実行時には、異常な状態になるものがいろいろあります。会話プロトコルは、エラーからのリカバリーをサポートし、両側の歩調が常に合うようにします。このようなプロトコルの使い方を**同期**といいます。

同期によって、一時データ・キューやファイルなどのリソースを保護することができます。トランザクションの実行時に異常が起きた場合、関連するリソースを不整合のままにしておくことはできません。

使用例

あるトランザクションから 1 つのキュー分のデータを別のシステムに伝送し、DASD ファイルに書き込むとします。さらに、なんらかの理由で (相互通信のアクティビティーに必ずしも関連していなくても)、受信側のトランザクションが異常終了したとします。

これ以上の異常終了を防げたとしても、データの損失なしに処理を続行するにはどうするかという問題があります。いくつのキュー項目が受信され、そのうちいくつが DASD ファイルに正しく書き込まれたのかは明らかではありません。これを続けるための唯一の安全な方法は、キューの内容とファイルの内容が一致していることが分かっている所まで戻ることです。しかし、これには 2 つの問題があります。一

方の側では、送信済みのキュー項目を復元する必要があり、他方の側では、それに対応する項目を DASD ファイルから削除する必要があります。

最後の整合状態以降にリカバリー可能リソースに行われたすべての変更をアプリケーション・プログラムによって取り消すことを、**ロールバック**といいます。リソースをリカバリーするための物理的な処理のことを**バックアウト**といいます。分散されたリソースの間の整合性が失われていない状態のことを、**データ保全性**といいます。

エラーの状態でない場合でも、リソースをリカバリーしたい場合があります。受注システムを考えてみます。顧客の注文を入力しているときに、オペレーターがその注文を受けると、その顧客の貸付限度額が超過してしまうことをシステムから知らされたとします。顧客の調査が終わらなければ入力を続けても意味がないので、PF キーを押してその注文を破棄します。それに対応して、トランザクションは、データ・リソースをその注文の開始時点の状態に復元するようにプログラムされています。

同期点をとる

ユーザー自身のデータ移動のログを取る場合は、ファイルとキューのバックアウトを行うことができます。

しかし、そのためには非常に複雑なプログラミングが必要ですし、同じことを同様のアプリケーションごとに行わなければなりません。このオーバーヘッドを軽減させるために、リソース・リカバリーを CICS がユーザーに代わって行います。LU 管理とリソース管理は、協調して、リソースが確実に回復できるようにします。

リソースが整合状態にあると宣言されているポイント (処理における) を**同期点 (synchronization points)** といい、しばしば **syncpoints** と略されます。同期点は暗黙的にトランザクションの始めと終わりにあります。トランザクションは、プログラム・コマンドによって他の同期点を定義することができます。2 つの連続する同期点の間にあるすべての処理は、1 つの**作業単位** (UOW) に属します。

同期点をとると、すべてのリカバリー可能リソースが**コミット**されます。つまり、分散処理にかかわるすべてのシステムは、リカバリー可能リソースのデータ移動について保存していたすべての情報を消去します。これで、バックアウトを行うことはできなくなり、最後の同期点よりあとに行ったリソースへのすべての変更は取り消すことができなくなります。

リソースの変更に対するコミットとバックアウトは CICS がユーザーに代わって行いますが、このサービスの代償がパフォーマンスに現れます。トランザクションによっては、このような複雑なことは必要ありませんので、このサービスを使用してもむだになる場合があります。リソースのリカバリーが問題にならないときは、簡単な同期方式が使用できます。

3 つの同期レベル

APPC 体系では、以下の 3 つのレベルの同期 (**同期レベル**) が定義されています。

- レベル 0 - なし
- レベル 1 - 確認
- レベル 2 - 同期点

同期レベル 0 では、同期に対するシステムのサポートはありません。しかし、SEND コマンドと RECEIVE コマンドを使用すれば、データを交換することによってある程度の同期をとることができます。

同期レベル 1 を選択すると、2 つの会話パートナーの間で通信のための特別なコマンドを使用することができます。一方のトランザクションは、もう一方のトランザクションが継続して存在することと、準備ができていることを*確認* することができます。ユーザーには、リカバリー可能リソースのデータ保全性を維持する責任があります。

この項の最初の部分で説明した同期のレベルは、同期レベル 2 に相当します。ここでは、リカバリー可能リソースのデータ保全性を維持するためのシステム・サポートが利用できます。

CICS は、トランザクションを開始するときに同期点を暗黙指定します。つまり、リカバリー可能リソースへの変更のログを開始しますが、制御フローは発生しません。CICS は、トランザクションが通常終了したときに完全な同期点を取ります。トランザクションが異常終了すると、ロールバックが発生します。トランザクションは、同期点要求またはロールバック要求を開始することができます。しかし、同期点要求やロールバック要求は、発信元のトランザクションが他方のトランザクションと会話しており、その会話に同期レベル 2 が選択されている場合にのみ、他方のトランザクションへ伝達されます。

同期点やロールバックは、トランザクション内のどれか 1 つの会話に特有なものではないことを覚えておいてください。同期点やロールバックは、現在ブラケット*内*にあるすべての同期レベル 2 の会話で伝達されます。

DTP では MRO か APPC か

DTP アプリケーションは、MRO と APPC どちらのリンクに対してもプログラムできます。これらの 2 つの会話プロトコルは同じではありません。特定のアプリケーションに対してどちらかを選択できる機会はほとんどありませんが、それらの相違点と類似点を知っていれば、互換性について判断が必要なときに役立ちます。

MRO を選ぶか APPC を選ぶかを決めるのは、きわめて簡単なことです。どちらを選択すべきであるのかは、CICS 複合システムの構成、および会話パートナーの性質よって決まります。非 CICS システムにいるパートナーと通信する場合には、MRO は使用できません。さらに MRO では、異なる MVS イメージにある CICS システムで稼働しているトランザクションの間の通信がサポートされますが、それらの MVS イメージは、同じ MVS シスプレックスにあり、しかもシステム間カップリング・ファシリティー (XCF) リンクによって接続されていなければなりません。XCF/MRO のハードウェアとソフトウェアの要件については、XCF/MRO のインストール要件を参照してください。

CICS システムが同じ MVS イメージにあるか、同じシスプレックスにあれば、その CICS にあるパートナーと通信するために MRO と APPC のどちらのプロトコルでも使用できます。パフォーマンスの点では MRO を使用する価値があります。しかし、分散トランザクションで他のオペレーティング・システムのパートナーと通信する可能性がある場合は、トランザクションの変更が不要なため APPC が優れています。

表 3 に、これら 2 つのプロトコルの主な違いを要約します。

表 3. MRO と APPC の比較

MRO	APPC
機能は CICS 内で実現される	VTAM または同等の機能に依存する
非標準体系	SNA 体系
CICS-CICS 間リンクのみ	非 CICS システムへのリンクが可能
単一 MVS イメージ内、または (XCF/MRO を使って) 同じシスプレックス内の MVS イメージ間で通信する	複数の MVS イメージや他のオペレーティング・システムとの間で通信する
PIP データはサポートされない	PIP データがサポートされる
データ伝送は据え置かれない	データ伝送が据え置かれる
パートナー・トランザクションがデータ内で識別される	パートナー・トランザクションはプログラム・コマンドによって定義される
RECEIVE は受信状態でのみ出せる	マップ式会話の送信状態で RECEIVE が出されると、送受反転が起こる
急送流れは不可能	ISSUE SIGNAL コマンドの流れは急送される
WAIT コマンドは機能しない	WAIT コマンドによって据え置きデータの伝送が起こる

APPC マップ式会話か、基本会話か

APPC 会話は**マップ式**か**基本**のどちらかです。CICS-CICS 間のアプリケーションを使用する場合には、マップ式会話だけが使用されます。基本会話 (「非マップ式」とも言います) は、マップ式会話をサポートしないシステムと通信する場合に使用します。これには一部の APPC デバイスが含まれます。

これら 2 つのプロトコルは似ています。主な違いは、伝送のためのユーザー・データの形式設定にあります。マップ式会話ではパートナーに送りたいデータだけを送信しますが、基本会話では、いくつかの制御バイトを追加し、データを**汎用データ・ストリーム** (GDS) と呼ぶ SNA 定義形式に変換する必要があります。さらに、基本会話では EXEC CICS コマンドに キーワード GDS を指定しなければなりません。

表 4 に、マップ式会話と基本会話の違いを要約します。これは CICS API だけに適用されることに注意してください。次項で説明する CPI コミュニケーションには、それ独自の規則があります。

表 4. APPC 会話 - マップ式か基本か

マップ式	基本
会話の両パートナーは、そのアプリケーションだけに関連するデータを交換する。	両パートナーは、ユーザー・データを送信前にパックし、受信したらアンパックしなければならない。
1 つのトランザクションに対するすべての会話は、状況報告に同じ EXEC インターフェース・ブロックを共用する。	各会話には、状態情報のためのそれ独自の区域がある。

表 4. APPC 会話 - マップ式か基本か (続き)

マップ式	基本
トランザクションは、例外条件を処理することもできるし、デフォルト解釈にまかせることもできる。	トランザクションは、例外条件がないかどうかを、そのために確保されているデータ域をみて検査する。
送信状態で RECEIVE コマンドを出すと、会話の送受反転が起こる。	送信状態で RECEIVE コマンドを出すことはできない。
トランザクションは任意のサポート言語で作成できる。	トランザクションはアセンブラー言語か C でなければ作成できない。

EXEC CICS か CPI コミュニケーションか

CICS では、APPC セッションの DTP 会話をコーディングする場合、2 つのアプリケーション・プログラミング・インターフェース (API) から選択することができます。

最初の **CICS API** は、APPC 体系を使用する CICS のプログラミング・インターフェースです。これは、EXEC CICS コマンド群からなり、CICS によってサポートされるすべての言語で使用できます。もう一つの**共通プログラミング・インターフェース通信** (CPI コミュニケーション) は、SAA 環境のために定義された通信インターフェースです。これは定義された動詞 (verb) 群からなり、これらの verb は、使用する言語に合ったプログラム呼び出し形式になっています。

表 5 は、2 つの方式を比較したもので、特定のアプリケーションでどちらの API を使用すべきかを決める際に便利です。

表 5. CICS API と CPI コミュニケーションの比較

CICS API	CPI コミュニケーション
CICS ファミリーの異なるメンバーの間で移植が可能。	SAA 機能をサポートするシステムの間で移植が可能。
基本会話は、アセンブラー言語か C でのみプログラミングが可能。	基本会話は、使用可能な任意の言語でプログラミングが可能。
同期レベル 0、1、および 2 がサポートされる。	同期レベル 0、1、および 2 がサポートされる。*ただし、トランザクション・ルーティングの場合は、同期レベル 0 と 1 だけがサポートされる。*
PIP データがサポートされる。	PIP データはサポートされない。
プログラム可能な会話特性は少ない。残りの特性は、リソース定義によって定義される。	ほとんどの会話特性は、トランザクション・プログラムで動的に変更できる。
ATI によって開始されたトランザクションに対する基本機能で使用できる。	ATI によって開始されたトランザクションに対する基本機能で使用することはできない。
MRO との互換性は限定される。	MRO との互換性はない。

同じトランザクションに CPI コミュニケーション呼び出しと EXEC CICS コマンドを混在させることはできますが、同じ会話の同じ側でそれらを混在させることはできません。分散トランザクションにおいて、会話の一方のパートナーが CPI コミュニケーション呼び出しを使用し、他方が CICS API を使用することができます。

このような場合、両側の API が整合性をもって APPC 体系にマップされるようにプログラミングするのはユーザーの責任です。

第 2 部 相互通信サポートのインストールおよび構成

CICS システムがシステム間連絡と複数領域操作のどちらに関与するかによって、インストールと構成の要件は異なります。

CICS インストールの一般的な要件については、*CICS Transaction Server for z/OS* インストール・ガイドを参照してください。CICS のシステム初期設定パラメーターのコーディングについては、*CICS System Definition Guide*の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。

135 ページの『第 11 章 複数領域操作の構成』では、複数領域操作に対する CICS の設定方法について説明します。

131 ページの『第 10 章 システム間通信の構成』では、システム間通信に対する CICS の設定方法について説明します。さらに、この章では、ACF/VTAM や IMS といった製品を CICS と一緒にシステム間通信の環境で使用する場合のインストール要件についての注意事項も記載しています。

137 ページの『第 12 章 VTAM 総称リソースの構成』では、端末専有領域を VTAM 総称リソース・グループのメンバーとして登録する方法と、その際に考慮すべき事項について説明します。

© Copyright IBM Corp. 1977, 2009

第 10 章 システム間通信の構成

CICS を構成することにより、システム間通信環境で、TCP/IP または SNA を介して通信することができます。

TCP/IP ネットワークを介した通信のサポートの構成

デュアル・モード環境で作動している CICS は、IPv4 および IPv6 の両方のネットワークを使用します。IPv4 ネットワークを使用する前に、常に IPv6 を使用した通信が試行されます。単一モード環境は、IPv4 ネットワークでのみ実現可能です。HTTP や IPIC を含む、CICS のサポートするいくつかのプロトコルを使用するように TCP/IP サービスをセットアップできます。

始める前に

IPv6 で通信するには CICS TS 4.1 の最小レベルが必要です。CICS TS 4.1 領域はデュアル・モード (IPv4 および IPv6) 環境で実行しており、CICS の通信先のクライアントまたはサーバーもデュアル・モード環境で実行されている必要があります。領域が単一モード (IPv4) 環境で実行している場合、または領域が CICS TS 4.1 より前のリリースで実行している場合は、IPv4 のみを使用して通信することができます。

このタスクについて

次の手順に従って、IPv4 と IPv6 アドレッシングのいずれか、またはその両方の形式を組み合わせて使用するように接続を構成します。

1. システム初期設定パラメーターとして **TCPIP=YES** を指定して、TCP/IP サービスをアクティブ化します。
2. TCP/IP ネットワークで通信に使用するプロトコルをサポートするように、リソースを定義します。次に、リソースを使用して定義できる 2 つの別個のプロトコルの例を示します。
   a. IPIC を使用する場合は、IPCONN リソース定義と TCPIPSERVICE リソース定義を両方のパートナー領域に定義してインストールします。 リソース定義を定義およびインストールする方法と例については、 169 ページの『IP 相互接続 (IPIC) の定義』を参照してください。
   b. HTTP クライアントとしての CICS で HTTP を使用する場合は、送出側領域に URIMAP(CLIENT) リソース定義を、listen 側領域に TCPIPSERVICE リソース定義を定義してインストールします。URIMAP(CLIENT) リソース定義の HOST 属性で使用するホスト名、IPv4 アドレスまたは IPv6 アドレスを定義します。 HTTP 要求用の URIMAP 定義に関する情報は、HTTP クライアントとしての CICS からの HTTP 要求用の URIMAP 定義の作成 (Creating a URIMAP definition for an HTTP request by CICS as an HTTP client) 「*CICS インターネット・ガイド*」を参照してください。
3. オプション: ネットワーク管理者に、IPv4 のプライマリー・インターフェース・アドレスを定義するように指示し、CICSplex の外部で通信する際に問題が発生しないことを確認します。プライマリー・インターフェース・アドレスは、

© Copyright IBM Corp. 1977, 2009

TCPIP.PROFILE の PRIMARYINTERFACE ステートメントで指定されたアドレスです。 GETHOSTID 呼び出しを発行すると、GETHOSTID は Ipv4 プライマリー・インターフェース・アドレスまたはループバック・アドレスを返します (GETHOSTID がホスト・アドレスを見つけられなかった場合)。**IPRESOLVED** オプションを指定すると、GETHOSTID によって返されたアドレスが格納されます。このため、**IPRESOLVED** にはプライマリー・インターフェース・アドレスまたはループバック・アドレスが格納されます。CICSplex の外部で通信している場合、ループバック・アドレスが返される際の結果は予測不能です。プライマリー・インターフェース・アドレスを定義する方法については、「*z/OS Communications Server IP 構成ガイド*」で TCP/IP アドレス・スペース (PROFILE.TCPIP) に関する説明を参照してください。

タスクの結果

TCP/IP 接続が正しく構成され、IPv4 接続で使用可能になります。

CICS のレベルが適正であり、使用している環境にデュアル・モード機能がある場合は、IPv6 で接続を使用することも可能になります。

次のタスク

接続時に問題が発生する場合は、「*CICS Problem Determination Guide*」を参照してください。

SNA を介した ISC のサポートの構成

このセクションで示す ACF/VTAM と IMS の情報は、概要を示すためのものです。最新情報について、現行の ACF/VTAM または IMS の資料を必ず参照してください。

ISC over SNA では ACF/VTAM アクセス方式が使用されます。このため、ACF/VTAM をインストールする場合は、システム間連絡プログラムとオペランドをシステムに組み込み、SNA を介したシステム間通信 (SNA を介した ISC) を許可する必要があります。

1. システムにシステム間連絡プログラムを組み込みます。それには、VTAM および ISC の各システム初期設定パラメーターで YES を指定します。
2. CICS システムを ACF/VTAM に対して定義する場合は、システム間連絡オペランドを VTAM APPL ステートメントに含めます。
3. CICS インストールで CICS-IMS 間のシステム間連絡を使用する場合は、CICS と IMS のインストール・システムの間に完全な互換性があることを確認してください互換性のある CICS ノードと IMS ノードの定義方法については、165 ページの『第 13 章 リモート・システムへの接続の定義方法』を参照してください。IMS インストールの詳細については、「*IMS Installation Guide*」を参照してください。
   a. VTAM APPL ステートメントにシステム間連絡オペランドを含めます。
   b. IMS ISC 関連のマクロおよびパラメーターを定義します。197 ページの『互換の CICS ノードと IMS ノードの定義』を参照してください。

詳しくは、「*CICS Transaction Server for zOS インストール・ガイド*」を参照してください。

第 11 章 複数領域操作の構成

CICS を構成することにより、複数領域操作を使用して通信することができます。

この章には、次のトピックが収められています。

- 『MRO サポートの提供』
- 『MRO 構成後の手順』.

このトピックで示す MVS/ESA の情報は、概要を示すためのものです。最新情報について、現行の MVS/ESA の資料を必ず参照してください。

MRO サポートの提供

複数領域操作のサポートを提供するには、CICS を MVS サブシステムとして定義し、必要な CICS モジュール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

これらのトピックに対するリンクは、「*CICS Transaction Server for z/OS インストール・ガイド*」ですべて使用可能です。

1. CICS を MVS サブシステムとして定義します。を参照してください。
2. 必要な CICS モジュールが CICS システムに組み込まれていることを確認します。
3. DFHIRP と DFHCSVC の各モジュールを MVS リンク・パック域 (LPA) にインストールします。

次のタスク

システム間 MRO (XCF/MRO) のサポートをインストールする場合、追加の管理タスクを実行する必要があります。こうしたタスクについても、「*CICS Transaction Server for z/OS インストール・ガイド*」で説明されています。

MRO 構成後の手順

MRO サポートの構成後、MRO 接続とリソースを定義する必要があります。

1. リモート・システムへの MRO 接続を定義します。詳しくは、181 ページの『複数領域操作のリンクの定義』を参照してください。
2. ローカル CICS 領域とリモート・システムの両方でリソースを定義します。詳しくは、253 ページの『第 17 章 ローカル・リソースの定義』および 223 ページの『第 16 章 リモート・リソースの定義』を参照してください。

© Copyright IBM Corp. 1977, 2009

第 12 章 VTAM 総称リソースの構成

機能的に同等な CICS 端末専有領域 (TOR) 群を含む CICSplex では、VTAM 総称リソース機能を使用して、使用可能な TOR 間で端末セッションのバランスをとることができます。

このトピックでは、リモート・システムへの接続の定義などの作業についての知識が読者にあるものとします。リモート・システムへのリンクの定義方法については、165 ページの『第 13 章 リモート・システムへの接続の定義方法』を参照してください。

VTAM 総称リソースの概要については、34 ページの『シスプレックスにおけるワークロード・バランシング』を参照してください。

このセクションには、以下のトピックが含まれています。

- 『VTAM 総称リソースの前提条件』
- 138 ページの『VTAM 総称リソースを使用するための CICSplex の計画』
- 139 ページの『総称リソース環境における接続の定義』
- 141 ページの『VTAM 総称リソース・サポートの生成』
- 142 ページの『総称リソースへの TOR のマイグレーション』
- 144 ページの『TOR を総称リソースから除去する』
- 145 ページの『TOR を別の総称リソースへ移動する』
- 145 ページの『総称リソース間でのシスプレックス間通信の設定』
- 150 ページの『類縁性の終了』
- 154 ページの『ATI での総称リソースの使用』
- 157 ページの『ISSUE PASS コマンドの使用』
- 158 ページの『規則のチェックリスト』
- 159 ページの『特殊な事例の対処』.

VTAM 総称リソースの前提条件

VTAM 総称リソースを使用するには、ACF/VTAM バージョン 4 リリース 2 またはそれ以降の上位互換リリースが必要です。

VTAM は、次の状態でなければなりません。

- シスプレックスの一部である MVS のもとで稼働している。
- シスプレックス・カップリング・ファシリティーに接続されている。シスプレックス・カップリング・ファシリティーについては、「*MVS/ESA Setting Up a Sysplex*」(GC28-1449) を参照してください。
- シスプレックスの少なくとも 1 つの VTAM が拡張対等ネットワーク機能 (APPN) ネットワーク・ノードで、その他の VTAM が APPN エンド・ノードである。

© Copyright IBM Corp. 1977, 2009

VTAM 総称リソースを使用するための CICSplex の計画

VTAM 総称リソース機能を使用することにより、複数の CICS 領域で端末セッションのワークロード・バランスを取ることができます。

そのためには、それらの CICS 領域を単一の総称リソースにグループ化します。それぞれの領域は、その総称リソースの**メンバー**です。端末ユーザーがその総称リソースの名前 (**総称リソース名**) でログオンすると、VTAM は、そのときのセッション・ワークロードに基づいて、その端末とメンバーのどれかとの間でセッションを確立します。端末ユーザーは、自身がどのメンバーに接続されているのかは関知しません。端末ユーザーは、総称リソースのメンバーの名前 (**メンバー名**) を使ってログオンすることもできます。その場合、端末は指定されたメンバーに接続されます。

APPC 接続と LUTYPE6.1 接続の場合、ログオンの方法は端末と同じではありません。しかし、それらの場合にも、総称リソース名 (接続先のメンバーは VTAM が選択する) かメンバー名 (指定されたメンバーに接続する) を指定することによって、総称リソースへの接続を確立することができます。

CICSplex で VTAM 総称リソースを使用する場合には、次の点を考慮してください。

- どの CICS 領域を総称リソースのメンバーにするか。

  次の点に注意してください。

  – 同じ総称リソースのメンバーにするのは、端末ユーザーに対して同等の機能を提供する CICS 領域だけにする。
  – XRF を使用する CICS 領域を総称リソースのメンバーにすることはできない。
  – 端末専有領域とアプリケーション所有領域 (AOR) を含む CICSplex では、TOR と AOR が同じ総称リソース・グループのメンバーであってはならない。

- CICSplex には、総称リソースを 1 つ設定するか、それとも多数設定するか。

  異なるアプリケーションを使用するエンド・ユーザー・グループがいくつかある場合には、ユーザー・グループごとに 1 つずつ総称リソースを設定したい場合があります。この場合、同じ CICS 領域が同時に複数の総称リソースのメンバーになることはできないことに注意してください。

- 次のメンバーの間で APPC または LUTYPE6.1 [2]の接続があるか。
  – 総称リソースのメンバーとメンバー[3]
  – ある総称リソースのメンバーと別の総称リソースのメンバー
  – 総称リソースのメンバーと、総称リソースのメンバーではないシステム

  これらのどの場合でも、次の機能をいつ使用できるのかを理解することが必要です。

  – パートナー・システムの総称リソース名を指定する接続定義

2. CICS-CICS 間接続では、LUTYPE6.1 よりも APPC の使用を推奨します。

3. 総称リソースのメンバー間で LUTYPE6.1 接続を使用することはできません。

– パートナー・システムのメンバー名を指定する接続定義
– パートナー・システムの定義を提供する自動インストール

CICS 領域の命名

それぞれの CICS 領域には VTAM APPL ステートメントで定義されたネットワーク名があり、各領域はそれによって VTAM に固有に識別されます。

この名前、つまり *applid* は、APPLID システム初期設定パラメーターに指定します。ある領域が総称リソースのメンバーであれば、その applid とメンバー名は 1 つで、かつ同じです。

総称リソース (一連の CICS 領域) には総称リソース名が付けられています。CICS 領域が総称リソースのメンバーとなる場合には、その総称リソース名をその領域の GRNAME システム初期設定パラメーターに指定します。ネットワーク名と異なり、総称リソース名を VTAM に定義する必要はありません。しかし、総称リソース名は、ネットワーク名とは同じでなく、かつネットワーク内で固有でなければなりません。CICS 総称リソースの命名規則については、「*System/390 MVS シスプレックス・アプリケーションの移行*」で示されています。

総称リソースの使用を始める際には、そのメンバー領域を認識するのにそれまで使用されていた applid に対し、総称リソース名とのメンバー名がどのように関連するのかを決めなければなりません。

- TOR がいくつかある場合には、それらの TOR に対し同じ applid を引き続き使用し、総称リソースに対して新しい名前を選択することができます。その総称リソース名を使用するには、端末ログオン手順と、その総称リソース名を使用する接続定義を変更する必要があります。
- TOR が 1 つだけの場合には、その applid を総称リソース名として使用し、その領域には新しい applid を付けることができます。端末ログオン手順 (および接続定義) の変更は最も少なくなりますが、VTAM 定義、MRO を使って接続されている AOR の CONNECTION 定義、および古い applid を指定する RACF® プロファイルの変更が必要になります。

総称リソースと XRF

XRF は VTAM 総称リソースと一緒には使用できないので、「特定」および「総称」の CICS applid の概念は、総称リソース・グループのメンバーである領域にとって意味はありません。それぞれの総称リソース・メンバーには applid が 1 つだけあります。

総称および特定の CICS applid、VTAM APPL ステートメント、および VTAM 総称リソース名の関連に関する詳しい説明については、208 ページの『XRF のための総称 applid と特定 applid』を参照してください。

総称リソース環境における接続の定義

VTAM 総称リソース機能を使用することにより、APPC 接続や LUTYPE6.1 接続におけるセッションのワークロード・バランスを取ることができます。

接続は端末セッションとは次の点で異なります。

- 1 つの接続で複数のセッションが可能。VTAM の総称リソース・サポートによって、依存要素、つまり**アフィニティー**が作成されます。このため、最初のセッションが確立されると、同じ総称リソースとのそれ以後のセッションは、最初のセッションと同じメンバーに対して行われます。
- 接続のどちらの側でも (原則として) 最初のセッションを確立できる。総称リソース環境において接続をどのように定義すべきかは、どちらの側が (実際に) 最初のセッションを開始するかによる。
- 失敗した接続で再同期化を必要とする場合には、同じメンバーの間で再び接続を確立しなければならない。VTAM では、類縁性を使って再接続が正しく行われます。

接続の定義

総称リソースへの接続を定義する場合、DEFINE CONNECTION の NETNAME オプションには、以下の 2 つの方法で指定することができます。

このタスクについて

1. 総称リソース・メンバーの名前 (applid) を使用する。このタイプの接続を**メンバー名接続**といいます。
2. 総称リソースの名前を使用する。このタイプの接続を**総称リソース名接続**といいます。

総称リソースへの接続を定義する場合には、正しい選択を行う必要があります。

- CICS がメンバー名定義を使って接続を開始すると、VTAM は、指定されたメンバーとセッションを確立します。
- CICS が総称リソース名接続を使って接続を開始すると、VTAM は、その総称リソースのメンバーの 1 つと接続を確立します。どのメンバーが選択されるかは、類縁性が存在するかどうかと、VTAM のセッション・バランシング・アルゴリズムによります。

CICS Transaction Server for z/OS の総称リソース・メンバーが接続で BIND 要求を送信する場合、その要求には、送信側の総称リソース名とそのメンバー名が入ります。パートナーも CICS TS for z/OS 総称リソースであれば、両方の名前を区別することができます。他の CICS システムは、総称リソース名をバインドからとり、それと一致する接続定義を探します。

それ自身は CICS TS for z/OS 総称リソースのメンバーでない LUtype 6 が、メンバー名を使って総称リソースと正しく接続できるのは、その総称リソース・メンバーがセッションを開始することがない場合に限られます。このようなことは通常ありませんので、CICS TS for z/OS 総称リソース・メンバーでないシステムから総称リソースへ接続する場合には、総称リソース名を使用するようにしてください。

GR メンバーと非 GR メンバー間での接続の定義

総称リソース・メンバーは、別の LUtype 6 への接続を開始する (つまり、最初の BIND を送る) 場合、パートナーに対してそれ自身が総称リソース名で識別されるようにします。パートナーが開始するセッションも、この接続を開始した LU の総称リソース名を使用しなければなりません。

同じ総称リソース内のメンバー間での接続の定義

同じ総称リソース内のメンバー間での接続を定義したい場合があります。この場合には、パートナーのメンバー名 (その総称リソース名ではなく) を CONNECTION 定義の NETNAME オプションに必ず指定する必要があります。

CICS TS for z/OS 総称リソース間での接続の定義

CICS TS for z/OS 総称リソースが 2 つある場合、それらの間での可能な接続ごとにメンバー名接続を定義しインストールする必要はありません。

つまり、パートナー総称リソースへの接続を開始する可能性のあるメンバーごとに総称リソース名接続を 1 つ定義しインストールすることができます。それによって CICS が必要に応じてメンバー名接続を自動インストールします。

接続を開始しない唯一の接続定義で CICS 領域に必要なものに、自動インストールのテンプレートとして使用される定義があります。総称リソース名接続がインストールされていると、それがテンプレートとして使用されるので、この目的のために総称リソース名接続を定義することをお勧めします。

VTAM 総称リソース・サポートの生成

CICS TOR についての VTAM 総称リソース・サポートを生成するには、次の手順を実行する必要があります。

このタスクについて

1. GRNAME システム初期設定パラメーターを使って総称リソース名を定義する。CICS はこの名前で VTAM に登録します。CICS の命名規則に従い、この名前は #、@、$ のいずれかを埋め込み文字として使用して 8 文字で指定してください。

   以下に例を挙げます。

   ```
   GRNAME=CICSH###
   ```

   GRNAME システム初期設定パラメーターの詳細については、「*CICS System Definition Guide*」の GRNAMEを参照してください。CICS の命名規則については、「*S/390 MVS シスプレックス アプリケーションのマイグレーション*」を参照してください。
2. APPL ステートメントを使って、参加する各 TOR の属性を VTAM に定義する。個々の APPL ステートメントには、同一の属性を指定しなければなりません。各 APPL ステートメントに指定する名前は固有のものでなければなりません。個々の TOR は、総称リソース・グループにおいてこの名前で識別されます。
3. 個々の端末専有領域を総称リソースのメンバーとして登録する前に、その領域を正しくシャットダウンする。「正しく」とは、CICS を CEMT PERFORM SHUTDOWN NOSDTRAN コマンドでシャットダウンすることを意味します。CEMT PERFORM SHUTDOWN IMMEDIATE では十分ではなく、CICS が失敗したあとのコールド・スタートでも不十分です。シャットダウン補助トランザクションによって強制的に VTAM がクローズされたり、即時シャットダウンが行われたりすることのないように、NOSDTRAN を指定する必要があります (デフ

ォルトのシャットダウン補助トランザクション DFHCESD については、「*CICS Operations and Utilities Guide*」の シャットダウン補助プログラム (DFHCESD) (Shutdown assist program (DFHCESD))に説明があります。)

CICS が、総称リソースのメンバーとして登録される前に正しくシャットダウンされていない と、VTAM は (持続セッションがあるために) その登録に失敗し、戻りコード・フィードバック (RTNCD-FDB2) として X'14'、X'86' を発行することがあります (VTAM RTNCD-FDB2 については、「*OS/390 eNetwork Communications Server: SNA Programming*」を参照してください)。これを訂正するには、(同じ APPLID で) CICS を再始動し、CEMT PERFORM SHUTDOWN NOSDTRAN コマンドでそれを正しくシャットダウンしてください。あるいは、アフィニティーを終了するバッチ・プログラムを作成しているなら (151 ページの『アフィニティーを終わらせるバッチ・プログラムの作成』を参照)、それを使って同じ効果が得られます。 151 ページの『アフィニティーを終わらせるバッチ・プログラムの作成』に示すスケルトン・プログラムは、その処理の一部として、元の VTAM ACB を元の APPLID で開き、持続セッションがあればアンバインドし、その ACB を閉じます。

注:

1. CICSplex に別個の複数の端末専有領域とアプリケーション所有領域がある場合には、これらの TOR と AOR を同じ総称リソース・グループに入れないでください。
2. VTAM 総称リソースと XRF を一緒に使用することはできません。XRF のシステム初期設定パラメーターに「YES」を指定すると、GRNAME に値を指定しても無視されます。
3. GRNAME に有効な総称リソース名を指定する場合には、APPLID システム初期設定パラメーターに *name1* だけを指定してください。(*name1* と *name2* を両方とも APPLID パラメーターに指定すると、CICS は、*name1* を無視し、VTAM applid として *name2* を使用します。)

VTAM 総称リソース・サポートの生成についての詳細は、「*OS/390 eNetwork Communications Server: SNA Network Implementation*」を参照してください。

総称リソースへの TOR のマイグレーション

この項では、TOR を CICS Transaction Server for z/OS 総称リソースのメンバーへマイグレーションする場合、既存の端末と接続をどのように管理するかについて説明します。

2 つの CICS TS for z/OS 総称リソース間で接続を確立する方法については、145 ページの『総称リソース間でのシスプレックス間通信の設定』で別に説明します。

注: ここでは、「端末専有領域」は、端末を所有し、総称リソースのメンバーとなり得る、任意の CICS 領域を表します。

推奨される方法

分かりやすくするために、最初にメンバーが 1 つだけの総称リソースを作成します。単一メンバーの総称リソースが正しく機能するまでは、メンバーを追加しないでください。

総称リソースのすべてのメンバーは機能的に同等なはずですから、最初のメンバーにならって追加のメンバーを作成します。(この方法を選択すべきでない状況については、あとで説明します。)

TOR を総称リソースへマイグレーションする方法として 2 つの方法をお勧めします。どちらを使用するかは、LU6 接続が既にあるかどうかによります。

LU6 接続がない場合

端末専有領域への LU6 (つまり、APPC か LU6.1) 接続がない場合には、総称リソースとして新しい名前を選び、古い applid は保持することを推奨します。非 LU6 端末は applid でも総称リソース名でもログオンできますので、総称リソース名を導入しても影響ありません。

このタスクについて

次に、その総称リソース名を使うように端末を順次マイグレーションします。このあと、最初のメンバー TOR と同じようにすれば、この総称リソースを拡張することができます。

注: 機能的に同等な TOR が既にいくつかある場合には、最初のメンバーをまねるよりも、それらの applid をメンバー名として使用し、これらの既存領域を追加することによって、総称リソースを拡張する方がよい場合があります。

LU6 接続

端末専有領域に対し LU6 (APPC または LU6.1) 接続がある場合、その総称リソースの他のメンバーへの接続は除きます。総称リソースを使用してログオンすることをお勧めします。しかし、総称リソースへマイグレーションするとき、LU6 ネットワークのパートナーではそれぞれのログオン手順を変更したくないのが普通です。

このタスクについて

1 つの方法としては、既存の端末専有領域の applid を新しい総称リソース名として使用する方法があります。そのためには、新しい applid を選ぶ必要がありますので、MRO 接続のアプリケーション所有領域の CONNECTION 定義と、その古い applid を指定する RACF プロファイルを変更することが必要になります。ただし、ユーザーが許可されている APPL プロファイルを変更する必要はありません。CICS は、サインオンの妥当性検査で GRNAME を APPL 名として RACF に渡しますが、その古い applid が GRNAME であるためです。お勧めするマイグレーション・ステップは次のとおりです。

1. CICSplex を単一端末専有領域で構成する。
2. その端末専有領域の現行 applid を総称リソース名とする。
3. 現行 applid を新しい値に変更する。
4. MRO パートナーの CONNECTION 定義を変更して、その端末専有領域の新しい applid を使用するようにする。
5. 古い applid を指定する RACF プロファイルを変更する。
6. CICSplex を再始動する。

この時点で、

- 非 LU6 端末は、(VTAM 総称リソースが使用されるようになったことを意識せずに) 古い名前を使ってログオンすることができます。その総称リソース群には TOR が 1 つしかないので、それらの端末はもちろん前と同じものに接続されます。
- LU6 接続のログオンでは古い名前が使用されます。(したがって、それらの接続は総称リソース名によるべきである、という推奨方法にかなっています)。

7. 同じ総称リソース名と AOR 群に対する同じ接続性を使って、前の端末専有領域を複製した新しい端末専有領域をインストールする。

   この時点で、
   - 自動インストールされた非 LU6 端末でセッション・バランシングが利用できるようになります。
   - 自動インストールされた APPC 同期レベル 1 接続でセッション・バランシングが利用できるようになります。
   - 類縁性のために、既存の LU6.1 と APPC 同期レベル 2 の接続は、(総称リソース名によって) 引き続き元の端末専有領域に接続されます。
   - 自動インストールでない端末と接続、およびアウトバウンド要求に対して使用される LU6 接続には、特別な考慮事項があります。これについては、159 ページの『特殊な事例の対処』で説明します。

TOR を総称リソースから除去する

領域を総称リソースから除去するには、いくつかの方法があります。

このタスクについて

- SET VTAM CLOSED コマンドを出して、その VTAM ACB をクローズする。
- CICS をシャットダウンする。その領域を永続的に除去するには、CICS を再始動する前に、その総称リソース名を GRNAME システム初期設定パラメーターから除去しなければなりません。
- SET VTAM DEREGISTERED コマンドを出して、その領域を*動的に* 除去する。つまり、VTAM ACB のクローズも、CICS のシャットダウンも行いません。この方法は、例えば TOR に小規模な変更を行う場合には便利かもしれません。

  TOR を総称リソースから動的に除去すると、ログオンされている端末は、ログオフされ、再びログオンされているうちに、その総称リソースの残りのメンバーに次第に宛先変更されます。

  CICS をその総称リソースに再び登録するには、その VTAM ACB をクローズし、再びオープンする必要があります。

  SET VTAM DEREGISTERED コマンドの詳細については、「*CICS System Programming Reference*」の SET VTAMおよび「*CICS Supplied Transactions*」の CEMT SET VTAMを参照してください。

重要:

領域を総称リソースから除去する場合には、次の注意が必要です。

- その領域が所有する類縁性を終了する必要があります。そうしないと、VTAM において、影響を受ける APPC パートナーと LU6.1 パートナーはその総称リソースの別のメンバーに接続できません。 150 ページの『類縁性の終了』を参照してください。
- 除去された領域は、それを総称リソース名で認識しているパートナーへの接続を獲得しようとしないでください。ただし、そのパートナーが除去された領域に対する類縁性を終了している場合を除きます。

TOR を別の総称リソースへ移動する

領域をある総称リソースから別の総称リソースへ移動するには、次の手順を実行する必要があります。

このタスクについて

1. それが所有する類縁性を終了する。 150 ページの『類縁性の終了』を参照してください。
2. それを正しくシャットダウンする。 141 ページの『VTAM 総称リソース・サポートの生成』を参照してください。

   CICS が、新しい総称リソースのメンバーとして登録される前に、正しくシャットダウンされていない と、VTAM が、それを登録できず、RTNCD-FDB2 として X'14'、X'86' を発行することがあります。これを訂正するには、*元の* GRNAME と APPLID で CICS を再始動し、CEMT PERFORM SHUTDOWN NOSDTRAN コマンドでそれを正しくシャットダウンしてください。あるいは、類縁性を終了するバッチ・プログラムを作成しているなら、それを使えば同じ効果が得られます。 151 ページの『アフィニティーを終わらせるバッチ・プログラムの作成』で説明するスケルトン・プログラムは、その処理の一部として、元の VTAM ACB を元の GRNAME で開き、持続セッションがあればアンバインドし、その ACB を閉じます。
3. 代替総称リソースの名前を GRNAME システム初期設定パラメーターに指定し、CICS を再始動する。

総称リソース間でのシスプレックス間通信の設定

この項では、複数のパートナー・シスプレックスにおける CICS Transaction Server for z/OS の総称リソース間での通信について説明します。CICS TS for z/OS の総称リソース間のリンクには、APPC 並列セッション接続を使用しなければなりません。

CICS TS for z/OS 総称リソース間での接続の確立

SYSPLEXL と SYSPLEXR という 2 つのシスプレックスがあり、それぞれに CICSL、CICSR という CICS TS for z/OS 総称リソース・グループがあるものとします。

このタスクについて

これは、 147 ページの図 36 に示されています。CICSL と CICSR 間の接続を確立するステップは、次のとおりです。

1. CICSR への接続を開始する CICSL のメンバーごとに、APPC 並列セッション接続を静的に定義およびインストールします (NETNAME は CICSR の総称リソース名です)。つまり、*総称リソース名接続* を定義します。同じように、CICSL への接続を開始する CICSR のメンバーごとに、APPC 並列セッション接続を静的に定義およびインストールします (NETNAME は CICSL の総称リソース名です)。

   注: 総称リソース名接続以外には、定義済みの接続をインストールしないでください。

   CICSL の任意のメンバーが CICSR への接続を最初に獲得しようとするとき (または、その逆)、総称リソース名接続を使用します。
2. VTAM がバインド要求を送信する先の CICSR メンバーは、CICSL の総称リソース名接続の定義を探します。(それがない場合には、接続を自動インストールする場合の通常の規則に従って、その定義が自動インストールされます。)
3. VTAM がたまたまルーティングした CICSR の同じメンバーへの、CICSL の異なるメンバーから行われるそれ以後の接続は、CICSL のメンバー名を NETNAME として、CICSR メンバーに自動インストールされます。つまり、CICS はメンバー*名接続* を自動インストールします。同じように、CICSR の異なるメンバーから CICSL の同じメンバーへのそれ以後の接続は、その CICSR のメンバー名を NETNAME として、その CICSL メンバーに自動インストールされます。『例』は、この状態を示しています。

   後から行われるこれらの接続の自動接続に使用されるテンプレートには、インストール済みの任意の接続が使用できます。CICS は、総称リソース名接続をデフォルト・テンプレートとして使用します。

   メンバー名接続に対しデフォルト・テンプレート以外のものを使用すると、これらの接続でのセッションの使用はパートナーによって開始されることに注意してください。したがって、MAXIMUM オプションの定義には競合勝者を含めないようにしてください。[4] (これは、メンバー名接続が自動インストールされるシステムのアプリケーションにはメンバー名が認識されませんので便利です。これらのアプリケーションは、アウトバウンド要求に GR 名を使用します。したがって、アウトバウンド要求にはメンバー名接続が使用されないため、競合勝者として定義されたセッションは必要はありません。パートナー・システムに競合勝者としてすべてのセッションをもたせることによって、競合敗者セッションのための送信権を要求するオーバーヘッドが避けられます。)

   テンプレートは、CONNECTION と SESSIONS によって定義される通常のインストール済み接続です。この接続は、テンプレートとしてだけ使用することもできますし、実際の接続として使用することもできます。テンプレートは、さらに接続を自動インストールするとき、そのモデルとして使用されます。

例

次に、CICS TS for z/OS 総称リソース間で接続を確立する例を示します。

---

4. DEFINE SESSIONS の MAXIMUM オプションについては、188 ページの『APPC セッション・グループの定義』を参照してください。

図 36 から 149 ページの図 39 において、それぞれの総称リソースは、接続を開始する際、パートナー・シスプレックスの総称リソース名を使用します。すべての総称リソース・メンバーは接続を開始することができます。つまり、それらはどれも総称リソース名接続 (NETNAME がパートナー・シスプレックスの総称リソース名である定義済み接続項目) をもっています。開始される接続は、APPC 並列セッション同期レベル 2 のリンクです。

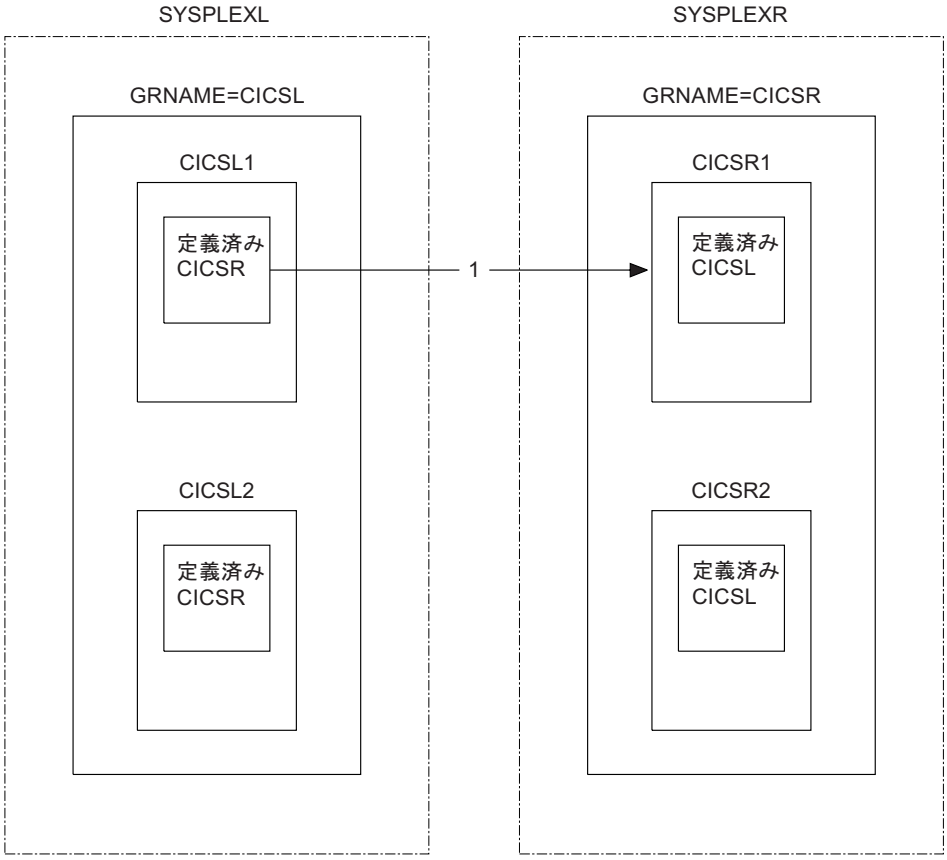


図 36. この図は 2 つのシスプレックス SYSPLEXL と SYSPLEXR を示しています: それぞれのシスプレックスには CICS 総称リソース・グループがあります。CICSL グループのメンバー CICSL1 が、SYSPLEXR にある CICSR グループのメンバーへの接続を獲得しようとしています。

図 36 において、CICSL1 から CICSR へ流れる最初のバインドは、負荷が最も少ないと VTAM が判断する CICSR のメンバーにルーティングされます。この例では、CICSR1 へ行きます。CICSL1 と CICSR1 にある総称リソース名 CICSR と CICSL の定義済み接続が使用されます。

SYSPLEXL と SYSPLEXR に*類縁性* が作られ、CICSL1 と CICSR1 が関連付けられます。これらのアフィニティーを終わらせる場合は、明示的でも明示的でなくても構いません (150 ページの『類縁性の終了』 および 334 ページの『APPC 接続静止処理』 を参照)。類縁性が終了するまで、CICSL1 が CICSR に再接続しようとするたびに、VTAM はその要求を CICSR1 にルーティングします。また、CICSR1 が CICSL に再接続しようとするたびに、VTAM はその要求を CICSL1 にルーティングします。

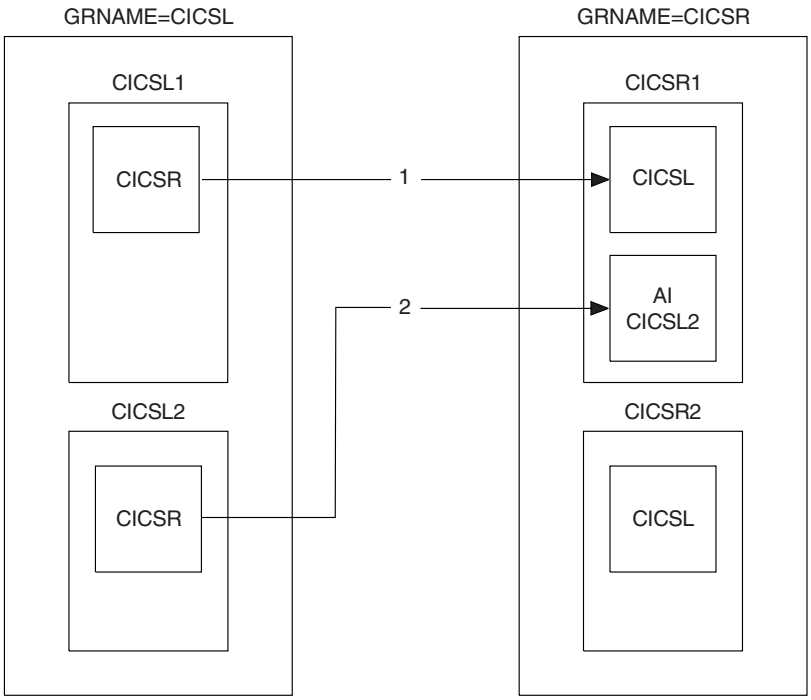


図 37. 2 番目の流れ、CICSL2-CICSR

図 37 は、CICSL2 から CICSR へのバインドの流れを示しています。この例で、VTAM はこの流れを再び CICSR1 へルーティングしていますが、CICSR の他のメンバーへルーティングすることも可能です。

CICSL2 にある CICSR の定義済み接続が使用されます。CICSR1 は、CICSL の接続項目を探します。これは既に使用中ですので、メンバー名 CICSL2 を使って、新しい接続が自動インストールされます。

SYSPLEXL と SYSPLEXR に類縁性が作られ、CICSL2 と CICSR1 が関連付けられます。これらの類縁性を終わらせる場合は、明示的でも明示的でなくても構いません。

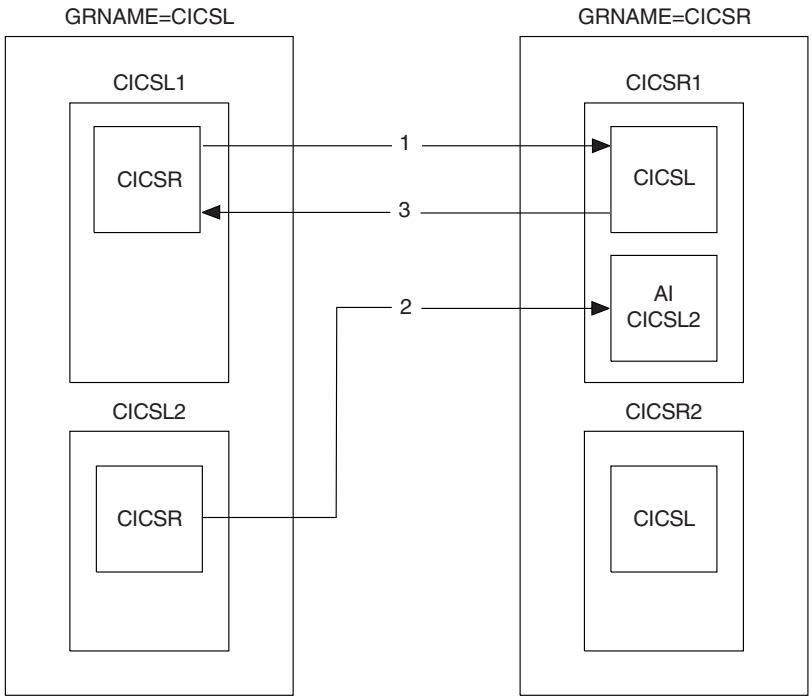


図 38. 3 番目の流れ、CICSR1-CICSL

図 38 は、今度は 3 番目の CICSR1 から CICSL への流れを示しています。類縁性が既に存在しますので、その流れは CICSL1 へ強制されます。

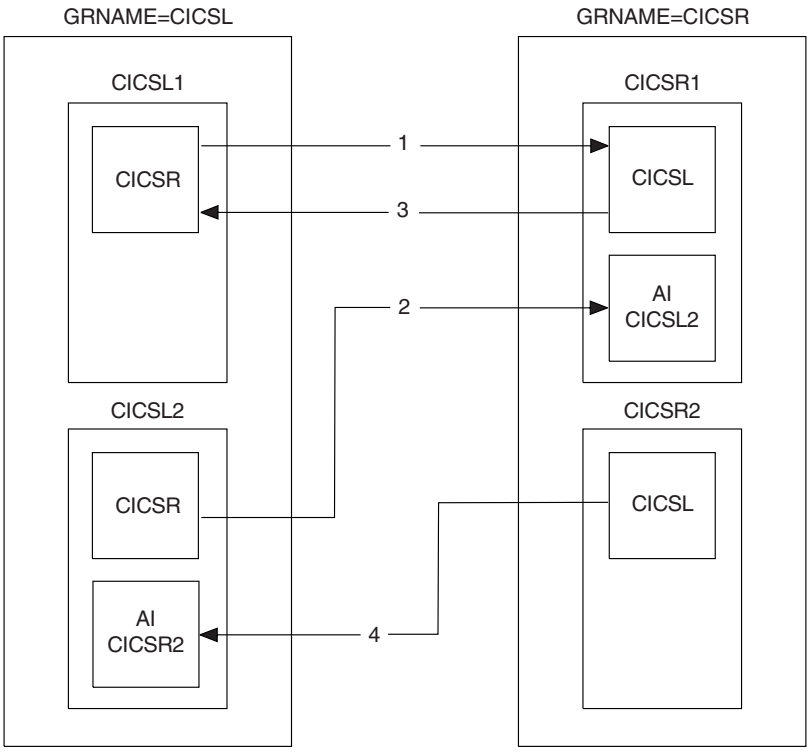


図 39. 4 番目の流れ、CICSR2-CICSL

149 ページの図 39 は、今度は 4 番目の CICSR2 から CICSL への流れを示しています。この流れは CICSL のどのメンバーへ行くことも可能ですが、この例の場合 VTAM は CICSL2 へルーティングしています。

CICSR2 にある CICSL の定義済み接続項目は使用されていませんので、今回、それが使用されます。CICSL2 は、CICSR の定義済み接続項目を探します。その項目は使用中なので、CICSR2 の項目が自動インストールされます。

SYSPLEXL と SYSPLEXR に類縁性が作られ、CICSL2 と CICSR2 が関連付けられます。これらの類縁性を終わらせる場合は、明示的でも明示的でなくても構いません。

類縁性の終了

セッションが総称リソース・メンバーとの間に確立されると、VTAM は、その総称リソース・メンバーとパートナー LU との間に類縁性と呼ばれる関係を設定します。それによって、以降の流れをどこにルーティングすべきかが分かります。

ほとんどの場合、VTAM は、そのセッションのすべてのアクティビティーが終わると類縁性を終わらせます。しかし、セッションのタイプによっては、VTAM は、再同期データがあるものと見なし、類縁性の終了を CICS に任せます。影響を受けるセッションは、次のとおりです。

- APPC 同期レベル 2 セッション
- 限定リソース・サポートを使用する APPC セッション
- LU6.1 セッション

VTAM 用語では、CICS 総称リソース・メンバーが類縁性を「所有し」、それを終わらせる責任があります。類縁性は、接続が削除されたり、CICS が初期始動やコールド・スタートを実行したあとでも存続します。*2 つの総称リソースの間の接続では、両方のパートナーが類縁性を所有するので、それぞれの類縁性を終了させなければなりません。*CICS TS OS/390 バージョン 1.3 以降の領域間の APPC 接続は、APPC 接続静止プロトコルによって自動的に行われます (334 ページの『APPC 接続静止処理』 を参照)。その他の接続では、類縁性は明示的に終了しなければなりません。

CICS には、類縁性を明示的に終了させるためのコマンドが用意されています。

- インストール済みの接続定義がある場合には、SET CONNECTION ENDAFFINITY が使用できます。
- 自動インストールされた接続が存在する間だけでなく、それが削除されたあとでは、PERFORM ENDAFFINITY が使用できます。リモート・システムの NETNAME (および、接続が既に削除されている場合には、NETID) を指定する必要があります。NETNAME は、そのリモート・システムが VTAM によって認識されている名前です。(リモート・システムも総称リソースの場合は、NETNAME は、その接続が総称リソース名を使って定義されている場合でも常にメンバー名であることに注意してください。)

これらのコマンドは、LU6.1 接続と APPC 接続だけに有効です。この接続は (存在する場合) サービス不能になっていて、そのリカバリー状況 (INQUIRE CONNECTION コマンドの RECOVSTATUS オプションによって示される) は

NORECOVDATA でなければなりません。CICS によって所有されている類縁性だけが CICS によって終了できることに注意してください。

CICS には、接続に類縁性が存在するのかどうかは分かりません。明示的に終了させる必要がある類縁性が作成された可能性がある場合には、参考のためにメッセージ DFHZC0177 が出されます。このメッセージには、PERFORM ENDAFFINITY コマンドで使用する NETNAME と NETID が示されます。

メッセージ DFHZC0177 を受け取った場合には、SNA D NET,GRAFFIN コマンドを使用することにより、明示的に終了しなければならない類縁性が実際に存在しているかどうかを調べることができます。このコマンドが生成するメッセージ IST1706 および IST1707 には、類縁性の有無を知るために必要な情報が含まれています。また、「*MVS/ESA Version 5 Interactive Problem Control System (IPCS) Commands*」(GC28-1491) には、VTAM ISTGENERIC データ域のダンプの作成方法が記載されています。このダンプの SPTE レコードを見ると、どの類縁性が存在するかが分かります。例えば、ダンプを開始するには、次のコマンドを入力します。

```
DUMP COMM=(title)
```

次のように応答します。

```
r xx ,STRLIST=(STRNAME=ISTGENERIC,
                  ACC=NOLIMIT,(LNUM=ALL,ADJ=CAP,EDATA=SER))
```

ダンプを見るには、次のコマンドを入力します。

```
STRDATA DETAIL ALLSTRS ALLDATA
```

類縁性が存在しないために、類縁性を終わらせる要求が VTAM によって拒否されると、メッセージ DFHZC0181 が出されます。これはおそらく、指定した NETNAME か NETID が正しくないか、ユーザー (または CICS) が想定していた類縁性の存在が正しくないことを意味します。

アフィニティーを終了すべき場合

シスプレックスを再構成する場合には、類縁性を終わらせる必要があります。

例えば、次のいずれかを行う場合には、その前に関連する類縁性を終わらせ**なければなりません**。

- 総称リソースの名前を変更する。
- 総称リソース名接続をメンバー名接続に変更する。
- 並列セッション接続を単一セッション接続に変更する。
- 総称リソースからシステムを除去する。システムを総称リソースから除去し、その類縁性を終わらせないと、VTAM は、それが依然としてその総称リソースのメンバーであるものと見なします。

注: 総称リソース間の接続の場合には、両方の総称リソースによって所有されている類縁性を終わらせる必要があります。

アフィニティーを終わらせるバッチ・プログラムの作成

類縁性を所有する総称リソース・メンバーが失敗し、リカバリーできないときは、それらの類縁性を終わらせる必要があります。

このような場合には、SET CONNECTION ENDAFFINITY コマンドや PERFORM ENDAFFINITY コマンドは使用できません。その代わりに、バッチ・プログラムを使って、失敗したメンバーによって所有されている類縁性をクリアすることができます。この項では、そのようなバッチ・プログラムの作成方法について示します。このプログラムはアセンブラー言語で作成する必要があります。

注: 「*MVS/ESA Version 5 Interactive Problem Control System (IPCS) Commands*」に記載されているダンプ技法を使用して、障害のある総称リソース・メンバーが所有するアフィニティーを見つることができます。

重要:

この技法を使用するのは、障害のある CICS システムを再始動できない場合だけにしてください。

プログラム入力

次の入力パラメーターをプログラムに指定する必要があります。

- 失敗したシステムの (総称リソース・グループの) メンバー名
- 失敗したシステムの総称リソース名
- パートナー・システムの APPLID
- パートナー・システムの NETID

プログラム出力

このプログラムは、VTAM CHANGE OPTCD=ENDAFFIN マクロを使って、類縁性を終わらせます。

このプログラムは、VTAM CHANGE OPTCD=ENDAFFIN マクロを使って、類縁性を終わらせます。プログラムが使用するこのマクロやその他の VTAM マクロ呼び出しが正常であったか、失敗であったかを示すレポートがおそらく必要になります。RTNCD/FDB2 の値の意味については、「*OS/390 eNetwork Communications Server: SNA Programming*」を参照してください。

処理

プログラムで、次の処理を実行する必要があります。

このタスクについて

1. 次のストレージを予約する。
   - 失敗したシスプレックス・メンバーの ACB

     ```
     acb-name ACB AM=VTAM,
                  PARMS=(PERSIST=YES)
     ```

     この例では、持続セッションが使用されているものとします。
   - RPL。これは VTAM マクロが必要とします。

     ```
     rpl-name RPL   AM=VTAM,OPTCD=(SYN)
     ```
   - NIB。これは CHANGE OPTCD=ENDAFFIN マクロが必要とします。

     ```
     nib-name NIB
     ```
2. 類縁性を所有するメンバーの ACB に対し VTAM OPEN コマンドを出し、このメンバーの入力 APPLID を渡す。

3. 持続するセッションがある場合には、VTAM SENDCMD マクロでそれらを終わらせます。(持続セッションを使用していなければ、これは不要です。)
   a. 次のコマンドをストレージ内のある区域へ移動する。この例で *applid1* は失敗したメンバーのメンバー名、*applid2* はパートナー・システムの APPLID です。

      ```
      'VARY NET,TERM,LU1=applid1,LU2=applid2,TYPE=FORCE,SCOPE=ALL'
      ```

   b. 次の例のようにして SENDCMD マクロを出す。この例では、次のようになっています。
      - *rpl-name* は RPL の名前です。
      - *acb-name* は失敗したシスプレックス・メンバーの ACB です。
      - *output-area* は VARY コマンドを保持するストレージ内の区域です。
      - *command-length* はコマンドの長さです。

      ```
      SENDCMD  RPL=rpl-name,
         ACB=acb-name,
         AREA=output-area,
         RECLEN=command-length,
         OPTCD=(SYN)
      ```

4. VTAM RCVCMD マクロを使って VTAM のメッセージを受信する。VARY コマンドが正しく実行されたことを確認するために、RCVCMD は、SENDCMD のあとに 3 回出す必要があることに注意してください。次の例で、
   - *rpl-name* と *acb-name* は上で示したとおりです。
   - *input-area* は、メッセージを受信するストレージの区域です。
   - *receive_length* は受信するデータの長さです。

   ```
   RCVCMD  RPL=rpl-name,
      ACB=acb-name,
      AREA=input-area,
      AREALEN=receive-length,
      OPTCD=(SYN,TRUNC)
   ```

5. VTAM からのすべての出力を確実に受信するために、このコマンドをさらに 2 回出す。
6. VTAM CHANGE OPTCD=ENDAFFIN マクロを出して、その類縁性を終わらせる。このマクロを出す前に、NIB のフィールドを次のように初期設定する必要があります。
   - NIBSYM をパートナー・システムの APPLID に設定する。
   - NIBGENN を失敗したシステムの総称リソース名に設定する。
   - NIBNET をパートナー・システムの NETID に設定する。

   ```
   CHANGE  RPL=rpl-name,
           ACB=acb-name,
           NIB=nib-name,
           OPTCD=(SYN,ENDAFFIN)
   ```

7. その ACB に対し VTAM CLOSE コマンドを出す。

タスクの結果

プログラミングのための注意事項:

1. VTAM コマンドは、出口が使用されないように、同期していなければなりません (OPTCD=SYN)。

2. 稼働中の CICS の APPLID に対してこのプログラムを実行**しない**ように注意してください。VTAM 持続セッションが使用されている場合、これを実行すると *強制的な引き継ぎ* が起こります。つまり、その APPLID に属するセッションの制御をそのプログラムが行うことになります。

ENDAFFINITY プログラムを実行依頼するための JCL

これは、ENDAFFINITY プログラムを実行依頼するための JCL (ジョブ制御言語) の例です。

```
//JOBNAME   JOB 1,userid,
// NOTIFY=userid,CLASS=n,MSGLEVEL=(n,n),MSGCLASS=n,REGION=1024K
//*
//JOBLIB    DD DSN=loadlib-name,DISP=SHR
//*
//*****************************************************************
//* PARM='FAILED_APPLID,FAILED_GENERIC,PARTNER_NETID,PARTNER_APPLID'
//*****************************************************************
//*
//RUN    EXEC PGM=ENDAFFIN,PARM='parm1,parm2,parm3,parm4'
//*
//REPORT    DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//
```

図 40. ENDAFFINITY プログラムを実行依頼するための JCL 例

ATI での総称リソースの使用

自動トランザクション開始 (ATI) とは、端末エンド・ユーザーがトランザクション名を入力するのではなく、CICS システムで内部的に行われる要求によってトランザクションが開始されるプロセスです。

例えば、これは、アプリケーション・プログラムが EXEC CICS START コマンドを出したり、一時データ・キューのトリガー・レベルに達したりしたときに起こります。開始されるトランザクションは端末と関連付けられるのが普通です。この端末は、そのトランザクションが実行される領域によって所有されている場合もあれば、されていない場合もあります。

ATI については、79 ページの『ATI によって開始されたトランザクションの従来のルーティング』に説明があります。特に、79 ページの『ATI によって開始されたトランザクションの従来のルーティング』では、CICS がどのようにして「端末未認識」のグローバル・ユーザー出口 XICTENF と XALTENF を呼び出して、端末が AOR に定義されていない状態を処理するかが説明されています。

アプリケーション所有領域 (AOR) において、TOR にログオンされている端末に対し自動トランザクション開始 (ATI) 要求が出されると、CICS は、その AOR にある端末定義を使って、その要求のシップ先の TOR を判別します。AOR にその端末の定義がない場合は、「端末未認識」のグローバル・ユーザー出口 (XICTENF と XALTENF) を使って、その TOR の名前を指定できることがあります。

しかし、ユーザーが (総称リソース名を使って) 総称リソースにログオンすると、VTAM は、この端末をその総称リソースのどれかの領域に接続します。そのあとユーザーがログオフし、再びログオンすると、VTAM は、この端末を同じ領域に接続することもありますし、異なる領域に接続することもあります。この状態では、AOR の端末定義はその端末の正しい場所を反映していない場合があります。そして、端末未認識の出口プログラムには、ATI 要求の正しい宛先を知る方法はありません。

CICS は、この端末がどこにログオンされているかを示す VTAM の情報を使って、ATI 要求を正しい TOR へシップすることによって、この問題を解決します。

1. まず ATI 要求は、リモート端末定義に指定されている (または端末未認識出口によって指定された) TOR にシップされます (これを「最初の選択 TOR」と呼ぶことにします)。端末が最初の選択 TOR にログオンされていれば、その ATI 要求は正しく完了します。
2. 最初の選択 TOR 上に端末が見つからない場合は、その TOR は、端末がログオンされている総称リソース・メンバーの applid を VTAM から入手します。端末が総称リソース・グループ内のいずれの applid にもログオンしていない場合、ATI 要求は失敗します。

   最初の選択 TOR 上に端末が見つかった場合でも、ログオンされていなければ、その TOR は、端末がログオンされている総称リソース・メンバーの applid を VTAM から入手します。端末が総称リソース・グループ内のいずれの applid にもログオンされていない場合、ATI 要求は最初の選択 TOR 上でスケジュール化されます。端末が総称リソース・グループ内の異なる applid にログオンされている場合、この情報が AOR に渡され、ATI 要求が正しい TOR に送達されます。
3. 最初の選択 TOR が使用できないと (その場合、照会が可能)、AOR は、端末の場所を VTAM から入手します。次のすべての条件に当てはまるとき、照会が可能です。
   - AOR の VTAM がバージョン 4.2 かそれ以降である (つまり、総称リソースをサポートする)。
   - その AOR は、VTAM システム初期設定パラメーターの設定が「YES」で開始された。
   - 端末がログオンされる VTAM 総称リソース名が AOR に対して既知である。この情報は、リモート端末を表すスケルトン TCTTE から取得されます。最初の選択 TOR 名が端末が不明なユーザー出口から提供される場合、このような照会を行うことはできません。スケルトン TCTTE に検出される VTAM 総称リソース名に、端末がログオンされない場合は、照会が失敗することに注意してください。

   AOR と TOR が別のネットワークにあると、照会は失敗します。

   照会が正常に終われば、ATI 要求は、端末がログオンされている TOR にシップされます。

VTAM は、端末をその CICS 端末 ID (TERMID) ではなく、そのネット名で認識します。START が出されたときに AOR に端末定義があれば、CICS はその定義からネット名を入手します。それがない場合には、端末未認識出口プログラムが次の情報を戻す必要があります。

- 端末を見つけるために VTAM が使用するネット名
- アクティブになっているであろう総称リソースの任意のメンバーへの接続の名前

注:

1. CICS が端末のネット名を知らない場合には、ATI 要求は最初の選択 TOR へシップされ、その端末 ID を使ってその端末を見つけます。端末が最初の選択 TOR で検出されないと、ATI 要求は失敗します。
2. CICS は端末のネット名によって総称リソース・グループにおけるその端末の位置を見つけますので、2 回目以降のログオンで端末 ID が変わっても (例えば、自動インストール・ユーザー・プログラムがネット名と端末 ID との間のマッピングを一貫性をもって行っていない場合)、ATI 要求は正しく行われます。
3. この項で説明した ATI サポートは、総称名を使って総称リソースにログオンする端末だけに適用されます。ユーザーがメンバー名を使って TOR にログオンする場合には、CICS は、端末がどの TOR 端末に接続されているかの情報を VTAM から入手しようとはしません。
4. この項で説明した ATI サポートは、APPC 接続に対する ATI には適用されません。
5. TOR は、自動インストール、または CEDA 定義の端末定義を使用することができます。

   AOR には、CEDA 定義のリモート端末定義を使用**しないでください**。CEDA 定義の端末が使用された場合、ATI 要求は常に最初の選択 TOR にシップされ、端末が他の TOR にログオン可能であっても、同じ VTAM 総称リソース・グループ内の異なる TOR に再度ルーティングされることはありません。

例 1:

1. ユーザーが総称リソース名 CICS を使ってログオンします。この総称リソース名は TOR 群 (TOR1 から TOR6) の名前です。TOR1 の負荷が最も軽いため、ユーザーはこの領域に接続されます。
2. このユーザーがトランザクションを実行すると、このトランザクションは AOR1 という AOR にルーティングされます。その端末定義が AOR1 にシップされます。
3. そのトランザクションが、しばらくしてから同じ端末に対し EXEC CICS START 要求を出して、別のトランザクションを開始します。2 番目のトランザクションも、最初のものと同じように AOR1 にあります。
4. 最初のトランザクションが完了したら、ユーザーはログオフします。そのあと、2 番目のトランザクションの出力を得るために再びログオンします。再び総称リソース名 CICS を使って 2 回目のログオンをすると、ユーザーは、TOR2 の負荷がその時点では最も軽いため、その領域に接続されます。
5. START 要求に指定された間隔が経過します。しかし、その端末は TOR1 にはもう定義されていません。シップされた端末定義は、タイムアウト削除メカニズムによって AOR1 からまだ削除されていません。

- **結果:**

  ユーザーの端末のシップされた定義はまだ AOR1 にあるので、AOR1 は ATI 要求を TOR1 (その定義で参照されている TOR) にシップします。その端末は TOR1 にログオンされていませんので、TOR1 は VTAM に照会し、その結果を AOR1 に戻します。次に AOR1 は、その要求を正しい TOR (TOR2) へシップします。

例 2:

1. ユーザーが総称リソース名 CICS を使ってログオンします。この総称リソース名は TOR 群 (TOR1 から TOR6) の名前です。TOR1 の負荷が最も軽いため、ユーザーはこの領域に接続されます。
2. このユーザーがトランザクションを実行すると、このトランザクションは AOR1 という AOR にルーティングされます。その端末定義が AOR1 にシップされます。
3. そのトランザクションが非同期の処理を行います。つまり、2 番目のトランザクションを開始します。そのトランザクションは別の AOR である AOR2 にあります。2 番目のトランザクションは、処理が終わると、メッセージを TOR1 のユーザー端末に送信するために、元のトランザクションを再び呼び出すことになります。
4. アプリケーションの処理中にユーザーはログオフします。そして、あとで、メッセージを得るために再びログオンします。再び総称リソース名 CICS を使って 2 回目のログオンをすると、ユーザーは、TOR2 の負荷がその時点では最も軽いため、その領域に接続されます。
5. 2 番目のトランザクションが処理を完了し、EXEC CICS START コマンドを出して、元の端末と関連させて元のトランザクションを再び呼び出します。この START 要求は AOR1 にシップされます。しかし、その端末は TOR1 にはもう定義されていません。また、シップされた端末定義は、タイムアウト削除メカニズムによって AOR1 から既に削除されています。

- **結果:**

  シップされた端末定義は AOR1 から既に削除されているので、CICS は XICTENF と XALTENF 出口を呼び出します。この出口プログラムは、次の情報を戻す必要があります。

  – ユーザーの端末のネット名
  – 現在アクティブにあるであろう総称リソースの任意のメンバーへの接続の名前

  これで CICS は、例 1 のようにして VTAM に照会し、要求を正しい TOR (TOR2) へシップすることができます。

ISSUE PASS コマンドの使用

EXEC CICS ISSUE PASS コマンドを使用することにより、端末の接続を CICS から解除し、その端末を LUNAME オプションに指定されている VTAM アプリケーションに渡すことができます。

例えば、次のコマンドを出せば、端末をこの CICS から別の端末専有領域に移すことができます。

```
EXEC CICS ISSUE PASS
LUNAME(applid)
```

ここで、applid は、その端末を移す先の TOR の applid です。

TOR が総称リソース・グループのメンバーなら、LUNAME を総称リソース名として指定すれば、端末をそのグループの任意のメンバーに移すことができます。以下に例を挙げます。

```
EXEC CICS ISSUE PASS LUNAME(grname)
```

ここで、grname は総称リソース名です。VTAM は、その総称リソースのメンバーのうち負荷が最も軽いものにその端末を移します。(ISSUE PASS コマンドを出したシステム自体が、負荷が最も軽いメンバーである場合には、VTAM は負荷が次に軽いメンバーにその端末を移します。)

ISSUE PASS LUNAME(grname) コマンドを発行するシステムが、その総称リソース名のもとに現在登録されている*唯一* の CICS である場合 (例えば、他の領域がすべてシャットダウンされている場合) でも、その ISSUE PASS コマンドが INVREQ で失敗することは**ありません**。その代わりに、その端末がログオフされ、メッセージ DFHZC3490 が CSNE ログに書き込まれます。独自のノード・エラー・プログラムを作成すれば、この状態を扱うことができます。ノード・エラー・プログラムのコーディングに関するヒントについては、「*CICS Customization Guide*」の ノード・エラー・プログラムの作成を参照してください。

CICS の総称リソース・グループ内の特定の TOR に端末を移す場合は、最初の例のコマンドのように、LUNAME をそのメンバー名 (つまり CICS APPLID) として指定する必要があります。

規則のチェックリスト

CICS が VTAM の総称リソース機能を使用する場合の規則について、そのチェックリストを次に示します。

- 総称リソース名はネットワーク内で固有でなければならない。
- CICS 領域は、同時に、総称リソースのメンバーであり、XRF パートナーであることはできない。
- 総称リソースのメンバーである CICS 領域は、1 つの総称リソース名と 1 つの applid しかもてない。
- 同じネットワーク内では、総称リソース名が VTAM applid と同じであってはならない。
- 同じ総称リソース内では、メンバー名だけを使用しなければならない。総称リソースのどのメンバーにも、その総称リソース名の定義があってはならない。
- 順序番号の再同期が必要な非 LU6 デバイスは、総称リソース名を使ってログオンできない。それらのデバイスは applid を使用しなければならないので、セッション・バランシングを利用することはできません。
- パートナーによって開始された総称リソースへの APPC 接続 (この場合は、非総称リソースが最初のバインドを送信する) は、メンバー名を使ってログオンできる。

- 総称リソース・メンバーによって開始された LU6.1 接続の場合には、そのパートナーは、メンバーをその総称リソース名で認識できなければならない。

  したがって、同じ LU6.1 パートナーを総称リソースの複数のメンバーから決してアクセスしないようにしてください。
- 総称リソース・メンバーによって開始された APPC 接続 (この場合、パートナー自身はCICS Transaction Server for z/OS 総称リソースのメンバーではない) では、パートナーは、メンバー TOR をその総称リソース名で認識できなければならない。

  したがって、そのようなパートナーを総称リソースの複数のメンバーから決してアクセスしないようにしてください。
- 同じ総称リソースに対し、APPC 総称リソース名接続と APPC メンバー名接続を両方とも静的に定義することはできない。(総称リソース名接続とメンバー名接続については、 145 ページの『CICS TS for z/OS 総称リソース間での接続の確立』を参照してください。)

  さらに、同じ総称リソースのすべてのメンバーは、同じ方式を選択しなければなりません。つまり、(パートナー総称リソースに対する静的に定義された APPC 接続の場合)、すべてがメンバー名接続を使用するか、すべてが総称リソース名接続を使用する必要があります。

特殊な事例の対処

この項では、考慮を払わなければならない場合がある特別な場合についていくつか説明します。

この情報の多くは、バックレベルのシステムへのリンク (例えば、非 CICS TS for z/OS システムへの接続を開始する場合) にのみ適用されることに注意してください。CICS TS for z/OS 総称リソース間の接続では、この情報はほとんど無視して構いません。

非自動インストール端末と接続

総称リソースのメンバーは機能的に同等でなければならないので、端末を総称リソースの特定のメンバーに事前定義しないでください。

重要:

その代わりに、自動インストールを使い、VTAM によって TOR のワークロードを動的に平衡化してください。ただし、例えば、既存の TOR を総称リソースにマイグレーションしている場合には、静的定義を使用しなければならないときがあります。

LU が特定の端末専有領域に事前に定義され、LU がその TOR の総称リソース名を使って接続を開始する (つまり、最初のバインド要求を送信する) 場合、総称リソース機能によって、その接続は「正しい」端末専有領域 (その定義がある領域) に対して実行されなければなりません。つまり、VTAM 総称リソース解決出口プログラム ISTEXCGR をインストールして、(端末専有領域の) 正しい applid が選択されるようにしなければなりません。

ただし、接続が常に端末専有領域によって開始される (例えば、START 要求によって) 場合には、これは必要ありません。

サンプルの ISTEXCGR 出口プログラムが VTAM 4.2 で提供されます。詳しくは、「*OS/390 eNetwork Communications Server: SNA Customization*」を参照してください。

アウトバウンドの LU6 接続

この項では、総称リソース・グループのメンバーである TOR からのアウトバウンド LU6 接続について説明します。ここでは、「アウトバウンド」の意味を CICSplex の外のシステムへの接続ということにします。

「ハブ」の使用

総称リソース・メンバーによって開始された LU6 接続 (この場合、パートナー自身は CICS Transaction Server for z/OS 総称リソースのメンバーではない) では、パートナーは、メンバー TOR をその総称リソース名で認識できなければならない。

したがって、この要件は、総称リソース・メンバーが次の種類の接続のどれかを開始する場合に当てはまります。

- 単一システムへの APPC 接続
- 総称リソース・メンバーではない CICSplex メンバーへの APPC 接続
- すべての LU6.1 接続

(パートナーも CICS TS for z/OS 総称リソースである場合を除いて) 総称リソース・メンバーによる LU6 パートナーへの接続の試みは、パートナーが TOR をその総称リソース名で認識できる場合にのみ成功するので、パートナーは、一度に総称リソースの 1 つのメンバーだけへの接続を受け入れることが可能になります。総称リソースの複数のメンバーから同じリモート・システムに接続しなければならない構成では、その CICSplex 内の 1 つの領域を選択して、それを**ネットワーク・ハブ**とすることができます。この場合、すべての総称リソース・メンバーは、リモート・システムのサービスに対する要求をハブを通してデイジー・チェーンします。

ネットワーク・ハブは総称リソースのメンバーであっても構いません。この場合には、VTAM 総称リソース解決出口プログラムをインストールして、こちらをその総称リソース名で認識している LU6 パートナーからの*着信* バインドをネットワーク・ハブ領域へ送る必要があります。

これに代わる解決策は、総称リソースのメンバーでは**ない**ネットワーク・ハブをもつことです。こうすると、VTAM 総称リソース解決出口プログラムは必要ありませんが、この CICSplex への接続を開始する可能性のある LU6 パートナーは、そのネットワーク・ハブ領域の applid を使ってログオンする必要があります。

161 ページの図 41 は、総称リソースのメンバーではないネットワーク・ハブを示しています。

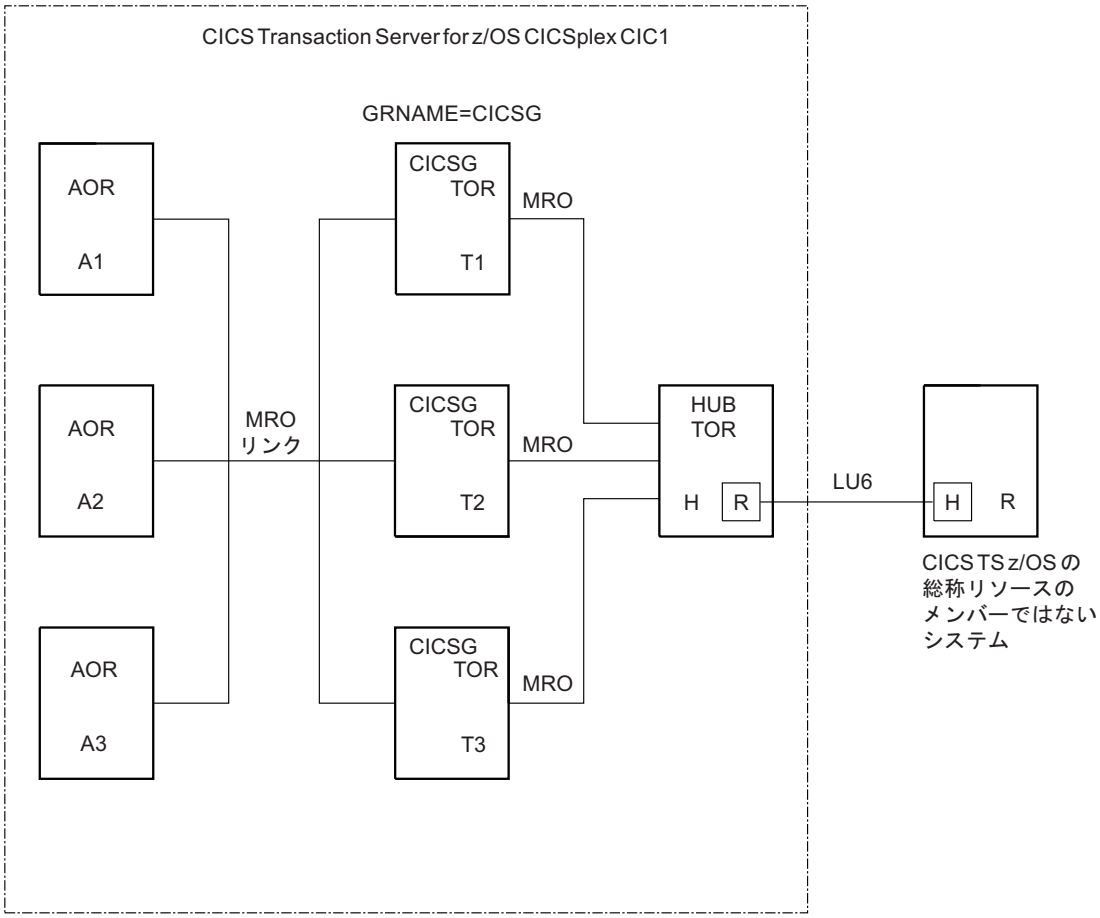


図 41. ネットワーク・ハブ: ハブは通常、総称リソース・グループのメンバーから、CICS Transaction Server for z/OS 総称リソースのメンバーではないシステムへのアウトバウンド LU6 要求に使用されます。

図 41 では、CICSplex CIC1 の領域同士が MRO リンクによって接続されています。端末専有領域 T1、T2、T3 は総称リソース・グループ CICSG のメンバーですが、ハブ TOR の H はそうではありません。H とリモート領域 R は、LU6.1 接続か APPC 接続です。TOR は、R への要求を H を通してデイジー・チェーンします。

第 3 部 相互通信リソースの定義

このパートでは、CICS 相互通信環境で必要なさまざまなリソースの定義方法について説明します。CICS リソースは、オンライン・リソース定義 (RDO) を使って定義します。

リソース定義について詳しくは、「*CICS Resource Definition Guide*」のリソース定義についてを参照してください。

165 ページの『第 13 章 リモート・システムへの接続の定義方法』では、リモート・システムへのリンクの定義方法について説明します。説明されているリンクは以下のようなものです。

- 他の CICS 領域への MRO リンク
- 外部 CICS インターフェースによって使用するための MRO リンク
- 分散プログラム・リンクと併用するための IP interconnectivity (IPIC) リンク
- 他の APPC システム (CICS または非 CICS) への複数セッション APPC リンク
- APPC 端末への単一セッション APPC リンク
- IMS システムへの LUTYPE6.1 リンク

213 ページの『第 15 章 APPC 接続の管理』では、マスター端末トランザクション (CEMT) を使って APPC リンクを管理する方法について説明します。

223 ページの『第 16 章 リモート・リソースの定義』では、リモート・リソースをローカル CICS システムに定義する方法について説明します。次のリソースが定義できます。

- リモート・ファイル
- リモート DL/I PSB
- リモート一時データ・キュー
- リモート一時記憶域キュー
- リモート端末
- リモート APPC 接続
- リモート・プログラム
- リモート・トランザクション

253 ページの『第 17 章 ローカル・リソースの定義』では、ISC および MRO のローカル・リソースの定義方法について説明します。一般に、これらのリソースは、ISC と MRO に必要なものであり、関連する機能グループを該当のテーブルに含めることによって得られます。ただし、提供された定義の一部を修正して、独自の通信プロファイルを用意することもできます。

© Copyright IBM Corp. 1977, 2009

第 13 章 リモート・システムへの接続の定義方法

CICS 領域間、または CICS 領域から非 CICS システムへのさまざまなタイプの接続を定義および管理することができます。

作成できる接続のタイプは次のとおりです。

- 複数領域操作 (MRO) 用の接続
- 外部 CICS インターフェース (EXCI) で使用するための接続
- リモート CICS TS for z/OS バージョン 3.2 またはそれ以降の領域への IPIC 接続
- 論理装置タイプ 6.2 (APPC) プロトコルを使用した、リモート・システムへの SNA を介した ISC 接続
- 論理装置タイプ 6.1 プロトコルを使用した、リモート IMS システムへの SNA を介した ISC 接続
- CICS トランザクション・ルーティングのための間接接続

ACF/VTAM アプリケーション間機能を使用する接続は、システム間接続とまったく同様に扱われるため、LUTYPE6.1 リンクまたは APPC リンクとして定義できます。

このセクションには、以下のトピックが含まれています。

- 『接続定義の紹介』
- 168 ページの『リモート・システムの識別』
- 181 ページの『複数領域操作のリンクの定義』
- 184 ページの『外部 CICS インターフェースが使用するリンクの定義』
- 169 ページの『IP 相互接続 (IPIC) の定義』
- 186 ページの『APPC 接続の定義』
- 195 ページの『論理装置タイプ 6.1 リンクの定義』
- 196 ページの『CICS-IMS 間 LUTYPE6.1 リンクの定義』
- 202 ページの『トランザクション・ルーティングのための間接リンクの定義』
- 208 ページの『XRF のための総称 applid と特定 applid』.

接続定義の紹介

CICS では、さまざまなタイプの接続を定義できます。MRO は SNA を介した ISC (APPC および LUTYPE 6.1) 接続、または TCP/IP 経由の IP 相互接続とともに使用できます。

MRO 接続と SNA を介した ISC 接続

リモート・システムへの MRO または SNA を介した ISC 接続の定義は、次の 2 つの部分で構成されています。

- リモート・システムそのものの定義

© Copyright IBM Corp. 1977, 2009

- リモート・システムとのセッションの定義

リモート・システムは、DEFINE CONNECTION コマンドによって定義されます。各セッション、または並列セッションのグループは、DEFINE SESSIONS コマンドによって定義されます。リモート・システムとセッションの定義は常に別々のものであり、インストールされるまで相互に関連付けられることはありません。

単一セッション APPC 端末では、DEFINE TERMINAL コマンドや DEFINE TYPETERM コマンドを使用する代替の定義方法を使用できます。

リモート・システムが CICS 領域であるか、リソース定義を使用してシステム間セッションを定義する他のシステム (IMS など) である場合、その接続定義は、リモート・システムの互換性のある定義と一致している必要があります。セッション特性に柔軟性がまったくない、またはほとんどないリモート・システム (APPC 端末など) の場合、その接続定義は、関連するリモート・システムの固定属性と一致している必要があります。

IPIC接続

2 つの CICS 領域間の IPIC 接続を定義する際には、次の 2 つを定義します。

- 接続のアウトバウンド属性の定義 (ターゲット CICS 領域を含む)
- 接続のインバウンド属性の定義 (CICS が要求を listen するポート番号を含む)

ローカル CICS 領域名

CICS Transaction Server for z/OS 領域は、複数の名前で認識されます。

- アプリケーション ID (APPLID)
- システム ID (SYSID)
- VTAM 総称リソース名

すべての CICS 領域には、APPLID と SYSID があります。VTAM 総称リソース・グループのメンバーである端末専有領域には、VTAM 総称リソース名もあります。VTAM 総称リソース名については、137 ページの『第 12 章 VTAM 総称リソースの構成』で説明しています。

CICS 領域の APPLID

CICS システムの APPLID は、それが相互通信ネットワークで認識されている名前、つまりそのネット名です。

- MRO の場合、CICS は、始動時か SET IRC OPEN コマンドへの応答で、CICS 領域間 SVC にサインオンするときに、この APPLID 名によって識別されます。
- SNA を介した ISC の場合、VTAM APPL ステートメントの APPLID によって、CICS が VTAM に識別されます。
- IPIC では、IPCONN 定義の APPLID オプションはリモート・システムの APPLID を識別します。

この CICS APPLID は、**APPLID** システム初期設定パラメーターに指定します。デフォルトは DBDCCICS です。この値は、CICS 始動時に指定変更できます。

z/OS シスプレックス内で、各 CICS 領域の APPLID は固有でなければなりません。CICS 領域がシスプレックスの一部でなく、ご使用のネットワークが複数のシスプレックスから成る場合、あるいは、CICS 領域がローカルのシスプレックス以外のシステムと通信する場合、可能であれば、ネットワーク全体で APPLID を固有に保つことをお勧めします。ネットワーク上に、同じ APPLID を持つシステムが存在する場合、IPIC による接続で NETWORKID オプションを指定することができます。この固有値を使用して、同じ APPLID を持つ 2 つ以上のリモート領域に接続することができます。

注: XRF を使用する CICS システムには APPLID が 2 つあり、それによって活動システムと代替システムが区別されます。この特別な場合については、208 ページの『XRF のための総称 applid と特定 applid』で説明します。

CICS 領域の SYSID

CICS 領域の SYSID は、その CICS 領域だけで認識される名前 (1 から 4 文字) です。

これは、次のものから獲得されます (優先順位の高いものから)。

1. 始動時指定変更
2. DFHSIT マクロの SYSIDNT オペランド
3. デフォルト **CICS**

注: マクロ・レベルのリソース定義を使用する場合には、DFHTCT TYPE=INITIAL マクロにローカル CICS 領域の SYSID を指定しなければならない場合があります。DFHTCT TYPE=INITIAL の SYSIDNT オペランドの唯一の目的は、端末管理テーブルにおけるローカル端末とリモート端末の定義の組み立てを制御することです。実行中の CICS 領域の SYSID は常に、システム初期設定パラメーターで指定されたものです。

ローカル CICS システムの applid

CICS システムの applid は、それが相互通信ネットワークで認識されている名前、つまりそのネット名です。

MRO の場合、CICS は、始動時か SET IRC OPEN 発信端末コマンドへの応答で、CICS 領域間 SVC にサインオンするときに、この applid 名によって識別されます。

SNA を介した ISC の場合、VTAM APPL ステートメントの applid によって、CICS が VTAM に識別されます。

IPICでは、IPCONN 定義の APPLID オプションはリモート・システムの applid を識別します。

この CICS applid は、APPLID システム初期設定パラメーターに指定します。デフォルトは DBDCCICS です。この値は、CICS 始動時に指定変更できます。

z/OS シスプレックス内で、各 CICS 領域の applid は固有でなければなりません。CICS 領域がシスプレックスの一部でなく、ご使用のネットワークが複数のシスプレックスから成る場合、あるいは、CICS 領域がローカルのシスプレックス以外のシス

テムと通信する場合、可能であれば、ネットワーク全体で applid を固有に保つことをお勧めします。ネットワーク上に同じ applid があるシステムが存在する場合、IPICによる接続で NETWORKID オプションを指定することができます。この固有値は、同じ applid を持つ 2 つ以上のリモート・システムに接続することを可能にします。

注: XRF を使用する CICS システムには applid が *2* つ あり、それによって、活動システムと代替システムが区別されます。この特別な場合については、208 ページの『XRF のための総称 applid と特定 applid』で説明します。

ローカル CICS システムの sysidnt

CICS システムの sysidnt は、その CICS システムだけで認識される名前 (1 から 4 文字) です。

これは、次のものから獲得されます (優先順位の高いものから)。

1. 始動時指定変更
2. DFHSIT マクロの SYSIDNT オペランド
3. デフォルト **CICS**

注: マクロ・レベルのリソース定義を使用する場合には、DFHTCT TYPE=INITIAL マクロにローカル CICS システムの sysidnt を指定しなければならない場合があります。DFHTCT TYPE=INITIAL の SYSIDNT オペランドの唯一の目的は、端末管理テーブルにおけるローカル端末とリモート端末の定義の組み立てを制御することです。(端末定義については、223 ページの『第 16 章 リモート・リソースの定義』で説明します。) 実行中の CICS システムの SYSIDNT は常に、システム初期設定パラメーターに指定されたものです。

リモート・システムの識別

CICS システムは、それ自身の sysidnt のほかに、通信可能な他のシステムが持つそれぞれの sysidnt を必要とします。sysidnt 名は、セッション定義とシステム定義を関連付け、ファイルなどのリモート・リソースがあるシステムを識別し、アプリケーション・プログラムで特定のシステムを参照するために使用されます。

sysidnt 名は、それが定義されている CICS システムだけのもので、他のシステムには知られていません。具体的にいえば、リモート CICS システム用に定義されている sysidnt は、リモート・システムがそれ自身を認識するための sysidnt とは別のものですので、これらを同じにする必要はありません。

リモート・システムに割り当てられているローカル (プライベート) sysidnt と、ネットワーク内でそのリモート・システムをグローバルに認識する際に使用する applid (そのネット名) の間のマッピングは、相互通信リンクを定義する際に行われます。例えば、MRO または SNA を介した ISC 接続の場合、CONNECTION 定義に以下のように指定できます。

```
DEFINE CONNECTION(sysidnt) リモート・システムのローカル名
       NETNAME(applid)     リモート・システムの applid
```

IPICによる接続の場合、IPCONN 定義に以下のように指定できます。

```
DEFINE IPCONN(sysidnt)  リモート・システムのローカル名
       APPLID(applid)  リモート・システムの applid
```

CICS システムに定義される sysidnt 名は固有でなければなりません。

IP 相互接続 (IPIC) の定義

TCP/IP 接続を定義することにより、IP 相互接続 (IPIC) を使用してリモート CICS 領域と通信することができます。

始める前に

制約事項: IPIC は特定の相互通信機能とリリースをサポートしています。詳しくは、このトピックに関連するリンクを参照してください。

TCP/IP サービスは、CICS 領域でアクティブにする必要があります。TCP/IP サービスをアクティブにするには、**TCPIP** システム初期設定パラメーターを YES に設定します。

このタスクについて

IPIC 接続を定義するには、接続する各 CICS 領域で、IPCONN と TCPIPSERVICE の 2 つのリソースを定義します。CEDA または CICSPlex SM を使用して新しい IPIC 接続を定義することも、既存の APPC 接続をマイグレーションすることもできます。

1. ローカル CICS 領域で IPCONN リソース定義を定義します。
   a. IPCONN 名を指定します。CICS-CICS 間通信用に、4 文字と 4 つの末尾スペースのある IPCONN 名を指定してください。
   b. リモート CICS 領域の TCPIPSERVICE 定義に指定されている値を使用して、HOST 属性でホスト名を指定します。 例えば、「hostb.example.com」と指定します。 ホスト名の長さは最大で 116 文字です。ホスト名には IPv4 アドレスまたは IPv6 アドレスを指定することもできます。IPv6 アドレス (または IPv6 アドレスに解決されるホスト名) を指定した場合、デュアル・モード (IPv4 および IPv6) 環境で実行していることと、通信先のクライアントまたはサーバーもデュアル・モード (IPv4 および IPv6) 環境で実行されていることを確認してください。
   c. PORT 属性で、リモート CICS 領域が listen するポート番号を指定します。この IPCONN リソースがアウトバウンド要求に使用されず、ユーザーが CICS Transaction Gateway を使用している場合には、NO を指定します。
   d. ローカルの CICS 領域の TCPIPSERVICE リソースの名前を指定します。このリソースは、IPIC 接続のインバウンド属性を TCPIPSERVICE 属性の値として定義します。
   e. オプション: 異なるネットワークにあるリモート・システムに接続する場合は、APPLID 属性および NETWORKID 属性の値を指定します。 APPLID 属性と NETWORKID 属性を組み合わせることにより、リモート CICS 領域が固有の名前で参照されます。
   f. オプション: INSERVICE 属性に YES または NO を指定して、リソースの定義時に接続できるようにするかどうかを設定します。

g. RECEIVECOUNT 属性および SENDCOUNT 属性の値を指定して、IPIC 接続に許可する受信セッションと送信セッションの数を設定します。

```
DEFINE
  IPCONN(CICB)                  接続のローカル名
  GROUP(groupname)              関連定義のグループ名
  HOST(hostb.example.com)       リモート・システムのホスト名
  PORT(32022)                   リモート領域が listen するポート番号
TCPIPSERVICE(TSA)             インバウンド要求の属性を定義する
                                 PROTOCOL(IPIC) TCPIPSERVICE の名前
APPLID(AOR001)                リモート・システムの SIT APPLID
  INSERVICE(YES)
  RECEIVECOUNT(20)              受信セッションの数
  SENDCOUNT(20)                 送信セッションの数
```

図 42. IPIC 接続の IPCONN 定義の例

このリソースで使用可能なすべての属性の詳細については、「*CICS Resource Definition Guide*」を参照してください。

2. ローカル CICS 領域でインバウンド要求を受け取るには、TCPIPSERVICE リソース定義を定義します。 TCPIPSERVICE リソースの名前は、IPCONN リソースの TCPIPSERVICE 属性の値と一致させる必要があります。
   a. HOST 属性で、ローカル CICS 領域の IP アドレスを指定します。 ホスト名の長さは最大で 116 文字です。ホスト名には IPv4 アドレスまたは IPv6 アドレスを指定することもできます。IPv6 アドレスを使用する場合、デュアル・モード環境で実行していることと、通信先のクライアントまたはサーバーもデュアル・モード環境で実行されていることを確認してください。
   b. PORT 属性で、ローカル CICS 領域が着信クライアント要求を listen するポート番号を指定します。
   c. PROTOCOL 属性に IPIC を指定します。
   d. SOCKETCLOSE 属性に NO を指定します。
   e. DFHISCOP プログラムを実行する CICS トランザクションの ID (4 文字) を TRANSACTION 属性の値として指定します。 IPIC のデフォルト・トランザクションは CISS です。
   f. オプション: IPCONN 自動インストール・ユーザー・プログラムの名前を URM 属性の値として指定します。 この属性を指定しない場合、CICS は CICS 提供のデフォルトの IPCONN 自動インストール・ユーザー・プログラムである DFHISAIP を使用します。自動インストールを無効にするため NO を指定します。

```
DEFINE
  TCPIPSERVICE(TSB)             インバウンド要求を受信する
                                TCPIPSERVICE の名前
                                (IPCONN 定義で指定されたものと同じ)
GROUP(groupname)              関連定義のグループ名
  HOST(hostb.example.com)       listen する場所の IP アドレス
  PORTNUMBER(32022)             listen するポート番号
  PROTOCOL(IPIC)                IP 相互接続リンクの IPIC でなければなりません
  SOCKETCLOSE(NO)               IP 相互接続リンクの NO でなければなりません
  TRANSACTION(CISS)             新しい要求を処理するトランザクション
  URM(NO)                       IPCONN 自動インストール・ユーザー・プログラム
```

図 43. IPIC 接続の TCPIPSERVICE 定義の例

TCPIPSERVICE 定義のすべての属性の詳細については、「*CICS Resource Definition Guide*」を参照してください。

3. DPL 呼び出しを使用する場合、IPIC 接続を使用するように DPL プログラムの PROGRAM リソース定義を定義または変更します。 IPCONN リソース定義の名前を REMOTESYSTEM 属性の値として指定します。
4. リモート CICS 領域で TCPIPSERVICE リソース定義を定義します。
5. リモート CICS 領域で IPCONN リソース定義を定義します。 AUTOCONNECT(YES) を指定して、2 つの CICS 領域間の接続を確立します。

タスクの結果

ローカル CICS 領域およびリモート CICS 領域でリソースを定義して有効化すると、CICS 領域間で接続が確立されます。

次のタスク

IBM CICS Explorer または Web User Interface を使用して、IPIC 接続を表示および更新することができます。いずれかの IPCONN リソースに AUTOCONNECT (YES) を指定しない場合は、リソースの状況を更新して、接続を獲得する必要があります。

APPC 接続と MRO 接続を IPIC にマイグレーションする

既存の MRO 接続、APPC 接続、および LUTYPE6.1 接続を IPIC 接続にマイグレーションすることができます。既存の接続は、マイグレーション前と同様に動作します。

始める前に

APPC 接続と MRO 接続を IPIC にマイグレーションする前に、IPIC のサポートをインストールする必要があります。その方法については、「*CICS Transaction Server for z/OS インストール・ガイド*」で説明しています。

このタスクについて

既存の接続を IPIC にマイグレーションする方法については、175 ページの『DFH0IPCC マイグレーション・ユーティリティー』を参照してください。

1. 相互接続された各領域で、TCPIPSERVICE リソース定義を作成します。
   a. PROTOCOL(IPIC) を指定します。
   b. TCPIPSERVICE(DFHIPIC) または TCPIPSERVICE(*servicename*) を指定します。 ユーザー定義名を指定する場合は、作成するすべての TCPIPSERVICE 定義でこの名前を指定します。
   c. PORTNUMBER などの他のオプションを、TCPIPSERVICE 定義をインストールする領域の要件に従って指定します。

   必要な定義の数は、指定する固有ポート番号の数などにより異なります。
2. 各 TCPIPSERVICE 定義を自身のリソース定義グループに入力します。

3. 相互接続された領域によって使用される 各 CICS システム定義ファイル (CSD) に 1 つ以上のリソース・グループを追加し、CSD が提供する CICS 領域の数に応じた数、および必要な固有 TCPIPSERVICE 定義の数を入力します。
4. 相互接続されている各領域で、1 つの TCPIPSERVICE を DFHIPIC という名前かユーザー定義のサービス名でインストールします。
5. 以下の例 1 に示すように、相互接続された CICS 領域についてアプリケーション ID テーブルを作成します。
   a. テーブルは、固定長ブロック (80 バイトのレコード・フォーマット) で作成します。
   b. 任意の方法 (手動、スプレッドシートやスクリプトなどのユーティリティーを使用するなど) でテーブルに入力します。このとき、固定長形式は保持する必要があります。提供されているコメントやヘッダー行は、削除したり省略したりすることができます。
   c. テーブルには、アプリケーション ID (APPLID) とネットワーク ID を含める必要があります。また、必要に応じて、TCP/IP ポート番号、相互接続される全 CICS 領域のホスト名も含めます。
   d. 以前に定義された TCPIPSERVICE 定義に DFHIPIC 以外の名前が付けられている場合は、テーブルの HOST 列に `TCPIPSERVICE=servicename` を含む .DEFAULT レコードを入力する必要があります。
6. アプリケーション ID テーブルを、相互接続される領域で使用される CSD を格納するすべてのシステムにコピーします。
7. 以下の 例 2 に示すように、DFHCSDUP を介して DFH0IPCC を起動するための JCL (ジョブ制御言語) を作成します。 DFH0IPCC が CONNECTION 定義および SESSIONS 定義に関する情報を検索する際に対象となるリストとリソース・グループを指定します。 JCL (ジョブ制御言語) は DFHCSDUP EXTRACT コマンドを発行します。これは、ユーティリティー・プログラムを *USERPROGRAM* として渡します。
8. いずれかの CSD 所有システムで、カスタマイズした JCL (ジョブ制御言語) ファイルを使用して DFH0IPCC ユーティリティー・プログラムを起動します。ユーティリティー・プログラムによって、CONNECTION 定義および SESSIONS 定義に関する情報が収集され、IPCONN 定義が作成されて、一連の DEFINE ステートメントが記述されます。これらによって、DFHCSDUP を起動する JCL (ジョブ制御言語) の SYSIN が形成されます。
9. ユーティリティー・プログラムによって生成された出力を確認します。
   a. IPCONN 定義がインストールに適正であることを確認します。 デフォルトの SSL 設定を変更して、特定の接続についてのセキュリティー管理を高めることができます。
   b. 生成された JCL (ジョブ制御言語) で、USER、PASSWORD、およびライブラリー名をローカルで使用されているものと一致するように変更します。
10. 生成された JCL (ジョブ制御言語) を実行し、新しい IPCONN リソースを CSD ファイルに追加します。
11. 相互接続される CICS 領域で使用される各 CSD ファイルで、手順 8、9、および 10 を繰り返します。

例

この アプリケーション ID テーブルの例には、使用する必要がある形式が示されています。例に続くテーブルには、テーブル形式の参照情報が示されています。

```
*********************************************************************************
*                                                                               *
* Description:                                                                  *
*     This Applid Table is for DFH0IPCC. This table must contain the            *
*     APPLIDs, NETWORKIDs (where applicable for foreign network connectivity),  *
*     PORT numbers, and TCP/IP HOST names for all CICS regions in the systems   *
*     for which IPCONN definitions are to be created.                           *
*                                                                               *
* File Format:                                                                  *
*     This file must be in FB80 format, and relies on a tabular layout shown    *
*     below. Any characters can be used as separators. Add comments using an    *
*     asterisk in the first column of the line. A HOST name that is too long    *
*     to fit into the table can be continued by placing an asterisk in column   *
*     80, and continuing on column 25 of the next row (the first column of the  *
*     space for HOST). The APPLID field of any continuation record(s) must be   *
*     left blank.                                                               *
*                                                                               *
* Notes:                                                                        *
*     The optional .DEFAULT record (shown below) can be used to provide either  *
*     one or both of the following parameters:                                  *
*     > A TCPIPSERVICE name, which must be provided immediately after           *
*       'TCPIPSERVICE=' in the HOST column. If a name is not provided, it       *
*       defaults to 'DFHIPIC'. In either case, this value is the name that must *
*       be used when defining the TCPIPSERVICEs for the CICS systems referred   *
*       to in this table.                                                       *
*     > A default NETWORKID, which must be provided in the NET-ID column.       *
*       Its omission results in the omission of the NETWORKID parameter in      *
*       the generated IPCONN definition statements for those APPLIDs that had   *
*       a blank NET-ID column.                                                  *
*                                                                               *
*     Examples of various valid table entries are shown following the .DEFAULT  *
*     record. These are examples only. Ensure that all rows adhere to your      *
*     site's standards and conventions.                                         *
*                                                                               *
*     Important! When editing this file, ensure that the CAPS setting is OFF.   *
*     Otherwise, the case-sensitive HOST names might be destroyed.              *
*                                                                               *
*********************************************************************************
*
*********************************************************************************
APPLID. |NET-ID. |PORT.|HOST.
*********************************************************************************
.DEFAULT|LOCALNET|     |TCPIPSERVICE=TCPSERV1
APPL1A  |         |9876 |my.local.hostname
OTHERCIC|OTHERNET|12345|this.host.has.a.very.long.name.which.is.going.to.requir*
        |         |     |e.a.continuation.record
* Comments such as this are entirely free-form other than the * in column 1
CICSXYZ |        |9875 |10.2.156.221
```

図 44. 例 1: アプリケーション ID テーブル

表 6. アプリケーション ID テーブルの形式

テーブル列	長さ	説明
APPLID	8 文字	固有 ID または .DEFAULT。.DEFAULT を使用して、NETID または TCPIPSERVICE のデフォルト値を指定します。最初のドットは、「DEFAULT」という単語が有効なアプリケーション ID として使用されるのを避けるために必要です。テーブルには、.DEFAULT を 1 行のみ指定することができます。
セパレーター	1 文字	任意の英数字。
NETID	8 文字	ネットワーク ID。空のままにすると、.DEFAULT 行で指定されたデフォルトの NETID が使用されます。
セパレーター	1 文字	任意の英数字。
PORT	5 文字	listen するポート番号。
セパレーター	1 文字	任意の英数字。
HOST	55 文字	TCP/IP ホスト名。
継続列	1 文字	通常は空です。このフィールドに非ブランク文字が入力されている場合、ホスト名が 55 文字よりも長いことを示しています。つまり、次の行の HOST 列に値の続きが入力されます。

次の JCL (ジョブ制御言語) の例を使用することにより、DFHCSDUP を介して DFH0IPCC を呼び出すことができます。

```
//IPCJOB   JOB user,CLASS=A,USER=user,PASSWORD=pass
/*ROUTE PRINT user
//CSDUPJOB EXEC PGM=DFHCSDUP,REGION=0M
//STEPLIB  DD DSN=loadlibrary,DISP=SHR
//         DD DSN=loadlibrary,DISP=SHR
//DFHCSD   DD DSN=csdfilename,DISP=SHR
//SYSPRINT DD SYSOUT=A
//CSDCOPY  DD UNIT=VIO
//APPLTABL DD DSN=applidtablename,
//            DISP=SHR,UNIT=SYSDA,SPACE=(CYL,(2,1)),
//            DCB=(RECFM=FB,BLKSIZE=15360,LRECL=80)
//LOGFILE     DD DSN=logfilename,
//            DISP=(MOD,CATLG,CATLG),UNIT=SYSDA,SPACE=(CYL,(2,1)),
//            DCB=(RECFM=FB,BLKSIZE=15360,LRECL=80)
//OUTFILE  DD DSN=outputfilename,
//            DISP=(MOD,CATLG,DELETE),UNIT=SYSDA,SPACE=(CYL,(2,1)),
//            DCB=(RECFM=FB,BLKSIZE=15360,LRECL=80)
//SYSUDUMP DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSIN    DD *
EXTRACT GR(group1) USERPROGRAM(DFH0IPCC) OBJECTS
EXTRACT GR(group2) USERPROGRAM(DFH0IPCC) OBJECTS
EXTRACT GR(list1) USERPROGRAM(DFH0IPCC) OBJECTS
EXTRACT GR(list2) USERPROGRAM(DFH0IPCC) OBJECTS
/*
//
```

図 45. 例 2: DFHCSDUP を介して DFH0IPCC を起動する JCL (ジョブ制御言語)

DFH0IPCC マイグレーション・ユーティリティー

CICS とともに提供される DFH0IPCC ユーティリティー・プログラムによって、既存の APPC 接続と MRO 接続が IPIC 接続 (IPCONN) に変換されます。DFH0IPCC は、DFHCSDUP システム定義ユーティリティー・プログラムとともに使用することを目的としたサンプル・プログラムです。このユーティリティーによって、DFHCSDUP への入力を形成する一連のステートメントが生成されます。

DFH0IPCC プログラムは、テーブルで提供される入力データを取得します。このテーブルは「アプリケーション *ID* テーブル」と呼ばれ、編集可能です。このテーブルは、関連セットアップの全領域のアプリケーション ID、領域の対応ホスト名、およびインバウンド TCP/IP 接続を処理する TCPIPSERVICE 定義が listen するポートを格納するために使用されます。

DFH0IPCC プログラムは、CICS 領域の CSD にあるリストとリソース・グループを検査し、見つかった CONNECTION 定義と SESSIONS 定義に関する情報を収集します。このプログラムは、CONNECTION 定義と SESSIONS 定義の各 APPC ペアまたは MRO ペアについて、IPCONN 定義を作成します。必要な場合は、アプリケーション ID テーブルから取得された残りの属性の値とともに、IPCONN 定義の属性が CONNECTION 定義と SESSIONS 定義から取得されます。また、IPCONN 定義の属性に、そのデフォルト値が使用されることもあります。IPCONN 定義の作成後、ユーティリティー・プログラムは一連の DEFINE ステートメントを記述します。このステートメントは、DFHCSDUP を起動する JCL (ジョブ制御言語) の SYSIN を形成します。

IPCONN 属性マッピング

次のテーブルには、DFH0IPCC ユーティリティー・プログラムが CONNECTION の属性を IPCONN 定義にマップする様子が示されています。

表 7. IPCONN 属性マッピング

IPCONN 定義の属性	マイグレーション元または作成元	コメント
APPLID	CONNECTION (NETNAME)	ダイレクト・マイグレーション
AUTOCONNECT	CONNECTION (AUTOCONNECT)	ダイレクト・マイグレーション。ただし、ALL の場合は、新しい値を Yes に設定します。
CERTIFICATE	N/A	ブランク
CIPHERS	N/A	ブランク
DESCRIPTION	N/A	ブランク。マイグレーションされません。これを DFH0IPCC 出力に追加できます。
GROUP	CONNECTION (GROUP) SESSIONS (GROUP)	変更されません
HOST	アプリケーション ID テーブル	アプリケーション ID テーブルで指定する必要があります。
INSERVICE	CONNECTION (INSERVICE)	ダイレクト・マイグレーション
IPCONN	CONNECTION (CONNECTION)	ダイレクト・マイグレーション。177 ページの『IPCONN 名』を参照してください。
MAXQTIME	CONNECTION (MAXQTIME)	ダイレクト・マイグレーション
NETWORKID	アプリケーション ID テーブル	同等の属性はありません。アプリケーション ID テーブルで指定されていない場合、またはデフォルトを使用する場合は空のままにしておきます。
PORT	アプリケーション ID テーブル	アプリケーション ID テーブルで指定する必要があります。
QUEUELIMIT	CONNECTION (QUEUELIMIT)	ダイレクト・マイグレーション
RECEIVECOUNT	SESSIONS (MAXIMUM) の合計	MRO SESSIONS の同等の設定からダイレクト・マイグレーションされるか、APPC SESSIONS MAXIMUM 設定から派生されます。
SENDCOUNT	SESSIONS (MAXIMUM) の合計	MRO SESSIONS の同等の設定からダイレクト・マイグレーションされるか、APPC SESSIONS MAXIMUM 設定から派生されます。
SSL	N/A	空のままにしておきます。これは、DFH0IPCC 出力で変更できます。

表 7. IPCONN 属性マッピング (続き)

IPCONN 定義の属性	マイグレーション元または作成元	コメント
TCPIPSERVICE	アプリケーション ID テーブル	常に「DFHIPIC」であるかアプリケーション ID テーブルと同じです。『TCPIPSERVICE 名』を参照してください。
XLNACTION	CONNECTION (XLNACTION)	ダイレクト・マイグレーション

IPCONN 名

IPCONN 名は重複を避けるために生成されます。DFH0IPCC ユーティリティー・プログラムでは CONNECTION 定義の名前が使用されます。これは、CONNECTION 定義と、CONNECTION 定義から作成される IPCONN 定義との間に 1 対 1 の関係があるためです。CONNECTION NETNAME と IPCONN APPLID が同じ場合、CICS では同じ名前の CONNECTION 定義と IPCONN 定義を共存させることができます。この場合、CICS では、サポートされている機能のルーティングに、CONNECTION 定義ではなく IPCONN 定義が選択されます。

TCPIPSERVICE 名

IPCONN 定義ではパートナー領域の TCPIPSERVICE 名を確認できないため、ユーティリティーを使用して TCPIPSERVICE 定義を生成することはできません。これは手動で定義する必要があります。このユーティリティーを使用するには、ユーティリティーが IPCONN 定義を生成する領域のすべての TCPIPSERVICE 名を同じにする必要があります。

APPLID ファイルの .DEFAULT 行を使用して異なる名前を指定していない限り、DFH0IPCC ユーティリティー・プログラムによって作成される IPCONN 定義には、すべてデフォルトの属性 (TCPIPSERVICE (DFHIPIC)) があります。他の名前を指定している場合は、作成するすべての TCPIPSERVICE 定義でその名前を使用します。

IPCONN 定義における同等の属性

DFH0IPCC マイグレーション・ユーティリティーを実行する代わりに、APPC 接続および MRO 接続を手動でマイグレーションする場合は、次の表を参照してください。この表には、MRO 接続と APPC 接続のための CONNECTION リソース定義と SESSION リソース定義の属性、および IPCONN 定義における同等の属性が示されています。

APPC 接続

表 8. APPC 接続を IPICにマイグレーションする CONNECTION のオプションと IPCONN における同等のオプション

CONNECTION のオプション	APPC の使用可能な値	IPCONN における同等の値
ACCESSMETHOD	VTAM	適用外
ATTACHSEC	LOCAL | IDENTIFY | VERIFY | PERSISTENT | MIXIDPE	USERAUTH LOCAL | IDENTIFY | VERIFY | NO | CERTIFICATE
AUTOCONNECT	NO | YES | ALL	NO | YES

表 8. APPC 接続を IPICにマイグレーションする CONNECTION のオプションと IPCONN における同等のオプション (続き)

CONNECTION のオプション	APPC の使用可能な値	IPCONN における同等の値
BINDSECURITY	NO | YES	SSL NO | YES
DATASTREAM	USER	適用外
INDSYS	適用外 (間接接続のみ)	適用外 (間接接続のみ)
INSERVICE	YES | NO	同じ
MAXQTIME	NO | 0 - 9999	同じ
NETNAME	リモート領域の VTAM APPLID(XRF の場合は総称 applid。VTAM 総称リソースへの接続の場合は、applid か総称リソース名)	APPLID と NETWORKID の組み合わせ
PROTOCOL	APPC	適用外
PSRECOVERY	SYSDEFAULT | NONE	適用外
QUEUELIMIT	NO | 0 - 9999	同じ
RECORDFORMAT	U	適用外
REMOTENAME	リモート・システムが自身を識別するための名前 (システム ID)	適用外
REMOTESYSNET	リモート・リソースを所有するリモート・システムのアプリケーション ID (リモート・システムへのリンクが間接の場合)	適用外
REMOTESYSTEM	リモート・システムの名前 (システム ID) またはパスの次のシステムのシステム ID (リモート・システムへのリンクが間接の場合)。	適用外
SECURITYNAME	リモート・システムの RACF ID	同じ
SINGLESESS	NO | YES	適用外
USEDFLTUSER	NO | YES	適用外
XLNACTION	KEEP | FORCE	同じ

表 9. APPC 接続を IPICにマイグレーションする SESSIONS のオプションと IPCONN における同等のオプション

SESSIONS のオプション	APPC の使用可能な値	IPCONN における同等の値
AUTOCONNECT	NO | YES | ALL	適用外
BUILDCHAIN	YES	適用外
CONNECTION	この SESSION 定義が適用される CONNECTION の名前	適用外
DISCREQ	適用外	適用外
IOAREALEN	適用外	適用外
MAXIMUM	1 - 999, 0 - 999	SENDCOUNT & RECEIVECOUNT
MODENAME	VTAM LOGMODE の名前	適用外
NEPCLASS	ノード・エラー・プログラムのトランザクション・クラス	適用外
NETNAMEQ	適用外	適用外

表 9. APPC 接続を IPICにマイグレーションする SESSIONS のオプションと IPCONN における同等のオプション (続き)

SESSIONS のオプション	APPC の使用可能な値	IPCONN における同等の値
PROTOCOL	APPC	適用外
RECEIVECOUNT	適用外	MAXIMUM から派生
RECEIVEPFX	適用外	適用外
RECEIVESIZE	受け取る RU サイズ: 1 - 30720	適用外
RECOVOPTION	SYSDEFAULT | CLEARCONV | RELEASESESS | UNCONDREL | NONE	適用外
RELREQ	NO | YES	適用外
SENDCOUNT	適用外	MAXIMUM から派生
SENDPFX	適用外	適用外
SENDSIZE	送信する RU サイズ: 1 - 30720	適用外
SESSNAME	適用外	適用外
SESSPRIORITY	0 - 255	適用外
USERAREALEN	TCTTE ユーザー域の長さ: 0 - 255	適用外
USERID	サインオンの ID	適用外

MRO 接続

MRO 接続は、同じシスプレックス内における領域間の CICS-CICS 間接続です。このタイプの接続では、MRO のほうが IPIC よりも便利な場合があります。これは、IPIC がサブセットをサポートしている一方で、MRO はすべての基本 CICS 相互通信機能をサポートしているためです。

表 10. MRO 接続を IPICにマイグレーションする CONNECTION のオプションと IPCONN における同等のオプション

CONNECTION のオプション	MRO の使用可能な値	IPCONN における同等の値
ACCESSMETHOD	IRC | XM	適用外
ATTACHSEC	LOCAL | IDENTIFY	USERAUTH LOCAL | IDENTIFY | VERIFY | NO | CERTIFICATE
AUTOCONNECT	適用外	NO | YES
BINDSECURITY	適用外	SSL NO | YES
DATASTREAM	USER	適用外
INDSYS	適用外 (間接接続のみ)	適用外 (間接接続のみ)
INSERVICE	YES | NO	同じ
MAXQTIME	NO | 0 - 9999	同じ
NETNAME	リモート領域の SIT で指定された APPLID	`host.domain.country[:port]`
PROTOCOL	ブランク	適用外
PSRECOVERY	適用外	適用外
QUEUELIMIT	NO | 0 - 9999	同じ
RECORDFORMAT	U	適用外

表 10. MRO 接続を IPICにマイグレーションする CONNECTION のオプションと IPCONN における同等のオプション (続き)

CONNECTION のオプション	MRO の使用可能な値	IPCONN における同等の値
REMOTENAME	適用外	適用外
REMOTESYSNET	適用外	適用外
REMOTESYSTEM	適用外	適用外
SECURITYNAME	適用外	同じ
SINGLESESS	適用外	適用外
USEDFLTUSER	NO | YES	適用外
XLNACTION	KEEP | FORCE	同じ

表 11. MRO 接続を IPICにマイグレーションする SESSIONS のオプションと IPCONN における同等のオプション

SESSIONS のオプション	MRO の使用可能な値	IPCONN における同等の値
AUTOCONNECT	適用外	適用外
BUILDCHAIN	適用外	適用外
CONNECTION	この SESSION 定義が適用される CONNECTION の名前	適用外
DISCREQ	適用外	適用外
IOAREALEN	デフォルトの TIOA サイズ: 0 - 32767、0 - 32767	適用外
MAXIMUM	適用外	適用外
MODENAME	適用外	適用外
NEPCLASS	ノード・エラー・プログラムのトランザクション・クラス	適用外
NETNAMEQ	適用外	適用外
PROTOCOL	LU61	適用外
RECEIVECOUNT	受信セッションの数: 1 - 999	同じ
RECEIVEPFX	Termid 接頭部	適用外
RECEIVESIZE	適用外	適用外
RECOVOPTION	適用外	適用外
RELREQ	適用外	適用外
SENDCOUNT	送信セッションの数: 1 - 999	同じ
SENDPFX	Termid 接頭部	適用外
SENDSIZE	適用外	適用外
SESSNAME	適用外	適用外
SESSPRIORITY	0 - 255	適用外
USERAREALEN	TCTTE ユーザー域の長さ: 0 - 255	適用外
USERID	サインインする ID	適用外

複数領域操作のリンクの定義

このセクションでは、ローカル CICS システムと、同じオペレーティング・システム内にある他の CICS 領域との間の領域間通信接続の定義方法について説明します。

注: 外部 CICS インターフェース (EXCI) では、特殊形式の MRO リンクが使用されます (184 ページの『外部 CICS インターフェースが使用するリンクの定義』を参照)。この項では、CICS システム間の MRO リンクを説明します。ただし、この説明のほとんどは、184 ページの『外部 CICS インターフェースが使用するリンクの定義』で特に述べる点以外、EXCI リンクにも適用されます。

ローカル CICS システムの観点からいえば、このリンクの各セッションは、SEND セッションか RECEIVE セッションと見なすことができます。SEND セッションは、ローカル・システムからリモート・システムへの最初の要求と、この要求に関連するそれ以降のすべてのデータ・フローを送信するために使用されます。同様に、RECEIVE セッションは、リモート・システムからの最初の要求を受信するために使用されます。

MRO リンクの定義

接続と、それに関連する**セッション**のグループは、別々に定義する必要があります。この 2 つの定義は、CICS システム定義ファイル (CSD) の個別の「オブジェクト」であり、グループがインストールされるまで、相互に対応付けられることはありません。

このタスクについて

MRO リンクでは次の規則が適用されます。

- CONNECTION と SESSIONS は、同じ GROUP 内になければならない。
- SESSIONS には PROTOCOL(LU61) を指定する必要がありますが、CONNECTION の PROTOCOL オプションはブランクでなければなりません。
- SESSIONS の CONNECTION オプションは、CONNECTION に指定する sysidnt と一致しなければなりません。
- MRO CONNECTION には 1 つの SESSIONS 定義しか対応付けられません。
- 任意の 2 つの CICS 領域の間には 1 つの MRO リンクしか定義できません。つまり、それぞれの DEFINE CONNECTION には固有のネット名を指定する必要があります。

この章で既に述べたように、**sysidnt** は、リンクを定義している CICS システムのローカル名になります。ネット名は、リモート・システムが領域間 SVC にログオンするときに使用する名前 (つまり、その applid) でなければなりません。ネット名を指定しない場合は、sysidnt がこれらの要件を満たしていなければなりません。

注: 明確かつ簡潔にするために、関係のないオプションや本質的でないオプションは、182 ページの図 46 およびこの章のすべての定義例から除いています。また、CEDA DEFINE パネルのレイアウトを模写して示すこともしていません。すべての RDO オプションについての詳細は、「*CICS Resource Definition Guide*」を参照してください。

```
:NONE.DEFINE
  CONNECTION(sysidnt)
  GROUP(groupname)
  NETNAME(name)
  ACCESSMETHOD(IRC|XM)
  QUEUELIMIT(NO|0-9999)
  MAXQTIME(NO|0-9999)
  INSERVICE(YES)
  ATTACHSEC(LOCAL|IDENTIFY)
  USEDFLTUSER(NO|YES)
DEFINE
  SESSIONS(csdname)
  GROUP(groupname)
  CONNECTION(sysidnt)
  PROTOCOL(LU61)
  RECEIVEPFX(prefix1)
  RECEIVECOUNT(number1)
  SENDPFX(prefix2)
  SENDCOUNT(number2)
  SESSPRIORITY(number)
  IOAREALEN(value)
```

図 46. MRO リンクの定義

CONNECTION 定義の QUEUELIMIT オプションには、リモート・システムに対する空きセッションを待つキューに入れることのできる要求の最大数を指定します。MAXQTIME オプションには、キューがいっぱいになってから、リモート・システムが応答しないためにそれを除去するまでの時間の最大を指定します。これについては、さらに 307 ページの『第 24 章 システム間のセッション・キューの管理』で説明します。

ATTACHSEC と USEDFLTUSER のセキュリティー・オプションについては、「*CICS RACF Security Guide*」の リンク定義におけるユーザー・セキュリティーの指定 (Specifying user security in link definitions)を参照してください。

SESSIONS 定義には、必要な SEND セッションと RECEIVE セッションの数 (少なくとも、それぞれに 1) を指定する必要があります。最初の要求を RECEIVE セッションで送信することはできません。必要な SEND セッションと RECEIVE セッションの数を決めるときには、このことに注意してください。

セッションに名前を付けるための接頭部が指定できます。接頭部とは、セッション ID (TRMIDNT) を生成するために使用される 1 文字または 2 文字のストリングです。接頭部を指定しないと、デフォルト「>」(SEND) と「<」(RECEIVE) が使用されます。接頭部には、次の理由により、できるだけデフォルトを使ってください。

- CICS によって生成されるセッション名が必ず固有になる。接頭部を指定することによって、既にある接続名や端末名と競合するようなことがあってはなりません。
- 独自の 2 文字の接頭部を使用すると、接続ごとに定義できるセッション数は最大 99 です。独自の 1 文字の接頭部を使用すると、最大で 999 に増えます (デフォルトの接頭部の場合と同じ) が、固有のセッション名を保証するのはさらに難しくなります。

CICS が MRO セッションの名前をどのように生成するかについては、「*CICS Resource Definition Guide*」のSESSIONS 定義の属性を参照してください。

MRO のアクセス方式の選択

ACCESSMETHOD(XM) を指定することにより、MRO リンクにおける MVS の仮想記憶間サービスを選択することができます。仮想記憶間サービスを使用するには、リンクのもう一方の端も仮想記憶間サービスを指定していなければなりません。

領域間通信に CICS タイプ 3 SVC を選択するには、ACCESSMETHOD(IRC) を使用してください。

MVS 仮想記憶間サービスを使用すると、領域間でメッセージを伝送するための命令数が減少します。さらに、MVS 共通サービス域に必要な仮想記憶域が少なくてすみます。しかし、セキュリティーの点から見ると、仮想記憶間サービスは他の方法よりも劣っている場合があります (「*CICS RACF Security Guide*」の MRO のアクセス方式を選択する際のセキュリティーへの影響 (Security implications of choice of MRO access method)を参照)。

さらに、仮想記憶間サービスでは、CICS アドレス・スペースがスワップ不能でなければなりません。アドレス・スペースのスワップの対象となるような、アクティビティーが少ないシステムの場合には、スワップ不能なアドレス・スペースによって大きな実記憶域が必要になるよりは、CICS 領域間 SVC のパスを長くする方がよいでしょう。

注: システム間複数領域操作 (XCF/MRO) を使用した場合、CICS は、XM か IRC が指定できる CONNECTION 定義を指定変更して、XCF アクセス方式を動的に選択します。

図 47 に MRO リンクの一般的な定義を示します。

```
DEFINE
  CONNECTION(CICB)      リモート・システムのローカル名
  GROUP(groupname)      関連定義のグループ名
  NETNAME(CICSB)        リモート・システムの applid
  ACCESSMETHOD(XM)      仮想記憶間サービス
  QUEUELIMIT(NO)        空きセッションがないとき、すべての要求をキューイング
  INSERVICE(YES)
  ATTACHSEC(LOCAL)      そのリンクだけのセキュリティーを使用
  USEDFLTUSER(NO)
DEFINE
  SESSIONS(csdname)     固有の csd 名
  GROUP(groupname)      接続と同じグループ
  CONNECTION(CICB)      関連する接続
  PROTOCOL(LU61)
  RECEIVEPFX(<)
  RECEIVECOUNT(5)       5 つの受信セッション
  SENDPFX(>)
  SENDCOUNT(3)          3 つの送信セッション
  SESSPRIORITY(100)
  IOAREALEN(300)        セッションの最小 TIOA サイズ
```

図 47. MRO リンクの定義例

互換 MRO ノードの定義

MRO リンクは、接続する両方のシステムで定義する必要があります。この 2 つの定義は相互に互換性がなければなりません。例えば、一方の定義で 6 つの送信セッションを指定するなら、他方の定義で 6 つの受信セッションを指定しなければなりません。

このタスクについて

互換性要件を図 48 に示します。

```
       CICSA                                  CICSB

DFHSIT  TYPE=CSECT                         DFHSIT  TYPE=CSECT
       ,APPLID=CICSA       1           4          ,APPLID=CICSB

DEFINE CONNECTION(CICB)    2           8   DEFINE CONNECTION(CICA)
    GROUP(PRODSYS)         3           9       GROUP(TESTSYS)
    NETNAME(CICSB)         4           1       NETNAME(CICSA)
    ACCESSMETHOD(IRC)                          ACCESSMEHOD(IRC)
    QUEUELIMIT(500)                            QUEUELIMIT(NO)
    MAXQTIME(500)
    INSERVICE(YES)                             INSERVICE(YES)
                                               ATTACHSEC(LOCAL)

DEFINE SESSIONS(SESS01)                    DEFINE SESSIONS(SESS02)
    GROUP(PRODSYS)         3           9       GROUP(TESTSYS)
    CONNECTION(CICB)       2           8       CONNECTION(CICA)
    PROTOCOL(LU61)         5           5       PROTOCOL(LU61)
    RECEIVEPFX(<)                              RECEIVEPFX(<)
    RECEIVECOUNT(8)        6           7       RECEIVECOUNT(10)
    SENDPFX(>)                                 SENDPFX(>)
    SENDCOUNT(10)          7           6       SENDCOUNT(8)
```

図 48. 互換 MRO ノードの定義

図 48 では、関連するオプションが同じ番号で示されています。

外部 CICS インターフェースが使用するリンクの定義

この項では、非 CICS プログラムが外部 CICS インターフェース (EXCI) を使って CICS サーバー・プログラムにリンクするための接続の定義方法を説明します。これに必要な定義は、CICS システム間の MRO リンクの場合と似ています。それぞれの接続には、CONNECTION と SESSIONS の定義が必要です。

EXCI 接続は外部ソースからの仕事を処理するために使用されますので、SEND セッションは定義できません。

EXCI 接続は「特定」または「総称」としての定義が可能です。特定 EXCI 接続とは、すべての RECEIVE セッションが単一ユーザー (クライアント・プログラム) の専用であるような MRO リンクのことです。総称 EXCI 接続とは、RECEIVE セッションが複数のユーザーによって共用される MRO リンクのことです。それぞれの CICS 領域には、1 つの総称 EXCI 接続しか定義できません。

特定接続と総称接続の定義では両方とも、次の指定が必要です。

- PROTOCOL(EXCI) を指定する。

- ACCESSMETHOD(IRC) を指定する。外部 CICS インターフェースでは、MRO 仮想記憶間アクセス方式 (XM) はサポートされません。システム間カップリング・ファシリティー (XCF) はサポートされます。
- SENDCOUNT と SENDPFX はデフォルトのブランクにする。

図 49 に特定 EXCI 接続の定義を示します。

```
DEFINE
  CONNECTION(EIP1)      接続のローカル名
  GROUP(groupname)      関連定義のグループ名
  NETNAME(CLAP1)        INITIALIZE_USER コマンド上のユーザー名
  ACCESSMETHOD(IRC)
  PROTOCOL(EXCI)
  CONNTYPE(Specific)    単一ユーザー専用のパイプ
  INSERVICE(YES)
  ATTACHSEC(LOCAL)
DEFINE
  SESSIONS(csdname)     固有の csd 名
  GROUP(groupname)      接続と同じグループ
  CONNECTION(EIP1)      関連する接続
  PROTOCOL(EXCI)        外部 CICS インターフェース
  RECEIVEPFX(<)
  RECEIVECOUNT(5)       5 つの受信セッション
  SENDPFX               ブランクのまま
  SENDCOUNT             ブランクのまま
```

図 49. 特定 EXCI 接続の定義例: 非 CICS クライアント・プログラムが外部 CICS インターフェースを使用する場合。

特定接続の場合、NETNAME には、EXCI INITIALIZE_USER コマンドで渡されるユーザー・プログラムの名前を指定します。CONNTYPE は特定のものでなければなりません。

図 50 に総称 EXCI 接続の定義を示します。

```
DEFINE
  CONNECTION(EIP2)      接続のローカル名
  GROUP(groupname)      関連定義のグループ名
  ACCESSMETHOD(IRC)
  NETNAME()             総称接続ではブランクでなければならない
  INSERVICE(YES)
  PROTOCOL(EXCI)
  CONNTYPE(Generic)     複数ユーザー共用のパイプ
  ATTACHSEC(LOCAL)
DEFINE
  SESSIONS(csdname)     固有の csd 名
  GROUP(groupname)      接続と同じグループ
  CONNECTION(EIP2)      関連する接続
  PROTOCOL(EXCI)        外部 CICS インターフェース
  RECEIVEPFX(<)
  RECEIVECOUNT(5)       5 つの受信セッション
  SENDPFX               ブランクのまま
  SENDCOUNT             ブランクのまま
```

図 50. 総称 EXCI 接続の定義例: 複数の非 CICS クライアント・プログラムが外部 CICS インターフェースを使用する場合。

総称接続の場合、NETNAME はブランクです。CONNTYPE は総称でなければなりません。

MRO と EXCI リンク定義のインストール

CICS が完全に操作可能な状態のままで、*新しい* MRO 接続や EXCI 接続を動的にインストールすることができます。このために領域間通信 (IRC) を終了する必要はありません。

CICS は接続定義のインストールをグループ・レベルでコミットすることに注意してください。接続や端末のインストールで失敗したものがあると、CICS はそのグループのすべての接続のインストールをバックアウトします。したがって、IRC がオープンされている状態の CICS 領域に新しい接続を追加する場合には、それらの新しい接続だけで 1 つのグループとしてください。

IRC がオープンされている間は、*既にある* MRO (または EXCI) リンクを修正することはできません。したがって、MRO リンクを定義するときには、予期されるワークロードに対応できるだけの SEND セッションと RECEIVE セッションを指定してください。

MRO リンクのインストールについては、「*CICS Resource Definition Guide*」の CONNECTION 定義の属性を参照してください。

APPC 接続の定義

APPC 接続は、1 つ以上のセッションで構成されています。各セット内のセッションは、競合勝者と競合敗者のいずれかであることを除けば、同一の特性を持ちます。

それぞれのセッション・セットにはモード名が割り当てられ、このモード名によって、そのセッション・セットが VTAM ログモード名にマップされ、そこからさらにサービス・クラス (COS) にマップされます。そのため、APPC セッションのセットをモードセットといいます。

APPC 端末は、単一のセッションだけをサポートし、LU サービス・マネージャーをサポートしない APPC システムであることがよくあります。このような端末を定義する方法がいくつかあります。詳細については、191 ページの『単一セッション APPC 端末の定義』で説明します。この項では、複数のセッションを含む 1 つまたは複数のモードセットの定義を説明します。

リモート・システムへの APPC 接続を定義するには、以下のことを行う必要があります。

1. DEFINE CONNECTION でリモート・システムを定義する。
2. DEFINE SESSIONS でリモート・システムとのセッションの各セットを定義する。

ただし、LU-LU ペア間に、同時に複数の APPC 接続をインストールすることはできません。また、LU-LU ペアの間で APPC 接続と LUTYPE6.1 接続を同時にインストールすることもできません。

APPC 端末への単一セッション接続を除くすべての APPC 接続で、CICS は、モード名 SNASVCMG を使用して、特殊なセッションのセットを自動的に作成し、LU サービス・マネージャーで排他的に使用します。これは予約名であるため、ユーザーが定義するセットに使用することはできません。

VTAM ログオン・モード・テーブルを定義する場合には、必ず SNASVCMG セッションの項目を含めてください。CICS の ACF/VTAM LOGMODE テーブル項目を参照してください。

リモート APPC システムの定義

これは、APPC システムの定義の形式です。

```
DEFINE
  CONNECTION(name)
  GROUP(groupname)
  NETNAME(name)
  ACCESSMETHOD(VTAM)
  PROTOCOL(APPC)
  SINGLESESS(NO)
  QUEUELIMIT(NO|0-9999)
  MAXQTIME(NO|0-9999)
  AUTOCONNECT(NO|YES|ALL)
  SECURITYNAME(value)
  ATTACHSEC(LOCAL|IDENTIFY|VERIFY|PERSISTENT|MIXIDPE)
  BINDPASSWORD(password)
  BINDSECURITY(YES|NO)
  USEDFLTUSER(NO|YES)
  PSRECOVERY(SYSDEFAULT|NONE)
```

図 51. APPC システムの定義

APPC システムの定義には、ACCESSMETHOD(VTAM) と PROTOCOL(APPC) の指定が必要です。CONNECTION 名 (つまり、sysidnt) とネット名の意味は、168 ページの『リモート・システムの識別』にあります (ただし、次の囲みを読んでください)。

重要:

VTAM 総称リソース・グループ / メンバーである端末専有領域への APPC リンクを定義する場合、NETNAME には、TOR の総称リソース名 かその applid を指定できます。 (208 ページの『XRF のための総称 applid と特定 applid』の VTAM 総称リソース名についての注を参照してください。) 総称リソースへの接続に対する NETNAME の指定方法のヒントについては、137 ページの『第 12 章 VTAM 総称リソースの構成』を参照してください。

この接続には複数のセッションがあるので、SINGLESESS(N) を指定するか、デフォルトとする必要があります。 (単一セッション APPC 端末の定義については、191 ページの『単一セッション APPC 端末の定義』で説明します。)

AUTOCONNECT オプションには、CICS の初期設定時に、その接続に対応するセッションのうちどれをバインドするのかを指定します。これについては、さらに 192 ページの『AUTOCONNECT オプション』で説明します。

QUEUELIMIT オプションには、リモート・システムとの空きセッションを待つ要求をいくつまでキューに入れることができるのかを指定します。MAXQTIME オプションには、キューがいっぱいになってから、リモート・システムが応答しないためにそれを除去するまでの時間の最大を指定します。これについては、さらに 307 ページの『第 24 章 システム間のセッション・キューの管理』で説明します。

VTAM の持続セッションのサポートを使用する場合、PSRECOVERY オプションには、ローカル CICS が失敗し、持続セッション遅延間隔内に再始動されたとき、リモート・システムとのセッションをリカバリーするかどうかを指定します。これについては、さらに 194 ページの『APPC リンクでの VTAM 持続セッションの使用』で説明します。

セキュリティー・オプションについては、「*CICS RACF Security Guide*」の LU6.2 セキュリティーの実装 (Implementing LU6.2 security)を参照してください。

注: LUTYPE6.1 リンクで実行されるように設計された既存のアプリケーションでシステム間リンクを使用する場合には、DATASTREAM と RECORDFORMAT オプションを使用すれば、非同期処理を行うためのデータ・ストリーム情報を指定することができます。これらのオプションによって与えられる情報は、APPC アプリケーション・プログラムでは使用されません。

APPC セッション・グループの定義

APPC システムの各セッション・グループは、DEFINE SESSIONS コマンドで定義します。

この定義を図 52 に示します。各セッション・グループは、**モードセット**と呼ばれます。

```
DEFINE
   SESSIONS(csdname)
   GROUP(groupname)
   CONNECTION(name)
   MODENAME(name)
   PROTOCOL(APPC)
   MAXIMUM(m1,m2)
   SENDSIZE(size)
   RECEIVESIZE(size)
   SESSPRIORITY(number)
   AUTOCONNECT(NO|YES|ALL)
   USERAREALEN(value)
   RECOVOPTION(SYSDEFAULT|UNCONDREL|NONE)
```

図 52. APPC セッション・グループの定義

CONNECTION オプションでは、そのグループが定義される APPC システムの名前 (1 から 4 文字) を指定します (つまり、対応する DEFINE CONNECTION コマンドの CONNECTION 名)。

MODENAME オプションでは、関連するセッションのグループを識別する名前 (1 から 8 文字) を指定することができます。この名前は、同じ APPC システム間リンクに対するモード名の中で固有でなければなりません。また、予約名 SNASVCMG や CPSVCMG は使用できません。

MAXIMUM(m1,m2) オプションには、そのグループでサポートされるセッションの最大数を指定します。このオプションのパラメーターの意味は次のとおりです。

- **m1** には、そのグループ内のセッションの最大数を指定します。デフォルトは 1 です。

- **m2** には、競合勝者としてサポートされるセッションの最大数を指定します。m2 に指定する数字は、m1 に指定する数字以下でなければなりません。m2 のデフォルトはゼロです。

RECEIVESIZE オプションは受信する要求単位 (RU) の最大サイズを指定しますが、これは 256 から 30 720 の範囲になければなりません。

AUTOCONNECT オプションには、CICS の初期設定時に、それらのセッションをバインドするかどうかを指定します。これについては、さらに 192 ページの『AUTOCONNECT オプション』で説明します。

VTAM の持続セッションのサポートを使用する場合、RECOVOPTION オプションには、CICS が失敗し、持続セッション遅延間隔内に再始動されたとき、それらのセッションを CICS によってどのようにリカバリーするかを指定します (RECOVNOTIFY オプションは APPC セッションには適用されません)。これについては、さらに 194 ページの『APPC リンクでの VTAM 持続セッションの使用』で説明します。

互換 CICS APPC ノードの定義

2 つの CICS システム間の APPC リンクを定義する場合、それらのシステムのリンク定義に互換性がなければなりません。

互換性要件の要約を図 53 に示します。

```
          CICSA                                  CICSB

DFHSIT  TYPE=CSECT                          DFHSIT  TYPE=CSECT
       ,APPLID=CICSA       1             3         ,APPLID=CICSB

DEFINE CONNECTION(CICB)    2            10  DEFINE CONNECTION(CICA)
    GROUP(groupname)                            GROUP(groupname)
    NETNAME(CICSB)         3             1      NETNAME(CICSA)
    ACCESSMETHOD(VTAM)                          ACCESSMEHOD(VTAM)
    PROTOCOL(APPC)                              PROTOCOL(APPC)
    SINGLESESS(N)          4             4      SINGLESESS(N)
    QUEUELIMIT(500)                             QUEUELIMIT(NO)
    MAXQTIME(500)                               ATTACHSEC(IDENTIFY)
    BINDPASSWORD(pw)       5             5      BINDPASSWORD(pw)


DEFINE SESSIONS(csdname)                     DEFINE SESSIONS(csdname)
    GROUP(groupname)                            GROUP(groupname)
    CONNECTION(CICB)       2            10      CONNECTION(CICA)
    MODENAME(M1)           6             6      MODENAME(M1)
    PROTOCOL(APPC)                              PROTOCOL(APPC)
    MAXIMUM(ss,ww)         7             7      MAXIMUM(ss,ww)
    SENDSIZE(kkk)          8             9      SENDSIZE(jjj)
    RECEIVESIZE(jjj)       9             8      RECEIVESIZE(kkk)
```

図 53. 互換 CICS APPC ISC ノードの定義

図 53 では、関連するオプションとオペランドが同じ番号で示されています。

注:

1. MAXIMUM に指定する値はリンクの両側で一致する必要はありません。これは、この値が LU サービス・マネージャーによって折衝で決められるからで

す。しかし、指定が一致していれば、使用されない TCTTE 項目が作られたり、『競合勝者』折衝によって予期しない送信権要求が出されたりすることが避けられます。

2. リンクの一方の側で指定した SENDSIZE の値が、他方で指定した RECEIVESIZE の値と一致しないと、CICS はこれらの値を BIND 時に折衝して決めます。

APPC リンクの自動インストール

CICS 自動インストール機能を使用すれば、APPC リンクを初めて使用するときに、それを動的に定義することができます。こうすることで、インストール済み定義のストレージや、それらの定義を作成するための時間が節約できます。

注: ここで説明する方法は、BIND 要求によって開始された APPC の並列セッションと単一セッションのリンクだけに適用されます。VTAM CINIT 要求によって開始された APPC 単一セッションのリンクで使う方法については、191 ページの『単一セッション APPC 端末の定義』で説明します。CINIT 要求によって開始された APPC 並列セッションのリンクを自動インストールすることはできません。

自動インストールが使用可能な場合、APPC サービス管理 (SNASVCMG) セッション (または単一セッション定義の唯一のセッション) に対する APPC BIND 要求が受信されたときに、一致する CICS CONNECTION 定義がないと、新しい接続が自動的に作成およびインストールされます。

端末の自動インストールと同じように、APPC リンクの自動インストールにもモデル定義が必要です。しかし、端末の自動インストールで使用するモデル定義と違い、APPC リンクの自動インストールに使用するモデル定義は、モデルとして明示的に定義する必要はありません。CICS は、インストール済みのリンク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。自動インストールが機能するためには、自動インストールを適用するリンクの種類ごとにテンプレートが必要です。

テンプレートの目的は、同じ特性をもつすべての接続に使用できる定義を CICS に提供することです。VTAM から受け取る情報に基づいて、新しいリンクごとに適切なテンプレートを選択するには、用意されている自動インストール・ユーザー・プログラム DFHZATDY をカスタマイズしてください。

テンプレートは、CONNECTION 定義とそれに対応する SESSIONS 定義からなります。必要になるセッション特性群ごとに 1 つの定義をインストールする必要があります。

インストール済みのリンク定義をテンプレートとして使用することはできますが、パフォーマンスを考慮すると、使用しないインストール済みリンク定義をテンプレートとする必要があります。定義は、CICS がそれをコピーしている間、ロックされますので、多数のセッションを自動インストールする場合には、それによる遅れが顕著になることがあります。

同じような特性をもつ APPC 並列セッション・デバイスがたくさんある場合には、おそらく、自動インストールのサポートによって利益が得られます。例えば、同じ特性のパーソナル・コンピューター (PC) が 1000 台あれば、1 つのテンプレート

でそれらすべてを自動インストールすることができます。また、ある特性の PC が 500 台、別の特性の PC が 500 台ある場合には、2 つのテンプレートを使ってそれらを自動インストールすることができます。

APPC リンクでの自動インストールの使用については、「*CICS Resource Definition Guide*」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・ユーザー・プログラムのプログラミング情報については、「*CICS Customization Guide*」を参照してください。

単一セッション APPC 端末の定義

単一セッション APPC 端末を定義するには、2 つの方法があります。つまり、CONNECTION と SESSIONS のペアを定義し、その接続に SINGLESESS(Y) を指定する方法と、TERMINAL と TYPETERM のペアを定義する方法です。

APPC 端末の定義 - 方法 1

CONNECTION と SESSIONS のペアを定義することによって、単一セッション APPC 端末を表すことができます。

このタスクについて

必要な DEFINE CONNECTION コマンドと DEFINE SESSIONS コマンドの形は、187 ページの図 51 と 188 ページの図 52 に示したものに似ています。違いを次に示します。

```
DEFINE CONNECTION(sysidnt)
       .
       SINGLESESS(Y)
       .
DEFINE SESSIONS(csdname)
       .
       MAXIMUM(1,0)
       .
```

この接続には、SINGLESESS(Y) を指定する必要があります。MAXIMUM オプションには、1 つのセッションを指定します。CICS は常に競合勝者としてバインドしますので、単一セッションの定義では 2 つめの値は意味がありません。しかし、CICS は、折衝バインドや折衝バインド応答を受け入れることによって、自身が競合敗者に変更されることはあります。

APPC 端末の定義 - 方法 2

TERMINAL およびそれに関連付けられた TYPETERM として、単一セッション APPC 端末を定義することができます。

このタスクについて

この定義方式には、次の 2 つの利点があります。

1. 単一の TYPETERM が同じタイプのすべての APPC 端末に使用できる。
2. AUTOINSTALL 機能が APPC 単一セッション端末で使用できる。

   VTAM CINIT によって開始された APPC 単一セッションの自動インストールは、TERMINAL と TYPETERM のモデル・ペアを提供しなければならないという点で、他の端末の自動インストールと同じように機能します。APPC 単一セッ

ション端末での自動インストールの使用については、「*CICS Resource Definition Guide*」のAPPC 接続の自動インストールを参照してください。

APPC 端末を定義する基本的な方法は次のとおりです。

```
DEFINE TERMINAL(sysid)
       MODENAME(modename)
       TYPETERM(typeterm)
       .
       .
DEFINE TYPETERM(typeterm)
       DEVICE(APPC)
       .
       .
```

すべての APPC デバイスは CICS にとってシステムに見えますので、TERMINAL オプションの名前は事実上システム名です。例えば、APPC 端末の照会を行う場合は、CEMT INQUIRE TERMINAL ではなく、CEMT INQUIRE CONNECTION を使用します。

DEFINE TERMINAL によって、単一の競合勝者セッションが暗黙指定されます。しかし、APPC 端末の場合、CICS は、自身が競合敗者に変更される折衝バインドを受け入れます。

CICS 提供の CSD グループ DFHTYPE には、APPC 端末に適した TYPETERM、DFHLU62T が含まれています。この TYPETERM をそのまま使用するか、あるいは独自の定義の基礎として使用することができます。

APPC 端末で自動インストールを使用する場合は、CICS 提供の CSD グループ DFHTERM に用意されているモデル端末定義 (LU62) が必要です。さらに、自動インストール・ユーザー・プログラムを作成して、適切な VTAM LOGMODE 項目を指定する必要があります。

TERMINAL と TYPETERM の定義、CICS 提供の CSD グループ、および自動インストールについての詳細は、「*CICS Resource Definition Guide*」を参照してください。VTAM LOGMODE 項目、および自動インストール・ユーザー・プログラムのプログラミング情報については、「*CICS Customization Guide*」を参照してください。

AUTOCONNECT オプション

DEFINE CONNECTION と DEFINE SESSIONS (APPC 端末の場合は、DEFINE TYPETERM) の AUTOCONNECT オプションを使用すると、リモート APPC システムとの通信を確立する CICS を制御することができます。

単一セッション APPC 端末 (191 ページの『単一セッション APPC 端末の定義』を参照) の場合を除き、リモート APPC システムとのセッションの確立には、次の 2 つのイベントが必要です。

1. リモート・システムへの接続が確立されなければならない。これは、LU サービス管理セッション (SNASVCMG) のバインドと、初期折衝の実行を意味します。
2. そのモードセットのセッションがバインドされなければならない。

これらのイベントは、一部を DEFINE CONNECTION コマンドの AUTOCONNECT オプションが、一部を DEFINE SESSIONS コマンドの AUTOCONNECT オプションが制御します。

DEFINE CONNECTION の AUTOCONNECT オプション

DEFINE CONNECTION コマンドの AUTOCONNECT オプションは、CICS ができるだけ早い時点 (VTAM ACB のオープン時) で LU サービス管理セッションをバインドするかどうかを指定します。この値には、次のものがあります。

AUTOCONNECT(NO)
CICS は、LU サービス管理セッションをバインド**しません**。

AUTOCONNECT(YES)
CICS は、LU サービス管理セッションをバインドしようと**します**。

AUTOCONNECT(ALL)
YES と同じです。これは、対応する DEFINE SESSIONS で ALL を指定すべきことを覚えておくために使用することも可能です。

リモート・システムが使用できなければ、LU サービス管理セッションは当然バインドされません。何らかの理由で、これらのセッションが CICS の初期設定中にバインドされない場合は、CEMT SET CONNECTION INSERVICE ACQUIRED コマンドによってそれらをバインドすることができます。また、リモート・システム自身が通信を開始した場合にもバインドされます。単一セッション APPC 端末の場合には、AUTOCONNECT(YES) や AUTOCONNECT(ALL) を DEFINE CONNECTION コマンドに指定しても効果はありません。これは、単一セッション接続には LU サービス・マネージャーがないためです。

DEFINE SESSIONS の AUTOCONNECT オプション

DEFINE SESSIONS コマンドの AUTOCONNECT オプションは、関連付けられた LU サービス管理セッションがバインドされたときに、どのセッションがバインドされるかを指定します。(これより前の時点でユーザー・セッションをバインドすることはできません)。

このオプションは、次の値をとることができます。

AUTOCONNECT(NO)
セッションはバインドされません。

AUTOCONNECT(YES)
競合勝者セッションがバインドされます。

AUTOCONNECT(ALL)
競合勝者セッションと競合敗者セッションがバインドされます。

AUTOCONNECT(ALL) を使用すると、CICS は、バインド要求を送信**できない**リモート・システムとの競合敗者セッションをバインドすることができます。
AUTOCONNECT(ALL) を指定すると、このシステムに元もと指定されている数以外に、CICS がいくつかの競合勝者をバインドできることになります。CICS がバインドする競合勝者の数は、セッション開始要求に対するパートナー・システムの応答 (CNOS 交換) によって異なります。CICS は、CNOS 応答で競合敗者と指定されない限り、すべての セッションを競合勝者としてバインドしようとします。例えば、このシステムに DEFINE SESSIONS MAXIMUM(10,4) を、リモート・システムに

DEFINE SESSIONS MAXIMUM(10,2) を指定して、モードグループを定義するとします。セッションがこのシステムから獲得され、競合敗者セッションのバインドが正常であれば、結果は 8 つの 1 次競合勝者セッションになります。

重要: 他の CICS システムや、バインド要求を送信する可能性のあるシステムとのセッションには、AUTOCONNECT(ALL) を決して指定しないでください。もし指定すると、CICS が解決できないバインド競争状態が起こるおそれがあります。

AUTOCONNECT(NO) を指定する場合には、CEMT SET MODENAME ACQUIRED AVAILABLE コマンドによって、セッションをバインドし、使用可能にすることができます。 (CEMT SET MODENAME コマンドの詳細については、「*CICS Supplied Transactions*」の CEMT SET MODENAMEを参照してください。) これを行わないと、セッションは、アプリケーション・プログラムの要求に従って、個々にバインドされます。

単一セッション APPC 端末の場合は、DEFINE SESSIONS か DEFINE TYPETERM の AUTOCONNECT に指定する値によって、CICS がその単一セッションをバインドするかどうかが決まります。

APPC リンクでの VTAM 持続セッションの使用

VTAM 持続セッションを使用すれば、APPC リンクの使用可能度を増すことができます。VTAM 持続セッションをサポートすることにより、CICS または VTAM に障害が発生した場合でも、ネットワークの流れなしでセッションをリカバリーすることができます。

「 *CICS Recovery and Restart Guide*」では、持続セッションをサポートしている場合に発生すること、および持続セッションのサポートなしで CICS 領域を実行する理由について説明しています。

CICS、VTAM、または z/OS に障害が発生したときに APPC セッションがアクティブである場合、持続セッションのリカバリーによって、APPC パートナーには CICS が停止しているように見えます。VTAM は、APPC パートナーによって出された要求を保存し、リカバリーが完了したときに、それらを CICS に渡します。CICS が VTAM で接続を再確立すると、端末セッションのリカバリーが CONNECTION リソース定義の PSRECOVERY オプションおよび SESSIONS リソース定義の RECOVOPTION オプションの設定によって決定されます。リカバリーするセッションについて、CONNECTION リソース定義の PSRECOVERY オプションをデフォルト値の SYSDEFAULT に設定する必要があります。もう 1 つの NONE を指定すると、セッションはリカバリーされません。適切なリカバリー・オプションを選択した場合で、APPC セッションが正しい状態にあるとき、CICS は ISSUE ABEND を実行して、現在の会話が異常終了したことをパートナーに通知します。

DEFINE CONNECTION の PSRECOVERY オプション

持続セッションがサポートされる CICS 領域において、この接続によって使用される APPC セッションを、持続セッション遅延間隔内に行われるシステム再始動の際にリカバリーするかどうか を指定するには、このオプションを使用します。この値には、次のものがあります。

SYSDEFAULT
障害のあった CICS システムが持続セッション遅延間隔内に再始動されると、次のことが起こります。

- ユーザー・モードグループが SESSIONS RECOVOPTION 値にリカバリーされる。
- SNASVCMG モードグループがリカバリーされる。
- 接続が ACQUIRED 状態に戻され、最後に折衝された CNOS 状態が戻される。

NONE
すべてのセッションは、CNOS のリカバリーなしにアンバインドされ、サービス不能となります。

DEFINE SESSIONS と DEFINE TYPETERM の RECOVOPTION オプション

持続セッションがサポートされる CICS 領域において、持続セッション遅延間隔内に行われるシステム再始動のあとに APPC セッションをどのように リカバリーするかを指定するには、DEFINE SESSIONS の RECOVOPTION オプションを使用します。

セッションを持続セッションにするためには、この値にデフォルト (SYSDEFAULT) を使用します。これによって、CICS は、持続セッション遅延間隔内に行われるシステム再始動の際、最適な手順を使ってセッションをリカバリーします。

単一セッション APPC 端末の場合は、DEFINE SESSIONS か DEFINE TYPETERM の RECOVOPTION オプションによって、持続セッション遅延間隔内に行われるシステム再始動のあとに端末をどのようにしてサービス状態に戻すかを指定します。

持続セッションのサポートがない場合は、端末に対し AUTOCONNECT(YES) が指定されていると、エンド・ユーザーは、GMTRAN トランザクションが実行されるまで、仕事を続けることはできません。AUTOCONNECT(NO) が指定されている場合には、(サポート・スタッフから教えてもらわない限り) CICS が再びいつ操作可能になるかは、ログオンを行わないと分かりません。どちらの場合にも、ユーザーは CICS から切断されますので、セッションを再び確立して、作業環境をリカバリーする必要があります。持続セッションのサポートがある場合には、セッションは、CICS の障害の際、リカバリー保留状態に置かれます。CICS が指定間隔内に開始され、RECOVOPTION が SYSDEFAULT に設定されていれば、ユーザーがセッションを再び確立して作業環境を回復する必要はありません。

SYSDEFAULT 値の詳細と、RECOVOPTION のその他の値については、「*CICS Resource Definition Guide*」のSESSIONS 定義の属性を参照してください。

CICS の持続セッション・サポートの詳細については、 351 ページの『第 28 章 相互通信と VTAM 持続セッション』を参照してください。

論理装置タイプ 6.1 リンクの定義

CICS-CICS 間通信には MRO リンクまたは APPC リンクの使用をお勧めします。

重要:

LUTYPE6.1 リンクは、LUTYPE6.1 プロトコルをサポートする一方で APPC プロトコルを完全にはサポートしない、IMS などのシステムと CICS との間のシステム間通信に必要です。

LU-LU ペアの間で APPC 接続と LUTYPE6.1 接続を同時にインストールすることはできません。

LUTYPE6.1 リンク上にリモート・システムを定義するには、必ず DEFINE CONNECTION が必要です。しかし、セッションは、次のいずれかの方法で定義することができます。

1. 単一の DEFINE SESSIONS コマンドを使用して、同じ特性をもつセッションのプールを定義する。
2. 別々の DEFINE SESSIONS コマンドを使用して個々のセッションを定義する。*個々のセッションを明示的に指定する必要がある IMS などのシステムとのセッションを定義する場合には、この方式を使用する必要があります。*

CICS-IMS 間 LUTYPE6.1 リンクの定義

IMS システムへのリンクの場合は、その接続 (またはシステム) を定義するとともに、各セッションを個別に定義する必要があります。

個々の LUTYPE6.1 セッションの定義の形式を、図 54 に示します。

```
DEFINE
  CONNECTION(sysidnt)
  GROUP(groupname)
  NETNAME(name)
  ACCESSMETHOD(VTAM)
  PROTOCOL(LU61)
  DATASTREAM(USER|3270|SCS|STRFIELD|LMS)
  RECORDFORMAT(U|VB)
  QUEUELIMIT(NO|0-9999)
  MAXQTIME(NO|0-9999)
  INSERVICE(YES)
  SECURITYNAME(name)
  ATTACHSEC(LOCAL)
次に、個々のセッションが通常次のように定義されます。
DEFINE
  SESSIONS(csdname)
  GROUP(groupname)
  CONNECTION(sysidnt)
  SESSNAME(name)
  NETNAMEQ(name)
  PROTOCOL(LU61)
  RECEIVECOUNT(1|0)
  SENDCOUNT(0|1)
  SENDSIZE(size)
  RECEIVESIZE(size)
  SESSPRIORITY(number)
  AUTOCONNECT(NO|YES|ALL)
  BUILDCHAIN(YES)
  IOAREALEN(value)
```

図 54. LUTYPE6.1 リンクと個々のセッションの定義

互換の CICS ノードと IMS ノードの定義

この項では、対応する IMS 定義と互換性のある適切な CICS 定義を作成する方法について説明します。

IMS システム定義の概要については、131 ページの『第 10 章 システム間通信の構成』で説明します。CICS 定義と IMS 定義の関係については、その要約を 200 ページの図 55 で示します。

システム名

CICS システムのネットワーク名 (その applid) は、APPLID CICS システム初期設定パラメーターに指定します。

この名前は、その CICS システムを定義する IMS TERMINAL マクロの NAME オペランドに指定する必要があります。XRF を使用する CICS システムの場合、その名前は CICS の総称 applid です。XRF を使用しない CICS システムの場合、その名前は APPLID SIT パラメーターに指定された単一の applid です (208 ページの『XRF のための総称 applid と特定 applid』を参照)。

IMS システムのネットワーク名は、いくつかの方法で指定できます。

- XRF サポートのあるシステムの場合、IMS.PROCLIB の DFSHSBxx メンバーに定義される USERVAR として指定する。
- XRF のないシステムの場合、
  - IMS COMM マクロの APPLID オペランドに指定する。
  - IMS 始動ジョブの EXEC ステートメントのラベルとして指定する (APPLID の指定が NONE の場合)。
  - 開始済みタスク名として指定する (APPLID の指定が NONE の場合)。

IMS システムのネットワーク名は、その IMS システムを定義する CICS の DEFINE CONNECTION コマンドの NETNAME オプションに指定する必要があります。

セッションの数

IMS では、CICS システムと IMS システムの間に必要な並列セッションの数は、IMS TERMINAL マクロの SESSION オペランドに指定する必要があります。

各セッションは、IMS VTAMPOOL 内の SUBPOOL 項目によって表されます。CICS では、これらの各セッションは、個々のセッション定義によって表されます。

セッション名

CICS-IMS 間の各セッションは、セッションと修飾子の対によって固有に識別されます。これは、そのセッションの CICS 名とそのセッションの IMS 名から構成されます。

セッションの CICS 名は、DEFINE SESSIONS コマンドの SESSNAME オプションに指定されます。IMS によって開始されるセッションの場合、この名前は、そのセッションに対する IMS OPNDST コマンドの ID パラメーターに対応していなければなりません。CICS によって開始されるセッションの場合、この名前は、CICS OPNDST コマンドに指定され、IMS によって保管されます。

セッションの IMS 名は、IMS SUBPOOL マクロの NAME オペランドに指定されます。これらのセッション名の間の関係は、対応する DEFINE SESSIONS コマンドの NETNAMEQ オプションにこの名前をコーディングすることによって、明示的に指定しなければなりません。

セッションの CICS 名と IMS 名は同じにすることができます。この方法は操作を容易にするうえで推奨されます。

その他のセッション・パラメーター

この項では、DEFINE CONNECTION コマンドと DEFINE SESSIONS コマンドのその他のオプションのうち、CICS-IMS 間セッションに重要なものをリストします。

ATTACHSEC
LOCAL と指定しなければなりません。

BUILDCHAIN(YES)
複数の RU チェーンがアセンブルされてからアプリケーション・プログラムに渡されます。それぞれの RECEIVE コマンドへの応答として、完全なチェーンがアプリケーション・プログラムに渡されます。アプリケーションは必要な非ブロック化を行います。

LUTYPE6.1 セッションには、BUILDCHAIN(YES) を指定 (または、デフォルトを使用) しなければなりません。

DATASTREAM(USER)
値 USER を指定するか、デフォルトを使用しなければなりません。

このオプションは、CICS が、START コマンド (非同期処理) を使用して IMS と通信する場合にのみ使用されます。START コマンドによって CICS メッセージが生成されると、IMS は常にそのデータ・ストリーム・プロファイルをコンポーネント 1 への入力として解釈します。

分散トランザクション処理のためのデータ・ストリーム・プロファイルは、BUILD ATTACH コマンドの DATASTR オプションを使って、アプリケーション・プログラムから指定することができます。

QUEUELIMIT(NO|0-9999)
リモート・システムとの空きセッションの要求をいくつまでキューイングできるかを指定します。これについては、さらに 307 ページの『第 24 章 システム間のセッション・キューの管理』で説明します。

MAXQTIME(NO|0-9999)
リモート・システムとのセッションのキューがいっぱいになってから (つまり、QUEUELIMIT に指定された限界に達してから)、リモート・システムの応答がないためそのキューが除去されるまでの時間の最大 (秒) を指定します。これについては、さらに 307 ページの『第 24 章 システム間のセッション・キューの管理』で説明します。

RECORDFORMAT(U|VB)
START コマンド (非同期処理) によって開始されたこのセッションでの伝送に CICS が使用するチェーンのタイプを指定します。

CICS と IMS の間で 2 種類のデータ処理アルゴリズムがサポートされます。

チェーン
メッセージは SNA チェーンとして送信されます。ユーザーは、専用のブロック化と非ブロック化のアルゴリズムが使用できます。この形式は RECORDFORMAT(U) に対応します。

可変長可変ブロック化レコード (VLVB)
メッセージは、各レコードの前にハーフワードの長さのフィールドをもつ可変長可変ブロック化形式で送信されます。この形式は RECORDFORMAT(VB) に対応します。

分散トランザクション処理のためのデータ・ストリーム形式は、BUILD ATTACH コマンドの RECFM オプションを使って、アプリケーション・プログラムから指定することができます。

これらのデータ形式については、さらに 283 ページの『第 23 章 CICS-IMS 間アプリケーション』で説明します。

SENDCOUNT および RECEIVECOUNT
セッションが SEND セッションなのか RECEIVE セッションなのかを指定します。

SEND セッションとは、ローカル CICS が 2 次で、競合勝者であるセッションのことをいいます。次のように指定します。

- SENDCOUNT(1)
- RECEIVECOUNT にはデフォルトを使用します。RECEIVECOUNT(0) と指定しないで ください。

RECEIVE セッションとは、ローカル CICS が 1 次で、競合敗者であるセッションのことをいいます。次のように指定します。

- RECEIVECOUNT(1)
- SENDCOUNT にはデフォルトを使用します。SENDCOUNT(0) と指定しないで ください。

すべての CICS-IMS 間セッションには、SEND セッションを使用するようにしてください。

SENDPFX や RECEIVEPFX を指定する必要はありません。セッションの名前は SESSNAME オプションからとられます。

SENDSIZE および RECEIVESIZE
これらのセッションに VTAM 要求単位 (RU) の最大サイズを指定します。

- CICS が 1 次側ハーフセッションである場合、以下の項目を確認します。
  1. CICS SENDSIZE が IMS COMM マクロの RECANY パラメーターで指定した値以下であること。
  2. CICS RECEIVESIZE が IMS OUTBUF サイズ以上であること。
- IMS が 1 次側ハーフセッションである場合、以下の項目を確認します。
  1. CICS SENDSIZE が IMS OUTBUF サイズ以上であること。
  2. CICS RECEIVESIZE が IMS RECANY サイズ以下であること。

```
          CICS                                IMS
  DFHSIT TYPE=CSECT                     COMM      APPLID=SYSIMS

       ,SYSIDNT=CICL                  7             RECANY=nnn+22
       ,APPLID=SYSCICS     1                        EDTNAME=ISCEDT

                                      4   TYPE      UNITYPE=LUTYPE6

                                      1   TERMINAL  NAME=SYSCICS
DEFINE CONNECTION(IMSR)    2                        SESSION=2
    GROUP(groupname)                               COMPT1
    NETNAME(SYSIMS)        3                       COMPT2
    ACCESSMETHOD(VTAM)                 6           OUTBUF=mmm
    PROTOCOL(LU61)
    DATASTREAM(USER)
    ATTACHSEC(LOCAL)

DEFINE SESSIONS(csdname)
    GROUP(groupname)                        VTAMPOOL
    CONNECTION(IMSR)       2
    SESSNAME(IMS1)                     5    SUBPOOL   NAME=CIC1
    NETNAMEQ(CIC1)         5                NAME      CICLT1 COMPT=1
    PROTOCOL(LU61)         4
    SENDCOUNT(1)
    SENDSIZE(nnn)          7                NAME       CICLT1A
    RECEIVESIZE(mmm)       6
    IOAREALEN(nnn,16364)

DEFINE SESSIONS(csdname)
    GROUP(groupname)
    CONNECTION(IMSR)       2
    SESSNAME(IMS2)                     8    SUBPOOL   NAME=CIC2
    NETNAMEQ(CIC2)         8                NAME      CICLT2 COMPT=2
    PROTOCOL(LU61)         4
    SENDCOUNT(1)
    SENDSIZE(nnn)          7           3    DFSHSBxx  USERVAR=SYSIMS
    RECEIVESIZE(mmm)       6
    IOAREALEN(nnn,16364)

Note: For SEND sessions, allow RECEIVECOUNT to default. For RECEIVE
      sessions, allow SENDCOUNT to default.
```

注: SEND セッションの場合、RECEIVECOUNT にはデフォルトを使用してください。RECEIVE セッションの場合、SENDCOUNT にはデフォルトを使用してください。

図 55. 互換の CICS ノードと IMS ノードの定義

図 55 は、システム間リンクの CICS 定義と IMS 定義の関係を示したものです。関連するオプションとオペランドは同じ番号で示してあります。

注: IMS 用の VTAM ログモード・テーブル項目の例については、IMS の ACF/VTAM LOGMODE テーブル・エントリーを参照してください。

IMS システムへの複数リンクの定義

CICS と IMS システムの間には、複数のシステム間リンクを定義することができます。

このタスクについて

そのためには、同じネット名と異なる sysidnt を指定して複数の CONNECTION 定義 (とそれぞれに対応する SESSION 定義) を作成する必要があります (202 ページの図 56)。すべてのシステム定義が同じネット名 (したがって同じ IMS システム)

になりますが、CICS で sysidnt 名を使用すれば、CICS は、指定された sysidnt をもつリンクからセッションを割り振ります。

CICS システムと IMS システムの間には、ご使用のシステムのアプリケーションの必要性に応じて、最大 3 つのリンク (つまり、最大 3 つのセッション・グループ) を定義するようにしてください。

1. CICS によって開始される分散トランザクション処理 (同期処理) の場合、

   SEND/RECEIVE インターフェースを使用する CICS アプリケーションは、このグループの sysidnt を使って、リモート・システムとのセッションを割り振ることができます。このセッションは、会話が終了するまで保持されます (「使用中」)。
2. CICS によって開始される非同期処理の場合、

   START コマンドを使用する CICS アプリケーションは、このグループの sysidnt を指定することができます。CICS は、最初の「使用中ではない」セッションを使用して、開始要求をシップします。

   IMS はその開始要求をキューイングするとすぐに CICS に肯定応答を送るので、セッションが使用中になるのは比較的短時間で済みます。したがって、グループ内の最初のセッションが最も頻繁に使用され、グループ内の後のセッションほど使用頻度が少なくなります。
3. IMS によって開始される非同期処理の場合、

   このグループは、CICS によって開始された非同期処理で生じることがあるパフォーマンスの問題の解決策の一助としても有効です。特定のセッションでシップされた START コマンドによって IMS トランザクションが開始されると、そのトランザクションは、同じセッションを使用して、「応答」START コマンドを CICS にシップします。上記の (2) に示した理由により、CICS START コマンドは、おそらく最も使用頻度の高いセッションでシップされています。そして、セッションが使用中で CICS が競合勝者であるため、IMS からの応答は、そのセッションが使用できるまでキューに入れられることがあります。

   しかし、IMS にはトランザクションがそのデフォルトの出力セッションを変更する機能があります。そして、この 3 番目のグループのセッションに切り替えることによって、この種のキューイングの問題を減らすことができます。

```
  DFHSIT TYPE=CSECT,
         SYSIDNT=CICL,
         APPLID=SYSCICS
CICS によって開始される分散トランザクション処理
  DEFINE CONNECTION(IMSA)
         NETNAME(SYSIMS)
         ACCESSMETHOD(VTAM)
  DEFINE SESSIONS(csdname)
         CONNECTION(IMSA)
         SESSNAME(IMS1)
         NETNAMEQ(DTP1)
         PROTOCOL(LU61)
  DEFINE SESSIONS(csdname)
         .
         .
CICS によって開始される非同期処理
  DEFINE CONNECTION(IMSB)
         NETNAME(SYSIMS)
         ACCESSMETHOD(VTAM)
  DEFINE SESSIONS(csdname)
         CONNECTION(IMSB)
         SESSNAME(IMS1)
         NETNAMEQ(ASP1)
         PROTOCOL(LU61)
  DEFINE SESSIONS(csdname)
         .
         .
IMS によって開始される非同期処理
  DEFINE CONNECTION(IMSC)
         NETNAME(SYSIMS)
         ACCESSMETHOD(VTAM)
  DEFINE SESSIONS(csdname)
         CONNECTION(IMSC)
         SESSNAME(IMS1)
         NETNAMEQ(IST1)
         PROTOCOL(LU61)
  DEFINE SESSIONS(csdname)
         .                                .
         .                                .
```

図 56. IMS ノードへの複数リンクの定義

トランザクション・ルーティングのための間接リンクの定義

(サポートされなくなった) 古い CICS リリースの中には、中間領域でトランザクション・ルーティングを行うために、CICS 領域間での間接リンクを必要とするものがありました。現在使用可能な CICS システムのみで構成されているネットワークでは、間接リンクが必要になるのは、非 VTAM 端末を使用している場合にかぎられます。必要に応じて、間接リンクを VTAM 端末に使用するために定義することもできます。間接リンクは、機能シップ、分散プログラム・リンク、非同期処理、または分散トランザクション処理には使用されません。

次の図は、間接リンクの概念を示したものです。

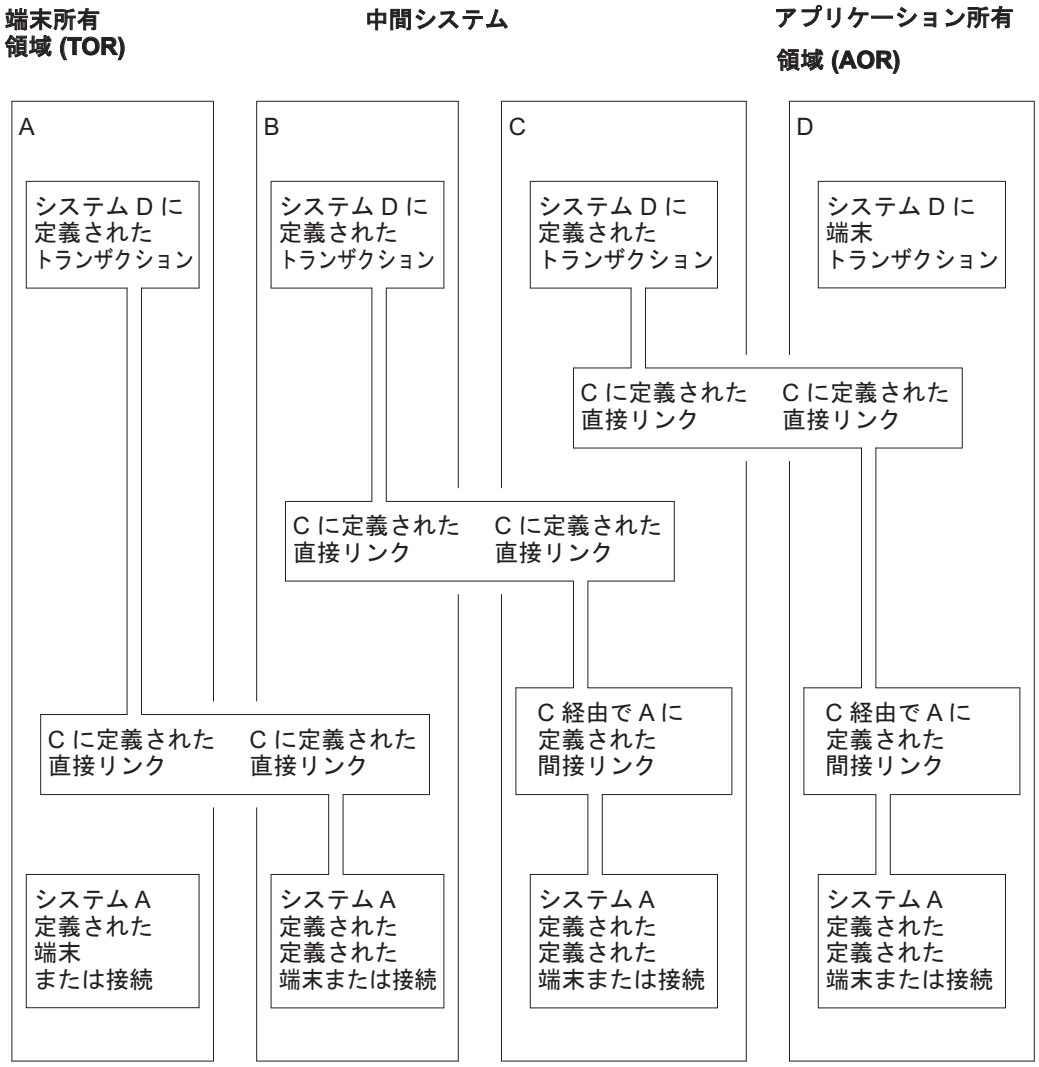


図 57. トランザクション・ルーティングのための間接リンク

この図は、MRO リンクか APPC リンクによってリンクされるシステム (A、B、C、D) のチェーンを示しています (LUTYPE6.1 リンクによってトランザクション・ルーティングを行うことはできない)。

端末専有領域 A とアプリケーション所有領域 D の間にトランザクション・ルーティング・パスを確立する場合を想定します。システム A と D の間の直接リンクはありませんが、**中間**システム B と C を経由するパスが使用できます。

トランザクション・ルーティング要求がこのパスを通るためには、端末 (APPC 接続の場合もある) とトランザクションのリソース定義が両方とも 4 つのシステムすべてで使用できなければなりません。端末は、端末所有システム A のローカル・リソースであり、システム B、C、D のリモート・リソースです。同様に、トランザクションは、トランザクション所有システム D のローカル・リソースであり、システム A、B、C のリモート・リソースです。

CICS Transaction Server for z/OS での間接リンクの定義

CICS システムは、固有 ID を使用してリモート端末を参照します。この固有 ID は、端末専有領域 (TOR) の applid (ネット名) と、端末専有領域で端末の認識に使用される ID から構成されます。

リモート・リソース定義について詳しくは、 223 ページの『第 16 章 リモート・リソースの定義』を参照してください。

CICS が完全修飾の端末 ID を形成するためには、その TOR のネット名にアクセスできなければなりません。(サポートされなくなった) 古い CICS リリースでは、間接リンクを定義する目的が 2 つありました。TOR への直接リンクがない場合、間接リンクによって、

1. その端末専有領域のネット名を提供する。
2. その端末専有領域へのパスの始まりである**直接**リンクを識別する。

したがって、 203 ページの図 57 において、システム D の間接リンク定義は、システム A のネット名を提供し、システム C をそのパスにおける次のシステムとして識別します。同様に、システム C の間接リンク定義は、システム A のネット名を提供し、システム B をそのパスにおける次のシステムとして識別します。システム B は、システム A への直接リンクをもつため、間接リンクは必要ありません。

CICS Transaction Server for z/OS では、非 VTAM 端末を使用する場合を除いて、間接リンクの指定はオプションです。シップ可能な端末定義を使用するか、ハードコーディングされた端末定義を使用するかによって、考慮事項が異なります。

シップ可能端末

中間システム経由で端末定義を AOR にシップする場合、間接リンクは必要ありません。シップされる定義にはそれぞれ、トランザクション・ルーティング・パスにおける前のシステム (または、その TOR への間接接続がある場合には、その間接接続) へのポインターがあります。そのため、ルーティングされたトランザクションは、TOR のネット名と AOR から TOR へのパスの指定により、接続されます。

パスがいくつかある場合には、間接リンクを使って、TOR への優先パスを指定することができます。

注: 非 VTAM 端末はシップ可能ではありません。

ハードコーディング端末

VTAM 端末だけを使用する場合には、間接リンクは必要ありません。TOR のネット名を指定するには、TERMINAL 定義 (その「端末」が APPC デバイスの場合は、CONNECTION 定義) の REMOTESYSNET オプションを、TOR へのパスにおける次のシステムを指定するには、REMOTESYSTEM オプションを、それぞれ使用します。パスがいくつかある場合には、REMOTESYSTEM を使って、優先パスにおける次のシステムを指定してください。

非 VTAM 端末を使用する場合には、間接リンクが必要です。これは、RDO を使って非 VTAM 端末を定義することができないからです。リモート定義を作成するのに使用する DFHTCT TYPE=REMOTE または TYPE=REGION マクロには、CEDA DEFINE TERMINAL の REMOTESYSNET オプションに相当するものはありません。

このように、CICS Transaction Server for z/OS では、次のような場合に間接リンクを定義することができます。

- 複数のパスがあり、シップ可能端末を使っている場合に、TOR への優先パスを指定する。
- 中間システムを経由するトランザクション・ルーティングに非 VTAM 端末を使用する場合。
- REMOTESYSNET オプションを指定していない既存のリモート端末定義を使用できるようにする。例えば、数百のリモート VTAM 端末がバックレベルのシステムに定義されているとします。新しい CICS Transaction Server for z/OS バックエンド・システムをネットワークに導入する場合には、既存の定義を新しいシステムの CSD へコピーすることができます。そのネットワークの構造に TOR への直接リンクがない場合は、コピーしたすべての定義に REMOTESYSNET オプションを指定するよりも、間接リンクを 1 つ定義する方が簡単です。

間接リンクを使用するトランザクション・ルーティングのためのリソース定義

この項では、端末専有領域 SYS01 とアプリケーション所有領域 SYS04 の間で、2 つの中間システム SYS02 と SYS03 を経由し、間接リンクを使って、トランザクション・ルーティング・パスを確立するために必要なリソース定義について説明します。

必要なリソース定義を 206 ページの図 58 に示します。

注: 分かりやすくするために、図には、REMOTESYSNET オプションを使用しないハードコーディングされたリモート端末の定義を示しています (REMOTESYSNET を使用していれば、間接リンクは必要ありません)。シップ可能端末も同じように使用することが可能です。

図 58. トランザクション・ルーティングのための間接リンクの定義： SYS04 と SYS03 のリモート端末定義には REMOTESYSNET オプションが指定されていないので、間接リンクが必要です。

直接リンクの定義

SYS01 と SYS02、SYS02 と SYS03、SYS03 と SYS04 の間の直接リンクは、この章で前述した方法で定義された MRO リンクか APPC リンクです。

間接リンクの定義

TOR への間接リンクをトランザクション・ルーティング・パスにある一部のシステムには定義し、他のシステムには定義しないということが可能です。どのようにするかは、ネットワークの構造と、リモート端末定義の指定方法によって決まります。

例えば、中間システムの 1 つで、REMOTESYSNET が指定されていないハードコーディングされた端末定義が使用されていて、そのシステムに TOR への直接リンクがない場合には、間接リンクが必要になります。端末専有領域への直接リンクがあるシステムには、間接リンクは必要ありません。

この例の場合、間接リンクは SYS04 と SYS03 に定義されます。間接リンクの定義には、次の規則が適用されます。

- ACCESSMETHOD は INDIRECT でなければならない。
- NETNAME は端末専有領域の applid でなければならない。
- INDSYS (間接システムを示す) は、端末専有領域へのパスの始まりである、MRO リンクか APPC リンクの CONNECTION 名を指定しなければならない。
- 間接接続には SESSIONS 定義は必要ない。INDSYS オプションに指定する直接リンクのセッションが使用されます。

端末の定義

シップ可能端末を使用する場合には、リモート端末定義は必要ありません。

リモート端末と接続を CICS Transaction Server for z/OS システムに定義する方法としては、223 ページの『第 16 章 リモート・リソースの定義』で説明されている方法を推奨します。

206 ページの図 58 に示すハードコーディングされたリモート端末定義には、REMOTESYSNET オプションが指定されていません。これを使用する場合には、

- リモート端末定義や接続定義の REMOTESYSTEM (または SYSIDNT) オプションに、TOR へのリンク (つまり、CONNECTION 定義。この定義の NETNAME によって端末専有領域の applid を指定する) を必ず指定する必要があります。
- 指定するリンクは、端末専有領域への直接リンク (ある場合) でなければなりません。それがない場合は、間接リンクでなければなりません。

トランザクションの定義

リモート・トランザクションの定義については、223 ページの『第 16 章 リモート・リソースの定義』で説明します。

XRF のための総称 applid と特定 applid

XRF を使用する CICS システムには、総称名と特定名の 2 つ の applid 名があります。XRF を使用する場合には、両方の名前を APPLID パラメーターに指定しなければなりません。これは、活動 CICS システムと代替 CICS システムでは、総称 applid が同じで、特定 applid が異なっていなければならないためです。

APPLID システム初期設定パラメーターを使用して、総称名と特定名を指定します。

```
APPLID(=generic-applid,specific-applid)
```

- CICS 領域の**総称 applid** は、CICS がそのネットワークで認識される名前です。つまり、これは、この CICS を識別するために、リモート CICS 領域によって CONNECTION 定義の NETNAME オプションまたは IPCONN 定義の APPLID オプションで引用される名前です。
- CICS システムの**特定 applid** は、ペアになっている XRF システムのそれぞれを区別するために使用されます。これは、VTAM APPL ステートメントの中で、この CICS を VTAM に指定するために引用されている名前です。

重要:

「*総称 applid*」と「*総称リソース名*」を混同しないでください。総称 applid と特定 applid は、XRF を使用するシステムだけに適用されます。XRF を使用しない CICS システムには applid は 1 つしかありません。

CICS システムの総称リソース名は、**GRNAME** システム初期設定パラメーターに定義されるもので、これによって、CICS は VTAM 総称リソース・グループのメンバーになります。137 ページの『第 12 章 VTAM 総称リソースの構成』を参照してください。

VTAM 総称リソースと XRF を両方とも使用することはできません。VTAM 総称リソースを使用する場合、APPLID システム初期設定パラメーターには、名前を 1 つだけ指定してください。

第 14 章 TCP/IP 管理および制御

TCP/IP 管理および制御を使用して、CICS over TCP/IP (Transmission Control Protocol/Internet Protocol) 接続を行う作業または切断する作業をモニターすることができます。

TCP/IP 管理および制御により、TCP/IP ネットワークには、APPC ネットワークに既に提供されている管理機能のサブセットと、APPC または MRO ネットワークでは使用できない追加機能が提供されます。

TCP/IP ネットワークは、次のような方法で相互接続されるシステムです。

- IPIC 接続 (IPCONN)。

  IPIC では、以下の相互通信機能とリリースがサポートされています。
  - CICS TS 3.2 以降の領域との間の分散プログラム・リンク (DPL) 呼び出し
  - CICS TS および TXSeries バージョン 7.1 との間の分散プログラム・リンク (DPL) 呼び出し
  - 非同期に行われる非端末 EXEC CICS START、START CHANNEL、および CANCEL の各コマンドの機能シップ (CICS TS 4.1 領域間) に制限される処理
  - 3270 端末の従来のトランザクション・ルーティング (この端末では、端末専有領域 (TOR) は APPLID によって CICS TS 4.1 領域間で固有に識別されます)
  - CICS Transaction Gateway バージョン 7.1 以降からの ECI 要求
- Web インターフェースや、IIOP、CICS へのインバウンド SOAP over HTTP 要求などを伝送する、クライアントからの TCP/IP 接続。

TCP/IP 管理および制御を使用して、以下の機能などを実行することができます。

- CICSPlex SM または同等のツールは、以下のような目的で使用します。
  - TCP/IP ネットワークの CICSplex の全体像を把握することができます。
  - リアルタイムで以下について検査できます。
    - ある特定の CICS 領域が使用中の TCP/IP ネットワーク・リソース
    - TCP/IP ネットワークを介してある特定の CICS 領域を出入りする作業
    - TCP/IP ネットワークを介して CICSplex 全体を流れる分散トランザクションに関連付けられている CICS のリソースおよびタスク
    - 分散トランザクションが発信された CICS 領域
- CICS によって収集されたデータを保管して、関係しているタスクおよびリソースが使用できなくなった後のあるポイントで、オフライン検査をできるようにすることが可能です。

TCP/IP 管理および制御を次のような目的で使用することができます。

- 接続問題の診断
- トランザクション遅延などの問題の調査
- CICSplex 全体にわたる作業の追跡

© Copyright IBM Corp. 1977, 2009

- キャパシティー・プランニングに使用するシステム・データの長期にわたる収集
- CICSplex のモニター

役に立つ SPI コマンド

以下のシステム・プログラミング・インターフェース (SPI) コマンドを使用して、IPICによる接続についての情報を取得することができます。

EXEC CICS EXTRACT STATISTICS
RESTYPE「IPCONN」を指定して、IPIC のリソース統計を取得します。グローバル統計は使用不可です。

EXEC CICS INQUIRE ASSOCIATION
TCP/IP ネットワークでは、このコマンドはタスクに関する情報を返します。例えば、タスクが開始された方法、その開始を要求した TCP/IP クライアントの IP アドレスなどです。タスクはタスク番号で指定されます。タスク番号は、EXEC CICS INQUIRE ASSOCIATION LIST コマンドによって、番号リストの 1 つとして通常は返されます。

EXEC CICS INQUIRE ASSOCIATION LIST
このコマンドは、関連したデータ制御ブロック (ADCB) のユーザー相関データを突き合わせるタスクのリスト (ローカル領域にある) を返します。通常、ユーザー相関データは、CICS XAPADMGR グローバル・ユーザー出口プログラムによって、分散トランザクションの起点で追加されています。 211 ページの『XAPADMGR グローバル・ユーザー出口』を参照してください。

EXEC CICS INQUIRE TASK
IPALTFACILITIES オプションは、ID リストのアドレスを返します。各 ID は、タスクが別のシステムとの通信に使用した IPCONN セッションを識別します。LISTSIZE オプションは、リストの項目数を返します。

EXEC CICS PERFORM STATISTICS
統計タイプ「IPCONN」を指定して、IPIC 接続のリソース統計を記録します。グローバル統計は使用不可です。

ソケット・アプリケーション・データ (ApplData)

CICS は、所有する TCP ソケットごとに、40 バイトのソケット・アプリケーション・データ (ApplData) を生成します。CICS は SIOCSAPPLDATA IOCTL ソケット機能を使用して、この情報と z/OS Communications Server の TCP/IP ソケットを関連付けます。この情報を使用して、TCP/IP 接続と、CICS 領域およびその領域を使用するトランザクションを相互に関連付けることができます。

CICS では、CECI INQUIRE ASSOCIATION トランザクション、CICSPlex SM 表示、および SMF レコードを使用して、ApplData 情報を取得することができます。TCP/IP では、Netstat の ALL/-A、ALLConn/-a、および COnn/-c レポートで ApplData 情報を取得することができ、APPLD/-G フィルターで検索することができます。Netstat での ApplData の使用に関する追加情報については、「*System Administrator's Commands*」を参照してください。ApplData 情報は、SMF 119 TCP Connection Termination レコードで取得することができます。追加情報については、「*IP Configuration Reference*」を参照してください。ApplData 情報は、ネットワーク管理インターフェースを介して取得できます。詳しくは、「*IP Programmer's*

Guide and Reference」を参照してください。

XAPADMGR グローバル・ユーザー出口

使用可能になっている場合、この出口プログラムは、入力発信元記述子レコードが提供されていない非システム・タスクに接続したときに呼び出されます。

XAPADMGR 出口についての詳細は、「*CICS Customization Guide*」を参照してください。

CICS では、XAPADMGR 出口点で使用するためのサンプル・グローバル・ユーザー出口プログラム DFH$APAD を提供しています。使用可能になっている場合、この出口プログラムは、入力発信元記述子レコードが提供されていない非システム・タスクに接続したときに、呼び出されます。

DFH$APAD は次の処理を実行します。

- 出口に入力データとして提供された関連データへのアドレス可能度を提供します。
- このデータからフィールドを選択し、これを出力バッファーに置きます。
- 出力バッファーのユーザー相関データにフィールドを追加します。

CICSPlex SM を使用した TCP/IP トラフィックの分析

『XAPADMGR グローバル・ユーザー出口』に記載されているように、分散トランザクションの起点にある、タスクの関連データの起点記述子に追加されたユーザー相関情報は、CICSPlex SM を介して後で処理を遂行する際に検索キーとして使用することができます。

検索キー (または「フィルター・ストリング」) には、以下の「ワイルドカード」文字を含めることができます

? 任意の 1 文字に完全一致

***** 文字数がゼロ個以上の任意の文字列に一致

ワイルドカードのないフィルター・ストリングは、相関関係子の全体と完全に一致しなければなりません。したがって、任意のユーザー相関関係子ストリングに一致させるために、相関関係子のサブストリングであるフィルター・ストリングには、少なくとも 1 つのワイルドカード文字が含まれていなければなりません。例えば、データ内のどこかに存在する可能性があるサブストリングを見つけるには、フィルター・ストリングの前と後ろの両方に「*」を追加します。

CICSPlex SM TASKASSC リソース・テーブルは、分散トランザクションを構成するタスクについての情報を提供します。レコードは、ユーザー相関データのサブストリングを使用してフィルターに掛けることができます。このユーザー相関データは、CICS XAPADMGR グローバル・ユーザー出口プログラムによって、タスクの関連データの起点記述子のユーザー・データ・セクションに追加されています。

詳しくは、「 *CICSPlex System Manager Operations Views Reference*」を参照してください。

CICS モニターを使用した TCP/IP トラフィックの分析

グループ DFHCICS のパフォーマンス・クラス・モニター・レコードの フィールド 360 - 372 は、TCP/IP に関連しています。「*CICS* パフォーマンス・ガイド」を参照してください。

第 15 章 APPC 接続の管理

マスター端末トランザクション (CEMT) を使用して、APPC 接続を管理することができます。CEMT コマンドのアクションは、接続が CICS にどのように定義されているかによって変わりますが、これについても説明します。

コマンドについては、以下の項で説明します。

- 接続の獲得
- 接続でのセッションの制御とモニター
- 接続の解放

上記のアクションを実行するために使用するコマンドは、次のとおりです。

- CEMT SET CONNECTION ACQUIRED|RELEASED
- CEMT SET MODENAME AVAILABLE|ACQUIRED|CLOSED

CEMT コマンドの形式とオプションについての詳細は、「*CICS Supplied Transactions*」マニュアルを参照してください。

記載されている情報は、主に CICS 領域間の並列セッション接続に関するものです。

APPC リンク管理に関する一般情報

APPC 接続を制御するオペレーター・コマンドが出されると、CICS は、多数の内部プロセスを実行します。この中には、パートナー・システムとの通信も含まれます。

これらのプロセスの主な機能については、以降のページで説明しますが、これらのプロセスがときには相互に無関係であり、非同期の場合もあることに注意してください。このため、これらのプロセスを単純に説明しただけでは、いくつかの点で不正確です。実行は、場合によって、ネットワーク内で発生する個別のイベントや、または APPC 接続の両端での同時オペレーター・アクティビティーにより、さらに修正される可能性があります。このような状況は、ネットワークのコンポーネントに障害が起こって、リカバリーが進行中の場合によく起こります。以降の項では、コマンドの通常の操作について説明します。

注: 以下のセクションで説明する操作規則は、EXEC CICS INQUIRE CONNECTION、INQUIRE MODENAME、SET CONNECTION、SET MODENAME の各コマンドにも当てはまります。これらのコマンドのプログラミング情報については、「*CICS System Programming Reference*」を参照してください。

この章の残りの部分には、次のトピックが含まれています。

- 214 ページの『接続の獲得』
- 216 ページの『SET MODENAME コマンドによるセッションの制御』
- 218 ページの『接続の解放』

© Copyright IBM Corp. 1977, 2009

- 221 ページの『APPC リンク管理の要約』.

接続の獲得

SET CONNECTION ACQUIRED コマンドが出されると、CICS は、パートナー・システムとの接続を確立します。

この操作に関与する主要なプロセスは、次のとおりです。

- モードグループ SNASVCMG における 2 つの LU サービス管理セッションの確立。
- 接続を開始するパートナーによるセッション数変更 (CNOS) プロセスの開始。

  CNOS 折衝は、LU サービス管理セッションのいずれかを使用して、接続に定義された競合勝者と競合敗者の各セッションの数を判別するために実行されます。折衝の結果は、メッセージ DFHZC4900 と DFHZC4901 で報告されます。
- CICS アプリケーション・データを伝送するセッションの確立。

次のプロセスも接続確立の一部ですが、これらについては 317 ページの『第 26 章 相互接続されたシステムにおけるリカバリーと再始動』で説明します。

- ログ名の交換。
- 同期情報の解決と報告。

獲得プロセス中の接続状況

獲得プロセスの前またはその最中の接続状況は、INQUIRE CONNECTION コマンドによって報告されます。

解放済み
SET CONNECTION ACQUIRED コマンドを出す前の初期状態。接続内のすべてのセッションが解放されます。

獲得中 パートナー・システムとの接触が行われ、CNOS 折衝が進行中です。

獲得済み
CNOS 折衝は、すべてのモードグループで完了しました。この状況で、CICS は、モードグループ SNASVCMG 内の LU サービス管理セッションをバインドしました。ユーザー・モードグループ内のセッションのいくつかが、SESSIONS 定義の AUTOCONNECT オプションの結果、またはアプリケーションからの割り振り要求を満たすために、バインドされた可能性もあります。

アプリケーション・プログラムによる接続の使用要求の結果は、セッションの状況によって異なります。セッションの状況は、SESSIONS 定義の AUTOCONNECT オプションによって、以下の項で説明するように制御することができます。

AUTOCONNECT オプションの効果

SESSIONS 定義で AUTOCONNECT オプションを使用すると、接続に関連するモードグループ内でのセッション獲得が制御されます。

APPC 接続で AUTOCONNECT オプションを使用した場合の効果については、 192 ページの『AUTOCONNECT オプション』に説明があります。各モードグループ

は、独自の AUTOCONNECT オプションをもっており、このオプションを設定すると、表 12 で説明するように、モードグループ内のセッションに影響が出ます。

表 12. AUTOCONNECT による SESSIONS 定義への効果

設定値	効果
YES	パートナー・システムとの CNOS 折衝がモードグループについて実行され、接続が獲得されると、折衝されたすべての競合勝者セッションが獲得されます。
NO	パートナー・システムとの CNOS 折衝が実行されますが、セッションは獲得されません。競合勝者セッションは、アプリケーション・プログラムの要求に従って個々にバインドすることができます (例えば、プログラムが ALLOCATE コマンドを出したときなど)。あるいは、SET MODENAME ACQUIRED コマンドを使用して、競合勝者セッションをバインドすることもできます。
ALL	パートナー・システムとの CNOS 折衝がモードグループについて実行され、接続が獲得されると、折衝されたすべてのセッション、競合勝者、および競合敗者が獲得されます。この設定値は、非 CICS システムへの接続においてのみ必要になります。

接続が獲得済み状況にある場合は、INQUIRE MODENAME コマンドを使用して、ユーザー・セッションが必要に応じて使用可能になり、活動化されたかどうかを判別することができます。ユーザー・セッションのバインドはすぐには完了しないため、コマンドを繰り返し使用して、プロセスの最終結果を確認する必要があります。

CICS は、競合勝者セッションをバインドしてアプリケーションの要求に応じることができますが、競合敗者はバインドしません。ただし、競合敗者セッションが既にバインドされている場合、アプリケーションの要求に対してこれらを割り当てることができます。競合敗者のバインドに関する考慮事項については、次の項で説明します。

競合敗者セッションのバインディング

あるシステムの競合敗者セッションは、パートナー・システムの競合勝者セッションであり、パートナーによって上記の説明のようにバインドされます。すべてのセッションをバインドしたい場合は、各側がその競合勝者をバインドするように確認する必要があります。

接続が 2 つの CICS システム間で行われる場合は、AUTOCONNECT(YES) を各システムの SESSIONS 定義に指定するか、または両方のシステムから CEMT SET MODENAME ACQUIRED を出す必要があります。バインド要求を送信できない非 CICS システムにリンクしている場合は、AUTOCONNECT(ALL) を SESSIONS 定義に指定してください。

リモート・システムがバインド要求を送信できる場合は、その競合勝者をバインドする方法を見つけて、SNASVCMG セッションがバインドされたらすぐに、バインドできるようにしてください。

ALLOCATE コマンドは、競合敗者セッションが既にバインドされている場合、それらを会話に割り当てることはできます。しかし、アプリケーションの明示コマンド

としても、自動トランザクション開始 (ATI) によって暗黙指定されたとしても、競合敗者セッションをバインドすることができません。

MAXIMUM オプションの効果

SESSIONS 定義の MAXIMUM オプションでは、モードグループでサポート可能なセッションの最大数、およびコンテンション勝者としてサポートされるセッションの数を指定します。

APPC 接続の操作は、接続の各端でのセッションの最大数が一致し、2 つの端で指定された競合勝者セッションの数の合計がこの最大数になっていれば、より容易になります。このようにすると、CNOS 折衝は、指定された数を変更しません。

接続の各端の指定が一致しないと、説明したとおり、実際の値が LU サービス・マネージャーによって折衝されます。セッションの最大数についての折衝の結果、2 つの値のうち小さい方が採用されます。各パートナーの競合勝者数を判別するために体系化されたアルゴリズムが使用され、折衝の結果は、メッセージ DFHZC4900 と DFHZC4901 で報告されます。

これらの結果は、表 13 に示すように、CEMT INQUIRE MODENAME コマンドを出して知ることもできます。

表 13. INQ MODENAME によって表示されるデータ

表示	解釈
MAXimum	このモードグループのセッション定義に指定された値。これは、この値がパートナー・システムで表示される対応する値以下の場合にのみ、使用可能セッションの実際の値を表します。
AVAilable	使用可能になり、活動化される可能性があるセッションの数についての、最新の CNOS 折衝の結果を表します。 最初の CNOS 折衝に続いて、MAXIMUM オプションの最初の値についての折衝結果を報告します。
ACTive	現在バインドされているセッションの数。

MAXIMUM 値を変更するには、接続を解放してこれを OUTSERVICE に設定し、新しい値で再定義してから、CEDA トランザクションを使用してインストールします。

SET MODENAME コマンドによるセッションの制御

SET MODENAME コマンドを使用すると、接続を解放したり、再獲得することなく、APPC 接続に関連するモードグループ内のセッションを制御することができます。

これを達成するために実行されるプロセスは次のとおりです。

- 行われる変更を定義するための、パートナー・システムとの CNOS 折衝。
- 適切なセッションのバインドまたはアンバインド。

使用可能にするセッションの数をパートナーと折衝するために CICS で使用されるアルゴリズムは複雑で、獲得されるセッションの数が予想と異なる場合があります。この結果は、次のものによって決まります。

- 先行する SET MODENAME コマンドのヒストリー
- パートナー・システム内のアクティビティー
- CICS がセッションへのサービスを停止する原因となったエラー

モードグループは、通常、表 14 で説明する少数の単純なコマンドによって制御することができます。

表 14. SET MODENAME コマンド

コマンド	効果
SET MODENAME ACQUIRED	折衝されたすべての競合勝者セッションを獲得します。
SET MODENAME CLOSED	使用可能なセッションの数をゼロに減らすようにパートナーと折衝し、セッションを解放し、また、パートナーがモードグループ内のセッションを折衝したり、活動化しようとするのを防ぎます。したがって、コマンドを出したシステムだけが、セッション・カウントを増やすことができます。 セッションがアンバインドされる前に、キューイングされたセッション要求が処理されます。
SET MODENAME AVAIL(maximum) ACQUIRED	モードグループがクローズされているときにこのコマンドを出すと、接続が新たに獲得されたものとしてセッションの折衝が行われ、競合勝者セッションが獲得されます。これは、CICS がセッションへのサービスを停止する原因となったエラーにより失われたセッションを再バインドするために使用することもできます。

コマンドの有効範囲と制約事項

ユーザー・モードグループは、CEDA DEFINE SESSIONS (または同等のマクロ) 定義によって作成されますが、これは SET MODENAME コマンドを使用するか、あるいは INQUIRE MODENAME 表示データを上書きすることによって修正することができます。

SNASVCMG モードグループは、CONNECTION 定義から作成されます。この場合は、SET MODENAME コマンドを使用するか、INQUIRE MODENAME 表示データを上書きするかして、その状況を修正しようとする試みは抑制されます。これは、SET CONNECTION コマンドによって、または INQUIRE CONNECTION 表示データを上書きすることによって制御されます。このことは、関連するユーザー・モードグループにも影響します。

CEMT INQUIRE NETNAME では、ネット名がパートナー・システムの applid を示し、接続に関連するすべてのセッションの状況が表示されるため、エラー診断に有効です。上書きによってこれらのセッション状況を変更しようとしても抑制されます。

ユーザー・セッションの状況の管理と、リモート・システムとの折衝の制御には、SET|INQ CONNECTIONvMODENAME を使用しなければなりません。

APPC 接続またはモードグループに対する変更は、オペレーターが CEMT SET コマンドを出すか、またはアプリケーション・プログラムが EXEC CICS SET コマンドを出すことによって要求することができます。これらの SET コマンドの 1 つを出したときに、これとはおそらく矛盾する前の SET コマンドがまだ進行中のこともあり得ます。この状況は特に、多数の並列セッションで構成されたシステムでよく起こります。このようなシステムでは、多数のセッションの状況が、接続またはモードグループへの個々の変更によって影響を受ける場合があります。このように SET コマンドがオーバーラップすると、予測不能な結果が生じるおそれがあります。したがって、必ず以前に出された SET コマンドが完全に終了してから、次の SET コマンドを出すようにしてください。

SET CONNECTION コマンドまたは SET MODEGROUP コマンドを、セッションの自動接続中に出すと、同様の状態が始動時に起こる可能性があります。したがって、ここでも、すべてのセッションの自動接続が終了したことを確認してから、SET コマンドを出すようにしてください。

接続の解放

SET CONNECTION RELEASED コマンドが出されると、CICS は接続を静止し、その接続に関連するすべてのセッションを解放します。

この操作に関与する主要なプロセスは、次のとおりです。

- CNOS プロセスを実行して、パートナー・システムに対し、接続が静止することを通知する。すべてのモードグループで使用可能なセッションの数がゼロになる。
- 接続を使用するトランザクション・アクティビティーを静止させる。このプロセスによって、セッションを使用するトランザクションと待ち状態の ALLOCATE 要求を完了させることができます。セッション割り振りに対する新しい要求は、SYSIDERR 条件によって拒否されます。
- ユーザー・セッションと LU サービス管理セッションのアンバインド。

解放プロセス中の接続状況

次の状態は、解放プロセスの前およびその最中に CEMT INQUIRE CONNECTION コマンドによって報告されます。

獲得済み
セッションが獲得されます。セッションは、トランザクションに割り振ることができます。

解放中 接続の解放が要求され、進行中です。

解放済み
すべてのセッションが解放されます。

接続の両端が制御下にある場合、あるいはパートナーが対立するコマンドを出すことがないように思われる場合は、SET CONNECTION RELEASED を使用して、接続でのアクティビティーを静止させることができます。接続が解放済み状態にある

場合、SET CONNECTION OUTSERVICE を使用すると、パートナーによる接続の再獲得の試みを防止することができます。

限定リソースの効果

APPC 接続で非専用リンク (ダイヤル、ISDN、X.25、X.21、トークンリングなどのリンク) を経由してリモート・システムと通信する場合には、それらのリンクをそのネットワークの限定リソースとして定義することができます。CICS はこの定義を認識し、それらのリンクを必要とするトランザクションがなくなると、ただちにセッションを自動的にアンバインドします。その接続を必要とするトランザクションが新たに呼び出されると、CICS は適切な数のセッションをバインドします。

CEMT INQUIRE CONNECTION コマンドを使用すると、接続状況が次のように表示されます。

獲得済み
: この接続のセッションのいくつかはバインドされており、おそらく使用されています。モードグループ SNASVCMG 内の LU サービス管理セッションは、アンバインドされているかもしれません。

使用可能
: この接続は既に獲得されているが、その使用を必要とするトランザクションが現在ありません。セッションがこのネットワークの限定リソースとして定義されているため、すべてのセッションは既にアンバインドされています。

この接続の動作は、その他の点では、非限定リソースのリンクによる接続とまったく同じです。SET MODENAME コマンドと SET CONNECTION RELEASED コマンドは正常に機能します。

接続を使用不能にする

SET CONNECTION RELEASED コマンドは、この接続を使用するトランザクションを静止させたあと、この接続を解放します。

このコマンドは、それ自体では、パートナー・システムによる接続の再獲得を防止することはできません。パートナーによる接続の再獲得を防止するには、一連のコマンドを実行する必要があります。コマンド順序の選択によって、接続の状況と、いずれかのパートナーからのコマンドに対する応答が決まります。

接続のモードグループすべてで使用可能なセッションの数がゼロになると (例えば、CEMT SET MODENAME AVAILABLE(0) コマンドによって)、ALLOCATE 要求は拒否されます。トランザクション・ルーティングと機能シップ要求も拒否されます。接続は事実上、使用不能になります。ただし、リモート・システムは、セッションの可用性を再折衝して、これらのセッションをバインドさせることができるため、この状態が確実に保持されるものと考えることはできません。

使用不能にしたセッションをパートナーが獲得するのを防止するには、CEMT SET MODENAME CLOSED コマンドを使用してください。これにより、モードグループ内の使用可能なユーザー・セッションの数はゼロになり、モードグループは***ロック***されます。パートナーがここで SET CONNECTION RELEASED に続いて SET CONNECTION ACQUIRED を出しても、ロックされているモードグループのセッションは、AVAILABLE の値をゼロより大きくしない限り、バインドされません。

すべてのモードグループをロックすると、接続は使用不能になります。これは、リモート・システムが、セッションをバインドすることも、この状態を変更するための処置を行うこともできないためです。

接続のモードグループすべてがクローズしたら、CEMT SET CONNECTION RELEASED を出して、さらにステップを進めることができます。これにより、SNASVCMG (LU サービス管理) セッションをアンバインドすることができます。CONNECTION に関する照会を行うと、INSERVICE RELEASED (または、解放プロセスが未完了の場合は INSERVICE FREEING) が返されます。

ここで SET CONNECTION ACQUIRED を入力すると、ロックされたすべてのモードグループが解放されて、接続が完全に確立されます。パートナーがこれと同じコマンドを出した場合は、SNASVCMG セッションだけがバインドされます。

CEMT SET CONNECTION OUTSERVICE を呼び出すと、パートナーによる SNASVCMG セッションのバインドを防ぐことができます。このコマンドは、接続が既に解放状態にある場合以外は無視されます。

要約すると、以下のコマンドを次の順序で出すことによって、接続を使用不能にし、各自の制御下に保持することができます。

```
CEMT SET MODENAME(*) CONNECTION(....) CLOSED

        [CONNECTION オプションは、
        MODENAME が複数の接続に適用される場合に
                のみ有効です。]
INQ MODENAME(*) CONNECTION(....)

 [すべての非 SNAVCMG モードグループの AVAILABLE カウントが
ゼロになるまで、このコマンドを繰り返してください。]
SET CONNECTION(....) RELEASED
       INQ CONNECTION(....)

 [RELEASED 状況が表示されるまで、このコマンドを繰り返してください。]
SET CONNECTION(....) OUTSERVICE
```

図 59. 接続を使用不能にする

使用可能セッションを持たない APPC モードグループからの割り振り

アプリケーション・プログラムは、次の 2 つの方法のいずれかで満たすことができる、APPC セッションに対する ALLOCATE コマンドを出すことができます。

1. 特定のモードグループ内のセッションによってのみ。
2. 接続のすべてのモードグループ内のセッションによって。

オペレーターは、CEMT SET MODENAME AVAILABLE(0) または CEMT SET MODENAME CLOSE を出して、各モードグループの使用可能セッションの数をゼロに減らすことができます。

特定のモードグループが使用可能セッションを持たないときに、そのモードグループに対して ALLOCATE を出すと、このコマンドは SYSIDERR 条件によってすぐに拒否されます。

特定のモードグループを指定しないで ALLOCATE コマンドを出した場合、その接続のどのモードグループにも使用可能なセッションがないと、このコマンドは、SYSIDERR 条件でただちに拒否されます。

関連するモードグループが、割り振り要求を受け取ったときに、まだキューを処理している場合、割り振り要求は対応されて、処理キューに追加されます。使用可能セッションの数をゼロにするオペレーター・コマンドは、ドレーンが完了するまで完了しません。多数のセッションを割り振る非常に使用頻度の高いシステムの場合、このことは、このようなモードグループ・オペレーター・コマンドが完了するまでに長時間を要することを示します。

エラー条件の診断と訂正

前に発生した障害が原因で使用不能になったユーザー・セッションは、*使用可能カウント* を SET MODENAME AVAILABLE(n) コマンドによって復元するか、増やすことによって使用可能に戻すことができます。このコマンドに ACQUIRED オプションを追加すると、アンバインドされた競合勝者セッションがバインドされます。

ユーザー・セッションの活動中に SNASVCMG セッションがアンバインドされても、接続は獲得済みのままです。 SET CONNECTION ACQUIRED コマンドは、すべてのモードグループ内のすべての競合勝者セッションをバインドしますので、SNASVCMG セッションを再確立するには、このコマンドで十分かもしれません。

障害の原因が除去されても、セッションをリカバリーできない場合があります。このような場合は、まず接続を解放してから再獲得する必要があります。

APPC リンク管理の要約

表 15 は、APPC リンクの状況に対する CEMT コマンドの効果を示しています。

表 15. 作動可能 APPC リンクに対する CEMT コマンドの影響

コマンドは以下の順に発行される								
1	1	1						SET MODENAME AVAILABLE(0)
			1	1	1			SET MODENAME CLOSED
	2	2		2	2	1	1	SET CONNECTION RELEASED
		3			3		2	SET CONNECTION OUTSERVICE
結果の状態および反応								
N	N	N	N	N	N	N	N	ALLOCATE 要求中断
Y	Y	N	N	N	N	Y	N	パートナーが折衝可能
Y	Y	Y	Y	Y	Y	Y	Y	SYSIDERR で ALLOCATE がリジェクト
N	Y	Y	N	Y	Y	Y	Y	SNASVCMG セッションが解放

表 15. 作動可能 APPC リンクに対する CEMT コマンドの影響 (続き)

—	Y	N	—	Y	N	Y	N	パートナーは SNASVCMG を再バインド可能

コマンドの有効範囲と制約事項

ユーザー・モードセットは CEDA DEFINE SESSIONS 定義から作成されますが、これは SET MODENAME コマンドを使用するか、あるいは INQUIRE MODENAME 表示データを上書きすることによって修正することができます。

これに対し、SNASVCMG モードセットは CONNECTION 定義から作成され、SET または INQUIRE MODENAME コマンドによってこの状況を修正しようとしても抑制されます。ただし、このモードセットは SET|INQ CONNECTION によって制御されます。そして、このコマンドはユーザー・モードセットにも影響を与えます。

CEMT INQUIRE NETNAME では、ネット名はパートナー・システムの applid を示しますが、そのリンクに関連するすべてのセッションの状況が表示されます。これらのセッション状況を変更しようとしても抑制されます。ユーザー・セッションの状況の管理と、リモート・システムとの折衝の制御には、SET|INQ CONNECTION|MODENAME を使用しなければなりません。INQ NETNAME は、エラー診断にも役立ちます。

第 16 章 リモート・リソースの定義

この章には、リモート・リソースの指定と定義に関するガイダンス情報が記載されています。

章には、以下のトピックが含まれています。

- 『どのリモート・リソースを定義しなければならないか』
- 224 ページの『リソースのローカル名とリモート名』
- 225 ページの『機能シップのためのリモート・リソースの定義』
- 229 ページの『DPL のリモート・リソースの定義』
- 232 ページの『非同期処理のためのリモート・リソースの定義』
- 233 ページの『トランザクション・ルーティングのためのリモート・リソースの定義』
- 251 ページの『DTP のリモート・リソースの定義』.

どのリモート・リソースを定義しなければならないか

リモート・リソースとは、リモート・システムに置かれている一方で、ローカル CICS システムからアクセスする必要のあるリソースを言います。一般に、これらのリソースはすべて、ローカル・リソースを定義するのと同じようにして、各自のローカル CICS システムに定義する必要があります。この定義には、リソースのタイプによって、CICS のオンライン・リソース定義 (RDO) かリソース定義マクロを使います。

CICS 機能シップ、DPL、非同期処理 (START コマンドのシップ)、トランザクション・ルーティングには、リモート・リソース定義が必要になる場合があります。分散トランザクション処理の場合は、リモート・リソース定義は必要ありません。224 ページの『「デイジー・チェーン」に関する注記』を参照してください。

定義できるリモート・リソースは次のとおりです。

- リモート・ファイル (機能シップ)
- リモート DL/I PSB (機能シップ)
- リモート一時データ宛先 (機能シップ)
- リモート一時記憶域キュー (機能シップ)
- 分散プログラム・リンク (DPL) のためのリモート・プログラム
- リモート端末 (トランザクション・ルーティング)
- リモート APPC 接続 (トランザクション・ルーティング)
- リモート・トランザクション (トランザクション・ルーティングと非同期処理)

すべてのリモート・リソースは、当然、それを所有するシステムにも定義されていなければなりません。

© Copyright IBM Corp. 1977, 2009

「デイジー・チェーン」に関する注記

この章で、リモート・リソースの定義方法を述べる場合には、通常、ローカル CICS とリモート・リソースが常駐するシステムの間に直接リンクがあるものとします。

実際には、すべてのタイプの CICS 相互通信において、ローカル・システムとリモート・システムが直接接続されている必要はありません。リモート・リソースに対する要求は、そのリソースを各中間システムと (必要なら) ローカル・システムでリモートとして定義することで、複数の CICS システムにわたって「デイジー・チェーン」にすることができます。

注: 次のタイプの要求をデイジー・チェーンすることはできません。

- 動的ルーティングされる DPL 要求 (112 ページの『DPL 要求の「デイジー・チェーン」』を参照)
- 非端末関連 START コマンドによって開始された、動的ルーティング・トランザクション
- 動的ルーティング・トランザクション。これは、CICS ビジネス・トランザクション・サービス活動に関連付けられています。

リソースのローカル名とリモート名

通常、CICS リソースは名前で参照されます。例えば、ファイルの場合はファイル名、一時記憶域キューの場合はデータ ID で参照されます。リモート・リソースを定義する場合は、リモート・システム上のリソースの名前と、ローカル・システムにおけるその名前の両方を考慮する必要があります。

リモート・リソースの CICS 定義のすべてに REMOTENAME オプション (マクロ・レベル定義の RMTNAME) があり、これにより、リモート・システムでのリソース名を指定することができます。このオプションを省略すると、CICS によって、そのリソースのローカル名とリモート名は同じであるものと見なされます。

ローカルとリモートでのリソースの命名方法は、 225 ページの図 60 に示すとおりです。

```
    CICSA                                    CICSB
 (ローカル・システム)                      (リモート・システム)

DFHSIT TYPE=                                DFHSIT TYPE=
       ,APPLID=CICSA       1
                                          3          ,APPLID=

DEFINE CONNECTION(CICR)    2
       NETNAME(CICSB)      3
                                            DEFINE CONNECTION(CICL)
                                          1         NETNAME(CICSA)

                                          4 DEFINE FILE(FILEA)
DEFINE FILE(FILEA)         4
       REMOTESYSTEM(CICR)  2

DEFINE FILE(FILEB)

                                          5 DEFINE FILE(FILEB)
DEFINE FILE(local-name)
       REMOTESYSTEM(CICR)  2
       REMOTENAME(FILEB)   5
```

図 60. ローカル・リソース名とリモート・リソース名

図 60 は、ローカル・リソース名とリモート・リソース名の関係を示しています。これは、リモート CICS システム (CICSB) によって所有される FILEA と FILEB という 2 つのファイルと、ローカル CICS システム CICSA におけるリモート・リソースとしてのそれらの定義を示しています。

FILEA は両方のシステムで同じ名前をもつため、どちらのシステムで FILEA を参照しても、同じファイルを示すことになります。

FILEB には、ローカル・システムでのローカル名が与えられているため、このファイルは、ローカル・システムではそのローカル名によって参照され、リモート・システムでは FILEB で参照されます。リモート・ファイルの「実」名は、REMOTENAME オプションに指定されます。CICSA は、FILEB と呼ばれるローカル・ファイルを所有することもできることに注意してください。

リモート・リソースに名前を付ける場合、問題が起きないように**注意してください**。例えば、図 60 において、REMOTESYSTEM(CICL) を使用して CICSB に FILEA を定義したとします。この場合、CICS は、使用可能なすべてのセッションが割り振られるまで、FILEA に対する要求があれば繰り返し再シップします。

機能シップのためのリモート・リソースの定義

CICS 機能シップを使用している場合は、リモート・ソースを定義する必要があることがあります。

- リモート・ファイル
- リモート DL/I PSB
- リモート一時データ宛先
- リモート一時記憶域キュー

リモート・ファイルの定義

リモート・ファイルとは、別の CICS システムに常駐するファイルのことをいいます。

リモート・ファイルに対して行われた CICS ファイル制御要求は、CICS 機能シップによってリモート・システムにシップされます。

アプリケーションは、ファイルの位置を知らなくても、それらのファイルにアクセスできるように設計することができます。この機能をサポートするには、リモート・ファイルを (REMOTESYSTEM オプションによって) ローカル・システムに定義する必要があります。

あるいは、CICS アプリケーション・プログラムによって、ファイル制御要求にリモート・システムの名前を SYSID オプションで明示的に指定することができます。これにより、ローカル CICS システムにリモート・ファイルを定義する必要がなくなります。

リモート・ファイルは、RDO を使用して定義されます。以下に示した定義は、CICS に十分な情報を与えて、ファイル制御要求を指定のリモート・システムにシップできるようにしたものです。

```
オンライン・リソース定義
DEFINE
   FILE(name)
   GROUP(.....)
   DESCRIPTION(......)
 リモート属性
   REMOTESYSTEM(name)
   REMOTENAME(name)
   RECORDSIZE(record-size)
   KEYLENGTH(key-length)
```

図 61. リモート・ファイルの定義 (機能シップ)

MRO はユーザー保守のリモート・データ・テーブルと CICS 保守のリモート・データ・テーブルの両方に対してサポートされますが、CICS では、リモート・ソース・データ・セットに基づいてローカル・データ・テーブルを定義することはできません。しかし、この制約を回避する方法がいくつかあります (39 ページの『ファイル制御』を参照)。

リモート・システムの名前

このファイルに対するファイル制御要求のシップ先のリモート・システムの名前を REMOTESYSTEM オプションに指定します。その名前がローカル・システムの名前である場合には、要求はシップされません。

ファイル名

そのファイルがローカル CICS システムで認識されている名前は、FILE オプションに指定します。これは、ローカル・システム内のアプリケーション・プログラムによって、ファイル制御要求で使用される名前です。

そのファイルがリモート CICS システムで認識されている名前は、REMOTENAME オプションに指定します。これは、CICS によってリモート・システムにシップされるファイル制御要求で使用される名前です。

ファイルの名前がローカル・システムとリモート・システムの両方で同じなら、REMOTENAME オプションを指定する必要はありません。

レコード長

リモート・ファイルのレコード長を RECORDSIZE オプションで指定することができます。

ご使用のシステムで C 言語を使用している場合には、固定長レコードをもつすべてのファイルについて、レコード長を指定する必要があります。

これ以外の場合、レコード長は、ファイル制御コマンドの必須オプションであるか、コマンド言語変換プログラムによって推定できるかのどちらかです。

ファイル定義の共用

状況によっては、複数の CICS システムで、共通 CICS システム定義 (CSD) ファイルを共用できる場合があります。

(CSD の共用については、「*CICS System Definition Guide*」の 非 RLS モードでの CSD の共用を参照してください。) ローカル・システムとリモート・システムで CSD を共用する場合には、機能シップで使用する各 VSAM ファイルを 1 度だけ定義する必要があります。

ファイルは、ローカル・ファイルの定義の場合と同じように、DEFINE FILE で完全に定義しなければなりません。さらに、REMOTESYSTEM オプションによって、そのファイル所有領域の sysidnt を指定する必要があります。そのようなファイルがそのファイル所有領域にインストールされると、完全なローカル・ファイル定義が作られます。その他のシステムでは、リモート・ファイル定義が作られます。

リモート DL/I PSB の定義

ローカル CICS システムからリモート DL/I データベースにアクセスするためには、PDIR にリモート PSB を定義する必要があります。

この目的で使用するマクロの形式は、次のとおりです。

```
DFHDLPSB TYPE=ENTRY
         ,PSB=psbname
         ,SYSIDNT=name
         ,MXSSASZ=value
         [,RMTNAME=name]
```

図 62. リモート DL/I PSB を定義するためのマクロ

この項目は、SYSIDNT オプションによって識別されるシステム上の IMS/ESA® DM に知られている PSB を参照します。

PDIR にはリモート項目しか入っていないので、SYSIDNT オペランドと MXSSASZ オペランドが必須となります。

リモート一時データ宛先の定義

リモート一時データ宛先とは、別の CICS システムにある宛先です。リモート宛先に対して行われた CICS 一時データ要求は、CICS 機能シップによってリモート・システムにシップされます。

CICS アプリケーション・プログラムは、SYSID オプションを使用して、一時データ要求に明示的にリモート・システムを指定することができます。これにより、ローカル CICS システムにリモート一時データ宛先を定義する必要がなくなります。

ただし、一般にアプリケーションは、一時データ宛先の位置を意識せずにアクセスできるように設計されています。その場合には、一時データ・キューはリモートの宛先として定義する必要があります。

リモート定義の情報があれば、CICS は、一時データ要求を指定されたリモート・システムにシップすることができます。リモート定義は、図 63 のように作成されます。

```
CEDA を使用する定義
DEFINE
   TDQUEUE(name)
   GROUP(groupname)
   DESCRIPTION(text)
 リモート属性
   REMOTESYSTEM(sysidnt)
   REMOTENAME(name)
   REMOTELENGTH(length)
```

図 63. リモート一時データ・キューのための定義例

リモート一時記憶域キューの定義

リモート一時記憶域キューとは、別の CICS システムにあるキューです。リモート・キューに対して行われた CICS 一時記憶域要求は、CICS 機能シップによって、リモート・システムにシップされます。

CICS アプリケーション・プログラムは、SYSID オプションを使用して、一時記憶域要求に明示的にリモート・システムを指定することができます。これにより、ローカル CICS システムにリモート一時記憶域キューを定義する必要がなくなります。

しかし一般には、アプリケーションは、一時記憶域キューの位置を意識せずにそれにアクセスできるように設計されています。SYSID オプションが一時記憶域要求に指定されていてもいなくても、グローバル・ユーザー出口プログラム XTSEREQ を使用すれば、その要求を適切なキューが定義されているシステムへ送ることができます。この方法を使う場合にも、リモート一時記憶域キューをローカル・システムに定義する必要はありません。XTSEREQ および XTSEREQC の各グローバル・ユーザー出口のプログラミング情報については、「*CICS Customization Guide*」の一時ストレージ EXEC インターフェース・プログラム出口を参照してください。

リモート・システムの名前を一時記憶域要求に明示的に指定せず、XTSEREQ 出口も使用しない場合には、リモート宛先をローカルの一時記憶域テーブルに定義しなければなりません。

一時記憶域テーブル内のリモート項目は、CICS に対して十分な情報を提供し、一時記憶域要求を指定のリモート・システムにシップできるようにするものです。これは、DFHTST TYPE=REMOTE リソース定義マクロによって定義されます。このマクロの形式を図 64 に示します。

```
DFHTST   TYPE=REMOTE
         ,SYSIDNT=name
         ,DATAID=character-string
         [,RMTNAME=character-string]
```

図 64. リモート一時記憶域キューを定義するためのマクロ

DPL のリモート・リソースの定義

CICS DPL を使用している場合は、リモート・サーバー・プログラムを定義しなければならないことがあります。

リモート・サーバー・プログラムは、別の CICS システムにあるプログラムです。リモート・プログラムに対して行われた CICS プログラム制御リンク要求は、CICS DPL によってリモート・システムにシップされます。

リモート・サーバー・プログラムの定義

リモート・サーバー・プログラムは、CEDA トランザクションを使用して定義することができます。

図 65 は、指定する必要のあるプログラムの属性を示しています。属性の指定方法は、プログラムの DPL 要求をリモート領域に*静的* にルーティングするか、*動的* にルーティングするかによって異なります。

```
DEFINE
   PROGRAM(name)
   GROUP(.....)
   DESCRIPTION(......)
 リモート属性
   REMOTESYSTEM(name)
   REMOTENAME(name)
   TRANSID(name)
   DYNAMIC(NO|YES)
```

図 65. リモート・プログラムの定義 (DPL)

リモート・システムの名前

プログラムの DPL 要求を静的にルーティングする場合は、次の作業を実行する必要があります。

- DYNAMIC オプションの値をデフォルトで NO に設定するようにします。
- REMOTESYSTEM オプションで、このプログラムの LINK 要求のシップ先であるサーバー領域の名前を指定します。名前は、インストール済み CONNECTION 定義またはインストール済み IPCONN 定義の名前である必要があります。

プログラムを指定する EXEC CICS LINK コマンドは、REMOTESYSTEM オプションで指定されたサーバー領域にシップされます。

プログラムの DPL 要求を動的にルーティングする場合は、次のようにします。

- DYNAMIC(YES) を指定します。
- REMOTESYSTEM オプションを指定してはなりません。または、REMOTESYSTEM を使用してデフォルトのサーバー領域を指定します。

プログラムを指定する EXEC CICS LINK コマンドにより、動的ルーティング・プログラムが呼び出されます。ルーティング・プログラムは、要求のシップ先のサーバー領域を選択することができます。

プログラム名

そのサーバー・プログラムがローカル CICS システムで認識される名前は、PROGRAM オプションに指定します。これは、ローカル・システム内のクライアント・プログラムによって、リンク要求で使用される名前です。

リモート CICS システムでサーバー・プログラムが認識される名前は、REMOTENAME オプションに指定します。これは、CICS によってリモート・システムにシップされるリンク要求で使用される名前です。

サーバー・プログラムの名前を、ローカル・システムとリモート・システムの両方で同じにする場合は、REMOTENAME オプションを指定する必要はありません。

トランザクション名

プログラム・リソース定義を使用して、DPL サーバーとして使用された場合のプログラムが実行されるときのミラー・トランザクションの名前を指定することができます。TRANSID オプションは、このために使用されます。

CICSPlex SM のルーティング論理はトランザクション・ベースであるため、CICSPlex System Manager (CICSPlex SM) を使用してルーティングを行う動的要求の場合は、TRANSID オプションが特に重要となります。CICSPlex SM は、対応付けられたトランザクションに対して指定されたルールに従って、各 DPL 要求をルーティングします。

注: CICSPlex SM システム・プログラマーは、EYU9WRAM というユーザーが置き換え可能なモジュールを使用して、DPL 要求に関連付けられたトランザクション ID を変更することができます。

CICSPlex SM の入門情報については、「*CICSPlex SM Concepts and Planning*」を参照してください。

リモート・サーバー・プログラムの定義が必要ない場合

リモート・サーバー・プログラムの静的定義をインストールする必要がない場合もあります。

- サーバー・プログラムは自動インストールされます。

  リモート・サーバー・プログラムは、クライアント・システムに静的に定義する代わりに、そのプログラムに対する DPL 要求が最初に出されたときに、自動イ

ンストールすることができます。この方法を使う場合には、自動インストール・ユーザー・プログラムを作成してリモート・システムの名前を指定する必要があります(プログラム用の CICS 自動インストール機能の詳細については、「*CICS Resource Definition Guide*」のプログラム、マップ・セット、および区画セットの自動インストールを参照してください。プログラムの自動インストール・ユーザー・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、「*CICS Customization Guide*」の APPC 接続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・ユーザー・プログラムは、呼び出されると、次のものをインストールすることができます。

サーバー・プログラムのローカル定義
CICS は、ローカル領域でサーバー・プログラムを実行します。

REMOTESYSTEM(remote_region) と DYNAMIC(NO) を指定する定義
CICS は、LINK 要求をリモート領域にシップします。

DYNAMIC(YES) を指定する定義
CICS は、動的ルーティング・プログラムを呼び出して、LINK 要求をルーティングします。

注: DYNAMIC 属性は、REMOTESYSTEM 属性に優先します。したがって、REMOTESYSTEM(remote_region) と DYNAMIC(YES) の両方を指定する定義では、プログラムは特定のリモート領域に常駐するものとしてではなく、動的なものとして定義されます (この場合、REMOTESYSTEM 属性では、動的ルーティング・プログラムに渡すデフォルトのサーバー領域を指定します)。

サーバー・プログラムの定義がない
CICS は、動的ルーティング・プログラムを呼び出して、LINK 要求をルーティングします。

注: ここでは、自動インストール制御プログラムが定義をインストールしないことを*選択している* と見なします。自動インストールが失敗したために定義がインストールされていない場合、動的ルーティング・プログラムは呼び出されません。

- クライアント・プログラムは、EXEC CICS LINK コマンドの SYSID オプションで、ターゲット領域を明示的に指定します。

  注:

  1. LINK コマンドで指定されたプログラムの定義がインストールされていない場合には、動的ルーティング・プログラムは呼び出されますが、SYSID オプションで指定されたリモート領域をシップ先とする要求をルーティングすることはできません。
  2. SYSID オプションでローカル CICS 領域を指定した場合には、動的ルーティング・プログラムは要求をルーティングできます。
- サーバー・プログラムの DPL 呼び出しは動的にルーティングされます。

LINK コマンドで指定されたプログラムの定義がインストールされていない場合には、動的ルーティング・プログラムは呼び出され、(SYSID オプションが指定されていなければ) 要求をルーティングすることができます。

注: サーバー・プログラムのリモート定義が必要ないこともありますが、プログラムの REMOTENAME 属性や TRANSID 属性などを設定する場合には、このリモート定義が必要です。これらの場合には、DYNAMIC(YES) を指定する定義をインストールしてください。

非同期処理のためのリモート・リソースの定義

非同期処理には、START コマンドの TRANSID オプションに指定されたトランザクションに関するリモート・リソース定義が必要です。

しかし、アプリケーションは CICS RETRIEVE コマンドを使用して、リモート一時記憶域キューの名前を入手できることに注意してください。アプリケーションは、あとでその名前を機能シップ要求で指名します。

リモート・トランザクションの定義

CICS 非同期処理のリモート・トランザクションは、別のシステムによって所有されるトランザクションで、ローカル CICS システムからの START コマンドによってのみ呼び出されます。

CICS アプリケーション・プログラムは、SYSID オプションによって、START コマンドに明示的にリモート・システムを指定することができます。これにより、ローカル CICS システムにリモート・トランザクションを定義する必要がなくなります。

ただし、一般にアプリケーションは、トランザクションの位置を意識することなくそれらを開始するように設計されています。この場合、そのトランザクションについてインストールされたトランザクション定義が使用できなければなりません。

注: トランザクションが別の CICS システムによって所有されていて、START コマンドだけでなく、CICS トランザクション・ルーティングでも呼び出される場合は、そのトランザクションをトランザクション・ルーティング用に定義する必要があります。

SYSID オプションを指定しない START コマンドによってのみ呼び出されるリモート・トランザクションは、インストールされたトランザクション定義の基本的な情報だけを必要とします。この場合に使用するリソース定義の形式を 233 ページの図 66 に示します。

```
DEFINE
  TRANSACTION(name)
  GROUP(groupname)
 リモート属性
  REMOTESYSTEM(sysidnt)
  REMOTENAME(name)
  LOCALQ(NO|YES)
```

図 66. リモート・トランザクションの定義 (非同期処理)

ローカル・キュー (LOCALQ) には、START 要求によって開始されたリモート・トランザクションに指定することができます。詳細については、 51 ページの『第 5 章 非同期処理』を参照してください。

REMOTENAME オプションの制約事項

非同期処理要求のいくつかは、トランザクション・ルーティングが関与するプロセスに対して行われます。

この例として、ローカル端末でリモート・トランザクションに接続するための START コマンドがあります。このような要求をサポートするには、REMOTENAME オプションの値とトランザクション名が、開始されるトランザクションのローカル・リソース定義において同じでなければなりません。これらが異なると、要求されたトランザクションは開始されずに、メッセージ DFHCR4310 が、要求側システムの CSMT 一時データ・キューに送信されます。

トランザクション・ルーティングのためのリモート・リソースの定義

CICS トランザクションは、静的、動的のどちらでもリモート領域にルーティングすることができます。

ルーティングするトランザクションは、さまざまな方法で開始することができます。以下に例を挙げます。

- ユーザー端末から。
- 端末関連の ATI 要求 (例えば、端末関連の EXEC CICS START コマンド) から。
- 非端末関連の ATI 要求 (例えば、非端末関連の EXEC CICS START コマンド) によって。
- トランザクションが CICS ビジネス・トランザクション・サービス (BTS) 活動に関連付けられている場合は、BTS RUN ASYNCHRONOUS コマンドによって (BTS については、「*CICS Business Transaction Services*」の CICS ビジネス・トランザクション・サービスについて (What are CICS business transaction services?) に説明があります。)

定義する必要のあるリソースは次のとおりです。

- トランザクションの開始要求が端末に関連付けられている場合は、端末 (234 ページの『トランザクション・ルーティングにおける端末の定義』を参照)。
- トランザクション (すべての場合。 244 ページの『トランザクション・ルーティングのトランザクションの定義』を参照)。

トランザクション・ルーティングにおける端末の定義

端末関連のトランザクション・ルーティングは、ユーザー端末から開始されたトランザクションおよび端末関連の ATI 要求によって開始されたトランザクションのルーティングです。端末がトランザクション・ルーティングの対象となるかどうかを定義するための規則が複数あります。

CICS によってサポートされる端末とセッションのほとんどのタイプは、トランザクション・ルーティングの対象となります。しかし、次の端末は**対象とはならない**ので、リモート・リソースとして定義することができません。

- LUTYPE6.1 の接続およびセッション
- MRO の接続とセッション
- IBM 7770 または 2260 端末
- プールされた 3600 または 3650 パイプライン論理装置
- MVS システム・コンソール

端末とトランザクションを両方の CICS システムで次のように定義する必要があります。

1. 端末専有領域の場合
   a. 端末をローカル・リソースとして定義する必要があります (あるいは自動インストール可能でなければなりません)。
   b. 端末または ATI によって開始されるトランザクションは、リモート・リソースとして定義する必要があります。
2. アプリケーション所有領域の場合
   a. 端末をリモート・リソースとして定義する必要があります (シップされた端末定義が使用可能な場合を除く。 236 ページの『端末定義と接続定義のシップ』を参照)。
   b. トランザクションをローカル・リソースとして定義する必要があります。

トランザクション・ルーティング要求が複数の中間システムにわたって「デイジー・チェーン」される場合も、同じ規則が適用されます。さらに、端末とトランザクションの両方を、中間の CICS システムでリモート・リソースとして定義する必要があります。非 VTAM 端末を使用する場合には、TOR への間接リンクを AOR と中間システムに定義する必要もあります (202 ページの『トランザクション・ルーティングのための間接リンクの定義』を参照)。

トランザクションは、オンライン・リソース定義 (RDO) によって定義されます。

VTAM 端末も RDO によって定義されますが、非 VTAM 端末では、マクロ・レベル定義を使用する必要があります。

リモート VTAM 端末の定義

オンライン・リソース定義 (RDO) を使用してリモート VTAM 端末を定義します。

アプリケーション所有領域で端末を定義する代わりに、必要なときに適切な定義が端末専有領域から**シップされる**ようにすることができます。シップ定義の詳細については、 236 ページの『端末定義と接続定義のシップ』を参照してください。

リモート VTAM 端末は、DEFINE TERMINAL コマンドを使って定義します。このコマンドには次のオプションがあります。

- REMOTESYSNET オプションには、TOR のネット名 (applid) を指定します。CICS は、TOR への直接リンクがない場合でも、この情報を使ってリモート端末の完全修飾 ID を作成することができます (242 ページの『端末のローカル名とリモート名』を参照してください。)
- REMOTESYSTEM オプションには、TOR へのパスにおける次のリンクの名前を指定します。TOR へのパスが複数ある場合には、REMOTESYSTEM を使って、優先パスにおける次のリンクを指定します。

  REMOTESYSTEM に TOR への直接リンクを指定する場合には、通常、REMOTESYSNET を指定する必要はありません。しかし、その直接リンクが、VTAM 総称リソース・グループのメンバーである TOR への APPC 接続の場合には、REMOTESYSNET を指定しなければならない場合があります。つまり、CONNECTION 定義に指定した NETNAME がその TOR の (applid ではなく) 総称リソース名なら、REMOTESYSNET が必要です。

各種端末特性のごく一部については、リモート端末定義に指定する必要があります。これらの特性は次のとおりです。

```
DEFINE
 TERMINAL(trmidnt)
 GROUP(groupname)
端末 ID
 TYPETERM(terminal-type)
 NETNAME(netname_of_terminal)
 REMOTESYSTEM(sysidnt_of_next_system)
 REMOTESYSNET(netname_of_TOR)
 REMOTENAME(trmidnt_on_TOR)
```

図 67. リモート VTAM 端末の定義 (トランザクション・ルーティング)

リモート端末定義によって参照される TYPETERM は、特定の端末タイプの CICS 提供バージョンにすることも、DEFINE TYPETERM コマンドによって定義されたものにすることもできます。リモート端末に**のみ**使用される TYPETERM を定義する場合は、**セッション特性**、**ページング特性**、および**操作可能特性**を無視することができます。また、**アプリケーション機能**の BUILDCHAIN を無視することもできます。

リモート APPC 接続の定義

リモート単一セッション APPC 端末は、リモート VTAM 端末を定義したときと同様に、DEFINE TERMINAL および DEFINE TYPETERM の各コマンドを使用して定義することができます。

リモート VTAM 端末の定義方法については、234 ページの『リモート VTAM 端末の定義』を参照してください。リモートの並列セッション APPC システムとデバイスの場合は、リモート接続を 236 ページの図 68 に示すように定義しなければなりません。SESSIONS 定義は、リモート接続には必要ありません。

```
DEFINE
 CONNECTION(sysidnt_of_device)
 GROUP(groupname)
接続 ID
 NETNAME(netname_of_device)
リモート属性
 REMOTESYSTEM(sysidnt_of_next_system)
 REMOTESYSNET(netname_of_TOR)
 REMOTENAME(sysidnt_of_device_on_TOR)
接続特性
 ACCESSMETHOD(VTAM)
 PROTOCOL(APPC)
```

図 68. リモート APPC 接続の定義 (トランザクション・ルーティング)

端末定義と接続定義の共用方法

状況によっては、複数の CICS システムで、共通 CICS システム定義 (CSD) ファイルを共用できる場合があります。ローカル・システムとリモート・システムで CSD を共用する場合には、それぞれの端末と APPC 接続を 1 度だけ定義する必要があります。

DEFINE TERMINAL コマンドを使用して端末を定義し、ローカル端末定義と同様の関連 TYPETERM 定義を含めます。この端末が端末専有領域にインストールされる際に、完全なローカル端末定義が作成されるように、他のオプションを指定する必要があります。他のシステムには、リモート端末定義が作成されます。

- REMOTESYSNET オプションで、端末専有領域の NETNAME を指定します。
- REMOTESYSTEM オプションで、端末専有領域の SYSIDNT を指定します。

同様に、APPC 接続は、DEFINE CONNECTION コマンドを使用して完全に定義する必要があり、1 つ以上の対応する SESSIONS 定義が必要です。端末定義の場合と同様に、REMOTESYSNET オプションで端末専有領域の NETNAME を指定し、REMOTESYSTEM オプションで端末専有領域の SYSIDNT を指定します。接続が、端末専有領域にインストールされると、接続定義が作成されます。他のシステムには、リモート接続定義が作成され、SESSIONS 定義は無視されます。

トランザクション・ルーティング・パスにあるシステムの間で共通端末または接続定義を共用する場合、それらのシステム間に定義するリンクには、同じ名前を指定する必要があります。つまり、それぞれの CONNECTION 定義には、共通 TERMINAL 定義の REMOTESYSTEM オプションに指定する名前を指定する必要があります。

端末定義と接続定義のシップ

端末専有領域の VTAM 端末を使用している場合、その端末専有領域からアプリケーション所有領域に、端末定義が必要に応じてシップされるようにすることができます。この方式を使用すると、アプリケーション所有領域で端末を定義する必要がなくなります。

リモート・トランザクションがシップ可能端末から呼び出されると、アプリケーション所有領域に伝送される要求には、シップ可能端末定義が使用可能であることを示すフラグが付けられます。アプリケーション所有領域が既にその端末の有効な定

義を持っている (おそらく以前にシップされていた) 場合は、このフラグは無視されます。持っていない場合は、定義のシップを要求します。

シップされた端末定義は、接続されている CICS システムに、その接続を提供している通信セッションを使用して伝搬されます。端末定義が別の領域にシップされるときには、その基本機能が APPC 並列セッションの場合を除いて、TCTUA もシップされます。ルーティングされたトランザクションが終了すると、TCTTE と TCTUA の情報がその端末を所有する領域に戻されます。

注: APPC 接続定義と APPC 端末定義は常にシップ可能です。特殊リソース定義は必要ありません。

端末定義は、複数の中間システムを通してシップすることができます。シップ可能端末を使用していて、AOR から TOR へのパスが複数ある場合には、その TOR への間接リンクをその AOR と中間システムに定義することによって、優先パスを指定することができます (202 ページの『トランザクション・ルーティングのための間接リンクの定義』を参照)。

シップされた定義が中間領域かアプリケーション所有領域にインストールされるときは、その領域で自動インストール・ユーザー・プログラムが呼び出されます。シップされた定義の名前が、その領域に既にインストールされているリモート端末かリモート接続の名前と衝突する場合には、シップされた定義に CICS が*別名* を割り当て、それを自動インストール・ユーザー・プログラムに渡します (端末の別名は、243 ページの『端末の別名』 に記載されています。) シップされた端末や接続に対して CICS が生成した別名は、その最初の文字によって分かります。これは、常に '{' です。残りの 3 文字は、'AAA' から '999' までの値です。自動インストール・ユーザー・プログラムは、CICS 生成の別名を受け入れることも、指定変更することも、拒否することもできます。さらに、このプログラムは、シップされた定義とインストール済みのリモート定義が衝突しない 場合でも、別名を指定することができます。

例えば、同じような端末 ID 群を使って同じ AOR へトランザクション・ルーティングする端末専有領域が複数ある場合には、シップされた定義に別名を割り当てることが必要になります。シップされた端末のインストールを制御する自動インストール・ユーザー・プログラムの作成については、「*CICS Customization Guide*」のシップされた端末の自動インストールを制御するプログラムの作成 (Writing a program to control autoinstall of shipped terminals)を参照してください。

ATI 要求での端末シップ:

ATI によって開始されたトランザクションでリモート端末を獲得する必要がある場合には、通常、その端末を AOR と中間システムに静的に定義します。

このようにする理由は、例えば、区画内一時データ・キュー (259 ページの『区画内一時データ・キューの定義』を参照) のためのリモート端末を指定しても、端末定義がそのリモート・システムからシップされない からです。しかし、前のトランザクション・ルーティング要求によって、シップされた端末定義が既に受信されている場合には、その端末を ATI 要求で使用することが可能です。

しかし、その TOR と AOR が直接に接続されている場合には、CICS が、端末定義を AOR にシップして、ATI 要求を満たせるようにします。その AOR でユーザ

一出口 XALTENF を使用可能にした場合、CICS は、「端末未認識」条件が満たされるたびに、この出口を呼び出します。ユーザー作成のそのプログラムは、ATI 要求の発生元と特性を詳しく示すパラメーターにアクセスします。これらのパラメーターを使えば、CICS にシップしてもらいたい端末定義を所有する領域の ID を知ることができます。EXEC CICS START による開始要求に対しては、同様のユーザー出口 XICTENF があります。

XALTENF と XICTENF を使って端末定義をシップできるのは、TOR と AOR の間に直接リンクがある場合だけであることに注意してください。詳細については、80 ページの『自動トランザクション開始用端末のシップ』を参照してください。

START 要求を端末専有領域からアプリケーション所有領域へ機能シップする場合には、FSSTAFF (機能シップされた START 類縁性) システム初期設定パラメーターを使用しなければならない場合があります。詳細については、85 ページの『複数 TOR の ATI 用端末のシップ』を参照してください。

端末関連の START 要求を処理する方法としては、87 ページの『START コマンドで呼び出されたトランザクションのルーティング』に説明した拡張ルーティング方法の使用をお勧めします。START 要求は、TOR で発行された場合に AOR に機能シップされません。したがって、「認識されない端末」が発生することはなく、また、FSSTAFF を使用して、「間違った」端末でトランザクションが開始されるのを防ぐ必要もありません。代わりに、START は TOR で直接実行され、トランザクションは端末から開始されたかのようにルーティングされます。シップ可能端末を使用した場合は、必要に応じて端末定義が AOR にシップされます。

端末をシップ可能として定義:

端末定義をシップ可能にするには、それを、SHIPPABLE(YES) を指定する TYPETERM に関連付ける必要があります。

```
DEFINE
  TERMINAL(trmidnt)
  GROUP(groupname)
  AUTINSTMODEL(YES|NO|ONLY)
  AUTINSTNAME(name)
  TYPETERM(TRTERM1)
  .
  .
DEFINE
  TYPETERM(TRTERM1)
  .
  .
  SHIPPABLE(YES)
```

図 69. シップ可能端末の定義 (トランザクション・ルーティング)

この方式は、どの VTAM 端末でも使用できますし、TOR で自動インストールを使用する場合には、特に適しています。

アプリケーション所有領域にシップされた端末定義は最後には冗長になるので、それらを AOR (および、その TOR と AOR の間の中間システム) から削除する必要があります。これについては、311 ページの『第 25 章 シップされた端末定義の効率的な削除』を参照してください。

リモート非 VTAM 端末の定義

非 VTAM 端末は、リソース定義マクロを使用して定義する必要があります。RDO は使用できません。

注: CICS Transaction Server for z/OS バージョン 4 リリース 1 では、通信アクセス方式 (TCAM) は直接的にはサポートされません。ただし、TCAM は間接的にサポートされます。つまり、TCAM/DCB (TCAM/ACB ではない) によって、CICS TS for z/OS より前の TOR に接続されている端末は、トランザクション・ルーティングを使用して CICS Transaction Server for z/OS バージョン 4 リリース 1 AOR にアクセスすることができます。したがって、リモート (ローカルでない) TCAM 端末を CICS Transaction Server for z/OS バージョン 4 リリース 1 システムに定義することができます。

CICS Transaction Server for z/OS バージョン 4 リリース 1は、基本通信アクセス方式 (BTAM) をまったくサポートしません。間接的にもサポートしません。したがって、リモート BTAM 端末を定義することはできません。

リモート非 VTAM 端末では、リモート・システム (TOR) に完全な端末管理テーブル項目が必要になり、ローカル・システム (AOR) に、CICS がトランザクション・ルーティングを実行するための端末に関する十分な情報を含む端末管理テーブル項目が必要になります。データ・セット制御情報と回線情報は、リモート端末の定義には必要ありません。

非 VTAM 端末定義はシップ可能ではありません。

リソース定義マクロを使用することにより、リモート非 VTAM 端末を次の 2 つの方法のいずれかで定義することができます。

1. DFHTCT TYPE=REMOTE マクロによって
2. DFHTCT TYPE=REGION マクロに続けて、通常の DFHTCT TYPE=TERMINAL マクロによって

いずれの方式でも、同じ端末定義を使用して、ローカル・システムとリモート・システムの両方に必要な項目を生成することができます。

DFHTCT TYPE=REMOTE を使用する定義:

DFHTCT TYPE=REMOTE マクロの形式を参照しやすいように以下に複製します。

```
DFHTCT   TYPE=REMOTE
    ,ACCMETH=access-method
    ,SYSIDNT=name-of-CONNECTION-to-TOR
    ,TRMIDNT=name
    ,TRMTYPE=terminal-type
    [,ALTPGE=(lines,columns)]
    [,ALTSCRN=(lines,columns)]
    [,ALTSFX=number]
    [,DEFSCRN=(lines,columns)]
    [,ERRATT={NO|([LASTLINE][,INTENSIFY]
    [,{BLUE|RED|PINK|GREEN|TURQUOISE|YELLOW
       |NEUTRAL}]
    [,{BLINK|REVERSE|UNDERLINE}])}]
    [,FEATURE=(feature[,feature],...)]
    [,LPLEN={132|value}]
    [,PGESIZE=(lines,columns)]
    [,RMTNAME={name-specified-in-TRMIDNT|name}]
    [,STN2980=number]
    [,TAB2980={1|value}]
    [,TCTUAL=number]
    [,TIOAL={value|(value1,value2)}]
    [,TRMMODL=numbercharacter]
    TCAM SNA のみ
    [,BMSFEAT=([FMHPARM][,NOROUTE][,NOROUTEALL]
        [,OBFMT][,OBOPID])]
    [,HF={NO|YES}]
    [,LDC={listname|(aa[=nnn],bb[=nnn],cc[=nnn],...)
    [,SESTYPE=session-type]
    [,VF={NO|YES}]
```

図 70. リモート非 VTAM 端末の定義 (トランザクション・ルーティング)

SYSIDNT には、端末専有領域への接続の名前を指定します。TOR への直接リンクがない場合には、SYSIDNT に**間接リンク**の名前を指定しなければなりません (202 ページの『トランザクション・ルーティングのための間接リンクの定義』を参照)。

端末定義の共用:

このセクションは、TCAM 以外の、サポートされるすべてのタイプの非 VTAM 端末に適用されます。CICS Transaction Server for z/OS バージョン 4 リリース 1 は、ローカルの TCAM 端末をサポートしないため、ローカル定義も作成されません。

TCAM 端末:

SYSIDNT を除き、DFHTCT TYPE=REMOTE のオペランドは、DFHTCT TYPE=TERMINAL で指定可能なオペランドのサブセットです。残りのオペランドはどれでも指定できますが、SYSIDNT オペランドがローカル・システムを指定していない限り、それらは無視されます。ローカル・システムを指定している場合は、このマクロは DFHTCT TYPE=TERMINAL 形式と同等です。

したがって、単一の DFHTCT TYPE=REMOTE マクロを使って、同じ端末をローカル・システムとリモート・システムの両方に定義することができます。この定義方式の典型的な使用法を 241 ページの図 71 に示します。

```
  Local System CICL                  Remote System CICR
        AOR                                TOR

DFHSIT TYPE=                       DFHSIT TYPE=
       SYSIDNT=CICL                       SYSIDNT=CICR

DFHTCT TYPE=INITIAL,               DFHTCT TYPE=INITIAL,
       ACCMETH=NONVTAM,                   ACCMETH=NONVTAM,
       SYSIDNT=CICL,                      SYSIDNT=CICR,
       .                                  .
       .                                  .

DFHTCT TYPE=REMOTE,                DFHTCT TYPE=REMOTE,
       SYSIDNT=CICR                       SYSIDNT=CICR
       TRMIDNT=aaaa,                      TRMIDNT=aaaa,
       TRMTYPE=3277,                      TRMTYPE=3277,
       TRMMODL=2,                         TRMMODL=2,
       ALTSCRN=(43,80)                    ALTSCRN=(43,80)
       .                                   .
       .                                   .
DFHTCT TYPE=FINAL                  DFHTCT TYPE=FINAL
```

図 71. DFHTCT TYPE=REMOTE マクロの典型的な使用

図 71 では、同じ端末定義がローカル・システムとリモート・システムの両方で使用されます。

ローカル・システムでは、端末 sysidnt がローカル・システムの sysidnt (DFHTCT TYPE=INITIAL マクロに指定されている) と異なるため、リモート端末項目が作成されます。リモート・システムでは、端末 sysidnt がリモート・システムそのものと同じであるため、TYPE=REMOTE マクロは、TYPE=TERMINAL マクロとまったく同様に扱われます。

注: TCAM 端末では、ローカル端末定義は作成されません。

注: この方式が正しく機能するためには、ローカル・システムからリモート・システムへの CONNECTION に、リモート・システム自体の sysidnt 名 (この例では、CICR) を指定しなければなりません。

端末識別は、いずれのシステムでも ″aaaa″ です。

DFHTCT TYPE=REGION を使用する定義:

DFHTCT TYPE=REGION マクロを使用すると、DFHTCT TYPE=SDSCI、TYPE=LINE、および TYPE=TERMINAL の各マクロを使用して、ローカル端末と同じ方法でリモート端末を定義することができます。

ただし、定義の前には、次の形式の DFHTCT TYPE=REGION マクロを付ける必要があります。

```
DFHTCT   TYPE=REGION
         ,SYSIDNT={name-of-CONNECTION-to-TOR|LOCAL}
```

SYSIDNT には、端末専有領域への接続の名前を指定します。TOR への直接リンクがない場合には、SYSIDNT に**間接リンク**の名前を指定しなければなりません (202 ページの『トランザクション・ルーティングのための間接リンクの定義』を参照)。

端末定義の共用:

SYSIDNT がローカル・システムを指定しない場合は、リモート端末項目を作成するための情報だけがそのあとの定義から抽出されます。DFHTCT TYPE=SDSCI と TYPE=LINE の定義は無視されます。 TYPE=REMOTE サブセットの一部ではない TYPE=TERMINAL 定義のパラメーターも無視されます。

ローカル・システム定義へ戻るには、DFHTCT TYPE=REGION,SYSIDNT=LOCAL を使用します。

この定義方式の典型的な使用法を図 72 に示します。

```
端末専有領域          アプリケーション所有領域

  DFHTCT TYPE=INITIAL,          DFHTCT TYPE=INITIAL,
         SYSIDNT=TERM,                 SYSIDNT=TRAN,
         ACCMETH=NONVTAM               ACCMETH=NONVTAM
         .                             .

                                DFHTCT TYPE=REGION,
                                       SYSIDNT=TERM

  COPY TERMDEFS                 COPY TERMDEFS

                                DFHTCT TYPE=REGION,
                                       SYSIDNT=LOCAL

  DFHTCT TYPE=FINAL             DFHTCT TYPE=FINAL
```

図 72. DFHTCT TYPE=REGION マクロの典型的な使用法

図 72 では、端末定義の同じサンプル集が、端末専有領域とアプリケーション所有領域の両方で使用されています。

端末専有領域の場合、ローカル端末項目が作成されます。

注: TCAM 端末では、ローカル端末定義は作成されません。

アプリケーション所有領域では、TYPE=REGION マクロに指定された sysidnt が DFHTCT TYPE=INITIAL マクロに指定された sysidnt と異なるため、リモート端末項目が作成されます。

端末のローカル名とリモート名

CICS は、トランザクション・ルーティングに関与するすべての端末に固有の ID を使用します。この ID は、その端末を所有する CICS システムの applid (ネット名) と、端末専有領域の端末定義に指定された端末 ID から構成されます。

例えば、CICS システムの applid が PRODSYS で、端末 ID が L77A であれば、完全修飾の端末 ID は PRODSYS.L77A になります。

次の規則は、ハードコーディングされたリモート端末のあらゆる形式の定義に適用されます。

- この定義によって、CICS が端末専有領域のネット名にアクセスできなければならない。例えば、VTAM 端末を使用するが、TOR への直接リンクがない場合には、REMOTESYSNET オプションを使って、その TOR のネット名を指定する必要があります。

非 VTAM 端末を使用するが、TOR への直接リンクがない場合には、DFHTCT TYPE=REMOTE か TYPE=REGION マクロの SYSIDNT オペランドに**間接リンク** (ここの NETNAME オプションにその TOR の applid を指定する) の名前を指定する必要があります。

- 「実」端末 ID は、直接または別名によって、常に指定しなければならない。

TOR のネット名の指定:

リモート端末の定義は、TOR のネット名を CICS が必ずアクセスできるようになっていなければなりません。

以下の例では、端末専有領域の applid を PRODSYS とします。

```
VTAM 端末定義
DEFINE TERMINAL             DEFINE CONNECTION(PD1)     TOR への
REMOTESYSTEM(PD1)           NETNAME(PRODSYS)           直接リンク
.                           .
.                           .
VTAM 端末定義
DEFINE TERMINAL             DEFINE CONNECTION(NEXT)    TOR への
REMOTESYSTEM(NEXT)          NETNAME(INTER1)            非直接リンク
REMOTESYSNET(PRODSYS)
.                           .
.                           .
非 VTAM 端末定義 (メソッド 1)
DFHTCT TYPE=REMOTE,         DEFINE CONNECTION(PD1)     TOR への
SYSIDNT=PD1,                NETNAME(PRODSYS)           直接リンク
.                           .
.                           .
非 VTAM 端末定義 (メソッド 2)
DFHTCT TYPE=REGION,         DEFINE CONNECTION(PD1)     TOR への
SYSIDNT=PD1                 NETNAME(PRODSYS)           直接リンク
.                           .
.                           .
非 VTAM 端末定義 (メソッド 1)
DFHTCT TYPE=REMOTE,         DEFINE CONNECTION(REMT)    TOR への
SYSIDNT=REMT,               NETNAME(PRODSYS)           非直接リンク
                            ACCESSMETHOD(INDIRECT)
                            INDSYS(NEXT)
DFHTCT TYPE=TERMINAL,
.
```

図 73. 端末専有領域の識別

端末の別名:

端末がアプリケーション所有領域で認識されている名前は、通常、端末専有領域のその名前と同じです。ただし、アプリケーション所有領域では、異なる名前 (別名) によってリモート端末を呼び出すことができます。

端末専有領域とアプリケーション所有領域が、それぞれ同じ名前の端末を所有する場合は、別名を指定する必要があります。同じ名前のローカル端末定義とリモート端末定義をもつことはできません (別々のリモート領域にある端末に対し、同じ名前のリモート端末定義をもつこともできません)。

別名を使用する場合には、次のように、端末の「実」名をそのリモート名としてさらに指定する必要があります。

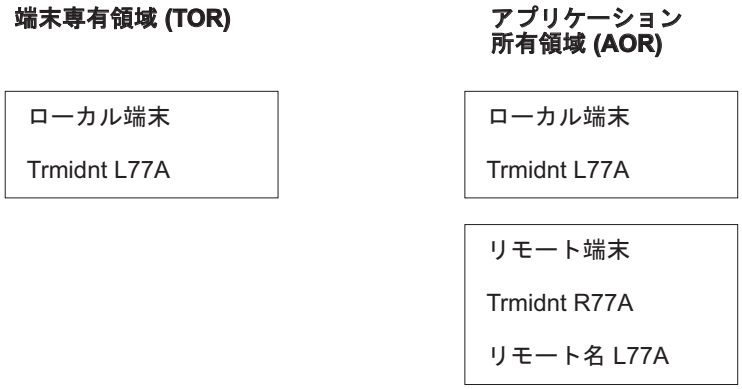


図 74. リモート端末のローカル名とリモート名

リモート名は DEFINE TERMINAL の REMOTENAME オプション、または DFHTCT TYPE=REMOTE の RMTNAME オペランドに指定します。

トランザクション・ルーティングのトランザクションの定義

この項では、トランザクション・ルーティングによって呼び出される可能性のあるトランザクションの定義について説明します。これは、すべての形式のトランザクション・ルーティングに適用されます。

トランザクションに対する CEDA DEFINE コマンドの一般的な形式を 245 ページの図 75 に示します。

```
  DEFINE
    TRANSACTION(name)
    GROUP(groupname)
    PROGRAM(name)
    TWASIZE(0|value)
    PROFILE(DFHCICST|name)
    PARTITIONSET(name)
    STATUS(ENABLED|DISABLED)
    PRIMEDSIZE(00000|value)
    TASKDATALOC(BELOW|ANY)
    TASKDATAKEY(USER|CICS)
    STORAGECLEAR(NO|YES)
    RUNAWAY(SYSTEM|value)
    SHUTDOWN(DISABLED|ENABLED)
    ISOLATE(YES|NO)
リモート属性
    DYNAMIC(NO|YES)
    REMOTESYSTEM(name)
    REMOTENAME(local-name|remote-name)
    TRPROF(DFHCICSS|name)
    LOCALQ(NO|YES)
    ROUTABLE(NO|YES)
スケジューリング
    PRIORITY(1|value)
    TCLASS(NO|value)
    TRANCLASS(DFHTLC00|name)
別名
    ALIAS(name)
    TASKREQ(value)
    XTRANID(value)
    TPNAME(name)
    XTPNAME(name)
リカバリー
    DTIMOUT(NO|value)
    INDOUBT(BACKOUT|COMMIT|WAIT)
    RESTART(NO|YES)
    SPURGE(NO|YES)
    TPURGE(NO|YES)
    DUMP(YES|NO)
    TRACE(YES|NO)
セキュリティー
    RESSEC(NO|YES)
    CMDSEC(NO|YES)
    EXTSEC(NO|YES)
    TRANSEC(01|value)
    RSL(00|value|Public)
```

図 75. CEDA DEFINE TRANSACTION のオプション

トランザクションがローカル実行用またはリモート実行用に選択される方法は、トランザクション定義に指定されたリモート属性 によって決まります。[5] この方法には、次の 3 つが考えられます。

1. リモート属性が DYNAMIC(NO) を指定しているときに、REMOTESYSTEM 名がブランクであるか、あるいはローカル・システムの sysid である。

   この場合、トランザクションはローカルで実行され、トランザクション・ルーティングは呼び出されません。

5. ここでは、SYSID オプションを使用して、トランザクションの実行環境であるリモート領域を指定する EXEC CICS START コマンドの特殊ケースは無視します。START コマンドで明示的に指定されたリモート領域が、トランザクション定義で指定されたリモート領域に優先します。

2. リモート属性は DYNAMIC(NO) を指定し、REMOTESYSTEM 名がローカル・システムの sysid と異なる。

   この場合、トランザクションは REMOTESYSTEM オプションで指定されたシステムにルーティングされます。これは、**静的**トランザクション・ルーティングといいます。[6]

3. リモート属性が DYNAMIC(YES) を指定する。

   この場合、トランザクションをどこで実行するかは、動的ルーティング・プログラムまたは分散ルーティング・プログラムで決定されます。70 ページの『2 つのルーティング・プログラム』を参照してください。

   注: このルールの例外は、拡張ルーティングすることができる EXEC CICS START コマンドで開始されるトランザクションです。例えば、これらのトランザクションのうちの 1 つが DYNAMIC(YES) と定義されている場合、動的ルーティング・プログラムは呼び出されますが、トランザクションをルーティングすることはできません。87 ページの『START コマンドで呼び出されたトランザクションのルーティング』を参照してください。

TRANSACTION オプションで指定する名前は、ローカル領域でトランザクションを呼び出すときのトランザクションの名前です。プログラム・アテンション (PA) キー、プログラム機能 (PF) キー、ライト・ペン、磁気スロット読取装置、オペレーター ID カード読取機構などの特殊な入力を使用する場合には、TASKREQ を指定できます。

トランザクションがローカルで実行される可能性があるなら、この定義は、ローカル・トランザクションを定義するときの通常の規則に従わなければなりません。特に、PROGRAM オプションには、ローカル・システムにインストールするユーザー・プログラムを指定する必要があります。トランザクションが別のシステムにルーティングされる場合には、それに関連するプログラムは、PROGRAM オプションに指定する名前が何であれ、常に中継プログラム DFHAPRT です。

PROFILE オプションは、端末と中継トランザクション (またはトランザクションがローカルに実行される場合はユーザー・トランザクション) の間の通信に使用されるプロファイルを指定します。リモート実行の場合、TRPROF オプションでは、中継トランザクションとリモート・トランザクション所有システム間のセッションでの通信に使用されるプロファイルを指定します。プロファイルについては、253 ページの『通信プロファイルの定義』を参照してください。

トランザクションが常にリモート・システムにルーティングされる場合、ローカル・システムで実行されるトランザクションは常に中継トランザクションであるため、中継トランザクションの制御にいくつかのオプションを指定することができます。

- 中継トランザクションは TWA を必要としないため、TWASIZE をゼロに設定、またはデフォルト解釈することができます。

---

6. REMOTESYSTEM オプションで、別のシステムへの**直接**リンクを指定します (間接リンクでもリモート APPC 接続でもありません)。

- オペレーターが開始したルーティングされるトランザクションには、トランザクション・セキュリティーを指定する必要があります。中継トランザクションはリソースにアクセスしないため、リソース保護検査を指定する必要はありません。セキュリティーについては、「*CICS RACF Security Guide*」の トランザクションのセキュリティー (Transaction security)を参照してください。
- マップ式 APPC 接続または MRO セッションでトランザクション・ルーティングを行う場合は、トランザクション定義の TRPROF オプションに指定する通信プロファイルに、RTIMOUT オプションを指定する必要があります。これによって、トランザクションがルーティングされた先のシステムから適当な時間内に応答がなければ、その中継トランザクションはタイムアウトになります。

  (トランザクション定義の DTIMOUT オプションに指定する) デッドロック・タイムアウトは、端末入出力待ちでは起こりません。中継トランザクションは、セッションを取得した後はリソースにアクセスしないので、中断状態にある割り振り要求をトラップする以外には DTIMOUT の必要はほとんどありません。(リモート・システムに対する空きセッションがないときに、ALLOCATE 要求をキューイングするか拒否するかを指定する方法については、307 ページの『第 24 章 システム間のセッション・キューの管理』で説明します。)

ルーティングのためのトランザクションを定義する方法は、そのトランザクションのルーティングが静的か動的かによって異なります。

静的トランザクション・ルーティング

静的にルーティングするトランザクションの定義には、2 つの方法があります。

別々のローカル定義とリモート定義を使用する:

トランザクションのリモート定義を作成し、要求側領域にインストールします。REMOTESYSTEM オプションには、ターゲット領域の名前 (または、その要求が「デイジー・チェーンされる」のであれば、中間システムの名前) を指定する必要があります。

次に、そのトランザクションの別のリモート定義を中間システムにインストールします。REMOTESYSTEM オプションには、ルーティング・チェーンにおける次のシステムの名前を指定しなければなりません。次に、そのトランザクションのローカル定義を作成し、ターゲット領域にインストールします。REMOTESYSTEM オプションはブランクか、領域の名前でなければなりません。

トランザクションが EXEC CICS START コマンドで開始される場合には、拡張ルーティング方式 (87 ページの『START コマンドで呼び出されたトランザクションのルーティング』を参照) を使用できるかどうか確認します。拡張ルーティングが可能な場合には、START を発行する領域で、トランザクションを ROUTABLE(YES) として定義します。

トランザクション・ルーティングのパスにある複数のシステムが同じ CSD を共用する場合には、それらのトランザクション定義は別々のグループになければなりません。

二重目的定義の使用:

単一のトランザクション定義を作成し、要求側領域とターゲット領域間で (さらに、「デイジー・チェーン」が関係する場合には、複数の中間システム間で) 共用します。REMOTESYSTEM オプションには、ターゲット領域の名前を指定します。

トランザクションが EXEC CICS START コマンドで開始される場合には、拡張ルーティング方式 (87 ページの『START コマンドで呼び出されたトランザクションのルーティング』を参照) を使用できるかどうか確認します。拡張ルーティングが可能な場合は、単一の定義を ROUTABLE(YES) として指定します。

定義が各システムにインストールされるとき、ローカル CICS は、その SYSIDNT と REMOTESYSTEM 名を比べます。それらが異なっていると (要求側領域の場合のように)、リモート・トランザクション定義が作成されます。それらが同じだと (ターゲット領域の場合のように)、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。

静的トランザクション・ルーティングの場合には、できる限りこの方法を使用してください。一組の CSD レコードだけを保守すればよいため、ディスク装置と時間の節約になります。しかし、この方法が使用できるのは、システム間で CSD を共用する場合だけです。CSD の共用については、「*CICS System Definition Guide*」の非 RLS モードでの CSD の共用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

動的にルーティングするトランザクションの定義には、3 つの方法があります。

注: 4 つ目の方法として二重目的定義を使用する方法 (REMOTESYSTEM オプションでデフォルトのターゲット領域を指定する方法) がありますが、動的にルーティングするトランザクションにはお勧めできません。これは、トランザクションがルーティングされたあとで、共用定義の DYNAMIC(YES) 属性によって、動的トランザクション・ルーティング・プログラムがターゲット領域で不必要に呼び出されるためです。

別々のローカル定義とリモート定義を使用する:

端末関連の EXEC CICS START コマンドで開始できるトランザクションには、この方式をお勧めします。

この方式は、 247 ページの『静的トランザクション・ルーティング』で既に説明しました。

START コマンドで開始されるトランザクションの動的ルーティングの場合には、START を発行する領域で、トランザクションを ROUTABLE(YES) として定義します。

同一定義の使用:

次のタイプのトランザクションには、この方式をお勧めします。

- CICS ビジネス・トランザクション・サービス (BTS) アクティビティーに関連付けられたトランザクション
- Enterprise Beans または CORBA ステートレス・オブジェクトに対するメソッド要求に関連付けられたトランザクション (REQUESTMODEL 定義で指定されている要求プロセッサー・トランザクション)

- 非端末関連の START コマンドで開始される可能性があるトランザクション

これらのタイプのトランザクションは、ピアツーピア・システムである分散ルーティング・モデルを使用してルーティングされます。この場合、各領域を、要求側/ルーティング領域とターゲット領域の両方にすることができます。したがって、トランザクションは、関与している各領域で同じように定義しなければなりません。領域では、CSD を共用可能にすることも共用不可にすることもできます。「*CICS System Definition Guide*」の非 RLS モードでの CSD の共用を参照してください。

各 TRANSACTION 定義では、次のようにします。

- DYNAMIC(YES) を指定します。
- REMOTESYSTEM オプションには値を指定しないでください。
- トランザクションが非端末関連の START コマンドで開始される可能性がある場合には、ROUTABLE(YES) を指定します。

「同一定義」方式は、いくつかの点で「二重目的定義」方式と異なっていることに注意してください。

- 静的ルーティングではなく、動的ルーティングに使用します。
- TRANSACTION 定義では、REMOTESYSTEM オプションを指定しないでください。
- 関与している領域で、CSD を共用する必要はありません。

TOR での単一トランザクション定義の使用:

この方式は、端末開始トランザクションにお勧めします。

この方式では、TOR (および中間システム) に、DYNAMIC(YES) を指定する *1* つのトランザクション定義しかインストールしません。この単一の定義によって、動的にルーティングされるすべての トランザクションに対しデフォルト属性群が与えられます。共通定義の名前は、DTRTRAN システム初期設定パラメーターに指定する名前です。デフォルトの名前は CRTX です。これは、CSD グループ DFHISC に組み込まれている CICS 提供のトランザクション定義の名前です。

トランザクションの接続時に、ユーザー・トランザクション ID (transid) のインストール済みリソース定義が見つからないと、CICS は、そのユーザー・トランザクション ID と、共通トランザクション定義からとった属性群からトランザクションを作成し、それに接続します。(DTRTRAN パラメーターに指定されたトランザクション定義がインストールされていないと、CICS は CICS 提供のトランザクション CSAC に接続します。このとき、メッセージ DFHAC2001-「Transaction *'tranid'* is unrecognized (トランザクション *'tranid'* が認識されていません)」) がユーザーの端末に送信されます。共通トランザクション定義に DYNAMIC(YES) が指定されているので、CICS は、動的トランザクション・ルーティング・プログラムを呼び出して、ターゲットのアプリケーション所有領域を選択し、必要に応じてリモート・トランザクションを指定します。

ターゲット AOR には、動的にルーティングするトランザクションごとにローカル定義をインストールします。

すべての端末開始トランザクションでこの方法を使用する場合は、次のようにします。

- 動的ルーティング・トランザクションを端末専有領域 (TOR に対しローカルの場合) またはアプリケーション所有領域 (AOR に対しローカルの場合) にインストールします。ただし、両方にはインストールしません。
- 動的として定義しなければならない端末開始トランザクションは、DTRTRAN パラメーターで指定される動的トランザクション・ルーティング定義だけです。
- リモートとして定義する端末開始トランザクションは、静的にルーティングするトランザクションだけです。

これによって、リソース定義の管理が非常に簡単になります。

動的ルーティングの共通トランザクション定義は、CRTX をモデルとして作成することをお勧めします。CRTX 定義に指定する属性を図 76 に示します。

```
  DEFINE
    TRANSACTION(CRTX)
    GROUP(DFHISC)
    PROGRAM(########)
    TWASIZE(00000)
    PROFILE(DFHCICST)
    STATUS(ENABLED)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
リモート属性
    DYNAMIC(YES)
    REMOTESYSTEM()
    REMOTENAME()
    TRPROF(DFHCICSS)
    ROUTABLE(NO)
リカバリー
    DTIMOUT(NO)
    INDOUBT(BACKOUT)
    RESTART(NO)
    SPURGE(YES)
    TPURGE(YES)
```

図 76. CICS 提供の CRTX トランザクションの主な属性

このトランザクション定義の主なパラメーターを下記に示します。

DYNAMIC(YES)
これは、DTRTRAN システム初期設定パラメーターで指定される動的トランザクション・ルーティング定義に必要になります。独自の定義を作成するときに他のパラメーターを変更できますが、DYNAMIC(YES) は指定しなければなりません。

PROGRAM(########)
CICS 提供のデフォルト・トランザクションには、ダミーのプログラム名 ######## が指定されています。動的トランザクション・ルーティング・プログラムがトランザクションをローカル領域で実行できるようにしている場合、その定義がダミーのプログラム名を指定していると、CICS はそのようなプログラムを見つけることができないため、「プログラムなし」条件が起こります。

このプログラム名には、次の条件のときに CICS が呼び出すべきプログラムの名前を指定してください。

- トランザクションが、リモート・システムへルーティングされず、かつ
- DYRDTRRJ パラメーターを使用した動的トランザクション・ルーティング・プログラムによって拒否されることがなく、かつ
- ローカル領域で実行される。

トランザクションをリモート・システムにルーティングできないと動的ルーティング・プログラムが判断した場合には、このローカル・プログラムを使って適切な応答をユーザーの端末に送ることができます。

TRANSACTION(CRTX)
CICS 提供の動的トランザクション・ルーティング定義の名前。これを変更して、独自のトランザクション ID を指定してください。

RESTART(NO)
ルーティングされるトランザクションの場合には、この属性が使われます。

REMOTESYSTEM
動的にルーティングされるトランザクションに対し、これによってデフォルトの AOR を指定することができます。

ROUTABLE(NO)
この属性は、EXEC CICS START コマンドで開始されるトランザクションの拡張ルーティングに関連します。

ROUTABLE(YES) を指定するということは、トランザクションが START コマンドの対象として有効な場合に、そのトランザクションは拡張ルーティング方式 (87 ページの『START コマンドで呼び出されたトランザクションのルーティング』を参照) を使用してルーティングされることを意味します。次のようにすることをお勧めします。

- 共通トランザクション定義で ROUTABLE(NO) を指定する。
- START コマンドで開始される可能性があるトランザクションの個々の定義をインストールする。

ユーザー端末から開始するトランザクションに使用する共通定義を予約することで、端末関連の START コマンドで開始するトランザクションが、「誤って」動的にルーティングされることを防ぎます。

DTP のリモート・リソースの定義

MRO および LUTYPE6.1 リンクでは、フロントエンド・システムとバックエンド・システムが直接接続されていれば、DTP のためにリモート・リソースを定義する必要はありません。フロントエンド・トランザクションによって出される EXEC CICS コマンドには、リモート・システムとリモート・トランザクションが両方とも指定されます。したがって、CICS には、セッションを接続し、バックエンド・トランザクションに接続するための情報がすべて備わっています。

ただし、バックエンド・トランザクションへルーティングする場合には、それをリモート・リソースとして中間システムに定義する必要があります (224 ページの『「デイジー・チェーン」に関する注記』を参照)。

EXEC CICS API を APPC リンクで使用する場合には、リモート・システムとリモート・トランザクションを、MRO リンクや LUTYPE6.1 リンクの場合のように明

示的に指定することも、PARTNER 定義を参照することによって指定することもできます。後者の方法を選ぶ場合には、適切な PARTNER 定義を作成する必要があります。CPI コミュニケーション API を APPC リンクで使用する場合には、コマンドの構文で、参照されるリモート・パートナーごとに PARTNER 定義を作成する必要があります。

図 77 に、CEDA DEFINE PARTNER コマンドの一般形式を示します。

```
DEFINE
  PARTNER(sym_dest_name)
  [GROUP(groupname)]
  [NETWORK(name)]
  NETNAME(name)
  [PROFILE(name)]
  {TPNAME(name)|XTPNAME(value)}
```

図 77. リモート・パートナーの定義

PARTNER リソースは、**システム・アプリケーション体系 (SAA) の規則**をサポートするために特に設計されたものです。これについて詳しくは、「*CICS Resource Definition Guide*」のPARTNER リソース定義および「*SAA CPI-C 解説書*」を参照してください。

分散トランザクション処理アプリケーションの設計と開発については、「*CICS Distributed Transaction Programming Guide*」を参照してください。

第 17 章 ローカル・リソースの定義

この章では、ローカル CICS システムにある、システム間通信に必要なリソースをどのように定義するかを説明します。

章には、以下のトピックが含まれています。

- 『通信プロファイルの定義』
- 256 ページの『体系化プロセス』
- 258 ページの『インストールが必要なリソース定義の選択』
- 259 ページの『区画内一時データ・キューの定義』
- 262 ページの『DPL のローカル・リソースの定義』.

通信プロファイルの定義

トランザクションが他のシステムとのセッションを ALLOCATE コマンドによって明示的に、または機能シップなどを使用することで暗黙的に獲得すると、そのトランザクションとセッション間の通信に通信プロファイルが関連付けられます。

この通信プロファイルには、次の情報を指定します。

- セッションから受け取った機能管理ヘッダー (FMH) を、トランザクションに渡すかどうか。
- 入出力メッセージをジャーナル処理するかどうか。処理する場合、そのジャーナルの位置はどこか。
- そのセッションにおけるエラーのノード・エラー・プログラム (NEP) クラス。
- APPC セッションの場合は、セッション・グループのモード名 (セッションはそこから割り振られる)。(プロファイルにモード名がない場合、CICS は使用可能な任意のグループからセッションを選択します。)

CICS には、この章で後に説明する一連のデフォルト・プロファイルがあり、各種の形式の通信に使用されます。しかし、独自のプロファイルを定義し、それを ALLOCATE コマンドに明示的に指定することもできます。

システム間セッションに関連する CEDA DEFINE PROFILE コマンドのオプションを 254 ページの図 78 に示します。CEDA DEFINE PROFILE コマンドの詳細については、「*CICS Resource Definition Guide*」のPROFILE 定義の属性を参照してください。

プロファイルは、ALLOCATE コマンドによって獲得されるセッションで常に必要です。ユーザーが定義して、コマンドに明示的に指定されたプロファイルか、あるいはデフォルト・プロファイルの DFHCICSA のいずれかが必要です。 CICS がプロファイルを検出できないと、アプリケーション・プログラムで CBIDERR 条件が起こります。

© Copyright IBM Corp. 1977, 2009

図 78 に示すオプションのうち MRO セッションに当てはまるのは INBFMH だけです。また、ALLOCATE コマンドによって獲得される MRO セッションの場合、CICS は、プロファイルの指定内容にかかわらず、常に INBFMH(ALL) を使用します。

APPC 会話の場合、INBFMH 指定は無視されます。APPC FMH は、決して CICS アプリケーション・プログラムに渡されません。

```
DEFINE PROFILE(name)
   [GROUP(groupname)]
   [MODENAME(name)]
   Protocols
   [INBFMH(NO|ALL)]
   Journaling
   [JOURNAL(NO|value)]
   [MSGJRNL(NO|INPUT|OUTPUT|INOUT)]
   Recovery
   [NEPCLASS(0|value)]
   [RTIMOUT(NO|value)]
```

図 78. 通信プロファイルの定義

通常、相互通信するトランザクションが、そのパートナー・トランザクションからのデータを無限に待つことが決してないようにする必要があります。RTIMOUT オプションには、システム間作業に適した値を指定する必要があります。これは、オペレーター・インターフェースとして使用される端末に通常指定されるタイムアウト期間よりもやや短めにしてください。RTIMOUT 値は、パートナー・トランザクション定義に指定された DTIMOUT 値よりも大きくなければなりません。

基本機能の通信プロファイル

プロファイルは、トランザクションとその基本機能の間の通信にも対応付けられます。CEDA DEFINE TRANSACTION コマンドにプロファイルを指定するか、またはデフォルトをとることもできます。基本機能プロファイルに対する CEDA DEFINE PROFILE コマンドは、代替機能に必要な形式よりも多くのオプションを持ちます。

バックエンド・トランザクションに定義される RTIMOUT 値は、そのフロントエンド・パートナーの基本機能に指定された値以上でなければなりません。これは、バックエンド・トランザクションが、そのフロントエンドからデータを受信するために、その期間の大半 (さらに、実行時間とネットワーク時間) 待機する場合に対応するためです。

デフォルト・プロファイル

CICS には、ユーザーがプロファイルを明示的に指定しない、あるいはできない場合に使用される一連の通信プロファイルが用意されています。

DFHCICST
基本機能のデフォルト・プロファイル。CEDA DEFINE TRANSACTION コマンドの PROFILE オプションによって、特定のトランザクションに異なるプロファイルを指定することができます。

DFHCICSV
CICS 提供トランザクション CSNE、CSLG、および CSRS の基本機能のプロファイル。これは、DVSUPRT(VTAM) が DVSUPRT(ALL) の代わりに指定される以外は、DFHCICST と同じです。

このプロファイルを修正することはできません。

DFHCICSP
CICS 提供のページ取り出しトランザクション CSPG の基本機能のプロファイル。CSPG トランザクション定義を変更して異なるプロファイルを指定しても、CICS は、CSPG にこのプロファイルを使用します。CICS 提供トランザクションで使用される通信プロファイルについての詳細は、「*CICS Supplied Transactions*」の CSPG - ページ検索を参照してください。

DFHCICSE
基本機能のエラー・プロファイル。CICS は、必要なプロファイルが見つからない場合、このプロファイルを使用して、基本機能にエラー・メッセージを渡します。

DFHCICSA INBFMH(ALL)
アプリケーション・プログラムの ALLOCATE コマンドによって獲得される代替機能のデフォルト・プロファイル。異なるプロファイルを ALLOCATE コマンドに明示的に指定することができます。

このプロファイルは、いくつかの CICS 提供トランザクションの基本機能プロファイルとして使用することもできます。

DFHCICSF INBFMH(ALL)
CICS アプリケーション・プログラムが機能シップ要求か DPL 要求を出したとき、CICS がリモート・システムまたは領域とのセッションに使用するプロファイル。

DPL を使用する場合には、RTIMEOUT に指定された値を増やす必要があることがあります。『デフォルト・プロファイルの修正』を参照してください。

DFHCICSS INBFMH(ALL)
CICS が、中継トランザクション (端末専有領域で実行) と領域間リンクまたは APPC リンクの間の通信で、トランザクション・ルーティングに使用するプロファイル。

DFHCICSR INBFMH(ALL)
CICS が、ユーザー・トランザクション (トランザクション所有領域で実行される) と領域間リンクまたは APPC リンクの間の通信で、トランザクション・ルーティングに使用するプロファイル。

ユーザー・トランザクションの基本機能は、トランザクション所有領域のサロゲート TCTTE であることに注意してください。このデフォルト・プロファイルは DFHCICST です。

デフォルト・プロファイルの修正

デフォルト・プロファイルを修正するには CEDA トランザクションを使用します。

修正の典型的な理由の 1 つに、モード名を含めることで、APPC リンクでの機能シップ要求でサービス・クラスを選択できるようにするといったことがあります。こ

れを行うには、ご使用のシステム内のすべての APPC リンクに、指定されたモード名をもつセッション・グループがあることを確認する必要があります。

いくつかの CICS 提供トランザクションによって排他使用される DFHCICSV を修正してはなりません。

CSPG ページ取り出しトランザクションによって使用される DFHCICSP は修正できます。提供される DFHCICSP には、UCTRAN(YES) が指定されています。これを UCTRAN(NO) にすると、UCTRAN(NO) が定義されている端末では、ページ取り出し機能のすべてを使用することはできません。

DFHCICSA を修正する場合は、いくつかの CICS 提供トランザクションで必要になるため、INBFMH(ALL) を保存する必要があります。このプロファイルを修正しても、MRO セッションに想定されるプロファイル・オプションは影響されません。

機能シップと DPL の要求に使用される DFHCICSF は修正できます。修正する理由の 1 つに RTIMEOUT オプションの値を増やす場合があります。例えば、このデフォルトは、単一機能シップの要求には十分かもしれませんが、データベースからレコードを続けて取り出すバックエンド・プログラムに対し DPL 呼び出しを行う場合には不十分かもしれません。

体系化プロセス

体系化プロセスは、異なる製品間で、両方の製品に理解される方法によって相互通信要求を交換できるようにする、IBM によって定義された方式です。

例えば、システム間通信の典型的な要件に、一方のシステムが他のシステムで実行するトランザクションをスケジューリングできなければならないというものがあります。CICS と IMS のいずれにも、トランザクション・スケジューラーがありますが、その具体的な方法はかなり異なります。相互通信アーキテクチャーは、「汎用の」トランザクション・スケジューリング・プロセスのモデルを定義することによって、この問題を解決しています。いずれの製品も、この体系化プロセスを独自の内部プロセスにマップすることによって実現しているので、スケジューリング要求を交換することができます。

CICS によって実施される体系化プロセスは次のとおりです。

- システム・メッセージ・モデル - システム間でやりとりする必要がある各種の情報を含むメッセージの処理 (通常は、IMS からの DFS™ メッセージ)。
- スケジューラー・モデル - スケジュール要求の処理
- キュー・モデル - キューイング要求 (CICS の用語では、一時記憶域要求または一時データ要求) の処理
- DL/I モデル - DL/I 要求の処理
- LU サービス・モデル - APPC サービス・マネージャー間の要求の処理

注: APPC LU サービス・モデルを除いて、体系化プロセスは、LUTYPE6.1 アーキテクチャーに定義されています。ただし、CICS は、APPC マイグレーション・モードを使用することによって、APPC リンクでの機能シップにもこれらを使用します。

適切なモデルは、CICS-CICS 間通信にも使用されます。例外として、CICS 定義のファイル制御モデルによって処理される CICS ファイル制御要求、および CICS にプライベートなプロトコルを使用する CICS トランザクション・ルーティングがあります。

リソース定義の際、体系化プロセスについてユーザーが関与するのは、CICS システムに適切なトランザクションとプログラムが含まれるようにすることと、必要な場合にそれらの優先度を変更することだけです。

プロセス名

体系化プロセス名は、1 バイトから 4 バイトの長さで、最初のバイト値は X'40' 未満になります。

CICS では、この名前は、4 バイトの 16 進トランザクション ID として指定されます。CICS は、4 バイト未満の体系化プロセス名を受け取ると、ヌル文字 (X'00') でこの名前を埋めてから、トランザクション ID を検索します。

CICS は、図 79 に示すプロセスを提供します。

```
  XTRANID     TRANSID     PROGRAM     DESCRIPTION
For CICS file control
     -        CSMI        DFHMIRS     File control model

For LUTYPE6.1 architected processes
  01000000    CSM1        DFHMIRS     System message model
  02000000    CSM2        DFHMIRS     Scheduler model
  03000000    CSM3        DFHMIRS     Queue model
  05000000    CSM5        DFHMIRS     DL/I model

For APPC architected processes
  06F10000    CLS1        DFHZLS1     LU services model
  06F20000    CLS2        DFHLUP      LU services model
     -        CLS3        DFHLUP      LU services model
```

図 79. CICS 体系化プロセス名

体系化プロセス定義の修正

CEDA トランザクションによって、いくつかの定義を修正することができます。特に、ミラー・トランザクションの DTIMOUT 値を変更することができます。

前述のリストは、CICS ファイル制御モデルと機能シップの体系化プロセスすべてが、CICS ミラー・プログラムであるプログラム DFHMIRS にマップされることを示しています。各種のモデルのさまざまなトランザクション名を組み込むと、いくつかのトランザクション属性を修正することができます。ただし、XTRANID、TRANSID、または PROGRAM の各値を変更することはできません。

ミラー・トランザクションの定義には、DTIMOUT(NO) が指定されています。この状態が望ましくない場合は、この定義を変更して、DTIMOUT オプションに NO 以外の値を指定する必要があります。ただし、これらの定義を変更するにあたっては、まず新しいグループにこれらをコピーする必要があります。

領域間機能シップ

MRO リンクを介した機能シップは、長期実行ミラー・タスクと短パス変換プログラムを使用することができます。

(44 ページの『MRO 機能シップ』を参照。)

1 つまたは複数のミラー・トランザクション定義を修正する場合は、このことによって領域間機能シップに生じる影響を評価する必要があります。

短パス変換プログラムは、常にトランザクション CSMI を指定します。ただし、これは DL/I 要求には使用されません。これらの要求は、トランザクション CSM5 に対応する、プロセス X'05000000' に対する要求として到着します。

インストールが必要なリソース定義の選択

この章で説明するプロファイルと体系化プロセス、および ISC と MRO に必要なその他のトランザクションとプログラムは、IBM 保護グループの DFHISC と DFHSTAND に含まれています。

このタスクについて

これらの事前生成された CEDA グループを CICS システムに組み込む方法については、「*CICS Resource Definition Guide*」のCICS 提供のリソース定義、グループ、およびリストを参照してください。

グループ DFHISC と DFHSTAND の内容の一部を要約したものを 259 ページの図 80 に示します。

```
TRANSACTIONS
XTRANID   TRANSID   PROGRAM   GROUP
   -       CSMI     DFHMIRS   DFHISC   CICS file control model
01000000   CSM1     DFHMIRS   DFHISC   System message model
02000000   CSM2     DFHMIRS   DFHISC   Scheduler model
03000000   CSM3     DFHMIRS   DFHISC   Queue model
05000000   CSM5     DFHMIRS   DFHISC   DL/I model
06F10000   CLS1     DFHZLS1   DFHISC   LU services model
06F20000   CLS2     DFHLUP    DFHISC   LU services model
   -       CLS3     DFHLUP    DFHISC   LU services model
   -       CEHP     DFHCHS    DFHISC   CICS/VM request handler
   -       CEHS     DFHCHS    DFHISC   CICS/VM request handler
   -       CMPX     DFHMXP    DFHISC   Local queue shipper
   -       CPMI     DFHMIRS   DFHISC   Synclevel 1 mirror
   -       CRSQ     DFHCRQ    DFHISC   Remote schedule purge program
   -       CRSR     DFHCRS    DFHISC   Remote scheduler program
   -       CRTE     DFHRTE    DFHISC   Routing transaction
   -       CSNC     DFHCRNP   DFHISC   Interregion connection manager
   -       CSSF     DFHRTC    DFHISC   CRTE cancel command processor
   -       CVMI     DFHMIRS   DFHISC   APPC sync level-1 mirror
   -       CXRT     DFHCRT    DFHISC   Relay transaction for LU6.2

PROGRAMS
NAME      GROUP
DFHCCNV   DFHISC    CICS data conversion program
DFHCRNP   DFHISC    Interregion new connection manager
DFHCRQ    DFHISC    ATI purge program
DFHCRR    DFHISC    IRC session recovery program
DFHCRS    DFHISC    Remote scheduler program
DFHCRSP   DFHISC    Interregion control initialization program
DFHCRT    DFHISC    Transaction routing relay program for APPC
                     alternate facilities
DFHDYP    DFHISC    Standard dynamic transaction routing program
DFHLUP    DFHISC    LU services program
DFHMIRS   DFHISC    Mirror program
DFHMXP    DFHISC    Local queuing shipper program
DFHRTC    DFHISC    CRTE cancel command processor
DFHRTE    DFHISC    Transaction routing program

PROFILES
NAME      GROUP
DFHCICSF  DFHISC    Function shipping profile
DFHCICSR  DFHISC    Transaction routing receive profile
DFHCICSS  DFHISC    Transaction routing send profile
DFHCICSA  DFHSTAND  Distributed transaction processing profile
DFHCICSE  DFHSTAND  Principal facility error profile
DFHCICST  DFHSTAND  Principal facility default profile
DFHCICSV  DFHSTAND  Principal facility special profile
```

図 80. ISC と MRO に必要な定義の一部

区画内一時データ・キューの定義

区画内一時データ・キューの定義は、次のように行うことができます。

```
DEFINE
  TDQUEUE(name)
  GROUP(groupname)
  DESCRIPTION(text)
  TYPE(Intra)
  Intrapartition Attributes
  ATIFACILITY(terminal)
  RECOVSTATUS(logical)
  FACILITYID (terminal)
  RECOVSTATUS(name)
  TRANSID ()
  TRIGGERLEVEL(value)
  USERID(userid)
  Indoubt Attributes:
  WAIT(yes)
  WAITACTION(reject)
        ...
```

図 81. 区画内一時データ・キューの定義

一時データ・キューの定義についての詳細は、「*CICS Resource Definition Guide*」のTDQUEUE リソースの定義を参照してください。この項では、次のことを行うキューについて、CICS 相互通信の観点から説明します。

- 自動トランザクション開始を行う。
- 関連する基本機能を指定する (端末や別のシステムなど)。

トランザクション

区画内一時データ・キューによって開始されるトランザクションは、キューと同じシステム上になければなりません。つまり、キュー定義に指定するトランザクションは、リモート・トランザクションとして定義することはできません。

基本機能

ATI によって開始されるトランザクションに関連付ける基本機能は、一時データ・キューの定義に指定します。

基本機能としては次のものが可能です。

- ローカル端末
- リモート端末
- ローカル・セッションまたは APPC デバイス
- リモート APPC セッションまたはデバイス

ローカル端末

ローカル端末とは、一時データ・キューとトランザクションを所有する同じシステムによって所有される端末のことをいいます。

APPC 端末以外のローカル端末には、端末宛先と端末 ID が必要です。端末 ID を省略すると、端末の名前はキューの名前になります。

リモート端末

リモート端末とは、一時データ・キューとそれに関連するトランザクションを所有するシステム上で、リモートと定義された端末のことをいいます。

リモート端末での自動トランザクション開始は、CICS トランザクション・ルーティング
(73 ページの『第 7 章 CICS トランザクション・ルーティング』を参照) の一種であり、通常のトランザクション・ルーティングの規則が適用されます。

APPC 端末以外のリモート端末には、端末宛先と端末 ID を指定する必要があります。

端末自体がリモート端末として定義され (または、シップされた端末定義が使用可能にされ)、その端末専有領域が IRC リンクか APPC リンクによってローカル・システムに接続されていなければなりません。

ローカル・セッションと APPC デバイス

一時データ・キューの定義にローカル接続の定義を指定することができます。リモート・システムは、IRC、LUTYPE6.1、または APPC のいずれかのリンクによって接続することができます。APPC の場合、「システム」は、ハードコーディングされた端末のようなデバイスで構いません。

CICS は、指定されたシステムにセッションを割り振ります。そして、これが **transid** に対する基本機能になります。トランザクション・プログラムは、適切な DTP プロトコルを使用して、そのセッションで会話をします。DTP の紹介については、 117 ページの『第 9 章 分散トランザクション処理』を参照してください。

トランザクションは、その基本機能で「割り振られた」状態で開始されます。トランザクションは次に、そのパートナー・トランザクション (つまり、セッションのもう一方の端に接続されるプロセス) を識別します。APPC プロトコルの場合、トランザクションは、EXEC CICS CONNECT PROCESS コマンドを出して、この識別を行います。このコマンドは通常、代替機能で会話を開始するためにのみ使用されます。

パートナー・トランザクションは、バックエンドにおいて、受信状態の会話によって開始されていますが、この場合もセッションをその基本機能と見なします。これは、CICS が基本機能としてどちらかのセッションの終わりを扱うという点で珍しいケースです。両側で、会話 ID は、必要に応じて EIBTRMID からとられますが、基本機能の場合と同じように、その後のコマンドでも暗黙指定されます。

リモート APPC セッションおよびデバイス

リモート接続は、その一時データ・キューと関連トランザクションを所有するシステムにリモートとして定義します。

リモート APPC 接続での自動トランザクション開始は、CICS トランザクション・ルーティング (73 ページの『第 7 章 CICS トランザクション・ルーティング』を参照) の一種であり、通常のトランザクション・ルーティングの規則が適用されます。

リモート接続は、一時データ・キューの定義に指定することができます。

接続自体がリモート接続として定義され (または、シップされた接続定義が使用可能にされ)、その端末専有領域が IRC リンクか APPC リンクによってローカル・シ

ステムに接続されていなければなりません。 261 ページの『ローカル・セッションと APPC デバイス』のトランザクション開始後のリンク処理に関する注釈は、ルーティングされたトランザクションにも適用されます。

DPL のローカル・リソースの定義

DPL をサポートする場合、サーバー・プログラムとミラー・トランザクションに、特殊リソース定義が必要になる場合があります。

ミラー・トランザクション

DPL 要求によって開始されるミラー・トランザクションには、任意の名前を指定することができます。このトランザクション名は、サーバー領域において、ミラー・プログラム DFHMIRS を呼び出すトランザクションに定義する必要があります。

ユーザー・トランザクションを定義してミラー・プログラムを呼び出すと、トランザクション・リソース定義の他のオプションすべてに、適切な値を自由に指定することができます。

サーバー・プログラム

ローカル・プログラムが、他の領域によって DPL サーバーとして要求される場合は、そのプログラムのリソース定義が必要です。

この定義は、静的に定義することもできますし、そのプログラムが最初に呼び出されたときに自動的にインストール (自動インストール) することもできます。プログラム用の CICS 自動インストール機能の詳細については、「*CICS Resource Definition Guide*」のプログラム、マップ・セット、および区画セットの自動インストールを参照してください。

第 4 部 システム間環境におけるアプリケーション・プログラミング

このパートでは、CICS 相互通信のアプリケーション・プログラミングについて説明します。

このパートには、以下の章が含まれています。

- 265 ページの『第 18 章 アプリケーション・プログラミングの概要』
- 267 ページの『第 19 章 CICS 機能シップのアプリケーション・プログラミング』
- 271 ページの『第 20 章 CICS DPL のアプリケーション・プログラミング』
- 277 ページの『第 21 章 非同期処理のアプリケーション・プログラミング』
- 279 ページの『第 22 章 CICS トランザクション・ルーティングのアプリケーション・プログラミング』
- 283 ページの『第 23 章 CICS-IMS 間アプリケーション』.

分散トランザクション処理のアプリケーション設計およびプログラミングについては、「*CICS Distributed Transaction Programming Guide*」を参照してください。

このパートには、汎用プログラミング・インターフェースとそれに関連するガイダンス情報が記載されています。

© Copyright IBM Corp. 1977, 2009

第 18 章 アプリケーション・プログラミングの概要

CICS 相互通信環境で実行するように設計されたアプリケーション・プログラムでは、以下の機能のうち 1 つまたは複数を使用することができます。

- 機能シップ
- 分散プログラム・リンク
- 非同期処理
- トランザクション・ルーティング
- 分散トランザクション処理

これらの機能に対するアプリケーション・プログラミングの要件については、このパートの各章で個別に説明します。アプリケーション・プログラムが複数の機能を使用している場合は、それに対応するプログラム・フラグメントの設計の援助として、該当する章を使用することができます。同様に、プログラムが、分散トランザクション処理に複数のシステム間セッションを使用している場合は、適切なセッション・タイプについての規則に従って、各セッションを制御する必要があります。

分散トランザクション処理のアプリケーション設計およびプログラミングについては、「*CICS Distributed Transaction Programming Guide*」を参照してください。

用語

次の用語は、このパートの以下の章で、詳しい説明なしに使用される場合があります。

基本機能
この用語は、トランザクションの開始時にそのトランザクションに関連付けられる端末やセッションを意味します。SEND や RECEIVE などの、機能を明示的に指定しない CICS コマンドは、基本機能を指すものと見なされます。1 つのトランザクションが所有できるのは 1 つの基本機能だけです。

代替機能
分散トランザクション処理では、トランザクションが、リモート・システムとの間におけるセッションの使用を獲得することができます。このセッションは、代替機能と呼ばれます。これは、その機能を参照する CICS コマンドに明示的に指定する必要があります。 1 つのトランザクションが、複数の代替機能を所有することができます。

機能シップなどに使用される他のシステム間セッションは、トランザクションによって所有されず、トランザクションの代替機能と見なされません。

フロントエンド・トランザクションおよびバックエンド・トランザクション
分散トランザクション処理では、対になったトランザクションが互いに会話をします。*フロントエンド・トランザクション* がまず初めに開始され、リモート・システムとのセッションを獲得します。そして、*バックエンド・トランザクション* が開始されます。

© Copyright IBM Corp. 1977, 2009

1 つのトランザクションが、同時に、ある会話でのバックエンド・トランザクションになり、1 つまたは複数の別の会話でのフロントエンド・トランザクションになる場合があることに注意してください。

問題判別

CICS 相互通信機能を使用するアプリケーション・プログラムでは、単一の CICS システムでは起こらないエラー条件が起こるおそれがあります。

リソースがリモートの場合、機能管理もリモートで行われるため、トランザクション異常終了は、リモート・トランザクションで起こります。さらに、ローカル・トランザクションも、リモート・トランザクションを異常終了させるために使用された特定のコードではなく、トランザクション異常終了コード ATNI (VTAM を介した通信の場合) または AZI6 (MRO を介した通信の場合) を出して異常終了します。ただし、リモート・システムは、ローカル CICS システムに、リモート障害の理由を示すエラー・メッセージを送ります。このメッセージは、ローカル CSMT 宛先に送られます。したがって、リソースへのアクセス中に異常終了が起こったときに、アプリケーション・プログラムが HANDLE ABEND を使って処理を続ける場合、これらのリソースがリモートにあると、同じ方法で処理を続けることはできません。

トレース機能とダンプ機能は、ローカルおよびリモートの各 CICS システムに定義されています。リモート・トランザクションが異常終了すると、その CICS トランザクション・ダンプをリモート・サイトで使用して、異常終了条件の理由を見つけることができます。

アプリケーションでリモート・システムを使用する場合には、リモート・リソースにアクセスしたときに障害が起こらないよう、十分テストする必要があります。「リモート・テスト・システム」をローカル・システムと同じプロセッサーに常駐させて、両方のシステムのトランザクション・ダンプ、および対応するトレース・データをすぐに使用できる単一の場所でテストできることを覚えておいてください。2 つのトランザクションは、MRO を介して、あるいは VTAM アプリケーション間機能を介して接続することができます。

CICS 相互通信の問題の診断に関する順序および要求形式についての詳細は、「*CICS Problem Determination Guide*」を参照してください。

第 19 章 CICS 機能シップのアプリケーション・プログラミング

この章には以下のトピックが含まれています。

- 『機能シップのためのプログラミングの紹介』
- 268 ページの『ファイル制御』
- 268 ページの『DL/I』
- 268 ページの『一時記憶域』
- 269 ページの『一時データ』
- 269 ページの『機能シップの例外条件』.

機能シップのためのプログラミングの紹介

リモート・システムのリソースにアクセスするプログラムを作成する場合も、コーディングの方法は、それらのリソースがローカル・システムにある場合とほとんど同じです。機能シップは、EXEC CICS コマンド、DL/I 呼び出し、または EXEC DLI コマンドで使用できます。

リモート・リソースにアクセスするために使用できるコマンドは、次のとおりです。

- ファイル制御コマンド
- DL/I 呼び出しまたは EXEC DLI コマンド
- 一時記憶域コマンド
- 一時データ・コマンド

インターバル制御コマンドについては、 277 ページの『第 21 章 非同期処理のアプリケーション・プログラミング』を参照してください。

アプリケーションは、CICS 相互通信環境で実行して、アクセスするリソースの位置を意識することなく、相互通信機能を使用することができます。リソースの位置は、リソース定義に指定されています。任意指定として、SYSID オプションを EXEC コマンドに指定すれば、そのコマンドを実行するシステムを選択することができます。この場合、ローカル・システムのリソース定義は、SYSID オプションでそのローカル・システムを指定する場合を除いて、参照されません。

アプリケーションがリモート・リソースに対してコマンドを出すと、CICS はその要求をリモート・システムにシップし、そこでミラー・トランザクションが開始されます。ミラー・トランザクションは、ユーザーに代わってこの要求を実行し、アプリケーション・プログラムに出力を返します。ミラー・トランザクションは、アプリケーション・プログラムのリモート拡張機能になります。このメカニズムの詳細については、 37 ページの『第 4 章 CICS 機能シップ』を参照してください。

ローカル・リソースおよびリモート・リソースへのアクセスには、同じコマンドが使用されますが、リソースがリモートの場合には、制約が適用されます。また、機能シップを使用すると、単一システムでは起こらないエラーが起こる可能性があり

© Copyright IBM Corp. 1977, 2009

ます。このため、プログラムがアクセスするリソースがリモートである可能性があるかどうかを、常に知っている必要があります。

ファイル制御

機能シップを使用すると、リモート・システムにあるファイルにアクセスすることができます。

SYSID オプションを使用して、リモート・システムに直接アクセスする場合は、次の 2 つの規則を守る必要があります。

1. キー順データ・セットを参照するファイルでは、相対バイト・アドレス (RBA) または相対レコード番号 (RRN) を使用している場合を除き、RIDFLD が指定されているときは KEYLENGTH を指定しなければなりません。

   リモート BDAM ファイルにおいて、DEBKEY オプションか DEBREC オプションを指定する場合には、KEYLENGTH はキーの長さの合計でなければなりません。
2. ファイルが固定長レコードの場合には、レコード長 (LENGTH) を指定しなければなりません。

これらの規則は、この CICS に対するファイルの定義が適切な値を指定していない場合にも適用されます。

DL/I

機能シップを使用すると、リモート CICS システムに関連した IMS Database Manager サブシステム、あるいはリモートの CICS Transaction Server for VSE システムに関連したデータベースにアクセスすることができます。

制約事項については、「*CICS IMS Database Control Guide*」を参照してください。

一時記憶域

機能シップを使用すると、リモート・システムにある一時記憶域キューとの間でデータをやりとりすることができます。リモート一時記憶域キューの定義は、システム・プログラマーが作成することができます。しかし、SYSID オプションを WRITEQ TS、READQ TS、および DELETEQ TS の各コマンドに使用すると、要求が実行されるシステムを指定することができます。

MRO セッションの場合、MAIN および AUXILIARY の各オプションを WRITEQ TS コマンドに使用すると、必要なタイプのストレージを選択することができます。

APPC セッションの場合、MAIN および AUXILIARY の各オプションは無視されます。リモート・システムでは常に補助記憶装置が使用されます。

一時データ

機能シップを使用すると、リモート・システムにある区画内または区画外の一時データ・キューにアクセスすることができます。リモート一時データ・キューの定義は、システム・プログラマーが作成することができます。しかし、SYSID オプションを WRITEQ TD、READQ TD、および DELETEQ TD の各コマンドに使用すると、要求が実行されるシステムを指定することができます。

リモート一時データ・キューに固定長レコードがある場合、インストール済みの一時データ・リソース定義にレコード長が指定されていないなら、それを指定しなければなりません。

機能シップの例外条件

リモート・システムにシップされる要求によって、リソースがローカルの場合に発生する可能性がある、コマンドの例外条件のいずれかが発生することがあります。

また、リソースがリモートの場合にのみ適用されるいくつかの条件もあります。

リモート・システム使用不能

SYSIDERR 条件は、ある特定の状態になった場合に、アプリケーション・プログラムで発生します。

- 例えば、以下の状態になった場合です。
- リモート・システムへのリンクに対するサービスが停止した場合。
- 指定のシステムが定義されていない場合。このエラーは、アプリケーションが端末オペレーターからリモート・システムの名前を入手するように設計されている場合を除いて、実動システムでは起こりません。
- リモート・システムへのリンクが使用中で、かつ要求数が、CONNECTION 定義の QUEUELIMIT オプションに指定されているキューイング要求の最大数に達した場合。
- リモート・システムへのリンクが使用中で、要求数がキューイング要求の最大数には達していない が、XZIQUE または XISCONA グローバル・ユーザー出口プログラムによって、その要求をキューイングしないように指定した場合XZIQUE 出口および XISCONA 出口のプログラミング情報については、「*CICS Customization Guide*」を参照してください。

SYSIDERR 条件に対するデフォルト・アクションは、タスクの異常終了です。

無効な要求

ISCINVREQ 条件は、リモート・システムが、既知の条件に対応しない障害を示すと発生します。このデフォルト・アクションは、タスクの異常終了です。

ミラー・トランザクションの異常終了

リモート・リソースに対するアプリケーション要求によって、リモート CICS のミラー・トランザクションが異常終了する場合があります。例えば、デッドロック・タイムアウトによって、ミラーが ATSC というコードで異常終了することがあります。

このような状態では、アプリケーション・プログラムも異常終了しますが、異常終了コードの AIPM (IPIC 接続の場合)、ATNI (ISC 接続の場合)、または AZI6 (MRO 接続の場合) が出されます。エラー条件は、CICS によって、CSMT 宛先に送られるエラー・メッセージに記録されます。アプリケーションによって出される HANDLE ABEND コマンドは、条件の原因を識別して、明示的な訂正アクションをとることができません。リソースがローカルである場合、訂正アクションが実行可能なこともあります。ミラー・トランザクションが DL/I プログラム分離デッドロックによって異常終了した場合には、MRO 機能シップで例外が発生します。この場合、アプリケーションは、通常のデッドロック異常終了コード (ADCD) を出して異常終了します。

ミラー・トランザクション異常終了によって発生した ATNI 異常終了は、端末管理コマンドには関連しないため、TERMERR 条件は起こりません。

第 20 章 CICS DPL のアプリケーション・プログラミング

この章には以下のトピックが含まれています。

- 『DPL プログラミングの紹介』
- 『クライアント・プログラム』
- 272 ページの『サーバー・プログラム』
- 273 ページの『DPL の例外条件』.

DPL プログラミングの紹介

CICS 分散プログラム・リンク (DPL) を使用すると、リモート・システムにあるサーバー・プログラムにリンクすることができます。

CICS Transaction Server for z/OS 領域で稼働するクライアント・プログラムは、リモート CICS 領域で稼働する 1 つまたは複数のサーバー・プログラムにリンクすることができます。リモート領域は、CICS Transaction Server for z/OS システムであっても、他のシステムであっても構いません。CICS Transaction Server for z/OS の通信相手となるシステムのリストについては、3 ページの『第 1 章 CICS 相互通信の紹介』を参照してください。

DPL プログラムは、PL/I、C、COBOL、またはアセンブラー言語で作成することができます。

105 ページの『第 8 章 CICS 分散プログラム・リンク』の説明のように、DPL には、クライアント・プログラムとサーバー・プログラムという 2 つの側 (プログラム) があります。DPL を実行するには、各プログラムがとらなければならないいくつかのアクションがあります。これらのアクションについて、以下に説明します。

クライアント・プログラム

リモート・システムのサーバー・プログラムにリンクするクライアント・プログラムを作成する場合も、コーディングの方法は、それらのサーバー・プログラムがローカル・システムにある場合とほとんど同じです。

クライアント・プログラムは、リンクするサーバー・プログラムの位置を知らなくても、CICS 相互通信環境で実行して、相互通信機能を使用することができます。サーバー・プログラムの位置は、動的ルーティング・プログラムのプログラム・リソース定義で指定されます。必要に応じて、LINK コマンドで SYSID オプションを使用して、コマンドを実行するシステムを選択することができます。

クライアント・プログラムがサーバー・プログラムに対して LINK コマンドを出すと、CICS はその要求をリモート・システムにシップし、そこでミラー・トランザクションが開始されます。ミラー・トランザクションは、ユーザーに代わってこの LINK 要求を実行し、サーバー・プログラムを実行させます。サーバー・プログラムが RETURN コマンドを出すと、ミラー・トランザクションは、連絡域データをクライアント・プログラムに返します。ミラー・トランザクションは、アプリケー

© Copyright IBM Corp. 1977, 2009

ション・プログラムのリモート拡張機能になります。このメカニズムの詳細については、 105 ページの『第 8 章 CICS 分散プログラム・リンク』を参照してください。

同じコマンドを使用して、ローカルおよびリモートの両方のサーバー・プログラムにアクセスすることができますが、サーバー・プログラムがリモートの場合には、いくつかの制約事項が適用されます。また、DPL を使用すると、単一システムでは起こらないエラーが起こる可能性があります。このため、クライアント・プログラムがリンクするサーバー・プログラムがリモートかどうかを常に検出しなければなりません。サーバー・プログラムがリモートの可能性がある場合は、クライアント・プログラムに、リモート・サーバー・プログラムによって返される可能性のある例外条件に対する追加検査を組み込む必要があります。

サーバー・プログラムの障害

サーバー・プログラムに障害が起こると、ABEND 条件と異常終了コードがクライアント・プログラムに返されます。したがって、HANDLE ABEND コマンドを出してから LINK コマンドを出さない限り、クライアント・トランザクションも異常終了します。

サーバー・プログラム

許可されているコマンド

DPL サーバー・プログラムが発行できる EXEC CICS コマンドは、CICS API のサブセットに限定されます。

制限された DPL サブセットの詳細については、「*CICS アプリケーション・プログラミング・リファレンス*」の 分散プログラム・リンクの API に関する制約事項 (API restrictions for distributed program link)を参照してください。

同期点

サーバー・プログラムが、SYNCONRETURN オプションが指定された LINK コマンドによって開始されている場合、そのプログラムは同期点を出すことができます。

しかし、サーバー・プログラムが同期点を出しても、クライアント・プログラムによって行われた変更はコミットされ**ません**。分散作業単位全体で変更をコミットするには、クライアント・プログラムが同期点を出す必要があります。クライアント・プログラムは、サーバー・プログラムがその変更をまだコミットしていなければ、分散作業単位全体で変更をバックアウトすることもできます。

サーバー・プログラムは、ASSIGN STARTCODE コマンドを出すことにより、その開始方法と同時に、独立した同期点要求を出すことができるかどうかを知ることができます。このコマンドは、DPL サーバー・プログラムに関連する次の値を返します。

- SYNCONRETURN オプションの指定が**ない** LINK 要求によって開始されたため、プログラムから SYNCPOINT 要求を出せない場合には、「D」を返します。

- SYNCONRETURN オプションの指定が**ある** LINK 要求によって開始されたため、プログラムから SYNCPOINT 要求を出せる場合には、「DS」を返します。しかし、サーバー・プログラムは、明示的に同期点要求を出す必要はありません。これは、CICS が、サーバー・プログラムが RETURN コマンドを出すとすぐに同期点をとるためです。
- プログラムがリモート LINK 要求によって開始されていない場合には、「D」や「DS」以外の値。

DPL の例外条件

リモート・システムにシップされる LINK 要求によって、サーバー・プログラムがローカルの場合に発生する可能性がある、コマンドの例外条件のいずれかが発生することがあります。

また、サーバー・プログラムがリモートの場合にのみ適用されるいくつかの条件もあります。

リモート・システム使用不能

リモート・システムが利用できない場合、 269 ページの『リモート・システム使用不能』の機能シップに関する説明にある理由とまったく同じ理由で SYSIDERR 条件がクライアント・プログラムで起こることがあります。

SYSIDERR 条件に対するデフォルト・アクションは、タスクの異常終了です。

サーバー作業のバックアウト

クライアント・プログラムが SYNCONRETURN オプションを指定して LINK コマンドを出すと、ミラー・プログラムは、サーバー・プログラムが正常に終了するとすぐに、同期点を出します。

この同期点に障害が起こる可能性があります。障害が起こった場合は、ROLLEDBACK 条件がクライアント・プログラムに返されます。サーバー・プログラムによって行われた作業も、サーバー・プログラムが*独自の同期点要求を出して作業を既にコミットしていない限り*、バックアウトされます。

同じサーバー領域に対する複数のリンク

クライアント・プログラムが SYNCONRETURN オプションを*指定*して LINK コマンドを出すと、ミラー・トランザクションは、制御がクライアント・プログラムに返されるとすぐに終了します。したがって、クライアント・プログラムは、同じサーバー領域に続けて LINK コマンドを出すことができます。

しかし、クライアント・プログラムが SYNCONRETURN オプションを*指定しない*で LINK コマンドを出すと、ミラー・トランザクションは、クライアント領域から同期点要求がくるまで待機します。クライアント・プログラムは、SYNCONRETURN オプションが省略されていて、TRANSID 値が変更されない限り、同じサーバー領域に対して続けて LINK コマンドを出すことができます。

SYNCONRETURN オプションが指定されているか、異なる TRANSID 値が指定されている、それ以降の LINK コマンドは、その前に SYNCPOINT コマンドが出されていない限り失敗します。

注: クライアント・プログラムが機能シップ要求をサーバー領域に送信し、その機能シップ要求のためのミラーが延期される場合にも、同じ考慮事項が当てはまります。以下に例を挙げます。

```
EXEC CICS LINK PROGRAM('PGA') SYSID(SERV)
EXEC CICS SYNCPOINT
EXEC CICS READQ TS QUEUE('RQUEUE') SYSID(SERV)
EXEC CICS LINK PROGRAM('PGB') SYSID(SERV) TRANSID(TRN1)
```

最後の LINK コマンドは、例えば、CICS サーバー領域 (SERV) に MROLRM=YES が指定されている場合には、失敗します。これは、READQ TS コマンドのミラーが依然として生きているからです。上記の順序のコマンドを機能させるためには、クライアント・プログラムは、READQ TS コマンドのあとに SYNCPOINT を出す必要があります。あるいは、サーバー領域の MROLRM システム初期設定パラメーターを「NO」に設定することもできます。DPL 要求と機能シップ要求を同じプログラムで使用する方法については、「*CICS アプリケーション・プログラミング・ガイド*」の 同じ CICS システムでの DPL と機能シップの混合を参照してください。

これらのエラーは、INVREQ 条件と PGMIDERR 条件によって示されます。

INVREQ 条件では、14 という RESP2 値は、障害が発生した LINK コマンドを正しく試行するには同期点が必要であることを示します。15 という RESP2 値は、TRANSID 値が、リンクされたミラー・トランザクションの値と異なることを示します。16 という RESP2 値は、スペース (ブランク) の TRANSID 値が LINK コマンドに指定されたことを示します。17 という RESP2 値は、スペース (ブランク) の TRANSID 値が動的ルーティング・プログラムで提供されたことを示します。

PGMIDERR 条件では、25 という RESP2 値は、動的ルーティング・プログラムがリンク要求を拒否したことを示します。

ミラー・トランザクションの異常終了

サーバー・プログラムとは対照的に、ミラー・プログラムが異常終了するか、サーバー領域とのセッションに障害が発生すると、TERMERR 条件がクライアント・プログラムに返されます。

同じ分散 UOW による 1 つのリカバリー可能リソースに対する複数の更新

非 DPL 環境では、1 つの作業単位 (UOW) 内にある複数のプログラムが同一のリカバリー可能リソースを更新する可能性があります。

例えば、program1 がリカバリー可能ファイル内の Record1 を更新し、その後で program2 にリンクした場合、program2 が同じファイル内の同じレコード Record1 を更新する可能性があります。これは、プログラミング慣行としては必ずしもお勧めできませんが、CICS は、リソースの所有者がプログラムではなくタスクであると見なすため、このようなことが可能になっています。

しかし、DPL 環境では、関係するプログラムがさまざまな CICS 領域 で実行されているため、複数のプログラムが同じ UOW 内にある同じリカバリー可能リソースを更新することはありえません。上記と同じ例で、program1 がリカバリー可能ファイル内の Record1 を更新してから、別の領域内のミラー・タスクで実行されている program2 にリンクします。program2 が同じファイル内の Record1 を更新するためにファイル制御要求を機能シップすると、要求はハングします。要求がハングするのは、program2 のファイル制御要求を処理しているミラー・タスクが Record1 のレコード・ロックを獲得することができないためです。このロックは、program1 の実行が行われているタスクによって所有されています。ファイル制御ミラー・タスクと、program1 が実行されているタスクとが、同じ分散 UOW の一部であっても、CICS は更新を許可しません。これは、CICS がリカバリー可能リソースのロックの基礎として、分散 UOW ではなくタスクを使用しているためです。

第 21 章 非同期処理のアプリケーション・プログラミング

この章では、CICS 間の非同期処理のアプリケーション・プログラミング要件について説明します。

START コマンドまたは RETRIEVE コマンドを使用する CICS トランザクションに関する一般情報は、CICS-IMS 間通信にも適用されます。

非同期処理の概念については、51 ページの『第 5 章 非同期処理』に説明があります。ここでは、読者が CICS インターバル制御機能の概念を理解していることを前提としています。インターバル制御機能での EXEC CICS 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、「*CICS アプリケーション・プログラミング・リファレンス*」の STARTを参照してください。

リモート・システムでのトランザクションの開始

ローカル・トランザクションと同様に EXEC CICS START コマンドを出して、リモート・システムでトランザクションを開始することができます。

このタスクについて

一般に、トランザクションは、システム・プログラマーによって、リモートとして定義されています。しかし、SYSID オプションにリモート・システムを明示的に指名することもできます。したがって、このような START コマンドの使用は、実際には CICS 機能シップの特殊なケースです。

アプリケーションで、リモート・トランザクションの開始時刻を指定する必要がある場合は、リモート・システムの時間帯が異なる可能性があることに注意してください。このような状況では、INTERVAL 形式の制御を使用するようにしてください。

START コマンドの例外条件

リモート・トランザクションに START 要求を出したあとに起こる可能性がある例外条件は、START コマンドに NOCHECK パフォーマンス・オプションが指定されているかどうかによって異なります。

NOCHECK が指定されていない場合、条件の発生は、通常の機能シップの規則に従います (269 ページの『機能シップの例外条件』を参照)。

NOCHECK が指定されている場合は、START コマンドがリモート実行されても条件はなにも起こりません。ただし、システム・プログラマーが開始要求のローカル・キューを用意していない場合は、リモート・システムへのリンクが使用できないと、SYSIDERR 条件が起こります (58 ページの『START コマンドのローカル・キューイング』を参照)。

© Copyright IBM Corp. 1977, 2009

リモートで出された開始要求に関連したデータの検索

RETRIEVE コマンドを使用すると、リモートから出された開始要求の結果としてタスクで保管されたデータを検索することができます。これは、このようなデータにアクセスするためにのみ使用可能な方式です。

このタスクについて

トランザクションに関する限りは、リモート開始要求によって保管されたデータと、ローカル開始要求によって保管されたデータの間に区別がありません。そのため、RETRIEVE コマンドの使用に関しては通常の考慮事項が適用されます。

第 22 章 CICS トランザクション・ルーティングのアプリケーション・プログラミング

トランザクション・ルーティング環境で使用できるトランザクションを作成する場合、一般には、単一の CICS システムの場合とまったく同じように設計し、コーディングすることができます。

ただし、注意すべき制約事項がいくつかあります。この章では、これらの制約事項について説明します。既存のトランザクションをトランザクション・ルーティング環境にマイグレーションする場合、同じ考慮事項が適用されます。

アプリケーション・プログラミングの制限

トランザクション・ルーティングのためのアプリケーション・プログラムを記述する際には、複数の制限と考慮事項に注意する必要があります。

プログラムは、PL/I、COBOL、C、またはアセンブラー言語で作成することができます。この選択は、もちろん、端末タイプやセッション・タイプによって制約される場合があります。例えば、基本的な APPC 会話は、C かアセンブラー言語で作成する必要があります。

基本マッピング・サポート

プログラムで使用する BMS マップ・セットや区分セットは、そのプログラムと同じ CICS システム内になければなりません。

BMS ルーティング・アプリケーションでは、オペレーターまたはオペレーター・クラスを指定したルーティング要求によって、トランザクションが実行されているシステムの所有する端末にサインオンしたオペレーターにのみ、出力が送られます。

最新の SEND MAP コマンドに指定されたマップ・セット名が、TCTTE に保管されます。ルーティングされるトランザクションの場合、これは、そのマップ・セット名がサロゲート TCTTE に保管されることを意味し、ルーティングされたトランザクションが終了するとき、最も新しく使用されたマップ・セット名が DETACH 順序列を使って AOR から TOR へ渡されることを意味します。

同様に、ルーティングされたトランザクションが開始されるときにも、最も新しく使用されたマップ・セット名が、ATTACH 順序列を使って TOR から AOR へ渡されます。

マップ 名は、マップ・セット 名と同じようにサポートされます。しかし、(サポートされなくなった) 古い CICS プロダクトの中には、マップ名が ATTACH 順序列と DETACH 順序列で渡されることを認識しないものがあります。CICS Transaction Server for z/OS システムは、ATTACH 順序列を送信する際、AOR が DETACH 順序列を使ってマップ名を戻せない場合に備えて、「実際の」TCTTE のマップ名をヌル値に設定します。つまり、TOR の TCTTE には、保管されたマップ名として、誤りの可能性がある名前の代わりにヌル値が入っています。

© Copyright IBM Corp. 1977, 2009

TCTTE に保管されたマップ・セットとマップの名前は両方とも、INQUIRE TERMINAL と SET TERMINAL コマンドの MAPNAME と MAPSETNAME オプションによって照会および更新ができます。これらのオプションの詳細については、「*CICS System Programming Reference*」を参照してください。

疑似会話型トランザクション

ルーティングされたトランザクションでは、その実行中、領域間またはシステム間 (APPC) セッションを使用する必要があります。このため、長期実行会話型トランザクションは、2 つのシステムで重複させてもつか、あるいは疑似会話型トランザクションとして設計するようにしてください。

疑似会話型トランザクションを構成する各トランザクションの命名と定義には注意してください。CICS RETURN コマンドに指定された TRANSID は、端末専有領域に返されるときに、その領域のローカル・トランザクションの可能性があるからです。

ただし、疑似会話型トランザクションをローカル・トランザクションとリモート・トランザクションの両方から構成することは可能です。

端末

トランザクションの実行に使用される「端末」は、端末管理テーブルの端末項目 (TCTTE) によって表されます。

この TCTTE は、**サロゲート TCTTE** と呼ばれ、多くの点で、端末専有領域内にある「実際の」端末の TCTTE のコピーです。 CICS は、トランザクションが終了すると、サロゲート TCTTE を解放します。後続のタスクは、実際の端末の TCTTE の新しいコピーを使用して実行されます。

プログラムから端末に関連する情報を入手する場合には、次の点に注意してください。

- プログラムから TCTTE のフィールドを直接検査しないこと。この代わりに、EXEC インターフェース・ブロック (EIB) 内の同等のフィールドを検査してください。
- 新しいタスクが ATI によって開始される場合、EIB 内の端末に関連するフィールドの中には内容が予測できないものがあります。EIBAID は、アテンション ID を含んでいて、セッションの開始時に、常にゼロに設定されます。

アプリケーション所有領域で EXEC CICS ASSIGN コマンドによって返された値を確認する

EXEC CICS ASSIGN コマンドを使用する場合は、PRINSYSID と USERID の各オプションによって返された値を確認します。これは、値が複数のソースから取得されるためです。

PRINSYSID
このオプションは、トランザクションの基本機能のシステム ID (SYSID) を返します。返される値は、このシステムに定義されたリモート接続または端末の名

前です。接続または端末がシップされている場合、その名前は、端末専有領域 (TOR) に定義された元の名前になります。基本機能が APPC セッションでない場合は、INVREQ 条件が出されます。

USERID

取引ルーティングされたトランザクションでは、CICS はセキュリティー要件の指定に従って、いくつかのソースの 1 つからユーザー ID を取得します。詳細については、「*CICS RACF Security Guide*」のLU6.2 でのトランザクション・ルーティング・セキュリティーを参照してください。

表 16 では、USERID オプションによって返される値について説明しています。次のような値があります。

- 接続が ATTACHSEC(LOCAL) オプションによって定義され、SEC=YES または MIGRATE がアプリケーション所有領域 (AOR) のシステム初期設定パラメーターに指定されている場合、CICS は接続タイプに応じて次のような異なる値を返します。
  - ISC over SNA 接続および IPIC 接続では、返される値は USERID 属性 (この値が SESSIONS 定義で指定されている場合)、または CONNECTION 定義で指定されている SECURITYNAME 属性の値です。
  - MRO 接続では、TOR の RACF ユーザー ID が返されます。
- 接続が ATTACHSEC(LOCAL) オプションによって定義され、SEC=NO が AOR のシステム初期設定パラメーターに指定されている場合、CICS は AOR の DFLTUSER 値を返します。
- 接続が ATTACHSEC(IDENTIFY) オプション、または APPC 接続の場合は VERIFY、PERSISTENT、または MIXIDPE オプションによって定義され、SEC=YES または MIGRATE が TOR のシステム初期設定パラメーターに指定されている場合、CICS は、接続時に送信されたユーザー ID を返します。
- 接続が ATTACHSEC(IDENTIFY) オプション、または APPC 接続の場合は VERIFY、PERSISTENT、または MIXIDPE オプションによって定義され、SEC=NO が TOR のシステム初期設定パラメーターに指定されている場合、CICS は、TOR の DFLTUSER 値を返します。

表 16. ルーティングされたトランザクションに対し EXEC CICS ASSIGN の USERID オプションによって返される値

<table>
<tr><th rowspan="3">TOR のシステム初期設定パラメーター SEC = 値</th><th colspan="3">CONNECTION 定義の ATTACHSEC 値</th></tr>
<tr><th rowspan="2">IDENTIFY
VERIFY
PERSISTENT
MIXIDPE</th><th colspan="2">LOCAL</th></tr>
<tr><th>AOR のシステム初期設定パラメーター SEC=YES または MIGRATE 値</th><th>AOR のシステム初期設定パラメーター SEC=NO 値</th></tr>
<tr><td>YES
または
MIGRATE</td><td>接続時に送られたユーザー ID</td><td rowspan="2">ISC over SNA および IPIC:
1. セッションの USERID
2. 接続の SECURITYNAME
MRO: TOR の RACF ユーザー ID</td><td rowspan="2">AOR の DFLTUSER</td></tr>
<tr><td>NO</td><td>接続時に送られたユーザー ID (TOR の DFLTUSER)</td></tr>
</table>

第 23 章 CICS-IMS 間アプリケーション

この章では、IMS システムと通信する CICS トランザクションのコーディング方法を説明します。

IMS ISC の詳細については、該当する IMS の資料を参照してください。この章は、IMS に関する十分な情報を提供して、お客様が IMS で作業できるようにし、CICS-IMS 間 ISC アプリケーションを実装できるようにするものです。

章には、以下のトピックが含まれています。

- 『CICS-IMS 間 ISC アプリケーションの設計』
- 285 ページの『CICS-IMS 間アプリケーション - 非同期処理』
- 291 ページの『CICS-IMS 間アプリケーション - DTP』.

CICS-IMS 間 ISC アプリケーションの設計

CICS と IMS との間には、そのアーキテクチャーにも、アプリケーションおよびシステムのプログラミング要件にも、多くの違いがあります。

CICS-IMS 間 ISC アプリケーションの設計には、基本的には、CICS アプリケーション・プログラミングと IMS システム定義が関与します。この違いは、2 つの各システムにおける制御の所在を反映するものです。

CICS は、**直接制御**システムです。端末で入力されたデータによって、CICS は適切なアプリケーション・プログラムを呼び出し、着信データを処理します。データは、キューイングされるのではなく保管され、アプリケーションは、その処理を完了して終了するまで、端末を「所有」します。CICS ISC では、アプリケーション・プログラムに、データ・フロー制御プロトコル、同期点処理、および通常は、ほとんどのシステム・サービスが関与します。

これに対して、IMS は**キュー化**システムです。すべての入力メッセージと出力メッセージは、関連アプリケーション・プログラムと端末の代わりに、IMS 制御領域によってキュー化されます。したがって、メッセージの・キュー化とメッセージの処理は、非同期的に実行されます。これは、 284 ページの図 82 に示されています。

このタイプのシステム設計の結果、IMS アプリケーション・プログラムは IMS システム・リソースを直接制御することはなく、システム間通信の制御に直接関与することもありません。IMS メッセージ交換は、完全に IMS 制御領域内で処理されます。メッセージ処理領域は関与しません。

データ形式

CICS と IMS の間で伝送されるメッセージでは、次のいずれかのデータ形式を使用することができます。

- 可変長可変ブロック化 (VLVB)
- RU のチェーン

© Copyright IBM Corp. 1977, 2009

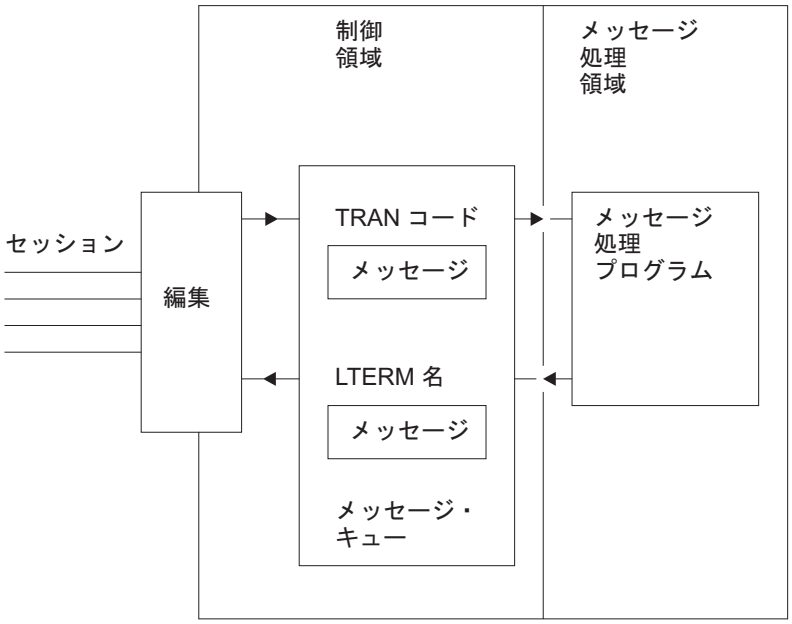


図 82. 基本的な IMS メッセージ・キュー化

論理装置との通常の CICS 通信では、RU のチェーンがデフォルト・データ形式として使用されます。IMS では、VLVB がデフォルトです。CICS-IMS 間通信では、使用される形式は、初期データとともに送られた LUTYPE6.1 付加ヘッダーに指定されます。

可変長可変ブロック化

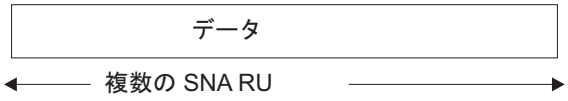
VLVB 形式では、メッセージに複数のレコードを含めることができます。

各レコードの前には、次に示すように、2 バイトの長さフィールドが付きます。

CICS では、入出力域に、1 つまたは複数のレコードを含むことができる完全なメッセージが含まれます。出力レコードのブロック化と入力での非ブロック化は、各自の CICS アプリケーション・プログラムで実行する必要があります。

RU のチェーン

最も一般的な CICS 形式であるこの形式では、次に示すように、メッセージは複数の SNA RU として伝送されます。



CICS では、入出力域に完全なメッセージが含まれます。

IMS とのシステム間通信の形式

CICS と IMS の間の通信が関与する特定のアプリケーションでは、2 つのシステムのいずれかがシステム間通信を開始しなければなりません。例えば、CICS 端末オペレーターが、リモート IMS システムからデータを獲得するように設計された CICS トランザクションを開始すると、このアプリケーション目的のためのシステム間通信が、CICS によって開始されます。

考慮しなければならない次の 3 つの CICS-IMS 間通信形式があります。

1. CICS START と RETRIEVE コマンドを使用する非同期処理
2. CICS SEND LAST と RECEIVE コマンドを使用する非同期処理
3. CICS SEND と RECEIVE コマンドを使用する分散トランザクション処理 (つまり、同期処理)

これらの通信形式の基本的な違いは、51 ページの『第 5 章 非同期処理』と 117 ページの『第 9 章 分散トランザクション処理』に説明してあります。

いずれかの特定のアプリケーションでシステム間通信を開始するシステムは、そのアプリケーションに関する限り、フロントエンド・システムになります。もう一方のシステムは、バックエンド・システムと呼ばれます。

CICS がフロントエンドの場合は、上記にリストしたシステム間通信の 3 つのタイプがすべてサポートされます。個々のアプリケーションに使用できる通信の形式は、開始される IMS トランザクション・タイプ、または IMS 機能によって決まります。IMS がバックエンド・システムの場合にサポートする通信形式については、「*IMS Programming Guide for Remote SNA Systems*」を参照してください。

IMS がフロントエンド・システムの場合は、CICS との通信を開始するために、常に非同期処理 (CICS START と RETRIEVE インターフェースに対応) が使用されます。

CICS-IMS 間アプリケーション - 非同期処理

非同期処理では、システム間セッションは、一方のシステムからもう一方のシステムに、各種のデータ項目とともに開始要求を渡すためにのみ使用されます。それ以外の処理はすべて、要求を渡すために使用されるセッションとは無関係です。

CICS において、非同期処理に使用できる 2 つのアプリケーション・プログラミング・インターフェースは次のとおりです。

1. START と RETRIEVE インターフェース
2. SEND と RECEIVE インターフェース

START と RETRIEVE インターフェース

この項では、これらのコマンドの適切な形式について、CICS-IMS システム間通信環境でのコマンド・オプションの意味とともに説明します。

CICS START と RETRIEVE の「インターバル制御機能」コマンドのプログラミング情報については、「*CICS アプリケーション・プログラミング・リファレンス*」のを参照してください。

CICS フロントエンド

CICS がフロントエンド・システムの場合は、CICS START コマンドおよび RETRIEVE コマンドを使用して、IMS 非応答モード・トランザクションと非会話型トランザクション、メッセージ交換、および IMS の /DIS、/RDIS、/FOR の各オペレーター・コマンドを処理することができます。

注: 上のオペレーター・コマンドを出した場合、方向転換 (CD) を送信しない限り、IMS は確定応答が要求されたものと見なします。このためには、START コマンドに PROTECT オプションを指定しなければなりません。

アプリケーション・プログラムでの一般的なコマンド順序は、図 83 に示すとおりです。

トランザクション TRANA は、端末からの入力メッセージを獲得すると、START NOCHECK コマンドを出して、リモート IMS トランザクションを開始します。この START コマンドは、メッセージを処理するために開始される IMS エディターの名前と、メッセージを受け取る IMS トランザクションまたは論理端末 (LTERM) を指定します。これは、応答を受信する CICS トランザクションの名前と、関連する CICS 端末の名前も指定します。

PROTECT オプションを START コマンドに指定すると、IMS へのメッセージの送達を確実に行うことができます。

開始要求は、アプリケーション・プログラムが SYNCPOINT コマンドを出すか、または終了するまでシップされません。ただし、この要求は、PROTECT が START コマンドに指定されていない限り、同期点標識を伝送しません。

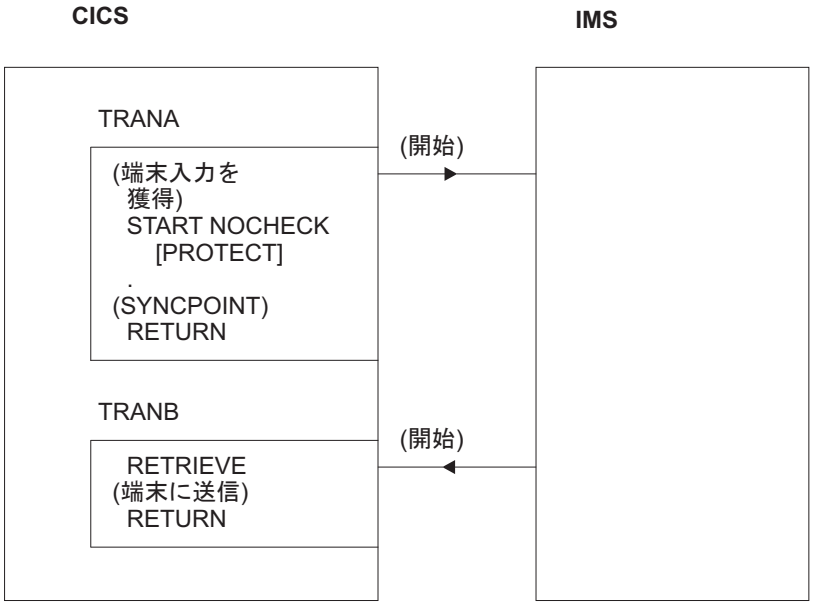


図 83. START と RETRIEVE の非同期処理 - CICS フロントエンド

CICS ではアプリケーション・プログラムが、同期点を間に入れなくても複数の START NOCHECK コマンドを出せますが (57 ページの『NOCHECK オプションを指定した START 要求の ISC リンクへの据え置き伝送』を参照)、この技法は CICS-IMS 間通信には適していません。

IMS は、CICS ミラー・トランザクションによって通常の方法で処理される開始要求を出すことによって、応答を送信します。この要求は、元の START コマンドに指定された CICS トランザクションと端末を指定します。開始されたトランザクション (TRANB) は、RETRIEVE コマンドを出すことによって、応答を取得することができます。

上記の例では、2 つの独立した CICS トランザクションがあるものと想定されています。1 つは START コマンドを出し、もう 1 つは応答を受け取って、それを端末に返します。これらの 2 つのトランザクションは組み合わせることができますが、これには以下で説明する 2 つの方法があります。

- 最初の方法は、START および RETRIEVE の両方の処理を含むが、ある特定の実行ではこれらの機能のうち一方だけを実行するトランザクションを作成するというものです。CICS ASSIGN STARTCODE コマンドを使用すれば、トランザクションが端末から開始されたのか (この場合 START 処理が必要)、開始要求によって開始されたのか (この場合 RETRIEVE 処理が必要) を判別することができます。
- 2 番目の方法は、START コマンドを出した後で SYNCPOINT コマンドを出してその開始要求をクリアし、RETRIEVE コマンドに WAIT オプションを指定して応答を待つトランザクションを作成するというものです。端末はこの間、トランザクションによって保持され、CICS は、同じトランザクションおよび端末に送られた入力が受信されると、そのトランザクションに制御を返します。

いずれの場合も、応答のタイミング、または応答とその前に出された特定の要求との関係について想定することはできません。RETRIEVE コマンドは、同じトランザクションおよび端末に向けられている未解決のデータをすべて検索します。要求と応答は、各自のアプリケーション・プログラムで対応付ける必要があります。

IMS フロントエンド

IMS がフロントエンド・システムの場合、サポートされているフローは非同期開始要求のみです。アプリケーション・プログラムは、RETRIEVE コマンドを使用して IMS からの要求を獲得し、続けて START コマンドを使用して、必要であれば応答を送信しなければなりません。

アプリケーション・プログラムでの一般的なコマンド順序は、288 ページの図 84 に示すとおりです。

検索されたデータに対する応答が必要な場合は、開始コマンドに RETRIEVE コマンドによって獲得された IMS エディターとトランザクションまたは LTERM 名を指定する必要があります。

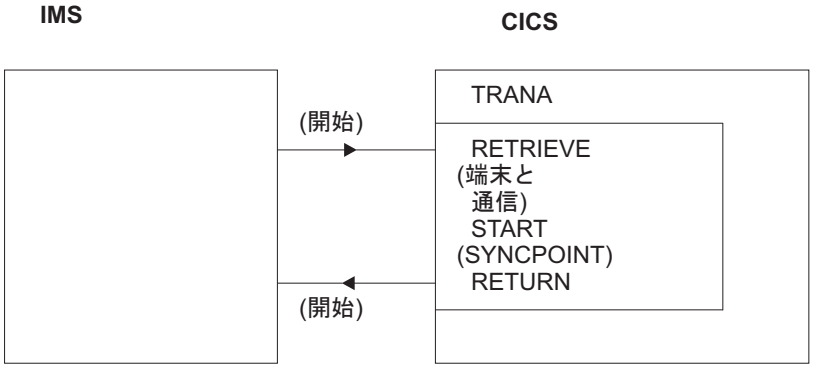


図 84. RETRIEVE と START の非同期処理 - IMS フロントエンド

START コマンド

このセクションでは、リモート IMS トランザクションをスケジュールに入れるために使用される START コマンドの形式を示します。インターバル制御は不可能であり (INTERVAL(0) を指定してもエラーにはなりませんが)、NOCHECK と PROTECT オプションを指定する必要があることに注意してください。

```
EXEC CICS START TRANSID(name)
     [SYSID(name)]
     [FROM(data-area) LENGTH(value)]
     [TERMID(name)]
     [RTRANSID(name)]
     [RTERMID(name)]
      NOCHECK
      PROTECT
     [FMH]
```

TRANSID(name)

メッセージ処理のために開始される IMS エディターの名前を指定します。これは、ISCEDT の別名 (4 文字以下) または MFS MID 名でなければなりません。

あるいは、「リモート」トランザクションのインストール済み定義を指定することができます。この場合、SYSID オプションは使用されません。リモート・トランザクションの定義では、RMTNAME オプションに必要な IMS エディターを指定する必要があります。これは、最大 8 文字の長さにすることができます。

SYSID(name)

リモート IMS システムの名前を指定します。これは、システム・プログラマーによって、リモート・システムへのリンクを定義する DEFINE CONNECTION コマンドの CONNECTION オプションに指定された名前です。このオプションは、リモート・システムを明示的に指定しなければならない場合にのみ必要です。

FROM(data-area)

送信されるデータを指定します。データ形式 (VLVB または RU のチェーン) は、リモート IMS システムを定義する DEFINE CONNECTION コマンドの RECORDFORMAT オプションに指定された形式に一致しなければなりません (165 ページの『第 13 章 リモート・システムへの接続の定義方法』を参照)。

LENGTH(value)
FROM オプションに指定されたデータの長さを、ハーフワードの 2 進値として指定します。

TERMID(name)
リモート・プロセスに割り当てられる 1 次リソース名を指定します。IMS の場合、これはトランザクション・コードまたは LTERM 名になります。

このオプションを省略する場合は、FROM オプションに指定されたデータの最初の 8 文字に、トランザクション・コードまたは LTERM 名を指定する必要があります。名前が 4 文字 (TERMID オプションについての CICS の限界) を超えるか、あるいは IMS パスワード処理が必要な場合は、この方式を使用する必要があります。

RTRANSID(name)
IMS が CICS に応答を戻すときに呼び出されるトランザクションの名前を指定します。この名前は、長さ 4 文字以下でなければなりません。

RTERMID(name)
呼び出し時に、RTRANSID オプションに指定されたトランザクションが呼び出されるときに接続する端末の名前を指定します。この名前は、長さ 4 文字以下でなければなりません。

NOCHECK
このオプションは必須です。

PROTECT
ローカル CICS トランザクションが同期点をとるまで、リモート IMS トランザクションがスケジューリングされないように指定します。PROTECT は必須です。

FMH
開始されたタスクに渡すユーザー・データに機能管理ヘッダーを入れるように指定します。このオプションは通常使用されません。

RETRIEVE コマンド

このセクションでは、IMS によって送られるデータの検索に使用される RETRIEVE コマンドの形式を示します。

```
EXEC CICS RETRIEVE
     [{INTO(data-area)|SET(pointer-ref)}
       LENGTH(data-area)]
     [RTRANSID(data-area)]
     [RTERMID(data-area)]
     [WAIT]
```

INTO(data-area)
IMS から検索されたデータが書き込まれるユーザー・データ域を指定します。

SET(pointer-ref)
ポインター参照が、IMS から検索されるデータのアドレスに設定されるように指定します。

LENGTH(data-area)
検索されたデータのハーフワード 2 進数に長さを指定します。

INTO オプションを付けた RETRIEVE コマンドの場合、これは、プログラムが処理できるデータの最大長を指定するデータ域でなければなりません。指定された値がゼロ未満の場合は、ゼロが想定されます。データの長さが指定値を超えると、データはその値に切り捨てられて、LENGERR 条件が起こります。検索操作が完了すると、データ域はデータの元の長さに設定されます。

SET オプションを指定した RETRIEVE コマンドの場合、これはデータ域でなければなりません。検索操作が完了すると、データ域はデータの長さに設定されます。

RTRANSID(data-area)
IMS によって送られる戻り宛先プロセス名を受け取る区域を指定します。これは、出力 MOD からチェーニングする MFS MID 名かブランクのいずれかになります。

アプリケーションは、後の START コマンドの TRANSID オプションにこの名前を使用することができます。

RTERMID(data-area)
IMS によって送られる戻り 1 次リソース名を受け取る区域を指定します。これは、トランザクション名または LTERM 名のいずれかです。

アプリケーションは、応答の送信に使用される START コマンドの TERMID オプションにこの名前を使用することができます。

WAIT
データが IMS によって送られるまで、制御がアプリケーション・プログラムに返されないように指定します。

WAIT を指定しないと、使用可能なデータがない場合、ENDDATA 条件が起こります。WAIT を指定すると、データが使用可能になる前に CICS がシャットダウンされる場合にのみ、ENDDATA 条件が起こります。

WAIT オプションを使用すると、その間に発生したメッセージ (予期した応答ではない) が検索されることになるため、通常は使用しないでください。

非同期の SEND と RECEIVE インターフェース

CICS の場合、この形式の非同期処理は、分散トランザクション処理の特殊なケースです。

CICS トランザクションは、リモート・システムへのセッションの使用を獲得し、そのセッションを単一の伝送 (LAST オプションを付けた SEND コマンドを使用する) に使用して、リモート・トランザクションを開始し、それにデータを送信します。リモート・システムからの応答によって、CICS トランザクションは、通常の DTP におけるバックエンド・トランザクションと同じように開始されます。ただし、このトランザクションは、1 回の RECEIVE コマンドしか出せないため、その後でセッションを解放しなければなりません。

これらの追加の制約事項を除けば、この章の後半で説明する分散トランザクション処理の規則に従って、アプリケーションを設計することができます。

非同期の SEND と RECEIVE アプリケーション・プログラムでの一般的なコマンド順序は、291 ページの図 85 に示すとおりです。

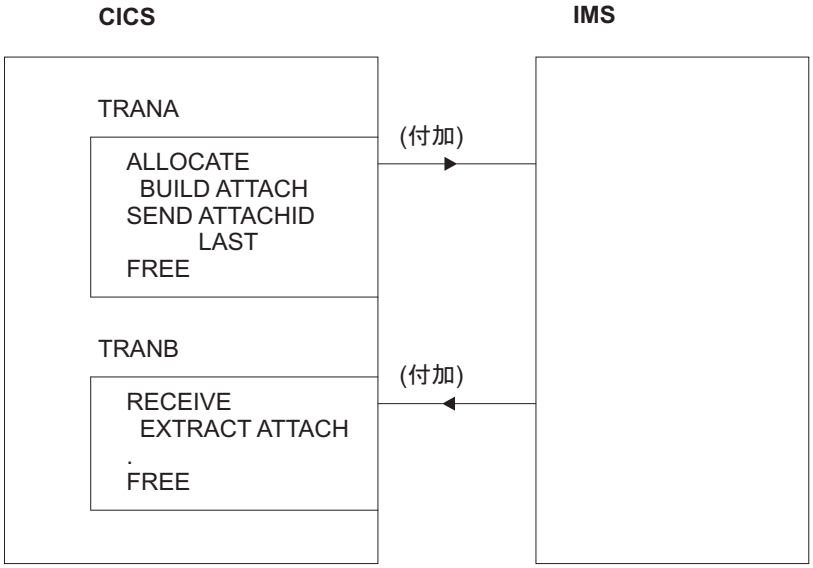


図 85. SEND と RECEIVE の非同期処理 - CICS フロントエンド

CICS-IMS 間アプリケーション - DTP

このセクションでは、CICS-IMS 間分散トランザクション処理 (DTP) のためのアプリケーション・プログラミングについて説明します。

DTP の詳細については、「*CICS Distributed Transaction Programming Guide*」を参照してください。

CICS-IMS 間セッションに対する CICS コマンド

次のコマンドは、CICS-IMS 間セッションを獲得して使用する際に使用できます。

- **ALLOCATE** - リモート IMS システムへのセッションを獲得する際に使用されます。
- **BUILD ATTACH** - リモート IMS システムでトランザクションを開始するための LUTYPE6.1 付加ヘッダーを作成するために使用されます。
- **EXTRACT ATTACH** - CICS トランザクションで、このトランザクションを開始させる原因となった LUTYPE6.1 付加ヘッダーから情報をリカバリーする際に使用されます。このコマンドは、SEND/RECEIVE 非同期処理にのみ必要とされます。
- **SEND**、**RECEIVE**、および **CONVERSE** - CICS トランザクションによって、セッションでデータを送受信する際に使用されます。フロントエンド CICS トランザクションによって出される最初の SEND コマンドまたは CONVERSE コマンドは、BUILD ATTACH コマンドによって定義された付加ヘッダーを指定しなければなりません。
- **WAIT TERMINAL SESSION(name)** - CICS が、処理をさらに続ける前に、累積されたデータまたはデータ・フロー制御標識を確実に伝送するために使用されます。

- **ISSUE SIGNAL SESSION(name)** - 受信状態にあるトランザクションによって、IMS からの送信勧誘 (方向変換) を要求するために使用されます。
- **FREE** - CICS トランザクションによって、そのセッション使用を解放するために使用されます。

フロントエンド・トランザクションに関する考慮事項

SEND および RECEIVE の非同期処理における受信側トランザクションという特殊な場合を除いて、CICS トランザクションは、CICS-IMS 間 DTP では常にフロントエンド・トランザクションになります。

フロントエンド・トランザクションは、リモート IMS システムとのセッションを獲得して、リモート・トランザクションを開始します。その後、2 つのトランザクションは等しくなります。ただし、フロントエンド・トランザクションは、通常はクライアント、つまり駆動側トランザクションとして設計されています。

セッション割り振り

リモート IMS システムとの LUTYPE6.1 セッションは、次の形式の ALLOCATE コマンドを使用して獲得します。

```
ALLOCATE {SYSID(name)|SESSION(name)}
  [PROFILE(name)]
  [NOQUEUE]
```

SESSION オプションを使用すると、リモート IMS システムとの特定セッションの使用を要求することができます。あるいは、SYSID オプションを使用してリモート・システムを指定し、CICS が使用可能なセッションを選択できるようにすることもできます。SESSION オプションを使用すると、他のセッションを使用できるのに、アプリケーション・プログラムが特定のセッション待ちになる可能性があるため、通常このオプションは勧められません。したがって、ほとんどの場合、SYSID オプションを使用して、セッション相手として必要なシステムを指定します。

CICS では、指定されたシステムが見つからないか、セッションが使用できないと、SYSIDERR 条件が起こります。また、CICS では、指定されたセッションが見つからないか、セッションがサービス不能になっていると、SESSIONERR 条件が起こります。

PROFILE オプションを使用すると、LUTYPE6.1 セッションの通信プロファイルを指定することができます。リソース定義中に設定されるプロファイルには、セッションに使用される、端末制御処理オプションの集合が含まれます。

PROFILE オプションを省略すると、CICS は、デフォルト・プロファイルの DFHCICSA を使用します。このプロファイルは、INBFMH(ALL) を指定します。これは、着信機能管理ヘッダーが各自のプログラムに渡され、INBFMH 条件が起こることを示します。

NOQUEUE オプションを使用すると、セッションをすぐに使用できないときに、セッションに対する要求をキューに入れないように、明示的に指定することができます。セッションは、次のいずれかの場合、「すぐには使用できない」状態になります。

- 指定のシステムとのセッションすべてが使用中である。

- 使用可能なセッションだけがバインドされていない (この場合、CICS はセッションをバインドする必要があります)。
- 使用可能なセッションだけが競合敗者である (この場合、CICS はブラケットを開始するように要求する必要があります)。

セッションがすぐに使用できない場合に CICS がとるアクションは、NOQUEUE を指定したかどうか、およびアプリケーションが SYSBUSY 条件に対して HANDLE コマンド (依然としてアクティブにある) を出したかどうかによっても異なります。次の組み合わせが考えられます。

- SYSBUSY 条件に対する HANDLE がアクティブ
  - 制御は、NOQUEUE を指定したかどうかに関係なく、HANDLE コマンドに指定されたラベルにただちに返されます。
- SYSBUSY 条件に対する HANDLE がアクティブでない
  - NOQUEUE を指定した場合、制御はただちにアプリケーション・プログラムに返されます。EXEC インターフェース・ブロックの EIBRCODE フィールドに、SYSBUSY コード (X'D3') が設定されます。ALLOCATE コマンドを出したらすぐに、このフィールドをテストする必要があります。
  - NOQUEUE オプションを省略した場合、CICS は、セッションが使用可能になるまで要求をキュー化します。

セッション獲得における遅延が受け入れられるかどうかは、アプリケーションによって異なります。

SYSID ではなく SESSION を指定する ALLOCATE コマンドにも、同様の考慮事項が当てはまります。関連する条件は「SESSBUSY」(EIBRCODE=X'D2') です。

セッション ID

セッションが割り振られると、その名前が EIB の EIBRSRCE フィールドに入ります。

EIBRSRCE は、次の EXEC CICS コマンドによって上書きされる可能性が高いため、すぐにセッション名を入手する必要があります。これは、このセッションに関連するすべての後続コマンドの SESSION パラメーターに使用しなければならない名前です。

自動トランザクション開始

フロントエンド・トランザクションが、ローカル・システムの自動トランザクション開始 (ATI) によって開始されるように設計されていて、その基本機能として LUTYPE6.1 セッションで会話を保持する必要がある場合、セッションは、トランザクションの開始時に既に割り振られています。

この基本機能に関連するコマンドからは SESSION パラメーターを省略することができます。ただし、これらのコマンドにセッションを明示的に指定したい場合は、EIBTRMID から名前を入手する必要があります。

リモート・トランザクションの接続

セッションを獲得すると、次のステップとしてリモート IMS プロセスが開始されます。

LUTYPE6.1 アーキテクチャーは、付加ヘッダーと呼ばれる特殊な機能管理ヘッダーを定義しています。これは、開始されるリモート・プロセス (CICS ではトランザクションといいます) の名前を示し、さらに詳しいセッション関連の情報を含みます。

CICS の BUILD ATTACH コマンドを使用すると、CICS アプリケーション・プログラムで付加ヘッダーを作成して IMS に送信することができます。また、EXTRACT ATTACH コマンドを使用すると、IMS から受け取った付加ヘッダーから情報を入手することができます。

これらのコマンドを使用することができるため、LUTYPE6.1 付加ヘッダーの詳細な形式を知る必要はありません。ただし、ほとんどの場合、これによって伝えられる情報を理解する必要があります。

BUILD ATTACH コマンドの形式は次のとおりです。

```
BUILD ATTACH
   ATTACHID(name)
  [PROCESS(ISCEDT|BASICEDTvname)]
  [RESOURCE(name)]
  [RPROCESS(name)]
  [RRESOURCE(name)]
  [QUEUE(name)]
  [IUTYPE(0|data-value)]
  [DATASTR(0|data-value)]
  [RECFM(data-value)]
```

BUILD ATTACH コマンドのパラメーターの意味は、次のとおりです。

ATTACHID(name)
ATTACHID オプションを使用すると、付加ヘッダーに名前を割り当てて、後続の SEND または CONVERSE コマンドでそれを参照することができます。(BUILD ATTACH コマンドは、付加ヘッダーを作成するだけで、伝送はしません。)

PROCESS(name)
これは、付加 FMH 内の プロセス名 ATTDPN に対応します。開始するリモート・プロセスを指定します。

CICS-IMS 間通信では、リモート・プロセスは常にエディターになります。これは、ISCEDT (またはその別名)、BASICEDT、または MFS MID 名のいずれかにすることができます。プロセス名は 8 文字以下でなければなりません。

PROCESS オプションを省略すると、IMS は ISCEDT を想定します。

RESOURCE(name)
これは、付加 FMH 内のリソース名 ATTPRN に対応します。

RESOURCE オプションでは、開始されるリモート・プロセスに割り当てられる 1 次リソース名 (8 文字まで) を指定します。

CICS-IMS 間通信では、1 次リソース名は、IMS トランザクション・コードまたは論理端末名のいずれかになります。IMS メッセージ宛先がメッセージの最初の 8 バイトに指定されている場合、あるいは宛先が IMS オペレーターによって事前設定されている場合は、RESOURCE オプションを省略することができます。

1 次リソース名が IMS に指定されている場合、データ・ストリームの宛先とセキュリティー情報は編集されません。したがって、IMS パスワード処理が必要な場合は、RESOURCE オプションを省略する必要があります。

会話型処理中、またはリモート・プロセスが BASICEDT の場合、RESOURCE オプションは無視されます。

RPROCESS(name)
これは、付加 FMH 内の戻りプロセス名 ATTRDPN に対応します。

RPROCESS オプションは、指示された戻り宛先プロセス名を指定します。IMS は、CICS に応答を送るときに、宛先プロセス名 (ATTDPN) としてこの名前を返します。ただし、この名前は、MFS によって上書きされる可能性があります。

CICS は、返された宛先プロセス名を使用して、セッション再始動後に接続されるトランザクションを判別します。それ以外の場合、これは無視されます。したがって、RPROCESS オプションには、セッション障害後のセッション再始動時に CICS によって接続されたときにすべてのキュー化されたメッセージを処理するトランザクションを指定する必要があります。

RRESOURCE(name)
これは、付加 FMH 内の戻りリソース名 ATTRPRN に対応します。

RRESOURCE オプションは、戻りプロセスに割り当てられる指示された 1 次リソース名を指定します。IMS は、CICS に応答を送信するときに、この名前をリソース名 (ATTPRN) として返します。

CICS は、通常このフィールドを無視しますが、ISC でこれを使用すると、セッション再始動後に起こった出力メッセージの送信先となる CICS 端末を指定することができます。

QUEUE(name)
これは、付加 FMH 内のキュー名 ATTDQN に対応します。

QUEUE オプションでは、リモート・プロセスに対応付けることができるキューを指定します。CICS-IMS 間通信において、これは、要求時ページング中にページング要求を IMS に送るためにのみ使用されます。使用される名前は、前の EXTRACT ATTACH QNAME コマンドによって獲得された名前でなければなりません。この名前は、長さ 8 文字以下でなければなりません。

IUTYPE(data-value)
これは、付加 FMH 内の交換単位フィールド ATTIU に対応します。

IUTYPE オプションは、メッセージの SNA チェーン情報を指定します。この値はハーフワード 2 進値です。2 進値内のビットは、次のように使用されます。

0 ～ 7	X'00' - ゼロでなければならない
8 ～ 15	X'00' - 複数の RU チェーン
	X'01' - 単一の RU チェーン

DATASTR(data-value)
これは、付加 FMH 内のデータ・ストリーム・プロファイル・フィールドの ATTDSP に対応します。

DATASTR オプションは、IMS コンポーネントを選択するために使用されます。この値はハーフワード 2 進値です。2 進値内のビットは、次のように使用されます。

0 〜 7	X'00' - ゼロでなければならない
8 〜 11	0000 - (ユーザー定義のデータ・ストリーム)
12 〜 15	0000 - IMS コンポーネント 1 0001 - IMS コンポーネント 2 0010 - IMS コンポーネント 3 0011 - IMS コンポーネント 4

DATASTR オプションを省略すると、IMS コンポーネント 1 が想定されます。

RECFM(data-value)
これは、付加 FMH 内の非ブロック化アルゴリズム・フィールド ATTDBA に対応します。

RECFM オプションでは、リモート・プロセスに送信するユーザー・データの形式を指定します。この名前は、ハーフワード 2 進値を表さなければなりません。2 進値内のビットは、次のように使用されます。

0 〜 7	X'00' - 予約済み - ゼロでなければならない
8 〜 15	X'01' - 可変長可変ブロック化 (VLVB) 形式 X'04' - RU のチェーン

VLVB を指定する場合、アプリケーション・プログラムは、各レコードの前に、2 バイト 2 進数の長さフィールドを追加する必要があります。RU のチェーンを指定すると、データを通常の方法で送信することができます。長さフィールドは不要です。

レコードは、メッセージ・セグメント (MFS なし) または MFS レコード (MFS 付き) のいずれかとして IMS によって解釈されます。

RECFM オプションは、メッセージ形式のタイプだけを示します。複数のレコードを 1 つの SEND コマンドで送信することができます。この場合、アプリケーション・プログラムがブロック化を実行する必要があります。

付加ヘッダーを作成したら、SEND コマンドまたは CONVERSE コマンドの ATTACHID オプションにそれを指定して、リモート・システムに送る最初のデータとともに、その付加ヘッダーを伝送する必要があります。

独自の付加ヘッダーの作成

CICS では、出力データの一部として、付加ヘッダー、つまり機能管理ヘッダーを作成することができます。

したがって、最初の SEND コマンドまたは CONVERSE コマンドによって参照される出力域に LUTYPE6.1 付加ヘッダーを含めることによって、リモート・トランザクションを開始することができます。コマンドの FMH オプションを指定して、CICS に対し、データに FMH が含まれていることを知らせる必要があります。

バックエンド・トランザクションに関する考慮事項

CICS トランザクションは、CICS-IMS 間通信において、特殊な SEND/RECEIVE 非同期処理の場合にのみバックエンド・トランザクションにすることができます。

トランザクションは、リモート IMS システムから受け取った LUTYPE6.1 付加 FMH によって開始され、RECEIVE コマンドを 1 回だけ出すことができます。また、それに続けて、EXTRACT ATTACH コマンドを出すことができます。

セッション関連情報の獲得

EXTRACT ATTACH コマンドを使用すると、必要に応じて付加 FMH からセッション関連情報を回復することができますが、このコマンドを使用しなければならないというわけではありません。

付加ヘッダーの存在は、EIBATT によって示されます。これは、最初の RECEIVE コマンドが出された後で設定されます。

EXTRACT ATTACH コマンドの形式は次のとおりです。

```
EXTRACT ATTACH
  [SESSION(data-area)]
  [PROCESS(data-area)]
  [RESOURCE(data-area)]
  [RPROCESS(data-area)]
  [RRESOURCE(data-area)]
  [QUEUE(data-area)]
  [IUTYPE(data-area)]
  [DATASTR(data-area)]
  [RECFM(data-area)]
```

EXTRACT ATTACH コマンドのパラメーターの意味は、次のとおりです。

DATASTR(data-area)

IMS 出力コンポーネントを指定する値が入ります。

データ域は、ハーフワード 2 進数フィールドでなければなりません。この値は、IMS によって次のように設定されます。

0 ～ 7	X'00'– (ゼロ)
8 ～ 11	0000 - (ユーザー定義のデータ・ストリーム)
12 ～ 15	0000 - IMS コンポーネント 1
	0001 - IMS コンポーネント 2
	0010 - IMS コンポーネント 3
	0011 - IMS コンポーネント 4

IUTYPE(data-area)

メッセージの SNA チェーン情報と MFS ページ出力のタイプを示します。

データ域は、ハーフワード 2 進数フィールドでなければなりません。この値は、IMS によって次のように設定されます。

0 ～ 7	X'00' – (ゼロ)
8 ～ 15	X'00' - 複数の RU チェーン、MFS 自動ページ出力
	X'01' - 単一の RU チェーン、MFS 非ページ出力
	X'05' - 単一の RU チェーン、MFS 要求時ページ出力

PROCESS(data-area)
IMS は、BUILD ATTACH コマンドの RPROCESS オプションに指定された戻り宛先プロセス名、または MFS MOD によって設定された値のいずれかを返します。

QUEUE(data-area)
IMS は、MFS 要求時ページ出力の送信準備ができると、ISC セッションに関連する LTERM 名を戻します。戻された値は、ページング要求が送信されるときに、QMODEL FMH および BUILD ATTACH QNAME で使用する必要があります。

RECFM(data-area)
着信ユーザー・メッセージのデータ形式が入ります。

データ域は、ハーフワード 2 進数フィールドでなければなりません。この値は、IMS によって次のように設定されます。

0 ～ 7	X'00' – (ゼロ)
8 ～ 15	X'01' - 可変長可変ブロック化 (VLVB) 形式 X'04' - RU のチェーン (X'00' または X'05' の場合もある)

VLVB が指定されている場合、アプリケーション・プログラムは、各レコードの前にあるハーフワード 2 進数の長さフィールドを使用して、メッセージを非ブロック化する必要があります。

RESOURCE(data-area)
IMS は、BUILD ATTACH コマンドの RRESOURCE オプションに指定された戻りリソース名、または MFS MOD によって設定された値のいずれかを返します。

RPROCESS(data-area)
IMS は、MFS が使用される場合、チェーニングされた MFS MID 名を送ります。それ以外の場合、値は送られません。

RRESOURCE(data-area)
IMS は、MFS が使用される場合、MFS MOD によって設定された値を送ります。それ以外の場合、値は送られません。

バックエンド・トランザクションの初期状態
バックエンド・トランザクションは、受信状態で開始し、RECEIVE を最初のコマンドとして、あるいは EXTRACT ATTACH の後で出す必要があります。

会話
フロントエンド・トランザクションとバックエンド・トランザクションの間の会話は、通常の SEND、RECEIVE、および CONVERSE の各コマンドによって維持されます。

これらのコマンドのプログラミング情報については、「*CICS* アプリケーション・プログラミング・リファレンス」の SEND (LUTYPE6.1)、RECEIVE (LUTYPE6.1)、および CONVERSE (LUTYPE6.1)を参照してください。

基本機能を使っている会話でなければ、これらの各コマンドの SESSION オプションにはセッションを指定する必要があります。

据え置き伝送

ISC セッションで SEND コマンドを出すと、CICS は、通常、送信側の意図が分かるまでデータの送信を据え置きます。このメカニズムにより、CICS は、伝送待ち状態のデータに制御標識を追加することによって、不要な流れを回避することができます。

一般に、IMS は、方向転換、同期点要求、またはブラケット終了などの標識を、ヌル RU の独立型伝送として受け入れません。したがって、据え置き伝送を常に作動可能な状態にしておくと同時に、伝送を強制的に実行させるための WAIT オプションや WAIT TERMINAL コマンドの使用を避ける必要があります。

LAST オプションの使用

SEND コマンドの LAST オプションは、会話の終わりを示します。セッションでのデータ・フローはそれ以上起こらないため、セッションを解放する必要があります。ただし、セッションは、解放されるまでに、CICS 同期点処理フローを伝送することができます。

LAST オプションと同期点フロー

ISC セッションでの同期点は、SYNCPOINT コマンドによって明示的に、あるいは RETURN コマンドによって暗黙指定で開始されます。

会話が WAIT オプションを指定しない SEND LAST コマンドによって終了した場合、伝送は据え置かれているので、同期点処理アクティビティーによって、同期点要求が追加された形で最終伝送が行われます。したがって、会話は自動的に同期点に含まれます。

セッションの解放

セッションは、SEND LAST コマンドを出した後、または EIBFREE フィールドが設定されたときに解放する必要があります。

セッションの解放に使用されるコマンドの形式は、次のとおりです。

```
FREE SESSION(conversation-name)
```

CICS では、トランザクションが送信状態にある場合、いつでも FREE コマンドを出すことができます。CICS は、ブラケット終了標識が既に伝送されたかどうかを判別し、必要に応じてそれを伝送してから、セッションを解放します。伝送を据え置かれたデータがある場合、ブラケット終了標識はそのデータとともに送られます。それ以外の場合、標識はそれだけで伝送されます。

独立型ブラケット終了標識を受け入れる IMS 入力コンポーネントは一部であるため、FREE の使用は、CICS-IMS 間通信には勧められません。

EXEC インターフェース・ブロック (EIB)

この項では、ISC アプリケーションで特に重要なフィールドを中心に説明します。

EXEC インターフェース・ブロック (EIB) のプログラミング情報については、「*CICS アプリケーション・プログラミング・リファレンス*」の EXEC インターフ

ェース・ブロックを参照してください。 これらのフィールドのテストまたは保管の方法と時期に関する詳細については、 301 ページの『CICS-IMS 間セッションのコマンド順序』を参照してください。

会話 ID フィールド

EIB フィールドである EIBTRMID と EIBRSRCE を使用すると、ISC セッションの名前を獲得することができます。

EIBTRMID
基本機能の名前が入ります。バックエンド・トランザクション、または ATI によって開始されたフロントエンド・トランザクションの場合は、会話 ID (SESSION) になります。基本機能のセッション名を明示的に指定したい場合は、この名前を獲得する必要があります。

EIBRSRCE
ALLOCATE コマンドによって獲得されたセッションのセッション ID (SESSION) が入ります。この名前は、ALLOCATE コマンドを出した直後に獲得する必要があります。

プロシージャー・フィールド

これらのフィールドには、セッションの状態に関する情報が含まれます。ほとんどの場合、設定値は、最後に実行された RECEIVE コマンドまたは CONVERSE コマンドに指定されたセッションに関連します。これは、コマンドが出された直後に、テストするか、または後でテストするために保管する必要があります。

これらのフィールドの使用方法の詳細については、 301 ページの『CICS-IMS 間セッションのコマンド順序』を参照してください。

EIBRECV
会話が受信状態にあり、通常 RECEIVE コマンドが出されることを示します。

EIBCOMPL
このフィールドは、RECEIVE NOTRUNCATE コマンドとの組み合わせで使用されます。これは、使用可能なデータがない場合に設定されます。

EIBSYNC
アプリケーションで同期点をとるか、アプリケーションを終了する必要があることを示します。

EIBSIG
会話パートナーが ISSUE SIGNAL コマンドを出したことを示します。

EIBFREE
受信側が、セッションに対して FREE コマンドを出さなければならないことを示します。

情報フィールド

次のフィールドには、リモート・トランザクションから受信された FMH に関する情報が含まれます。

EIBATT
受信されたデータに付加ヘッダーが含まれることを示します。付加ヘッダーはア

プリケーション・プログラムには渡されません。ただし、EIBATT は、EXTRACT ATTACH コマンドが適切であることを示します。

EIBFMH
アプリケーション・プログラムに渡されたデータに連結 FMH が含まれることを示します。

これらの機能を使用したい場合は、必ず INBFMH(ALL) を指定した通信プロファイルを使用する必要があります。この指定は、CICS フロントエンド・トランザクションによって割り振られたセッションのデフォルト・プロファイル (DFHCICSA) にはありますが、CICS バックエンド・トランザクションのデフォルト基本機能プロファイル (DFHCICST) にはありません。この主題の詳細については、253 ページの『通信プロファイルの定義』を参照してください。

CICS-IMS 間セッションのコマンド順序

フロントエンド・トランザクションとバックエンド・トランザクション間の通信に使用するコマンド順序は、アプリケーションの要件と、コマンドが不適切な環境で発行されないように設計された一連の高水準プロトコルとの両方によって決まります。

この項で示すプロトコルは、考えられるコマンド順序すべてに適用されるものではありません。ただし、これらのプロトコルに従うと、各トランザクションでもう一方のトランザクションの要件を考慮することができます。このことは、プログラム開発中のエラーを回避するうえで役立ちます。

会話状態

これらのプロトコルは、いくつかの個別の状態の概念に基づいています。

これらの状態は、アプリケーション・プログラム全体ではなく、特定の会話にのみ適用されます。各状態では、最も適切なコマンドを選択することができます。コマンドが出されると、EIB 内のフィールドをテストして、会話における現在の要件を知ることができます。これらのテスト結果と出されたコマンドによっては、別の状態に遷移するので、そこでまた別のコマンド集合を出せるようになります。

この項で定義される状態は次のとおりです。

- 状態 1 - セッション未割り振り
- 状態 2 - 送信状態
- 状態 3 - SEND INVITE 後の受信保留状態
- 状態 4 - 受信状態
- 状態 5 - 受信側による同期点
- 状態 6 - SEND LAST 後の解放保留状態
- 状態 7 - セッション解放

初期状態

通常、会話のフロントエンド・トランザクションは、状態 1 (セッション未割り振り) で開始します。そして、ALLOCATE コマンドを出してセッションを獲得しなければなりません。

ただし、ローカル・システムで、フロントエンド・トランザクションがその基本機能として LUTYPE6.1 セッションを使用して自動トランザクション開始 (ATI) によって開始される場合は例外です。この場合は、セッションが既に割り振られていて、トランザクションは状態 2 にあります。このタイプのトランザクションでは、EIBTRMID からすぐにセッション名を獲得して、後のコマンドにそのセッションを明示的に指定する必要があります。

バックエンド・トランザクションは初期設定では状態 4 (受信状態) であると常に想定する必要があります。フロントエンド・トランザクションへデータを送信するようにしか設計されていない場合でも、RECEIVE を出してフロントエンド・トランザクションによって出された SEND INVITE を受け取り、送信状態に入る必要があります。

状態遷移

この後の図を使用すると、有効なコマンド順序を簡単に作成できます。各図は、上記で定義した特定の状態に関連しており、適切なコマンドと、そのコマンドを出した後に行う必要があるテストを示しています。複数のテストが示されている場合は、表示の順序でそれらを実行してください。

出されたコマンドに対し、肯定的な特定のテスト結果が得られれば、最終列に示された新しい状態に移行します。

その他のテスト

表に示されたテストは、会話の状態に重要なものです。発生する可能性のあるその他の条件、例えば INVREQ や NOTALLOC についてのテストは、通常どおりに行ってください。

表 17. 状態 1 - セッション未割り振り

状態 1 - CICS-IMS 問会話 - SESSION NOT ALLOCATED		
発行できるコマンド	**テスト内容**	**新規状態**
ALLOCATE [NOQUEUE] *	SYSIDERR	1
上に同じ	SYSBUSY *	1
上に同じ	その他の場合 (EIBRSRCE からセッション名を獲得)	2

セッションが使用可能になるまでプログラムを待機させたい場合は、NOQUEUE オプションを ALLOCATE コマンドから省略し、SYSBUSY 条件に対して HANDLE コマンドをコーディングしないようにしてください。

セッションがすぐに使用できない場合に、プログラムに制御を戻したい場合は、ALLOCATE コマンドに NOQUEUE を指定して EIBRCODE が SYSBUSY (X'D3') かどうか EIBRCODE をテストするか、あるいは HANDLE CONDITION SYSBUSY コマンドをコーディングしてください。

表 18. 状態 2 - 送信状態

状態 2 - CICS-IMS 問会話 SEND STATE		
発行できるコマンド *	**テスト内容**	**新規状態**
SEND		2

表 18. 状態 2 - 送信状態 (続き)

状態 2 - CICS-IMS 間会話 SEND STATE		
発行できるコマンド *	**テスト内容**	**新規状態**
SEND INVITE	—	3 または 4
SEND LAST	—	6

CONVERSE
次のものと同等
`SEND INVITE WAIT`
`RECEIVE`
	STATE 4 のテーブルを参照し、RECEIVE コマンド用のテストを行う	—

RECEIVE	STATE 4 のテーブルを参照し、RECEIVE コマンド用のテストを行う	—
SYNCPOINT	(SYNCPOINT が失敗した場合、トランザクションは異常終了する)	2

FREE
次のものと同等
`SEND LAST WAIT`
`FREE`
	—	1

フロントエンド・トランザクションの場合、セッションの割り振り後に初めて使用するコマンドは、 293 ページの『リモート・トランザクションの接続』で説明したいずれかの方法によってバックエンド・トランザクションを開始する SEND コマンドか CONVERSE コマンドでなければなりません。

表 19. 状態 3 - SEND INVITE 後の受信保留状態

状態 3 - CICS-IMS 間会話 - SEND INVITE 後の受信保留状態		
発行できるコマンド	**テスト内容**	**新規状態**
SYNCPOINT	(SYNCPOINT が失敗した場合、トランザクションは異常終了する)	4

表 20. 状態 4 - 受信状態

状態 4 - CICS-IMS 間会話 - RECEIVE STATE		
発行できるコマンド	**テスト内容**	**新規状態**
RECEIVE [NOTRUNCATE] *	EIBCOMPL *	—
上に同じ	EIBSYNC	5
上に同じ	EIBFREE	7
上に同じ	EIBRECV	4
上に同じ	その他の場合	2

NOTRUNCATE を指定した場合、EIBCOMPL のゼロ値は、CICS によってアプリケーションに渡されたデータが不完全であることを示します (例えば、RECEIVE コマンドに指定されたデータ域が小さ過ぎるなどの原因のため)。CICS は、後の RECEIVE NOTRUNCATE コマンドで取得できるように、残りのデータを保管します。 EIBCOMPL は、データの最後の部分が返されると設定されます。 NOTRUNCATE オプションを指定しないと、長過ぎるデータは LENGERR 条件によって示され、残りのデータは CICS によって廃棄されます。

表 21. 状態 5 - 受信側による同期点

状態 5 - CICS-IMS 問会話 - RECEIVER TAKE SYNCPOINT		
発行できるコマンド	**テスト内容**	**新規状態**
SYNCPOINT	EIBFREE (保管された値)	7
上に同じ	EIBRECV (保管された値)	4
上に同じ	その他の場合	2

表 22. 状態 6 - SEND LAST 後の解放保留状態

状態 6 - CICS-IMS 問会話 - FREE PENDING AFTER SEND LAST		
発行できるコマンド	**テスト内容**	**新規状態**
SYNCPOINT	—	7
FREE	—	1

表 23. 状態 7 - セッション解放

状態 7 - CICS-IMS 問会話 - FREE SESSION		
発行できるコマンド	**テスト内容**	**新規状態**
FREE	—	1

第 5 部 システム間環境におけるパフォーマンス

このパートでは、複数システム環境における CICS のパフォーマンスを改善するためのヒントを説明します。

307 ページの『第 24 章 システム間のセッション・キューの管理』では、システム間キューの長さを制御する方法について説明します。

311 ページの『第 25 章 シップされた端末定義の効率的な削除』では、重複してシップされた端末定義を AOR と中間システムから削除する方法について説明します。

© Copyright IBM Corp. 1977, 2009

第 24 章 システム間のセッション・キューの管理

この章では、システム間リンクのセッションに対する待機要求の数をいかに制御するか (割り振りキュー) を説明します。

注: この章では、確立された接続のセッションに対するキューをいかに制御するかを説明します。機能シップされた EXEC CICS START NOCHECK 要求のためにローカル・キューを使用する方法については、58 ページの『START コマンドのローカル・キューイング』を参照してください。

セッション・キュー管理の概要

完全な相互通信環境では、キューは起こらないでしょう。この環境では、作業の流れが時間の経過に従って平均に分散され、任意の時点に到着する最大の要求数を処理できるだけのシステム間セッションが使用可能だからです。

しかし、実世界ではこういうわけにはいきません。ワークロードには山と谷がありますので、キューが起こります。キューはワークロードに応じて発生したり、消滅したりします。しかし、相互接続された CICS 領域間で作業の流れを阻害したり、スループットを低下または停止させて端末のエンド・ユーザーに対してパフォーマンスの問題を生じさせたりするような、受け入れがたいほど極端なキューイングは避けなければなりません。このような予期しない異常なキューイングは、起こらないようにするか、起こった場合には対処しなければなりません。「通常の」キューイングまたは最適化された水準のキューイングは許容範囲です。

例えば、CICS アプリケーション所有領域と、接続されたファイル所有領域との間の機能シップ要求は、空きセッションを待っている間、発行元の領域にキューイングすることができます。ファイル所有領域がそれらの要求を適度の応答速度で処理し、未処理の要求がキューから適度の速さで取り除かれれば、問題はありません。しかし、ファイル所有領域の応答が悪いと、キューが長くなってストレージが占有され、接続されているアプリケーション所有領域のパフォーマンスが著しく低下することがあります。さらに、アプリケーション所有領域のパフォーマンスが低下すると、それが他の領域に広がる場合があります。この状態は「同情病」と呼ばれることがあります。しかし、この状態をより適切に表現するなら、制御しないと複数の領域にわたってパフォーマンスの低下を招くおそれがあるシステム間のキューイングであるということができます。

割り振りキューの管理

割り振りキューを管理するには、3つの方法があります。

リソース定義を使用してキューを管理する

単純な制御要件を含むシステム間リンク (重要ではないトラフィックを伝送するリンクなど) には、CONNECTION と IPCONN の各リソース定義で QUEUELIMIT と MAXQTIME の各オプションを指定できます。

© Copyright IBM Corp. 1977, 2009

QUEUELIMIT は、接続の空きセッションを待っている間、CICS がキューイングする割り振り要求の最大数を定義します。

MAXQTIME は、応答がない接続の空きセッションを待って、割り振り要求がキューに留まるおおよその時間を定義します。MAXQTIME は、キューイング限度が QUEUELIMIT に指定され、キューがこの限度に達したときだけ使用されます。

割り振り要求を受信することによって QUEUELIMIT 値を超えてしまう場合には、CICS によって、新しい要求を最大キューイング時間内に処理できるかどうかが、キューの処理速度に基づいて計算されます。要求が処理されない場合、CICS によってそのキューが除去されます。その接続のセッションが空くまでは、キューイングはそれ以上行われません。セッションが空くと、キューイングが再び始まります。

QUEUELIMIT および MAXQTIME 設定値を超えたために CICS が割り振り要求をパージすると、SYSIDERR 条件がアプリケーション・プログラムに戻されます。

QUEUELIMIT 属性および MAXQTIME 属性については、「*CICS Resource Definition Guide*」のCONNECTION 定義の属性およびIPCONN 定義の属性を参照してください。

NOQUEUE オプションの使用

明示的な 割り振り要求を制御する方式としては、さらに、EXEC CICS ALLOCATE コマンドの NOQUEUE|NOSUSPEND オプションを指定する方法があります。

しかし、この方法は特定の要求を制御することはできますが、それらの要求が出されたときのキューの状態は考慮に入れません。さらに、この方法は、*暗黙の* 割り振り要求 (セッション要求が、例えば、機能シップ要求によって行われた場合) を制御する点では役に立ちません。API オプションのプログラミング情報については、「*CICS アプリケーション・プログラミング・リファレンス*」の ALLOCATE (APPC)を参照してください。

XISQUE および XZIQUE グローバル・ユーザー出口の使用

割り振り要求のキューイングは、グローバル・ユーザー出口プログラムで制御できます。こうすると、接続でキュー限度を設定するよりも柔軟に制御することができます。XISQUE を使用して IPIC キューを、XZIQUE を使用して MRO キューと APPC キューを管理します。

XISQUE 出口および XZIQUE 出口を使用して、キューイングの問題 (ボトルネック) を迅速に検出することができます。両方の出口を使用することにより、キューの長さに応じて、割り振り要求をキューに入れたり、拒否したりすることができます。また、XISQUE と XZIQUE を使用して、ボトルネックのある接続を停止した後、再確立することもできます。

XZIQUE 出口は、XISCONA 出口が MRO 接続および APPC 接続に提供する機能を拡張します。XISCONA は、機能シップおよび DPL 要求 (非同期処理のために機能シップされた EXEC CICS START 要求を含む) の場合にのみ呼び出されます。XZIQUE は、機能シップおよび DPL の他に、トランザクション・ルーティング、非同期処理、分散トランザクション処理の各要求の場合にも呼び出されます。XISCONA 出口と比べて、XZIQUE はそのアクションが基づいているより詳細な情

報を受け取ります。XISCONA と XZIQUE の関係については、「*CICS Customization Guide*」を参照してください。

キューイング・グローバル・ユーザー出口プログラムの使用

この出口が使用可能である場合、XZIQUE または XISQUE グローバル・ユーザー出口プログラムによって、ローカル・システムの特定の接続に対する割り振りキューの状態を検査することができます。

この情報はパラメーター・リストで出口プログラムに渡されます。このリストは、セッション要求に応じて、非特定の割り振り要求または特定のモードグループの要求に関するデータを提供するために構造化されます。XZIQUE 出口を使用する場合、非特定の割り振り要求は、モードグループを指定しない MRO、LU6.1、および APPC の各セッションについて行われます。

グローバル・ユーザー出口プログラムでは、パラメーター・リストで渡される情報を使用して、以下のシステム処置が選択されます。

- 割り振り要求をキューに入れる。この処置は、キュー限度に達していない場合にのみ実行可能です。
- 割り振り要求を拒否する。
- この割り振り要求を拒否し、この接続に対してキューイングされている要求をすべて除去する。
- この割り振り要求を拒否し、このモードグループに対してキューイングされている要求をすべて除去する。

出口プログラムでは、以下のいずれかの基準に基づいて、その処置が決定されます。

- 割り振りキューの長さ
- キューイングされている要求の数が QUEUELIMIT オプションによって設定された限度に達したかどうか。キュー限度に達していなければ、その要求をキューに入れることができます。
- セッションがその接続に割り振られる速度。キュー限度に既に達していても、セッションの割り振りが十分に速い場合は、現在の要求だけを拒否することができます。キュー限度に既に達していて、セッションの割り振りが遅い場合は、キュー全体を消去することができます。

XISQUE パラメーター・リストで渡される情報の詳細および XISQUE 出口プログラムの設計とコーディングのヒントについては、「*CICS Customization Guide*」に記載されているプログラミング情報を参照してください。

XZIQUE パラメーター・リストで渡される情報の詳細および XZIQUE 出口プログラムの設計とコーディングのヒントについては、「*CICS Customization Guide*」に記載されているプログラミング情報を参照してください。

第 25 章 シップされた端末定義の効率的な削除

この章では、重複してシップされた端末定義を CICS がどのようにして削除するかを説明します。

この章には、次のトピックが収められています。

- 『シップされた端末の削除方法の概要』
- 312 ページの『タイムアウト削除の実装』
- 313 ページの『タイムアウト削除のパフォーマンスのチューニング』

シップされた端末の削除方法の概要

トランザクション・ルーティングの環境において、端末定義は、アプリケーション所有領域 (AOR) に静的に定義する代わりに、それが最初に必要になったときに、端末専有領域 (TOR) から AOR へ「シップ」することができます。

注: この「端末」は、APPC デバイスかシステムです。この場合、シップされる定義は APPC 接続の定義です。

シップされた定義が冗長になるのは、次の場合です。

- 端末ユーザーがログオフする。
- 端末ユーザーがリモート・トランザクションの使用をやめる。
- TOR がシャットダウンされる。
- TOR が再始動され、自動インストールされた端末定義がリカバリーされず、自動インストール・ユーザー・プログラム DFHZATDX が同じ端末群に新しい端末 ID 群を割り当てる。

冗長定義は、ある段階で、AOR (および TOR と AOR の間の中間システム) から削除する必要があります。便宜上、AOR と中間システムを合わせて「バックエンド・システム」と呼びます。これは、とりわけ上記の最後のケースで、TOR とバックエンド・システムの端末 ID が不一致になるのを避けるために必要です。

重複してシップされた定義を CICS によって削除するための方法には、次の 2 つがあります。

- 選択的削除
- タイムアウト削除メカニズム

選択的削除

端末定義がインストールされるたびに、CICS は、固有の「インスタンス・トークン」を作成し、それをその定義内に格納します。

したがって、その定義が別の領域にシップされると、そのトークンの値も一緒にシップされます。トランザクション・ルーティングによるすべての接続要求では、このトークンが機能管理ヘッダー (FMH) で渡されます。接続処理の際、シップされた定義がリモート領域に既にあれば、シップされた定義のトークンが *TOR* によっ

© Copyright IBM Corp. 1977, 2009

て渡されたものと一致する場合のみ そのリモート領域の定義が使われます。リモート領域にない場合は、その定義は削除され、最新の定義がシップされます。

タイムアウト削除メカニズム

バックエンド・システムでタイムアウト削除メカニズムを使用すれば、定義された期間にわたってトランザクション・ルーティングに使用されなかったシップされた定義を削除することができます。この目的は、シップされた定義が、使用されている間だけインストールされているようにすることです。

注: シップされた定義は、その端末に自動開始記述子 (AID) が関連付けられている場合には削除されません。

タイムアウト削除を使えば、シップされた定義を柔軟に制御することができます。CICS によって、次のことが可能です。

- シップされた定義を削除の対象になるまでインストール状態にしておく最小時間を規定する。
- このメカニズムを呼び出す時間間隔を規定する。
- これらの時間をオンラインでリセットする。
- タイムアウト削除メカニズムをただちに呼び出す。

このメカニズムを制御するパラメーターを使用すれば、システムが最もビジーでない状態のときに「正確な」操作が行われるようにすることができます。オペレーターは、細かい調整が必要になったら、CEMT トランザクションを使って、これらのパラメーターをオンラインで修正したり、このメカニズムをただちに呼び出したりすることができます。

タイムアウト削除の実装

端末がシップされた先の CICS Transaction Server for z/OS システムでタイムアウト削除を使用する場合には、2 つのシステム初期設定パラメーターを設定するだけで十分です。

DSHIPIDL={020000|hhmmss}

非アクティブの シップされた端末定義がこの領域でインストール状態になっていなければならない最小限の時間を時、分、秒で指定します。CICS タイムアウト削除メカニズムが呼び出されると、シップされた定義のうち、この時間よりも長い間非アクティブにあったものだけが削除されます。

トランザクション・ルーティング環境のアプリケーション所有領域と中間領域でこのパラメーターを使えば、端末定義の削除が早過ぎたためにそれらを再びシップしなければならない事態が避けられます。

hhmmss

1 桁から 6 桁の数字を 0 から 995959 の範囲で指定します。数字が 6 桁より短いと、先行ゼロが入れられます。

DSHIPINT={120000|0|hhmmss}

CICS タイムアウト削除メカニズムを呼び出す間隔を指定します。タイムアウト削除メカニズムは、シップされた端末定義のうち、DSHIPIDL パラメーターによって指定された時間よりも長い間使われなかったものを削除します。

このパラメーターは、トランザクション・ルーティング環境のアプリケーション所有領域と中間領域において、次のものを制御するために使用できます。

- タイムアウト削除メカニズムを呼び出す頻度
- 一括削除操作を行うおおよその時刻。CICS 始動時からの相対時間です。

0
タイムアウト削除メカニズムは呼び出されません。端末専有領域内や、シップされた定義を使用しない場合に、この値を設定することがあります。

hhmmss
1 桁から 6 桁の数字を 1 から 995959 の範囲で指定します。数字が 6 桁より短いと、先行ゼロが入れられます。

システム初期設定パラメーターの指定方法についての詳細は、「*CICS System Definition Guide*」の 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。

CICS が始動したあとで、CEMT か EXEC CICS INQUIRE DELETSHIPPED コマンドを使用すれば、DSHIPIDL と DSHIPINT の現在の設定値を知ることができます。SET DELETSHIPPED コマンドを使用すれば、一括削除操作の時期を制御するために、タイムアウト削除メカニズムの次の呼び出しまでの間隔を設定し直すことができます。 (この改訂された間隔は、*そのコマンドが出されたときから* 始まります。リモート削除メカニズムが最後に呼び出されたときからでも、CICS 始動時からでも**ありません**。) あるいは、PERFORM DELETSHIPPED コマンドを使用すれば、タイムアウト削除メカニズムをただちに呼び出すことができます。

CEMT INQUIRE、PERFORM、SET DELETSHIPPED の各コマンドについては、「*CICS Supplied Transactions*」の CEMT INQUIRE DELETSHIPPED、CEMT PERFORM DELETSHIPPED、およびCEMT SET DELETSHIPPEDを参照してください。EXEC CICS でそれらに対応するコマンドについてのプログラミング情報は、「*CICS System Programming Reference*」の INQUIRE DELETSHIPPED、PERFORM DELETSHIPPED、およびSET DELETSHIPPEDを参照してください。

タイムアウト削除のパフォーマンスのチューニング

DSHIPINT と DSHIPIDL の設定を慎重に選択すれば、シップされた定義の一括削除は最低限の回数で済み、システムの負荷が軽いときに一括削除がスケジュールされます。

反対に設定の選択が悪いと、不必要な一括削除操作が行われます。DSHIPINT と DSHIPIDL の指定方法のヒントを下記に示します。

DSHIPIDL

この値を設定する場合には、リモート・ユーザーがこのシステムのリソースにアクセスする作業期間の長さを考慮する必要があります。かれらは、システムに一日中断続的にアクセスするのか。それとも、かれらの作業は、短い期間に集中して行われるのか。

値の設定が低過ぎると、不必要に定義の削除と再シップが行われることがあります。さらに、自動トランザクション開始 (ATI) 要求が「端末未認識」条件で失敗す

ることもあり得ます。この条件は、ATI 要求が、このシステムに定義されていない端末を指定すると起こります。通常、端末が定義されていない理由は、その端末がリモート・システムによって所有され、シップ可能端末が使用されているとき、トランザクション・ルーティングが前にその端末から行われていないためです。一時的に非アクティブにあるシップされた定義の存続期間が短過ぎると、「端末未認識」条件を処理する XALTENF と XICTENF グローバル・ユーザー出口を呼び出す回数が増加する可能性があります。

DSHIPINT

この値を使用すれば、一括削除操作を行う時刻を制御できます。

例えば、CICS を通常午前 7 時に開始する場合には、DSHIPINT を 150000 に設定すると、システムにアクセスするユーザーが少なくなる午後 10 時にタイムアウト削除メカニズムが呼び出されます。

重要: CICS が、例えば、障害のためにリサイクルされると、タイムアウト削除間隔はリセットされます。上の例で言えば、CICS が午後 8 時にリサイクルされると、タイムアウト削除メカニズムは、次の日の午前 11 時に呼び出されます (CICS の初期設定から 15 時間後)。このような状況では、SET DELETSHIPPED と PERFORM DELETSHIPPED コマンドを使用することによって、タイムアウト削除を行う時期を正確に制御することができます。

CICS の統計を使えば、DFHIPIDL と DFHIPINT パラメーターの調整が容易になります。統計はオンラインで使用可能であり、DFHA04DS DSECT によってマップされます。提供されている統計について詳しくは、 「*CICS パフォーマンス・ガイド*」を参照してください。

第 6 部 システム間環境におけるリカバリーおよび再始動

このパートでは、相互通信環境で問題が発生したときに、CICS でどのようなことが起こるのか、また、ユーザーがそれに対してどのような処置を行うことができるのかについて説明します。

317 ページの『第 26 章 相互接続されたシステムにおけるリカバリーと再始動』では、個々のセッション障害と、システム障害および再始動について説明します。

349 ページの『第 27 章 相互通信と XRF』では、CICS 拡張リカバリー機能 (XRF) について、相互通信に影響がある部分を説明します。

351 ページの『第 28 章 相互通信と VTAM 持続セッション』では、VTAM 持続セッションのための CICS サポートについて、相互通信に影響がある部分を説明します。

© Copyright IBM Corp. 1977, 2009

第 26 章 相互接続されたシステムにおけるリカバリーと再始動

この章では、CICS のリカバリーおよび再始動に関して、特に相互通信環境に当てはまる内容について説明しています。ユーザーは、作業単位 (UOW)、同期点、動的トランザクション・バックアウト、および単一の CICS システムにおけるリカバリーと再始動に関連するその他のトピックについて、その概念を理解しているものとします。

これらのトピックの詳細については、「*CICS Recovery and Restart Guide*」を参照してください。

相互通信環境では、単一システムのほとんどの概念がそのまま適用されます。それぞれのシステムには、独自のシステム・ログ (非 CICS システムではこれに相当するもの) があります。それらのシステムは、通常、そのリカバリー可能リソースに行った変更をコミットしたり、バックアウトしたりすることができます。

しかし、相互通信環境では、接続されている複数のシステムによってとられるアクションを 1 つの作業単位に含めることができます。そのような作業単位は、アクセスするリソースが複数のシステムに分散しているため、*分散* 作業単位と呼ばれます。分散作業単位は複数の*ローカル* 作業単位からなり、それぞれ、参加システムの 1 つで行われる作業を表しています。分散作業単位では、各参加システムが、自身の行った変更のコミットに同意することが必要です。このことは、それらのシステムが同期点要求と応答をシステム間セッションで交換しなければならないことを意味します。単一システムと複数システムのリカバリー機能では、この点が 1 つの大きな違いです。

この章の残りの部分には、次のトピックが含まれています。

- 『用語』
- 318 ページの『同期点交換』
- 321 ページの『リカバリーの機能とインターフェース』
- 326 ページの『初期始動とコールド・スタート』
- 329 ページの『接続定義の管理』
- 331 ページの『棚上げを完全にはサポートしない接続』
- 334 ページの『APPC 接続静止処理』
- 334 ページの『問題判別』

用語

ローカル UOW に関する情報は、システム・プログラミング・コマンドや CEMT コマンド、およびこの章で後述する CICS メッセージによって戻されます。

重要:

1. この章では、表記を簡潔にするために、以下の CICS プロダクトを、*CICS Transaction Server for z/OS* および *CICS TS for z/OS* という用語で表すものとします。

© Copyright IBM Corp. 1977, 2009

- CICS Transaction Server for z/OS バージョン 3 リリース 2
- CICS Transaction Server for z/OS バージョン 3 リリース 1
- CICS Transaction Server for z/OS バージョン 2 リリース 3
- CICS Transaction Server for z/OS バージョン 2 リリース 2

2. この章では、「作業単位」および「UOW」という用語は、*ローカルな* 作業単位、つまり*分散作業単位のうちローカル・システムのリソースに関連する部分*を意味します。

   分散 作業単位を意味する場合には、この用語を明示的に使用します。

この章では、*未確定*、*開始プログラム*、*エージェント*、*コーディネーター*、*従属*、*棚上げ*、*再同期* など、いくつかの新しい用語が使用されています。これらの用語は、本文の中で例とともに説明します。CICS の用語集を参照することもお勧めします。

この章のこれ以降の部分には、以下のセクションが含まれています。

- 『同期点交換』では、CICS の同期点の流れの例と、それに使用する用語を説明します。
- 321 ページの『リカバリーの機能とインターフェース』では、CICS が通信障害からリカバリーする方法と、CICS のリカバリー・アクションを制御するためのコマンドを説明します。このセクションとそれに続く 2 つのセクションは、他の CICS Transaction Server for z/OS (CICS TS for z/OS) システムへの MRO、IPIC、および SNA を介した ISC (APPC) 並列セッション接続だけに適用されます。
- 326 ページの『初期始動とコールド・スタート』では、相互接続されたシステムにおける初期始動とコールド・スタートの効果と、コールド・スタートがいつ可能なのかを説明します。
- 329 ページの『接続定義の管理』では、他の CICS TS for z/OS システムへの MRO、IPIC、およびSNA を介した ISC (APPC) 並列セッション接続の変更や廃棄を安全に行う方法について説明します。
- 331 ページの『棚上げを完全にはサポートしない接続』では、他の CICS TS for z/OS システムへの MRO リンク、IPIC リンク、またはSNA を介した ISC (APPC) 並列セッション・リンク以外の接続に適用される例外について説明します。
- 334 ページの『問題判別』では、通信の障害やリカバリー中に CICS が出すメッセージについて説明し、未確定や再同期の障害を解決する方法の例を示します。

同期点交換

次の例を考えてみます。

同期点の例:

品目の注文が端末から入力されると、受注入力トランザクションは、次のことを行います。

1. ***在庫ファイルを照会し、注文の数だけ減らします。***
2. ***商品の発送指示を区画内一時データ・キューに書き込みます。***

3. ***現行 UOW の終了を示すために同期点をとります。***

単一の CICS システムでは、この同期点によって、1 (318 ページ) と 2 (318 ページ) の両方のステップがコミットされます。

在庫ファイルがリモート・システムにあり、それを例えば CICS 機能シップでアクセスする場合も、同じ結果が得られるようにしなければなりません。これは、次のようにして達成されます。

1. ローカル・トランザクションが同期点要求を出すと、CICS がリモート・トランザクション (この場合 CICS ミラー・トランザクション) に同期点要求を送ります。
2. リモート・トランザクションは、在庫ファイルの変更をコミットして、ローカル CICS システムに肯定応答を送ります。
3. CICS は、一時データ・キューの変更をコミットします。

リモート・システムへの同期点要求の送信と応答の受信の間、ローカル・システムは、リモート・システムが変更をコミットしたのかどうか分かりません。この期間は**未確定**期間と呼ばれます。これを 320 ページの図 86 に示します。

未確定期間に入るまでにシステム間セッションに障害が発生すると、このセッションの両側で通常の方法を使ってバックアウトが行われます。この期間が終了すれば、変更は両側でコミットされています。しかし、未確定期間の間にシステム間セッションに障害が発生した場合、ローカル CICS システムは、リモート・システムがその変更をコミットしたのかバックアウトしたのかを判断できません。

同期点フロー

同期点要求と応答がシステム間会話でどのように交換されるかは、APPC と LUTYPE6.1 のアーキテクチャーで定義されています。CICS MRO および IPICは APPC リカバリー・プロトコルを使用します。 APPC と LUTYPE6.1 では同期点フローの形式が異なりますが、同期点交換の概念は似ています。

CICS では、同期点の交換に関連するフローは、トランザクションによって明示的または暗黙に出される SYNCPOINT に対応して自動的に生成されます。しかし、これらのフローを基本的に理解すれば、アプリケーションの設計が容易になるとともに、同期点アクティビティーにおけるセッションやシステムの障害の結果が理解できるようになります。これらのフローについて詳しくは、「*CICS Distributed Transaction Programming Guide*」を参照してください。

320 ページの図 86 から 321 ページの図 88 で、同期点フローの例をいくつか示します。これらの図で、(1) のように括弧で囲んだ数字は、各フローにおける処理の順序を示しています。

CICS タスクは 1 つ以上の UOW を持つことができます。EXEC CICS SYNCPOINT コマンドや EXEC CICS RETURN コマンドなどを出すことによって同期点アクティビティーを開始するローカル UOW を**開始プログラム**と言います。開始プログラムから同期点要求を受け取るローカル UOW を、**エージェント**と言います。最も簡単な例を 320 ページの図 86 に示します。開始プログラムとエージェントの間で会話が 1 つ行われます。同期点アクティビティーの始めに、開始プログラムは**コミット**要求をエージェントに送ります。エージェントは、その変更をコミッ

トし、コミット応答を返します。次に、開始プログラムがその変更をコミットし、作業単位が完了します。しかし、エージェントは、その UOW のリカバリー情報を、廃棄してよいという通知を (「廃棄」フローによって) パートナーから受け取るまで、その情報を保持します。

コミット・フローとコミット済みフローの間では、開始プログラムは未確定の状態ですが、エージェントはそうではありません。未確定の状態でないローカル UOW は、両システムのリソースのコミットメントを調整するため、**コーディネーター**と呼ばれます。未確定の状態にあるローカル UOW は、コーディネーターによって出されたコミットまたはバックアウトの決定に従う必要があるため、**従属**と呼ばれます。

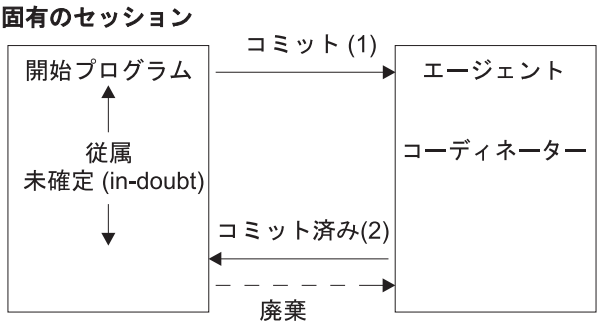


図 86. 同起点処理フロー - 固有のセッション: この分散 UOW には、1 つのコーディネーターと 1 つの従属があります。コーディネーターは未確定の状態ではありません。

図 87 に、もっと複雑な例を示します。ここでは、エージェント UOW (エージェント 1) が 3 つ目のローカル UOW (エージェント 2) との会話を持ちます。エージェント 1 は、開始プログラムに応答を返す前に、この会話で同期点アクティビティーを開始します。エージェント 2 がまずコミットし、次にエージェント 1 がコミットし、最後に開始プログラムがコミットします。図 87 では、エージェント 1 は開始プログラムのコーディネーターであると同時に、エージェント 2 の従属であることに注意してください。

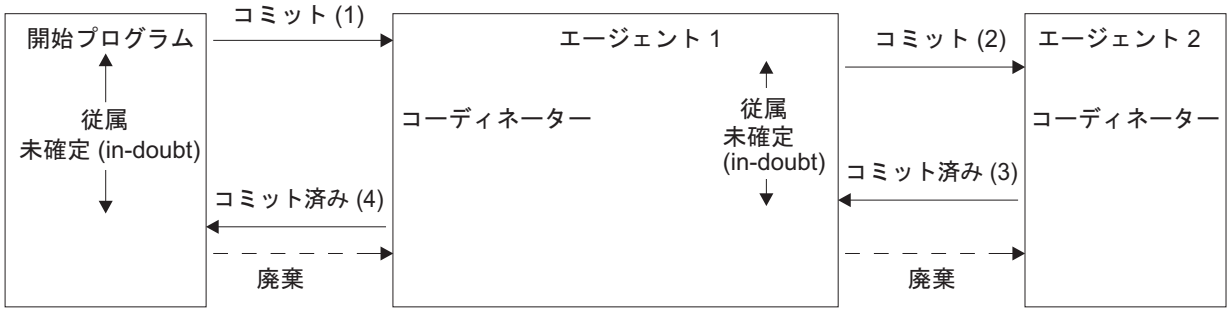


図 87. 同起点処理フロー - チェーン・セッション: この分散 UOW では、エージェント 1 は開始プログラムのコーディネーターであると同時に、エージェント 2 の従属です。

321 ページの図 88 に、もっと一般的な場合を示します。開始プログラム UOW には、複数の (直接接続された) エージェントがあります。開始プログラムは、その各エージェントに同期点をとることを知らせなければなりません。これは、最後を除

くすべてのエージェントに「コミットの準備」要求を送ることによって行われます。**最後のエージェント**とは、コミットの準備をするように通知を受けないエージェントのことです。

注: CICS は、同期点をとるときに最後のエージェントを動的に選びます。CICS 外部インターフェースには、最後のエージェントを特定する方法はありません。

各エージェントは、「準備」要求を受け取ると、「コミット」要求を返します。このような「準備」要求がすべて送信され、「コミット」応答がすべて受信されると、開始プログラムは「コミット」要求をその最後のエージェントに送ります。最後のエージェントから「コミット済み」標識が返されると、開始プログラムは、他のすべてのエージェントに「コミット済み」要求を送ります。

図 88 では、開始プログラムは、エージェント 1 のコーディネーターであると同時に、エージェント 2 の従属であることに注意してください。エージェント 2 が最後のエージェントです。

複数セッション - 開始プログラムが複数のエージェントを所有する

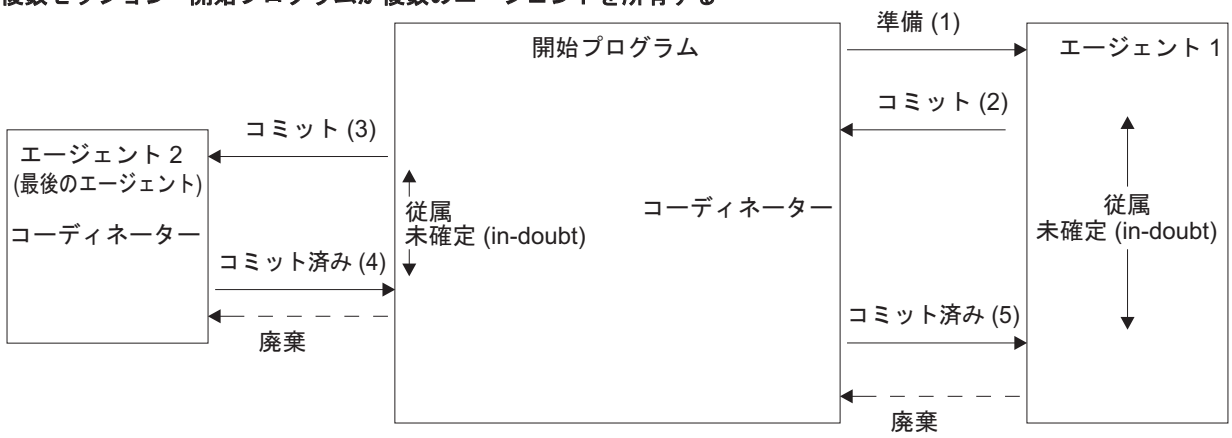


図 88. 同起点処理フロー - 複数セッション: この分散 UOW では、開始プログラムは、エージェント 1 のコーディネーターであると同時に、エージェント 2 の従属です。エージェント 2 は最後のエージェントですので、コミットの準備の通知は受けません。

リカバリーの機能とインターフェース

この項では、通信障害や CICS システムの障害をリカバリーするために、CICS によって提供される機能とインターフェースについて説明します。

重要:

CICS のすべてのリリースが同じレベルのサポートを提供するわけではありません。このセクションでは、他の CICS Transaction Server for z/OS システムへの MRO 接続、IPICによる接続、および SNA を介した ISC (APPC) 並列セッション接続について説明します。このほとんどは他の種類の接続にも当てはまりますが、制限がいくつかあります。CICS Transaction Server for z/OS 以外のシステムとの接続および LU6.1 や APPC の単一セッション接続における制限については、 331 ページの『棚上げを完全にはサポートしない接続』を参照してください。

さらに、この項では、それぞれの CICS システムが正しく再始動される (つまり、START システム初期設定パラメーターに AUTO が指定されている) ものとします。初期始動が行われた場合は、接続されているシステムに関連してきます。これについては、326 ページの『初期始動とコールド・スタート』を参照してください。

リカバリー機能

通信障害のために作業単位の状態が未確定のままになっている場合、CICS は、2 つのアクションのどちらかを行います。

どのような場合に CICS が各アクションを実行するかについては、『トランザクション定義の未確定属性』で説明しています。

- 更新されたリソースのコミットメントを、すべてのシステムが次に通信状態になるまで延期する。作業単位は**棚上げ**されます。通信が回復すると、コミットするかバックアウトするかの決定をコーディネーター・システムから入手します。その作業単位は**棚上げが解除され**、更新は、**再同期**と呼ぶ処理によってローカル・システムでコミットまたはバックアウトされます。
- ローカル・リソースをコミットするか、バックアウトするかの決定を一方的に行う。この場合、この決定が他のシステムと矛盾することがあります。通信が回復すると、それらの決定が比較され、隣接するシステムの間に矛盾があれば警告が出されます (334 ページの『CICS のリカバリー・アクションを知らせるメッセージ』を参照)。

この 2 つの機能には、相反する利点と欠点があります。未確定の UOW を延期すると、更新されたデータはそのあとのアクセスに対しロックされます。したがって、この欠点と、一方的な決定を行うことによってデータの整合性が破損する可能性とを比較検討する必要があります。一方的な決定を行う場合には、整合性を復元するために、調整ジョブなどのアプリケーションに依存する処理がおそらく必要になりますが、CICS ではこれに対する一般的な方式は提供されません。

リカバリー・インターフェース

このセクションでは、未確定の間に失敗した作業単位の制御と調査に使用できるリソース定義オプション、システム・プログラミング・コマンド、および CICS 提供のトランザクションの概要を示します。

リソース定義、システム・プログラミング・コマンド、および CEMT トランザクションの詳細については、それぞれ、「*CICS Resource Definition Guide*」の「CICS Transaction Server for z/OS Resource Definition Guide」、「*CICS System Programming Reference*」マニュアルの「CICS Transaction Server for z/OS System Programming Reference」、および「*CICS Supplied Transactions*」マニュアルの「CICS Transaction Server for z/OS Supplied Transactions」を参照してください。

トランザクション定義の未確定属性

トランザクションを定義するときに未確定属性を指定すれば、未確定の期間に通信障害が起こったあとに CICS がとるアクションを制御することができます。それには、TRANSACTION 定義の WAIT、WAITTIME、および ACTION オプションを使用します。

これらのオプションは、コーディネーターとの通信が切れ、UOW が未確定の期間にあるときに使用されます。

WAIT({YES|NO})
作業単位が、ACTION で指定されたアクションを実行する前に、未確定期間に入った後で発生した障害のリカバリー処理を保留しながら待機するかどうかを指定します。

YES
UOW は、障害のリカバリーを保留しながら待機して、その未確定状態を解決し、リカバリー可能リソースをバックアウトするかコミットするかを決定します。つまり、棚上げされます。

NO
UOW は待機しません。CICS は、ACTION 属性で指定されたアクションが何であれ、それをただちに実行します。

注: WAIT オプションの設定は、他のシステム設定によって指定変更されることがあります。「 *CICS Resource Definition Guide*」のTRANSACTION 定義属性にある DEFINE TRANSACTION の説明を参照してください。

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一方で、作業単位に適切な時間内にロックを解放させることができます。

ACTION({BACKOUT|COMMIT})
作業単位のコーディネーターとの通信が切れ、その UOW が未確定期間に入った場合に実行するアクションを指定します。

BACKOUT
リカバリー可能リソースのすべての変更がバックアウトされ、それらのリソースは UOW が開始される前の状態に戻されます。

COMMIT
リカバリー可能リソースのすべての変更がコミットされ、UOW は完了と見なされます。

このアクションは WAIT 属性に依存します。WAIT が YES の場合、WAITTIME オプションに指定された間隔が障害のリカバリーより前に経過しない限り、ACTION は行われません。

BACKOUT を指定するか COMMIT を指定するかは、トランザクションがリモート・システムのリソースにどのような変更を行うかによって異なることがあります。『未確定属性の指定例』を参照してください。

未確定属性の指定例:

次の例は、トランザクションの未確定属性を指定するケースを示しています。

例:

トランザクションに部品番号が指定されます。このトランザクションは、ローカル・ファイルの項目を検査して、その部品の在庫があるかどうかを確認し、在庫ファイルを更新して、在庫の数量を減らし、リモート一時データ・キューにレコードを送って、部品の発送を開始します。

ローカル・ファイルの更新は、リモート一時データ (TD) キューに追加が行われたときだけ行うべきです。また、TD キューの更新は、ローカル・ファイルが更新されたときだけ行うべきです。これを達成するにはまず、このファイルと TD キューを両方ともリカバリー可能リソースとして指定する必要があります。こうすることにより、同期点処理の未確定期間中にシステムやセッションに障害が発生した場合を除くすべてのケースで、リソースに対する変更の同期が保証されます (つまり、両方の変更がバックアウトされるか、コミットされます)。

未確定期間中の通信障害 (例えば、リモート・システムの障害) に対処するには、ローカル・トランザクションの定義、WAIT(YES)、ACTION(BACKOUT)、および WAITTIME に、リモート・システムのリサイクルに必要なだけの時間を指定します。これによって、指定の時間内に通信が回復すれば、再同期が自動的に行われます。WAITTIME の期間内であれば、再同期が行われるまでは、ローカル UOW は棚上げされ、在庫ファイルのそのレコードに対するロックが保持されます。

時間限度内に通信が回復しないと、ローカル・システムの在庫ファイルの変更はバックアウトされます。リモート・システムの TD キューへの追加がコミットされているかどうかは分かりません。通信が回復したあとで調べる必要があります。

INQUIRE コマンド

CEMT と EXEC CICS のインターフェースにはいくつかの照会コマンドがあるので、それらを使用すれば、分散作業単位の実行を調べたり、問題を診断したりすることができます。

注: 以下のコマンドのリストでは、INQUIRE CONNECTION は MRO 接続および SNA を介した ISC (APPC) 接続に適用されます。INQUIRE IPCONN は IPIC 接続に適用されます。
下記にこれらのコマンドの要約を示します。

INQUIRE {CONNECTION | IPCONN} RECOVSTATUS
ローカル・システムと接続システムの間に未解決の再同期作業があるかどうかを調べるときに使用します。返される CVDA 値は次のとおりです。

NORECOVDATA
未解決のリカバリー情報は両側ともありません。

NOTAPPLIC
これは IPICでも、APPC 並列セッションでも、CICS-CICS 間 MRO 接続でもありませんので、2 フェーズ・コミット・プロトコルはサポートされません。

NRS CICS にはこの接続に対し未解決のリカバリーはないが、パートナーにはあるかもしれません。

RECOVDATA
その接続に関連した未確定の作業単位があるか、その接続で

FORGET を待っている未解決の再同期があります。再同期は、その接続が次にアクティブになるか、UOW が棚上げ解除になるときに行われます。

INQUIRE {CONNECTION | IPCONN} PENDSTATUS
UOW の中に、接続システムによる初期始動のために再同期が不可能なものがあるかどうかを調べるために使用します。

INQUIRE CONNECTION XLNSTATUS (APPC 並列セッションのみ)
そのリンクが同期点 (同期レベル 2) 作業を現在サポートできるかどうかを調べるために使用します。詳細については、328 ページの『ログ名交換プロセス』を参照してください。

注: XLNSTATUS は IPCONN には適用されません。

INQUIRE UOW
作業単位が待っている理由、または、棚上げされている理由を調べるときに使用します。理由が接続障害の場合は (WAITCAUSE オプションが CONNECTION という CVDA 値を戻す)、SYSID と LINK オプションは、その UOW を待機状態にした、または、棚上げされている原因となった、リモート・システムのシステム ID とネット名を戻します。

INQUIRE UOW は*ローカル* UOW の情報を返すことに注意してください。つまり、分散 UOW の場合は、そのローカル・システムで必要な作業の情報だけを返します。分散 UOW の情報は、NETUOWID フィールドで戻されたネットワーク全体での ID と、他のシステムのローカル UOW の ID を突き合わせることによって得られます。この方法については、その例を343 ページの『再同期の失敗の解決』で示します。

INQUIRE UOWLINK
このコマンドによって、個々の UOW で再同期が必要かどうかを照会することができます。これは、分散 UOW に関係する接続の情報を調べるときに使用します。

ローカル UOW の場合には、INQUIRE UOWLINK は、その分散 UOW に関係するシステムへの接続を表すトークンのリスト (*UOW リンク*) を戻します。UOW リンクごとに、INQUIRE UOWLINK は次の情報を戻します。

- CONNECTION 名
- その接続の再同期状態
- その接続がコーディネーターに対するものなのか、従属システムに対するものなのか

これらのコマンドを使って、分散作業単位の問題を診断する例については、338 ページの『問題判別の例』を参照してください。

SET {CONNECTION | IPCONN} コマンド

通常はトランザクション定義によって制御される未確定アクションを、例外的に指定変更したい場合があります。

例えば、接続システムが再始動に予期したより長くかかるような場合です。その接続システムが UOW のコーディネーターであれば、EXEC CICS または CEMT SET

{CONNECTION | IPCONN} UOWACTION(FORCE|COMMIT|BACKOUT) コマンドを使って、コミットかバックアウトかの決定をその UOW にローカルで一方的に行わせることができます。

注: SET CONNECTION は MRO 接続および SNA を介した ISC (APPC) 接続に適用されます。SET IPCONN は IPIC (IP) 接続に適用されます。

次のコマンドについては、328 ページの『ログ名交換プロセス』と 329 ページの『接続定義の管理』を参照してください。

- SET {CONNECTION | IPCONN} PENDSTATUS
- SET {CONNECTION | IPCONN} RECOVSTATUS.

初期始動とコールド・スタート

あるシステムが初期始動かコールド・スタートを行うと、トランザクション処理ネットワークに例外条件が発生することがありますが、この項では、これを管理する機能について説明します。

重要:

- 特に断りがない限り、このセクションでは、MRO リンク、IPICリンク、または SNA を介した ISC (APPC) 並列セッション・リンクで接続されている CICS Transaction Server for z/OS システムで初期始動やコールド・スタートを行ったときの影響について説明します。その他の接続が使用されているときの影響については、331 ページの『棚上げを完全にはサポートしない接続』を参照してください。
- この章では、これ以降、「コールド・スタート」という用語は、CICS TS for z/OS の句におけるコールド・スタートを意味します (下記参照)。「初期始動」を意図する場合には、明示的にその用語を使用します。

CICS Transaction Server for z/OS システムは、完全にリカバリーしなくても、以下の 2 通りの方法で開始することができます。

初期始動

初期始動 は、次のどちらかの状況で行われます。

- START システム初期設定パラメーターに 'INITIAL' が指定されている。
- START システム初期設定パラメーターに 'AUTO' が指定されており、リカバリー管理ユーティリティー・プログラム DFHRMUTL によって、AUTOINIT 自動開始指定変更がグローバル・カタログに設定されている。

初期始動時に、ローカル・リソースとリモート・リソースの情報はすべて消去され、すべてのリソース定義が CSD または CICS テーブルから再インストールされます。

初期始動は、例外的な状況でのみ行われます。初期始動が適している例としては、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後に、グローバル・カタログやシステム・ログが破壊された場合。

コールド・スタート
コールド・スタート は、次のどちらかの状況で行われます。

- START システム初期設定パラメーターに 'COLD' が指定されている。
- START システム初期設定パラメーターに 'AUTO' が指定されており、DFHRMUTL ユーティリティーによって、AUTOCOLD 自動開始指定変更がグローバル・カタログに設定されている。

CICS TS for z/OS では、コールド・スタートは、*ローカル・リソース*に関するログ情報が消去され、リソース定義が CSD または CICS テーブルから再インストールされることを意味します。ただし、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再同期情報は保存されます。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。)

CICS を開始する各種の方法については、「*CICS Recovery and Restart Guide*」を参照してください。

コールド・スタートがいつ可能かの決定

コールド・スタート時、システム間リカバリーに関連する情報がシステム・ログから読み取られます。

接続されているシステムは、ローカル・システムが正常に再始動されたものと見なし、未解決の作業を再同期します。CICS の前の実行で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*の更新は、*それらの更新が分散作業単位の一部*であっても、コールド・スタートではリカバリーされません。

コールド・スタートを行っても、次の条件が**すべて**真であれば、データの整合性は損なわれません。

1. *次のどちらか*
   - そのローカル・システムにローカルのリカバリー可能リソースがない (例えば、TOR)、*または*
   - CICS の前の実行が正常に静止され (シャットダウンが即時ではなく通常に行われた)、棚上げされている作業単位がない。

     注: シャットダウンが正常に行われると、CICS がメッセージを出しますので、コールド・スタートを行っても安全かどうかが分かります。棚上げされた UOW がない場合、CICS はメッセージ DFHRM0204 を出します。棚上げされた UOW がある場合、メッセージ DFHRM0203 を出します。この場合、コールド・スタートを実行しないでください。
2. RMI を使用する接続されたリソース・マネージャーがあとで再接続され、再同期が可能である。
3. 再同期に必要なリモート・システムへの接続があとで獲得される。

   コールド・スタートされたシステムには、前のシャットダウン時に使用されていたのと同じ接続定義がある場合もありますし、ない場合もあります。自動的にイ

ンストールされた接続がないと、リモート・システムは、それらが再作成されるようにします。この場合、再同期が行われます。このアクションが行われない場合、または、CEDA か GRPLIST でインストールされた定義がない場合は、何らかのアクションが必要です。これについては、 329 ページの『接続定義の管理』を参照してください。

コールド・スタートされるシステムが VTAM 総称リソース・グループの一部として定義されている場合には、VTAM によって維持される類縁性関係がまだ有効であれば、その接続は正しく再確立されます。しかし、自動インストールの定義が失われると、VTAM 類縁性を終了するのは、必要であっても困難な場合があります。 330 ページの『VTAM 総称リソースとの間の APPC 接続』を参照してください。

DFHRMUTL ユーティリティーは、最後に行われた CICS シャットダウンのタイプに関する情報を戻します。この情報は、コールド・リスタートが可能であるかどうかを判別するために役立ちます。詳細については、「*CICS 操作およびユーティリティー・ガイド*」を参照してください。

ログ名交換プロセス

同期点でのコミットとバックアウトの決定の通信を制御するプロトコルは、システム・ログの情報に依存します。

CICS システムは相互に接続するたびに**ログ名**というトークンを交換します。ログ名は、再同期の際に検査されます。*ログ名交換の失敗* は、そのリカバリー・プロトコルが既に破壊されていることを意味します。失敗は 2 つの形をとります。

1. **コールド / ウォーム・ログ不一致**。コールド / ウォーム・ログ不一致は、一方のパートナーに未解決の再同期作業があるのに、他方のパートナーでログ・データが失われたときに起こります。

   注: 「コールド・スタート」という用語は、「*SNA 対等機能プロトコル*」や、CICS TS for z/OS と通信するその他のプロダクトによって、ログ・データの消失の原因を述べるために使用されます。

   さらに、「コールド・スタート」は、CICS TS for z/OS のメッセージとインターフェースによって、CICS TS for z/OS のログ・データの消失の原因となるパートナー・システムのアクションを述べるときに使用されます。

   しかし、CICS TS for z/OS では、接続されているシステムに関するログ・データの消失は、*初期* 始動 (コールド・スタートではない) によるか、SET CONNECTION NORECOVDATA コマンドによって引き起こされます。
2. **ログ名不一致**。ログ名不一致は、ログ名データの破壊によって起こります。これが起こる原因としては、次のものがあります。
   a. システムの論理エラー
   b. 操作エラー。例えば、バックレベルの CICS リリースから CICS Transaction Server for z/OS へアップグレードするときに、初期始動を行わなかった場合。

ログ名交換プロセスは、APPC アーキテクチャーによって定義されています。この概念と物理的な流れの詳細については、*SNA 対等機能プロトコル* を参照してくださ

い。MRO および IPIC は、APPC と類似したプロトコルを使用します。大きな違いは、パートナーでログ情報が消去された後、これらでは、既存の作業がどのような状態であっても新規の作業を開始できるという点です。APPC 同期レベル 2 のセッションでは、アクションを実行して未解決の再同期作業がすべて削除されるまで、他の作業を実行することはできません。

パートナー・システムが再接続されたあと、INQUIRE CONNECTION PENDSTATUS コマンドを使用して、パートナーでのログ情報の消去によって無効にされた未解決の再同期作業があるかどうかを調べることができます。「PENDING」という状況はそれがあることを示します。APPC 接続で新しい同期レベル 2 作業を実行できるかどうかを調べるには、INQUIRE CONNECTION XLNSTATUS コマンドを使用します。「XNOTDONE」という状況は、ログ名交換プロセスが、おそらくログ・データの喪失のために、正常に完了しなかったことを示します。

パートナー・システムがログ・データを喪失したことを CICS が検知した場合には、次のアクションが可能です。

1. なし。未解決の再同期作業がローカル・システムになければ、ログ・データの消失による影響はありません。
2. 未解決の再同期作業 (これには、通信が切れたときに未確定の状態であった UOW が含まれる場合がある) を調査のために保持する。
3. 未解決の再同期作業を削除する。未確定の UOW は、対応するトランザクション定義の ACTION オプションに従ってコミットまたはバックアウトされ、そのパートナーのための決定は破棄されます。

未解決の再同期作業がある場合には、CICS が (IPIC、MRO、および APPC の各接続に対し) アクション 2 と 3 のどちらを実行するかを制御することができます。

- 接続定義の XLNACTION オプションを使って、**自動的に**行う。パートナーのログ・データが失われたことが検知されたあと、すぐに再同期作業を削除する場合には、XLNACTION(FORCE) を使用します。
- SET UOW と SET CONNECTION PENDSTATUS(NOTPENDING) コマンドを使って、**手動で**行う。

APPC 接続に関する考慮事項

ログ名交換プロセスは、同期レベル 2 の会話にのみ影響を与えます。そのプロセスが失敗した場合には、その失敗がオペレーターによって解決されない限り、同期レベル 2 会話をそのリンクで行うことはできません。しかし、そのリンクの同期レベル 0 と同期レベル 1 の通信は失敗の影響を受けないので、引き続き正常に行われます。

接続定義の管理

このセクションでは、CICS Transaction Server for z/OS システム間の MRO 接続、IPICによる接続、および APPC 並列セッション接続の定義をどのように管理するかについて説明します。

重要:

他のタイプの接続に関する考慮事項については、 331 ページの『棚上げを完全にはサポートしない接続』を参照してください。

リモート・システムのためのリカバリー情報は、そのシステムの接続定義とはほとんど関係ありません。したがって、未解決かもしれないリカバリー情報とは無関係に接続定義を管理する (例えば、修正する) ことができます。しかし、場合によっては、接続定義に重要な情報が入っていることがあります。その場合には、その定義は、システム間のリカバリーが完了するまで、修正せずに保持されなければなりません。

CICS TS for z/OS システムへの MRO 接続と IPIC 接続

他の CICS Transaction Server for z/OS システムへの接続の場合は、その接続定義にリカバリー情報は含まれません。その接続のネット名を変えない限り、リカバリーとは無関係に接続を修正することができます。

コールド・スタートで接続定義が失われてしまった場合は、CEMT INQUIRE UOWLINK RESYNCSTATUS(UNCONNECTED) コマンドを使用して、前に接続されていたシステムのリカバリー情報が CICS に保持されているかどうかを調べることができます。このコマンドによって、UOW と消失した接続定義を対応付けるトークン (UOW リンク) が CICS にあるかどうかが分かります。UOW リンクがあれば、次のどちらかを行うことができます。

- UOW リンク属性に基づいて適切な接続定義を再インストールし、その接続を再び確立する。
- 対応付けられた UOW 情報が役に立たない場合には、SET UOWLINK(xxxxxxx) ACTION(DELETE) コマンドでその UOW リンクを削除する。UOW リンクを削除する前に、SET UOW コマンドで未確定の UOW のコミットまたはバックアウトを強制的に行う必要がある場合があります。

接続が既に廃棄されている場合にも、同じ UOWLINK コマンドを使用できます。

接続を廃棄する前に、INQUIRE CONNECTION RECOVSTATUS コマンドを使って、未解決のリカバリー情報がないかどうかを調べてください。未解決のリカバリー情報がある場合には、パートナーとの正常な再同期を確立する可能性がないときに限り、その接続を廃棄してください。この例外的な状況では、接続を廃棄する前に、SET CONNECTION UOWACTION コマンドを使って、未確定の作業単位についての強制的な決定を行うことができます。

CICS TS for z/OS システムへの APPC 並列セッション接続

VTAM 総称リソースのメンバーとして登録されていない CICS Transaction Server for z/OS システムにおける APPC 並列セッション接続は、リカバリー情報を含んでいないため、CICS TS for z/OS システムへの MRO 接続と同じ方法で管理することができます。

VTAM 総称リソースとの間の APPC 接続

CICS が VTAM 総称リソース・グループのメンバーである場合は、このローカル VTAM がアフィニティーをもち、それによってパートナーからの新しいバインドがこの同一のローカル・システムへ向けられることもあります。

VTAM によって保持されているこのアフィニティーは、パートナーとの再同期が必要になる可能性がある限り終わらせてはなりません。これを終わらせると、バインド (および、それ以後の再同期メッセージ) は、その総称リソースの別のメンバーへ送られるおそれがあります。ほとんどの場合は、APPC 接続静止プロトコルを使用してアフィニティーを自動的に終了させる方法が一番安全です。334 ページの『APPC 接続静止処理』を参照してください。

CICS は、パートナー・システムから既にログ名を受け取っていると、SET CONNECTION ENDAFFINITY コマンドを実行しません。パートナーがリカバリー可能作業を始め、再同期を開始できるためには、この条件が必要だからです。接続を廃棄すると、ログ名が分からなくなりますので、接続の廃棄も行われません。アフィニティーを終わらせる場合には、コールド・スタートを行う前の CICS のシャットダウンより*前に* 行う必要があります。コールド・スタートを行うと、ログ名が、対応する接続なしに復元されるからです。ログ名を除去せずにアフィニティーを終わらせると、ログ名交換が後で失敗することがあります。

アフィニティーとその終了の方法については、150 ページの『類縁性の終了』を参照してください。

接続定義の管理

総称リソースのメンバーにとって、接続定義は、ログ名と類縁性を安全に管理する唯一の方法です (INQUIRE と SET CONNECTION RECOVSTATUS コマンドの使用)。

接続を廃棄するには、そのリカバリー状況 (RECOVSTATUS) が NORECOVDATA でなければなりません。ローカル・システムにもそのパートナーにも相手に依存する未確定の作業単位がない場合は、SET CONNECTION RECOVSTATUS コマンドで接続のリカバリー状況を NORECOVDATA に設定することができます。簡単で安全なテストは、どちらのシステムも、相手との接続の状況が RECOVSTATUS (RECOVDATA) であってはならない、ということです。このテストにパスしたら、両方に対し SET CONNECTION NORECOVDATA を、総称リソース・メンバーに対し SET CONNECTION ENDAFFINITY を出すことができます。

棚上げを完全にはサポートしない接続

この項では、バックレベル・システムへの接続などに適用される例外について説明します。

前の項の情報は、他の CICS Transaction Server for z/OS システムへの MRO、IPIC、または APPC 並列セッション接続を使用することを前提にしています。つまり、ネットワークが、棚上げを完全にサポートする現行システムから構成されることを想定しています。これまでの情報のほとんどは、他のタイプの接続にも同様に適用されます。

LU6.1 接続

この項では、CICS TS for z/OS システムへの接続において、LU6.1 接続が APPC 並列セッション接続や MRO 接続とどのように異なるかを説明します。

リカバリーの機能とインターフェース

一部のリカバリー機能は、LU6.1 接続では使用できません。

- 棚上げは常にサポートされるわけではありません。
- リカバリー関連のコマンドとオプションにはサポートされないものがあります。
- 再同期はセッションごとに行われます。

棚上げサポートの制限

LU6.1 プロトコルには、作業単位が棚上げされたことをあるシステムから別のシステムへ知らせる機能はありません。LU6.1 セッションが含まれる UOW を棚上げできるのは、次の条件がすべて真の場合だけです。

- ローカル UOW にある LU6.1 セッションは 1 つだけである。
- その LU6.1 セッションはコーディネーターである。
- その LU6.1 セッションが未確定期間の間に失敗した。
- その LU6.1 セッションは最後のエージェントとのセッションである。

これらの条件のもとでは、LU6.1 パートナーは棚上げを通知される必要がないため、その UOW を棚上げすることが可能です。

他の条件のもとでは、未確定期間中に失敗した UOW のうち、LU6.1 セッションが関係しているものについて、一方的な決定を行います。WAIT(YES) が トランザクション定義に指定されていても効果はありません。WAIT(NO) が強制的に使用されます。

サポート対象外のコマンド

LU6.1 接続では、次のコマンドはサポートされません。

- INQUIRE CONNECTION PENDSTATUS
- INQUIRE CONNECTION RECOVSTATUS
- INQUIRE CONNECTION XLNSTATUS

SYNCPOINT ROLLBACK サポートの欠如

LU6.1 プロトコルには、UOW がバックアウトされたことを、その会話を終了せずに、あるシステムから別のシステムへ知らせる機能はありません。LU6.1 セッションが含まれる UOW で EXEC CICS SYNCPOINT ROLLBACK コマンドを出すと、ASP8 の異常終了となります。この異常終了をアプリケーション・プログラムで処理することはできません。

UOW のリソースはバックアウトされますが、トランザクションは続行できません。

セッションごとの再同期

APPC 並列セッション接続や CICS TS for z/OS-CICS TS for z/OS MRO 接続とは異なり、LU6.1 セッションはバインドされるに従って 1 つずつ再同期されます。したがって、再同期を必要とする UOW は、障害のあったセッションが再び接続されるまで再同期されません。

初期始動とコールド・スタート

LU6.1 接続定義には、リカバリーのために使用される順序番号が入っています。リカバリーが未解決の LU6.1 接続がある場合、CICS の初期始動やコールド・スタートを行うと、順序番号が失われるため、パートナー・システムは未解決の作業単位を再同期することができなくなります。

ログ名は使用されません。したがって、LU6.1 接続には、CEDA DEFINE CONNECTION コマンドの XLNACTION オプションは意味がありません。

接続定義の管理

リモート・システムのリカバリー情報は、そのシステムの接続定義と無関係に保管されるわけではありません。LU6.1 接続定義には、リカバリー時に使用される順序番号が格納されています。したがって、接続のリカバリー情報が未解決の場合には、その接続を修正したり、廃棄したりすることはしないでください。

非 CICS TS for z/OS システムへの APPC 接続

APPC リンクによって接続可能な非 CICS Transaction Server for z/OS システムの中には、棚上げをサポートせず、未確定期間にセッションの障害が起こると常に一方的なアクションをとるものがあります。

棚上げをサポートしないシステムへの通信では、必然的に、一方的な決定が行われることによってデータの整合性を損ねるおそれがあります。CICS は、棚上げをサポートしないシステムとサポートする システムを区別することはできません。したがって、サポートするシステムを作業単位のコーディネーターとして優先的に選択することはできません。

次の点に注意してください。

- 棚上げ解除が行われたときに、非 CICS TS for z/OS システムへの棚上げ解除の通知が遅れる場合があります。
- セッションは、通常の棚上げと再同期プロセスの一部として CICS またはそのパートナー・システムによってアンバインドされることがあります。

APPC 単一セッション接続

SINGLESESS(YES) と定義された接続では、通常の同期点プロトコルを使用することはできません。

機能シップが使用されている場合 (インバウンドまたはアウトバウンド)、CICS は作業単位の結果を通知します。これについては、「*CICS Family: Communicating from CICS on zSeries*」マニュアルの同期点指定 (LU 6.2)で説明しています。ただし、セッション障害が発生しても、再同期を行うことはできません。

CICS は、メッセージによって、作業単位が棚上げされたことをユーザーに通知します (棚上げ解除は通知しません)。

機能シップ要求に使用される接続がリモートとして定義されている (つまり、リモート領域によって所有されている) 場合には、そのリモート領域への接続は、この

リソース所有システムとのリカバリー・プロトコルを使用可能にするのであれば、並列セッション・リンクとして定義する必要があります。

APPC 接続静止処理

CICS Transaction Server for z/OS 領域との APPC 並列セッション接続が正常にシャットダウンされると、CICS は、パートナーと情報を交換して、接続が再開されたときに再同期が必要な可能性があるかどうかを調べます。

この情報交換のことを、接続静止プロトコル (CQP) と言います。

CICS は、次の条件をすべて満たしている場合には、再同期の必要はないと判断します。

- 接続がシャットダウンされている。
- 活動中のユーザー・セッションがない (CQP は SNASVCMG セッションを使用している)。SNASVCMG セッションがユーザー・セッションの前に非活動になった場合、CQP は行われません。
- CICS リカバリー・マネージャーに、未解決の同期点作業または接続の再同期作業の記録がない。

CQP が完了すると、CICS は、新しいログ名交換が発生するまでリカバリー可能作業を開始できないようにします。

CQP が再同期の必要がないことを判断すると、CICS は次のようにします。

- 接続のリカバリー状態を NORECOVDATA に設定します。
- CICS が総称リソース・グループのメンバーである場合は、VTAM が保持している類縁性を終了し、類縁性が終了したことを通知するメッセージを発行します。

CQP で障害が発生すると、CICS は再同期が必要になる可能性が生じたと見なします。本当に再同期しなければならないかどうかを判断するには、ここで記述されている手順従って、必要なアクションを手作業で実行します。また、接続を再取得してから再び解放し、CICS が CQP を再試行できるようにする方法もあります。

問題判別

このセクションでは、CICS のリカバリー・アクションを知らせるメッセージについて説明し、未確定や再同期の障害を解決する方法を例を挙げて示しています。この例の中では、既に説明した一部のコマンドの使用方法についても示しています。

CICS のリカバリー・アクションを知らせるメッセージ

通信障害が発生した場合、接続されているシステムは、それぞれ分散作業単位の各自のローカル部分を解決しますが、その際に相互のシステムで整合性が取れないことがあります。この可能性について警告するために、CICS 領域は、パートナーとの通信が切れると、UOW が未確定期間にあるセッションごとに DFHRMxxxx メッセージを出します。

このメッセージは、セッション障害、パートナーの障害、または緊急時再始動中に出されます。

接続が再度確立されると、関係するセッションごとに、UOW が棚上げ解除され、その状態が判別され、別のメッセージが出されます。LUTYPE6.1 会話では、これらのメッセージは開始プログラムの側だけに表示されます。

すべてのメッセージには次の情報が含まれており、メッセージを相互に関連付けることができます。

- 元の障害の日付と時刻
- トランザクション ID とタスク番号
- リモート・システムのネット名
- オペレーター ID
- オペレーター端末 ID
- ネットワーク全体の作業単位 ID
- ローカル作業単位 ID

システム間セッション障害およびリカバリーに関連する次のタイプのメッセージが生成されます。

- UOW のコーディネーターとの接続が切れたとき。メッセージについては、表 24 と 337 ページの表 25 を参照してください。
- WAIT(YES) がトランザクション定義に指定されており、棚上げが可能なとき。メッセージについては、表 24 を参照してください。
- WAIT(NO) が指定されているか、棚上げが不可能なとき。メッセージについては、 337 ページの表 25 を参照してください。
- UOW の従属との接続が切れたとき。メッセージについては、 337 ページの表 26 を参照してください。

詳細については、*CICS Messages and Codes*を参照してください。

表 24. WAIT(YES) セッション障害メッセージ: セッションと UOW のコーディネーターとの間に障害があります。WAIT(YES) がトランザクション定義で指定されており、棚上げが可能です。

各ステージ (1 および 2) において、CICS が発行するメッセージは、該当する状況によって異なります (カラム 2 および 4 に表示)。ステージ 1 は MRO のメッセージに適用され、ステージ 2 は IPIC と APPC の各メッセージに適用されます。

メッセージ・シーケンス	状況	発行されるメッセージ	メッセージの意味
ステージ 1	セッションに障害が発生した。	DFHRM0106	システム間のセッションに障害が発生した。リソース変更は、セッション・リカバリーまでコミットまたはバックアウトされない。
ステージ 1	システム障害または再始動	—	—
ステージ 2	セッション・リカバリーが正常に行われた	DFHRM0108	システム間のセッションがリカバリーされた。中断状態のリソース変更をコミット中。
ステージ 2	セッション・リカバリーが正常に行われた	DFHRM0109	システム間のセッションがリカバリーされた。中断状態のリソース変更をバックアウト中。

表 24. WAIT(YES) セッション障害メッセージ (続き): セッションと UOW のコーディネーターとの間に障害があります。WAIT(YES) がトランザクション定義で指定されており、棚上げが可能です。

各ステージ (1 および 2) において、CICS が発行するメッセージは、該当する状況によって異なります (カラム 2 および 4 に表示)。ステージ 1 は MRO のメッセージに適用され、ステージ 2 は IPIC と APPC の各メッセージに適用されます。

メッセージ・シーケンス	状況	発行されるメッセージ	メッセージの意味
ステージ 2	待ち時間を超過した、または SET UOW ACTION が発行された	DFHRM0104 DFHRM0105	次のテーブルを参照。
ステージ 2	SET CONNECTION NOTPENDING、XLNACTION (FORCE)、または NORECOVDATA が発行された	DFHRM0125 DFHRM0126	ローカル・リソースがコミットまたはバックアウトされた。
ステージ 2	ローカル・リソースのコールド・スタート後にセッションがリカバリーした。	DFHRM0209	UOW がバックアウトされた。
ステージ 2	ローカル・リソースのコールド・スタート後にセッションがリカバリーした	DFHRM0208	UOW がコミットされた。
ステージ 2	セッション・リカバリー・エラー — 例えば、パートナーがコールド・スタートした 1	DFHRM0112 DFHRM0113 DFHRM0115 DFHRM0116 DFHRM0118 DFHRM0119 DFHRM0121 DFHRM0122	システム間リカバリー・エラー。ローカル・リソース変更がコミットまたはバックアウトされた。
キー: 1. LU6.1 のみ			

表 25. WAIT(NO) セッション障害メッセージ： セッションと UOW のコーディネーターとの間に障害があります。WAIT(NO) がトランザクション定義に指定されているか、棚上げが不可能です。

各ステージ (1 および 2) において、CICS が発行するメッセージは、該当する状況によって異なります (カラム 2 および 4 に表示)。ステージ 1 は MRO のメッセージに適用され、ステージ 2 は IPIC と APPC の各メッセージに適用されます。

メッセージ・シーケンス	状況	発行されるメッセージ	メッセージの意味
ステージ 1	セッションに障害が発生した。	DFHRM0104 DFHRM0105	システム間のセッションに障害が発生した。リソース変更がコミット中またはバックアウト中で、パートナーとの同期が取れなくなる可能性がある。
ステージ 1	システム障害または再始動	—	—
ステージ 2	セッション・リカバリーが正常に行われた	DFHRM0110	システム間のセッションがリカバリーされた。リソース更新は同期化されている。
ステージ 2	セッション・リカバリーが正常に行われた	DFHRM0111	システム間のセッションがリカバリーされた。リソース更新は同期化されていない。
ステージ 2	SET CONNECTION NOTPENDING、XLNACTION (FORCE)、または NORECOVDATA が発行された	DFHRM0127	SET NOTPENDING が発行された。
ステージ 2	セッション・リカバリー・エラー — 例えば、パートナーがコールド・スタートした 1	DFHRM0112 DFHRM0113 DFHRM0115 DFHRM0116 DFHRM0118 DFHRM0119 DFHRM0121 DFHRM0122	ローカル・リソース変更がコミットまたはバックアウトされた。
キー: 1. LU6.1 のみ			

表 26. 従属セッション障害メッセージ： セッションと UOW の従属との間に障害があります。

各ステージ (1 および 2) において、CICS が発行するメッセージは、該当する状況によって異なります (カラム 2 および 4 に表示)。ステージ 1 は MRO のメッセージに適用され、ステージ 2 は IPIC と APPC の各メッセージに適用されます。

メッセージ・シーケンス	状況	発行されるメッセージ	メッセージの意味
ステージ 1	コーディネーターへのセッションの障害により UOW が棚上げされた	—	—

表 26. 従属セッション障害メッセージ (続き): セッションと UOW の従属との間に障害があります。

各ステージ (1 および 2) において、CICS が発行するメッセージは、該当する状況によって異なります (カラム 2 および 4 に表示)。ステージ 1 は MRO のメッセージに適用され、ステージ 2 は IPIC と APPC の各メッセージに適用されます。

メッセージ・シーケンス	状況	発行されるメッセージ	メッセージの意味
ステージ 1	セッションに障害が発生した。	DFHRM0107	システム間のセッションに障害が発生した。決定の通知がリモート・システムに届かない可能性がある。
ステージ 1	システム障害または再始動	—	—
ステージ 2	セッション・リカバリーが正常に行われた	DFHRM0135 DFHRM0148 1	システム間のセッションがリカバリーされた。リソース更新は同期化されている。
ステージ 2	セッション・リカバリーが正常に行われた	DFHRM0110	システム間のセッションがリカバリーされた。リソース更新は、リモート・システムにおける一方的な決定後に同期化される。
ステージ 2	セッション・リカバリーが正常に行われた	DFHRM0111 DFHRM0124	システム間のセッションがリカバリーされた。リソース更新は、リモート・システムにおける一方的な決定後に同期化されない。
ステージ 2	SET CONNECTION NOTPENDING、XLNACTION (FORCE)、または NORECOVDATA が発行された	DFHRM0127	SET NOTPENDING が発行された。
ステージ 2	セッション・リカバリー・エラー — 例えば、パートナーがコールド・スタートした 2	DFHRM0114 DFHRM0117 DFHRM0120 DFHRM0123	システム間セッション・リカバリー・エラー。リソース変更は同期化されない可能性がある。

キー:
1. DFHRM0124 および DFHRM0148 は、先行するセッション障害メッセージ (DFHRM0107) または棚上げが無いままで、発行される可能性がある。
2. LU6.1 のみ

問題判別の例

このセクションでは、未確定と再同期の失敗の解決方法の例を示します。

未確定の失敗の解決

このセクションでは、未確定期間中に失敗した作業単位を解決する方法の例を示します。

これには、次のコマンドを使用します。

- CEMT INQUIRE TASK
- CEMT INQUIRE UOWENQ
- CEMT INQUIRE UOW
- CEMT INQUIRE UOWLINK
- CEMT INQUIRE CONNECTION

ユーザーによると、かれらのタスクが領域 IYM51 で停止しました。CEMT INQUIRE TASK コマンドを実行すると、次のように表示されます。

```
INQUIRE TASK
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Tas(0000061) Tra(RTD1) Fac(S254) Sus Ter Pri( 001 )
    Sta(TO) Use(CICSUSER) Uow(AB1DF09A54115600) Hty(ENQUEUE ) Hva(TDNQ    )
 Tas(0000064) Tra(CEMT) Fac(S255) Run Ter Pri( 255 )
    Sta(TO) Use(CICSUSER) Uow(AB1DF16E3B78B403)
```

停止しているタスクは 61 で、トランザクション ID が RTD1 です。これは、一時データ・リソースに対して待機しています。CEMT INQUIRE UOWENQ コマンドを実行すると、次のように表示されます。

```
INQUIRE UOWENQ
STATUS:  RESULTS
 Uow(AB1DF0804B0F5801) Tra(RFS4) Tas(0000060) Ret Tsq Own
    Res(RMLTSQ                   ) Rle(008) Enq(00000000)
 Uow(AB1DF0804B0F5801) Tra(RFS4) Tas(0000060) Ret Dat Own
    Res(DCXISCG.IYLX1.RMLFILE    ) Rle(021) Enq(00000000)
 Uow(AB1DF0804B0F5801) Tra(RFS4) Tas(0000060) Act Tdq Own
    Res(QILR                     ) Rle(004) Enq(00000000)
 Uow(AB1DF0804B0F5801) Tra(RFS4) Tas(0000060) Act Tdq Own
    Res(QILR                     ) Rle(004) Enq(00000000)
 Uow(AB1DF09A54115600) Tra(RTD1) Tas(0000061) Act Tdq Wai
    Res(QILR                     ) Rle(004) Enq(00000000)
```

この場合、タスク 61 が唯一の待機者でタスク 60 が唯一の所有者なので、キュー項目所有者を識別するのは簡単です。タスク 60 が所有しているキュー項目は、タイプ TSQUEUE が 1 つ、タイプ DATASET が 1 つ、タイプ TDQ が 2 つです。これらのキュー項目はそれぞれ、リソース RMLTSQ、DCXISCG.IYLX1.RMLFILE、QILR に対するものです。

CEMT INQUIRE TASK 画面によると、タスク 60 が終了しました。CEMT INQUIRE UOW コマンドを使用すれば、終了したタスクだけでなく、アクティブ状態にあるタスクについても、それらに対応する作業単位状況の情報を戻すことができます。

```
INQUIRE UOW
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Uow(AB1DD0FE5F219205) Inf Act Tra(CSSY) Tas(0000005)
    Age(00002569)                          Use(CICSUSER)
 Uow(AB1DD0FE5FEF9C00) Inf Act Tra(CSSY) Tas(0000006)
    Age(00002569)                          Use(CICSUSER)
 Uow(AB1DD0FE7FB82600) Inf Act Tra(CSTP) Tas(0000008)
    Age(00002569)                          Use(CICSUSER)
 Uow(AB1DD98323E1C005) Inf Act Tra(CSNC) Tas(0000018)
    Age(00000282)                          Use(CICSUSER)
 Uow(AB1DF0804B0F5801) Ind Shu Tra(RFS4) Tas(0000060)
    Age(00002699) Ter(S255) Netn(IGCS255 ) Use(CICSUSER) Con Lin(IYM52   )
 Uow(AB1DF09A54115600) Inf Act Tra(RTD1) Tas(0000061)
    Age(00002673) Ter(S254) Netn(IGCS254 ) Use(CICSUSER)
 Uow(AB1DF0B309126800) Inf Act Tra(CSNE) Tas(0000021)
    Age(00002647)                          Use(CICSUSER)
 Uow(AB1DF16E3B78B403) Inf Act Tra(CEMT) Tas(0000064)
    Age(00002451) Ter(S255) Netn(IGCS255 ) Use(CICSUSER)
```

CEMT INQUIRE UOW コマンドのフィルターを使用すれば、特定のタスクの UOW を表示することができます。例えば、CEMT INQUIRE UOW TASK(60) と指定すれば、次のように表示されます。

```
INQUIRE UOW TASK(60)
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Uow(AB1DF0804B0F5801) Ind Shu Tra(RFS4) Tas(0000060)
    Age(00002699) Ter(S255) Netn(IGCS255 ) Use(CICSUSER) Con Lin(IYM52   )
```

注: CEMT INQUIRE UOW コマンドには、UOW のフィルターとしてワイルドカードが使用できます。例えば、CEMT INQUIRE UOW(*5801) と指定すれば、UOW AB1DF0804B0F5801 だけの情報が戻されます。

特定の UOW の詳細情報を見る場合は、カーソルをその UOW の横に置いて、ENTER を押します。

```
INQUIRE UOW
RESULT - OVERTYPE TO MODIFY
  Uow(AB1DF0804B0F5801)
  Uowstate( Indoubt )
  Waitstate(Shunted)
  Transid(RFS4)
  Taskid(0000060)
  Age(00002801)
  Termid(S255)
  Netname(IGCS255)
  Userid(CICSUSER)
  Waitcause(Connection)
  Link(IYM52)
  Sysid(ISC2)
  Netuowid(..GBIBMIYA.IGCS255 .0......)
```

問題の UOW は AB1DF0804B0F5801 です。Uowstate (UOW 状態) は **Shunted** (棚上げされている) です。つまり、同期点処理が既に延期されており、ロックは、リソースの整合性が確保されるまで保存されます。この場合、UOW は **Indoubt** (未確定) の状態で棚上げされています。つまり、タスク 60 は、同期点処理の未確定期間中に失敗しました。

UOW が棚上げされている理由は、Waitcause (待機理由) に示されます。この場合は、**Connection** (接続) です。この UOW は、接続 **ISC2** の障害のために棚上げされています。その接続に対応する Link (つまり、ネット名) は **IYM52** です。

CEMT INQUIRE UOWLINK コマンドを実行すると、分散 UOW に関連する接続情報が以下のように表示されます。

```
INQUIRE UOWLINK
STATUS:  RESULTS
 Uowl(02EC0011) Uow(AB1DF0804B0F5801) Con Lin(IYM52   )
    Coo Appc Una Sys(ISC2)               Net(..GBIBMIYA.IGCS255 .0......)
```

Link の詳細情報を見るには、カーソルを UOW の横に置いて、ENTER を押します。

```
INQUIRE UOWLINK
RESULT
  Uowlink(02EC0011)
  Uow(AB1DF0804B0F5801)
  Type(Connection)
  Link(IYM52)
  Action(              )
  Role(Coordinator)
  Protocol(Appc)
  Resyncstatus(Unavailable)
  Sysid(ISC2)
  Rmiqfy()
  Netuowid(..GBIBMIYA.IGCS255 .0......)
```

この例で、システム IYM52 への接続 ISC2 が、この UOW に対する同期点 **Coordinator** (コーディネーター) であることが分かります。Resyncstatus (再同期状況) が **Unavailable** (使用不能) ですので、その接続は現在のところ獲得されていません。

CEMT INQUIRE CONNECTION コマンドを実行すると、このことが確認できます。

```
I INQUIRE CONNECTION
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Con(ISC2) Net(IYM52   )     Ins Rel Vta Appc         Rec
 Con(ISC4) Net(IYM54   )     Ins Acq Vta Appc     Xok Unk
 Con(ISC5) Net(IYM55   )     Ins Acq Vta Appc     Xok Unk
```

接続 ISC2 の情報をさらに見る場合には、カーソルをその接続の横に置き、ENTER を押します。

```
INQUIRE CONNECTION
RESULT
  Connection(ISC2)
  Netname(IYM52)
  Pendstatus( Notpending )
  Servstatus( Inservice )
  Connstatus( Released )
  Accessmethod(Vtam)
  Protocol(Appc)
  Purgetype(              )
  Xlnstatus()
  Recovstatus( Recovdata )
  Uowaction(              )
  Grname()
  Membername()
  Affinity(              )
  Remotesystem()
  Rname()
  Rnetname()
```

これによると、接続 ISC2 は Recovstatus (回復状況) が **Recovdata** (回復データ) で、**Released** (解放) されています。つまり、この接続で再同期が未解決です。

この段階で、システム IYM52 への接続を獲得することが可能なら、再同期が自動的に行われ、UOW AB1DF0804B0F5801 の棚上げが解除され、その待機がすべて解放されます。これによって、タスク 61 の完了が可能になります。しかし、その接続を獲得することができないのであれば、その UOW の棚上げを解除し、通常の再同期の指定を変更することができます。その UOW をコミットするか、バックアウトするかを決めるには、システム IYM52 の対応する UOW を照会する必要があります。システム IYM52 で CEMT INQUIRE UOW コマンドを実行すると、次のように表示されます。

```
INQUIRE UOW
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Uow(AB1DD01221BA6E01) Inf Act Tra(CSSY) Tas(0000005)
    Age(00003191)                         Use(CICSUSER)
 Uow(AB1DD0122276C201) Inf Act Tra(CSSY) Tas(0000006)
    Age(00003191)                         Use(CICSUSER)
 Uow(AB1DD01248A7B005) Inf Act Tra(CSTP) Tas(0000008)
    Age(00003191)                         Use(CICSUSER)
 Uow(AB1DD9057B8DD800) Inf Act Tra(CSNC) Tas(0000018)
    Age(00000789)                         Use(CICSUSER)
 Uow(AB1DF0805E76B400) Com Wai Tra(CSM3) Tas(0000079)
    Age(00003003) Ter(-AC3) Netn(IYM51   ) Use(CICSUSER) Wai
 Uow(AB1DF0B2FDD36400) Inf Act Tra(CSNE) Tas(0000019)
    Age(00003024)                         Use(CICSUSER)
 Uow(AB1DF15502238000) Inf Act Tra(CEMT) Tas(0000086)
    Age(00002853) Ter(S25C) Netn(IGCS25C ) Use(CICSUSER)
```

端末から開始されたトランザクションの場合には、CEMT INQUIRE UOW コマンドのフィルターとして **Netuowid** (ネットワーク UOW ID) を使用できるので、特定の端末から実行されたトランザクションに関連する UOW だけを表示することができます。この場合、システム IYM51 のタスク 60 は端末 S255 から実行されました。システム IYM51 の UOW AB1DF0804B0F5801 の Netuowid には、端末 S255 の luname (LU 名) が入っています。

Netuowid は、単一の分散作業単位内で接続されているすべての UOW で同じなので、Netuowid はこれらの UOW を一緒に結び付けるときに便利です。この例では、コマンド CEMT INQUIRE UOW NETUOWID(*S255*) によって、CEMT INQUIRE UOW コマンドは次のようにフィルターに掛けられます。

```
INQUIRE UOW NETUOWID(*S255*)
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Uow(AB1DF0805E76B400) Com Wai Tra(CSM3) Tas(0000079)
    Age(00003003) Ter(-AC3) Netn(IYM51   ) Use(CICSUSER) Wai
```

UOW AB1DF0805E76B400 の詳細情報を見るには、カーソルを UOW の横に置いて、ENTER を押します。

```
INQUIRE UOW
RESULT - OVERTYPE TO MODIFY
  Uow(AB1DF0805E76B400)
  Uowstate( Commit )
  Waitstate(Waiting)
  Transid(CSM3)
  Taskid(0000079)
  Age(00003003)
  Termid(-AC3)
  Netname(IYM51  )
  Userid(CICSUSER)
  Waitcause(Waitforget)
  Link(     )
  Sysid(    )
  Netuowid(..GBIBMIYA.IGCS255 .0......)
```

UOW AB1DF0805E76B400 は、機能シップで使用されるミラー・タスクに関連付けられていることがわかります。Uowstate (UOW 状況) **Commit** (コミット) は、そのUOW が既にコミットされていることを、Waitstate (待機状況) **Waiting** (待機) は、決定が IYM51 にまだ通知されていないため、待機中であることをそれぞれ意味します。システム IYM51 のリソースの更新が IYM52 の更新と、この分散作業単位については同期することがわかったので、システム IYM51 の棚上げされているUOW を安心してコミットすることができます。CEMT SET UOW コマンドを使用すれば、この棚上げされた UOW をコミットできます。棚上げされた UOW がコミットされると、その ENQ が解放され、タスク 61 の続行が可能になります。

考えられるもう 1 つのシナリオは、IYM52 が使用不能の場合です。IYM52 が使用可能になるまで待つのが実際的でなく、データの整合性に対するリスクを受け入れる準備ができている場合は、CEMT SET CONNECTION コマンドを使用して、接続 ISC2 の障害によって未確定の状態で失敗したすべての UOW についてコミット、バックアウト、または強制的な決定を行うことができます。

この例で、トランザクション RTD1 は、一時データ・キューに対する ENQUEUE のために延期されています。そのキューのアクティブなロックは UOW AB1DF0804B0F5801 (未確定の状態で失敗) によって所有されていました。タスクがこのようにして延期されるのを防ぐには、WAITACTION オプションを **REJECT** (WAITACTION のデフォルト) に設定して、一時データ・キューを定義することもできます。このようにした場合、そのキューを更新するタスクが未確定状態で失敗すると、保存ロックがその棚上げされた UOW によって保持されます。次に、その保存ロックに対する要求があると、LOCKED 条件によって拒否されます。

CEMT コマンドの詳細については、「*CICS Supplied Transactions*」の CEMT マスター端末を参照してください。

再同期の失敗の解決

この項では、再同期の失敗をどのように解決するかの例を示します。

この項では、再同期の失敗をどのように解決するかの例を示します。これには、次のコマンドを使用します。

- CEMT INQUIRE CONNECTION
- CEMT INQUIRE UOWLINK
- CEMT INQUIRE UOW

- CEMT INQUIRE UOWENQ
- SET CONNECTION NOTPENDING

ユーザーは、システム IYLX1 のトランザクション (システム IYLX4 への機能シップ要求が含まれます) が、「SYSIDERR」でエラーになっていることを報告しました。システム IYLX1 で CEMT INQUIRE CONNECTION コマンドを実行すると、次のように表示されます。

```
INQUIRE CONNECTION
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Con(ISC2) Net(IYLX2   )     Ins Rel Vta Appc         Unk
 Con(ISC4) Net(IYLX4   ) Pen Ins Acq Vta Appc     Xno Unk
 Con(ISC5) Net(IYLX5   )     Ins Acq Vta Appc     Xok Unk
```

図 89. CEMT INQUIRE CONNECTION - システム IYLX1 が所有する接続

この接続の詳細情報を見るには、カーソルを ISC4 の行に置いて、ENTER を押します。図 90 を参照してください。

```
INQUIRE CONNECTION
RESULT - OVERTYPE TO MODIFY
  Connection(ISC4)
  Netname(IYLX4)
  Pendstatus( Pending )
  Servstatus( Inservice )
  Connstatus( Acquired )
  Accessmethod(Vtam)
  Protocol(Appc)
  Purgetype(              )
  Xlnstatus(Xnotdone)
  Recovstatus( Nrs )
  Uowaction(              )
  Grname()
  Membername()
  Affinity(              )
  Remotesystem()
  Rname()
  Rnetname()
```

図 90. CEMT INQUIRE CONNECTION - 接続 ISC4 の詳細

接続 ISC4 の Connstatus (接続状況) は **Acquired** (獲得済み) ですが、Xlnstatus (XLN 状況) は **Xnotdone** (未完) です。この接続のログ名交換 (XLN) フローは、正常に完了していません。(CICS システムは相互に接続すると、ログ名を交換します。これらのログ名は再同期を行う前に検査されます。ログ名交換の失敗は、再同期が不可能であることを意味します。) 機能シップの場合、接続の失敗は SYSIDERR になります。ログ名が正しく交換されるまでは、この接続で同期レベル 2 の会話を行うことはできません (この制限は MRO 接続には適用されません)。

ログ名交換の失敗理由は、CSMT ログに書き込まれます。CICS Transaction Server for z/OS システムにおける障害の発生原因は、次のとおりです。

- CICS TS for z/OS システム、またはパートナーの初期始動 (START=INITIAL)

  注: CICS TS for z/OS システムのコールド・スタート (START=COLD) では、再同期の情報 (ログ名を含む) が保存されるので、ログ名交換が失敗することはありません。

- CEMT SET CONNECTION NORECOVDATA コマンドの使用

- システムの論理エラーや操作エラー

接続 ISC4 の Pendstatus (保留状況) は **Pending** (保留) ですが、これは、その接続で再同期の作業が未解決だが、ログ名交換の失敗のためこの作業を完了できないことを意味します。

この段階で、同期が失われるのを心配する必要がない場合は、SET CONNECTION NOTPENDING コマンドを発行してすべての未確定 UOW を強制的にコミットまたはバックアウトすることも可能です。しかし、保留状態をクリアする前に、存在する未解決の再同期作業をコマンドで調べることができます。

CEMT INQUIRE UOWLINK コマンドを使用すれば、システム IYLX4 との再同期が必要な UOW の情報を表示することができます。

```
INQUIRE UOWLINK LINK(IYLX4)
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Uowl(016C0005) Uow(ABD40B40C1334401) Con Lin(IYLX4   )
    Coo Appc Col Sys(ISC4)            Net(..GBIBMIYA.IYLX150 M. A....)
 Uowl(01680005) Uow(ABD40B40C67C8201) Con Lin(IYLX4   )
    Coo Appc Col Sys(ISC4)            Net(..GBIBMIYA.IYLX151 M. F@b..)
 Uowl(016D0005) Uow(ABD40B40DA5A8803) Con Lin(IYLX4   )
    Coo Appc Col Sys(ISC4)            Net(..GBIBMIYA.IYLX156 M. .!h..)
```

図 91. CEMT INQUIRE UOWLINK - システム IYLX4 との再同期が必要な UOW

各 UOW リンクの情報の詳細を見るには、その横にカーソルを置いて ENTER を押します。例えば、UOW リンク 016C0005 の詳細情報は次のように表示されます。

```
I UOWLINK LINK(IYLX4)
RESULT - OVERTYPE TO MODIFY
  Uowlink(016C0005)
  Uow(ABD40B40C1334401)
  Type(Connection)
  Link(IYLX4)
  Action(             )
  Role(Coordinator)
  Protocol(Appc)
  Resyncstatus(Coldstart)
  Sysid(ISC4)
  Rmiqfy()
  Netuowid(..GBIBMIYA.IYLX150 M. A....)
```

図 92. CEMT INQUIRE UOWLINK - UOW リンク 016C0005 の詳細情報

Resyncstatus (再同期状況) の **Coldstart** (コールド・スタート) は、システム IYLX4 が新しいログ名で開始されたことを示します。UOW リンクの Role (役割) が **Coordinator** (コーディネーター) ですので、IYLX4 はこの同期点のコーディネーターです。

次に、CEMT INQUIRE UOW LINK(IYLX4) コマンドを使用すると、未確定の状態にあり、システム IYLX4 をコーディネーター・システムにしているすべての UOW が表示されます。

```
INQUIRE UOW LINK(IYLX4)
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Uow(ABD40B40C1334401) Ind Shu Tra(RFS1) Tas(0000674)
    Age(00003560) Ter(X150) Netn(IYLX150 ) Use(CICSUSER) Con Lin(IYLX4   )
 Uow(ABD40B40C67C8201) Ind Shu Tra(RFS1) Tas(0000675)
    Age(00003465) Ter(X151) Netn(IYLX151 ) Use(CICSUSER) Con Lin(IYLX4   )
 Uow(ABD40B40DA5A8803) Ind Shu Tra(RFS1) Tas(0000676)
    Age(00003462) Ter(X156) Netn(IYLX156 ) Use(CICSUSER) Con Lin(IYLX4   )
```

図 93. CEMT INQUIRE UOW LINK(IYLX4) - IYLX4 をコーディネーターにしているすべての UOW

各未確定 UOW の詳細情報を見るには、その行で ENTER を押します。例えば、UOW ABD40B40C1334401 の詳細情報は次のように表示されます。

```
INQUIRE UOW LINK(IYLX4)
RESULT - OVERTYPE TO MODIFY
  Uow(ABD40B40C1334401)
  Uowstate( Indoubt )
  Waitstate(Shunted)
  Transid(RFS1)
  Taskid(0000674)
  Age(00003906)
  Termid(X150)
  Netname(IYLX150)
  Userid(CICSUSER)
  Waitcause(Connection)
  Link(IYLX4)
  Sysid(ISC4)
  Netuowid(..GBIBMIYA.IYLX150 M. A....)
```

図 94. CEMT INQUIRE UOW LINK(IYLX4) - UOW ABD40B40C1334401 の詳細情報

この UOW をシステム IYLX4 によって再同期させることはできません (状況が **Indoubt** (未確定) と表示されています)。これは、IYLX4 で実行された関連する UOW がコミットされたのか、バックアウトされたのかが IYLX4 にはわからないためです。

CEMT INQUIRE UOWENQ コマンドを使用すれば、すべての棚上げされた UOW によってロックされているリソース (保存ロックを所有するリソース) を表示することができます。

```
INQUIRE UOWENQ OWN RETAINED
STATUS:  RESULTS
 Uow(ABD40B40C1334401) Tra(RFS1) Tas(0000674) Ret Tsq Own
    Res(RFS1X150                 ) Rle(008) Enq(00000008)
 Uow(ABD40B40C67C8201) Tra(RFS1) Tas(0000675) Ret Tsq Own
    Res(RFS1X151                 ) Rle(008) Enq(00000008)
 Uow(ABD40B40DA5A8803) Tra(RFS1) Tas(0000676) Ret Tsq Own
    Res(RFS1X156                 ) Rle(008) Enq(00000008)
```

図 95. CEMT INQUIRE UOWENQ - すべての棚上げされた UOW によってロックされたリソース

INQUIRE UOWENQ コマンドにフィルターを指定すれば、特定の UOW によって所有されるキュー項目だけを表示することができます。例えば、フィルター操作によって UOW ABD40B40C1334401 が所有する ENQ だけを表示するには、次のようにします。

```
INQUIRE UOWENQ OWN UOW(*4401)
STATUS:  RESULTS
 Uow(ABD40B40C1334401) Tra(RFS1) Tas(0000674) Ret Tsq Own
    Res(RFS1X150                 ) Rle(008) Enq(00000008)
```

図 96. CEMT INQUIRE UOWENQ - UOW ABD40B40C1334401 によってロックされたリソース

この UOWENQ の詳細情報を見るには、カーソルをその横に置いて、ENTER を押します。

```
INQUIRE UOWENQ OWN UOW(*4401)
RESULT
  Uowenq
  Uow(ABD40B40C1334401)
  Transid(RFS1)
  Taskid(0000674)
  State(Retained)
  Type(Tsq)
  Relation(Owner)
  Resource(RFS1X150)
  Rlen(008)
  Enqfails(00000008)
  Netuowid(..GBIBMIYA.IYLX150 M. A....)
  Qualifier()
  Qlen(000)
```

図 97. CEMT INQUIRE UOWENQ - UOWENQ ABD40B40C1334401 の詳細情報

そのアプリケーションの知識があれば、これで、ロックされたリソースの更新をコミットするか、バックアウトするかが決定できるかもしれません。UOW ABD40B40C1334401 の場合には、ロックされたリソースは一時記憶域キュー RFS1X150 です。このリソースの ENQFAILS 値は 8 になっています。これは、このキューが保存状態で保持されていたために、**LOCKED** の応答を受け取ったタスクの数です。

SET UOW コマンドを使用すれば、棚上げされた UOW によって行われた未確約の更新についてコミット、バックアウト、または強制的な決定を行うことができます。次に SET CONNECTION(ISC4) NOTPENDING コマンドを使って保留条件をクリアし、同期レベル 2 の会話 (これには、前に SYSIDERR で失敗した機能シップ要求も含む) ができるようにする必要があります。

CONNECTION 定義の XLNACTION オプションを使用すれば、ログ名交換の失敗の結果を制御することができます。この例では、接続 ISC4 の XLNACTION は **KEEP** です。これは次のことを意味します。

- システム IYLX1 の棚上げされた UOW は、コールド/ウォーム・ログの IYLX4 との不一致のあと保持される。
- IYLX1 と IYLX4 の間の APPC 接続は、保留条件が解決されるまで、機能シップ要求に使用できない。

接続 ISC4 に対し XLNACTION が **FORCE** であれば、コールド / ウォーム・ログの不一致が起こると、SET CONNECTION NOTPENDING コマンドが自動的に出されます。この結果、棚上げされた UOW は、関連するトランザクション定義の ACTION オプションに従って強制的にコミットまたはバックアウトされます。そうすれば、接続 ISC4 が **Pending** 状況になることはなかったはずです。しかし、XLNACTION を FORCE に設定すると、ログ名交換の失敗のあとに、棚上げされた

UOW を調べることができないので、XLNACTION を KEEP に設定するよりも、データの整合性の点でリスクが大きいと言えます。

第 27 章 相互通信と XRF

CICS Transaction Server for z/OS の拡張回復機能 (XRF) の詳細については、「*CICS/ESA® 3.3 CICS XRF Guide*」を参照してください。この章では、ISC と MRO の各セッションに適用される XRF の機能について説明します。

この章で説明するリンク定義の詳細については、165 ページの『第 13 章 リモート・システムへの接続の定義方法』を参照してください。

MRO および ISC の各セッションは、XRF 可能ではありません。これは、これらのセッションが、代替 CICS システムとのバックアップ・セッションをもつことができないためです。

AUTOCONNECT オプションをリンク定義に使用すると、代替 CICS システムによる引き継ぎのあとに、CICS にセッションの再確立を試行させることができます。

また、いくつかの ISC セッション・タイプのバインドまたはアンバインド状況を追跡することができます。このような場合、CICS は、AUTOCONNECT の指定に関係なく、バインドされたセッションの再獲得を試行することができます。

いずれの場合も、セッションの再確立を試行するタイミングは、AUTCONN システム初期設定パラメーターによって制御されます。システム初期設定パラメーターについては、「*CICS System Definition Guide*」の 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。

この章の残りの部分には、次のトピックが含まれています。

- 『MRO セッション』
- 『LUTYPE6.1 セッション』
- 350 ページの『単一セッション APPC デバイス』
- 350 ページの『並列 APPC セッション』
- 350 ページの『アプリケーション・プログラムへの影響』

MRO セッション

MRO セッションの状況を追跡することはできません。代替 CICS システムによる引き継ぎに続いて、CICS は、CONNECTION 定義の INSERVICE オプションに指定された値に従って、MRO セッションを再確立しようとします。

LUTYPE6.1 セッション

引き継ぎが行われると、CICS は、次のいずれかの場合に LUTYPE6.1 セッションを再確立しようとします。

1. SESSIONS 定義の AUTOCONNECT オプションに YES が指定してある場合。

© Copyright IBM Corp. 1977, 2009

2. セッションが追跡されて、引き継ぎが起こるとバインドされる場合。LUTYPE6.1 セッションの状況は、RECOVOPTION(NONE) が SESSIONS 定義に指定されている場合を除いて追跡されます。

単一セッション APPC デバイス

引き継ぎが行われると、CICS は、次の場合に単一 APPC セッションを再確立しようとします。

1. SESSIONS 定義または TYPETERM 定義の AUTOCONNECT オプションに YES が指定してある場合。
2. 追跡されているセッションが、活動 CICS に障害が起きたときにバインドされる場合。単一 APPC セッションは、RECOVOPTION(NONE) が SESSIONS 定義または TYPETERM 定義 (どの形式の定義が使用されるかによって) に指定されている場合を除いて追跡されます。RECOVOPTION は 5 つの値をとることができますが、ISC の場合は、NONE (追跡なし) か、または残りのオプションのうち *どれか 1 つ (追跡)* のいずれかになります。

並列 APPC セッション

引き継ぎが行われると、CICS は、次の場合に LU サービス管理セッションを再確立しようとします。

- CONNECTION 定義の AUTOCONNECT オプションが YES または ALL を指定する場合。
- 追跡されているセッションが、活動 CICS の障害発生時にバインドされる場合。この場合は、LU サービス管理セッション (SNASVCMG) だけを追跡することができます。ユーザー・セッションを追跡することはできません。

LU サービス管理セッションが再確立されると、CICS はすぐに、自動接続を指定するすべてのモードグループのセッションを確立しようとします。

アプリケーション・プログラムへの影響

相互通信機能を使用するアプリケーション・プログラムには、リモート CICS システムにおける引き継ぎとセッション障害を区別することができません。

第 28 章 相互通信と VTAM 持続セッション

VTAM 持続セッションをサポートすると、システム間通信に影響を与えます。

VTAM 持続セッションのための CICS サポートの詳細については、「*CICS Recovery and Restart Guide*」の持続セッションを使用したリカバリーを参照してください。

VTAM 持続セッションを使用すると、APPC プロトコルを使用する DTP アプリケーションが影響を受けます。この影響については、「*CICS Distributed Transaction Programming Guide*」のAPPC セッションの DTP 会話における、VTAM 持続セッション・サポートの影響で説明しています。

関連概念

165 ページの『第 13 章 リモート・システムへの接続の定義方法』
CICS 領域間、または CICS 領域から非 CICS システムへのさまざまなタイプの接続を定義および管理することができます。

持続セッションのサポートと XRF との比較

XRF を使用すると、アクティブな CICS システムに障害があったときに、部分的に初期設定された代替 CICS システムに制御を引き継がせることができます。この XRF の代わりに、VTAM 持続セッション・サポートを使用することができます。

持続セッションでは、CICS に障害があったときに、CICS セッションを再バインドするためのネットワーク・フローを必要とせずに、その場で再始動することができます。同じシステムに XRF と CICS 持続セッション・サポートの両方を指定することはできません。

XRF では、活動システムと代替システムを使用することによってシステムの可用性が高まります。またシステムの高い可用性とバックアップ・セッションを利用することによって、ユーザーに対して高い可用性を実現することができます。活動システムと代替システムのペアは、一部のデータ・セット (補助トレース・データ・セットやダンプ・データ・セットなど) について、独自のバージョンが必要です。

持続セッション・サポートでは、それを 1 つのシステムの代わりに再始動することによってシステムの可用性が高まります。また、システムの高い可用性と持続セッションを利用することによって、エンド・ユーザーに対して高い可用性を実現することができます。必要なデータ・セットおよびシステムは 1 つだけです。XRF と比較して、持続セッション・サポートには次の利点があります。

- MRO、LU6.1、LU0 のパイプライン・セッションを除き、すべてのセッション・タイプをサポートする。XRF は、ローカル端末、MRO、ISC (LU6.1 か LU6.2) のセッションをサポートしません。
- XRF よりもインストールと管理が容易である。1 つのシステムしか必要ない。

単一ノード持続セッション・サポートでは、VTAM または z/OS に障害が発生した際に、セッションが保存されません。マルチノード持続セッション・サポートには

© Copyright IBM Corp. 1977, 2009

この機能があります。マルチノード持続セッション・サポートに相当する機能がない場合で、これらの障害が発生した際に速やかに再始動する必要があるときには、持続セッションの代わりに XRF を使用します。

相互接続された CICS の環境、リカバリー、および再始動

CICS システムは、MRO、LU6.1、LU6.2 の接続とセッションを使用して相互に接続することができます。リカバリーと再始動の動作は、セッション・タイプ、および VTAM 持続セッション・サポートが使用されているかどうかによって異なります。

MRO セッション

MRO 接続を、CICS の障害とそのあとの緊急時再始動にまたがって持続させることはできません。

LU6.1 セッション

多重システム環境で CICS 領域に障害が発生した場合、それが緊急時再始動によって再始動されるか、持続セッション遅延間隔が終了するまで、それに接続されているすべての LU6.1 セッションはリカバリー保留状態におかれます。上のどちらかが起こると、それらの LU6.1 セッションはアンバインドされます。それらを再び使用できるようにするには、再獲得する必要があります。

持続セッション・サポートが使用されるかどうかによって、システム・プログラマーやオペレーターに示される CICS 障害の症状は多少異なります。持続セッション・サポートがないシステムでは、すべての LU6.1 セッションは、障害発生後、ただちにアンバインドされます。

持続セッション・サポートがあるシステムでは、LU6.1 セッションは、緊急時再始動 (持続セッション遅延間隔内に行われた場合) か、持続セッション遅延間隔が終了するまでアンバインドされません。その結果、これらのセッションがアンバインドされるまでに時間がかかることがあります。

LU6.2 セッション

異なる CICS システムを接続する LU6.2 セッションは、1 つまたは複数のシステムの障害と、そのあとの持続セッション遅延間隔内での緊急再始動にまたがって持続します。

しかし、持続セッションがシステムでサポートされていても、ある状況ではこれらのセッションがアンバインドされます。下記のセッションは、たとえ持続セッションとして定義されていても、CICS の障害と緊急再始動のあとでアンバインドされます。

- カタログ項目が見つからないセッション。
  - 自動インストールされた LU6.2 並列セッション
  - 自動インストールされた LU6.2 単一セッションのうち、BIND 要求によって開始されたもの
  - AIRDELAY システム初期設定パラメーターがゼロに設定されている場合、自動インストールされた LU6.2 単一セッションのうち、VTAM CINIT 要求によ

って開始されたもの。(AIRDELAY は、緊急再始動から、セッション中状態にない自動インストールされた端末項目が削除されるまでに経過する間隔を指定します。)

つまり、アンバインドされない自動インストールの LU6.2 セッションは、CINIT 要求によって開始された単一セッションで、AIRDELAY がゼロより大きい場合だけです。

- 障害のある TOR への LU6.2 接続の 1 つまたは複数のセッションで、その TOR に対し AOR から ATI 要求を機能シップしている場合は、その接続のすべて のセッション。これは、要求が、その TOR によって所有される端末に関連付けられているからです。ATI によって開始されるトランザクション・ルーティングについては、79 ページの『ATI によって開始されたトランザクションの従来のルーティング』で説明しています。
- LU6.2 接続の 1 つまたは複数のセッションで、CRTE によるトランザクション・ルーティングが行われていても、この障害が発生した時点で進行している会話がない場合は、その接続のすべて のセッション。会話が進行している場合には、障害のある CICS のパートナーに DEALLOCATE(ABEND) が送られます。

LU6.2 相互接続環境で CICS の障害が発生し、それが持続セッション遅延間隔内に緊急時再始動された場合、その接続の一方の側が、その障害が発生する前にゼロにするための CNOS 要求を出しているか、または障害発生時にその接続で CNOS 折衝が進行中でなかった限り、トランザクション CLS1 (CNOS) は実行されません。

障害のあるシステムは、持続セッション遅延間隔内での緊急再始動のあと、できるだけ早くトランザクション CLS2 (XLN、ログ名交換) を実行します。CLS2 が実行されない限り、どちらの接続システムでも、それ以後の同期レベル 2 の会話を処理することはできません。

第 7 部 システム間環境におけるデータ変換

CICS Transaction Server for z/OS のアプリケーション・プログラムでは、通常 EBCDIC フォーマットを使用して文字データが表されます。CICS がリモート・システムとデータを交換する際、これらのシステムでは多くの場合、文字データを表記するのに ASCII または Unicode が使用されます。

文字データを表記するのに異なるフォーマットを使用する複数のシステムで交換されるデータは、通常それらの異なるフォーマット間で変換を行う必要があります。

注: データ変換にチャネルを使用している場合は、このトピックではなく、チャネルを使用したデータ変換を参照してください。

© Copyright IBM Corp. 1977, 2009

第 29 章 データが変換される場所

CICS 相互通信で SNA リンクを使用している場合、システム・データは EBCDIC フォーマットで伝送されます。このため、ASCII ベースのシステムでは、アプリケーション・データ域を除くすべてのデータが変換されます。アプリケーション・データ域は、データを受信するシステムによって変換されます。

機能シップと DPL

機能シップおよび DPL では、データは ASCII ベースのシステムまたは CICS Transaction Server for z/OS で変換することができます。

ASCII ベースのシステムから CICS Transaction Server for z/OS への機能シップおよび DPL では、ASCII ベースのシステムによってリソース名が変換され、CICS Transaction Server for z/OS によってユーザー・データが変換されます。

表 27. 機能シップと DPL のデータ変換

要求タイプ	データ	変換タイプ	変換される場所
TS	キュー名	文字	ASCII システム
TS	FROM 区域	DFHCNV テーブルで指定	受信システム
TD	キュー名	文字	ASCII システム
TD	INTO 区域	DFHCNV テーブルで指定	受信システム
FC	ファイル名	文字	ASCII システム
FC	SET 域	DFHCNV テーブルで指定	受信システム
FC	キー	DFHCNV テーブルで指定	受信システム
IC	トランザクション ID	文字	ASCII システム
IC	FROM 区域	DFHCNV テーブルで指定	受信システム
IC	RTERMID、RTRANSID、 REQID	文字	ASCII システム
PC	プログラム名	文字	ASCII システム
PC	COMMAREA	DFHCNV テーブルで指定	受信システム

CICS Transaction Server for z/OS から ASCII ベースのシステムへの機能シップおよび DPL では、ASCII ベースのシステムによってすべてのデータが変換されます。

アプリケーション・データの変換は、フィールドごとに行われます。このため、アプリケーション・データの各フィールドのサイズは、そのフィールドに適用される

© Copyright IBM Corp. 1977, 2009

変換の結果を格納するのに十分な大きさにする必要があります。これについては、アプリケーション・データのフィールドに SBCS 文字と DBCS 文字の両方が格納される場合に特に注意する必要があります。

分散トランザクション処理

分散トランザクション処理では、すべてのデータ域はアプリケーションによって管理されます。このため、データ変換はアプリケーションによって行われます。

アプリケーションの設計時に、データを CICS Transaction Server for z/OS、ASCII ベースのシステム、またはその両方で変換することができます。

トランザクション・ルーティング

CICS Transaction Server for z/OS では、トランザクション・ルーティングのデータは変換されません。画面データは、常に 3270 データ・ストリームとして伝達されます。COMMAREA および TCTUA (疑似会話型トランザクションに関連) は、ASCII システムによって変換されます。

第 30 章 データ変換の回避

多くの場合、アプリケーションを設計して、変換されるデータの量を削減することができます。

例えば、EBCDIC ベースのシステムが ASCII ベースのシステムのファイル・マネージャーとして機能する場合は、ASCII を使用してファイル内のデータをエンコードすることにより、データが変換されるのを避けることができます。

反対に、EBCDIC ベースのシステムとの通信だけを目的として、データを ASCII ベースのシステムに格納している場合は、そのデータを EBCDIC でコード化することにより、データが変換されるのを避けることができます。

© Copyright IBM Corp. 1977, 2009

第 31 章 変換の種類

実行可能な変換の種類として、標準の変換、変換なし、およびユーザー定義の非標準の変換があります。

標準の変換

これは次のセッションに適用されます。

- 1 バイト文字セット (SBCS)
- グラフィックまたは 2 バイト文字セット (DBCS)
- 混合文字セット (SBCS データと DBCS データを含む)
- マルチバイト文字セット (MBCS)
- デフォルトでは、INTEL 形式のバイナリー・データに変換

変換なし

これは次のセッションに適用されます。

- UCS-2 または UTF-8 としてエンコードされた文字データ
- デフォルトでは、z/Architecture 形式のバイナリー・データに変換
- パック 10 進数データ

ユーザー定義の非標準変換

非標準のデータ変換を適用するには、独自のバージョンのユーザー置き換え可能変換プログラムを作成します。

ユーザー定義の変換を選択フィールドに適用し、他のフィールドを CICS の標準変換プログラムで変換することができます。

CICS Transaction Server for z/OS では、次のいずれかを指定することができます。

1. 独自にカスタマイズした DFHUCNV。または
2. 異なる名前が付けられた 1 つ以上の変換プログラム。

非標準の変換を文字データだけに適用する場合は、独自のデータ変換プログラムを作成する必要はありません。代わりに、標準の変換プログラムである DFHCCNV と共に使用する独自の変換テーブルを作成します。 403 ページの『第 38 章 ユーザー定義の変換テーブル』を参照してください。

重要: ユーザー提供の変換プログラムを使用して、標準の変換プログラムが変換を試みたデータを変換しないでください。データを 2 度に渡って変換すると、予測不能な結果が生じるおそれがあります。この状態を避けるために、独自の変換プログラムでは、DATATYP=USERDATA として定義されたフィールドのみを変換してください (DFHCNV TYPE=FIELD マクロの DATATYP オプションを参照)。

© Copyright IBM Corp. 1977, 2009

第 32 章 文字データ

文字データは文字セット *ID* およびコード・ページ *ID* で表されます。コード・ページ ID では、各文字のエンコード方法が定義されます。例えば、「A」は ASCII では X'41' とエンコードされ、EBCDIC では X'C1' とエンコードされます。

DFHCNV TYPE=ENTRY マクロの SRVERCPキーワードでは、EBCDIC コード・ページが指定されます。これは、CICS Transaction Server for z/OS において、リソースに関連する文字データがエンコードされるページです。

DFHCNV TYPE=ENTRY マクロの CLINTCP キーワードでは、デフォルトのコード・ページが指定されます。これは、CICS Transaction Server for z/OS が受け取った、または CICS Transaction Server for z/OS から送信された、特定リソースに関連する文字データをエンコードするページです。通常、データは ASCII フォーマットでエンコードされます。ただし、場合によっては、EBCDIC フォーマットでエンコードされることもあります。データが EBCDIC フォーマットでエンコードされる場合、コード・ページは SRVERCP キーワードで指定されたものと異なることがあります。

CLINTCP キーワードで指定されたコード・ページは指定変更することができます。こうすることにより、CICS Transaction Server for z/OS で、コード・ページを使用して文字データを表している各システムと通信することができます。

© Copyright IBM Corp. 1977, 2009

第 33 章 バイナリー・データ

DFHCNV TYPE=ENTRY マクロの DATATYP キーワードでは、CICS Transaction Server for z/OS が受け取ったバイナリー・データのデフォルトの形式が指定されます。

DATATYP=BINARY
バイナリー・データのデフォルトの形式にビッグ・エンディアンが指定されます。つまり、マルチバイトの数値の最初 (低いマシン・アドレス) に最上位バイトの値が含まれます。

DATATYP=NUMERIC
バイナリー・データのデフォルトの形式にリトル・エンディアンが指定されます。つまり、マルチバイトの数値の最初に、最下位バイトの値が含まれます。

デフォルトのバイナリー・フォーマットは指定変更することができます。このため、すべての 2 進数フィールドの DFHCNV TYPE=FIELD マクロをコード化することが重要です。

© Copyright IBM Corp. 1977, 2009

第 34 章 CICS がサポートされている変換

この解説書では、DFHCCNV 変換プログラムを使用したコード・ページ変換の、コード化文字セット ID (CCSID) の完全なリストを示しています。

非サポート CCSID では、標準の変換プログラムである DFHCCNV と共に使用できる独自の変換テーブルを作成できます。 403 ページの『第 38 章 ユーザー定義の変換テーブル』を参照してください。

非標準の変換では、独自の変換プログラムを提供する必要があります (382 ページの『ユーザー/CICS 変換』を参照)。

ASCII と EBCDIC 間の変換のグループ

両方の CCSID が同じグループに属している場合、通常 CICS Transaction Server for z/OS では ASCII と EBCDIC との間で文字データを変換することができます。ただし、同じグループ内であっても、変換には一定の制約があります。例えば、新しい CCSID を定義して文字セットを拡張する場合、同等の新しい ASCII CCSID と EBCDIC CCSID 間の変換はサポートされますが、両者に古いものと新しいものが混在するときには、変換はサポートされません。

グループは次のとおりです。

アラビア語

Baltic Rim
ラトビア、リトアニア、エストニア

キリル文字
東ヨーロッパ - ブルガリア、ロシア、ユーゴスラビア

デーバナーガリー (ヒンディ語)
インド

ペルシア語
イラン

ギリシャ語
ギリシャ

ヘブライ語
イスラエル

日本語
日本

韓国語
韓国

ラオ語
ラオス

Latin 1 と Latin 9
米国、西ヨーロッパ、その他の複数の国

© Copyright IBM Corp. 1977, 2009

Latin 2
東ヨーロッパ - アルバニア、チェコ、ハンガリー、ポーランド、ルーマニア、スロバキア、ユーゴスラビア、旧ユーゴスラビア

Latin 5
トルコ

中国語 (簡体字)
中華人民共和国

タイ語
タイ

中国語 (繁体字)
台湾

ウルドゥー語
パキスタン

ベトナム語
ベトナム

以下のセクションのテーブルには、各グループでサポートされている CCSID が示されています。テーブルでは、各 CCSID について次の方法が提供されています。

- CLINTCP キーワードまたは SRVERCP キーワードに指定される値。
- コード・ページ ID または ID (CPGID)。
- コード・ページの IANA 登録文字セット名 (適切な名前が存在し、CICS で EXEC CICS コマンドにこの名前を使用できる場合)。CICS によってサポートされている名前は基本名または推奨される別名です。場合によっては、複数の名前や別名がサポートされています。

アラビア語

アラビア語の変換

表 28. アラビア語、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
864	00864	00864	ibm864	PC データ: アラビア語
1089 8859-6	01089	01089	iso-8859-6 iso_8859-6	ISO 8859-6: アラビア語
1256	01256	01256	windows-1256	MS Windows: アラビア語
5352	05352	01256		MS Windows: アラビア語、バージョン 2 (ユーロを含む)
9448	09448	09448		MS Windows: アラビア語、2001
17248	17248	00864		PC データ: アラビア語 (ユーロを含む)

表 29. アラビア語、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
420	00420	00420	ibm420	ホスト: アラビア語
16804	16804	00420		ホスト: アラビア語 (ユーロを含む)

注: データを変換しても、アラビア語データの方向は変わりません。

Baltic Rim

Baltic Rim の変換

表 30. Baltic Rim、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
901	00901	00901		PC データ: ラトビア、リトアニア (ユーロを含む)
902	00902	00902		PC データ: エストニア (ユーロを含む)
921	00921	00921		PC データ: ラトビア、リトアニア
922	00922	00922		PC データ: エストニア
1257	01257	01257	windows-1257	MS Windows: Baltic Rim
5353	05353	01257		MS Windows: Baltic Rim、バージョン 2 (ユーロを含む)

表 31. Baltic Rim、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
1112	01112	01112		ホスト: ラトビア、リトアニア
1122	01122	01122		ホスト: エストニア
1156	01156	01156		ホスト: ラトビア、リトアニア (ユーロを含む)
1157	01157	01157		ホスト: エストニア (ユーロを含む)

キリル文字

キリル文字の変換

表 32. キリル文字、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
808	00808	00808		PC データ: キリル文字、ロシア (ユーロを含む)
848	00848	00848		PC データ: キリル文字、ウクライナ (ユーロを含む)
849	00849	00849		PC データ: キリル文字、ベラルーシ (ユーロを含む)
855	00855	00855	ibm855	PC データ: キリル文字
866	00866	00866	ibm866	PC データ: キリル文字、ロシア
872	00872	00872		PC データ: キリル文字 (ユーロを含む)
915 8859-5	00915	00915	iso-8859-5 iso_8859-5	ISO 8859-5: キリル文字
1124	01124	01124		8 ビット: キリル文字、ベラルーシ

表 32. キリル文字、クライアント *CCSID (続き)*

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
1125	01125	01125		PC データ: キリル文字、ウクライナ
1131	01131	01131		PC データ: キリル文字、ベラルーシ
1251	01251	01251	windows-1251	MS Windows: キリル文字
5347	05347	01251		MS Windows: キリル文字、バージョン 2 (ユーロを含む)

表 33. キリル文字、サーバー *CCSID*

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
1025	01025	01025		ホスト: キリル文字 (マルチリンガル)
1123	01123	01123		ホスト: キリル文字、ウクライナ
1154	01154	01154		ホスト: キリル文字 (マルチリンガル) (ユーロを含む)
1158	01158	01158		ホスト: キリル文字、ウクライナ (ユーロを含む)

デーバナーガリー

デーバナーガリーの変換

表 34. デーバナーガリー、クライアント *CCSID*

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
806	00806	00806		PC データ: ISCII-91、デーバナーガリーのスクリプト・コード

表 35. デーバナーガリー、サーバー *CCSID*

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
1137	01137	01137		ホスト: デーバナーガリー

注: これらのデーバナーガリー CCSID は、マラーティー語で使用される同一のデーバナーガリー文字のエンコードにも使用することができます。

ペルシア語

ペルシア語の変換

表 36. ペルシア語、クライアント *CCSID*

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
1098	01098	01098		PC データ: ペルシア語

表 37. ペルシア語、サーバー *CCSID*

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
1097	01097	01097		ホスト: ペルシア語

注: データを変換しても、ペルシア語データの方向は変わりません。

ギリシャ語

ギリシャ語の変換

表 38. ギリシャ語、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
813 8859-7	00813	00813	iso-8859-7 iso_8859-7	ISO 8859-7: ギリシャ
869	00869	00869	ibm869	PC データ: ギリシャ
1253	01253	01253	windows-1253	MS Windows: ギリシャ
4909	04909	00813		ISO 8859-7: ギリシャ (ユーロを含む)
5349	05349	01253		MS Windows: ギリシャ、バージョン 2 (ユーロを含む)
9061	09061	00869		PC データ: ギリシャ (ユーロを含む)

表 39. ギリシャ語、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
875	00875	00875		ホスト: ギリシャ
4971	04971	00875		ホスト: ギリシャ (ユーロを含む)

ヘブライ語

ヘブライ語の変換

表 40. ヘブライ語、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
856	00856	00856		PC データ: ヘブライ語
862	00862	00862	ibm862	PC データ: ヘブライ語 (マイグレーション)
867	00867	00867		PC データ: ヘブライ語 (ユーロを含む)
916 8859-8	00916	00916	iso-8859-8 iso_8859-8	ISO 8859-8: ヘブライ語
1255	01255	01255	windows-1255	MS Windows: ヘブライ語
5351	05351	01255		MS Windows: ヘブライ語、バージョン 2 (ユーロを含む)
9447	09447	01255		MS Windows: ヘブライ語、バージョン 2 (ユーロと新シェケルを含む)

表 41. ヘブライ語、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
424	00424	00424	ibm424	ホスト: ヘブライ語
803	00803	00803		ホスト: ヘブライ語 (文字セット A)

表 41. ヘブライ語、サーバー CCSID (続き)

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
4899	04899	00803		ホスト: ヘブライ語 (文字セット A) (ユーロを含む)
12712	12712	00424		ホスト: ヘブライ語 (ユーロと新シェケルを含む)

注: データを変換しても、ヘブライ語データの方向は変わりません。

日本語

日本語の変換

表 42. 日本語、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
932	00932	1. 00897 2. 00301		1. PC データ: SBCS 2. PC データ: DBCS (ユーザー定義文字 (1880 文字) を含む)
942	00942	1. 01041 2. 00301		1. PC データ: 拡張 SBCS 2. PC データ: DBCS (ユーザー定義文字 (1880 文字) を含む)
943	00943	1. 00897 2. 00941	シフト JIS x-sjis	1. PC データ: SBCS 2. PC データ: オープン環境用 DBCS (IBM ユーザー定義文字 (1880 文字) を含む)
954 EUCJP	00954	1. 00895 2. 00952 3. 00896 4. 00953	euc-jp	1. G0: JIS X201 Roman 2. G1: JIS X208-1990 3. G1: JIS X201 カタカナ 4. G1: JIS X212
5050	05050	1. 00895 2. 00952 3. 00896 4. 00953		1. G0: JIS X201 Roman 2. G1: JIS X208-1990 3. G1: JIS X201 カタカナ 4. G1: JIS X212

表 43. 日本語、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
930	00930	1. 00290 2. 00300 3. 00290 4. 00300		1. カタカナ・ホスト: 拡張 SBCS 2. 漢字ホスト: DBCS (ユーザー定義文字 (4370 文字) を含む) 3. カタカナ・ホスト: 拡張 SBCS 4. 漢字ホスト: DBCS (ユーザー定義文字 (1880 文字) を含む)
931	00931	1. 00037 2. 00300		1. ローマ字ホスト: SBCS 2. 漢字ホスト: DBCS (ユーザー定義文字 (4370 文字) を含む)

表 43. 日本語、サーバー CCSID (続き)

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
939	00939	1. 01027 2. 00300 3. 01027 4. 00300		1. ローマ字ホスト: 拡張 SBCS 2. 漢字ホスト: DBCS (ユーザー定義文字 (4370 文字) を含む) 3. ローマ字ホスト: 拡張 SBCS 4. 漢字ホスト: DBCS (ユーザー定義文字 (1880 文字) を含む)
1390	01390	1. 00290 2. 00300		1. カタカナ・ホスト: 拡張 SBCS (ユーロを含む) 2. 漢字ホスト: DBCS (ユーザー定義文字 (6205 文字) を含む)
1399	01399	1. 01027 2. 00300		1. ローマ字ホスト: 拡張 SBCS (ユーロを含む) 2. 漢字ホスト: DBCS (ユーザー定義文字 (4370 文字)、ユーロを含む)

韓国語

韓国語の変換

表 44. 韓国語、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
934	00934	1. 00891 2. 00926		1. PC データ: SBCS 2. PC データ: DBCS (ユーザー定義文字 (1880 文字) を含む)
944	00944	1. 01040 2. 00926		1. PC データ: 拡張 SBCS 2. PC データ: DBCS (ユーザー定義文字 (1880 文字) を含む)
949	00949	1. 01088 2. 00951		1. IBM KS コード - PC データ: SBCS 2. IBM KS コード - PC データ: DBCS (ユーザー定義文字 (1880 文字) を含む)
970 EUCKR	00970	1. 00367 2. 00971	euc-kr	1. G0: ASCII 2. G1: KSC X5601-1989 (ユーザー定義文字 (1880 文字) を含む)
1363	01363	1. 01126 2. 01362		1. PC データ: MS Windows® 韓国語 SBCS 2. PC データ: MS Windows 韓国語 DBCS (11172 文字のハングルを含む)

表 45. 韓国語、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
933	00933	1. 00833 2. 00834		1. ホスト: 拡張 SBCS 2. ホスト: DBCS (1880 文字のユーザー定義文字 と 11172 文字の全ハングル文字を含む)

表 45. 韓国語、サーバー CCSID (続き)

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
1364	01364	1. 00833 2. 00834		1. ホスト: 拡張 SBCS 2. ホスト: DBCS (1880 文字のユーザー定義文字 と 11172 文字の全ハングル文字を含む)

ラオ語

ラオ語の変換

表 46. ラオ語、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
1133	01133	01133		ISO-8: ラオ語

表 47. ラオ語、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
1132	01132	01132		ホスト: ラオ語

Latin 1 と Latin 9

Latin 1 と Latin 9 の変換

表 48. Latin 1、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
437	00437	00437	ibm437	PC データ: PC ベース - 米国およびその他の複数の国
819 8859-1	00819	00819	iso-8859-1 iso_8859-1	ISO 8859-1: Latin 1 の国
850	00850	00850	ibm850	PC データ: Latin 1 の国
858	00858	00858	ibm00858	PC データ: Latin 1 の国 (ユーロを含む)
923	00923	00923	iso-8859-15 iso_8859-15	ISO 8859-15: Latin 9
924	00924	00924	ibm00924	ISO 8859-15: Latin 9
1047	01047	01047		ホスト: Latin 1
1252	01252	01252	windows-1252	MS Windows: Latin 1 の国
5348	05348	01252		MS Windows: Latin 1 の国、バージョン 2 (ユーロを含む)

表 49. Latin 1 と Latin 9、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
037	00037	00037	ibm037	ホスト: 米国、カナダ (ESA)、オランダ、ポルトガル、ブラジル、オーストラリア、ニュージーランド

表 49. Latin 1 と Latin 9、サーバー CCSID (続き)

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
273	00273	00273	ibm273	ホスト: オーストリア、ドイツ
277	00277	00277	ibm277	ホスト: デンマーク、ノルウェー
278	00278	00278	ibm278	ホスト: フィンランド、スウェーデン
280	00280	00280	ibm280	ホスト: イタリア
284	00284	00284	ibm284	ホスト: スペイン、ラテンアメリカ (スペイン語)
285	00285	00285	ibm285	ホスト: 英国
297	00297	00297	ibm297	ホスト: フランス
500	00500	00500	ibm500	ホスト: ベルギー、カナダ (AS/400®)、スイス、International Latin 1
871	00871	00871	ibm871	ホスト: アイスランド
924	00924	00924	ibm00924	ホスト: Latin 9
1047	01047	01047		ホスト: Latin 1
1140	01140	01140	ibm01140	ホスト: 米国、カナダ (ESA)、オランダ、ポルトガル、ブラジル、オーストラリア、ニュージーランド (ユーロを含む)
1141	01141	01141	ibm01141	ホスト: オーストリア、ドイツ (ユーロを含む)
1142	01142	01142	ibm01142	ホスト: デンマーク、ノルウェー (ユーロを含む)
1143	01143	01143	ibm01143	ホスト: フィンランド、スウェーデン (ユーロを含む)
1144	01144	01144	ibm01144	ホスト: イタリア (ユーロを含む)
1145	01145	01145	ibm01145	ホスト: スペイン、ラテンアメリカ (スペイン語) (ユーロを含む)
1146	01146	01146	ibm01146	ホスト: 英国 (ユーロを含む)
1147	01147	01147	ibm01147	ホスト: フランス (ユーロを含む)
1148	01148	01148	ibm01148	ホスト: ベルギー、カナダ (AS/400)、スイス、International Latin 1 (ユーロを含む)
1149	01149	01149	ibm01149	ホスト: アイスランド (ユーロを含む)

注: ユーロがサポートされていない CCSID とユーロがサポートされている CCSID との間でも変換を行うことができます。ただし、次のような理由により、この変換を行う際には十分な注意が必要です。

- ユーロがサポートされていない各 EBCDIC CCSID (00500 など) の国際通貨記号は、ユーロがサポートされている同等の EBCDIC CCSID (01148 など) において、ユーロ記号に置換されています。
- ユーロがサポートされていない ASCII CCSID 00850 の dotless *i* は、ユーロがサポートされている同等の ASCII CCSID 00858 において、ユーロ記号に置換されています。

Latin 2

Latin 2 の変換

表 50. Latin 2、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
852	00852	00852	ibm852	PC データ: Latin 2 (マルチリンガル)
912 8859-2	00912	00912	iso-8859-2 iso_8859-2	ISO 8859-2: Latin 2 (マルチリンガル)
1250	01250	01250	windows-1250	MS Windows: Latin 2
5346	05346	01250		MS Windows: Latin 2、バージョン 2 (ユーロを含む)
9044	09044	00852		PC データ: Latin 2 (マルチリンガル) (ユーロを含む)

表 51. Latin 2、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
500	00500	00500	ibm500	ホスト: International Latin 1
870	00870	00870	ibm870	ホスト: Latin 2 (マルチリンガル)
924	00924	00924	ibm00924	ホスト: Latin 9
1140	01140	01140	ibm01140	ホスト: 米国、カナダ (ESA)、オランダ、ポルトガル、ブラジル、オーストラリア、ニュージーランド (ユーロを含む)
1141	01141	01141	ibm01141	ホスト: オーストリア、ドイツ (ユーロを含む)
1142	01142	01142	ibm01142	ホスト: デンマーク、ノルウェー (ユーロを含む)
1143	01143	01143	ibm01143	ホスト: フィンランド、スウェーデン (ユーロを含む)
1144	01144	01144	ibm01144	ホスト: イタリア (ユーロを含む)
1145	01145	01145	ibm01145	ホスト: スペイン、ラテンアメリカ (スペイン語) (ユーロを含む)
1146	01146	01146	ibm01146	ホスト: 英国 (ユーロを含む)
1147	01147	01147	ibm01147	ホスト: フランス (ユーロを含む)
1148	01148	01148	ibm01148	ホスト: International Latin 1 (ユーロを含む)
1149	01149	01149	ibm01149	ホスト: アイスランド (ユーロを含む)
1153	01153	01153		ホスト: Latin 2 (マルチリンガル) (ユーロを含む)

注: Latin 2 の ASCII CCSID と Latin 1 の EBCDIC CCSID の一部の組み合わせについて、変換がサポートされています。

Latin 5

Latin 5 の変換

表 52. Latin 5、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
857	00857	00857	ibm857	PC データ: Latin 5 (トルコ)
920 8859-9	00920	00920	iso-8859-9 iso_8859-9	ISO 8859-9: Latin 5 (ECMA-128、トルコ TS-5881)
1254	01254	01254	windows-1254	MS Windows: トルコ
5350	05350	01254		MS Windows: トルコ、バージョン 2 (ユーロを含む)
9049	09049	00857		PC データ: Latin 5 (トルコ) (ユーロを含む)

表 53. Latin 5、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
1026	01026	01026	ibm1026	ホスト: Latin 5 (トルコ)
1155	01155	01155		ホスト: Latin 5 (トルコ) (ユーロを含む)

中国語 (簡体字)

中国語 (簡体字) の変換

表 54. 中国語 (簡体字)、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
946	00946	1. 01042 2. 00928		1. PC データ: 拡張 SBCS 2. PC データ: DBCS (ユーザー定義文字 (1880 文字) を含む)
1381	01381	1. 01115 2. 01380	gb2312	1. PC データ: 拡張 SBCS (IBM GB) 2. PC データ: DBCS (IBM GB) (IBM が選択した 31 文字、および 1880 文字のユーザー定義文字を含む)
1383 EUCCN	01383	1. 00367 2. 01382		1. G0: ASCII 2. G1: GB 2312-80 セット
1386	01386	1. 01114 2. 01385		1. PC データ: 中国語 (簡体字) および中国語 (繁体字) IBM BIG-5 2. PC データ: 中国語 (簡体字) GBK
5488	05488	1. 01252 2. 01385 3. 01391	gb18030	1. GB18030、1 バイトのデータ 2. GB18030、2 バイトのデータ 3. GB18030、4 バイトのデータ

表 55. 中国語 (簡体字)、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
935	00935	1. 00836 2. 00837		1. ホスト: 拡張 SBCS 2. ホスト: DBCS (ユーザー定義文字 (1880 文字) を含む)
1388	01388	1. 00836 2. 00837		1. ホスト: 拡張 SBCS 2. ホスト: DBCS (ユーザー定義文字 (1880 文字) を含む)
9127	09127	1. 00836 2. 00837		1. ホスト: 拡張 SBCS 2. ホスト: DBCS (ユーザー定義文字 (1880 文字) を含む)

タイ語

タイ語の変換

表 56. タイ語、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
1161	01161	01161		PC データ: タイ語 (ユーロを含む)
1162	01162	01162		MS Windows: タイ語 (ユーロを含む)
9066	09066	00874		PC データ: タイ語拡張 SBCS

表 57. タイ語、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
1160	01160	01160		ホスト: タイ語 (ユーロを含む)
9030	09030	00838		ホスト: タイ語拡張 SBCS

中国語 (繁体字)

中国語 (繁体字) の変換

表 58. 中国語 (繁体字)、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
938	00938	1. 00904 2. 00927		1. PC データ: SBCS 2. PC データ: DBCS (ユーザー定義文字 (6204 文字) を含む)
948	00948	1. 01043 2. 00927		1. PC データ: 拡張 SBCS 2. PC データ: DBCS (ユーザー定義文字 (6204 文字) を含む)
950 BIG5	00950	1. 01114 2. 00947	big5	1. PC データ: SBCS (IBM BIG5) 2. PC データ: DBCS (13493 文字の CNS 文字、IBM が選択した 566 文字、および 6204 文字のユーザー定義文字を含む)

表 58. 中国語 (繁体字)、クライアント CCSID (続き)

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
964 EUCTW	00964	1. 00367 2. 00960 3. 00961		1. G0: ASCII 2. G1: CNS 11643 Plane 1 3. G1: CNS 11643 Plane 2
1370	01370	1. 01114 2. 00947		1. PC データ: 拡張 SBCS (ユーロを含む) 2. PC データ: DBCS (ユーザー定義文字 (6204 文字)、ユーロを含む)

表 59. 中国語 (繁体字)、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
937	00937	1. 00037 2. 00835		1. ホスト: 拡張 SBCS 2. ホスト: DBCS (ユーザー定義文字 (6204 文字) を含む)
1371	01371	1. 01159 2. 00835		1. ホスト: 拡張 SBCS (ユーロを含む) 2. ホスト: DBCS (ユーザー定義文字 (6204 文字)、ユーロを含む)

ウルドゥー語

ウルドゥー語の変換

表 60. ウルドゥー語、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
868	00868	00868	ibm868	PC データ: ウルドゥー語
1006	01006	01006		ISO-8: ウルドゥー語

表 61. ウルドゥー語、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
918	00918	00918	ibm918	ホスト: ウルドゥー語

注: データを変換しても、ウルドゥー語データの方向は変わりません。

ベトナム語

ベトナム語の変換

表 62. ベトナム語、クライアント CCSID

CLINTCP	CCSID	CPGID	IANA 文字セット名	コメント
1129	01129	01129		ISO-8: ベトナム語
1163	01163	01163		ISO-8: ベトナム語 (ユーロを含む)
1258	01258	01258	windows-1258	MS Windows: ベトナム語
5354	05354	01258		MS Windows: ベトナム語、バージョン 2 (ユーロを含む)

表 63. ベトナム語、サーバー CCSID

SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
1130	01130	01130		ホスト: ベトナム語
1164	01164	01164		ホスト: ベトナム語 (ユーロを含む)

Unicode データ

CICS Transaction Server for z/OS では、Unicode でエンコードされた文字データのサポートは制限されています。このため、変換が必要でない場合に限り、ワークステーションと CICS Transaction Server for z/OS との間で、UCS-2 または UTF-8 でエンコードされたデータを共用することができます。

注: チャネルを使用してデータを通信する場合、CICS では Unicode データへの変換、または Unicode データからの変換について、より強力なサポートが提供されます。「*CICS アプリケーション・プログラミング・ガイド*」を参照してください。

表 64. Unicode

CLINTCP SRVERCP	CCSID	CPGID	IANA 文字セット名	コメント
1200 UCS-2	01200	01400	utf-16	文字セット 65535 を含む Unicode。バイト・オーダー・マーク (BOM) がない場合は、UTF-16 (ビッグ・エンディアン) であると仮定されます。
1208 UTF-8	01208	01400	utf-8	文字セット 65535 を含む Unicode。UTF-8。
13488	13488	01400	iso-10646-ucs-2	文字セット 3001 を含む Unicode (Unicode 2.0 文字レパートリーで固定)。バイト・オーダー・マークがない場合は、UTF16-BE (ビッグ・エンディアン) であると仮定されます。
17584	17584	01400		文字セット 3004 を含む Unicode (Unicode 3.0 文字レパートリーで固定)。バイト・オーダー・マークがない場合は、UTF16-BE (ビッグ・エンディアン) であると仮定されます。

第 35 章 変換処理

このセクションでは、CICS でのデータ変換方法について詳しく説明します。

コンポーネント

CICS やユーザー提供のミラー・トランザクションでは、DFHCNV、DFHCCNV、およびユーザー置き換え可能変換プログラムである DFHUCNV を使用して、データが変換されます。

DFHCNV
変換テーブルです。DFHCNV には、変換が必要な各リソースについての変換テンプレートが格納されています。変換テンプレートは、変換対象の、データ域のフィールド、および各フィールドに適用される変換方式を定義するテーブル項目です。

DFHCNV テーブルは、DFHCNV リソース定義マクロを使用して定義します (389 ページの『第 37 章 変換テーブルの定義』を参照)。

DFHCCNV
変換処理を制御する CICS プログラムです。DFHCCNV は、DFHCNV テーブルを使用して、必要な変換を確認します。変換テンプレート内の、ユーザーが制御する非標準の変換が指定されていないフィールドには、標準の変換が適用されます。

ユーザー置き換え可能変換プログラム DFHUCNV
CICS によって適用される標準の変換を指定変更するためのユーザー置き換え可能プログラムです。これを使用して、独自の変換ロジックを特定のデータ・フィールドに適用することができます。その方法については、 382 ページの『ユーザー/CICS 変換』を参照してください。

提供されたプログラムは、独自のバージョンを作成する際のモデルとして使用できます。

次のいずれかの プログラムを提供する必要があります。

- 独自にカスタマイズした DFHUCNV。または
- 異なる名前が付けられた 1 つ以上の変換プログラム。

処理

このセクションでは、DFHCCNV で変換テンプレートの特定のフィールドに適用することができる標準の変換について説明します。DFHUCNV プログラムを作成することによって、他のタイプの変換を行うこともできます。

文字データ
文字データは、次のように変換することができます。

- ASCII から EBCDIC への変換。この変換は、接続されたシステムから要求を受信した際、EXEC インターフェースが呼び出される前に行われます。

© Copyright IBM Corp. 1977, 2009

- EBCDIC から ASCII への変換。この変換は、EXEC インターフェースから戻る際、応答が伝送される前に行われます。

CICS に付属している変換テーブルは、*IBM Character Data Representation Architecture Level 2 - Registry* (SC09-1391) に示されている標準に準拠しています。

バイナリー・データ

バイナリー・データは、次のように変換することができます。

- リトル・エンディアン形式からビッグ・エンディアン形式への変換。この変換は、接続されたシステムから要求を受信した際に行われます。
- ビッグ・エンディアン形式からリトル・エンディアン形式への変換。この変換は、応答が伝送される前に行われます。

標準の変換と非標準の変換

ファイルなどの単一リソースを変換するには、次の 3 つの方法があります。

- CICS による変換 - すべてのデータ・フィールドが標準の CICS 変換プログラムである DFHCCNV によって処理されます。
- ユーザー/CICS 変換 - 非標準の変換と標準の変換の組み合わせ。つまり、一部のデータ・フィールドがユーザーの変換プログラムのコードによって処理され、他のデータ・フィールドが DFHCCNV によって処理されます。
- ユーザーによる変換 - すべてのデータ・フィールドがユーザーの変換プログラムによって処理されます。

CICS による変換

リソースに、非標準の変換を必要とするデータ・フィールドが含まれていない場合は、CICS による変換を使用します。こうすると、すべてのデータ・フィールドが標準の方法で変換されます。

1. DFHCNV マクロを使用して、変換テンプレートを作成します (389 ページの『第 37 章 変換テーブルの定義』) を参照。 こうすると、リソースが DFHCCNV によって処理されます。
2. リソースを定義する DFHCNV TYPE=ENTRY マクロで USREXIT=NO を指定します。 こうすると、DFHUCNV が不必要に呼び出されるのを避けることができます。データ・フィールドを定義する DFHCNV TYPE=FIELD マクロで DATATYP=USERDATA を指定しないでください。

ユーザー/CICS 変換

リソースに、標準の方法で変換できるフィールドと非標準の変換が必要なフィールドが混在している場合は、ユーザー/CICS 変換を使用します。

1. 変換テンプレートを作成します。
2. リソースを定義する DFHCNV TYPE=ENTRY マクロで USREXIT キーワードを指定します。
   - USREXIT=YES を指定すると、データの変換時に、CICS によって DFHUCNV が呼び出されます。

- USREXIT=*プログラム* を指定すると、データの変換時に、CICS によって指定プログラムが呼び出されます。

3. 非標準データ・フィールドを定義する DFHCNV TYPE=FIELD マクロで DATATYP=USERDATA を指定します。
   a. オプション: 非標準フィールドを USRTYPE 値で定義します。このとき、X'50' から X'80' の範囲で指定します。 この値はユーザー・プログラムに渡され、異なるタイプの非標準フィールドとの区別に使用することができます。
4. 必要に応じて、標準フィールドを DATATYP=CHARACTER、PD、BINARY、GRAPHIC、または NUMERIC として定義します。
5. 非標準フィールドを処理するユーザー作成バージョンの DFHUCNV、または別に指定された変換プログラムを指定します。 413 ページの『第 41 章 ユーザー置き換え可能変換プログラム』では、DFHUCNV の詳細とそのリスト、および DFHUCNV を独自の変換プログラムの基礎として使用する方法について説明しています。

ユーザーによる変換

リソースに、標準の方法で変換できるフィールドが含まれていません。すべてのフィールドで、非標準の変換を行う必要があります。ユーザーによる変換を有効にするには、2 つの方法があります。

1. 変換テンプレートを作成します。
2. リソースを定義する DFHCNV TYPE=ENTRY マクロで USREXIT キーワードを指定します。
   - USREXIT=YES を指定すると、データの変換時に、CICS によって DFHUCNV が呼び出されます。
   - USREXIT=*プログラム* を指定すると、データの変換時に、CICS によって指定プログラムが呼び出されます。
3. 非標準データ・フィールドを定義する DFHCNV TYPE=FIELD マクロで DATATYP=USERDATA を指定します。
   a. オプション: 非標準フィールドを USRTYPE 値で定義します。このとき、X'50' から X'80' の範囲で指定します。 この値はユーザー・プログラムに渡され、異なるタイプの非標準フィールドとの区別に使用することができます。
4. すべてのフィールドを処理するユーザー作成バージョンの DFHUCNV、または別に指定された変換プログラムを指定します。 413 ページの『第 41 章 ユーザー置き換え可能変換プログラム』では、DFHUCNV の詳細とそのリスト、および DFHUCNV を独自の変換プログラムの基礎として使用する方法について説明しています。
5.

変換処理の順序

次に、変換処理の順序を示します。

1. リソースの変換テンプレートを定義する DFHCNV TYPE=ENTRY マクロで USREXIT=NO を指定しない限り、DFHCCNV は DFHUCNV にリンクし、 413 ページの『パラメーター・リスト (DFHUVNDS)』に示されているパラメーターのリストを渡します。

   注:

   a. テンプレートが定義されていない場合は、リソースのすべての変換をユーザー・プログラムで処理するという前提のもとに、DFHUCNV が呼び出されます。

   b. すべての DFHCNV TYPE=ENTRY マクロで USREXIT=NO を指定している場合を除き、DFHUCNV がシステムに存在している必要があります。

2. リソースの変換テンプレートが定義されている場合、DFHUCNV によって、ユーザー・データの範囲にある、タイプが指定されたフィールドの変換が行われます。

   リソースの変換テンプレートが定義されていない場合は、DFHUCNV によってデータ形式が確認され、適切なフィールドの変換が行われます。

3. DFHUCNV から戻る際に、ユーザー定義の変換の対象ではないフィールドにおいて、変換テンプレートで指定された標準の変換が DFHCCNV によって実行されます。

4. シップされた要求が実行されます。

385 ページの図 98 はこの変換処理を示しています。

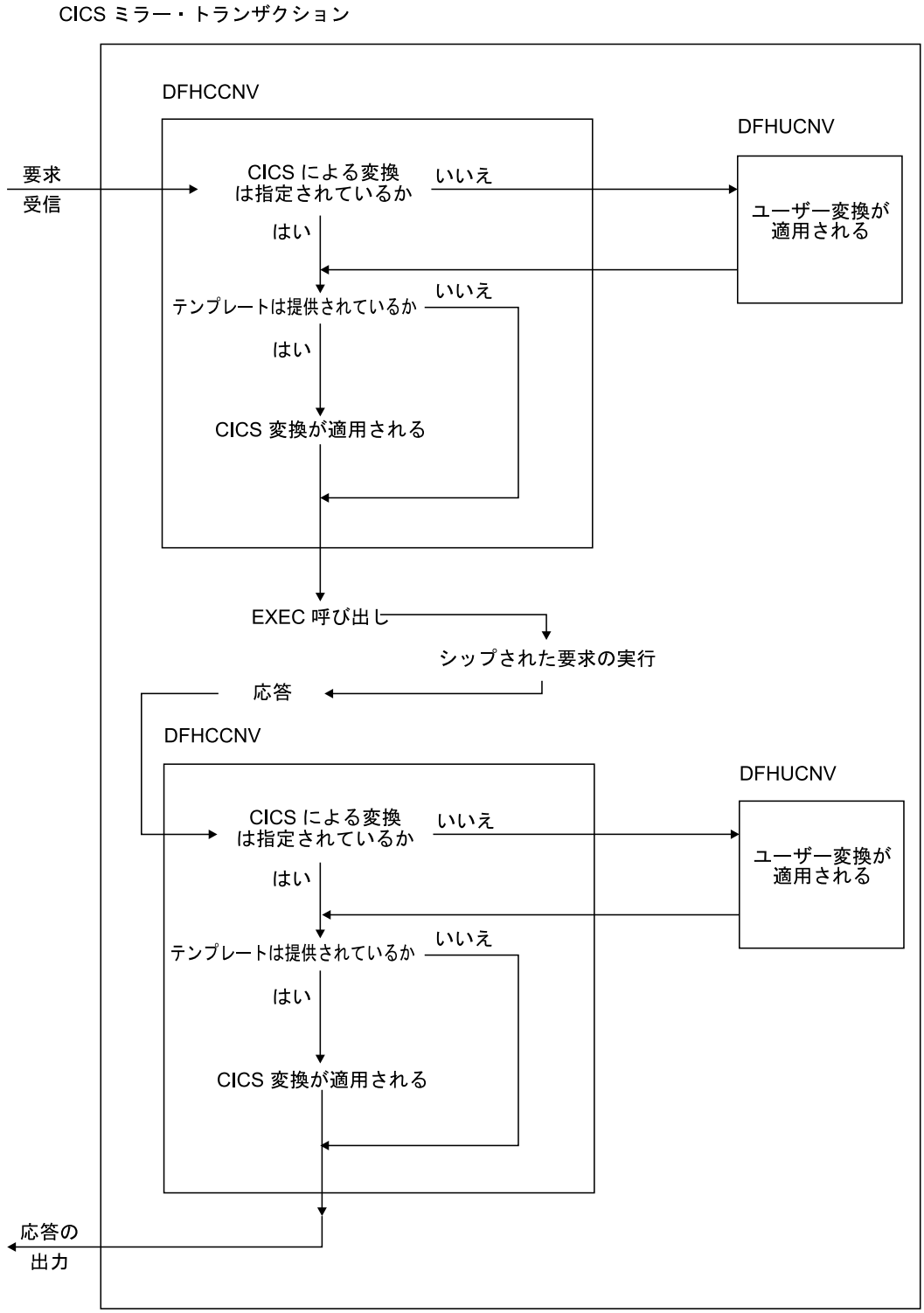

CICS ミラー・トランザクション
DFHCCNV
DFHUCNV
要求
受信
CICS による変換
は指定されているか
いいえ
ユーザー変換が
適用される
はい
テンプレートは提供されているか
いいえ
はい
CICS 変換が適用される
EXEC 呼び出し
シップされた要求の実行
応答
DFHCCNV
DFHUCNV
CICS による変換
は指定されているか
いいえ
ユーザー変換が
適用される
はい
テンプレートは提供されているか
いいえ
はい
CICS 変換が適用される
応答の
出力

図 98. データ変換処理

第 36 章 データ変換を可能にするためのリソース定義

CICS Transaction Server for z/OS でデータを変換するには、CICS 領域で一部のリソースを定義する必要があります。

定義する必要があるリソースは次のとおりです。

- DFHCNV - 変換テーブル
- DFHCCNV - 標準の変換プログラム
- DFHUCNV - ユーザー定義の変換プログラム

© Copyright IBM Corp. 1977, 2009

第 37 章 変換テーブルの定義

DFHCNV リソース定義マクロを使用して、変換テーブルを定義します。

DFHCNV マクロ・アセンブリーの出力には、リソース変換要件を指定するテンプレート、および必要な変換を可能にする変換テーブルが含まれています。ユーザーが生成した変換テーブルは、DFHCNV マクロ・ソースに配置する必要があります。

DFHCNV マクロのタイプ

DFHCNV マクロを使用して、変換テーブルを定義します。

DFHCNV TYPE=INITIAL
変換テーブルの開始を定義します。このマクロでは、デフォルトのクライアントとサーバー CCSID を定義します。

DFHCNV TYPE=ENTRY
データ・リソースを固有に識別するための名前およびタイプを指定します。変換が必要な各リソースに DFHCNV TYPE=ENTRY マクロを指定します。DFHCNV TYPE=ENTRY マクロで定義されていないリソースでは、データは変換されません。1 つのリソースの項目は次の TYPE=ENTRY ステートメントで、またはテーブルの終了時に完了します。使用する CCSID が指定されます。

同じリソース・タイプの複数のリソースに適用する汎用テンプレートを作成することができます。それには、DFHCNV TYPE=ENTRY マクロの RPFX パラメーターまたは XRPFX パラメーターを使用して、特定リソースの絶対パス名ではなく、複数のリソース名に一致する接頭部を指定します。

このようにリソースを定義するのは、変換テーブルでは順序が重要であるためです。例えば、ファイル・リソースの指定時に、接頭部 AB が接頭部 ABCD の前にある場合、ABCDEFGH という名前のファイル・リソースのデータを変換する際には、最初の項目が使用されます。この例では、変換テーブルのアセンブル時にエラーが発生します。エラーを避けるには、変換テーブルの上部に最も具体的なリソース名を入力し、下部に最も具体的ではない接頭部を入力します。

リソース名や接頭部が指定されていない場合は、その特定のリソース・タイプの変換時に、デフォルトの変換テンプレートが使用されます。

DFHCNV TYPE=ENTRY マクロの例については、 394 ページの『DFHCNV TYPE=ENTRY』を参照してください。

DFHCNV TYPE=KEY
FC 項目にのみ適用されます。このマクロは、レコードにキーでアクセスする必要がある場合にのみ使用します。レコードへのアクセス時に、常に相対レコード番号や相対バイト・アドレスを使用する場合には、TYPE=KEY マクロをコード化しないでください。使用する場合は、このマクロを TYPE=ENTRY の直後に、また 1 つ以上の TYPE=FIELD マクロの前に配置してください。TYPE=FIELD マクロは、キーに適用するデータ変換を定義します。

DFHCNV TYPE=SELECT
レコード内のフィールドの値に基づいて、データ変換のための選択レコード

© Copyright IBM Corp. 1977, 2009

(FC レコード、TS データ、TD データ、IC 開始元データ、または DPL で伝送される COMMAREA) を定義します。各 TYPE=SELECT マクロの後に 1 つ以上の TYPE=FIELD マクロを配置します。このマクロは、TYPE=SELECT マクロで定義された条件をレコードが満たした場合に適用されるデータ変換を定義します。各項目の最後の TYPE=SELECT マクロは OPTION=DEFAULT マクロです。このマクロは、前の TYPE=SELECT マクロで定義された条件を満たしていないレコードに適用される変換を定義します。

DFHCNV TYPE=FIELD
フィールドの位置と長さ、およびそのフィールドに適用する変換を指定します。変換が必要な各フィールドに TYPE=FIELD マクロを指定してください。

DFHCNV TYPE=FINAL
変換テーブル定義を終了します。

変換テンプレートとキー・テンプレート

テンプレートは、変換対象の、データ域またはキーのフィールド、および各フィールドに適用される変換方式を定義するテーブル項目です。テンプレートには、変換テンプレートとキー・テンプレートの 2 つのタイプがあります。

- 変換テンプレートは、DFHCNV TYPE=SELECT マクロに続けて DFHCNV TYPE=FIELD マクロを 1 つ以上指定して定義します。
- キー・テンプレートは、DFHCNV TYPE=KEY マクロに続けて DFHCNV TYPE=FIELD マクロを 1 つ以上指定して定義します。

これらのテンプレートは、両方とも、テーブル定義の中の次に非 FIELD マクロが出現したところで停止します。 392 ページの図 100 は、変換テーブル定義内のテンプレートを示しています。

クライアント・コード・ページおよびサーバー・コード・ページのデフォルト

必要な変換テーブルの数を減らすには、システム初期設定テーブルで、デフォルトのクライアント・コード・ページまたはサーバー・コード・ページを定義することを指定します。

クライアント・コード・ページでは、次の操作を行います。

1. DFHCNV TYPE=ENTRY マクロおよび TYPE=SELECT マクロで、CLINTCP パラメーターに値 SYSDEF を指定します。
2. システム初期設定テーブルで、CLINTCP パラメーターに値を指定して、デフォルトのクライアント・コード・ページを設定します。DFHCNV マクロの CLINTCP パラメーターにサポートされている任意の値を使用できます。デフォルトは CLINTCP=437 です。

サーバー・コード・ページでは、以下の操作を行います。

1. DFHCNV TYPE=ENTRY マクロおよび TYPE=SELECT マクロで、SRVERCP パラメーターに値 SYSDEF を指定します。

2. システム初期設定テーブルで、SRVERCP パラメーターに値を指定して、サーバー・コード・ページを設定します。DFHCNV マクロの SRVERCP パラメーターにサポートされている任意の値を使用できます。デフォルトは SRVERCP=037 です。

初期プログラム検査 (IVP) 用の変換テーブル

CICS Transaction Server for z/OS の IVP ジョブを実行する際には、変換テーブルが必要です。

図 99 には、変換テーブル定義の簡単な例が示されています。これらのマクロをすべてコード化する必要はありません。まったく同じ変換テーブルを生成するには、特殊なマクロである DFHCNV TYPE=IVP をアセンブルします。

フィールドはすべて文字であるため、TYPE=SELECT マクロは 1 つだけ必要です。これは OPTION=DEFAULT を指定しており、1 つの TYPE=FIELD マクロでデータ・レコード全体を定義しています。

TYPE=KEY マクロの後に、TYPE=FIELD が 1 つあります。これは、データ・レコードの最初の 6 バイトを再定義します。

```
DFHCNV TYPE=INITIAL
DFHCNV TYPE=ENTRY,RTYPE=FC,RNAME=FILEA,USREXIT=NO
DFHCNV TYPE=KEY
DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=6,LAST=YES
DFHCNV TYPE=SELECT,OPTION=DEFAULT
DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=80,LAST=YES
DFHCNV TYPE=FINAL
```

図 99. IVP の変換テーブル

392 ページの図 100 には、典型的な DFHCNV マクロの順序が示されています。この図には、リソース項目、変換テンプレート、およびキー・テンプレートに対応する一連の項目を示す注釈が付けられています。インデントは、そのマクロがネストされていることを示しています。マクロのコーディング時には、すべての CICS リソース定義マクロと同様に、アセンブラー規則を順守します。

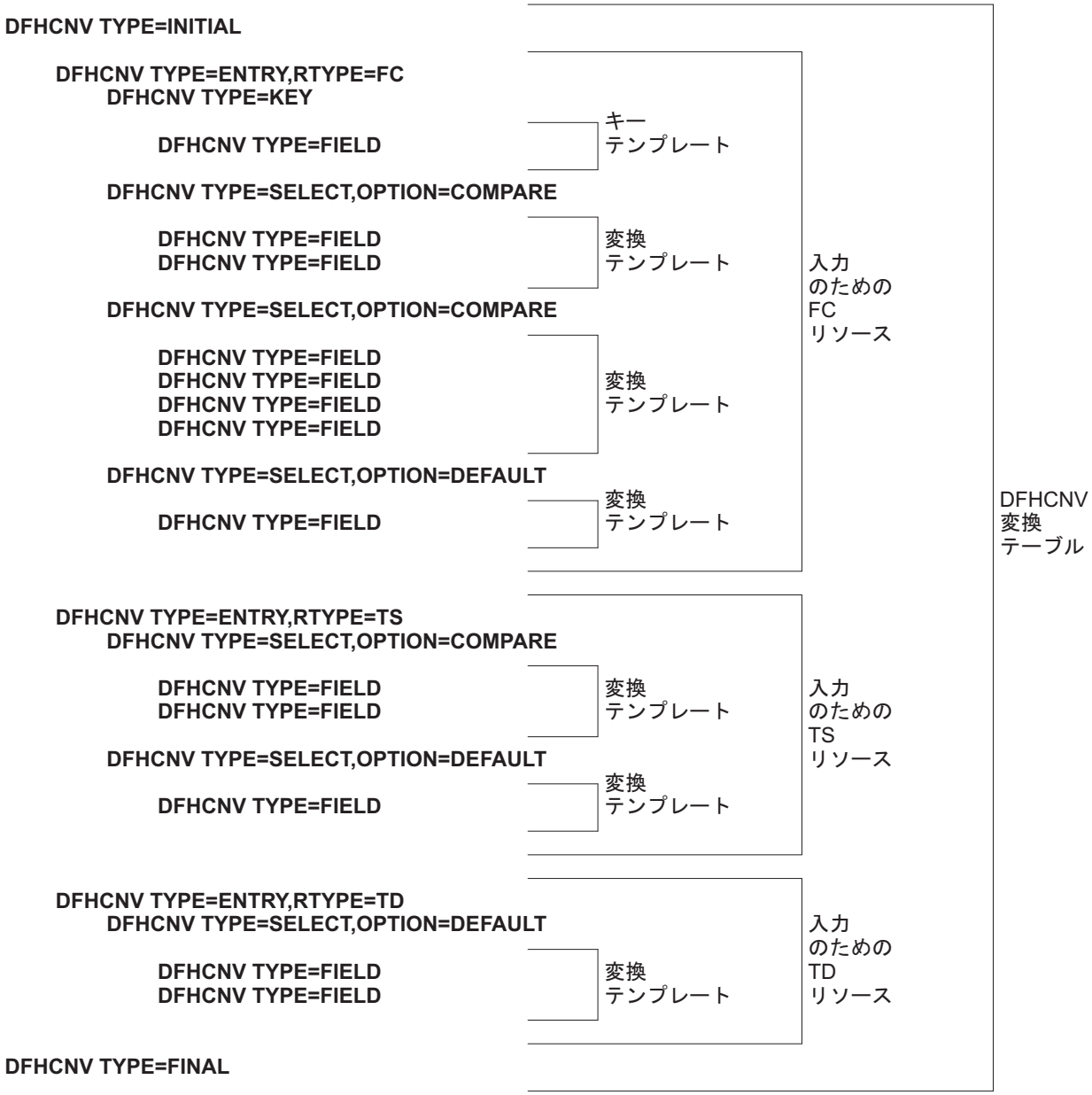
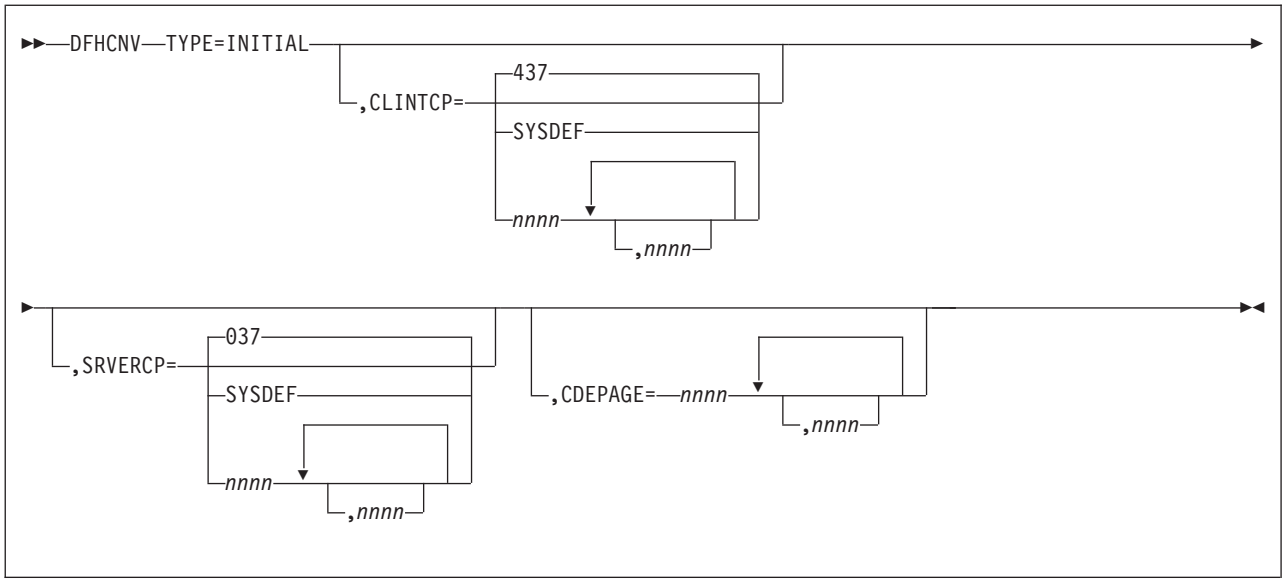


図 100. DFHCNV マクロ順序の例

DFHCNV TYPE=INITIAL

これは、DFHCNV TYPE=INITIAL マクロのフォーマットです。



TYPE=INITIAL
変換テーブルの開始を定義します。

CLINTCP={437|SYSDEF|nnnn[,nnnn, ...]}
第 1 オペランドは、CLINTCP と CDEPAGE の各オペランドが DFHCNV TYPE=ENTRY マクロにない場合に使用されるデフォルトのクライアント CCSID を定義します。

SYSDEF は、デフォルトのクライアント・コード・ページがシステム初期設定テーブル・パラメーター CLINTCP によって決定されることを示しています。

コード・ページの説明、および指定可能なコード・ページのリストについては、363 ページの『第 32 章 文字データ』を参照してください。

SRVERCP={037|SYSDEF|nnnn[,nnnn, ...]}
第 1 オペランドは、SRVERCP と CDEPAGE の各オペランドが DFHCNV TYPE=ENTRY マクロにない場合に使用されるサーバー CCSID を定義します。

SYSDEF は、デフォルトのサーバー・コード・ページがシステム初期設定テーブル・パラメーター SRVERCP によって決定されることを示しています。

コード・ページの説明、および指定可能なコード・ページのリストについては、363 ページの『第 32 章 文字データ』を参照してください。

CDEPAGE=nnnn[,nnnn...]

制約事項: このパラメーターは新しい定義に使用しないでください。これは、以前のリリースとの互換性を保つためにサポートされています。

使用可能な値は、CLINTCP 項目と SRVERCP 項目のペア、またはユーザー定義の変換を実行する場合は SRVERCP 項目と同等です。

437
以下の値と同等です。

- CLINTCP=437
- SRVERCP=037

932K
以下の値と同等です。

- CLINTCP=932
- SRVERCP=930

932
以下の値と同等です。

- CLINTCP=932
- SRVERCP=931

USR
以下の値と同等です。

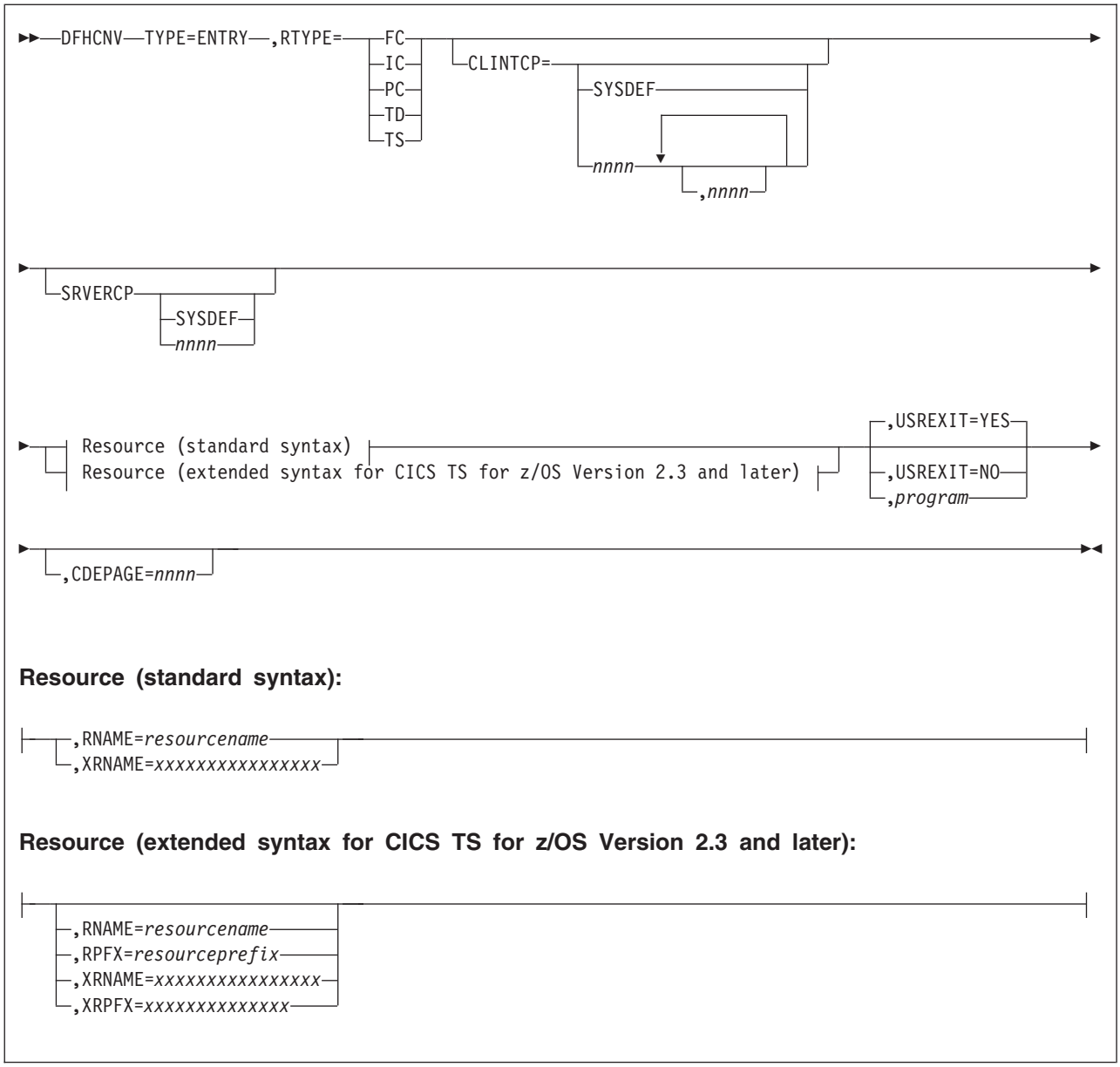
- SRVERCP=USR

USRD
以下の値と同等です。

- SRVERCP=USRD

DFHCNV TYPE=ENTRY

これは、DFHCNV TYPE=ENTRY マクロ命令のフォーマットです。



Resource (standard syntax):

Resource (extended syntax for CICS TS for z/OS Version 2.3 and later):

TYPE=ENTRY
このマクロでは、リソースを名前とタイプで定義することを指定します。

RTYPE={FC|TS|TD|IC|PC}
リソースのタイプを指定します。

FC ファイル

TS 一時記憶域キュー

TD 一時データ・キュー

IC データによるインターバル制御機能の開始

PC COMMAREA とのプログラム・リンク

CLINTCP={nnnn[,nnnn, ...]|SYSDEF}
第 1 オペランドは、使用するデフォルトのクライアント・コード・ページを定義します。

SYSDEF は、デフォルトのクライアント・コード・ページが、システム初期設定テーブル・パラメーター CLINTCP によって決定されることを示しています。

コード・ページの説明、および指定可能なコード・ページのリストについては、363 ページの『第 32 章 文字データ』を参照してください。

SRVERCP={nnnn|SYSDEF}
第 1 オペランドは、使用するサーバー・コード・ページを定義します。

SYSDEF は、サーバー・コード・ページが、システム初期設定テーブル・パラメーター SRVERCP によって決定されることを示しています。

コード・ページの説明、および指定可能なコード・ページのリストについては、363 ページの『第 32 章 文字データ』を参照してください。

RNAME=resourcename
リソースの名前を最大 8 文字で指定します。名前が 8 文字よりも短い場合はブランクが埋め込まれ、8 文字よりも長い場合は切り捨てられます。次の名前を指定できます。

- ファイル名 (最大 8 文字)
- TS キュー名 (最大 8 文字)

  注: CICS では、16 文字までの TS キュー名がサポートされていますが、DFHCNV では 8 文字までの TS キュー名がサポートされています。
- TD キュー名 (最大 4 文字)
- IC 開始トランザクション ID (最大 4 文字)
- リンクされるプログラムの名前 (最大 8 文字)

RPFX=resourceprefix
プログラム、TS キュー、およびファイルのリソース接頭部を最大 7 文字で指定するか、TD キューとトランザクションのリソース接頭部を 3 文字で指定します。リソース接頭部を使用すると、1 つのマクロを使用して、特定のタイプのリソースをグループ化することができます。指定したタイプと接頭部のリソースは、すべて同様に処理されます。順序は重要です。このため、変換テーブルの上部に最も具体的なリソース名を配置し、下部に最も具体的ではない接頭部を配置します。この時点で、マクロにパラメーターを指定しないと、指定したリソース・タイプ内のすべてのリソースに、デフォルトのテンプレートが使用されます。

XRNAME=xxxxxxxxxxxxxxxx (RTYPE=TS のみ)
リソース名を 16 進表記で指定します。これには最大で 16 桁の 16 進数字を含めることができ、必要に応じてブランクを埋め込むことができます。

XRPFX=xxxxxxxxxxxxxx (RTYPE=TS のみ)
リソース接頭部を最大 14 桁の16 進数字で指定します。リソース接頭部を使用すると、特定のタイプのリソースをグループ化することができます。指定したタイプと接頭部のリソースは、すべて同様に処理されます。順序は重要です。このため、変換テーブルの上部に最も具体的なリソース名を配置し、下部に最も具体的ではない接頭部を配置します。この時点で、マクロにパラメーターを指定しないと、指定したリソース・タイプ内のすべてのリソースに、デフォルトのテンプレートが使用されます。

USREXIT={YES|NO∨*program*}

ユーザー・データ変換出口を呼び出すかどうかを指定します。

YES

このリソースには、ユーザー定義の変換が必要です。DFHUCNV が呼び出されます。カスタマイズした DFHUCNV でこのリソースの一部のデータを変換する必要がある場合は、これをコード化します。

NO

このリソースには、ユーザー定義の変換は必要ありません。ユーザー置き換え可能変換プログラムは呼び出されません。プログラムを不必要に呼び出すことから生じるオーバーヘッドを除去するには、これをコード化します。

program

このリソースには、ユーザー定義の変換が必要です。*program* が呼び出されます。ユーザー提供のプログラム (*program*) でこのリソースの一部のデータを変換する必要がある場合は、これをコード化します。

CDEPAGE=nnnn

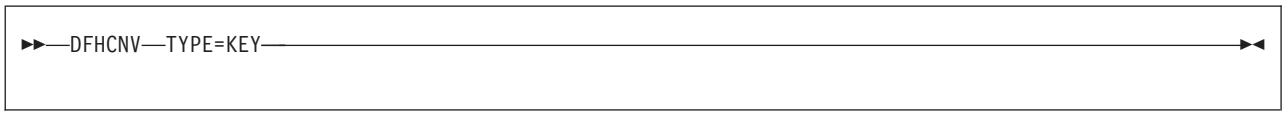
制約事項: このパラメーターは新しい定義に使用しないでください。これは、以前のリリースとの互換性を保つためにサポートされています。

コード・ページは、DFHCNV TYPE=INITIAL マクロの CDEPAGE オプションに入力されているもののいずれかにする必要があります。使用可能な値は、CLINTCP 項目と SRVERCP 項目のペア、またはユーザー定義の変換を実行する場合は SRVERCP 項目と同等です。各値が解決する CLINTCP 値と SRVERCP 値は、DFHCNV TYPE=INITIAL マクロの CDEPAGE オプションの説明で指定されます。

DFHCNV TYPE=KEY

DFHCNV TYPE=KEY マクロは FC RTYPE 要求にのみ有効です。コード化する際には、このマクロを DFHCNV TYPE=ENTRY マクロの直後に配置する必要があります。

マクロのフォーマットは次のとおりです。

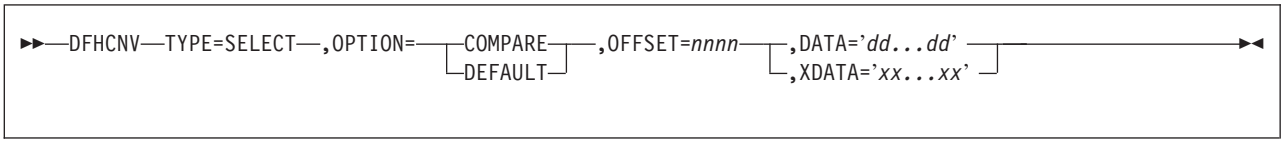


TYPE=KEY

キーに適用される変換の開始を示します。アクセス時に RRN または RBA のみが使用される場合、このマクロは必要ありません。アクセス時にキーが使用される場合、TYPE=KEY ステートメントが存在しないときには、キーは変換されません。キーの変換時に使用するマッチングの詳細 (DFHCNV TYPE=FIELD マクロ) を、このファイルに適用される各変換テンプレートの一部として提供する必要があります。こうしないと、INVREQ 条件がファイル制御 EXEC CICS 要求に返されることがあります。

DFHCNV TYPE=SELECT

これは、DFHCNV TYPE=SELECT マクロ命令のフォーマットです。



TYPE=SELECT
このマクロで定義された比較に当てはまるレコードに適用される変換定義 (DFHCNV TYPE=FIELD マクロ) の開始を示します。定義された比較にレコード内のデータが当てはまらない場合、変換プログラム (DFHCCNV) では、その比較に当てはまるデータが見つかるまで、または OPTION=DEFAULT マクロに到達するまで、次の TYPE=SELECT マクロにスキップします。すべての TYPE=SELECT マクロの後に、1 つ以上の TYPE=FIELD マクロを配置する必要があります。

OPTION={COMPARE|DEFAULT}
基本的な選択オプションを提示します。

COMPARE
このマクロ (OFFSET および DATA オプションまたは XDATA オプション) で定義された比較にレコードが当てはまる場合、データは次の DFHCNV TYPE=FIELD マクロの指定に従って変換されることを示します。

DEFAULT
前の DFHCNV TYPE=SELECT COMPARE マクロで定義された比較にレコードが当てはまらない場合、データは次の DFHCNV TYPE=FIELD マクロの指定に従って変換されることを示します。

TYPE=ENTRY マクロで開始される各リソース項目では、最後の TYPE=SELECT マクロで OPTION=DEFAULT を指定する必要があります。項目内の他の TYPE=SELECT マクロでは OPTION=DEFAULT を指定しないでください。

OPTION=DEFAULT がコード化される際には、次のオプションは無視されます。

OFFSET=nnnn
レコード内の、比較が行われる場所のバイト・オフセットを指定します。最大値は 65535 です。

DATA='dd...dd'

制約事項: テスト対象のデータが「DATATYP=CHARACTER, SOSI=NO」と定義されている場合にのみ使用します。

比較データを EBCDIC 文字ストリングとして指定します。最長 255 文字まで指定できます。CICS は、ASCII からの着信データを、比較データと照らし合わせる前に EBCDIC に変換します。このため、EBCDIC は EBCDIC と比較されます。発信データは EBCDIC フォーマットです。このため、EBCDIC での比較時に、変換は行われません。

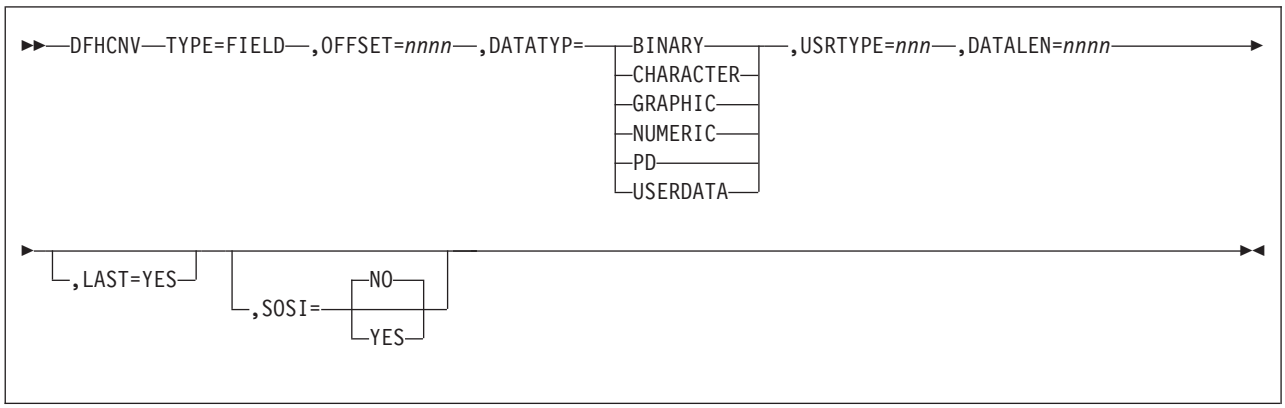
XDATA='xx...xx'

制約事項: DATA オプションを使用していない場合にのみ使用します。

比較データを、16 進数ストリングとして偶数の桁数で指定します。最長で 254 桁まで指定することができます。データは、変換されずに、このフィールドと比較されます。

DFHCNV TYPE=FIELD

これは、DFHCNV TYPE=FIELD マクロ命令のフォーマットです。このマクロは、必要に応じて何度でも実行することができます。



TYPE=FIELD
データ・フィールドの変換の仕様を指定します。レコードの各フィールドには、このようなステートメントが 1 つ必要です。TYPE=SELECT マクロをコード化するまで、TYPE=FIELD マクロをコード化することはできません。

OFFSET=nnnn
レコードまたはキー内の、変換が開始される場所のバイト・オフセットを指定します。最大値は 65535 です。TYPE=KEY 変換では、これはレコードの開始位置からではなく、キー の開始位置からのバイト・オフセットです。ファイルの FCT 項目では、レコードの開始位置からの、キーの最初のバイトのオフセットが定義されます。

DATATYP={CHARACTER|PD|BINARY|USERDATA|GRAPHIC|NUMERIC}
必要な変換のタイプを指定します。

CHARACTER
文字フィールドを指定します。

PD
パック 10 進数データを z/Architecture 形式で指定します。

他の形式のパック 10 進数データは USERDATA 変換の場合に定義する必要があります。ユーザー置き換え可能プログラム DFHUCNV には必要な変換コードを含める必要があります。

BINARY
バイナリー・データをビッグ・エンディアン形式で指定します。

デフォルトでは、バイナリー・データは変換されません。このデフォルト・アクションを指定変更して、異なるバイナリー・アーキテクチャーをサポートする他のプラットフォームからの要求が、同じ変換テーブルを使用して、同じ CICS リソースにアクセスできるようにすることができます。

USERDATA
ユーザー置き換え可能プログラム DFHUCNV によって変換されるデータを指定します。DFHCCNV 変換コードによって、これらのフィールドが迂回されます。下記の USRTYPE オペランドを参照してください。

GRAPHIC
DBCS 文字のみを含むフィールドを指定します。

NUMERIC
ワークステーションに INTEL 形式で保持されている 2 進数フィールド (C 言語の整数データ型など) を z/Architecture 形式に変換する必要があることを示します。整数 (4 バイト) または短整数 (2 バイト) を変換できます。

USRTYPE=nnn
ユーザー置き換え可能変換プログラム DFHUCNV で使用可能にする値を指定します。値は 80 ～ 128 (X'50' ～ X'80') の範囲で指定できます。デフォルト値は 80 (X'50') です。複数のタイプのユーザー定義変換を使用できる場合は、この値を使用して、各フィールドに実行する必要がある変換を DFHUCNV に指定することができます。

DATATYP=USERDATA が指定されていない場合、このオプションは無視されます。

DATALEN=n
変換するデータ・フィールドの長さをバイト単位で指定します。最大値は 65535 です。可変長フィールドには、予想される最大の長さを指定します。

DATATYP=NUMERIC の場合、DATALEN は 2 または 4 に設定する必要があります。

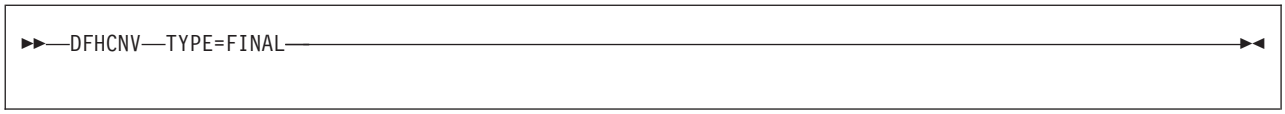
LAST=YES
これが、この TYPE=SELECT ステートメントの最後のフィールド定義であることを指定します。

SOSI=YES|NO
SBCS 文字および DBCS 文字を含む混合ストリングには YES を入力します。SBCS ストリングには NO を入力します。このフィールドは、このマクロに DATATYPE=CHARACTER が入力されている場合にのみ有効です。デフォルトは NO です。

DFHCNV TYPE=FINAL

DFHCNV TYPE=FINAL マクロ命令によってテーブルが終了します。

これは、最後の定義として一度だけ出現します。



マクロのコード化に関するヒント

マクロをコード化すると、CICS が変換テーブルを迅速に処理できるようになり、データ変換のパフォーマンスを向上させることができます。

1. 最も頻繁に使用するリソースの項目を最初に定義し、検索時間を短縮します。
2. 連続する項目で USERDATA フィールドを定義します。こうすることにより、変換プログラムがテンプレートをスキャンするために必要な時間を短縮することができます。
3. 可変長フィールドには、最長の値を定義します。実際のデータの長さとテンプレートの長さが異なる場合、比較と変換は短いほうの長さに適用されます。例えば、実際のデータの長さが 100 バイトである一方で、テンプレートに 120 バイトと示されている場合、最大 100 バイトが変換されます。また、データの長さが 100 バイトである一方で、テンプレートに 80 バイトと示されている場合、80 バイトのみが変換されます。
4. 機能シップされたデータに、CICS Transaction Server for z/OS ではなく、接続されたシステムでのみアクセスする場合は、変換の詳細を指定する必要はありません。例えば、CICS Transaction Server for z/OS ファイルを使用して、複数の ASCII ベースのシステムで共用するデータを保管する場合などです。

第 38 章 ユーザー定義の変換テーブル

DFHCNV TYPE=ENTRY マクロで SRVERCP=USR または USRD を指定する場合は、ユーザー定義の変換テーブルを提供する必要があります。標準の変換プログラム (DFHCCNV) ではこれらのテーブルが使用されます。また、これらのテーブルは、ユーザー置き換え可能変換プログラム DFHUCNV でも使用できます。

DFHCNV マクロ・ソース内の DFHCNV TYPE=INITIAL マクロ以後の任意の場所に、ユーザー定義の変換テーブルを配置します。

ヒント: ソースを読みやすくするために、変換テーブルは DFHCNV TYPE=FINAL マクロの後に配置することをお勧めします。

SRVERCP=USR

ASTOEB および EBTOAS という名前の 2 つの文字変換テーブルを提供する必要があります。

各テーブルの長さは 256 バイトにする必要があります。ASTOEB は ASCII から EBCDIC への変換時に使用され、EBTOAS は EBCDIC から ASCII への変換時に使用されます。16 進値の文字バイトは、文字の変換値を取得する際に、変換テーブルで相対位置として使用されます。 404 ページの図 101 はこの処理を示しています。

© Copyright IBM Corp. 1977, 2009

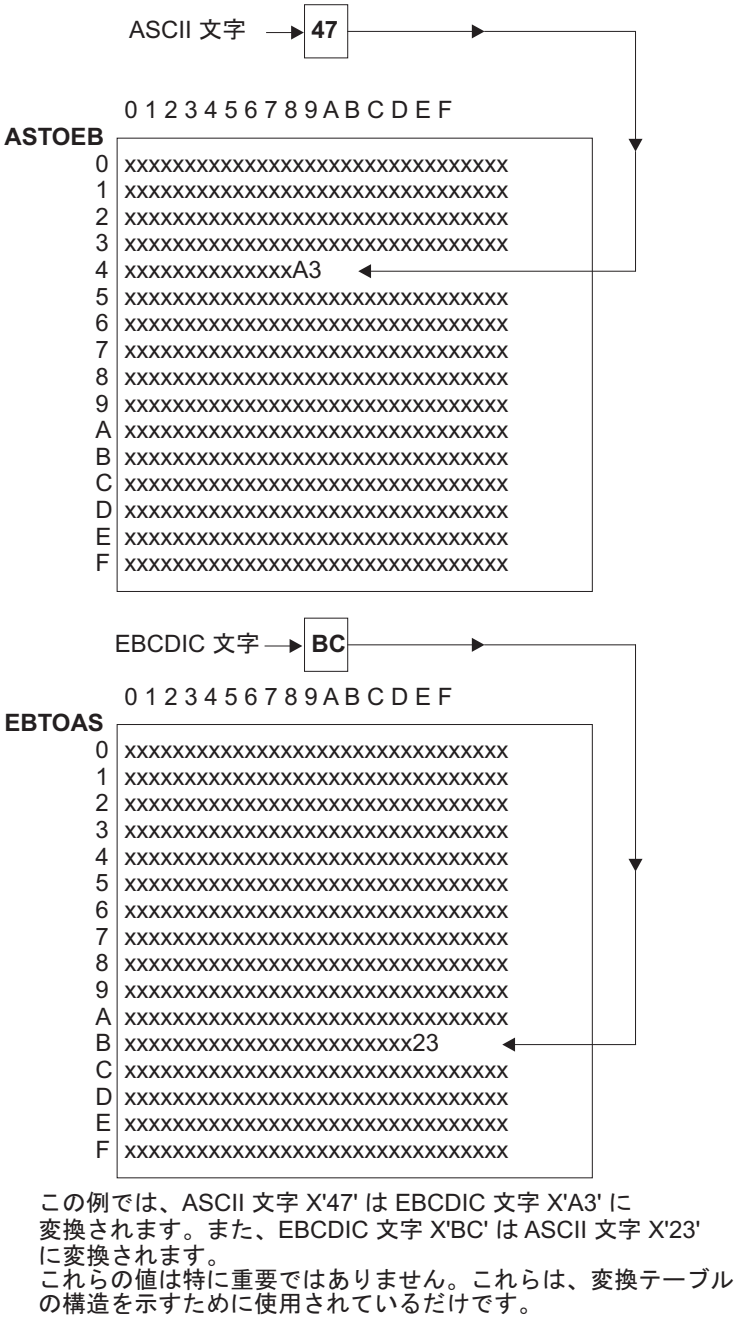


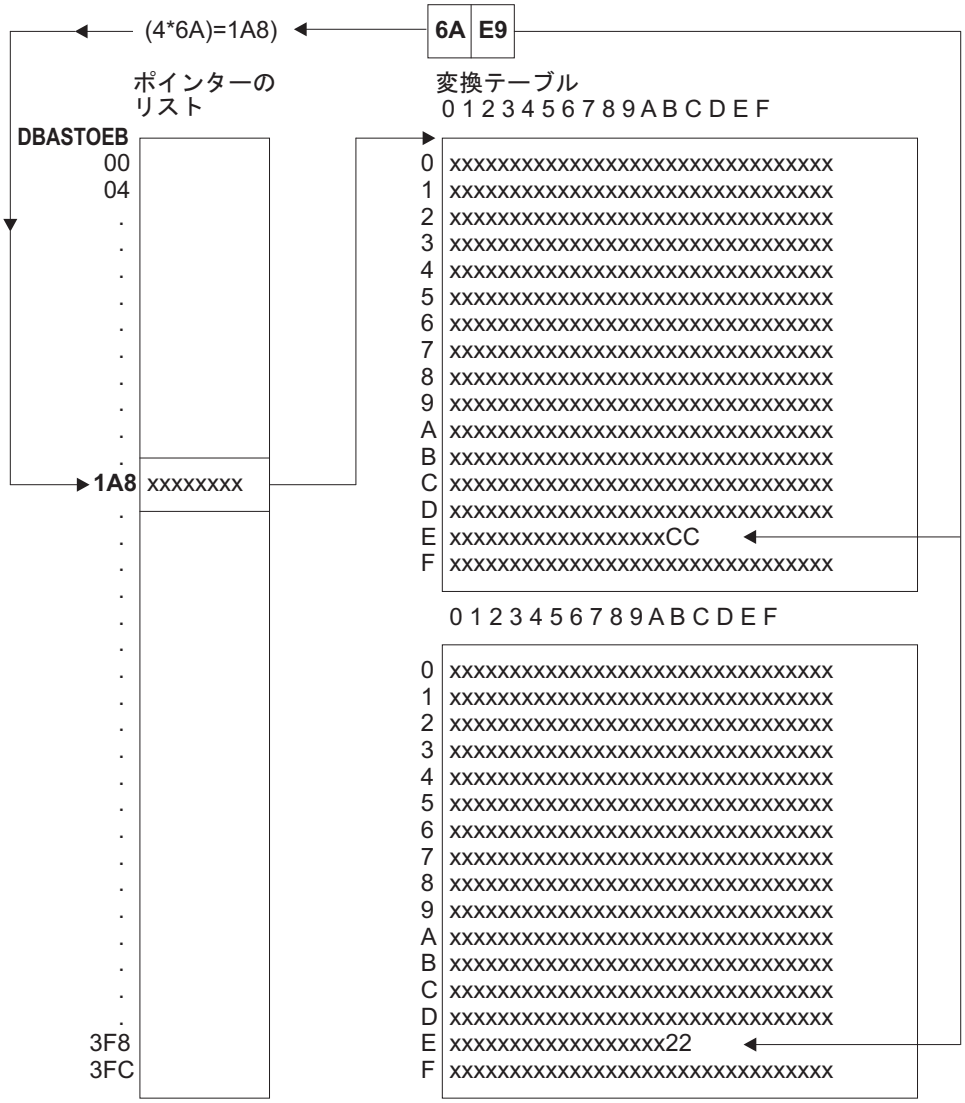
図 101. SBCS 変換テーブルの構造

SRVERCP=USRD

DFHCNV ソースで、DBASTOEB および DBEBTOAS という名前の DBCS 文字変換テーブルを提供する必要があります。変換テーブルは、DFHCNV TYPE=INITIAL マクロの後であれば、ソース内の任意の場所に配置できます。各テーブルは、4 バイトのポインターを 256 個、および 256 バイトの変換テーブルのペアを 256 個含むリストにする必要があります。DBCS 文字の最初のバイトは、ポインター・リストの索引として使用されます。DBCS 文字の最初のバイトをリスト内における 16 進数の相対位置として使用することにより、見つかったポインターが 256 バイトの変換テーブル・ペアのアドレスとなります。DBCS 文字の 2 番目のバイトは、変換

された DBCS 文字の最初と 2 番目のバイトを取得する際に、256 バイトの各変換テーブル内における相対位置として使用されます。図 102 はこの処理を示しています。

また、前述の USR で指定した SBCS 変換テーブルを提供する必要があります。



この例では、2 バイト文字 X'6AE9' が X'CC22' に変換されます。
ポインター・リストの相対位置 6A にある値は、256バイトの変換テーブル・ペアのアドレスになります。これらのテーブルの相対位置 E9 のバイト値は、それぞれ X'CC' と X'22' です。これらはランダム値であり、図で使用されているだけです。
ポインター・リストの名前が DBASTOEB であるため、これは ASCII から EBCDIC への変換です。ASCII から EBCDIC への変換テーブルには、256 バイトのテーブルのペアが 256 個含まれています。つまり、2 バイト文字の最初のバイトの値につき、1 つのペアが含まれています。
DBEBTOAS は、これと同様の EBCDIC から ASCII への変換テーブルの名前です。

図 102. DEBUTS 変換テーブルの構造

無効な、または未定義の DBCS 文字

ASCII と EBCDIC の特定コード範囲が、有効な DBCS コードです。この範囲以外の 2 バイトの値は、無効な DBCS 文字となります。コード・ページ・アーキテクチャーで定義されているように、提供されている変換テーブルでは、無効な DBCS 文字は X'FFFF' に変換されます。

有効なコード範囲では、数千個の 2 バイト値が実際の DBCS 文字として定義されています。有効なコード範囲内の 2 バイト値で、DBCS 文字として定義されていないものは、未定義の DBCS 文字になります。

ユーザー定義のテーブルでは、無効な文字と未定義の文字についての上記規則に従う必要があります。

第 39 章 マクロの例

次に、データ変換マクロの使用例を示します。

図 103 は、VSAM99 というファイルのレコード・レイアウトの例を示しています。キーは長さ 6、オフセット 0 です。また、レコードには再定義が含まれていません。

```
02  FILEREC.
  03  STAT         PIC X.
  03  NUMB         PIC X(6).
  03  NAME         PIC X(20).
  03  ADDRX        PIC X(20).
  03  PHONE        PIC X(8).
  03  DATEX        PIC X(8).
  03  AMOUNT       PIC X(8).
  03  COMMENT      PIC X(9).
  03  COUNTER1     PIC 9999 USAGE COMP-4.
  03  COUNTER2     PIC 9999 USAGE COMP-4.
  03  ADDLCMT      PIC X(30).
```

図 103. VSAM99 のレコード・レイアウト

図 104 は、ファイル VSAM99 のすべての変換マクロを示しています。 408 ページの図 105 は、同じタイプの隣接フィールドを結合して、同じ変換をより簡単に表したものです。

```
    DFHCNV TYPE=INITIAL,CLINTCP=437,SRVERCP=037
    DFHCNV TYPE=ENTRY,RTYPE=FC,RNAME=VSAM99
    DFHCNV TYPE=KEY
    DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=6,LAST=YES
    DFHCNV TYPE=SELECT,OPTION=DEFAULT
    DFHCNV TYPE=FIELD,OFFSET=00,DATATYP=CHARACTER,DATALEN=1
    DFHCNV TYPE=FIELD,OFFSET=01,DATATYP=CHARACTER,DATALEN=6
    DFHCNV TYPE=FIELD,OFFSET=07,DATATYP=CHARACTER,DATALEN=20
    DFHCNV TYPE=FIELD,OFFSET=27,DATATYP=CHARACTER,DATALEN=20
    DFHCNV TYPE=FIELD,OFFSET=47,DATATYP=CHARACTER,DATALEN=8
    DFHCNV TYPE=FIELD,OFFSET=55,DATATYP=CHARACTER,DATALEN=8
    DFHCNV TYPE=FIELD,OFFSET=63,DATATYP=CHARACTER,DATALEN=8
    DFHCNV TYPE=FIELD,OFFSET=71,DATATYP=CHARACTER,DATALEN=9
    DFHCNV TYPE=FIELD,OFFSET=80,DATATYP=BINARY,DATALEN=2
    DFHCNV TYPE=FIELD,OFFSET=82,DATATYP=BINARY,DATALEN=2
    DFHCNV TYPE=FIELD,OFFSET=84,DATATYP=CHARACTER,DATALEN=30,LAST=YES
    DFHCNV TYPE=FINAL
```

図 104. VSAM99 の説明

© Copyright IBM Corp. 1977, 2009

```
DFHCNV TYPE=INITIAL,CLINTCP=437,SRVERCP=037
DFHCNV TYPE=ENTRY,RTYPE=FC,RNAME=VSAM99
DFHCNV TYPE=KEY
DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=6,LAST=YES
DFHCNV TYPE=SELECT,OPTION=DEFAULT
DFHCNV TYPE=FIELD,OFFSET=00,DATATYP=CHARACTER,DATALEN=80
DFHCNV TYPE=FIELD,OFFSET=80,DATATYP=BINARY,DATALEN=4
DFHCNV TYPE=FIELD,OFFSET=84,DATATYP=CHARACTER,DATALEN=30,LAST=YES
DFHCNV TYPE=FINAL
```

図 105. VSAM99 の要約バージョン

注: データ型が同じ場合でも、隣接フィールドを結合する際には注意が必要です。NUMERIC フィールドは結合しないでください。1 つ以上の CHARACTER で SOSI=YES が指定されている場合は、CHARACTER として定義されたフィールドを結合しないでください。USERDATA フィールドを結合できるかどうかは、ユーザー定義のデータ構造と変換コードにより異なります。

図 106 は、ファイル VSAM99 の再定義されたレコード・レイアウトを示しています。 409 ページの図 107 は、図 106 の再定義されたレコード・レイアウトの一連の変換マクロを示しています。

```
02  FILEREC.
  03  STAT         PIC X.
  03  NUMB         PIC X(6).
  03  NAME         PIC X(20).
  03  ADDRX        PIC X(20).
  03  PHONE        PIC X(8).
  03  DATEX        PIC X(8).
  03  AMOUNT       PIC X(8).
  03  COMMENT      PIC X(9).
  03  VARINF1.
  03  COUNTER1     PIC 9999 USAGE COMP-4.
  03  COUNTER2     PIC 9999 USAGE COMP-4.
  03  ADDLCMT      PIC X(30).
  03  VARINF2 REDEFINES VARINF1.
  03  COUNTER1     PIC 9999 USAGE COMP-4.
  03  COUNTER2     PIC 9999 USAGE COMP-4.
  03  COUNTER3     PIC 9999 USAGE COMP-4.
  03  COUNTER4     PIC 9999 USAGE COMP-4.
  03  ADDLCMT2     PIC X(26).
```

図 106. VSAM99 の再定義されたレコード・レイアウト

```
   DFHCNV TYPE=INITIAL
   DFHCNV TYPE=ENTRY,RTYPE=FC,RNAME=VSAM99
   DFHCNV TYPE=KEY
   DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=6,LAST=YES
*
*  offset 00 が文字 'X' の場合は、以下の
*  変換定義を使用します。*

   DFHCNV TYPE=SELECT,OPTION=COMPARE,OFFSET=00,DATA='X'
   DFHCNV TYPE=FIELD,OFFSET=00,DATATYP=CHARACTER,DATALEN=80
   DFHCNV TYPE=FIELD,OFFSET=80,DATATYP=BINARY,DATALEN=4
   DFHCNV TYPE=FIELD,OFFSET=84,DATATYP=CHARACTER,DATALEN=30,LAST=YES
*
*  それ以外の場合は、以下のデフォルトの
*  変換定義を使用します。*
   DFHCNV TYPE=SELECT,OPTION=DEFAULT
   DFHCNV TYPE=FIELD,OFFSET=00,DATATYP=CHARACTER,DATALEN=80
   DFHCNV TYPE=FIELD,OFFSET=80,DATATYP=BINARY,DATALEN=8
   DFHCNV TYPE=FIELD,OFFSET=88,DATATYP=CHARACTER,DATALEN=26,LAST=YES
   DFHCNV TYPE=FINAL
```

図 107. VSAM99 の再定義されたレコード・レイアウトの説明

410 ページの図 108 は、ユーザー定義の変換テーブルである EBTOAS と ASTOEB を示しています。これらのテーブルは、アセンブラーに送信されるソースの中で、DFHCNV マクロの後ろに配置されます。

```
*
LABL1    DFHCNV TYPE=INITIAL,CLINTCP=437,SRVERCP=037
*
         DFHCNV TYPE=ENTRY,RTYPE=FC,RNAME=VSAM80
         DFHCNV TYPE=KEY
         DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=BINARY,DATALEN=2
         DFHCNV TYPE=FIELD,OFFSET=2,DATATYP=CHARACTER,DATALEN=4,   X
               LAST=YES
LABLX    DFHCNV TYPE=SELECT,OPTION=COMPARE,OFFSET=6,XDATA='C1C2C3'
         DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=BINARY,DATALEN=2
         DFHCNV TYPE=FIELD,OFFSET=2,DATATYP=CHARACTER,DATALEN=4
         DFHCNV TYPE=FIELD,OFFSET=9,DATATYP=CHARACTER,DATALEN=8,   X
               LAST=YES

.
.
.
         DFHCNV TYPE=ENTRY,RTYPE=TS,RNAME=ABCD
         DFHCNV TYPE=SELECT,OPTION=DEFAULT
         DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=40
         DFHCNV TYPE=FIELD,OFFSET=40,DATATYP=BINARY,DATALEN=4,     X
               LAST=YES
LABLN    DFHCNV TYPE=FINAL
*
*  EXAMPLE OF A USER-DEFINED CONVERSION TABLE EBCDIC to ASCII
EBTOAS   DC    XL16'000102030405060708090A0B0C0D0E0F'
         DC    XL16'101112131415161718191A1B1C1D1E1F'
         DC    XL16'202122232425262728292A2B2C2D2E2F'
         DC    XL16'303132333435363738393A3B3C3D3E3F'
         DC    XL16'404142434445464748494A4B4C4D4E4F'
         DC    XL16'505152535455565758595A5B5C5D5E5F'
         DC    XL16'606162636465666768696A6B6C6D6E6F'
         DC    XL16'707172737475767778797A7B7C7D7E7F'
         DC    XL16'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'
         DC    XL16'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'
         DC    XL16'A0A1E2E3E4E5E6E7E8E9AAABACADAEAF'
         DC    XL16'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'
         DC    XL16'C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF'
         DC    XL16'D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF'
         DC    XL16'E0E1E2A3E4E5E6E7E8E9EAEBECEDEEEF'
         DC    XL16'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'
*
*  EXAMPLE OF A USER-DEFINED CONVERSION TABLE ASCII to EBCDIC
*
ASTOEB   DC    XL16'000102030405060708090A0B0C0D0E0F'
         DC    XL16'101112131415161718191A1B1C1D1E1F'
         DC    XL16'202122232425262728292A2B2C2D2E2F'
         DC    XL16'303132333435363738393A3B3C3D3E3F'
         DC    XL16'404142434445464748494A4B4C4D4E4F'
         DC    XL16'505152535455565758595A5B5C5D5E5F'
         DC    XL16'606162636465666768696A6B6C6D6E6F'
         DC    XL16'707172737475767778797A7B7C7D7E7F'
         DC    XL16'80818283848586878889CACBCCCDCECF'
         DC    XL16'90919293949596979899A9B9C9D9E9F'
         DC    XL16'A0A1A2A3A4A5A6A7A8A9AAABACADAEAF'
         DC    XL16'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'
         DC    XL16'C081828384858687888898A8B8C8D8E8F'
         DC    XL16'D091929394959697989899A9B9C9D9E9F'
         DC    XL16'E0E1A2A3A4A5A6A7A8A9EAEBECEDEEEF'
         DC    XL16'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'
         END   DFHCNVBA
```

図 108. SBCS - ユーザー定義の変換テーブル

第 40 章 変換プログラムのアセンブルとリンク・エディット

標準のプロシージャーである DFHAUPLE と DFHAUPLK のいずれかを使用して、DFHCNV テーブルをアセンブルできます。

このタスクについて

CICS 仮想ストレージの使用方法を最適化するには、DFHCNV テーブルと DFHUCNV プログラムをリンク・エディットします。このとき、MODE ステートメントで AMODE(31) および RMODE(ANY) を指定します。こうすると、十分な CICS ストレージが使用可能な場合に、テーブルとプログラムが 16 MB 境界の上でロードされます。

© Copyright IBM Corp. 1977, 2009

第 41 章 ユーザー置き換え可能変換プログラム

このセクションでは、ユーザー置き換え可能データ変換プログラムについて説明します。

ユーザー指定の変換プログラム

デフォルトのユーザー置き換え可能変換プログラムである DFHUCNV に、1 つ以上のユーザー指定の変換プログラムを置き換えることができます。

DFHUCNV は、次のような場合に呼び出されます。

- リソースの変換テンプレートが定義されていない場合、または
- リソースの変換テンプレートが定義されており、そのテンプレートで USREXIT=YES が指定されている場合。

ユーザー指定の変換プログラムは、次のような場合に呼び出されます。

- リソースの変換テンプレートが定義されており、そのテンプレートで USREXIT=*ユーザー・プログラム* が指定されている場合。

  ユーザー・プログラム は、ユーザー提供の変換プログラムの名前です。

DFHUCNV への入力

提供バージョンの DFHUCNV の最初のステートメントは DFHCNV TYPE=DSECT マクロです。これは、パラメーター・リストと変換テンプレートについて説明する DSECT を生成します。

DFHUCNV は、DFHCNV TYPE=DSECT を次の形式で指定することによって開始されます。

```
DFHCNV  TYPE=DSECT
```

DFHCNV TYPE=DSECT マクロによって、次のものが生成されます。

- DFHUNVDS DSECT。DFHCCNV で渡された COMMAREA 内のパラメーター・リストをマップします。
- フィールド変換レコードのためのアセンブラー DSECT。これは、テンプレートの基本コンポーネントです (418 ページの図 111 を参照)。
- リソース・タイプとフィールド・タイプが等価になります。

パラメーター・リスト (DFHUVNDS)

DFHUNVDS DSECT は、COMMAREA で、DFHUCNV に渡されたパラメーター・リストをマップします。

パラメーターがゼロの場合、データは使用できません。*リソースの変換テンプレートを作成していない場合、DFHUCNV が呼び出されますが、パラメーター・リストの以下のフィールドのみにデータが入力されます。*

© Copyright IBM Corp. 1977, 2009

- UNVRSTP
- UNVRNMP
- UNVDIRP
- UNVOVLY

```
DFHUNVDS DSECT
UNVRSTP  DS    AL4                PTR-TO-RESOURCE TYPE
UNVRNMP  DS    AL4                PTR-TO-RESOURCE NAME
UNVDIRP  DS    AL4                PTR-TO-CONVERSION DIRECTIVE
CNVRQATE EQU   X'02'              REQUEST ASCII TO EBCDIC
CNVRPETA EQU   X'04'              RESPONSE EBCDIC TO ASCII
UNVDTMP  DS    AL4                PTR-TO-DATA CONV TEMPLATE
UNVDLNP  DS    AL4                PTR-TO-DATA TEMPLATE LENGTH
UNVKTMP  DS    AL4                PTR-TO-KEY CONV TEMPLATE
UNVKLNP  DS    AL4                PTR-TO-KEY TEMPLATE LENGTH
UNVATEP  DS    AL4                PTR-TO-ASCII/EBCDIC TRANS TABLE
UNVETAP  DS    AL4                PTR-TO-EBCDIC/ASCII TRANS TABLE
UNVATED  DS    AL4                PTR-TO-DBCS ASCII/EBCDIC TRANS TABLE
UNVETAD  DS    AL4                PTR-TO-DBCS EBCDIC/ASCII TRANS TABLE

UNVOVLY  DS    0H                 OVERLAY SECTION
         ORG   UNVOVLY            TS REQUEST OVERLAY
UNVTSDP  DS    AL4                PTR-TO-TS DATA
UNVTSLNP DS    AL4                PTR-TO-TS DATA LENGTH
         ORG   UNVOVLY            TD REQUEST OVERLAY
UNVTDDP  DS    AL4                PTR-TO-TD DATA
UNVTDLNP DS    AL4                PTR-TO-TD DATA LENGTH
         ORG   UNVOVLY            IC REQUEST OVERLAY
UNVICDP  DS    AL4                PTR-TO-IC DATA
UNVICLNP DS    AL4                PTR-TO-IC DATA LENGTH
         ORG   UNVOVLY            PC REQUEST OVERLAY
UNVPCDP  DS    AL4                PTR-TO-PC DATA
UNVPCLNP DS    AL4                PTR-TO-PC DATA LENGTH
         ORG   UNVOVLY            FC REQUEST OVERLAY
UNVFCDP  DS    AL4                PTR-TO-FC DATA
UNVFCLNP DS    AL4                PTR-TO-FC DATA LENGTH
UNVFCKP  DS    AL4                PTR-TO-FC KEY
UNVFCKLP DS    AL4                PTR-TO-FC KEY  LENGTH
         ORG   ,
UNVMRTNE DS    A                  PTR-TO-MBCS TRANSLATION ROUTINE
UNVCLIDP DS    AL4                A "client" CCSID
*                                   (for example, 00819)
UNVSRIDP DS    AL4                A "server" CCSID
*                                   (for example, 00285)
```

図 109. DFHUCNV に渡されたパラメーター・リストをマップする DFHUNVDS—DSECT

次に、パラメーターについて詳しく説明します。

UNVRSTP
この要求が参照するリソースを示す 1 バイトのリソース・タイプを指します。リソース・タイプの意味は、DSECT DFHCNVDS で定義されます。リソース・タイプは FC、IC、TS、TD、および PC です。

UNVRNMP
リソース名 (必要に応じてブランクが埋め込まれる) を含む 8 バイトのフィールドを指します。例えば、次のようになります。

- FC 要求の場合は 8 バイトのファイル名
- TS 要求の場合は 8 バイトの TS キュー名

- TD 要求の場合は 4 バイトの TD キュー名
- IC 要求の場合は 4 バイトのトランザクション名
- PC 要求の場合は 8 バイトのプログラム名

UNVDIRP
必要な変換を示す 1 バイトのフィールドを指します。
- CNVRQATE (X'02') は、クライアント・エンコードからサーバー・エンコードに変換する必要がある要求を示します。
- CNVRPETA (X'04') は、サーバー・エンコードからクライアント・エンコードへの変換が必要な応答を示します。

UNVDTMP
このリソースと一致させるために、CICS によって検出された変換テンプレートの開始位置を指します。UNVDTMP がゼロの場合、テンプレートは検出されていません。

UNVDLNP
変換テンプレートの長さを示すフィールドを指します。フィールドの長さは次のとおりです。
- CICS Transaction Server for z/OS の場合はフルワード
- 他のプラットフォームではハーフワード

UNVKTMP (ファイル制御要求のみ)
要求または応答のキーの部分について、CICS によって検出されたテンプレートの開始位置を指します。UNVKTMP がゼロの場合、キー・テンプレートがないか、レコードが相対レコード番号または相対バイト・アドレスによってアクセスされています。

UNVKLNP (ファイル制御要求のみ)
キー変換テンプレートの長さを示すフィールドを指します。フィールドの長さは次のとおりです。
- CICS Transaction Server for z/OS の場合はフルワード
- 他のプラットフォームではハーフワード

UNVATEP
クライアント・エンコードからサーバー・エンコードに文字データを変換する際に使用される、256 バイトの SBCS 変換テーブルを指します。

UNVETAP
サーバー・エンコードからクライアント・エンコードに文字データを変換する際に使用される、256 バイトの SBCS 変換テーブルを指します。

UNVATED
クライアント・エンコードからサーバー・エンコードに文字データを変換する際に使用される DBCS 変換テーブルを指します。

UNVETAD
サーバー・エンコードからクライアント・エンコードに文字データを変換する際に使用される DBCS 変換テーブルを指します。

オーバーレイ・セクションはリソース・タイプにより異なります。

TS 要求:

UNVTSDP
読み取りまたは書き込み対象の TS レコードの開始位置を指します。フィールドの長さは次のとおりです。
- CICS Transaction Server for z/OS の場合はフルワード
- 他のプラットフォームではハーフワード

UNVTSLNP
TS レコードの長さを示すフィールドを指します。

TD 要求:

UNVTDDP
読み取りまたは書き込み対象の TD レコードの開始位置を指します。

UNVTDLNP
TD レコードの長さを示すフィールドを指します。フィールドの長さは次のとおりです。
- CICS Transaction Server for z/OS の場合はフルワード
- 他のプラットフォームではハーフワード

IC 要求:

UNVICDP
IC START 要求の「開始」領域を指します。

UNVICLNP
「開始」領域の長さを示すフィールドを指します。フィールドの長さは次のとおりです。
- CICS Transaction Server for z/OS の場合はフルワード
- 他のプラットフォームではハーフワード

PC 要求:

UNVPCDP
提供される COMMAREA の開始位置を指します。

UNVPCLNP
COMMAREA の長さを示すフィールドを指します。フィールドの長さは次のとおりです。
- CICS Transaction Server for z/OS の場合はフルワード
- 他のプラットフォームではハーフワード

FC 要求:

UNVFCDP
読み取りまたは書き込み対象のファイル制御レコードの開始位置を指します。

UNVFCLNP
ファイル制御レコードの長さを示すフィールドを指します。フィールドの長さは次のとおりです。
- CICS Transaction Server for z/OS の場合はフルワード
- 他のプラットフォームではハーフワード

UNVFCKP
読み取りまたは書き込み対象のファイル制御レコードのキーの開始位置を指します。

UNVFCKLP
キーの長さを示すフィールドを指します。フィールドの長さは次のとおりです。
- CICS Transaction Server for z/OS の場合はフルワード
- 他のプラットフォームではハーフワード

UNVMRTNE
MBCS コード・ページにおける変換に使用する必要がある変換ルーチンを指します。関連クライアント・コード・ページは 954、964、および 970 です。

ルーチンでは、Register 1 が、DFHUNVM DSECT によって定義された構造を指していることを前提としています。

```
DFHUNVM DSECT
UNVMTABP  DS    AL4           UNVATED または UNVETAD に値を設定
UNVMINP   DS    AL4           ソース・データのアドレス
INVMINL   DS    FL4           ソース・データの長さ
UNVMOUTP  DS    AL4           ターゲット・バッファーのアドレス
UNVMOUTL  DS    FL4           ターゲット・バッファーの長さ
```

UNVCLIDP
「クライアント」コード・ページに対応する、IBM 定義の CCSID (00819 など) を指定するフルワード・フィールドを指します。

UNVSRIDP
「サーバー」コード・ページに対応する、IBM 定義の CCSID (00285 など) を指定するフルワード・フィールドを指します。

変換テンプレートとキー・テンプレート

COMMAREA では、UNVDTMP フィールドと UNVDLNP フィールドが変換テンプレートとその長さを指します。

UNVKTMP フィールドと UNVKLNP フィールドは、キー・テンプレートとその長さを指します。図 110 は、これらのフィールドの使用方法とその意味を示しています。

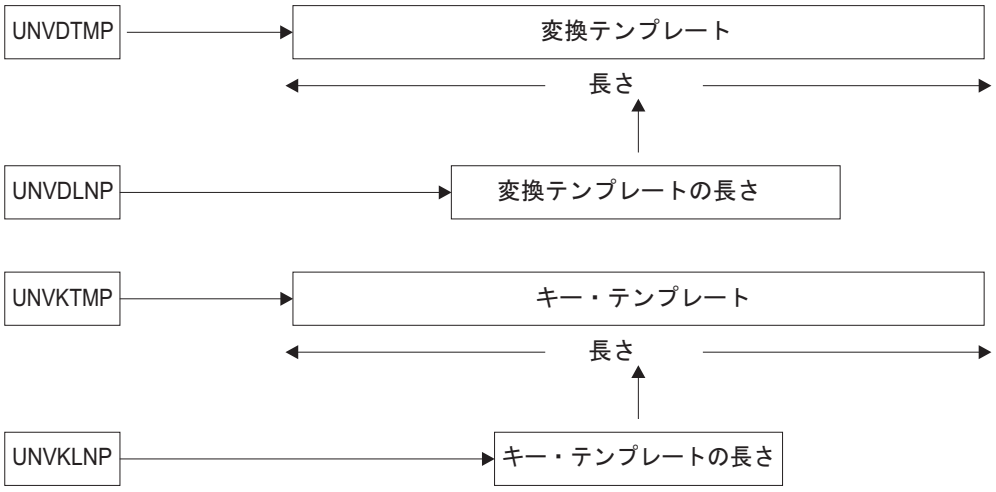


図 110. パラメーター・フィールドと変換テンプレート

各種類のテンプレートは、フィールド変換レコードで構成されています。フィールド変換レコードは、データ・レコードまたはキーの各フィールドにつき 1 つずつあ

ります。『フィールド変換レコード』に示されているように、フィールド変換レコードのレイアウトはすべて同じです。これらは、提供されている DSECT、DFHCNVDS によってマップされます (419 ページの『フィールド変換レコード用の DFHCNVDS DSECT』を参照)。図 111 は、テンプレート、フィールド変換レコード、および DFHCNVDS 間の関係を示しています。この図は、6 つのフィールドを含むデータ・レコードまたはキーのテンプレートにおいて、最初のフィールド変換レコードをオーバーレイする DFHCNVDS を示しています。

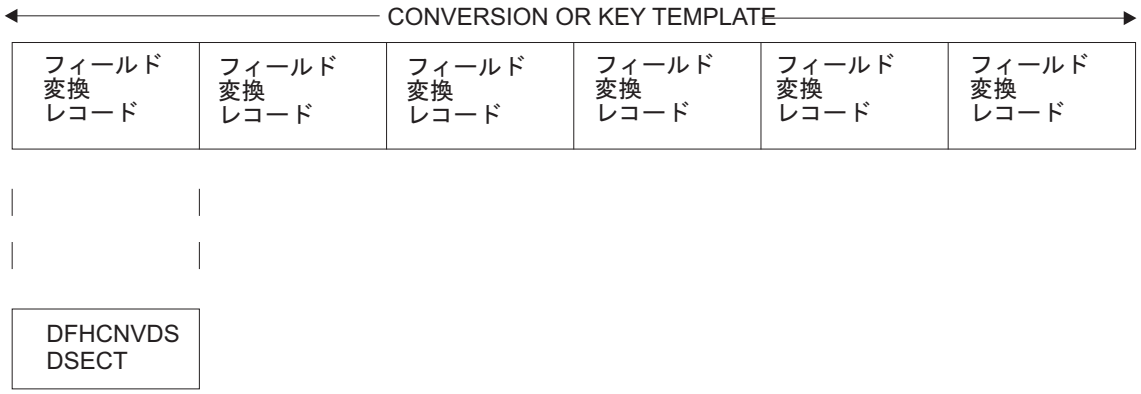


図 111. フィールド変換レコードと変換テンプレートまたはキー・テンプレート

フィールド変換レコード

ここでは、フィールド変換レコードのレイアウトについて説明します。

フィールド変換レコードのレイアウトは次のとおりです。

表 65. フィールド変換レコードのレイアウト

CNVRLEN	CNVRTYPE	予約	CNVDATTY	CNVDATAO	CNVDATAL
レコード長	レコード・タイプ	予約	データ型	データの相対位置	データ長
バイト 1	バイト 2	バイト 3	バイト 4	バイト 5 ～ 8	バイト 9 ～ 12

表 65 では、レコード長とレコード・タイプは、フィールド変換レコードの長さとタイプを指しています。一番上の行に表示されている名前は、フィールド変換フィールドをマップする DSECT DFHCNVDS で使用される名前です (420 ページの図 112 を参照)。テンプレートには、必要な数のフィールド変換レコードを含めることができます。このため、データ・レコードまたはキー内のすべてのフィールドについて説明することができます。

DFHUCNV では CNVRLEN は X'0C' であり、CNVRTYPE は常に X'04' (フィールド) です。DFHUCNV では、ユーザー仕様に従って、CNVDATTY の値を X'50' ～ X'80' の範囲で解釈し、適切な変換を適用する必要があります。DFHUCNV では、X'50' ～ X'80' の範囲に含まれない CNVDATTY の値を含むフィールドを無視する必要があります。

DFHCNVDS の EQUATE

DFHCNVDS には、変換プログラムで利用できる EQUATE が含まれています。

パラメーター・リストで処理されるリソース・タイプ

```
CNVFC           FILE CONTROL
CNVTS           TEMPORARY STORAGE
CNVTD           TRANSIENT DATA
CNVIC           INTERVAL CONTROL
CNVPC           PROGRAM CONTROL
```

テンプレート内のフィールド・タイプ

追加された 2 つの EQUATE、つまり DTUSRMIN と DTUSRMAX によって、ユーザー定義に予約されているデータ型の範囲の制限 (X'50' 〜 X'80') が定義されます。DFHUCNV で、インストールに使用できる、この範囲のデータ型をすべて処理できることを確認してください。

```
DTBIN           BINARY
DTPD            PACKED DECIMAL
DTCHAR          CHARACTER
DTMIX           MIXED CHARACTER
DTDBCS          DBCS CHARACTER
DTNUM           INTEL INTEGER
```

提供されている DFHUCNV プログラムには、CNVTS、DTUSRMIN、および DTUSRMAX の使用例が含まれています (420 ページの『提供されているユーザー置き換え可能変換プログラム』を参照)。

フィールド変換レコード用の DFHCNVDS DSECT

```
DFHCNVDS DSECT
*
*        PROVIDES A MAPPING OF THE FIELD CONVERSION RECORDS USED
*        WHEN DECIDING WHETHER TO CONVERT USER DATA.
*        A SET OF FIELD DEFINITIONS MAKE UP A TEMPLATE
*
CNVRLEN  DS    AL1                 LENGTH OF THIS RECORD
CNVRTYPE DS    XL1                 TYPE OF RECORD
*
*        EQUATES FOR RECORD TYPES
*
CNVTFLD  EQU   X'04'               FIELD (ONLY VALID TYPE IN
*                                  TEMPLATE)
CNVOVLY  DS    0H
**
**
         ORG   CNVOVLY             TYPE FIELD
         DS    XL1                 RESERVED
CNVDATTY DS    XL1                 DATA TYPE
*
*        EQUATES FOR DATA TYPES
*
DTBIN    EQU   X'01'               BINARY
DTPD     EQU   X'02'               PACKED DECIMAL
DTCHAR   EQU   X'03'               CHARACTER
DTMIX    EQU   X'04'               MIXED CHARACTER
DTDBCS   EQU   X'05'               DBCS
DTNUM    EQU   X'06'               NUMERIC
DTUSRMIN EQU   X'50'               MINIMUM USER DATA TYPE
DTUSRMAX EQU   X'80'               MAXIMUM USER DATA TYPE
*
CNVDATAO DS    AL4                 DATA OFFSET
CNVDATAL DS    AL4                 DATA LENGTH
**
*
*        EQUATES FOR RESOURCE TYPES
*
CNVFC    EQU   X'01'               FILE CONTROL
CNVTS    EQU   X'02'               TEMP STORAGE
CNVTD    EQU   X'03'               TRANS DATA
CNVIC    EQU   X'05'               INTERVAL CONTROL
CNVPC    EQU   X'06'               PROGRAM CONTROL
```

図 112. DFHUCNV に渡された変換/キー・テンプレートをマップする DFHCNVDS DSECT

提供されているユーザー置き換え可能変換プログラム

提供されているバージョンの DFHUCNV では、TS のリソース・タイプがチェックされます。リソース・タイプが見つかると、渡されたテンプレートがスキャンされ、ユーザー・データ範囲で、タイプが定義されたフィールドが検索されます。フィールドが存在する場合、DFHUCNV によってそれらが文字に変換されます。変換コードは、要件に応じて書き換えることができます。

421 ページの図 113 には、CICS バージョン 2.2 以降で提供されている DFHUCNV のバージョンが示されています。これらのバージョンは、両方ともアセンブラーで作成されています。

提供されているバージョンの DFHUCNV とその紹介コメントを確認してください。こうすることにより、独自の変換プログラムを作成することができます。プログラムでは、31 ビットのアドレスを処理できるようにする必要があります。

ユーザー置き換え可能変換プログラム

このトピックでは、ユーザー置き換え可能変換プログラムのソースを示します。

```
*   MODULE NAME = DFHUCNV
*
* DESCRIPTIVE NAME = C.I.C.S./.......
**       CICS TS for Windows USER CONVERSION SAMPLE PROGRAM
**
*
* TRANSACTION NAME = Cxxx
**       NOT A TRANSACTION
*
* STATUS =n.n.n
*
* FUNCTION =
*        THIS IS A SAMPLE PROGRAM FOR USER DATA CONVERSION
*        IT IS INVOKED AS A RESULT OF A FUNCTION
*        SHIPPED REQUEST OR RESPONSE VIA THE LU2 REMOTE SERVER
*        OR LU6.2 HOST MIRROR PROGRAM. IT IS ACTUALLY  CICS
*        LINKED FROM DFHCCNV TO ALLOW A USER PROGRAM TO
*        CONVERT DATA OF TYPE USERDATA AS DEFINED IN THE
*        CICS TS for Windows CONVERSION MACROS (DFHCNV).
*
*        THIS PROGRAM IS CALLED FOR EACH EXEC CICS REQUEST/RESPONSE
*        FOR WHICH DATA EXISTS FOR CONVERSION FROM ASCII TO EBCDIC.
*        IF A REQUEST DOES NOT CONTAIN ANY SUCH DATA, THIS PROGRAM
*        IS NOT INVOKED. THE PROGRAM IS INVOKED BEFORE THE CICS
*        CONVERSION PROGRAM (DFHCCNV) ATTEMPTS ANY CONVERSION
*        INBOUND FROM CICS TS for Windows (ASCII TO EBCDIC) OR
*        OUTBOUND FROM CICS TS for Windows (EBCDIC TO ASCII).
*
*        A COMMAREA IS  PASSED WITHIN WHICH IS A SERIES OF POINTERS
*        TO INFORMATION THAT CAN BE USED BY THE PROGRAM TO
*        DETERMINE HOW TO CONVERT ANY RELEVANT DATA. THIS PROGRAM
*        SHOULD ONLY CONVERT DATA OF TYPE USERDATA AS INDICATED IN
*        THE CONVERSION TEMPLATES. ANY DATA OF TYPE CHARACTER
*        WILL BE CONVERTED BY THE CICS CONVERSION MODULE DFHCCNV.
*
*        SEE A LATER DESCRIPTION FOR WHAT THE SAMPLE DOES
*
```

図 113. DFHUCNV: z/OS から Windows リンクへのユーザー置き換え可能変換プログラム (パート 1)

```
* NOTES :
*    DEPENDENCIES = S/370
*        IN A CICS MVS/XA ENVIRONMENT, THIS PROGRAM SHOULD  BE
*          LINKED WITH RMODE(ANY) AND RMODE(31). ALL ADDRESSES SHOULD
*          BE TREATED AS 31 BIT.
*    RESTRICTIONS =
*          NONE
*    REGISTER CONVENTIONS =
*          STANDARD EXEC
*    PATCH LABEL = Via DFHPATCH macro
*    MODULE TYPE = EXECUTABLE
*    PROCESSOR = ASSEMBLER
*    ATTRIBUTES = READ ONLY, SERIALLY REUSABLE
*
* ENTRY POINT = DFHUCNV
*
*     PURPOSE =
*         THIS IS THE ONLY ENTRY POINT FOR ALL FUNCTIONS
*
*     LINKAGE =
*         EXEC CICS LINK FROM DFHCCNV IS THE ONLY WAY THIS PROGRAM
*         IS INVOKED
*
*     INPUT =
*         THE PARAMETERS ARE PASSED USING A COMMAREA AND THE
*         DSECT DFHUNVDS DESCRIBES THE STRUCTURE OF THESE PARAMETERS
*         THIS DSECT IS INCLUDED IN THIS PROGRAM BY ISSUING THE
*         DFHCNV TYPE=DSECT MACRO CALL.
*
*     OUTPUT =
*         NO SPECIFIC PARAMETERS  ARE RETURNED, AS THE PURPOSE OF
*         THIS PROGRAM IS PERFORM CONVERSION ON USER DATA.
*
*     EXIT-NORMAL =
*         NORMAL RETURN IS VIA AN EXEC CICS RETURN
*
*     EXIT-ERROR =
**        SAME AS EXIT NORMAL
*
*---------------------------------------------------------------- *
*
* EXTERNAL REFERENCES =
*         NONE
*
*     ROUTINES =
*         NONE
*
*     DATA AREAS =
*         NONE
*
```

図 114. DFHUCNV: z/OS から Windows リンクへのユーザー置き換え可能変換プログラム (パート 2)

```
*     CONTROL BLOCKS =
*         THE 2 MAIN CONTROL BLOCKS REFERENCED ARE
*        DFHUNVDS
*         DESCRIBES THE PARAMETER LIST PASSED IN THE COMMAREA FROM
*         DFHCCNV. IT IS BASICALLY CONSISTS OF A LIST OF POINTERS
*         TO INFORMATION OF INTEREST TO THIS USER PROGRAM. THE FIRST
*         PART OF THE LIST IS FIXED, AND THE SECOND PART CONSISTS
*         OF OVERLAYS DEPENDING ON THE RESOURCE TYPE IN QUESTION.
*        DFHCNVDS
*         DESCRIBES THE STRUCTURE OF INDIVIDUAL FIELDS IN THE
*         PASSED TEMPLATE.
*
*     GLOBAL VARIABLES =
*         NONE
*
* TABLES =
*         DATA FROM THE DFHCNV TABLE IS USED BUT THE NECESSARY
*         ADDRESSES ARE OBTAINED BY DFHCCNV AND PASSED IN THE
*         COMMAREA
*
* MACROS =
*         NONE
*
*---------------------------------------------------------------- *
*
* DESCRIPTION
*    WHAT THIS SAMPLE DOES
*
*        DFHUCNV EXECUTES AS AN EXEC CICS PROGRAM.
*        DFHUCNV IS CALLED FOR ALL EXEC CICS REQUESTS/RESPONSES THAT
*        HAVE RESULTED FROM A CICS TS for Windows FUNCTION SHIP REQUEST
*        AND MAY REQUIRE CONVERSION OF USER DATA FROM ASCII TO EBCDIC
*        OR VICE VERSA. THE FIRST THING THAT THE SAMPLE DOES IS TO
*        OBTAIN ADDRESSABILITY TO THE PASSED COMMAREA, AND THEN
*        CHECK THAT THE REQUEST IS A TEMPORARY STORAGE (TS) REQUEST.
*        IF NOT WE JUST RETURN.
*        NEXT WE CHECK IF DFHCCNV MANAGED TO LOCATE A CONVERSION
*        TEMPLATE FOR THE RESOURCE (TS QUEUE) WITH THIS NAME.
*        IF ONE WAS NOT FOUND (UNVDTMP IS ZERO) THIS MEANS THAT
*        NO CONVERSION INFORMATION WAS PROVIDED (USING DFHCNV MACROS)
*        FOR THIS RESOURCE. IN THIS CASE WE WILL NEVER BE ABLE TO
*        LOCATE ANY USERDATA FIELDS, SO WE JUST RETURN.
*        ASSUMING WE DID HAVE A TEMPLATE, WE NOW SCAN DOWN THE
*        TEMPLATE USING THE SUPPLIED TEMPLATE PTR AND LENGTH. THE
*        MAPPING OF THIS IS PROVIDED BY DFHCNVDS WHICH GIVES
*        THE STRUCTURE OF THE CONSTITUENT FIELDS.
*        EACH FIELD IS EXAMINED, AND WHEN ONE OF TYPE USERDATA
*        IS FOUND WE DO SOME FURTHER CHECKS AS FOLLOWS.
*
```

図 115. DFHUCNV: z/OS から Windows リンクへのユーザー置き換え可能変換プログラム (パート 3)

```
*        IT IS POSSIBLE THAT THE CONVERSION TEMPLATE HAS
*        DEFINITIONS FOR OFFSETS (AND OFFSETS PLUS LENGTHS) THAT ARE
*        GREATER THAN THE ACTUAL DATA ON THE EXEC REQUEST/RESPONSE.
*        OBVIOUSLY IT IS VERY IMPORTANT TO DETERMINE THE LESSER OF
*        THE ACTUAL DATA AND THE PARTICULAR TEMPLATE FIELD DEFINITION
*        TO ENSURE WE DO NOT PERFORM CONVERSION OFF THE END OF THE
*        REAL DATA. ONCE THESE CHECKS ARE DONE THE USERDATA FIELD
*        CAN BE TRANSLATED AS APPROPRIATE. PURELY AS AN EXAMPLE,
*        THE SAMPLE PROGRAM CONVERTS THE USERDATA FIELDS AS CHARACTER,
*        BUT IN A REAL PROGRAM, YOU WOULD PERFORM YOUR OWN SPECIAL
*        TESTING AND CONVERSION AT THIS POINT.
*        THIS LAST STEP IS REPEATED FOR EACH FIELD IN THE TEMPLATE
*        OF TYPE USERDATA, UNTIL THE END OF THE TEMPLATE IS FOUND,
*        AT WHICH TIME A RETURN IS MADE TO THE CALLER (DFHCCNV).
*
*        WHEN WRITING A VERSION OF THIS PROGRAM TO EXECUTE IN A
*        CICS MVS/XA ENVIRONMENT, YOU MUST BE PREPARED TO HANDLE
*        ALL ADDRESSES AS POSSIBLY 31 BIT, AS DFHCCNV AND THE
*        DFHCNV TABLE (CONTAINING THE TEMPLATES) WILL BE LOADED
*        ABOVE THE 16M LINE.
*
*    CAVEAT
*
*        FULLWORD VALUES ARE NOW PASSED IN THE FOLLOWING
*        FIELDS:
*
*          CNVDATAL
*          CNVDATAO
*
*          UNVFCLNP
*          UNVFCKLP
*          UNVICLNP
*          UNVPCLNP
*          UNVTDLNP
*          UNVTSLNP
*
*********************************************************************
```

図 116. DFHUCNV: z/OS から Windows リンクへのユーザー置き換え可能変換プログラム (パート 4)

```
         DFHCNV TYPE=DSECT
DFHUCNV  CSECT
         B     IDBYP
         DFHVM UCNV,ENTRY=DFHUCNV,RMODE=ANY
IDBYP    DS    0H
         DFHREGS ,
         OC    EIBCALEN,EIBCALEN   ANY COMMAREA ?
         BZ    RETURN              NO, JUST RETURN
         L     R2,DFHEICAP
         USING DFHUNVDS,R2         ADDRESSABILITY TO COMMAREA
         L     R10,UNVRSTP         ADDRESS THE RESOURCE TYPE
         CLI   0(R10),CNVTS        IS IT A TEMPORARY STORAGE TYPE
         BNE   RETURN              NO, JUST RETURN
         ICM   R10,B'1111',UNVDTMP IS THERE A CONVERSION TEMPLATE ?
         BZ    RETURN              NO, JUST RETURN
         USING DFHCNVDS,R10        ADDRESSABILITY TO CONVERSION RECS
         L     R4,UNVDLNP
         L     R5,0(0,R4)          GET TOTAL TEMPLATE LENGTH
         AR    R5,R10              END OF TEMPLATE
PROCESS  DS    0H
         CR    R10,R5              HAVE WE REACHED THE END OF TEMPL
         BNL   RETURN              YES
         CLI   CNVRTYPE,CNVTFLD    DOUBLE CHECK ITS A FIELD TYPE REC
         BNE   RETURN              NO, BETTER RETURN
         CLI   CNVDATTY,DTUSRMIN   IN THE USER RANGE ?
         BL    NEXTREC             NO, TOO LOW
         CLI   CNVDATTY,DTUSRMAX   IN THE USER RANGE ?
         BH    NEXTREC             NO, TOO HIGH
         L     R4,UNVDIRP          CHECK THE TYPE OF CONVERSION
         CLI   0(R4),CNVRQATE      ASCII TO EBCDIC REQUEST
         BNE   TRYEBC              NO...
         L     R6,UNVATEP          YES, ADDRESS THE RELEVANT TABLE
         B     CONT1
TRYEBC   DS    0H                  MUST BE EBCDIC TO ASCII
         L     R6,UNVETAP          ADDRESS THE RELEVANT TABLE
CONT1    DS    0H
*
*        GET LOWER VALUE OF ACTUAL LENGTH AND POTENTIAL LENGTH
*        INTO R4
         L     R4,UNVTSLNP
         L     R4,0(0,R4)          PICK UP ACTUAL TS DATA LENGTH
         LTR   R4,R4               JUST CHECK ITS POSITIVE
         BNP   RETURN              IF NOT RETURN
         L     R7,CNVDATAO         GET THE OFFSET FROM THE TEMPLATE
         CR    R7,R4               IS THE OFFSET PAST THE DATA
         BNL   NEXTREC             YES, TRY THE NEXT RECORD
         A     R7,CNVDATAL         ADD IN THE LENGTH(TEMPLATE)
         CR    R7,R4               COMPARE OFFSET+LEN WITH REAL DATA
         BH    LENOK
         LR    R4,R7
LENOK    DS    0H
```

図 117. DFHUCNV: z/OS から Windows リンクへのユーザー置き換え可能変換プログラム (パート 5)

```
*        HERE R4 SHOULD BE THE SMALLER OF THE 2 LENGTHS
*        NOW CALCULATE THE  REAL LENGTH FOR CONVERSION
         S     R4,CNVDATAO         SUBTRACT THE OFFSET
         L     R7,UNVTSDP          ADDRESS ACTUAL DATA
         A     R7,CNVDATAO         ....PLUS OFFSET
*
*  R7 POINTS AT THE START OF WHERE WE TRANSLATE AND R4
*  INDICATES THE LENGTH (ENSURING WE DONT GO FURTHER THAN THE
*  ACTUAL DATA)
*
TRANSMOR DS    0H
         CH    R4,=H'256'          AT LEAST 256 BYTES TO DO
         BL    TRREST              NO
         TR    0(256,R7),0(R6)     TRANSLATE 256 BYTES
         SH    R4,=H'256'          DECREMENT THE COUNT
         AH    R7,=H'256'          INCREMENT THE POINTER
         B     TRANSMOR            DO SOME MORE
TRREST   DS    0H
         LTR   R4,R4               ANY LEFT TO DO ?
         BNP   DONETR              NO
         SH    R4,=H'1'            DECREMENT THE COUNTER FOR THE TR
         EX    R4,TRNSLT
         B     DONETR
TRNSLT   TR    0(0,R7),0(R6)
DONETR   DS    0H                  ALL DATA TRANSLATED
NEXTREC  DS    0H
         SR    R4,R4
         IC    R4,CNVRLEN          GET LENGTH OF THIS RECORD
         AR    R10,R4              AND ADDRESS THE NEXT ONE
         B     PROCESS
RETURN   DS    0H
         EXEC  CICS RETURN
        END   DFHUCNV
```

図 118. DFHUCNV: z/OS から Windows リンクへのユーザー置き換え可能変換プログラム (パート 6)

第 8 部 付録

© Copyright IBM Corp. 1977, 2009

付録 A. 相互通信規則と制約事項のチェックリスト

この付録では、システム間通信および複数領域操作に適用される規則と制約事項のチェックリストを示します。

これらの規則と制約事項のほとんどは、本文にも示してあります。これらの規則は、次のものに適用されます。

- 『トランザクション・ルーティング』
- 431 ページの『DPL 要求の動的ルーティング』
- 432 ページの『自動トランザクション開始』
- 432 ページの『基本マッピング・サポート』
- 432 ページの『LUTYPE6.1 セッションの獲得』
- 432 ページの『同期点処理』
- 432 ページの『ローカル名とリモート名』
- 433 ページの『マスター端末トランザクション』
- 433 ページの『インストールと操作』
- 433 ページの『リソース定義』
- 433 ページの『カスタマイズ』
- 434 ページの『MRO 異常終了コード』

トランザクション・ルーティング

トランザクション・ルーティングに適用される規則と制約事項については、このチェックリストを確認してください。

- 端末とトランザクションの間のトランザクション・ルーティング・パスは、そのパス自体に戻るものであってはなりません。例えば、システム A が、トランザクションがシステム B 上にあるものと指定し、システム B がそれをシステム C にあるものと指定し、さらにシステム C がそれをシステム A にあるものと指定した場合、そのトランザクションの使用をシステム A から試みると、システム C がシステム A に戻る経路を指定しようとしたときに異常終了が起こります。

  この制約事項は、ルーティング・トランザクション (CRTE) が、それ自体に戻るパスの全部または一部を確立するために使用される場合にも適用されます。
- 次の「端末」を使用するトランザクション・ルーティングはサポートされていません。
  - LUTYPE6.1 セッション
  - MRO セッション
  - IBM 7770 および 2260 端末
  - プールを使用するパイプライン論理装置
  - MVS システム・コンソールコンソールから入力されたメッセージは、MODIFY コマンドを使用して、任意の CICS システムに送ることができます。
- トランザクション CEOT は、トランザクション・ルーティング機能ではサポートされていません。

© Copyright IBM Corp. 1977, 2009

- 実行診断機能 (EDF) は、リモート・トランザクションをテストするために単一端末モードで使用することができます。

  2 端末モードで実行される EDF は、両方の端末とユーザー・トランザクションが同じシステム上にある場合、つまりトランザクション・ルーティングが関与しない場合にのみサポートされます。
- TCTTE のユーザー域は、タスク生成時とタスク消去時に更新されます。したがって、端末専有領域で実行され、端末がリモート・トランザクションを実行している間にユーザー域を検査するユーザー出口プログラムは、アプリケーション所有領域で同時に実行されているユーザー出口プログラムと必ずしも同じ値を確認するわけではありません。また、ユーザー域が、両方のシステムで同じ長さになるように定義する必要があることにも注意してください。
- トランザクションによって使用されるすべてのプログラム、テーブル、およびマップは、そのトランザクションを所有するシステムになければなりません。それらのプログラム、テーブル、およびマップは、複数のシステムに必要に応じて複写することができます。
- APPC デバイスへの、または APPC デバイスからの、トランザクション・ルーティングでは、CICS は、同期レベル特性が CM_SYNC_POINT である CPI コミュニケーション会話はサポートしていません。
- 基本機能が APPC 並列セッションの場合には、TCTUA はシップされません。
- 端末関連の EXEC CICS START コマンドで開始されたトランザクションを拡張ルーティングするには、次の条件をすべて 満たす必要があります。
  - START コマンドが、適格な START コマンドのサブセットのメンバーである。すなわち、次の条件をすべて満たしている。
    - START コマンドの TERMID オプションで、コマンドを発行するタスクの基本機能を指定している。つまり、開始するトランザクションは端末に関連しており、開始タスクの基本機能に関連付けられていなければならない。
    - START コマンドを発行するタスクの基本機能が、代理クライアントの仮想端末ではない。
    - START コマンドの SYSID オプションで、リモート領域の名前が指定されていない。つまり、トランザクションが開始されるリモート領域が明示的に指定されていない。
  - 要求側領域と TOR が異なる場合、これらが次のいずれかのリンクで接続されている。
    - MRO リンク
    - APPC 並列セッション・リンク
  - TOR とターゲット領域が次のいずれかのリンクで接続されている。
    - MRO リンク
    - APPC 単一または並列セッション・リンク。 APPC リンクを使用している場合は、少なくとも次のうちの 1 つを満たしていなければならない。
      1. 端末開始トランザクションのルーティングが、リンクを介して既に行われている。
      2. CICSPlex SM がルーティングで使用されている。

– 要求側領域のトランザクション定義では、ROUTABLE(YES) が指定されている。
– トランザクションを動的にルーティングする場合は、TOR のトランザクション定義で DYNAMIC(YES) が指定されている。

拡張ルーティングについて詳しくは、 87 ページの『START コマンドで呼び出されたトランザクションのルーティング』を参照してください。

- 非端末関連 START 要求が*拡張* ルーティング可能であるためには、次の条件をすべて 満たしていなければなりません。
  – 要求側領域とターゲット領域が次のいずれかのリンクで接続されている。
    - MRO リンク
    - APPC 単一または並列セッション・リンク。APPC リンクを使用しており、分散ルーティング・プログラムをターゲット領域で呼び出す場合は、少なくとも次のうちの 1 つを満たしていなければならない。
      1. 端末開始トランザクションのルーティングが、リンクを介して既に行われている。
      2. CICSPlex SM がルーティングで使用されている。
  – 要求側領域のトランザクション定義では、ROUTABLE(YES) が指定されている。
  – 要求を動的にルーティングする場合、次の条件を満たしている。
    - 要求側領域のトランザクション定義で、DYNAMIC(YES) が指定されている。
    - START コマンドの SYSID オプションで、リモート領域の名前が指定されていない。つまり、トランザクションが開始されるリモート領域が明示的に指定されていない。

  拡張ルーティングについて詳しくは、 87 ページの『START コマンドで呼び出されたトランザクションのルーティング』を参照してください。
- 次のタイプの動的トランザクション・ルーティング要求をデイジー・チェーンすることはできません。
  – 非端末関連の START 要求
  – CICS ビジネス・トランザクション・サービスのプロセスおよび活動

DPL 要求の動的ルーティング

分散プログラム・リンク要求が動的ルーティングに対して適格である場合、リモート・プログラムをローカル・システムに対して DYNAMIC と定義するか、リモート・プログラムをローカル・システムに対して定義しないようにします。

動的にルーティングされる DPL 要求のデイジー・チェーンはサポートされていません (112 ページの『DPL 要求の「デイジー・チェーン」』を参照してください)。

自動トランザクション開始

- 一時データ・トリガー・レベル機能によって開始される端末関連トランザクションは、それを開始する一時データ・キューと同じシステム上になければなりません。この制約事項は、マクロ・レベルおよびコマンド・レベルの両方のアプリケーション・プログラムに適用されます。
- EXEC CICS START コマンドで開始されたトランザクションの動的ルーティングには制約があります (429 ページの『トランザクション・ルーティング』の条件リストを参照)。

基本マッピング・サポート

- BMS サポートは、ページング・コマンドを入力するための端末を所有する各システムに必要です。
- BMS ROUTE 要求は、メッセージが送達される端末が経路リストに指定されていない場合、指定のリモート・オペレーターまたはオペレーター・クラスにメッセージを送るために使用することはできません。

LUTYPE6.1 セッションの獲得

- アプリケーションが LUTYPE6.1 接続を獲得しようとした際にリモート・システムが使用できない場合、接続はサービス休止状態に置かれます。
- リモート・システムが AUTOCONNECT を使用する CICS システムである場合は、リモート・システムの初期設定が終了したときに、接続が再びサービス可能になります。
- リモート・システムが AUTOCONNECT(YES|ALL) を指定しないか、自動接続機能のない非 CICS システムである場合は、CEMT SET CONNECTION コマンドを使用するか、アプリケーション・プログラムから EXEC CICS SET CONNECTION コマンドを発行して、その接続を再びサービス可能にする必要があります。

同期点処理

SYNCPOINT ROLLBACK コマンドは、APPC セッション、IPIC セッション、および MRO セッションでサポートされています。

ローカル名とリモート名

ローカル名は、規則に従ってリモート名に変換されます。

- トランザクション ID は、トランザクション実行要求が、ある CICS システムから別の CICS システムに伝送されるときに、ローカル名からリモート名に変換されます。

  しかし、EXEC CICS RETURN コマンドに指定されたトランザクション ID は、アプリケーション所有領域から端末専有領域に伝送されるときに変換されません。

- 端末 ID は、指定端末でトランザクションを実行するためのトランザクション・ルーティング要求が、ある CICS システムから別の CICS システムにシップされるときに、ローカル名からリモート名に変換されます。

  しかし、端末 ID を指定する EXEC CICS START コマンドが、ある CICS システムから別の CICS システムに機能シップされる場合は、端末 ID がローカル名からリモート名に変換されません。

マスター端末トランザクション

マスター端末トランザクション CEMT が照会して修正できるのは、ローカル所有の端末だけです。このトランザクションの対象となるのは、マスター端末トランザクションが実行されているシステムが所有する端末だけです。

インストールと操作

- モジュール DFHIRP は LPA 常駐にする必要があります。そうしないと、ジョブやコンソール・コマンドは、完了時に異常終了することがあります。
- 領域間通信には、サブシステム・インターフェース (SSI) のサポートが必要です。
- LU-LU ペアの間に複数の APPC 接続をインストールしないでください。
- LU-LU ペアの間に APPC 接続と LUTYPE6.1 接続を同時にインストールしないでください。
- 同じ 2 つの CICS 領域間に複数の MRO 接続をインストールしないでください。
- 同じ CICS 領域に複数の総称 EXCI 接続をインストールしないでください。

リソース定義

- VTAM 端末に関する PRINTER および ALTPRINTER オプションは、定義される端末を所有するシステムと同じシステムによって所有されるプリンター (指定されていれば) を指名しなければなりません。
- 端末リスト・テーブル (DFHTLT) にリストされた端末は、端末リスト・テーブルと同じシステムになければなりません。

カスタマイズ

- ノード・エラー・プログラム、ユーザー出口、およびユーザー・プログラムの間の通信は、ユーザーが行う必要があります。
- システム障害後に保護タスクに関する入力メッセージをリカバリーするトランザクションは、その保護タスクを呼び出した端末と同じシステムで実行する必要があります。

MRO 異常終了コード

- 送信状態の IRC トランザクションは、そのパートナーが異常終了しなければならない場合には、エラー理由コードを受け取ることができません。このトランザクション自体が、コード AZI2 を出して異常終了します。このコードは、もう一方の側が存在しないことを示す一般的な標識として解釈されます。障害の実際の理由は、最初にエラーを検出した CICS 領域の CSMT 宛先から読み取ることができます。例えば、バックエンド・トランザクションの接続におけるセキュリティー違反は、開始コマンドが SEND ではなく、CONVERSE の場合にのみ、フロントエンド・トランザクションによって報告されます。

付録 B. CICS の APPC アーキテクチャーへのマッピング

この付録では、APPC プログラミング言語が CICS によってどのように実装されているかについて説明しています。

APPC プログラミング言語については、SNA 資料の「*Transaction Programmer's Reference Manual for LU Type 6.2*」で説明しています。この付録には、以下のトピックが含まれています。

- 『サポートされるオプション・セット』.

  これは、CICS によってどの APPC オプション・セットがサポートされ、どれがサポートされないかを示す表です。
- 437 ページの『制御オペレーター verb の CICS による実装』.

  ここでは、CICS が APPC 制御オペレーター verb をどのように実現しているのかを説明します。さらに、これらの verb が CICS コマンドにどのように対応するのかを示す表を示します。
- 445 ページの『APPC アーキテクチャーからの逸脱』.

  このセクションでは、CICS による APPC の実装が「*Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*」で述べられているアーキテクチャーと異なる点を説明します。

基本会話と非マップ式会話の CICS アプリケーション・プログラミング・インターフェースが APPC verb にどのように対応するのかについては、「*CICS Distributed Transaction Programming Guide*」を参照してください。

サポートされるオプション・セット

表 66. CICS による APPC オプション・セットのサポート

セット #	セット名	サポートの有無
101	LU の送信バッファーをクリアする	使用可
102	属性を入手する	使用可
103	受信時に通知して通知のテストをする	使用不可
104	受信時に通知して待つ	使用不可
105	受信の準備をする	使用可
106	ただちに受信する **注:** CICS プログラムは receive_immediate 要求をサポートします。ただし、これらの要求が通信用共通プログラミング・インターフェースを使用してコーディングされていることが条件です。	使用可
108	同期点サービス	使用可
109	TP 名とインスタンス ID を入手する	使用不可
110	会話タイプを入手する	使用可
111	同期点の際に検出されたプログラム・エラーのリカバリー	使用可

© Copyright IBM Corp. 1977, 2009

表 66. CICS による APPC オプション・セットのサポート (続き)

セット #	セット名	サポートの有無
201	競合勝者セッションの割り振りのキューイング	使用不可
203	セッションの即時割り振り	使用可
204	同じ LU にあるプログラム間の会話	使用不可
211	セッション・レベル LU-LU の検査	使用可
212	ユーザー ID の検査	使用可
213	プログラム指定のユーザー ID とパスワード	使用不可
214	ユーザー ID の許可	使用可
215	プロファイルの検査と許可	使用可
217	プロファイルのパススルー	使用不可
218	プログラム指定のプロファイル	使用不可
241	PIP データを送信する	使用可
242	PIP データを受信する	使用可
243	アカウンティング	使用可
244	長いロック	使用不可
245	送信要求の受信をテストする	使用可
246	データのマッピング	使用不可
247	FMH データ	使用不可
249	同期点操作に対し読み取り専用の応答を送る	使用不可
251	トランザクションと会話の識別情報を抜き出す	使用不可
290	データのシステム・ログへのロギング	使用不可
291	マップ式会話 LU サービス・コンポーネント	使用可
401	信頼可能一方向ブラケット	使用不可
501	CHANGE_SESSION_LIMIT verb	使用可
502	ACTIVATE_SESSION verb	使用可
504	DEACTIVATE_SESSION verb	使用不可
505	LU- 定義 verb	使用可
601	MIN_CONWINNERS_TARGET パラメーター	使用不可
602	RESPONSIBLE(TARGET) パラメーター	使用不可
603	DRAIN_TARGET(NO) パラメーター	使用不可
604	FORCE パラメーター	使用不可
605	LU-LU セッション限界	使用不可
606	ローカルで既知の LU 名	使用可
607	解釈されない LU 名	使用不可
608	単一セッションの再開	使用不可
610	最大 RU サイズ境界	使用可
611	セッション・レベルの必須暗号	使用不可
612	競合勝者自動活動化限界	使用不可
613	ローカル最大 (LU、モード) セッション限度	使用可
616	CPSVCMG モード名サポート	使用不可
617	セッション・レベルの選択暗号	使用不可

制御オペレーター verb の CICS による実装

CICS は、制御オペレーターの verb をさまざまな方法でサポートします。

一部の verb は、CICS マスター端末トランザクション CEMT によってサポートされます。これに該当する CEMT コマンドには、次のものがあります。

- CEMT INQUIRE CONNECTION
- CEMT SET CONNECTION
- CEMT INQUIRE MODENAME
- CEMT SET MODENAME

CEMT は通常、オペレーターによってディスプレイ装置から入力されます。これについては、「*CICS Supplied Transactions*」の CEMT マスター端末に説明があります。

接続とモード名に対する照会と設定の操作は、次のコマンドを使用することによって、CICS API でも可能です。

- EXEC CICS INQUIRE CONNECTION
- EXEC CICS SET CONNECTION
- EXEC CICS INQUIRE MODENAME
- EXEC CICS SET MODENAME

これらのコマンドのプログラミング情報は、「*CICS System Programming Reference*」の INQUIRE CONNECTIONに記載されています。

一部の制御オペレーターの verb は、CICS リソース定義によってサポートされます。APPC リンクの定義については、186 ページの『APPC 接続の定義』を参照してください。リソース定義構文の詳細については、「*CICS Resource Definition Guide*」を参照してください。

オンラインのリソース定義を使用すれば、CICS が実行されている間に、CEDA トランザクションを使って、CONNECTION と SESSION の一部のオプションを変更することができます。マクロ・レベルの定義では、それらのオプションは、CICS が実行される間固定されます。

制御オペレーターの verb

APPC 制御オペレーター verb が CICS によってどのように実現されているかを下記の表で示します。

それらに対応する戻りコードの詳細については、444 ページの『制御オペレーター verb の戻りコード』を参照してください。

注: CEMT が表示されている場合には、同じ形式の EXEC CICS コマンドが使用できます。

表 67. CHANGE_SESSION_LIMIT

CHANGE_SESSION_LIMIT	**CEMT SET MODENAME**
LU_NAME(vble)	CONNECTION( )
MODE_NAME(vble)	MODENAME( )
LU_MODE_SESSION_LIMIT(vble)	AVAILABLE( )
MIN_CONWINNERS_SOURCE(vble)	CICS は、AVAILABLE 要求およびグループの DEFINE SESSIONS の MAXIMUM 値に基づいて改定値を折衝する
MIN_CONWINNERS_TARGET(vnle)	サポートなし
RESPONSIBLE(source)	サポートあり
RESPONSIBLE(target)	サポートなし。CICS は、RESP(TARGET) の受信をサポートしない。
RETURN_CODE	サポートあり

表 68. INITIALIZE_SESSION_LIMIT

INITIALIZE_SESSION_LIMIT	**DEFINE SESSIONS (CICS リソース定義)**
LU_NAME(vble)	CONNECTION( )
MODE_NAME(vble)	MODENAME( )
LU_MODE_SESSION_LIMIT(vble)	MAXIMUM(value1,)
MIN_CONWINNERS_SOURCE(vble)	MAXIMUM(,value2)
MIN_CONWINNERS_TARGET(vnle)	サポートなし
RETURN_CODE	サポートあり

表 69. PROCESS_SESSION_LIMIT

PROCESS_SESSION_LIMIT	**CNOS がターゲット CICS システムによって受信された場合の CICS 提供トランザクション CLS1 による自動アクション**
RESOURCE(vble)	接続 RDO
LU_NAME(vble)	内部的に渡される
MODE_NAME(vble1,vble2)	内部的に渡される
RETURN_CODE	サポートあり

表 70. RESET_SESSION_LIMIT

RESET_SESSION_LIMIT	**CEMT SET MODENAME (個々のモード・グループに対する) または CEMT SET CONNECTION RELEASED (すべてのモード・グループをリセット)**
LU_NAME(vble)	CONNECTION( )
MODE_NAME(ALL)	SET CONNECTION( ) RELEASED
MODE_NAME(ONE(vble))	MODENAME( ) AVAILABLE(0)
MODE_NAME(ONE('SNASVCMG'))	SET CONNECTION( ) RELEASED
RESPONSIBLE(SOURCE)	サポートあり
RESPONSIBLE(TARGET)	サポートなし
DRAIN_SOURCE(NO|YES)	CICS サポート YES

表 70. RESET_SESSION_LIMIT (続き)

RESET_SESSION_LIMIT	**CEMT SET MODENAME (個々のモード・グループに対する) または CEMT SET CONNECTION RELEASED (すべてのモード・グループをリセット)**
DRAIN_TARGET(NO|YES)	CICS サポート YES
FORCE(NO|YES)	サポートなし
RETURN_CODE	サポートあり

表 71. ACTIVATE_SESSION

ACTIVATE_SESSION	**CEMT SET MODENAME ACQUIRED (個々のモード・グループに対する) または CEMT SET CONNECTION ACQUIRED (SNASVCMG セッションに対する)**
LU_NAME(vble)	CONNECTION( )
MODE_NAME(vble)	MODENAME( ) ACQUIRED
MODE_NAME('SNASVCMG')	CEMT SET CONNECTION ACQUIRED 発行時に活動化される
RETURN_CODE	サポートあり

表 72. DEACTIVATE_CONVERSATION_GROUP

DEACTIVATE_CONVERSATION_GROUP	**サポートなし**

表 73. DEACTIVATE_SESSION

DEACTIVATE_SESSION	**サポートなし**

表 74. DEFINE_LOCAL_LU

DEFINE_LOCAL_LU	**DEFINE SESSIONS + DFHSIT マクロ (CICS リソース定義)**
FULLY_QUALIFIED_LU_NAME(vble)	指定不可。CICS は、ネットワーク LU 名 (DFHSIT の APPLID) を使用する
LU_SESSION_LIMIT(NONE)	サポートなし
LU_SESSION_LIMIT(VALUE(vble))	すべてのセッションの MAX(nn) の合計
SECURITY(ADD USER_ID(vble))	外部セキュリティー・マネージャー (ESM) において
SECURITY(ADD PASSWORD(vble))	サポートなし。ESM に定義
SECURITY(ADD PROFILE(vble))	サポートなし。ESM に定義
SECURITY(DELETE USER_ID(vble))	ESM においてサポート
SECURITY(DELETE PASSWORD(vble))	サポートなし。ESM に定義
MAP_NAME(ADD(vble))	サポートなし
MAP_NAME(DELETE(vble))	サポートなし
BIND_RSP_QUEUE_CAPACITY(YES|NO)	サポートなし

表 75. DEFINE_MODE

DEFINE_MODE	EXEC CICS CONNECT PROCESS + MODEENT マクロ (ACF/VTAM システム定義) + DEFINE SESSIONS (CICS リソース定義)
FULLY_QUALIFIED_LU_NAME(vble)	指定不可。SESSIONS の CONNECTION で識別される LU
MODE_NAME(vble)	SESSIONS の MODENAME は MODEENT の LOGMODE にマップされる
SEND_MAX_RU_SIZE_LOWER_BOUND (vble)	8 に固定される
SEND_MAX_RU_SIZE_UPPER_BOUND (vble)	SESSIONS の SENDSIZE
PREFERRED_RECEIVE_RU_SIZE (vble)	サポートなし
PREFERRED_SEND_RU_SIZE(vble)	サポートなし
RECEIVE_MAX_RU_SIZE_LOWER_BOUND (vble)	256 に固定される
RECEIVE_MAX_RU_SIZE_UPPER_BOUND (vble)	SESSIONS の RECEIVESIZE
SINGLE_SESSION_REINITIATION OPERATOR	サポートなし
SINGLE_SESSION_REINITIATION PLU	サポートなし
SINGLE_SESSION_REINITIATION SLU	サポートなし
SINGLE_SESSION_REINITIATION PLU_OR_SLU	サポートなし
SESSION_LEVEL_CRYPTOGRAPHY (NOT_SUPPORTED)	デフォルト
SESSION_LEVEL_CRYPTOGRAPHY (MANDATORY)	サポートなし
SESSION_LEVEL_CRYPTOGRAPHY (SELECTIVE)	サポートなし
CONWINNER_AUTO_ACTIVATE_LIMIT (vble)	SESSIONS の MAXIMUM(,value2)
SESSION_DEACTIVATED_TP_NAME (vble)	サポートなし
LOCAL_MAX_SESSION_LIMIT(vble)	SESSIONS の MAXIMUM(nn,)

表 76. DEFINE_REMOTE_LU

DEFINE_REMOTE_LU	DEFINE CONNECTION (CICS リソース定義)
FULLY_QUALIFIED_LU_NAME(vble)	指定不可。
LOCALLY_KNOWN_LU_NAME(NONE)	サポートなし
LOCALLY_KNOWN_LU_NAME(NAME(vble))	CONNECTION(name)
UNINTERPRETED_LU_NAME(NONE)	デフォルト CONNECTION(name)
UNINTERPRETED_LU_NAME (NAME(vble))	CONNECTION の NETNAME
INITIATE_TYPE(INITIATE_ONLY)	サポートなし
INITIATE_TYPE(INITIATE_OR_QUEUE)	サポートなし

表 76. DEFINE_REMOTE_LU (続き)

DEFINE_REMOTE_LU	DEFINE CONNECTION (CICS リソース定義)
PARALLEL_SESSION_SUPPORT(YES|NO)	CONNECTION の SINGLESESS(NO|YES)
CNOS_SUPPORT(YES|NO)	常に YES
LU_LU_PASSWORD(NONE)	CONNECTION のデフォルト
LU_LU_PASSWORD(VALUE(vble))	CONNECTION の BINDPASSWORD、または RACF APPCLU プロファイルの SESSKEY
SECURITY_ACCEPTANCE(NONE)	ATTACHSEC(LOCAL)
SECURITY_ACCEPTANCE (CONVERSATION)	ATTACHSEC(VERIFY)
SECURITY_ACCEPTANCE (ALREADY_VERIFIED)	ATTACHSEC(IDENTIFY) または ATTACHSEC (PERSISTENT)

表 77. DEFINE_TP

DEFINE_TP	DEFINE TRANSACTION (CICS リソース定義)
TP_NAME(vble)	TRANSACTION(name)
STATUS(ENABLED)	STATUS(ENABLED)
STATUS(TEMP_DISABLED)	サポートなし
STATUS(PERM_DISABLED)	STATUS(DISABLED)
CONVERSATION_TYPE(MAPPED|BASIC)	すべての TP についてサポート (コマンドの選択によって決まる)
SYNC_LEVEL(NONE|CONFIRM∨SYNCPT)	すべての TP の SYNCPT (CONNECT PROCESS に指定された実レベル)
SECURITY_REQUIRED(NONE)	サポートなし。ESM に定義
SECURITY_REQUIRED(CONVERSATION)	サポートなし。ESM に定義
SECURITY_REQUIRED (ACCESS(PROFILE))	サポートなし
SECURITY_REQUIRED (ACCESS(USER_ID))	サポートなし。ESM に定義
SECURITY_REQUIRED (ACCESS(USER_ID_PROFILE))	サポートなし
SECURITY_ACCESS(ADD(USER_ID(vble)))	トランザクションを再定義できる
SECURITY_ACCESS(ADD(PROFILE(vble)))	トランザクションを再定義できる
SECURITY_ACCESS (DELETE(USER_ID(vble)))	トランザクションを再定義できる
SECURITY_ACCESS (DELETE(PROFILE(vble)))	トランザクションを再定義できる
PIP(NO)	すべての TP について指定される
PIP(YES(vble))	CONNECT PROCESS で指定される
PIP(NO_LU_VERIFICATION)	すべての PIP データのデフォルト
DATA_MAPPING(NO|YES)	すべての TP について DATA_MAPPING(NO)
FMH_DATA(NO|YES)	すべての TP について FMH_DATA(YES)
PRIVILEGE(NONE)	サポートなし
PRIVILEGE(CNOS)	サポートなし
PRIVILEGE(SESSION_CONTROL)	サポートなし
PRIVILEGE(DEFINE)	サポートなし
PRIVILEGE(DISPLAY)	サポートなし
PRIVILEGE(ALLOCATE_SERVICE_TP)	サポートなし
INSTANCE_LIMIT(vble)	サポートなし

表 77. DEFINE_TP (続き)

DEFINE_TP	DEFINE TRANSACTION (CICS リソース定義)
RETURN_CODE	サポートあり

表 78. DELETE

DELETE	EXEC CICS DISCARD
LOCAL_LU_NAME(vble)	サポートなし
REMOTE_LU_NAME	サポートなし
MODE_NAME	サポートなし
TP_NAME	DISCARD TRANSACTION( )
RETURN_CODE	サポートあり

表 79. DISPLAY_LOCAL_LU

DISPLAY_LOCAL_LU	CEMT INQUIRE CONNECTION + CEMT INQUIRE MODENAME + CEMT INQUIRE TRANSACTION
FULLY_QUALIFIED_LU_NAME(vble)	CICS では指定不可。DFHSIT の APPLID がローカル LU の ID の役割を持つ。リモート LU の識別により、特定の情報を得られる。それ以外の場合、ユニバーサル ID * を使用できる。
LU_SESSION_LIMIT(vble)	INQ MODENAME の MAXIMUM
LU_SESSION_COUNT(vble)	INQ MODENAME で ACTIVE
SECURITY(vble)	使用不可
MAP_NAMES(vble)	サポートなし
REMOTE_LU_NAMES(vble)	INQ CONNECTION(*)
TP_NAMES(vble)	INQ TRANSACTION(*)
BIND_RSP_QUEUE_CAPABILITY(vble)	サポートなし
RETURN_CODE	サポートあり

表 80. DISPLAY_REMOTE_LU

DISPLAY_REMOTE_LU	CEMT INQUIRE CONNECTION + CEMT INQUIRE MODENAME
FULLY_QUALIFIED_LU_NAME(vble)	指定不可。CONNECTION または MODENAME が使用可能
LOCALLY_KNOWN_LU_NAME(vble)	CONNECTION 名
UNINTERPRETED_LU_NAME(vble)	INQ CONNECTION の NETNAME
INITIATE_TYPE(vble)	サポートなし
PARALLEL_SESSION_SUPPORT(vble)	CEDA VIEW の SINGLESESS(Y|N)
CNOS_SUPPORT(vble)	常に YES
SECURITY_ACCEPTANCE_LOCAL_LU (vble)	使用不可
SECURITY_ACCEPTANCE_REMOTE_LU (vble)	使用不可
MODE_NAMES(vble)	ローカルで既知の LU 名がある CEDA VIEW SESSIONS
RETURN_CODE	サポートあり

表 81. *DISPLAY_MODE*

DISPLAY_MODE	**CEMT INQUIRE MODENAME + CEMT INQUIRE TERMINAL**
FULLY_QUALIFIED_LU_NAME(vble)	指定不可。
MODE_NAME(vble)	MODENAME
LOCAL_MAX_SESSION_LIMIT(vble)	CEMT INQ MODENAME の AVA
CONVERSATION_GROUP_IDS(vble)	サポートなし
SEND_MAX_RU_SIZE_LOWER_BOUND (vble)	8 に固定される
SEND_MAX_RU_SIZE_UPPER_BOUND (vble)	使用不可
RECEIVE_MAX_RU_SIZE_LOWER_BOUND (vble)	256 に固定される
RECEIVE_MAX_RU_SIZE_UPPER_BOUND (vble)	使用不可
PREFERRED_SEND_RU_SIZE(vble)	サポートなし
PREFERRED_RECEIVE_RU_SIZE (vble)	サポートなし
SINGLE_SESSION_REINITIATION(vble)	サポートなし
SESSION_LEVEL_CRYPTOGRAPHY(vble)	使用不可
SESSION_DEACTIVATED_TP_NAME	サポートなし
CONWINNER_AUTO_ACTIVATE_LIMIT (vble)	使用不可
LU_MODE_SESSION_LIMIT(vble)	INQ MODENAME の MAXIMUM
MIN_CONWINNERS(vble)	サポートなし
MIN_CONLOSERS(vble)	サポートなし
TERMINATION_COUNT(vble)	サポートなし
DRAIN_LOCAL_LU(vble)	サポートなし
DRAIN_REMOTE_LU(vble)	サポートなし
LU_MODE_SESSION_COUNT(vble)	INQ MODENAME で ACTIVE
CONWINNERS_SESSION_COUNT(vble)	使用不可
CONLOSERS_SESSION_COUNT(vble)	使用不可
SESSION_IDS(vble)	INQ TERMINAL(*)
RETURN_CODE	サポートあり

表 82. *DISPLAY_TP*

DISPLAY_TP	**CEMT INQUIRE TRANSACTION**
TP_NAME(vble)	TRANSACTION(tranid)
STATUS(vble)	ENABLED/DISABLED
CONVERSATION_TYPE(vble)	CICS TP は両者のタイプを許可
SYNC_LEVEL(vble)	CICS TP はすべての同期レベルを許可
SECURITY_REQUIRED(vble)	使用不可
SECURITY_ACCESS(vble)	使用不可
PIP(vble)	CICS TP は PIP YES と NO を許可
DATA_MAPPING(vble)	常に NO
FMH_DATA(vble)	常に YES
PRIVILEGE(vble)	サポートなし
INSTANCE_LIMIT(vble)	サポートなし

表 82. DISPLAY_TP (続き)

DISPLAY_TP	CEMT INQUIRE TRANSACTION
INSTANCE_COUNT(vble)	CEMT INQ TRAN( )
RETURN_CODE	サポートあり

制御オペレーター verb の戻りコード

CEMT INQUIRE および SET CONNECTION か MODENAME、およびそれと同等の EXEC CICS コマンドを使用すると、CICS は LU サービス・マネージャーを非同期に開始します。

このときのエラーは、一部が CEMT か CICS API によって検出され、ただちに戻されます。他のエラーは、LU サービス管理トランザクション (CLS1) があとから実行されるまで検出されません。

CLS1 によってエラーが検出されると、表 83 に示すメッセージが CSMT ログに書き込まれます。通常の操作では、CICS マスター端末オペレーターは、コマンドを出したあとに、CSMT ログを調べることはしません。したがって、一般にオペレーターは、パラメーターを変えるコマンド (例えば、SET MODENAME( ) ...) を出したら、その要求が実行されるまで数秒間待ち、それから同じコマンドの INQUIRE を出して、要求した変更が行われているかどうかを検査します。エラーが起こるといったごくまれな場合に、マスター端末の制御オペレーターは CSMT ログを参照することができます。

CEMT をメニュー・パネルから実行する場合には、上の一連の操作を行うのは非常に簡単です。

CLS1 の実行結果を知らせるメッセージは DFHZC4900 です。メッセージに付随する説明文はメッセージによって異なります。それらのメッセージの要約を 表 83 に示します。メッセージの詳細については、「*CICS Messages and Codes*」を参照してください。場合によっては、DFHZC4901 がさらに出されることがあります。

表 83. CLS1 によって出されるメッセージ

APPC RETURN CODE	CICS MESSAGE
OK	DFHZC4900 result = SUCCESSFUL
ACTIVATION_FAILURE_RETRY	DFHZC4900 result = VALUES AMENDED + DFHZC4901 MAX = 0
ACTIVATION_FAILURE_NO_RETRY	DFHZC4900 result = VALUES AMENDED + DFHZC4901 MAX = 0
ALLOCATION_ERROR	CEMT によりチェックされる。割り振りが失敗すると、SYSTEM NOT ACQUIRED がオペレーターに戻される。
COMMAND_RACE_REJECT	DFHZC4900 result = RACE DETECTED
LU_MODE_SESSION_LIMIT_CLOSED	DFHZC4900 result = VALUES AMENDED + DFHZC4901 MAX = 0
LU_MODE_SESSION_LIMIT_EXCEEDED	DFHZC4900 result = VALUES AMENDED + DFHZC4901 MAX = (negotiated value)
LU_MODE_SESSION_LIMIT_NOT_ZERO	DFHZC4900 result = VALUES AMENDED + DFHZC4901 MAX = (negotiated value)

表 83. CLS1 によって出されるメッセージ (続き)

APPC RETURN CODE	CICS MESSAGE
LU_MODE_SESSION_LIMIT_ZERO	DFHZC4900 result = VALUES AMENDED + DFHZC4901 MAX = 0
LU_SESSION_LIMIT_EXCEEDED	DFHZC4900 result = VALUES AMENDED + DFHZC4901 MAX = (negotiated value)
PARAMETER_ERROR	CEMT によりチェックされる。
REQUEST_EXCEEDS_MAX_ALLOWED	CEMT によりチェックされる。
RESOURCE_FAILURE_NO_RETRY	LU サービス管理トランザクション (CLS1) は、異常終了コード ATNI を出して異常終了
UNRECOGNIZED_MODE_NAME	DFHZC4900 result = MODENAME NOT RECOGNIZED

APPC アーキテクチャーからの逸脱

このセクションでは、CICS による APPC の実装と、「*Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*」で述べられているアーキテクチャーとの相違点について説明します。

次の点が異なります。

- **CICS によるインプリメンテーション**: CICS は、着信 BIND 要求をみて、CNOS 標識 (BIND RQ バイト 24 ビット 6) と PARALLEL-SESSIONS 標識 (BIND RQ バイト 24 ビット 7) の組み合わせが正しいかどうかを調べます。組み合わせが正しくないと (つまり、PARALLEL-SESSIONS が指定されているが、CNOS が指定されていない)、CICS は BIND 要求に対し否定応答を返します。

  APPC アーキテクチャー: 2 次論理装置 (SLU)、つまり BIND 要求の受信側は、CNOS と PARALLEL-SESSIONS の標識をサポートされるレベルになるように折衝し、それらを BIND 応答に入れて戻す必要があります。SLU は、これらの標識の組み合わせが正しいかどうかは検査しません。

APPC トランザクション・ルーティングの APPC アーキテクチャーからの逸脱

トランザクション・プログラムは、同期解放、同期送信、同期受信の状態で ISSUE SIGNAL を出すことはできません。これを行うと、状態チェックになることがあります。この 1 つの逸脱は、APPC トランザクション・ルーティングだけに適用されます。

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あり、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。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。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。

〒106-8711
東京都港区六本木 3-2-12
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

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

本書には、技術的に正確でない記述や誤植がある場合があります。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 United Kingdom Hursley Park Winchester Eglinton Avenue East, Hampshire York, England M3C 1H7, SO21. 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。

© Copyright IBM Corp. 1977, 2009

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corp. の商標です。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。

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

Microsoft、および Windows 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。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

参考文献

CICS Transaction Server for z/OS の CICS の資料

一般

CICS Transaction Server for z/OS Program Directory (GI13-0536)
CICS Transaction Server for z/OS リリース・ガイド (GC88-5845)
CICS Transaction Server for z/OS CICS TS v2.3 からのアップグレード (GC88-5847)
CICS Transaction Server for z/OS CICS TS v3.1 からのアップグレード (GC88-5848)
CICS Transaction Server for z/OS CICS TS v3.2 からのアップグレード (GC88-5849)
CICS Transaction Server for z/OS インストール・ガイド (GC88-5846)

CICS へのアクセス

CICS インターネット・ガイド (SC88-5853)
CICS Web サービス・ガイド (SC88-5852)

管理

CICS System Definition Guide (SC34-6999)
CICS Customization Guide (SC34-7001)
CICS Resource Definition Guide (SC34-7000)
CICS CICS Operations and Utilities Guide (SC34-7002)
CICS RACF Security Guide (SC34-7003)
CICS Supplied Transactions (SC34-7004)

プログラミング

CICS アプリケーション・プログラミング・ガイド (SC88-5850)
CICS アプリケーション・プログラミング・リファレンス (SC88-5851)
CICS System Programming Reference (SC34-7024)
CICS Front End Programming Interface User's Guide (SC34-7027)
CICS C++ OO Class Libraries (SC34-7026)
CICS Distributed Transaction Programming Guide (SC34-7028)
CICS Business Transaction Services (SC34-7029)
Java Applications in CICS (SC34-7025)

診断

CICS Problem Determination Guide (SC34-7034)
CICS パフォーマンス・ガイド (SC88-5854)
CICS Messages and Codes (SC34-7035)
CICS Diagnosis Reference (GC34-7038)
CICS Recovery and Restart Guide (SC34-7012)
CICS Data Areas (GC34-7014)
CICS Trace Entries (SC34-7013)
CICS Supplementary Data Areas (GC34-7015)

© Copyright IBM Corp. 1977, 2009

CICS Debugging Tools Interfaces Reference (GC34-7039)

通信

CICS 相互通信ガイド (SC88-5857)
External Interfaces Guide (SC34-7019)

データベース

CICS DB2 Guide (SC34-7011)

CICS IMS Database Control Guide (SC34-7016)

CICS Shared Data Tables Guide (SC34-7017)

CICS Transaction Server for z/OS の SCICSPlex SM の資料

一般

CICSPlex SM 概念および計画 (SC88-5855)
CICSPlex SM WUI ガイド (SC88-5856)

管理

CICSPlex SM Administration (SC34-7005)
CICSPlex SM Operations Views Reference (SC34-7006)
CICSPlex SM Monitor Views Reference (SC34-7007)
CICSPlex SM Managing Workloads (SC34-7008)
CICSPlex SM Managing Resource Usage (SC34-7009)
CICSPlex SM Managing Business Applications (SC34-7010)

プログラミング

CICSPlex SM Application Programming Guide (SC34-7030)
CICSPlex SM Application Programming Reference (SC34-7031)

診断

CICSPlex SM Resource Tables Reference (SC34-7032)
CICSPlex SM Messages and Codes (GC34-7035)
CICSPlex SM Problem Determination (GC34-7037)

その他の CICS の資料

以下の資料は、CICS に関する詳細な情報を含む (CICS Transaction Server for z/OS バージョン 4 リリース 1 に同梱ではない) ものです。

Designing and Programming CICS Applications (SR23-9692)

CICS Application Migration Aid Guide (SC33-0768)

CICS Family: API Structure (SC33-1007)

CICS ファミリー クライアント・サーバー プログラミングの手引き (SC88-7429)

CICS Family: Interproduct Communication (SC34-6853)

CICS Family: Communicating from CICS on System/390 (SC34-6854)

CICS Transaction Gateway for z/OS Administration (SC34-5528)

CICS Family: General Information (GC33-0155)

CICS 4.1 Sample Applications Guide (SC33-1173)

CICS/ESA 3.3 XRF Guide (SC33-0661)

その他の IBM の資料

次の資料には、関連 IBM 製品についての情報が含まれています。

IMS

IMS Communications and Connections Guide (SC18-9703)

IMS Installation Guide (GC18-9710)

IMS Operations and Automation Guide (SC18-9716)

MVS

z/OS MVS シスプレックスのセットアップ (SA88-8591)

ネットワーク・プログラム・プロダクト

ネットワーク・プログラム・プロダクト概説書 (N:GC30-3350)

システム・アプリケーション体系 (SAA)

SAA CPI-C 解説書 (SC88-7217)

システム・ネットワーク体系 (SNA)

概念と諸製品 (N:GC30-3072)

Format and Protocol Reference Manual: Architecture Logic (SC30-3112)

Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2 (SC30-3269)

Format and Protocol Reference Manual: Distribution Services (SC30-3098)

Reference: Peer Protocols (SC31-6808-1)

Sessions Between Logical Units (GC20-1868)

SNA Formats (GA27-3136)

Technical Overview (GC30-3073)

Transaction Programmer's Reference Manual for LU Type 6.2 (GC30-3084)

VTAM

VTAM Customization (LY43-0075)

VTAM Data Areas for MVS Volume 1 (LY43-0076)

VTAM Data Areas MVS Volume 2 (LY43-0077)

VTAM Diagnosis (LY43-0078)

VTAM Migration Guide (GC31-6416)

VTAM Messages and Codes (GC31-6418)

VTAM Network Implementation Guide (GC31-6419)

VTAM Operation (GC31-6420)

VTAM Programming (SC31-6421)

VTAM Release Guide (GC31-6441)

VTAM Resource Definition Reference (SC31-6428)

アクセシビリティー

アクセシビリティー機能は、運動障害または視覚障害など身体に障害を持つユーザーがソフトウェア・プロダクトを快適に使用できるようにサポートします。

CICS システムのセットアップ、実行、および保守で必要になる作業の大部分は、以下のいずれかの方法で実行することができます。

- CICS にログオンした 3270 エミュレーターを使用する
- TSO にログオンした 3270 エミュレーターを使用する
- MVS システム・コンソールとしての 3270 エミュレーターを使用する

IBM パーソナル・コミュニケーションズの 3270 エミュレーションには、身体障害者用のアクセシビリティー機能が用意されています。この製品を使用することで、CICS システムで必要なアクセシビリティー機能を使用することができます。

© Copyright IBM Corp. 1977, 2009

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]

アフィニティー
  CICS Interdependency Analyzer 78
アプリケーション ID テーブル 171, 175
アプリケーション所有領域 (AOR) 74
アプリケーション・プログラミング
  概説 265
  機能シップ用のための 267
  トランザクション・ルーティングのための 279
  非同期処理の 277
  CICS のマッピング、APPC verb への 435
  CICS-IMS 間 283
  DPL のための 271
  LUTYPE6.1 会話 (CICS-IMS 間) 283
一時記憶域
  機能シップ 40, 268
一時データ
  機能シップ 40, 269
インストール 129
  総称リソース、VTAM 141
  複数領域操作 135
  VTAM 総称リソース 141
インターバル制御機能
  機能シップ 37
オンライン・リソース定義 (RDO)
  間接リンク 207
  シップ可能端末定義 236
  接続、リモート・システムへの 165
  複数領域操作のリンク 181
  リモート VTAM 端末 234
  リモート・トランザクション 232
  リモート・リソース 223
  APPC 端末 191
  APPC リンク 186
  LUTYPE6.1 リンク 196
  LUTYPE6.2 リンク 186

[カ行]

解放済み、接続状況 214, 218
解放中、接続状況 218
会話
  LUTYPE6.1 セッション (CICS-IMS 間) 298
拡張対等ネットワーク機能 (APPN) 137
獲得済み、接続状況 214
間接リンク
  リソース定義 205
間接リンク、トランザクション・ルーティングのための
  概説 202
  シップ可能端末での 204
  どのような場合に必要か 205
  ハードコーディング端末での 204
  例 205
疑似会話型トランザクション
  トランザクション・ルーティングを使用した 280
機能シップ
  アプリケーション・プログラミング 267
  一時記憶域 40, 268
  一時データ 40, 269
  インターバル制御機能 37
  起こるキューイング 41
  主要な説明 37
  設計上の考慮事項 38
  短パス変換プログラム 45
  ファイル制御 39, 268
  ミラー・トランザクション 42
  ミラー・トランザクションの異常終了 270
  リモート・リソースの定義 225
    一時ストレージ・キュー 228
    一時データ宛先 228
    ファイル 226
    DL/I PSB 227
  例外条件 269
  DL/I 要求 39
基本会話 24
基本機能
  定義された 265
  デフォルト・プロファイル 254
基本マッピング・サポート (BMS)
  規則と制約事項の要約 432
  トランザクション・ルーティングを使用した 101, 279
キュー・モデル 256
競合勝者 26
競合敗者 26
共通プログラミング・インターフェース・コミュニケーション (CPI コミュニケーション)
  定義、パートナー 252
  同期レベル 24
  PIP データ 24
グローバル・ユーザー出口
  XALTENF 55, 82, 103
  XICTENF 55, 82, 103
  XISCONA 308
  XISQUE 308
  XPCREQ 109
  XPCREQC 109
  XZIQUE 308
検索、START コマンドによってシップされる情報の 59
限定リソース 25
  効果 219
交換回線
  コスト効率 25
構成 129
コマンド順序
  LUTYPE6.1 セッション (CICS-IMS 間) 301

[サ行]

サービス・クラス (COS) 25
  修正、モード名を提供するためのデフォルト・プロファイル 255
  モードセット 25, 186
サロゲート TCTTE 280
システム間 MRO (XCF/MRO)
  概説 28
システム間カップリング・ファシリティー (XCF)
  概説 28
  領域間通信に使用される 28
システム間キュー
  待機セッション要求の制御 41, 307
システム間セッション 23
システム間通信 (ISC)
  概念 3, 13
  機能 5
  互換の CICS ノードと IMS ノードの定義 197
  システム間の接続 21
  セッション 23
  待機セッション要求の制御 307
  多重チャネル・アダプター 22
  チャネル間通信 22

© Copyright IBM Corp. 1977, 2009

システム間通信 (ISC) *(続き)*
定義、互換 APPC ノード 189
定義、APPC 端末 191
定義、APPC モードセット 188
定義、APPC リンク 186
定義、LUTYPE6.1 リンク 196
定義された 3
トランザクション・ルーティング 73
ホスト内通信 22
SNA 経由 3, 13, 20, 132
VTAM 持続セッションの使用 194, 351
システム初期設定パラメーター
シップされた端末の削除のための 312
APPLID 166, 167, 208
DSHIPIDL 312
DSHIPINT 312
DTRTRAN 249
FSSTAFF 85
GRNAME 141
SYSIDNT 167, 168
VTAM 総称リソースのための 141
システム・メッセージ・モデル 256
シスプレックス、MVS
システム間カップリング・ファシリティー (XCF)
異なる MVS イメージにわたる MRO リンクのための 28
動的トランザクション・ルーティング 34
パフォーマンス
CICSPlex SM の使用 34
MVS ワークロード・マネージャーの使用 34
VTAM 総称リソースの使用 34, 137
持続セッション、VTAM 188, 189, 194, 351, 352
シップ可能端末
シップされるもの 237
選択的削除 311
「端末未認識」状態 81
リソース定義 238
ATI による 80
シップされた端末定義
削除
システム初期設定パラメーター 312
パフォーマンスの考慮事項 313
INQUIRE DELETSHIPPED コマンド 313
SET DELETSHIPPED コマンド 313
選択的削除メカニズム 311
タイムアウト削除メカニズム 312
シップされた端末定義の削除 311
自動インストール
シップされた端末定義の削除 311
ユーザー・プログラム、DFHZATDY 190
APPC 単一セッション端末の 191
APPC 単一セッションの
BIND 要求によって開始された 190
CINIT 要求によって開始された 191
APPC 並列セッションの 190
自動開始記述子 (AID) 79
自動トランザクション開始 (ATI)
一時データ・トリガー・レベルによる 260
規則と制約事項の要約 432
シップされた端末定義の制約事項 237
端末未認識状態 80
定義 79
トランザクション・ルーティング 79
非同期処理による 55
ルーティング・トランザクションによる制限 103
商標 448
据え置き伝送
LUTYPE6.1 セッション (CICS-IMS 間) 299
START NOCHECK 要求 57
スケジューラー・モデル 256
静止
接続処理 334
静的トランザクション・ルーティング
トランザクション定義
二重目的定義の使用 248
別々のローカル定義とリモート定義を使用する 247
セキュリティー
ルーティングされたトランザクションの 247
RTIMOUT オプション 247
セッションの障害
未確定期間中 319
セッション割り振り
LUTYPE6.1 セッション (CICS-IMS 間) 292
セッション・キューの管理
概説 307
QUEUELIMIT オプションの使用 308
XZIQUE グローバル・ユーザー出口の使用 308
セッション・バランシング
VTAM 総称リソースの使用 137
接続
定義、IPIC の 169
接続、リモート・システムへの
解放済み、状況 218
接続、リモート・システムへの *(続き)*
解放中、状況 218
獲得済み、状況 214
数の制限 23, 186
接続の解放 218
接続の獲得 214
定義 165
接続、リモート・トランザクションの
LUTYPE6.1 セッション (CICS-IMS 間) 294
接続静止プロトコル (CQP) 334
選択的削除、シップされた定義の 311
相互通信機能
概念 3, 13
総称 applid
特定 applid との関係 208
総称リソース、VTAM
アウトバウンドの LU6 接続 160
インストール 141
概説 34
シスプレックス間通信 145
制約事項 158
非自動インストール接続での使用 159
非自動インストール端末での使用 159
へのマイグレーション 142
要件 137
類縁性の終了 150

[タ行]

体系化プロセス
修正、デフォルト定義 257
プロセス名 257
リソース定義 257
体系化プロセス (モデル) 256
代替機能
定義された 265
デフォルト・プロファイル 254
タイプ 3 SVC ルーチン
および CICS applid 166, 167
領域間通信に使用される 27
領域間リンクについての指定 183
タイムアウト削除メカニズム、シップされた端末の 312
多重チャネル・アダプター 22
多重ミラー状態 44
短パス変換プログラム 45
端末専有領域 (TOR) 74
CICSplex における複数の総称リソース・グループのメンバーとしての 137
領域間のセッション・バランシング 137
端末の別名 243
チェーン・ミラー状態 44
チャネル間通信 22

中継トランザクション 100
トランザクション・ルーティングのための 73
長期実行ミラー・タスク 44
重複してシップされた端末定義 311
通信プロファイル 253
データ変換
アセンブル/リンク・エディット、変換プログラム 411
アラビア語の変換 368
ウルドゥー語の変換 379
韓国語の変換 373
キー・テンプレート 391
ギリシャ語の変換 371
キリル文字の変換 369
タイ語の変換 378
中国語 (簡体字) の変換 377
中国語 (繁体字) の変換 378
データ変換テンプレート用の DSECT 419
デーバナーガリーの変換 370
日本語の変換 372
非標準の変換 382
標準の変換 382
ベトナム語の変換 379
ヘブライ語の変換 371
ペルシア語の変換 370
変換処理 381
変換処理の順序 384
変換テーブルの定義 389, 411
変換テンプレート 391
変換の種類 361
文字データ 363
ラオ語の変換 374
リソース定義 387, 411
2 進整数 (INTEL 形式) 400
Baltic Rim の変換 369
C プログラミング言語、整数データ型 400
IVP (初期プログラム検査) 391
Latin 1 の変換 374
Latin 2 の変換 376
Latin 5 の変換 377
Latin 9 の変換 374
データ・ストリーム
IMS 通信のユーザー・データ・ストリーム 198
データ・テーブル 226
定義、IPIC 接続の 169
同期点 123, 318, 432
同期レベル 24, 123
CPI コミュニケーション 24
同情病
削減 307
動的トランザクション・ルーティング
シスプレックスにおける 34
動的トランザクション・ルーティング *(続き)*
紹介 75
トランザクション定義
同一定義の使用 248
別々のローカル定義とリモート定義を使用する 248
CRTX トランザクションの使用 249
TOR で単一定義を使用する 249
ルーティング・プログラムに渡される情報 76
ルーティング・プログラムの使用目的 77
ルーティング・プログラムの呼び出し 76
CICS Interdependency Analyzer 78
CICSPlex SM による制御 34, 78
CICSplex における 33
動的ルーティング
インターフェースの概要 65
動的ルーティング・プログラム、DFHDYP 75, 110
特定 applid
総称 applid との関係 208
トランザクション・ルーティング
アプリケーション・プログラミング 279
起こるキューイング 103
概説 73
間接リンク
概説 202
シップ可能端末での 204
定義方法 207
どのような場合に必要か 205
ハードコーディング端末での 204
例 205
疑似会話型トランザクション 280
基本マッピング・サポート 101, 279
システム・プログラミングに関する考慮事項 103
シップされた端末定義の削除 311
自動開始記述子 (AID) 79
自動トランザクション開始 80
セキュリティーの考慮事項 247
端末から開始された
静的 75
動的 75
動的ルーティング・プログラムに渡される情報 76
動的ルーティング・プログラムの使用目的 77
動的ルーティング・プログラムの呼び出し 76
端末のシップ 80
中継トランザクション 73
トランザクション・ルーティング *(続き)*
中継プログラム 100
リモート・リソースの定義
静的にルーティングされるトランザクション 247
端末 234, 239
動的にルーティングされるトランザクション 248
トランザクション 244
ルーティング・トランザクション、CRTE 102
AOR での ASSIGN コマンドの使用 280
APPC 端末 97
ATI 要求によって開始された 79
CICS Interdependency Analyzer 78

[ナ行]

名前
リモート・システム 168
ローカル CICS システム 166
二重目的 RDO 定義 248

[ハ行]

バックエンド・トランザクション
定義された 265
LUTYPE6.1 セッション (CICS-IMS 間) 297
パフォーマンス
シップされた端末定義の削除 311, 313
静的トランザクション・ルーティングの使用 33
待機セッション要求の制御 41, 58, 103, 113, 307
重複してシップされた端末定義 311
動的トランザクション・ルーティングの使用 34
CICSPlex SM の使用 34
DPL 要求の動的ルーティングの使用 34
MVS ワークロード・マネージャーの使用 34
VTAM 総称リソースの使用 34
非同期処理
アプリケーション・プログラミング 277
起こるキューイング 58
開始、リモート・トランザクション 54
システム・プログラミングに関する考慮事項 60
情報検索 59

非同期処理 *(続き)*
代表的なアプリケーション 51
端末の獲得 60
同期処理 (DTP) との比較 51
取り消し、リモート・トランザクション 54
パフォーマンスの向上 56
リモート・トランザクションの定義 232
例 60
ローカル・キュー 58
CICS-IMS 間 285
DTP によって開始される 52
NOCHECK オプション 56
PROTECT オプション 56
RETRIEVE コマンド 59
SEND および RECEIVE インターフェース 53
CICS-IMS 間アプリケーション 290
START コマンドによって渡される情報 54
START と RETRIEVE インターフェース 52, 53
CICS-IMS 間アプリケーション 285
非同期処理の方式 52
ファイル制御
機能シップ 39, 268
フィールド変換レコード 417, 420
フィールド変換レコード用の DFHCNVDS DSECT 419
複数領域操作 (MRO)
アプリケーション
タイム・シェアリング 33
プログラム開発 32
異常終了コード 434
インストールの注意点 135
応用方法 32
信頼できるデータベース・アクセス 33
部門の分離 33
マルチプロセッシング 33
ワークロード・バランシング 34
概念 27
間接リンク 202
機能 5, 27
互換ノードの定義 184
システム間 MRO (XCF/MRO) 28
シスプレックスにおける 34
待機セッション要求の制御 307
単一領域からの変換 35
短パス変換プログラム 45
長期実行ミラー・タスク 44
定義された 4
トランザクション・ルーティング 73
複数領域操作 (MRO) *(続き)*
領域間通信 27
リンクの定義 181
CICSplex における 33
MRO リンクの定義 181
VTAM 持続セッションの使用 351
プロファイル
基本機能 254
修正、デフォルト定義 255
代替機能のための 253
読み取りタイムアウト 254
リソース定義 253
CICS 提供のデフォルト 254
フロントエンド・トランザクション
定義された 265
LUTYPE6.1 セッション (CICS-IMS 間) 292
分散トランザクション処理 (DTP)
アプリケーション・プログラミング 283
概説 117
非同期処理との比較 51
リモート・リソースの定義 251
APPC 端末の API として 98
CICS-IMS 間 291
PARTNER 定義 251
分散プログラム・リンク (DPL)
アプリケーション・プログラミング 271
起こるキューイング 113
概説 105
グローバル・ユーザー出口 109
サーバー・プログラム 271
リソース定義 262
サーバー・プログラムでの制約事項 113
デイジー・チェーン要求 112
ミラー・トランザクションの異常終了 274
要求の静的ルーティング
サーバー・プログラムの定義 229
説明 106
要求の動的ルーティング
サーバー・プログラムの定義 229
紹介 109
ルーティングに対し適格 110
ルーティング・プログラムを呼び出す場合 111
リモート・サーバー・プログラムの定義 229
例 114
例外条件 273
ローカル・リソース定義 262
CICSPlex SM による制御 111, 230
分散ルーティング
トランザクション定義
同一定義の使用 248
BTS アクティビティーのルーティングのための 248
変換テンプレート 390, 391, 417, 420
フィールド変換レコード 417, 418, 419, 420
ホスト内 ISC 22

[マ行]

マイグレーション
単一領域操作から MRO への 35
トランザクション・ルーティング環境へのトランザクションの 279
マクロ・レベル・リソース定義
リモート DL/I PSB 227
リモート一時記憶域キュー 228
リモート一時データ宛先 228
リモート・サーバー・プログラム 229
リモート・トランザクション 232
リモート・ファイル 226
リモート・リソース 223
マッピング、APPC アーキテクチャーへの 435
制御オペレーターの verb 437
マップ式会話 24
未確定期間 319
セッションの障害 319
未定義の DBCS 文字 406
ミラー・トランザクション 42
長期実行ミラー・タスク 44
DPL のリソース定義 262
ミラー・トランザクションの異常終了 270, 274
無効な DBCS 文字 406
モードグループ
定義 25
SNASVCMG 214
モードセット 188
定義 25, 186
モード名 186
モデル 256

[ヤ行]

ユーザー置き換え可能プログラム
DFHDYP、動的ルーティング・プログラム 75

[ラ行]

リカバリーと再始動 317
同期点交換 318

リカバリーと再始動 *(続き)*
同期点フロー 319
動的トランザクション・バックアウト 323
未確定期間 319
リソース定義
概説 163
間接リンク 202
機能シップ 225
互換 MRO ノードの定義 184
互換の CICS ノードと IMS ノードの定義 197
修正、体系化プロセス定義 257
修正、デフォルト・プロファイル 255
体系化プロセス 257
データ変換 387, 411
定義、互換 APPC ノード 189
デフォルト・プロファイル 254
非同期処理 232
複数領域操作のリンク 181
プロファイル 253
分散トランザクション処理 251
変換テーブルの定義 389
ミラー・トランザクション 262
リモート DL/I PSB 227
リモート一時記憶域キュー 228
リモート一時データ宛先 228
リモート端末 234, 239
リモート・サーバー・プログラム 229
リモート・システムへのリンク 165
リモート・トランザクション 232, 244
リモート・パートナー 251
リモート・ファイル 226
リモート・リソース 223
ローカル・リソース 253
APPC 端末 191
APPC モードセット 188
APPC リンク 186
CICS-IMS 間 LUTYPE6.1 リンク 196
定義、複数のリンク 200
DPL 229, 262
サーバー・プログラム 262
LUTYPE6.1 リンク 196
LUTYPE6.2 リンク 186
リモート DL/I PSB 227
リモート一時記憶域キュー
定義 228
リモート一時データ宛先
定義 228
リモート端末
端末 ID 242
DFHTCT TYPE=REGION を使用する定義 241
DFHTCT TYPE=REMOTE を使用する定義 240
リモート・サーバー・プログラム
定義 229
プログラム名 230
リモート・システムへのリンク 165
リモート・トランザクション
トランザクション・ルーティングのための定義 244
静的ルーティング 247
動的ルーティング 248
非同期処理のための定義 232
ルーティングされたトランザクションのセキュリティー 247
リモート・ファイル
定義 226
ファイル名 226
レコード長 227
リモート・ファイルのレコード長 227
リモート・リソース
定義 223
命名 224
リモート・リソース定義の REMOTENAME オプション 224
リモート・リソースのローカル名 224
領域間通信 (IRC) 27
短パス変換プログラム 45
ルーティング・トランザクション、CRTE 102
自動トランザクション開始 103
CEMT の呼び出し 103
類縁性、総称リソースとパートナー LU の間の 150
例外条件
機能シップ 269
DPL 273
レベル、同期の 24
ローカル CICS システム
applid 167
総称と特定 208
sysidnt 168
ローカル CICS 領域
命名 166
applid 166
sysid 167
ローカル・キュー、START 要求の 58
ローカル・リソースの定義
区画内一時データ・キュー 259
体系化プロセス 256
通信プロファイル 253
DPL のための 262

[ワ行]

ワークロード・バランシング
動的トランザクション・ルーティングの使用 34
CICSPlex SM の使用 34
ワークロード・バランシング *(続き)*
DPL 要求の動的ルーティングの使用 34
MVS ワークロード・マネージャーの使用 34
VTAM 総称リソースの使用 34, 137

[数字]

1 次論理装置 (PLU) 26
2 次論理装置 (SLU) 26
2 進整数 (INTEL 形式)、変換 400

A

ACTION オプション 323
ACTION 属性
TRANSACTION 定義 323
AID (自動開始記述子) 79
ALLOCATE コマンド
APPC セッションを使用可能にする 215
LUTYPE6.1 セッション (CICS-IMS 間) 291, 292
SYSIDERR の後で LUTYPE6.1 接続をサービス可能にする 432
AOR (アプリケーション所有領域) 74
AOR での ASSIGN コマンド 280
APPC
概説 23
基本会話 24
サービス・クラス 25
持続セッション 194, 352
自動インストール
単一セッション端末の 191
並列セッション・リンクの 190
単一セッション
持続セッションの定義 195
自動インストール 190, 191
制限 25
リモート・リンク 191
同期レベル 24
並列セッション
持続セッションの定義 194
自動インストール 190
マスター端末操作 213
マッピング、APPC アーキテクチャーへの 435
マップ式会話 24
モードセット定義 188
リンク定義 186
リンク定義、端末の 191
LU サービス管理 24
LU サービス・マネージャー 186
APPC アーキテクチャーからの逸脱 445

APPC アーキテクチャーへの CICS マッピング
逸脱 445
APPC アーキテクチャーからの逸脱 445
APPC アーキテクチャーへのマッピング
逸脱 445
APPC 端末
持続セッション 195
自動インストール 190
代替機能としての 98
端末定義のシップ 237
トランザクション・ルーティング
ALLOCATE による 75, 98
ALLOCATE を使用 97
リモート定義 235
リンク定義 191
API 98
CEMT コマンドの使用 192
TYPETERM の AUTOCONNECT オプションの効果 194
APPLID
START コマンドによる受け渡し 55
applid
総称、XRF のための 208
特定、XRF のための 208
ローカル CICS の 166, 167
sysid との関係 167
sysidnt との関係 168
APPN (拡張対等ネットワーク機能) 137
ATI 中の端末未認識状態 81
AUTOCONNECT オプション
APPC 端末に対する DEFINE TYPETERM の 194
APPC での CEMT コマンドへの効果 214
APPC リソース定義 192
DEFINE CONNECTION の APPC に対する 193
DEFINE SESSIONS の APPC に対する 193

B

BIND
送信側と受信側 26
BTS アクティビティーのルーティング
トランザクション定義 248
BUILD ATTACH コマンド
LUTYPE6.1 セッション (CICS-IMS 間) 291, 294

C

C プログラミング言語、整数データ型の変換 400
CANCEL コマンド 54
CEMT マスター端末トランザクション
リモート端末による制限 433
ルーティング・トランザクションによる 103
APPC 端末に対する 192
DELETSHIPPED オプション 313
CICS Interdependency Analyzer 78
CICS のマッピング、APPC アーキテクチャーへの 435
CICSplex
トランザクション・ルーティング 33
パフォーマンス
VTAM 総称リソースの使用 137
CICSPlex SM による制御 34, 78, 111
CICSPlex SM
トランザクション・ルーティングを制御するための使用 34, 78
DPL 要求のルーティングの制御に使用 111, 230
CICS-CICS 間通信
互換ノードの定義
APPC セッション 189
MRO セッション 184
CICS-IMS 間通信
アプリケーション設計 283
アプリケーション・プログラミング 283
互換ノードの定義 197
通信形式 285
データ形式 283
非同期処理 285
CICS フロントエンド 286
IMS フロントエンド 287
CICS と IMS の比較 283
RETRIEVE コマンド 290
RU のチェーン形式 284
SEND および RECEIVE インターフェース 290
START コマンド 288
START と RETRIEVE インターフェース 285
VLVB 形式 284
CLINTCP 390
CNOS 折衝 216
CONNECTION 定義
PSRECOVERY オプション 195
CONVERSE コマンド
LUTYPE6.1 セッション (CICS-IMS 間) 291
CQP、接続静止プロトコルを参照 334
CRTE トランザクション 102
CRTX、CICS 提供のトランザクション定義 249
CSD (CICS システム定義ファイル)
領域間の共有
二重目的 RDO 定義 248

D

DBCS (2 バイト文字セット)
混合ストリング、SBCS/DBCS 400
定義、DBCS データ・フィールド 400
標準の変換に含まれる 361
無効な文字と未定義の文字 406
ユーザー定義の変換テーブル 403, 406
DBDCCICS 166, 167
DEFINE CONNECTION
間接リンク 207
APPC 端末 191
LUTYPE6.1 リンク 187, 196
MRO リンク 181
NETNAME オプション 168
DEFINE PROFILE 254
DEFINE SESSIONS
間接リンク 207
APPC 端末 191
LUTYPE6.1 リンク 188, 196
MAXIMUM オプション
APPC での CEMT コマンドへの効果 216
MRO リンク 181
DEFINE TERMINAL
シップ可能端末定義 238
リモート VTAM 端末 235
APPC 端末 191
DEFINE TRANSACTION
トランザクション・ルーティング 244
DYNAMIC オプション 245
PROFILE オプション 246
PROGRAM オプション 246
REMOTESYSTEM オプション 245
TASKREQ オプション 246
TRPROF オプション 246
TWASIZE オプション 246
非同期処理 232
ACTION オプション 323
WAIT オプション 323
DEFINE TYPETERM
APPC 端末 191
DFH0IPCC 171, 175
DFHCCNV、標準の変換プログラム 382

DFHCICSA
ALLOCATE によって獲得された代替機能のデフォルト・プロファイル 255
DFHCICSE
基本機能のデフォルト・エラー・プロファイル 255
DFHCICSF
機能シップのデフォルト・プロファイル 255
DFHCICSP
CSPG の基本機能のプロファイル 255
DFHCICSR
トランザクション・ルーティングのデフォルト・プロファイル
ユーザー・プログラムと領域間リンクの間で使われる 255
DFHCICSS
トランザクション・ルーティングのデフォルト・プロファイル
中継プログラムと領域間リンクの間で使われる 255
DFHCICST
基本機能のデフォルト・プロファイル 254
DFHCICSV
CSNE、CSLG、CSRS の基本機能のプロファイル 255
DFHCNV
CICSplex 管理 390
DFHCNV TYPE=DSECT マクロ 413
DFHCNV および SRVERCP の SYSDEF 値 390
DFHCNV、リソース定義マクロ 389, 411
コード化に関するヒント 401
コード化例 407
マクロのタイプ 389
TYPE=ENTRY 394
TYPE=FIELD 399
TYPE=FINAL 400
TYPE=INITIAL 392
TYPE=IVP 391
TYPE=KEY 397
TYPE=SELECT 398
DFHDLPSB TYPE=ENTRY マクロ 227
DFHDYP、動的ルーティング・プログラム 75, 110
DFHTCT TYPE=REGION マクロ 241
DFHTCT TYPE=REMOTE マクロ 240
DFHTST TYPE=REMOTE マクロ 229
DFHUCNV、ユーザー置き換え可能変換プログラム
データ変換テンプレート用の DSECT 419
提供されているバージョン 420
DFHUCNV、ユーザー置き換え可能変換プログラム *(続き)*
パラメーター・リスト、DFHUCNV 413
パラメーター・リストの DSECT 414
変換処理 382, 385
変換テンプレート 417
DFHCNV TYPE=DSECT マクロ 413
DFHUNVDS、DFHUCNV パラメーター・リストの DSECT 414
DFHZATDY、自動インストール・ユーザー・プログラム 190
DL/I
機能シップ 39
定義、リモート PSB 227
DL/I モデル 256
DPL 要求の動的ルーティング
サーバー・プログラムの定義 229
シスプレックスにおける 34
紹介 109
ルーティングに対し適格 110
ルーティング・プログラムを呼び出す場合 111
CICSPlex SM による制御 34
DSHIPIDL、システム初期設定パラメーター 312
DSHIPINT、システム初期設定パラメーター 312
DTRTRAN、システム初期設定パラメーター 249
DYNAMIC オプション
リモート・トランザクション定義での 245

E

EIB フィールド
LUTYPE6.1 セッション (CICS-IMS 間) 300
EXTRACT ATTACH コマンド
LUTYPE6.1 セッション (CICS-IMS 間) 291, 297

F

FREE コマンド
LUTYPE6.1 セッション (CICS-IMS 間) 292, 299
FSSTAFF、システム初期設定パラメーター 85

G

GRNAME、システム初期設定パラメーター 141

I

IMS
非応答モード・トランザクション 286
非会話型トランザクション 286
メッセージ交換 286
CICS との比較 283
IP 相互接続
概念 13
IPIC 13
IP 相互接続 (IPIC)
概念 3
相互通信機能 14
定義された 3
IPCONN
マイグレーション、APPC 接続と MRO 接続 171, 175
IPIC
概念 3
IPIC 接続
定義 169
IPIC 接続性
マイグレーション、APPC 接続と MRO 接続 171, 175
ISC
相互通信機能 21
ISC over SNA
相互通信機能 21
ISSUE SIGNAL コマンド
LUTYPE6.1 セッション (CICS-IMS 間) 292
IVP (初期プログラム検査)、データ変換テーブル 391

L

LAST オプション 299
LU サービス管理
説明 24
LU サービス・マネージャー
SNASVCMG セッション 186
LU サービス・モデル 256
LUTYPE6.1
リンク定義 196
CICS-IMS 間アプリケーション・プログラミング 283
LUTYPE6.2
リンク定義 186
LU-LU セッション 23
競合 26
1 次 LU、2 次 LU 26

M

MAXIMUM オプション、DEFINE SESSIONS コマンド
  APPC での CEMT コマンドへの効果 216
MAXQTIME オプション、CONNECTION 定義 41, 308
MAXQTIME オプション、IPCONN 定義 308
MODENAME 216
MVS イメージ
  シスプレックスにおけるイメージ間の MRO リンク 28
MVS 仮想記憶間サービス
  領域間リンクについての指定 183

N

NETNAME 属性、CONNECTION リソースの
  デフォルト 169
  sysidnt へのマッピング 168
NOCHECK オプション
  START コマンドの 56
    ローカル・キューに必須の 58
NOQUEUE オプション
  ALLOCATE コマンドの
    LUTYPE6.1 セッション (CICS-IMS 間) 292

P

PARTNER 定義、DTP のための 251
PIP データ
  紹介 24
  CPI コミュニケーションでの 24
PRINSYSID オプション、ASSIGN コマンド 280
PROFILE オプション、ALLOCATE コマンドの
  リモート・トランザクション定義での 246
  LUTYPE6.1 セッション (CICS-IMS 間) 292
PROGRAM オプション
  リモート・トランザクション定義での 246
PROTECT オプション、START コマンドの 56
PSRECOVERY オプション
  CONNECTION 定義 195

Q

QUEUELIMIT オプション、CONNECTION 定義 41, 308
QUEUELIMIT オプション、IPCONN 定義 308

R

RECEIVE コマンド
  LUTYPE6.1 セッション (CICS-IMS 間) 291
RECOVOPTION オプション
  SESSIONS 定義 195
  TYPETERM 定義 195
REMOTESYSNET オプション
  CONNECTION 定義 204, 235
  TERMINAL 定義 204, 235
REMOTESYSTEM オプション
  CONNECTION 定義 204, 235
  TERMINAL 定義 204, 235
  TRANSACTION 定義 245
RETRIEVE コマンド
  CICS-IMS 間通信 290
  WAIT オプション 59
RTIMOUT オプション
  通信プロファイルの 247
  PROFILE 定義 254
RU のチェーン形式 284

S

SEND および RECEIVE、非同期処理 53
  CICS-IMS 間通信 290
SEND コマンド
  LUTYPE6.1 セッション (CICS-IMS 間) 291
SESSION オプション、ALLOCATE コマンドの
  LUTYPE6.1 セッション (CICS-IMS 間) 292
SESSIONS 定義
  RECOVOPTION オプション 195
SNA 経由のシステム間通信
  概念 3, 13, 20
  構成 132
SNASVCMG セッション
  目的 25
  CICS による生成 186
SRVERCP 390
START NOCHECK コマンド
  据え置き送信 57
  ローカル・キューのための 58
START PROTECT コマンド 56
START および RETRIEVE の非同期処理
  CICS-IMS 間通信 285
START コマンド
  CICS-IMS 間通信 288
  NOCHECK オプション 56
    ローカル・キューのための 58
START と RETRIEVE の非同期処理 52, 53
sysid
  ローカル CICS 領域の 167
  applid との関係 167
SYSID 値
  デフォルト 167
  ローカル CICS 領域の 167
SYSID キーワード、ALLOCATE コマンドの
  LUTYPE6.1 セッション (CICS-IMS 間) 292
sysidnt
  リモート・システムの 168
  ローカル CICS システムの 168
  applid との関係 168
SYSIDNT 値
  デフォルト 168
  リモート・システムの 168
  ローカル CICS システム 167, 168
  ローカル CICS システムの 168
  NETNAME へのマッピング 168

T

TASKREQ オプション
  リモート・トランザクション定義での 246
TCP/IP (Transport Control Protocol/Internet Protocol) 3, 13
TCP/IP 管理および制御
  概説 209
TCTTE、サロゲート 280
TERMINAL 定義
  REMOTENAME オプション 243
  REMOTESYSNET オプション 235
  REMOTESYSTEM オプション 235
TOR (端末専有領域) 74
  CICSplex における複数の
    総称リソース・グループのメンバーとしての 137
    領域間のセッション・バランシング 137
TRANSACTION 定義
  ACTION 属性 323
  WAIT 属性 323
  WAITTIME 属性 323
Transport Control Protocol/Internet Protocol (TCP/IP) 3, 13
TRPROF オプション
  リモート・トランザクション定義での 246

TRPROF オプション *(続き)*
ルーティング・トランザクション (CRTE) の 102
TWASIZE オプション
リモート・トランザクション定義での 246
TYPETERM 定義
RECOVOPTION オプション 195

U

USERID オプション、ASSIGN コマンド 281

V

VLVB 形式 284
VTAM
限定リソース 25
持続セッション
リカバリーと再始動への影響 352
リンク定義 194
MRO と ISC リンク 351
XRF との比較 351
総称リソース
アウトバウンドの LU6 接続 160
インストール 141
概説 34
シスプレックス間通信 145
制約事項 158
非自動インストール接続での使用 159
非自動インストール端末での使用 159
へのマイグレーション 142
要件 137
モードグループ (複数) 25
類縁性の終了 150
APPN ネットワーク・ノード 137
LOGMODE 項目 25, 186

W

WAIT オプション 323
RETRIEVE コマンドの 59
WAIT コマンド
LUTYPE6.1 セッション (CICS-IMS 間) 291
WAIT 属性
TRANSACTION 定義 323
WAITTIME 属性
TRANSACTION 定義 323

X

XALTENF、グローバル・ユーザー出口 55, 82, 103, 238
XCF (システム間カップリング・ファシリティー)
概説 28
XCF/MRO (システム間 MRO)
概説 28
XICTENF、グローバル・ユーザー出口 55, 82, 103, 238
XISCONA、グローバル・ユーザー出口
システム間のキューイングを制御するための 41
XPCREQC、グローバル・ユーザー出口 109
XPCREQ、グローバル・ユーザー出口 109
XRF (拡張リカバリー機能) 349
持続セッションとの比較 351
applid、総称と特定 208
XZIQUE、グローバル・ユーザー出口
システム間のキューイングを制御するための 41

SC88-5857-00

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