

CICS Transaction Server for z/OS



相互通信ガイド

バージョン 3 リリース 2

CICS Transaction Server for z/OS



相互通信ガイド

バージョン 3 リリース 2

ご注意

本書および本書で紹介する製品をご使用になる前に、385 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 3 リリース 2 (プログラム番号 5655-M15)、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。必ず、製品レベルに対応した正しい版を使用してください。

IBM 発行のマニュアルに関する情報のページ

<http://www.ibm.com/jp/manuals/>

こちらから、日本語版および英語版のオンライン・ライブラリーをご利用いただけます。また、マニュアルに関するご意見やご感想を、上記ページよりお送りください。今後の参考にさせていただきます。

(URL は、変更になる場合があります)

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： SC34-6829-00
CICS Transaction Server for z/OS
Version 3 Release 2
Intercommunication Guide

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2007.6

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

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

© Copyright International Business Machines Corporation 1977, 2007. All rights reserved.

© Copyright IBM Japan 2007

目次

前書き	xiii
本書について	xiii
本書で取り扱わない内容	xiii
本書の対象読者	xiv
本書を理解するために必要な知識	xiv
本書の利用方法	xiv
本書の構成	xiv
用語	xv
変更の要約	xvii
CICS Transaction Server for z/OS, バージョン 3 リリース 2 の変更点	xvii
CICS Transaction Server for z/OS バージョン 3 リリース 1 における変更点	xvii
CICS Transaction Server for z/OS バージョン 2 リリース 3 における変更点	xvii

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
複数領域操作	3
システム間通信	4
相互通信機能	5
CICS 機能シッ	6
非同期処理	6
CICS トランザクション・ルーティング	7
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	8
地域本部の接続	10
組織内の各部門の接続	11
第 2 章 複数領域操作	13
MRO の概要	13
MRO を介して使用できる機能	14
システム間複数領域操作 (XCF/MRO)	14
XCF/MRO の利点	18
複数領域操作の応用	18
プログラム開発	18
タイム・シェアリング	19
信頼できるデータベース・アクセス	19
部門の分離	19
マルチプロセッサのパフォーマンス	19
シスプレックスにおけるワークロード・balancing	20
仮想記憶域制約解放	21
単一領域システムからの変換	21
第 3 章 ISC および IPIC 相互通信機能	23
SNA 経由のシステム間通信	23
サブシステム間の接続	23
システム間セッション	25

I	システム間セッションの確立	28
I	IP 相互接続	28
	第 4 章 CICS 機能シッ	31
	機能シッの概要	31
	設計上の考慮事項	32
	ファイル制御	32
	DL/I	33
	一時記憶域	33
	一時データ	34
	システム間のキューイング	34
	ミラー・トランザクションと変換プログラム	35
	ISC 機能シッ	36
	MRO 機能シッ	37
	ミラー・トランザクションの異常終了処理	39
	機能シッの例	39
	第 5 章 非同期処理	43
	非同期処理の概要	43
	非同期処理方式	44
	START と RETRIEVE コマンドを使用する非同期処理	45
	リモート・トランザクションの開始と取り消し	45
	START コマンドによって渡される情報	46
	システム間 START 要求のパフォーマンスの向上	47
	作業単位への開始要求送達の組み込み	48
	NOCHECK オプションを指定した START 要求の据え置き送信	48
	システム間のキューイング	49
	開始されたトランザクションによるデータ検索	50
	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	51
	システム・プログラミングに関する考慮事項	51
	非同期処理の例	52
	第 6 章 CICS 動的ルーティングの紹介	57
	動的ルーティングとは何か	57
	2 つのルーティング・モデル	58
	「ハブ」モデル	59
	分散モデル	60
	2 つのルーティング・プログラム	61
	第 7 章 CICS トランザクション・ルーティング	65
	トランザクション・ルーティングの概要	65
	トランザクション・ルーティングの開始	66
	端末開始トランザクション・ルーティング	67
	静的トランザクション・ルーティング	67
	動的トランザクション・ルーティング	67
	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	70
	自動トランザクション開始用端末のシッ	72
	ATI および総称リソース	79
	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	79
	拡張方法の利点	79
	端末関連の START コマンド	80
	非端末関連の START コマンド	85
	リモート APPC 接続の割り振り	89

APPC デバイスを使用したトランザクション・ルーティング	89
代替機能の割り振り	90
端末としてのシステム	90
中継プログラム	92
基本マッピング・サポート (BMS)	93
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	93
ルーティング・トランザクション (CRTE) の使用	94
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	95
システム間のキューイング	95
第 8 章 CICS 分散プログラム・リンク	97
DPL の概要	97
DPL 要求の静的ルーティング	98
ミラー・トランザクションの使用	99
グローバル・ユーザー出口による DPL 要求の宛先変更	101
DPL 要求の動的ルーティング	101
動的にルーティングできる要求	102
動的ルーティング・プログラムを呼び出す場合	103
要求のルーティングのための CICSplex SM の使用	103
DPL 要求の「デ이지ー・チェーン」	104
DPL サーバー・プログラムでの制約事項	105
システム間のキューイング	105
DPL の例	106
第 9 章 分散トランザクション処理	109
DTP の概要	109
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	109
なぜ分散トランザクション処理なのか	111
会話とは何か、なぜ必要なのか	111
会話の開始とトランザクション階層	111
2 つのトランザクション間でのダイアログ	112
制御フローと制御ブラケット	113
会話の状態とエラーの検出	114
同期	114
DTP では MRO か APPC か	116
APPC マップ式会話か、基本会話か	117
EXEC CICS か CPI コミュニケーションか	118

第 2 部 相互通信サポートのインストール 119

第 10 章 複数領域操作のインストールの注意点	121
MRO のインストール手順	121
CICS を MVS のサブシステムとして追加する	121
MRO で必要なモジュール	121
MVS リンク・パック域の MRO モジュール	121
MRO データ・セットと起動システム	122
XCF/MRO の要件	122
シスプレックスのハードウェアとソフトウェアの要件	122
XCF/MRO サポートの生成	122
インストール後の手順	124
第 11 章 システム間通信のインストールの注意点	125

	SNA 経由のシステム間通信用のサポートのインストール	125
	ISC に必要なモジュール	125
	CICS のための ACF/VTAM 定義	125
	IMS の考慮事項	127
I	IP 相互接続用のサポートのインストール	132
	第 12 章 VTAM 総称リソースのインストールの注意点	133
	VTAM 総称リソースの前提条件	133
	VTAM 総称リソースを使用するための CICSplex の計画	134
	CICS 領域の命名	135
	総称リソース環境における接続の定義	136
	接続の定義	136
	VTAM 総称リソース・サポートの生成	137
	総称リソースへの TOR のマイグレーション	138
	推奨される方法	139
	TOR を総称リソースから除去する	140
	TOR を別の総称リソースへ移動する	141
	総称リソース間でのシスプレックス間通信の設定	141
	CICS TS for z/OS 総称リソース間での接続の確立	141
	類縁性の終了	146
	アフィニティを終了すべき場合	147
	アフィニティを終わらせるバッチ・プログラムの作成	147
	ATI での総称リソースの使用	150
	ISSUE PASS コマンドの使用	153
	規則のチェックリスト	154
	特殊な事例の対処	155
	非自動インストール端末と接続	155
	アウトバウンドの LU6 接続	156
	第 3 部 相互通信リソースの定義	159
	第 13 章 リモート・システムへのリンクの定義	161
	リンク定義の紹介	161
	ローカル CICS システムの命名	162
	リモート・システムの識別	163
	複数領域操作のリンクの定義	164
	MRO リンクの定義	164
	MRO のアクセス方式の選択	166
	互換 MRO ノードの定義	167
	外部 CICS インターフェイスが使用するリンクの定義	168
	MRO と EXCI リンク定義のインストール	169
I	IP 相互接続リンクの定義	169
	APPC リンクの定義	174
	リモート APPC システムの定義	175
	APPC セッション・グループの定義	176
	互換 CICS APPC ノードの定義	177
	APPC リンクの自動インストール	178
	単一セッション APPC 端末の定義	179
	AUTOCONNECT オプション	181
	APPC リンクでの VTAM 持続セッションの使用	182
	論理装置タイプ 6.1 リンクの定義	184
	CICS-IMS 間 LUTYPE6.1 リンクの定義	184

互換の CICS ノードと IMS ノードの定義	185
IMS システムへの複数リンクの定義	189
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	191
CICS Transaction Server for z/OS で間接リンクの定義が必要になる理由	193
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	194
XRF のための総称 applid と特定 applid	196
第 14 章 APPC リンクの管理	199
APPC リンク管理に関する一般情報	199
接続の獲得	200
獲得プロセス中の接続状況	200
AUTOCONNECT オプションの効果	200
MAXIMUM オプションの効果	202
SET MODENAME コマンドによるセッションの制御	202
コマンドの有効範囲と制約事項	203
接続の解放	204
解放プロセス中の接続状況	204
限定リソースの効果	205
接続を使用不能にする	205
APPC リンク管理の要約	207
コマンドの有効範囲と制約事項	208
第 15 章 TCP/IP 管理および制御	209
第 16 章 リモート・リソースの定義	213
どのリモート・リソースを定義しなければならないか	213
「デイジー・チェーン」に関する注記	214
リソースのローカル名とリモート名	214
機能シップのためのリモート・リソースの定義	215
リモート・ファイルの定義	216
リモート DL/I PSB の定義	217
リモート一時データ宛先の定義	218
リモート一時記憶域キューの定義	218
DPL のリモート・リソースの定義	219
リモート・サーバー・プログラムの定義	219
リモート・サーバー・プログラムの定義が必要ない場合	220
非同期処理のためのリモート・リソースの定義	222
リモート・トランザクションの定義	222
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	223
トランザクション・ルーティングにおける端末の定義	223
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	234
DTP のリモート・リソースの定義	241
第 17 章 ローカル・リソースの定義	243
通信プロファイルの定義	243
基本機能の通信プロファイル	244
デフォルト・プロファイル	244
デフォルト・プロファイルの修正	245
体系化プロセス	246
プロセス名	247
体系化プロセス定義の修正	247
インストールが必要なリソース定義の選択	248

区画内一時データ・キューの定義	249
トランザクション	250
基本機能	250
DPL のローカル・リソースの定義	252
ミラー・トランザクション	252
サーバー・プログラム	252

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 253

第 18 章 アプリケーション・プログラミングの概要	255
用語	255
問題判別	256

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	257
機能シップのためのプログラミングの紹介	257
ファイル制御	258
DL/I	258
一時記憶域	258
一時データ	259
機能シップの例外条件	259
リモート・システム使用不能	259
無効な要求	259
ミラー・トランザクションの異常終了	260

第 20 章 CICS DPL のアプリケーション・プログラミング	261
DPL プログラムの紹介	261
クライアント・プログラム	261
サーバー・プログラムの障害	262
サーバー・プログラム	262
許可されているコマンド	262
同期点	262
DPL の例外条件	263
リモート・システム使用不能	263
サーバー作業のバックアウト	263
同じサーバー領域に対する複数のリンク	263
ミラー・トランザクションの異常終了	264
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	264

第 21 章 非同期処理のアプリケーション・プログラミング	267
リモート・システムでのトランザクションの開始	267
START コマンドの例外条件	267
リモートで出された開始要求に関連したデータの検索	267

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	269
注意すべき事項	269
基本マッピング・サポート	269
疑似会話型トランザクション	270
AOR での EXEC CICS ASSIGN コマンドの使用	270

第 23 章 CICS-IMS 間アプリケーション	273
CICS-IMS 間 ISC アプリケーションの設計	273
データ形式	273

IMS とのシステム間通信の形式	275
CICS-IMS 間アプリケーション - 非同期処理	275
START と RETRIEVE インターフェース	275
非同期の SEND と RECEIVE インターフェース	280
CICS-IMS 間アプリケーションの DTP	281
CICS-IMS 間セッションに対する CICS コマンド	281
フロントエンド・トランザクションに関する考慮事項	282
リモート・トランザクションの接続	284
バックエンド・トランザクションに関する考慮事項	287
会話	288
セッションの解放	289
EXEC インターフェース・ブロック (EIB)	289
CICS-IMS 間セッションのコマンド順序	291
状態遷移	292

第 5 部 システム間環境におけるパフォーマンス 295

第 24 章 システム間のセッション・キューの管理	297
セッション・キュー管理の概要	297
割り振りキューの管理	297
接続定義だけの使用	297
NOQUEUE オプションの使用	298
XZIQUE グローバル・ユーザー出口の使用	298
第 25 章 シップされた端末定義の効率的な削除	301
シップされた端末の削除方法の概要	301
選択的削除	301
タイムアウト削除メカニズム	302
タイムアウト削除の実装	302
タイムアウト削除のパフォーマンスのチューニング	303
DSHIPIDL	303
DSHIPINT	304

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 305

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	307
用語	307
同期点交換	308
同期点フロー	309
リカバリーの機能とインターフェース	311
リカバリー機能	312
リカバリー・インターフェース	312
初期始動とコールド・スタート	316
コールド・スタートがいつ可能かの決定	317
ログ名交換プロセス	318
接続定義の管理	319
CICS TS for z/OS システムへの MRO 接続	320
CICS TS for z/OS システムへの IPIC 接続	320
CICS TS for z/OS システムへの APPC 並列セッション接続	320
VTAM 総称リソースとの間の APPC 接続	320
棚上げを完全にはサポートしない接続	321
LU6.1 接続	321

非 CICS TS for z/OS システムへの APPC 接続	323
APPC 単一セッション接続	323
APPC 接続静止処理	324
問題判別	324
CICS のリカバリー・アクションを知らせるメッセージ	324
問題判別の例	328
第 27 章 相互通信と XRF	339
MRO セッション	339
LUTYPE6.1 セッション	339
単一セッション APPC デバイス	340
並列 APPC セッション	340
アプリケーション・プログラムへの影響	340
第 28 章 相互通信と VTAM 持続セッション	341
持続セッションのサポートと XRF との比較	341
相互接続された CICS の環境、リカバリー、および再始動	342
MRO セッション	342
LU6.1 セッション	342
LU6.2 セッション	342
アプリケーション・プログラムへの影響	343

第 7 部 付録 345

付録 A. 相互通信規則と制約事項のチェックリスト	347
トランザクション・ルーティング	347
DPL 要求の動的ルーティング	349
自動トランザクション開始	349
基本マッピング・サポート	350
LUTYPE6.1 セッションの獲得	350
同期点処理	350
ローカル名とリモート名	350
マスター端末トランザクション	351
インストールと操作	351
リソース定義	351
カスタマイズ	351
MRO 異常終了コード	351
付録 B. CICS の APPC アーキテクチャーへのマッピング	353
サポートされるオプション・セット	353
制御オペレーター verb の CICS による実装	354
制御オペレーターの verb	355
制御オペレーター verb の戻りコード	362
APPC アーキテクチャーからの逸脱	363
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	363
参考文献	365
CICS Transaction Server for z/OS ライブラリー	365
同梱セット	365
PDF のみの資料	365
その他の CICS の資料	367
関連ライブラリーの資料	368

IMS.	368
MVS/ESA.	368
ネットワーク・プログラム・プロダクト	368
システム・アプリケーション体系 (SAA)	368
システム・ネットワーク体系 (SNA).	368
VTAM.	368
出版物が最新であるかどうかの判別	369
 アクセシビリティ	 371
 索引	 373
 特記事項.	 385
プログラミング・インターフェース情報	386
商標	386

前書き

本書について

本書の内容は、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) のネットワーク機能を使用しない、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 経由のシステム間通信 (ISC over SNA): 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 相互接続 (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 3 リリース 2 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 3.2 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。

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。

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。

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 経由のシステム間通信 (ISC over SNA)、IP 相互接続 (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すで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 3 リリース 2 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 経由のシステム間通信 (ISC over SNA) および IP 相互接続 (IPIC) を意味する総称名です。ISC over SNA 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、4 ページの『システム間通信』を参照してください。
3. 用語「IP 接続」は、IP 相互接続を意味します。
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。

変更の要約

本書は、「CICS Intercommunication Guide for CICS Transaction Server for z/OS, Version 3 Release 1」(SC34-6448-00) を基に作成されています。この版から変更された個所には、その左マージンに垂直バーのマークが示されています。

CICS Transaction Server for z/OS, バージョン 3 リリース 2 の変更点

CICS Transaction Server for z/OS, バージョン 3 リリース 2 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 CICS TS V3.1* からのマイグレーション
- *CICS Transaction Server for z/OS CICS TS V2.3* からのマイグレーション
- *CICS Transaction Server for z/OS CICS TS V2.2* からのマイグレーション
- *CICS Transaction Server for z/OS CICS TS V1.3* からのマイグレーション

CICS Transaction Server for z/OS バージョン 3 リリース 1 における変更点

この版における大きな変更点はありません。

CICS Transaction Server for z/OS バージョン 2 リリース 3 における変更点

この版における主な変更点には、以下のものがあります。

- 以下の CICS 製品はサポートが終了しているため、これらの製品に言及している個所を削除しました。
 - CICS/ESA バージョン 4
 - CICS/ESA バージョン 3
 - CICS/MVS バージョン 2

第 1 部 相互通信の概念と機能

本書のこのパートでは、CICS 相互通信の基本概念および提供される各種機能について説明します。CICS 相互通信の定義を示し、複数領域操作およびシステム間通信という 2 つのタイプの相互通信を紹介します。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)

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。

重要: この文書の情報は、主として、CICS Transaction Server for z/OS と、他の System/390 の CICS システムまたは IMS システムの間の通信に関するものです。CICS Transaction Server for z/OS 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 zSeries*」を参照してください。

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、「*CICS インターネット・ガイド*」を参照してください。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』
- 5 ページの『相互通信機能』
- 8 ページの『CICS 相互通信機能の使用』

相互通信方式

CICS が他のシステムと通信を行う方法には、**複数領域操作 (MRO)** と **システム間通信 (ISC)** の 2 つがあります。

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**があります。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

システム間通信

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

SNA 経由のシステム間の通信の一般名は、**システム間通信 (ISC)** です。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。

注: 通常、ISC および IPIC 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のに使用されます。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 接続がすでに存在する場合に、その 2 つの CICS 領域間で ISC 接続を作成する場合などです。

ISC と IPIC では、異なったレベルの機能が提供されます。

SNA 経由のシステム間通信 (ISC over SNA)

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、ISC over SNA 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。基本的な CICS のすべての相互通信機能 (後で説明) がサポートされています。

CICS Transaction Server for z/OS は、ISC over SNA 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
- CICS Transaction Server for VSE/ESA™
- CICS/VSE® バージョン 2
- CICS Transaction Server for iSeries™
- CICS/400® バージョン 4
- オープン・システム CICS
- CICS Transaction Server for Windows

- IMS/ESA[®] バージョン 4
- IMS/ESA バージョン 5
- 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

IP 相互接続 (IPIC)

IPICは、CICS と以下の間で使用することができます。

- 別の CICS TS for z/OS、バージョン 3.2 (またはそれ以降) の領域。同じ z/OS シスプレックス内にリモート領域が存在したり、存在しないことがあります。現在、サポートされる基本的な CICS 相互通信機能のみが分散プログラム・リンク (DPL) です。
- Java クライアント。IPICによる接続は、MRO および ISC over SNA 接続が対応していない方法で、Java[™] Connector Architecture (JCA) をサポートします。提供される JCA リソース・アダプターは、CICS 以外で実行される J2EE コンポーネント (サーブレットまたはエンタープライズ Bean) が CICS TS サーバー・プログラムを TCP/IP 経由で呼び出すことを可能にします。

ISC over SNA および IPICの追加情報については、23 ページの『第 3 章 ISC および IPIC 相互通信機能』を参照してください。

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、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 には以下の基本的な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)

注: 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する、以下の相互通信機能については、本書ではなく、「*CICS External Interfaces Guide*」の『CICS 外部インターフェースの概説』または「*CICS インターネット・ガイド*」の `./../com.ibm.cics.ts.internet.doc/topics/dfhtl_part1.dita#dfhtl50` を参照してください。

- CICS ブリッジ
- 外部 CICS インターフェース
- DCE リモート手続き呼び出しのサポート
- ONC リモート手続き呼び出しのサポート
- CICS Web インターフェース

これらの基本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	システム間通信SNA 経由 (ACF/VTAM による)				システム間通信TCP/IP 経由	
	MRO	LUTYPE6.2 (APPC)		LUTYPE6.1		TCP/IP	
機能	CICS	CICS	非 CICS	CICS	IMS	CICS	非 CICS
機能シップ	使用可	使用可	使用不可	使用可	使用不可	使用不可	使用不可
非同期処理	使用可	使用可	使用不可	使用可	使用可	使用不可	使用不可
トランザクション・ルーティング	使用可	使用可	使用不可	使用不可	使用不可	使用不可	使用不可
分散プログラム・リンク	使用可	使用可	使用不可	使用不可	使用不可	使用可	使用不可
分散トランザクション処理	使用可	使用可	使用可	使用可	使用可	使用不可	使用不可

CICS 機能シップ

CICS 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。

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。

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、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、領域間リンク (MRO) または APPC リンクのいずれかによって接続された複数の CICS システム間で使用することができます。

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。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、機能シッパ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、ISC over SNA 接続同様、IPICでサポートされます。

分散トランザクション処理 (DTP)

CICS は、機能シッパ、分散プログラム・リンク、非同期トランザクション処理、またはトランザクション・ルーティングを準備するときに、リモート・システムとの論理データ・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。アプリケーション・プログラムでも同等のコマンドを出して、アプリケーションが CICS または非 CICS のアプリケーションと会話できるようにすることができます。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。

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。

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。この項では、いくつかの代表的な応用について説明します。

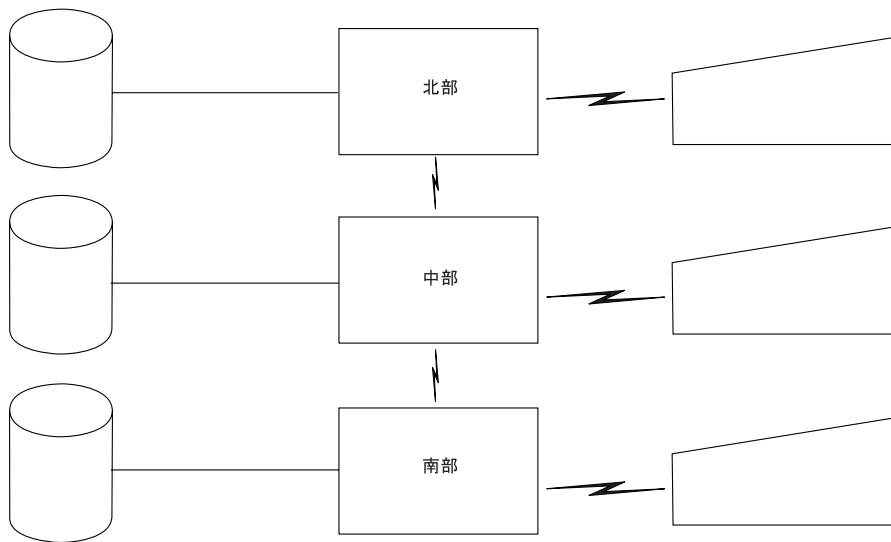
複数領域操作を行うと、2 つの CICS 領域が、選択されたシステム・リソースを共用しながら、端末オペレーターに対しては「単一システム」としての外観を示すことができます。また同時に、各領域を相互に無関係な状態で実行して、一方の領域で起こったエラーからもう一方の領域を保護することができます。MRO の各種の応用方法については、13 ページの『第 2 章 複数領域操作』で説明します。

CICS SNA 経由のシステム間通信を ACF/VTAM アクセス方式および ACF/NCP/VS ネットワーク制御とともに使用すると、異なる複数のシステム間でリソースを分散し、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。

注: IP 相互接続を使用して分散システムに接続することもできますが、現在サポートされている基本 CICS 相互接続機能は、分散プログラム・リンクのみです。

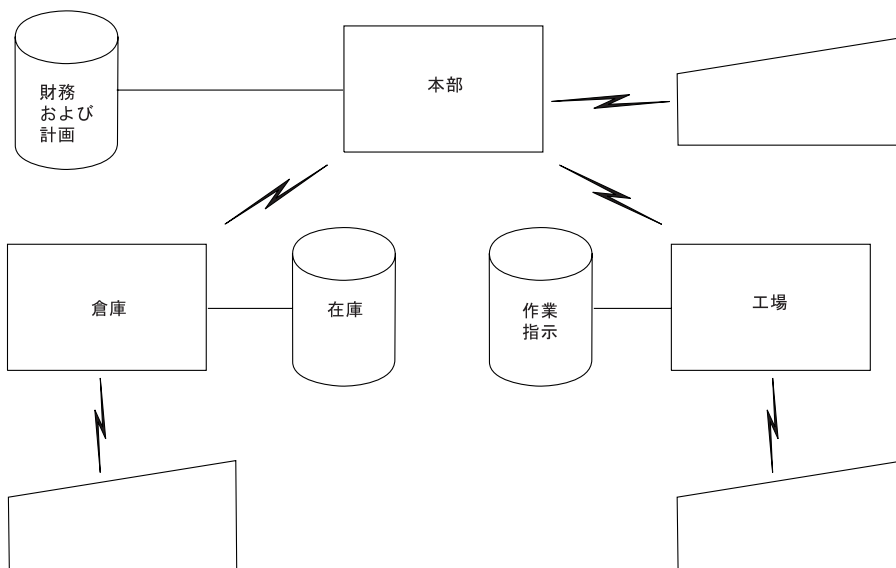
9 ページの図 1 に、いくつかの代表的な形態を示します。

地域本部の接続



- データベースは地域ごとに区分される
- 同じアプリケーションがそれぞれのセンターで実行される
- すべての端末ユーザーがすべてのシステム内のアプリケーションまたはデータにアクセスできる
- 端末オペレーターおよびアプリケーションはデータの場所を意識しない
- 地域外要求は該当のシステムにルーティングされる

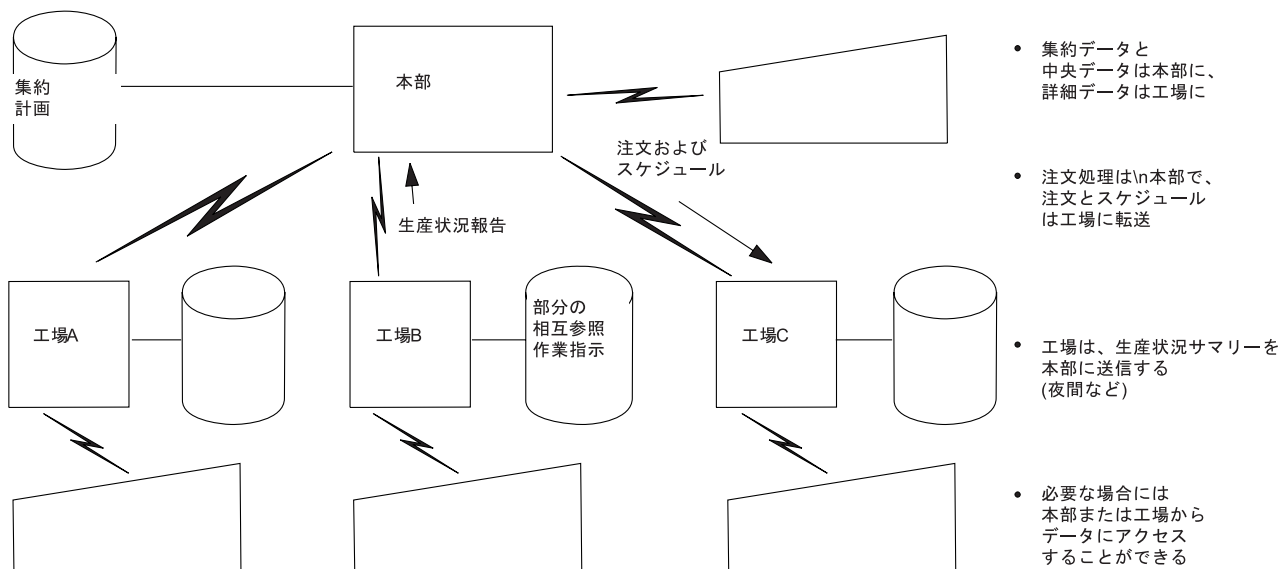
部門の接続: 分散アプリケーションと分散データ



- データベースは部門ごとに区分される
- アプリケーションは部門ごとに区分される
- すべての端末ユーザーおよびアプリケーションがすべてのシステム内のデータにアクセスできる
- 非ローカル・データへの要求は該当のシステムにルーティングされる

図1. 分散リソースの例 (パート 1)

データベースの階層的な分割



部門の接続: データとアプリケーションの階層的な分割

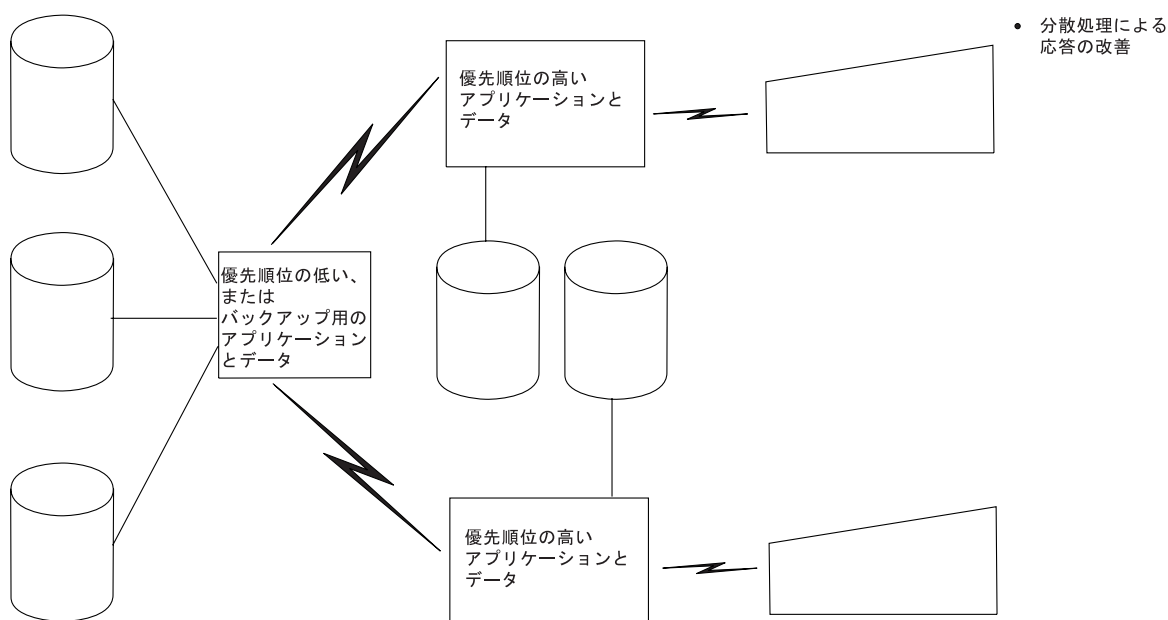


図2. 分散リソースの例 (パート 2)

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。

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な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 複数領域操作

この章には以下のトピックが含まれています。

- 『MRO の概要』
- 14 ページの『MRO を介して使用できる機能』
- 14 ページの『システム間複数領域操作 (XCF/MRO)』
- 18 ページの『複数領域操作の応用』
- 21 ページの『単一領域システムからの変換』

MRO の概要

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。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。

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

ACF/VTAM および SNA のネットワーク機能は、MRO には必要ありません。領域同士の通信を可能にする CICS 内のサポートは、**領域間通信 (IRC)** と呼ばれます。IRC は、次の 3 つの方法で実行することができます。

- 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・パッチ域 (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) によって呼び出されます。便宜上、この実装の複数領域操作 MRO(IRC) が呼び出されます。CONNECTION 定義で ACCESSMETHOD(IRC) を指定してこれを選択したためです。166 ページの『MRO のアクセス方式の選択』を参照してください。
- 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。166 ページの『MRO のアクセス方式の選択』を参照してください。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。
- IBM MVS/ESA™ のシステム間カップリング・ファシリティ (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。システム間 MRO の利点の詳細については、18 ページの『XCF/MRO の利点』を参照してください。

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、3 ページの『複数領域操作』を参照してください。MVS イメージが異なるリリースの CICS を含んでいて、そのすべてが互いに通信する MRO (またはそのシスプレックス内の他のイメージの領域と通信する SCF/MRO)

を使用する場合は、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。XCF/MRO のソフトウェアとハードウェアの要件については、122 ページの『XCF/MRO の要件』を参照してください。

CICS 複数領域操作のインストールについては、121 ページの『第 10 章 複数領域操作のインストールの注意点』で説明します。

MRO を介して使用できる機能

MRO を介して使用できる相互通信機能は次のとおりです。

- 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

これらについては、5 ページの『相互通信機能』で説明しています。

MRO での分散トランザクション処理には、ISC には適用されないいくつかの制約事項があります。

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 (**systems complex**) において、チャネル間リンク、ESCON[®] 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、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相互に通信するとき、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できるようになります。そして、次のような通常の MRO 操作がすべてサポートされます。²

- 機能シッパ
- 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

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、シスプレックスに持たせることができる CICS 領域の数を制限する場合など)、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持つことができます。複数の XCF グループを持たせたい場合は、以下を実行することをお勧めします。

- 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 グループ間のメンバーは、同じ MVS イメージ内にある場合でも、MRO (または XCF/MRO) 経由で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、3 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・バ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も構いません。単一の MVS イメージが異なるリリースの CICS を含んでいて、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 を使用する場合は、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 3.2 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、122 ページの『XCF/MRO の要件』を参照してください。

2. 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、機能シッパを使用することができます。

図3 はシスプレックス環境における XCF/MRO の使用についての単純な例です。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。

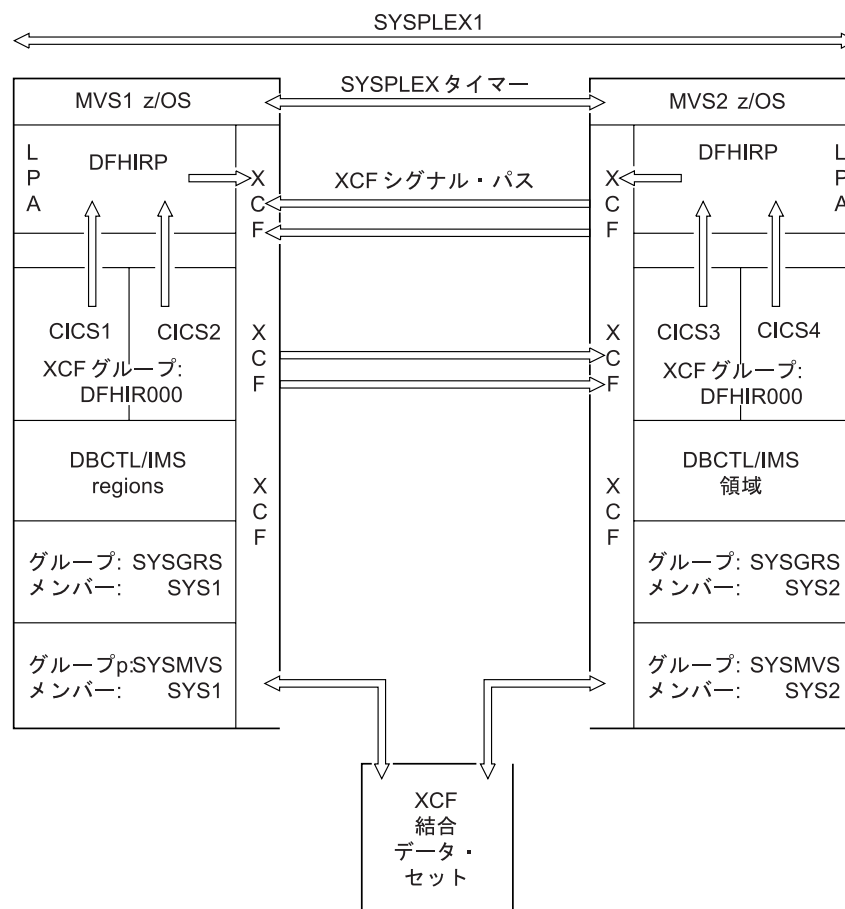


図3. 単一の CICS XCF グループを含むシスプレックス (SYSPLEX1) :

17 ページの図4 では、もう少し複雑な例を示しています。この場合、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できます。

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 以降から複数の 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 グループへの参加が必要) 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。)

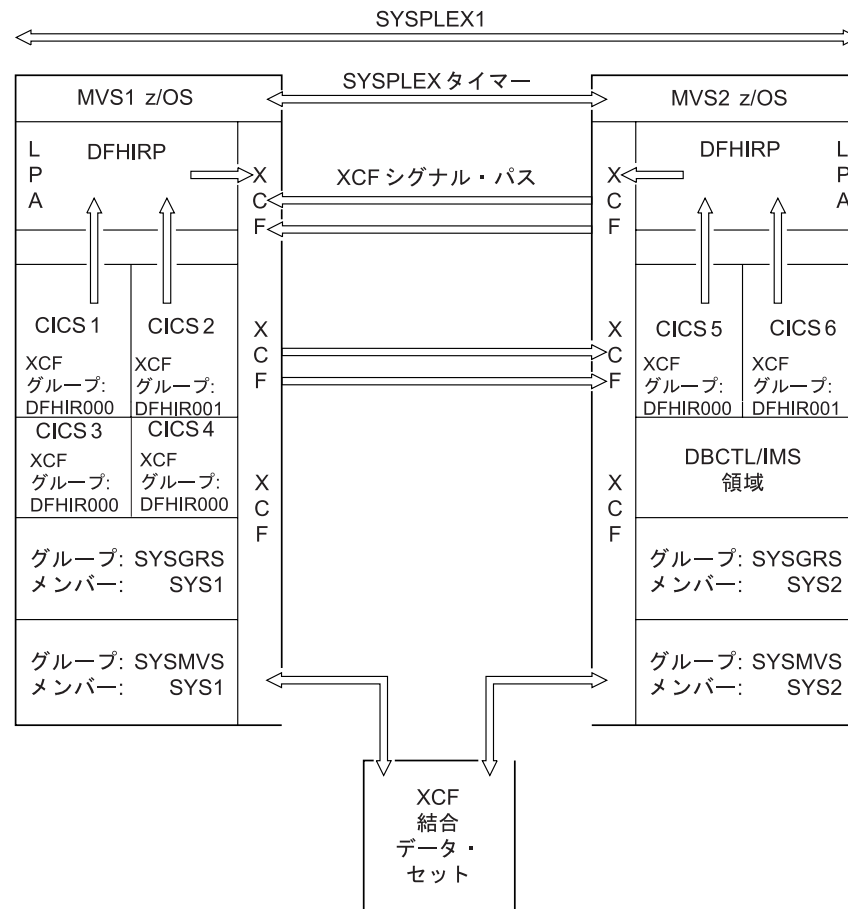


図 4. 2 つの CICS XCF グループを含むシスプレックス (SYSPLEX1)

図 4 の以下の点をご確認ください。

- 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 の利点には、次のものがあります。

- 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 へ変更する必要はなくなります (いずれにしても、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・スタートの場合だけ可能です)。
- (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。これは、例えば (偶然起こり得る) 開発領域またはテスト領域から稼働領域へのアクセスを避けたい場合には便利かもしれません。

複数領域操作の応用

このセクションでは、複数領域操作の代表的な応用について説明します。

プログラム開発

テスト用に別の CICS 領域を実行することによって、新規作成プログラムのテストを実動作業から分離することができます。これにより、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。

機能シップを使用すると、テスト・トランザクションから、ファイルや一時データ・キューなどの実動システムのリソースにアクセス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、実動システムに接続された端末を使用して、テスト・トランザクションを実行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終了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 などの計算作業に使用される場合、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。これは、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改善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 Transaction Server for z/OS, バージョン 3 リリース 2 のストレージ保護とトランザクション分離の機能を使用すれば、信頼性の低いアプリケーションのためにシステムが異常停止したり、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。データベースを所有する領域内で実行されるアプリケーションが少ないほど、この領域の信頼性は高くなります。ただし、領域間通信量は増加するため、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時間を考えれば、妥協できる場合があります。

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すると、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。

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、単一のシステム・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、65 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。

その後、次の機能を使ってワークロード・バランシングを行うことができます。

- 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れば、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 製品を使用すれば、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。DPL 要求の動的ルーティングについては、101 ページの『DPL 要求の動的ルーティング』で説明しています。動的トランザクション・ルーティングについては、67 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。CICSplex SM を使用したワークロード管理については、「CICSplex System Manager Managing Workloads」を参照してください。MVS ワークロード・マネージャーについては、「CICS パフォーマンス・ガイド」を参照してください。

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よくあります。MRO のすべての機能を使用して、エンド・ユーザーに対する単一システムのイメージを維持することができます。

注: 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 機能シップを使用するよりも DBCTL を使って、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考慮してください。

単一領域システムからの変換

既存の単一領域 CICS システムは、通常、プログラムをほとんど、またはまったく変更することなく、複数領域 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、257 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には、単一領域 CICS システム用に作成されたプログラムを修正しなければならない場合があります。

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。このような共存の考慮事項については、「*CICS Transaction Server for z/OS CICS TS V3.1*」からのマイグレーション」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、269 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。

常に 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 3 章 ISC および IPIC 相互通信機能

4 ページの『システム間通信』に示されているとおり、CICS は、以下を行うための相互通信機能を備えています。

SNA 経由のシステム間通信 (ISC over SNA)

ISC over SNA 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。これは、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されます。すべての基本 CICS 相互通信機能をサポートしています。

IP 相互接続 (IPIC)

IPIC は、CICS と、他の CICS TS for z/OS、バージョン 3.2 (以降) 領域との間、および CICS TS と Java クライアントの間に使用することができます。CICS 間リンクでは、基本 CICS 相互通信機能の中で現在サポートされているものは DPL のみです。

この章には以下のトピックが含まれています。

- 『SNA 経由のシステム間通信』
- 28 ページの『IP 相互接続』

SNA 経由のシステム間通信

この章では、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。

- 『サブシステム間の接続』
- 25 ページの『システム間セッション』
- 28 ページの『システム間セッションの確立』

サブシステム間の接続

この項では、システム間通信にサブシステムを接続する方法を簡単に説明します。

次の 3 つの基本形式が考えられます。

- 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

24 ページの図 5 に可能な構成を示します。

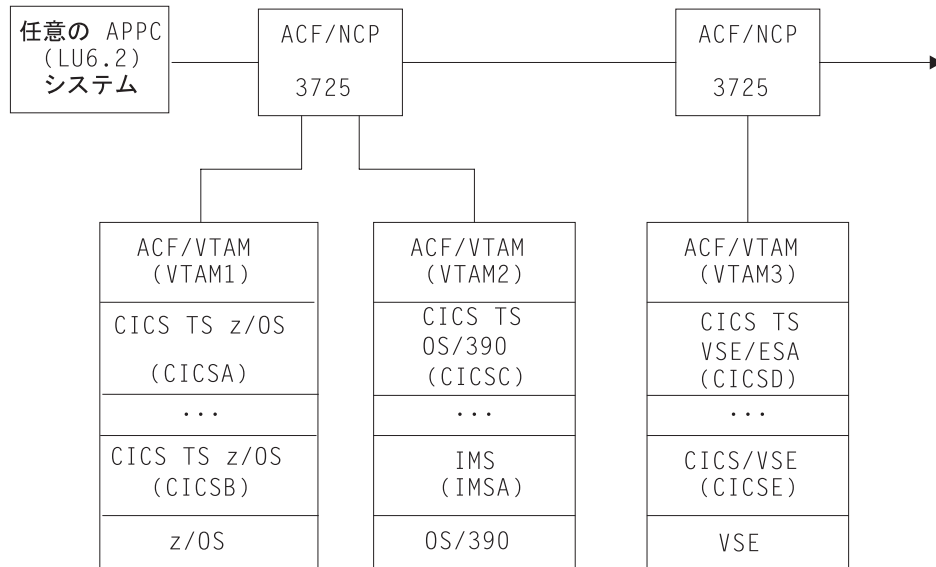


図5. 相互通信するシステムの可能な構成

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5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5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

これは、システム間通信の最も典型的な構成です。例えば、図5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。

ISC を使って CICS Transaction Server for z/OS が接続できる CICS および 非 CICS のシステムのリストについては、4 ページの『システム間通信』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実際の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 (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。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。

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 間通信で使うことができます。(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。

次の段落では、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

マップされる会話では、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・データだけです。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。

これらの 3 つのレベルは、EXEC CICS コマンドと CPI コミュニケーションの両方でサポートされます。

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りモート・トランザクションを開始する際、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

複数セッション APPC 接続では、**LU サービス・マネージャー**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」の選択に対する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。これにより、ユーザーは次の機能を使用することができます。

- 代替ルーティング - 指定の COS に対する仮想経路を、異なる物理パス (明示経路) に割り当てることができます。
- 各種通信量 - 異なる種類の通信量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介入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れるために、異なる物理リンクを使用することができます。また、セッションの優先順位は、それらが伝送する通信量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**と呼ばれるグループで指定され、各モードセットには**モード名**が割り当てられています。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。(126 ページの『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たびに、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、他のタスクがその使用を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複数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。これにより、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 のことを言います。

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は、システム・プログラマーが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。(174 ページの『APPC リンクの定義』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、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。

セッションの開始は、次のいずれかの方法で行うことができます。

- 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。資料については、161 ページの『第 13 章 リモート・システムへのリンクの定義』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。

IP 相互接続

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。

システム初期設定パラメーター

IP 相互接続では CICS TCP/IP サービスを活動化する必要があります。CICS 開始時にこれらを活動化するには、システム初期設定パラメーターとして TCPIP=YES を指定します。（TCPIP パラメーターのデフォルト値は NO です。）

TCPIP システム初期設定パラメーターの参照情報については、「*CICS System Definition Guide*」の TCPIP をご覧ください。

システム間セッションの確立

2 つの CICS 領域間の IPICによる接続を定義する方法は、169 ページの『IP 相互接続リンクの定義』に説明されています。

第 4 章 CICS 機能シップ

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』
- 32 ページの『設計上の考慮事項』
- 35 ページの『ミラー・トランザクションと変換プログラム』
- 39 ページの『機能シップの例』

機能シップの概要

CICS 機能シップを使用すると、CICS アプリケーション・プログラムで次のことが可能になります。

- 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、43 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。そ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域コマンドなどの機能を全く同じ方法で使用します。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、32 ページの図 6 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。CICS は、ファイル管理テーブルから、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を発見します。CICS は、READ 要求を適切な伝送形式に変更してから、それを CICB にシップして実行させます。

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の機能によって、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必ず同期点をとります。これによって、リモート・システムとローカル・システムの保護リソース

の変更が一貫したものになります。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。

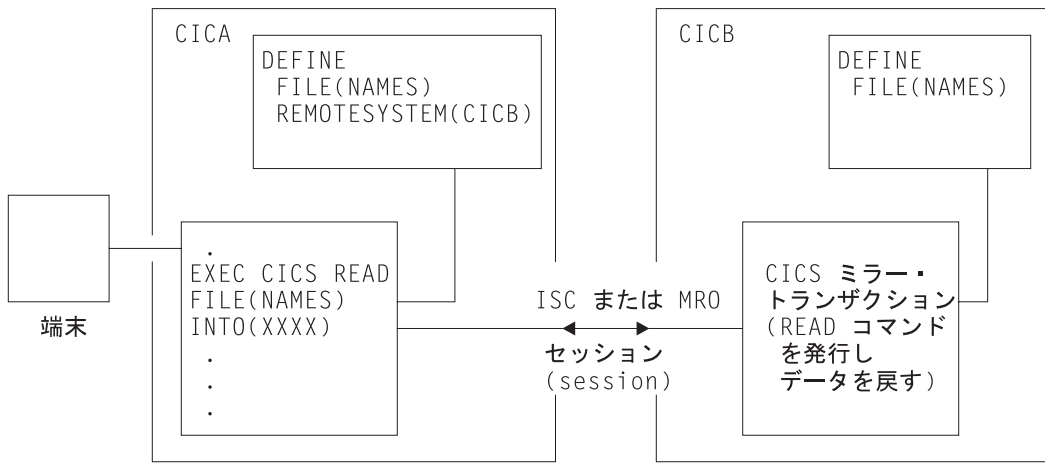


図 6. 機能シッブ

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。(詳細は、213 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。この機能は、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。

プログラムの独立性を損なうおそれがありますが、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、機能シッブ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、要求は指定の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、ローカル・リソースにアクセスするかリモート・リソースにアクセスするかの決定を実行時に行うことができます。

ファイル制御

機能シッブを使用すると、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。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、ファイルは、それが存在する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 トランザクションは、リモートの CICS OS/390 システムに関連付けられた IMS/ESA DM および IMS/VS DB データベース、あるいはリモートの CICS/VSE システムに関連付けられた DL/I DOS/VS 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 3 リリース 2 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/ESA DM (DL/I) 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 システム側からは、このデータベースは単にリモートです。

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/ESA 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、異なるシステムでのリンクされた DL/I 更新をリンクさせることは、単一の作業単位ではできません。

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・プログラムは、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、そのキュー名 (またはリモート名) を、リモート・システムの TST に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もつものとして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・システム環境でも使用することができます。例えば、システムで一晩中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されなければなりません。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れば、プログラムは、ローカル・システムとリモート・システム両方の 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競合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と、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。

パフォーマンスの悪化の症状は、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 が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れば、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セッションが使用できるの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。競合勝者セッションがすぐには使用できないと、XZIQUE または XISCONA 出口プログラムが呼び出されます。これらの出口プログラムは、CICS によって、その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 `../com.ibm.cics.ts.performance.doc/topics/dfht3_stats_intro.dita` を参照してください。

注: 出口には、XISCONA よりも XZIQUE の使用をお勧めします。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ため、両方の出口を使用可能にすることはお勧めしません。

XISCONA 出口プログラムがすでにある場合は、これを修正することによって XZIQUE 出口点で 사용할 ことができる場合があります。

システム間キューの制御については、さらに 297 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。そのうちいくつかは、「体系化プロセス」(246 ページの『体系化プロセス』を参照) に対応します。提供されるミラー・トランザクションの詳細については、243 ページの『第 17 章 ローカル・リソースの定義』で説明します。本書では、これらを一般にミラー・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に若干の違いがあり、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、37 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の機能を使用して、要求側システムと通信を行います。

要求側システム (図 7 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) がこの処理を示します)。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*」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の機能を使用して、ミラー・トランザクションに要求を送信します。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終了したかどうかを追跡され、必要であればそれを再び呼び出します。

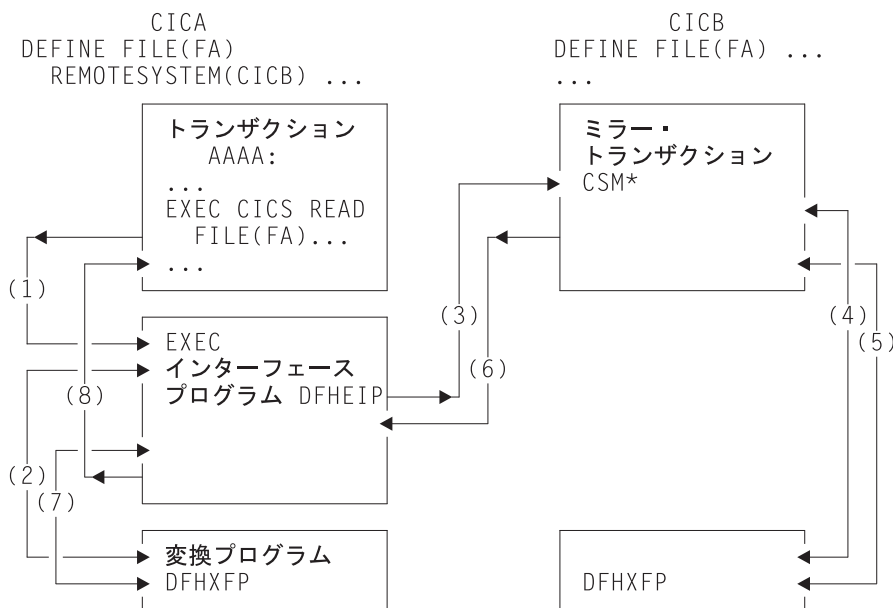


図 7. 機能シップにおける変換プログラムとミラー

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 7 の (4))。そして、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

の応答はデコードされ、再度変換プログラムにかけられ (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、応答の送信後に終了します。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終了しません。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終了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。CICS は、DL/I 要求を他のシステムから受け取ると、アプリケーション・プログラムによって出されて、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を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・トランザクションがあれば、それと同期点メッセージを交換します。これを**多重ミラー状態**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、指定の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。このことを**チェーン・ミラー状態**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、照会のみ要求の場合は除きます。

MRO 機能シップ

MRO 機能シップでは、ミラー・トランザクションの動作が、この前の項の説明と若干異なります。

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（36 ページの『ISC 機能シップ』を参照）。これは、活動タスクの数をできるだけ少なくするとともに、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、アプリケーションの中には、データ保全性のためには不要であっても、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）によって終了します。

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。

『機能シップの例』 39 ページの『機能シップの例』の図 40 ページの図 9と図 40 ページの図 10 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、要求が、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。

注: 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** を指定してはなりません。これは、未使用の場合、SEND セッションが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を最適化するように設計されています。この最適化は、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 形式を使用することによって行われます。

CICS は、短パス変換プログラム (DFHXFX) をファイル制御、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のに使用します。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異常終了処理

リモート領域のミラー・トランザクションに障害が起こった場合、機能シップ要求を発行したアプリケーション・プログラムは、以下の両方に当てはまる場合にのみミラーの異常終了を処理して自らのローカル・リソースをコミットすることができます。

1. アプリケーション・プログラムが、ミラーの障害により発生した異常終了を明示的に処理し、かつ次の 2 つのいずれかを行う。
 - アプリケーション・プログラムが、通常のトランザクション終了による暗黙的な同期点をとるか、または
 - アプリケーション・プログラムが、明示的な同期点要求を発行する。
2. リモート・ミラー・トランザクションが、アプリケーション・プログラムの作業単位の有効範囲内でリカバリー可能な作業を実行しない。つまり、ミラーが以下のいずれか 1 つに対してのみ起動されている。
 - 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)

その他の場合、つまり、アプリケーション・プログラムが異常終了を処理しない、またはミラーがどのようなリカバリー可能な作業でも実施する (例えば、リカバリー不能ファイルであってもファイル更新する) 場合は、CICS はトランザクションのバックアウトを強制します。

機能シップの例

この項では、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す例をいくつか示します。

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。

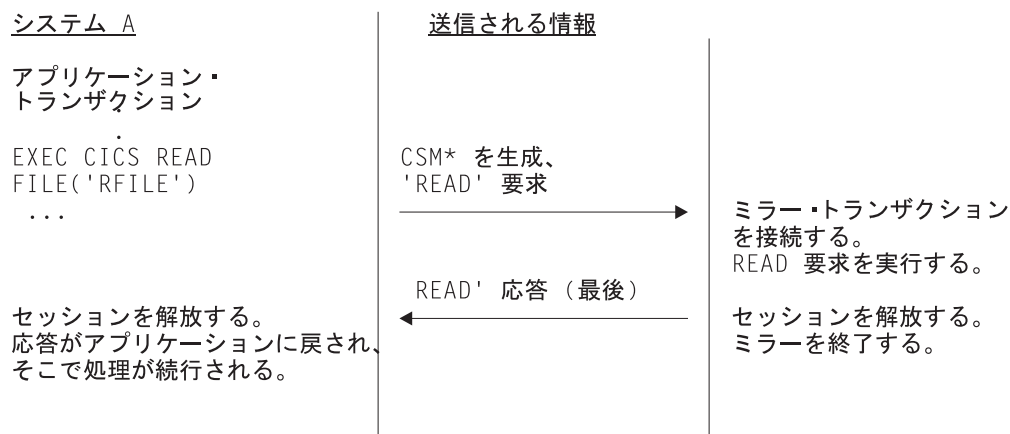


図 8. ISC 機能シップ - 単純照会：ここでは、リソースは変更されません。セッションは解放されて、ミラー・タスクは即座に終了します。

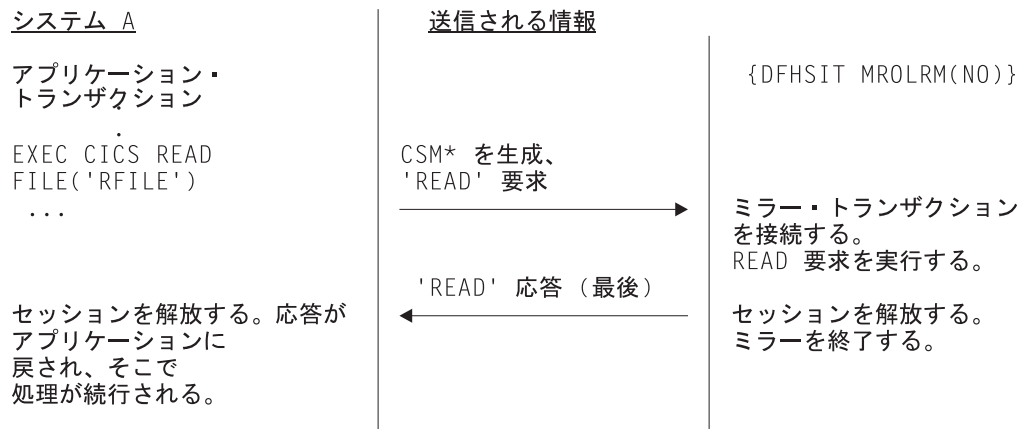


図 9. MRO 機能シップ - 単純照会：ここでは、リソースは変更されません。長期実行ミラー・タスクは指定されていないため、セッションはシステム B によって解放されて、ミラー・タスクはすぐに終了します。

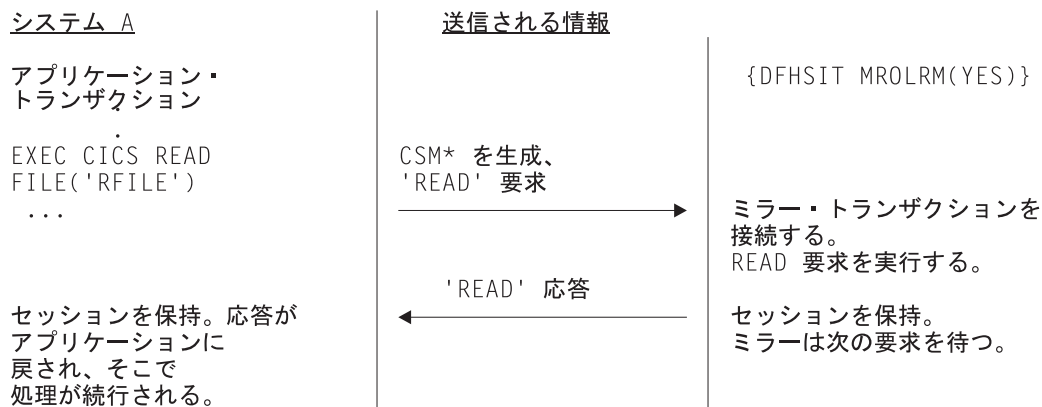


図 10. MRO 機能シップ - 単純照会：ここでは、リソースは変更されません。ただし、長期実行ミラー・タスクが指定されているため、セッションはシステム B によって保持されて、ミラー・タスクは次の要求まで待機します。

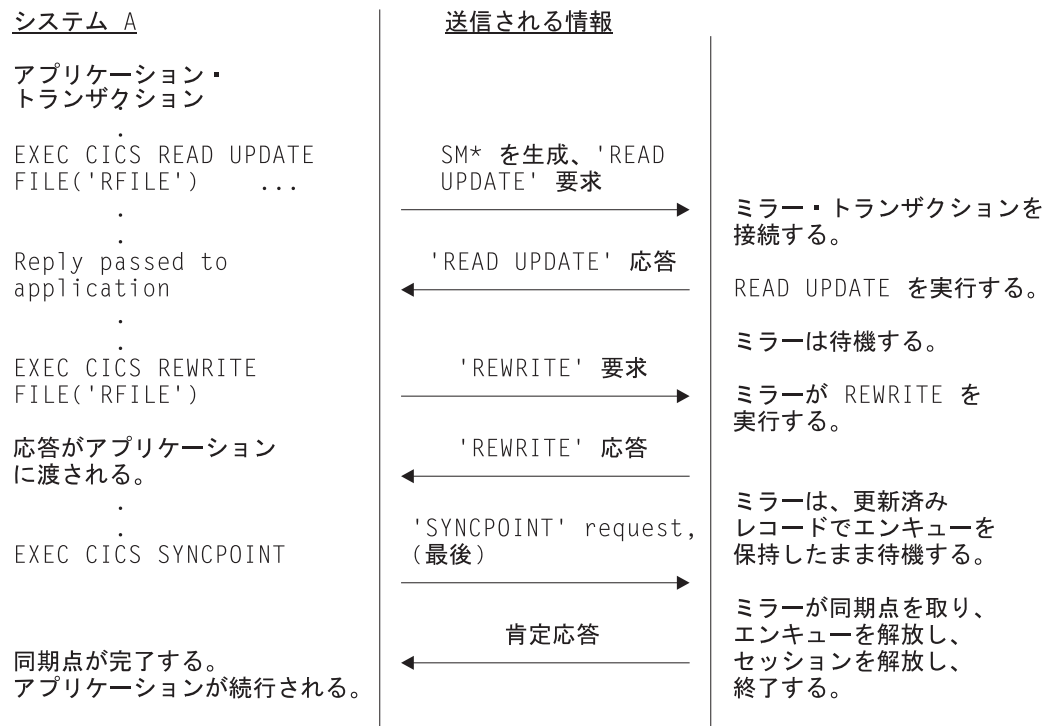


図 11. ISC または MRO 機能シップ - 更新：ミラーは、REWRITE まで待機しなければならないため長期実行タスクとなり、SYNCPOINT を受け取るまで終了しません。ファイルがリカバリー可能でない限り、更新されたレコードについての待機は、REWRITE コマンドが終わった後は保持されないことに注意してください。

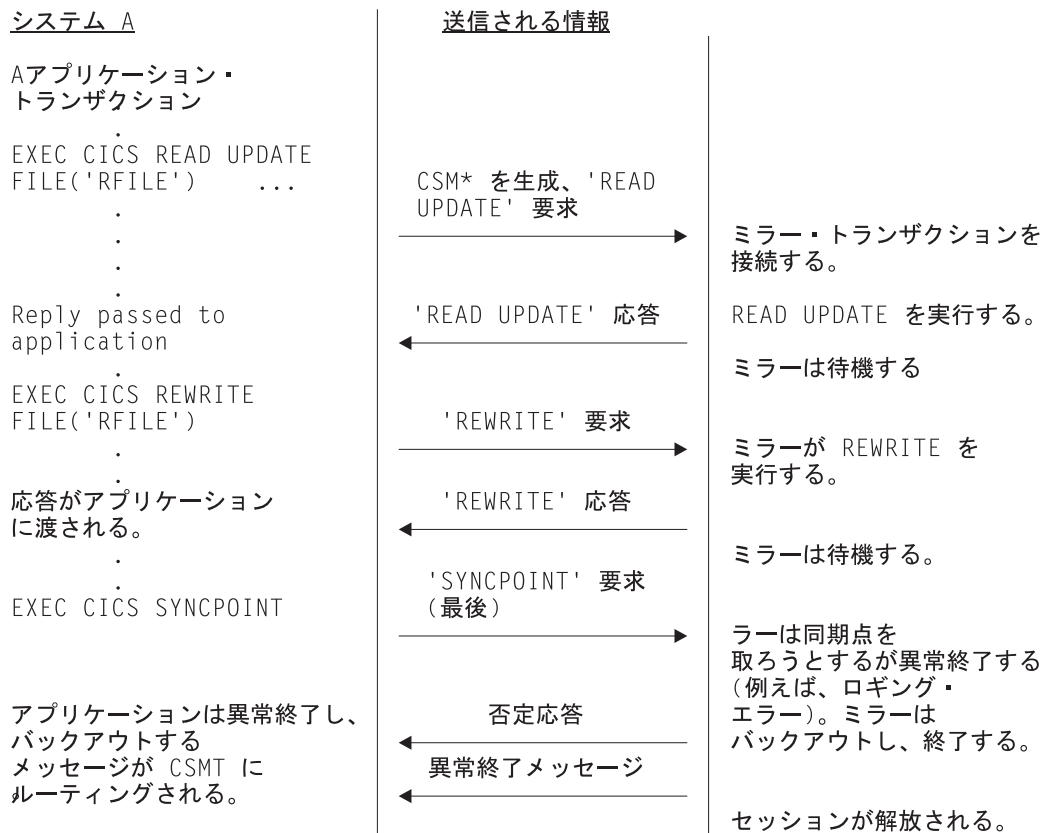


図 12. ISC または MRO 機能シップ - 更新 (ABEND)

図 12 は、41 ページの図 11 に似ていますが、異常終了が同期点処理中に起こる点が異なります。

第 5 章 非同期処理

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』
- 44 ページの『非同期処理方式』
- 45 ページの『START と RETRIEVE コマンドを使用する非同期処理』
- 51 ページの『システム・プログラミングに関する考慮事項』
- 52 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 13 に示します。

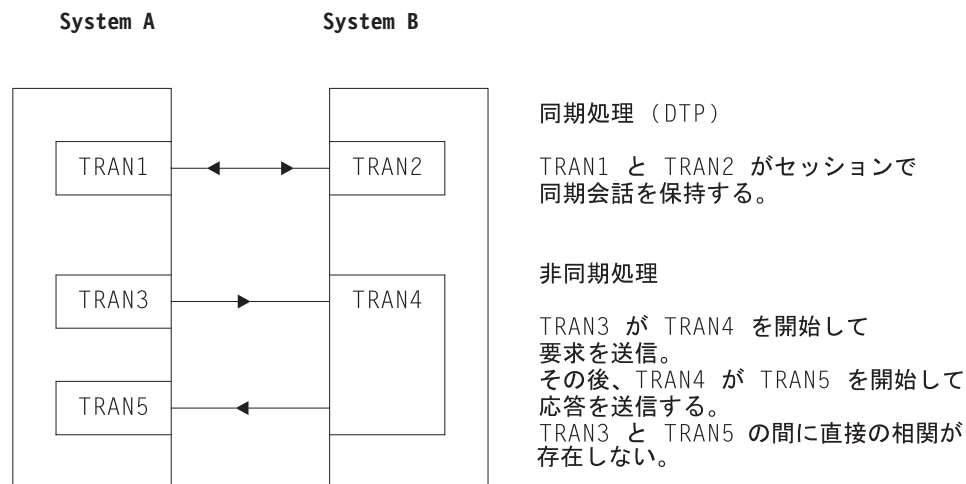


図 13. 同期処理と非同期処理の比較

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・データベースに対する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 における非同期処理は、次の 2 つの方法のどちらかで行うことができます。

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チャーは、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、273 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、109 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

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『インターバル制御機能 (Interval control)』を参照してください。非同期処理を行うために使用できるインターバル制御コマンドは、次のとおりです。

- START
- CANCEL
- RETRIEVE

リモート・トランザクションの開始と取り消し

注:

動的に転送される START コマンドの取り消しについては、88 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。

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。

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」すべてのタスクが出すことができます。

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。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 RTRANSID 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、70 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、72 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、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。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・プロセス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。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、54 ページの図 15 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 (49 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れば、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。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 要求の据え置き送信

NOCHECK オプションを指定した START コマンドでは、PROTECT が指定されているかどうかに関係なく、CICS は、環境に応じて、リモート・システムに対する要求の伝送を据え置く場合があります。

MRO リnkの場合、NOCHECK を指定した START 要求は据え置かれません。

ISC リnk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 (または機能シッブ要求) 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終了する (暗黙同期点)。

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 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。

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。

一連の要求は 1 つの SNA ブラケット内で伝送され、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。

IMS では、START 要求とその後に続く同期点の間に、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。

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ブされた 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、34 ページの『システム間のキューイング』のページ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、297 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、そこに対して START 要求を機能シッブすると、通常、SYSIDERR 条件がアプリケーションに戻されます。これは、リモート・システムへの接続が確立されていても、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をキューに入れられることになっていない場合にも起こります。しかし、リモート・システムがこの CICS 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れば、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。こうするには、2 つの方法があります。

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*」の『TRANSACTION 定義属性』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を使用する。XISLCLQ は、次の場合に、機能シッパされた 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。

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。

XZIQUE、XISCONA、XISLCLQ 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『システム間通信プログラム 出口 XISCONA および XISLCLQ』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すで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たらします。

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・システムにシッパ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れば、リモート・システムから次の開始要求が到着するまで、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、79 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 (TERMINAL) 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ると、CICS は、その端末を、トランザクションがその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 TERMINAL 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。

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 (255 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

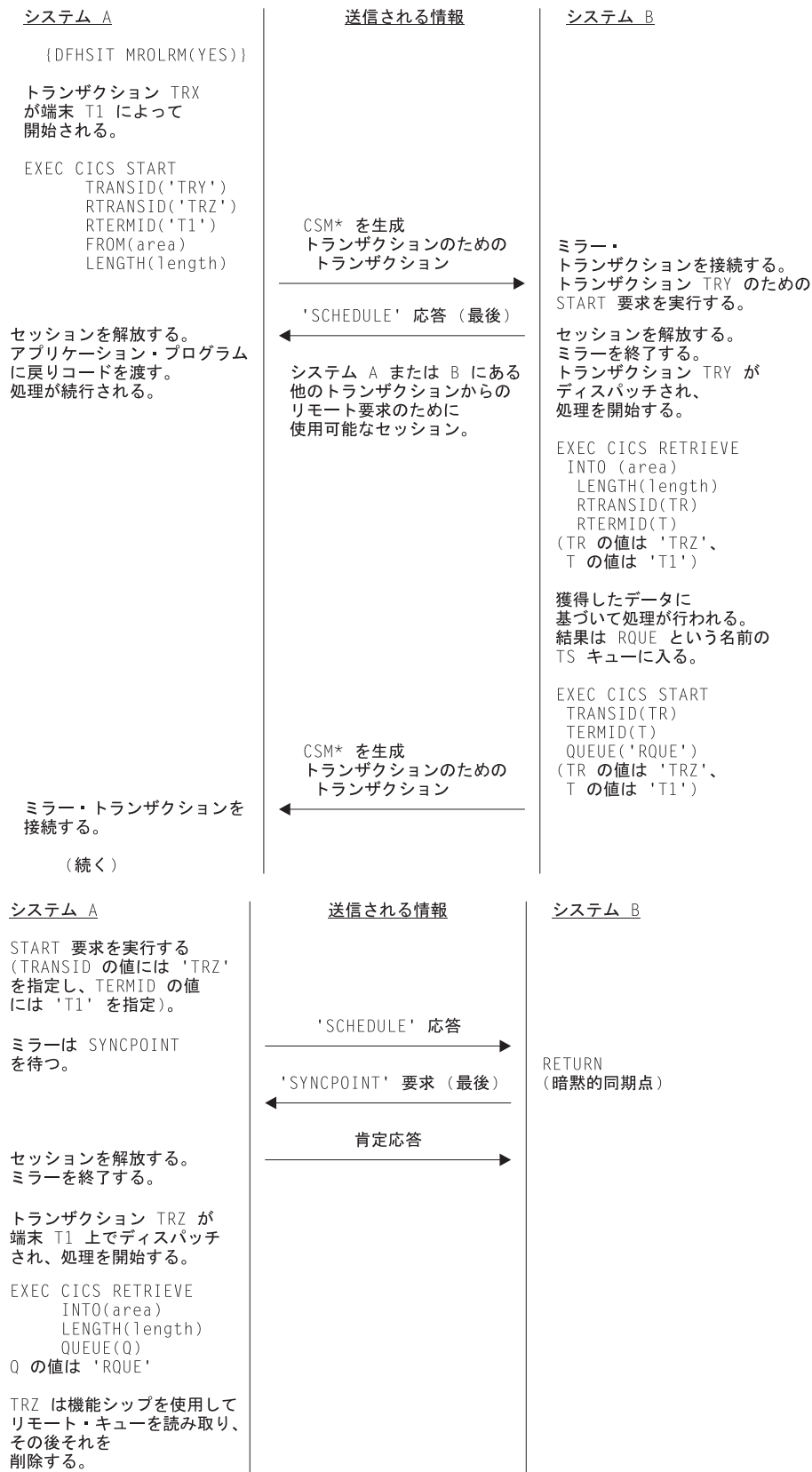


図 14. 非同期処理 - リモート・トランザクションの開始： この例では、長期実行ミラー (MROLRM) がシステム A に対しては指定されているが、システム B に対しては指定されていない MRO 接続を示します。2 つのシステムでのミラー・トランザクションのアクションの違いに注意してください。

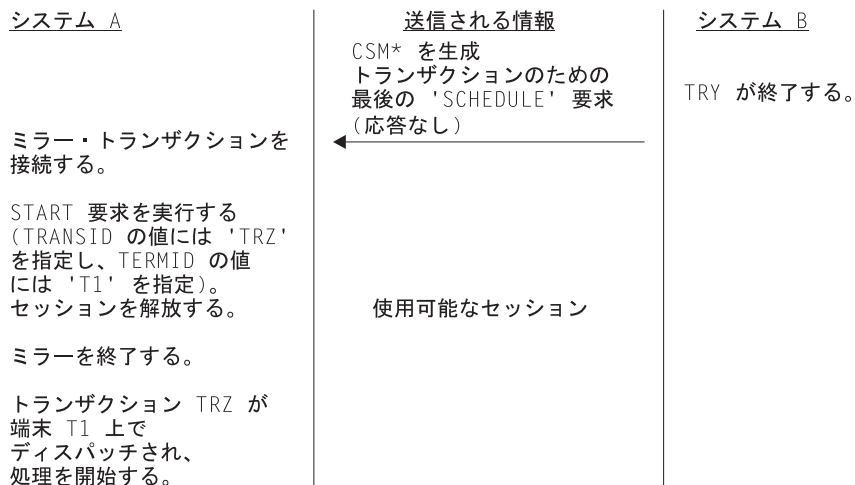
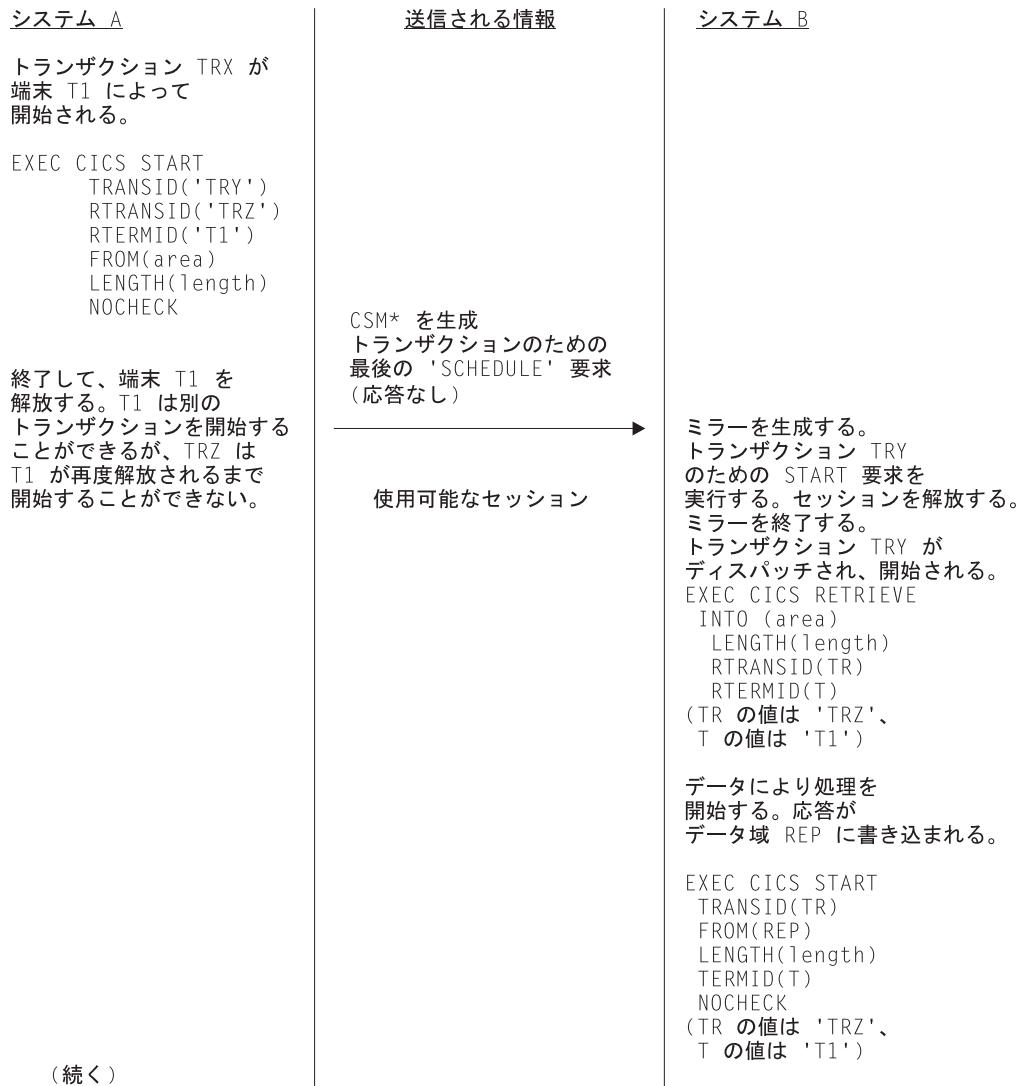


図 15. 非同期処理 - NOCHECK を使用したリモート・トランザクションの開始： この例は、ISC 接続、または長期実行ミラーのない MRO 接続を示しています。

54 ページの図 15 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、65 ページの『第 7 章 CICS トランザクション・ルーティング』と 97 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 (Overview of BTS)』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。

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 (例えば、リモート (CICS 以外) IIOP クライアント) 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 (例えば、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) 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、79 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 16 は「ハブ」ルーティング・モデルです。

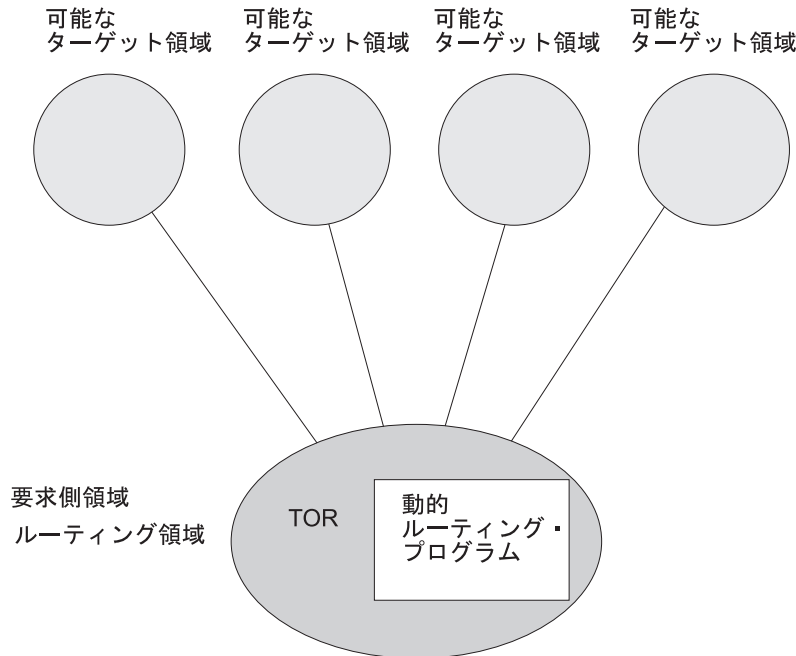


図 16. 「ハブ」ルーティング・モデルを使用した動的ルーティング：1 つのルーティング領域 (TOR) が、複数のターゲット領域の間で選択します。

「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 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 内ではしか実行されません。

「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。図 17 は、分散ルーティング・モデルです。

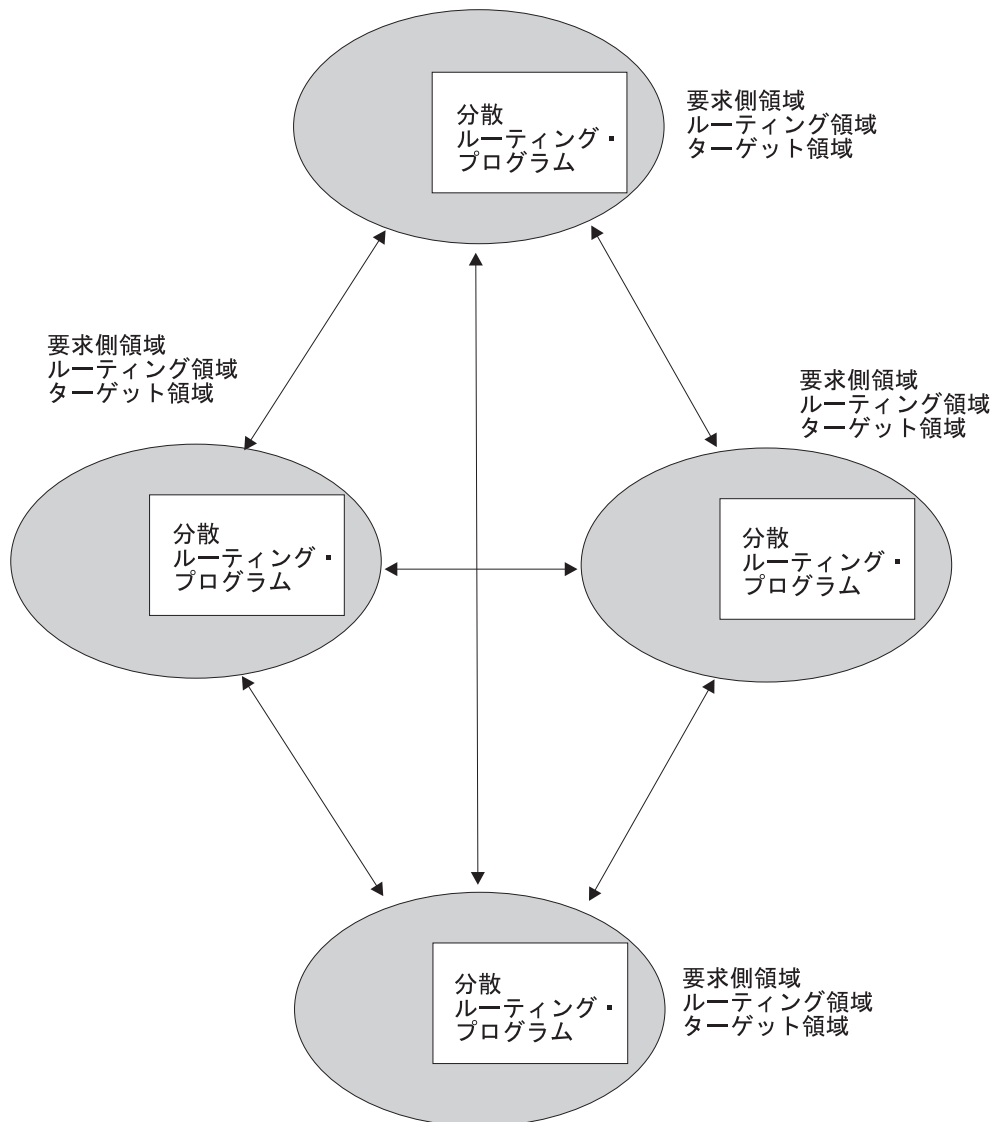


図 17. 分散ルーティング・モデルを使用した動的ルーティング：各領域がルーティング領域とターゲット領域の両方になることがあります。

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となります。

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すルーティング・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 3 リリース 2 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

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』
- ・ 67 ページの『端末開始トランザクション・ルーティング』
- ・ 70 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』
- ・ 79 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』
- ・ 89 ページの『リモート APPC 接続の割り振り』
- ・ 92 ページの『中継プログラム』
- ・ 93 ページの『基本マッピング・サポート (BMS)』
- ・ 94 ページの『ルーティング・トランザクション (CRTE) の使用』
- ・ 95 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。つまり、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 18 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。

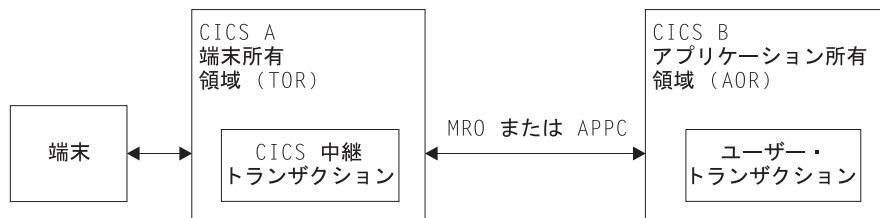


図 18. トランザクション・ルーティングの要素

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、MRO リンクまたは APPC リンクによって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、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 セッションなどを総称して使用します。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 (BMS の詳細については、93 ページの『基本マッピング・サポート (BMS)』で説明します)。

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く)。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・ルーティングで使用するトランザクションの作成方法の詳細については、269 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』を参照してください。

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。
 - 70 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』
 - 79 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。資料については、94 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。資料については、234 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 (59 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 (Specifying CICS system initialization parameters)』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、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。資料については、239 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이지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は、ルーティング・プログラムは、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。79 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す。
- 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、101 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できます。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もの。何も指定されていないと、ローカル・システムの sysid が渡されます。

注: 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は、単一の共通定義を使用することをお勧めします。資料については、239 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。何も指定されていないと、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 に対応します。
- ターゲット・トランザクションの連絡域のアドレス。
- 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・ルーティングによって、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もできます。

- ワークロード・バランシングする。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・システムに対し使用できるセッションがない場合、要求をキューイングするかどうかを規定する。(システム間キューの長さの制御については、297 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 リンクの場合のみ、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か、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ができますが、EXEC CICS RECEIVE を出しても、ルーティング先のトランザクションが初期端末データ入手することはできません。

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) 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 3 リリース 2 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、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。

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 (Affinity)』を参照してください。

CICSplex SM の使用

通常、動的ランザクション・ルーティングを利用するには、動的ランザクション・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、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ランザクションの従来のルーティング

このセクションでは、自動ランザクション開始 (ATI) によって開始されたルーティング・ランザクションの「従来」の方法について説明します。

重要:

可能な限り、79 ページの『START コマンドで呼び出された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、このセクションの「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、72 ページの図 19 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 要求を出すとしてします。

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い限りなりません。それが見つかないと、要求は失敗します。³

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。³

3. ここでは、START コマンドによって開始されたトランザクションの「従来」のルーティングについて説明しています。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、79 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。

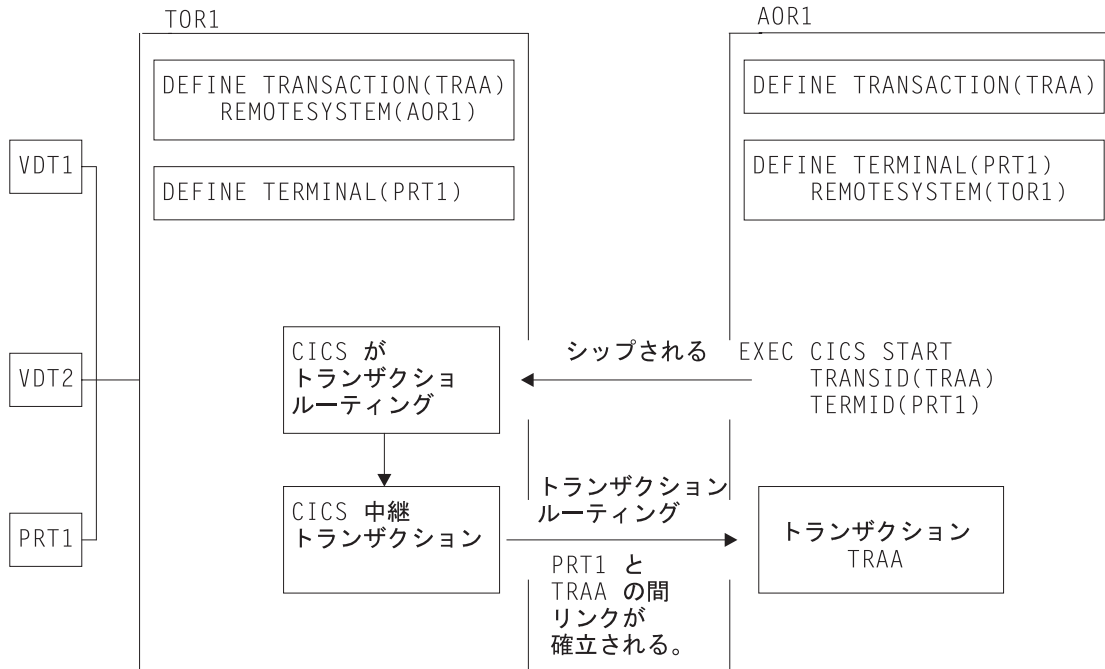


図 19. ATI によって開始されたトランザクション・ルーティング

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パ

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 要求を機能シッパする。
2. CICA が、CICB の所有する一時データ・キューに対する WRITEQ 要求を機能シッパして、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

重要:

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。79 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 が必ず正しく作動するように、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することができます。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(227 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。

74 ページの図 20 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。

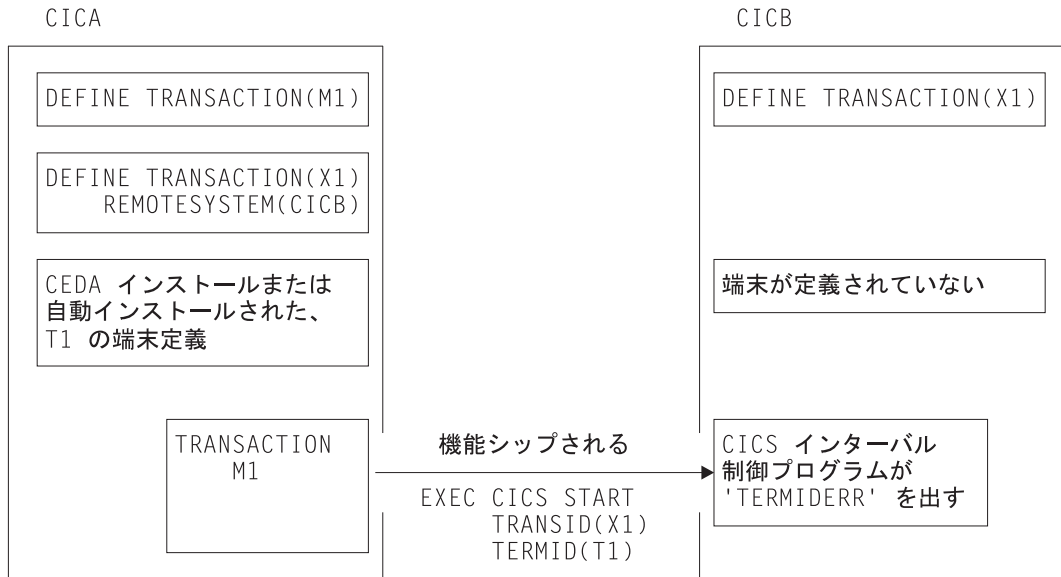


図 20. 端末 ID が認識されていないシステムでの ATI 要求の失敗

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ったと実行されます。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ブされたかどうか。
- 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ブされた場合は、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、74 ページの図 20 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。

76 ページの図 21 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。

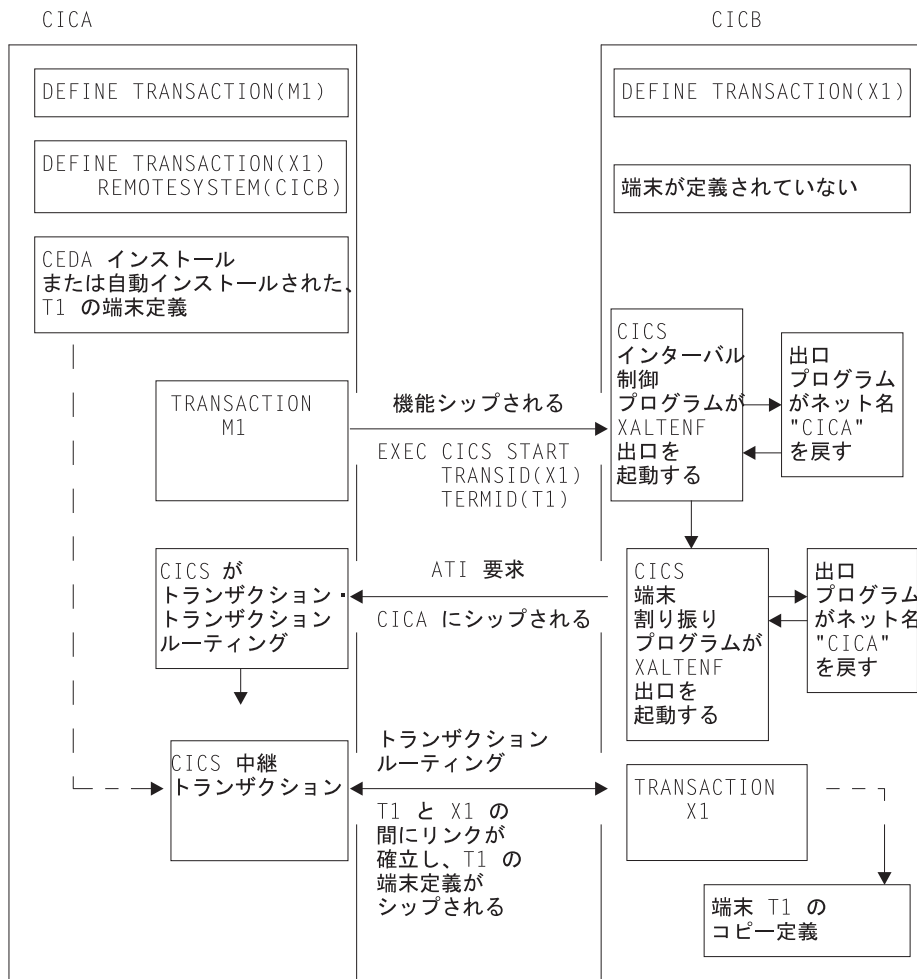


図 21. START 要求における「端末未認識」状態の解決

図 21 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 (Writing initialization and shutdown programs)』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 ('Terminal not known' condition exits XALTENF and 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 CICSTS32.CICS.SDFHSAMP の DFHXTENF メンバーに用意されています。

複数 TOR の ATI 用端末のシッパ

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、シッパ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 AORI 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 TI に対して実行されます。一方、領域 TORA の端末 TI がすでに AORI にトランザクション・ルーティングしており、TI の定義がすでに TORA から AORI ヘシッパされています。その START 要求は、AORI に到達すると、端末 TI 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 ヘシッパされます。

78 ページの図 22 にこの様子を示します。

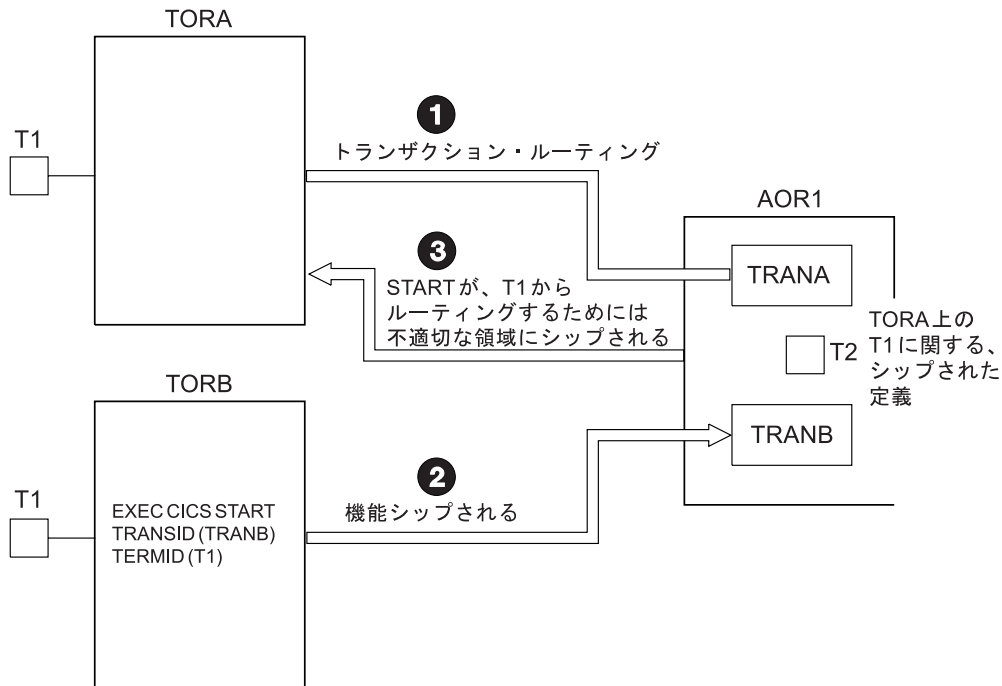


図 22. 正しくない端末に対して開始される機能シップされた *START* 要求： 端末 *T1* (*TORA* によって所有される) のシップされた定義は *AOR1* にインストールされていますので、*TORB* から受信された *START* 要求は、ルーティングに対し、*TORB* ではなく *TORA* へシップされます。

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。

79 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、すで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、233 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* にすで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 ヘシップされ、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、すで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、150 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

この項では、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の推奨方法について説明します。便宜上、この項で説明する方法のことを拡張方法と呼ぶことにします。この拡張方法は、「従来」の方法 (70 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) の代わりとなるものです。ただし、拡張方法では、次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。

- 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。
- 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。

これらの場合は、「従来」の方法を使用してください。

トランザクションが EXEC CICS START コマンドで呼び出す場合に、そのトランザクションをこの項で説明する拡張方法でルーティングすることを指定するには、トランザクションを要求側領域で ROUTABLE(YES) として定義します (START コマンドが発行された領域)。

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しか呼び出されません。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 に機能シッパされます。要求は、TOR からルーティングするために、再度シッパされます。

拡張方法では、AOR へのシッパと TOR への再シッパの 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 に機能シッパされたときに、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 に機能シッパされないで、「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 から定義がシッパされます。

端末関連の START コマンド

端末関連の START コマンドによって呼び出されたトランザクションが、拡張転送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。

- 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、次の条件すべてを満たしている。
 - 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が DPL によりリンクされていた場合などを除きます。この場合、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、トランザクションを開始す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。(トランザクションを開始す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、TOR、およびターゲット領域はすべて CICS Transaction Server for OS/390、バージョン 1 リリース 3 以降である。

注: 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 は (異なる領域の場合)、次のいずれかで接続されている。
 - MRO リンク
 - APPC 並列セッション・リンク

- TOR とターゲット領域は、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、リンクを介してすでに行われている(端末開始トランザクションのルーティングにより、TOR は、ターゲット領域が CICS Transaction Server for OS/390、バージョン 1 リリース 3 以降のシステムであるかどうか、つまり拡張ルーティングに適格であるかどうかを判別できる)。
 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場合に候補とする START 開始トランザクションを考える場合には、 START コマンドで次のオプションを指定しているかどうか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

START コマンド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、トランザクション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)。

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、TERMINID 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。

静的ルーティング:

AOR 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。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 REMOTESYSYEM 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は、トランザクションは TOR 内でローカルに実行されます。

注: 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「従来」の方法で処理されます (70 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 23 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。

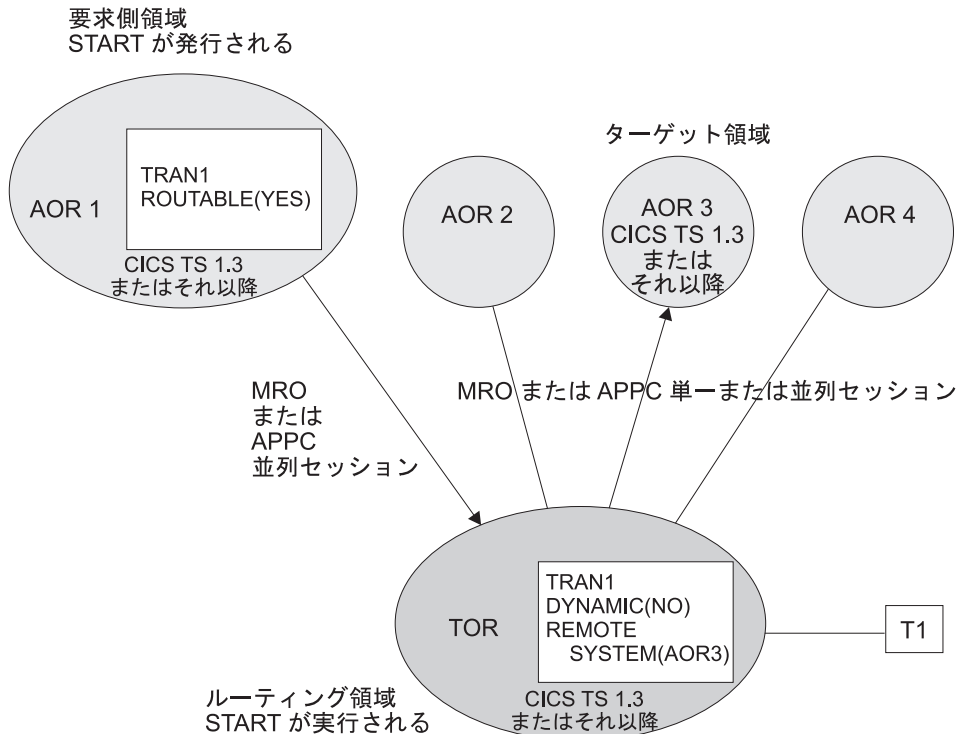


図 23. 拡張方法を使用した、AOR で発行された端末関連 START コマンドの静的ルーティング：要求側領域、TOR、およびターゲット領域はすべて CICS TS OS/390 バージョン 1.3 以降です。要求側領域と TOR は、MRO または APPC 並列セッション・リンクで接続されています。TOR とターゲット領域は、MRO または APPC (単一または並列セッション) リンクで接続されています。要求側領域のトランザクション定義では、ROUTABLE(YES) が指定されています。TOR のトランザクション定義では、DYNAMIC(NO) が指定されています。REMOTESYSTEM 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。

動的ルーティング:

動的ルーティング・モデル:

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 (59 ページの『「ハブ」モデル』を参照)。

AOR 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。動的ルーティング・プログラムは TOR で呼び出されます。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。

注: 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しか呼び出されません。トランザクションを転送することはできません。この場合、トランザクションは「従来」の方法で処理されます。つまり、要求元である AOR で実行されるように、その AOR にルーティングされます。

図 24 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。

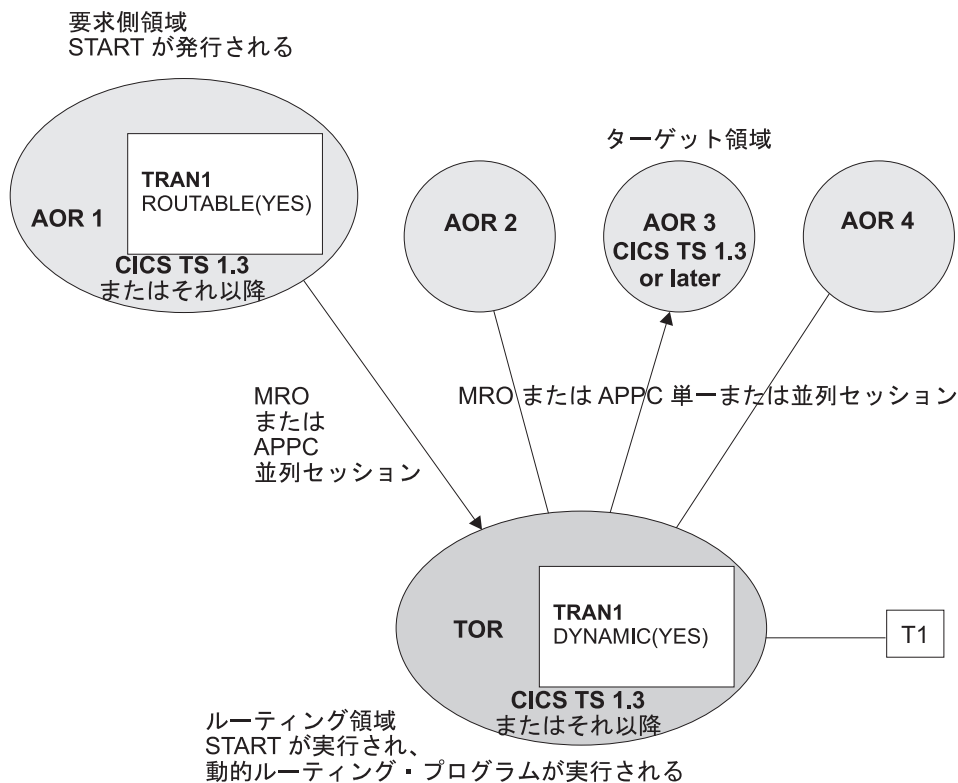


図 24. AOR で発行された端末関連 START コマンドの動的ルーティング： 要求側領域、TOR、およびターゲット領域はすべて CICS Transaction Server for OS/390、バージョン 1 リリース 3 以降です。要求側領域と TOR は、MRO または APPC 並列セッション・リンクで接続されています。TOR とターゲット領域は、MRO または APPC (単一または並列セッション) リンクで接続されています。要求側領域のトランザクション定義では、ROUTABLE(YES) が指定されています。TOR のトランザクション定義では、DYNAMIC(YES) が指定されています。

TOR で発行された START コマンド

静的ルーティング: TOR 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。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である場合 (拡張ルーティングを可能にするための START コマンドの条件リストを参照)、以下のようになります。

1. START は 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。

注: 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「従来」の方法で処理されます (70 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 に機能シフトされます。 REMOTESYSTEM が指定されていない場合は、START は TOR 内でローカルに実行されます。

図 25 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。

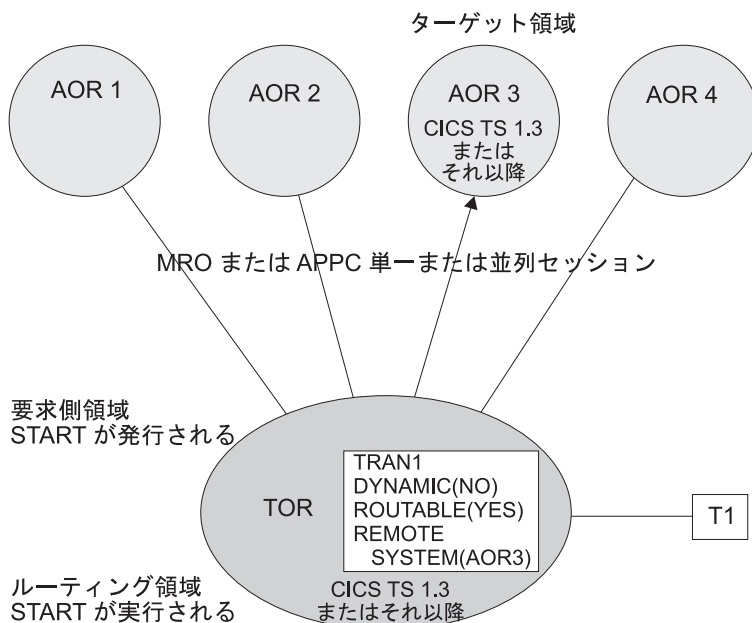


図 25. 拡張方法による、TOR で発行された端末関連 START コマンドの静的ルーティング： TOR とターゲット領域は、ともに CICS Transaction Server for OS/390、バージョン 1 リリース 3 以降です。TOR とターゲット領域は、MRO または APPC (単一または並列セッション) リンクで接続されています。TOR のトランザクション定義では、DYNAMIC(NO) と ROUTABLE(YES) が指定されています。 REMOTESYSTEM オプションでは、トランザクションのルーティング先の AOR が指定されています。

動的ルーティング:

動的ルーティング・モデル:

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 (59 ページの『「ハブ」モデル』を参照)。

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。動的ルーティング・プログラムが呼び出されます。トランザクションが拡張ルーティング可能である場合には、START は TOR で実行され、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することができます。

注: 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しか呼び出されません。トランザクションを送信することはできません。START 要求は「従来」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は、START は TOR 内でローカルに実行されます。

図 26 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。

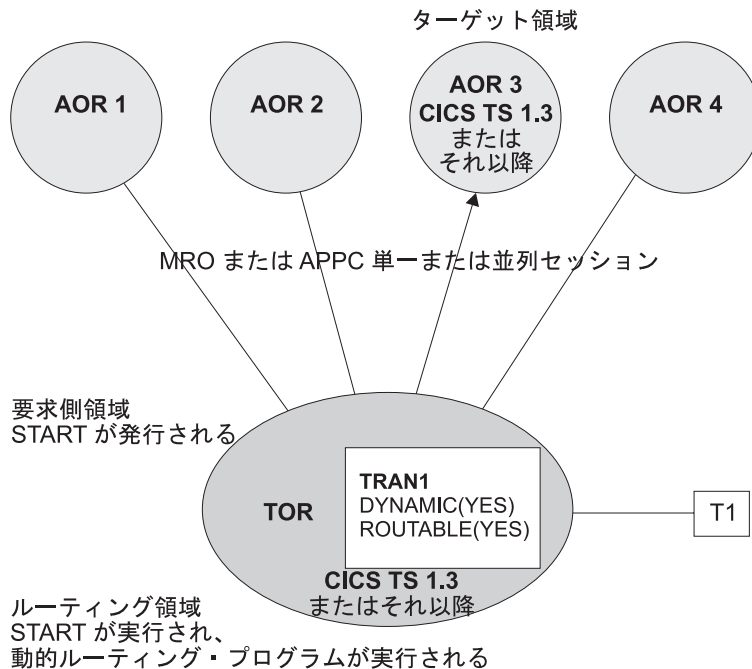


図 26. TOR で発行された端末関連 START コマンドの動的ルーティング: TOR とターゲット領域は、ともに CICS Transaction Server for OS/390、バージョン 1 リリース 3 以降です。TOR とターゲット領域は、MRO または APPC (単一または並列セッション) リンクで接続されています。TOR のトランザクション定義では、ROUTABLE(YES) と DYNAMIC(YES) が指定されています。

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。

- 要求領域が CICS Transaction Server for OS/390、バージョン 1 リリース 3 以降である。

注: 分散ルーティング・プログラムを要求領域および ターゲット 領域で起動するには、ターゲット領域も CICS Transaction Server for OS/390、バージョン

1 リリース 3 以降である必要があります。(分散ルーティング・プログラムを起動するポイントについては、「*CICS Customization Guide*」の『分散ルーティング・プログラムの作成 (Writing a distributed routing program)』を参照してください。)

- 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 リンクを使用していて、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には、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、リンクを介してすでに行われている(端末開始トランザクションのルーティングにより、要求領域は、ターゲット領域が CICS Transaction Server for OS/390、バージョン 1 リリース 3 以降のシステムであるかどうかを判別できる)。
 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。

さらに、要求を動的にルーティングする場合には、次のことを満たしていなければなりません。

- 要求側領域のトランザクション定義で、DYNAMIC(YES) を明示的に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ない)。

重要: 動的ルーティングの場合に候補とする 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を指定しているかどうか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。

遅延がある場合には、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 と一緒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など)。

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 (上記を参照)、分散ルーティング・プログラム、すなわち、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 (60 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パされます。

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 に機能シッパされます。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が、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には、分散ルーティング・プログラムは通知の場合しか呼び出されません。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り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 3 リリース 2 から CICS Transaction Server for OS/390、バージョン 1 リリース 3 領域ヘルパ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。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 (Interval control EXEC interface program exits)』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 アーキテクチャの実装である CICS に対するデータ・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、109 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。

図 27 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 27. 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。資料については、191 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、別のシステムヘル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」端末項目と区別が付きません。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、端末状況と入力の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 (すなわち、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。

表 2 は、経路リスト項目と OPCLASS オプションの組み合わせによって、リモート端末にルーティングされたメッセージの送達がどのように行われるかを示しています。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すでに使用可能でなければなりません。 227 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)。 72 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。

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
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。
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。これを使用することにより、端末オペレーターは、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すことができます。これは、リモート・トランザクションをローカル・システムに定義する必要がないという点で、通常のトランザクション・ルーティングとは異なります。しかし、CRTE を呼び出すときに使用する端末は、リモート・システムに定義されていなければなりません (または、『シップ可能』としてローカル・システムに定義されていなければなりません)。さらに、リモート・システムが保護されている場合には、端末オペレーターには RACF® 権限が必要です。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}]
```

ここで、xxxx は、DEFINE CONNECTION コマンドの CONNECTION オプションに指定されたリモート・システムの名前であり、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。(243 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呼び出すことはできません。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、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行されますが、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 を呼び出す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。

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 161 ページの『第 13 章 リモート・システムへのリンクの定義』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 213 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、243 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、297 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』
- 98 ページの『DPL 要求の静的ルーティング』
- 101 ページの『DPL 要求の動的ルーティング』
- 105 ページの『DPL サーバー・プログラムでの制約事項』
- 105 ページの『システム間のキューイング』
- 106 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。

アプリケーションは、要求されたプログラムの位置に関係なく作成することができます。プログラム制御 LINK コマンドを通常の方法で使用するだけです。一般に、CICS プログラム定義テーブル内の項目を使用すると、指定のプログラムをローカル領域 (**クライアント領域**と呼ばれる) ではなく、リモート領域 (**サーバー領域**と呼ばれる) に入れるように指定できます。

DPL 要求の図を 98 ページの図 28 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が分かります。CICS は、LINK 要求を適切な伝送形式に変更してから、それを CICB にシップして実行させます。

CICB では、ミラー・トランザクション (31 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、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。

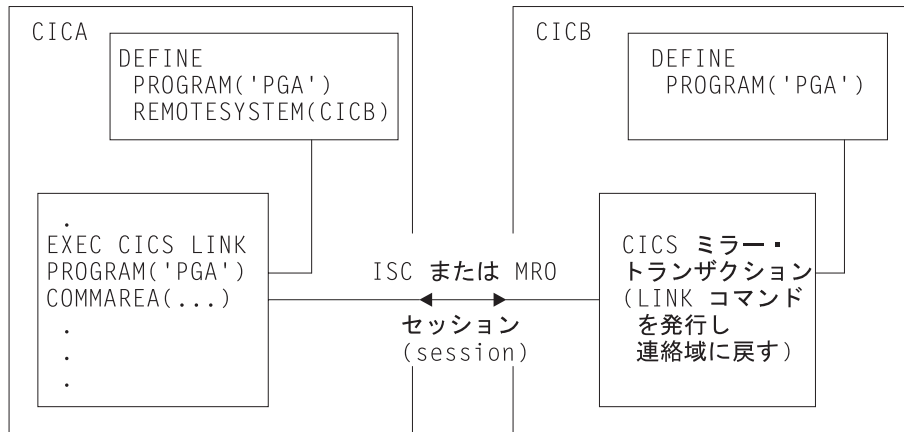


図 28. 分散プログラム・リンク

CICS リカバリーおよび再始動の各機能を使用すると、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るので、リモート・システムとローカル・システムでの保護リソースに対する変更が必ず整合性をもつことが保証されます。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・プログラムの位置を意識しないで、DPL を使用することができます。

CICS がサーバー・プログラムの位置を認識する方法は 2 通りです。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。

注: 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 3.2 以降であれば、DPL は、IPICによる接続だけでなく、MRO および ISC over SNA 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 ISC over SNA 接続の両方が存在し、両方に同じ名前を付けた場合には、IPICによる接続が優先されます。すなわち、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には、CICS は IPCONN 定義を使用します。

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、219 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。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 オプションの値が実行時に「ハードコーディング」されている場合 (すなわち、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 (100 ページの図 29 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。

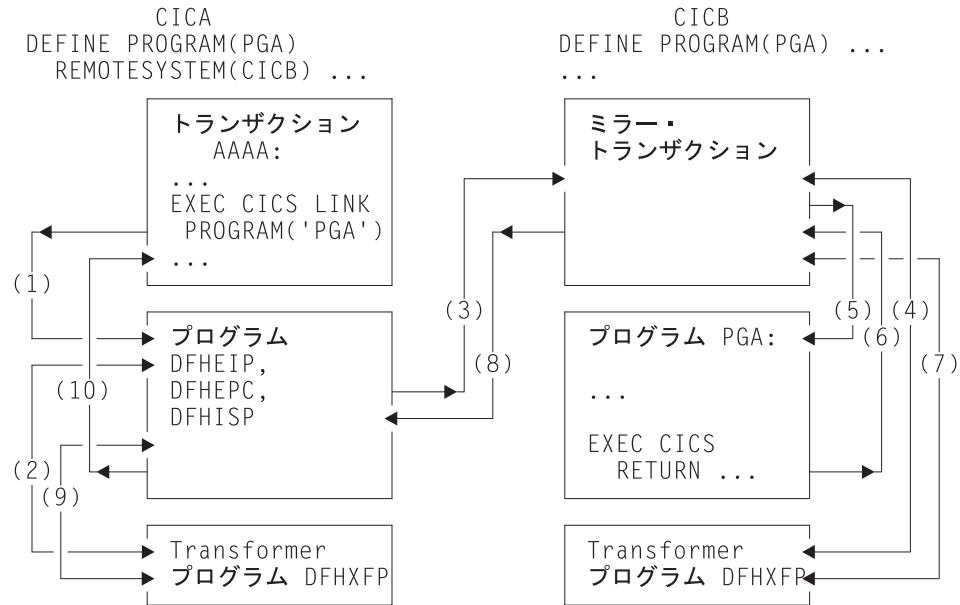


図 29. DPL における変換プログラムとミラー

図 29 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、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。

- 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 領域ヘルパーティグすれば、簡単なロード・balancing のメカニズムとして使用できます。ただし、ロード・balancing を取る場合には、CICS 動的ルーティング・プログラムを使用する方がよいでしょう。『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 (Program control program exits)』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・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 (59 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 (60 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。

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。

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 として定義したトラ

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。 67 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・プログラムは次のいずれかのようにならない限りなりません。

- ローカル・システムに対して DYNAMIC(YES) と定義されている。あるいは、
- ローカル・システムに対して定義されていない。

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、220 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・リンク要求が適格な場合、⁴ 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。これが発生するのは、プログラムが DYNAMIC(YES) と定義されている、またはプログラムが定義されていないのに、呼び出し側が LINK オプションの SYSID オプションでリモート領域の名前を指定した場合です。

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。

- 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・プログラムから戻された SYSID を利用できない、または認識しない、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には、代替の SYSID を提供してください。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

4. プログラム・リンク要求によって、「従来」の CICS-CICS 間の DPL 呼び出しと要求の両方を CICS の外側から受け取ったことを意味します。

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 は重要とならないことがあります。例えば、単に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作成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) ものとします。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、261 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、297 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。

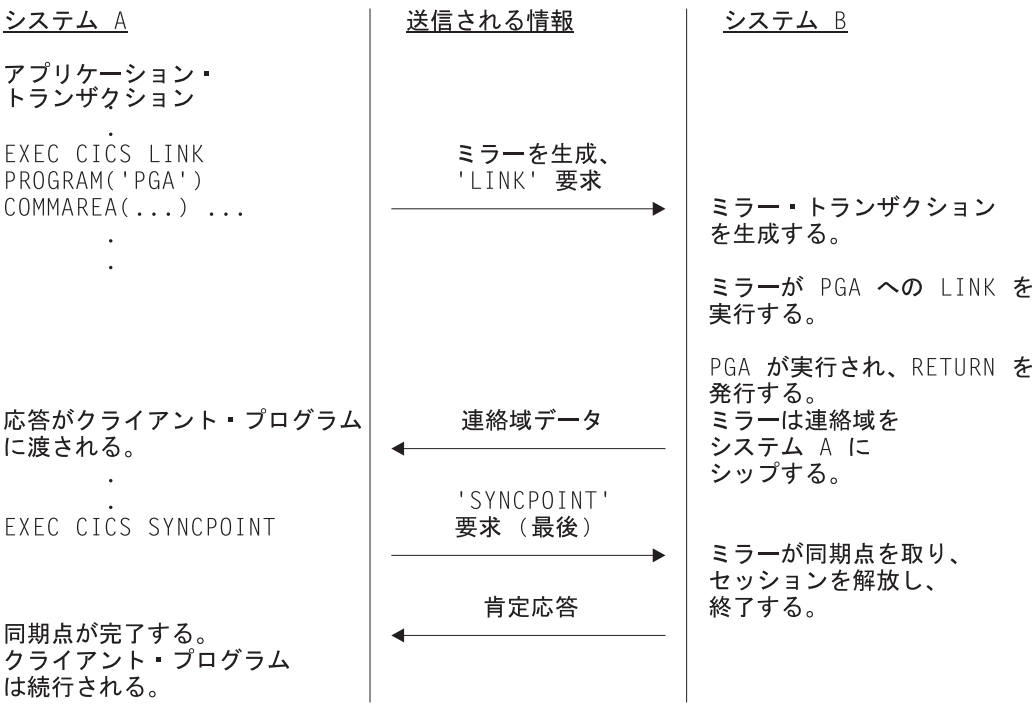


図 30. DPL でクライアント・トランザクションが同期点を出す

図 30 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。

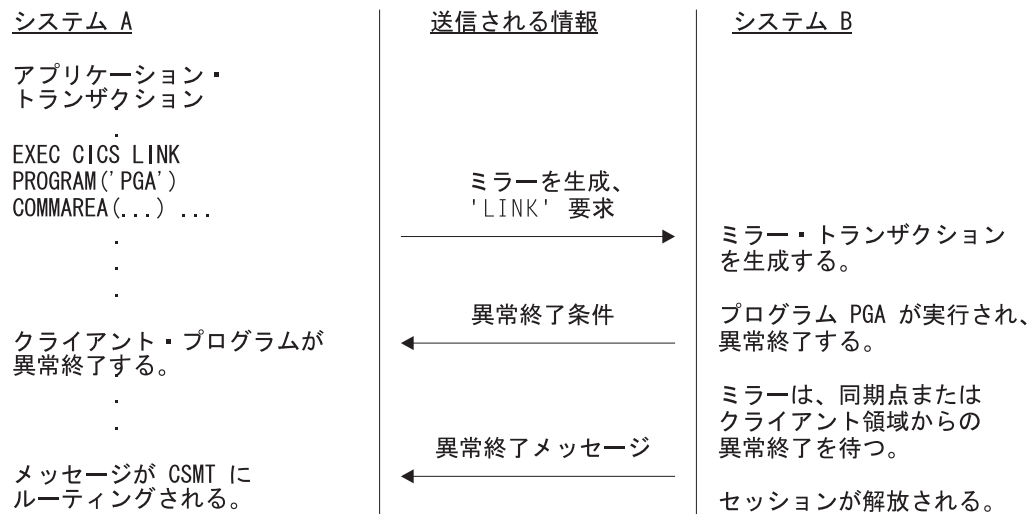


図 31. DPL でサーバー・プログラムが異常終了

図 31 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

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』
- 111 ページの『なぜ分散トランザクション処理なのか』
- 111 ページの『会話とは何か、なぜ必要なのか』
- 116 ページの『DTP では MRO か APPC か』
- 117 ページの『APPC マップ式会話か、基本会話か』
- 118 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取りたくなります。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ます。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。

これには多くのアクティビティーがかかわります。つまり、ネットワークには 2 つの流れが必要になり、流れるデータ量も非常に多くなります。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場合がありますが、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。112 ページの図 32 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。

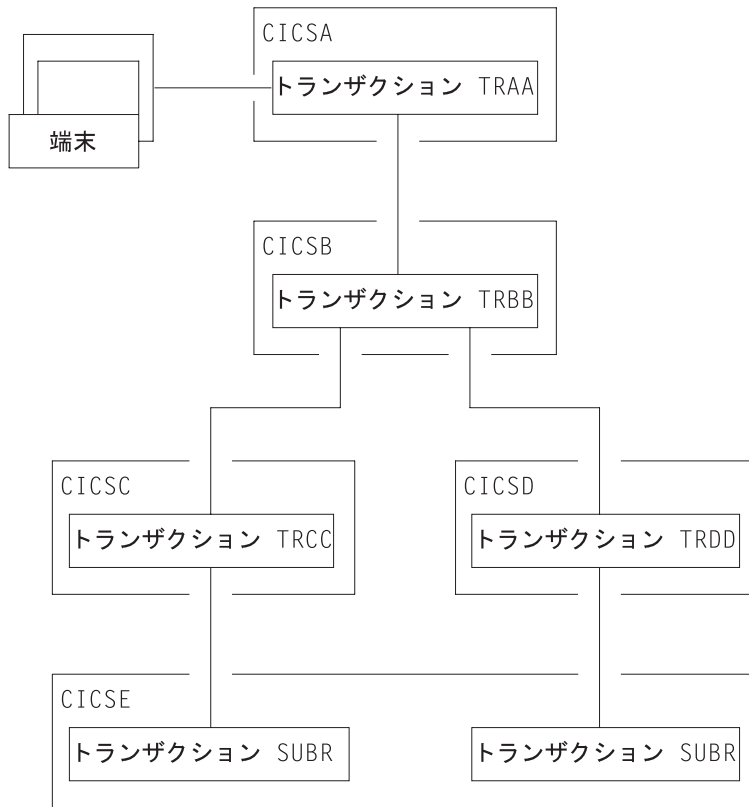


図 32. 複数システム構成の DTP

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。

109 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、単に 1 つのファイルのデータが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。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に保持されることがあります。データの形式とバック方法は、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 と全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。いくつかのキュー項目が受信され、そのうちいくつかは 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が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。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 つのレベルの同期が定義されています (**同期レベル**と呼びます)。

- 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ことはめったにありませんが、それらの相違点と類似点を知っていれば、互換性とマイグレーション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によって決まります。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 (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 のソフトウェアとハードウェアの詳しい要件については、122 ページの『XCF/MRO の要件』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

表 3. MRO と APPC の比較 (続き)

MRO	APPC
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る	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。
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。

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利用できる。	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

本書のこのパートでは、システム間通信や複数領域操作に関与する CICS システムのインストール要件について説明します。説明する内容は以下のとおりです。

- 複数領域操作を行う CICS の設定方法。
- システム間通信を行う CICS の設定方法。また、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。
- 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、およびその際に考慮すべき事項。

第 10 章 複数領域操作のインストールの注意点

このセクションでは、CICS 複数領域操作に特に当てはまるインストールの注意点について説明します。この章には、次のトピックが収められています。

- 『MRO のインストール手順』
- 122 ページの『XCF/MRO の要件』
- 124 ページの『インストール後の手順』

この章で示す **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 のインストール手順

複数領域操作のサポートをインストールするには、次のようにします。

1. CICS を MVS のサブシステムとして定義する。
2. 必要な CICS モジュールが、各自の CICS システムに組み込まれていることを確認する。
3. 一部のモジュールを MVS リンク・パック域 (LPA) に入れる。

システム間 MRO (XCF/MRO) のサポートをインストールすると、さらに管理の必要なことが発生します。これについては、122 ページの『XCF/MRO の要件』で説明します。

CICS を MVS のサブシステムとして追加する

CICS Transaction Server for z/OS で複数領域操作を行うには、MVS/VS 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。したがって、CICS を MVS のサブシステムとしてインストールしなければなりません。その方法については、「*CICS Transaction Server for z/OS インストール・ガイド*」の『クロスドメインに関する考慮事項 (Cross-domain considerations)』を参照してください。

MRO で必要なモジュール

システム初期設定パラメーターに **ISC=YES** を指定することによって、使用するシステムにシステム間通信管理プログラムを組み込む必要があります。

MVS リンク・パック域の MRO モジュール

複数領域操作の場合、整合性を保つために、共用域に常駐させるか、保護ストレージにロードする必要があるモジュールがいくつかあります。

CICS Transaction Server for z/OS, バージョン 3 リリース 2 用の次のモジュールのバージョンを MVS のリンク・パック域 (LPA) に入れる必要があります。

- DFHCSVC – CICS タイプ 3 SVC モジュール

複数領域操作では、異なる領域間でデータを転送するため、CICS 領域間通信モジュールを監視プログラム状態で実行する必要があります。CICS は、このため

に、この始動 SVC ルーチン (事前生成されたシステム・ロード・ライブラリー (CICSTS32.CICS.SDFHLOAD) 内にある) に対し通常の監視プログラム呼び出しを行います。

この SVC は MVS に定義されていなければなりません。その方法については、「*CICS Transaction Server for z/OS インストール・ガイド*」の『MVS への CICS SVC の定義 (Defining the CICS SVCs to your MVS)』を参照してください。

- DFHIRP - CICS 領域間通信プログラム

MRO データ・セットと起動システム

MRO を容易に使用できるように、CICS 配布ボリュームに CICS ジョブと CICS 始動プロシージャが用意されています。各 MRO 領域には、必要な CICS システム・データ・セットも作成する必要があります。詳しくは、「*CICS Transaction Server for z/OS インストール・ガイド*」の『CICS データ・セットの作成 (Creating the CICS data sets)』を参照してください。

XCF/MRO の要件

XCF/MRO を使って MVS イメージの間で通信する場合には、それらの MVS イメージをシスプレックスに含めなければなりません。

シスプレックスは複数の MVS イメージからなり、それらのイメージはハードウェア・エレメントとソフトウェア・サービスによって結合されます。シスプレックスにおいては、MVS イメージが基本サービスのプラットフォームとなりますので、CICS のような多重システム・アプリケーションはこれらのサービスを利用することができます。インストール・システムのワークロードが増加したら、MVS イメージをシスプレックスに追加すれば、ワークロードの増加に対処することができます。

通常、MVS アプリケーション・サブシステム (CICS など) の特定の機能 (1 つまたは複数のモジュール / ルーチン) は 1 つのメンバーとして結合されます (メンバーはシスプレックスの同じ MVS イメージにある)。そして、関連するメンバー群がそのグループとなります (グループはシスプレックスにおいて 1 つまたは複数の MVS イメージにまたがることのできる)。グループは、シスプレックスにおける完結した論理的なエンティティーです。XCF を使用してシスプレックスで通信する場合には、各 CICS 領域は、DFHIRP のサービスを使って、XCF グループにメンバーとして加わります。

シスプレックスのハードウェアとソフトウェアの要件

シスプレックスにおける MVS のインストールと管理の詳細については、「*MVS/ESA シスプレックスのセットアップ*」(GC88-6279) を参照してください。

XCF/MRO サポートの生成

1. 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、イメージにインストールされる DFHIRP のバージョンは異なっておりません。リンクされる CICS システムを含むすべての MVS イメージにおいて、拡張リンク・バック域 (ELPA) の DFHIRP バージョンは、要求され

るレベルのものでなければなりません。 DFHIRP モジュールは、そのイメージ内で最新の CICS リリースまたはそれより新しいリリースからのものである必要があります。

注: CICS TS for z/OS、バージョン 3.2 バージョンの DFHIRP (複数の XCF グループのサポートに必要) が使用できるのは、z/OS バージョン 1.7 またはそれ以降のバージョンに限られます。(z/OS では、バージョン 1.6 以降から複数の 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 グループへの参加が必要) 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。)

2. 各 CICS APPLID はそのシスプレックスで固有のものでなければなりません。
3. XCF 結合データ・セットを定義するときに使用する MVS のパラメーター MAXMEMBER の値が CICS XCF グループの最大サイズを満たすのに十分な高い値であることが必要です。シスプレックス内の XCF グループの最大サイズはこの値によって決まります。XCF グループの最大論理サイズのメンバー数は 2047 です。

XCF/MRO リンクを使用する外部 CICS インターフェース (EXCI) ユーザーもこの XCF グループに加わります。したがって、MAXMEMBER の値は、CICS XCF 最大グループのすべての CICS 領域と EXCI XCF/MRO ユーザーがこのグループに同時に加わるのに十分な大きさに設定しなければなりません。

XCF グループの CICS 領域と EXCI ユーザーをリストするには、MVS DISPLAY コマンドを使用します。例えば、DFHIR001 XCF グループの CICS 領域と EXCI ユーザーをリストするには、次のコマンドを使用します。

```
DISPLAY XCF,GROUP,DFHIR001,ALL
```

重要:

MAXMEMBER のデフォルトに注意してください。この値が小さすぎて、XCF 最大グループのすべての CICS 領域と EXCI ユーザーがグループに加われないことがあります。これは、ごく少数の CICS XCF グループを持つ場合、特に重要です。

同じように、必要以上に大きな値を設定すると、XCF 用の結合データ・セット (複数) が大きなものになります。データ・セットが大きいと、その中の項目を見つけるのにそれだけ多くの時間がかかります。

MAXMEMBER には、最大の CICS XCF グループ内の CICS 領域と EXCI ユーザーの合計よりも 10 から 15 大きい値を設定するのがよいでしょう。

各 CICS 領域は、DFHIRP にログオンすると、XCF グループに加わります。そのメンバー名は、その APPLID (NETNAME) で、MRO パートナーに使用されます。XCFCGROUP システム初期設定パラメーターに XCF グループ名が指定されています。XCFCGROUP が指定されない場合、XCF グループ名はデフォルトの DFHIR000 になります。

CICS は、接続時、IXCQUERY マクロを呼び出して、接続先の CICS 領域が同じ MVS イメージにあるかどうかを判断します。同じイメージにあれば、CICS は、

MRO アクセス方式として IRC か XM を使用します (どちらを使用するかは接続定義による)。パートナーが異なる MVS イメージにある場合、CICS は、接続定義にどのアクセス方式が定義されていても、XCF をアクセス方式として使用します。

注: 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 グループ間のメンバーは、同じ MVS イメージ内にある場合でも、MRO (または XCF/MRO) 経由で通信することはできません。

インストール後の手順

MRO サポートをインストールした後で、CICS でそれを使用できるようにするために、次のようにします。

1. リモート・システムへの MRO リンクを定義する。資料については、164 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・システムとリモート・システムのリソースを定義する。243 ページの『第 17 章 ローカル・リソースの定義』および 213 ページの『第 16 章 リモート・リソースの定義』をそれぞれ参照してください。
3. CICS が IRC アクセス方式にログオンすることを指定する。「*CICS Transaction Server for z/OS インストール・ガイド*」の『MRO サポートおよび ISC サポートのインストール (Installing MRO and ISC support)』を参照してください。

第 11 章 システム間通信のインストールの注意点

この章では、CICS をシステム間通信環境で使用する場合に特に当てはまるインストール上の注意点について説明します。

章には、以下のトピックが含まれています。

- 『SNA 経由のシステム間通信用のサポートのインストール』
- 132 ページの『IP 相互接続用のサポートのインストール』

SNA 経由のシステム間通信用のサポートのインストール

このトピックでは、SNA 経由のシステム間通信 (ISC over SNA) 用のサポートをインストールする方法について説明します。「*CICS Transaction Server for z/OS インストール・ガイド*」と合わせてお読みください。

また、このトピックには、ACF/VTAM および IMS をISC over SNA と併用する場合における、これらの製品のインストール時の注意事項も記載されています。

このトピックには、以下のトピックが含まれています。

- 『ISC に必要なモジュール』
- 『CICS のための ACF/VTAM 定義』
- 127 ページの『IMS の考慮事項』

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 に必要なモジュール

システムには、システム間通信プログラムを組み込む (VTAM および ISC のシステム初期設定パラメーターで「YES」を指定する) 必要があります。システム初期設定パラメーターの指定については、「*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。

CICS のための ACF/VTAM 定義

CICS システムを ACF/VTAM に対して定義する場合は、VTAM APPL ステートメントに次のオペランドを指定してください。

MODETAB=logon-mode-table-name

このオペランドは、カスタマイズされたログオン・モード項目を含む VTAM ログオン・モード・テーブルを指定します。(126 ページの『CICS の ACF/VTAM LOGMODE テーブル項目』を参照してください。) このオペランドは、MODEENT 項目を IBM 提供のデフォルト・ログオン・モード・テーブルに (名前を変更せずに) 追加する場合は、省略することができます。

AUTH=(ACQ,SPO,VPAGE[,PASS])

ACQ は、CICS が LU タイプ 6 セッションを獲得できるようにするために必

要です。SPO は、CICS が MVS MODIFY *vtamname* USERVAR コマンドを出せるようにするために必要です (USERVAR の重要度に関する詳細については、「CICS/ESA® 3.3 CICS XRF Guide」を参照してください)。VPACE は、システム間フローのペーシングを可能にするために必要です。

PASS は、EXEC CICS ISSUE PASS コマンドを使用する場合、必要です。このコマンドは、既存の端末セッションを他の VTAM アプリケーションに渡します。

VPACING=number

このオペランドは、別の論理装置が、ペーシング応答の受信待機に入る前に、システム間セッションで送信できる通常フロー要求の最大数を指定するものです。

適切なペーシング・カウントを選択しようとする際には注意してください。値が小さすぎると、必要な回線反転の回数が多くなりますので、スループットが低下するおそれがあります。値が大きすぎると、ストレージが余分に必要になります。

EAS=number

このオペランドは、CICS がセッションを確立できるネットワーク・アドレス可能単位の数指定します。この数字には、当該 CICS システムでの並列セッションの総数も含める必要があります。

PARSESS=YES

このオプションは、LU タイプ 6 並列セッションのサポートを指定します。

SONSCIP=YES

このオペランドは、セッション障害通知 (SON) のサポートを指定します。SON を指定すると、CICS は、特定の場合、オペレーターの介入を必要とせずに障害のあったセッションをリカバリーすることができます。

APPC=NO

ACF/VTAM バージョン 3.2 以降では、CICS に VTAM マクロを使用させるために必要です。CICS は APPCCMD マクロを出しません。

VTAM APPL ステートメントの詳細については、「OS/390 *eNetwork Communications Server: SNA Resource Definition Reference*」を参照してください。

CICS の ACF/VTAM LOGMODE テーブル項目

APPC セッションでは、CICS DEFINE SESSIONS コマンドの MODENAME オプション (174 ページの『APPC リンクの定義』を参照) を使用して、VTAM ログモード項目を識別し、それによって、VTAM サービス・クラス・テーブルの必要な項目を識別することができます。一群の APPC セッションを CICS に定義するときに指定するモード名は、VTAM LOGMODE 名と一致しなければなりません。VTAM LOGMODE テーブルには、次の形式の項目があれば十分です。

```
MODEENT LOGMODE=modename
MODEEND
```

LU サービス・マネージャー・モードセット (SNASVCMG) 用の項目も必要です。

```
MODEENT LOGMODE=SNASVCMG
MODEEND
```

単一セッション APPC 端末で自動インストールを使用する場合には、MODEENT 項目に追加情報が必要です。VTAM LOGON モード・テーブルのコーディングに関するプログラミング情報については、「CICS カスタマイズ・ガイド」を参照してください。

クロスドメインのリンクである CICS-IMS 間リンクでは、IMS LOGMODE 項目と CICS アプリケーション ID (XRF システムの総称アプリケーション ID) を、DLOGMOD または MODETAB パラメーターを使用して関連付ける必要があります。

IMS の考慮事項

CICS システムで CICS-IMS 間のシステム間通信を使用する場合は、CICS と IMS のインストール・システムの間に完全な互換性がなければなりません。

次に続く各項の目的は、IMS システムのインストール担当者との意思疎通を効果的に行えるようにすることです。これらの項は、自身がその担当者である場合にも役立ちます。さらに、161 ページの『第 13 章 リモート・システムへのリンクの定義』、特に、互換性のある CICS ノードと IMS ノードの定義に関する項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/ESA 導入の手引き」を参照してください。

IMS のための ACF/VTAM 定義

IMS システムを VTAM に定義する場合は、VTAM APPL ステートメントに次のオペランドを指定する必要があります。

AUTH=(ACQ,VPACE)

ACQ は、IMS が LU タイプ 6 セッションを獲得できるようにするために必要です。VPACE は、システム間フローのペーシングを可能にするために必要です。

VPACING=number

このオペランドは、別の論理装置が、ペーシング応答の受信待機に入る前に、システム間セッションで送信できる通常フロー要求の最大数を指定するものです。初期値には 5 を指定するのがよいでしょう。

EAS=number

ネットワーク・アドレス可能単位の数には、この IMS システムの並列セッションの総数も含める必要があります。

PARSESS=YES

このオペランドは、LU タイプ 6 並列セッションのサポートを指定します。

VTAM APPL ステートメントの詳細については、「OS/390 eNetwork Communications Server SNA 資源定義解説」を参照してください。

IMS のための ACF/VTAM LOGMODE テーブル項目: IMS では、VTAM ログモード・テーブルの項目にいくつかの BIND パラメーターを指定することができます。CICS ログモード・テーブル項目は、IMS システムの該当項目と一致しなければなりません。IMS は、次の場所に指定されたモード・テーブル項目を (この順序で) 使用します。

1. TERMINAL マクロの MODETBL パラメーター

2. CINIT に指定されたモード・テーブル項目
3. VTAMLST APPL ステートメントの DLOGMODE パラメーター、または IMS /OPNDST コマンドの MODE パラメーター
4. ACF/VTAM のデフォルト

図 33 に一般的な IMS ログモード・テーブル項目を示します。

```

LU6NEGPS  MODEENT LOGMODE=LU6NEGPS,  NEGOTIABLE BIND
          PSNDPAC=X'01',              PRIMARY SEND PACING COUNT
          SRCVPAC=X'01',              SECONDARY RECEIVE PACING COUNT
          SSNDPAC=X'01',              SECONDARY SEND PACING COUNT
          TYPE=0,                     NEGOTIABLE
          FMPROF=X'12',              FM PROFILE 18
          TSPROF=X'04',              TS PROFILE 4
          PRIPROT=X'B1',              PRIMARY PROTOCOLS
          SECROT=X'B1',              SECONDARY PROTOCOLS
          COMPROT=X'70A0',            COMMON PROTOCOLS
          RUSIZES=X'8585',            RU SIZES 256
          PSERVIC=X'060038000000380000000000'  SYSSMSG/Q MODEL
MODEEND

```

図 33. 一般的な IMS ログモード・テーブル項目

システム間通信での IMS システム定義

このセクションでは、IMS システム定義で使用する IMS ISC 関連のマクロとパラメーターを要約します。さらに、185 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。IMS インストールの詳細については、IMS 製品のインストールの手引きを参照してください。

COMM マクロ:

APPLID=name

IMS システムの applid を指定します。XRF サポートなしで生成された IMS システムの場合、その IMS システムを CICS に定義する際、DEFINE CONNECTION の NETNAME オプションには通常、この名前を指定します。

しかし、次の点に注意してください。

- XRF のサポートがある IMS システムでは、CICS NETNAME オプションには、COMM マクロの applid の代わりに、IMS.PROCLIB の DFSHSBxx メンバーに定義されている USERVAR (つまり、総称 applid) を指定します。
- COMM マクロの APPLID に NONE を指定し、XRF を使用しない場合は、CICS NETNAME オプションには IMS 始動ジョブの EXEC ステートメントのラベルを指定します。
- IMS システムを開始済みタスクとして開始する場合は、NETNAME には開始済みのタスク名を指定します。

IMS システム名の指定方法については、185 ページの『システム名』のページを参照してください。

RECANY=(number,size)

VTAM の「receive any」コマンドに使用される IMS バッファの数とサイズ

を指定します。ISC セッションの場合は、バッファ・サイズのオーバーヘッドが 22 バイトあります。したがって、この値は、`DEFINE SESSIONS` の `SENDSIZE` オプションに指定された CICS バッファ・サイズよりも少なくとも 22 バイト大きくなければなりません。

このサイズは、IMS システムに接続されている他のすべての ACF/VTAM 端末に適用されますので、IMS ネットワークのどの端末からの入力に対しても十分な大きさでなければなりません。

EDTNAME=name

IMS システムにおける ISCEDT の別名を指定します。CICS-IMS 間 ISC では、別名は 4 文字以下でなければなりません。

TYPE マクロ:

UNITYPE=LUTYPE6

ISC にはこれを指定しなければなりません。

`TERMINAL` マクロのパラメーターを `TYPE` マクロに指定する場合には、それらのパラメーターは、そのタイプに定義されているすべての端末に共通でなければなりません。

TERMINAL マクロ:

`TERMINAL` マクロは、IMS に対してリモート CICS システムを識別するものです。したがって、これは、CICS における `DEFINE CONNECTION` と同じことです。

NAME=name

IMS に対して CICS ノードを識別します。これは、CICS システムの `applid` (XRF システムの場合は総称 `applid`) と同じでなければなりません。

OUTBUF=number

IMS 出力バッファのサイズを指定します。これは 256 以上でなければなりません。また、データとともに送られる機能管理ヘッダーのサイズも含める必要があります。これは、システム間セッションでの `DEFINE SESSIONS` コマンドの `RECEIVESIZE` オプションに指定された値を超えてはなりません。

SEGSIZE=number

IMS が着信メッセージの非ブロック化に使用する作業域のサイズを指定します。CICS が送信できる最長チェーンのサイズを使用するようにしてください。ただし、IMS レコード・モード (VLVB) が排他的に使用されている場合は、最大レコード (RU) サイズを指定することができます。

MODETBL=name

使用される VTAM モード・テーブル項目の名前を指定します。CICS システムが異なる SNA ドメインにある場合は、このパラメーターを省略する必要があります。

OPTIONS=[NOLTWAITWA]

ログ・テープ先行書出し (LTWA) が必要かどうかを指定します。LTWA では、IMS は、すべての活動並列セッションのセッション再始動情報をログ記録してから、同期点要求を送信します。LTWA は、保全性の点から推奨されますが、パフォーマンスが低下するおそれがあります。NOLTWA がデフォルトです。

OPTIONS=[SYNCSESSIFORCSESS]

セッションが異常終了したときにメッセージの再同期が必要かどうかを指定します。SYNCSCESS がデフォルトです。この場合、着信と発信の両方の順序番号が一致しないと（または CICS をコールド・スタートしないと）、セッションを再始動することはできません。FORCSESS を指定すると、不一致が生じた場合でも、セッションを再始動することができます。SYNCSCESS を指定するようにしてください。

OPTIONS=[TRANSRESPINORESPIFORCRESP]

必要な応答モードを指定します。

TRANSRESP

応答モードはトランザクション単位で決められます。これはデフォルトです。

NORESP

応答モードのトランザクションは許可されません。CICS 用語でいえば、CICS アプリケーションは、SEND コマンドを使って IMS トランザクションを開始することはできません。つまり、START コマンドを使用しなければなりません。

FORCRESP

すべてのトランザクションを応答モードにします。CICS 用語でいえば、CICS アプリケーションは、START コマンドを使って IMS トランザクションを開始することはできません。つまり、SEND コマンドを使用しなければなりません。

TRANSRESP を指定するようにしてください。

OPTIONS=[OPNDSTINOPNDST]

セッションをこの IMS システムから確立できるかどうかを指定します。OPNDST を指定するようにしてください。

{COMPT1|COMPT2|COMPT3|COMPT4}={SINGLEn|MULn}

IMS ISC ノードの IMS コンポーネントを指定します。最大 4 つのコンポーネントを各ノードに定義することができます。各セッションに使用される入力コンポーネントと出力コンポーネントは、SUBPOOL マクロの ICOMPT および COMPT の各パラメーターによって選択されます。

次のタイプのコンポーネントが定義できます。

SINGLE1

IMS によって非同期出力に使用されます。各 SNA ブラケットについて 1 つの出力メッセージが送信されます。このメッセージは、ブラケットを開始する場合もしない場合もありますが、常にブラケットを終了します。

SINGLE2

各メッセージは、SNA 方向転換標識 (CD) とともに送信されます。

MULT1

指定の LTERM についての非同期メッセージがすべて送信されてから、ブラケットが終了します。ブラケット終了 (EB) は、LTERM についての最後のメッセージが確認されて、デキューされると起こります。

MULT2

MULT1 と同じですが、CD が EB の代わりに送信されます。

SESSION=number

リンクの並列セッションの数を指定します。各セッションは、IMS SUBPOOL マクロおよび CICS DEFINE SESSIONS コマンドによって表されます。

EDIT=[{NOIYES}][,{NOIYES}]

ユーザー提供の物理出力編集ルーチンと入力編集ルーチンを使用するかどうかを指定します。

VTAMPOOL マクロ:

リモート・システムへの個々のセッションを定義する SUBPOOL マクロのリストの始めには、SUBPOOL マクロを指定します。

SUBPOOL マクロ:

SUBPOOL マクロは、リモート・システムに対する各セッションに必要です。

NAME=subpool-name

このセッションの IMS 名を指定します。CICS-IMS 間セッションは、そのセッションの CICS 名と、IMS サブプール名から作成される「セッションと修飾子の対」によって識別されます。

セッションの CICS 名は、そのセッションに関する DEFINE SESSIONS コマンドの SESSNAME オプションに指定されます。

IMS サブプール名は、DEFINE SESSIONS コマンドの NETNAMEQ オプションによって、CICS に指定されます。

NAME マクロ:

NAME マクロは、サブプールに関連する論理端末名を定義します。複数の LTERM を各サブプールに定義することができます。

COMPT={1|2|3|4}

このセッションに関連する出力コンポーネントを指定します。指定するコンポーネントによって、IMS ISC がメッセージの処理に使用するプロトコルが決まります。SINGLE1 として定義されている出力コンポーネントを使用するようにしてください。

ICOMPT={1|2|3|4}

このセッションに関連する入力コンポーネントを指定します。IMS は、メッセージを受け取ると、一致する入力コンポーネント番号をもつ NAME マクロを検索して、入力ソース端末を判別します。COMPT1 入力コンポーネントを、CICS が START コマンドの送信に使用する各セッションに定義する必要があります。

EDIT=[{NOIYES}][,{ULCIUC}]

最初のパラメーターは、ユーザー提供の論理端末編集ルーチン (DFSCNTEO) を使用するかどうかを指定します。

2 番目のパラメーターは、伝送前に出力を大文字に変換するか (UC)、またはしないか (ULC) を指定します。

IP 相互接続用のサポートのインストール

このトピックでは、IP 相互接続 (IPIC) 用のサポートをインストールする方法について説明します。「*CICS Transaction Server for z/OS* インストール・ガイド」と合わせてお読みください。

- CICS システム初期設定テーブルで、またはシステム初期設定のオーバーライドとして TCPIP=YES を指定します。
- 接続する 2 つの CICS 領域間の IPIC による接続を 169 ページの『IP 相互接続リンクの定義』に従って定義します。

第 12 章 VTAM 総称リソースのインストールの注意点

重要:

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。

#

- 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
- CICS Transaction Server for OS/390、バージョン 1 リリース 3

この章では、機能的に同等な CICS 端末専有領域 (TOR) 群を含む CICSplex において、VTAM 総称リソースをどのように使用することにより、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るかについて説明します。

VTAM 総称リソースの概要については、20 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。

注: この章では、リモート・システムへの接続の定義などの作業 (本書の後のページで説明します) についての知識が読者にあるものとします。この章の理解には、161 ページの『第 13 章 リモート・システムへのリンクの定義』が役立ちます。

章には、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』
- 134 ページの『VTAM 総称リソースを使用するための CICSplex の計画』
- 136 ページの『総称リソース環境における接続の定義』
- 137 ページの『VTAM 総称リソース・サポートの生成』
- 138 ページの『総称リソースへの TOR のマイグレーション』
- 140 ページの『TOR を総称リソースから除去する』
- 141 ページの『TOR を別の総称リソースへ移動する』
- 141 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』
- 146 ページの『類縁性の終了』
- 150 ページの『ATI での総称リソースの使用』
- 153 ページの『ISSUE PASS コマンドの使用』
- 154 ページの『規則のチェックリスト』
- 155 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。

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 領域だけにします。
- 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 LUTYPE6.1⁵の接続があるか。
 - 総称リソースのメンバーとメンバー⁶
 - 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
- 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、つまり CICS 領域の集合は、総称リソース名を持ちます。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。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。

5. CICS-CICS 間接続では、LUTYPE6.1 よりも APPC の使用を推奨します。

6. 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。

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、196 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**が作成されるので、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 オプションの指定には 2 つの方法があります。

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 に関する VTAM 総称リソース・サポートを生成するには、次のようにします。

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 (147 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』のページを参照)、それを使って同じ効果が得られます。147 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』のページに示すスケルトン・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、141 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

一般的には、次の方法をお勧めします。

- 分かりやすくするために、メンバーが 1 つだけの総称リソースをまず作成します。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 接続がすで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。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 がすでに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、⁷ それらの接続では、総称リソース名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。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 プロファイルは変える必要がありません。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 だからです。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。

7. その総称リソースの他のメンバーへの接続は除きます。

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、155 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。

- 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。資料については、146 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、次のようにします。

1. それが所有する類縁性を終了する。資料については、146 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。137 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。147 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』のページ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 (143 ページの図 34 を参照)。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 CICS 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 CICS 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。

CICS の任意のメンバーが CICS 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 CICS メンバーは、CICS の総称リソース名接続の定義を探します。(それがなければ、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 CICS の同じメンバーへの、CICS の異なるメンバーから行われるそれ以後の接続は、CICS のメンバー名を NETNAME として、CICS 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、CICS の異なるメンバーから CICS の同じメンバーへのそれ以後の接続は、その CICS のメンバー名を NETNAME として、その CICS メンバーに自動インストールされます。『例』 の例は、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。⁸ (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

143 ページの図 34 から 145 ページの図 37 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、

8. DEFINE SESSIONS の MAXIMUM オプションについては、176 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。

APPC 並列セッション同期レベル 2 のリンクです。

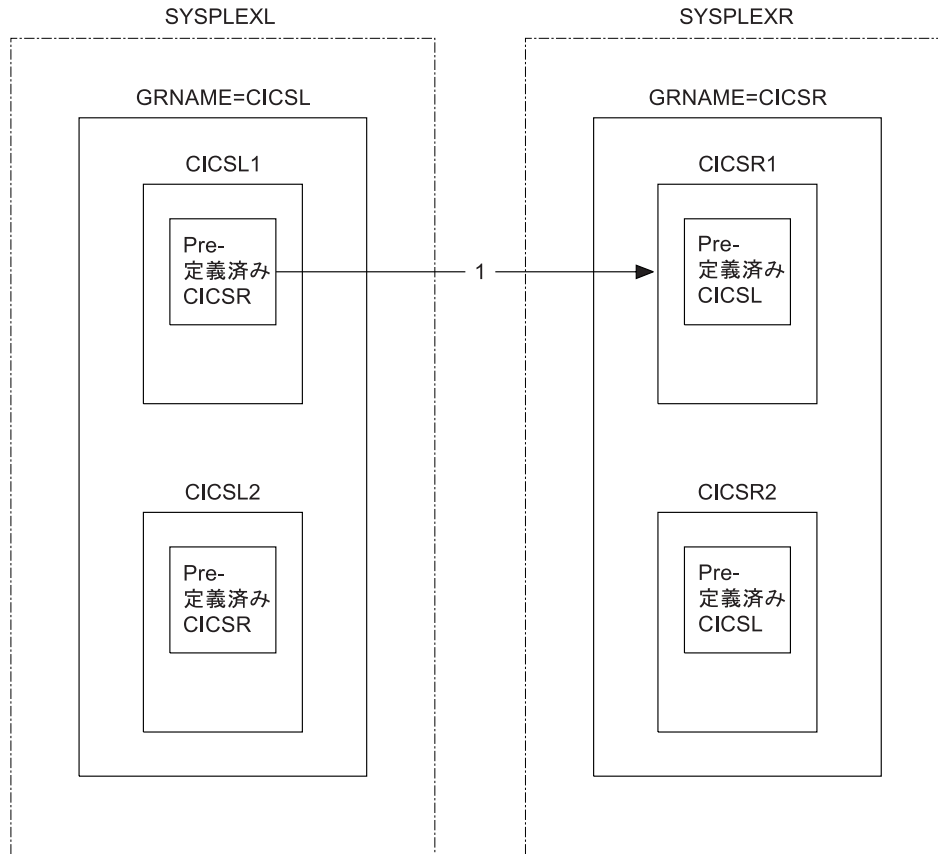


図 34. この図は 2 つのシスプレックス SYSPLEXL と SYSPLEXR を示しています：それぞれのシスプレックスには CICS 総称リソース・グループがあります。CICSL グループのメンバー CICSL1 が、SYSPLEXR にある CICSR グループのメンバーへの接続を獲得しようとしています。

図 34 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、CICS1 へ行きます。CICSL1 と CICS1 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 CICS1 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。146 ページの『類縁性の終了』および 324 ページの『APPC 接続静止処理』を参照してください。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 CICS1 にルーティングします。また、CICS1 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。

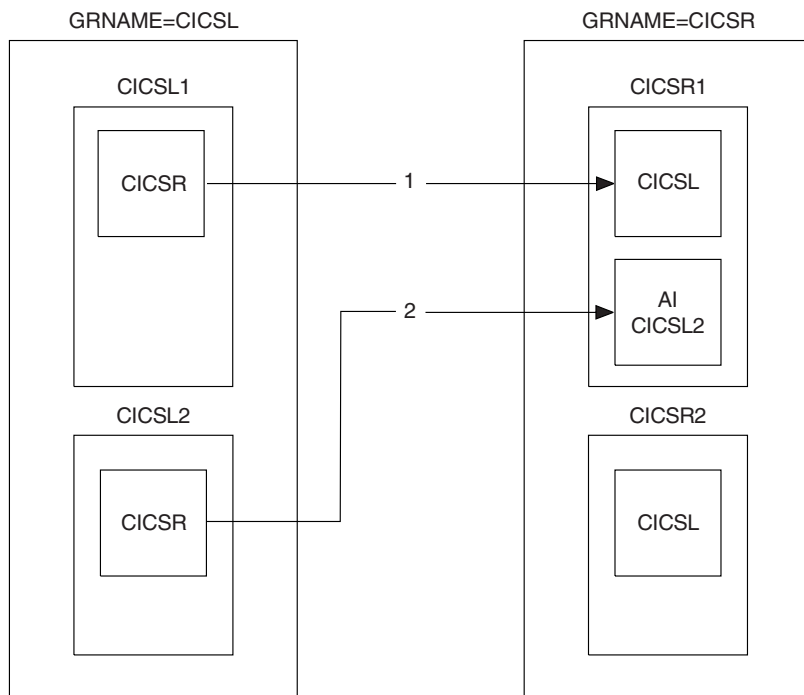


図 35. 2 番目の流れ、CICSL2-CICSR

図 35 は、CICSL2 から CICS1 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 CICS1 ヘルパーティングしていますが、CICS1 の他のメンバーヘルパーティングすることも可能です。

CICSL2 にある CICS2 の定義済み接続が使用されます。CICS1 は、CICS1 の接続項目を探します。これはすでに使用中ですので、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 CICS1 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。

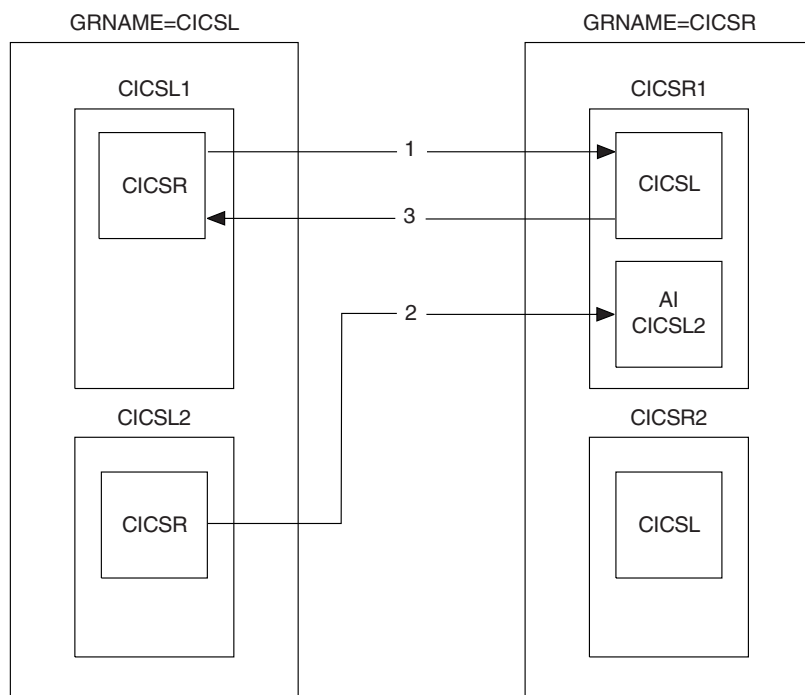


図 36. 3 番目の流れ、CICSR1-CICSL

図 36 は、今度は 3 番目の CICSR1 から CICSL への流れを示しています。類縁性がすでに存在しますので、その流れは CICSL1 へ強制されます。

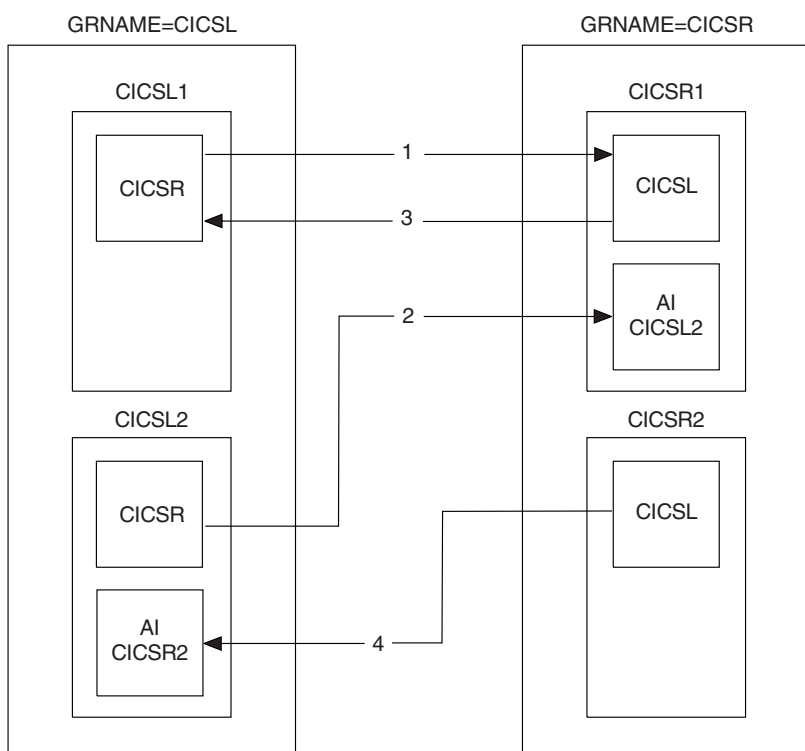


図 37. 4 番目の流れ、CICSR2-CICSL

145 ページの図 37 は、今度は 4 番目の CICS R2 から CICS L への流れを示しています。この流れは CICS L 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 CICS L2 ヘルパーティングしています。

CICS R2 にある CICS L の定義済み接続項目は使用されていないので、今回、それが使用されます。CICS L2 は、CICS R 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、CICS R2 の項目が自動インストールされます。

SYS PLEX L と SYS PLEX R に類縁性が作られ、CICS L2 と CICS R2 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 が初期始動やワールド・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が自動的に行ってくれます。324 ページの『APPC 接続静止処理』を参照してください。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、接続がすで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けることができます。

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が必要です。

- 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。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

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 RCVCMDC 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、RCVCMDC 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。

```
RCVCMDC 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。

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。これを実行すると、VTAM 持続セッションが使用されている場合、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

```
//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 38. 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、70 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、70 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 にシッパされ、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 にシッパされます。
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。その後、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。シッパされた端末定義は、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パされた定義はまだ 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) にシッパします。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) にシッパします。

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 にシッパされます。
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。そして、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 にシッパされます。しかし、その端末は TOR1 にはもう定義されていません。また、シッパされた端末定義は、タイムアウト削除メカニズムによって AOR1 からすでに削除されています。

• **結果:**

シッパされた端末定義は AOR1 からすで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) にシッパ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 (Writing a node error program)』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 に端末を移す場合には、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、141 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

重要:

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。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。

157 ページの図 39 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。

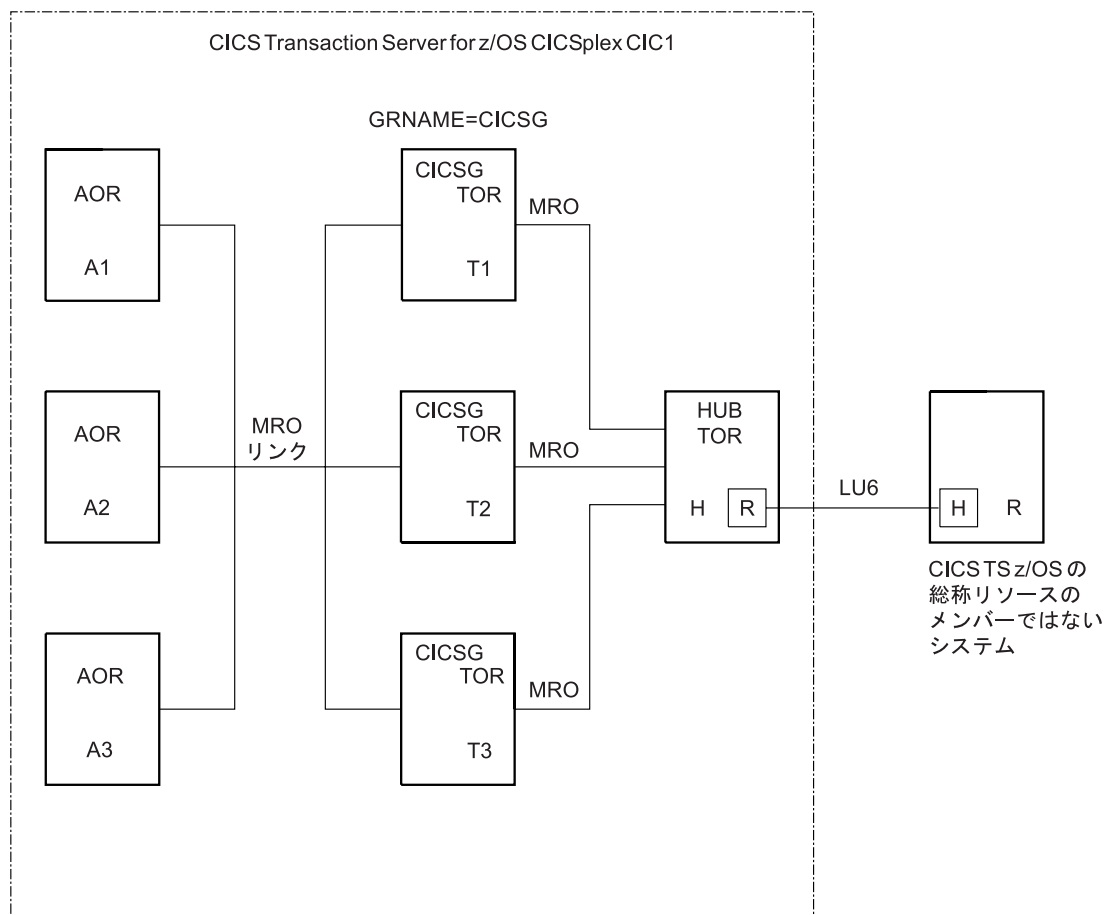


図 39. ネットワーク・ハブ： ハブは通常、総称リソース・グループのメンバーから、CICS Transaction Server for z/OS 総称リソースのメンバーではないシステムへのアウトバウンド LU6 要求に使用されます。

図 39 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

本書のこのパートでは、CICS 相互通信環境で必要になる各種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。このパートでは、以下を実行する方法について説明します。

- リモート・システムへのリンクの定義。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
 - 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
- マスター端末トランザクション (CEMT) を使用した、APPC リンクの管理。
- ローカル CICS システムへのリモート・リソースの定義。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
- ISC および MRO のローカル・リソースの定義。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変更して、独自の通信プロファイルを用意することもできます。

第 13 章 リモート・システムへのリンクの定義

この章では、他の CICS 領域や他の非 CICS システムへの通信リンク（「接続」）を定義し、管理する方法を説明します。

この章で説明するリンクは、次のとおりです。

- 複数領域操作 (MRO) のリンク
- 外部 CICS インターフェース (EXCI) によって使用するためのリンク
- CICS TS for z/OS、バージョン 3.2 (以降) のリモート領域に対する IPICリンク (分散プログラム・リンクと併用するための)
- 論理装置タイプ 6.2 (APPC) プロトコルを使用するリモート・システムへの ISC over SNA リンク
- 論理装置タイプ 6.1 プロトコルを使用するリモート IMS システムへの ISC over SNA リンク
- CICS トランザクション・ルーティングのための間接リンク

ACF/VTAM アプリケーション間機能を使用するリンクは、システム間リンクとまったく同じに扱われるため、LUTYPE6.1 リンクまたは APPC リンクとして定義できます。

章には、以下のトピックが含まれています。

- 『リンク定義の紹介』
- 163 ページの『リモート・システムの識別』
- 164 ページの『複数領域操作のリンクの定義』
- 168 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 相互接続リンクの定義』
- 174 ページの『APPC リンクの定義』
- 184 ページの『論理装置タイプ 6.1 リンクの定義』
- 184 ページの『CICS-IMS 間 LUTYPE6.1 リンクの定義』
- 191 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』
- 196 ページの『XRF のための総称 applid と特定 applid』

リンク定義の紹介

MRO 接続と ISC over SNA (APPC および LUTYPE 6.1) 接続は、IPICによる接続とは別に定義されます。

MRO 接続と ISC over SNA 接続

リモート・システムへの MRO または ISC over SNA 接続の定義は、次の 2 つの部分で構成されています。

1. リモート・システムそのものの定義
2. 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 端末の場合は、DEFINE TERMINAL や DEFINE TYPETERM を使用する代替の定義方法が使用できます。

リモート・システムが CICS 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 (例えば IMS) であるなら、そのリンク定義は、リモート・システムの互換性のある定義と一致しなければなりません。セッション特性にほとんど、あるいはまったく柔軟性のないリモート・システム (例えば、APPC 端末) の場合には、そのリンク定義は、関連するリモート・システムの固定属性と一致しなければなりません。

IPICによる接続

IPIC 接続 (「IPCONN」。)

IPIC 接続を作成するには、補完関係にある次の 2 つのリソースをインストールする必要があります。

1. IPCONN 定義では、TCP/IP 接続のアウトバウンド 属性を指定します。
2. TCPIPService 定義では、この接続のインバウンド 属性を指定します。使用する TCPIPService は、IPCONN 定義の TCPIPService オプションによって指定します。

詳しくは、169 ページの『IP 相互接続リンクの定義』を参照してください。

ローカル CICS システムの命名

CICS Transaction Server for z/OS システムは、複数の名前によって認識されます。

- アプリケーション ID (applid)
- システム ID (sysidnt)
- VTAM 総称リソース名

すべての CICS システムには、applid と sysidnt 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、133 ページの『第 12 章 VTAM 総称リソースのインストールの注意点』を参照)。

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。

ISC over SNA 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 は一意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 を一意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、196 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、213 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。具体的に言えば、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 ISC over SNA 接続の場合、CONNECTION 定義に以下のように指定できます。

```
DEFINE CONNECTION(sysidnt) The local name for the remote system
      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。

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 (168 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』のページを参照)。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、168 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』のページ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

MRO リンクの定義を 165 ページの図 40 に示します。

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、165 ページの図 40 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 リソース定義ガイド」を参照してください。

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。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。

この章ですでに述べたように、**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。

```

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 40. 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 297 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・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、すで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。

MRO セッションの名前を CICS が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 (SESSIONS definition attributes)』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 リンクに対する 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済みます。しかし、セキュリティの点から見ると、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) を使用すると、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。

167 ページの図 41 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
  USEDFTUSER(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 41. MRO リンクの定義例

互換 MRO ノードの定義

MRO リンクは、接続される両方のシステムに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。

互換性要件を図 42 に示します。

CICSA			CICSB		
DFHSIT	TYPE=CSECT		DFHSIT	TYPE=CSECT	
	,APPLID=CICSA	1		,APPLID=CICSB	4
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 42. 互換 MRO ノードの定義

図 42 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 43 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 43. 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。

169 ページの図 44 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 44. 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、すでにある 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の詳細は、「*CICS Resource Definition Guide*」の『CONNECTION 定義の属性 (CONNECTION definition attributes)』を参照してください。

IP 相互接続リンクの定義

このセクションでは、IP 相互接続 (IPIC) を使用した分散プログラム・リンク (DPL) で使用する、リモートの CICS TS for z/OS、バージョン 3.2 (以降) 領域に接続を定義する方法について説明します。

注: DPL プログラムは、MRO、APPC、または IPIC による接続を使用して通信できます。静的ルーティングの DPL の場合、ターゲット領域は PROGRAM 定義の REMOTESYSTEM 名によって識別されます。REMOTESYSTEM には、IPCONN 定義または CONNECTION 定義の名前を指定できます。

IPIC 接続を作成するには、補完関係にある次の 2 つのリソースをインストールする必要があります。

1. IPCONN 定義では、TCP/IP 接続のアウトバウンド 属性を指定します。
2. TCPIPSERVICE 定義では、この接続のインバウンド 属性を指定します。使用する TCPIPSERVICE は、IPCONN 定義の TCPIPSERVICE オプションによって指定します。

IPCONN 定義の最重要属性は次のとおりです。

APPLID(*applid*)

リモート・システムのシステム初期設定テーブル (SIT) の **APPLID** パラメータに定義されているアプリケーション ID (*applid*) を指定します。

重複 APPLID については、以下の規則があります。

- 同一の APPLID と 同一の NETWORKID を指定した IPCONN 定義を 2 つ以上インストールすることはできません。(APPLID と NETWORKID の組み合わせは、ネットワーク全体で各システムに固有の名前を付ける目的に使用できます。後述の NETWORKID オプションの説明を参照してください。)
- インストール済みの MRO、APPC、または LUTYPE6.1 CONNECTION 定義の NETNAME と同じ APPLID を指定した IPCONN 定義をインストールすることができます。
- インストール済みの IPCONN 定義がインストール済みの CONNECTION 定義と同一の名前を持つ場合、IPCONN 定義の APPLID は CONNECTION 定義の NETNAME と一致していなければなりません。一致していない場合、状態に応じて以下のメッセージが出されます。
 - DFHIS3009 (IPCONN の自動インストール時にエラーを検出した場合)
 - DFHAM4913 (IPCONN のインストール時にエラーを検出した場合)
 - DFHZC6312 (CONNECTION のインストール時または自動インストール時にエラーを検出した場合)

IPCONN 定義は CONNECTION 定義に優先します。つまり、IPCONN と CONNECTION が同じ名前を持っている場合、CICS は IPCONN を使用します。

- 同じ NETNAME と APPLID を持つ CONNECTION と IPCONN が同じ名前を持つ必要はありません。

これによって、IPCONN がある場合に、IPCONN によってすべてのサポート機能をルーティングするという CICS のデフォルトを使用する代わりに、TCP/IP を介した通信で個別のシステム ID を使用できるようになります。

上述の規則は、インストール時に検証されます。

AUTOCONNECT(**{NO|YES}**)

IPCONN 定義をインストールするときにセッションを確立したい場合は **YES** を指定し、そうでない場合は **NO** を指定します。**YES** を指定する場合、接続が達成されるのは以下の IPCONN 定義をインストールしている場合であることを念頭に置いてください。

1. この IPCONN 定義の TCPIP SERVICE オプションで指定した TCPIP SERVICE 定義も、この領域にインストールし、PROTOCOL(IPIC) を指定する必要があります。
2. 対応する IPCONN 定義と TCPIP SERVICE 定義は、リモート領域にインストールする必要があります。「対応」とは、以下のことを意味します。
 - a. リモート領域の IPCONN 定義の HOST オプションでは、この領域を指定する必要があります。
 - b. リモート領域の IPCONN 定義の PORT オプションでは、この IPCONN で指定したローカル TCPIP SERVICE 定義の PORTNUMBER オプションに指定したものと同一ポート番号を指定する必要があります。
 - c. リモート領域の TCPIP SERVICE 定義 (リモート領域の IPCONN 定義で指定された) では、PROTOCOL(IPIC) を指定し、その PORTNUMBER オプションに、この IPCONN の PORT オプションで指定したものと同一ポート番号を指定する必要があります。

HOST(*hostname*)

ホスト名またはターゲット・システムのドット 10 進 IP アドレスを指定します。例えば、abc.example.com、または 9.20.183.7。

INSERVICE(**{NO|YES}**)

接続がインストールされている場合に使用可能にするには **YES** を指定し、使用不可能にするには **NO** を選択します。

IPCONN(*name*)

この IPCONN 定義の 8 文字の名前。4 文字の「ローカル名」(SYSID) を指定します。CICS はこれによってリモート・システムを認識します。残りには、4 つの末尾ブランクが埋め込まれます。

NETWORKID(*name*)

リモート・システムのネットワーク ID を指定します。(リモート・システムのネットワーク ID は、その VTAM NETID、またはその **UOWNETQL** システム初期設定パラメーターの値 (VTAM=NO システムの場合) のいずれかです。)

NETWORKID を指定しない場合、CICS はこの CICS (つまり、この定義がインストールされている CICS) の VTAM NETID、または **UOWNETQL** システム初期設定パラメーターの値を使用します。

同じアプリケーション ID を持つ 2 つ以上のリモート・システムに接続する場合は、**NETWORKID** を指定します。(CICS のアプリケーション ID はシスプレックス内では固有でなければなりませんが、例えば、シスプレックス外または異なるシスプレックスのシステムに接続することはできます。) **APPLID** と **NETWORKID** の組み合わせにより、リモート・システムが固有の名前によって参照されるようになります。

PORT(*port*)

リモート領域が listen するポートの 10 進数番号 (1 から 65535 までの範囲にある) を指定します。ポート番号と HOST 値を結合したものが、この IPCONN のアウトバウンド要求の宛先になります。

RECEIVECOUNT(*number*)

受信セッション (つまり、着信要求を受信するセッション) の数 (1 から 999 までの範囲) を指定します。使用される受信セッションの実際数は、リモート・

システムに定義されている送信セッション数にも左右されます。接続が確立すると、これらの値は交換されて低い方の値が使用されます。

SENDCOUNT(*number*)

送信セッション (つまり、発信要求を送信するセッション) の数 (1 から 999 までの範囲) を指定します。使用される送信セッションの実際数は、リモート・システムに定義されている受信セッション数にも左右されます。接続が確立すると、これらの値は交換されて低い方の値が使用されます。

TCPIPSERVICE(*name*)

この IPCONN 用のインバウンド処理の属性を定義する、PROTOCOL(IPIC) の TCPIPSERVICE 定義の名前を指定します。

図 45 は、典型的な IPCONN 定義を示しています。

```
DEFINE
  IPCONN(IP01)                接続のローカル名
  GROUP(groupname)           関連定義のグループ名
  HOST(aor001.widgets.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(IP01tcps)     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 45. IPIC による接続の IPCONN 定義の例

IPCONN 定義のすべての使用可能な属性に関する詳細な説明は、「*CICS Resource Definition Guide*」を参照してください。

IPICリンクによる DPL の場合、TCPIPSERVICE 定義の最重要属性は次のとおりです。

IPADDRESS(**{INADDR_ANY}***ipaddress*)

この TCPIPSERVICE が着信接続を listen するドット 10 進 IP アドレスを指定します。これは nnn.nnn.nnn.nnn という形式で、nnn は 0 から 255 である必要があります。これがリモート・システムの IPCONN 定義の HOST オプションに指定しなければならない IP アドレスです。

PORTNUMBER(*port*)

CICS が着信クライアント要求を listen するポートの 10 進数番号 (1 から 65535 までの範囲にある) を指定します。これがリモート・システムの IPCONN 定義の PORT オプションに指定しなければならない数値です。

PROTOCOL(**{ECI|HTTP|IOPI|IPIC|USER}**)

TCP/IP ポートで使用するアプリケーション・レベルのプロトコル。IPICリンクの場合は、IPIC を指定します。

SOCKETCLOSE(**{NO|}***hmmss*)

ソケットでの着信データの受け取りを発行してからそのソケットを閉じるまで CICS を待機させるかどうか、およびその場合の待機時間。IPICリンクの場合は、SOCKETCLOSE(NO) を指定する必要があります。

TCPIPSERVICE(*name*)

このサービスの 8 文字の名前。これは、対応する IPCONN 定義の TCPIPSERVICE オプションに指定しなければならない名前です。

TRANSACTION(*transaction*)

このサービスを受け取る新しい要求を処理するよう付加された、CICS トランザクションの 4 文字の ID。IPIC の場合は、CISS (または、プログラム DFHISCOP を実行する別のトランザクション) を指定します。

URM(*program*)

IPCONN 自動インストール・ユーザー・プログラムの名前を指定します (必要な場合)。この属性を指定しない場合、CICS は CICS 提供のデフォルトである、IPCONN 自動インストール・ユーザー・プログラム、DFHISAIP を使用します。

図 46 は、IPIC による接続の典型的な TCPIPSERVICE 定義を示しています。

DEFINE	
TCPIPSERVICE(IP01tcps)	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
IPADDRESS(9.20.185.7)	listen する IP アドレス
PORTNUMBER(1500)	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(DFHISAIP)	IPCONN 自動インストール・ユーザー・プログラム

図 46. IPIC による接続の TCPIPSERVICE 定義の例

次の図は、IPCONN 定義と TCPIPSERVICE 定義との関係を示しています。

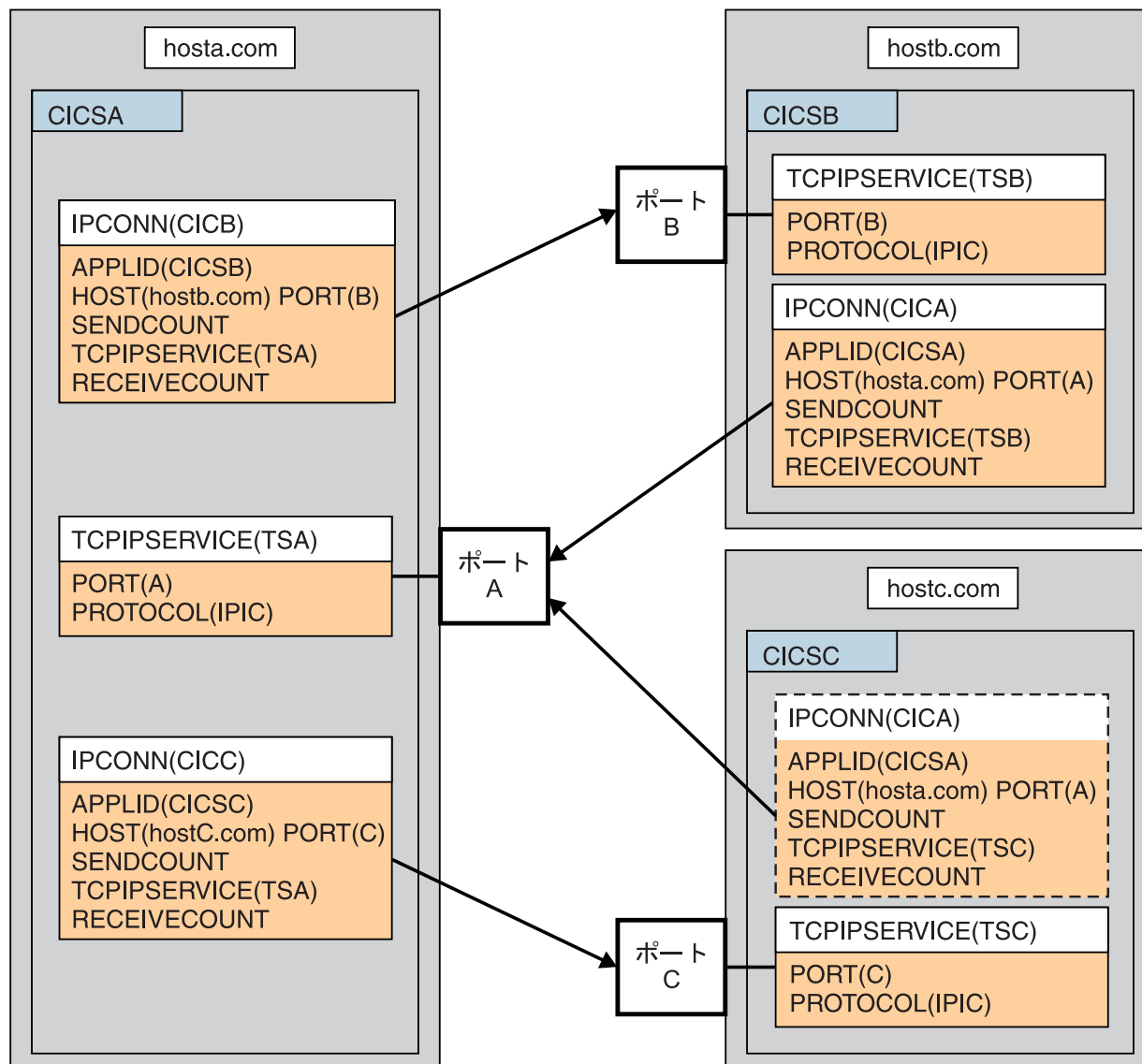


図 47. 関連した IPCONN 定義と TCPIP SERVICE 定義

TCPIP SERVICE 定義のすべての使用可能な属性に関する詳細な説明は、「CICS Resource Definition Guide」を参照してください。

APPC リンクの定義

APPC リンクは、複数のセッションからなる 1 つ以上の「セット」から構成されています。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。

注: 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、179 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 リンクを定義するには、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してはなりません。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・リンクを除くすべての APPC リンクで、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。(126 ページの『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

APPC システムの定義の形式を図 48 に示します。

```
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 48. 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、163 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。(196 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、133 ページの『第 12 章 VTAM 総称リソースのインストールの注意点』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、179 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 181 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 297 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 182 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 システムに対する各セッション・グループは、DEFINE SESSIONS コマンドで定義します。この定義を 177 ページの図 49 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 49. 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 181 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 182 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 178 ページの図 50 に示します。

CICSA			CICSB		
DFHSIT	TYPE=CSECT ,APPLID=CICSA	1	3	DFHSIT	TYPE=CSECT ,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 50. 互換 CICS APPC ISC ノードの定義

図 50 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. リンク的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、179 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えれば、実際には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べきです。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 (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 (Autoinstalling APPC connections)』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 カスタマイズ・ガイド*」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。CONNECTION と SESSIONS のペアを定義し、その接続に SINGLESESS(Y) を指定する方法か、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。

必要な DEFINE CONNECTION コマンドと DEFINE SESSIONS コマンドの形は、175 ページの図 48 と 177 ページの図 49 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、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 カスタマイズ・ガイド*」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 (179 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。障害のあった CICS が再始動されると、CICS 持続セッションのサポートによって、ネットワークの流れなしでセッションがリカバリーされます。CICS は、セッションを保持する期間を PSDINT システム初期設定パラメーターから判断します。したがって、持続セッションのサポートを使用する場合には、PSDINT に 0

より大きな値を指定する必要があります (そして、XRF システム初期設定パラメーターに 値 'NO' を指定します。持続セッションのサポートは XRF と相いれませんが)。障害のあった CICS が PSDINT の間隔内に再始動されると、保持されているセッションが直ちに使用されます (ネットワーク・フローに対してそれらを再パインドする必要はありません)。この間隔は、CEMT SET VTAM コマンドか EXEC CICS SET VTAM コマンドによって変更できます。

CICS が CEMT PERFORM SHUTDOWN IMMEDIATE によってシャットダウンされるか、障害が起こると、セッションは「リカバリー保留」状態に置かれます。緊急時再始動の際、CICS は、持続セッションとして定義されている APPC セッションを「セッション中」状態に復元します。

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 状態に戻される。

NONE

すべてのセッションは、CNOS のリカバリーなしにアンパインドされ、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、RECOVPTION 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、RECOVPTION 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、341 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

重要:

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。

LUTYPE6.1 リンクは、CICS と、LUTYPE6.1 プロトコルをサポートし、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。

DEFINE CONNECTION は、必ず、LUTYPE6.1 リンク上でリモート・システムを定義する必要があります。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、185 ページの図 51 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 51. 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、125 ページの『第 11 章 システム間通信のインストールの注意点』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 189 ページの図 52 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 (196 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(NOIO-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 297 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(NOIO-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 297 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(UIVB)

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 273 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
DEFINE CONNECTION(IMSR)	2	1	TERMINAL NAME=SYSCICS
GROUP(groupname)			SESSION=2
NETNAME(SYSIMS)	3		COMPT1
ACCESSMETHOD(VTAM)		6	COMPT2
PROTOCOL(LU61)			OUTBUF=mmm
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 52. 互換の CICS ノードと IMS ノードの定義

図 52 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、128 ページの図 33 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。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 (191 ページの図 53)。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 53. 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。

端末所有
領域 (TOR)

中間システム

アプリケーション所有
領域 (AOR)

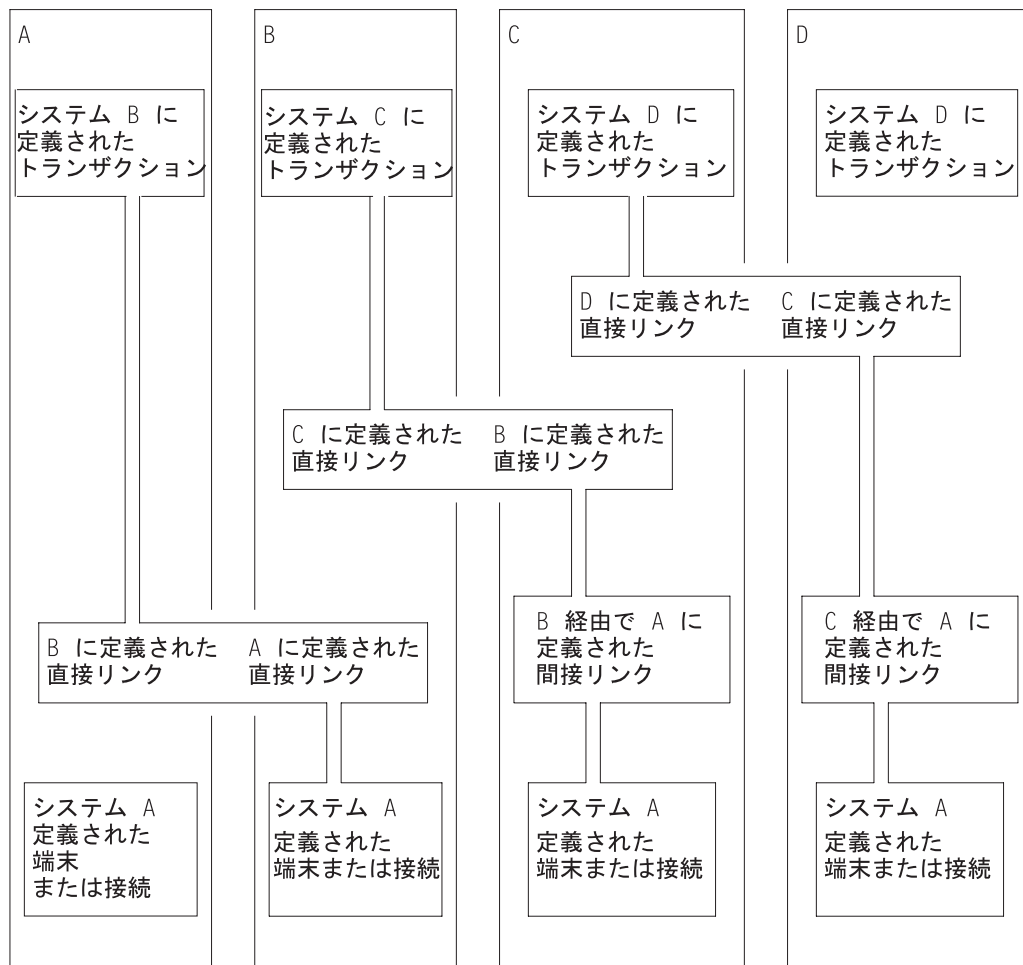


図 54. トランザクション・ルーティングのための間接リンク

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 で間接リンクの定義が必要になる理由

213 ページの『第 16 章 リモート・リソースの定義』で説明したように、CICS システムは、次のものから構成される固有の ID によってリモート端末を参照します。

- その端末専有領域の applid (ネット名)
- その端末が端末専有領域で認識されている ID

CICS が完全修飾の端末 ID を得るためには、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、192 ページの図 54 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ではありません。

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 へコピーしたいと思うでしょう。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 つ定義する方がおそらく速いでしょう。

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 195 ページの図 55 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り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 55. 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

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、213 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。

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。

195 ページの図 55 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、213 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。これらの名前は、APPLID (= 総称 applid、特定 applid) システム初期設定パラメーターに指定します。

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 システムは、総称 applid が同じで、特定 applid が異ならないからです。

注: 同じ総称 applid をもつ活動システムと代替システムは、sysidnt も同じでなければなりません。総称 applid と特定 applid について詳しくは、「*CICS/ESA 3.3 CICS XRF Guide*」を参照してください。

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。

XRF の場合、CICS システムの**総称 applid** は、APPLID システム初期設定パラメーターに定義されるもので、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 システムによって、CONNECTION 定義の NETNAME オプション、または IPCONN 定義の APPLID オプションに指定される名前です。)

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。

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。133 ページの『第 12 章 VTAM 総称リソースのインストールの注意点』を参照してください。

特に次の点に注意してください。

- 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のであれば、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 APPC リンクの管理

この章では、マスター端末トランザクション CEMT を使って APPC 接続を管理する方法について説明します。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|ACQUIREDvCLOSED

CEMT のコマンドの形式とオプションについての詳細は、「*CICS Supplied Transactions*」の『CEMT SET CONNECTION』を参照してください。

記載されている情報は、主に CICS システム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。

- 200 ページの『接続の獲得』
- 202 ページの『SET MODENAME コマンドによるセッションの制御』
- 204 ページの『接続の解放』
- 207 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 307 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、次のように 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は、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

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、181 ページの『AUTOCONNECT オプション』に説明があります。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。各モードグループは、独自の AUTOCONNECT オプションを持っており、このオプションを設定すると、201 ページの表 6 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 6. 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すで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すで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ができます。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 オプションは、次のものを指定します。

- そのモードグループに対してサポートできるセッションの最大数。
- 競合勝者としてサポートされるセッションの数。

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 7 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 7. 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、実際に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

- 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 8 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 8. 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、SETIINQ 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て、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

この接続はすで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すで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す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、接続がすで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 56. 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、次のいずれかの方法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、使用可能カウンタを 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 9 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 9. 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 9. 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 SETINQ 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、SETINQ CONNECTIONMODENAME 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 15 章 TCP/IP 管理および制御

TCP/IP 管理および制御を使用することにより、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。これにより、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。

注: 「TCP/IP ネットワーク」という場合、以下によって相互接続されているシステムを意味します。

- IPIC 接続 (IPCONN)。現在は、CICS TS 3.2 領域間と、CICS TS 3.2 領域と Java クライアントの間でのみ使用できます。
- 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すれば、以下のようなことができます。

- CICSplex SM または同等のツールを使用して、
 - 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のことを検査できます。
 - 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 管理および制御を使用する理由は、以下のとおりです。

- 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
- 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できません。)

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)。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) のユーザー関連データ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、ユーザー関連データは、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。資料については、『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できません。)

CICS ApplData

CICS TS 3.2 は、それが所有する各 TCP ソケットについてのアプリケーション固有の情報 (40 バイト) を生成します。CICS は SIOCSAPPLDATA ioctl 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。この新規の ioctl は、z/OS V1R7 の APAR PK32534 および z/OS V1R8 の APAR PK40411 で提供された新規機能に含まれています。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。これは、問題判別、キャパシティー・プランニング、および会計アプリケーションなどにおいて役に立ちます。

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

XAPADMGR 出口は、分散トランザクションで使います。これにより、分散トランザクションの発信時点で、タスクの関連データ発信元記述子へユーザー情報を追加することができます。この情報は、例えば、CICSplex SM により実行される処理のための検索キーとして後で使用可能です。

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。出口プログラムが使用可能になっている場合、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、プログラムが呼び出されます。

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性を提供します。
- 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

上述の 210 ページの『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ための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 Resource Tables Reference*」および「*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 から 371 は、TCP/IP 関係です。「*CICS パフォーマンス・ガイド*」を参照してください。

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』
- 214 ページの『リソースのローカル名とリモート名』
- 215 ページの『機能シップのためのリモート・リソースの定義』
- 219 ページの『DPL のリモート・リソースの定義』
- 222 ページの『非同期処理のためのリモート・リソースの定義』
- 223 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』
- 241 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が、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のことを言います。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) カ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・リソースの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・リソースの定義は必要ありません。⁹

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。

9. ただし、214 ページの『「デ이지ー・チェーン」に関する注記』を参照してください。

「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上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（104 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

CICS リソースは、通常、名前によって参照されます。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 などです。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、215 ページの図 57 に示すとおりです。

CICSA (ローカル・システム)		CICSB (リモート・システム)
DFHSIT TYPE= ,APPLID=CICSA	1	DFHSIT TYPE= 3 ,APPLID=
DEFINE CONNECTION(CICR) NETNAME(CICSB)	2 3	DEFINE CONNECTION(CICL) 1 NETNAME(CICSA)
DEFINE FILE(FILEA) REMOTESYSTEM(CICR)	4 2	4 DEFINE FILE(FILEA)
DEFINE FILE(FILEB)		
DEFINE FILE(local-name) REMOTESYSTEM(CICR) REMOTENAME(FILEB)	2 5	5 DEFINE FILE(FILEB)

図 57. ローカル・リソース名とリモート・リソース名

図 57 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 57 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 機能シップを使用する場合に定義しなければならない可能性のあるリモート・リソースは、次のとおりです。

- リモート・ファイル
- 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 58. 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 (32 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

ローカル CICS システムでそのファイルを識別する際に使用する名前を FILE オプションで指定します。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 (Sharing the CSD in non-RLS mode)』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 59. 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 60 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 60. 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 XTSERREQ 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。XTSERREQ および XTSERREQC 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『一時記憶域の 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・システムの名前を一時記憶域要求に明示的に指定せず、XTSREQ 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 61 に示します。

```
DFHTST  TYPE=REMOTE
        ,SYSIDNT=name
        ,DATAID=character-string
        [,RMTNAME=character-string]
```

図 61. 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 62 は、指定する必要があるプログラムの属性を示します。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 62. 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、次のようにします。

- 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

ローカル CICS システムでサーバー・プログラムが認識される名前は、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みラー・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が必要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、および区分セットの自動インストール (Autoinstalling programs, map sets, and partition sets)』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 (Writing a program to control autoinstall of APPC connections)』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いければ) 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は必要ないこともありますが、例えば、プログラムの REMOTENAME 属性や TRANSID 属性などを設定する場合には、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

非同期処理に必要な唯一のリモート・リソース定義は、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ものです。

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よう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、これを呼び出すのは、ローカル CICS システムからの START コマンドのみです。

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 223 ページの図 63 に示します。

```
DEFINE
  TRANSACTION(name)
  GROUP(groupname)
  リモート属性
  REMOTESYSTEM(sysidnt)
  REMOTENAME(name)
  LOCALQ(NO|YES)
```

図 63. 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、43 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 (『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。234 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

このセクションの情報は、端末関連のトランザクション・ルーティング、すなわち次のルーティングにのみ適用されます。

- ユーザー端末から開始されるトランザクション。
- 端末関連の ATI コマンドで開始されるトランザクション。

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 (端末専有領域) が所有する「端末」を、別の CICS システム (アプリケーション所有領域) が所有するトランザクションに接続することができます。端末専有領域とアプリケーション所有領域は、MRO リンクか APPC リンクで接続されていなければなりません。

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の両方を、どちらの CICS システムにも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。227 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「デジタイズ・チェーン」される場合も、上で述べた規則は有効です。さらに、端末とトランザクションの両方を、中間の CICS システムにリモート・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 (191 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

このセクションでは、リモート VTAM 端末を RDO を使って定義する方法を説明します。しかし、この端末をアプリケーション所有領域に定義する必要はありません。

ん。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。この方式については、227 ページの『端末定義と接続定義のシップ』で説明します。

リモート VTAM 端末は、DEFINE TERMINAL コマンドを使って定義します。このコマンドにおいて、

- 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 (232 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 64. 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、前のセクションの VTAM 端末の説明と同じように、TERMINAL 定義および TYPETERM 定義によって定義することができます。

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 226 ページの図 65 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 65. 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

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、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。

端末は、ローカル端末定義と同様に、DEFINE TERMINAL によって完全に定義する必要があり、対応する TYPETERM 定義が必要です。さらに、

- REMOTESYSNET オプションには、その端末専有領域のネット名を指定します。
- REMOTESYSTEM オプションには、その端末専有領域自体の sysidnt を指定します。

このような端末が端末専有領域にインストールされると、完全なローカル端末定義が作成されます。他のシステムには、リモート端末定義が作成されます。

同様に、APPC 接続は、DEFINE CONNECTION によって完全に定義する必要があり、1 つまたは複数の対応する SESSIONS 定義が必要です。さらに、REMOTESYSNET オプションにはその TOR のネット名を、REMOTESYSTEM オプションには TOR 自体の sysidnt をそれぞれ指定する必要があります。このような接続が、端末専有領域にインストールされると、完全なローカル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。

注: 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、同じ名前を指定しなければなりません。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

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へ必要に応じて端末定義がシッ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すで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、定義のシッを要求します。

シッされた端末定義は、接続されている CICS システムに、その接続を提供している ISC セッションまたは MRO 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可能です。特別な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することができます。シッ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 (191 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。シッされた定義の名称が、その領域にすでにインストールされているリモート端末からリモート接続の名称と衝突する場合には、シッされた定義に 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、233 ページの『端末の別名』のページに記載されています。) シッ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、シッされた定義に別名を割り当てる必要があります。シッ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: 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（249 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すでに受信されている場合には、その端末を ATI 要求で 사용할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 でユーザー出口 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、同じようなユーザー出口 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、72 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、77 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、79 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い、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ためには、それを、SHIPPABLE(YES) を指定した TYPETERM と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 66. 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、301 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 3 リリース 2 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 3 リリース 2 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 3 リリース 2 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 3 リリース 2 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) には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) には、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:

参照しやすいように DFHTCT TYPE=REMOTE マクロの形式を以下に示しておきます。

```
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={1istname|(aa[=nnn],bb[=nnn],cc[=nnn],...)]
[,SESTYPE=session-type]
[,VF={NO|YES}]
```

図 67. 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 (191 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:

TCAM 端末:

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 3 リリース 2 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。

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図 68 に示します。

Local System CICL AOR	Remote System CICR TOR
DFHSIT TYPE= SYSIDNT=CICL	DFHSIT TYPE= SYSIDNT=CICR
DFHTCT TYPE=INITIAL, ACCMETH=NONVTAM, SYSIDNT=CICL, : :	DFHTCT TYPE=INITIAL, ACCMETH=NONVTAM, SYSIDNT=CICR, : :
DFHTCT TYPE=REMOTE, SYSIDNT=CICR TRMIDNT=aaaa, TRMTYPE=3277, TRMMODL=2, ALTSCRN=(43,80) : :	DFHTCT TYPE=REMOTE, SYSIDNT=CICR TRMIDNT=aaaa, TRMTYPE=3277, TRMMODL=2, ALTSCRN=(43,80) : :
DFHTCT TYPE=FINAL	DFHTCT TYPE=FINAL

図 68. DFHTCT TYPE=REMOTE マクロの典型的な使用

図 68 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（191 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 69 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, SYSIDNT=TERM, ACCMETH=NONVTAM .	DFHTCT TYPE=INITIAL, SYSIDNT=TRAN, ACCMETH=NONVTAM .
	DFHTCT TYPE=REGION, SYSIDNT=TERM
COPY TERMDEFS	COPY TERMDEFS
	DFHTCT TYPE=REGION, SYSIDNT=LOCAL
DFHTCT TYPE=FINAL	DFHTCT TYPE=FINAL

図 69. DFHTCT TYPE=REGION マクロの典型的な使用法

図 69 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・ルーティングに關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 REMOTESYSTEM(PD1) . .	DEFINE CONNECTION(PD1) NETNAME(PRODSYS) . .	TOR への 直接リンク
VTAM 端末定義 DEFINE TERMINAL REMOTESYSTEM(NEXT) REMOTESYSNET(PRODSYS) . .	DEFINE CONNECTION(NEXT) NETNAME(INTER1) . .	TOR への 非直接リンク
非 VTAM 端末定義 (メソッド 1) DFHTCT TYPE=REMOTE, SYSIDNT=PD1, . .	DEFINE CONNECTION(PD1) NETNAME(PRODSYS) . .	TOR への 直接リンク
非 VTAM 端末定義 (メソッド 2) DFHTCT TYPE=REGION, SYSIDNT=PD1 . .	DEFINE CONNECTION(PD1) NETNAME(PRODSYS) . .	TOR への 直接リンク
非 VTAM 端末定義 (メソッド 1) DFHTCT TYPE=REMOTE, SYSIDNT=REMT, . . DFHTCT TYPE=TERMINAL, .	DEFINE CONNECTION(REMT) NETNAME(PRODSYS) ACCESSMETHOD(INDIRECT) INDSYS(NEXT) . .	TOR への 非直接リンク

図 70. 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 71. 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 235 ページの図 72 に示します。


```

DEFINE
  TRANSACTION(name)
  GROUP(groupname)
  PROGRAM(name)
  TWASIZE(0|value)
  PROFILE(DFHICIST|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(DFHICSS|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
  RESSEC(NO|YES)
  CMDSEC(NO|YES)
  EXTSEC(NO|YES)
  TRANSEC(01|value)
  RSL(00|value|Public)

```

図 72. 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。¹⁰ 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。

10. 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・ルーティングといます。¹¹

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。61 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。資料については、79 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の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、243 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。

11. 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 (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、297 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이지・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 (79 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이지・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 (79 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と、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: この方式は、237 ページの『静的トランザクション・ルーティング』ですでに説明しました。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。

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: 次のトランザクションでは、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・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します。この場合、各領域は、要求側 / 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。領域は、CSD を共用可能でも共用不可でも構いません。「*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 を共有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。これにより、メッセージ DFHAC2001 - 「Transaction 'transid' is unrecognized (トランザクション transid は認識されませんでした)」がユーザー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 73 に示します。

```
DEFINE
  TRANSACTION(CRTX)
  GROUP(DFHISC)
  PROGRAM(#####)
  TWASIZE(00000)
  PROFILE(DFHICIST)
  STATUS(ENABLED)
  TASKDATALOC(ANY)
  TASKDATAKEY(CICS)
リモート属性
  DYNAMIC(YES)
  REMOTESYSTEM()
  REMOTENAME()
  TRPROF(DFHICISS)
  ROUTABLE(NO)
リカバリー
  DTIMOUT(NO)
  INDOUBT(BACKOUT)
  RESTART(NO)
  SPURGE(YES)
  TPURGE(YES)
```

図 73. 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・システムヘルレーティングされず、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 (79 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。214 ページの『「デ이지ー・チェーン」に関する注記』を参照してください。)

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、コマンドの構文上、参照されるリモート・パートナーごとに PARTNER 定義を作成することが必要になります。

242 ページの図 74 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 74. 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。これについては、「*CICS Resource Definition Guide*」の『PARTNER リソースの定義 (PARTNER resource definitions)』および「*SAA Common Programming Interface Communications Reference*」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』
- 246 ページの『体系化プロセス』
- 248 ページの『インストールが必要なリソース定義の選択』
- 249 ページの『区画内一時データ・キューの定義』
- 252 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 244 ページの図 75 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 (PROFILE definition attributes)』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` 条件が起ります。

244 ページの図 75 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
  [NEPCCLASS(0|value)]
  [RTIMOUT(NO|value)]
```

図 75. 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い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 (CSPG - page retrieval)』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レクエストか DPL レクエストを出したとき、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かもしれません。『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・プロファイルは、CEDA トランザクションによって修正することができます。

修正の典型的な理由の 1 つに、モード名を含めることで、例えば APPC リンクでの機能シップレクエストでサービス・クラスを選択できるようにするというものがあります。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 - これは、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を処理するためのものです (通常、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 76 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 76. CICS 体系化プロセス名

体系化プロセス定義の修正

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。

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。

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。（37 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 (CICS-supplied resource definitions, groups, and lists)』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 249 ページの図 77 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
-	CMXP	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 77. 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 78. 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 (Defining TDQUEUE resources)』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

(65 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、109 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 (65 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。『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、特別な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なければなりません。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、および区分セットの自動インストール (Autoinstalling programs, map sets, and partition sets)』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

本書のこのパートでは、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。システム間環境での CICS アプリケーション・プログラミングについて概説してから、以下のそれぞれの CICS 相互通信機能に関するプログラミングについて説明します。

- 機能シップ
- 分散プログラム・リンク (DPL)
- 非同期処理
- トランザクション・ルーティング
- CICS-IMS 間アプリケーション

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。

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。これらの新しい条件は、相互通信のコンポーネントが、必要なシステムとセッションを確立できないことに起因します (例えば、そのシステムが CICS に定義されていないか使用できない、セッションが失敗した、など)。

また、一部のタイプの要求によって、無効なデータが CICS 機能管理 (例えば、ファイル制御プログラム) に渡されるため、トランザクション異常終了が起こる場合があります。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、実際にローカル・システムと同じプロセッサに常駐させて、両方のシステムのトランザクション・ダンプと対応するトレース・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』
- 258 ページの『ファイル制御』
- 258 ページの『DL/I』
- 258 ページの『一時記憶域』
- 259 ページの『一時データ』
- 259 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。プログラムは、PL/I、C、COBOL、またはアセンブラ言語で作成することができます。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、267 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、31 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ます。

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ります。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、リモートの CICS OS/390 システムに関連した IMS/ESADDM または IMS/VS DB データベース、あるいはリモートの CICS/VSE システムに関連した DL/I DOS/VS 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 3 リリース 2 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 DL/I データベースの定義は、システム・プログラマーによって提供されます。CICS アプリケーション・プログラムには、特定のシステムを選択する機能はありません。

リモート CICS システムに機能シップできるのは、DL/I 要求のサブセットだけです。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ず。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的データ・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起こる可能性がある、コマンドの例外条件のいずれかが起こることがあります。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、次の場合にアプリケーション・プログラムで発生します。

- 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 および XISCONA の各出口に関するプログラミング情報は、「CICS Customization Guide」の『システム間通信プログラムの出口 XISCONA および XISLCLQ』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)。

こういった状態では、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 ATNI (ISC 接続の場合) または AZI6 (MRO 接続の場合) が出されます。実際のエラー条件は、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 (これは、リソースがローカルであれば可能な場合があります)。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』
- 262 ページの『サーバー・プログラム』
- 263 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 (例えば、CICS Transaction Server for Windows でも、CICS 6000 システムでも構いません)。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、またはアセンブラ言語で作成することができます。

97 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、コーディング方法は、そのサーバー・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

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、97 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起こる可能性のある、コマンドの例外条件のいずれかが起こることがあります。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、259 ページの『リモート・システム使用不能』のページ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すで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 (Mixing DPL and function shipping to the same CICS system)』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

ミラー・プログラム (サーバー・プログラムではなく) が異常終了するか、サーバー領域とのセッションで障害が起こると、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、43 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。

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 (259 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 (49 ページの『START コマンドのローカル・キューイング』を参照)。

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。

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利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。

注意すべき事項

プログラムは、PL/I、COBOL、C、またはアセンブラ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。

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のであります。

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。

AOR での EXEC CICS ASSIGN コマンドの使用

EXEC CICS ASSIGN コマンドの、次の 2 つのオプションによって、予期しない値が返される場合があります。

PRINSYSID

このオプションは、トランザクションの基本機能の SYSID を返します。返される値は、このシステムに定義されたり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、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 条件が起こります。

USERID

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 をとります。「CICS RACF Security Guide」の『LU6.2 でのトランザクション・ルーティングのセキュリティー (Transaction routing security with LU6.2)』を参照してください。

271 ページの表 10 で示すように、CICS は次の値を返します。

- 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 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、次の値を返します。
 - ISC 接続の場合、次のどちらかの値。
 1. セッション定義の USERID (これが指定されている場合)
 2. 接続定義の SECURITYNAME 値
 - MRO 接続の場合、TOR の RACF ユーザー ID
- 接続が ATTACHSEC(LOCAL) オプションによって定義されていて、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 10. 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

TOR の DFHSIT SEC=	CONNECTION 定義の ATTACHSEC 値		
	IDENTIFY VERIFY PERSISTENT MIXIDPE	LOCAL	
		AOR の DFHSIT SEC=YES または MIGRATE	AOR の DFHSIT SEC=NO
YES または MIGRATE	接続時に送られたユーザー ID	<u>ISC</u> 1. セッションの USERID 2. 接続の SECURITYNAME	AOR の DFLTUSER
NO	接続時に送られたユーザー ID (TOR の DFLTUSER)	<u>MRO</u> TOR の RACF ユーザー ID	

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』
- 275 ページの『CICS-IMS 間アプリケーション - 非同期処理』
- 281 ページの『CICS-IMS 間アプリケーションの 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、274 ページの図 79 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ではなく、システム間通信の制御に直接関与することはありません。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

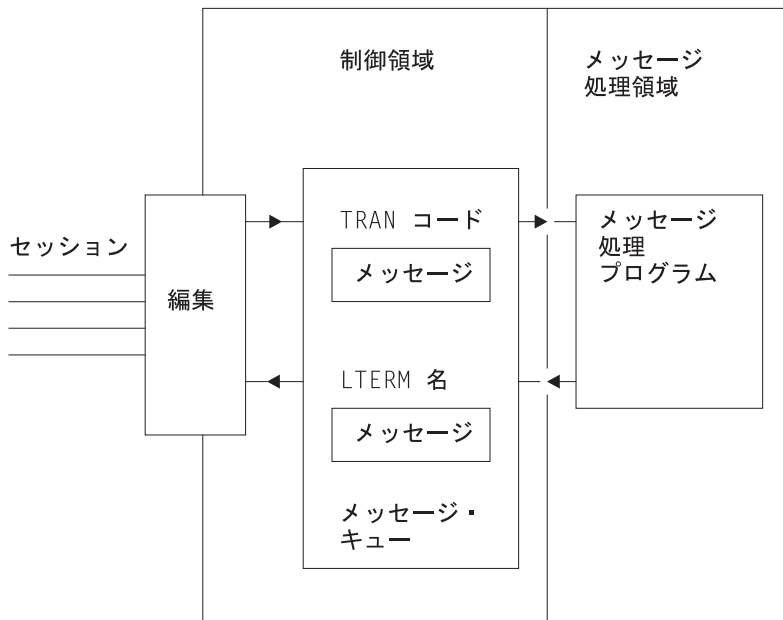


図 79. 基本的な IMS メッセージ・キュー化

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

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、43 ページの『第 5 章 非同期処理』と 109 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。

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。

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へ、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

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。これらのコマンドの適切な形式については、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、この項の後半で示します。

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 80 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。

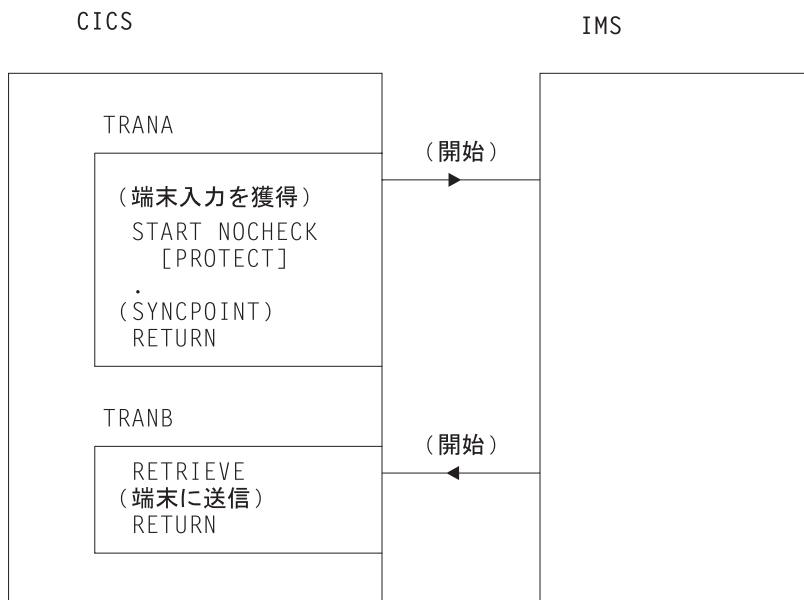


図 80. START と RETRIEVE の非同期処理 - CICS フロントエンド

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 (48 ページの『NOCHECK オプションを指定した START 要求の据え置き送信』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を受信されると、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、278 ページの図 81 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。

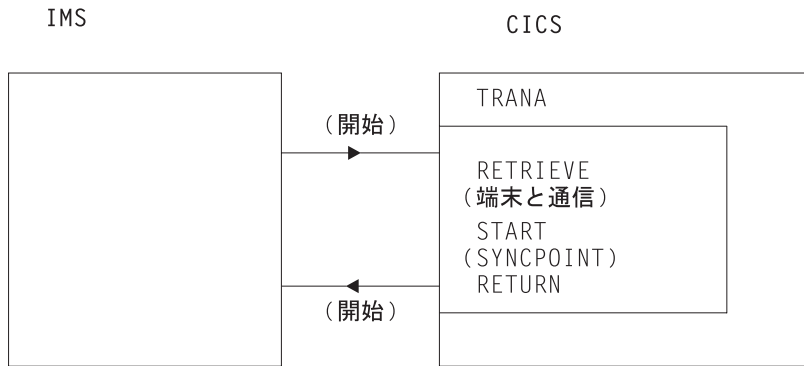


図 81. RETRIEVE/START 非同期処理 - IMS フロントエンド

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)、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 (161 ページの『第 13 章 リモート・システムへのリンクの定義』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、図 82 に示すとおりです。

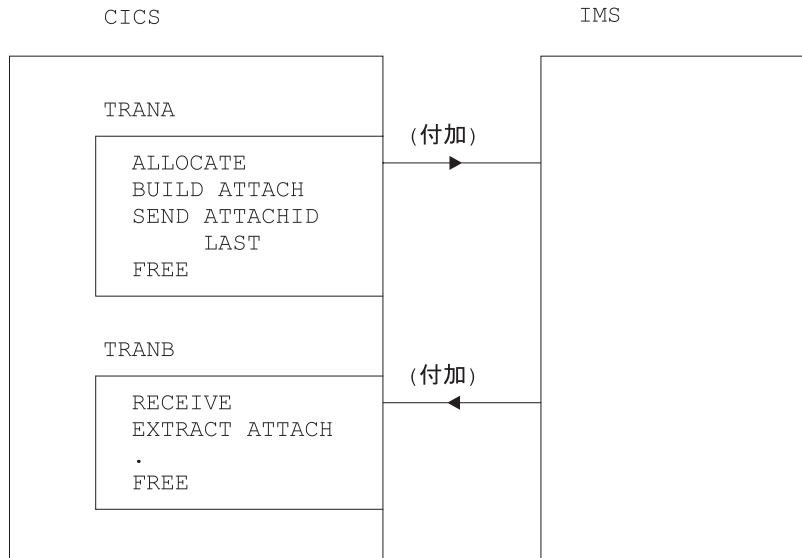


図 82. SEND/RECEIVE 非同期処理 - CICS フロントエンド

CICS-IMS 間アプリケーションの 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

CICS-IMS 間セッションを獲得して使用するために使用できるコマンドは、次のとおりです。

- **ALLOCATE** - リモート IMS システムへのセッションを獲得するために使用されます。
- **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 トランザクションによって、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ために使用されます。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ために使用されます。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/RECEIVE 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すで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|BASICEDT|name)]
  [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、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 ATTPDN 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 (ATTRDPN) 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 (ATTRPRN) 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あればなりません。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 から 7 | X'00' - ゼロでなければならない |
| 8 から 15 | X'00' - 複数の 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 から 7	X'00' – ゼロでなければならない
8 から 11	0000 – (ユーザー定義のデータ・ストリーム)
12 から 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 から 7	X'00' – 予約済み – ゼロでなければならない
8 から 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 から 7	X'00' - (ゼロ)
8 から 11	0000 - (ユーザー定義のデータ・ストリーム)
12 から 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 から 7	X'00' - (ゼロ)
--------	--------------

8 から 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 名を返します。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、送信側がどのような意図でいるのか分かるまで、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)
```

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。

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すで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)

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 (EXEC interface block)』を参照してください。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。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、291 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

次の EIB フィールドを使用すると、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。

EIBSRCE

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・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、291 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、243 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・トランザクションの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 後の開放保留状態
- 状態 7 - セッション開放

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、セッションがすで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・トランザクションヘ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れば、有効なコマンド順序の作成が容易になります。各表は、上記で定義した特定の状態に関連していて、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 11. 状態 1 – セッション未割り振り

状態 1 – CICS-IMS 間会話 – 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 12. 状態 2 – 送信状態

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		2
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、284 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 13. 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 14. 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 15. 状態 5 – 受信側による同期点

状態 5 – CICS-IMS 間会話 – 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 16. 状態 6 – SEND LAST 後の解放保留状態

状態 6 – CICS-IMS 間会話 – 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 17. 状態 7 – セッション解放

状態 7 – CICS-IMS 間会話 – 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

本書のこのパートでは、複数システム環境における CICS のパフォーマンスを向上させるためのヒントを示します。以下を実行する方法について説明します。

- システム間キューの長さの制御
- 重複しているシップ端末定義の、アプリケーション所有領域と中間システムからの削除

注: CICS パフォーマンスの概要については、「*CICS パフォーマンス・ガイド*」を参照してください。

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。機能シッパされた 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、49 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、キューが起きます。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パ要求は、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、この状態は、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ということです。

割り振りキューの管理

下記の各項では、割り振りキューを管理するための 3 つの方式について説明します。

接続定義だけの使用

単純な制御要件しかないようなシステム間リンク（例えば、通信量がクリティカルでないようなリンク）では、CONNECTION リソース定義に QUEUELIMIT と MAXQTIME オプションが指定できます。

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。これには、0 (要求を一切キューイングしない) から 9999 の数字が指定できます。または、キューの長さがどうであれ、必要ならすべての要求をキューイングするように指定することもできます。

MAXQTIME は、応答がないように見える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。この値は、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。この時間には、0 (つまり、割り振り要求を受信することでキューイング限度を超えてしまうような場合には、すぐにそのキューを除去する) から 9999 秒が指定できます。または、必要なだけ要求をキューに留めるように指定することもできます。

割り振り要求を受信することで QUEUELIMIT 値を超えてしまうような場合には、キューの処理速度に基づいて、新しい要求を満足するのに最大待ち時間より長い時間が必要かどうかを CICS が計算します。それが必要な場合は、CICS がそのキューを除去します。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。

注: 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、アプリケーション・プログラムに SYSIDERR 状態が戻されます。

CEDA DEFINE CONNECTION コマンドの QUEUELIMIT オプションおよび MAXQTIME オプションについては、「*CICS Resource Definition Guide*」の『CONNECTION 定義の属性』を参照してください。

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 NOQUEUEINOSUSPEND 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。

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、XZIQUE グローバル・ユーザー出口プログラムでも制御できます。この方法には、単に接続のキュー限界を設定するよりも柔軟性があります。

XZIQUE 出口を使用すれば、キューイングの問題 (ボトルネック) を早い段階で見つけることができます。この出口は、XISCONA グローバル・ユーザー出口によって提供される機能を拡張したもので、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) の場合しか呼び出されません。一方、XZIQUE は、機能シップおよび DPL の他に、トラン

ザクシオン・ルーティング、非同期処理、分散トランザクシオン処理の各要求でも呼び出されます。XZIQUE は、XISCONA の場合よりも詳細な情報を受信し、それに基づいて判断を行います。

XZIQUE を使用すれば、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。さらに、ボトルネックのある接続を終了してから、それを再び確立することができます。

XISCONA 出口との対話

XZIQUE と XISCONA のグローバル・ユーザー出口の間で対話することはありません。両方の出口を使用可能にすると、両方とも 機能シップや DPL 要求で呼び出される可能性があるため、この方法はお勧めしません。これらの出口のうちの 1 つだけが使用可能となるようにしてください。機能性と柔軟性が強化されていますので、XISCONA ではなく XZIQUE を使用することをお勧めします。

XISCONA グローバル・ユーザー出口プログラムがすでにある場合には、おそらくこの出口を修正すれば、XZIQUE 出口点で 사용할ことができます。

XZIQUE 出口が呼び出される時点

XZIQUE グローバル・ユーザー出口は、使用可能になっていれば、次の時点で呼び出されます。

- CICS がリモート・システムとのセッションを獲得しようとしたが、使用可能な空きセッションがないとき。この出口は、CONNECTION 定義に QUEUELIMIT オプションが指定されているかどうかや、その限度を超えているかどうかにかかわらず、呼び出されます。割り振り要求に NOQUEUE か NOSUSPEND が指定されている場合には、呼び出されません。

セッションに対する要求は、DTP プログラムによって出された明示的な EXEC CICS ALLOCATE コマンドや、トランザクシオン・ルーティングや機能シップの要求など、いくつかの方法で行われます。

- 接続のキューが出口プログラムの前の呼び出しによって除去されたあとに、割り振り要求が空きセッションを見つけることができた場合。この場合には、CICS が通常どおり処理を続け、必要ならキューイングを再開するように、出口プログラムで指定することができます。

XZIQUE 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になっていれば、XZIQUE 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、不特定の割り振り要求のデータ用か、特定のモードグループのデータ用に構造化されます（どちらになるかは、そのセッション要求による）。不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、APPC の各セッションに対するものです。

パラメーター・リストで渡される情報を使用すれば、グローバル・ユーザー出口プログラムは、CICS によって次のことを行うべきかどうかを決めることができます。

- 割り振り要求をキューイングする（キュー限界にまだ達していないときだけ可能）。

- 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。

出口プログラムは、例えば、次の要素に基づいて決定を行うことができます。

- 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、その要求をキューイングすることができます。
- セッションがその接続に割り振られる速度。キュー限界にすでに達していても、セッションの割り振りが十分速ければ、現在の要求だけを拒否することもできます。キュー限界にすでに達していて、セッションの割り振りが遅くて十分でない場合には、キュー全体を除去することもできます。

`XZIQUE` パラメーター・リストで渡される情報の詳細と、`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*」の『VTAM 実効ページ・セット・モジュールの出口 (VTAM working-set module exits)』に記載された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』
- 302 ページの『タイムアウト削除の実装』
- 303 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 デバイスカシステムです。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、どこかの段階で、AOR (および TOR と AOR¹² の間の中間システム) から削除する必要があります。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すでにある場合は、シップされた定義のトークンが TOR によっ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。

12. 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」と呼ぶことにします。

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={020000lhhmmss}

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、先行ゼロが入られます。

DSHIPINT={120000l0lhhmmss}

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、シッパ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、シッパ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パ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 DELETSHIPED と PERFORM DELETSHIPED 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

本書のこのパートでは、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。以下のトピックを取り上げています。

- 個々のセッション障害
- システム障害と再始動
- CICS 拡張回復機能 (XRF) の相互通信に影響がある部分
- VTAM 持続セッション用 CICS サポートの相互通信に影響がある部分

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ついて、特に相互通信環境に当てはまる点について説明します。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』
- 308 ページの『同期点交換』
- 311 ページの『リカバリーの機能とインターフェース』
- 316 ページの『初期始動とコールド・スタート』
- 319 ページの『接続定義の管理』
- 321 ページの『棚上げを完全にはサポートしない接続』
- 324 ページの『APPC 接続静止処理』
- 324 ページの『問題判別』

用語

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。

#

- 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、すなわち、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。

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。

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。
- 311 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 ISC over SNA (APPC) 並列セッション接続だけに適用されます。
- 316 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。
- 319 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ISC over SNA (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。
- 321 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 ISC over SNA (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。
- 324 ページの『問題判別』では、通信の障害やリカバリーで CICS が出すメッセージと、未確定や再同期の障害を解決する例を説明します。

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、受注入カ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 (308 ページ) と 2 (308 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 310 ページの図 83 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あると、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あった場合、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。

図 310 ページの図 83 から図 311 ページの図 85 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。同期点アクティビティーを開始する (例えば、EXEC CICS SYNCPOINT コマンドや EXEC CICS RETURN コマンドによって) ローカル UOW のことを**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 310 ページの図 83 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、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、コミットかバックアウトかのコーディネーターによる決定に従わなければならないため、**従属**と呼ばれます。

固有のセッション

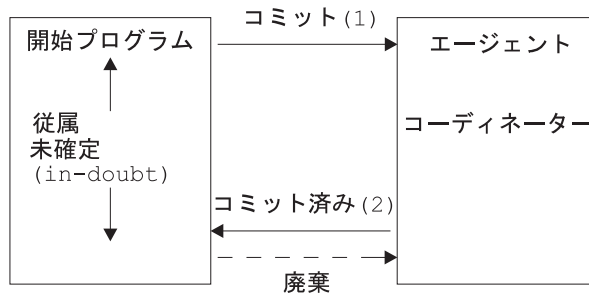


図 83. 同期点処理フロー — 固有セッション： この分散 UOW には、1 つのコーディネーターと 1 つの従属があります。コーディネーターは未確定の状態ではありません。

図 84 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 84 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。

チェーン・セッション — エージェント UOW が独自のエージェントを所有する

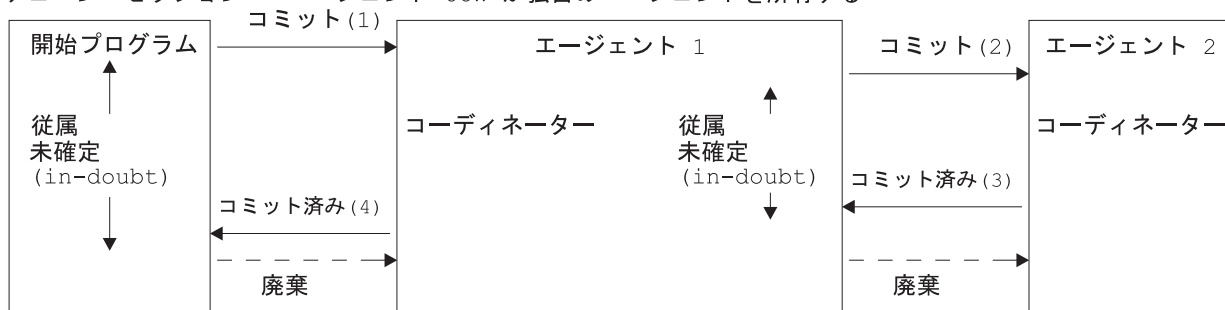


図 84. 同期点処理フロー — チェーン・セッション： この分散 UOW では、エージェント 1 は開始プログラムのコーディネーターであると同時に、エージェント 2 の従属です。

311 ページの図 85 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 85 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

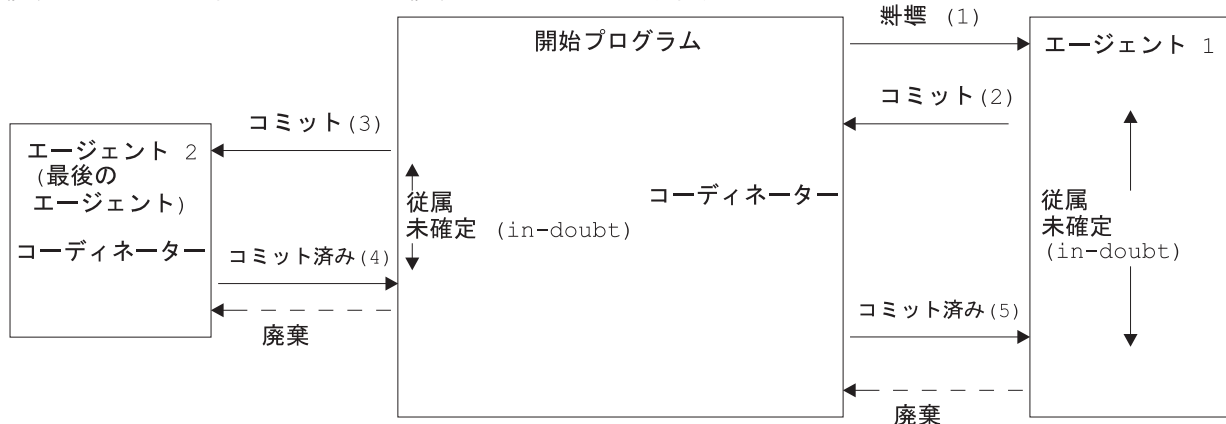


図 85. 同期点処理フロー - 複数セッション: この分散 UOW では、開始プログラムは、エージェント 1 のコーディネーターであると同時に、エージェント 2 の従属です。エージェント 2 は最後のエージェントですので、コミットの準備の通知は受けません。

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 ISC over SNA (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、321 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、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、316 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されると、CICS は、2 つのアクションのどちらかを行います (どちらのアクションをとるかについては、『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 (324 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が不整合にな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のため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、CICS 提供のトランザクションを要約して説明します。リソースの定義、システム・プログラミング・コマンド、CEMT トランザクションの詳細については、それぞれ、「CICS Transaction Server for z/OS Resource Definition Guide」(「*CICS Resource Definition Guide*」)、 「CICS Transaction Server for z/OS System Programming Reference」(「*CICS System Programming Reference*」)、および「CICS Transaction Server for z/OS Supplied Transactions」(「*CICS 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 がどのアクションをとるか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、未確定期間に入ったあとで起こった障害のリカバリー処理を待ってから、ACTION で指定されたアクションを取るべき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めます。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 定義の属性 (TRANSACTION definition attributes)』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取りべき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、おそらく、トランザクションがリモート・システムのリソースに対しどのような変更を行うかによって異なります。『未確定属性の指定の例』を参照してください。

未確定属性の指定の例: トランザクションの未確定属性を指定するケースとして、次の簡単な例をみてみます。

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。これによって、同期点処理の未確定期間に中のシステムやセッションの障害を除いたすべてのケースで、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 ISC over SNA (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、318 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 IPCONN XLNSTATUS は常に NOTAPPLIC を返します。

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 値を返す)、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 とネット名を返します。

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 333 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) を返します。UOW リンクごとに、INQUIRE UOWLINK は次の情報を返します。

- 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、328 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す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 ISC over SNA (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、318 ページの『ログ名交換プロセス』と 319 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 ISC over SNA (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、321 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・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・スタートを実行してはなりません。

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。これが行われない場合には (ま

たは、CEDA か GRPLIST でインストールされた定義がない場合には)、何らかのアクションが必要です。これについては、319 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。資料については、320 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・プロトコルがすで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 へマイグレーションするときに、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

重要:

| 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。他のタイプの接続に関する考慮事項につ
いては、321 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

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には、CEMT INQUIRE UOWLINK RESYNCSTATUS(UNCONNECTED) コマンドを使って、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。
- その対応付けられた 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。)

接続がすでに廃棄されている場合にも、これらのコマンドが使用できます。

注: 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。ある場合には、パートナーとの正常な再同期を確立できる可能性がないときだけその接続を廃棄してください。この例外的な状況では、この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 IPIC 接続

IPICによる接続は、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。

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ります。これを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。たいていの場合は、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。324 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・システムからすで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、146 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ければ、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

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だけ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。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。

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 セッションは、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 は、LU6.1 セッションが関係している場合は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格納されるわけではありません。LU6.1 接続定義には、リカバリーで使用する順序番号が入っています。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システムがあります。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る (インバウンドまたはアウトバウンド) と、CICS は作業単位の結果を通知します (「*CICS Family: Communicating from CICS on zSeries*」の『同期点指定 (LU 6.2) (Syncpointing (LU 6.2))』に説明があります)。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、前述の手順に従って、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

この項では、次の項目について説明します。

- CICS のリカバリー・アクションを知らせるメッセージ
- 未確定と再同期の失敗をどのように解決するか。この例では、すでに述べたコマンドのうち一部のものについてその使用方法を示します。

CICS のリカバリー・アクションを知らせるメッセージ

通信障害があると、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、あるいは緊急再始動時に表示されます。

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の障害とリカバリーに関連するメッセージを 3 つの図で示します。表 18 および 326 ページの表 19 には、UOW のコーディネーターとの接続が切れたときに生成されるメッセージを示します。表 18 には、WAIT(YES) がトランザクション定義に指定されて、棚上げが可能な場合に生成されるメッセージを示し、326 ページの表 19 には、WAIT(NO) が指定されて、棚上げが不可能な場合に生成されるメッセージを示します。327 ページの表 20 には、UOW の従属との接続が切れたときに生成されるメッセージを示します。詳細については、「*CICS Messages and Codes*」を参照してください。

表 18. セッション障害メッセージ：障害が起こったのは UOW のコーディネーターとのセッションです。WAIT(YES) がトランザクション定義に指定され、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)。

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	システムに障害が発生した、またはシステムが再始動した。	—	—
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。
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。

表 18. セッション障害メッセージ (続き): 障害が起こったのは UOW のコーディネーターとのセッションです。 WAIT(YES) がトランザクション定義に指定され、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)。

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	SET CONNECTION NOTPENDING ¢ または 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。 *	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: • * LU6.1 のみ • ¢ MRO、IPIC ??????????????????、および APPC のみ • \$ APPC および IPIC ?????????????????? のみ			

表 19. セッション障害メッセージ: 障害が起こったのは UOW のコーディネーターとのセッションです。 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)。

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	システムに障害が発生した、またはシステムが再始動した。	—	—

表 19. セッション障害メッセージ (続き): 障害が起こったのは UOW のコーディネーターとのセッションです。 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)。

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	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 ¢ または 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。 *	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: <ul style="list-style-type: none"> * LU6.1 のみ ¢ MRO、IPIC ??????????????????, および APPC のみ \$ APPC および IPIC ?????????????????? のみ 			

表 20. セッション障害メッセージ: 障害が起こったのは UOW における従属とのセッションです。

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)。

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。	—	—
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	システムに障害が発生した、またはシステムが再始動した。	—	—

表 20. セッション障害メッセージ (続き) : 障害が起こったのは UOW における従属とのセッションです。

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)。

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	セッション・リカバリーが正常に行われた。	DFHRM0135 DFHRM0148 +	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 ¢ または 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。 *	DFHRM0114 DFHRM0117 DFHRM0120 DFHRM0123	システム間セッション・リカバリー・エラー。リソース更新は同期化されない可能性がある。
キー: <ul style="list-style-type: none"> * LU6.1 のみ ¢ MRO、IPIC ???????????????????、および APPC のみ \$ APPC および IPIC ??????????????????? のみ + 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。 			

問題判別の例

この項では、未確定と再同期の失敗をどのように解決するか 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.IY LX1.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.IY LX1.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、同期点処理がすでに延期されており、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

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(00000005)
    Age(00003191)                               Use(CICSUSER)
Uow(AB1DD0122276C201) Inf Act Tra(CSSY) Tas(00000006)
    Age(00003191)                               Use(CICSUSER)
Uow(AB1DD01248A7B005) Inf Act Tra(CSTP) Tas(00000008)
    Age(00003191)                               Use(CICSUSER)
Uow(AB1DD9057B8DD800) Inf Act Tra(CSNC) Tas(00000018)
    Age(00000789)                               Use(CICSUSER)
Uow(AB1DF0805E76B400) Com Wai Tra(CSM3) Tas(00000079)
    Age(00003003) Ter(-AC3) Netn(IYM51 ) Use(CICSUSER) Wai
Uow(AB1DF0B2FDD36400) Inf Act Tra(CSNE) Tas(00000019)
    Age(00003024)                               Use(CICSUSER)
Uow(AB1DF15502238000) Inf Act Tra(CEMT) Tas(00000086)
    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(00000079)
    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(00000079)
Age(00003003)
Termid(-AC3)
Netname(IYM51 )
Userid(CICSUSER)
Waitcause(Waitforget)
Link( )
Sysid( )
Netuowid(..GBIBMIYA.IGCS255 .0.....)
```

UOW AB1DF0805E76B400 は、機能シップで使用するミラー・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、またデータの整合性が損なわれてもしかたがないと考える場合には、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に失敗したすべての 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 (未確定の状態に失敗) 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 86. CEMT INQUIRE CONNECTION - システム IYLX1 が所有する接続

システム IYLX4 への接続は ISC4 という APPC 接続です。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。334 ページの図

87 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 87. 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しなくてもよければ、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 Co1 Sys(ISC4) Net(..GBIBMIYA.IYLX150 M. A....)
Uowl(01680005) Uow(ABD40B40C67C8201) Con Lin(IYLX4 )
Coo Appc Co1 Sys(ISC4) Net(..GBIBMIYA.IYLX151 M. F@b..)
Uowl(016D0005) Uow(ABD40B40DA5A8803) Con Lin(IYLX4 )
Coo Appc Co1 Sys(ISC4) Net(..GBIBMIYA.IYLX156 M. .!h..)

```

図 88. 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 89. 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 90. 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 91. 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) ) R1e(008) Enq(00000008)
  Uow(ABD40B40C67C8201) Tra(RFS1) Tas(0000675) Ret Tsq Own
    Res(RFS1X151) ) R1e(008) Enq(00000008)
  Uow(ABD40B40DA5A8803) Tra(RFS1) Tas(0000676) Ret Tsq Own
    Res(RFS1X156) ) R1e(008) Enq(00000008)

```

図 92. 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) ) R1e(008) Enq(00000008)

```

図 93. 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.IYLY150 M. A....)
  Qualifier()
  Qlen(000)

```

図 94. 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 で失敗した機能シブ要求も含む）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 IYLY1 の棚上げされた UOW は、コールド/ウォーム・ログの IYLY4 との不一致のあと保持される。
- IYLY1 と IYLY4 の間の APPC 接続は、保留条件が解決されるまで、機能シブ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れば、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が、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、161 ページの『第 13 章 リモート・システムへのリンクの定義』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 (Specifying CICS system initialization parameters)』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』
- 340 ページの『単一セッション APPC デバイス』
- 340 ページの『並列 APPC セッション』
- 340 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。
2. セッションが追跡されて、引き継ぎが起これ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、次のいずれかの場合に APPC セッションを再確立しようとしています。

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、次のいずれかの場合に LU サービス管理セッションを再確立しようとしています。

- 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ています。

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 持続セッションのための CICS サポートの詳細については、「*CICS Recovery and Restart Guide*」を参照してください。この章では、持続セッションのうち、特にシステム間通信に適用される部分を説明します。持続セッション・サポートに必要なリンク定義の詳細については、161 ページの『第 13 章 リモート・システムへのリンクの定義』および「*CICS Resource Definition Guide*」の『CONNECTION 定義の属性 (CONNECTION definition attributes)』を参照してください。持続セッション・サポートの指定に使用する PSDINT 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『PSDINT』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。

- 『持続セッションのサポートと XRF との比較』
- 342 ページの『相互接続された CICS の環境、リカバリー、および再始動』
- 343 ページの『アプリケーション・プログラムへの影響』

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 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、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 持続セッションのサポートを両方とも指定することはできません。)

XRF では、システムの可用性 (活動システムと代替システムによって) とユーザーのための可用性 (システムの可用性とバックアップ・セッションの利用によって) が得られます。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 (例えば、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、それぞれのものがが必要です。

持続セッションのサポートでは、システムの可用性 (1 つのシステムの代わりに再始動することによって) とエンド・ユーザーのための可用性 (システムの可用性と持続セッションによって) が得られます。必要なデータ・セットは一組だけです。必要なシステムは 1 つだけです。持続セッションのサポートには、XRF に比べ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。

ただし、持続セッション・サポートでは、VTAM、MVS、または CEC に障害があったときには、セッションが保存されません。そのような障害のとき速やかに再始動する必要がある場合には、持続セッションの代わりに XRF を使用することができます。

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によって相互に接続することができます。

MRO セッション

MRO 接続は、CICS の障害とそのあとの緊急再始動にまたがって持続することはできません。

LU6.1 セッション

複数システム環境で CICS に障害があると、それが緊急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。

持続セッションの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。持続セッションの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、障害のあとただちにアンバインドされます。

持続セッションのサポートがあるシステムでは、LU6.1 セッションは、緊急再始動(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。

- そのセッションに対しカタログ項目が見つからない場合。これは次のセッションに適用されます。
 - 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 開始トランザクション・ルーティングについては、70 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』のページを参照してください。)
- 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、この障害の時点で進行している会話がなければ、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、DEALLOCATE(ABEND) が障害のある CICS のパートナーに送られます。)

LU6.2 セッション制御への影響

LU6.2 相互接続環境で CICS の障害があり、それが持続セッション遅延間隔内に緊急再始動されたあと、トランザクション CLS1 (CNOS) は、その接続の一方の側が、ゼロにするための CNOS 要求をその障害より前に出しているか、または、その障害時にその接続で CNOS 折衝が進行中でない限り、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 が実行されないと、接続システムのいずれによっても、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。

アプリケーション・プログラムへの影響

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。これについては、「*CICS Distributed Transaction Programming Guide*」を参照してください。

第 7 部 付録

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』
- 349 ページの『DPL 要求の動的ルーティング』
- 349 ページの『自動トランザクション開始』
- 350 ページの『基本マッピング・サポート』
- 350 ページの『LUTYPE6.1 セッションの獲得』
- 350 ページの『同期点処理』
- 350 ページの『ローカル名とリモート名』
- 351 ページの『マスター端末トランザクション』
- 351 ページの『インストールと操作』
- 351 ページの『リソース定義』
- 351 ページの『カスタマイズ』
- 351 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

- 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 コマンドによって任意の 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。
- 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 사용할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、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、マップは、必要なシステムにはどこにでも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 コマンドで呼び出されたトランザクションが拡張ルーティング可能であるためには、次のすべての条件を満たしていなければなりません。
 - 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、次の条件すべてを満たしている。
 - 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。すなわち、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、トランザクションを開始す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、TOR、およびターゲット領域はすべて CICS Transaction Server for OS/390、バージョン 1 リリース 3 以降である。
 - 要求側領域と TOR は (異なる領域の場合)、次のいずれかで接続されている。
 - MRO リンク
 - APPC 並列セッション・リンク
 - TOR とターゲット領域は、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、リンクを介してすでに行われている。
 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。

- 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。
 - 要求領域が CICS Transaction Server for OS/390、バージョン 1 リリース 3 以降である。
- 注: 分散ルーティング・プログラムを要求領域および ターゲット 領域で起動するには、ターゲット領域も CICS Transaction Server for OS/390、バージョン 1 リリース 3 以降である必要があります。
- 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 リンクを使用していて、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には、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、リンクを介してすでに行われている。
 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:
 - 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、トランザクションを開始するリモート領域が明示的に指定されていない)。
 - 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이지ー・チェーンすることはできない。
 - 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可能であるためには、リモート・プログラムが次のようになっていなければなりません。
 - ローカル・システムに対して DYNAMIC と定義されている。あるいは、
 - ローカル・システムに対して定義されていない。
- 動的にルーティングされる DPL 要求のデ이지ー・チェーンはサポートされていません (104 ページの『DPL 要求の「デ이지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 (347 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してください)。

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場合に、リモート・システムが使用できないと、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 セッションと MRO セッションでのみサポートされています。

ローカル名とリモート名

- 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端末は、そのマスター端末トランザクションが実際に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 ペアの上に複数の APPC 接続をインストールしないでください。
- LU-LU ペアの上に 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ぬ場合には、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 プログラム言語 (SNA 資料の「*Transaction Programmer's Reference Manual for LU Type 6.2*」を参照) が CICS によってどのように実装されているかを示します。この付録には、以下のトピックが収められています。

- 『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。

- 354 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。

- 363 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 21. 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 <code>receive_immediate</code> 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
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

表 21. 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
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・ログへのログイン	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 (CEMT master terminal)』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、174 ページの『APPC リンクの定義』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 リソース定義ガイド*」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、362 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 22. 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

表 22. CHANGE_SESSION_LIMIT (続き)

CHANGE_SESSION_LIMIT	CEMT SET MODENAME
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 23. 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 24. 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 25. 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
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 26. 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 27. DEACTIVATE_CONVERSATION_GROUP

DEACTIVATE_CONVERSATION_GROUP	サポートなし
--------------------------------------	--------

表 28. DEACTIVATE_SESSION

DEACTIVATE_SESSION	サポートなし
---------------------------	--------

表 29. 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 30. 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

表 30. DEFINE_MODE (続き)

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)
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 31. 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
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

表 31. DEFINE_REMOTE_LU (続き)

DEFINE_REMOTE_LU	DEFINE CONNECTION (CICS リソース定義)
SECURITY_ACCEPTANCE(NONE)	ATTACHSEC(LOCAL)
SECURITY_ACCEPTANCE (CONVERSATION)	ATTACHSEC(VERIFY)
SECURITY_ACCEPTANCE (ALREADY_VERIFIED)	ATTACHSEC(IDENTIFY) または ATTACHSEC (PERSISTENT)

表 32. 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(MAPPEDIBASIC)	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 CONFIRMvSYNCPT)	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
RETURN_CODE	サポートあり

表 33. 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 34. 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 35. 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(YIN)
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 36. DISPLAY_MODE

DISPLAY_MODE	CEMT INQUIRE MODENAME + CEMT INQUIRE TERMINAL
FULLY_QUALIFIED_LU_NAME(vble)	指定不可

表 36. DISPLAY_MODE (続き)

DISPLAY_MODE	CEMT INQUIRE MODENAME + CEMENT INQUIRE TERMINAL
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 37. 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
INSTANCE_COUNT(vble)	CEMT INQ TRAN()

表 37. DISPLAY_TP (続き)

DISPLAY_TP	CEMT INQUIRE TRANSACTION
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、あとで LU サービス管理トランザクション (CLS1) が実際に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 38 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 マスター端末のオペレーターは、コマンドを出したあとに、いちいち 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。エラーが実際に起こるといったごくまれな場合、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 38 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 38. 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 38. CLSI 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。

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

APPC トランザクション・ルーティングだけに適用される逸脱が 1 つあります。

- トランザクション・プログラムは、同期解放、同期送信、同期受信の状態では 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。

参考文献

CICS Transaction Server for z/OS ライブラリー

CICS Transaction Server for z/OS について公開されている情報は、以下の形式で提供されます。

CICS Transaction Server for z/OS インフォメーション・センター

CICS Transaction Server for z/OS インフォメーション・センターは、CICS Transaction Server のユーザー情報の主要な情報源です。インフォメーション・センターには、以下のものが含まれています。

- CICS Transaction Server に関する情報 (HTML 形式)
- ライセンス交付を受けた、または受けていない CICS Transaction Server の資料 (Adobe PDF ファイル形式)。これらのファイルを使用すると、資料のハードコピーを印刷できます。詳しくは、『PDF のみの資料』を参照してください。
- 関連製品についての情報 (HTML 形式および PDF ファイル形式)

本製品の CD-ROM には、CICS インフォメーション・センターのコピーが自動的に提供されます。インフォメーション・センターのフィーチャー番号 7014 を指定すると、追加料金なしでさらに多くのコピーを注文することができます。

ライセンス交付を受けた文書は、ライセンス所有者のみが入手できます。ライセンス交付を受けていない情報のみを含む インフォメーション・センターのバージョンは、資料注文システムを通じて、注文番号 SK3T-6945 で入手できます。

同梱されているハードコピーの資料

以下のハードコピー形式の基本資料は、製品に自動的に同梱されます。詳しくは、『同梱セット』を参照してください。

同梱セット

同梱セットは以下のハードコピー資料で構成されています。これらのハードコピー資料は、CICS Transaction Server for z/OS バージョン 3 リリース 2を注文すると自動的に提供されます。

Memo to Licensees, GI10-2559

CICS Transaction Server for z/OS Program Directory, GI13-0515

CICS Transaction Server for z/OS リリース・ガイド, GC88-4364

CICS Transaction Server for z/OS インストール・ガイド, GC88-4365

CICS Transaction Server for z/OS Licensed Program Specification, GC34-6608

上記の注文番号を使用すると、同梱セット内の以下の資料のコピーを追加注文できます。

CICS Transaction Server for z/OS リリース・ガイド

CICS Transaction Server for z/OS インストール・ガイド

CICS Transaction Server for z/OS Licensed Program Specification

PDF のみの資料

以下の資料は、CICS インフォメーション・センターで Adobe PDF ファイル形式で入手できます。

CICS Transaction Server for z/OS の CICS の資料

汎用

CICS Transaction Server for z/OS Program Directory, GI13-0515
CICS Transaction Server for z/OS リリース・ガイド, GC88-4364
CICS Transaction Server for z/OS CICS TS バージョン 3.1 からのマイグレーション, GC88-4369-00
CICS Transaction Server for z/OS CICS TS バージョン 1.3 からのマイグレーション, GC88-4366-00
CICS Transaction Server for z/OS CICS TS バージョン 2.2 からのマイグレーション, GC88-4367-00
CICS Transaction Server for z/OS インストール・ガイド, GC88-4365

管理

CICS System Definition Guide, SC34-6813
CICS Customization Guide, SC34-6814
CICS Resource Definition Guide, SC34-6815
CICS Operations and Utilities Guide, SC34-6816
CICS Supplied Transactions, SC34-6817

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, SC88-4370
CICS アプリケーション・プログラミング・リファレンス, SC88-4371
CICS System Programming Reference, SC34-6820
CICS Front End Programming Interface User's Guide, SC34-6821
CICS C++ OO Class Libraries, SC34-6822
CICS Distributed Transaction Programming Guide, SC34-6823
CICS Business Transaction Services, SC34-6824
Java Applications in CICS, SC34-6825
JCICS Class Reference, SC34-6001

診断

CICS Problem Determination Guide, SC34-6826
CICS Messages and Codes, GC34-6827
CICS Diagnosis Reference, GC34-6862
CICS Data Areas, GC34-6863-00
CICS Trace Entries, SC34-6828
CICS Supplementary Data Areas, GC34-6864-00

通信

CICS 相互通信ガイド, SC88-4373
CICS External Interfaces Guide, SC34-6830
CICS インターネット・ガイド, SC88-4374

特殊なトピック

CICS Recovery and Restart Guide, SC34-6832
CICS パフォーマンス・ガイド, SC88-4375
CICS IMS Database Control Guide, SC34-6834
CICS RACF Security Guide, SC34-6835
CICS Shared Data Tables Guide, SC34-6836
CICS DB2 Guide, SC34-6837
CICS Debugging Tools Interfaces Reference, GC34-6865

CICS Transaction Server for z/OS の CICSplex SM 関連の資料 汎用

CICSplex SM Concepts and Planning, SC34-6839
CICSplex SM User Interface Guide, SC34-6840
CICSplex SM Web User Interface Guide, SC34-6841

管理

CICSplex SM Administration, SC34-6842
CICSplex SM Operations Views Reference, SC34-6843
CICSplex SM Monitor Views Reference, SC34-6844
CICSplex SM Managing Workloads, SC34-6845
CICSplex SM Managing Resource Usage, SC34-6846
CICSplex SM Managing Business Applications, SC34-6847

プログラミング

CICSplex SM Application Programming Guide, SC34-6848
CICSplex SM Application Programming Reference, SC34-6849

診断

CICSplex SM Resource Tables Reference, SC34-6850
CICSplex SM Messages and Codes, GC34-6851
CICSplex SM Problem Determination, SC34-6852

CICS ファミリーの資料

通信

CICS Family: Interproduct Communication, SC34-6853
CICS Family: Communicating from CICS on zSeries, SC34-6854

ライセンス出版物

以下のライセンス出版物は、ライセンスが交付されていないバージョンのインフォメーション・センターには含まれていません。

CICS Diagnosis Reference, GC34-6862
CICS Data Areas, GC34-6863-00
CICS Supplementary Data Areas, GC34-6864-00
CICS Debugging Tools Interfaces Reference, GC34-6865

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 3 リリース 2 に同梱ではない) ものです。

<i>Designing and Programming CICS Applications</i>	SR23-9692
<i>CICS Application Migration Aid Guide</i>	SC33-0768
<i>CICS Family: API Structure</i>	SC33-1007
<i>CICS Family: Client/Server Programming</i>	SC33-1435
<i>CICS Transaction Gateway for z/OS Administration</i>	SC34-5528
<i>CICS Family: General Information</i>	GC33-0155
<i>CICS 4.1 Sample Applications Guide</i>	SC33-1173
<i>CICS/ESA 3.3 XRF Guide</i>	SC33-0661

関連ライブラリーの資料

IMS

- *CICS/VS to IMS/VS Intersystem Communication Primer*, SH19-6247 から SH19-6254
- *IMS/ESA データ通信管理の手引き*, SC88-7321
- *IMS/ESA Installation Volume 1: Installation and Verification*, GC26-8736
- *IMS/ESA 導入 第 2 巻*, GD88-7125
- *IMS/ESA Operations Guide*, SC26-8741
- *IMS Programming Guide for Remote SNA Systems*, SC26-4186

MVS/ESA

- *OS/390 MVS シスプレックスのセットアップ*, GC88-6590
- *OS/390 並列シスプレックス アプリケーションのマイグレーション*, GC88-6616

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)

- *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

- *OS/390 IBM Communications Server SNA カスタマイズ*, LY88-7817
- *OS/390 eNetwork Communications Server: SNA Data Areas Volume 1*, LY43-0111
- *OS/390 eNetwork Communications Server: SNA Data Areas Volume 2*, LY43-0112
- *OS/390 eNetwork Communications Server: SNA Diagnosis*, LY43-0079
- *OS/390 IBM Communications Server SNA マイグレーション*, SD88-7862
- *OS/390 eNetwork Communications Server: SNA Messages*, SC31-8569
- *OS/390 IBM Communications Server SNA ネットワーク・インプリメンテーション・ガイド*, SD88-7854
- *OS/390 IBM Communications Server SNA オペレーション*, SD88-7856
- *OS/390 eNetwork Communications Server: SNA Programming*, SC31-8573

- VTAM (MVS/ESA 版) リリースの手引き, GC88-7585
- OS/390 IBM Communications Server SNA リソース定義解説書, SD88-7855

出版物が最新であるかどうかの判別

IBM は、新規情報や変更された情報を使用して、その資料を定期的に更新しています。通常、最初の発行時には、資料のハードコピー・バージョンと BookManager[®] ソフトコピー・バージョンの両方がそろって作成されます。ただし、ハードコピーの資料には印刷と配布のための時間が必要になるので、BookManager バージョンの方が、公開の直前に加えられた変更内容が反映される可能性が高くなります。

それ以降の更新は、ハードコピーよりも早くソフトコピーで入手できるようになります。つまり、リリースが出荷されてから以降は、常にソフトコピー・バージョンが最新と見なされます。

CICS Transaction Server の資料の場合、これらのソフトコピーの更新は定期的に *Transaction Processing and Data Collection Kit* CD-ROM (SK2T-0730-xx) に反映されます。コレクション・キットは、再発行されるたびに、その注文番号の接尾部 (-xx 部) が更新されます。例えば、コレクション・キット SK2T-0730-06 は、SK2T-0730-05 よりも新しいキットです。コレクション・キットも、カバーに日付が明記されます。

ソフトコピーの更新箇所は、変更部分の左側にリビジョン・コード (通常は # 文字) でマークが付けられます。

アクセシビリティ

アクセシビリティ機能は、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 システムで必要になるアクセシビリティ機能を利用することができます。

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 69

アプリケーション所有領域 (AOR) 65

アプリケーション・プログラミング

概説 255

機能シップのための 257

トランザクション・ルーティングのための 269

非同期処理のための 267

CICS の APPC verb への対比 353

CICS-IMS 間 273

DPL のための 261

LUTYPE6.1 会話 (CICS-IMS 間) 273

一時記憶域

機能シップ 33, 258

一時データ

機能シップ 34, 259

インストール

システム間通信 125

総称リソース、VTAM 137

タイプ 3 SVC ルーチン 121

複数領域操作 121

リンク・バック域の MRO モジュール 121

CICS のための ACF/VTAM 定義 125

LOGMODE 項目 126

IMS システム定義 128

IMS の考慮事項 127

IMS のための ACF/VTAM 定義 127

LOGMODE 項目 127

MRO のためのサブシステム・サポート 121

VTAM 総称リソース 137

インターバル制御機能

機能シップ 31

オンライン・リソース定義 (RDO)

間接リンク 196

シップ可能端末定義 227

複数領域操作のためのリンク 164

リモート VTAM 端末 224

リモート・システムへのリンク 162

リモート・トランザクション 222

リモート・リソース 213

APPC 端末 179, 180

APPC リンク 175

LUTYPE6.1 リンク 184

オンライン・リソース定義 (RDO) (続き)

LUTYPE6.2 リンク 174

[カ行]

解放済み、接続状況 200, 204

解放中、接続状況 204

会話

LUTYPE6.1 セッション (CICS-IMS 間) 288

拡張対等ネットワーク機能 (APPN) 134

獲得済み、接続状況 200

獲得済み、接続状況 200

間接リンク

リソース定義 194

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 191

シップ可能端末での 193

どのような場合に必要か 194

ハードコーディング端末での 193

例 194

疑似会話型トランザクション

トランザクション・ルーティングでの 270

機能シップ

アプリケーション・プログラミング 257

一時記憶域 33, 258

一時データ 34, 259

インターバル制御機能 31

起こるキューイング 34

主要な説明 31

設計上の考慮事項 32

短パス変換プログラム 39

ファイル制御 32, 258

ミラー・トランザクション 35

ミラー・トランザクションの異常終了 260

リモート・リソースの定義 215

一時記憶域キュー 218

一時データ宛先 218

ファイル 216

DL/I PSB 217

例外条件 259

DL/I 要求 33, 258

基本会話 26

基本機能

定義された 255

デフォルト・プロファイル 244

基本マッピング・サポート (BMS)

規則と制約事項の要約 350

トランザクション・ルーティングでの 93, 269

キュー・モデル 246

- 競合 敗者 28
- 競合勝者 28
- 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)
 - 同期レベル 26
 - パートナーの定義 241
 - PIP データ 26
- グローバル・ユーザー出口
 - XALTENF 47, 74, 95
 - XICTENF 47, 74, 95
 - XISCONA 298
 - XPCREQ 101
 - XPCREQC 101
 - XZIQUE 298
- 検索、START コマンドによってシップされる情報の 50
- 限定リソース 27
 - 効果 205
- 交換回線
 - コスト効率 27
- コマンド順序
 - LUTYPE6.1 セッション (CICS-IMS 間) 291

[サ行]

- サービス・クラス (COS) 27
 - モードセット 27, 174
 - モード名を提供するためのデフォルト・プロファイルの修正 245
 - ACF/VTAM LOGMODE 項目 126
- サブシステム・インターフェース (SSI)
 - での MRO に必要 121
- サロゲート TCTTE 270
- システム間 MRO (XCF/MRO)
 - 概説 14
 - サポートの生成 122
 - ハードウェア要件 122
- システム間カップリング・ファシリティー (XCF)
 - 概説 14
 - システム間 MRO のための 122
 - 領域間通信に使用される 13
- システム間キュー
 - 待機セッション要求の制御 34, 297
- システム間セッション 25
- システム間通信 (ISC)
 - インストールの注意点 125
 - 概念 4, 23
 - 機能 6
 - 互換 APPC ノードの定義 177
 - 互換の CICS ノードと IMS ノードの定義 185
 - システム間の接続 23
 - セッション 25
- システム間通信 (ISC) (続き)
 - 待機セッション要求の制御 297
 - 多重チャネル・アダプター 24
 - チャネル間 通信 24
 - 定義された 4
 - トランザクション・ルーティング 65
 - ホスト内通信 24
 - APPC 端末の定義 179
 - APPC モードセットの定義 176
 - APPC リンクの定義 174
 - LUTYPE6.1 リンクの定義 184
 - SNA 経由 4, 23, 125
 - TCP/IP 経由 132
 - VTAM 持続セッションの使用 182, 341
- システム初期設定パラメーター
 - システム間通信の 125
 - シップされた端末の削除のための 302
 - 複数領域操作のための 121
- APPLID 162, 196
- DSHIPIDL 302
- DSHIPINT 302
- DTRTRAN 239
- FSSTAFF 77
- GRNAME 137
- PSDINT 182
- SYSIDNT 163
- TCPIP 29
- VTAM 総称リソースのための 137
- XRF 182

- システム・メッセージ・モデル 246
- シスプレックス、MVS
 - システム間 MRO の要件 122
 - 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 13, 14
 - 動的トランザクション・ルーティング 20
- パフォーマンス
 - CICSplex SM の使用 20
 - MVS ワークロード・マネージャーの使用 20
 - VTAM 総称リソースの使用 20, 133
- 持続セッション、VTAM 176, 177, 182, 341, 342, 343
- シップ可能端末
 - シップされるもの 227
 - 選択的削除 301
 - 「端末未認識」状態 73
 - リソース定義 228
 - ATI による 72
- 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 302
 - パフォーマンスの考慮事項 303
 - INQUIRE DELETSIPPED コマンド 303

シッパされた端末定義 (続き)
削除 (続き)
SET DELETSHIPED コマンド 303
選択的削除メカニズム 301
タイムアウト削除メカニズム 302
シッパされた端末定義の削除 301
自動インストール
シッパされた端末定義の削除 301
ユーザー・プログラム、DFHZATDY 179
APPC 単一セッション端末の 180
APPC 単一セッションの
BIND 要求によって開始された 178
CINIT 要求によって開始された 180
APPC 並列セッションの 178
自動開始記述子 (AID) 71
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 250
規則と制約事項の要約 349
シッパされた端末定義の制約事項 227
端末未認識状態 72
定義 70
トランザクション・ルーティング 70
非同期処理による 47
ルーティング・トランザクションによる制限 95
据え置き伝送
LUTYPE6.1 セッション (CICS-IMS 間) 289
START NOCHECK 要求 48
スケジューラー・モデル 246
静止
接続処理 324
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 237
別々のローカル定義とリモート定義を使用する
237
セキュリティ
ルーティングされたトランザクションの 237
RTIMOUT オプション 237
セッションの障害
未確定期間中 309
セッション割り振り
LUTYPE6.1 セッション (CICS-IMS 間) 282
セッション・キューの管理
概説 297
QUEUELIMIT オプションの使用 297
XZIQUE グローバル・ユーザー出口の使用 298, 299
セッション・バランシング
VTAM 総称リソースの使用 133
接続、リモート・システムへの
数の制限 25
接続、リモート・システムへの
解放済み、状況 204

接続、リモート・システムへの (続き)
解放中、状況 204
獲得済み、状況 200
数の制限 175
接続の解放 204
接続の獲得 200
定義 161
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) 284
接続静止プロトコル (CQP) 324
選択的削除、シッパされた定義の 301
相互通信機能
概念 4, 23
総称 applid
総称リソース名との混同 197
特定 applid との関係 196
総称リソース、VTAM
アウトバウンドの LU6 接続 156
インストール 137
概説 20
シスプレックス間通信 141
制約事項 154
非自動インストール接続での使用 155
非自動インストール端末での使用 155
へのマイグレーション 138
要件 133
類縁性の終了 146

[タ行]

体系化プロセス
デフォルト定義の修正 247
プロセス名 247
リソース定義 247
体系化プロセス (モデル) 246
代替機能
定義された 255
デフォルト・プロファイル 244
タイプ 3 SVC ルーチン
および CICS applid 162
領域間通信に使用される 13
領域間リンクに対する指定 166
LPA にある 121
タイムアウト削除メカニズム、シッパされた端末の
302
多重チャネル・アダプター 24
多重ミラー状態 37
短パス変換プログラム 39
端末専有領域 (TOR) 65
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 133
領域間のセッション・バランシング 133

- 端末の別名 234
- 端末未認識状態、ATI 中の 73
- チェーン・ミラー状態 37
- チャンネル間 通信 24
- 中継トランザクション 92
 - トランザクション・ルーティングのための 65
- 長期実行ミラー・タスク 38
- 重複してシップされた端末定義 301
- 通信プロファイル 243
- データ・ストリーム
 - IMS 通信のユーザー・データ・ストリーム 187
- データ・テーブル 216
- 同期点 115, 308, 350
- 同期レベル 26, 115
 - CPI コミュニケーション 26
- 同情病
 - 削減 297
- 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 20
 - 紹介 67
 - トランザクション定義
 - 同一定義の使用 238
 - 別々のローカル定義とリモート定義を使用する 238
 - CRTX トランザクションの使用 239
 - TOR で単一定義を使用する 239
 - ルーティング・プログラムに渡される情報 68
 - ルーティング・プログラムの使用目的 69
 - ルーティング・プログラムの呼び出し 68
 - CICS Interdependency Analyzer 69
 - CICSplex SM による制御 20, 70
 - CICSplex における 19
- 動的ルーティング
 - インターフェースの概要 57
- 動的ルーティング・プログラム、DFHDYP 67, 102
- 特定 applid
 - 総称 applid との関係 196
 - XRF のための 197
- 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 269
 - 起こるキューイング 95
 - 概説 65
 - 間接リンク
 - 概説 191
 - シップ可能端末での 193
 - 定義方法 196
 - どのような場合に必要か 194
 - ハードコーディング端末での 193
 - 例 194
 - 疑似会話型トランザクション 270
 - 基本マッピング・サポート 93, 269
 - システム・プログラミングに関する考慮事項 95

- トランザクション・ルーティング (続き)
 - シップされた端末定義の削除 301
 - 自動開始記述子 (AID) 71
 - 自動トランザクション開始 72
 - セキュリティの考慮事項 237
- 端末から開始された
 - 静的 67
 - 動的 67
 - 動的ルーティング・プログラムに渡される情報 68
 - 動的ルーティング・プログラムの使用目的 69
 - 動的ルーティング・プログラムの呼び出し 68
- 端末のシップ 72
- 中継トランザクション 65
- 中継プログラム 92
- リモート・リソースの定義 224
 - 静的にルーティングされるトランザクション 237
 - 端末 224, 229
 - 動的にルーティングされるトランザクション 238
 - トランザクション 234
- ルーティング・トランザクション、CRTE 94
- AOR での ASSIGN コマンドの使用 270
- APPC 端末 89
- ATI 要求によって開始された 70
- CICS Interdependency Analyzer 69

[ナ行]

- 名前
 - リモート・システム 163
 - ローカル CICS システム 162
- 二重目的 RDO 定義 237

[ハ行]

- バックエンド・トランザクション
 - 定義された 255
 - LUTYPE6.1 セッション (CICS-IMS 間) 287
- パフォーマンス
 - シップされた端末定義の削除 301, 303
 - 静的トランザクション・ルーティングの使用 19
 - 待機セッション要求の制御 34, 49, 95, 105, 297
 - 重複してシップされた端末定義 301
 - 動的トランザクション・ルーティングの使用 20
 - CICSplex SM の使用 20
 - DPL 要求の動的ルーティングの使用 20
 - MVS ワークロード・マネージャーの使用 20
 - VTAM 総称リソースの使用 20
- 非同期処理
 - アプリケーション・プログラミング 267
 - 起こるキューイング 49
 - システム・プログラミングに関する考慮事項 51

非同期処理 (続き)

- 情報検索 50
- 代表的なアプリケーション 43
- 端末の獲得 51
- 同期処理 (DTP) との比較 43
- パフォーマンスの向上 47
- リモート・トランザクションの開始 45
- リモート・トランザクションの定義 222
- リモート・トランザクションの取り消し 46
- 例 52
- ローカル・キュー 49
- CICS-IMS 間 275
- DTP によって開始される 44
- NOCHECK オプション 47
- PROTECT オプション 48
- RETRIEVE コマンド 50
- SEND および RECEIVE インターフェース 45
 - CICS-IMS 間アプリケーション 280
- START および RETRIEVE インターフェース
 - CICS-IMS 間アプリケーション 275
- START コマンドによって渡される情報 46
- START と RETRIEVE インターフェース 44, 45

非同期処理の方式 44

ファイル制御

- 機能シッパ 32, 258

複数領域操作 (MRO)

アプリケーション

- タイム・シェアリング 19
- プログラム開発 18
- あらかじめ用意された起動システム 122
- 異常終了コード 351
- インストールの注意点 121
- 応用 18
- 応用方法

- 信頼できるデータベース・アクセス 19
- 部門の分離 19
- マルチプロセッシング 19
- ワークロード・バランシング 20

概念 13

間接リンク 191

機能 6, 14

- 互換ノードの定義 167
- サブシステムとして定義 121
- システム間 MRO (XCF/MRO) 14, 122
- シスプレックスにおける 20
- 待機セッション要求の制御 297
- 単一領域からの変換 21
- 短パス変換プログラム 39
- 長期実行ミラー・タスク 38
- 定義された 3
- トランザクション・ルーティング 65
- 領域間通信 13

複数領域操作 (MRO) (続き)

- リンクの定義 164
- リンク・バック域のモジュール 121
- CICSplex における 19
- MRO リンクの定義 164
- VTAM 持続セッションの使用 341

プロファイル

- 基本機能の 244
- 代替機能のための 243
- デフォルト定義の修正 245
- 読み取りタイムアウト 244
- リソース定義 243
- CICS 提供のデフォルト 244
- フロントエンド・トランザクション
 - 定義された 255
- LUTYPE6.1 セッション (CICS-IMS 間) 282

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 273
- 概説 109
- 非同期処理との比較 43
- リモート・リソースの定義 241
- APPC 端末の API として 90
- CICS-IMS 間 281
- PARTNER 定義 241

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 261
- 起こるキューイング 105
- 概説 97
- グローバル・ユーザー出口 101
- サーバー・プログラム 261
 - リソース定義 252
- サーバー・プログラムでの制約事項 105
- デ이지ー・チェーン要求 104
- ミラー・トランザクションの異常終了 264
- 要求の静的ルーティング
 - サーバー・プログラムの定義 219
 - 説明 98
- 要求の動的ルーティング
 - サーバー・プログラムの定義 219
 - 紹介 101
 - ルーティングに対し適格 102
 - ルーティング・プログラムを呼び出す場合 103
- リモート・サーバー・プログラムの定義 219
- 例 106
- 例外条件 263
- ローカル・リソース定義 252
- CICSplex SM による制御 103, 220
- 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 238
 - BTS アクティビティのルーティングのための 238

ホスト内 ISC 24

[マ行]

マイグレーション

単一領域操作から MRO への 21

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 269

マクロ・レベル・リソース定義

リモート DL/I PSB 217

リモート一時記憶域キュー 218

リモート一時データ宛先 218

リモート・サーバー・プログラム 219

リモート・トランザクション 222

リモート・ファイル 216

リモート・リソース 213

マップ式会話 26

未確定期間 309

セッションの障害 309

ミラー・トランザクション 35

長期実行ミラー・タスク 38

DPL でのリソース定義 252

ミラー・トランザクションの異常終了 260, 264

モードグループ

定義 27

SNASVCMG 200

VTAM LOGMODE 項目 126

モードセット 176

定義 27, 174

LU サービス・マネージャー 126

モード名 174

モデル 246

[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 67

[ラ行]

リカバリーおよび再始動 307

リカバリーと再始動

同期点交換 308

同期点フロー 309

動的トランザクション・バックアウト 312

未確定期間 309

リソース定義

間接リンク 191

機能シップ 215

互換 APPC ノードの定義 177

互換 MRO ノードの定義 167

互換の CICS ノードと IMS ノードの定義 185

リソース定義 (続き)

体系化プロセス 247

体系化プロセス定義の修正 247

デフォルト・プロファイル 244

デフォルト・プロファイルの修正 245

トランザクション・ルーティング 224

非同期処理 222

複数領域操作のためのリンク 164

プロファイル 243

分散トランザクション処理 241

ミラー・トランザクション 252

リモート DL/I PSB 217

リモート一時記憶域キュー 218

リモート一時データ宛先 218

リモート端末 224, 229

リモート・サーバー・プログラム 219

リモート・システムへのリンク 161

リモート・トランザクション 222, 234

リモート・パートナー 241

リモート・ファイル 216

リモート・リソース 213

ローカル・リソース 243

APPC 端末 179

APPC モードセット 176

APPC リンク 174

CICS-IMS 間 LUTYPE6.1 リンク 184

複数リンクの定義 189

DPL 219, 252

サーバー・プログラム 252

LUTYPE6.1 リンク 184

LUTYPE6.2 リンク 174

リモート DL/I PSB 217

リモート一時記憶域キュー

定義 218

リモート一時データ宛先

定義 218

リモート端末

端末 ID 232

DFHTCT TYPE=REGION を使用する定義 231

DFHTCT TYPE=REMOTE を使用する定義 230

リモート・サーバー・プログラム

定義 219

プログラム名 220

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 234

静的ルーティング 237

動的ルーティング 238

非同期処理の定義 222

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 237

リモート・ファイル

定義 216

- リモート・ファイル (続き)
 - ファイル名 216
 - レコード長 217
- リモート・リソース
 - 定義 213
 - 命名 214
- 領域間通信 (IRC) 13
 - 短パス変換プログラム 39
- リンク、リモート・システムへの 161
- リンク・バック域のモジュール、MRO のための 121
- ルーティング・トランザクション、CRTE 94
 - 自動トランザクション開始 95
 - CEMT の呼び出し 95
- 類縁性、総称リソースとパートナー LU の間の 146
- 例外条件
 - 機能シップ 259
 - DPL 263
- レコード長、リモート・ファイルの 217
- レベル、同期の 26
- ローカル CICS システム
 - 総称リソース名 197
 - 命名 162
 - applid 162
 - 総称と特定 196
 - sysidnt 163
- ローカル名、リモート・リソースの 214
- ローカル・キュー、START 要求の 49
- ローカル・リソースの定義
 - 区画内一時データ・キュー 249
 - 体系化プロセス 246
 - 通信プロファイル 243
 - DPL のための 252

[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 20
 - CICSplex SM の使用 20
 - DPL 要求の動的ルーティングの使用 20
 - MVS ワークロード・マネージャーの使用 20
 - VTAM 総称リソースの使用 20, 133

[数字]

- 1 次論理装置 (PLU) 28
- 2 次論理装置 (SLU) 28

A

- ACTION オプション 312
- ACTION 属性
 - TRANSACTION 定義 313

- AID (自動開始記述子) 71
- ALLOCATE コマンド
 - APPC セッションを使用可能にする 201
 - LUTYPE6.1 セッション (CICS-IMS 間) 281, 282
 - SYSIDERR の後で LUTYPE6.1 接続をサービス可能にする 350
- AOR (アプリケーション所有領域) 65
- APPC
 - 概説 25
 - 基本会話 26
 - サービス・クラス 27
 - 持続セッション 182, 342
 - 自動インストール
 - 単一セッション端末の 180
 - 並列セッション・リンクの 178
 - 単一セッション
 - 持続セッションの定義 183
 - 自動インストール 178, 180
 - 制限 27
 - 定義 179
 - 端末のリンク定義 179
 - 同期レベル 26
 - 並列セッション
 - 持続セッションの定義 182
 - 自動インストール 178
 - マスター端末操作 199
 - マップ式会話 26
 - モードセット定義 176
 - リンク定義 174
 - APPC アーキテクチャーへのマッピング 353
 - LU サービス・マネージャー 26, 175
- APPC アーキテクチャーからの逸脱 363
- APPC アーキテクチャーとの対比
 - 逸脱 363
 - 制御オペレーターの verb 354
- APPC アーキテクチャーへのマッピング 353
- APPC 端末
 - 持続セッション 183
 - 自動インストール 178
 - 代替機能としての 90
 - 端末定義のシップ 227
 - トランザクション・ルーティング
 - ALLOCATE による 66, 89, 90
 - リモート定義 225
 - リンク定義 179
- API 90
- CEMT コマンドの使用 180
- TYPETERM の AUTOCONNECT オプションの効果 182
- APPLID
 - IMS LOGMODE 項目 127
 - START コマンドによる受け渡し 47

applid

総称

総称リソース名との混同 197

総称、XRF のための 196

特定、XRF のための 196

ローカル CICS の 162

sysidnt との関係 163

APPN (拡張対等ネットワーク機能) 134

ASSIGN コマンド、AOR での 270

AUTOCONNECT オプション

APPC 端末に対する DEFINE TYPETERM の 182

APPC での CEMT コマンドへの効果 200

APPC リソース定義 181

DEFINE CONNECTION の

APPC に対する 181

DEFINE SESSIONS の

APPC に対する 182

B

BIND

送信側と受信側 28

BTS アクティビティのルーティング

トランザクション定義 238

BUILD ATTACH コマンド

LUTYPE6.1 セッション (CICS-IMS 間) 281, 284

C

CANCEL コマンド 46

CEMT マスター端末トランザクション

リモート端末での制約事項 351

ルーティング・トランザクションによる 95

APPC 端末に対する 180

DELETSHPED オプション 303

CICS Interdependency Analyzer 69

CICS 間通信

互換ノードの定義

APPC セッション 177

CICS と APPC アーキテクチャーの対比

逸脱 363

APPC アーキテクチャーからの逸脱 363

CICS の APPC アーキテクチャーへのマッピング 353

CICSplex

トランザクション・ルーティング 19

パフォーマンス

VTAM 総称リソースの使用 133

CICSplex SM による制御 20, 70, 103

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 20, 70

DPL 要求のルーティングの制御に使用 103, 220

CICS-CICS 間通信

互換ノードの定義

MRO セッション 167

CICS-IMS 間通信

アプリケーション設計 273

アプリケーション・プログラミング 273

互換ノードの定義 185

通信形式 275

データ形式 273

非同期処理 275

CICS フロントエンド 276

IMS フロントエンド 277

CICS と IMS の比較 273

RETRIEVE コマンド 279

RU のチェーン形式 274

SEND および RECEIVE インターフェース 280

START および RETRIEVE インターフェース 275

START コマンド 278

VLVB 形式 274

CNOS 折衝 202

CONNECTION 定義

PSRECOVERY オプション 183

CONVERSE コマンド

LUTYPE6.1 セッション (CICS-IMS 間) 281

CQP、接続静止プロトコルを参照 324

CRTE トランザクション 94

CRTX、CICS 提供のトランザクション定義 239

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 237

D

DBDCCICS 162

DEFINE CONNECTION

間接リンク 196

APPC 端末 179

LUTYPE6.1 リンク 175, 184

MRO リンク 164

NETNAME オプション 164

DEFINE PROFILE 244

DEFINE SESSIONS

間接リンク 196

APPC 端末 179

LUTYPE6.1 リンク 176, 184

MAXIMUM オプション

APPC での CEMT コマンドへの効果 202

MRO リンク 164

DEFINE TERMINAL

シップ可能端末定義 228

リモート VTAM 端末 225

APPC 端末 180

DEFINE TRANSACTION
 トランザクション・ルーティング 234
 DYNAMIC オプション 235
 PROFILE オプション 236
 PROGRAM オプション 236
 REMOTESYSTEM オプション 235
 TASKREQ オプション 236
 TRPROF オプション 236
 TWASIZE オプション 236
 非同期処理 222
 ACTION オプション 312
 WAIT オプション 312
 DEFINE TYPETERM
 APPC 端末 180
 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 245
 DFHCICSE
 基本機能のデフォルト・エラー・プロファイル 245
 DFHCICSF
 機能シップのデフォルト・プロファイル 245
 DFHCICSP
 CSPG の基本機能のプロファイル 245
 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 245
 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 245
 DFHCICST
 基本機能のデフォルト・プロファイル 244
 DFHCICSV
 CSNE、CSLG、CSRS の基本機能のプロファイル 244
 DFHDLPSB TYPE=ENTRY マクロ 217
 DFHDYP、動的ルーティング・プログラム 67, 102
 DFHTCT TYPE=REGION マクロ 231
 DFHTCT TYPE=REMOTE マクロ 230
 DFHTST TYPE=REMOTE マクロ 219
 DFHZATDY、自動インストール・ユーザー・プログラム 179
 DL/I
 機能シップ 33, 258
 リモート PSB の定義 217
 DL/I モデル 246
 DPL 要求の動的ルーティング
 サーバー・プログラムの定義 219
 シスプレックスにおける 20

DPL 要求の動的ルーティング (続き)
 紹介 101
 ルーティングに対し適格 102
 ルーティング・プログラムを呼び出す場合 103
 CICSplex SM による制御 20
 DSHIPIDL、システム初期設定パラメーター 302
 DSHIPINT、システム初期設定パラメーター 302
 DTRTRAN、システム初期設定パラメーター 239
 DYNAMIC オプション
 リモート・トランザクション定義での 235

E

EIB フィールド
 LUTYPE6.1 セッション (CICS-IMS 間) 290
 EXTRACT ATTACH コマンド
 LUTYPE6.1 セッション (CICS-IMS 間) 281, 287

F

FREE コマンド
 LUTYPE6.1 セッション (CICS-IMS 間) 282, 289
 FSSTAFF、システム初期設定パラメーター 77

G

GRNAME、システム初期設定パラメーター 137

I

IMS
 インストールの注意点 127
 システム定義 128
 非応答モード・トランザクション 276
 非会話型トランザクション 276
 メッセージ交換 276
 CICS との比較 273
 IP 相互接続
 概念 23, 28
 IPIC 28
 IP 相互接続 (IPIC)
 概念 4
 定義された 4
 IPIC
 概念 4
 定義された 320
 CICS TS for z/OS システムへの接続 320
 TCPIP 初期設定パラメーター 29
 ISSUE SIGNAL コマンド
 LUTYPE6.1 セッション (CICS-IMS 間) 282

L

LAST オプション 289
LOGMODE 項目
 CICS 126
 IMS 127
LU サービス・マネージャー
 説明 26
 SNASVCMG セッション 175
LU サービス・モデル 246
LUTYPE6.1
 リンク定義 184
 CICS-IMS 間アプリケーション・プログラミング
 273
LUTYPE6.2
 リンク定義 174
LU-LU セッション 25
 競合 28
 1 次 LU、2 次 LU 28

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 202
MAXQTIME オプション、CONNECTION 定義 34, 297
MODENAME 202
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
 13, 14
MVS 仮想記憶間サービス
 領域間リンクに対する指定 166

N

NETNAME 属性、CONNECTION リソースの
 デフォルト 164
 sysidnt へのマッピング 164
NOCHECK オプション
 START コマンドの 47
 ローカル・キューに必須の 49
NOQUEUE オプション
 ALLOCATE コマンドの
 LUTYPE6.1 セッション (CICS-IMS 間) 283

P

PARTNER 定義、DTP のための 241
PIP データ
 紹介 26
 CPI コミュニケーションでの 26
PRINSYSID オプション、ASSIGN コマンド 270

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 236
 LUTYPE6.1 セッション (CICS-IMS 間) 282
PROGRAM オプション
 リモート・トランザクション定義での 236
PROTECT オプション、START コマンドの 48
PSDINT、システム初期設定パラメーター 182
PSRECOVERY オプション
 CONNECTION 定義 183

Q

QUEUELIMIT オプション、CONNECTION 定義 34,
 297

R

RECEIVE コマンド
 LUTYPE6.1 セッション (CICS-IMS 間) 281
RECOVOPTION オプション
 SESSIONS 定義 183
 TYPETERM 定義 183
REMOTENAME オプション、リモート・リソース定義
 の 214
REMOTESYSNET オプション
 CONNECTION 定義 193, 225
 TERMINAL 定義 193, 225
REMOTESYSTEM オプション
 CONNECTION 定義 193, 225
 TERMINAL 定義 193, 225
 TRANSACTION 定義 235
RETRIEVE コマンド
 CICS-IMS 間通信 279
 WAIT オプション 50
RTIMOUT オプション
 通信プロファイルの 237
 PROFILE 定義 244
RU のチェーン形式 274

S

SEND および RECEIVE、非同期処理 45
 CICS-IMS 間通信 280
SEND コマンド
 LUTYPE6.1 セッション (CICS-IMS 間) 281
SESSION オプション、ALLOCATE コマンドの
 LUTYPE6.1 セッション (CICS-IMS 間) 282
SESSIONS 定義
 RECOVOPTION オプション 183
SNA 経由のシステム間通信
 インストールの注意点 125
 概念 4, 23

SNASVCMG セッション
 目的 27
 CICS による生成 175
START NOCHECK コマンド
 据え置き送信 48
 ローカル・キューのための 49
START PROTECT コマンド 48
START および RETRIEVE の非同期処理
 CICS-IMS 間通信 275
START コマンド
 CICS-IMS 間通信 278
 NOCHECK オプション 47
 ローカル・キューのための 49
START と RETRIEVE の 非同期処理 45
START と RETRIEVE の非同期処理 44
SYSID キーワード、ALLOCATE コマンドの
 LUTYPE6.1 セッション (CICS-IMS 間) 282
sysidnt
 リモート・システムの 163
 ローカル CICS システムの 163
 applid との関係 163
SYSIDNT 値
 デフォルト 163
 リモート・システムの 163
 ローカル CICS システム 163
 ローカル CICS システムの 163
 NETNAME へのマッピング 164

T

TASKREQ オプション
 リモート・トランザクション定義での 236
TCPIP、システム初期設定パラメーター 29
TCP/IP (Transport Control Protocol/Internet Protocol
) 23, 28, 132, 320
TCP/IP (Transport Control Protocol/Internet Protocol) 4
TCP/IP 管理および制御
 概説 209
TCP/IP 経由のシステム間通信
 インストールの注意点 132
TCTTE、サロゲート 270
TERMINAL 定義
 REMOTENAME オプション 234
 REMOTESYSNET オプション 225
 REMOTESYSTEM オプション 225
TOR (端末専有領域) 65
 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 133
 領域間のセッション・バランシング 133
TRANSACTION 定義
 ACTION 属性 313
 WAIT 属性 313

TRANSACTION 定義 (続き)
 WAITTIME 属性 313
Transport Control Protocol/Internet Protocol (TCP/IP) 4,
 23
TRPROF オプション
 リモート・トランザクション定義での 236
 ルーティング・トランザクション (CRTE) の 94
TWASIZE オプション
 リモート・トランザクション定義での 236
TYPETERM 定義
 RECOVOPTION オプション 183

U

USERID オプション、ASSIGN コマンド 270

V

VLVB 形式 274
VTAM
 限定リソース 27
 持続セッション
 アプリケーション・プログラムへの影響 343
 リカバリーと再始動への影響 342
 リンク定義 182
 MRO と ISC リンク 341
 XRF との比較 341
 総称リソース
 アウトバウンドの LU6 接続 156
 インストール 137
 概説 20
 シスプレックス間通信 141
 制約事項 154
 非自動インストール接続での使用 155
 非自動インストール端末での使用 155
 へのマイグレーション 138
 要件 133
 モードグループ 27, 126
 類縁性の終了 146
 APPN ネットワーク・ノード 134
 LOGMODE 項目 27, 126, 174

W

WAIT オプション 312
 RETRIEVE コマンドの 50
WAIT コマンド
 LUTYPE6.1 セッション (CICS-IMS 間) 282
WAIT 属性
 TRANSACTION 定義 313
WAITIME 属性
 TRANSACTION 定義 313

X

XALTENF、グローバル・ユーザー出口 47, 74, 95, 228

XCF (システム間カップリング・ファシリティー)

概説 14

システム間 MRO のための 122

XCF/MRO (システム間 MRO)

概説 14

サポートの生成 122

ハードウェア要件 122

XICTENF、グローバル・ユーザー出口 47, 74, 95, 228

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 34

XZIQUE との使用 299

XPCREQC、グローバル・ユーザー出口 101

XPCREQ、グローバル・ユーザー出口 101

XRF (拡張回復機能) 339

XRF (拡張リカバリー機能)

持続セッションとの比較 341

applid、総称と特定 196

XRF、システム初期設定パラメーター 182

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 34,
299

呼び出される時点 299

XISCONA との使用 299

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。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
IBM World Trade Asia Corporation
Intellectual Property Law & Licensing

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

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。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。

プログラミング・インターフェース情報

本書の目的は、CICS システムが相互に（または他のシステムと）通信できるようにするための方法を理解することにあります。また、汎用プログラミング・インターフェースおよびそれに関連するガイダンス情報も記載されています。お客様は、汎用プログラミング・インターフェースを使用して、CICS のサービスを取得するプログラムを作成することができます。

汎用プログラミング・インターフェースおよびそれに関連するガイダンス情報が記載されている個所では、パート、章、またはセクションの冒頭の文でそのことが示されます。

商標

以下は、IBM Corporation の商標です。

ACF/VTAM	DFS	RACF
APPN	ES/9000	SAA
BookManager	ESCON	System/370
BookMaster	IBM	System/390
CICS	IMS	Systems Application Architecture
CICS/400	IMS/ESA	VSE/ESA
CICS/ESA	MVS	VTAM
CICS/MVS	MVS/ESA	iSeries
CICS/VM	MVS/XA	z/OS
CICS/VSE	OS/390	zSeries
CICSplex	PR/SM	
DB2	Parallel Sysplex	

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、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

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。



プログラム番号: 5655-M15

SC88-4373-00



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