

CICS Transaction Server for z/OS  
Version 5 Release 6

*Gemeinsam genutzte Datentabellen*



**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter [Rechtliche Hinweise zum Produkt](#) gelesen werden.

Diese Ausgabe bezieht sich auf IBM® CICS Transaction Server for z/OS, Version 5 Release 6 (Produktnummer 5655-Y305655-BTA ) und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

© **Copyright International Business Machines Corporation 1974, 2020.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Informationen zu dieser PDF.....</b>	<b>V</b>
<b>Chapter 1. Übersicht über gemeinsam genutzte Datentabellen.....</b>	<b>1</b>
Beschreibung von Datentabellen.....	2
CICS-verwaltete Datentabelle.....	2
Benutzerverwaltete Datentabelle.....	2
Die Umgebung für die gemeinsame Nutzung einer Datentabelle.....	3
Die Quellendatei für Datentabellen.....	3
Datenspeicherbereiche für Datentabellen.....	4
Globale Benutzerexits für Datentabellen.....	5
Gemeinsam genutzte Datentabellenservices und Zugriff auf ferne Dateien.....	5
Vorgehensweise bei der gemeinsamen Nutzung einer Datentabelle.....	9
LOGON.....	9
CONNECT.....	9
<b>Chapter 2. Gemeinsam genutzte Datentabellen konfigurieren.....</b>	<b>11</b>
Die Verwendung von Datentabellen planen.....	11
Leistung einer CICS-verwalteten Datentabelle.....	11
Leistung einer benutzerverwalteten Datentabelle.....	11
Speichernutzung für gemeinsam genutzte Datentabellen.....	11
MVS-JCL-Anforderungen bei Verwendung von gemeinsam genutzten Datentabellen.....	13
Dateien für die Verwendung als Datentabellen auswählen.....	13
Statistikdaten zum Auswählen von Datentabellen verwenden.....	15
Sicherheitsprüfung für Datentabellen.....	18
Verwendung der Unterstützung für gemeinsam genutzte Datentabellen vorbereiten.....	19
Ressourcendefinition für Datentabellen.....	20
Ressourcendefinition für CICS-verwaltete Datentabellen.....	21
Ressourcendefinition für benutzerverwaltete Datentabellen.....	22
Datentabellen mit dem Befehl <b>DEFINE FILE</b> definieren.....	22
EXEC CICS-Befehle für Datentabellen.....	26
CEMT-Befehle für Datentabellen.....	27
<b>Chapter 3. Zugriff auf Datentabellen entwickeln.....</b>	<b>29</b>
Anwendungsprogrammierung für CICS-verwaltete Datentabellen.....	29
CICS-verwaltete Datentabelle beim Laden verwenden.....	30
Anwendungsprogrammierung für eine benutzerverwaltete Datentabelle.....	30
Benutzerverwaltete Datentabelle beim Laden verwenden.....	31
Speicherübergreifende Services für gemeinsam genutzte Datentabellen verwenden.....	32
Verbindung herstellen.....	32
Unterschiede zwischen Funktionsverlagerung und speicherübergreifenden Services.....	33
Unterschiede zwischen gemeinsam genutzten Datentabellenservices und VSAM.....	34
<b>Chapter 4. Datentabellen mithilfe von Benutzerexits anpassen.....</b>	<b>37</b>
Kommunikation zwischen CICS und Exitprogrammen für gemeinsam genutzte Datentabellen.....	37
Benutzerexit XDTRD.....	40
Benutzerexit XDTAD.....	41
Benutzerexit XD TLC.....	42
Benutzerexits für Datentabellen aktivieren.....	42
<b>Chapter 5. Datentabellen verwalten.....</b>	<b>45</b>

Datentabellen öffnen.....	45
Datentabellen schließen.....	46
Unterstützung für gemeinsam genutzte Datentabellen in einem Sysplex verwenden.....	46
Übersicht über die Unterstützung für gemeinsam genutzte Datentabellen in einem Sysplex.....	46
Vorgehensweise beim Aktualisieren von replizierten benutzerverwalteten Datentabellen.....	47
Beispielprogramm zum Aktualisieren einer benutzerverwalteten Datentabelle.....	49
<b>Chapter 6. Fehlerbehebung für Datentabellen.....</b>	<b>61</b>
Traceinformationen für Datentabellenservices.....	61
Eingangs- und Ausgangstracepunkte für gemeinsam genutzte Datentabellen.....	61
Ausnahmebedingungstracepunkte für gemeinsam genutzte Datentabellen.....	65
Fehler aus den SVC-Datentabellen analysieren.....	66
Werte für alle Tracepunkte für gemeinsam genutzte Datentabellen.....	66
Werte für den Tracepunkt 0B12.....	67
Werte für den Tracepunkt 0B19.....	67
Werte für den Tracepunkt 0B1A.....	67
Werte für den Tracepunkt AP 0B29.....	68
Werte für den Tracepunkt 0B2A.....	68
Fehler von speicherübergreifenden Datentabellenservices analysieren.....	69
Speicherauszugsinformationen für Datentabellen.....	70
<b>Bemerkungen.....</b>	<b>73</b>
<b>Index.....</b>	<b>79</b>

# Informationen zu dieser PDF

---

Diese PDF-Datei enthält Informationen zu gemeinsam genutzten CICS-Datentabellenservices.

Es richtet sich an alle Personen, die an gemeinsam genutzten CICS-Datentabellen in einem oder mehreren der folgenden Bereiche beteiligt sind:

- Planung
- Anwendungsprogrammierung
- Ressourcendefinition
- Anpassung
- Operationen
- Problembestimmung

Ausführliche Informationen zu den verwendeten Begriffen und Notationen finden Sie in In der Dokumentation zu CICS verwendete Konventionen und Terminologie im IBM Knowledge Center.

## **Datum dieser PDF**

Diese PDF wurde am 28. Mai 2020 erstellt.



---

# Kapitel 1. Übersicht über gemeinsam genutzte Datentabellen

Die CICS-Einrichtung für gemeinsam genutzte Datentabellen ist eine Erweiterung der CICS-Dateiverwaltungsservices. Bei der Verwendung gemeinsam genutzter Datentabellen können alle Dateien, die als Datentabellen definiert sind, möglicherweise mithilfe von speicherübergreifenden Services freigegeben werden. Es sind keine Änderungen für die Dateidefinitionen für vorhandene Datentabellen erforderlich.

Das Konzept der gemeinsam genutzten Datentabellen besagt, dass es bei Folgendem effizienter ist:

- Wenn Sie speicherübergreifende MVS-Services anstelle der CICS-Funktionsverlagerung verwenden möchten, um eine Datendatei zwischen zwei oder mehreren CICS-Regionen im selben MVS-Image gemeinsam zu nutzen.
- Um auf Daten aus dem Speicher anstelle aus der DASD-Einheit zuzugreifen.
- Um auf eine Datendatei aus dem Speicher zuzugreifen, indem Sie Services verwenden, die in der CICS-Dateiverwaltung integriert sind, anstatt VSAM-Services sowie einen LSR-Pool zu verwenden.

Die zwei Versionen von Datentabellen sind:

- Unterstützung für Basisdatentabellen, die als Teil von CICS Transaction Server for z/OS bereitgestellt werden.
- Unterstützung für gemeinsam genutzte Datentabellen, die als Teil von CICS Transaction Server for z/OS bereitgestellt werden.

## **Vorteile gemeinsam genutzter Datentabellen**

Die Verwendung von speicherübergreifenden Services ist einer der Hauptvorteile von gemeinsam genutzten Datentabellen. Die Leistung von Anwendungen, die zurzeit eine Funktionsverlagerung verwenden, wird verbessert und das Freigeben von Dateien für Anwendungen wird ermöglicht, die die Leistungseinbußen der Funktionsverlagerung nicht akzeptieren können.

Die andere wichtige Erweiterung besteht darin, dass fast alle Leseanforderungen für die Verwendung mit Datentabellen unterstützt werden. Diese Erweiterung erweitert die Verwendung von Datentabellen auf Anwendungen, die Folgendes enthalten:

- Suchanforderungen
- Leseanforderungen, die einen ungenauen Schlüssel verwenden

Neben der Verwendung von speicherübergreifenden Services bieten die gemeinsam genutzten Datentabellen Vorteile für die Leistung und Sicherheit:

- Für ferne Zugriffe können sehr große Reduktionen in der Pfadlänge erzielt werden, da die Funktionsverlagerung für die meisten Lese- und Suchanforderungen vermieden wird.
- Wenn speicherübergreifende Services verwendet werden, werden die Anforderungen von der AOR verarbeitet, sodass die FOR andere Anforderungen bearbeiten kann. Dies erhöht die Multiprozessor-Ausnutzung.
- Es wird eine erhöhte Sicherheit von Daten bereitgestellt, da die Datensatzinformationen in gemeinsam genutzten Datentabellen außerhalb der CICS-Region gespeichert werden und nicht in CICS-Systemspeicherausügen enthalten sind (entweder formatiert oder nicht formatiert).
- Bei CICS-verwalteten Datentabellen werden alle Formen des nicht aktualisierten, verschlüsselten Zugriffs (einschließlich Suchanfragen und Leseanforderungen mit unpräzisen Schlüsseln) durch Verweis auf die Datentabelle verarbeitet.
- Bei benutzerverwalteten Datentabellen werden alle Formen des nicht aktualisierten, verschlüsselten Zugriffs (einschließlich Suchanfragen und Leseanforderungen mit unpräzisen Schlüsseln) unterstützt.

- Eine beliebige Anzahl gleichzeitig geöffneter Dateien, die sich auf dieselbe Quellendatei beziehen, können Daten aus der jeweiligen CICS-verwalteten Datentabelle abrufen.
- Eine Erweiterung des Benutzerexits XDTRD ermöglicht es Ihnen, einen Bereich von Datensätzen zu überspringen, während die Datentabelle geladen wird.

## Beschreibung von Datentabellen

Eine CICS-Datei ist die Darstellung einer Datei für DASD. Wenn Sie angeben, dass die Datei Datentabellenservices verwenden soll, kopiert CICS den Inhalt der Datei in einen MVS-Datenspeicherbereich, wenn die Datei geöffnet wird, und verwendet diese Kopie, wann immer dies möglich ist.

Aufgrund der Art und Weise, in der die Datentabellenservices auf die Datensätze zugreifen, können sie nur mit einer VSAM-Datei in Schlüsselfolge (KSDS) verwendet werden. Die KSDS wird als *Quellendatei* bezeichnet. Die Kopie im Speicher heißt *Datentabelle*. Das Kopieren der Datensätze wird als *Laden* in die Datentabelle bezeichnet.

Wenn die Datei von einer CICS-Anwendung gelesen wird, wird der Datensatz normalerweise aus der Datentabelle abgerufen. Wenn die Datei von einer CICS-Anwendung aktualisiert wird, hängt der Effekt vom Typ der Datentabelle ab, den Sie für die Datei definiert haben.

CICS-Datentabellenservices unterstützen zwei Typen von Datentabellen:

- CICS-verwaltete Datentabelle (CMT)
- Benutzerverwaltete Datentabelle (UMT)

### CICS-verwaltete Datentabelle

Bei einer CICS-verwalteten Datentabelle handelt es sich um eine Datentabelle, deren Datensätze automatisch in der Quellendatei abgebildet werden. Wenn Sie die Datei aktualisieren, ändert CICS sowohl die Quellendatei als auch die Datentabelle.

Die Behandlung der Quellendatei und der Datentabelle als eine einzige Entität bedeutet Folgendes:

- Die Änderungen an der Datei werden sowohl in der Quellendatei als auch in der Datentabelle vorgenommen.
- Wenn eine andere Datei für die Verwendung derselben Quellendatei definiert ist, werden die Änderungen, die von dieser Datei an der Quellendatei vorgenommen werden, ebenfalls an der Datentabelle vorgenommen.
- Wenn eine andere Datei für die Verwendung derselben Quellendatei definiert ist, können die Datensätze von dieser Datei aus der Datentabelle abgerufen werden.

Eine CICS-verwaltete Datentabelle ist einfach zu implementieren. Sie müssen wenig über die Datentabellenservices wissen, Sie müssen Ihre vorhandenen Anwendungsprogramme nicht ändern und die Unterstützung für eine vollständige Wiederherstellung der Datei wird beibehalten.

Eine Datei, auf die im RLS-Modus (Record Level Sharing) zugegriffen wird, kann nicht als Quelle für eine CICS-verwaltete Datentabelle verwendet werden. Der Zugriff auf die Quellendatei muss im Nicht-RLS-Modus erfolgen.

### Benutzerverwaltete Datentabelle

Bei einer benutzerverwalteten Datentabelle handelt es sich um eine Datentabelle, deren Datensätze nicht automatisch in der Quellendatei abgebildet werden. Wenn Sie die Datei aktualisieren, ändert CICS nur die Datentabelle.

Nachdem eine benutzerverwaltete Datentabelle geladen wurde, ist sie unabhängig von ihrer Quellendatei. Die Quellendatei wird nicht aktualisiert, wenn die Datentabelle aktualisiert wird. Eine benutzerverwaltete Datentabelle ermöglicht es Ihnen, die Vorteile der Verwendung einer Datentabelle zu optimieren, indem Sie die Aktivität in der Quellendatei, für Aktualisierungsanforderungen und Leseanforderungen entfernen.



Daher ist eine benutzerverwaltete Datentabelle besonders für Anwendungen geeignet, die häufige Aktualisierungen von Daten transienter Art vornehmen.

Eine kleine Anzahl von Dateioperationen wird nicht für benutzerverwaltete Datentabellen unterstützt. Daher müssen Sie möglicherweise geringfügige Änderungen an vorhandenen Anwendungsprogrammen vornehmen. Darüber hinaus wird die Wiederherstellung der Datei nach einem Transaktionsfehler, aber nicht bei einem Systemausfall unterstützt.

Eine Basis-VSAM-KSDS, auf die im Nicht-RLS- oder RLS-Modus zugegriffen wird, kann als Quellendatei für eine benutzerverwaltete Datentabelle verwendet werden. Möglicherweise möchten Sie eine RLS-Modus-Datei als Quelle einer benutzerverwalteten Datentabelle definieren, wenn Sie über andere Dateidefinitionen verfügen, die auf die Datei zugreifen, und die Datei von anderen CICS-Regionen aktualisiert wird.

## Die Umgebung für die gemeinsame Nutzung einer Datentabelle

Die Umgebung für die gemeinsame Nutzung einer Datentabelle ist die gleiche wie für jede Datei, auf die im Nicht-RLS-Modus zugegriffen wird.

Eine CICS-Region ist der Eigner der Datentabelle. Diese Region wird als *File-Ownning Region* (FOR) bezeichnet. Alle anderen Regionen, die die Datentabelle verwenden, werden als *Application-Ownning Region* (AOR) bezeichnet. In der FOR ist die Datei als *lokale Datei* bekannt und wird in der AOR als *ferne Datei* bezeichnet.

Im Zusammenhang mit gemeinsam genutzten Datentabellen wird die FOR auch als *Server* und die AOR auch als *Anforderer* bezeichnet.

Dieselbe Region kann sowohl eine FOR für einige Datentabellen als auch eine AOR für andere sein.

Informationen zu diesen übergreifenden Kommunikationskonzepten finden Sie in [Methoden der übergreifenden Kommunikation](#).

Die Unterstützung für gemeinsam genutzte Datentabellen verwendet (wenn möglich) die regionsübergreifende gemeinsame Nutzung, um den Zugriff auf Datentabellen zu ermöglichen, die sich in demselben MVS-Image befinden wie der anfordernde CICS-Bereich. Dies bedeutet, dass die meisten Lesezugriffe innerhalb desselben MVS-Images durch die regionsübergreifende gemeinsame Nutzung mit gemeinsam genutzten Datentabellenservices erfüllt werden. Wenn die regionsübergreifende gemeinsame Nutzung für die Anforderung nicht möglich ist, wird die Funktionsverlagerung verwendet. Dies bedeutet, dass Aktualisierungsanforderungen von CICS-Regionen innerhalb desselben MVS-Images sowie alle Anforderungen von CICS-Regionen in verschiedenen MVS-Images die Funktionsverlagerung verwenden. [Anwendungsprogrammierung für CICS-verwaltete Datentabellen](#) und [Anwendungsprogrammierung für benutzerverwaltete Datentabellen](#) teilen Ihnen mit, wann Befehle über die gemeinsame regionsübergreifende Nutzung oder die Funktionsverlagerung erfüllt werden.

**Anmerkung:** Ebenso stellt XCF/MRO keinen Zugriff auf die gemeinsame Datentabelle zwischen CICS-Regionen in verschiedenen MVS-Images bereit.

Obwohl die Unterstützung für gemeinsam genutzte Datentabellen hauptsächlich für die gemeinsame Nutzung von Daten innerhalb eines MVS-Images vorgesehen ist, kann die Unterstützung auf eine Sysplex-Umgebung für Anwendungen erweitert werden, die nur Lesezugriff auf eine gemeinsam genutzte benutzerverwaltete Datentabelle benötigen oder mit Daten arbeiten können, die möglicherweise nicht auf dem neuesten Stand sind. Die Datentabelle muss in jeder MVS-Region im Sysplex repliziert und in regelmäßigen Abständen aktualisiert werden. Siehe [Unterstützung für gemeinsam genutzte Datentabellen in einem Sysplex verwenden](#).

## Die Quellendatei für Datentabellen

Die Quellendatei muss eine Basis-VSAM-KSDS (kein Alternativindex) sein. Aktualisierungen, die über einen Alternativindex an der Datei in Schlüsselfolge (KSDS) vorgenommen werden, werden in einer von CICS-verwalteten Datentabelle abgebildet.

Die VSAM-Definition der Datei in Schlüsselfolge (KSDS) liefert die Werte für die maximale Datensatzlänge und Schlüssellänge.

Bei einer benutzerverwalteten Datentabelle (UMT) werden die Aktualisierungen weder in der Quellendatei noch in den zugehörigen Alternativindizes abgebildet. Eine benutzerverwaltete Datentabelle ist nach Abschluss des Ladevorgangs vollständig unabhängig von der zugehörigen Quellendatei.

## Datenspeicherbereiche für Datentabellen

Die Datentabellendatensätze werden in einem oder mehreren MVS-Datenspeicherbereichen gespeichert, unabhängig davon, ob die Datentabelle von mehreren Regionen gemeinsam genutzt werden soll. Für jede CICS-Region werden separate Gruppen von Datenspeicherbereichen verwendet.

Die erste Gruppe aus den Datenspeicherbereichen mit den Namen DFHDT001 (für Tabelleneintragsdeskriptoren), DFHDT002 (für Indexknoten) und DFHDT003 (für bis zu 2 GB Datensatzdaten) werden abgerufen, wenn die erste Datei, die als Datentabelle definiert ist, in der Region geöffnet wird. Zusätzliche Datenspeicherbereiche von DFHDT004 und höher können bei Bedarf für Datensatzdaten zugeordnet werden (bis zu maximal 100 Datenspeicherbereiche pro Region). Die Datenspeicherbereiche werden von allen CICS-Datentabellen verwendet, deren Eigner diese Region ist, und sie werden so lange beibehalten, bis die CICS-Region heruntergefahren wird.

Jeder Datenspeicherbereich hat eine maximale Größe von 2 GB, sodass die maximale Größe des Datenspeicherbereichs 200 GB beträgt, den eine CICS-Region zuordnen kann (vorausgesetzt, es sind ausreichend Betriebssystemressourcen vorhanden). Der MVS-Exit IEFUSI, mit dem der gesamte Datenspeicherbereich, den ein bestimmter Adressraum besitzen kann, gesteuert werden kann, kann die maximale Größe auf weniger als diesen Wert und sogar auf weniger als 2 GB reduzieren. Innerhalb dieses Grenzwerts ordnet CICS den Datenspeicherbereich in Einheiten von 16 MB zu und ordnet diesen Speicher dann den Datentabellen in Schritten von 32 KB für Tabelleneintragsdeskriptoren oder Indexknoten und in Schritten von 128 KB für Datensatzdaten zu. Wenn für eine Datentabelle ein neues Speicherinkrement benötigt wird, aber alle vorhandenen Datenspeicherbereiche bereits Tabellen zugeordnet sind, versucht CICS, den Datenspeicherbereich um 16 MB zu erweitern. Wenn der Datenspeicherbereich für Datensatzdaten gilt, aber bereits die maximale Größe von 2 GB erreicht hat und die maximale Anzahl an Datenspeicherbereichen nicht erreicht wurde, erstellt CICS statt des vorhandenen Datenspeicherbereichs einen neuen Datenspeicherbereich, der als logische Erweiterung der vorhandenen Gruppe von Datenspeicherbereichen behandelt wird. Wenn CICS den Datenspeicherbereich für Tabelleneintragsdeskriptoren oder Indexknoten nicht erweitert kann, weil der Datenspeicherbereich die maximale Größe 2 GB erreicht hat, oder wenn CICS keinen weiteren Datenspeicherplatz zuordnen kann weil die gesamte Datenspeicherplatzgröße, die vom Exit IEFUSI der Installation erreicht wurde, stellt CICS fest, dass der Datenspeicherbereich voll ist.

Wenn ein Datenspeicherbereich voll ist, können alle gemeinsam genutzten Datentabellenanforderungen, die zusätzlichen Speicher benötigen, aufgrund unzureichenden Speichers fehlschlagen. Bei einer CICS-verwalteten Datentabelle bedeutet dies, dass alle zukünftigen Lesevorgänge für die betroffenen Datensätze (einschließlich aller näherungsweise berechneten Lesevorgänge dieses Schlüssels) via Funktionsverlagerung auf die VSAM-Datei zugreifen müssen, wenn die Anforderung nicht von der File-Ownning Region (FOR) abgesetzt wird. Bei einer benutzerverwalteten Datentabelle bedeutet dies, dass der Datensatz nicht in die Tabelle geschrieben werden kann. Entwicklung für Zugriff auf Datentabellen enthält Informationen zu der Antwort, die in dieser Situation zurückgegeben wird.

CICS stellt keine Funktion zum Anzeigen des aktuellen Gesamtvolumens des zugeordneten Datenspeicherbereichs bereit. Die Statistik der CICS-Dateisteuerung kann jedoch einen genauen Hinweis auf den Speicher geben, der jeder Datentabelle zugeordnet und verwendet wird. Insbesondere enthält das Feld A17DTALD das Volumen des Datenspeicherplatzes (in KB), der der Tabelle momentan zugeordnet ist.

Der Datenspeicherbereich, der von der Datentabelle verwendet wird, wird freigegeben, wenn die Datei in der FOR geschlossen wird. Dieser Speicher wird für die Wiederverwendung so verfügbar gemacht, dass die Integrität einer beliebigen Application-Ownning Region (AOR), die die Datentabelle verwendet, geschützt ist.

## Globale Benutzerexits für Datentabellen

Es stehen drei globale Benutzerexits zur Verfügung, um die normale Verarbeitung durch Datentabellenservices zu erweitern.

- XDTRD wählt die Datensätze aus, die beim Laden in die Datentabelle kopiert werden, wenn die Datei geöffnet wird. Es kann für eine benutzerverwaltete Datentabelle auch zum Ändern der Datensätze verwendet werden.
- XDTAD wählt die Datensätze aus, die in die Datentabelle kopiert werden, wenn neue Datensätze in die Datei aufgenommen werden.
- XDTLC führt die Verarbeitung am Ende des Ladevorgangs durch.

## Gemeinsam genutzte Datentabellenservices und Zugriff auf ferne Dateien

Diese Diagramme veranschaulichen die Unterschiede zwischen der Verwendung der Funktionsverlagerung und der Verwendung von gemeinsam genutzten Datentabellenservices für den Zugriff auf eine CICS-Datei in einer anderen Region.

### Funktionsverlagerung verwenden

Dieses Diagramm zeigt die Verwendung der Funktionsverlagerung für den Zugriff auf eine Datei, deren Eigner eine andere CICS-Region ist.

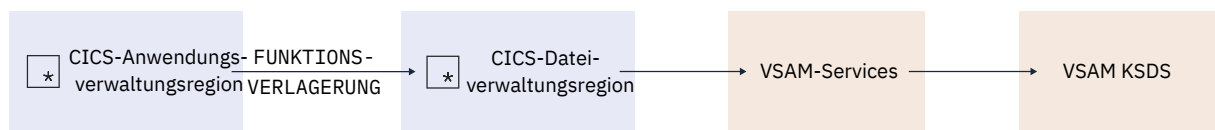


Abbildung 1. Datenzugriff unter Verwendung der Funktionsverlagerung

## **Gemeinsam genutzte Datentabellenservices verwenden**

Dieses Diagramm zeigt, wie eine Reihe von AORs (AOR = Application-Owning Region, Anwendungsverwaltungsregion) speicherübergreifende Services zum Ausführen von Lese- oder Suchvorgängen verwenden kann, wobei gemeinsam genutzte Datentabellenservices in einer File-Owning Region (FOR) verwendet werden, um auf die Datentabelle zugreifen zu können. (Die Funktionsverlagerung wird für Aktualisierungsanforderungen und für alle Anforderungen verwendet, die auf dieselbe Weise wie in [Abbildung 1 auf Seite 6](#) dargestellt auf die Quelldatei zugreifen müssen.)

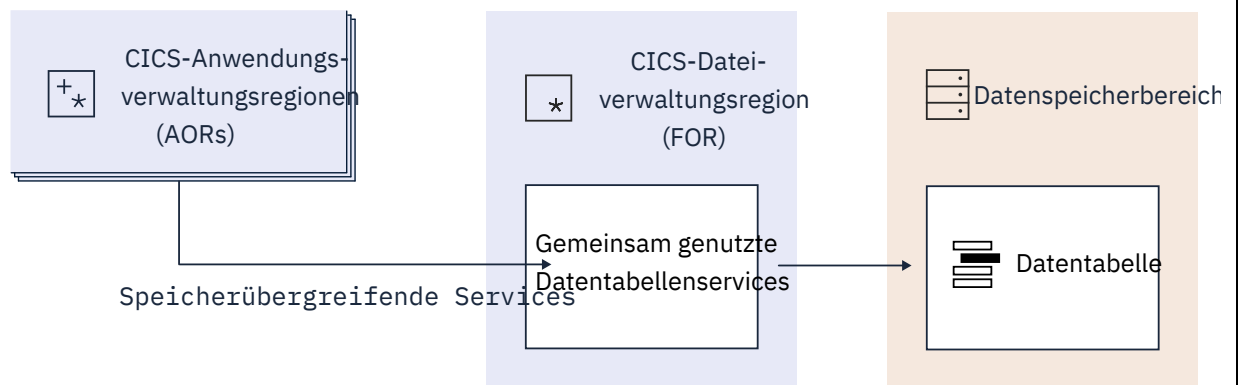


Abbildung 2. Datenzugriff mit gemeinsam genutzten Datentabellenservices

# Vorgehensweise bei der gemeinsamen Nutzung einer Datentabelle

Die zwei Operationen LOGON und CONNECT erstellen eine Datentabelle zur gemeinsamen Nutzung.

## LOGON

Wenn die erste Datei, die als Datentabelle definiert ist, in einer FOR (File-Ownning Region) geöffnet wird, versucht die FOR, sich als gemeinsam genutzter Datentabellenserver zu registrieren. Diese Operation wird automatisch ausgeführt und wird als **SDT LOGON** bezeichnet. Das Öffnen der Datei kann durch die FOR oder durch die AOR (Application-Ownning Region) verursacht werden, die zum ersten Mal auf die Datei zugreift.

Unabhängig davon, ob die LOGON-Operation erfolgreich ist, wird die Datei geöffnet und die Datentabelle geladen. Wenn die LOGON-Operation erfolgreich ist, werden alle anderen CICS-Regionen im MVS-Betriebssystem darüber benachrichtigt, dass die Datentabelle verfügbar ist.

Wenn die LOGON-Operation aufgrund einer permanenten Bedingung fehlschlägt (z. B. CICS nicht als MVS-Subsystem definiert wurde), werden während der CICS-Ausführung keine weiteren LOGON-Versuche unternommen.

Wenn die LOGON-Operation aufgrund einer möglicherweise vorübergehenden Bedingung fehlschlägt, wird ein weiterer LOGON-Versuch unternommen, wenn das nächste Mal eine Datei geöffnet wird, die als Datentabelle definiert ist. Zu dieser Art von Bedingung gehören:

- Fehlgeschlagene Sicherheitsprüfung
- Fehler beim Abrufen des Speichers
- Fehler beim Laden eines Programms

Wenn LOGON-Anforderungen einer Region aufgrund eines Fehlers bei der Sicherheitsprüfung zurückgewiesen werden, werden möglicherweise Sicherheitsverstoßnachrichten ausgegeben, wenn eine Datei geöffnet wird, die als Datentabelle definiert ist.

Nachdem eine FOR-Anmeldung erfolgreich ausgeführt wurde, verbleibt die FOR für die restliche Zeit der CICS-Ausführung in diesem Status; es werden keine weiteren LOGON-Anforderungen abgesetzt.

## CONNECT

Wenn eine AOR (Application-Ownning Region) mit Unterstützung für gemeinsam genutzte Datentabellen eine Leseanforderung für eine ferne Datei ausgibt (oder eine Suchsequenz startet), versucht CICS, eine Verbindung zu einer Datentabelle für diese Datei herzustellen. Diese Operation wird automatisch ausgeführt und wird als **SDT CONNECT** bezeichnet.

Wenn die FOR (File-Ownning Region) als gemeinsam genutzter Datentabellenserver registriert ist, erstellt CICS eine speicherübergreifende Verbindung von der AOR zur FOR (unterliegt Sicherheitsprüfungen) und ruft den gemeinsam genutzten Datentabellenserver auf, um eine verfügbare Datentabelle für die Datei zu ermitteln. Wenn ja, wird eine Verbindung zwischen der AOR und der Datentabelle hergestellt.

Wenn die CONNECT-Operation erfolgreich ist, werden die speicherübergreifenden Services verwendet (wenn möglich), um auf die Datei zuzugreifen, während die Verbindung besteht.

Wenn die CONNECT-Operation fehlschlägt, wird die Dateianforderung so funktionsverlagert, wie wenn die Unterstützung für gemeinsam genutzte Datentabellen nicht verfügbar wäre. Die Aktion, die für nachfolgende ferne Dateianforderungen ausgeführt wird, hängt von der Art des Fehlers ab, wie im Folgenden beschrieben.

Wenn die CONNECT-Operation aufgrund einer permanenten Bedingung fehlschlägt (z. B. CICS nicht als MVS-Subsystem definiert wurde), werden während der CICS-Ausführung keine weiteren CONNECT-Versuche unternommen.

Wenn die CONNECT-Operation aufgrund einer möglicherweise vorübergehenden Bedingung fehlschlägt, die nicht vom Dateieigner kontrolliert wird, wird nach etwa zehn Minuten ein weiterer CONNECT-Versuch für die nächste geeignete Anforderung ausgeführt. Zu dieser Art von Bedingung gehören:

- Fehlgeschlagene Sicherheitsprüfung
- Fehler beim Abrufen des Speichers
- Fehler beim Laden eines Programms

Wenn die CONNECT-Anforderungen einer Region aufgrund eines Fehlers bei der Sicherheitsprüfung zurückgewiesen werden, können die zugehörigen Nachrichten zum Sicherheitsverstoß in 10-Minuten-Intervallen ausgegeben werden.

Schlägt die CONNECT-Operation aufgrund einer möglicherweise vorübergehenden Bedingung fehl, die vom Dateieigner kontrolliert wird, wird nach der Benachrichtigung für die nächste geeignete Anforderung versucht, dass mindestens eine neue Datei für den gemeinsamen Zugriff auf dem MVS-System verfügbar ist. Zu dieser Art von Bedingung gehören:

- Dateieigner ist nicht als Server angemeldet
- Datei ist keiner Datentabelle zugeordnet
- Datei ist inaktiviert, obwohl sie einer Datentabelle zugeordnet ist
- Datei wird geschlossen, obwohl sie als Datentabelle definiert ist

Nachdem eine AOR eine Verbindung zu einer fernen Datei erfolgreich hergestellt hat, bleibt sie verbunden, es sei denn, es tritt eine der folgenden Ereignisse auf:

- Die AOR löscht ihre Definition der fernen Datei.

In diesem Fall wird die Verbindung sofort unterbrochen.

- Die FOR schließt oder inaktiviert die Datei.

In diesem Fall wird der Verbindungsabbau bei der nächsten Nicht-Aktualisierungsanforderung geplant und erfolgt, nachdem alle aktuellen Suchsequenzen beendet wurden. Siehe Verbindungsabbau.

Wenn diese Ereignisse später umgekehrt werden, wird wie zuvor eine gültige Verbindung hergestellt.

### **Benachrichtigung, dass eine neue Datei für den gemeinsamen Zugriff verfügbar ist**

Wenn eine Datentabelle von einer FOR geöffnet wird, wird sie für CONNECT-Versuche zu Beginn des Ladens (wenn es sich um eine CICS-verwaltete Datentabelle handelt) oder beim Abschluss des Ladens (wenn es sich um eine benutzerverwalteten Datentabelle handelt) zur Verfügung gestellt. Andere CICS-Regionen werden darüber informiert, dass eine Datentabelle verfügbar geworden ist. Eine Benachrichtigung erfolgt auch dann, wenn eine zuvor inaktivierte Datentabelle (oder eine Datei, die eine CICS-verwaltete Datentabelle verwendet) aktiviert wird.



---

## Kapitel 2. Gemeinsam genutzte Datentabellen konfigurieren

Sie können gemeinsam genutzte Datentabellen konfigurieren, um Dateien mithilfe von speicherübergreifenden Services gemeinsam zu nutzen. Die Einrichtung für gemeinsam genutzte Datentabellen ist eine Erweiterung zu den CICS-Dateiverwaltungsservices und kann die Leistung von Anwendungen verbessern, die eine Funktionsverlagerung verwenden.

### Die Verwendung von Datentabellen planen

Der Hauptgrund zur Verwendung von Datentabellen besteht darin, deren Leistungsvorteile zu nutzen.

Dieser Abschnitt enthält Informationen zur Diagnose, zu Modifikationen oder zur Optimierung.

### Leistung einer CICS-verwalteten Datentabelle

Wenn alle Daten- und Indexdatensätze einer Datei vollständig in einem LSR-Pool enthalten sind, wird die DASD-E/A-Aktivität durch die Definition der Datei als CICS-verwaltete Datentabelle nicht reduziert. Es besteht jedoch ein erhebliches Potenzial zur Reduzierung der CPU-Belegung. Möglicherweise können Sie auch die Anzahl der Puffer im LSR-Pool reduzieren.

Wenn die Datei nicht vollständig in einem LSR-Pool enthalten ist, kann die Verwendung einer von CICS-verwalteten Datentabelle zu einer Verringerung der DASD-E/A-Aktivität und der CPU-Belegung führen.

Eine geringere CPU-Belegung für eine CICS-verwaltete Datentabelle im Vergleich zu einer VSAM-KSDS, die in einem LSR-Pool resident ist, hängt von der Anwendungsnutzung ab.

### Leistung einer benutzerverwalteten Datentabelle

Nach dem Laden benutzerverwalteten Datentabelle wird die DASD-E/A-Aktivität aus allen Datentabellenoperationen entfernt. Die Verringerung der CPU-Belegung ist im Vergleich zu einer in einem LSR-Pool residenten VSAM KSDS daher beträchtlich.

### Speichernutzung für gemeinsam genutzte Datentabellen

Gemeinsam genutzte Datentabellen stellen die effiziente Nutzung von Daten im Speicher zur Verfügung. Dies bedeutet, dass erhebliche Leistungsvorteile für die Kosten einer zusätzlichen Nutzung des Speichers erzielt werden.

Diese Übersicht über die Speichernutzung setzt voraus, dass Sie die Unterscheidung zwischen verschiedenen Speichertypen, wie z. B. Realspeicher und virtueller Speicher, sowie Adressraumspeicher und Datenspeicherbereich verstanden haben. Der überwiegend genutzte Speicher ist der Datenspeicherbereich, der der virtuelle Speicher ist, der vom virtuellen Adressraumspeicher getrennt ist.

Gemeinsam genutzte Datentabellen verwenden den virtuellen Speicher wie folgt:

- Die Datensatzdaten werden in den Datenspeicherbereichen DFHDT003, DFHDT004, DFHDT005 usw. gespeichert, wobei die neuen Datenspeicherbereiche nach Bedarf zugeordnet werden. Der gesamte Datensatzspeicher ist beim Ladezeitpunkt im Wesentlichen die Gesamtgröße aller Datensätze (ohne die im Tabelleneintragsspeicher gespeicherten Schlüssel) plus wenige Steuerinformationen. Der Speicher für den Datenspeicherbereich wird in Einheiten von 16 MB erfasst und in 128-KB-Schritten einzelnen Tabellen zugeordnet. Der Speicher wird dann in seitenorientierten Frames untergeordnet, die groß genug sind, um die maximale Datensatzlänge für die Tabelle zu enthalten. Datentabellen-Frames sind grob äquivalent zu VSAM-Steuerintervallen und enthalten normalerweise eine Gruppe von Datensätzen mit ähnlichen Schlüsseln. Wenn möglich, wird jeder neue Datensatz im selben Frame wie der vorhandene Datensatz mit dem nächsten niedrigeren Schlüssel gespeichert.

Wenn viele Datensätze nach dem Laden länger werden oder neue Datensätze zufällig in einem großen Teil der Datei hinzugefügt werden, erhöht sich der Speicherplatz möglicherweise auf das Doppelte der ursprünglichen Größe.

- Der Deskriptorspeicher für den Tabelleneintrag wird dem Datenspeicherbereich DFHDT001 zugeordnet. Er wird in Inkrementen von 32 KB zugeordnet.

Es gibt einen Eintragsdeskriptor für jeden Datensatz in der Tabelle und einen Eintragsdeskriptor für jede Lücke in der Schlüsselfolge (wobei mindestens ein Datensatz aus einer CICS-verwalteten Datentabelle ausgeschlossen wurde). Die Größe der einzelnen Einträge ist die Schlüssellänge + 9 Byte, die auf das nächste Vielfache von 8 Byte aufgerundet werden.

- Der Indexknotenspeicher wird dem Datenspeicherbereich DFHDT002 zugeordnet. Er wird in Inkrementen von 32 KB zugeordnet.

Die Größe dieses Bereichs ist abhängig von der Verteilung und dem Format der Schlüsselwerte sowie der tatsächlichen Anzahl der Datensätze, wie in [Tabelle 1 auf Seite 12](#) angegeben.

<i>Tabelle 1. Schlüsselverteilung und -format</i>		
<b>Schlüsselverteilung</b>	<b>Schlüsselformat</b>	<b>Byte pro Datensatz</b>
Dense (alle Schlüssel sind aufeinanderfolgend)	binär dezimal alphabetisch	5,1 8,5 19
Sparse (keine Schlüssel sind aufeinanderfolgend)	dezimal alphabetisch	44 51
Im schlechtesten Fall	-	76

- Der ECSA-Speicher wird für einige kleine Steuerblöcke verwendet, auf die alle Regionen zugreifen müssen, die Datentabellen gemeinsam nutzen.

Das Konvertieren einer Datei in eine gemeinsam genutzte Datentabelle kann zu einer erhöhten Verwendung des Realspeichers führen, aber die Verwendung des Realspeichers für VSAM-LSR-Puffer kann reduziert werden, wenn wenige Aktualisierungen vorgenommen werden. Außerdem kann eine Anwendung, die derzeit eine hohe Leistung durch das Replizieren von schreibgeschützten Tabellen in jeder CICS-Region erzielt, möglicherweise große Speichereinsparungen erzielen, indem sie eine einzige Kopie jeder Tabelle gemeinsam nutzt.

Sobald Speicher einer Datentabelle zugeordnet wurde, bleibt sie dieser bestimmten Tabelle so lange zugeordnet, bis die Tabelle geschlossen wird. Wenn eine Datentabelle beispielsweise auf 1 GB anwächst und dann alle Datensätze aus der Tabelle gelöscht werden, verfügt die Tabelle noch über einen 1-GB-Datenspeicherbereich. Eine andere Datentabelle kann diesen Speicher erst dann verwenden, wenn die Owinging-Datentabelle geschlossen ist.

Freier Speicherplatz in einer Datentabelle wird verfolgt und gegebenenfalls wiederverwendet. Wenn beispielsweise Tabelleneintragsdeskriptoren oder Indexknoten nicht mehr benötigt werden, werden sie zu einer Kette verfügbarer Blöcke hinzugefügt, damit sie in derselben Tabelle wiederverwendet werden können. Und wenn alle Datensätze in einem Datensatzdaten-Frame gelöscht wurden, wird der leere Frame wieder in eine Kette verfügbarer Blöcke zurückgestellt. Wenn nur einige der Datensätze in einem Frame gelöscht wurden, wird der Speicherplatz nur erneut verwendet, wenn ein neuer Datensatz einen Schlüssel enthält, der unmittelbar auf einen anderen vorhandenen Datensatz im selben Frame folgt (oder auf den vorherigen Frame, wenn in diesem Frame kein Platz vorhanden ist). Im Gegensatz zu VSAM-Steuerintervallen sind die Datensätze in einem Frame nicht unbedingt in der Schlüsselfolge enthalten, da sie indirekt über Deskriptoren lokalisiert werden. Datensätze können nicht verschoben werden, um den freien Speicherbereich zu konsolidieren, da dies keine gleichzeitigen Zählerstände zulässt.

Wenn Datensätzen Schlüssel zugewiesen werden, die kontinuierlich zunehmen und in etwa derselben Reihenfolge gelöscht werden, wird der Speicherplatz in der Regel sehr effizient wiederverwendet, da neue Datensätze normalerweise einen Frame füllen, bevor sie zum nächsten weitergehen. Und alte Frames sind dann vollständig leer, sodass sie wiederverwendet werden können. Dies gilt auch für die Erhöhung von Schlüsseln in mehreren separaten Bereichen, vorausgesetzt, die Bereiche sind groß genug, um ganze Frames freizugeben. In dieser Situation entspricht die Menge des Speichers, der Datentabellen zugeordnet ist, fast nahezu der Menge an Speicher, die in Gebrauch ist.

Wenn neue Datentabellenanwendungen eingeführt werden, kann es hilfreich sein, den für jede Datentabelle zugewiesenen Speicher und den verwendeten Speicher zu überwachen, um sicherzustellen, dass ausreichend Betriebssystemressourcen verfügbar sind, um die aktuelle und zukünftige Nutzung zu unterstützen. Die Zählerstände für den zugeordneten Speicher zeigen den Speicher jeder Datentabelle an, der erst dann aufgegeben wird, wenn die Datentabelle gelöscht wird. Die Zählerstände des verwendeten Speichers zeigen an, wie viel des zugeordneten Speichers belegt ist. Das CICS-Beispielstatistikprogramm DFHOSTAT stellt diese Informationen zur Verfügung. DFHOSTAT wird in [Beispielstatistikprogramm DFHOSTAT](#) beschrieben.

Es ist möglich, dass für gemeinsam genutzte Datentabellen für Deskriptoren, Indexeinträge oder Daten kein Speicherplatz mehr vorhanden ist. Ein Mangel an Speicherplatz kann nicht nur beim Laden auftreten, sondern auch während der normalen Ausführung, wenn Datensätze hinzugefügt oder sogar aktualisiert werden. Da CICS jetzt mehrere Datenbereiche für die Unterstützung von gemeinsam genutzten Datentabellen verwendet, wurden die Grenzwerte für alle drei Speichertypen stark erhöht und sind von anderen Aspekten unabhängig, z. B. sind die Einträge nicht mehr im CICS-Adressraum enthalten. Dennoch ist der verfügbare Speicher weiterhin begrenzt. Zum Beispiel kann es eine extrem große Anzahl von relativ kleinen Datensätzen geben, insbesondere wenn sie hauptsächlich aus Schlüsseldaten bestehen. In diesem Fall kann entweder für die Eintragsdeskriptoren oder für die Indexknoten vor dem Speichern der Datensatzdaten selbst (je nach Schlüssellänge und anderen Faktoren) kein Speicherplatz mehr vorhanden sein. Wenn der Speicher für Eintragsdeskriptoren oder Indexknoten nicht ausreicht, sollten Sie die Datentabellen in verschiedene CICS-Regionen (z. B. verschiedene FORs) aufteilen. Wenn eine einzelne Datentabelle alleine nicht mehr über genügend Speicherplatz verfügt, wurde die für sie bestimmte Speichergrenze erreicht. In diesem Fall müssen Sie überlegen, ob sie in zwei oder mehr separate Tabellen aufgeteilt werden soll.

## **MVS-JCL-Anforderungen bei Verwendung von gemeinsam genutzten Datentabellen**

Bevor Sie gemeinsam genutzte Datentabellen verwenden können, müssen Sie möglicherweise einige Ihrer JCL-Anweisungen ändern, Ihre Betriebsprozesse ändern oder den Wert des Initialisierungsparameters MAXUSER MVS erhöhen.

Dies liegt daran, dass MVS bei einem Job nicht mehr als einen Schritt zulässt, der als gemeinsam genutzter Datentabellenserver ausgeführt wird. Wenn ein zweiter Jobschritt versucht, als gemeinsam genutzter Datentabellenserver zu agieren, gibt CICS die Nachricht DFHFC0405 aus. Da auch Jobschritte, die auf den Serverschritt folgen, keine speicherübergreifenden Services mit MRO verwenden können, wird empfohlen, darauf zu achten, dass es sich bei keinem der Jobschritte, die auf den Serverschritt folgen, um eine weitere Ausführung von CICS handelt.

Wenn ein Job, der einen Serverschritt für gemeinsam genutzte Datentabellen enthält, beendet wird, bevor alle mit diesem Server verbundenen Anforderer-Jobschritte beendet sind, wird der Serveradressraum von MVS beendet. Wenn der gemeinsam genutzte Datentabellenserver nicht als gestartete Task, sondern unter der Steuerung eines Batch-Initiators ausgeführt wird, muss in dieser Situation ein neuer Initiator gestartet werden.

MVS beendet den Batch-Initiator mit der Nachricht *IEF355A INITIATOR TERMINATED, RESTART INITIATOR*, da MVS aus Integritätsgründen andernfalls die Funktionen einschränken muss, die vom nächsten unter diesem Initiator ausgeführten Job verwendet werden könnten, wodurch der Job möglicherweise fehlschlägt. MVS erlaubt die Wiederverwendung der ASID eines gemeinsam genutzten Datentabellenservers erst dann, wenn alle mit dem Server verbundenen Anforderer-Jobschritte beendet sind.

## **Dateien für die Verwendung als Datentabellen auswählen**

Es ist nicht möglich, exakte Regeln festzulegen, ob eine Datei von der Konvertierung in eine gemeinsam genutzte Datentabelle profitieren wird. Die Checkliste in diesem Abschnitt enthält einige allgemeine Anleitungen.

Es gibt viele Überlegungen dazu. Eine Analyse der möglichen Verwendung der Unterstützung gemeinsam genutzter Datentabellen sollte von jemandem durchgeführt werden, der die Verwendung der Dateien in den verschiedenen Anwendungen und die Konfiguration der CICS-Regionen versteht.

Weitere Informationsquellen, die Sie bei der Auswahl der Dateien unterstützen könnten:

- Dateistatistik. In „Statistikdaten zum Auswählen von Datentabellen verwenden“ auf Seite 15 wird beschrieben, wie Sie Statistikdaten als eine der Eingaben für die Auswahl verwenden können.
- LSR-Pool-Statistik.
- Traceeinträge.
- Überwachungsdaten.

Die beste Eingabe für den Auswahlprozess ist jedoch ein gründliches Verständnis der Anwendungen und der Art und Weise, wie sie die Dateien verwenden.

Wenn Ihre Installation zum ersten Mal Datentabellen verwendet, enthält die folgende Checkliste einige allgemeine Prinzipien, die Sie bei der Auswahl von Dateien für die Definition als Datentabellen unterstützen.

- Sie sollten zuerst die Verwendung von CICS-verwalteten Datentabellen in Betracht ziehen, da diese einfacher implementiert werden können. Wenn Sie eine CICS-verwaltete Datentabelle verwenden, müssen an den Anwendungen keine Änderungen vorgenommen werden. Wenn Sie eine benutzerverwaltete Datentabelle verwenden, können einige Änderungen erforderlich sein.
- Verwenden Sie eine CICS-verwaltete Datentabelle, wenn Sie die Integrität der Datentabelle bei einem CICS-Neustart sicherstellen müssen.
- Verwenden Sie eine CICS-verwaltete Datentabelle, wenn Sie ein Journaling von Aktualisierungen benötigen. Wenn für alle Zugriffsanforderungen ein Journaling erforderlich ist, ist die Datei nicht als Datentabelle geeignet.
- Die Benutzerexits XEIIIN und XEIIOUT der EXEC-Schnittstelle sowie die Benutzerexits XFCREQ und XFCREQC der Dateisteuerung werden nicht in der File-Ownning Region (FOR) aufgerufen, wenn eine Anforderung zum Zugriff auf eine Datentabelle durch speicherübergreifende Services gegeben ist. Wenn Sie eine Datei auswählen, sollten Sie sicherstellen, dass der erfolgreiche Betrieb Ihrer Anwendung nicht von einer Aktivität abhängig ist, die an diesen Benutzerexits ausgeführt wird.
- Sie sollten sich der Sicherheitsauswirkungen auf die gemeinsame Nutzung einer Datentabelle bewusst sein, wie in „Sicherheitsprüfung für Datentabellen“ auf Seite 18 beschrieben.
- Wenn von einer anderen Region aus häufig auf eine Datei zugegriffen wird oder wenn sie von vielen anderen Regionen aufgerufen wird oder wenn die Zugriffe überwiegend Leseanforderungen sind, können die Vorteile überwiegen, aus der Datei eine Datentabelle zu machen. Beachten Sie, dass der Leistungsgewinn für eine ferne Datei größer ist als für eine lokale Datei.
- Wählen Sie bei einer CICS-verwalteten Datentabelle Dateien mit einer relativ hohen Anzahl an Anforderungen aus, die nur auf die Datentabelle zugreifen (siehe Entwicklung für Zugriff auf Datentabellen). Wählen Sie unter diesen Dateien die Dateien mit der intensivsten Nutzung dieser Anforderungen aus, um die Leistungssteigerung zu maximieren.

Informationen zur Dateiverwendung finden Sie in Dateisteuerungsstatistik in DFHSTUP-Berichten. Nicht alle Leseanforderungen profitieren von der Datentabelle, daher sollten Sie die Datentabelleninformationen im CICS-Statistikbericht nachlesen, um zu überprüfen, ob die Datentabelle effektiv verwendet wird. Weitere Informationen finden Sie in Datentabellen überwachen.

- Wählen Sie bei einer benutzerverwalteten Datentabelle Dateien mit hoher Aktualisierungsaktivität aus, bei deren Aktualisierungen es sich um Aktualisierungen handelt, die nicht bei einem CICS-Neustart wiederhergestellt werden müssen (siehe „Datenintegrität“ auf Seite 22).
- Verwenden Sie Leistungsmessungen, um die ungefähren CPU-Einsparungen zu schätzen, und berücksichtigen Sie dabei alle Prognosen für die zukünftige Verwendung.
- Wählen Sie eine oder zwei Dateien mit den realistischsten Schätzwerten aus. Geben Sie einer kleinen Datei Vorrang vor einer großen Datei, wenn die geschätzten Einsparungen ähnlich sind, da eine kleine Datei wahrscheinlich weniger Realspeicher verwendet.
- Überwachen Sie die Belegung Ihres Realspeichers. Wenn Ihr System bereits eine Einschränkung des Realspeichers aufweist, kann die Verwendung einer großen Datentabelle die Einlagerungsrate erhöhen. Dies kann sich wiederum negativ auf die Leistung des CICS-Systems auswirken. Verwenden Sie Ihre normalen Leistungsanalysetools, wie z. B. RMF (Version 5), um die Verwendung des Realspeichers und der Paging-Raten zu untersuchen.

- Ziehen Sie in Betracht, die Anzahl der Puffer im LSR-Pool zu reduzieren, da die Verwendung von Datentabellen die Häufigkeit der Nutzung des LSR-Pools reduzieren könnte.
- Mit dem Benutzerexit XDTRD können Sie die Datensätze auswählen, die in der Datentabelle enthalten sind. Außerdem können Sie für eine benutzerverwaltete Datentabelle den Benutzerexit XDTRD verwenden, um die Datensätze zu ändern. Sie können so die Verwendung von virtuellem und realen Speicher optimieren, indem Sie in der Datentabelle nur die Daten speichern, die Sie benötigen.
- Eine sehr große Datentabelle benötigt möglicherweise einen größeren Datenspeicherbereich als die vom Exit MVS IEFUSI festgelegte normale Regionsbegrenzung. In diesem Fall können Sie den Grenzwert erhöhen, indem Sie den Exit IEFUSI ändern oder das globale CICS-Benutzerexitprogramm XDTRD verwenden, um einige Datensätze zu unterdrücken. Der Exit IEFUSI wird in der Veröffentlichung *z/OS MVS Installation Exits* (IBM Formnummer SA22-7593) beschrieben.

## Statistikdaten zum Auswählen von Datentabellen verwenden

Wenn Ihre gemeinsame Nutzung auf ein einzelnes MVS-Image beschränkt ist, sollten Sie überlegen, welche Dateien Zugriffsmuster haben, die die Verwendung von gemeinsam genutzten Datentabellen vorteilhaft machen.

Wenn Sie Daten zwischen mehr als einem MVS-Image gemeinsam nutzen müssen, sollten Sie den RLS-Modus verwenden, um die Dateien freizugeben.

Abbildung 3 auf Seite 15, Abbildung 4 auf Seite 16 und Abbildung 5 auf Seite 16 zeigen einige Ausschnitte aus einem hypothetischen Satz von Dateistatistiken für Dateien, auf die im Nicht-RLS-Modus zugegriffen wird. Diese werden in der folgenden Beschreibung verwendet, um zu zeigen, wie CICS-Statistiken das Auswahlverfahren unterstützen können.

Die Statistiken werden so angezeigt, wie sie auch vom CICS-Offline-Formatierungsdienstprogramm gemeldet werden. Die angeforderten Dateistatistikdaten werden angezeigt, aber die Statistiken für "Intervall" oder "Ende der Tageszeit" sind ebenfalls geeignet. Der Abschnitt der Datei mit der "Performance Information"-Statistik, in dem die Verwendung von VSAM-Zeichenfolgen und Puffern gemeldet wird, wird hier nicht angezeigt.

Die in den Abbildungen angegebenen Zahlen dienen lediglich zur Veranschaulichung, und Sie sollten nicht davon ausgehen, dass die Statistiken in Ihrer Installation diesen entsprechen. In ähnlicher Weise wurde die Konfiguration von CICS-Regionen und -Dateien ausgewählt, um bestimmte Punkte hervorzuheben. Dies impliziert nicht, dass es sich dabei um eine typische oder wünschenswerte Konfiguration handelt.

In Datentabellen überwachen werden die Statistiken beschrieben, die für Dateien angegeben wurden, die als Datentabellen definiert sind. Mit diesen können Sie die erzielten Vorteile bewerten.

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

Abbildung 3. Von C1CFOR angeforderte Dateistatistiken

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

Abbildung 4. Von CICAOR1 angeforderte Dateistatistiken

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

Abbildung 5. Von CICAOR2 angeforderte Dateistatistiken

In den Beispielen wird eine hypothetische Konfiguration von drei CICS-Regionen verwendet. Die meisten Dateien, die von CICS-Anwendungen verwendet werden, sind Eigentum der File-Owning Region CICFOR und die Anwendungen werden hauptsächlich in den Application-Owning Regionen CICAOR1 und CICAOR2 ausgeführt. Bei dieser Diskussion wird davon ausgegangen, dass es sich bei allen in den Statistikberichten angezeigten Datensätzen um ein VSAM-Basis-KSDS handelt (wie durch Data Set Type von K angegeben), sodass jeder von ihnen als Datentabellen definiert werden kann.

Dieser Abschnitt konzentriert sich auf die Identifizierung von Kandidaten, die als CICS-verwaltete Datentabellen definiert werden sollen, da die Entscheidung, eine benutzerverwaltete Datentabelle zu definieren, eher im Hinblick auf bestimmte Anwendungen als bei einer Untersuchung der Dateileistung im Allgemeinen gefällt wird. Aufgrund dieses Fokus gilt keine der angezeigten Statistiken für Dateien, auf die im RLS-Modus zugegriffen wird, da ein Datensatz im RLS-Modus nicht die Quelle für eine CICS-verwaltete Datentabelle sein kann.

Die Statistikdaten zeigen außerdem an, welche Dateinamen in einer Region für den Zugriff auf Dateinamen in einer anderen Region definiert sind. Die *Remote Sysid* ist der Name, der in der Verbindung zwischen den beiden Regionen angegeben wurde. In den Beispielen lautet die SYSID von CICFOR "CIF2" und die von CICAOR2 lautet "CIA2".



### **Eine Datei mit einem hohen Lese-Aktualisierungs-Rate**

Die Datei APPLE wird von Anwendungen verwendet, die in der Application-Owning Region CICAOR1 ausgeführt werden. Sie wird in CICAOR1 als ferne Datei definiert, und die Dateidefinition verweist auf die Datei APPLE, die CICFOR gehört.

Diese Datei würde von einer Neudefinition in CICFOR als CICS-verwaltete Datentabelle profitieren, da sie einen hohen Wert für das Verhältnis von Fernlesevorgängen (1158701 Get Requests in dem von den Berichten abgedeckten Zeitraum) zu Fernaktualisierungen (11 Hinzufügungen, 1 Löschen und 531 Updates) aufweist, wie in Abbildung 4 auf Seite 16 beschrieben.

Weitere Informationen zu den Bedeutungen des Abschnitts "FILES - Requests Information" eines Statistikberichts finden Sie unter Dateisteuerungsstatistik in DFHSTUP-Berichten.

### **Eine Datei mit einem hohen Anteil an fernen Lesevorgängen**

Die Datei BANANA wird aktualisiert und von CICFOR gelesen, sie wird aber auch von CICAOR1 aufgerufen.

Da es sich bei allen Fernzugriffen um Lese- und Suchvorgänge ohne Aktualisierungen handelt, würden die in CICAOR1 ausgeführten Anwendungen wahrscheinlich große Vorteile haben, wenn BANANA als Datentabelle definiert wäre, und die Anwendungen auf CICFOR würden auch vom Lesen aus der lokalen Datentabelle profitieren.

### **Eine Datei, die von mehreren Regionen gemeinsam genutzt wird**

Es kann sein, dass ORANGE kein besonders geeigneter Datentabellenkandidat ist.

Die Statistiken in Abbildung 4 auf Seite 16 zeigen, dass die Anzahl der fernen Abrufe von CICAOR1 (58709 Get Requests und 4265 Browse Requests) relativ niedrig ist. Die ferne Datei LEMON in CICAOR2 verweist jedoch auch auf ORANGE in CICFOR, sodass die Definition von ORANGE in CICFOR als gemeinsam genutzte CICS-verwaltete Datentabelle wahrscheinlich für die Leistung der Anwendungen in beiden AORs von Vorteil ist.

### **Ein guter Kandidat für eine benutzerverwaltete Datentabelle (UMT)**

Auf die Datei COURGETT, deren Eigner CICAOR2 ist, wird über den Dateinamen UCCHINI in CICAOR1 zugegriffen.

CICAOR1 liest oder durchsucht nur die Datei. Die Aktualisierung wird von der Owning Region ausgegeben. Außerdem ist bekannt, dass diese Aktualisierungen nur für die CICS-Ausführung des Tages relevant sind und nicht permanent beibehalten werden müssen. (Sie werden beim Herunterfahren tatsächlich gelöscht.) Die Datei ist deshalb ein ausgezeichnete Kandidat für die Definition als benutzerverwaltete Datentabelle. Alle Aktualisierungen können dann ohne VSAM-E/A-Aktivität an der Datentabelle vorgenommen werden, und alle fernen Abrufe können ohne Funktionsverlagerung ausgeführt werden.

### **Ein ziemlich schlechter Kandidat**

Die Datei PEAR profitiert wahrscheinlich nicht viel von der Unterstützung gemeinsam genutzter Datentabellen, da auf sie nicht über Fernzugriff zugegriffen wird und viele Aktualisierungs- und Anzeigeanforderungen vorhanden sind.

Das lokale Browsing bietet nicht so viele Vorteile wie das lokale Lesen oder jede Form des fernen Abrufs, da das VSAM-Browsing (abgesehen von der Verarbeitung des Befehls STARTBR) sehr effizient ist. Diese Analyse berücksichtigt natürlich nicht die relative Wichtigkeit der verschiedenen Dateizugriffe. Das Lesen kann von kritischen Anwendungen ausgeführt werden, die Zeit, die für Aktualisierungen benötigt wird, ist jedoch möglicherweise nicht von Bedeutung.

### **Andere mögliche Kandidaten**

Die vorhergehenden Beispiele veranschaulichen nur eine kleine Auswahl der möglichen Konfigurationen und Verwendungen von Dateien, die von der Unterstützung für gemeinsam genutzte Datentabellen profitieren könnten.

Sie können auch die Unterstützung für gemeinsam genutzte Datentabellen verwenden, um zu vermeiden, dass Dateien oder Datentabellen in jeder Region dupliziert werden müssen. Zusätzlich zu den vorhandenen Dateien können Sie das Verschieben von Dateien von einer Application-Owning Region (AOR) in eine File-Owning Region (FOR) in Betracht ziehen. Das Verschieben von Dateien von einer AOR in

eine FOR war nicht praktisch, als die Unterstützung für gemeinsam genutzte Datentabellen aufgrund der Kosten für Dateizugriffe mit Funktionsverlagerung nicht verfügbar war.

## Sicherheitsprüfung für Datentabellen

Gemeinsam genutzte Datentabellen führen Sicherheitsprüfungen zum Zeitpunkt der LOGON- oder CONNECT-Operationen durch, um die Sicherheit zu gewährleisten, wenn speicherübergreifende Services verwendet werden. Sie sollten die Auswirkungen der Sicherheitsprüfungen berücksichtigen, bevor Sie eine Datei gemeinsam nutzen, die einer Datentabelle zugeordnet ist.

Gemeinsam genutzte Datentabellen müssen sicherstellen, dass:

- Die FOR kann keine Identität annehmen. Dies wird dadurch verhindert, dass beim Zeitpunkt der LOGON-Operation geprüft wird, ob die FOR-Anmeldung mit der angegebenen generischen *applid* der CICS-Region zulässig ist.
- Eine Application-Ownning Region (AOR) kann keinen Zugriff auf Daten erhalten, die sie nicht sehen darf. Dies wird dadurch verhindert, dass beim Zeitpunkt der CONNECT-Operation geprüft wird, ob die AOR auf die File-Ownning Region (FOR) zugreifen darf, wenn die Dateisicherheit in Kraft ist, und ob die AOR auf die angeforderte Datei zugreifen darf.

Diese Sicherheitsprüfungen werden mithilfe der Systemberechtigungsfunktion (SAF) ausgeführt, indem die Resource Access Control Facility (RACF) oder ein funktional entsprechender Sicherheitsmanager verwendet wird.

**Anmerkung:** Eine Region kann die Datentabellen weiterhin lokal verwenden, selbst wenn sie nicht über die Berechtigung als gemeinsam genutzter Datentabellenserver verfügt.

Die Unterstützung für gemeinsam genutzte Datentabellen reproduziert die Hauptmerkmale der Sicherheit für die Funktionsverlagerung, die auf Regionsebene ausgeführt werden. Die folgenden Unterschiede sollten jedoch beachtet werden:

- Die Unterstützung für gemeinsam genutzte Datentabellen stellt keinen Mechanismus für die FOR zur Ausführung von Sicherheitsprüfungen auf Transaktionsebene bereit (entspricht ATTACHSEC(IDENTIFY) oder ATTACHSEC(VERIFY)). Wenn Sie der Ansicht sind, dass die von der AOR durchgeführten Prüfungen auf Transaktionsebene für einige Dateien unzulänglich sind, müssen Sie daher sicherstellen, dass diese Dateien nicht mit Datentabellen in der FOR verknüpft sind.
- Die Unterstützung für gemeinsam genutzte Datentabellen unterstützt keine voreingestellte Sicherheit.
- Die Unterstützung für gemeinsam genutzte Datentabellen gibt keine Informationen zur Installationsparameterliste (INSTLN) an die Sicherheitsbenutzerexits weiter.

Eine Beschreibung der erforderlichen Schritte zum Implementieren der Sicherheit für gemeinsam genutzte Datentabellen finden Sie unter [RACF-Klassen zum Schutz von Systemressourcen](#).

### LOGON-Sicherheitsprüfung

Die LOGON-Verarbeitung enthält eine Sicherheitsprüfung, um zu prüfen, ob die File-Ownning Region (FOR) berechtigt ist, als Server mit dem angegebenen Anwendungsnamen zu fungieren.

Durch diese Prüfung wird das Risiko minimiert, dass eine Application-Ownning Region (AOR) gefälschte Datensätze aus einer File-Ownning Region (FOR) akzeptiert, die in Wirklichkeit ein Betrüger ist. Die Prüfung wird nie umgangen, selbst wenn bei der Systeminitialisierung SEC=NO angegeben ist.

### CONNECT-Sicherheitsprüfungen

Die Sicherheitsprüfungen, die zum Zeitpunkt der CONNECT-Operation ausgeführt werden, stellen zwei Sicherheitsstufen zur Verfügung.

#### Bindungssicherheit

Ermöglicht einer FOR (File-Ownning Region), die ohne CICS-Dateisicherheit ausgeführt wird, den gemeinsamen Zugriff auf ausgewählte AORs (Application-Ownning Region). (Die Ausführung ohne Dateisicherheit minimiert den Laufzeitaufwand und die Anzahl der Sicherheitsdefinitionen.)



## **Dateisicherheit**

Kann in der FOR aktiviert werden, wenn Sie eine feinere Granularität der Sicherheitsprüfung benötigen. Die Unterstützung für gemeinsam genutzte Datentabellen implementiert dann die Prüfungen, die für die AOR als Ganzes gelten.

Die Unterstützung für gemeinsam genutzte Datentabellen bietet keine Möglichkeit, diese Sicherheitsprüfungen zu implementieren, die eine FOR auf Transaktionsebene durchführt, wenn ATTACHSEC(IDENTIFY) oder ATTACHSEC(VERIFY) mit der Funktionsverlagerung verwendet wird.

## **Verwendung der Unterstützung für gemeinsam genutzte Datentabellen vorbereiten**

Wenn Sie die Unterstützung für gemeinsam genutzte Datentabellen verwenden möchten, müssen Sie die folgenden Tasks ausführen. Einige von ihnen werden bereits für eine Installation ausgeführt, die derzeit eine Funktionsverlagerung oder Datentabellen verwendet.

### **Informationen zu diesem Vorgang**

- Stellen Sie entweder sicher, dass sich die folgenden Module in einer autorisierten Systembibliothek in der LNKST des MVS-Systems befinden, oder verschieben Sie sie in eine Bibliothek in der LPALST-Verkettung.
  - DFHDTSVC und DFHDTVCV, da alle Regionen, die gemeinsam genutzte Datentabellen verwenden, dieselbe Ebene des SVC-Codes verwenden müssen.
  - DFHMVRMS, der RESMGR-Exit-Stub, da CICS JOBLIB/STEPLIB-Dateien am Speicherende nicht verfügbar sind.
- Die folgenden Module werden von der CICS-Installation in die Zielbibliothek SDFHLINK gestellt, die normalerweise in der LNKST-Verkettung enthalten ist.
  - Wenn SDFHLINK in der LNKST-Verkettung enthalten ist, sollten Sie den Bedienerbefehl MODIFY LLA, REFRESH absetzen und auf die Bestätigungsnachricht CSV210I LIBRARY LOOKASIDE REFRESHED warten, um die Module verfügbar zu machen.
  - Wenn SDFHLINK nicht in der LNKST-Verkettung enthalten ist, sollten Sie die Module entweder in eine geeignete eingeschlossene Bibliothek kopieren und einen Befehl für LLA-Aktualisierung ausgeben oder die Module in eine Bibliothek in der LPALST-Verkettung kopieren und das MVS-System mit CLPA wieder anlaufen lassen.
- Wenn Dateien in einer AOR (Application-Ownning Region) gemeinsam genutzt werden sollen, stellen Sie sicher, dass CICS als MVS-Subsystem definiert ist.
- Definieren Sie die Sicherheitsberechtigung, so dass FORs (File-Ownning Region) als gemeinsam genutzte Datentabellenserver agieren können, und AORs können auf Dateien zugreifen, deren Eigner Server sind, abhängig von der erforderlichen Sicherheitsstufe. In einem einzelnen MVS-Image:
  - kann eine beliebige Anzahl von FORs als gemeinsam genutzter Datentabellenserver verwendet werden.
  - kann eine einzelne AOR eine beliebige Anzahl dieser FORs verwenden.
  - kann eine einzelne FOR eine beliebige Anzahl von AORs bedienen.
  - kann eine Region als AOR für eine Datentabelle und als FOR für eine andere Datentabelle fungieren.
- Wenn zwei FORs dieselbe APPLID aufweisen sollten, wird zu einem beliebigen Zeitpunkt nur eine dieser FORs als gemeinsam genutzter Datentabellenserver verwendet. Es gibt jedoch keine Möglichkeit, die verhindert, dass eine FOR als gemeinsam genutzter Datentabellenserver und eine andere FOR mit derselben APPLID für funktionsverlagerte Anforderungen verwendet werden. Sie sollten prüfen, dass Ihre Betriebsprozesse dies nicht zulassen. Es besteht sonst das Risiko, dass Datentabellanforderungen, die gemeinsam genutzte Datentabellenservices verwenden, nicht in dieselbe Region übertragen werden wie Anforderungen, die eine Funktionsverlagerung verwenden.
- Definieren Sie in der FOR die Dateien, die Datentabellen sind, entweder als CICS-verwaltete Datentabellen oder als benutzerverwaltete Datentabellen.
- Erstellen Sie bei Bedarf zusätzliche ferne Dateidefinitionen in der AOR. Für vorhandene ferne Dateidefinitionen sind keine Änderungen erforderlich.

- Geben Sie für jede AOR, die Datentabellen gemeinsam nutzen soll, ISC=YES als Systeminitialisierungsparameter an und definieren Sie MRO- oder ISC-Links zu den relevanten FORs. Geben Sie für IP-Interkonnektivitäts-Verbindungen (IPIC) den äquivalenten Systeminitialisierungsparameter TCP/IP=YES an und definieren Sie einen IPIC-Link zur relevanten FOR.
- Bevor Sie gemeinsam genutzte Datentabellen verwenden können, müssen Sie möglicherweise einige Ihrer JCL-Anweisungen ändern, Ihre Betriebsprozesse ändern oder den Wert des Initialisierungsparameters MAXUSER MVS erhöhen. Weitere Informationen finden Sie in „MVS-JCL-Anforderungen bei Verwendung von gemeinsam genutzten Datentabellen“ auf Seite 13.

### Lademodule

Die in diesem Abschnitt aufgelisteten Lademodule müssen in Ihrer CICS-Region installiert sein, damit gemeinsam genutzte Datentabellen verwendet werden können.

<i>Tabelle 2. Lademodule, die von der Unterstützung für gemeinsam genutzte Datentabellen verwendet werden</i>			
<b>Lademodul</b>	<b>Ladebibliothek</b>	<b>Vorgehensweise beim Laden</b>	<b>Beschreibung</b>
DFHDTINS	SDFHLOAD	CICS-Ladevorgang oberhalb der 16-MB-Grenze	Initialisierung
DFHDTVC	SDFHLINK	MVS LOAD oberhalb der 16-MB-Grenze aus Linkliste	Führt alle Funktionen aus, die MVS-Berechtigungen erfordern
DFHDTFOR	SDFHAUTH	MVS LOAD oberhalb der 16-MB-Grenze	Datentabelle-FOR-Modul
DFHDTAM	SDFHAUTH	MVS LOAD in den Subpool-252-Speicher oberhalb der 16-MB-Grenze	Datentabellen-Zugriffsmanager. Enthält Code, der von einer Application-Ownning Region (AOR) im speicherübergreifenden Modus ausgeführt wird.
DFHDTAOR	SDFHAUTH	MVS LOAD oberhalb der 16-MB-Grenze	Datentabelle-AOR-Modul
DFHDTVCV	SDFHLINK	MVS LOAD in ECSA aus Linkliste	Verbindungsvalidierung (AOR)
DFHDTXS	SDFHAUTH	MVS LOAD in ECSA	Prüfung der Verbindungssicherheit (FOR)
DFHDMVRMS	SDFHLINK	MVS LOAD oberhalb der 16-MB-Grenze aus Linkliste	EOT/EOM-Schnittstellencode des Ressourcenmanagers

### Speicherbelegung

Die Gesamtgröße der Module, die Speicher oberhalb der 16-MB-Grenze belegen, beträgt ca. 41 KB. Für Module, die sich im ECSA-Speicher befinden, sind für jede angemeldete FOR-Operation ca. 1,5 KB und für jede Application-Ownning Region (AOR) ca. 0,5 KB erforderlich.

Die Module sind alle für die Aufnahme in den Link-Pack-Bereich (LPA) auswählbar, aber nur DFHDTFOR, DFHDTAM, DFHDTAOR und möglicherweise DFHDTVCV werden ausreichend häufig verwendet und sind eine Überlegung wert.

## Ressourcendefinition für Datentabellen

Sie definieren eine Datentabelle auf dieselbe Weise wie eine CICS-Datei, mit der Ausnahme, dass Sie auch den Typ der zu verwendenden Datentabelle angeben müssen und die maximale Anzahl der Datensätze, die in der Datentabelle enthalten sein können.

Die VSAM-KSDS-Definition stellt die maximale Datensatzlänge und Schlüssellänge bereit.

Sie können eine Datei als Datentabelle definieren, indem Sie den Befehl CEDA DEFINE FILE verwenden, wie in „Datentabellen mit dem Befehl DEFINE FILE definieren“ auf Seite 22 beschrieben.

Um die Datentabellenattribute einer vorhandenen Datei zu ändern oder zu überprüfen, können Sie Folgendes verwenden:

- Befehle EXEC CICS SET FILE und INQUIRE FILE (siehe „EXEC CICS-Befehle für Datentabellen“ auf Seite 26)
- Befehle CEMT SET FILE und INQUIRE FILE (siehe „CEMT-Befehle für Datentabellen“ auf Seite 27)

## Ressourcendefinition für CICS-verwaltete Datentabellen

Für eine CICS-verwaltete Datentabelle kann entweder das Format für einen Datensatz variabler Länge oder für einen Datensatz fester Länge angegeben werden.

Die maximale Datensatzlänge, die von der Unterstützung für gemeinsam genutzte Datentabellen verwendet werden kann, beträgt 32 KB. Diese Länge überschreitet die von der CICS-Dateiverwaltung unterstützte Länge, die somit die tatsächliche Begrenzung festlegt. Siehe [Länge von an CICS-Befehle übergebenen Bereichen](#). Die maximale Anzahl der Datensätze, die unterstützt wird, beträgt 16.777.215.

Nur der VSAM-Basiscluster kann als Basis für eine CICS-verwaltete Datentabelle (CMT) dienen. Leseanforderungen über alternative Indexpfade verwenden die Datentabelle nicht, aber Änderungen an der Quellendatei über alternative Indexpfade werden in die Datentabelle übernommen.

Beachten Sie, dass die Quellendatei für eine CICS-verwaltete Datentabelle im RLS-Zugriffsmodus nicht geöffnet werden kann. Daher muss die Dateidefinition RLSACCESS(NO) angeben, sodass alle anderen Dateien derselben Basisdatei zugeordnet werden.

Nach dem Öffnen einer Datei, die als CICS-verwaltete Datentabelle (CMT) definiert ist, verwendet jede andere Nicht-UMT-Datei (unabhängig davon, ob sie als CMT definiert ist), die dieselbe Quellendatei in der zugehörigen Definition angibt, automatisch dieselbe Datentabelle. Wenn eine dieser anderen Dateien als CMT definiert ist, wird beim Öffnen die Nachricht DFHFC0937 an die Konsole ausgegeben. Dies ist keine Fehlersituation. Die Dateien werden geöffnet und verwenden die vorhandene Datentabelle, wann immer dies möglich ist.

### VSAM SHAREOPTION

Wenn die Quellendatei zu DISP=SHR zugeordnet ist, besteht die Gefahr, dass es durch eine andere Region als die File-Ownning Region (FOR) aktualisiert wird. Wenn das passiert ist, stimmt die Datentabelle nicht mehr mit der Quellendatei überein. Um dieses Risiko zu minimieren, sollte die regionsübergreifende VSAM SHAREOPTION auf 1 oder 2 gesetzt werden.

- 1 bedeutet, dass entweder eine Region über einen Aktualisierungszugriff auf die Datei verfügt oder dass viele Regionen nur über Lesezugriff verfügen.
- 2 bedeutet, dass eine Region über einen Aktualisierungszugriff auf die Datei verfügt und dass gleichzeitig viele Regionen nur über Lesezugriff verfügen.

Unabhängig von der Einstellung von DISP wird eine Warnung ausgegeben, wenn die regionsübergreifende SHAREOPTION 3 oder 4 ist, oder wenn es sich um 2 handelt, aber die CICS-verwaltete Datentabelle über Lesezugriff verfügt (was bedeutet, dass eine andere Region in der Lage sein könnte, die Datei zu aktualisieren).

### Datenintegrität

Eine Datei, die eine CICS-verwaltete Datentabelle verwendet, kann als wiederherstellbare Ressource definiert werden. Die Quellendatei wird nach einem System- oder Transaktionsfehler auf normale Weise wiederhergestellt.

- Nach einem Systemausfall wird die Datentabelle beim erneuten Öffnen der Datei aus der wiederhergestellten Quellendatei neu geladen.
- Nach einem Transaktionsfehler werden die Änderungen an der Quellendatei durch dynamisches Zurücksetzen von Transaktionen auch in der Datentabelle vorgenommen.

Das automatische Journaling wird (in derselben Weise wie für jede andere Datei) für Dateioperationen unterstützt, die auf die Quellendatei zugreifen. Dateioperationen, die nicht auf die Quellendatei zugreifen, werden im nicht Journal aufgezeichnet.

## Ressourcendefinition für benutzerverwaltete Datentabellen

Für eine benutzerverwaltete Datentabelle (UMT) muss das Format für einen Datensatz variabler Länge angegeben werden.

Die maximale Datensatzlänge, die von der Unterstützung für gemeinsam genutzte Datentabellen verwendet werden kann, beträgt 32 KB. Diese Länge überschreitet die von der CICS-Dateiverwaltung unterstützte Länge, die die tatsächliche Begrenzung festlegt. Siehe [Länge von an CICS-Befehle übergebenen Bereichen](#). Die maximale Anzahl der Datensätze, die unterstützt wird, beträgt 16 777 215.

Die Quellendatei für eine benutzerverwaltete Datentabelle kann im RLS-Zugriffsmodus geöffnet werden. Möglicherweise möchten Sie eine RLS-Modus-Datei als Quelle einer benutzerverwalteten Datentabelle definieren, wenn Sie über andere Dateidefinitionen verfügen, die auf die Datei zugreifen, und die Datei von anderen CICS-Regionen aktualisiert wird.

Sie können mehrere benutzerverwaltete Datentabellen aus derselben Quellendatei laden, indem Sie die einzelnen Datentabellen mit einem separaten Befehl oder Makro definieren und alle Definitionen auf diese Datei verweisen lassen.

Obwohl eine Datentabelle aus einer VSAM-KSDS geladen werden muss, kann eine Anwendung anschließend Datensätze aus jeder Datenquelle, die über den CICS-Adressraum zugänglich ist, in eine benutzerverwaltete Datentabelle kopieren. Hierbei kann es sich um eine IMS- oder Db2-Datei handeln. Die KSDS (Datei in Schlüsselstelle), die als Quellendatei für die Datentabelle verwendet wird, kann leer sein. Sie ist nur erforderlich, um die maximale Datensatzlänge sowie die Schlüssellänge und -position zu definieren.

### Datenintegrität

Eine benutzerverwaltete Datentabelle kann als wiederherstellbare Ressource definiert werden. Änderungen an der Datentabelle werden nicht im Systemprotokoll aufgezeichnet, jedoch intern im CICS-Speicher gespeichert. Daher kann die Datentabelle nach einem Transaktionsfehler (durch dynamisches Zurücksetzen), allerdings nicht nach einem Systemausfall, wiederhergestellt werden.

Dies liegt daran, dass die CICS-Einrichtung für gemeinsam genutzte Datentabellen ihre eigene Wiederherstellung verwaltet und nicht die Services des Protokollmanagers oder des Wiederherstellungsmanagers verwendet. Die Ausnahme ist, wenn Änderungen an einer wiederherstellbaren Datentabelle als Teil einer verteilten Arbeitseinheit (DUOW) vorgenommen werden. In diesem Fall wird wie bei anderen wiederherstellbaren Ressourcen, ein Datensatz der Verbindung als Teil des zweiphasigen Festschreibungsprozesses in das Systemprotokoll geschrieben. Die Änderungen selbst werden jedoch nicht im Systemprotokoll aufgezeichnet.

Nach einem Systemausfall wird die Datentabelle beim erneuten Öffnen der Datei aus der Quellendatei neu geladen. Beachten Sie, dass der Inhalt der Quellendatei und der Datentabelle zum Zeitpunkt des Fehlers nicht identisch gewesen wäre, es sei denn, Sie hätten Folgendes sichergestellt:

- Es wurde in beiden keine Änderung vorgenommen, oder
- Es wurde Änderungen in beiden vorgenommen.

Das automatische Journaling wird nur für Anforderungen unterstützt, die während des Ladens auf die Quellendatei zugreifen. Die Datensätze, auf die durch den Ladeprozess zugegriffen wird, werden vor dem Benutzerexit XDTRD aufgezeichnet, und die Datensätze, auf die aufgrund von Anwendungsanforderungen zugegriffen wird, werden nach dem Benutzerexit XDTRD aufgezeichnet.

## Datentabellen mit dem Befehl **DEFINE FILE** definieren

Mit dem Befehl **DEFINE FILE** können Sie eine Datei als CICS-verwaltete Datentabelle oder als benutzerverwaltete Datentabelle definieren.

Vollständige Details zu FILE-Definitionen finden Sie in [FILE-Ressourcen](#). In diesem Thema werden nur die Attribute beschrieben, die sich auf Datentabellen beziehen.

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

Geben Sie **TABLE (CICS)** an, um die Datei als CICS-verwaltete Datentabelle zu definieren.

Geben Sie **TABLE (USER)** an, um die Datei als eine benutzerverwaltete Datentabelle zu definieren.

Wenn Sie weder den Parameter TABLE noch **TABLE(NO)** oder **TABLE(CF)** angeben, wird die Datei nicht als gemeinsam genutzte CICS-Datentabelle definiert.

**MAXNUMRECS(NOLIMIT|number)**

Gibt die maximale Anzahl der Datensätze im Bereich von 1 bis 99999999 an, die in der Datentabelle enthalten sein können. Der Standardwert gibt keine Begrenzung der maximalen Anzahl der Datensätze an.

**FILE(name)**

Gibt den Namen der Datei an.

Bei einer CICS-verwalteten Datentabelle wird dieser Name verwendet, um sowohl auf die Datentabelle als auch auf die Quellendatei zu verweisen, die von CICS als einzelne Entität behandelt werden.

Bei einer benutzerverwalteten Datentabelle wird dieser Name verwendet, um nur auf die Datentabelle zu verweisen.

**DSNAME(name)**

Gibt den Namen der VSAM-KSDS an, die als Quellendatei verwendet werden soll. Hierbei muss es sich um eine Basisdatei handeln, nicht um einen Pfad oder eine Alternativindexdatei. Wenn der Quellendatei ein Pfad oder ein Alternativindex zugeordnet ist, werden alle Aktualisierungen für eine CICS-verwaltete Datentabelle, die über die Datei durchgeführt wurden, sowohl in der Quellendatei als auch in den zugehörigen Alternativindizes abgebildet. Bei einer benutzerverwalteten Datentabelle (UMT) werden die Aktualisierungen weder in der Quellendatei noch in den zugehörigen Alternativindizes abgebildet. Nach dem Abschluss des Ladevorgangs ist eine benutzerverwaltete Datentabelle vollständig unabhängig von der zugehörigen Quellendatei.

**LSRPOOLID(number|1)**

Dieses Attribut ist veraltet, wird jedoch unterstützt, um die Kompatibilität mit früheren Releases von CICS zu gewährleisten.

**LSRPOOLNUM(number|1|NONE)**

Gibt die Nummer des VSAM-LSR-Pools (LSR; Local Shared Resource, gemeinsam genutzte lokale Ressource) an, der von der Datentabelle verwendet werden soll. Sie müssen eine LSRPOOL-Nummer im Bereich von 1 bis 255 angeben. Der Standardwert ist 1, es sei denn, es wurde ein Wert für das Attribut NSRGROUP angegeben. In diesem Fall lautet der LSRPOOLNUM-Standardwert NONE.

**OPENTIME({FIRSTREF|STARTUP})**

Gibt an, wann die Datei beim ersten Verweis oder unmittelbar nach dem Start durch die automatisch eingeleitete Transaktion CSFU geöffnet werden soll. **OPENTIME(FIRSTREF)** wird standardmäßig angenommen.

Denken Sie daran, dass die Datentabelle geladen wird, wenn die Datei geöffnet wird. Wenn Sie also den Benutzerexit XDTRD verwenden, stellen Sie sicher, dass der Benutzerexit aktiviert ist, bevor die Datei geöffnet wird (siehe [Benutzerexits für Datentabellen aktivieren](#)).

**RECORDFORMAT({V|F})**

Gibt das Format der Datensätze in der Datei an: Entweder **RECORDFORMAT(V)** für Datensätze variabler Länge oder **RECORDFORMAT(F)** für Datensätze fester Länge.

**RECORDFORMAT(V)** wird standardmäßig angenommen. Eine benutzerverwaltete Datentabelle muss Datensätze variabler Länge haben.

**ADD(NO|YES), BROWSE(NO|YES), DELETE(NO|YES), READ(YES|NO) und UPDATE(NO|YES)**

Gibt die Dateioperationen an, die für die Datentabelle angefordert werden können.

**RECOVERY({NONE|BACKOUTONLY|ALL})**

Gibt die Art der Wiederherstellungsunterstützung an, die für die Datentabelle erforderlich ist. Der Standardwert lautet **RECOVERY(NONE)**.

Bei einer benutzerverwalteten Datentabelle wird von CICS nur das dynamische Zurücksetzen von Transaktionen unterstützt, sodass **RECOVERY(BACKOUTONLY)** und **RECOVERY(ALL)** identisch sind.

Bei einer CICS-verwalteten Datentabelle gilt der Parameter RECOVERY für die Quellendatei; sie muss mit jeder anderen Dateidefinition für denselben Datensatz konsistent sein.

Die Wiederherstellungsattribute einer benutzerverwalteten Datentabelle sind unabhängig von allen Wiederherstellungsattributen, die die zugehörige Quellendatei haben kann.

Wenn Sie eine benutzerverwaltete Datentabelle definieren, geben Sie die Wiederherstellungsattribute für die Dateidefinition an, indem Sie entweder **RECOVERY (NONE)** angeben, wenn sie nicht wiederherstellbar sein soll, oder **RECOVERY (BACKOUTONLY | ALL)** angeben, wenn sie nach einem Transaktionsfehler wiederhergestellt werden soll.

Die Quellendatei für die benutzerverwaltete Datentabelle kann nicht wiederherstellbar sein, nur für Backouts (nach Transaktions- und Systemausfällen) wiederherstellbar sein oder vorwärts wiederherstellbar sein, unabhängig von der Angabe, die Sie für die benutzerverwaltete Datentabelle angegeben haben.

Die Quellendatei kann ihre Wiederherstellungsattribute auf eine der beiden folgenden Arten anfordern:

- Indem die Wiederherstellungsattribute für die Datei im ICF-Katalog definiert werden. (Dies ist in CICS Transaction Server for z/OS, Version 5 Release 6 sowohl in RLS- als auch Nicht-RLS-Modusdateien möglich.)
- Indem ein anderer Dateiname verwendet wird, um auf die Datei als gewöhnliche CICS-Datei zuzugreifen, wobei die Wiederherstellungsattribute in der Dateidefinition angegeben werden. (Dies ist in CICS Transaction Server for z/OS, Version 5 Release 6 ausschließlich in Nicht-RLS-Modusdateien möglich.)

### **Beispiel für die Definition einer CICS-verwalteten Datentabelle**

In diesem Beispiel wird die Definition einer CICS-verwalteten Datentabelle dargestellt. Es werden nur die relevanten Parameter angezeigt.

```

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
Nsrgrgroup    ==>
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
JNLSYNCRRead  ==> No          No | Yes
JNLUpdate     ==> No          No | Yes
JNLAdd        ==> None        None | Before | AFter | All
JNLSYNCRWrite ==> Yes         Yes | No
RECOVERY PARAMETERS
RECOVery      ==> All         None | Backoutonly | All
Fwdrecovlog   ==> 10         No | 1-99
Backuptype    ==> STAtic      STAtic | DYNamic
SECURITY
RESsecnum     : 00           0-24 | Public

```

### Beispiel einer benutzerverwalteten Datentabellendefinition

Dieses Beispiel zeigt die Definition einer benutzerverwalteten Datentabelle. Es werden nur die relevanten Parameter angezeigt.

```

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|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
JNLSYNCRd  ==> No      No | Yes
JNLUpdate  ==> No      No | Yes
JNLAdd     ==> None    None | Before | After | All
JNLSYNCRw  ==> Yes     Yes | No
RECOVERY PARAMETERS
RECOVery   ==> Backoutonly None | Backoutonly | All
Fwdrecovlog ==> No      No | 1-99
Backuptype ==> STAtic   STAtic | DYNamic
SECURITY
RESsecnum  : 00      0-24 | Public

```

## EXEC CICS-Befehle für Datentabellen

Sie können den Befehl **EXEC CICS SET FILE** verwenden, um die Definition einer vorhandenen Datei zu ändern, und mit dem Befehl **EXEC CICS INQUIRE FILE** können Sie die Definition einer vorhandenen Datei überprüfen.

Informationen zur Programmierung, einschließlich der Details zur Verwendung dieser Befehle und der hier beschriebenen Parameter finden Sie in [SET FILE](#). Die Parameter, die für Datentabelle relevant sind, werden im Folgenden beschrieben.

*Dieser Abschnitt enthält eine allgemeine Programmierschnittstelle und zugehörige Informationen zur Anleitung.*

### SET FILE

Die folgenden Parameter sind für Datentabellen relevant. Sie können sie nur verwenden, wenn die Datei geschlossen und inaktiviert ist.

Sie können ein Datentabellenattribut einer Datei in einem Datenbereich für CICS-Werte (CVDA) angeben:



**TABLE(cvda)**

Geben Sie einen CVDA-Wert für **CICSTABLE** an, um die Datei als CICS-verwaltete Datentabelle zu definieren.

Geben Sie einen CVDA-Wert für **USERTABLE** an, um die Datei als benutzerverwaltete Datentabelle zu definieren.

Geben Sie einen CVDA-Wert für **NOTTABLE** an, um anzugeben, dass die Datei keine Datentabelle ist.

**Anmerkung:** Sie können auch CFTABLE angeben, um eine Coupling-Facility-Datentabelle anzugeben.

**MAXNUMRECS(value)**

Gibt die maximale Anzahl der Datensätze im Bereich von 1 bis 99999999 an, die in der Datentabelle enthalten sein können. Der Wert null bedeutet, dass es keine maximale Begrenzung gibt.

**INQUIRE FILE**

Die folgenden Parameter sind für Datentabellen relevant.

Sie können anfordern, dass jedes Datentabellenattribut einer Datei in einem Datenbereich für CICS-Werte (CVDA) zurückgegeben wird, indem Sie Folgendes angeben:

**TABLE(cvda)**

Wenn der Wert **CICSTABLE** zurückgegeben wird, wurde die Datei als CICS-verwaltete Datentabelle definiert.

Wenn der Wert **USERTABLE** zurückgegeben wird, wurde die Datei als benutzerverwaltete Datentabelle definiert.

Wenn der Wert **CFTABLE** zurückgegeben wird, wurde die Datei als Coupling-Facility-Datentabelle definiert.

Wenn der Wert **NOTTABLE** zurückgegeben wird, ist die Datei momentan nicht als Datentabelle definiert.

Wenn der Wert **NOTAPPLIC** zurückgegeben wird, ist die Option nicht anwendbar, da die Datei eine ferne Datei ist.

**MAXNUMRECS(cvda)**

Der zurückgegebene Wert gibt die maximale Anzahl der Datensätze an, die in der Datentabelle enthalten sein können. Der Wert null bedeutet, dass es keine maximale Begrenzung gibt.

**CEMT-Befehle für Datentabellen**

Mit dem Befehl CEMT SET FILE können Sie die Definition einer vorhandenen Datei ändern und mit dem Befehl CEMT INQUIRE FILE die Definition einer vorhandenen Datei überprüfen.

Ausführliche Informationen zur Verwendung dieser Befehle, einschließlich der hier beschriebenen Parameter, finden Sie in [INQUIRE FILE](#). Die Parameter, die für Datentabelle relevant sind, werden im Folgenden beschrieben.

**SET FILE**

Die folgenden Parameter sind für Datentabellen relevant. Sie können sie nur verwenden, wenn die Datei geschlossen und inaktiviert ist.

**{CICSTABLE|USERTABLE|CFTABLE|NOTTABLE}**

Geben Sie **CICSTABLE** an, um die Datei als eine CICS-verwaltete Datentabelle zu definieren.

Geben Sie **USERTABLE** an, um die Datei als benutzerverwaltete Datentabelle zu definieren.

**Anmerkung:** Sie können auch CFTABLE angeben, um eine Coupling-Facility-Datentabelle anzugeben.

Geben Sie **NOTTABLE** an, um anzugeben, dass die Datei keine Datentabelle ist.

**MAXNUMRECS(value)**

Geben Sie die maximale Anzahl der Datensätze im Bereich von 1 bis 99999999 an, die in der Datentabelle enthalten sein können. Der Wert null bedeutet, dass es keine maximale Begrenzung gibt.

**INQUIRE FILE**

Die folgenden Parameter sind für Datentabellen relevant.

**Datentabelle**

Wenn der Wert **CICSTABLE** zurückgegeben wird, wurde die Datei als CICS-verwaltete Datentabelle definiert.

Wenn der Wert **USERTABLE** zurückgegeben wird, wurde die Datei als benutzerverwaltete Datentabelle definiert.

Wenn der Wert **CFTABLE** zurückgegeben wird, wurde die Datei als Coupling-Facility-Datentabelle definiert.

Wenn der Wert **NOTTABLE** zurückgegeben wird, ist die Datei momentan nicht als Datentabelle definiert.

**MAXNUMRECS(value)**

Der zurückgegebene Wert gibt die maximale Anzahl der Datensätze an, die in der Datentabelle enthalten sein können. Der Wert null bedeutet, dass es keine maximale Begrenzung gibt.

---

## Kapitel 3. Zugriff auf Datentabellen entwickeln

Sie können auf eine Datentabelle mit den EXEC CICS-Dateisteuerungsbefehlen zugreifen, die Sie auch für eine normale CICS-Datei verwenden. Diese Befehle können vollständig für eine CICS-verwaltete Datentabelle sowie mit einigen Einschränkungen für eine benutzerverwaltete Datentabelle verwendet werden.

Allgemeine Informationen zur Verwendung dieser Befehle finden Sie in [Funktion Distributed Program Link verwenden](#). Informationen zur Programmierung finden Sie in [Zusammenfassung der CICS-Befehle](#).

### Anwendungsprogrammierung für CICS-verwaltete Datentabellen

CICS behandelt eine CICS-verwaltete Datentabelle und die zugehörige Quellendatei als eine einzige Entität. Nachdem die Datentabelle geladen wurde, hält CICS den Inhalt der Datentabelle und die Quellendatei automatisch konsistent. Alle Änderungen, die eine Anwendung an der Datei vornimmt, werden in Tabelle und Quellendatei abgebildet. In fast allen Situationen ist die Verwendung einer Datentabelle für den Anwendungsprogrammierer transparent.

Alle Befehle und Optionen für die Dateisteuerung können für eine CICS-verwaltete Datentabelle verwendet werden. Einige Befehle werden nur durch den Zugriff auf die Datentabelle (mithilfe von speicherübergreifenden Services für gemeinsam genutzte Dateien) ausgeführt. Andere Befehle werden nur durch den Zugriff auf die Quellendatei (mithilfe der Funktionsverlagerung für gemeinsam genutzte Dateien) oder durch Zugriff auf beides ausgeführt.

Die folgenden Befehle greifen in der Regel nur auf die Datentabelle zu:

- READ-Befehle ohne die Optionen UPDATE oder RBA
- Befehle STARTBR, RESETBR, READNEXT und READPREV ohne die RBA-Option
- Befehl ENDBR (es sei denn, die Suchsequenz hat auf die Quellendatei zugegriffen)

Die folgenden Befehle greifen nur auf die Quellendatei zu:

- READ-Befehle mit den Optionen UPDATE oder RBA
- Befehle STARTBR, RESETBR, READNEXT und READPREV mit der RBA-Option
- Befehl ENDBR für eine Suchsequenz, die auf die Quellendatei zugegriffen hat

Die folgenden Befehle greifen möglicherweise sowohl auf die Datentabelle als auch auf die Quellendatei zu:

- READ- und Suchbefehle (die normalerweise nur auf die Datentabelle zugreifen), die eine Lücke in der Schlüsselfolge von Datensätzen in der Datentabelle finden. Diese Lücke kann darauf hinweisen, dass ein oder mehrere Datensätze in der Datentabelle fehlen. Hierfür können folgende Gründe vorliegen:
  - Datensätze wurden durch einen Benutzerexit unterdrückt.
  - Die maximale Anzahl der Datensätze ist erreicht.
  - Es ist nicht genügend virtueller Speicher für die Datentabelle verfügbar.
  - Ein abnormales Ereignis ist aufgetreten.
- Befehle READ, READNEXT und READPREV für Datensätze, die gerade vom Befehl WRITE, REWRITE oder DELETE verarbeitet werden: Diese Befehle müssen zunächst auf die Datentabelle zugreifen, um festzustellen, ob diese Situation vorhanden ist.
- Befehle WRITE, REWRITE und DELETE: Diese Befehle werden immer in der FOR ausgeführt, wo sie zuerst die Quellendatei aktualisieren. Wenn dies erfolgreich ist, wird eine entsprechende Änderung an der Datentabelle unter Verwendung der lokalen gemeinsam genutzten Datentabellenservices in der FOR versucht. Im Falle des Befehls WRITE kann das Hinzufügen des Datensatzes in die Datentabelle vom Benutzerexit XDTAD zurückgewiesen werden oder fehlschlagen, da die Datentabelle voll ist oder nicht ausreichend virtueller Speicher verfügbar ist.

## **Generische Lesevorgänge für eine CICS-verwaltete Datentabelle**

Bei Anwendungen, die generische Lesevorgänge (Befehl Read mit der Option GENERIC) ausführen, gibt es einen Unterschied im Verhalten einer CICS-verwalteten Datentabelle im Vergleich zu einer VSAM-Datei. Möglicherweise müssen Sie diese Anwendungen ändern, wenn Sie eine VSAM-Datei in eine CICS-verwaltete Datentabelle konvertieren.

Wenn CICS beim generischen Lesevorgang einer VSAM-Datei eine NOTFND-Bedingung zurückgibt, weil der Datensatz in der Tabelle nicht gefunden wird, bleiben die Bereiche INTO() und RIDFLD() des Befehls READ unverändert. Wenn CICS beim generischen Lesevorgang einer CICS-verwalteten Datentabelle eine NOTFND-Bedingung zurückgibt, löscht CICS die Bereiche INTO() und RIDFLD(), um sicherzustellen, dass kein falscher Datensatz zurückgegeben wird.

Dieses Verhalten optimiert die Leistung von CICS-verwalteten Datentabellen, es bedeutet jedoch, dass Anwendungen nicht mehr davon abhängig sein können, dass die ursprünglichen Werte in den Bereichen INTO() und RIDFLD() zurückgegeben werden. Wenn Anwendungen vorhanden sind, die generische Lesevorgänge ausführen, ändern Sie sie nach Bedarf, um geeignete Maßnahmen zu ergreifen, wenn eine NOTFND-Bedingung zurückgegeben wird und die Bereiche INTO() und RIDFLD() gelöscht werden.

## **CICS-verwaltete Datentabelle beim Laden verwenden**

Es ist möglich, eine CICS-verwaltete Datentabelle zu verwenden, während sie geladen wird. Wenn der erforderliche Datensatz bereits geladen wurde, wird die Verarbeitung der Anforderung auf die normale Art und Weise gehandhabt.

Wenn der Datensatz noch nicht geladen wurde, wird Folgendes ausgeführt:

- Bei einem READ-Befehl wird der Datensatz aus der Quellendatei gelesen und an das Anwendungsprogramm zurückgegeben. Er wird der Datentabelle hinzugefügt, wenn die normale Ladesequenz sie erreicht.
- Bei einem WRITE-Befehl wird der Datensatz der Quellendatei und der Datentabelle hinzugefügt (wenn sie nicht vom Benutzerexit XDTRD unterdrückt werden).
- Bei einem REWRITE- oder DELETE-Befehl wird die Änderung auf die Quellendatei angewendet. Diese Änderung wird dann durch den normalen Ladevorgang in die Datentabelle übernommen.

## **Anwendungsprogrammierung für eine benutzerverwaltete Datentabelle**

CICS behandelt eine benutzerverwaltete Datentabelle (UMT) und die zugehörige Quellendatei als separate Entitäten. Wenn das Laden abgeschlossen ist, werden alle Dateisteuerungsbefehle, die auf den Dateinamen zugreifen, nur für die Datentabelle ausgeführt.

Wenn eine Anforderung nicht aus einer benutzerverwalteten Datentabelle erfüllt werden kann, greift CICS nicht (wie bei einer CICS-verwalteten Datentabelle) auf die Quellendatei zu. CICS gibt eine Ausnahmebedingung anstelle einer Antwort zurück.

Sie können die Benutzerexits in Datentabellenservices verwenden, um nur die Datensätze einzureihen, auf die Sie in der Datentabelle zugreifen müssen. Es besteht keine Möglichkeit, dass auf die Quellendatei zugegriffen wird, für die sie nicht geladen werden. Sie können auch den Benutzerexit XDTRD verwenden, um jeden Datensatz zu ändern (indem z. B. nur eine Teilmenge seiner Felder ausgewählt wird), wenn er geladen wird.

Datensätze, die sich in der Quellendatei befanden, als die Datentabelle geöffnet wurde, können in der Datentabelle abwesend sein, da sie beim Laden nicht kopiert wurden. Dies kann an der Unterdrückung durch den Benutzerexit XDTRD oder an einem abnormalen Ereignis liegen, z. B. wenn die Datentabelle voll wird.

Einige Anwendungsprogrammierungsanforderungen werden für eine benutzerverwaltete Datentabelle nicht unterstützt. Zum Beispiel werden Leseanforderungen, die die Option UPDATE mit einem ungenauen Schlüssel verwenden, nicht unterstützt. Außerdem sind einige Ausnahmebedingungen einzigartig für benutzerverwaltete Datentabellen. Möglicherweise müssen Sie vorhandene Anwendungen ändern, um den Einschränkungen bei der Verwendung von Befehlen und Optionen zu entsprechen oder um die Ausnahmebedingungen zu behandeln, die CICS zurückgibt.

Die folgenden Befehle werden nicht unterstützt. Sie geben die Bedingung INVREQ und den Wert 44 im Feld EIBRESP2 zurück:

- Befehle mit der Option RBA
- WRITE-Befehle mit der Option MASSINSERT

Die folgenden Befehle werden unterstützt (über speicherübergreifende Services für den Fernzugriff):

- READ-Befehle ohne die Option RBA oder ohne die Option UPDATE. Wenn der Datensatz in der Datentabelle nicht vorhanden ist, wird die Bedingung NOTFND zurückgegeben.
- Befehle STARTBR, RESETBR, READNEXT und READPREV ohne die RBA-Option.
- ENDBR-Befehle.

Die folgenden Befehle werden unterstützt (mit Funktionsverlagerung für Fernanforderungen):

- WRITE-Befehle ohne die Optionen RBA oder MASSINSERT. Der Datensatz wird der Datentabelle hinzugefügt (wenn er nicht durch den Benutzerexit XDTAD unterdrückt wird).

Die Bedingung NOSPACE wird in den folgenden Situationen zurückgegeben:

- Es ist nicht genügend Speicher im Datenspeicherbereich vorhanden, um den Datensatz zur Datentabelle hinzuzufügen.
- Die Datentabelle enthält bereits die maximale Anzahl der Datensätze, die in der Dateidefinition angegeben ist.

CICS gibt die Nachricht DFHFC0432 aus, wenn ein Schreibvorgang versucht wird, der verfügbare Speicherplatz jedoch nicht ausreicht.

Die Bedingung SUPPRESSED wird zurückgegeben, wenn der Benutzerexit XDTAD das Hinzufügen des Datensatzes in die Datentabelle unterdrückt.

- REWRITE-Befehle ohne die Option RBA. Der Datensatz wird in der Datentabelle aktualisiert. Die Bedingung NOSPACE wird zurückgegeben, wenn für den aktualisierten Datensatz nicht genügend virtueller Speicher vorhanden ist. CICS gibt die Nachricht DFHFC0432 aus, wenn nicht genügend Speicher verfügbar ist.
- DELETE-Befehle ohne die Option RBA. Der Datensatz wird aus der Datentabelle gelöscht. Die Bedingung NOTFND wird zurückgegeben, wenn der Datensatz in der Datentabelle nicht vorhanden ist. Die Bedingung NOSPACE wird zurückgegeben, wenn die Datentabelle wiederherstellbar ist und ausreichend virtueller Speicher für die Informationen vorhanden ist, die CICS für den gelöschten Datensatz schreibt.

## **Benutzerverwaltete Datentabelle beim Laden verwenden**

Auf eine benutzerverwaltete Datentabelle kann nur von der File-Ownning Region (FOR) während des Ladens zugegriffen werden. Alle Fernanforderungen sind für die FOR funktionsverlagert, die sie in derselben Weise wie lokale Anforderungen verarbeitet.

Während eine benutzerverwaltete Datentabelle (UMT) geladen wird, können Sie nur Leseanforderungen ohne Aktualisierung mit präzisen Schlüsseln verwenden. Wenn der Datensatz bereits geladen wurde, wird die Verarbeitung der Anforderung auf die normale Art und Weise gehandhabt. Wenn der Datensatz noch nicht geladen wurde, wird der Datensatz aus der Quellendatei gelesen und an den Benutzerexit XDTRD übergeben (falls aktiviert):

- Wenn er nicht durch XDTRD unterdrückt wird, wird der Datensatz zur Datentabelle hinzugefügt und an das Anwendungsprogramm zurückgegeben.
- Wird sie durch XDTRD unterdrückt, wird die Bedingung NOTFND zurückgegeben.

Die LOADING-Bedingung wird für andere Anforderungen zurückgegeben, die gültig gewesen wären, wenn das Laden abgeschlossen worden wäre.

## Speicherübergreifende Services für gemeinsam genutzte Datentabellen verwenden

Speicherübergreifende Services werden verwendet, um einen Anwendungsprogrammierbefehl zu erfüllen, wenn alle hier aufgeführten Bedingungen erfüllt sind.

- CICS muss die SYSID des Zielsystems aus der Ressourcendefinition der Datei in der AOR (Application-Owning Region) abrufen. Diese Bedingung wird erfüllt, wenn der Anwendungsprogrammierbefehl entweder keine explizite SYSID angibt oder eine SYSID angibt, die mit der AOR selbst identisch ist. Die in der Ressourcendefinition der Datei angegebene SYSID entspricht der ID der FOR (File-Owning Region).

Innerhalb einer einzigen Suchsequenz darf eine Anwendung nicht zwischen der Angabe einer expliziten SYSID und der Nicht-Angabe einer SYSID wechseln, da dies zu unvorhersehbaren Ergebnissen führen kann.

- Das Serving-System hat sich angemeldet, das heißt, es hat sich selbst als Eigner der gemeinsam genutzten Datentabelle registriert.
- Das anfordernde System hat die Verbindung zum Server für die Dateien hergestellt, die im Anwendungsprogrammierbefehl angegeben sind.
- Die Datei unterstützt die angeforderte Funktion.

**Anmerkung:** Die Funktionsverlagerung einer Anforderung kann zu einer "Verbindung in Serie" (Daisy Chain) führen. Das heißt, die Anforderung dazu führen, dass ein oder mehrere CICS-Zwischenknoten zwischen der Region, die die Anforderung ausgibt (einer AOR), und der Region, der die Ressource (die FOR) gehört, durchlaufen werden. In darartigen Fällen beschränkt sich die Verwendung von speicherübergreifenden Services auf den Endlink (vom letzten temporären System bis zur FOR).

### Verbindung herstellen

Befehle können erst dann speicherübergreifende Services verwenden, wenn die Verbindung zwischen der AOR (Application-Owning Region) und der fernen Datentabelle hergestellt wurde.

Wenn vor dem Herstellen einer Verbindung eine Suchsequenz gestartet wird, verwenden alle nachfolgenden Anforderungen in der Sequenz Funktionsverlagerungsservices. Dies ist wahrscheinlich der Fall, wenn die Verbindung mit dem Befehl STARTBR nicht hergestellt werden kann, weil die Datentabelle nicht geöffnet ist und der Befehl dazu führt, dass die Datentabelle implizit geöffnet wird. Die Verbindung wird dann bei der nächsten neuen Anforderung an die Datentabelle hergestellt, aber die ursprüngliche Suchsequenz verwendet weiterhin Funktionsverlagerungsservices.

### Verbindungsabbau

Wenn eine Verbindung hergestellt wurde, bleibt sie in Kraft, bis entweder die AOR (Application-Owning Region) ihre ferne Dateidefinition löscht oder die FOR (File-Owning Region) die Datei schließt oder inaktiviert.

Die Effekte der Schließ- oder Inaktivierungsvorgänge sind wie folgt:

- Wenn die FOR die Datei (mit oder ohne die Option FORCE) schließt, wird der Verbindungsabbau bei der nächsten Nicht-Aktualisierungsanforderung geplant, die für die Datei ausgegeben wird (d. h. die nächste Anforderung, um speicherübergreifende Services für den Zugriff auf die Daten zu verwenden).

Der Verbindungsabbau findet statt, sobald alle ausstehenden Suchsequenzen (falls vorhanden) für die Datei beendet wurden. Jede Suchsequenz wird entweder bei der nächsten Suchanforderung beendet (und die Transaktion wird mit dem Code AFCH abnormal beendet, es sei denn, die Anforderung ist der Befehl ENDBR) oder bei der Beendigung der Transaktion beendet.

Nachdem der Verbindungsabbau geplant wurde, sind alle Anforderungen (mit Ausnahme der ausstehenden Suchanforderungen, wie oben beschrieben) so lange funktionsverlagert, bis wieder eine Verbindung hergestellt wird.

- Wenn die FOR die Datei ohne die Option FORCE inaktiviert, wird der Verbindungsabbau beim nächsten Nicht-Aktualisierungsbefehl READ oder STARTBR, der für die Datei ausgegeben wird, geplant, es sei denn, die FOR aktiviert die Datei vorher erneut.

Falls der Verbindungsabbau geplant wird, findet der Verbindungsabbau statt, sobald alle ausstehenden Suchsequenzen (falls vorhanden) für die Datei beendet wurden. Solche Suchsequenzen werden normal fortgesetzt; sie werden durch die Inaktivierung nicht beeinflusst, es sei denn, ein Suchvorgang in der Quellendatei wird in der FOR gestartet, um eine Anforderung in der Suchsequenz zu erfüllen.

- Wenn die Option FOR die Datei mit der Option FORCE inaktiviert, ist der Effekt derselbe wie beim Schließen einer Datei. Dabei gilt diese Ausnahme: Wenn die FOR die Datei erneut aktiviert, bevor die AOR die nächste Nicht-Aktualisierungsanforderung für die Datei ausgibt, wird das Inaktivieren nicht von der AOR überwacht und der Verbindungsabbau wird nicht geplant.

## **Unterschiede zwischen Funktionsverlagerung und speicherübergreifenden Services**

Je nachdem, ob eine Funktionsverlagerung oder speicherübergreifende Services für den Zugriff auf die Datentabelle verwendet werden, gibt es eine Reihe von Unterschieden zwischen der Art und Weise, wie Anforderungen verarbeitet werden.

### **Datentabellen schließen**

Wenn die Funktionsverlagerung für eine Suchsequenz einer fernen Datei verwendet wird, kann die Datei erst nach dem Ende der Suchsequenz geschlossen werden (es sei denn, es wird die Option FORCE verwendet).

Wenn speicherübergreifende Services verwendet werden, ist es möglich, dass die Datei während der Suchsequenz geschlossen wird. In diesem Fall wird die Transaktion bei der nächsten Anforderung für diese Datei mit dem Abbruchcode AFCH beendet. Wenn Ihre Anwendungen oder Betriebsprozesse sich auf das Versetzen in den Wartemodus Suchaktivitäten verlassen, wenn entweder eine Datei geschlossen oder eine File-Ownning Region (FOR) normal heruntergefahren wird, sollten Sie sie überprüfen, bevor Sie eine gemeinsam genutzte Datentabelle für die Datei verwenden.

### **Datentabelle inaktivieren**

Wenn die Funktionsverlagerung für eine Suchsequenz einer fernen Datei verwendet wird, kann die Suchsequenz, sobald sie gestartet wurde, normal fortgesetzt werden, selbst wenn die Datei dann inaktiviert wird (es sei denn, die Option FORCE wird verwendet).

Wenn speicherübergreifende Services verwendet werden, ist der Effekt derselbe, es sei denn, es ist während der Suchsequenz erforderlich, die Funktionsverlagerung des Befehls STARTER an die FOR zu verwenden. Dies kann z. B. der Fall sein, wenn eine Lücke in einer CICS-verwalteten Datentabelle es erforderlich macht, dass die VSAM-Quellendatei zum Abrufen von Datensätzen durchsucht wird. Der funktionsverlagerte Befehl STARTBR schlägt fehl, wenn die Datei dann durch eine Anforderung inaktiviert wird, die von der FOR nach dem Start der Suchsequenz, die in der Application-Ownning Region (AOR) gestartet wurde, ausgegeben wurde. In diesem Fall kann die Suchsequenz nicht normal fortgesetzt werden, sodass die Transaktion in der AOR mit dem Code AFCH abnormal beendet wird.

Wenn die Option FORCE zusammen mit der Inaktivierungsanforderung verwendet wird, werden immer alle funktionsverlagerten Suchanforderungen beendet. Wenn die Datei erneut aktiviert wird, ist es möglich, dass Suchanforderungen, die speicherübergreifende Services verwenden, nicht betroffen sind. (Informationen zu FORCE finden Sie in „Verbindungsabbau“ auf Seite 32).

### **Benutzerexits**

Bei funktionsverlagerten Anforderungen werden die Benutzerexits XEIIN und XEIOUT für die EXEC-Schnittstelle sowie die Benutzerexits XFCREQ und XFCREQC für die Dateisteuerung sowohl in der AOR als auch in der FOR aufgerufen.

Bei speicherübergreifenden Anforderungen werden diese Benutzerexits nur in der AOR aufgerufen.

### **Sicherheitsprüfung**

Bei funktionsverlagerten Anforderungen wird die Sicherheitsprüfung in der FOR für die erste Anforderung aufgerufen, die sich auf eine bestimmte Datei in jeder Arbeitseinheit (UOW) bezieht. Deshalb können Sicherheitsprüfungen auf Transaktionsebene in der FOR ausgeführt werden.

Bei speicherübergreifenden Anforderungen wird die Sicherheitsprüfung nur beim Zeitpunkt der CONNECT-Operation aufgerufen. Deshalb können Sicherheitsprüfungen auf Transaktionsebene nicht in der FOR ausgeführt werden.

### **Fehler bei Leseanforderung**

Wenn eine Leseanforderung mit einer Funktionsverlagerung fehlschlägt, bleibt der Eingabebereich unverändert.

Wenn eine Leseanforderung unter Verwendung von speicherübergreifenden Services fehlschlägt, besteht die Möglichkeit, dass der Eingabebereich geändert wird, obwohl kein Datensatz abgerufen wurde. Sie sollten sich daher nicht darauf verlassen, dass der Eingabebereich unverändert bleibt, auch wenn Sie sicher sein können, dass der Schlüssel nicht geändert wurde.

### **EXEC-Schnittstellenblock**

Möglicherweise stellen Sie fest, dass Leseanforderungen, die speicherübergreifende Services verwenden, einen Wert im Feld EIBRESP2 zurückgeben. Funktionsverlagerte Anforderungen tun dies jedoch nicht, sodass Ihre Anwendungen nicht von diesem Feld abhängig sein sollten, das durch Leseanforderungen festgelegt wird.

### **Schlüssellänge**

Bei funktionsverlagerten Anforderungen müssen Sie die korrekte Schlüssellänge entweder in der Definition der fernen Datei in der AOR oder explizit in der Dateianforderung angeben (damit die Schlüssellänge in der VSAM-Definition in der FOR übereinstimmt). Wenn Sie dies nicht tun, wird die Bedingung INVREQ für jede Anforderung zurückgegeben, die auf die Datei zugreift. Dies gilt für jede Datei, nicht nur für die Datei, die als Datentabelle definiert ist.

Bei speicherübergreifenden Anforderungen wird die Schlüssellänge in der AOR nicht verwendet; Anforderungen können auch dann erfolgreich ausgeführt werden, wenn die Schlüssellänge nicht in der AOR angegeben ist, oder wenn die in der AOR angegebene Schlüssellänge nicht mit der in der FOR angegebenen übereinstimmt. Ihre Anwendungen sollten aber nicht davon abhängig sein, da einige der Anforderungen möglicherweise funktionsverlagert sind.

## **Unterschiede zwischen gemeinsam genutzten Datentabellenservices und VSAM**

Da gemeinsam genutzte Datentabellenservices VSAM für viele Datentabellenanforderungen ersetzen, gibt es Unterschiede in der Art und Weise, in der bestimmte Anforderungen implementiert werden.

### **Lesevorgang während der Aktualisierung (mit mehreren Transaktionen)**

Bei einem READ-Befehl für einen Datentabellendatensatz nach einer READ UPDATE-Operation für diesen Datensatz durch eine *andere* Transaktion und vor der zugeordneten Aktualisierungsanforderung, wird der READ-Befehl sofort verarbeitet, wenn gemeinsam genutzte Datentabellenservices verwendet werden.

Wenn VSAM verwendet wird, wartet der Befehl READ, bis die Aktualisierungsanforderung abgeschlossen ist.

### **Lesevorgang während der Aktualisierung (mit derselben Transaktion)**

Bei einem READ-Befehl für einen Datentabellendatensatz nach einer READ UPDATE-Operation für diesen Datensatz durch *dieselbe* Transaktion und vor der zugeordneten Aktualisierungsanforderung, wird der READ-Befehl sofort verarbeitet, wenn gemeinsam genutzte Datentabellenservices verwendet werden.

Wenn VSAM verwendet wird, führt die Transaktion zum Deadlock-Abbruchcode AFCG.

### **Löschvorgang während des Suchens**

Wenn gemeinsam genutzte Datentabellenservices für den Befehl STARTBR oder RESETBR für einen Datentabellendatensatz verwendet werden, ist es möglich, dass der Datensatz gelöscht wird, bevor der zugehörige Befehl READNEXT oder READPREV ausgegeben wird. Wenn VSAM verwendet wird,



kann der Datensatz nicht gelöscht werden, bevor der zugehörige Befehl READNEXT oder READPREV ausgegeben wurde.

Wenn also gemeinsam genutzte Datentabellenservices verwendet werden, wird der Befehl STARTBR oder RESETBR mit einem anderen Schlüssel als dem speziellen Schlüssel "Letzter Datensatz" (X'FF...') ausgegeben, und wenn der ausgewählte Datensatz vor dem Befehl READNEXT gelöscht wird, liest der Befehl READNEXT den nachfolgenden Datensatz.

Wenn es keinen nachfolgenden Datensatz gibt, wird die ENDFILE-Bedingung zurückgegeben. Wenn die Option EQUAL in STARTBR oder RESETBR verwendet wurde, stimmt der Schlüssel des gelesenen Datensatzes möglicherweise nicht mit dem angegebenen Schlüssel überein.

Wenn der Befehl STARTBR oder RESETBR mit dem speziellen Schlüssel "Letzter Datensatz" ausgegeben und der ausgewählte Datensatz vor dem Befehl READPREV gelöscht wird, liest der Befehl READPREV den vorhergehenden Datensatz oder gibt eine ENDFILE Bedingung zurück, wenn es einen solchen Datensatz nicht gibt.

### **Schreibvorgang während des Suchens**

Wenn gemeinsam genutzte Datentabellenservices verwendet werden und Lesevorgänge bis zum Ende einer Datei gelesen werden, wodurch die ENDFILE-Bedingung ausgelöst wird und dann ein neuer Datensatz hinter das Ende der Datei eingefügt wird, kann ein nachfolgender READNEXT den neuen Datensatz lesen.

Wenn VSAM verwendet wird, kann ein nachfolgender READNEXT den neuen Datensatz möglicherweise nicht finden, sondern meldet erneut die Bedingung ENDFILE.

### **Löschvorgang während der Aktualisierung (mit derselben Transaktion)**

Wenn gemeinsam genutzte Datentabellenservices für einen DELETE-Befehl verwendet werden, der einen RIDFLD für einen Datentabellendatensatz angibt, nachdem für diesen Datensatz ein READ UPDATE von derselben Transaktion und vor der zugehörigen Aktualisierungsanforderung ausgegeben wird, wird der DELETE-Befehl erfolgreich verarbeitet und die zugehörige Aktualisierungsanforderung empfängt eine NOTFND-Bedingung.



---

## Kapitel 4. Datentabellen mithilfe von Benutzerexits anpassen

In Datentabellenservices sind drei globale Benutzerexitpunkte eingebunden. Sie können ein oder mehrere Assemblersprachprogramme bereitstellen, die an jedem dieser Punkte ausgeführt werden, um die von CICS bereitgestellte Funktion zu erweitern oder zu ändern.

**Anmerkung:** Dieser Abschnitt enthält eine produktabhängige Programmierschnittstelle und zugehörige Informationen.

CICS stellt Beispielbenutzerexitprogramme in der Bibliothek SDFHSAMP für die globalen XDTRD-, XDTAD- und XD TLC-Benutzerexits zur Verfügung. Diese Programme, die in dieser Dokumentation wiedergegeben werden, beschreiben mit Beispielen für Codes und Datendefinitionssequenzen die Konventionen, die in Benutzerexitprogrammen verwendet werden, die mit gemeinsam genutzten Datentabellen verwendet werden. Diese Beispiele sind nur als allgemeine Anleitung gedacht und definieren keine Programmierschnittstelle.

Programmierinformationen zu globalen Benutzerexits und deren Verwendung finden Sie in [Globale Benutzerexitprogramme](#).

**Anmerkung:** Die Benutzerexits XEIIIN und XEIIOUT der EXEC-Schnittstelle sowie die Benutzerexits XFCREQ und XFCREQC der Dateisteuerung werden nicht in der File-Ownning Region (FOR) gestartet, wenn eine Anforderung zum Zugriff auf eine Datentabelle durch speicherübergreifende Services gegeben ist.

### Kommunikation zwischen CICS und Exitprogrammen für gemeinsam genutzte Datentabellen

Mithilfe einer Parameterliste werden Informationen zwischen CICS und den Exitprogrammen für Datentabellen übergeben.

In der Bibliothek CICSTS56.CICS.SDFHMAC stellt CICS ein Copybook mit dem Namen DFHXDTDS zur Verfügung, das einen DSECT zum Definieren dieser Parameterliste enthält. Beziehen Sie die Anweisung COPY DFHXDTDS in jedes der Exitprogramme ein. Der DSECT wird in [Abbildung 6 auf Seite 38](#) angezeigt.

Die in diesem DSECT verwendeten Feldnamen werden in den Benutzerexitbeschreibungen referenziert, die der Abbildung folgen.

```

*****
*
*   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 UEPDTSOT 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 UEPDTSOT 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)
UEPDFTLG           DS  0CL1          Flags (In):

```

Abbildung 6. Parameterliste der Datentabellen-Benutzerexits

```

*-----*
*       The UEPDTSMT 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 UEPDTCPT 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).
*-----*

```

UEPDTSMT	DS BL1	Exit invoked by SDT support	
UEPDTCMT	EQU X'80'	Table is CICS-maintained	
UEPDTCPT	EQU X'40'	Exit invoked by table loader,	X
	EQU X'20'	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
UEPDTCMT	EQU X'08'	Table is user-maintained	
*	EQU X'07'	Reserved	

Abbildung 7. Parameterliste der Datentabellen-Benutzerexits (Fortsetzung)

```

*-----*
*       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.    *
*-----*

```

UEPDTCRC	DS AL1	Data table load return code -	X
		XDTRD only, values below (In)	
UEPDTRA	DS BL2	Reserved	
	DS A	Data record address - XDTRD	X
UEPDTRBL	DS F	and XDTRD only (In)	
		Data buffer length - XDTRD and	X
UEPDTRL	DS F	XDTRD only (In)	
		Data table record length -	X
		XDTRD and XDTRD 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 XDTRD	X
		only (In)	
UEPDTKL	DS F	Key length - XDTRD and XDTRD	X
		only (In)	

Abbildung 8. Parameterliste der Datentabellen-Benutzerexits (Fortsetzung)

```

*-----*
*      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 XDTRC exit only)      *
*-----*
UEPDTLCS      EQU 0      load completed successfully
UEPDTLFL      EQU 128    load failed

```

Abbildung 9. Parameterliste der Datentabellen-Benutzerexits (Fortsetzung)

Die Benutzerexits sollten einen Rückgabecode in Register 15 setzen. Die Rückgabecodewerte werden vom Makro DFHUEXIT bereitgestellt. Die gültigen Werte für jeden Benutzerexit sind in den folgenden Beschreibungen angegeben.

Wenn die Exitprogramme weiterhin sowohl für Basis-Datentabellen als auch für gemeinsam genutzte Datentabellen verwendet werden sollen, können Sie UEPDTFLG überprüfen, um herauszufinden, welche Version der Datentabellenunterstützung das Exitprogramm aufgerufen hat. Bei gemeinsam genutzten Datentabellen gibt dieses Markierungsbyte auch an, welcher Typ der Datentabelle verwendet wird und ob das Exitprogramm während des Ladens aufgerufen wird.

Das Exitprogramm sollte entweder den Dateinamen (Feld UEPDTNAM) oder den Namen der Quellendatei (siehe Felder UEPDTDSN und UEPDTDSL) verwenden, um zu bestimmen, ob für diese Datei eine Aktion ausgeführt werden soll.

Sie können mehrere Exitprogramme an demselben Exitpunkt aktivieren, von denen jede z. B. die Aktion für eine bestimmte Datei oder einen bestimmten Datensatz ergreift.

## Benutzerexit XDTRD

Der Benutzerexit XDTRD wird aufgerufen, bevor CICS versucht, der Datentabelle einen Datensatz hinzuzufügen, der aus der Quellendatei abgerufen wird. Sie können auswählen, ob der Datensatz in die Datentabelle geladen werden soll. Bei einer benutzerverwalteten Datentabelle können Sie den Datensatz auch ändern.

XDTRD wird normalerweise aufgerufen, wenn der Ladeprozess während des sequenziellen Kopierens der Quellendatei einen Datensatz abruft. Es kann jedoch auch aufgerufen werden, wenn eine Anwendung einen Datensatz abruft, der sich nicht in der Datentabelle befindet, und wenn eine der folgenden Bedingungen zutrifft:

- Bei einer benutzerverwalteten Datentabelle befindet sich das Laden noch in Bearbeitung.
- Bei einer CICS-verwalteten Datentabelle wurde das Laden beendet, bevor das Ende der Quellendatei erreicht wurde (z. B. weil die Datentabelle vollständig belegt war).

Der aus der Quellendatei abgerufene Datensatz wird als Parameter an das Benutzerexitprogramm übergeben (siehe Felder UEPDTTRA und UEPDTRL). Dieses Programm kann (z. B. abhängig vom Schlüsselwert; siehe Felder UEPDTKA und UEPDTKL) auswählen, ob der Datensatz in die Datentabelle eingeschlossen werden soll.

Alternativ kann das Exitprogramm anfordern, dass alle nachfolgenden Datensätze bis zu einem angegebenen Schlüssel übersprungen werden; siehe Feld UEPDTSKA. Diese Datensätze werden nicht an das Exitprogramm übergeben. Diese Funktion ist nur während des Ladens verfügbar. Sie können den Schlüssel als vollständigen Schlüssel angeben oder Sie können nur die führenden Zeichen angeben, indem Sie den zu überspringenden Schlüsselbereich mit binären Nullen auffüllen.

Die erforderliche Aktion wird durch Festlegen des Rückgabecodes angegeben. Je nach Rückgabecodewert führt CICS die folgende Aktion durch:

<i>Tabelle 3. Rückgabecodes für den Benutzerexit XDTRD. Der Wert UERCPURG sollte zurückgegeben werden, wenn das Exitprogramm eine PURGED-Antwort auf einen von ihm abgesetzten Aufruf empfangen hat.</i>	
<b>Rückgabecode</b>	<b>Aktion</b>
UERCDTAC	Schließen Sie den Datensatz in der Datentabelle ein. Dies ist der Standard, wenn der Exit nicht aktiviert ist.
UERCDTRJ	Schließen Sie den Datensatz in der Datentabelle nicht ein.
UERCDTOP	Überspringen Sie diesen Datensatz und die folgenden Datensätze, bis ein Schlüssel gefunden wird, der größer als oder gleich dem Schlüssel ist, der im überspringenden Schlüsselbereich angegeben wurde.

Bei einer benutzerverwalteten Datentabelle kann das Programm auch die Daten im Datensatz ändern, um den für die Datentabelle erforderlichen Speicher zu reduzieren. Anwendungsprogramme, die die Datentabelle verwenden, müssen alle Änderungen beachten, die vom Exitprogramm an dem Datensatzformat vorgenommen wurden. Wenn die Datensatzlänge geändert wird, muss das Exitprogramm die neue Länge in der Parameterliste festlegen (siehe Feld UEPDTRL). Die neue Länge darf die Datenpufferlänge nicht überschreiten (siehe Feld UEPDTRBL).

#### **Beispiel für ein XDTRD-Exitprogramm: DFH\$DTRD**

DFH\$DTRD ist ein Beispiel für ein globales XDTRD-Benutzerexitprogramm. Es veranschaulicht die Verwendung des Benutzerexits XDTRD für gemeinsam genutzte Datentabellen. Das Beispielprogramm wird in der Bibliothek SDFHSAMP bereitgestellt.

## **Benutzerexit XDTAD**

XDTAD wird für jeden Datensatz aufgerufen, der nach dem ersten Laden der Quellendatei hinzugefügt wird. Sie können auswählen, ob der Datensatz zur Datentabelle hinzugefügt werden soll. Dieser Benutzerexit kann die Datensätze nicht ändern, da davon ausgegangen wird (und da die Datensätze von der Anwendung geschrieben werden), dass sie sich bereits in dem Format befinden, das in der Datentabelle verwendet wird.

Der Benutzerexit XDTAD wird aufgerufen, wenn eine Schreibanforderung an eine Datentabelle abgesetzt wird.

- Bei einer benutzerverwalteten Datentabelle wird der Benutzerexit ein Mal aufgerufen, bevor der Datensatz zur Datentabelle hinzugefügt wird.
- Bei einer CICS-verwalteten Datentabelle wird der Benutzerexit zweimal aufgerufen: bevor der Datensatz zur Quellendatei hinzugefügt wird und dann erneut, bevor der Datensatz zur Datentabelle hinzugefügt wird.

**Anmerkung:** Bei Coupling-Facility-Datentabellen kann der Exit für einen offenen TCB (Tasksteuerblock) aufgerufen werden. Stellen Sie daher sicher, dass der Exit threadsicher ist und als für CICS threadsicher aktiviert ist, um ein übermäßiges TCB-Switching zu vermeiden.

Der von der Anwendung geschriebene Datensatz wird als Parameter an das Benutzerexitprogramm übergeben (siehe Felder UEPDTRA und UEPDTRL). Dieses Programm kann (z. B. abhängig vom Schlüsselwert; siehe Felder UEPDTKA und UEPDTKL) auswählen, ob der Datensatz in die Datentabelle eingeschlossen werden soll. Diese Entscheidung wird durch das Festlegen des Rückgabecodes angegeben.

Je nach Rückgabecodewert führt CICS die folgende Aktion durch:

*Tabelle 4. Rückgabecodes für den Benutzerexit XDTAD. Der Wert UERCPURG sollte zurückgegeben werden, wenn das Exitprogramm eine PURGED-Antwort auf einen von ihm abgesetzten Aufruf empfangen hat.*

Rückgabecode	Aktion
UERCDTAC	Fügen Sie der Datentabelle den Datensatz hinzu. Dies ist der Standard, wenn der Exit nicht aktiviert ist.
UERCDTRJ	Fügen Sie der Datentabelle den Datensatz nicht hinzu.

Der Exit XDTAD darf die Daten im Datensatz nicht ändern. Wenn Sie die Datensätze mithilfe von XDTRD gekürzt haben, als die benutzerverwaltete Datentabelle geladen wurde, müssen Sie Ihre Anwendung so codieren, dass nur versucht wird, Datensätze im richtigen Format für die Datentabelle zu schreiben.

#### Beispiel für ein XDTAD-Exitprogramm: DFH\$DTAD

DFH\$DTAD ist ein Beispiel für ein globales XDTRD-Benutzerexitprogramm. Es veranschaulicht die Verwendung des Benutzerexits XDTAD für gemeinsam genutzte Datentabellen. Das Beispielprogramm wird in der Bibliothek SDFHSAMP bereitgestellt.

## Benutzerexit XDTLC

Der Benutzerexit XDTLC wird aufgerufen, wenn das Laden der Datentabelle abgeschlossen ist, unabhängig davon, ob das Laden erfolgreich ist oder nicht. Der Benutzerexit wird nicht aufgerufen, wenn die Datentabelle aus irgendeinem Grund geschlossen wird, bevor das Laden abgeschlossen ist.

Das Exitprogramm wird informiert, wenn das Laden nicht erfolgreich abgeschlossen wurde; siehe Feld UEPDTCORC. Dies kann z. B. der Fall sein, wenn die maximale Anzahl der Datensätze erreicht wurde oder wenn der virtuelle Speicher nicht ausreicht. In diesem Fall kann das Exitprogramm anfordern, dass die Datei sofort geschlossen wird, indem der Rückgabecode festgelegt wird.

Je nach Rückgabecodewert führt CICS die folgende Aktion durch:

*Tabelle 5. Rückgabecodes für den XDTLC-Benutzerexit. Der Wert UERCPURG sollte zurückgegeben werden, wenn das Exitprogramm eine PURGED-Antwort auf einen von ihm abgesetzten Aufruf empfangen hat.*

Rückgabecode	Aktion
UERCDTOK	Keine Aktion; die Datei bleibt geöffnet. Dies ist der Standard, wenn der Exit nicht aktiviert ist.
UERCDTCL	Schließen Sie die Datei.

#### Beispiel für XDTLC-Exitprogramm: DFH\$DTLC

DFH\$DTLC ist ein Beispiel für ein globales XDTLC-Benutzerexitprogramm. Es veranschaulicht die Verwendung des Benutzerexits XDTLC für gemeinsam genutzte Datentabellen. Das Beispielprogramm wird in der Bibliothek SDFHSAMP bereitgestellt.

## Benutzerexits für Datentabellen aktivieren

Führen Sie die folgenden Schritte aus, um die Datentabellen-Benutzerexits zu aktivieren.

#### Vorgehensweise

1. Entscheiden Sie, welche Benutzerexits Sie verwenden möchten.

Die Beschreibung der einzelnen Benutzerexits finden Sie in [Kapitel 4, „Datentabellen mithilfe von Benutzerexits anpassen“](#), auf Seite 37.

2. Schreiben Sie die Benutzerexitprogramme.



Beispiele sind in Kapitel 4, „Datentabellen mithilfe von Benutzerexits anpassen“, auf Seite 37 enthalten.

3. Definieren Sie die Benutzerexitprogramme für CICS, indem Sie den Befehl `CEDA DEFINE PROGRAM` verwenden, wie in PROGRAM-Ressourcen beschrieben.

4. Aktivieren Sie die Benutzerexits, indem Sie den Befehl **EXEC CICS ENABLE** verwenden.

Sie können, falls erforderlich, die Benutzerexits später mit dem Befehl **EXEC CICS DISABLE** inaktivieren.

Sofern Sie das Öffnen einer Datentabelle nicht explizit mit einem CEMT- oder EXEC-CICS-Befehl steuern, sollten Sie die Benutzerexits besser während des CICS-Starts aktivieren. Andernfalls beginnt möglicherweise das Laden der Datentabelle, bevor die Benutzerexits aktiviert sind. Gehen Sie wie folgt vor, um die Benutzerexits beim Start zu aktivieren:

5. Schreiben Sie eines oder mehrere PLTPI-Programme (Program List Table Post-Initialization), die den Befehl **EXEC CICS ENABLE** enthalten, um die Benutzerexits zu aktivieren.  
Programmierungsinformationen zu PLTPI-Programmen finden Sie in Initialisierungs- und Beendigungsprogramme schreiben.
6. Definieren Sie eine Programmlistentabelle (PLT; Program List Table) mit einem Eintrag für jedes dieser PLTPI-Programme, wie in Programmlistentabelle (PLT) beschrieben.
7. Geben Sie den Parameter **PLTPI=suffix** für die Systeminitialisierung an, wie im Abschnitt Systeminitialisierungsparameter PLTPI beschrieben. Verwenden Sie das Suffix der PLT, das im vorherigen Schritt definiert wurde. Dies bewirkt, dass die PLTPI-Programme in der zweiten Phase der Initialisierung ausgeführt werden, bevor Dateien geöffnet werden.

### Nächste Schritte

Sie können PLT Shutdown (PLTSD)-Programme in ähnlicher Weise verwenden, um die Benutzerexits während der CICS-Beendigung zu inaktivieren.



---

# Kapitel 5. Datentabellen verwalten

Informationen zu den operativen Aspekten von Datentabellen.

## Datentabellen öffnen

Eine Datentabelle muss geöffnet werden, bevor sie von einer Anwendung verwendet werden kann.

Sie öffnen eine Datentabelle auf die gleiche Weise wie eine beliebige CICS-Datei mit einer der folgenden Methoden:

- Wird automatisch von der von CICS bereitgestellten Transaktion CSFU am Ende des CICS-Starts ausgeführt, wenn die Datentabelle mit OPENTIME(STARTUP) definiert ist.
- Explizit durch eine CEMT- oder EXEC CICS-Anforderung, die vom Benutzer ausgegeben wurde.
- Beim ersten Verweis auf die Datentabelle implizit, wenn die Datentabelle mit OPENTIME(FIRSTREF) definiert ist. Der erste Fernzugriff auf eine geschlossene Datentabelle öffnet diese implizit.

Alle Regeln und Optionen zum Öffnen einer CICS-Datei gelten auch für eine Datei, die als Datentabelle definiert ist. Darüber hinaus wird das Laden der Datentabelle initialisiert.

Bei einer großen Datentabelle kann das Laden beträchtlich viel Zeit in Anspruch nehmen. Entwicklung für Zugriff auf Datentabellen beschreibt die Anwendungsprogrammierungsbefehle, die mit einer benutzerverwalteten Datentabelle verwendet werden können, sowie die Art und Weise, in der Leistungssteigerungen begrenzt werden, die mit einer CICS-verwalteten Datentabelle erzielt werden können, bis der Ladevorgang abgeschlossen ist.

Die folgenden Schritte werden beim Öffnen der Datei ausgeführt:

1. Der Steuerblock für die Zugriffsmethode (ACB, Access Method Control Block) für die VSAM-Quellendatei wird unter einem separaten MVS-Tasksteuerblock (TCB, Task Control Block) geöffnet. Dieser Schritt ist derselbe wie für jede CICS-Datei.
2. Für die erste Datentabelle, die von einer Region verwendet wird, führt CICS Folgendes aus:
  - Erstellung eines MVS-Speicherpools, der von der Unterstützung für gemeinsam genutzte Datentabellen verwendet wird
  - Erstellung eines MVS-Datenspeicherbereichs, der von den Datentabellen dieser Region verwendet wird
  - Versuch einer LOGON-Operation als Server
3. Die spezielle CICS-Transaktion CFTL wird angehängt, um die Datentabelle in den Datenspeicherbereich zu laden.
4. Die Transaktion, die die Anforderung zum Öffnen der Datentabelle abgesetzt hat, kann nun die Verarbeitung fortsetzen.
5. CICS gibt die Nachricht DFHFC0940 aus, um anzuzeigen, dass das Laden gestartet wurde. Die Nachricht wird an die CSFL-Warteschlange mit transienten Daten gesendet.
6. Die Transaktion, die die Datentabelle lädt, liest die Quellendatei sequenziell. Unter der optionalen Steuerung des Benutzerexits XDTRD kopiert die Transaktion die Datensätze in den Datenspeicherbereich.
7. CICS gibt eine Nachricht aus, um das Ergebnis des Ladens anzuzeigen. Die Nachrichtennummer lautet:
  - Wenn das Laden erfolgreich war: DFHFC0941
  - Wenn das Laden fehlschlägt: DFHFC0942, DFHFC0943, DFHFC0945, DFHFC0946, DFHFC0947 oder DFHFC0948

Die Nachricht wird an die CSFL-Warteschlange mit transienten Daten gesendet. Wenn das Laden fehlschlägt, wird die Nachricht auch an die Konsole gesendet.

8. Wenn das Laden abgeschlossen ist (ob erfolgreich oder nicht erfolgreich), wird der Benutzerexit XDTLC aufgerufen, wenn er aktiv ist. Wenn das Laden nicht erfolgreich abgeschlossen wurde, kann das Exitprogramm anfordern, dass die Datentabelle geschlossen wird.
9. Bei einer benutzerverwalteten Datentabelle wird der Steuerblock für die Zugriffsmethode (ACB) für die Quellendatei geschlossen, wenn das Laden abgeschlossen ist. Die Datei wird freigegeben, wenn sie ursprünglich dynamisch zugeordnet wurde, und wird anderen Jobs zur Verfügung gestellt, sofern für sie noch keine anderen ACBs geöffnet sind.

**Anmerkung:** Bei einem Wiederanlauf nach Systemabsturz wird jede Datei, die eine Backout-Aktion erfordert, erneut geöffnet. Wenn die Datei jedoch als Datentabelle definiert ist, wird das Laden zu diesem Zeitpunkt nicht eingeleitet; stattdessen wird sie von der CSFU-Transaktion am Ende des Wiederanlaufs nach Systemabsturz eingeleitet. Dies bietet die Möglichkeit für alle Benutzerexits, die das Kopieren von Datensätzen in die Datentabelle während des Ladens steuern, damit sie in jeder Phase der PLTPI-Verarbeitung aktiviert werden können.

## Datentabellen schließen

Eine Datentabelle wird auf dieselbe Weise wie jede andere CICS-Datei geschlossen.

Verwenden Sie eine der folgenden Methoden:

- Explizit durch eine CEMT- oder EXEC CICS-Anforderung, die vom Benutzer ausgegeben wurde
- Implizit durch das normale Beenden von CICS

Alle Regeln und Optionen zum Schließen einer CICS-Datei gelten auch für eine Datentabelle.

Insbesondere gilt Folgendes:

- Die Regeln für das Versetzen von aktuellen Dateibenutzern in den Wartemodus werden angewendet (mit Ausnahme der Möglichkeit, dass die Datei geschlossen werden kann, selbst wenn eine Transaktion, die in einer Application-Owning Region (AOR) ausgeführt wird, in der Mitte einer Suchsequenz steht).
- Wenn bei einer benutzerverwalteten Datentabelle die Datentabelle als wiederherstellbar definiert ist, müssen alle Arbeitseinheiten, die die Datentabelle geändert haben, abgeschlossen sein, bevor die Datentabelle geschlossen werden kann.

Der Datenspeicherbereich, der für die Datentabellendatensätze verwendet wird, wird im Rahmen der Schließoperation freigegeben. Wenn eine Datei nach dem Schließen erneut geöffnet wird, entspricht die Verarbeitung dieselbe, als wäre die Datei noch nicht geöffnet worden.

## Unterstützung für gemeinsam genutzte Datentabellen in einem Sysplex verwenden

Lesen Sie diese Informationen, wenn Sie momentan benutzerverwaltete Datentabellen in einer einzigen MVS-Umgebung verwenden, aber planen, in eine Sysplex-Umgebung zu wechseln. Es kann auch hilfreich sein, wenn Sie bereits einen Sysplex haben, da er Ihnen zeigen kann, wie die Unterstützung für gemeinsam genutzte Datentabellen in dieser Umgebung genutzt werden kann.

## Übersicht über die Unterstützung für gemeinsam genutzte Datentabellen in einem Sysplex

Eine gemeinsam genutzte Datentabelle kann die Unterstützung für gemeinsam genutzte Datentabellen nur in einem einzigen MVS-Image nutzen. Sie können jedoch die Verwendung von gemeinsam genutzten Datentabellen in einer Sysplex-Umgebung für eine Anwendung erweitern, die nur Lesezugriff auf eine gemeinsam genutzte benutzerverwaltete Datentabelle erfordert, oder für eine Anwendung, für die nicht erforderlich ist, dass Änderungen sofort angezeigt werden.

Beachten Sie, dass eine gemeinsam genutzte Datentabelle mit der Funktionsverlagerung über MVS-Images gemeinsam genutzt werden kann.

Sie können eine benutzerverwaltete Datentabelle im gesamten Sysplex mit einer Datentabelle pro MVS replizieren. Sie müssen in jedem MVS-Image über genau eine Region, die als gemeinsam genutzter Datentabellenserver eingesetzt wird, verfügen, die jeweils eine benutzerverwaltete Datentabelle (UMT) beinhaltet, auf die von allen anderen CICS-Regionen in diesem MVS zugegriffen werden kann. Diese anderen Regionen erfordern ferne Dateidefinitionen, die sich auf die benutzerverwaltete Datentabelle in ihrer Serverregion beziehen. Jede benutzerdefinierte Datentabelle (UMT) muss über dieselbe Quellendatei verfügen, und diese Datei muss von allen gemeinsam genutzten Datentabellenserverbereichen gelesen werden können. Wenn der Zugriff schreibgeschützt ist und die Daten nie aktualisiert werden, wird dadurch eine gemeinsam genutzte benutzerverwaltete Datentabelle in einem Sysplex zur Verfügung gestellt.

Wenn sich die zu Grunde liegenden Daten von Zeit zu Zeit ändern (was wahrscheinlich ist) und diese Änderungen jedoch nicht sofort in den benutzerverwalteten Datentabellen (UMTs) berücksichtigt werden müssen, können Sie in regelmäßigen Abständen eine Verarbeitung durchführen, um den Inhalt der UMTs zu aktualisieren, damit sie aktualisiert und den zu Grunde liegenden Daten entsprechen, ohne dass die UMTs geschlossen und neu geladen werden müssen. Änderungen werden nicht auf die benutzerverwaltete Datentabelle angewendet, sondern auf die Quellendatei, wobei CICS-Anwendungen, die sich durch eine Nicht-Datentabellendateidefinition auf die Datei beziehen, oder Stapelverarbeitungsprogramme verwendet werden. Ein Beispiel für ein COBOL-Anwendungsprogramm, das in „Quellcode für das Beispielprogramm zum Aktualisieren einer replizierten benutzerverwalteten Datentabelle“ auf Seite 51 beschrieben wird, veranschaulicht, wie Sie die benutzerverwalteten Datentabellen (UMTs) aktualisieren können, um den aktuellen Inhalt der Quellendatei wiederzugeben. Das Programm wird auf jedem MVS-Image ausgeführt und aktualisiert die UMT in diesem Image. Ein solches Programm kann in regelmäßigen Abständen am Tag oder auf Benutzeranforderung ausgeführt werden. Am effizientesten ist es, sie in den Regionen des gemeinsam genutzten Datentabellenservers auszuführen, um Funktionsverlagerung-Aktualisierungen an der UMT zu vermeiden.

Wenn es wichtig ist, dass die CICS-Regionen in allen MVS-Images im Sysplex in ihrer Datenansicht synchronisiert werden, müssen die Transaktionen gestoppt werden, die die Daten lesen, während die Aktualisierungsprogramme ausgeführt werden, und erst dann erneut gestartet werden, nachdem die Programme auf allen MVS-Systemen abgeschlossen wurden.

Dieses Verfahren ist nur für benutzerverwaltete Datentabellen geeignet, da Folgendes gilt:

- Wenn Lesezugriff erforderlich ist, wird normalerweise eine benutzerverwaltete Datentabelle verwendet.
- Mit einer CICS-verwalteten Datentabelle ist es nicht möglich, Aktualisierungen auf die Quellendatei anzuwenden und die Tabellen dabei unverändert beizubehalten.
- Jede Aktualisierung, die an der Quellendatei vorgenommen wurde, würde nur in der Tabelle auf dem System übernommen werden, auf dem die Aktualisierung durchgeführt wurde.

## **Vorgehensweise beim Aktualisieren von replizierten benutzerverwalteten Datentabellen**

In den folgenden Schritten wird beschrieben, wie eine Umgebung für die Aktualisierung replizierter benutzerverwalteten Datentabellen (UMTs) eingerichtet wird. In der Praxis haben Sie vielleicht schon einiges eingerichtet. Möglicherweise verfügen Sie bereits über Dateien, die als Datentabellen definiert sind. Bei den hier beschriebenen Schritten wird davon ausgegangen, dass Sie bereits über eine Sysplex-Umgebung verfügen.

### **1. Wählen Sie eine geeignete Datei aus.**

Stellen Sie sich als Beispiel eine Anwendung vor, die Kreditkartennummern anhand einer Liste gestohlener Kreditkarten überprüft und die einen schnellen Zugriff auf diese Liste benötigt. Die Liste wird in regelmäßigen Abständen mit neuen Stapeln von gestohlenen Karten aktualisiert. Die Anwendung greift auf Datensätze in einer VSAM-KSDS-Datei mit dem Namen PRODN.SOURCEDS unter Verwendung des Dateinamens UMTNAME zu. Die Anwendung wird in einem Sysplex ausgeführt, der aus zwei MVS-Images besteht. Die CICS-Regionen CICS1A, CICS1B sowie CICS1C werden im ersten Image ausgeführt, und CICS2A, CICS2B sowie CICS2C werden im zweiten ausgeführt.

### **2. Richten Sie Dateidefinitionen ein:**

- Wählen Sie in jedem MVS im Sysplex eine CICS-Region aus, die der gemeinsam genutzte Datentabellenserver für diese Datei sein soll. Definieren Sie in dieser Region den Dateinamen, unter dem Ihre Anwendungen die Daten als eine benutzerverwaltete Datentabelle lesen, wobei der Dateiname auf den Dateinamen festgelegt wird, der die Quelldaten enthält.

In diesem Beispiel werden CICS1A und CICS2A als Serverregionen konfiguriert, und es werden Dateien für sie mit dem Namen UMTNAME definiert. Die Dateidefinitionen geben DSNAME als PRODN.SOURCEDS, TABLE als USER und zulässige Operationen von YES für READ, BROWSE, ADD, DELETE und UPDATE an. (Da diese Dateidefinition sowohl zum Lesen der Daten als auch zum Aktualisieren der UMT verwendet wird, wenn sie aktualisiert wird.)

- Definieren Sie für alle anderen Regionen im Sysplex den Dateinamen, unter dem die Anwendungen die Daten als ferne Datei mit REMOTESYSTEM als gemeinsam genutzte Datentabellen-Serverregion im selben MVS lesen, sowie REMOTENAME als Namen der benutzerdefinierten Datentabelle (UMT) in dieser Region.

In diesem Beispiel werden die Dateien UMTNAME in CICS1B und CICS1C mit dem Befehl REMOTESYSTEM als SYSID für CICS1A und REMOTENAME als UMTNAME definiert, diesmal mit READ und BROWSE als die einzigen zulässigen Operationen, da die UMT nicht über diese fernen Definitionen aktualisiert werden muss. Ähnliche Dateidefinitionen werden in CICS2B und CICS2C konfiguriert, aber für diese CICS2A ist das ferne System vorhanden.

- Konfigurieren Sie in jeder gemeinsam genutzten Datentabellen-Serverregion eine Dateidefinition, die zum Lesen der Quelldatei verwendet werden kann, wenn die benutzerdefinierten Datentabellen (UMTs) aktualisiert werden.

In diesem Beispiel sind Dateien mit dem Namen SOURCEDS für CICS1A und CICS2A definiert, wobei DSNAME als PRODN.SOURCEDS, TABLE als NO sowie nur READ- und BROWSE-Operationen zulässig sind.

- Definieren Sie in einer Region im Sysplex (die Zugriff auf die Quelldatei hat) eine Datei, die für die Anwendung von Aktualisierungen auf die Quelle verwendet wird. Die Dateidefinition kann dieselbe sein wie die, die das Aktualisierungsprogramm zum Lesen der Quelldatei verwendet. Aber in diesem Fall müssen sowohl Lese- als auch Aktualisierungsoperationen zulässig sein. Sie können die Datei auch mit einem Stapelverarbeitungsprogramm aktualisieren, falls Sie dies bevorzugen. In diesem Fall ist diese CICS-Dateidefinition nicht erforderlich.

In diesem Beispiel wird dieselbe Dateidefinition wie bei der Aktualisierung der benutzerdefinierten Datentabellen (UMTs) verwendet. In diesem Fall müsste eine der Regionen SOURCEDS definieren, damit alle Dateioperationen zulässig sind.

3. Richten Sie die Quelldatei so ein, dass alle Anwendungen, die sie lesen oder aktualisieren müssen, darauf zugreifen können.

Wenn Sie DFSMS/MVS Version 1 Release 3 besitzen, können Sie zum Lesen oder Aktualisieren aus einer beliebigen CICS-Region auf die Datei zugreifen, indem Sie RLSACCESS(Yes) in den Dateidefinitionen angeben. Wenn Sie den RLS-Zugriffsmodus verwenden, können Sie die Aktualisierung, sofern es sich nicht um eine nicht wiederherstellbare Datei handelt, nicht über ein Stapelverarbeitungsprogramm auf die Datei anwenden (da nur CICS eine wiederherstellbare Datei für die Aktualisierung im RLS-Modus öffnen kann).

Wenn Sie ein früheres Release von DFSMS/MVS besitzen, können Sie die Datei SHAREOPTIONS so konfigurieren, dass sie durch das Programm, das Aktualisierungen auf die Quelle anwendet, aktualisiert und von allen anderen Programmen gelesen werden kann. Alternativ können Sie die Datei so konfigurieren, dass sie nur dann aktualisiert werden kann, wenn sie nicht gelesen wird, und damit sicherstellen, dass ihre Öffnen serialisiert wird. Damit SHAREOPTIONS im gesamten Sysplex ausgeführt werden kann, müssen Sie GRS (Global Resource Serialization) verwenden.

Definieren Sie, wenn RLS nicht verfügbar ist, in diesem Beispiel PRODN.SOURCEDS entweder mit:

- SHR (2), sodass es von der Region aktualisiert werden kann, die das Programm ausführt, das Änderungen auf die Datei anwendet und gleichzeitig von allen Aktualisierungsprogrammen gelesen wird,

oder

- SHR (1), normalerweise haben Sie es für das Programm offen, das Änderungen anwendet. Wenn es aktualisiert werden soll, schließen Sie den Zugriff darauf, und öffnen Sie ihn in jedem Serverbereich, führen Sie das Aktualisierungsprogramm aus und inaktivieren Sie es, damit die nächste Region geöffnet werden kann.
- 4. Ändern Sie das Beispielprogramm so, dass es Ihre Dateien für die benutzerverwaltete Datentabelle (UMT) und die Quellendatei benennt und dass die Datendefinitionen dem Layout Ihrer Datensätze entsprechen. Definieren Sie das Programm und die Transaktion in Ihren Serverregionen.  
  
Die Dateinamen im Beispiel sind dieselben wie die im Programm (UMTNAME und SOURCEDs). Definieren Sie das Programm und eine Transaktion für die Ausführung in CICS1A und CICS2A.
- 5. Sie sollten jetzt die replizierten benutzerverwalteten Datentabellen (UMTs) verwenden können.
- 6. Bereiten Sie die Quellendatei mit seinem ursprünglichen Inhalt vor.
- 7. Öffnen Sie die UMTs in den gemeinsam genutzten Datentabellen-Serverregionen, damit der Inhalt der Quellendatei in jede geladen wird.
- 8. Starten Sie die Anwendungen in allen Regionen des Sysplex. Sie können dank der gemeinsamen Nutzung der Datentabelle auf die Daten zugreifen.  
  
Die Anwendungen, die in MVS 1 ausgeführt werden, greifen auf die Daten über die UMT in CICS1A zu, und die in MVS 2 ausgeführten Anwendungen greifen über die UMT in CICS2A auf sie zu.
- 9. Wenn neue Daten eintreffen, aktualisieren Sie die Quellendatei.  
  
In diesem Beispiel werden die Daten durch die Datei SOURCEDs aktualisiert.
- 10. Wenn die Anwendungen auf die neuen Daten zugreifen sollen, führen Sie die Transaktionen in den einzelnen Serverregionen aus, die die Quellendatei und die benutzerdefinierte Datentabelle (UMT) lesen. Aktualisieren Sie sie, damit sie dem vorherigen Schritt entspricht. Wenn Ihre Anwendungen nicht ungültig werden, wenn sich die Daten in einem MVS geringfügig von den Daten in einem anderen MVS unterscheiden, müssen Sie sie während der Aktualisierung nicht anhalten.

## Beispielprogramm zum Aktualisieren einer benutzerverwalteten Datentabelle

Um Ihnen beim Schreiben eines eigenen Programms zu helfen, haben wir hier ein Beispiel für ein COBOL-Programm, das veranschaulicht, wie eine benutzerverwaltete Datentabelle (UMT) aktualisiert wird, während sie geöffnet ist, sodass sie mit der Quellendatei übereinstimmt.

Wenn Aktualisierungen häufig auf die Quellendatei angewendet und möglicherweise angewendet werden, während das Aktualisierungsprogramm ausgeführt wird, kann dies bedeuten, dass die Quellendatei nie genau von der UMT abgebildet wird, da der verarbeitete Datensatz oder bereits verarbeitete Datensätze geändert werden können. Dies bedeutet, dass das Programm tolerant gegenüber der Möglichkeit sein muss, dass Datensätze geändert werden. Das Programm wurde auch so geschrieben, dass die Möglichkeit besteht, dass die benutzerverwaltete Datentabelle (UMT) selbst von anderen Programmen aktualisiert wird. Es wird jedoch nicht empfohlen, auf diese Weise zu arbeiten (d. h. das einzige Programm, das die UMT aktualisiert, sollte das Aktualisierungsprogramm sein).

### Funktionsweise des Beispielprogramms

Zuerst wird die Umgebung initialisiert. Es wird geprüft, ob die UMT-Datei lokal und bereits geöffnet ist. Wenn es sich um die UMT-Datei um eine ferne Datei handelt, gibt das Programm eine Nachricht aus und wird beendet. Wenn die UMT-Datei nicht geöffnet ist, wird sie geöffnet und beendet (weil beim Öffnen der UMT die neuesten Daten aus der Quellendatei geladen werden, ohne dass eine weitere Verarbeitung erforderlich ist). Es wird auch überprüft, ob die Quellendatei lokal ist. Wenn es sich um eine ferne Quellendatei handelt, gibt das Programm eine Nachricht aus und wird beendet. Die Datei, die direkt auf die Quellendatei der UMT zugreift, wird geöffnet. Anschließend werden Suchoperationen für beide Dateien ausgeführt, damit das Programm nacheinander durchlaufen werden kann.

Wenn die Umgebung ohne Fehler konfiguriert ist, wird die Aktualisierung der UMT gestartet. Dies umfasst das Abrufen und Vergleichen von Datensatzpaaren, eine aus der UMT und eine aus der Basisdatei.

Die abgerufenen Datensätze werden verglichen:

- Wenn die Datensätze gleich sind, werden die Flags gesetzt, um den nächsten Datensatz aus der benutzerverwalteten Datentabelle (UMT) und der Datei zu lesen.
- Wenn die UMT einen größeren Schlüssel als die Datei hat, gibt es einen Datensatz in der Datei, der der UMT hinzugefügt werden muss.
- Wenn die Datei einen größeren Schlüssel als die UMT hat, ist in der UMT ein zusätzlicher Datensatz vorhanden, der entfernt werden muss.
- Wenn die Schlüssel gleich sind, die Datensätze jedoch unterschiedlich sind, muss die UMT mit dem Datensatz in der Datei aktualisiert werden.

Wenn der UMT ein Datensatz hinzugefügt werden muss, wird eine Schreiboperation ausgeführt.

- Wenn die Schreiboperation erfolgreich ist, setzt das Programm die Verarbeitung mit dem nächsten Datensatzpaar fort.
- Wenn die Schreiboperation aufgrund eines Datensatzes fehlschlägt, der von einer anderen Transaktion zwischen dem Lesevorgang und der vom Programm ausgeführten Schreiboperation eingefügt wurde, wird versucht, den Datensatz zu löschen und ihn erneut zu schreiben.
- Wenn der zweite Versuch fehlschlägt, verarbeitet das Programm das nächste Datensatzpaar.
- Wenn das nächste Datensatzpaar verarbeitet wird, wird der aktuelle UMT-Datensatz mit dem nächsten Datensatz in der Datei verglichen, um zu prüfen, ob weitere UMT-Datensatzauslassungen vorhanden sind.

Wenn ein Datensatz aus der UMT gelöscht werden muss, wird eine Löschoperation ausgeführt.

- Wenn die Löschoperation erfolgreich ist, setzt das Programm die Verarbeitung mit dem nächsten Datensatzpaar fort.
- Wenn die Löschoperation fehlschlägt, weil der Datensatz bereits zwischen den Lese- und Löschoperationen gelöscht wurde, verarbeitet das Programm das nächste Datensatzpaar.
- Wenn das nächste Datensatzpaar verarbeitet wird, wird der Datensatz der aktuellen Datei mit dem nächsten Datensatz in der UMT verglichen, um weitere Datensätze zu überprüfen, die nicht in der UMT enthalten sein sollten.

Wenn ein Datensatz in der UMT aktualisiert werden muss, wird ein Lesevorgang für die Aktualisierungsoperation ausgeführt, um eine Sperre für den Datensatz zu erhalten.

- Wenn dies erfolgreich ist, wird der aktualisierte Datensatz in die UMT umgeschrieben, setzt das Programm die Verarbeitung mit dem nächsten Datensatzpaar fort.
- Wenn die Operation fehlschlägt, weil eine andere Transaktion den Datensatz gelöscht hat, wird eine Schreiboperation ausgeführt, um sie wieder zurückzuholen.
- Wenn die Schreiboperation fehlschlägt, setzt das Programm die Verarbeitung mit dem nächsten Datensatzpaar fort.
- Wenn das nächste Datensatzpaar verarbeitet wird, werden neue Datensätze sowohl aus der UMT als auch aus der Datei gelesen.

Wenn das Ende der beiden Dateien erreicht ist und keine weiteren Datensätze mehr für die Verarbeitung vorhanden sind, beendet das Programm das Durchsuchen sowohl in der Datei als auch in der UMT und kehrt zurück. Beachten Sie, dass in diesem Beispiel die Datei, die direkt auf den Datensatz zugreift, nicht geschlossen wird. Wenn die Datei in einer gemeinsam genutzten Umgebung nicht zur Aktualisierung ausgeführt werden kann, muss die Datei, auf die zugegriffen wird, auf CLOSED DISABLED gesetzt sein, damit eine Aktualisierung möglich ist.

Das Programm fängt alle unerwarteten Fehler ab und gibt eine Fehlermeldung auf dem Bildschirm aus. Nur die erste Operation in der UMT wird geprüft (entweder die Löscho-, Schreib- oder Lese-/Überschreiboperation). Wenn dies mit einem Rückgabecode fehlschlägt, der möglicherweise durch einen Datensatz verursacht wurde, der nach dem ursprünglichen Lesen geändert wurde, wird ein letzter Versuch unternommen, den Datensatz zu korrigieren, aber dieser Versuch wird nicht überprüft. Dadurch wird verhindert, dass das Programm in einen Schleifenzustand eintritt.

Es gibt weitere Kommentare im Code.



### **Beispielprogramm einrichten und ausführen**

Bearbeiten Sie das Programm entsprechend den Kommentaren in dem Beispiel, um das Format der Datensätze, die aktualisiert werden, abzugleichen.

UMTNAME und SOURCEDS sollten so umbenannt werden, dass sie mit den Dateidefinitionen übereinstimmen.

Führen Sie mit einem COBOL-Compiler die Umsetzung, Kompilierung und Verlinkung des Programms durch.

Definieren Sie das Programm für CICS und definieren Sie eine Transaktion für das Programm. Definieren Sie die Datei (UMTNAME) so, dass sie auf die benutzerverwaltete Datentabelle (UMT) verweist, und weisen Sie ihr eine Quellendatei zu, aus der sie beim ersten Öffnen geladen werden soll. Definieren Sie die andere Datei (SOURCEDS) so, dass sie direkt auf die Quellendatei verweist, aus der die benutzerverwaltete Datentabelle (UMT) geladen werden soll.

Jeder Sysplex sollte über eine CICS-Region verfügen, in der sich die benutzerverwaltete Datentabelle (UMT) befindet, die aktualisiert werden soll. In diesen Regionen müssen die Definitionen installiert sein, die für die Ausführung der Aktualisierungstransaktion erforderlich sind. In allen anderen Regionen des Sysplex sollte die UMT als ferne Datei definiert werden, die auf die UMT in der UMT-Ownning-Region verweist. Es ist nicht erforderlich, die Aktualisierungstransaktion in den Regionen auszuführen, für die die UMT als ferne UMT definiert ist.

Die verwendete Aktualisierungsstrategie hängt von der Art und Weise ab, wie die Quellendatei konfiguriert ist. Wenn die Quellendatei als RLS konfiguriert ist, können alle benutzerverwalteten Datentabellen (UMTs) gleichzeitig aktualisiert werden. Alle Aktualisierungen der Quellendatei können ebenfalls angewendet werden. Wenn für die Datei SHAREOPTIONS festgelegt ist, damit sie von mehreren Systemen gleichzeitig gelesen werden kann, kann wie bei RLS auch eine gleichzeitige Aktualisierung erfolgen. Wenn die Quellendatei aktualisiert wird, muss anderenfalls die Datei, die zum Lesen der Quellendatei für die Aktualisierung verwendet wird, auf jedem System für die Dauer der Aktualisierung geschlossen und inaktiviert werden. Wenn alle benutzerverwalteten Datentabellen (UMTs) seriell aktualisiert werden, kann die Quellendatei für jede UMT-Ownning-Region geöffnet und geschlossen werden, wenn sie für die Aktualisierung benötigt wird.

### **Quellcode für das Beispielprogramm zum Aktualisieren einer replizierten benutzerverwalteten Datentabelle**

Dieser Quellcode wird nur in dieser Dokumentation, nicht in der CICS-Beispielbibliothek bereitgestellt.

### **Beispielprogramm zum Aktualisieren einer replizierten benutzerverwalteten Datentabelle (UMT): CBL XOPTS (SP)**

#### **Programmname**

UMTUPDT COBOL

#### **Beschreibender Name**

CICS-Anwendung zum dynamischen Aktualisieren einer benutzerverwalteten Datentabelle (UMT) mit dem aktuellen Inhalt einer Datei

#### **Übersicht**

Dieses Programm veranschaulicht, wie eine benutzerverwaltete Datentabelle (UMT) so aktualisiert wird, dass sie mit den Daten in der Quellendatei übereinstimmt, aus der sie geladen wurde, während sie weiterhin von einem (oder mehreren) CICS-Systemen verwendet wird. Es kann verwendet werden, um eine benutzerverwaltete Datentabelle (UMT) zu aktualisieren, das in verschiedenen Sysplexen repliziert wird, sodass sie alle mit der Quellendatei übereinstimmen. Es sollte in der File-Ownning Region (FOR) ausgeführt werden.

#### **Anforderungen**

Dieses Programm sollte umgesetzt und kompiliert sowie als CICS COBOL-Programm verknüpft und für CICS definiert werden. Für dieses Programm sollte ein Transaktionsname definiert werden. Eine UMT-Datei mit dem Namen UMTNAME wird für den Zugriff auf die UMT-Datei verwendet. Und eine Quellendatei, die derzeit SOURCEDS heißt, wird verwendet, um direkt auf die Datei zuzugreifen, von der die UMT geladen wird. Diese Definitionen dürfen nur in der Region installiert werden, in der sich die benutzerverwaltete Datentabelle (UMT) befindet, also in der File-Ownning Region (FOR). Alle

Regionen im selben Sysplex, die die benutzerverwaltete Datentabelle (UMT) fern verwenden, müssen keinen Aktualisierungsprozess ausführen.

### **Beschreibung**

Das Programm initialisiert zunächst die zwei erforderlichen Dateien und beginnt mit dem Durchsuchen der Dateien von Anfang an. Wenn Sie die benutzerverwaltete Datentabelle (UMT) öffnen, wird sie geladen, wenn sie noch nicht geöffnet ist. Wenn sie nicht geöffnet ist und die benutzerverwaltete Datentabelle (UMT) geladen wird, ist der Programmbetrieb effektiv redundant und der Aktualisierungscode wird nicht ausgeführt. Das Programm prüft außerdem den fernen Systemnamen. Wenn ein Name für eine der Dateien vorhanden ist, wird das Programm nicht ausgeführt. Dadurch wird verhindert, dass eine Funktionsverlagerung auftritt, die die Leistung deutlich vermindert.

Das Programm liest kontinuierlich ein Datensatzpaar aus den zwei Dateien und vergleicht sie, indem es alle Datensätze in der UMT, die nicht mit der Quellendatei übereinstimmen, hinzufügt, löscht oder aktualisiert.

Die Schlüssel der Datensatzpaare werden verglichen. Wenn der Schlüssel für die UMT und der Schlüssel für die Quellendatei identisch sind und die Datensätze übereinstimmen, ist keine Aktualisierung erforderlich. Wenn beide Schlüssel identisch, aber die Datensätze unterschiedlich sind, wird der Datensatz in der Quellendatei verwendet, um die UMT zu aktualisieren. Wenn der Schlüssel in der UMT größer als der Quellendateischlüssel ist, werden die Datensätze in der Quellendatei erst dann in die UMT geschrieben, wenn die Schlüssel identisch sind oder der UMT-Schlüssel kleiner als der Quellendateischlüssel ist. Wenn der UMT-Schlüssel kleiner als der Quellendateischlüssel ist, werden die Datensätze in der UMT erst dann entfernt, wenn die Schlüssel identisch sind oder der UMT-Schlüssel größer als der Quellendateischlüssel ist. Dies wird so lange fortgesetzt, bis das Ende der beiden Dateien erreicht ist oder ein unerwarteter Fehler auftritt.

Alle unerwarteten Fehler werden auf dem Bildschirm angezeigt und das Programm wird gestoppt. Einige Fehler werden abgefangen und es wird ein weiterer Versuch unternommen, die UMT zu aktualisieren. Schlägt dieser Versuch fehl, wird für diese Datensätze keine weitere Aktion ausgeführt, und das Programm setzt die Verarbeitung mit dem nächsten Paar fort.

### **Programm ändern**

Dieses Programm funktioniert möglicherweise nicht wie gewünscht. Die vom Programm verwendete Datensatzstruktur setzt voraus, dass ein 4-Zeichen-Schlüssel für den Zugriff auf einen Datensatz mit 40 Zeichen verwendet wird. Daher müssen die folgenden Änderungen vorgenommen werden, damit dieses Programm mit unterschiedlichen Datensatztypen arbeiten kann.

Der Schlüssel, der auf die benutzerverwaltete Datentabelle (UMT) und auf die Quellendatei zugreift, muss geändert werden. Die Variablen, die den Schlüssel speichern, lauten UMT-KEY und DS-KEY.

Die Länge der Datensätze wird in UMT-LEN und DS-LEN festgelegt.

Die Datensatzvariablen für die benutzerverwaltete Datentabelle (UMT) und die Quellendatei müssen geändert werden. Die Variablen, die diese Werte speichern, heißen UMT-REC (enthält UMT-REC-KEY und UMT-REC-TEXT) und DS-REC (enthält DS-REC-KEY und DS-REC-TEXT). Weitere Felder können selbstverständlich nach Bedarf hinzugefügt werden.

Der Dateiname der benutzerverwalteten Datentabelle (UMT) wird als UMTNAME festgelegt. Dieser Wert kann so geändert werden, dass er mit einer bereits definierten UMT übereinstimmt. Die Quellendatei ist als SOURCEDS festgelegt und kann auch geändert werden.

### **Quellcode**

```
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.
* 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.

```



---

# Kapitel 6. Fehlerbehebung für Datentabellen

Verwenden Sie die Trace- und Speicherauszugsdaten, die von CICS erstellt werden, um die Ursache eines Problems mit gemeinsam genutzten Datentabellen zu ermitteln.

Erläuterungen zu den Diagnosenachrichten und den Abbruchcodes, die von gemeinsam genutzten Datentabellen generiert werden, sind in [CICS-Nachrichten](#) enthalten.

## Traceinformationen für Datentabellenservices

Die von CICS erstellte Tracetabelle hilft Ihnen, die Ursache eines Problems zu ermitteln. Sie zeigt den Steuerungsablauf durch die CICS-Module an. Die hier beschriebenen Einträge werden von Datentabellenservices in die Tracetabelle aufgenommen.

Informationen zum Inhalt der Tracetabelle und zum Abrufen der Tracetabelle finden Sie im Abschnitt [CICS-Trace verwenden](#).

Es gibt zwei Typen von Tracepunkten:

- Eingangs- und Ausgangstracepunkte für jeden der Services, die von der Unterstützung für gemeinsam genutzte Datentabellen bereitgestellt werden. Die Tracefunktion der Dateisteuerung auf Ebene 2 muss aktiviert sein, damit diese Tracepunkte abgerufen werden können.
- Ausnahmebedingungstracepunkte.

## Eingangs- und Ausgangstracepunkte für gemeinsam genutzte Datentabellen

Diese Eingangs- und Ausgangstracepunkte werden von den gemeinsam genutzten Datentabellenservices bereitgestellt.

### **OB13**

Eingangspunkt zum Remote-Read-Service

### **OB14**

Ausgangspunkt vom Remote-Read-Service

### **OB1B**

Eingangspunkt zum Initialisieren des Datentabellen-Unterstützungsservices

### **OB1C**

Ausgangspunkt von der Initialisierung des Datentabellen-Unterstützungsservices

### **OB1D**

Eingangspunkt zum Logon-Service

### **OB1E**

Ausgangspunkt vom Logon-Service

### **OB1F**

Eingangspunkt zum Load-Service

### **OB20**

Ausgangspunkt vom Load-Service

### **OB21**

Eingangspunkt zu Services zum Öffnen, Schließen, Festlegen der Aktivierung und zur Statistik

### **OB22**

Ausgangspunkt von Services zum Öffnen, Schließen, Festlegen der Aktivierung und zur Statistik

### **OB23**

Eingangspunkt zu Services für das lokale Lesen

### **OB24**

Ausgangspunkt von Services für das lokale Lesen

**OB25**

Eingangspunkt zu Aktualisierungsservices (Datensatz hinzufügen, hinzufügen, ersetzen, löschen)

**OB26**

Ausgangspunkt von Aktualisierungsservices

**OB2D**

Eingangspunkt zu Connect- und Disconnect-Services

**OB2E**

Ausgangspunkt von Connect- und Disconnect-Services

Das Format jeder dieser Tracepunkte wird in [CICS-Trace verwenden](#) beschrieben.

**Funktion- und Qualifikationsmerkmal-Flags für gemeinsam genutzte Datentabellen**

Bei gemeinsam genutzten Datentabellen enthält jeder Eintrags- und Austragstracepunkt ein Funktionsfeld, und die meisten von ihnen enthalten ein Qualifikationsmerkmal-Flag. Das Funktionsfeld ist ein Byte, das die Funktion angibt, die ausgeführt wurde. Das Qualifikationsmerkmal-Flag ist ein Byte, das Flags enthält, die einige der Funktionen qualifizieren.

Die Werte dieser Felder lauten wie folgt:

<i>Tabelle 6. Funktion- und Qualifikationsmerkmal-Flags sowie Werte</i>	
<b>Funktion</b>	<b>Qualifikationsmerkmal-Flags</b>
X'00' Initialisieren	X'00' als gemeinsam genutzter Datentabellenserver X'80' als Anforderer der gemeinsam genutzten Datentabelle
X'02' Eintrag aus Quelle hinzufügen	X'00' Ausgabe als Ergebnis von Daten hinzufügen Tabellen-Leseanforderung festlegen X'40' Ausgabe durch Ladetransaktion hinzufügen
X'03' Eintrag in Tabelle schreiben	X'00' Schreiben abgeschlossen X'80' Vor dem Schreiben für CMT
X'04' Eintrag neu in Tabelle schreiben	X'00' Neu schreiben abgeschlossen X'80' Vor dem neuen Schreiben für CMT
X'05' Eintrag aus Tabelle löschen	X'00' Löschen abgeschlossen X'80' Vor dem Löschen für CMT
X'06' Festschreiben der benutzerverwalteten Datentabellenaktualisierungen durch dieses UOW	
X'07' Rollback-Operation der benutzerverwalteten Datentabellenaktualisierungen durch dieses UOW	
X'08' Datentabelle laden (nur für Exit-Trace)	X'00' Laden OK X'80' Quellendatei leer
X'09' Verweis auf einen Datensatz	X'80' Abgleich X'40' größer als Abgleich X'20' kleiner als Abgleich (kann in verschiedenen Kombinationen vorkommen) X'10' Test, ob Datentabelle aktiviert ist

Tabelle 6. Funktion- und Qualifikationsmerkmal-Flags sowie Werte (Forts.)

Funktion	Qualifikationsmerkmal-Flags
X'0A' Datensatz nach Schlüssel abrufen	X'80' Abgleich X'40' größer als Abgleich X'20' kleiner als Abgleich (kann in verschiedenen Kombinationen vorkommen) X'10' Test, ob Datentabelle aktiviert ist
X'0B' Datensatz nach Token abrufen	X'80' Abgleich (interner Direktaufruf für eine Datensatzsequenz) X'40' größer als Abgleich X'20' kleiner als Abgleich (kann in verschiedenen Kombinationen vorkommen) X'10' Test, ob Datentabelle aktiviert ist
X'0C' Anmeldung als Server	
X'0E' Öffnen einer Datentabelle	
X'0F' Schließen einer Datentabelle	
X'10' Erfassung von Statistikdaten	
X'11' Aktivierungsstatus festlegen	X'00' Datentabelle aktivieren X'80' Datentabelle inaktivieren X'40' Inaktivierung erzwingen (immer in Kombination mit Inaktivierung)
X'15' Verbindung zu einer gemeinsam genutzten Datentabelle herstellen	
X'16' Verbindung zu einer gemeinsam genutzten Datentabelle abbrechen	
X'17' Abschluss des Ladens verarbeiten	

#### Antwortcodes für gemeinsam genutzte Datentabellen

Jeder Exit-Tracepunkt für gemeinsam genutzte Datentabellen enthält ein Feld für den Antwortcode und ein Feld für den Ursachencode aus 2 Byte.

Das erste Byte ist der Antwortcode, für den die möglichen Werte wie folgt lauten:

##### **X'01'**

Erfolgreich

##### **X'02'**

Ausnahmebedingung

##### **X'03'**

Disaster

##### **X'04'**

Ungültig

##### **X'06'**

Bereinigt

## Ursachencodes für gemeinsam genutzte Datentabellen

Jeder Exit-Tracepunkt für gemeinsam genutzte Datentabellen enthält ein Feld für den Antwortcode und ein Feld für den Ursachencode aus 2 Byte.

Das zweite Byte ist der Ursachencode, für den die möglichen Werte unten angegeben werden. Dieser Ursachencode enthält möglicherweise zugehörige Fehlercodeinformationen. Der Fehlercode ist ein 4-Byte-Feld, das auch entweder in einer Fehlermeldung oder in einem Ausnahmebedingungs-Tracepunkt angegeben wird. Die möglichen Werte sind in CICS-Nachrichten, „Fehler aus den SVC-Datentabellen analysieren“ auf Seite 66 und „Fehler von speicherübergreifenden Datentabellenservices analysieren“ auf Seite 69 beschrieben.

### **X'01'**

Datensatz nicht in Datentabelle

### **X'02'**

Duplikat (Datensatz bereits in Datentabelle)

### **X'03'**

Datentabelle voll (enthält bereits die maximale Anzahl von Datensätzen)

### **X'04'**

Von Benutzerexit abgelehnter Datensatz

### **X'05'**

Fehler beim Abrufen des Speichers

### **X'06'**

Datensatz nicht in Datentabelle (und Tabelle ist bekannterweise vollständig)

### **X'07'**

Datentabellenservice fehlgeschlagen

### **X'08'**

Keine Berechtigung zum Herstellen der Verbindung zur Datei

### **X'09'**

Ressource ist keine Datentabelle

### **X'0A'**

Fernes System hat sich nicht als Server angemeldet

### **X'0B'**

Ladeanforderung fehlgeschlagen

### **X'0C'**

Datentabelle ist inaktiviert

### **X'0D'**

Hinzufüganforderung (von DASD) absichtlich nicht verarbeitet

### **X'0E'**

Datensatz ist zu lang

### **X'0F'**

Datentabellentoken ist ungültig

### **X'10'**

Datensatz nicht in der Datentabelle (aber möglicherweise in der Quelldatei)

### **X'11'**

Datentabelle wird nicht geschlossen, da andere Dateien sie noch verwenden

### **X'12'**

Reserviert

### **X'13'**

Datensatz ist in der Datentabelle vorhanden, ist aber momentan nicht gültig

### **X'14'**

Datei kann nicht geschlossen werden, da sie inaktiviert ist

**X'15'**

Protokollfehler

**X'16'**

CICS ist kein MVS-Subsystem

**X'17'**

Keine Berechtigung zum Herstellen einer Verbindung zu dieser Datei

**X'18'**

CICS kann keine speicherübergreifenden Services verwenden

**X'19'**

Blockformat des Schnittstellenparameters wurde nicht erkannt

**Benutzerverwaltete Datentabelle (UMT) und andere Flags für gemeinsam genutzte Datentabellen**

Dieses Markierungsbyte ist im Eingangstracepunkt für OPEN enthalten.

Die signifikanten Bits bei der Öffnungszeit sind:

**B'1.....'**

CICS-verwaltete Datentabelle

**B'01.....'**

Wiederherstellbare benutzerverwaltete Datentabelle

**B'00.....'**

Nicht wiederherstellbare benutzerverwaltete Datentabelle

**Ausnahmebedingungstracepunkte für gemeinsam genutzte Datentabellen**

Diese Ausnahmebedingungstracepunkte werden von gemeinsam genutzten Datentabellenservices bereitgestellt.

**AP 0B0A**

Nicht erkannte Funktion beim Aufruf von DFHDTRE

**AP 0B0B**

Nicht erkannte Funktion beim Aufruf von DFHDTRR

**AP 0B0C**

Nicht erkannte Funktion beim Aufruf von DFHDTUP

**AP 0B0D**

Nicht erkannte Funktion beim Aufruf von DFHDTST

**AP 0B0E**

Nicht erkannte Funktion beim Aufruf von DFHDTSS

**AP 0B0F**

Nicht erkannte Funktion beim Aufruf von DFHDTRC

**AP 0B10**

Fehler beim Initialisieren der Datensatzverwaltung

**AP 0B11**

Fehler beim Datensatzmanager OPEN

**AP 0B12**

Fehler beim Datensatzmanager CLOSE

**AP 0B15**

Unerwarteter Fehler beim Aufruf zum Abrufen des PCs

**AP 0B19**

Fehler beim Aufrufen der Datentabellen SVC beim Initialisieren als Server

**AP 0B1A**

Fehler beim Aufrufen der Datentabellen SVC beim Initialisieren als Anforderer

**AP 0B27**

CLOSE konnte Tabellenblock nicht finden

**AP 0B28**

CLOSE konnte Dateiblock nicht finden

**AP 0B29**

Fehler beim Aufrufen der Datentabellen SVC beim Anmelden als Server

**AP 0B2A**

Fehler beim Aufrufen der Datentabellen SVC beim Verbinden oder Trennen

**AP 0B2B**

XDTRD-Exit hat eine ungültige Datensatzlänge zurückgegeben (d. h., er hat die Länge für eine CMT geändert oder die Länge für eine UMT erhöht)

**AP 0B2C**

Verbindungsindex überschreitet die maximal unterstützte Größe

**AP 0B2F**

Katastrophaler Fehler beim Anfordern von Speicher, Parameter zum Laden von Transaktionen zu übergeben

Das Format jeder dieser Tracepunkte wird in CICS-Trace verwenden beschrieben.

## Fehler aus den SVC-Datentabellen analysieren

Nach einem Fehler von einem Aufruf der Datentabellen SVC wird immer ein Ausnahmebedingungs-Tracepunkt erstellt, einschließlich eines Fehlercodefelds, um die Ursache für den Fehler zu identifizieren. Diese Trace-Punkte sind AP 0B12, 0B19, 0B1A, 0B29 und 0B2A.

Es gibt drei Kategorien von SVC-Fehlern:

1. Zu erwartende Bedingungen, z. B. wenn die ferne Datei bei einem Verbindungsversuch keine Datentabelle ist oder das ferne System nicht als freigegebener Datentabellenserver angemeldet ist. CICS führt die entsprechende Aktion für solche Bedingungen aus und es sind keine Diagnoseinformationen erforderlich.
2. Fehler, die durch Probleme in der Umgebung verursacht werden können, die möglicherweise korrigiert werden können. Für diese Fehler wird eine Nachricht mit dem Ursachencode für den Fehler ausgegeben. Die Erklärung des Ursachencodes ist in der Erläuterung der Nachricht in CICS-Nachrichten enthalten.
3. Fehler, die auf einen Logikfehler oder unsachgemäßen Gebrauch der Routinen hinweisen, möglicherweise beim Versuch, Integritäts- oder Sicherheitsprüfungen zu umgehen. Diese Fehler werden von der CICS-Dateikontrolle als katastrophale Fehler behandelt, was einen Systemspeicherauszug (wenn Sie derartige Speicherauszüge aktiviert haben) zur Folge hat und in den meisten Fällen dazu führt, dass die Transaktion mit AFCZ ABEND abgelehnt wird. Hier lautet der Wert für das Antwort- und Ursachenfeld normalerweise X'0215'.

In den folgenden Abschnitten werden die Fehlercodes für die dritte Kategorie von Fehlern erläutert. Diese Fehlercodes werden nur im Eintrag im Ausnahmebedingungs-Trace angezeigt. Das Format des Fehlercodes ist X'ffaaaaaa', wobei *ff* den Fehlertyp angibt und *aaaaaa* zusätzliche Informationen sind, die für einige der Fehler bereitgestellt werden. Die möglichen Werte von *ff* für jeden Tracepunkt werden in den folgenden Abschnitten beschrieben.

## Werte für alle Tracepunkte für gemeinsam genutzte Datentabellen

Die folgenden Fehlercodes können bei den Ausnahmeablaufverfolgungspunkten 0B12, 0B19, 0B1A, 0B29 und 0B2A auftreten.

**X'01'**

Es wurde eine Funktion angegeben, die erfordert, dass der Aufrufende über den CICS AFCB (Authorized Function Control Block) autorisiert wird. Der Aufrufende wurde jedoch nicht autorisiert.

**X'0A'**

Der Aufrufende hat einen ungültigen Funktionscode übergeben.



**X'0B'**

Der Aufrufende hat ein ungültiges Format des SVC-Aufrufs angegeben.

**X'0C'**

Eine ungültige Parameterlistenadresse wurde an den SVC übergeben.

**X'0D'**

Es wurde eine Funktion angegeben, die erfordert, dass der in Register 1 übergebene Wert 0 ist, aber dies war nicht der Wert. Die zusätzlichen Informationen enthalten die niedrigwertigen drei Byte des übergebenen Werts.

**X'12'**

Es wurde eine Funktion angegeben, die den Aufrufenden im Supervisorstatus "Schlüssel 0" benötigt, aber der Anrufer war nicht in diesem Status.

**Werte für den Tracepunkt 0B12**

Der Ausnahmebedingungs-Tracepunkt AP 0B12 wird ausgegeben, wenn vom SVC beim Hinzufügen oder Löschen eines Zugriffslisteneintrags ein Fehler zurückgegeben wird, wenn eine gemeinsam genutzte Datentabelle geschlossen wird.

Zusätzlich zu den Fehlern, die an allen Tracepunkten auftreten können, sind die folgenden Fehler möglich:

**X'02'**

Gemeinsam genutzte Datentabellenservices wurden noch nicht initialisiert (es wurde kein Ankerblock für die Region erstellt).

**X'0E'**

Der angegebene Datenspeicherbereich STOKEN ist ungültig, oder der Aufrufende ist nicht berechtigt, ihn zu verwenden.

**X'0F'**

Die CICS-Region hat die Initialisierung als Server nicht abgeschlossen.

**X'13'**

Der Versuch, einen Zugriffslisteneintrag zu löschen, ist fehlgeschlagen, da der angegebene Eintrag nicht von den Datentabellen SVC erstellt wurde.

Alle anderen Fehler führen dazu, dass eine Nachricht ausgegeben wird, die den Fehlercode enthält.

**Werte für den Tracepunkt 0B19**

Der Ausnahmebedingungs-Tracepunkt AP 0B19 wird ausgegeben, wenn der SVC beim Initialisieren als gemeinsam genutzter Datentabellenserver einen Fehler zurückgibt.

Zusätzlich zu den Fehlern, die an allen Tracepunkten auftreten können, sind die folgenden Fehler möglich:

**X'02'**

Es wird versucht, einen Zugriffslisteneintrag hinzuzufügen, bevor die CICS-Region die Initialisierung der gemeinsam genutzten Datentabelle ausgeführt hat (ein Ankerblock für die Region wurde noch nicht erstellt).

**X'0E'**

Der angegebene Datenspeicherbereich STOKEN ist ungültig, oder der Aufrufende ist nicht berechtigt, ihn zu verwenden.

**X'0F'**

Es wurde versucht, einen Zugriffslisteneintrag hinzuzufügen, bevor die CICS-Region die Serverinitialisierung abgeschlossen hat.

Alle anderen Fehler führen dazu, dass eine Nachricht ausgegeben wird, die den Fehlercode enthält.

**Werte für den Tracepunkt 0B1A**

Der Ausnahmebedingungs-Tracepunkt AP 0B1A wird ausgegeben, wenn der SVC beim Initialisieren als gemeinsam genutzter Datentabellenserver einen Fehler zurückgibt.

Zusätzlich zu den Fehlern, die an allen Tracepunkten auftreten können, sind die folgenden Fehler möglich:

**X'05'**

Die CICS-Region wurde bereits als Anforderer der gemeinsam genutzten Datentabelle initialisiert, wird jedoch jetzt unter einem anderen Anforderungsblock ausgeführt, als sie ursprünglich initialisiert wurde.

Alle anderen Fehler führen dazu, dass eine Nachricht mit dem entsprechenden Fehlercode ausgegeben wird.

## **Werte für den Tracepunkt AP 0B29**

Der Ausnahmebedingungs-Tracepunkt AP 0B29 wird ausgegeben, wenn der SVC beim Anmelden als gemeinsam genutzter Datentabellenserver einen Fehler zurückgibt.

Zusätzlich zu den Fehlern, die an allen Tracepunkten auftreten können, sind die folgenden Fehler möglich:

**X'02'**

Die CICS-Region, die versucht, sich als Server zu registrieren (anzumelden), wurde noch nicht initialisiert. (Es wurde kein Ankerblock für die Region erstellt.)

**X'04'**

Die CICS-Region ist bereits als gemeinsam genutzter Datentabellenserver registriert (angemeldet).

**X'0F'**

Die CICS-Region hat die Serverinitialisierung nicht abgeschlossen.

**X'14'**

Der AFCS-Ankerblock ist nicht vorhanden.

**X'15'**

Der CICS-Sicherheitsblock ist nicht vorhanden.

**X'16'**

Entweder wird der Aufrufende nicht in einem Benutzer-Speicherschutzschlüssel ausgeführt (der PSW-Schlüssel ist kleiner als 8) oder der TCB des Aufrufenden wird normalerweise nicht in einem Benutzer-Speicherschutzschlüssel ausgeführt (TCBPKF ist kleiner als 8).

Alle anderen Fehler führen dazu, dass eine Nachricht ausgegeben wird, die den Fehlercode enthält.

## **Werte für den Tracepunkt 0B2A**

Wenn das Funktionscode-Feld X'15' enthält, zeigt der Ausnahmebedingungs-Tracepunkt AP 0B2A einen Fehler bei der CONNECT-Operation an (d. h. bei dem Versuch, eine Verbindung zu einer fernen Datei herzustellen).

Zusätzlich zu den Fehlern, die an allen Tracepunkten auftreten können, sind die folgenden Fehler möglich:

**X'02'**

Gemeinsam genutzte Datentabellenservices wurden noch nicht initialisiert (es wurde kein Ankerblock für die Region erstellt).

**X'03'**

Die anfordernde Region hat die Initialisierung nicht als Anforderer für gemeinsam genutzte Datentabellen abgeschlossen.

**X'05'**

Die CICS-Region wird unter einem anderen Anforderungsblock (RB, Request Block) ausgeführt als dem, unter dem sie als Anforderer der Datentabelle initialisiert wurde. Der Teil des Fehlercodes mit zusätzlichen Informationen enthält die RB-Adresse, unter der der Aufruf getätigt wurde.

**X'72'**

Der LINK zum durch den Benutzer austauschbaren DFHACEE-Modul, um die Sicherheitsbenutzer-ID des Speicherbereichs für die Ausgangsadresse zu suchen, ist fehlgeschlagen. Der zusätzliche

Informationsteil des Fehlercodes enthält zwei Byte des ABEND-Codes aus dem LINK. Das Antwort- und Ursachenfeld für diesen Fehler lautet X'020B'.

Alle anderen Fehler führen dazu, dass eine Nachricht ausgegeben wird, die den Fehlercode enthält.

Wenn das Funktionscode-Feld X'16' enthält, zeigt der Ausnahmebedingungstracepunkt AP 0B2A einen Fehler bei der DISCONNECT-Operation an (d. h. bei dem Versuch, eine Verbindung zu einer fernen Datei zu unterbrechen). Zusätzlich zu den Fehlern, die an allen Tracepunkten auftreten können, sind die folgenden Fehler möglich:

**X'02'**

Gemeinsam genutzte Datentabellenservices wurden noch nicht initialisiert (es wurde kein Ankerblock für die Region erstellt).

**X'03'**

Die anfordernde Region hat die Initialisierung nicht als Anforderer für gemeinsam genutzte Datentabellen abgeschlossen.

**X'05'**

Die CICS-Region wird unter einem anderen Anforderungsblock (RB, Request Block) ausgeführt, als er als Anforderer der Datentabelle initialisiert wurde. Der zusätzliche Informationsteil des Fehlercodes enthält die RB-Adresse, unter der der Aufruf getätigt wurde.

**X'07'**

Der Aufrufende hat einen ungültigen Index in den Vektor der Dateiverbindungen gestellt. Der zusätzliche Teil des Fehlercodes enthält die drei niedrigstwertigen Byte des Index des Aufrufenden.

**X'10'**

Die angegebene Verbindung wurde zuvor unterbrochen und ist nicht mehr vorhanden. Der zusätzliche Informationsteil des Fehlercodes enthält die drei niedrigstwertigen Byte des Index des Aufrufenden im Vektor der Dateiverbindungen.

Alle anderen Fehler führen dazu, dass eine Nachricht ausgegeben wird, die den Fehlercode enthält.

## Fehler von speicherübergreifenden Datentabellenservices analysieren

Nach einem unerwarteten Fehler von speicherübergreifenden Datentabellenservices wird der Eintrag im **X'0B15'-Eintrag im Ausnahmebedingungstrace** erstellt. Dazu gehören die Antwort- und Ursachencodes sowie ein Fehlercodefeld, das die Fehlerursache angibt. Diese Fehler werden alle durch eine Beschädigung der Routinen oder des Systems bzw. durch einen unsachgemäßen Gebrauch der Routinen verursacht.

Bei Antwort und Ursache von X'0215' lautet das Format des Fehlercodes X'ffaaaaaa'. Dabei gibt *ff* den Fehlertyp an und *aaaaaa* sind zusätzliche Informationen, die für einige Fehler bereitgestellt werden. Die möglichen Werte für *ff* sind:

**X'01'**

Der Versuch, den CICS AFCB (Authorized Function Control Block) zu lokalisieren, der entweder durch die speicherübergreifende Abrufroutine oder durch die Suchroutine für Verbindungsvektoren ausgeführt wurde, ist fehlgeschlagen.

**X'02'**

Die anfordernde CICS-Region hat noch keine Initialisierung der gemeinsam genutzten Datentabellen ausgeführt. (Es wurde noch kein Ankerblock für die Region erstellt und eingerichtet.)

**X'03'**

Die anfordernde Region hat die Initialisierung nicht als Anforderer für gemeinsam genutzte Datentabellen abgeschlossen.

**X'05'**

Die Abrufanforderung wurde unter einem anderen Anforderungsblock abgesetzt, als der, der die Initialisierung als gemeinsam genutzter Datentabellenanforderer ausgeführt hat.

**X'06'**

Der Verbindungsvektoreintrag für die ferne Datei enthält keinen korrekten Linkage-Index.

### **X'07'**

Der Index des Verbindungsvektoreintrags für die ferne Datei befindet sich hinter dem Ende des Verbindungsvektors.

### **X'08'**

Der Eintrag für den Verbindungsvektor für die ferne Datei ist nicht als im Gebrauch markiert.

### **X'09'**

Die speicherübergreifende Abrufroutine wurde über den falschen Mechanismus abgerufen.

Antwort und Ursache von X'0400' bedeuten, dass der Funktionscode, der an den in der Serverregion ausgeführten Datensatzverwaltungscode übergeben wurde, einen nicht erkennbaren Wert aufweist.

## **Speicherauszugsinformationen für Datentabellen**

Informationen, die für Datentabellen relevant sind, werden in einem CICS-Systemspeicherauszug aufgenommen, um die Ursache eines Problems zu ermitteln.

Informationen zum Inhalt von Speicherauszügen und zum Abrufen dieser Speicherauszüge finden Sie in Speicherauszüge bei der Fehlerbestimmung verwenden.

Die wichtigsten Steuerblöcke, die von der Unterstützung für gemeinsam genutzte Datentabellen verwendet werden, sind im Bereich FILE CONTROL eines formatierten Speicherauszugs der File-Ownning Region enthalten. Diese Steuerblöcke heißen:

### **Globaler Bereich der Datentabelle**

Dies wird auch als *Headerblock der gemeinsam genutzten Datentabelle* bezeichnet, deshalb verwendet er die Strukturkennung DFHDTHDR.

### **Basis-Bereich der Datentabelle**

Dies wird auch als *Tabellenblock der gemeinsam genutzten Datentabelle* bezeichnet, deshalb verwendet er die Strukturkennung DFHDTTABLE.

### **Pfad-Bereich der Datentabelle**

Dies wird auch als *Dateiblock der gemeinsam genutzten Datentabelle* bezeichnet, deshalb verwendet er die Strukturkennung DFHDTFILE.

Der Inhalt der Datentabelle ist nicht im CICS-Systemspeicherauszug enthalten, da der Datenspeicherbereich, in dem sich die Datentabelle befindet, nicht Teil des CICS-Adressraums ist. Die Tabelleneinträge befinden sich im Datenspeicherbereich DFHDT001, die Indexknoten in DFHDT002 und die Datensatzdaten in den Datenspeicherbereichen DFHDT003, DFHDT004, DFHDT005 usw., wobei neue Datenspeicherbereiche nach Bedarf hinzugefügt werden. Wenn der Inhalt der Datentabelle angezeigt werden soll, bitten Sie den Systembediener, den MVS-Befehl **DUMP** zu verwenden, um einen Speicherauszug des jeweiligen Datenspeicherbereichs anzufordern, dessen Eigner der entsprechende CICS-Startjob ist.

Der Bedienerbefehl `DISPLAY J,CICS-Startjobname` zeigt Informationen zu einem CICS-Job an, einschließlich der DSPNAME-Werte der Datenspeicherbereiche, deren Eigner der jeweilige Job ist. Um den Inhalt des Datenspeicherbereichs DFHDT003 zu erstellen, können Sie den MVS-Befehl **DUMP** wie folgt verwenden:

1. Geben Sie Folgendes ein:

```
DUMP COMM=(Titel für Ihren Speicherauszug)
```

2. Dadurch wird eine MVS-Konsolnachricht generiert:

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

3. Antworten Sie auf die Nachricht mit:

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

Der Datenspeicherbereich wird erstellt.

**Anmerkung:** Durch Verwendung eines Sterns im Befehl kann ein Speicherauszug für *alle* Datenspeicherbereiche erstellt werden, deren Eigner CICS ist:

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

Dies sollte jedoch mit Bedacht verwendet werden, da bei vielen Datenspeicherbereichen die Speicherauszugsdatei riesig sein kann.

4. Verwenden Sie `DISPLAY DUMP ,TITLE`, um zu sehen, welche `SYS1.DUMPnn`-Datei verwendet wurde.



## Bemerkungen

---

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. IBM stellt dieses Material möglicherweise auch in anderen Sprachen zur Verfügung. Für den Zugriff auf das Material in einer anderen Sprache kann eine Kopie des Produkts oder der Produktversion in der jeweiligen Sprache erforderlich sein.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

*IBM Director of Licensing  
IBM Europe, Middle East & Africa  
Tour Descartes 2, avenue Gambetta  
92066 Paris La Defense  
France*

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuauflage veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

*IBM Director of Licensing  
IBM Europe, Middle East & Africa  
Tour Descartes  
2, avenue Gambetta  
92066 Paris La Defense  
France*

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesen Informationen beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen

Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden und jede Ähnlichkeit mit Namen und Adressen tatsächlicher Personen oder Unternehmen ist rein zufällig.

#### **COPYRIGHTLIZENZ:**

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch die Verwendung der Beispielprogramme entstehen.

#### **Informationen zu Programmierschnittstellen**

Die von CICS zur Verfügung gestellte Dokumentation kann teilweise als Programmierschnittstelle betrachtet werden und zum Teil nicht.

Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zur Nutzung der Services von CICS Transaction Server for z/OS, Version 5 Release 6 zu schreiben, sind in folgenden Abschnitten der Online-Produktdokumentation enthalten:

- [Anwendungen entwickeln](#)
- [Systemprogramme entwickeln](#)
- [CICS TS-Sicherheit](#)
- [Entwicklung für externe Schnittstellen](#)
- [Referenz zur Anwendungsentwicklung](#)
- [Referenz: Systemprogrammierung](#)
- [Referenz: Konnektivität](#)

Informationen, die NICHT zur Verwendung als Programmierschnittstelle von CICS Transaction Server for z/OS, Version 5 Release 6 bestimmt sind, die aber als Programmierschnittstelle missverstanden werden können, sind in folgenden Abschnitten der Online-Produktdokumentation enthalten:

- [Fehlerbehebung und Unterstützung](#)
- [CICS TS - Referenz zur Diagnose](#)

Wenn Sie auf die CICS-Dokumentation in Handbüchern im PDF-Format zugreifen, sind Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zur Nutzung der Services von CICS Transaction Server for z/OS, Version 5 Release 6 zu schreiben, in den folgenden Handbüchern enthalten:

- Application Programming Guide und Application Programming Reference
- 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
- Installation Guide
- Security Guide
- Supplied Transactions
- CICSplex SM Managing Workloads
- CICSplex SM Managing Resource Usage
- CICSplex SM Application Programming Guide and Application Programming Reference
- Java™ Applications in CICS

Wenn Sie auf die CICS-Dokumentation in Handbüchern im PDF-Format zugreifen, sind Informationen, die NICHT zur Verwendung als Programmierschnittstelle von CICS Transaction Server for z/OS, Version 5 Release 6 bestimmt sind, die aber als Programmierschnittstelle missverstanden werden können, in den folgenden Handbüchern enthalten:

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

## Marken

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der International Business Machines Corporation in den USA und/oder anderen Ländern. Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite [Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml) unter [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe, das Adobe-Logo, PostScript und das PostScript-Logo sind Marken oder eingetragene Marken von Adobe Systems Incorporated in den USA und/oder anderen Ländern.

Apache, Apache Axis2, Apache Maven, Apache Ivy, das Apache Software Foundation-Logo (ASF-Logo) und das Logo mit der ASF-Feder sind Marken der Apache Software Foundation.

Gradle und das Gradlephant-Logo sind eingetragene Marken von Gradle, Inc. oder der zugehörigen Tochtergesellschaften in den USA und/oder anderen Ländern.

Intel, das Intel-Logo, Intel Inside, das Intel Inside-Logo, Intel Centrino, das Intel Centrino-Logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken der Intel Corporation oder ihrer Tochtergesellschaften in den USA oder anderen Ländern.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Die eingetragene Marke Linux® wird gemäß einer Unterlizenz der Linux Foundation verwendet, dem exklusiven Lizenznehmer von Linus Torvalds, dem Eigentümer der Marke auf einer weltweiten Basis.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Red Hat® und Hibernate® sind Marken oder eingetragene Marken von Red Hat, Inc. oder der zugehörigen Tochtergesellschaften in den USA und/oder anderen Ländern.

Spring Boot ist eine Marke von Pivotal Software, Inc. in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Zowe™, das Zowe-Logo und Open Mainframe Project™ sind Marken von The Linux Foundation.

## **Nutzungsbedingungen für die Produktdokumentation**

Die Berechtigungen zur Nutzung dieser Veröffentlichungen werden Ihnen auf der Basis der folgenden Bedingungen gewährt.

### **Anwendbarkeit**

Diese Bedingungen sind eine Ergänzung der Nutzungsbedingungen auf der IBM Website.

### **Persönliche Nutzung**

Sie dürfen diese Veröffentlichungen für Ihre persönliche, nicht kommerzielle Nutzung unter der Voraussetzung vervielfältigen, dass alle Eigentumsvermerke erhalten bleiben. Sie dürfen diese Veröffentlichungen oder Teile der Veröffentlichungen ohne ausdrückliche Genehmigung von IBM nicht weitergeben, anzeigen oder abgeleitete Werke davon erstellen.

### **Kommerzielle Nutzung**

Sie dürfen diese Veröffentlichungen nur innerhalb Ihres Unternehmens und unter der Voraussetzung, dass alle Eigentumsvermerke erhalten bleiben, vervielfältigen, weitergeben und anzeigen. Sie dürfen diese Veröffentlichungen oder Teile der Veröffentlichungen ohne ausdrückliche Genehmigung von IBM außerhalb Ihres Unternehmens weder vervielfältigen, weitergeben oder anzeigen noch abgeleitete Werke davon erstellen.

### **Rechte**

Abgesehen von den hier gewährten Berechtigungen werden keine weiteren Berechtigungen, Lizenzen oder Rechte (veröffentlicht oder stillschweigend) in Bezug auf die Veröffentlichungen oder darin enthaltene Informationen, Daten, Software oder geistiges Eigentum gewährt.

IBM behält sich das Recht vor, die hierin gewährten Berechtigungen nach eigenem Ermessen zurückzuziehen, wenn sich die Nutzung der Veröffentlichungen für IBM als nachteilig erweist oder wenn die obigen Nutzungsbestimmungen nicht genau befolgt werden.

Sie dürfen diese Informationen nur in Übereinstimmung mit allen anwendbaren Gesetzen und Vorschriften, einschließlich aller US-amerikanischen Exportgesetze und Verordnungen, herunterladen und exportieren.

IBM übernimmt keine Gewährleistung für den Inhalt dieser Veröffentlichungen. Diese Veröffentlichungen werden auf der Grundlage des gegenwärtigen Zustands (auf "as-is"-Basis) und ohne eine ausdrückliche oder stillschweigende Gewährleistung für die Handelsüblichkeit, die Verwendungsfähigkeit für einen bestimmten Zweck oder die Freiheit von Rechten Dritter zur Verfügung gestellt.

## **IBM Online-Datenschutzerklärung**

IBM Softwareprodukte, einschließlich Software as a Service-Lösungen (*Softwareangebote*), können Cookies oder andere Technologien verwenden, um Informationen zur Produktnutzung zu erfassen, die Endbenutzererfahrung zu verbessern und Interaktionen mit dem Endbenutzer anzupassen oder zu anderen Zwecken. In vielen Fällen werden von den Softwareangeboten keine personenbezogenen Daten erfasst. Einige der IBM Softwareangebote können Sie jedoch bei der Erfassung personenbezogener Daten unterstützen. Wenn dieses Softwareangebot Cookies zur Erfassung personenbezogener Daten verwendet, sind nachfolgend nähere Informationen über die Verwendung von Cookies durch dieses Angebot zu finden:

### **Für die Webbenutzerschnittstelle von CICSplex System Manager (Hauptschnittstelle):**

Abhängig von den bereitgestellten Konfigurationen kann dieses Softwareangebot Sitzungscookies und persistente Cookies zum Erfassen der Benutzernamen und anderer personenbezogener Daten einzelner Benutzer für das Sitzungsmanagement, die Authentifizierung, einen besseren Bedienungskomfort, zur Nutzungsüberwachung und für andere funktionale Zwecke verwenden. Diese Cookies können nicht inaktiviert werden.

### **Für die Webbenutzerschnittstelle von CICSplex System Manager (Datenschnittstelle):**

Abhängig von den bereitgestellten Konfigurationen kann dieses Softwareangebot Sitzungscookies und persistente Cookies zum Erfassen der Benutzernamen und anderer personenbezogener Daten

einzelner Benutzer für das Sitzungsmanagement, die Authentifizierung, einen besseren Bedienungskomfort, zur Nutzungsüberwachung und für andere funktionale Zwecke verwenden. Diese Cookies können nicht inaktiviert werden.

**Für die Webbenutzerschnittstelle von CICSplex System Manager ("hello world"-Seite):**

Abhängig von den bereitgestellten Konfigurationen kann dieses Softwareangebot Sitzungscookies verwenden, die keine personenbezogenen Daten erfassen. Diese Cookies können nicht inaktiviert werden.

**Für CICS Explorer:**

Abhängig von den bereitgestellten Konfigurationen kann dieses Softwareangebot persistente Vorgaben und Sitzungsvorgaben zum Erfassen der Benutzernamen und Kennwörter von Benutzern für das Sitzungsmanagement, die Authentifizierung und zur Single Sign-on-Konfiguration (einmalige Anmeldung) verwenden. Diese Vorgaben können nicht inaktiviert werden, auch wenn die Speicherung eines Benutzerkennworts auf einem Datenträger in verschlüsselter Form nur aktiviert werden kann, indem der Benutzer bei der Anmeldung explizit ein Kontrollkästchen aktiviert.

Wenn es die für dieses Softwareangebot bereitgestellten Konfigurationen Ihnen als Kunde ermöglichen, personenbezogene Daten von Endbenutzern über Cookies und andere Technologien zu erfassen, müssen Sie sich zu allen gesetzlichen Bestimmungen in Bezug auf eine solche Datenerfassung, einschließlich aller Mitteilungspflichten und Zustimmungsanforderungen, rechtlich beraten lassen.

Weitere Informationen zur Nutzung verschiedener Technologien, einschließlich Cookies, für diese Zwecke finden Sie unter [IBM Datenschutzrichtlinie](#) und in der [IBM Online-Datenschutzerklärung](#) im Abschnitt *Cookies, Web-Beacons und sonstige Technologien* sowie auf der Seite [IBM Softwareprodukte und Software-as-a-Service-Datenschutzerklärung](#).



# Index

## Special Characters

Übersicht über gemeinsam genutzte Datentabellen [1](#)

## A

Abbruchcodes

AFCH [32](#), [33](#)

AFCZ [66](#)

AFCH, Abbruchcode [32](#), [33](#)

AFCZ, Abbruchcode [66](#)

Aktivierung von Benutzerexits [42](#)

Aktualisierungsanforderungen

Einführung [2](#)

für eine benutzerverwaltete Datentabelle (UMT) [30](#)

für eine CICS-verwaltete Datentabelle (CMT) [29](#), [30](#)

Alternativindizes [3](#), [21](#)

Anfangsstatus einer Datentabelle

durch CEDA definieren [22](#)

Anforderer

Definition [3](#)

Anpassung durch Benutzerexits [37](#)

Antwortcodes

in Tracepunkten [63](#)

Anwendungsprogrammierung

für benutzerverwaltete Datentabelle (UMT)

Übersicht [30](#)

für eine benutzerverwaltete Datentabelle (UMT)

Beschreibung [30](#)

für eine CICS-verwaltete Datentabelle (CMT)

Beschreibung [29](#)

AOR (Application-Owning Region)

CONNECT-Operation [9](#)

Definition [3](#)

automatisches Journaling [21](#), [22](#)

## B

Benachrichtigung

für CONNECT-Operationen [10](#)

Benutzerexits

aktivieren [42](#)

am Ende des Ladens [42](#)

beim Hinzufügen von Datensätzen [41](#)

beim Laden [40](#)

Beschreibung [37](#)

Definition [42](#)

DSECT für Parameterliste [37](#)

Exitprogrammbeispiele [37](#)

für Dateisteuerung [33](#), [37](#)

für EXEC-Schnittstelle [33](#), [37](#)

Kommunikation mit CICS [37](#)

Parameterliste [37](#)

Übersicht [5](#)

XDTAD, Exit [41](#)

XDTLC, Exit [42](#)

XDTRD, Exit [40](#)

Benutzerexits aktivieren [42](#)

benutzerverwaltete Datentabelle

Aktualisierungsanforderungen [30](#)

beim Laden verwenden [31](#)

Beschreibung [2](#)

Datenintegrität [22](#)

Journaling [22](#)

Leistung [11](#)

Leseanforderungen [30](#)

Replikation in einem Sysplex [46](#)

Ressourcendefinition [20](#)

Suchanforderungen [30](#)

Übersicht [2](#)

Bindungssicherheit [18](#)

## C

CEDA DEFINE FILE, Befehl

Beispiel für eine benutzerverwaltete Datentabelle (UMT) [25](#)

Beispiel für eine CICS-verwaltete Datentabelle (CMT) [24](#)

Beschreibung [22](#)

LOG, Parameter [22](#)

MAXNUMRECS, Parameter [22](#)

OPENTIME, Parameter [22](#)

RECORDFORMAT, Parameter [22](#)

TABLE, Parameter [22](#)

CEMT

INQUIRE, Befehl [27](#), [28](#)

SET, Befehl [26](#), [27](#)

CFTL, Transaktion [45](#)

CICS-verwaltete Datentabelle

Aktualisierungsanforderungen [29](#)

Anforderungen aktualisieren [30](#)

beim Laden verwenden [30](#)

Beschreibung [2](#)

Datenintegrität [21](#)

Journaling [21](#)

Leistung [11](#)

Leseanforderungen [29](#), [30](#)

Ressourcendefinition [20](#)

Suchanforderungen [29](#)

Übersicht [2](#)

CONNECT

durch AOR [9](#), [32](#)

Sicherheitsprüfung [18](#)

CSFU, Transaktion [45](#), [46](#)

## D

Datei

als Datentabelle verwendet [1](#), [13](#)

Dateien auswählen

zur Verwendung als Datentabelle [13](#)

Dateisicherheit [18](#)

Dateisteuerung

Befehle

## Dateisteuerung (*continued*)

### Befehle (*continued*)

Übersicht für benutzerverwaltete Datentabelle (UMT) [30](#)

Übersicht für CICS-verwaltete Datentabelle (CMT) [29](#)

von speicherübergreifenden Services unterstützt [29](#)

Benutzerexits [37](#)

## Dateiverwaltung

Funktionsverlagerung verwenden [5](#)

speicherübergreifende Services verwenden [7](#)

## Datenintegrität

einer benutzerverwalteten Datentabelle (UMT) [22](#)

einer CICS-verwalteten Datentabelle (CMT) [21](#)

## Datenspeicherbereich

Speicherauszug des Inhalts [70](#)

Verwendung durch Datentabellen [11](#)

## Datenspeicherbereiche

Verwendung durch Datentabellen [4](#)

Datentabelle inaktivieren [33](#)

Datentabelle öffnen [45](#)

Datentabelle schließen [33](#), [46](#)

Datentabellen planen [11](#)

DFHDTCV [19](#)

DFHDSVC [19](#)

DFHMRMS [19](#)

DFHXDTS, Copybook [37](#)

## DSECT

für Parameterlisten mit Benutzerexits [37](#)

dynamisches Zurücksetzen von Transaktionen [22](#)

## E

EIBRESP2, Feld [30](#), [34](#)

EXEC-Schnittstelle

Benutzerexits [37](#)

## F

Fehlerbestimmung für Datentabellen [61](#)

ferne Datei

Definition [3](#)

FOR (File-Owning Region)

Definition [3](#)

LOGON-Operation [9](#)

Funktion

für Tracepunkte [62](#)

## G

gemeinsame Nutzung

CONNECT-Operation [9](#)

gemeinsam genutzte Datentabellenoperationen [5](#), [9](#)

in einem Sysplex [3](#)

LOGON-Operation [9](#)

Umgebung [3](#)

Global Resource Serialization, siehe GRS [48](#)

Größe der Datentabelle

durch CEDA definieren [22](#)

mit dem Befehl INQUIRE suchen [27](#), [28](#)

mit dem Befehl SET definieren [27](#)

GRS (Global Resource Serialization) [48](#)

## H

Hinweise zu MVS [13](#), [19](#)

## I

in Serie miteinander verbinden (Daisy Chain) [32](#)

Informationen zum Systemspeicherauszug [70](#)

INQUIRE FILE, Befehl

Beschreibung [27](#), [28](#)

MAXNUMRECS, Parameter [27](#), [28](#)

TABLE, Parameter [27](#), [28](#)

Installation

Hinweise zu MVS [19](#)

Parameterliste [18](#)

INSTLN, Parameter [18](#)

Integrität

von CMT-Daten [21](#)

von UMT-Daten [22](#)

INVREQ, Bedingung [30](#)

## J

Journaling [21](#)

## K

Kommunikation

zwischen CICS und Benutzerexits [37](#)

Konzepte von Datentabellen [1](#)

KSDS (Datei in Schlüsselfolge)

als Quellendatei verwendet [2](#), [3](#)

mit einer benutzerverwalteten Datentabelle (UMT) [22](#)

## L

Lademodule

für Datentabellen erforderlich [20](#)

Ladevorgang

von einer benutzerverwalteten Datentabelle (UMT)

währenddessen verwendet [31](#)

von einer CICS-verwalteten Datentabelle (CMT)

währenddessen verwendet [30](#)

Leistung

einer benutzerverwalteten Datentabelle (UMT) [11](#)

einer CICS-verwalteten Datentabelle (CMT) [11](#)

Vorteile von Datentabellen [11](#)

Leseanforderungen

Einführung [2](#)

für eine benutzerverwaltete Datentabelle (UMT) [30](#)

für eine CICS-verwaltete Datentabelle (CMT) [29](#), [30](#)

Vergleich mit Funktionsverlagerung [34](#)

Vergleich mit VSAM [34](#)

LOADING, Bedingung [31](#)

LOGON

durch die FOR (File-Owning Region) [9](#)

Sicherheitsprüfung [18](#)

lokale Datei

Definition [3](#)

Löschanforderungen

Vergleich mit VSAM [35](#)

Lücke [12](#), [29](#), [33](#)

## M

mehrere Dateien  
mit derselben Quellendatei [21](#)

## N

Nachrichten  
am Ende des Ladens [45](#)  
zu Beginn des Ladens [45](#)  
NOSPACE, Bedingung [30](#)  
NOTFND, Bedingung [30](#), [31](#)

## O

Operationen für Datentabellen [45](#)

## P

Parameterliste  
für Benutzerexits [37](#)  
Planung für Datentabellen [19](#)  
Produktabhängige Programmierschnittstelle [37](#)

## Q

Qualifikationsmerkmal-Flag  
für Tracepunkte [62](#)  
Quellendatei  
für Datentabellen [2](#)  
mit mehreren Dateien [21](#)  
muss eine KSDS (Datei in Schlüssel­folge) sein [3](#)  
unabhängig von der benutzerverwalteten Datentabelle (UMT) [2](#)  
Verwendung mit CICS-verwalteter Datentabelle (CMT) [2](#)

## R

RACF  
als Sicherheitsmanager verwendet [18](#)  
Record Level Sharing [2](#)  
replizierte benutzerverwaltete Datentabellen (UMTs)  
aktualisieren [47](#)  
Ressourcendefinition  
Beschreibung [20](#)  
DEFINE FILE, Befehl [22](#)  
Übersicht für benutzerverwaltete Datentabelle (UMT) [22](#)  
Übersicht für CICS-verwaltete Datentabelle (CMT) [21](#)  
RLS (Record Level Sharing) [2](#)

## S

SAF, Systemberechtigungs­funktion  
für Sicherheitsprüfungen verwendet [18](#)  
Schlüssellänge  
Vergleich mit Funktionsverlagerung [34](#)  
Schnittstelle  
für Benutzerexits [37](#)  
produktabhängige Programmierung [37](#)  
Server  
Definition [3](#)  
SET FILE, Befehl

SET FILE, Befehl (*continued*)  
Beschreibung [26](#), [27](#)  
MAXNUMRECS, Parameter [27](#)  
TABLE, Parameter [27](#)  
SHAREOPTION, VSAM [21](#)  
Sicherheitsprüfung  
bei AOR-Verbindung [18](#)  
bei FOR-Anmeldung [18](#)  
für Datentabellen [18](#)  
Hinweise zu RACF [18](#)  
Vergleich mit Funktionsverlagerung [18](#), [33](#)  
Verwendung von SAF [18](#)  
Speicherauszugsinformationen für Datentabellen [70](#)  
Speichernutzung  
Beschreibung [11](#)  
speicherübergreifende Services  
Fehler analysieren [69](#)  
unterstützte Befehle [29](#)  
Vergleich mit Funktionsverlagerung [5](#), [33](#)  
Verwendung durch Anwendung [32](#)  
Statistiken  
zum Auswählen von Datentabellen [15](#)  
Suchanforderungen  
für eine benutzerverwaltete Datentabelle (UMT) [30](#)  
für eine CICS-verwaltete Datentabelle (CMT) [29](#)  
Vergleich mit Funktionsverlagerung [33](#)  
Vergleich mit VSAM [34](#)  
SUPPRESSED, Bedingung [30](#)  
SVC-Fehler [66](#)  
SYSID, Parameter [32](#)  
Sysplex-Umgebung  
Beispielprogramm einrichten und ausführen [51](#)  
Beispielprogrammcode [51](#)  
Beispielprogrammoperation [49](#)  
Einführung [3](#)  
gemeinsam genutzte Datentabellen in Umgebung  
verwenden [46](#)  
replizierte benutzerverwaltete Datentabellen (UMTs)  
aktualisieren [47](#)  
Systemberechtigungs­funktion (SAF) [18](#)

## T

Traceinformationen  
Antwortcodes [63](#)  
Ausnahmebedingungspunkte [65](#)  
Eingangs- und Ausgangspunkte [61](#)  
Funktion- und Qualifikationsmerkmal-Flags [62](#)  
für Datentabellen [61](#)  
Ursachencodes [64](#)  
Typ der Datentabelle  
durch CEDA definieren [22](#)  
mit dem Befehl INQUIRE suchen [27](#), [28](#)  
mit dem Befehl SET definieren [27](#)

## U

Ursachencodes  
in Tracepunkten [64](#)

## V

Verbindungsabbau

- Verbindungsabbau (*continued*)
  - einer AOR und Datentabelle [10](#), [32](#)
- Verfügbarkeit von Datentabellen [1](#)
- Vorteile
  - von Datentabellen [11](#)
- VSAM
  - Alternativindizes [3](#), [21](#)
  - Basiscluster [21](#)
  - SHAREOPTION [21](#)
  - Steuerblock für Zugriffsmethode [45](#), [46](#)
  - Vergleich mit Datentabellen [34](#)
- VSAM-RLS
  - für die gemeinsame Nutzung von Daten [15](#)
  - geeignet für benutzerverwaltete Datentabelle (UMT) [2](#), [22](#)
  - ungeeignet für CICS-verwaltete Datentabelle (CMT) [2](#), [21](#)
  - Wiederherstellungsattribute [22](#)

## W

- Warteschlangen mit transienten Daten
  - für Nachrichten verwendet [45](#)
- Wiederherstellung von Datentabellen
  - beim Wiederanlauf nach Systemabsturz [46](#)
  - durch CEDA definieren [22](#)

## X

- XDTAD, Benutzerexit
  - Beschreibung [41](#)
- XDTLC, Benutzerexit
  - Beschreibung [42](#)
- XDTRD, Benutzerexit
  - Beschreibung [40](#)





