CICS Transaction Server for z/OS

共用データ・テーブル・ガイド

バージョン *5* リリース *5*

CICS Transaction Server for z/OS

共用データ・テーブル・ガイド

バージョン *5* リリース *5*

注記

本書および本書で紹介する製品をご使用になる前に、 83 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、IBM CICS Transaction Server for z/OS バージョン 5 リリース 5 (製品番号 5655-Y04) および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

原典： CICS Transaction Server for z/OS
Shared Data Tables Guide
Version 5 Release 5

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

© Copyright IBM Corporation 2005, 2018.

目次

この PDF について v

第 1 章 共用データ・テーブルの概要 . . . 1
共用データ・テーブルの概念 1
データ・テーブルの説明 2
CICS 保守データ・テーブル 2
ユーザー保守データ・テーブル 3
データ・テーブルの共用環境 3
データ・テーブルのソース・データ・セット 4
データ・テーブルのデータ・スペース 4
データ・テーブルのグローバル・ユーザー出口 . . . 6
共用データ・テーブル・サービスとリモート・ファイル・アクセス 6
データ・テーブルの共用の仕組み 7
LOGON 7
CONNECT 8

第 2 章 共用データ・テーブルの構成 . . 11
データ・テーブルの使用計画 11
CICS 保守データ・テーブルのパフォーマンス . . 11
ユーザー保守データ・テーブルのパフォーマンス 11
共用データ・テーブル用のストレージの使用 . . 11
共用データ・テーブルを使用する際の MVS JCL 要件 14
データ・テーブルとして使用するファイルの選択 14
データ・テーブル選択のために統計を使用する . 16
データ・テーブルのセキュリティー検査 20
共用データ・テーブルのサポートの使用準備 . . 21
データ・テーブルのリソース定義 23
CICS 保守データ・テーブルのリソース定義 . . 24
ユーザー保守データ・テーブルのリソース定義 . 25
DEFINE FILE コマンドでデータ・テーブルを定義する 26
データ・テーブルに関する EXEC CICS コマンド 30
データ・テーブルに関する CEMT コマンド . . 32

第 3 章 データ・テーブルへのアクセスのための開発 33
CICS 保守データ・テーブル用のアプリケーション・プログラミング 33
ロード中の CICS 保守データ・テーブルの使用 34
ユーザー保守データ・テーブル用のアプリケーション・プログラミング 35
ロード中のユーザー保守データ・テーブルの使用 36
共用データ・テーブルの仮想記憶間サービスの使用 36
接続 37
機能シップと仮想記憶間サービスの相違点 38
共用データ・テーブル・サービスと VSAM の相違点 39

第 4 章 ユーザー出口を使用したデータ・テーブルのカスタマイズ 43
CICS と共用データ・テーブル出口プログラム間の通信 43
XDTRD ユーザー出口 47
XDTAD ユーザー出口 48
XDTLC ユーザー出口 49
データ・テーブル用ユーザー出口の活動化 49

第 5 章 データ・テーブルの管理 51
データ・テーブルのオープン 51
データ・テーブルのクローズ 52
シスプレックス内での共用データ・テーブル・サポートの使用 53
シスプレックス内の共用データ・テーブル・サポートの概要 53
複製されたユーザー保守データ・テーブルのリフレッシュ方法 54
ユーザー保守データ・テーブルをリフレッシュするためのサンプル・プログラム 57

第 6 章 データ・テーブルのトラブルシューティング 71
データ・テーブル・サービスのトレース情報 . . . 71
共用データ・テーブルの入り口トレース・ポイントと出口トレース・ポイント 71
共用データ・テーブルの例外トレース・ポイント 75
データ・テーブル SVC からのエラーの分析 . . . 76
すべての共用データ・テーブル・トレース・ポイントの値 77
0B12 トレース・ポイントの値 77
0B19 トレース・ポイントの値 77
0B1A トレース・ポイントの値 78
AP 0B29 トレース・ポイントの値 78
0B2A トレース・ポイントの値 78
データ・テーブルのクロスメモリー・サービスからのエラーの分析 79
データ・テーブルに関するダンプ情報 80

特記事項 83

索引 89

© Copyright IBM Corp. 2005, 2018

この PDF について

この PDF では、CICS® の共用データ・テーブル・サービスに関する情報を提供しています。

本書は、以下の 1 つ以上の領域で CICS の共用データ・テーブルを扱うすべてのユーザーを対象としています。

- 計画
- アプリケーション・プログラミング
- リソース定義
- カスタマイズ
- 操作
- 問題判別

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

この PDF の作成日

この PDF は、2018 年 12 月 14 日に作成されました。

© Copyright IBM Corp. 2005, 2018

第 1 章 共用データ・テーブルの概要

CICS 共用データ・テーブル機能は、CICS ファイル管理サービスの拡張です。

共用データ・テーブルの概念

共用データ・テーブルを使用すると、データ・テーブルとして定義されているすべてのファイルを仮想記憶間サービスを使用して共用できる可能性があります。既存のデータ・テーブルのファイル定義を変更する必要はありません。

共用データ・テーブルの概念には、以下の面でより効率的であるというということが活用されています。

- CICS 機能シップではなく MVS™ 仮想記憶間サービスを使用して、同じ MVS イメージ内の複数の CICS 領域間でデータ・ファイルを共用する。
- DASD からではなく、メモリーからデータにアクセスする。
- VSAM サービスとローカル共用リソース (LSR) プールを使用するのではなく、CICS ファイル管理内で統合されたサービスを使用して、メモリーからデータ・ファイルにアクセスする。

データ・テーブルには、次の 2 つのバージョンがあります。

- CICS Transaction Server for z/OS® の一部として提供されている基本データ・テーブル・サポート
- CICS Transaction Server for z/OS の一部として提供されている共用データ・テーブル・サポート

仮想記憶間サービスを使用するということが、共用データ・テーブルの主な利点の 1 つです。これにより、現在機能シップを使用しているアプリケーションのパフォーマンスが向上し、機能シップのパフォーマンス・オーバーヘッドを受け入れられないアプリケーションでファイル共用が可能になります。

その他の主な機能拡張として、ほぼすべての読み取り要求がデータ・テーブルに対する使用をサポートされています。この機能拡張により、以下のようなアプリケーションにまでデータ・テーブルの使用が拡張されます。

- Browse requests (ブラウズ要求)
- 不正確なキーを使用する読み取り要求

仮想記憶間サービスの使用に加えて、共用データ・テーブルには、パフォーマンスやセキュリティーの面で以下のようなメリットがあります。

- 大部分の読み取りおよびブラウズ要求で機能シップが回避されるため、リモート・アクセスでパス長を大幅に削減できます。
- 仮想記憶間サービスを使用すると、要求は AOR によって処理されるため、FOR を解放して他の要求を処理できます。これにより、マルチプロセッサーをさらに効率的に活用できます。

© Copyright IBM Corp. 2005, 2018

- 共用データ・テーブル内のレコード情報は CICS 領域の外部に保管され、CICS システム・ダンプ (フォーマット済みと未フォーマットの両方) に含まれないため、データの安全性が高まります。
- CICS 保守データ・テーブルの場合、あらゆる形式の非更新のキー順アクセス (ブラウズ要求および不正確なキーによる読み取り要求を含む) が、データ・テーブルへの参照により処理されます。
- ユーザー保守データ・テーブルの場合、あらゆる形式の非更新のキー順アクセス (ブラウズ要求および不正確なキーによる読み取り要求を含む) がサポートされます。
- 同じソース・データ・セットを参照している、同時に開かれた任意の数のファイルで、1 つの CICS 保守データ・テーブルからデータを取得できます。
- XDTRD ユーザー出口の機能拡張により、データ・テーブルのロード中に一定の範囲のレコードをスキップできます。

データ・テーブルの説明

CICS ファイルは、DASD 上のデータ・セットを表現したものです。データ・テーブル・サービスを使用するようにファイルを指定すると、ファイルが開かれたときに CICS はデータ・セットの内容を MVS データ・スペースにコピーし、可能な場合は常にそのコピーを使用します。

データ・テーブル・サービスがレコードにアクセスする方法により、サービスは VSAM キー順データ・セット (KSDS) でのみ使用できます。KSDS は、ソース・データ・セット と呼ばれます。メモリー内のコピーは、データ・テーブル と呼ばれます。レコードをコピーする処理は、データ・テーブルのロード と呼ばれます。

ファイルが CICS アプリケーションによって読み取られるときに、レコードは通常データ・テーブルから取得されます。CICS アプリケーションによってファイルが更新されるときの影響は、そのファイルに対して定義したデータ・テーブルのタイプによって異なります。

CICS データ・テーブル・サービスでは、次の 2 つのタイプのデータ・テーブルがサポートされます。

- CICS 保守データ・テーブル (CMT)
- ユーザー保守データ・テーブル (UMT)

CICS 保守データ・テーブル

CICS 保守データ・テーブルは、ソース・データ・セットにレコードが自動的に反映されるデータ・テーブルです。ユーザーがファイルと更新すると、CICS によってソース・データ・セットとデータ・テーブルの両方が変更されます。

ソース・データ・セットとデータ・テーブルを単一のエンティティーとして扱うと、以下のようになります。

- ファイルへの変更がソース・データ・セットとデータ・テーブルの両方に行われます。

- 同じソース・データ・セットを使用するように別のファイルが定義されている場合、そのファイルによってソース・データセットに行われた変更がデータ・テーブルにも行われます。
- 同じソース・データ・セットを使用するように別のファイルが定義されている場合、データ・テーブルからそのファイルによってレコードを取得できます。

CICS 保守データ・テーブルは簡単に実装できるため、データ・テーブル・サービスについてユーザーが把握しておく必要のある事柄はほとんどありません。また、既存のアプリケーション・プログラムを変更する必要はなく、ファイルの完全リカバリー・サポートは維持されます。

レコード・レベル共用 (RLS) モードでアクセスされるデータ・セットを、CICS 保守データ・テーブルのソースとして使用することはできません。ソース・データ・セットには、非 RLS モードでアクセスする必要があります。

ユーザー保守データ・テーブル

ユーザー保守データ・テーブルは、ソース・データ・セットにレコードが自動的に反映されないデータ・テーブルです。ユーザーがファイルを更新すると、CICS によってデータ・テーブルのみが変更されます。

ユーザー保守データ・テーブルは、ロード後はそのソース・データ・セットから独立しています。つまり、データ・テーブルが更新されても、ソース・データ・セットは更新されません。ユーザー保守データ・テーブルを使用すると、更新要求に加え、読み取り要求に対しても、ソース・データ・セットでのアクティビティーを除去することができるため、データ・テーブルを使用する利点を最大限に活用できます。そのため、ユーザー保守データ・テーブルは特に、一時的な特徴を持つデータへの更新が頻繁に行われるアプリケーションに適しています。

ユーザー保守データ・テーブルではいくつかファイル操作がサポートされていません。そのため、既存のアプリケーション・プログラムに多少の変更が必要になる場合があります。また、トランザクション障害後のファイルのリカバリーはサポートされていますが、システム障害後のリカバリーはサポートされません。

非 RLS または RLS モードのいずれかでアクセスされる基本 VSAM KSDS を、ユーザー保守データ・テーブルのソース・データ・セットとして使用できます。データ・セットにアクセスする他のファイル定義があり、そのデータ・セットが他の CICS 領域によって更新される場合、RLS モードのデータ・セットをユーザー保守データ・テーブルのソースにすることが推奨されます。

データ・テーブルの共用環境

データ・テーブルを共用するための環境は、非 RLS モードでアクセスされるファイルの場合と同じです。

1 つの CICS 領域がデータ・テーブルを所有している場合、その領域はファイル所有領域 (FOR) と呼ばれます。データ・テーブルを使用するその他の領域は、アプリケーション所有領域 (AOR) と呼ばれます。FOR では、ファイルはローカル・ファイル と呼ばれ、AOR では、ファイルはリモート・ファイル と呼ばれます。

共用データ・テーブルのコンテキストでは、FOR はサーバー とも呼ばれ、AOR はリクエスター とも呼ばれます。

同じ領域が、一部のデータ・テーブルについては FOR になり、その他のデータ・テーブルについては AOR になる場合があります。

これらの相互通信の概念については、相互通信方式を参照してください。

共用データ・テーブル・サポートでは、可能な場合は常に領域間共用が使用され、CICS 要求側領域と同じ MVS イメージにあるデータ・テーブルへのアクセスが提供されます。つまり、同じ MVS イメージ内の大部分の読み取りアクセスは、共用データ・テーブル・サービスを使用する領域間共用で処理できます。要求に領域間共用を使用できない場合、機能シップが使用されます。つまり、同じ MVS イメージ内の CICS 領域からの更新要求と、異なる MVS イメージ内の CICS 領域からのすべての要求で、機能シップが使用されます。 CICS 保守データ・テーブル用のアプリケーション・プログラミングとユーザー保守データ・テーブル用のアプリケーション・プログラミングには、領域間共用と機能シップのいずれかでどのような場合にコマンドが処理されるかが説明されています。

注**:** 同様に、XCF/MRO では、異なる MVS イメージにある CICS 領域間での共用データ・テーブル・アクセスは提供されません。

共用データ・テーブルのサポートは、主に MVS イメージ内でのデータ共用を目的としていますが、共用のユーザー保守データ・テーブルへの読み取りアクセスのみを必要とするか、あるいは最新ではない可能性のあるデータで動作できるアプリケーションのシスプレックス環境にサポートを拡張できます。データ・テーブルは、シスプレックス内の各 MVS 領域全体にわたって複製し、定期的に更新する必要があります。シスプレックスにおける共用データ・テーブル・サポートの使用を参照してください。

データ・テーブルのソース・データ・セット

ソース・データ・セットは代替索引ではなく、基本 VSAM KSDS である必要があります。ただし、代替索引を介して KSDS に行った更新は、CICS 保守データ・テーブルに反映されます。

KSDS の VSAM 定義によって、最大レコード長とキー長の値が指定されます。

ユーザー保守データ・テーブルの場合、更新はソース・データ・セットとその代替索引のどちらにおいても反映されません。ユーザー保守データ・テーブルは、ロードが完了した後はソース・データ・セットからは完全に独立します。

データ・テーブルのデータ・スペース

データ・テーブルのレコードは、データ・テーブルが複数の領域で共用されているかどうかに関係なく、1 つ以上の MVS データ・スペースに保管されます。 CICS 領域ごとに別個のデータ・スペースのセットが使用されます。

データ・スペースの初期セットである DFHDT001 (テーブル・エントリー記述子用)、DFHDT002 (索引ノード用)、および DFHDT003 (最大 2 GB までのレコー

ド・データ用) は、データ・テーブルとして定義されている最初のファイルが領域で開かれると取得されます。DFHDT004 以降の追加データ・スペースは、領域ごとに最大で 100 個のデータ・スペースまで、レコード・データ用に必要に応じて割り振ることができます。これらのデータ・スペースは、その領域が所有しているすべての CICS データ・テーブルで使用され、CICS 領域がシャットダウンされるまで保持されます。

各データ・スペースの最大サイズは 2 GB あるため、CICS 領域で割り振れるデータ・スペース・ストレージの最大量は (オペレーティング・システム・リソースが十分であると想定すると) 200 GB です。MVS 出口 IEFUSI を使用して、所定のアドレス・スペースが所有できるデータ・スペースの総量を制御できます。これにより、最大サイズをこの値未満、さらには 2 GB 未満にまで下げることができます。この制限内で、CICS は 16 MB 単位でデータ・スペース・ストレージを割り振り、その後このストレージを、、テーブル・エントリーの記述子または索引ノード用に 32 KB の増加で、レコード・データ用に 128 KB の増加でデータ・テーブルに再度割り振ります。データ・テーブル用に新しい増加ストレージが必要だが、既存のすべてのデータ・スペース・ストレージが既にテーブルに割り振られている場合、CICS はデータ・スペースを 16 MB 拡張しようとします。データ・スペースがレコード・データ用のものであるが、既にその最大サイズの 2 GB に達しており、データ・スペースの最大数にはまだ達していない場合、CICS は既存のデータ・スペースを拡張するのではなく、データ・スペースの既存セットの論理拡張として扱われる新しいデータ・スペースを作成します。データ・スペースが最大サイズの 2 GB に達するか、インストールの IEFUSI 出口によって設定されたデータ・スペースの合計サイズに達したために CICS がそれ以上のデータ・スペース・ストレージを割り振ることができないために、CICS がテーブル・エントリー記述子または索引ノード用のデータ・スペースを拡張できない場合、CICS はデータ・スペースが現在フルであることを通知します。

データ・スペースがフルになると、追加ストレージが必要なすべての共用データ・テーブル要求はストレージの不足により失敗します。 CICS 保守データ・テーブルの場合、影響を受けるレコードに対するそれ以降のすべての読み取り (そのキーに近い概算読み取りを含む) では、要求がファイル所有領域から発行されていない場合、機能シップを使用して VSAM データ・セットにアクセスする必要があります。ユーザー保守データ・テーブルの場合、レコードをテーブルに書き込めなくなります。データ・テーブルへのアクセス用の開発には、この状況で返される応答に関する情報が記載されています。

CICS には、現在割り振られているデータ・スペース・ストレージの合計量を表示する機能はありません。ただし、CICS ファイル制御統計を使用して、データ・テーブルごとに、割り振り済みのストレージと使用中のストレージを正確に示すことができます。特に、フィールド A17DTALD にはテーブルに現在割り振られているデータ・スペース・ストレージの量 (KB 単位) が含まれています。

データ・テーブルによって使用されるデータ・スペース・ストレージは、FOR でファイルが閉じられると解放されます。このストレージは、データ・テーブルを使用していた AOR の保全性が保護される方法で再利用できます。

データ・テーブルのグローバル・ユーザー出口

データ・テーブル・サービスによって実行される通常の処理を拡張するために、3 つのグローバル・ユーザー出口が提供されています。

- XDTRD は、ファイルが開かれるときに、ロード中にデータ・テーブルにコピーされるレコードを選択します。ユーザー保守データ・テーブルの場合、これはレコードの変更にも使用できます。
- XDTAD は、新しいレコードがファイルに追加されるときに、データ・テーブルにコピーされるレコードを選択します。
- XDTLC は、ロード操作の最後に処理を行います。

共用データ・テーブル・サービスとリモート・ファイル・アクセス

以下の図は、別の領域にある CICS ファイルにアクセスするために、機能シップを使用する場合と、共用データ・テーブル・サービスを使用する場合の相違を示しています。

機能シップの使用

この図は、別の CICS 領域が所有するデータ・セットにアクセスするために機能シップを使用する場合を示しています。

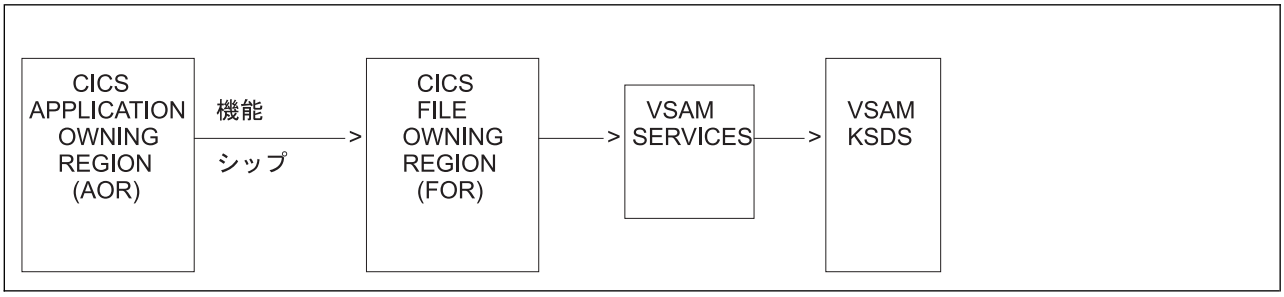

図 *1.* 機能シップを使用したデータ・アクセス

共用データ・テーブル・サービスの使用

この図は、データ・テーブルにアクセスするために FOR で共用データ・テーブル・サービスを使用することで、多数の AOR がどのようにして仮想記憶間サービスを使用して読み取りまたはブラウズを実行できるかを示しています。(更新要求と、ソース・データ・セットにアクセスする必要のあるすべての要求については、図 1 に示されているものと同じ方法で、機能シップが使用されます)。

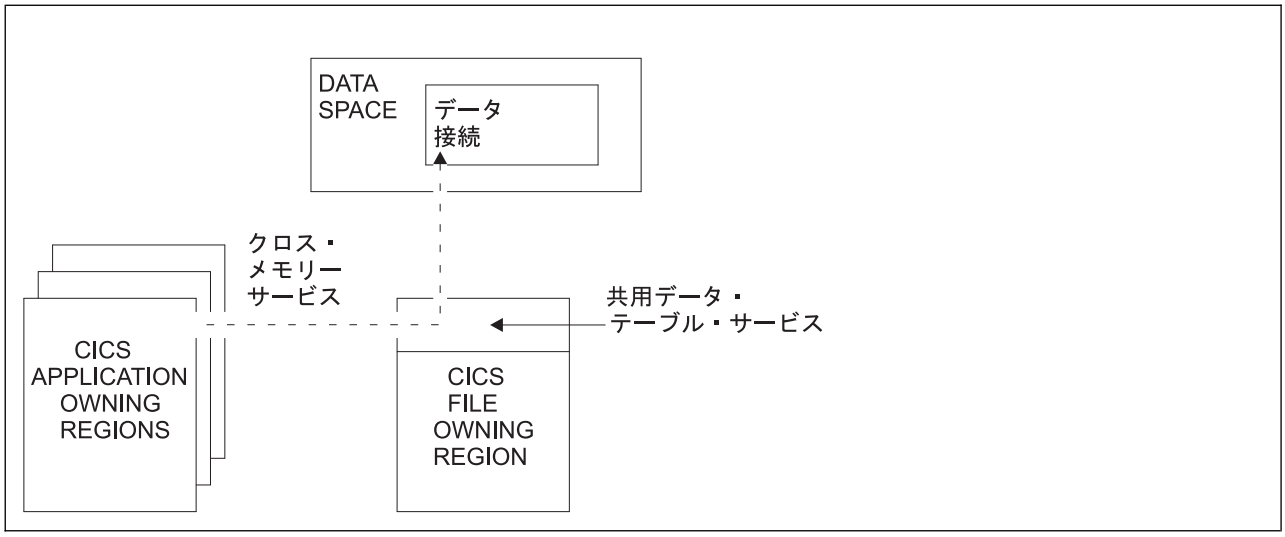

図 2. 共用データ・テーブル・サービスを使用したデータ・アクセス： この図は、読み取り専用アクセスのみを示しています。

データ・テーブルの共用の仕組み

2 つの操作、LOGON および CONNECT によってデータ・テーブルの共用が設定されます。

LOGON

データ・テーブルとして定義された最初のファイルが FOR で開かれると、FOR はそれ自体を共用データ・テーブル・サーバーとして登録しようとします。この操作は自動的に実行され、**SDT LOGON** と呼ばれています。ファイルを開く操作は、ファイルに最初にアクセスする AOR または FOR によって行われます。

LOGON が正常に完了したかどうかに関係なく、ファイルは開かれ、データ・テーブルはロードされます。LOGON が正常に完了すると、MVS オペレーティング・システム内のその他のすべての CICS 領域に対して、データ・テーブルが使用可能になったことが通知されます。

永続的な状態によって LOGON が失敗すると (CICSが MVS サブシステムとして定義されていない場合など)、CICS の実行中にそれ以上の LOGON は試行されません。

一時的な可能性がある状態によって LOGON が失敗すると、データ・テーブルとして定義されているファイルが次回開かれるときに、別の LOGON が試行されます。このタイプの状態には、以下のようなものがあります。

- セキュリティー検査の失敗
- ストレージを取得できない
- プログラムをロードできない

セキュリティー検査に失敗して領域の LOGON 要求が拒否されると、データ・テーブルとして定義されているファイルが開かれるたびに、セキュリティー違反のメッセージが発行される可能性があります。

正常にログオンした FOR では、それ以降の CICS 実行の間、その状態が保たれます。つまり、それ以上の LOGON 要求は発行されません。

CONNECT

共用データ・テーブルがサポートされている AOR によってリモート・ファイルに対する読み取り要求が発行される (または参照シーケンスが開始される) と、CICS はそのファイルのデータ・テーブルに対して接続を確立しようとします。この操作は自動的に実行され、**SDT CONNECT** と呼ばれます。

FOR が共用データ・テーブル・サーバーとして登録されている場合、CICS は AOR から FOR に仮想記憶間リンクを確立し (セキュリティー検査の対象)、共用データ・テーブル・サーバーを呼び出して、ファイルに使用可能なデータ・テーブルがあるかどうかを確認します。ある場合、AOR とデータ・テーブル間に接続が確立されます。

CONNECT に成功すると、接続が確立されている間のファイルへのアクセスには、可能なかぎり仮想記憶間サービスが使用されます。

CONNECT に失敗すると、ファイル要求は、共用データ・テーブルのサポートを使用できない場合とまったく同様に機能シップされます。後続のリモート・ファイル要求に対して行われる処置は、以下に示されているように、失敗の種類によって異なります。

永続的な状態が原因で CONNECT が失敗する場合 (CICSが MVS サブシステムとして定義されていない場合など)、CICS の実行中にそれ以上の CONNECT は試行されません。

ファイル所有者の制御下にない、一時的な可能性がある状態によって CONNECT が失敗する場合、約 10 分間が経過した後に、該当する次の要求に対して別の CONNECT が試行されます。このタイプの状態には、以下のようなものがあります。

- セキュリティー検査の失敗
- ストレージを取得できない
- プログラムをロードできない

セキュリティー検査に失敗して領域の CONNECT 要求が拒否されると、関連するセキュリティー違反のメッセージが 10 分間隔で発行される可能性があります。

ファイル所有者の制御下にある、一時的な可能性がある状態によって CONNECT が失敗する場合、MVS システムの共用アクセスで少なくとも 1 つの新しいファイルが使用できるという通知に続いて、次に該当する要求に対し、別の CONNECT 要求が試行されます。このタイプの状態には、以下のようなものがあります。

- ファイル所有者がサーバーとしてログオンしていない
- ファイルがデータ・テーブルと関連付けられていない
- ファイルが無効になっているが、データ・テーブルと関連付けられている

- ファイルが閉じられているが、データ・テーブルとして定義されている

リモート・ファイルに AOR が正常に接続すると、以下のいずれかのイベントが発生しない限り、接続されたままになります。

- AOR によってそのリモート・ファイル定義が削除される

  この場合、接続は即時に切断されます。

- FOR によってファイルが閉じられるか、無効に設定される

  この場合、切断は次の非更新要求時にスケジュール設定され、現在のすべての参照シーケンスが終了した後に、有効になります。切断を参照してください。

これらのイベントが後で元に戻された場合、以前と同様の方法で有効な接続が確立されます。

新しいファイルを共用アクセスで使用できることを示す通知

FOR によってデータ・テーブルが開かれると、CICS 保守データ・テーブルのロードを開始した時点、またはユーザー保守データ・テーブルのロードが完了した時点で、CONNECT の試行に対してそのデータ・テーブルを使用できるようになります。その他の CICS 領域には、データ・テーブルが使用可能になったことが通知されます。通知は、以前は無効に設定されていたデータ・テーブル (または CICS 保守データ・テーブルを使用するファイル) が有効になった場合にも発行されます。

第 2 章 共用データ・テーブルの構成

仮想記憶間サービスを使用してファイルを共用するように、共用データ・テーブルを構成できます。共用データ・テーブル機能は、CICS ファイル管理サービスの拡張機能で、機能シップを使用するアプリケーションのパフォーマンスを改善できます。

データ・テーブルの使用計画

データ・テーブルを使用する主な理由は、パフォーマンス上の利点を活用することです。

このセクションでは、診断、変更、またはチューニングについて説明します。

CICS 保守データ・テーブルのパフォーマンス

ファイルのすべてのデータ・レコードと索引レコードが完全に LSR プールに含まれている場合は、そのファイルを CICS 保守データ・テーブルとして定義しても DASD I/O アクティビティーは減りません。しかし、CPU 消費量が減少する可能性が大いにあります。また、LSR プール内のバッファー数を減らすこともできます。

ファイルが LSR プール内に完全に含まれていない場合、CICS 保守データ・テーブルを使用すると、DASD I/O アクティビティーと CPU 消費量の両方とも減少することになります。

ローカル共用リソース (LSR) プール内にある VSAM KSDS と比較して、CICS 保守データ・テーブルの CPU 消費量をどの程度節約できるかは、アプリケーションの使用状況に応じて異なります。

ユーザー保守データ・テーブルのパフォーマンス

ユーザー保守データ・テーブルのロード後、すべてのデータ・テーブル操作から DASD I/O アクティビティーが除外されるので、LSR プール内にある VSAM KSDS と比較して CPU 消費量を大幅に節約できます。

共用データ・テーブル用のストレージの使用

共用データ・テーブルにより、メモリー内のデータを効率的に使用できるようになります。つまり、ストレージの使用量が少し増えるというコストだけで、パフォーマンス上の大きな利点が生じます。

このストレージ使用の概説では、実記憶域と仮想記憶域や、アドレス・スペース・ストレージとデータ・スペース・ストレージなどの、さまざまなタイプのストレージの違いについて理解していることが前提になっています。ほとんどの場合、データ・スペース・ストレージがストレージとして使用されます。このストレージは、アドレス・スペース仮想記憶域から分離された仮想記憶域です。

共用データ・テーブルは仮想記憶域を以下のように使用します。

© Copyright IBM Corp. 2005, 2018

- レコード・データは DFHDT003、DFHDT004、DFHDT005 と続くデータ・スペースに保管され、必要に応じて新しいデータ・スペースが割り振られます。ロード時の合計レコード・データ・ストレージは、基本的にすべてのレコードの合計サイズ (キーはテーブル・エントリー・ストレージに保管されるので除きます) に少量の制御情報を加算したサイズになります。データ・スペース・ストレージは、16 MB 単位で獲得され、個別のテーブルに割り振られる際に 128 KB ずつ増加します。その後ストレージは、ページ位置合わせされたフレームに 2 次割り振りされます。このフレームは、テーブルの最大レコード長を十分組み込むことができる大きさです。データ・テーブル・フレームは、大まかに VSAM 制御間隔と同等で、通常は同じようなキーがあるレコードのセットを保持します。可能な場合、新しいレコードはそれぞれ、近接するキーのうち小さな方の値を持つ既存のレコードと同じフレームに保管されます。

  ロード後に多くのレコードの長さが増加したり、新しいレコードがファイルの大部分にランダムに追加されたりすると、ストレージの量は増え、最大で元のサイズの 2 倍になることがあります。
- テーブル･エントリー記述子ストレージは、データ・スペース DFHDT001 から割り振られます。割り振られる際に 32 KB ずつ増加します。

  テーブル内のレコードごとにエントリー記述子が 1 つずつあり、さらにキー・シーケンス内のギャップ (CICS 保守データ・テーブルから 1 つ以上のレコードが省かれている場所) ごとにエントリー記述子が 1 つずつあります。各エントリーのサイズは、キー長に 9 バイトを加算し、次の 8 KB の倍数に切り上げた値です。
- 索引ノード・ストレージは、データ・スペース DFHDT002 から割り振られます。割り振られる際に 32 KB ずつ増加します。

  この領域のサイズは、表 1 で示されているキー値の分散とフォーマット、さらに実際のレコード数によって決まります。

表 *1.* キーの分散とフォーマット

キーの分散	キー・フォーマット	レコード当たりのバイト数
高密度 (すべてのキーが連続している)	バイナリー 10 進数 英字	5.1 8.5 19
低密度 (連続しているキーがない)	10 進数 英字	44 51
最悪の事例	-	76

- ECSA ストレージは、データ・テーブルを共用するすべての領域がアクセスする必要がある、いくつかの小さな制御ブロックに使用されます。

ファイルを共用データ・テーブルに変換すると実記憶域の使用量が増える可能性がありますが、更新がほとんどない場合は VSAM LSR バッファー用の実記憶域の使用量が減る可能性があります。また、各 CICS 領域内で読み取り専用テーブルのレプリカを生成して現在高性能を実現しているアプリケーションも、各テーブルの単一コピーを共用してストレージを大幅に節約できる可能性があります。

特定のデータ・テーブルに割り振られたストレージは、そのテーブルが閉じられるまで割り振られたままの状態になります。例えば、データ・テーブルが 1 GB に拡大した後にそのテーブルからすべてのレコードが削除されても、引き続きそのテーブルは 1 GB のデータ・スペース・ストレージを所有しています。所有しているデータ・テーブルを閉じるまで、他のデータ・テーブルはそのストレージを使用できません。

データ・テーブル内のフリー・スペースは追跡され、該当する場合には再利用されます。例えば、テーブル・エントリー記述子や索引ノードが不要になると、同じテーブル内で再利用できるようにフリー・チェーンに追加されます。同様に、レコード・データ・フレーム内のすべてのレコードが削除されると、空のフレームがフリー・チェーンに戻されます。フレーム内の一部のレコードのみ削除された場合に、そのスペースが再利用されるのは、新しいレコードのキーが、偶然同じフレーム (または、そのフレーム内にスペースがない場合は直前のフレーム) 内の別の既存のレコードの直後のキーだった場合だけです。VSAM 制御間隔とは違って、フレーム内のレコードは、記述子によって間接的に配置されるので、必ずしもキー・シーケンスにはなりません。また、統合フリー･スペースでは同時読み取りできないので、レコードを統合フリー･スペースに移動できません。

継続的に増えてほぼ同じシーケンス内で削除されるキーがレコードに割り振られると、通常、スペースは非常に効率的に再利用されます。その理由は、通常新しいレコードでフレームがいっぱいになってから新しいフレームに移行するので、結局古いフレームは完全に空になり、再利用できるようになるからです。また、複数の別個の範囲が、フレーム全体を解放するのに十分な大きさである場合も、スペースが同様に効率的に再利用されます。このような状態では、データ・テーブル・ファイルに割り振られるストレージの量は、使用中のストレージの量に近くなります。

新しいデータ・テーブル・アプリケーションを導入すると、データ・テーブルごとに割り振られるストレージと使用中のストレージをモニターし、現在と将来の使用を十分にサポートできるだけのオペレーティング・システム・リソースが使用可能であることを確認するのに役立つことがあります。割り振られたストレージを読み取る際には、各データ・テーブルで所有されるストレージが示されます。このストレージはデータ・テーブルが削除されるまで解放されません。使用中のストレージを読み取る際には、使用中の割り振られたストレージの量が表示されます。この情報は CICS サンプル統計プログラム DFH0STAT により提供されます。DFH0STAT については、サンプルの統計プログラム DFH0STAT に説明があります。

共用データ・テーブルにより記述子、索引エントリー、またはデータ用のスペースが使い尽くされることがあります。ロード時だけでなく、通常の実行中にレコードが追加される際や、更新される際であっても、スペースが使い尽くされることがあります。現在 CICS は共用データ・テーブルをサポートするために複数のデータ・スペースを使用しているので、全 3 種類のストレージに関する限度は非常に大きくなり、CICS アドレス・スペース内のエントリーがなくなったなどの他の考慮事項とは無関係になります。ただし、使用可能なストレージは依然として有限です。例として、非常に多数の比較的小さなレコードがあり、特にそのほとんどがキー・データで構成されている場合、キー長や他の要因によっては、レコード・データ自体のためのストレージより前にエントリー記述子や索引ノード用のスペースが使い尽くされることがあります。エントリー記述子や索引ノード用のスペースが不足している場合は、データ・テーブルを別個の FOR など、異なる CICS 領域に分割するこ

とを考慮してください。単一のデータ・テーブルが独自のスペースを使い尽くすと、そのデータ・テーブル用のスペースの限度に達するので、複数の別個のテーブルに分割する必要があるか考慮しなければなりません。

共用データ・テーブルを使用する際の MVS JCL 要件

共用データ・テーブルを使用する前に、ご使用の JCL ステートメントへの変更、操作手順の変更、または MAXUSER MVS 初期化パラメーターの値の増加が必要になる場合があります。

これは、ジョブの複数のステップが共用データ・テーブル・サーバーとして動作することを MVS が許可しないからです。2 番目のジョブ・ステップが共用データ・テーブル・サーバーとして動作しようとすると、CICS はメッセージ DFHFC0405 を発行します。さらに、サーバー・ステップの後に続くジョブ・ステップも MRO で仮想記憶間サービスを使用することができないので、サーバー・ステップの後に続くジョブ・ステップがどれも CICS の別の実行にならないようにすることをお勧めします。

共用データ・テーブル・サーバー・ステップを含むジョブが、このサーバーに接続したすべての要求側ジョブ・ステップが終了する前に終わる場合、サーバーのアドレス・スペースが MVS によって終了させられます。共用データ・テーブル・サーバーが、開始タスクではなくバッチ・イニシエーターの制御下で実行されている場合は、前述の状況が発生したときに、新しいイニシエーターを開始しなければなりません。

MVS はメッセージ *IEF355A INITIATOR TERMINATED, RESTART INITIATOR* を出してバッチ・イニシエーターを終了します。これは整合性の理由からであり、これを行わなければ MVS はそのイニシエーターの下で実行される次のジョブで使用される可能性がある機能を制限しなければならなくなって、そのジョブが失敗する原因になる可能性があるからです。MVS は、そのサーバーに接続したすべての要求側ジョブ・ステップが終了するまで、共用データ・テーブル・サーバーの ASID が再使用されることを許可しません。

データ・テーブルとして使用するファイルの選択

ファイルを共用データ・テーブルに変換することで利点が得られるかどうかに関して、正確な規則を定めることはできません。このトピックのチェックリストに、一般的な説明が記載されています。

多数の考慮事項があるので、各種アプリケーションがファイルをどのように使用しているかや、CICS 領域の構成について把握している人が、共用データ・テーブル・サポートの潜在的使用について分析する必要があります。

ファイルの選択に役立つ追加の情報源には以下のものがあります。

- ファイル統計。選択作業に対する入力データの 1 つとして統計情報を使用する方法については、 16 ページの『データ・テーブル選択のために統計を使用する』を参照してください。
- LSR プール統計。
- トレース・エントリー。
- モニター・データ。

しかし、選択プロセスに対する最も役立つ入力は、アプリケーションとそのアプリケーションがファイルを使用する方法について十分に理解していることです。

インストール済み環境で初めてデータ・テーブルを使用する場合は、以下のチェックリストの一般原則を、データ・テーブルとして定義するファイルの選択に役立ててください。

- CICS 保守データ・テーブルが実装しやすいので、最初にこのデータ・テーブルの使用を考慮する必要があります。CICS 保守データ・テーブルを使用する場合、アプリケーションに対する変更は不要です。ユーザー保守データ・テーブルを使用する場合、変更が必要になることがあります。
- CICS リスタートの間データ・テーブルの保全性を保証する必要がある場合は、CICS 保守データ・テーブルを使用します。
- 更新のジャーナリングが必要な場合は、CICS 保守データ・テーブルを使用します。すべてのアクセス要求のジャーナリングが必要な場合、ファイルをデータ・テーブルとして使用することは適切ではありません。
- 仮想記憶間サービスでデータ・テーブルへのアクセス要求が満たされる場合、EXEC インターフェースのユーザー出口である XEIIN と XEIOUT、ファイル制御のユーザー出口である XFCREQ と XFCREQC は、file-owning 領域で呼び出されません。ファイルを選択する際には、アプリケーションの正常な操作が、これらのユーザー出口で実行されるアクティビティーに依存しないことを確認する必要があります。
- 20 ページの『データ・テーブルのセキュリティー検査』で説明されている、データ・テーブルを共用する際のセキュリティーへの影響に注意する必要があります。
- 別の領域からファイルに頻繁にアクセスする場合、多数の他の領域からアクセスする場合、またはアクセスが主に読み取り要求である場合は、そのファイルをデータ・テーブルにする利点は非常に大きくなります。ローカル・ファイルよりリモート・ファイルの方がパフォーマンスの向上が大きいことに留意してください。
- CICS 保守データ・テーブルの場合、データ・テーブルのみにアクセスする要求の比率が適度に高いファイルを選択します (データ・テーブルへのアクセス用の開発を参照)。最大のパフォーマンスを得るには、その中から、このような要求の使用率が最も高いファイルを選択します。

  ファイルの使用状況に関する情報は、DFHSTUP レポートのファイル制御統計にあります。データ・テーブルを活用できない読み取り要求もあるので、後で CICS 統計レポート内のデータ・テーブル情報を調べて、そのデータ・テーブルが効率的に使用されていることを確認する必要があります。詳細については、データ・テーブルのモニターを参照してください。
- ユーザー保守データ・テーブルの場合、更新アクティビティーの比率が多く、CICS リスタート中にその更新のリカバリーの必要がないファイルを選択します (25 ページの『データ保全性』を参照)。
- パフォーマンス測定を使用して、将来の使用予測に留意し、おおよその CPU 使用率の削減を見積もります。

- 最善の見積もりのファイルを 1 つまたは 2 つ選択します。削減の見積もりが同じである場合は、大きなファイルより小さなファイルを優先します。小さなファイルの方が実記憶域の使用量が少ない可能性が高いからです。
- 実記憶使用量をモニターしてください。既にシステムに実記憶の制約がある場合は、大きいデータ・テーブルを使用するとページイン率が増す可能性があります。これにより、CICS システム・パフォーマンスに悪影響を与えます。RMF™ (バージョン 5) など、標準のパフォーマンス・ツールを使用して、実記憶の使用量とページング率を監視してください。
- LSR プール内のバッファー数を減らすことを考慮してください。データ・テーブルを使用すると、LSR プールを使用する回数が減る可能性があるからです。
- ユーザー出口 XDTRD を使用して、データ・テーブル内に含まれるレコードを選択してください。さらに、ユーザー保守データ・テーブルの場合、ユーザー出口 XDTRD を使用してレコードを変更することもできます。つまり、必要なデータのみデータ・テーブルに保管して、仮想記憶域や実記憶域の使用を最適化できます。
- 非常に大きなデータ・テーブルの場合、MVS IEFUSI 出口で設定される普通の領域制限より多くのデータ・スペース・ストレージが必要になることがあります。この場合、IEFUSI 出口を変更して制限を増やすか、CICS XDTRD グローバル・ユーザー出口プログラムを使用して一部のレコードを抑制することができます。IEFUSI 出口については、「*z/OS MVS* 導入システム出口」マニュアル (SA88-8565) を参照してください。

データ・テーブル選択のために統計を使用する

共用が単一 MVS イメージに限定されている場合、共用データ・テーブルを活用するアクセス・パターンをどのファイルが持つかを検討する必要があります。

データを複数の MVS イメージで共用する必要がある場合、RLS モードを使用したファイルの共用について調べるべきです。

17 ページの図 3、 17 ページの図 4、および 18 ページの図 5 に、非 RLS モードでアクセスされるファイルの架空のファイル統計セットからの抜粋を示します。これは、後述する CICS 統計をどのように選択処理に役立てることができるかを示す例で使用します。

統計は、CICS オフライン・フォーマット・ユーティリティーによって報告された場合と同様に表示します。要求されたファイル統計を示していますが、間隔統計または終業時統計も同様に適しています。ファイル・セクションの「パフォーマンス情報」統計は VSAM ストリングとバッファーの使用量を報告しますが、ここには表示していません。

図に示す数値は純粋に図示を目的としています。ご使用のシステムの統計がこれに似ているとは期待しないでください。また、CICS 領域とファイルの構成は、特定のポイントを強調する目的で選択しました。これが典型的な、あるいは望ましい構成だというわけではありません。

データ・テーブルのモニターで、データ・テーブルとして定義されたファイルに関して報告される統計について説明しています。これは、得られる利点を評価するた

めに使用できます。

```
Requested Statistics Report        Collection Date-Time 12/25/99-11:51:51  Last Reset 09:00:00  Applid CICFOR    Jobname SDTGSTF1
FILES - Resource Information
  File      Data Set Name                                Data Set   RLS      DT      Time     Time     Remote    Remote   Lsrpool
  Name          Base Data Set Name (If Applicable)        Type     File  Indicator Opened   Closed     Name     Sysid      ID
  APPLE     CIC01.CICOWN.APPLES                           K       NO              07:44:12  OPEN                          1
  BANANA    CIC01.CICOWN.BANANAS                          K       NO              09:45:08  OPEN                          1
  ORANGE    CIC01.CICOWN.CITRUS                           K       NO              10:51:10  OPEN                          2
  PEAR      CIC01.CICOWN.PEARS                            K       NO              07:30:14  OPEN                          3
Requested Statistics Report        Collection Date-Time 12/25/99-11:51:51  Last Reset 09:00:00  Applid CICFOR    Jobname SDTGSTF1
FILES - Requests Information
  File     Get      Get Upd  Browse   Update   Add      Delete   Brws Upd VSAM EXCP Requests RLS req
  Name     Requests Requests Requests Requests Requests Requests Requests   Data    Index    Timeouts
  APPLE     2317265     1020        0     1019       21        1        0     11503      310        0
  BANANA     536452     1674    20344     1674      908        0        0      2651       70        0
  ORANGE    2069454    98560    17831    98327     4543     2563        0      8511      481        0
  PEAR        45871    65493     6512    65493    30109      362        0      3773      231        0
  *TOTALS*  4969042   166747    44687   166513    35581     2926        0                          0
Requested Statistics Report        Collection Date-Time 12/25/99-11:51:51  Last Reset 09:00:00  Applid CICFOR    Jobname SDTGSTF1
FILES - Data Table Requests Information
  File     Close    Read       Recs ¬   Adds from Add      Adds rejected Adds rejected Rewrite  Delete   Highest    Storage
  Name     Type     Requests   in Table   Reads   Requests    - Exit     - Table Full  Requests Requests Table Size Alloc(K)
DFHST0223 I There are no data table statistics to report.
```

図 *3. CICFOR* の要求されたファイル統計

```
Requested Statistics Report        Collection Date-Time 12/25/99-11:51:38  Last Reset 09:00:00  Applid CICAOR1   Jobname SDTGSTA1
FILES - Resource Information
  File      Data Set Name                                Data Set   RLS      DT      Time     Time     Remote    Remote   Lsrpool
  Name          Base Data Set Name (If Applicable)        Type     File  Indicator Opened   Closed     Name     Sysid      ID
  APPLE     REMOTE                                                          CLOSED   CLOSED   APPLE     CIF1        N
  BANANA    REMOTE                                                          CLOSED   CLOSED   BANANA    CIF1        N
  ORANGE    REMOTE                                                          CLOSED   CLOSED   ORANGE    CIF1        N
  ZUCCHINI  REMOTE                                                          CLOSED   CLOSED   COURGETT  CIA2        N
Requested Statistics Report        Collection Date-Time 12/25/99-11:51:38  Last Reset 09:00:00  Applid CICAOR1   Jobname SDTGSTA1
FILES - Requests Information
  File     Get      Get Upd  Browse   Update   Add      Delete   Brws Upd VSAM EXCP Requests RLS req
  Name     Requests Requests Requests Requests Requests Requests Requests   Data    Index    Timeouts
  APPLE     1158701      532        0      531       11        1        0         0        0        0
  BANANA     305641        0    19067        0        0        0        0         0        0        0
  ORANGE      58709    32854     4265    32621     1018     1001        0         0        0        0
  ZUCCHINI    78914        0    14765        0        0        0        0         0        0        0
  *TOTALS*  1601965    33386    38097    33152     1029     1002        0                          0
Requested Statistics Report        Collection Date-Time 12/25/99-11:51:38  Last Reset 09:00:00  Applid CICAOR1   Jobname SDTGSTA1
FILES - Data Table Requests Information
  File     Close    Read       Recs ¬   Adds from Add      Adds rejected Adds rejected Rewrite  Delete   Highest    Storage
  Name     Type     Requests   in Table   Reads   Requests    - Exit     - Table Full  Requests Requests Table Size Alloc(K)
DFHST0223 I There are no data table statistics to report.
```

図 *4. CICAOR1* の要求されたファイル統計

```
Requested Statistics Report        Collection Date-Time 12/25/99-11:49:31  Last Reset 09:00:00  Applid CICAOR2   Jobname SDTGSTA2
FILES - Resource Information
  File      Data Set Name                                Data Set   RLS      DT      Time     Time     Remote    Remote   Lsrpool
  Name          Base Data Set Name (If Applicable)        Type     File  Indicator Opened   Closed     Name     Sysid      ID
  COURGETT  CIC02.CICOWN.COURGETT                          K       NO              08:22:15  OPEN                           1
  LEMON     REMOTE                                                 NO               CLOSED   CLOSED   ORANGE    CIF1        N
Requested Statistics Report        Collection Date-Time 12/25/99-11:49:31  Last Reset 09:00:00  Applid CICAOR2   Jobname SDTGSTA2
FILES - Requests Information
  File     Get      Get Upd  Browse   Update   Add      Delete   Brws Upd VSAM EXCP Requests RLS req
  Name     Requests Requests Requests Requests Requests Requests Requests   Data    Index    Timeouts
  COURGETT    78914    27469    14765    27469   336472        0        0      8212      481        0
  LEMON     2010745    65706    13566    65706     3525     1562        0         0        0        0
  *TOTALS*  2089659    93175    28331    93175   339997     1562        0                           0
Requested Statistics Report        Collection Date-Time 12/25/99-11:49:31  Last Reset 09:00:00  Applid CICAOR2   Jobname SDTGSTA2
FILES - Data Table Requests Information
  File     Close    Read       Recs ¬   Adds from Add      Adds rejected Adds rejected Rewrite  Delete   Highest    Storage
  Name     Type     Requests   in Table   Reads   Requests    - Exit     - Table Full  Requests Requests Table Size Alloc(K)
DFHST0223 I There are no data table statistics to report.
```

図 *5. CICAOR2* の要求されたファイル統計

このサンプルでは、3 つの CICS 領域からなる架空の構成を使用しています。CICS アプリケーションによって使用されるファイルの大部分はファイル所有領域 CICFOR によって所有されており、アプリケーションの大部分はアプリケーション所有領域 CICAOR1 および CICAOR2 で実行されます。この説明では、統計レポートに示された各データ・セットは VSAM ベース KSDS (`Data Set Type` の `K` で示されている) なので、これらのいずれもデータ・テーブルとして定義することができます。

このセクションは、CICS 保守データ・テーブルとして定義する候補を識別することを中心としています。なぜなら、ユーザー保守データ・テーブルの定義は、一般に、ファイルのパフォーマンスの検討からではなく、特定のアプリケーションの考慮事項から決定されることが多いからです。この点が中心となっているため、示す統計はどれも RLS モードでアクセスされるファイルのものではありません。RLS モードのデータ・セットは CICS 保守データ・テーブルのソースになることができないからです。

また、ここで示す統計では 1 つの領域のどのファイル名が別の領域のファイル名にアクセスするように定義されているかを示します。*Remote Sysid* (リモート・システム識別名) は、2 つの領域間の接続で指定された名前です。このサンプルでは、CICFOR の SYSID は CIF2 であり、CICAOR2 の SYSID は CIA2 です。

更新に対する読み取りの比率が高いファイル

ファイル APPLE は、アプリケーション所有領域 CICAOR1 上で実行しているアプリケーションで使用されます。このファイルは CICAOR1 内でリモート・ファイルとして定義され、このファイル定義は CICFOR に所有されるファイル APPLE を指します。

このファイルを CICFOR 内で CICS 保守データ・テーブルとして再定義すると利点があります。それは、 17 ページの図 4 で示されているように、リモート更新 (追加 11、削除 1、更新 531) に対するリモート読み取り (レポート対象の期間内に 1158701 の `Get Requests`) の比率が高いためです。

統計レポートの「FILES - Requests Information」セクションの意味については、DFHSTUP レポートのファイル制御統計の説明を参照してください。

リモート読み取りの比率が高いファイル

ファイル BANANA は CICFOR で更新されたり読み取られたりしますが、CICAOR1 でもアクセスされます。

リモート・アクセスはすべて更新ではなく読み取りとブラウズなので、BANANA がデータ・テーブルとして定義済みの場合は CICAOR1 内で実行しているアプリケーションには大きな利点がある可能性が高く、ローカル・データ・テーブルから読み取る CICFOR 上のアプリケーションにも利点がある可能性があります。

複数の領域で共用されるファイル

ORANGE が特に適切なデータ・テーブル候補でないように見える場合があります。

17 ページの図 4 の統計は、CICAOR1 からのリモート取り出しの数 (`Get Requests` は 58709、`Browse Requests` は 4265) が比較的少ないことを示しています。しかし、CICAOR2 内のリモート・ファイル LEMON も CICFOR 内の ORANGE を指しているので、CICFOR 内で ORANGE を共用 CICS 保守データ・テーブルとして定義すると、両方の AOR 内のアプリケーションのパフォーマンスにとって利点がある可能性が高くなります。

適した **UMT** 候補

CICAOR2 で所有されるファイル COURGETT は、CICAOR1 内でファイル名 ZUCCHINI によりアクセスされます。

CICAOR1 はこのファイルの読み取りとブラウズのみ行います。更新は所有領域で発行されます。また、これらの更新は当日の CICS の実行だけに関係することがわかっているので、永続的に保存する必要はありません (実際、シャットダウン時に削除されます)。したがって、このファイルはユーザー保守データ・テーブルとして定義するのに適した候補です。その後、VSAM I/O アクティビティーなしですべての更新をデータ・テーブルに対して行うことができ、機能シップなしですべてのリモート取り出しを行うことができます。

あまり適していない候補

ファイル PEAR はおそらく共用データ・テーブル・サポートの利点をあまり得られません。リモートにアクセスされず、多数の更新要求やブラウズ要求があるからです。

ローカル・ブラウズからは、ローカル読み取りやいずれの形式のリモート取り出しよりも利点を得られません。VSAM ブラウズ (STARTBR コマンドの処理を除く) が非常に効率的だからです。もちろんこの分析は、さまざまなファイル・アクセスの相対的な重要度を考慮に入れていません。読み取りは重要なアプリケーションによって行われる一方で、更新に費やされる時間は重要でない場合もあります。

その他の可能な候補

前述の例は、共用データ・テーブル・サポートの利点が得られる可能性があるファイルの構成と使用法のごく小規模なサンプルを図示しています。

各領域内でファイルやデータ・テーブルを重複しなくてもよいように、共用データ・テーブル・サポートを使用することもできます。また、既存のファイルを監視することに加えて、AOR から FOR へのファイルの移動も考慮できます。機能シップ使用時のファイル・アクセスのコストのため、共用データ・テーブルのサポートが使用できない場合、AOR から FOR へのファイルの移動は実際的ではありませんでした。

データ・テーブルのセキュリティー検査

共用データ・テーブルは、LOGON 時や CONNECT 時にセキュリティー検査を実行して、仮想記憶間サービスの使用時にセキュリティーを提供します。データ・テーブルに関連付けられているファイルを共用する前に、セキュリティー検査の影響を考慮する必要があります。

共用データ・テーブルで、以下を確認しなければなりません。

- FOR は偽名を使用できません。LOGON 時に、CICS 領域の指定済みの総称 *applid* を使用して FOR にログオンできるか検査し、偽名の使用を防止します。
- AOR は認識していないと思われるデータにアクセスできません。CONNECT 時に、AOR が FOR にアクセスできるか検査し、さらにファイル・セキュリティーが実施されている場合は AOR が要求されたファイルにアクセスできるか検査してこのアクセスを防止します。

これらのセキュリティー検査は、System Authorization Facility (SAF) を使用して、リソース・アクセス管理機能 (RACF®) か同等のセキュリティー・マネージャーを呼び出して実行します。

注**:** 領域に共用データ・テーブル・サーバーとして動作する権限がない場合でも、その領域でローカルにデータ・テーブルを使用できます。

共用データ・テーブル・サポートには、領域レベルで作動する機能シップ・セキュリティーの主な特性が再現されていますが、以下の違いに注意する必要があります。

- 共用データ・テーブル・サポートは、FOR がトランザクション・レベルでセキュリティー検査を実行するメカニズム (ATTACHSEC(IDENTIFY) や ATTACHSEC(VERIFY) と同等) を提供しません。したがって、AOR によって実行されるトランザクション・レベルの検査が一部のファイルにとって不適切と見なされる場合は、それらのファイルが FOR 内のデータ・テーブルと関連付けられていないことを確認しなければなりません。
- 共用データ・テーブル・サポートは、事前設定セキュリティーをサポートしていません。
- 共用データ・テーブル・サポートは、インストール・パラメーター・リスト (INSTLN) 情報をセキュリティー・ユーザー出口に渡しません。

共用データ・テーブル・セキュリティーの実装に必要なステップの説明については、システム・リソースを保護するための RACF クラスを参照してください。

LOGON セキュリティー検査

LOGON 処理には、指定されたアプリケーション名の FOR がサーバーとして動作するよう許可されているか検証するセキュリティー検査が含まれます。

この検査により、アプリケーション所有領域 (AOR) が、 事実上偽のファイル所有領域 (FOR) から偽のデータ・レコードを受け入れるリスクを最小限にすることができます。システム初期化時に SEC=NO を指定した場合であっても、この検査をバイパスすることはできません。

CONNECT セキュリティー検査

CONNECT 時に実行されるセキュリティー検査により、2 つのレベルのセキュリティーが得られます。

バインド・セキュリティー
: CICS ファイル・セキュリティーがない状態で実行している FOR が、共用アクセスを選択済みの AOR に制限できます。(ファイル・セキュリティーなしで実行すると、実行時のオーバーヘッドとセキュリティー定義の数が最小になります。)

ファイル・セキュリティー
: よりきめ細かいセキュリティー検査が必要な場合に FOR 内で活動化できます。その後、共用データ・テーブル・サポートによりこの種の検査が実装され、全体として AOR に適用されます。

ATTACHSEC(IDENTIFY) または ATTACHSEC(VERIFY) と機能シップを併用する場合、共用データ・テーブル・サポートでは、FOR がトランザクション・レベルで行うセキュリティー検査を実装する方法は提供されません。

共用データ・テーブルのサポートの使用準備

共用データ・テーブルのサポートを使用するには、以下の作業を実行する必要があります。一部の作業は、機能シップやデータ・テーブルを現在使用しているインストールの際に既に実行済みです。

このタスクについて

- 以下のモジュールが MVS システムの LNKLST 内の許可システム・ライブラリー内にあることを確認するか、これらのモジュールを LPALST 連結内のライブラリー内に移動します。
  - DFHDTSVC および DFHDTCV (共用データ・テーブルを使用するすべての領域は、同レベルの SVC コードを使用しなければならないため)
  - DFHMVRMS、つまり RESMGR 出口スタブ (CICS JOBLIB/STEPLIB データ・セットがメモリーの終わりで使用できないため)

  以下のモジュールは、CICS のインストール時にターゲット・ライブラリー SDFHLINK に入れられます。このライブラリーは通常 LNKLST 連結に含まれています。
  - SDFHLINK が LNKLST 連結内にある場合は、モジュールを使用可能にするには、オペレーター・コマンド `MODIFY LLA,REFRESH` を発行し、確認メッセージ `CSV210I LIBRARY LOOKASIDE REFRESHED` を待つ必要があります。
  - SDFHLINK が LNKLST 連結内にない場合は、モジュールをその連結に含まれる適切なライブラリーにコピーし、LLA リフレッシュを発行するか、LPALST 連結内のライブラリーにコピーし、CLPA を指定して MVS システムを再 IPL する必要があります。

- AOR 内のファイルが共用を使用する予定の場合は、CICS が MVS サブシステムとして定義されていることを確認してください。
- 必要なセキュリティーのレベルに応じて、FOR が共用データ・テーブル・サーバーとして動作し、サーバーが所有するファイルに AOR がアクセスできるように、セキュリティー許可を定義します。単一の MVS イメージで、以下のことが可能です。
  - 任意の数の FOR が共用データ・テーブル・サーバーとして動作できます。
  - 単一の AOR で任意の数の FOR を使用できます。
  - 単一の FOR が任意の数の AOR にサービスを提供できます。
  - 1 つの領域が、あるデータ・テーブルにとっては AOR として、別のデータ・テーブルにとっては FOR として機能することが可能です。
- 2 つの FOR に同じ APPLID がある場合、特定の時間には一方の FOR のみが共用データ・テーブル・サーバーとして使用されます。しかし、一方の FOR が共用データ・テーブル・サーバーとして動作し、同じ APPLID の他方の FOR が機能シップされた要求に使用されることは妨げられません。共用データ・テーブル・サービスを使用するデータ・テーブル要求が機能シップを使用する要求と同じ領域に送信されないリスクがあるので、操作手順でこの状況が許可されていないことを確認する必要があります。
- FOR 内のデータ・テーブルにしているファイルを、CICS 保守データ・テーブルまたはユーザー保守データ・テーブルとして定義します。
- 必要に応じて、AOR 内で追加のリモート・ファイル定義を作成します。既存のリモート・ファイル定義に対する変更は不要です。
- データ・テーブルを共用する予定の AOR の場合、システム初期設定パラメーターとして ISC=YES を指定し、関連する FOR への MRO リンクか ISC リンクを定義します。IP 相互接続性 (IPIC) 接続の場合、同等のシステム初期設定パラメーター TCPIP=YES を指定し、関連する FOR への IPIC リンクを定義します。
- 共用データ・テーブルを使用する前に、ご使用の JCL ステートメントへの変更、操作手順の変更、または MAXUSER MVS 初期化パラメーターの値の増加が必要になる場合があります。詳しくは、 14 ページの『共用データ・テーブルを使用する際の MVS JCL 要件』を参照してください。

ロード・モジュール

共用データ・テーブルを使用するには、以下のロード・モジュールを CICS 領域にインストールしなければなりません。

表 2. 共用データ・テーブル・サポートによって使用されるロード・モジュール

ロード・モジュール	ロード・ライブラリー	ロード方法	説明
DFHDTINS	SDFHLOAD	16 MB 境界より上の CICS ロード	初期設定
DFHDTSVC	SDFHLINK	リンク・リストからの 16 MB 境界より上の MVS LOAD	MVS 許可が必要なすべての機能を実行します。
DFHDTFOR	SDFHAUTH	16 MB 境界より上の MVS LOAD	データ・テーブル FOR モジュール

表 2. 共用データ・テーブル・サポートによって使用されるロード・モジュール (続き)

ロード・モジュール	ロード・ライブラリー	ロード方法	説明
DFHDTAM	SDFHAUTH	16 MB 境界より上のサブプール 252 ストレージへの MVS LOAD	データ・テーブル・アクセス・マネージャー。AOR から仮想記憶間モードで実行されるコードが含まれます。
DFHDTAOR	SDFHAUTH	16 MB 境界より上の MVS LOAD	データ・テーブル AOR モジュール
DFHDTCV	SDFHLINK	リンク・リストから ECSA への MVS LOAD	接続の妥当性検査 (AOR)
DFHDTXS	SDFHAUTH	ECSA への MVS LOAD	接続のセキュリティー検査 (FOR)
DFHMVRMS	SDFHLINK	リンク・リストからの 16 MB 境界より上の MVS LOAD	リソース・マネージャー EOT/EOM インターフェース・コード

ストレージ占有:

16 MB 境界より上のストレージを占めるモジュールの合計サイズは、約 41KB です。ECSA ストレージ内のモジュールの場合、ログオンされる FOR ごとに約 1.5KB と AOR ごとに約 0.5KB 必要です。

これらはすべてリンク・パック域 (LPA) に組み込むのに適格なモジュールですが、DFHDTFOR、DFHDTAM、DFHDTAOR、また場合によっては DFHDTCV はかなり頻繁に使用されるのでリンク・パック域へ入れることを考慮してください。

データ・テーブルのリソース定義

データ・テーブルを定義する方法は、CICS ファイルを定義する方法と同じですが、使用するデータ・テーブルのタイプと、データ・テーブル内に保持できるレコードの最大数も指定する必要があることが違います。

VSAM KSDS 定義で、最大レコード長とキー長が提供されます。

26 ページの『**DEFINE FILE** コマンドでデータ・テーブルを定義する』で説明されているように、CEDA DEFINE FILE コマンドを使用してファイルをデータ・テーブルとして定義できます。

さらに、既存のファイルのデータ・テーブル属性を変更したり検査したりするには、以下のコマンドを使用できます。

- EXEC CICS SET FILE コマンドと INQUIRE FILE コマンド (30 ページの『データ・テーブルに関する EXEC CICS コマンド』を参照)。
- CEMT SET FILE コマンドと INQUIRE FILE コマンド (32 ページの『データ・テーブルに関する CEMT コマンド』を参照)。

CICS 保守データ・テーブルのリソース定義

CICS 保守データ・テーブルには、固定長レコードか可変長レコードのどちらかの形式を指定できます。

共用データ・テーブル・サポートでサポートされる最大レコード長は 32KB です。この長さは CICS ファイル管理でサポートされる長さより長いので、実際には制限があります。CICS コマンドに渡される区域の長さを参照してください。サポートされるレコードの最大数は 16,777,215 です。

基本 VSAM クラスターのみが、そのクラスターに基づく CICS 保守データ・テーブルを持つことができます。代替索引パスを介する読み取り要求はデータ・テーブルを使用しませんが、代替索引パスを介するソース・データ・セットへの変更はデータ・テーブルに反映されます。

CICS 保守データ・テーブルのソース・データ・セットを RLS アクセス・モードで開くことはできないので注意してください。したがって、ファイル定義で RLSACCESS(NO) を指定しなければならないので、その他のファイルをすべて同じ基本データ・セットに関連付ける必要があります。

CICS 保守データ・テーブルとして定義されたファイルが開かれた後、その定義内の同じソース・データ・セットの名前を指定しているその他の非 UMT ファイルは (CMT として定義されているかどうかにかかわらず)、自動的に同じデータ・テーブルを使用します。これらの他のファイルのいずれかが CMT として定義されている場合、そのファイルが開かれる際にメッセージ DFHFC0937 がコンソールに対して発行されます。これはエラー状態ではありません。ファイルが開かれ、可能な場合は必ず既存のデータ・テーブルが使用されます。

VSAM SHAREOPTION

DISP=SHR を使用してソース・データ・セットを割り振ると、FOR 以外の領域により更新される可能性があるというリスクが生じます。更新された場合は、データ・テーブルはソース・データ・セットと一致しなくなります。このリスクを最小限にするには、VSAM 領域間 SHAREOPTION を 1 または 2 に設定する必要があります。

- 1 に設定すると、1 つの領域がデータ・セットへの更新アクセス権を持つこと、または多数の領域が読み取り専用アクセス権を持つことができます。
- 2 に設定すると、1 つの領域がデータ・セットへの更新アクセス権を持ち、同時に多数の領域が読み取り専用アクセス権を持つことができます。

DISP の設定にかかわらず、領域間 SHAREOPTION が 3 または 4 の場合や、2 で CICS 保守データ・テーブルに読み取り専用アクセス権がある (つまり別の領域がデータ・セットを更新できない) 場合は、警告メッセージが発行されます。

データ保全性

CICS 保守データ・テーブルを使用するファイルを、リカバリー可能なリソースとして定義できます。システムやトランザクションの障害後に、ソース・データ・セットが通常の方法でリカバリーされます。

- システム障害後、ファイルの再オープン時にデータ・テーブルはリカバリーされたソース・データ・セットから再ロードされます。

- トランザクション障害後、動的トランザクション・バックアウトによりソース・データ・セットに加えられた変更が、データ・テーブルにも加えられます。

ソース・データ・セットにアクセスするファイル操作に関する自動ジャーナリングが (その他のファイルと同じ方法で) サポートされています。ソース・データ・セットにアクセスしないファイル操作はジャーナル処理されません。

ユーザー保守データ・テーブルのリソース定義

ユーザー保守データ・テーブルには、可変長レコード形式を指定しなければなりません。

共用データ・テーブル・サポートでサポートされる最大レコード長は 32KB です。この長さは CICS ファイル管理でサポートされる長さより長いため、実際には制限があります。CICS コマンドに渡される区域の長さを参照してください。サポートされるレコードの最大数は 16 777 215 です。

ユーザー保守データ・テーブルのソース・データ・セットを RLS アクセス・モードで開くことができます。RLS モードのデータ・セットにアクセスするその他のファイル定義がある場合や、このデータ・セットがその他の CICS 領域で更新される場合は、このデータ・セットをユーザー保守データ・テーブルのソースにすることもできます。

別々のコマンドかマクロを使用して各データ・テーブルを定義し、すべての定義が同じソース・データ・セットを参照するようにして、そのデータ・セットから複数のユーザー保守データ・テーブルをロードできます。

データ・テーブルは VSAM KSDS からロードしなければなりませんが、その後アプリケーションにより、CICS アドレス・スペースからアクセスできるデータ・ソースから、ユーザー保守データ・テーブルにレコードをコピーできます。このデータ・ソースは、IMS™ ファイルか Db2® ファイルです。データ・テーブルのソース・データ・セットとして使用する KSDS は空でもかまいません。最大レコード長と、キーの長さと位置を定義する必要のみあります。

データ保全性

ユーザー保守データ・テーブルを、リカバリー可能なリソースとして定義できます。データ・テーブルへの変更は、システム・ログに記録されませんが、CICS メモリー内に内部的に保持されます。したがって、トランザクション障害後に (動的バックアウトにより) データ・テーブルをリカバリーできますが、システム障害後はリカバリーできません。

これは、CICS 共有データ・テーブル機能では独自のリカバリーを管理し、ログ・マネージャーやリカバリー・マネージャーのサービスを使用しないためです。ただし、分散作業単位の一部としてリカバリー可能データ・テーブルに変更を加える場合は例外です。この場合、その他のリカバリー可能リソースと同様に、リンクのレコードが 2 フェーズ・コミット・プロセスの一部としてシステム・ログに書き込まれます。しかし、変更自体はシステム・ログに記録されません。

システム障害後、ファイルの再オープン時にデータ・テーブルはソース・データ・セットから再ロードされます。以下のことを確認していない場合は、障害発生時に、ソース・データ・セットのコンテンツとデータ・テーブルのコンテンツが同じでない可能性に留意してください。

- どちらにも変更が加えられていない。
- または、両方に変更が加えられている。

ロード時にソース・データ・セットにアクセスする要求の場合のみ、自動ジャーナリングがサポートされます。ロード・プロセスでアクセスされるレコードはユーザー出口 XDTRD の前にジャーナル処理され、アプリケーション要求のためにアクセスされるレコードはユーザー出口 XDTRD の後にジャーナル処理されます。

DEFINE FILE コマンドでデータ・テーブルを定義する

ファイルを CICS 保守データ・テーブルまたはユーザー保守データ・テーブルのいずれかとして定義するには、**DEFINE FILE** コマンドを使用します。

FILE 定義の詳細については、 FILE リソースを参照してください。このトピックでは、データ・テーブルに関係する属性のみ記述されています。

TABLE({NO|CICS|USER|CF})

ファイルを CICS 保守データ・テーブルとして定義するには、**TABLE(CICS)** を指定します。

ファイルをユーザー保守データ・テーブルとして定義するには、**TABLE(USER)** を指定します。

TABLE パラメーターを指定しない場合や、**TABLE(NO)** か **TABLE(CF)** を指定する場合は、ファイルは CICS 共用データ・テーブルとして定義されません。

MAXNUMRECS(NOLIMIT|number)

データ・テーブルに格納できる最大レコード数を指定します。範囲は 1 から 99999999 です。デフォルトでは、最大レコード数に制限がありません。

FILE(name)

ファイルの名前を指定します。

CICS 保守データ・テーブルの場合、この名前を使用して、データ・テーブルとソース・データ・セットの両方を参照します。CICS ではこの両方が単一のエンティティーとして扱われます。

ユーザー保守データ・テーブルの場合、この名前を使用して、データ・テーブルのみ参照します。

DSNAME(name)

ソース・データとして使用される VSAM KSDS の名前を指定します。パスではなく基本データ・セットか、代替索引データ・セットでなければなりません。ソース・データ・セットに関連付けられているパスまたは代替索引がある場合、ファイルを介した CICS 保守データ・テーブルの更新は、ソース・データ・セットとその代替索引の両方に反映されます。ユーザー保守データ・テーブルの場合、更新はソース・データ・セットとその代替索引のどちらにおいても反映されません。ロードが完了した後、ユーザー保守データ・テーブルは、そのソース・データ・セットから完全に独立しています。

LSRPOOLID(number|1)
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

LSRPOOLNUM(number|1|NONE)
データ・テーブルで使用される VSAM ローカル共用リソース (LSR) プールの数を指定します。LSRPOOL 数は 1 から 255 の範囲で指定しなければなりません。NSRGROUP 属性の値が指定されている場合以外は、デフォルト値は 1 です。NSRGROUP 属性に値が指定されている場合、LSRPOOLNUM のデフォルト値は NONE です。

OPENTIME({FIRSTREF|STARTUP})
自動的に開始されるトランザクション CSFU によりファイルが開かれる時点を指定します。最初の参照時か、始動直後のいずれかです。デフォルトでは、**OPENTIME(FIRSTREF)** が想定されます。

データ・テーブルはファイルが開かれる時点でロードされるので、ユーザー出口 XDTRD を使用する場合は、ファイルが開かれる前にこのユーザー出口を活動化していることを確認してください (データ・テーブル用ユーザー出口の活動化を参照)。

RECORDFORMAT({V|F})
ファイル内のレコードの形式を指定します。可変長レコードの場合は **RECORDFORMAT(V)** で、固定長レコードの場合は **RECORDFORMAT(F)** です。

デフォルトでは、**RECORDFORMAT(V)** が想定されます。ユーザー保守データ・テーブルの場合は、可変長レコードでなければなりません。

ADD(NO|YES)、**BROWSE(NO|YES)**、**DELETE(NO|YES)**、 **READ(YES|NO)**、**UPDATE(NO|YES)**
データ・テーブルに関する要求可能なファイル操作を指定します。

RECOVERY({NONE|BACKOUTONLY|ALL})
データ・テーブルにとって必要なリカバリー・サポートのタイプを指定します。デフォルトは **RECOVERY(NONE)** です。

ユーザー保守データ・テーブルの場合、CICS では動的トランザクション・バックアウトのみサポートされているので、**RECOVERY(BACKOUTONLY)** と **RECOVERY(ALL)** は同じ意味になります。

CICS 保守データ・テーブルの場合、RECOVERY パラメーターはソース・データ・セットに適用されます。この値は、同じデータ・セットに関するその他のファイル定義と整合していなければなりません。

ユーザー保守データ・テーブルのリカバリー属性は、そのソース・データ・セットのリカバリー属性とは無関係です。

ユーザー保守データ・テーブルを定義する際には、ファイル定義上でそのリカバリー属性を指定します。リカバリー不能にする場合は **RECOVERY(NONE)** を指定し、トランザクション障害後にリカバリー可能にする場合は **RECOVERY(BACKOUTONLY|ALL)** を指定します。

ユーザー保守データ・テーブルの指定内容にかかわらず、ユーザー保守データ・テーブルのソース・データ・セットをリカバリー不能にしたり、バックアウト (トランザクションとシステムの両方の障害後) に限りリカバリー可能にしたり、順方向リカバリー可能にしたりすることができます。

ソース・データ・セットは、以下の 2 つの方法のいずれかでそのリカバリー属性を獲得できます。

- ICF カタログ内で定義されているデータ・セットのリカバリー属性を獲得します (CICS Transaction Server for z/OS, バージョン 5 リリース 5 で、RLS モードと非 RLS モードの両方のファイルで可能)。
- 別のファイル名を使用し、ファイル定義にリカバリー属性が指定されているデータ・セットに通常の CICS ファイルとしてアクセスして獲得します (CICS Transaction Server for z/OS, バージョン 5 リリース 5 で、非 RLS モードのファイルのみ可能)。

CICS 保守データ・テーブル定義の例

この例は、CICS 保守データ・テーブルの定義を示しています。これは、必要なパラメーターだけを示したものです。

```
 File         ==> APPLE
 Group        ==> FRUIT
 DEscription  ==>
VSAM PARAMETERS
 DSNAme       ==> CIC01.CICOWN.APPLES
 Password       :                    PASSWORD NOT SPECIFIED
 RLSACCESS    ==> NO                 YES|NO
 LSRPOOLId    ==> 1                  1-8 | None
 LSRPOOLNum   ==> 002                1-255 | None
 READINTEG    ==> UNCOMMITTED        UNCOMMITTED|CONSISTENT|REPEATABLE
 DSNSharing   ==> Allreqs            Allreqs | Modifyreqs
 STRings      ==> 005                1 - 255
 Nsrgroup     ==>
REMOTE ATTRIBUTES
 REMOTESystem ==>
 REMOTEName   ==>
REMOTE AND CFDATATABLE PARAMETERS
 RECORDSize   ==> 00080              1-32767
 Keylength    ==> 006                1-255 (1-16 For CF Datatable)
INITIAL STATUS
 STAtus       ==> Enabled            Enabled | Disabled | Unenabled
 Opentime     ==> Startup            Firstref | Startup
 DIsposition  ==> Share              Share | Old
BUFFERS
 DAtabuffers  ==> 00002              2 - 32767
 Indexbuffers ==> 00001              1 - 32767
DATATABLE PARAMETERS
 TABLE        ==> CICS               No | Cics | User | CF
 Maxnumrecs   ==> 1000000            Nolimit  | 1-99999999
CFDATATABLE PARAMETERS
 Cfdtpool     ==>
 TABLEName    ==>
 UPDATEModel  ==> Locking            Contention | Locking
 LOad         ==> No                 No | Yes
DATA FORMAT
 RECORDFormat ==> F                  V | F
OPERATIONS
 Add          ==> Yes                No | Yes
 BRowse       ==> No                 No | Yes
 DELete       ==> Yes                No | Yes
 READ         ==> Yes                Yes | No
 Update       ==> Yes                No | Yes
AUTO JOURNALING
 JOurnal      ==> No                 No | 1 - 99
 JNLRead      ==> None               None | Updateonly | Readonly | All
 JNLSYNCRead  ==> No                 No | Yes
 JNLUpdate    ==> No                 No | Yes
 JNLAdd       ==> None               None | Before | AFter |ALl
 JNLSYNCWrite ==> Yes                Yes | No
RECOVERY PARAMETERS
 RECOVery     ==> All                None | Backoutonly | All
 Fwdrecovlog  ==> 10                 No | 1-99
 BAckuptype   ==> STAtic             STAtic | DYNamic
セキュリティー
 RESsecnum      : 00                 0-24 | Public
```

ユーザー保守データ・テーブル定義の例

この例は、ユーザー保守データ・テーブルの定義を示しています。これは、必要なパラメーターだけを示したものです。

```
 File         ==> COURGETT
 Group        ==> VEGS
 DEscription  ==>
VSAM PARAMETERS
 DSNAme       ==> CIC02.CICOWN.COURGETT
 Password       :                    PASSWORD NOT SPECIFIED
 RLSACCESS    ==> NO                 YES|NO
 LSRPOOLId    ==> 1                  1-8 | None
 LSRPOOLNum   ==> 002                1-255 | None
 READINTEG    ==> UNCOMMITTED        UNCOMMITTED|CONSISTENT|REPEATABLE
 DSNSharing   ==> Allreqs            Allreqs | Modifyreqs
 STRings      ==> 005                1 - 255
 Nsrgroup     ==>
REMOTE ATTRIBUTES
 REMOTESystem ==>
 REMOTEName   ==>
REMOTE AND CFDATATABLE PARAMETERS
 RECORDSize   ==> 00080              1-32767
 Keylength    ==> 006                1-255 (1-16 For CF Datatable)
INITIAL STATUS
 STAtus       ==> Enabled            Enabled | Disabled | Unenabled
 Opentime     ==> Firstref           Firstref | Startup
 DIsposition  ==> Share              Share | Old
BUFFERS
 DAtabuffers  ==> 00002              2 - 32767
 Indexbuffers ==> 00001              1 - 32767
DATATABLE PARAMETERS
 TABLE        ==> User               No | CIcs | User | CF
 Maxnumrecs   ==> 2000000            Nolimit | 1-99999999
CFDATATABLE PARAMETERS
 Cfdtpool     ==>
 TABLEName    ==>
 UPDATEModel  ==> Locking            Contention | Locking
 LOad         ==> No                 No | Yes
DATA FORMAT
 RECORDFormat ==> V                  V | F
OPERATIONS
 Add          ==> Yes                No | Yes
 BRowse       ==> Yes                No | Yes
 DELete       ==> No                 No | Yes
 REAd         ==> Yes                Yes | No
 Update       ==> Yes                No | Yes
AUTO JOURNALING
 JOurnal      ==> No                 No | 1 - 99
 JNLRead      ==> None               None | Updateonly | Readonly | All
 JNLSYNCRead  ==> No                 No | Yes
 JNLUpdate    ==> No                 No | Yes
 JNLAdd       ==> None               None | Before | AFter |ALl
 JNLSYNCWrite ==> Yes                Yes | No
RECOVERY PARAMETERS
 RECOVery     ==> Backoutonly        None | Backoutonly | All
 Fwdrecovlog  ==> No                 No | 1-99
 BAckuptype   ==> STAtic             STAtic | DYNamic
セキュリティー
 RESsecnum      : 00                 0-24 | Public
```

データ・テーブルに関する EXEC CICS コマンド

EXEC CICS SET FILE コマンドを使用して既存のファイルの定義を変更したり、**EXEC CICS INQUIRE FILE** コマンドを使用して既存のファイルの定義を検査したりできます。

これらのコマンドや、ここで説明されているパラメーターの使用法の詳細などのプログラミング情報については、SET FILE を参照してください。 データ・テーブルに関係するパラメーターについて、以下で説明されています。

このセクションには、汎用プログラミング・インターフェースと関連ガイダンス情報が含まれています。

SET FILE

以下のパラメーターはデータ・テーブルに関係しています。これらのパラメーターは、ファイルが閉じていて使用不可になっている場合に限り使用できます。

CICS 値データ域 (cvda) 内のファイルのデータ・テーブル属性を指定できます。

TABLE(cvda)

ファイルを CICS 保守データ・テーブルとして定義するには、**CICSTABLE** の CVDA 値を指定します。

ファイルをユーザー保守データ・テーブルとして定義するには、**USERTABLE** の CVDA 値を指定します。

ファイルがデータ・テーブルでないことを示すには、**NOTTABLE** の CVDA 値を指定します。

注**:** CFTABLE を指定してカップリング・ファシリティー・データ・テーブルを示すこともできます。

MAXNUMRECS(value)

データ・テーブルに格納できる最大レコード数を指定します。範囲は 1 から 99999999 です。値ゼロは限度がないことを意味します。

INQUIRE FILE

データ・テーブルに関係するパラメーターは以下のとおりです。

以下のように指定して、ファイルの各データ・テーブル属性が CICS 値データ域 (cvda) 内に返されるように要求できます。

TABLE(cvda)

値 **CICSTABLE** が返された場合、ファイルは CICS 保守データ・テーブルとして定義されています。

値 **USERTABLE** が返された場合、ファイルはユーザー保守データ・テーブルとして定義されています。

値 **CFTABLE** が返された場合、ファイルはカップリング・ファシリティー・データ・テーブルとして定義されています。

値 **NOTTABLE** が返された場合、現在ファイルはデータ・テーブルとして定義されていません。

値 **NOTAPPLIC** が返された場合、ファイルがリモート・ファイルなのでオプションを適用できません。

MAXNUMRECS(cvda)

返される値は、データ・テーブルに格納できる最大レコード数を指定します。値ゼロは限度がないことを意味します。

データ・テーブルに関する CEMT コマンド

CEMT SET FILE コマンドを使用して既存のファイルの定義を変更したり、CEMT INQUIRE FILE コマンドを使用して既存のファイルの定義を検査したりできます。

ここで説明されているパラメーターを含む、これらのコマンドの使用法の詳細については、INQUIRE FILE を参照してください。 データ・テーブルに関係するパラメーターについて、以下で説明されています。

SET FILE

以下のパラメーターはデータ・テーブルに関係しています。これらのパラメーターは、ファイルが閉じていて使用不可になっている場合に限り使用できます。

{CICSTABLE|USERTABLE|CFTABLE|NOTTABLE}

ファイルを CICS 保守データ・テーブルとして定義するには、**CICSTABLE** を指定します。

ファイルをユーザー保守データ・テーブルとして定義するには、**USERTABLE** を指定します。

注**:** CFTABLE を指定してカップリング・ファシリティー・データ・テーブルを示すこともできます。

ファイルがデータ・テーブルでないことを示すには、**NOTTABLE** を指定します。

MAXNUMRECS(value)

データ・テーブルに格納できる最大レコード数を指定します。範囲は、1 から 99999999 です。値ゼロは限度がないことを意味します。

INQUIRE FILE

データ・テーブルに関係するパラメーターは以下のとおりです。

Data table

値 **CICSTABLE** が返された場合、ファイルは CICS 保守データ・テーブルとして定義されています。

値 **USERTABLE** が返された場合、ファイルはユーザー保守データ・テーブルとして定義されています。

値 **CFTABLE** が返された場合、ファイルはカップリング・ファシリティー・データ・テーブルとして定義されています。

値 **NOTTABLE** が返された場合、現在ファイルはデータ・テーブルとして定義されていません。

MAXNUMRECS(value)

返される値は、データ・テーブルに格納できる最大レコード数を指定します。値ゼロは、最大数の限度がないことを意味します。

第 3 章 データ・テーブルへのアクセスのための開発

通常の CICS ファイルとともに使用する場合と同じ EXEC CICS ファイル制御コマンドを使用して、データ・テーブルにアクセスします。これらのコマンドは、CICS 保守データ・テーブルと、一部の制約事項はありますが、ユーザー保守データ・テーブルを使用して駆使することができます。

これらのコマンドの使用法に関する一般情報については、分散プログラム・リンク機能の使用を参照してください。プログラミング情報については、CICS コマンド・サマリーを参照してください。

CICS 保守データ・テーブル用のアプリケーション・プログラミング

CICS では、CICS 保守データ・テーブルとそのソース・データ・セットを単一エンティティーとして処理します。データ・テーブルがロードされた後、CICS では自動的にデータ・テーブルとソース・データ・セットの内容の整合性が保たれ、アプリケーションがファイルに対して行う変更はどちらにも反映されます。ほとんどすべての状況で、データ・テーブルの使用については、アプリケーション・プログラマーに対して透過的です。

CICS 保守データ・テーブルには、すべてのファイル制御コマンドおよびオプションを使用できます。データ・テーブルにのみアクセスすることによって (共用ファイルに仮想記憶間サービスを使用) 実行されるコマンドや、ソース・データ・セットにのみアクセスすることによって (共用ファイルに機能シップを使用) 実行されるもの、その両方へのアクセスによって実行されるものがあります。

以下のコマンドは通常、データ・テーブルにのみアクセスします。

- UPDATE または RBA オプションを指定しない READ コマンド
- RBA オプションを指定しない STARTBR、RESETBR、READNEXT、および READPREV コマンド
- ENDBR コマンド (ブラウズ・シーケンスがソース・データ・セットにアクセスしている場合を除く)

以下のコマンドは、ソース・データ・セットにのみアクセスします。

- UPDATE または RBA オプションを指定した READ コマンド
- RBA オプションを指定した STARTBR、RESETBR、READNEXT、および READPREV コマンド
- ソース・データ・セットにアクセスしているブラウズ・シーケンスの ENDBR コマンド

以下のコマンドは、データ・テーブルとソース・データ・セットの両方にアクセスできます。

- データ・テーブル内のレコードのキー・シーケンスでギャップを検出する READ およびブラウズ・コマンド (通常はデータ・テーブルにのみアクセスするコマンド)。このギャップは、以下の理由で、データ・テーブルから 1 つ以上のレコードが見つからないことを示すことがあります。

© Copyright IBM Corp. 2005, 2018

- レコードがユーザー出口によって抑制されている
- レコードの最大数に達している
- データ・テーブルに使用できる仮想記憶域が不足している
- 何か異常なイベントが発生している

- WRITE、REWRITE、または DELETE コマンドによって現在処理中のレコードに対する READ、READNEXT、および READPREV コマンド。これらのコマンドは、最初にデータ・テーブルにアクセスして、この状態が生じているか判断する必要があります。
- WRITE、REWRITE、および DELETE コマンド。これらのコマンドは常に FOR で実行され、そこで最初にソース・データ・セットを更新します。これが成功した場合は、FOR のローカル共用データ・テーブル・サービスを使用して、データ・テーブルに対応する変更が試行されます。 WRITE コマンドの場合は、データ・テーブルへのレコードの追加が XDTAD ユーザー出口によって拒否されたり、データ・テーブルがいっぱいか、使用可能な仮想記憶域の不足によって失敗したりすることがあります。

CICS 保守データ・テーブルの総称読み取り

READ コマンドで GENERIC オプションを使用して、総称読み取りを実行するアプリケーションの場合、VSAM ファイルと比較される CICS 保守データ・テーブルの動作には差異があります。 VSAM ファイルを CICS 保守データ・テーブルに変換する場合、これらのアプリケーションの変更が必要な場合があります。

VSAM ファイルの総称読み取りでは、テーブルでレコードが見つからないために CICS が NOTFND 状態を返す場合、READ コマンドからの INTO() および RIDFLD() 領域は変更のないままです。ただし、CICS 保守データ・テーブルの総称読み取りでは、CICS が NOTFND 状態を返す場合、CICS は、INTO() および RIDFLD() 領域をクリアして、誤ったレコードを返さないようにします。

この動作により、CICS 保守データ・テーブルのパフォーマンスが最適化されますが、これは、返される INTO() および RIDFLD() 領域の元の値にアプリケーションが依存しなくなるということを意味します。総称読み取りを実行するアプリケーションを使用する場合は、NOTFND 状態が返され、さらに INTO() および RIDFLD() 領域がクリアされているという条件で適切なアクションを実行するよう、必要に応じてアプリケーションを変更します。

ロード中の CICS 保守データ・テーブルの使用

CICS 保守データ・テーブルは、ロード中に使用することができます。必要なレコードが既にロードされている場合は、通常の方法で要求の処理が行われます。

レコードがまだロードされていない場合は、以下を実行します。

- READ コマンドの場合は、レコードがソース・データ・セットから読み取られ、アプリケーション・プログラムに返されます。通常のロード順序でレコードに到達すると、そのレコードがデータ・テーブルに追加されます。
- WRITE コマンドの場合、レコードがソース・データ・セットとデータ・テーブルに追加されます (ユーザー出口 XDTAD によって抑制されていない場合)。

- REWRITE または DELETE コマンドの場合、ソース・データ・セットに変更が適用されます。次に、通常のロード・プロセスによって、この変更がデータ・テーブルに反映されます。

ユーザー保守データ・テーブル用のアプリケーション・プログラミング

CICS では、ユーザー保守データ・テーブルとそのソース・データ・セットを別個のエンティティーとして処理します。ロードが完了すると、ファイル名にアクセスするすべてのファイル制御コマンドが、データ・テーブル上でのみ実行されます。

ユーザー保守データ・テーブルからの要求を満たせない場合、CICS はソース・データ・セット (本来は CICS 保守データ・テーブル用) にアクセスしません。 CICS は、代わりに例外条件の応答を返します。

データ・テーブル・サービスでユーザー出口を使用して、アクセスに必要なレコードのみをデータ・テーブルに配置できます。したがって、ロードしないソース・データ・セットについては、アクセスされる可能性がありません。また、ユーザー出口 XDTRD を使用して、各レコードをロードするときに (例えば、そのフィールドのサブセットのみを選択するなどして) 変更できます。

データ・テーブルを開いたときにソース・データ・セットに存在したレコードは、ロード中にコピーされないため、データ・テーブルに存在しない場合があります。これは、ユーザー出口 XDTRD による抑制や、データ・テーブルがいっぱいになるなどの異常なイベントが原因になっている場合があります。

一部のアプリケーション・プログラミングの要求は、ユーザー保守データ・テーブルでサポートされません。例えば、不正確なキーを指定して UPDATE オプションを使用する読み取り要求はサポートされません。また、一部の例外条件は、ユーザー保守データ・テーブルに固有です。使用可能なコマンドとオプションに関する制約事項に従うように、または CICS が返す例外条件を処理するように、既存のアプリケーションの変更が必要な場合があります。

以下のコマンドはサポートされません。これらのコマンドは、INVREQ 状態および EIBRESP2 フィールドに 44 という値を返します。

- RBA オプションを指定したコマンド
- MASSINSERT オプションを指定した WRITE コマンド

以下のコマンドはサポートされます (リモート・アクセスに仮想記憶間サービスを使用)。

- RBA オプションまたは UPDATE オプションを指定しない READ コマンド。データ・テーブルにレコードが存在しない場合、NOTFND 状態が返されます。
- RBA オプションを指定しない STARTBR、RESETBR、READNEXT、および READPREV コマンド。
- ENDBR コマンド。

以下のコマンドはサポートされます (リモート要求に機能シップを使用)。

- RBA または MASSINSERT オプションを指定しない WRITE コマンド。データ・テーブルにレコードが追加されます (XDTAD ユーザー出口で抑制されていない場合)。

以下の状況において、NOSPACE 状態が返されます。

– データ・テーブルにレコードを追加するための十分なデータ・スペース・ストレージがない。
– ファイル定義で指定された最大数のレコードが、既にデータ・テーブルに含まれている。

書き込みの試行に対して使用可能なスペースが不足している場合、CICS はメッセージ DFHFC0432 を出します。

ユーザー出口 XDTAD でデータ・テーブルへのレコードの追加を抑制する場合は、SUPPRESSED 状態が返されます。

- RBA オプションを指定しない REWRITE コマンド。レコードがデータ・テーブルで更新されます。更新されるレコードに対して仮想記憶域が十分にない場合は、NOSPACE 状態が返されます。使用可能なスペースが十分にない場合は、CICS によってメッセージ DFHFC0432 が発行されます。
- RBA オプションを指定しない DELETE コマンド。データ・テーブルからレコードが削除されます。データ・テーブルにレコードが存在しない場合は、NOTFND 状態が返されます。データ・テーブルがリカバリー可能であるが、削除されたデータに関して CICS が書き込む情報に対して十分な仮想記憶域がない場合は、NOSPACE 状態が返されます。

ロード中のユーザー保守データ・テーブルの使用

ユーザー保守データ・テーブルは、ロード中、FOR によってのみアクセスできます。すべてのリモート要求は、FOR に機能シップされ、FOR によってローカル要求の場合と同じ方法で処理されます。

ユーザー保守データ・テーブルのロード中は、正確なキーを指定した非更新読み取り要求しか使用できません。レコードが既にロードされている場合は、通常の方法で要求の処理が行われます。レコードがまだロードされていない場合は、レコードがソース・データ・セットから読み取られ、ユーザー出口 XDTRD (活動化されている場合) に実行依頼されます。

- レコードは、XDTRD によって抑制されていない場合は、レコードがデータ・テーブルに追加され、アプリケーション・プログラムに返されます。
- XDTRD によって抑制されている場合は、NOTFND 状態が返されます。

ロードが完了していれば有効になる、その他の要求に対しては、LOADING 状態が返されます。

共用データ・テーブルの仮想記憶間サービスの使用

仮想記憶間サービスは、ここでリストするすべての条件を満たした場合に、アプリケーション・プログラミング・コマンドを満足させるために使用されます。

- CICS では、AOR のファイルのリソース定義からターゲット・システムの SYSID を取得する必要があります。アプリケーション・プログラミング・コマンドで明示的な SYSID を指定していない場合、または AOR 自体と同じ SYSID を指定していて、ファイル・リソース定義に指定された SYSID が FOR と同じである場合、この条件が満たされます。

単一ブラウズ・シーケンス内では、アプリケーションで明示的な SYSID を指定している場合も、指定していない場合も、その指定は変更しないでください。変更すると予期せぬ結果を招く可能性があるためです。

- サービス提供システムがログオンしている。つまり、共用データ・テーブルの所有者としてそのシステム自体が登録されている。
- 要求側システムが、アプリケーション・プログラミング・コマンドで指定されたファイルのサーバーに接続されている。
- ファイルが、要求された機能をサポートしている。

注: 要求を機能シップすると、「デイジー・チェーン」になる場合があります。つまり、この要求は、要求を発行する領域 (AOR) とリソースを所有する領域 (FOR) 間で 1 つ以上の中間 CICS ノードを通ります。このような場合、共用データ・テーブルの仮想記憶間サービスの使用は、(最後の中間システムから FOR までの) 最後のリンクに制限されます。

接続

コマンドでは、AOR とリモート・データ・テーブル間の接続が確立するまで、仮想記憶間サービスを使用できません。

また、接続が確立される前にブラウズ・シーケンスが開始された場合、シーケンス内の以降のすべての要求で機能シップ・サービスが使用されます。データ・テーブルが開いていないため、STARTBR コマンドで接続が確立できない場合に、このようになる可能性があり、このコマンドによって、データ・テーブルが暗黙的に開きます。データ・テーブルへの次の新しい要求で接続が確立されますが、元のブラウズ・シーケンスでは、引き続き機能シップ・サービスが使用されます。

切断

接続が確立すると、AOR がそのリモート・ファイル定義を削除する、または FOR がファイルを閉じるか使用不可にするまで、その接続は有効な状態で続きます。

閉じたり使用不可にしたりするときの影響は以下のとおりです。

- FOR でファイルを閉じる場合 (FORCE オプション指定の有無に関わらず)、そのファイルに対して発行される、次の非更新要求時に (つまり次回、データ・テーブルへのアクセスのために仮想記憶間サービスの使用要求を行うときに)、切断がスケジュールされます。

  ファイルに対して未解決のすべてのブラウズ・シーケンス (存在する場合) が終了すると、すぐに切断が実行されます。各ブラウズ・シーケンスは、次回ブラウズ要求時 (その要求が ENDBR コマンドである場合を除き、トランザクションはコード AFCH で異常終了する)、およびトランザクションの終了時に、終了します。

  切断がスケジュールされた後、すべての要求 (前述のように、未解決のブラウズ要求を除く) は、接続が再確立されるまで機能シップされます。
- FOR で FORCE オプションを指定せずにファイルを使用不可にしている場合は、先に FOR でファイルを再度使用可能にしておかない限り、そのファイルに対する次の非更新 READ コマンドまたは STARTBR コマンドの発行時に切断がスケジュールされます。

スケジュールされた場合は、ファイルに対して未解決のすべてのブラウズ・シーケンス (存在する場合) が終了すると、すぐに切断が実行されます。このようなブラウズ・シーケンスは正常に継続されます。ブラウズ・シーケンスの要求を満たすためにソース・データ・セットのブラウズが FOR で開始されない限り、使用不可にすることによる影響はありません。

- FOR で FORCE オプションを指定してファイルを使用不可にした場合、その影響はファイルを閉じたときと同じです。ただし、そのファイルに対して AOR が次の非更新要求を発行する前に、FOR でファイルを再度使用可能にしていれば、AOR では使用不可と認識されないため、切断がスケジュールされることはありません。

機能シップと仮想記憶間サービスの相違点

データ・テーブルへのアクセスに機能シップを使用するか、仮想記憶間サービスを使用するかによって、要求を処理する方法にはいくつかの相違点があります。

データ・テーブルのクローズ

リモート・ファイルのブラウズ・シーケンスに機能シップを使用する場合は、ブラウズ・シーケンスが終了した後まで、ファイルを閉じることはできません (FORCE オプションを使用する場合を除く)。

仮想記憶間サービスを使用する場合は、ブラウズ・シーケンスの途中でファイルを閉じることができます。この場合、トランザクションは、そのファイルの次の要求で、異常終了コード AFCH で終了します。アプリケーションまたは操作手順が、ファイルを閉じる際または FOR の正常なシャットダウンのときのブラウズ・アクティビティーの静止に依存する場合は、ファイルの共用データ・テーブルを使用する前に確認する必要があります。

データ・テーブルを使用不可にする

リモート・ファイルのブラウズ・シーケンスに機能シップを使用する場合は、ブラウズ・シーケンスが一度開始されると、ファイルが使用不可になっていても正常に続行されます (FORCE オプションを使用している場合を除く)。

仮想記憶間サービスを使用する場合、ブラウズ・シーケンスの途中で STARTBR コマンドを FOR に機能シップする必要がある場合を除き、影響は同じです。例えば、CICS 保守データ・テーブルのギャップにより、VSAM ソース・データ・セットをブラウズしてレコードを取得することが必要になるなどの場合に、この状況になります。ブラウズ・シーケンスが AOR で開始された後に発行された要求によって、ファイルが使用不可になった場合、機能シップされた STARTBR コマンドは失敗します。この場合、ブラウズ・シーケンスは正常に続行できないため、AOR のトランザクションがコード AFCH で異常終了します。

使用不可にする要求で FORCE オプションを使用した場合は、機能シップされたすべてブラウズ要求がいつでも終了されます。ファイルが再度使用可能になった場合、仮想記憶間サービスを使用するブラウズ要求は、影響を受けていない状態で保つことができます。 (FORCE については、 37 ページの『切断』を参照してください)。

ユーザー出口

機能シップされた要求の場合、EXEC インターフェースのユーザー出口である XEIIN と XEIOUT、ファイル制御のユーザー出口である XFCREQ と XFCREQC は、AOR と FOR の両方で呼び出されます。

仮想記憶間の要求の場合、これらのユーザー出口は AOR でのみ呼び出されます。

セキュリティー検査

機能シップされた要求の場合、FOR のセキュリティー検査は、各作業単位で特定のファイルを参照する最初の要求で呼び出されます。このため、トランザクション・レベルのセキュリティー検査は、FOR で実行できます。

仮想記憶間の要求の場合、セキュリティー検査は、CONNECT のときにのみ呼び出されます。このため、トランザクション・レベルのセキュリティー検査は、FOR で実行できません。

読み取り要求の失敗

機能シップを使用する読み取り要求に失敗した場合、入力域は変わりません。

仮想記憶間サービスを使用する読み取り要求に失敗した場合、レコードを取得していなくても入力域が変更されるという可能性があります。このため、キーに変更がないと分かっている場合でも、入力域に変更がないことを当てにしないでください。

EXEC インターフェース・ブロック

場合によっては、仮想記憶間サービスを使用する読み取り要求が EIBRESP2 フィールドの値を返すことに注意してください。ただし、機能シップされた要求は値を返すことがないため、読み取り要求によって設定されるこのフィールドにアプリケーションが依存しないようにしてください。

キーの長さ

機能シップされた要求の場合、正しいキーの長さを、AOR のリモート・ファイル定義に指定するか、または明示的にファイル要求で指定する必要があります (FOR の VSAM 定義のキーの長さに合わせて)。指定しない場合は、ファイルにアクセスするどの要求に対しても、INVREQ 状態が返されます。これは、データ・テーブルとして定義されたファイルだけでなく、どのファイルにも適用されます。

仮想記憶間の要求の場合、AOR のキーの長さは使用しません。 AOR でキーの長さが指定されていない場合でも、AOR で指定されたキーの長さが FOR のキーの長さと一致しない場合でも、要求は正常に実行できます。ただし、要求の一部は機能シップされる可能性があるため、アプリケーションではこれに依存しないようにしてください。

共用データ・テーブル・サービスと **VSAM** の相違点

多くのデータ・テーブル要求で、共用データ・テーブル・サービスが VSAM に取って代わったため、特定の要求を実装する方法には相違があります。

更新中の読み取り **(**異なるトランザクション**)**

データ・テーブル・レコードに対する READ コマンドが、別の トランザクションによってそのレコードに対して発行された READ UPDATE の後に続き、その関連付けられた更新要求よりも先行する場合、共用データ・テーブル・サービスを使用すれば、READ コマンドがただちに処理されます。

VSAM を使用する場合、READ コマンドは、更新要求が完了するまで待機します。

更新中の読み取り (同一トランザクション)

データ・テーブル・レコードに対する READ コマンドが、同じ トランザクションによってそのレコードに対して発行された READ UPDATE の後に続き、その関連付けられた更新要求よりも先行する場合、共用データ・テーブル・サービスを使用すれば、READ コマンドがただちに処理されます。

VSAM を使用する場合、トランザクションでデッドロックの異常終了 AFCG が発生します。

ブラウズ中の削除

データ・テーブル・レコードに対する STARTBR または RESETBR コマンドに共用データ・テーブル・サービスを使用する場合、関連付けられた READNEXT または READPREV コマンドが発行される前にレコードを削除できます。 VSAM を使用する場合、関連付けられた READNEXT または READPREV コマンドを発行する前にレコードを削除できません。

このため、共用データ・テーブル・サービスを使用する場合は、特別な「最終レコード」キー (X'FF....') 以外のキーを指定して STARTBR または RESETBR コマンドが発行され、選択されたレコードが READNEXT コマンドより先に削除されていれば、READNEXT コマンドが後続のレコードを読み取ります。

後続のレコードがない場合は、ENDFILE 状態が返されます。STARTBR または RESETBR で EQUAL オプションが使用された場合、読み取られるレコードのキーは、指定されたキーと一致しないことがあります。

STARTBR または RESETBR コマンドが特別な「最終レコード」キーを指定して発行された場合で、選択されたレコードが READPREV コマンドより先に削除されていれば、READPREV コマンドは先行のレコードを読み取ります。レコードがない場合は ENDFILE 状態を返します。

ブラウズ中の書き込み

共用データ・テーブル・サービスを使用する場合、ブラウズでファイルの最後まで読み取って、ENDFILE 状態が発生したときに、ファイルの最後より後ろに新しいレコードが挿入されれば、以降の READNEXT では新しいレコードを読み取ることができます。

VSAM を使用する場合、以降の READNEXT では、新しいレコードを見つけられない可能性がありますが、代わりに再度 ENDFILE 状態が報告されます。

更新中の削除 (同一トランザクション)

データ・テーブル・レコードに対して RIDFLD を指定する DELETE コマンドに共用データ・テーブル・サービスを使用すると、その DELETE コマ

ンドが、同じトランザクションによってそのレコードに対して発行された READ UPDATE の後に続き、その関連付けられた更新要求よりも先行する場合、DELETE コマンドは正常に処理され、関連付けられた更新要求は NOTFND 状態を受信します。

第 4 章 ユーザー出口を使用したデータ・テーブルのカスタマイズ

データ・テーブル・サービスには、3 つのグローバル・ユーザー出口点があります。 これらの各ユーザー出口点で実行される 1 つ以上のアセンブラー言語プログラムを提供することにより、CICS が提供する機能を拡張または変更することができます。

注: このセクションには、プロダクト・センシティブ・プログラミング・インターフェースと関連ガイダンス情報が含まれています。

CICS の SDFHSAMP ライブラリーには、グローバル・ユーザー出口 XDTRD、XDTAD、および XDTLC のサンプル・ユーザー出口プログラムが用意されています。本書で再現されているこれらのプログラムは、コーディングとデータ定義シーケンスのサンプルを用いて、共用データ・テーブルで使用するユーザー出口プログラムに適用される規則を説明しています。これらのサンプルは、一般的なガイドとしてのみ提供されているため、プログラミング・インターフェースは定義していません。

グローバル・ユーザー出口に関するプログラミング情報と、それらの使用方法については、Global user exit programsを参照してください。

注: 仮想記憶間サービスでデータ・テーブルへのアクセス要求が満たされる場合、EXEC インターフェースのユーザー出口である XEIIN と XEIOUT、ファイル制御のユーザー出口である XFCREQ と XFCREQC は、ファイル所有領域で開始されません。

CICS と共用データ・テーブル出口プログラム間の通信

CICS とデータ・テーブル出口プログラムの間で情報を受け渡すためには、パラメーター・リストを使用します。

CICS の CICSTS55.CICS.SDFHMAC ライブラリーには、DFHXDTDS という名前のコピーブックがあり、この中に、このパラメーター・リストを定義するための DSECT が含まれています。各出口プログラムに COPY DFHXDTDS ステートメントを組み込みます。 44 ページの図 6に DSECT を示します。

この DSECT で使用されているフィールド名を、図の後に続くユーザー出口の説明の中で参照します。

© Copyright IBM Corp. 2005, 2018

```
************************************************************************
*                                                                      *
*   Data Table Parameter List for User Exits XDTRD, XDTAD and XDTLC.  *
*                                                                      *
*   Some of the parameters are only used by one or two of the exits.  *
*   This is indicated in the comments for those parameters.           *
*   The comments also indicate whether the field is used for input    *
*   (In), output (Out), or both (In/Out).                             *
*                                                                      *
*   This definition can be used by exit programs running on CICS      *
*   regions which are at a level to support coupling facility data    *
*   tables (CFDT), providing the UEPDTCFT flag is used to test        *
*   whether the exit has been invoked from within coupling facility   *
*   data tables support, and that parameters which are specific to    *
*   CFDT support are only used when it is set.  CFDT support will     *
*   only be available to exit programs running on CICS regions at     *
*   the CICS Transaction Server version 1 release 3 level or higher.  *
*                                                                      *
*   This definition can be used by exit programs running on CICS      *
*   regions which are at a level to support shared data tables (SDT), *
*   or which have SDT support installed, providing the UEPDTSDT flag  *
*   is used to test whether the exit has been invoked from within     *
*   shared data tables support, and that the parameters which are     *
*   specific to SDT support are only used when it is set.  SDT        *
*   support will only be available to CICS regions running at the     *
*   CICS/ESA version 4 release 1 level or higher (or running on       *
*   CICS/ESA version 3 release 3 if the Shared Data Tables feature    *
*   is installed).                                                    *
*                                                                      *
*   This definition can also be used by exit programs running on      *
*   CICS regions which are not at a level to support either SDT or    *
*   CFDT, indicated by both the UEPDTCFT and UEPDTSDT flags being     *
*   off.  In this case, only the parameters which relate to the       *
*   basic data tables support can be used.  Basic data tables         *
*   support will only be available to CICS regions running on one     *
*   of the following levels:                                          *
*   - CICS/MVS version 2 (plus data tables SPE on some releases)      *
*   - CICS/ESA version 3 releases 1 or 2                              *
*   - CICS/ESA version 3 release 3 if SDT feature is NOT installed    *
*                                                                      *
*   Careful use of these flags, and of the parameters which relate    *
*   to the various kinds of data tables support, should allow the     *
*   same user exit program to be used for more than one kind of data  *
*   table.                                                            *
*                                                                      *
************************************************************************
DT_UE_PLIST_DSECT  DSECT ,
DT_UE_PLIST        DS  0XL84            Data Table User Exits          X
                                        Parameter List
UEPDTNAM             DS  CL8            Data table name (In)
UEPDTFLG             DS  0CL1           Flags (In):
```

図 *6.* データ・テーブル・ユーザー出口パラメーター・リスト

```
*-------------------------------------------------------------------*
*           The UEPDTSDT and UEPDTCFT flags indicate whether the    *
*           exit has been invoked for shared data tables or for     *
*           coupling facility data tables support.  If neither is   *
*           set, then the exit has been invoked under the basic     *
*           data tables support which pre-dated shared data tables. *
*                                                                   *
*           The UEPDTCMT and UEPDTUMT flags are available only to   *
*           exits which have been invoked by shared data tables     *
*           support.  They distinguish the two kinds of shared      *
*           data table.  Please note that on releases earlier than  *
*           CICS Transaction Server 1.3, the UEPDTUMT flag is NOT   *
*           available; on these releases, a user-maintained data    *
*           table is implied by the UEPDTCMT flag being turned off. *
*                                                                   *
*           If the exit has been invoked by CFDT support, then the  *
*           data table can only be a coupling facility data table,  *
*           so there are no extra flags to identify the kind of     *
*           data table when the exit has been invoked by coupling   *
*           facility data tables support.                           *
*                                                                   *
*           The UEPDTOPT flag is available to exits which have      *
*           been invoked by either shared data tables support or    *
*           coupling facility data tables support (but not to exits *
*           which have been invoked by basic data tables support).  *
*           This flag is therefore not available on releases        *
*           earlier than CICS/ESA 3.3 (plus SDT support).           *
*-------------------------------------------------------------------*
                      DS  BL1
UEPDTSDT              EQU X'80'        Exit invoked by SDT support
UEPDTCMT              EQU X'40'        Table is CICS-maintained
UEPDTOPT              EQU X'20'        Exit invoked by table loader,  X
                                       so optimization of the load by X
                                       skipping may be requested      X
                                       (flag is for XDTRD only)
UEPDTCFT              EQU X'10'        Exit invoked by CFDT support   X

UEPDTUMT              EQU X'08'        Table is user-maintained
*                     EQU X'07'        Reserved
```

図 7. データ・テーブル・ユーザー出口パラメーター・リスト (続き)

```
*-----------------------------------------------------------------------*
*          The following fields are available to exits which          *
*          have been invoked by all flavors of data tables support.   *
*          Not all fields are available at all of the exit points.    *
*-----------------------------------------------------------------------*
UEPDTORC             DS  AL1            Data table load return code -  X
                                        XDTLC only, values below (In)
                     DS  BL2            Reserved
UEPDTRA              DS  A              Data record address - XDTRD    X
                                        and XDTAD only (In)
UEPDTRBL             DS  F              Data buffer length - XDTRD and X
                                        XDTAD only (In)
UEPDTRL              DS  F              Data table record length -     X
                                        XDTRD and XDTAD only, XDTRD    X
                                        can return new length in here  X
                                        if it amends record (only      X
                                        allowed for UMT or CFDT)       X
                                        (In/Out)
UEPDTKA              DS  A              Key address - XDTRD and XDTAD  X
                                        only (In)
UEPDTKL              DS  F              Key length - XDTRD and XDTAD   X
                                        only (In)
```

図 *8.* データ・テーブル・ユーザー出口パラメーター・リスト *(*続き*)*

```
*-----------------------------------------------------------------------*
*          The following fields are available to exits which          *
*          have been invoked either by shared data tables support     *
*          or by coupling facility data tables support.               *
*          Not all fields are available at all of the exit points.    *
*-----------------------------------------------------------------------*
UEPDTDSL             DS  F              Length of data set name (In)
UEPDTDSN             DS  CL44           Source data set name (In)
UEPDTSKA             DS  A              Address of skip-key area: exit X
                                        should return a key of length  X
                                        UEPDTKL in this area if it has X
                                        requested optimisation of load X
                                        by skipping - XDTRD only (In)
*-----------------------------------------------------------------------*
*   Values for UEPDTORC (supplied to XDTLC exit only)                   *
*-----------------------------------------------------------------------*
UEPDTLCS           EQU 0               load completed successfully
UEPDTLFL           EQU 128             load failed
```

図 *9.* データ・テーブル・ユーザー出口パラメーター・リスト *(*続き*)*

ユーザー出口では、レジスター 15 に戻りコードを設定する必要があります。戻りコード値は DFHUEXIT マクロによって指定されます。各ユーザー出口の有効な値は、以下の説明で示します。

出口プログラムが引き続き基本データ・テーブルと共用データ・テーブルで機能するようにするには、 UEPDTFLG をチェックして、どのバージョンのデータ・テーブル・サポートが出口プログラムを呼び出したかを確認します。共用データ・テーブルでは、このフラグ・バイトはどのタイプのデータ・テーブルが使用されているかと、ロード中に出口プログラムが呼び出されているかどうかも示します。

出口プログラムは、ファイル名 (UEPDTNAM フィールド) またはソース・データ・セットの名前 (UEPDTDSN および UEPDTDSL のフィールドを参照) を使用して、このファイルに対して実行するアクションを決定する必要があります。

同じ出口点で複数の出口プログラムを有効にすることができます。例えば、それぞれの出口プログラムで特定のファイルやデータ・セットに対してアクションを実行することができます。

XDTRD ユーザー出口

XDTRD ユーザー出口は、CICS がソース・データ・セットから取得したレコードをデータ・テーブルに追加しようとする直前に呼び出されます。このレコードをデータ・テーブルにロードするかしないかを選択できます。ユーザー保守データ・テーブルの場合、レコードを変更することもできます。

XDTRD は通常、ロード処理がソース・データ・セットの順次コピー中にレコードを取得するときに呼び出されます。ただし、アプリケーションがデータ・テーブルにはないレコードを取得するときに、以下のいずれかの条件が当てはまる場合にも呼び出されます。

- ユーザー保守データ・テーブルの場合で、ロードがまだ進行中。
- CICS 保守データ・テーブルの場合で、ソース・データ・セットの最後に達する前にロードが終了した (例えば、データ・テーブルがフルになったため)。

ソース・データ・セットから取得されたレコードは、ユーザー出口プログラムにパラメーターとして渡されます (フィールド UEPDTRA および UEPDTRL を参照)。このプログラムは、例えばキー値に応じて (フィールド UEPDTKA および UEPDTKL を参照)、レコードをデータ・テーブルに含めるか含めないかを選択できます。

あるいは、出口プログラムは、指定されたキーに至るまでの後続するすべてのレコードをスキップするように要求できます (フィールド UEPDTSKA を参照)。スキップされるレコードは出口プログラムには渡されません。この機能はロード中にのみ使用可能です。このキーは、完全なキーとして指定するか、またはスキップ・キー域に 2 進ゼロを埋め込むことによって先行文字だけを指定することができます。

必要なアクションは、戻りコードを設定することによって指示します。戻りコード値に応じて、CICS は以下のアクションをとります。

表 3. *XDTRD* ユーザー出口の戻りコード： 出口プログラムが発行した呼び出しで PURGED 応答を受け取った場合、UERCPURG の値を返す必要があります。

戻りコード	アクション
UERCDTAC	このレコードをデータ・テーブルに含めます。出口が起動されない場合、これがデフォルトです。
UERCDTRJ	このレコードをデータ・テーブルに含めません。
UERCDTOP	スキップ・キー域に指定したキー以上のキーが見つかるまで、このレコードと後続するレコードをスキップします。

ユーザー保守データ・テーブルの場合、プログラムでレコード中のデータを変更し、データ・テーブルのストレージ必要量を削減することもできます。データ・テーブルを使用するアプリケーション・プログラムは、出口プログラムがレコード形式に加えるあらゆる変更を認識していなければなりません。レコード長を変更する場合、出口プログラムは新しい長さをパラメーター・リストに設定する必要があり

ます (フィールド UEPDTRL を参照)。新しい長さがデータ・バッファー長を超えてはなりません (フィールド UEPDTRBL を参照)。

サンプル XDTRD 出口プログラム: DFH$DTRD

DFH$DTRD は、サンプル XDTRD グローバル・ユーザー出口プログラムです。これは、共用データ・テーブル用の XDTRD ユーザー出口の使用法を示します。このサンプル・プログラムは、SDFHSAMP ライブラリーにあります。

XDTAD ユーザー出口

XDTAD は、初期ロード後にソース・データ・セットに追加されるレコードごとに呼び出されます。このレコードをデータ・テーブルに追加するかどうかを選択できます。このユーザー出口は、レコードの変更はできません。アプリケーションによってレコードが記述されるときにレコードはすでにデータ・テーブルで使用されるフォーマットになっていると想定されるためです。

XDTAD ユーザー出口は、データ・テーブルに対して書き込み要求が発行されると呼び出されます。

- ユーザー保守データ・テーブルの場合、レコードがデータ・テーブルに追加される前に、このユーザー出口が 1 回呼び出されます。
- CICS 保守データ・テーブルの場合、このユーザー出口は 2 回呼び出されます。まずソース・データ・セットにレコードが追加される前に 1 回呼び出され、データ・テーブルにレコードが追加される前に再び呼び出されます。

注: カップリング・ファシリティー・データ・テーブルでは、オープン TCB で出口を呼び出すことができます。したがって、過度な TCB 切り替えが発生しないように、スレッド・セーフな出口を作成し、CICS に対してスレッド・セーフとして有効にしてください。

アプリケーションによって書き込まれたレコードは、パラメーターとしてユーザー出口プログラムに渡されます (UEPDTRA および UEPDTRL フィールドを参照)。このプログラムは、(キー値 (例えば、UEPDTKA および UEPDTKL フィールドを参照) に応じて) レコードをデータ・テーブルに含めるか含めないかを選択できます。この決定は戻りコードを設定することによって指示します。

戻りコード値に応じて、CICS は以下のアクションを実行します。

表 *4. XDTAD* ユーザー出口の戻りコード： 出口プログラムが発行した呼び出しで PURGED 応答を受け取った場合、UERCPURG の値を返す必要があります。

戻りコード	アクション
UERCDTAC	レコードをデータ・テーブルに追加します。出口が起動されない場合、これがデフォルトです。
UERCDTRJ	レコードをデータ・テーブルに追加しません。

XDTAD 出口では、レコード内のデータを変更してはなりません。ユーザー保守データ・テーブルのロード時に XDTRD を使用してデータ・レコードを切り捨てた場合は、そのデータ・テーブル用の正しい形式でのみレコードの書き込みを試行するようにアプリケーションをコーディングする必要があります。

サンプル XDTAD 出口プログラム: DFH$DTAD

DFH$DTAD は、サンプル XDTAD グローバル・ユーザー出口プログラムです。これは、共用データ・テーブル用の XDTAD ユーザー出口の使用法を示します。このサンプル・プログラムは、SDFHSAMP ライブラリーにあります。

XDTLC ユーザー出口

XDTLC ユーザー出口は、データ・テーブルのロードの完了時に、ロードが成功したかどうかに関係なく呼び出されます。何らかの理由でロードの完了前にデータ・テーブルが閉じた場合、このユーザー出口は呼び出されません。

ロードが正常に完了しなかった場合は、出口プログラムに通知されます (フィールド UEPDTORC を参照)。これが発生するのは、例えば、最大レコード数に達した場合や仮想ストレージが不十分だった場合です。その場合、出口プログラムは、戻りコードを設定することによって、ファイルの即時クローズを要求できます。

戻りコード値に応じて、CICS は以下のアクションを実行します。

表 *5. XDTLC* ユーザー出口の戻りコード： 出口プログラムが発行した呼び出しで PURGED 応答を受け取った場合、UERCPURG の値を返す必要があります。

戻りコード	アクション
UERCDTOK	アクションはありません。ファイルは開いたままとなります。出口が起動されない場合、これがデフォルトです。
UERCDTCL	ファイルを閉じます。

サンプル XDTLC 出口プログラム: DFH$DTLC

DFH$DTLC は、サンプル XDTLC グローバル・ユーザー出口プログラムです。これは、共用データ・テーブル用の XDTLC ユーザー出口の使用法を示します。このサンプル・プログラムは、SDFHSAMP ライブラリーにあります。

データ・テーブル用ユーザー出口の活動化

データ・テーブルのユーザー出口を活動化するには、ここに示す手順を実行します。

手順

1. どのユーザー出口を使用するかを決定します。 各ユーザー出口の説明については、 43 ページの『第 4 章 ユーザー出口を使用したデータ・テーブルのカスタマイズ』を参照してください。
2. ユーザー出口プログラムを記述します。 サンプルは 43 ページの『第 4 章 ユーザー出口を使用したデータ・テーブルのカスタマイズ』にあります。
3. CICS にユーザー出口プログラムを定義します。これを行うには、PROGRAM リソースの説明に従って、CEDA DEFINE PROGRAM コマンドを使用します。
4. **EXEC CICS ENABLE** コマンドを使用して、ユーザー出口を活動化します。 後で必要に応じて、ユーザー出口は **EXEC CICS DISABLE** コマンドを使用して非活動化できます。

CEMT コマンドまたは EXEC CICS コマンドを使用してデータ・テーブルのオープンを明示的に制御しているのではない限り、多くの場合、CICS 始動中にユーザー出口を活動化する必要があります。これを行わないと、ユーザー出口が活動化する前にデータ・テーブルのロードが開始する可能性があります。始動時にユーザー出口を活動化するには、以下のようにします。

5. ユーザー出口を活動化する **EXEC CICS ENABLE** コマンドが含まれる、1 つ以上のプログラム・リスト・テーブル初期化後 (PLTPI) プログラムを記述します。PLTPI プログラムに関するプログラミング情報について詳しくは、Writing initialization and shutdown programsを参照してください。
6. それらの各 PLTPI プログラムをエントリーとして持つプログラム・リスト・テーブル (PLT) を定義します。これは、Program list table (PLT)にある説明に従って行います。
7. システム初期化パラメーター **PLTPI=suffix** を指定します。これは、PLTPI システム初期設定パラメーターの説明に従って行います。前のステップで定義した PLT の接尾部を使用します。これにより、PLTPI プログラムは初期化の第 2 段階で、どのファイルも開かれる前に実行されるようになります。

次のタスク

ユーザー出口を CICS シャットダウン中に無効にするには、同様の方法で、PLT シャットダウン (PLTSD) プログラムを使用できます。

第 5 章 データ・テーブルの管理

データ・テーブルの操作面に関する情報。

データ・テーブルのオープン

データ・テーブルは、アプリケーションによって使用される前に開かれなければなりません。

データ・テーブルを開く方法は、CICS ファイルの場合と同じです。以下の方法の 1 つを使用します。

- データ・テーブルが OPENTIME(STARTUP) を使用して定義されている場合は、CICS 提供トランザクション CSFU によって、CICS 始動の最後に自動的に開きます。
- ユーザーが発行した CEMT 要求または EXEC CICS 要求によって明示的に開きます。
- データ・テーブルが OPENTIME(FIRSTREF) を使用して定義されている場合は、データ・テーブルへの最初の参照で暗黙のうちに開かれます。閉じているデータ・テーブルへの最初のリモート・アクセスで、暗黙のうちに開かれます。

CICS ファイルを開くためのすべての規則とオプションは、データ・テーブルとして定義されたファイルにも適用されます。さらに、データ・テーブルのロードが開始されます。

大規模なデータ・テーブルでは、ロードに多大な時間がかかる可能性があります。データ・テーブルへのアクセス用の開発では、ユーザー保守データ・テーブルで使用できるアプリケーション・プログラミング・コマンドについて、および CICS 保守データ・テーブルで得られるパフォーマンス向上がロードの完了までどのように制限されるかについて説明しています。

ファイルのオープン処理中に以下のステップが実行されます。

1. VSAM ソース・データ・セットのアクセス方式制御ブロック (ACB) が、分離した MVS タスク制御ブロック (TCB) の下で開かれます。このステップは、すべての CICS ファイルの場合と同様です。
2. 領域によって使用される最初のデータ・テーブルの場合、CICS によって以下が行われます。
   - 共用データ・テーブル・サポートによって使用される MVS ストレージ・プールの作成
   - この領域のデータ・テーブルによって使用される MVS データ・スペースの作成
   - サーバーとしての LOGON 操作試行
3. データ・テーブルをデータ・スペースにロードするため、特殊 CICS トランザクション CFTL を接続します。

© Copyright IBM Corp. 2005, 2018

4. データ・テーブルのオープン要求を発行するトランザクションが、処理を続行できるようになります。
5. CICS はメッセージ DFHFC0940 を発行して、ロードが開始したことを示します。このメッセージは CSFL 一時データ・キューに送信されます。
6. データ・テーブルをロードするトランザクションによって、ソース・データ・セットが順次読み取られます。ユーザー出口 XDTRD のオプション制御の下で、トランザクションはレコードをデータ・スペース・ストレージにコピーします。
7. CICS はロード結果を示すメッセージを発行します。メッセージ番号は以下のとおりです。
   - ロード成功の場合: DFHFC0941
   - ロード失敗の場合:
     DFHFC0942、DFHFC0943、DFHFC0945、DFHFC0946、DFHFC0947、または DFHFC0948

   このメッセージは CSFL 一時データ・キューに送信されます。また、ロード失敗の場合、メッセージはコンソールにも送られます。
8. ロードが完了したとき (成功したかどうかによらず)、ユーザー出口 XDTLC がアクティブであればそれが呼び出されます。ロードが正常に完了しなかった場合、出口プログラムはデータ・テーブルを閉じるように要求できます。
9. ユーザー保守データ・テーブルの場合、ソース・データ・セットの ACB はロード完了時に閉じられます。データ・セットは、最初に動的に割り振られていた場合は割り振り解除され、他の ACB がまだそのデータ・セット用に開かれていない限り、他のジョブが使用できるようになります。

注: 緊急時再始動中に、バックアウト・アクションを必要とするすべてのファイルは再オープンされます。ただし、ファイルがデータ・テーブルとして定義されている場合、ロードはその時点では開始しません。その代わり、緊急時再始動の最後に CSFU トランザクションによってロードが開始されます。これによって、ロード中にデータ・テーブルへのレコードのコピーを制御するユーザー出口には、PLTPI 処理の任意の段階で活動化される機会が与えられます。

データ・テーブルのクローズ

すべての CICS ファイルの場合と同じ方法で、データ・テーブルは閉じられます。

以下のいずれかの方法を使用します。

- ユーザーが発行した CEMT 要求または EXEC CICS 要求によって明示的に閉じます。
- CICS が正常にシャットダウンするときに暗黙的に閉じます。

CICS ファイルを閉じるためのすべての規則とオプションは、データ・テーブルにも適用されます。特に以下の点に注意してください。

- ファイルの現行ユーザーの静止に関する規則が適用されます (ただし、AOR で実行中トランザクションがブラウズ・シーケンスの最中であっても、ファイルは閉じられる場合があります)。

- ユーザー保守データ・テーブルの場合、データ・テーブルがリカバリー可能として定義されているときは、データ・テーブルがクローズ可能になる前に、データ・テーブルを変更するすべての作業単位が完了していなければなりません。

データ・テーブル・レコードで使用されるデータ・スペース・ストレージは、クローズ操作の一部として解放されます。ファイルがクローズ後に再オープンされる場合の処理は、ファイルが前に開かれなかった場合と同じです。

シスプレックス内での共用データ・テーブル・サポートの使用

単一の MVS 環境でユーザー保守データ・テーブルを現在使用しているが、シスプレックス環境に移動することを予定している場合は、この情報をお読みください。また、既にシスプレックスがある場合も、この情報は、その環境での共用データ・テーブル・サポートの活用方法を示すことができるので、役立つ可能性があります。

シスプレックス内の共用データ・テーブル・サポートの概要

共用データ・テーブルは、単一の MVS イメージ内でのみ共用データ・テーブル・サポートを活用できます。ただし、共用ユーザー保守データ・テーブルへの読み取りアクセスのみが必要なアプリケーション、または変更が即時表示される必要がないアプリケーションについては、共用データ・テーブルの使用をシスプレックス環境に拡張することができます。

共用データ・テーブルは、複数の MVS イメージにわたり機能シップを使用して共用できることに留意してください。

ユーザー保守データ・テーブルは、1 つの MVS につき 1 つのデータ・テーブルずつ、シスプレックス全体で複製することができます。各 MVS イメージ内に 1 つの共用データ・テーブル・サーバー領域を持っている必要があります。そして、それぞれの領域は、その MVS 内の他の CICS 領域が共用データ・テーブルを使用してアクセスできるユーザー保守データ・テーブルを所有している必要があります。これらの他の領域には、それらのサーバー領域内のユーザー保守データ・テーブルを参照するリモート・ファイル定義が必要です。各ユーザー保守データ・テーブル (UMT) は、同じソース・データ・セットを持っており、このデータ・セットはすべての共用データ・テーブル・サーバー領域によって読み取り可能でなければなりません。アクセスが読み取り専用で、データが決して更新されない場合、これにより、事実上、シスプレックス内の共用ユーザー保守データ・テーブルが提供されます。

おそらく可能性として高いと思いますが、基礎となるデータは時々変更されるが、そのような変更を UMT に即時反映する必要がないという場合は、定期的に何らかの処理を実行して UMT の内容をリフレッシュすることができます。そうすることにより、UMT をクローズして再ロードせずに、基礎となるデータに一致するように UMT の内容が更新されます。変更は、非データ・テーブル・ファイル定義によりデータ・セットを参照する CICS アプリケーションを使用するか、バッチ・プログラムを使用して、ユーザー保守データ・テーブルではなく、ソース・データ・セットに適用されます。 59 ページの『複製されたユーザー保守データ・テーブルをリフレッシュするためのサンプル・プログラムのソース・コード』で説明されているサンプル COBOL アプリケーション・プログラムは、UMT をリフレッシュし

て、ソース・データ・セットの現行内容を反映する方法を示しています。このプログラムは、各 MVS イメージで実行され、そのイメージ内の UMT を更新します。そのようなプログラムは、一日のうちの規定の時刻に実行したり、ユーザー要求に応じて実行したりすることができます。そのようなプログラムは、UMT への機能シップ更新を避けるために、共用データ・テーブル・サーバー領域で実行するのが最も効果的です。

シスプレックス内のすべての MVS イメージの CICS 領域が、それらのデータ・ビューで同期していることが重要である場合は、リフレッシュ・プログラムが実行されている間、データを読み取るトランザクションを停止し、すべての MVS システムでプログラムが完了した後に再始動する必要があります。

この手法は、ユーザー保守データ・テーブルにのみ適切です。その理由は以下のとおりです。

- 読み取り専用アクセスが必要なところでは、ユーザー保守データ・テーブルが通常の選択である。
- CICS 保守データ・テーブルでは、テーブルに影響を与えないまま、ソース・データ・セットに更新を適用することは不可能である。
- ソース・データ・セットに対して行われた更新は、更新が行われたシステム上のテーブルにのみ反映される。

複製されたユーザー保守データ・テーブルのリフレッシュ方法

以下のステップでは、複製された UMT をリフレッシュするための環境のセットアップ方法について説明します。実際のところ、このうちのいくつかを既に実施されている可能性があります。例えば、データ・テーブルとして定義されているファイルは既にある可能性があります。ここで説明するステップでは、既にシスプレックス環境があることを想定しています。

1. 適切なファイルを選択します。

   実例として、クレジットカード番号を、盗まれたクレジットカード・リストに照らし合わせてチェックし、このリストへの迅速なアクセスを必要とするアプリケーションについて検討します。このリストは、盗まれたカード番号の新しいバッチを使用して定期的に更新されます。このアプリケーションは、ファイル名 UMTNAME を使用して、PRODN.SOURCEDS という名前の VSAM KSDS データ・セット内のレコードにアクセスします。このアプリケーションは、2 つの MVS イメージから成るシスプレックスで稼働します。CICS 領域の CICS1A、CICS1B、および CICS1C は 1 番目のイメージで稼働し、CICS2A、CICS2B、および CICS2C は 2 番目のイメージで稼働します。
2. ファイル定義を以下のようにセットアップします。
   - シスプレックス内の各 MVS で、このファイルの共用データ・テーブル・サーバーになる CICS 領域を選択します。この領域内に、アプリケーションがユーザー保守データ・テーブルとしてデータを読み取るために使用するファイル名と、ソース・データを含むデータ・セット名を指定します。

     この図では、CICS1A と CICS2A がサーバー領域としてセットアップされ、UMTNAME という名前のファイルがそれらに定義されています。ファイル定義は、DSNAME を PRODN.SOURCEDS、TABLE を USER、そして許可される操作 YES を READ、BROWSE、ADD、DELETE、および

UPDATE に対して指定しています (この理由は、このファイル定義が、リフレッシュ時に、データの読み取りと UMT の更新の両方に使用されるためです)。

- シスプレックス内のその他のすべての領域については、同じ MVS 内の共用データ・テーブル・サーバー領域として REMOTESYSTEM、およびその領域内の UMT の名前として REMOTENAME を指定して、アプリケーションがリモート・ファイルとしてデータを読み取るために使用するファイル名を指定します。

  したがって、この図では、UMTNAME という名前のファイルが CICS1B および CICS1C に定義され、CICS1A のシステム ID として REMOTESYSTEM、UMTNAME として REMOTENAME が指定されます。そして今回は、これらのリモート定義を介して UMT を更新する必要がないため、唯一許可される操作として READ および BROWSE が指定されています。同様のファイル定義が CICS2B と CICS2C にセットアップされますが、それらにとっては CICS2A がリモート・システムになります。
- それぞれの共用データ・テーブル・サーバー領域に、UMT のリフレッシュの際にソース・データ・セットを読み取るために使用できるファイル定義をセットアップします。

  この図では、SOURCEDS という名前のファイルが CICS1A と CICS2A に定義されており、DSNAME は PRODN.SOURCEDS、TABLE は NO に指定され、許可される操作は READ 操作と BROWSE 操作のみになります。
- (ソース・データ・セットにアクセスできる) シスプレックス内の 1 つの領域に、ソースに更新を適用するために使用されるファイルを定義します。このファイル定義は、ソース・データ・セットを読み取るためにリフレッシュ・プログラムによって使用されるものと同じ可能性がありますが、この場合は、読み取り操作と更新操作の両方を許可する必要があります。ユーザーが希望する場合、バッチ・プログラムを使用してデータ・セットを更新することもできます。その場合、この CICS ファイル定義は必要ありません。

  この図は、UMT のリフレッシュに使用されるファイル定義と同じファイル定義を使用しています。この場合、いずれかの領域が、すべてのファイル操作を許可するものとして SOURCEDS を定義する必要があります。

3. ソース・データ・セットの読み取りまたは更新が必要なすべてのアプリケーションがアクセスできるように、ソース・データ・セットをセットアップします。

   DFSMS/MVS バージョン 1 リリース 3 を使用している場合は、ファイル定義に RLSACCESS(Yes) を指定することにより、読み取りまたは更新のためにどの CICS 領域からもデータ・セットにアクセスできます。RLS アクセス・モードを使用している場合は、データ・セットがリカバリー不能でない限り、バッチ・プログラムから更新を適用することはできないことに留意してください (この理由は、RLS モードでは、リカバリー可能データ・セットを更新のためにオープンできるのは CICS のみだからです)。

   DFSMS/MVS の以前のリリースを使用している場合は、ソースに更新を適用するプログラムによって更新でき、他のすべてのプログラムによって読み取れるように、データ・セット SHAREOPTIONS をセットアップすることができ

ます。あるいは、読み取り中でない時にのみ更新できるようにデータ・セットをセットアップし、データ・セットのオープンがシリアライズされるようにすることもできます。これらの共用オプションがシスプレックス全体で機能するようにするためには、GRS (グローバル・リソースの逐次化) を使用する必要があります。

この図では、RLS が使用可能でない場合、次のいずれかを使用して PRODN.SOURCEDS を定義します。

- SHR(2)。これにより、データ・セットに変更を適用するプログラムを実行する領域による更新が可能になり、同時にすべてのリフレッシュ・プログラムによる読み取りが可能になります。

または

- SHR(1)。そして、通常、変更を適用するプログラムに対してオープンしておきます。次に、リフレッシュされる時には、それへのアクセスをクローズし、各サーバー領域で順番に、オープンし、リフレッシュ・プログラムを実行し、次の領域がオープンできるように CLOSE DISABLE を実行します。

4. UMT およびソース・データ・セットとしてユーザーのファイルを指定し、データ定義がユーザーのレコードのレイアウトに一致するように、サンプル・プログラムを変更します。サーバー領域にプログラムとトランザクションを定義します。

   図のファイル名は、プログラム内のものと同じです (UMTNAME および SOURCEDS)。プログラムとそれを実行するトランザクションを、CICS1A および CICS2A に定義します。
5. これで、複製した UMT の使用を開始する準備ができました。
6. ソース・データ・セットとその初期内容を事前準備します。
7. 共用データ・テーブル・サーバー領域で UMT をオープンして、ソース・データ・セットの内容を各領域にロードします。
8. シスプレックス内のすべての領域でアプリケーションを開始します。それらのアプリケーションはすべて、データ・テーブル共用を使用してデータにアクセスすることができます。

   MVS 1 で実行されているアプリケーションは、CICS1A 内の UMT を介してデータにアクセスし、MVS 2 で実行されているアプリケーションは CICS2A 内の UMT を介してデータにアクセスします。
9. 新しいデータが到着したら、ソース・データ・セットを更新します。

   この図では、データはファイル SOURCEDS によって更新されます。
10. アプリケーションが新しいデータにアクセスするようにしたい場合は、各サーバー領域でソース・データ・セットと UMT を読み取るトランザクションを実行し、UMT をリフレッシュして、ソース・データ・セットに一致するようにします。ある MVS に表示されるデータが別の MVS に表示されるデータと若干異なっている場合にアプリケーションが無効化されないのであれば、リフレッシュの実行中にそれらのアプリケーションの実行を停止する必要はありません。

ユーザー保守データ・テーブルをリフレッシュするためのサンプル・プログラム

独自のプログラムの作成に役立つように、UMT をオープンしたままの状態で、ソース・データ・セットに一致するようにリフレッシュする方法を示す COBOL プログラムのサンプルを以下に示します。

ソース・データ・セットに対して更新が頻繁に適用され、リフレッシュ・プログラムの実行中にそれらの更新が適用される可能性がある場合、それは、ソース・データ・セットが UMT によって正確に反映されないことを意味している可能性があります。なぜなら、処理されているレコードまたは既に処理されたレコードが変更される可能性があるからです。これは、プログラムが、レコードの変更の可能性に対して寛容でなければならないことを意味します。プログラムはまた、UMT 自体が他のプログラムによって更新される可能性についても容認するように作成されています。ただし、このような方法での運用は推奨されていません (つまり、UMT を更新する唯一のプログラムは、リフレッシュ・プログラムであるべきです)。

サンプル・プログラムの作動方法

最初に、環境が初期化されます。UMT ファイルがローカルであり、既にオープンしていることが確認されます。UMT ファイルがリモートの場合、プログラムはメッセージを発行して終了します。UMT ファイルがオープンしていない場合、プログラムは UMT ファイルをオープンして終了します (この理由は、UMT をオープンすると、追加の処理を実行せずに、ソース・データ・セットから最新データがロードされるからです)。さらに、ソース・ファイルがローカルかどうかのチェックも行われます。リモートの場合、プログラムはメッセージを発行して終了します。UMT のソース・データ・セットに直接アクセスするファイルがオープンされます。次に、プログラムがそれらの両方のファイルを順次ステップスルーできるように、両方のファイルでブラウズの開始操作が実行されます。

環境がエラーなしでセットアップされると、UMT の更新が開始されます。この作業では、UMT のレコードと基本データ・セットのレコードから成るレコード・ペアの取り出しと比較が実行されます。

取り出されたレコードは以下のように比較されます。

- レコードが等しい場合、UMT とデータ・セットから次のレコードを読み取るためのフラグが設定されます。
- UMT がデータ・セットより大きいキーを持っている場合は、UMT に追加しなければならないレコードがデータ・セット内にあります。
- データ・セットが UMT より大きいキーを持っている場合は、除去しなければならない余分なレコードが UMT 内にあります。
- キーは等しいが、レコードが異なる場合は、データ・セット内のレコードで UMT を更新する必要があります。

UMT にレコードを追加する必要がある場合は、書き込み操作が実行されます。

- 書き込み操作が成功すると、プログラムは、次のレコード・ペアの処理に進みます。

- プログラムによって行われた読み取り操作と書き込み操作の間に別のトランザクションによって挿入されたレコードが原因で書き込み操作が失敗すると、そのレコードを削除し、再書き込みが試行されます。
- 2 度目の試行が失敗すると、プログラムは次のレコード・ペアを処理します。
- 次のレコード・ペアが処理されると、現在の UMT レコードが、データ・セット内の次のレコードと比較され、さらなる UMT レコードの脱落がないかチェックが行われます。

レコードを UMT から削除する必要がある場合は、削除操作が実行されます。

- 削除操作が成功すると、プログラムは次のレコード・ペアの処理に進みます。
- 読み取り操作と削除操作の間にレコードが既に削除されていたために削除操作が失敗すると、プログラムは次のレコード・ペアの処理に進みます。
- 次のレコード・ペアが処理されると、現在のデータ・セット・レコードが、UMT 内の次のレコードと比較されて、UMT 内にあってはならないレコードか他にないかチェックが行われます。

UMT 内でレコードを更新する必要がある場合、更新のための読み取り操作が実行されて、そのレコードがロックされます。

- これが成功すると、更新されたレコードが UMT に再書き込みされ、プログラムは次のレコード・ペアの処理に進みます。
- 別のトランザクションがレコードを削除していたためにこの操作が失敗すると、そのレコードを元に戻すための書き込み操作が実行されます。
- 書き込み操作が失敗すると、プログラムは次のレコード・ペアの処理に進みます。
- 次のレコード・ペアが処理されると、UMT とデータ・セットの両方から次のレコードが読み取られます。

両方のファイルの終わりに達し、もう処理するレコードが残っていないと、プログラムは、データ・セットと UMT の両方でブラウズの終了を実行して戻ります。このサンプルは、データ・セットに直接アクセスするファイルをクローズしないことに留意してください。データ・セットが共用環境で更新のために作動できない場合は、そのデータ・セットにアクセスするファイルを CLOSED DISABLED に設定して、更新を可能にする必要があります。

プログラムは、予期しないエラーをすべてトラップし、画面にエラー・メッセージを発行します。UMT に対する最初の操作のみがチェックされます (削除操作、書き込み操作、または読み取り/再書き込み操作のいずれか)。それが失敗し、レコードが最初に読み取られた後で変更されていることが原因の可能性がある戻りコードが返された場合は、最後に 1 回、そのレコードの訂正が試行されますが、この試行はチェックされません。これは、プログラムがループ状態に入ることを避けるためです。

コード内に追加のコメントがあります。

サンプル・プログラムのセットアップと実行

サンプルに含まれているコメントに従って、更新するレコードのフォーマットに一致するようにプログラムを編集します。

ファイル定義に一致するように、「UMTNAME」と「SOURCEDS」を名前変更します。

COBOL コンパイラーを使用して、プログラムの変換、コンパイル、およびリンクを実行します。

プログラムを CICS に定義し、トランザクションをプログラムに定義します。UMT を指すようにファイル (UMTNAME) を定義し、最初にオープンした時のロード元であるソース・データ・セットを指定します。その他のファイル (SOURCEDS) は、UMT のロード元として定義されているソース・データ・セットを直接指すように定義します。

各シスプレックスには、リフレッシュされる UMT がある CICS 領域が 1 つあります。これらの領域に、トランザクションのリフレッシュを実行する必要があった定義をインストールする必要があります。シスプレックス内のその他のすべての領域では、UMT はリモート・ファイルとして定義されており、UMT が所有する領域内の UMT を指している必要があります。UMT がリモートとして定義されている領域ではトランザクションのリフレッシュを実行する必要はありません。

使用される更新戦略は、ソース・データ・セットのセットアップ方法によって決まります。ソース・データ・セットが RLS としてセットアップされている場合は、すべての UMT を同時にリフレッシュできます。ソース・データ・セットへの更新も適用できます。複数のシステムが 1 度に読み取れるように、データ・セットに SHAREOPTIONS が設定されている場合は、RLS と同様に、同時リフレッシュを実行できます。そうでない場合、ソース・データ・セットが更新される時、リフレッシュのためのソース・データ・セットの読み取りに使用されるファイルは、更新期間中、各システムでクローズし、使用不可にする必要があります。すべての UMT が順次リフレッシュされる場合は、更新のために必要な時に、UMT が所有する領域に対してソース・データ・セットを順番にオープンしてクローズすることができます。

複製されたユーザー保守データ・テーブルをリフレッシュするためのサンプル・プログラムのソース・コード

このソース・コードは、CICS サンプル・ライブラリーでは提供されず、この文書でのみ提供されます。

複製された UMT をリフレッシュするためのサンプル・プログラム: CBL XOPTS(SP)

プログラム名
: UMTUPDT COBOL

記述名
: データ・セットの現在の内容で UMT を動的に更新するための CICS アプリケーション

概説
: このプログラムは、ユーザー保守テーブル (UMT) を、1 つ (または複数の) CICS システムで使用したまま、オープンされた時のロード元であったソース・データ・セット内のデータに一致するように更新する方法を示します。このプログラムを使用して、さまざまなシスプレックス内で複製されている

UMT を更新し、それらがすべてソース・データ・セットに一致するようにすることができます。このプログラムは、FOR で実行する必要があります。

要件　このプログラムは、CICS COBOL プログラムとして変換、コンパイル、およびリンクされ、CICS に定義されている必要があります。トランザクション名をこのプログラムに定義する必要があります。UMT ファイル (現在、UMTNAME と呼ばれている) は、UMT にアクセスするために使用され、ソース・データ・セット・ファイル (現在、SOURCEDS と呼ばれている) は、UMT のロード元であるデータ・セットに直接アクセスするために使用されます。これらの定義は、UMT がある領域 (FOR) にのみインストールする必要があります。UMT をリモート側で使用している、同じシスプレックス内の領域は、更新処理を実行する必要はありません。

説明　プログラムは、必要な 2 つのファイルを最初に初期化し、先頭からそれらの参照を開始します。オープンしていない場合、UMT をオープンするとロードされます。オープンしておらず、UMT がロードされると、プログラムの操作は事実上重複となり、更新コードは実行されません。プログラムは、リモート・システム名についてのチェックも行います。いずれかのファイルで存在する場合、プログラムは実行されません。これは、明らかにパフォーマンスを低下させる機能シップの発生を予防するためです。

プログラムは、これらの 2 つのファイルからレコード・ペアを継続的に読み取ってそれらを比較し、ソース・データ・セットに一致しない UMT 内のレコードを追加、削除、または更新します。

レコード・ペアのキーが比較されます。UMT のキーとソース・データ・セットのキーが等しく、それらのレコードが一致する場合、更新は必要ありません。両方のキーは等しいが、レコードが異なる場合は、ソース・データ・セット内のレコードが UMT の更新に使用されます。UMT 内のキーがソース・データ・セット内のキーより大きい場合は、それらのキーが等しくなるか、UMT のキーがソース・データ・セットのキーより小さくなるまで、ソース・データ・セット内のレコードが UMT に書き込まれます。UMT のキーがソース・データ・セットのキーより小さい場合は、それらのキーが等しくなるか、UMT のキーがソース・データ・セットのキーより大きくなるまで、UMT 内のレコードが除去されます。これは、両方のファイルの終わりに達するか、予期しないエラーが発生するまで続行されます。

予期しないエラーはすべて画面に報告され、プログラムの操作が停止されます。一部のエラーはトラップされ、UMT を更新するためにさらなる試行が行われます。この試行が失敗すると、それらのレコードに対して追加のアクションは実行されず、プログラムは次のペアの処理に進みます。

プログラムの変更

このプログラムは、現状のままでは機能しない可能性があります。使用されているレコード構造は、40 文字のレコードにアクセスするために 4 文字のキーが使用されることを想定しています。このプログラムがさまざまなレコード・タイプを処理できるようにするには、以下の変更を行う必要があります。

UMT とソース・データ・セットにアクセスするキーを変更する必要があります。このキーを保管する変数は UMT-KEY と DS-KEY です。

レコードの長さは、UMT-LEN と DS-LEN に保持されます。

UMT とソース・データ・セットのレコード変数を変更する必要があります。これらを保管する変数は、UMT-REC (UMT-REC-KEY と UMT-REC-TEXT を含む) および DS-REC (DS-REC-KEY と DS-REC-TEXT を含む) です。必要に応じて、明らかに追加フィールドを追加できます。

UMT のファイル名は UMTNAME と設定されています。これは、既に定義されている UMT に一致するように変更できます。ソース・データ・セット・ファイルは SOURCEDS と設定されており、これも変更可能です。

ソース・コード

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID. UMTUPDT.

 ENVIRONMENT DIVISION.
     EJECT.

 DATA DIVISION.

 WORKING-STORAGE SECTION.

* Declare the UMT and DS record variables
 77 UMT-KEY          PIC X(4)  VALUE '0000'.
 77 UMT-LEN          PIC 9(2)  VALUE 40.
 01  UMT-REC.
   03  UMT-REC-KEY   PIC X(4)  VALUE SPACES.
   03  UMT-REC-TEXT  PIC X(36) VALUE SPACES.

 77 DS-KEY           PIC X(4)  VALUE '0000'.
 77 DS-LEN           PIC 9(2)  VALUE 40.
 01  DS-REC.
   03  DS-REC-KEY    PIC X(4)  VALUE SPACES.
   03  DS-REC-TEXT   PIC X(36) VALUE SPACES.

* Declare other work variables
* Screen output strings
 01  MESSAGE-OUTPUT  PIC X(26) VALUE 'UMT SUCCESSFULLY REFRESHED'.
 01  REMOTE-OUTPUT   PIC X(25) VALUE 'FILE RESOURCE NOT LOCAL'.
 01  ERROR-OUTPUT.
   03  ERROR-OPNAME  PIC X(8)  VALUE SPACES.
   03  FILLER        PIC X(15) VALUE ' RETURNED RESP '.
   03  ERROR-RESP    PIC X(8)  VALUE SPACES.
   03  FILLER        PIC X(7)  VALUE ' RESP2 '.
   03  ERROR-RESP2   PIC X(8)  VALUE SPACES.
   03  FILLER        PIC X(10) VALUE ' FOR FILE '.
   03  ERROR-FILE    PIC X(8)  VALUE SPACES.

* End of file flags
 77  UMT-EOF         PIC 9(1)  VALUE 0.
 77  DS-EOF          PIC 9(1)  VALUE 0.

* Record retrieval flags
 77  GET-NEXT-UMT    PIC 9(1)  VALUE 1.
 77  GET-NEXT-DS     PIC 9(1)  VALUE 1.

* File inquire variables
 77  REM-SYS-NAME    PIC X(4)  VALUE SPACES.
 77  OPEN-STAT       PIC S9(8) BINARY.

* Program operation flags
 77  PROCESS-FILES   PIC 9(1)  VALUE 1.
 77  REM-FILE        PIC 9(1)  VALUE 0.
 77  UMT-STARTBR     PIC 9(1)  VALUE 0.
 77  DS-STARTBR      PIC 9(1)  VALUE 0.
```

```
* EXEC CICS response variables
 77  RESPONSE        PIC S9(8) BINARY.
 77  RESPONSE2       PIC S9(8) BINARY.


 COPY DFHAID.
 COPY DFHBMSCA.


 LINKAGE SECTION.
     EJECT.

 PROCEDURE DIVISION USING DFHEIBLK.

*****************************************************************
* Main processing starts here.                                  *
*****************************************************************
 MAIN-PROCESSING SECTION.

* Check the UMT and data set for processing
     PERFORM FILE-CHECK.

* If the file check completed okay, process the UMT
     IF (PROCESS-FILES = 1)

* Ready the UMT and DS for access
        PERFORM INITIALIZE

* Call the update routine until the end of both files reached
        PERFORM UPDATE-UMT UNTIL (DS-EOF = 1 AND UMT-EOF = 1)

     END-IF.

* Exit the program cleanly
     PERFORM TRAN-FINISH.

 MAIN-PROCESSING-EXIT.
     GOBACK.
     EJECT


*****************************************************************
* Procedures start here.                                        *
*****************************************************************
*****************************************************************
* Check the files open status and that they aren't remote       *
*****************************************************************
 FILE-CHECK SECTION.

* Inquire on the UMT to get remote and open status information
     MOVE SPACES TO REM-SYS-NAME.
     EXEC CICS INQUIRE FILE('UMTNAME')
          OPENSTATUS(OPEN-STAT)
          REMOTESYSTEM(REM-SYS-NAME)
          RESP(RESPONSE)
          RESP2(RESPONSE2)
     END-EXEC.
* Output an error if inquire on the UMT failed
     IF (RESPONSE NOT = DFHRESP(NORMAL))
        MOVE 'INQUIRE ' TO ERROR-OPNAME
        MOVE 'UMTNAME ' TO ERROR-FILE
        PERFORM PROCESS-ERROR
     END-IF.
* System name is not blank if the file is defined as remote
* We don't want to do any processing if the file is remote
     IF (REM-SYS-NAME NOT = SPACES)
        MOVE 0 TO PROCESS-FILES
```

```
        MOVE 1 TO REM-FILE
     ELSE
* If the UMT is not open, then opening it will update it
        IF (OPEN-STAT NOT = DFHVALUE(OPEN))
           EXEC CICS SET FILE('UMTNAME')
                OPEN
                RESP(RESPONSE)
                RESP2(RESPONSE2)
           END-EXEC
* Check open of UMT was successful
           IF (RESPONSE NOT = DFHRESP(NORMAL))
              MOVE 'OPEN    ' TO ERROR-OPNAME
              MOVE 'UMTNAME ' TO ERROR-FILE
              PERFORM PROCESS-ERROR
           ELSE
* Don't want to do any processing, as open will update UMT
              MOVE 0 TO PROCESS-FILES
           END-IF
        END-IF
     END-IF.

* Inquire on the source data set to get remote and open status
     MOVE SPACES TO REM-SYS-NAME.
     EXEC CICS INQUIRE FILE('SOURCEDS')
          REMOTESYSTEM(REM-SYS-NAME)
          OPENSTATUS(OPEN-STAT)
          RESP(RESPONSE)
          RESP2(RESPONSE2)
     END-EXEC.
* Output an error if inquire on the data set failed
     IF (RESPONSE NOT = DFHRESP(NORMAL))
        MOVE 'INQUIRE ' TO ERROR-OPNAME
        MOVE 'SOURCEDS' TO ERROR-FILE
        PERFORM PROCESS-ERROR
     END-IF.
* Don't do any processing if it's a remote file
     IF (REM-SYS-NAME NOT = SPACES)
        MOVE 0 TO PROCESS-FILES
        MOVE 1 TO REM-FILE
     ELSE
* Open the source data set
        IF (OPEN-STAT = DFHVALUE(CLOSED))
           EXEC CICS SET FILE('SOURCEDS')
                OPEN
                RESP(RESPONSE)
                RESP2(RESPONSE2)
           END-EXEC
* Check open of data set was successful
           IF (RESPONSE NOT = DFHRESP(NORMAL))
              MOVE 'OPEN    ' TO ERROR-OPNAME
              MOVE 'SOURCEDS' TO ERROR-FILE
              PERFORM PROCESS-ERROR
           END-IF
        END-IF
     END-IF.

 FILE-CHECK-EXIT.
     EXIT.
     EJECT



*****************************************************************
* Initialize the files ready for sequential reading             *
*****************************************************************
 INITIALIZE SECTION.
```

```
* Start browsing the UMT from the first record
     EXEC CICS STARTBR FILE('UMTNAME')
          RIDFLD(UMT-KEY)
          GTEQ
          RESP(RESPONSE)
          RESP2(RESPONSE2)
     END-EXEC.
* If UMT is empty (NOTFND) then treat as end of UMT and fill
     IF (RESPONSE = DFHRESP(NOTFND))
        MOVE 1 TO UMT-EOF
     ELSE
* Output an error if the start browse for the UMT failed
        IF (RESPONSE NOT = DFHRESP(NORMAL))
           MOVE 'STARTBR ' TO ERROR-OPNAME
           MOVE 'UMTNAME ' TO ERROR-FILE
           PERFORM PROCESS-ERROR
        END-IF
     END-IF.
* Set UMT start browse flag
     MOVE 1 TO UMT-STARTBR.

* Start browsing the data set from the first record
     EXEC CICS STARTBR FILE('SOURCEDS')
          RIDFLD(DS-KEY)
          GTEQ
          RESP(RESPONSE)
          RESP2(RESPONSE2)
     END-EXEC.
* If data set is empty then treat as end of data set an empty UMT
     IF (RESPONSE = DFHRESP(NOTFND))
        MOVE 1 TO DS-EOF
     ELSE
* Output an error if the start browse for the data set failed
        IF (RESPONSE NOT = DFHRESP(NORMAL))
           MOVE 'STARTBR ' TO ERROR-OPNAME
           MOVE 'SOURCEDS' TO ERROR-FILE
           PERFORM PROCESS-ERROR
        END-IF
     END-IF.
* Set data set start browse flag
     MOVE 1 TO DS-STARTBR.

 INITIALIZE-EXIT.
     EXIT.
     EJECT



*****************************************************************
* Update the UMT according to the record/key states             *
*****************************************************************
 UPDATE-UMT SECTION.

* Get the next records from the UMT and data set
     PERFORM READ-FILES.

* If both records are the same, move to the next record
     IF UMT-REC = DS-REC
        MOVE 1 TO GET-NEXT-UMT
        MOVE 1 TO GET-NEXT-DS
     ELSE
* If UMT is behind data set then extra record in UMT so delete it.
* Also delete records from UMT if EOF DS reached before EOF UMT
        IF (UMT-EOF = 0 AND (UMT-KEY < DS-KEY OR DS-EOF = 1))
           PERFORM UMT-DELETE
        END-IF
```

```
* If UMT ahead of data set then extra record in DS so add to UMT
* Also add records to the UMT if the EOF reached before EOF DS
        IF (DS-EOF = 0 AND (UMT-KEY > DS-KEY OR UMT-EOF = 1))
           PERFORM UMT-WRITE
        END-IF

* If both keys equal but record different, update UMT
        IF ((DS-EOF = 0 AND UMT-EOF = 0) AND UMT-KEY = DS-KEY)
           PERFORM UMT-UPDATE
        END-IF

     END-IF.


 UPDATE-UMT-EXIT.
     EXIT.
     EJECT


*****************************************************************
* Read the next record from both files                          *
*****************************************************************
 READ-FILES SECTION.

* If the flags are set to read the next UMT record, do so
     IF (GET-NEXT-UMT = 1 AND UMT-EOF = 0)
        MOVE SPACES TO UMT-REC
        EXEC CICS READNEXT FILE('UMTNAME')
             RIDFLD(UMT-KEY)
             INTO(UMT-REC)
             RESP(RESPONSE)
             RESP2(RESPONSE2)
        END-EXEC
* Set the EOF flag if the end of the UMT has been reached
        IF (RESPONSE = DFHRESP(ENDFILE))
              MOVE 1 TO UMT-EOF
        ELSE
* Output an error if the return code from the READ is unexpected
           IF (RESPONSE NOT = DFHRESP(DUPKEY) AND
              RESPONSE NOT = DFHRESP(NORMAL))
                 MOVE 'READNEXT' TO ERROR-OPNAME
                 MOVE 'UMTNAME ' TO ERROR-FILE
                 PERFORM PROCESS-ERROR
           END-IF
        END-IF
     END-IF.
* If the flags are set to read the next data set record, do so
     IF (GET-NEXT-DS = 1 AND DS-EOF = 0)
        MOVE SPACES TO DS-REC
        EXEC CICS READNEXT FILE('SOURCEDS')
             RIDFLD(DS-KEY)
             INTO(DS-REC)
             RESP(RESPONSE)
             RESP2(RESPONSE2)
        END-EXEC
* Set the EOF flag if the end of the data set has been reached
        IF (RESPONSE = DFHRESP(ENDFILE))
              MOVE 1 TO DS-EOF
        ELSE
* Output an error if the return code from the READ is unexpected
           IF (RESPONSE NOT = DFHRESP(DUPKEY) AND
              RESPONSE NOT = DFHRESP(NORMAL))
                 MOVE 'READNEXT' TO ERROR-OPNAME
                 MOVE 'SOURCEDS' TO ERROR-FILE
                 PERFORM PROCESS-ERROR
           END-IF
```

```
        END-IF
     END-IF.

 READ-FILES-EXIT.
     EXIT.
     EJECT

*****************************************************************
* Attempt to delete a record from the UMT                       *
*****************************************************************
 UMT-DELETE SECTION.

* Delete the last read record in the UMT
     EXEC CICS DELETE FILE('UMTNAME')
          RIDFLD(UMT-KEY)
          RESP(RESPONSE)
          RESP2(RESPONSE2)
     END-EXEC.
* Allow NORMAL and NOTFND return codes in case record has been
* deleted since it was first read, otherwise output an error
     IF (RESPONSE = DFHRESP(NORMAL) OR
       RESPONSE = DFHRESP(NOTFND))
* Set flags to get next UMT record, but keep same data set record
        MOVE 1 TO GET-NEXT-UMT
        MOVE 0 TO GET-NEXT-DS
     ELSE
        MOVE 'DELETE  ' TO ERROR-OPNAME
        MOVE 'UMTNAME ' TO ERROR-FILE
        PERFORM PROCESS-ERROR
     END-IF.

 UMT-DELETE-EXIT.
     EXIT.
     EJECT

*****************************************************************
* Attempt to write a record to the UMT                          *
*****************************************************************
 UMT-WRITE SECTION.

* Attempt to write the missing record using the data set key
     EXEC CICS WRITE FILE('UMTNAME')
          RIDFLD(DS-KEY)
          FROM(DS-REC)
          RESP(RESPONSE)
          RESP2(RESPONSE2)
     END-EXEC.
* If the UMT has had a record written to this position since the
* read then delete it and try one last time.
* If write still unsuccessful, move to the next pair of records
     IF RESPONSE = DFHRESP(DUPREC)
        EXEC CICS DELETE FILE('UMTNAME')
             RIDFLD(DS-KEY)
             RESP(RESPONSE)
             RESP2(RESPONSE2)
        END-EXEC
        EXEC CICS WRITE FILE('UMTNAME')
             RIDFLD(DS-KEY)
             FROM(DS-REC)
             RESP(RESPONSE)
             RESP2(RESPONSE2)
        END-EXEC
     ELSE
* Output an error if return code from first write was bad
* (but allow suppression return code by user exit)
        IF (RESPONSE NOT = DFHRESP(NORMAL) AND
           RESPONSE NOT = DFHRESP(SUPPRESSED))
              MOVE 'UMTNAME ' TO ERROR-FILE
```

```
              MOVE 'WRITE   ' TO ERROR-OPNAME
              PERFORM PROCESS-ERROR
        END-IF
     END-IF.

* Set flags to keep same UMT record, and get next data set record
     MOVE 0 TO GET-NEXT-UMT.
     MOVE 1 TO GET-NEXT-DS.

 UMT-WRITE-EXIT.
     EXIT.
     EJECT

*****************************************************************
* Attempt to update a record in the UMT to match the DS         *
*****************************************************************
 UMT-UPDATE SECTION.

* Attempt to get a lock on the record using read for update
     EXEC CICS READ FILE('UMTNAME')
          RIDFLD(UMT-KEY)
          INTO(UMT-REC)
          UPDATE
          RESP(RESPONSE)
          RESP2(RESPONSE2)
     END-EXEC.
* If record has been deleted since original read, write it.
* If write is unsuccessful, move to next pair of records
     IF RESPONSE = DFHRESP(NOTFND)
        EXEC CICS WRITE FILE('UMTNAME')
             RIDFLD(UMT-KEY)
             FROM(DS-REC)
             RESP(RESPONSE)
             RESP2(RESPONSE2)
        END-EXEC
     ELSE
* If read for update was successful, write data set record to UMT
        IF RESPONSE = DFHRESP(NORMAL)
           EXEC CICS REWRITE FILE('UMTNAME')
                FROM(DS-REC)
                RESP(RESPONSE)
                RESP2(RESPONSE2)
           END-EXEC
* Output an error if rewrite failed
           IF RESPONSE NOT = DFHRESP(NORMAL)
              MOVE 'REWRITE ' TO ERROR-OPNAME
              MOVE 'UMTNAME ' TO ERROR-FILE
              PERFORM PROCESS-ERROR
           END-IF
        ELSE
* Output an error if the read for update failed
           MOVE 'READUPDT' TO ERROR-OPNAME
           MOVE 'UMTNAME ' TO ERROR-FILE
           PERFORM PROCESS-ERROR
        END-IF
     END-IF.

* Set flags to get next record for both UMT and data set
     MOVE 1 TO GET-NEXT-UMT.
     MOVE 1 TO GET-NEXT-DS.

 UMT-UPDATE-EXIT.
     EXIT.
     EJECT
```

```
****************************************************************
* Exit from the program cleanly                                *
****************************************************************
 TRAN-FINISH SECTION.

* End the browse operation for the UMT
     IF (UMT-STARTBR = 1)
        EXEC CICS ENDBR FILE('UMTNAME')
             RESP(RESPONSE)
             RESP2(RESPONSE2)
        END-EXEC
     END-IF.

* End the browse operation for the data set
     IF (DS-STARTBR = 1)
        EXEC CICS ENDBR FILE('SOURCEDS')
             RESP(RESPONSE)
             RESP2(RESPONSE2)
        END-EXEC
     END-IF

* Output a message to the screen if UMT was updated
     IF (REM-FILE = 0)
        EXEC CICS SEND TEXT
             FROM(MESSAGE-OUTPUT)
             ERASE
             RESP(RESPONSE)
             RESP2(RESPONSE2)
        END-EXEC
     ELSE
* Output a message if either file was defined as remote
        EXEC CICS SEND TEXT
             FROM(REMOTE-OUTPUT)
             ERASE
             RESP(RESPONSE)
             RESP2(RESPONSE2)
        END-EXEC
     END-IF.


* End the program and return to CICS
     EXEC CICS RETURN
     END-EXEC.

 TRAN-FINISH-EXIT.
     EXIT.
     EJECT
****************************************************************
* Display error message on screen and exit program             *
****************************************************************
 PROCESS-ERROR SECTION.

* Copy last return codes into the message
     MOVE RESPONSE TO ERROR-RESP.
     MOVE RESPONSE2 TO ERROR-RESP2.

* Output message to the screen
     EXEC CICS SEND TEXT
          FROM(ERROR-OUTPUT)
          ERASE
          RESP(RESPONSE)
          RESP2(RESPONSE2)
     END-EXEC.

* End the program and return to CICS
     EXEC CICS RETURN
```

```
    END-EXEC.

PROCESS-ERROR-EXIT.
    EXIT.
```

第 6 章 データ・テーブルのトラブルシューティング

CICS によって生成されたトレース情報とダンプ情報を使用すると、共用データ・テーブルでの問題の原因を判別するために役立てることができます。

共用データ・テーブルによって生成される診断メッセージと異常終了コードの説明はCICS メッセージに記載されています。

データ・テーブル・サービスのトレース情報

CICS によって生成されるトレース・テーブルは、問題の原因判別に役立ちます。ここには、CICS モジュールによる制御の流れが示されます。ここで説明するエントリーは、データ・テーブル・サービスによってトレース・テーブルに含まれます。

トレース・テーブルの内容について、およびその内容の取得方法については、CICS トレースの使用を参照してください。

トレース・ポイントには以下の 2 つのタイプがあります。

- 共用データ・テーブル・サポートによって提供される、各サービスの入り口トレース・ポイントと出口トレース・ポイント。これらのトレース・ポイントを取得するには、ファイル制御レベル 2 トレースを有効にする必要があります。
- 例外トレース・ポイント。

共用データ・テーブルの入り口トレース・ポイントと出口トレース・ポイント

以下の入り口トレース・ポイントと出口トレース・ポイントが共用データ・テーブル・サービスによって提供されます。

0B13 リモート読み取りサービスへの入り口

0B14 リモート読み取りサービスからの出口

0B1B データ・テーブル・サポート初期設定サービスへの入り口

0B1C データ・テーブル・サポート初期設定サービスからの出口

0B1D ログオン・サービスへの入り口

0B1E ログオン・サービスからの出口

0B1F ロード・サービスへの入り口

0B20 ロード・サービスからの出口

0B21 使用可能化および統計のオープン、クローズ、設定サービスへの入り口

0B22 使用可能化および統計のオープン、クローズ、設定サービスからの出口

0B23 ローカル読み取りサービスへの入り口

0B24 ローカル読み取りサービスからの出口

0B25 更新 (レコード追加、追加、置換、削除) サービスへの入り口

© Copyright IBM Corp. 2005, 2018

0B26 更新サービスからの出口

0B2D 接続および切断サービスへの入り口

0B2E 接続および切断サービスからの出口

これらの各トレース・ポイントのフォーマットは、CICS トレースの使用で説明されています。

共用データ・テーブルの機能および修飾子フラグ

共用データ・テーブルでは、各入り口トレース・ポイントと出口トレース・ポイントには機能フィールドが含まれており、ほとんどのトレース・ポイントには修飾子フラグ・フィールドが含まれています。機能フィールドは、実行されていた機能を識別するバイトです。修飾子フラグ・フィールドは、一部の機能を修飾するフラグを含むバイトです。

これらのフィールドの値を以下に示します。

表 *6.* 機能および修飾子フラグと値

機能	**Qualifier flags**
X'00': 初期設定	X'00': 共用データ・テーブル・サーバーとして X'80': 共用データ・テーブル・リクエスターとして
X'02': ソースからのエントリーの追加	X'00': テーブル読み取り要求へのデータ・セットの結果として発行された追加 X'40': ロード・トランザクションによって発行された追加
X'03': テーブルへのエントリーの書き込み	X'00': 完了済み書き込み X'80': CMT の事前書き込み
X'04': テーブルへのエントリーの再書き込み	X'00': 完了済み再書き込み X'80': CMT の事前再書き込み
X'05': テーブルのエントリーの削除	X'00': 完了済み削除 X'80': CMT の事前削除
X'06': この作業単位によって行われたユーザー保守データ・テーブル更新のコミット	
X'07': この作業単位によって行われたユーザー保守データ・テーブル更新のロールバック	
X'08': データ・テーブルのロード (出口トレースのみ)	X'00': ロードは OK X'80': ソース・ファイルが空

表 6. 機能および修飾子フラグと値 (続き)

機能	**Qualifier flags**
X'09': レコードにおけるポイント	X'80': 「等しい」の一致 X'40': 「より大きい」の一致 X'20': 「より小さい」の一致 (上記はさまざまな組み合わせが可能) X'10': データ・テーブルが使用可能かどうかのテスト
X'0A': キーによるレコードの取得	X'80': 「等しい」の一致 X'40': 「より大きい」の一致 X'20': 「より小さい」の一致 (上記はさまざまな組み合わせが可能) X'10': データ・テーブルが使用可能かどうかのテスト
X'0B': トークンによるレコードの取得	X'80': 「等しい」の一致 (レコード・シーケンスの内部高速パス) X'40': 「より大きい」の一致 X'20': 「より小さい」の一致 (上記はさまざまな組み合わせが可能) X'10': データ・テーブルが使用可能かどうかのテスト
X'0C': サーバーとしてログオン	
X'0E': データ・テーブルのオープン	
X'0F': データ・テーブルのクローズ	
X'10' 統計の収集	
X'11': 使用可能化状態の設定	X'00': データ・テーブルを使用可能にする X'80': データ・テーブルを使用不可にする X'40': 使用不可化を強制 (常に使用不可化と組み合わせて使用)
X'15': 共用データ・テーブルへの接続	
X'16': 共用データ・テーブルとの接続の切断	
X'17': ロードの完了の処理	

共用データ・テーブルの応答コード

共用データ・テーブルの各出口トレース・ポイントには、2 バイトの応答コードと理由コードのフィールドが含まれます。

最初のバイトは応答コードです。考えられる値は以下のとおりです。

X'01' 成功

X'02' Exception

X'03' Disaster

X'04' 無効

X'06' Purged

共用データ・テーブルの理由コード

共用データ・テーブルの各出口トレース・ポイントには、2 バイトの応答コードと理由コードのフィールドが含まれます。

2 番目のバイトは理由コードです。考えられる値は以下のとおりです。この理由コードは、エラー・コード情報を伴う場合があります。エラー・コードは 4 バイト・フィールドであり、エラー・メッセージや例外トレース・ポイントでも報告されます。考えられる値は、CICS メッセージ、 76 ページの『データ・テーブル SVC からのエラーの分析』、および 79 ページの『データ・テーブルのクロスメモリー・サービスからのエラーの分析』で説明されています。

X'01' レコードがデータ・テーブルにない

X'02' 重複 (レコードが既にデータ・テーブルにある)

X'03' データ・テーブルが満杯 (既に最大数のレコードが含まれている)

X'04' レコードがユーザー出口によって拒否された

X'05' ストレージを取得できなかった

X'06' レコードがデータ・テーブルにない (テーブルは完成済みであることがわかっている)

X'07' データ・テーブルのサービスが失敗した

X'08' ファイルへの接続が許可されていない

X'09' リソースがデータ・テーブルにない

X'0A' リモート・システムがサーバーとしてログオンしていない

X'0B' ロード要求が失敗した

X'0C' データ・テーブルが使用不可である

X'0D' (DASD からの) 追加要求が意図的に処理されていない

X'0E' レコードが長すぎる

X'0F' データ・テーブルのトークンが無効

X'10' レコードがデータ・テーブルにない (しかし、ソース・データ・セット内にある可能性がある)

X'11' データ・テーブルが、他のファイルが引き続きそれを使用しているためにクローズされていない

X'12' 予約

X'13' レコードはデータ・テーブルにあるが、現在有効でない

X'14' ファイルは使用不可であるためクローズできない

X'15' Protocol error

X'16' CICS が MVS サブシステムでない

X'17' このファイルへの接続が許可されていない

X'18' CICS がクロスメモリー・サービスを使用できない

X'19' インターフェース・パラメーター・ブロックのフォーマットが認識されない

共用データ・テーブルの UMT および他のフラグ

このフラグ・バイトは OPEN 時にエントリー・トレース・ポイントに含まれています。

オープン時の有効ビットは以下のとおりです。

B'1.....'
CICS 保守データ・テーブル

B'01.....'
リカバリー可能ユーザー保守データ・テーブル

B'00.....'
リカバリー不能ユーザー保守データ・テーブル

共用データ・テーブルの例外トレース・ポイント

以下の例外トレース・ポイントが共用データ・テーブル・サービスによって提供されます。

AP 0B0A
DFHDTRE への呼び出しで認識されない機能

AP 0B0B
DFHDTRR への呼び出しで認識されない機能

AP 0B0C
DFHDTUP への呼び出しで認識されない機能

AP 0B0D
DFHDTST への呼び出しで認識されない機能

AP 0B0E
DFHDTSS への呼び出しで認識されない機能

AP 0B0F
DFHDTRC への呼び出しで認識されない機能

AP 0B10
レコード管理の初期設定時のエラー

AP 0B11
レコード・マネージャー OPEN 時のエラー

AP 0B12
レコード・マネージャー CLOSE 時のエラー

AP 0B15
取得 PC への呼び出し時の予期しないエラー

AP 0B19
サーバーとして初期設定する際のデータ・テーブル SVC の呼び出しでのエラー

AP 0B1A
リクエスターとして初期設定する際のデータ・テーブル SVC の呼び出しでのエラー

AP 0B27
CLOSE がテーブル・ブロックを検出できなかった

AP 0B28
CLOSE がファイル・ブロックを検出できなかった

AP 0B29
サーバーとしてログオンする際のデータ・テーブル SVC の呼び出しでのエラー

AP 0B2A
接続または切断する際のデータ・テーブル SVC の呼び出しでのエラー

AP 0B2B
XDTRD 出口が無効な長さのレコードを返した (つまり、CMT の長さを変更したか、UMT の長さを増やした)

AP 0B2C
接続索引がサポートされている最大サイズを超えた

AP 0B2F
ロード・トランザクションにパラメーターを渡すためのストレージを獲得する際の致命的なエラー

これらの各トレース・ポイントのフォーマットは、CICS トレースの使用で説明されています。

データ・テーブル SVC からのエラーの分析

データ・テーブル SVC への呼び出しから発生したエラーの後には、常に例外トレース・ポイント (エラーの理由を識別するエラー・コード・フィールドを含む) が作成されます。これらのトレース・ポイントは AP 0B12、0B19、0B1A、0B29、および 0B2A です。

SVC エラーには以下の 3 つのカテゴリーがあります。

1. 発生すると予想される条件。例えば、接続を試みているリモート・ファイルがデータ・テーブルではない場合や、リモート・システムが共用データ・テーブル・サーバーとしてログオンしていない場合です。このような条件では CICS が適切なアクションを実行するため、診断情報は不要です。
2. 環境内の問題によって引き起こされた可能性があり、修正可能と思われるエラー。これらのエラーの場合は、エラーの理由コードを示すメッセージが発行されます。理由コードの説明は、CICS メッセージのメッセージの説明に記載されています。
3. 何らかの種類のロジックの問題、またはルーチンの誤用 (保全性検査やセキュリティー検査を回避しようとする可能性があるもの) を示すエラー。これらのエラーは CICS ファイル制御によって破壊的エラーとして扱われ、結果的にシステム・ダンプ (システム・ダンプが有効になっている場合) が生成され、ほとんどの場合は AFCZ ABEND でトランザクションが異常終了します。これらのエラーでは、応答フィールドと理由フィールドの値は通常 X'0215' になります。

以下のトピックでは、3 番目のエラー・カテゴリーのエラー・コードについて説明します。これらのエラー・コードは例外トレース項目内にのみ表示されます。この

エラー・コードのフォーマットは X'ffaaaaaa' です。*ff* は障害のタイプを識別し、*aaaaaa* は一部の障害の場合に提供される追加情報です。各トレース・ポイントで表示される可能性のある *ff* の値については、以下のトピックで説明されています。

すべての共用データ・テーブル・トレース・ポイントの値

以下のエラー・コードが、例外トレース・ポイント 0B12、0B19、0B1A、0B29、および 0B2A で発生する可能性があります。

X'01' 呼び出し元が CICS AFCB (許可機能制御ブロック) を通じて許可される必要がある機能が指定されましたが、その呼び出し元は許可されていませんでした。

X'0A' 呼び出し元が無効な機能コードを渡しました。

X'0B' 呼び出し元が無効なフォーマットの SVC 呼び出しを指定しました。

X'0C' SVC に無効なパラメーター・リスト・アドレスが渡されました。

X'0D' レジスター 1 で渡される値が 0 でなければならない機能が指定されましたが、0 になっていませんでした。追加情報には、渡される値の下位の 3 バイトが含まれます。

X'12' 呼び出し元がキー 0 監視プログラム状態でなければならない機能が指定されましたが、呼び出し元がその状態ではありませんでした。

0B12 トレース・ポイントの値

共用データ・テーブルがクローズされる際、アクセス・リスト・エントリーの追加または削除時に SVC によってエラーが返されると、AP 0B12 例外トレース・ポイントが発行されます。

すべてのトレース・ポイントで発生する可能性があるエラーに加え、以下のような可能性も考えられます。

X'02' 共用データ・テーブル・サービスはまだ初期設定されていません (領域のアンカー・ブロックは作成されていません)。

X'0E' 指定されたデータ・スペース STOKEN が無効であるか、呼び出し元がその使用を許可されていません。

X'0F' CICS 領域がサーバーとして初期設定を完了していません。

X'13' アクセス・リスト・エントリーを削除しようとしましたが、指定されたエントリーがデータ・テーブルの SVC によって作成されていなかったため、失敗しました。

これ以外のエラーではすべて、エラー・コードを含むメッセージが発行されます。

0B19 トレース・ポイントの値

共用データ・テーブル・サーバーとしての初期設定時に、SVC によってエラーが返されると、AP 0B19 例外トレース・ポイントが発行されます。

すべてのトレース・ポイントで発生する可能性があるエラーに加え、以下のような可能性も考えられます。

X'02' CICS 領域で共用データ・テーブルの初期設定が実行される前に (その領域

のアンカー・ブロックがまだ作成されていない時点で) アクセス・リスト・エントリーを追加しようとしています。

X'0E' 指定されたデータ・スペース STOKEN が無効であるか、呼び出し元がその使用を許可されていません。

X'0F' CICS 領域でサーバーとしての初期設定が完了する前にアクセス・リスト・エントリーを追加しようとしました。

これ以外のエラーではすべて、エラー・コードを含むメッセージが発行されます。

0B1A トレース・ポイントの値

共用データ・テーブル・リクエスターとしての初期設定時に、SVC によってエラーが返されると、AP 0B1A 例外トレース・ポイントが発行されます。

すべてのトレース・ポイントで発生する可能性があるエラーに加え、以下のような可能性も考えられます。

X'05' CICS 領域は既に共用データ・テーブル・リクエスターとして初期設定されていますが、現在、当初の初期設定時とは別の要求ブロックのもとで実行されています。

これ以外のエラーではすべて、エラー・コードを含むメッセージが発行されます。

AP 0B29 トレース・ポイントの値

共用データ・テーブル・サーバーとしてのログオン時に、SVC によってエラーが返されると、AP 0B29 例外トレース・ポイントが発行されます。

すべてのトレース・ポイントで発生する可能性があるエラーに加え、以下のような可能性も考えられます。

X'02' サーバーとして登録 (ログオン) しようとしている CICS 領域がまだ初期設定されていません (その領域のアンカー・ブロックが作成されていません)。

X'04' この CICS 領域は既に共用データ・テーブル・サーバーとして登録 (ログオン) しています。

X'0F' CICS 領域ではサーバーとしての初期設定が完了していません。

X'14' AFCS アンカー・ブロックが存在しません。

X'15' CICS セキュリティー・ブロックが存在しません。

X'16' 呼び出し元がユーザー保護キーで実行されていない (その PSW キーが 8 未満である) か、呼び出し元の TCB がユーザー保護キーで正常に実行されていません (TCBPKF が 8 未満です)。

これ以外のエラーではすべて、エラー・コードを含むメッセージが発行されます。

0B2A トレース・ポイントの値

機能コード・フィールドに X'15' が含まれている場合、AP 0B2A 例外トレース・ポイントは CONNECT 時 (つまり、リモート・ファイルへの接続を確立しようとしているとき) のエラーを示します。

すべてのトレース・ポイントで発生する可能性があるエラーに加え、以下のような可能性も考えられます。

X'02' 共用データ・テーブル・サービスはまだ初期設定されていません (領域のアンカー・ブロックは作成されていません)。

X'03' 要求側の領域が、共用データ・テーブル・リクエスターとして初期設定を完了していません。

X'05' CICS 領域が、データ・テーブル・リクエスターとして初期設定されたときと異なる要求ブロック (RB) のもとで実行されています。エラー・コードの追加情報部分に、呼び出しで使用された RB アドレスが含まれています。

X'72' ホーム・アドレス・スペースのセキュリティー・ユーザー ID を見つけるためのユーザー交換可能 DFHACEE モジュールへの LINK が失敗しました。エラー・コードの追加情報部分に、LINK から出された 2 バイトの ABEND コードが含まれています。このエラーに伴う応答と理由のフィールドは X'020B' です。

これ以外のエラーではすべて、エラー・コードを含むメッセージが発行されます。

機能コード・フィールドに X'16' が含まれている場合、0B2A 例外トレース・ポイントは DISCONNECT 時 (つまり、リモート・ファイルとの接続を切断しようとしているとき) のエラーを示します。すべてのトレース・ポイントで発生する可能性があるエラーに加え、以下のような可能性も考えられます。

X'02' 共用データ・テーブル・サービスはまだ初期設定されていません (領域のアンカー・ブロックは作成されていません)。

X'03' 要求側の領域が、共用データ・テーブル・リクエスターとして初期設定を完了していません。

X'05' CICS 領域が、データ・テーブル・リクエスターとして初期設定されたときと異なる要求ブロック (RB) のもとで実行されています。エラー・コードの追加情報部分に、呼び出しで使用された RB アドレスが含まれています。

X'07' 呼び出し元が提供した、ファイル接続のベクトルへの索引が無効です。エラー・コードの追加情報部分に、呼び出し元の索引の下位 3 バイトが含まれています。

X'10' 指定された接続は既に切断されていて、存在しなくなっています。エラー・コードの追加情報部分に、ファイル接続のベクトルへの呼び出し元の索引の下位 3 バイトが含まれています。

これ以外のエラーではすべて、エラー・コードを含むメッセージが発行されます。

データ・テーブルのクロスメモリー・サービスからのエラーの分析

データ・テーブルのクロスメモリー・サービスから発生した予期しないエラーの後には、**X'0B15'** 例外トレース項目が作成されます。これには、応答コードと理由コード、およびエラーの原因を識別するエラー・コード・フィールドが含まれます。これらのエラーはすべてルーチンまたはシステムの破損が原因であり、場合によってはルーチンの誤用と思われることが原因の場合もあります。

応答と理由が X'0215' の場合、エラー・コードのフォーマットは X'ffaaaaaa' です。*ff* は障害のタイプを識別し、*aaaaaa* は一部の障害の場合に提供される追加情報です。表示される可能性のある *ff* の値は以下のとおりです。

X'01' クロスメモリー取得ルーチンまたは接続ベクトル・ルックアップ・ルーチンによって行われた、CICS AFCB (許可機能制御ブロック) を検出する試みが失敗しました。

X'02' 要求側の CICS 領域がまだ共用データ・テーブルの初期設定を実行していません (その領域のアンカー・ブロックがまだ作成もセットアップもされていません)。

X'03' 要求側の領域が、共用データ・テーブル・リクエスターとして初期設定を完了していません。

X'05' 共用データ・テーブル・リクエスターとして初期設定を行った要求ブロックとは異なる要求ブロックのもとで、取得要求が発行されました。

X'06' リモート・ファイルの接続ベクトル・エントリーに、正しいリンケージ索引が含まれていません。

X'07' リモート・ファイルの接続ベクトル・エントリーの索引が、その接続ベクトルの終わりを越えています。

X'08' リモート・ファイルの接続ベクトル・エントリーに使用中のマークが付いていません。

X'09' クロスメモリー取得ルーチンが、正しいメカニズムを介して呼び出されていません。

応答と理由が X'0400' の場合、サーバー領域で実行されているレコード管理コードに渡された機能コードが認識されない値であったことを意味します。

データ・テーブルに関するダンプ情報

データ・テーブルに関連する情報は CICS システム・ダンプに含まれており、問題の原因判別に利用できます。

ダンプの内容について、およびその内容の取得方法については、Using dumps in problem determinationを参照してください。

共用データ・テーブル・サポートによって使用される主制御ブロックは、ファイル占有領域のフォーマット済みダンプの FILE CONTROL 領域に含まれています。これらの制御ブロックは以下のように呼ばれます。

データ・テーブル・グローバル域
: これは共用データ・テーブル・ヘッダー・ブロック とも呼ばれるため、DFHDTHEADER という目印が使用されます。

データ・テーブル・ベース域
: これは共用データ・テーブル・ブロック とも呼ばれるため、DFHDTTABLE という目印が使用されます。

データ・テーブル・パス域
: これは共用データ・テーブル・ファイル・ブロック とも呼ばれるため、DFHDTFILE という目印が使用されます。

データ・テーブルの内容は CICS システム・ダンプに含まれません。データ・テーブルが格納されているデータ・スペース・ストレージが CICS アドレス・スペースに含まれていないためです。テーブル・エントリーはデータ・スペース DFHDT001 に格納され、索引ノードはデータ・スペース DFHDT002 に格納され、レコード・データはデータ・スペース DFHDT003、DFHDT004、DFHDT005 に格納される、というようになっており、新規データ・スペースは必要に応じて追加されます。データ・テーブルの内容を確認する必要がある場合は、MVS **DUMP** コマンドを使用して該当する CICS 始動ジョブが所有する該当するデータ・スペースのダンプを要求するように、システム・オペレーターに依頼してください。

オペレーター・コマンド `DISPLAY J,CICS-startup-jobname` は、CICS ジョブに関する情報 (CICS ジョブが所有するデータ・スペースの DSPNAME を含む) を表示します。データ・スペース DFHDT003 の内容をダンプする場合は、次のように MVS **DUMP** コマンドを使用できます。

1. 入力

   ```
   DUMP COMM=(title for your dump)
   ```

2. これによって次の MVS コンソール・メッセージが生成されます。

   ```
   * id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
   ```

3. このメッセージに以下のように応答します。

   ```
   REPLY id, DSPNAME='jobname'.DFHDT003
   ```

   これで、データ・スペース・ストレージがダンプされます。

   注: 次のアスタリスク表記を使用すると、CICS が所有するすべての データ・スペースの内容をダンプできます。

   ```
   REPLY id, DSPNAME='jobname'.DFHDT*
   ```

   ただし、アスタリスクの使用は慎重に行ってください。多数のデータ・スペースがある場合はダンプ・データ・セットが巨大になる可能性があるためです。

4. `DISPLAY DUMP,TITLE` を使用して、どの SYS1.DUMPnn データ・セットが使用されたかを確認します。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。この資料の他の言語版を IBM® から入手できる場合があります。ただし、これを入手するには、本製品または当該言語版製品を所有している必要がある場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。 IBM 製品、プログラムまたはサービスに代えて、IBM の知的所有権を侵害することのない機能的に同等のプログラムまたは製品を使用することができます。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。

〒*103-8510*
東京都中央区日本橋箱崎町*19*番*21*号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

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

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

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様自身の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

© Copyright IBM Corp. 2005, 2018

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119 Armonk,
NY 10504-1785
United States of America

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。 IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

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

著作権使用許諾:

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

プログラミング・インターフェース情報

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

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

- アプリケーションの開発
- Developing system programs
- 保護の概要
- 外部インターフェースに向けた開発
- リファレンス: アプリケーション開発h
- リファレンス: システム・プログラミング
- リファレンス: 接続

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

- Troubleshooting and support
- リファレンス: 診断

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

- アプリケーション・プログラミング・ガイドおよびアプリケーション・プログラミング・リファレンス
- Business Transaction Services
- Customization Guide
- C++ OO Class Libraries
- Debugging Tools Interfaces Reference
- Distributed Transaction Programming Guide
- External Interfaces Guide
- Front End Programming Interface Guide
- IMS Database Control Guide
- インストール・ガイド
- セキュリティー・ガイド
- Supplied Transactions
- CICSPlex® SM Managing Workloads
- CICSPlex SM Managing Resource Usage
- CICSPlex SM アプリケーション・プログラミング・ガイドおよび CICSPlex SM アプリケーション・プログラミング・リファレンス
- Java™ Applications in CICS

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 5 のプログラミング・インターフェース

として意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が以下のマニュアルに含まれています。

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

商標

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

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

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

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

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

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

製品資料に関するご使用条件

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

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

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

商用使用
: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することがで

きます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

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

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

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

IBM これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態で提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

IBM オンラインでのプライバシー・ステートメント

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

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

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

CICSPlex SM Web ユーザー・インターフェース **(**「**Hello World**」ページ**)** の場合:　このソフトウェア・オファリングは、展開される構成に応じて、個人情報を収集しないセッションごとの Cookie を使用する場合があります。これらの Cookie を無効にすることはできません。

CICS Explorer® の場合**:**

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

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

このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、『IBM オンラインでのプライバシー・ステートメント』(http://www.ibm.com/privacy/details/jp/ja/) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』および「IBM Software Products and Software-as-a-Service Privacy Statement」(http://www.ibm.com/software/info/product-privacy) を参照してください。

索引

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

[ア行]

アプリケーション・プログラミング
　共用データ・テーブルの機能拡張 1
　CMT 用
　　説明 33
　UMT 用
　　概要 35
　　説明 35
異常終了コード
　AFCH 37, 38
　AFCZ 76
一時データ・キュー
　メッセージで使用される 52
インストール
　パラメーター・リスト 20
　MVS の考慮事項 21
インターフェース
　プロダクト・センシティブ・プログラミング 43
　ユーザー出口 43
応答コード
　トレース・ポイントの 73

[カ行]

仮想記憶間サービス
　アプリケーションによる使用 36
　エラーの分析 80
　機能シップとの比較 6, 38
　サポートされるコマンド 33
　利点 1
キーの長さ
　機能シップとの比較 39
機能
　トレース・ポイントの 72
ギャップ 12, 33, 38
共用
　環境 3
　共用データ・テーブルの操作 6, 7
　シスプレックスで 4
　CONNECT 操作 8
　LOGON 操作 7
共用データ・テーブルの概要 1
グローバル・リソースの逐次化、GRS を参照 56
更新要求
　概要 2
　CMT 用 33, 34
　UMT 用 35

[サ行]

サーバー
　定義 3
削除要求
　VSAM との比較 40
システム・ダンプ情報 80
シスプレックス環境
　共用データ・テーブルの使用 53
　サンプル・プログラムの作動 57
　サンプル・プログラムのセットアップと実行 59
　サンプル・プログラム・コード 59
　導入 4
　複製された UMT のリフレッシュ 54
自動ジャーナリング 24, 26
ジャーナリング 24
修飾子フラグ
　トレース・ポイントの 72
使用ストレージ
　説明 11
セキュリティー検査
　機能シップとの比較 20, 39
　データ・テーブルの 20
　AOR 接続時 21
　FOR ログオン時 21
　RACF の考慮事項 20
　SAF の使用 20
切断
　AOR およびデータ・テーブル 9, 37
ソース・データ・セット
　データ・テーブルの 2
　ファイルが複数の 24
　CMT で使用 2
　KSDS である必要がある 4
　UMT とは独立 3

[タ行]

代替索引 4, 24
通信
　CICS とユーザー出口間 43
通知
　CONNECT 操作の場合 9
データ保全性
　CMT の 24
　UMT の 25
データ・スペース
　データ・テーブルによる使用 4, 12
　内容のダンプ 80
データ・テーブルに関するダンプ情報 80
データ・テーブルのオープン 51
データ・テーブルの概念 1
データ・テーブルの可用性 1
データ・テーブルのクローズ 38, 52
データ・テーブルの計画 11, 21
データ・テーブルのサイズ
　CEDA による定義 26
　INQUIRE コマンドによる検索 31, 32
　SET コマンドによる定義 31, 32
データ・テーブルの初期状態
　CEDA による定義 26
データ・テーブルの操作 51
データ・テーブルのタイプ
　CEDA による定義 26
　INQUIRE コマンドによる検索 31, 32
　SET コマンドによる定義 31, 32
データ・テーブルの問題判別 71
データ・テーブルのリカバリー
　緊急時再始動中 52
　CEDA による定義 26
データ・テーブルを使用不可にする 38
デイジー・チェーン 37
統計
　データ・テーブル選択のため 16
動的トランザクション・バックアウト 26
トレース情報
　入り口と出口のポイント 71
　応答コード 73
　機能および修飾子フラグ 72
　データ・テーブルの 71
　理由コード 74
　例外ポイント 75

[ハ行]

バインド・セキュリティー 21
パフォーマンス
　データ・テーブルの利点 11
　CMT の 11
　UMT の 11
パラメーター・リスト
　ユーザー出口 43
ファイル
　データ・テーブルとして使用 1, 14

© Copyright IBM Corp. 2005, 2018

ファイル管理
仮想記憶間サービスの使用 6
機能シップの使用 6
ファイル制御
コマンド
仮想記憶間サービスによるサポート 33
CMT の概要 33
UMT の概要 35
ユーザー出口 43
ファイルの選択
データ・テーブルとして使用する 14
ファイル・セキュリティー 21
複数のファイル
ソース・データ・セットが同じ 24
複製された UMT のリフレッシュ 54
ブラウズ要求
機能シップとの比較 38
CMT 用 33
UMT 用 35
VSAM との比較 40
プロダクト・センシティブ・プログラミング・インターフェース 43
保全性
CMT データの 24
UMT データの 25

［マ行］

メッセージ
ロード開始時点の 52
ロード終了時点の 52

［ヤ行］

ユーザー出口
概要 6
活動化 49
使用可能化 49
説明 43
定義 49
出口プログラムのサンプル 43
パラメーター・リスト 43
パラメーター・リストの DSECT 43
ファイル制御用 39, 43
レコードの追加時 48
ロード終了時点の 49
ロード中 47
CICS との通信 43
EXEC インターフェース用 39, 43
XDTAD 出口 48
XDTLC 出口 49
XDTRD 出口 47
ユーザー出口によるカスタマイズ 43
ユーザー出口の活動化 49
ユーザー出口の使用可能化 49
ユーザー保守データ・テーブル
概要 3
更新要求 35
シスプレックス内の複製 53
ジャーナリング 26
説明 3
データ保全性 25
パフォーマンス 11
ブラウズ要求 35
読み取り要求 35
リソース定義 23
ロード中の使用 36
読み取り要求
概要 2
機能シップとの比較 39
CMT 用 33, 34
UMT 用 35
VSAM との比較 40

［ラ行］

リクエスター
定義 3
リソース定義
説明 23
CMT の概要 24
DEFINE FILE コマンド 26
UMT の概要 25
利点
データ・テーブルの 11
リモート・ファイル
定義 3
理由コード
トレース・ポイントの 74
レコード・レベル共用 2
ローカル・ファイル
定義 3
ロード
CMT の使用 34
UMT の使用 36
ロード・モジュール
データ・テーブルで必要 22

A

AFCH 異常終了コード 37, 38
AFCZ 異常終了コード 76
AOR (アプリケーション専有領域)
定義 3
CONNECT 操作 8

C

CEDA DEFINE FILE コマンド
説明 26
CMT の例 28
LOG パラメーター 26
MAXNUMRECS パラメーター 26
OPENTIME パラメーター 26
RECORDFORMAT パラメーター 26
TABLE パラメーター 26
UMT の例 29
CEMT
INQUIRE コマンド 31, 32
SET コマンド 31, 32
CFTL トランザクション 51
CICS 保守データ・テーブル
概要 2
更新要求 33, 34
ジャーナリング 24
説明 2
データ保全性 24
パフォーマンス 11
ブラウズ要求 33
読み取り要求 33, 34
リソース定義 23
ロード中の使用 34
CONNECT
セキュリティー検査 21
AOR による 8, 37
CSFU トランザクション 51, 52

D

DFHDTCV 21
DFHDTSVC 21
DFHMVRMS 21
DFHXDTS コピーブック 43
DSECT
ユーザー出口パラメーター・リスト 43

E

EIBRESP2 フィールド 35, 39
exec インターフェース
ユーザー出口 43

F

FOR (ファイル所有領域)
定義 3
LOGON 操作 7

G

GRS (グローバル・リソースの逐次化) 56

I

INQUIRE FILE コマンド
説明 31, 32
MAXNUMRECS パラメーター 31, 32
TABLE パラメーター 31, 32
INSTLN パラメーター 20
INVREQ 状態 35

K

KSDS (キー順データ・セット)
ソース・データ・セットとして使用 2, 4
UMT を使用 25

L

LOADING 状態 36
LOGON
セキュリティー検査 21
FOR によって 7

M

MVS の考慮事項 14, 21

N

NOSPACE 状態 35
NOTFND 状態 35, 36

R

RACF
セキュリティー・マネージャーとして使用 20
RLS (レコード・レベル共用) 2

S

SAF、System Authorization Facility
セキュリティー検査のための使用 20
SET FILE コマンド
説明 31, 32
MAXNUMRECS パラメーター 31, 32
TABLE パラメーター 31, 32
SHAREOPTION、VSAM 24
SUPPRESSED 状態 35
SVC エラー 76
SYSID パラメーター 36
System Authorization Facility 20

V

VSAM
アクセス方式制御ブロック 51, 52
基本クラスター (base cluster) 24
代替索引 4, 24
データ・テーブルとの比較 39
SHAREOPTION 24
VSAM RLS
データ共用のため 16
リカバリー属性 26
CMT に不適切 2, 24
UMT に適切 3, 25

X

XDTAD ユーザー出口
説明 48
XDTLC ユーザー出口
説明 49
XDTRD ユーザー出口
説明 47

日本アイ･ビー･エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21