

CICS Transaction Server for z/OS
Version 5 Release 6

Db2-Handbuch



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.

Inhaltsverzeichnis

Informationen zu dieser PDF-Datei	vii
Kapitel 1. Übersicht über die CICS-Db2-Schnittstelle.....	1
Übersicht: Herstellung einer Verbindung von CICS zu Db2.....	1
Db2-Adressräume	2
Übersicht: Funktionsweise von Threads	3
Thread-Tasksteuerblöcke in der Umgebung für offene Transaktionen	4
Übersicht: Zugriff auf Db2 für CICS-Anwendungsprogramme aktivieren	6
CICS-Anwendungsprogramm mit Zugriff auf Db2 vorbereiten.....	6
Bindeprozess	7
Pläne, Pakete und dynamische Planexits	8
Kapitel 2. CICS-Db2-Verbindung definieren.....	11
Übersicht: Vorgehensweise beim Definieren der CICS-Db2-Verbindung.....	11
Einrichtung zum Anhängen von Db2-Gruppen verwenden.	14
TCBLIMIT und der Systeminitialisierungsparameter MAXOPENTCBS	15
Auswirkungen bei der SQL-Verarbeitung	16
Threaderstellung.....	16
SQL-Verarbeitung.....	16
Verarbeitung der Festschreibung	17
Threadfreigabe	17
Threadbeendigung.....	17
Threads erstellen, verwenden und beenden	18
Geschützte Einstiegsthreads	19
Ungeschützte Einstiegsthreads für kritische Transaktionen	20
Ungeschützte Einstiegsthreads für Hintergrundtransaktionen	21
Pool-Threads.....	21
Threadtypen für optimale Leistung auswählen	22
BIND-Optionen für optimale Leistung auswählen	23
DB2CONN-, DB2ENTRY- und BIND-Optionen koordinieren	24
Kapitel 3. Verwalten mit CICS Db2	27
CICS-Db2-Anschlussfunktion starten.....	27
CICS-Db2-Anschlussfunktion stoppen.....	27
Automatischer Verbindungsabbau bei Beendigung von CICS.....	27
Manueller Verbindungsabbau	28
Unbestätigte Arbeitseinheiten auflösen	28
Unbestätigte Arbeitseinheiten bei Verwendung des Anhängens von Gruppen auflösen	29
Unbestätigte Arbeitseinheiten mithilfe eines Light-Neustarts von Db2 auflösen	29
Wiederherstellung von Resynchronisationsinformationen für unbestätigte Arbeitseinheiten	30
CICS-Db2-Anschlussfunktion verwalten.....	30
Db2-Befehle eingeben	31
SMF für Abrechnung, Statistik und Optimierung von Db2 starten	32
GTF für Abrechnung, Statistik und Optimierung von Db2 starten	33
Von CICS bereitgestellte Transaktionen für CICS Db2.....	33
Befehle an Db2 mithilfe der Transaktion DSNB absetzen	34
DSNB DISCONNECT.....	36
DSNB DISPLAY.....	37
DSNB MODIFY	41
DSNB STOP.....	43

DSNC STRT.....	45
Kapitel 4. Sicherheit für Db2	49
Zugriff auf Db2-bezogene Ressourcen in CICS steuern	50
Benutzerzugriff auf DB2CONN-, DB2TRAN- und DB2ENTRY-Ressourcendefinitionen steuern	51
Ressourcensicherheit verwenden, um den Zugriff auf die DB2ENTRY- und DB2TRAN-Ressourcendefinitionen zu steuern	51
Befehlssicherheit verwenden, um das Ausgeben von SPI-Befehlen für DB2CONN-, DB2ENTRY- und DB2TRAN-Ressourcendefinitionen zu steuern	53
Ersatzsicherheit und AUTHTYPE-Sicherheit zum Steuern des Zugriffs auf die Berechtigungs-IDs verwenden, die CICS für Db2 bereitstellt.....	55
Benutzerzugriff auf Db2-bezogene CICS-Transaktionen steuern.....	57
Für Db2 Berechtigungs-IDs für die CICS-Region und für CICS-Transaktionen bereitstellen	58
Für Db2 Berechtigungs-IDs für eine CICS-Region bereitstellen	59
Primäre Berechtigungs-ID für eine CICS-Region bereitstellen.....	59
Sekundäre Berechtigungs-IDs für eine CICS-Region bereitstellen	60
Db2 Berechtigungs-IDs für CICS-Transaktionen bereitstellen.....	61
Primäre Berechtigungs-ID für CICS-Transaktionen bereitstellen	62
Sekundäre Berechtigungs-IDs für CICS-Transaktionen bereitstellen.....	64
Benutzer für den Zugriff auf Ressourcen in Db2 berechtigen	66
Benutzerzugriff auf Db2-Befehle steuern	66
Benutzerzugriff auf Pläne steuern	67
Db2-Sicherheit auf mehreren Ebenen und auf Zeilenebene	69
Kapitel 5. Aspekte des Anwendungsdesigns und der Anwendungsentwicklung für CICS Db2	71
Beziehung zwischen CICS-Anwendungen und Db2-Plänen und -Paketen gestalten	72
Beispielanwendung.....	73
Db2-Pakete verwenden	74
Für alle Transaktionen einen großen Plan verwenden.....	78
Viele kleine Pläne verwenden	78
Pläne auf der Grundlage der Transaktionsgruppierung verwenden	79
Dynamische Planexits	80
Wenn Sie Pläne für eine Anwendung erstellen müssen, die bereits entwickelt wurde	83
Wenn Sie Pläne innerhalb einer Transaktion wechseln müssen.....	83
Dynamischer Planwechsel	84
Transaktions-ID wechseln, um Pläne zu wechseln	84
Sperrstrategie in der CICS-Db2-Umgebung entwickeln.....	89
SQL, Threadsicherheit und andere Erwägungen bei der Programmierung für CICS-Db2-Anwendungen.....	89
CICS-Db2-Anwendungen für die Verwendung von OTE durch threadsichere Programmierung aktivieren	90
SQL (Sprache).....	93
Qualifiziertes oder nicht qualifiziertes SQL verwenden	93
Ansichten	94
Indexspalten aktualisieren	95
Abhängigkeit von eindeutigen Indizes.....	95
Verarbeitung der Festschreibung	95
Transaktionen serialisieren	95
Seitenkonflikt	96
CICS und die Option CURSOR WITH HOLD.....	98
Befehl EXEC CICS RETURN IMMEDIATE.....	98
AEY9-Abbrüche vermeiden	99
Kapitel 6. Verwendung von JDBC und SQLJ für den Zugriff auf Db2-Daten aus Java-Programmen	101
IBM Data Server Driver for JDBC and SQLJ	101

JVM-Server für die Unterstützung von Db2 konfigurieren	101
Db2-Schema in einem JVM-Server festlegen	102
Programmierung für JDBC- und SQLJ-APIs	102
Serialisiertes SQLJ-Profil in einem JVM-Server bereitstellen	103
Verbindung zu einer Datenbank anfordern	103
DriverManager-Verbindung zu einer Datenbank anfordern	104
DataSource-Verbindung zu einer Datenbank anfordern	105
Überlegungen zu JDBC- und SQLJ-Verbindungen	106
Verbindung zu einer Datenbank schließen	106
Arbeitseinheit festschreiben.....	107
Automatische Festschreibung.....	107
DriverManager-Probleme mit dem Synchronisationspunkt bei Verwendung expliziter und Standard-URLs	108
CICS-Abbrüche während JDBC- oder SQLJ-Anforderungen.....	108
Kapitel 7. CICS-Db2-Programme für Ausführung und Produktion vorbereiten	109
CICS-Db2-Testumgebung.....	109
CICS-Db2-Programm Vorbereitung.....	109
CICS-Formatierungsroutine SQLCA.....	112
Zu bindende Elemente nach einer Programmänderung	113
Bindeoptionen und -aspekte bei Programmen	114
RETAIN.....	114
Isolationsstufe	114
Zeit für Validierung von Plänen	115
ACQUIRE und RELEASE	115
CICS-Db2-Programme testen und debuggen	115
In Produktion gehen: Checkliste für CICS-Db2-Anwendungen	115
CICS-Anwendung optimieren, die auf Db2 zugreift	118
Kapitel 8. Db2 überwachen	121
Von CICS bereitgestellte Abrechnungs- und Überwachungsinformationen	121
Von Db2 bereitgestellte Abrechnungs- und Überwachungsinformationen	122
CICS-Db2-Umgebung überwachen: Übersicht.....	123
CICS-Db2-Anschlussfunktion überwachen.....	124
Mithilfe von CICS-Befehlen überwachen.....	124
Mithilfe von Db2-Befehlen überwachen.....	124
Überwachung mithilfe von CICS Db2-Statistiken.....	125
CICS-Transaktionen überwachen, die auf Db2-Ressourcen zugreifen	127
Db2 bei Verwendung mit CICS überwachen	128
Db2 mithilfe der Db2-Statistikfunktion überwachen	128
Db2 mithilfe der Db2-Abrechnungsfunktion überwachen	130
Db2 mithilfe der Db2-Leistungsfunktion überwachen	130
CICS-System in einer CICS-Db2-Umgebung überwachen.....	130
Abrechnung in einer CICS-Db2-Umgebung: Übersicht.....	131
Abrechnungsdaten, die von der Db2-Abrechnungsfunktion bereitgestellt werden	132
Datentypen in Db2-Abrechnungsdatensätzen	132
Db2-Abrechnungsberichte	134
Db2-Abrechnungsdatensätze und CICS-Leistungsklassendatensätze in Beziehung zueinander setzen	135
Probleme, die beim Abgleich von Db2-Abrechnungsdatensätzen und CICS-Leistungsdatensät- zen auftreten können	136
Beziehung zwischen Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendatensätzen steuern	136
Daten des Db2-Abrechnungsdatensatzes zum Ermitteln der entsprechenden CICS-Leistungs- klassendatensätze verwenden	137
Strategien für den Abgleich von Db2-Abrechnungsdatensätzen und CICS-Leistungsklassenda- tensätzen und für die Zuordnung der Kosten zum Benutzer	138

Abrechnung zur Prozessorbelegung in einer CICS-Db2-Umgebung	141
CLASS1-Prozessorzeit berechnen	146
Prozessorzeit der Abrechnungsklasse CLASS 2	147
CICS- und Db2-Prozessorzeiten für DB2 Version 5 oder früher berechnen	148
CICS- und Db2-Prozessorzeit für Db2 berechnen	148
Kapitel 9. Fehlerbehebung für Db2	151
Thread-Tasksteuerblöcke	151
Wartezustandstypen für CICS-Db2.....	151
Nachrichten für CICS Db2	154
Trace für CICS Db2	154
CSUB-Trace	160
Speicherauszug für CICS-Db2.....	162
Db2-Threadidentifikation	163
Transaktionsabbruchcodes für CICS Db2.....	163
Execution Diagnostic Facility (EDF) für CICS Db2.....	163
Deadlocks in der CICS-Db2-Umgebung bearbeiten.....	165
Zwei Typen von Deadlocks	166
Deadlocks erkennen.....	166
Beteiligte Ressourcen suchen	167
Beteiligte SQL-Anweisungen suchen	167
Verwendeten Zugriffspfad suchen	167
Ermitteln, aus welchem Grund der Deadlock aufgetreten ist.	167
Deadlocks beheben	168
Bemerkungen.....	171
Index.....	177

Informationen zu dieser PDF-Datei

Diese PDF-Datei enthält einführende und den Benutzer unterstützende Informationen zur Bewertung, Installation und Verwendung der CICS-Db2-Anschlussfunktion und zur Definition und Verwaltung Ihrer CICS-Db2-Umgebung.

Details zu den in diesem Buch verwendeten Begriffen und der verwendeten Schreibweise finden Sie in [In der Dokumentation zu CICS verwendete Konventionen und Terminologie](#) im IBM Knowledge Center.

Datum dieser PDF-Datei

Diese PDF-Datei wurde am 28. Mai 2020 erstellt.

Kapitel 1. Übersicht über die CICS-Db2-Schnittstelle

Dieser Abschnitt bietet eine Übersicht über die CICS-Schnittstelle zu Db2.

Übersicht: Herstellung einer Verbindung von CICS zu Db2

Mit CICS wird eine CICS-Db2-Anschlussfunktion bereitgestellt. Die CICS-Db2-Anschlussfunktion bietet CICS-Anwendungen Zugriff auf Db2-Daten, während die Anwendungen in der CICS-Umgebung in Betrieb sind.

CICS koordiniert beim Auftreten von Transaktions- oder Systemfehlern die Wiederherstellung sowohl der Db2-Daten als auch der CICS-Daten.

Die CICS-Db2-Anschlussfunktion stellt eine allgemeine Verbindung zwischen CICS und Db2 her. CICS-Anwendungen verwenden diese Verbindung, um Befehle und Anforderungen an Db2 abzusetzen. Die Verbindung zwischen CICS und Db2 kann jederzeit hergestellt oder beendet werden und CICS und Db2 können unabhängig voneinander gestartet und gestoppt werden. Sie können ein einzelnes Db2-Subsystem benennen, zu dem CICS eine Verbindung herstellt, oder Sie können (wenn Sie Db2 Version 7 oder höher nutzen) die Einrichtung zum Anhängen von Gruppen verwenden, um Db2 zu ermöglichen, ein beliebiges aktives Member einer Db2-Subsystemgruppe mit gemeinsamer Datennutzung für die Verbindung auszuwählen. Sie haben darüber hinaus die Option, dass CICS eine Verbindung zu Db2 automatisch herstellt bzw. erneut herstellt. Ein Db2-System kann von mehreren CICS-Systemen gemeinsam genutzt werden, aber ein CICS-System kann gleichzeitig nur mit einem einzigen Db2-Subsystem verbunden sein.

Sie definieren die CICS-Db2-Verbindung mithilfe von drei unterschiedlichen CICS-Ressourcendefinitionen: DB2CONN (die Db2-Verbindungsdefinition), DB2ENTRY (die Db2-Eintragsdefinition) und DB2TRAN (die Db2-Transaktionsdefinition). Sie müssen eine DB2CONN-Ressourcendefinition installieren, bevor Sie die CICS-Db2-Verbindung starten können. (Verwechseln Sie diese Ressourcendefinition nicht mit dem Systeminitialisierungsparameter DB2CONN, mit dem angegeben wird, ob Sie möchten, dass CICS die Db2-Verbindung während der Initialisierung automatisch startet.) Sie können auch DB2ENTRY- und, falls erforderlich, DB2TRAN-Definitionen erstellen, um sicherzustellen, dass Ihren wichtigen Transaktionen Priorität zugeordnet wird. In [Übersicht: Vorgehensweise beim Definieren der CICS-Db2-Verbindung](#) finden Sie weitere Informationen zu diesen Ressourcendefinitionen.

Anschlussbefehle steuern und zeigen den Status der CICS-Db2-Anschlussfunktion an und sie werden mithilfe der von CICS bereitgestellten Transaktion DSNCL abgesetzt. Die Anschlussbefehle lauten wie folgt:

- STRT – Verbindung zu Db2 starten
- STOP – Verbindung zu Db2 stoppen
- DISP – Status der Threads anzeigen und Statistikdaten anzeigen
- MODI – Merkmale der Verbindung zu Db2 ändern
- DISC – Threads trennen

Die Verbindung zwischen CICS und Db2 ist eine Multithreadverbindung. In der allgemeinen Verbindung zwischen CICS und Db2 gibt es für jede aktive CICS-Transaktion, die auf Db2 zugreift, einen Thread; dabei handelt es sich um eine individuelle Verbindung zu Db2. Threads ermöglichen es jeder einzelnen CICS-Transaktion, auf Db2-Ressourcen wie beispielsweise einen Befehlsprozessor oder einen Anwendungsplan (Informationen, die für Db2 angeben, wie die SQL-Anforderungen des Anwendungsprogramms lauten und welcher Weg für deren Ausführung am effizientesten ist) zuzugreifen. In [Übersicht: Funktionsweise von Threads](#) finden Sie eine vollständige Erläuterung dazu, wie Threads arbeiten.

Wenn ein Anwendungsprogramm, das in der CICS-Umgebung in Betrieb ist, seine erste SQL-Anforderung ausgibt, verarbeiten CICS und Db2 die Anforderung wie folgt:

- Eine Sprachschnittstelle oder ein Stubprogramm mit der Bezeichnung DSNCLI, für die bzw. für das eine Programmverbindung zum Anwendungsprogramm hergestellt wurde, ruft die CICS-Ressourcenmanagerschnittstelle (RMI – Resource Manager Interface) auf.

- Die RMI verarbeitet die Anforderung und übergibt die Steuerung an den taskbezogenen Benutzerexit (TRUE) der CICS-Db2-Anschlussfunktion; bei dem Benutzerexit handelt es sich um das Modul, mit dem Db2 für die einzelnen Tasks aufgerufen wird.
- Die CICS-Db2-Anschlussfunktion terminiert einen Thread für die Transaktion. In dieser Phase überprüft Db2 die Berechtigung und sucht den richtigen Anwendungsplan.
- Db2 übernimmt die Steuerung und die CICS-Db2-Anschlussfunktion wartet, während Db2 die Anforderung verarbeitet.
- Wenn die SQL-Anforderung abgeschlossen ist, übergibt Db2 die angeforderten Daten zurück an die CICS-Db2-Anschlussfunktion.
- CICS erhält die Steuerung nun zurück und die CICS-Db2-Anschlussfunktion übergibt die Daten und gibt die Steuerung an das CICS-Anwendungsprogramm zurück.

Db2-Adressräume

Für Db2 sind einige unterschiedliche Adressräume erforderlich.

In [Abbildung 1 auf Seite 2](#) werden diese Adressräume gezeigt.

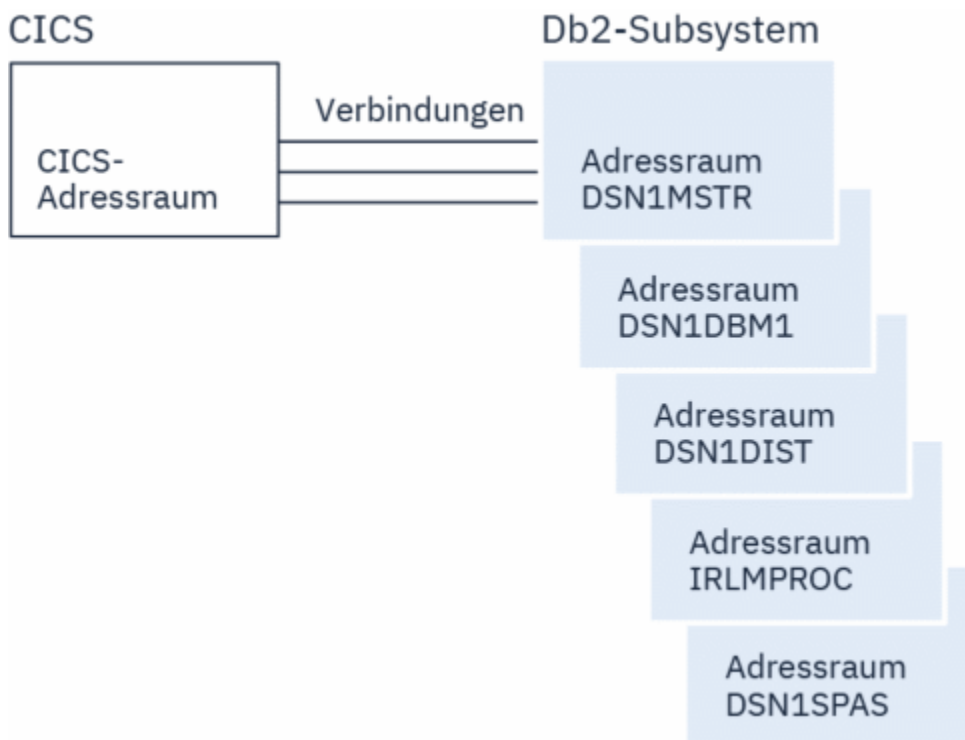


Abbildung 1. Db2-Adressräume

In den verschiedenen Adressräumen werden eine Reihe von Tasks ausgeführt; diese lauten wie folgt:

DSN1MSTR

Für Systemservices, die eine Reihe von systembezogenen Funktionen ausführen.

DSN1DBM1

Für Datenbankservices, die den überwiegenden Teil der Strukturen in benutzererstellten Datenbanken bearbeiten.

DSN1DIST

Für Einrichtungen für verteilte Daten, die Unterstützung für ferne Anforderungen bereitstellen.

IRLMPROC

Für den internen Ressourcensperrenmanager (IRLM – Internal Resource Lock Manager), der Sperrungen in Db2 steuert.

DSN1SPAS

Für gespeicherte Prozeduren, die eine isolierte Ausführungsumgebung für vom Benutzer geschriebenes SQL bereitstellen.

Übersicht: Funktionsweise von Threads

Jede CICS-Transaktion benötigt für den Zugriff auf Db2 einen eigenen zugehörigen Thread.

Threads werden erstellt, wenn sie von Transaktionen benötigt werden, und zwar an dem Zeitpunkt, wenn die Anwendung ihre erste SQL-Anforderung oder Befehlsanforderung ausgibt. Die Transaktion verwendet den Thread für den Zugriff auf Ressourcen, die von Db2 verwaltet werden. Wird ein Thread nicht mehr von der Transaktion benötigt, da die Transaktion auf alle Ressourcen, die sie verwenden muss, zugegriffen hat, wird der Thread freigegeben (in der Regel nach Abschluss des Synchronisationspunkts). Zum Erstellen eines Threads werden Prozessorressourcen benötigt; wenn ein Thread freigegeben wird, überprüft die CICS-Db2-Anschlussfunktion daher, ob der Thread von einer anderen Transaktion benötigt wird. Wenn eine andere Transaktion auf einen Thread wartet, verwendet die CICS-Db2-Anschlussfunktion den vorhandenen Thread wieder, damit diese Transaktion auf Db2 zugreifen kann. Gibt es keine Transaktion mehr, die diesen Thread benötigt, wird der Thread beendet; es sei denn, Sie haben angefordert, dass er für einen gewissen Zeitraum geschützt (beibehalten) werden soll. Ein geschützter Thread wird wiederverwendet, wenn eine andere Transaktion ihn innerhalb dieses Zeitraums anfordert; ist dies nicht der Fall, wird der Thread bei Ablauf des geschützten Zeitraums beendet.

Es gibt unterschiedliche Threadtypen und Sie können für jeden Threadtyp Grenzwerte festlegen, wie viele Threads dieses Typs gleichzeitig aktiv sein können. Dadurch wird verhindert, dass die allgemeine CICS-Db2-Verbindung mit Arbeit überlastet wird. Für Db2-Befehle, die mithilfe der Transaktion DSNK ausgegeben wurden, wird ein besonderer Threadtyp verwendet, und Sie können auch besondere Threads für CICS-Transaktionen definieren, die besondere Anforderungen haben, wie beispielsweise Transaktionen, die eine kurze Antwortzeit erfordern. Sie können definieren, was eine Transaktion unternehmen muss, wenn keine Threads des für sie benötigten Typs mehr verfügbar sind: die Transaktion kann warten, bis ein Thread des erforderlichen Typs verfügbar ist, sie kann einen Thread für allgemeine Zwecke (als Pool-Thread bezeichnet) verwenden oder sie kann abgebrochen werden.

Folgende Threadtypen werden von der CICS-Db2-Anschlussfunktion bereitgestellt:

Befehlsthreads

Befehlsthreads sind von der CICS-Db2-Anschlussfunktion für das Absetzen von Befehlen an Db2 reserviert, wobei die Transaktion DSNK verwendet wird. Sie werden nicht für Befehle verwendet, die für die CICS-Db2-Anschlussfunktion selbst wirksam werden, da diese Befehle nicht an Db2 übergeben werden. Wenn kein Befehlsthread verfügbar ist, wird für Befehle automatisch ein Überlauf an den Pool ausgeführt und diese Befehle verwenden einen Pool-Thread. Befehlsthreads sind in der DB2CONN-Definition im Abschnitt für Befehlsthreads definiert.

Einstiegsthreads

Einstiegsthreads sind in besonderer Weise definierte Threads, die für Transaktionen mit besonderen Anforderungen bestimmt sind, wie beispielsweise Transaktionen, die eine kurze Antwortzeit erfordern, oder Transaktionen mit besonderen Anforderungen hinsichtlich der Abrechnung. Sie können die CICS-Db2-Anschlussfunktion anweisen, Einstiegsthreads an bestimmte CICS-Transaktionen auszugeben. Sie definieren die verschiedenen Typen von Einstiegsthreads, die für unterschiedliche Transaktionen benötigt werden, und Sie können für die Anzahl jeder dieser Threadtypen einen Grenzwert festlegen. Wenn es für eine Transaktion zulässig ist, einen Einstiegsthread zu verwenden, jedoch kein passender Thread verfügbar ist, kann für die Transaktion ein Überlauf an den Pool ausgeführt werden und sie verwendet einen Pool-Thread, die Transaktion kann auf einen passenden Einstiegsthread warten oder sie kann abgebrochen werden, je nachdem, was Sie in der Definition für den Einstiegsthread ausgewählt haben.

Eine bestimmte Anzahl jeden Typs von Einstiegsthread kann geschützt werden. Wenn ein geschützter Einstiegsthread freigegeben wird, wird er nicht sofort beendet. Er wird für einen gewissen Zeitraum beibehalten und wenn eine andere CICS-Transaktion während dieses Zeitraums denselben Typ von Einstiegsthread benötigt, wird er wiederverwendet. Dadurch wird der Systemaufwand vermieden, der erforderlich ist, um für jede einzelne Transaktionen einen Thread zu erstellen und zu beenden. Ein un-

geschützter Einstiegsthread wird sofort beendet, es sei denn, zum Zeitpunkt der Freigabe des Threads ist eine wartende CICS-Transaktion vorhanden, die den Thread benutzen kann.

Einstiegsthreads werden mithilfe einer DB2ENTRY-Definition definiert.

Pool-Threads

Pool-Threads werden für alle Transaktionen und Befehle verwendet, die keinen Einstiegsthread oder einen Db2-Befehlsthread verwenden. Pool-Threads sind für Transaktionen mit geringem Volumen bestimmt sowie für Überlauftransaktionen, die keinen Einstiegsthread oder Db2-Befehlsthread abrufen konnten. Ein Pool-Thread wird sofort beendet, wenn keine CICS-Transaktion vorhanden ist, die ihn verwenden könnte. Pool-Threads werden in der DB2CONN-Definition im Abschnitt für Pool-Threads definiert.

Weitere detaillierte Informationen dazu, wie die unterschiedlichen Threadtypen erstellt, verwendet und beendet werden, finden Sie in [Threads erstellen, verwenden und beenden](#).

Jeder Thread wird unter einem Thread-Tasksteuerblock ausgeführt, der zu CICS gehört. CICS und Db2 verfügen beide über Verbindungssteuerblöcke, für die eine Programmverbindung zum Thread-Tasksteuerblock hergestellt ist. Sie verwenden diese Verbindungssteuerblöcke, um den Thread in Db2 zu verwalten und um untereinander Informationen zum Thread auszutauschen. Der Db2-Verbindungssteuerblock steuert den Thread innerhalb von Db2. Der CICS-Verbindungssteuerblock, der als CSUB bezeichnet wird, agiert als Verweis auf den Db2-Verbindungssteuerblock und enthält die Informationen, die für CICS zum Aufrufen des Db2-Verbindungssteuerblocks erforderlich sind, wenn der Thread benötigt wird. In Db2 werden diese Verbindungssteuerblöcke als 'Agentenstrukturen' bezeichnet.

Während CICS eine Verbindung zu einem Db2-Subsystem herstellt, wird der taskbezogene CICS-Db2-Benutzerexit (das Modul, das Db2 für die einzelnen Tasks aufruft) automatisch als offene API aktiviert, so dass die Umgebung für offene Transaktionen (OTE – Open Transaction Environment) verwendet werden kann.

Beide Typen von Tasksteuerblöcken, die die CICS-Db2-Anschlussfunktion zur Ausführung der Threads verwendet, d. h. die offenen Tasksteuerblöcke und die Subtask-Tasksteuerblöcke, werden in dieser Dokumentation als 'Thread-Tasksteuerblöcke' bezeichnet. In vielen Situationen führt die unterschiedliche Art der beiden Typen von Thread-Tasksteuerblöcken nicht zu Unterschieden im Betrieb der CICS-Db2-Verbindung. Dort, wo die unterschiedlichen Typen von Thread-Tasksteuerblöcken auch zu einem unterschiedlichen Verhalten der CICS-Db2-Verbindung führen, wird zwischen den beiden Typen unterschieden. Weitere technische Informationen zu Thread-Tasksteuerblöcken finden Sie in [Thread-TCBs \(Tasksteuerblöcke\)](#).

Thread-Tasksteuerblöcke in der Umgebung für offene Transaktionen

Wenn CICS mit DB2 Version 7 oder höher oder mit JDBC 2.0 oder höher verbunden ist, wird die Umgebung für offene Transaktionen verwendet. In dieser Umgebung verwendet die CICS-Db2-Anschlussfunktion als Thread-Tasksteuerblöcke offene Tasksteuerblöcke (im L8-Modus), anstatt extra erstellte Subtask-Tasksteuerblöcke zu verwenden.

Offene Tasksteuerblöcke greifen auf Db2-Ressourcen zu und führen außerdem noch andere Tasks aus. In der Umgebung für offene Transaktionen (OTE – Open Transaction Environment) wird der taskbezogene CICS-Db2-Benutzerexit in einem offenen Tasksteuerblock und nicht im CICS-Haupt-Tasksteuerblock ausgeführt. Wenn das Anwendungsprogramm, das die Db2-Anforderung ausgegeben hat, threadsicher ist, kann es auch im offenen Tasksteuerblock ausgeführt werden. (Weitere Informationen zu Anwendungsprogrammen in der Umgebung für offene Transaktionen finden Sie in [CICS-Db2-Anwendungen für die Verwendung von OTE durch threadsichere Programmierung aktivieren](#).)

Offene Tasksteuerblöcke sind nicht ständig einem CICS-Verbindungssteuerblock (CSUB) und einem Db2-Verbindungssteuerblock zugeordnet. Die CICS-Db2-Anschlussfunktion kann sie einem beliebigen verfügbaren CICS-Verbindungssteuerblock und Db2-Verbindungssteuerblock und damit jedem verfügbaren Thread zuordnen. Wenn der offene Tasksteuerblock die Verbindung zu Db2 nicht mehr benötigt, hebt er die Zuordnung zum Thread, zum CICS-Verbindungssteuerblock und zum Db2-Verbindungssteuerblock auf bzw. gibt diese frei. Der Thread, der CICS-Verbindungssteuerblock und der Db2-Verbindungssteuerblock können dann von einem anderen offenen Tasksteuerblock für die Tasks verwendet werden, die dieser Steuerblock in Db2 ausführen möchte.

Wenn der Thread vor der Wiederverwendung beendet wird, bleiben der CICS-Verbindungssteuerblock und der Db2-Verbindungssteuerblock im System verfügbar. Sind keine verfügbaren Threads vorhanden und ein offener Tasksteuerblock benötigt eine Verbindung zu Db2, kann die CICS-Db2-Anschlussfunktion den nicht verwendeten CICS-Verbindungssteuerblock und den Db2-Verbindungssteuerblock dem offenen Tasksteuerblock zuordnen und sie zur Ausführung eines neuen Threads in Db2 wiederverwenden.

In Abbildung 2 auf Seite 5 finden Sie eine Zusammenfassung zur Funktionsweise von Thread-Tasksteuerblöcken (TCB) in der Umgebung für offene Transaktionen.

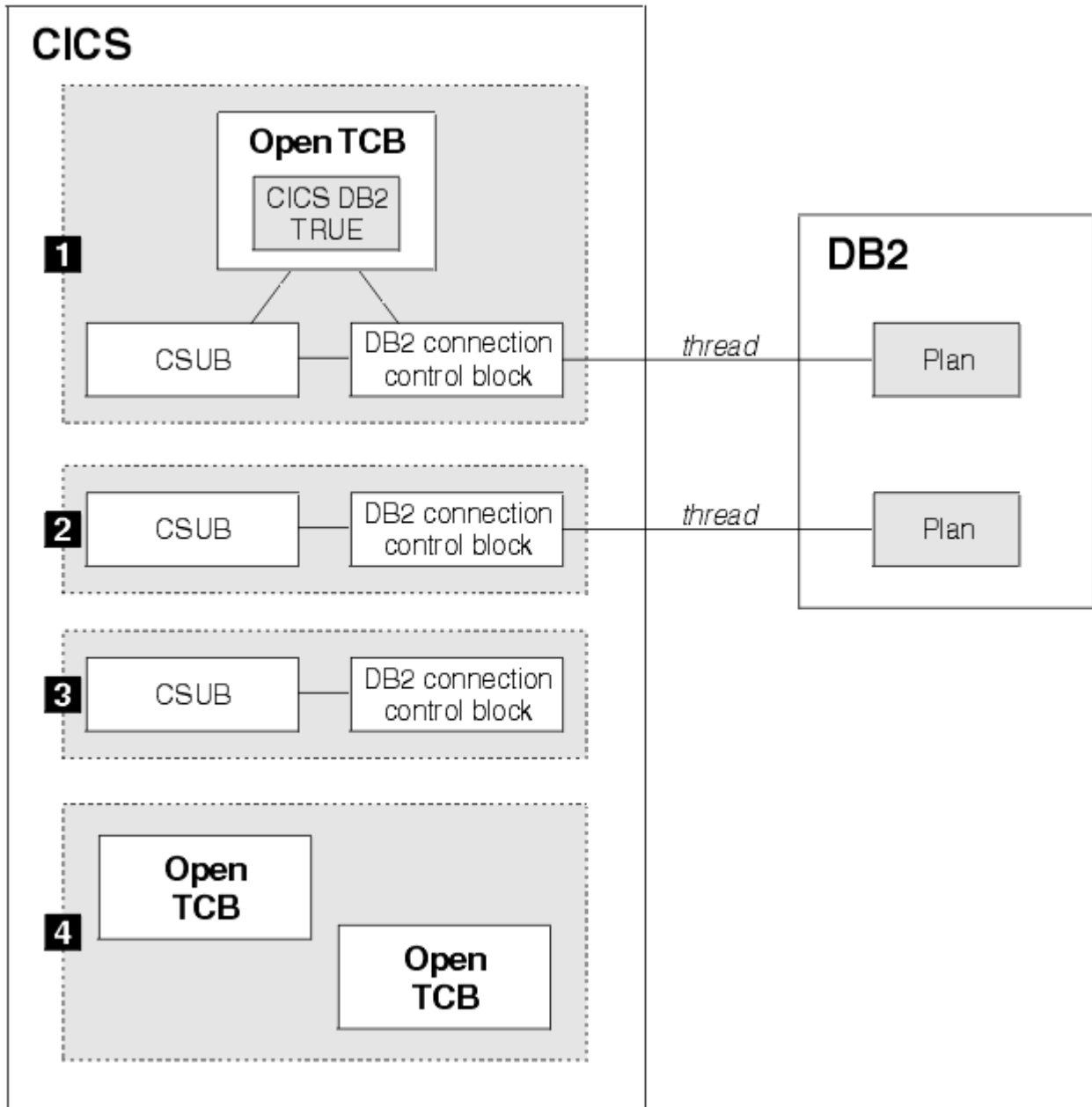


Abbildung 2. Thread-Tasksteuerblöcke in der Umgebung für offene Transaktionen

In Abbildung 2 auf Seite 5 zeigt Situation 1, wie CICS einen Thread für den Zugriff auf Db2 in der Umgebung für offene Transaktionen verwendet. Der taskbezogene CICS-Db2-Benutzerexit (CICS Db2 TRUE) wurde von der RMI-Schnittstelle (RMI – Resource Manager Interface) aufgerufen und ist in einem offenen Tasksteuerblock in Betrieb. Die CICS-Db2-Anschlussfunktion hat dem offenen Tasksteuerblock einen CICS-Verbindungssteuerblock und einen Db2-Verbindungssteuerblock zugeordnet. Der Db2-Verbindungssteuerblock verfügt über einen Thread in Db2. Der dem Thread zugeordnete Plan wird in Db2 gehalten.

Situation 2 zeigt einen Thread, der aktuell nicht in Verwendung aber geschützt ist. Der CICS-Verbindungssteuerblock und der Db2-Verbindungssteuerblock sind immer noch miteinander verbunden und sie haben einen Thread; es ist jedoch kein offener Tasksteuerblock an sie angeschlossen. Der Thread steht zur Wiederverwendung zur Verfügung.

In Situation 3 wird ein Aufbau gezeigt, der nach dem Beenden eines Threads verbleibt. Der CICS-Verbindungssteuerblock und der Db2-Verbindungssteuerblock stehen für die Wiederverwendung zur Verfügung. Sie benötigen einen neuen Thread.

In Situation 4 werden offene Tasksteuerblöcke gezeigt, die für die Wiederverwendung verfügbar sind. Die CICS-Db2-Anschlussfunktion kann diese offenen Tasksteuerblöcke verwenden und ihnen Kombinationen aus CICS-Verbindungssteuerblock und Db2-Verbindungssteuerblock zuordnen, um Threads in Db2 auszuführen.

Übersicht: Zugriff auf Db2 für CICS-Anwendungsprogramme aktivieren

Ein CICS-Anwendungsprogramm, das auf Db2 zugreift, muss in derselben Weise wie ein normales CICS-Anwendungsprogramm vorbereitet werden und es muss auch den Db2-Bindeprozess durchlaufen, mit dem ein Anwendungsplan erzeugt wird.

Informationen zu diesem Vorgang

Der Db2-Bindeprozess erzeugt einen Anwendungsplan (häufig nur als 'Plan' bezeichnet). Jeder Plan enthält in gebundener Form alle SQL-Anweisungen der Anwendungsprogramme, die diesen Plan verwenden; mit dem Plan wird diesen Anwendungsprogrammen ermöglicht, zur Laufzeit auf Db2-Daten zuzugreifen.

Die Pläne werden in Db2 beibehalten und jeder Thread in Db2 hat Bezug zu einem Plan. Der jeweilige Plan, den die einzelnen Threadtypen verwenden, ist in der DB2CONN-Definition (für Pool-Threads) oder in der DB2ENTRY-Definition des Threads (bei Einstiegsthreads) benannt. Der Plan für einen bestimmten Threadtyp muss die gebundene Form der SQL-Anweisungen aller Anwendungsprogramme enthalten, die diesen Threadtyp für den Zugriff auf Db2 verwenden. Sie können den Plan entweder explizit benennen oder Sie können einen dynamischen Planexit benennen. Ein dynamischer Planexit ist eine Routine, die festlegt, welcher Plan für die Transaktion verwendet werden soll, die einen Thread diesen Typs angefordert hat.

CICS-Anwendungsprogramm mit Zugriff auf Db2 vorbereiten

Ein CICS-Anwendungsprogramm, das auf Db2 zugreift, muss eine Reihe von Prozessen durchlaufen, damit ein Datenbankanforderungsmodul und ein Anwendungslademodul erstellt werden, bevor das Programm in den Bindeprozess eintritt.

Informationen zu diesem Vorgang

In [Abbildung 3 auf Seite 7](#) werden die Schritte gezeigt, die zum Vorbereiten eines CICS-Anwendungsprogramms mit Zugriff auf Db2 erforderlich sind.

Der erste Schritt besteht darin, das Programm durch den Db2-Vorcompiler laufen zu lassen. Der Db2-Vorcompiler erstellt ein Datenbankanforderungsmodul (DBRM – Database Request Module), das Informationen zu den einzelnen SQL-Anweisungen des Programms enthält.

Der zweite, dritte und vierte Schritt sind gleich den Schritten in einem normalen Prozess zum Vorbereiten einer beliebigen CICS-Anwendung, unabhängig davon, ob die Anwendung auf Db2 zugreift. Der zweite Schritt besteht darin, das Programm durch den CICS-Umsetzer für die Befehlssprache laufen zu lassen. Im dritten Schritt wird das Programm kompiliert oder assembliert. Im vierten Schritt wird für das Programm eine Programmverbindung zu den erforderlichen Schnittstellen (einschließlich zum CICS-Db2-Sprachschnittstellenmodul DSNCLI) hergestellt. Das Endprodukt der Schritte 2, 3 und 4 ist ein Anwendungslademodul, das die Ausführung des Programms ermöglicht. Weitere Informationen zu diesen Schritten finden Sie in [CICS-Db2-Programmerstellung](#).

Es ist ein gesonderter Schritt erforderlich, um dem Programm die Verwendung der Informationen des in Schritt 1 erstellten Datenbankanforderungsmoduls zu ermöglichen. Dieser fünfte Schritt ist der Bindeprozess. Für den Bindeprozess ist Db2 erforderlich und er erstellt mithilfe des Datenbankanforderungsmoduls

duls einen Anwendungsplan, der dem Programm den Zugriff auf Db2-Daten ermöglicht. In „[Bindeprozess](#)“ auf Seite 7 finden Sie eine Erläuterung des Bindeprozesses.

Wenn Sie einen der Language Environment-konformen Compiler für COBOL und PL/I verwenden, können Sie einige dieser Schritte in einer einzigen Task zusammenfassen, da der CICS-Umsetzer für die Befehlssprache und (in Abhängigkeit von Ihrer Db2-Version) ein Koprozessor für SQL-Anweisungen in diese Compiler integriert ist. Weitere Informationen finden Sie in [CICS-Db2-Programmerstellung](#).

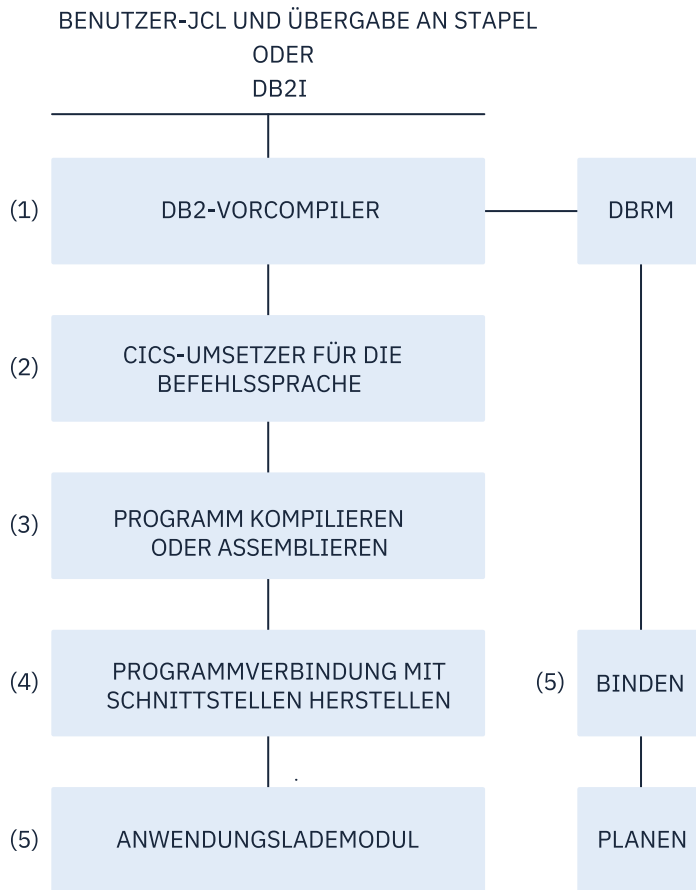


Abbildung 3. Schritte für die Vorbereitung eines CICS-Anwendungsprogramms mit Zugriff auf Db2

Bindeprozess

Bevor eine CICS-Anwendung, die auf Db2 zugreift, ausgeführt werden kann, muss sie den Bindeprozess durchlaufen.

Für den Bindeprozess ist Folgendes erforderlich:

- Db2
- Das Datenbankanforderungsmodul (DBRM – Database Request Module), das vom Db2-Vorcompiler für jedes Programm in der Anwendung erstellt wird.

Das Datenbankanforderungsmodul enthält die SQL-Anweisungen, die der Db2-Vorcompiler aus dem Anwendungsprogramm extrahiert hat. Im Bindeprozess werden die SQL-Anweisungen in den Datenbankanforderungsmodulen in eine betriebsbereite ('gebundene') Form gebracht, indem sie in die Steuerstrukturen umgesetzt werden, die Db2 bei der Ausführung von SQL-Anweisungen verwendet. Das sich ergebende Material kann in ein Paket zusammengefasst oder es kann direkt in einen Anwendungsplan platziert werden (siehe „[Pläne, Pakete und dynamische Planexits](#)“ auf Seite 8). Der gesamte Prozess wird als 'Binden' des Datenbankanforderungsmoduls bezeichnet. Weitere Informationen zum Binden finden Sie in [Programmierung für Db2 for z/OS in der Produktdokumentation zu Db2 for z/OS](#). Weitere Informationen zu den Optionen, die Sie für den Bindeprozess in der CICS-Db2-Umgebung auswählen sollten, finden Sie in [Bindeoptionen und -aspekte bei Programmen](#).

Pläne, Pakete und dynamische Planexits

Ein Anwendungsplan ermöglicht Anwendungsprogrammen, während der Ausführungszeit auf Db2-Daten zuzugreifen. Der Plan enthält die gebundene, operative Form der SQL-Anweisungen aus den Datenbankankforderungsmodulen, die aus den Anwendungsprogrammen erstellt wurden. Er setzt den gesamten Anwendungsprozess auch in Beziehung zur lokalen Instanz von Db2, in der er verwendet wird.

Die operativen SQL-Anweisungen aus einem Datenbankankforderungsmodul können direkt in den Plan gesetzt werden; in diesen Fall spricht man davon, dass das Datenbankankforderungsmodul in einen Plan gebunden wird. Alternativ können Sie ein Datenbankankforderungsmodul (mithilfe des Befehls BIND PACKAGE) in ein Paket binden, das die operativen SQL-Anweisungen eines einzigen Datenbankankforderungsmoduls enthält. Sie können zusammengehörige Pakete in Sammlungen zusammenfassen. Sie können anschließend den Paketnamen oder den Namen der Sammlung in eine Liste mit Paketen einschließen, und die Liste der Pakete in den Plan einbinden. Ein einzelner Plan kann sowohl eine Paketliste als auch direkt in den Plan gebundene Datenbankankforderungsmodule enthalten. Sie können einen Plan mithilfe der Datenbankankforderungsmodule aus einer einzigen CICS-Anwendung erstellen, oder Sie können die Datenbankankforderungsmodule mehrerer Anwendungen verwenden, um einen einzigen Plan zu erstellen. In [Beziehung zwischen CICS-Anwendungen und Db2-Plänen und -Paketen gestalten](#) werden die Vorteile und Nachteile verschiedener Designs Ihrer Pläne und Pakete erörtert.

Anwendungspläne müssen nicht nur erstellt, sondern auch verwaltet werden. Wenn die SQL-Anweisungen in mindestens einem der Anwendungsprogramme, die einen Plan verwenden, geändert werden, müssen Sie die Datenbankankforderungsmodule für die geänderten Anwendungsprogramme erneut erstellen. Wenn Sie die alten Versionen dieser Datenbankankforderungsmodule direkt in Ihren Plan gebunden haben, müssen Sie alle Datenbankankforderungsmodule identifizieren, die direkt in diesen Plan gebunden wurden (und zwar für die geänderten Programme und für alle nicht geänderten Programme) und Sie müssen sie erneut in den Plan binden. Während Sie die Datenbankankforderungsmodule in den Plan binden, können Anwendungen den Plan nicht verwenden, um auf Db2 zuzugreifen. Wenn Sie jedoch die alten Versionen der Datenbankankforderungsmodule für die geänderten Anwendungsprogramme in Pakete gebunden haben und anschließend die Namen der Pakete (oder der Sammlungen, die die Pakete enthalten) in der Paketliste in den Plan einschließen, so müssen Sie keine weiteren Pakete oder direkt gebundenen Datenbankankforderungsmodule erneut in den Plan binden. Sie binden die neuen Versionen der Datenbankankforderungsmodule für die geänderten Anwendungsprogramme in Pakete, die dieselben Namen aufweisen wie die alten Versionen. Sie müssen den Plan nicht erneut binden – er lokalisiert die neuen Versionen der Pakete. Während Sie die Pakete ändern, können Anwendungsprogramme weiterhin die übrigen Pakete und direkt eingebundenen Datenbankankforderungsmodule im Plan verwenden. Weitere Informationen zur Verwaltung von Plänen finden Sie in [Zu bindende Elemente nach einer Programmänderung](#).

Jeder Thread in Db2 steht in Beziehung zu einem Plan; weitere Informationen zu Threads finden Sie in [„Übersicht: Funktionsweise von Threads“](#) auf Seite 3. Der Plan, der von den einzelnen Threadtypen verwendet wird, ist in der DB2CONN-Definition (für Pool-Threads) oder in der DB2ENTRY-Definition des Threads (für Einstiegsthreads) angegeben. Wenn CICS die Verwendung eines Threads anfordert, damit eine Anwendung auf Db2 zugreifen kann, teilt es Db2 den Namen des Plans mit, der diesem Threadtyp zugeordnet ist, und Db2 sucht den Plan. Die Definition für den einzelnen Threadtyp kann entweder den Namen eines bestimmten Plans oder es kann einen dynamischen Planexit nennen, eine Routine, die festlegt, welcher Plan für die Transaktion verwendet werden soll, die den Thread angefordert hat.

Wenn in der Definition des Pool-Threads oder Einstiegsthreads ein bestimmter Plan genannt wird, müssen alle Transaktionen, die diesen Threadtyp verwenden, diesen Plan verwenden. Die Transaktionen, die einen Einstiegsthread-Typ verwenden können, sind in der DB2ENTRY- und der DB2TRAN-Definition für den Thread angegeben. Wenn die DB2ENTRY-Definition für den Thread einen bestimmten Plan nennt, müssen die Datenbankankforderungsmodule aller Anwendungsprogramme, die unter allen diesen Transaktions-IDs ausgeführt werden können, in denselben Plan gebunden oder in Pakete gebunden werden, die anschließend im selben Plan aufgelistet sind. Wenn die Datenbankankforderungsmodule eines der Anwendungsprogramme, die unter diesen Transaktions-IDs ausgeführt werden, direkt in den Plan gebunden werden und Sie die SQL-Anweisungen in einem dieser Anwendungsprogramme ändern, kann auf den gesamten Plan nicht mehr zugegriffen werden, während Sie alle direkt gebundenen Datenbankankforderungsmodule wieder in den Plan binden. Dies bedeutet, dass keine Transaktion diesen Typ von Einstiegsthread verwenden kann, während Sie eine Wartung für den Plan ausführen. Pool-Threads können von *allen* Ihren CICS-Anwendungen verwendet werden, die auf Db2 zugreifen; wenn also die DB2CONN-

Definition einen bestimmten Plan für die Pool-Threads angibt, muss der Plan unter Verwendung der Datenbankankforderungsmodule aller dieser Anwendungen vorbereitet und gewartet werden. Wenn einige der Datenbankankforderungsmodule direkt in den Plan gebunden wurden, kann während der Wartung kein Zugriff auf den Plan erfolgen und keine Transaktion kann einen Pool-Thread verwenden.

Es gibt zwei Möglichkeiten, wie vermieden werden kann, dass Threads während der Wartung von Plänen nicht verfügbar werden. Die beste Lösung liegt darin zu vermeiden, dass Datenbankankforderungsmodule direkt in Pläne gebunden werden; verwenden Sie stattdessen Pakete. Wenn Sie jedes jedes einzelne Datenbankankforderungsmodule als Paket binden und diese in Paketlisten in Pläne einschließen, kann auf die Pläne auch zugegriffen werden, während Sie für einzelne Pakete eine Wartung ausführen. Während Sie für ein bestimmtes Programm Wartungsarbeiten ausführen, bleiben die Pool-Threads oder Einstiegsthreads verfügbar, die sich auf Pläne beziehen, bei denen das betreffende Programm beteiligt ist, weil auf die Pläne weiterhin zugegriffen werden kann. Dies bedeutet, dass Sie für jeden Thread sicher einen bestimmten Plan nennen können. Wenn Sie beginnen wollen, Pakete zu verwenden, finden Sie in [Db2-Pakete verwenden](#) und [Programmierung für Db2 for z/OS](#) in der Produktdokumentation zu Db2 for z/OS Informationen dazu, wie Pakete implementiert werden.

Als alternative Lösung, die entwickelt wurde, bevor Pakete in Db2 verfügbar wurden, können Sie einen dynamischen Planexit verwenden. Einen dynamischen Planexit zu verwenden bedeutet, dass Sie nicht für jeden einzelnen Threadtyp einen bestimmten Plan benennen müssen; wenn also während der Wartung auf einen bestimmten Plan nicht zugegriffen werden kann, stehen die Threads weiterhin zur Verfügung. Zum Implementieren dieser Lösung erstellen Sie für Ihre CICS-Anwendungen viele kleine Pläne, von denen jeder die Datenbankankforderungsmodule aus einigen eng zusammengehörigen Programmen enthält. Anstatt dann für jeden Threadtyp im Attribut PLAN der DB2CONN- oder DB2ENTRY-Definition einen Planamen anzugeben, geben Sie im Attribut PLANEXITNAME ein Exitprogramm an. Wenn ein Anwendungsprogramm seine erste SQL-Anweisung ausgibt und ein bestimmter Typ von Thread angefordert wird, wählt das von Ihnen in der Threaddefinition angegebene Exitprogramm den Plan aus, der für das betreffende Anwendungsprogramm zu verwenden ist. Wenn während der Wartung nicht auf einen bestimmten Plan zugegriffen werden kann, können die Anwendungsprogramme, die diesen Plan erfordern, einen Thread nicht verwenden, andere Anwendungsprogramme können jedoch denselben Threadtyp mit ihren eigenen Plänen verwenden. Beachten Sie jedoch, dass sobald eine bestimmte Threadtyp-Instanz erstellt wurde, die von einem Anwendungsprogramm verwendet werden soll, diese Threadinstanz dem Plan zugeordnet wird, der vom dynamischen Planexit ausgewählt wurde. Damit ein anderes Anwendungsprogramm den Thread wiederverwenden kann, muss es denselben Plan verwenden. Wenn der dynamische Planexit einen anderen Plan für das Anwendungsprogramm auswählt, muss er einen anderen Thread mit dem richtigen Plan suchen oder erstellen. Dies schränkt die Möglichkeiten für die Wiederverwendung von Threads ein. Weitere Informationen zu dynamischen Planexits finden Sie in [Dynamische Planexits](#).

Kapitel 2. CICS-Db2-Verbindung definieren

Die Struktur und Leistung Ihrer CICS-Db2-Verbindung wird festgelegt, wenn Sie die DB2CONN-, DB2ENTRY- und DB2TRAN-Objekte definieren.

Informationen zu diesem Vorgang

Die DB2CONN-, DB2ENTRY- und DB2TRAN-Objekte beschreiben die globalen Attribute der CICS-Db2-Verbindung, die Beziehungen zwischen CICS-Transaktionen und Db2-Ressourcen (einschließlich der Anwendungspläne und Befehlsprozessoren), die Attribute der einzelnen Threadtypen sowie den jeweiligen Threadtyp, den die einzelnen Transaktionen verwenden können.

Übersicht: Vorgehensweise beim Definieren der CICS-Db2-Verbindung

Beim Definieren der CICS-Db2-Verbindung sind drei unterschiedliche CICS-Ressourcendefinitionen beteiligt: DB2CONN (die Db2-Verbindungsdefinition), DB2ENTRY (die Db2-Eintragsdefinition) und DB2TRAN (die Db2-Transaktionsdefinition).

Die CICS-Db2-Verbindung besteht aus einer allgemeinen Verbindung zwischen CICS und Db2 sowie aus einzelnen Verbindungen, die als Threads bezeichnet werden. Sie können diese Ressourcendefinitionen verwenden, um die Attribute der allgemeinen Verbindung sowie die Attribute der unterschiedlichen Typen von Threads zu definieren. Wenn Sie für wichtige Transaktionen gesondert Einstiegsthreads definiert haben, können Sie der CICS-Db2-Anschlussfunktion mitteilen, welche CICS-Transaktionen diese Threads verwenden können.

Es gibt mehrere Möglichkeiten, wie DB2CONN-, DB2ENTRY- und DB2TRAN-Objekte für CICS definiert werden können. Sie können Folgendes ausführen:

- Definieren und installieren Sie DB2CONN-, DB2ENTRY- und DB2TRAN-Objekte mithilfe von IBM CICS Explorer.
- Verwenden Sie CICSplex SM zum Definieren der Objekte.
- Definieren und installieren Sie DB2CONN-, DB2ENTRY- und DB2TRAN-Objekte mithilfe von RDO (RDO – Resource Definition Online, Onlineresourcendefinition).
- Definieren Sie die Objekte unter Verwendung des CICS-Stapeldienstprogramms DFHCSDUP.

Der Gültigkeitsbereich der einzelnen Objekte lautet wie folgt:

DB2CONN

DB2CONN ist die Hauptdefinition der CICS-Db2-Verbindung. Sie müssen eine DB2CONN-Ressourcendefinition installieren, bevor Sie die CICS-Db2-Verbindung starten können. Sie verwenden die DB2CONN-Definition zum Definieren der folgenden Elemente:

- Die Attribute der allgemeinen CICS-Db2-Verbindung:
 - Das Db2-Subsystem, zu dem CICS eine Verbindung herstellt, oder die Db2-Subsystemgruppe mit gemeinsamer Datennutzung, aus der Db2 ein aktives Mitglied auswählt, zu dem CICS eine Verbindung herstellt.
 - Ob CICS den Verbindungsaufbau beim Fehlschlagen der Verbindung zu Db2 automatisch wiederholt und ob CICS eine erneute Verbindung zu demselben Db2-Subsystem herstellt, falls Sie die Einrichtung zum Anhängen von Gruppen verwenden.
 - Eine Universal-Berechtigungs-ID für die CICS-Db2-Anschlussfunktion zur Anmeldung für einen Thread, falls keine andere Berechtigung für den Thread erforderlich ist.
 - Der Grenzwert für die Gesamtzahl von Tasksteuerblöcken, die CICS zum Ausführen von Threads in Db2 gleichzeitig verwenden kann.
 - Wie lange geschützte Threads vor der Beendigung beibehalten werden.

- Der Grenzwert für die Anzahl, wie häufig ein Thread wiederverwendet werden kann, bevor er beendet wird.
- Wie Fehlernachrichten an CICS übertragen werden und was Transaktionen ausführen sollen, wenn ihr Thread fehlschlägt.
- Wohin Nachrichten und Statistikdaten gesendet werden.
- Die Attribute von Befehlsthreads, bei denen es sich um die besonderen Threads handelt, die von der Transaktion DSNB zum Absetzen von Db2-Befehlen verwendet werden:
 - Der Grenzwert für die Anzahl von Befehlsthreads, die CICS gleichzeitig verwenden kann.
 - Welchen Typ von Berechtigungs-IDs Db2 prüft, wenn ein Befehlsthread angefordert wird (beispielsweise die ID des Benutzers, die ID der Transaktion und die Universal-ID der CICS-Db2-Anschlussfunktion).
- Die Attribute von Pool-Threads, bei denen es sich um die Universalthreads handelt, die verwendet werden, wenn Transaktionen keinen bestimmten Befehlsthread oder Einstiegsthread benötigen oder wenn keine bestimmten Threads übrig sind, die eine Transaktion verwenden könnte:
 - Welchen Typ von Berechtigungs-IDs Db2 prüft, wenn ein Pool-Thread angefordert wird (beispielsweise die ID des Benutzers, die ID der Transaktion, die Universal-ID der CICS-Db2-Anschlussfunktion).
 - Wie hoch die Priorität der Thread-Tasksteuerblöcke in Relation zum CICS-Haupt-Tasksteuerblock ist.
 - Der Grenzwert für die Anzahl von Pool-Threads, die CICS gleichzeitig verwenden kann.
 - Ob eine Transaktion, die keinen Pool-Thread abrufen kann, auf einen Thread warten oder ob sie abgebrochen werden soll.
 - Welcher Anwendungsplan oder dynamische Planexit für die Pool-Threads verwendet werden soll (siehe [Pläne](#), [Pakete](#) und [dynamische Planexits](#)).
 - An welchem Punkt während der Verwendung des Threads durch die Transaktion Db2-Abrechnungsdatensätze erstellt werden.
 - Falls ein Deadlock aufgetreten ist: Ob die Änderungen, die durch die den Thread verwendende Transaktion vorgenommen wurden, rückgängig gemacht werden sollen.

In einem CICS-System kann gleichzeitig nur eine einzige DB2CONN-Definition installiert sein und alle DB2ENTRY- und DB2TRAN-Definitionen, die Sie im System installieren, sind der DB2CONN-Definition zugeordnet. Wenn Sie eine DB2CONN-Definition löschen, werden alle DB2ENTRY- und DB2TRAN-Definitionen ebenfalls gelöscht. Sie können eine DB2CONN-Definition nicht löschen und eine andere installieren, während CICS mit Db2 verbunden ist.

Verwechseln Sie nicht die Ressourcendefinition DB2CONN mit dem Systeminitialisierungsparameter DB2CONN, mit dem angegeben wird, ob Sie möchten, dass CICS die Verbindung zu Db2 während der Initialisierung automatisch herstellt.

Sie können die CICS-Db2-Verbindung starten, wenn nur eine DB2CONN-Definition installiert ist; für den Verbindungsaufbau benötigen Sie keine DB2ENTRY- und DB2TRAN-Definitionen. Wenn Sie dies ausführen, gibt es keine besonderen Threads für wichtige Transaktionen (Einstiegsthreads). Alle Transaktionen verwenden Universalthreads aus dem Pool und die wichtigsten Transaktionen müssen genau so lange warten wie die am wenigsten wichtigen Transaktionen, bis ihre jeweilige Verbindung zu Db2 abgerufen wird. Um sicherzustellen, dass Ihre wichtigen Transaktionen eine höhere Priorität erhalten, erstellen Sie für sie DB2ENTRY- und, falls erforderlich, DB2TRAN-Definitionen.

DB2ENTRY

Um verschiedene Typen von Einstiegsthreads zu definieren, können Sie viele DB2ENTRY-Definitionen erstellen. Die Einstiegsthreads können von den von Ihnen angegebenen Transaktionen für den bevorzugten Zugriff (oder besonderen Zugriff) auf Db2-Ressourcen verwendet werden. Tatsächlich reservieren Sie eine bestimmte Anzahl von Threads, die nur von diesen Transaktionen verwendet werden können. Sie können auch eine gewisse Anzahl jeden Typs von Einstiegsthread schützen, wodurch die Leistung für stark verwendete Transaktionen verbessert wird.

In einer DB2ENTRY-Definition können Sie eine bestimmte Transaktion oder (durch Verwendung eines Platzhalters) eine Gruppe von Transaktionen angeben, die der DB2ENTRY-Definition zugeordnet sind und die den von ihr definierten Typ von Einstiegsthread verwenden können. Wenn diese Transaktionen einen Thread anfordern, liefert die CICS-Db2-Anschlussfunktion ihnen diesen Typ von Einstiegsthread, falls ein solcher verfügbar ist. Wenn Sie möchten, dass andere Transaktionen denselben Typ von Einstiegsthread verwenden, können Sie für diese Transaktionen eine DB2TRAN-Definition erstellen, in der für CICS angegeben ist, dass die Transaktionen einem bestimmten DB2ENTRY-Objekt zugeordnet sind.

Sie verwenden die einzelnen DB2ENTRY-Definitionen zum Definieren der folgenden Elemente:

- Eine Transaktion oder (durch Verwendung eines Platzhalters im Namen) eine Gruppe von Transaktionen, die diesen Typ von Einstiegsthread verwenden kann.
- Die Attribute dieses Typs von Einstiegsthread:
 - Welchen Typ von Berechtigungs-IDs Db2 prüft, wenn dieser Typ von Einstiegsthread angefordert wird (beispielsweise die ID des Benutzers, die ID der Transaktion und die Universal-ID der CICS-Db2-Anschlussfunktion).
 - Wie hoch die Priorität der Thread-Tasksteuerblöcke in Relation zum CICS-Haupt-Tasksteuerblock ist.
 - Der Grenzwert der Anzahl diesen Typs von Einstiegsthread, die CICS gleichzeitig verwenden kann.
 - Die Anzahl diesen Typs von Einstiegsthread, die für einen gewissen Zeitraum geschützt ist, während sie nicht in Verwendung ist.
 - Ob eine dem DB2ENTRY-Objekt zugeordnete Transaktion, die diesen Typ von Einstiegsthread nicht abrufen kann, auf diesen Typ von Einstiegsthread warten soll, durch einen Überlauf einen Pool-Thread verwenden oder ob sie abgebrochen werden soll.
 - Welcher Anwendungsplan oder dynamischer Planexit für diesen Typ von Einstiegsthread verwendet werden soll (siehe [Pläne](#), [Pakete](#) und [dynamische Planexits](#)).
 - An welchem Punkt während der Verwendung des Threads durch die Transaktion Db2-Abrechnungsdatensätze erstellt werden.
 - Falls ein Deadlock aufgetreten ist: Ob die Änderungen, die durch die den Thread verwendende Transaktion vorgenommen wurden, rückgängig gemacht werden sollen.

Sie können ein DB2ENTRY-Objekt nicht löschen, während Transaktionen diesen Typ von Einstiegsthread verwenden. Sie müssen es zuerst inaktivieren, um die Aktivität für das DB2ENTRY-Objekt in den Wartemodus zu versetzen, und es anschließend löschen. Wenn Sie ein DB2ENTRY-Objekt löschen, sind die ihm zugeordneten DB2TRAN-Objekte 'verwaist' und die in ihnen aufgelisteten Transaktionen verwenden Pool-Threads.

Wenn Sie CECI, die Transaktion des Befehlsinterpreters verwenden, um die Syntax von CICS-Befehlen in einem Anwendungsprogramm zu testen, müssen Sie beachten, dass CICS in dieser Situation die Transaktions-ID CECI verwendet, um eine Überprüfung auf übereinstimmende DB2ENTRY-Definitionen auszuführen. Wenn Sie für Ihr Anwendungsprogramm ein DB2ENTRY-Objekt konfiguriert haben und es replizieren wollen, während Sie CECI verwenden, müssen Sie ein DB2ENTRY-Objekt mit denselben Eigenschaften für die Transaktion CECI konfigurieren.

DB2TRAN

Sie können DB2TRAN-Definitionen verwenden, um einem bestimmten DB2ENTRY-Objekt zusätzliche Transaktionen zuzuordnen. Die Transaktionen verwenden dann diesen Typ von Einstiegsthread. Für eine bestimmte Transaktion kann jeweils nur eine DB2TRAN-Definition installiert werden. In jeder DB2TRAN-Definition geben Sie folgende Elemente an:

- Der Name eines DB2ENTRY-Objekts.
- Eine Transaktion oder (durch Verwendung eines Platzhalters im Namen) eine Gruppe von Transaktionen, die dem bestimmten DB2ENTRY-Objekt zugeordnet sind und diesen Typ von Einstiegsthread verwenden können.

Wenn diese Transaktionen einen Thread anfordern, erkennt die CICS-Db2-Anschlussfunktion, dass sie dem bestimmten DB2ENTRY-Objekt zugeordnet sind und behandelt sie wie die Transaktion bzw. Transaktionen, die in der DB2ENTRY-Definition selbst genannt ist/sind.

Einrichtung zum Anhängen von Db2-Gruppen verwenden.

Wenn Sie die Verbindung zwischen CICS und Db2 definieren, können Sie auswählen, dass CICS eine Verbindung zu einem bestimmten Db2-Subsystem herstellt.

Informationen zu diesem Vorgang

Sie können den Namen dieses Db2-Subsystems mithilfe des Attributs DB2ID der DB2CONN-Definition angeben. Wenn Sie jedoch über mehrere Db2-Subsysteme verfügen, die Db2 Version 7 oder höher verwenden, sollten Sie die Einrichtung zum Anhängen von Db2-Gruppen verwenden, um es CICS zu ermöglichen, eine Verbindung zu einem beliebigen Ihrer Subsysteme und nicht nur zu einem namentlich genannten Subsystem herzustellen.

Das Anhängen von Gruppen ist eine Db2-Einrichtung, mit der es CICS ermöglicht wird, eine Verbindung zu einem beliebigen Member einer Db2-Subsystemgruppe mit gemeinsamer Datennutzung herzustellen, anstatt zu einem bestimmten Db2-Subsystem. Die Einrichtung zum Anhängen von Gruppen wählt ein beliebiges Member der im lokalen MVS-Image aktiven Gruppe aus, um eine Verbindung zur CICS-Instanz herzustellen (in anderen MVS-Images aktive Member sind nicht auswählbar).

Gehen Sie wie folgt vor, um die Einrichtung zum Anhängen von Gruppen zu verwenden:

Vorgehensweise

1. Aktivieren Sie die Einrichtung zum Anhängen von Gruppen mithilfe des Attributs DB2GROUPID der DB2CONN-Definition. Geben Sie den Gruppenanschlussnamen für die Gruppe der Db2-Subsysteme an, anstatt mithilfe des Attributs DB2ID die ID eines einzelnen Db2-Subsystems anzugeben.
Mit Db2 Version 10 können Sie auch einen Anschlussnamen für Untergruppen angeben, um eine Untergruppe der Gruppe zu bezeichnen. 'Anhängen von Gruppen' bedeutet, dass Sie für mehrere geklonnte Anwendungsverwaltungsregionen (AOR – Application-owning Region) eine gemeinsame DB2CONN-Definition verwenden und einen Wert für DB2GROUPID mit dem Ergebnis angeben können, dass CICS zu einem beliebigen aktiven Member dieser Gruppe oder Untergruppe mit gemeinsamer Datennutzung eine Verbindung herstellt. Informationen zur Definition und Installation einer DB2CONN-Definition finden Sie in DB2CONN-Ressourcen.
2. Geben Sie das Attribut RESYNCMEMBER der Definition DB2CONN an, um jegliche unbestätigten Arbeitseinheiten zu beheben, wenn die Verbindung zwischen CICS und Db2 unterbrochen wird.
Wenn die Verbindung unterbrochen wurde, stellt CICS möglicherweise nicht eine erneute Verbindung zum selben Db2-Subsystem her; es wird möglicherweise ein anderes Member der Db2-Subsystemgruppe mit gemeinsamer Datennutzung ausgewählt. Werden also vom ersten Db2-Subsystem, zu dem CICS eine Verbindung hergestellt hat, unbestätigte Arbeitseinheiten gehalten, können diese nicht behoben werden. Informationen zum Attribut RESYNCMEMBER und zu seiner Festlegung finden Sie in [Unbestätigte Arbeitseinheiten auflösen](#).
3. Ermitteln Sie mithilfe des Befehls **INQUIRE DB2CONN DB2ID()** nach dem Herstellen der Verbindung, welches Member der Gruppe mit gemeinsamer Datennutzung für die aktuelle Verbindung ausgewählt wurde.
4. Wenn Sie möchten, dass CICS eine Verbindung zu einem bestimmten Db2-Subsystem herstellt, können Sie das Anhängen von Gruppen außer Kraft setzen.
Wenn Sie beispielsweise möchten, dass CICS eine Verbindung zu einem Db2-Subsystem mit der ID 'xyz' herstellt, können Sie die DB2ID mithilfe von Folgendem angeben:
 - Befehl **SET DB2CONN DB2ID(xyz)**
 - Befehl **DSNC STRT xyz** (siehe [DSNC STRT](#))

Mit jeder der vorstehenden Methoden wird das Anhängen von Gruppen durch Festlegen einer DB2ID in der installierten DB2CONN-Definition außer Kraft gesetzt.

Ergebnisse

Wenn in den vorherigen Schritten eine DB2ID angegeben wird, bewirkt dies, dass das Attribut DB2GROUPID der installierten DB2CONN-Definition mit Leerzeichen überschrieben wird. Wenn Sie zur Verwendung des Anhängens von Gruppen zurückkehren möchten, legen Sie das Attribut DB2GROUPID erneut mithilfe eines **SET DB2CONN DB2GROUPID()**-Befehls fest. Das Anhängen von Gruppen wird jedoch nicht durch Angabe einer DB2ID im Systeminitialisierungsparameter **INITPARM**=(DFHD2INI=*db2id*) außer Kraft gesetzt. Wenn in der DB2CONN-Definition eine DB2GROUPID angegeben ist, wird die Einstellung **INITPARM** ignoriert. Weitere Informationen zu diesem Parameter finden Sie in [Systeminitialisierungsparameter INITPARM](#).

TCBLIMIT und der Systeminitialisierungsparameter MAXOPENTCBS

So wie sich die von Ihnen definierten DB2CONN-, DB2ENTRY- und DB2TRAN-Objekte auf die CICS-Db2-Verbindung auswirken, so wirkt sich auch der Systeminitialisierungsparameter MAXOPENTCBS auf diese aus. Mit MAXOPENTCBS wird die Gesamtzahl der offenen L8- und L9-Tasksteuerblöcke gesteuert, die die CICS-Region gleichzeitig in Betrieb haben kann.

MAXOPENTCBS ist relevant, wenn CICS mit Db2 verbunden ist, weil CICS offene L8- und L9-Tasksteuerblöcke verwendet, um Threads in Db2 auszuführen. Offene L8- und L9-Tasksteuerblöcke sind für die Verwendung durch taskbezogene Benutzerexits reserviert, die durch die Option OPENAPI aktiviert wurden. Zu diesen gehört der taskbezogene CICS-Db2-Benutzerexit.

In der Umgebung für offene Transaktionen steuert das Attribut TCBLIMIT der DB2CONN-Definition die Anzahl der offenen L8- und L9-Tasksteuerblöcke, die vom taskbezogenen CICS-Db2-Benutzerexit zum Ausführen von Threads in Db2 verwendet werden können. Ist der durch TCBLIMIT festgelegte Grenzwert erreicht, kann der taskbezogene CICS-Db2-Benutzerexit einen Tasksteuerblock aus dem von MAXOPENTCBS gesteuerten Pool abrufen, aber der Benutzerexit muss warten, bevor er den Tasksteuerblock zum Ausführen eines Threads in Db2 verwenden kann. Wenn eine weitere Task ihren L8- oder L9-Tasksteuerblock nicht mehr zum Ausführen eines Threads in Db2 verwendet, und die zur Ausführung von Threads verwendete Anzahl von Tasksteuerblöcken unter die durch TCBLIMIT festgelegte Anzahl fällt, wird für die wartende Task zugelassen, dass sie ihren eigenen zugehörigen L8- oder L9-Tasksteuerblock zum Ausführen eines Threads in Db2 verwendet. Wenn jedoch der durch MAXOPENTCBS festgelegte Wert erreicht ist, sind keine weiteren offenen L8- und L9-Tasksteuerblöcke in der CICS-Region zulässig und der taskbezogene CICS-Db2-Benutzerexit kann nicht einmal einen L8- oder L9-Tasksteuerblock zur eigenen Verwendung abrufen. Es muss gewartet werden, bis ein L8- oder L9-Tasksteuerblock (L9 wird von OPENAPI-Programmen verwendet) von einer anderen Task freigegeben und an den von MAXOPENTCBS gesteuerten Pool zurückgegeben wird. Wird ein L8-Tasksteuerblock freigegeben, kann die Task diesen Tasksteuerblock verwenden; wurde jedoch ein L9-Tasksteuerblock freigegeben, so 'stiehlt' CICS diesen Tasksteuerblock, was bedeutet, dass die Zuordnung des L9-Tasksteuerblocks aufgehoben und an seiner Stelle ein L8-Tasksteuerblock angehängt wird, den die Task verwenden soll.

Um sicherzustellen, dass eine ausreichende Anzahl offener L8- und L9-Tasksteuerblöcke verfügbar ist, um die Db2-Workload bewältigen zu können, legen Sie den Grenzwert in Ihrem Systeminitialisierungsparameter MAXOPENTCBS auf einen Wert fest, der größer ist als der im Attribut TCBLIMIT Ihrer DB2CONN-Definition festgelegte Wert. Wenn MAXOPENTCBS niedriger ist als TCBLIMIT, kann es sein, dass im System keine offenen L8- oder L9-Tasksteuerblöcke mehr vorhanden sind, bevor der Wert von TCBLIMIT erreicht wird. Stellt CICS eine Verbindung zu Db2 her, wird der Warnhinweis DFHDB2211 abgesetzt, wenn die CICS-Db2-Anschlussfunktion feststellt, dass die Einstellung für MAXOPENTCBS in der Systeminitialisierungstabelle (SIT) niedriger als die Einstellung für TCBLIMIT in der DB2CONN-Definition ist. Wenn Sie diesen Warnhinweis erhalten, müssen Sie den Grenzwert MAXOPENTCBS anpassen.

Legen Sie darüber hinaus für MAXOPENTCBS den Wert für maximale Tasks (MXT) oder höher fest, wenn eine Ausführung erfolgt, bei der die Transaktionsisolation aktiv ist. Diese Einstellung minimiert die Möglichkeit, dass Tasksteuerblöcke aufgrund der Tatsache, dass ein Tasksteuerblock dem falschen Unterbereich zugeordnet wurde, gestohlen werden.

Auswirkungen bei der SQL-Verarbeitung

Die Db2-Hauptaktivitäten bei der Verarbeitung von CICS-Transaktionen mit SQL-Aufrufen sind folgende: Threaderstellung, SQL-Verarbeitung, Verarbeitung der Festschreibung, Threadfreigabe und Threadbeendigung.

Diese Hauptaktivitäten werden für jede Transaktion ausgeführt. Welche Arbeit bei jeder einzelnen Aktivität genau auszuführen ist, hängt von den DB2CONN- und DB2ENTRY-Optionen sowie den BIND-Optionen und von der Frage ab, ob Pakete verwendet werden.

Threaderstellung

Zum Zeitpunkt der Threaderstellung treten eine Reihe von Aktivitäten auf, zu denen die Berechtigungsprüfung für den Plan gehört, die stets stattfinden muss.

In Abhängigkeit von den angegebenen BIND-Optionen können zum Zeitpunkt der Threaderstellung folgende Aktivitäten auftreten:

- Anmeldung.
- Prüfen der maximalen Anzahl von Threads.
- Für einen Anwendungsplan: Laden der Gerüstcursortabelle (SKCT – Skeleton Cursor Table) und des Headers, falls diese noch nicht im EDM-Pool vorhanden sind (EDM - Environmental Description Manager).
- Kopie des SKCT-Headers in der Cursortabelle (CT) erstellen.
- Berechtigungsprüfung für den Plan ausführen.

Der Steuerblock für einen Anwendungsplan, die Gerüstcursortabelle, ist in Abschnitte unterteilt. Der Header und das Verzeichnis einer Gerüstcursortabelle enthalten Steuerinformationen; SQL-Abschnitte enthalten SQL-Anweisungen aus der Anwendung. Für jeden Thread, der den Plan ausführt, wird eine als Cursortabelle (CT) bezeichnete Kopie der Gerüstcursortabelle erstellt. Bei Erstellung des Threads werden nur der Header und das Verzeichnis geladen, falls sie sich noch nicht im EDM-Pool befinden.

Die SQL-Abschnitte der Plansegmente werden zum Zeitpunkt der Threaderstellung nie in die Cursortabelle kopiert. Sie werden Abschnitt für Abschnitt dorthin kopiert, wenn die entsprechenden SQL-Anweisungen ausgeführt werden. Bei einem geschützten Thread mit RELEASE(DEALLOCATE) nimmt die Größe der Cursortabelle so lange zu, bis alle Segmente in die Cursortabelle kopiert wurden.

Wenn die SQL-Anweisungen für die Transaktionen als Paket und nicht als Plan gebunden werden, verwendet Db2 eine Gerüstpakettabelle (SKPT – Skeleton Package Table) und keine Gerüstcursortabelle sowie eine Pakettabelle und keine Cursortabelle. Die Gerüstpakettabelle wird bei Ausführung der ersten SQL-Anweisung zugeordnet; sie wird nicht bei Threaderstellung zugeordnet.

SQL-Verarbeitung

Bei jeder Verarbeitung einer SQL-Anweisung gibt es eine Reihe von Aktivitäten, die auftreten können. Diese Aktivitäten variieren je nach Threadwiederverwendung und angegebenen BIND-Optionen.

- Bei dem ersten SQL-Aufruf in einer Transaktion, die einen Thread mit einer neuen Berechtigungs-ID wiederverwendet, können folgende Aktionen auftreten:
 - Anmeldung
 - Berechtigungsprüfung
- Laden des SQL-Abschnitts der Gerüstcursortabelle, falls er sich noch nicht im EDM-Pool befindet.
- Kopie des SQL-Abschnitts der Gerüstcursortabelle in der Cursortabelle erstellen, falls er dort noch nicht vorhanden ist.
- Referenzierte TS-Sperren anfordern, falls dies noch nicht geschehen ist.
- Datenbankanforderungsmodule in den EDM-Pool laden, falls dort noch keine vorhanden sind.
- SQL-Anweisung verarbeiten.

Wenn sich die SQL-Anweisung in einem Paket befindet, werden das Verzeichnis und der Header der Gerüstpakettabelle geladen. Die Pakettabelle wird zum Zeitpunkt der Anweisungsausführung und nicht zum Zeitpunkt der Threaderstellung zugeordnet, wie es auch bei der Gerüstcursortabelle und der Cursortabelle zum Binden von Plänen der Fall ist.

Verarbeitung der Festschreibung

Zum Zeitpunkt der Festschreibung (Commit) kann eine Reihe von Aktivitäten auftreten; zu diesen Aktivitäten gehört das Freigeben von Seitensperren und TS-Sperren.

In Abhängigkeit von den angegebenen BIND-Optionen können zum Zeitpunkt der Festschreibung folgende Aktivitäten auftreten:

- Freigeben der Seitensperren
- Wenn RELEASE(COMMIT) angegeben wurde:
 - Freigeben der TS-Sperren
 - Freigeben der CT-Seiten.

Threadfreigabe

Der Zeitpunkt, an dem Transaktionen einen verwendeten Thread freigeben, hängt von den Optionen ab, die angegeben sind, und davon, ob die Transaktion terminalorientiert ist.

Wenn die Transaktion terminalorientiert ist oder wenn sie nicht terminalorientiert ist **1**, im DB2CONN-Objekt aber NONTERMREL=YES angegeben ist, wird der Thread sowohl am Synchronisationspunkt (SYNCPOINT) als auch am Taskende (EOT – End of Task) freigegeben. Dadurch ist es effizient, für Transaktionen, die viele Synchronisationspunkte (SYNCPOINT) setzen, einen geschützten Thread zu verwenden, wenn er mit den BIND-Optionen ACQUIRE(USE) und RELEASE(DEALLOCATE) kombiniert wird. In diesem Fall werden die Ressourcen, die zum Ausführen folgender Aktivitäten erforderlich sind, für jeden Synchronisationspunkt gespeichert:

- Thread beenden und starten.
- TS-Sperren freigeben und anfordern.
- Segmente des Plans freigeben und in die Cursortabelle (CT) kopieren.

Threads werden nicht zum Zeitpunkt von SYNCPOINT freigegeben, wenn Folgendes gilt:

- Hold-Cursor sind offen.
- Bestimmte änderbare Db2-Sonderregister weisen nicht den Anfangswert auf. Diese Sonderregister sind in [SQL: Sprache von Db2 in der Produktdokumentation zu Db2 for z/OS](#) aufgelistet.
- Das änderbare Db2-Sonderregister CURRENT DEGREE wird während der Lebensdauer der CICS-Task ständig geändert.

Wenn die Transaktion nicht terminalorientiert ist und Sie NONTERMREL=NO angeben, wird der Thread erst am Taskende (EOT) freigegeben. Sie müssen nicht einen geschützten Thread verwenden, um nach dem Absetzen des Befehls **EXEC CICS SYNCPOINT** die Möglichkeit der Threadwiederverwendung zu haben. Sie sollten jedoch dennoch einen geschützten Thread verwenden, wenn es sich um eine häufig verwendete Transaktion handelt. Die BIND-Optionen ACQUIRE(USE) und RELEASE(DEALLOCATE) bieten dieselben Vorteile wie für terminalorientierte Transaktionen mit häufigen Synchronisationspunkten.

Anmerkung: 1 Nicht terminalorientierte Transaktionen sind Transaktionen, die keinem Terminal zugeordnet sind.

Threadbeendigung

Zum Zeitpunkt der Threadbeendigung kann eine Reihe von Aktivitäten auftreten; zu diesen Aktivitäten gehört die Freigabe von TS-Sperren und das Freigeben von Arbeitsspeicher.

In Abhängigkeit von den BIND-Optionen können zum Zeitpunkt der Threadbeendigung folgende Aktivitäten auftreten:

- Wenn RELEASE(DEALLOCATE) angegeben wurde:

- Freigeben der TS-Sperren
- Freigeben der CT-Seiten
- Freigeben des Arbeitsspeichers.

Threads erstellen, verwenden und beenden

Von der CICS-Db2-Anschlussfunktion werden drei Haupttypen von Threads verwendet: Befehlsthreads, Einstiegsthreads und Pool-Threads. Es gibt bestimmte Regeln, die für die Erstellung, Verwendung und Beendigung von Threads gelten.

Wenn CICS mit Db2 verbunden ist, ist der Thread-Tasksteuerblock der offene Tasksteuerblock, in dem die CICS-Db2-Anschlussfunktion bereits ausgeführt wird. Wenn sie einen Thread ausführen muss, ordnet die CICS-Db2-Anschlussfunktion den offenen Tasksteuerblock dem Db2-Verbindungssteuerblock und dem Thread zu, die verwendet werden sollen, und der offene Tasksteuerblock führt den Thread anschließend aus. Wenn der Thread nicht mehr benötigt wird, hebt der offene Tasksteuerblock die Zuordnung zum Thread auf und der Db2-Verbindungssteuerblock und der Thread werden für die Wiederverwendung durch einen anderen offenen Tasksteuerblock verfügbar.

Folgende allgemeine Regeln gelten für die Erstellung, Verwendung und Beendigung von Threads:

- Wenn CICS mit Db2 verbunden ist und offene Tasksteuerblöcke als Thread-Tasksteuerblöcke verwendet, muss ein Thread für die Transaktion verfügbar sein, bevor eine SQL-Anforderung an Db2 übergeben werden kann. Der offene Tasksteuerblock ordnet sich selbst dem Thread zu und wird zum Thread-Tasksteuerblock, bis er die Zuordnung zum Thread aufhebt.
- Wenn nach Erstellen eines Threads eine andere Transaktion mit einer neuen Berechtigungs-ID einen Thread wiederverwendet, führt Db2 eine Berechtigungsprüfung für die neue Berechtigungs-ID durch.
- Eine terminalorientierte Transaktion gibt den Thread in der Regel am Synchronisationspunkt und am Taskende frei. Der Thread wird *nicht* am Synchronisationspunkt freigegeben, wenn Hold-Cursor offen sind oder wenn sich änderbare Sonderregister nicht in ihrem ursprünglichen Zustand befinden.
- Eine nicht terminalorientierte Transaktion gibt den Thread erst am Taskende frei, es sei denn, im DB2CONN-Objekt ist NONTERMREL=YES angegeben.
- Wenn eine Transaktion einen Thread freigibt, kann der Thread von einer anderen Transaktion wiederverwendet werden, die denselben Plan angibt und die im selben DB2ENTRY-Objekt definiert ist. Pool-Threads können von einer beliebigen wartenden (in die Warteschlange eingereihten) Transaktion wiederverwendet werden, die denselben Plan angibt und einen Pool-Thread verwendet.
- Ein ungeschützter Thread wird sofort nach Freigabe beendet, es sei denn, eine andere Transaktion wartet auf den Thread (ist für den Thread in die Warteschlange eingereiht).
- Ein geschützter Thread wird beendet, wenn er während zweier aufeinanderfolgender Reinigungszyklen nicht verwendet wird. Bei der Festlegung von Standardeinstellungen sind dies durchschnittlich 45 Sekunden. Dieser Wert kann durch den Parameter PURGECYCLE des DB2CONN-Objekts geändert werden.
- Ein Thread wird abgebrochen und beendet, wenn er in Db2 an dem Zeitpunkt aktiv ist, an dem die den Thread verwendende CICS-Task bereinigt oder erzwungen bereinigt wird (Force Purge).
- Der Parameter REUSELIMIT des DB2CONN-Objekts gibt die maximale Anzahl an, wie häufig ein Thread wiederverwendet werden kann, bevor er beendet wird. Der Grenzwert für die Wiederverwendung gilt für ungeschützte Threads im Pool und im DB2ENTRY-Objekt sowie für geschützte DB2ENTRY-Threads.
- Mit dem Parameter THREADWAIT wird definiert, ob die Anforderungen für einen Thread in die Warteschlange eingereiht, abgebrochen oder an den Pool-Thread gesendet werden sollen, falls es einen Engpass bei Einstiegs- oder Befehlsthreads gibt. Wenn THREADWAIT=YES statt THREADWAIT=POOL angegeben ist, wird die Transaktion in die Warteschlange eingereiht, statt an den Pool-Thread gesendet. Durch Verwenden der Einstellung THREADWAIT=YES wird der Systemaufwand für die Threadinitialisierung und -beendigung vermieden. Wenn eine Transaktion aufgrund des Mangels an Einstiegsthreads warten muss, reiht die CICS-Db2-Anschlussfunktion die Transaktion in die Warteschlange ein. Die Vorteile dieses Einreihens in die Warteschlange bestehen darin, dass der Einstiegsthread bei Beenden seiner aktuellen Arbeitseinheit sofort mit der nächsten Transaktion fortfährt.

- TCBLIMIT gibt die maximale Anzahl von Tasksteuerblöcken an, die zum Ausführen von Db2-Threads verwendet werden kann, wodurch wiederum die maximale Anzahl aktiver Db2-Threads begrenzt wird. THREADLIMIT gibt die maximale Anzahl aktiver Db2-Threads an. THREADLIMIT wird dynamisch geändert. CTHREAD wird in ZPARMS angegeben und definiert die Anzahl der gleichzeitig ausgeführten Threads in der gesamten Db2-Instanz. Die Summe aller aktiven Threads, die von TSO-Benutzern, allen CICS- und IMS-Systemen und anderen Systemen verwendet werden, die auf Db2 zugreifen, sollte die durch CTHREAD festgelegte Anzahl nicht überschreiten. Andernfalls könnte dies zu unvorhersehbaren Antwortzeiten führen. Tritt dies auf, wird die von der CICS-Db2-Anschlussfunktion ausgehende Anforderung "create thread" (Thread erstellen) von Db2 in die Warteschlange eingereiht und die CICS-Transaktion wird in den Wartestatus versetzt, bis ein Thread zur Verfügung steht.

CICS verwaltet die gesamte Anzahl der L8- und L9-Tasksteuerblöcke, die die CICS-Region gleichzeitig in Betrieb hat. Die maximale Anzahl der L8- und L9-Tasksteuerblöcke im offenen Tasksteuerblockpool wird entweder vom Systeminitialisierungsparameter MXT oder MAXOPENTCBS gesteuert. Der Systeminitialisierungsparameter MAXOPENTCBS legt, falls angegeben, den Wert für den Pool der offenen Tasksteuerblöcke fest. Wenn der Systeminitialisierungsparameter MAXOPENTCBS nicht angegeben ist, legt CICS den Grenzwert für den Pool der offenen L8- und L9-Tasksteuerblöcke automatisch auf der Basis der maximalen Anzahl von Tasks fest, die für die CICS-Region angegeben wurden (der Wert MXT); dies erfolgt mithilfe der Formel $(2 * MXT \text{ Value}) + 32$. In der Umgebung für offene Transaktionen steuert TCBLIMIT, wie viele dieser offenen Tasksteuerblöcke von dem taskbezogenen CICS-Db2-Benutzerexit verwendet werden können, um Threads in Db2 auszuführen. Wenn der durch CICS festgelegte Grenzwert erreicht ist, sind keine weiteren offenen Tasksteuerblöcke in der CICS-Region zulässig und der taskbezogene CICS-Db2-Benutzerexit kann keinen offenen Tasksteuerblock zur Verwendung anfordern.

Jeder Thread, der CICS mit Db2 verbindet, verfügt über einen entsprechenden Tasksteuerblock im CICS-Adressraum. Sind zu viele Tasksteuerblöcke pro Adressraum vorhanden, wird der MVS-Dispatcher einbezogen, der die Tasksteuerblöcke durchsucht, um einen aktiven Tasksteuerblock zu ermitteln. Ist eine große Anzahl von Tasksteuerblöcken vorhanden, könnte dies viel Prozessorzeit kosten. Wenn Sie jedoch zu wenige Tasksteuerblöcke zur Verfügung haben, um Ihre Db2-Workload zu bewältigen, müssen Transaktionen warten, bis sie einen Tasksteuerblock erhalten.

Durch Anheben des Werts TCBLIMIT oder Einrichten eines zusätzlichen CICS-Systems, das Zugriff auf dasselbe Db2-System hat, wird möglicherweise erforderlich, dass der Db2-Parameter CTHREAD erhöht werden muss.

Überprüfen Sie bei einer Umgebung mit geschützten Threads die Anzahl der Anwendungspläne und reduzieren Sie, falls möglich, die Anzahl der Pläne, indem Sie die wenig verwendeten Pläne miteinander kombinieren, während Sie die Probleme der Plangröße und der Sicherheit gegeneinander abwägen. Sie sollten mit einem einzigen Thread pro Plan beginnen. In einer Umgebung, in der ein hohes Transaktionsvolumen verarbeitet wird, können Sie die Anfangszahl schätzen, indem Sie die Zeit, wie lange eine Transaktion einen Thread belegt, mit der erwarteten Transaktionsrate multiplizieren. Eine Belegungszeit von 0,2 Sekunden und eine Transaktionsrate von 20 Transaktionen pro Sekunde ($0,2 \times 20$) würde als Anfangszahl für Threads einen Wert zwischen drei und vier ergeben.

Geschützte Einstiegsthreads

Geschützte Einstiegsthreads werden mithilfe folgender DB2ENTRY-Parameter definiert: PROTECTNUM(n) und THREADLIMIT(n).

Geschützte Einstiegsthreads werden für Folgendes empfohlen:

- Transaktionen mit großem Volumen unabhängig vom Typ
- Terminalorientierte Transaktionen mit vielen Festschreibungen
- Nicht terminalorientierte Transaktionen mit vielen Festschreibungen (wenn NONTERMREL=YES im DB2CONN-Objekt angegeben ist)

Geschützte Threads werden wie folgt erstellt, verwendet und beendet:

Anhängen von Tasksteuerblöcken

Wenn CICS mit Db2 verbunden ist und die Umgebung für offene Transaktionen verwendet, wird der CICS-Task vor dem Aufrufen der CICS-Db2-Anschlussfunktion ein neuer oder vorhandener offener

Tasksteuerblock zugeordnet. Wenn der automatisch von CICS für die Anzahl von offenen L8- und L9-Tasksteuerblöcken festgelegte Grenzwert erreicht ist, können keine weiteren offenen Tasksteuerblöcke erstellt werden und die Task geht in einen Wartezustand des CICS-Dispatchers über, bis ein offener Tasksteuerblock verfügbar ist. Am Ende der Task wird der offene Tasksteuerblock an den vom CICS-Dispatcher verwalteten Pool offener Tasksteuerblöcke zurückgegeben.

Threaderstellung

Ein Thread wird nur erstellt, wenn kein vorhandener Thread verfügbar ist. Wenn für eine Task keine Threads verfügbar sind, wird ein neuer Thread erstellt und dem offenen Tasksteuerblock der Task zugeordnet, vorausgesetzt, der Wert von THREADLIMIT wird dadurch nicht überschritten. Wenn der durch TCBLIMIT gesetzte Grenzwert erreicht ist, können für das DB2ENTRY-Objekt keine weiteren offenen Tasksteuerblöcke als Thread-Tasksteuerblöcke verwendet werden.

Threadbeendigung

Wenn der Thread freigegeben wird und keine neue Arbeit in der Warteschlange vorhanden ist, wird der Thread als geschützt markiert, falls die aktuelle Anzahl geschützter Threads geringer als der durch PROTECTNUM festgelegte Wert ist. Andernfalls wird er beendet. Ein geschützter Thread wird beendet, wenn er während zweier aufeinanderfolgender Bereinigungszyklen nicht verwendet wird.

Threadwiederverwendung

Andere Transaktionen, die dasselbe DB2ENTRY-Objekt verwenden, können einen Thread wiederverwenden, falls er verfügbar ist. Dies ist wahrscheinlich, da die Threads nach der Freigabe für ca. 45 Sekunden aktiv bleiben (dies hängt vom Wert für PURGECYCLE ab).

Überlauf in Pool

Ist THREADWAIT=POOL angegeben, werden Anforderungen nach Threads an den Pool übertragen, wenn der Wert für THREADLIMIT überschritten wird. Wenn eine Transaktion durch Überlauf an den Pool übertragen wird, wird die Transaktion nun durch die Attribute PRIORITY, THREADLIMIT und THREADWAIT gesteuert, die für Pool-Threads in der DB2CONN-Definition angegeben sind; die Attribute in der DB2ENTRY-Definition dagegen werden ignoriert. Die verbleibenden Attribute, die in der DB2ENTRY-Definition für den Einstiegsthread angegeben sind (d. h. die Attribute ACCOUNTREC, AUTHID bzw. AUTHTYPE, DROLLBACK und PLAN bzw. PLANEXITNAME), gelten weiterhin für die Transaktion. Das Attribut PROTECTNUM in der DB2ENTRY-Definition ist für eine Transaktion, die durch einen Überlauf an den Pool übertragen wird, nicht mehr relevant, sodass diese Einstellung ignoriert wird.

Ungeschützte Einstiegsthreads für kritische Transaktionen

Ungeschützte Einstiegsthreads für kritische Transaktionen werden mithilfe der folgenden DB2ENTRY-Parameter definiert: PROTECTNUM(0) und THREADLIMIT(n).

Dies ist der empfohlene Definitionstyp für:

- Kritische Transaktionen, für die eine kurze Antwortzeit erforderlich ist, deren Volumen jedoch so gering ist, dass kein geschützter Thread verwendet werden kann.
- Transaktionen mit eingeschränktem gemeinsamem Zugriff

Sie könnten einen geschützten Thread für Transaktionen mit eingeschränktem gemeinsamem Zugriff verwenden, wenn es aufgrund der Transaktionsrate möglich ist, den Thread wiederzuverwenden.

Ungeschützte Einstiegsthreads für kritische Transaktionen werden wie folgt erstellt, verwendet und beendet:

Anhängen von Tasksteuerblöcken

Wenn die CICS-Db2-Anschlussfunktion gestartet wurde, werden keine Thread-Tasksteuerblöcke angehängt.

Ein Tasksteuerblock wird nur angehängt, wenn er von einem Thread benötigt wird.

Threaderstellung

Ein Thread wird nur erstellt, wenn er von einer Transaktion benötigt wird.

Wenn CICS mit Db2 verbunden ist (und dadurch die Umgebung für offene Transaktionen verwendet) und kein Thread verfügbar ist, ein offener Tasksteuerblock jedoch als Thread-Tasksteuerblock für dieses DB2ENTRY-Objekt verwendet werden kann, wird ein neuer Thread erstellt und dem Tasksteuer-

block zugeordnet, vorausgesetzt, der Wert für THREADLIMIT wird nicht überschritten. Wenn der durch TCBLIMIT gesetzte Grenzwert erreicht ist, können für das DB2ENTRY-Objekt keine weiteren offenen Tasksteuerblöcke als Thread-Tasksteuerblöcke verwendet werden.

Threadbeendigung

Der Thread wird sofort nach Freigabe beendet, es sei denn, für den Thread ist eine Transaktion in der Warteschlange vorhanden.

Threadwiederverwendung

Andere Transaktionen, für die die Verwendung desselben DB2ENTRY-Objekts angegeben ist, können einen Thread wiederverwenden, falls er verfügbar ist. Dies passiert nur, wenn bei Verfügbarwerden des Threads eine Transaktion auf den Thread wartet.

Überlauf in Pool

Ist THREADWAIT=POOL angegeben, werden Anforderungen nach Threads an den Pool übertragen, wenn der Wert für THREADLIMIT überschritten wird. Wenn eine Transaktion durch Überlauf an den Pool übertragen wird, wird die Transaktion nun durch die Attribute PRIORITY, THREADLIMIT und THREADWAIT gesteuert, die für Pool-Threads in der DB2CONN-Definition angegeben sind; die Attribute in der DB2ENTRY-Definition dagegen werden ignoriert. Die verbleibenden Attribute, die in der DB2ENTRY-Definition für den Einstiegsthread angegeben sind (d. h. die Attribute ACCOUNTREC, AUTHID bzw. AUTHTYPE, DROLLBACK und PLAN bzw. PLANEXITNAME), gelten weiterhin für die Transaktion. Das Attribut PROTECTNUM in der DB2ENTRY-Definition ist für eine Transaktion, die durch einen Überlauf an den Pool übertragen wird, nicht mehr relevant, sodass diese Einstellung ignoriert wird.

Beachten Sie, dass Sie den Überlauf an den Pool nicht für Transaktionen mit eingeschränktem gemeinsamem Zugriff zulassen sollten.

Ungeschützte Einstiegsthreads für Hintergrundtransaktionen

Ungeschützte Einstiegsthreads werden mithilfe folgender DB2ENTRY-Parameter definiert: PROTECTNUM(0), THREADLIMIT(0) und THREADWAIT(POL).

Ungeschützte Einstiegsthreads werden für Transaktionen mit niedrigem Volumen empfohlen, für die keine kurze Antwortzeit erforderlich ist. Für alle Transaktionen wird die Verwendung von Pool-Threads erzwungen.

Ungeschützte Einstiegsthreads für Hintergrundtransaktionen werden wie folgt erstellt, verwendet und beendet:

Anhängen von Tasksteuerblöcken

Da THREADLIMIT=0 gilt, wird für diese Threaddefinition nie ein Thread-Tasksteuerblock für Subtasks angehängt. Es wird ein Pool-Thread-Tasksteuerblock (oder, in der Umgebung für offene Transaktionen, ein offener Tasksteuerblock) verwendet. Für jede Aktivität, die dieser Eintragsdefinition zugeordnet sind, wird ein Thread und ein Tasksteuerblock im Pool erzwungen. Eine Transaktion unterliegt dann der Steuerung durch die Parameter PRIORITY, THREADLIMIT und THREADWAIT für den Pool. Die Transaktion behält die Werte für PLAN und AUTHID/AUTHTYPE bei, die Sie für den Einstiegsthread angegeben haben.

Threaderstellung

Ein Thread wird im Pool erstellt, wenn er von einer Transaktion benötigt wird, es sei denn, der für den Pool geltende Wert für THREADLIMIT wurde erreicht.

Threadbeendigung

Der Thread wird nach der Freigabe beendet, es sei denn, für den Thread ist eine Transaktion in der Warteschlange vorhanden.

Threadwiederverwendung

Andere Transaktionen, die denselben Plan verwenden, können den Thread wiederverwenden, wenn er verfügbar wird.

Pool-Threads

Pool-Threads werden mit dem DB2CONN-Parameter THREADLIMIT(n) definiert.

Es gibt vier Situationen, in denen ein Pool-Thread verwendet werden kann:

1. Eine Transaktion ist in keinem DB2ENTRY- oder DB2TRAN-Objekt angegeben, aber sie gibt SQL-Anforderungen aus. Diese Transaktion geht standardmäßig an den Pool über und verwendet den für den Pool angegebenen Plan.
2. Eine Transaktion ist in einem DB2ENTRY- oder DB2TRAN-Objekt angegeben, das auf ein DB2ENTRY-Objekt verweist; für die Transaktion wird aber eine Übergabe an den Pool erzwungen, da das DB2ENTRY-Objekt THREADLIMIT(0) und THREADWAIT(PPOOL) angibt. Diese Transaktion verwendet den im DB2ENTRY-Objekt angegebenen Plan.
3. Eine Transaktion ist in einem DB2ENTRY- oder DB2TRAN-Objekt angegeben, das auf ein DB2ENTRY-Objekt verweist; es wird für die Transaktion aber ein Überlauf an den Pool ausgeführt (THREADWAIT=PPOOL), wenn der Wert für THREADLIMIT überschritten wird. Diese Transaktion verwendet den im DB2ENTRY-Objekt angegebenen Plan.
4. Eine Transaktion ist in einem DB2ENTRY- oder DB2TRAN-Objekt angegeben, das auf ein DB2ENTRY-Objekt verweist, aber das DB2ENTRY-Objekt ist inaktiviert. Für das Schlüsselwort DISABLEDACT ist die Einstellung POOL festgelegt, daher wird ein Pool-Thread verwendet. Diese Transaktion verwendet den für den Pool angegebenen Plan.

Pool-Threads sind immer ungeschützte Threads.

Pool-Threads werden wie folgt erstellt, verwendet und beendet:

Anhängen von Tasksteuerblöcken

Wenn die CICS-Db2-Anschlussfunktion gestartet wurde, werden keine Thread-Tasksteuerblöcke angehängt.

Ein Tasksteuerblock wird nur angehängt, wenn er von einem Thread benötigt wird.

Threaderstellung

Ein Thread wird nur erstellt, wenn er von einer Transaktion benötigt wird.

Threadbeendigung

Der Thread wird sofort nach Freigabe beendet, es sei denn, für den Thread ist eine Transaktion in der Warteschlange vorhanden.

Threadwiederverwendung

Andere Transaktionen, die denselben Plan verwenden, können einen Thread wiederverwenden, wenn er verfügbar wird. Im Pool tritt dies nur auf, wenn eine Warteschlange für Threads vorhanden ist und die erste Transaktion in der Warteschlange denselben Plan anfordert, der auch von dem freigegebenen Thread verwendet wurde.

Threadtypen für optimale Leistung auswählen

Wählen Sie zum Optimieren der Leistung den für eine Gruppe von Transaktionen zu verwendenden Threadtyp zusammen mit den BIND-Parametern für den entsprechenden Plan aus.

Es ist wichtig, Threadtyp und BIND-Parameter zusammen auszuwählen, da durch die BIND-Parameter festgelegt wird, ob eine Anzahl von Aktivitäten auf den Thread oder auf die Transaktionen bezogen ist. Weitere Informationen hierzu finden Sie in „[BIND-Optionen für optimale Leistung auswählen](#)“ auf Seite 23.

Folgende Threadtypen sind verfügbar:

Geschützte Einstiegsthreads

Geschützte Einstiegsthreads werden für Folgendes empfohlen:

- Bei hohen Transaktionsvolumen jeden Typs
- Terminalorientierte Transaktionen mit vielen Festschreibungen
- Nicht terminalorientierte Transaktionen mit vielen Festschreibungen (wenn NONTERMREL=YES im DB2CONN-Objekt angegeben ist)

Weitere Informationen hierzu finden Sie in „[Threads erstellen, verwenden und beenden](#)“ auf Seite 18.

Ungeschützte Einstiegsthreads für kritische Transaktionen

Ungeschützte Einstiegsthreads für kritische Transaktionen werden für Folgendes empfohlen:

- Kritische Transaktionen, für die eine kurze Antwortzeit erforderlich ist, deren Volumen jedoch so gering ist, dass ein geschützter Thread nicht verwendet werden kann.
- Transaktionen mit eingeschränktem gemeinsamem Zugriff

Sie können für Transaktionen mit eingeschränktem gemeinsamem Zugriff einen geschützten Thread verwenden, wenn es aufgrund der Transaktionsrate möglich ist, den Thread wiederzuverwenden.

Ungeschützte Einstiegsthreads für Hintergrundtransaktionen

Ungeschützte Einstiegsthreads für Hintergrundtransaktionen werden für Transaktionen mit einem geringen Volumen empfohlen, für die eine kurze Antwortzeit nicht erforderlich ist. Die Verwendung von Pool-Threads wird für alle Transaktionen erzwungen.

Geschützte Threads zu verwenden (indem in der DB2ENTRY-Definition für einen Einstiegsthread PROTECTNUM=n angegeben wird) ist eine Leistungsoption, durch die die Anzahl der an der Erstellung und Beendigung eines Threads beteiligten Ressourcen verringert wird. Ein geschützter Thread wird bei Beendigung einer Transaktion nicht beendet und die nächste Transaktion, die demselben DB2ENTRY-Objekt zugeordnet ist, verwendet den Thread wieder. Durch Verwendung von geschützten Threads eliminieren Sie einen großen Teil der für die Threaderstellung und -beendigung von einzelnen Transaktionen erforderlichen Arbeit. Aufgrund von Leistungsaspekten wird empfohlen, dass Sie geschützte Einstiegsthreads verwenden, wann immer es möglich ist, insbesondere, wenn eine Transaktion häufig verwendet wird oder viele SYNCPOINTS ausgibt. Eine Ausnahme sollte gemacht werden, wenn eine Transaktion nicht häufig verwendet wird, weil es wahrscheinlich ist, dass zwischen solchen Transaktionen auch ein geschützter Thread beendet wird.

Geht man vom Gesichtspunkt der Abrechnung aus, sieht die Situation anders aus. Bei jeder Threadbeendigung und jeder neuen Benutzeranmeldung wird ein Abrechnungsdatensatz erstellt. Dies bedeutet, dass nur ein einziger Abrechnungsdatensatz erstellt wird, wenn der Thread aktiv bleibt und die Benutzer-ID nicht geändert wird. Dieser Datensatz enthält zusammengefasste Werte für alle Transaktionen, die denselben Thread verwenden; ein Wert kann jedoch nicht einer bestimmten Transaktion zugeordnet werden. Dies wird behoben, wenn ACCOUNTREC(UOW) angegeben wird, um sicherzustellen, dass pro Arbeitseinheit ein Abrechnungsdatensatz erstellt wird, oder wenn ACCOUNTREC(TASK) angegeben wird, um sicherzustellen, dass pro CICS-Task ein Abrechnungsdatensatz erstellt wird. In [Abrechnung und Überwachung in einer CICS-Db2-Umgebung unter "Überwachen"](#) finden Sie weitere Informationen zur Abrechnung in einer CICS-Db2-Umgebung.

Es kann angegeben werden, dass mehrere Transaktionen dasselbe DB2ENTRY-Objekt verwenden. Idealerweise sollten sie alle denselben Plan verwenden. Jede Transaktion mit niedrigem Volumen kann ein eigenes DB2ENTRY-Objekt haben. In diesem Fall ist die Threadwiederverwendung nicht wahrscheinlich und Sie sollten PROTECTNUM=0 angeben. Als Alternative dazu können Sie auch eine Anzahl von Transaktionen mit niedrigem Volumen im selben DB2ENTRY-Objekt gruppieren und PROTECTNUM=n angeben. Dadurch wird eine bessere Threadauslastung erzielt und der Systemaufwand ist geringer.

Wenn ein Thread erstellt wird und wenn ein Thread mit einem neuen Benutzer wiederverwendet wird, finden Berechtigungsprüfungen statt. Das Vermeiden des Systemaufwands für die Sicherheitsprüfung ist Teil der Leistungsvorteile, die durch die Benutzung geschützter Threads entstehen. Dies bedeutet, dass vom Gesichtspunkt der Leistung aus alle Transaktionen, für die die Verwendung desselben DB2ENTRY-Objekts mit PROTECTNUM>0 definiert ist, dieselbe Berechtigungs-ID verwenden sollten. Es sollte zumindest vermieden werden, TERM, OPID und USERID für den Parameter AUTHTYPE zu verwenden, da diese Werte innerhalb der Instanzen einer Transaktion häufig variieren.

Es ist wichtig, dass Sie Ihre DB2CONN-, DB2ENTRY- und Ihre BIND-Optionen koordinieren. Weitere Informationen hierzu finden Sie in [„DB2CONN-, DB2ENTRY- und BIND-Optionen koordinieren“](#) auf Seite 24.

BIND-Optionen für optimale Leistung auswählen

Sie müssen die BIND-Optionen, die Sie für einen Plan auswählen, mit den Threadtypen koordinieren, die von den Transaktionen verwendet werden, die diesem Plan zugeordnet sind.

Eine Übersicht des Bindeprozesses finden Sie in [Bindeprozess](#). Informationen zur Auswahl von Threadtypen finden Sie in [„Threadtypen für optimale Leistung auswählen“](#) auf Seite 22.

Die Bindeoption VALIDATE kann sich auf die Leistung auswirken. Verwenden Sie in einer CICS-Umgebung VALIDATE(BIND). Weitere Informationen hierzu finden Sie in [Bindeoptionen und -aspekte bei Programmen](#).

Es ist wichtig, dass Sie Ihre DB2CONN-, DB2ENTRY- und Ihre BIND-Optionen koordinieren. Weitere Informationen finden Sie in [„DB2CONN-, DB2ENTRY- und BIND-Optionen koordinieren“](#) auf Seite 24.

DB2CONN-, DB2ENTRY- und BIND-Optionen koordinieren

Sie können aus DB2CONN-, DB2ENTRY- und BIND-Optionen viele verschiedene Kombinationen erstellen.

Eines der wichtigsten Dinge, die Sie für die Leistungsoptimierung ausführen müssen, ist zu definieren, ob eine bestimmte Gruppe von Transaktionen einen oder mehrere geschützte Threads verwenden muss. Ausführlichere Informationen dazu finden Sie in [„Threadtypen für optimale Leistung auswählen“](#) auf Seite 22. Erwägen Sie als nächstes, die BIND-Parameter zu definieren, um den Gesamtumfang der Arbeit zu minimieren, der für die Hauptaktivitäten erforderlich ist, die bei der Verarbeitung von SQL-Transaktionen beteiligt sind. Weitere Informationen finden Sie in [„BIND-Optionen für optimale Leistung auswählen“](#) auf Seite 23.

Im Allgemeinen wird empfohlen, als Anfangswerte für Ihre DB2CONN-, DB2ENTRY- und BIND-Optionen die in [Tabelle 1 auf Seite 24](#) gezeigten Werte festzulegen. Möglicherweise stellen Sie fest, dass andere Kombinationen für Ihre Transaktionen zu einer besseren Leistung führen. Es werden für jeden Transaktionstyp ein empfohlener Threadtyp und eine empfohlene BIND-Option gezeigt. Es sind auch Empfehlungen dafür aufgeführt, ob für Transaktionen ein Überlauf in den Pool ausgeführt werden soll.

Tabelle 1. Empfohlene Kombinationen aus DB2CONN-, DB2ENTRY- und BIND-Optionen				
Transaktionsbeschreibung	Threadtyp	Überlauf	ACQUIRE	RELEASE
Großes Volumen (alle Typen)	Geschützter Einstiegsthread	Hinweis 1	USE	DEALLOCATE
Terminalorientiert mit vielen Festschreibungen (plus nicht terminalorientierte, wenn NONTERMREL=YES gilt)	Geschützter Einstiegsthread	Hinweis 2	USE	DEALLOCATE
Geringes Volumen, kurze Antwortzeit erforderlich	Ungeschützter Einstiegsthread	Ja	USE	COMMIT
Geringes Volumen, eingeschränkter gemeinsamer Zugriff	Ungeschützter Einstiegsthread	Nie	USE	COMMIT
Geringes Volumen, kurze Antwortzeit nicht erforderlich	Pool	Nicht zutreffend	USE	COMMIT
Nicht terminalorientiert mit vielen Festschreibungen (NONTERMREL=NO)	Hinweis 3	Hinweis 3	USE	DEALLOCATE
Hinweise: <ol style="list-style-type: none"> 1. Ja, definieren Sie jedoch ausreichend Einstiegsthreads, sodass dies selten auftritt. 2. Ja, wenn jedoch ein Überlauf in den Pool stattfindet, werden keine geschützten Threads verwendet. 3. Threads werden bis zum Ende der Task (EOT) beibehalten. Verwenden Sie für eine kurze Transaktion Pool-Threads. Erwägen Sie für Transaktionen mit einer längeren Ausführungszeit die Verwendung von Einstiegsthreads. 				

In Tabelle 1 auf Seite 24 ist mit 'eingeschränktem gemeinsamen Zugriff' gemeint, dass nur eine eingeschränkte Zahl (n) von Transaktionen gleichzeitig ausgeführt werden kann. Es liegt ein besonderer Fall vor, wenn n=1 gilt; in diesem Fall werden die Transaktionen serialisiert. Sie können weiterhin einen geschützten Thread verwenden, wenn die Transaktionsrate so hoch ist, dass sich dies lohnt. Die Transaktionen können nicht gesteuert werden, wenn der Überlauf in den Pool zulässig ist. Es wird empfohlen, den CICS-Mechanismus zu verwenden, mit dem die Zahl der in einer bestimmten Klasse ausgeführten Transaktionen eingeschränkt wird, statt zu erzwingen, dass Transaktionen wegen einer begrenzten Anzahl von Threads in die Warteschlange eingereiht werden.

Wie in Tabelle 1 auf Seite 24 gezeigt sind einige Kombinationen aus DB2CONN-, DB2ENTRY- und BIND-Optionen allgemein empfohlen. In bestimmten Situationen können jedoch andere Kombinationen verwendet werden.

Tabelle 2 auf Seite 25 zeigt eine Zusammenfassung der Aktivitäten, die bei der Verarbeitung von SQL-Anforderungen für die drei empfohlenen Gruppen aus DB2CONN-, DB2ENTRY- und BIND-Spezifikationen beteiligt sind. Die Tabelle zeigt auch den Leistungsvorteil, der sich ergibt, wenn geschützte Threads ohne Änderung der Berechtigungs-ID verwendet werden. Erforderliche Aktivitäten sind in der Tabelle wie folgt gekennzeichnet:

X

Erforderliche Aktivität

(1)

Nur erforderlich, wenn die Berechtigungs-ID neu ist

(2)

Nur erforderlich, wenn sich der SQL-Abschnitt noch nicht im EDM-Pool (EDM – Environmental Description Manager) befindet

(3)

Nur erforderlich, wenn sich der SQL-Abschnitt noch nicht in der Cursortabelle befindet

<i>Tabelle 2. Aktivitäten, die an der Verarbeitung von SQL-Anforderungen für verschiedene DB2CONN-, DB2ENTRY- und BIND-Spezifikationen beteiligt sind</i>				
Aktivität	Geschützte Threads		Ungeschützte Threads	
	ACQUIRE(USE) RELEASE(DEALLOCATE)		ACQUIRE(USE) RELEASE(COMMIT)	ACQUIRE(USE) RELEASE(DEALLOCATE)
	Aktivität für den jeweiligen Thread	Aktivität für die jeweilige Transaktion	Aktivität für die jeweilige Transaktion	Aktivität für die jeweilige Transaktion
Thread erstellen:	X		X	X
SIGNON	X	(1)	X	X
Berechtigungsprüfung	X	(1)	X	X
SKCT-Header laden	X		X	X
CT-Header laden	X		X	X
Alle TS-Sperren anfordern		X	X	X
Alle Datenbankbeschreibungen (DBD) laden		X	X	X
Für jede SQL-Anweisung:				
SKCT-SQL-Abschnitt laden		(2)	(2)	(2)
CT-Kopie erstellen		(3)	X	X

Tabelle 2. Aktivitäten, die an der Verarbeitung von SQL-Anforderungen für verschiedene DB2CONN-, DB2ENTRY- und BIND-Spezifikationen beteiligt sind (Forts.)

Aktivität	Geschützte Threads		Ungeschützte Threads	
	ACQUIRE(USE) RELEASE(DEAL-LOCATE)		ACQUIRE(USE) RE-LEASE(COMMIT)	ACQUIRE(USE) RE-LEASE(DEALLOCATE)
	Aktivität für den jeweiligen Thread	Aktivität für die jeweilige Transaktion	Aktivität für die jeweilige Transaktion	Aktivität für die jeweilige Transaktion
Alle TS-Sperren anfordern			X	X
Alle Datenbankbeschreibungen (DBD) laden			X	X
Festschreibung: Seitensperren freigeben		X	X	X
TS-Sperren freigeben			X	
CT-Seiten freigeben			X	
Thread beenden:	X		X	X
TS-Sperren freigeben	X		X	X
CT-Seiten freigeben	X			X
Arbeitsspeicher freigeben	X		X	X

Kapitel 3. Verwalten mit CICS Db2

In diesem Abschnitt werden Operationen mit der CICS-Db2-Anschlussfunktion beschrieben.

CICS-Db2-Anschlussfunktion starten

Die CICS-Db2-Anschlussfunktion kann automatisch bei der Initialisierung oder manuell gestartet werden.

Informationen zu diesem Vorgang

Automatische Verbindung bei der CICS-Initialisierung

Sie können CICS mit einer der folgenden Methoden so konfigurieren, dass bei der Initialisierung eine automatische Verbindung zwischen CICS und Db2 aufgebaut wird:

- Durch Angabe von DB2CONN=YES in der Systeminitialisierungstabelle (SIT) oder als SIT-Überschreibung
- Durch Angabe des Programms DFHD2CM0 nach der Anweisung DFHDELIM in Ihrem Systeminitialisierungsparameter PLTPI
- Durch Angabe eines vom Benutzer geschriebenen Programms, das nach der Anweisung DFHDELIM Ihres Systeminitialisierungsparameters PLTPI den Befehl **EXEC CICS SET DB2CONN CONNECTED** absetzt

Manuelle Verbindung

Sie können die Verbindung zwischen CICS und Db2 mithilfe einer der folgenden Methoden manuell aufbauen:

- Mithilfe des Befehls **DSNC STRT**
- Mithilfe des Befehls **CEMT SET DB2CONN CONNECTED**
- Durch Ausführen einer Benutzeranwendung, die den Befehl **EXEC CICS SET DB2CONN CONNECTED** absetzt

CICS-Db2-Anschlussfunktion stoppen

Die CICS-Db2-Anschlussfunktion und damit auch die CICS-Db2-Verbindung können durch Versetzen in den Wartemodus (Quiesce) oder durch Erzwingen (Force) gestoppt werden.

Informationen zu diesem Vorgang

Bei einem Stoppen mit Quiesce wird auf den Abschluss aller CICS-Transaktionen gewartet, die Db2 zurzeit verwenden.

Bei einem erzwungenen Stopp wird für alle CICS-Transaktionen, die Db2 aktuell verwenden, eine erzwungene Bereinigung ausgeführt (Force Purge).

Automatischer Verbindungsabbau bei Beendigung von CICS

Während des Starts der CICS-Db2-Anschlussfunktion wird der taskbezogene CICS-Db2-Benutzerexit (TRUE) mithilfe der Option SHUTDOWN aktiviert. Dies bedeutet, dass CICS beim Beenden automatisch TRUE aufruft und CICS herunterfährt und die CICS-Db2-Verbindung abbaut.

Informationen zu diesem Vorgang

Wird CICS DB2 TRUE während eines gesteuerten CICS-Systemabschlusses aufgerufen, leitet CICS DB2 TRUE für die Anschlussfunktion ein Stoppen mit Quiesce ein. Entsprechend führt eine sofortige Beendigung von CICS zu einer erzwungenen Beendigung der Anschlussfunktion, die von TRUE eingeleitet wird. Wird CICS abgebrochen, wird von TRUE keine Beendigung der Anschlussfunktion eingeleitet.

Manueller Verbindungsabbau

Sie haben verschiedene Möglichkeiten, um die Verbindung zwischen CICS und Db2 abzubauen.

Informationen zu diesem Vorgang

Die Verbindung zwischen CICS und Db2 kann mithilfe einer der folgenden Methoden gestoppt oder getrennt werden:

- Mithilfe des Befehls `DSNC STOP`.

Informationen zum Befehl `DSNC STOP` finden Sie in „[DSNC STOP](#)“ auf Seite 43.

- Mithilfe des Befehls `CEMT SET DB2CONN NOTCONNECTED`.
- Durch Ausführen der von CICS bereitgestellten Transaktion `CDBQ`, die den Befehl `EXEC CICS SET DB2CONN NOTCONNECTED` absetzt.

Die Transaktion `CDBQ` führt das Programm `DFHD2CM2` aus. Die Transaktion kann direkt über das Terminal oder mithilfe des Befehls `EXEC CICS START` ausgeführt werden. Es werden keine Nachrichten an das Terminal ausgegeben. Der CICS-Db2-Adapter gibt jedoch im Rahmen des Systemabschlussverfahrens Nachrichten mit transienten Daten aus.

- Durch Ausführen der von CICS bereitgestellten Transaktion `CDBF`, die den Befehl `EXEC CICS SET DB2CONN NOTCONNECTED FORCE` absetzt.

Die Transaktion `CDBF` führt das Programm `DFHD2CM3` aus. Die Transaktion kann direkt über das Terminal oder mithilfe des Befehls `EXEC CICS STARTED` ausgeführt werden. Es werden keine Nachrichten an das Terminal ausgegeben. Der CICS-Db2-Adapter gibt jedoch im Rahmen des Systemabschlussverfahrens Nachrichten mit transienten Daten aus.

- Durch Ausführen einer Benutzeranwendung, die den Befehl `EXEC CICS SET DB2CONN NOTCONNECTED` absetzt.

Unbestätigte Arbeitseinheiten auflösen

Unbestätigte Arbeitseinheiten (UOWs, Unit of Work) können auftreten, wenn CICS, Db2 oder das gesamte System fehlschlägt, während eine Transaktion einen Synchronisationspunkt verarbeitet, d. h., während der Verarbeitung des Befehls **EXEC CICS SYNCPOINT** oder **EXEC CICS RETURN**. Unbestätigte Arbeitseinheiten werden in der Regel aufgelöst, wenn die Verbindung zwischen CICS und Db2 erneut aufgebaut wird.

Wenn mehrere wiederherstellbare Ressourcen beteiligt sind, verwendet CICS bei dem Versuch, die Arbeitseinheit festzuschreiben, das Protokoll für die zweiphasige Festschreibung. Zwischen der Antwort 'yes' auf den Aufruf `PREPARE` der Phase 1 und vor dem Empfangen des Aufrufs für die Festschreibung oder das Backout zum Zeitpunkt der Phase 2 gilt das Ergebnis der Arbeitseinheit für Db2 als 'unbestätigt'. Wenn zu diesem Zeitpunkt ein Fehler auftritt, gilt die Arbeitseinheit für Db2 weiterhin als 'unbestätigt'; Db2 hat eine unbestätigte Arbeitseinheit und muss CICS zur Resynchronisation auffordern.

In Situationen, an denen nur Db2 und ein einzelnes CICS-System an der Arbeitseinheit beteiligt sind, ist CICS stets der Koordinator. Wenn weitere Parteien beteiligt sind, zum Beispiel über die LU6.2-Kommunikationsverbindung, ist es möglich, dass statt des lokalen CICS-Systems ein fernes CICS-System die Aufgabe des Gesamtkoordinators für die Arbeitseinheit übernimmt. In dieser Situation ist es möglich, dass das Ergebnis der Arbeitseinheit sowohl für Db2 als auch für das lokale CICS-System als 'unbestätigt' gilt. Wenn in dieser Situation ein Fehler auftritt, kann das lokale CICS-System die Arbeitseinheit verzögern; dies hängt von der Definition der Transaktion ab. Die Arbeitseinheit wird dann als 'unbestätigt verzögert' betrachtet.

Unbestätigte Arbeitseinheiten werden in der Regel automatisch aufgelöst, wenn die Verbindung zwischen CICS und Db2 erneut aufgebaut wird. CICS und Db2 tauschen hinsichtlich der unbestätigten Arbeitseinheiten Informationen aus; d. h., CICS informiert Db2 darüber, ob eine Arbeitseinheit zurückgesetzt (Backout) oder festgeschrieben wurde. Wenn eine Arbeitseinheit unbestätigt verzögert wird, wird der Informationsaustausch so lange verzögert, bis der ferne Koordinator CICS über das Ergebnis informiert hat. Wenn Sie jedoch die Einrichtung zum Anhängen von Gruppen verwenden, kann CICS eine erneute Verbindung zu einem anderen Db2-Subsystem herstellen und nicht in der Lage sein, Informationen zu den unbestätigten

Arbeitseinheiten auszutauschen, die von dem zuvor verbundenen Db2-Subsystem aufrecht erhalten wurden. Zum Lösen dieses Problems wird das Attribut RESYNCMEMBER der DB2CONN-Definition verwendet.

Unbestätigte Arbeitseinheiten bei Verwendung des Anhängens von Gruppen auflösen

Wenn Sie das Anhängen von Gruppen verwenden und die Verbindung zwischen CICS und Db2 unterbrochen ist, stellt CICS möglicherweise keine erneute Verbindung zum selben Db2-Subsystem her – möglicherweise wird aus der Gruppe der Db2-Subsysteme mit gemeinsamer Datennutzung ein anderes Member ausgewählt. Werden also vom ersten Db2-Subsystem, zu dem CICS eine Verbindung hergestellt hat, unbestätigte Arbeitseinheiten gehalten, können diese nicht behoben werden.

Zum Beheben dieses Problems verwaltet CICS ein Protokoll darüber, zu welchem Mitglied der Db2-Gruppe mit gemeinsamer Datennutzung zuletzt eine Verbindung hergestellt wurde; dieses Protokoll wird katalogisiert und während Warmstarts, Notfallstarts und Kaltstarts (jedoch nicht während Erststarts) beibehalten. Während des Herstellens einer Verbindung oder des erneuten Verbindens zu Db2 prüft die CICS-Db2-Anschlussfunktion dieses Protokoll, um festzustellen, ob zu dem Member einer Db2-Gruppe mit gemeinsamer Datennutzung, mit dem zuletzt eine Verbindung hergestellt wurde, Informationen zu ausstehenden Arbeitseinheiten vorhanden sind und geht wie folgt vor:

- Wenn keine Informationen zu ausstehenden Arbeitseinheiten vorhanden sind und wenn von CICS keinen Wiederanlauf nach Systemabsturz ausgeführt wurde, funktioniert das Anhängen von Gruppen normal und es werden aktive Member der Gruppe mit gemeinsamer Datennutzung für die Verbindung ausgewählt. Wenn von CICS ein Wiederanlauf nach Systemabsturz ausgeführt wurde, versucht CICS eine Verbindung zu dem Member herzustellen, mit dem es zuvor verbunden war.
- Wenn Informationen zu ausstehenden Arbeitseinheiten vorhanden sind, hängt die nächste Aktion von der Einstellung ab, die Sie für das Attribut RESYNCMEMBER der DB2CONN-Definition ausgewählt haben.
 - Wenn für RESYNCMEMBER die Einstellung YES gewählt wurde, die angibt, dass eine Resynchronisation mit dem Member der Db2-Gruppe mit gemeinsamer Datennutzung erforderlich ist, das zuletzt aufgezeichnet wurde, ignoriert CICS die Einrichtung zum Anhängen von Gruppen und die CICS-Db2-Anschlussfunktion wartet, bis sie zu dem Member der Db2-Gruppe mit gemeinsamer Datennutzung, mit dem sie zuletzt verbunden war, eine erneute Verbindung herstellen kann, um die unbestätigten Arbeitseinheiten aufzulösen. Unbestätigt verzögerte Arbeitseinheiten sind in diesen Prozess nicht eingeschlossen, da CICS selbst diese Arbeitseinheiten zu diesem Zeitpunkt nicht auflösen kann. Die Resynchronisation für diese Arbeitseinheiten findet statt, wenn CICS eine Resynchronisation mit dem zugehörigen fernen Koordinator ausgeführt hat.
 - Wenn für RESYNCMEMBER die Einstellung NO gewählt wurde (weil Sie möglicherweise so schnell wie möglich eine erneute Verbindung wünschen), führt CICS einmalig einen Versuch durch, erneut eine Verbindung zu dem Member der Db2-Gruppe mit gemeinsamer Datennutzung herzustellen, das zuletzt aufgezeichnet wurde. Ist dieser Versuch erfolgreich, können die unbestätigten Arbeitseinheiten (mit Ausnahme der unbestätigt verzögerten Arbeitseinheiten) aufgelöst werden. Ist dies nicht erfolgreich, verwendet CICS das Anhängen von Gruppen, um eine Verbindung zu einem beliebigen aktiven Member der Db2-Gruppe mit gemeinsamer Datennutzung herzustellen; außerdem wird der Warnhinweis DFHDB2064 mit dem Inhalt ausgegeben, dass zum zuletzt aufgezeichneten Member möglicherweise nicht aufgelöste unbestätigte Arbeitseinheiten vorhanden sind.

Informationen zu Einstellungen für RESYNCMEMBER finden Sie in DB2CONN-Ressourcen. Die Option RESYNCMEMBER kann auch mithilfe des Befehls **SET DB2CONN RESYNCMEMBER** festgelegt werden.

Unbestätigte Arbeitseinheiten mithilfe eines Light-Neustarts von Db2 auflösen

CICS unterstützt die erweiterte Db2-Funktionalität für Light-Neustarts, die in Db2 Version 8 bereitgestellt wird.

Der Modus für den Light-Neustart ist für einen systemübergreifenden Neustart im Fall eines MVS-Systemausfalls bestimmt, bei dem die CICS-Systeme für die Verwendung des Anhängens von Gruppen (siehe [Einrichtung zum Anhängen von Db2-Gruppen verwenden](#)) konfiguriert sind. In Db2 Version 7 ließ die Db2-Funktionalität für Light-Neustarts für ein Member einer Db2-Gruppe mit gemeinsamer Datennutzung zu, dass ein Neustart mit minimalen Speicheranforderungen ausgeführt wurde, beibehaltene Sperren freigegeben wurden und anschließend eine normale Beendigung erfolgte. Dies galt jedoch nur für momentan

ausgeführte Arbeitseinheiten, nicht für unbestätigte Arbeitseinheiten. Für DB2 Version 8 wurde diese Funktionalität so erweitert, dass sie auch unbestätigte Arbeitseinheiten umfasst. Dadurch kann ein CICS-System eine Verbindung zu einem mit Light-Neustart ausgestatteten Db2-Subsystem herstellen, um unbestätigte Arbeitseinheiten zu resynchronisieren. Diese Funktionalität ist insbesondere deshalb hilfreich, da Db2 die Peerwiederherstellung nicht unterstützt; das heißt, ein Db2-Subsystem kann unbestätigte Arbeitseinheiten nicht für ein anderes Db2-Subsystem resynchronisieren.

Wenn in einem typischen Szenario eine logische MVS-Partition fehlschlägt, kann ein systemübergreifender Neustart initiiert werden, zu dem das erneute Starten des fehlgeschlagenen CICS-Systems in einer anderen logischen Partition gehört; außerdem muss das fehlgeschlagene Db2-Subsystem in dieser logischen Partition im Light-Neustart-Modus hochgefahren werden. Angenommen, die CICS-Systeme wurden für die Verwendung des Anhängens von Gruppen konfiguriert, wobei in der DB2CONN-Definition RESYNC-MEMBER(YES) und STANDBYMODE(RECONNECT) festgelegt wurden, und in Db2 sind unbestätigte Arbeitseinheiten ausstehend. Das mit Light-Neustart ausgestattete Db2-Subsystem wird initialisiert, gibt beibehaltene Sperren für eventuell vorhandene, momentan ausgeführte Arbeitseinheiten frei und wartet dann bezüglich der unbestätigten Arbeitseinheiten auf die Resynchronisation mit CICS. Jedes CICS-System wird initialisiert und erkennt, ob ausstehende Arbeitseinheiten vorhanden sind und ob RESYNCMEMBER(YES) angegeben wurde. Wenn beide Bedingungen wahr sind, ignoriert das CICS-System das Anhängen von Gruppen und stellt erneut eine Verbindung zu dem zuletzt verwendeten Db2-Subsystem her, bei dem es sich um das mit Light-Neustart ausgestattete Db2-Subsystem handelt. Die unbestätigten Arbeitseinheiten werden nun resynchronisiert, jedoch können keine neue Transaktionen auf Db2 zugreifen. Wenn alle unbestätigten Arbeitseinheiten in dem mit Light-Neustart ausgestatteten Db2-Subsystem aufgelöst wurden, wird es beendet. Da die CICS-Systeme mit STANDBYMODE(RECONNECT) definiert sind, werden sie bei Beendigung des mit Light-Neustart ausgestatteten Db2-Subsystems in den Standby-Modus versetzt und versuchen, erneut eine Verbindung zu Db2 herzustellen. Da nun alle unbestätigten Arbeitseinheiten aufgelöst sind, findet RESYNCMEMBER keine Anwendung und das Anhängen von Gruppen kann verwendet werden. Die CICS-Systeme stellen erneut eine Verbindung zu einem normalen, aktiven Db2-Subsystem her.

Wiederherstellung von Resynchronisationsinformationen für unbestätigte Arbeitseinheiten

CICS verwaltet im Systemprotokoll für die Resynchronisation erforderliche Informationen zu Arbeitseinheiten. Resynchronisationsinformationen werden von CICS während Warmstarts, Notfallstarts und Kaltstarts beibehalten.

Die Resynchronisationsinformationen sind nach Ausführen eines Erststarts von CICS nicht mehr vorhanden, da das Systemprotokoll initialisiert wird und alle Informationen verloren gehen, weil sämtliche Informationen zu vorherigen Arbeitseinheiten gelöscht werden. Es sollte nur selten erforderlich sein, dass Sie für CICS einen Erststart ausführen müssen. Führen Sie einen Kaltstart durch, wenn Sie Ressourcen aus der CICS-Systemdefinitionsdatei (CSD) erneut installieren möchten; dadurch können alle Resynchronisationsinformationen wiederhergestellt werden. Ein Erststart sollte insbesondere vermieden werden, wenn ein vorheriger kontrollierter Systemabschluss von CICS zur Ausgabe der Nachricht DFHRM0131 geführt hat, mit der angegeben wird, dass die Resynchronisation aussteht.

Ist für Db2 eine Resynchronisation erforderlich und für CICS wird ein Erststart ausgeführt, wird bei erneutem Herstellen einer Verbindung zwischen CICS und Db2 die Nachricht DFHDB2001 für jede Arbeitseinheit ausgegeben, deren Resynchronisation fehlgeschlagen ist, und eine solche Arbeitseinheit muss in Db2 mithilfe des Befehls **DB2 RECOVER INDOUBT** resynchronisiert werden.

CICS-Db2-Anschlussfunktion verwalten

Sie können den Status der Verbindung zwischen CICS und Db2 mithilfe der CEMT-Befehle und der von der CICS-Db2-Anschlussfunktion bereitgestellten Befehle verwalten. Dieselbe, von CEMT bereitgestellte Funktion ist auch über EXEC CICS INQUIRE- und SET-Befehle verfügbar.

Informationen zu diesem Vorgang

Im Folgenden finden Sie eine Übersicht über die verfügbaren Funktionen:

CEMT INQUIRE und SET DB2CONN

Mithilfe dieser Befehle können der globale Status der Verbindung und die zugehörigen Attribute sowie Attribute von Pool-Threads und Befehlsthreads abgefragt und festgelegt werden. Weitere Informationen finden Sie unter [CEMT INQUIRE DB2CONN](#) und [CEMT SET DB2CONN](#).

CEMT INQUIRE und SET DB2ENTRY

Mithilfe dieser Befehle können die Attribute einer bestimmten DB2ENTRY-Ressource, die Threads definiert, die von bestimmten Transaktionen oder einer Gruppe von Transaktionen verwendet werden sollen, abgefragt und festgelegt werden. Weitere Informationen finden Sie unter [CEMT INQUIRE DB2ENTRY](#) und [CEMT SET DB2ENTRY](#).

CEMT INQUIRE und SET DB2TRAN

Ermitteln Sie mithilfe dieser Befehle und ändern Sie ggf., welche Transaktions-IDs welche DB2ENTRY-Ressourcen verwenden. Weitere Informationen finden Sie unter [CEMT INQUIRE DB2TRAN](#) und [CEMT SET DB2TRAN](#).

DSNC DISC (Verbindung trennen)

Dieser CICS-Db2-Anschlussbefehl kann verwendet werden, um zu bewirken, dass aktuell verbundene Threads beendet werden, sobald sie nicht mehr von einer Transaktion verwendet werden. Bei aktiven Threads ist dies der Zeitpunkt, an dem die Transaktion den Thread freigibt; in der Regel am Synchronisationspunkt. Geschützte Threads sind Threads, die aktuell nicht von einer Transaktion verwendet werden und in der Regel freigegeben werden, wenn sie nach zwei Bereinigungszyklen für geschützte Threads nicht erneut verwendet werden. Der Befehl DSNC DISCONNECT stellt die Bereinigungszyklen zurück und bewirkt, dass sie sofort beendet werden. Weitere Informationen zum Befehl DSNC DISCONNECT finden Sie in „DSNC DISCONNECT“ auf Seite 36.

DSNC DISP (anzeigen)

Dieser CICS-Db2-Anschlussbefehl kann verwendet werden, um den Status der aktiven Threads für einen bestimmten Plan, für eine bestimmte Transaktion oder für alle Pläne und Transaktionen anzuzeigen. Weitere Informationen hierzu finden Sie in „DSNC DISPLAY“ auf Seite 37.

Mit dem Befehl DSNC DISP können auch CICS-Db2-Statistiken in einem CICS-Terminal angezeigt werden. Diese Statistiken sind nur ein Subset der CICS-Db2-Statistiken, die sie mithilfe der Befehle CICS COLLECT STATISTICS und PERFORM STATISTICS abrufen können. Diese Statistiken unterliegen denselben Zurücksetzungskriterien wie alle CICS-Statistiken.

Weitere Informationen zum Befehl DSNC DISP STAT finden Sie in „Ausgabe von DISPLAY STATISTICS“ auf Seite 39. Weitere Informationen zu verfügbaren vollständigen CICS-Db2-Statistiken finden Sie in CICS-Db2-Statistik unter "Referenz".

DSNC MODI (ändern)

Dieser CICS-Db2-Anschlussbefehl kann verwendet werden, um das Ziel zu ändern, an das nicht erwartete Nachrichten gesendet werden, sowie um die Anzahl der für eine DB2ENTRY-Ressource zulässigen Threads oder die Anzahl der Pool-Threads oder Befehlsthreads zu ändern. Diese Funktion wird durch die beschriebenen CEMT-Befehle ersetzt, die es ermöglichen, diese Attribute und alle anderen Attribute von DB2CONN-, DB2ENTRY- und DB2TRAN-Ressourcen zu ändern. Weitere Informationen hierzu finden Sie in „DSNC MODIFY“ auf Seite 41.

Db2-Befehle eingeben

Sobald die Verbindung zwischen CICS und Db2 hergestellt ist, können von CICS berechnete Benutzer die Transaktion DSNC zum Weiterleiten von Befehlen an das Db2-Subsystem verwenden.

Informationen zu diesem Vorgang

Befehle, die die Transaktion DSNC zum Weiterleiten von Befehlen an das Db2-Subsystem verwenden, sind wie folgt definiert:

```
DSNC -DB2command
```

Der Befehl wird zur Verarbeitung an Db2 weitergeleitet. Db2 überprüft, dass der Benutzer zum Absetzen des eingegebenen Befehls berechtigt ist. Antworten werden an den CICS-Benutzer zurückgeleitet, von

dem die Befehle stammen. Das Befehlserkennungszeichen (CRC, Command Recognition Character) '-' muss verwendet werden, um Db2-Befehle von Befehlen der CICS-Db2-Anschlussfunktion zu unterscheiden. Dieses Befehlserkennungszeichen wird nicht verwendet, um das Db2-Subsystem anzugeben, an das der Befehl zu senden ist. Der Befehl wird an das Db2-Subsystem gesendet, mit dem CICS aktuell verbunden ist. Abbildung 4 auf Seite 32 zeigt die Befehle der CICS-Db2-Anschlussfunktion. Für diese ist die CICS-Berechtigung erforderlich, damit die Transaktion DSNC und die Db2-Befehle verwendet werden können. Weitere Informationen zu DSNC -DB2COMMAND finden Sie in „Befehle an Db2 mithilfe der Transaktion DSNC absetzen“ auf Seite 34.

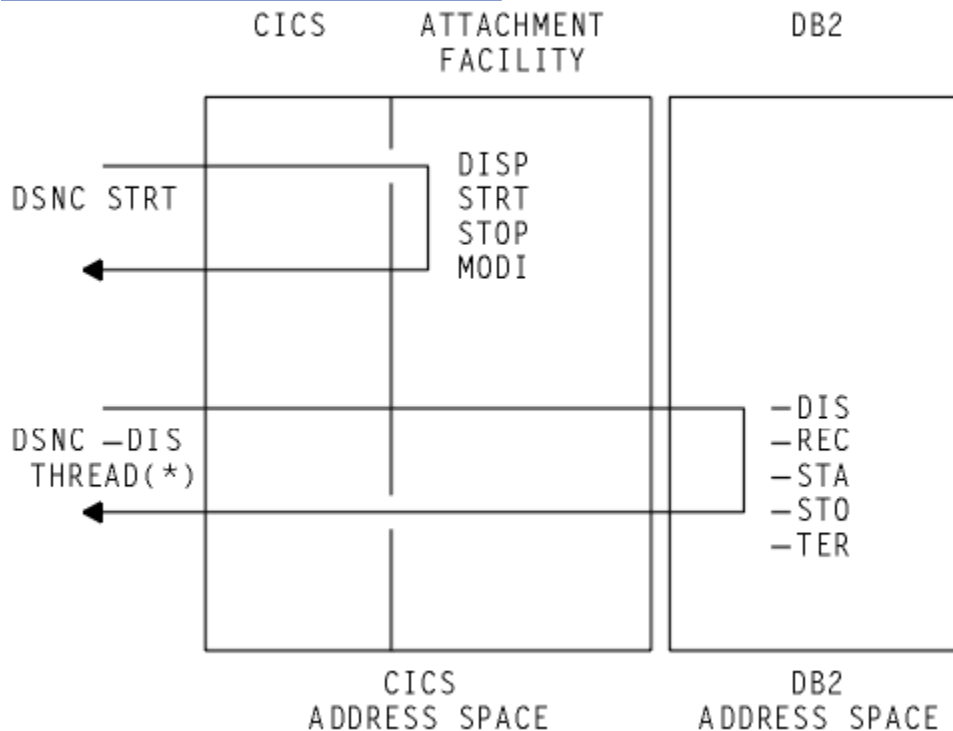


Abbildung 4. Beispiele für Befehle der CICS-Db2-Anschlussfunktion und für einige Db2-Befehle

SMF für Abrechnung, Statistik und Optimierung von Db2 starten

Db2 erzeugt mit seiner Instrumentierungsfunktion bei jeder Beendigung eines CICS-Db2-Threads und bei jeder CICS-Db2-Anmeldung einen SMF-Abrechnungsdatensatz (SMF – System Management Facility, Systemverwaltungsfunktion) mit dem Typ 101. Db2 erzeugt auch einen SMF-Datensatz mit dem Typs 100, der Db2-Statistiken auf Basis eines Db2-Subsystems enthalten soll. Datensätze für Leistung und globale Traces werden als SMF-Datensätze mit dem Typ 102 erzeugt.

Informationen zu diesem Vorgang

Die von Db2 erzeugten SMF-Datensätze können mithilfe der folgenden Möglichkeiten an eine SMF-Datei übertragen werden:

- Indem zum Zeitpunkt der Db2-Installation angegeben wird, dass Abrechnungs- und/oder Statistikdaten erforderlich sind.

Dazu wird für MON, SMFACCT oder SMFSTAT des Initialisierungsmakros DSN6SYSP die Einstellung YES festgelegt. Weitere Details finden Sie in [Produktdokumentation zu Db2 for z/OS](#).

- Indem SMF zum Schreiben der Datensätze aktiviert wird.

Fügen Sie dem vorhandenen Eintrag eines SMF-Elements (SMFPRMxx) in SYS1.PARMLIB Einträge für Datensätze des Typs 100, 101 und 102 hinzu.

- Indem Sie einen Db2-Trace starten.

Das Starten eines Db2-Trace kann zum Zeitpunkt des Db2-Starts vorgenommen werden, indem die Parameter des Initialisierungsmakros DSN6SYSP festgelegt werden oder indem der Db2-Befehl -START

TRACE unter genauer Angabe des zu startenden Trace und der zu startenden Klassen verwendet wird. Mithilfe der letztgenannten Möglichkeit wird die regelmäßige Aufzeichnung von Abrechnungs-, Statistik- und Leistungsdaten gestartet.

Db2 erzeugt aus den aufgezeichneten Daten keine Berichte für Abrechnung, Überwachung oder Leistung. Um eigene Berichte zu erzeugen, können Sie zur Verarbeitung dieser Daten eigene Programme schreiben oder das Lizenzprogramm Db2 Performance Monitor (DB2PM) verwenden.

GTF für Abrechnung, Statistik und Optimierung von Db2 starten

Anstatt Abrechnungs-, Statistik- und Leistungsdaten für SMF zu erfassen (oder zusätzlich dazu) können Sie diese Daten mithilfe von Db2 an eine allgemeine Tracefunktion senden (GTF – Generalized Trace Facility, Allgemeine Tracefunktion). Db2 stellt die Befehle -START TRACE und -STOP TRACE bereit, die Sie mithilfe der Transaktion DSNB von einem CICS-Terminal absetzen können.

Informationen zu diesem Vorgang

Verwenden Sie die folgenden Befehle mit der Transaktion DSNB, um den Tracebereich, das Traceziel und die Traceeinschränkungen anzugeben:

Tracebereich

Legen Sie hierfür GLOBAL für die Db2-Subsystemaktivität fest, PERF für die Leistungsaktivität, ACCTG für die Abrechnungsaktivität, STAT für die Statistikaktivität oder MONITOR für die Überwachungsaktivität.

Traceziel

GTF im Hauptspeicher oder SMF.

Traceeinschränkungen

Bestimmte Pläne, Berechtigungs-IDs, Klassen, eine bestimmte Position und bestimmte Ressourcenmanager.

Von CICS bereitgestellte Transaktionen für CICS Db2

CICS stellt eine Reihe von Transaktionen bereit, die Sie verwenden können, um die Schnittstelle mithilfe von Db2 zu verwalten.

CEMT

Die folgenden CEMT-Optionen sind verfügbar:

- [CEMT INQUIRE DB2CONN](#)
- [CEMT SET DB2CONN](#)
- [CEMT INQUIRE DB2ENTRY](#)
- [CEMT SET DB2ENTRY](#)
- [CEMT INQUIRE DB2TRAN](#)
- [CEMT SET DB2TRAN](#)

Die funktional entsprechenden Befehle **EXEC CICS INQUIRE** und **EXEC CICS SET** werden in [Systembefehle](#) beschrieben.

Optionen für PLAN und PLANEXITNAME sind im Befehl **INQUIRE DB2TRAN** verfügbar; Sie können also mit einem einzigen Schritt herausfinden, welcher Plan von einer bestimmten Transaktion oder Gruppe von Transaktionen verwendet wird oder welche Transaktionen einen bestimmten Plan verwenden.

DSNB

Die Transaktion DSNB kann verwendet werden, um Folgendes auszuführen:

- Db2-Befehle über ein CICS-Terminal eingeben.
- Bewirken, dass Threads abgebrochen werden, wenn Sie freigegeben werden (DSNB DISCONNECT).

- Informationen zu Transaktionen anzeigen, die die CICS-Db2-Schnittstelle verwenden, und Statistikdaten anzeigen (DSNC DISPLAY).
- Ziele für nicht erwartete Nachrichten ändern und die Anzahl der aktiven Threads ändern, die von einer DB2ENTRY-Ressource, vom Pool oder für Befehle verwendet werden (DSNC MODIFY).
- CICS-Db2-Schnittstelle beenden (DSNC STOP).
- CICS-Db2-Schnittstelle starten (DSNC STRT).

Die Transaktion DSNC führt das Programm DFHD2CM1 aus, das sowohl Befehle der CICS-Db2-Anschlussfunktion als auch Db2-Befehle verarbeitet. Sie können Db2-Befehle von Befehlen der CICS-Db2-Anschlussfunktion aufgrund des Bindestrichs (-) unterscheiden, der mit Db2-Befehlen eingegeben wird. Dieses Zeichen ist nicht ein Erkennungszeichen des Db2-Subsystems, sondern ein Befehlserkennungszeichen. Es handelt sich immer um einen Bindestrich (-), unabhängig von dem Zeichen, durch das das Db2-Subsystem definiert wird, da CICS jeweils nur eine Verbindung zu einem einzigen Db2-Subsystem aufbauen kann. Es ist nicht erforderlich, andere CICS-Erkennungszeichen für Db2-Subsysteme zu verwenden, und daher wird nur das Standardzeichen - verwendet.

DFHD2CM1 kann auch durch andere Transaktionen als DSNC aktiviert werden. Auf diese Weise können Sie jedem Befehl der CICS-Db2-Anschlussfunktion und jedem Db2-Befehl einen anderen Transaktionscode zuordnen. Dadurch können Sie auf jeden Befehl eine andere Sicherheitsstufe anwenden.

Alternative Transaktionsdefinitionen für CICS Db2, bereitgestellt in Beispielgruppe DFH\$DB2

Alternative Transaktionsdefinitionen für Befehle der CICS-Db2-Anschlussfunktion werden mit folgenden Namen bereitgestellt:

- DISC
- DISP
- STRT
- STOP
- MODI

Alternative Transaktionsdefinitionen für Db2-Befehle werden auch mit folgenden Namen bereitgestellt:

-ALT	-CAN	-ARC
-DIS	-MOD	-REC
-RES	-SET	-STA
-STO	-TER	

Befehle an Db2 mithilfe der Transaktion DSNC absetzen

Die Transaktion DSNC ermöglicht Ihnen, Db2-Befehle über CICS einzugeben.

Umgebung

Dieser Befehl kann nur über ein CICS-Terminal abgesetzt werden.

Syntax

Syntax von DSNC

```

➡➡ DSNC ──────────── db2-command ➡➡
      |               |
      └── destination ─┘

```

Abkürzung

Sie können DSNCR im Db2-Befehl auslassen, wenn die entsprechende Transaktionsdefinition aus der CICS-Db2-Beispielgruppe DFH\$DB2 installiert wird. Beispielsweise, wenn Sie die Transaktionsdefinition -DIS installieren.

```
DSNCR -DIS THD(*)
```

kann abgekürzt werden in:

```
-DIS THD(*)
```

Die CICS-Db2-Beispielgruppe DFH\$DB2 enthält die folgenden Transaktionsdefinitionen zum Absetzen von Db2-Befehlen:

-ALT	-ARC	-CAN
-DIS	-MOD	-REC
-RES	-SET	-STA
-STO	-TER	

Berechtigung

Zum Absetzen von Db2-Befehlen mithilfe von DSNCR ist von CICS keine weitere Berechtigung außer der CICS-Sicherheitsprüfung 'Transaction-attach Security' erforderlich, die zum Ausführen der Transaktion DSNCR erforderlich ist. Erforderlich sind jedoch Db2-Zugriffsrechte. Weitere Informationen zur CICS-Sicherheit finden Sie in [Sicherheit in einer CICS-DB2-Umgebung](#).

Parameterbeschreibung

destination

Gibt ein weiteres Terminal an, an dem Anzeigeeinformationen empfangen werden können. Dabei muss es sich um ein gültiges Terminal handeln, das für CICS definiert ist und vom Basic Mapping Support (BMS) unterstützt wird.

db2-command

Gibt den genauen Db2-Befehl an, den Sie über ein CICS-Terminal eingeben möchten. Dem Befehl muss ein Bindestrich vorangestellt werden.

Hinweis zur Verwendung

Screen scrolling

Die BMS-Unterstützung für den Seitenwechsel kann verwendet werden, um das Blättern in DSNCR-Db2-Befehlen an Ihrem Terminal zu unterstützen. Geben Sie beispielsweise SIT-Schlüsselwörter an:

```
PGCHAIN=X/  
PGCOPY=C/  
PGPURGE=T/  
PGRET=P/  
SKRPF7='P'  
SKRPF8='N'
```

Weitere Informationen zu SIT-Schlüsselwörtern und -Parametern finden Sie in [Beschreibungen und Zusammenfassungen der Systeminitialisierungsparameter](#).

Beispiel

Setzen Sie den Db2-Befehl -DISPLAY THREAD über ein CICS-Terminal ab, um Threads für eine CICS-Anwendung mit der Anwendungs-ID IYK4Z2G1 anzuzeigen.

```
DSNC -DISPLAY THREAD(IYK4Z2G1)
```

```
DSNV401I : DISPLAY THREAD REPORT FOLLOWS -
DSNV402I : ACTIVE THREADS -
NAME      ST A  REQ ID          AUTHID  PLAN      ASID TOKEN
IYK4Z2G1  N    3              JTILLI1  00BA      0
IYK4Z2G1  T    3 ENTRXC060001 CICSUSER TESTC06 00BA    16
IYK4Z2G1  T    3 POOLXP050002 CICSUSER TESTP05 00BA    17
IYK4Z2G1  T    6 COMDDSN0003 JTILLI1  00BA    18
DISPLAY ACTIVE REPORT COMPLETE
DSN9022I : DSNVDT '-DIS THREAD' NORMAL COMPLETION
DFHDB2301 07/09/98 13:36:36 IYK4Z2G1 DSNC DB2 command complete.
```

Abbildung 5. Beispielausgabe des Befehls DSNC -DISPLAY

DSNC DISCONNECT

Verwenden Sie den Befehl **DSNC DISCONNECT** der CICS-Db2-Anschlussfunktion, um die Verbindung für Threads zu trennen. Dieser Befehl bietet manuelle Steuerung für die Freigabe von Ressourcen, die von normalen Transaktionen gemeinsam genutzt werden, sodass Prozesse für besondere Zwecke wie Db2-Dienstprogramme exklusiven Zugriff auf die Ressourcen haben können.

Umgebung

Dieser Befehl kann nur über ein CICS-Terminal abgesetzt werden.

Syntax

Syntax von DISC

➡ DSNC DISConnect — plan-name →

Abkürzung

DSNC DISC oder DISC (bei Verwendung der Transaktion DISC über die CICS-Db2-Beispielgruppe DFH \$DB2).

Berechtigung

Der Zugriff auf diesen Befehl kann mithilfe der CICS-Sicherheitsprüfung 'Transaction-attach Security' (auf die Transaktionen selbst bezogene Sicherheitsprüfungen) für die Transaktion DSNC und mithilfe der CICS-Prüfung der Befehlssicherheit für die Ressource DB2CONN gesteuert werden. Für diesen Befehl ist der Zugriff READ (Lesezugriff) erforderlich.

Weitere Informationen zur CICS-Sicherheit finden Sie in [Sicherheit in einer CICS-DB2-Umgebung](#).

Parameterbeschreibung

plan-name

Gibt einen gültigen Anwendungsplan an.

Hinweise zur Verwendung

Erstellung von Threads verhindern

Der Befehl DSNB DISCONNECT verhindert nicht, dass Threads im Namen von Transaktionen erstellt werden. Der Befehl bewirkt nur, dass aktuell verbundene Threads beendet werden, sobald sie nicht mehr von einer Transaktion verwendet werden. Zum Unterbrechen einer Transaktion und zum schnelleren Abbrechen eines Threads können Sie den Db2-Befehl CANCEL THREAD verwenden.

Sie können die Transaktionen, die einer bestimmten Plan-ID in CICS zugeordnet sind, mit der Einstellung MAXACTIVE für TRANCLASS stoppen. Dadurch wird verhindert, dass neue Instanzen der Transaktion eine erneute Erstellung eines Threads bewirken.

Alternative für geschützte Threads

Möglicherweise möchten Sie die Zuordnung eines Plans aufheben, um ihn erneut zu binden oder um ein Dienstprogramm für die Datenbank auszuführen. Wenn Sie einen geschützten Thread verwenden, sollten Sie zum Senden aller Threads an den Pool statt DSNB DISCONNECT besser EXEC CICS SET DB2ENTRY(*Eintragsname*) THREADLIMIT(0) oder DSNB MODIFY verwenden. Der geschützte Thread wird nach zwei Bereinigungszyklen von allein beendet. Informationen finden Sie im Attribut PURGE-CYCLE von DB2CONN.

Beispiel

Trennen Sie die Verbindung von aktiven und geschützten Threads für Plan TESTP05:

```
DSNB DISC TESTP05
```

```
DFHDB2021 07/09/98 13:46:29 IYK4Z2G1 The disconnect command is complete.
```

Abbildung 6. Beispielausgabe des Befehls DSNB -DISCONNECT

DSNB DISPLAY

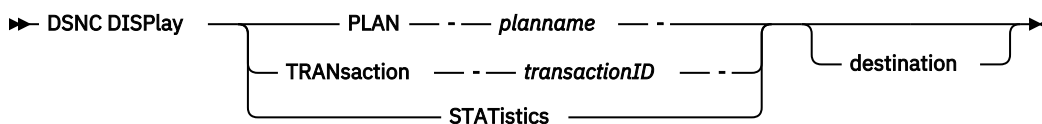
Verwenden Sie den Befehl der CICS-Db2-Anschlussfunktion **DSNB DISPLAY**, um Informationen zu aktiven CICS-Db2-Threads und die sie verwendende CICS-Transaktion anzuzeigen oder um statistische Informationen zu DB2ENTRY-Objekten und zur Ressource DB2CONN anzuzeigen.

Umgebung

Dieser Befehl kann nur über ein CICS-Terminal abgesetzt werden.

Syntax

Syntax von DISPLAY



Abkürzung

DSNB DISP oder DISP (bei Verwendung der Transaktion DISP über die CICS-Db2-Beispielgruppe DFH \$DB2).

Berechtigung

Der Zugriff auf diesen Befehl kann mithilfe der CICS-Sicherheitsprüfung 'Transaction-attach Security' (auf die Transaktionen selbst bezogene Sicherheitsprüfungen) für die Transaktion DSNB und mithilfe der CICS-Prüfung der Befehlssicherheit für die Ressource DB2CONN gesteuert werden. Für diesen Befehl ist der Zugriff READ (Lesezugriff) erforderlich.

Weitere Informationen zur CICS-Sicherheit finden Sie in [Sicherheit in einer CICS-DB2-Umgebung](#).

Parameterbeschreibung

PLAN *planname*

Zeigt Informationen zu den Threads für *planname* an. *Planname* ist der Name eines gültigen Plans, zu dem Informationen angezeigt werden.

Wenn Sie *planname* nicht angeben (oder wenn Sie einen Stern (*) angeben), werden Informationen zu allen aktiven Threads angezeigt.

Der folgende Befehl zeigt beispielsweise Informationen zu allen aktiven CICS-Db2-Plan-IDs an und sendet die Anzeigeeinformationen an ein weiteres Terminal mit der Bezeichnung MTO2.

```
DSNC DISP PLAN * MTO2
```

Tran *transaktions-id*

Enthält die Transaktions-ID, zu der Threadinformationen angezeigt werden. *transaktions-id* ist eine gültige Transaktions-ID, zu der Threadinformationen angezeigt werden.

Wenn Sie keine Transaktions-ID angeben, werden Informationen zu allen aktiven Threads angezeigt.

Der folgende Befehl zeigt beispielsweise Informationen zu allen aktiven CICS-Db2-Transaktionen an.

```
DSNC DISP TRAN
```

STAT

Zeigt die statistischen Zähler an, die den einzelnen CICS-Db2-Ressourcendefinitionen zugeordnet sind. Die Zähler betreffen die Verwendung der verfügbaren Verbindungen zwischen der CICS-Db2-Anschlussfunktion und Db2.

Beachten Sie, dass Sie eine detailliertere Gruppe von CICS-Db2-Statistikdaten mithilfe der standardmäßigen Schnittstellen für CICS-Statistiken abrufen können, beispielsweise mit den Befehlen **EXEC CICS COLLECT STATISTICS** und **EXEC CICS PERFORM STATISTICS** oder mithilfe des Beispielsprogramms DFHOSTAT.

destination

Geben Sie ein anderes Terminal an, um die angeforderten Informationen zu erhalten. *destination* ist die ID eines anderen Terminals, das für den Erhalt der angeforderten Anzeigeeinformationen verwendet wird. Dabei muss es sich um ein gültiges Terminal handeln, das für CICS definiert ist und vom Basic Mapping Support (BMS) unterstützt wird.

Da dem optionalen Ziel im Befehl manchmal ein optionaler Planname oder eine optionale Transaktions-ID vorangestellt ist, muss jeder Parameter eindeutig und separat als Name oder als Terminal-ID identifizierbar sein. Wird nur ein einziger Parameter eingegeben, wird dieser zunächst geprüft, um festzustellen, ob es sich um einen Plannamen oder eine Transaktions-ID handelt; anschließend wird er als Ziel ausgewählt. Um eine Zeichenfolge zu verwenden, bei der es sich sowohl um einen Plannamen bzw. eine Transaktions-ID als auch um eine gültige Terminal-ID handelt, müssen Sie sowohl den Parameter für den Namen als auch den Parameter für das Ziel verwenden, damit die erforderlichen Informationen am erforderlichen Terminal angezeigt werden.

Wenn für den Erhalt der angeforderten Anzeigeeinformationen ein alternatives Ziel angegeben wird, wird folgende Nachricht an das anfordernde Terminal gesendet:

```
DFHDB2032 date time applid alternate destination display command complete
```

Hinweise zur Verwendung

Setzen Sie diesen Befehl in CICS ab, während die CICS-Db2-Anschlussfunktion aktiv ist, das Db2-Subsystem jedoch nicht, wird eine Statistikanzeige erzeugt, in der nicht deutlich erkennbar angegeben ist, dass das Subsystem nicht aktiv ist.

Im CICS-Nachrichtenprotokoll wird die Nachricht DFHDB2037 angezeigt, die angibt, dass die Anschlussfunktion auf den Start von Db2 wartet.

Ausgabe von DISPLAY PLAN oder TRAN

Die Ausgabe des Befehls **DSNC DISPLAY PLAN** oder **DSNC DISPLAY TRANSACTION** schließt den Namen des DB2ENTRY-Objekts, *POOL oder *COMMAND für den Aufruf des Befehls DSNC ein.

In [Abbildung 7 auf Seite 39](#) wird ein Beispiel für die Ausgabe des Befehls **DSNC DISPLAY** (PLAN oder TRANSACTION) gezeigt. Die Ausgabe zeigt für jeden erstellten Thread den Namen des DB2ENTRY-Objekts, *POOL für den Pool oder *COMMAND für den Aufruf des Befehls DSNC.

```
DFHDB2013 07/09/98 15:26:47 IYK4Z2G1 Display report follows for threads
accessing DB2 DB3A
DB2ENTRY S PLAN PRI-AUTH SEC-AUTH CORRELATION TRAN TASK UOW-ID
*POOL A TESTC05 JTILLI1 POOLXC050001 XC05 01208 AEEEC0321ACDCE00
XC06 * TESTC06 JTILLI1 ENTRXC060003 XC06 01215 AEEEC0432F8EFE01
XP05 A TESTP05 JTILLI1 ENTRXP050002 XP05 01209 AEEEC03835230C00
XP05 I TESTP05 JTILLI1 ENTRXP050004
DFHDB2020 07/09/98 15:26:47 IYK4Z2G1 The display command is complete.
```

Abbildung 7. Beispielausgabe des Befehls DSNC DISPLAY (PLAN oder TRANSACTION)

Die Spalte mit der Bezeichnung 'S' bezeichnet den Status des Threads und sie kann folgende Werte annehmen:

Der Thread ist in einer Arbeitseinheit aktiv und wird aktuell in Db2 ausgeführt.

A

Der Thread ist in einer Arbeitseinheit aktiv, wird jedoch aktuell nicht in Db2 ausgeführt.

I

Der Thread ist inaktiv; es handelt sich um einen geschützten Thread, der auf neue Arbeit wartet.

Das dem Thread zugeordnete Element PLAN wird angezeigt (für Befehlsthreads gibt es keinen Plan).

Das Feld PRI-AUTH zeigt die für den Thread verwendete primäre Berechtigungs-ID. Das Feld SEC-AUTH zeigt die sekundäre Berechtigungs-ID (falls vorhanden) für den Thread.

Die CORRELATION-Felder zeigen die 12 Byte lange Korrelations-ID des Threads, die als *eeeeettttnnnn* gebildet wird, wobei *eeee* entweder die Einstellung COMD, POOL oder ENTR ist, durch die angegeben wird, ob es sich um einen Befehlsthread, Pool-Thread oder DB2ENTRY-Thread handelt; *tttt* ist die Transaktions-ID und *nnnn* ist eine eindeutige Nummer.

Anmerkung: Eine an Db2 übergebene Korrelations-ID kann nur geändert werden, wenn die CICS-Anschlussfunktion eine Anmeldung für Db2 ausgibt. Wenn die Wiederverwendung der Anmeldung durch einen Thread erfolgt, der eine primäre Berechtigungs-ID verwendet, die während mehrerer Transaktionen konstant bleibt (beispielsweise durch Verwendung von AUTHID(name)), findet nur eine einzige Anmeldung statt. In dieser Instanz stimmt die Angabe für *tttt* in der Korrelations-ID nicht mit der ID der ausgeführten Transaktion überein. Es ist die ID der Transaktion, für die die ursprüngliche Anmeldung erfolgte.

Wenn der Thread in einer Arbeitseinheit aktiv ist, werden der Name der CICS-Transaktion, die zugehörige Tasknummer und schließlich die ID der lokalen CICS-Arbeitseinheit angezeigt.

Die in dieser Anzeige verwendete Korrelations-ID wird auch in Db2-Befehlen wie beispielsweise DISPLAY LOCK ausgegeben. Wenn Sie zum Beispiel diese Anzeige zusammen mit dem Befehl DISPLAY LOCK verwenden, können Sie ermitteln, welche CICS-Task eine Sperre in Db2 hält.

Ausgabe von DISPLAY STATISTICS

In diesem Abschnitt wird die Ausgabe des Befehls **DSNC DISPLAY STATISTICS** beschrieben.

```
DFHDB2014 07/09/98 14:35:45 IYK4Z2G1 Statistics report follows for RCTJT
accessing DB2 DB3A
```

DB2ENTRY	PLAN	CALLS	AUTHS	W/P	HIGH	ABORTS	-----COMMIT-----	
							1-PHASE	2-PHASE
*COMMAND		1	1	1	1	0	0	0
*POOL	*****	4	1	0	1	0	2	0
XC05	TESTP05	22	1	11	2	0	7	5
XP05	*****	5	2	0	1	0	1	1

```
DFHDB2020 01/17/98 15:45:27 IYKA4Z2G1 The display command is complete.
```

Abbildung 8. Beispielausgabe des Befehls DSNCL DISPLAY STATISTICS

DB2ENTRY

Name des DB2ENTRY-Objekts, *COMMAND für Aufrufe des Befehls DSNCL oder *POOL für die Poolstatistik.

PLAN

Der Planname, der diesem Eintrag zugeordnet ist. Acht Sterne in diesem Feld geben an, dass diese Transaktion die dynamische Planzuordnung verwendet. Der Transaktion DSNCL des Befehlsprozessors ist kein Plan zugeordnet.

Bei dynamischer Änderung eines Plannamens, der einem Eintrag zugeordnet ist, ist der letzte Plannamen der Name, der verwendet wird.

CALLS

Die Gesamtzahl der SQL-Anweisungen, die von Transaktionen ausgegeben werden, die diesem Eintrag zugeordnet sind.

AUTHS

Die Gesamtzahl der Anmeldeaufrufe für Transaktionen, die diesem Eintrag zugeordnet sind. Eine Anmeldung gibt nicht an, ob ein neuer Thread erstellt wird oder ein vorhandener Thread wiederverwendet wird. Wenn der Thread wiederverwendet wird, findet eine Anmeldung nur statt, wenn die Berechtigungs-ID oder die Transaktions-ID geändert wurde.

W/P

Die Häufigkeit, mit der alle verfügbaren Threads für diesen Eintrag belegt waren. Dieser Wert hängt vom THREADWAIT-Wert für den Eintrag ab. Wenn für THREADWAIT die Einstellung POOL festgelegt ist, gibt W/P die Häufigkeit an, mit der für die Transaktion ein Überlauf in den Pool ausgeführt wurde. Der Überlauf in den Pool ist nur an den Statistikdaten für das einzelne DB2ENTRY-Objekt erkennbar; er wird nicht in der Poolstatistik dargestellt.

Wenn für THREADWAIT die Einstellung YES festgelegt ist, wird dadurch angegeben, wie oft die Transaktion auf einen Thread warten musste (da die Anzahl der aktiven Threads den Wert für THREADLIMIT erreicht hatte), oder wie oft die Transaktion keinen neuen Thread-Tasksteuerblock verwenden konnte (da die Anzahl der zur Ausführung von Threads verwendeten Tasksteuerblöcke den Wert für TCBLIMIT erreicht hatte).

HIGH

Die maximale Anzahl von Threads, die von Transaktionen, die diesem Eintrag zugeordnet sind, zu einem beliebigen Zeitpunkt nach Start der Verbindung angefordert wurde, d. h. ein oberer Grenzwert für Threads.

Anmerkung: In CICS-Releases vor CICS Transaction Server for OS/390 Version 1 Release 2 beinhaltet der Wert für HIGH auch Transaktionen, deren Warten auf einen Thread erzwungen wurde, oder die Transaktionen, die an den Pool umgelenkt wurden. Ab CICS Transaction Server for OS/390 Version 1 Release 2 stellt der Wert für HIGH *nur* Threads dar, die für den Eintrag erstellt wurden.

ABORTS

Die Gesamtzahl der Arbeitseinheiten mit Wiederherstellung, die rückgängig gemacht wurden (Rollback). Dies umfasst sowohl abnormale Beendigungen als auch Rollbacks für Synchronisationspunkte einschließlich von Rollbacks für Synchronisationspunkte, die durch SQL-Codes des Typs -911 erzeugt wurden.

Parameterbeschreibung

DESTination

Gibt an, dass der Parameter MSGQUEUE der Tabelle DB2CONN geändert werden muss, wobei die alte Ziel-ID ('old') durch die neue Ziel-ID ('new') zu ersetzen ist.

old

Jede Ziel-ID, die aktuell in der MSGQUEUE des DB2CONN-Objekts festgelegt ist.

new

Eine neue Ziel-ID.

TRANsaction *transaktions-id ganze_zahl*

Gibt an, dass der der betreffenden Transaktion oder Gruppe zugeordnete Wert für THREADLIMIT geändert werden muss. Im Befehl wird eine Transaktions-ID verwendet, die den Pool, den Befehl oder den zu ändernden DB2ENTRY-Wert THREADLIMIT angibt.

- Zum Ändern des THREADLIMIT-Werts des Pools muss die Transaktions-ID CEPL verwendet werden.
- Zum Ändern des THREADLIMIT-Werts für Befehlsthreads muss die Transaktions-ID DSNL verwendet werden.
- Zum Ändern des THREADLIMIT-Werts für ein DB2ENTRY-Objekt verwenden Sie die Transaktions-ID einer beliebigen Transaktion, die für die Verwendung des DB2ENTRY-Objekts definiert ist.

ganze_zahl ist ein neuer Maximalwert.

Hinweise zur Verwendung

Die im Befehl DSNL MODIFY TRANSACTION angegebene ganze Zahl darf nicht größer als der Wert sein, der im DB2CONN-Objekt für den Parameter TCBLIMIT angegeben ist. Der niedrigste mögliche Wert ist null.

Beispiele

Beispiel 1

Gehen Sie wie folgt vor, um die Spezifikation des Parameters MSGQUEUE in DB2CONN von MTO1 in MTO2 zu ändern:

```
DSNL MODI DEST MTO1 MTO2
```

```
DFHDB2039 07/09/98 14:47:17 IYK4Z2G1 The error destinations are: MTO2 ****
****.
```

Abbildung 9. Beispielausgabe des Befehls DSNL MODIFY DESTINATION

Beispiel 2

Gehen Sie wie folgt vor, um den Grenzwert für den Pool-Thread in 12 zu ändern:

```
DSNL MODI TRAN CEPL 12
```

```
DFHDB2019 07/09/98 14:49:28 IYK4Z2G1 The modify command is complete.
```

Abbildung 10. Beispielausgabe des Befehls DSNL MODIFY TRANSACTION (Pool-Thread)

Beispiel 3

Gehen Sie wie folgt vor, um den Grenzwert für den Befehlsthread in 3 zu ändern:

```
DSNC MODI TRAN DSNC 3
```

```
DFHDB2019 07/09/98 14:49:28 IYK4Z2G1 The modify command is complete.
```

Abbildung 11. Beispielausgabe des Befehls *DSNC MODIFY TRANSACTION* (für einen Befehlsthread)

Beispiel 4

Gehen Sie wie folgt vor, um den von der Transaktion XP05 verwendeten Threadgrenzwert in 8 zu ändern:

```
DSNC MODI TRAN XP05 8
```

```
DFHDB2019 07/09/98 14:49:28 IYK4Z2G1 The modify command is complete.
```

Abbildung 12. Beispielausgabe des Befehls *DSNC MODIFY TRANSACTION* (Ändern des Threadgrenzwerts für das *DB2ENTRY*-Objekt)

DSNC STOP

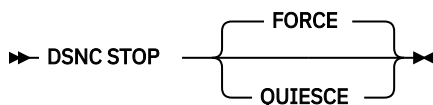
Verwenden Sie den Befehl **DSNC STOP** der CICS-Db2-Anschlussfunktion, um die Anschlussfunktion zu stoppen. Sie können alternativ den Befehl **CEMT** oder **EXEC CICS SET DB2CONN NOTCONNECTED** absetzen, um die Anschlussfunktion zu stoppen.

Umgebung

Dieser Befehl kann nur über ein CICS-Terminal abgesetzt werden.

Syntax

Syntax von STOP



Abkürzung

DSNC STOP oder STOP (bei Verwendung der Transaktion STOP über die CICS-Db2-Beispielgruppe DFH \$DB2).

Berechtigung

Der Zugriff auf diesen Befehl kann mithilfe der CICS-Sicherheitsprüfung 'Transaction-attach Security' (auf die Transaktionen selbst bezogene Sicherheitsprüfungen) für die Transaktion DSNC und mithilfe der CICS-Prüfung der Befehlssicherheit für die Ressource DB2CONN gesteuert werden. Für diesen Befehl ist der Zugriff UPDATE (Aktualisierungszugriff) erforderlich.

Weitere Informationen zur CICS-Sicherheit finden Sie in [Sicherheit in einer CICS-DB2-Umgebung](#).

Parameterbeschreibung

QUIESCE

Gibt an, dass die CICS-Db2-Anschlussfunktion nach der Beendigung der aktuell ausgeführten CICS-Transaktionen gestoppt werden soll. QUIESCE wartet, bis alle aktiven Transaktionen beendet sind, damit neue Arbeitseinheiten starten und Threads anfordern können.

FORCE

Gibt an, dass die CICS-Db2-Anschlussfunktion durch das Erzwingen eines Abbaus der Verbindung zu Db2 sofort gestoppt werden soll, ohne dass dabei Rücksicht auf eventuell ausgeführte Transaktionen genommen wird. Für aktuell ausgeführte Transaktionen, die auf Db2 zugegriffen haben, wird eine erzwungene Bereinigung (Force Purge) ausgeführt. Dazu gehören Transaktionen, die in einer vorherigen Arbeitseinheit möglicherweise Aktualisierungen in Db2 festgeschrieben haben, jedoch in ihrer aktuellen Arbeitseinheit noch nicht auf Db2 zugegriffen haben.

Hinweise zur Verwendung

Für DSNB STOP QUIESCE wird am Terminal die Nachricht DFHDB2012 ausgegeben. Das Terminal bleibt dann bis zur Beendigung des Systemabschlusses (bei Ausgabe der Nachricht DFHDB2025) gesperrt.

Für DSNB STOP FORCE wird am Terminal die Nachricht DFHDB2022 ausgegeben. Das Terminal bleibt dann bis zur Beendigung des Systemabschlusses (bei Ausgabe der Nachricht DFHDB2025) gesperrt.

Beispiele

Beispiel 1

Gehen Sie wie folgt vor, um die CICS-Db2-Anschlussfunktion mit Quiesce zu stoppen (Quiesce Stop):

```
DSNB STOP
```

```
DFHDB2012 07/09/98 14:54:28 IYK4Z2G1 Stop quiesce of the CICS-DB2 attachment  
facility from DB2 subsystem DB3A is proceeding.
```

Abbildung 13. Beispielausgabe des Befehls DSNB STOP

Die Nachricht, die das Ergebnis des in [Abbildung 13](#) auf Seite 44 gezeigten Befehls DSNB STOP ist, wird bei der Beendigung des Systemabschlusses durch die in [Abbildung 14](#) auf Seite 44 gezeigte Nachricht ersetzt.

```
DFHDB2025I 07/09/98 14:58:53 IYK4Z2G1 The CICS-DB2 attachment has disconnected  
from DB2 subsystem DB3A
```

Abbildung 14. Beispielausgabe des Befehls DSNB STOP nach Beendigung des Systemabschlusses

Beispiel 2

Gehen Sie wie folgt vor, um die CICS-Db2-Anschlussfunktion mit Quiesce zu stoppen (Quiesce Stop):

```
DSNB STOP FORCE
```

```
DFHDB2022 07/09/98 15:01:51 IYK4Z2G1 Stop force of the CICS-DB2 attachment  
facility from DF2D is proceeding.
```

Abbildung 15. Beispielausgabe des Befehls DSNB STOP FORCE

Die Nachricht, die das Ergebnis des in [Abbildung 15](#) auf Seite 44 gezeigten Befehls DSNB STOP FORCE ist, wird bei der Beendigung des Systemabschlusses durch die in [Abbildung 16](#) auf Seite 45 gezeigte Nachricht ersetzt.

```
DFHDB2025I 07/09/98 15:10:55 IYK4Z2G1 The CICS-DB2 attachment has disconnected
from DB2 subsystem DF2D group DFP2
```

Abbildung 16. Beispielausgabe des Befehls **DSNC STOP FORCE** nach Beendigung des Systemabschlusses

In diesem Beispiel wurde die Einrichtung zum Anhängen von Gruppen verwendet, sodass der Name des Db2-Subsystems und der Name der zugehörigen Gruppe angezeigt werden.

DSNC STRT

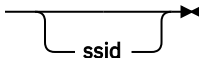
Verwenden Sie den Befehl **DSNC STRT**, um die CICS-Db2-Anschlussfunktion zu starten. Die CICS-Db2-Anschlussfunktion ermöglicht CICS-Anwendungsprogrammen den Zugriff auf Db2-Datenbanken. Setzen Sie alternativ den Befehl **CEMT** oder **EXEC CICS SET DB2CONN CONNECTED** ab, um Ihren CICS-Anwendungsprogrammen den Zugriff auf Db2-Datenbanken zu ermöglichen.

Umgebung

Dieser Befehl kann nur über ein CICS-Terminal abgesetzt werden.

Syntax

Syntax von STRT

➔ **DSNC STRT** 

Abkürzung

DSNC STRT oder STRT (bei Verwendung der Transaktion STRT über die CICS-Db2-Beispielgruppe DFH\$DB2).

Berechtigung

Der Zugriff auf diesen Befehl kann mithilfe der CICS-Sicherheitsprüfung 'Transaction-attach Security' (auf die Transaktionen selbst bezogene Sicherheitsprüfungen) für die Transaktion DSNC und mithilfe der CICS-Prüfung der Befehlssicherheit für die Ressource DB2CONN gesteuert werden. Für diesen Befehl ist der Zugriff UPDATE (Aktualisierungszugriff) erforderlich.

Weitere Informationen zur CICS-Sicherheit finden Sie in [Sicherheit in einer CICS-DB2-Umgebung](#).

Parameterbeschreibung

ssid

Gibt die ID eines Db2-Subsystems an, um die ID des Db2-Subsystems (DB2ID) oder die ID der Db2-Gruppe mit gemeinsamer Datennutzung (DB2GROUPEID) zu überschreiben, die im DB2CONN-Objekt angegeben ist. Sie können in einem DSNC STRT-Befehl keine ID einer Db2-Gruppe mit gemeinsamer Datennutzung angeben.

Hinweise zur Verwendung

Wenn beim Absetzen des Befehls DSNC STRT kein DB2CONN-Objekt installiert ist, wird die Fehlernachricht DFHDB2031 generiert, die angibt, dass kein DB2CONN-Objekt installiert ist. Ressourcendefinitionen müssen aus der CICS-Systemdefinitionstabelle installiert werden, bevor versucht werden kann, die CICS-Db2-Anschlussfunktion zu starten.

Wenn Sie den Befehl DSNC STRT absetzen und eine Db2-Subsystem-ID angeben, werden alle DB2GROUPEID-Elemente in der installierten DB2CONN-Definition mit Leerzeichen überschrieben und sie müssen er-

neut festgelegt werden (mithilfe von CEDA INSTALL oder des Befehls SET DB2CONN), damit das Anhängen von Gruppen nachfolgend möglich ist.

Die Hierarchie für die Entscheidung, welches Db2-Subsystems zu verwenden ist, lautet wie folgt:

1. Verwenden Sie die Subsystem-ID, wenn diese in einem DSNB STRT-Befehl angegeben ist.
2. Verwenden Sie das im installierten DB2CONN-Objekt verwendete DB2ID-Element, falls dieses nicht leer ist.
3. Verwenden Sie für das Anhängen von Gruppen das DB2GROUPID-Element im installierten DB2CONN-Objekt.
4. Verwenden Sie die Subsystem-ID (falls in INITPARM angegeben), wenn DB2ID und DB2GROUPID im zuletzt installierten DB2CONN-Objekt leer sind (oder wenn nachfolgend für sie Leerzeichen angegeben wurden). Bei jedem Start wird stets INITPARM verwendet, wenn das zuletzt installierte DB2CONN-Objekt eine leere DB2ID und eine leere DB2GROUPID enthielt, auch wenn die DB2ID oder die DB2GROUPID nachfolgend mithilfe des Befehls SET geändert wurden.
5. Verwenden Sie DSNB als Standard-ID des Subsystems.

Beispiele

Beispiel 1

Gehen Sie wie folgt vor, um die CICS-Db2-Anschlussfunktion mithilfe der Db2-Subsystem-ID (DB2ID) oder der ID einer Db2-Gruppe mit gemeinsamer Datennutzung (DB2GROUPID) von einem installierten DB2CONN-Objekt aus zu starten:

```
DSNB STRT
```

In diesem Beispiel wird die Einrichtung zum Anhängen von Gruppen verwendet, sodass der Name des Db2-Subsystems und der Name der zugehörigen Gruppe angezeigt werden.

```
DFHDB2023I 07/09/98 15:06:07 IYK4Z2G1 The CICS DB2 attachment has connected to  
DB2 subsystem DF2D group DFP2
```

Abbildung 17. Beispielausgabe des Befehls DSNB STRT

Beispiel 2

Gehen Sie wie folgt vor, um die CICS-Db2-Anschlussfunktion mithilfe eines installierten DB2CONN-Objekts zu starten, dabei jedoch zugleich die Db2-Subsystem-ID (DB2ID) oder die ID der Db2-Gruppe mit gemeinsamer Datennutzung (DB2GROUPID) im DB2CONN-Objekt mit der Db2-Subsystem-ID DB3A zu überschreiben:

```
DSNB STRT DB3A
```

```
DFHDB2023I 07/09/97 15:06:07 IYK4Z2G1 The CICS DB2 attachment has connected to  
DB2 subsystem DB3A
```

Abbildung 18. Beispielausgabe des Befehls DSNB STRT unter Verwendung von DB3A

Wenn Sie die Einrichtung zum Anhängen von Gruppen nicht verwenden, und das Db2-Subsystem nicht aktiv ist, wenn Sie versuchen, die CICS-Db2-Anschlussfunktion, erhalten Sie folgende Ausgabe, wenn im DB2CONN-Objekt STANDBYMODE=NOCONNECT angegeben ist:

```
DFHDB2018 07/09/98 15:14:10 IYK4Z2G1 DB3A DB2 subsystem is not active.
```

Abbildung 19. Beispielausgabe des Befehls DSNCLSTRT wenn Db2 nicht aktiv ist und STANDBYMODE=NOCONNECT angegeben ist

Wenn STANDBYMODE=CONNECT oder RECONNECT angegeben ist, erhalten Sie folgende Ausgabe:

```
DFHDB2037 07/09/98 15:15:42 IYK4Z2G1 DB2 subsystem DB3A is not active.  
The CICS DB2 attachment facility is waiting.
```

Abbildung 20. Beispielausgabe des Befehls DSNCLSTRT, wenn Db2 nicht aktiv ist und STANDBYMODE=CONNECT oder RECONNECT angegeben ist

Wenn Sie die Einrichtung zum Anhängen von Gruppen verwenden, und bei dem Versuch, die CICS-Db2-Anschlussfunktion zu starten, keine Db2-Subsysteme in der Gruppe mit gemeinsamer Datennutzung aktiv sind, erhalten Sie folgende Ausgabe, wenn im DB2CONN-Objekt STANDBYMODE=NOCONNECT angegeben ist:

```
DFHDB2037 07/09/01 12:30:10 IYK2ZFV1 DB2 group DFP2 has no active members.
```

Abbildung 21. Beispielausgabe des Befehls DSNCLSTRT bei Verwendung der Einrichtung zum Anhängen von Gruppen, wenn keine Db2-Subsysteme in der Gruppe mit gemeinsamer Datennutzung aktiv sind und STANDBYMODE=NOCONNECT angegeben ist

Wenn STANDBYMODE=CONNECT oder RECONNECT angegeben ist, erhalten Sie folgende Ausgabe:

```
DFHDB2037 07/09/01 12:55:00 IYK2ZFV1 DB2 group DFP2 has no active members.  
The CICS DB2 attachment facility is waiting.
```

Abbildung 22. Beispielausgabe des Befehls DSNCLSTRT bei Verwendung der Einrichtung zum Anhängen von Gruppen, wenn keine Db2-Subsysteme in der Gruppe mit gemeinsamer Datennutzung aktiv sind und STANDBYMODE=CONNECT oder RECONNECT angegeben ist

Kapitel 4. Sicherheit für Db2

In der CICS-Db2-Umgebung gibt es vier Hauptphasen, in denen Sie Sicherheitsprüfungen implementieren können.

Die vier Phasen lauten wie folgt:

- Wenn ein CICS-Benutzer sich in einer CICS-Region anmeldet. Die CICS-Anmeldung authentifiziert Benutzer, indem sie überprüft, ob die Benutzer eine gültige Benutzer-ID und ein zugehöriges Kennwort angeben.
- Wenn ein CICS-Benutzer versucht, eine CICS-Ressource, die sich auf Db2 bezieht, zu verwenden oder zu ändern. Dabei kann es sich um eine DB2CONN-, DB2ENTRY- oder eine DB2TRAN-Ressourcendefinition handeln, um eine CICS-Transaktion, die zum Abrufen von Daten auf Db2 zugreift, oder um eine CICS-Transaktion, die Befehle für die CICS-Db2-Anschlussfunktion oder an Db2 selbst absetzt. In dieser Phase können Sie von RACF oder einem äquivalenten externen Sicherheitsmanager verwaltete CICS-Sicherheitsmechanismen verwenden, um den Zugriff des CICS-Benutzers auf die Ressource zu steuern.
- Wenn eine CICS-Region eine Verbindung zu Db2 herstellt und wenn eine Transaktion einen Thread in Db2 anfordert. Sowohl die CICS-Region als auch die Transaktion müssen Berechtigungs-IDs für Db2 bereitstellen und diese Berechtigungs-IDs werden von RACF oder einem äquivalenten externen Sicherheitsmanager geprüft.
- Wenn ein CICS-Benutzer versucht, eine CICS-Transaktion zum Ausführen oder Ändern einer Db2-Ressource zu verwenden. Dies könnte ein Plan, ein Db2-Befehl oder eine Ressource sein, die zur Ausführung von dynamischem SQL benötigt wird. In dieser Phase können Sie die Db2-Sicherheitsprüfung verwenden, die entweder von Db2 selbst oder von RACF oder von einem äquivalenten, externen Sicherheitsmanager verwaltet wird, um den Zugriff eines CICS-Benutzers auf die Ressource zu steuern.

Sie können auch RACF oder einen äquivalenten, externen Sicherheitsmanager verwenden, um die Komponenten, aus denen CICS und Db2 gebildet wird, vor unbefugtem Zugriff zu schützen. Sie können diesen Schutz auf Db2-Datenbanken, Protokolle, Bootstrap-Dateien (BSDSs) und Bibliotheken außerhalb des Umfangs von Db2 und auf CICS-Dateien und Bibliotheken anwenden. Sie können den VSAM-Kennwortschutz als teilweisen Ersatz für den durch RACF bereitgestellten Schutz verwenden. Weitere Informationen finden Sie in [Sicherheit für CICS-Systemressourcen](#).

Anmerkung: Als RACF wird hier der externe Sicherheitsmanager bezeichnet, den CICS verwendet. Abgesehen von den expliziten RACF-Beispielen gilt die allgemeine Beschreibung in gleicher Weise für alle funktional entsprechenden externen Sicherheitsmanager, die nicht von IBM stammen.

[Abbildung 23 auf Seite 50](#) zeigt die an einer CICS-Db2>-Umgebung beteiligten Sicherheitsmechanismen.

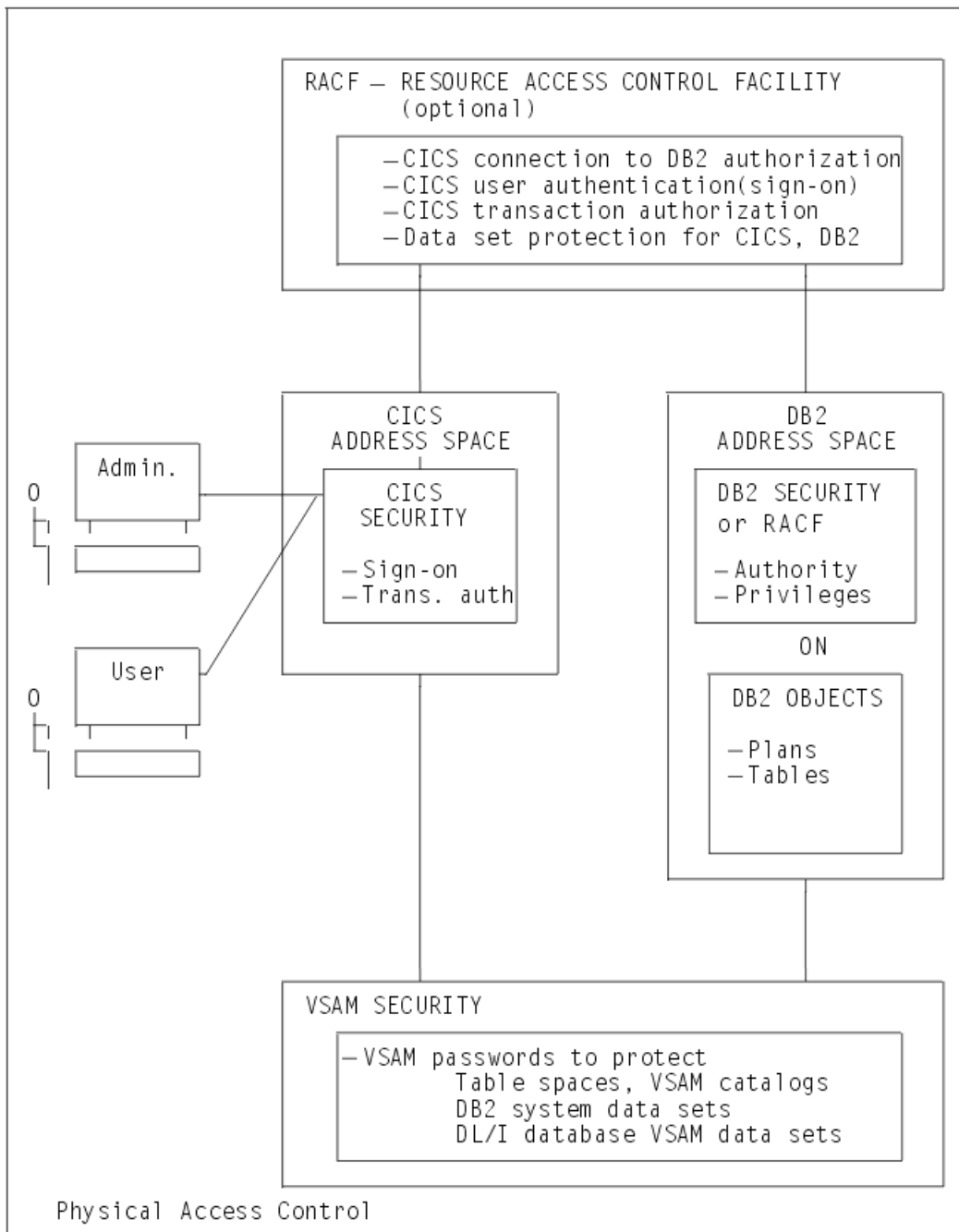


Abbildung 23. Übersicht über die CICS-Db2-Sicherheitsmechanismen

Zugriff auf Db2-bezogene Ressourcen in CICS steuern

Sie können den Zugriff auf Db2-Ressourcen in Ihrer CICS-Region steuern und eine Sicherheitsprüfung für die Ressourcen starten, indem Sie einen externen Sicherheitsmanager und den passenden CICS-Sicherheitsmechanismus aktivieren.

Informationen zu diesem Vorgang

CICS-Benutzer möchten möglicherweise folgende Aktivitäten ausführen, bei denen Db2 beteiligt ist:

- DB2CONN-, DB2ENTRY- und DB2TRAN-Ressourcendefinitionen abfragen, ändern, erstellen oder löschen.

- Eine Transaktion verwenden, die zum Abrufen von Daten auf Db2 zugreift, oder Befehle der CICS-Db2-Anschlussfunktion oder Db2-Befehle mithilfe der Transaktion DSNB absetzen.

Verwenden Sie RACF oder einen äquivalenten externen Sicherheitsmanager, um Sicherheitsprüfungen in Ihrer CICS-Region auszuführen. Wenn ein Benutzer versucht, auf geschützte Ressourcen zuzugreifen, ruft CICS den externen Sicherheitsmanager auf, um eine Sicherheitsprüfung auszuführen. RACF führt Sicherheitsprüfungen unter Verwendung der ID des CICS-Benutzers aus, die bei der Anmeldung des Benutzers in CICS authentifiziert wird. Benutzer, die sich nicht bei CICS anmelden, erhalten die Standard-Benutzer-ID, es sei denn, einem Terminal, das voreingestellte Sicherheit verwendet, wurde eine Benutzer-ID ständig zugeordnet.

Aktivieren Sie den für Ihre CICS-Region passenden Sicherheitsmechanismus: auf die Transaktion selbst bezogene Sicherheitsprüfung (Transaction-attach Security), Ressourcensicherheit, Befehlssicherheit oder Ersatzsicherheit.

Weitere Informationen zur CICS-Sicherheit finden Sie in [CICS TS-Sicherheit](#).

Benutzerzugriff auf DB2CONN-, DB2TRAN- und DB2ENTRY-Ressourcendefinitionen steuern

Sie können den Benutzerzugriff auf DB2CONN-, DB2TRAN- und DB2ENTRY-Ressourcendefinitionen steuern, indem Sie verschiedene CICS-Sicherheitsmechanismen aktivieren.

Informationen zu diesem Vorgang

Folgende Sicherheitsmechanismen können für die Steuerung des Benutzerzugriffs auf Db2-Ressourcendefinitionen verwendet werden.

- Steuern Sie die Möglichkeit von Benutzern, auf bestimmte Ressourcen zuzugreifen, indem Sie den CICS-Mechanismus der **Ressourcensicherheit** verwenden. Die Ressourcensicherheit ist auf Transaktionsebene implementiert. Sie können beispielsweise einige Benutzer daran hindern, eine bestimmte DB2ENTRY-Definition zu ändern. In [„Ressourcensicherheit verwenden, um den Zugriff auf die DB2ENTRY- und DB2TRAN-Ressourcendefinitionen zu steuern“](#) auf Seite 51 erfahren Sie, wie dieser Sicherheitsmechanismus verwendet wird.
- Steuern Sie die Möglichkeit von Benutzern, bestimmte SPI-Befehle für Db2-bezogene Ressourcen abzusetzen, indem Sie den CICS-Mechanismus der **Befehlssicherheit** verwenden. Die Befehlssicherheit wird auch auf Transaktionsebene implementiert. Sie könnten beispielsweise nur bestimmten Benutzern das Absetzen von CREATE- und DISCARD-Befehlen für DB2ENTRY-Ressourcendefinitionen ermöglichen. In [„Befehlssicherheit verwenden, um das Ausgeben von SPI-Befehlen für DB2CONN-, DB2ENTRY- und DB2TRAN-Ressourcendefinitionen zu steuern“](#) auf Seite 53 erfahren Sie, wie dieser Sicherheitsmechanismus verwendet wird.
- Steuern Sie die Möglichkeit von Benutzern, die Berechtigungs-IDs zu ändern, die CICS für Db2 bereitstellt, indem Sie die CICS-Mechanismen der **Ersatzsicherheit und AUTHTYPE-Sicherheit** verwenden. Die Berechtigungs-IDs werden für die Db2-Sicherheitsprüfung verwendet und sie werden festgelegt von den Attributen AUTHID, COMAUTHID, AUTHTYPE und COMAUTHTYPE in Db2-bezogenen Ressourcendefinitionen und durch das Attribut SIGNID in der DB2CONN-Definition für die CICS-Region. CICS überprüft, dass der Benutzer, der die Berechtigungs-ID ändern möchte, dazu berechtigt ist, für die vorhandene Berechtigungs-ID zu handeln, die in der Ressourcendefinition angegeben ist. In [„Ersatzsicherheit und AUTHTYPE-Sicherheit zum Steuern des Zugriffs auf die Berechtigungs-IDs verwenden, die CICS für Db2 bereitstellt“](#) auf Seite 55 erfahren Sie, wie diese Sicherheitsmechanismen verwendet werden.

Ressourcensicherheit verwenden, um den Zugriff auf die DB2ENTRY- und DB2TRAN-Ressourcendefinitionen zu steuern

Der CICS-Mechanismus für die Ressourcensicherheit steuert den Benutzerzugriff auf benannte CICS-Ressourcen. Sie können ihn zum Beispiel verwenden, um bestimmte Ressourcen (zum Beispiel eine bestimmte DB2ENTRY-Definition) zu schützen, damit diese nicht von bestimmten Benutzern geändert werden.

Informationen zu diesem Vorgang

Die CICS-Befehlssicherheit kann verhindern, dass Benutzer bestimmte Aktionen für Typen von Ressourcen (zum Beispiel für "alle DB2ENTRY-Definitionen") ausführen, aber sie kann nicht einzelne Elemente innerhalb des Ressourcentyps schützen.

Da Ihre CICS-Region nur über eine einzige DB2CONN-Definition verfügt, müssen Sie sie nicht mit der Ressourcensicherheit schützen; Sie können den Zugriff auf die DB2CONN-Definition mit der Befehlssicherheit steuern. Darüber hinaus werden DB2TRAN-Definitionen zum Zweck der Ressourcensicherheit als Erweiterungen der DB2ENTRY-Definition behandelt, auf die sie verweisen, und sie sind nicht selbst für die Ressourcensicherheit definiert. Wenn Sie einem Benutzer die Berechtigung für den Zugriff auf eine DB2ENTRY-Definition erteilen, erteilen Sie ihm gleichzeitig die Berechtigung, auf die DB2TRAN-Definitionen zuzugreifen, die auf die DB2ENTRY-Definition verweisen. (Wenn eine Transaktion den Namen des DB2ENTRY-Objekts ändert, dem eine DB2TRAN-Definition zugeordnet ist, wird eine doppelte Sicherheitsprüfung ausgeführt, um zu prüfen, ob der Benutzer über die Berechtigung verfügt, sowohl das alte DB2ENTRY-Objekt, auf das die Definition verweist, als auch das neue DB2ENTRY-Objekt zu ändern, auf das die Definition verweisen wird.) Für die Ressourcensicherheit müssen Sie daher nur Ihre DB2ENTRY-Definitionen für RACF definieren.

Wenn die Ressourcensicherheit für eine Transaktion aktiviert ist, überprüft der externe Sicherheitsmanager, ob die der Transaktion zugeordnete Benutzer-ID berechtigt ist, die beteiligte Ressource zu ändern. In Sicherheit von Ressourcendefinitionen finden Sie weitere Informationen zu diesem Prozess.

Führen Sie folgende Schritte aus, um Ihre Db2-bezogenen Ressourcen mithilfe der Ressourcensicherheit zu schützen.

Vorgehensweise

1. Um RACF oder einen äquivalenten externen Sicherheitsmanager zu aktivieren und die Ressourcensicherheit für eine CICS-Region verfügbar zu machen, geben Sie SEC=YES als Systeminitialisierungsparameter für die CICS-Region an.
2. Erstellen Sie in RACF allgemeine Ressourcenklassen, die Ihre Db2-bezogenen Ressourcen enthalten sollen. Sie benötigen eine Memberklasse und eine Gruppierungsklasse.

Im Gegensatz zu den standardmäßigen RACF-Ressourcenklassennamen für CICS gibt es für DB2ENTRY-Objekte keine von IBM gelieferten standardmäßigen Klassennamen. Erstellen Sie Ihre eigenen installationsdefinierten Klassennamen, indem Sie dem installationsdefinierten Teil (Modul ICHRRCDE) der RACF-Klassendeskriptortabelle (CDT – Class Descriptor Table) neue Klassendeskriptoren hinzufügen. Ein Beispiel für die Vorgehensweise finden Sie in dem von IBM gelieferten Beispieljob RRCDE, der in dem Member DFH\$RACF von CICSTS56.CICS.SDFHSAMP bereitgestellt wird. Hier finden Sie ein Beispiel für eine Memberklasse mit dem Namen XCICSDB2 und eine Gruppierungsklasse mit dem Namen ZCICSDB2. Dieses Beispiel verwendet dieselbe Namenskonvention wie die standardmäßigen Ressourcenklassennamen für CICS. Verwenden Sie für CICS class names for Db2-bezogene Ressourcendefinitionen nicht vorhandene CICS-Klassennamen, sondern erstellen Sie neue Klassennamen mithilfe einer ähnlichen Namenskonvention.

3. Definieren Sie Profile für Ihre DB2ENTRY-Definitionen in den Ressourcenklassen, die Sie erstellt haben.

Wenn Sie beispielsweise der Ressourcenklasse XCICSDB2 eine Reihe von DB2ENTRY-Namen hinzufügen möchten, verwenden Sie den Befehl RDEFINE wie folgt:

```
RDEFINE XCICSDB2 (db2ent1, db2ent2, db2ent3., db2entn) UACC(NONE)
                NOTIFY(sys_admin_userid)
```

Durch das Schützen von DB2ENTRY-Ressourcendefinitionen wird auch der Zugriff auf zugeordnete DB2TRAN-Definitionen geschützt, da ein DB2TRAN-Objekt als Erweiterung des DB2ENTRY-Objekts betrachtet wird, auf das es verweist. Sie müssen Ihre DB2CONN-Definition nicht mithilfe der Ressourcensicherheit schützen.

4. Um die Ressourcensicherheit für Ihre Db2-bezogenen Ressourcen zu aktivieren, geben Sie als Systeminitialisierungsparameter für die CICS-Region XDB2=*name* an, wobei *name* der Name der allgemeinen Ressourcenklasse ist, den Sie für Ihre Db2-bezogenen Ressourcen definiert haben.

5. Geben Sie in der Ressourcendefinition RESSEC=YES für alle Transaktionen an, bei denen Db2-bezogene Ressourcen beteiligt sind, für die Sie die Ressourcensicherheit aktivieren möchten. Wenn nun ein Benutzer versucht, eine dieser Transaktionen für den Zugriff auf eine der von Ihnen geschützten Db2-bezogenen Ressourcen zu verwenden, überprüft RACF, dass die Benutzer-ID für den Zugriff auf diese Ressource berechtigt ist.
6. Erteilen Sie Ihren CICS-Benutzern oder Benutzergruppen die Berechtigung, geeignete Aktionen für die einzelnen Db2-bezogenen Ressource auszuführen, die Sie geschützt haben.

Beachten Sie, dass ein Benutzer, der über die Berechtigung zum Ausführen von Aktionen für eine DB2ENTRY-Definition verfügt, automatisch berechtigt ist, dieselben Aktionen für DB2TRAN-Definitionen auszuführen, die dieser DB2ENTRY-Definition zugeordnet sind. Der Zugriff, über den Benutzer verfügen müssen, um bestimmte Aktionen auszuführen, lautet wie folgt:

INQUIRE (Befehl)

Erfordert die Berechtigung READ

SET (Befehl)

Erfordert die Berechtigung UPDATE

CREATE (Befehl)

Erfordert die Berechtigung ALTER

DISCARD (Befehl)

Erfordert die Berechtigung ALTER

Beispielsweise können Sie mit dem Befehl PERMIT eine Gruppe von Benutzern dazu berechtigen, das geschützte DB2ENTRY-Element db2ent1 in der Klasse XCICSDB2 mithilfe der Berechtigung UPDATE wie folgt zu ändern:

```
PERMIT db2ent1 CLASS(XCICSDB2) ID(group1) ACCESS(UPDATE)
```

Befehlssicherheit verwenden, um das Ausgeben von SPI-Befehlen für DB2CONN-, DB2ENTRY- und DB2TRAN-Ressourcendefinitionen zu steuern

Verwenden Sie die Mechanismen der CICS-Befehlssicherheit, um DB2CONN-, DB2ENTRY- und DB2TRAN-Ressourcendefinitionen zu schützen.

Informationen zu diesem Vorgang

Der Mechanismus der CICS-Befehlssicherheit steuert die Möglichkeit des Benutzers, bestimmte SPI-Befehle für Typen von Db2-bezogenen Ressourcen abzusetzen. Sie können ihn z. B. verwenden, um zu steuern, für welche Benutzer es zulässig ist, CREATE- und DISCARD-Befehle für die DB2ENTRY-Ressourcendefinitionen abzusetzen. Anders als die Ressourcensicherheit kann die CICS-Befehlssicherheit einzelne benannte Ressourcen nicht schützen; sie ist dafür vorgesehen, Typen von Ressourcen zu schützen. Sie können mit der Befehlssicherheit DB2CONN-, DB2ENTRY- und DB2TRAN-Ressourcendefinitionen schützen.

Wenn die Befehlssicherheit für eine Transaktion aktiviert ist, überprüft der externe Sicherheitsmanager, ob die der Transaktion zugeordnete Benutzer-ID für die Verwendung dieses Befehls berechtigt ist, mit dem der Typ der beteiligten Ressource geändert werden soll. In [CICS-Befehlssicherheit](#) finden Sie weitere Informationen zu diesem Prozess.

Wenn Sie für eine bestimmte Transaktion sowohl die Ressourcensicherheit als auch die Befehlssicherheit aktiviert haben, führt RACF zwei Sicherheitsprüfungen für die Benutzer-ID aus. Wenn bei einer Transaktion beispielsweise der Benutzer beteiligt ist, der den Befehl DISCARD für die DB2ENTRY-Definition db2ent1 absetzt, überprüft RACF Folgendes:

1. Die Benutzer-ID ist berechtigt, den Befehl DISCARD (Berechtigung ALTER) für den Ressourcentyp DB2ENTRY auszugeben.
2. Die Benutzer-ID ist berechtigt, auf die DB2ENTRY-Definition db2ent1 mit der Berechtigung ALTER zuzugreifen.

Führen Sie folgende Schritte aus, um Ihre Db2-bezogenen Ressourcen mithilfe der Befehlssicherheit zu schützen.

Vorgehensweise

1. Um RACF oder einen äquivalenten externen Sicherheitsmanager zu aktivieren, müssen Sie für eine CICS-Region SEC=YES als Systeminitialisierungsparameter für die Region angeben.
2. Fügen Sie die Db2>-Ressourcennamen DB2CONN, DB2ENTRY und DB2TRAN als Ressourcen-IDs in einer der von IBM gelieferten IBM-Ressourcenklassen für RACF-Ressourcenklassen für CICS--Befehle, CCICSCMD oder VCICSCMD, hinzu.

Alternativ dazu können Sie für Ihre CICS-Befehle eine benutzerdefinierte allgemeine Ressourcenklasse verwenden. Weitere Informationen hierzu finden Sie in [CICS-Ressourcen, die der Prüfung der Befehlssicherheit unterliegen](#).

Beispielsweise können Sie wie folgt mit dem Befehl REDEFINE ein Profil mit dem Namen CMDSAMP in der Standardklasse VCICSCMD definieren und den Operanden ADDMEM verwenden, um anzugeben, dass die Db2-Ressourcentypen durch dieses Profil geschützt werden sollen:

```
RDEFINE VCICSCMD CMDSAMP UACC(NONE)
          NOTIFY(sys_admin_userid)
          ADDMEM(DB2CONN, DB2ENTRY, DB2TRAN)
```

3. Gehen Sie wie folgt vor, um die Befehlssicherheit für eine CICS-Region verfügbar zu machen:
 - a) Wenn Sie die von IBM gelieferten RACF-Ressourcenklassen CCICSCMD oder VCICSCMD für CICS-Befehlsprofile verwendet haben, geben Sie XCMD=YES als Systeminitialisierungsparameter für die Region an.
Die Angabe von YES bedeutet, dass CCICSCMD und VCICSCMD zum Erstellen von RACF-Speicherprofilen verwendet werden.
 - b) Wenn Sie eine benutzerdefinierte allgemeine Ressourcenklasse für CICS-Befehle verwendet haben, geben Sie XCMD=*benutzerklasse* als Systeminitialisierungsparameter für die Region an, wobei *benutzerklasse* für den Namen der benutzerdefinierten allgemeinen Ressourcenklasse steht.
4. Geben Sie in der Ressourcendefinition CMDSEC=YES für alle Transaktionen an, bei denen Db2-bezogene Ressourcen beteiligt sind, für die Sie die Befehlssicherheit aktivieren möchten. Wenn nun ein Benutzer versucht, eine dieser Transaktionen zum Absetzen eines Befehls zu verwenden, mit dem eine der von Ihnen geschützten, Db2-bezogenen Ressourcen geändert wird, überprüft RACF, ob die Benutzer-ID zum Absetzen dieses Befehls für den genannten Typ von Ressource berechtigt ist.
5. Erteilen Sie Ihren CICS-Benutzern oder -Benutzergruppen die Berechtigung, entsprechende Befehle für alle Typen von Db2-bezogenen Ressourcen abzusetzen. Für die Befehlssicherheit müssen Sie gesonderte Berechtigungen erteilen, sowohl bezogen auf den Ressourcentyp DB2TRAN als auch bezogen auf den Ressourcentyp DB2ENTRY. Sie können auch den Ressourcentyp DB2CONN schützen (das heißt, die DB2CONN-Definition der CICS-Region).

Der Zugriff, über den Benutzer verfügen müssen, um bestimmte Befehle abzusetzen, lautet wie folgt:

INQUIRE (Befehl)

Erfordert die Berechtigung READ

SET (Befehl)

Erfordert die Berechtigung UPDATE

CREATE (Befehl)

Erfordert die Berechtigung ALTER

DISCARD (Befehl)

Erfordert die Berechtigung ALTER

Wenn Sie beispielsweise die Db2-Ressourcentypen im Profil CMDSAMP wie im Beispiel in Schritt 2 definiert haben, können Sie den Befehl PERMIT verwenden, um wie folgt eine Gruppe von Benutzern für das Absetzen von EXEC CICS INQUIRE-Befehlen für die Db2-Ressourcentypen zu berechtigen:

```
PERMIT CMDSAMP CLASS(VCICSCMD) ID(operator_group) ACCESS(READ)
```

Sie können innerhalb einer Transaktion abfragen, ob eine Benutzer-ID über den Befehl **EXEC CICS QUERY SECURITY RESTYPE(SPCOMMAND)** auf Db2-Ressourcentypen zugreifen kann, wobei der Parameter RESID die Einstellung DB2CONN, DB2ENTRY oder DB2TRAN angibt.

Ersatzsicherheit und AUTHTYPE-Sicherheit zum Steuern des Zugriffs auf die Berechtigungs-IDs verwenden, die CICS für Db2 bereitstellt

Die CICS-Sicherheitsmechanismen der Ersatzsicherheit und AUTHTYPE-Sicherheit steuern die Möglichkeit des Benutzers, die Berechtigungs-IDs zu ändern, die CICS für Db2 bereitstellt.

Informationen zu diesem Vorgang

Verwenden Sie die Ersatzsicherheit und die AUTHTYPE-Sicherheit, um sicherzustellen, dass nur bestimmte Benutzer die Berechtigungs-IDs ändern dürfen, die für die Db2-eigene Sicherheitsprüfung verwendet werden. Die Ersatzsicherheit und die AUTHTYPE-Sicherheit werden für die gesamte CICS-Region festgelegt und alle Transaktionen, die Änderungen an den Berechtigungs-IDs beinhalten, unterliegen diesen Sicherheitsmechanismen.

In „Für Db2 Berechtigungs-IDs für die CICS-Region und für CICS-Transaktionen bereitstellen“ auf Seite 58 wird erläutert, wie diese Berechtigungs-IDs ausgewählt und geändert werden. Zusammenfassend kann man sagen, dass die Berechtigungs-IDs, die CICS für Db2 bereitstellt, festgelegt werden durch die Attribute AUTHID, COMAUTHID, AUTHTYPE und COMAUTHTYPE in Db2-bezogenen Ressourcendefinitionen sowie durch das Attribut SIGNID in der DB2CONN-Definition für die CICS-Region. Zum Ändern der Berechtigungs-ID benötigen Sie zunächst die Berechtigung zum Ändern von DB2CONN- und DB2ENTRY-Definitionen, die möglicherweise durch die Befehlssicherheit oder die Ressourcensicherheit geschützt werden. Die Ersatzsicherheit stellt eine zusätzliche Ebene des Zugriffsschutzes bereit, da hier CICS im Namen von Db2 überprüft, dass der Benutzer, der die Berechtigungs-ID ändert, dazu berechtigt ist, als Ersatz für die vorhandene Berechtigungs-ID zu agieren, die in der Ressourcendefinition angegeben ist.

Bei der echten Ersatzsicherheit wird eine Sicherheitsprüfung bereitgestellt, wenn ein Benutzer versucht, die Attribute SIGNID, AUTHID oder COMAUTHID in einer DB2CONN- oder DB2ENTRY-Definition zu ändern, die alle eine Berechtigungs-ID angeben, die verwendet wird, wenn sich ein Prozess bei Db2 anmeldet. CICS verwendet die Ersatzbenutzerfunktionalität von RACF, um diese Prüfung auszuführen. Ein Ersatzbenutzer ist ein Benutzer, der die Berechtigung hat, Arbeit im Namen eines anderen Benutzers auszuführen, ohne das Kennwort dieses anderen Benutzers zu kennen. Wenn ein Benutzer versucht, eines der Attribute SIGNID, AUTHID oder COMAUTHID zu ändern, wird RACF von CICS aufgerufen, um zu überprüfen, dass der Benutzer als Ersatz der Berechtigungs-ID berechtigt ist, die zurzeit im Attribut SIGNID, AUTHID oder COMAUTHID angegeben ist.

Für die Attribute AUTHTYPE und COMAUTHTYPE, die einen Typ einer zu verwendenden Berechtigungs-ID bereitstellen, statt eine genaue Berechtigungs-ID anzugeben, kann CICS keine echte Ersatzsicherheit verwenden. Stattdessen wird der Mechanismus der AUTHTYPE-Sicherheit verwendet. Wenn ein Benutzer versucht, eines der Attribute AUTHTYPE oder COMAUTHTYPE zu ändern, wird RACF von CICS aufgerufen, um zu überprüfen, dass der Benutzer über ein Profil, das Sie für die allgemeine RACF-Ressourcenklasse FACILITY definiert haben, berechtigt ist. Obwohl die AUTHTYPE-Sicherheit keine echte Ersatzsicherheit ist, wird sie durch denselben Systeminitialisierungsparameter aktiviert und Sie möchten sie wahrscheinlich zusätzlich zur Ersatzsicherheit verwenden; Sie erfahren daher in den Anweisungen in diesem Abschnitt, wie beide Typen von Sicherheit eingerichtet werden.

Wenn die DB2CONN- und DB2ENTRY-Ressourcendefinitionen im Rahmen eines Kaltstarts oder Erststarts von CICS installiert werden und die Ersatzsicherheit und die AUTHTYPE-Sicherheit aktiviert sind, müssen Sie beachten, dass RACF Ersatzsicherheitsprüfungen und AUTHTYPE-Sicherheitsprüfungen für die Benutzer-ID der CICS-Region ausführt. Wenn Sie auf diese Weise DB2CONN- und DB2ENTRY-Ressourcendefinitionen installieren, müssen Sie sicherstellen, dass die Benutzer-ID für die CICS-Region als Ersatz für al-

le Berechtigungs-IDs definiert ist, die in den Ressourcendefinitionen angegeben sind, und dass sie über die richtigen Profile in der allgemeinen RACF-Ressourcenklasse FACILITY berechtigt wurde.

Um die Ersatzsicherheit und die AUTHTYPE-Sicherheit zum Schutz der Berechtigungs-IDs zu implementieren, die CICS für Db2 bereitstellt, führen Sie folgende Schritte aus:

Vorgehensweise

1. Um RACF oder einen äquivalenten externen Sicherheitsmanager zu aktivieren, müssen Sie für eine CICS-Region SEC=YES als Systeminitialisierungsparameter für die Region angeben.
2. Geben Sie als Systeminitialisierungsparameter für eine CICS-Region XUSER=YES an, um die Ersatzsicherheit und die AUTHTYPE-Sicherheit für die Region zu aktivieren. Mit diesem Systeminitialisierungsparameter werden beide Sicherheitsmechanismen aktiviert. Wenn die Sicherheitsmechanismen aktiviert sind, wird RACF von CICS aufgerufen, um Sicherheitsprüfungen auszuführen, sobald in einer Transaktion EXEC CICS SET-, CREATE- und INSTALL-Befehle enthalten sind, die Operationen für die Attribute SIGNID, AUTHID, COMAUTHID, AUTHTYPE und COMAUTHTYPE der DB2CONN- und DB2ENTRY-Ressourcendefinitionen ausführen. Für die Attribute SIGNID, AUTHID und COMAUTHID führt RACF die Ersatzsicherheitsprüfung und für das Attribut AUTHTYPE oder COMAUTHTYPE die AUTHTYPE-Sicherheitsprüfung aus.
3. Für die Ersatzsicherheit müssen Sie entsprechende CICS-Benutzer oder -Benutzergruppen als Ersatz für alle Berechtigungs-IDs definieren, die in den Attributen SIGNID, AUTHID oder COMAUTHID Ihrer DB2CONN- und DB2ENTRY-Definitionen angegeben sind. Gehen Sie wie folgt vor, um eine Benutzer-ID als Ersatz für eine Berechtigungs-ID zu definieren:

- a) Erstellen Sie in der RACF-Klasse SURROGAT ein Profil für die Berechtigungs-ID mit einem Namen im Format *berechtigungs-id.DFHINSTL*, wobei die Berechtigungs-ID als Eigner definiert ist. Wenn Sie im Attribut SIGNID, AUTHID oder COMAUTHID beispielsweise DB2AUTH1 als Berechtigungs-ID angegeben haben, verwenden Sie zum Erstellen eines Profils den folgenden Befehl:

```
RDEFINE SURROGAT DB2AUTH1.DFHINSTL UACC(NONE) OWNER(DB2AUTH1)
```

- b) Erteilen Sie entsprechenden CICS-Benutzern die Berechtigung, als Ersatz für die Berechtigungs-ID zu agieren, indem Sie ihnen die Leseberechtigung (READ) für das Profil erteilen, das Sie erstellt haben. Wenn Sie beispielsweise einem Benutzer mit der ID CICSUSR1 die Berechtigung erteilen möchten, als Ersatz für die Berechtigungs-ID DB2AUTH1 zu agieren und somit alle SIGNID, AUTHID oder COMAUTHID-Attribute zu installieren oder zu ändern, die DB2AUTH1 als vorhandene Berechtigungs-ID angeben, verwenden Sie den folgenden Befehl:

```
PERMIT DB2AUTH1.DFHINSTL CLASS(SURROGAT) ID(CICSUSR1) ACCESS(READ)
```

Wiederholen Sie diesen Prozess für alle Berechtigungs-IDs, die Sie in den Attributen SIGNID, AUTHID oder COMAUTHID angegeben haben.

- c) Wenn Sie im Rahmen eines CICS-Kaltstarts oder CICS-Erststarts möglicherweise DB2CONN- und DB2ENTRY-Ressourcendefinitionen installieren müssen, in denen die Attribute SIGNID, AUTHID oder COMAUTHID enthalten sind, erteilen Sie der Benutzer-ID für die CICS-Region die Berechtigung, als Ersatz für alle Berechtigungs-IDs zu agieren, die durch diese Attribute angegeben werden.

In den Standardwerten der DB2CONN- und DB2ENTRY-Ressourcendefinitionen sind die Attribute AUTHID und COMAUTHID nicht enthalten. Der Standardwert für SIGNID für eine installierte DB2CONN-Definition ist die Anwendungs-ID der CICS-Region.

4. Für die AUTHTYPE-Sicherheit müssen Sie für jede Ihrer DB2CONN- oder DB2ENTRY-Ressourcendefinitionen in der allgemeinen RACF-Ressourcenklasse FACILITY ein Profil erstellen und den entsprechenden CICS-Benutzern oder -Benutzergruppen Leseberechtigung (READ) für die Profile erteilen. (Dieser Prozess imitiert den echten Mechanismus für die Ersatzsicherheit, es muss jedoch keine besondere Berechtigungs-ID verwendet werden; dieser Prozess schützt stattdessen alle Ressourcendefinitionen.) Gehen Sie dazu wie folgt vor:

- a) Erstellen Sie in der allgemeinen RACF-Ressourcenklasse FACILITY ein Profil für die DB2CONN- oder DB2ENTRY-Ressourcendefinition mit einem Namen im Format DFHDB2.AUTHTYPE.*berechtigungsname*, wobei *berechtigungsname* der Name der DB2CONN- oder DB2ENTRY-Ressourcendefinition ist.

Wenn Sie beispielsweise ein Profil für eine DB2CONN-Ressourcendefinition mit dem Namen DB2CONN1 definieren möchten, verwenden Sie den folgenden Befehl:

```
RDEFINE FACILITY DFHDB2.AUTHTYPE.DB2CONN1 UACC(NONE)
```

- b) Erteilen Sie entsprechenden CICS-Benutzern Leseberechtigung (READ) für das von Ihnen erstellte Profil.

Wenn Sie beispielsweise einem Benutzer mit der ID CICSUSR2 die Berechtigung erteilen möchten, das Attribut AUTHTYPE oder COMAUTHTYPE in einer DB2CONN-Ressourcendefinition mit dem Namen DB2CONN1 zu installieren oder zu ändern, verwenden Sie den folgenden Befehl:

```
PERMIT DFHDB2.AUTHTYPE.DB2CONN1 CLASS(FACILITY) ID(CICSUSR2) ACCESS(READ)
```

Wiederholen Sie diesen Prozess für jede Ihrer DB2CONN- und DB2ENTRY-Ressourcendefinitionen. Wenn es möglicherweise erforderlich ist, dass Sie DB2CONN- und DB2ENTRY-Ressourcendefinitionen, die die Attribute AUTHTYPE oder COMAUTHTYPE enthalten, im Rahmen eines CICS-Kaltstarts oder CICS-Erststarts installieren, müssen Sie der Benutzer-ID für die CICS-Region Leseberechtigung (READ) für die Profile dieser Ressourcendefinitionen erteilen.

Benutzerzugriff auf Db2-bezogene CICS-Transaktionen steuern

Mit dem Sicherheitsmechanismus der auf die Transaktion selbst bezogenen Sicherheitsprüfung (Transaction-attach Security) können Sie den Benutzerzugriff auf folgende Elemente steuern: auf CICS-Transaktionen, die auf Db2 zugreifen, um Daten abzurufen, auf die Transaktion DSNB und auf alle anderen Transaktionen, die Befehle der CICS-Db2-Anschlussfunktion und Db2-Befehle absetzen.

Informationen zu diesem Vorgang

Wenn der Sicherheitsmechanismus der auf die Transaktion selbst bezogenen Sicherheitsprüfung (Transaction-attach Security) aktiviert ist, wird von RACF oder einen äquivalenten externen Sicherheitsmanager überprüft, ob der CICS-Benutzer berechtigt ist, die angeforderte Transaktion auszuführen.

Um Db2-bezogene Transaktionen mithilfe der auf die Transaktion selbst bezogenen Sicherheitsprüfung zu schützen, folgen Sie den Anweisungen in [Transaktionssicherheit](#). Der Prozess ist für alle CICS-Transaktionen gleich, es gibt keine Besonderheiten für Db2-bezogene Transaktionen, soweit dies den Sicherheitsmechanismus der auf die Transaktion selbst bezogenen Sicherheitsprüfung betrifft. In den Anweisungen wird Folgendes beschrieben:

- Festlegen von entsprechenden Systeminitialisierungsparametern für die CICS-Region, um die auf die Transaktion selbst bezogene Sicherheitsprüfung zu aktivieren (siehe [CICS-Parameter zur Steuerung von auf die Transaktionen selbst bezogenen Sicherheitsprüfungen](#)).
- Definieren von Transaktionsprofilen für RACF hinsichtlich der Transaktionen, die Sie schützen möchten (siehe [RACF-Profil](#)).

Wenn Sie außer DSNB andere Transaktionen definiert haben, um Befehle der CICS-Db2-Anschlussfunktion und Db2-Befehle abzusetzen (zum Beispiel, wenn Sie zum Ausführen der einzelnen Befehle gesonderte Transaktionen erstellt haben), dürfen Sie nicht vergessen, diese Transaktionen auch für RACF zu definieren.

Sie können jetzt steuern, welche CICS-Benutzer Transaktionen verwenden können, die auf Db2 zugreifen. Fügen Sie die entsprechenden Benutzer oder Benutzergruppen mit Leseberechtigung (READ) der Zugriffsliste für Transaktionsprofile hinzu. In [RACF-Profil](#) finden Sie einige Empfehlungen hierzu.

Beachten Sie für Transaktionen, die Befehle der CICS-Db2-Anschlussfunktion und Db2-Befehle absetzen, Folgendes:

- Befehle der CICS-Db2-Anschlussfunktion bedienen sich der Verbindung zwischen CICS und Db2 und sie werden vollständig in CICS ausgeführt. Db2-Befehle werden in Db2 selbst ausgeführt und an Db2 wei-

tergeleitet. Sie können Db2-Befehle von Befehlen der CICS-Db2-Anschlussfunktion aufgrund des Bindestrichs (-) unterscheiden, der mit Db2-Befehlen eingegeben wird.

- Wenn Sie Zugriff auf die Transaktion DSNB haben, ermöglicht CICS Ihnen, alle Befehle der CICS-Db2-Anschlussfunktion sowie Db2-Befehle abzusetzen.
- Wenn Sie separate Transaktionen definiert haben, damit Sie einzelne Befehle der CICS-Db2-Anschlussfunktion sowie Db2-Befehle ausführen können, können Sie verschiedenen CICS-Benutzern Berechtigung auf Subsets von Transaktionscodes und damit auf Subsets der Befehle geben. Sie könnten beispielsweise einigen Benutzern die Berechtigung erteilen, Befehle der CICS-Db2-Anschlussfunktion, jedoch keine Db2-Befehle abzusetzen. In Von CICS bereitgestellte Transaktionen für CICS Db2 sind die Namen der gesonderten Transaktionsdefinitionen enthalten, die CICS für Befehle der CICS-Db2-Anschlussfunktion und für die Db2-Befehle bereitstellt.

Befehle der CICS-Db2-Anschlussfunktion fließen nicht an Db2, sie unterliegen also keiner weiteren Sicherheitsprüfung. Sie werden nur durch die auf die CICS-Transaktion selbst bezogene Sicherheitsprüfung (Transaction-attach Security) geschützt. Db2-Befehle und CICS-Transaktionen, die auf Db2 zugreifen, um Daten abzurufen, unterliegen weiteren Phasen der Sicherheitsprüfung, die durch Db2-Sicherheitsmechanismen wie folgt ausgeführt wird:

- Wenn eine Transaktion sich bei Db2 anmeldet, müssen für Db2 gültige Berechtigungs-IDs angegeben werden. Die Berechtigungs-IDs werden von RACF oder einem äquivalenten externen Sicherheitsmanager überprüft.
- Da die Transaktion einen Db2-Befehl absetzt oder auf Db2-Daten zugreift, müssen die von ihr angegebenen Berechtigungs-IDs über die Berechtigung verfügen, diese Aktionen innerhalb von Db2 auszuführen. Sie können in Db2 GRANT-Anweisungen verwenden, um den Berechtigungs-IDs die Berechtigung zum Ausführen von Aktionen zu erteilen.

Darüber hinaus muss die CICS-Region selbst für das Herstellen einer Verbindung zum Db2-Subsystem berechtigt sein.

In „Für Db2 Berechtigungs-IDs für die CICS-Region und für CICS-Transaktionen bereitstellen“ auf Seite 58 erfahren Sie, wie der CICS-Region die Berechtigung zum Herstellen einer Verbindung zum Db2-Subsystem erteilt wird und wie gültige Transaktions-IDs für Transaktionen angegeben werden.

In „Benutzer für den Zugriff auf Ressourcen in Db2 berechtigen“ auf Seite 66 erfahren Sie, wie den Berechtigungs-IDs, die die Transaktionen für Db2 angeben haben, Berechtigungen erteilt werden.

Für Db2 Berechtigungs-IDs für die CICS-Region und für CICS-Transaktionen bereitstellen

CICS hat zwei Typen von Prozessen, die Berechtigungs-IDs für Db2 bereitstellen müssen: die allgemeine Verbindung zwischen einer CICS-Region und Db2, und CICS-Transaktionen, die einen Thread in Db2 anfordern.

Informationen zu diesem Vorgang

Für Sicherheitszwecke verwendet Db2 die Bezeichnung 'Prozess', um alle Formen des Zugriffs auf Daten darzustellen, entweder durch Benutzer, die direkt mit Db2 interagieren, oder durch Benutzer, die mit Db2 über andere Programme einschließlich CICS interagieren. Ein Prozess, der zu Db2 eine Verbindung herstellt oder eine Anmeldung ausführt, muss mindestens eine kurze, als Berechtigungs-ID bezeichnete Db2-Kennung aufweisen, die für die Sicherheitsprüfung im Db2-Adressraum verwendet werden kann. Jeder Prozess muss eine primäre Berechtigungs-ID bereitstellen und er kann optional mindestens eine sekundäre Berechtigungs-ID bereitstellen. Db2-Zugriffsrechte und -Berechtigung kann den primären oder den sekundären Berechtigungs-IDs erteilt werden. Benutzer können beispielsweise eine Tabelle mithilfe ihrer sekundären Berechtigungs-ID erstellen. Der Eigner dieser Tabelle ist dann diese sekundäre Berechtigungs-ID. Jeder andere Benutzer, der für Db2 dieselbe sekundäre Berechtigungs-ID angibt, hat zugeordnete Zugriffsrechte für die Tabelle. Um einem Benutzer Zugriffsrechte zu entziehen, kann der Administrator den Benutzer von dieser Berechtigungs-ID trennen.

CICS hat zwei Typen von Prozessen, die Berechtigungs-IDs für Db2 bereitstellen müssen:

- Die allgemeine Verbindung zwischen einer CICS-Region und Db2, die von der CICS-Db2-Anschlussfunktion hergestellt wird. Dieser Prozess muss die Db2-Verbindungsverarbeitung durchlaufen, um für Db2 Berechtigungs-IDs bereitzustellen.
- CICS-Transaktionen, für die ein Thread in Db2 erforderlich ist. Dabei könnte es sich beispielsweise um eine Transaktion handeln, die Daten aus einer Db2-Datenbank abrufen, oder um die Transaktion DSNB, die einen Db2-Befehl absetzt. Bei jeder CICS-Transaktion ist der tatsächliche Prozess, den Db2 erkennt, der Thread-Tasksteuerblock, mit dem CICS den Thread einer Transaktion in Db2 steuert. Diese Prozesse müssen die Db2-Anmeldeverarbeitung durchlaufen, um Berechtigungs-IDs für Db2 bereitzustellen.

Während der Verbindungsverarbeitung und der Anmeldeverarbeitung legt Db2 die primären und sekundären Berechtigungs-IDs für den Prozess fest, die im Db2-Adressraum verwendet werden. Db2 verwendet standardmäßig die vom Prozess bereitgestellten Berechtigungs-IDs. Sowohl an der Verbindungsverarbeitung als auch an der Anmeldeverarbeitung sind jedoch Exitroutinen beteiligt und diese Exitroutinen ermöglichen Ihnen, Einfluss auf die Festlegung der primären und sekundären Berechtigungs-IDs zu nehmen. Db2 verfügt über eine standardmäßige Exitroutine für die Verbindung und über eine standardmäßige Exitroutine für die Anmeldung. Sie können diese durch Ihre eigenen Exitroutinen ersetzen; Beispiele für eine Exitroutine für die Verbindung und für eine Exitroutine für die Anmeldung sind als Unterstützung für Sie im Lieferumfang von Db2 enthalten.

Für Db2 Berechtigungs-IDs für eine CICS-Region bereitstellen

CICS stellt für Db2 eine primäre Berechtigungs-ID und mindestens eine sekundäre Berechtigungs-ID bereit.

Informationen zu diesem Vorgang

Wenn die CICS-Db2-Anschlussfunktion die allgemeine Verbindung zwischen einer CICS-Region und Db2 herstellt, durchläuft der Prozess die Db2-Verbindungsverarbeitung. Die CICS-Region kann Folgendes bereitstellen:

- Eine primäre Berechtigungs-ID. Die primäre Berechtigungs-ID wird in Db2 zur primären ID der CICS-Region. Für die Verbindung zwischen einer CICS-Region und Db2 können Sie nicht die primäre Berechtigungs-ID auswählen, die ursprünglich an die Db2-Verbindungsverarbeitung übergeben wurde; dabei handelt es sich um die Benutzer-ID der CICS-Region. Es ist jedoch möglich, die während der Verbindungsverarbeitung von Db2 festgelegte primäre ID zu ändern, indem Sie Ihre eigene Exitroutine für die Verarbeitung schreiben. Wenn RACF oder ein funktional entsprechender externer Sicherheitsmanager aktiv ist, muss die Benutzer-ID für die CICS-Region für diesen definiert sein. In [„Primäre Berechtigungs-ID für eine CICS-Region bereitstellen“](#) auf Seite 59 finden Sie Informationen zu den möglichen primären Berechtigungs-IDs für eine CICS-Region.
- Mindestens eine sekundäre Berechtigungs-ID. Sie können den Namen einer RACF-Gruppe oder RACF-Gruppenliste als sekundäre Berechtigungs-IDs für die CICS-Region verwenden. Falls Sie dies vornehmen, müssen Sie die standardmäßige Exitroutine DSNB3@ATH für die Db2-Verbindung ersetzen, die nur primäre Berechtigungs-IDs an Db2 übergibt. Die Beispiel-Exitroutine DSNB3SATH für die Db2-Verbindung übergibt die Namen von RACF-Gruppen als sekundäre Berechtigungs-IDs an Db2. Alternativ dazu können Sie Ihre eigene Exitroutine für Verbindungen schreiben, die sekundäre Berechtigungs-IDs für die CICS-Region festlegt. In [„Sekundäre Berechtigungs-IDs für eine CICS-Region bereitstellen“](#) auf Seite 60 finden Sie Informationen dazu, wie sekundäre Berechtigungs-IDs für eine CICS-Region festgelegt werden.

Primäre Berechtigungs-ID für eine CICS-Region bereitstellen

Die primäre Berechtigungs-ID, die an Db2 übergeben wird, hängt davon ab, ob CICS als gestartete Task, als gestarteter Job oder als Job ausgeführt wird.

Informationen zu diesem Vorgang

Der Verbindungstyp, den CICS von Db2 anfordert, ist SASS (Single Address Space Subsystem, Subsystem mit einem einzigen Adressraum). Für die Verbindung zwischen einer CICS-Region und Db2 können Sie nicht die primäre Berechtigungs-ID verwenden, die ursprünglich an die Db2-Verbindungsverarbeitung

übergeben wurde. Die ID, die als primäre Berechtigungs-ID für die CICS-Region an Db2 übergeben wird, ist eine der folgenden:

- Die Benutzer-ID aus der RACF-Tabelle mit gestarteten Prozeduren ICHRIN03, wenn CICS als gestartete Task ausgeführt wird.
- Der Benutzerparameter des Segments STDATA im allgemeinen Ressourcenklassenprofil STARTED, wenn CICS als gestarteter Job ausgeführt wird.
- Die Benutzer-ID, die im Parameter USER der Jobkarte (JOB) angegeben ist, wenn CICS als Job ausgeführt wird.

Die Benutzer-ID, die eine CICS-Region möglicherweise verwendet, muss für RACF bzw. für Ihren äquivalenten externen Sicherheitsmanager definiert werden, falls der externe Sicherheitsmanager aktiv ist. Definieren Sie die Benutzer-ID für RACF als Benutzerprofil (USER). Es reicht nicht aus, sie als Ressourcenprofil (RESOURCE) zu definieren.

Sobald Sie die Benutzer-ID für die CICS-Region für RACF definiert haben, müssen Sie sie wie folgt für den Zugriff auf Db2 berechtigen:

1. Definieren Sie in der RACF-Klasse DSNR ein Profil für das Db2-Subsystem mit dem Verbindungstyp SASS (Single Address Space Subsystem). Beispiel: Mit dem folgenden RACF-Befehl wird ein Profil für SASS-Verbindungen zum Db2-Subsystem DB2A in der Klasse DSNR erstellt:

```
RDEFINE DSNR (DB2A.SASS) OWNER(DB2OWNER)
```

2. Erteilen Sie der Benutzer-ID für die CICS-Region den Zugriff auf das Db2-Subsystem. Mit dem folgenden RACF-Befehl beispielsweise wird eine CICS-Region mit der Benutzer-ID CICSHA11 für das Herstellen einer Verbindung zum Db2-Subsystem DB2A berechtigt:

```
PERMIT DB2A.SASS CLASS(DSNR) ID(CICSHA11) ACCESS(READ)
```

Die Db2-Exitroutine für Verbindungen verwendet die von der CICS-Region bereitgestellte primäre Berechtigungs-ID (die Benutzer-ID) und legt sie als die primäre ID für die CICS-Region in Db2 fest. Die standardmäßige Db2-Exitroutine für die Verbindung DSN3@ATH und die Db2-Beispiel-Exitroutine für die Verbindung DSN3SATH zeigen beide dieses Verhalten. Es ist möglich, die von Db2 festgelegte primäre ID zu ändern, indem Sie Ihre eigene Exitroutine für Verbindungen schreiben. Weitere Informationen zur Beispiel-Exitroutine für Verbindungen und zum Schreiben von Exitroutinen finden Sie in [Db2 schützen in der Produktdokumentation zu Db2 for z/OS](#). Möglicherweise ist es jedoch einfacher, sekundäre Berechtigungs-IDs für die CICS-Region bereitzustellen und auf deren Basis Berechtigungen für die CICS-Region zu erteilen, anstatt auf Basis der primären Berechtigungs-IDs.

Sekundäre Berechtigungs-IDs für eine CICS-Region bereitstellen

Wenn eine CICS-Region eine Verbindung zu Db2 herstellt, kann sie zusätzlich zur primären Berechtigungs-ID mindestens eine sekundäre Berechtigungs-ID für Db2 bereitstellen. Sie können den Namen der RACF-Gruppe oder der RACF-Gruppenliste, mit der die CICS-Region verbunden ist, als sekundäre Berechtigungs-IDs verwenden. Dadurch können Sie RACF-Gruppen Db2-Zugriffsrechte und Berechtigungen erteilen und anschließend mehrere CICS-Regionen mit denselben Gruppen verbinden, anstatt allen möglichen primären Berechtigungs-IDs für die einzelnen CICS-Regionen Db2-Zugriffsrechte zu erteilen.

Informationen zu diesem Vorgang

Um den Namen der RACF-Gruppe oder der RACF-Gruppenliste, mit der eine CICS-Region verbunden ist, als sekundäre Berechtigungs-ID für Db2 bereitzustellen, führen Sie folgende Schritte aus.

Vorgehensweise

1. Geben Sie als Systeminitialisierungsparameter für die CICS-Region SEC=YES an, um sicherzustellen, dass CICS RACF verwendet.
2. Verbinden Sie die CICS-Region mit der entsprechenden RACF-Gruppe oder RACF-Gruppenliste. Siehe [RACF-Gruppenprofile](#).

3. Ersetzen Sie die standardmäßige Db2-Exitroutine für Verbindungen DSN3@ATH.

Diese Exitroutine wird während der Verbindungsverarbeitung ausgelöst. Die standardmäßige Exitroutine für Verbindungen unterstützt keine sekundären Berechtigungs-IDs, deshalb müssen Sie sie ersetzen durch die Beispiel-Exitroutine für die Anmeldung DSN3SATH, die im Lieferumfang von Db2 enthalten ist, oder durch ihre eigene Routine. DSN3SATH wird im Quellenformat in der Db2-Bibliothek SDSNSAMP geliefert und Sie können sie als Basis für Ihre eigene Routine verwenden. DSN3SATH übergibt die Namen der RACF-Gruppen, mit denen die CICS-Region verbunden ist, als sekundäre Berechtigungs-IDs an Db2. Wenn die Option für die RACF-Gruppenliste aktiv ist, ruft DSN3SATH alle Gruppennamen ab, mit denen die CICS-Region verbunden ist, und verwendet sie als sekundäre Berechtigungs-IDs. Wenn die Option für die RACF-Gruppenliste nicht aktiv ist, verwendet DSN3SATH den Namen der zurzeit mit der CICS-Region verbundenen Gruppe als einzige sekundäre Berechtigungs-ID.

Ergebnisse

Wenn die CICS-Region eine Verbindung zu Db2 herstellt, legt die Beispiel-Exitroutine für Verbindungen die primäre Berechtigungs-ID der CICS-Region (die Benutzer-ID der Region) als primäre ID fest und legt die Namen der RACF-Gruppen, mit denen die CICS-Region verbunden ist, als sekundäre IDs fest.

Nächste Schritte

Als Alternative dazu, die Namen von RACF-Gruppen als sekundäre Berechtigungs-IDs für Db2 bereitzustellen, können Sie Ihre eigenen Exitroutine für Verbindungen schreiben, mit der sekundäre IDs für die CICS-Region festgelegt werden. Informationen zum Schreiben von Exitroutinen für Verbindungen finden Sie in [Db2 schützen in der Produktdokumentation zu Db2 for z/OS](#).

Db2 Berechtigungs-IDs für CICS-Transaktionen bereitstellen

Damit CICS Berechtigungs-IDs an Db2 übergeben kann, muss RACF von CICS verwendet werden; im Systeminitialisierungsparameter muss SEC=YES angegeben sein. Der Grund hierfür liegt darin, dass CICS ein RACF-Zugriffssteuerungsumgebungselement (ACEE) an Db2 übergeben muss.

Informationen zu diesem Vorgang

Wenn sich der Thread-Tasksteuerblock einer CICS-Transaktion bei Db2 anmeldet und die Db2-Anmeldeverarbeitung durchläuft, kann er Folgendes bereitstellen:

- Eine primäre Berechtigungs-ID. Für CICS-Transaktionen können Sie die primäre Berechtigungs-ID auswählen. Dabei kann es sich um die Benutzer-ID oder die Operator-ID des CICS-Benutzers oder um eine Terminal-ID oder eine Transaktions-ID handeln; oder es kann sich um eine ID handeln, die Sie angegeben haben. Die ID, die als primäre Berechtigungs-ID verwendet wird, ist durch Attribute der DB2ENTRY-Definition (für Einstiegsthreads) oder der DB2CONN-Definition (für Pool-Threads und Befehlsthreads) festgelegt. In [„Primäre Berechtigungs-ID für CICS-Transaktionen bereitstellen“](#) auf Seite 62 erfahren Sie, wie Sie die primäre Berechtigungs-ID für eine CICS-Transaktion auswählen.
- Mindestens eine sekundäre Berechtigungs-ID. Sie können als sekundäre Berechtigungs-ID den Namen einer RACF-Gruppe oder einer RACF-Gruppenliste verwenden. Der Vorteil hierbei ist, dass Sie RACF-Gruppen und nicht jedem einzelnen CICS-Benutzer Db2-Zugriffsrechte und Berechtigungen erteilen können. Zum Verwenden von sekundären Berechtigungs-IDs verwenden Sie das Attribut AUTHTYPE in der DB2ENTRY-Definition (für Einstiegsthreads) oder das Attribut AUTHTYPE oder COMAUTHTYPE in der DB2CONN-Definition (für Pool-Threads oder Befehlsthreads), um die Option GROUP anzugeben. Sie müssen auch die standardmäßige Db2-Exitroutine für die Anmeldung DSN3@SGN ersetzen, da die Standardroutine keine sekundären Berechtigungs-IDs an Db2 übergibt. Wenn Sie die Option GROUP angeben, wird die primäre Berechtigungs-ID automatisch als die Benutzer-ID des CICS-Benutzers definiert, der der Transaktion zugeordnet ist. In [„Sekundäre Berechtigungs-IDs für CICS-Transaktionen bereitstellen“](#) auf Seite 64 erfahren Sie, wie sekundäre Berechtigungs-IDs eingerichtet und verwendet werden.

Ein wichtiger Aspekt bei der Auswahl der Berechtigungs-IDs, die CICS-Transaktionen für Db2 bereitstellen, ist der Sicherheitsmechanismus, den Sie für die Sicherheitsprüfung im Db2-Adressraum ausgewählt

haben. Diese Sicherheitsprüfung umfasst den Zugriff auf Befehle, Pläne und dynamisches SQL in Db2. Sie können auswählen, dass diese Sicherheitsprüfung durch folgende Elemente ausgeführt wird:

- Interne Db2-Sicherheit.
- RACF oder ein äquivalenter externer Sicherheitsmanager.
- Zum Teil durch Db2 und zum Teil durch RACF.

Wenn Sie für einen Teil der Sicherheitsprüfung bzw. für die gesamte Sicherheitsprüfung in Ihrem Db2-Adressraum RACF verwenden, **müssen** CICS-Transaktionen, die sich bei Db2 anmelden, eine Berechtigungs-ID mithilfe einer der folgenden Methoden bereitstellen:

- AUTHTYPE(USERID) oder COMAUTHTYPE(USERID) in der entsprechenden Definition für den Thread (DB2ENTRY oder DB2CONN) angeben, um die Benutzer-ID des CICS-Benutzers, der der Transaktion zugeordnet ist, als primäre Berechtigungs-ID für Db2 bereitzustellen.
- AUTHTYPE(GROUP) oder COMAUTHTYPE(GROUP) in der entsprechenden Definition für den Thread (DB2ENTRY oder DB2CONN) angeben, um die Benutzer-ID des CICS-Benutzers, der der Transaktion zugeordnet ist, als primäre Berechtigungs-ID für Db2 und den Namen einer RACF-Gruppe oder RACF-Gruppenliste als sekundäre Berechtigungs-IDs bereitzustellen.
- AUTHTYPE(SIGN) in der entsprechenden Definition für den Thread (DB2ENTRY oder DB2CONN) angeben und die Benutzer-ID für die CICS-Region im Attribut SIGNID der DB2CONN-Definition angeben, um die Benutzer-ID für die CICS-Region als primäre Berechtigungs-ID für Db2 bereitzustellen.

Wenn das RACF-Zugriffssteuerungsumgebungselement (ACEE) in der CICS-Region auf eine Art und Weise geändert wurde, die sich auf die CICS-Db2-Anschlussfunktion auswirkt, müssen Sie beachten, dass Db2 die Änderung erst erkennt, wenn eine Anmeldung stattfindet. Sie können den Befehl CEMT oder EXEC CICS SET DB2CONN SECURITY(REBUILD) verwenden, damit die CICS-Db2-Anschlussfunktion eine Db2-Anmeldung ausgibt, wenn der nächste Thread wiederverwendet wird oder ein Thread in einem bereits angemeldeten Tasksteuerblock erstellt wird. Dadurch wird sichergestellt, dass Db2 die Sicherheitsänderung erkennt.

Primäre Berechtigungs-ID für CICS-Transaktionen bereitstellen

Wenn sich der Thread-Tasksteuerblock einer CICS-Transaktion bei Db2 anmeldet, muss er eine sekundäre Berechtigungs-ID für Db2 angeben. Die von einer Transaktion als primäre Berechtigungs-ID verwendete ID wird durch ein Attribut in der Ressourcendefinition des Threads festgelegt, den die Transaktion für den Zugriff auf Db2 verwendet.

Informationen zu diesem Vorgang

Dies bedeutet, dass alle Transaktionen, die denselben Threadtyp (denselben Typ von Einstiegsthread oder Pool-Thread oder Befehlsthread) verwenden, auch denselben Typ von primärer Berechtigungs-ID verwenden müssen. Sie müssen in jeder CICS-Region eine primäre Berechtigungs-ID für Folgendes festlegen:

- Alle Typen von Einstiegsthread, mithilfe Ihrer DB2ENTRY-Definitionen.
- Die Pool-Threads, mithilfe Ihrer DB2CONN-Definition.
- Die Befehlsthreads (die für die Transaktion DSNCR verwendet werden), mithilfe Ihrer DB2CONN-Definition.

Stellen Sie vor dem Festlegen von primären Berechtigungs-IDs sicher, dass Sie über die entsprechende Berechtigung verfügen. Wenn für die CICS-Region die Ersatzbenutzerprüfung aktiv ist (das heißt, für den Systeminitialisierungsparameter XUSER ist YES festgestellt), müssen Sie nicht nur die Berechtigung haben, Ihre DB2CONN- oder DB2ENTRY-Definitionen zu ändern, sondern Sie müssen auch eine Sonderberechtigung anfordern, um Operationen ausführen zu können, an denen Db2-Berechtigungs-IDs beteiligt sind. Mit diesen Operationen werden die Attribute AUTHID, COMAUTHID, AUTHTYPE oder COMAUTHTYPE in einer DB2ENTRY- oder DB2CONN-Definition und das Attribut SIGNID in einer DB2CONN-Definition geändert. In „Ersatzsicherheit und AUTHTYPE-Sicherheit zum Steuern des Zugriffs auf die Berechtigungs-IDs verwenden, die CICS für Db2 bereitstellt“ auf Seite 55 erfahren Sie, wie Benutzern die Berechtigung zum Ausführen dieser Operationen erteilt wird.

Zum Festlegen der primären Berechtigungs-ID für einen bestimmten Threadtyp gibt es zwei Methoden:

1. Verwenden Sie das Attribut AUTHID in der DB2ENTRY-Definition (für Einstiegsthreads) oder das Attribut AUTHID oder COMAUTHID in der DB2CONN-Definition (für Pool-Threads oder Befehlsthreads), um eine primäre Berechtigungs-ID anzugeben. Sie könnten beispielsweise AUTHID=test2 definieren. In diesem Fall übergibt die CICS-Db2-Anschlussfunktion die Zeichen TEST2 als primäre Berechtigungs-ID an Db2.

Bei Verwendung von AUTHID oder COMAUTHID ist die Verwendung sekundärer Berechtigungs-IDs nicht zulässig und dies ist auch nicht kompatibel mit der Verwendung von RACF oder einem äquivalenten, externen Sicherheitsmanager für die Sicherheitsprüfung im Db2-Adressraum.

2. Verwenden Sie das Attribut AUTHTYPE in der DB2ENTRY-Definition (für Einstiegsthreads) oder das Attribut AUTHTYPE oder COMAUTHTYPE in der DB2CONN-Definition (für Pool-Threads oder Befehlsthreads), um CICS anzuweisen, eine vorhandene, für die Transaktion relevante ID als primäre Berechtigungs-ID zu verwenden. Bei dieser ID kann es sich um eine CICS-Benutzer-ID, eine Operator-ID, Terminal-ID oder Transaktions-ID handeln, oder es kann eine ID sein, die Sie in der DB2CONN-Definition für die CICS-Region angegeben haben.

Die Verwendung von AUTHTYPE oder COMAUTHTYPE ist mit der Verwendung von RACF (oder einem äquivalenten externen Sicherheitsmanager) für die Sicherheitsprüfung im Db2-Adressraum kompatibel, wenn Sie die Optionen USERID oder GROUP verwenden, und sie ist mit der Verwendung von sekundären Berechtigungs-IDs kompatibel, wenn Sie die Option GROUP verwenden.

Die beiden Methoden zum Festlegen der primären Berechtigungs-ID schließen sich gegenseitig aus; Sie können in derselben Ressourcendefinition nicht sowohl AUTHID als auch AUTHTYPE oder COMAUTHID und COMAUTHTYPE angeben.

Vergessen Sie nicht, dass alle als primäre Berechtigungs-IDs ausgewählte Berechtigungs-IDs für RACF oder Ihren äquivalenten externen Sicherheitsmanager definiert sein müssen, wenn der Sicherheitsmanager für das Db2-Subsystem aktiv ist. Für RACF müssen die primären Berechtigungs-IDs als USER-Profil von RACF und nicht nur als RESOURCE-Profil (beispielsweise als Terminal oder Transaktion) definiert sein.

Folgen Sie den Anweisungen in [DB2CONN-Ressourcen](#) und [DB2ENTRY-Ressourcen](#), um DB2CONN- und DB2ENTRY-Definitionen einzurichten oder zu ändern. Wenn Sie die Attribute AUTHTYPE oder COMAUTHTYPE zum Festlegen der primären Berechtigungs-ID für einen Threadtyp verwenden, müssen Sie anhand der [Tabelle 3 auf Seite 64](#) die Optionen angeben, die die erforderliche Berechtigungs-ID bereitstellen und die von Ihnen gewünschten Funktionen unterstützen. Die wichtigsten Punkte, die berücksichtigt werden müssen, sind folgende:

- Wenn Sie für Db2 sowohl sekundäre Berechtigungs-IDs als auch eine primäre Berechtigungs-ID angeben wollen, müssen Sie die Option GROUP auswählen. Wenn Sie die Option GROUP angeben, wird die primäre Berechtigungs-ID automatisch als Ihre CICS-Benutzer-ID definiert, aber Sie können als Basis für Ihre Sicherheitsprüfung stattdessen die sekundären Berechtigungs-IDs auswählen.
- Wenn Sie für die Sicherheitsprüfung im Db2-Adressraum RACF verwenden, müssen Sie entweder die Option GROUP oder USERID verwenden. Nur diese Optionen können das RACF-Zugriffssteuerungsumgebungselement (ACEE) an Db2 übergeben, das erforderlich ist, wenn für die Sicherheitsprüfung RACF verwendet wird.
- Bedenken Sie, wie sich Ihre Wahl der Berechtigungs-ID auf die Leistung und Wartung auswirkt. In [Berechtigungs-IDs für Leistung und Wartung auswählen](#) wird dies dargestellt. Mit den Optionen USERID, OPID, TERM, TX oder GROUP tritt häufiger eine Anmeldeverarbeitung auf und für die Wartung wird mehr Zeit benötigt, da Sie einer größeren Anzahl von Berechtigungs-IDs Berechtigungen erteilen müssen. Mit der Option SIGN oder der Verwendung des Attributs AUTHID statt des Attributs AUTHTYPE ist der Umfang der Anmeldeverarbeitung geringer und die Wartung ist einfacher. Durch die Verwendung der standardmäßigen Berechtigungs-IDs fällt die Db2-Sicherheitsprüfung jedoch weniger differenziert aus.
- Bedenken Sie, wie sich Ihre Wahl der Berechtigungs-ID auf die Abrechnung auswirkt. Die Berechtigungs-ID wird in jedem Db2-Abrechnungsdatensatz verwendet. Vom Standpunkt der Abrechnung aus werden die detailliertesten Informationen erzielt, wenn USERID, OPID, GROUP oder TERM verwendet wird. Je nach der ACCOUNTREC-Spezifikation ist es jedoch möglicherweise nicht in jedem Fall möglich, eine Abrechnung auf der Ebene des einzelnen Benutzers zu erstellen. Weitere Informationen zur Ab-

rechnung in einer CICS-Db2-Umgebung finden Sie in [Abrechnung und Überwachung in einer CICS-Db2-Umgebung unter "Überwachen"](#).

Tabelle 3 auf Seite 64 zeigt die primären Berechtigungs-IDs, die die CICS-Db2-Anschlussfunktion an Db2 übergibt, wenn Sie die einzelnen Optionen für das Attribut AUTHTYPE oder COMAUTHTYPE auswählen.

Tabelle 3. Für die Attribute AUTHTYPE und COMAUTHTYPE verfügbare Optionen			
Option	Primäre Berechtigungs-ID, die an Db2 übergeben wird	Wird die RACF-Prüfung für Db2 unterstützt?	Werden sekundäre Berechtigungs-IDs unterstützt?
USERID	Der CICS-Transaktion zugeordnete Benutzer-ID, die in RACF definiert ist und bei der CICS-Anmeldung verwendet wird	Ja	Nein
OPID	CICS-Operator-ID des Benutzers, die im CICS-Segment des RACF-Benutzerprofils definiert ist	Nein	Nein
SIGN	Eine ID, die Sie in dem Attribut SIGNID der DB2CONN-Definition für die CICS-Region angegeben haben. Standardmäßig wird als Wert die Anwendungs-ID der CICS-Region angenommen.	Ja, wenn das Attribut SIGNID der DB2CONN-Ressource mit der Benutzer-ID der CICS-Region übereinstimmt.	Nein
TERM	Terminal-ID des Terminals, das der Transaktion zugeordnet ist	Nein	Nein
TX	Transaktions-ID	Nein	Nein
GROUP	CICS-RACF-Benutzer-ID des Benutzers, die bei der CICS-Anmeldung verwendet wird	Ja	Ja

Wenn Sie nicht planen, sekundäre Berechtigungs-IDs für Ihre CICS-Transaktionen anzugeben, müssen Sie die standardmäßige Db2-Exitroutine für die Anmeldung DSN3@SGN nicht ersetzen. Die standardmäßige Exitroutine für die Anmeldung verarbeitet primäre Berechtigungs-IDs. Das Db2-Subsystem, zu dem Sie eine Verbindung herstellen, verwendet aus einem anderen Grund jedoch möglicherweise eine andere Exitroutine für die Anmeldung. Wenn das Db2-Subsystem die Beispiel-Exitroutine für die Anmeldung DSN3SSGN verwendet, müssen Sie eine Änderung an DSN3SSGN vornehmen, wenn **alle** der folgenden Bedingungen zutreffen:

- Sie haben für AUTHID oder AUTHTYPE eine andere Option als GROUP ausgewählt.
- Die Verarbeitung von RACF-Gruppenlisten ist aktiv.
- Sie verfügen über Transaktionen, deren primäre Berechtigungs-ID nicht für RACF definiert wurde.

Ist dies der Fall, lesen Sie in [Db2 schützen in der Produktdokumentation zu Db2 for z/OS](#) nach, welche Änderungen Sie an der Beispiel-Exitroutine für die Anmeldung vornehmen müssen.

Sekundäre Berechtigungs-IDs für CICS-Transaktionen bereitstellen

Wenn sich ein Thread-Tasksteuerblock, der zu einer CICS-Transaktion gehört, bei Db2 anmeldet, kann er zusätzlich zur primären Berechtigungs-ID mindestens eine sekundäre Berechtigungs-ID für Db2 bereitstellen.

Informationen zu diesem Vorgang

Sie können als sekundäre Berechtigungs-IDs für Db2 den Namen der RACF-Gruppe oder der RACF-Gruppenliste eines CICS-Benutzers bereitstellen. Der Vorteil hierbei ist, dass Sie RACF-Gruppen und nicht jedem einzelnen CICS-Benutzer Db2-Zugriffsrechte und Berechtigungen erteilen können. CICS-Benutzer

können dann nach Bedarf mit RACF-Gruppen verbunden oder aus RACF-Gruppen entfernt werden. In RACF-Gruppenprofile wird die Vorgehensweise für das Platzieren von Benutzern in RACF-Gruppen erläutert.

Sie können sekundäre Berechtigungs-IDs für Db2 nur für CICS-Transaktionen bereitstellen, wenn Sie in der DB2ENTRY-Definition die Option GROUP für das Attribut AUTHTYPE (für Einstiegsthreads) oder das Attribut AUTHTYPE oder COMAUTHTYPE in der DB2CONN-Definition (für Pool-Threads oder Befehlsthreads) angeben. Wenn Sie für AUTHTYPE oder COMAUTHTYPE eine beliebige andere Option angeben, werden für die sekundäre Berechtigungs-ID Leerzeichen festgelegt. Wenn Sie die Option GROUP angeben, können Sie die primäre Berechtigungs-ID nicht für den Threadtyp auswählen; sie wird automatisch als Benutzer-ID des CICS-Benutzers definiert, der der Transaktion zugeordnet ist. Sie sollten als Basis für Ihre Sicherheitsprüfung stattdessen die sekundären Berechtigungs-IDs nehmen.

Führen Sie folgende Schritte aus, um die Namen der RACF-Gruppen eines Benutzers als sekundäre Berechtigungs-IDs für Db2 bereitzustellen:

1. Geben Sie als Systeminitialisierungsparameter für die CICS-Region SEC=YES an, um sicherzustellen, dass CICS RACF verwendet. Wenn die Namen Ihrer CICS-Transaktionsprofile mit einem Präfix definiert sind, müssen Sie zusätzlich den Systeminitialisierungsparameter SECPRFX=präfix angeben.
2. Wenn die CICS-Region den Mehrregionenbetrieb verwendet:
 - a. Stellen Sie sicher, dass jede verbundene CICS-Region auch die RACF-Sicherheit verwendet (SEC=YES).
 - b. Geben Sie in der Definition CONNECTION für die Terminalverwaltungsregion (TOR) ATTACH-SEC=IDENTIFY an, um sicherzustellen, dass die Anmeldeinformationen von der Terminalverwaltungsregion an die Anwendungsverwaltungsregionen (AORs) weitergegeben werden.
3. Ersetzen Sie die standardmäßige Db2-Exitroutine für die Anmeldung DSN3@SGN. Diese Exitroutine für die Anmeldung wird während der Anmeldeverarbeitung ausgelöst. Die standardmäßige Exitroutine für die Anmeldung unterstützt keine sekundären Berechtigungs-IDs, deshalb müssen Sie sie ersetzen durch die Beispiel-Exitroutine für die Anmeldung DSN3SSGN, die im Lieferumfang von Db2 enthalten ist, oder durch ihre eigene Routine. DSN3SSGN wird im Quellenformat in der Db2-Bibliothek SDSNSAMP geliefert und Sie können sie als Basis für Ihre eigene Routine verwenden. Informationen zu Anmeldeexits und zum Schreiben von Exitroutinen finden Sie in Db2 schützen in der Produktdokumentation zu Db2 for z/OS. Wenn die Option für die RACF-Gruppenliste nicht aktiv ist, übergibt DSN3SSGN den Namen der aktuell verbundenen Gruppe als sekundäre Berechtigungs-ID. Wenn die Option für die RACF-Gruppenliste aktiv ist, ruft DSN3SSGN die Namen aller Gruppen ab, mit denen der Benutzer verbunden ist, und übergibt sie als sekundäre Berechtigungs-IDs an Db2.
4. Verwenden Sie das Attribut AUTHTYPE in der DB2ENTRY-Definition (für Einstiegsthreads) oder das Attribut AUTHTYPE oder COMAUTHTYPE in der DB2CONN-Definition (für Pool-Threads oder Befehlsthreads), um die Option GROUP als Berechtigungstyp für alle Threadtypen anzugeben, für die Sie sekundäre Berechtigungs-IDs bereitstellen wollen. Wenn für die CICS-Region die Ersatzbenutzerprüfung aktiv ist (das heißt, für den Systeminitialisierungsparameter XUSER ist YES festgelegt), müssen Sie eine Sonderberechtigung anfordern, um Operationen auszuführen, bei denen Db2-Berechtigungs-IDs beteiligt sind. Dies wird in „Ersatzsicherheit und AUTHTYPE-Sicherheit zum Steuern des Zugriffs auf die Berechtigungs-IDs verwenden, die CICS für Db2 bereitstellt“ auf Seite 55 erläutert.

Wenn Sie alle oben aufgelisteten Schritte erfolgreich ausgeführt haben, wird die Benutzer-ID des CICS-Benutzers als primäre Berechtigungs-ID an Db2 übergeben und die RACF-Gruppe oder -Gruppenliste des Benutzers wird als sekundäre Berechtigungs-IDs an Db2 übergeben, wenn sich der Thread-Tasksteuerblock einer CICS-Transaktion bei Db2 mit den Threadtypen anmeldet, die Sie mit der Option GROUP definiert haben. Wenn Sie nicht alle Schritte erfolgreich abgeschlossen haben, gibt die CICS-Db2-Anschlussfunktion die Nachricht aus, dass die Berechtigung fehlgeschlagen ist.

Als Alternative zur Bereitstellung der Namen von RACF-Gruppen als sekundäre Berechtigungs-IDs für Db2 können Sie Ihre eigene Exitroutine für die Anmeldung schreiben, mit der sekundäre IDs für CICS-Transaktionen festgelegt werden. Siehe Db2 schützen in der Produktdokumentation zu Db2 for z/OS.

Benutzer für den Zugriff auf Ressourcen in Db2 berechtigen

Sobald ein Benutzer über eine CICS-Transaktion auf den Db2-Adressraum zugegriffen hat, benötigt er möglicherweise eine Berechtigung, um Db2-Befehle abzusetzen oder einen Plan auszuführen.

Informationen zu diesem Vorgang

Der Zugriff auf Db2-Ressourcen, die ein CICS-Benutzer benötigt, um Db2-Befehle abzusetzen oder einen Plan auszuführen, unterliegt der Sicherheitsprüfung, die mit Db2-Sicherheitsmechanismen ausgeführt wird. Sie können auswählen, dass diese Sicherheitsprüfung durch folgende Elemente ausgeführt wird:

- Interne Db2-Sicherheit.
- RACF oder ein äquivalenter externer Sicherheitsmanager.
- Zum Teil durch Db2 und zum Teil durch RACF.

Weitere Informationen zur Einrichtung von RACF für die Ausführung der Sicherheitsprüfung im Db2-Adressraum finden Sie in [Db2 schützen in der Produktdokumentation zu Db2 for z/OS](#).

Wenn Sie RACF für einen Teil bzw. der gesamten Sicherheitsprüfung in Ihrem Db2-Adressraum verwenden, dürfen Sie nicht vergessen, dass CICS-Transaktionen, die sich bei Db2 anmelden, eine Berechtigungs-ID benötigen. Weitere Informationen hierzu finden Sie in [„Db2 Berechtigungs-IDs für CICS-Transaktionen bereitstellen“](#) auf Seite 61. CICS muss darüber hinaus RACF verwenden (in der Systeminitialisierungstabelle (SIT) muss SEC=YES angegeben sein). Der Grund hierfür liegt darin, dass bei Verwendung von RACF für die Sicherheitsprüfung im Db2-Adressraum CICS ein RACF-Zugriffssteuerungsumgebungselement (ACEE – Access Control Environment Element) an Db2 übergeben muss. CICS kann ein Zugriffsteuerungsumgebungselement nur erzeugen, wenn RACF aktiv ist und nur mit der Option GROUP, SIGN oder USERID definierte Threads können das Zugriffsteuerungsumgebungselement an Db2 übergeben.

Nach Übergabe des Zugriffsteuerungsumgebungselements an Db2 wird es vom Db2-Exit DSNX@XAC verwendet, der festlegt, ob für die Sicherheitsprüfung RACF oder ein äquivalenter externer Sicherheitsmanager, der nicht von IBM stammt, oder die interne Db2-Sicherheit verwendet wird. DSNX@XAC wird ausgelöst, wenn eine Transaktion, deren Thread sich bei Db2 angemeldet hat, API-Anfragen ausgibt. Sie können DSNX@XAC ändern. Weitere Informationen hierzu finden Sie in [Db2 schützen in der Produktdokumentation zu Db2 for z/OS](#).

Db2 oder der externe Sicherheitsmanager führt die Sicherheitsprüfung mithilfe der Berechtigungs-IDs aus, die die CICS-Transaktion für Db2 bereitgestellt hat, als der von der Transaktion verwendete Thread sich bei Db2 anmeldete. Die Berechtigungs-IDs können sich auf den einzelnen CICS-Benutzer beziehen (beispielsweise die Benutzer-ID des CICS-Benutzers und die RACF-Gruppen, mit denen der Benutzer verbunden ist) oder sie können sich auf die Transaktion beziehen (beispielsweise die Terminal-ID oder die Transaktions-ID) oder sie können sich auf die gesamte CICS-Region beziehen. Weitere Informationen hierzu finden Sie in [„Für Db2 Berechtigungs-IDs für die CICS-Region und für CICS-Transaktionen bereitstellen“](#) auf Seite 58.

Von Db2 oder vom externen Sicherheitsmanager wird überprüft, ob Sie den Berechtigungs-IDs die Berechtigung erteilt haben, die relevanten Aktionen in Db2 auszuführen. Sie können den Berechtigungs-IDs diese Berechtigung erteilen, indem Sie in Db2 GRANT-Anweisungen verwenden. Informationen zur Vorgehensweise für die Erteilung und die Entziehung von Db2-Berechtigungen für Berechtigungs-IDs finden Sie in [Db2 schützen in der Produktdokumentation zu Db2 for z/OS](#).

Benutzerzugriff auf Db2-Befehle steuern

Für CICS-Benutzer wird die erste auf Db2-Befehle bezogene Sicherheitsprüfung im CICS-Adressraum ausgeführt, wenn der Benutzer versucht, auf eine CICS-Transaktion zuzugreifen, die Db2-Befehle absetzt. Dabei kann es sich um DSNB oder um eine benutzerdefinierte Transaktion handeln, die DFHD2CM1 aufruft und einen einzelnen Db2-Befehl ausführt.

Informationen zu diesem Vorgang

In [„Benutzerzugriff auf Db2-bezogene CICS-Transaktionen steuern“](#) auf Seite 57 wird beschrieben, wie der Benutzerzugriff auf Transaktionen gesteuert wird, die Db2-Befehle im CICS-Adressraum absetzen.

Wenn Benutzer einen Db2-Befehl über eine CICS-Transaktion absetzen, unterliegen sie auch der Db2-Sicherheitsprüfung, die überprüft, dass sie bei Db2 zum Absetzen dieses Befehls berechtigt sind. Bei dieser Sicherheitsprüfung werden die Berechtigungs-IDs (primär oder sekundär) verwendet, die die Transaktion von CICS übergeben hat. In „Für Db2 Berechtigungs-IDs für die CICS-Region und für CICS-Transaktionen bereitstellen“ auf Seite 58 erfahren Sie, wie diese Berechtigungs-IDs ausgewählt und für Db2 bereitgestellt werden. Für Transaktionen, die DFHD2CM1 zum Absetzen von Db2-Befehlen verwenden, werden die Berechtigungs-IDs durch das Attribut COMAUTHID oder COMAUTHTYPE der DB2CONN-Definition der CICS-Region festgelegt. Für andere Anwendungen, die Db2-Befehle absetzen, werden die Berechtigungs-IDs vom Attribut AUTHID oder AUTHTYPE für die Ressourcendefinition der CICS-Region für den von der Transaktion verwendeten Threadtyp (Pool-Thread oder Einstiegsthread) festgelegt. Diese Attribute steuern die Berechtigungs-ID oder den Typ der Berechtigungs-ID, die bzw. der von einer Transaktion, die diesen Typ von Thread verwendet, an Db2 übergeben wird.

Db2-Befehle unterliegen daher zwei Sicherheitsprüfungen, einer im CICS-Adressraum und einer im Db2-Adressraum. In [Abbildung 24 auf Seite 67](#) wird dieser Prozess veranschaulicht.

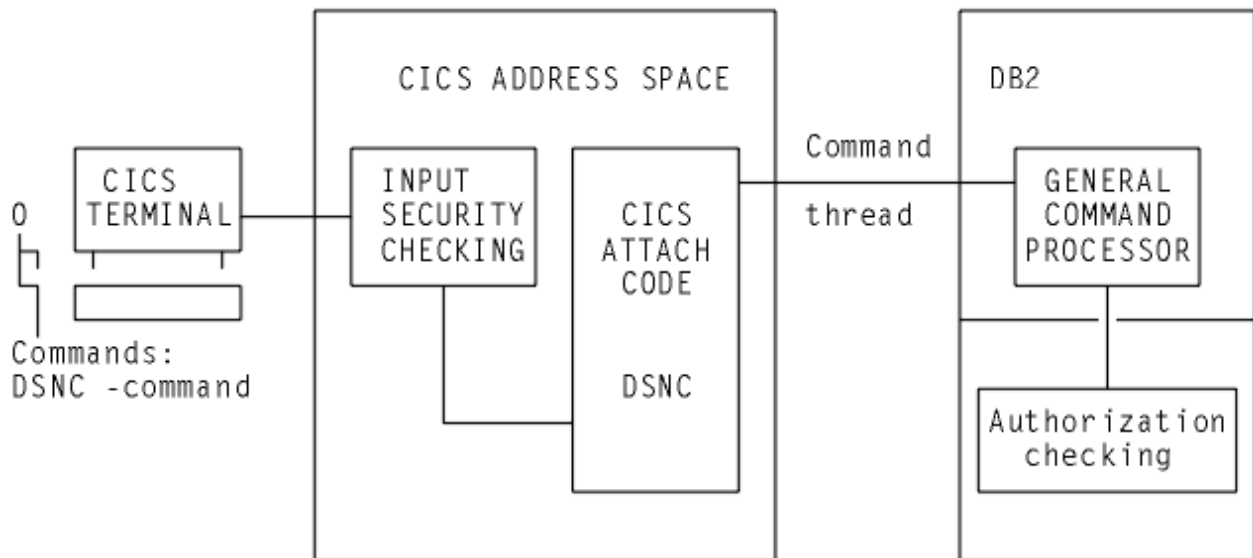


Abbildung 24. Sicherheitsmechanismen für Db2-Befehle

In den meisten Fällen, ist es nur für eine begrenzte Anzahl von Benutzern zulässig, Db2-Befehle auszuführen. Es kann eine geeignete Lösung sein, in der DB2CONN-Definition COMAUTHTYPE(USERID) anzugeben, das in die CICS-Benutzer-ID (mit 8 Byte) als Berechtigungs-ID in Db2 aufgelöst wird. Mit dieser Methode können Sie verschiedene Db2-Zugriffsrechte explizit an CICS-Benutzer-IDs erteilen. Sie können z. B. GRANT DISPLAY verwenden, um bestimmten CICS-Benutzer-IDs die Berechtigung zu erteilen, nur den Befehl -DIS zu verwenden.

Wenn Sie einen Benutzer für das Absetzen eines Db2-Befehls berechtigen möchten, verwenden Sie den Befehl GRANT, um Db2-Befehlsberechtigungen der Berechtigungs-ID zu erteilen, die die Transaktion von CICS übergeben hat. Informationen zur Vorgehensweise für die Erteilung und die Entziehung von Db2-Berechtigungen für Berechtigungs-IDs finden Sie in [Db2 schützen in der Produktdokumentation zu Db2 for z/OS](#).

Benutzerzugriff auf Pläne steuern

Bevor ein Benutzer in Db2 auf Pläne zugreifen kann, werden eine Reihe von Prüfungen ausgeführt. Diese Prüfungen starten im CICS-Adressraum, bevor die Anforderung zu weiteren Prüfungen an Db2 übergeben wird.

Informationen zu diesem Vorgang

Für Db2-Befehle erfolgt die erste Sicherheitsprüfung für den Zugriff von Benutzern auf Pläne im CICS-Adressraum, wenn CICS überprüft, ob der Benutzer berechtigt ist, auf die Transaktion zuzugreifen, die den Plan ausführt. Die zweite Sicherheitsprüfung findet im Db2-Adressraum statt, wenn Db2 prüft, ob die von

der Transaktion angegebene Berechtigungs-ID berechtigt ist, den Plan auszuführen. In [Abbildung 25](#) auf [Seite 68](#) wird dieser Prozess veranschaulicht.

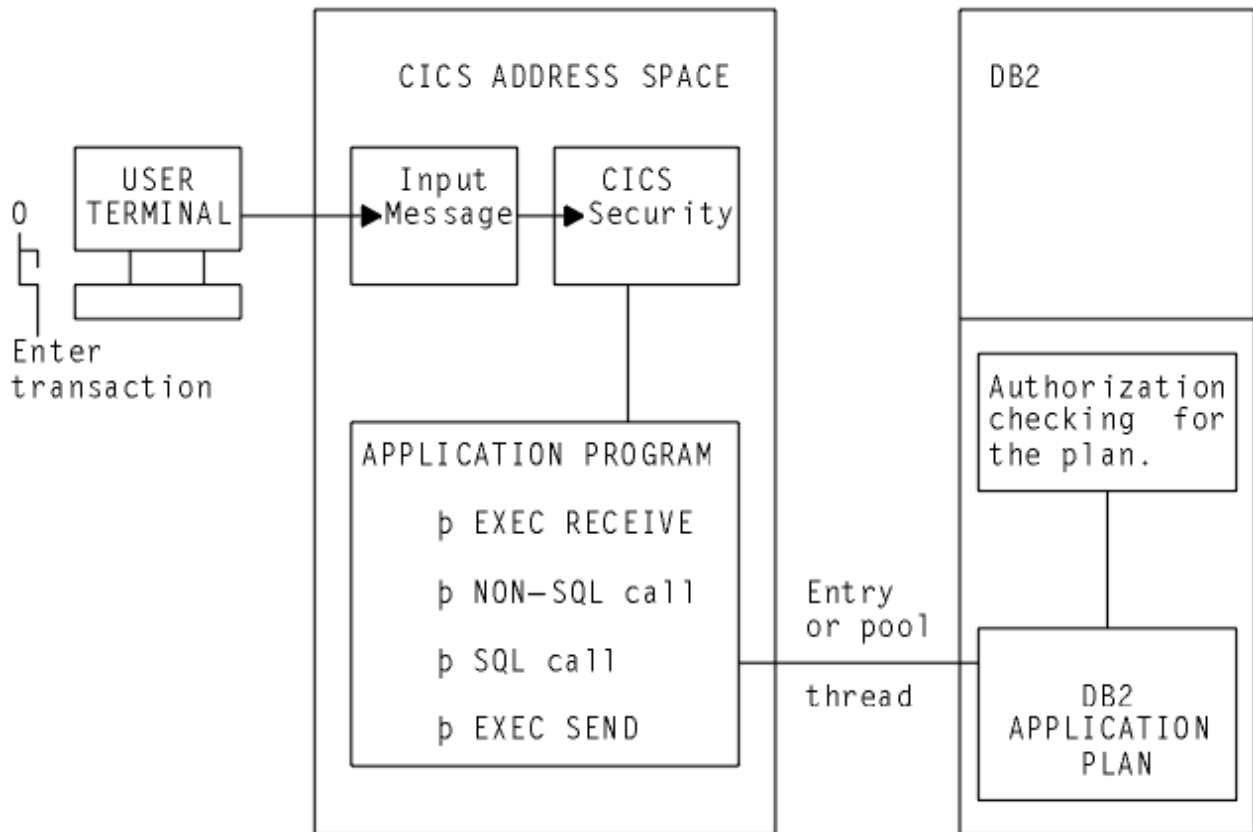


Abbildung 25. Sicherheitsmechanismen für die Ausführung eines Plans

Zum Berechtigen eines Benutzers für die Ausführung eines Plans verwenden Sie den Befehl GRANT, um der Berechtigungs-ID, die die Transaktion von CICS übergeben hat, Db2-Befehlsberechtigungen zu erteilen. Informationen zur Vorgehensweise für die Erteilung und die Entziehung von Db2-Berechtigungen für Berechtigungs-IDs finden Sie in [Db2 schützen in der Produktdokumentation zu Db2 for z/OS](#).

Wenn ein Plan dynamisches SQL enthält

Bei der Verwendung von statischem SQL muss der Binder des Plans die erforderlichen Berechtigungen für den Zugriff auf die Daten besitzen, und die von CICS an Db2 übergebene Berechtigungs-ID muss nur über die Berechtigungen verfügen, die zum Ausführen des Plans erforderlich sind.

Informationen zu diesem Vorgang

Wenn jedoch die Verwendung von dynamischem SQL für einen Plan erforderlich ist, muss die von CICS an Db2 übergebene Berechtigungs-ID die Berechtigungen aufweisen, die für den Zugriff auf alle beteiligten Db2-Ressourcen – sowohl Plan als auch Daten – erforderlich sind. Wenn Sie beispielsweise AUTHTYPE(USERID) angeben, müssen der CICS-Benutzer-ID Db2-Zugriffsrechte auf die Db2-Ressourcen erteilt werden, die in dynamisches SQL einbezogen sind. Wenn diese Benutzer-ID gleichzeitig eine TSO-Benutzer-ID ist, hat sie Zugriff auf die Db2-Ressourcen direkt von SPUFI, QMF und anderen Dienstprogrammen aus.

Wenn Sie nicht viel Zeit aufwenden wollen, um Db2-Zugriffsrechte zu erteilen, wenn eine Transaktion einen Plan ausführt, für den die Verwendung von dynamischem SQL erforderlich ist, können Sie die Verwendung einer der folgenden Methoden in Betracht ziehen, mit denen eine Berechtigungs-ID für Db2 bereitgestellt wird:

- Verwenden Sie die Option SIGN für das Attribut AUTHTYPE der DB2ENTRY-Definition für den Thread, der von der Transaktion verwendet wird. Dies führt dazu, dass die Transaktion die primäre Berechtigungs-ID aufweist, die Sie im Attribut SIGNID der DB2CONN-Definition für die CICS-Region angegeben

haben. (Diese Methode ist ungeeignet, wenn für die Sicherheitsprüfung im Db2-Adressraum RACF verwendet wird.)

- Verwenden Sie das Attribut AUTHID der DB2ENTRY-Definition für den Thread, der von der Transaktion verwendet wird, um eine standardmäßige Berechtigungs-ID anzugeben. Verwenden Sie dieselbe Berechtigungs-ID für alle Transaktionen, die auf dynamisches SQL zugreifen müssen. (Diese Methode ist ungeeignet, wenn für die Sicherheitsprüfung im Db2-Adressraum RACF verwendet wird.)
- Erstellen Sie eine RACF-Gruppe und verbinden Sie Ihre CICS-Benutzer mit dieser RACF-Gruppe. Verwenden Sie das Attribut GROUP der DB2ENTRY-Definition für den Thread, den die Transaktion verwendet, sodass die RACF-Gruppe eine der sekundären IDs ist, die an Db2 übergeben werden.

Sie können dann in jedem Fall Db2-Zugriffsrechte für Db2-Ressourcen, die in das gesamte dynamische SQL einbezogen sind, einer einzigen ID erteilen; dies kann die standardmäßige Berechtigungs-ID der DB2CONN-Definition oder das Attribut AUTHID oder auch der Name der RACF-Gruppe sein. In „Für Db2 Berechtigungs-IDs für die CICS-Region und für CICS-Transaktionen bereitstellen“ auf Seite 58 erfahren Sie, wie Sie Berechtigungs-IDs anhand all dieser Methoden bereitstellen.

Db2-Sicherheit auf mehreren Ebenen und auf Zeilenebene

In DB2 Version 8 wurde Unterstützung für die Verwendung mehrerer Sicherheitsebenen eingeführt. CICS bietet keine gesonderte Unterstützung für die Verwendung mehrerer Sicherheitsebenen; Sie können CICS jedoch in einer Umgebung mit mehreren Sicherheitsebenen verwenden, vorausgesetzt, Sie führen eine sorgfältige Konfiguration aus:

Weitere Informationen zur Verwendung der Sicherheit auf mehreren Ebenen finden Sie in den folgenden Veröffentlichungen:

- [Db2 schützen in der Produktdokumentation zu Db2 for z/OS](#)
- [z/OS Planning for Multilevel Security and the Common Criteria](#)
- [IBM Redbooks: Securing DB2 and Implementing MLS on z/OS](#)

Wenn die Sicherheit für mehrere Ebenen auf der Zeilenebene (Sicherheit auf Zeilenebene) für Daten in DB2 Version 8 oder höher implementiert wird, wird die RACF-Klasse SECLABEL aktiviert und es wird eine Gruppe von Sicherheitskennsätzen für Benutzer und für die Db2-Tabellenzeilen definiert. Es ist nicht erforderlich, dass die RACF-Optionen SETR MLS und MLACTIVE aktiv sind. Sie können die Db2-Sicherheit auf Zeilenebene verwenden, ohne dass sich dies auf den übrigen Teil des MVS-Systems auswirkt.

CICS kann auf Db2-Zeilen zugreifen, die auf diese Weise gesichert wurden. Für CICS müssen Sie sicherstellen, dass das RACF-Benutzerprofil für einen CICS-Benutzer, der Zugriff auf die Db2-Zeilen benötigt, in RACF so definiert ist, dass ein standardmäßiges SECLABEL-Element eingeschlossen ist. Details finden Sie in [z/OS Security Server RACF Security Administrator's Guide](#).

Wenn sich ein CICS-Benutzer bei einer CICS-Region anmeldet und im Systeminitialisierungsparameter SEC=YES angegeben ist, ordnet RACF das standardmäßige SECLABEL-Element dem RACF-Zugriffssteuerungsumgebungselement (ACEE) für den Benutzer zu. In der DB2ENTRY-Definition (oder der DB2CONN-Definition, falls der Pool verwendet wird) muss AUTHTYPE=USERID oder AUTHTYPE=GROUP angegeben sein, wodurch sichergestellt ist, dass das Zugriffssteuerungsumgebungselement für weitere Sicherheitsprüfungen an Db2 übergeben wird. Ein einzelner CICS-Benutzer kann daher nur ein einziges zugeordnetes SECLABEL-Element aufweisen.

Wenn bei Nichtterminaltransaktionen oder -programmen wie beispielsweise PLT-Programmen (PLT – Programmliistentabelle) der Systeminitialisierungsparameter PLTPUSER nicht angegeben und der Systeminitialisierungsparameter PLTPSEC=NONE angegeben ist, werden PLT-Programme unter der Benutzer-ID für die CICS-Region ausgeführt. In diesem Fall müssen Sie die Benutzer-ID für die CICS-Region mit einem standardmäßigen SECLABEL-Element definieren. Wenn Sie für eine Transaktion unterschiedliche SECLABEL-Elemente definieren müssen, müssten Sie jede Transaktion jeweils in einer separaten CICS-Region ausführen, die alle unterschiedliche Benutzer-IDs für die CICS-Region und unterschiedliche zugeordnete SECLABEL-Elemente aufweisen.

Kapitel 5. Aspekte des Anwendungsdesigns und der Anwendungsentwicklung für CICS Db2

Während des Designprozesses

Während des Designprozesses für die Anwendung können sich von Ihnen getroffene Entscheidungen auf Anwendungen auswirken, die aktuell in der Entwicklung sind sowie auf geplante zukünftige Anwendungen. Zu den wichtigsten Aspekten bezüglich des Designs gehören Folgende:

- Die Beziehung zwischen CICS-Anwendungen und Db2-Plänen sowie -Paketen. Damit Sie für die Leistung und Verwaltung Ihres Systems den größten Nutzen erzielen können, muss die Beziehung zwischen Db2-Plänen, Transaktionen und Anwendungsprogrammen definiert werden, wenn Sie die Anwendungen entwerfen.
- Die Sperrstrategie. Sperren sind durch Optionen betroffen, die Sie beim Erstellen von Tabellen in Db2 auswählen, durch das Design Ihrer Anwendungsprogramme und durch Optionen, die Sie beim Binden von Plänen auswählen; das heißt, Sie müssen Sperren während des gesamten Entwicklungsprozesses berücksichtigen.
- Die Sicherheitsaspekte sowohl des CICS-Db2-Testsystems als auch des CICS-Db2-Produktionssystems müssen berücksichtigt werden.
- Verwendung von Befehlen und Programmiertechniken. Diese können die Leistung Ihrer Anwendung verbessern und potenzielle Probleme vermeiden. Die Verwendung von qualifiziertem und nicht qualifiziertem SQL kann sich auch auf andere Aspekte der CICS-Db2-Umgebung auswirken. Damit Java™-Anwendungen in der JVM-Serverumgebung ausgeführt werden können und um Leistungsvorteile aus der Umgebung für offene Transaktionen (OTE) zu erzielen, müssen Anwendungsprogramme threadsicher sein. Weitere Informationen hierzu finden Sie in [„SQL, Threadsicherheit und andere Erwägungen bei der Programmierung für CICS-Db2-Anwendungen“](#) auf Seite 89.
- Java-Programme in der CICS-Db2-Umgebung. Informationen zu Aspekten der Unterstützung und Programmierung finden Sie in [Kapitel 6, „Verwendung von JDBC und SQLJ für den Zugriff auf Db2-Daten aus Java-Programmen“](#), auf Seite 101.
- Zusätzliche Schritte, die vorzunehmen sind, wenn die Anwendungsentwicklung abgeschlossen ist und die Anwendung in die Produktion versetzt wird.
- Für eine verbesserte Leistung definierte Verbindungen. Definieren Sie Ihre CICS-Db2-Verbindungen, um die Anwendungsleistung zu verbessern. Wenn Sie beispielsweise geschützte Einstiegsthreads für die Anwendungsprogramme verwenden, um auf Db2 zuzugreifen, verbessert dies die Leistung von intensiv genutzten Programmen.

Faktoren, die vom Design betroffen sind

Das von Ihnen implementierte Design kann sich auf folgende Elemente auswirken:

- Leistung
- Gemeinsamer Zugriff
- Operation
- Sicherheit
- Abrechnung
- Entwicklungsumgebung

Faktoren, die sich auf die Anwendungsleistung auswirken

Eine gut entworfene CICS-Db2-Anwendung sollte ordnungsgemäß funktionieren, wenn sie erstmalig in einer Produktionsumgebung bereitgestellt wird. Einige Faktoren können sich jedoch auf die Leistung gut entworfener Anwendungen auswirken:

- Erhöhte Transaktionsrate
- Fortlaufende Entwicklung vorhandener Anwendungen
- Eine höhere Anzahl von Personen, die an der Entwicklung von CICS-Db2-Anwendungen beteiligt sind
- Vorhandene Tabellen, die in neuen Anwendungen verwendet werden
- Integration von Anwendungen

Aufgrund dieser Faktoren ist es wichtig, eine konsistente Gruppe von Standards für die Verwendung von Db2 in der CICS-Umgebung zu entwickeln.

Anwendungen mit Daten, die in VSAM und DL/I gespeichert sind

Für Anwendungen mit Daten, die in VSAM (VSAM – Virtual Storage Access Methode) oder DL/I (DL/I – Data Language/Interface) gespeichert sind, gibt es Unterschiede in der Art und Weise, in der einige Prozesse im Vergleich zu CICS-Db2-Anwendungen funktionieren. Folgende Unterschiede sind zu berücksichtigen:

- Sperrmechanismus
- Sicherheit
- Wiederherstellung und Neustart
- Bindeprozess (BIND)
- Betriebsprozesse
- Leistung
- Programmiertechniken

Einer der Hauptunterschiede zwischen dem Design von Stapelverarbeitungsprogrammen und Onlineprogrammen besteht darin, dass Onlinesysteme für einen hohen Grad an gemeinsamen Zugriffen entworfen werden müssen. Gleichzeitig sollte die Datenintegrität nicht beeinträchtigt werden. Darüber hinaus haben die meisten Onlinesysteme Designkriterien, die sich auf die Leistung beziehen.

Zusätzliche Informationen

In den folgenden zusätzlichen Quellen finden Sie Informationen zum Anwendungsdesign:

- [Programmierung für Db2 for z/OS in der Produktdokumentation zu Db2 for z/OS](#)
- [Anwendungen entwerfen](#)

Beziehung zwischen CICS-Anwendungen und Db2-Plänen und -Paketen gestalten

Wenn CICS-Anwendungen Db2-Daten verwenden, muss die Anwendungsarchitektur Designaspekte in Betracht ziehen, die die CICS-Db2-Anschlussfunktion betreffen. Einer der wichtigsten zu berücksichtigenden Aspekte ist die Beziehung zwischen Transaktions-IDs, Db2-Plänen und -Paketen sowie Programmmodulen.

Wenn in einem der Programmmodule SQL-Aufrufe verwendet werden, muss ein entsprechender Db2- oder ein entsprechendes Db2-Paket verfügbar sein. Der für eine Transaktions-ID zu verwendende Plan ist in der DB2CONN- oder DB2ENTRY-Definition für den Thread angegeben, den die Transaktion für den Zugriff auf Db2 verwendet. Der Plan kann explizit genannt werden oder es kann eine Exitroutine für den Plan benannt werden, die den Plannamen auswählt. Der Plan muss die Datenbankankforderungsmodule aus sämtlichen Modulen enthalten, die unter dieser Transaktions-ID ausgeführt werden könnten. Die Datenbankankforderungsmodule können direkt in den Plan gebunden werden oder sie können als Pakete gebunden und in einer Paketliste im Plan benannt werden.

Zur Steuerung der Merkmale des Plans und der Threads der CICS-Db2-Anschlussfunktion muss die Beziehung zwischen Transaktions-IDs, Db2-Plänen und den Programmmodulen während des Designs definiert werden. Einige der Merkmale der Threads, des EDM-Pools (EDM – Environmental Description Manager) und der Pläne, die vom Design abhängen, sind im Folgenden aufgelistet:

- Plangrößen
- Anzahl der verschiedenen Pläne, die gleichzeitig verwendet werden
- Anzahl der Threads, die gleichzeitig verwendet werden
- Möglichkeit, einen Thread wiederzuverwenden
- Größe des EDM-Pools
- E/A-Aktivität im EDM-Pool
- Gesamt-Paging-Rate im System
- Berechtigungsgranularität
- Verwendung der Db2-Pakete

Es gibt verschiedene Designtechniken, mit denen CICS-Transaktionen mit Db2-Plänen verbunden werden können; es wird empfohlen, Pläne auf der Basis von Paketen zu verwenden.

Beispielanwendung

Zum Erläutern der Konsequenzen verschiedener für das Anwendungsdesign verwendeter Verfahren kann ein einfaches Beispiel verwendet werden.

Abbildung 26 auf Seite 74 zeigt, wie CICS-Zuordnungen (CICS MAP) und Transaktions-IDs miteinander korrelieren und wie die Transaktionen funktionieren; Überlegungen zu Db2 bleiben dabei unberücksichtigt.

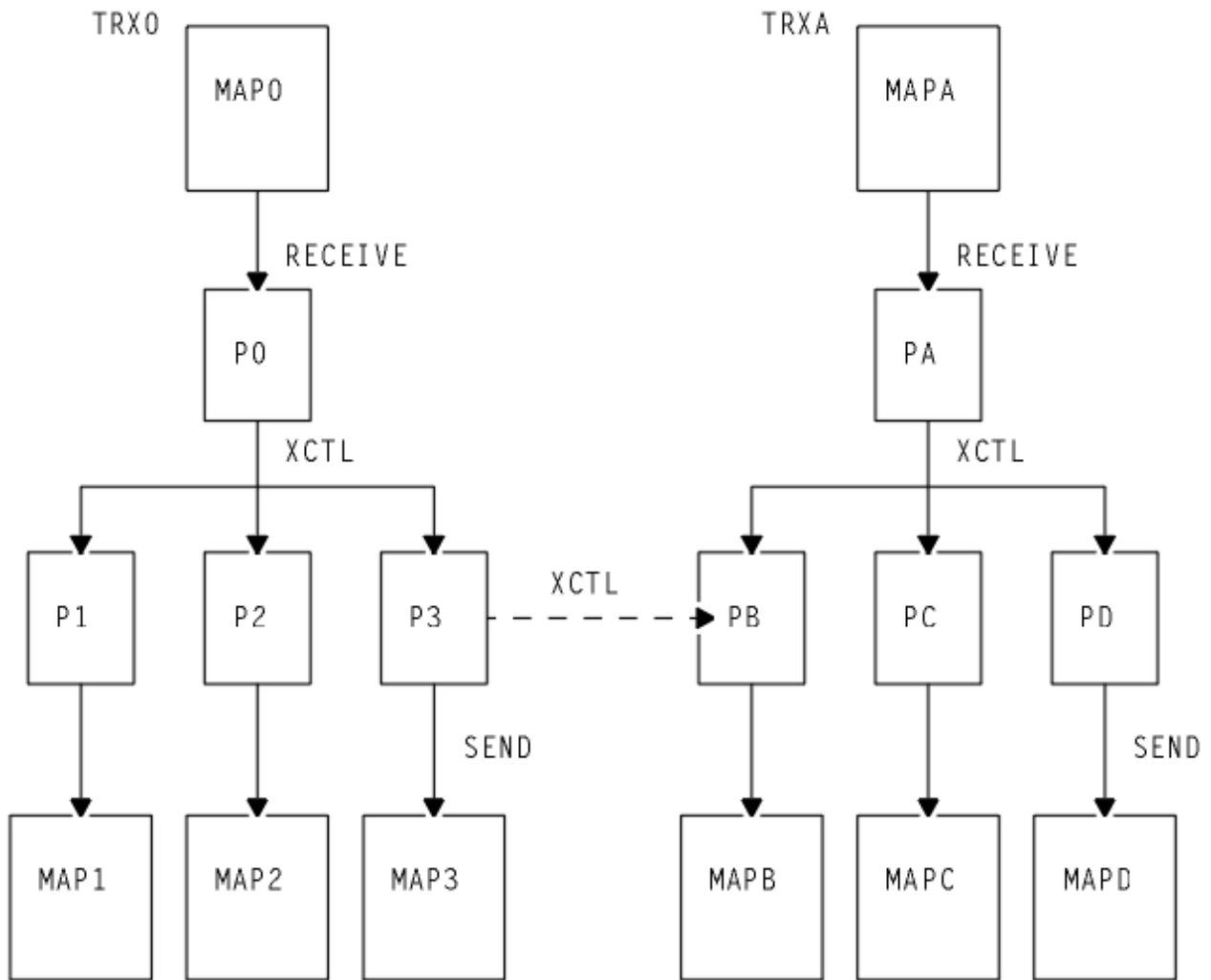


Abbildung 26. Beispiel für ein typisches Anwendungsdesign

In diesem Beispiel gilt Folgendes:

- Die Transaktions-ID TRX0 wird im Befehl EXEC CICS RETURN TRANSID(TRX0) angegeben, wenn ein Programm (nicht dargestellt) die Steuerung nach Anzeige von MAP0 zurückgibt.
- Die nächste Transaktion verwendet dann die Transaktions-ID TRX0 unabhängig dessen, welche Aktion der Terminalbenutzer entschieden hat auszuführen.
- Programm P0 ist das Startprogramm für Transaktion TRX0.
- Es wird vorausgesetzt, dass alle dargestellten Programme SQL-Aufrufe absetzen.
- Abhängig von der vom Terminalbenutzer ausgewählten Option führt Programm P0 den CICS-Befehl zur Übergabe der Steuerung (XCTL) an eines der Programme P1, P2 oder P3 aus.
- Nach dem Absetzen einiger SQL-Aufrufe zeigen diese Programme eine der folgenden Zuordnungen an: MAP1, MAP2 oder MAP3.
- Das auf der rechten Seite der Abbildung dargestellte Beispiel funktioniert in derselben Weise.
- In einigen Situationen überträgt Programm P3 die Steuerung an Programm PB.

Db2-Pakete verwenden

Die Verwendung von Paketen ist die beste Möglichkeit sicherzustellen, dass sich die Beziehung zwischen CICS-Transaktionen und Db2-Plänen einfach verwalten lässt.

Bevor Pakete zur Verfügung standen, mussten in früheren Releases von Db2 die Datenbankanforderungsmodule aller Programme, die unter einer bestimmten CICS-Transaktion ausgeführt werden konnten, di-

rekt in einen einzigen Plan gebunden werden. Wurde ein Datenbankanforderungsmodul in einem Plan geändert (weil sich die SQL-Aufrufe für dieses Programm geändert hatten), mussten alle Datenbankanforderungsmodule in diesem Plan erneut gebunden werden. Das Binden eines großen Plans kann sehr lange dauern und während der Operation ist die gesamte Transaktion für die Verarbeitung nicht verfügbar. Es kann bereits sehr schwierig sein, eine stark verwendete Anwendung in den Wartemodus zu versetzen; den Plan jedoch für einen größeren Zeitraum außer Betrieb zu nehmen, während er erneut gebunden wird, ist in der Regel nicht akzeptabel.

Dynamische Planexits waren eine Zwischenlösung, mit der dieses Problem behoben werden sollte. Der dynamische Planexit ist ein Exitprogramm, das Sie in der DB2CONN- oder DB2ENTRY-Definition angeben, anstatt einen Plannamen anzugeben. Sie erstellen für Ihre CICS-Anwendungen viele kleine Pläne, von denen jeder die Datenbankanforderungsmodule einiger Programme enthält, und das Exitprogramm wählt beim Start der einzelnen Arbeitseinheiten den richtigen Plan aus. Sie können den dynamischen Planexit auch verwenden, um innerhalb einer Transaktion zwischen Plänen zu wechseln. Jeder kleine Plan muss jedoch erneut gebunden und außer Betrieb gesetzt werden, sobald sich eine SQL-Anweisung in einem der Programme ändert, die diese Anweisung verwenden. Darüber hinaus erfordert die Verwendung des dynamischen Planwechsels, dass der Wechsel zwischen Plänen durch einen impliziten oder expliziten **SYNC-POINT** von CICS ermöglicht wird.

Da nun Pakete in Db2 zur Verfügung stehen, bieten sie die beste Möglichkeit für die Verwaltung Ihrer Pläne. Mit Paketen können Sie die Programmeinheiten in kleinere Teile aufteilen, die Sie einzeln erneut binden können, ohne dass sich dies auf den gesamten Plan oder auch nur auf die aktuellen Benutzer des jeweiligen erneut zu bindenden Pakets auswirkt.

Da die Aktualisierung eines Plans mit Paketen einfacher ist, können Sie viel größere Anwendungen erstellen, ohne dass Sie Transaktionen, Programme oder Pläne wechseln müssen, um die Leistung oder Verfügbarkeit von Db2 aufrecht zu erhalten. Dies bedeutet auch, dass Sie nicht so viele RDO-Definitionen verwalten müssen. Sie vermeiden so auch Situationen, in denen Sie andernfalls dynamisches SQL verwenden würden, um Programmflexibilität zu erhalten. In einem Plan können Sie sogar Pakete angeben, die noch nicht vorhanden sind, oder Paketsammlungen angeben, denen Sie Pakete hinzufügen können. Dies bedeutet, dass Datenbankanforderungsmodule aus neuen Programmen hinzugefügt werden können, ohne dass der vorhandene Plan in geringster Weise gestört wird.

Die Option für das Qualifikationsmerkmal für Pakete und Pläne, mit der Sie auf unterschiedliche Tabellensätze verweisen können, kann Ihnen eine größere Flexibilität bieten, wie Sie Planwechsel vermeiden können.

Zusammenfassend bieten Pakete Folgendes:

- Die Zeit, in der der Plan nicht betriebsbereit ist, und die Prozessorzeit sowie die Anzahl der für das Binden erforderlichen Katalogtabellensperren werden für Programme reduziert, die mithilfe von **START**, **LINK** oder **RETURN TRANSID** logisch verknüpft sind; außerdem werden Datenbankanforderungsmodule gebunden, wodurch die Anzahl der DB2ENTRY-Definitionen verringert wird.
- Die Anzahl der CICS-Starts (START) oder -Exits wird verringert.
- Das Klonen von CICS- und DB2ENTRY-Definitionen kann vermieden werden.
- Es besteht die Möglichkeit, einen Plan mit null Paketen zu binden, um Elemente später in den Plan einzubeziehen.
- Sie haben die Möglichkeit, die Zusammenstellung zur Ausführungszeit anzugeben, indem Sie **CURRENT PACKAGESET=variable** verwenden; dies ist ein leistungsfähiges Merkmal, wenn es in Verbindung mit **QUALIFIER** verwendet wird.
- Der Parameter **QUALIFIER** wird bereitgestellt, der Flexibilität für Folgendes bietet:
 - Mithilfe von nicht qualifiziertem SQL können Sie auf unterschiedliche Tabellensätze verweisen.
 - Es sind weniger Synonyme und Aliasnamen erforderlich.
 - Dynamisches SQL ist in geringerem Maße erforderlich.

Es gibt weitere Vorteile, die die Verwendung von Paketen in einer CICS-Umgebung bieten kann:

- Es ermöglicht Ihnen, weniger Pläne zu verwenden.

- Es ermöglicht Ihnen, weniger benötigte Transaktionen in einen einzigen Plan zu binden.
- Es erhöht das Potenzial für die Threadwiederverwendung.

Sie können Ihre Anwendung auch optimieren, indem Sie Paketlisten verwenden, in denen die Einträge zum Verringern der Suchzeit geordnet sind, oder indem Sie die Anzahl der Paketlisteneinträge so gering wie möglich halten.

Db2 bietet auch eine Abrechnung auf Paketebene. Weitere Informationen zu Paketen finden Sie in den Beschreibungen, wie das Binden und die Erstellung eines Anwendungsprogramms geplant werden können, die in [Programmierung für Db2 for z/OS in der Produktdokumentation zu Db2 for z/OS und der IBM Redbooks-Veröffentlichung Db2 9 for z/OS: Packages Revisited \(Neues zu Paketen\)](#) enthalten sind.

In Pakete konvertieren

Sie können Transaktionen konvertieren, die dynamische Planexits oder Verfahren für dynamische Planwechsel verwenden, sodass stattdessen Pakete verwendet werden.

Informationen zu diesem Vorgang

Eine Transaktion, die aktuell einen dynamischen Planexit oder Verfahren für dynamische Planwechsel verwendet, kann wie folgt für die Verwendung von Paketen konvertiert werden:

- Binden Sie alle Datenbankanforderungsmodule, die im der Transaktion zugeordneten Plan enthalten sind, in Pakete, die sich in einer einzigen Sammlung befinden.
- Binden Sie mithilfe eines PKLIST-Elements, das für diese Sammlung einen einzelnen Platzhaltereintrag enthält, einen neuen Plan.
- Ändern Sie den DB2ENTRY-Eintrag für diese Transaktion, um den neuen Plan zu verwenden. Geschützte Threads können jetzt für diese Transaktion verwendet werden, um die Threadwiederverwendung zu optimieren.

Sie haben die Möglichkeit, einen einzigen Plan für die gesamte Anwendung oder einen Plan pro Transaktion zu verwenden. In den folgenden Abschnitten finden Sie ausführlichere Anweisungen, wie Ihre Transaktionen auf der Basis dieser Auswahl konvertiert werden.

Ein ähnlicher Ansatz kann für die Konvertierung aller CICS-Anwendungen verwendet werden, unabhängig davon, ob sie einen dynamischen Planexit verwenden oder nicht.

Beachten Sie, dass intensiv genutzte Pakete mit RELEASE(DEALLOCATE) gebunden werden können; während weniger genutzte Pakete mit RELEASE(COMMIT) gebunden werden. Dies führt dazu, dass die intensiv genutzten Pakete so lange im Paketverzeichnis des Plans gespeichert werden, bis die Zuordnung des Plans aufgehoben wird, während die weniger genutzten Pakete aus dem Verzeichnis entfernt werden und der Speicherbereich im EDM-Pool freigegeben wird. Der Nachteil dieser Methode besteht darin, dass Sie, wenn Sie RELEASE(DEALLOCATE) verwenden und dann ein intensiv genutztes Paket erneut binden müssen, eingreifen müssen, um die Aufhebung der Zuordnung des Pakets zu erzwingen, wenn es einem Thread mit langer Laufzeit zugeordnet ist. Beachten Sie, dass hinsichtlich der aufzuwendenden Zeit das erneute Binden eines Pakets weniger kostenintensiv ist als das erneute Binden eines Plans.

Pro Anwendung einen Plan verwenden

Die Verwendung eines einzigen Plans für die Anwendung bietet die größte Flexibilität bei der Definition der DB2ENTRY- und DB2TRAN-Objekte für die Anwendung, da die beteiligten Transaktionen gruppiert werden können, um geschützte Threads besser zu nutzen und die Wiederverwendung von Threads zu optimieren. Folgen Sie dem folgenden Verfahren, um eine Konvertierung in diese Umgebung auszuführen.

Vorgehensweise

1. Binden Sie alle für die Transaktionen in der Anwendung erforderlichen Datenbankanforderungsmodule in Pakete und verwenden Sie dazu eine einzige Sammlung wie beispielsweise COLLAPP1.
2. Binden Sie einen neuen Plan mit dem Namen PLANAPP1, der eine Paketliste aufweist, die als einzigen Eintrag COLLAPP1.* enthält.

```
BIND PLAN (PLANAPP1) ..... PKLIST (COLLAPP1.*) ..
```

3. Ersetzen Sie im DB2ENTRY-Objekt den Programmnamen des dynamischen Planexits durch den Namen dieses neuen Plans.
Ersetzen Sie beispielsweise

```
PLANEXITNAME=DSNCUEXT
```

durch

```
PLAN=PLANAPP1
```

Pro Transaktion einen Plan verwenden

Diese Lösung wurde vor DB2 Version 3 empfohlen, da die Abrechnung auf der Ebene einzelner Pakete nicht möglich war; die Abrechnung musste daher nach Plannamen vorgenommen werden. Aktuelle Releases von Db2 bieten jedoch die Abrechnung auf Paketebene, wodurch Pläne aus einer großen Anzahl von Paketen bestehen können und dennoch die für die Abrechnung erforderliche Granularität bereitgestellt wird.

Vorgehensweise

1. Binden Sie die für Gruppen von Transaktionen oder für alle Transaktionen erforderlichen Datenbankforderungsmodule in Pakete.
Die zu verwendenden Sammlungen können auf den zu konvertierenden Transaktionen wie beispielsweise TRAN1 oder auf der Anwendung basieren. Der zuletzt genannte Ansatz ist vorzuziehen, da das Erstellen und Verwalten von Sammlungen auf Transaktionsbasis einen größeren Verwaltungsaufwand erfordert; insbesondere, wenn viele gemeinsame Routinen vorhanden sind.
2. Binden Sie für jede Transaktion einen neuen Plan mit einer Paketliste, die auf einen einzelnen Paketlisteintrag mit einem Platzhalterzeichen verweist, das sich nach dem gewählten Ansatz richtet.
 - Verwenden Sie TRAN1.*, wenn die Sammlungen auf Transaktionen basieren:

```
BIND PLAN (TRAN1) .... PKLIST (TRAN1.*) ...
```

- Verwenden Sie COLLAPP1.*, wenn für alle Transaktionen eine einzelne Sammlung eingerichtet wurde:

```
BIND PLAN (TRAN1) .... PKLIST (COLLAPP1.*) ...
```

3. Ändern Sie die DB2ENTRY-Definitionen, um den dynamischen Planexit durch die Plannamen zu ersetzen, die den Transaktionen zugeordnet sind.

Wenn Sie Pakete haben und weiterhin Verfahren für den dynamischen Planwechsel verwenden möchten

Wenn für Benutzer des dynamischen Planwechsels der Wert eines änderbaren Sonderregisters (z. B., das Register CURRENT PACKAGESET) während der Verarbeitung geändert wird und nicht in den Anfangszustand zurückgesetzt wird, bevor der SYNCPOINT erfasst wird, tritt Folgendes auf:

- Der Thread wird von der CICS-Db2-Anschlussfunktion nicht freigegeben.
- Die Threadwiederverwendung findet nicht statt, weil die Task diesen Thread weiterhin mit demselben zugeordneten Plan verwendet.
- Der dynamische Planwechsel findet nicht statt, da derselbe Thread und derselbe Plan verwendet werden; das heißt, der Exit für den dynamischen Planwechsel wurde in der ersten SQL-Anweisung, die nach dem SYNCPOINT ausgegeben wurde, nicht erfasst.

Wenn Sie den dynamischen Planwechsel verwenden möchten, müssen Sie daher sorgfältig sicherstellen, dass alle änderbaren Sonderregister auf ihre Anfangswerte zurückgesetzt sind, bevor der SYNCPOINT erfasst wird. In SQL: Sprache von Db2 in der Produktdokumentation zu Db2 for z/OS sind die änderbaren Sonderregister aufgelistet.

Für alle Transaktionen einen großen Plan verwenden

Eine einfache Möglichkeit, das Wechseln von Plänen während einer Transaktion zu vermeiden, besteht darin, für alle CICS-Transaktionen, die SQL-Aufrufe absetzen, einen einzigen großen Plan zu verwenden.

Zu dem Beispiel in Abbildung 26 auf Seite 74:

- Es gibt einen Plan, PLAN0, für den die Datenbankanforderungsmodule aus P0, P1, P2, P3, PA, PB, PC und PD verwendet werden.
- Definieren Sie in CICS ein einziges DB2ENTRY-Objekt, indem Sie PLAN0 plus ein DB2TRAN-Objekt pro erforderliche Transaktions-ID angeben (es sei denn, es kann eine Transaktions-ID mit Platzhalterzeichen angegeben werden). Ein einziges DB2ENTRY-Objekt bietet bei Verwendung von geschützten Threads die beste Gesamtnutzung von Threads.

Vorteile

- Es gibt keine Einschränkungen hinsichtlich dessen, welche Programmmodule unter einer bestimmten Transaktions-ID ausgeführt werden können. Es ist z. B. möglich, dass Programm P3 die Steuerung an Programm B übertragen kann. Dies erfordert keine Änderung des Plans oder der Definition in CICS.
- Jedes Datenbankanforderungsmodul ist nur in einem einzigen Plan (PLAN0) vorhanden.
- Die Threadwiederverwendung ist einfach anzufordern. Alle Transaktionen können dasselbe DB2ENTRY-Objekt verwenden.

Nachteile

Es gibt bestimmte Nachteile, wenn ein einzelner großer Plan für alle Transaktion verwendet werden soll.

- Bei jeder Db2-Programmänderung muss der gesamte Plan neu gebunden werden.
- Das Binden (BIND) kann für große Pläne zeitintensiv sein.
- Der BIND-Prozess kann nur stattfinden, während der Plan nicht verwendet wird. Es ist anzunehmen, dass der Plan in einem Produktionssystem aufgrund der normalen Aktivität die überwiegende Zeit in Verwendung ist. In einer Testumgebung ist die Transaktionsrate in der Regel niedrig, aber Programmierer können Debugging-Tools verwenden, durch die die Antwortzeiten mit dialogorientierten Programmen länger werden. Dadurch bleiben Thread und Plan effektiv belegt.
- Eine von Db2 aufgerufene REBIND-Operation lässt (aufgrund der Planinaktivierung) nicht zu, dass Db2-Transaktionen diesen Plan ausführen.
- In Nachrichten und Statistikdaten, die den Plannamen anzeigen, ist kein realer Informationswert vorhanden, da es nur einen einzigen Plan gibt.
- EDMPOOL muss groß genug sein, um Datenbankbeschreibungen (DBD – Database Description), Gerüstcursortabellen (SKCT – Skeleton Cursor Table) und Cursortabellen (CT – Cursor Table) handhaben zu können und der Pool muss einen gewissen Grad an Fragmentierung zulassen. Beachten Sie, dass die Funktion für die Plansegmentierung es Db2 nur ermöglicht, Teile der auszuführenden Anwendungspläne in Cursortabellen zu laden. Dennoch werden die Header- und Verzeichnisteile eines Anwendungsplans in ihrer Gesamtheit in die Gerüstcursortabelle geladen (falls nicht bereits geladen) und anschließend aus der Gerüstcursortabelle in die Cursortabellen kopiert. Dies wird zum Zeitpunkt der Threaderstellung ausgeführt.

Da die Verzeichnisgröße des Anwendungsplans direkt von der Anzahl der Segmente im Plan abhängt, hat die Verwendung eines großen Plans Einfluss auf die Größe von EDMPOOL und die Anzahl der E/A-Operationen, die für die Steuerung von EDMPOOL erforderlich sind.

Viele kleine Pläne verwenden

Durch Verwenden vieler kleiner Pläne werden viele Transaktions-ID benötigt, von denen jede einen entsprechenden Plan hat. Dieses Verfahren reduziert die Plangrößen und das Überlappen von Plänen.

Es wird empfohlen, statt vieler kleiner Pläne Pakete zu verwenden.

Die Verwendung vieler kleiner Pläne beinhaltet, dass der Programmablauf einem schmalen Pfad mit begrenzten Möglichkeiten zur Verzweigung folgt oder dass häufig Planwechsel auftreten.

In dem Beispiel in [Abbildung 26 auf Seite 74](#) könnte der Wechsel zwischen Programm P0 und den Programmen auf der nächstniedrigeren Ebene oder zwischen Programm PA und den Programmen auf der nächstniedrigeren Ebene stattfinden.

- PLAN1 für (TRX0), unter Verwendung der Datenbankanforderungsmodule aus den Programmen P0, P1, P2 und P3.
- PLANA für (TRXA), unter Verwendung der Datenbankanforderungsmodule aus den Programmen PA, PB, PC und PD.

Programm P3 kann jedoch die Steuerung (mit dem Befehl XCTL) an Programm PB übertragen. Es muss dann ein Verfahren zum Planwechsel verwendet werden.

Es gibt eine besondere Variante der Verwendung von kleinen Plänen. In einigen Anwendungen kann es praktisch sein, dass es Aufgabe des Terminalbenutzers ist, die Transaktions-ID für die nächste Transaktion anzugeben. Dies ist typisch für schreibgeschützte Anwendungen, bei denen der Benutzer zwischen vielen verschiedenen Informationsanzeigen wählen kann, indem er eine systematisch erstellte Transaktions-ID mit ein bis vier Zeichen eingibt. Der Vorteil für den Benutzer ist dabei die Möglichkeit, beliebig zwischen Anzeigen hin- und herzuwechseln, ohne eine Hierarchie von Untermenüs zu durchlaufen.

Wenn jeder Transaktions-ID ein Db2-Plan zugeordnet wird, führt dies schließlich dazu, dass die Anwendung über viele kleine Pläne verfügt.

Vorteile

- Die Verwaltung des Plans ist relativ einfach, da die Pläne nur geringfügig überlappen.
- Hoher Informationswert in Nachrichten, Statistikdaten usw., die den Plannamen anzeigen.

Anmerkung: Pakete bieten diese Vorteile ebenfalls.

Nachteile

- Es tritt häufig ein Planwechsel auf, es sei denn, der Anwendungsablauf folgt einem schmalen Pfad.
- Es ist schwierig, geschützte Threads zu verwenden, da die Transaktionen über mehrere Gruppen von Transaktions-IDs, Plänen und Threads verteilt sind.
- Die Ressourcenauslastung kann wegen der Planwechsel und der geringen Wiederverwendung von Threads hoch sein.

Anmerkung: Mit Paketen werden diese Nachteile vermieden.

Pläne auf der Grundlage der Transaktionsgruppierung verwenden

Die Gruppierung von Transaktionen kann eine Reihe unabhängiger Pläne mittlerer Größe erzeugen, bei denen, falls erforderlich, ein Planwechsel auftreten kann.

Es ist häufig möglich, solche Gruppen von Programmen zu definieren, bei denen die Programme innerhalb einer Gruppe in enger Beziehung zueinander stehen. Dies bedeutet, dass sie häufig in derselben Transaktion ausgeführt werden oder dass sie in verschiedenen Transaktionen nacheinander ausgeführt werden. Für jede Gruppe sollte dann ein eigener Plan verwendet werden.

In dem Beispiel in [Abbildung 26 auf Seite 74](#) könnten zwei Pläne erstellt werden:

- PLAN1 für (TRX0), unter Verwendung der Datenbankanforderungsmodule aus den Programmen P0, P1, P2 und P3.
- PLANA für (TRXA), unter Verwendung der Datenbankanforderungsmodule aus den Programmen PA, PB, PC und PD.

Programm P3 kann jedoch die Steuerung an Programm PB übertragen. Es kann dann ein Verfahren für den Planwechsel verwendet werden. Es wird empfohlen, dass Planwechsel die Ausnahme und nicht der Normalfall sind, hauptsächlich wegen des zusätzlichen Prozessoraufwands.

In diesem Fall stimmt das Ergebnis des Verfahrens zur Transaktionsgruppierung mit dem Ergebnis des Verfahrens überein, bei dem viele kleine Pläne verwendet werden. Dies liegt an der Einfachheit des verwendeten Beispiels. Normalerweise erzeugt das Verfahren zur Transaktionsgruppierung einen größeren Plan.

Vorteile

- Die Plangröße und die Anzahl der verschiedenen Pläne können vom Benutzer gesteuert werden.
- Die Wiederverwendung von Threads ist wahrscheinlich; sie hängt von der Transaktionsrate innerhalb der Gruppe ab.

Nachteile

- Es kann eine Planüberlappung auftreten.
- Die optimale Gruppierung kann sich im Zeitverlauf ändern.
- Möglicherweise werden Planwechsel erforderlich.
- Ein Planwechsel wird im Anwendungscode ausgeführt.

Dynamische Planexits

Aufgrund der Vorteile von Paketen wird empfohlen, statt dynamischen Planexits Pakete zu verwenden. Dynamische Planexits stellten eine Zwischenlösung dar, die zur Behebung von Problemen in der CICS-Db2-Umgebung entworfen wurden, bevor Pakete in Db2 verfügbar wurden.

Sie können CICS-Anwendungen für viele kleine Pläne entwerfen und den jeweiligen Plan zur Ausführungszeit dynamisch auswählen. Ein kleiner Plan ist nicht dasselbe wie ein Paket, das eine strikte Eins-zu-eins-Entsprechung zu einem Datenbankanforderungsmodul (DBRM – Database Request Module) aufweist.

In der Regel wird ein dynamischer Planexit ausgelöst, um zu ermitteln, welcher Plan beim Start der ersten Arbeitseinheit (UOW – Unit of Work) der Transaktion zu verwenden ist. Dies wird als *dynamische Planauswahl* bezeichnet.

Ein dynamischer Planexit kann auch beim Start einer nachfolgenden Arbeitseinheit ausgelöst werden (vorausgesetzt der Thread wurde am Synchronisationspunkt freigegeben), um zu ermitteln, welcher Plan für die nächste Arbeitseinheit verwendet werden soll. Der Planexit kann entscheiden, einen anderen Plan zu verwenden. Dies wird als *dynamischer Planwechsel* bezeichnet.

Wenn ein dynamischer Planexit verwendet wird, tritt die Db2-Planzuordnung erst bei Ausführung der ersten SQL-Anweisung in einem Programm auf oder nachdem das Programm einen Synchronisationspunkt setzt und die Steuerung durch Verlinken oder Übertragen an ein anderes Programm übergeht, das ein separates Datenbankanforderungsmodul aufweist.

Dies wird durch die Verwendung eines Exitprogramms erreicht, das in einem der folgenden Objekte angegeben ist:

- DB2ENTRY-Objekt, Exit für einen bestimmten Transaktionscode, der im Schlüsselwort PLANEXITNAME angegeben ist.
- DB2CONN-Objekt, Exit für Transaktionen unter Verwendung des Pools, der im Schlüsselwort PLANEXITNAME angegeben ist.

IBM stellt mit DSNCEXIT und DFHD2PXT zwei Beispiexitprogramme für Assemblersprache sowohl als Quelle als auch als Objektcode bereit. Sie können auch andere Exitprogramme schreiben.

Beachten Sie, dass die Verwendung der folgenden Elemente innerhalb des dynamischen Exitprogramms nicht unterstützt wird:

- SQL-Befehle.
- IFI-Aufrufe.
- EXEC CICS SYNCPOINT-Befehle.

Beispielexitprogramme DSNCUEXT und DFHD2PXT

Die zwei Exitprogramme für Assemblersprache DSNCUEXT und DFHD2PXT werden sowohl in Quellcode als auch in Objektcode bereitgestellt. Die Programme sind funktional gleich; die von CICS für DSNCUEXT bereitgestellte Programmdefinition gibt CONCURRENCY(QUASIRENT) an, während die von CICS für DFHD2PXT bereitgestellte Programmdefinition CONCURRENCY(THREADSAFE) angibt.

Es gibt zwei Beispielprogramme, da in dem Fall, wenn CICS die Umgebung für offene Transaktionen (OTE) nutzt, der taskbezogene CICS-Db2-Benutzerexit als threadsicheres Programm betrieben wird und die Steuerung für einen offenen Tasksteuerblock (Modus L8) übernehmen kann. Weitere Erläuterungen finden Sie in „CICS-Db2-Anwendungen für die Verwendung von OTE durch threadsichere Programmierung aktivieren“ auf Seite 90. Wenn das Anwendungsprogramm, das die Db2-Anforderung ausgegeben hat, threadsicher ist, kann es auch im offenen Tasksteuerblock ausgeführt werden. In dieser Situation sollte kein Tasksteuerblockwechsel erforderlich sein.

Wenn jedoch ein dynamischer Planexit verwendet wird, der mithilfe von CONCURRENCY(QUASIRENT) definiert wurde (wie DSNCUEXT), bewirkt dies einen Wechsel zurück zum QR-Tasksteuerblock, wodurch zusätzliche Kosten verursacht werden. Dynamische Planexits, die mit CONCURRENCY(THREADSAFE) definiert wurden und für threadsichere Standards codiert wurden (wie DFHD2PXT), können im offenen Tasksteuerblock ausgeführt werden und verursachen keine zusätzlichen Kosten.

Wenn also zwischen CICS mit Db2 verbunden ist, verwenden Sie DFHD2PXT als dynamischen Planexit und nicht DSNCUEXT. Wenn Sie dem bereitgestellten Beispielprogramm nicht threadsichere Logik hinzufügen oder nicht threadsichere CICS-Befehle im Exit absetzen, müssen Sie beachten, dass hierbei ein Wechsel zurück zum QR-Tasksteuerblock auftritt und zusätzliche Kosten verursacht werden. Um durch das Vermeiden von Tasksteuerblockwechseln Leistungsvorteile zu erzielen, verwenden Sie eine threadsichere Codelogik und einen dynamischen Planexit, der als threadsicher definiert ist, und stellen Sie außerdem sicher, dass das Programm nur threadsichere Befehle enthält.

Der Objektcode (Lademodul) für die Beispiel-Planexits ist in der Bibliothek SDFHLOAD enthalten. DSNCUEXT ist der standardmäßige dynamische Planexit und er wird als ein durch den Benutzer austauschbares CICS-Programm aufgerufen. Der Quellcode für beide Beispielprogramme ist in der Assemblersprache geschrieben und wird in der Bibliothek bereitgestellt. Die Beispielprogramme zeigen, wie die Parameterliste angesprochen werden kann; sie ändern jedoch nicht den Plannamen.

Eine Parameterliste wird über einen Kommunikationsbereich (COMMAREA) an ein Beispielprogramm übergeben und weist in der Assemblerversion folgendes Format auf:

Format der Assemblerversion			
CPRMPLAN	DS	CL8	Der Name des Datenbankanforderungsmoduls/Plans der ersten SQL-Anweisung bei Eingabe in das Beispielprogramm. Das Feld kann geändert werden, um einen neuen Plan einzurichten.
CPRMAUTH	DS	CL8	Die aktuelle Berechtigungs-ID, die zum Zeitpunkt der Anmeldung an Db2 übergeben wird. Dies dient nur der Information. Alle an der ID vorgenommenen Änderungen werden ignoriert.
CPRMUSER	DS	CL4	Ein Benutzerbereich, der zur Verwendung durch das Beispielprogramm reserviert ist. Die CICS-Db2-Anschlussfunktion behält dieses Feld auch bei erneuten Aufrufen des Beispielprogramms bei.
CPRMAPPL	DS	CL8	Der Name des Anwendungsprogramms, das den SQL-Aufruf ausgegeben hat.

- Die Assemblerversion der Parameterliste wird als Member DSNCPRMA in der Bibliothek SDFHMAC geliefert.
- Die COBOL-Version ist DSNCPRMC in der Bibliothek SDFHCOB.
- Die PL/I-Version ist DSNCPRMP in der Bibliothek SDFHPLI.

Bevor Sie den dynamischen Planexit aufrufen, legt die CICS-Db2-Anschlussfunktion als Wert für CPRMPLAN den Namen des Datenbankankforderungsmoduls fest, der in der Parameterliste der ersten EXEC SQL-Anweisung festgelegt ist, die in der Arbeitseinheit ausgeführt wurde. Wie von CICS bereitgestellt, ändern die dynamischen Planexits DSNCUEXT und DFHD2PXT nicht den Plannamen, der von der CICS-Db2-Anschlussfunktion in CPRMPLAN eingegeben wurde, sondern kehren sofort zurück, wobei der Planname so wie von der CICS-Db2-Anschlussfunktion ausgewählt beibehalten wird.

Durch das Hinzufügen der JDBC- und SQLJ-Unterstützung für Java-Anwendungen für CICS wurden die von CICS bereitgestellten dynamischen Planexits geändert. Für SQLJ und JDBC ist erforderlich, dass Db2 für jedes Anwendungsprogramm vier Datenbankankforderungsmodule erstellt, um die dynamische Änderung von Isolationsstufen zu unterstützen. Für JDBC- und SQLJ-Anwendungen ist der Name des Datenbankankforderungsmoduls auf sieben Zeichen beschränkt, damit das achte Zeichen als Suffix (1, 2, 3 oder 4) verwendet werden kann. Daher können Sie für JDBC- und SQLJ-Anwendungen nicht eine standardmäßige Namenskonvention wie *programmname* gleich *name_des_datenbankanforderungsmoduls* gleich *planname* verwenden, da der Name des Datenbankankforderungsmoduls das Suffix 1, 2, 3 oder 4 aufweist.

Für JDBC-Anwendungen, SQLJ-Anwendungen und gemischte JDBC- und SQLJ-Anwendungen verwendet der Db2-JDBC-Treiber Informationen aus dem JDBC-Profil, um den Namen des Datenbankankforderungsmoduls in der Parameterliste der ersten ausgeführten EXEC-SQL-Anweisung festzulegen. Für das erste ausgegebene SQL wird als Name des Datenbankankforderungsmoduls stets das JDBC-Basisprogramm festgelegt, wobei die Standardisolationsstufe angehängt ist; das heißt, standardmäßig DSNJDBC2. Wenn zum Beispiel das JDBC-Profil mithilfe von *pgmname=OTHER* generiert wird, lautet der Name des Datenbankankforderungsmoduls OTHER2. Wenn keine dynamischen Planexits verwendet werden, wird der Planname aus der DB2CONN- oder der DB2ENTRY-Definition abgerufen; der Planname in der Eigenschaftendatei wird ignoriert.

Zur Unterstützung einer standardmäßigen Namenskonvention für IBM Data Server Driver for JDBC and SQLJ können die von CICS bereitgestellten dynamischen Planexits DSNCUEXT und DFHD2PXT einen eingegebenen CPRMPLAN-Namen erkennen, dessen erste Zeichen SYSSTAT, SYSLH oder SYSLN sind. Wenn ein solcher Planname erkannt wird, wird der Planname in "DSNJCC " geändert (das siebte und achte Zeichen werden als Leerzeichen festgelegt). Wenn Sie die standardmäßigen dynamischen Planexits mit Java-Anwendungen für CICS verwenden möchten, müssen Sie die Datenbankankforderungsmodule in einen Plan binden, der den Namen DSNJCC aufweist.

Eigenes Exitprogramm schreiben

Sie können Ihr Exitprogramm in Assembler, COBOL oder PL/I schreiben.

Informationen zu diesem Vorgang

Das Exitprogramm ist ein CICS-Programm, d. h. es muss:

- Normale CICS-Konventionen beachten.
- Für CICS definiert sein (es sei denn, die automatische Installation von Programmen wird verwendet).
- Anweisungen auf CICS-Befehlsebene verwenden.
- Zu CICS mithilfe des Befehls EXEC CICS RETURN zurückkehren.

Das Programm der CICS-Db2-Anschlussfunktion übergibt mithilfe eines Kommunikationsbereichs (COMMAREA) eine Parameterliste an das Exitprogramm. Das Exitprogramm kann den in der Parameterliste angegebenen Standardplannamen (Namen des Datenbankankforderungsmoduls) ändern, wenn die erste SQL-Anweisung von der CICS-Db2-Anschlussfunktion verarbeitet wird. Der Name gibt den Plannamen für diese Ausführung der Transaktion an.

Wenn CICS mit DB2 Version 7 oder höher verbunden ist, sollten Sie wie in „CICS-Db2-Anwendungen für die Verwendung von OTE durch threadsichere Programmierung aktivieren“ auf Seite 90 erklärt sicherstellen, dass die Logik in Ihrem dynamischen Planexit für threadsichere Standards codiert und das Programm als threadsicher definiert wurde (wie das Beispiexitprogramm DFHD2PXT). Wurde das Programm nicht als threadsicher definiert, bewirkt jede Verwendung des Programms ein Umschalten zum QR-Tasksteuerblock, wodurch zusätzliche Kosten verursacht werden. Ist das Programm als threadsicher definiert, verwendet jedoch keine threadsicheren CICS-Befehle (dies ist zulässig), bewirkt jeder nicht

threadsichere Befehl, dass eine Umschaltung zum QR-Tasksteuerblock erfolgt und zusätzliche Kosten verursacht werden.

Wenn Sie Pläne für eine Anwendung erstellen müssen, die bereits entwickelt wurde

Verwenden Sie dieses Verfahren, wenn die Db2-Planaspekte bei der Entwicklung der Anwendungen wenig berücksichtigt wurden. Nachdem die Anwendung vollständig entwickelt wurde, werden die Pläne so definiert, dass sie mit der Transaktion übereinstimmen.

Informationen zu diesem Vorgang

Im Allgemeinen wird nicht empfohlen, Pläne zu definieren, wenn die Anwendung bereits entwickelt wurde; dieses Verfahren ist jedoch für *Konvertierungsprojekte* nützlich, bei denen das Anwendungsdesign unverändert bleibt, die Anwendung jetzt jedoch Db2 verwendet.

Die Vor- und Nachteile dieses Verfahrens hängen vollständig vom tatsächlichen Anwendungsdesign und dem Ergebnis der Schritte ab.

Vorgehensweise

Beim Definieren der Db2-Pläne und der DB2ENTRY-Spezifikationen können Sie folgende Schritte ausführen:

1. Analysieren Sie für jedes Programmmodul mit SQL-Aufrufen, unter welchen CICS-Transaktionscodes es ausgeführt werden kann.
Es ist wahrscheinlich, dass ein bestimmtes Programmmodul unter mehreren CICS-Transaktionscodes verwendet wird.

Die Ausgabe dieses Schritts ist eine Liste von Datenbankankforderungsmodulen für jede einzelne Transaktion.

2. Entscheiden Sie für jeden CICS-Transaktionscode, welchen Plan er verwenden soll.
Pro CICS-Transaktionscode kann im DB2ENTRY-Objekt nur ein einziger Plan angegeben werden. Mehrere Transaktionen können denselben Plan verwenden.

Für diesen Schritt haben Sie viele Alternativen. Die mögliche Anzahl der zu verwendenden Pläne liegt zwischen eins und der Anzahl der verschiedenen Transaktions-IDs. Die beste Möglichkeit für die Verwaltung der Pläne besteht darin, Pakete zu verwenden; dabei werden alle Datenbankankforderungsmodule in Pakete gebunden, die dann in Plänen aufgelistet werden. Weitere Informationen finden Sie unter [„Db2-Pakete verwenden“](#) auf Seite 74.

Wenn Sie keine Pakete verwenden, finden Sie eine Beschreibung alternativer Verfahren in [„Für alle Transaktionen einen großen Plan verwenden“](#) auf Seite 78, [„Viele kleine Pläne verwenden“](#) auf Seite 78 und [„Pläne auf der Grundlage der Transaktionsgruppierung verwenden“](#) auf Seite 79.

Beispiel

Angewendet auf das Beispiel in [Abbildung 26 auf Seite 74](#) wäre Folgendes eine mögliche Lösung:

- Ein Plan, PLAN0, der die Datenbankankforderungsmodule aus P0, P1, P2, P3 und PB verwendet, die von der Transaktions-ID TRX0 verwendet werden
- Ein Plan, PLAN1, der die Datenbankankforderungsmodule aus PA, PB, PC und PD verwendet, die von der Transaktions-ID TRXA verwendet werden
- Zwei DB2ENTRY-Definitionen, eine für jeden Plan

Wenn Sie Pläne innerhalb einer Transaktion wechseln müssen

Der Plan, den eine Transaktion für den Zugriff auf Db2 verwendet, muss die Datenbankankforderungsmodule aller Anwendungsprogramme, die unter dieser Transaktions-ID ausgeführt werden können, enthalten oder auf sie verweisen. Wenn der Plan nicht alle relevanten Datenbankankforderungsmodule enthält, sollten Sie möglicherweise in Betracht ziehen, innerhalb einer Transaktion Pläne zu wechseln.

Wenn das Design Ihrer Anwendungen geändert wurde, enthalten einige Ihrer älteren Pläne möglicherweise nicht alle relevanten Datenbankanforderungsmodule. In diesem Fall könnte das Anwendungsdesign im Verlauf einer Transaktion einen Wechsel zu einem anderen Plan erforderlich machen, wenn die Transaktion erfordert, dass ein nicht im aktuellen Plan enthaltenes Programm ausgeführt werden muss. Diese Situation könnte auch auftreten, wenn die Struktur der Db2-Pläne beim Entwerfen der Anwendung unzureichend berücksichtigt wurde.

Die beste Lösung für dieses Problem besteht darin, Pakete zu verwenden. Sie können das Datenbankanforderungsmodul des fehlenden Programms in ein Paket binden und das Paket einer Paketliste im vorhandenen Plan hinzufügen. Die Transaktion kann jetzt auf das fehlende Programm zugreifen, ohne dass Pläne gewechselt werden müssen. Es gibt zwei weitere mögliche Lösungen, die wie folgt erläutert werden:

- [„Dynamischer Planwechsel“ auf Seite 84](#)
- [„Transaktions-ID wechseln, um Pläne zu wechseln“ auf Seite 84](#)

Dynamischer Planwechsel

Verwenden Sie den dynamischen Planwechsel (DPS – Dynamic Plan Switching), wenn Sie in einer Transaktion mehrere Pläne verwenden möchten.

Mit dem dynamischen Planwechsel können Sie in einer Transaktion mehrere Pläne verwenden. Wie bereits bemerkt, sollte der dynamische Planwechsel in einer CICS-Transaktionsinstanz jedoch selten auftreten. Der dynamische Planexit wurde entworfen, um beim Start der Transaktion dynamisch einen Plan auszuwählen (dynamische Planauswahl), nicht um die Pläne innerhalb einer Transaktion häufig zu ändern.

Zum Ausführen des dynamischen Planwechsels muss der Thread am Synchronisationspunkt freigegeben und erneut angefordert werden, um den dynamischen Planexit auszulösen. Der dynamische Planexit kann anschließend verwendet werden, um den Plan für das Programm auszuwählen, das Sie ausführen müssen. Dadurch wird ermöglicht, dass Arbeitseinheiten innerhalb einer Transaktion unterschiedliche Pläne verwenden können.

Wenn Sie Ihren eigenen dynamischen Planexit codiert haben, müssen Sie überprüfen, dass die Logik mit nachfolgenden Aufrufen derselben Task zurechtkommt. Entweder die Benutzeranwendung oder der dynamische Planexit muss so geschrieben sein, dass die Folgen nachfolgender Aufrufe des Exits toleriert werden. Wenn der dynamische Planexit den Plan ändern würde, wenn dies nicht gewünscht ist, kann die Benutzeranwendung dies vermeiden, indem sie sicherstellt, dass der Thread am Synchronisationspunkt nicht freigegeben wird. Wenn der Thread freigegeben wird, muss der dynamische Planexit beim Aufrufen möglichst den richtigen Plan für die neuen Fälle bereitstellen, das heißt, ein DB2ENTRY-Objekt mit `THREADLIMIT > 0`.

Damit der dynamische Planexit aufgerufen werden kann, um nach dem Beenden einer Arbeitseinheit einen Planwechsel auszuführen, muss Ihre Transaktion den Thread am Synchronisationspunkt freigeben. Eine Transaktion gibt einen Thread am Synchronisationspunkt nur frei, wenn Folgendes gilt:

- Es handelt sich um eine vom Terminal gesteuerte Task oder um eine nicht vom Terminal gesteuerte Task und im DB2CONN-Objekt ist `NONTERMREL=YES` festgelegt.
- Es gibt keine Hold-Cursor, die offen sind.
- Alle geänderten Db2-Sonderregister wurden wieder in ihren Anfangszustand zurückgesetzt.
- Das Db2-Sonderregister `CURRENT DEGREE` wurde durch diese Transaktion nicht geändert.

Transaktions-ID wechseln, um Pläne zu wechseln

Sobald eine Transaktion gestartet wurde, können Sie den von der Transaktions-ID verwendeten Plan nicht mehr steuern; Sie können jedoch die Transaktions-ID selbst steuern. Daher können Sie Pläne wechseln, indem Sie Transaktions-ID wechseln. Dies wird jedoch nicht empfohlen.

Informationen zu diesem Vorgang

Statt für einen Wechsel von Plänen Transaktions-IDs zu wechseln, wird empfohlen, das Datenbankanforderungsmodul des fehlenden Programms in ein Paket einzubinden und das Paket einer Paketliste im vorhandenen Plan hinzuzufügen.

Wenn Sie Transaktions-IDs wechseln müssen, ist zu beachten, dass in den meisten Fällen das erste Programm Daten an das nächste Programm überträgt. Die bevorzugte Methode, um dies auszuführen, ist die Verwendung des Befehls EXEC CICS RETURN IMMEDIATE. Alternativ können Sie mithilfe des Befehls EXEC CICS START oder durch Verwenden einer Warteschlange mit transienten Daten mit der Auslöseebene eine neue CICS-Task für dasselbe Terminal starten. Das alte Programm sollte RETURN an CICS absetzen, damit die neue Task startet. Bei beiden dieser Verfahren für einen Wechsel wird die in einer Benutzertransaktion ausgeführte Arbeit in mindestens zwei Arbeitseinheiten aufgeteilt. Wenn die neue Task zurückgesetzt wird, bleibt die in der ersten Task ausgeführte Arbeit festgeschrieben.

Wenn Sie zum Wechseln von Plänen Transaktions-IDs wechseln, ist die Logik, die für die Entscheidung erforderlich ist, wann die Transaktions-ID gewechselt werden muss, in den Anwendungsprogrammen enthalten. Wenn Sie die Planstruktur ändern möchten (beispielsweise aus Leistungs- oder operativen Gründen) bedeutet dies, dass Sie auch die Anwendungsprogramme ändern müssen.

Mithilfe eines anderen Verfahrens können Sie den Programmablauf über eine Tabelle steuern, anstatt dies mithilfe von Code in mehreren Programmen auszuführen. Der Hauptgedanke dabei ist, dass es einfacher ist, die Beziehung zwischen den Plänen, Programmen und Transaktions-IDs in einer Tabelle zu verwalten, als in jedem Anwendungsprogramm Code zum Ausführen der Logik zu verwalten. Das Entwickeln der Anwendungsprogramme ist darüber hinaus einfacher, falls eine Standardschnittstelle zur Verfügung steht.

Steuertabelle zum Steuern des Programmablaufs verwenden

Es wird empfohlen, dass Sie zum Steuern des Programmablaufs Pakete verwenden, statt dieses Verfahrens.

Bei dem dargestellten Entwurfsprinzip handelt es sich um ein Beispiel für eine Standardmethode, die zum Implementieren verschiedener Typen von Anwendungsdesign verwendet werden kann. Sie ermöglicht beispielsweise die Verwendung eines einzigen großen Plans oder vieler kleiner Pläne, *ohne* dass Programme geändert werden müssen.

In [Tabelle 5 auf Seite 85](#) wird ein Beispiel für den Inhalt einer Steuertabelle gezeigt. Das Beispiel basiert auf den in [Abbildung 26 auf Seite 74](#) beschriebenen Designsituationen.

Tabelle 5. Steuertabelle für Beispielanwendung				
Funktionsname	Programm	Transaktion	Planname	Neue Transaktions-ID (TRANS ID)
	P0	TRX0	PLAN0	*
Sales	P1	TRX0	PLAN0	
Order	P2	TRX0	PLAN0	
Pay	P3	TRX0	PLAN0	
	PA	TRXA	PLANA	*
Price	PB	TRXA	PLANA	
Order	PC	TRXA	PLANA	
Parts	PD	TRXA	PLANA	
Price	PB	TEMP	PLANX	*

Funktionsname

Das Feld für den Funktionsnamen in der Tabelle ergänzt das Programmfeld. Es arbeitet in exakt derselben Weise. Es wird in den Fällen verwendet, in denen der Terminalbenutzer einen Funktionsnamen in ein Befehlsfeld eingibt, das mit einem Schlüssel bereitgestellt wird. Das Programm PFCP (Program Flow Control Program) kann entweder einen Funktionsnamen oder einen Programmnamen akzeptieren.

PFCP verwendet anschließend die Funktionsspalte, um nach einer gültigen Transaktion zu suchen.

Auf diese Weise wird die Logik für die Interpretation der Auswahlmöglichkeiten des Benutzers auch aus den Programmen entfernt und in die Tabelle versetzt, wo sie einfach zu verwalten ist.

Programm

Der Name des in dieser Zeile beschriebenen Programms.

Transaktion

Der Name der Transaktions-ID, unter dem das Programm in der Programmspalte ausgeführt werden kann.

Plannamen

Das Feld für den Plannamen wird nicht verwendet. Es wird nur zur Veranschaulichung angezeigt. Es zeigt den Plannamen an, der von der entsprechenden Transaktion verwendet wird.

Neue Transaktions-ID (TRANS ID)

Ein * in dieser Spalte der Tabelle bedeutet, dass die entsprechende Zeile verwendet werden kann, wenn nach einer neuen Transaktions-ID gesucht wird, um ein bestimmtes Programm zu starten.

Die Steuertabelle spiegelt die folgenden wichtigsten Beziehungen wider:

- Beziehungen zwischen Plänen und Programmen/Datenbankanforderungsmodulen, die in der Db2-Katalogtabelle SYSIBM.SYSDBRM beschrieben sind
- Beziehungen zwischen Transaktionscodes und Plänen, die mithilfe der CICS-Definitionen DB2ENTRY und DB2TRAN beschrieben werden

Es gibt mehrere Möglichkeiten, um die Steuertabelle zu implementieren. Die nützlichsten Lösungen sind möglicherweise entweder eine Db2-Tabelle oder eine Hauptspeichertabelle.

Eine *Db2-Tabelle* ist hinsichtlich der Entwicklung und Verwaltung am einfachsten. Für jeden Aufruf von PFCP sind ein oder zwei SELECT-Aufrufe erforderlich. Diese SELECT-Aufrufe sollen nur eine einzige Zeile zurückgeben und es können Indizes verwendet werden. Die Daten- und Indexseiten werden häufig referenziert und bleiben wahrscheinlich im Pufferpool. Die Auswirkung auf die Antwortzeit ist also minimal.

Auf eine *Hauptspeichertabelle* kann schneller zugegriffen werden, zumindest ist dies der Fall, bis eine bestimmte Anzahl von Zeilen in der Tabelle erreicht wurde. Die Verwaltung ist komplizierter.

Das Prinzip des Programmablaufs wird in [Abbildung 27 auf Seite 87](#) gezeigt.

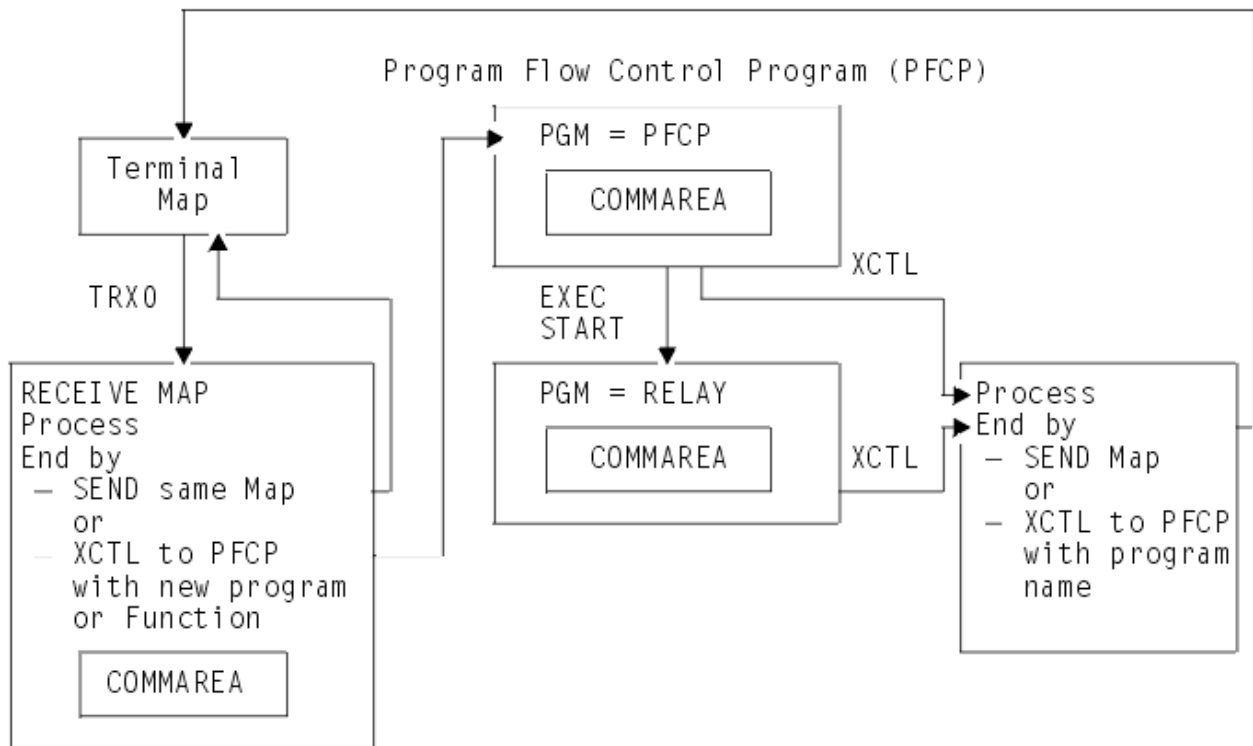


Abbildung 27. Tabellengesteuerter Programmablauf

Der Ablauf für das in [Abbildung 26 auf Seite 74](#) verwendete Anwendungsdesign ist wie folgt:

1. Der Terminalbenutzer sendet eine Transaktion an CICS. Die Transaktions-ID lautet TRX0.
2. Die Transaktionsdefinition verweist auf Programm P0.
3. Programm P0 empfängt die Zuordnung, führt einige Verarbeitungsschritte aus und entscheidet, dass Programm 3 benötigt wird.
4. Anstatt die Steuerung an Programm 3 (das Datenbankanforderungsmodul für P3 könnte Teil eines anderen Plan sein) zu übertragen, überträgt P0 die Steuerung an das Steuerprogramm für den Programmablauf (in diesem Beispiel PFCP). P0 übergibt auch einen Kommunikationsbereich (COMMAREA).
5. PFCP führt eine Tabellensuche in der Steuertabelle aus, um zu ermitteln, ob Programm P3 in den aktuell verwendeten Plan (PLAN0) eingeschlossen ist. Dies wird durch Prüfen ausgeführt, ob P3 in derselben Zeile wie die aktuelle Transaktions-ID (TRX0) angegeben ist. Im Beispiel ist dies der Fall (Zeile 4 in der Tabelle).
6. PFCP überträgt die Steuerung anschließend an Programm P3. Es überträgt auch den von P0 empfangenen Kommunikationsbereich (COMMAREA) an P3.
7. P3 verarbeitet die erforderlichen SQL-Aufrufe und wird beendet, entweder durch Senden einer Zuordnung an das Terminal oder durch Übertragen der Steuerung an PFCP (nicht dargestellt), um ein anderes Programm auszuführen.
8. Vorausgesetzt, dieses andere Programm ist PB, so prüft PFCP erneut, ob PB die Berechtigung hat, unter der aktuellen Transaktions-ID (dies ist immer noch TRX0) ausgeführt zu werden.

Die Tabelle zeigt, dass PB nicht unter TRX0 ausgeführt werden darf. PFCP prüft dann die Tabelle, um eine Transaktions-ID zu finden, unter der Programm PB ausgeführt werden kann. Im Beispiel sind sowohl TRXA als auch TEMP gültige Transaktions-ID. TRXA verweist jedoch in der Transaktionsdefinition auf Programm PA. Die Spalte 'New_TRANS_ID' der Tabelle zeigt, dass nur die Zeilen mit einem Stern (*) verwendet werden können, wenn eine neue Transaktions-ID gesucht wird, die ein bestimmtes Programm starten soll. In diesem Fall ist dies die Transaktion TEMP.

Es gibt im Eintrag der RDO-Transaktionsdefinition für die Transaktions-ID TEMP zwei Möglichkeiten für Programmnamen:

- Die RDO-Transaktionsdefinition kann auf das neue Programm (PB) direkt verweisen. In diesem Fall muss für jedes Programm, das auf diese Weise gestartet werden könnte, eine Transaktions-ID vorhanden sein. Damit der Kommunikationsbereich (COMMAREA) verwendet werden kann, muss das gestartete Programm außerdem Logik enthalten, um zu ermitteln, ob es durch START oder durch Übergabe der Steuerung mithilfe des Befehls XCTL gestartet wird.
- Die RDO-Transaktionsdefinition kann auf ein gemeinsames Programm verweisen, das hier als Relay-Programm (RELAY) bezeichnet wird. In diesem Fall können eine oder mehrere Transaktions-IDs verwendet werden. Alle von ihnen verweisen auf das Relay-Programm (RELAY) in der RDO-Transaktionsdefinition. Der Zweck von RELAY ist die Übergabe der Steuerung an das entsprechende Programm. Alle diese Programme werden anschließend nicht mehr mit START gestartet und müssen diese Situation nicht mehr handhaben.

Die Lösung mithilfe des Relay-Programms (RELAY) wird in [Abbildung 27 auf Seite 87](#) dargestellt.

9. PFCP startet die Transaktion TEMP, wobei der Kommunikationsbereich (COMMAREA) übergeben wird.
10. Das Relay-Programm (RELAY) wird gestartet. Es muss den Befehl EXEC CICS RETRIEVE verwenden, um den Kommunikationsbereich (COMMAREA) abzurufen.
11. RELAY nimmt aus dem Kommunikationsbereich (COMMAREA) den Programmnamen PB auf.
12. RELAY überträgt die Steuerung an PB, wobei der Kommunikationsbereich (COMMAREA) übergeben wird.
13. Der Planwechsel ist abgeschlossen.

Vorteile

- Diese Methode ermöglicht Ihnen die Implementierung verschiedener Typen von Anwendungsdesign, wie beispielsweise die Möglichkeit, einen einzigen großen Plan oder viele kleine Pläne zu verwenden.
- Die Entscheidung darüber, wann Pläne gewechselt werden, wird aus dem Entwicklungsprozess entfernt und ist nicht Teil der Codierung.
- Wenn neue Programme in die Produktion versetzt werden, muss nur die Steuertabelle aktualisiert werden. Die vorhandenen Programme müssen nicht geändert werden, auch wenn sie die neuen Funktionen aufrufen können.
- Die Beziehung zwischen den Transaktions-IDs, den Db2-Plänen und den Programmen kann geändert werden, ohne dass die Programme selbst geändert werden müssen. Die Steuertabelle muss dann jedoch geändert werden.
- Informationen aus dem Db2-Katalog (SYSPLAN und SYSDBRM) können zum Erstellen der Steuertabelle verwendet werden.
- Alternativ dazu kann die Steuertabelle zum Generieren von Informationen zur Struktur der Datenbankanforderungsmodule der Pläne verwendet werden.
- Die Steuertabelle enthält Informationen, die Unterstützung bei der Definition von DB2ENTRY- und DB2TRAN-Objekten in CICS leisten können (wenn die Spalte für Plannamen verfügbar ist).
- Es können weitere Funktionen in eine Steuertabellenstruktur eingeschlossen werden, wie beispielsweise Informationen dazu, welche Transaktions-ID in der Option TRANSID des Befehls EXEC CICS RETURN zu verwenden ist.

Nachteile

Es gibt mehrere Nachteile, wenn zum Wechseln von Plänen Transaktions-IDs gewechselt werden.

Die beiden größten Nachteile dieses Verfahrens liegen in den Kosten für das Design und die Entwicklung der Lösung sowie in der aufzuwendenden Ausführungszeit.

Die Kosten für den Übergang von Programm zu Programm sind ungefähr doppelt so hoch. Dies sollte jedoch normalerweise nur einen Anstieg von einigen wenigen Prozent bei der für die Transaktion erforderli-

chen Prozessorzeit bedeuten. Um zu entscheiden, ob diese Lösung zu verwenden ist, sollten Sie die Nachteile und die Vorteile gegeneinander abwägen.

Sperrstrategie in der CICS-Db2-Umgebung entwickeln

Db2 verwendet einen Sperrmechanismus, um gemeinsamen Zugriff zu ermöglichen, während die Datenintegrität beibehalten wird.

In einer CICS-Umgebung gibt es erwartungsgemäß ein hohes Maß an gemeinsamem Zugriff. Um einen maximal möglichen gemeinsamen Zugriff zu ermöglichen, sollten Sie Sperrungen auf Zeilenebene oder Seitensperren statt Sperrungen des Tabellenbereichs verwenden. Sie erreichen dies, indem Sie beim Erstellen des Tabellenbereichs LOCKSIZE(PAGE), LOCKSIZE(ROW) oder LOCKSIZE(ANY) definieren und die Cursorstabilität zur Bindezeit (BIND) als Isolationsstufe definieren. Weitere Informationen hierzu finden Sie in [Db2-Leistung verwalten in der Produktdokumentation zu Db2 for z/OS](#).

Das Angeben von LOCKSIZE(ANY) ermöglicht Db2 zu entscheiden, ob die Sperreneskalation für den Tabellenbereich stattfinden kann. Mit dem Db2-Parameter NUMLKTS wird die Anzahl der gleichzeitigen Sperren für einen Tabellenbereich angegeben. Wenn die Anzahl der Sperren den Wert für NUMLKTS überschreitet, findet die Sperreneskalation statt. Für NUMLKTS sollte dann ein Wert festgelegt werden, der so hoch ist, dass die Sperreneskalation beim normalen CICS-Betrieb nicht stattfindet.

Wenn ein Tabellenbereich gesperrt wird und der Plan mit RELEASE(DEALLOCATE) gebunden wurde, wird der Tabellenbereich zum Zeitpunkt von COMMIT nicht freigegeben, da nur Seitensperren freigegeben werden. Dies kann bedeuten, dass ein Thread und ein Plan die Verwendung des Tabellenbereichs monopolisieren.

Durch die Verwendung von ANY statt PAGE hat Db2 die Option, die Sperreneskalation für Programme zu verwenden, die vor der Festschreibung viele Seitensperren erfordern. Dies ist in der Regel in Stapelverarbeitungsprogrammen der Fall. Db2 stellt auch die Möglichkeit bereit, auf Zeilenebene und nicht auf Seiten- oder Tabellenbereichsebene zu sperren, wodurch eine bessere Granularität erzielt wird und weniger Sperrenkonflikte auftreten.

Sie können die Db2-Regeln für die Auswahl der ursprünglichen Sperrenattribute überschreiben, indem Sie in einem Anwendungsprogramm die SQL-Anweisung LOCK TABLE verwenden. Sie sollten jedoch vermeiden, die Anweisung LOCK TABLE zu verwenden, es sei denn, es ist unbedingt erforderlich. Wird die Anweisung LOCK TABLE in einem Onlineprogramm verwendet, wird die Verwendung von RELEASE(DEALLOCATE) und die Verwendung geschützter Threads möglicherweise behindert. Wenn Sie die Anweisung LOCK TABLE doch verwenden, sollte in Ihrem Plan die Bindeoption RELEASE(COMMIT) verwendet werden.

Im Allgemeinen wird empfohlen, CICS-Programme so zu entwerfen, dass Folgendes gilt:

- Sperren werden so kurz wie möglich gehalten.
- Die Anzahl der gleichzeitigen Sperren wird minimiert.
- Die Zugriffsreihenfolge der Tabellen ist für alle Transaktionen gleich.
- Die Zugriffsreihenfolge der Zeilen innerhalb einer Tabelle ist für alle Transaktionen gleich.

SQL, Threadsicherheit und andere Erwägungen bei der Programmierung für CICS-Db2-Anwendungen

Bei der Entwicklung von Anwendungsprogrammen müssen Sie bei der Programmierung bestimmte Aspekte beachten, um die Leistung Ihrer Anwendung zu verbessern und mögliche Probleme zu vermeiden. Ob Sie qualifiziertes oder nicht qualifiziertes SQL verwenden, beeinflusst viele andere Aspekte der CICS-Db2-Umgebung. Damit Sie die Leistungsvorteile der Umgebung für offene Transaktionen (OTE) nutzen können, müssen Ihre Anwendungsprogramme threadsicher sein.

CICS-Db2-Anwendungen für die Verwendung von OTE durch threadsichere Programmierung aktivieren

Die CICS-Db2-Anschlussfunktion enthält DFHD2EX1, einen taskbezogenen CICS-Db2-Benutzerexit (TRUE – Task-related User Exit), der aufgerufen wird, wenn ein Anwendungsprogramm eine SQL-Anforderung sendet. Dieser verwaltet den Prozess, mit dem eine Threadverbindung zu Db2 angefordert wird und die Steuerung nach dem Abschluss der Db2-Verarbeitung an das Anwendungsprogramm zurückgegeben wird.

Informationen zu diesem Vorgang

Die CICS-Db2-Anschlussfunktion verwendet die Umgebung für offene Transaktionen (OTE – Open Transaction Environment), um den taskbezogenen CICS-Db2-Benutzerexit (TRUE – Task-related User Exit) zu aktivieren, um Db2 aufzurufen und von Db2 zurückzukehren, ohne dass ein Wechsel von Tasksteuerblöcken erfolgt. In der Umgebung für offene Transaktionen wird CICS Db2 TRUE als ein threadsicheres und mit offener API ausgestattetes TRUE-Programm betrieben; es wird automatisch aktiviert, wenn während der Verbindungsverarbeitung die Option OPENAPI im Befehl ENABLE PROGRAM angegeben wird. Dadurch übernimmt es die Steuerung für einen offenen L8-Tasksteuerblock. Anforderungen an Db2 werden ebenfalls in einem L8-Tasksteuerblock ausgegeben; das heißt, er agiert als der Thread-Tasksteuerblock und ein Wechsel zu einem Subtask-Tasksteuerblock ist nicht erforderlich.

Wenn in der Umgebung für offene Transaktionen das Benutzeranwendungsprogramm, das TRUE aufgerufen hat, den threadsicheren Codierungskonventionen entspricht und für CICS als threadsicher definiert ist, kann es auch im L8-Tasksteuerblock ausgeführt werden. Bevor das Anwendungsprogramm seine erste SQL-Anforderung ausgibt, wird es im CICS-Haupt-Tasksteuerblock ausgeführt, dem QR-Tasksteuerblock (QR – Quasi-Reentrant, quasiwiedereintrittsfähig). Wenn es eine SQL-Anforderung ausführt und den taskbezogenen Benutzerexit (TRUE) aufruft, wird die Steuerung an den L8-Tasksteuerblock übergeben und die Db2-Verarbeitung wird ausgeführt. Ist das Anwendungsprogramm threadsicher, wird die Ausführung des Programms bei Rückgabe von Db2 im L8-Tasksteuerblock fortgesetzt.

Mit CONCURRENCY(REQUIRED) definierte Programme werden ab dem Start des Programms in einem offenen Tasksteuerblock ausgeführt. Für CICSAPI-Programme verwendet CICS einen offenen L8-Tasksteuerblock, unabhängig vom Ausführungsschlüssel des Programms. Für OPENAPI-Programme verwendet CICS einen L9-Tasksteuerblock, wenn EXECKEY(USER) festgelegt wurde, und einen L8-Tasksteuerblock, wenn EXECKEY(CICS) festgelegt wurde.

Wenn die richtigen Bedingungen erfüllt sind, verringert die Verwendung offener Tasksteuerblöcke für CICS-Db2-Anwendungen die Nutzung des QR-Tasksteuerblocks und ein Tasksteuerblockwechsel wird vermieden. Das ideale CICS-Db2-Anwendungsprogramm für die Umgebung für offene Transaktionen ist ein threadsicheres Programm, das nur threadsichere EXEC CICS-Befehle enthält und nur threadsichere Benutzerexitprogramme verwendet. Wenn eine solche Anwendung ihre erste SQL-Anforderung absetzt, wird sie in einen L8-Tasksteuerblock verschoben und ihre Verarbeitung wird dann weiterhin für eine beliebige Anzahl von Db2-Anforderungen und -Anwendungscode im L8-Tasksteuerblock fortgesetzt, wobei kein Tasksteuerblockwechsel erforderlich ist. Diese Situation führt zu einer erheblichen Leistungsverbesserung, bei der ein Anwendungsprogramm mehrere SQL-Aufrufe ausgibt. Wenn das Anwendungsprogramm nicht viele SQL-Aufrufe absetzt, sind die Leistungsvorteile möglicherweise nicht so erheblich.

Wenn die Ausführung des Programms Aktionen beinhaltet, die nicht threadsicher sind, wechselt CICS an diesem Punkt zurück zum QR-Tasksteuerblock. Zu solchen Aktionen gehören beispielsweise vom Programm ausgegebene, nicht threadsichere CICS-Anforderungen, die Verwendung von nicht threadsicheren dynamischen Planexits, die Verwendung von nicht threadsicheren TRUEs und die Beteiligung nicht threadsicherer globaler Benutzerexits (GLUEs). Das Wechseln zwischen dem offenen Tasksteuerblock und dem QR-Tasksteuerblock wirkt sich nachteilig auf die Leistung der Anwendung aus.

Damit Sie von den Leistungsvorteilen der Umgebung für offene Transaktionen (OTE) für CICS-Db2-Anwendungen profitieren können, müssen folgende Bedingungen erfüllt sein:

- CICS muss mit einer unterstützten Version von Db2 verbunden sein. Details zu den unterstützten Db2-Releases und den anzuwendenden APARs finden Sie unter <https://www.ibm.com/support/pages/uid/swg27018287>.

- Für den Systeminitialisierungsparameter **FORCEQR** darf nicht YES festgelegt sein. **FORCEQR** erzwingt, dass als threadsicher definierte Programme im QR-Tasksteuerblock ausgeführt werden; die Einstellung YES kann als vorübergehende Maßnahme verwendet werden, um Probleme im Zusammenhang mit als threadsicher definierten Programmen zu untersuchen und zu lösen.
- Die CICS-Db2-Anwendung muss eine threadsichere Anwendungslogik aufweisen (das heißt, der für EXEC CICS-Befehle verwendete landessprachliche Code muss threadsicher sein), nur threadsichere EXEC CICS-Befehle verwenden und für CICS als threadsicher definiert sein. In offenen Tasksteuerblöcken darf nur als threadsicher angegebener Code ausgeführt werden. Wenn Ihre CICS-Db2-Anwendung nicht als threadsicher definiert ist oder wenn sie nicht threadsichere EXEC CICS-Befehle verwendet, finden Tasksteuerblockwechsel statt und einige oder alle der durch die Verwendung der Umgebung für offene Transaktionen entstehenden Leistungsvorteile gehen verloren.
- Alle dynamischen Planexits, die von der CICS-Db2-Anschlussfunktion verwendet werden, müssen gemäß threadsicheren Standards codiert und für CICS als threadsicher definiert werden. Der standardmäßige dynamische Planexit DSNCEXT, der als vom Benutzer austauschbares CICS-Programm gestartet wird, ist für CICS nicht als threadsicher definiert, aber der alternative, von CICS bereitgestellte dynamische Beispielplanexit DFHD2PXT ist als threadsicher definiert. Weitere Informationen finden Sie in „[Dynamische Planexits](#)“ auf Seite 80.
- Alle globalen Benutzerexits (GLUEs) im von der Anwendung verwendeten Ausführungspfad müssen gemäß threadsicheren Standards codiert und für CICS als threadsicher definiert sein (für CICS-Db2-Anwendungen. Insbesondere die globalen Benutzerexits XRMIIN und XRMIOU).
- Alle anderen von der Anwendung verwendeten TRUEs müssen für CICS als threadsicher oder als OPEN-API definiert sein.

Informationen dazu, wie Anwendungsprogramme und Benutzerexitprogramme threadsicher gemacht werden können, finden Sie in [Threadsichere Programme](#). Wenn Sie für CICS ein Programm als threadsicher angeben, geben Sie damit nur an, dass die Anwendungslogik threadsicher ist; Sie geben damit nicht an, dass alle im Programm enthaltenen EXEC CICS-Befehle threadsicher sind. CICS kann sicherstellen, dass CICS-Befehle sicher verarbeitet werden, indem für Befehle, die noch nicht konvertiert sind und die noch auf die Quasiwiedereintrittsfähigkeit angewiesen sind, ein Wechsel zum QR-Tasksteuerblock erfolgt. Damit die Ausführung Ihres Programms in einem offenen Tasksteuerblock zugelassen wird, muss CICS Ihnen die Threadsicherheit Ihrer Anwendungslogik garantieren.

Die EXEC CICS-Befehle, die threadsicher sind und für die daher kein Tasksteuerblockwechsel erforderlich ist, werden in den Befehlssyntaxdiagrammen in der Beschreibung der API- und SPI-Befehle angegeben.

Wenn ein Benutzeranwendungsprogramm in der Umgebung für offene Transaktionen nicht als threadsicher definiert ist, wird CICS Db2 TRUE dennoch in einem L8-Tasksteuerblock ausgeführt, aber das Anwendungsprogramm wird während der gesamten Task im QR-Tasksteuerblock ausgeführt. Bei jeder vom Programm ausgeführten SQL-Anforderung wechselt CICS vom QR-Tasksteuerblock zum L8-Tasksteuerblock und wieder zurück; dadurch werden die durch die Umgebung für offene Transaktionen gewonnenen Leistungsvorteile zunichte gemacht. Die maximale Anzahl von Tasksteuerblockwechseln für eine CICS-Db2-Anwendung würde auftreten, wenn Ihr Programm nach jeder Db2-Anforderung ein nicht threadsicheres Benutzerexitprogramm und einen nicht threadsicheren EXEC CICS-Befehl verwendete. Insbesondere durch die Verwendung eines nicht threadsicheren Exitprogramms im CICS-Db2-Hauptpfad (beispielsweise ein Programm, das an XRMIIN oder XRMIOU aktiviert wird) wird eine höhere Anzahl von Tasksteuerblockwechseln verursacht, als wenn CICS mit früheren Versionen von Db2 verbunden ist.

Die Tabelle zeigt die Auswirkungen, die auftreten, wenn Anwendungsprogramme mit unterschiedlichen Attributen für den gemeinsamen Zugriff CICS Db2 TRUE aufrufen, während CICS mit verschiedenen Versionen von Db2 verbunden ist.

Tabelle 6. Kombinationen aus Anwendungsprogrammen und CICS Db2 TRUE

Programmattribut für den gemeinsamen Zugriff	Operation von CICS Db2 TRUE	Auswirkung
QUASIRENT	Threadsicher und offene API	Anwendungsprogramm wird im QR-Tasksteuerblock von CICS ausgeführt. Taskbezogene Benutzerexits (TRUEs) werden in einem L8-Tasksteuerblock ausgeführt und Db2-Anforderungen werden im L8-Tasksteuerblock ausgeführt. CICS wechselt für jede Db2-Anforderung zwischen dem QR-Tasksteuerblock und dem L8-Tasksteuerblock von CICS hin und her.
THREADSAFE	Threadsicher und offene API	Nutzung der Umgebung für offene Transaktionen. Taskbezogene Benutzerexits (TRUEs) werden in einem L8-Tasksteuerblock ausgeführt und Db2-Anforderungen werden im L8-Tasksteuerblock ausgeführt. Das Anwendungsprogramm wird auch im L8-Tasksteuerblock ausgeführt, wenn die Steuerung an diesen zurückgegeben wird. Bis zur Beendigung der Task oder bis zu dem Zeitpunkt, an dem eine nicht threadsichere CICS-Anforderung abgesetzt wird, die einen Wechsel zurück zum QR-Tasksteuerblock erzwingen würde, ist kein Tasksteuerblockwechsel erforderlich.
REQUIRED mit API(CICS-API)	Threadsicher und offene API	Nutzung der Umgebung für offene Transaktionen. Taskbezogene Benutzerexits (TRUEs) werden in einem L8-Tasksteuerblock ausgeführt und Db2-Anforderungen werden im L8-Tasksteuerblock ausgeführt. Das Anwendungsprogramm wird von Anfang an im L8-Tasksteuerblock ausgeführt. Das Programm verwendet immer einen L8-Tasksteuerblock, unabhängig vom Ausführungsschlüssel des Programms. Bis zur Beendigung der Task oder bis zu dem Zeitpunkt, an dem eine nicht threadsichere CICS-Anforderung abgesetzt wird, die einen Wechsel zurück zum QR-Tasksteuerblock und anschließend wieder einen Wechsel zum L8-Tasksteuerblock erzwingen würde, ist kein Tasksteuerblockwechsel erforderlich.
REQUIRED mit API(OPENAPI)	Threadsicher und offene API	Nutzung der Umgebung für offene Transaktionen. Ungünstig für CICS-Db2-Anwendungen mit Benutzerschlüssel (und wenn der Speicherschutz aktiv ist), da bei jeder Db2-Anforderung ein Wechsel vom L9-Tasksteuerblock zum L8-Tasksteuerblock und zurück erforderlich ist.

Gehen Sie wie folgt vor, um die Leistungsvorteile der Umgebung für offene Transaktionen zu erzielen (Zusammenfassung):

- CICS muss mit Db2 verbunden sein.
- Für FORCEQR darf nicht YES festgelegt sein.

- Die CICS-Db2-Anwendung muss eine threadsichere Anwendungslogik aufweisen (das heißt, der für EXEC CICS-Befehle verwendete landessprachliche Code muss threadsicher sein). Wenn die Anwendungslogik nicht threadsicher ist, muss das Programm als CONCURRENCY(QUASIRENT) definiert werden und damit im CICS-QR-Tasksteuerblock ausgeführt werden. In diesem Fall tritt bei jeder Db2-Anforderung ein Wechsel des Tasksteuerblocks auf, auch wenn der taskbezogene Benutzerexit (TRUE) in einem offenen Tasksteuerblock ausgeführt wird.
- Eine threadsichere Anwendung kann für CICS als CONCURRENCY(THREADSAFE) API(CICSAPI) oder CONCURRENCY(REQUIRED) API(CICSAPI) definiert werden. Welche Einstellung verwendet wird hängt von der Anzahl der nicht threadsicheren EXEC-Befehle ab, die vom Programm verwendet werden. Wenn viele nicht threadsichere CICS-Befehle vorhanden sind, wird das Programm am besten als CONCURRENCY(THREADSAFE) definiert. Wenn das Programm wenige oder keine nicht threadsicheren CICS-Befehle aufweist, kann CONCURRENCY(REQUIRED) verwendet werden. Programme, die mit CONCURRENCY(REQUIRED) definiert werden, haben den Vorzug, dass sie in einem offenen L8-Tasksteuerblock starten, aber jeder nicht threadsichere CICS-Befehl führt zu zwei Tasksteuerblockwechseln.
- Die CICS-Db2-Anwendung darf nur threadsichere oder dynamische Planexits mit offener API sowie TRUEs und GLUEs verwenden. Wenn nicht threadsichere Exits verwendet werden, wird dadurch ein Wechsel zurück zum QR-Tasksteuerblock erzwungen.

Sind alle diese Bedingungen erfüllt, können Sie die Leistungsvorteile der Umgebung für offene Transaktionen nutzen.

SQL (Sprache)

Die vollständige Sprache SQL steht dem CICS-Programmierer mit nur geringen Einschränkungen zur Verfügung.

Eine detaillierte Beschreibung zur Verwendung der Sprache SQL in einem CICS-Programm finden Sie in [Programmierung für Db2 for z/OS in der Produktdokumentation zu Db2 for z/OS](#).

Sie können in einem CICS-Programm Folgendes verwenden:

- Data Manipulation Language (DML)
- Data Description Language (DDL)
- Anweisungen GRANT und REVOKE

CICS unterstützt auch sowohl dynamische als auch statische SQL-Anweisungen.

Aus Gründen der Leistung und wegen des gemeinsamen Zugriffs wird jedoch empfohlen, dass Sie in CICS im Allgemeinen keine DDL-, GRANT- oder REVOKE-Anweisungen absetzen. Sie sollten auch die Verwendung von dynamischem SQL einschränken.

Der Grund für diese Empfehlungen besteht darin, dass die Db2-Katalogseiten gesperrt werden können; dies führt als Folge zu einer niedrigeren Standardstufe für den gemeinsamen Zugriff. Darüber hinaus ist die Ressourcennutzung für diese Typen von SQL-Anweisungen in der Regel höher als die Ressourcennutzung für statische SQL-DML-Anweisungen.

Qualifiziertes oder nicht qualifiziertes SQL verwenden

Programmierer, die CICS-Db2-Programme schreiben, können qualifiziertes oder nicht qualifiziertes SQL verwenden. Bei qualifiziertem SQL wird vor dem Namen der Tabelle oder Ansicht der Ersteller angegeben. Bei nicht qualifiziertem SQL wird der Ersteller nicht angegeben.

Wenn Programmierer CICS-Db2-Standards entwickeln, ist es wichtig, die Verwendung von qualifiziertem und nicht qualifiziertem SQL festzulegen. Diese Entscheidung wirkt sich auf viele andere Aspekte der Db2-Umgebung aus. Die wichtigsten Beziehungen zu anderen Db2-Bereichen und einige Folgen für die zwei Typen der SQL-Anweisungen werden in [Tabelle 7 auf Seite 93](#) dargestellt.

<i>Tabelle 7. Qualifiziertes und nicht qualifiziertes SQL</i>		
Beziehungen zu anderen Db2-Bereichen	Qualifiziertes SQL	Nicht qualifiziertes SQL
Verwendung von Synonymen	Nicht möglich	Möglich

<i>Tabelle 7. Qualifiziertes und nicht qualifiziertes SQL (Forts.)</i>		
Beziehungen zu anderen Db2-Bereichen	Qualifiziertes SQL	Nicht qualifiziertes SQL
Binder-ID	Beliebige	Wie Ersteller
Anzahl der Ersteller für Tabellen und Tabellenbereiche	Beliebige	Einer
Verwendung von VALIDATE(RUN)	Ist qualifiziert	Verwendet den Binder zum Qualifizieren
Verwendung von dynamischem SQL	Ist qualifiziert	Verwendet den Executor zum Qualifizieren
Erfordert für Tests ein separates Db2-Subsystem	Ja	Nein
Erfordert im Db2-Testsystem und im Db2-Produktionssystem denselben Ersteller	Ja	Nein
Möglichkeit, im selben für Tests bestimmten Db2-Subsystem mehrere Versionen der Testtabellen zu verwenden	Nein	Ja

Einige der in Tabelle 7 auf Seite 93 dargestellten Einschränkungen können umgangen werden, indem Sie Ihren eigenen Vorprozessor entwickeln, um den Quellencode zu ändern, bevor der Db2-Vorcompiler aufgerufen wird. Dadurch können Sie beispielsweise den Ersteller in den SQL-Anweisungen ändern.

Es wird empfohlen, dass Sie für dynamische SQL-Anweisungen qualifiziertes SQL verwenden, da es einfacher zu verwalten ist.

Wenn Sie nicht qualifiziertes SQL verwenden, müssen Sie entscheiden, wie das Element CREATOR bereitgestellt wird, um die Tabellen und Ansichten vollständig anzugeben. Es gibt zwei Möglichkeiten:

- Sie können Synonyme benutzen. Das Synonym muss mithilfe der Berechtigungs-ID erstellt werden, die im DB2ENTRY-Objekt und im DB2CONN-Objekt angegeben ist. Synonyme können nur von der Berechtigungs-ID selbst erstellt werden. Dies bedeutet, dass Sie eine Methode zum Erstellen der Synonyme entwickeln müssen. Sie können auch eine TSO-ID mit derselben ID verwenden, die im DB2ENTRY-Objekt und im DB2CONN-Objekt als Berechtigungs-ID angegeben ist. Eine andere Möglichkeit besteht darin, eine CICS-Transaktions-ID zu entwerfen (mithilfe derselben Berechtigungs-ID), die die Anweisung CREATE SYNONYM selbst erstellen kann. Keine dieser Methoden ist jedoch empfehlenswert.
- Wenn Sie keine Synonyme verwenden, ist der im Bindeprozess verwendete CREATOR die Berechtigungs-ID des Binders. Alle Tabellen und Ansichten, auf die im dynamischen SQL verwiesen wird, müssen dann mit dieser ID erstellt werden. Alle Transaktionen, die für den Zugriff auf eine gemeinsame Gruppe von Db2-Ressourcen dynamisches SQL verwenden, müssen dann die Berechtigungs-ID aufweisen, die auch im DB2ENTRY-Objekt oder DB2CONN-Objekt angegeben ist. In den meisten Fällen muss es sich um das SIGNID-Element oder um eine Zeichenfolge handeln. Diese Einschränkung ist normalerweise nicht akzeptabel.

Aus diesen Gründen wird die Verwendung von nicht qualifiziertem SQL in dynamischen SQL-Anweisungen nicht empfohlen.

Ansichten

Es wird generell empfohlen, Ansichten zu verwenden, wo dies angebracht ist. Beachten Sie, dass einige Ansichten nicht aktualisiert werden können.

In einem echtzeitorientierten Onlinesystem müssen Sie häufig Zeilen aktualisieren, die Sie mithilfe von Ansichten abgerufen haben. Wenn die Einschränkung für die Aktualisierung von Ansichten erzwingt, dass die Basistabellen direkt (oder durch Verwendung anderer Ansichten) aktualisiert werden, sollten Sie nur Ansichten in Betracht ziehen, die aktualisiert werden können. In den meisten Fällen erleichtert dies das Lesen und Ändern des Programms.

Indexspalten aktualisieren

Beim Aktualisieren von Indexspalten müssen Sie zwei Dinge berücksichtigen.

- Beim Aktualisieren eines Felds in einer Tabelle verwendet Db2 zum Abrufen der Zeilen keinen Index, der dieses Feld enthält. Dies schließt auch die Felder ein, die in der Liste FOR UPDATE OF und in der Anweisung DECLARE CURSOR aufgelistet sind. Dies ist unabhängig davon, ob das Feld aktualisiert wurde.
- Ein Tabellenbereich, der aktuell nicht partitioniert ist, kann neu erstellt werden und anschließend mehrere Partitionen aufweisen. SQL-Aktualisierungen sind für Schlüsselfelder der Partitionierung nicht zulässig. Dies bedeutet, dass Programme, die Aktualisierungen dieser Felder ausführen, so geändert werden müssen, dass sie stattdessen DELETE- und INSERT-Anweisungen verwenden.

Abhängigkeit von eindeutigen Indizes

Ein Programmierer kann die Tatsache nutzen, dass Db2 aus einer Tabelle mit einem eindeutigen Index nur eine Zeile zurückgibt, wenn der vollständige Schlüssel in der SELECT-Anweisung angegeben ist. Ein Cursor ist in diesem Fall nicht erforderlich.

Wird der Index geändert und die Eindeutigkeit gelöscht, so wird das Programm nicht ordnungsgemäß ausgeführt, wenn zwei oder mehr Zeilen zurückgegeben werden. Das Programm empfängt einen SQL-Fehlercode.

Verarbeitung der Festschreibung

CICS ignoriert in Ihren Anwendungsprogrammen alle EXEC SQL COMMIT-Anweisungen. Die Operation für die Db2-Festschreibung muss mit CICS synchronisiert werden; dies bedeutet, dass Ihr Programm den Befehl EXEC CICS SYNCPOINT absetzen muss. CICS führt dann die Verarbeitung der Festschreibung mit Db2 aus. Ein impliziter Synchronisationspunkt (SYNCPOINT) wird stets durch EXEC CICS RETURN am Ende einer Task aufgerufen.

Berücksichtigen Sie die Aktionen, die am SYNCPOINT stattfinden:

- Die Arbeitseinheit ist abgeschlossen. Dies bedeutet, dass alle Aktualisierungen sowohl in CICS als auch in Db2 festgeschrieben sind.
- Der Thread wird bei terminalorientierten Transaktionen freigegeben (es sei denn, ein Hold-Cursor ist offen). Wenn der Thread freigegeben wird und es keine Verwendung für ihn gibt, so wird er beendet, falls es sich nicht um einen geschützten Thread handelt.
- Der Thread wird bei nicht terminalorientierten Transaktionen nur freigegeben, wenn im DB2CONN-Objekt die Angabe NONTERMREL=YES vorhanden ist. Dies tritt erstmals beim Beenden der Transaktion auf.
- Alle geöffneten Cursor werden geschlossen.
- Alle Seitensperren werden freigegeben.
- Wenn im BIND-Prozess RELEASE(COMMIT) angegeben wurde:
 - Tabellenbereichssperren werden freigegeben
 - Die Cursortabellensegmente des Plans im EDM-Pool werden freigegeben.
- Tabellenbereichssperren, die mit dynamischem SQL abgerufen wurden, werden unabhängig von den BIND-Parametern freigegeben.

Transaktionen serialisieren

Möglicherweise müssen Sie die Ausführung einer oder mehrerer Transaktionen serialisieren. Dies tritt in der Regel auf, wenn die Anwendungslogik nicht für den Umgang mit gemeinsamem Zugriff entworfen wurde sowie in den Fällen, wenn das Risiko von Deadlocks hoch ist.

Informationen zu diesem Vorgang

Wegen der möglicherweise erforderlichen Zeit für die Warteschlangensteuerung dürfen Sie die Serialisierung nur für Transaktionen geringen Umfangs zulassen.

Die folgenden Methoden haben alle unterschiedliche Start- und Endzeiten für die Serialisierung:

CICS-Transaktionsklassen

Die CICS-Funktion, mit der in einer Klasse (CLASS) zu einem Zeitpunkt die Ausführung nur jeweils einer Transaktion zulässig ist, ist nützlich, um die vollständige Transaktion zu serialisieren.

Serialisierung von Db2-Threads

In den Fällen, in denen die Serialisierung möglicherweise auf das Intervall zwischen erstem SQL-Aufruf und Synchronisationspunkt begrenzt ist (für terminalorientierte Transaktionen und nicht terminalorientierte Transaktionen, wenn NONTERMREL=YES definiert ist), können Sie mithilfe Ihrer DB2ENTRY-Spezifikationen sicherstellen, dass gleichzeitig nur ein einziger Thread eines bestimmten Typs erstellt wird. Dieses Verfahren ermöglicht gemeinsamen Zugriff für den ersten Teil der Transaktion und es ist nützlich, wenn der erste SQL-Aufruf nicht am Anfang der Transaktion ist. Verwenden Sie dieses Verfahren nicht, wenn Ihre Transaktion andere Ressourcen aktualisiert hat, bevor sie ihre erste SQL-Anweisung ausgibt.

Verfahren zum Einreihen in eine Warteschlange und zum Entfernen aus einer Warteschlange in CICS

Wenn Sie wissen, dass der erforderliche Serialisierungszeitraum nur einen kleinen Teil des Programms darstellt, können die Verfahren zum Einreihen in eine Warteschlange bzw. zum Entfernen aus einer Warteschlange in CICS nützlich sein. Der Vorteil liegt darin, dass nur der kritische Teil der Transaktion serialisiert wird. Dieser Teil kann so klein sein, dass er nur eine einzige SQL-Anweisung umfasst. So wird eine höhere Transaktionsrate als bei den anderen Methoden ermöglicht, da die Serialisierung auf ein Minimum reduziert ist.

Der Nachteil gegenüber den anderen Verfahren liegt darin, dass die Serialisierung im Anwendungskode vorgenommen wird und daher eine Änderung des Programms erforderlich ist.

Anweisung LOCK TABLE

Es wird empfohlen, dass Sie die Anweisung LOCK TABLE *nicht* verwenden.

Die Anweisung LOCK TABLE kann verwendet werden, um CICS-Transaktionen und andere Programme zu serialisieren, falls der Modus EXCLUSIVE angegeben ist. Beachten Sie, dass der gesamte Tabellenbereich gesperrt wird, nicht die Tabelle, auf die in der Anweisung verwiesen wird.

Die Serialisierung startet, wenn die Anweisung LOCK ausgeführt wird. Die Endzeit für die Serialisierung ist der Zeitpunkt, an dem die Tabellenbereichssperre freigegeben wird. Dies kann am Synchronisationspunkt oder an dem Zeitpunkt sein, wenn die Zuordnung des Threads aufgehoben wird.

Verwenden Sie dieses Verfahren vorsichtig, da das Risiko besteht, dass der Tabellenbereich bis zu dem Zeitpunkt gesperrt bleibt, an dem die Zuordnung des Threads aufgehoben wird. Dieses Verfahren ist jedoch das einzige, das für das gesamte Db2-System funktioniert. Die anderen Verfahren beschränken sich auf die Steuerung der Serialisierung von ausschließlich CICS-Transaktionen.

Seitenkonflikt

Beim Entwerfen von Anwendungen und Datenbanken müssen Sie die Auswirkungen berücksichtigen, die auftreten, wenn viele Transaktionen auf denselben Teil eines Tabellenbereichs zugreifen. Der Begriff "Brennpunkt" (Hot Spot) wird häufig verwendet, um einen kleinen Teil des Tabellenbereichs zu beschreiben, in dem die Zugriffsdichte beträchtlich höher ist als im übrigen Teil des Tabellenbereichs.

Wenn die Seiten nur für die SELECT-Verarbeitung verwendet werden, treten keine Probleme mit dem gemeinsamen Zugriff auf. Die Seiten verbleiben wahrscheinlich im Pufferpool, sodass wenig E/A-Aktivität stattfindet. Wenn die Seiten jedoch häufig aktualisiert werden, kann es sein, dass Sie Probleme mit dem gemeinsamen Zugriff feststellen, da die Seiten von der ersten Aktualisierung bis zum Synchronisationspunkt gesperrt sind. Andere Transaktionen, die die gleichen Seiten verwenden, müssen warten. Häufig treten an Brennpunkten Deadlocks und Zeitlimitüberschreitungen auf.

Zwei Beispiele für Brennpunkte sind die Zuordnung fortlaufender Zahlen und aufeinander folgende Einfügungen.

Zuordnung fortlaufender Zahlen

Wenn Sie zum Bereitstellen neuer fortlaufender Zahlen für Ihre Anwendung einen oder mehrere Zähler verwenden, müssen Sie folgende Punkte beachten.

- Sie sollten für jeden Zähler die Häufigkeit der Aktualisierungen berechnen. Sie sollten auch die für die Aktualisierungstransaktion abgelaufene Zeit berechnen, gemessen ab Aktualisierung des Zählers bis zur Festschreibung. Wenn die Aktualisierungshäufigkeit multipliziert mit der berechneten abgelaufenen Zeit in Zeiten hoher Systemauslastung um ca. 0,5 höher liegt, ist die Zeit in der Warteschlange möglicherweise nicht akzeptabel.
- Wenn Sie erwägen, im selben Tabellenbereich mehrere Zähler einzusetzen, sollten Sie die Gesamtzeit berechnen, in der die Zähler beschäftigt sind.
- Sind die Zähler in dieselbe Zeile gesetzt, werden sie stets zusammen gesperrt.
- Wenn sie im selben Tabellenbereich in unterschiedliche Zeilen gesetzt sind, können sie sich auf derselben Seite befinden. Da Sperren auf Seitenebene angefordert werden, werden die Zeilen in diesem Fall auch gemeinsam gesperrt.
- Wenn erzwungen wird, dass die Zeilen im selben Tabellenbereich in verschiedene Seiten gesetzt werden (beispielsweise, indem 99 % freier Speicherbereich gewährt wird), ist es dennoch möglich, dass die Transaktionen in die Warteschlange eingereiht werden können.

Wenn beispielsweise auf Zeile 2 der Seite 2 zugegriffen wird, kann eine Durchsuchung des Tabellenbereichs auftreten. Die Durchsuchung stoppt für eine Wartezeit bei der Seitennummer 1, wenn diese Seite durch eine andere Transaktion gesperrt ist. Sie sollten daher eine Durchsuchung des Tabellenbereichs vermeiden.

- Wenn ein Index definiert ist, um die Durchsuchung des Tabellenbereichs zu vermeiden, ist unsicher, ob er verwendet werden kann. Wenn die Anzahl der Seiten im Tabellenbereich niedrig ist, wird der Index nicht verwendet.
- Es wäre dann eine Lösung, wenn in jedem Tabellenbereich nur ein einziger Zähler vorhanden ist. Diese Lösung wird bevorzugt, wenn mehrere CICS-Systeme auf die Zähler zugreifen.
- Wenn nur ein CICS-System auf die Zähler zugreift, kann die Verwendung einer BDAM-Datei (BDAM – Basic Direct Access Method) eine alternative Lösung darstellen. Die Möglichkeit, dass das CICS-System zu einem späteren Zeitpunkt in mehrere CICS-Systeme aufgeteilt werden kann, macht diese Lösung möglicherweise jedoch wenig attraktiv.

Aufeinander folgende Einfügungen

In Situationen, in denen mehrere Transaktionen Zeilen in denselben Tabellenbereich einfügen, müssen Sie die Reihenfolge der eingefügten Zeilen beachten.

Wenn Sie als Basis eines Clusterindex eine fortlaufende Zahl oder ein Feld mit einer Zeitmarke verwenden, versucht Db2, alle Zeilen angrenzend zueinander einzufügen. Die Seiten, auf denen die Zeilen eingefügt werden, können dann als Brennpunkt (Hot Spot) bezeichnet werden.

Beachten Sie, dass sich im Clusterindex alle Einfügungen innerhalb eines bestimmten Zeitraums auch auf derselben Seite befinden.

Wenn mehrere Indizes vorhanden sind und der Index ohne Clustering für den Datenabruf verwendet wird, erhöht sich das Risiko, dass zwischen Index und Daten ein Deadlock auftritt. Unter allgemeinen Bedingungen fordert INSERT die exklusiven Sperren (X-Locks) in der folgenden Reihenfolge an:

1. Blattseite des Clusterindex
2. Datenseite
3. Blattseite des Index ohne Clustering

Wenn die Anweisung SELECT den Index ohne Clustering verwendet, werden die gemeinsamen Sperren (S-Locks) in der folgenden Reihenfolge angefordert:

1. Blattseite des Index ohne Clustering
2. Datenseite

Dies ist gegenüber der Reihenfolge der INSERT-Sperren die gegensätzliche Reihenfolge. Häufig ist die Geschwindigkeit von SELECT für die neuen Zeilen höher. Dies bedeutet, dass die Anweisungen INSERT und SELECT gemeinsame Datenseiten aufweisen. Dort, wo die Indexseite auch dieselbe ist, kann ein Deadlock auftreten.

Die Lösung für das Risiko des Deadlocks liegt darin, die Zeilen zu verteilen, indem ein anderer Index als der Clusterindex ausgewählt wird.

Allgemeine Methoden für die Handhabung von Deadlock-Situationen finden Sie in [Deadlocks in der CICS-Db2-Umgebung bearbeiten](#).

CICS und die Option **CURSOR WITH HOLD**

Verwenden Sie in Ihrem CICS-Programm **CURSOR WITH HOLD** (SQL-Cursor beibehalten), um den Cursor während eines Synchronisationspunkts (**SYNCPOINT**) offen und in Position zu halten.

Die Option **WITH HOLD** für eine Cursordeklaration (**CURSOR**) in einem CICS-Programm führt während eines Synchronisationspunkts (**SYNCPOINT**) zu folgenden Auswirkungen:

- Der Cursor wird offen gehalten.
- Der Cursor bleibt nach dem Abrufen der letzten Zeile und vor der nächsten Zeile in der Ergebnistabelle in Position.
- Es werden weiterhin dynamische SQL-Anweisungen vorbereitet.

Alle Sperren werden freigegeben, außer der Sperren, die erforderlich sind, um die Position des Cursors beizubehalten. Alle exklusiven Seitensperren werden in gemeinsam genutzte Sperren herabgestuft.

In dialogorientierten CICS-Anwendungen können Sie **DECLARE CURSOR...WITH HOLD** verwenden, um anzufordern, dass der Cursor am Synchronisationspunkt nicht geschlossen wird. Am Ende einer Task (**EOT – End of Task**) oder bei **SYNCPOINT ROLLBACK** werden jedoch alle Cursor *stets* geschlossen. Über das Ende von Tasks (**EOT**) hinaus gilt, dass ein mit **WITH HOLD** deklarerter Cursor erneut geöffnet und positioniert werden muss, genauso, als sei die Option **WITH HOLD** nicht angegeben. Als Bereich des Hold-Cursors (gehaltener Cursor) gilt eine einzelne Task.

Zusammenfassung:

- Der nächste **FETCH**-Aufruf, der auf einen Synchronisationspunkt folgt, muss von derselben Task stammen.
- Sie können einen Cursor nicht über das Ende einer Task hinaus geöffnet halten.
- Cursor werden daher *nicht* über die **EOT**-Abschnitte von pseudodialogfähigen Transaktionen hinaus gehalten.

Wenn Sie versuchen, einen Cursor über **EOT** hinaus zu halten, wird der Cursor geschlossen und Sie erhalten den **SQLCODE -501**, wenn Sie den nächsten **FETCH**-Aufruf ausführen. Der Vorcompiler kann dies nicht erkennen und Sie erhalten keinen Warnhinweis, der Sie über diese Situation benachrichtigt.

Im Allgemeinen können Threads an jedem Synchronisationspunkt Kandidaten für eine erneute Verwendung werden. Wenn Sie **DECLARE CURSOR...WITH HOLD** in den CICS-Anwendungen verwenden, sollten Sie folgende Empfehlungen beachten:

- Schließen Sie die Hold-Cursor, sobald sie nicht mehr benötigt werden. Sobald alle Hold-Cursor geschlossen sind, kann der Synchronisationspunkt den Thread für die Threadwiederverwendung freigeben.
- Schließen Sie Hold-Cursor stets vor dem Ende einer Task (**EOT**). Wenn Sie Ihre gehaltenen Cursor nicht schließen, erzwingt die CICS-Db2-Anschlussfunktion eine Anmeldung, um den Thread im ursprünglichen Zustand wiederherzustellen, wodurch zusätzliche Prozessorzeit in Anspruch genommen wird.

Befehl **EXEC CICS RETURN IMMEDIATE**

Wenn die Option **TRANSID** zusammen mit der Option **IMMEDIATE** angegeben wird, vermeidet es CICS, während der Beendigung der Transaktion, die den Befehl **RETURN** ausgegeben hat, eine Endklammer (**EB – End Bracket**) an das Terminal zu senden, und ruft sofort die Transaktion auf, die durch die Option **TRANSID** angegeben ist. Die Tastatur bleibt während dieser Transaktion gesperrt, da keine Endklammer an das Terminal gesendet wurde.

Die neue Transaktion verhält sich so, als sei sie durch eine Eingabe über das Terminal gestartet worden. Sie können mithilfe eines Kommunikationsbereichs (**COMMAREA**) Daten an die Transaktion übergeben, die durch die Option **TRANSID** angegeben ist. Wenn Sie dies auswählen, kann die Transaktion, die den

Befehl RETURN ausgibt, mithilfe der Optionen INPUTMSG und INPUTMSGLEN auch eine über das Terminal eingegebene Nachricht übergeben. Mit dieser Funktion ist es möglich, sofort eine Transaktion aufzurufen, die es erwartet, als Ergebnis einer Terminaleingabe aufgerufen zu werden.

Diese Funktion stellt dieselbe allgemeine Funktionalität bereit, die auch durch Absetzen des EXEC CICS START TRANSID(...) mit der Einstellung des Werts TERMID(...) erreicht wird; der Aufwand ist jedoch wesentlich geringer und das kurze Entsperren der Tastatur ist nicht erforderlich. Der EXEC CICS-Befehl RETURN TRANSID() IMMEDIATE lässt für eine pseudodialogfähige Transaktion das Wechseln von Transaktionscodes zu. Dies kann zum Beispiel empfehlenswert sein, um die Größen von Db2-Plänen kleiner zu halten oder um über bessere Abrechnungsstatistiken für die Kostenzuordnung zu verfügen.

AEY9-Abbrüche vermeiden

AEY9-Abbrüche treten auf, wenn eine Anwendung einen EXEC-SQL-Befehl absetzt, während die CICS-Db2-Anschlussfunktion nicht aktiv ist.

Informationen zu diesem Vorgang

Sie können mithilfe des folgenden CICS-Befehls erkennen, ob die CICS-Db2-Anschlussfunktion aktiviert wurde:

```
EXEC CICS EXTRACT EXIT PROGRAM('DFHD2EX1')
ENTRY('DSNCSQL')
GASET(name1)
GALENGTH(name2)
```

Wenn Sie den Programmnamen DSNCEXT1 oder DSN2EXT1 angeben, ändert CICS diesen dynamisch in den erforderlichen Namen DFHD2EX1. Wenn Sie die Bedingung INVEXITREQ erhalten, ist die CICS-Db2-Anschlussfunktion nicht aktiviert.

Wenn die CICS-Db2-Anschlussfunktion aktiviert wurde, ist sie nicht unbedingt mit Db2 verbunden. Möglicherweise wartet sie darauf, dass Db2 initialisiert wird. Wenn dies auftritt und eine Anwendung einen EXEC-SQL-Befehl absetzt, während im DB2CONN-Objekt die Angabe CONNECTERROR=ABEND festgelegt ist, würde ein Abbruch des Typs AEY9 auftreten. CONNECTERROR=SQLCODE würde dazu führen, dass der SQL-Code -923 an die Anwendung zurückgegeben wird.

Sie können statt des Befehls EXTRACT EXIT den Befehl INQUIRE EXITPROGRAM mit dem Schlüsselwort CONNECTST verwenden, um festzustellen, ob CICS mit Db2 verbunden ist.

Das Schlüsselwort CONNECTST des Befehls INQUIRE EXITPROGRAM gibt folgende Werte zurück:

- CONNECTED, wenn die CICS-Db2-Anschlussfunktion für die Annahme von SQL-Anforderungen bereit ist
- NOTCONNECTED, wenn die CICS-Db2-Anschlussfunktion nicht für die Annahme von SQL-Anforderungen bereit ist

Wenn der Befehl mit PGMIDERR fehlschlägt, ist dies gleichbedeutend mit NOTCONNECTED.

Abbildung 28 auf Seite 99 zeigt ein Beispiel für Assembler-Code mithilfe des Befehls INQUIRE EXITPROGRAM.

```
CSTAT DS F
ENTNAME DS CL8
EXITPROG DS CL8
MVC ENTNAME,=CL8'DSNCSQL'
MVC EXITPROG,=CL8'DFHD2EX1'
EXEC CICS INQUIRE EXITPROGRAM(EXITPROG) X
ENTRYNAME(ENTNAME) CONNECTST(CSTAT) NOHANDLE
CLC EIBRESP,DFHRESP(NORMAL)
BNE NOTREADY
CLC CSTAT,DFHVALUE(CONNECTED)
BNE NOTREADY
```

Abbildung 28. Beispiel für den Befehl INQUIRE EXITPROGRAM

Wenn Sie den Programmnamen DSN2EXT1 angeben, ändert CICS diesen dynamisch in den erforderlichen Namen DFHD2EX1.

Die Verwendung der Befehle EXTRACT EXIT oder INQUIRE EXITPROGRAM durch Anwendungen muss genauer betrachtet werden, wenn die Ausführung in einer Umgebung erfolgt, in der der dynamische Lastausgleich mithilfe von z/OS Workload Manager (WLM) stattfindet.

Ein Test auf Verfügbarkeit von Db2 kann zum Storm-Drain-Effekt (Gully-Effekt) führen. Wenn eine Anwendung normal zurückgegeben wird, falls eine Verbindung zu einem Workload-Manager nicht verfügbar ist, kann dies den Workload-Manager insofern täuschen, als dieser weitere Arbeit an die CICS-Region weiterleitet, da er davon ausgeht, dass gute Antwortzeiten erreicht werden. Eine Erläuterung des Storm-Drain-Effekts finden Sie in [Storm-Drain-Effekt vermeiden](#). Führen Sie daher Folgendes aus:

- Geben Sie im DB2CONN-Objekt die Information STANDBYMODE=RECONNECT an. Dadurch wird sichergestellt, dass die CICS-Db2-Anschlussfunktion wartet (im Standby-Modus), bis Db2 initialisiert wird und automatisch eine Verbindung herstellt, sollte Db2 beim ersten Verbindungsversuch inaktiv sein. Falls Db2 nachfolgend fehlschlägt, wird die CICS-Db2-Anschlussfunktion darüber hinaus erneut in den Standby-Modus zurückgesetzt und wartet auf Db2. Anschließend wird die Verbindung automatisch hergestellt, wenn Db2 zurückgegeben wird.
- Verwenden Sie CONNECTERROR=SQLCODE, vorausgesetzt Anwendungen verarbeiten den Code -923 ordnungsgemäß.
- Vermeiden Sie, die Befehle EXTRACT EXIT oder INQUIRE EXITPROGRAM zu verwenden, wenn CONNECTERROR=SQLCODE verwendet werden kann.
- Verwenden Sie CONNECTERROR=ABEND, wenn ein Abbruch des Typs AEY9 erforderlich ist. Verwenden Sie anstelle des Befehls EXTRACT EXIT den Befehl INQUIRE EXITPROGRAM.
- Es lohnt sich zu beachten, dass AEY9-Abbrüche auch auftreten können, wenn STANDBYMODE=RECONNECT und CONNECTERROR=SQLCODE angegeben wurden, und zwar in folgenden Fällen:
 - Die CICS-Db2-Anschlussfunktion wurde nie gestartet. Eine Anwendung gibt einen EXEC-SQL-Befehl aus und das Ergebnis ist ein Abbruch des Typs AEY9. Sie sollten stets DB2CONN=YES in der Systeminitialisierungstabelle (SIT – System Initialization Table) oder das Programm DFHD2CM0 im Systeminitialisierungsparameter PLTPI angeben. Daher ist die CICS-Db2-Anschlussfunktion mindestens im Standby-Modus.
 - Die CICS-Db2-Anschlussfunktion wird mithilfe des Befehls DSNB STOP oder CEMT/EXEC CICS SET DB2CONN NOTCONNECTED beendet.

Es ist empfehlenswert, die Anschlussfunktion nicht zu beenden. Die CICS-Db2-SPI-Befehle ermöglichen die dynamische Änderung der Umgebung, ohne dass die Anschlussfunktion beendet werden muss.

Kapitel 6. Verwendung von JDBC und SQLJ für den Zugriff auf Db2-Daten aus Java-Programmen

Java-Programme, die in CICS ausgeführt werden, können mithilfe mehrerer Methoden auf die Daten zugreifen, die in einer Db2-Datenbank enthalten sind.

Java-Programme können Folgendes:

- Verwendung des JCICS-Befehls LINK, um eine Verknüpfung zu einem CICS-Programm herzustellen, das für den Zugriff auf die Daten SQL-Befehle (SQL – Structured Query Language) verwendet.
- Direkter Zugriff auf die Daten mithilfe Java Data Base Connectivity (JDBC) oder Structured Query Language für Java-Anwendungsprogrammierschnittstellen (SQLJ).

IBM Data Server Driver for JDBC and SQLJ

Wenn eine Java-Anwendung für CICS JDBC- und SQLJ-Anforderungen mit Typ 2- oder Typ 4-Konnektivität an IBM Db2- und an andere Programme für die Db2 for z/OS-Datenbank richtet, werden die Anforderungen von IBM Data Server Driver for JDBC and SQLJ verarbeitet.

In einer CICS-Umgebung, in der Typ 2-Konnektivität verwendet wird, konvertiert IBM Data Server Driver for JDBC and SQLJ die JDBC- oder SQLJ-Anforderungen in die entsprechenden EXEC-SQL-Äquivalente. Die konvertierten Anforderungen werden an die CICS-Db2-Anschlussfunktion in derselben Weise übertragen, wie EXEC-SQL-Anforderungen von Programmen, die keine Java-Programme sind. Die Anpassungs- und Optimierungsoptionen für die CICS-Db2-Anschlussfunktion gelten in gleicher Weise für Java-Programme und für Programme, bei denen es sich nicht um Java-Programme handelt.

Wenn Sie die Typ 2-Konnektivität verwenden, werden Db2-Clientinformationen wie beispielsweise die Werte für 'ClientUser' und 'Clienthostname' nicht unterstützt und daher ignoriert.

Der Liberty-JVM-Server unterstützt auch die Typ 4-Konnektivität, bei der TCP/IP zum Herstellen der Verbindung zum Db2-Subsystem statt zur CICS-Db2-Anschlussfunktion verwendet wird. Zum Verwenden von IBM Data Server Driver for JDBC and SQLJ mit CICS müssen Sie mit dem Db2-Subsystem verbunden sein, das die entsprechende Stufe Ihrer Treiberversion unterstützt.

Details zum Codieren und Erstellen von Java-Anwendungen, die für Ihre Db2-Version gültige JDBC- und SQLJ-Anwendungsprogrammierschnittstellen verwenden, finden Sie in [Java-Anwendungsentwicklung für IBM Data Server](#).

JVM-Server für die Unterstützung von Db2 konfigurieren

Ein JVM-Server ist die Laufzeitumgebung für Java-Anwendungen. Sie können den JVM-Server für die Unterstützung von JDBC- und SQLJ-basierten Anwendungen konfigurieren.

Vorbereitende Schritte

Um den JVM-Server mit Db2 verwenden zu können, ist das bewährte Verfahren, die neueste Version von IBM Data Server Driver for JDBC and SQLJ zu installieren. Sie müssen die Db2-Bibliothek SDSNLOD2 zur CICS-STEPLIB-Verkettung hinzufügen. Weitere Informationen zu erforderlichen APARs finden Sie in [Detaillierte Systemvoraussetzungen](#).

Informationen zu diesem Vorgang

Mit den folgenden Schritten aktivieren Sie Ihre Anwendungen für die Verwendung von IBM Data Server Driver for JDBC and SQLJ, das im Lieferumfang von Db2 enthalten ist.

Vorgehensweise

1. Lesen Sie die Informationen in [Liberty-JVM-Server konfigurieren](#), um den Treiber für einen Liberty-JVM-Server einzurichten. Informationen zum OSGi-JVM-Server finden Sie in [JVM-Server für OSGi-Anwendungen konfigurieren](#).
2. Die Db2-Umgebungsvariable **DB2SQLJPROPERTIES** wird in einem JVM-Server nicht unterstützt. Sie müssen daher die Eigenschaften, die sich auf den Db2-Treiber beziehen, direkt in Ihrem JVM-Profil festlegen. Eine Liste der verfügbaren Eigenschaften finden Sie in [Programmierung für Db2 for z/OS in der Produktdokumentation zu Db2 for z/OS](#).
3. Optional: Wenn Sie einen Db2-Trace für einen JVM-Server (JVMSEVER) generieren möchten, sind folgende JVM-Profileigenschaften nützlich:

```
-Ddb2.jcc.traceDirectory=/u/<benutzer-id>/db2trace/  
-Ddb2.jcc.traceFile=jccTrace.txt  
-Ddb2.jcc.appendFile=true  
-Ddb2.jcc.traceLevel=-1
```

Wichtig: Ist eine Db2-Eigenschaft im JVM-Profil festgelegt, wird sie für alle Datenquellen (dataSources>), für die diese Eigenschaft nicht anderweitig festgelegt ist, zur Standardeigenschaft.

Nächste Schritte

Wenn Sie JDBC oder SQLJ aus einer Java-Anwendung in einem OSGi-JVM-Server verwenden wollen, auf dem ein Java 2-Sicherheitsrichtlinienmechanismus aktiv ist, finden Sie Informationen dazu in [Java-Sicherheitsmanager aktivieren](#).

Db2-Schema in einem JVM-Server festlegen

Sie haben mehrere Möglichkeiten, um das aktuelle Db2-Schema für einen JVM-Server festzulegen, der auf Db2 zugreift.

Wenn das Db2-Standardschema nicht das von Ihnen gewünschte Schema ist, haben Sie zwei Optionen:

- Sie können im JVM-Profil für `db2.jcc.currentSchema` den von Ihnen gewünschten Schemanamen festlegen. Zum Beispiel stellt das Schema `-Ddb2.jcc.currentSchema=CICSDB2Setting` im JVM-Profil sicher, dass bei dem gesamten Db2-Zugriff über den JCC-Treiber das ausgewählte Schema berücksichtigt wird. Im Fall eines Liberty-JVM-Servers berücksichtigen alle `dataSource`-Definitionen diesen Wert, es sei denn, sie werden in der `dataSource`-Definition anderweitig überschrieben.
- Wenn Sie einen Liberty-JVM-Server verwenden, können Sie Schemas auch für einzelne `dataSource`-Elemente festlegen. Beispiel:

```
<dataSource id="ds1" jndiName="jdbc/ds1">  
  <jdbcDriver libraryRef="db2Lib"/>  
  <properties.db2.jcc currentSchema="TESTER"/>  
</dataSource>
```

Tipp: Wenn Sie das standardmäßige CICS-`dataSource`-Element (Datenquellenement) verwenden (erstellt mithilfe der automatischen Konfiguration), werden bei jedem Serverstart alle in `server.xml` festgelegten Werte überschrieben. Erstellen Sie Ihr eigenes `dataSource`-Element, damit durch mögliche Aktualisierungen des standardmäßigen CICS-`dataSource`-Elements nicht Ihre Werte überschrieben werden.

Programmierung für JDBC- und SQLJ-APIs

Java-Programme für CICS müssen die Programmierregeln der JDBC-Anwendungsprogrammierschnittstellen (API) einhalten, die möglicherweise enger gefasst sind, als Regeln im allgemeinen CICS-Programmiermodell.

IBM Data Server Driver for JDBC and SQLJ bietet Unterstützung für JDBC 4.2 und frühere Versionen.

Beim Herstellen einer Verbindung zwischen einem OSGi-Java-Programm und Db2 müssen Sie die Angabe `com.ibm.db2.jcc;resolution:=optional` zum Importpaket in Ihrem Bundlemanifest hinzufügen.

Anmerkung: Die Verwendung von SQLJ mit Typ 2-Konnektivität wird in einem Liberty-JVM-Server unterstützt. Die Verwendung von SQLJ mit Typ 4-Konnektivität wird nicht unterstützt.

Weitere Informationen zu den JDBC-APIs finden Sie auf der [Oracle-JDBC-Website](#). Informationen zur Entwicklung von Java-Anwendungen, die JDBC- und SQLJ-APIs verwenden, finden Sie in [Java-Anwendungsentwicklung für IBM Data Server in der Produktdokumentation zu Db2 for z/OS](#).

Serialisiertes SQLJ-Profil in einem JVM-Server bereitstellen

SQLJ bietet Unterstützung für eingebettetes statisches SQL in Java-Anwendungen. Sie stellen ein serialisiertes SQLJ-Profil in einem OSGi-JVM-Server oder in einem Liberty-JVM-Server als Alternative zur Verwendung der Programmierschnittstelle der JDBC-Anwendung bereit.

Vorbereitende Schritte

Sie müssen Ihr SQLJ-Programm mithilfe des folgenden Prozesses vorbereiten: [Programmerstellung für SQLJ-Programme](#).

Sobald die Datenbankanforderungsmodule erstellt wurden, müssen Sie in einen Db2-Plan oder ein Db2-Paket und einen Db2-Eintrag gebunden werden, der den Db2-Plan angibt, der für CICS definiert ist. Wird dies nicht ausgeführt, führt es höchstwahrscheinlich zu SQL-Fehlern des Typs -805 oder dazu, dass SQLJ standardmäßig JDBC verwendet. Laden Sie die Datei mit dem serialisierten Profil in die Workstation herunter, in der das Bundle erstellt wird. Stellen Sie sicher, dass die Übertragung mithilfe des Binärformats erfolgt, sodass keine Codepagekonvertierung ausgeführt wird. Dadurch werden spätere Probleme vermieden.

Es muss nun entschieden werden, an welcher Position die Datei mit dem serialisierten Profil gespeichert wird. Es gibt zwei Szenarios:

Vorgehensweise

1. Szenario eins – das serialisierte Profil verbleibt im bereitgestellten Bundle.
 - a) Bei OSGi-Bundles kann das serialisierte Profil im Stammverzeichnis des Bundles oder im Klassenverzeichnis des Bundles verbleiben. Bei Liberty-Bundles muss es zusammen mit den übrigen Klassen in das Verzeichnis 'bin' gestellt werden.
2. Szenario zwei – das Profil wird extern bereitgestellt.
 - a) Verschieben Sie das serialisierte Profil aus dem JAR-Verzeichnis in ein Verzeichnis im USS-Dateisystem, zum Beispiel: /usr/lpp/cicsts/dev/sqlj.profile.dir. Die Datei muss sich in einem Verzeichnis befinden, das dieselbe Paketstruktur aufweist, zum Beispiel /usr/lpp/cicsts/dev/sqlj.profile.dir/com/ibm/cics/test/sqlj/CurrentTimeStamp_SJProfile0.ser.
 - b) Fügen Sie dem Bundlemanifest einen Eintrag hinzu.

```
Bundle-ClassPath: .,external:$sqlj.profile.dir$
```

Die Zeichenfolge 'external' gibt eine für das Bundle externe Position an und die in \$ eingeschlossene Zeichenfolge ersetzt den Wert einer Java-Systemeigenschaft, das heißt sqlj.profile.dir.

- c) Fügen Sie im JVM-Profil eine Java-Systemeigenschaft hinzu, die dem folgenden Beispiel ähnelt:

```
-Dsqlj.profile.dir=/usr/lpp/cicsts/dev/sqlj.profile.dir
```

Ergebnisse

Sie haben erfolgreich ein serialisiertes SQLJ-Profil bereitgestellt.

Verbindung zu einer Datenbank anfordern

Bevor SQL-Anweisungen ausgeführt werden können, muss eine JDBC- oder SQLJ-Anwendung eine Verbindung zu einer Datenbank anfordern. Die Anwendung stellt eine Verbindung zu einer Zieldatenquelle mithilfe einer Java-Schnittstelle her, wobei zwei zur Auswahl stehen.

Informationen zu diesem Vorgang

Die zwei Java-Schnittstellen lauten wie folgt:

- **DriverManager:** Diese Klasse verbindet eine Anwendung mit einer Datenbank, die durch eine Datenbank-URL angegeben wird.
- **DataSource:** Diese Schnittstelle wird DriverManager vorgezogen, da sie zulässt, dass Details zur zugrunde liegenden Datenbank für Ihre Anwendung transparent sind; Anwendungen sind dadurch besser portierbar. Die DataSource-Implementierung wird nicht im Java-Programm, sondern extern erstellt und mit der JNDI-Suche (JNDI – Java Naming and Directory Interface) nach DataSource-Namen abgerufen. Die Schnittstelle 'DataSource' ist nur in Liberty-JVM-Servern verfügbar.

In OSGi-JVM-Servern wird nur die Konnektivität des Typs 2 mithilfe von DriverManager für die Verwendung mit Db2 unterstützt.

In Liberty-JVM-Servern können Sie die Schnittstelle 'DriverManager' oder 'DataSource' verwenden, um auf Db2 zuzugreifen.

Wenn Sie die JDBC-Konnektivität des Typs 2 verwenden, müssen Sie Benutzer-ID und zugehöriges Kennwort nicht angeben, da stattdessen die CICS-Db2-Sicherheitsverfahren angewendet werden. Es wird außerdem empfohlen, eine Standard-URL zu verwenden; weitere Informationen finden Sie in [Arbeitseinheit festschreiben](#).

Es gibt sowohl für OSGi- als auch für Liberty-JVM-Server JDBC-Beispiele, die in IBM CICS SDK for Java bereitgestellt sind. Weitere Informationen hierzu finden Sie in [Java-Anwendungen für die Ausführung auf einem Liberty-JVM-Server entwickeln](#).

Weitere Informationen dazu, wie die JDBC-Schnittstellen 'DriverManager' und 'DataSource' zum Anfordern einer Verbindung verwendet werden, sowie Beispielcode, den Sie in Ihrer Anwendung verwenden können, finden Sie in der für Ihre Db2-Version passenden Dokumentation von 'Db2 for z/OS: Programming for Java'.

DriverManager-Verbindung zu einer Datenbank anfordern

Die Verbindung zur Datenbank wird mithilfe der Klasse DriverManager über eine Datenbank-URL (URL – Uniform Resource Locator) angegeben, die für den JDBC-Treiber bereitgestellt wird.

Informationen zu diesem Vorgang

Eine DriverManager-Verbindung wird für einen OSGi-JVM-Server durch Angeben der Db2-JDBC-JAR-Dateien und der nativen DLL-Dateien im JVM-Profil OSGI_BUNDLES und im JVM-Profil LIBPATH_SUFFIX konfiguriert.

In einem Liberty-JVM-Server wird die DriverManager-Konfiguration durch das CICS-JDBC-Liberty-Feature bereitgestellt.

Der JDBC-Treiber erkennt zwei Typen von URLs:

Standard-URL

Eine Standard-URL enthält nicht den Verzeichnisnamen eines Db2-Subsystems. Eine Standard-URL für Db2 for z/OS kann in einem der folgenden Formate angegeben werden:

```
jdbc:db2os390sqlj:  
Alternative:  
jdbc:default:connection
```

Bei Angabe einer Standard-URL wird für die Anwendung eine Verbindung zur lokalen Db2-Instanz hergestellt, mit der CICS verbunden ist. Wenn Ihre Installation für Db2 die gemeinsame Datennutzung verwendet, können Sie auf alle Daten in Ihrem Sysplex von der lokalen Db2-Instanz aus zugreifen.

Explizite URL

Eine explizite URL enthält den Verzeichnisnamen eines Db2-Subsystems. Die Basisstruktur einer expliziten URL für Db2 for z/OS lautet:

```
jdbc:db2os390:<verzeichnisname>  
Alternative:  
jdbc:db2os390sqlj:<verzeichnisname>
```

In der Regel ist der Verzeichnisname der Name der lokalen Db2-Instanz, mit der CICS verbunden ist. Sie können jedoch den Namen einer fernen Db2-Instanz angeben, auf die zugegriffen werden soll. In diesem Fall verwendet CICS die lokale Db2-Instanz als Durchgriff und verwendet Db2 Distributed Data Facility, um auf die ferne Db2-Instanz zuzugreifen.

Es ist empfehlenswert, für den Zugriff auf Db2 mit Typ-2-Konnektivität die Standard-URL zu verwenden. Die Verwendung einer expliziten URL kann beim Schließen einer Verbindung zu bestimmten Aktionen führen, die ungünstig sein können, wenn in derselben Anwendungssuite mehrere Programme verwendet werden. Außerdem ist das Verhalten der Verbindung bei Verwendung einer Standard-URL nicht von der JDBC-Version betroffen.

Zum Anfordern einer Verbindung muss Ihre Java-Anwendung die Methode `getConnection()` mit der URL aufrufen. Beispiel:

```
Connection connection =  
DriverManager.getConnection("jdbc:default:connection");
```

Beim Herstellen einer Verbindung zwischen einem OSGi-Java-Programm und Db2 müssen Sie die Angabe `com.ibm.db2.jcc;resolution:=optional` zum Importpaket in Ihrem Bundlemanifest hinzufügen.

DataSource-Verbindung zu einer Datenbank anfordern

Die Verbindung zur Datenbank wird über die DataSource-Schnittstelle mithilfe der JNDI-Suche (JNDI – Java Naming and Directory Interface) nach DataSource-Namen abgerufen. Die JDBC-DataSource-Implementierung wird entweder über das Feature `cicsts:jdbc-1.0` oder das Liberty-Feature `jdbc-4.1` oder `jdbc-4.2` bereitgestellt.

Informationen zu diesem Vorgang

Der Begriff 'Datenquelle' (data source) kann in zwei verschiedenen Kontexten verwendet werden; es ist Sorgfalt erforderlich, da die Unterscheidung über die Großschreibung erfolgt:

- Eine Datenquelle ist eine Quelle mit Daten wie beispielsweise eine Datenbank.
- Ein DataSource-Element ist die in Java EE verwendete Art und Weise, einen Satz von Eigenschaften zusammenzufassen, die die echte Datenquelle, die durch sie dargestellt wird, anzugeben und zu beschreiben. Ein DataSource-Objekt kann als Factory für Verbindungen zu der bestimmten Datenquelle bezeichnet werden, die durch das Objekt dargestellt wird.
- Wenn ein DataSource-Objekt mit einem JNDI-Namensservice registriert wird, kann eine Anwendung die JNDI-API für den Zugriff auf dieses DataSource-Objekt verwenden, welches dann zum Herstellen der Verbindung zu der Datenquelle verwendet werden kann, die es darstellt.
- Ein DataSource-Objekt kann auswählen, einen Verbindungspool (eine Gruppe von temporären logischen Darstellungen einer physischen Verbindung) zu implementieren. Wenn die Anwendung eine solche Verbindung schließt, wird die temporäre Verbindung (auch als JDBC-Verbindung bezeichnet) nicht geschlossen, sondern zur Wiederverwendung an den Pool zurückgegeben.

Wenn Sie JDBC mit Typ 4-Konnektivität verwenden und Datenbankaktualisierungen mit den Aktualisierungen koordinieren wollen, die in der CICS-Arbeitseinheit ausgeführt wurden, müssen Sie die CICS-JTA-Integrationsunterstützung (JTA – Java Transaction API) verwenden und Ihre Aktualisierungen innerhalb des Bereichs einer Benutzertransaktion (UserTransaction) ausführen. Weitere Informationen finden Sie in [Java Transaction API \(JTA\)](#). Sie sollten zusätzlich sicherstellen, dass Ihr DataSource-Element ein XADataSource-Element ist (dies wird durch Hinzufügen des Elements `type="javax.sql.XADataSource"` zu

Ihrer Definition angegeben). Falls Sie das standardmäßige Liberty-DataSource-Element <dataSource id="DefaultDataSource"> verwenden: dies ist standardmäßig ein XADataSource-Element.

Nach dem Konfigurieren des DataSource-Elements kann die Anwendung den Wert von jndiName verwenden, der im Element cicsts_dataSource bzw. dataSource in der Liberty-Server-Konfiguration angegeben ist, um eine Instanz dieser DataSource-Klasse vom JNDI-Namensservice abzurufen. Für dieses DataSource-Objekt kann dann die Methode getConnection() aufgerufen werden, um eine Verbindung anzufordern. Beispiel:

```
Context context = new InitialContext();
DataSource dataSource = (DataSource)
context.lookup("jdbc/defaultCICSDataSource");

Connection connection = dataSource.getConnection();
```

Überlegungen zu JDBC- und SQLJ-Verbindungen

Bei Verwendung von IBM Data Server Driver for JDBC and SQLJ in CICS-Java-Anwendungen, sollten folgende Informationen berücksichtigt werden.

IBM Data Server Driver for JDBC and SQLJ mit Typ 2-Konnektivität verwenden

Wenn Sie Typ 2-Konnektivität verwenden, kann in einer Java-Anwendung für CICS gleichzeitig nur eine einzige JDBC- bzw. eine einzige SQLJ-Verbindung geöffnet sein.

Obwohl JDBC es ermöglicht, dass eine Anwendung gleichzeitig über mehrere Verbindungen verfügt, wird dies durch die Verwendung der Typ 2-Konnektivität auf eine einzige Verbindung eingeschränkt. Eine Anwendung kann jedoch eine vorhandene Verbindung schließen und eine Verbindung zu einer neuen Db2-Position öffnen, falls erforderlich.

Als Richtlinie sollte eine Anwendung mit einer offenen Transaktion die Verbindung schließen, bevor eine Verbindung zu einer anderen Anwendung hergestellt wird, die möglicherweise JDBC oder SQLJ verwendet. Für Java-Programme, die Teil einer Anwendungssuite sind, müssen Sie die Auswirkungen beachten, die durch das Schließen der Verbindung auftreten, da das Schließen der Verbindung bei Verwendung einer expliziten URL dazu führt, dass ein Synchronisationspunkt erfasst wird. Wenn Sie eine Standard-URL verwenden, muss beim Schließen der Verbindung kein Synchronisationspunkt erfasst werden.

In „[Arbeitseinheit festschreiben](#)“ auf Seite 107 finden Sie weitere Informationen zur Typ 2-Konnektivität.

IBM Data Server Driver for JDBC and SQLJ mit Typ 4-Konnektivität verwenden (nur Liberty-JVM-Server)

Wenn Sie Typ 4-Konnektivität verwenden, können in einer Java-Anwendung für CICS gleichzeitig mehrere JDBC-Verbindungen oder mehrere SQLJ-Verbindungen geöffnet sein. Es sollte darauf geachtet werden, dass die Datenintegrität beibehalten wird. Es ist empfehlenswert, Java Transaction API zu verwenden, um die Transaktionsgrenzen klar zu markieren.

Anmerkung:

Die Implementierungen JDBC 4.0, 4.1 und 4.2 befinden sich im selben Treiber.

Wenn Sie die Features jdbc-4.1 oder jdbc-4.2 verwenden, versuchen die Schnittstelle **RowSetFactory** und die Klasse **RowSetProvider** eine neue JDBC-Verbindung zu erstellen.

Verbindung zu einer Datenbank schließen

Wenn Sie eine Verbindung zu einer Datenbank schließen, werden JDBC- und SQLJ-Ressourcen automatisch freigegeben. In der Regel wird eine Verbindung zu einer Datenbank geschlossen, wenn die Task endet.

Aus Gründen der Leistung hält eine Anwendung die JDBC-Verbindung möglicherweise offen, sodass ein nachfolgender Benutzer derselben JVM diese Verbindung verwenden kann. Wenn die JDBC-Verbindung offen gehalten wird, muss die Anwendung sicherstellen, dass JDBC-Ressourcen im Zeitverlauf nicht ein Leck bekommen.

Wenn die Anwendung die JDBC- oder SQLJ-Verbindung offen hält, muss die Anwendung Folgendes ausführen:

- Sicherstellen, dass JDBC- und SQLJ-Ressourcen freigegeben sind.
- Nach einer Db2-Anmeldung (SIGNON) eine Wiederherstellung für die zugrunde liegende Db2-Verbindung durchführen.
- Die zwischengespeicherte Verbindung erneut starten, wenn sie ungültig geworden ist; Beispiel: Stale-Connection, SQLCODE=4499.

Wenn Verbindungen zwischengespeichert werden, wird empfohlen, in Ihre Anwendung Logik einzuschließen, mit der die Verbindung nach einer bestimmten Anzahl von Transaktionen erneut gestartet wird. Das Stoppen und erneute Starten von zwischengespeicherten Verbindungen schützt vor Ressourcenlecks.

Arbeitseinheit festschreiben

Wenn Sie die Typ 2-Konnektivität verwenden, können Ihre JDBC- und SQLJ-Anwendungen JDBC- und SQLJ-Methodenaufrufe für die Festschreibung und die Wiederherstellung (Rollback) ausgeben. IBM Data Server Driver for JDBC and SQLJ konvertiert diese Aufrufe in einen JCICS-Commitaufruf oder einen JCICS-Rollbackaufruf, sodass ein CICS-Synchronisationspunkt erfasst wird.

Informationen zu diesem Vorgang

Eine JDBC- oder SQLJ-Festschreibung führt dazu, dass die gesamte CICS-Arbeitseinheit festgeschrieben wird, nicht nur die Aktualisierungen, die an Db2 vorgenommen wurden. CICS unterstützt nicht die Festschreibung geänderter Daten mithilfe einer JDBC-Verbindung unabhängig von der übrigen CICS-Arbeitseinheit.

Eine JDBC- oder SQLJ-Anwendung kann auch JCICS-Commits oder -Rollbacks direkt ausgeben; dies führt zu demselben Ergebnis, wie das Ausgeben von JDBC- und SQLJ-Methodenaufrufen für ein Commit oder einen Rollback. Die gesamte Arbeitseinheit wird gemeinsam festgeschrieben oder rückgängig gemacht und zwar sowohl Db2-Aktualisierungen als auch Aktualisierungen an gesteuerten CICS-Ressourcen.

Bei der Arbeit mit JDBC-Verbindungen gibt es einige Umstände, bei denen Sie nicht vermeiden können, dass bei Schließen der Verbindung zur Db2-Datenbank ein Synchronisationspunkt erfasst und die Arbeitseinheit festgeschrieben wird. Dies gilt, wenn einer der folgenden Umstände zutrifft:

- Sie haben die Eigenschaft für automatische Festschreibung (autocommit) einer JDBC-Verbindung verwendet.
- Sie haben die DriverManager-Verbindung mithilfe einer expliziten URL angefordert.

Bei einer eigenständigen Anwendung stellen diese Regeln kein Problem dar, da CICS sicherstellt, dass zusätzlich zu den Synchronisationspunkten, die beim Schließen der Verbindung erfasst werden, auch ein Synchronisationspunkt am Ende der Task erfasst wird. Die JDBC- und SQLJ-Anwendungsprogrammierschnittstellen unterstützen jedoch nicht das Konzept, bei dem für eine einzelne Arbeitseinheit mehrere Anwendungsprogramme verwendet werden. Wenn Sie über eine Reihe von Programmen verfügen, die zusammen eine Anwendung bilden, kann ein Programm im Verlauf einer einzigen Arbeitseinheit auf Db2 zugreifen und dann ein anderes Programm aufrufen, das auch auf Db2 zugreift. Wenn Sie möchten, dass diese Programme Java-Programme sind, die JDBC- oder SQLJ verwenden, müssen Sie sicherstellen, dass die Arbeitseinheit beim Schließen der Verbindung zu Db2 nicht festgeschrieben wird; andernfalls funktioniert die Anwendung nicht wie geplant. Sie sollten diese Voraussetzung insbesondere dann beachten, wenn Sie in einer bestehenden Anwendung Programme durch Java-Programme ersetzen, die JDBC oder SQLJ verwenden, und Sie zwischen diesen Programmen denselben CICS-Db2-Thread gemeinsam nutzen wollen. Verwenden Sie zum Beheben dieses Problems DriverManager mit der Standard-URL oder eine DataSource-Verbindung.

Automatische Festschreibung

JDBC-Anwendungen können für eine JDBC-Verbindung die Eigenschaft für die automatische Festschreibung (Autocommit) verwenden. Die Eigenschaft für die automatische Festschreibung verursacht nach jeder Aktualisierung von Db2 eine Festschreibung. Wenn die Typ 2-Konnektivität verwendet wird, ist diese

Festschreibung eine CICS®-Festschreibung und führt dazu, dass als Ergebnis die gesamte Arbeitseinheit festgeschrieben wird.

Bei Verwendung der Eigenschaft für die automatische Festschreibung wird auch beim Schließen einer Verbindung eine Festschreibung ausgeführt. Die Verwendung der automatischen Festschreibung in Verbindung mit der Typ 2-Konnektivität wird nicht empfohlen. Aus diesem Grund legt IBM Data Server Driver for JDBC and SQLJ als Standard `autocommit(false)` fest, wenn es in einer CICS-Umgebung mit Typ 2-Konnektivität ausgeführt wird. Bei Typ 4-Konnektivität lautet die Standardeinstellung `autocommit(true)`.

DriverManager-Probleme mit dem Synchronisationspunkt bei Verwendung expliziter und Standard-URLs

Wenn eine Java-Anwendung für CICS, die JDBC oder SQLJ verwendet, mithilfe einer expliziten URL eine Verbindung anfordert, wird sie in einer Umgebung ausgeführt, die der eines DPL-Serverprogramms ähnelt, das mit dem Attribut `SYNCONRETURN` verknüpft ist.

Wenn ein Anwendungsprogramm, das JDBC oder SQLJ verwendet, die über die explizite URL hergestellte Verbindung schließt und CICS dabei IBM Data Server Driver for JDBC and SQLJ verwendet, wird kein impliziter Synchronisationspunkt erfasst.

Das Schließen einer über eine explizite URL hergestellten Verbindung ist jedoch nur an einer UOW-Grenze erfolgreich (UOW – Unit of Work, Arbeitseinheit). Die Anwendung muss deshalb vor dem Schließen der Verbindung einen Synchronisationspunkt erfassen, indem ein JDBC- oder SQLJ-Methodenaufruf für eine Festschreibung oder eine JCICS-Festschreibung ausgegeben wird. (Die Anwendung könnte `'autocommit(true)'` verwenden, um sicherzustellen, dass ein Synchronisationspunkt erfasst wird, aber von der Verwendung dieser Eigenschaft in der CICS-Umgebung wird abgeraten.) Wenn das Anwendungsprogramm eine über eine explizite URL hergestellte Verbindung schließt, bezeichnet dies auch das Ende der Arbeitseinheit.

Sie können diese Einschränkung umgehen, indem Sie die Verbindung mithilfe einer **Standard-URL** statt mithilfe einer expliziten URL anfordern oder indem Sie eine Datenquelle verwenden, die eine über eine Standard-URL hergestellte Verbindung bereitstellt (siehe „Verbindung zu einer Datenbank anfordern“ auf Seite 103). Wenn eine Standard-URL verwendet wird, muss die Java-Anwendung die Verbindung nicht an einer UOW-Grenze schließen und es wird beim Schließen der Verbindung kein Synchronisationspunkt erfasst (vorausgesetzt, `'autocommit(true)'` wurde nicht angegeben).

Es wird empfohlen, dass Sie stets über eine Standard-URL hergestellte Verbindungen verwenden, wenn DriverManager mit Typ 2-Konnektivität verwendet wird.

CICS-Abbrüche während JDBC- oder SQLJ-Anforderungen

CICS-Abbrüche, die während der Verarbeitung einer von IBM Data Server Driver for JDBC and SQLJ erstellten EXEC-SQL-Anforderung ausgegeben werden, werden nicht in Java-Ausnahmebedingungen konvertiert und können daher von einer für CICS ausgeführten Java-Anwendung nicht erkannt werden. Die CICS-Transaktion wird abnormal beendet und es wird ein Rollback auf den zuletzt gesetzten Synchronisationspunkt ausgeführt.

Kapitel 7. CICS-Db2-Programme für Ausführung und Produktion vorbereiten

In diesem Abschnitt wird die Programmerstellung in einer CICS-Db2-Umgebung beschrieben.

Informationen zur Unterstützung für Java-Programme in der CICS-Db2-Umgebung finden Sie in [Verwendung von JDBC und SQLJ für den Zugriff auf Db2-Daten aus Java-Programmen](#).

Diese Themen enthalten Informationen zu Diagnose, Modifikation und Optimierung.

CICS-Db2-Testumgebung

Überlegen Sie beim Einrichten Ihrer CICS-Db2-Testumgebung, wie viele CICS-Systeme Sie mit Ihren Db2-Systemen verbinden wollen.

Sie können mehrere CICS-Systeme mit demselben Db2-System verbinden. Die CICS-Db2-Anschlussfunktion ermöglicht Ihnen jedoch nicht, ein CICS-System gleichzeitig mit mehreren Db2-Systemen zu verbinden.

Sie haben zum Einrichten der Produktions- und der Testumgebung folgende Möglichkeiten:

- Ein einziges CICS-System, das mit einem einzigen Db2-System verbunden ist
- Mindestens zwei CICS-Systeme für die Produktion und den Test, die mit demselben Db2-System verbunden sind
- Mindestens zwei CICS-Systeme, die mit mindestens zwei unterschiedlichen Db2-Systemen verbunden sind

Die erste Alternative, bei der ein einziges CICS-System sowohl für die Produktion als auch den Test verwendet wird, ist nicht zu empfehlen, da Anwendungen im Teststatus die Leistung des Produktionssystems beeinflussen könnten.

Die zweite Alternative, bei der nur ein einziges Db2-System verwendet wird, könnte sowohl für Test als auch Produktion verwendet werden. Ob dies geeignet ist, hängt von den beteiligten Entwicklungs- und Produktionsumgebungen ab. Werden ein CICS-Testsystem und ein CICS-Produktionssystem separat ausgeführt, haben Testfehler keinen Einfluss auf die Produktion.

Die dritte Alternative mit beispielsweise einem Db2-System für Tests und einem solchen System für die Produktion bietet die meiste Flexibilität. Zwei CICS-Subsysteme können mit einem oder mehreren Db2-System ausgeführt werden. Folgendes gilt, wenn die CICS-Systeme an unterschiedliche Db2-Systeme angeschlossen sind:

- Benutzerdaten und der Db2-Katalog werden nicht gemeinsam genutzt. Dies ist von Vorteil, wenn Sie Testdaten von Produktionsdaten getrennt halten möchten.
- Ein falsches Design oder Programmfehler in getesteten Anwendungen wirken sich nicht auf die Leistung des Produktionssystems aus.
- Die Berechtigung innerhalb des Testsystems kann weniger strikt gehandhabt werden, da keine Produktionsdaten verfügbar sind. Wenn zwei CICS-Systeme mit demselben Db2-System verbunden sind, muss die Berechtigung strikt gesteuert werden, sowohl hinsichtlich der Funktionen als auch der Daten, die Programmierern zur Verfügung stehen.

CICS-Db2-Programm Vorbereitung

Sie können Ihr CICS-Db2-Programm mit der Schnittstelle Db2 Interactive (DB2I) erstellen oder Sie können eigene JCL zur Stapelverarbeitung übergeben.

Informationen zu diesem Vorgang

In den in Abbildung 29 auf Seite 110 gezeigten Schritten ist zusammengefasst, wie Ihr Programm für die Ausführung vorbereitet wird, nachdem der Entwurf und die Codierung Ihres Anwendungsprogramms abgeschlossen sind.

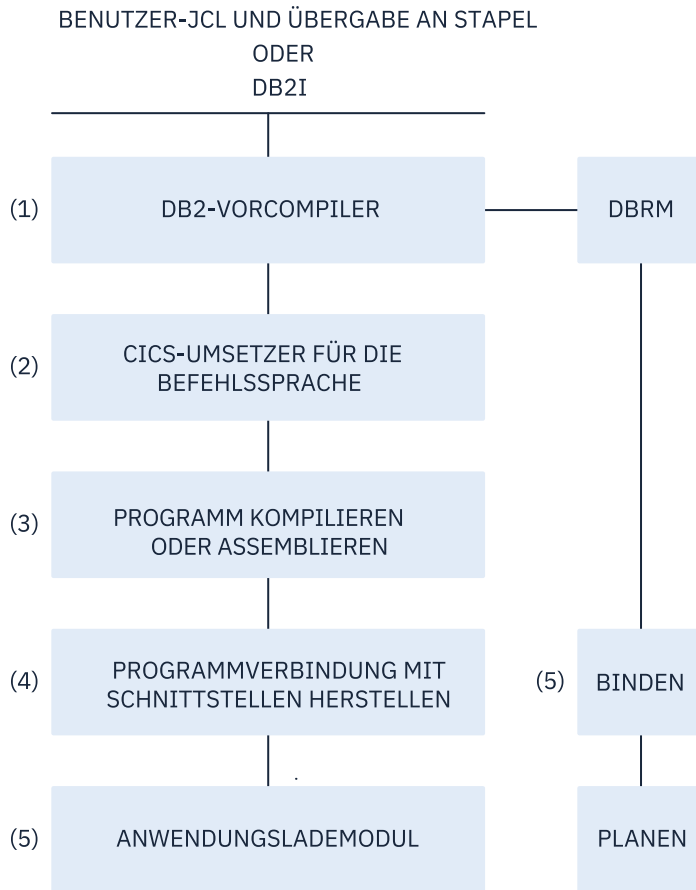


Abbildung 29. Schritte für die Vorbereitung eines CICS-Anwendungsprogramms mit Zugriff auf Db2

Eine Übersicht der einzelnen Phasen dieses Prozesses finden Sie in [CICS-Anwendungsprogramm mit Zugriff auf Db2 vorbereiten](#).

Beachten Sie Folgendes, wenn Sie CICS-Anwendungsprogramme erstellen, die auf Db2 zugreifen:

- Der Db2-Vorcompiler (Schritt 1) erstellt ein Datenbankanforderungsmodul, das Informationen zu den einzelnen SQL-Anweisungen des Programms enthält. Dieses Element überprüft auch SQL-Anweisungen im Programm. Weitere Informationen zur Verwendung des Db2-Vorcompilers finden Sie in [Programmierung für Db2 for z/OS](#) in der Produktdokumentation zu Db2 for z/OS.
- Wenn das Quellenprogramm in PL/I geschrieben wurde, ist die Eingabe für Schritt 1 (den Db2-Vorcompiler) die Ausgabe der PL/I-Makrophase (falls verwendet).
- Sie können Schritt 1 (den Db2-Vorcompiler) und Schritt 2 (den CICS-Umsetzer für die Befehlssprache) in beliebiger Reihenfolge verwenden. Die gezeigte Reihenfolge ist die bevorzugte Methode und dies ist auch die Methode, die von den DB2I-Anzeigen für die Programmerstellung unterstützt wird. Wenn Sie zuerst den CICS-Umsetzer für die Befehlssprache ausführen, generiert er bei jeder auftretenden SQL-Anweisung einen Warnhinweis; diese Nachrichten wirken sich jedoch nicht auf das Ergebnis aus.
- Wenn Sie einen der Language Environment-konformen Compiler (COBOL und PL/I) verwenden, die einen integrierten CICS-Umsetzer aufweisen, findet die Umsetzung der EXEC CICS-Befehle (Schritt 2) während der Programmkompilierung statt (Schritt 3). Weitere Informationen zum integrierten CICS-Umsetzer und zu den Compilern, die den Umsetzer unterstützen, finden Sie in [Umsetzung und Kompilierung](#).

- Wenn Sie DB2 Version 7 oder höher ausführen und ein COBOL- oder PL/I-Programm mithilfe einer der Language Environment-konformen COBOL- oder PL/I-Compiler erstellen, stellt der Compiler auch einen Koprozessor für SQL-Anweisungen bereit (der ein Datenbankanforderungsmodul generiert), sodass Sie nicht den gesonderten Db2-Vorcompiler (Schritt 1) verwenden müssen. Weitere Informationen zur Verwendung des Koprozessors für SQL-Anweisungen finden Sie in Programmierung für Db2 for z/OS in der Produktdokumentation zu Db2 for z/OS.
- Wenn Sie DB2 Version 6 oder früher ausführen und ein COBOL- oder PL/I-Programm erstellen, müssen Sie den gesonderten Db2-Vorcompiler verwenden. Stellen Sie bei einem COBOL-Programm sicher, dass Sie einen Zeichenfolgebegrenzer angeben, der für den Db2-Vorcompiler und den integrierten CICS-Umsetzer gleich ist. Die Standardbegrenzer sind nicht kompatibel.
- Schließen Sie beim Herstellen der Programmverbindungen (Schritt 4) sowohl das entsprechende CICS-EXEC-Schnittstellenmodul oder den Stub für die Sprache ein, in der Sie codieren, als auch das CICS-Db2-Sprachschnittstellenmodul DSNCLI. Das CICS-EXEC-Schnittstellenmodul *muss* zuerst in das Lademodul eingeschlossen werden. Sie können DSNCLI entweder mithilfe des 24-Bit- oder des 31-Bit-Adressierungsmodus (AMODE=31) mit Ihrem Programm verbinden. Wenn Ihr Anwendungsprogramm im 31-Bit-Adressierungsmodus ausgeführt wird, sollten Sie für den DSNCLI-Stub Programmverbindungen zu Ihrer Anwendung mithilfe der Attribute AMODE=31 und RMODE=ANY herstellen, sodass Ihre Anwendung mit mehr als 16 MB ausgeführt werden kann.
- Für den Bindeprozess (Schritt 5) ist Db2 erforderlich. Im Bindeprozess wird das Datenbankanforderungsmodul zum Generieren eines Anwendungsplans (häufig nur als 'Plan' bezeichnet) verwendet, der dem Programm den Zugriff auf Db2-Daten ermöglicht. Weitere Informationen zum Bindeprozess finden Sie in Bindeprozess. Eine Gruppe von Transaktionen, die denselben Einstiegsthread verwenden (die also in demselben DB2ENTRY-Objekt angegeben sind), muss auch denselben Anwendungsplan verwenden. Deren Datenbankanforderungsmodule müssen in denselben Anwendungsplan gebunden oder in Pakete gebunden werden, die dann in demselben Anwendungsplan aufgelistet werden.

In Tabelle 8 auf Seite 111 werden die Tasks gezeigt, die Sie zum Erstellen eines CICS-Db2-Programms ausführen müssen; dies ist abhängig von der Programmsprache und Ihrer Version von Db2:

<i>Tabelle 8. Tasks zum Erstellen eines CICS-Programms, das auf Db2 zugreift</i>					
Version von Db2 und Programm-sprache	Schritt 1 (Verarbeitung von SQL-Anweisungen)	Schritt 2 (Umsetzung von CICS-Befehlen)	Schritt 3 (Programmkompileierung)	Schritt 4 (Programmverbindungen erstellen)	Schritt 5 (Binden)
DB2 Version 6 und Assembler	Db2-Vorcompiler	Von CICS bereitgestellter separater Umsetzer	Sprachcompiler	Programmverbindungen zur EXEC-Schnittstelle und zur DSNCLI herstellen	Bindeprozess
DB2 Version 6 und PL/I	Db2-Vorcompiler	Sprachcompiler, der den integrierten CICS-Umsetzer unterstützt		Programmverbindungen zur EXEC-Schnittstelle und zur DSNCLI herstellen	Bindeprozess
DB2 Version 6 und COBOL	Db2-Vorcompiler	Sprachcompiler, der den integrierten CICS-Umsetzer unterstützt		Programmverbindungen zur EXEC-Schnittstelle und zur DSNCLI herstellen	Bindeprozess

Tabelle 8. Tasks zum Erstellen eines CICS-Programms, das auf Db2 zugreift (Forts.)

Version von Db2 und Programm-sprache	Schritt 1 (Verarbeitung von SQL-Anweisungen)	Schritt 2 (Umsetzung von CICS-Befehlen)	Schritt 3 (Programmkompileierung)	Schritt 4 (Programmverbindungen erstellen)	Schritt 5 (Binden)
DB2 Version 6 und andere Sprachen	Db2-Vorcompiler	Von CICS bereitgestellter separater Umsetzer	Sprachcompiler	Programmverbindungen zur EXEC-Schnittstelle und zur DSNCLI herstellen	Bindeprozess
DB2 Version 7 (oder höher) und Assembler	Db2-Vorcompiler	Von CICS bereitgestellter separater Umsetzer	Sprachcompiler	Programmverbindungen zur EXEC-Schnittstelle und zur DSNCLI herstellen	Bindeprozess
DB2 Version 7 (oder höher) und PL/I	Sprachcompiler, der den integrierten CICS-Umsetzer und den Koprozessor für SQL-Anweisungen unterstützt			Programmverbindungen zur EXEC-Schnittstelle und zur DSNCLI herstellen	Bindeprozess
DB2 Version 7 (oder höher) und COBOL	Sprachcompiler, der den integrierten CICS-Umsetzer und den Koprozessor für SQL-Anweisungen unterstützt			Programmverbindungen zur EXEC-Schnittstelle und zur DSNCLI herstellen	Bindeprozess
DB2 Version 7 (oder höher) und andere Sprachen	Db2-Vorcompiler	Von CICS bereitgestellter separater Umsetzer	Sprachcompiler	Programmverbindungen zur EXEC-Schnittstelle und zur DSNCLI herstellen	Bindeprozess

Sie erstellen Ihr CICS-Db2-Programm mithilfe der Schnittstelle Db2 Interactive (DB2I) oder indem Sie Ihre eigene JCL zur Stapelverarbeitung übergeben.

- Schnittstelle Db2 Interactive (DB2I): DB2I bietet Anzeigen, mithilfe derer Sie ein Anwendungsprogramm vorkompilieren, kompilieren oder assemblieren sowie Programmverbindungen für das Anwendungsprogramm herstellen und den Plan binden können. Details zur Erstellung eines Anwendungsprogramms finden Sie in [Programmierung für Db2 for z/OS](#) in der Produktdokumentation zu Db2 for z/OS.
- Vom Benutzer zur Stapelverarbeitung übergebene JCL: Die Member DSNTEJ5C und DSNTEJ5P in der Db2-Bibliothek SDSNSAMP enthalten Beispiele für die JCL, die zur Erstellung von COBOL- und PL/I-Programmen für CICS erforderlich ist.

Wenn Sie Ihr Programm für die Ausführung erstellen, während CICS ausgeführt wird, müssen Sie möglicherweise den Befehl CEMT NEWCOPY ausgeben, damit CICS die neue Version des Programms erkennt.

CICS-Formatierungsroutine SQLCA

Die von IBM bereitgestellte Formatierungsprozedur für SQLCODE-Nachrichten DSNTIAR ermöglicht Ihnen, SQL-Nachrichten online an Ihre Anwendung zu senden.

In DB2 Version 3.1 wurde DSNTIAR in zwei Front-End-Module (DSNTIAC und DSNTIAR) sowie ein Laufzeitmodell (DSNTIA1) aufgeteilt. DSNTIAC wird für CICS-Anwendungen und DSNTIAR für andere Db2-Schnittstellen verwendet. Wegen dieser Änderung ist es ab DB2 Version 3.1 nicht mehr erforderlich, bei jeder an DSNTIAR vorgenommenen Änderung (Änderung des Releases oder Anwenden von Wartung) die

Programmverbindungen für die Anwendungsmodule erneut herzustellen. Wenn Sie über Anwendungen verfügen, für die zuvor Programmverbindungen mit DSNTIAR hergestellt wurden, sollten Sie in Betracht ziehen, diese Programmverbindungen stattdessen mithilfe von DSNTIAC neu herzustellen; dadurch erzielen Sie Leistungsverbesserungen und die Anwendungen sind isoliert gegen Änderungen, die an DSNTIAR vorgenommen werden.

Die CICS-Front-End-Komponente DSNTIAC wird in der Db2-Bibliothek SDSNSAMP als Quellenmember bereitgestellt.

Die erforderlichen Programmdefinitionen für DSNTIAC und DSNTIA1 werden in der von IBM bereitgestellten Gruppe DFHDB2 in der CICS-Systemdefinitionsdatei bereitgestellt. Sie müssen die Bibliothek SDSNLOAD zur CICS-DFHRPL-Verkettung (nach den CICS-Bibliotheken) hinzufügen, sodass DSNTIA1 geladen werden kann.

Zu bindende Elemente nach einer Programmänderung

Wenn Sie ein Programm ändern, müssen Sie es erstellen und erneut binden, bevor es verwendet werden kann.

Informationen zu diesem Vorgang

Eine Übersicht des Bindeprozesses finden Sie in [Bindeprozess](#). Eine Übersicht über Pläne und Pakete finden Sie in [Pläne, Pakete und dynamische Planexits](#).

Stellen Sie sich vor, Sie haben eine CICS-Transaktion, die aus vier Programmmodulen besteht: Modul 1 ist das Hauptmodul. Modul 1 ruft Modul 2 auf. Modul 1 ruft außerdem Modul 3 auf, und Modul 3 ruft Modul 4 auf. Es ist nicht ungewöhnlich, dass die Anzahl der Module in einer echten Transaktion hoch ist. Wenn vorausgesetzt wird, dass mindestens eine SQL-Anweisung in einem der Module geändert wurde, müssen Sie die folgende Prozedur ausführen, um das Programm vorzubereiten und die Transaktion erneut ausführbar zu machen.

Vorgehensweise

1. Führen Sie für das Programm eine Vorkompilierung in Db2 aus.
2. Setzen Sie das Programm mithilfe des CICS-Umsetzers um.
3. Kompilieren Sie die Quellenanweisungen der Hostprogrammiersprache.
4. Stellen Sie Programmverbindungen her.
5. **Wenn das Datenbankankforderungsmodul für Programm C in ein Paket gebunden wurde**, müssen Sie dieses Paket mithilfe des neuen Datenbankankforderungsmoduls binden, sodass alle Anwendungspläne, die Programm C verwenden, das neue Paket automatisch lokalisieren können.
6. **Wenn das Datenbankankforderungsmodul für Programm C direkt in Anwendungspläne gebunden wurde**, müssen Sie alle Anwendungspläne lokalisieren, die das Datenbankankforderungsmodul für Programm C enthalten. Um neue Anwendungspläne zu abzurufen, binden Sie alle Anwendungspläne erneut, wobei Sie die Datenbankankforderungsmodule aller Programme verwenden, die direkt in ihnen gebunden sind. Verwenden Sie für die Programme, die nicht geändert wurden, die zugehörigen alten Datenbankankforderungsmodule. Beachten Sie, dass Sie *nicht* den Unterbefehl REBIND verwenden können, weil *nicht* die Datenbankankforderungsmodule, sondern der Plan die Eingabe für REBIND ist.

Anmerkung: Falls Sie zuvor noch keine Pakete verwendet haben, müssen Sie beachten, dass die Verwendung von Paketen den Prozess für das erneute Binden vereinfacht. Sie können jedes einzelne Datenbankankforderungsmodul als Paket binden und diese in eine Paketliste einschließen. Die Paketliste kann anschließend in einen PLAN eingeschlossen werden. Sie können anschließend den Befehl BIND PACKAGE verwenden, um für beliebige geänderte Programme die Datenbankankforderungsmodule zu binden, anstatt mithilfe des Befehls BIND PLAN den gesamten Anwendungsplan zu binden. Dadurch wird die Verfügbarkeit von Transaktionen erhöht und die Leistung verbessert. Weitere Informationen zur Verwendung von Paketen finden Sie in [Db2-Pakete verwenden](#).

Bindeoptionen und -aspekte bei Programmen

Wenn mehrere Programme in einen Anwendungsplan gebunden werden, müssen Sie auf die Art und Weise achten, wie Db2 Zeitmarken verwendet.

Für jedes Programm erstellt der Db2-Vorcompiler Folgendes:

- Der Db2-Vorcompiler erstellt ein Datenbankankforderungsmodul mit der Zeitmarke 'Td x'. Beispielsweise für das erste Programm 'Td1', für das zweite Programm 'Td2' und so weiter.
- Der Db2-Vorcompiler erstellt in der Liste mit den SQL-Parametern ein modifiziertes Quellenprogramm mit der Zeitmarke 'Ts x'. Beispielsweise 'Ts1' und 'Ts2', falls zwei Programme beteiligt sind.

Zur Bindezeit wird das jeweilige Datenbankankforderungsmodul der einzelnen Programme in das Paket oder den Plan gebunden, das bzw. den Sie angegeben haben. Darüber hinaus aktualisiert Db2 seine Katalogtabelle SYSIBM.SYSDBRM, indem pro Datenbankankforderungsmodul eine Zeile einschließlich der zugehörigen Zeitmarke hinzugefügt wird. Während der Ausführung überprüft Db2 die Zeitmarken der einzelnen SQL-Anweisungen und gibt den SQL-Code -818 zurück, wenn die Zeitmarke für das Datenbankankforderungsmodul und die Zeitmarke, die im Quellenprogramm gesetzt wurde, unterschiedlich sind (in unserem Beispiel, wenn 'Td1' und 'Ts1' bzw. wenn 'Td2' und 'Ts2' unterschiedlich sind). Verwenden Sie eine der folgenden Strategien, um SQL-Codes des Typs -818 zu vermeiden:

- Binden Sie alle Programme in Pakete und listen Sie diese Pakete im Anwendungsplan auf. Wenn ein Programm geändert wird, führen Sie eine Vorkompilierung durch, Sie kompilieren das Programm, stellen Programmverbindungen für das Programm her und binden es erneut in ein Paket.
- Wenn Sie Programme direkt in Anwendungspläne binden, müssen Sie sicherstellen, dass Sie für jedes neue oder geänderte Programm eine Vorkompilierung ausführen, es kompilieren und Programmverbindungen für das Programm herstellen, und anschließend alle Anwendungspläne binden, die das Programm enthalten, wobei die Datenbankankforderungsmodule aller derer Programme verwendet werden müssen, die direkt in diese Pläne gebunden sind. Verwenden Sie hierfür den Befehl BIND, nicht den Befehl REBIND.

Wenn Sie einen Plan binden, steht eine Reihe von Optionen zur Verfügung. Fast alle Bindeoptionen sind anwendungsabhängig; dies sollte beim Anwendungsdesign berücksichtigt werden. Sie sollten Prozeduren entwickeln, um für unterschiedliche Pläne unterschiedliche BIND-Optionen zu verarbeiten. Außerdem sollten die Prozeduren in der Lage sein, im Zeitverlauf Änderungen für ein- und denselben Plan zu verarbeiten.

In den folgenden Abschnitten werden einige besondere Empfehlungen für Bindeoptionen in Verbindung mit CICS beschrieben.

RETAIN

RETAIN bedeutet, dass die BIND- und EXECUTE-Berechtigungen aus dem alten Plan nicht geändert werden.

Wenn die Option RETAIN nicht verwendet wird, werden alle Berechtigungen aus früheren GRANT-Operationen entzogen (REVOKE). Der Benutzer, der den Befehl BIND ausführt, wird Ersteller des Plans und alle Berechtigungen müssen mithilfe von neuen GRANT-Befehlen erneut erteilt werden.

Aus diesem Grund wird empfohlen, dass Sie beim Binden Ihrer Pläne in der CICS-Umgebung die Option RETAIN anwenden.

Isolationsstufe

Die Isolationsstufe wird für den vollständigen Plan angegeben. Es wird empfohlen, Cursorstabilität (CS – Cursor Stability) zu verwenden, es sei denn, es besteht ein besonderer Bedarf für die Verwendung des wiederholbaren Lesevorgangs (RR – Repeatable Read). Die Verwendung der Cursorstabilität ermöglicht eine hohe Anzahl gemeinsamer Zugriffe und verringert das Risiko von Deadlocks.

Beachten Sie, dass die Isolationsstufe für den gesamten Plan angegeben wird. Wenn also der wiederholbare Lesevorgang für ein bestimmtes Modul in CICS erforderlich ist, bedeutet dies, dass alle im Plan enthaltenen Datenbankankforderungsmodule ebenfalls den wiederholbaren Lesevorgang verwenden müssen.

Wenn Sie darüber hinaus aufgrund von Leistungsaspekten entscheiden, eine Reihe von selten verwendeten Transaktionen zu gruppieren, um dasselbe DB2ENTRY-Objekt zu verwenden und diese Gruppe einen gemeinsamen Plan verwenden zu lassen, muss dieser neue Plan auch den wiederholbaren Lesevorgang verwenden, selbst wenn nur eine einzige der Transaktionen den wiederholbaren Lesevorgang erfordert.

Zeit für Validierung von Plänen

Ein Plan wird mit VALIDATE(RUN) oder VALIDATE(BIND) gebunden. VALIDATE(RUN) wird verwendet, um festzulegen, wie SQL-Anweisungen zu verarbeiten sind, die nicht gebunden werden können.

Wenn eine Anweisung zur Ausführungszeit gebunden werden muss, wird sie für jede Ausführung erneut gebunden. Dies bedeutet, dass die Anweisung für jede neue Arbeitseinheit (UOW – Unit of Work) erneut gebunden wird.

Das Binden einer Anweisung während der Ausführung kann sich auf die Leistung auswirken. Eine während der Ausführung gebundene Anweisung wird für jede Ausführung erneut gebunden. Dies bedeutet, dass die Anweisung nach jedem Synchronisationspunkt erneut gebunden werden muss. Es wird nicht empfohlen, diese Option mit CICS zu verwenden.

Beachten Sie, dass VALIDATE(RUN) bei Verwendung von dynamischem SQL nicht erforderlich ist. Das dynamische SQL impliziert jedoch, dass eine Anweisung während der Ausführung gebunden wird.

Sie sollten in einer CICS-Db2-Umgebung VALIDATE(BIND) verwenden.

ACQUIRE und RELEASE

Die Parameter ACQUIRE und RELEASE ändern sich im Zeitverlauf von Plan zu Plan, weil sie in Beziehung zur Transaktionsrate stehen.

Allgemeine Empfehlungen zu diesen Parametern sind in [BIND-Optionen für optimale Leistung auswählen](#) beschrieben.

CICS-Db2-Programme testen und debuggen

Die standardmäßig in einer CICS-Umgebung verwendeten Tools können zum Testen und Debuggen eines CICS-Anwendungsprogramms verwendet werden, das auf Db2 zugreift. Dazu gehören Folgende: die Execution Diagnostic Facility (EDF), der CICS-Hilfstrace und Transaktionsspeicherauszüge.

Informationen zu diesen und weiteren Problembestimmungsprozessen finden Sie in [Fehlerbehebung bei Db2](#).

In Produktion gehen: Checkliste für CICS-Db2-Anwendungen

In dieser Checkliste sind die Tasks aufgeführt, die Sie nach dem Entwerfen, Entwickeln und Testen einer Anwendung ausführen müssen, um die Anwendung in Produktion zu bringen.

Informationen zu diesem Vorgang

Diese Tasks hängen stark von den Standards ab, die Sie im Testsystem verwendet haben. Die auszuführenden Aufgaben unterscheiden sich aufgrund folgender Faktoren:

- Es gibt für Test und Produktion unterschiedliche Db2-Systeme.
- Nur ein einziges Db2-System wird sowohl für Tests als auch für die Produktion verwendet.

Für die folgende Beschreibung wird vorausgesetzt, dass Sie für Test und Produktion unterschiedliche Db2- und CICS-Subsysteme verwenden.

Geht eine Anwendung in Produktion, umfasst dies folgende Aktivitäten:

DDL zum Erstellen von Produktionsdatenbanken verwenden

Alle DDL-Operationen müssen im Db2-Produktionssystem ausgeführt werden, wobei DDL-Anweisungen des Testsystems als Basis verwendet werden. Es sind möglicherweise einige Änderungen erforderlich, beispielsweise müssen die primären und sekundären Zuordnungen erhöht werden; außerdem

müssen andere Seriennummern für Datenträger sowie ein neuer VCAT-Wert in den CREATE STO-GROUP-Anweisungen definiert werden.

DCLGEN erstellen

Für COBOL- und PL/I-Programme müssen Sie im Db2-Produktionssystem möglicherweise DCLGEN-Operationen (DCLGEN – Deklarationsgenerator) ausführen, für die Sie DCLGEN-Eingaben aus dem Db2-Testsystem verwenden.

In Abhängigkeit von der für Kompilierungen ausgewählten Option (wenn im Produktionssystem keine Kompilierungen ausgeführt werden) wäre es eine Alternative, die DCLGEN-Ausgabestrukturen aus den Testbibliotheken in die Produktionsbibliotheken zu kopieren. Dadurch werden sämtliche Informationen der Test- und Produktionssysteme separat gehalten.

Vorkompilierung des Produktionssystems

Wenn Sie Ihre Programme im Testsystem in Pakete gebunden haben, müssen Sie diesen Schritt nicht ausführen. Sie können die Pakete direkt in das Produktionssystem verschieben. Weitere Informationen zur Vorgehensweise finden Sie im Abschnitt 'Anwendungsplan für das Produktionssystem generieren'. Wenn Sie Ihre Programme jedoch direkt in Anwendungspläne binden wollen oder wenn Sie die Programme im Produktionssystem in Pakete binden wollen, müssen Sie die Datenbankanforderungsmodule für die Programme in das Produktionssystem versetzen. Wählen Sie eine der folgenden Möglichkeiten:

- Führen Sie für CICS-Module, die EXEC SQL-Anweisungen enthalten, eine Vorkompilierung im Produktionssystem aus, oder
- Kopieren Sie Datenbankanforderungsmodule aus dem Testsystem in die Bibliotheken des Produktionssystems.

Produktionssystem kompilieren und Programmverbindungen herstellen

Gehen Sie wie folgt vor, um Lademodule zu generieren:

- Wurden die Datenbankanforderungsmodule durch Vorkompilierung im Produktionssystem generiert, müssen Sie die CICS-Module im Produktionssystem kompilieren und die zugehörigen Programmverbindungen herstellen, oder:
- Wenn die Datenbankanforderungsmodule kopiert wurden oder wenn Sie Pakete aus dem Testsystem in das Produktionssystem verschieben, müssen Sie die Lademodule aus dem Testsystem in die Bibliotheken des Produktionssystems kopieren.

In Tabelle 9 auf Seite 116 wird ein Verfahren dargestellt, das Sie verwenden können, um ein geändertes Lademodul aus einem Testsystem in die Bibliothek eines Produktionssystem zu kopieren, wodurch die alte Version des Lademoduls ersetzt wird.

Tabelle 9. Geändertes Programm aus der Testumgebung in die Produktionsumgebung verschieben		
Testsystem	Produktionssystem	Hinweise
	USER.PROD.LOADLIB(PGM3)	Das ursprüngliche Lademodul.
USER.TEST.LOADLIB(PGM3)		Das Lademodul der Testumgebung.
	USER.OLD.PROD.LOADLIB(PGM3)	Die alte Version des Programms wird in eine andere Produktionsbibliothek gestellt.
	USER.PROD.LOADLIB(PGM3)	Die neue Version des Programms wird in die Produktionsbibliothek gestellt.

Wenn Sie unter Verwendung der richtigen JCL die Bibliothek für die Produktionsausführung auswählen, können Sie entweder die alte Version oder die neue Version des Programms verwenden. Es wird dann die richtige Version des Pakets ausgeführt, die durch das Konsistenztoken festgelegt ist, das im Lademodul des Programms integriert ist.

Anwendungsplan für das Produktionssystem generieren

Wenn Sie Ihre Programme auf dem Testsystem in Pakete gebunden haben, können Sie die Pakete in das Produktionssystem kopieren, indem Sie sie in eine Sammlung einschließen, die im Anwendungsplan aufgeführt ist. Wenn Sie ein Paket in das Produktionssystem kopieren, müssen Sie den Anwendungsplan im Produktionssystem nicht erneut binden, solange das Paket in eine Sammlung eingeschlossen wurde, die bereits im Anwendungsplan aufgeführt ist.

In [Tabelle 10](#) auf Seite 117 wird ein Verfahren gezeigt, das Sie zum Kopieren eines geänderten Pakets aus einem Testsystem in die Bibliothek eines Produktionssystems verwenden können, wobei die alte Version des Pakets ersetzt wird. In diesem Beispiel wird zum Zeitpunkt der Vorkompilierung das Schlüsselwort VERSION verwendet, um die einzelnen Versionen der Pakete zu unterscheiden. Eine vollständige Erläuterung und Hinweise zur Verwendung des Schlüsselworts VERSION finden Sie in [Programmierung für Db2 for z/OS](#) in der Produktdokumentation zu Db2 for z/OS.

Tabelle 10. Geändertes Paket aus der Testumgebung in die Produktionsumgebung verschieben		
Testsystem	Produktionssystem	Hinweise
	location_name. PROD_COLL.PRG3.VER1	Die alte Version des Pakets.
location_name. TEST_COLL.PRG3.VER2		Eine neue Version des Pakets wird im Testsystem gebunden und anschließend in das Produktionssystem kopiert.
	location_name. PROD_COLL.PRG3.VER1	Die alte Version befindet sich weiterhin in der Sammlung des Produktionssystems.
	location_name. PROD_COLL.PRG3.VER2	Die neue Version wird in die Sammlung des Produktionssystem gestellt.

Wenn Sie Ihre Programme direkt in Anwendungspläne binden wollen oder wenn Sie die Programme im Produktionssystem in Pakete binden wollen, müssen Sie den Bindeprozess für die Datenbankanforderungsmodule ausführen, die Sie in das Produktionssystem gestellt haben. Wenn Sie Ihre Programme direkt in Anwendungspläne binden, müssen Sie alle Anwendungspläne im Produktionssystem binden, die diese Programme enthalten. Weitere Informationen zum Bindeprozess finden Sie in [Bindeprozess](#). Beachten Sie, dass aufgrund verschiedener Faktoren wie beispielsweise der Größe von Tabellen und Indizes ein Vergleich der EXPLAIN-Ausgabe des Testsystems mit der Ausgabe des Produktionssystems nutzlos sein kann. Es wird dennoch empfohlen, dass Sie beim ersten Binden eines Plans im Produktionssystem EXPLAIN ausführen, um die Entscheidungen des Db2-Optimierungsprogramms zu prüfen.

GRANT EXECUTE

Sie müssen Benutzern die Ausführungsberechtigung (EXECUTE) für die Db2-Anwendungspläne im Produktionssystem erteilen.

Tests

Obwohl zu diesem Zeitpunkt keine weiteren Tests erforderlich sein sollten, sind Belastungstests nützlich und empfohlen, um das Auftreten von Ressourcenkonflikten, Deadlocks und Zeitlimitüberschreitungen zu minimieren und um zu prüfen, ob die Antwortzeit von Transaktionen wie erwartet ausfällt.

CICS-Definitionen

Damit neue Anwendungsprogramme zur Ausführung bereit sind, aktualisieren Sie die folgenden RDO-Definitionen (RDO – Resource Definition Online) im CICS-Produktionssystem.

- RDO-Transaktionsdefinitionen für neue Transaktionscodes
- RDO-Programmdefinitionen für neue Anwendungsprogramme und Zuordnungen

- SIT (SIT – System Initialization Parameter, Systeminitialisierungsparameter) für bestimmte Anforderungen von Db2, wenn es sich um die erste Db2-orientierte Anwendung handelt, die in Produktion geht
- RDO-DB2ENTRY- und RDO-DB2TRAN-Definitionen für die Anwendungen. Die RDO-DB2CONN-Definition, wenn es sich um die erste Db2-orientierte Anwendung handelt, die in Produktion geht. Bei der Definition der neuen Transaktionen und Anwendungspläne im DB2ENTRY-Objekt können Sie ungeschützte Threads verwenden, um am Anfang detaillierte Abrechnungs- und Leistungsinformationen zu erhalten. Später können Sie geschützte Threads nach Bedarf verwenden.

Wenn RACF installiert ist, müssen Sie darüber hinaus neue Benutzer und Db2-Objekte definieren.

CICS-Anwendung optimieren, die auf Db2 zugreift

CICS-Anwendungen, die auf Db2 zugreifen, müssen optimiert werden, bevor Sie in die Produktion versetzt werden, und sie müssen in regelmäßigen Abständen optimiert werden, während sie sich in Produktion befinden.

Informationen zu diesem Vorgang

Wenn eine CICS-Anwendung, die auf Db2 zugreift, in die Produktion verschoben wird, müssen Sie zu den Prüfungen, die bereits für CICS ausgeführt werden, noch folgende Prüfungen hinzufügen:

- Prüfen Sie, dass alle Anwendungsprogramme, die Db2-Anforderungen erstellen, threadsicher sind. Ist dies der Fall, können Sie die Umgebung für offene Transaktionen (OTE – Open Transaction Environment) nutzen und die Leistung der Anwendung wird verbessert. In CICS-Db2-Anwendungen für die Verwendung von OTE durch threadsichere Programmierung aktivieren finden Sie eine Erläuterung der Vorgehensweise, wie Anwendungsprogramme in der Umgebung für offene Transaktionen funktionieren.
- Stellen Sie sicher, dass Anzahl und Typ der verwendeten SQL-Anweisungen mit den Programmspezifikationen übereinstimmen (verwenden Sie die Db2-Abrechnungsfunktion).
- Prüfen Sie, ob die Anzahl der abgerufenen und aktualisierten Seiten im Pufferpool höher als erwartet ist (verwenden Sie die Db2-Abrechnungsfunktion).
- Prüfen Sie, ob geplante Indizes verwendet werden (verwenden Sie EXPLAIN) und dass nicht effiziente SQL-Anweisungen nicht verwendet werden.
- Prüfen Sie, ob DDL (DDL – Data Definition Language) verwendet wird; ist dies der Fall, prüfen Sie, aus welchen Gründen DDL verwendet wird (verwenden Sie die Db2-Abrechnungsfunktion).
- Prüfen Sie, ob Dialogtransaktionen verwendet werden.

Stellen Sie fest, ob stattdessen pseudodialogfähige Transaktionen verwendet werden können. Ist das dialogfähige Design erforderlich, müssen Sie die Db2-Objekte prüfen, die konversationsübergreifend gesperrt sind. Prüfen Sie auch, dass die Anzahl der aufgrund dieses dialogorientierten Designs erforderlichen neuen Threads akzeptabel ist.

- Prüfen Sie die verwendeten Sperren und ihre Dauer.

Stellen Sie sicher, dass Tabellenbereichssperren nicht wegen falscher oder suboptimaler Spezifikation folgender Elemente verwendet werden (Beispiele):

- Anweisung LOCK TABLE
- Spezifikation LOCKSIZE=TS
- Spezifikation ISOLATION LEVEL(RR)
- Sperreneskalation.

Diese Informationen sind in den Katalogtabellen verfügbar, mit Ausnahme der Sperreneskalation, bei der es sich um einen Installationsparameter (DSNZPARM) handelt.

- Prüfen Sie die verwendeten Pläne und ihre Größe. Auch wenn die Anwendungspläne segmentiert sind, gilt doch, dass je mehr Datenbankanforderungsmodule im Plan verwendet werden, desto länger ist die Zeit, die im Fall einer Änderung zum Binden (BIND) und erneuten Binden (REBIND) der Pläne benötigt

wird. Versuchen Sie nach Möglichkeit, Pakete zu verwenden. Pakete wurden entwickelt, um Probleme der folgenden Situationen zu lösen:

- Erneutes Binden des gesamten Plans nach Änderung Ihrer SQL-Anwendung. (Dies wurde durch die dynamische Planauswahl gelöst, mit Einbußen bei der Leistung.)
- Binden jedes einzelnen Anwendungsplans, wenn die geänderte SQL-Anwendung von vielen Anwendungen verwendet wird.

Verwenden Sie nach Abschluss dieser Optimierung die erwartete Transaktionslast, um über die Art der erforderlichen DB2ENTRY-Definitionen und die erforderliche Threadanzahl zu entscheiden. Prüfen Sie auch den Einfluss dieser Transaktionen auf die Db2- und CICS-Subsysteme.

Gehen Sie beim Optimieren einer CICS-Anwendung, die im Produktionsstatus auf Db2 zugreift, wie folgt vor:

- Überprüfen Sie, ob die CICS-Anwendungen die geplanten Indizes verwenden, indem Sie die Anzahl der GET PAGES-Elemente im Pufferpool überwachen (verwenden Sie die Db2-Abrechnungsfunktion). Der Grund dafür, dass ein Index nicht verwendet wird, kann sein, dass der Index gelöscht wurde oder dass der Index nach dem Binden des Plans erstellt wurde.
- Verwenden Sie die Daten des Sperrenmanagers aus der Abrechnungsfunktion, um zu prüfen, ob Aussetzungen, Deadlocks und Zeitlimitüberschreitungen vorhanden sind.

Kapitel 8. Db2 überwachen

Abrechnung und Überwachung in einer CICS-Db2-Umgebung.

Von CICS bereitgestellte Abrechnungs- und Überwachungsinformationen

CICS schließt einige Funktionen zur Abrechnung und Überwachung ein. Sie können diese Funktionen verwenden, um die Verwendung verschiedener Ressourcen in einem CICS-System zu messen.

Die am häufigsten verwendeten Tools lauten wie folgt:

- Statistische Daten.

CICS-Statistiken sind das Tool, mit dem ein CICS-System am einfachsten überwacht werden kann. Sie enthalten Informationen zum CICS-System als Ganzes, wie beispielsweise zur Leistung und zur Verwendung der Ressourcen. CICS-Statistiken sind für die Leistungsoptimierung und Kapazitätsplanung geeignet. CICS erfasst Statistikdaten während der Onlineverarbeitung, um sie später offline zu verarbeiten. Die Statistikdomäne erfasst diese Daten und schreibt anschließend Datensätze in die von MVS bereitgestellte SMF-Datei (SMF – System Management Facility). Die Datensätze haben den SMF-Typ 110. Sie können diese Datensätze mithilfe des Programms DFHSTUP offline verarbeiten.

- Überwachungsdaten.

Die CICS-Überwachung erfasst während der Onlineverarbeitung Daten zu allen von Benutzern und von CICS bereitgestellten Transaktionen, um sie später offline zu analysieren. Die von der CICS-Überwachung erzeugten Datensätze haben ebenfalls den SMF-Typ 110 und sie werden in die SMF-Dateien geschrieben. Die von der CICS-Überwachung bereitgestellten Daten sind für die Leistungsoptimierung nützlich und Sie können Ihren Benutzern mithilfe dieser Daten die Kosten für verwendete Ressourcen zuordnen. Die Überwachung stellt folgende Klassen von Daten bereit:

- Leistungsklasse für detaillierte Informationen auf Transaktionsebene
- Transaktionsressourcendaten für zusätzliche, auf Transaktionsebene vorhandene Informationen zu einzelnen Ressourcen, auf die von einer Transaktion zugegriffen wird.
- Ausnahmeklasse für Ausnahmebedingungen

CICS ist ein Adressraum mit Multitasking und die CICS-Überwachungsfunktionen werden in der Regel verwendet, um die Prozessorzeit und andere Ressourcen zu ermitteln, die von den einzelnen Transaktionen oder Funktionen verbraucht wurden, die von CICS ausgeführt werden.

Eine vollständige Beschreibung der CICS-Überwachungsfunktionen sowie Details zur Aktivierung, Erfassung und Verarbeitung dieser Informationen finden Sie in [CICS-Überwachungsfunktion: Leistung und Optimierung](#).

Sowohl für Statistikdaten als auch für Überwachungsdaten können Sie eine offline ausgeführte Verarbeitungsfunktion verwenden. CICS Performance Analyzer und IBM Z Decision Support sind zwei Tools, die Daten aus CICS und anderen Systemen und Produkten von IBM erfassen und analysieren. Sie können Berichte erstellen, die Sie bei Folgendem unterstützen:

- Systemübersichten
- Service-Levels
- Verfügbarkeit
- Leistung und Optimierung
- Kapazitätsplanung

Weitere Informationen finden Sie in [CICS Performance Analyzer for z/OS \(CICS PA\)](#) und [IBM Z Decision Support](#) unter "[Leistung verbessern](#)".

Von Db2 bereitgestellte Abrechnungs- und Überwachungsinformationen

Die Komponente der Instrumentierungsfunktion von Db2 ermöglicht Ihnen, sechs verschiedene Tracetypen zu verwenden: Statistik, Abrechnung, Prüfen, Leistung, Überwachung und Global. Für jeden Tracetypen können Sie eine Reihe von Traceklassen aktivieren.

Sie können als Ziel der Traceausgabe SMF (SMF – System Management Facility, Systemverwaltungsfunktion) verwenden. Als Alternative können Sie die Traceausgabe unter der Steuerung von GTF auslagern.

Statistik

Beschreibt die gesamte in Db2 ausgeführte Arbeit. Diese Informationen sind nicht auf einen bestimmten Benutzer bezogen. Der Hauptzweck des Db2-Statistiktrace besteht in Folgendem:

- Daten für die Db2-Kapazitätsplanung bereitstellen.
- Überwachung und Optimierung auf Db2-Subsystemebene unterstützen.
- Abrechnung der Db2-Aktivität unterstützen.

Die Statistikdatensätze werden in benutzerdefinierten Intervallen geschrieben. Sie können das Intervall für die Statistikerfassung und die Entstehungszeit ohne Stoppen und Starten des Trace zurücksetzen, indem Sie den Befehl MODIFY TRACE verwenden. Jegliche Db2-Aktivität im Intervall für die Statistikerfassung wird im Datensatz erfasst. Dadurch wird es möglicherweise schwierig, zwischen der Aktivität und bestimmten Benutzern einen direkten Bezug herzustellen.

Der Db2-Statistiktrace kann für verschiedene Klassen aktiviert werden. Wenn die Statistikdatensätze in SMF geschrieben werden, so lauten die SMF-Typen 100 und 102.

Abrechnung

Beschreibt die Arbeit, die für einen bestimmten Benutzer ausgeführt wird (Berechtigungs-ID aus dem DB2CONN-Objekt bzw. dem DB2ENTRY-Objekt). Der Hauptzweck der Abrechnungsdatensätze besteht darin, die Db2-Kosten der Berechtigungs-ID zuzuordnen und Überwachung und Optimierung auf Programmebene auszuführen. Db2 erzeugt einen Abrechnungsdatensatz bei Beendigung des Threads oder wenn eine Transaktion einen Thread mit einer neuen Berechtigungs-ID wiederverwendet. Wenn ein Thread als geschützt definiert ist (PROTECTNUM>0) und alle Transaktionen mit demselben Transaktionscode für dieses DB2ENTRY-Objekt dieselbe Berechtigungs-ID verwenden, bedeutet dies, dass nur ein einziger Abrechnungsdatensatz erzeugt wird, der jegliche im Thread ausgeführte Aktivität beschreibt. Darüber hinaus werden Abrechnungsdatensätze geschrieben, wenn Sie in Ihren DB2ENTRY- oder DB2CONN-Definitionen für ACCOUNTREC die Einstellungen UOW, TASK oder TXID festlegen. Wenn für diese Optionen ACCOUNTREC festgelegt wird, so wird dies als Anmeldung bewertet, auch wenn Sie dieselbe Berechtigungs-ID verwenden.

Sie können den Db2-Abrechnungstrace für mehrere Klassen aktivieren. Wenn die Abrechnungsdatensätze in SMF geschrieben werden, so lauten die SMF-Typen 101 und 102.

Prüfen

Erfasst Informationen zu Db2-Sicherheitsmaßnahmen und wird verwendet, um sicherzustellen, dass der Datenzugriff nur für berechtigte Zwecke zulässig ist. Wenn die Prüfdatensätze in SMF geschrieben werden, so lautet der SMF-Typ 102.

Leistung

Erfasst Informationen zur einer Reihe von verschiedenen Ereignisklassen. Die Informationen sind für Folgendes bestimmt:

- Programmbezogene Überwachung und Optimierung
- Ressourcenbezogene Überwachung und Optimierung
- Benutzerbezogene Überwachung und Optimierung
- Systembezogene Überwachung und Optimierung
- Abrechnungsbezogene Profilerstellung

Sie können den Db2-Leistungstrace für mehrere Klassen aktivieren. Wenn die Leistungsdatensätze in SMF geschrieben werden, so lautet der SMF-Typ 102.

Überwachung

Erfasst mithilfe von Programmen, die vom Benutzer geschrieben wurden, Daten für die *Onlineüberwachung*.

Global

Unterstützt die Wartungsfreundlichkeit. Wenn die Datensätze für den globalen Trace in SMF geschrieben werden, so lautet der SMF-Typ 102.

CICS-Db2-Umgebung überwachen: Übersicht

Das Ziel einer Überwachung der CICS-Db2-Anschlussfunktion besteht darin, eine Basis für die Abrechnung und die Optimierung bereitzustellen.

Informationen zu diesem Vorgang

Durch die Überwachung können Sie folgende Daten erhalten:

- Die Anzahl der Transaktionen, die auf Db2-Ressourcen zugreifen.
- Die durchschnittliche Anzahl von SQL-Anweisungen, die von einer Transaktion ausgegeben werden.
- Die durchschnittliche Prozessorbelegung für eine Transaktion.
- Die durchschnittliche Antwortzeit für eine Transaktion.
- Die Kosten, die bestimmten Transaktionen zuzuordnen sind.
- Die einer Transaktion zugeordnete Pufferpoolaktivität.
- Die einer Transaktion zugeordnete Sperrenaktivität. Dies schließt Informationen dazu ein, ob Tabellenbereichssperren statt Seitensperren verwendet werden, und ob Sperreneskaltungen auftreten, beispielsweise aufgrund von wiederholbaren Lesevorgängen.
- Der Umfang der Threadverwendung für DB2ENTRY-Objekte und den Pool.
- Der Umfang der Threadwiederverwendung von geschützten Threads in DB2ENTRY-Objekten.

Es wird empfohlen, die Testumgebung aus den folgenden Gründen zu überwachen:

- Um zu prüfen, dass neue Programme ordnungsgemäß für Testdatenbanken funktionieren (d. h., dass sie die richtige Aufrufreihenfolge verwenden).
- Um alle Leistungsprobleme zu ermitteln, die aufgrund einer übermäßigen Anzahl von E/A-Operationen oder aufgrund von nicht effizienten SQL-Anweisungen auftreten.
- Um fehlerhafte Designverfahren zu erkennen, wie z. B., dass Db2-Ressourcen für verschiedene Bildschirmkonversationen beibehalten werden.
- Um optimale Sperrenprotokolle festzulegen, mit denen die Anforderungen der Anwendungsisolierung mit den Anforderungen vorhandener Anwendungen ausgeglichen werden.

Schließen Sie die Überwachung in die Abnahmeverfahren für neue Anwendungen ein, sodass alle während der Testzeit nicht erkannten Probleme schnell gefunden und korrigiert werden können.

Sie können zur Überwachung der CICS-Db2-Anschlussfunktion und der CICS-Transaktionen, die auf Db2-Ressourcen zugreifen, einige oder alle der folgenden Tools verwenden.

- Überwachen Sie die CICS-Db2-Anschlussfunktion mithilfe von Folgendem:
 - Befehle der CICS-Db2-Anschlussfunktion
 - Db2-Befehle
 - CICS-Db2-Statistiken

Siehe „CICS-Db2-Anschlussfunktion überwachen“ auf Seite 124.

- Überwachen Sie CICS-Transaktionen mithilfe von Folgendem:
 - CICS-Überwachungsfunktion (CMF – CICS Monitoring Facility)
 - CICS-Hilfstrace

Siehe „CICS-Transaktionen überwachen, die auf Db2-Ressourcen zugreifen“ auf Seite 127.

- Überwachen Sie Db2 mithilfe von Folgendem:

- Db2-Statistikdatensätze
- Db2-Abrechnungsdatensätze
- Db2-Leistungsdatensätze

Siehe „Db2 bei Verwendung mit CICS überwachen“ auf Seite 128.

- Überwachen Sie das CICS-System (z. B. den Dispatcher) mit CICS-Statistiken. Siehe „CICS-System in einer CICS-Db2-Umgebung überwachen“ auf Seite 130.

CICS-Db2-Anschlussfunktion überwachen

Überwachen Sie die CICS-Db2-Anschlussfunktion mithilfe von Befehlen, die an Db2 und die CICS-Db2-Anschlussfunktion selbst gerichtet sind.

Informationen zu diesem Vorgang

CICS-Db2-Anschlussfunktion mithilfe von Befehlen der CICS-Db2-Anschlussfunktion überwachen

Sie können den Status der CICS-Db2-Threads und der entsprechenden CICS-Transaktionen überwachen, die diese Threads verwenden, indem Sie den Befehl `DSNC DISPLAY PLAN` oder `TRAN` nutzen, der von der CICS-Db2-Anschlussfunktion bereitgestellt wird.

Sie können auf das `DB2CONN`-Objekt oder auf einzelne `DB2ENTRY`-Objekte auch Befehle verwenden, die von CICS bereitgestellt werden, wie beispielsweise den Befehl `CEMT` oder **`EXEC CICS INQUIRE`**. Werden diese Befehle im `DB2CONN`-Objekt verwendet, ermöglichen sie Ihnen die Überwachung des Status der allgemeinen Verbindung zwischen CICS und Db2 sowie die Überwachung der Verwendung des Pools. Bei Verwendung für ein einzelnes `DB2ENTRY`-Objekt ermöglichen die Befehle Ihnen zu überwachen, wie das `DB2ENTRY`-Objekt verwendet wird.

Weitere Informationen hierzu finden Sie in [Von CICS bereitgestellte Transaktionen für CICS Db2](#).

CICS-Db2>-Anschlussfunktion mithilfe von Db2-Befehlen überwachen

Sobald eine Verbindung zwischen CICS und Db2 hergestellt ist, können von der CICS-Sicherheit berechnigte Terminalbenutzer die Transaktion `DSNC` zum Weiterleiten von Befehlen an das Db2-System verwenden.

Informationen zu diesem Vorgang

Die Befehle haben das folgende Format:

```
DSNC-db2-befehl
```

Beispielsweise können mit dem Befehl `DSNC-DIS THREAD` CICS-Db2-Threads angezeigt werden.

Der Befehl wird zur Verarbeitung an Db2 weitergeleitet. Db2 überprüft, dass die von CICS übergebene Berechtigungs-ID zum Absetzen des eingegebenen Befehls berechtigt ist.

Antworten werden an den CICS-Benutzer zurückgeleitet, von dem die Befehle stammen. Der Bindestrich (-) muss als Befehlserkennungszeichen verwendet werden, um Db2-Befehle von Befehlen der CICS-Db2-Anschlussfunktion zu unterscheiden. Für Db2-Befehle, die in CICS abgesetzt werden, lautet das Befehlserkennungszeichen stets -, unabhängig davon, welches Befehlserkennungszeichen vom Subsystem verwendet wird.

Sowohl die CICS- als auch die Db2-Berechtigung sind erforderlich, damit Db2-Befehle von einem CICS-Terminal aus abgesetzt werden können:

- Die CICS-Berechtigung ist erforderlich, um die Transaktion `DSNC` zu verwenden, und
- Die Db2-Berechtigung ist erforderlich, um Db2-Befehle abzusetzen.

Weitere Informationen finden Sie in [Von CICS bereitgestellte Transaktionen für CICS Db2](#).

CICS-Db2-Anschlussfunktion mithilfe von CICS-Db2-Statistiken überwachen

Zusätzlich zu der begrenzten Statistikmenge, die mithilfe des Befehls **DSNC DISP STAT** ausgegeben wird, und der Statistikdaten, die während des Schließens der Anschlussfunktion an das Ziel STATSQUEUE des DB2CONN-Objekts ausgegeben werden, kann eine umfassendere Gruppe von CICS-Db2-Statistikdaten mithilfe der standardmäßigen CICS-Statistikschnittstellen erfasst werden.

Informationen zu diesem Vorgang

Die globale Statistik und die Ressourcenstatistik von CICS Db2 werden in [CICS-Db2-Statistik unter "Referenz"](#) detailliert beschrieben.

CICS-Db2-Statistiken werden für alle CICS-Statistiktypen unterstützt, insbesondere:

- **Angeforderte Statistik** - CICS-Db2-Statistiken werden als Ergebnis des Befehls **EXEC CICS PERFORM STATISTICS RECORD** geschrieben, der mit dem Schlüsselwort Db2 ausgegeben wurde.
- **Angeforderte Zurücksetzungsstatistik** – eine besondere Art einer angeforderten Statistik, in der die Statistikzähler nach der Erfassung zurückgesetzt werden.
- **Intervallstatistik** – Statistiken, die bei Ablauf eines angeforderten Intervalls geschrieben werden.
- **Tagesabschlussstatistik** – eine besondere Art der Intervallstatistik.
- **Nicht angeforderte Statistik** – CICS schreibt eine globale Db2-Statistik und eine Ressourcenstatistik an SMF, wenn die Anschlussfunktion beendet wird. Darüber hinaus wird eine Db2-Ressourcenstatistik an SMF geschrieben, wenn ein DB2ENTRY-Element gelöscht wird.

Vorgehensweise

1. Verwenden Sie eine der folgenden Methoden, um Db2-Statistiken zu erfassen.

- Verwenden Sie den Statistikbefehl **EXEC CICS COLLECT** mit dem Schlüsselwort DB2CONN, um zuzulassen, dass globale Statistikdaten von CICS Db2 erfasst werden. Globale Statistikdaten von CICS Db2 werden mit DFHD2GDS DSECT zugeordnet.
- Verwenden Sie den Statistikbefehl **EXEC CICS COLLECT** mit dem Schlüsselwort DB2ENTRY(), um zuzulassen, dass eine CICS Db2-Ressourcenstatistik für ein bestimmtes DB2ENTRY-Element erfasst wird. Die CICS-Db2-Ressourcenstatistik wird mit DFHD2RDS DSECT zugeordnet.
- Verwenden Sie den Befehl **EXEC CICS PERFORM STATISTICS** mit dem Db2-Schlüsselwort, damit der Benutzer anfordern kann, dass eine globale Statistik und eine Ressourcenstatistik zu CICS Db2 an SMF geschrieben wird.

Alternativ können Sie das CICS-Beispielstatistikprogramm DFHOSTAT verwenden, um Db2-Statistikdaten zu erfassen. Es verwendet **EXEC CICS COLLECT STATISTICS**-Befehle mit den Schlüsselwörtern DB2CONN und DB2ENTRY, um Statistikdaten zu erfassen. Es verwendet auch **EXEC CICS INQUIRE**-Befehle für die DB2CONN- und DB2ENTRY-Elemente, um Daten zu erfassen. Ein Beispiel für eine Ausgabe von DFHOSTAT ist in [Abbildung 30 auf Seite 126](#) dargestellt.

2. Nach dem Erfassen der Statistikdaten ist es wichtig, hinsichtlich der Leistung und der Optimierung folgende Felder anzusehen:

- a) Die Anzahl der Aufrufe, die mithilfe eines Threads aus einem DB2ENTRY-Element oder dem Pool erstellt wurden, und die Anzahl der Threadwiederverwendungen (d. h., die Anzahl, wie oft ein vorhandener Thread wiederverwendet wurde). Die Threadwiederverwendung wird pro DB2ENTRY-Element und (in der Db2-Verbindungsstatistik) pro Pool berichtet. Die Threadwiederverwendung wirkt sich günstig auf die Leistung aus, da der Systemaufwand vermieden wird, der erforderlich ist, um für jede einzelne CICS-Transaktion oder jede Arbeitseinheit Threads zu erstellen. Durch die Verwendung geschützter Threads kann die Threadwiederverwendung erhöht werden.
- b) Wenn THREADWAIT(YES) angegeben ist, die maximale Anzahl von Tasks in der Warteschlange 'Readyq', die auf einen Thread warten. Es ist besser, Transaktionen mithilfe einer Transaktionsklasse zu begrenzen, anstatt sie in die Warteschlange für Threads zu stellen.

- c) Überprüfen Sie in der Db2-Verbindungsstatistik das Feld für die maximale Anzahl der Tasks in der Readyq-Warteschlange des Pools und auch das Feld für die maximale Anzahl der Tasks in der Readyq-Warteschlange des Tasksteuerblocks. Wenn das zuletzt Genannte ungleich null ist, wurden Tasks in die Warteschlange gestellt, die auf eine Db2-Verbindung warten, um sie mit ihren offenen Tasksteuerblöcken zu verwenden, statt auf einen Thread zu warten. Die Tasks wurden in die Warteschlange gestellt, da TCBLIMIT, die maximale Anzahl von Tasksteuerblöcken, die zur Steuerung von Threads in Db2 verwendet werden kann, erreicht wurde. Dies zeigt, dass die Anzahl der verfügbaren Threads (die Summe der THREADLIMIT-Werte für den Pool, für Befehlsthreads und für alle DB2ENTRY-Elemente) die zulässige Anzahl von Tasksteuerblöcken überschreitet. Die Werte von TCBLIMIT oder THREADLIMIT sollten in diesem Fall angepasst werden.

Beispiel

Applid IYK2Z2G1	Sysid JOHN	Jobname CI13JTD5	Date 09/09/2001	Time 10:38:50	CICS
6.2.0	PAGE	2			
<u>DB2 Connection</u>					
DB2 Connection Name.	RCTJT				
DB2 Group ID		Resync Group Member.			N/A
DB2 Sysid.	DE2D				
DB2 Release.	6.2.0				
DB2 Connection Status.	CONNECTED	DB2 Connect Date and Time			09/09/2001
10:37:19.21354					
DB2 Connection Error	SQLCODE				
DB2 Standby Mode	RECONNECT				
DB2 Pool Thread Plan Name.					
DB2 Pool Thread Dynamic Plan Exit Name . .	DSNCUEXT				
Pool Thread Authtype	USERID	Command Thread Authtype.			N/A
Pool Thread Authid		Command Thread Authid.			JTILLI1
Signid for Pool/Entry/Command Threads. . .	SSSSSSSS				
Create Thread Error.	ABEND	Message TD Queue 1.			CDB2
Protected Thread Purge Cycle	00.30	Message TD Queue 2.			
Deadlock Resolution.	ROLLBACK	Message TD Queue 3.			
Non-Terminal Intermediate Syncpoint. . . .	NORELEASE	Statistics TD Queue			CDB2
Pool Thread Wait Setting	WAIT	DB2 Accounting records by			NONE
Pool Thread Priority	HIGH				
Current TCB Limit.	100				
Current number of TCBs	10				
Peak number of TCBs.	10				
Current number of free TCBs.	5				
Current number of tasks on TCB Readyq. . .	0				
Peak number of tasks on TCB Readyq	0				
Pool Thread Limit.	3	Number of Calls using Pool			
Threads.	0	Number of Pool Thread Sig□			
Current number of Pool Threads	0	Number of Pool Thread Partial Sig□			
nons	0	Number of Pool Thread Com□			
Peak number of Pool Threads.	2	Number of Pool Thread Ab□			
nons	0	Number of Pool Thread Single Pha□			
Number of Pool Thread Waits.	0	Number of Pool Thread Reu□			
mits	0	Number of Pool Thread Termin□			
orts.	0				
Current number of Pool Tasks	0				
se.	0				
Peak number of Pool Tasks.	0				
ses.	0				
Current Total number of Pool Tasks	0				
ates.	0				
Current number of Tasks on Pool Readyq . .	0				
Peak number of Tasks on Pool Readyq. . . .	0				
Current number of DSNC Command threads . .	0	Number of DSNC Command			
Calls.	0	Number of DSNC Command Sig□			
Peak number of DSNC Command threads. . . .	0	Number of DSNC Command Thread Termin□			
nons.	0				
DSNC Command Thread Limit.	2				
ates.	0				
flows	0				

Abbildung 30. Beispiel einer DFH0STAT-Ausgabe: der Bericht zu den Db2-Verbindungen

Applid IYK2Z2G1	Sysid JOHN	Jobname CI13JTD5	Date 09/09/2001	Time 10:38:50	CICS
6.2.0	PAGE	3			
DB2 Entries					
DB2Entry Name.	XP05	DB2Entry Status	ENABLED		
DB2Entry Static Plan Name.	TESTP05	DB2Entry Disabled Action.	POOL		
DB2Entry Dynamic Plan Exit Name.		DB2Entry Deadlock Resolution.	ROLLBACK		
DB2Entry Authtype.	N/A	DB2Entry Accounting records by.	NONE		
DB2Entry Authid.	JTILLI1	Number of Calls using DB2Entry.			
16,500		Number of DB2Entry Sig□			
DB2Entry Thread Wait Setting	WAIT	Number of DB2Entry Partial Sig□			
nons.	10	Number of DB2Entry Com□			
nons.	0	Number of DB2Entry Ab□			
DB2Entry Thread Priority	HIGH	Number of DB2Entry Single Phase			
mits.	0	Number of DB2Entry Thread Reuses.			
DB2Entry Thread Limit.	10	Number of DB2Entry Thread Termin□			
orts	0	Number of DB2Entry Thread Waits/Over□			
Current number of DB2Entry Threads	0				
5,500					
Peak number of DB2Entry Threads.	10				
5,031					
ates.	464				
DB2Entry Protected Thread Limit.	5				
flows	306				
Current number of DB2Entry Protected Threads	5				
Peak number of DB2Entry Protected Threads.	5				
Current number of DB2Entry Tasks	0				
Peak number of DB2Entry Tasks.	28				
Current Total number of DB2Entry Tasks	5,500				
Current number of Tasks on DB2Entry Readyq	0				
Peak number of Tasks on DB2Entry Readyq.	18				

Abbildung 31. Beispiel einer DFH0STAT-Ausgabe: der Bericht zu den Db2-Einträgen

CICS-Transaktionen überwachen, die auf Db2-Ressourcen zugreifen

CICS bietet Abrechnungs- und Überwachungsfunktionen für die Ressourcen, die für CICS-Transaktionen im CICS-Adressraum erforderlich sind.

Informationen zu diesem Vorgang

Diese CICS-Abrechnungs- und Überwachungsfunktionen können drei Typen von Datensätzen generieren:

- Leistungsdatensätze, die die von den einzelnen Transaktionen im CICS-Adressraum verwendeten Ressourcen erfassen.
- Transaktionsressourcendaten, um zusätzliche Informationen zu einzelnen Ressourcen zu erfassen, auf die eine Transaktion zugegriffen hat.
- Ausnahmedatensätze, die Ressourcenengpässe erfassen.

Zu den Überwachungsdatensätzen für CICS-Leistungsklassen gehören folgende Db2-bezogene Datenfelder in der Gruppe DFHDATA:

DB2REQCT (180)

Die Gesamtzahl der Db2-EXEC-SQL- und IFI-Anforderungen (IFI – Instrumentation Facility Interface), die von einer Transaktion ausgegeben werden.

DB2RDYQW (187)

Die abgelaufene Zeit, in der die Transaktion darauf gewartet hat, dass ein Db2-Thread verfügbar wird.

DB2CONWT (188)

Die abgelaufene Zeit, in der die Transaktion darauf gewartet hat, dass eine Db2-Verbindung verfügbar wird, damit diese mit dem zugehörigen offenen Tasksteuerblock verwendet werden kann.

Die CICS-Überwachung wird in der CICS-Db2-Umgebung mit der Db2-Abrechnungsfunktion verwendet, um die Leistung zu überwachen und um Abrechnungsinformationen zu erfassen.

Weitere Informationen zum Abgleich von CICS-Leistungsklassendatensätzen und Db2-Abrechnungsdatsätzen finden Sie in „[Db2-Abrechnungsdatsätze und CICS-Leistungsklassendatsätze in Beziehung zueinander setzen](#)“ auf Seite 135. Weitere Informationen zur Berechnung des Prozessorverbrauchs finden Sie in „[Abrechnung zur Prozessorbelegung in einer CICS-Db2-Umgebung](#)“ auf Seite 141.

Sie können mithilfe der CICS-Hilfstracefunktion SQL-Aufrufe verfolgen, die von einem CICS-Anwendungsprogramm abgesetzt wurden.

Weitere Informationen zu Traces, die von der CICS-Db2-Anschlussfunktion ausgegeben werden, finden Sie in [Fehlerbehebung bei Db2](#).

Db2 bei Verwendung mit CICS überwachen

Verwenden Sie die von Db2 erzeugten SMF- oder GTF-Datensätze, um die Verwendung von Db2 mit CICS zu überwachen.

Informationen zu diesem Vorgang

Das Lizenzprogramm Db2 Performance Monitor (DB2PM) ist nützlich, um Berichte bereitzustellen, die auf folgenden Elementen basieren:

- Statistikdatensätze
- Abrechnungsdatensätze
- Leistungsdatensätze

Die Berichte in diesem Abschnitt werden als Beispiele angezeigt. Lesen Sie in der Dokumentation des von Ihnen verwendeten DB2PM-Releases Informationen zum Format und zur Bedeutung der in diesen Berichten enthaltenen Felder nach.

Db2 mithilfe der Db2-Statistikfunktion überwachen

Db2 generiert am Ende jedes Zeitintervalls Statistikdaten auf Subsystembasis, wie zum Zeitpunkt der Installation angegeben. Diese Daten werden nur erfasst und in die SMF- und GTF-Datei geschrieben, wenn die Funktion aktiv ist.

Informationen zu diesem Vorgang

Weitere Informationen zur Aktivierung dieser Funktionen und dazu, wie die Ausgabe an SMF und GTF übertragen wird, finden Sie in [Befehle an Db2 mithilfe der Transaktion DSNB absetzen](#) und [GTF für Abrechnung, Statistik und Optimierung von Db2 starten](#).

Daten, die sich auf den Adressraum der Systemservices beziehen, werden als SMF-IFCID-0001-Datensätze (SMF – Instrumentation Facility Component Identifier) geschrieben. Daten, die sich auf den Adressraum der Datenbankservices beziehen, werden als SMF-IFCID-0002-Datensätze geschrieben. Eine Beschreibung dieser Datensätze finden Sie in [Db2-Leistung verwalten in der Produktdokumentation zu Db2 for z/OS](#).

Diese Statistikdaten sind für die Optimierung des Db2-Subsystems nützlich, da Sie die Aktivität aller Subsysteme widerspiegeln, die mit Db2 verbunden sind.

Die Interpretation dieser Daten ist schwierig, wenn mehrere Subsysteme mit Db2 verbunden sind (d. h., sowohl CICS als auch TSO). Die beim Ausführen der CICS-Db2-Anschlussfunktion in einer kontrollierten Umgebung (d. h., nur CICS ist als Subsystem verbunden oder die Aktivität von TSO ist eingeschränkt) erhaltenen Zählungen können sehr nützlich sein.

In der [Db2-Leistung verwalten in der Produktdokumentation zu Db2 for z/OS](#) werden ausgehend von Db2 die zu den Adressräumen der Datenbank- und Systemservices gemeldeten Statistikdaten dargestellt und analysiert. Darin enthalten ist eine reduzierte Version des Statistikberichts. Sie können diesen Bericht zum Überwachen durchschnittlicher CICS-Transaktionen verwenden. In [Abbildung 32 auf Seite 129](#) sehen Sie einen kleinen Teil des Berichts, der von DB2PM bereitgestellt wird. Weitere Informationen zu diesen Berichten finden Sie in [Db2-Leistung verwalten in der Produktdokumentation zu Db2 for z/OS](#).

LOCATION: DSN710P2
 GROUP: DSN710P2
 MEMBER: DF2D
 SUBSYSTEM: DF2D
 DB2 VERSION: V7

DB2 PERFORMANCE MONITOR (V7)
 STATISTICS REPORT - LONG

SCOPE: MEMBER

SQL DML	QUANTITY	/SECOND	/THREAD	/COMMIT
-----	-----	-----	-----	-----
SELECT	1.00	0.00	0.05	0.02
INSERT	9.00	0.00	0.41	0.21
UPDATE	0.00	0.00	0.00	0.00
DELETE	0.00	0.00	0.00	0.00
PREPARE	17.00	0.00	0.77	0.40
DESCRIBE	34.00	0.00	1.55	0.79
DESCRIBE TABLE	0.00	0.00	0.00	0.00
OPEN	31.00	0.00	1.41	0.72
CLOSE	26.00	0.00	1.18	0.60
FETCH	827.00	0.00	37.59	19.23
TOTAL	945.00	0.00	42.95	21.98

SQL DCL	QUANTITY	/SECOND	/THREAD	/COMMIT
-----	-----	-----	-----	-----
LOCK TABLE	0.00	0.00	0.00	0.00
GRANT	0.00	0.00	0.00	0.00
REVOKE	0.00	0.00	0.00	0.00
SET HOST VARIABLE	0.00	0.00	0.00	0.00
SET CURRENT SQLID	0.00	0.00	0.00	0.00
SET CURRENT DEGREE	0.00	0.00	0.00	0.00
SET CURRENT RULES	0.00	0.00	0.00	0.00
SET CURRENT PATH	0.00	0.00	0.00	0.00
SET CURRENT PRECISION	0.00	0.00	0.00	0.00
CONNECT TYPE 1	0.00	0.00	0.00	0.00
CONNECT TYPE 2	29.00	0.00	1.32	0.67
RELEASE	0.00	0.00	0.00	0.00
SET CONNECTION	0.00	0.00	0.00	0.00
ASSOCIATE LOCATORS	0.00	0.00	0.00	0.00
ALLOCATE CURSOR	0.00	0.00	0.00	0.00
HOLD LOCATOR	0.00	0.00	0.00	0.00
FREE LOCATOR	0.00	0.00	0.00	0.00
TOTAL	29.00	0.00	1.32	0.67

SUBSYSTEM SERVICES	QUANTITY	/SECOND	/THREAD	/COMMIT
-----	-----	-----	-----	-----
IDENTIFY	23.00	0.00	1.05	0.53
CREATE THREAD	22.00	0.00	1.00	0.51
SIGNON	39.00	0.00	1.77	0.91
TERMINATE	57.00	0.00	2.59	1.33
ROLLBACK	8.00	0.00	0.36	0.19
COMMIT PHASE 1	0.00	0.00	0.00	0.00
COMMIT PHASE 2	0.00	0.00	0.00	0.00
READ ONLY COMMIT	0.00	0.00	0.00	0.00
UNITS OF RECOVERY INDOUBT	0.00	0.00	0.00	0.00
UNITS OF REC.INDBT RESOLVED	0.00	0.00	0.00	0.00
SYNCHS(SINGLE PHASE COMMIT)	35.00	0.00	1.59	0.81
QUEUED AT CREATE THREAD	0.00	0.00	0.00	0.00
SUBSYSTEM ALLIED MEMORY EOT	0.00	0.00	0.00	0.00
SUBSYSTEM ALLIED MEMORY EOM	0.00	0.00	0.00	0.00
SYSTEM EVENT CHECKPOINT	3.00	0.00	0.14	0.07

CPU TIMES	TCB TIME	SRB TIME	TOTAL TIME	/THREAD	/COMMIT
-----	-----	-----	-----	-----	-----
SYSTEM SERVICES ADDRESS SPACE	17:40.602755	1:09.182200	18:49.784954	51.353862	26.274069
DATABASE SERVICES ADDRESS SPACE	6.100449	11.626277	17.726726	0.805760	0.412249
IRLM	0.051894	2:43.867972	2:43.919867	7.450903	3.812090
DDF ADDRESS SPACE	1.195607	0.212343	1.407950	0.063998	0.032743
TOTAL	17:47.950705	4:04.888792	21:52.839497	59.674523	30.531151

Abbildung 32. Statistikbericht von DB2PM – Beispiel

In Abbildung 32 auf Seite 129 sind Informationen zu Folgendem enthalten:

- **SQL DML.** Diese Informationen können verwendet werden, um die ausgegebenen SQL-Anforderungen zu überwachen.

- *SQL DCL*. Diese Informationen können verwendet werden, um zu überprüfen, ob die Anwendung LOCK-Anweisungen verwendet.
- *Subsystem Services*. Dieser Abschnitt enthält Informationen zur Threadverwendung und zur Anmeldeaktivität. Aus Leistungsgründen wird die Threadwiederverwendung in CICS-Umgebungen empfohlen, um den Systemaufwand zu vermeiden, der entsteht, wenn für jede CICS-Transaktion ein Thread erstellt werden muss.

Weitere hilfreiche Informationen in den Statistikberichten, die in [Abbildung 32 auf Seite 129](#) nicht angezeigt werden, sind:

- *Locking* kann verwendet werden, um die Anzahl von Zeitlimitüberschreitungen und Deadlocks zu überwachen. Weitere Informationen zu Deadlocks finden Sie in [Deadlocks in der CICS-Db2-Umgebung bearbeiten](#).
- *Buffer Pool* bietet Informationen zu Folgendem:
 - Die Anzahl der geöffneten Datensätze (Dateien geöffnet)
 - Die Anzahl der abgerufenen Seiten (GETPAGE-Anforderungen)
 - Die Anzahl der Ein-/Ausgaben (Leseoperationen und E/A-Schreiboperationen)

Diese Statistikdaten weisen keine Prüfpunkte auf und werden bei Neustarts nicht von Db2 beibehalten.

Db2 mithilfe der Db2-Abrechnungsfunktion überwachen

Die Ausgabe der Db2-Abrechnungsfunktion für eine einzelne Transaktion kann zu Überwachungs- und Optimierungszwecken verwendet werden.

Informationen zu diesem Vorgang

Informationen zur Verwendung der Db2-Abrechnungsfunktion finden Sie in [„Db2-Abrechnungsberichte“ auf Seite 134](#).

Db2 mithilfe der Db2-Leistungsfunktion überwachen

Der Db2-Leistungsfunktionstrace bietet detaillierte Informationen zum Ablauf der Steuerung innerhalb von Db2.

Informationen zu diesem Vorgang

Obwohl der Hauptzweck dieses Trace das Bereitstellen von Debugging-Informationen ist, kann er wegen der zu jedem Eintrag bereitgestellten Ablaufsteuerungsdaten auch als Überwachungstool verwendet werden.

Wegen des hohen Ressourcenverbrauchs sollte der Db2-Leistungstrace nur in besonderen Fällen verwendet werden, in denen es schwierig ist, zur Überwachung von Db2-orientierten Transaktionen ein anderes Tool zu verwenden.

Auch in einem solchen Fall sollten nur die benötigten Klassen des Leistungstrace gestartet werden und dies auch nur für einen begrenzten Zeitraum und für ausschließlich solche Transaktionen, die sorgfältig überwacht werden müssen.

CICS-System in einer CICS-Db2-Umgebung überwachen

Sie können die Aktivität der CICS-Db2-Tasksteuerblöcke mithilfe der Statistik des CICS-Dispatchers überwachen.

Informationen zu diesem Vorgang

Weitere Informationen zur Verwendung der vom CICS-Dispatcher bereitgestellten Daten finden Sie in [CICS-Dispatcher: Leistung und Optimierung](#).

Die Daten im Abschnitt mit der Statistik des Dispatchers geben beispielsweise die kumulierte Zeit für jeden der CICS-Tasksteuerblöcke an. Das Feld 'Accum/TCB' gibt die gesamte Prozessorzeit an, die von dem

entsprechenden Tasksteuerblock verwendet wurde. Weitere Informationen zur Statistik des CICS-Dispatchers finden Sie in [Statistik zur Dispatcherdomäne](#).

Sie können den Wert der insgesamt vom CICS-Adressraum verwendeten Prozessorzeit aus den RMF Monitor II-Berichten abrufen. Diese Zeit ist in der Regel größer als die Summe aller Tasksteuerblöcke von CICS-Tasks.

Die Differenz zwischen der von RMF berichteten Prozessorzeit und der Summe der CICS-Tasksteuerblöcke im Statistikbericht des CICS-Dispatchers ist die Prozessorzeit, die von allen übrigen Subtask-Tasksteuerblöcken verwendet wurde. Die Subtasks werden für Folgendes verwendet:

- In den DBCTL-Threads verbrauchte Prozessorzeit
- Von den IBM MQ-Threads verbrauchte Prozessorzeit

Dies sind globale Leistungsberichte, die Sie dabei unterstützen können, den Anteil Ihrer Prozessorzeit zu ermitteln, der von CICS verbraucht wird.

Abrechnung in einer CICS-Db2-Umgebung: Übersicht

Über die Abrechnung in einer CICS-Db2-Umgebung können die im System ausgeführten Transaktionen analysiert werden und die Kosten für die Gesamtmenge der Ressourcen, die für eine bestimmte Gruppe von Transaktionen verbraucht wurde, kann einer klar strukturierten Gruppe von Benutzern zugeordnet werden. Bei dem Benutzer kann es sich um einen realen Benutzer, eine Gruppe von Benutzern, um Transaktionen oder um einen beliebigen anderen Ausdruck für die Einheit handeln, der Ressourcen zugeordnet werden müssen.

In der Regel gehören zu den Verbrauchseinheiten der Prozessor, die Ein-/Ausgabe und der Hauptspeicher in einem gewichteten Verhältnis. Eine typische CICS-Transaktion, die auf Db2 zugreift, verbraucht Ressourcen im Betriebssystem, dem CICS-System, im Anwendungscode und in den Db2-Adressräumen. Jede dieser Komponenten kann Daten erzeugen, die als Eingabe für den Abrechnungsprozess verwendet werden können. Sie können die Ausgabe der verschiedenen Quellen kombinieren, um ein vollständiges Bild der Ressourcennutzung einer Transaktion zu erstellen.

Eine typische Anforderung an ein Abrechnungsverfahren besteht darin, dass die berechneten Ergebnisse reproduzierbar sind; dies bedeutet, dass die Kosten für eine Transaktion, die auf eine Gruppe von Daten zugreift, bei jeder Ausführung der Transaktion immer gleich hoch sind. In den meisten Fällen bedeutet dies, dass die Eingabedaten für den Abrechnungsprozess ebenfalls wiederholbar sein müssen.

Bei der Planung der Abrechnungsstrategie für Ihre CICS-Db2-Umgebung müssen Sie Folgendes ausführen:

- Legen Sie fest, welche Typen von Db2-Abrechnungsdaten in Ihrem Abrechnungsprozess verwendet werden sollen (Prozessorbelegung, Ein-/Ausgabe, Aufrufe usw.). In „[Abrechnungsdaten, die von der Db2-Abrechnungsfunktion bereitgestellt werden](#)“ auf Seite 132 finden Sie Informationen zu den Abrechnungsdaten, die Sie über die Db2-Abrechnungsfunktion abrufen können.
- Legen Sie fest, wie die Daten aus dem Db2-Abrechnungsdatensatz der einzelnen Transaktionen mit den CICS-Leistungsklassendaten für diese Transaktion in Beziehung zu setzen sind, um ein vollständiges Bild der Ressourcennutzung für die Transaktion zu erstellen. In „[Db2-Abrechnungsdatensätze und CICS-Leistungsklassendatensätze in Beziehung zueinander setzen](#)“ auf Seite 135 finden Sie Informationen dazu, wie Sie die beiden Typen von Daten abgleichen können.
- Entscheiden Sie, ob Sie die CICS-Leistungsdatsätze und die Db2-Abrechnungsdatensätze der einzelnen Transaktionen in Beziehung zu dem bestimmten Benutzer setzen möchten oder ob Sie eine Reihe von Modelltransaktionen definieren und kalibrieren wollen, diese Transaktionen in einer kontrollierten Umgebung messen und nur die Anzahl der Modelltransaktionen zählen wollen, die von den einzelnen Benutzern ausgeführt werden. In „[Strategien für den Abgleich von Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendatensätzen und für die Zuordnung der Kosten zum Benutzer](#)“ auf Seite 138 finden Sie Vorschläge, welche Methode in welcher Situation am geeignetsten ist.

Wenn Sie sich für die Prozessorbelegung als Basis Ihrer Abrechnung entscheiden, finden Sie in „[Abrechnung zur Prozessorbelegung in einer CICS-Db2-Umgebung](#)“ auf Seite 141 weitere Informationen zu den

verschiedenen Klassen von Prozessorzeit, die in den Db2-Abrechnungsdatensätzen berichtet werden sowie dazu, wie die gesamte von einer Transaktion verbrauchte Prozessorzeit berechnet wird.

Abrechnungsdaten, die von der Db2-Abrechnungsfunktion bereitgestellt werden

Die Db2-Abrechnungsfunktion stellt detaillierte Statistikdaten zur Verwendung von Db2-Ressourcen durch CICS-Transaktionen bereit. Sie können Db2-Abrechnungsdatensätze als Basis für die Abrechnung und die Optimierung von Db2-Ressourcen verwenden, die von CICS-Transaktionen verwendet werden.

Db2 stellt Abrechnungsdaten auf der Basis von Berechtigungs-IDs innerhalb von Threads bereit. Bei Anforderung erfasst die Abrechnungsfunktion diese Daten und überträgt sie an SMF (Systemverwaltungsfunktion) und/oder GTF (Allgemeine Tracefunktion), sobald der Thread beendet ist oder wenn die Berechtigungs-ID geändert wird. Informationen zur Aktivierung der Db2-Abrechnungsfunktion und dazu, wie die Ausgabe an SMF und GTF übertragen wird, finden Sie in [SMF für Abrechnung, Statistik und Optimierung von Db2 starten](#) und [GTF für Abrechnung, Statistik und Optimierung von Db2 starten](#). In [Db2-Leistung verwalten in der Produktdokumentation zu Db2 for z/OS](#) finden Sie Informationen zur allgemeinen Struktur von Db2-SMF und -GTF-Datensätzen.

In den einzelnen Db2-Abrechnungsdatensätzen, die an SMF und GTF geschrieben werden, ist jeweils ein Identifikationsabschnitt enthalten, in dem eine Reihe von Schlüsseln bereitgestellt wird, nach denen die Daten sortiert und zusammengefasst werden können. Dazu gehören die Berechtigungs-ID, die Transaktions-ID, der Planname und der Paketname.

Das Lizenzprogramm Db2 Performance Monitor (DB2PM) stellt Abrechnungsberichte bereit, deren Grundlage die Db2-Abrechnungsdatensätze sind.

Datentypen in Db2-Abrechnungsdatensätzen

Dieser Abschnitt enthält eine Übersicht über die verschiedenen Datentypen in Db2-Abrechnungsdatensätzen.

Wenn Sie ein Kalkulationsverfahren definieren möchten, das auf Db2-Abrechnungsdatensätzen basiert, haben Sie mehrere Möglichkeiten.

- Wenn eine Kombination aus Wiederholgenauigkeit und einem angemessenen Ausdruck der Komplexität von Transaktionen eine hohe Priorität hat, so sind die Prozessorbelegung, die GETPAGE-Anzahl und die Anzahl der festgelegten Schreibabsichten (Set Write Intent) gute Kandidaten.
- Wenn der Zweck des Abrechnungsprozesses die Analyse des Verhaltens der CICS-Transaktionen ist, können alle in den Db2-Abrechnungsdatensätzen enthaltenen Informationen verwendet werden.

In [Db2-Leistung verwalten in der Produktdokumentation zu Db2 for z/OS](#) finden Sie Details zu den einzelnen Feldern im Db2-Abrechnungsdatensatz.

Prozessorbelegung

Die im Db2-Abrechnungsdatensatz angegebenen Informationen zur Prozessorbelegung zeigen in der Regel den größten Teil der insgesamt für die SQL-Aufrufe verwendeten Prozessorzeit.

Die Db2-Statistikdatensätze berichten die in den Db2-Adressräumen verwendete Prozessorzeit, die den einzelnen Threads nicht direkt zugeordnet werden konnte.

Ziehen Sie in Betracht, die in den Db2-Statistikdatensätzen berichtete Prozessorzeit proportional zwischen allen Benutzern des Db2-Subsystems (Transaktionen, Stapelverarbeitungsprogramme, TSO-Benutzer) aufzuteilen.

Der Umfang der in den Db2-Abrechnungsdatensätzen berichteten Prozessorzeit ist (für dieselbe Arbeit) im Zeitverlauf relativ wiederholbar.

In [„Abrechnung zur Prozessorbelegung in einer CICS-Db2-Umgebung“](#) auf Seite 141 finden Sie weitere Informationen zur Berichterstellung für die Prozessorbelegung in einer CICS Db2-Umgebung.

Ein-/Ausgabe

In einem Db2-System kann die Ein-/Ausgabe in fünf Typen kategorisiert werden.

- Ein-/Ausgabe synchroner Lesevorgänge
- Sequenzieller Vorablesezugriff (asynchrone Lesevorgänge)
- Asynchrone Schreibvorgänge
- Lesevorgänge in EDM-Pools (Datenbankbeschreibungen und Plansegmente)
- Ein-/Ausgabe für Protokolle (hauptsächlich Schreibvorgänge).

Von diesen fünf Ein-/Ausgabe-Typen wird nur die Ein-/Ausgabe synchroner Lesevorgänge im Db2-Abrechnungsdatensatz erfasst.

Die Anzahl der sequenziellen Vorablesezugriffe wird auch berichtet, aber die Anzahl der Leseanforderungen ist ungleich der Anzahl von Ein-/Ausgabeoperationen.

Keiner der Ein-/Ausgabetypen sollte als im Zeitverlauf wiederholbar betrachtet werden. Sie alle hängen von den Puffergrößen und der Auslastungsaktivität ab.

Db2 erkennt nicht, dass Caches verwendet werden. Dies bedeutet, dass Db2 das Auftreten einer Ein-/Ausgabe berichtet, auch wenn der Cachepuffer die Anforderung erledigt.

GETPAGE (getpage-Operation)

Die getpage-Operation (GETPAGE) zeigt die Anzahl an, wie oft Db2 vom Puffermanager eine Seite angefordert hat.

GETPAGE stellt im Db2-Abrechnungsdatensatz eine Anzahl dar, die im Zeitverlauf für dieselbe Transaktion recht konstant ist. Es zeigt die Anzahl an, wie oft Db2 vom Puffermanager eine Seite angefordert hat. Jedes Mal, wenn Db2 Daten in einer Seite lesen oder schreiben soll, muss die Seite verfügbar sein und für die Seite wird mindestens eine getpage-Operation (GETPAGE) gezählt. Dies gilt sowohl für Index- als auch für Datenseiten. Wie oft der GETPAGE-Zähler für eine bestimmte, mehrmals verwendete Seite erhöht wird, hängt vom ausgewählten Zugriffspfad ab. Für dieselbe Transaktion, die auf dieselben Daten zugreift, bleibt die Anzahl von getpage-Operationen (GETPAGE) im Zeitverlauf jedoch recht konstant; allerdings kann sich der Algorithmus der getpage-Operation (GETPAGE) je nach Db2-Release unterscheiden.

Wenn der Pufferpool die angeforderte Seite enthält, findet keine Ein-/Ausgabe statt. Wenn die Seite nicht im Puffer vorhanden ist, fordert der Puffermanager die Seite vom Datenträgermanager an, und die Ein-/Ausgabe erfolgt.

Die Anzahl von getpage-Operationen (GETPAGE) ist daher ein Indikator für die in Db2 für die Ausführung von SQL-Anforderungen erforderliche Aktivität.

Schreibabsichten

Die Anzahl der festgelegten Schreibabsichten (Set Write Intents) ist im Feld QBACSWs des Db2-Abrechnungsdatensatzes enthalten.

Die Anzahl der festgelegten Schreibabsichten steht nicht in Bezug zur tatsächlichen Anzahl der Ein-/Ausgabe-Operationen für Schreibvorgänge aus den Pufferpools, stattdessen stellt sie die Anzahl dar, wie oft eine Seite für die Aktualisierung markiert wurde. Auch in einer schreibgeschützten Transaktion kann diese Anzahl vorhanden sein, da die beabsichtigten Schreibvorgänge in den temporären Arbeitsdateien, die in einer Db2-Sortierung verwendet werden, auch gezählt werden.

Der typische Fall ist, dass die Anzahl der festgelegten Schreibabsichten sehr viel höher als die Anzahl der Ein-/Ausgaben für Schreibvorgänge ist. Das Verhältnis zwischen diesen beiden Zahlen hängt von der Größe des Pufferpools und der Auslastung ab. Es ist kein guter Messwert für die Ein-/Ausgabeaktivität für Schreibvorgänge, gibt aber die Komplexität der Transaktionen an.

SQL-Aufrufaktivität

Die Anzahl und der Typ der in einer Transaktion ausgeführten SQL-Aufrufe werden im Db2-Abrechnungsdatensatz berichtet.

Die Werte der Aktivität für SQL-Aufrufe sind wiederholbar, es sei denn, es sind viele verschiedene Pfade durch ein komplexes Programm möglich oder der Zugriffspfad wird geändert. Der ausgewählte Zugriffspfad kann im Zeitverlauf geändert werden, beispielsweise durch Hinzufügen eines Index.

Ein bestimmter SQL-Aufruf kann einfach oder komplex sein; dies hängt von Faktoren ab wie beispielsweise dem ausgewählten Zugriffspfad und der Anzahl der Tabellen und Zeilen, die bei den Anforderungen beteiligt sind.

Die Anzahl der getpage-Operationen (GETPAGE) ist in den meisten Fällen ein präziserer Indikator für die Db2-Aktivität als die Anzahl der verschiedenen SQL-Aufrufe.

Transaktionsvorkommen

Eine einfache Art der Abrechnung ist es, die Anzahl und den Typ der ausgeführten Transaktionen zu verfolgen.

Speicher

Der Db2-Abrechnungsdatensatz enthält keine Informationen zu tatsächlichem oder virtuellem Speicher, der in Beziehung zu der Ausführung der Transaktionen steht. Einer der Zwecke des Db2-Subsystems ist die Optimierung der Speicherverwendung. Diese Optimierung wird auf der Ebene von Db2 und nicht auf der Transaktionsebene vorgenommen.

Eine Transaktion, die Db2-Services anfordert, verwendet Speicher, der an mehreren Positionen vorhanden ist. Die wichtigsten Positionen dabei sind der Thread, der EDM-Pool und die Pufferpools.

Da im Db2-Abrechnungsdatensatz keine Informationen zur Speicherbelegung enthalten sind und da die Speicherbelegung auf der Subsystemebene optimiert wird, ist es schwierig, in einer Db2-Umgebung Nachweise zum Speicher zu erbringen.

Db2-Abrechnungsberichte

Das Lizenzprogramm Db2 Performance Monitor (DB2PM) stellt Abrechnungsberichte bereit, deren Grundlage die Db2-Abrechnungsdatensätze sind.

In Abbildung 33 auf Seite 134 und Abbildung 34 auf Seite 135 werden Beispiele langer und kurzer Abrechnungsberichte für eine CICS-Transaktion gezeigt, die auf Db2Db2-Ressourcen zugreift.

LOCATION: DSN710P2 GROUP: DSN710P2 MEMBER: DF2D SUBSYSTEM: DF2D DB2 VERSION: V7				DB2 PERFORMANCE MONITOR (V7) ACCOUNTING REPORT - LONG ORDER: PRIMAUTH-PLANNAME SCOPE: MEMBER				PAGE: 1-1 REQUESTED FROM: NOT SPECIFIED TO: NOT SPECIFIED INTERVAL FROM: 11/05/01 10:42:31.25 TO: 11/05/01 10:51:03.70			
PRIMAUTH: JTILLI1 PLANNAME: DSNJDBC											
ELAPSED TIME DISTRIBUTION						CLASS 2 TIME DISTRIBUTION					
-----						-----					
APPL	=====> 98%					CPU	=> 3%				
DB2						NOTACC	=> 2%				
SUSP	=> 2%					SUSP	=====> 95%				
-----						-----					
AVERAGE	APPL (CL.1)	DB2 (CL.2)	IFI (CL.5)	CLASS 3 SUSPENSIONS	AVERAGE TIME	AV.EVENT	HIGHLIGHTS				

ELAPSED TIME	25.435644	0.504442	N/P	LOCK/LATCH(DB2+IRLM)	0.000000	0.00	#OCCURRENCES	:	2		
NONNESTED	25.435644	0.504442	N/A	SYNCHRON. I/O	0.085908	6.50	#ALLIEDS	:	2		
STORED PROC	0.000000	0.000000	N/A	DATABASE I/O	0.085908	6.50	#ALLIEDS DISTRIB:	:	0		
UDF	0.000000	0.000000	N/A	LOG WRITE I/O	0.000000	0.00	#DBATS	:	0		
TRIGGER	0.000000	0.000000	N/A	OTHER READ I/O	0.042337	1.00	#DBATS DISTRIB.	:	0		
				OTHER WRTE I/O	0.000000	0.00	#NO PROGRAM DATA:	:	2		
CPU TIME	0.016663	0.015404	N/P	SER.TASK SWITCH	0.352902	4.00	#NORMAL TERMINAT:	:	2		
AGENT	0.016663	0.015404	N/A	UPDATE COMMIT	0.000000	0.00	#ABNORMAL TERMIN:	:	0		
NONNESTED	0.016663	0.015404	N/P	OPEN/CLOSE	0.206822	1.50	#CP/X PARALLEL.	:	0		
STORED PRC	0.000000	0.000000	N/A	SYSLGRNG REC	0.024259	1.00	#IO PARALLELISM	:	0		
UDF	0.000000	0.000000	N/A	EXT/DEL/DEF	0.121821	1.50	#INCREMENT. BIND:	:	0		
TRIGGER	0.000000	0.000000	N/A	OTHER SERVICE	0.000000	0.00	#COMMITTS	:	3		
PAR.TASKS	0.000000	0.000000	N/A	ARC.LOG (QUIES)	0.000000	0.00	#ROLLBACKS	:	0		
				ARC.LOG READ	0.000000	0.00	#SVPT REQUESTS	:	0		
SUSPEND TIME	N/A	0.481147	N/A	STOR.PRC SCHED	0.000000	0.00	#SVPT RELEASE	:	0		
AGENT	N/A	0.481147	N/A	UDF SCHEDULE	0.000000	0.00	#SVPT ROLLBACK	:	0		
PAR.TASKS	N/A	0.000000	N/A	DRAIN LOCK	0.000000	0.00	MAX SQL CASC LVL:	:	0		
				CLAIM RELEASE	0.000000	0.00	UPDATE/COMMIT	:	0.00		
NOT ACCOUNT.	N/A	0.007891	N/A	PAGE LATCH	0.000000	0.00	SYNCH I/O AVG.	:	0.013217		
DB2 ENT/EXIT	N/A	35.00	N/A	NOTIFY MSGS	0.000000	0.00					
EN/EX-STPROC	N/A	0.00	N/A	GLOBAL CONT.	0.000000	0.00					
EN/EX-UDF	N/A	0.00	N/A	FORCE-AT-COMMIT	0.000000	0.00					
DCAPT.DESCR.	N/A	N/A	N/P	ASYNCH IXL REQUESTS	0.000000	0.00					
LOG EXTRACT.	N/A	N/A	N/P	TOTAL CLASS 3	0.481147	11.50					

Abbildung 33. Ausführlicher Abrechnungsbericht für eine CICS-Transaktion, die auf Db2-Ressourcen zugreift

LOCATION: DSN710P2		DB2 PERFORMANCE MONITOR (V7)							PAGE: 1-1					
GROUP: DSN710P2		ACCOUNTING REPORT - SHORT							REQUESTED FROM: NOT SPECIFIED					
MEMBER: DF2D		ORDER: PLANNAME SCOPE: MEMBER							TO: NOT SPECIFIED					
SUBSYSTEM: DF2D									INTERVAL FROM: 11/05/01 10:42:31.25					
DB2 VERSION: V7									TO: 11/05/01 10:50:14.53					
PLANNAME	#OCCURS #DISTR	#ROLLBK #COMMIT	SELECTS FETCHES	INSERTS OPENS	UPDATES CLOSES	DELETES PREPARE	CLASS1 CLASS1	EL.TIME CPUTIME	CLASS2 CLASS2	EL.TIME CPUTIME	GETPAGES BUF.UPDT	SYN.READ TOT.PREF	LOCK #LOCKOUT	SUS
DSNJDBC	1 0	0 2	0.00 4.00	0.00 2.00	0.00 2.00	0.00 2.00		1.706541 0.027471	1.003194 0.025984	249.00 0.00	13.00 5.00	0.00 0		
TESTP05	2 0	0 2	0.00 1.50	0.50 0.50	0.00 0.00	0.00 0.00		33.283119 0.001908	0.215656 0.001389	7.00 1.00	0.00 0.00	0.00 0		
*** GRAND TOTAL ***	3 0	0 4	0.00 2.33	0.33 1.00	0.00 0.67	0.00 0.67		22.757593 0.010429	0.478169 0.009587	87.67 0.67	4.33 1.67	0.00 0		

Abbildung 34. Kurzer Abrechnungsbericht für eine CICS-Transaktion, die auf Db2-Ressourcen zugreift

Db2-Abrechnungsdatensätze und CICS-Leistungsklassendatensätze in Beziehung zueinander setzen

Um geordnet nach CICS-Transaktion ein vollständiges Bild der Ressourcenverwendung sowohl im CICS- als auch im Db2-Adressraum zu erhalten, müssen Sie die Daten in den Db2-Abrechnungsdatensätzen mit den Daten in den CICS-Leistungsklassendatensätzen abgleichen.

Informationen zu diesem Vorgang

Wenn Sie über ein vollständiges Bild der Ressourcenverwendung verfügen, können Sie die von den einzelnen Transaktionen verwendeten Ressourcen analysieren und Sie können auch dem Benutzer die Kosten für die Gesamtmenge der Ressourcen zuordnen, die für eine bestimmte Gruppe von Transaktionen verbraucht wurden, wobei alle von Ihnen in das Kalkulationsverfahren eingeschlossenen Datentypen verwendet werden können.

Wenn Sie die Ressourcenverwendung nicht wegen einer Abrechnung, sondern wegen einer Leistungsanalyse untersuchen, müssen Sie stets die Db2-Abrechnungsdatensätze und die CICS-Leistungsklassendatensätze miteinander abgleichen. Wenn Sie die Ressourcenverwendung jedoch aus Abrechnungsgründen untersuchen, müssen Sie die Db2-Abrechnungsdatensätze möglicherweise nicht mit den CICS-Leistungsklassendatensätzen abgleichen.

Sie müssen die Datensätze nicht miteinander abgleichen, wenn Sie als Basis für Ihre Abrechnung den Verbrauch an Prozessorzeit ausgewählt haben und Sie nicht die Db2-Sysplex-Abfrageparallelität (parallele Abfrage) verwenden. Wird die parallele Abfrage nicht verwendet, wird die in Db2 verbrauchte Prozessorzeit von den CICS-Leistungsklassendatensätzen und auch den Db2-Abrechnungsdatensätzen berichtet, sodass der CICS-Leistungsklassendatensatz für eine Transaktion Ihnen alle zur Prozessorzeit erforderlichen Informationen liefert, die Sie benötigen, um dem Benutzer die Kosten der für diese Transaktion verbrauchten Ressourcen zuzuordnen. Wenn diese Situation auf Sie zutrifft, überspringen Sie den restlichen Abschnitt und lesen Sie stattdessen die Abschnitte „[Abrechnung zur Prozessorbelegung in einer CICS-Db2-Umgebung](#)“ auf Seite 141 (um zu verstehen, wie der Verbrauch an Prozessorzeit berichtet wird) und „[CICS- und Db2-Prozessorzeit für Db2 berechnen](#)“ auf Seite 148 (Informationen dazu, wie die für Sie verfügbaren Informationen zur Prozessorzeit zu verwenden sind).

Wenn Sie ausgewählt haben, statt der verbrauchten Prozessorzeit Daten als Basis für Ihre Abrechnung zu verwenden bzw. wenn Sie beide Möglichkeiten nutzen wollen, lesen Sie die zu diesem Abschnitt zugehörigen Konzepte, um zu erfahren, wie Db2-Abrechnungsdatensätze und CICS-Leistungsklassendatensätze miteinander abgeglichen werden. Wenn Sie für Ihre Abrechnung die verbrauchte Prozessorzeit zugrunde legen, können Sie den Abschnitt „[Abrechnung zur Prozessorbelegung in einer CICS-Db2-Umgebung](#)“ auf Seite 141 lesen, um zu erfahren, wie der Verbrauch der Prozessorzeit aus den miteinander abgeglichenen Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendatensätzen berechnet werden kann.

Probleme, die beim Abgleich von Db2-Abrechnungsdatensätzen und CICS-Leistungsdaten­sätzen auftreten können

Da für CICS und Db2 unterschiedliche Abrechnungsanforderungen bestehen, ist es nicht immer einfach, Db2-Abrechnungsdatensätze und CICS-Leistungsklassendaten­sätze abzugleichen.

Beim Abgleich dieser Datensätze gibt es zwei Hauptprobleme:

1. Es ist nicht notwendigerweise eine Eins-zu-eins-Beziehung zwischen CICS-Leistungsklassendaten­sätzen und den Db2-Abrechnungsdatensätzen vorhanden. Ein Db2-Abrechnungsdatensatz kann Informationen zu einer CICS-Transaktion, mehreren CICS-Transaktionen oder einem Teil einer CICS-Transaktion enthalten.
2. Die Db2-Abrechnungsdatensätze weisen kein Feld auf, das genau mit den entsprechenden CICS-Leistungsdaten­sätzen übereinstimmt.

Um dem Benutzer Kosten für Ressourcen zuordnen zu können, ist es möglich, vom Standpunkt von Db2 aus jedem Benutzer eine andere Berechtigungs-ID zu geben, indem für den DB2ENTRY- oder DB2CONN-Parameter AUTHTYPE die Einstellung OPID, USERID, GROUP oder TERM angegeben wird. In diesem Fall wird ein Db2-Abrechnungsdatensatz generiert, der nur Daten für die Berechtigungs-ID enthält. Sie können dann alle Db2-Abrechnungsdatensätze über die Berechtigungs-ID zusammen erfassen und dem Benutzer die Kosten für die verbrauchten Ressourcen direkt zuordnen. Wenden Sie diese Methode an, bedeutet dies, dass Sie die Db2-Abrechnungsdatensätze nicht mit den CICS-Leistungsklassendaten­sätzen abgleichen müssen. Vom Standpunkt der Benutzerfreundlichkeit und Leistung aus gesehen, ist die Verwendung von OPID, USERID, GROUP und TERM wegen der in [Benutzerzugriff auf Pläne steuern](#) beschriebenen Probleme keine attraktive Lösung. Bei großen Netzen kann die Angabe dieser Berechtigungs-IDs die Wartung komplizieren und zu Leistungseinbußen führen. Es empfiehlt sich, bei der Planung für die Verwendung der Berechtigungs-IDs die Leistung zu berücksichtigen und dem Endbenutzer Db2-Abrechnungsdatensätze zuzuweisen, indem Sie sie mit CICS-Leistungsklassendaten­sätzen abgleichen.

In diesem Abschnitt erfahren Sie Folgendes:

- Wie die Beziehung zwischen Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendaten­sätzen unkomplizierter gestaltet werden kann.
- Welche Informationen in einem Db2-Abrechnungsdatensatz zum Ermitteln der entsprechenden CICS-Leistungsklassendaten­sätze verwendet werden können.
- Welche Strategien für den Abgleich von Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendaten­sätzen in vier typischen Szenarios verwendet werden können.

Beziehung zwischen Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendaten­sätzen steuern

Db2 schreibt die zugehörigen Abrechnungsdatensätze standardmäßig zum Zeitpunkt der Threadbeendigung oder bei der Anmeldung einer neuen Berechtigungs-ID, die den Thread wiederverwendet.

Informationen zu diesem Vorgang

Wenn ein Thread von einer Transaktion wiederverwendet wird, die dieselbe CICS-Transaktions-ID und dieselbe Db2-Berechtigungs-ID aufweist wie die vorherige Transaktion, die den Thread verwendet hat, schreibt Db2 an diesem Zeitpunkt keinen Abrechnungsdatensatz. (In [DB2 Berechtigungs-IDs für die CICS-Region und für CICS-Transaktionen bereitstellen](#) finden Sie Informationen zur Beziehung zwischen der CICS-Transaktions-ID und der Db2-Berechtigungs-ID.) Dies bedeutet, dass jeder Db2-Abrechnungsdatensatz für den Thread Informationen zu mehreren CICS-Transaktionen enthalten kann. Wenn verschiedene Typen von CICS-Transaktionen für den Zugriff auf Db2 dieselbe Transaktions-ID verwenden, kann der Db2-Abrechnungsdatensatz darüber hinaus Informationen zu verschiedenen Typen von CICS-Transaktionen enthalten.

Es gibt drei Möglichkeiten, wie Sie Einfluss auf die Beziehung zwischen Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendaten­sätzen nehmen können, um folgende Probleme zu behandeln:

- Sie können Ihre CICS-Anwendungen so entwerfen, dass jede CICS-Transaktions-ID und jede Db2-Berechtigungs-ID stets dieselbe Arbeitseinheit darstellt, die dieselben Ressourcen verbraucht. Dadurch

wird sichergestellt, dass jeder Db2-Abrechnungsdatensatz entweder eine einzelne Arbeitseinheit oder mehrere Vorkommen derselben Arbeitseinheit enthält und dass der Abrechnungsdatensatz keine unterschiedlichen Elemente enthalten wird. Wenn ein solcher Db2-Abrechnungsdatensatz mehrere Elemente enthält, können Sie die verbrauchten Ressourcen zu gleichen Anteilen zwischen ihnen aufteilen, weil die Elemente identisch sind. Es ist jedoch möglicherweise unpraktisch, Anwendungen auf diese Weise zu entwerfen. Dies ist beispielsweise dann der Fall, wenn ein Terminalbenutzer am Terminal ein Menü anzeigt, in dem er verschiedene Optionen (bei denen verschiedene Arbeitseinheiten beteiligt sind) für die nächste Transaktion auswählen kann. Wenn die vorherige Transaktion durch Festlegen des Befehls EXEC CICS RETURN TRANSID(zzzz) für die CICS-Transaktions-ID endete, wird die nächste Transaktion unter der Transaktions-ID 'zzzz' ausgeführt, ohne dass die vom Terminalbenutzer ausgewählte Arbeitseinheit berücksichtigt wird. Diese Anwendung möchten Sie möglicherweise nicht für Abrechnungszwecke neu entwerfen.

- Sie könnten die Wiederverwendung von Threads vermeiden. Dadurch wird sichergestellt, dass der Thread beendet wird und Db2 nach jeder Task einen Abrechnungsdatensatz schreibt, sodass jeder Db2-Abrechnungsdatensatz eine einzelne Task darstellt. Dadurch würden Sie jedoch die beträchtlichen Leistungsvorteile verlieren, die durch die Threadwiederverwendung entstehen. Wenn verschiedene Transaktionstypen für den Zugriff auf Db2 dieselbe Transaktions-ID verwenden würden, könnte sich darüber hinaus jeder Db2-Abrechnungsdatensatz dennoch weiterhin auf eine von mehreren möglichen Tasks beziehen.
- Sie können durch Angeben von ACCOUNTREC(TASK) in der Definition DB2CONN oder DB2ENTRY veranlassen, dass Db2 jedes Mal einen Abrechnungsdatensatz schreibt, wenn eine CICS-Task die Verwendung von Db2-Ressourcen beendet. ACCOUNTREC(TASK) wird empfohlen, nicht ACCOUNTREC(UOW). Dadurch wird sichergestellt, dass mindestens ein identifizierbarer Db2-Abrechnungsdatensatz pro Task vorhanden ist und dass der Db2-Abrechnungsdatensatz nicht mehrere Tasks enthält. Um das Problem zu lösen, dass unterschiedliche Transaktionstypen für den Zugriff auf Db2 dieselbe Transaktions-ID verwenden, übergibt CICS, wenn Sie ACCOUNTREC(TASK) angeben, das zugehörige LU6.2-Token an Db2, sodass es in den Abrechnungsdatensatz eingeschlossen wird. Sie können dieses Token verwenden, um die einzelnen Db2-Abrechnungsdatensätze mit den relevanten CICS-Transaktionen abzugleichen. Die Verwendung von ACCOUNTREC(TASK) bietet in der Regel die praktischste und umfassendste Lösung, um die Beziehung zwischen Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendaten-sätzen zu steuern. Dabei entsteht für jede Transaktion ein Systemaufwand, aber der Nutzen für Abrechnungszwecke ist in der Regel größer als dieser Systemaufwand.

Beachten Sie, dass Db2 auch bei Angabe von ACCOUNTREC(TASK) eine CICS-Task nur solange als eine einzige Task erkennen kann, wie die Task denselben Thread verwendet. Wenn eine Transaktion mehrere Arbeitseinheiten enthält und vorausgesetzt wird, dass sie den Thread am Ende einer Arbeitseinheit freigibt, könnte sie für jede ihrer Arbeitseinheiten einen anderen Thread verwenden. Dies kann bei terminalorientierten Transaktionen auftreten, die mehrere Synchronisationspunkte (Festschreibung oder Rollback) setzen, und auch bei nicht terminalorientierten Transaktionen, wenn im DB2CONN-Objekt NONTERM-REL(YES) festgelegt ist. In diesen Fällen erzeugt Db2 für jede Arbeitseinheit einen Abrechnungsdatensatz, da es die Arbeitseinheiten nicht als einzige Task erkennt. Bei dieser Art von Transaktion kann also jeder Db2-Abrechnungsdatensatz Informationen zu nur einem Teil der Transaktion enthalten und Sie müssen sicherstellen, dass alle für die Transaktion relevanten Db2-Abrechnungsdatensätze angegeben werden.

Daten des Db2-Abrechnungsdatensatzes zum Ermitteln der entsprechenden CICS-Leistungsklassendaten-sätze verwenden

Verwenden Sie Felder aus Db2-Abrechnungsdatensätzen, um die entsprechenden CICSLeistungsdaten-sätze zu ermitteln.

Informationen zu diesem Vorgang

Die Felder in den Db2-Abrechnungsdatensätzen, die Sie verwenden können, lauten wie folgt:

- Das CICS-LU6.2-Token. Wenn Sie im DB2ENTRY-Objekt oder im DB2CONN-Objekt ACCOUNTREC(TASK) oder ACCOUNTREC(UOW) angeben, übergibt CICS das zugehörige LU6.2-Token an Db2, so dass es in Db2-Tracedatensätze eingeschlossen wird. Das Token wird im Korrelationsheader in QWHCTOKN geschrieben. Das Vorhandensein dieses Tokens macht den Abgleich der beiden Sätze wesentlich einfacher.

- Die Korrelations-ID des Threads, die die vier Zeichen umfassende CICS-Transaktions-ID enthält. Wenn Sie ACCOUNTREC(TASK) oder ACCOUNTREC(UOW) nicht angegeben haben, müssen Sie darauf achten, dass der Db2-Abrechnungsdatensatz Informationen zu mehreren Transaktionen enthalten kann, die diese ID verwendet haben. Wenn Sie ACCOUNTREC(TASK) oder ACCOUNTREC(UOW) angegeben haben, enthält der Db2-Abrechnungsdatensatz möglicherweise nur einen Teil einer Transaktion (eine einzelne Arbeitseinheit) und Sie müssen die übrigen Datensätze suchen, die sich auf diese Transaktion beziehen. Wenn unterschiedliche Typen von CICS-Transaktionen dieselbe Transaktions-ID verwenden, ist dieses eine Element nicht ausreichend, um eine positive Identifikation vorzunehmen.
- Das Feld für die Berechtigungs-ID. Wie auch bei der Korrelations-ID des Threads enthält der Db2-Abrechnungsdatensatz möglicherweise Informationen zu mehreren Transaktionen, die diese Berechtigungs-ID verwendet haben, oder er enthält nur einen Teil einer Transaktion. Die Berechtigungs-ID, die von einer CICS-Transaktion verwendet wird, ist durch den Parameter AUTHID oder AUTHTYPE im DB2CONN-Objekt oder im DB2ENTRY-Objekt festgelegt. Wenn unterschiedliche Typen von CICS-Transaktionen dieselbe Berechtigungs-ID verwenden, ist dieses eine Element nicht ausreichend, um eine positive Identifikation vorzunehmen.
- Die Zeitmarkenfelder. Der Startzeitpunkt und der Endzeitpunkt des Threads kann bei der Ermittlung der CICS-Transaktionen helfen, die durch den Db2-Abrechnungsdatensatz abgedeckt werden.

Strategien für den Abgleich von Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendaten­sätzen und für die Zuordnung der Kosten zum Benutzer

Es gibt keinen idealen Weg, um Db2-Abrechnungsdatensätze und CICS-Leistungsklassendaten­sätze abzugleichen. In einigen Fällen ist es möglicherweise nicht möglich, einen korrekten Abgleich auszuführen, da Transaktionen gleichzeitig ausgeführt werden. Für die meisten Situationen sind jedoch Strategien vorhanden, mithilfe derer Sie die zwei Typen von Datensätzen mit angemessener Genauigkeit abgleichen können.

Wenn die in der einzelnen Transaktion verwendeten Ressourcen die Basis für die Abrechnung bilden und Sie die CICS-Leistungsdatsätze und die Db2-Abrechnungsdatensätze miteinander abgeglichen haben, können Sie den Bezug zu einem bestimmten Benutzer herstellen. Alternativ dazu können Sie eine gewisse Anzahl von Modelltransaktionen definieren und kalibrieren, sie anschließend in einer kontrollierten Umgebung messen und nur die Anzahl der von jedem Benutzer ausgeführten Modelltransaktionen zählen.

Die beiden Hauptfaktoren, durch die festgelegt wird, welche Strategien Sie benutzen sollten, lauten wie folgt:

- Ob eine CICS-Transaktions-ID nur einen möglichen Transaktionspfad darstellt (und dadurch stets dieselbe Menge an verbrauchten Ressourcen darstellt) oder ob viele unterschiedliche Transaktionspfade dieselbe CICS-Transaktions-ID gemeinsam nutzen (sodass diese unterschiedliche Mengen an verbrauchten Ressourcen darstellen kann).
- Ob ein Db2-Abrechnungsdatensatz sich nur auf eine Transaktion oder auf einen Teil einer Transaktion bezieht (weil Sie eine der in „Beziehung zwischen Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendaten­sätzen steuern“ auf Seite 136 beschriebenen Maßnahmen angewendet haben) oder ob er Informationen zu mehreren Transaktionen enthält.

In Abbildung 35 auf Seite 139 wird gezeigt, wie diese Faktoren kombiniert werden, um vier typische Szenarios zu erstellen, die beim Abgleich von Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendaten­sätzen auftreten können. In den folgenden Abschnitten werden Strategien vorgeschlagen, wie die Datensätze in den einzelnen Fällen abgeglichen und wie die Kosten für die verbrauchten Ressourcen dem Benutzer zugeordnet werden können.

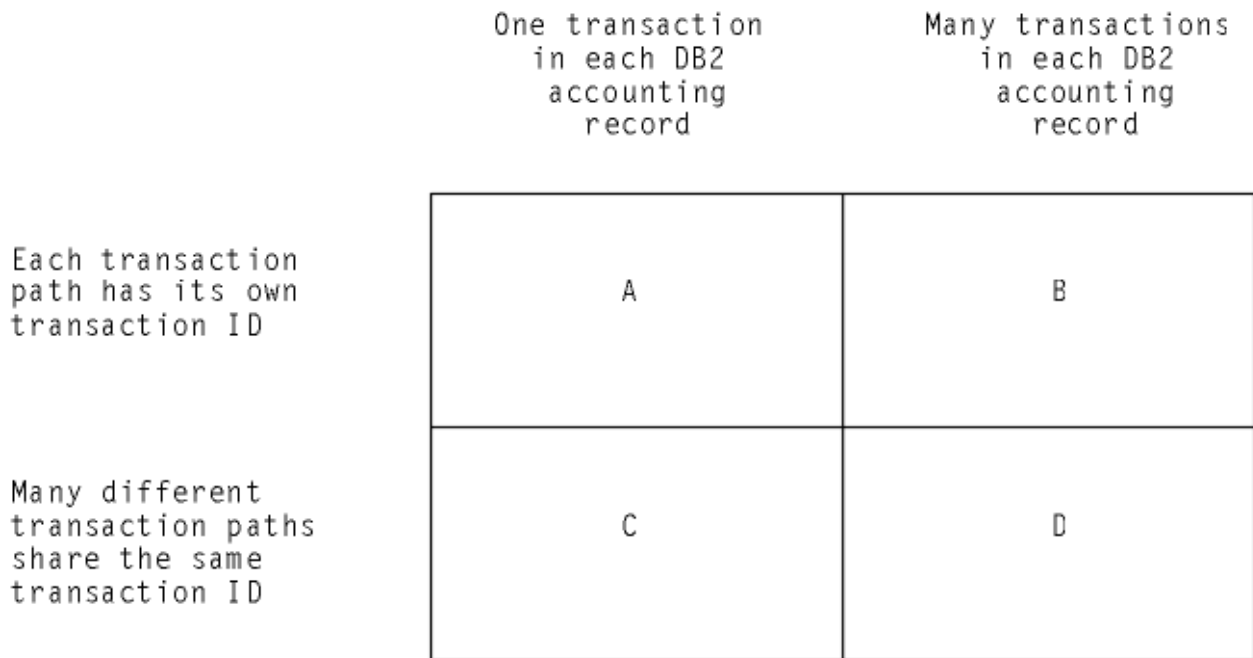


Abbildung 35. Unterschiedliche Abrechnungsszenarios

Szenario A: In jedem Db2-Abrechnungsdatensatz ist eine Transaktion vorhanden, die eine eigene Transaktions-ID aufweist

In diesem Szenario wissen Sie, dass jeder Db2-Abrechnungsdatensatz Informationen enthält, die sich auf eine einzige, identifizierbare CICS-Transaktion beziehen. Wenn die Transaktion mehrmals auf Db2-Ressourcen zugegriffen hat, sind von Db2 möglicherweise mehrere Abrechnungsdatensätze für diese Transaktion erstellt worden. Sie müssen die auf die Transaktion Db2 bezogenen Abrechnungsdatensätze mit dem CICS-Leistungsdatensatz für die Transaktion abgleichen.

Der Benutzer der Transaktion kann über den CICS-Leistungsdatensatz ermittelt werden. Dieser Datensatz enthält die auf diese Transaktion bezogenen CICS-Aktivitäten. Sie können die auf diese Transaktion anwendbaren Db2-Abrechnungsdatensätze mithilfe eines der in „Daten des Db2-Abrechnungsdatensatzes zum Ermitteln der entsprechenden CICS-Leistungsklassendatenätze verwenden“ auf Seite 137 aufgelisteten Datenelemente ermitteln.

Da alle diese Transaktionen identisch sind, ist zu erwarten, dass sie vergleichbare Ressourcenmengen verbrauchen. Sie könnten zu Abrechnungszwecken für jeden Transaktionstyp Modelltransaktionen erstellen. Da Sie ermitteln können, welche Db2-Abrechnungsdatensätze auf welche CICS-Transaktionen anwendbar sind, können Sie die Db2-Abrechnungsdatensätze und den CICS-Leistungsdatensatz für eine einzige Transaktion abgleichen und anschließend die in diesen Abrechnungsdatensätzen verwendeten Db2-Ressourcenmengen jeder nachfolgenden Transaktion desselben Typs zuordnen. Sie sollten die Richtigkeit Ihrer Modelle regelmäßig prüfen, falls sich die Ressourcennutzung ändert.

Szenario B: In jedem Db2-Abrechnungsdatensatz sind einige Transaktionen enthalten, aber jeder Transaktionstyp hat eine eigene Transaktions-ID

In diesem Szenario können in jedem Db2-Abrechnungsdatensatz Informationen zu mehreren Transaktionen enthalten sein, sodass Sie die einzelnen Db2-Abrechnungsdatensätze nicht direkt dem entsprechenden CICS-Leistungsdatensatz zuordnen können. Sie können jedoch die in den Db2-Abrechnungsdatensätzen vorhandenen Transaktionstypen ermitteln, da sich jede Transaktions-ID auf nur einen einzigen Transaktionstyp bezieht.

Wenn nur ein einziger CICS-Transaktionstyp in einem bestimmten Db2-Abrechnungsdatensatz vorhanden ist, können die in Db2 verbrauchten Ressourcen gleichmäßig zwischen den einzelnen Transaktionen aufgeteilt werden. Dies ist angemessen, da die Transaktionen (definitionsbedingt) nahezu identisch sind. Die Anzahl der Festschreibungen und Zurücksetzungen im Db2-Abrechnungsdatensatz gibt die Anzahl der Arbeitseinheiten an, die durch diesen Datensatz abgedeckt werden. Wie jedoch in „Beziehung zwischen Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendatenätzen steuern“ auf Seite 136 bemerkt,

können Arbeitseinheiten, die sich in derselben Transaktion befinden, unterschiedliche Threads verwenden und dadurch nicht im selben Db2-Abrechnungsdatensatz vorhanden sein. Stellen Sie sicher, dass Sie den einzelnen Transaktionen alle relevanten Db2-Ressourcen zugeordnet haben; dies kann beinhalten, dass mehrere Db2-Abrechnungsdatensätze geprüft werden müssen.

Wenn mindestens zwei unterschiedliche CICS-Transaktionstypen in einem bestimmten Db2-Abrechnungsdatensatz vorhanden sind (da sie dasselbe DB2ENTRY-Objekt und daher denselben Thread verwenden), können Sie nicht die Methode verwenden, bei der die Ressourcen gleichmäßig aufgeteilt werden, da die unterschiedlichen Transaktionstypen möglicherweise unterschiedliche Ressourcen verwenden. In diesem Fall können Sie Modelltransaktionen erstellen, indem Sie die Menge der von den einzelnen CICS-Transaktionstypen verwendeten Db2-Ressourcen in regelmäßigen Abständen messen. Nehmen Sie diese Messungen vor, indem Sie die Threadwiederverwendung zeitweise nicht zulassen und den sich ergebenden Db2-Abrechnungsdatensatz prüfen, der Informationen zu nur einer einzigen Transaktion enthält. Verwenden Sie diese Modelltransaktionen, um dem Benutzer die Kosten für die Ressourcen zuzuordnen. Sie sollten die Richtigkeit der Modelltransaktionen regelmäßig überprüfen.

Szenario C: In jedem Db2-Abrechnungsdatensatz ist eine Transaktion vorhanden, aber mehrere Transaktionstypen verwenden dieselbe Transaktions-ID

In diesem Szenario wissen Sie, dass jeder Db2-Abrechnungsdatensatz Informationen zu einer einzigen CICS-Transaktion enthält; da jedoch mehrere Transaktionstypen dieselbe Transaktions-ID verwenden, können Sie nicht sicher sein, welcher Transaktionstyp durch einen bestimmten Db2-Abrechnungsdatensatz dargestellt wird.

Sie können nicht eine Gruppe von Datensätzen für eine einzige Instanz einer Transaktion abgleichen und diese Zahlen anschließend wiederverwenden, wie Sie dies in Szenario A tun könnten. Sie müssen alle CICS-Leistungsdatensätze einzeln mit ihren entsprechenden Db2-Abrechnungsdatensätzen abgleichen. Nur wenn Sie dies tun, können Sie ermitteln, welcher Transaktionstyp von den einzelnen Db2-Datensätzen dargestellt wird.

Sie können die einzelnen Db2-Abrechnungsdatensätze mit dem jeweils relevanten CICS-Leistungsdatensatz abgleichen, indem Sie die in „Daten des Db2-Abrechnungsdatensatzes zum Ermitteln der entsprechenden CICS-Leistungsklassendatensätze verwenden“ auf Seite 137 aufgelisteten Datenelemente verwenden. Wenn Sie im DB2ENTRY-Objekt oder im DB2CONN-Objekt entweder ACCOUNTREC(TASK) oder ACCOUNTREC(UOW) angegeben haben, sodass CICS das zugehörige LU6.2-Token an Db2 übergibt, können Sie die Datensätze leicht miteinander abgleichen. Ist dies nicht möglich, müssen Sie die Datensätze auf der Basis ihrer Zeitmarken abgleichen. In diesem Fall ist der Abgleich möglicherweise nicht korrekt, wenn Transaktionen gleichzeitig ausgeführt werden.

Sie können Ihre abgeglichenen Gruppen von Datensätzen dann verwenden, um die Kosten für jede für die Transaktion verwendete Ressource dem durch den CICS-Leistungsdatensatz angegebenen Benutzer zuzuordnen.

Szenario D: In jedem Db2-Abrechnungsdatensatz sind mehrere Transaktionen vorhanden und mehrere Transaktionstypen verwenden dieselbe Transaktions-ID.

In diesem Szenario kann jeder der Db2-Abrechnungsdatensätze Informationen zu mehreren Transaktionen enthalten und Sie können anhand der Transaktions-IDs auch nicht feststellen, welche Transaktionstypen im Abrechnungsdatensatz vorhanden sind.

Diese Situation wird am besten vermieden, da Sie wahrscheinlich nicht in der Lage sind, Datensätze korrekt abzugleichen. Wenn diese Situation auftritt, besteht die beste Lösung darin, Modelltransaktionen wie in Szenario B beschrieben zu erstellen. Suchen Sie als nächstes eine Möglichkeit, wie Sie die CICS-Leistungsdatensätze mit einer Kennung markieren können, die für jede Transaktion eindeutig ist. Beispielsweise könnte der Benutzer in einem Benutzerfeld in den Leistungsdatensätzen Informationen bereitstellen, mit denen die zurzeit ausgeführte Transaktion angegeben wird. Sie können dieses Feld dann verwenden, um zu ermitteln, welche der Modelltransaktionen in diesem Fall für die Abrechnung verwendet werden sollte.

Abrechnung zur Prozessorbelegung in einer CICS-Db2-Umgebung

In den Datensätzen des Abrechnungs- und des Statistiktrace von Db2 werden Informationen zum Verbrauch der Prozessorzeit bereitgestellt. Sie können diese Datensätze zum Berechnen der in CICS und Db2 insgesamt verbrauchten Prozessorzeit verwenden.

Der *Abrechnungstrace* von Db2 kann mit den Klassen CLASS 1, CLASS 2 oder CLASS 3 gestartet werden. CLASS 1 muss jedoch stets aktiv sein, um die durch Aktivieren von CLASS 2 und/oder CLASS 3 erfassten Informationen extern bereitzustellen.

Bei den in den Db2-Abrechnungsdatensätzen berichteten Prozessorzeiten handelt es sich um die Tasksteuerblockzeit für den Thread-Tasksteuerblock, der im CICS- oder Db2-Adressraum mithilfe von speicherübergreifenden Services Code ausführt sowie um die SRB-Zeit für geplante Arbeit in CICS.

Die Einstellung CLASS 1 (die Standardeinstellung) führt dazu, dass von mehreren Db2-Komponenten während der Ausführung Abrechnungsdaten kumuliert werden. Diese Daten werden dann erfasst, um den Db2-Abrechnungsdatensatz zu schreiben. Die Datenerfassung schließt nicht den Systemaufwand für das Tracing einzelner Ereignisse ein.

CLASS 2 und CLASS 3 aktivieren viele zusätzliche Tracepunkte. Jedes Auftreten dieser Ereignisse wird intern aufgezeichnet, aber es wird *nicht* an ein externes Ziel geschrieben. Die Abrechnungsfunktion verwendet diese Traces stattdessen, um die zusätzlichen Gesamtstatistikdaten zu berechnen, die im Abrechnungsdatensatz auftreten, wenn CLASS 2 oder CLASS 3 aktiviert ist. Abrechnungsklasse CLASS 1 muss aktiv sein, um die Informationen extern bereitzustellen.

CLASS 2 erfasst das abgelaufene Delta und die 'IN DB2' verbrauchten Prozessorzeiten und erfasst dies im Abrechnungsdatensatz.

CLASS 3 erfasst die für die Ein-/Ausgabe abgelaufene Zeit und die für Sperren und Verriegelungen 'IN DB2' verbrauchte Zeit für Aussetzungen und erfasst diese im Abrechnungsdatensatz.

CLASS 7 und CLASS 8 in Db2 erfassen die Db2-Abrechnung auf Paketebene und die Db2-Abrechnung auf Paketebene im Wartemodus. Informationen zur Abrechnung auf Paketebene finden Sie in [Db2-Leistung verwalten in der Produktdokumentation zu Db2 for z/OS](#).

Der *Statistiktrace* berichtet die Prozessorzeit in den Statistikdatensätzen. Die berichteten Prozessorzeiten lauten wie folgt:

- Zeit, die in einem Tasksteuerblock für einen Db2-Adressraum verbraucht wurde, der asynchron zum CICS-Adressraum ausgeführt wird. Beispiele hierfür sind das Db2-Protokoll und Schreibvorgänge aus den Pufferpools.
- Zeit, die unter SRBs verbraucht wurde, die unter Db2-Adressräumen terminiert waren. Ein Beispiel hierfür ist die Engine für asynchrone Lesevorgänge, mit der sequenzielle Vorablesezugriffe ausgeführt werden.

Die in den Statistikdatensätzen berichteten Db2-Adressräume lauten wie folgt:

- Adressraum des Datenbankmanagers
- Adressraum der Systemservices
- IRLM (Interner Ressourcensperrenmanager).

In einer CICS-Db2-Umgebung ist die in den Db2-Abrechnungsdatensätzen berichtete Prozessorzeit in der Regel viel größer als die in den Db2-Statistikdatensätzen berichtete Prozessorzeit, weil der größte Teil der Prozessorzeit im Thread-Tasksteuerblock selbst und in den Db2-Adressräumen verbraucht wurde, die speicherübergreifende Services verwenden.

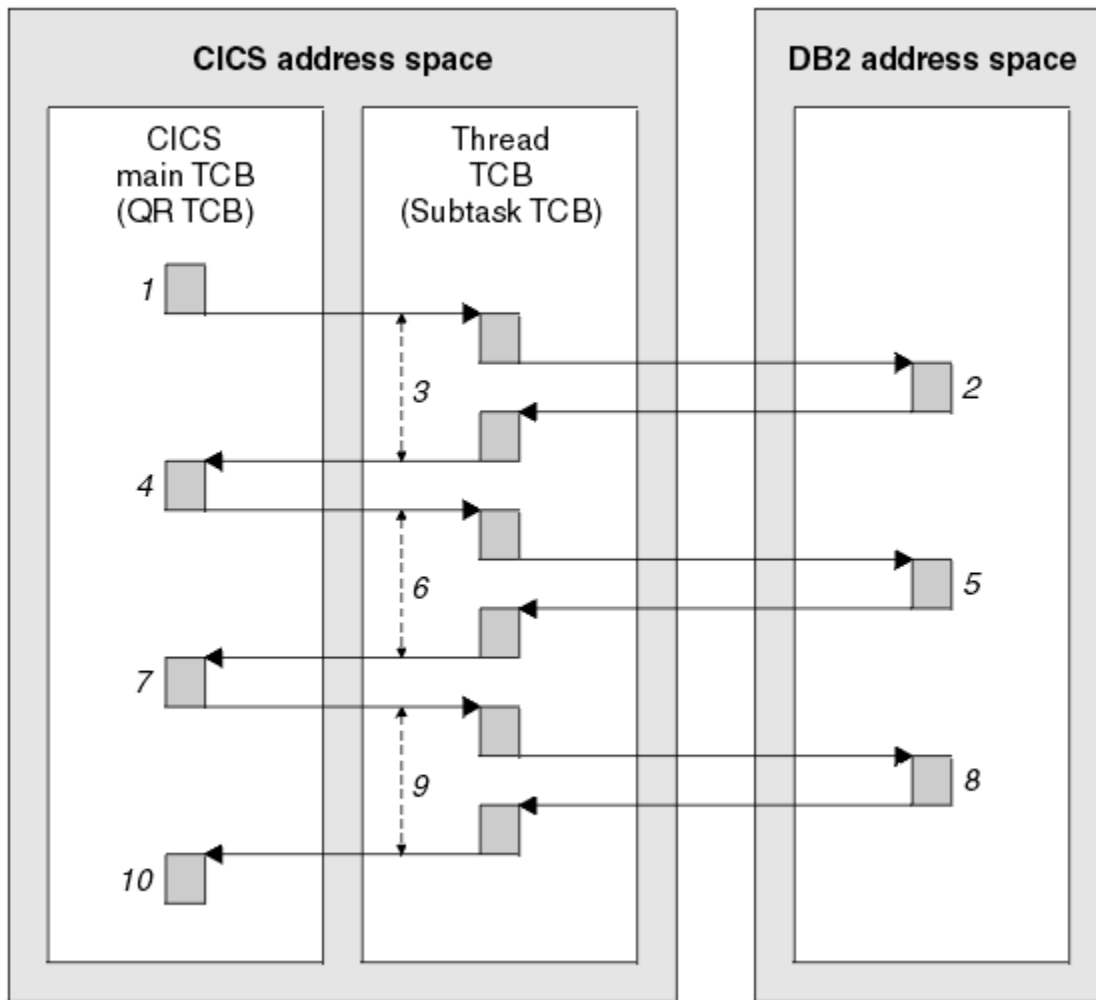
Db2-Abrechnungsdatensätze werden erzeugt, wenn ein Thread beendet wird oder eine Anmeldung auftritt. Dies bedeutet, dass der im Db2-Abrechnungsdatensatz berichtete Zeitraum der Zeitraum zwischen dem Threadstart oder der Benutzeranmeldung (falls der Thread zuvor von einem anderen Benutzer verwendet wurde) und der Threadbeendigung oder einer weiteren Anmeldung ist. Wenn für dasselbe DB2ENTRY-Objekt mehrere unterschiedliche Transaktionen angegeben sind und diese Transaktionen für die Threadanmeldung dieselbe CICS-Transaktions-ID und dieselbe Db2-Berechtigungs-ID verwenden, so kann der Db2-Abrechnungsdatensatz für diesen Thread Daten für mehrere Transaktionen enthalten.

Wenn eine Transaktion ihren zugehörigen Db2-Thread am Synchronisationspunkt freigibt und der Thread beendet oder von einer anderen Transaktion wiederverwendet wird, so muss die ursprüngliche Transaktion einen anderen Thread verwenden. Eine einzelne Transaktion kann daher während ihres Ausführungszyklus mehreren Db2-Threads zugeordnet sein. Da Db2 für jeden einzelnen Thread Abrechnungsdatensätze erzeugt, kann dies bedeuten, dass mehrere Abrechnungsdatensätze für die Transaktion erzeugt werden. In „Beziehung zwischen Db2-Abrechnungsdatensätzen und CICS-Leistungsklassendatensätzen steuern“ auf Seite 136 finden Sie Informationen dazu, wie Sie die Transaktionen ermitteln können, die in den einzelnen Db2-Abrechnungsdatensätzen enthalten sind.

Die Prozessorzeit kann innerhalb der Transaktion präzise nach Berechtigungs-ID angegeben werden, indem alle Abrechnungsdatensätze zu dieser Berechtigungs-ID und Transaktion zusammengefasst werden. Beachten Sie jedoch, dass die dem Abrechnungsdatensatz zugeordnete verstrichene Zeit in diesem Fall wenig Wert aufwiese, da die Summe der verstrichenen Zeiträume nicht die Zeit einschließt, die zwischen einem Festschreibungspunkt und dem ersten SQL-Aufruf in der nächsten Arbeitseinheit liegt. Die für einen Thread verstrichene Zeit ist dem Thread zugeordnet und nicht der Transaktion.

In Abbildung 36 auf Seite 143 und Abbildung 37 auf Seite 145 werden alle Prozessorzeiträume gezeigt, die von CICS und Db2 berichtet werden, sowie ihr Entstehungsort. Die Position der Verarbeitung unterscheidet sich danach, ob die Anwendung, die auf Db2 zugreift, threadsicher ist. Wenn nämlich CICS mit Db2 verbunden ist und CICS die Umgebung für offene Transaktionen nutzt, verwendet die CICS-Db2-Anschlussfunktion von CICS verwaltete offene Tasksteuerblöcke und keine CICS-Db2-Tasksteuerblöcke für Subtasks.

In Abbildung 36 auf Seite 143 wird die Situation gezeigt, dass CICS mit Db2 verbunden ist und die Anwendung, die auf Db2 zugreift, nicht threadsicher ist.



Diese Abbildung zeigt die Umgebung, während CICS mit Db2 verbunden ist und wenn die Anwendung, die auf Db2 zugreift, nicht threadsicher ist.

Zeiträume 3 + 6 + 9 werden als CLASS 1-Prozessorzeit berichtet (Zeit, die unter dem Thread-Tasksteuerblock verbraucht wurde)
 Zeiträume 2 + 5 + 8 werden als CLASS 2-Prozessorzeit berichtet (Zeit, die in Db2 verbraucht wurde)
 Zeiträume 1 + 4 + 7 + 10 werden als CICS-Prozessorzeit berichtet (Zeit, die im Haupt-Tasksteuerblock von CICS verbraucht wurde)

Abbildung 36. CICS mit Db2 und einer nicht threadsicheren Anwendung: erfasste Prozessorzeiten

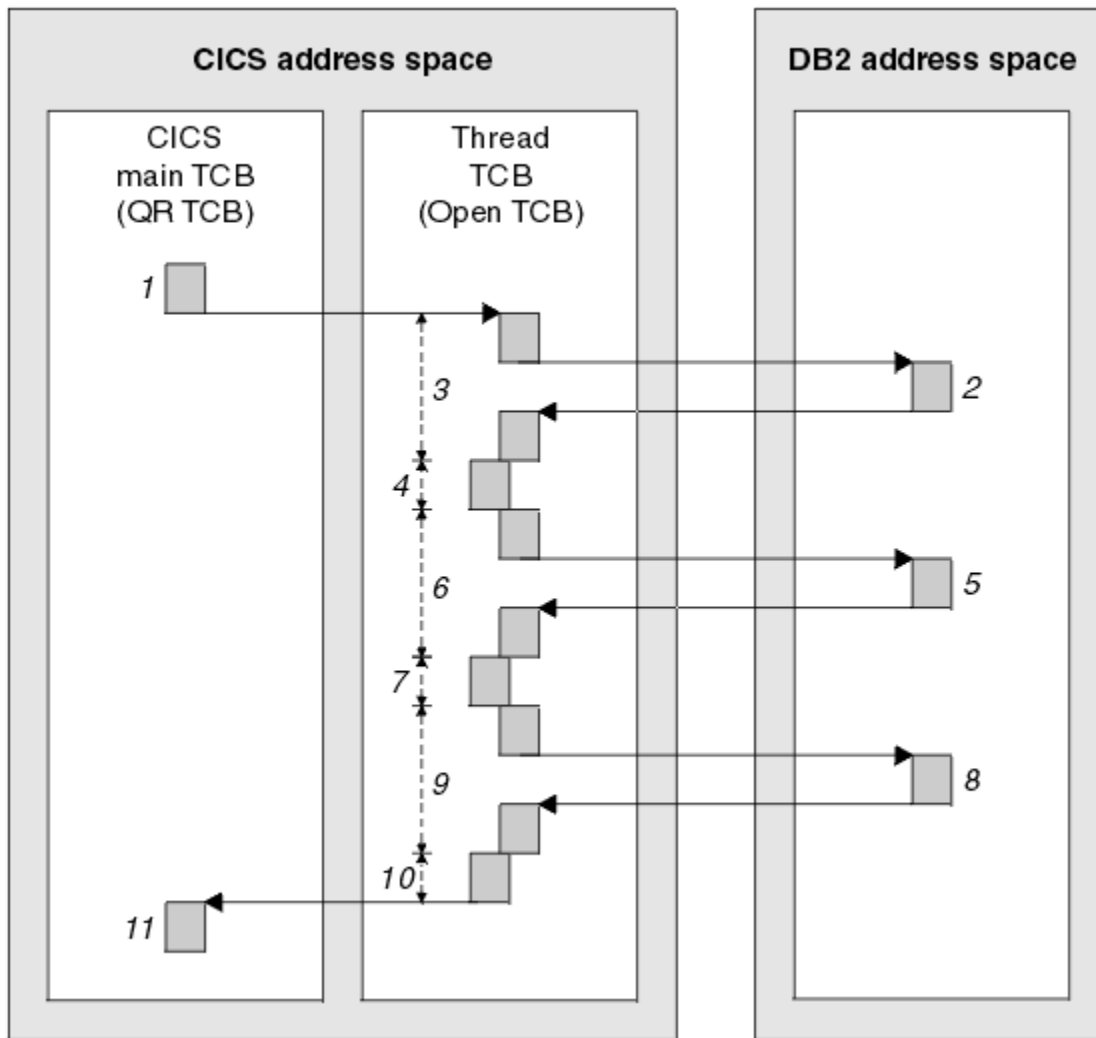
Wenn CICS mit Db2 verbunden ist, aber die auf Db2 zugreifende Umgebung nicht threadsicher ist, verwendet CICS einen offenen Tasksteuerblock, um den Thread in Db2 auszuführen; CICS verwendet den offenen Tasksteuerblock jedoch in derselben Weise, wie auch ein Tasksteuerblock für Subtasks verwendet würde. Der Tasksteuerblock, den CICS zum Ausführen des Threads in Db2 verwendet, wird als Thread-Tasksteuerblock bezeichnet.

Die in der Abbildung dargestellten Zeiträume für die Prozessorzeit lauten wie folgt:

- Zeitraum 1: Die Anwendung startet im Haupt-Tasksteuerblock von CICS. Am Ende von Zeitraum 1 gibt die Anwendung eine EXEC-SQL-Anforderung aus.
- Zeitraum 2: Db2 bearbeitet die Anforderung der Anwendung. Diese Prozessorzeit wird im Db2-Adressraum verbraucht. Am Ende von Zeitraum 2 übergibt die Db2-Instanz ihre Antwort an die CICS-Db2-Anschlussfunktion.

- Zeitraum 3: Die CICS-Db2-Anschlussfunktion führt Verarbeitungsschritte im Thread-Tasksteuerblock aus. Dadurch wird sowohl die Verarbeitung abgedeckt, die erforderlich ist, damit die Anwendung auf Db2 zugreifen kann, als auch die Verarbeitung, die erforderlich ist, damit die Db2-Antwort zurück an die Anwendung übergeben werden kann. Enthalten ist auch die Prozessorzeit, die erforderlich ist, damit Db2 die Anforderung bearbeiten und antworten kann. Zeitraum 2 ist daher in Zeitraum 3 verschachtelt. Am Ende des Zeitraums 3 wird die Db2-Antwort an die Anwendung übergeben und der Zugriff auf Db2 ist abgeschlossen.
- Zeitraum 4: Die Anwendung hat aktuell keine weitere Verarbeitung in Db2 auszuführen. Der Anwendungscode wird im Haupt-Tasksteuerblock von CICS ausgeführt, bis die Anwendung erneut auf Db2 zugreifen muss. Am Ende von Zeitraum 4 gibt die Anwendung eine zweite EXEC-SQL-Anforderung aus.
- Zeitraum 5: Wie in Zeitraum 2 bearbeitet Db2 die zweite Anforderung der Anwendung.
- Zeitraum 6: Die CICS-Db2-Anschlussfunktion führt im Thread-Tasksteuerblock Verarbeitungsschritte aus, die in Beziehung zu dieser zweiten Anforderung stehen, oder sie wartet auf eine Db2-Antwort.
- Zeitraum 7: Der Anwendungscode wird im Haupt-Tasksteuerblock von CICS erneut ausgeführt. Es wird eine dritte EXEC-SQL-Anforderung ausgegeben.
- Zeitraum 8: Wie in Zeitraum 2 bearbeitet Db2 die dritte Anforderung der Anwendung.
- Zeitraum 9: Die CICS-Db2-Anschlussfunktion führt im Thread-Tasksteuerblock Verarbeitungsschritte aus, die in Beziehung zu dieser dritten Anforderung stehen, oder sie wartet auf eine Db2-Antwort.
- Zeitraum 10: Die Anwendung hat nun alle ihre EXEC-SQL-Anforderungen ausgeführt. Der Anwendungscode wird bis zur Beendigung weiterhin im Haupt-Tasksteuerblock von CICS ausgeführt.

In [Abbildung 37 auf Seite 145](#) wird die Situation gezeigt, dass CICS mit Db2 verbunden ist und die Anwendung, die auf Db2 zugreift, threadsicher ist.



Diese Abbildung zeigt die Umgebung, während CICS mit Db2 verbunden ist und wenn die Anwendung, die auf Db2 zugreift, threadsicher ist.

Zeiträume 3 + 4 + 6 + 7 + 9 + 10 werden als CLASS 1-Prozessorzeit berichtet (Zeit, die unter dem offenen Tasksteuerblock für die Verarbeitung des Zugriffs auf Db2

und zum Ausführen von Anwendungscode verbraucht wurde)

Zeiträume 2 + 5 + 8 werden als CLASS 2-Prozessorzeit berichtet (Zeit, die in Db2 verbraucht wurde)

Zeiträume 1 + 11 werden als CICS-Prozessorzeit berichtet (Zeit, die im Haupt-Tasksteuerblock von CICS verbraucht wurde)

Abbildung 37. CICS mit Db2 und einer threadsicheren Anwendung: erfasste Prozessorzeiten

CICS verwendet hier einen offenen Tasksteuerblock, um den Thread in Db2 auszuführen. Da die Anwendung threadsicher ist, kann der Anwendungscode auch im offenen Tasksteuerblock ausgeführt werden. Die in der Abbildung dargestellten Zeiträume für die Prozessorzeit lauten wie folgt:

- Zeitraum 1: Die Anwendung startet im Haupt-Tasksteuerblock von CICS. Am Ende von Zeitraum 1 gibt die Anwendung eine EXEC-SQL-Anforderung aus.
- Zeitraum 2: Db2 bearbeitet die Anforderung der Anwendung. Diese Prozessorzeit wird im Db2-Adressraum verbraucht. Am Ende von Zeitraum 2 übergibt die Db2-Instanz ihre Antwort an die CICS-Db2-Anschlussfunktion.

- Zeitraum 3: Die CICS-Db2-Anschlussfunktion führt Verarbeitungsschritte im Thread-Tasksteuerblock aus. Dadurch wird sowohl die Verarbeitung abgedeckt, die erforderlich ist, damit die Anwendung auf Db2 zugreifen kann, als auch die Verarbeitung, die erforderlich ist, damit die Antwort von Db2 zurück an die Anwendung übergeben werden kann. Enthalten ist auch die Prozessorzeit, die erforderlich ist, damit Db2 die Anforderung bearbeiten und antworten kann. Zeitraum 2 ist daher in Zeitraum 3 verschachtelt. Am Ende des Zeitraums 3 wird die Db2-Antwort an die Anwendung übergeben und der Zugriff auf Db2 ist abgeschlossen.
- Zeitraum 4: Die Anwendung hat aktuell keine weitere Verarbeitung in Db2 auszuführen. Da die Anwendung threadsicher ist, wird der Anwendungscode nun im Thread-Tasksteuerblock (dem offenen Tasksteuerblock) ausgeführt. Am Ende von Zeitraum 4 gibt die Anwendung eine zweite EXEC-SQL-Anforderung aus.
- Zeitraum 5: Wie in Zeitraum 2 bearbeitet Db2 die zweite Anforderung der Anwendung.
- Zeitraum 6: Die CICS-Db2-Anschlussfunktion führt im Thread-Tasksteuerblock Verarbeitungsschritte aus, die in Beziehung zu dieser zweiten Anforderung stehen, oder sie wartet auf eine Db2-Antwort.
- Zeitraum 7: Der Anwendungscode wird erneut im Thread-Tasksteuerblock ausgeführt. Es wird eine dritte EXEC-SQL-Anforderung ausgegeben.
- Zeitraum 8: Wie in Zeitraum 2 bearbeitet Db2 die dritte Anforderung der Anwendung.
- Zeitraum 9: Die CICS-Db2-Anschlussfunktion führt im Thread-Tasksteuerblock Verarbeitungsschritte aus, die in Beziehung zu dieser dritten Anforderung stehen, oder sie wartet auf eine Db2-Antwort.
- Zeitraum 10: Die Anwendung hat nun alle ihre EXEC-SQL-Anforderungen ausgeführt. Der Anwendungscode wird erneut im Thread-Tasksteuerblock ausgeführt.
- Zeitraum 11: Der Anwendungscode hat seine Verarbeitung abgeschlossen. Die Anwendung wird beendet und die Verarbeitung kehrt zum Haupt-Tasksteuerblock von CICS zurück.

Beachten Sie, dass eine als threadsicher definierte Umgebung (d. h., Sie haben CICS mitgeteilt, dass der Anwendungscode threadsicher ist), dennoch CICS-Befehle absetzen kann, die nicht threadsicher sind. Diese Befehle erzwingen, dass ein Wechsel zurück zum Haupt-Tasksteuerblock von CICS erfolgt; der Anwendungscode wird anschließend weiterhin im Haupt-Tasksteuerblock von CICS ausgeführt, bis die Anwendung ihre nächste EXEC-SQL-Anforderung ausführt. Die Aktivität, die dem nicht threadsicheren CICS-Befehl und dem im Haupt-Tasksteuerblock von CICS ausgeführten Anwendungscode zugeordnet ist, wird als CICS-Prozessorzeit und nicht als CLASS 1-Prozessorzeit berichtet. Sobald die EXEC-SQL-Anforderung abgesetzt ist, kann der Anwendungscode erneut im offenen Tasksteuerblock ausgeführt werden; ab diesem Zeitpunkt wird er als CLASS 1-Prozessorzeit berichtet.

CLASS1-Prozessorzeit berechnen

Für CLASS 1 wird beim Anhängen des Tasksteuerblocks ein Prozessorzeitgeber für Tasks erstellt. Wenn ein Thread zu Db2 gestartet wird, wird der Zeitgeberwert gespeichert. Bei Beendigung des Threads oder bei Änderung der Berechtigungs-ID wird der Zeitgeber erneut geprüft und sowohl der Start- als auch der Endwert des Zeitgebers werden im SMF-Datensatz mit dem Typ 101 gespeichert.

In [Abbildung 36 auf Seite 143](#) ist CLASS 1 die Summe der Zeitwerte für 3, 6 und 9. In [Abbildung 37 auf Seite 145](#) ist CLASS 1 die Summe der Zeitwerte für 3, 4, 6, 7, 9 und 10.

Die Felder im SMF-Datensatz mit dem Typ 101, dem Db2-Abrechnungsdatensatz, der für die Berechnung der CLASS 1-Prozessorzeit verwendet wird, lauten wie folgt:

- QWACBJST für Beginn der Thread-Tasksteuerblock-Zeit
- QWACEJST für Ende der Thread-Tasksteuerblock-Zeit
- QWACBSRB für Beginn der ASCB-SRB-Zeit
- QWACESRB für Ende der ASCB-SRB-Zeit.

Sie finden eine Beschreibung der Inhalte des Db2-Abrechnungsdatensatzes in [Db2-Leistung verwalten in der Produktdokumentation zu Db2 for z/OS](#). Es gibt auch noch die Beschreibung der Abrechnungsdatensatzfelder im Member DSNWMSGs, das im Lieferumfang von SDSNSAMP enthalten ist.

Wenn die CLASS 1-Erfassung für einen Thread aktiviert wird, wird die im offenen L8-Tasksteuerblock verbrauchte Zeit erfasst. Da der L8-Tasksteuerblock sowohl für die CICS-Aktivität als auch für die Db2-Aktivität verwendet wird, ist hier die in der CICS-Db2-Anschlussfunktion verbrauchte Prozessorzeit einschließlich von Traceaufrufen eingeschlossen. Auch die für die Ausführung von Anwendungscode (wenn die Anwendung threadsicher ist) und threadsicheren CICS-Befehlen im offenen L8-Tasksteuerblock verbrauchte Prozessorzeit ist eingeschlossen. Wenn ein Thread wiederverwendet wird, ist die für den Thread erforderliche Prozessorzeit für die Systemverwaltung auch in der CLASS 1-Prozessorzeit eingeschlossen. Wie in früheren Releases gibt es bei der Verarbeitung der Threaderstellung und der Threadbeendigung einen Anteil, der von der CLASS 1-Zeit nicht erfasst wird. Die CLASS 1-Prozessorzeit schließt nicht die Zeit ein, die für die Ausführung von Anwendungscode im QR-Tasksteuerblock verbraucht wird; dies läuft folgendermaßen ab:

- Im Anfangszeitraum, bevor die Anwendung ihre erste EXEC-SQL-Anforderung absetzt und in den offenen Tasksteuerblock (Zeitraum 1 in der Abbildung) verschoben wird. Dieser erste Anwendungscode wird im Haupt-Tasksteuerblock von CICS ausgeführt und wird von der Db2-Abrechnung nicht erkannt.
- Wenn die Anwendung einen nicht threadsicheren CICS-Befehl ausgibt. Dadurch wird erzwungen, dass die Verarbeitung an den Haupt-Tasksteuerblock von CICS zurückgegeben wird. Der Anwendungscode wird dann weiterhin im Haupt-Tasksteuerblock von CICS ausgeführt, in dem er von der Db2-Abrechnung nicht erkannt wird, bis die Anwendung ihre nächste EXEC-SQL-Anforderung absetzt. An diesem Punkt kann die Verarbeitung zurück in den offenen Tasksteuerblock verschoben werden.

Wenn Sie die Db2-Sysplex-Abfrageparallelität (parallele Abfrage) nicht verwenden, müssen Sie die Db2-CLASS 1-Prozessorzeit auch nicht für Abrechnungszwecke verwenden. Die im CICS-SMF-Datensatz mit dem Typ 110 erfasste Prozessorzeit ist alles, was Sie brauchen, um eine vollständige Darstellung der von einer Anwendung verbrauchten Prozessorzeit anzugeben. Dieser Datensatz enthält die im Anwendungscode verbrauchte Zeit, die Kosten für die Threaderstellung und -beendigung sowie die Zeit, die von der Db2-CLASS 1-Prozessorzeit abgedeckt ist. Weitere Informationen hierzu finden Sie in [„CICS- und Db2-Prozessorzeit für Db2 berechnen“](#) auf Seite 148.

Prozessorzeit der Abrechnungsklasse CLASS 2

Für Abrechnungsklasse CLASS 2 wird der Zeitgeber bei jedem Einstieg in und bei jedem Verlassen von Db2 geprüft, um die in Db2 verbrauchte Zeit ('IN DB2') im SMF-Datensatz mit dem Typ 101 zu erfassen. In diesem Fall handelt es sich um die Differenz, die im Datensatz gespeichert ist. Das Feld im SMF-Datensatz mit dem Typ 101, das für die Prozessorzeit der Abrechnungsklasse CLASS 2 verwendet wird, lautet QWACAJST.

In [Abbildung 36 auf Seite 143](#) und [Abbildung 37 auf Seite 145](#) ist CLASS 2 die Summe der Zeitwerte für 2, 5 und 8.

Eine Beschreibung des Inhalts der Db2-Statistikdaten finden Sie in [Db2-Leistung verwalten in der Produktdokumentation zu Db2 for z/OS](#).

Die abgelaufene Zeit (Start- und Endzeiten zwischen den definierten Punkten) wird auch im SMF-Datensatz des Typs 101 erfasst (QWACASC).

Wenn CICS mit Db2 verbunden ist, kann eine beträchtliche Differenz zwischen der CLASS 1- und der CLASS 2-Prozessorzeit auftreten, wie es auch bei IMS und TSO der Fall ist. Dies liegt daran, dass bei Bestehen einer Verbindung zwischen CICS und Db2 die CLASS 1-Prozessorzeit die Prozessorzeit einschließt, die in der CICS-Db2-Anschlussfunktion einschließlich Traceaufrufen verbraucht wurde und auch Prozessorzeit einschließt, die für die Ausführung von threadsicherem Anwendungscode und threadsicheren CICS-Befehlen im offenen Tasksteuerblock verbraucht wurde. Diese gesamte Prozessorzeit fällt im CICS-Adressraum an und wird daher nicht für Abrechnungsklasse CLASS 2 berichtet. Die CLASS 2-Prozessorzeit selbst ist durch die Umgebung für offene Transaktionen nicht betroffen.

Anders als bei Abrechnungsklasse CLASS 1 spiegeln die für CLASS 2 abgelaufenen Zeiten und Prozessorzeiten die abgelaufenen Zeiten und in Db2 verbrauchten Prozessorzeiten ('IN DB2') richtig wider. Informationen zur Abrechnungsklasse CLASS 2 sind daher für die Überwachung der Leistung der SQL-Ausführung sehr nützlich. Der Prozessoraufwand, der bei der Klassenabrechnung anfällt, ist jedoch äußerst beträchtlich.

Um den Aufwand zu verringern, wurde in der Regel empfohlen, dass der CLASS 2-Trace auf das Verfolgen von Daten für Pläne und Positionen bestimmter Berechtigungs-IDs begrenzt wird. Der Prozessoraufwand beträgt 0-5 %, wobei 2 % typisch sind.

CICS- und Db2-Prozessorzeiten für DB2 Version 5 oder früher berechnen

Wenn Ihr CICS-System mit DB2 Version 5 oder früher verbunden ist, enthält der CICS-Leistungsklassendatensatz nicht die in Db2 verbrauchte Prozessorzeit. Zum Schätzen der gesamten für eine einzelne Transaktion verbrauchten Prozessorzeit fügen Sie Informationen aus dem entsprechenden CICS-Leistungsdatsatz und dem Db2-Abrechnungsdatsatz (Datensatz mit SMF-Typ 101) hinzu.

Informationen zu diesem Vorgang

Weitere Informationen zum Abgleich von CICS-Leistungsdatsätzen mit Db2-Abrechnungsdatsätzen finden Sie in „Db2-Abrechnungsdatsätze und CICS-Leistungsklassendatsätze in Beziehung zueinander setzen“ auf Seite 135.

Sehen Sie sich das CPU-Feld im CICS-Leistungsklassendatsatz an. Mithilfe des Db2-Abrechnungsdatsatzes können Sie die Prozessorzeit (T1) für Db2-Threads wie folgt berechnen:

$$T1 = QWACEJST - QWACBJST$$

Hiermit wird die CLASS 1-Prozessorzeit des Tasksteuerblocks berechnet. Die Summe der Prozessorzeit aus dem CICS-CPU-Feld und dem berechneten Wert für T1 ist ein Ausdruck der Prozessorzeit, die in CICS und Db2 verbraucht wurde.

Hinweise:

- Die CLASS 2-Prozessorzeit ist Teil der CLASS 1-Prozessorzeit und darf der Summe nicht hinzugefügt werden.
- Wenn die CLASS 2-Prozessorzeit von der CLASS 1-Prozessorzeit subtrahiert wird, ergibt dies die ungefähre CPU-Auslastung durch die CICS-Db2-Anschlussfunktion. Dies ist eine gute Unterstützung für die Optimierung.
- Die in den Db2-Adressräumen verwendete und in den Db2-Statistikdatsätzen erfasste Prozessorzeit hat keinen Bezug zu einem bestimmten Thread. Sie kann proportional zur CPU-Zeit verteilt sein, die in den Db2-Abrechnungsdatsätzen erfasst ist.
- Es ist nicht einfach, die im CICS-Adressraum unter den Subtask-Tasksteuerblöcken verbrauchte Prozessorzeit auf die CICS-Leistungsdatsätze zu verteilen, da sie die Prozessorzeiten für die Db2-Subtasks einschließt, die bereits im berechneten Wert T1 enthalten sind. Dies bedeutet, dass Prozessorzeit, die in anderen Subtasks als den Thread-Subtasks verbraucht wurde, nicht in der Addition eingeschlossen ist.
- Der größte Teil der in den Thread-Tasksteuerblöcken zum Erstellen des Threads verwendeten Prozessorzeit ist in keinen der Db2-Abrechnungsdatsätze enthalten, die diesem Thread zugeordnet sind, da diese Prozessorzeit verbraucht wird, bevor der Thread erstellt ist.
- Das Aufzeichnungsverhältnis für CICS und Db2 muss berücksichtigt werden. Das Aufzeichnungsverhältnis ist das Verhältnis der berichteten CPU-Zeit zur insgesamt verwendeten CPU-Zeit (weitere Informationen finden Sie unter [z/OS Resource Measurement Facility \(RMF\) User's Guide](#)).

CICS- und Db2-Prozessorzeit für Db2 berechnen

Wenn CICS mit Db2 verbunden ist und die Umgebung für offene Transaktionen genutzt wird, verwendet die CICS-Db2-Anschlussfunktion von CICS-verwaltete offene Tasksteuerblöcke und keine CICS-Db2-Subtask-Tasksteuerblöcke.

Informationen zu diesem Vorgang

Wenn CICS-verwaltete offene Tasksteuerblöcke verwendet werden, bedeutet dies, dass die CICS-Überwachungsfunktion die Aktivität messen kann, die zuvor nur im Db2-Abrechnungsdatsatz (dem SMF-Datsatz mit dem Typ 101) berichtet wurde. CICS kann nun beispielsweise die Prozessorzeit messen, die für den Db2-Thread verbraucht wurde, und die Prozessorzeit, die in Db2 verbraucht wurde (die CLASS 1-

und CLASS 2-CPU-Zeit). Wenn CICS offene L8-Tasksteuerblöcke verwendet, beinhaltet die für diese Tasksteuerblöcke von der CICS-Überwachungsfunktion berichtete Prozessorzeit die Db2-CLASS 1-Prozessorzeit.

In der Umgebung für offene Transaktionen kann die CICS-L8-Taskprozessorzeit auch die Kosten für die Erstellung eines Db2-Threads beinhalten.

Wenn Sie die Db2-Abfrageparallelität nicht verwenden, dürfen Sie die Prozessorzeit aus den CICS-Datensätzen (SMF-Datensätze mit dem Typ 110) und den Db2-Abrechnungsdatensätzen (SMF-Datensätze mit dem Typ 101) bei der Berechnung der gesamten Prozessorzeit für eine einzige Transaktion **nicht** addieren, da die Db2-Prozessorzeit dann zweimal eingeschlossen würde. Die gesamte Prozessorzeit für eine einzige Transaktion wird im Feld USRCPUT der CICS-Datensätze (Leistungsklassendatenfeld 008 in der Gruppe DFHTASK) erfasst. Dieses Feld enthält die gesamte Prozessorzeit, die von der Transaktion verwendet wurde, als die Transaktion in einem beliebigen Tasksteuerblock ausgeführt wurde, der vom CICS-Dispatcher verwaltet wird. Zu den von CICS-verwalteten Tasksteuerblöcken gehören die Tasksteuerblöcke des Typs QR, RO, CO und die Tasksteuerblöcke mit dem Modus L8.

Wenn Sie die parallele Db2-Abfrage verwenden, müssen Sie einige Prozessorzeiten aus den Db2-Abrechnungsdatensätzen hinzufügen. Mit der Db2-Sysplex-Abfrageparallelität verwendet Db2 zum Verarbeiten einer Abfrage möglicherweise mehrere Tasksteuerblöcke innerhalb des Db2-Adressraums. Die zu diesem Zweck verwendeten Tasksteuerblöcke sind keine CICS-Tasksteuerblöcke, das heißt, von ihnen verbrauchte Prozessorzeit wird in den CICS-SMF-Datensätzen mit dem Typ 110 nicht berücksichtigt. Um die Gesamtsumme der für eine parallele Abfrage verwendeten Prozessorzeit zu erhalten, müssen Sie daher die PAR.TASKS-Prozessorzeit, die im zur Transaktion zugehörigen Db2-SMF-Datensatz mit dem Typ 101 erfasst wurde, zur Prozessorzeit addieren, die im Feld USRCPUT im CICS-SMF-Datensatz mit dem Typ 110 erfasst wurde.

Kapitel 9. Fehlerbehebung für Db2

In diesem Abschnitt wird die Fehlerbestimmung für CICS Db2 beschrieben.

Dieser Abschnitt enthält Informationen zu Diagnose, Änderung und Optimierung.

Thread-Tasksteuerblöcke

In der CICS-Db2-Umgebung wird jeder Thread in Db2 unter einem Thread-Tasksteuerblock ausgeführt. Dieser Abschnitt bietet weitere technische Informationen zu Thread-Tasksteuerblöcken und soll eine Hilfe bei der Fehlerbestimmung sein.

In [Übersicht: Funktionsweise von Threads](#) finden Sie eine Übersicht über Thread-Tasksteuerblöcke.

Thread-Tasksteuerblöcke sind offene Tasksteuerblöcke mit dem Modus L8. Die offenen Tasksteuerblöcke sind untergeordnete Elemente ('Töchter') des Haupt-Tasksteuerblocks von CICS (des QR-Tasksteuerblocks). Der taskbezogene CICS-Db2-Benutzerexit selbst wird im offenen Tasksteuerblock ausgeführt und verwendet diesen auch, um den Thread auszuführen. Der taskbezogene Benutzerexit verwendet das CICS-Db2-Anhangsmodul DFHD2D2, um Db2 aufzurufen, wenn ein Thread benötigt wird. Ein weiteres Modul mit der Bezeichnung DFHD2CO, das in einem anderen Tasksteuerblock ausgeführt wird, bearbeitet die Aspekte der allgemeinen CICS-Db2-Verbindung, einschließlich der Identifizierung für Db2 und des Unterbrechens der Verbindung zwischen CICS und Db2.

Die maximale Anzahl der offenen Tasksteuerblöcke, die gleichzeitig Threads in Db2 ausführen können, wird durch den Parameter TCBLIMIT des DB2CONN-Objekts gesteuert. Ein offener Tasksteuerblock, der einen Thread ausführt, wird nicht beendet, wenn der Thread beendet wird. Ein offener Tasksteuerblock kann in folgenden Fällen beendet werden:

- Für eine CICS-Transaktion wird über CICS eine erzwungene Bereinigung (Force Purge) ausgeführt und der Thread ist in Db2 immer noch aktiv. In diesem Fall wird der Tasksteuerblock beendet, um die Anforderung aus Db2 zu löschen. Die aktuelle Arbeitseinheit in Db2 wird zurückgesetzt.
- Für CICS ist das Ergebnis der Arbeitseinheit unbestätigt, da der Kontakt mit dem entsprechenden Koordinator verloren ist. Durch Beenden des Tasksteuerblocks wird bewirkt, dass Db2 den Thread freigibt, die Arbeitseinheit jedoch als unbestätigt beibehält und die zugehörigen Sperren beibehält. Die Arbeitseinheit wird durch eine spätere Resynchronisation abgeschlossen, wenn CICS den Kontakt zum zugehörigen Koordinator wieder einrichtet.
- Der CICS-Dispatcher, bei dem offene Tasksteuerblöcke zurückgegeben werden, wenn sie nicht von der CICS-Db2-Anschlussfunktion verwendet werden, bereinigt nicht verwendete Tasksteuerblöcke nach 30 Minuten.

Wartezustandstypen für CICS-Db2

Der taskbezogene CICS-Db2-Benutzerexit DFHD2EX1 gibt **WAIT_MVS**-Aufrufe für verschiedene Zwecke aus. Diese Aufrufe werden durch ihren Ressourcennamen und ihre Typwerte unterschieden. Der Ressourcenname und die Ressourcentypwerte sind im Dispatcherabschnitt eines CICS-Systemspeicherauszugs sichtbar und sie werden auch in der Anzeige **CEMT INQUIRE TASK** als Wert 'Hty' bzw. 'Hva' angezeigt.

Der taskbezogene CICS-Db2-Benutzerexit und die Anforderung an Db2 werden in einem offenen L8-Tasksteuerblock ausgeführt. Die CICS-Task wird nicht in den Wartezustand des CICS-Dispatchers versetzt, wenn Sie in Db2 aktiv ist; das heißt, es gibt keine entsprechenden Informationen im Dispatcherabschnitt des CICS-Systemspeicherauszugs. Die Anzeige **CEMT INQUIRE TASK** zeigt, dass die CICS-Db2-Task in einem offenen Tasksteuerblock ausgeführt wird. Um zu ermitteln, welche Task in Db2 aktiv ist, verwenden Sie den Befehl **DSNC DISPLAY TRAN**. Mit diesem Befehl werden alle zurzeit verwendeten Threads sowie die jeweilige Task angezeigt, der die einzelnen Threads zugeordnet sind. In [Ausgabe von DISPLAY PLAN oder TRAN](#) finden Sie ein Beispiel für die Befehlsausgabe. Wenn der der Task zugeordnete Thread im Feld 'S' den Status '*' aufweist, zeigt dies, dass der Thread nur in Db2 aktiv ist. Die Task wird daher in Db2 entweder ausgeführt oder sie wartet.

Die vollständige Liste der Wartezustände, die vom taskbezogenen CICS-Db2-Benutzerexit DFHD2EX1 ausgegeben werden, deren Bedeutung und die Angabe, ob die Task aus diesem Wartezustand bereinigt oder erzwungen bereinigt (Force Purge) werden kann, ist in [Tabelle 11 auf Seite 152](#) angegeben. Eine ausführlichere Erläuterung der einzelnen Wartezustände folgt auf diese Tabelle.

<i>Tabelle 11. Wartezustände (WAITs), die von CICS-Db2 TRUE ausgegeben werden</i>			
Ressourcentyp oder Wert für 'Hty'	Ressourcenna-me oder Wert für 'Hva'	Bedeutung	Bereinigen und erzwungen Bereinigen
DB2	LOT_ECB	CICS-Task wartet auf Db2, das heißt, sie wartet darauf, dass die CICS-Db2-Task die Anforderung beendet. Wird verwendet, wenn CICS mit DB2 Version 5 oder früher verbunden ist.	Bereinigen: Nein. Erzwungen bereinigen: Ja
CDB2RDYQ	Name von DB2ENTRY oder *POOL	CICS-Task wartet auf einen verfügbaren Thread. Die Details zum Namen der Ressource, für die das DB2ENTRY-Objekt oder der Pool einen Engpass an Threads hat.	Nein
CDB2CONN		Die CICS-Task hat einen offenen Tasksteuerblock, wartet jedoch darauf, dass eine Db2-Verbindung verfügbar wird, damit sie den offenen Tasksteuerblock verwenden kann. Dies gibt an, dass der Wert von TCBLIMIT erreicht wurde und dass die maximal zulässige Anzahl offener Tasksteuerblöcke (und damit Verbindungen) für den Zugriff auf Db2 verwendet wird.	Nein

CDB2RDYQ gibt an, dass der Wert für THREADLIMIT eines DB2ENTRY-Objekts oder des Pools überschritten wurde und dass für THREADWAIT die Einstellung YES festgelegt ist; dies gibt an, dass die Task warten soll. Die Bereinigung der Task in diesem Zustand ist nicht zulässig. Wenn Sie versuchen, eine erzwungene Bereinigung (Force Purge) für die Task auszuführen, wird die Nachricht DFHAP0604 an die Konsole ausgegeben und die erzwungene Bereinigung wird verzögert, bis ein Thread angefordert wurde und die Task nicht mehr in die Warteschlange für einen Thread eingereiht ist. Anstatt die Task zu bereinigen, können Sie den Befehl SET DB2ENTRY() THREADLIMIT(n) verwenden, um die Anzahl der für das DB2ENTRY-Objekt verfügbaren Threads zu erhöhen (wenn es sich um den Pool handelt, verwenden Sie SET DB2CONN THREADLIMIT(n)). CICS übergibt Tasks für den erneuten Versuch, einen Thread anzufordern, wenn der Grenzwert für den Thread erhöht wurde.

CDB2CONN gibt an, dass die CICS-Task einen offenen Tasksteuerblock aus dem Pool der offenen L8- und L9-Tasksteuerblocks abgerufen hat, jedoch darauf wartet, dass eine Db2-Verbindung verfügbar wird, die mit dem offenen Tasksteuerblock verwendet werden kann. Dies zeigt, dass der im DB2CONN-Objekt angegebene Wert für TCBLIMIT erreicht wurde; dadurch wird die Anzahl der offenen Tasksteuerblöcke (und damit Verbindungen) begrenzt, die für den Zugriff auf Db2 verwendet werden können. Ist der durch TCBLIMIT festgelegte Grenzwert erreicht, kann der taskbezogene CICS-Db2-Benutzerexit einen Tasksteuerblock aus dem Pool der offenen L8- und L9-Tasksteuerblöcke abrufen, er muss jedoch warten, bevor der Tasksteuerblock zum Ausführen eines Threads in Db2 verwendet werden kann. Wenn eine weitere Task ihren L8- oder L9-Tasksteuerblock nicht mehr zum Ausführen eines Threads in Db2 verwendet, und die zur Ausführung von Threads verwendete Anzahl von Tasksteuerblöcken unter die durch TCBLIMIT festgelegte Anzahl fällt, wird für die wartende Task zugelassen, dass sie ihren eigenen zugehörigen L8- oder L9-Tasksteuerblock zum Ausführen eines Threads in Db2 verwendet.

Die Bereinigung der Task ist für den Wartezustand CDB2CONN nicht zulässig. Wenn Sie versuchen, eine erzwungene Bereinigung (Force Purge) für die Task auszuführen, wird die Nachricht DFHAP0604 an die Konsole ausgegeben und die erzwungene Bereinigung wird verzögert, bis eine Db2-Verbindung angefordert wurde. Anstatt die Task zu bereinigen, können Sie den Befehl SET DB2CONN TCBLIMIT verwenden, um den Wert für TCBLIMIT zu erhöhen, wodurch die Anzahl der offenen Tasksteuerblöcke erhöht wird, für

die ein Zugriff auf Db2 zulässig ist. Wenn Sie den Wert für TCBLIMIT erhöhen, übergibt CICS Tasks für den erneuten Versuch, eine Db2-Verbindung anzufordern.

Wenn der automatisch von CICS für die Anzahl der offenen L8- und L9-Tasksteuerblöcken festgelegte Grenzwert erreicht ist, sodass keine weiteren offenen Tasksteuerblöcke erstellt werden können, wird die Task mithilfe von HTYPE(DISPATCH) und HVALUE(OPEN_TCB) ausgesetzt. CICS legt diesen Grenzwert mithilfe der Formel $(2 * \text{MXT_value}) + 32$ mit dem Wert MXT oder MAXTASKS für die CICS-Region fest. Die Task wird mit HTYPE(CDB2CONN) ausgesetzt, wenn die Situation auftritt, dass dieser Grenzwert nicht überschritten wird, jedoch der Wert für TCBLIMIT überschritten wird.

In [Tabelle 12 auf Seite 153](#) finden Sie Details zu Wartezuständen (WAITS), die mithilfe von WAIT_OLDC-Dispatcheraufrufen ausgegeben wurden, wobei der Ereignissteuerblock manuell übergeben wird:

<i>Tabelle 12. Wartezustände (WAITS, die mithilfe von WAIT_OLDC-Dispatcheraufrufen ausgegeben werden)</i>			
Ressourcentyp oder Wert für 'Hty'	Ressourcenna-me oder Wert für 'Hva'	Bedeutung	Bereinigen und erzwun-gen Bereinigen
DB2_INIT		Das CICS-Db2-Initialisierungsprogramm DFHD2IN1 gibt den Wartezustand aus, um auf die Beendigung der Task für die CICS-Systemi-nialisierung zu warten, die das Programm DFHD2IN2 ausführt.	Ja
DB2CDISC	Name des DB2CONN-Ob-jekts	Es wurde der Befehl SET DB2CONN NOTCON-NECTED mit der Option WAIT oder FORCE ab-gesetzt. DFHD2TM wartet darauf, dass der Zähler für Tasks, die Db2 verwenden, null er-reicht.	Ja
DB2EDISA	Name des DB2ENTRY-Ob-jekts	Es wurde der Befehl SET DB2ENTRY DISAB-LED mit der Option WAIT oder FORCE abge-setzt. DFHD2TM wartet darauf, dass der Zäh-ler für Tasks, die das DB2ENTRY-Objekt ver-wenden, null erreicht.	Ja

In [Tabelle 13 auf Seite 153](#) finden Sie Details zu EXEC CICS WAIT EXTERNAL-Anforderungen, die von der CICS-Db2-Anschlussfunktion ausgegeben wurden.

<i>Tabelle 13. EXEC CICS WAIT EXTERNAL-Anforderungen, die von der Anschlussfunktion ausgegeben wur-den</i>			
Ressourcentyp oder Wert für 'Hty'	Ressourcenna-me oder Wert für 'Hva'	Bedeutung	Bereinigen und erzwun-gen Bereinigen
USERWAIT	CDB2TIME	Das CICS-Db2-Programm für Service-Tasks DFHD2EX2 befindet sich im Wartezyklus des zugehörigen Zeitgebers und wartet entweder darauf, dass der Bereinigungszyklus für den geschützten Thread abgerufen wird oder dass er für ein anderes Ereignis übergeben wird.	Ja
USERWAIT	DB2START	Das CICS-Db2-Serviceprogramm DFHD2EX2 wartet darauf, dass Db2 es übergibt, wenn es aktiv wird.	Ja

In [Tabelle 14 auf Seite 154](#) finden Sie Details zu EXEC CICS WAIT EVENT-Anforderungen, die von der CICS-Db2-Anschlussfunktion ausgegeben wurden:

Tabelle 14. EXEC CICS WAIT EVENTT-Anforderungen

Modul	Ressourcenname	Bedeutung
DFHD2EX2	PROTTERM	Für einen geschützten Thread wurde während des Bereinigungszyklus für den geschützten Thread der Aufruf TERM abgesetzt.
DFHD2STP	CEX2TERM	Das CICS-Db2-Programm für den Systemabschluss wartet darauf, dass die CICS-Db2-Service-Task CEX2, die das Programm DFHD2EX2 ausführt, alle Subtasks beendet und dann selbst beendet wird.

Nachrichten für CICS Db2

Nachrichten, die von der CICS-Db2-Anschlussfunktion ausgegeben werden, verwenden das Präfix DFHDB, das auch für DBCTL-Nachrichten von CICS verwendet wird. Die Nachrichtennummern umfassen den Bereich 2000 bis 2999 und sind für CICS Db2 reserviert. Dadurch haben Nachrichten der CICS-Db2-Anschlussfunktion das Format DFHDB2xxx.

Wenn möglich, wurden Nachrichtennummern aus früheren Releases der Anschlussfunktion beibehalten. Die Nachricht DFHDB2023 dokumentiert z. B. dieselbe Bedingung wie die alten Nachrichten DSN2023 oder DSN2023. Die Inhalte der Nachricht wurden jedoch geändert. Alle Nachrichten enthalten beispielsweise das Datum, die Uhrzeit und die Anwendungs-ID.

Das Ziel für Nachrichten mit transienten Daten, die von der CICS-Db2-Anschlussfunktion ausgegeben werden, wird mithilfe der Parameter MSGQUEUE1, MSGQUEUE2 und MSGQUEUE3 der DB2CONN-Definition gesteuert. Eine Nachricht kann an drei Warteschlangen mit transienten Daten weitergeleitet werden. Standardmäßig werden Nachrichten an eine einzelne Warteschlange mit dem Namen CDB2 gesendet, die für die Warteschlange CSSL als indirekt definiert ist.

Alle Nachrichten werden in [CICS-Nachrichten](#) dokumentiert und sind mithilfe der von CICS bereitgestellten CMAC-Transaktion verfügbar.

Trace für CICS Db2

Die CICS-Db2-Anschlussfunktion verwendet Tracepunkte der AP-Domäne im Bereich von 3100 bis 33FF.

Die Inhalte aller Tracepunkte, die von der CICS-Db2-Anschlussfunktion ausgegeben werden, sind in [CICS-Db2-Tracepunkte](#) dokumentiert.

Die standardmäßige Traceerstellung, die von der CICS-Db2-Anschlussfunktion ausgeführt wird, wird von den RA-Trace-Flags (RA – Resource Manager Adapter, Ressourcenmanageradapter) und den RI-Trace-Flags (RMI – Resource Manager Interface, Ressourcenmanagerschnittstelle) gesteuert. Der RMI-Trace steuert Folgendes:

- Traceausgabe des taskbezogenen CICS-Db2-Benutzerexits DFHD2EX1
- Traceausgabe aus dem CICS-Db2-Threadprozessor DFHD2D2

Jegliche andere Standardtraceerstellung der Anschlussfunktion wird mithilfe der Tracefunktion des Ressourcenmanageradapters (Resource Manager Adapter) gesteuert. Ein großer Anteil des möglichen Trace, der von der CICS-Db2-Anschlussfunktion ausgegeben wird, ist das Ausnahmentracing, das unabhängig davon ausgegeben wird, ob das RI- oder RA-Tracing aktiv ist. Sie können die erforderlichen Stufen des RA- und RI-Trace festlegen, indem Sie die von CICS bereitgestellten Transaktion CETR verwenden oder die Systeminitialisierungsparameter **STNTRRA** und **STNTRRI** festlegen.

Die CICS-Db2-Traceausgabe wird in die CICS-Tabelle für interne Traces und an die Ziele für den Hilfstrace und den GTF-Trace geschrieben, falls diese aktiv sind.

Beispiel für eine Traceausgabe der Ressourcenmanagerschnittstelle (RMI) und von CICS Db2 TRUE

Dies ist eine Beispiel-Traceausgabe der Ressourcenmanagerschnittstelle (RMI) und des taskbezogenen CICS-Db2-Benutzerexits DFHD2EX1, wenn das RI-Tracing der Stufe 1 und 2 aktiv ist. Der Trace zeigt eine SQL-Anweisung, die ausgeführt wird. In diesem Beispiel wird der taskbezogene CICS-Db2-Benutzerexit in einem offenen L8- oder L9-Tasksteuerblock ausgeführt und verwendet denselben Tasksteuerblock, um Threads in Db2 auszuführen.

```
AP 2520 ERM ENTRY PLI-APPLICATION-CALL-TO-TRUE(DSNCSQL )
      TASK-00035 KE_NUM-003F TCB-QR /007C5B60 RET-99902F92 TIME-08:32:00.7673270795 INTER
VAL-00.0000135312      =000162=
      1-0000 01
*.
      2-0000 C4E2D5C3 E2D8D340
*DSNCSQL
      3-0000 B60AF030 A54A8EC0
      {
      4-0000 E3C5E2E3 D7F0F540 00000001 00100000 19900000 999000E0 000114E0 00000000
*TESTP05 .....I.....*
      0020 00000020 00007BB0 00000000 98400000 0040000A 00101100 00000000 00000000
*.....#. ....q .....*
      0040 00000000 00000000 00000000
*.....
DS 0002 DSAT ENTRY - FUNCTION(CHANGE_MODE) MODENAME(L8)
      TASK-00035 KE_NUM-003F TCB-QR /007C5B60 RET-8009E318 TIME-08:32:00.7673341108 INTER
VAL-00.0000070312      =000163=
      1-0000 00980000 00000003 00000001 00000000 A0000040 00000000 05000102 01000002
*.q.....*
      0020 01007450 00000000 000002B0 18CCB490 18CCB6D8 9801014C 01031498 18220F30
*...&.....Qq.<...q....*
      0040 18CCBB70 1833E000 9800E8F0 00007000 190441EC 18CCBB70 00000008 D3F8D8D9
*.....\..q.Y0.....L8QR*
      0060 18CCB638 FFFFFFFF 18CB4B00 987D9520 00000001 18DEA4A0 18DEA010 00000001
*.....q'n.....u.....*
      0080 00007550 18CCB776 00000000 0000E8F0 18CCBA28 18CCBA28
*...&.....Y0.....
DS 0003 DSAT EXIT - FUNCTION(CHANGE_MODE) RESPONSE(OK)
      TASK-00035 KE_NUM-003F TCB-L8000/007C6B90 RET-8009E318 TIME-08:32:00.7673710483 INTER
VAL-00.0000369375      =000164=
      1-0000 00980000 00000003 00000001 00000000 A0000040 00000000 05000102 01000002
*.q.....*
      0020 01007450 00000000 000002B0 18CCB490 18CCB6D8 9801014C 01031498 18220F30
*...&.....Qq.<...q....*
      0040 18CCBB70 1833E000 9800E8F0 00007000 190441EC 18CCBB70 00000001 D3F8D8D9
*.....\..q.Y0.....L8QR*
      0060 18CCB638 FFFFFFFF 18CB4B00 987D9520 00000001 18DEA4A0 18DEA010 00000001
*.....q'n.....u.....*
      0080 00007550 18CCB776 00000000 0000E8F0 18CCBA28 18CCBA28
*...&.....Y0.....
AP 2522 ERM EVENT PASSING-CONTROL-TO-OPENAPI-TRUE(DSNCSQL )
      TASK-00035 KE_NUM-003F TCB-L8000/007C6B90 RET-99902F92 TIME-08:32:00.7673729077 INTER
VAL-00.0000018593      =000165=
      1-0000 01
*.
      2-0000 C4E2D5C3 E2D8D340
*DSNCSQL
      3-0000 B60AF030 A54A8EC0
      {
      4-0000 18CCB654 0006F004 19044204 00000000 007A8000 007A8000 18F0005C 18F0CCD0
*.....0.....:.....0.*.0.**
      0020 190441C0 19044210 19044206 18F000D0 19044200 18CCB490 190441C8 190441B0 *...
{.....0.*}.....H.....*
      0040 190441AC 190441EC 18CCB5F3 190441FD 18CCB5F4 190441B4 190441B2 00020000
*.....3.....4.....*
      5-0000 D3F8
*L8
AP 3180 D2EX1 ENTRY - APPLICATION REQUEST - EXEC SQL FETCH
      TASK-00035 KE_NUM-003F TCB-L8000/007C6B90 RET-8009E4D0 TIME-08:32:00.7673780483 INTER
VAL-00.0000051406      =000166=
      1-0000 02000000 00280800 001EE3C5 E2E3D7F0 F5401663 1BA501CD D1DC0002 18F0BA40
*.....TESTP05 ...v..J....0.*
      0020 00000000 18F0CDB0 018D0004
*.....0.....
      2-0000 00DE6EC4 C6C8C4F2 D3D6E340 40404040 E7D7F0F5 184AC080 18DF2D78 18DCD030 *...>DFHD2LOT
XP05.${.....}.*
      0020 00000000 0006EFF0 00000000 18DF2E0C 18F0CDE0 00000000 00000000 FF6CD800
*.....0.....0.\.....%Q.*
      0040 00000000 00000000 00000000 00000000 00000000 E3C5E2E3 D7F0F540 000C010C
*.....TESTP05 .....*
      0060 A000C000 00000000 00000000 00000000 C9E8D2F2 E9F2C7F1 B60AF030 A54A8EC0 *..
```

```

{.....IYK2Z2G1..0.v$.}*
0080 D1E3C9D3 D3C9F140 40404040 40404040 00000000 00000000 C7C2C9C2 D4C9E8C1 *JTIL
LI1 .....GBIBMIYA*
00A0 C9E8C1D8 E3C3F0F3 0AF030A5 4A8EC6D9 C2400003 000118F0 CD680000 00000000 *IYAQTC03.0.v
$.FRB .....0.....*
00C0 00000000 00000000 39050001 04000000 00000000 00000000 00000000
*.....*

```

AP 3250 D2D2 ENTRY - FUNCTION(DB2_API_CALL) CSUB_TOKEN(18DCD030)

```

TASK-00035 KE_NUM-003F TCB-L8000/007C6B90 RET-98E3321C TIME-08:32:00.7674652827 INTER
VAL-00.0000872343 =000167=
1-0000 00200000 00000034 00000000 00000000 B8000000 00000000 03000100 18DCD030
*.....*
2-0000 03006EC4 C6C8C4F2 C3E2C240 40404040 B60AF031 FAD0CB43 18D305E0 18DF2D78
*..>DFHD2CSB ..0..}...L..\.....*
0020 19044210 007C6B90 00000000 00000000 B60AF030 A54A8EC0 00000000 00000000 *.....@,.....0.v
$.{.....*
0040 18DF2DE4 00000000 00000000 00000000 00000000 00000000 00000000
*...U.....*
0060 00000000 00000000 00000000 E3C5E2E3 D7F0F540 D1E3C9D3 D3C9F140 40404040 *.....TESTP05
JTILLI1 *
0080 40404040 C5D5E3D9 E7D7F0F5 F0F0F0F1 00000000 B60AF032 01359743 C7C2C9C2 *
ENTRXP050001.....0...p.GBIB*
00A0 D4C9E8C1 C9E8C1D8 E3C3F0F3 0AF030A5 4A8E0000 44800000 00000001 00000000 *MIYAIYAQTC03.0.v
$......*
00C0 00000000 00000000 00000000 40404040 40404040 40404040 40404040 FF6CD800
*.....%Q.*
00E0 C6D9C240 00030001 18F0CD68 00000000 00000000 00000000 00003905 00010400
*FRB .....0.....*
0100 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0120 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0140 00000000 00000000 00000000 00000000 00000000 00000000 98DCD110 00000000
*.....q.J.....*
0160 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0180 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
01A0 00000000 00000000 40404040 40404040 40404040 40404040 40404040
*.....*
01C0 40404040 40404040 40404040 40404040 40404040 40404040 00000000
*.....*
01E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0200 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0220 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....K.*
0240 6E6EE399 81838540 E2A38199 A3406E6E 0100035C C9C4C5D5 00000000 00000000 *>>Trace Start
>>...*IDEN.....*
0260 0200035C E2C9C7D5 00000000 00000000 0300035C C3E3C8C4 00000000 00000000
*...*SIGN.....*CTHD.....*
0280 0400035C C1D7C940 00000000 00000000 00000000 00000000 00000000 00000000
*...*API .....*
02A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
02C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
02E0 00000000 00000000 00000000 00000000 4C4CE399 81838540 C5958440 40404C4C
*.....<<Trace End <<*
3-0000 00DE6EC4 C6C8C4F2 D3D6E340 40404040 E7D7F0F5 184AC080 18DF2D78 18DCD030 *..>DFHD2LOT
XP05.{$.....}*
0020 00000000 0006EFF0 00000000 18DF2E0C 18F0CDE0 00000000 00000000 FF6CD800
*.....0.....%Q.*
0040 00000000 00000000 00000000 00000000 00000000 E3C5E2E3 D7F0F540 010C010C
*.....TESTP05 .....*
0060 A000C000 00000000 00000000 00000000 C9E8D2F2 E9F2C7F1 B60AF030 A54A8EC0 *..
{.....IYK2Z2G1..0.v$.}*
0080 D1E3C9D3 D3C9F140 40404040 40404040 00000000 00000000 C7C2C9C2 D4C9E8C1 *JTIL
LI1 .....GBIBMIYA*
00A0 C9E8C1D8 E3C3F0F3 0AF030A5 4A8EC6D9 C2400003 000118F0 CD680000 00000000 *IYAQTC03.0.v
$.FRB .....0.....*
00C0 00000000 00000000 39050001 04000000 00000000 00000000 00000000
*.....*

```

AP 326C D2D2 EVENT - ABOUT_TO_ISSUE_DB2_API_REQUEST

```

TASK-00035 KE_NUM-003F TCB-L8000/007C6B90 RET-98E3321C TIME-08:32:00.7674700952 INTER
VAL-00.0000048125 =000168=
1-0000 98DCD110 00000000 00000000 00000000 00000000 00000000 00000000
*q.J.....*
2-0000 C6D9C240 00030001 18F0CDE0 00000000 00000000 00000000 00003905 00010400
*FRB .....0..\.....*
0020 00000000 00000000 00000000 00000000
*.....*
3-0000 E3C5E2E3 D7F0F540
*TESTP05 *
4-0000 03006EC4 C6C8C4F2 C3E2C240 40404040 B60AF031 FAD0CB43 18D305E0 18DF2D78
*..>DFHD2CSB ..0..}...L..\.....*

```

\$.{.....*	0020	19044210	007C6B90	00000000	00000000	B60AF030	A54A8EC0	00000000	00000000	*.....@,.....0.v
...U.....	0040	18DF2DE4	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
JTILLI1 *	0060	00000000	00000000	00000000	E3C5E2E3	D7F0F540	D1E3C9D3	D3C9F140	40404040	*.....TESTP05
ENTRXP050001.....	0080	40404040	C5D5E3D9	E7D7F0F5	F0F0F0F1	00000000	B60AF032	01359743	C7C2C9C2	*
0...p.GBIB*	00A0	D4C9E8C1	C9E8C1D8	E3C3F0F3	0AF030A5	4A8E0000	44800000	00000001	00000000	*MIYAIYAQTC03.0.v
\$.{.....*	00C0	00000000	00000000	00000000	40404040	40404040	40404040	40404040	FF6CD800	
.....	00E0	C6D9C240	00030001	18F0CDE0	00000000	00000000	00000000	00003905	00010400	
FRB0.\.....	0100	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
.....	0120	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
.....	0140	00000000	00000000	00000000	00000000	00000000	00000000	98DCD110	00000000	
.....q.J.....	0160	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
.....	0180	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
.....	01A0	00000000	00000000	40404040	40404040	40404040	40404040	40404040	40404040	
.....	01C0	40404040	40404040	40404040	40404040	40404040	40404040	40404040	00000000	
.....	01E0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
.....	0200	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
.....	0220	00000000	00000000	00000000	00000000	00000000	00000000	00000004	18DCD2B0	
.....K.	0240	6E6EE399	81838540	E2A38199	A3406E6E	0100035C	C9C4C5D5	00000000	00000000	*>>Trace Start
>>...*IDEN.....*	0260	0200035C	E2C9C7D5	00000000	00000000	0300035C	C3E3C8C4	00000000	00000000	
*...*SIGN.....*CTHD.....*	0280	0400035C	C1D7C940	00000000	00000000	00000000	00000000	00000000	00000000	
*...*API.....*	02A0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
.....	02C0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
.....	02E0	00000000	00000000	00000000	00000000	4C4CE399	81838540	C5958440	40404C4C	
.....<<Trace End <<	5-0000	00DE6EC4	C6C8C4F2	D3D6E340	40404040	E7D7F0F5	184AC080	18DF2D78	18DCD030	*...>DFHD2LOT
XP05.\${}.....*.	0020	00000000	0006EFF0	00000000	18DF2E0C	18F0CDE0	00000000	00000000	FF6CD800	
.....0.....	0040	00000000	00000000	00000000	00000000	00000000	E3C5E2E3	D7F0F540	010C010C	
.....TESTP05.....	0060	A000C000	00000000	00000000	00000000	C9E8D2F2	E9F2C7F1	B60AF030	A54A8EC0	*..
{.....IYK2Z2G1..0.v\$.{*	0080	D1E3C9D3	D3C9F140	40404040	40404040	00000000	00000000	C7C2C9C2	D4C9E8C1	*JTILLI
LI1.....GBIBMIYA*	00A0	C9E8C1D8	E3C3F0F3	0AF030A5	4A8EC6D9	C2400003	000118F0	CD680000	00000000	*IYAQTC03.0.v
\$.FRB0.....*	00C0	00000000	00000000	39050001	04000000	00000000	00000000	00000000	0000	
.....										

AP 326D D2D2 EVENT - RETURN_FROM_DB2_API_REQUEST

TASK-00035 KE_NUM-003F TCB-L8000/007C6B90 RET-98E3321C TIME-08:32:00.7678738139 INTER	VAL-00.0004037187	=000169=	1-0000	98DCD110	00000000	00000000	00000000	00000000	00000000	00000000
q.J.....	2-0000	C6D9C240	00030001	18F0CDE0	00000000	00000000	00000000	00003905	00010400	
FRB0.\.....	0020	00000000	00000000	00000000	00000000					
.....	3-0000	E3C5E2E3	D7F0F540							
*TESTP05	4-0000	03006EC4	C6C8C4F2	C3E2C240	40404040	B60AF031	FAD0CB43	18D305E0	18DF2D78	
...>DFHD2CSB	..0..{	L.\.....*	0020	19044210	007C6B90	00000000	00000000	00000000	00000000	*.....@,.....0.v
\$.{.....*	0040	18DF2DE4	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
...U.....	0060	00000000	00000000	00000000	E3C5E2E3	D7F0F540	D1E3C9D3	D3C9F140	40404040	*.....TESTP05
JTILLI1 *	0080	40404040	C5D5E3D9	E7D7F0F5	F0F0F0F1	00000000	B60AF032	01359743	C7C2C9C2	*
ENTRXP050001.....	0...p.GBIB*	00A0	D4C9E8C1	C9E8C1D8	E3C3F0F3	0AF030A5	4A8E0000	44800000	00000001	00000000
\$.{.....*	00C0	00000000	00000000	00000000	40404040	40404040	40404040	40404040	FF6CD800	
.....	00E0	C6D9C240	00030001	18F0CDE0	00000000	00000000	00000000	00003905	00010400	
FRB0.\.....	0100	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	
.....										

```

0120 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0140 00000000 00000000 00000000 00000000 00000000 00000000 98DCD110 00000000
*.....q.J.....*
0160 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0180 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
01A0 00000000 00000000 40404040 40404040 40404040 40404040 40404040 40404040
*.....*
01C0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 00000000
*.....*
01E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0200 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0220 00000000 00000000 00000000 00000000 00000000 00000000 00000005 18DCD2C0
*.....K{*
0240 6E6EE399 81838540 E2A38199 A3406E6E 0100035C C9C4C5D5 00000000 00000000 *>>Trace Start
>>...*IDEN.....*
0260 0200035C E2C9C7D5 00000000 00000000 0300035C C3E3C8C4 00000000 00000000
*...*SIGN.....*CTHD.....*
0280 0400035C C1D7C940 00000000 00000000 0500035C C1D7C940 00000000 00000000
*...*API .....*API .....*
02A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
02C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
02E0 00000000 00000000 00000000 00000000 4C4CE399 81838540 C5958440 40404C4C
*.....<<Trace End <<*

5-0000 00DE6EC4 C6C8C4F2 D3D6E340 40404040 E7D7F0F5 184AC080 18DF2D78 18DCD030 *..>DFHD2LOT
XP05.${.....}.*
0020 00000000 0006EFF0 00000000 18DF2E0C 18F0CDE0 00000000 00000000 FF6CD800
*.....0.....0.\.....%Q.*
0040 00000000 00000000 00000000 00000000 00000000 E3C5E2E3 D7F0F540 010C010C
*.....TESTP05 .....*
0060 A000C000 00000000 00000000 00000000 C9E8D2F2 E9F2C7F1 B60AF030 A54A8EC0 *..
{.....IYK2Z2G1..0.v$.{*
0080 D1E3C9D3 D3C9F140 40404040 40404040 00000000 00000000 C7C2C9C2 D4C9E8C1 *JTIL
LI1 .....GBIBMIYA*
00A0 C9E8C1D8 E3C3F0F3 0AF030A5 4A8EC6D9 C2400003 000118F0 CD680000 00000000 *IYAQTC03.0.v
$.FRB .....0.....*
00C0 00000000 00000000 39050001 04000000 00000000 00000000 00000000 0000
*.....*

AP 3251 D2D2 EXIT - FUNCTION(DB2_API_CALL) RESPONSE(OK)

TASK-00035 KE_NUM-003F TCB-L8000/007C6B90 RET-98E3321C TIME-08:32:00.7697274233 INTER
VAL-00.0018536093 =000170=
1-0000 00200000 00000034 00000000 00000000 B8000000 00000000 03000100 18DCD030
*.....*
2-0000 03006EC4 C6C8C4F2 C3E2C240 40404040 B60AF031 FAD0CB43 18D305E0 18DF2D78
*..>DFHD2CSB .....L.\.....*
0020 19044210 007C6B90 00000000 00000000 B60AF030 A54A8EC0 00000000 00000000 *.....@,.....0.v
$.{.....*
0040 18DF2DE4 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*...U.....*
0060 00000000 00000000 00000000 E3C5E2E3 D7F0F540 D1E3C9D3 D3C9F140 40404040 *.....TESTP05
JTILLI1 *
0080 40404040 C5D5E3D9 E7D7F0F5 F0F0F0F1 00000000 B60AF032 01359743 C7C2C9C2 *
ENTRXP050001.....0...p.GBIB*
00A0 D4C9E8C1 C9E8C1D8 E3C3F0F3 0AF030A5 4A8E0000 44800000 00000001 00000000 *MIYAIYAQTC03.0.v
$.....*
00C0 00000000 00000000 00000000 40404040 40404040 40404040 40404040 FF6CD800
*.....%Q.*
00E0 C6D9C240 00030001 18F0CDE0 00000000 00000000 00000000 00003905 00010400
*FRB .....0.\.....*
0100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0120 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0140 00000000 00000000 00000000 00000000 00000000 00000000 98DCD110 00000000
*.....q.J.....*
0160 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0180 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
01A0 00000000 00000000 40404040 40404040 40404040 40404040 40404040 40404040
*.....*
01C0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 00000000
*.....*
01E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0200 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0220 00000000 00000000 00000000 00000000 00000000 00000000 00000005 18DCD2C0
*.....K{*
0240 6E6EE399 81838540 E2A38199 A3406E6E 0100035C C9C4C5D5 00000000 00000000 *>>Trace Start
>>...*IDEN.....*
0260 0200035C E2C9C7D5 00000000 00000000 0300035C C3E3C8C4 00000000 00000000
*...*SIGN.....*CTHD.....*
0280 0400035C C1D7C940 00000000 00000000 0500035C C1D7C940 00000000 00000000

```

```

*...*API .....*API .....*
      02A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
      02C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
      02E0 00000000 00000000 00000000 00000000 4C4CE399 81838540 C5958440 40404C4C
*.....<<Trace End <<*
      3-0000 00DE6EC4 C6C8C4F2 D3D6E340 40404040 E7D7F0F5 184AC080 18DF2D78 18DCD030 *..>DFHD2LOT
XP05.${.....}.*
      0020 00000000 0006EFF0 00000000 18DF2E0C 18F0CDE0 00000000 00000000 FF6CD800
*.....0.....0.\.....%Q.*
      0040 00000000 00000000 00000000 00000000 00000000 E3C5E2E3 D7F0F540 010C010C
*.....TESTP05 .....*
      0060 A000C000 00000000 00000000 00000000 C9E8D2F2 E9F2C7F1 B60AF030 A54A8EC0 *..
{.....IYK2Z2G1..0.v$.{*
      0080 D1E3C9D3 D3C9F140 40404040 40404040 00000000 00000000 C7C2C9C2 D4C9E8C1 *JTIL
LI1 .....GBIBMIYA*
      00A0 C9E8C1D8 E3C3F0F3 0AF030A5 4A8EC6D9 C2400003 000118F0 CDE00000 00000000 *IYAQTC03.0.v
$.FRB .....0.\.....*
      00C0 00000000 00000000 39050001 04000000 00000000 00000000 00000000 0000
*.....*

```

AP 3181 D2EX1 EXIT - APPLICATION REQUEST - EXEC SQL FETCH

```

      TASK-00035 KE_NUM-003F TCB-L8000/007C6B90 RET-8009E4D0 TIME-08:32:00.7697348608 INTER
VAL-00.0000074375 =000171=
      1-0000 02000000 00280800 001EE3C5 E2E3D7F0 F5401663 1BA501CD D1DC0002 18F0BA40
*.....TESTP05 ...v..J...0.*
      0020 00000000 18F0CDB0 018D0004
*.....0.....*
      2-0000 00DE6EC4 C6C8C4F2 D3D6E340 40404040 E7D7F0F5 184AC080 18DF2D78 18DCD030 *..>DFHD2LOT
XP05.${.....}.*
      0020 00000000 0006EFF0 00000000 18DF2E0C 18F0CDE0 00000000 00000000 FF6CD800
*.....0.....0.\.....%Q.*
      0040 00000000 00000000 00000000 00000000 00000000 E3C5E2E3 D7F0F540 010C010C
*.....TESTP05 .....*
      0060 A000C000 00000000 00000000 00000000 C9E8D2F2 E9F2C7F1 B60AF030 A54A8EC0 *..
{.....IYK2Z2G1..0.v$.{*
      0080 D1E3C9D3 D3C9F140 40404040 40404040 00000000 00000000 C7C2C9C2 D4C9E8C1 *JTIL
LI1 .....GBIBMIYA*
      00A0 C9E8C1D8 E3C3F0F3 0AF030A5 4A8EC6D9 C2400003 000118F0 CDE00000 00000000 *IYAQTC03.0.v
$.FRB .....0.\.....*
      00C0 00000000 00000000 39050001 04000000 00000000 00000000 00000000 0000
*.....*
      3-0000 00
*.....*
      4-0000 E2D8D3C3 C1404040 00000088 00000000 00004040 40404040 40404040 40404040
*SQLCA ...h.....*
      0020 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
*.....*
      0040 40404040 40404040 40404040 40404040 40404040 40404040 C4E2D540 40404040
*.....DSN.....*
      0060 00000000 00000000 00000000 FFFFFFFF 00000000 00000000 40404040 40404040
*.....*
      0080 404040F0 F0F0F0F0
00000 *
      5-0000 00C86EC4 C6C8C4F2 C5D5E340 40404040 E7D7F0F5 40404040 B60AF017 63D17AC6 *.H>DFHD2ENT
XP05 ...0..J:F*
      0020 E3C5E2E3 D7F0F540 00000000 00000000 00000000 00000000 D1E3C9D3 D3C9F140
*TESTP05 .....JTILLI1*
      0040 00808080 80000000 67E939C4 B0188B86 B60AF017 63D17AC6 0000000C 00000003
*.....Z.D...f...0..J:F.....*
      0060 00000001 00000001 00000000 00000000 00000001 00000001 00000000 00000000
*.....*
      0080 00000000 00000002 00000001 00000000 00000000 00000000 00000000 00000000
*.....*
      00A0 00000000 00000000 00000000 00000000 18DCD030 00000000 00000000 19044210
*.....*
      00C0 00000000 00000000
*.....*
      6-0000 03006EC4 C6C8C4F2 C3E2C240 40404040 B60AF031 FAD0CB43 18D305E0 18DF2D78
*..>DFHD2CSB ...0..}...L.\.....*
      0020 19044210 007C6B90 00000000 00000000 B60AF030 A54A8EC0 00000000 00000000 *.....@,.....0.v
$.{.....*
      0040 18DF2DE4 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*...U.....*
      0060 00000000 00000000 00000000 E3C5E2E3 D7F0F540 D1E3C9D3 D3C9F140 40404040 *.....TESTP05
JTILLI1 *
      0080 40404040 C5D5E3D9 E7D7F0F5 F0F0F0F1 00000000 B60AF032 01359743 C7C2C9C2 *
ENTRXP050001.....0...p.GBIB*
      00A0 D4C9E8C1 C9E8C1D8 E3C3F0F3 0AF030A5 4A8E0000 44800000 00000001 00000000 *MIYAIYAQTC03.0.v
$.{.....*
      00C0 00000000 00000000 00000000 40404040 40404040 40404040 40404040 FF6CD800
*.....%Q.*
      00E0 C6D9C240 00030001 18F0CDE0 00000000 00000000 00000000 00003905 00010400
*FRB .....0.\.....*
      0100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
      0120 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
      0140 00000000 00000000 00000000 00000000 00000000 00000000 98DCD110 00000000

```

```

*.....q.J.....*
0160 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0180 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
01A0 00000000 00000000 40404040 40404040 40404040 40404040 40404040
*.....*
01C0 40404040 40404040 40404040 40404040 40404040 40404040 00000000
*.....*
01E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0200 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0220 00000000 00000000 00000000 00000000 00000000 00000000 18DCD2C0
*.....K{*
0240 6E6EE399 81838540 E2A38199 A3406E6E 0100035C C9C4C5D5 00000000 00000000 *>>Trace Start
>>...*IDEN.....*
0260 0200035C E2C9C7D5 00000000 00000000 0300035C C3E3C8C4 00000000 00000000
*...*SIGN.....*CTHD.....*
0280 0400035C C1D7C940 00000000 00000000 0500035C C1D7C940 00000000 00000000
*...*API .....*API .....*
02A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
02C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
02E0 00000000 00000000 00000000 00000000 4C4CE399 81838540 C5958440 40404C4C
*.....<<Trace End <<*

```

AP 2523 ERM EVENT REGAINING-CONTROL-FROM-OPENAPI-TRUE(DSNCSQL)

```

TASK-00035 KE_NUM-003F TCB-L8000/007C6B90 RET-99902F92 TIME-08:32:00.7697405327 INTER□
VAL-00.0000056718 =000172=
1-0000 01
*.....*
2-0000 C4E2D5C3 E2D8D340
*DSNCSQL .....*
3-0000 B60AF030 A54A8EC0 .....*..0.v$.
{ .....*
4-0000 18CCB654 0006F004 19044204 00000000 007A8000 007A8000 18F0005C 18F0CCD0
*.....0.....:.....0.*.0.}*
0020 190441C0 19044210 19044206 18F000D0 19044200 18CCBCA0 190441C8 190441B0 *...
{.....0.}*
0040 190441AC 190441EC 18CCB5F3 190441FD 18CCB5F4 190441B4 190441B2 00020000
*.....3.....4.....*
5-0000 D3F8
*L8 .....*

```

AP 2521 ERM EXIT PLI-APPLICATION-CALL-TO-TRUE(DSNCSQL)

```

TASK-00035 KE_NUM-003F TCB-L8000/007C6B90 RET-99902F92 TIME-08:32:00.7697463608 INTER□
VAL-00.0000058281 =000173=
1-0000 01
*.....*
2-0000 C4E2D5C3 E2D8D340
*DSNCSQL .....*
3-0000 B60AF030 A54A8EC0 .....*..0.v$.
{ .....*
4-0000 E3C5E2E3 D7F0F540 00000001 00100000 19900000 999000E0 000114E0 00000000
*TESTP05 .....I.....*
0020 00000020 00007BB0 00000000 98400000 0040000A 00101100 00000000 00000000
*.....#. ....q .....*
0040 00000000 00000000 00000000
*.....*

```

CSUB-Trace

Der 'CSUB' ist der Verbindungssteuerblock, den CICS verwendet, um Threads in Db2 zu verwalten und Informationen zu ihnen auszutauschen. Es ist das CICS-Äquivalent zum Db2-Verbindungssteuerblock. Der CSUB ist mit dem Thread-Tasksteuerblock verknüpft, den CICS zum Ausführen des Threads verwendet.

In [Übersicht: Herstellung einer Verbindung von CICS zu Db2](#) finden Sie eine vollständige Erläuterung der Beziehung zwischen CSUB, Db2-Verbindungssteuerblock und Thread-Tasksteuerblock.

Neben der Verwendung des CICS-Tracings verwaltet DFHD2D2 am Ende des CSUB-Steuerblocks eine Tracetabelle, um die an Db2 gerichteten Anforderungen und die auf diese Anforderungen erhaltenen Antworten zu erfassen.

Die CSUB-Tracetabelle hat eine Länge von 160 Byte und lässt das Schreiben von zehn Einträgen mit jeweils 16 Byte zu. Die Tracetabelle wird wiederverwendet, wenn alle zehn Einträge verwendet wurden. Das Format der einzelnen Traceeinträge wird in [Tabelle 15 auf Seite 161](#) gezeigt.

Tabelle 15. Layout der Einträge der CSUB-Tracetabelle		
Byte	Inhalt	Informationen
Byte 0-3	Nummer der Traceanforderung	Vollwort-Nummer des geschriebenen Traceeintrags. Die Nummer wird verwendet, um den zuletzt geschriebenen Eintrag zu suchen.
Byte 4-7	Traceanforderung	<p>Darstellung der ausgegebenen Db2-Anforderung durch vier Zeichen. Folgende Werte sind möglich:</p> <p>ABRT - Abbruchanforderung API - SQL- oder IFI-Anforderung ASSO - Zuordnungsanforderung COMM - Festschreibanforderung CTHD - Anforderung für Threaderstellung DISS - Anforderung für Zuordnung aufheben ERRH - Anforderung für Fehlerbehandlungsroutine IDEN - Identifizierungsanforderung PREP - Erstellungsanforderung PSGN - Anforderung für teilweise Anmeldung SIGN - Anforderung für vollständige Anmeldung SYNC - Anforderung für einphasig TERM - Anforderung für Threadbeendigung TIDN - Anforderung für Identifikation beenden TSGN - Anforderung für Anmeldung beenden *REC - Wiederherstellungsroutine eingegeben</p>
Byte 8-9	Reserviert	
Byte 10-11	Rückgabecode 'frb' oder 'ifc'	Rückgabecode mit zwei Byte
Byte 12-15	Ursachencode 'frb' oder 'ifc'	Ursachencode mit vier Byte

Der Steuerblock CSUB wird in einem CICS-Systemspeicherauszug formatiert. Er wird zusätzlich zu allen Ausnahmentraces von DFHD2EX1 in Traces des taskbezogenen CICS-Benutzerexits DFHD2EX1 ausgegeben, wenn ein Trace der RI-Stufe (RMI) 2 in Traces aktiv ist.

Abbildung 38 auf Seite 162 zeigt, dass ein Identifizierungsthread, ein Anmeldungsthread und ein Erstellungsthread an Db2 ausgegeben wurden. Es gibt eine API-Anforderung, gefolgt von einer Anforderung nach einem Synchronisationspunkt und einer Anforderung für 'Zuordnung aufheben' (wodurch die Zuordnung zwischen Db2-Verbindungssteuerblock und L8-Tasksteuerblock aufgehoben wird). Die Transaktion führt nun eine weitere API-Anforderung aus, indem eine weitere Arbeitseinheit gestartet wird, und der Db2-Verbindungssteuerblock wird erneut dem L8-Tasksteuerblock zugeordnet (ASSO). Es tritt eine teilweise Anmeldung auf, um einen Abrechnungsdatensatz für die vorherige Arbeitseinheit zu erstellen. Die API-Anforderung wird nun an Db2 ausgegeben.

```

6-0000 03206EC4 C6C8C4F2 C3E2C240 40404040 CFE6D380 716D6B32 2B0A1300 2B0D36C0
*...>DFHD2CSB .WL.....i*
0020 2C6A8370 0098DC88 00000000 00000000 CFE6D381 5F3C62FE 00000000 00000000
*...c..q..h.....WLa~.....*
0040 2B0D372C 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0060 00000000 00000000 00000000 E3C5E2E3 D7F0F540 C3C9C3E2 E4E2C5D9 40404040 *.....TESTP05
CICSUSER *
0080 40404040 C5D5E3D9 E7D7F0F5 F0F0F0F1 00000000 CFE6D380 742927E2 C7C2C9C2 *
ENTRXP050001.....WL.....SGBIB*
00A0 D4C9E8C1 C9E8C3E6 E3F1F5F0 E6D37E22 16AC8000 44A00000 00000001 C3C9C3E2
*MIYAIYCWT150WL=.....CICS*
00C0 E4E2C5D9 40404040 40404040 40404040 40404040 40404040 40404040 FF4EA000
*USER .+...*
00E0 C6D9C240 00030001 2AF0DCB0 00000000 00000000 00000000 00180A05 00010000
*FRB .....0.....*
0100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0120 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0140 00000000 00000000 00000000 00000000 00000000 00000000 ACB8C110 00000000
*.....A.....*
0160 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0180 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
01A0 00000000 00000000 00000000 00000000 00000000 40404040 40404040 40404040
*.....*
01C0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
*.....*
01E0 40404040 00000001 00000048 42600048 2C6511B8 00000000 00000000 00000000
*.....-.....*
0200 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0220 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
*.....*
0240 00000000 00000000 00000000 00000000 00000000 00000000 0000000C 2CB8C2B0
*.....B.*
0260 6E6EE399 81838540 E2A38199 A3406E6E 0100067C C9C4C5D5 00000000 00000000 *>>Trace Start
>>....IDEN.....*
0280 0200067C E2C9C7D5 00000000 00000000 0300067C C3E3C8C4 00000000 00000000
*....SIGN.....CTHD.....*
02A0 0400067C C1D7C940 00000000 00000000 0500067C E2E8D5C3 00000000 00000000
*....API.....SYNC.....*
02C0 0600067C C4C9E2E2 00000000 00000000 0700067C C1E2E2D6 00000000 00000000
*....DISS.....ASS0.....*
02E0 0800067C D7E2C7D5 00000000 00000000 0900067C C1D7C940 00000000 00000000
*....PSGN.....API.....*
0300 00000000 00000000 00000000 00000000 4C4CE399 81838540 C5958440 40404C4C *.....<

```

Abbildung 38. Beispiel für einen CSUB-Trace – CICS ist mit Db2 verbunden

CSUB-Abbruchinformationen

Wenn eine CICS-Db2-Subtask im Rahmen des Wiederherstellungsprozesses abgebrochen wird, speichert CICS Informationen aus dem MVS-SDWA-Element im CSUB-Steuerblock.

Zum Beispiel die Felder CSB_SDWA_REGS (regs 0 -15) und CSB_SDWA_PSW. CSB_SDWA_REGS besteht aus den 16 Wörtern, die auf die SDWA-Strukturkennung folgen. Das Feld CSB_SDWA_PSW besteht aus den zwei Wörtern, die auf das Feld CSB_SDWA_REGS folgen. Obwohl der CSUB in der Regel mithilfe der Funktion FREEMAIN gelöscht wird, die auf den Fehler folgt, erfasst der Ausnahmebedingungs-Trace AP 319D, der zum Zeitpunkt des Fehlschlagens geschrieben wird, den CSUB-Steuerblock, der die SDWA-Informationen enthält.

Speicherauszug für CICS-Db2

Steuerblöcke der CICS-Db2-Anschlussfunktion werden in einem CICS-Systemspeicherauszug formatiert; dies erfolgt unter Steuerung eines Db2-Schlüsselworts von CICS IPCS.

Das Db2-Schlüsselwort kann folgende Werte angeben:

- 0 - keine Db2-Ausgabe
- 1 - nur Übersichtsdaten

- 2 - nur Steuerblöcke
- 3 - Übersicht und Steuerblöcke

In einem CICS-Transaktionsspeicherauszug gibt es weder Übersicht noch Steuerblöcke, aber die Tracetabelle enthält die CICS-Db2-Traceeinträge.

Ein Beispiel, das CICS-Db2-Übersichtsdaten aus einem CICS-Speicherauszug zeigt, ist in [Abbildung 39 auf Seite 163](#) enthalten. Das Beispiel bietet Informationen zum globalen Status der CICS-Db2-Verbindung und eine Übersicht der Transaktionen (unter den Überschriften 'Tran id', 'Task num', 'TcaAddr', 'TieAddr', 'LotAddr', 'Rctename', 'RcteAddr', 'CsubAddr', 'Correlation id', 'Uowid', 'Subtask running' und 'TCB in DB2'). Dieses Beispiel wurde ausgegeben, während CICS unter Verwendung der Umgebung für offene Transaktionen mit DB2 Version 6 verbunden war.

```

===DB2: CICS/DB2 - SUMMARY
===DB2: GLOBAL STATE SUMMARY
  Db2conn name:          RCTJT
  Connection status:     Connected
  In standby mode:       No
  DB2 id:                DE2D
  DB2 Group id:
  DB2 release:           0610
  Operating in OpenAPI mode: Yes
  Service task started:  Yes
  Master subtask started: No - not required
  Tcb limit:             12
  Currently active tcbs: 2
  Message Queue1:       CDB2
  Message Queue2:
  Message Queue3:
  Statistics Queue:     CDB2
  Standby mode:         Reconnect
  Connector error:      Sqlcode

===DB2: TRANSACTION SUMMARY
Tran Task TcaAddr TieAddr LotAddr Rctename RcteAddr CsubAddr Correlation Uowid Subtask TCB
id num                                     id                                     id                                     id                                     running in DB2
-----
XP05 00050 184AB680 190442F0 19044370 XP05 18DF2D78 18D10330 ENTRXP050002 B60A17A16316E1C7 N/A Yes
XP05 00048 184AC680 19044190 19044210 XP05 18DF2D78 18D10030 ENTRXP050001 B60A1794A7A9E7C4 N/A No

```

Abbildung 39. Speicherauszugsbeispiel

Db2-Threadidentifikation

Ein Thread, der in Db2 im Namen einer CICS-Transaktion ausgeführt wird, ist durch seine Korrelations-ID identifiziert, die von der CICS-Db2-Anschlussfunktion festgelegt wird.

Db2 ermöglicht, dass für die Korrelations-ID bis zu 12 Byte verwendet werden.

Das Format der Korrelations-ID mit 12 Byte wird als *eeeeettttnnnn* erstellt, wobei *eeee* COMD, POOL oder ENTR ist und angibt, ob es sich um einen Befehlsthread, Pool-Thread oder einen DB2ENTRY-Thread handelt; *tttt* ist die Transaktions-ID und *nnnn* ist eine eindeutige Nummer.

Anmerkung: Eine an Db2 übergebene Korrelations-ID kann nur geändert werden, wenn die CICS-Anschlussfunktion eine Anmeldung für Db2 ausgibt. Wenn die Anmeldung durch einen Thread wiederverwendet wird, der eine primäre Berechtigungs-ID verwendet, die während mehrerer Transaktionen konstant bleibt (beispielsweise durch Verwendung von AUTHID(name)), findet nur eine einzige Anmeldung statt. In dieser Instanz stimmt die Angabe für *tttt* in der Korrelations-ID nicht mit der ID der ausgeführten Transaktion überein. Es ist die ID der Transaktion, für die die ursprüngliche Anmeldung erfolgte.

Transaktionsabbruchcodes für CICS Db2

Die CICS-Db2-Anschlussfunktion verwendet mehrere Abbruchcodes, von denen jeder einen bestimmten Fehler eindeutig benennt.

Die Transaktionsabbruchcodes lauten AD2x oder AD3x. Sie sind in [Transaktionsabbruchcodes](#) dokumentiert und können mithilfe der von CICS bereitgestellten Transaktion CMAC verwendet werden.

Execution Diagnostic Facility (EDF) für CICS Db2

Der taskbezogene CICS-Db2-Benutzerexit DFHD2EX1 wird mithilfe des Schlüsselworts FORMATEDF aktiviert und von CICS aufgerufen, um die Anzeige für SQL-API-Anforderungen zu formatieren, wenn die Transaktion unter EDF ausgeführt wird.

Im Modus EDF führt die CICS-Db2-Anschlussfunktion Folgendes aus:

- Stoppt bei jedem EXEC-SQL-Befehl, entschlüsselt die SQL-Anweisung und zeigt sie in einer Datei in der Anzeige an
- Zeigt die Ergebnisse vor und nach der Verarbeitung von SQL-Aufrufen an
- Zeigt Folgendes an:
 - Typ der SQL-Anweisung.
 - Alle Eingabe- und Ausgabevariablen.
 - Inhalt der SQLCA (Formatierungsroutine).
 - Primäre und sekundäre Berechtigungs-IDs. (Dies hilft bei der Diagnose von SQLCODE -922.)

In einer EDF-Anzeige werden maximal 55 Variablen angezeigt; dies sind etwa zehn Bildschirmseiten. Für jede EDF-SQL-Sitzung sind 12 KB temporärer CICS-Speicher erforderlich, der beim Beenden von EDF freigegeben wird.

EDF-Anzeigen für SQL-Anweisungen werden in [Abbildung 40 auf Seite 164](#) und [Abbildung 41 auf Seite 164](#) gezeigt.

```
TRANSACTION: XC05 PROGRAM: TESTC05 TASK:0000097 APPLID: CICS41 DISPLAY:00
STATUS: ABOUT TO EXECUTE COMMAND
```

```
EXEC SQL OPEN
  DBRM=TESTC05, STMT=00221, SECT=00001
```

```
OFFSET: X'001692' LINE: UNKNOWN EIBFN=X'0E0E'
```

```
ENTER:
PF1 : UNDEFINED PF2 : UNDEFINED PF3 : UNDEFINED
PF4 : PF5 : PF6 :
PF7 : PF8 : PF9 :
PF10: PF11: UNDEFINED PF12:
```

Abbildung 40. EDF-Beispiel für die SQL-EXEC-Anzeige ('vorher')

```
TRANSACTION: XC05 PROGRAM: TESTC05 TASK:0000097 APPLID: CICS41 DISPLAY:00
STATUS: COMMAND EXECUTION COMPLETE
CALL TO RESOURCE MANAGER DSNCSQL
```

```
EXEC SQL OPEN P.AUTH=SYSADM , S.AUTH=
PLAN=TESTC05, DBRM=TESTC05, STMT=00221, SECT=00001
```

```
SQL COMMUNICATION AREA:
```

```
SQLCABC = 136 AT X'03907C00'
SQLCODE = -923 AT X'03907C04'
SQLERRML = 070 AT X'03907C08'
SQLERRMC = ' ACCESS,00000000,00000000, '... AT X'03907C0A'
SQLERRP = 'DSNAET03' AT X'03907C50'
SQLERRD(1-6) = '000, 000, 000000, 000000000000, 000000, 000' AT X'03907C58'
SQLWARN(0-A) = ' _ _ _ _ _ _ _ _ _ _ ' AT X'03907C70'
SQLSTATE = 57015 AT X'03907C7B'
```

```
OFFSET: X'001692' LINE: UNKNOWN EIBFN=X'0E0E'
```

```
ENTER: CONTINUE
PF1 : UNDEFINED PF2 : UNDEFINED PF3 : END EDF SESSION
PF4 : SUPPRESS DISPLAYS PF5 : WORKING STORAGE PF6 : USER DISPLAY
PF7 : SCROLL BACK PF8 : SCROLL FORWARD PF9 : STOP CONDITIONS
PF10: PREVIOUS DISPLAY PF11: UNDEFINED PF12: ABEND USER TASK
```

Abbildung 41. EDF-Beispiel für die SQL-EXEC-Anzeige ('nachher')

Deadlocks in der CICS-Db2-Umgebung bearbeiten

Deadlocks können in einem CICS-Db2-System zwischen zwei oder mehr Transaktionen oder zwischen einer Transaktion und einem weiteren Db2-Benutzer auftreten. An Deadlocks ist/sind eine oder zwei Ressourcen beteiligt.

Informationen zu diesem Vorgang

In diesem Abschnitt werden nur Deadlocks innerhalb von Db2 behandelt. Wenn bei diesem Deadlocktyp Db2-Ressourcen beteiligt sind, wird für einen der Partner im Deadlock das Zeitlimit entsprechend den benutzerdefinierten IRLM-Parametern überschritten. Andere mögliche Deadlocks treten auf, wenn Ressourcen außerhalb von Db2 beteiligt sind.

Es ist zu erwarten, dass Deadlocks auftreten, jedoch nicht allzu häufig. Besondere Berücksichtigung benötigen Deadlocks in den folgenden Situationen:

- Andere Transaktionen sind häufig verzögert, da sie auf Ressourcen zugreifen, die von den am Deadlock beteiligten Partnern gehalten werden. Dadurch erhöht sich die Antwortzeit dieser Transaktionen, was möglicherweise zu einer kaskadierenden Reaktion führt.
- Es ist zu erwarten, dass die an einem Deadlock beteiligten Ressourcen in Zukunft intensiver verwendet werden, da die Transaktionsrate für die am Deadlock beteiligten Transaktionen oder für andere Transaktionen erhöht ist.

Die Komponente IRLM des Db2-Subsystem führt in benutzerdefinierten Intervallen eine Deadlock-Erkennung aus. Einer der Partner des Deadlocks ist das Opfer und empfängt von Db2 den Rückgabecode -911 oder -913. Der tatsächliche Rückgabecode ist durch den Parameter DROLLBACK für das DB2CONN-Objekt (wenn eine Transaktion einen Pool-Thread verwendet) oder das DB2ENTRY-Objekt festgelegt, das von der Transaktion verwendet wird. Wenn DROLLBACK (YES) angegeben wird, leitet die Anschlussfunktion einen SYNCPOINT ROLLBACK ein, bevor die Steuerung an die Anwendung zurückgegeben wird. Darüber hinaus ändert die Anschlussfunktion den SQL-Rückgabecode, der von Db2 von -913 bis -911 zurückgegeben wird. Sie gibt -911 an die Anwendung zurück. Der andere Partner setzt die Verarbeitung fort, nachdem für das Opfer ein Rollback ausgeführt wurde.

Um Deadlock-Situationen zu beheben, müssen Sie eine Reihe von Aktivitäten ausführen. Beheben von Deadlocks bedeutet, dass an einer Position im System Änderungen anzuwenden sind, damit die Wahrscheinlichkeit von Deadlocks reduziert wird.

Häufig sind folgende Schritte erforderlich, um eine Deadlock-Situation zu beheben:

1. Deadlock erkennen.
2. Beteiligte Ressourcen suchen.
3. Beteiligte SQL-Anweisungen suchen.
4. Verwendeten Zugriffspfad suchen.
5. Ermitteln, warum der Deadlock aufgetreten ist.
6. Änderungen vornehmen, um ihn zu vermeiden.

Zugehörige Konzepte

„Fehlerbehebung für Db2 ” auf Seite 151

In diesem Abschnitt wird die Fehlerbestimmung für CICS Db2 beschrieben.

„Thread-Tasksteuerblöcke ” auf Seite 151

In der CICS-Db2-Umgebung wird jeder Thread in Db2 unter einem Thread-Tasksteuerblock ausgeführt. Dieser Abschnitt bietet weitere technische Informationen zu Thread-Tasksteuerblöcken und soll eine Hilfe bei der Fehlerbestimmung sein.

„Wartezustandstypen für CICS-Db2” auf Seite 151

Der taskbezogene CICS-Db2-Benutzerexit DFHD2EX1 gibt **WAIT_MVS**-Aufrufe für verschiedene Zwecke aus. Diese Aufrufe werden durch ihren Ressourcennamen und ihre Typwerte unterschieden. Der Ressourcenname und die Ressourcentypwerte sind im Dispatcherabschnitt eines CICS-Systemspeicherauszugs sichtbar und sie werden auch in der Anzeige **CEMT INQUIRE TASK** als Wert 'Hty' bzw. 'Hva' angezeigt.

„Nachrichten für CICS Db2 ” auf Seite 154

Nachrichten, die von der CICS-Db2-Anschlussfunktion ausgegeben werden, verwenden das Präfix DFHDB, das auch für DBCTL-Nachrichten von CICS verwendet wird. Die Nachrichtennummern umfassen den Bereich 2000 bis 2999 und sind für CICS Db2 reserviert. Dadurch haben Nachrichten der CICS-Db2-Anschlussfunktion das Format DFHDB2xxx.

„Trace für CICS Db2 ” auf Seite 154

Die CICS-Db2-Anschlussfunktion verwendet Tracepunkte der AP-Domäne im Bereich von 3100 bis 33FF.

„Speicherauszug für CICS-Db2” auf Seite 162

Steuerblöcke der CICS-Db2-Anschlussfunktion werden in einem CICS-Systemspeicherauszug formatiert; dies erfolgt unter Steuerung eines Db2-Schlüsselworts von CICS IPCS.

„Db2-Threadidentifikation ” auf Seite 163

Ein Thread, der in Db2 im Namen einer CICS-Transaktion ausgeführt wird, ist durch seine Korrelations-ID identifiziert, die von der CICS-Db2-Anschlussfunktion festgelegt wird.

„Transaktionsabbruchcodes für CICS Db2” auf Seite 163

Die CICS-Db2-Anschlussfunktion verwendet mehrere Abbruchcodes, von denen jeder einen bestimmten Fehler eindeutig benennt.

„Execution Diagnostic Facility (EDF) für CICS Db2” auf Seite 163

Der taskbezogene CICS-Db2-Benutzerexit DFHD2EX1 wird mithilfe des Schlüsselworts FORMATEDF aktiviert und von CICS aufgerufen, um die Anzeige für SQL-API-Anforderungen zu formatieren, wenn die Transaktion unter EDF ausgeführt wird.

Zwei Typen von Deadlocks

Ein Deadlock in Db2 kann auftreten, wenn zwei Transaktionen eine Sperre halten, die von der jeweils anderen Transaktion benötigt wird.

In einer Db2-Umgebung können zwei Typen von Deadlocks auftreten, wenn Folgendes auftritt:

- Zwei Ressourcen sind beteiligt. Jede Transaktion hat eine Ressource gesperrt und benötigt die andere Ressource in einem nicht kompatiblen Modus. Bei den Ressourcen handelt es sich in der Regel um Indexseiten und Datenseiten. Dies ist die klassische Deadlock-Situation.
- Nur eine Ressource ist beteiligt. Der Hauptzweck von Aktualisierungssperren (U, Update) ist die Reduzierung der Anzahl von Situationen, in denen eine *Sperrenumstufung* einen Deadlock verursacht. Die Aktualisierungssperre hat die meisten dieser Situationen behoben, aber es ist in bestimmten Situationen weiterhin möglich, dass ein Deadlock mit nur einer beteiligten Ressource vorhanden ist, da es mehrere Möglichkeiten gibt, die Ressource zu sperren.

Ein typisches Beispiel dafür ist die Situation, wenn eine Transaktion einen Cursor mit der Option ORDER BY öffnet und einen Index verwendet, um die Sortierung zu vermeiden. Wenn eine Zeile auf einer Seite abgerufen wird, erfasst Db2 für diese Seite eine gemeinsame Sperre (S, Share). Wenn die Transaktion dann eine Aktualisierung ohne einen Cursor für die zuletzt abgerufene Zeile ausgibt, wird die S-Sperre in eine exklusive Sperre (X, Exclusive) hochgestuft.

Wenn zwei solcher Transaktionen gleichzeitig ausgeführt werden und beide die gemeinsame Sperre (S) auf derselben Seite abrufen, bevor sie die exklusive Sperre (X) erfassen, tritt ein Deadlock auf.

Deadlocks erkennen

Deadlockinformationen sind im Db2-Leistungstrace oder im MVS-Protokoll verfügbar.

Informationen zu diesem Vorgang

Wenn Sie in einer normalen Umgebung, die ohne aktivierte Db2-Leistungstraces ausgeführt wird, Informationen zu einem Deadlock abrufen möchten, ist die einfachste Möglichkeit dazu, das MVS-Protokoll zu durchsuchen, um die in [Abbildung 42 auf Seite 167](#) gezeigten Nachrichten zu finden.

```
DSNT375I PLAN p1 WITH CORRELATION ID id1
AND CONNECTION ID id2 IS DEADLOCKED with
PLAN p2 WITH CORRELATION ID id3
AND CONNECTION ID id4.

DSNT501I DSNILMCL RESOURCE UNAVAILABLE
CORRELATION-ID=id1,CONNECTION-ID=id2
REASON=r-code
TYPE name
NAME name
```

Abbildung 42. Deadlock-Nachrichten

Mithilfe dieser Nachrichten werden die beiden am Deadlock beteiligten Partner identifiziert. Die Partner werden sowohl durch den Plannamen als auch durch die Korrelations-ID angegeben.

Darüber hinaus gibt eine zweite Nachricht die Ressource an, die das Opfer nicht abrufen konnte. Die andere Ressource (unabhängig davon, ob es sich dabei um dieselbe handelt) wird in der Nachricht nicht angezeigt.

Beteiligte Ressourcen suchen

Um die anderen an einem Deadlock beteiligten Ressourcen zu suchen, müssen Sie möglicherweise einen Db2-Leistungstrace aktivieren und den Deadlock reproduzieren.

Angenommen, der Grund für das Beheben des Deadlocks besteht darin, dass die Anzahl der Deadlocks zu hoch ist. Normalerweise ist es kein großes Problem, den Deadlock nach dem Start des Trace zu reproduzieren.

Beschränken Sie den Db2-Leistungstrace auf die beiden Pläne, die in der MVS-Protokollnachricht angegeben sind. Es kann auch sinnvoll sein, den Trace auf die beiden beteiligten CICS-Transaktions-IDs (Berechtigungs-IDs) zu beschränken. Schließen Sie in den Leistungstrace `class(06)` für allgemeine Sperrenereignisse und `class(03)` für SQL-Ereignisse ein. Db2 Performance Monitor (DB2PM) ist ein nützliches Tool, um die Traceausgabe zu formatieren. Der DB2PM-Bericht zu Sperrenkonflikten und der Bericht zu Sperraussetzungen kann dazu beitragen, die an dem Deadlock beteiligten Ressourcen zu ermitteln.

Wenn die Ausgabe aus den DB2PM-Berichten zu groß ist, können Sie ein Benutzerprogramm entwickeln, um die Ausgabe der Traces zu analysieren. Ziel ist es, die am Deadlock beteiligten Ressourcen sowie alle beteiligten SQL-Anweisungen zu ermitteln.

Beteiligte SQL-Anweisungen suchen

An einem Deadlock können viele SQL-Anweisungen beteiligt sein. Es ist zum Beheben des Deadlocks häufig erforderlich, alle SQL-Anweisungen zu suchen.

Informationen zu diesem Vorgang

Wenn die beteiligten Ressourcen über die Sperrentraces identifiziert sind, können Sie die beteiligten SQL-Anweisungen in einem SQL-Tracebericht suchen, indem Sie die Zeitmarken aus den beiden Traces kombinieren.

Verwendeten Zugriffspfad suchen

Verwenden Sie die EXPLAIN-Option von Db2 für die entsprechenden Pläne, um den Zugriffspfad zu suchen, der von den am Deadlock beteiligten SQL-Anweisungen verwendet wurde.

Ermitteln, aus welchem Grund der Deadlock aufgetreten ist.

Wenn Sie die am Deadlock beteiligten SQL-Anweisungen und Ressourcen identifizieren und den Zugriffspfad finden, zeigt dies Ihnen in der Regel, aus welchem Grund der Deadlock aufgetreten ist. Das Wissen darum, aus welchem Grund der Deadlock aufgetreten ist, hilft Ihnen beim Entwickeln einer Lösung. Dieser Prozess kann jedoch zeitintensiv sein.

Deadlocks beheben

Sie müssen versuchen zu verstehen, warum Deadlocks aufgetreten sind, damit Sie die richtige Aktion auswählen können, um den Deadlock zu beheben.

Informationen zu diesem Vorgang

Im Allgemeinen tritt ein Deadlock auf, weil zwei oder mehr Transaktionen dieselben Ressourcen in umgekehrter Reihenfolge gleichzeitig und in einem in Konflikt stehenden Modus verwenden wollen. Die Aktionen, die vorgenommen werden, um einen Deadlock zu verhindern, müssen diese Merkmale berücksichtigen.

In Tabelle 16 auf Seite 168 wird eine Liste mit den vorbeugenden Maßnahmen und den entsprechenden Hauptauswirkungen gezeigt.

<i>Tabelle 16. Deadlockvermeidung</i>				
Aktionen	Ressourcen verteilen	Sperrenfolge ändern	Gemeinsamen Zugriff verringern	Sperrmodus ändern
Freien Speicherbereich im Index vergrößern	X			
Größe untergeordneter Indexseiten vergrößern	X			
Freien TS-Speicherbereich erhöhen	X			
Clusterindex ändern	X	X		
Tabellenbereich reorganisieren	X	X	X	
Index hinzufügen		X	X (1)	
Index löschen		X		
Transaktionen serialisieren			X	
Zusätzliche COMMITS verwenden			X	
Antwortzeit minimieren			X	
Isolationsstufe (2) ändern			X	X
Anwendung neu entwerfen	X	X	X	X
Datenbank neu entwerfen	X	X	X	X
Hinweise: 1. Aufgrund von Änderungen im Zugriffspfad. 2. Cursorstabilität ist in der Regel besser als ein wiederholbarer Lesevorgang.				

Zum Auswählen der richtigen Aktion müssen Sie zunächst verstehen, warum der Deadlock aufgetreten ist. Sie können dann die Aktionen bewerten, um Ihre Auswahl zu treffen. Diese Aktionen können mehrere Auswirkungen haben. Sie können:

- Das Deadlock-Problem beheben
- Eine Änderung im Zugriffspfad anderer Transaktionen erzwingen, die neue Deadlocks verursachen.
- Neue Deadlocks im System verursachen.

Es ist daher wichtig, dass Sie sorgfältig den Zugriffspfad überwachen, der von den betroffenen Transaktionen verwendet wurde, beispielsweise mithilfe der Db2-Funktion EXPLAIN. In vielen Fällen ist das Beheben von Deadlocks ein iterativer Prozess.

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 Neuausgabe 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 Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

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 Geschäftsbedin-

gungen 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.

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

A

Abbrüche
 AD2x und AD3x [163](#)
 AEY9 [99](#)
 AEY9-Abbrüche vermeiden [99](#)
 Transaktionsabbruchcodes [163](#)
Abrechnung
 CICS- und Db2-Datensätze kombinieren [135](#)
 Kombinationen in einem Datensatz [138](#)
Abrechnungsinformationen, von CICS bereitgestellt
 SMF (Systemverwaltungsfunktion) [121](#)
 Statistikdaten [121](#)
 Überwachungsdaten [121](#)
Abrechnungsinformationen, von Db2 bereitgestellt
 Traces [122](#)
Abrechnungstrace [141](#)
ADDMEM, Operand [54](#)
Adressräume
 DSN1DBM1 [2](#)
 DSN1DIST [2](#)
 DSN1IRLMPROC [2](#)
 DSN1MSTR [2](#)
 DSN1SPAS [2](#)
AEY9 [99](#)
Anhängen von Gruppen
 DB2GROUPLD, Attribut [14](#)
 in Verbindung mit unbestätigten Arbeitseinheiten [29](#)
 RESYNCMEMBER [29](#)
Anhängen von Tasksteuerblöcken [20, 22](#)
Anhängen von Tasksteuerblöcken, Threads [19](#)
Anpassung
 Planwechsel, dynamisch [84](#)
Anschlussbefehle [1](#)
Anschlussfunktion überwachen
 CICS-Transaktionen [127](#)
 Funktionen [30](#)
 Leistung [124](#)
 mithilfe von CEMT-Befehlen [30](#)
 mithilfe von EXEC CICS-Befehlen [30](#)
 Tools [123](#)
Ansichten [94](#)
Anwendungsarchitektur [72](#)
Anwendungsdesign
 Abbrüche vermeiden [99](#)
 Anwendungsprogramme threadsicher machen [90](#)
 Aufeinander folgende Einfügung [97](#)
 Bindeoptionen (BIND) [114](#)
 CICS-Anwendungen konvertieren [76](#)
 CICS-Db2-Designkriterien [71](#)
 CICS-Transaktionscodes wechseln [83](#)
 Exitprogramm [82](#)
 Hold-Cursor [98](#)
 Indexspalten aktualisieren [95](#)
 Pakete [74](#)
 Pakete verwenden [74](#)
 Planexits, dynamisch [80](#)

Anwendungsdesign (*Forts.*)
 Programmablauf, tabellengesteuert [85](#)
 RETURN IMMEDIATE, Befehl [98](#)
 Seitenkonflikt [96](#)
 Sicherheit [71](#)
 Sperrstrategie [89](#)
 SQL-Anweisungen im Anwendungsdesign [93](#)
 SQL, Sprache [89](#)
 tabellengesteuertes Verfahren für den Programmablauf [85](#)
 Transaktionsgruppierung [79](#)
 Übersicht [71](#)
Arbeitseinheiten, unbestätigt
 Auflösung [28](#)
Aufeinander folgende Einfügung [97](#)
AUTHTYPE-Sicherheit [55](#)
AUTHTYPE-Werte für die Sicherheit
 Berechtigungs-ID [58](#)
AUTHTYPE-Werte für Sicherheit
 Anmelde-ID aus DB2CONN-Objekt [58](#)
 Benutzer-ID [58](#)
 Gruppen-ID [58](#)
 Terminal-ID [58](#)
 Transaktions-ID [58](#)

B

Befehl EXEC CICS RETURN IMMEDIATE [98](#)
Befehlsberechtigung
 Db2 [66](#)
Befehlserkennungszeichen [31](#)
Befehlsthreads [3](#)
Beispiel-Exitroutine für die Anmeldung (DSN3SSGN) [65](#)
Beispiel-Exitroutine für Verbindungen (DSN3SATH) [60](#)
Beispieljob, RRCODE [52](#)
Berechtigungs-IDs, primäre [62](#)
Berechtigungs-IDs, sekundäre [64](#)
Bericht der Statistikzusammenfassung [125](#)
BIND-Zeitmarken [114](#)
Bindeoptionen
 Bindeoptionen
 Cursorstabilität [114](#)
 Cursorstabilität [114](#)
 Isolationsstufe [114](#)
 Lesevorgang, wiederholbar [114](#)
 RELEASE(COMMIT) [17](#)
 Validierung [115](#)
Bindeoptionen (BIND)
 bei der Programmerstellung [114](#)
 im Anwendungsdesign [114](#)
 RETAIN [114](#)
Bindeprozess
 nach einer Programmänderung [113](#)
 Optionen und Aspekte [114](#)
 Übersicht [7](#)
BMS (Basic Mapping Support) [35](#)
Brennpunkte, Beispiele [96](#)

C

- CEMT INQUIRE-Befehle [30](#)
- CEMT SET-Befehle [30](#)
- CICS Db2
 - automatische Verbindung [27](#)
 - DFH\$DB2, Beispielgruppe [35](#)
 - Operationen [27](#)
- CICS Performance Analyzer [121](#)
- CICS-Anschlussfunktion
 - Anschlussbefehle [1](#)
 - Übersicht [1](#)
 - überwachen [124](#)
- CICS-Anwendungen konvertieren [76](#)
- CICS-Befehlssicherheit
 - VCICSCMD, allgemeine Ressourcenklasse [54](#)
- CICS-Db2-Anschlussfunktion
 - Befehlsthreads [3](#)
 - Einstiegsthreads [3](#)
 - Multithread-Verbindungen [3](#)
 - Pool-Threads [4](#)
 - Ressourcenmanagerschnittstelle (RMI) [1](#)
 - SQL-Anforderung [1](#)
 - Threads [3](#)
 - Übersicht [3](#)
- CICS-Db2-Anwendungsprogramm
 - Übersicht [6](#)
- CICS-Db2-Ressourcendefinitionen
 - Übersicht [11](#)
- CICS-Db2-Schnittstelle
 - Übersicht [1](#)
- CICS-Db2-Statistik [125](#)
- CICS-Db2-Umgebung
 - testen [109](#)
 - Vorbereitung [109](#)
- CICS-Db2-Verbindung
 - definiert mithilfe von RDO [11](#)
- CICS-Ressourcensicherheit
 - XCICSDB2, allgemeine Ressourcenklasse [52](#)
- CICS-Sicherheit [50](#)
- CICS-Systemspeicherauszug bei der Fehlerbestimmung [162](#)
- CICS-Transaktionscodes wechseln [83](#)
- CLASS 1, Prozessorzeit [141](#)
- CLASS 2, Prozessorzeit [141](#)
- CONNECTERROR, Befehl [100](#)
- CONNECTST [99](#)
- CPRMPLAN [82](#)
- CREATE TABLESPACE
 - LOCKSIZE [89](#)
- CSUB-Trace [160](#)
- CURSOR WITH HOLD, Option [98](#)
- Cursortabelle (CT) [16](#)

D

- Datenbankanforderungsmodule, Produktion von [111](#)
- Db2 überwachen [128](#)
- Db2-Abrechnungsberichte [134](#)
- Db2-Abrechnungsfunktion [132](#)
- Db2-Abrechnungsverfahren
 - Db2-Abrechnungsdaten in Beziehung setzen zu CICS-Datensätzen [135](#)
- Db2-Befehle [31](#)
- Db2-Kataloge

Db2-Kataloge (*Forts.*)

- SYSIBM.SYSDBRM, Tabelle [86](#)
- SYSPLAN und SYSDBRM [86](#)

Db2-Sicherheit

- Abrechnung [63](#)
- Berechtigung zum Ausführen eines Plans [67](#)
- Berechtigungs-IDs erstellen [62](#), [64](#)
- primäre Berechtigungs-ID [59](#)
- sekundäre Berechtigungs-ID [59](#)
- Sicherheitsmechanismen [51](#)

Db2-Thread, Serialisierung [95](#)

DB2CONN [11](#)

DB2CONN-, DB2ENTRY- und BIND-Optionen koordinieren [24](#)

DB2CONN-Nachrichtenparameter [154](#)

DB2CONN, DB2ENTRY, DB2TRAN für RDO definieren [11](#)

DB2ENTRY [11](#)

DB2SQLJPLANNAME [82](#)

DB2TRAN [11](#)

DCLGEN-Operationen [116](#)

Deadlock-Erkennung [166](#)

Deadlock-Typen [166](#)

Deadlocks [165](#)

Deadlocks bearbeiten [165](#)

Designkriterien [71](#)

DFH0STAT-Bericht

- Bericht der Statistikzusammenfassung [125](#)

DFHD2PXT, Beispiexitprogramm [81](#)

DISCONNECT, Befehl der Anschlussfunktion [36](#)

DISPLAY STATISTICS, Ausgabe [39](#)

DISPLAY, Befehl der Anschlussfunktion [37](#)

DSN3SATH, Beispiel [60](#)

DSN3SSGN, Beispiel [65](#)

DSNC-Transaktionen

- DISCONNECT [33](#), [36](#)

- DISPLAY [33](#), [37](#)

- MODIFY [33](#), [41](#)

- STOP [33](#), [43](#)

- STRT [33](#), [45](#)

DSNCUEXT, Beispiexitprogramm [81](#)

DSNJCC [82](#)

DSNTIAR [112](#)

E

EDF [163](#)

EDF-Anzeige für SQL-Anweisungen [164](#)

Einstiegsthreads [3](#)

Einzelner Adressraum (SASS) [59](#)

Ersatzsicherheit [55](#)

EXEC CICS INQUIRE- und SET-Befehle [30](#)

EXEC SQL COMMIT [95](#)

Exitroutine für die Anmeldung [65](#)

Exitroutine für Verbindung [60](#)

EXPLAIN [117](#)

EXTRACT EXIT, Programm [99](#)

F

Fehlerbestimmung

- CSUB-Trace [160](#)

- Nachrichten [154](#)

- Speicherauszug [162](#)

Fehlerbestimmung (*Forts.*)

Trace [154](#)

Wartezustandstypen [151](#)

Festschreibung, zweiphasig [41](#)

G

Gerüstcursortabelle (SKCT – Skeleton Cursor Table) [16](#)

Gerüstpakettabelle (SKPT – Skeleton Package Table) [16](#)

Geschützte Threads [19](#), [23](#)

GETPAGE [133](#)

Globaler Trace, Datensätze [123](#)

GRANT, Befehl [67](#), [117](#)

GTF (Allgemeine Tracefunktion) [33](#), [122](#), [128](#)

H

Häufigkeit von Aktualisierungen [96](#)

Hilfstracefunktion [127](#)

Hold-Cursor [17](#), [98](#)

I

In Warteschlange stellen und aus Warteschlange entfernen [95](#)

Indexspalten aktualisieren [95](#)

Indizes, eindeutig [95](#)

IRLM, Komponente [165](#)

Isolationsstufe [114](#)

J

JDBC

CICS-Abbrüche [108](#)

Festschreibung, automatisch [107](#)

Synchronisationspunkt [108](#)

JDBC-Profil [82](#)

JDBC-Unterstützung [82](#)

K

Konvertierungsprojekte [83](#)

Korrelations-ID [39](#), [163](#)

L

L8-Tasksteuerblock, offen [90](#)

Leistung

CICS-Db2-Anschlussfunktion [124](#)

CICS-Transaktionen [127](#)

überwachen [123](#)

Lesevorgang, wiederholbar [114](#)

LOCK TABLE, Anweisung [89](#)

LOCKSIZE [89](#)

M

MAXOPENTCBS, Systeminitialisierungsparameter [15](#)

MODIFY TRACE, Befehl [122](#)

MODIFY, Befehl der Anschlussfunktion [41](#)

Multithread-Verbindungen [3](#)

N

Nachrichten in der Fehlerbestimmung [154](#)

NUMLKTS [89](#)

O

Operationen mit der CICS-Db2-Anschlussfunktion

CICS-Db2-Anschlussfunktion starten [27](#)

CICS-Db2-Anschlussfunktion stoppen [27](#)

Optimieren

CICS-Anwendungen [118](#)

P

Pakete

Anwendungsdesign [74](#)

Übersicht [8](#)

vorhandene Anwendungen konvertieren [76](#)

Vorteile gegenüber dynamischem Planwechsel [74](#)

Pakettabelle (PT) [16](#)

Planauswahl, dynamisch

Voraussetzung für Pool-Thread [84](#)

Pläne

Übersicht [8](#)

Planexits, dynamisch

Hinweise zur Verwendung von JDBC oder SQLJ [82](#)

Übersicht [8](#)

Planwechsel, dynamisch [77](#)

Pool-Threads [4](#), [21](#)

Produktionsverfahren [115](#)

Programmablauf, tabellengesteuert [85](#)

PROTECTNUM [21](#), [23](#)

Prozessorbelegung [141](#)

Prozessorzeit

für Db2 Version 6 oder höher berechnen [148](#)

Klasse 1 [141](#)

Klasse 2 [147](#)

Verbrauch [141](#)

Prozessorzeit abrechnen, CLASS 1 und CLASS 2 [146](#)

R

RACF

externer Sicherheitsmanager [49](#)

RACF-Gruppenliste, Option [65](#)

RACF-Klasse

DSNR [60](#)

RACF-Standardressourcenprofile

VCICSCMD, allgemeine Ressourcenklasse [54](#)

RELEASE(COMMIT) [17](#)

Resynchronisationsinformationen [30](#)

RESYNCMEMBER [29](#)

RETAIN, Option [114](#)

RETURN IMMEDIATE [98](#)

RMI (Ressourcenmanagerschnittstelle) [1](#)

S

SASS (einzelner Adressraum) [59](#)

Schreibabsichten [133](#)

Seitenkonflikt [96](#)

Serialisierung [95](#)

Sicherheit

- AUTHTYPE [55](#)
- Befehlssicherheit [51](#)
- DB2TRAN-Ressourcensicherheit [53](#)
- Ersatzbenutzerprüfung [55](#)
- RACF [49](#)
- RACF-Klasse, DSNR [60](#)
- RACF-Profil definieren [52](#)
- Ressourcensicherheit [51](#)
- SASS [59](#)

SMF (Systemverwaltungsfunktion) [32](#), [128](#)

SMF-101-Datensatz

Felder [146](#)

Sonderregister [17](#)

Sperrenskalation [89](#)

Sperrmechanismus, Db2 [89](#)

Sperrstrategie [89](#)

SQL

- dynamisches [68](#)
- qualifiziertes oder nicht qualifiziertes [94](#)
- statisches [68](#)

SQL-Anforderung [1](#)

SQL-Rückgabecode

- 501 [98](#)
- 818 [114](#)
- 911 [165](#)
- 913 [165](#)

SQL-Verarbeitung, Hauptaktivitäten [16](#)

SQL, Sprache [93](#)

SQLCA, Formatierungsroutine [112](#)

SQLJ

CICS-Abbrüche [108](#)

Synchronisationspunkt [108](#)

SQLJ-Unterstützung für Java-Anwendungen für CICS [82](#)

STANDBYMODE, Befehl [100](#)

Statistikdaten [121](#)

Statistiküberwachung in der Leistung [128](#)

STOP, Befehl der Anschlussfunktion [43](#)

STRT, Befehl der Anschlussfunktion [45](#)

SYNCONRETURN [108](#)

Systemdefinitionsparameter

PROTECTNUM [22](#)

THREADLIMIT [22](#)

THREADWAIT [22](#)

Systeminitialisierungsparameter

PROTECTNUM [19](#)

THREADLIMIT [19](#)

THREADWAIT [18](#)

Systemprogrammierungsfunktion mithilfe von CEMT [33](#)

T

Taskbezogener Benutzerexit (TRUE) [2](#)

Tasksteuerblöcke, offen

Abrechnung [148](#)

als Thread-Tasksteuerblöcke [4](#), [151](#)

Anwendungsprogramme in [90](#)

TCBLIMIT [15](#)

THRDA [84](#)

Thread-Tasksteuerblöcke

CSUB, CICS-Verbindungssteuerblock [4](#)

Db2-Verbindungssteuerblock [4](#)

DFHD2CO, Modul [151](#)

DFHD2D2, Modul [151](#)

Thread-Tasksteuerblöcke (*Forts.*)

in der Umgebung für offene Transaktionen [4](#), [151](#)

Threaderstellung [16](#)

Threadfreigabe [17](#)

Threadidentifikation, Db2 [163](#)

THREADLIMIT [21](#), [84](#)

Threads

Auswirkung auf die Leistung [22](#)

erstellen, verwenden und beenden [18](#)

freigeben [18](#)

ungeschützte, für Hintergrundtransaktionen [21](#)

ungeschützte, für kritische Transaktionen [20](#)

Threads wiederverwenden

Sicherheit [23](#)

Threadtypen

Befehlsthreads [3](#)

Einstiegsthreads [3](#)

Pool-Threads [4](#)

THREADWAIT [21](#)

Transaktionen, von CICS bereitgestellt

DSNC-Transaktionen [33](#)

Systemprogrammierung mithilfe von CEMT [33](#)

Transaktionsabbruchcodes [163](#)

Transaktionsdefinitionen zum Absetzen von Db2-Befehlen [35](#)

TYPE=ENTRY, Transaktionen gruppieren [23](#)

U

Überwachen der Anschlussfunktion

CICS-Transaktionen [124](#)

Überwachungsdaten [121](#)

Umgebung für offene Transaktionen (OTE)

CICS-Db2-Benutzerexit, taskbezogen [90](#)

MAXOPENTCBS, Systeminitialisierungsparameter, Einstellung [15](#)

Prozessorzeit für Transaktionen [148](#)

TCBLIMIT, Einstellung [15](#)

Thread-Tasksteuerblöcke [4](#), [151](#)

threadsichere Anwendungen [90](#)

und Anwendungsprogramme [90](#)

UOW (Arbeitseinheit) [28](#)

V

VALIDATE [115](#)

Validierung [115](#)

VCICSCMD, allgemeine Ressourcenklasse [54](#)

Verarbeitung der Festschreibung [17](#)

Verbindung zwischen CICS und Db2 trennen [28](#)

Verbindungsberechtigung [59](#)

VERSION, Schlüsselwort [117](#)

W

Wartezustandstypen [151](#)

X

XCICSDb2, allgemeine Ressourcenklasse [52](#)

XCICSDb2, Memberklasse [52](#)

XCTL, CICS-Übergabe der Steuerung [74](#)

Z

ZCICSDB2, Gruppierungsklasse [52](#)

Zeitmarken [114](#)

Zuordnung fortlaufender Zahlen [96](#)

