

CICS ファミリー



## プロダクト間通信ガイド



CICS ファミリー



## プロダクト間通信ガイド

お願い

本書の情報およびそれによってサポートされる製品を使用する前に、79 ページの『特記事項』に記載する一般情報をお読みください。

本書は、以下のライセンス・プログラムに適用されます。

- CICS Transaction Server for z/OS バージョン 2 (プログラム番号 5697-E93)
- CICS Transaction Server for OS/390 (プログラム番号 5655-147)
- CICS Transaction Server for VSE/ESA (プログラム番号 5648-054)
- CICS/VSE バージョン 2 (プログラム番号 5686-026)
- CICS Transaction Server for Windows、バージョン 5.0 (プログラム番号 5724-D05)
- CICS Transaction Server for iSeries (プログラム番号 5722-DFH)
- CICS/400 バージョン 4 (プログラム番号 5769-DFH)
- TXSeries for Multiplatforms バージョン 5.0 (部品番号 5724-B44)
- TXSeries for HP-UX バージョン 4.2 (プログラム番号 5801-AAR)

本書は、第 8 版「CICS ファミリー プロダクト間通信ガイド」(SC88-9084-00) を基にしています。第 7 版からの変更箇所の左端には、変更を示す縦線が付けてあります。

資料のご注文方法については、<http://www.ibm.com/jp/manuals> の「ご注文について」をご覧ください。(URL は、変更になる場合があります)

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

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

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

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

(URL は、変更になる場合があります)

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： SC34-6267-00  
CICS Family  
Interproduct Communication

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 2004.4

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

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

© Copyright International Business Machines Corporation 1992, 2003. All rights reserved.

© Copyright IBM Japan 2004

# 目次

まえがき	vii
本書について	vii
本書の対象読者	vii
本書を理解するための前提知識	vii
用語	vii
変更の要約	xi
第 9 版の変更内容	xi
第 8 版の変更内容	xi
第 7 版の変更内容	xi
第 6 版の変更内容	xi
第 5 版の変更内容	xii
第 4 版の変更内容	xii

---

## 第 1 部 CICS プロダクト間通信の紹介 . . . . . 1

第 1 章 CICS プロダクト間通信の紹介	3
資料について	3
第 2 章 CICS 通信のサポート	5
プロダクトの通信機能とは何か	5
CICS システム間通信機能	5
通信プロトコル	6
同期	6
データ変換	7
CICS プロダクト間通信サポート	8
System/390 CICS プロダクト間通信	8
CICS Transaction Server for Windows プロダクト間通信	8
オープン・システム CICS プロダクト間通信	9
CICS/400 プロダクト間通信	10
第 3 章 CICS クライアント	11
CICS クライアントが提供する機能	11
外部呼び出しインターフェース	11
外部表示インターフェース	12
外部セキュリティー・インターフェース	12
端末エミュレーション	12
CICS クライアント	13
サポートされる機能とプロトコル	13
第 4 章 データ変換	15
数字データ	15
文字データ	15
コード・ページ	16
第 5 章 SNA 通信用の CICS の構成	19
SNA 用語の紹介	19
SNA の概念	20
SNA プロダクト	22
SNA 構成の準備	22

パラメーターの一致 . . . . .	23
プラットフォーム特定のインプリメンテーション . . . . .	23
シナリオ . . . . .	23
構成についての詳細情報 . . . . .	23
メインフレーム・ホストの構成 . . . . .	24
AIX マシンの構成 . . . . .	25
CICS for SNA の構成 - 次のステップ . . . . .	26

---

## 第 2 部 CICS 相互通信機能. . . . . 29

<b>第 6 章 CICS 相互通信機能の紹介</b> . . . . .	31
CICS 相互通信機能の要約 . . . . .	31
機能シップ . . . . .	31
トランザクション・ルーティング . . . . .	32
分散プログラム・リンク . . . . .	32
分散トランザクション・プログラミング . . . . .	32
相互通信機能の選択 . . . . .	33
<b>第 7 章 機能シップ</b> . . . . .	35
機能シップの紹介 . . . . .	35
アプリケーションの透過性 . . . . .	36
アクセスできるリモート・リソース . . . . .	36
CICS ファイル制御データ・セット . . . . .	37
IMS データベース . . . . .	37
一時記憶および一時データ . . . . .	38
機能シップの動作の仕方 . . . . .	38
変換プログラム . . . . .	39
ミラー・トランザクション . . . . .	39
同期 . . . . .	41
機能シップの例 . . . . .	41
<b>第 8 章 トランザクション・ルーティング</b> . . . . .	43
トランザクション・ルーティングの紹介 . . . . .	43
トランザクション・ルーティングの開始 . . . . .	44
端末開始のトランザクション・ルーティング . . . . .	44
自動トランザクション開始 . . . . .	46
中継プログラム . . . . .	48
基本マッピング・サポート . . . . .	49
ルーティング・トランザクション (CRTE) . . . . .	49
<b>第 9 章 分散プログラム・リンク</b> . . . . .	51
DPL の紹介 . . . . .	51
DPL を使用する理由 . . . . .	52
同期 . . . . .	52
DL/I および SQL データベース . . . . .	53
DPL を使用するときの制限 . . . . .	53
DPL を使用するとき発生する異常終了 . . . . .	53
<b>第 10 章 非同期処理</b> . . . . .	55
非同期処理の紹介 . . . . .	55
例 . . . . .	56
非同期処理方式 . . . . .	56
START/RETRIEVE コマンドを使用した非同期処理 . . . . .	57

リモート・トランザクションの開始および取り消し	57
START コマンドを使用した情報の受け渡し	58
システム間 START 要求のパフォーマンスの向上	58
作業論理単位への開始要求配布の組み込み	59
NOCHECK オプションが指定された START 要求の送信の据え置き	59
リモート・トランザクション用 START コマンドのローカル・キューイング	60
開始されたトランザクションによるデータ検索	60
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	61
システム・プログラミングに関する考慮事項	61
非同期処理の例 (NOCHECK を使用)	62
<b>第 11 章 分散トランザクション・プログラミング</b>	<b>63</b>
分散トランザクション・プログラミングを使用する理由	63
機能シップの制限	63
分散トランザクション・プログラミングの利点	65
会話	65
会話の開始およびトランザクションの階層	65
アプリケーションの設計	67
制御フロー	67
会話の状態およびエラーの検出	67
同期	68
EXEC CICS と CPI コミュニケーションのどちらを使用すべきか ?	70
2 つの API についてのその他の注意事項	71

---

## 第 3 部 付録 . . . . . 73

<b>参考文献</b>	<b>75</b>
CICS ファミリーの相互通信に関する資料	75
System/390 CICS の相互通信に関する資料	75
CICS Transaction Server for z/OS バージョン 2 リリース 3	75
CICS Transaction Server for z/OS バージョン 2 リリース 2	75
CICS Transaction Server for OS/390 リリース 3	75
CICS Transaction Server for VSE/ESA リリース 1.1.1	75
CICS/VSE バージョン 2	75
CICS 非 System/390 の相互通信に関する資料	75
CICS Transaction Gateway および CICS Universal Client	75
CICS 以外の資料	76
SNA 資料	76
最新の資料かどうかの判断	76
<b>アクセシビリティ</b>	<b>77</b>
<b>特記事項</b>	<b>79</b>
商標	80
<b>索引</b>	<b>81</b>





---

## まえがき

---

### 本書について

本書は、CICS ファミリー・メンバー間の相互通信を主題として解説しています。また、使用できる機能とシステムの構成方法についても示しています。

本書で説明している CICS プロダクトは以下のものです。

- System/390<sup>®</sup> CICS<sup>®</sup>
- CICS Transaction Server for Windows<sup>®</sup>
- オープン・システム CICS
- CICS/400
- CICS クライアント

**重要:** これらの用語は、CICS プロダクトのサブセットに関して一般的に使用されるものです。用語の意味については、『用語』で定義しています。

---

### 本書の対象読者

本書は、各種の CICS プロダクト間の通信の計画およびインプリメンテーションに関係する担当者を対象としています。

---

### 本書を理解するための前提知識

CICS の機能について、またユーザーの CICS システムの操作環境の通信機能について、読者は一般的な知識をもっている必要があります。CICS プロダクト間通信をインプリメントするためには、ご使用の CICS プロダクトの相互通信ガイドに書かれた、プロダクト固有の詳細な情報も必要になります。それぞれの CICS プロダクトの文書化計画については、3 ページの『第 1 章 CICS プロダクト間通信の紹介』で説明しています。

---

### 用語

本書において、特定のプロダクトまたはバージョン・レベルを指定せずに『CICS』という用語を使用している場合は、すべての CICS ファミリー・プロダクトの総称として解釈してください。

以下の CICS プロダクトは、System/370、System/390、または zSeries ファミリーのコンピューターで稼働し、他のハードウェア・プラットフォームで稼働する CICS プロダクトとの通信をサポートします。(これらのプロダクトのすべてが、上記のコンピューターで稼働するわけではありません。たとえば、CICS Transaction Server for z/OS<sup>™</sup> バージョン 2 は、System/370 では稼働しません。)

- CICS Transaction Server for z/OS バージョン 2 (プログラム番号 5697-E93)
- CICS Transaction Server for OS/390<sup>®</sup> バージョン 1 (プログラム番号 5655-147)
- CICS Transaction Server for VSE/ESA<sup>™</sup> (プログラム番号 5648-054)
- CICS/VSE バージョン 2 (プログラム番号 5686-026)

本書において **System/390** という用語は、上記のいずれかのプロダクトが稼働できる System/370、System/390、または zSeries コンピューターを指すために使用しています。**非 System/390** という用語は、上記以外の CICS プロダクトが使用するハードウェア・プラットフォームを示します。上記以外の CICS プロダクトには、たとえば、iSeries (CICS/400 で使用)、IBM 互換パーソナル・コンピューター (CICS Transaction Server for Windows で使用)、および RISC System/6000 (オープン・システム CICS で使用) があります。

System/390 ハードウェア・プラットフォームで稼働するそれぞれの CICS プロダクトに適用される記述では、それらのすべてを表すために総称用語 **System/390 CICS** を使用しています。これらの CICS プロダクトの 1 つが名前では呼ばれるのは、インターフェースの比較において非 System/390 プロダクトへのそのインターフェースが、他の System/390 プロダクトと異なる場合だけです。例外として明示的に述べる場合がありますので、CICS を指すすべての箇所は、ご使用の System/390 CICS プロダクトに適用されるものとして解釈してください。

バージョン番号を限定せずに *CICS Transaction Server for z/OS* という用語を使用する場合は、以下についての総称用語として使用しています。

- 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

バージョン番号を限定せずに *CICS Transaction Server for OS/390* という用語を使用する場合は、CICS Transaction Server for OS/390 リリース 3 のことを指します。

*CICS Transaction Server for Windows* という用語は、特に指定がない限り、CICS Transaction Server for Windows、バージョン 5.0 のことを示します。

オープン・システム *CICS* という用語は、以下についての総称名として使用しています。

- TXSeries for Multiplatforms バージョン 5.0。これには以下のものが含まれていません。
  - CICS for AIX
  - CICS for HP-UX
  - CICS for Sun Solaris
  - CICS for Windows NT
- TXSeries for AIX バージョン 4.3 (CICS for AIX を含む)
- TXSeries for Sun Solaris バージョン 4.3 (CICS for Sun Solaris を含む)
- TXSeries for Windows NT バージョン 4.3 (CICS for Windows NT を含む)
- TXSeries for HP-UX バージョン 4.2 (CICS for HP-UX を含む)

これらのプロダクトを区別する必要があるときは、完全なプロダクト名を示しています。

*CICS Transaction Server for VSE/ESA* という用語は、CICS Transaction Server for VSE/ESA リリース 1.1.1 を意味します。

*CICS/VSE* という用語は、CICS for VSE/ESA バージョン 2 リリース 3 を意味します。

*CICS/400* という用語は、以下についての総称名として使用しています。

- CICS/400 バージョン 4 リリース 5

- CICS Transaction Server for iSeries

| *CICS* クライアント という用語は、下記のプロダクトの総称名として使われます。

- | • CICS Universal Client (Windows NT、Windows 2000 および Windows XP 用)
- | • CICS Transaction Gateway プロダクトの CICS クライアント・エレメント
- | • CICS Transaction Gateway プロダクトのクライアント・デーモン

**CICS-CICS Transaction Server for Windows** という表記は、いずれかの方向での通信を指すために使用しています。1 つの方向だけの通信を特定するには、矢印を追加しています。たとえば、「CICS - オープン・システム CICS」機能シップは、CICS からオープン・システム CICS、またはオープン・システム CICS から CICS への機能シップを表します。CICS/400▶CICS 機能シップは、CICS/400 から CICS への機能シップだけを指します。



---

## 変更の要約

本書は、第 8 版「CICS プロダクト間通信ガイド」(SC88-9084-00) を基にしています。第 7 版から変更された個所には、左マージンに縦線を付けてマークしていません。

ここでは、以下の各版で行われた変更について簡単に記載しています。

---

### 第 9 版の変更内容

この版の重要な変更内容は次のとおりです。

- この資料は、次の新規プロダクトをサポートするために改訂されています。
  - CICS Transaction Server for z/OS バージョン 2 リリース 3
  - CICS Transaction Server for Windows、バージョン 5.0
- 以下の CICS プロダクトへの言及は、これらのプロダクトがサポートされなくなったため、除去されました。
  - CICS Transaction Server for z/OS バージョン 2 リリース 1
  - CICS Transaction Server for OS/390 リリース 2
  - CICS Transaction Server for OS/390 リリース 1
  - CICS Transaction Server for VSE/ESA リリース 1.0
  - CICS/ESA バージョン 4.1
  - CICS Transaction Server for OS/2 Warp バージョン 4
  - CICS for OS/2 バージョン 3.1

---

### 第 8 版の変更内容

この版の重要な変更内容は次のとおりです。

- 本書は、次の新規プロダクトについての記述を組み込むように改訂されました。
  - CICS Transaction Server for z/OS バージョン 2 リリース 2
- 新規機能、TCP/IP での ECI のサポートについては、14 ページの表 5 で説明しています。

---

### 第 7 版の変更内容

この版の重要な変更内容は次のとおりです。

- 本書は、次の新規プロダクトについての記述を組み込むように改訂されました。
  - CICS Transaction Server for z/OS バージョン 2 リリース 1
- CICS/ESA バージョン 3 への言及は、このプロダクトがサポートされなくなったため、除去されました。

---

### 第 6 版の変更内容

CICS Transaction Server for VSE/ESA への言及が追加されました。

---

## 第 5 版の変更内容

第 5 版の主な変更内容は、次に示すプロダクトのサポートです。

- CICS for HP-UX
- Transaction Server for Windows NT、バージョン 4

---

## 第 4 版の変更内容

第 4 版の主な変更内容は、次に示すプロダクトのサポートです。

- CICS Transaction Server for OS/390 リリース 1
- CICS for OS/2 バージョン 3.0
- オープン・システム CICS バージョン 2 リリース 1

新しい章（19 ページの『第 5 章 SNA 通信用の CICS の構成』）では、SNA 通信用の CICS の構成についてその概念と実施方法を説明しています。

---

## 第 1 部 CICS プロダクト間通信の紹介

第 1 部は次の章で構成されています。

- 3 ページの『第 1 章 CICS プロダクト間通信の紹介』では、CICS プロダクト間通信を紹介し、それぞれの CICS プロダクトで使用される文書化計画について説明しています。
- 5 ページの『第 2 章 CICS 通信のサポート』では、CICS 相互通信機能について説明し、2 つの CICS システム間でサポートされる機能を示します。
- 11 ページの『第 3 章 CICS クライアント』では、CICS クライアントのプロダクトを紹介し、各 CICS サーバーでサポートされる機能を示します。
- 15 ページの『第 4 章 データ変換』では、データ変換が必要な理由を説明し、データ変換の概念と用語について解説します。
- 19 ページの『第 5 章 SNA 通信用の CICS の構成』では、SNA 通信用のシステム構成の概念と実施方法を紹介します。





---

## 第 1 章 CICS プロダクト間通信の紹介

本書では、CICS システムの接続方法を解説し、その接続された CICS システムが使用できる機能について説明します。以下の CICS システムのグループについて説明します。

- System/390 CICS
- CICS Transaction Server for Windows
- オープン・システム CICS
- CICS/400<sup>®</sup>

---

### 資料について

本書では、すべての CICS プロダクトの通信方法についての概要を説明しています。各 CICS プロダクトには、そのプロダクト固有の相互通信についての資料があります。この資料には、プロダクトの構成方法と使用可能な機能が詳しく説明されています。

すべての CICS プロダクトの構成と関連資料については、4 ページの図 1 に図解しています。ここでは、同じタイプの他のプロダクトと通信している各 CICS プロダクトと、さらにすべてのプロダクトが相互接続している状態を示しています。図中の白抜きの番号は、次の資料の番号を示しています。

1. *CICS on System 390 Intercommunication Guide*。(これは、ご使用の System/390 CICS プロダクトの相互通信ガイドです。)
2. *CICS ファミリー: System/390 CICS からの通信*
3. *CICS TS for Windows, 相互通信ガイド*
4. *Open System Intercommunication Guide*
5. *CICS/400 相互通信*

上記資料の資料番号は、75 ページの『参照文献』にリストされています。

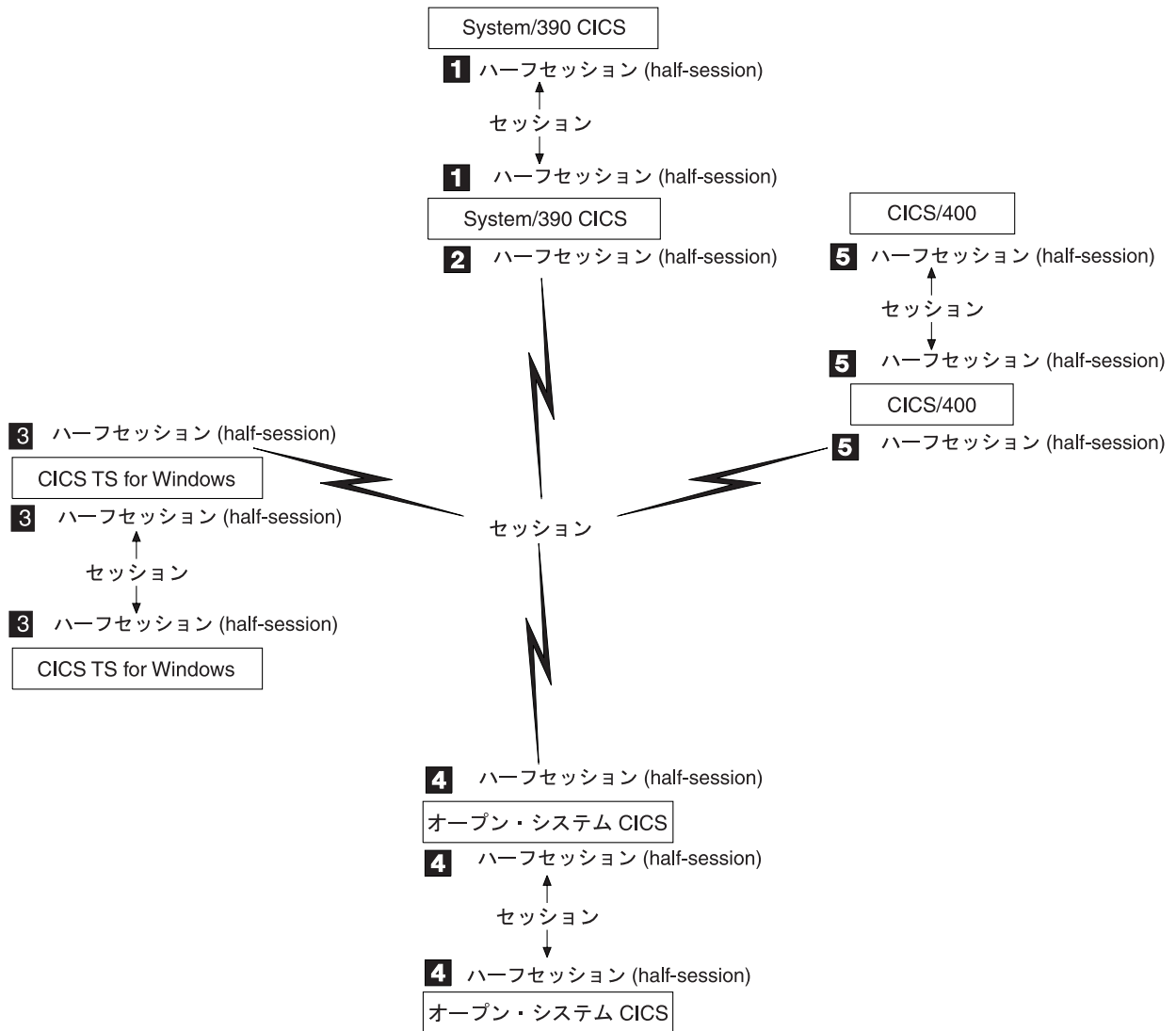


図 1. CICS システム間のセッション：この図は、各 CICS 相互通信の資料の適用範囲を示しています。番号は、3 ページの『資料のリスト』中の資料を表しています。

---

## 第 2 章 CICS 通信のサポート

本章は、2 つのセクションで構成されています。

- 『プロダクトの通信機能とは何か』では、システム間通信の用語と、CICS 通信機能を紹介します。
- 8 ページの『CICS プロダクト間通信サポート』では、2 つの CICS プロダクトが相互に通信する方法を示します。

---

### プロダクトの通信機能とは何か

異なる 2 つの CICS システム間の通信機能を設計する場合、次の項目について考える必要があります。

- CICS プロダクトがサポートするシステム間通信機能
- サポートする通信プロトコル
- サポートする同期レベル
- 必要となるデータ変換サポート

上記の項目については、次のセクションで説明します。

### CICS システム間通信機能

CICS システム間通信では、次の 5 つの基本機能をサポートしています。

#### 機能シッブ

ある CICS システム内で稼働中のアプリケーション・プログラムが、別の CICS システムが所有するデータ・リソース (ファイルやキューなど) にアクセスできるようにします。

#### トランザクション・ルーティング

ある CICS システムに接続されている端末が、別の CICS システム内のトランザクションを稼働できるようにします。

#### 分散プログラム・リンク (DPL)

ある CICS システム内で実行しているアプリケーション・プログラムが、別の CICS システム内のプログラムとリンクする (制御権を渡す) ことができるようにします。リンクされたプログラムが実行され、リンクする側のプログラムに結果を戻します。

#### 非同期処理

ある CICS システム内で実行しているトランザクションが、別のシステム内のトランザクションを開始できるようにします。2 つのトランザクションは、その後独立して実行します。

#### 分散トランザクション・プログラミング

ある CICS システム内で稼働しているトランザクションが、別のシステム内で稼働しているトランザクションと通信できるようにします。この場合のトランザクションは、相互に通信を行うために特別に設計され、コーディングされたものです。

これらの機能の詳細については、29 ページの『第 2 部 CICS 相互通信機能』を参照してください。

## 通信プロトコル

2 つのシステムの通信が成功するには、両方のシステムが理解できる共通の規則を使用しなければなりません。通信プロトコルとは、たとえば要求と応答の標準や送信できる順序などを定義する規則のセットのことです。

CICS プロダクトの場合には以下の 3 つの通信プロトコルが重要です。

### SNA LU TYPE 6.2

LU TYPE 6.2 (LU 6.2) は、システム間通信およびシステムと装置間の通信の両方をサポートするシステム・ネットワーク体系 (SNA) プロトコルです。LU 6.2 は、拡張プログラム間通信機能 (APPC) でもあります。

CICS では、SNA フローが TCP/IP ネットワークを使って伝送できるようにする AnyNet<sup>®</sup> プロダクトを使用することができます。

すべての CICS プロダクトは、LU 6.2 プロトコルをサポートします。

### TCP/IP

伝送制御プロトコル/インターネット・プロトコル (TCP/IP) は、LAN (ローカル・エリア・ネットワーク) と広域ネットワーク (WAN) の両方で使用できる一連のプロトコルです。

TCP/IP は、CICS Transaction Server for Windows、オープン・システム CICS、および CICS Transaction Server for z/OS のバージョン 2.2 以降のバージョンでネイティブにサポートされます。

**注:** CICS TS for z/OS バージョン 2.2 以降は、クライアントへのネイティブ TCP/IP 接続をサポートしています。CICS 外部呼び出しインターフェース (ECI) はサポートしますが、外部表示インターフェース (EPI) と外部セキュリティ・インターフェース (ESI) はサポートしません。14 ページの表 5 を参照してください。

CICS TS for VSE/ESA および CICS/VSE<sup>®</sup> 2.3 を除くすべての System/390 CICS プロダクトでは、AnyNet プロダクトを利用して、SNA フローを TCP/IP ネットワークを使って伝送できます。

### NetBIOS

ネットワーク基本入出力システム (NetBIOS) は、パーソナル・コンピューター用のローカル・エリア・ネットワーク (LAN) プロトコルです。このプロトコルは CICS Transaction Server for Windows でサポートされます。

IBM<sup>®</sup> NetBIOS を使用することも、別の NetBIOS エミュレーターを使用することもできます。たとえば、Novell の NetBIOS エミュレーターは、Internetwork Packet Exchange (IPX) プロトコルを介して NetBIOS フローを提供します。

## 同期

CICS プロダクト間通信中に、パートナー・トランザクションが、データ・ストア、すなわちデータ・セット、データベース、一時記憶、一時データなどに論理的に関連する更新を行い、両方のトランザクションが、そのリソースに対して行った更新をコミット (またはバックアウト) しない場合は、データ保全性は失われます。

データ保全性を確実にするために使用される処理を同期と呼びます。同期は、そのパートナー・トランザクションが異常終了して更新をバックアウトするとき、もう

一方のトランザクションが正常終了してその更新内容をコミットしてしまうことを防ぎます。同期はまた、ネットワーク上の問題によって、トランザクションが相互通信して互いにそのアクションの情報交換を行うことができなくなった場合に、その状況の処理も行う必要があります。

## 同期レベル

SNA APPC 体系では、3つのレベルの同期が定義されており、このレベルは、NONE、CONFIRM、SYNCPOINT と呼ばれています。CICS では、この3つのレベルをそれぞれ、同期レベル 0、同期レベル 1、同期レベル 2、と呼びます。

**レベル 0**      同期サポートはありません。

**レベル 1**      トランザクションが、ある段階の同期を指定するために使用する確認要求を交換できるようにします。CICS は、この処理には関与していません。

**レベル 2**      システム・レベルの同期点の交換を行います。

同期レベル 2 では、**2 フェーズ・コミット**が使用できます。2 フェーズ・コミット処理では、1つの CICS システムが同期点処理を開始し、そのシステムが操作の調整者となります。調整側システムは、下記の処理を行います。

1. コミットの**準備**をするように、接続された各システムに要求します。
2. 各システムが準備のできたことを知らせると、**コミット**するように各システムに要求します。あるいは リソース・マネージャーがコミットできないことを知らせた場合は、**バックアウト**するように各システムに要求します。

2つの接続された CICS システムが使用できる同期レベルは、その2つのシステムが最初に接続を行ったときに設定されます。同期レベル 2 で確立された接続は、同期レベル 0、1、または 2 の会話をサポートします。また、同期レベル 1 で確立された接続は、同期レベル 0 または 1 の会話をサポートすることができます。

使用できる同期レベルは、CICS システム固有の機能と、システムが接続に使用しているネットワークの能力によって決まります。互いに接続された2つの CICS システムすべての同期レベルは、8ページの『CICS プロダクト間通信サポート』に要約されています。

## データ変換

CICS プロダクトでは、文字データを表現するために2つの交換コード、EBCDIC および ASCII を使用します。System/390 CICS プロダクトおよび CICS/400 におけるデータは、EBCDIC 形式で保持されます。CICS Transaction Server for Windows およびオープン・システム CICS におけるデータは、一般に ASCII 形式で保持されます。

2つのプロダクト間でシステム間通信機能をサポートするには、必要なデータ変換をサポートする必要があります。詳細については、15ページの『第4章 データ変換』を参照してください。

## CICS プロダクト間通信サポート

このセクションでは、各 CICS プロダクトが、同じタイプの他の CICS プロダクトと通信しているとき、および他の CICS ファミリー・プロダクトのいずれかと通信しているときに提供するサポートについて要約します。

### System/390 CICS プロダクト間通信

#### サポートされる機能

System/390 CICS ではすべての CICS システム間通信機能をサポートします。

#### サポートされる通信プロトコル

すべての System/390 CICS プロダクトで SNA がサポートされます。

CICS TS for VSE/ESA および CICS/VSE 2.3 を除くすべての System/390 CICS プロダクトでは、AnyNet プロダクトを利用して、SNA フローを TCP/IP ネットワークを使って伝送できます。

CICS TS for z/OS バージョン 2.2 以降だけは、クライアントへのネイティブ TCP/IP 接続をサポートしています。CICS 外部呼び出しインターフェース (ECI) はサポートしますが、外部表示インターフェース (EPI) と外部セキュリティー・インターフェース (ESI) はサポートしません。14 ページの表 5 を参照してください。

#### サポートされる同期レベル

System/390 CICS は同期レベル 0、1、および 2 をサポートすることができます。使用する同期レベルは、パートナー・システムが使用可能なサポートによって決まります。これは、表 1 に要約しています。

表 1. System/390 CICS がサポートする同期レベル

	System/390 CICS	オープン・ システム CICS	CICS/400	CICS Transaction Server for Windows
サポートされる最大同期レベル	2	2	2	1

### CICS Transaction Server for Windows プロダクト間通信

#### サポートされる機能

CICS Transaction Server for Windows ではすべての CICS システム間通信機能をサポートします。

#### サポートされる通信プロトコル

CICS Transaction Server for Windows は、SNA、TCP/IP および NetBIOS をサポートします。使用できる通信プロトコルは、パートナー・システム内で使用可能なサポートによって決まります。これは、9 ページの表 2 に要約しています。

表 2. CICS Transaction Server for Windows がサポートする通信プロトコル

	CICS Transaction Server for Windows	オープン・システム CICS	System/390 CICS	CICS/400
サポートされる通信プロトコル	SNA TCP/IP NetBIOS	SNA TCP/IP	SNA	SNA
注: 分散トランザクション処理は、SNA によってのみサポートされます。				

### サポートされる同期レベル

CICS Transaction Server for Windows では、同期レベル 0 および 1 のみをサポートし、どちらのレベルもすべてのパートナー CICS システムで使用できます。

## オープン・システム CICS プロダクト間通信

### サポートされる機能

オープン・システム CICS ではすべての CICS システム間通信機能をサポートします。

### サポートされる通信プロトコル

オープン・システム CICS では、SNA と次の 2 つのタイプの TCP/IP をサポートします。

- CICS ファミリー TCP/IP。**他のオープン・システム CICS 領域、CICS Transaction Server for Windows、および CICS クライアントへの通信に使用します。
 

CICS ファミリー TCP/IP サポートでは、同期レベル 0 および 1 のみサポートします。またこのプロトコルは分散トランザクション処理には使用できません。
- Encina® PPC TCP/IP。**他のオープン・システム CICS 領域との通信に使用します。
 

Encina PPC TCP/IP では、同期レベル 0、1 および 2 をサポートします。またこのプロトコルは分散トランザクション処理に使用できます。

通信プロトコルの選択は、パートナー・システムによってサポートされる機能によって決まります。これは、表 3 に要約しています。

表 3. オープン・システム CICS がサポートする通信プロトコルと同期レベル

	オープン・システム CICS	CICS Transaction Server for Windows	System/390 CICS	CICS/400
サポートされる通信プロトコル	SNA TCP/IP (1) TCP/IP (2)	SNA TCP/IP (1)	SNA	SNA
サポートされる最大同期レベル	2	1	2	2
注: 1. TCP/IP (1) - CICS ファミリー TCP/IP サポート 2. TCP/IP (2) - Encina PPC TCP/IP サポート				

### サポートされる同期レベル

オープン・システム CICS では、SNA および Encina PPC TCP/IP を使用している場合は、同期レベル 0、1 および 2 をサポートし、CICS ファミリーを使用している場合は、同期レベル 0 および 1 をサポートできます。オープン・システム CICS が他の CICS システムでサポートする同期レベルは、9 ページの表 3 に要約しています。

## CICS/400 プロダクト間通信

### サポートされる機能

CICS/400 では、すべての CICS システム間通信機能をサポートします。

### サポートされる通信プロトコル

CICS/400 は SNA のみサポートします。

### サポートされる同期レベル

CICS/400 は、同期レベル 0、1、および 2 をサポートします。使用する同期レベルは、パートナー・システムで使用可能なサポートによって決まります。これについては、表 4 に要約しています。

表 4. CICS/400 同期レベル・サポート

	CICS/400	System/390 CICS	オープン・ システム CICS	CICS Transaction Server for Windows
サポートされる最大同期レベル	2	2	2	1



---

## 第 3 章 CICS クライアント

### 用語

本書では、CICS クライアント という用語は、下記のプロダクトの総称名として使用しています。

- CICS Universal Client (Windows NT<sup>®</sup>、Windows 2000 および Windows XP 用)
- CICS Transaction Gateway プロダクトの CICS クライアント・エレメント (13 ページの『CICS クライアント』 にリストされているすべてのオペレーティング・システムで使用可能)
- CICS Transaction Gateway プロダクトのクライアント・デーモン

CICS クライアントを使用すると、ワークステーションは、CICS システム内のトランザクションとリソースにアクセスすることができます。

CICS クライアントそのものは、全機能 CICS システムではありませんが、ネットワーク内の他のマシンで稼働している CICS システムのリソースへのアクセスを可能にする機能が組み込まれています。クライアントと接続する CICS システムは、**CICS サーバー**と呼ばれます。

CICS クライアントには、多くの種類のオペレーティング・システム用のものがあります。これについては、13 ページの『CICS クライアント』で説明しています。

---

## CICS クライアントが提供する機能

CICS クライアントには、クライアント/サーバー・コンピューティング用の機能セットが用意されています。このセクションは、特に重要な機能の概要を説明するもので、詳細な説明はしていません。

## 外部呼び出しインターフェース

外部呼び出しインターフェース (ECI) とは、ワークステーションで稼働中の CICS 以外のプログラムによって CICS サーバーにある CICS プログラムを呼び出すことができるアプリケーション・プログラミング・インターフェース (API) です。外部呼び出しインターフェースによって、プログラムは、たとえばデータベースの検索に使用される既存のサーバー・ルーチンを利用することができます。

クライアント・プログラムは、CICS サーバーに対して以下の呼び出しを行うことができます。

- プログラム・リンク呼び出し。**同期** (つまり、呼び出しプログラムがリンクされたプログラムからの応答を待つ) であることも、**非同期** (つまり、2 つのプログラムが独立して実行を続ける) であることもできます。クライアント・プログラムは、このような呼び出しをいくつか発行することができます。この呼び出しは、すべて同じ作業単位 (UOW) 内で稼働するか、または、別々の作業単位として稼働します。
- 前回の非同期呼び出しからの応答を検索する呼び出し。

- CICS システムの状況を示す値を戻す呼び出し。この呼び出しによって、アプリケーションは、CICS サーバーの使用可能性をテストしたり、状況が変わるまで CICS サーバーを監視したりすることができます。

## 外部表示インターフェース

外部表示インターフェース (EPI) とは、クライアント・プログラムが CICS サーバーに 3270 端末として認識されるようにするための API です。たとえば、EPI を使用することによって、ワークステーション・ユーザーは、サーバー・コードを変更しないで、3270 端末用に書かれた CICS サーバー・トランザクションにアクセスすることができます。

クライアント・プログラムは、CICS トランザクションを開始し、標準 3270 データ・ストリームを、トランザクションへ送信したり、トランザクションから受信したりすることができます。3270 端末をエミュレートするか、Windows (Windows クライアント) などのグラフィカル・ユーザー・インターフェースを使用すれば、3270 データを表現することができます。

EPI は、アプリケーション・プログラムからの一連の呼び出しによって構成されます。呼び出しは、アプリケーションにリンクしているライブラリーに提供されています。利用できる機能には、以下を行うための呼び出しがあります。

- EPI を初期化する
- EPI を終了する
- 端末を接続する CICS サーバーのリストを獲得する
- 疑似端末を接続する
- 疑似端末を切り離す
- 端末用のトランザクションを開始する
- 端末からトランザクションヘデータを送る
- 端末に起きたイベントの詳細を獲得する。イベントとは、たとえば、トランザクションが端末からの応答待ちをしていることなどをいいます。
- 端末に起きた最後のエラーに関する詳細なエラー情報を獲得する

## 外部セキュリティー・インターフェース

外部セキュリティー・インターフェース (ESI) は、非 CICS クライアントのプログラムが、CICS サーバーに接続するためにクライアントで使用されるパスワードを検査および変更できるようにする API です。

## 端末エミュレーション

CICS クライアントは、3270 端末エミュレーターを稼働させることができます。クライアント端末エミュレーターは、標準 CICS トランザクション・ルーティング・フローを、CICS サーバーへ送信したり、CICS サーバーから受信したりすることができます。したがって、ユーザーは、クライアントがローカル接続の 3270 端末であるかのように、サーバーと対話し、トランザクションを稼働させることができます。

1 つのクライアントで複数の端末エミュレーターを稼働させることができます。これらのエミュレーターは、同一の CICS サーバーにも、異なる複数のサーバーにも接続することができます。

ユーザーは、これらのエミュレーターで色やキーボード・マッピングをカスタマイズすることができます。2 バイト文字セット (DBCS) がサポートされますが、CICS/400 サーバーに接続している CICS クライアントは、2 バイト文字セット (DBCS) をサポートしないことに注意してください。

---

## CICS クライアント

次のオペレーティング・システムごとに CICS クライアントがあります。

- AIX<sup>®</sup>
- Microsoft<sup>®</sup> Windows XP
- Microsoft Windows 2000
- Microsoft Windows NT
- Solaris
- HP-UX
- Linux 390

各クライアントは、以下の CICS サーバーのいずれか、あるいはすべてに接続することができます。

- CICS Transaction Server for z/OS
- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA
- CICS/VSE バージョン 2.3
- CICS/400
- CICS Transaction Server for Windows
- オープン・システム CICS

## サポートされる機能とプロトコル

14 ページの表 5 は、各 CICS クライアントと CICS サーバーとの間のリンクでサポートされる機能と通信プロトコルを示したものです。

表 5. IBM CICS クライアントのサポートする機能とプロトコル

CICS サーバー	ECI	EPI	端末 エミュ レーター	自動イン ストール	LU 6.2	NetBIOS	TCP/IP
CICS Transaction Server for Windows	Y	Y	Y	Y	Y	Y	Y
オープン・システム CICS	Y	Y	Y	Y	Y	-	Y
CICS/400	Y	Y	Y	Y	Y	-	-
CICS TS for z/OS バージョン 2.2 以降	Y	Y	Y	Y	Y	-	Y <sup>1 2</sup>
CICS TS for z/OS バージョン 2.1	Y	Y	Y	Y	Y	-	Y <sup>2</sup>
CICS TS for OS/390	Y	Y	Y	Y	Y	-	Y <sup>2</sup>
CICS TS for VSE/ESA	Y	Y	Y	Y	Y	-	-
CICS/VSE V2.3	Y	-	-	Y <sup>3</sup>	Y	-	-

**注:**

- Y** 機能またはプロトコルがサポートされる
- 機能またはプロトコルがサポートされない

**ECI** 外部呼び出しインターフェース

**EPI** 外部表示インターフェース

**自動インストール**

ユーザーは、事前にクライアントとサーバーの接続を定義する必要がない

1. ネイティブ TCP/IP はサポートされます。TCP/IP での ECI の使用により、すべてのクライアントがサポートされますが、ECI だけしか使用できません (EPI および ESI はいずれも使用できません)。
2. TCP/IP は、IBM TCP62 および VTAM<sup>®</sup> の AnyNet フィーチャーの使用によりサポートされます。この方式を使用してサポートされるのは、Windows クライアントだけです。ECI と EPI の両方とも使用できます。
3. 単一セッションの LU6.2 接続しか自動インストールできません。

---

## 第 4 章 データ変換

CICS ファミリーのプロダクトは、さまざまなオペレーティング・システムやハードウェア・プラットフォームで稼働します。数字データや文字データは、システムごとに異なる方法で保持することができます。CICS プロダクトの相互通信でデータ転送が必要になったときには、データ変換を行わなければならないことがあります。

本章ではデータ変換の概要を説明します。これは、ご使用の CICS プロダクト用の CICS 相互通信ガイドに記載されたタスク中心の情報の紹介です。処理のメカニズムについては説明しません。

CICS プロダクトは、必要となるほとんどのデータ変換を自動的に行います。変換には、ユーザーが何もセットアップする必要のないものもあります。たとえば、機能シッパ要求のファイル名などです。たとえば、ファイル・レコードのようなその他のデータの場合は、データ・レコード内の指定のフィールドに適用する変換のタイプを識別するリソース定義をユーザーが指定します。非自動変換の場合、出口、またはユーザー置き換え可能の変換プログラムを一部のプロダクトで使用することができます。

---

### 数字データ

数字データは主に 2 進数とパック 10 進数の形で保持されます。この 2 つのタイプのデータが 2 つの CICS システム内で異なる方法で保持されているとしても、双方のシステムにリソース定義があれば、自動データ変換を行うことができます。

場合によっては、変換の必要をなくすように、一方のシステムにおいて、もう一方のシステムと互換性のある方法でデータを保持するようにしておくことができます。たとえば、CICS Transaction Server for Windows COBOL II アプリケーション・プログラムが IBMCOMP および SIGN EBCDIC の指示でコンパイルされると、パック 10 進データは System/390 と互換性のある形式で保持されます。

自動データ変換を使用できない場合があります。たとえば、ワークステーションのパック 10 進データと System/390 のパック 10 進データにおける変換の場合、ユーザー作成の変換コードが必要です。このコードは、System/390 CICS プロダクト上のユーザー置き換え可能変換プログラムに挿入することができます。

2 つのシステムがデータを交換する場合、どちらの CICS システムがデータ変換の責任があるかについての詳細は、プラットフォーム固有の「CICS 相互通信ガイド」を参照してください。

---

### 文字データ

コンピューターのデータの最小単位は、ビットすなわち 2 進数です。ビットには 0 と 1 の 2 つの値しかありません。文字データを表すためには、ビットをグループ化しなければなりません。最も一般的なグループ化は 8 ビットのバイトで、最大 256 個の異なる文字を表します。

名前付きの**文字セット**は、特定セットの文字です。たとえば、“Latin-1”は、西洋の大文字および小文字、数字、および記号のセレクションのセットです。

**1 バイト文字セット (SBCS)** では、最大 256 個の文字を表すことができます。各文字が 1 バイトで表されます。

**2 バイト文字セット (DBCS)** では、256 個の以上の文字を表すことができます。各文字が 2 バイトで表されます。言語によっては、文字を複数のバイトで表す必要があります。これを、**マルチバイト文字セット (MBCS)** といいます。この文字セットでは、各文字は、最高 4 バイトで表されます。

国によっては (日本など)、SBCS と DBCS の両方を使用する言語もあります。

コンピューター・プログラムまたはオペレーティング・システムでは、表したい各文字に、1 バイト、2 バイト、またはマルチバイトの値を割り当てる必要があります。文字表記のための規則がいくつかあります。本章では、このような規則を交換コードといいます。CICS プロダクトを稼働できるハードウェア・プラットフォームでは、2 種類の交換コードが使用されます。

- System/370™、System/390 および AS/400® マシンで一般に使用される、拡張 2 進化 10 進コード (EBCDIC)。
- パーソナル・コンピューターおよび RISC System/6000® (RS/6000®) マシンで一般に使用される、情報交換用米国標準コード (ASCII)。

## コード・ページ

**コード・ページ**は、特定の文字セットの文字用のコード・ポイントを定義するものです。コード・ページとは、1 バイトまたは 2 バイトの値と、それらが表す文字のリストのことです。EBCDIC 交換コードおよび ASCII 交換コードには、2 つ以上のコード・ページがあります。したがって、2 つのシステムで同じ交換コードを使用している場合であっても、データ変換が必要になることがあります。

**コード化文字セット ID (CCSID)** は、文字セットとそれに関連したコード・ページの組み合わせです。CCSID は複合のこともあります。つまり、複数の文字セットとコード・ページを含む場合があります。例えば、カタカナ (1 バイト) と漢字 (2 バイト) です。

ほとんどの CICS プロダクトは、組み込まれた変換テーブルを使用して、共通コード・ページ間の変換を処理します。プロダクトによっては、ユーザー独自の交換テーブルを定義することができます。標準外の変換の場合、ユーザー作成の変換プログラムを使用することができます。

データ変換の詳細については、ご使用の CICS プロダクトの「CICS 相互通信ガイド」を参照してください。

**注:**

1. **CICS for OS/400®** では、AS/400 によって提供される包括的な変換テーブルを使用しますので、ユーザー定義のテーブルはサポートしません。
2. **オープン・システム CICS** では、テーブル起動およびアルゴリズム方式でデータ変換を行う、オペレーティング・システムの **iconv** ルーチンを使用します。

包括的な変換プログラムが用意されています。オープン・システム CICS ユーザーは、変換プログラムを作成することも、カスタマイズすることもできます。





---

## 第 5 章 SNA 通信用の CICS の構成

2 つのシステムが通信を行うには、それぞれのシステムで次の情報が必要です。

- もう一方のシステムの識別
- もう一方のシステムの特長
- 使用する通信方式
- 使用するサービスと機能

この章では、SNA 通信用に CICS を構成する場合に、上記の情報を指定する方法を示します。

通信用にシステムを構成するプロセスは、次のようないくつかのステージに分かれています。

1. 2 つのシステム間に基本接続を確立し、単純なテスト・アプリケーションが使用できるようにします。
2. 初期構成を構築し、ユーザー・アプリケーションがサポートできるようにします。
3. 作業構成に改良を加えて、機能を拡張し、パフォーマンスを向上させます。

この章では、最初の基本構成を確立するために必要となる SNA および CICS に関して説明します。第 1 段階として、基本的で容易な SNA 接続のテストが正常に終了してから、先に進むことを考えます。

他に、SNA についてより広く、詳細に説明している資料が複数あります。SNA アーキテクチャーについて説明している資料、個々のプロダクトでどのようにアーキテクチャーをインプリメントしているかについて説明している資料、また、異なるシステムのあらゆる組み合わせを構成する方法について説明している資料などがあります。76 ページの『SNA 資料』に、役に立つ資料をリストしています。

ここで説明のために選択したシナリオは、相互に接続された System/390 CICS システムの、確立された SNA ネットワークにもとづいたものです。これに、SNA を使用して System/390 CICS システムのうちのいずれかに接続する、CICS for AIX システムの説明を加えています。

この章の以降の部分では、以下について説明します。

- 『SNA 用語の紹介』
- 22 ページの『SNA 構成の準備』
- 23 ページの『構成についての詳細情報』
- 26 ページの『CICS for SNA の構成 - 次のステップ』

---

### SNA 用語の紹介

IBM のシステム・ネットワーク体系 (SNA) では、システムが通信に使用する一連の規則を定義しています。これらの規則は、システム間をフローするデータのレイアウト、およびデータを受信したときにシステムがとるアクションを定義しています。データのレイアウトは、**フォーマット**といい、データを受信したときにシステムがとるアクションを**プロトコル**といいます。フォーマットとプロトコルは、ともにアーキテクチャーのコンポーネントとなります。

SNA では、システムがアーキテクチャーをインプリメントする方法は指定していません。実際、SNA の主要な目的は、内部ハードウェアおよびソフトウェアの設計が大幅に異なるシステム間で通信できるようにすることなのです。ネットワーク・フローがアーキテクチャー規則に合っていることだけが、必要条件となります。

このように各システムが SNA を別々にインプリメントすると、各システムがその独自の用語をそれぞれの SNA のインプリメンテーションに持ち込む結果にもなり、このことが、異なる 2 つのシステム間で接続の確立を開始するときに、用語の混乱を招くと考えられます。このセクションでは、CICS システムを構成する際に遭遇する、これらの特別な SNA の概念と用語について紹介します。

## SNA の概念

### ネットワーク

ネットワークとは、物理的接続および論理的接続を含む、相互に接続されたコンピュータおよび装置の集合をいいます。

### ネットワーク・アドレス

ネットワーク・アドレスとは、ネットワーク内の各装置に割り当てられた固有のコードです。たとえば、パーソナル・コンピュータでは、そのネットワーク・アダプター・カード内のメディア・アクセス制御 (MAC) アドレスのようなものです。

### リンクおよびノード

リンクは、2 つのノードを接続します。ここで、ノードはデータを伝送および受信するネットワーク内の装置です。

### 論理装置 (LU)

論理装置は、通信データ・フローの論理宛先を表します。LU の正式の定義は、エンド・ユーザーをネットワーク内に登録するための手段であり、エンド・ユーザーはネットワーク内のデータ・フローの最終的なソースまたは宛先として定義されます。SNA は、数種類の異なるタイプの LU をサポートします。論理装置は、3270 ディスプレイ端末用の LU タイプ 2、プリンター用の LU タイプ 4 のように、番号の付いた **LU タイプ** にグループ分けされています。CICS 間通信用の LU タイプは LU タイプ 6.2 で、通常は、**拡張プログラム間通信 (APPC)** で参照しています。それぞれの LU には、ネットワーク内で識別するための固有の名前があり、これを **LU 名** と呼びます。

LU を中に含むネットワークの名前が、LU の名前に付加されることもあります。これは、**ネットワーク修飾 LU 名**、または**完全修飾 LU 名**といい、ネットワーク名.LU 名 という形式をとります。

### 物理装置 (PU)

物理装置は、そのネットワーク・リソースを管理する装置内のハードウェアおよびソフトウェアのコンポーネントです。LU は、PU の中に存在し、1 つの PU で多数の LU を保持することができます。PU には、数種類の異なるタイプがあります。メインフレーム・ホスト内で稼働する VTAM は、PU タイプ 5 で、37x5 ネットワーク・コントローラー内で稼働する NCP は、PU タイプ 4 です。ワークステーションが互いにピアツーピア方式で接続されている場合は、これらのワークステーションは PU タイプ 2.1 として機能します。ワークステーションがメインフレーム・ホストに階層方式で接続されている場合は、そのワークステーションは PU タイプ 2.0 として機能します。PU タイプ 2.1 は、独立

ノードとして記述します (メインフレーム・ホストからは独立したものであるからです)。また、PU タイプ 2.0 は従属ノードになります。

### 制御点 (CP)

LU を PU に関連付ける SNA 概念です。CP は、LU を管理する PU の一部と考えることができます。

### 交換 ID (XID)

PU と CP に関連したものが、交換 ID です。XID は、実際には、PU が接続を確立しようとする初期のステージの間に交換するデータ・フローの名前ですが、SNA 構成の状況において XID は、その中で XID がフローするフィールドの 1 つと見なされます。XID は、PU が互いの ID を確認するために使用する 16 進のフィールドです。

### セッション

SNA は、セッションという用語を使用してネットワーク内のさまざまなタイプのデータ・フローを表します。セッションは他との混同を避けるために、たとえば CP-CP セッションのように、必ずデータ・フロー・タイプの記述で修飾される必要があります。ただし、APPC 用 CICS で使用される場合は、LU 間のデータ・フローであると見なされます。つまり、LU-LU セッションです。2 つの LU 間には通常いくつかのセッションがあり、これらは並列セッションと呼ばれます。

### 接続

CICS では、接続という用語を使用して、2 つの CICS システムを接続する 1 つのセッションのグループを表します。

### トランザクション・プログラム (TP)

SNA では、トランザクション・プログラムという用語は、APPC 環境内のアプリケーション・プログラムを指します。TP は、LU を使用してネットワークにアクセスします。CICS が APPC を使用している場合、TP が CICS トランザクションになります。

### 会話

SNA では、会話という用語で 2 つの TP 間の通信を表します。つまり、2 つの APPC TP が通信中の場合、その 2 つの APPC TP は会話中であるといえます。会話は、LU-LU セッションをフローします。それぞれの会話は、その独自の使用のためのセッションを割り振られます。会話が終了すると、セッションは解放され、また別な会話によって使用されます。2 つの TP 間では、可能な会話は 1 つだけですが、1 つの TP は、別々の TP と複数の会話を行うことができます。

### モード

LU-LU セッションが使用するネットワーク内では、経路およびパスの選択肢が数多くあります。1 つの経路は、大量のバッチ・データに適切であり、別の経路は、少量のデータを高速で交換するために確保する、というようになります。SNA では、これらの異なる経路タイプをモードにグループ化することができます。TP では、最初に会話を確立したときに、適切なモードを選択できます。会話は、ノードによって定義された経路に従って、LU-LU セッションをフローします。

### ローカル、リモート、パートナー

これらの用語は、多くのコンテキストで使用されます。たとえば、ローカル・シ

システム、リモート・システム、パートナー・システムとして参照できます。ユーザーがあるワークステーションにいるとき、ユーザーにとってそのワークステーションがローカル・システムで、ワークステーションが通信する相手のマシンがリモート・システムで、そのリモート・システムがユーザーのパートナーとなります。ただし、これらの用語は、すべてユーザーの参照点にとっては相対的なものです。そのためユーザーのワークステーションをリモート・システムと考えると、先程のリモート・システムがローカル・システムと見なされます。さらに、それぞれのシステムがお互いのパートナーになります。

わかりやすく言えば、セッションを LU をリンクするパイプのようなものと思えば、ここで説明した用語が理解しやすくなります。LU がパイプにデータを挿入すると、そのデータは必然的にパイプのもう一方の端にある LU に渡されます。類似した特性をもつパイプは、モードというグループに分けられます。これらのパイプは、システムが最初に初期設定されて SNA が開始したとき、または TP が最初にパイプの使用を要求したときに作成されます。いくつかのパイプが通常、2 つの CICS システム間で並列に実行されます。CICS トランザクションが別のシステムとの会話を保持したい場合、CICS トランザクションはパイプの使用を要求します。CICS トランザクションは、パイプを単独で使用します。トランザクションが終了すると、パイプはプールに戻され、また次のトランザクションに割り振ることが可能になります。

## SNA プロダクト

CICS は、固有の SNA サポートは提供していません。つまり、CICS はネットワーク・データ・フローを SNA フォーマットに構築せず、しかも、データ・フローを受信したときに SNA プロトコルを開始する責任もありません。これらの SNA 機能は別の SNA プロダクトによって提供され、CICS はそのプロダクトのサービスを使用します。このサービスを提供する SNA プロダクトは、System/390 ホスト上では VTAM、および AIX マシン上では AIX SNA Server/6000 (AIX SNA) になります。

システムを構成する場合、CICS プロダクトとそれに関連する SNA プロダクトの両方を構成する必要があります。

---

## SNA 構成の準備

各システムに SNA 接続を確立するには、以下のことを行う必要があります。

1. システムをネットワークに定義する。
2. パートナーへの接続を定義する。
3. リモート・システムにローカル・システムの定義があることを確認する。

つまり、第 1 ステップは、システムがその名前とアドレスをネットワークに登録すること、第 2 ステップは、ユーザーのシステムがそのパートナーにするシステムの名前とアドレスを指定し、ネットワークがそのシステムとのリンクを確立できるようにすること、第 3 ステップは、ユーザーのパートナーがユーザーのシステムの名前を登録し、ユーザーのシステムからの要求を受信したときにその要求を認識できるようにすることです。

## パラメーターの一致

システムのリソースを定義するために使用する名前には、ネットワーク名、LU 名、PU 名、CP 名、および XID があります。

これらの名前は、CICS プロダクトの構成時にパラメーターとして使用されるものと、SNA プロダクトの構成時にパラメーターとして使用されるものがあります。また、両方のプロダクトで使用されるものもあり、ローカル・システム上のあるプロダクトで使用されるものと、リモート・システム上の別のプロダクトで使用されるものがあります。1 つのパラメーターが複数の場所で使用される場合、同じ名前を使用することが重要です。つまり、パラメーターは一致していなければなりません。

### モード名

パートナーは、同一のモード名を使用する必要があります。どのようなモードでも使用できますが、**#INTER** 設計済みモードを使用すると便利です。

### 別名

リソースのなかには (代表的なものは LU)、実名と同時に別名またはローカル名をもっているものがあります。別名は、ローカル・システム内で内部的にのみ使用されます。パートナー・システムは実名のみを認識し、その実名はすべてのシステムを通じて一致していなければなりません。

## プラットフォーム特定のインプリメンテーション

特定のネットワーク構成および使用する SNA プロダクトによっては、すべてのパラメーターが必要とは限りません。たとえば、パートナーを識別するには、CP 名よりも XID が使用されます。また別の環境では、たとえば、PU 内に LU が 1 つだけある場合、CP はその LU と同一の名前をもつと考えられます。この最初の基本構成よりも先に進む準備ができたときに、プロダクト固有の資料を調べて、上記の点についての詳細な説明をお読みください。

## シナリオ

この解説用に選択したシナリオでは、相互に接続された System/390 CICS システムの SNA ネットワークが確立されていて、そこに、CICS for AIX システムを加えることを想定しています。この章は、メインフレーム・ホストにワークステーションを接続する AIX のシステム管理担当者の見方から書かれています。

使用される名前の重複という潜在的な問題を避けるため、またネットワーク管理を支援するために、VTAM システム管理担当者がネットワーク・リソース名の調整を行います。そのため、ユーザーは VTAM システム管理担当者が指定したワークステーションのネットワーク名に従わなければなりません。

---

## 構成についての詳細情報

次のセクションの表は、CICS プロダクトと SNA プロダクト内でリソース名が構成される場所を示しています。通常、情報は、リソース名をパラメーターとして指定する属性に名前を付けることによって提供され、その属性が現れる定義ステートメントまたはフロントエンド・パネルを与えることによって提供されます。各種のプ

ログダクトが使用する構成ツールについての詳細は、76 ページの『SNA 資料』にリストされたプロダクト固有の資料を参照してください。

表に示された名前が、CICS プロダクトと SNA プロダクトの両方に指定されている場合は、両方のプロダクトで同一の名前を使用する必要があります。どのプロダクトにも定義されていない名前は、ダッシュ (-) で示されています。

メインフレーム・ホストとワークステーションは、構成の詳細部分において完全に対称ではありません。たとえば、ワークステーションの構成では、ワークステーションがメインフレーム・ホストのアドレスを認知していることが必要ですが、メインフレームでは、ワークステーションのアドレスは与えられません。これは、ワークステーションは呼び出し システムとして、メインフレームは *listener* として理解されているためです。つまり、セッションは、ワークステーションがメインフレームを呼び出すことによって開始されます。そのため、ワークステーションが必要とするメインフレームについての情報は、メインフレームが必要とするワークステーションについての情報とは異なります。

## メインフレーム・ホストの構成

System/390 CICS と VTAM は、すでにそのネットワーク名と LU 名をネットワークに対し定義しています。表 6 は、System/390 CICS および VTAM の中で定義されるこれらの名前が定義される場所を示しています。これをメインフレームのローカル・リソースと呼びます。これらの名前は、この後ワークステーションを構成するときを使用し、そこではこれらの名前をワークステーションのパートナー・リソースと呼びます。

ワークステーションでは、メインフレーム・ホストの名前を認知していなければなりません。この名前は 12 文字の 16 進コードで、フロントエンド・プロセッサに割り当てられています。このコードは、System/390 CICS または VTAM の構成では使用されないため、この表には記載されていません。ユーザーは、ネットワーク管理担当者からアドレスを聞くことができます。

表 6. System/390 CICS および VTAM へのローカル・リソースの定義

	System/390 CICS	VTAM
ネットワーク名	-	NETID= VTAM 始動プロシージャ内の属性
LU 名	APPLID= SIT 内の属性	CICS を VTAM に定義する APPL ステートメント上のラベル

## System/390 CICS および VTAM へのワークステーションの定義

ユーザーは、ワークステーションに使用するリソース名については、CICS および VTAM システム管理担当者に従わなければなりません。ワークステーションからのセッション開始要求を System/390 CICS が認識して受信できるように、System/390 CICS と VTAM にワークステーションを定義していなければなりません。システム管理担当者はそれぞれのワークステーション用に、CICS 内に新しい CONNECTION 定義を作成し、VTAM 内には新しい PU、LU および MODEENT 定義ステートメントを作成します。

表7 は、LU 名、PU 名、CP 名および XID 名を定義する場所を示しています。これらの名前は、メインフレームのパートナー・リソースの名前であることを覚えておいてください。ワークステーションを構成するとき、これらの名前がワークステーションのローカル・リソースを参照するところでこれと同一の名前を使用します(表8 を参照してください)。

表7. CICS Transaction Server および VTAM へのワークステーションの定義

	System/390 CICS	VTAM
LU 名	CONNECTION 定義内の NETNAME 属性	LU ステートメント上のラベル
PU 名	-	PU ステートメント上のラベル
CP 名	-	CPNAME= PU ステートメント内の属性
XID	-	IDBLK= と IDNUM= PU ステートメント内の属性
モード	SESSIONS 定義内の MODENAME 属性	LOGMODE= MODEENT ステートメント内の属性

注: ワークステーションはメインフレームと同じネットワーク内に存在するため、24 ページの表6 の VTAM に定義されたものと同じネットワーク名を使用します。

## AIX マシンの構成

AIX マシンを構成するには、以下のことを行う必要があります。

1. マシンをネットワークに定義する。
2. パートナーへの接続を定義する。

AIX は、広範囲に対応するプラットフォームで、CICS をネットワークに接続する方法を多数提供します。AIX SNA プロダクトは、CICS for AIX プロダクトとは異なるマシン内におくことができます。このプロダクトでは Encina PPC ゲートウェイを使用し、SNA ネットワークにアクセスする前に CICS が TCP/IP を使用してこのゲートウェイへの通信を行うことができます。マシンは異なるネットワークにあってもかまいません。ここに示す例では、CICS for AIX と AIX SNA の両方が稼働し、メインフレームと同じネットワーク内にある、単一のマシンの基本的な事例を説明しています。

### ネットワークへの AIX マシンの定義

表8 は、CICS for AIX と AIX SNA に対して、ローカル・リソースが定義される場所を示しています。名前は、表7 に示された、以前にユーザーがメインフレームのシステム管理担当者の指定に同意したものを使用します。

表8. CICS for AIX および AIX SNA におけるローカル・リソースの定義

	CICS for AIX	AIX SNA
ネットワーク名	LocalNetworkName= 領域定義内の属性	「初期ノード・セットアップ (Initial Node Setup)」パネル内のローカル・ネットワーク名の属性

表 8. CICS for AIX および AIX SNA におけるローカル・リソースの定義 (続き)

	CICS for AIX	AIX SNA
LU 名	LocalLUName= 領域定義内の属性	「LU 6.2 ローカル LU プロファイルの追加 (Add LU 6.2 Local LU Profile)」パネルと「サイド情報プロファイル (Side Information Profile)」パネル内のローカル LU 名の属性
CP 名	-	「初期ノード・セットアップ (Initial Node Setup)」パネル内の制御点名の属性
XID	-	「初期ノード・セットアップ (Initial Node Setup)」パネル内の XID ノード ID の属性
モード	DefaultSNAModeName= 通信定義内の属性	「LU 6.2 モード・プロファイルの追加 (Add LU 6.2 Mode Profile)」パネル内のプロファイル名の属性

### System/390 CICS への接続の定義

表 9 は、メインフレームへの接続の定義方法を示しています。指定しなければならないのは、LU 名 (実際には System/390 CICS システムの名前) とハードウェア・アドレスです。

表 9. System/390 CICS への接続の定義

	CICS for AIX	AIX SNA
LU 名	通信定義内の RemoteLUName 属性	「LU 6.2 パートナー LU プロファイルの追加 (Add LU 6.2 Partner LU Profile)」パネル内の完全修飾パートナー LU 名の属性
アドレス	-	「初期ノード・セットアップ (Initial Node Setup)」パネル内のリンク・ステーション情報のリンク・アドレスの属性

## CICS for SNA の構成 - 次のステップ

この章では、SNA 通信用にワークステーションを構成するために必要な作業について説明しました。ここでは論理接続についての情報しか記載されていないので、ユーザーが使用する特定の物理接続についての情報 (たとえば、メインフレームに対してトークンリングまたはイーサネット LAN、または SDLC 接続のいずれを使用するかなど) が必要になります。

さまざまな接続タイプの構成例は、*Open System Intercommunication Guide* にすべて記載しています。

この紹介で述べたように、最初の目標は、基本的なセットアップを構成することと、簡単なアプリケーションを使用してそれをテストすることです。たとえば次のようなものを使用します。



- APING は TCP/IP PING プログラムと同等の APPC プログラムです。APING は、独立型の APPC アプリケーションとして CICS から独立して使用できるので、SNA プロダクトの構成をテストすることができます。
- CICS Server ISC PING Transaction は、次の Web サイトで、SupportPac™ CC02 として使用することができます。

[www.ibm.com/software/ts/cics/txppacs/cc02.html](http://www.ibm.com/software/ts/cics/txppacs/cc02.html)

ここで作動可能 APPC リンクがあることを確認した後でのみ、構成を拡張して、ユーザー独自のアプリケーションを組み込んでください。



---

## 第 2 部 CICS 相互通信機能

第 2 部では CICS 相互通信機能について説明します。この情報は、計画担当者およびアナリスト向けのものであり、プログラマーには入門用の資料となるものです。

第 2 部は次の章で構成されています。

- 31 ページの『第 6 章 CICS 相互通信機能の紹介』
- 35 ページの『第 7 章 機能シップ』
- 43 ページの『第 8 章 トランザクション・ルーティング』
- 51 ページの『第 9 章 分散プログラム・リンク』
- 55 ページの『第 10 章 非同期処理』
- 63 ページの『第 11 章 分散トランザクション・プログラミング』



## 第 6 章 CICS 相互通信機能の紹介

本章では、CICS 相互通信機能について簡単に説明します。

『CICS 相互通信機能の要約』に記載した図と説明は、技術的な詳細について熟考の上解説しています。これらの図と説明は、アプリケーション・プログラマー側から見たときの機能を表しています。重要な点は、機能シップ、非同期処理、トランザクション・ルーティング、および分散プログラム・リンク (DPL) は強力な分散機能ですが、アプリケーション・プログラマーはそれらの機能の存在を知らなくてもよいということです。アプリケーションは、システム・プログラマーによって作成された定義の全面的な制御下で、ローカルまたはリモートのリソースやプログラムにアクセスすることができます。

分散トランザクション・プログラミングは、より複雑な機能です。アプリケーション・プログラマーは、ダイアログがリモート・トランザクションによって引き起こされることを知っておく必要があります。プログラム論理は、そのダイアログの現行状態に反応するものでなければなりません。

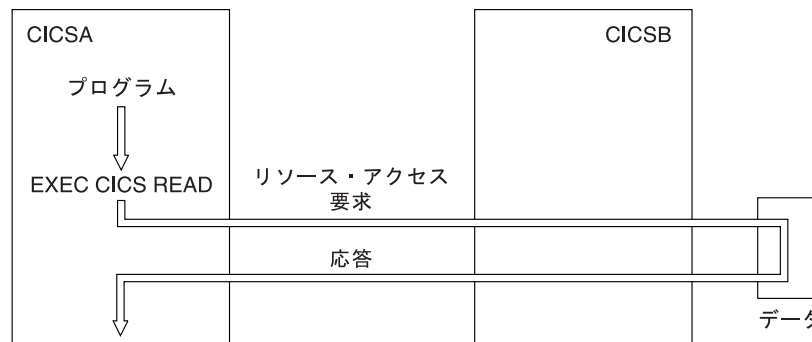
33 ページの『相互通信機能の選択』は、特定の要件について正しい相互通信機能を選択するための指針を示します。

### CICS 相互通信機能の要約

このセクションでは、新しい CICS ユーザーのために、CICS 相互通信機能のごく簡単な定義を紹介します。

#### 機能シップ

システム CICS A にあるプログラムは、リモート・システム CICS B が所有しているリソース (ファイルや一時データ・キューなど) について、あたかもローカルで所有しているかのようにアクセスします。次の図はデータ・アクセス要求を示しています。



非同期処理は、データにアクセスしない機能シップの例です。伝送された要求は、リモート・トランザクション用の EXEC CICS START コマンドです。

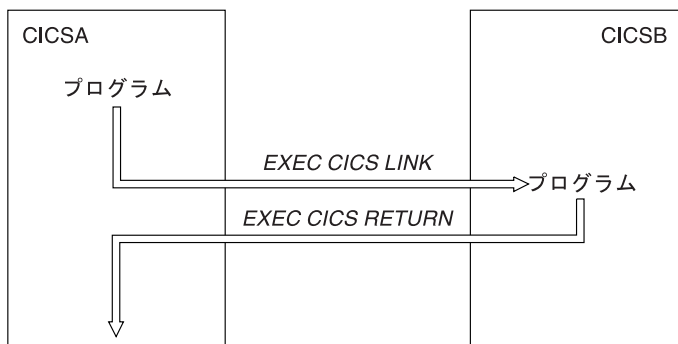
## トランザクション・ルーティング

システム CICS A に接続されている端末は、ローカル・トランザクションであるかのようにリモート・システム CICS B 内のトランザクションを実行します。



## 分散プログラム・リンク

システム CICS A にあるプログラムは、ローカル・システムで稼働されているかのように、リモート・システム CICS B にあるプログラムをリンクします。



## 分散トランザクション・プログラミング

システム CICS A のプログラムとシステム CICS B のプログラムの 2 つのプログラムは、相互に同期通信します。このダイアログのことを会話といいます。



---

## 相互通信機能の選択

下記のリストは、代表的な要件とそれに合った機能を示しています。これを参照して、的確な相互通信機能を選択してください。

- **端末ユーザーが、接続しているシステムとは異なるシステムが所有しているトランザクションを実行させたい場合**  
トランザクション・ルーティングを使用してください。
- **トランザクションが、別のシステムの所有しているデータを読み書きしたい場合**  
機能シップを使用します。ただし、機能シップはトランザクション・ルーティングよりもオーバーヘッドが高いため、リモート・システム内のデータと同様にローカル・システム内のデータにトランザクションがアクセスしなければ、トランザクション・ルーティングを使用する方がよいでしょう。
- **トランザクションが、別のシステムがアクセスしている IMS™ データまたは DLI データを読み書きしたい場合**  
分散プログラム・リンクを使用します。
- **トランザクションが、指定のトランザクションを開始しなければならないことをリモート・システムに知らせる必要がある場合**  
非同期処理を使用してください。
- **トランザクションが、アクセスとアクセスの間になんらかの処理をしながら、リモート・データに何回かアクセスしたい場合**  
可能であれば分散プログラム・リンクを使用してください。この要件は、いくつかの機能シップ要求を使用すると満たすことができますが、DPL を使用すれば、ネットワークにおけるデータ・フローが最少に抑えられます。DPL が要求全体を満たすことができない場合は、DPL と機能シップを併用すると、機能シップだけを使用するよりも、ネットワークの使用効率がよくなります。
- **アプリケーションが、2 つ以上のシステムで通信し、データ交換にもとづいて相互に依存したリソース更新を行う必要がある場合**  
分散トランザクション・プログラミングを使用してください。





---

## 第 7 章 機能シップ

この章では、以下のトピックを記載します。

- 『機能シップの紹介』
- 36 ページの『アプリケーションの透過性』
- 36 ページの『アクセスできるリモート・リソース』
- 38 ページの『機能シップの動作の仕方』
- 41 ページの『同期』
- 41 ページの『機能シップの例』

---

### 機能シップの紹介

CICS 機能シップを使用すると、CICS アプリケーション・プログラムは、EXEC CICS コマンドを使用して次のような処理を行えるようになります。

- ファイル制御要求をシップすることによって、その他の CICS システムが所有している CICS ファイルにアクセスする。
- 一時データおよび一時記憶機能についての要求をシップすることによって、その他の CICS システム内にある一時データや一時記憶キューに (またはそこから) データを転送する。
- START 要求をシップすることによって、その他の CICS システム、または IMS などの SNA LU タイプ 6 プロトコルをインプリメントしているその他の非 CICS システムにある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、55 ページの『第 10 章 非同期処理』に説明があります。

伝送されたファイル制御要求を、36 ページの図 2 に示します。この図では、CICS システム CICA で稼働しているトランザクションは、NAMES というファイルにファイル制御の READ コマンドを出します。CICA は、リソース定義テーブルから、CICB というリモート CICS システムがこのファイルを所有していることを知ります。CICA は、READ 要求を適切なシップ形式に変更してから、実行のためにそれを CICB に伝送します。

CICB では、その要求はミラー・トランザクションと呼ばれる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 にそれを出し、必要なデータを CICA に戻します。

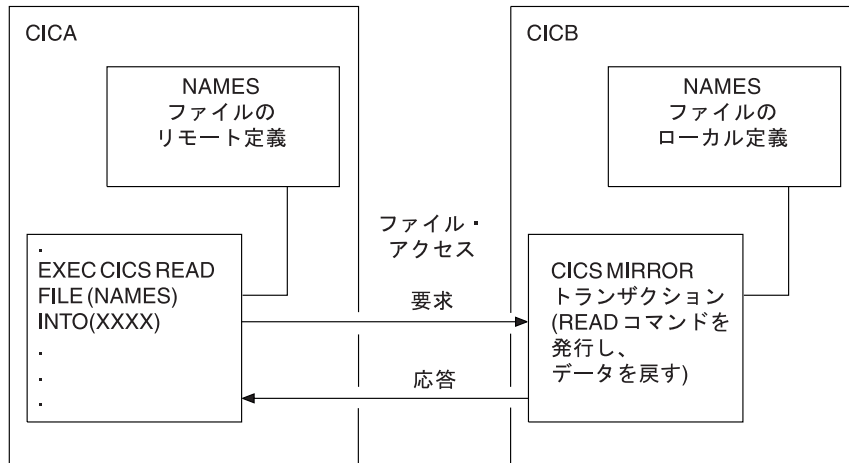


図2. 機能シップ

## アプリケーションの透過性

機能シップを使用するアプリケーションは、要求されたリソースのある場所を知らなくてもかまいません。このアプリケーションは、それ自体が実行されているシステムがすべてのリソースを所有しているかのように、ファイル制御コマンド、一時記憶コマンドなどを使用します。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）システムにあることを指定することができます。

リモート・リソースの定義には、リモート・システムがリソースを認識するときの名前と、ローカルの範囲内で認識される別の名前との両方を指定できます。リソースをローカル名で要求すると、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。この機能が役立つのは、2 つ以上のシステムに同じ名前のリソースが存在し、それぞれが配置されているシステムに特有のデータを含んでいる場合です。

アプリケーション・プログラムは、さまざまな EXEC CICS コマンドで、SYSID オプションを使用してリモート・システムを明示的に指定することができます。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て、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。このように SYSID を使用すると、プログラムとリソースのある場所との独立性はくずれません。これを使用する際の利点は、ローカル・システムを含む**すべての**システムを SYSID オプションに指定できることです。ローカル・リソースとリモート・リソースのどちらにアクセスするのかは、実行時に決定することができます。

## アクセスできるリモート・リソース

機能シップ要求は、下記のリモート・リソースにアクセスすることができます。

- CICS ファイル制御データ・セット
- 一時記憶
- 一時データ
- IMS および DOS DLI データベース (System/390 CICS システムから)

## CICS ファイル制御データ・セット

機能シップを用いると、リモート CICS 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。次の EXEC CICS コマンドはサポートされていません。

- INQUIRE FILE
- INQUIRE DSNAME
- SET FILE
- SET DSNAME

読み取り専用要求と更新要求は行うことができます。データの保護は、さまざまな CICS プロダクトで使用できるセキュリティー機能によって異なります。リモート・リカバリー可能ファイルの更新は、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 2 つ以上の接続された CICS システムにあっても、ローカル・ファイルとリモート・ファイルの論理的に関連した更新は、同じ作業論理単位内で行うことができます。

物理レコード ID の値 (たとえば、BDAM、VSAM RBA、およびレコードにキーが組み込まれていないファイル) が入ったリモート・ファイル要求を使用するシステムを設計するときは、注意が必要です。リモート・システムにあるアプリケーション・プログラムは、上記のファイルを更新または再編成した後で正しい値にアクセスできなければなりません。

## IMS データベース

機能シップを使用すると、System/390 CICS トランザクションは、リモート CICS OS/390 システムに関連した IMS/ESA<sup>®</sup> DM および IMS/VS DB データベース、あるいはリモート CICS/VSE システムに関連した DL/I for VSE/ESA データベースにアクセスすることができます。

CICS Transaction Server for Windows、CICS/400、およびオープン・システム CICS システムは、機能シップを使用しても IMS または DL/I のデータベースにはアクセスできませんが、分散プログラム・リンクを使用するとアクセスできます (51 ページの『第 9 章 分散プログラム・リンク』を参照してください)。以下の説明は、System/390 CICS システムにのみ当てはまります。

リモート CICS 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・システムが所有しているローカル・データベースであっても、IMS データベース制御 (DBCTL) を使用してアクセスされるデータベースであってもかまいません。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、PSB が異なるリモート・システムにあると定義されていても、1 つの作業論理単位に 2 つ以上の PSB をスケジュールすることはできません。そのため異なるシステムにおいて、1 つの作業論理単位内で DL/I の論理的に関連する更新を実行できません。

PSB ディレクトリー・リスト (PDIR または DLZACT) は、PSB がリモート・システムにあることを定義するときに使用します。リモート・システムは、データベースおよび関連 PCB 定義を所有します。CICS システムが DL/I アクセス要求を

別のプロセッサ・システムに出し、ローカル要求を出さないときは、要求側のシステムに IMS をインストールする必要はありません。

## 一時記憶および一時データ

多くの場合、1 つの CICS システムで一時データと一時記憶キューを使用してきましたが、それらを相互接続システム環境に拡張することができます。たとえば、リモート・システムで夜間処理を行うためのレコードのキューを作成することができます。キューは、その他の要求のために端末を解放している間に、リモート・システムからの応答を処理するための手段にもなります。応答は、準備ができた時点で端末に戻され、トランザクションを入力しているときに一時休止があると、オペレーターに送信されます。

### 一時記憶

アプリケーション・プログラムは、機能シップを使用して、リモート・システムにある一時記憶キューにデータを送信したり、そのキューからデータを検索したりすることができます。リカバリー可能キューは、ローカル (リソースを所有している) システムでリカバリー可能と定義しなければなりません。

### 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一時データ・キューまたは区画外一時データ・キューにアクセスすることができます。要求側のシステムにあるリソース定義には、指定のキューがリモート・システムにあることを定義します。リモート・システムにあるキュー定義には、そのキューがリカバリー可能かどうか、また、トリガー・レベルや関連端末があるかどうかを指定します。

一時データ宛先にトランザクションが関連付けられている場合、指定したトランザクションは、キューを所有しているシステムで実行するように定義されていなければなりません。これは、リモートとして定義することはできません。端末がそのトランザクションに関連付けられている場合、その端末は、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。

プログラムは、リモート命名機能を使用して、ローカル・システムとリモート・システムの両方にある CSMT などの CICS サービス宛先にデータを送信することができます。

---

## 機能シップの動作の仕方

以下の 2 つの CICS コンポーネントによって機能シップがインプリメントされます。

- CICS 変換プログラム
- CICS ミラー・トランザクション

2 つのコンポーネントがどのように一緒に動作するのかを、39 ページの図 3 に示します。

## 変換プログラム

CICS トランザクションがリソースにアクセスするための要求を出すと、コマンド・レベルの EXEC インターフェース・プログラムは、ローカル CICS システムがリソースを所有しているかどうかを判別します。リソースが別のシステムにあると、制御権は機能シッポ変換プログラムに渡されます。

DL/I (EXEC DL/I または CALL DLI) 要求では、変換プログラムの機能の一部も提供する DL/I インターフェースを使用します。

変換プログラムは、伝送に適した形式に要求を変換し、相互通信コンポーネントを呼び出して、リソース所有システムに要求を送信します。

CICS 相互通信コンポーネントは、要求をリモート・システムに送信します。トランザクションの代わりに特定のリモート・システムへの最初の要求で、ローカル・システムにある通信コンポーネントは、リモート・システムにあるミラー・トランザクションのタスク生成用に、形式設定された要求の前にミラー・トランザクション ID を付けます。ローカル変換プログラムは、リモート・ミラー・トランザクションが終了したかどうかを記録しておき、必要なときにそれを再度呼び出します。

リモート・システムからの応答を受け取ると、別の変換プログラムがそれをデコードします。CICS は、デコードされた応答を使用して元のコマンド・レベルの要求を完了します。

リモート・システムは、要求を処理する際に自分の変換プログラムを使用します(次のセクションを参照してください)。

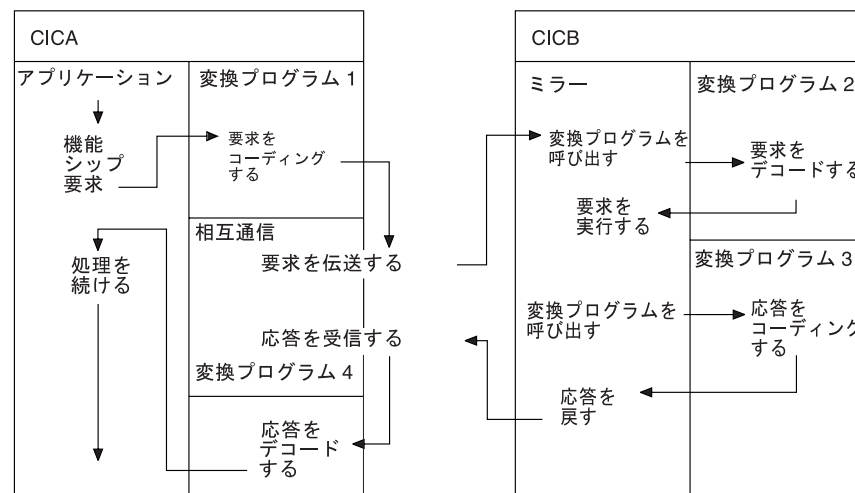


図 3. ミラー・トランザクションと変換プログラムの関係

## ミラー・トランザクション

リソース所有システムは、着信した機能シッポ要求をミラー・トランザクションに渡します。特定のリモート・トランザクションからの最初の要求によって、ミラー・トランザクションの新しいインスタンスが開始されます。ミラー・トランザクションは、CICS 相互通信機能を使用して要求側のシステムと通信します。

ミラー・トランザクションは、CICS 変換プログラムを使用して、形式設定された要求をデコードし、コマンドを実行します。このコマンドが完了した時点で、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、これを要求側のシステムへ戻します。

次のような場合、ミラー・トランザクションは、現行コマンドへの応答を送信後も活動状態のままです。

- 以降のコマンドが実行されるかどうか、現行のコマンドの実行時に設定されたシステム特定情報の保存内容によって決まる場合。たとえば、次のようになります。

REWRITE は、前の READ UPDATE によって、実行されるかどうかが決まります。

READNEXT は、前の STARTBR によって、実行されるかどうかが決まります。

- 以降のコマンドが実行されるかどうか、現行のコマンドの実行時に設定されたアプリケーション特定情報の保存内容によって決まる場合。たとえば、リカバリー可能リソースが更新された場合。
- 上記の理由の 1 つにより、ミラー・トランザクションは、前のコマンドに対して応答した後も活動状態のままです (その後、ミラー・トランザクションは、要求側のシステム内の作業論理単位が終了するまで活動状態のままです)。

その他のすべての場合、ミラー・トランザクションは、現行のコマンドに応答すると、終了します。

活動状態のミラー・トランザクションは、要求側のトランザクションが同期要求を出すか、または正常終了すると、必ず終了します。ミラー・トランザクションは、SYNCONRETURN オプションを指定した LINK コマンドを実行したあとも、必ず終了します。SYNCONRETURN の詳細については 52 ページの『同期』を参照してください。

## 多重ミラー

トランザクションは、任意の順序でリカバリー可能リソースやリカバリー不能リソースにアクセスすることができます。このトランザクションは、リカバリー可能リソースのある場所によって影響を受けません (たとえば、すべてのリソースが異なるリモート・システムにあってもかまいません)。ローカル・トランザクションが複数のリモート・システムのリソースにアクセスすると、相互通信コンポーネントは、各リモート・システム内のミラー・トランザクションを呼び出して、ローカル・トランザクションごとに要求を実行します。各ミラー・トランザクションの終了は、上記の規則に従います。トランザクション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。

## チェーン・ミラー

ミラー・トランザクションは、EXEC CICS 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 DLI 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ます。要求されたリソースは、適切なリソース・テーブルに置かれます。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、シッフ用に形式設定され、そのミラー・トランザクションを活動化する、指定のリモート・システムに送信されます。このことをチェーン・ミラーといいます。

## 同期

CICS リカバリーおよび再始動機能は、リモート・システムとローカル・システムのリカバリー可能リソースの変更に整合性を保つために、要求側のトランザクションが同期点に到達した時点で、リカバリー可能リソースを更新しているすべてのミラー・トランザクションも同期点を取るようにするものです。CICS マスター端末 (あるいは、CICS/400 の場合は制御領域) は、この処理で発生した障害の通知を受け取ります。これにより、適切な修正処置がとられます。この処置は、手動で、またはユーザー作成のコードによって実行することができます。

トランザクションが同期点要求を出すか、正常終了すると、相互通信コンポーネントは、同期点要求を出して終了するように、ミラー・トランザクションにメッセージを送信します。ミラー・トランザクションの出す正常な同期点は、要求側のシステムに送信される応答内に示されます。要求側のシステムは、次にすべてのリカバリー可能リソースの変更をコミットして、同期点処理を完了します。

## 機能シッフの例

図 4 および 42 ページの図 5 に、ミラー・トランザクションの存続時間の例を示します。

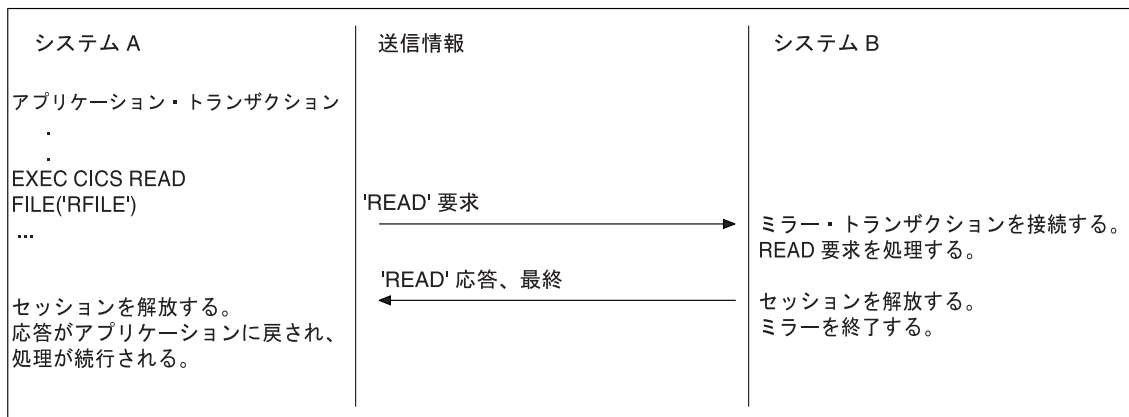


図 4. 機能シッフ - 単純な照会：ここではリソースは変更しません。セッションは解放されて、ミラー・タスクは即座に終了します。

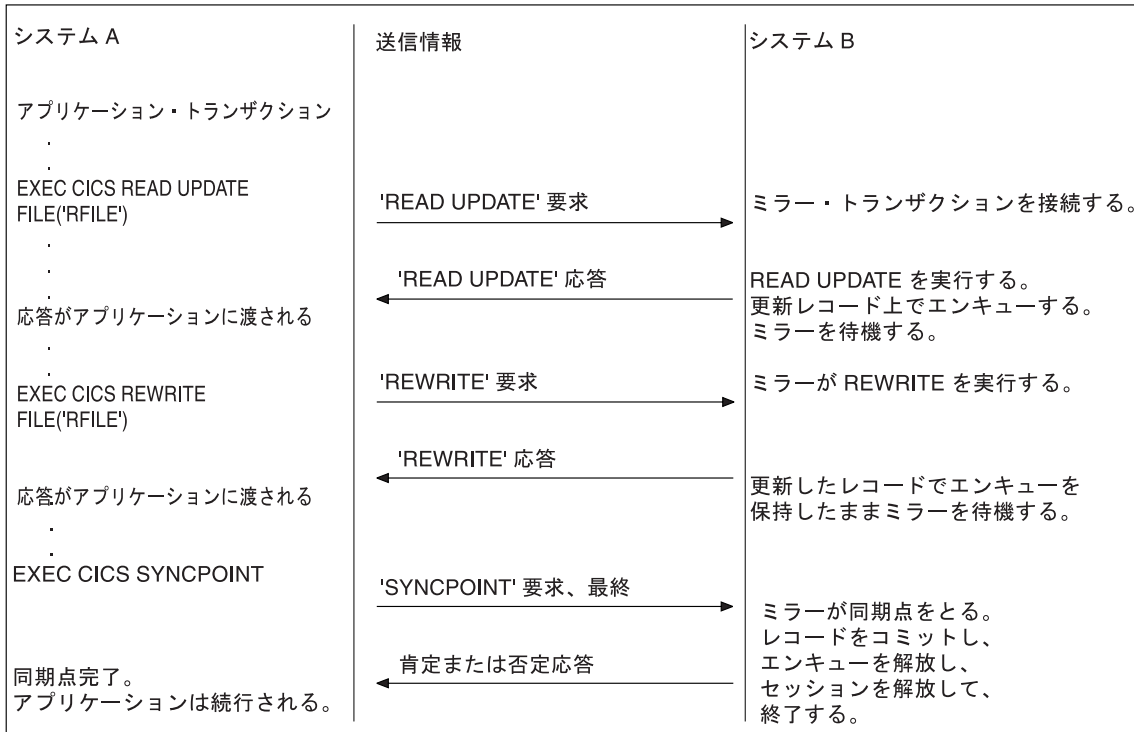


図 5. 機能シップ - 更新：ミラー・トランザクションが *REWRITE* を待機するので、*SYNCPOINT* を受け取るまで終了しません。ファイルがリカバリー可能でない限り、更新されたレコードのエンキューは、*REWRITE* コマンドの外では保持されないことに注意してください。

図 5 では、ミラーが長時間実行になります。



## 第 8 章 トランザクション・ルーティング

この章では、以下のトピックを記載します。

- 『トランザクション・ルーティングの紹介』
- 44 ページの『トランザクション・ルーティングの開始』
- 48 ページの『中継プログラム』
- 49 ページの『基本マッピング・サポート』
- 49 ページの『ルーティング・トランザクション (CRTE)』

### トランザクション・ルーティングの紹介

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が、接続されている別の CICS システムにあるトランザクションを実行させることができます。つまり、CICS システムの周りに端末とトランザクションを分散することができ、任意の端末で任意のトランザクションを稼働することができます。

図 6 は、ある CICS システムに接続された端末が、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のを示しています。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。

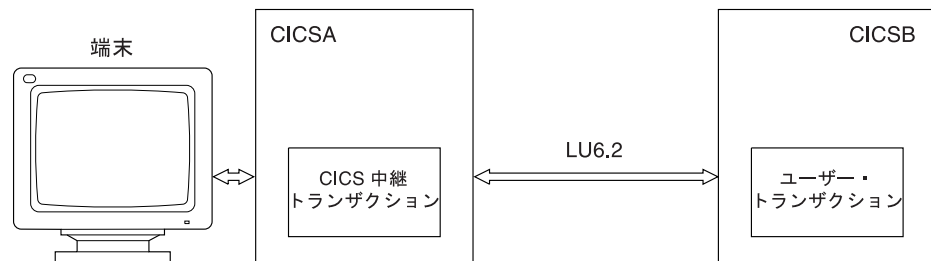


図 6. トランザクション・ルーティングのエレメント

2 つの異なる CICS プロダクトは、LU6.2 リンクによって接続されなければなりません。LU6.1 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 装置、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。System/390 CICS によってサポートされる**すべての**端末とセッション・タイプは、トランザクション・ルーティングに適します。ただし、次にリストした端末とセッション・タイプは**除きます**。

- LUTYPE6.1 の接続およびセッション
- 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

CICS Transaction Server for Windows、CICS/400、およびオープン・システム CICS は、最低限の BMS をサポートします。(SEND TEXT はサポートされています。) System/390 CICS システムだけが、バッチ・データ交換、標準 BMS、および全機能

BMS をサポートします。これらのプロダクトの機能によって異なりますが、ユーザー・トランザクションは、端末またはセッション・タイプに合った、CICS 端末制御、BMS、またはバッチ・データ交換を使用して端末と通信します。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システム (図 6 の CICSB) で行われます。BMS ページング操作は、端末所有システム (図 6 の CICS A) で行われます。

疑似会話はサポートされます。ただし、端末が APPC セッションである場合を除きます。疑似会話を構成するさまざまなトランザクションは、異なるシステムに置くことができます。

---

## トランザクション・ルーティングの開始

トランザクション・ルーティングは、下記の 4 通りの方法で開始することができます。

1. 端末が、CICS 端末所有システムへトランザクションを開始するための要求を送る。トランザクションのインストール済みリソース定義に基づいて (おそらく、System/390 CICS および CICS Transaction Server for Windows では、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)、要求は適切なトランザクション所有システムにルーティングされ、トランザクションは端末がトランザクション所有システムに接続されているかのように稼働します。
2. トランザクションが、自動トランザクション開始 (ATI) によって開始し、別の CICS システムが所有している端末を獲得する。
3. System/390 CICS トランザクションが、別の CICS システムが所有している APPC 端末または接続とのセッションに ALLOCATE コマンドを出す。
4. CICS 提供のトランザクション CRTE が、他のシステムでトランザクションを呼び出すために使用される。49 ページの『ルーティング・トランザクション (CRTE)』を参照してください。

## 端末開始のトランザクション・ルーティング

端末でトランザクションの開始を要求すると、CICS は、インストールされているトランザクション定義を調べます。トランザクションに REMOTESYSTEM オプション (またはオープン・システム CICS では REMOTESYSID) 中に指定したリモート・システムの定義があると、CICS は、中継プログラムに制御権を渡してトランザクション・ルーティングを駆動する中継トランザクションを開始します。

### 静的トランザクション・ルーティング

静的トランザクション・ルーティングという用語は、標準トランザクション・ルーティングと動的トランザクション・ルーティングを区別するために使用します。CICS/400 では、トランザクション・ルーティングはすべて静的です。System/390 CICS とオープン・システム CICS において静的ルーティングが起るのは、トランザクション定義に DYNAMIC(NO) を指定している場合です。

静的トランザクション・ルーティングの場合、要求は REMOTESYSTEM オプションで指定したシステムにルーティングされます。REMOTESYSTEM を指定してい

ない場合、またはローカル CICS システムを指定している場合、トランザクションはローカル・トランザクションになり、トランザクション・ルーティングは不要です。

## 動的トランザクション・ルーティング

動的トランザクション・ルーティングは、トランザクション・ルーティング要求の宛先であるシステムをユーザー作成プログラムによって選択するものです。動的トランザクション・ルーティングは、System/390 CICS、オープン・システム CICS、および CICS Transaction Server for Windows によって提供されます。インプリメンテーションは異なります (次の説明を参照してください)。CICS/400 は、動的トランザクション・ルーティングをサポートしていません。

**System/390 CICS および オープン・システム CICS:** System/390 CICS またはオープン・システム CICS システムでは、トランザクションの定義時に DYNAMIC オプションを使用することができます。これには、CICS Transaction Server for Windows または CICS/400 トランザクションのリモート定義が組み込まれています。

DYNAMIC(YES) を指定するということは、定義し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可能性があるということを意味します。CICS でこれを行う際には、**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提供のプログラムを使って端末入力データを調べて、選択した任意のトランザクションやシステムにそれを再度宛先指定できるようにします。System/390 CICS では、このプログラムにはデフォルト名 DFHDYP がありますが、DTRPGM システム初期設定パラメーターを使用して代替名を定義することもできます。

パラメーターは、CICS とルーティング・プログラム間の連絡域で渡されます。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。たとえば、System/390 CICS では、パラメーターには次のようなものが入ります。

- 呼び出しの理由。ルーティングの試みが失敗したとき、またはターゲット・トランザクションが終了したときに、ルーティング・プログラムを再度呼び出すことができます。
- エラー情報。
- ターゲット・システムの SYSID。最初は、インストールされたトランザクションについて REMOTESYSTEM (またはオープン・システム CICS では REMOTESYSID) オプションに指定したものの。
- ターゲット・トランザクションの名前。最初は、インストールされたトランザクションについて REMOTENAME オプションに指定した名前。
- 端末から最初に入力されたデータが入ったデータ域を指すポインター。

動的トランザクション・ルーティング・プログラムは、ほとんどの EXEC CICS コマンドを出すことができます。

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ロードなどの要因にもとづいてトランザクション・ルーティングの判断をすることができます。

**CICS Transaction Server for Windows:** CICS Transaction Server for Windows にはユーザー出口が用意されており、その中でユーザー・プログラムはトランザクション・ルーティング要求のターゲット・システムを変更することができます。この出口では、シッパされたどの機能要求のターゲット・システムも変更することができます。さらに、ローカル・システムに定義されていないトランザクション用のターゲット・システムを提供することもできます。

出口は、定義されていれば、以下のものが処理される前に呼び出されます。

- 未定義のトランザクション・コード
- トランザクション・ルーティング要求
- 機能シッパ要求
- DPL 要求
- SYSID オプションを指定した任意のコマンド

### 端末定義の伝送

伝送可能として定義された端末の場合、トランザクション所有システムに定義は不要です。トランザクション・ルーティングをサポートする必要がある場合、端末所有システムは、トランザクション所有システムに端末定義を送信します。

## 自動トランザクション開始

自動トランザクション開始 (ATI) は、CICS システムまたはシステム・ネットワークにおいて内部的に作成されたトランザクション要求によってトランザクションがスケジュールされるための処理です。

ATI 要求によって、リモート・システム内のトランザクションを開始することができます (47 ページの図 7 を参照してください)。

さらに、ATI を使用して、特定の CICS システムが所有しているトランザクションについての要求で、別に接続されているシステムが所有している端末を指定することができます (図 7 を参照してください)。元の ATI 要求はアプリケーション所有システムで出されますが、この要求は、端末所有システムに CICS が送信します。このシステムでは、指定された端末と一緒に中継トランザクションが開始されま

す。

次に、アプリケーション所有システムのユーザー・トランザクションは、端末開始のトランザクション・ルーティング用に 44 ページで説明したように開始されます。ただし、重要な違いが 1 つあります。この場合の追加要因として、ATI 要求に関連した AID (自動開始記述子) があります。AID は、リモート・トランザクションおよびシステムの名前を指定します (図 7 の **P1** および **S1**)。図 7 では、端末所有システム (**S2**) は、**REMOTENAME (P1)** および **REMOTESYSTEM (S1)** (またはオープン・システム CICS では **REMOTESYSID**) を指定するトランザクション定義を検出しなければなりません。

System/390 CICS システムでは、**DYNAMIC(YES)** が **REMOTESYSTEM** の代わりになります。CICS/VSE および CICS Transaction Server for VSE/ESA では、**DYNAMIC(YES)** がコーディングされていると、動的ルーティング・プログラムは駆動されません。AID からリモート・システム名が取られます。CICS Transaction Server for OS/390 および CICS Transaction Server for z/OS では、**DYNAMIC(YES)** がコーディングされていると、動的ルーティング・プログラムは駆動されますが、

リモート・システム名は依然として AID から取られます。ルーティング・プログラムはその他のことも行えます。たとえば、さまざまなりモート・システムにルーティングされた要求数を更新したりできます。

つまり、**S2** が、**REMOTENAME (P1)** および **REMOTESYSTEM (S1)** か **DYNAMIC(YES)** を指定したトランザクション定義を見つけると、**START** コマンドは **S1** にルーティングされます。これが見つからないと、**START** コマンドは拒否されます。

ATI 要求はキューに入れられます。

- 端末所有システムとのリンクをアプリケーション所有システムで使用できない場合
- 端末所有システムで端末を使用できない場合

アプリケーション所有システムを考慮する限り、ATI の「単一システム」ビューを作成すると全体的に効果があります。端末がリモートであっても、ATI が作動するための方法に影響はありません。

トランザクション所有システムでは、ATI についての通常の規則が適用されます。トリガー・レベルに達するか、またはインターバル制御の **START** 要求が満了した時点で、一時データ・キューからトランザクションを開始することができます。特に、一時データの開始の場合、一時データ・キューは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。

図7 は、3 つのシステムが関与する ATI の例を示します。

- S0** 開始するシステム
- S1** トランザクション所有システム
- S2** 端末所有システム

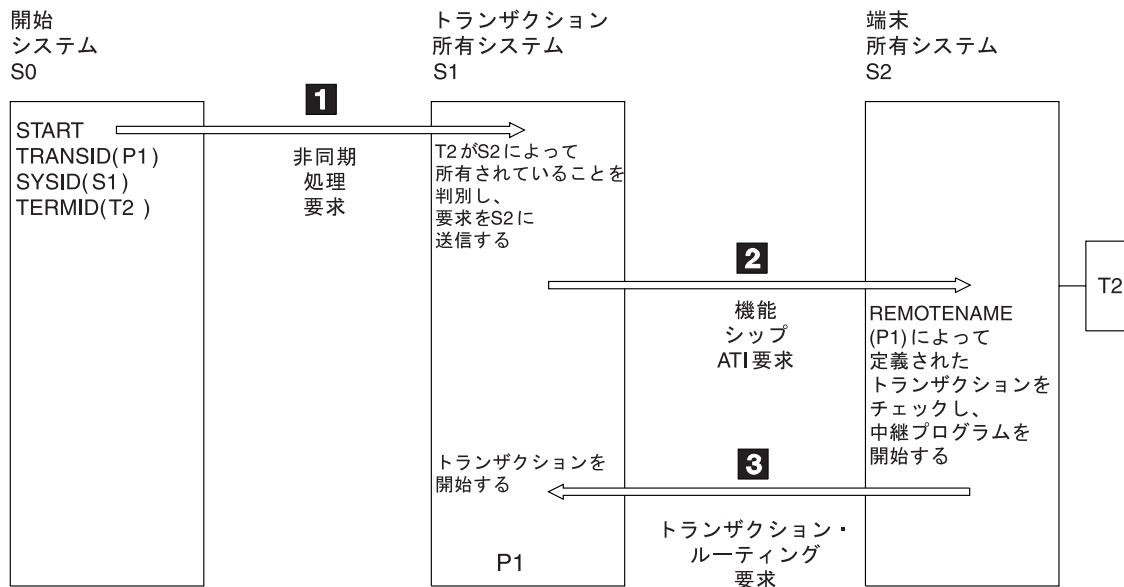


図7. 3 つの CICS システムが関与する自動トランザクション開始

## ATI 要求で伝送されない端末定義

46 ページの『端末定義の伝送』で説明したように、CICS は、トランザクション・ルーティングをサポートする必要があるとき、トランザクション所有システムに端末定義のコピーを伝送します。この定義は、トランザクション所有システムが後に受け取る ATI 要求で使用することができます。

しかし、端末定義は、ATI 要求を受け取った時点では伝送されません。ATI 要求に、トランザクション所有システムで認識されていない端末を指定していると、端末未知の状態が発生します。

**端末未知状態のユーザー出口:** System/390 CICS システムの場合のみ、2 つのグローバル・ユーザー出口 (XALTENF と XICTENF) が用意されており、端末未知の状態を処理する上で役立ちます。

---

## 中継プログラム

端末オペレーターがトランザクション・コードを入力し、トランザクションがリモート・システムにあることを CICS が判別すると、ローカル中継トランザクションが接続され、**中継プログラム**と呼ばれている CICS 提供プログラムを実行します。**中継プログラム**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を提供します。オープン・システム CICS において、中継とは共用ライブラリーで稼働する機能です。

CICS は中継トランザクションに関連付けられるプログラムを判別しますが、リモート・トランザクションのユーザーの定義によって、中継トランザクションのその他の属性が判別されます。これは通常、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、

中継トランザクションはタスク生成されるときに、システム間セッションを確立し、「実際の」ユーザー・トランザクションを開始するための要求を、アプリケーション所有システムに送信します。アプリケーション所有システムにおける端末は、**代用 TCTTE** (オープン・システム CICS では端末代用) という制御ブロックで表されます。この TCTTE はトランザクションの基本機能になり、トランザクションはこれと「実際の」端末入力の区別がつきません。トランザクションが基本機能に要求を出すと、その要求は、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伝送されます。中継トランザクションは次に、要求または出力を端末に出します。同じような方法で、端末状況と入力は、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伝送されます。

自動トランザクション開始は、同じような方法で処理されます。ATI によって開始されるトランザクションに、別のシステムに接続された端末が必要な場合は、中継トランザクションを開始するための要求が、端末所有システムに送信されます。端末が解放されていると、中継トランザクションはそれに接続されます。

中継トランザクションは、ユーザー・トランザクションの存続期間中に存在し、この期間はそのリモート・システムのみセッションを使用できます。ユーザーのトランザクションが終了すると、端末およびシステム間セッションを終了して解放するように、指示が中継トランザクションに出されます。

---

## 基本マッピング・サポート

CICS Transaction Server for Windows、オープン・システム CICS、および CICS/400 は、最小レベルの BMS のみをサポートします。(SEND TEXT はサポートされていません。)

BMS のマッピング操作は、ユーザーのトランザクションが実行されるシステム (つまり、アプリケーション所有システム) で実行されます。端末制御操作の場合と同じように、マップされた情報は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 を使用してトランザクション・ルーティングを行う場合は、BMS の制約と、アプリケーション所有領域と端末所有領域を異なるプラットフォーム上の CICS システムで稼働しているときは、2 つの領域でサポートできるレベルが異なることに注意してください。

---

## ルーティング・トランザクション (CRTE)

ルーティング・トランザクション (CRTE) は、接続されている CICS システムが所有しているトランザクションを端末オペレーターが呼び出せるようにする、CICS 提供のトランザクションです。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のない、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・システムに定義しておかなければなりません (または、ローカル・システムに「伝送可能」と定義しておかなければなりません)。リモート・システムが保護されている場合、端末オペレーターにはセキュリティ権限が必要です。使用可能なセキュリティのサポートについては、各プロダクトの「CICS 相互通信ガイド」を参照してください。

CRTE は、任意の 3270 表示装置から使用することができます。

CRTE を使用する際に、端末オペレーターは次のように入力してください。

```
CRTE SYSID=xxxx
```

ここで、xxxx はリモート・システムのローカル名です。次に、トランザクションは、ルーティング・セッションが確立されたことを示し、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、リモート・システムで認識されている必要なリモート・トランザクションの名前です。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。その後のすべての入力は、オペレーターがルーティング・セッションを終了するまで、リモート・システムに宛先指定されます。

セキュア・システムの場合、オペレーターは、通常、トランザクションを呼び出す前にサインオンしなければなりません。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 になります。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。CRTE 自体の使用も、セキュリティーによって制限することができます。

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インプリメントされますが、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に入れられます。

CRTE 機能は、特定のリモート・システムにある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。これは、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方法に代わる手段です。CRTE は、最終的な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。



---

## 第 9 章 分散プログラム・リンク

この章では、以下のトピックを記載します。

- 『DPL の紹介』
- 52 ページの『DPL を使用する理由』
- 52 ページの『同期』
- 53 ページの『DL/I および SQL データベース』
- 53 ページの『DPL を使用するときの制限』
- 53 ページの『DPL を使用するとき発生する異常終了』

---

### DPL の紹介

CICS アプリケーション・プログラムが EXEC CICS LINK コマンドを出すと、制御権は、LINK コマンドで名前を指定した 2 番目のプログラムに渡されます。その 2 番目のプログラムが実行されて、完了すると、制御権は、最初のプログラムの LINK コマンドの次の命令に戻されます。LINK コマンドに、連絡域のアドレスを渡すための COMMAREA オプションを使用していると、リンクされるプログラムは、リンクする側のプログラムにデータを戻すことができます。

分散プログラム・リンク (DPL) によって EXEC CICS LINK コマンドの用途が拡張され、リンクする側のプログラムとリンクされるプログラムを異なる CICS システムに置くことができるようになりました。システムは、同じ CICS プロダクトのコピーにすることも、異なる CICS プロダクトのコピーにすることもできます。

特定の環境で DPL を使用すれば、接続された CICS システム相互間のデータ・フローの数が減少するため、パフォーマンスは向上します。

LINK コマンドに SYSID オプションをコーディングするか、またはプログラムのローカル定義にリモート・システム名を指定すると、接続された CICS システムでリンクされたプログラムを稼働するように指定することができます。

52 ページの図 8 では、プログラム A が、別の CICS システムにあるプログラム B とリンクされています。矢印のついた線は、制御フローを表します。プログラム A からは、プログラム B は呼び出されたサブルーチンに見えます。

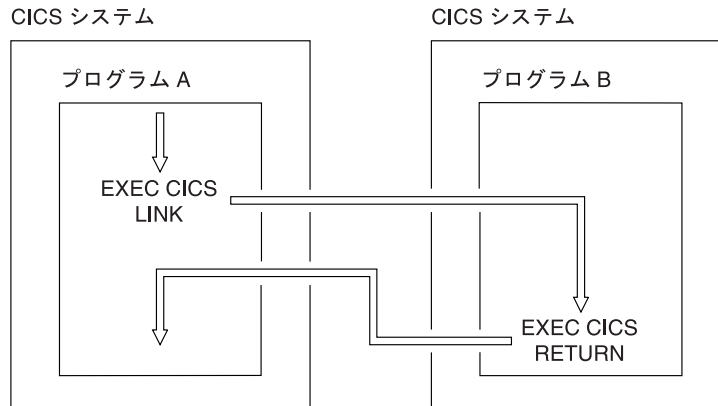


図8. 分散プログラム・リンク

## DPL を使用する理由

DPL を使用する理由を次に説明します。

- エンド・ユーザー・インターフェース (BMS 画面処理など) とアプリケーション・ビジネス・ロジック (データのアクセスや処理) を区別するため。これで、システム間でアプリケーションの一部を移送しやすくなります。たとえば、エンド・ユーザー・インターフェースを 370/390 システムからワークステーションに移動することができます。
- 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で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。
- 適用できる場合は、分散トランザクション・プログラミング (DTP) よりも簡単な解決法を用いるため。

## 同期

DPL には、同期を処理するための 2 通りの方法があります。

1. SYNCONRETURN をコーディングすると、リンクされるプログラムは、リンクする側のプログラムに制御権を戻す直前にリソースの更新をコミットします。2 つの通信システムには、別個の作業単位があります。  
リンクされるプログラムは、その実行中に 1 つまたは複数の同期点を取ることができます。ただし、リンクする側のプログラムへの応答 (通常またはロールバック) は、リンクされるプログラムからのリターン上で CICS が取った同期点の結果と対応し、リンクされるプログラムが開始したいずれの同期点の結果とは関係しません。
2. SYNCONRETURN をコーディングしなければ、リンクする側のプログラムは、SYNCPPOINT コマンドを出すか、タスクが終了した時点で暗黙的に、両方のシステムでコミットを開始します。

データ保全性についての考慮事項によって、SYNCONRETURN オプションを使用するかどうかを決定します。

---

## DL/I および SQL データベース

DPL は、CICS Transaction Server for Windows、オープン・システム CICS、または CICS/400 アプリケーション・プログラムが、リモート CICS システムにある DL/I データベース、SQL データベース、および BDAM ファイルにアクセスするための簡単な手段です。プログラムは、データベースまたはファイルを読み取って更新するアプリケーション・プログラム (データ所有システムにある) にリンクするだけです。

このデータにアクセスするための別の手段は、63 ページの『第 11 章 分散トランザクション・プログラミング』で説明する分散トランザクション・プログラミング (DTP) を使用する方法です。DTP は多少難しいので、開発するのに費用がかかりますが、適切に設計されていれば、使用効率はよくなるはずですが。

---

## DPL を使用するときの制限

DPL を使用してリンクできるプログラムには、多数の制約事項があります。

同期点を出すプログラムにリンクしてはなりません (LINK コマンドに SYNCONRETURN をコーディングしていない場合)。69 ページの『同期点の取り方』を参照してください。

端末は関与していないので、下記のコマンドを出すプログラムにリンクしてはなりません。

- リンクする側のプログラムが実行されているシステムへの端末制御コマンド
- 端末属性を検索するコマンド (ASSIGN コマンドなど)
- BMS コマンド
- バッチ・データ交換コマンド
- TCTUA をアドレス指定するコマンド (プログラムが連絡域を使用してデータを渡す)

任意の組み合わせの CICS プロダクトの間での DPL の場合、連絡域の最大長としては 32500 バイトをお勧めします。

---

## DPL を使用するとき発生する異常終了

リンクされるプログラムが異常終了すると、ミラー・トランザクションは、リンクする側のシステムに最後の異常終了コードを戻します。戻された異常終了コードは、リンクされるプログラムで**最後に**発生した異常終了です。このプログラムは、終了の前に他の異常終了を処理している場合もあります。



## 第 10 章 非同期処理

この章では、以下のトピックを記載します。

- 『非同期処理の紹介』
- 56 ページの『例』
- 56 ページの『非同期処理方式』
- 57 ページの『START/RETRIEVE コマンドを使用した非同期処理』
- 61 ページの『システム・プログラミングに関する考慮事項』
- 62 ページの『非同期処理の例 (NOCHECK を使用)』

### 非同期処理の紹介

非同期処理は、接続されたシステムの間でアプリケーションの処理を分散するための手段の 1 つです。分散トランザクション・プログラミングとは異なり、処理は非同期です。

分散トランザクション・プログラミングにおけるセッションは、それらの「会話」の存続期間中に 2 つのトランザクションによって保持されます。要求や応答があれば、それらを直接関連させることができます。

非同期処理における要求や応答は、別個のセッションで伝送されます。要求と応答の間には処理上の依存性はなく、また、応答のタイミングについてはなにも想定されていません。同期処理と非同期処理の違いについて、図 9 に示します。TRAN4 の開始処理は時間に依存することができ、システム B のスケジュール処理の制約によって遅らせることができます。

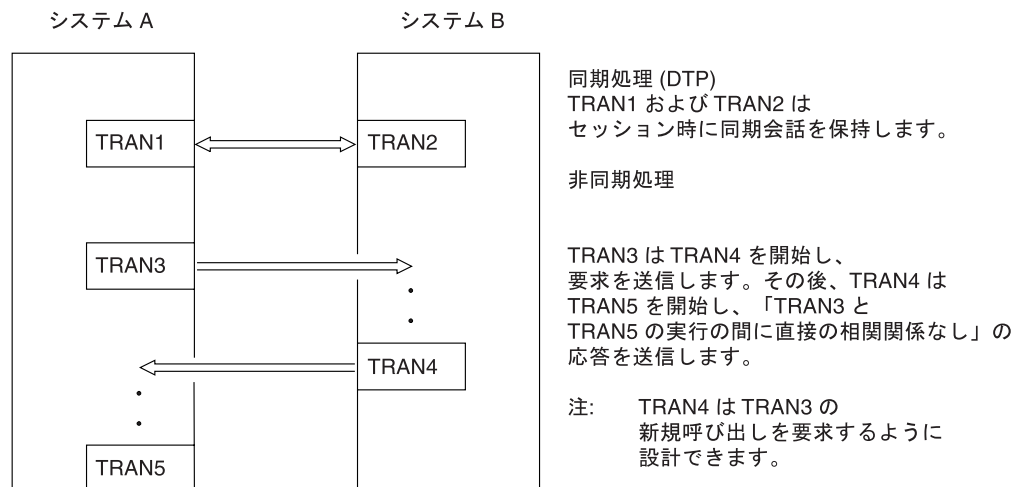


図 9. 同期処理と非同期処理の比較

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・リソースを結合する必要がないか、結合したくない状況で適用することができます。

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。たとえば、異なるシステムにあるデータの論理関連の更新を同時並行して処理するために使用することはできません。

55 ページの図9 において、同期トランザクション TRAN1 および TRAN2 によって行われたすべての変更は、リカバリー目的のために整合させることができることに注意してください。非同期トランザクション TRAN3、TRAN4、および TRAN5 により行われた変更を整合させることはできません。

---

## 例

一般的な非同期処理のアプリケーションは、リモート・データベースにおけるオンライン照会です。たとえば、借入限度額確認の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、応答を待たずに一連の照会を入力します。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ので、リモート・トランザクションの多数のコピーを並行して稼働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 (同じトランザクションの可能性もある) を開始してオペレーター端末に出力を伝送することによって、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するとは限りません。照会と応答の相関関係は、ユーザー・データ内のフィールドを使って行う必要があります。

---

## 非同期処理方式

CICS における非同期処理は、2 通りの方法で行うことができます。

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

1 つの CICS システムで行うのと同じ方法で、START コマンドを使用すると、リモート・システムにおいて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、CICS 機能シップの形式です。そのため、通常、アプリケーションに対しては透過的です。CICS は、トランザクションがリモートであることを次のいずれかの方法で認識します。

- トランザクション・リソース定義にリモートであることを指定する。
- START コマンドに、リモート CICS システムを指定する SYSID オプションを指定する。

リモート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 コマンドを使用して、要求に関連するデータを検索することができます。データ転送できるのは、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す 1 つのレコードだけです。

CICS トランザクションは、EXEC CICS ASSIGN STARTCODE コマンドを使用して、どのような方法で開始されたのかを判別することができます。

非同期処理の詳細については、57 ページの『START/RETRIEVE コマンドを使用した非同期処理』で詳しく説明します。

### 2. 分散トランザクション・プログラミング (DTP) を使用して行う、単一システムとは異なるシステム間方式。

DTP を使用してリモート・トランザクションに接続するときは、セッションを割り振って会話を開始します。これで、データを直接送信できるようになり、必要であれば、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わかります。たとえば、繰り返し SEND コマンドを使用して、複数レコード・ファイルを渡すことができます。会話はいくら短くても、会話の進行中の処理は同期で行われます。通常の DTP コマンドを使用して、エラー・リカバリーと同期点機能を使用することができます。

データを交換が終わると、会話を終了し、ローカル・トランザクションを終了しますが、リモート・トランザクションには非同期処理を続行させます。

DTP は、「CICS - CICS」通信だけでなく、「CICS - 非 CICS」通信にも使用できます。

データ変換が必要な場合、DTP アプリケーションに、これを処理するためのロジックを含む必要があります。

分散トランザクション・プログラミングの詳細については、63 ページの『第 11 章 分散トランザクション・プログラミング』で詳しく説明します。

---

## START/RETRIEVE コマンドを使用した非同期処理

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および取り消し

インターバル制御の START コマンドを使用して、リモート CICS システム内でトランザクション開始要求をキューに入れます。このコマンドは、効率よく機能シッパされます。リモート・システムでは、START コマンドを出すためにミラー・トランザクションが呼び出されます。

INTERVAL または TIME オプションを使用すると、伝送された START コマンドに時刻制御情報を指定することができます。コマンドが伝送される前に、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ます。リンクの終端が別の時間帯にある場合は、INTERVAL オプションを使用してください。

START コマンドに指定する時間間隔は、**リモート・トランザクションを開始しようとする時刻であって、要求をリモート・システムに伝送する時刻ではありません。**

**時間間隔が満了する前に** CANCEL コマンドを同じシステムに伝送すると、リモート CICS システムに伝送される START コマンドを取り消すことができます。取り消す START コマンドは、START コマンドとそれに関連した CANCEL コマンドに指定した REQID の値によって固有に識別されます。どのタスクも CANCEL コマンドを出すことができます。

## START コマンドを使用した情報の受け渡し

START コマンドには、リモート・トランザクションが開始されたときに情報を使用できるようにするためのオプションが多数あります。リモート・トランザクションが CICS システムにある場合は、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。

- FROM オプションに指定されたユーザー・データ。これは、リモート・トランザクションにデータを渡すための基本的な方法です。
- 追加のユーザー・データは QUEUE オプションを使用して位置付けられます。(たとえば) これは、一時記憶キューを識別することができます。
- TERMID オプションに指定された端末名。これは、開始されるときにリモート・トランザクションに関連付けられる端末の名前です。リモート・トランザクションを所有しているシステムに端末が定義されているのにそのシステムがその端末を所有していない場合は、トランザクション・ルーティングによってその端末が獲得されます。47 ページの図 7 を参照してください。

System/390 CICS システムでは、グローバル・ユーザー出口 (XICTENF と XALTENF) をコーディングしておく、端末がアプリケーション所有システムに定義されていない状況に対処することができます。

- RTRANSID および RTERMID オプションに指定され、応答に使用されるトランザクション名および端末名。これらのオプションを指定すると、リモート・トランザクションは、ローカル・システムに応答を渡します。(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。)

### START コマンドによる 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、トランザクションを呼び出すたびに、その START コマンドのユーザー・データの一部としてローカル・システム APPLID を送信するように準備することができます。トランザクションは、ASSIGN APPLID コマンドを使用してローカル APPLID を獲得することができます。

## システム間 START 要求のパフォーマンスの向上

一部の照会専用アプリケーションでは、複雑なエラー検査およびリカバリー・プロシージャを調整することはできません。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・システムへのデータ・フローとリモート・システムからのデータ・フロー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。パフォーマンスを向上させるには、複雑なリカバリー・プロシージャが必要になります。

一般的に、START NOCHECK コマンドは、56 ページの『例』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の結果タスク生成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。これで、1 つのメッセ



ージが適切なリモート・システムに送信されて、照会を行うトランザクションが非同期的に開始します。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。これで、オペレーターの端末にタスク生成されたトランザクションは、端末で返答を受け取るか別の要求を開始できるようにして、終了することができます。

リモート・システムは、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。

システムまたはセッションが失敗した場合、端末オペレーターは照会を入力し直して、同じ応答を受け取る準備をしなければなりません。オペレーターに役立つよう、それぞれの要求と一緒に相関フィールドを伝送しなければなりません。あるいは、すべての応答が自己記述型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は、62 ページの図 10 に示しています。

NOCHECK オプションは、リモート・システムとのリンクの確立が保留になって START コマンドのシップがキューに入れられているときは常に必要です (60 ページの『リモート・トランザクション用 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)。

## 作業論理単位への開始要求配布の組み込み

START コマンドに PROTECT オプションを指定すると、リモート・システムへの開始要求の配布を作業論理単位の一部に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リングしてはならないことを示します。(このトランザクションは、SYNCPOINT コマンドを出すか、終了すると、同期点を取ることができます。)

同期点が正しく完了すると、開始要求がリモート・システムに配布されたことが保証されます。ただし、リモート・トランザクションが完了したことも、リモート・トランザクションが開始されることも保証されません。

## NOCHECK オプションが指定された START 要求の送信の据え置き

NOCHECK オプションを指定した START コマンドを出すと、CICS はリモート・システムへの要求の伝送を据え置きます。

NOCHECK を指定した START 要求は、次のいずれかのイベントが発生するまで据え置かれます。

- トランザクションが同期点を出す。
- トランザクションが暗黙同期点で終了する。
- トランザクションがその他のタイプの機能シップ要求 (たとえば、WRITE TS や NOCHECK オプションを指定しない START) を出す。

- 同じリモート・システムについて出された多数の START NOCHECK 要求が、効率よく伝送を行うためにローカル・システムに蓄積される。

上記の規則には、次のような例外があります。

- System/390 CICS MRO 相互通信では、NOCHECK が指定された START コマンドは据え置かれません。
- オープン・システム CICS では、PROTECT オプションが指定された START NOCHECK によってバッファがフラッシュされます。

トランザクションからリモート・システムに最初に (または唯一の) 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任意の開始要求に PROTECT を指定すると、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、必要な伝送の個数が減ります。一連の要求は 1 つの SNA ブラケット内で伝送され、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。

## リモート・トランザクション用 START コマンドのローカル・キューイング

リモート・システムが活動状態になっていないかまたは接続が確立できないために、ローカル・トランザクションが START コマンドを伝送できる状態になっても、システム間機能を使用できないことがあります。このような場合の通常の CICS の処置は、SYSIDERR 状態にすることです。これは、NOCHECK オプションを使用したり、ローカルで要求をキューに入れて、必要なリンクが作業中のときにそれを送信するように CICS を調整したりすると避けることができます。

必要な接続が作動していない場合には、下記の 2 つの条件が満たされると、CICS はリモート・トランザクション用の START NOCHECK コマンドをキューに入れます。

1. SYSID オプションを START コマンドにコーディングしていない
2. トランザクションのローカル定義に LOCALQ(YES) を指定している

System/390 CICS プロダクトは、XISLCLQ グローバル・ユーザー出口をサポートします。この出口を使用すると、ユーザー作成のプログラムが LOCALQ オプションを指定変更してローカル・キューイングを決定できるようになり、柔軟性が高くなります。

## 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。

トランザクションがすでに活動状態で、同じ端末に関連付けられている場合に、インターバル制御についての通常の規則に従って、CICS は、ユーザー・データと端末 ID の両方を送るトランザクションの開始要求をキューに入れます。待機中、キューに入れられた要求に関連したデータには、活動状態のトランザクションがさらに RETRIEVE コマンドを使用すればアクセスすることができます。そのアクセスは、キューに入れられた開始要求を自動的に取り消します。

したがって、複数の開始要求に関連した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できます。通常、長時間実行されるトランザクションは、端末から複数の照会を受け入れ、リモート・システムに開始要求を送信することができます。開始されたトランザクションは、次々と RETRIEVE コマンドを出して、応答を受信することができます。応答がこれ以上存在しないときは、ENDDATA 条件を示します。

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の到着を保留にして、トランザクションを WAIT 状態にすることができます。全体的なアプリケーションの設計では、その後の開始要求がないためにトランザクションが永続待機状態にならないようにしなければなりません。たとえば、トランザクションにタイムアウトの間隔を定義することができます。(このオプションは、オープン・システム CICS ではサポートされません。)

## リモートで開始された CICS トランザクションによる端末の獲得

端末 (TERMID) 名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、端末をトランザクションで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。

### トランザクションの開始

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

CICS は、開始要求に指定した「端末」名がシステム名であることを知ると、そのシステムとのセッションのうち使用可能なものを選択し、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。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に入れられます。

---

## システム・プログラミングに関する考慮事項

非同期処理についてリソースを定義しなければなりません。それについては、これから簡単に説明します。リソースを定義する方法については、CICS プロダクト用のリソース定義ガイドで詳しく説明されています。

- リモート・システムとのリンクを定義しなければなりません。
- 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しなければなりません。ただし、SYSID オプションに明示的にリモート・システム名を指定した 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、これを行う必要はありません。
- QUEUE オプションを使用する場合、このオプションによって位置付けられた追加のユーザー・データは、START コマンドの伝送先のシステムからアクセス可能でなければなりません。
- START コマンドの RTRANSID オプションには、開始されたトランザクションが検索できて、後続の START コマンドで使用できる、トランザクション ID を指定します。RTRANSID オプションで指定したトランザクション名は、リモート・システムに定義しなければなりません。

たとえば、55 ページの図 9 では、TRAN3 の START コマンドには、TRANSID(TRAN4) および RTRANSID(TRAN5) オプションを指定しています。これらのオプションによって、TRAN4 が開始され、次に TRAN5 を開始するよう TRAN4 に指示します。これが、RTRANSID オプションの一般的な使い方です。実際には、使い方はアプリケーションに依存します。TRAN5 はシステム B に定義しなければなりません。

## 非同期処理の例 (NOCHECK を使用)

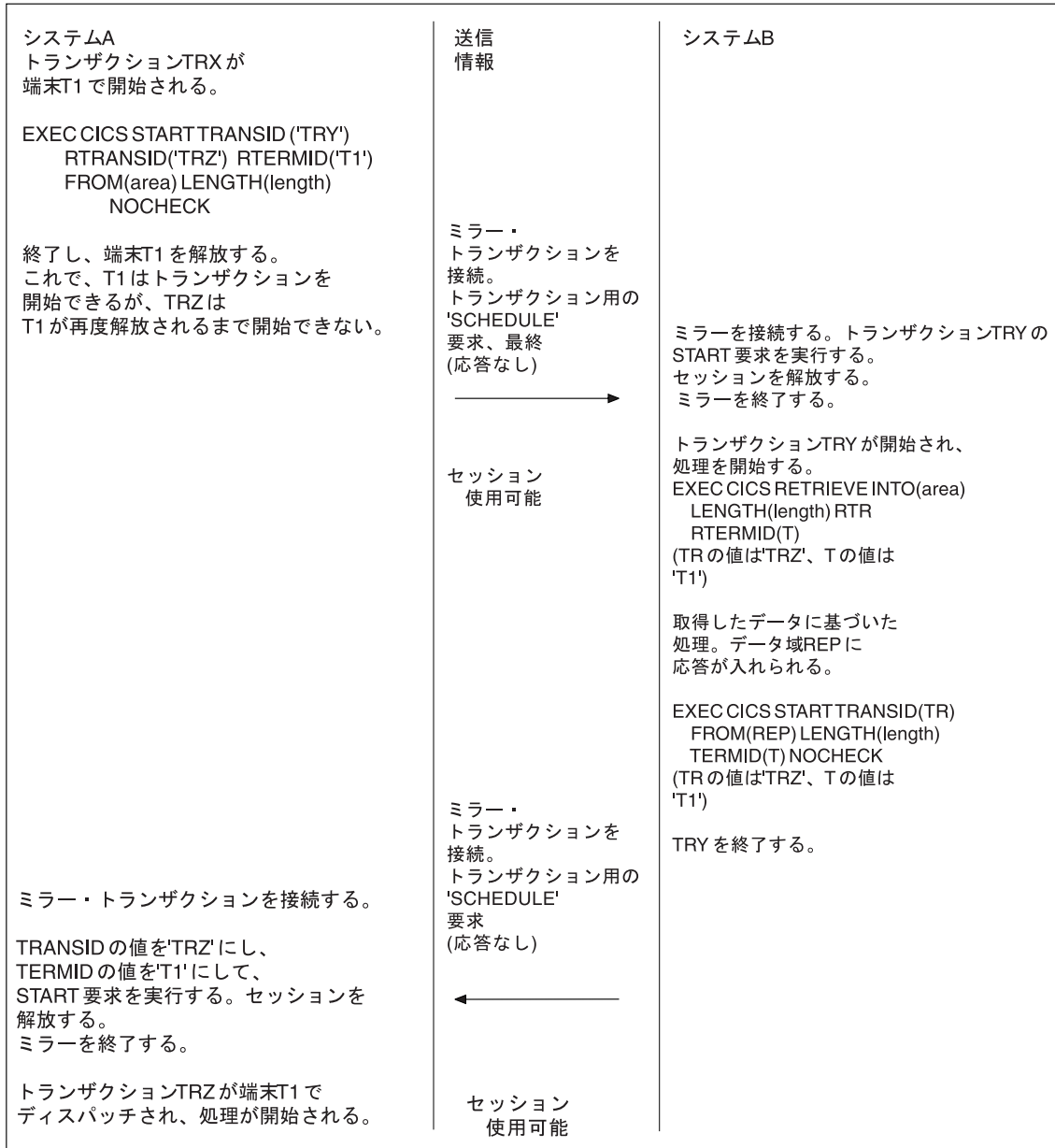


図 10. 非同期処理 - NOCHECK を使用したリモート・トランザクションの開始

図 10 は、START コマンドの NOCHECK オプションを使用する非同期処理の例を示しています。

---

## 第 11 章 分散トランザクション・プログラミング

CICS は、機能シップ、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、または分散プログラム・リンクを準備するときに、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 つのシステムの間データ交換が行われます。CICS 提供のプログラムは、会話を割り振るためのコマンドを出し、システム間でデータを送信および受信して、この交換を制御します。

CICS は、同等のコマンドを出して、アプリケーション・プログラムが自分の制御下で相互通信リンクを介して会話できるようにします。これらのコマンドを使用して、1 つのネットワーク内でいくつかのトランザクション・プログラムを介して業務上のトランザクションの機能を分散することができます。この技法のことを**分散トランザクション・プログラミング (DTP)** といいます。

DTP は、最も柔軟性のある、最強の CICS 相互通信機能ですが、最も複雑でもあります。本章では、基本的な概念について紹介します。

---

### 分散トランザクション・プログラミングを使用する理由

分散トランザクション・プログラミングが必要な理由は、他の相互通信機能にはコストがかかりすぎることと (『機能シップの制限』を参照)、このプログラミングには大きな利点があることです (65 ページの『分散トランザクション・プログラミングの利点』を参照)。

### 機能シップの制限

機能シップを使用すると、リモート・リソースにアクセスできます。トランザクション・ルーティングを使用すると、端末でリモート・トランザクションと通信できます。機能の点からいうと、これらの 2 つの機能は、ほとんどの相互通信の要件を満たします。しかし、設計基準は本来の機能を超えています。機械の負荷、応答時間、サービスの継続性、回線通信量、およびすべてのリソースの合理的な使用法などが、トランザクションの設計に影響する要因になります。

下記の 2 つの例では、機能シップの使用が明らかでも理想的な解決法とはいえませんが、場合を説明します。

#### 例 1

リモート・ファイルをブラウズしていくつかの基準を満たしたレコードを選択します。

**解決法 1:** 機能シップを使用します。CICS はリンクを介して、各 GETNEXT 要求を伝送し、ミラーはレコードを読み込み、それを要求者に伝送します。

これでは、レコードごとに 2 つのネットワーク・データ・フローが発生します。データ・フローはきわめて重要です。大規模なファイルをブラウズする場合は、オーバーヘッドが非常に高くなる可能性があります。

**解決法 2:** 分散プログラム・リンクを使用します。CICS は、ファイルを所有するシステム内で実行中のプログラムにリンクします。

**解決法 3:** DTP 会話を使用します。ローカル・トランザクションは選択基準を送信します。リモート・トランザクションは、選択されたレコードからキーと適切なフィールドを戻します。これで、リンクを介して送信されるフローの数とデータ量の両方が大幅に減少します。

## 例 2

スーパーマーケット・チェーンには、多数の支店があり、いくつかの流通センターから品物を仕入れます。それぞれの支店にはさまざまな種類の製品が貯蔵されています。支店におけるローカル在庫記録は、POS 端末からオンラインで更新されます。販売情報は、それぞれの流通センターごとに分類して、再注文や配達ができるようにセンターに伝送しなければなりません。

**解決法 1:** 機能シップを使用して、再注文が発生したときに再注文レコードをリモート・ファイルに書き込みます。この方法は簡単ですが、欠点がいくつかあります。

- データは、小さなパケットで不定期的にリモート・システムに伝送されます。つまり、使用中のリンクの使用効率がよくありません。
- POS 装置に関連したトランザクションは、リモート・システムとのセッションを競合します。つまり、POS において大幅な遅延が生じる可能性があるということです。
- リンクに障害が発生すると、支店での操作が大幅に延期されます。
- 相互通信活動が集中すると (ピーク時など)、POS 端末でのパフォーマンスが低下します。

**解決法 2:** それぞれの販売トランザクションは、再注文レコードを一時データ・キューに書き込み、端末との会話を続行します。

補充要求が緊急ではないので、データの分類や送信は、ピークを過ぎるまで行われません。これとは別に、一時データ・キューは、事前定義のデータ・レベルに到達したときに、送信側のトランザクションを起動するように設定することができます。どちらの方法であっても、送信側のトランザクションが行うジョブは同じです。

送信側のトランザクションは、機能シップを使用して再注文レコードを伝送することができます。各レコードは、分類処理の後で、対応するリモート・システムにあるリモート・ファイルに書き込まれます。

これも理想的な方法とはいえません。送信側のトランザクションは、正しい応答を受け取れるように、各レコードが書き込まれるまで待たなければなりません。リンクの使用効率がよくないことを別にしても、レコードとレコードの間で待機すると、処理全体の速度が非常に遅くなります。

**解決法 3:** 分散トランザクション・プログラミングを使用すると、支店のトランザクションは次のような処理を行います。

1. 再注文レコードを分類して、流通センターごとにファイルを作成します。
2. 適切な流通センターにあるパートナー・トランザクションに各ファイルを送信します。

次に、各流通センターでは、他のローカル・ファイルと同じように再注文レコードを処理します。この解決法を用いると、リンクの使用効率はよくなります。

## 分散トランザクション・プログラミングの利点

エンド・ユーザーがリモート・リソースにアクセスしなければならないため、多重システム環境でシステム間のデータ転送が必要になります。これらのリソースを管理する際に、ネットワーク・リソースが使用されます。しかし、ネットワークを使いすぎると、パフォーマンスは低下します。したがって、アプリケーション設計によって、あるリソースに関連している処理がそのリソースを所有している領域に置かれると、パフォーマンスが向上します。

DTP (非同期処理および分散プログラム・リンクと同様) を使用すると、中央処理側でアSEMBルすることでネットワーク・リソースを使いすぎるのではなく、発生した場所でデータを処理できます。ただし、DTP は、非同期処理や DPL よりも適応性があります。たとえば、次のようなことを行います。

- CICS システムおよび CICS 以外のシステムの両方と通信するために使用できます。
- さまざまなシステムで稼働するアプリケーション間で同期通信とデータ転送を行うことができます。
- いくつかの異なるシステムによって所有されるトランザクションとの共通インターフェースを提供します。
- 並列処理により応答時間を短縮することができます。
- アプリケーションがファイル・レコードの形式を知らなくてもいいように、機密性の高いファイルかデータベースとアプリケーションの間のバッファを提供します。
- リモート・システム用に宛先指定された、緊急データでないものをバッチ処理できるようにします。

---

## 会話

DTP におけるトランザクションは、直接相互にデータを渡します。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のことを**会話**といいます。いくつかのトランザクションを1 つの分散処理に含めることができますが、それらの間の通信は、ペア間のいくつかの独立型の会話に分かれます。このような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。

## 会話の開始およびトランザクションの階層

トランザクションは、リモート・システムと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、会話のパートナーになるトランザクションを活動化するためのタスク生成 要求が他のシステムに送信されます。

トランザクションは、他のトランザクションつまり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、独特の階層ができ上がります。その最上部には、端末開始トランザクションがあります。その構成例を 66 ページの図 11 に示します。トランザクション TRAA は、端末セッションを介してタスク生成されます。トランザクション TRAA はトランザクション TRBB をタスク生成します。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 をタスク生成します。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 をタスク生成します。これで、SUBR を稼働する 2 つの異なるタスクになります。

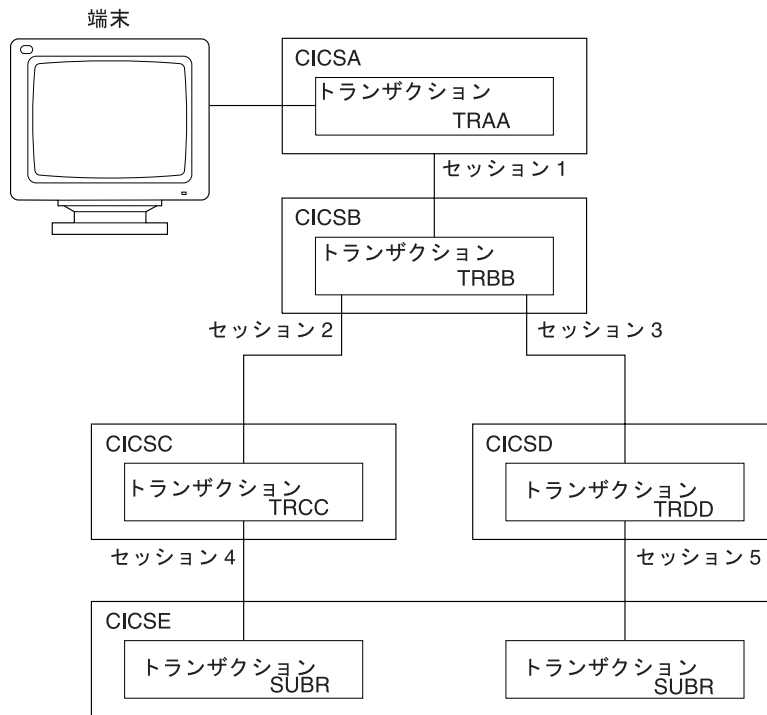


図 11. 多重システム構成の DTP

## 分散処理の構造

分散処理の構造は、動的に判別されます。これは、トランザクション定義に前もって指定することができません。各トランザクションには、インバウンド・タスク生成要求は 1 つしかありませんが、アウトバウンド・タスク生成要求はいくつあってもかまいません。トランザクションを活動化するセッションのことを**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 1 つのトランザクションによって割り振られるセッションのことを、割り振る側のトランザクションの**代替機能**といいます。図 11 のセッション 1 は、トランザクション TRBB の基本機能であり、TRAA の代替機能でもあります。トランザクションには基本機能が 1 つしかありませんが、代替機能はいくつもあります。トランザクション TRBB には、2 つの代替機能 (セッション 2 と 3) 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**になります。その会話のパートナーは、同じ会話の**バックエンド**になります。図 11 では、トランザクション TRBB はセッション 2 と 3 の会話のフロントエンドであり、セッション 1 の会話のバックエンドでもあります。(資料によってはフロントエンドのことを起動側、バックエンドのことを受信側と表すことがあります。) 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。必要であれば、バックエンドをフロントエンドを引き継ぐように用意することができますが、多重処理でこのようにすると、不要なコンパイルが行われることがあります。当件の詳細については、本章後方の同期の項で説明します。



## アプリケーションの設計

DTP の場合は、機能シップまたはトランザクション・ルーティングについての透過性を持っていません。会話によって、データがトランザクションからトランザクションに転送されます。これを正しく機能させるために、各トランザクションは、他のトランザクションが意図していることを知っておかなければなりません。したがって、フロントエンドとバックエンドは、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。複数の会話と複数のトランザクション・プログラムがあるときにも同様です。新しい会話はそれぞれ、全体の設計をより複雑にします。

64 ページの『例 2』の DTP 解決法 (解決法 3) では、トランザクションからトランザクションにデータのファイルが転送されます。つまり、この場合、一時データ・キューの内容全体がフロントエンドからバックエンドに伝送されます。次に複雑なことに、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) を戻します。ここでは、フロントエンドを適切な個所で会話の方向変換を要求するようにプログラミングします。

DTP アプリケーションの設計者が決定する必要があるその他の事項は以下のとおりです。

- 会話に使用する同期レベルをどのレベルにするか
- データ変換が必要な場合、会話のどちらのパートナーにデータ変換を処理させるか

## 制御フロー

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 つの伝送のことをフローといいます。SEND コマンドを出しても、必ずしもフローは発生しません。それは、ユーザー・データの伝送を遅らせること、つまり、なんらかのイベントが発生するまでユーザー・データをバッファーに保持することができるからです。APPC 体系によってデータ形式とパッケージが定義されます。CICS は、このようなことをユーザーの代わりに処理します。デバッグのためにフローをトレースする必要がある場合のみユーザーが考慮してください。

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。このヘッダーには、他の場所に制御情報を送るためのビット標識も入っています。たとえば、一方が送信を開始できることを他方に伝えたいときに、CICS がそのヘッダーのビットを設定して、会話の方向の変更を知らせます。

フローを最少に抑えるために、緊急ではない制御標識は、ユーザー・データの送信が必要になるまで蓄積されます。次に、それらはヘッダーに追加されます。

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。

## 会話の状態およびエラーの検出

会話が進行するにつれて、会話している両方のトランザクションにおける状態が変わります。会話の状態によって、出すことができるコマンドが決定されます。たと

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い場合は、データを送信または受信しようとしても意味がありません。同様に、バックエンドが会話の終わりを知らせると、フロントエンドはそれ以上のデータを受信する状態ではできなくなります。

どちらかの会話の終わりで、特定の状態から特定のコマンドを出すと、状態が変更されます。CICS はこの変更を追跡し、トランザクションがその現行状態に合わないコマンドを出すのを防ぎます。

## 同期

トランザクションの実行中に失敗することがあります。会話プロトコルは、エラーからリカバリーする際に役立ち、両側がお互いに状態をとどめているようにします。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中に失敗した場合に、関連リソースを不整合状態のままにしておいてはなりません。

### 例

トランザクションは、ファイルに書き込むデータのキューを別のシステムに伝送しています。受信側のトランザクションが異常終了しました。

その後の異常終了は防いでも、データを損失しないで処理を続行するという問題があります。受信されたキュー項目の数と、ファイルに正しく書き込まれたキュー項目数がわかりません。続行する際は、キューの内容とファイルの内容が一致していることがわかる箇所に戻るのが唯一安全な方法です。送信側のシステムは送信したキュー項目を復元し、受信側のシステムはファイルに書き込んだ項目を削除しなければなりません。CICS はこれを行う上で役立ちます（69 ページの『同期点の取り方』を参照してください）。

### ロールバックおよびバックアウト

確認できる最新の整合状態以降に、リカバリー可能リソースに対して行われたすべての変更を、アプリケーション・プログラムが取り消すことを**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。2つの用語は、置き換えて使用されることが多くあります。分散データ・リソースの間に整合性が失われていない状態のことを、**データ保天性**といいます。

**アプリケーション開始のロールバック:** CICS によって検出可能なエラー状態がなくても、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。顧客ごとに注文を入力しているときに、注文が通れば顧客の貸付限度額を超えることが、システムからオペレーターに伝えられます。顧客に相談するまで続行し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。トランザクションを、注文の最初の状態にデータ・リソースを回復して応答するように、プログラミングすることができます。同期レベル 2（7 ページの『同期レベル』を参照）では、すべてのリモート・パートナー・トランザクションで自動的にロールバックが行われます。

## 同期点の取り方

データ移動のログを取ると、ファイルとキューのバックアウトを用意することができます。しかし、これを行うと、プログラミングが非常に複雑になります。CICS が、ユーザーの代わりにリソースのリカバリーを準備すると、この問題は解決します。

処理において、認知されている整合状態になるようにリソースを宣言した個所のことを**同期点 (synchronization point)** といい、**syncpoint** と略すこともあります。同期点は、トランザクションの最初と最後に暗黙指定されています。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。

同期点が正しく取られると、リカバリー可能リソースに対するすべての変更が**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におけるデータ移動について保存しておいたすべての情報を消去するということです。これで、バックアウトが発生する可能性はなくなり、最後の同期点以降にリソースに行ったすべての変更が取り消しできなくなりました。

同期点が失敗すると、ロールバックが発生します。リカバリー可能リソースは、UOW の始まりの状態にリストアされます。

CICS は、リソースの変更をコミットしたりバックアウトしたりできますが、そのサービスとパフォーマンスはトレードオフです。このような機能を必要としないトランザクションもあります。リソースのリカバリーが問題ではないときは、簡単な同期方式を使用します。

## 3 つの同期レベル

APPC 体系では、3 つのレベルの同期が定義されています。

- レベル 0 - なし
- レベル 1 - 確認
- レベル 2 - 同期点

同期レベル 0 では、同期についてのシステム・サポートがありません。しかし、SEND コマンドと RECEIVE コマンドを使用すれば、データの交換にある程度の同期を達成することができます。

同期レベル 1 を選択すると、2 つの会話パートナー間で通信するために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。

このセクション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の変更のログ処理を開始しますが、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とります。

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、同期点要求またはロールバック要求が別のトランザクションに伝搬されるのは、発信元のトランザクションが他のトランザクションと会話していて、その会話に同期レベル 2 が選択されている場合のみです。

同期点やロールバックは、1 つのトランザクション内の任意の 1 つの会話に特有ではないことを覚えておいてください。これらは、現在トランザクション内にあるすべての同期レベル 2 の会話で伝搬されます。

トランザクションは、会話を開始する `CONNECT PROCESS` コマンドに必要な同期レベルを指定します。要求するレベルは、2 つのプロダクトでサポートされるレベルより高くなってはなりません。さまざまな同期レベルについてのサポートは、プロダクトによって異なります。8 ページの『CICS プロダクト間通信サポート』を参照してください。

---

## EXEC CICS と CPI コミュニケーションのどちらを使用すべきか？

一部の CICS プロダクトでは、APPC セッションで DTP 会話をコーディングするために 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) のうちのどちらかを選択できます。1 つは **CICS API** です。これは、APPC 体系で CICS を実施するためのエンド・ユーザー・インターフェースです。これは、EXEC CICS コマンドで構成され、CICS サポートのすべての言語で使用することができます。もう一つは、**通信用共通プログラミング・インターフェース (CPI コミュニケーション)** です。これは、システム・アプリケーション体系 (SAA<sup>®</sup>) によって定義されたコミュニケーション・インターフェースです。これは、使用される言語に適合した、プログラム呼び出しの形式の、一連の定義済み verb で構成されます。

71 ページの表 10 は、2 つの方式の比較を記載したもので、アプリケーションにどちらの API を使用すればよいのかを判別する上で役立ちます。

表 10. 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の可搬性。	SAA をサポートするシステムの間での可搬性。
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。同期レベル 0 および 1 のみがサポートされるトランザクション・ルーティングを除く。
プログラム初期設定パラメーター (PIP) データがサポートされる (System/390 CICS およびオープン・システム CICS のみ)。	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で基本機能として使用することができる。	ATI によって開始されたトランザクションで基本機能として使用することができない。
マップ式会話 (注意事項 3 を参照) は、CICS がサポートする任意の言語でプログラムできる。	マップ式会話は、CICS がサポートする任意の言語でプログラムできる。
基本会話 (注意事項 3 を参照) は、System/390 CICS システムのみで、アセンブラ言語または C 言語でのみプログラムできる。	基本会話は、System/390 CICS システムのみで、CICS がサポートする任意の言語でプログラムできる。

## 2 つの API についてのその他の注意事項

1. 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ができますが、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。つまり、**ハーフセッションが使用できるアプリケーション・インターフェースは 1 つだけです。**
2. 会話の片方のパートナーが CPI コミュニケーション呼び出しを使用し、もう片方が CICS API を使用することができます。つまり、**同じ会話のどちらかの終端にあるハーフセッションでは、異なるアプリケーション・インターフェースを使用することができます。**  
各ハーフセッションで異なる API を使用する会話を正しく調整するために、プログラマーは、両方の API を APPC 体系にマップする方法を詳しく知っていなければなりません。
3. 両方のインターフェース (CICS API と CPI コミュニケーション) は、APPC の**マップ式会話**をサポートします。この場合、システムはプロトコル・ヘッダーを提供して解釈し、アプリケーション・プログラムはユーザー・データを処理するだけです。APPC の**基本会話**では、送信側のアプリケーションは、通信プロトコルに必要なヘッダーをデータの前に付けなければなりません。受信側のアプリケーションは、このヘッダーを解釈しなければなりません。
4. CICS/VSE 2.3 は、CPI コミュニケーション API をサポートしていません。



---

## 第 3 部 付録





---

## 参考文献

このセクションには、相互通信に関連する、System/390 および非 System/390 CICS のライブラリーの資料をリストしてあります。

注: ユーザーが必要な情報を見つけやすいように、一部の資料は複数のカテゴリーにリストされています。

---

### CICS ファミリーの相互通信に関する資料

*CICS Family: Communicating from CICS on System/390*, SC34-6268

*CICS ファミリー プロダクト間通信ガイド*, SC88-9639

---

### System/390 CICS の相互通信に関する資料

#### CICS Transaction Server for z/OS バージョン 2 リリース 3

*CICS Distributed Transaction Programming Guide*, SC34-6236-00

*CICS External Interfaces Guide*, SC34-6244-00

*CICS Front End Programming Interface User's Guide*, SC34-6234-00

*CICS Intercommunication Guide*, SC34-6243-00

*CICS Internet Guide*, SC34-6245-00

#### CICS Transaction Server for z/OS バージョン 2 リリース 2

*CICS 分散トランザクション・プログラミング・ガイド*, SC88-9068-00

*CICS 外部インターフェース・ガイド*, SC88-9050-00

*CICS フロント・エンド・プログラミング・インターフェース ユーザーズ・ガイド*, SC88-9083-00

*CICS 相互通信ガイド*, SC88-9049-00

*CICS インターネット・ガイド*, SC88-9069-00

#### CICS Transaction Server for OS/390 リリース 3

*CICS 分散トランザクション・プログラミングの手引き*, SC88-7691-02

*CICS 外部インターフェース・ガイド*, SD88-7026-02

*CICS FEPI 使用者の手引き*, SC88-7692-02

*CICS 相互通信の手引き*, SC88-7694-03

*CICS インターネットの手引き*, SD88-7164-00

#### CICS Transaction Server for VSE/ESA リリース 1.1.1

*分散トランザクション・プログラミングの手引き*, SD88-7181

*外部 CICS インターフェースの手引き*, SD88-7188

*フロントエンド・プログラミング・インターフェースの使用者の手引き*, SD88-7182

*相互通信の手引き*, SD88-7184

#### CICS/VSE バージョン 2

*Distributed Transaction Programming Guide*, SC33-0898

*Intercommunication Guide*, SC33-0701

*サーバーの CICS クライアント・サポート*, SC88-7751

---

### CICS 非 System/390 の相互通信に関する資料

*CICS TS for Windows, Intercommunication*, SC34-6209

*Open System Intercommunication Guide*, SC33-1564

*CICS/400 相互通信*, SC88-7313

---

### CICS Transaction Gateway および CICS Universal Client

*CICS Transaction Gateway: プログラミング・ガイド*, SC88-9371

*CICS Transaction Gateway: プログラミング・リファレンス*, SC88-9370

CICS/VSE V2R3 サーバーの CICS クライアント・サポート、SC88-7751

---

## CICS 以外の資料

### SNA 資料

*Systems Network Architecture Technical Overview*、GC30-3073

*Systems Network Architecture Transaction Programmer's Reference Manual for LU Type 6.2*、GC30-3084

*Systems Network Architecture--Sessions Between Logical Units*、GC20-1868

*Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*、SC30-3269

*Systems Network Architecture LU 6.2 Reference-Peer Protocols*、SC31-6808

ー更新だけでなく、プロダクトのリリース変更時にも更新されるので、ご注意ください。)

- 資料のソフトコピー・バージョンが新規コレクション・キットについて更新されるときは、ハードコピー・バージョンと共用する資料番号は変更されません。また、版情報の日付は、元の資料の日付のままとなります。ソフトコピーをハードコピーと比較したり、ソフトコピーとソフトコピーを (たとえば、コレクション・キットの 2 つの版で) 比較したりするときは、資料のファイル名の最後の 2 文字をチェックしてください。番号が大きいほど、新しい資料です。たとえば、DFHPF104 は DFHPF103 より新しい資料です。CD-ROM ブックレットおよび README ファイルのなかの資料名称の隣に示されるアスタリスクは、新規または変更された資料を示します。
- ソフトコピー・バージョンへの更新は、変更箇所左に改訂コード (通常は “#” 文字) を付けて明確に示してあります。

---

## 最新の資料かどうかの判断

IBM では、新規情報および改訂情報で資料を定期的に更新しています。最初の出版では、資料のハードコピー・バージョンと BookManager® ソフトコピー・バージョンがそろって出版されます。ただし、それ以降の更新では通常、更新バージョンがハードコピーとして提供される前に、ソフトコピーの形で入手可能になります。

CICS 資料の場合、ソフトコピーの更新バージョンは、「*Transaction Processing and Data Collection Kit*」CD-ROM (SK2T-0730-xx) として定期的に提供されます。コレクション・キットが再発行された場合は、オーダー番号の接尾部 (-xx 部分) が更新されています。たとえば、コレクション・キット SK2T-0730-06 の方が、SK2T-0730-05 よりも新しいものです。また、コレクション・キットのカバーには日付が示されています。

最新の資料かどうかを判断する方法は次のとおりです。

- 接尾部の番号が大きい資料の方が、接尾部番号が小さいものより新しい資料です。たとえば、資料番号が SC33-0667-02 の資料は、資料番号が SC33-0667-01 の資料より新しい資料です。(接尾部番号は、特定リリース内でのハードコピ

---

## 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 システムの設定、実行、および保守に関するほとんどの作業は、以下のいずれかの方法で実行できます。

- CICS にログオンした 3270 エミュレーターを使用
- TSO にログオンした 3270 エミュレーターを使用
- MVS システム・コンソールとして 3270 エミュレーターを使用

ハンディキャップのある方のために、IBM パーソナル・コミュニケーションズ (Windows 95、Windows 98、Windows NT、および Windows 2000 の場合はバージョン 5.0.1。OS/2<sup>®</sup> の場合はバージョン 4.3。) は、アクセシビリティ機能を備えた 3270 エミュレーションを提供しています。この製品を使用すると、CICS システムで必要となるアクセシビリティ機能を提供できます。



## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、米国以外の国においては本書で述べる製品、サービス、またはプログラムを提供しない場合があります。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。

〒106-0032  
東京都港区六本木 3-2-31  
IBM World Trade Asia Corporation  
Licensing

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、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

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

IBM United Kingdom Laboratories, MP151,  
Hursley Park, Winchester, Hampshire,  
England, SO21 2JN

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。

---

## 商標

以下は、IBM Corporation の商標です。

AIX	Encina	
AS/400	IBM	SupportPac
AnyNet	IMS	System/370
BookManager	IMS/ESA	System/390
C/370	MVS	Systems Application Architecture
CICS	MVS/ESA	TXSeries
CICS/400	OS/2	VSE/ESA
CICS/ESA	OS/390	VTAM
CICS/MVS	OS/400	iSeries
CICS/VSE	Presentation Manager	z/OS
CICSplex	RISC System/6000	zSeries
DB2	RS/6000	

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または登録商標です。

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

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名などはそれぞれ各社の商標または登録商標です。

# 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]

アプリケーションの設計, DTP 67  
アプリケーション・プログラミング・インターフェース  
70  
一時記憶, 機能シップ 38  
一時データ, 機能シップ 38  
インターバル制御, 機能シップ 35  
エラーの検出, 会話 67

## [カ行]

開始側, 会話 66  
階層, トランザクションの 65  
外部表示インターフェース (EPI) 12  
外部呼び出しインターフェース (ECI) 11  
会話 65, 70  
エラーの検出 67  
開始 65  
基本 71  
基本機能 66  
状態 67  
制御フロー 67  
セッション 65  
代替機能 66  
データ・ヘッダー 67  
同期 68, 70  
バックエンド 66  
フロントエンド 66  
拡張 2 進化 10 進コード (EBCDIC) 16  
疑似会話 44  
機能シップ 31, 35, 42  
アプリケーションの透過性 36  
一時記憶 38  
一時データ 38  
インターバル制御 35  
制限 63  
同期 41  
動作の仕方 38  
ファイル制御 37  
変換プログラム 39  
ミラー・トランザクション 35, 39  
例 41  
DL/I データベース 37  
IMS データベース 37

基本会話 71  
基本機能 66  
基本マッピング・サポート (BMS), トランザクション・  
ルーティング 44, 49  
クライアント, CICS  
概要 11  
提供される機能  
外部表示インターフェース 12  
外部呼び出しインターフェース 11  
端末エミュレーション 12  
クライアント/サーバー・コンピューティング 11  
検索, START コマンドによって送信されたデータの  
60  
コード・ページ 16  
コミット, リソースの変更の 69

## [サ行]

作業単位 (UOW) 69  
受信側, 会話 66  
状態, 会話の 67  
情報交換用米国標準コード (ASCII) 16  
据え置き送信, START NOCHECK 59  
制御フロー 67  
セッション  
会話で使用される 65  
選択, 相互通信機能の 33  
相互通信機能 31  
簡単な定義 31  
選択 33

## [タ行]

代替機能 66  
端末, 定義の伝送 48  
端末未知状態 48  
中継トランザクション 43  
中継プログラム (DFHCRP) 48  
通信機能  
プロダクト・サポート 5, 11  
リストされた機能 5  
通信プロトコル  
APPC 6  
IPX 6  
LU6.2 6  
NetBIOS 6  
TCP/IP 6  
データ変換 15, 19  
コード・ページ 16

## データ変換 (続き)

- 数字データ 15
- 文字データ 15, 19
- ASCII 16
- EBCDIC 16

## データ保全性 6, 68

## データ・ヘッダー、APPC 体系 67

## 出口、ユーザー

- XALTENF 58
- XICTENF 58

## 伝送、端末定義の 46, 48

## 同期 6

- 機能シップ 41
- 同期レベル 7
- 2 フェーズ・コミット 7
- DPL 52
- DTP 68, 70

- コミット、リソースの変更の 69
- 作業単位 (UOW) 69
- データ保全性 68
- 同期点 69
- バックアウト 68
- ロールバック 68

## 同期点 69

## 同期レベル 7, 69

## 動的トランザクション・ルーティング 45

## トランザクションの階層 65

## トランザクション・ルーティング 32, 43, 50

- 疑似会話 44
- 基本マッピング・サポート (BMS) 44, 49
- 静的トランザクション・ルーティング 44
- 端末開始のトランザクション・ルーティング 44
- 中継プログラム 48
- 適格なセッション 43
- 適格な端末 43
- 伝送、端末定義の 48
- 動的トランザクション・ルーティング 45
- トランザクション・ルーティングの開始 44
- ルーティング・トランザクション、CRTE 49
- ATI (自動トランザクション開始) 46

## [ハ行]

## バックアウト 68

## バックエンド 66

## 非同期処理 55, 62

- 一般的なアプリケーション 56
- システム・プログラミングに関する考慮事項 61
- 同期処理 (DTP) との比較 55
- 非同期処理の開始 56
- DTP によって開始される 56
- START/RETRIEVE インターフェース 57, 61

## 非同期処理 (続き)

- 情報検索 60
- 端末の獲得 61
- パフォーマンスの向上 58
- リモート・トランザクションの開始 57
- リモート・トランザクションの取り消し 57
- ローカル・キューイング化、START 要求の 60
- NOCHECK オプション、START コマンド 58
- PROTECT オプション、START コマンド 59
- RETRIEVE コマンド 60
- START コマンドによって渡される情報 58
- “端末” がシステムの場合 61

## ファイル制御 37

## プロダクト間通信

- プロダクトの通信機能の定義 5
- 2 つのプロダクトの相互通信方法 8

## プロトコル、通信

- APPC 6
- IPX 6
- LU6.2 6
- NetBIOS 6
- TCP/IP 6

## フロントエンド 66

## 分散トランザクション・プログラミング 32

## 分散プログラム・リンク 32, 51, 53

## 変換プログラム 39

## [マ行]

## マルチバイト文字セット (MBCS) 16

## ミラー・トランザクション 35, 39, 41

- 多重ミラー 40
- チェーン・ミラー 40

## [ヤ行]

## ユーザー出口

- XALTENF 58
- XICTENF 58

## [ラ行]

## ルーティング・トランザクション、CRTE 49

## 例

- 機能シップ 41
- ATI 47
- ローカル・キューイング化、START 要求の 60
- ロールバック 68



## [ワ行]

割り振り、APPC 端末または接続の 44

## [数字]

2 バイト文字セット (DBCS) 16

2 フェーズ・コミット 7

## A

API 70

APPC 端末、トランザクション・ルーティング 44

APPC、通信プロトコル 6

APPLID、START コマンドによって渡される 58

ASCII 16

ATI (自動トランザクション開始) 46

トランザクション・ルーティングを使用した 46

CRTE ルーティング・トランザクションによって制限される 50

## C

CANCEL コマンド 57

CEMT マスター端末トランザクション、CRTE によって呼び出される 50

CICS Transaction Server for Windows

動的トランザクション・ルーティング 46

CICS クライアント

概要 11

サポートされるサーバー 13

提供される機能

外部セキュリティ・インターフェース 12

外部表示インターフェース 12

外部呼び出しインターフェース 11

端末エミュレーション 12

AIX 13

HP-UX 13

Linux 390 13

Microsoft Windows 13

Sun Solaris 13

CPI コミュニケーション 70

CRTE トランザクション 49

## D

DBCS、2 バイト文字セット 16

DL/I データベース

機能シップによってアクセスされる 37

DPL によってアクセスされる 53

DPL (分散プログラム・リンク) 51, 53

DTP (分散トランザクション・プログラミング) 63, 71

DTP (分散トランザクション・プログラミング) (続き)  
アプリケーションの設計 67

会話 65, 70

同期 68, 70

必要な理由 63

非同期処理との比較 55

利点 65

## E

EBCDIC 16

ECI (外部呼び出しインターフェース) 11

EPI (外部表示インターフェース) 12

ESI (外部セキュリティー・インターフェース)  
概要 12

EXEC CICS、API 70

## I

IMS データベース、機能シップ 37

IPX、通信プロトコル 6

## L

LU6.2、通信プロトコル 6

## M

MBCS、マルチバイト文字セット 16

## N

NetBIOS、通信プロトコル 6

NOCHECK オプション、START コマンド  
据え置き送信 59

パフォーマンスの向上 58, 59

ローカル・キューイング 60

## P

PROTECT オプション、START コマンド 59

## R

RETRIEVE コマンド 56

RETRIEVE コマンド、WAIT オプション 61

## S

SBCS、1 バイト文字セット 16

SNA 構成 19

SNA 構成 (続き)  
    CICS for AIX 25  
    System/390 CICS 24  
SNA 通信用の CICS の構成 19  
SNA 用語 19  
SQL データベース、DPL によってアクセスされる 53  
START コマンド 56  
    据え置き送信 59  
    ローカル・キューイング 60  
    NOCHECK オプション 58  
    PROTECT オプション 59  
START/RETRIEVE、非同期処理 57  
System/390 CICS  
    動的トランザクション・ルーティング 45

## T

TCP/IP、通信プロトコル 6

## U

UOW (作業単位) 69

## W

WAIT オプション、RETRIEVE コマンド 61

## X

XALTENF、ユーザー出口 58

XICTENF、ユーザー出口 58





Printed in Japan

SC88-9639-00



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