

CICS Transaction Server for z/OS



# Benutzerexitreferenz

*Version 5 Release 5*



CICS Transaction Server for z/OS



# Benutzerexitreferenz

*Version 5 Release 5*

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 311 gelesen werden.

# Inhaltsverzeichnis

## Informationen zu vorliegender PDF-

Datei . . . . . vii

## Globale Benutzerexitpunkte (nach Funktion) . . . . . 1

Globale Benutzerexitpunkte (in alphabetischer Reihenfolge) . . . . .	1
Programmexit für Aktivitätsschlüsselpunkte (XAKUSER). . . . .	10
Exit für Anwendungsbezugsdaten in der AP-Domäne (XAPADMGR) . . . . .	11
Basic Mapping Support-Exits (XBMIN, XBMOUT) . . . . .	12
Exit XBMIN . . . . .	13
Exit XBMOUT . . . . .	14
Struktur der Feldelementtabelle. . . . .	14
Programmierung des Exits XBMIN . . . . .	16
Programmierung des Exits XBMOUT . . . . .	16
Exit XFAINTU für Brückeneinrichtung . . . . .	17
Exits XDTRD, XDTAD und XDTLC zur Datentabellenverwaltung . . . . .	19
Exit XDTRD . . . . .	20
Exit XDTAD . . . . .	22
Exit XDTLC . . . . .	24
Exit XXDFA für das DBCTL-Schnittstellensteuerprogramm . . . . .	25
Exits für DBCTL-Überwachungsprogramm (XXDFB, XXDTO) . . . . .	26
Exit XXDFB . . . . .	26
Exit XXDTO . . . . .	27
Exits XDSBWT und XDSAWT für die Dispatcherdomäne . . . . .	27
Exit XDSBWT . . . . .	28
Exit XDSAWT . . . . .	28
Exits XDLIPRE und XDLIPOST des DL/I-Schnittstellenprogramms . . . . .	29
Exit XDLIPRE . . . . .	30
Exit XDLIPOST . . . . .	32
Beispiel für Verwendung des globalen Benutzerexits XDLIPRE. . . . .	34
Exits XDUREQ, XDUREQC, XDUCLE und XDOUT für die Speicherauszugsdomäne . . . . .	40
Exit XDUREQ . . . . .	40
Exit XDUREQC . . . . .	44
Exit XDUCLE . . . . .	47
Exit XDOUT . . . . .	47
Exits XNQEREQ und XNQEREC des Einreihungsprogramms der EXEC-Schnittstelle . . . . .	48
Exit XNQEREQ . . . . .	49
Exit XNQEREC . . . . .	51
Parameterstruktur der Befehlsebene . . . . .	52
Exit XEPCAP für Ereigniserfassung . . . . .	56
Exits XEIIIN, XEIOU, XEISPIN und XEISPOUT für das EXEC-Schnittstellenprogramm. . . . .	57
Die Befehlsparameterliste. . . . .	58
Umgehen von Befehlen . . . . .	58

Exit XEIIIN. . . . .	59
Exit XEISPIN . . . . .	59
Exit XEIOU . . . . .	60
Exit XEISPOUT . . . . .	61
Exits XSZARQ und XSZBRQ für die Front-End-Programmierschnittstelle (FEPI). . . . .	62
XSZBRQ . . . . .	63
XSZARQ . . . . .	64
Exitspezifische Parameter UEPSZACT und UEPSZACN . . . . .	65
Verwendung von XMEOU zur Steuerung der Nachrichtenausgabe . . . . .	66
Exits XFCFRIN und XFCFROUT für die Dateisteuerungsdomäne. . . . .	66
Exit XFCFRIN . . . . .	68
Exit XFCFROUT . . . . .	75
Exits XFCREQ und XFCREQC für die Dateisteuerungs-API der EXEC-Schnittstelle . . . . .	81
Parameterstruktur der Befehlsebene . . . . .	82
Ändern von Feldern in der Parameterstruktur der Befehlsebene . . . . .	87
Ändern des EXEC-Schnittstellendeskriptors (EID) . . . . .	88
Verwendung des Parameters UEPFSHIP . . . . .	90
EIB (EXEC-Schnittstellenblock) . . . . .	90
Beispiel für die Verwendung von XFCREQ und XFCREQC. . . . .	91
Exit XFCREQ. . . . .	91
Exit XFCREQC . . . . .	93
Exits XFCAREQ und XFCAREQC für die Dateisteuerungs-SPI der EXEC-Schnittstelle . . . . .	94
Exit XFCAREQ . . . . .	95
Exit XFCAREQC. . . . .	96
Parameterstruktur der Befehlsebene . . . . .	98
Ändern von Feldern in der Parameterstruktur der Befehlsebene . . . . .	106
Ändern des EXEC-Schnittstellendeskriptors (EID) . . . . .	109
Ändern von Benutzerargumenten. . . . .	110
Exits XFCSREQ und XFCSREQC für das Dateisteuerprogramm für Dateistatus . . . . .	111
Exit XFCSREQ . . . . .	112
Exit XFCSREQC . . . . .	116
Exit XFCNREC für das Dateisteuerprogramm für Öffnen/Schließen . . . . .	121
Exit XFCNREC bei abweichenden Einstellungen für Rückwärtswiederherstellung . . . . .	122
Verwendung von XFCNREC bei einer BWO-Abweichung. . . . .	123
Exit XFCVSDS zum Empfangen von Quiesce-Anforderungen zur Dateisteuerung . . . . .	123
Exit XFCVSDS . . . . .	124
Exit XFCQUIS zum Senden von Quiesce-Anforderungen zur Dateisteuerung . . . . .	125
Exits XFCBFAIL, XFCBOUT, XFCBOVER und XFCDEL für das Dateisteuerprogramm für Wiederherstellung . . . . .	127

Reihenfolge des Aufrufs . . . . .	128	Exit XWSSRROO . . . . .	203
Aktivierung der Exitprogramme . . . . .	128	Exit XWSSRROI . . . . .	204
Exit XFCBFAIL beim Fehlschlagen eines Dateisteuerungsbackouts . . . . .	129	Exit XWSSRRWI . . . . .	205
Exit XFCBOUT für Dateisteuerungsbackout . . . . .	133	Exits für das Programmsteuerprogramm (XPCREQ, XPCERES, XPCREQC, XPCFTCH, XPCHAIR, XPC-TA und XPCABND) . . . . .	206
Exit XFCBOVER bei Überschreibung eines Dateisteuerungsbackouts . . . . .	134	Programmsteuerungsexits XPCREQ, XPCERES und XPCREQC . . . . .	206
Exit XFCLDEL zur Dateisteuerung beim logischen Löschen . . . . .	136	Exit XPCFTCH . . . . .	218
Exit XFCRLSCO für das Dateisteuerprogramm für RLS-Koexistenz. . . . .	138	Exit XPCHAIR . . . . .	221
Exit für das Programm für Guten-Morgen-Nachricht (XGMTEXT) . . . . .	140	Exit XPCTA . . . . .	223
HTTP-Client-Exits für Öffnen und Senden: XWBAUTH, XWBOPEN und XWBSNDO . . . . .	140	Exit XPCABND . . . . .	225
Sendeexit XWBAUTH für HTTP-Client. . . . .	140	Exits für das Ressourcenmanagerschnittstellenprogramm (XRMIIN, XRMIOU) . . . . .	226
Typische Verwendung der LDAP-XPI-Funktionen durch XWBAUTH . . . . .	143	Exit XRMIIN . . . . .	226
Exit XWBOPEN für Öffnen für HTTP-Client . . . . .	144	Exit XRMIOU . . . . .	228
Sendeexit XWBSNDO für HTTP-Client . . . . .	146	Exit XRSINDI für das Ressourcenmanagement beim Installieren und Verwerfen . . . . .	229
Exits XISCONA, XISLCLQ und XISQLCL für das Programm für systemübergreifende Kommunikation . . . . .	148	Exit XRSINDI . . . . .	230
Exit XISCONA . . . . .	148	Exits XSNON, XSNOFF und XSNEQ für Anmeldungen und Abmeldungen . . . . .	237
Der Exit XISLCLQ. . . . .	152	Exit XSNON. . . . .	237
Der Exit XISQLCL. . . . .	153	Exit XSNOFF . . . . .	238
Exits XICREQ, XICEXP und XICTENF für das Intervallsteuerprogramm . . . . .	155	Exit XSNEQ . . . . .	239
Exit XICREQ . . . . .	155	Exit XSTOUT für die Statistikdomäne . . . . .	240
Exit XICEXP. . . . .	157	Exit XSTOUT . . . . .	240
Exit XICTENF . . . . .	157	Exit XSRAB für das Systemwiederherstellungsprogramm . . . . .	242
Exits des Intervallsteuerprogramms der EXEC-Schnittstelle (XICEREQ, XICERES und XICEREQC). . . . .	157	Exit XSTERM für das Systemabschlussprogramm . . . . .	246
Parameter, die an die einzelnen Exits übergeben werden . . . . .	159	Exits für die temporäre Speicherdomäne (XTSQRIN, XTSQROUT, XTSPTIN, XTSPTOUT) . . . . .	246
Exit XICEREQ . . . . .	159	Exit XTSQRIN . . . . .	247
Exit XICERES . . . . .	161	Exit XTSQROUT . . . . .	248
Exit XICEREQC . . . . .	162	Exit XTSPTIN . . . . .	250
Parameterstruktur der Befehlsebene . . . . .	164	Exit XTSPTOUT . . . . .	251
Exits XLDLOAD und XLDELETE für die Ladedomäne . . . . .	178	Exits XTSEREQ und XTSEREQC für das EXEC-Schnittstellenprogramm für temporären Speicher . . . . .	252
Exit XLDLOAD. . . . .	178	Exit XTSEREQ . . . . .	253
Exit XLDELETE . . . . .	179	Exit XTSEREQC . . . . .	254
Exit XLGSTRM für die Protokollmanagerdomäne . . . . .	180	Parameterstruktur der Befehlsebene . . . . .	256
Exit XLGSTRM. . . . .	182	Exit XALCAID für das Terminalzuordnungsprogramm . . . . .	263
Beispiel für die Verwendung des Exits XLGSTRM . . . . .	184	Exits für das Terminalsteuerprogramm (XTCIN, XTCOUT, XTCATT) . . . . .	264
Exit XMEOUT für die Nachrichtendomäne . . . . .	184	Exit XTCIN . . . . .	264
Exit XMEOUT . . . . .	187	Exit XTCOUT . . . . .	265
Exit für die Überwachungsdomäne (XMNOUT) . . . . .	189	Exit XTCATT . . . . .	265
Exit XMNOUT . . . . .	189	Exits XALTENF und XICTENF für die Bedingung "Terminal unbekannt" . . . . .	266
Exits für die Pipelinedomäne . . . . .	191	Die Exits . . . . .	267
Exit XWSPRRWI . . . . .	193	Exit XALTENF . . . . .	268
Exit XWSPRROI . . . . .	194	Exit XICTENF . . . . .	270
Exit XWSPRROO . . . . .	195	Exit XXMATT für die Transaktionsmanagerdomäne . . . . .	273
Exit XWSPRRWO . . . . .	196	Exits für das Programm für transiente Daten (XTDREQ, XTDIN, XTDOUT) . . . . .	275
Exit XWSRQRWO . . . . .	198	Exit XTDREQ . . . . .	275
Exit XWSRQROO . . . . .	199	Exit XTDIN . . . . .	276
Exit XWSRQROI . . . . .	200	Exit XTDOUT . . . . .	277
Exit XWSRQRWI . . . . .	201	Exits XTDEREQ und XTDEREQC des EXEC-Schnittstellenprogramms für transiente Daten . . . . .	278
Exit XWSSRRWO . . . . .	202	Exit XTDEREQ . . . . .	279

Exit XTDEREQC . . . . .	280
Parameterstruktur der Befehlsebene . . . . .	281
Exits XRCINIT und XRCINPT des Wiederherstellungsprogramms für Benutzerprotokolldatensätze . . . . .	287
Codierung der Exitprogramme . . . . .	288
Aktivierung der Exitprogramme . . . . .	289
Exit XRCINIT . . . . .	289
Exit XRCINPT . . . . .	290
SNA-LU-Verwaltungsprogrammexit (XZCATT) . . . . .	291
Exits für SNA-Arbeitsbereichsmodule (XZCIN, XZCOUT, XZCOUT1 und XZIQUE) . . . . .	292
Exit XZCIN . . . . .	292
Exit XZCOUT . . . . .	292
Exit XZCOUT1 . . . . .	293
Exit XZIQUE zum Verwalten systemübergreifender MRO- und APPC-Warteschlangen . . . . .	294
Entwerfen eines globalen Benutzerexitprogramms für XZIQUE . . . . .	301

Exit XISQUE zum Verwalten systemübergreifender IPIC-Warteschlangen . . . . .	302
Exit XISQUE . . . . .	303
Verwendung eines globalen Benutzerexitprogramms für XISQUE . . . . .	305
Statistikfelder in DFHISRDS . . . . .	307
Entwerfen eines globalen Benutzerexitprogramms für XISQUE . . . . .	307
Exit XXRSTAT für das Programm zur Verarbeitung von XRF-Anforderungen . . . . .	308
Exit XXRSTAT . . . . .	309

<b>Bemerkungen . . . . .</b>	<b>311</b>
------------------------------	------------

<b>Index . . . . .</b>	<b>317</b>
------------------------	------------





---

## Informationen zu vorliegender PDF-Datei

Die vorliegende PDF-Datei ist eine Referenz zu den globalen Benutzerexitpunkten, die bereitgestellt werden, damit CICS die Steuerung an ein von Ihnen geschriebenes globales Benutzerexitprogramm übergeben kann. Informationen dazu, wie diese globalen Benutzerexitpunkte in Programmen verwenden werden, finden Sie in der PDF-Datei mit dem Namen *Developing CICS System Programs*. Vor CICS TS V5.4 befanden sich die Informationen aus der vorliegenden PDF-Datei in der Veröffentlichung *Customization Guide*.

Ausführliche Informationen zu den in vorliegendem Handbuch verwendeten Begriffen und Schreibweisen finden Sie unter *Conventions and terminology used in the CICS documentation* im IBM Knowledge Center.

### Datum der vorliegenden PDF-Datei

Die vorliegende PDF-Datei wurde am 14. Dezember 2018 erstellt.



---

## Globale Benutzerexitpunkte (nach Funktion)

Die Exitpunkte sind nach ihren Funktionsbeziehungen gruppiert.

Die Gruppierung basiert im Allgemeinen auf dem Modul bzw. der Domäne von CICS, in dem/der die Exitpunkte vorkommen. Wenn Exitpunkte jedoch in unterschiedlichen Modulen vorkommen, aber eine ähnliche Funktion haben, werden sie unter einem generischen Namen gruppiert. Die Exitgruppen werden nach Modulnamen bzw. generischen Namen in alphabetischer Reihenfolge dargestellt.

Für jeden globalen Benutzerausgangspunkt werden die folgenden Informationen bereitgestellt:

- Exit-ID.
- Exitposition.
- DFHUEPAR-Parameter (sofern vorhanden), die für den Exit eindeutig sind.
- Gültige Rückgabecodes.
- XPI-Aufrufe, die aufgerufen werden können.

---

## Globale Benutzerexitpunkte (in alphabetischer Reihenfolge)

Für jeden Exit wird in dieser Tabelle Folgendes angezeigt: der Exitname, das Modul bzw. die Domäne, Informationen dazu, wo bzw. wann der Exit aufgerufen wird, sowie ein Link zu weiteren Informationen.

*Tabelle 1. Alphabetische Liste der globalen Benutzerexitpunkte*

Exitname	Modul bzw. Domäne	Wo bzw. wann aufgerufen	Thema
XAKUSER	Programm für Aktivitätsschlüsselpunkte	Unmittelbar vor dem Schreiben des Eintrags für das Ende eines Schlüsselpunkts (end of keypoint)	„Programmexit für Aktivitätsschlüsselpunkte (XAKUSER)“ auf Seite 10
XALCAID	Terminalzuordnungsprogramm	Sobald ein automatischer Initialisierungsdeskriptor (AID) mit Daten abgebrochen wird.	„Exit XALCAID für das Terminalzuordnungsprogramm“ auf Seite 263
XALTENF	Terminalzuordnungsprogramm	Wenn für eine ATI-Anforderung von transienten Daten oder der Intervallsteuerung ein Terminal erforderlich ist, das in diesem System unbekannt ist.	„Exit XALTENF“ auf Seite 268
XAPADMGR	Anwendungsdomäne	Wenn eine Nicht-Systemtask angehängt wird, für die keine übernommenen Daten des zugeordneten Datenursprungsdeskriptors vorliegen.	„Exit für Anwendungsbezugsdaten in der AP-Domäne (XAPADMGR)“ auf Seite 11
XBMIN	Basic Mapping Support	Wenn eine Eingabezuordnungsoperation erfolgreich abgeschlossen wird.	„Exit XBMIN“ auf Seite 13
XBMOUT	Basic Mapping Support	Wenn eine Ausgabeseite erfolgreich erstellt worden ist.	„Exit XBMOUT“ auf Seite 14

Tabelle 1. Alphabetische Liste der globalen Benutzerexitpunkte (Forts.)

Exitname	Modul bzw. Domäne	Wo bzw. wann aufgerufen	Thema
XDLIPST	DL/I-Schnittstellenprogramm	Beim Verlassen des DL/I-Schnittstellenprogramms.	„Exit XDLIPST“ auf Seite 32
XDLIPRE	DL/I-Schnittstellenprogramm	Beim Einstieg in das DL/I-Schnittstellenprogramm.	„Exit XDLIPRE“ auf Seite 30
XDSAWT	Dispatcherdomäne	Nach einer WAIT-Anforderung für das Betriebssystem.	„Exit XDSAWT“ auf Seite 28
XDSBWT	Dispatcherdomäne	Vor einer WAIT-Anforderung für das Betriebssystem.	„Exit XDSBWT“ auf Seite 28
XDTAD	Datentabellenverwaltung	Wenn eine Schreibanforderung an eine Datentabelle ausgegeben wird.	„Exit XDTAD“ auf Seite 22
XDTLC	Datentabellenverwaltung	Bei Abschluss eines Ladevorgangs für eine Datentabelle.	„Exit XDTLC“ auf Seite 24
XDTRD	Datentabellenverwaltung	Während des Ladens einer Datentabelle, sobald ein Datensatz aus dem Quelldataset abgerufen wird.	„Exit XDTRD“ auf Seite 20
XDUCLSE	Speicherauszugsdomäne	Nachdem die Domäne eine Transaktionsspeicherauszugsdatei geschlossen hat.	„Exit XDUCLSE“ auf Seite 47
XDUOUT	Speicherauszugsdomäne	Bevor die Domäne einen Datensatz in die Transaktionsspeicherauszugsdatei schreibt.	„Exit XDUOUT“ auf Seite 47
XDUREQ	Speicherauszugsdomäne	Bevor die Domäne einen System- oder Transaktionsspeicherauszug erstellt.	„Exit XDUREQ“ auf Seite 40
XDUREQC	Speicherauszugsdomäne	Nachdem ein System- oder Transaktionsspeicherauszug erstellt wurde (oder fehlgeschlagen ist oder unterdrückt wurde).	„Exit XDUREQC“ auf Seite 44
XEIIN	EXEC-Schnittstellenprogramm	Bevor ein API- oder SPI-Befehl von EXEC CICS ausgeführt wird.	„Exit XEIIN“ auf Seite 59
XEIOUT	EXEC-Schnittstellenprogramm	Nachdem ein API- oder SPI-Befehl von EXEC CICS ausgeführt worden ist.	„Exit XEIOUT“ auf Seite 60
XEISPIN	EXEC-Schnittstellenprogramm	Bevor ein SPI-Befehl von EXEC CICS ausgeführt wird. <i>Ausnahmen:</i> EXEC CICS ENABLE, EXEC CICS DISABLE, EXEC CICS EXTRACT EXIT, EXEC CICS PERFORM DUMP und EXEC CICS RESYNC ENTRYNAME.	„Exit XEISPIN“ auf Seite 59
XEISPOUT	EXEC-Schnittstellenprogramm	Nachdem ein SPI-Befehl von EXEC CICS ausgeführt worden ist. <i>Ausnahmen:</i> <b>EXEC CICS ENABLE, EXEC CICS DISABLE, EXEC CICS EXTRACT EXIT, EXEC CICS PERFORM DUMP und EXEC CICS RESYNC ENTRYNAME.</b>	„Exit XEISPOUT“ auf Seite 61
XEPCAP	Ereigniserfassung	Bevor ein Ereignis von der CICS-Ereignisverarbeitung erfasst wird.	„Exit XEPCAP für Ereigniserfassung“ auf Seite 56
XFAINTU	Verwaltungsprogramm für 3270-Brückeneinrichtungen	Wenn eine Brückeneinrichtung erstellt oder gelöscht wird.	„Exit XFAINTU für Brückeneinrichtung“ auf Seite 17

Tabelle 1. Alphabetische Liste der globalen Benutzerexitpunkte (Forts.)

Exitname	Modul bzw. Domäne	Wo bzw. wann aufgerufen	Thema
XFCAREQ	Dateisteuerprogramm für die EXEC-Schnittstelle	Bevor CICS eine SPI-Anforderung für die Dateisteuerung verarbeitet.	„Exits XFCAREQ und XFCAREQC für die Dateisteuerungs-SPI der EXEC-Schnittstelle“ auf Seite 94
XFCAREQC	Dateisteuerprogramm für die EXEC-Schnittstelle	Nach Abschluss einer SPI-Anforderung für die Dateisteuerung.	„Exits XFCAREQ und XFCAREQC für die Dateisteuerungs-SPI der EXEC-Schnittstelle“ auf Seite 94
XFCBFAIL	Wiederherstellungssteuerprogramm für die Dateisteuerung	Wenn während des Backouts einer Arbeitseinheit (UOW) ein Fehler auftritt.	„Exit XFCBFAIL beim Fehlschlagen eines Dateisteuerungsbackouts“ auf Seite 129
XFCBOUT	Wiederherstellungssteuerprogramm für die Dateisteuerung	Wenn CICS im Begriff ist, eine Dateiaktualisierung zurückzusetzen.	„Exit XFCBOUT für Dateisteuerungsbackout“ auf Seite 133
XFCBOVER	Wiederherstellungssteuerprogramm für die Dateisteuerung	Wenn CICS im Begriff ist, das Backout einer Arbeitseinheit (UOW) zu überspringen, weil ein Stapelverarbeitungsprogramm den Schutz für beibehaltene RLS-Sperren außer Kraft gesetzt und ein Dataset für die Stapelverarbeitung geöffnet hat.	„Exit XFCBOVER bei Überschreitung eines Dateisteuerungsbackouts“ auf Seite 134
XFCFRIN	Dateisteuerungsdomäne	Vor Ausführung einer Dateisteueranforderung.	„Exit XFCFRIN“ auf Seite 68
XFCFROUT	Dateisteuerungsdomäne	Nach Ausführung einer Dateisteueranforderung.	„Exit XFCFROUT“ auf Seite 75
XFCLDEL	Wiederherstellungssteuerprogramm für die Dateisteuerung	Beim Zurücksetzen von Schreibvorgängen für ein VSAM-ESDS oder ein BDAM-Dataset.	„Exit XFCLDEL zur Dateisteuerung beim logischen Löschen“ auf Seite 136
XFCNREC	Dateisteuerprogramm für Öffnen/Schließen	Wenn während der Verarbeitung einer Anforderung zum Öffnen einer Datei eine Abweichung zwischen der Wiederherstellungseinstellung für Backouts der Datei und des zugehörigen Datasets festgestellt wird.	„Exit XFCNREC für das Dateisteuerprogramm für Öffnen/Schließen“ auf Seite 121
XFCQUIS	Dateisteuerprogramm für das Senden von Quiesce-Anforderungen	Bei Abschluss (erfolgreich oder fehlgeschlagen) eines Befehls vom Typ SET DSNAME QUIESCESTATE.	„Exit XFCQUIS zum Senden von Quiesce-Anforderungen zur Dateisteuerung“ auf Seite 125
XFCREQ	Dateisteuerprogramm für die EXEC-Schnittstelle	Bevor CICS eine API-Anforderung für die Dateisteuerung verarbeitet.	„Exit XFCREQ“ auf Seite 91

Tabelle 1. Alphabetische Liste der globalen Benutzerexitpunkte (Forts.)

Exitname	Modul bzw. Domäne	Wo bzw. wann aufgerufen	Thema
XFCREQC	Dateisteuerprogramm für die EXEC-Schnittstelle	Nach Abschluss einer API-Anforderung für die Dateisteuerung.	„Exit XFCREQC“ auf Seite 93
XFCRLSCO	Dateisteuerprogramm für RLS-Koexistenz	Wenn das Öffnen einer VSAM-RLS-Datei oder einer schreibgeschützten Nicht-RLS-Datei ansonsten mit einem RLS-Koexistenzfehler fehlschlagen würde.	„Exit XFCRLSCO für das Dateisteuerprogramm für RLS-Koexistenz“ auf Seite 138
XFCSREQ	Dateisteuerprogramm für Dateistatus	Vor dem Versuch, einen Befehl vom Typ OPEN, CLOSE, ENABLE oder DISABLE für eine Datei auszuführen.	„Exits XFCSREQ und XFCREQC für das Dateisteuerprogramm für Dateistatus“ auf Seite 111
XFCSREQC	Dateisteuerprogramm für Dateistatus	Nach Ausführung eines Befehls vom Typ OPEN, CLOSE, CANCEL CLOSE, ENABLE oder DISABLE für eine Datei.	„Exits XFCSREQ und XFCREQC für das Dateisteuerprogramm für Dateistatus“ auf Seite 111
XFCVSDS	Dateisteuerprogramm für das Empfangen von Quiesce-Anforderungen	Nachdem CICS von RLS darüber informiert wurde, dass infolge einer im Sysplex auftretenden datasetbezogenen Aktion eine Verarbeitung erforderlich ist.	„Exit XFCVSDS zum Empfangen von Quiesce-Anforderungen zur Dateisteuerung“ auf Seite 123
XGMTEXT	Programm für Guten-Morgen-Nachricht ("good morning")	Vor dem Senden der Guten-Morgen-Nachricht („good morning“).	„Exit für das Programm für Guten-Morgen-Nachricht (XGMTEXT)“ auf Seite 140
XICEREQ	Intervallsteuerprogramm für die EXEC-Schnittstelle	Bevor CICS eine API-Anforderung für die Intervallsteuerung verarbeitet.	„Exit XICEREQ“ auf Seite 159
XICEREQC	Intervallsteuerprogramm für die EXEC-Schnittstelle	Nach Abschluss einer API-Anforderung für die Intervallsteuerung.	„Exit XICEREQC“ auf Seite 162
XICERES	Intervallsteuerprogramm für die EXEC-Schnittstelle	Bevor CICS eine nicht terminalbezogene Anforderung vom Typ EXEC CICS START verarbeitet, die dynamisch an diese Region weitergeleitet wurde, wobei die Routing-Region die Bedingung RESUNAVAIL (Ressource nicht verfügbar) unterstützt.	„Exit XICERES“ auf Seite 161
XICEXP	Intervallsteuerprogramm	Nach Ablauf eines Zeitintervalls in der Intervallsteuerung.	„Exit XICEXP“ auf Seite 157
XICREQ	Intervallsteuerprogramm	Beim Start des Intervallsteuerprogramms, vor der Anforderungsanalyse.	„Exit XICREQ“ auf Seite 155
XICTENF	Intervallsteuerprogramm	Wenn für einen Befehl vom Typ EXEC CICS START ein Terminal erforderlich ist, das in diesem System unbekannt ist.	„Exit XICTENF“ auf Seite 270

Tabelle 1. Alphabetische Liste der globalen Benutzerexitpunkte (Forts.)

Exitname	Modul bzw. Domäne	Wo bzw. wann aufgerufen	Thema
XISCONA	Programm für systemübergreifende Kommunikation	Wenn eine Funktionsverlagerungs- oder DPL-Anforderung im Begriff ist, in die Warteschlange gestellt zu werden, weil für die ferne Region keine Sitzungen unmittelbar verfügbar sind.	„Exits XISCONA, XISLCLQ und XISQLCL für das Programm für systemübergreifende Kommunikation“ auf Seite 148
XISLCLQ	Programm für systemübergreifende Kommunikation	Nachdem ein Versuch, eine Sitzung für eine über Funktionen übertragene Anforderung vom Typ START NOCHECK zuzuordnen, fehlgeschlagen ist, weil das ferne System nicht in Betrieb ist, keine Verbindung zum fernen System hergestellt werden kann oder keine Sitzungen unmittelbar verfügbar sind, während das verwendete Exitprogramm für XISCONA angegeben hat, dass die Anforderung in der ausgehenden Region nicht in die Warteschlange gestellt werden soll.	„Der Exit XISLCLQ“ auf Seite 152
XISQLCL	Programm für systemübergreifende Kommunikation	Nach dem Fehlschlagen eines Versuchs, eine Sitzung für eine Anforderung vom Typ START NOCHECK zuzuordnen, die für eine IPIC-Verbindung terminiert ist, weil die IPIC-Verbindung außer Betrieb ist, nicht angefordert wurde oder keine Sitzung verfügbar ist, während CICS die Anforderung nicht für eine neue Sitzung in die Warteschlange stellt.	„Der Exit XISQLCL“ auf Seite 153
XISQUE	Programm zur Steuerung der Anzahl von Anforderungen in der Warteschlange für Sitzungen in IPIC-Verbindungen (IPCONNs).	Wenn 1. eine Zuordnungsanforderung für eine Sitzung in einer IPIC-Verbindung (IPCONN) im Begriff ist, in die Warteschlange gestellt zu werden. 2. eine IP-Zuordnungsanforderung nach vorheriger Unterdrückung der Warteschlangensteuerung erfolgreich ausgeführt wird.	„Exit XISQUE zum Verwalten systemübergreifender IPIC-Warteschlangen“ auf Seite 302
XLDELETE	Ladedomäne	Nach der Freigabe einer Instanz eines Programms durch CICS und kurz vor der Freigabe des Programms aus dem Speicher.	„Exit XLDELETE“ auf Seite 179
XLDLOAD	Ladedomäne	Nach der Überführung einer Instanz eines Programms in den Speicher und vor der Bereitstellung des Programms zur Verwendung.	„Exit XLDLOAD“ auf Seite 178
XLGSTRM	Protokollmanagerdomäne	Nachdem der CICS-Protokollmanager feststellt, dass ein Protokolldatenstrom nicht vorhanden ist, und vor dem Aufrufen der MVS-Systemprotokollfunktion zum Definieren des Protokolldatenstroms.	„Exit XLGSTRM für die Protokollmanagerdomäne“ auf Seite 180
XMEOUT	Nachrichtendomäne	Bevor eine Nachricht von der Nachrichtendomäne an ihr Ziel gesendet wird.	„Exit XMEOUT“ auf Seite 187
XMNOUT	Überwachungsdomäne	Bevor ein Datensatz entweder in SMF geschrieben wird oder vor dem Schreiben in SMF gepuffert wird.	„Exit XMNOUT“ auf Seite 189
XNQEREQ	Einreihungsprogramm für die EXEC-Schnittstelle	Bevor CICS eine API-Anforderung für das Einreihen in die Datenbank verarbeitet.	„Exit XNQEREQ“ auf Seite 49

Tabelle 1. Alphabetische Liste der globalen Benutzerexitpunkte (Forts.)

Exitname	Modul bzw. Domäne	Wo bzw. wann aufgerufen	Thema
XNQEREQC	Einreihungsprogramm für die EXEC-Schnittstelle	Nach Abschluss einer API-Anforderung für das Einreihen in die Datenbank.	„Exit XNQEREQC“ auf Seite 51
XPCABND	Programmsteuerprogramm	Nach dem Abbruch einer Transaktion und vor der Ausgabe eines Speicherauszugsaufrufs.	„Exit XPCABND“ auf Seite 225
XPCERES	Programmsteuerprogramm	Bevor CICS eine Programmverknüpfungs- oder Link3270-Brückenanforderung verarbeitet, die dynamisch an diese Region weitergeleitet wurde, wobei die Routing-Region die Bedingung RESUNAVAIL (Ressource nicht verfügbar) unterstützt.	„Exit XPCERES“ auf Seite 209
XPCFTCH	Programmsteuerprogramm	Bevor ein Anwendungsprogramm die Steuerung erhält.	„Exit XPCFTCH“ auf Seite 218
XPCHAIR	Programmsteuerprogramm	Bevor eine Routine vom Typ HANDLE ABEND die Steuerung erhält.	„Exit XPCHAIR“ auf Seite 221
XPCREQ	Programmsteuerprogramm	Vor der Verarbeitung einer Verknüpfungsanforderung (LINK).	„Exit XPCREQ“ auf Seite 208
XPCREQC	Programmsteuerprogramm	Nach Abschluss einer Verknüpfungsanforderung (LINK).	„Exit XPCREQC“ auf Seite 211
XPCTA	Programmsteuerprogramm	Nach dem Auftreten eines Abbruchs und vor dem Ändern der Umgebung.	„Exit XPCTA“ auf Seite 223
XRCINIT	Wiederherstellungsprogramm für Benutzerprotokolldatensätze	Während eines Warmstarts und eines Wiederanlaufs nach Systemabsturz, wenn Benutzerprotokolldatensätze für Wiederherstellung im CICS-Systemprotokoll gefunden werden.	„Exit XRCINIT“ auf Seite 289
XRCINPT	Wiederherstellungsprogramm für Benutzerprotokolldatensätze	Während eines Warmstarts und eines Wiederanlaufs nach Systemabsturz, und zwar für jeden Benutzerprotokolldatensatz für Wiederherstellung, der im CICS-Systemprotokoll gefunden wird.	„Exit XRCINPT“ auf Seite 290
XRMIIN	Ressourcenmanagerschnittstellenprogramm	Vor Ausführung eines Befehls vom Typ EXEC DLI, EXEC SQL oder RMI.	„Exit XRMIIN“ auf Seite 226
XRMIOUT	Ressourcenmanagerschnittstellenprogramm	Nach Ausführung eines Befehls vom Typ EXEC DLI, EXEC SQL oder RMI.	„Exit XRMIIN“ auf Seite 226
XRSINDI	Ressourcenmanagementmodule	Unmittelbar nachdem eine Ressource erfolgreich installiert oder verworfen worden ist.	„Exit XRSINDI für das Ressourcenmanagement beim Installieren und Verwerfen“ auf Seite 229
XSNEX	Sicherheitsmanagerdomäne	Wiederherstellung eines alten CICS-Verhaltens für An-/Abmeldung (vor CICS TS 2.1).	„Exit XSNEX“ auf Seite 239
XSNOFF	Sicherheitsmanagerdomäne	Nachdem sich ein Terminalbenutzer abgemeldet hat.	„Exit XSNOFF“ auf Seite 238
XSNON	Sicherheitsmanagerdomäne	Nachdem sich ein Terminalbenutzer angemeldet hat.	„Exit XSNON“ auf Seite 237



Tabelle 1. Alphabetische Liste der globalen Benutzerexitpunkte (Forts.)

Exitname	Modul bzw. Domäne	Wo bzw. wann aufgerufen	Thema
XSRAB	Systemwiederherstellungsprogramm	Wenn das Systemwiederherstellungsprogramm eine Übereinstimmung für einen MVS-Abbruchcode in der Systemwiederherstellungstabelle (SRT) findet.	„Exit XSRAB für das Systemwiederherstellungsprogramm“ auf Seite 242
XSTERM	Systemabschlussprogramm	Während eines normalen Systemabschlusses, unmittelbar vor dem Löschen des Inhalts von TD-Puffern.	„Exit XSTERM für das Systemabschlussprogramm“ auf Seite 246
XSTOUT	Statistikdomäne	Bevor ein Statistikdatensatz in SMF geschrieben wird.	„Exit XSTOUT“ auf Seite 240
XSZARQ	Front-End-Programmierschnittstelle	Nach Abschluss einer FEPI-Anforderung.	„Exits XSZARQ und XSZBRQ für die Front-End-Programmierschnittstelle (FEPI)“ auf Seite 62
XSZBRQ	Front-End-Programmierschnittstelle	Vor Ausführung einer FEPI-Anforderung.	„Exits XSZARQ und XSZBRQ für die Front-End-Programmierschnittstelle (FEPI)“ auf Seite 62
XTCATT	Terminalsteuerprogramm	Vor dem Anhängen einer Task.	„Exit XTCATT“ auf Seite 265
XTCIN	Terminalsteuerprogramm	Nach einem Eingabeereignis.	„Exit XTCIN“ auf Seite 264
XTCOUT	Terminalsteuerprogramm	Vor einem Ausgabeereignis.	„Exit XTCOUT“ auf Seite 265
XTDEREQ	EXEC-Schnittstellenprogramm für transiente Daten	Bevor CICS eine API-Anforderung für transiente Daten verarbeitet.	„Exit XTDEREQ“ auf Seite 279
XTDEREQC	EXEC-Schnittstellenprogramm für transiente Daten	Nach Abschluss einer API-Anforderung für transiente Daten.	„Exit XTDEREQC“ auf Seite 280
XTDIN	Programm für transiente Daten	Nach dem Empfang von Daten von QSAM (partitionsübergreifend) oder VSAM (partitionsintern).	„Exit XTDIN“ auf Seite 276
XTDOUT	Programm für transiente Daten	Vor der Übergabe von Daten an eine benutzerdefinierte Warteschlange mit transienten Daten unter QSAM (partitionsübergreifend) oder VSAM (partitionsintern).	„Exit XTDOUT“ auf Seite 277
XTDREQ	Programm für transiente Daten	Vor der Anforderungsanalyse.	„Exit XTDREQ“ auf Seite 275
XTSREQ	EXEC-Schnittstellenprogramm für temporären Speicher	Bevor CICS eine API-Anforderung für temporären Speicher verarbeitet.	„Exit XTSREQ“ auf Seite 253
XTSREQC	EXEC-Schnittstellenprogramm für temporären Speicher	Nach Abschluss einer API-Anforderung für temporären Speicher.	„Exit XTSREQC“ auf Seite 254

Tabelle 1. Alphabetische Liste der globalen Benutzerexitpunkte (Forts.)

Exitname	Modul bzw. Domäne	Wo bzw. wann aufgerufen	Thema
XTSPTIN	Temporäre Speicherdomäne	Vor dem Aufruf einer TSPT-Funktion.	„Exit XTSPTIN“ auf Seite 250
XTSPTOUT	Temporäre Speicherdomäne	Nach dem Aufruf einer TSPT-Funktion.	„Exit XTSPTOUT“ auf Seite 251
XTSQRIN	Temporäre Speicherdomäne	Vor dem Aufruf einer TSQR-Funktion.	„Exit XTSQRIN“ auf Seite 247
XTSQROUT	Temporäre Speicherdomäne	Nach dem Aufruf einer TSQR-Funktion.	„Exit XTSQROUT“ auf Seite 248
XWBAUTH	Webdomäne	Während der Verarbeitung eines Befehls vom Typ <b>EXEC CICS WEB SEND</b> oder <b>EXEC CICS WEB CONVERSE</b> .	„HTTP-Client-Exits für Öffnen und Senden: XWBAUTH, XWBOPEN und XWBSNDO“ auf Seite 140
XWBOPEN	Webdomäne	Während der Verarbeitung eines Befehls vom Typ <b>EXEC CICS WEB OPEN</b> oder <b>EXEC CICS INVOKE SERVICE</b> .	„HTTP-Client-Exits für Öffnen und Senden: XWBAUTH, XWBOPEN und XWBSNDO“ auf Seite 140
XWBSNDO	Webdomäne	Während der Verarbeitung eines Befehls vom Typ <b>EXEC CICS WEB SEND</b> oder <b>EXEC CICS WEB CONVERSE</b> .	„HTTP-Client-Exits für Öffnen und Senden: XWBAUTH, XWBOPEN und XWBSNDO“ auf Seite 140
XWSPRROI	Pipelinedomäne	Nachdem eine Instanz des Exits XWSPRRWI aufgerufen wurde und vor der Geschäftsanwendung des Web-Service-Providers.	„Exit XWSPRROI“ auf Seite 194
XWSPRROO	Pipelinedomäne	Nach der Rückkehr der Web-Service-Provider-Anwendung und bevor CICS den Hauptteil der Antwortnachricht erstellt.	„Exit XWSPRROO“ auf Seite 195
XWSPRRWI	Pipelinedomäne	Nachdem CICS den Hauptteil der Web-Service-Anforderung in eine Sprachstruktur konvertiert hat und bevor eine Instanz des Exits XWSPRROI aufgerufen wird.	„Exit XWSPRRWI“ auf Seite 193
XWSPRRWO	Pipelinedomäne	Nach dem Aufruf einer Instanz des Exits XWSPRROO und bevor CICS den Hauptteil der Antwortnachricht erstellt.	„Exit XWSPRRWO“ auf Seite 196
XWSRQROI	Pipelinedomäne	Nachdem CICS die abgehende Web-Service-Antwort verarbeitet hat und bevor eine Instanz des Exits XWSRQRWI aufgerufen wird.	„Exit XWSRQROI“ auf Seite 200
XWSRQROO	Pipelinedomäne	Nachdem eine Instanz des Exits XWSRQRWO verarbeitet wurde und vor den abgehenden Datenflüssen im Web-Service-Transport.	„Exit XWSRQROO“ auf Seite 199
XWSRQRWI	Pipelinedomäne	Nachdem CICS die abgehende Web-Service-Antwort verarbeitet hat und nachdem eine Instanz des Exits XWSRQROI aufgerufen worden ist.	„Exit XWSRQRWI“ auf Seite 201

Tabelle 1. Alphabetische Liste der globalen Benutzerexitpunkte (Forts.)

Exitname	Modul bzw. Domäne	Wo bzw. wann aufgerufen	Thema
XWSRQRWO	Pipelinedomäne	Nachdem CICS die Sprachstruktur der Anwendung in den Hauptteil einer Web-Service-Anforderung konvertiert hat und bevor CICS den optionalen Exitpunkt XWSRQROO verarbeitet.	„Exit XWSRQRWO“ auf Seite 198
XWSSRROI	Pipelinedomäne	Nachdem CICS die abgehende Web-Service-Antwort verarbeitet hat und bevor eine Instanz des Exits XWSSRRWI aufgerufen wird.	„Exit XWSSRROI“ auf Seite 204
XWSSRROO	Pipelinedomäne	Nachdem eine Instanz des Exits XWSSRRWO verarbeitet wurde und vor der Verschlüsselung abgehender Datenflüsse im Web-Service-Transport.	„Exit XWSSRROO“ auf Seite 203
XWSSRRWI	Pipelinedomäne	Nachdem CICS die abgehende Web-Service-Antwort verarbeitet hat und nachdem eine Instanz des Exits XWSSRROI aufgerufen worden ist.	„Exit XWSSRRWI“ auf Seite 205
XWSSRRWO	Pipelinedomäne	Nachdem CICS die Sprachstruktur der Anwendung in den Hauptteil einer Web-Service-Anforderung konvertiert hat und bevor CICS den optionalen Exitpunkt XWSSRRWO verarbeitet und vor der Verschlüsselung durch den Sicherheitshandler der Pipeline.	„Exit XWSSRRWO“ auf Seite 202
XXDFA	Steuerprogramm für DBCTL-Schnittstelle	In der aktiven CICS-Region, wenn die CICS-DBCTL-Verbindung fehlschlägt.	„Exit XXDFA für das DBCTL-Schnittstellensteuerprogramm“ auf Seite 25
XXDFB	DBCTL-Überwachungsprogramm	In der alternativen CICS-Region, wenn die Datenbanksteuerung (DBCTL) fehlschlägt.	„Exit XXDFB“ auf Seite 26
XXDTO	DBCTL-Überwachungsprogramm	In der alternativen CICS-Region, wenn die aktive Datenbanksteuerung (DBCTL) fehlschlägt.	„Exit XXDTO“ auf Seite 27
XXMATT	Transaktionsmanagerdomäne	Wenn eine Benutzertransaktion angehängt wird.	„Exit XXMATT für die Transaktionsmanagerdomäne“ auf Seite 273
XXRSTAT	Verarbeitungsprogramm für XRF-Anforderungen	Nach einem Fehler bei z/OS Communications Server oder einer unfreiwilligen Systemübernahme durch z/OS Communications Server.	„Exit XXRSTAT“ auf Seite 309
XZCATT	Terminalverwaltungsprogramm für z/OS Communications Server	Vor dem Anhängen einer Task.	„SNA-LU-Verwaltungsprogramme-Exit (XZCATT)“ auf Seite 291
XZCIN	Arbeitsbereichsmodul für z/OS Communications Server	Nach einem Eingabeereignis.	„Exit XZCIN“ auf Seite 292
XZCOUT	Arbeitsbereichsmodul für z/OS Communications Server	Vor einem Ausgabeereignis.	„Exit XZCOUT“ auf Seite 292
XZCOUT1	Arbeitsbereichsmodul für z/OS Communications Server	Bevor eine Nachricht in RUs aufgeteilt wird.	„Exit XZCOUT1“ auf Seite 293

Tabelle 1. Alphabetische Liste der globalen Benutzerexitpunkte (Forts.)

Exitname	Modul bzw. Domäne	Wo bzw. wann aufgerufen	Thema
XZIQUE	Arbeitsbereichsmodul für z/OS Communications Server	<ol style="list-style-type: none"> <li>1. Wenn eine Zuordnungsanforderung für eine Sitzung im Begriff ist, in die Warteschlange gestellt zu werden.</li> <li>2. Wenn eine Zuordnungsanforderung nach vorheriger Unterdrückung der Warteschlangensteuerung erfolgreich ausgeführt wird.</li> </ol>	„Exit XZIQUE zum Verwalten systemübergreifender MRO- und APPC-Warteschlangen“ auf Seite 294

## Programmexit für Aktivitätsschlüsselpunkte (XAKUSER)

Der Exit XAKUSER wird während der Verarbeitung von Aktivitätsschlüsselpunkten aufgerufen. Mithilfe dieses Exits können Sie im Systemprotokoll Benutzerdaten aufzeichnen, die nach einem Wiederanlauf nach Systemabsturz wiederhergestellt werden müssen.

Um eine optimale Leistung zu erzielen, sollte in Journalsteueranforderungen nicht WAIT angegeben werden. CICS erzwingt die Aufzeichnung der Datensätze, indem bei der Rückkehr vom Exitprogramm ein synchroner Datensatz für Schlüsselpunktende geschrieben wird.

Das Exitprogramm sollte mit der Option NOEDF umgesetzt werden. Darüber hinaus sollte auch jedes Programm, mit dem das Exitprogramm verknüpft ist, mit dieser Option umgesetzt werden. Es ist nicht möglich, eine Verknüpfung mit Programmen zu erstellen, die in PL/I geschrieben sind.

Um sicherzustellen, dass das Exitprogramm bei der Verarbeitung eines jeden Schlüsselpunkts aufgerufen wird, sollte das Programm mithilfe eines PLTPI-Programms der ersten Phase aktiviert werden (siehe Writing initialization programs). Wird das Programm in dieser Phase aktiviert, sollte es jedoch nicht versuchen, eine Verknüpfung mit Programmen zu erstellen, die in COBOL oder C codiert sind, da es möglicherweise aufgerufen wird, bevor diese Programme initialisiert werden.

**Anmerkung:** Ihr Exitprogramm ist Teil einer kritischen CICS-Systemaktivität. Schlägt das Programm fehl, wird CICS beendet. Nur die aufgeführten EXEC CICS-Befehle sind im Exit XAKUSER zulässig. Der Exit sollte Verknüpfungen ausschließlich mit Programmen erstellen, die denselben Einschränkungen unterliegen.

### Exit XAKUSER

#### Zeitpunkt des Aufrufs

Während der Verarbeitung von Aktivitätsschlüsselpunkten.

#### Exitspezifische Parameter

##### UEPAKTYP

Adresse eines 1-Byte-Felds, das den Schlüsselpunkttyp angibt, für den der Exit aufgerufen wird. Mögliche Werte:

##### UEPAKPER

Aktivitätsschlüsselpunkt.

##### UEPAKWSD

Schlüsselpunkt für Warm-Systemabschluss.

#### Rückgabecodes

## UERCNORM

Verarbeitung fortsetzen.

## XPI-Aufrufe

XPI darf nicht verwendet werden.

## API- und SPI-Aufrufe

Die folgenden Befehle werden unterstützt:

- ADDRESS CWA
- ADDRESS EIB
- LINK (jedoch nur Verknüpfungen mit lokalen Programmen; Verknüpfungen mit verteilten Programmen dürfen nicht verwendet werden.)
- RETURN
- WRITE JOURNALNAME

---

## Exit für Anwendungsbezugsdaten in der AP-Domäne (XAPADMGR)

Verwenden Sie den Exit XAPADMGR für verteilte Transaktionen. Mit XAPADMGR können Sie Benutzerinformationen zu den Bezugsdaten einer Task am Ausgangspunkt (Ursprung) der verteilten Transaktion hinzufügen. Diese Informationen könnten zu einem späteren Zeitpunkt verwendet werden, beispielsweise als Suchschlüssel für die über CICSplex SM ausgeführte Verarbeitung.

Das Exitprogramm (sofern aktiviert) wird aufgerufen, wenn außerhalb des Systems befindliche Tasks angehängt werden, für die kein Ursprungsdeskriptorsatz als Eingabe bereitgestellt wird.

Bei Eingabe werden die Bezugsdaten der Task an das Exitprogramm übergeben. Mithilfe von CICS-Befehlen findet der Exit möglicherweise in anderen Quellen weitere relevante Informationen zwecks Einschluss in die Bezugsdaten.

**Anmerkung:** Verteilte Transaktionen, die DPL über IPIC-Verbindungen verwenden, übergeben ihre Transaktionsgruppen-ID und Ursprungsdaten, einschließlich des Benutzerkorrelators, die von der Spiegeltask in der Zielregion übernommen werden sollen.

Das Exitprogramm könnte weitere Aktivitäten ausführen wie das Protokollieren von Informationen, die in den Bezugsdaten gefunden werden, um beispielsweise Workloads zu prüfen oder abzurechnen. Weitere Informationen zu Bezugsdaten und Ursprungsdaten finden Sie unter Association data.

## Exit XAPADMGR

### Zeitpunkt des Aufrufs

Beim Anhängen einer außerhalb des Systems befindlichen Task, an die keine übernommenen Bezugsdaten übergeben werden.

### Exitspezifische Parameter

#### UEPADCB

Adresse des auswählbaren Bezugsdatensteuerblocks. Diese Zuordnung erfolgt durch den Pseudobereich (DSECT) DFHMNADS.

#### UEPADCBL

Länge des Bezugsdatensteuerblocks in Byte.

### UEPUCD

Adresse eines 64-Byte-Ausgabebereichs, in den das Exitprogramm die Benutzerkorrelationsdaten stellen kann.

Dieser Bereich wird vor dem Aufrufen des ersten Exitprogramms, das im Benutzerexitpunkt XAPADMGR aktiv ist, von CICS auf null gesetzt. Sind hingegen mehrere Exitprogramme aktiv, wird der Bereich zwischen Programmaufrufen nicht von CICS zurückgesetzt. Derselbe Speicherbereich wird an nachfolgende Exitprogramme übergeben, die in demselben Exitpunkt aktiv sind.

### Rückgabecodes

#### UERCNORM

Verarbeitung fortsetzen.

### XPI-Aufrufe

Es können alle verwendet werden.

### API- und SPI-Aufrufe

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

- EXEC CICS ABEND
- EXEC CICS PERFORM SHUTDOWN

### Beispielexitprogramm

DFH\$APAD

---

## Basic Mapping Support-Exits (XBMIN, XBMOUT)

Es werden zwei Basic Mapping Support-Exits bereitgestellt: XBMIN und XBMOUT. Mit dem Exit XBMIN können Sie eine Anforderung vom Typ RECEIVE MAP abfangen, nachdem die Anforderung von Basic Mapping Support (BMS) erfolgreich verarbeitet worden ist. Mit dem Exit XBMOUT können Sie eine Anforderung vom Typ SEND MAP abfangen, nachdem die Anforderung von BMS erfolgreich verarbeitet worden ist oder - wenn kumulatives Mapping in Bearbeitung ist - jeweils bei Fertigstellung der einzelnen Ausgabeseiten.

Der Exit XBMIN (sofern aktiviert) wird aufgerufen, wenn alle folgenden Bedingungen erfüllt sind:

- Der Befehl RECEIVE MAP wurde erfolgreich verarbeitet.
- Die Maske, auf die im Befehl verwiesen wird, enthält mindestens ein Feld, das als VALIDN=USEREXIT angegeben ist.
- Es wurde mindestens ein USEREXIT-Feld im eingehenden Datenstrom zurückgegeben und der Anwendungsdatenstruktur zugeordnet.

Mithilfe von XBMIN können Sie folgende Aktionen ausführen:

- Analysieren aller als VALIDN=USEREXIT definierten Felder, die der Anwendung in dieser Anforderung zugeordnet sind.
- Verwenden des Maskengruppennamens, des Maskennamens und der Feldlänge, die in der Maske definiert sind, sowie der tatsächlichen Länge der Felddaten, die im eingehenden Datenstrom zurückgegeben werden.
- Ändern der Daten in den einzelnen Feldern.

Der Exit XBMOUT (sofern aktiviert) wird aufgerufen, wenn alle folgenden Bedingungen erfüllt sind:

- Der Befehl SEND MAP wurde erfolgreich verarbeitet.

- Die Maske, auf die im Befehl verwiesen wird, enthält mindestens ein Feld, das als VALIDN=USEREXIT angegeben ist.
- Es wurde mindestens ein USEREXIT-Feld im abgehenden Datenstrom generiert.

Mithilfe von XBMOOUT können Sie folgende Aktionen ausführen:

- Analysieren aller als VALIDN=USEREXIT definierten Felder, die im abgehenden Datenstrom generiert wurden.
- Verwenden des Maskengruppennamens, des Maskennamens und der Feldlänge, die in der Maske definiert sind, sowie der tatsächlichen Länge der Felddaten, die in den abgehenden Datenstrom gestellt werden.
- Ändern der Daten in den einzelnen Feldern.
- Ändern der Attribute, die mit den einzelnen Feldern gesendet werden.

An beide Exits werden vier exitspezifische Parameter wie folgt übergeben:

1. Adresse des Tabelleneintrags in der Terminalsteuertabelle (TCTTE - Terminal Control Table Terminal Entry), der der Zuordnungsanforderung zugeordnet ist.
2. Adresse des EXEC-Schnittstellenblocks (EIB) des Systems, der der Task zugeordnet ist, die die Zuordnungsanforderung ausgibt.
3. Adresse einer Halbwort-Binärzahl zur Angabe der Anzahl der Elemente in der *Feldelementtabelle*.
4. Adresse der Feldelementtabelle.

## Beispielprogramm: DFH\$BMXT

CICS stellt ein Beispielprogramm namens DFH\$BMXT bereit, das veranschaulicht, wie zugeordnete Eingabe- und Ausgabedaten mit Bezug auf die in der Feldelementtabelle bereitgestellten Informationen geändert werden können. Darüber hinaus wird auch ein Copybook namens DFHXBMDs bereitgestellt. Bei diesem Copybook handelt es sich um einen Pseudobereich (DSECT), der die Struktur des Feldelements definiert.

## Exit XBMIN

Dieser Exit wird aufgerufen, nachdem Basic Mapping Support (BMS) Eingabezuordnungsoperation erfolgreich verarbeitet hat.

### Zeitpunkt des Aufrufs

Nachdem BMS eine Eingabezuordnungsoperation erfolgreich verarbeitet hat.

### Exitspezifische Parameter

#### UEPBMTCT

Adresse des Terminaleintrags in der Terminalsteuertabelle (TCTTE - Terminal Control Table Terminal Entry), der der Zuordnungsanforderung zugeordnet ist.

#### UEPEXECB

Adresse des EXEC-Schnittstellenblocks (EIB) des Systems, der der Task zugeordnet ist.

#### UEPBMCNT

Adresse der Halbwort-Binärzahl für die Anzahl der "Feldelemente" in der Feldelementtabelle.

#### UEPBMTAB

Adresse der Feldelementtabelle.

## **Rückgabecodes**

### **UERCNORM**

Verarbeitung fortsetzen.

### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

## **XPI-Aufrufe**

Es können alle verwendet werden.

## **Exit XBMOUT**

Dieser Exit wird aufgerufen, nachdem Basic Mapping Support (BMS) eine Ausgabe-seite während einer Ausgabezuordnungsoperation erfolgreich abgeschlossen hat.

### **Zeitpunkt des Aufrufs**

Nachdem Basic Mapping Support (BMS) eine Ausgabe-seite während einer Ausgabezuordnungsoperation erfolgreich abgeschlossen hat.

### **Exitspezifische Parameter**

#### **UEPBMCT**

Adresse des Terminaleintrags in der Terminalsteuertabelle (TCTTE - Terminal Control Table Terminal Entry), der der Zuordnungsanforderung zugeordnet ist.

#### **UEPEXECB**

Adresse des EXEC-Schnittstellenblocks (EIB) des Systems, der der Task zugeordnet ist.

#### **UEPBMCNT**

Adresse der Halbwort-Binärzahl für die Anzahl der "Feldelemente" in der Feldelementtabelle.

#### **UEPBMTAB**

Adresse der Feldelementtabelle.

## **Rückgabecodes**

### **UERCNORM**

Verarbeitung fortsetzen.

### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

## **XPI-Aufrufe**

Es können alle verwendet werden.

## **Struktur der Feldelementtabelle**

Die *Feldelementtabelle* enthält mindestens ein Element, das Informationen zum entsprechenden "relevanten Feld" zur Verfügung stellt, das an den Exit übergeben wird.

Ein "relevantes Feld" ist ein Feld, das in der Maskenquellendatei, die zum Erstellen der in der Zuordnungsoperation referenzierten Maskengruppe verwendet wird, als VALIDN=USEREXIT definiert ist.

Jedes Feldelement hat die folgende Struktur:

### **BMXMAPST**

8-Byte-Bereich mit dem Namen der Maskengruppe, die diesem Feld zugeordnet ist. Wenn in Ihrer CICS-Installation Terminal- oder Alternativsuffixe mit



Maskengruppennamen verwendet werden, kann es sein, dass an den in der Zuordnungsanforderung angegebenen Maskengruppennamen ein Suffix angehängt wird.

#### **BMXMAP**

7-Byte-Bereich mit dem Namen der Maske, die diesem Feld zugeordnet ist.

#### **BMXFDFB**

Ein-Byte-Feld, das aus der Feldspezifikation im Maskenlademodul kopiert wurde. Es enthält folgende Indikatoren:

**X'80'** CASE=MIXED

**X'40'** Gruppenfeldeintrag

**X'20'** Gruppenfelddeskriptor

**X'10'** ATTRB=DET

**X'08'** JUSTIFY=ZERO

**X'04'** JUSTIFY=RIGHT

**X'02'** INITIAL, XINIT oder GINIT ist angegeben

**X'01'** Benanntes Feld (DSECT-Eintrag ist vorhanden)

#### **BMXMAPLN**

Halbwort-Binärwert mit der Feldlänge, die in der Option LENGTH des Makros DFHMDF definiert ist.

#### **BMXACTLN**

Halbwort-Binärwert mit der tatsächlichen Länge der Daten, die in diesem Feld empfangen oder übertragen werden.

#### **BMXDATA**

Adresse der Felddaten.

Im Exit XBMIN verweist BMXDATA auf einen Arbeitsbereich, der von BMS für Eingabezuordnungszwecke abgerufen wurde. Wenn der Exit die Steuerung zurückgibt, wird dieser Arbeitsbereich in die Anwendungsdatenstruktur kopiert, die dieser Maske zugeordnet ist.

Im Exit XBMOUOUT verweist BMXDATA auf einen Terminal-E/A-Bereich (TIOA), in dem BMS einen Ausgabedatenstrom generiert hat. Wenn der Exit die Steuerung zurückgibt, entspricht die Disposition für den TIOA der Disposition der Option TERMINAL (Standard), SET oder PAGING, die in der Anforderung SEND MAP angegeben ist.

#### **BMXATTR**

Nur im Exit XBMOUOUT relevant. Hierbei handelt es sich um die Adresse der Attribute (sofern vorhanden), die BMS in den Ausgabedatenstrom gestellt hat, der diesem Feld vorausgeht.

#### **BMXMAPOF**

Offset des Felds in der Maske. Beispiel: Wenn eine Maske als

MYMAP DFHMDF SIZE=(12,40)

definiert ist und ein Feld in dieser Maske als

FLDA DFHMDF POS=(5,1)

definiert ist, dann ist der Offset dieses Felds (in Bezug auf null) 160 in Dezimalschreibweise. In diesem Beispiel würde BMXMAPOF den Wert X'00A0' enthalten.

### **BMXBUF**

Offset des Felds im Einheitenpuffer. Normalerweise entspricht dieser Wert dem Wert für BMXMAPOF. Dies ist der Fall, wenn die Dimensionen der Maske mit der aktuellen Anzeigegröße übereinstimmen, die von der Einheit verwendet wird. Im Hinblick auf das Beispiel in der Beschreibung von BMXMAPOF bedeutet dies jedoch Folgendes: Wenn MYMAP an eine Einheit gesendet wird, die momentan eine Anzeigegröße von 24x80 verwendet, dann ist der Offset des Felds im Einheitenpuffer (wiederum in Bezug auf null) 320 in Dezimalschreibweise. In diesem Beispiel würde BMXBUF den Wert X'0140' enthalten.

## **Programmierung des Exits XBMIN**

Bei der Programmierung des Exits XBMIN ist es wichtig, die Datenlänge in Betracht zu ziehen.

Die tatsächliche Datenlänge (in BMXACTLN) ist unter Umständen kleiner als die Länge, die in der Maske (in BMXMAPLN) definiert ist. Dies kann beispielsweise der Fall sein, wenn ein Terminalbediener ein Dateneingabefeld nicht vollständig ausgefüllt hat. In diesem Fall hat BMS die Daten im Feld rechts- oder linksbündig ausgerichtet und das Feld mit Leerzeichen oder Nullzeichen aufgefüllt. Diese Ausrichtung und Auffüllung erfolgt vor Aufruf des Exits. Das Exitprogramm kann durch Überprüfen der Biteinstellungen im Feld BMXFDFB ermitteln, wie BMS die Ausrichtung und Auffüllung für das Feld ausgeführt hat.

Die tatsächliche Datenlänge (in BMXACTLN) ist unter Umständen aber auch größer als die Länge, die in der Maske (in BMXMAPLN) definiert ist. Dies kann beispielsweise der Fall sein, wenn eine Maske ein ungeschütztes Feld enthält, auf das nicht unmittelbar ein anderes Feld folgt. Dadurch kann der Terminalbediener Daten über das Ende des Feldes hinaus eingeben. In diesem Fall wird das Datenfeld von BMS entsprechend der Länge abgeschnitten, die in der Maske für das Feld definiert wurde. BMXACTLN enthält jedoch die Länge der Daten, die im eingehenden Datenstrom gefunden wurden.

Wenn Sie Daten im Exit XBMIN ändern, besteht die sicherste Methode darin, die in BMXMAPLN angegebene Länge zu verwenden und dabei gleichzeitig sicherzustellen, dass alle von BMS hinzugefügten Füllzeichen beibehalten werden.

BMXATTR muss im Exit XBMIN ignoriert werden; BMXATTR enthält immer binäre Nullen.

## **Programmierung des Exits XBMOUT**

Bei der Programmierung des Exits XBMOUT ist es wichtig, die tatsächliche Datenlänge in Betracht zu ziehen.

Die tatsächliche Datenlänge (in BMXACTLN) ist unter Umständen kleiner als die Länge, die in der Maske (in BMXMAPLN) definiert ist. Dies liegt daran, dass BMS für jedes Ausgabefeld eine Komprimierung der abschließenden Nullen durchführt.

Die tatsächliche Länge der Daten kann im Exitprogramm nicht geändert werden. Der Exit wird nach der Generierung des Ausgabedatenstroms aufgerufen. Daher könnte ein Versuch, die Datenlänge zu ändern, zu einem ungültigen Datenstrom führen. Wenn also ein Exitprogramm für XBMOUT Daten ändert, muss es dabei den Längenwert in BMXACTLN berücksichtigen.

BMXDATA kann einen Nullwert enthalten. Dies kann durch eine Anforderung vom Typ SEND MAP mit der Option MAPONLY verursacht werden, wenn die Maske

Felder ohne Standarddaten enthält. Dies führt dazu, dass BMS eine Attributsequenz für das Feld sendet, aber keine Daten.

BMXATTR kann einen Nullwert enthalten. Dies kann durch eine Anforderung vom Typ SEND MAP mit der Option DATAONLY verursacht werden, wenn die Anwendung die Daten in einem Feld aktualisiert und nicht die Attribute.

### **Kumulatives Mapping**

Wenn eine Anwendung kumulatives Mapping durchführt, d. h. eine Folge von SEND MAP-Befehlen mit der Option ACCUM ausgibt, dann erstellt BMS eine Verbundanzeige, in der eine einzelne Ausgabeseite aus mehreren SEND-MAP-Anforderungen erstellt worden sein kann.

Beim kumulativen Mapping wird der Exit XBMOUT aufgerufen, nachdem eine Seite erstellt worden ist, nicht bei der Verarbeitung jeder einzelnen SEND MAP-Anforderung.

### **Nachrichtenrouting**

Wenn eine Anwendung eine Routing-Nachricht erstellt, also beispielsweise einen Befehl vom Typ ROUTE ausgibt, gefolgt von mindestens einem Befehl vom Typ SEND MAP mit Angabe der Option SET oder PAGING, dann wird der Exit XBMOUT auf dieselbe Weise aufgerufen wie bei einer nicht weitergeleiteten Zuordnungsanforderung.

Der Parameter UEPBMTCT wird bei einer weitergeleiteten Nachricht jedoch als Nullwert übergeben. Dies liegt daran, dass eine weitergeleitete Nachricht für mehrere Einheiten bestimmt sein kann; und dass BMS die Funktionen optimiert hat, die von den Einheiten unterstützt werden, an die die weitergeleitete Nachricht gerichtet ist. Bei der Verarbeitung einer weitergeleiteten Nachricht im Exit XBMOUT wäre die Referenzierung des Terminaleintrags in der Terminalsteuertabelle (TCTTE - Terminal Control Table Terminal Entry) wahrscheinlich für keine dieser Einheiten relevant.

---

## **Exit XFAINTU für Brückeneinrichtung**

Der Exit für Brückeneinrichtung wird unmittelbar nach dem Erstellen einer neuen Brückeneinrichtung bzw. unmittelbar vor dem Löschen einer Brückeneinrichtung aufgerufen.

Eine Brückeneinrichtung kann am Ende einer Task gelöscht werden, wenn für den Erhaltungszeitraum null angegeben ist oder wenn die Erhaltungszeit abläuft, bevor die Einrichtung erneut verwendet wird.

### **Exit XFAINTU**

#### **Zeitpunkt des Aufrufs**

Unmittelbar nach dem Erstellen bzw. unmittelbar vor dem Löschen einer Brückeneinrichtung.

#### **Exitspezifische Parameter**

##### **UEPFAREQ**

Adresse eines 1-Byte-Felds zur Angabe des Grundes für den Aufruf des Exits. Mögliche Werte:

##### **UEPFAIN**

Initialisierung.

**UEPFATU**  
Bereinigung.

**UEPFATUT**  
Adresse eines 1-Byte-Felds zur Angabe des Typs der erforderlichen  
Bereinigung. Mögliche Werte:

**UEPFANTU**  
Normale Bereinigung.

**UEPFAETU**  
Bereinigung wegen Zeitablauf.

**UEPFANAM**  
Adresse des Namens der Brückeneinrichtung.

**UEPFATYP**  
Adresse eines 1-Byte-Felds zur Angabe des Typs der Brückenein-  
richtung. Der Wert lautet immer wie folgt:

**UEPFABR**  
3270-Brückeneinrichtung.

**UEPFAUAA**  
Adresse des Benutzerbereichs der Brückeneinrichtung (TCTUA).

**UEPFAUAL**  
Adresse eines Ein-Byte-Felds mit der Länge des Benutzerbereichs  
der Brückeneinrichtung.

**UEPFATK**  
Adresse des 8-Byte-Tokens der Brückeneinrichtung.

**UEPFAMCH**  
Adresse eines 1-Byte-Felds zur Angabe des verwendeten Mechanis-  
mus zum Starten der überbrückten Transaktion mithilfe dieser  
Brückeneinrichtung. Mögliche Werte:

**UEPFASTA**  
Start mithilfe von START BREXIT.

**UEPFALNK**  
Start mithilfe eines Links zu DFHL3270.

**UEPFAREG**  
Adresse eines 1-Byte-Felds zur Angabe, ob die Region Eigner der  
Brückeneinrichtung ist oder ob es sich um eine ferne Einrichtung  
handelt. Eigner einer Brückeneinrichtung kann die Anwendungs-  
verwaltungsregion (AOR) sein, in der die Einrichtung lokal vor-  
handen ist, während eine Brückeneinrichtung für die Routerregion  
als fern gilt. Hierbei ist zu beachten, dass XFAINTU in einer Regi-  
on zweimal aufgerufen werden kann, wenn es sich bei der AOR  
und dem Router um dieselbe Region handelt. Mögliche Werte:

**UEPFAROU**  
Diese Region ist der Router für diese Brückeneinrichtung.

**UEPFAAOR**  
Diese Region ist die Anwendungsverwaltungsregion (AOR)  
für diese Brückeneinrichtung.

## **Rückgabecodes**

## UERCNORM

Verarbeitung fortsetzen.

### XPI-Aufrufe

Es können alle Befehle verwendet werden, mit Ausnahme derjenigen, die wiederherstellbare Ressourcen verwenden.

### API-Aufrufe

Es können alle Befehle verwendet werden, mit Ausnahme derjenigen, die taskbezogene Benutzerexits aufrufen oder wiederherstellbare Ressourcen verwenden.

---

## Exits XDTRD, XDTAD und XD TLC zur Datentabellenverwaltung

Die Exits zur Datentabellenverwaltung gelten sowohl für gemeinsam genutzte CICS-Datentabellen als auch für Datentabellen der CICS-Coupling-Facility.

Die Exits XDTRD und XDTAD ermöglichen es Ihnen, die Auswahl von Datensätzen für die Aufnahme in eine Datentabelle zu steuern. Hierbei wird XDTRD verwendet, um eine entsprechende Auswahl beim Laden vorzunehmen, und XDTAD wird aufgerufen, wenn Datensätze anschließend zu einer geladenen Datentabelle hinzugefügt werden (oder zu einer Coupling-Facility-Datentabelle, die nicht geladen werden musste). Mithilfe von XDTRD kann darüber hinaus der Inhalt von Datensätzen, die in einer benutzerverwalteten Tabelle oder einer Coupling-Facility-Datentabelle enthalten sind, vor dem Hinzufügen geändert werden.

Bei gemeinsam genutzten CICS-Datentabellen bietet Ihnen XD TLC die Möglichkeit, nach Beendigung des Ladevorgangs für eine Datentabelle entsprechende Aktionen auszuführen, beispielsweise um bestimmte Einschränkungen aufzuheben, die Sie unter Umständen für den Zugriff auf die Datentabelle während des Ladens eingerichtet haben, oder um Maßnahmen im Falle einer nicht erfolgreichen Beendigung des Ladevorgangs zu ergreifen.

Bei Coupling-Facility-Datentabellen ermöglicht XD TLC es Ihrem globalen Benutzerexitprogramm zu entscheiden, ob eine nicht erfolgreich geladene Coupling-Facility-Datentabelle akzeptiert werden soll oder nicht. Wenn das Benutzerexitprogramm entscheidet, die Tabelle zu akzeptieren, bleibt sie geöffnet und steht für den Zugriff zur Verfügung, wird von CICS jedoch nicht als vollständig geladen markiert. Dies ist auch die Standardaktion, wenn kein XD TLC-Exit aktiviert ist. Dies bedeutet, dass Anwendungsprogramme weiterhin die Bedingung **LOADING** für alle Datensätze erhalten, die außerhalb des Schlüsselbereichs von Datensätzen liegen, die erfolgreich in die Tabelle geladen wurden. Dadurch wird sichergestellt, dass die Anwendungsprogramme wissen, dass nicht alle erwarteten Daten verfügbar sind. Außerdem haben Sie so die Möglichkeit, den Ladevorgang zu wiederholen, nachdem die Ursache des Fehlers behoben wurde, indem Sie die Datei, die den Ladevorgang eingeleitet hat, schließen und dann erneut öffnen. Als Alternative können Sie auch eine andere ladefähige Datei öffnen, die auf dieselbe Datentabelle verweist. Wenn Ihr Exitprogramm entscheidet, die Tabelle zurückzuweisen, wird sie geschlossen, und die bereits geladenen Datensätze verbleiben in der Tabelle. Im Anschluss an die Behebung der Fehlerursache kann der Ladevorgang durch Öffnen der Datentabelle abgeschlossen werden. Bei Coupling-Facility-Datentabellen, die nicht aus einem Quelldataset geladen werden, wird der Exit XD TLC nicht aufgerufen.

Bitte beachten Sie, dass ein Programm, das über einen dieser Exitpunkte aufgerufen wird, einen Pseudobereich (DSECT) deklarieren muss, der die Parameterliste der Benutzerexits für Datentabellen definiert, auf die durch das Feld UEPDTPL

verwiesen wird. (Wenngleich das Feld UEPDTPL durch einen DFHUEEXIT-Aufruf definiert wird, gilt dies nicht für die Parameterliste, die durch dieses Feld adressiert wird.) Zu diesem Zweck kann Ihr Programm das Copybook DFHXDTDS einschließen, das den Pseudobereich (DSECT) DT\_UE\_PLIST definiert.

Wenn für Tabellen OPENTIME=STARTUP angegeben ist oder wenn Tabellen implizit geöffnet werden, dann sollten Sie ein PLTPI-Programm (Program List Table Post-Initialization) bereitstellen, um die Benutzerexits zu aktivieren. Andernfalls kann es sein, dass der Ladevorgang für die Datentabelle bereits beginnt, bevor die Exits aktiviert werden können. Weitere Informationen zu PLTPI-Programmen finden Sie unter Writing initialization and shutdown programs.

**Anmerkung:** Weitere Informationen zur Verwendung dieser Exits mit der Unterstützung für gemeinsam genutzte CICS-Datentabellen finden Sie unter Shared data tables overview.

## Exit XDTRD

Der Benutzerexit XDTRD wird unmittelbar vor einem Versuch von CICS aufgerufen, einen Datensatz, der aus dem Quellendataset abgerufen wurde, zur Datentabelle hinzuzufügen.

Dies ist normalerweise dann der Fall, wenn im Rahmen des Ladeprozesses während des sequenziellen Kopierens des Quellendatasets ein Datensatz abgerufen wird. Diese Situation kann jedoch auch auftreten, wenn eine Anwendung einen Datensatz abrufen, der nicht in der Datentabelle enthalten ist, weil einer der folgenden Gründe vorliegt:

- Bei einer benutzerverwalteten Datentabelle: Der Ladevorgang ist noch nicht abgeschlossen.
- Bei einer CICS-verwalteten Datentabelle: Der Ladevorgang wurde beendet, bevor das Ende des Quellendatasets erreicht wurde (beispielsweise weil die Datentabelle voll war).

**Anmerkung:** Bei einer Coupling-Facility-Datentabelle wird der Exit XDTRD nur dann aufgerufen, wenn die Tabelle aus einem Quellendataset geladen wird.

Der aus dem Quellendataset abgerufene Datensatz wird als Parameter an das Benutzerexitprogramm übergeben (siehe Felder UEPDTRA und UEPDTRL). Ihr Exitprogramm kann entscheiden, ob der Datensatz in die Datentabelle eingeschlossen werden soll oder nicht. Diese Entscheidung hängt beispielsweise von den Schlüsselwerten ab (siehe Felder UEPDTKA und UEPDTKL).

Alternativ kann das Exitprogramm anfordern, dass alle nachfolgenden Datensätze bis zu einem angegebenen Schlüssel übersprungen werden (siehe Feld UEPDTSKA); diese Datensätze werden dann nicht an das Exitprogramm übergeben. Diese Funktion steht nur beim Laden zur Verfügung. Sie können den Schlüssel als vollständigen Schlüssel angeben, oder Sie können nur die führenden Zeichen angeben, indem Sie den Bereich mit dem Schlüssel für das Überspringen mit binären Nullen auffüllen.

Bei einer benutzerverwalteten Datentabelle kann das Programm auch die Daten im Datensatz ändern, um den Speicherbedarf für die Datentabelle zu reduzieren. Anwendungsprogramme, die die Datentabelle verwenden, müssen über alle Änderungen informiert werden, die das Exitprogramm am Datensatzformat vornimmt. Wird die Satzlänge geändert, muss das Exitprogramm die neue Länge in der Para-

meterliste entsprechend festlegen (siehe Feld UEPDTRL). Die neue Länge darf die Datenpufferlänge nicht überschreiten (siehe Feld UEPDTRBL).

#### **Zeitpunkt des Aufrufs**

Unmittelbar bevor CICS versucht, einen Datensatz, der aus dem Quelldatensatz abgerufen wurde, zur Datentabelle hinzuzufügen.

#### **Exitspezifische Parameter**

##### **UEPDTPPL**

Adresse der Parameterliste des Benutzerexits für die Datentabelle; diese Liste wird durch den Pseudobereich (DSECT) DT\_UE\_PLIST im Copybook DFHXDTDS zugeordnet. Die Parameterliste des Benutzerexits für die Datentabelle enthält Folgendes:

##### **UEPDTCNAM**

Name der Datentabelle aus 8 Zeichen.

##### **UEPDTCFLG**

1-Byte-Flagfeld. Mögliche Biteinstellungen:

##### **UEPDTCSDT (X'80')**

Der Exit wurde durch die CICS-Unterstützung für gemeinsam genutzte Datentabellen (Shared Data Tables, SDT) aufgerufen.

##### **UEPDTCMT (X'40')**

Es handelt sich um eine CICS-verwaltete Tabelle. Dies ist nur dann aussagekräftig, wenn UEPDTCSDT aktiviert ist.

##### **UEPDTCOPT (X'20')**

Der Exit wurde für einen Tabellenladevorgang aufgerufen. Dies bedeutet, dass eine Optimierung durch Überspringen angefordert werden kann.

##### **UEPDTCFT(X'10')**

Der Exit wurde von der Unterstützung für Coupling-Facility-Datentabellen aufgerufen.

##### **UEPDTCUMT (X'08')**

Es handelt sich um eine benutzerverwaltete Tabelle. Dies ist nur dann aussagekräftig, wenn UEPDTCSDT aktiviert ist.

##### **UEPDTCRA**

Adresse des Datensatzes.

##### **UEPDTCRBL**

Vollwortlänge des Datentabellenpuffers.

##### **UEPDTCRL**

Vollwortlänge des Datensatzes.

Bei benutzerverwalteten Tabellen kann das Exitprogramm in diesem Feld eine neue Länge festlegen, wenn der Datensatz geändert wird.

##### **UEPDTCKA**

Adresse des Datentabellenschlüssels.

##### **UEPDTCKL**

Vollwortlänge des Datentabellenschlüssels.

#### **UEPDSDL**

Vollwortlänge des Namens des Quelldatensatzes. Dies ist nur dann aussagekräftig, wenn entweder UEPDSDT oder UEPDTCFT aktiviert ist.

#### **UEPDTSN**

Feld aus 44 Zeichen mit dem Namen des Quelldatensatzes. Dies ist nur dann aussagekräftig, wenn entweder UEPDSDT oder UEPDTCFT aktiviert ist.

#### **UEPDTSKA**

Adresse eines Bereichs für einen Schlüssel für Überspringen. Wird das Exitprogramm zum Laden von Tabellen aufgerufen, kann es in diesem Bereich einen Schlüssel mit der in UEPDTKL angegebenen Länge zurückgeben und eine Optimierung des Ladevorgangs anfordern, indem es den Rückgabecode UERCDDTOP festlegt. Dies ist nur dann aussagekräftig, wenn entweder UEPDSDT oder UEPDTCFT aktiviert ist.

### **Rückgabecodes**

#### **UERCDDTAC**

Datensatz zur Datentabelle hinzufügen.

#### **UERCDDTRJ**

Datensatz zurückweisen. Dies bedeutet, dass der Datensatz nicht zur Tabelle hinzugefügt wird.

#### **UERCDDTOP**

Diesen Datensatz und alle folgenden Datensätze überspringen, bis ein Schlüssel gefunden wird, der größer-gleich dem Schlüssel ist, der im Bereich für den Schlüssel für Überspringen angegeben wurde. Dies ist nur dann aussagekräftig, wenn entweder UEPDSDT oder UEPDTCFT aktiviert ist.

### **XPI-Aufrufe**

Es können alle verwendet werden.

## **Exit XDTAD**

Der Exit XDTAD wird aufgerufen, wenn eine Schreib Anforderung an eine Datentabelle ausgegeben wird.

Bei benutzerverwalteten Datentabellen und Coupling-Facility-Datentabellen wird der Benutzerexit einmal aufgerufen, und zwar bevor der Datensatz zur Datentabelle hinzugefügt wird. Bei CICS-verwalteten Datentabellen wird der Benutzerexit zweimal aufgerufen, und zwar erstmalig, bevor der Datensatz zum Quelldatensatz hinzugefügt wird, und dann erneut, bevor der Datensatz zur Datentabelle hinzugefügt wird.

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

Der von der Anwendung geschriebene Datensatz wird als Parameter an das Benutzerexitprogramm übergeben (siehe Felder UEPDTRA und UEPDTRL). Ihr Exitprogramm kann entscheiden, ob der Datensatz in die Datentabelle eingeschlossen werden soll oder nicht. Diese Entscheidung hängt beispielsweise von den



Schlüsselwerten ab (siehe Felder UEPDTKA und UEPDTKL). Diese Entscheidung wird durch Festlegung des entsprechenden Rückgabecodes angezeigt.

Der Exit XDTAD darf die Daten im Datensatz nicht ändern. Wenn Sie XDTRD verwendet haben, um die Datensätze beim Laden der Datentabelle abzuschneiden, müssen Sie Ihre Anwendung so codieren, dass nur versucht wird, solche Datensätze zu schreiben, die das korrekte Format für die Datentabelle aufweisen.

Ein Beispiexitprogramm für XDTAD finden Sie unter Shared data tables overview.

### **Zeitpunkt des Aufrufs**

Einmal oder mehrmals während der Verarbeitung einer Schreib Anforderung an eine Datentabelle.

### **Exitspezifische Parameter**

#### **UEPDTPPL**

Adresse der Parameterliste des Benutzerexits für die Datentabelle; diese Liste wird durch den Pseudobereich (DSECT) DT\_UE\_PLIST im Copybook DFHXDTDS zugeordnet. Die Parameterliste des Benutzerexits für die Datentabelle enthält Folgendes:

#### **UEPDTPNAM**

Name der Datentabelle aus 8 Zeichen.

#### **UEPDTPFLG**

1-Byte-Flagfeld. Mögliche Biteinstellungen:

#### **UEPDTPSDT (X'80')**

Der Exit wurde durch die CICS-Unterstützung für gemeinsam genutzte Datentabellen (Shared Data Tables, SDT) aufgerufen.

#### **UEPDTPCMT (X'40')**

Es handelt sich um eine CICS-verwaltete Tabelle. Dies ist nur dann aussagekräftig, wenn UEPDTPSDT aktiviert ist.

#### **UEPDTPCFT(X'10')**

Der Exit wurde von der Unterstützung für Coupling-Facility-Datentabellen aufgerufen.

#### **UEPDTPUMT (X'08')**

Es handelt sich um eine benutzerverwaltete Tabelle. Dies ist nur dann aussagekräftig, wenn UEPDTPSDT aktiviert ist.

#### **UEPDTPRA**

Adresse des Datensatzes.

#### **UEPDTPRBL**

Vollwortlänge des Datentabellenpuffers.

#### **UEPDTPRL**

Vollwortlänge des Datensatzes.

#### **UEPDTPKA**

Adresse des Datentabellenschlüssels.

#### **UEPDTPKL**

Vollwortlänge des Datentabellenschlüssels.

#### **UEPDTDSL**

Vollwortlänge des Namens des Quelldatensets. Dies ist nur dann aussagekräftig, wenn entweder UEPDTSMT oder UEPDTCFT aktiviert ist.

#### **UEPDTDSN**

Feld aus 44 Zeichen mit dem Namen des Quelldatensets. Dies ist nur dann aussagekräftig, wenn entweder UEPDTSMT oder UEPDTCFT aktiviert ist.

### **Rückgabecodes**

#### **UERCMTAC**

Datensatz zur Datentabelle hinzufügen.

#### **UERCMTRJ**

Datensatz zurückweisen. Dies bedeutet, dass der Datensatz nicht zur Tabelle hinzugefügt wird.

### **XPI-Aufrufe**

Es können alle verwendet werden.

## **Exit XDTC**

Der Benutzerexit XDTC wird bei Beendigung des Ladens der Datentabelle aufgerufen, und zwar unabhängig davon, ob der Ladevorgang erfolgreich war oder nicht. Der Benutzerexit wird nicht aufgerufen, wenn die Datentabelle aus irgendeinem Grund vor Beendigung des Ladevorgangs geschlossen wird. Bei einer Coupling-Facility-Datentabelle wird der Exit XDTC nur dann aufgerufen, wenn die Tabelle aufgerufen, die aus einem Quelldatensatz geladen wird.

Das Exitprogramm wird benachrichtigt, wenn der Ladevorgang nicht erfolgreich beendet wurde (siehe Feld UEPDTORC). Dies kann beispielsweise vorkommen, wenn die maximal zulässige Anzahl an Datensätzen erreicht wurde oder der virtuelle Speicher nicht ausreichte. In diesem Fall kann das Exitprogramm die sofortige Schließung der Datei anfordern, indem der entsprechende Rückgabecode gesetzt wird.

### **Zeitpunkt des Aufrufs**

Bei Beendigung des Ladevorgangs für eine Tabelle. Der Exit wird hingegen nicht aufgerufen, wenn der Ladevorgang aufgrund einer Schließung der Datentabelle beendet wurde.

### **Exitspezifische Parameter**

#### **UEPDTPM**

Adresse der Parameterliste des Benutzerexits für die Datentabelle; diese Liste wird durch den Pseudobereich (DSECT) DT\_UE\_PLIST im Copybook DFHXDTDS zugeordnet. Die Parameterliste des Benutzerexits für die Datentabelle enthält Folgendes:

#### **UEPDTCNM**

Name der Datentabelle aus 8 Zeichen.

#### **UEPDTCFL**

1-Byte-Flagfeld. Mögliche Biteinstellungen:

#### **UEPDTSMT (X'80')**

Der Exit wurde durch die CICS-Unterstützung für gemeinsam genutzte Datentabellen (Shared Data Tables, SMT) aufgerufen.

**UEPDTCMT (X'40')**

Es handelt sich um eine CICS-verwaltete Tabelle. Dies ist nur dann aussagekräftig, wenn UEPDTSMT aktiviert ist.

**UEPDTCFT(X'10')**

Der Exit wurde von der Unterstützung für Coupling-Facility-Datentabellen aufgerufen.

**UEPDTSMT (X'08')**

Es handelt sich um eine benutzerverwaltete Tabelle. Dies ist nur dann aussagekräftig, wenn UEPDTSMT aktiviert ist.

**UEPDTCRC**

Ergebniscode für offene Datentabellen. Mögliche Werte:

**UEPDTCRC**

Laden war erfolgreich.

**UEPDTCFL**

Laden war nicht erfolgreich.

**UEPDTCDSL**

Vollwortlänge des Namens des Quelldatensatzes. Dies ist nur dann aussagekräftig, wenn entweder UEPDTSMT oder UEPDTCFT aktiviert ist.

**UEPDTCDSN**

Feld aus 44 Zeichen mit dem Namen des Quelldatensatzes. Dies ist nur dann aussagekräftig, wenn entweder UEPDTSMT oder UEPDTCFT aktiviert ist.

**Rückgabecodes****UERCDCOK**

Datentabelle im aktuellen Status akzeptieren.

**UERCDCCL**

Datentabelle schließen.

**XPI-Aufrufe**

Es können alle verwendet werden.

---

## **Exit XXDFA für das DBCTL-Schnittstellensteuerprogramm**

Dieser Exit wird durch ein aktives CICS-System aufgerufen, wenn dessen Verbindung zu DBCTL fehlschlägt.

**Zeitpunkt des Aufrufs**

Durch ein aktives CICS-System, wenn dessen Verbindung zu DBCTL fehlschlägt. Das Exitprogramm wird aufgerufen, nachdem das aktive CICS-System das alternative CICS-System über den Fehler informiert hat.

**Exit-spezifische Parameter****UEPDBXR**

Adresse von CICS-XRF-Informationen zur Verwendung mit DBCTL. Die CICS-XRF-Informationen können mithilfe des Pseudobereichs (DSECT) DFHDXUEP zugeordnet werden.

**Rückgabecodes**

**UERCNOAC**

Keine Aktion ausführen.

**UERCSWCH**

Zur alternativen Datenbanksteuerung (DBCTL) umschalten.

**UERCABNO**

CICS ohne Speicherauszug abbrechen.

**UERCABDU**

CICS mit Speicherauszug abbrechen.

**XPI-Aufrufe**

TRANSACTION\_DUMP darf nicht verwendet werden.

---

## Exits für DBCTL-Überwachungsprogramm (XXDFB, XXDTO)

Diese Exits werden aufgerufen, wenn die Verbindung zu DBCTL fehlschlägt oder wenn CICS eine Übernahme durchführt.

### Exit XXDFB

Der Exit XXDFB wird aufgerufen, wenn eine Nachricht vom aktiven CICS-System empfangen wird, die angibt, dass die Verbindung zu DBCTL fehlgeschlagen ist.

**Zeitpunkt des Aufrufs**

Durch das alternative CICS-System, wenn es vom aktiven CICS-System eine Nachricht empfängt, die angibt, dass die Verbindung zu DBCTL fehlgeschlagen ist. Das alternative und das aktive CICS-System werden in unterschiedlichen MVS-Images ausgeführt, möglicherweise in unterschiedlichen zentralen Verarbeitungskomplexen (Central Processing Complexes, CPCs). Weitere Informationen zu diesen Exits finden Sie unter Overview of Database Control (DBCTL).

**Exitspezifische Parameter****UEPDBXR**

Adresse von CICS-XRF-Informationen zur Verwendung mit DBCTL. Die CICS-XRF-Informationen können mithilfe des Pseudobereichs (DSECT) DFHDXUEP zugeordnet werden.

**Rückgabecodes****UERCNOAC**

Keine Aktion ausführen.

**UERCSWCH**

Zur alternativen Datenbanksteuerung (DBCTL) umschalten.

**UERCABNO**

CICS ohne Speicherauszug abbrechen.

**UERCABDU**

CICS mit Speicherauszug abbrechen.

Der Rückgabecode UERCNORM ist an diesem Exitpunkt nicht verfügbar.

**XPI-Aufrufe**

Folgende Aufrufe dürfen nicht verwendet werden:

- INQUIRE\_MONITORING\_DATA
- MONITOR
- TRANSACTION\_DUMP

- WRITE\_JOURNAL\_DATA

## Exit XXDTO

Der Exit XXDTO wird durch ein alternatives CICS-System aufgerufen, wenn es eine Übernahme durchführt.

### Zeitpunkt des Aufrufs

Durch ein alternatives CICS-System aufgerufen, wenn es unter den folgenden Bedingungen eine Übernahme durchführt:

- Das aktive und das alternative CICS-System befinden sich in unterschiedlichen MVS-Images, möglicherweise auf unterschiedlichen Prozessoren.
- Das aktive CICS-System war mit einem DBCTL-Subsystem verbunden oder versuchte, eine Verbindung zu einem solchen herzustellen. (Dies gilt nicht für das Unterbrechen einer Verbindung zu einem DBCTL-Subsystem und das Wiederherstellen einer Verbindung zu einem anderen DBCTL-Subsystem.)
- Die Übernahme wurde nicht durch den Exit XXDFB eingeleitet, oder die Übernahme wurde von XXDFB eingeleitet, aber das aktive System stellte eine DBCTL-Verbindung wieder her, bevor die Übernahme stattfand, und XXDTO wurde für eine neue DBCTL-Übernahmeentscheidung gesteuert.

### Exitspezifische Parameter

#### UEPDBXR

Adresse von CICS-XRF-Informationen zur Verwendung mit DBCTL. Die CICS-XRF-Informationen können mithilfe des Pseudobereichs (DSECT) DFHDXUEP zugeordnet werden.

### Rückgabecodes

#### UERCNOAC

Keine Aktion ausführen.

#### UERCWCH

Zur alternativen Datenbanksteuerung (DBCTL) umschalten.

#### UERCABNO

CICS ohne Speicherauszug abbrechen.

#### UERCABDU

CICS mit Speicherauszug abbrechen.

Der Rückgabecode UERCNORM ist an diesem Exitpunkt nicht verfügbar.

### XPI-Aufrufe

Folgende Aufrufe dürfen nicht verwendet werden:

- INQUIRE\_MONITORING\_DATA
- MONITOR
- TRANSACTION\_DUMP
- WRITE\_JOURNAL\_DATA

---

## Exits XDSBWT und XDSAWT für die Dispatcherdomäne

Die Exitpunkte XDSBWT und XDSAWT befinden sich vor und nach dem Wartestatus des Betriebssystems. In Exitprogrammen, die über diese Exitpunkte aufgerufen werden, können keine CICS-Services verwendet werden.

Die Exits XDSBWT und XDSAWT können verwendet werden, um den Auslagerungsstatus des CICS-Adressraums zu steuern. Wenn der Standardstatus des Adressraums jedoch 'Nicht verlagerbar' (NON-SWAPPABLE) ist, können Sie diese Exits nicht verwenden, um diesen Status außer Kraft zu setzen.

CICS verwendet einen Zähler, der für jede Anforderung vom Typ SYSEVENT DONTSWAP erhöht und für jede Anforderung vom Typ SYSEVENT OKSWAP bis zu einem Minimum von 0 verringert wird. Eine SYSEVENT DONTSWAP-Anforderung wird ausgegeben, wenn dieser Zähler von 0 auf 1 steigt. Eine SYSEVENT OKSWAP-Anforderung wird ausgegeben, wenn dieser Zähler von 1 auf 0 zurückgeht. In allen anderen Fällen wird das SYSEVENT nicht ausgegeben.

## Exit XDSBWT

Dieser Exit wird vor einer vom quasiwiedereintrittsfähigen CICS-Tasksteuerblock (TCB) ausgegebenen WAIT-Anforderung für das Betriebssystem aufgerufen.

### Zeitpunkt des Aufrufs

Vor einer vom quasiwiedereintrittsfähigen CICS-Tasksteuerblock (TCB) ausgegebenen WAIT-Anforderung für das Betriebssystem.

### Exitspezifische Parameter

Keine.

### Rückgabecodes

#### UERCNORM

Verarbeitung fortsetzen.

#### UERC\_SWAP

Ausgabe von SYSEVENT, um die Auslagerungsfunktion für Adressräume zu ermöglichen.

### XPI-Aufrufe

Dürfen nicht verwendet werden.

## Exit XDSAWT

Dieser Exit wird nach einer vom quasiwiedereintrittsfähigen CICS-Tasksteuerblock (TCB) ausgegebenen WAIT-Anforderung für das Betriebssystem aufgerufen.

### Zeitpunkt des Aufrufs

Nach einer vom quasiwiedereintrittsfähigen CICS-Tasksteuerblock (TCB) ausgegebenen WAIT-Anforderung für das Betriebssystem.

### Exitspezifische Parameter

#### UEPSYSRC

Adresse des 4-Byte-Rückgabecodes aus der Anforderung vom Typ SYSEVENT, die vor der WAIT-Anforderung für das Betriebssystem ausgegeben wurde. Dieser Rückgabecode liegt in einem der beiden folgenden Formate vor:

1. Rückgabecode von SYSEVENT OKSWAP.
2. Wenn die SYSEVENT-Anforderung von CICS zurückgewiesen wurde: spezieller CICS-Rückgabecode mit einem der folgenden Dezimalwerte:

- |    |   |
|----|---|
| 17 | Die Anforderung SYSEVENT OKSWAP wurde nicht ausgegeben. Die Anzahl der ausstehenden Anforderungen vom Typ SYSEVENT OKSWAP überschreitet die Anzahl der Anforderungen vom Typ SYSEVENT |
|----|---|

DONTSWAP. Bevor eine Anforderung vom Typ SYSEVENT OKSWAP ausgegeben werden kann, muss eine Anforderung vom Typ SYSEVENT DONTSWAP angefordert werden.

- 19 Die Anforderung SYSEVENT OKSWAP wurde nicht ausgegeben. Die Anzahl der ausstehenden Anforderungen vom Typ SYSEVENT DONTSWAP überschreitet weiterhin die Anzahl der Anforderungen vom Typ SYSEVENT OKSWAP. Es müssen weitere Anforderungen vom Typ SYSEVENT OKSWAP angefordert werden, bevor eine Anforderung vom Typ SYSEVENT OKSWAP von CICS ausgegeben wird.

#### Rückgabecodes

##### UERCNORM

Verarbeitung fortsetzen.

##### UERCNOSW

SYSEVENT zum Unterdrücken der Auslagerungsfunktion für Adressräume ausgeben.

#### XPI-Aufrufe

Dürfen nicht verwendet werden.

---

## Exits XDLIPRE und XDLIPOST des DL/I-Schnittstellenprogramms

Die Exitpunkte XDLIPRE und XDLIPOST werden im Anschluss an die Ausgabe eines **EXEC-DLI**-Befehls oder DL/I-Aufrufs aufgerufen. Der Exit XDLIPRE wird aufgerufen, bevor die Anforderung verarbeitet wird, und XDLIPOST wird nach der Verarbeitung der Anforderung aufgerufen.

Wenn die Anforderung über Funktionen übertragen wird, werden die Exits sowohl aus der Anwendungsverwaltungsregion als auch aus der Datenbankverwaltungsregion heraus aufgerufen. Für den Aufruf in einer Datenbankverwaltungsregion gibt es jedoch Einschränkungen:

1. Die Beschreibungen der Exits zeigen das allgemeine Format der Parameterliste der Anwendung an. Ausführliche Informationen zum Format der DL/I-Parameterliste auf Aufrufebene (CALL) finden Sie unter DL/I calls reference in IMS product documentation.
2. Bei allen EXEC-DLI-Aufrufen liegt die Parameterliste der Anwendung im Format der Assemblersprache vor. Dies bedeutet Folgendes: Der Wert des Byte für die Programmsprache, auf den von UEPLANG verwiesen wird, lautet immer UEPASM, und die Parameterliste, auf die von UEPAPLIST verwiesen wird, liegt stets im Format der Assemblersprache vor. Dieses Format wird verwendet, weil alle EXEC-DLI-Aufrufe in Anforderungen in der Assemblersprache auf Aufrufebene (CALL) konvertiert werden.

Eine EXEC-DLI-Onlineanforderung wird von DFHEDP in eine Anforderung auf CALL-Ebene für DFHDLI konvertiert. IMS befasst sich nicht direkt mit Parameterlisten auf EXEC-Ebene. Der erste Parameter in der Parameterliste für CALL enthält die Adresse der Parameteranzahl. Der zweite Parameter in der Parameterliste für CALL enthält die Adresse der Funktion. Alle anderen Parameter hängen von der Funktion ab.

3. In einem Exitprogramm für XDLIPRE können Sie den PSB-Namen und den SYSID-Namen ändern. Durch eine Änderung des Namens wird die Verfügbarkeit verbessert, wenn die ursprünglich angegebene SYSID fehlschlägt.

Sie können die SYSID wie folgt ändern:

- Sie können einen fernen Wert in einen anderen fernen Wert ändern.
- Sie können einen lokalen Wert in einen fernen Wert ändern.
- Sie können einen fernen Wert in einen lokalen Wert ändern.

Das Ändern der SYSID wirkt sich nur dann aus, wenn der zugeordnete PSB einen PDIR-Eintrag aufweist. Bei der SYSID kann es sich um das lokale CICS-System handeln (das heißt, um den in der CICS-Region angegebenen Namen im Systeminitialisierungsparameter SYSIDNT) oder um einen fernen Verbindungsnamen. Damit die neue SYSID verwendet werden kann, muss der PSB-Name einen PDIR-Eintrag aufweisen. Ist kein PDIR-Eintrag vorhanden, wird davon ausgegangen, dass das lokale CICS-System mit DBCTL verbunden ist, und es wird versucht, die IMS-Anforderung dort auszuführen. Ein IMS-Zeitplanfehler wird auf die gleiche Art und Weise behandelt wie ein Fehler beim Weiterleiten an eine Verbindung, die nicht vorhanden ist. Wenn die SYSID entweder in denselben Wert wie für SYSIDNT im lokalen CICS-System oder in Leerzeichen (Hexadezimalwert '40404040') geändert wird, versucht CICS, die IMS-Anforderung auf dem lokalen System auszuführen.

## Exit XDLIPRE

Der Exit XDLIPRE wird beim Eintritt in das DL/I-Schnittstellenprogramm aufgerufen.

Programme, die in diesem Exit ausgeführt werden, müssen gemäß threadsicheren Standards codiert und für CICS als threadsicher definiert werden.

### Exitspezifische Parameter

#### UEPCTYPE

Adresse des Byte für den Anforderungstyp. Mögliche Werte:

#### UEPCEXEC

Die ursprüngliche Anforderung war eine EXEC-DLI-Anforderung.

#### UEPCCALL

Die ursprüngliche Anforderung war eine Anforderung auf CALL-Ebene.

#### UEPCSHIP

Die Anforderung wurde über Funktionen aus einer anderen Region übertragen. Wenn dieser Wert festgelegt ist, gelten bestimmte Einschränkungen für die Einstellung und Verwendung der übrigen Exitparameter (wie beschrieben).

#### UEPAPLIST

Adresse der Parameterliste der Anwendung. Das allgemeine Format für COBOL und die Assemblersprache sieht wie folgt aus:

```
plist address --> parm1 address --> parm1
                parm2 address --> parm2
                parm3 address --> parm3
                .....
                up to a maximum of 18 parameters
                excluding the optional parmcount.
```

Das allgemeine Format für PL/I sieht wie folgt aus:



```
plist address --> parm1 address --> parm1 (parmcount)
    parm2 address --> locator descriptor --> parm2
    parm3 address --> locator descriptor --> parm3
    .....
    up to a maximum of 18 parameters
```

Wenn UEPCTYPE nicht UEPCSHIP ist, kann Ihr Exitprogramm alle Parameter in der Anwendungsparameterliste ändern. Bei Anforderungen vom Typ UEPCSHIP hingegen kann Ihr Exitprogramm **keine** Parameter in der Liste ändern. Des Weiteren gilt bei UEPCSHIP-Anforderungen, dass UEPAPLIST auf eine Kopie der Parameterliste im vorstehenden Format verweist, die jedoch nur die ersten beiden Parameter (parm1 und parm2) enthält.

**Anmerkung:** Bei PL/I-Anwendungen kann parm1 einen Parameterzähler enthalten, muss aber nicht. Ihr Exitprogramm sollte dieses Feld überprüfen, bevor es verwendet wird.

#### UEPLANG

Adresse des Byte für die Programmsprache. Mögliche Werte:

##### UEPPLI

PL/I

##### UEPCBL

COBOL

##### UEPASM

Assemblersprache

Bei UEPCSHIP-Anforderungen ist die Sprache immer die Assemblersprache.

#### UEPIOAX

Adresse des Flag-Byte zur Angabe der Existenz des Ein-/Ausgabebereichs:

##### UEPIOA1

Ein-/Ausgabebereich ist vorhanden.

Bei UEPCSHIP-Anforderungen ist das Flag zur Angabe der Existenz des Ein-/Ausgabebereichs immer inaktiviert.

#### UEPIOA

Adresse des Ein-/Ausgabebereichs. Dies ist der Ein-/Ausgabebereich (IOAREA) der Anwendung oder - im Fall von EXEC-DLI - von DFHEDP. Der Inhalt des Ein-/Ausgabebereichs kann im Exit überschrieben werden: Der neue Inhalt wird verwendet, wenn die DL/I-Anforderung verarbeitet wird. Es sollte jedoch beachtet werden, dass sich Ein-/Ausgabebereiche im statischen Speicher eines Programms befinden können und in diesem Fall nicht überschrieben werden sollten.

Bei UEPCSHIP-Anforderungen ist UEPIOA immer null.

#### UEPPSBNX

Adresse des Flag-Byte zur Angabe der Existenz eines Programmspezifikationsblocks (PSB):

##### UEPPSB1

Ein PSB ist vorhanden.

**UEPPSBNM**

Adresse eines Bereichs mit dem PSB-Namen aus 8 Zeichen. Der Inhalt des Bereichs kann vom Exit überschrieben werden. Dies gilt für alle Anforderungstypen, einschließlich UEPCSHIP. Der neue Inhalt wird verwendet, wenn die DL/I-Anforderung verarbeitet wird.

**UEPSYSDX**

Adresse des Flag-Byte zur Angabe der Existenz einer System-ID (SYSID):

**UEPSYS1**

Eine SYSID ist vorhanden.

**UEPSYSID**

Adresse eines Bereichs mit dem SYSID-Namen aus 4 Zeichen. Der Inhalt des Bereichs kann vom Exit überschrieben werden. Dies gilt für alle Anforderungstypen, einschließlich UEPCSHIP. Der neue Inhalt wird verwendet, wenn die DL/I-Anforderung verarbeitet wird.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen.

**UERCBYP**

DL/I-Anforderung umgehen und zurückkehren.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

**Exit XDLIPOST**

Der Exit XDLIPOST wird beim Verlassen des DL/I-Schnittstellenprogramms aufgerufen.

Programme, die in diesem Exit ausgeführt werden, müssen gemäß threadsicheren Standards codiert und für CICS als threadsicher definiert werden.

**Exitspezifische Parameter****UEPCTYPE**

Adresse des Byte für den Anforderungstyp. Mögliche Werte:

**UEPCEXEC**

EXEC-DLI-Anforderung.

**UEPCCALL**

Anforderung auf CALL-Ebene.

**UEPCSHIP**

Die Anforderung wurde über Funktionen aus einer anderen Region übertragen. Wenn dieser Wert festgelegt ist, gelten bestimmte Einschränkungen für die Einstellung und Verwendung der übrigen Exitparameter (wie beschrieben).

**UEPAPLIST**

Adresse der Parameterliste der Anwendung. Das allgemeine Format für COBOL und die Assemblersprache sieht wie folgt aus:

```

plist address --> parm1 address --> parm1
                parm2 address --> parm2
                parm3 address --> parm3
                .....
                up to a maximum of 18 parameters
                excluding the optional parmcount.

```

Das allgemeine Format für PL/I sieht wie folgt aus:

```

plist address --> parm1 address --> parm1 (parmcount)
                parm2 address --> locator descriptor --> parm2
                parm3 address --> locator descriptor --> parm3
                .....
                up to a maximum of 18 parameters.

```

Wenn UEPCTYPE nicht UEPCSHIP ist, kann Ihr Exitprogramm alle Parameter in der Anwendungsparameterliste ändern. Bei Anforderungen vom Typ UEPCSHIP hingegen kann Ihr Exitprogramm **keine** Parameter in der Liste ändern. Des Weiteren gilt bei UEPCSHIP-Anforderungen, dass UEPAPLIST auf eine Kopie der Parameterliste im vorstehenden Format verweist, die jedoch nur die ersten beiden Parameter (parm1 und parm2) enthält. Siehe auch „Exits XDLIPRE und XDLIPOST des DL/I-Schnittstellenprogramms“ auf Seite 29.

**Anmerkung:** Bei PL/I-Anwendungen kann parm1 unter Umständen einen Parameterzähler enthalten, muss aber nicht. Ihr Exitprogramm sollte dieses Feld überprüfen, bevor es verwendet wird.

#### UEPLANG

Adresse des Byte für die Programmsprache. Mögliche Werte:

##### UEPPLI

PL/I

##### UEPCBL

COBOL

##### UEPASM

Assemblersprache

Bei UEPCSHIP-Anforderungen ist die Sprache immer die Assemblersprache.

#### UEPIOAX

Adresse des Flag-Byte zur Angabe der Existenz des Ein-/Ausgabebereichs:

##### UEPIOA1

Ein-/Ausgabebereich ist vorhanden.

Bei UEPCSHIP-Anforderungen ist das Flag zur Angabe der Existenz des Ein-/Ausgabebereichs immer inaktiviert.

#### UEPIOA

Adresse des Ein-/Ausgabebereichs. Dies ist der Ein-/Ausgabebereich (IOAREA) der Anwendung oder - im Fall von EXEC-DLI - von DFHEDP. Der Inhalt des Ein-/Ausgabebereichs kann im Exit überschrieben werden und wird im neuen Format an das Anwendungsprogramm zurückgegeben. Es sollte jedoch beachtet werden, dass sich der Ein-/Ausgabebereiche der Anwendung im statischen Speicher des Programms befinden könnte und in diesem Fall nicht überschrieben werden sollte.

Bei UEPCSHIP-Anforderungen ist UEPIOA immer null.

#### **UEPUIBX**

Adresse des Flag-Byte zur Angabe der Existenz eines Benutzerschnittstellenblocks (UIB):

#### **UEPUIB1**

Ein UIB ist vorhanden.

#### **UEPUIB**

Adresse des UIB, der von DFHUIB im Modul DFHDBCOP zugeordnet ist. Der Inhalt des UIB kann im Exit für alle Anforderungstypen (einschließlich UEPCSHIP) überschrieben werden. Der neue Inhalt wird entweder an die Anwendung zurückgegeben oder an die Region, von der aus die Anforderung über Funktionen übertragen wurde.

#### **Rückgabecodes**

##### **UERCNORM**

Verarbeitung fortsetzen.

##### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

#### **XPI-Aufrufe**

Es können alle verwendet werden.

## **Beispiel für Verwendung des globalen Benutzerexits XDLIPRE**

Mithilfe des globalen Benutzerexits XDLIPRE können Sie den Namen des Programmspezifikationsblocks (PSB) ändern, den das Anwendungsprogramm während der Ausführung terminiert hat. Darüber hinaus können Sie den Exit XDLIPRE auch dazu verwenden, die Identität der SYSID zu ändern. Auf diese Weise kann die Verarbeitung von einer SYSID, die nicht mehr verfügbar ist, an eine verfügbare SYSID umgeleitet werden.

Dieser Abschnitt enthält Informationen zur produktabhängigen Programmierschnittstelle.

Die folgenden Abbildungen zeigen ein Beispiel für den Exit XDLIPRE, das kopiert und geändert werden kann. Dieses Beispiel wird nur als Anleitung zur Verfügung gestellt. Informationen zur Programmierung von globalen Benutzerexits finden Sie unter DFHZNEPI TYPE=INITIAL—specifying the default routine.

```

*****
* This is an example for global user exit XDLIPRE *
* * *
* It is invoked before any DLI call being passed to *
* the remote or DBCTL processors. *
* * *
* A check is made for the presence of a PSB. *
* If not, a normal return is made *
* * *
* If the PSB is in a predefined table, it is changed to a *
* different value, and a normal return is made. *
* * *
* If not, set PSB name to blanks and normal return. *
* * *
* In all cases, a trace entry is written describing the action *
* taken, using TRACE-POINT 384 (hex '0180') *
* * *
*****
* * *
* The first few instructions set up the global user exit *
* environment, identify the user exit point, prepare for the use of *
* the exit programming interface, and copy in the definitions that *
* are to be used by the XPI function. *
* * *
*****
*
*          DFHUEXIT TYPE=EP,ID=XDLIPRE      PROVIDE DFHUEPAR PARAMETER
*                                           LIST AND LIST OF EXITID
*                                           EQUATES
*
*          DFHUEXIT TYPE=XPIENV             SET UP ENVIRONMENT FOR
*                                           EXIT PROGRAMMING INTERFACE
*                                           MUST BE ISSUED BEFORE ANY
*                                           XPI MACROS ARE ISSUED

```

Abbildung 1. Beispiel für Benutzerexit XDLIPRE zum Ändern von PSB-Namen (1/6)

```

*
*      COPY  DFHTRPTY              DEFINE PARAMETER LIST FOR
*                                  USE BY DFHTRPTX MACRO
*
*      COPY  DFHSMCMY              DEFINE PARAMETER LIST FOR
*                                  USE BY DFHSMCMX MACRO
*
*****
*The following DSECT maps a storage area to be used as work area *
*for the information in the TRACE entry.                          *
*****
*
DSA      DSECT                      DSECT FOR GETMAINED STORAGE
        USING DSA,R7
*
RETCODE  DS      F                  store return code
MESSAGEA DS      F                  message address for trace
MESSAGEL DS      F                  message length for trace
MESSAGE  DS      0CL37
OLDPSB   DS      CL8
MESS1    DS      CL21
NEWPSB   DS      CL8
*****
*The next instructions form the normal start of a global user *
*exit program, setting the program addressing mode to 31-bit, saving*
*the calling program's registers, establishing base addressing*
*and establishing the addressing of the user exit parameter list.  *
*****
*
DLIPR    CSECT
DLIPR    AMODE 31
*
*      SAVE (14,12)                SAVE CALLING PROGRAM'S RGSTRS
*
*      LR      R11,R15              SET UP USER EXIT PROGRAM'S
*      USING DLIPR,R11              BASE REGISTER
*
*      LR      R2,R1                SET UP ADDRESSING FOR USER
*      USING DFHUEPAR,R2            EXIT PARAMETER LIST -- USE
*                                  REGISTER 2 AS XPI CALLS USE
*                                  REGISTER 1
*
*****
*Before issuing an XPI function call, set up addressing to XPI *
*parameter list.                                                *
*****
*
*      L       R5,UEPXSTOR          SET UP ADDRESSING FOR XPI
*                                  PARAMETER LIST

```

Abbildung 2. Beispiel für Benutzerexit XDLIPRE zum Ändern von PSB-Namen (2/6)

```

*****
* Before issuing an XPI function call, you must ensure that register*
* 13 addresses the kernel stack.                                     *
*****
*
*           L      R13,UEPSTACK          ADDRESS KERNEL STACK
*
*****
* Issue a GETMAIN to get storage for work area                      *
*****
*
*           USING DFHSMC_ARG,R5          MAP PARAMETER LIST
*
*           DFHSMCX CALL,                X
*               CLEAR,                    X
*               IN,                        X
*               FUNCTION(GETMAIN),        X
*               GET_LENGTH(100),          X
*               STORAGE_CLASS(USER),     X
*               SUSPEND(NO),              X
*               OUT,                      X
*               ADDRESS((R7)),            X
*               RESPONSE(*),              X
*               REASON(*)
*
*****
* SET UP THE NORMAL RETURN CODE                                     *
*****
*
*           LA      R6,UERCNORM
*           ST      R6,RETCODE
*
*****
* See if a PSB exists                                             *
*****
*
*           L      R6,UEPPSBNX          PSB EXISTENCE FLAG
*           TM      0(R6),UEPPSB1       PSB EXISTS?
*           BO      PSBCALL              YES
*           MVC     MESSAGE,MESS3T      NO-MOVE MESSAGE TO DSA
*           B       TRACE
*
*****
* See if we want to change a PSB name                             *
*****
*
* PSBCALL EQU *
*           L      R6,UEPPSBNM          ADDRESS OF PASSED PSB NAME
*           LA      R8,PSBS              ADDRESS OF table of PSB pairs
*           CLC     0(8,R6),0(R8)       SAME?

```

Abbildung 3. Beispiel für Benutzerexit XDLIPRE zum Ändern von PSB-Namen (3/6)

```

        BE    FOUND                YES
        LA    R8,16(R8)            BUMP TO NEXT PAIR
        CLC   0(8,R6),0(R8)
        BE    FOUND
        LA    R8,16(R8)            BUMP TO NEXT PAIR
        CLC   0(8,R6),0(R8)
        BE    FOUND
        B     NOTFOUND             NO MATCH - END
*
*****
* Move new PSB name in *
*****
*
FOUND    EQU    *
        MVC    0(8,R6),8(R8)
*
*****
* SET UP MESSAGE BLOCK FOR TRACE ENTRY FOR CHANGED NAME *
*****
*
        MVC    MESS1,MESS1T        SET UP MESSAGE
        MVC    NEWPSB,8(R8)        NEW PSB NAME
        MVC    OLDPSB,0(R8)        OLD PSB NAME
        B     TRACE                GO PUT TRACE ENTRY
*
*****
* SET UP MESSAGE BLOCK FOR TRACE ENTRY FOR PSB NOT FOUND *
* SETUP THE NORMAL RETURN CODE *
*****
*
NOTFOUND EQU    *
        MVC    0(8,R6),DUMMYPST
        MVC    MESS1,MESS2T        SET UP MESSAGE
        MVC    OLDPSB,0(R6)        SUPPLIED PSB NAME
        MVC    NEWPSB,=CL8' '     CLEAR FIELD
        LA    R1,UERCNORM          SET UP NORMAL RETURN CODE
        B     TRACE                GO PUT TRACE ENTRY
*
*****
* Issue trace put macro *
*****
*
TRACE    EQU    *
        LA    R6,MESSAGE           STORE ADDRESS...
        ST    R6,MESSAGEA          ...INTO BLOCK DESCRIPTOR
        LA    R6,L'MESSAGE         STORE LENGTH...
        ST    R6,MESSAGEL          ...INTO BLOCK DESCRIPTOR
        LA    R8,384               SET UP TRACE-ID
*

```

Abbildung 4. Beispiel für Benutzerexit XDLIPRE zum Ändern von PSB-Namen (4/6)



```

DROP R5                      REUSE R5 TO MAP DFHTRPT
USING DFHTRPT_ARG,R5        XPI PARAMETER LIST

*
DFHTRPTX CALL,              X
    CLEAR,                  X
    IN,                     X
    FUNCTION(TRACE_PUT),    X
    POINT_ID((R8)),         X
    DATA1(MESSAGEA,MESSAGEL), X
    OUT,                    X
    RESPONSE(*)
*
*****
*When the rest of the exit program is complete, free the storage *
*and return.                                                    *
*****
*
DROP R5                      REUSE REGISTER 5 TO MAP DFHSMC
USING DFHSMC_ARG,R5        XPI PARAMETER LIST
*
*****
* Issue the DFHSMCX macro call                                  *
* Store the return code in register 6                            *
*****
*
L    R6,RETCODE              PICK UP SAVED RETURN CODE
*
DFHSMCX CALL,                X
    CLEAR,                  X
    IN,                     X
    FUNCTION(FREEMAIN),      X
    ADDRESS((R7)),           X
    STORAGE_CLASS(USER),     X
    OUT,                     X
    RESPONSE(*),             X
    REASON(*)
*
*****
*Restore registers, set return code, and return to user exit handler*
*****
*
L    R13,UEPEPSA
ST   R6,16(13)               STORE INTO R15 SLOT OF SA
RETURN (14,12)
*
*****
*old and new PSB names, in pairs                                *
*****
*

```

Abbildung 5. Beispiel für Benutzerexit XDLIPRE zum Ändern von PSB-Namen (5/6)

```

PSBS      EQU      *
           DC      CL8'PC3CONEW'          VALID
           DC      CL8'PC3CONE2'         VALID
           DC      CL8'PC3FRED'          INVALID
           DC      CL8'PC3CONEW'         VALID
           DC      CL8'PC3JOE'           INVALID
           DC      CL8'PC3JOEX'          INVALID

*
MESS1T    DC      CL21' HAS BEEN CHANGED TO '
MESS2T    DC      CL21' WAS NOT FOUND'
MESS3T    DC      CL37'THIS WAS NOT A DLI SCHEDULE CALL'
DUMMYPSB  DC      CL8' '
           LTORG
           END      DLIPR

```

Abbildung 6. Beispiel für Benutzerexit XDLPRE zum Ändern von PSB-Namen (6/6)

## Exits XDUREQ, XDUREQC, XDUCLE und XDOUT für die Speicherauszugsdomäne

Sie können die Exits für die Speicherauszugsdomäne verwenden, um Informationen vor und nach einem Transaktionsspeicherauszug oder einem Systemspeicherauszug zu erfassen.

### Exit XDUREQ

Der Exit XDUREQ wird unmittelbar vor Erstellung eines System- oder Transaktionsspeicherauszugs aufgerufen.

#### Exitspezifische Parameter

##### UEPTRANID

Adresse der 4-Byte-Transaktions-ID.

##### UEPUSER

Adresse der 8-Byte-Benutzer-ID.

##### UEPTERM

Adresse der 4-Byte-Terminal-ID.

##### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens oder Nullen, wenn keine aktuelle Anwendung vorhanden ist.

##### UEPDUMPC

Adresse der Kopie des 8-Byte-Speicherauszugscode.

##### UEPABCDE

Adresse einer Kopie des 8-Byte-Kernel-Fehlercodes im Format xxx/yyyy. Hierbei steht 'xxx' für den dreistelligen hexadezimalen MVS-Beendigungscode (beispielsweise 0C1 oder D37). Wenn kein MVS-Beendigungscode anwendbar ist, werden für 'xxx' drei Bindestriche angegeben. Der vierstellige Code 'yyyy' ist ein Benutzerabbruchcode, der entweder von CICS oder von einem anderen Produkt auf Ihrem System generiert wird. UEPABCDE wird nur für Abbruchcodes ausgefüllt, die den folgenden Speicherauszugscode entsprechen:

- AP0001
- SR0001
- ASRA

- ASRB
- ASRD

Andernfalls enthält dieses Feld Nullzeichen.

#### **UEPDUMPT**

Adresse der 1-Byte-ID des Speicherauszugstyps, die einen der folgenden Werte enthält:

#### **UEPDTRAN**

Es wurde ein Transaktionsspeicherauszug angefordert.

#### **UEPDSYST**

Es wurde ein Systemspeicherauszug angefordert.

**Anmerkung:** Die ID des Speicherauszugstyps gibt den Typ der Speicherauszugsanforderung an, die an die Speicherauszugsdomäne übergeben wurde. Sie spiegelt keine Änderungen wider, die möglicherweise durch einen Benutzereintrag in der Speicherauszugstabelle an der ursprünglichen Anforderung vorgenommen wurden.

#### **UEPXDSCP**

Adresse eines 1-Byte-Felds zur Angabe der aktuellen Einstellung von DUMPSCOPE für die Speicherauszugstabelle. Das Feld enthält einen der folgenden Werte:

#### **UEPXDLOC**

Ein Systemspeicherauszug wird nur für das lokale MVS-Image erstellt.

#### **UEPXDREL**

Systemspeicherauszüge werden sowohl für das lokale MVS-Image als auch für zugehörige MVS-Images innerhalb des Sysplex erstellt.

Dieses Feld kann durch den Exit geändert werden, um die Einstellung von DUMPSCOPE für die Speicherauszugstabelle zu aktualisieren.

#### **UEPXDTXN**

Adresse eines 1-Byte-Felds zur Angabe der aktuellen Einstellung von TRANDUMP für die Speicherauszugstabelle. Das Feld enthält einen der folgenden Werte:

#### **UEPXDYES**

Es wird ein Transaktionsspeicherauszug erstellt.

#### **UEPXDNO**

Es wird kein Transaktionsspeicherauszug erstellt.

Dieses Feld kann durch den Exit geändert werden, um die Einstellung von TRANDUMP für die Speicherauszugstabelle zu aktualisieren.

**Anmerkung:** Dieses Feld ist nur dann gültig, wenn UEPDUMPT den Wert UEPDTRAN enthält.

#### **UEPXDSYS**

Adresse eines 1-Byte-Felds zur Angabe der aktuellen Einstellung von SYSDUMP für die Speicherauszugstabelle. Das Feld enthält einen der folgenden Werte:

**UEPXDYES**

Es wird ein Systemspeicherauszug erstellt.

**UEPXDNO**

Es wird kein Systemspeicherauszug erstellt.

Dieses Feld kann durch den Exit geändert werden, um die Einstellung von SYSDUMP für die Speicherauszugstabelle zu aktualisieren.

**UEPXDTRM**

Adresse eines 1-Byte-Felds zur Angabe der aktuellen Einstellung von SHUTDOWN für die Speicherauszugstabelle. Das Feld enthält einen der folgenden Werte:

**UEPXDYES**

Das CICS-System soll beendet werden.

**UEPXDNO**

Das CICS-System soll nicht beendet werden.

Dieses Feld kann durch den Exit geändert werden, um die Einstellung von SHUTDOWN für die Speicherauszugstabelle zu aktualisieren.

**UEPXDMAX**

Adresse eines 4-Byte-Felds mit der aktuellen Einstellung von MAXIMUM für die Speicherauszugstabelle. Dieses Feld kann durch den Exit geändert werden, um die aktuelle Einstellung von MAXIMUM für die Speicherauszugstabelle zu ändern. Eine Änderung an der Einstellung von MAXIMUM führt nicht zur Unterdrückung dieser Speicherauszugsanforderung. Zur Unterdrückung der aktuellen Speicherauszugsanforderung kann ein Rückgabecode vom Typ UERCBYP verwendet werden.

**UEPDXDCNT**

Adresse eines 4-Byte-Felds mit der aktuellen Einstellung von CURRENT für die Speicherauszugstabelle.

**UEPXDST**

Adresse eines 16-Byte-Felds mit den aktuellen Statistikdaten zur Speicherauszugstabelle für diesen Speicherauszugscode. Das adressierte Feld besteht aus vier 4-Byte-Feldern, die folgende binäre Ganzzahlen enthalten:

- Anzahl der erstellten Transaktionsspeicherauszüge.
- Anzahl der unterdrückten Transaktionsspeicherauszüge.
- Anzahl der erstellten Systemspeicherauszüge.
- Anzahl der unterdrückten Systemspeicherauszüge.

**Anmerkung:** Statistiken für Transaktionsspeicherauszüge sind nur dann gültig, wenn UEPDUMPT den Wert UEPDTRAN enthält.

**UEPXDDAE**

Adresse eines 1-Byte-Felds mit der aktuellen Einstellung von DAE-OPTION für die Speicherauszugstabelle. Das Feld enthält einen der folgenden Werte:

**UEPXDYES**

Der Speicherauszug kommt für DAE-Unterdrückung infrage.

#### UEPXDNO

Der Speicherauszug wird von DAE nicht unterdrückt.

Dieses Feld kann durch den Exit geändert werden, um die Einstellung von DAEPTION für die Speicherauszugstabelle zu aktualisieren.

#### UEPDMPID

Adresse eines Felds aus 9 Zeichen im Format xxxx/xxxx mit der Speicherauszugs-ID. Die Speicherauszugs-ID stimmt mit derjenigen überein, die von der entsprechenden Speicherauszugsnachricht ausgegeben wird.

#### UEPFMOD

Adresse eines 8-Byte-Bereichs mit dem Namen des fehlerhaften Moduls, wenn der Speicherauszugscode AP0001 ist; andernfalls enthält der Bereich Nullzeichen.

Hierbei ist zu beachten, dass das Feld UEPPROG immer den Namen der *aktuellen* Anwendung adressiert, unabhängig davon, wo der Fehler aufgetreten ist. UEPFMOD adressiert den Namen des Moduls, in dem der Fehler aufgetreten ist, sofern bekannt.

Wenn der Speicherauszugscode AP0001 lautet, gibt es drei Möglichkeiten:

1. Das von UEPFMOD adressierte Feld enthält denselben Namen wie das vom UEPPROG adressierte Feld. Dies bedeutet, dass der Fehler im Anwendungscode aufgetreten ist.
2. Das von UEPFMOD adressierte Feld enthält einen anderen Namen als das von UEPPROG adressierte Feld. Dies bedeutet, dass der Fehler außerhalb des Anwendungscodes aufgetreten ist.
3. Das von UEPFMOD adressierte Feld enthält '????????'. Dies bedeutet, dass der Fehler nicht im Anwendungscode aufgetreten ist; CICS war jedoch nicht in der Lage, den Namen des fehlerhaften Moduls zu ermitteln.

#### UEPDLISI

Adresse des 4-Byte-Werts für DSPLIST.

#### UEPJLISI

Adresse des 4-Byte-Werts für JOBLIST.

#### Rückgabecodes

##### UERCNORM

Verarbeitung fortsetzen.

##### UERCBYP

Speicherauszug unterdrücken.

##### UERCPURG

Task während des XPI-Aufrufs bereinigt.

#### XPI-Aufrufe

WAIT\_MVS kann **nur dann** verwendet werden, wenn der Wert von UEPDUMPT angibt, dass ein Transaktionsspeicherauszug erstellt wird. **Verwenden Sie keine anderen Aufrufe.**

## Exit XDUREQC

Der Exit XDUREQC wird unmittelbar nach dem Erstellen, nach dem Fehlschlagen oder nach der Unterdrückung eines System- oder Transaktionsspeicherauszugs aufgerufen.

### Exitspezifische Parameter

#### UEPTRANID

Adresse der 4-Byte-Transaktions-ID.

#### UEPUSER

Adresse der 8-Byte-Benutzer-ID.

#### UEPTERM

Adresse der 4-Byte-Terminal-ID.

#### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens.

#### UEPDUMPC

Adresse der Kopie des 8-Byte-Speicherauszugscode.

#### UEPABCDE

Adresse einer Kopie des 8-Byte-Kernel-Fehlercodes im Format xxx/yyyy. Hierbei steht 'xxx' für den dreistelligen hexadezimalen MVS-Beendigungscode (beispielsweise X'0C1' oder X'D37'). Wenn kein MVS-Beendigungscode anwendbar ist, werden für 'xxx' drei Bindestriche angegeben. Der vierstellige Code 'yyyy' ist ein Benutzerabbruchcode, der entweder von CICS oder von einem anderen Produkt auf Ihrem System generiert wird. UEPABCDE wird nur für Abbruchcodes ausgefüllt, die den folgenden Speicherauszugscode entsprechen:

- AP0001
- SR0001
- ASRA
- ASRB
- ASRD

Andernfalls enthält dieses Feld Nullzeichen.

#### UEPDUMPT

Adresse der 1-Byte-ID des Speicherauszugstyps, die einen der folgenden Werte enthält:

#### UEPDTRAN

Es wurde ein Transaktionsspeicherauszug angefordert.

#### UEPDSYST

Es wurde ein Systemspeicherauszug angefordert.

**Anmerkung:** Die ID des Speicherauszugstyps gibt den Typ der Speicherauszugsanforderung an, die an die Speicherauszugsdomäne übergeben wurde. Sie spiegelt keine Änderungen wider, die möglicherweise durch einen Benutzereintrag in der Speicherauszugstabelle an der ursprünglichen Anforderung vorgenommen wurden.

#### **UEPXDSCP**

Adresse eines 1-Byte-Felds zur Angabe der aktuellen Einstellung von DUMPSCOPE für die Speicherauszugstabelle. Das Feld enthält einen der folgenden Werte:

##### **UEPXDLOC**

Ein Systemspeicherauszug wird nur für das lokale MVS-Image erstellt.

##### **UEPXDREL**

Systemspeicherauszüge werden sowohl für das lokale MVS-Image als auch für zugehörige MVS-Images innerhalb des Sysplex erstellt.

Dieses Feld kann durch den Exit geändert werden, um die Einstellung von DUMPSCOPE für die Speicherauszugstabelle zu aktualisieren.

#### **UEPXDTXN**

Adresse eines 1-Byte-Felds zur Angabe der aktuellen Einstellung von TRANDUMP für die Speicherauszugstabelle. Das Feld enthält einen der folgenden Werte:

##### **UEPXDYES**

Es wird ein Transaktionsspeicherauszug erstellt.

##### **UEPXDNO**

Es wird kein Transaktionsspeicherauszug erstellt.

Dieses Feld kann durch den Exit geändert werden, um die Einstellung von TRANDUMP für die Speicherauszugstabelle zu aktualisieren.

**Anmerkung:** Dieses Feld ist nur dann gültig, wenn UEPDUMPT den Wert UEPDTRAN enthält.

#### **UEPXDSYS**

Adresse eines 1-Byte-Felds zur Angabe der aktuellen Einstellung von SYSDUMP für die Speicherauszugstabelle. Das Feld enthält einen der folgenden Werte:

##### **UEPXDYES**

Es wird ein Systemspeicherauszug erstellt.

##### **UEPXDNO**

Es wird kein Systemspeicherauszug erstellt.

Dieses Feld kann durch den Exit geändert werden, um die Einstellung von SYSDUMP für die Speicherauszugstabelle zu aktualisieren.

#### **UEPXDTRM**

Adresse eines 1-Byte-Felds zur Angabe der aktuellen Einstellung von SHUTDOWN für die Speicherauszugstabelle. Das Feld enthält einen der folgenden Werte:

##### **UEPXDYES**

Das CICS-System soll beendet werden.

##### **UEPXDNO**

Das CICS-System soll nicht beendet werden.

Dieses Feld kann durch den Exit geändert werden, um die Einstellung von SHUTDOWN für die Speicherauszugstabelle zu aktualisieren.

#### **UEPXDMAX**

Adresse eines 4-Byte-Felds mit der aktuellen Einstellung von MAXIMUM für die Speicherauszugstabelle. Dieses Feld kann durch den Exit geändert werden, um die aktuelle Einstellung von MAXIMUM für die Speicherauszugstabelle zu ändern.

#### **UEPDXDCNT**

Adresse eines 4-Byte-Felds mit der aktuellen Einstellung von CURRENT für die Speicherauszugstabelle.

#### **UEPXDSTST**

Adresse eines 16-Byte-Felds mit den aktuellen Statistikdaten zur Speicherauszugstabelle für diesen Speicherauszugscode. Das adressierte Feld besteht aus vier 4-Byte-Feldern, die folgende binäre Ganzzahlen enthalten:

- Anzahl der erstellten Transaktionsspeicherauszüge.
- Anzahl der unterdrückten Transaktionsspeicherauszüge.
- Anzahl der erstellten Systemspeicherauszüge.
- Anzahl der unterdrückten Systemspeicherauszüge.

**Anmerkung:** Statistiken für Transaktionsspeicherauszüge sind nur dann gültig, wenn UEPDUMPT den Wert UEPDTRAN enthält.

#### **UEPXDDAE**

Adresse eines 1-Byte-Felds mit der aktuellen Einstellung von DAE-  
OPTION für die Speicherauszugstabelle. Das Feld enthält einen der folgenden Werte:

##### **UEPXDYES**

Der Speicherauszug wurde von DAE unterdrückt.

##### **UEPXDNO**

Der Speicherauszug wurde von DAE nicht unterdrückt.

Dieses Feld kann durch den Exit geändert werden, um die Einstellung von DAEOPTION für die Speicherauszugstabelle zu aktualisieren.

#### **UEPDMPID**

Adresse eines Felds aus 9 Zeichen im Format xxxx/xxxx mit der Speicherauszugs-ID. Die Speicherauszugs-ID stimmt mit derjenigen überein, die von der entsprechenden Speicherauszugsnachricht ausgegeben wird.

#### **UEPDRESP**

Adresse des 2-Byte-Antwortcodes für den Speicherauszug.

#### **UEPDREAS**

Adresse des 2-Byte-Ursachencodes für den Speicherauszug.

#### **UEPDLISO**

Adresse des 4-Byte-Werts für DSPLIST.

#### **UEPJLISO**

Adresse des 4-Byte-Werts für JOBLIST.

### **Rückgabecodes**



#### **UERCNORM**

Verarbeitung fortsetzen.

#### **XPI-Aufrufe**

WAIT\_MVS kann nur dann verwendet werden, wenn der Wert von UEPDUMPT angibt, dass ein Transaktionsspeicherauszug erstellt wird. Verwenden Sie keine anderen Aufrufe.

### **Exit XDUCLSE**

Dieser Exit wird unmittelbar nach dem Schließen einer Transaktionsspeicherauszugsdatei aufgerufen.

#### **Zeitpunkt des Aufrufs**

Unmittelbar nach dem Schließen einer Transaktionsspeicherauszugsdatei.

#### **Exitspezifische Parameter**

##### **UEPTRANID**

Adresse der 4-Byte-Transaktions-ID.

##### **UEPUSER**

Adresse der 8-Byte-Benutzer-ID.

##### **UEPTERM**

Adresse der 4-Byte-Terminal-ID.

##### **UEPPROG**

Adresse des 8-Byte-Anwendungsprogrammnamens.

##### **UEPDMPDD**

Adresse des 8-Byte-DD-Namens der Speicherauszugsdatei.

##### **UEPDMPDSN**

Adresse des 44-Byte-Dateinamens der Speicherauszugsdatei.

#### **Rückgabecodes**

##### **UERCNORM**

Verarbeitung fortsetzen.

##### **UERCSWCH**

Das Flag für automatisches Wechseln ist aktiviert.

#### **XPI-Aufrufe**

WAIT\_MVS kann verwendet werden. **Verwenden Sie keine anderen Aufrufe.**

### **Exit XDUOUT**

Dieser Exit wird aufgerufen, bevor ein Datensatz in die Transaktionsspeicherauszugsdatei geschrieben wird.

#### **Zeitpunkt des Aufrufs**

Bevor ein Datensatz in die Transaktionsspeicherauszugsdatei geschrieben wird.

#### **Exitspezifische Parameter**

##### **UEPTRANID**

Adresse der 4-Byte-Transaktions-ID.

##### **UEPUSER**

Adresse der 8-Byte-Benutzer-ID.

**UEPTERM**

Adresse der 4-Byte-Terminal-ID.

**UEPPROG**

Adresse des 8-Byte-Anwendungsprogrammnamens.

**UEPDMPFC**

Adresse des 1-Byte-Funktionscodes. Der Code wird durch folgende Werte entsprechend angegeben:

**UEPDMPWR**

Der Puffer ist im Begriff, geschrieben zu werden.

**UEPDMPRE**

Der Speicherauszug ist im Begriff, nach dem automatischen Umschalten erneut gestartet zu werden.

**UEPDMPAB**

Abbruch des Speicherauszugs.

**UEPDMPDY**

Der Puffer ist im Begriff, geschrieben zu werden, und die CICS-Speicherauszugsdatei ist entweder eine Pseudodatei oder geschlossen.

**UEPDMPBF**

Adresse des Speicherauszugspuffers, dessen Länge durch den Parameter UEPDMLEN adressiert wird.

**UEPDMPLEN**

Adresse der 2 Byte-Länge des Speicherauszugspuffers.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen.

**UERCBYP**

Ausgabe von Speicherauszugssätzen unterdrücken.

**XPI-Aufrufe**

WAIT\_MVS kann verwendet werden. **Verwenden Sie keine anderen Aufrufe.**

---

## Exits XNQEREQ und XNQEREQC des Einreihungsprogramms der EXEC-Schnittstelle

Mit dem Exit XNQEREQ können Sie API-Anforderungen für Einreihungen in die Warteschlange abfangen, bevor entsprechende Aktionen für diese Anforderungen ausgeführt werden. Mithilfe des Exits XNQEREQC können Sie die entsprechende Antwort abfangen, nachdem eine API-Einreihungsanforderung abgeschlossen worden ist.

Dies gilt für die folgenden API-Anforderungen:

- EXEC CICS ENQ
- EXEC CICS DEQ

**XNQEREQ**

Mithilfe von XNQEREQ können Sie folgende Aktionen ausführen:

- Analysieren der API-Parameterliste (Funktion, Schlüsselwörter, Argumentwerte und Antworten).

- Ändern beliebiger Eingabeparameterwerte vor Ausführung einer Anforderung.
- Verhindern der Ausführung einer Anforderung. Dies ermöglicht es Ihnen, die CICS-Funktion durch eine eigene Verarbeitung zu ersetzen.

#### **Hinweise zur Verwendung von XNQEREQ zum Ändern des Geltungsbereichs von ENQ bzw. DEQ:**

1. XNQEREQ ermöglicht es Ihnen, vorhandene Anwendungen so zu konvertieren, dass sie Sysplex-Einreihungen (Enqueues) verwenden, ohne dass die betreffenden Anwendungen geändert werden müssen.

**Anmerkung:** Bei Verwendung der ENQMODEL-Ressourcendefinition oder des Benutzerexits ist dies in den meisten Fällen möglich. Wird bei Anwendungen hingegen der Ressourcenname dynamisch festgelegt, ohne vorher bekannt zu sein, dann können diese Anwendungen nur mithilfe dieses Exits konvertiert werden.

2. Für Einreihungen, die in einem Sysplex gültig sind, wird ein anderer Namensbereich verwendet als für Einreihungen, die in einer Region gültig sind. Eine Einreihung mit Regionsgültigkeit wartet niemals auf eine Einreihung mit Sysplexgültigkeit und umgekehrt.

**Anmerkung:** Diese Situation kann nur dann eintreten, wenn Sie den Exit verwenden. Durch Verwendung der ENQMODEL-Ressourcendefinitionen als einzige Methode zur Definition des Geltungsbereichs (SCOPE) von ENQ- bzw. DEQ-Anforderungen lässt sich dieses potenzielle Risiko vermeiden.

3. Für ENQ-Anforderungen vom Typ Zeichenfolge werden beide Gültigkeitsbereiche (Region und Sysplex) unterstützt. Für ENQ-Anforderungen vom Typ Adresse hingegen wird Sysplex nicht als Gültigkeitsbereich unterstützt.

#### **XNQEREQC**

Mithilfe von XNQEREQC können Sie folgende Aktionen ausführen:

- Analysieren der API-Parameterliste.
- Übergeben von Daten zwischen den Exitprogrammen für XNQEREQ und XNQEREQC, wenn diese Programme für dieselbe Anforderung aufgerufen werden.
- Übergeben von Daten zwischen den Exitprogrammen für Einreihungen, wenn diese Programme innerhalb derselben Task aufgerufen werden.

CICS stellt ein Beispiexitprogramm namens DFH\$XNQE für die EXEC-Schnittstelle für Einreihungen bereit. Weitere Informationen finden Sie unter Enqueue EXEC interface sample exit program: DFH\$XNQE.

### **Exit XNQEREQ**

Dieser Exit wird aufgerufen, bevor CICS eine Anforderung vom Typ EXEC CICS ENQ oder EXEC CICS DEQ verarbeitet oder versucht, eine solche Anforderung mit einer installierten ENQMODEL-Ressourcendefinition abzugleichen.

#### **Zeitpunkt des Aufrufs**

Bevor CICS eine Anforderung vom Typ EXEC CICS ENQ oder EXEC CICS DEQ verarbeitet oder versucht, eine solche Anforderung mit einer installierten ENQMODEL-Ressourcendefinition abzugleichen.

#### **Exitspezifische Parameter**

**UEPCLPS**

Adresse einer Kopie der Befehlsparameterliste. Siehe „Parameterstruktur der Befehlsebene“ auf Seite 52.

**UEPNQTOK**

Adresse eines 4-Byte-Bereichs, der verwendet werden kann, um für eine einzelne Einreihungsanforderung (Enqueue) Informationen zwischen XNQEREQ und XNQEREQC zu übergeben.

**UEPRCODE**

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

**UEPRES**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

**UEPRES2**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

**UEPTSTOK**

Adresse eines 4-Byte-Tokens, das verwendet werden kann, um innerhalb derselben Task Informationen zwischen aufeinanderfolgenden Einreihungsanforderungen (Enqueue) zu übergeben (beispielsweise zwischen aufeinanderfolgenden Aufrufen des Exits XNQEREQ).

**UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

**UEPSCOPE**

Adresse des zu verwendenden 4-Byte-Namens für ENQSCOPE.

**Rückgabecodes****UERCBYP**

Diese Anforderung umgehen.

**UERCNORM**

Verarbeitung fortsetzen.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**UERCSCPE**

Ein ENQSCOPE-Name wurde angegeben.

**XPI-Aufrufe**

Es können alle verwendet werden.

**API- und SPI-Befehle**

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

EXEC CICS SHUTDOWN  
EXEC CICS XCTL

**Anmerkung:** Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor. So müssen Sie beispielsweise den Eintritt in eine Schleife vermeiden, wenn Sie eine Einreihungsanforderung über den Exit XNQEREQ ausgeben. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.

## Exit XNQEREQC

Der Exit XNQEREQC wird nach Abschluss einer API-Anforderung zur Einreihung in eine Warteschlange (Enqueue) und vor der Rückkehr vom Einreihungsprogramm der EXEC-Schnittstelle aufgerufen.

### Exitspezifische Parameter

#### UEPCLPS

Adresse einer Kopie der Befehlsparameterliste. Siehe „Parameterstruktur der Befehlsebene“ auf Seite 52.

#### UEPNQTOK

Adresse eines 4-Byte-Bereichs, der verwendet werden kann, um für eine einzelne Einreihungsanforderung (Enqueue) Informationen zwischen XNQEREQ und XNQEREQC zu übergeben.

#### UEPRCODE

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

#### UEPRES P

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

#### UEPRES P2

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

#### UEPTSTOK

Adresse eines 4-Byte-Tokens, das verwendet werden kann, um innerhalb derselben Task Informationen zwischen aufeinanderfolgenden Einreihungsanforderungen (Enqueue) zu übergeben (beispielsweise zwischen aufeinanderfolgenden Aufrufen des Exits XNQEREQC).

#### UEPRECUR

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

#### UEPSCOPE

Adresse des verwendeten 4-Byte-Namens für ENQSCOPE.

### Rückgabecodes

#### UERCNORM

Verarbeitung fortsetzen.

#### UERCPURG

Task während des XPI-Aufrufs bereinigt.

### XPI-Aufrufe

Es können alle verwendet werden.

### API- und SPI-Befehle

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

EXEC CICS SHUTDOWN  
EXEC CICS XCTL

Sie können die Kopien von EIBRCODE, EIBRESP und EIBRESP2, die Sie in der Parameterliste erhalten, aktualisieren. Wenn Sie die Werte aktualisieren, kopiert CICS

die neuen Werte in den EIB des Anwendungsprogramms, nachdem XNQEREQC beendet wurde oder wenn Sie den Rückgabecode UERCBYP in XNQEREQ angeben.

Sie müssen gültige Antworten für Einreihungen festlegen. Sie müssen für alle drei Codes (EIBRCODE, EIBRESP und EIBRESP2) eine konsistente Gruppe von Werten festlegen, so wie sie auch von der Enqueue-Domäne festgelegt würden, um eine gültige Ausführung zu beschreiben. Die Konsistenz der Werte für EIBRCODE, EIBRESP und EIBRESP2 wird von CICS nicht überprüft. Wenn EIBRCODE auf einen Wert ungleich null gesetzt wird und EIBRESP auf null gesetzt wird, überschreibt CICS diesen Wert für EIBRESP durch einen Wert, der ungleich null ist. Als Hilfestellung beim Festlegen von Werten für EIBRCODE, EIBRESP und EIBRESP2 können Sie die im Pseudobereich (DSECT) DFHNQUED angegebenen Werte heranziehen, die von der Enqueue-Domäne verwendet werden.

**Anmerkung:** Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor, um keine Schleife zu verursachen. So sind Sie beispielsweise dafür verantwortlich, das Auftreten einer Schleife zu vermeiden, wenn eine Einreihungsanforderung über den Exit XNQEREQC ausgegeben wird. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.

## Parameterstruktur der Befehlsebene

Die Parameterstruktur der Befehlsebene besteht aus einer Reihe von Adressen. Die erste Adresse verweist auf den EXEC-Schnittstellendeskriptor (EID), der aus einer Bitfolge besteht, die den Typ der Anforderung beschreibt und jedes mit der Anforderung angegebene Schlüsselwort identifiziert. Die übrigen Adressen verweisen auf Teile von Daten, die der Anforderung zugeordnet sind.

### Indikator für Ende der Parameterliste

Anhand des EXEC-Schnittstellendeskriptors können Sie den Typ der Anforderung sowie die angegebenen Schlüsselwörter ermitteln. Anhand der anderen Parameter in der Liste können Sie die Werte der Schlüsselwörter ermitteln. Sie können auch Werte von Schlüsselwörtern ändern, die in der Anforderung angegeben sind.

Das höchstwertige Bit wird für die letzte Adresse in der Parameterliste gesetzt, um anzuzeigen, dass es sich um die letzte Adresse in der Liste handelt. Bei der Rückkehr vom Benutzerexitprogramm sucht CICS in der Parameterliste nach dem höchstwertigen Bit, um den letzten Parameter ausfindig zu machen. Daher gilt: Wenn Sie die Länge der Parameterliste ändern, müssen Sie auch das höchstwertige Bit neu setzen, um so die aktuelle letzte Adresse zu markieren.

### Exitspezifischer Parameter UEPCLPS

Der exitspezifische Parameter UEPCLPS wird sowohl in den Exit XNQEREQ als auch in den Exit XNQEREQC eingeschlossen; er enthält die Adresse der Parameterstruktur der Befehlsebene.

Die Parameterstruktur der Befehlsebene enthält vier Adressen: NQ\_ADDR0 bis NQ\_ADDR3. Die Struktur wird im Pseudobereich (DSECT) NQ\_ADDR\_LIST definiert, den Sie in Ihr Exitprogramm kopieren sollten, indem Sie die Anweisung COPY DFHNQUED einschließen.

Die Parameterliste der Befehlsebene ist wie folgt aufgebaut:

## **NQ\_ADDR0**

Adresse eines 9-Byte-Bereichs, der als EID (EXEC Interface Descriptor) bezeichnet wird und wie folgt aufgebaut ist:

- **NQ\_GROUP**
- **NQ\_FUNCT**
- **NQ\_BITS1**
- **NQ\_BITS2**
- **NQ\_EIDOPT5**
- **NQ\_EIDOPT6**
- **NQ\_EIDOPT7**
- **NQ\_EIDOPT8**

## **NQ\_GROUP**

Immer X'12'; hierdurch wird angegeben, dass es sich um eine Tasksteueranforderung handelt.

## **NQ\_FUNCT**

Ein Byte, das den Typ der Anforderung definiert:

X'04' ENQ

X'06' DEQ

## **NQ\_BITS1**

Existenzbits zum Definieren der Argumente, die angegeben wurden. Um das Argument abzurufen, das einem Schlüsselwort zugeordnet ist, müssen Sie die entsprechende Adresse aus der Parameterstruktur der Befehlsebene verwenden. Bevor Sie diese Adresse verwenden, müssen Sie das zugehörige Existenzbit überprüfen. Wenn das Existenzbit inaktiviert ist, wurde das Argument in der Anforderung nicht angegeben, und die Adresse sollte nicht verwendet werden.

X'80' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort RESOURCE enthält. Ist dieses Existenzbit gesetzt, dann ist **NQ\_ADDR1** aussagekräftig.

X'40' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort LENGTH enthält. Ist dieses Existenzbit gesetzt, dann ist **NQ\_ADDR2** aussagekräftig.

X'20' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort MAXLIFETIME enthält. Ist dieses Existenzbit gesetzt, dann ist **NQ\_ADDR3** aussagekräftig.

## **NQ\_BITS2**

Zwei Byte, die nicht von der Enqueue-Domäne verwendet werden.

## **NQ\_EIDOPT5**

Ein Byte, das nicht von der Enqueue-Domäne verwendet wird.

## **NQ\_EIDOPT6**

Ein Byte, das nicht von der Enqueue-Domäne verwendet wird.

## **NQ\_EIDOPT7**

Ein Byte, das nicht von der Enqueue-Domäne verwendet wird.

## **NQ\_EIDOPT8**

Gibt an, ob bestimmte Schlüsselwörter in der Anforderung angegeben wurden.

X'04' NOSUSPEND wurde angegeben.

X'02' DEQ wurde angegeben.

X'01' ENQ wurde angegeben.

**NQ\_ADDR1**

Adresse eines Bereichs, der den Wert von RESOURCE enthält.

**NQ\_ADDR2**

Adresse des Halbwortwerts für LENGTH.

**NQ\_ADDR3**

Adresse des Vollwortwerts für MAXLIFETIME.

Die Beziehungen zwischen Argumenten, Schlüsselwörtern, Datentypen und den Typen 'Eingabe/Ausgabe' werden in Tabelle 2 auf Seite 55 für die Einreihungsbefehle (Enqueue-Befehle) zusammengefasst.

## **Ändern von Feldern in der Parameterstruktur der Befehlsebene**

Die Felder, die an die Enqueue-Domäne für Einreihungen übergeben werden, werden als Eingabe für die Anforderung verwendet. Die richtige Methode zum Ändern eines Eingabefelds besteht darin, eine neue Kopie desselben zu erstellen und die Adresse in der Parameterliste der Befehlsebene so zu ändern, dass sie auf die neuen Daten verweist.

**Important:**

1. Ändern Sie ein Eingabefeld nicht durch Ändern der Daten, auf die von der Parameterliste der Befehlsebene verwiesen wird. Dies würde zu einer Beschädigung von Speicher führen, der zum Anwendungsprogramm gehört; dies wiederum würde zu einem Fehler führen, wenn das Programm versucht, das Feld wiederzuverwenden.
2. In den EXEC CICS-Anforderungen vom Typ ENQ und DEQ gibt es keine Ausgabefelder.

## **Ändern des EXEC-Schnittstellendeskriptors (EID)**

Es ist nicht möglich, den EXEC-Schnittstellendeskriptor (EXEC Interface Descriptor, EID) zu ändern, um bedeutende Änderungen an Anforderungen vorzunehmen. So ist es beispielsweise nicht möglich, eine Anforderung vom Typ ENQ in eine Anforderung vom Typ DEQ zu ändern. Es können jedoch geringfügige Änderungen an Anforderungen vorgenommen werden. So ist es beispielsweise möglich, das Existenzbit für LENGTH zu aktivieren.

In der nachstehenden Liste werden diejenigen Bits im EID aufgeführt, die geändert werden können. Versuche, andere Bestandteile des EID zu ändern, werden ignoriert.

**NQ\_BITS1**

X'40' Existenzbit für LENGTH.

X'20' Existenzbit für MAXLIFETIME.

**NQ\_EIDOPT7**

Benutzerexitprogramme am Exit XNQEREQ können für ENQ-Befehle die folgenden Existenzbits aktivieren bzw. inaktivieren:

X'04' Existenzbit für NOSUSPEND.



Der EID wird vor der Rückkehr zum Anwendungsprogramm auf seinen ursprünglichen Wert zurückgesetzt. Dies bedeutet, dass Änderungen, die am EID vorgenommen werden, nur für die Dauer der Einreichungsanforderung (ENQ) beibehalten werden.

**Anmerkung:** Das Benutzerexitprogramm wird zwar daran gehindert, bedeutende Änderungen am EID vorzunehmen. Dennoch müssen Sie auch bei den geringfügigen Änderungen, die **zulässig sind**, sehr sorgfältig vorgehen.

## Verwendung des Task-Tokens UEPTSTOK

Das Task-Token UEPTSTOK stellt die Adresse eines 4-Byte-Bereichs bereit, mit dem Sie Informationen zwischen aufeinanderfolgenden Einreichungsanforderungen (Enqueue) in derselben Task übergeben können. So können Sie das Task-Token beispielsweise verwenden, um Informationen zwischen aufeinanderfolgenden Aufrufen des Exits XNQREQ zu übergeben.

Im Gegensatz dazu kann das Token UEPNQTOK nur für die Dauer einer einzelnen Einreichungsanforderung verwendet werden, da sein Inhalt am Ende der Anforderung gelöscht werden kann.

### Anmerkung:

1. Die Lebensdauer des Bereichs, auf den UEPTSTOK verweist, ist die Lebensdauer der Task.
2. Der Wert von UEPTSTOK wird von all den Exits gemeinsam genutzt, an die er während der Laufzeit der Task übergeben wird.

*Tabelle 2. Benutzerargumente und zugehörige Schlüsselwörter, Datentypen und Typen 'Eingabe/Ausgabe'*

Argument	Schlüsselwort	Datentyp	Eingabe/Ausgabe
Arg1	RESOURCE	DATA-AREA	Eingabe
Arg2	LENGTH	BIN(15)	Eingabe
Arg3	MAXLIFETIME	CVDA	Eingabe

## Ändern von Benutzerargumenten

Benutzerexitprogramme können Benutzereingabeargumente ändern, indem sie entweder Speicher abrufen bzw. festlegen oder einen Verweis setzen.

Benutzerexitprogramme können Benutzereingabeargumente wie folgt ändern:

1. Durch Abrufen von ausreichendem Speicher zum Speichern des geänderten Arguments.
2. Durch Festlegen des Speichers auf den erforderlichen Wert.
3. Durch Setzen des zugeordneten Verweises in der Parameterliste auf die Adresse des neu abgerufenen Bereichs.

### Anmerkung:

1. Änderungen an Argumentwerten werden von CICS nicht überprüft. Sämtliche Änderungen müssen demzufolge von dem Benutzerexitprogramm verifiziert werden, das die Änderungen vornimmt.
2. Das Ändern von Eingabeargumenten durch XNQREQC ist nicht empfehlenswert.

## Hinzufügen von Benutzerargumenten

Globale Benutzerexitprogramme können Argumente hinzufügen, die den Schlüsselwörtern LENGTH und MAXLIFETIME zugeordnet sind. Sie müssen sicherstellen, dass die Argumente, die Sie in Ihren Exitprogrammen angeben oder ändern, gültig sind.

Gültige Werte für MAXLIFETIME sind DFHVALUE(TASK) und DFHVALUE(UOW); dies entspricht 233 bzw. 246.

Unter der Voraussetzung, dass das hinzuzufügende Argument bisher nicht vorhanden ist, muss das Exitprogramm folgende Aktionen ausführen:

1. Abrufen von Speicher für das hinzuzufügende Argument.
2. Initialisieren des Speichers anhand des erforderlichen Werts.
3. Auswählen und Einstellen des entsprechenden Verweises in der Parameterliste.
4. Auswählen und Einstellen des Existenzbits des entsprechenden Arguments im EXEC-Schnittstellendeskriptor (EXEC Interface Descriptor, EID).
5. Ändern der Parameterliste zur Berücksichtigung des neuen Indikators für das Ende der Liste.

## Entfernen von Benutzerargumenten

Benutzerexitprogramme können Argumente entfernen, die den Schlüsselwörtern LENGTH und MAXLIFETIME zugeordnet sind. Hierbei liegt die vollständige Verantwortung für das Entfernen von Argumenten beim Programm.

Unter der Voraussetzung, dass das zu entfernende Argument vorhanden ist, muss das Exitprogramm folgende Aktionen ausführen:

1. Umschalten des Existenzbits des entsprechenden Arguments im EID auf '0'b.
2. Ändern der Parameterliste zur Berücksichtigung des neuen Indikators für das Ende der Liste.

---

## Exit XEPCAP für Ereigniserfassung

Der Exit XEPCAP wird unmittelbar vor der Erfassung eines Ereignisses durch die CICS-Ereignisverarbeitung aufgerufen. Verwenden Sie den Exit XEPCAP, um zu erkennen, wann Ereignisse erfasst werden.

### Exitspezifische Parameter

#### UEPEPCX

Adresse der Ereigniskontextdatenstruktur (EPCX). Dieser Parameter enthält Informationen zu dem Ereignis, das erfasst wird. Weitere Informationen zu EPCX finden Sie unter EPCX - Event Processing Context Container.

#### UEPEPTASK

Adresse eines 4-Byte-Felds (gepackt dezimal) mit der Tasknummer.

#### UEPLOAD

Adresse des Ladepunkts des Erfassungstransaktionsprogramms.

#### UEPRSA

Adresse des Registersicherungsbereichs für Erfassungstransaktionen, der den Inhalt der Register an dem Punkt enthält, an dem das Programm den EXEC CICS-Befehl abgesetzt hat. Wenn das Ereignis von der Programminitialisierung und nicht von einem API-Aufruf erfasst wird, wird dieser Parameter auf 0 gesetzt.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

### XPI-Aufrufe

Es können alle XPI-Aufrufe verwendet werden.

### API- und SPI-Befehle

Es können keine EXEC CICS-Befehle verwendet werden.

---

## Exits XEIIIN, XEIOU, XEISPIN und XEISPOUT für das EXEC-Schnittstellenprogramm

Es gibt im EXEC-Schnittstellenprogramm vier globale Benutzerexitpunkte, die Sie vor oder nach einem API- oder SPI-Aufruf verwenden können.

**XEIIIN** Wird vor der Ausführung eines beliebigen EXEC CICS-Befehls der Anwendungsprogrammierschnittstelle (API) oder der Systemprogrammierschnittstelle (SPI) aufgerufen.

### XEISPIN

Wird vor der Ausführung eines beliebigen EXEC CICS-SPI-Befehls aufgerufen, mit Ausnahme der folgenden Befehle:

- EXEC CICS ENABLE
- EXEC CICS DISABLE
- EXEC CICS EXTRACT EXIT
- EXEC CICS PERFORM DUMP
- EXEC CICS RESYNC ENTRYNAME

Es gilt folgende Reihenfolge:

TRACE – XEIIIN – XEISPIN – EDF – Befehl

### XEIOU

Wird nach der Ausführung eines beliebigen EXEC CICS-API-Befehls oder EXEC CICS-SPI-Befehls aufgerufen.

### XEISPOUT

Wird nach der Ausführung eines beliebigen EXEC CICS-SPI-Befehls aufgerufen, mit Ausnahme der für XEISPIN aufgelisteten Befehle.

Es gilt folgende Reihenfolge:

Befehl – EDF – XEISPOUT – XEIOU – TRACE

**Anmerkung:** Wird die Transaktion ausgesetzt (beispielsweise während eines E/A-Wartestatus), kann es zu einer asynchronen Verarbeitung dieser Exits kommen. Diese Situation kann auch unter der Transaktion CEDF auftreten, da CEDF zwischen den Aufrufen der Exits XEISPIN und XEISPOUT der Anwendung eigene EXEC CICS-Befehle ausgibt.

Wenn beispielsweise derselbe globale Arbeitsbereich (GWA) von den Exits XEIIIN und XEIOU gemeinsam genutzt wird, dann müssen Sie die Möglichkeit der asynchronen Verarbeitung zulassen, um die Integrität der Daten zu gewährleisten und unvorhersehbare Ergebnisse zu verhindern.

Beim Einstieg in die Exits enthält der exitspezifische Parameter UEPARG die Adresse der Befehlsparameterliste.

## Die Befehlsparameterliste

Der erste Parameter in der Liste verweist auf eine Zeichenfolge von Daten, die als *Argument 0* bezeichnet werden. Die anderen Parameter verweisen auf die Werte, die für die Parameter angegeben sind, die im Befehl übergeben werden.

*Argument 0* beginnt mit einem 2-Byte-Funktionscode, der den Befehl angibt. Die Funktionscodes sind unter EIB fields und EXEC interface block (EIB) response and function codes dokumentiert. Auf den Funktionscode folgt ein 2-Byte-Feld, das „Existenzbits“ enthält, die angeben, ob Argumente im Befehl übergeben werden. Beispiel für den folgenden Befehl:

```
EXEC CICS LINK PROGRAM('MYPROG')
```

In diesem Fall beginnt Argument 0 mit dem Funktionscode X'0E02' (LINK). Es ist das Existenzbit 1 gesetzt; dies weist darauf hin, dass es ein Argument 1 gibt (nämlich 'MYPROG').

Die Entsprechung zwischen Befehlsparametern (wie beispielsweise PROGRAM) und ihren Positionen und Werten in der Parameterliste (in diesem Fall Argument 1, 'MYPROG') kann vom umgesetzten Code für den jeweiligen Befehl abgeleitet werden.

### Important:

Das Ändern von CICS-Befehlen durch Ändern des Arguments 0 wird nicht unterstützt und führt zu unerwarteten Fehlern oder Ergebnissen.

Wenn beispielsweise ein Anwendungsprogramm in Assembler oder PL/I geschrieben ist und Sie das Argument 0 ändern, schreiben Sie in den Programmspeicher (also in den Speicher, der vom Programm selbst belegt ist), was zu einem Abbruch vom Typ '0C4' führen kann. Des Weiteren wird durch das Ändern von Argument 0 nicht nur der CICS-Befehl für diese Ausführung des Befehls im Anwendungsprogramm geändert, sondern es wird auch der CICS-Befehl in der Kopie des Anwendungsprogramms im virtuellen Speicher geändert. Dies bedeutet, dass die nächste Task, die dieselbe Kopie des Programms aufruft, ebenfalls den geänderten Befehl ausführt.

Dieses spezielle Beispiel zur Erläuterung der Gefahr, die mit dem Ändern von Argument 0 einhergeht, gilt nicht für COBOL- oder C-Anwendungsprogramme. Dennoch sollten Sie CICS-Befehle nicht für Anwendungsprogramme ändern, die in einer der unterstützten Sprache geschrieben sind.

## Umgehen von Befehlen

Ein Exitprogramm für XEIIIN oder für XEISPIN kann die Ausführung eines Befehls umgehen, indem der Rückgabecode UERCBYP gesetzt wird. Wird dieser Rückgabecode gesetzt, wird nicht die Ausführungsdiagnosefunktion (Execution Diagnostic Facility, EDF) aufgerufen, sondern es werden XEISPOUT, XEIOUT und der Exit-Trace aufgerufen, sofern aktiviert.

Durch das Umgehen eines EXEC CICS-Befehls kann ein Exitprogramm die CICS-Funktion beispielsweise durch die eigene Verarbeitung ersetzen.

Bevor UERCBYP gesetzt wird, sollte das Programm den Wert überprüfen, auf den von UEPPGM verwiesen wird, um sicherzustellen, dass es keinen EXEC CICS-Befehl umgeht, der von CICS ausgegeben wird.

## Exit XEIIIN

Der Exit XEIIIN wird vor der Ausführung eines beliebigen EXEC CICS-API-Befehls oder EXEC CICS-SPI-Befehls aufgerufen.

### Exitspezifische Parameter

#### UEPARG

Adresse der EXEC-Befehlsparameterliste.

#### UEPEXECB

Adresse des EXEC-Schnittstellenblocks (EIB) des Systems.

#### UEPUSID

Adresse der Benutzer-ID aus 8 Zeichen.

#### UEPPGM

Adresse des Anwendungsprogrammnamens aus 8 Zeichen.

#### UEPLOAD

Adresse des Ladepunkts des Anwendungsprogramms.

#### UEPRSA

Adresse des Registersicherungsbereichs der Anwendung. Dieser Bereich enthält den Inhalt der Register an dem Punkt, an dem das Programm den EXEC CICS-Befehl ausgegeben hat.

#### UEP\_EI\_PBTOK

Adresse eines 4-Byte-Felds mit dem Leistungsblocktoken (Performance Block Token, PBTOK) von z/OS Workload Manager (WLM). Mithilfe dieses Tokens kann ein Exitprogramm auf Informationen (wie beispielsweise das Serviceklassetoken SERVCLS) im WLM-Leistungsblock zugreifen. Hierzu muss das Programm das WLM EXTRACT-Makro IWMMECTR verwenden und dabei das Leistungsblocktoken als Eingabeparameter MONTKN übergeben. Weitere Informationen zum Makro IWMMECTR finden Sie unter z/OS MVS Programming: Workload Management Services.

Exitprogramme dürfen nicht versuchen, den Leistungsblock zu ändern, da die Ergebnisse eines solchen Versuchs unvorhersehbar sind.

### Rückgabecodes

#### UERCNORM

Verarbeitung fortsetzen.

#### UERCBYP

Ausführung dieses Befehls umgehen.

#### UERCPURG

Task während des XPI-Aufrufs bereinigt.

### XPI-Aufrufe

Es können alle verwendet werden.

## Exit XEISPIN

Der Exit XEISPIN wird vor der Ausführung bestimmter SPI-Befehle aufgerufen. Der Exit wird nicht für die Befehle **ENABLE**, **DISABLE**, **EXTRACT EXIT**, **PERFORM DUMP** und **RESYNC ENTRYNAME** aufgerufen.

### Exitspezifische Parameter

**UEPARG**

Adresse der EXEC-Befehlsparameterliste.

**UEPEXECB**

Adresse des EXEC-Schnittstellenblocks (EIB) des Systems.

**UEPUSID**

Adresse der Benutzer-ID aus 8 Zeichen.

**UEPPGM**

Adresse des Anwendungsprogrammnamens aus 8 Zeichen.

**UEPLOAD**

Adresse des Ladepunkts des Anwendungsprogramms.

**UEPRSA**

Adresse des Registersicherungsbereichs des Anwendungsprogramms. Dieser Bereich enthält den Inhalt der Register an dem Punkt, an dem das Programm den EXEC CICS-Befehl ausgegeben hat.

**UEP\_EI\_PBTOK**

Adresse eines 4-Byte-Felds mit dem Leistungblocktoken (Performance Block Token, PBTOK) von z/OS Workload Manager (WLM). Mithilfe dieses Tokens kann ein Exitprogramm auf Informationen im WLM-Leistungsblock zugreifen, wie beispielsweise auf das Serviceklassetoken SERVCLS. Hierzu muss das Programm das WLM EXTRACT-Makro IWMMECTR verwenden und dabei das Leistungsblocktoken als Eingabeparameter MONTKN übergeben. Weitere Informationen zum Makro IWMMECTR finden Sie unter z/OS MVS Programming: Workload Management Services.

Exitprogramme dürfen nicht versuchen, den Leistungsblock zu ändern, da die Ergebnisse eines solchen Versuchs unvorhersehbar sind.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen.

**UERCBYP**

Ausführung dieses Befehls umgehen.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

**Exit XEIOUT**

Der Exit XEIOUT wird nach der Ausführung eines beliebigen EXEC CICS-API-Befehls oder EXEC CICS-SPI-Befehls aufgerufen.

**Exitspezifische Parameter****UEPARG**

Adresse der EXEC-Befehlsparameterliste.

**UEPEXECB**

Adresse des EXEC-Schnittstellenblocks (EIB) des Systems.

**UEPUSID**

Adresse der Benutzer-ID aus 8 Zeichen.

**UEPPGM**

Adresse des Anwendungsprogrammnamens aus 8 Zeichen.

**UEPLOAD**

Adresse des Ladepunkts des Anwendungsprogramms.

**UEPRSA**

Adresse des Registersicherungsbereichs der Anwendung. Dieser Bereich enthält den Inhalt der Register an dem Punkt, an dem das Programm den EXEC CICS-Befehl ausgegeben hat.

**UEP\_EI\_PBTOK**

Adresse eines 4-Byte-Felds mit dem Leistungblocktoken (Performance Block Token, PBTOK) von z/OS Workload Manager (WLM). Mithilfe dieses Tokens kann ein Exitprogramm auf Informationen (wie beispielsweise das Serviceklassetoken SERVCLS) im WLM-Leistungsblock zugreifen. Hierzu muss das Programm das WLM EXTRACT-Makro IWMMECTR verwenden und dabei das Leistungsblocktoken als Eingabeparameter MONTKN übergeben. Weitere Informationen zum Makro IWMMECTR finden Sie unter z/OS MVS Programming: Workload Management Services.

Exitprogramme dürfen nicht versuchen, den Leistungsblock zu ändern, da die Ergebnisse eines solchen Versuchs unvorhersehbar sind.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

**Exit XEISPOUT**

Der Exit XEISPOUT wird nach der Ausführung bestimmter SPI-Befehle aufgerufen. Der Exit wird nicht für die Befehle **ENABLE**, **DISABLE**, **EXTRACT EXIT**, **PERFORM DUMP** und **RESYNC ENTRYNAME** aufgerufen.

**Exitspezifische Parameter****UEPARG**

Adresse der EXEC-Befehlsparameterliste.

**UEPEXECB**

Adresse des EXEC-Schnittstellenblocks (EIB) des Systems.

**UEPUSID**

Adresse der Benutzer-ID aus 8 Zeichen.

**UEPPGM**

Adresse des Anwendungsprogrammnamens aus 8 Zeichen.

**UEPLOAD**

Adresse des Ladepunkts des Anwendungsprogramms.

#### **UEPRSA**

Adresse des Registersicherungsbereichs der Anwendung. Dieser Bereich enthält den Inhalt der Register an dem Punkt, an dem das Programm den EXEC CICS-Befehl ausgegeben hat.

#### **UEP\_EI\_PBTOK**

Adresse eines 4-Byte-Felds mit dem Leistungsbloktoken (Performance Block Token, PBTOK) von z/OS Workload Manager (WLM). Mithilfe dieses Tokens kann ein Exitprogramm auf Informationen (wie beispielsweise das Serviceklassetoken SERVCLS) im WLM-Leistungsblock zugreifen. Hierzu muss das Programm das WLM EXTRACT-Makro IWMMECTR verwenden und dabei das Leistungsbloktoken als Eingabeparameter MONTKN übergeben. Weitere Informationen zum Makro IWMMECTR finden Sie unter z/OS MVS Programming: Workload Management Services.

Exitprogramme dürfen nicht versuchen, den Leistungsblock zu ändern, da die Ergebnisse eines solchen Versuchs unvorhersehbar sind.

#### **Rückgabecodes**

##### **UERCNORM**

Verarbeitung fortsetzen.

##### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

#### **XPI-Aufrufe**

Es können alle verwendet werden.

---

## **Exits XSZARQ und XSZBRQ für die Front-End-Programmierschnittstelle (FEPI)**

Wenn Sie die Front-End-Programmierschnittstelle (FEPI) installiert haben, können Sie die globalen Benutzerexits XSZARQ und XSZBRQ vor und nach FEPI-Befehlen verwenden.

#### **XSZBRQ**

Wird aufgerufen, bevor ein FEPI-Befehl ausgeführt wird (aber nachdem die Syntax des Befehls validiert wurde und daher nach der EDF-Verarbeitung).

#### **XSZARQ**

Wird unmittelbar nach dem Abschluss eines FEPI-Befehls aufgerufen (vor der EDF-Verarbeitung).

Es ist zu beachten, dass sowohl die Befehle der FEPI-Anwendungsprogrammierung als auch die Befehle der Systemprogrammierung dazu führen, dass XSZBRQ und XSZARQ aufgerufen werden; im Fall der Systemprogrammierungsbefehle werden den Exitprogrammen jedoch keine aussagekräftigen Informationen zur Verfügung gestellt.

In Programmen, die über diese Exits aufgerufen werden, können keine Aufrufe der Schnittstelle zur Programmierung von Exits (XPI-Aufrufe) und auch keine EXEC CICS-Befehle verwendet werden. Mit den Exits können Sie die FEPI-Befehle und die verarbeiteten Daten überwachen. Sie können Befehle unterdrücken und bestimmte Befehlsoptionen ändern. So können Sie die Exits beispielsweise für Folgendes verwenden:

- Überwachung der Ausgabe von FEPI-Befehle.



- Workload-Routing.
- Externe Sicherheit für Anwendungsprogrammierungsbefehle.

## **XSZBRQ**

Der Exit XSZBRQ wird vor Ausführung eines FEPI-Befehls aufgerufen; die Eingabeparameter für den Befehl werden an das Exitprogramm übergeben.

Der größte Teil der übergebenen Informationen ist schreibgeschützt; Sie können jedoch ein Programm schreiben, um bestimmte Parameter zu aktualisieren. Hierbei wird die Gültigkeit der neuen Werte für die aktualisierten Parameter von FEPI nicht geprüft. Darüber hinaus kann das Exitprogramm entscheiden, ob die Anforderung verarbeitet oder umgangen werden soll. So können Sie XSZBRQ beispielsweise verwenden, um Befehle zu protokollieren, um Befehle zu umgehen, die gegen die Konventionen Ihrer Installation verstoßen, oder um Befehle durch Ändern der angegebenen Ziele oder Pools umzuleiten.

Zusammen enthalten UEPSZALP und UEPSZALT die Informationen, die zum Einleiten eines Dialogs erforderlich sind.

### **Zeitpunkt des Aufrufs**

Der Exit wird durch FEPI aufgerufen, bevor ein FEPI-Befehl ausgeführt wird, aber nachdem eine Syntax- und Semantikprüfung durchgeführt wurde.

### **Exitspezifische Parameter**

#### **UEPSZACT**

2-Byte-Feld zur Angabe des Befehls. Die Werte sind in Tabelle 3 auf Seite 65 aufgeführt.

#### **UEPSZCNV**

Feld aus 8 Zeichen mit der Dialog-ID (CONVID) für den Befehl. Gültig für die FEPI-Befehle ALLOCATE, SEND, RECEIVE, CONVERSE, EXTRACT, ISSUE, START und FREE.

#### **UEPSZALP**

Feld aus 8 Zeichen mit dem Namen des Pools (POOL). Änderbar und gültig für die FEPI-Befehle ALLOCATE und CONVERSE.

#### **UEPSZALT**

Feld aus 8 Zeichen mit dem Namen des Ziels (TARGET). Änderbar und gültig für die FEPI-Befehle ALLOCATE und CONVERSE.

#### **UEPSZTIM**

Vollwort-Binärfeld mit dem Zeitlimitwert (TIMEOUT). Änderbar und gültig für die FEPI-Befehle ALLOCATE, RECEIVE, CONVERSE und START.

#### **UEPSZSND**

Adresse des Datenbereichs für Senden (FROM). Gültig für die FEPI-Befehle CONVERSE und SEND.

#### **UEPSZSNL**

Vollwort-Binärfeld mit der Länge der Daten im Datenbereich für Senden (FROMLENGTH, LENGTH). Gültig für die FEPI-Befehle CONVERSE und SEND.

#### **UEPSZSTT**

Feld aus 4 Zeichen mit der Transaktions-ID (TRANSID). Änderbar und gültig für den FEPI-Befehl START.

#### **UEPSZSTM**

Feld aus 4 Zeichen mit der Terminal-ID (TERMIN). Änderbar und gültig für den FEPI-Befehl START.

#### **UEPSZSNK**

1-Bit-Flagfeld zur Angabe, ob die Daten im Tastenanschlagsformat (KEYSTROKE) vorliegen. Gültig für die FEPI-Befehle CONVERSE FORMATTED und SEND FORMATTED. Mögliche Werte:

##### **UEPSZSNK\_OFF**

Kein Tastenanschlagsformat.

##### **UEPSZSNK\_ON**

Tastenanschlagsformat.

#### **UEPSZSNE**

Feld aus 1 Zeichen mit dem Escapezeichen (ESCAPE) für Tastenanschlag. Gültig für die FEPI-Befehle CONVERSE FORMATTED und SEND FORMATTED.

### **Rückgabecodes**

#### **UERCNORM**

Verarbeitung fortsetzen.

#### **UERCBYP**

Anforderung nicht verarbeiten; INVREQ an die Anwendung zurückgeben.

**Anmerkung:** Das Exitprogramm kann keine Ereignisse (wie beispielsweise 'CICS-Systemabschluss' oder 'Taskende') umgehen.

### **XPI-Aufrufe**

Es dürfen keine XPI-Aufrufe verwendet werden.

## **XSZARQ**

XSZARQ wird unmittelbar nach der Ausführung eines FEPI-Befehls aufgerufen. Dabei werden die vom Befehl ausgegebenen Parameter an das Exitprogramm übergeben. Sämtliche übergebenen Informationen sind schreibgeschützt.

### **Zeitpunkt des Aufrufs**

Der Aufruf erfolgt durch FEPI unmittelbar nach der Verarbeitung eines FEPI-Befehls.

### **Exitspezifische Parameter**

#### **UEPSZACN**

2-Byte-Feld zur Angabe des Befehls. Die Werte sind in Tabelle 3 auf Seite 65 aufgeführt.

#### **UEPSZCON**

Feld aus 8 Zeichen mit der Dialog-ID (CONVID) für den Befehl. Gültig für die FEPI-Befehle ALLOCATE, SEND, RECEIVE, CONVERSE, EXTRACT, ISSUE, START und FREE.

#### **UEPSZRP2**

Vollwort mit dem Antwortcode für den Befehl (RESP2).

#### **UEPSZRVD**

Adresse des Datenbereichs für Empfangen (INTO). Gültig für die FEPI-Befehle RECEIVE, CONVERSE und EXTRACT FIELD.

### UEPSZRVL

Vollwort-Binärdatenfeld mit der Länge der Daten im Bereich für Empfangen (FLENGTH, TOFLENGTH). Gültig für die FEPI-Befehle RECEIVE, CONVERSE und EXTRACT FIELD.

### Rückgabecode

#### UERCNORM

Verarbeitung fortsetzen.

### XPI-Aufrufe

Es dürfen keine XPI-Aufrufe verwendet werden.

## Exitsspezifische Parameter UEPSZACT und UEPSZACN

Sowohl an XSZBRQ als auch an XSZARQ wird ein Parameter übergeben (UEPSZACT an XSZBRQ und UEPSZACN an XSZARQ), der den verarbeiteten Befehl oder das verarbeitete Ereignis angibt.

Tabelle 3 enthält die Entsprechungen zwischen den in UEPSZACT und UEPSZACN übergebenen Hexadezimalwerten und den durch sie dargestellten FEPI-Befehlen.

*Tabelle 3. Einstellungen von UEPSZACT für Exit XSZBRQ und Einstellungen von UEPSZACN für Exit XSZARQ*

Name	Einstellung (hexadezimal)	FEPI-Befehl oder Ereignis
UEPSZNOA	820E	AP NOOP
UEPSZOAL	8210	ALLOCATE
UEPSZOCF	8212	CONVERSE FORMATTED
UEPSZOCD	8214	CONVERSE DATASTREAM
UEPSZOXC	8216	EXTRACT CONV
UEPSZOXF	8218	EXTRACT FIELD
UEPSZOXS	821A	EXTRACT STSN
UEPSZOFR	821C	FREE
UEPSZOSU	821E	ISSUE
UEPSZORF	8220	RECEIVE FORMATTED
UEPSZORD	8222	RECEIVE DATASTREAM
UEPSZOSF	8224	SEND FORMATTED
UEPSZOSD	8226	SEND DATASTREAM
UEPSZOST	8228	START
UEPSZSDN	8402	Normale Beendigung von CICS 1
UEPSZSDI	8404	Sofortige Beendigung von CICS 1
UEPSZSDF	8406	Erzwungene Beendigung von CICS 1
UEPSZEOT	8408	CICS-Taskabschluss 1
UEPSZNOS	840E	SP NOOP
UEPSZOQY	8422	INQUIRE PROPERTYSET
UEPSZOIY	8428	INSTALL PROPERTYSET
UEPSZODY	8430	DISCARD PROPERTYSET
UEPSZOQN	8442	INQUIRE NODE
UEPSZOTN	8444	SET NODE
UEPSZOIN	8448	INSTALL NODELIST
UEPSZOAD	844A	ADD POOL
UEPSZODE	844C	DELETE POOL
UEPSZODN	8450	DISCARD NODELIST
UEPSZOQP	8462	INQUIRE POOL
UEPSZOTP	8464	SET POOL

Tabelle 3. Einstellungen von UEPSZACT für Exit XSZBRQ und Einstellungen von UEPSZACN für Exit XSZARQ (Forts.)

Name	Einstellung (hexadezimal)	FEPI-Befehl oder Ereignis
UEPSZOIP	8468	INSTALL POOL
UEPSZODP	8470	DISCARD POOL
UEPSZOQT	8482	INQUIRE TARGET
UEPSZOTT	8484	SET TARGET
UEPSZOIT	8488	INSTALL TARGETLIST
UEPSZODT	8490	DISCARD TARGETLIST
UEPSZOQC	84A2	INQUIRE CONNECTION
UEPSZOTC	84A4	SET CONNECTION

**Anmerkung:**

- 1 Diese Ereignisse werden intern von CICS generiert. Sie können nicht umgangen werden.

## Verwendung von XMEOUT zur Steuerung der Nachrichtenausgabe

Sie können den globalen Benutzerexit XMEOUT in der CICS-Nachrichtendomäne verwenden, um FEPI-Nachrichten zu unterdrücken oder umzuleiten.

Hierbei ist jedoch zu beachten, dass Fehlerbedingungen, die eine Nachricht generieren, auch einen Datensatz in der Warteschlange mit transienten Daten generieren. Es ist effizienter, solche Ereignisse mithilfe eines Überwachungsprogramms über die Warteschlange mit transienten Daten zu verarbeiten, als eine Nachricht zu duplizieren und dann eine entsprechende Aktion auszuführen.

## Exits XFCFRIN und XFCFROUT für die Dateisteuerungsdomäne

Der Exit XFCFRIN wird bei Eintritt in den Hauptzugang für Dateisteueranforderungen (FCFR) aufgerufen; der Exit XFCFROUT wird nach Abschluss einer Dateisteueranforderung aufgerufen. Die Exits XFCFRIN und XFCFROUT müssen gemäß threadsicheren Standards codiert und als threadsicher deklariert werden, um die threadsichere Unterstützung für ferne Dateien nutzen zu können.

### XFCFRIN

XFCFRIN ermöglicht es Ihnen, ein Programm zu schreiben, um eine oder mehrere der folgenden Tasks auszuführen:

- Überwachung von Dateisteueranforderungen und Ermöglichung der fortgesetzten Verarbeitung dieser Anforderungen durch die CICS-Dateisteuerung.
- Abfangen von Dateisteueranforderungen und vollständige Umgehung der Verarbeitung durch die CICS-Dateisteuerung.
- Umleitung von Anforderungen an eine ferne Region.

Wenn das Exitprogramm eine Anforderung an die CICS-Dateisteuerung übergibt (ohne sie an eine ferne Region umzuleiten), sind keinerlei Änderungen an den Parametern zulässig. Wenn das Exitprogramm die Anforderung abfängt und die Dateisteuerung umgeht, gilt Folgendes:

- Das Exitprogramm muss alle Antworten und Ausgabeparameter zurückgeben, die ansonsten von der Dateisteuerung zurückgegeben worden

wären. Diese Antworten und Ausgabeparameter sind in den Beschreibungen der exitspezifischen Parameter als **Ausgabe** gekennzeichnet.

- Wenn die Anforderung über Funktionen übertragen wurde, muss das Exitprogramm angeben, ob die Spiegeltransaktion beendet werden darf. Bei bestimmten Dateisteueranforderungen muss zuvor eine andere Anforderung in derselben Transaktion ausgeführt worden sein. So muss beispielsweise vor READNEXT eine entsprechende Anforderung vom Typ STARTBR ausgeführt werden, und vor REWRITE muss eine entsprechende Anforderung vom Typ READ, READNEXT oder READPREV mit der Option UPDATE ausgeführt werden. Wenn die Spiegeltransaktion zwischen zwei solchen Anforderungen beendet wird, schlägt die zweite Anforderung wahrscheinlich fehl. Umgekehrt gilt: Eine Spiegeltransaktion, die unnötigerweise beibehalten wird, belegt weiterhin CICS-Ressourcen und trägt so möglicherweise zu Speicherproblemen und Sperrenkonflikten bei.
- CICS beendet das Durchsuchen von Dateien sowie ausstehende Aktualisierungen als Teil der Synchronisationspunktverarbeitung. Der Exit XFCFRIN wird jedoch nicht für Synchronisationspunkte aufgerufen. Wenn Sie diesen Aspekt des CICS-Verhaltens korrekt emulieren wollen oder wenn Sie wiederherstellbare Ressourcen unterstützen wollen, müssen Sie ein taskbezogene Benutzerexitprogramm aufrufen, das den Synchronisationspunktmanager terminiert (siehe Coding a program to be started by the CICS sync point manager).

Um die Anforderung an eine ferne Region umzuleiten, muss das Exitprogramm entweder einen entsprechenden Wert für den Parameter SYSID hinzufügen oder den Wert des Parameters SYSID entsprechend ändern. In diesem Fall kann es auch erforderlich sein, Werte für Schlüssellänge und Satzlänge anzugeben. Abgesehen von diesen Parametern sind keine Änderungen an anderen Parametern zulässig.

## XFCFROUT

Mithilfe von XFCFROUT können Sie die Ergebnisse abgeschlossener Dateisteueranforderungen überwachen. Wenn Sie sich beispielsweise dagegen entschieden haben, die Verarbeitung durch die CICS-Dateisteuerung zu umgehen, können Sie die (CICS-interne) Dateisteueranforderung analysieren, um Folgendes zu bestimmen: den Typ der Anforderung, die an die Dateisteuerung übergebenen Parameter sowie die zurückgegebenen Werte. Der Exit XFCFROUT wird in den beiden folgenden Fällen aufgerufen:

- Nachdem die CICS-Dateisteuerung ihre Verarbeitung beendet hat, entweder normal oder mit einem Fehler.
- Wenn das Exitprogramm für XFCFRIN die Verarbeitung durch die CICS-Dateisteuerung umgeht.

Sämtliche Parameter sind ausschließlich Eingabeparameter; das Exitprogramm kann keinen der Werte ändern.

Um IPIC-Verbindungen für das Übertragen von Dateisteueranforderungen über Funktionen zu verwenden, muss von XFCFRIN und XFCFROUT unbedingt überprüft werden, ob der Parameter UEPTERM einen Wert ungleich null aufweist, bevor versucht wird, ihn als Adresse zu verwenden. Bei Dateisteueranforderungen, die über Funktionen über eine IPIC-Verbindung übertragen worden sind, ist der Parameter UEPTERM null.

## Exit XFCFRIN

Der Exit XFCFRIN wird vor der Ausführung einer Dateisteueranforderung aufgerufen.

Die Anforderung kann folgenden Ursprung haben:

- Ausführung einer Anwendungsanforderung zum Verarbeiten einer Benutzerdatei.
- Empfang einer über Funktionen übertragenen Anforderung.
- Interne CICS-Anforderung zum Verarbeiten einer Systemdatei.

### Exitspezifische Parameter

#### UEPTRANID

Adresse der 4-Byte-Transaktions-ID.

#### UEPUSER

Adresse der 8-Byte-Benutzer-ID.

#### UEPTERM

Null oder die Adresse der 4-Byte-Terminal-ID. Wird keine Adresse zurückgegeben, so kann dies bedeuten, dass diese Anforderung mittels einer Funktion über eine IPIC-Verbindung übertragen wurde.

#### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens.

#### UEP\_FC\_FUNCTION

Adresse eines Byte, das die Funktion enthält. Die FCFR-Funktionen werden von denjenigen abgeleitet, die über die EXEC CICS-Schnittstelle verfügbar sind, wobei bestimmte Anforderungsoptionen (SET, INTO, UPDATE) in die Funktionswerte eingeschlossen sind. Beispiel: UEP\_FC\_FUN\_DELETE wird von EXEC CICS DELETE mit angegebener Option RIDFLD abgeleitet; UEP\_FC\_FUN\_REWRITE\_DELETE wird von EXEC CICS DELETE ohne RIDFLD abgeleitet. Mögliche Werte:

- UEP\_FC\_FUN\_READ\_INT0
- UEP\_FC\_FUN\_READ\_SET
- UEP\_FC\_FUN\_READ\_UPDATE\_INT0
- UEP\_FC\_FUN\_READ\_UPDATE\_SET
- UEP\_FC\_FUN\_WRITE
- UEP\_FC\_FUN\_REWRITE
- UEP\_FC\_FUN\_REWRITE\_DELETE
- UEP\_FC\_FUN\_DELETE
- UEP\_FC\_FUN\_UNLOCK
- UEP\_FC\_FUN\_START\_BROWSE
- UEP\_FC\_FUN\_READ\_NEXT\_INT0
- UEP\_FC\_FUN\_READ\_NEXT\_SET
- UEP\_FC\_FUN\_READ\_PREVIOUS\_INT0
- UEP\_FC\_FUN\_READ\_PREVIOUS\_SET
- UEP\_FC\_FUN\_READ\_NEXT\_UPDATE\_INT0
- UEP\_FC\_FUN\_READ\_NEXT\_UPDATE\_SET
- UEP\_FC\_FUN\_READ\_PREVIOUS\_UPDATE\_INT0
- UEP\_FC\_FUN\_READ\_PREVIOUS\_UPDATE\_SET
- UEP\_FC\_FUN\_RESET\_BROWSE
- UEP\_FC\_FUN\_END\_BROWSE

## UEPTSTOK

Adresse eines 4-Byte-Tokens, das während des gesamten Lebenszyklus einer Task gültig ist. Siehe Using the task token UEPTSTOK.

## UEP\_FC\_FILE\_NAME

Adresse eines änderbaren 8-Byte-Felds, das den Dateinamen enthält.

## UEP\_FC\_BUFFER\_P

Adresse eines Vollworts, das die Adresse des vom Ersteller der Anforderung bereitgestellten Puffers enthält, in dem der Datensatz (**Ausgabe**) bei Abschluss einer Anforderung vom Typ READ, READ NEXT oder READ PREV mit der Option INTO zurückgegeben werden soll.

## UEP\_FC\_BUFFER\_L

Adresse eines Vollworts, das (für Anforderungen vom Typ READ, READ NEXT und READ PREV) den Wert für die Länge (LENGTH) des Puffers enthält, in den der Datensatz gelesen werden soll.

## UEP\_FC\_RECORD\_P

Eine der folgenden Adressen:

- Bei Anforderungen vom Typ READ, READ NEXT oder READ PREV mit Option SET: Adresse eines Vollworts, in dem die Adresse (**Ausgabe**) eines Puffers zurückgegeben werden soll, in den der Datensatz gestellt wird. Der Puffer selbst wird entweder von der CICS-Dateisteuerung bereitgestellt oder vom Exitprogramm, wenn das Exitprogramm die Dateisteuerung umgeht.
- Bei Anforderungen vom Typ WRITE oder REWRITE: Adresse eines Vollworts, das die Adresse des Datensatzes enthält, der geschrieben werden soll.

## UEP\_FC\_RECORD\_L

Adresse eines Vollworts, das (für Anforderungen vom Typ READ, WRITE, REWRITE, READ NEXT und READ PREV) den Wert für die Länge (LENGTH) enthält.

Bei allen Anforderungen vom Typ READ, READ NEXT oder READ PREV handelt es sich um ein **Ausgabefeld**, in das bei Rückgabe die tatsächliche Länge des gelesenen Satzes gestellt wird.

**Warnung:** Bei Anforderungen mit Angabe von INTO darf der Wert für LENGTH nicht in einen Wert geändert werden, der größer ist als der im Feld UEP\_FC\_BUFFER\_L angegebene Wert. Andernfalls kann es in der Anwendung zu einer Speicherüberlagerung kommen.

Bei Anforderungen vom Typ WRITE oder REWRITE handelt es sich um ein optionales Feld, das bei Vorhandensein die Länge des zu schreibenden Datensatzes enthält. Wird das Feld nicht angegeben, enthält das Vollwort binäre Nullen. Verwenden Sie in diesem Fall zum Ändern der Satzlänge für eine ferne Datei stattdessen UEP\_FC\_M\_RECORD\_L.

Lesen Sie auch die Beschreibung des Parameters UEP\_FC\_SYSID.

## UEP\_FC\_MAX\_RECORD\_L

Adresse eines Vollworts, das die maximale Satzlänge (**Ausgabe**)

der Datei enthält. (Bei der CICS-Funktionsverlagerung wird dieser Wert verwendet, um den Eintrag der Datei in der Steuertabelle der fernen Datei zu aktualisieren.)

#### **UEP\_FC\_RECORD\_ID\_P**

Adresse eines Vollworts, das die Adresse des Werts für RIDFLD (Satzkennung) enthält. Erläuterungen dazu, wann es sich bei der Satz-kennung um ein Eingabe- bzw. **Ausgabefeld** handelt, finden Sie in Tabelle 4 auf Seite 87.

#### **UEP\_FC\_RECORD\_ID\_L**

Adresse des Halbwortwerts für KEYLENGTH; hierbei handelt es sich um die Länge (bzw. möglicherweise die Teillänge) der Satz-kennung.

KEYLENGTH ist ein optionaler Eingabeparameter bei Anforderungen vom Typ READ, WRITE, DELETE, START BR, READ NEXT, READ PREV und RESET BR. Wird das Feld nicht angegeben, enthält das Halbwort binäre Nullen. Verwenden Sie in diesem Fall zum Ändern der Schlüssellänge für eine ferne Datei stattdessen UEP\_FC\_M\_RECORD\_ID\_L.

Lesen Sie auch die Beschreibung des Parameters UEP\_FC\_SYSID.

#### **UEP\_FC\_FULL\_RECORD\_ID\_L**

Adresse des Halbwortwerts für die volle Länge der Satz-kennung.

Die volle Länge der Satz-kennung wird als obligatorisches **Ausgabefeld** in Anforderungen vom Typ READ NEXT und READ PREV zurückgegeben. Der Wert wird bei der CICS-Funktionsverlagerung verwendet.

#### **UEP\_FC\_RECORD\_ID\_TYPE**

Adresse eines Byte, das (für Anforderungen vom Typ READ, WRITE, DELETE, START BR, READ NEXT, READ PREV und RESET BR) den Typ für RIDFLD enthält. Bei Eingabe für den Exit wird dieser Parameter auf einen der folgenden Werte gesetzt:

##### **UEP\_FC\_KEY**

Zugriff auf VSAM KSDS oder AIX PATH.

##### **UEP\_FC\_RBA**

Zugriff auf VSAM ESDS oder KSDS über RBA.

##### **UEP\_FC\_RRN**

Zugriff auf VSAM RRDS.

##### **UEP\_FC\_DEBKEY**

BDAM-Entblockung nach Schlüssel (nur Anforderungen vom Typ READ, DELETE, START BR und RESET BR).

##### **UEP\_FC\_DEBREC**

BDAM-Entblockung nach relativem Datensatz (nur Anforderungen vom Typ READ, DELETE, START BR und RESET BR).

##### **UEP\_FC\_XRBA**

Zugriff auf VSAM ESDS (erweitert).

#### **UEP\_FC\_REQID**

Adresse (für Anforderungen vom Typ START BR, READ NEXT, READ PREV, RESET BR und END BR) des Halbwortwerts für REQID.



**UEP\_FC\_NUMREC**

Adresse des Vollwortwerts für NUMREC (**Ausgabe**), in dem (bei Anforderungen vom Typ DELETE mit RIDFLD) die Anzahl der gelöschten Datensätze zurückgegeben wird.

**UEP\_FC\_KEY\_COMPARISON**

Adresse eines Byte, das (für Anforderungen vom Typ READ, START BR und RESET BR) die Schlüsselvergleichseinstellung enthält. Bei Eingabe für den Exit wird dieser Parameter auf einen der folgenden Werte gesetzt:

**UEP\_FC\_EQUAL**

Vergleich vom Typ "Schlüssel ist gleich" soll verwendet werden.

**UEP\_FC\_GTEQ**

Vergleich vom Typ "Schlüssel ist größer oder gleich" soll verwendet werden.

**UEP\_FC\_GENERIC**

Adresse eines Byte, das (für Anforderungen vom Typ READ, DELETE, START BR und RESET BR) die generische Schlüsseleinstellung enthält. Bei Eingabe für den Exit wird dieser Parameter auf einen der folgenden Werte gesetzt:

**UEP\_FC\_GENERIC\_KEY**

Der generische Schlüssel soll für die Schlüsselsuche verwendet werden.

**UEP\_FC\_FULL\_KEY**

Der vollständige Schlüssel soll für die Schlüsselsuche verwendet werden.

**UEP\_FC\_MASS\_INSERT**

Adresse eines Byte, das (für Anforderungen vom Typ WRITE) die Einstellung für die Masseneinfügung enthält. Bei Eingabe für den Exit wird dieser Parameter auf einen der folgenden Werte gesetzt:

**UEP\_FC\_SEQUENTIAL\_WRITE**

Datensätze sollen im sequenziellen Modus geschrieben werden.

**UEP\_FC\_DIRECT\_WRITE**

Datensätze sollen im Direktmodus geschrieben werden.

**UEP\_FC\_READ\_INTEGRITY**

Adresse eines Byte, das (für Anforderungen vom Typ READ, READ NEXT und READ PREV ohne Aktualisierung) die Einstellung für Leseintegrität enthält. (In den aktuellen Versionen von CICS gilt diese Einstellung nur für VSAM RLS.) Bei Eingabe für den Exit wird dieser Parameter auf einen der folgenden Werte gesetzt:

**UEP\_FC\_CR**

Es soll die Leseintegrität für konsistentes Lesen verwendet werden.

**UEP\_FC\_FCT\_VALUE**

Die Leseintegrität entspricht der Einstellung in der Dateidefinition FILE.

#### UEP\_FC\_NRI

Der Datensatz soll ohne Leseintegrität gelesen werden.

#### UEP\_FC\_RR

Es soll die Leseintegrität für wiederholbares Lesen verwendet werden.

#### UEP\_FC\_TOKEN

Adresse eines Vollworts, das den Wert für TOKEN enthält.

Wenn es sich bei der Anforderung um den Typ READ, READ NEXT oder READ PREV mit Aktualisierung handelt und die Adresse ungleich null ist, so ist der Bereich ein **Ausgabefeld**, in dem das Token zurückgegeben werden soll.

Handelt es sich bei der Anforderung um den Typ REWRITE, DELETE ohne RIDFLD oder UNLOCK, ist der Bereich ein Eingabefeld.

#### UEP\_FC\_SYSID

Adresse eines 4-Byte-Bereichs, der die SYSID zur Angabe der fernen Region enthalten soll. Bei Eingabe für den Exit enthält der Bereich einen der beiden folgenden Werte:

- Wert der Option SYSID des API-Aufrufs *oder*
- Leerzeichen (wenn SYSID nicht angegeben wurde).

Um die Anforderung an eine andere Region umzuleiten, muss das Exitprogramm die SYSID der entsprechenden Zielregion in diesen **Ausgabebereich** stellen.

Wird dieser Parameter durch das Exitprogramm festgelegt, dann wird die Anforderung mittels Funktionen von der Dateisteuerung ohne Verweis auf die Attribute der Datei übertragen. Wurde die Schlüssellänge nicht in die Anforderung eingeschlossen, muss das Exitprogramm den entsprechenden Wert durch Einstellen des Parameters UEP\_FC\_RECORD\_ID\_L festlegen.

Ebenso gilt: Handelt es sich bei der Anforderung um den Typ WRITE oder REWRITE und die Schlüssellänge wurde in der Anforderung nicht angegeben, muss das Exitprogramm den entsprechenden Wert durch Einstellen des Parameters UEP\_FC\_RECORD\_L festlegen.

#### UEP\_FC\_LENGTH\_ERROR\_CODE

Adresse eines 1-Byte-**Ausgabebereichs**, der den Längenfehlercode enthält, der nach Abschluss einer Anforderung zurückgegeben werden soll. Mögliche Werte:

- UEP\_FC\_LENGTH\_OK
- UEP\_FC\_BUFFER\_LEN\_TOO\_SMALL
- UEP\_FC\_RECORD\_LEN\_TOO\_LARGE
- UEP\_FC\_BUFFER\_LEN\_NOT\_FILE\_LEN
- UEP\_FC\_RECORD\_LEN\_NOT\_FILE\_LEN

#### UEP\_FC\_DUPLICATE\_KEY\_CODE

Adresse eines 1-Byte-**Ausgabebereichs**, der angibt, ob die Anforderung mehr als einen Datensatz für den angegebenen Schlüssel gefunden hat. Mögliche Werte:

- UEP\_FC\_DUPLICATE KEY
- UEP\_FC\_NOT\_DUPLICATE KEY

#### **UEP\_FC\_ACCMETH\_RETURN\_CODE**

Adresse eines 4-Byte-**Ausgabebereichs**, in dem Informationen in Abhängigkeit von der Zugriffsmethode zurückgegeben werden sollen, wenn als Antwort entweder UEP\_FC\_REASON\_ACCMETH\_REQUEST\_ERROR oder UEP\_FC\_REASON\_IO\_ERROR zurückgegeben wird.

Der zurückgegebene Wert wird in Byte 2-5 von EIBRCODE eingefügt.

#### **UEP\_FC\_RESPONSE**

Adresse eines 1-Byte-**Ausgabebereichs**, der nach Abschluss einer Anforderung die entsprechende Antwort enthält:

##### **UEP\_FC\_RESPONSE\_OK**

Die Verarbeitung wurde ohne Fehler abgeschlossen.

##### **UEP\_FC\_RESPONSE\_EXCEPTION**

Die Verarbeitung wurde mit einer Fehlerbedingung abgeschlossen. Die Ursache wird in UEP\_FC\_REASON angegeben.

##### **UEP\_FC\_RESPONSE\_DISASTER**

Es ist ein Fehler aufgetreten, durch den der Abschluss der Verarbeitung verhindert wird. In der Regel ist dies die Folge einer Antwort vom Typ DISASTER eines XPI-Funktionsaufrufs oder die Folge einer Beschädigung von Daten, die über UEPGAA oder UEPTSTOK adressiert werden.

Wenn Sie diese Antwort festlegen, geht der Aufrufende der Dateisteuerung davon aus, dass eine Erfassung von Fehlerdaten beim ersten Auftreten erfolgt ist. Wenn Sie eine Antwort vom Typ DISASTER einer XPI-Anforderung unverändert weiterleiten, dann sollte die Erfassung von Fehlerdaten beim ersten Auftreten bereits erfolgt sein; falls nicht, sollte versucht werden, ausreichend Informationen zu erfassen, um den Fehler erfolgreich diagnostizieren zu können. Hierfür ist unter Umständen die XPI-Funktion DFHDUDUX SYSTEM\_DUMP geeignet.

##### **UEP\_FC\_RESPONSE\_INVALID**

Das Exitprogramm wurde mit einer ungültigen Parameterliste aufgerufen; dies weist auf einen internen Logikfehler von CICS hin. Es ist zu beachten, dass bei einer ungültigen Parameterliste, die auf einen Anwendungsfehler hinweist, eine Ausnahmeantwort vom Typ EXCEPTION ausgegeben werden sollte.

##### **UEP\_FC\_RESPONSE\_PURGED**

Ein XPI-Funktionsaufruf hat eine Antwort vom Typ PURGED empfangen. Die Einstellung dieser Antwort entspricht der Einstellung des Rückgabecodes UERCPURG, mit der Ausnahme, dass Änderungen an der Parameterliste berücksichtigt werden.

#### **UEP\_FC\_REASON**

Adresse eines 1-Byte-**Ausgabebereichs**, der nach Abschluss einer Anforderung mit einer Antwort vom Typ EXCEPTION die entsprechende Ursache enthält. Mögliche Ursachen:

- UEP\_FC\_REASON\_ACCMETH\_REQUEST\_ERROR

- UEP\_FC\_REASON\_DELETE\_AFTER\_READ\_UPDATE
- UEP\_FC\_REASON\_DELETE\_BEFORE\_READ\_UPDATE
- UEP\_FC\_REASON\_DUPLICATE\_READ\_UPDATE
- UEP\_FC\_REASON\_DUPLICATE\_RECORD
- UEP\_FC\_REASON\_DUPLICATE\_REQID
- UEP\_FC\_REASON\_END\_OF\_FILE
- UEP\_FC\_REASON\_FILE\_DISABLED
- UEP\_FC\_REASON\_FILE\_NOT\_OPEN
- UEP\_FC\_REASON\_FILE\_NOT\_FOUND
- UEP\_FC\_REASON\_FULL\_KEY\_WRONG\_LENGTH
- UEP\_FC\_REASON\_GENERIC\_DELETE\_NOT\_KSDS
- UEP\_FC\_REASON\_GENERIC\_KEY\_TOO\_LONG
- UEP\_FC\_REASON\_ILLEGAL\_KEY\_TYPE\_CHANGE
- UEP\_FC\_REASON\_INSUFFICIENT\_SPACE
- UEP\_FC\_REASON\_INVALID\_UPDATE\_TOKEN
- UEP\_FC\_REASON\_IO\_ERROR
- UEP\_FC\_REASON\_KEY\_LENGTH\_NEGATIVE
- UEP\_FC\_REASON\_KSDS\_AND\_XRBA
- UEP\_FC\_REASON\_NO\_VARIABLE\_LENGTH
- UEP\_FC\_REASON\_NOTAUTH
- UEP\_FC\_REASON\_NOT\_EXTENDED
- UEP\_FC\_REASON\_READPREV\_IN\_GENERIC\_BROWSE
- UEP\_FC\_REASON\_RECORD\_NOT\_FOUND
- UEP\_FC\_REASON\_REWRITE\_BEFORE\_READ\_UPDATE
- UEP\_FC\_REASON\_RIDFLD\_KEY\_NOT\_RECORD\_KEY
- UEP\_FC\_REASON\_UNKNOWN\_REQID\_ENDBR
- UEP\_FC\_REASON\_UNKNOWN\_REQID\_READNEXT
- UEP\_FC\_REASON\_UNKNOWN\_REQID\_READPREV
- UEP\_FC\_REASON\_UNKNOWN\_REQID\_RESETBR

#### **UEP\_FC\_EXIT\_TOKEN**

Adresse eines 4-Byte-Tokens, das an XFCFROUT übergeben werden soll. Dies ermöglicht es Ihnen beispielsweise, einen Arbeitsbereich an den Exit XFCFROUT zu übergeben.

#### **UEP\_FC\_M\_RECORD\_L**

Adresse eines Vollworts mit dem Feld für die geänderte Satzlänge. Auf diese Weise können Sie den Wert für LENGTH ändern (nur für ferne Dateien), wenn das Feld LENGTH im API-Aufruf nicht angegeben ist. Wenn das Feld LENGTH im API-Aufruf angegeben ist, verwenden Sie zum Ändern des Werts für LENGTH stattdessen UEP\_FC\_RECORD\_L, da die Änderung des Werts in der Adresse, die von UEP\_FC\_M\_RECORD\_L gespeichert wird, keine Auswirkungen hat.

#### **UEP\_FC\_M\_RECORD\_ID\_L**

Adresse eines Halbworts, das das Feld für die geänderte Schlüssellänge enthält. Auf diese Weise können Sie den Wert für KEYLENGTH ändern (nur für ferne Dateien), wenn das Feld KEYLENGTH im API-Aufruf nicht angegeben ist. Wenn das Feld KEYLENGTH angegeben ist, verwenden Sie zum Ändern des Werts für KEYLENGTH stattdessen UEP\_FC\_RECORD\_ID\_L, da die Änderung des Werts an der Adresse, die von UEP\_FC\_M\_RECORD\_ID\_L gespeichert wird, keine Auswirkungen hat.

### **Rückgabecodes**

**UERCNORM**

Verarbeitung fortsetzen.

**UERCBYP**

Verarbeitung dieser Anforderung durch CICS umgehen. Wenn das Exitprogramm für eine über Funktionen übertragene Anforderung aufgerufen wurde, darf die Spiegeltransaktion beendet werden.

**UERCBYPL**

Verarbeitung dieser Anforderung durch CICS umgehen. Wenn das Exitprogramm für eine über Funktionen übertragene Anforderung aufgerufen wurde, darf die Spiegeltransaktion nicht beendet werden.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

**API- und SPI-Aufrufe**

Es können keine verwendet werden.

**Exit XFCFROUT**

Der Exit XFCFROUT wird nach dem Abschluss einer Dateisteueranforderung aufgerufen.

**Exitspezifische Parameter****UEPTRANID**

Adresse der 4-Byte-Transaktions-ID.

**UEPUSER**

Adresse der 8-Byte-Benutzer-ID.

**UEPTERM**

Null oder die Adresse der 4-Byte-Terminal-ID. Wird keine Adresse zurückgegeben, so kann dies bedeuten, dass diese Anforderung mittels einer Funktion über eine IPIC-Verbindung übertragen wurde.

**UEPPROG**

Adresse des 8-Byte-Anwendungsprogrammnamens.

**UEP\_FC\_FUNCTION**

Adresse eines Byte, das die Funktion enthält. Mögliche Werte:

- UEP\_FC\_FUN\_READ\_INT0
- UEP\_FC\_FUN\_READ\_SET
- UEP\_FC\_FUN\_READ\_UPDATE\_INT0
- UEP\_FC\_FUN\_READ\_UPDATE\_SET
- UEP\_FC\_FUN\_WRITE
- UEP\_FC\_FUN\_REWRITE
- UEP\_FC\_FUN\_REWRITE\_DELETE
- UEP\_FC\_FUN\_DELETE
- UEP\_FC\_FUN\_UNLOCK
- UEP\_FC\_FUN\_START\_BROWSE
- UEP\_FC\_FUN\_READ\_NEXT\_INT0
- UEP\_FC\_FUN\_READ\_NEXT\_SET
- UEP\_FC\_FUN\_READ\_PREVIOUS\_INT0
- UEP\_FC\_FUN\_READ\_PREVIOUS\_SET
- UEP\_FC\_FUN\_READ\_NEXT\_UPDATE\_INT0

- UEP\_FC\_FUN\_READ\_NEXT\_UPDATE\_SET
- UEP\_FC\_FUN\_READ\_PREVIOUS\_UPDATE\_INT0
- UEP\_FC\_FUN\_READ\_PREVIOUS\_UPDATE\_SET
- UEP\_FC\_FUN\_RESET\_BROWSE
- UEP\_FC\_FUN\_END\_BROWSE

#### **UEPTSTOK**

Adresse eines 4-Byte-Tokens, das während des gesamten Lebenszyklus einer Task gültig ist. Siehe Using the task token UEPTSTOK.

#### **UEP\_FC\_FILE\_NAME**

Adresse eines 8-Byte-Felds, das den Dateinamen enthält.

#### **UEP\_FC\_BUFFER\_P**

Adresse eines Vollworts, das die Adresse des vom Ersteller der Anforderung bereitgestellten Puffers enthält, in dem der Datensatz bei Abschluss einer Anforderung vom Typ READ, READ NEXT oder READ PREV mit der Option INTO zurückgegeben wird.

#### **UEP\_FC\_BUFFER\_L**

Adresse eines Vollworts, das (für Anforderungen vom Typ READ, READ NEXT und READ PREV) den Wert für die Länge (LENGTH) des Puffers enthält, in den der Datensatz gelesen wurde.

#### **UEP\_FC\_RECORD\_P**

Eine der folgenden Adressen:

- Bei Anforderungen vom Typ READ, READ NEXT oder READ PREV mit Option SET: Adresse eines Vollworts, in dem die Adresse eines Puffers zurückgegeben wird, in den der Datensatz gestellt wurde.
- Bei Anforderungen vom Typ WRITE oder REWRITE: Adresse eines Vollworts, das die Adresse des Datensatzes enthält, der geschrieben wurde.

#### **UEP\_FC\_RECORD\_L**

Adresse eines Vollworts, das (für Anforderungen vom Typ READ, WRITE, REWRITE, READ NEXT und READ PREV) den Wert für die Länge (LENGTH) enthält.

Bei allen Anforderungen vom Typ READ, READ NEXT oder READ PREV handelt es sich um ein Ausgabefeld, das die tatsächliche Länge des gelesenen Satzes enthält. Bei diesen Anforderungstypen ist dieser Satzlängenwert immer vorhanden, selbst wenn die Option LENGTH im EXEC CICS-API-Aufruf nicht angegeben wurde.

Bei Anforderungen vom Typ WRITE oder REWRITE handelt es sich um ein optionales Feld, das bei Vorhandensein die Länge des geschriebenen Datensatzes enthält.

#### **UEP\_FC\_MAX\_RECORD\_L**

Adresse eines Vollworts, das die maximale Satzlänge der Datei enthält.

#### **UEP\_FC\_RECORD\_ID\_P**

Adresse eines Vollworts, das die Adresse des Wertes von RIDFLD (Satzkennung) enthält. Erläuterungen dazu, wann es sich bei der Satzkennung um ein Eingabe- bzw. Ausgabefeld handelt, finden Sie in Tabelle 4 auf Seite 87.

**UEP\_FC\_RECORD\_ID\_L**

Adresse des Halbwortwerts für KEYLENGTH; hierbei handelt es sich um die Länge (bzw. möglicherweise die Teillänge) der Satz-  
kennung.

Lie Länge der Satz-  
kennung ist ein optionaler Eingabeparameter bei  
Anforderungen vom Typ READ, WRITE, DELETE, START BR,  
READ NEXT, READ PREV und RESET BR.

**UEP\_FC\_FULL\_RECORD\_ID\_L**

Adresse des Halbwortwerts für die volle Länge der Satz-  
kennung.  
(Die volle Länge der Satz-  
kennung entspricht dem Schlüsselwort  
KEYLENGTH der EXEC CICS-Schnittstelle.)

Die volle Länge der Satz-  
kennung wird als obligatorisches Ausgabe-  
feld in Anforderungen vom Typ READ NEXT und READ PREV  
zurückgegeben.

**UEP\_FC\_RECORD\_ID\_TYPE**

Adresse eines Byte, das (für Anforderungen vom Typ READ, WRI-  
TE, DELETE, START BR, READ NEXT, READ PREV und RESET  
BR) den Typ für RIDFLD enthält.

**UEP\_FC\_KEY**

Zugriff auf VSAM KSDS oder AIX PATH.

**UEP\_FC\_RBA**

Zugriff auf VSAM ESDS oder KSDS über RBA.

**UEP\_FC\_RRN**

Zugriff auf VSAM RRDS.

**UEP\_FC\_DEBKEY**

BDAM-Entblockung nach Schlüssel (nur Anforderungen  
vom Typ READ, DELETE, START BR und RESET BR).

**UEP\_FC\_DEBREC**

BDAM-Entblockung nach relativem Datensatz (nur Anfor-  
derungen vom Typ READ, DELETE, START BR und RESET  
BR).

**UEP\_FC\_XRBA**

Zugriff auf VSAM ESDS (erweitert).

**UEP\_FC\_REQID**

Adresse (für Anforderungen vom Typ START BR, READ NEXT,  
READ PREV, RESET BR und END BR) des Halbwortwerts für  
REQID.

**UEP\_FC\_NUMREC**

Adresse des Vollwortwerts von NUMREC, in dem (wenn die An-  
forderung DELETE mit RIDFLD ist) die Anzahl der Sätze zurück-  
gegeben wird, die gelöscht wurden.

**UEP\_FC\_KEY\_COMPARISON**

Adresse eines Byte, das (für Anforderungen vom Typ READ,  
START BR und RESET BR) die Schlüsselvergleichseinstellung ent-  
hält.

**UEP\_FC\_EQUAL**

Vergleich vom Typ "Schlüssel ist gleich".

**UEP\_FC\_GTEQ**

Vergleich vom Typ "Schlüssel ist größer oder gleich".

**UEP\_FC\_GENERIC**

Adresse eines Byte, das (für Anforderungen vom Typ READ, DELETE, START BR und RESET BR) die generische Schlüsseinstellung enthält.

**UEP\_FC\_GENERIC\_KEY**

Generischer Schlüssel, der für die Schlüsselsuche verwendet wird.

**UEP\_FC\_FULL\_KEY**

Vollständiger Schlüssel, der für die Schlüsselsuche verwendet wird.

**UEP\_FC\_MASS\_INSERT**

Adresse eines Byte, das (für Anforderungen vom Typ WRITE) die Einstellung für die Masseneinfügung enthält.

**UEP\_FC\_SEQUENTIAL\_WRITE**

Sequenzieller Modus.

**UEP\_FC\_DIRECT\_WRITE**

Direktmodus.

**UEP\_FC\_READ\_INTEGRITY**

Adresse eines Byte, das (für Anforderungen vom Typ READ, READ NEXT und READ PREV ohne Aktualisierung) die Einstellung für Leseintegrität enthält. (In den aktuellen Versionen von CICS gilt diese Einstellung nur für VSAM RLS.)

**UEP\_FC\_CR**

Leseintegrität für konsistentes Lesen.

**UEP\_FC\_FCT\_VALUE**

Leseintegrität gemäß Einstellung in der Dateidefinition FILE.

**UEP\_FC\_NRI**

Keine Leseintegrität.

**UEP\_FC\_RR**

Leseintegrität für wiederholbares Lesen.

**UEP\_FC\_TOKEN**

Adresse eines Vollworts, das den Wert für TOKEN enthält.

Wenn es sich bei der Anforderung um den Typ READ, READ NEXT oder READ PREV mit Aktualisierung handelt und die Adresse ungleich null ist, so ist der Bereich ein Ausgabefeld, in dem das Token zurückgegeben wird.

Handelt es sich bei der Anforderung um den Typ REWRITE, DELETE ohne RIDFLD oder UNLOCK, ist der Bereich ein Eingabefeld.

**UEP\_FC\_SYSID**

Adresse eines 4-Byte-Bereichs, der die SYSID zur Angabe der fern-Region enthält. Bei Eingabe für den Exit XFCFROUT enthält der Bereich einen der beiden folgenden Werte:

- Wert der Option SYSID des API-Aufrufs *oder*
- Leerzeichen (wenn SYSID nicht angegeben wurde), oder



- Wert des SYSID-Werts, der durch den Exit XFCFRIN angegeben wird.

#### **UEP\_FC\_LENGTH\_ERROR\_CODE**

Adresse eines 1-Byte-Bereichs, der den Längenfehlercode enthält, der nach Abschluss der Anforderung zurückgegeben wird. Mögliche Werte:

- UEP\_FC\_LENGTH\_OK
- UEP\_FC\_BUFFER\_LEN\_TOO\_SMALL
- UEP\_FC\_RECORD\_LEN\_TOO\_LARGE
- UEP\_FC\_BUFFER\_LEN\_NOT\_FILE\_LEN
- UEP\_FC\_RECORD\_LEN\_NOT\_FILE\_LEN

#### **UEP\_FC\_DUPLICATE\_KEY\_CODE**

Adresse eines 1-Byte-Bereichs, der angibt, ob die Anforderung mehr als einen Datensatz für den angegebenen Schlüssel gefunden hat. Mögliche Werte:

- UEP\_FC\_DUPLICATE\_KEY
- UEP\_FC\_NOT\_DUPLICATE\_KEY

#### **UEP\_FC\_ACCMETH\_RETURN\_CODE**

Adresse eines 4-Byte-Bereichs, in dem Informationen in Abhängigkeit von der Zugriffsmethode zurückgegeben werden, wenn als Antwort entweder UEP\_FC\_REASON\_ACCMETH\_REQUEST\_ERROR oder UEP\_FC\_REASON\_IO\_ERROR zurückgegeben wird.

#### **UEP\_FC\_RESPONSE**

Adresse eines 1-Byte-Bereichs, der nach Abschluss der Anforderung die entsprechende Antwort enthält:

- UEP\_FC\_RESPONSE\_OK
- UEP\_FC\_RESPONSE\_EXCEPTION
- UEP\_FC\_RESPONSE\_DISASTER
- UEP\_FC\_RESPONSE\_INVALID
- UEP\_FC\_RESPONSE\_PURGED

#### **UEP\_FC\_REASON**

Adresse eines 1-Byte-Bereichs, der bei Abschluss der Anforderung mit einer Antwort vom Typ EXCEPTION die entsprechende Ursache enthält. Mögliche Ursachen:

- UEP\_FC\_REASON\_ABEND
- UEP\_FC\_REASON\_ACCMETH\_REQUEST\_ERROR
- UEP\_FC\_REASON\_BDAM\_DELETE
- UEP\_FC\_REASON\_BDAM\_LENGTH\_CHANGE
- UEP\_FC\_REASON\_BDAM\_KEY\_CONVERSION
- UEP\_FC\_REASON\_BDAM\_READ\_PREVIOUS
- UEP\_FC\_REASON\_BDAM\_WRITE\_MASS\_INSERT
- UEP\_FC\_REASON\_BROWSE\_UPD\_NOT\_RLS
- UEP\_FC\_REASON\_CACHE\_FAILURE
- UEP\_FC\_REASON\_CFDI\_CONNECT\_ERROR
- UEP\_FC\_REASON\_CFDI\_DISCONNECT\_ERROR
- UEP\_FC\_REASON\_CFDI\_INVALID\_CONTINUATION
- UEP\_FC\_REASON\_CFDI\_POOL\_FULL
- UEP\_FC\_REASON\_CFDI\_REOPEN\_ERROR
- UEP\_FC\_REASON\_CFDI\_SERVER\_NOT\_AVAILABLE
- UEP\_FC\_REASON\_CFDI\_SERVER\_NOT\_FOUND
- UEP\_FC\_REASON\_CFDI\_SYSIDERR
- UEP\_FC\_REASON\_CFDI\_TABLE\_GONE
- UEP\_FC\_REASON\_CHANGED

- UEP\_FC\_REASON\_CR\_NOT\_RLS
- UEP\_FC\_REASON\_DATASET\_BEING\_COPIED
- UEP\_FC\_REASON\_DEADLOCK\_DETECTED
- UEP\_FC\_REASON\_DELETE\_AFTER\_READ\_UPDATE
- UEP\_FC\_REASON\_DELETE\_BEFORE\_READ\_UPDATE
- UEP\_FC\_REASON\_DISASTER\_PERCOLATION
- UEP\_FC\_REASON\_DUPLICATE\_READ\_UPDATE
- UEP\_FC\_REASON\_DUPLICATE\_RECORD
- UEP\_FC\_REASON\_DUPLICATE\_REQID
- UEP\_FC\_REASON\_END\_OF\_FILE
- UEP\_FC\_REASON\_ESDS\_DELETE
- UEP\_FC\_REASON\_FILE\_DISABLED
- UEP\_FC\_REASON\_FILE\_NOT\_OPEN
- UEP\_FC\_REASON\_FILE\_NOT\_RECOVERABLE
- UEP\_FC\_REASON\_FILE\_NOT\_FOUND
- UEP\_FC\_REASON\_FULL\_KEY\_WRONG\_LENGTH
- UEP\_FC\_REASON\_GENERIC\_DELETE\_NOT\_KSDS
- UEP\_FC\_REASON\_GENERIC\_KEY\_TOO\_LONG
- UEP\_FC\_REASON\_ILLEGAL\_KEY\_TYPE\_CHANGE
- UEP\_FC\_REASON\_INSUFFICIENT\_SPACE
- UEP\_FC\_REASON\_INVALID\_UPDATE\_TOKEN
- UEP\_FC\_REASON\_IO\_ERROR
- UEP\_FC\_REASON\_ISCINVREQ
- UEP\_FC\_REASON\_ISC\_NOT\_SUPPORTED
- UEP\_FC\_REASON\_KEY\_LENGTH\_NEGATIVE
- UEP\_FC\_REASON\_KEY\_STOLEN
- UEP\_FC\_REASON\_KSDS\_AND\_XRBA
- UEP\_FC\_REASON\_LOADING
- UEP\_FC\_REASON\_LOCKED
- UEP\_FC\_REASON\_LOST\_LOCKS
- UEP\_FC\_REASON\_LOCK\_STRUCTURE\_FULL
- UEP\_FC\_REASON\_NOT\_IN\_SUBSET
- UEP\_FC\_REASON\_NO\_VARIABLE\_LENGTH
- UEP\_FC\_REASON\_NOSUSPEND\_NOT\_RLS
- UEP\_FC\_REASON\_NOTAUTH
- UEP\_FC\_REASON\_NOT\_EXTENDED
- UEP\_FC\_REASON\_PREVIOUS\_RLS\_FAILURE
- UEP\_FC\_REASON\_RBA\_ACCESS\_TO\_RLS\_KSDS
- UEP\_FC\_REASON\_READ\_NOT\_AUTHORISED
- UEP\_FC\_REASON\_READPREV\_IN\_GENERIC\_BROWSE
- UEP\_FC\_REASON\_RECLLEN\_EXCEEDS\_LOGGER\_BFSZ
- UEP\_FC\_REASON\_RECORD\_BUSY
- UEP\_FC\_REASON\_RECORD\_NOT\_FOUND
- UEP\_FC\_REASON\_REMOTE\_INVREQ
- UEP\_FC\_REASON\_RESTART\_FAILED
- UEP\_FC\_REASON\_REWRITE\_BEFORE\_READ\_UPDATE
- UEP\_FC\_REASON\_RIDFLD\_KEY\_NOT\_RECORD\_KEY
- UEP\_FC\_REASON\_RLS\_DEADLOCK\_DETECTED
- UEP\_FC\_REASON\_RLS\_DISABLED
- UEP\_FC\_REASON\_RLS\_FAILURE
- UEP\_FC\_REASON\_RR\_NOT\_RLS
- UEP\_FC\_REASON\_SECURITY\_FAILURE
- UEP\_FC\_REASON\_SELF\_DEADLOCK\_DETECTED
- UEP\_FC\_REASON\_SERVREQ\_VIOLATION
- UEP\_FC\_REASON\_SHIP
- UEP\_FC\_REASON\_SHIPPED\_SECURITY\_FAILURE

- UEP\_FC\_REASON\_STORE\_FAIL
- UEP\_FC\_REASON\_SUPPRESSED
- UEP\_FC\_REASON\_SYSIDERR
- UEP\_FC\_REASON\_TABLE\_FULL
- UEP\_FC\_REASON\_TABLE\_TOKEN\_INVALID
- UEP\_FC\_REASON\_TIMEOUT
- UEP\_FC\_REASON\_TOO\_MANY\_CFDTS\_IN\_UOW
- UEP\_FC\_REASON\_UNKNOWN\_REQID\_ENDBR
- UEP\_FC\_REASON\_UNKNOWN\_REQID\_READNEXT
- UEP\_FC\_REASON\_UNKNOWN\_REQID\_READPREV
- UEP\_FC\_REASON\_UNKNOWN\_REQID\_RESETBR
- UEP\_FC\_REASON\_UPDATE\_NOT\_AUTHORISED

#### **UEP\_FC\_EXIT\_TOKEN**

Adresse des 4-Byte-Tokens, das von XFCFRIN übergeben wurde.

#### **Rückgabecodes**

##### **UERCNORM**

Verarbeitung fortsetzen.

##### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

#### **XPI-Aufrufe**

Es können alle verwendet werden.

#### **API- und SPI-Aufrufe**

Es können keine verwendet werden.

---

## **Exits XFCREQ und XFCREQC für die Dateisteuerungs-API der EXEC-Schnittstelle**

Mit dem Exit XFCREQ können Sie eine Anforderung der Dateisteuerungs-API abfangen, bevor eine entsprechende Aktion von der Dateisteuerung ausgeführt wird. Der Exit XFCREQC ermöglicht es Ihnen wiederum, eine Anforderung der Dateisteuerungs-API abzufangen, nachdem die Dateisteuerung ihre Verarbeitung abgeschlossen hat.

### **Wichtig**

Bei Anforderungen, die über Funktionen übertragen werden, werden die Exits XFCREQ und XFCREQC in der Zielregion nicht aufgerufen. Dies bedeutet Folgendes: Wenn eine Anforderung der Dateisteuerungs-API über Funktionen an eine ferne Region übertragen wird, dann werden die Exits in der fernen Region nicht aufgerufen. Um eine über Funktionen übertragene Anforderung der Dateisteuerungs-API in der Zielregion abzufangen, müssen Sie den Exit XFCFRIN verwenden (siehe „Exits XFCFRIN und XFCFROUT für die Dateisteuerungsdomäne“ auf Seite 66).

Die folgenden Befehle der Dateisteuerungs-API werden abgefangen:

- READ
- WRITE
- REWRITE
- DELETE
- UNLOCK
- STARTBR
- READNEXT

- READPREV
- ENDBR
- RESETBR

Die Exits XFCREQ und XFCREQC können ausschließlich in der Assemblersprache geschrieben werden.

Mithilfe von XFCREQ können Sie folgende Aktionen ausführen:

- Analysieren der Anforderung zwecks Ermittlung ihres Typs sowie der angegebenen Schlüsselwörter und zugehörigen Werte.
- Ändern der in der Anforderung angegebenen Werte vor Ausführung des Befehls.
- Festlegen von Rückgabecodes, um Folgendes anzugeben:
  - CICS soll mit der (möglicherweise geänderten) Anforderung fortgesetzt werden.
  - CICS soll die Anforderung umgehen. (Hierbei ist Folgendes zu beachten: Wenn Sie diesen Rückgabecode festlegen, müssen Sie auch Rückgabecodes für den EXEC-Schnittstellenblock (EIB) festlegen, als hätten Sie die Anforderung selbst verarbeitet.)

Mithilfe von XFCREQC können Sie folgende Aktionen ausführen:

- Analysieren der Anforderung zwecks Ermittlung ihres Typs sowie der angegebenen Schlüsselwörter und zugehörigen Werte.
- Festlegen von Rückgabecodes für den EXEC-Schnittstellenblock (EIB).

An beide Exits werden neun Parameter wie folgt übergeben:

- Adresse der Parameterstruktur der Befehlsebene.
- Adresse eines Tokens (UEPFCTOK) zur Übergabe von 4 Byte Daten von XFCREQ an XFCREQC.
- Adressen von Kopien von vier Rückgabecodes und Ressourceninformationen vom EXEC-Schnittstellenblock (EIB).
- Adresse eines Tokens (UEPTSTOK), das während der gesamten Lebensdauer einer Task gültig ist.
- Adresse eines Zählerfelds für Rekursionen.
- Adresse eines 16-Byte-Bereichs, der verwendet wird, wenn die Anforderung über Funktionen übertragen worden ist.

## Parameterstruktur der Befehlsebene

Die Parameterstruktur der Befehlsebene besteht aus einer Reihe von Adressen. Die erste Adresse verweist auf den EXEC-Schnittstellendeskriptor (EID), der aus einer Bitfolge besteht, die den Typ der Anforderung beschreibt und jedes mit der Anforderung angegebene Schlüsselwort identifiziert. Die übrigen Adressen verweisen auf Teile von Daten, die der Anforderung zugeordnet sind. So verweist beispielsweise die zweite Adresse immer auf den Dateinamen.

Es können nur die ersten 8 Adressen und die letzte Adresse durch den Benutzerexit referenziert werden. Die neunte bis elfte Adresse sind für die interne Verwendung durch CICS reserviert.

Anhand des EXEC-Schnittstellendeskriptors können Sie den Typ der Anforderung sowie die angegebenen Schlüsselwörter ermitteln. Anhand der anderen Parameter in der Liste können Sie die Werte der Schlüsselwörter ermitteln. Sie können auch

Werte von Schlüsselwörtern ändern, die in der Anforderung angegeben sind. (So könnten Sie beispielsweise den in die Anforderung einbezogenen Namen der Datei ändern.)

## Indikator für Ende der Parameterliste

Das höchstwertige Bit wird für die letzte Adresse in der Parameterliste gesetzt, um anzuzeigen, dass es sich um die letzte Adresse in der Liste handelt. Bei der Rückkehr vom Benutzerexitprogramm sucht CICS in der Parameterliste nach dem höchstwertigen Bit, um den letzten Parameter ausfindig zu machen. Daher gilt: Wenn Sie die Länge der Parameterliste ändern, müssen Sie auch das höchstwertige Bit neu setzen, um so die aktuelle letzte Adresse zu markieren.

Die ursprüngliche Parameterliste wird nach Beendigung von XFCREQC in der Version wiederhergestellt, die vor dem Aufruf von XFCREQ vorlag. Daraus folgt, dass die Ausführungsdiagnosefunktion (Execution Diagnostic Facility, EDF) sowohl vor als auch nach der Ausführung jeweils den ursprünglichen Befehl anzeigt.

**Anmerkung:** EDF zeigt keine Änderungen an, die durch den Exit vorgenommen wurden.

## Exitspezifischer Parameter UEPCLPS

Der exitspezifische Parameter UEPCLPS ist die Adresse der Parameterstruktur der Befehlsebene und ist in den Exits XFCREQ und XFCREQC enthalten.

Die Parameterstruktur der Befehlsebene enthält 12 Adressen: FC\_ADDR0 bis FC\_ADDRB. Die Struktur wird im Pseudobereich (DSECT) FC\_ADDR\_LIST definiert, den Sie in Ihr Exitprogramm kopieren sollten, indem Sie die Anweisung COPY DFHFCEDS einschließen.

Die Parameterliste der Befehlsebene ist wie folgt aufgebaut:

### FC\_ADDR0

Adresse eines 9-Byte-Bereichs, der als EID (EXEC Interface Descriptor) bezeichnet wird und wie folgt aufgebaut ist:

- FC\_GROUP
- FC\_FUNCT
- FC\_BITS1
- FC\_BITS2
- FC\_EIDOPT5
- FC\_EIDOPT6
- FC\_EIDOPT7
- FC\_EIDOPT8

Der Name des Pseudobereichs (DSECT), der den EID zuordnet, lautet FC\_EID.

### FC\_GROUP

Immer X'06'; hierdurch wird angegeben, dass es sich um eine Dateisteueranforderung handelt.

### FC\_FUNCT

Ein Byte, das den Typ der Anforderung definiert:

- |       |         |
|-------|---------|
| X'02' | READ    |
| X'04' | WRITE   |
| X'06' | REWRITE |

X'08'	DELETE
X'0A'	UNLOCK
X'0C'	STARTBR
X'0E'	READNEXT
X'10'	READPREV
X'12'	ENDBR
X'14'	RESETBR

#### FC\_BITS1

Existenzbits zum Definieren der angegebenen Schlüsselwörter mit Werten. Um den Wert abzurufen, der einem Schlüsselwort zugeordnet ist, müssen Sie die entsprechende Adresse aus der Parameterstruktur der Befehlsebene verwenden. Bevor Sie diese Adresse verwenden, müssen Sie das zugehörige Existenzbit überprüfen. Wenn das Existenzbit inaktiviert ist, wurde das Schlüsselwort in der Anforderung nicht angegeben, und die Adresse sollte nicht verwendet werden.

X'80'	Wird gesetzt, wenn die Anforderung das Schlüsselwort FILE enthält. Ist dieses Existenzbit gesetzt, dann ist <b>FC_ADDR1</b> aussagekräftig.
X'40'	Wird gesetzt, wenn die Anforderung das Schlüsselwort INTO, SET oder FROM enthält. Ist dieses Existenzbit gesetzt, dann ist <b>FC_ADDR2</b> aussagekräftig.
X'20'	Wird gesetzt, wenn die Anforderung LENGTH oder NUMREC angibt, oder wenn REQID in einer Anforderung vom Typ STARTBR, RESETBR oder ENDBR angegeben wird. Ist dieses Existenzbit gesetzt, dann ist <b>FC_ADDR3</b> aussagekräftig.
X'10'	Wird gesetzt, wenn die Anforderung RIDFLD angibt. Ist dieses Existenzbit gesetzt, dann ist <b>FC_ADDR4</b> aussagekräftig.
X'08'	Wird gesetzt, wenn die Anforderung KEYLENGTH angibt. Ist dieses Existenzbit gesetzt, dann ist <b>FC_ADDR5</b> aussagekräftig.
X'04'	Wird gesetzt, wenn es sich bei der Anforderung um READNEXT oder READPREV handelt und REQID angegeben wird. Ist dieses Existenzbit gesetzt, dann ist <b>FC_ADDR6</b> aussagekräftig.
X'02'	Wird gesetzt, wenn die Anforderung SYSID angibt. Ist dieses Existenzbit gesetzt, dann ist <b>FC_ADDR7</b> aussagekräftig.
X'01'	Wird nicht von der Dateisteuerung verwendet.

#### FC\_BITS2

Zweite Gruppe von Existenzbits.

X'20'	Wird gesetzt, wenn die Anforderung TOKEN angibt. Ist dieses Existenzbit gesetzt, dann ist <b>FC_ADDRB</b> aussagekräftig.
-------	---

#### FC\_EIDOPT5

Gibt an, ob bestimmte Schlüsselwörter, die keine Werte annehmen, in der Anforderung angegeben wurden.

X'04'	MASSINSERT wurde angegeben.
X'02'	RRN wurde angegeben.

X'01' SET (und nicht INTO) wurde angegeben.

**Anmerkung:** Der Test auf das Vorhandensein von Schlüsselwörtern muss von Ihrem Programm auf Bitebene durchgeführt werden, da es sein kann, dass mehr als eines dieser Schlüsselwörter vorhanden ist.

#### FC\_EIDOPT6

Gibt an, ob bestimmte Schlüsselwörter, die keine Werte annehmen, in der Anforderung angegeben wurden.

X'80' RBA wurde angegeben.

X'40' GENERIC wurde angegeben.

X'20' GTEQ wurde angegeben.

X'10' UNCOMMITTED wurde angegeben.

X'08' CONSISTENT wurde angegeben.

X'04' REPEATABLE wurde angegeben.

X'01' NOSUSPEND wurde angegeben (in einer Anforderung vom Typ READ, READNEXT, READPREV, WRITE, DELETE oder REWRITE).

#### **Anmerkung:**

1. Wenn die Leseintegritätsbits (für UNCOMMITTED, CONSISTENT und REPEATABLE) im Befehl auf 'off' (null) gesetzt sind, dann werden die in der Dateiressourcendefinition angegebenen Leseintegritätsoptionen verwendet. Wenn Sie wissen müssen, um welche Optionen es sich hierbei handelt, können Sie den Befehl EXEC CICS INQUIRE FILE ausgeben.
2. Der Test auf das Vorhandensein von Schlüsselwörtern muss von Ihrem Programm auf Bitebene durchgeführt werden, da es sein kann, dass mehr als eines dieser Schlüsselwörter vorhanden ist.

#### FC\_EIDOPT7

Gibt an, ob bestimmte Schlüsselwörter, die keine Werte annehmen, in der Anforderung angegeben wurden.

X'04' UPDATE wurde angegeben. Diese Einstellung ist nur für Leseanforderungen (READ) aussagekräftig. Bei anderen Anforderungen kann X'04' gesetzt werden oder auch nicht.

X'01' Entweder DEBREC oder DEBKEY wurde angegeben (siehe FC\_EIDOPT8). Diese Einstellung ist nur für Leseanforderungen (READ) aussagekräftig. Bei anderen Anforderungen kann X'01' gesetzt werden oder auch nicht.

**Anmerkung:** Der Test auf das Vorhandensein von Schlüsselwörtern muss von Ihrem Programm auf Bitebene durchgeführt werden, da es sein kann, dass mehr als eines dieser Schlüsselwörter vorhanden ist.

#### FC\_EIDOPT8

Gibt an, ob bestimmte Schlüsselwörter, die keine Werte annehmen, in der Anforderung angegeben wurden.

X'80' DEBKEY wurde angegeben.

X'40' DEBREC wurde angegeben.

X'20' TOKEN wurde angegeben.

**X'08'** XRBA wurde angegeben. Ist das Bit XRBA aktiviert, dann zeigt FC\_RIDFLD (im Pseudobereich (DSECT) DFHFCEDS beschrieben) auf eine erweiterte relative Byteadresse (XRBA) von 8 Byte.

#### **FC\_ADDR1**

Adresse eines 8-Byte-Bereichs, der den im Schlüsselwort FILE angegebenen Namen enthält.

#### **FC\_ADDR2**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse einer 4-Byte-Adresse, die für SET zurückgegeben wurde (wenn die Anforderung READ, READNEXT oder READPREV ist und **FC\_EIDOPT5** anzeigt, dass es sich hierbei um SET handelt).
- Adresse der Daten, die für INTO zurückgegeben wurden (wenn die Anforderung READ, READNEXT oder READPREV ist und wenn **FC\_EIDOPT5** anzeigt, dass es sich hierbei nicht um SET handelt).
- Adresse der Daten aus FROM (wenn die Anforderung WRITE oder REWRITE ist).

#### **FC\_ADDR3**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse des Halbwortwerts für LENGTH (wenn die Anforderung READ, WRITE, REWRITE, READNEXT oder READPREV ist).  
**Warnung:** Bei Anforderungen mit Angabe von INTO darf der Wert für LENGTH nicht in einen Wert geändert werden, der größer ist als der von der Anwendung angegebene Wert. Wird ein größerer Wert angegeben, führt dies zu einer Speicherüberlagerung in der Anwendung.
- Adresse des zurückgegebenen Halbwortwerts für NUMREC (wenn die Anforderung DELETE ist).
- Adresse des Halbwortwerts für REQID (wenn die Anforderung STARTBR, RESETBR oder ENDBR ist).

#### **FC\_ADDR4**

Adresse eines Bereichs, der den Wert des Schlüsselworts RIDFLD enthält.

#### **FC\_ADDR5**

Adresse des Halbwortwerts für KEYLENGTH.

#### **FC\_ADDR6**

Adresse des Halbwortwerts für REQID (wenn die Anforderung READNEXT oder READPREV ist).

#### **FC\_ADDR7**

Adresse eines Bereichs, der den Wert für SYSID enthält.

#### **FC\_ADDR8**

Adresse eines Werts, der nur zur internen Verwendung durch CICS bestimmt ist. Dieser Wert darf nicht verwendet werden.

#### **FC\_ADDR9**

Adresse eines Werts, der nur zur internen Verwendung durch CICS bestimmt ist. Dieser Wert darf nicht verwendet werden.

#### **FC\_ADDRA**

Adresse eines Werts, der nur zur internen Verwendung durch CICS bestimmt ist. Dieser Wert darf nicht verwendet werden.



#### FC\_ADDRB

Adresse des Vollwortwerts für TOKEN (wenn die Anforderung READ, READ-NEXT, READPREV, REWRITE, DELETE oder UNLOCK ist).

## Ändern von Feldern in der Parameterstruktur der Befehlsebene

Von den Feldern, die an die Dateisteuerung übergeben werden, werden einige als Eingabefelder für die Anforderung, andere als Ausgabefelder und wieder andere sowohl als Eingabe- als auch als Ausgabefelder verwendet. Die Methode, die Ihr Benutzerexitprogramm zum Ändern eines Felds verwendet, hängt von der Verwendung des betreffenden Felds ab.

### Liste der Eingabe- und Ausgabefelder

Die folgenden Felder sind immer Eingabefelder:

- FILE
- FROM
- KEYLENGTH
- REQID
- SYSID

Die folgenden Felder sind immer Ausgabefelder:

- INTO
- NUMREC
- SET

Ob es sich bei LENGTH und RIDFLD um Eingabe- oder Ausgabefelder handelt, hängt von der Anforderung ab (siehe Tabelle 4). Ein Bindestrich (—) bedeutet, dass das Schlüsselwort in der Anforderung nicht angegeben werden kann.

Tabelle 4. LENGTH und RIDFLD als Eingabe- und Ausgabefelder

Anforderung	LENGTH	RIDFLD
READ	Ausgabe	Siehe Anmerkung 1.
WRITE	Eingabe	Siehe Anmerkung 2.
REWRITE	Eingabe	—
DELETE	—	Siehe Anmerkung 3 auf Seite 88.
UNLOCK	—	—
STARTBR	—	Eingabe
READNEXT	Ausgabe	Ausgabe
READPREV	Ausgabe	Ausgabe
ENDBR	—	—
RESETBR	—	Eingabe

#### Anmerkung:

1. Normalerweise handelt es sich hierbei um ein Eingabefeld. Wenn jedoch UPDATE angegeben ist und es sich bei der Datei um eine BDAM-Datei mit erweiterter Schlüsselsuche handelt, wird RIDFLD sowohl für die Eingabe als auch für die Ausgabe verwendet.
2. Die Verwendung von RIDFLD in einer WRITE-Anforderung hängt vom Dateityp ab. Bei einer Datei vom Typ VSAM KSDS oder VSAM RRDS oder einer

BDAM-Datei mit festem Format ist RIDFLD ein Eingabefeld. Bei allen anderen Dateitypen wird das Feld entweder ausschließlich für die Ausgabe oder sowohl für die Eingabe als auch für die Ausgabe verwendet und sollte wie ein Ausgabefeld behandelt werden.

- Bei DELETE-Anforderungen ohne vorangestelltes READ UPDATE ist RIDFLD ein Eingabefeld. Bei Anforderungen mit vorangestelltem READ UPDATE wird das Feld nicht angegeben.

### Ändern von Eingabefeldern

Die richtige Methode zum Ändern eines Eingabefelds besteht darin, eine neue Kopie desselben zu erstellen und die Adresse in der Parameterliste der Befehlsebene so zu ändern, dass sie auf die neuen Daten verweist.

**Anmerkung:** Sie dürfen ein Eingabefeld niemals ändern, indem Sie die Daten ändern, auf die durch die Parameterliste der Befehlsebene verwiesen wird. Dies würde zu einer Beschädigung von Speicher führen, der zum Anwendungsprogramm gehört; dies wiederum würde zu einem Fehler führen, wenn das Programm versucht, das Feld wiederzuverwenden.

### Ändern von Ausgabefeldern

Das unter „Ändern von Eingabefeldern“ beschriebene Verfahren ist für das Ändern von Ausgabefeldern nicht geeignet. (Die Ergebnisse würden an den neuen Bereich und nicht an den Bereich der Anwendung zurückgegeben und wären daher für die Anwendung nicht sichtbar.)

Ein Ausgabefeld wird geändert, indem die Daten geändert werden, auf die von der Parameterliste der Befehlsebene verwiesen wird. Bei einem Ausgabefeld können Sie die vorhandenen Anwendungsdaten ändern, da die Anwendung ohnehin erwartet, dass das Feld geändert wird.

### Ändern von Feldern für Eingabe und Ausgabe

Ein Beispiel eines Felds, das sowohl für Eingabe als auch für Ausgabe verwendet wird, ist das Längensfeld LENGTH in einer Anforderung vom Typ READ mit Angabe von INTO. Derartige Felder können in derselben Weise wie Ausgabefelder behandelt werden, und sie werden auch als solche angesehen.

## Ändern des EXEC-Schnittstellendesktors (EID)

Es ist nicht möglich, den EXEC-Schnittstellendesktors (EXEC Interface Descriptor, EID) zu ändern, um bedeutende Änderungen an Anforderungen vorzunehmen. So ist es beispielsweise nicht möglich, eine Anforderung vom Typ WRITE in eine Anforderung vom Typ READ zu ändern. Es können jedoch geringfügige Änderungen an Anforderungen vorgenommen werden. So ist es beispielsweise möglich, das Existenzbit für die System-ID (SYSID) zu aktivieren, sodass die betreffende Anforderung in eine Anforderung geändert werden kann, die an ein fernes System übertragen wird.

In der nachstehenden Liste werden diejenigen Bits im EID aufgeführt, die geändert werden können. Versuche, andere Bestandteile des EID zu ändern, werden ignoriert.

#### FC\_BITS1

- X'20'** Existenzbit für LENGTH, NUMREC oder (wenn die Anforderung vom Typ STARTBR, RESETBR oder ENDBR ist) REQID.
- X'08'** Existenzbit für KEYLENGTH.

X'04' Existenzbit für REQID (wenn die Anforderung vom Typ READNEXT oder READPREV ist).

X'02' Existenzbit für SYSID.

#### FC\_BITS2

X'20' Token wurde angegeben.

#### FC\_EIDOPT5

X'04' MASSINSERT wurde angegeben.

#### FC\_EIDOPT6

X'40' GENERIC wurde angegeben.

X'20' GTEQ wurde angegeben.

X'10' UNCOMMITTED wurde angegeben.

X'08' CONSISTENT wurde angegeben.

X'04' REPEATABLE wurde angegeben.

X'02' UPDATE wurde in READNEXT oder READPREV angegeben.

X'01' NOSUSPEND wurde angegeben (in einer Anforderung vom Typ READ, READNEXT, READPREV, WRITE, DELETE oder REWRITE).

Bits im EID sollten an Ort und Stelle geändert werden. Es sollte nicht der Verweis auf den EID geändert werden: Jeder Versuch, dies zu tun, wird von CICS ignoriert.

Der EID wird vor der Rückkehr zum Anwendungsprogramm auf seinen ursprünglichen Wert zurückgesetzt. Dies bedeutet, dass Änderungen, die am EID vorgenommen werden, nur für die Dauer der Dateisteueranforderung beibehalten werden.

Wenn UNCOMMITTED, CONSISTENT bzw. REPEATABLE nicht allein, sondern in irgendeiner Kombination angegeben sind, dann gilt Folgendes: CONSISTENT hat Vorrang vor UNCOMMITTED, und REPEATABLE hat Vorrang vor CONSISTENT und UNCOMMITTED.

### Beispiel für das Ändern von Leseintegritätsbits

In bestimmten Fällen könnte es ein, dass für sämtliche RLS-Leseanforderungen (READ-Anforderungen) von allen Programmen für eine bestimmte Datei CONSISTENT in der Leseanforderung angegeben werden soll. Sie könnten hierfür ein Benutzerexitprogramm codieren, welches in allen Anforderungen für die Datei das Bit für CONSISTENT aktiviert und die beiden anderen Leseintegritätsbits inaktiviert. Sie könnten diesen Effekt auch teilweise dadurch erreichen, dass Sie CONSISTENT in der Dateidefinition FILE angeben. Allerdings würden auf diese Weise lediglich Anforderungen überschrieben werden, in denen nicht explizit eine Leseintegritätsstufe angegeben wurde. Bei Verwendung eines globalen Benutzerexitprogramms für diesen Zweck werden hingegen auch Programme überschrieben, die explizit UNCOMMITTED oder REPEATABLE angeben.

#### Warnhinweise:

1. Wenn ein globales Benutzerexitprogramm eine Dateianforderung ändert, um eine höhere Stufe der Leseintegrität anzufordern (beispielsweise indem UNCOMMITTED in REPEATABLE geändert wird), könnte dies dazu führen, dass CICS entweder zusätzliche Lesesperren anfordert oder die vorhandenen Lesesperren über einen längeren Zeitraum beibehält. Wenn aus diesem Grund ande-

re Transaktionen warten müssen oder Deadlock auftreten, kann dies zu einer Verminderung des Systemdurchsatzes führen.

2. Wenn ein globales Benutzerexitprogramm eine Anforderung ändert, um eine niedrigere Leseintegritätsstufe anzufordern (beispielsweise indem REPEATABLE in UNCOMMITTED geändert wird), könnte dies dazu führen, dass Anwendungslogikfehler in dem Programm auftreten, aus dem die Anforderung stammt. Solche Fehler können auftreten, wenn das Anwendungsprogramm darauf angewiesen ist, dass ein Datensatz unverändert bleibt, während das Programm eine Reihe von anderen zugehörigen Datensätzen liest. Dies kann mit REPEATABLE garantiert werden, aber nicht, wenn die Option in UNCOMMITTED geändert wird.
3. Das Benutzerexitprogramm wird zwar daran gehindert, bedeutende Änderungen am EXEC-Schnittstellendeskriptor (EID) vorzunehmen. Dennoch müssen Sie auch bei den geringfügigen Änderungen, die **zulässig sind**, sehr sorgfältig vorgehen. So ist es beispielsweise möglich, eine Anforderung vom Typ DELETE in eine Anforderung vom Typ GENERIC DELETE zu ändern. Eine solche Änderung kann jedoch gefährlich sein.

## Verwendung des Parameters UEPFSHIP

UEPFSHIP enthält die Adresse eines 16-Byte-Bereichs. Dieser Bereich besteht aus 4 Zeichen, gefolgt von 3 Vollwörtern.

Wenn das erste Byte 'Y' enthält, dann wurde die betreffende Anforderung mittels Funktionen an diese Region übertragen. In diesem Fall gilt: Wenn Ihr Exitprogramm die Dateisteuerung umgehen will (indem es den Rückgabecode UERCBYP setzt), dann muss es die drei Vollwörter wie folgt festlegen:

### Vollwort 1

Die Länge des Pufferbereichs.

### Vollwort 2

Die Länge des Datensatzes.

### Vollwort 3

Die Länge des geänderten Werts für RIDFLD.

Dadurch wird sichergestellt, dass die Daten und RIDFLD ordnungsgemäß zurückübertragen werden.

## EIB (EXEC-Schnittstellenblock)

Es werden Kopien von EIBRSRCE, EIBRCODE, EIBRESP und EIBRESP2 an den Exit übergeben, sodass Sie die Ausführungs- und Ressourceninformationen in XF-CREQ und XF-CREQC ändern/festlegen und in XF-CREQC untersuchen können.

Sie können die Kopien von EIBRSRCE, EIBRCODE, EIBRESP und EIBRESP2, die Sie in der Parameterliste erhalten, aktualisieren. Die Dateisteuerung kopiert Ihre Werte in den realen EIB, nachdem der Exit XF-CREQC ausgeführt wurde oder wenn Sie den Rückgabecode für Umgehen in XF-CREQ angeben.

Sie müssen gültige Antworten für die Dateisteuerung festlegen. Sie müssen für die drei Codes EIBRCODE, EIBRESP und EIBRESP2 eine konsistente Gruppe von Werten festlegen, so wie sie auch von der Dateisteuerung festgelegt würden, um eine gültige Ausführung zu beschreiben. **Die Konsistenz der Werte für EIBRCODE, EIBRESP und EIBRESP2 wird von der Dateisteuerung nicht überprüft.** Als Hilfestellung beim Festlegen von Werten für EIBRCODE, EIBRESP und EIBRESP2 kön-

nen Sie die im Pseudobereich (DSECT) DFHFCEDS angegebenen Werte heranziehen, die von der Dateisteuerung verwendet werden.

## Beispiel für die Verwendung von XFCREQ und XFCREQC

In diesem Beispiel werden XFCREQ und XFCREQC verwendet, um einen Datensatz mit komprimierten Daten abzurufen, diese Daten zu dekomprimieren und anschließend an den Bereich zurückzugeben, der vom Benutzerprogramm mithilfe von INTO angegeben wird. Das Beispiel veranschaulicht lediglich das Leistungsspektrum der Exits; es ist nicht beabsichtigt aufzuzeigen, wie diese Funktionalität bestmöglich erreicht werden kann.

### Verarbeitungsschritte in XFCREQ:

1. Geben Sie den Befehl EXEC CICS GETMAIN aus, um einen Bereich abzurufen, der groß genug ist, um die dekomprimierten Daten zu speichern.
2. Ändern Sie den Wert für den Verweis INTO in diesen neuen Bereich, damit die Dateisteuerung diesen Bereich bei der Verarbeitung der Anforderung verwendet. (Die dekomprimierten Daten werden in den mit INTO angegebenen Bereich des *Benutzers* kopiert. Anschließend wird der Verweis INTO vor der Rückkehr zum Anwendungsprogramm zurückgesetzt. Siehe hierzu Schritt 4 und 7 der von XFCREQC auszuführenden Verarbeitung.)
3. Setzen Sie UEPFCTOK auf die Adresse des neuen Bereichs, damit XFCREQC diesen Bereich ebenfalls verwenden kann.
4. Kehren Sie zu CICS zurück.

### Verarbeitungsschritte in XFCREQC:

1. Überprüfen Sie UEPRCODE, um sicherzustellen, dass die Dateisteuerungsanforderung ohne Fehler abgeschlossen wurde.
2. Ermitteln Sie mithilfe von UEPFCTOK die Adresse des Bereichs. Dieser Bereich enthält nun die komprimierten Daten.
3. Dekomprimieren Sie die dort enthaltenen Daten.
4. Kopieren Sie die Daten aus dem neuen Bereich in den mit INTO angegebenen Bereich des Benutzers. Verwenden Sie die benutzerdefinierte Länge im Parameter LENGTH (aus der Parameterliste der Befehlsebene), um sicherzustellen, dass die Daten in den Bereich hineinpassen und die Kopie keinen Fehler im Speicher verursacht.
5. Legen Sie LENGERR in UEPRESP, UEPRESP2 und UEPRCODE für den Fall fest, dass die Daten nicht in den Bereich hineinpassen.
6. Verwenden Sie den Befehl EXEC CICS FREEMAIN, um den Arbeitsbereich freizugeben, auf den von UEPFCTOK verwiesen wird.
7. An dieser Stelle verweist die Parameterliste der Befehlsebene auf den jetzt freien Bereich als Adresse für INTO. Dies stellt kein Problem dar, weil dieser Verweis nach Beendigung von XFCREQC von der Dateisteuerung entsprechend zurückgesetzt wird, um wieder auf den Bereich zu verweisen, der vom Benutzerprogramm angegeben wurde.
8. Kehren Sie zu CICS zurück.

## Exit XFCREQ

Der Exit XFCREQ wird aufgerufen, bevor CICS eine API-Anforderung für die Dateisteuerung verarbeitet. Bei Anforderungen, die über Funktionen übertragen werden, wird der Exit in der Zielregion nicht aufgerufen.

## Exitspezifische Parameter

### UEPCLPS

Adresse der Parameterstruktur der Befehlsebene. Siehe „Exitspezifischer Parameter UEPCLPS“ auf Seite 83.

### UEPFCTOK

Adresse des 4-Byte-Tokens, das an XFCREQC übergeben werden soll. Dies ermöglicht es Ihnen beispielsweise, einen Arbeitsbereich an den Exit XFCREQC zu übergeben.

### UEPRCODE

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

### UEPRES

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

### UEPRES2

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

### UEPTSTOK

Adresse eines 4-Byte-Tokens, das während des gesamten Lebenszyklus einer Task gültig ist. Siehe Using the task token UEPTSTOK.

### UEPRECUR

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

### UEPFSHIP

Adresse eines 16-Byte-Bereichs. Siehe „Verwendung des Parameters UEPFSHIP“ auf Seite 90.

### UEPRSRCE

Adresse einer 8 Zeichen langen Kopie des EIB-Ressourcenwerts, EIBRSRCE.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

### UERCBYP

Das Dateisteuerprogramm der EXEC-Schnittstelle sollte diese Anforderung ignorieren.

### UERCPURG

Task während des XPI-Aufrufs bereinigt.

## XPI-Aufrufe

Es können alle verwendet werden.

Wenngleich der Exit die Verwendung der XPI-Aufrufe GETMAIN und FREEMAIN zulässt, wird empfohlen, stattdessen die Befehle EXEC CICS GETMAIN und EXEC CICS FREEMAIN zu verwenden.

## API- und SPI-Aufrufe

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:  
EXEC CICS SHUTDOWN  
EXEC CICS XCTL

**Anmerkung:**

1. Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor, um keine Schleife zu verursachen. So sind Sie beispielsweise dafür verantwortlich, das Auftreten einer Schleife zu vermeiden, wenn eine Dateisteueranforderung über den Exit XFCREQ ausgegeben wird. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.
2. Exitprogramme, die EXEC CICS-Befehle ausgeben, müssen zunächst den EXEC-Schnittstellenblock (EIB) adressieren. Siehe Using CICS services.
3. Exitprogramme, die EXEC CICS-Befehle ausgeben und das Makro DFHEIENT verwenden, sollten das Makro DFHEIRET verwenden, um einen Rückgabecode festzulegen und zu CICS zurückzukehren. Siehe Global user exit XPI examples, showing the use of storage.

## Exit XFCREQC

Der Exit XFCREQC wird nach Abschluss einer API-Anforderung für die Dateisteuerung und vor der Rückkehr vom Dateisteuerprogramm der EXEC-Schnittstelle aufgerufen. Bei Anforderungen, die über Funktionen übertragen werden, wird der Exit in der Zielregion nicht aufgerufen.

**Exitspezifische Parameter****UEPCLPS**

Adresse der Parameterstruktur der Befehlsebene. Siehe „Exitspezifischer Parameter UEPCLPS“ auf Seite 83.

**UEPFCTOK**

Adresse des 4-Byte-Tokens, das von XFCREQ übergeben wurde.

**UEPRCODE**

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

**UEPRES**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

**Anmerkung:** Wenn es sich bei der Datei, auf die soeben zugegriffen wurde, um eine ferne Datei handelt, enthält das adressierte Feld Nullen (selbst wenn UEPRCODE ungleich null ist).

**UEPRES2**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

**Anmerkung:** Wenn es sich bei der Datei, auf die soeben zugegriffen wurde, um eine ferne Datei handelt, enthält das adressierte Feld Nullen (selbst wenn UEPRCODE ungleich null ist).

**UEPTSTOK**

Adresse eines 4-Byte-Tokens, das während des gesamten Lebenszyklus einer Task gültig ist. Siehe Using the task token UEPTSTOK.

**UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

#### **UEPRSRCE**

Adresse einer 8 Zeichen langen Kopie des EIB-Ressourcenwerts, EIBRSRCE.

#### **Rückgabecodes**

##### **UERCNORM**

Verarbeitung fortsetzen.

##### **UERCPUrg**

Task während des XPI-Aufrufs bereinigt.

#### **XPI-Aufrufe**

Es können alle verwendet werden.

Wenngleich der Exit die Verwendung der XPI-Aufrufe GETMAIN und FREEMAIN zulässt, wird empfohlen, stattdessen die Befehle EXEC CICS GETMAIN und EXEC CICS FREEMAIN zu verwenden.

#### **API- und SPI-Aufrufe**

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

EXEC CICS SHUTDOWN  
EXEC CICS XCTL

#### **Anmerkung:**

1. Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor, um keine Schleife zu verursachen. So sind Sie beispielsweise dafür verantwortlich, das Auftreten einer Schleife zu vermeiden, wenn eine Dateisteueranforderung über den Exit XFCREQC ausgegeben wird. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.
2. Exitprogramme, die EXEC CICS-Befehle ausgeben, müssen zunächst den EXEC-Schnittstellenblock (EIB) adressieren. Siehe Using CICS services.
3. Exitprogramme, die EXEC CICS-Befehle ausgeben und das Makro DFHEIENT verwenden, sollten das Makro DFHEIRET verwenden, um einen Rückgabecode festzulegen und zu CICS zurückzukehren. Siehe Global user exit XPI examples, showing the use of storage.

#### **Beispielprogramm**

CICS stellt in CICSTS55.CICS.SDFHSAMP ein Beispielprogramm namens DFH\$XTSE bereit, das veranschaulicht, wie Felder in der Parameterstruktur der Befehlsebene, die an EXEC-Schnittstellenexits übergeben wird, geändert werden.

---

## **Exits XFCAREQ und XFCAREQC für die Dateisteuerungs-SPI der EXEC-Schnittstelle**

Mit dem Exit XFCAREQ können Sie eine Anforderung der Dateisteuerungs-SPI (Systemprogrammierschnittstelle) abfangen, bevor eine entsprechende Aktion von der Dateisteuerung ausgeführt wird. Mithilfe des Exits XFCAREQC können Sie die entsprechende Antwort abfangen, nachdem eine Anforderung der Dateisteuerungs-SPI abgeschlossen worden ist.

Die folgenden Anforderungen der Dateisteuerung-SPI werden abgefangen:

- **EXEC CICS INQUIRE FILE.**
- **EXEC CICS SET FILE.**

Mithilfe von XFCAREQ können Sie folgende Aktionen ausführen:



- Analysieren der SPI-Parameterliste (Funktion, Schlüsselwörter, Argumentwerte und Antworten).
- Ändern beliebiger Eingabeparameter vor Ausführung einer Anforderung.
- Verhindern der Ausführung einer Anforderung und Festlegen entsprechender Antworten.

Mithilfe von XFCAREQC können Sie folgende Aktionen ausführen:

- Analysieren der SPI-Parameterliste.
- Ändern beliebiger Ausgabeparameterwerte und Festlegen von Antworten nach der Ausführung.

Darüber hinaus ist Folgendes möglich:

- Übergeben von Daten zwischen den Exitprogrammen für XFCAREQ und XFCAREQC, wenn diese Programme für dieselbe Anforderung aufgerufen werden.
- Übergeben von Daten zwischen den Exitprogrammen für Dateisteuerung, wenn diese Programme innerhalb derselben Task aufgerufen werden. Sie können Daten zwischen aufeinanderfolgenden Aufrufen von XFCAREQ und XFCAREQC übergeben sowie auch zwischen Aufrufen von anderen für EXEC aktivierten Benutzerexits.

Wenn Sie Änderungen an den Dateistatus vornehmen (d. h. wenn Sie eine Datei öffnen, schließen, aktivieren oder inaktivieren), kann es möglicherweise vorkommen, dass die von XFCAREQ definierten Situationen von den Exits im Dateistatusänderungsprogramm (XFCSREQ und XFCSREQC) geändert werden. Daher müssen Sie die Reihenfolge berücksichtigen, in der die Exits aufgerufen werden. Wenn alle vier Exits aktiviert sind, lautet die Reihenfolge für das Aufrufen der Exits wie folgt:

- Für den Befehl **SET FILE**:
  1. XFCAREQ
  2. XFCSREQ
  3. XFCSREQC
  4. XFCAREQC
- Für den Befehl **INQUIRE FILE**:
  1. XFCAREQ
  2. XFCAREQC

## Exit XFCAREQ

Dieser Exit wird aufgerufen, bevor CICS eine SPI-Anforderung für die Dateisteuerung verarbeitet.

### Zeitpunkt des Aufrufs

Bevor CICS eine SPI-Anforderung für die Dateisteuerung verarbeitet.

### Exitspezifische Parameter

#### UEPCLPS

Adresse einer Kopie der SPI-Befehlsparameterliste. Siehe „Parameterstruktur der Befehlsebene“ auf Seite 98.

#### UEPFATOK

Adresse eines 4-Byte-Bereichs, der verwendet werden kann, um in einer einzelnen SPI-Anforderung für Dateisteuerung Informationen zwischen XFCAREQ und XFCAREQC zu übergeben.

**UEPRCODE**

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

**UEPRES**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

**UEPRES2**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

**UEPTSTOK**

Adresse eines 4-Byte-Tokens, das verwendet werden kann, um innerhalb derselben Task Informationen zwischen aufeinanderfolgenden Dateisteueranforderungen zu übergeben (beispielsweise zwischen aufeinanderfolgenden Aufrufen des Exits XFCAREQC). Siehe Using the task token UEPTSTOK.

**UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf null gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

**Rückgabecodes****UERCBY**

Diese Anforderung umgehen.

**UERCNORM**

Verarbeitung fortsetzen.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Befehle**

Es können alle verwendet werden.

**API- und SPI-Befehle**

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

EXEC CICS SHUTDOWN

EXEC CICS XCTL

**Anmerkung:** Gehen Sie beim Verwenden von rekursiven Befehlen mit Sorgfalt vor. So müssen Sie beispielsweise den Eintritt in eine Schleife vermeiden, wenn Sie eine SPI-Anforderung für die Dateisteuerung über den Exit XFCAREQC ausgeben. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.

**Exit XFCAREQC**

Der Exit XFCAREQC wird nach Abschluss einer SPI-Anforderung für die Dateisteuerung und vor der Rückkehr vom Dateisteuerprogramm der EXEC-Schnittstelle für SPI aufgerufen.

**Exitspezifische Parameter:****UEPCLPS**

Adresse einer Kopie der API-Befehlsparameterliste. Siehe „Parameterstruktur der Befehlsebene“ auf Seite 98.

**UEPFATOK**

Adresse eines 4-Byte-Bereichs, der verwendet werden kann, um in einer einzelnen SPI-Anforderung für Dateisteuerung Informationen zwischen XFCAREQ und XFCAREQC zu übergeben.

**UEPRCODE**

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

**UEPRES**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

**UEPRES2**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

**UEPTSTOK**

Adresse eines 4-Byte-Tokens, das verwendet werden kann, um innerhalb derselben Task Informationen zwischen aufeinanderfolgenden Dateisteueranforderungen zu übergeben (beispielsweise zwischen aufeinanderfolgenden Aufrufen des Exits XFCAREQC). Siehe Using the task token UEPTSTOK.

**UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf null gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Befehle**

Es können alle verwendet werden.

**API- und SPI-Befehle**

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

EXEC CICS SHUTDOWN  
EXEC CICS XCTL

Sie können die Kopien von EIBRCODE, EIBRESP und EIBRESP2, die Sie in der Parameterliste erhalten, aktualisieren. Wenn Sie die Werte aktualisieren, kopiert die Dateisteuerung die neuen Werte in den EXEC-Schnittstellenblock (EIB) des Anwendungsprogramms, nachdem XFCAREQC beendet wurde oder wenn Sie den Rückgabecode UERCBYP in XFCAREQ angeben.

Sie müssen gültige Antworten für die Dateisteuerung festlegen. Sie müssen für alle drei Codes (EIBRCODE, EIBRESP und EIBRESP2) eine konsistente Gruppe von Werten festlegen, so wie sie auch von der Dateisteuerung festgelegt würden, um einen gültigen Abschluss zu beschreiben. Die Konsistenz der von Ihnen festgelegten Werte wird von CICS nicht überprüft. Wenn EIBRCODE auf einen Wert ungleich null gesetzt wird und EIBRESP auf null gesetzt wird, überschreibt CICS diesen Wert für EIBRESP durch einen Wert, der ungleich null ist. Als Hilfestellung beim Festlegen von Werten für EIBRCODE, EIBRESP und EIBRESP2 können Sie die

im Pseudobereich (DSECT) DFHFAUED angegebenen Werte heranziehen, die von der Dateisteuerung für SPI-Anforderungen verwendet werden.

**Anmerkung:** Gehen Sie beim Verwenden von rekursiven Befehlen mit Sorgfalt vor. So müssen Sie beispielsweise den Eintritt in eine Schleife vermeiden, wenn Sie eine SPI-Anforderung für die Dateisteuerung über den Exit XFCAREQ ausgeben. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.

## Parameterstruktur der Befehlsebene

Die Parameterstruktur der Befehlsebene besteht aus einer Reihe von Adressen. Die erste Adresse verweist auf den EXEC-Schnittstellendeskriptor (EID), der aus einer Bitfolge besteht, die den Typ der Anforderung beschreibt und jedes mit der Anforderung angegebene Schlüsselwort identifiziert. Die übrigen Adressen verweisen auf Teile von Daten, die der Anforderung zugeordnet sind.

Anhand des EXEC-Schnittstellendeskriptors können Sie den Typ der Anforderung sowie die angegebenen Schlüsselwörter ermitteln. Anhand der anderen Parameter in der Liste können Sie die Werte der Schlüsselwörter ermitteln. Sie können auch Werte von Schlüsselwörtern ändern, die in der Anforderung angegeben sind.

**Anmerkung:** Die Beziehungen zwischen Argumenten, Schlüsselwörtern, Datentypen und den Typen 'Eingabe/Ausgabe' werden in den folgenden Tabellen für die SPI-Befehle zur Dateisteuerung zusammengefasst:

- Für INQUIRE FILE: in Tabelle 5 auf Seite 106.
- Für SET FILE: in Tabelle 6 auf Seite 108.

### Exitspezifischer Parameter UEPCLPS

Der exitspezifische Parameter UEPCLPS wird sowohl an XFCAREQ als auch an XFCAREQC übergeben; er enthält die Adresse der Parameterstruktur der Befehlsebene.

Die Parameterliste der Befehlsebene enthält 64 Adressen: FCIS\_ADDR0 bis FCIS\_ADDR63. Diese Adressen werden im Pseudobereich (DSECT) DFHFAUED beschrieben. Sie sollten DFHFAUED in Ihr Programm kopieren, indem Sie die Anweisung COPY DFHFAUED einschließen.

Die Parameterliste der Befehlsebene ist wie folgt aufgebaut:

#### FCIS\_ADDR0

Adresse eines 13-Byte-Bereichs, der als EID (EXED Interface Descriptor) bezeichnet wird und wie folgt aufgebaut ist:

- FCIS\_GROUP
- FCIS\_FUNCT
- FCIS\_EIDOPT2
- FCIS\_EIDOPT3
- FCIS\_EIDOPT4
- FCIS\_BITS1
- FCIS\_BITS2
- FCIS\_BITS3
- FCIS\_BITS4
- FCIS\_BITS5
- FCIS\_BITS6

- FCIS\_BITS7
- FCIS\_BITS8

#### FCIS\_GROUP

Immer X'4C'; hierdurch wird angegeben, dass es sich um eine SPI-Anforderung für die Dateisteuerung handelt.

#### FCIS\_FUNCT

Ein Byte, das den Typ der Anforderung definiert:

X'02' INQUIRE FILE.

X'04' SET FILE.

#### FCIS\_EIDOPT2

Wird nicht von der Dateisteuerung verwendet.

#### FCIS\_EIDOPT3

Wird nicht von der Dateisteuerung verwendet.

#### FCIS\_EIDOPT4

Wird nicht von der Dateisteuerung verwendet.

#### FCIS\_BITS1

Existenzbits zum Angeben der Argumente, die angegeben wurden. Um das Argument abzurufen, das einem Schlüsselwort zugeordnet ist, müssen Sie die entsprechende Adresse aus der Parameterstruktur der Befehlsebene abrufen. Bevor Sie diese Adresse verwenden, müssen Sie das zugehörige Existenzbit überprüfen. Wenn das Existenzbit inaktiviert ist, wurde das Argument in der Anforderung nicht angegeben, und die Adresse sollte nicht verwendet werden.

X'80' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort FILE enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR1 aussagekräftig.

X'40' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort DSNAME enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR2 aussagekräftig.

X'20' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort FWDRECSTATUS enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR3 aussagekräftig.

X'10' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort STRINGS enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR4 aussagekräftig.

X'08' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort BASEDSNAME enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR5 aussagekräftig.

X'04' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort LSRPOOLID enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR6 aussagekräftig.

X'02' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort READ enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR7 aussagekräftig.

X'01' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort UPDATE enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR8 aussagekräftig.

## FCIS\_BITS2

Existenzbits zum Angeben der Argumente, die angegeben wurden. Die Ausführungen zu FCIS\_BITS1 gelten auch für FCIS\_BITS2.

- X'80'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort BROWSE enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR9 aussagekräftig.
- X'40'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort ADD enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR10 aussagekräftig.
- X'20'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort DELETE enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR11 aussagekräftig.
- X'10'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort DISPOSITION enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR12 aussagekräftig.
- X'08'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort EMPTYSTATUS enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR13 aussagekräftig.
- X'04'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort OPENSTATUS enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR14 aussagekräftig.
- X'02'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort ENABLESTATUS enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR15 aussagekräftig.
- X'01'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort RECOVSTATUS enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR16 aussagekräftig.

## FCIS\_BITS3

Existenzbits zum Angeben der Argumente, die angegeben wurden. Die Ausführungen zu FCIS\_BITS1 gelten auch für FCIS\_BITS3.

- X'80'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort ACCESSMETHOD enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR17 aussagekräftig.
- X'40'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort TYPE enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR18 aussagekräftig.
- X'20'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort OBJECT enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR19 aussagekräftig.
- X'10'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort REMOTESYSTEM enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR20 aussagekräftig.
- X'08'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort REMOTENAME enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR21 aussagekräftig.
- X'04'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort RECORDFORMAT enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR22 aussagekräftig.

**X'02'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort BLOCKFORMAT enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR23 aussagekräftig.

**X'01'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort KEYLENGTH enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR24 aussagekräftig.

#### **FCIS\_BITS4**

Existenzbits zum Angeben der Argumente, die angegeben wurden. Die Ausführungen zu FCIS\_BITS1 gelten auch für FCIS\_BITS4.

**X'80'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort KEYPOSITION enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR25 aussagekräftig.

**X'40'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort RECORDSIZE enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR26 aussagekräftig.

**X'20'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort RELTYPE enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR27 aussagekräftig.

**X'10'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort EXCLUSIVE enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR28 aussagekräftig.

**X'08'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort BLOCKKEYLEN enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR29 aussagekräftig.

**X'04'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort BLOCKSIZE enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR30 aussagekräftig.

**X'02'** Wird nicht von der Dateisteuerung verwendet.

**X'01'** Wird nicht von der Dateisteuerung verwendet.

#### **FCIS\_BITS5**

Existenzbits zum Angeben der Argumente, die angegeben wurden. Die Ausführungen zu FCIS\_BITS1 gelten auch für FCIS\_BITS5.

**X'80'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort TABLE enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR33 aussagekräftig.

**X'40'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort MAXNUMRECS enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR34 aussagekräftig.

**X'20'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort READINTEG enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR35 aussagekräftig.

**X'10'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort RLSACCESS enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR36 aussagekräftig.

**X'08'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort DEFINESOURCE enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR37 aussagekräftig.

- X'04'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort **INSTALLAGT** enthält. Ist dieses Existenzbit gesetzt, dann ist **FCIS\_ADDR38** aussagekräftig.
- X'02'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort **INSTALLUSR** enthält. Ist dieses Existenzbit gesetzt, dann ist **FCIS\_ADDR39** aussagekräftig.
- X'01'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort **CHANGEAGENT** enthält. Ist dieses Existenzbit gesetzt, dann ist **FCIS\_ADDR40** aussagekräftig.

#### **FCIS\_BITS6**

Gibt an, ob bestimmte Schlüsselwörter im SPI-Befehl für die Dateisteuerung angegeben wurden.

- X'80'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **START** enthält.
- X'40'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **NEXT** enthält.
- X'20'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **END** enthält.
- X'10'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **WAIT** enthält.
- X'08'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **NO-WAIT** enthält.
- X'04'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **FORCE** enthält.
- X'02'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **ENABLED** enthält.
- X'01'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **DISABLED** enthält.

#### **FCIS\_BITS7**

Gibt an, ob bestimmte Schlüsselwörter im SPI-Befehl für die Dateisteuerung angegeben wurden. Enthält außerdem das Existenzbit für **JOURNALNUM**.

- X'80'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **OPEN** enthält.
- X'40'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **CLOSED** enthält.
- X'20'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **EMPTY** enthält.
- X'10'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort **JOURNALNUM** enthält. Ist dieses Existenzbit gesetzt, dann ist **FCIS\_ADDR52** aussagekräftig.
- X'08'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **LOADTYPE** enthält.
- X'04'** Wird gesetzt, wenn die Anforderung das Schlüsselwort **POOL** enthält.



- X'02' Wird gesetzt, wenn die Anforderung das Schlüsselwort TABLE-NAME enthält.
- X'01' Wird gesetzt, wenn die Anforderung das Schlüsselwort UP-DATEMODEL enthält.

#### **FCIS\_BITS8**

- X'80' Wird gesetzt, wenn die Anforderung das Schlüsselwort REMO-TETABLE enthält.
- X'40' Wird nicht von der Dateisteuerung verwendet.
- X'20' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort CHANGEUSRID enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR59 aussagekräftig.
- X'10' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort CHANGEAGREL enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR60 aussagekräftig.
- X'08' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort DEFINETIME enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR61 aussagekräftig.
- X'04' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort CHANGETIME enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR62 aussagekräftig.
- X'02' Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort INSTALLTIME enthält. Ist dieses Existenzbit gesetzt, dann ist FCIS\_ADDR63 aussagekräftig.
- X'01' Wird nicht von der Dateisteuerung verwendet.

#### **FCIS\_ADDR1**

Adresse eines 8-Byte-Bereichs, der den Namen aus FILE enthält.

#### **FCIS\_ADDR2**

Adresse eines 44-Byte-Bereichs, der den Namen von DSNAME enthält.

#### **FCIS\_ADDR3**

Adresse eines 4-Byte-Bereichs, der den CVDA-Wert von FWDRECOVSTATUS enthält.

#### **FCIS\_ADDR4**

Adresse eines 4-Byte-Bereichs, der die Daten aus STRINGS enthält.

#### **FCIS\_ADDR5**

Adresse eines 44-Byte-Bereichs, der den Namen aus BASEDSNAME enthält.

#### **FCIS\_ADDR6**

Adresse eines 4-Byte-Bereichs, der die Daten aus LSRPOOLID enthält.

#### **FCIS\_ADDR7**

Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus READ enthält.

#### **FCIS\_ADDR8**

Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus UPDATE enthält.

#### **FCIS\_ADDR9**

Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus BROWSE enthält.

#### **FCIS\_ADDR10**

Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus ADD enthält.

- FCIS\_ADDR11**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus DELETE enthält.
- FCIS\_ADDR12**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus DISPOSITION enthält.
- FCIS\_ADDR13**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus EMPTYSTATUS enthält.
- FCIS\_ADDR14**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus OPENSTATUS enthält.
- FCIS\_ADDR15**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus ENABLESTATUS enthält.
- FCIS\_ADDR16**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus RECOVSTATUS enthält.
- FCIS\_ADDR17**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus ACCESSMETHOD enthält.
- FCIS\_ADDR18**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus TYPE enthält.
- FCIS\_ADDR19**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus OBJECT enthält.
- FCIS\_ADDR20**  
Adresse eines 4-Byte-Bereichs, der den Namen aus REMOTESYSTEM enthält.
- FCIS\_ADDR21**  
Adresse eines 8-Byte-Bereichs, der den Namen aus REMOTENAME enthält.
- FCIS\_ADDR22**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus RECORDFORMAT enthält.
- FCIS\_ADDR23**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus BLOCKFORMAT enthält.
- FCIS\_ADDR24**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus KEYLENGTH enthält.
- FCIS\_ADDR25**  
Adresse eines 4-Byte-Bereichs, der die Daten aus KEYPOSITION enthält.
- FCIS\_ADDR26**  
Adresse eines 4-Byte-Bereichs, der die Daten aus RECORDSIZE enthält.
- FCIS\_ADDR27**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus RELTYPE enthält.
- FCIS\_ADDR28**  
Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus EXCLUSIVE enthält.
- FCIS\_ADDR29**  
Adresse eines 4-Byte-Bereichs, der die Daten aus BLOCKKEYLEN enthält.
- FCIS\_ADDR30**  
Adresse eines 4-Byte-Bereichs, der die Daten aus BLOCKSIZE enthält.

**FCIS\_ADDR31**

Wird nicht von der Dateisteuerung verwendet.

**FCIS\_ADDR32**

Adresse eines 4-Byte-Bereichs, der die Daten aus BUSY enthält.

**FCIS\_ADDR33**

Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus TABLE enthält.

**FCIS\_ADDR34**

Adresse eines 4-Byte-Bereichs, der die Daten aus MAXNUMRECS enthält.

**FCIS\_ADDR35**

Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus READINTEG enthält.

**FCIS\_ADDR36**

Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus RLSACCESS enthält.

**FCIS\_ADDR37**

Adresse eines 8-Byte-Bereichs, der die Daten aus DEFINESOURCE enthält.

**FCIS\_ADDR38**

Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus INSTALLAGENT enthält.

**FCIS\_ADDR39**

Adresse eines 8-Byte-Bereichs, der die Daten aus INSTALLUSRID enthält.

**FCIS\_ADDR40**

Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus CHANGEAGENT enthält.

**FCIS\_ADDR41 bis FCIS\_ADDR51**

Werden nicht von der Dateisteuerung verwendet.

**FCIS\_ADDR52**

Adresse eines 4-Byte-Bereichs, der die Daten aus JOURNALNUM enthält.

**FCIS\_ADDR53**

Adresse eines 4-Byte-Bereichs, der die Daten aus LOADTYPE enthält.

**FCIS\_ADDR54**

Adresse eines 4-Byte-Bereichs, der die Daten aus CFDTPOOL enthält.

**FCIS\_ADDR55**

Adresse eines 4-Byte-Bereichs, der die Daten aus TABLENAME enthält.

**FCIS\_ADDR56**

Adresse eines 4-Byte-Bereichs, der die Daten aus UPDATEMODEL enthält.

**FCIS\_ADDR57**

Adresse eines 4-Byte-Bereichs, der die Daten aus REMOTETABLE enthält.

**FCIS\_ADDR58**

Adresse eines 4-Byte-Bereichs, der den CVDA-Wert aus RBATYPE enthält.

**FCIS\_ADDR59**

Adresse eines 8-Byte-Bereichs, der die Daten aus CHANGEUSRID enthält.

**FCIS\_ADDR60**

Adresse eines 4-Byte-Bereichs, der die Daten von CHANGEAGREL enthält.

**FCIS\_ADDR61**

Adresse eines 8-Byte-Bereichs, der die Daten aus DEFINETIME enthält.

**FCIS\_ADDR62**

Adresse eines 8-Byte-Bereichs, der die Daten aus CHANGETIME enthält.

**FCIS\_ADDR63**

Adresse eines 8-Byte-Bereichs, der die Daten aus INSTALLTIME enthält.

## Ändern von Feldern in der Parameterstruktur der Befehlsebene

Einige Felder, die an eine SPI-Anforderung zur Dateisteuerung übergeben werden, werden als Eingabe für die Anforderung verwendet, während andere als Ausgabe für die Anforderung verwendet werden. Die Methode, die Ihr Benutzerexitprogramm zum Ändern eines Felds verwendet, hängt von der Verwendung des betreffenden Felds ab.

Als allgemeine Regel gilt Folgendes:

- Bei Anforderungen vom Typ INQUIRE FILE sind mit Ausnahme von FILE alle Felder Ausgabefelder.
- Bei Anforderungen vom Typ SET FILE sind alle Felder Eingabefelder.

Eine vollständige Beschreibung der Parameter für INQUIRE FILE finden Sie in Tabelle 5. Eine vollständige Beschreibung der Parameter für SET FILE finden Sie in Tabelle 6 auf Seite 108.

*Tabelle 5. Anforderungen vom Typ INQUIRE FILE.. Beziehungen zwischen Argumenten, Schlüsselwörtern, Datentypen und den Typen 'Eingabe/Ausgabe'.*

Argument	Schlüsselwort	Datentyp	Eingabe/Ausgabe
Arg1	FILE	CHAR(8)	Siehe Anmerkung.
Arg2	DSNAME	CHAR(44)	Ausgabe
Arg3	FWDRECSTATUS	BIN(31)	Ausgabe
Arg4	STRINGS	BIN(31)	Ausgabe
Arg5	BASEDSNAME	CHAR(44)	Ausgabe
Arg6	LSRPOOLNUM	BIN(31)	Ausgabe
Arg7	READ	BIN(31)	Ausgabe
Arg8	UPDATE	BIN(31)	Ausgabe
Arg9	BROWSE	BIN(31)	Ausgabe
Arg10	ADD	BIN(31)	Ausgabe
Arg11	DELETE	BIN(31)	Ausgabe
Arg12	DISPOSITION	BIN(31)	Ausgabe
Arg13	EMPTYSTATUS	BIN(31)	Ausgabe
Arg14	OPENSTATUS	BIN(31)	Ausgabe
Arg15	ENABLESTATUS	BIN(31)	Ausgabe
Arg16	RECOVSTATUS	BIN(31)	Ausgabe
Arg17	ACCESSMETHOD	BIN(31)	Ausgabe
Arg18	TYPE	BIN(31)	Ausgabe
Arg19	OBJECT	BIN(31)	Ausgabe
Arg20	REMOTESYSTEM	CHAR(4)	Ausgabe
Arg21	REMOTENAME	CHAR(8)	Ausgabe

Tabelle 5. Anforderungen vom Typ INQUIRE FILE. (Forts.). Beziehungen zwischen Argumenten, Schlüsselwörtern, Datentypen und den Typen 'Eingabe/Ausgabe'.

Argument	Schlüsselwort	Datentyp	Eingabe/Ausgabe
Arg22	RECORDFORMAT	BIN(31)	Ausgabe
Arg23	BLOCKFORMAT	BIN(31)	Ausgabe
Arg24	KEYLENGTH	BIN(31)	Ausgabe
Arg25	KEYPOSITION	BIN(31)	Ausgabe
Arg26	RECORDSIZE	BIN(31)	Ausgabe
Arg27	RELTYPE	BIN(31)	Ausgabe
Arg28	EXCLUSIVE	BIN(31)	Ausgabe
Arg29	BLOCKKEYLEN	BIN(31)	Ausgabe
Arg30	BLOCKSIZE	BIN(31)	Ausgabe
Arg31	*	*	*
Arg32	BUSY	BIN(31)	Ausgabe
Arg33	TABLE	BIN(31)	Ausgabe
Arg34	MAXNUMRECS	BIN(31)	Ausgabe
Arg35	READINTEG	BIN(31)	Ausgabe
Arg36	RLSACCESS	BIN(31)	Ausgabe
Arg37	DEFINESOURCE	CHAR(8)	Ausgabe
Arg38	INSTALLAGENT	BIN(31)	Ausgabe
Arg39	INSTALLUSRID	CHAR(8)	Ausgabe
Arg40	CHANGEAGENT	BIN(31)	Ausgabe
Arg41 bis Arg51	*	*	*
Arg52	JOURNALNUM	BIN(15)	Ausgabe
Arg53	LOADTYPE	BIN(31)	Ausgabe
Arg54	CEDTPOOL	CHAR(8)	Ausgabe
Arg55	TABLERNAME	CHAR(8)	Ausgabe
Arg56	UPDATERODEL	BIN(31)	Ausgabe
Arg57	RENOTETABLE	BIN(31)	Ausgabe
Arg58	RBATYPE	BIN(31)	Ausgabe
Arg59	CHANGEUSRID	CHAR(8)	Ausgabe
Arg60	CHANGEAGREL	BIN(31)	Ausgabe
Arg61	DEFINETIME	CHAR(8)	Ausgabe
Arg62	CHANGETIME	CHAR(8)	Ausgabe
Arg63	INSTALLTIME	CHAR(8)	Ausgabe

**Anmerkung:** Der Dateiparameter 'file' in Befehlen vom Typ INQUIRE FILE wird wie folgt behandelt:

- Als Eingabefeld, wenn die Anforderung nicht START, NEXT oder END angibt.
- Als Ausgabefeld, wenn die Anforderung NEXT angibt.
- Wird übergangen, wenn die Anforderung START oder END angibt.

*Tabelle 6. Anforderungen vom Typ SET FILE.. Beziehungen zwischen Argumenten, Schlüsselwörtern, Datentypen und den Typen 'Eingabe/Ausgabe'.*

Argument	Schlüsselwort	Datentyp	Eingabe/Ausgabe
Arg1	FILE	CHAR(8)	Eingabe
Arg2	DSNAME	CHAR(44)	Eingabe
Arg3	FWDRECSTATUS	BIN(31)	Eingabe
Arg4	STRINGS	BIN(31)	Eingabe
Arg5	*	*	*
Arg6	LSRPOOLNUM	BIN(31)	Eingabe
Arg7	READ	BIN(31)	Eingabe
Arg8	UPDATE	BIN(31)	Eingabe
Arg9	BROWSE	BIN(31)	Eingabe
Arg10	ADD	BIN(31)	Eingabe
Arg11	DELETE	BIN(31)	Eingabe
Arg12	DISPOSITION	BIN(31)	Eingabe
Arg13	EMPTYSTATUS	BIN(31)	Eingabe
Arg14	OPENSTATUS	BIN(31)	Eingabe
Arg15	ENABLESTATUS	BIN(31)	Eingabe
Arg16	RECOVSTATUS	BIN(31)	Eingabe
Arg17	*	*	*
Arg18	*	*	*
Arg19	*	*	*
Arg20	*	*	*
Arg21	*	*	*
Arg22	*	*	*
Arg23	*	*	*
Arg24	*	*	*
Arg25	*	*	*
Arg26	*	*	*
Arg27	*	*	*
Arg28	EXCLUSIVE	BIN(31)	Eingabe
Arg29	*	*	*
Arg30	*	*	*
Arg31	*	*	*
Arg32	*	*	*
Arg33	TABLE	BIN(31)	Eingabe
Arg34	MAXNUMRECS	BIN(31)	Eingabe
Arg35	READINTEG	BIN(31)	Eingabe
Arg36	RLSACCESS	BIN(31)	Eingabe
Arg37	*	*	*
Arg38	*	*	*
Arg39	*	*	*

Tabelle 6. Anforderungen vom Typ SET FILE. (Forts.). Beziehungen zwischen Argumenten, Schlüsselwörtern, Datentypen und den Typen 'Eingabe/Ausgabe'.

Argument	Schlüsselwort	Datentyp	Eingabe/Ausgabe
Arg40	*	*	*
Arg58	*	*	*
Arg59	*	*	*
Arg60	*	*	*
Arg61	*	*	*
Arg62	*	*	*
Arg63	*	*	*

## Ändern von Eingabefeldern

Die richtige Methode zum Ändern eines Eingabefelds besteht darin, eine neue Kopie desselben zu erstellen und die Adresse in der Parameterliste der Befehlsebene so zu ändern, dass sie auf die neuen Daten verweist.

Ändern Sie ein Eingabefeld nicht durch Ändern der Daten, auf die von der Parameterliste der Befehlsebene verwiesen wird. Dies würde zu einer Beschädigung von Speicher führen, der zum Anwendungsprogramm gehört; dies wiederum würde zu einem Fehler führen, wenn das Programm versucht, das Feld wiederzuverwenden.

## Ändern von Ausgabefeldern

Das unter „Ändern von Eingabefeldern“ beschriebene Verfahren ist für das Ändern von Ausgabefeldern nicht geeignet. (Die Ergebnisse würden an den neuen Bereich und nicht an den Bereich der Anwendung zurückgegeben und wären daher für die Anwendung nicht sichtbar.)

Ein Ausgabefeld wird geändert, indem die Daten geändert werden, auf die von der Parameterliste der Befehlsebene verwiesen wird. Bei einem Ausgabefeld können Sie die vorhandenen Anwendungsdaten ändern, da die Anwendung ohnehin erwartet, dass das Feld geändert wird.

## Ändern des EXEC-Schnittstellendesktors (EID)

Es ist nicht möglich, den EXEC-Schnittstellendesktors (EXEC Interface Descriptor, EID) zu ändern, um bedeutende Änderungen an Anforderungen vorzunehmen. So ist es beispielsweise nicht möglich, eine Anforderung vom Typ INQUIRE FILE in eine Anforderung vom Typ SET FILE zu ändern. Es können jedoch geringfügige Änderungen an Anforderungen vorgenommen werden. So ist es beispielsweise möglich, das Existenzbit für eine Variable zu aktivieren, die in der aktuellen Anforderung bisher nicht angegeben worden ist.

Im nachstehenden Abschnitt werden diejenigen Bits im EID aufgeführt, die geändert werden können. Versuche, andere Bestandteile des EID zu ändern, werden ignoriert.

Ihr Exitprogramm kann jedes Bit in FCIS\_BITS1, FCIS\_BITS2, FCIS\_BITS3, FCIS\_BITS4, FCIS\_BITS5, FCIS\_BITS6 und FCIS\_BITS7 ändern, mit Ausnahme der folgenden:

- Das Existenzbit für das Schlüsselwort FILE.

- Die Bits für die Schlüsselwörter START, NEXT, END, DEFINESOURCE, INSTALAGENT, INSTALLUSRID und CHANGEAGENT.
- Alle Bits, bei denen in der Beschreibung "Wird nicht von der Dateisteuerung verwendet" angegeben ist.
- Bits, die einem Schlüsselwort entsprechen, das für den ausgeführten Befehl nicht gültig ist. Beispiel: Das Bit für das Schlüsselwort CLOSED kann in einer Anforderung vom Typ SET FILE geändert werden, aber nicht in einer Anforderung vom Typ INQUIRE FILE, da CLOSED nur für SET FILE-Anforderungen von Bedeutung ist. Weitere Informationen liefern die Beschreibungen in Tabelle 5 auf Seite 106 und Tabelle 6 auf Seite 108.

Ihr Programm kann eine eigene Parameterstruktur für die Befehlsebene und einen eigenen EID bereitstellen. In diesem Fall müssen UEPCLPS und TS\_ADDR0 entsprechende geändert werden, um auf die neuen Strukturen zu verweisen.

Der EID wird vor der Rückkehr zum Anwendungsprogramm auf seinen ursprünglichen Wert zurückgesetzt. Dies bedeutet, dass Änderungen, die am EID vorgenommen werden, nur für die Dauer der SPI-Anforderung für Dateisteuerung beibehalten werden.

**Anmerkung:** Beim Ändern des EID muss mit Sorgfalt vorgegangen werden, damit keine inkonsistenten Parameter erstellt werden. Beispiel: Wenn in der ursprünglichen Anforderung SET FILE OPEN angegeben war und Ihr Exit das EID-Bit für CLOSED aktiviert hat, würde in der resultierenden SET FILE-Anforderung sowohl OPEN als auch CLOSED angegeben werden. In diesem Fall wären die Ergebnisse des Befehls unvorhersehbar.

## Ändern von Benutzerargumenten

Die Art und Weise, in der ein Benutzerexitprogramm ein Benutzerargument ändern kann, hängt davon ab, ob es sich bei dem Argument um eine Eingabe oder um eine Ausgabe handelt.

- Für Eingabeargumente sollte Ihr Exitprogramm ausreichend Speicher abrufen, um das geänderte Argument zu speichern, den erforderlichen Wert definieren und den zugehörigen Verweis in der Parameterliste auf die Adresse des neu angeforderten Bereichs setzen.
- Bei Ausgabeargumenten sowie bei Eingabe-/Ausgabeargumenten kann Ihr Exitprogramm das betreffende Argument vor Ort aktualisieren, da der Speicherbereich in der Anwendung durch eine Variable dargestellt wird, die erwartungsgemäß einen Wert von CICS empfangen soll.

## Hinzufügen von Benutzerargumenten

Ihr Exitprogramm kann Benutzerargumente hinzufügen, sofern es über die Berechtigung verfügt, das entsprechende Existenzbit im EXEC-Schnittstellendeskriptor (EXEC Interface Descriptor, EID) zu ändern.

Unter der Voraussetzung, dass das hinzuzufügende Argument bisher nicht vorhanden ist, muss das Exitprogramm folgende Aktionen ausführen:

1. Abrufen von Speicher für das hinzuzufügende Argument.
2. Initialisieren des Speichers anhand des erforderlichen Werts.
3. Auswählen und Einstellen des entsprechenden Verweises in der Parameterliste.
4. Auswählen und Einstellen des entsprechenden Existenzbits in Arg0.
5. Gegebenenfalls Ändern der Parameterliste zur Berücksichtigung des neuen Indikators für das Ende der Liste.



## Entfernen von Benutzerargumenten

Ihr Exitprogramm kann Benutzerargumente entfernen, sofern es über die Berechtigung verfügt, das entsprechende Existenzbit im EXEC-Schnittstellendescriptor (EXEC Interface Descriptor, EID) zu ändern.

Unter der Voraussetzung, dass das zu entfernende Argument vorhanden ist, muss das Exitprogramm folgende Aktionen ausführen:

1. Umschalten des Existenzbits des entsprechenden Arguments in Arg0 auf null.
2. Ändern der Parameterliste zur Berücksichtigung des neuen Indikators für das Ende der Liste.

---

## Exits XFCSREQ und XFCSREQC für das Dateisteuerprogramm für Dateistatus

Im Dateisteuerprogramm für Dateistatus werden zwei Benutzerexits bereitgestellt, die Sie vor und nach einer Dateianforderung aufrufen können.

### XFCSREQ

Dieser Exit wird vor der Bearbeitung einer Dateianforderung vom Typ ENABLE, DISABLE, OPEN, CLOSE oder CANCEL CLOSE aufgerufen. Mithilfe von XFCSREQ können Sie Informationen zum Status einer Datei erfassen und so beispielsweise ermitteln, welche Dateianforderungen (SERVREQs) gültig sind und welche Journalführungsoptionen festgelegt sind. Auf der Grundlage dieser Informationen können Sie Anforderungen im Bedarfsfall unterdrücken. Ausführliche Informationen hierzu finden Sie unter Rückgabecode UERCBYP.

### XFCSREQC

Dieser Exit wird nach der Bearbeitung einer Dateianforderung aufgerufen. Mithilfe von XFCSREQC können Sie Informationen zum Dataset erfassen, das einer Datei zugeordnet ist und so beispielsweise ermitteln, welche Wiederherstellungsoptionen festgelegt sind. XFCSREQC wird auch dann aufgerufen, wenn Sie XFCSREQ zum Unterdrücken der betreffenden Dateianforderung verwendet haben.

Für Anforderungen vom Typ ENABLE, DISABLE, OPEN und CANCEL CLOSE wird jeder Exit nur einmal aufgerufen. Bei Schließenanforderungen vom Typ CLOSE hingegen kann es sein, dass die Exits mehr als einmal aufgerufen werden, da eine Datei vor der eigentlichen Schließung in den Quiescemodus versetzt werden kann. Es gibt auch zwei Situationen, in denen die Benutzerexits XFCSREQ und XFCSREQC während einer CLOSE-Anforderung nicht aufgerufen werden:

1. Auf einem kontrollierten, nicht sofortigen CICS-Systemabschluss, wenn CICS alle Dateien schließt.
2. Nach dem Laden einer vom Benutzer verwalteten Datentabelle. Wenn der Ladevorgang für die Datentabelle abgeschlossen ist, ist das Quellendataset nicht mehr erforderlich. Im Folgenden wird die Datei von CICS geschlossen und die Zuordnung der Datei aufgehoben, wobei die Datentabelle geöffnet bleibt.

## Einzelne Anforderung vom Typ CLOSE

Bei einer einzelnen CLOSE-Anforderung werden XFCSREQ und XFCSREQC mehr als einmal aufgerufen, wenn versucht wird, die Schließung durchzuführen, während andere Tasks noch auf die Datei zugreifen. Beispiel: Ein Befehl vom Typ CLOSE NOWAIT führt unter diesen Umständen dazu, dass XFCSREQ aufgerufen wird, bevor versucht wird, die Schließung durchzuführen. Da die Datei immer

noch von Benutzern verwendet wird, wird die Schließung verzögert. Da jedoch NOWAIT angegeben wurde, wird die Anforderung CLOSE abgeschlossen, und es wird XFCSREQC mit dem Rückgabecode UEPFSRSP mit der Einstellung UEFSPEND aufgerufen, was bedeutet, dass die Schließung ansteht. Sobald alle Aktivitäten für die Datei beendet worden sind, wird die Datei geschlossen, und XFCSREQ sowie XFCSREQC werden unter der Task aufgerufen, die die Datei geschlossen hat.

### Anforderung vom Typ CLOSE WAIT

Bei einer Anforderung vom Typ CLOSE WAIT werden die Exits wie folgt aufgerufen: XFCSREQ wird aufgerufen, die Task fordert die Schließung der Datei an und wartet auf die Durchführung dieser Schließung. Sobald alle Aktivitäten für die Datei beendet worden sind, wird die Datei geschlossen, und XFCSREQ sowie XFCSREQC werden unter der Task aufgerufen, die die Datei geschlossen hat. Da die Schließung nun vollständig ausgeführt worden ist, wird die Task, die den Befehl CLOSE WAIT ausgegeben hat, fortgesetzt; sie beendet ihre CLOSE-Anforderung und ruft XFCSREQC auf.

### Anforderung vom Typ CANCEL CLOSE

Eine Anforderung vom Typ CANCEL CLOSE wird von CICS als Antwort auf einen Befehl vom Typ UNQUIESCE ausgegeben, der einen anstehenden Befehl vom Typ QUIESCE abbricht. Mit dem Befehl QUIESCE für ein Dataset werden sofort sämtliche Dateien, die für das angegebene Dataset geöffnet sind, auf 'nicht aktiviert' gesetzt, um zu verhindern, dass neuen Tasks Zugriff auf das Dataset gewährt wird. Vor dem Schließen einer Datei wird der Teil der Operation mit dem Schließvorgang jedoch so lange in den Wartestatus versetzt, bis die letzte Benutzertask beendet worden ist. (Dies ist dasselbe Verhalten wie bei jeder Schließoperation für eine Datei.) Ein Befehl vom Typ UNQUIESCE, der ausgegeben wird, während sich der Schließvorgang noch im Wartestatus befindet, führt zur Ausgabe einer Anforderung vom Typ CANCEL CLOSE und zum Aufruf der Exits XFCSREQ und XFCSREQC. Hierbei ist zu beachten, dass der Befehl CANCEL CLOSE ausschließlich für solche Schließanforderungen ausgegeben wird, die von einem QUIESCE-Befehl eingeleitet wurden, und nicht für andere Arten von Schließanforderungen.

## Exit XFCSREQ

Dieser Exit wird aufgerufen, bevor versucht wird, einen Dateibefehl vom Typ ENABLE, DISABLE, OPEN, CLOSE oder CANCEL CLOSE auszuführen.

### Zeitpunkt des Aufrufs

Vor dem Versuch, einen Dateibefehl vom Typ ENABLE, DISABLE, OPEN, CLOSE oder CANCEL CLOSE auszuführen.

**Anmerkung:** Bei Anforderungen, die über Funktionen übertragen werden, wird der Exit auf dem System aufgerufen, auf dem die betreffende Datei lokal gespeichert ist.

### Exitspezifische Parameter

#### UEPFSREQ

Adresse eines 2-Byte-Felds, das den Typ der Dateianforderung angibt. Das erste Byte enthält einen der folgenden Werte:

#### UEPFSOPN

Öffnungsanforderung.

**UEPFSCLS**  
Schließenanforderung.

**UEPFSENB**  
Aktivierungsanforderung.

**UEPFSDIS**  
Inaktivierungsanforderung.

**UEPFSCAN**  
Abbruch der Schließenanforderung für die Datei.

Wenn das erste Byte eine Öffnungsanforderung (UEPFSOPN) angibt, zeigt das zweite Byte den Typ des Öffnungsvorgangs an:

**UEPFSNOP**  
Normales Öffnen.

**UEPFSOFB**  
Öffnen für Backout.

Wenn das erste Byte eine Schließenanforderung (UEPFSCLS) angibt, zeigt das zweite Byte den Typ des Schließvorgangs an:

**UEPFSNC**  
Normales Schließen.

**UEPFSCP**  
Schließen anstehend.

**UEPFSERM**  
Schließen bei Ende des Lademodus.

**UEPFSIMM**  
Sofortiges Schließen.

**UEPFSICP**  
Sofortiges Schließen anstehend.

**UEPFSQU**  
Schließen für Quiesce (RLS).

**UEPFILE**  
Adresse des 8-Byte-Dateinamens.

**UEPFINFO**  
Adresse eines Speicherbereichs, der Informationen zur Datei enthält. Der Bereich kann mit Hilfe des Pseudobereichs (DSECT) DFHUEFDS zugeordnet werden, der die folgenden Felder enthält:

**UEFLNAME**  
Dateiname aus 8 Zeichen.

**UEDSNAME**  
Dateiname (DSNAME) aus 44 Zeichen für das Dataset, das der Datei zugeordnet ist, sofern dieser Name vor Ausgabe der Dateianforderung festgelegt wurde.

**UEFSERV**  
Ein Byte zum Angeben der SERVREQ-Einstellungen für diese Datei. Mögliche Werte:

**UEFRDIM**  
Lesen ist gültig.

**UEFUPDIM**

Aktualisieren ist gültig.

**UEFADDIM**

Hinzufügen ist gültig.

**UEFDELIM**

Löschen ist gültig.

**UEFBRZIM**

Durchsuchen ist gültig.

**UEFDSJL**

Ein Byte zum Angeben der festgelegten Optionen für automatische Journalführung für diese Datei. Mögliche Werte:

**UEFJRO**

Journal ist schreibgeschützt.

**UEFJRU**

Lesevorgang im Journal für Aktualisierung.

**UEFJWU**

Schreibvorgang im Journal für Aktualisierung.

**UEFJWA**

Schreibvorgang im Journal für Hinzufügung.

**UEFJDSN**

Dateiname (DSNAME) wurde im Journal aufgezeichnet.

**UEFJSYN**

Synchroner Lesevorgang im Journal.

**UEFJASY**

Asynchroner Schreibvorgang im Journal.

**UEFDSVJL**

Ein Byte, das eine weitere Option für automatische Journalführung angibt, die nur für VSAM-Dateien gilt. Der Wert lautet wie folgt:

**UEFJWAC**

Schreibvorgang für Hinzufügung abgeschlossen.

**UEFDSJID**

Ein Byte, das die Nummer des Journals enthält, das für die automatische Journalführung verwendet werden soll (sofern vorhanden).

**UEFDSACC**

Ein Byte zum Angeben der Zugriffsmethode der Datei. Mögliche Werte:

**UEFVSAM**

VSAM-Datei.

**UEFBDAM**

BDAM-Datei.

**UEFCFDT**

Coupling-Facility-Datentabelle.

**UEFBCRV**

Wird für diesen Exit auf Nullen gesetzt.

**UEFFRLOG**

Wird für diesen Exit auf Nullen gesetzt.

**UEFFRCLG**

Wird für diesen Exit auf Leerzeichen gesetzt.

**UEFCDATE**

Wird für diesen Exit auf Nullen gesetzt.

**UEFCTIME**

Wird für diesen Exit auf Nullen gesetzt.

**UEFBCAS**

Wird für diesen Exit auf Nullen gesetzt.

**UEFACBCP**

Dieses Feld wird für diesen Exit auf Nullen gesetzt.

**Anmerkung:** Für diesen Exit werden nur die ersten sieben Felder von UEPFINFO festgelegt. Von den verbleibenden Feldern wird URFFRCLG auf Leerzeichen gesetzt; die anderen werden auf Nullen gesetzt.

**UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen.

**UERCBYP**

Dateianforderung unterdrücken. UERCBYP kann für Folgendes nicht verwendet werden:

- Zum Unterdrücken einer Schließenanforderung (CLOSE), wenn das zweite Byte von UEPFSREQ angibt, dass es sich um einen der folgenden Schließenanforderungstypen handelt:
  - Schließen bei Ende des Lademodus (UEPFSELM).
  - Sofortiges Schließen (UEPFSIMM).
  - Sofortiges Schließen anstehend (UEPFSICP).
- Zum Unterdrücken einer Öffnungsanforderung (OPEN), wenn eine Datei geöffnet wird, um ein Backout zu verarbeiten, da dies zu einem Backoutfehler führen würde. Das zweite Byte von UEPFSREQ wird auf UEPFSOFB gesetzt, wenn die Datei für ein Backout geöffnet wird.

Im Falle einer gültigen Unterdrückung gibt CICS die Nachricht DFHFC0996 aus:

Open/Close/Enable/Disable/Cancel of close of file  
*filename* suppressed due to intervention of user exit

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

## API- und SPI-Aufrufe

Es können alle API- und SPI-Aufrufe mit Ausnahme von EXEC CICS SHUTDOWN und EXEC CICS XCTL verwendet werden.

### Anmerkung:

1. Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor, um keine Schleife zu verursachen. So sind Sie beispielsweise dafür verantwortlich, das Auftreten einer Schleife zu vermeiden, wenn eine Dateisteueranforderung über den Exit XFCSREQ ausgegeben wird. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.
2. Exitprogramme, die EXEC CICS-Befehle ausgeben, müssen zunächst den EXEC-Schnittstellenblock (EIB) adressieren. Siehe Using CICS services.
3. Exitprogramme, die EXEC CICS-Befehle ausgeben und das Makro DFHEIENT verwenden, sollten das Makro DFHEIRET verwenden, um einen Rückgabecode festzulegen und zu CICS zurückzukehren. Siehe Global user exit XPI examples, showing the use of storage.
4. Exitprogramme können Befehle vom Typ EXEC CICS SET für diejenige Datei aufrufen, deren Statusänderung zu dem Exit geführt hat, der aufgerufen wird. Je nachdem, welche anderen Aktivitäten im CICS-System gleichzeitig ablaufen, besteht jedoch die Möglichkeit, dass ein Deadlock zwischen den Tasks auftritt, die den Status der Datei mithilfe solcher SPI-Befehle bearbeiten.

## Exit XFCSREQC

Dieser Exit wird nach Ausführung eines Dateibefehls vom Typ ENABLE, DISABLE, OPEN, CLOSE oder CANCEL CLOSE aufgerufen.

### Zeitpunkt des Aufrufs

Nach Ausführung eines Dateibefehls vom Typ ENABLE, DISABLE, OPEN, CLOSE oder CANCEL CLOSE.

**Anmerkung:** Bei Anforderungen, die über Funktionen übertragen werden, wird der Exit auf dem System aufgerufen, auf dem die betreffende Datei lokal gespeichert ist.

### Exitspezifische Parameter

#### UEPFSREQ

Adresse eines 2-Byte-Felds, das den Typ der Dateianforderung angibt. Das erste Byte enthält einen der folgenden Werte:

#### UEPFSOPN

Öffnungsanforderung.

#### UEPFSCLS

Schließanforderung.

#### UEPFSENB

Aktivierungsanforderung.

#### UEPFSDIS

Inaktivierungsanforderung.

#### UEPFSCAN

Abbruch der Schließanforderung für die Datei.

Wenn das erste Byte eine Öffnungsanforderung (UEPFSOPN) angibt, zeigt das zweite Byte den Typ des Öffnungsvorgangs an:

**UEPFSNOP**

Normales Öffnen.

**UEPFSOFB**

Öffnen für Backout.

Wenn das erste Byte eine Schließenanforderung (UEPFSCLS) angibt, zeigt das zweite Byte den Typ des Schließvorgangs an:

**UEPFSNC**

Normales Schließen.

**UEPFSCP**

Schließen anstehend.

**UEPFSELM**

Schließen bei Ende des Lademodus.

**UEPFSIMM**

Sofortiges Schließen.

**UEPFSICP**

Sofortiges Schließen anstehend.

**UEPFSQU**

Schließen für Quiesce (RLS).

**UEPFILE**

Adresse des 8-Byte-Dateinamens.

**UEPFINFO**

Adresse eines Speicherbereichs, der Informationen zur Datei enthält. Der Bereich kann mit Hilfe des Pseudobereichs (Dummy Section, DSECT) namens DFHUEFDS zugeordnet werden, der die folgenden Felder enthält:

**UEFLNAME**

Dateiname aus 8 Zeichen.

**UEDSNAME**

Dateiname (DSNAME) aus 44 Zeichen für das Dataset, das der Datei zugeordnet ist.

**UEFSERV**

Ein Byte zum Angeben der SERVREQ-Einstellungen für diese Datei. Mögliche Werte:

**UEFRDIM**

Lesen ist gültig.

**UEFUPDIM**

Aktualisieren ist gültig.

**UEFADDIM**

Hinzufügen ist gültig.

**UEFDELIM**

Löschen ist gültig.

**UEFBRZIM**

Durchsuchen ist gültig.

**UEFDSJL**

Ein Byte zum Angeben der festgelegten Optionen für automatische Journalführung für diese Datei. Mögliche Werte:

- UEFJRO**  
Journal ist schreibgeschützt.
- UEFJRU**  
Lesevorgang im Journal für Aktualisierung.
- UEFJWU**  
Schreibvorgang im Journal für Aktualisierung.
- UEFJWA**  
Schreibvorgang im Journal für Hinzufügung.
- UEFJDSN**  
Dateiname (DSNAME) wurde im Journal auf-  
gezeichnet.
- UEFJSYN**  
Synchroner Lesevorgang im Journal.
- UEFJASY**  
Asynchroner Schreibvorgang im Journal.
- UEFDSVJL**  
Ein Byte, das eine weitere Option für automatische Journal-  
führung angibt, die nur für VSAM-Dateien gilt. Der Wert  
lautet wie folgt:
- UEFJWAC**  
Schreibvorgang für Hinzufügung abgeschlossen.
- UEFDSJID**  
Ein Byte, das die Nummer des Journals enthält, das für die  
automatische Journalführung verwendet werden soll (so-  
fern vorhanden).
- UEFDSACC**  
Ein Byte zum Angeben der Zugriffsmethode der Datei.  
Mögliche Werte:
- UEFVSAM**  
VSAM-Datei.
- UEFBDAM**  
BDAM-Datei.
- UEFCFDT**  
Coupling-Facility-Datentabelle.
- UEFBCRV**  
Ein Byte, das die Attribute für Wiederherstellung des Data-  
sets angibt, das dieser Datei zugeordnet ist. Mögliche Wer-  
te:
- UEFBCFR**  
Aktualisierende Wiederherstellung angegeben.
- UEFBCLOG**  
Protokollierung angegeben.
- UEFBCVAL**  
Flag zur Angabe, dass Attribute für Wiederherstel-  
lung gültig sind.
- UEFFRLOG**  
Ein 1-Byte-Feld, das die ID des Protokolls für aktualisieren-



de Wiederherstellung im Bereich von 1 bis 99 enthält; diese ID stammt aus den Attributen für Wiederherstellung in der CICS-Dateiressourcendefinition. Diese Zahl entspricht einem internen CICS-Journalnamen im Format DFHJ*nn*, wobei *nn* für die Nummer des Protokolls für aktualisierende Wiederherstellung steht. CICS ordnet diesen Journalnamen einem Protokolldatenstrom für aktualisierende Wiederherstellung zu.

Das Feld wird auf null gesetzt, wenn für die Datei keine aktualisierende Wiederherstellung angegeben ist oder wenn der Name des Protokolldatenstroms für aktualisierende Wiederherstellung aus dem ICF-Katalog abgerufen wurde.

#### **UEFFRCLG**

Ein 26-Byte-Feld, das den aus dem ICF-Katalog stammenden Namen des Protokolldatenstroms für aktualisierende Wiederherstellung enthält, der für die aktualisierende Wiederherstellung verwendet werden soll. Das Feld enthält Leerzeichen, wenn im ICF-Katalog keine entsprechende Angabe vorhanden ist oder wenn für die Datei keine aktualisierende Wiederherstellung verwendet wird.

#### **UEFCDATE**

Ein Datum (YYYYDDD+) im gepackten Dezimalformat. Dieses Feld wird nur dann gesetzt, wenn es sich bei der Datei um die letzte Datei handelt, die für die VSAM-Sphäre, der sie zugeordnet ist, geschlossen wird. Es enthält das Datum, an dem die Aktivität für die VSAM-Sphäre beendet wurde (in den Quiescemodus versetzt wurde).

#### **UEFCTIME**

Eine Uhrzeit (HHMMSST+) im gepackten Dezimalformat. Dieses Feld wird nur dann gesetzt, wenn es sich bei der Datei um die letzte Datei handelt, die für die VSAM-Sphäre, der sie zugeordnet ist, geschlossen wird. Es enthält die Uhrzeit, zu der die Aktivität für die VSAM-Sphäre beendet wurde.

#### **UEFBCAS**

Ein Flag-Byte, das die Verfügbarkeit dieser Datei angibt. Ist dieses Flag gesetzt, lautet der Wert wie folgt:

##### **UEPFBCAS**

Die Datei ist als nicht verfügbar markiert.

#### **UEFACBCP**

Adresse einer schreibgeschützten Kopie des ACB (Steuerblock für Zugriffsmethode) für eine VSAM-Datei oder des DCB (Datensteuerblock) für eine BDAM-Datei. Wird nur nach Abschluss einer erfolgreichen Öffnungsanforderung (OPEN) gesetzt.

#### **UEPFSRSP**

Adresse eines Byte, das die Rückgabecodes für die Anforderung enthält. Dieser Parameter hat einen der folgenden Werte:

##### **UEFSNORM**

Antwort 'Normal'.

**UEFSWARN**

Warnantwort (Warning).

**UEFSFAIL**

Fehlerantwort (Failure).

**UEFSPEND**

Antwort 'Anstehend' (Pending). Die Antwort 'Pending' kann nur nach einer Schließenanforderung (CLOSE) zurückgegeben werden. Sie weist darauf hin, dass aufgrund der Schließenanforderung die Schließung der Datei ansteht und die Datei in den Quiescemodus versetzt wird. Die Datei wird geschlossen, sobald alle Aktivitäten für sie beendet worden sind. Hierbei ist zu beachten, dass bei Aktivierung die Exits XFCSREQ und XFCSREQC erneut gesteuert werden, wenn die eigentliche Schließung der Datei stattfindet.

**UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

**Anmerkung:**

1. Die ersten sieben Felder von UEPFINFO (UEFLNAME bis UEFDSACC) werden für alle Anforderungen festgelegt, das heißt nach einer Anforderung vom Typ OPEN, CLOSE, ENABLE oder DISABLE.
2. Die nächsten drei Felder (UEFBCRV, UEFFRLOG und UEFFRCLG) sind nur nach einer erfolgreichen Öffnungsanforderung (OPEN) gültig.
3. Die Felder UEFCDATE bis UEFCBCAS werden nur nach einer erfolgreichen Schließenanforderung (CLOSE) gesetzt. Nach allen anderen Anforderungen werden diese Felder auf null gesetzt, wenn die Datei bereits geschlossen ist, die Schließung fehlschlägt oder die Schließung ansteht.
4. Exitprogramme können Befehle vom Typ EXEC CICS SET für diejenige Datei aufrufen, deren Statusänderung zu dem Exit führt, der aufgerufen wird. Je nachdem, welche anderen Aktivitäten im CICS-System gleichzeitig ablaufen, besteht jedoch die Möglichkeit, dass ein Deadlock zwischen den Tasks auftritt, die den Status der Datei mit solchen SPI-Befehlen bearbeiten.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

**API- und SPI-Aufrufe**

Es können alle API- und SPI-Aufrufe mit Ausnahme von EXEC CICS SHUTDOWN und EXEC CICS XCTL verwendet werden.

**Anmerkung:**

1. Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor, um keine Schleife zu verursachen. So sind Sie beispielsweise dafür verantwortlich, das Auftreten einer Schleife zu vermeiden, wenn eine Dateisteueranforderung über

den Exit XFCSREQC ausgegeben wird. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.

2. Exitprogramme, die EXEC CICS-Befehle ausgeben, müssen zunächst den EXEC-Schnittstellenblock (EIB) adressieren. Siehe Using CICS services.
3. Exitprogramme, die EXEC CICS-Befehle ausgeben und das Makro DFHEIENT verwenden, sollten das Makro DFHEIRET verwenden, um einen Rückgabecode festzulegen und zu CICS zurückzukehren. Siehe Global user exit XPI examples, showing the use of storage.

## Beispiel eines globalen Benutzerexitprogramms für XFCSREQC

DFH\$REQC stellt eine Beispielverarbeitung für den globalen Benutzerexit XFCSREQC des Programms für Dateisteuerungsstatus bereit. Weitere Informationen finden Sie unter File control state sample exit program: DFH\$REQC.

---

## Exit XFCNREC für das Dateisteuerprogramm für Öffnen/Schließen

Mithilfe des Exits XFCNREC können Sie Fehler beim Öffnen eines Nicht-RLS-Datasets unterdrücken.

Bei RLS-Datasets ist die Wiederherstellung eine Eigenschaft des Datasets. Daher dürfen Dateien und ihr zugehöriges Basisdataset keine Wiederherstellungsattribute aufweisen, die nicht übereinstimmen, also voneinander abweichen. Weitere Informationen zum Schreiben eines Exitprogramms für XFCNREC finden Sie unter Configuring for recovery of CICS-managed resources.

### Exit XFCNREC

#### Zeitpunkt des Aufrufs

Vor dem Öffnen einer Datei in folgenden Situationen:

1. Es wird eine Abweichung zwischen der Wiederherstellungseinstellung für Backouts der Datei und des zugehörigen Nicht-RLS-Datasets festgestellt.
2. Es wird eine Abweichung festgestellt, weil BWO (Backup-while-open; Sichern von geöffneten Dateien) erforderlich ist, die Attribute für Wiederherstellung jedoch angeben, dass keine zugehörige Datei für aktualisierende Wiederherstellung angegeben worden ist.

#### Exitspezifische Parameter

##### UEFILE

Adresse des 8-Byte-Dateinamens. Wenn der Dateiname weniger als 8 Zeichen lang ist, wird er mit Leerzeichen aufgefüllt.

##### UEDSETN

Adresse des 44-Byte-Namens des Basisdatasets. Wenn der Name des Datasets weniger als 44 Zeichen lang ist, wird er mit Leerzeichen aufgefüllt.

##### UEPFRCV

Adresse eines 1-Byte-Felds mit der Wiederherstellungseinstellung für Backouts der Datei gemäß Angabe in der FILE-Definition. Möglicher Wert:

##### UEPFLOG

Protokollierung für Backouts ist angegeben.

Ist RECOV(NONE) in der FILE-Definition angegeben, enthält das adressierte Feld hexadezimale Nullen.

Dieses Feld hat keine Bedeutung, wenn der Exit aufgrund einer BWO-Abweichung gesteuert wird.

#### **UEPFAIL**

Adresse eines 1-Byte-Felds mit dem Grund für die Abweichung. Mögliche Werte:

#### **UEPBWOF**

Es liegt eine BWO-Abweichung vor.

#### **UEPATTF**

Es liegt eine Abweichung in den Wiederherstellungseinstellungen für Backouts vor.

#### **UEPOPEN**

Adresse eines 1-Byte-Felds. Der Standardwert ist N. Um Fehler beim Öffnen von Dateien aufgrund von BWO-Abweichungen zu umgehen, setzen Sie das adressierte Feld auf Y.

### **Rückgabecodes**

#### **UERCNORM**

Fehlschlagen beim Öffnen, wie sonst auch.

#### **UERCBYP**

Umgehung von Fehlern beim Öffnen; Abweichungen werden akzeptiert.

#### **XPI-Aufrufe**

Dürfen nicht verwendet werden.

#### **SPI-Aufrufen**

Dürfen nicht verwendet werden.

#### **API- und SPI-Aufrufe**

Dürfen nicht verwendet werden.

## **Exit XFCNREC bei abweichenden Einstellungen für Rückwärtswiederherstellung**

Verwenden Sie den globalen Benutzerexit XFCNREC, wenn Sie die Verarbeitung zum Öffnen von Dateien fortsetzen wollen, obwohl die Einstellungen für Rückwärtswiederherstellung bei verschiedenen Dateien, die demselben Basisdataset zugeordnet sind, nicht einheitlich sind.

Nach dem Unterdrücken eines Fehlers beim Öffnen kann CICS die Integrität des betreffenden Datasets nicht länger garantieren und markiert das Dataset entsprechend. Wenn Sie nach dem Unterdrücken eines Fehlers beim Öffnen den Befehl **EXEC CICS INQUIRE DSNAME** oder **CEMT INQUIRE DSNAME RECOVSTATUS** verwenden, wird eine Antwort vom Typ NOTRECOVERABLE zurückgegeben. Anforderungen werden für das Dataset weiterhin protokolliert. Bei der Protokollierung werden nur Dateien verwendet, für die BACKOUT definiert ist.

Der abweichende Status des Datasets besteht so lange fort, bis ein Befehl vom Typ **EXEC CICS SET DSNAME REMOVE** oder **CEMT SET DSNAME REMOVE** ausgegeben wird oder bis zu einem Erst- oder Kaltstart von CICS, sofern sich das zugeordnete Dataset nicht im Status "Backout fehlgeschlagen" befindet.

An dem Punkt, an dem die Abweichung akzeptiert wird, gibt CICS eine Warnnachricht aus, die darauf hinweist, dass die Integrität nicht mehr garantiert werden kann. Die Reihenfolge, in der Dateien für dasselbe Basisdataset geöffnet werden, bestimmt den Inhalt der empfangenen Nachricht.

Wenn der Basisclusterblock als nicht wiederherstellbar definiert ist und eine Abweichung aufgetreten ist, wird Zugriff auf das Dataset mit einer nicht wiederherstellbaren Datei erteilt, bevor das Dataset vollständig wiederhergestellt wird.

Es werden drei Parameter an den Exit XFCNREC übergeben, mit deren Hilfe ausgewählt werden kann, welche Abweichungen akzeptiert und welche Abweichungen zurückgewiesen werden sollen. Bei diesen Parametern handelt es sich um die Adresse des Dateinamens, die Adresse des Namens des Basisdatasets und die Adresse eines Byte mit dem Backoutindikator für die Datei. Da der Exit nur dann gesteuert wird, wenn eine Abweichung vorliegt, kann der Backoutindikator für das Dataset von der Einstellung für die Datei abgeleitet werden.

**Anmerkung:** Wenn XFCNREC zum Unterdrücken eines Fehlers beim Öffnen aufgrund einer Abweichung verwendet wird, übergibt der globale Benutzerexit XFCSREQC die Backouteinstellung des Basisdatasets als Exitparameter UEFBCRV und nicht die möglicherweise anders lautende Backouteinstellung der Datei.

## Verwendung von XFCNREC bei einer BWO-Abweichung

Der Exit XFCNREC kann es ermöglichen, dass die Datei geöffnet wird und CICS weiterhin normal ausgeführt wird. Für das geöffnete Dataset ist jedoch keine aktualisierende Wiederherstellung verfügbar.

---

## Exit XFCVSDS zum Empfangen von Quiesce-Anforderungen zur Dateisteuerung

Der Exit XFCVSDS wird aufgerufen, wenn CICS von VSAM RLS darüber benachrichtigt wird, dass aufgrund bestimmter im Sysplex auftretender Ereignisse im Zusammenhang mit Datasets eine Verarbeitung erforderlich ist.

Der Exit XFCVSDS wird vor Ausführung der CICS-Verarbeitung aufgerufen und auch nur dann, wenn ein Datasetnamensblock (DSNB) für das Dataset vorhanden ist. Die folgenden Aktionen bewirken, dass XFCVSDS aufgerufen wird:

- **Ein Dataset wird im gesamten Sysplex in den Quiescemodus versetzt.**

CICS wird nur dann über diese Aktion informiert, wenn in CICS Dateien im RLS-Modus für das betreffende Dataset geöffnet sind.

Wenn CICS über eine Quiesce-Aktion informiert wird, kann das globale Benutzerexitprogramm für XFCVSDS die Versetzung des Datasets in den Quiescemodus abbrechen; in diesem Fall wird die Versetzung in den Quiescemodus im gesamten Sysplex abgebrochen, und das Dataset verbleibt im Unquiescemodus.

- **Ein Dataset wird im gesamten Sysplex in den Unquiescemodus versetzt.**

Sämtliche CICS-Regionen im Sysplex, die bei einem Steuerblock für Zugriffsmethode (Access Method Control Block, ACB) zur VSAM RLS-Steuerung registriert sind, werden über Unquiesce-Aktionen informiert.

- **DFSMSdss will eine Nicht-BWO-Sicherung für ein Dataset starten.**

CICS wird nur dann über den Start einer Nicht-BWO-Sicherung informiert, wenn in CICS Dateien im RLS-Modus für das betreffende Dataset geöffnet sind.

Wenn CICS über den Start einer Nicht-BWO-Sicherung informiert wird, kann die Sicherung mithilfe von XFCVSDS abgebrochen werden.

- **DFSMS hat eine Nicht-BWO-Sicherung für ein Dataset abgeschlossen.**  
Sämtliche CICS-Regionen im Sysplex, die bei einem Steuerblock für Zugriffsmethode (Access Method Control Block, ACB) zur VSAM RLS-Steuerung registriert sind, werden über Abschlüsse von Nicht-BWO-Sicherungen informiert.
- **DFSMS will eine BWO-Sicherung für ein Dataset starten.**  
CICS wird nur dann über den Start einer BWO-Sicherung informiert, wenn in CICS Dateien im RLS-Modus für das betreffende Dataset geöffnet sind.  
Wenn CICS über den Start einer BWO-Sicherung informiert wird, kann die Sicherung mithilfe von XFCVSDS abgebrochen werden.
- **DFSMS hat eine BWO-Sicherung für ein Dataset abgeschlossen.**  
Sämtliche CICS-Regionen im Sysplex, die bei einem Steuerblock für Zugriffsmethode (Access Method Control Block, ACB) zur VSAM RLS-Steuerung registriert sind, werden über Abschlüsse von BWO-Sicherungen informiert.

## Exit XFCVSDS

Der Exit XFCVSDS wird aufgerufen, nachdem CICS von VSAM RLS darüber informiert wurde, dass infolge einer im Sysplex auftretenden datasetbezogenen Aktion eine Verarbeitung erforderlich ist.

### Zeitpunkt des Aufrufs

Nachdem CICS von VSAM RLS darüber informiert wurde, dass infolge einer im Sysplex auftretenden datasetbezogenen Aktion eine Verarbeitung erforderlich ist.

### Exitspezifische Parameter

#### UEPDSNAM

Adresse eines 44-Byte-Felds mit dem Namen des Datasets, auf das die Aktion zutrifft.

#### UEPVSACT

Adresse eines 1-Byte-Felds, das die RLS-Aktion angibt, über die CICS informiert wurde. Mögliche Werte:

#### UEQUIES

Quiesce.

#### UEUNQUIS

Unquiesce.

#### UENBWST

Start von Nicht-BWO-Sicherung.

#### UENBWCMP

Nicht-BWO-Sicherung abgeschlossen.

#### UEBWOST

Start von BWO-Sicherung.

#### UEBWOCMP

BWO-Sicherung abgeschlossen.

#### UEPQUCLS

Adresse eines 1-Byte-Felds, das nur für UEQUIES angibt, wie Dateien, die im RLS-Modus geöffnet sind, geschlossen werden sollen. Mögliche Werte:

#### UEORDCLO

Vor dem Schließen warten, bis alle momentan ausgeführten

Arbeitseinheiten (UOWs) mit Zugriff auf das Dataset den Synchronisationspunkt beendet haben.

#### **UEIMMCLO**

Vor dem Schließen alle momentan ausgeführten Arbeitseinheiten (UOWs) mit Zugriff auf das Dataset abbrechen.

#### **UEPCPTEC**

Adresse eines 1-Byte-Felds, das nur für UENBWST und UEBWOST angibt, ob bei der Sicherung die Methode zum Erstellen einer gleichzeitigen Kopie verwendet werden soll. Mögliche Werte:

#### **UEORDCOP**

Gleichzeitige Kopie wird nicht verwendet.

#### **UECONCOP**

Gleichzeitige Kopie wird verwendet.

### **Rückgabecodes**

#### **UERCNORM**

Verarbeitung fortsetzen; Aktionen abschließen, die zur Unterstützung der VSAM-RLS-Operation erforderlich sind.

#### **UERCBYP**

Gilt nur für die Aktionen UEQUIES, UENBWST und UEBWOST. CICS *soll nicht* die Verarbeitung ausführen, die für die VSAM-RLS-Aktion erforderlich ist, sondern die Aktion im gesamten Sysplex abbrechen.

Der Rückgabecode UERCPURG ist nicht zulässig.

### **XPI-Aufrufe**

Es können alle verwendet werden.

### **API- und SPI-Aufrufe**

An diesem Exit können Sie API- und SPI-Befehle von CICS verwenden. Im Allgemeinen können alle verwendet werden, wobei die folgenden Einschränkungen gelten:

- Sie sollten die Verwendung von Befehlen vermeiden, die dazu führen, dass die ausgebende Task ausgesetzt wird.
- Die Befehle EXEC CICS SHUTDOWN und EXEC CICS XCTL dürfen nicht verwendet werden.
- Die Option QUIESCESTATE im Befehl EXEC CICS SET DSNAME darf nicht für das Dataset UEPDSNAM verwendet werden.
- Die Option OPENSTATUS im Befehl EXEC CICS SET FILE darf nicht für Dateien verwendet werden, die auf das Dataset UEPDSNAM. Ebenso dürfen für Dateien, die auf das Dataset UEPDSNAM verweisen, keine Dateisteueranforderungen ausgegeben werden.

---

## **Exit XFCQUIS zum Senden von Quiesce-Anforderungen zur Dateisteuerung**

Der globale Benutzerexit XFCQUIS wird beim Abschluss eines VSAM-RLS-Quiesce oder VSAM-RLS-Unquiesce eines Datasets aufgerufen, das entweder von einem Befehl vom Typ CEMT SET DSNAME QUIESCESTATE oder von einem Befehl vom Typ **EXEC CICS SET DSNAME QUIESCESTATE** angefordert wurde.

Der Exit wird unabhängig davon aufgerufen, ob die Aktion QUIESCESTATE erfolgreich oder nicht erfolgreich ausgeführt wurde. Dies ermöglicht es Ihnen, jede Art der Verarbeitung auszuführen oder zu planen, die erst dann stattfinden kann, wenn die Quiesce-oder Unquiesce-Verarbeitung abgeschlossen ist.

#### **Zeitpunkt des Aufrufs**

Bei Abschluss (erfolgreich oder fehlgeschlagen) eines Befehls vom Typ SET DSNAME QUIESCESTATE.

#### **Exitspezifische Parameter**

##### **UEPQDSNM**

Adresse eines 44-Byte-Felds mit dem Namen des Datasets, für das ein Quiesce oder Unquiesce durchgeführt wurde.

##### **UEPQSTAT**

Adresse eines 1-Byte-Felds zur Angabe, ob für das Dataset gerade ein Quiesce oder ein Unquiesce durchgeführt wurde. Mögliche Werte:

##### **UEQSD**

Für das Dataset wurde gerade von QUIESCESTATE-(QUIESCED) ein Quiesce durchgeführt. Unvollständige Arbeitseinheiten (UOWs) mit Zugriff auf das Dataset beendeten den Synchronisationspunkt, bevor Dateien im RLS-Modus geschlossen wurden.

##### **UEIMQSD**

Für das Dataset wurde gerade von QUIESCESTATE(IMM-QUIESCED) ein Quiesce durchgeführt. Unvollständige Arbeitseinheiten (UOWs) mit Zugriff auf das Dataset wurden abgebrochen, bevor Dateien im RLS-Modus geschlossen wurden.

##### **UEUNQSD**

Für das Dataset wurde gerade von QUIESCESTATE(UN-QUIESCED) ein Unquiesce durchgeführt.

##### **UEPQRCDE**

Adresse eines 1-Byte-Felds zur Angabe des Ergebnisses des Quiesce oder Unquiesce angibt. Mögliche Werte:

##### **UEQOK**

Erfolgreich.

##### **UEQREJEC**

Zurückgewiesen. Der Ursachencode befindet sich in UEPQCONF.

##### **UEQUNKNO**

Fehlgeschlagen, weil das Dataset im Data Facility Storage Management Subsystem (DFSMS) nicht als VSAM-Datei bekannt ist.

##### **UEQIOERR**

Fehlgeschlagen, weil entweder ein RLS-Fehler vorliegt oder der SMSVSAM-Server nicht verfügbar ist.

##### **UEQCANCL**

Fehlgeschlagen, weil das Quiesce durch den Benutzer abgebrochen wurde (nur UEQSD und UEIMQSD).



**UEQTIMED**

Fehlgeschlagen, weil das Quiesce wegen Zeitlimitüberschreitung abgebrochen wurde (nur UEQSD und UEIMQSD).

**UEQMIGRT**

Fehlgeschlagen, weil das Dataset migriert worden ist.

**UEPQCONF**

Adresse eines 1-Byte-Felds zur Angabe des Grundes, aus dem das Quiesce bzw. das Unquiesce zurückgewiesen wurde (nur für UEQREJEC). Mögliche Werte:

**UEQUIINP**

Quiesce ist in Bearbeitung (nur Status UEQSD und UEIMQSD).

**UEUNQINP**

Unquiesce ist in Bearbeitung.

**UENBWINP**

Nicht-BWO-Kopie ist in Bearbeitung.

**UEBWOINP**

BWO-Kopie ist in Bearbeitung.

**UEUNKINP**

Unbekanntes Ereignis ist in Bearbeitung.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen.

Der Rückgabecode UERCPURG ist nicht zulässig.

**XPI-Aufrufe**

Es können alle verwendet werden.

**API- und SPI-Aufrufe**

An diesem Exit können Sie API- und SPI-Befehle von CICS verwenden. Generell können alle Befehle mit Ausnahme von EXEC CICS SHUTDOWN und EXEC CICS XCTL verwendet werden; es darf jedoch nicht das Schlüsselwort QUIESCESTATE von EXEC CICS SET DSNAME verwendet werden.

---

## **Exits XFCBFAIL, XFCBOUT, XFCBOVER und XFCLDEL für das Dateisteuerprogramm für Wiederherstellung**

CICS stellt vier globale Benutzerexits zur Verfügung, die Sie in Verbindung mit Wiederherstellungsoperationen der Dateisteuerung verwenden können.

**XFCBFAIL**

Wird aufgerufen, wenn während des Backouts ein Fehler auftritt.

**XFCBOUT**

Wird aufgerufen, wenn CICS im Begriff ist, eine Dateiakualisierung zurückzusetzen.

**XFCBOVER**

Wird aufgerufen, wenn CICS im Begriff ist, das Backout einer Arbeitseinheit (UOW) zu überspringen, weil ein Stapelverarbeitungsprogramm den

Schutz für beibehaltene RLS-Sperren außer Kraft gesetzt und ein Dataset für die Stapelverarbeitung geöffnet hat.

#### **XFCLDEL**

Wird aufgerufen, wenn ein Befehl vom Typ WRITE für ein BDAM- oder VSAM-ESDS-Dataset zurückgesetzt wird.

### **Beispielexitprogramme für Dateisteuerung zur Wiederherstellung**

CICS umfasst drei Beispiele für globale Benutzerexitprogramme für Dateisteuerung: DFH\$FCBF, DFH\$FCBV und DFH\$FCLD.

### **Reihenfolge des Aufrufs**

Alle Exits im Dateisteuerprogramm für Wiederherstellung können während eines Versuchs, eine Dateiaktualisierung zurückzusetzen, aufgerufen werden. Wenn das Backout fehlschlägt, kann jeder Exit erneut aufgerufen werden, wenn der Backoutversuch wiederholt wird. Wenn ein Exitprogramm feststellen muss, ob es während des ursprünglichen Backoutversuchs oder während einer Wiederholung aufgerufen wird, kann es dazu den Wert des Felds RE\_ATTACHED\_TRANSACTION überprüfen, das von einem XPI-Aufruf des Typs INQUIRE\_TRANSACTION zurückgegeben wird.

Die Art und Weise, in der die Exits interagieren, und die Reihenfolge, in der sie aufgerufen werden, werden in der folgenden Liste dargestellt. Unter der Voraussetzung, dass alle Exits aktiviert sind, gilt für jeden Backoutversuch bzw. wiederholten Backoutversuch Folgendes:

1. Wenn ein Befehl vom Typ OPEN während eines Backouts fehlschlägt, wird XFCBFAIL aufgerufen. **Es wird keiner der anderen Exits aufgerufen.**
2. Wurde der Befehl SHCDS PERMITNONRLSUPDATE für das Dataset ausgegeben, das zurückgesetzt wird, dann wird XFCBOVER aufgerufen. **Wird UERCNORM zurückgegeben (das Backout nicht ausführen), werden keine weiteren Exits aufgerufen.**
3. Sofern nicht Punkt 1 zutrifft oder sofern nicht XFCBOVER aufgerufen wurde und UERCNORM zurückgegeben hat, wird XFCBOUT aufgerufen.
4. Die Backoutfunktion gibt eine Anforderung vom Typ READ UPDATE für den Datensatz aus, der zurückgesetzt wird.  
Schlägt die READ UPDATE-Anforderung fehl, wird XFCBFAIL aufgerufen. **Auf diesen Exit folgen keine weitere Exits.**
5. Wenn es sich bei der Aktualisierung, die zurückgesetzt werden soll, um eine Schreibanforderung (WRITE) für ein Dataset handelt, das keine physischen Löschungen unterstützt (d. h. um ein BDAM-Dataset oder ein VSAM-ESDS), wird XFCLDEL aufgerufen.
6. Tritt nach diesem Punkt ein Fehler auf, wird XFCBFAIL aufgerufen.

### **Aktivierung der Exitprogramme**

Um ein Exitprogramm zu aktivieren, müssen Sie eine von zwei möglichen Aktionen ausführen.

Um diese Exits zu aktivieren, müssen Sie einen der folgenden Schritte ausführen:

- Geben Sie den Systeminitialisierungsparameter TBEXITS=(name1,name2,name3,name4,name5,name6) an. Hierbei sind 'name1' bis 'name6' die Namen Ihrer Benutzerexitprogramme für XRCINIT, XRCINPT, XFCBFAIL, XFCLDEL, XFCBOVER und XFCBOUT.

- Aktivieren Sie die Exits während der ersten Phase der Initialisierung mithilfe eines PLTPI-Programms.

Wenn Sie den Parameter TBEXITS verwenden, um die Exits zu aktivieren, wird ein globaler Arbeitsbereich mit 4 Byte bereitgestellt. Wenn Sie ein PLTPI-Programm verwenden, können Sie die Größe des globalen Arbeitsbereichs selbst auswählen. Darüber hinaus können Sie auch mehrere Exitprogramme zur Verwendung an den einzelnen Exitpunkten auswählen, während der Parameter TBEXITS an jedem Exitpunkt jeweils immer nur ein Exitprogramm erlaubt. Informationen zur PLTPI-Verarbeitung finden Sie unter Writing initialization and shutdown programs.

## **Exit XFCBFAIL beim Fehlschlagen eines Dateisteuerungsbackouts**

Der Exit XFCBFAIL wird immer dann aufgerufen, wenn während des Backouts einer Aktualisierung an einem Datensatz ein Fehler auftritt.

Wenn innerhalb einer bestimmten Arbeitseinheit (UOW) Backoutfehler bei mehreren Datensätzen in derselben Datei oder bei Datensätzen in mehreren Dateien auftreten, wird der Exit wie folgt aufgerufen:

- Für den ersten Datensatz in jedem Dataset, für das das Backout fehlschlägt.  
Wenn einem einzelnen Dataset mehr als eine Datei zugeordnet ist, führt nur der erste Datensatz in der ersten Datei, deren Backout in der UOW fehlschlägt, dazu, dass CICS den Exit aufruft. Alle nachfolgenden Datensätze schlagen zwar mit demselben Fehler fehl, aber der Exit wird nicht erneut aufgerufen.
- Für den ersten Datensatz für jedes Dataset, das bei einer Wiederholung des Backouts für diese UOW fehlschlägt.

Bei Backoutfehlern im Zusammenhang mit anderen Ressourcen (außerhalb der Dateisteuerung) in der UOW wird der Exit nicht aufgerufen.

Bei VSAM-Dateien werden im Rahmen der Backoutfehlerverarbeitung Informationen gespeichert, die es ermöglichen, das Backout zu einem späteren Zeitpunkt zu wiederholen.

Bei BDAM-Dateien kann das Backout nicht erneut versucht werden. Wenn das Backout für eine BDAM-Datei fehlschlägt, können Sie den Exit XFCBFAIL verwenden, um die Datenintegrität aufrecht zu erhalten, indem CICS sofort beendet wird. Wenn der Exit XFCBFAIL nicht aktiviert ist oder CICS nicht beendet, werden die BDAM-Daten festgeschrieben, und die Sperren werden freigegeben. Wenn der Exit hingegen aktiviert ist, können Sie das globale Benutzerexitprogramm für XFCBFAIL verwenden, um Informationen zu speichern, die Sie zum manuellen Korrigieren der Daten verwenden können. Sie müssen jedoch darauf achten, dass Sie dabei keine anderen Änderungen zurücksetzen, die zwischen dem Zeitpunkt des Backoutfehlers und dem Zeitpunkt Ihrer eigenen manuellen Wiederherstellungsaktion vorgenommen wurden.

### **Zeitpunkt des Aufrufs**

Wenn ein Fehler während des Backouts einer Änderung an einer Datei ein Fehler auftritt (beim ersten Fehler in der UOW für das Dataset, das der Datei zugeordnet ist).

### **Exitspezifische Parameter**

#### **UEPBLOGR**

Adresse des Dateisteuerungsabschnitts des Protokollsatzes, der die Aktualisierung darstellt, die gerade zurückgesetzt wurde, als der

Dateisteuerungsfehler auftrat. Der Protokollsatz kann mithilfe des Pseudobereichs (DSECT) DFHFCLGD zugeordnet werden.

**UEPTRANS**

Adresse eines 4-Byte-Felds mit der Transaktions-ID, unter der die Aktualisierung durchgeführt wurde, die zurückgesetzt wird.

**UEPTRMNL**

Adresse der 4-Byte-Terminal-ID für die Terminal- oder Hauptfunktion, über die die Aktualisierung, die zurückgesetzt wird, durchgeführt wurde.

**UEPTASK**

Adresse des 4-Byte-Felds (gepackt dezimal) mit der Tasknummer derjenigen Task, unter der die Aktualisierung durchgeführt wurde, die zurückgesetzt wird.

**UEPFCRSP**

Adresse des Antwortbyte der Dateisteuerung. Mögliche Werte:

**UEAIXFUL**

Kein Speicherplatz in nicht eindeutigem Alternativindex.

**UECACHE**

RLS-Cachefehler oder Cachekonnektivitätsfehler.

**UENBWBAK**

Nicht-BWO-Sicherung in Bearbeitung.

**UEDLOCK**

Deadlock erkannt.

**UEDUPREC**

Doppelter Schlüssel in eindeutigem Alternativindex.

**UEIOEROR**

E/A-Fehler.

**UELCKFUL**

RLS-Sperrstruktur ist voll.

**UENOLDEL**

Logisches Löschen nicht ausgeführt (entweder ist der Exitpunkt XFCLDEL nicht aktiviert oder das globale Benutzer-  
ausgangsprogramm XFCLDEL hat sich gegen die Ausführung des logischen Löschvorgangs entschieden).

**UENOSPAC**

Dataset verfügt nicht über ausreichend Speicherplatz.

**UEOPENER**

Fehler beim Öffnen der Datei.

**UERLSERR**

SMSVSAM RLS-Serverfehler.

**UERLSDIS**

RLS-Zugriff ist derzeit inaktiviert.

**UERLSCON**

Versuch, einen Thread mit einer neuen Instanz des SMSV-SAM RLS-Servers fortzusetzen.

**UEUNEXP**

Unerwarteter Fehler.

### **UEPERR**

Adresse eines 1-Byte-Felds mit dem Fehlertyp. Die Werte des Fehlerbyte und ihre jeweilige Bedeutung werden unter „Werte des von UEPERR referenzierten Byte für den Fehlertyp“ auf Seite 132 beschrieben.

### **Rückgabecodes**

#### **UERCNORM**

Verarbeitung fortsetzen und CICS-Steuerung für Backoutfehler aufrufen.

Dies bewirkt die Ausgabe einer Fehlernachricht zum Backoutfehler (DFHFC4701 für eine VSAM-Datei und DFHFC4702 für eine BDAM-Datei). Bei einer VSAM-Datei konvertiert CICS die Datensatzsperrung in eine beibehaltene Sperrung, und der Protokollsatz wird für eine spätere Wiederholung des Backouts gespeichert.

#### **UERCBYD**

Fehler ignorieren (CICS-Steuerung für Backoutfehler nicht aufrufen) und fortsetzen. Das Festlegen dieses Rückgabecodes kann unter Umständen die Integrität Ihrer Daten beeinträchtigen.

Der Rückgabecode UERCPURG ist nicht zulässig. Es ist nicht erforderlich, den Rückgabecode UERCPURG festzulegen, da die Bedingungen, unter denen dieser Exit aufgerufen wird, bedeuten, dass eine Bedingung vom Typ 'bereinigt' von keinem XPI- oder API-Aufruf zurückgegeben werden kann.

### **XPI-Aufrufe**

Es können alle XPI-Aufrufe verwendet werden; hierbei gelten jedoch dieselben Hinweise zur Vorsicht wie bei API- und SPI-Aufrufen.

### **API- und SPI-Aufrufe**

Obwohl dieser Exit berechtigt ist, API- und SPI-Aufrufe auszugeben, sollten Sie bei der Wahl der verwendeten Befehle sehr vorsichtig sein, da der Exit während des Dateibackouts aufgerufen wird, der Teil der Synchronisationspunktphase 2 ist.

Es wird empfohlen, EXEC CICS-Befehle auf Abfragen zu beschränken und Befehle zu vermeiden, mit denen CICS-Ressourcen aktualisiert werden, da sich die Ressourcen selbst in einem Status der Wiederherstellung befinden können. Hierbei gelten insbesondere die folgenden Einschränkungen:

1. Geben Sie keine wiederherstellbaren Operationen aus.
2. Verwenden Sie keine Operationen, die auf Systeme oder Ressourceneigner außerhalb dieses CICS-Systems zugreifen, selbst wenn die Zielressource nicht wiederherstellbar ist.
3. Inaktivieren Sie keine Dateien und schließen Sie auch keine Dateien, da dies zu weiteren Fehlerbedingungen führen könnte.
4. Es ist möglich, diesen Exit in einer Transaktionsumgebung aufzurufen, die sich von der Transaktionsumgebung unterscheidet, in der die Aktualisierungen, die zurückgesetzt werden, ursprünglich vorgenommen wurden. Wenn Ihr Exitprogramm Aktionen ausführen will (beispielsweise das Schreiben einer Nachricht an das Terminal), die voraussetzen, dass das Programm in der ursprünglichen Transaktionsumgebung ausgeführt wird, muss zunächst der Wert überprüft werden, der im Parameter RE\_ATTACHED\_TRANSACTION eines XPI-Aufrufs vom Typ INQUIRE\_TRANSACTION des Transaktionsmanagers zurückgegeben wird.

## Beispiel eines globalen Benutzerexitprogramm für XFCBFAIL

DFH\$FCBF stellt eine Beispielverarbeitung für den globalen Benutzerexit XFCBFAIL beim Fehlschlagen eines Dateisteuerungsbackouts bereit. Weitere Informationen finden Sie unter File control recovery sample exit program: DFH\$FCBF.

### Werte des von UEPERR referenzierten Byte für den Fehlertyp

Das Feld UEPERR in der Parameterliste für XFCBFAIL verweist auf ein Byte für den Fehlertyp. Dieses Byte enthält einen von mehreren möglichen Werten.

Mögliche Werte:

#### **XBFERU**

Bei der Verarbeitung einer Anforderung vom Typ READ UPDATE wurde vom Dateisteuerprogramm für den Dateianforderungshandler eine Fehlerantwort zurückgegeben. Diese Anforderung wird durch das Dateisteuerungsbackout ausgegeben, um die vorhandene Kopie des Datensatzes abzurufen, bevor dieser zurückgesetzt wird.

Verwenden Sie UEPFCRSP in Kombination mit dem Datensatztyp, der im Dateisteuerungsabschnitt des Protokollsatzes angezeigt wird, der durch den Parameter UEPBLOGR adressiert wird, um das spezifische Problem zu ermitteln. Der von UEPBLOGR adressierte Speicherbereich enthält entweder das Vorimage eines Datensatzes vom Typ „read-update“ oder die neue Kopie eines Datensatzes vom Typ „write-add“, der gelöscht werden soll. Das Feld für den Datensatztyp, FLJB\_RECORD\_TYPE, wird im Pseudobereich (DSECT) DFHFCLGD definiert.

#### **XBFERE**

Bei der Verarbeitung einer Anforderung vom Typ REWRITE wurde vom Dateisteuerprogramm für den Dateianforderungshandler eine Fehlerantwort zurückgegeben. Diese Anforderung wird durch das Dateisteuerungsbackout ausgegeben, um die vorhandene Kopie des Datensatzes im Dataset durch das „Vorimage“ zu ersetzen, das in dem von UEPBLOGR adressierten Protokollsatz enthalten ist. Verwenden Sie den Parameter UEPFCRSP, um zu ermitteln, welcher Fehler aufgetreten ist.

#### **XBFEWR**

Bei der Verarbeitung einer Anforderung vom Typ WRITE wurde vom Dateisteuerprogramm für den Dateianforderungshandler eine Fehlerantwort zurückgegeben. Diese Anforderung wird durch das Dateisteuerungsbackout ausgegeben, um das „Vorimage“ eines gelöschten Datensatzes hinzuzufügen. Verwenden Sie den Parameter UEPFCRSP, um zu ermitteln, welcher Fehler aufgetreten ist.

#### **XBFEDL**

Bei der Verarbeitung einer Anforderung vom Typ REWRITE DELETE wurde vom Dateisteuerprogramm für den Dateianforderungshandler eine Fehlerantwort zurückgegeben. Diese Anforderung wird durch das Dateisteuerungsbackout ausgegeben, um einen neuen Datensatz zu löschen, der zu einem VSAM-Dataset hinzugefügt wurde. Verwenden Sie den Parameter UEPFCRSP, um zu ermitteln, welcher Fehler aufgetreten ist.

#### **XBFENO**

Der Fehler, der während des Dateisteuerungsbackouts aufgetreten ist, war nicht das Ergebnis einer Fehlerantwort, die durch das Dateisteuerprogramm für den Dateianforderungshandler ausgegeben wurde. Verwenden Sie den Parameter UEPFCRSP, um zu ermitteln, welcher Fehler aufgetreten ist.

## Exit XFCBOUT für Dateisteuerungsbackout

Der Exit XFCBOUT wird aufgerufen, wenn eine Dateisteuerungsaktualisierung im Begriff ist, zurückgesetzt zu werden. Der Protokollsatz, der das Vorimage des Datensatzes enthält, der zurückgesetzt wird, wird an das Exitprogramm übergeben.

XFCBOUT stellt keinen Rückgabecode bereit, mit dem Ihr Exitprogramm das Backout der Aktualisierung umgehen kann, da dies zu einem Datenverlust führen würde. Der Protokollsatz enthält jedoch den Dateinamen, sodass Ihr Exitprogramm einen Befehl vom Typ **EXEC CICS INQUIRE FILE** verwenden kann, um Informationen zu dieser Datei abzurufen.

### Zeitpunkt des Aufrufs

Wenn eine Aktualisierung (die durch einen Protokollsatz mit Vorimage dargestellt wird) im Begriff ist, durch die Dateisteuerung zurückgesetzt zu werden.

### Exitspezifische Parameter

#### UEPFLOGR

Adresse des Dateisteuerungsabschnitts des Protokollsatzes, der für das Backout angezeigt wird. Diese Zuordnung erfolgt durch den Pseudobereich (DSECT) DFHFCLGD.

### Rückgabecodes

#### UERCNORM

Verarbeitung fortsetzen.

Der Rückgabecode UERCPURG ist nicht zulässig. Es ist nicht erforderlich, den Rückgabecode UERCPURG festzulegen, da dieser Exit während der Synchronisationspunktphase 2 aufgerufen wird und daher von keinem Aufruf, den er ausgibt, eine Antwort vom Typ 'Bereinigt' erhalten kann.

### XPI-Aufrufe

Es können alle XPI-Aufrufe verwendet werden; hierbei gelten jedoch dieselben Hinweise zur Vorsicht wie bei API- und SPI-Aufrufen.

### API- und SPI-Aufrufe

Obwohl dieser Exit berechtigt ist, API- und SPI-Aufrufe auszugeben, sollten Sie bei der Wahl der verwendeten Befehle sehr vorsichtig sein, da der Exit während des Dateibackouts aufgerufen wird, der Teil der Synchronisationspunktphase 2 ist.

Es wird empfohlen, EXEC CICS-Befehle auf Abfragen zu beschränken und Befehle zu vermeiden, mit denen CICS-Ressourcen aktualisiert werden, da sich die Ressourcen selbst in einem Status der Wiederherstellung befinden können. Hierbei gelten insbesondere die folgenden Einschränkungen:

1. Geben Sie keine wiederherstellbaren Operationen aus.
2. Verwenden Sie keine Operationen, die auf Systeme oder Ressourceneigner außerhalb dieses CICS-Systems zugreifen, selbst wenn die Zielressource nicht wiederherstellbar ist.
3. Inaktivieren Sie keine Dateien und schließen Sie auch keine Dateien, da dies zu weiteren Fehlerbedingungen führen könnte.
4. Es ist möglich, diesen Exit in einer Transaktionsumgebung aufzurufen, die sich von der Transaktionsumgebung unterscheidet, in der die Aktualisierungen, die zurückgesetzt werden, ursprünglich vorgenommen wurden. Wenn Ihr Exitprogramm Aktionen ausführen will (beispielsweise das Schreiben einer Nachricht an das Terminal), die voraussetzen, dass das Programm in der ursprünglichen Transaktionsumgebung aus-

geführt wird, muss zunächst der Wert überprüft werden, der im Parameter RE\_ATTACHED\_TRANSACTION eines XPI-Aufrufs vom Typ INQUIRE\_TRANSACTION des Transaktionsmanagers zurückgegeben wird.

Da davon ausgegangen wird, dass XFCBOUT für bestimmte Anwendungen verwendet wird, wird kein universelles Beispiexitprogramm bereitgestellt. Als Basis für ein Exitprogramm für XFCBOUT können Sie beliebig eines der Beispielpprogramme für die anderen Exits zur Wiederherstellung der Dateisteuerung verwenden, also DFH\$FCBF, DFH\$FCBV oder DFH\$FCLD.

## **Exit XFCBOVER bei Überschreibung eines Dateisteuerungsbackouts**

Der Exit XFCBOVER ist Teil der Unterstützung, die die CICS-Dateisteuerung für *Stapelverarbeitungsfenster* in einer VSAM-RLS-Umgebung bietet.

VSAM RLS sperrt einzelne Datensätze in einem Dataset, und diese Sperren werden für diejenigen Arbeitseinheiten (UOWs), die aufgrund von Backoutfehlern oder unbestätigten Fehlern nicht abgeschlossen sind, in beibehaltene Sperren konvertiert, wodurch die Datenintegrität erhalten bleibt. Um zu vermeiden, dass ein Dataset durch einen Nicht-RLS-Batch-Job beschädigt wird, der die beibehaltenen Datensatzsperren nicht kennt, kann ein Dataset normalerweise nicht für Aktualisierungen im Nicht-RLS-Modus geöffnet werden, wenn er gesperrte Datensätze enthält.

### **Beispiel eines globalen Benutzerexitprogramms für XFCBOVER**

DFH\$FCBV stellt eine Beispielverarbeitung für den globalen Benutzerexit XFCBOVER beim Überschreiben eines Dateisteuerungsbackouts bereit. Weitere Informationen finden Sie unter File control recovery sample exit program: DFH\$FCBV.

### **Überschreibung beibehaltener Sperren für Stapel**

Es kann Situationen geben, in denen Sie diese Sperren außer Kraft setzen und das Öffnen eines Datasets für die Stapelverarbeitung erzwingen wollen.

Dies kann beispielsweise in folgenden Situationen der Fall sein:

- Vor Ausführung des Stapeljobs ist nicht genügend Zeit vorhanden, um die Situation zu beheben, die zum Sperrend der Datensätze geführt hat.
- Es ist bekannt, dass der Stapeljob die Datenintegrität nicht beeinträchtigen kann (da er keine vorhandenen Datensätze im Dataset aktualisiert bzw. keine Datensätze aktualisiert, die möglicherweise von CICS aktualisiert worden sind).

Um die Einschränkung für Öffnen außer Kraft zu setzen, stellt VSAM RLS den Befehl SHCDS PERMITNONRLSUPDATE bereit, um es einem Nicht-RLS-Stapeljob zu ermöglichen, auch dann einen Bereich für Aktualisierungen zu öffnen, wenn beibehaltene Sperren vorhanden sind.

### **Auswirkungen des Überschreibens beibehaltener Sperren auf CICS**

In VSAM wird die Verwendung der Option zum Überschreiben beibehaltener Sperren erfasst, damit eine CICS-Region beim nächsten Öffnen des betreffenden Datasets entsprechend benachrichtigt werden kann. Da Daten durch den Nicht-RLS-Stapeljob geändert worden sein können, sind die Ergebnisse von Wiederherstellungsoperationen, die von CICS durchgeführt werden (beispielsweise für Arbeitseinheiten (UOWs), die zum Zeitpunkt des Stapeljobs den Status 'Backout fehlgeschlagen' oder 'Unbestätigte Fehler' aufwiesen), unvorhersehbar. In die-



sem Fall besteht die Standardaktion von CICS daher darin, Aktualisierungen, die zum Zeitpunkt der Überschreibung der Sperren noch ausstanden, nicht zurückzusetzen und zu jedem ignorierten Backout entsprechende Diagnoseinformationen in die CSFL-Warteschlange mit transienten Daten zu schreiben.

Der globale Benutzerexit XFCBOVER soll es Ihnen ermöglichen, für jeden UOW-Protokolleintrag, für den ein Backout ignoriert wird, folgende Schritte auszuführen:

- Schreiben von anwendungsbezogenen Diagnoseinformationen als Ergänzung zu den von CICS bereitgestellten Informationen.
- Durchführung von anwendungsbezogenen Wiederherstellungsaktionen.
- Zurücknehmen der Standardaktion durch Anforderung, dass das Backout doch durchgeführt werden soll. Diese Option ist erforderlich, wenn bekannt ist, dass der betreffende Stapeljob die Datenintegrität nicht beschädigt (beispielsweise weil er Datensätze nur einfügt).

#### **Zeitpunkt des Aufrufs**

Sobald CICS im Begriff ist, einen UOW-Protokolleintrag, der zurückgesetzt werden soll, zu ignorieren, weil die Sperre zum Schutz des aktualisierten Datensatzes von einem Nicht-RLS-Stapelverarbeitungsprogramm überschrieben worden sein könnte.

#### **Exit-spezifische Parameter**

##### **UEPOLOGR**

Adresse des Dateisteuerungsabschnitts eines verzögerten Protokolleintrags, der eine Aktualisierung an einem Dataset darstellt, für das beibehaltene Sperren möglicherweise überschrieben worden sind. Der Dateisteuerungsabschnitt des Protokolleintrags kann mithilfe des Pseudobereichs (DSECT) DFHFCLGD zugeordnet werden.

##### **UEPODSN**

Adresse eines 44-Byte-Speicherbereichs mit dem Namen des Datensatzes, dessen Sperren überschrieben wurden.

#### **Rückgabecodes**

##### **UERCNORM**

Das Backout des betreffenden Protokolleintrags nicht durchführen. Alle durch den Stapelverarbeitungsablauf ausgeführten Aktualisierungen sollen Vorrang haben.

##### **UERCCKO**

Das Backout durchführen. Es ist bekannt, dass die Aktionen des Stapeljobs diese Aktualisierung nicht beeinflusst haben können.

Der Rückgabecode UERCPURG ist nicht zulässig. Es ist nicht erforderlich, den Rückgabecode UERCPURG festzulegen, da dieser globale Benutzerexit während der Synchronisationspunktphase 2 aufgerufen wird und daher von keinem Aufruf, den er ausgibt, eine Antwort vom Typ 'Bereinigt' erhalten kann.

#### **XPI-Aufrufe**

Es können alle XPI-Aufrufe verwendet werden; hierbei gelten jedoch dieselben Hinweise zur Vorsicht wie bei API- und SPI-Aufrufen.

#### **API- und SPI-Aufrufe**

Obwohl dieser Exit berechtigt ist, API- und SPI-Aufrufe auszugeben, sollten Sie bei der Wahl der verwendeten Befehle sehr vorsichtig sein, da der Exit während eines Dateibackouts aufgerufen wird, das Teil der Synchronisationspunktphase 2 ist.

Es wird empfohlen, EXEC CICS-Befehle auf Abfragen zu beschränken und Befehle zu vermeiden, mit denen CICS-Ressourcen aktualisiert werden, da sich die Ressourcen selbst in einem Status der Wiederherstellung befinden können. Hierbei gelten insbesondere die folgenden Einschränkungen:

1. Geben Sie keine wiederherstellbaren Operationen aus.
2. Verwenden Sie keine Operationen, die auf Systeme oder Ressourceneigner außerhalb dieses CICS-Systems zugreifen, selbst wenn die Zielresource nicht wiederherstellbar ist.
3. Inaktivieren Sie keine Dateien und schließen Sie auch keine Dateien, da dies zu weiteren Fehlerbedingungen führen könnte.
4. Es ist möglich, diesen Exit in einer Transaktionsumgebung aufzurufen, die sich von der Transaktionsumgebung unterscheidet, in der die Aktualisierungen, die zurückgesetzt werden, ursprünglich vorgenommen wurden. Wenn Ihr Exitprogramm Aktionen ausführen will (beispielsweise das Schreiben einer Nachricht an das Terminal), die voraussetzen, dass das Programm in der ursprünglichen Transaktionsumgebung ausgeführt wird, muss zunächst der Wert überprüft werden, der im Parameter `RE_ATTACHED_TRANSACTION` eines XPI-Aufrufs vom Typ `INQUIRE_TRANSACTION` des Transaktionsmanagers zurückgegeben wird.

## Exit XFCLDEL zur Dateisteuerung beim logischen Löschen

Der Exit XFCLDEL wird immer dann aufgerufen, wenn ein WRITE-Anforderung an eine VSAM-ESDS oder BDAM-Datei zurückgesetzt wird. Da diese Dateitypen keine Unterstützung zum Löschen bieten, können Sie mithilfe von XFCLDEL einen logischen Löschvorgang ausführen, indem Sie den Eintrag so ändern, dass er als gelöscht markiert wird.

### Exitspezifische Parameter

#### UEPBLOGR

Adresse des Dateisteuerungsabschnitts des Protokollsatzes, der die Aktualisierung darstellt, die durch logisches Löschen zurückgesetzt werden soll. Der Protokollsatz kann mithilfe des Pseudobereichs (DSECT) `DFHFCLGD` zugeordnet werden.

#### UEPTRANS

Adresse einer 4-Byte-Transaktions-ID, unter der die Aktualisierung durchgeführt wurde, die zurückgesetzt wird.

#### UEPTRMNL

Adresse der 4-Byte-Terminal-ID für die Terminal- oder Hauptfunktion, über die die Aktualisierung, die zurückgesetzt wird, durchgeführt wurde.

#### UEPTASK

Adresse der 4-Byte-Tasknummer (gepackt dezimal) derjenigen Task, unter der die Aktualisierung durchgeführt wurde, die zurückgesetzt wird.

#### UEPFDATA

Adresse eines Felds variabler Länge, das die Daten in der Dateisteueranforderung enthält. Das Exitprogramm kann die von diesem Feld adressierten Datensatzdaten ändern und so kennzeichnen, dass Anwendungen die Darstellung dieser Daten als logisch gelöschten Datensatz erkennen können.

## **UEPFLEN**

Adresse eines Vollworts mit der Länge der Daten in der Dateisteueranforderung.

## **Rückgabecodes**

### **UERCFAIL**

Den logischen Löschvorgang nicht durchführen und diesen Vorgang als Backoutfehler behandeln. Dies ist die Standardaktion, die ausgeführt wird, wenn der Exit nicht aktiviert ist.

### **UERCLDEL**

Den logischen Löschvorgang durch erneutes Anwenden des aktualisierten Datensatzes durchführen.

Der Rückgabecode UERCPURG ist nicht zulässig. Es ist nicht erforderlich, den Rückgabecode UERCPURG festzulegen, da die Bedingungen, unter denen dieser Exit aufgerufen wird, bedeuten, dass die Bedingung 'bereinigt' von keinem XPI- oder API-Aufruf zurückgegeben werden kann.

## **XPI-Aufrufe**

Es können alle XPI-Aufrufe verwendet werden; hierbei gelten jedoch dieselben Hinweise zur Vorsicht wie bei API- und SPI-Aufrufen.

## **API- und SPI-Aufrufe**

Obwohl dieser Exit berechtigt ist, API- und SPI-Aufrufe auszugeben, sollten Sie bei der Wahl der verwendeten Befehle sehr vorsichtig sein, da der Exit während des Dateibackouts aufgerufen wird, der Teil der Synchronisationspunktphase 2 ist.

Es wird empfohlen, EXEC CICS-Befehle auf Abfragen zu beschränken und Befehle zu vermeiden, mit denen CICS-Ressourcen aktualisiert werden, da sich die Ressourcen selbst in einem Status der Wiederherstellung befinden können. Hierbei gelten insbesondere die folgenden Einschränkungen:

1. Geben Sie keine wiederherstellbaren Operationen aus.
2. Verwenden Sie keine Operationen, die auf Systeme oder Ressourceneigner außerhalb dieses CICS-Systems zugreifen, selbst wenn die Zielressource nicht wiederherstellbar ist.
3. Inaktivieren Sie keine Dateien und schließen Sie auch keine Dateien, da dies zu weiteren Fehlerbedingungen führen könnte.
4. Es ist möglich, diesen Exit in einer Transaktionsumgebung aufzurufen, die sich von der Transaktionsumgebung unterscheidet, in der die Aktualisierungen, die zurückgesetzt werden, ursprünglich vorgenommen wurden. Wenn Ihr Exitprogramm Aktionen ausführen will (beispielsweise das Schreiben einer Nachricht an das Terminal), die voraussetzen, dass das Programm in der ursprünglichen Transaktionsumgebung ausgeführt wird, muss zunächst der Wert überprüft werden, der im Parameter RE\_ATTACHED\_TRANSACTION eines XPI-Aufrufs vom Typ INQUIRE\_TRANSACTION des Transaktionsmanagers zurückgegeben wird.

In der CICS-Dateidefinition braucht nicht UPDATE=YES angegeben zu werden, damit CICS den Datensatz anhand des Flags für logisches Löschen aktualisiert, das von einem Benutzerexitprogramm für XFCLDEL, wie beispielsweise DFH\$FCLD, gesetzt wird. Bei der Durchführung eines Backouts wird die Prüfung von SERVREQs wie beispielsweise UPDATE umgangen.

## Beispiel eines globalen Benutzerexitprogramms für XFCLDEL

DFH\$FCLD stellt eine Beispielverarbeitung für den globalen Benutzerexit XFCLDEL zur Dateisteuerung beim logischen Löschen bereit. Weitere Informationen finden Sie unter File control recovery sample exit program: DFH\$FCLD.

---

## Exit XFCRLSCO für das Dateisteuerprogramm für RLS-Koexistenz

Der Exit XFCRLSCO kann während einer Anforderung zum Öffnen einer Datei aufgerufen werden. Mit diesem Exit können Sie es einer Anwendung ermöglichen, zwischen dem Zugriffsmodus mit RLS-Dateien und dem Zugriffsmodus mit schreibgeschützten Nicht-RLS-Dateien zu wechseln, um auf ein bestimmtes Dataset zuzugreifen.

In CICS ist es nicht zulässig, eine Öffnungsanforderung für eine Nicht-RLS-Datei auszuführen, wenn eine RLS-Datei bereits für dasselbe Dataset geöffnet ist oder wenn für das Dataset noch eine ausstehende RLS-Verarbeitung im System vorliegt. Darüber hinaus ist es in CICS nicht zulässig, eine Öffnungsanforderung für eine RLS-Datei auszuführen, wenn eine vorhandene Nicht-RLS-Datei bereits für das Basisdataset geöffnet ist. Wenn in diesen Situationen eine Öffnungsanforderung vorkommt und die Nicht-RLS-Datei für Lesezugriff geöffnet ist, wird der Exit XFCRLSCO gesteuert. Sie können diesen Exit verwenden, um zu entscheiden, ob die Öffnungsanforderung fortgesetzt werden darf oder fehlschlagen soll; hierfür steht wie gewohnt die Nachricht DFHFC0511 bzw. DFHFC0512 zur Verfügung.

Um den Zugriffsmodus zu wechseln, kann die Anwendung das Dataset mithilfe einer neuen Datei mit einem anderen Zugriffsmodus öffnen. Verwenden Sie nicht beide Zugriffsmethoden über einen längeren Zeitraum gleichzeitig, um dasselbe Dataset geöffnet zu halten, da CICS kein einheitliches Bild des Datensets erhält, wenn darauf gleichzeitig mit RLS- und Nicht-RLS-Dateien zugegriffen wird. Insbesondere erhält CICS dann kein einheitliches Bild, wenn das Dataset von der Nicht-RLS-Datei gelesen und gleichzeitig von der RLS-Datei aktualisiert wird.

Wenn die Verarbeitung der VSAM-Upgradegruppe stattfindet, während das Dataset sowohl mit RLS-Dateien als auch mit Nicht-RLS-Dateien geöffnet ist, besteht ein erhöhtes Risiko für das Auftreten von Lesefehlern, da die Upgradeverarbeitung weder im Basiscluster noch in den zugeordneten Alternativindizes abgeschlossen worden ist.

### Anmerkung:

1. Der Exit wird nicht gesteuert, wenn für die Nicht-RLS-Datei SERVREQ-Indikatoren für Aktualisierbarkeit festgelegt sind, d. h. wenn Aktualisierungen, Hinzufügungen oder Löschungen zulässig sind.
2. Das Dataset, das geöffnet wird, muss Optionen für gemeinsame Nutzung der Stufe SHAREOPTION(2) im VSAM-Basiscluster angeben. Werden niedrigere Optionen für gemeinsame Nutzung angegeben, schlägt die zweite Öffnungsanforderung in VSAM fehl.
3. Bei Verwendung der statischen Zuordnung muss unbedingt DISP=SHR angegeben werden, da ansonsten die Öffnungsanforderung in VSAM fehlschlägt.

### Zeitpunkt des Aufrufs

Während einer OPEN-Anforderung zum Öffnen einer Datei, bevor die Öffnungsanforderung an VSAM ausgegeben wird.

### Exitspezifische Parameter

**UEPFILEN**

Adresse eines 8-Byte-Felds mit dem Dateinamen. Wenn der Dateiname weniger als 8 Zeichen lang ist, wird er mit Leerzeichen aufgefüllt.

**UEPDSNAME**

Adresse eines 44-Byte-Felds mit dem Namen des Basisdatasets. Wenn der Name des Datasets weniger als 44 Zeichen lang ist, wird er mit Leerzeichen aufgefüllt.

**UEPFSERV**

Adresse eines 1-Byte-Felds mit dem SERVREQ-Indikator für Dateien. Mögliche Werte:

**UEPFRDIM**

Indikator für 'Lesen gültig'.

**UEPFUPDIM**

Indikator für 'Aktualisieren gültig'.

**UEPFADDIM**

Indikator für 'Hinzufügen gültig'.

**UEPFDELIM**

Indikator für 'Löschen gültig'.

**UEPFBRZIM**

Indikator für 'Durchsuchen gültig'.

**UEPFDSACC**

Adresse eines 1-Byte-Felds mit dem Flag für die Dateizugriffsmethode. Mögliche Werte:

**UEPFVSAM**

Indikator für VSAM-Dateien.

**UEPFDTBL**

Indikator für Datentabellendateien.

**UEPFDTUM**

Indikator für Benutzerdatentabellendateien.

**UEPFRLS**

Indikator für RLS-Dateien.

**UEPFCFDT**

Indikator für CFDT-Dateien.

**UEPRECUR**

Adresse der Halbwort-Rekursionsebene.

**Rückgabecodes****UERCNORM**

Verarbeitung normal fortsetzen. Die Öffnungsanforderung schlägt fehl.

**UERCBYP**

Öffnungsanforderung zulassen und Koexistenzfehler umgehen.

**XPI-Aufrufe**

Es können XPI-Befehle verwendet werden. Diese Befehle dürfen jedoch nicht zu Statusänderungen an Dateien führen.

#### API- und SPI-Aufrufe

Es können EXEC CICS-API-Befehle und EXEC CICS-SPI-Befehle verwendet werden. Diese Befehle dürfen jedoch keine Statusänderungen an Dateien vornehmen. So können Sie beispielsweise den Befehl **EXEC CICS INQUIRE FILE** verwenden, nicht jedoch den Befehl **EXEC CICS SET FILE** und auch keine EXEC CICS-API-Befehle für Dateisteuerung, die zu Statusänderungen an Dateien führen.

---

## Exit für das Programm für Guten-Morgen-Nachricht (XGMTEXT)

Dieser Exit wird aufgerufen, bevor eine Guten-Morgen-Nachricht („good morning“) gesendet wird.

#### Zeitpunkt des Aufrufs

Vor Übertragung der Guten-Morgen-Nachricht.

#### Exitspezifische Parameter

##### UEPTCTTE

Adresse des Terminaleintrags in der Terminalsteuertabelle (TCTTE). Der TCTTE kann mithilfe des Pseudobereichs (DSECT) DFHTCTTE zugeordnet werden.

##### UEPTIOA

Adresse des Terminal-E/A-Bereichs (TIOA). Der TIOA kann mithilfe des Pseudobereichs (DSECT) DFHTIOA zugeordnet werden. Bei den Feldern TIOASAL und TIOASCA handelt es sich jedoch **nicht** um Programmierschnittstellen.

#### Rückgabecodes

##### UERCNORM

Verarbeitung fortsetzen.

##### UERCPUrg

Task während des XPI-Aufrufs bereinigt.

#### XPI-Aufrufe

Es können alle verwendet werden.

---

## HTTP-Client-Exits für Öffnen und Senden: XWBAUTH, XWBOPEN und XWBSNDO

Die Exits XWBAUTH, XWBOPEN und XWBSNDO werden während der Verarbeitung eines Befehls vom Typ **EXEC CICS WEB CONVERSE**, **EXEC CICS WEB OPEN**, **EXEC CICS INVOKE SERVICE** und **EXEC CICS WEB SEND** aufgerufen. Sie werden beim Ausgeben von HTTP-Clientanforderungen über CICS als HTTP-Client verwendet. Dies ist eine Funktion, die von der CICS-Webunterstützung bereitgestellt wird.

### Sendeexit XWBAUTH für HTTP-Client

Mithilfe von XWBAUTH können Sie Berechtigungsnachweise für die Basisauthentifizierung (Benutzername und Kennwort) für einen Zielsever oder einen Service-Provider angeben. XWBAUTH übergibt diese Berechtigungsnachweise auf Anforderung an CICS, um einen Berechtigungsheader zu erstellen, der mit HTTP weitergeleitet wird.

Wenn Sie AUTHENTICATE(BASICAUTH) im Befehl **EXEC CICS WEB SEND** (Client) oder **WEB CONVERSE** angeben, dann kann die Anwendung einen Benutzernamen und ein Kennwort bereitstellen. Werden kein Benutzername und kein Kenn-

wort bereitgestellt, wird XWBAUTH aufgerufen; dies stellt eine alternative Methode zum Angeben dieser Berechtigungsnachweise dar. XWBAUTH wird aufgerufen, wenn Sie AUTHENTICATE(BASIC) in einer URIMAP-Ressourcendefinition für USAGE(CLIENT) angeben, sofern die Anwendung selber keinen Benutzernamen und kein Kennwort im Befehl **EXEC CICS WEB SEND** (Client) oder **WEB CONVERSE** bereitstellt. XWBAUTH wird auch mit **EC INVOKE SERVICE()** **URIMAP()** aufgerufen, wenn URIMAP die Angabe AUTHENTICATE(BASIC) enthält.

Der Benutzername und das Kennwort gelten in der Regel spezifisch für die ferne Serverumgebung und können länger sein als die von RACF-Systemen standardmäßig verwendeten acht Zeichen. Die Felder für den Benutzernamen und das Kennwort können eine Länge von bis zu 256 Zeichen haben. Die Syntax dieser Felder wird nicht überprüft.

Der Host wird als Parameter UEPHOST an das Benutzerexitprogramm übergeben, und der Pfad wird als Parameter UEPPATH übergeben. Das Realm wird optional als Parameter UEPREALM übergeben. Als Antwort gibt das Benutzerexitprogramm den Benutzernamen und das Kennwort als Parameter UEPUSNM und UEPPSWD zurück. Bei der Codierung der bereitgestellten Benutzer-ID und des bereitgestellten Kennworts verwendet CICS die EBCDIC-Codepages.

In der CICS-Beispielbibliothek SDFHSAMP werden die folgenden Beispiexitprogramme geliefert:

- DFH\$WBPI
- DFH\$WBEX
- DFH\$WBX1
- DFH\$WBX2
- DFH\$WBGA, ein Copybook zwecks Zuordnung des globalen Arbeitsbereichs, der von den Beispielen DFH\$WBPI, DFH\$WBX1, DFH\$WBX2 und DFH\$WBEX verwendet wird.

Weitere Informationen zu den Beispiexitprogrammen für den Client finden Sie unter HTTP client sample exit programs (DFH\$WB\*). Weitere Informationen zum Konfigurieren des LDAP-Profiles finden Sie unter Configuring LDAP for CICS use.

## Exit XWBAUTH

### Zeitpunkt des Aufrufs

Wenn im Befehl **EXEC CICS WEB SEND** oder **WEB CONVERSE** die Option AUTHENTICATE(BASICAUTH) angegeben ist, aber kein Benutzername (USERNAME) und kein Kennwort (PASSWORD) angegeben werden.

### Exitspezifische Parameter

#### UEPHOST (Eingabe von CICS)

Adresse eines Felds mit der Adresse des Hostnamens, der IPv4- oder der IPv6-Adresse, die in der Option HOST des Befehls WEB OPEN für die Verbindung angegeben ist. Der Hostname wird beim Speichern in diesem Feld in Kleinbuchstaben konvertiert. Beim Abgleich des Hostnamens muss diese Konvertierung von Ihrem Benutzerexitprogramm berücksichtigt werden.

#### UEPHOSTL (Eingabe von CICS)

Adresse eines Felds mit der Halbwortlänge des Hostnamens.

#### UEPPATH (Eingabe von CICS)

Adresse eines Felds mit der Adresse des Pfads, der in der Option

PATH des Befehls WEB SEND oder WEB CONVERSE angegeben ist. Der Pfad liegt in gemischter Groß-/Kleinschreibung vor, so wie er angegeben wurde.

**UEPPATHL (Eingabe von CICS)**

Adresse eines Felds mit der Halbwortlänge des Pfads.

**UEPREALM (Eingabe von CICS)**

Adresse eines Felds mit der Adresse des Realmnamens, der dem Ziel zugeordnet ist, sofern ein Realmname in einer vorherigen HTTP-Antwort vom Typ 401 vom Server zurückgegeben wurde.

**UEPREALML (Eingabe von CICS)**

Adresse eines Felds mit der Halbwortlänge des Realmnamens.

**UEPAUTHT (Eingabe von CICS)**

Adresse eines 1-Byte-Codes, der den Authentifizierungstyp angibt. Bei diesem Code handelt es sich um die Binärzahl 01 zur Angabe der Basisauthentifizierung.

**UEPUSNM (Ausgabe vom Benutzerexit)**

Adresse eines Vollwortfelds mit der Adresse des Benutzernamens, der für den Zugriff auf den HTTP-Server erforderlich ist. Zum Speichern des Benutzernamens erstellt CICS eine vordefinierte Adresse und einen 64-Byte-Bereich. Sie können Ihren Benutzernamen in diesen 64-Byte-Bereich stellen und dabei die Adresse in UEPUSNM unverändert lassen. Alternativ können Sie Ihren Benutzernamen auch in Ihren eigenen Bereich stellen und die Adresse in UEPUSNM durch Ihre Benutzernamensadresse ersetzen. Wenn Sie einen eigenen Benutzernamensbereich erstellen, kann das Feld bis zu 256 Byte lang sein.

**UEPUSNML (Eingabe von CICS und Ausgabe vom Benutzerexit)**

Adresse eines Halbwortfeldes, das anfänglich die Länge der in UEPUSNM angegebenen Pufferadresse enthält. Ihr Benutzerexitprogramm muss die Länge dieses Puffers auf die Länge des Benutzernamens setzen, so wie in UEPUSNM angegeben.

**UEPPSWD (Ausgabe vom Benutzerexit)**

Adresse eines Vollwortfelds mit der Adresse des Kennworts, das für den Zugriff auf den HTTP-Server erforderlich ist. Zum Speichern des Kennworts oder der Kennwortphrase erstellt CICS eine vordefinierte Adresse und einen 100-Byte-Bereich. Sie können Ihr Kennwort in diesen 100-Byte-Bereich stellen und dabei die Adresse in UEPPSWD unverändert lassen. Alternativ können Sie Ihr Kennwort auch in Ihren eigenen Bereich stellen und die Adresse in UEPPSWD durch die Adresse Ihres Kennworts ersetzen. Wenn Sie einen eigenen Kennwortbereich erstellen, kann das Feld bis zu 256 Byte lang sein.

**UEPPSWDL (Eingabe von CICS und Ausgabe vom Benutzerexit)**

Adresse eines Halbwortfeldes, das anfänglich die Länge der in UEPPSWD angegebenen Pufferadresse enthält. Ihr Benutzerexitprogramm muss die Länge dieses Puffers auf die tatsächliche Länge des Kennworts setzen, so wie in UEPPSWD angegeben.

**UEPHOSTT (Eingabe von CICS)**

Adresse eines 1-Byte-Codes, der den Hosttyp angibt, der im Parameter UEPHOST enthalten ist.

Die Binärzahl 01 steht für einen Hostnamen, die Binärzahl 02 für eine IPv4-Adresse und die Binärzahl 03 für eine IPv6-Adresse.



## Rückgabecodes

### UERCNORM

Der Exit hat erfolgreich einen Benutzernamen und ein Kennwort zurückgegeben.

### UERCBYP

Der Exit kann keinen Benutzernamen und kein Kennwort angeben. Es wird kein Berechtigungsheader gesendet.

### UERCERR

Der Exit kann keinen Benutzernamen und kein Kennwort angeben. Der Befehl WEB SEND (Client) oder WEB CONVERSE muss gestoppt werden.

## XPI-Aufrufe

Es können alle XPI-Aufrufe verwendet werden.

## API- und SPI-Befehle

Es können alle API- und SPI-Befehle verwendet werden, mit Ausnahme von **EXEC CICS SHUTDOWN** und **EXEC CICS XCTL**.

## Typische Verwendung der LDAP-XPI-Funktionen durch XWBAUTH

Im Zusammenhang mit dem globalen Benutzerexit XWBAUTH werden die DFHD-DAPX-Funktionen üblicherweise für Folgendes verwendet: Öffnen und Schließen einer LDAP-Sitzung, Durchsuchen von Ergebnissen für Berechtigungsnachweise, Scannen und Lokalisieren von Ergebnissen, Schließen des Durchsuchungsvorgangs, Rückgabe des korrekten Werts und Schließen der Suche.

### BIND\_LDAP

Erstellt eine Sitzung mit einem LDAP-Server. Wird einmalig beim ersten Aufruf an den globalen Benutzerausgang XWBAUTH verwendet. Das LDAP-Sitzungstoken wird im globalen Arbeitsbereich von XWBAUTH gespeichert (sofern ein solcher bereitgestellt wird) und kann dann von nachfolgenden Aufrufen an SEARCH\_LDAP verwendet werden.

### UNBIND\_LDAP

Gibt die Verbindung zum LDAP-Server frei. Diese Funktion ist nur während der Verarbeitung des CICS-Systemabschlusses erforderlich. Diese Funktion kann gleichzeitig mit dem globalen Benutzerexit XSTERM (Systembeendigung) verwendet werden.

### SEARCH\_LDAP

Sucht nach Berechtigungsnachweisen und gibt einen definierten LDAP-Namen an, der die URL und das Realm der erforderlichen Benutzerinformationen angibt. Der definierte Name wird im folgenden Format angegeben:

```
racfcid=uuuuuuuu, ibm-httprealm=rrrrrrrr, labeledURI=xxxxxxx, cn=BasicAuth
```

Hierbei gilt Folgendes:

- uuuuuuuu ist die aktuelle Benutzer-ID aus dem XWBAUTH-Parameter UEPUSER.
- rrrrrrrr ist das HTTP 401-Realm aus dem XWBAUTH-Parameter UEPREALM (sofern vorhanden).
- xxxxxxxx ist die Ziel-URL, bei der es sich um eine Verknüpfung von http:// mit dem Hostnamen aus dem XWBAUTH-Parameter UEPHOST und dem Pfad aus dem XWBAUTH-Parameter UEPPATH handelt.

- `cn=BasicAuth` ist ein beliebiges Suffix, das für den LDAP-Server konfiguriert wird, um Berechtigungsnachweise für die Basisauthentifizierung zu speichern.

#### **START\_BROWSE\_RESULTS**

Startet das Scannen der Ergebnisse, die von `SEARCH_LDAP` zurückgegeben werden.

#### **GET\_NEXT\_ENTRY**

Lokalisiert den nächsten Ergebniseintrag in einer Reihe von Einträgen, die von `SEARCH_LDAP` zurückgegeben werden. In der Regel wird mit der in `SEARCH_LDAP` angegebenen URL jedoch ein eindeutiger Eintrag gefunden, sodass die Funktion `GET_NEXT_ENTRY` keine Anwendung findet.

#### **GET\_NEXT\_ATTRIBUTE**

Lokalisiert das nächste Attribut im aktuellen Ergebniseintrag. In der Regel werden jedoch spezifische Attribute ausgewählt, sodass die Funktion `GET_NEXT_ATTRIBUTE` keine Anwendung findet.

#### **END\_BROWSE\_RESULTS**

Beendet die Durchsuchungssitzung, die von `SEARCH_LDAP` gestartet wurde.

#### **GET\_ATTRIBUTE\_VALUE**

Gibt die Werte für eine Reihe von Attributen des definierten Zielnamens zurück. Bei `XWBAUTH` handelt es sich bei diesen Attributwerten um den Benutzernamen und das Kennwort, die in den Attributen `uid` und `userpassword` gespeichert sind. `XWBAUTH` gibt diese Attributwerte als Berechtigungsnachweise zurück.

#### **FREE\_SEARCH\_RESULTS**

Schließt die Suche, die von `SEARCH_LDAP` eingeleitet wurde, und gibt den zugeordneten Speicher frei.

## **Exit XWBOPEN für Öffnen für HTTP-Client**

Mit `XWBOPEN` können Sie Proxy-Server angeben, die für HTTP-Anforderungen durch CICS als HTTP-Client verwendet werden. Darüber hinaus können Sie auch eine Sicherheitsrichtlinie auf den Hostnamen anwenden, der für diese Anforderungen angegeben wird.

`XWBOPEN` wird während der Verarbeitung eines Befehls vom Typ **EXEC CICS WEB OPEN** aufgerufen, der von einem Anwendungsprogramm verwendet wird, um eine Verbindung mit einem Server zu öffnen. `XWBOPEN` wird auch während der Verarbeitung eines Befehls vom Typ **EXEC CICS INVOKE SERVICE** aufgerufen.

CICS macht keine Vorgaben bezüglich der Verwendung (oder sonstigen Nutzung) von Proxy-Servern für HTTP-Anforderungen durch CICS als HTTP-Client, und CICS wendet keinerlei Sicherheitsrichtlinien für diese Anforderungen an. Sie müssen diese Funktionen einrichten, wenn sie von Ihrem System oder Ihrer Organisation benötigt werden.

Mit dem Befehl **EXEC CICS WEB OPEN** wird die CICS-Webdomäne angewiesen, eine Verbindung zu einem Server zu öffnen. `XWBOPEN` wird aufgerufen, bevor eine solche Verbindung geöffnet wird. Der Hostname für die Verbindung (beispielsweise `www.example.com`), der durch die Option `HOST` im Befehl **EXEC CICS WEB OPEN** angegeben wird, wird als Parameter `UEPHOST` zur Überprüfung an das Benutzerexitprogramm übergeben. An diesem Punkt können Sie das Benutzerexitprogramm für zwei Zwecke verwenden:

- Um festzustellen, ob für die HTTP-Anforderung ein Proxy-Server verwendet werden muss, und um den Namen eines gegebenenfalls erforderlichen Proxy-Servers zurückzugeben. Wird ein Proxy-Server benötigt, wird der Rückgabecode UERCPROX verwendet, und der Name des Proxy-Servers wird an die CICS-Webdomäne in dem durch UEPPROXY angegebenen Puffer zurückgegeben und verwendet, um die Verbindung zum Server herzustellen. Wird kein Proxy-Server benötigt, wird der Rückgabecode UERCNORM verwendet.
- Um eine Sicherheitsrichtlinie auf den Hostnamen anzuwenden. Der Rückgabecode UERCBARR gibt an, dass der Zugriff auf den Host nicht zulässig ist und dass eine Antwort vom Typ NOTAUTH an den Befehl WEB OPEN zurückgegeben wird. Der Anwendungsprogrammierer darf nicht mehr versuchen, diese Verbindung zu öffnen. Wenn Sie eine Sicherheitsrichtlinie für einzelne Ressourcen sowie für den Host (oder anstelle von diesem) anwenden wollen, verwenden Sie den Benutzerexit XWBSNDO in den Befehlen **EXEC CICS WEB SEND** und **EXEC CICS WEB CONVERSE**, um eine Sicherheitsrichtlinie auf die Pfadkomponente der URL anzuwenden.

Der Benutzerexit XWBOPEN bietet keine Unterstützung für die Verwendung von **EXEC CICS**-Befehlen.

Die Beispielprogramme DFH\$WBPI und DFH\$WBEX mit dem zugehörigen Copybook DFH\$WBGA zeigen, wie Proxy-Server-Informationen bzw. Sicherheitsrichtlinien in einem globalen Arbeitsbereich konfiguriert werden. Wenn beispielsweise sämtliche Anforderungen Ihres CICS-Systems einen einzigen Proxy-Server verwenden sollen, können Sie den Namen des entsprechenden Proxy-Servers als Initialisierungsparameter angeben. Wenn Sie mehrere Proxy-Server verwenden oder eine Sicherheitsrichtlinie auf verschiedene Hostnamen anwenden wollen, können Sie eine Tabelle laden oder erstellen, die Hostnamen mit entsprechenden Proxy-Servern abgleicht oder als gesperrt markiert; diese Tabelle kann dann während der Verarbeitung des Befehls **EXEC CICS WEB OPEN** als Referenztabelle verwendet werden. Die Beispielprogramme können während der PLTPI-Verarbeitung (Program List Table Post Initialization) ausgeführt werden oder an einem beliebigen anderen Punkt, bevor der Befehl **EXEC CICS WEB OPEN** verwendet werden soll.

## Exit XWBOPEN

### Zeitpunkt des Aufrufs

Während der Verarbeitung eines Befehls vom Typ **EXEC CICS WEB OPEN** oder **EXEC CICS INVOKE SERVICE**.

### Exitspezifische Parameter

#### UEPHOST (Eingabe von CICS)

Adresse eines Felds mit dem Hostnamen, der IPv4- oder der IPv6-Adresse, der bzw. die in der Option HOST des Befehls WEB OPEN angegeben ist.

**Anmerkung:** Der Hostname wird beim Speichern in diesem Feld in Kleinbuchstaben konvertiert. Beim Abgleich des Hostnamens muss diese Konvertierung von Ihrem Benutzerexitprogramm berücksichtigt werden.

#### UEPHOSTL (Eingabe von CICS)

Adresse eines Felds mit der Halbwortlänge des Hostnamens.

#### UEPPROXY (Ausgabe vom Benutzerexit)

Adresse eines Felds mit der Adresse, die auf den Namen des Proxy-Servers verweist. Der Name des Proxy-Servers muss im URL-Format

vorliegen. Bei Eingabe in das Benutzerexitprogramm wird der Parameter auf die Adresse eines Felds gesetzt, das die Adresse eines 2046-Byte-Bereichs enthält. Sie können den Namen des Proxy-Servers in diesen Bereich stellen und die Adresse in UEPPROXY unverändert lassen. Alternativ können Sie den Namen des Proxy-Servers in Ihren eigenen Bereich stellen und die Adresse in UEPPROXY durch die Adresse eines Felds ersetzen, das die Adresse Ihres eigenen Bereichs enthält.

**UEPPROXYL (Ausgabe vom Benutzerexit)**

Adresse eines Felds mit der Halbwortlänge des Namens des Proxy-Servers.

**UEPHOSTT (Eingabe von CICS)**

Adresse eines 1-Byte-Codes, der den Hosttyp angibt, der im Parameter UEPHOST enthalten ist.

**Anmerkung:** Die Binärzahl 01 steht für einen Hostnamen, die Binärzahl 02 für eine IPv4-Adresse und die Binärzahl 03 für eine IPv6-Adresse.

**Rückgabecodes**

**UERCNORM**

Für diese HTTP-Anforderung ist kein Proxy-Server erforderlich, und der Hostname ist nicht gesperrt.

**UERCPROX**

Für diese HTTP-Anforderung ist ein Proxy-Server erforderlich. UEPPROXY wurde auf den Namen des erforderlichen Proxy-Servers gesetzt, und UEPPROXYL wurde auf die Länge des Namens des Proxy-Servers gesetzt.

**UERCBARR**

Der Hostname des Servers ist gesperrt.

**UERCERR**

Bei der Verarbeitung des Exits ist ein Fehler aufgetreten.

**XPI-Aufrufe**

Es können alle XPI-Aufrufe verwendet werden.

**API- und SPI-Befehle**

Es können keine **EXEC CICS**-Befehle verwendet werden.

## **Sendexit XWBSNDO für HTTP-Client**

Mithilfe von XWBSNDO können Sie eine Sicherheitsrichtlinie für HTTP-Anforderungen angeben, die durch CICS als HTTP-Client ausgegeben werden. XWBSNDO wird während der Verarbeitung eines Befehls vom Typ **EXEC CICS WEB SEND** oder **EXEC CICS WEB CONVERSE** aufgerufen. Der Hostname und die Pfadinformationen werden an den Exit übergeben, und es kann eine Sicherheitsrichtlinie auf eine oder beide dieser Komponenten angewendet werden.

CICS wendet keine Sicherheitsrichtlinien für HTTP-Anforderungen an, die von CICS als HTTP-Client ausgegeben werden. Es obliegt Ihnen, diese Funktion zu konfigurieren, wenn sie von Ihrem System oder Ihrer Organisation benötigt wird.

Sie können den Exit XWBOPEN im Befehl **WEB OPEN** verwenden, um den Zugriff auf einen Host insgesamt zu blockieren. Mit dem Exit XWBSNDO können Sie dasselbe erreichen oder den Zugriff auf bestimmte Pfade in einem Host zu blockieren. Beim Blockieren des Zugriffs auf einen Host insgesamt lässt sich mit dem Exit XW-

BOPEN Zeit sparen, da das Anwendungsprogramm die Verbindung ohnehin nicht öffnen kann und somit keine Zeit darauf verschwendet, die zu sendende Anforderung überhaupt zu erstellen. Der Hostname wird dem Exit XWBSNDO zur Verfügung gestellt, sodass Sie zwischen identischen Pfaden unterscheiden können, die von verschiedenen Hosts verwendet werden.

Wenn für die HTTP-Anforderung Chunked Transfer Coding verwendet wird, wird XWBSNDO nur für den ersten WEB SEND-Befehl für die in Blöcke aufgeteilte Nachricht aufgerufen.

Der Benutzerexit XWBSNDO bietet keine Unterstützung für die Verwendung von **EXEC CICS**-Befehlen.

Der Host wird als Parameter UEPHOST an das Benutzerexitprogramm übergeben, und der Pfad wird als Parameter UEPPATH übergeben. Der Rückgabecode UERC-NORM gibt an, dass der Pfad zulässig ist, und der Rückgabecode UERCBARR gibt an, dass der Pfad nicht zulässig ist. Wenn der Pfad nicht zulässig ist, wird eine Antwort vom Typ NOTAUTH an den Befehl WEB SEND oder WEB CONVERSE zurückgegeben, und der Anwendungsprogrammierer bearbeitet diese Antwort, indem er die Verbindung mit einem Befehl vom Typ WEB CLOSE schließt.

## Exit XWBSNDO

### Zeitpunkt des Aufrufs

Während der Verarbeitung eines Befehls vom Typ **EXEC CICS WEB SEND** oder **EXEC CICS WEB CONVERSE** für eine HTTP-Anforderung, die durch CICS als HTTP-Client ausgegeben wird. Eine Clientanforderung wird durch die Verwendung des Parameters SESSTOKEN im Befehl WEB SEND angegeben.

### Exitspezifische Parameter

#### UEPHOST

Adresse eines Felds mit dem Hostnamen, der IPv4- oder der IPv6-Adresse, der bzw. die in der Option HOST des Befehls WEB OPEN für die Verbindung angegeben ist.

**Anmerkung:** Der Hostname wird beim Speichern in diesem Feld in Kleinbuchstaben konvertiert. Beim Abgleich des Hostnamens muss diese Konvertierung von Ihrem Benutzerexitprogramm berücksichtigt werden.

#### UEPHOSTL

Adresse eines Felds mit der Halbwortlänge des Hostnamens.

#### UEPPATH

Adresse eines Felds mit dem Pfad, der in der Option PATH des Befehls WEB SEND angegeben ist. Der Pfad liegt in gemischter Groß-/Kleinschreibung vor, so wie er angegeben wurde.

#### UEPPATHL

Adresse eines Felds mit der Halbwortlänge des Pfads.

#### UEPHOSTT

Adresse eines 1-Byte-Codes, der den Hosttyp angibt, der im Parameter UEPHOST enthalten ist.

**Anmerkung:** Die Binärzahl 01 steht für einen Hostnamen, die Binärzahl 02 für eine IPv4-Adresse und die Binärzahl 03 für eine IPv6-Adresse.

#### **Rückgabecodes**

##### **UERCNORM**

Der Pfad ist zulässig.

##### **UERCBARR**

Der Pfad ist nicht zulässig.

#### **XPI-Aufrufe**

Es können alle XPI-Aufrufe verwendet werden.

#### **API- und SPI-Befehle**

Es können keine **EXEC CICS**-Befehle verwendet werden.

---

## **Exits XISCONA, XISLCLQ und XISQLCL für das Programm für systemübergreifende Kommunikation**

Mit den drei Exits im Programm für systemübergreifende Kommunikation können Sie die Länge der systemübergreifenden Warteschlangen steuern.

Es gibt mehrere Methoden, mit denen Sie die Länge von systemübergreifenden Warteschlangen steuern können. Eine Beschreibung der verfügbaren Methoden finden Sie unter *Intersystem session queue management*.

### **Exit XISCONA**

Der Zweck von XISCONA besteht darin, Leistungsprobleme vermeiden zu helfen, die auftreten können, wenn Funktionsverlagerungs- oder DPL-Anforderungen, die auf freie Sitzungen für eine Nicht-IPIC-Verbindung warten, in der Ausgaberegion in die Warteschlange gestellt werden. Der Exit ermöglicht es Ihnen, die Anzahl ausstehender ALLOCATE-Anforderungen zu steuern, indem Ihnen die Möglichkeit gegeben wird, alle Funktionsverlagerungs- bzw. DPL-Anforderungen zurückzuweisen, die andernfalls in die Warteschlange gestellt würden.

**Important:** Zur Steuerung der Länge von systemübergreifenden Warteschlangen ist der Exit XZIQUE im Arbeitsbereichsmodul von z/OS Communications Server besser geeignet als der Exit XISCONA. XZIQUE bietet mehr Funktionen und ist von allgemeinerem Nutzen als XISCONA. (XZIQUE wird für Anforderungen für Funktionsverlagerung, DPL, Transaktionsrouting und verteilte Transaktionsverarbeitung gesteuert, während XISCONA nur für Funktionsverlagerung und DPL gesteuert wird). Wenn Sie beide Exits aktivieren, könnte es sein, dass sowohl XZIQUE als auch XISCONA für Funktionsverlagerungs- und DPL-Anforderungen gesteuert werden, was nicht empfehlenswert ist.

Wenn Sie bereits über ein Exitprogramm für XISCONA verfügen, können Sie es unter Umständen entsprechend für die Verwendung am Exitpunkt XZIQUE ändern.

"Konfliktgewinner" ist die für LU6.2-Verbindungen verwendete Terminologie. Der Exit XISCONA gilt auch für MRO- und LU6.1-Verbindungen: Bei diesen Verbindungen werden die SEND-Sitzungen (die in den Sitzungsdefinitionen definiert sind) zuerst für ALLOCATE-Anforderungen verwendet; wenn alle SEND-Sitzungen belegt sind, wird die Warteschlangensteuerung gestartet.

Funktionsverlagerungs- und DPL-Anforderungen für eine Ressourcenverwaltungsregion (Resource-Owning Region, ROR) werden standardmäßig in die Warteschlange gestellt, wenn alle gebundenen Konfliktgewinnersitzungen ausgelastet sind, so dass keine Sitzungen unmittelbar verfügbar sind. Wenn die Ressourcenverwaltungsregion (ROR) nicht antwortet (wenn es sich beispielsweise um eine Dateiverwaltungsregion (FOR) handelt, kann es sein, dass sie auf die Archivierung eines Systemjournals wartet), kann die Warteschlange so lang werden, dass die Leistung der Ausgaberegion erheblich beeinträchtigt wird. Wenn die Ausgaberegion darüber hinaus noch eine Anwendungsverwaltungsregion (Application-Owning Region, AOR) ist, kann sich diese Leistungsbeeinträchtigung ausbreiten und auch die Terminalverwaltungsregion (Terminal-Owning Region, TOR) erfassen.

Verwenden Sie zur Steuerung der Warteschlange für Funktionsverlagerungs und DPL-Anforderungen den Exit XISCONA, um CICS über Folgendes zu informieren: Sobald eine Sitzung nicht sofort zugeordnet werden kann, ob die Anforderung in die Warteschlange gestellt werden soll oder ob 'SYSIDERR' an die Anwendung zurückgegeben werden soll. Der Exit funktioniert wie folgt:

1. Wenn das Exitprogramm für XISCONA **nicht** aktiv ist, stellt CICS die Anforderung in die Warteschlange, wenn dies erforderlich ist.
2. Wenn das Exitprogramm aktiv ist, wird es *nur dann aufgerufen, wenn alle gebundenen Konfliktgewinnersitzungen belegt sind*. Bei anderen Fehlern (beispielsweise 'Modusname nicht gefunden' oder 'Außer Betrieb') umgeht CICS den Exit und kehrt zur Anwendung zurück.
3. Wenn das Exitprogramm aufgerufen wird, muss es entscheiden, ob die Anforderung in die Warteschlange gestellt werden soll oder nicht; hierfür werden die Statistikdaten analysiert, die über die Parameterliste des Benutzerexits bereitgestellt werden. Das Exitprogramm könnte folgende Aktionen ausführen:
  - Festlegen, dass niemals eine Warteschlangensteuerung stattfinden soll. Dies ist die einfachste Möglichkeit, den Exit zu codieren, und vermeidet die Komplexität von Optimierungsmaßnahmen. Diese Methode sollte effektiv sein, wenn Sie genügend Konfliktgewinnersitzungen definieren, um Lastspitzen bei der Verarbeitung von Transaktionen für die Verbindung bewältigen zu können. Wenn Sie die Warteschlangensteuerung insgesamt unterdrücken, müssen Sie AUTOCONNECT(YES) in der SESSIONS-Definition angeben, da der Warteschlangensteuerungsmechanismus keine Sitzungen mehr für Sie bindet.

Dieser Ansatz birgt ein potenzielles Risiko, wenn Ihre Schätzung der erforderlichen Sitzungen auf durchschnittlichen Bedingungen basiert und die Transaktionslast nachher großen Schwankungen unterliegt. Wenn CICS keine Warteschlangensteuerung verwenden kann, um die Schwankungen zu bewältigen, kann es sein, dass Benutzer mit Transaktionsabbrüchen konfrontiert werden, selbst wenn in der Ressourcenverwaltungsregion keine signifikanten Probleme vorliegen.

- Überprüfung der Anzahl von Anforderungen, die sich gegenwärtig in der Warteschlange befinden. So könnte das Programm beispielsweise die Warteschlangensteuerung stoppen, wenn diese Anzahl mehr als 120% der maximalen Anzahl von Sitzungen beträgt. Mit diesem Ansatz könnten Sie das potenzielle Problem sporadisch auftretender Stopps in der Ressourcenverwaltungsregion angehen.

Sie könnten für die Verbindungen in Ihrem System eine Tabelle mit Schwellenwerten verwenden, die auf früheren Erfahrungen mit Problemen bei der Warteschlangensteuerung basieren. Alternativ dazu könnten Sie die EXEC CICS-Schnittstelle in einem separaten Programm verwenden, um den Status

der Verbindung abzufragen und die Informationen in einem Arbeitsbereich an das Exitprogramm für XISCONA zu übergeben.

- Überprüfung des Anforderungstyps und der Ressource, auf die zugegriffen wird. (Diese Informationen sind aus der Anforderungsparameterliste ersichtlich.) Das Programm könnte beispielsweise Dateileseanforderungen zurückweisen, aber Dateiaktualisierungen in die Warteschlange stellen.

**Anmerkung:** Da ein Ausfall des Exitprogramms die Systemverfügbarkeit beeinträchtigen könnte, wird empfohlen, die Logik Ihres Programms so einfach wie möglich zu gestalten und so die Wahrscheinlichkeit von Fehlern zu verringern.

Es gibt einige Probleme, die XISCONA nicht lösen kann. Wenn Sie beispielsweise sowohl eine große Anzahl von Sitzungen als auch einen großen Wert für MXT angegeben haben, kann dies unter Umständen bei CICS zu einer Speicherknappheit führen, *bevor* XISCONA aufgerufen wird, weil keine weiteren Sitzungen mehr verfügbar sind.

## Beispiel eines globalen Benutzerexitprogramms für XISCONA

DFHXIS ist ein Beispiexitprogramm, das eine Möglichkeit zum Begrenzen der Warteschlange für ALLOCATE-Anforderungen auf der Basis der an das Programm übergebenen Informationen zeigt. Weitere Informationen finden Sie unter Function-*shipping*/DPL queue control sample exit program: DFHXIS.

### Exit XISCONA

Der Exit XISCONA wird aufgerufen, wenn eine Funktionsverlagerungs- oder DPL-Anforderung im Begriff ist, in die Warteschlange gestellt zu werden, da alle gebundenen Konfliktgewinnersitzungen für die ferne Region bereits belegt sind.

#### Zeitpunkt des Aufrufs

Wenn eine Funktionsverlagerungs- oder DPL-Anforderung im Begriff ist, in die Warteschlange gestellt zu werden, da alle gebundenen Konfliktgewinnersitzungen für die ferne Region bereits belegt sind.

**Anmerkung:** Bei DPL-Anforderungen, die dynamisch weitergeleitet werden, wird das Programm für dynamisches Routing vor dem Exit XISCONA aufgerufen. Sind keine freien Sitzungen vorhanden, kann das Routing-Programm entscheiden, eine DPL-Anforderung nicht in die Warteschlange zu stellen. In einem solchen Fall wird XISCONA nicht aufgerufen. Informationen zum dynamischen Routing von DPL-Anforderungen finden Sie unter *Dynamically routing DPL requests*.

#### Exitspezifische Parameter

##### UEPISPCA

Adresse einer Parameterliste mit den nachstehenden Feldern. Die Parameterliste kann mithilfe des Pseudobereichs (DSECT) DFHXISDS zugeordnet werden.

##### UEPCONST

Adresse des Datensatzes mit der Verbindungsstatistik.

Verbindungsstatistikdatensätze sind vom Typ STICONSR (STID-Wert 52). Das Exitprogramm kann den Datensatz mithilfe des Pseudobereichs (DSECT) DFHA14DS zuordnen. (Siehe Anmerkungen.)

##### UEPMODST

Adresse des Statistikdatensatzes für Moduseinträge oder



null. Ein Statistikdatensatz für Moduseinträge wird *nur* unter folgenden Bedingungen erstellt:

- Der Verbindungstyp ist LU6.2 (siehe Feld UEPCONTY).
- Das Profil DFHCICSF (das immer für die Funktionsverlagerung verwendet wird) definiert einen bestimmten Modusnamen (MODENAME), der bei der Zuordnung von LU6.2-Sitzungen verwendet werden soll.

Statistikdatensätze für Moduseinträge sind vom Typ STICONMR (STID-Wert 76). Das Exitprogramm kann den Datensatz (sofern vorhanden) mithilfe des Pseudobereichs (DSECT) DFHA20DS zuordnen.

#### **UEPEIPPL**

Adresse der Anforderungsparameterliste.

#### **UEPCONTY**

1-Byte-Feld zur Angabe des Verbindungstyps. Mögliche Werte:

##### **UEPMRO (X'80')**

Anforderung für eine MRO-Verbindung (Mehrregionenbetrieb).

##### **UEPLU6 (X'40')**

Anforderung für eine LU6.1-Verbindung.

##### **UEPLUC (X'20')**

Anforderung für eine LU6.2-Verbindung.

#### **UEPNETNM**

Feld aus 8 Zeichen mit dem Netznamen (NETNAME) für die Verbindung, also mit der ID (Anwendungskennung) der fernen CICS-Region bzw. des fernen CICS-Systems.

#### **Anmerkung:**

1. Eine Beschreibung des allgemeinen Formats von Statistikdatensätzen finden Sie unter CICS statistics record format.
2. Eine Liste der Statistikdatensatztypen und zugehörigen Copybooks finden Sie unter CICS statistics data section.
3. Eine Beschreibung der Felder in Statistikdatensätzen für Verbindungen und Moduseinträge finden Sie unter CICS statistics in DSECTS and DFHSTUP report.

#### **Rückgabecodes**

##### **UERCAQUE**

Anforderung in die Warteschlange stellen. Dieser Code ist der Standardwert.

##### **UERCAPUR**

Anforderung nur dann in die Warteschlange stellen, wenn eine lokale Warteschlangensteuerung möglich ist.

#### **XPI-Aufrufe**

Es können alle verwendet werden.

## Wichtig

An diesem Exitpunkt ist der Rückgabecode UERCNORM nicht vorhanden, da der Exit nach einem Fehler aufgerufen wird. Es kann gewählt werden, ob die Standardaktion des Systems, also das Einreihen der Anforderung in die Warteschlange, ausgeführt werden soll oder nicht.

## Der Exit XISLCLQ

Der Exit XISLCLQ wird für EXEC CICS-Anforderungen vom Typ START NOCHECK verwendet, die für eine Nicht-IPIC-Verbindung terminiert sind.

Sofern er aktiviert ist, wird der Exit XISLCLQ in folgenden Situationen aufgerufen:

- Das ferne System ist außer Betrieb.
- Es kann keine Verbindung zum fernen System hergestellt werden.
- Es sind keine Sitzungen unmittelbar verfügbar, und das Exitprogramm für XISCONA hat angegeben, dass die Anforderung in der ausgehenden Region nicht in die Warteschlange gestellt werden soll.

Hierbei ist zu beachten, dass dieser Exit *nur dann* aufgerufen wird, wenn die zu übertragende Anforderungen vom Typ EXEC CICS START NOCHECK ist. Bei EXEC CICS-Anforderungen ohne die Option NOCHECK (die lediglich für Befehle vom Typ START verfügbar ist), wird die Bedingung SYSIDERR im Anwendungsprogramm ausgelöst.

Mithilfe des Exits können Sie angeben, ob die fehlgeschlagene Anforderung in eine lokale Warteschlange gestellt soll, um dann ausgeführt zu werden, nachdem die Verbindung wieder hergestellt worden ist.

Lokale Warteschlangen werden bei Durchführung eines Systemwiederanlaufs wiederhergestellt.

## Exit XISLCLQ

Der Exit XISLCLQ wird aufgerufen, nachdem eine Funktionsverlagerungsanforderung vom Typ EXEC CICS START NOCHECK fehlgeschlagen ist, weil das ferne System nicht in Betrieb ist, keine Verbindung zum fernen System hergestellt werden kann oder keine Sitzungen unmittelbar verfügbar sind, während das verwendete Exitprogramm für XISCONA angegeben hat, dass die Anforderung in der ausgehenden Region nicht in die Warteschlange gestellt werden soll.

### Zeitpunkt des Aufrufs

Nachdem eine Funktionsverlagerungsanforderung vom Typ EXEC CICS START NOCHECK fehlgeschlagen ist, weil das ferne System nicht in Betrieb ist, keine Verbindung zum fernen System hergestellt werden kann oder keine Sitzungen unmittelbar verfügbar sind, während das verwendete Exitprogramm für XISCONA angegeben hat, dass die Anforderung in der ausgehenden Region nicht in die Warteschlange gestellt werden soll.

### Exitspezifische Parameter

#### UEPISPP

Adresse einer Parameterliste, die Folgendes enthält:

#### UEPTCTSE

Adresse des Systemeintrags der relevanten Terminalsteuer-tabelle (TCT). Der TCT-Systemeintrag kann mithilfe des Pseudobereichs (DSECT) DFHTCTTE zugeordnet werden.

#### UEPXXTE

Die Adresse des lokalen Transaktionsnamens oder 0, wenn SYSID im Befehl angegeben wurde.

**Anmerkung:** Ihr Programm kann den XPI-Aufruf INQUIRE\_TRANDEF des Transaktionsmanagers verwenden, um Details zur lokalen Transaktion abzurufen (siehe The INQUIRE\_TRANDEF call).

#### UEPPLIST

Adresse der Parameterliste für den Befehl.

**Anmerkung:** Für diese Parameterliste wird kein Pseudobereich (DSECT) bereitgestellt. Für den Zugriff auf die benannten Felder müssen Sie Ihren eigenen DSECT codieren.

#### Rückgabecodes

##### UERC SYS

Systemaktion ausführen. Diese Aktion wird durch den Wert des Attributs LOCALQ in der lokalen TRANSACTION-Definition für die ferne Transaktion festgelegt:

##### LOCALQ(YES)

Die Anforderung wird lokal in die Warteschlange gestellt.

##### LOCALQ(NO)

SYSIDERR wird an das Anwendungsprogramm zurückgegeben.

##### UERC QUE

Anforderung lokal in die Warteschlange stellen (hierbei wird das Attribut LOCALQ(NO), sofern angegeben, außer Kraft gesetzt).

##### UERC IGN

Attribut LOCALQ(YES), sofern angegeben, außer Kraft setzen und mit SYSIDERR zurückkehren.

##### UERC PURG

Task während des XPI-Aufrufs bereinigt.

#### XPI-Aufrufe

Es können alle verwendet werden.

#### Wichtig

An diesem Exitpunkt ist der Rückgabecode UERCNORM nicht vorhanden, da der Exit nach einem Fehler aufgerufen wird. Es kann gewählt werden, ob die Standardaktion des Systems ausgeführt werden soll oder ob der Fehler auf andere Weise verarbeitet werden soll.

## Der Exit XISQLCL

Sie können den Exit XISQLCL für EXEC CICS-Befehle vom Typ START NOCHECK verwenden, die für eine IPIC-Verbindung terminiert sind.

Sofern er aktiviert ist, wird der Exit in folgenden Situationen aufgerufen:

- Die IPIC-Verbindung ist nicht angefordert worden.
- Es ist keine Sitzung verfügbar, und CICS stellt die Anforderung nicht für eine neue Sitzung in die Warteschlange.

Mithilfe von XISQLCL können Sie entscheiden, ob die Anforderung einer lokalen Warteschlange hinzugefügt werden soll oder ob eine Rückkehr mit einer Fehlerantwort erfolgen soll.

Lokale Warteschlangen werden bei Durchführung eines Systemwiederanlaufs wiederhergestellt.

## Beispielexitprogramm für XISQLCL

DFH£XISL ist ein Beispielexitprogramm für XISQLCL zur Warteschlangensteuerung für START NOCHECK-Anforderungen, die für eine IPIC-Verbindung terminiert sind. Weitere Informationen zu DFH£XISL finden Sie unter IPIC queue control sample exit program: DFH£XISL.

## Exit XISQLCL

Der Exit XISQLCL wird aufgerufen, nachdem eine Funktionsverlagerungsanforderung eines Befehls vom Typ **START NOCHECK** oder **START NOCHECK PROTECT** über IPIC fehlgeschlagen ist, weil das ferne System nicht in Betrieb ist, keine Verbindung zum fernen System hergestellt werden kann oder keine Sitzungen unmittelbar verfügbar sind, während das verwendete Exitprogramm für XISQUE angibt, dass die Anforderung in der ausgebenden Region nicht in die Warteschlange gestellt wird.

### Exitspezifische Parameter

Für diese Parameterliste wird der Pseudobereich (DSECT) DFH£ILDS bereitgestellt.

#### UEPISQPL

Adresse einer Parameterliste mit den folgenden Feldern:

#### UEPPLIST

Adresse der Parameterliste für den Befehl.

#### UEPQLEN

Halbwort-Binärfeld mit der Anzahl der Elemente, die sich derzeit in der Warteschlange befinden.

#### UEIPCNM

8-Byte-Name der IPIC-Verbindung (IPCONN).

#### UEPTRID

4-Byte-ID des lokalen Transaktionsnamens oder Leerzeichen, wenn SYSID im Befehl angegeben ist. Ihr Programm kann den XPI-Aufruf INQUIRE\_TRANDEF des Transaktionsmanagers verwenden, um Details zur lokalen Transaktion abzurufen. Siehe The INQUIRE\_TRANDEF call.

### Rückgabecodes

#### UERC SYS

Systemaktion ausführen. Diese Aktion wird durch den Wert des Attributs LOCALQ in der lokalen TRANSACTION-Definition für die ferne Transaktion festgelegt:

#### LOCALQ(YES)

Die Anforderung wird lokal in die Warteschlange gestellt.

#### LOCALQ(NO)

Eine Fehlermeldung vom Typ SYSIDERR wird an das Anwendungsprogramm zurückgegeben.

#### **UERCQUE**

Anforderung lokal in die Warteschlange stellen und dabei das Attribut LOCALQ(NO), sofern angegeben, außer Kraft setzen.

#### **UERCIGN**

Attribut LOCALQ(YES), sofern angegeben, außer Kraft setzen und mit einer Antwort vom Typ SYSIDERR zurückkehren.

#### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

#### **XPI-Aufrufe**

Es können alle verwendet werden.

### **Wichtig**

An diesem Exitpunkt ist der Rückgabecode UERCNORM nicht vorhanden, da der Exit nach einem Fehler aufgerufen wird. Sie müssen auswählen, ob die Standardaktion des Systems ausgeführt werden soll oder ob der Fehler auf andere Weise verarbeitet werden soll.

Als Beispielprogramm für den globalen Benutzerexit XISQLCL wird DFH\$XISL zur Verfügung gestellt.

---

## **Exits XICREQ, XICEXP und XICTENF für das Intervallsteuerprogramm**

Es ist möglich, in Exitprogrammen, die über das Intervallsteuerprogramm aufgerufen werden, eine Reihe von XPI-Aufrufen zu verwenden. Wenn jedoch irgendeiner dieser Exits für die Ablaufanalyse aufgerufen wird, können alle Aktionen, die die Ausführung des Intervallsteuerprogramms verzögern, nachteilige Auswirkungen auf andere Transaktionen haben, die darauf warten, dass Intervalle ablaufen.

Ob die Exits für die Ablaufanalyse aufgerufen wurden, lässt sich anhand des Felds TCAICTR mit dem Anforderungstyp ermitteln. Der exitspezifische Parameter UEPICRQ1 verweist auf eine Kopie dieses Felds.

Der Exit XICREQ wird durch interne Anforderungen aufgerufen, die vom CICS-Code ausgegeben werden, sowie durch Anforderungen, die von Anwendungen ausgegeben werden. DFHXRSP gibt alle 2 Sekunden eine WAIT-Anforderung für die Intervallsteuerung aus; dies bedeutet, dass auch alle Exitprogramme für die Intervallsteuerung alle 2 Sekunden aufgerufen werden.

### **Exit XICREQ**

Dieser Exit wird zu Beginn des Intervallsteuerprogramms aufgerufen, bevor eine Anforderungsanalyse durchgeführt wird.

#### **Zeitpunkt des Aufrufs**

Bei Beginn des Intervallsteuerprogramms vor der Anforderungsanalyse.

#### **Exitspezifische Parameter**

##### **UEPICQID**

Adresse eines 8-Byte-Felds, das gegebenenfalls den Parameter für die Anforderungs-ID der Anforderung enthält. Siehe Anmerkungen 1 und 2.

**UEPICTID**

Adresse eines 4-Byte-Felds mit der Terminal-ID (sofern vorhanden), die in einem Befehl vom Typ EXEC CICS START angegeben wurde. Siehe Anmerkungen 1 und 2.

**UEPICTI**

4-Byte-Adresse mit der Transaktions-ID, die in einem Befehl vom Typ EXEC CICS START angegeben wurde. Siehe Anmerkungen 1 und 2.

**UEPICRQ1**

Adresse eines 1-Byte-Felds mit einer Kopie von TCAICTR, dem ersten Anforderungscodefeld für Anforderungen an das Intervallsteuerprogramm.

**UEPICRQ2**

Adresse eines 1-Byte-Felds mit einer Kopie von TCAICTR2, dem zweiten Anforderungscodefeld für Anforderungen an das Intervallsteuerprogramm.

**UEPICRT**

Adresse eines 4-Byte-Felds mit der Ablaufzeit bzw. dem Ablaufintervall im gepackten Dezimalformat. Der Wert hat das Format 0HHMMSSF. Hierbei gilt: H=Stunden, M=Minuten, S=Sekunden, und F ist ein positives Vorzeichen.

**Anmerkung:**

1. Der Inhalt der Felder, die von UEPICQID und UEPICRT adressiert werden, ist unvorhersehbar, wenn die zugeordneten Datenelemente in der Anforderung nicht angegeben wurden. Sie müssen die Kopie von TCAICTR überprüfen, um zu ermitteln, ob die Felder aussagekräftige Werte enthalten.
2. Ihr Exitprogramm kann die Werte der Felder ändern, die von UEPICQID, UEPICRT, UEPICQID, UEPICRT, UEPICQID und UEPICRT adressiert werden. Änderungen an den Werten der Felder, die von UEPICRQ1 oder UEPICRQ2 adressiert werden, haben keine Auswirkungen.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen.

**UERC PURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Folgende Aufrufe dürfen nicht verwendet werden:

- ADD\_SUSPEND
- DELETE\_SUSPEND
- DEQUEUE
- ENQUEUE
- RESUME
- SUSPEND
- WAIT\_MVS

## Exit XICEXP

Dieser Exit wird aufgerufen, nachdem ein Zeitintervall der Intervallsteuerung abgelaufen ist.

### Zeitpunkt des Aufrufs

Nach Ablauf eines Zeitintervalls der Intervallsteuerung.

### Exitspezifische Parameter

#### UEPICE

Adresse des Intervallsteuerelements (ICE), das gerade abgelaufen ist. Das ICE kann mithilfe des Pseudobereichs (DSECT) DFHICEDS zugeordnet werden.

### Rückgabecodes

#### UERCNORM

Verarbeitung fortsetzen.

#### UERCPURG

Task während des XPI-Aufrufs bereinigt.

### XPI-Aufrufe

Folgende Aufrufe dürfen nicht verwendet werden:

- ADD\_SUSPEND
- DELETE\_SUSPEND
- DEQUEUE
- ENQUEUE
- RESUME
- SUSPEND
- WAIT\_MVS

## Exit XICTENF

Dieser Exit wird über das Intervallsteuerprogramm aufgerufen.

### Zeitpunkt des Aufrufs

Dieser Exit steht in Zusammenhang mit der Bedingung 'Terminal unbekannt'. Weitere Informationen finden Sie unter „Exits XALTENF und XICTENF für die Bedingung "Terminal unbekannt"“ auf Seite 266.

---

## Exits des Intervallsteuerprogramms der EXEC-Schnittstelle (XICEREQ, XICERES und XICEREQC)

Diese Exits werden aufgerufen, wenn Interaktionen mit Intervallsteuerprogrammen stattfinden.

### XICEREQ

XICEREQ wird beim Einstieg in das Intervallsteuerprogramm aufgerufen, bevor CICS eine Intervallsteueranforderung verarbeitet. Mithilfe von XICEREQ können Sie folgende Aktionen ausführen:

- Analysieren der Anforderung zwecks Ermittlung ihres Typs sowie der angegebenen Schlüsselwörter und zugehörigen Werte.
- Ändern der in der Anforderung angegebenen Werte vor Ausführung des Befehls.
- Festlegen von Rückgabecodes, um Folgendes anzugeben:

- CICS soll mit der Anforderung (geändert oder unverändert) fortgesetzt werden.
- CICS soll die Anforderung umgehen. (Hierbei ist Folgendes zu beachten: Wenn Sie diesen Rückgabecode festlegen, müssen Sie auch Rückgabecodes für den EXEC-Schnittstellenblock (EIB) festlegen, als hätten Sie die Anforderung selbst verarbeitet.)

**Anmerkung:** Der Exit XICEREQ wird durch interne Anforderungen aufgerufen, die vom CICS-Code ausgegeben werden, sowie durch Anforderungen, die von Anwendungen ausgegeben werden.

## XICERES

XICERES wird durch das Intervallsteuerprogramm aufgerufen, bevor CICS eine nicht-terminalbezogene Anforderung vom Typ **EXEC CICS START** verarbeitet, die dynamisch an diese Region weitergeleitet wurde.

Hierbei ist zu beachten, dass XICERES in den folgenden Situationen aufgerufen wird:

- Nach dem Exit XICEREQ und vor dem Exit XICEREQC (sofern diese Exits aktiviert sind). Dies bedeutet Folgendes:
  - Wenn ein Exitprogramm für XICEREQ die Anforderung umgeht, wird XICERES selbst bei Aktivierung nicht aufgerufen.
  - Wenn ein Exitprogramm für XICEREQ die Anforderung ändert, muss XICERES die geänderte Anforderung bearbeiten.
- In der *Zielregion*, also in der Region, an die die Startanforderung (START) weitergeleitet wurde.
- Nur dann, wenn die Routing-Region, also die Region, in der das Routing-Programm ausgeführt wird, die Bedingung RESUNAVAIL (Ressource nicht verfügbar) unterstützt. Um die Bedingung 'Ressource nicht verfügbar' unterstützen zu können, muss es sich bei der Region um ein unterstütztes Release von CICS TS handeln.
- Nur dann, wenn der Exit aktiviert ist. Es wird dringend empfohlen, diesen Exit nur in Anwendungsverwaltungsregionen (AORs) zu aktivieren, an die nicht-terminalbezogene Anforderungen vom Typ **EXEC CICS START** dynamisch weitergeleitet werden können.
- Durch interne Anforderungen, die vom CICS-Code ausgegeben werden, sowie durch Anforderungen, die von Anwendungen ausgegeben werden.

In den folgenden Situationen wird der Exit XICERES *nicht* aufgerufen:

- Bei statisch weitergeleiteten Anforderungen.
- Bei terminalbezogenen Anforderungen vom Typ **EXEC CICS START**. (Diese Anforderungen werden immer in der Terminalverwaltungsregion ausgeführt und können nicht weitergeleitet werden.)
- Bei dynamisch weitergeleiteten *Transaktionen*: Nur dynamisch weitergeleitete (nicht-terminalbezogene) START-Anforderungen führen zum Aufruf des Exits. Daher führt eine dynamisch weitergeleitete Transaktion, die von einem terminalbezogenen Befehl vom Typ **EXEC CICS START** eingeleitet wurde, nicht zum Aufruf des Exits.
- Wenn der Exit inaktiviert ist.
- Wenn ein Exitprogramm für XICEREQ die Anforderung umgeht.

Mithilfe von XICERES können Sie überprüfen, ob alle Ressourcen, die für die zu startende Transaktion erforderlich sind, in der Zielregion verfügbar sind. Wenn die Transaktion beispielsweise inaktiviert ist oder eine erforderliche Datei fehlt, kann Ihr Exitprogramm dem verteilten Routing-Programm die Gele-



genheit geben, die Anforderung an eine andere Region weiterzuleiten. Um dies zu erreichen, muss der Rückgabecode UERCRESU gesetzt werden. Daraufhin führt CICS die folgenden Aktionen aus:

1. Festlegen des Werts 'F' (Ressource nicht verfügbar) für das Feld DYRERROR im Kommunikationsbereich des verteilten Routing-Programms.
2. Erneutes Aufrufen des Routing-Programms in der Routing-Region bei fehlgeschlagener Routing-Auswahl.
3. Rückgabe einer Bedingung vom Typ RESUNAVAIL in dem vom Spiegel in der Zielregion ausgeführten **EXEC CICS START**-Befehl. (Diese Bedingung wird nicht an das Anwendungsprogramm zurückgegeben.)

CICS ignoriert sämtliche Änderungen, die das Exitprogramm an den Werten beliebiger Exitparameter vorgenommen hat. Das Exitprogramm kann einen Rückgabecode festlegen, aber keine Parameter ändern.

Informationen für den Benutzer zum dynamischen Routing von nicht-terminal-bezogenen **EXEC CICS START**-Anforderungen finden Sie unter Non-terminal-related START commands. Informationen zum Schreiben eines verteilten Routing-Programms zum Weiterleiten von nicht-terminalbezogenen **EXEC CICS START**-Anforderungen finden Sie unter Routing non-terminal-related START requests.

### **XICEREQC**

XICEREQC wird nach dem Abschluss einer Anforderung eines Intervallsteuerprogramms aufgerufen. Mithilfe von XICEREQC können Sie folgende Aktionen ausführen:

- Analysieren der Anforderung zwecks Ermittlung ihres Typs sowie der angegebenen Schlüsselwörter und zugehörigen Werte.
- Festlegen von Rückgabecodes für den EXEC-Schnittstellenblock (EIB). Bei der Überprüfung der Rückgabecodes ist der Wert für EIBRCODE maßgeblich.

**Anmerkung:** Der Exit XICEREQC wird durch interne Anforderungen aufgerufen, die vom CICS-Code ausgegeben werden, sowie durch Anforderungen, die von Anwendungen ausgegeben werden.

## **Parameter, die an die einzelnen Exits übergeben werden**

Es werden von CICS zehn Typen von Adressparametern an die Exits übergeben.

- Adresse der Parameterstruktur der Befehlsebene (UEPCLPS).
- Adresse eines Tokens (UEPICTOK), mit dem 4 Byte Daten von XICEREQ an XICEREQC übergeben werden.
- Adressen von Kopien von sechs Rückgabecodesegmenten, Ressourcen-, Datums- und Zeitinformationen des EXEC-Schnittstellenblocks (EXEC Interface Block, EIB).
- Adresse eines Tokens (UEPTSTOK), das während der gesamten Lebensdauer einer Task gültig ist.
- Adresse eines Rekursionszählers (UEPRECUR) für Exits.

## **Exit XICEREQ**

Mithilfe des Exits XICEREQ können Sie Folgendes tun: Die Anforderung analysieren, um ihren Typ sowie die angegebenen Schlüsselwörter mit den zugehörigen Werten zu ermitteln, die in der Anforderung angegebenen Werte vor Ausführung des Befehls ändern und Rückgabecodes festlegen, um anzugeben, ob CICS die Anforderung fortsetzen oder umgehen soll.

### **Zeitpunkt des Aufrufs**

Beim Einstieg in das Intervallsteuerprogramm, bevor CICS eine API-Anforderung zur Intervallsteuerung verarbeitet.

### **Exitspezifische Parameter**

#### **UEPCLPS**

Adresse der Parameterstruktur der Befehlsebene. Siehe „Exitspezifischer Parameter UEPCLPS“ auf Seite 166.

#### **UEPICTOK**

Adresse eines 4-Byte-Tokens, das an XICEREQC übergeben werden soll. Dies ermöglicht es Ihnen beispielsweise, einen Arbeitsbereich an den Exit XICEREQC zu übergeben.

#### **UEPRCODE**

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

#### **UEPRES P**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

#### **UEPRES P2**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

#### **UEPTSTOK**

Adresse eines 4-Byte-Tokens, das während des gesamten Lebenszyklus einer Task gültig ist. Siehe Using the task token UEPTSTOK.

#### **UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

#### **UEPRSRCE**

Adresse einer 8 Zeichen langen Kopie des EIB-Ressourcenwerts, EIBRSRCE.

#### **UEPDATE**

Adresse einer Vollwortkopie des EIB-Datumswerts, EIBDATE.

#### **UEPTIME**

Adresse einer Vollwortkopie des EIB-Zeitwerts, EIBTIME.

### **Rückgabecodes**

#### **UERCNORM**

Verarbeitung fortsetzen.

#### **UERC BYP**

Das Intervallsteuerprogramm der EXEC-Schnittstelle sollte diese Anforderung ignorieren.

#### **UERC PURG**

Task während des XPI-Aufrufs bereinigt.

### **XPI-Aufrufe**

Es können alle verwendet werden. An diesem Benutzerexit können Sie auch EXEC CICS-API-Befehle verwenden.

Wenngleich der Exit die Verwendung der XPI-Aufrufe GETMAIN und FREEMAIN zulässt, wird empfohlen, stattdessen die Befehle EXEC CICS GETMAIN und EXEC CICS FREEMAIN zu verwenden.

#### **API- und SPI-Befehle**

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

**EXEC CICS SHUTDOWN**  
**EXEC CICS XCTL**

**Anmerkung:** Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor, um keine Schleife zu verursachen. So sind Sie beispielsweise dafür verantwortlich, das Auftreten einer Schleife zu vermeiden, wenn eine Intervallsteueranforderung über den Exit XICEREQ ausgegeben wird. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.

## **Exit XICERES**

Der Exit XICERES wird durch das Intervallsteuerprogramm aufgerufen. Mithilfe von XICERES können Sie überprüfen, ob alle Ressourcen, die für die zu startende Transaktion erforderlich sind, in der Zielregion verfügbar sind.

#### **Zeitpunkt des Aufrufs**

Vor der Verarbeitung einer nicht terminalbezogene Anforderung vom Typ **EXEC CICS START**, die dynamisch an diese Region weitergeleitet wurde, wobei die Routing-Region die Bedingung RESUNAVAIL (Ressource nicht verfügbar) unterstützt.

#### **Exitspezifische Parameter**

**Anmerkung:** CICS ignoriert sämtliche Änderungen, die das Exitprogramm an den Werten beliebiger Exitparameter vorgenommen hat. Das Exitprogramm kann einen Rückgabecode festlegen, aber keine Parameter ändern.

##### **UEPCLPS**

Adresse der Parameterstruktur der Befehlsebene. Siehe „Exitspezifischer Parameter UEPCLPS“ auf Seite 166.

##### **UEPICTOK**

Adresse eines 4-Byte-Tokens, das an XICEREQC übergeben werden soll.

##### **UEPRCODE**

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

##### **UEPRES P**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

##### **UEPRES P2**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

##### **UEPTSTOK**

Adresse eines 4-Byte-Tokens, das während des gesamten Lebenszyklus einer Task gültig ist. Siehe Using the task token UEPTSTOK.

##### **UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Da der Exit XICE-

RES in derselben Transaktion niemals rekursiv aufgerufen werden kann, ist der Wert dieses Feldes immer 0.

#### **UEPRSRCE**

Adresse einer 8 Zeichen langen Kopie des EIB-Ressourcenwerts, EIBRSRCE.

#### **UEPDATE**

Adresse einer Vollwortkopie des EIB-Datumswerts, EIBDATE.

#### **UEPTIME**

Adresse einer Vollwortkopie des EIB-Zeitwerts, EIBTIME.

### **Rückgabecodes**

#### **UERCNORM**

Verarbeitung fortsetzen.

#### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

#### **UERCRESU**

Eine erforderliche Ressource ist nicht verfügbar. Wenn Sie diesen Wert festlegen, wird CICS die weitergeleitete Anforderung zurückweisen und den Wert 'F' (Ressource nicht verfügbar) im Feld DYRError im Kommunikationsbereich des Routing-Programms zurückgeben.

### **XPI-Aufrufe**

Es können alle verwendet werden. An diesem Benutzerexit können Sie auch EXEC CICS-API-Befehle verwenden.

Wenngleich der Exit die Verwendung der XPI-Aufrufe GETMAIN und FREEMAIN zulässt, wird empfohlen, stattdessen die Befehle EXEC CICS GETMAIN und EXEC CICS FREEMAIN zu verwenden.

### **API- und SPI-Befehle**

Es können alle API- und SPI-Befehle mit Ausnahme von **EXEC CICS SHUTDOWN** und **EXEC CICS XCTL** verwendet werden.

## **Exit XICEREQC**

Mit dem Exit XICEREQC können Sie die Anforderung analysieren, um ihren Typs sowie die angegebenen Schlüsselwörter und zugehörigen Werte zu ermitteln. Sie können auch Rückgabecodes für den EXEC-Schnittstellenblock (EIB) festlegen.

Bei der Überprüfung der Rückgabecodes ist der Wert für EIBRCODE maßgeblich. Weitere Informationen finden Sie unter „EXEC-Schnittstellenblock (EIB)“ auf Seite 176.

### **Zeitpunkt des Aufrufs**

Nach Abschluss einer API-Anforderung für die Intervallsteuerung und vor der Rückkehr vom Intervallsteuerprogramm der EXEC-Schnittstelle.

### **Exitspezifische Parameter**

#### **UEPCLPS**

Adresse der Parameterstruktur der Befehlsebene. Siehe „Exitspezifischer Parameter UEPCLPS“ auf Seite 166.

**UEPICTOK**

Adresse eines 4-Byte-Tokens, das von XICEREQ übergeben wird. Dadurch kann XICEREQ beispielsweise einen Arbeitsbereich an XICEREQC übergeben.

**UEPRCODE**

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

**UEPRES**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

**UEPRES2**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

**UEPTSTOK**

Adresse eines 4-Byte-Tokens, das während des gesamten Lebenszyklus einer Task gültig ist. Siehe Using the task token UEPTSTOK.

**UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

**UEPRSRCE**

Adresse einer 8 Zeichen langen Kopie des EIB-Ressourcenwerts, EIBRSRCE.

**UEPDATE**

Adresse einer Vollwortkopie des EIB-Datumswerts, EIBDATE.

**UEPTIME**

Adresse einer Vollwortkopie des EIB-Zeitwerts, EIBTIME.

**UEP\_IC\_REMOTE\_SYSTEM**

Wenn die Anforderung an eine ferne Region gesendet werden soll, ist dies die Adresse eines Bereichs mit dem 4-Byte-Namen der fernen Region. (Die ferne Region kann beispielsweise durch die Option SYSID des Befehls START, durch das Workload-Management oder durch die Option REMOTESYSTEM der Transaktionsdefinition angegeben worden sein.)

Wenn die Anforderung in der lokalen Region ausgeführt werden soll, ist dieser Parameter die Adresse eines 4-Byte-Bereichs mit Leerzeichen.

**UEP\_IC\_REMOTE\_NAME**

Wenn die Transaktion in einem fernen System ausgeführt werden soll, ist dies die Adresse eines Bereichs mit dem Namen der Transaktion, wie er im fernen System bekannt ist.

Das ferne System kann eine andere CICS-Region oder ein IMS-System sein. Wenn in UEP\_IC\_REMOTE\_SYSTEM eine CICS-Region benannt ist, dann ist der Name 1 bis 4 Zeichen lang. Wenn in UEP\_IC\_REMOTE\_SYSTEM ein IMS-System benannt ist, dann ist der Name 1 bis 8 Zeichen lang. IMS verwendet Namen mit einer Länge von 8 Zeichen: Wenn UEP\_IC\_REMOTE\_NAME weniger als 8 Zeichen enthält, setzt IMS den Namen in ein verwendbares Format um.

## **Rückgabecodes**

### **UERCNORM**

Verarbeitung fortsetzen.

### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

## **XPI-Aufrufe**

Es können alle verwendet werden.

Wenngleich der Exit die Verwendung der XPI-Aufrufe GETMAIN und FREEMAIN zulässt, wird empfohlen, stattdessen die Befehle EXEC CICS GETMAIN und EXEC CICS FREEMAIN zu verwenden.

## **API- und SPI-Befehle**

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

**EXEC CICS SHUTDOWN**

**EXEC CICS XCTL**

**Anmerkung:** Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor. So müssen Sie beispielsweise den Eintritt in eine Schleife vermeiden, wenn Sie eine Intervallsteueranforderung über den Exit XICEREQC ausgeben. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.

## **Parameterstruktur der Befehlsebene**

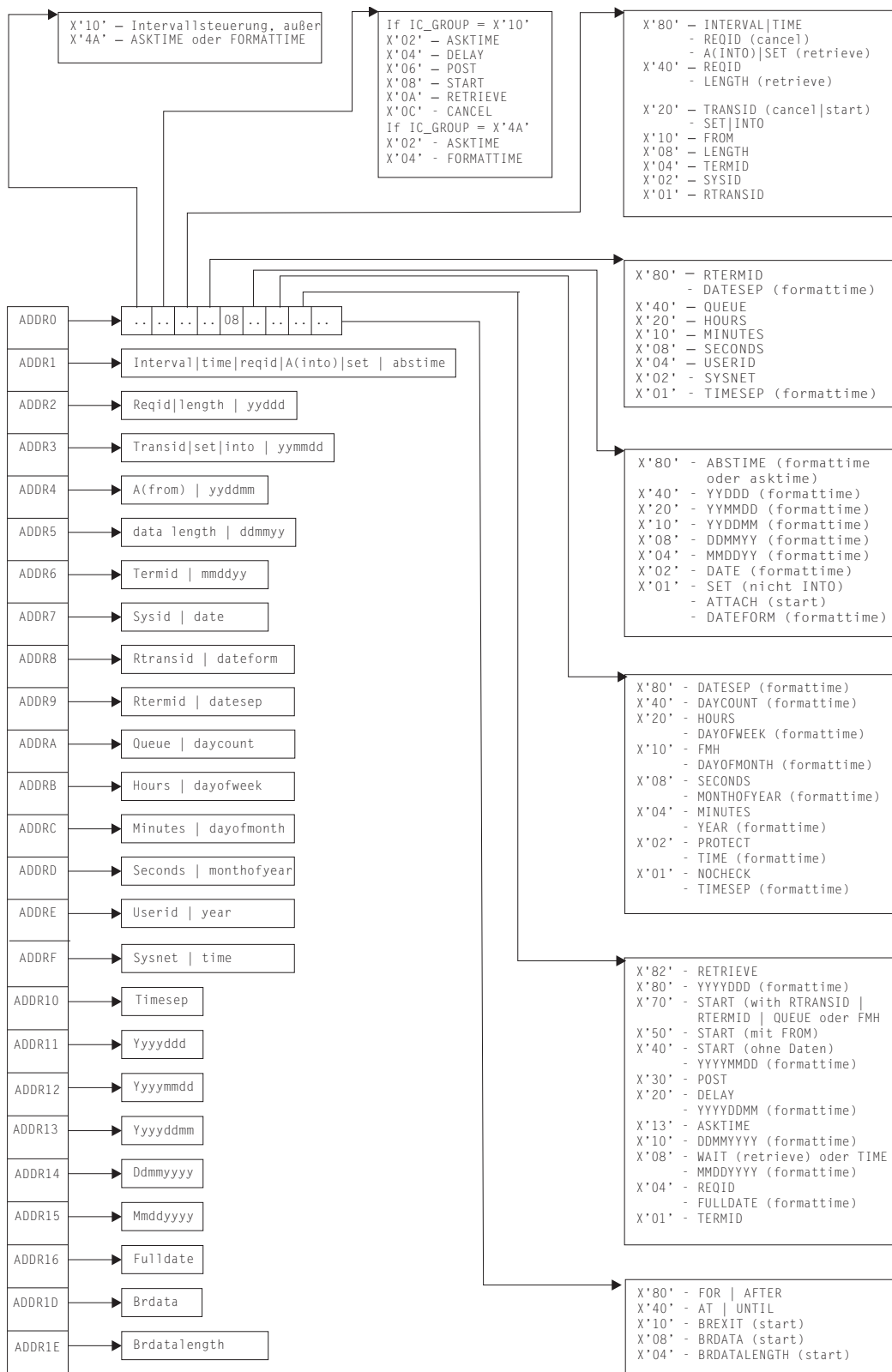


Abbildung 7. Parameterstruktur der Befehlsebene für die Intervallsteuerung

Die Parameterstruktur der Befehlsebene besteht aus einer Reihe von Adressen. Die erste Adresse verweist auf den EXEC-Schnittstellendescriptor (EID), der aus einem 9-Byte-Bereich besteht, der den Typ der Anforderung beschreibt und jedes mit der

Anforderung angegebene Schlüsselwort identifiziert. Die übrigen Adressen verweisen auf Teile von Daten, die der Anforderung zugeordnet sind. So verweist beispielsweise die zweite Adresse auf das Intervall für Startanforderungen (START).

Anhand des EXEC-Schnittstellendeskriptors können Sie den Typ der Anforderung sowie die angegebenen Schlüsselwörter ermitteln. Anhand der anderen Parameter in der Liste können Sie die Werte der Schlüsselwörter ermitteln. Sie können auch Werte von Schlüsselwörtern ändern, die in der Anforderung angegeben sind. So könnten Sie beispielsweise die in der Anforderung angegebene System-ID (SYSID) ändern.

## Indikator für Ende der Parameterliste

Das höchstwertige Bit wird für die letzte Adresse in der Parameterliste gesetzt, um anzuzeigen, dass es sich um die letzte Adresse in der Liste handelt. Bei der Rückkehr vom Benutzerexitprogramm sucht CICS in der Parameterliste nach dem höchstwertigen Bit, um den letzten Parameter ausfindig zu machen. Daher gilt: Wenn Sie die Länge der Parameterliste ändern, müssen Sie auch das höchstwertige Bit neu setzen, um so die aktuelle letzte Adresse zu markieren.

Beispiel: Wenn in der Parameterliste nur die ersten vier Adressen angegeben werden (IC\_ADDR0, die Adresse des EID, bis IC\_ADDR3, die Adresse des Namens der in einer START-Anforderung angegebenen Transaktion), dann wird das höchstwertige Bit in IC\_ADDR3 gesetzt. Wenn Sie dann die Parameterliste erweitern, indem Sie beispielsweise die Adresse einer SYSID in IC\_ADDR7 festlegen, müssen Sie das höchstwertige Bit in IC\_ADDR3 zurücksetzen und stattdessen in IC\_ADDR7 neu setzen.

Es wird die maximale Größe der Parameterliste an den Exit übergeben, sodass das Exitprogramm alle bisher noch nicht angegebenen Parameter hinzufügen kann, ohne zuvor mehr Speicher abrufen zu müssen.

Die ursprüngliche Parameterliste wird nach Beendigung von XICEREQC in der Version wiederhergestellt, die vor dem Aufruf von XICEREQ vorlag. Daraus folgt, dass die Ausführungsdiagnosefunktion (Execution Diagnostic Facility, EDF) vor **und** nach der Ausführung den ursprünglichen Befehl anzeigt: **EDF zeigt keine Änderungen an, die durch den Exit vorgenommen wurden.**

## Exitspezifischer Parameter UEPCLPS

Der exitspezifische Parameter UEPCLPS ist sowohl in Exit XICEREQ als auch in Exit XICEREQC enthalten. Es handelt sich um die Adresse der Parameterstruktur der Befehlsebene.

Die Parameterstruktur der Befehlsebene enthält 26 Adressen: IC\_ADDR0 bis IC\_ADDR1F. Die Struktur wird im Pseudobereich (DSECT)IC\_ADDR\_LIST definiert, den Sie in Ihr Exitprogramm kopieren sollten, indem Sie die Anweisung COPY DFHICUED einschließen.

Die Parameterliste der Befehlsebene ist wie folgt aufgebaut:

### IC\_ADDR0

Adresse eines 9-Byte-Bereichs, der als EID (EXEC Interface Descriptor) bezeichnet wird und wie folgt aufgebaut ist:

- IC\_GROUP
- IC\_FUNCT
- IC\_BITS1



- IC\_BITS2
- IC\_BITS3
- IC\_EIDOPT5
- IC\_EIDOPT6
- IC\_EIDOPT7
- IC\_EIDOPT8

#### IC\_GROUP

**X'10'** Dies ist eine Intervallsteueranforderung.

**X'4A'** Dies ist ein Befehl vom Typ ASKTIME oder FORMATTIME.

#### IC\_FUNCT

Ein Byte, das den Typ der Anforderung definiert.

Wenn IC\_GROUP = X'10':

**X'02'** ASKTIME

**X'04'** DELAY

**X'06'** POST

**X'08'** START

**X'0A'** RETRIEVE

**X'0C'** CANCEL

Wenn IC\_GROUP = X'4A':

**X'02'** ASKTIME

**X'04'** FORMATTIME

#### IC\_BITS1

Existenzbits zum Definieren der Argumente, die angegeben wurden. Um das Argument abzurufen, das einem Schlüsselwort zugeordnet ist, müssen Sie die entsprechende Adresse aus der Parameterstruktur der Befehlsebene verwenden. Bevor Sie diese Adresse verwenden, müssen Sie das zugehörige Existenzbit überprüfen. Wenn das Existenzbit inaktiviert ist, wurde das Argument in der Anforderung nicht angegeben, und die Adresse sollte nicht verwendet werden.

**X'80'** Wird gesetzt, wenn die Anforderung Argumente vom Typ INTERVAL oder TIME enthält oder wenn REQID in einer Anforderung vom Typ CANCEL angegeben wird, oder wenn SET oder INTO in einer Anforderung vom Typ RETRIEVE angegeben wird. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDR1** aussagekräftig.

**X'40'** Wird gesetzt, wenn REQID in einer Anforderung außer CANCEL angegeben wird oder wenn LENGTH in einer Anforderung vom Typ RETRIEVE angegeben wird. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDR2** aussagekräftig.

**X'20'** Wird gesetzt, wenn TRANSID in der Anforderung angegeben wird oder wenn SET oder INTO in einer Anforderung außer RETRIEVE angegeben wird. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDR3** aussagekräftig.

**X'10'** Wird gesetzt, wenn die Anforderung FROM angibt. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDR4** aussagekräftig.

- X'08' Wird gesetzt, wenn LENGTH in einer Anforderung außer RETRIEVE angegeben wird. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDR5** aussagekräftig.
- X'04' Wird gesetzt, wenn die Anforderung TERMID angibt. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDR6** aussagekräftig.
- X'02' Wird gesetzt, wenn die Anforderung SYSID angibt. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDR7** aussagekräftig.
- X'01' Wird gesetzt, wenn die Anforderung RTRANSID angibt. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDR8** aussagekräftig.

#### **IC\_BITS2**

Weitere Existenzbits für Argumente.

- X'80' Wird gesetzt, wenn die Anforderung RTERMID angibt oder wenn DATESEP in einer Anforderung vom Typ FORMATTIME angegeben wird. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDR9** aussagekräftig.
- X'40' Wird gesetzt, wenn die Anforderung QUEUE angibt. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDRA** aussagekräftig.
- X'20' Wird gesetzt, wenn die Anforderung HOURS angibt. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDRB** aussagekräftig.
- X'10' Wird gesetzt, wenn die Anforderung MINUTES angibt. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDRC** aussagekräftig.
- X'08' Wird gesetzt, wenn die Anforderung SECONDS angibt. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDRD** aussagekräftig.
- X'04' Wird gesetzt, wenn die Anforderung USERID angibt. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDRE** aussagekräftig.
- X'02' Wird gesetzt, wenn die Anforderung SYSNET angibt. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDRF** aussagekräftig.
- X'01' Wird gesetzt, wenn TIMESEP in einer Anforderung vom Typ FORMATTIME angegeben wird. Ist dieses Existenzbit gesetzt, dann ist **IC\_ADDR10** aussagekräftig.

#### **IC\_BITS3**

Ein Byte, das nicht von der Intervallsteuerung verwendet wird.

#### **IC\_EIDOPT5**

Gibt an, ob bestimmte Schlüsselwörter in der Anforderung angegeben wurden.

- X'80' ABSTIME wurde in einem Befehl vom Typ FORMATTIME oder ASKTIME angegeben.
- X'40' YYDDD wurde in einem Befehl vom Typ FORMATTIME angegeben.
- X'20' YYMMDD wurde in einem Befehl vom Typ FORMATTIME angegeben.
- X'10' YYDDMM wurde in einem Befehl vom Typ FORMATTIME angegeben.
- X'08' DDMMYY wurde in einem Befehl vom Typ FORMATTIME angegeben.

- X'04' MMDDYY wurde in einem Befehl vom Typ FORMATTIME angegeben.
- X'02' DATE wurde in einem Befehl vom Typ FORMATTIME angegeben.
- X'01' In einem Befehl vom Typ RETRIEVE wurde SET (und nicht INTO) angegeben. In einem Befehl vom Typ START wurde AT-TACH angegeben. In einem Befehl vom Typ FORMATTIME wurde DATEFORM angegeben. Sie können dieses Feld in Ihrem Benutzerexit nicht ändern.

#### IC\_EIDOPT6

Existenzbits, die angeben, ob bestimmte Schlüsselwörter in der Anforderung angegeben wurden.

- X'80' DATESEP wurde in einem Befehl vom Typ FORMATTIME angegeben.
- X'40' DAYCOUNT wurde in einem Befehl vom Typ FORMATTIME angegeben.
- X'20' DAYOFWEEK wurde in einem Befehl vom Typ FORMATTIME angegeben, oder es wurde HOURS angegeben.
- X'10' DAYOFMONTH wurde in einem Befehl vom Typ FORMATTIME angegeben, oder es wurde FMH angegeben.
- X'08' MONTHOFYEAR wurde in einem Befehl vom Typ FORMATTIME angegeben, oder es wurde SECONDS angegeben.
- X'04' YEAR wurde in einem Befehl vom Typ FORMATTIME angegeben, oder es wurde MINUTES angegeben.
- X'02' TIME wurde in einem Befehl vom Typ FORMATTIME angegeben, oder es wurde PROTECT angegeben.
- X'01' TIMESEP wurde in einem Befehl vom Typ FORMATTIME angegeben, oder es wurde NOCHECK angegeben.

#### IC\_EIDOPT7

Gibt an, ob bestimmte Funktionen oder Schlüsselwörter in der Anforderung angegeben wurden.

- X'F0' CANCEL wurde angegeben.
- X'82' RETRIEVE wurde angegeben.
- X'80' YYYYDD wurde in einem Befehl vom Typ FORMATTIME angegeben.
- X'40' YYYYMMDD wurde in einem Befehl vom Typ FORMATTIME angegeben, oder es wurde START angegeben.
- X'30' POST wurde angegeben.
- X'20' YYYYDDMM wurde in einem Befehl vom Typ FORMATTIME angegeben, oder es wurde DELAY, RTRANSID, RTERMID oder QUEUE angegeben und/oder FMH.
- X'13' ASKTIME wurde angegeben.
- X'10' DDMMYYYY wurde in einem Befehl vom Typ FORMATTIME angegeben, oder es wurde FROM, RTRANSID oder RTERMID angegeben und/oder QUEUE.

- X'08'** MMDDYYYY wurde in einem Befehl vom Typ FORMATTIME angegeben, oder es wurde TIME oder WAIT angegeben.
- X'04'** FULLDATE wurde in einem Befehl vom Typ FORMATTIME angegeben, oder es wurde REQID angegeben.
- X'01'** TERMID wurde angegeben.

#### **IC\_EIDOPT8**

Gibt an, ob bestimmte Schlüsselwörter in der Anforderung angegeben wurden.

- X'80'** FOR oder AFTER wurde angegeben.
- X'40'** AT oder UNTIL wurde angegeben.
- X'10'** BREXIT wurde angegeben.
- X'08'** BRDATA wurde angegeben.
- X'04'** BRDATALENGTH wurde angegeben.
- X'02'** CHANNEL wurde in einem Befehl vom Typ START angegeben.

#### **IC\_ADDR1**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines 8-Byte-Bereichs, der den Wert des Schlüsselworts INTERVAL enthält (oder des Schlüsselworts TIME, wenn **IC\_EIDOPT7** angibt, dass TIME angegeben ist).
- Adresse eines 8-Byte-Bereichs, der den Wert von REQID enthält (wenn die Anforderung CANCEL ist).
- Adresse eines 8-Byte-Bereichs, der den Wert des Schlüsselworts ABSTIME enthält.
- Adresse der Daten, die für INTO zurückgegeben wurden (wenn die Anforderung RETRIEVE ist und wenn **IC\_EIDOPT5** anzeigt, dass dies nicht SET ist).
- Adresse einer 4-Byte-Adresse, die für SET zurückgegeben wurde (wenn die Anforderung RETRIEVE ist und **IC\_EIDOPT5** anzeigt, dass dies SET ist).

#### **IC\_ADDR2**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines 8-Byte-Bereichs, der den Wert von REQID enthält (wenn die Anforderung DELAY, POST oder START ist).
- Adresse eines Halbwords, das den Wert von LENGTH enthält (wenn die Anforderung RETRIEVE ist).

**Warnung:** Bei Anforderungen mit Angabe von INTO darf der Wert für LENGTH nicht in einen Wert geändert werden, der größer ist als der von der Anwendung angegebene Wert. Wird ein größerer Wert angegeben, führt dies zu einer Speicherüberlagerung in der Anwendung.

- Adresse eines Bereichs, der den Wert von YYDD enthält.

#### **IC\_ADDR3**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines Bereichs, der den Wert von TRANSID enthält (wenn die Anforderung CANCEL oder START ist).
- Adresse einer 4-Byte-Adresse, die für SET zurückgegeben wurde (wenn die Anforderung START oder POST ist und **IC\_EIDOPT5** anzeigt, dass es sich um SET handelt).

- Adresse eines Bereichs, der den Wert von YYMMDD enthält.

#### **IC\_ADDR4**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines Bereichs, der die Daten aus FROM enthält.
- Adresse eines Bereichs, der den Wert von YYDDMM enthält.

#### **IC\_ADDR5**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines Bereichs, der den Halbwortwert von LENGTH enthält.

**Warnung:** Bei Anforderungen mit Angabe von INTO darf der Wert für LENGTH nicht in einen Wert geändert werden, der größer ist als der von der Anwendung angegebene Wert. Wird ein größerer Wert angegeben, führt dies zu einer Speicherüberlagerung in der Anwendung.

- Adresse eines Bereichs, der den Wert von DDMMYY enthält.

#### **IC\_ADDR6**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines Bereichs, der den Wert von TERMID enthält.
- Adresse eines Bereichs, der den Wert von MMDDYY enthält.

#### **IC\_ADDR7**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines Bereichs, der den Wert von SYSID enthält.
- Adresse eines Bereichs, der den Wert von DATE enthält.

#### **IC\_ADDR8**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines Bereichs, der den Wert von RTRANSID enthält.
- Adresse eines Bereichs, der den Wert von DATEFORM enthält.

#### **IC\_ADDR9**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines Bereichs, der den Wert von RTERMID enthält.
- Adresse eines Bereichs, der den Wert von DATESEP enthält.

#### **IC\_ADDRA**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines Bereichs, der den Wert von QUEUE enthält.
- Adresse eines Vollworts, das den Wert von DAYCOUNT enthält.

#### **IC\_ADDRB**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines Bereichs, der den Wert von HOURS enthält.
- Adresse eines Vollworts, das den Wert von DAYOFWEEK enthält.

#### **IC\_ADDRC**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines Bereichs, der den Wert von MINUTES enthält.
- Adresse eines Vollworts, das den Wert von DAYOFMONTH enthält.

#### **IC\_ADDRD**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines Bereichs, der den Wert von SECONDS enthält.
- Adresse eines Vollworts, das den Wert von MONTHOFYEAR enthält.

**IC\_ADDRE**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines Bereichs, der den Wert von USERID enthält.
- Adresse eines Vollworts, das den Wert von YEAR enthält.

**IC\_ADDRF**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse eines 8-Byte-Bereichs, der den Wert von SYSNET enthält.
- Adresse eines Bereichs, der den Wert von TIME enthält.

**IC\_ADDR10**

Adresse eines 1-Byte-Bereichs, der den Wert von TIMESEP enthält.

**IC\_ADDR11**

Adresse eines Bereichs, der den Wert von YYYYDDD enthält.

**IC\_ADDR12**

Adresse eines Bereichs, der den Wert von YYYYMMDD enthält.

**IC\_ADDR13**

Adresse eines Bereichs, der den Wert von YYYYDDMM enthält.

**IC\_ADDR14**

Adresse eines Bereichs, der den Wert von DDMMYYYY enthält.

**IC\_ADDR15**

Adresse eines Bereichs, der den Wert von MMDDYYYY enthält.

**IC\_ADDR16**

Adresse eines Bereichs, der den Wert von FULLDATE enthält.

**IC\_ADDR1D**

Adresse eines Bereichs, der den Wert von BRDATA enthält.

**IC\_ADDR1E**

Adresse eines Vollworts, das den Wert von BRDATALENGTH enthält.

**IC\_ADDR1F**

Adresse eines 16-Byte-Bereichs, der den Wert von CHANNEL enthält.

**Ändern von Feldern in der Parameterstruktur der Befehlsebene**

Von den Feldern, die an die Intervallsteuerung übergeben werden, werden einige als Eingabefelder für die Anforderung, andere als Ausgabefelder und wieder andere sowohl als Eingabe- als auch als Ausgabefelder verwendet. Die Methode, die Ihr Benutzerexitprogramm zum Ändern eines Felds verwendet, hängt von der Verwendung des betreffenden Felds ab.

Die folgenden Felder sind immer Eingabefelder:

- INTERVAL
- TIME
- REQID
- FROM
- TERMID
- SYSID
- HOURS
- MINUTES
- SECONDS
- USERID

- CHANNEL

Die folgenden Felder sind immer Ausgabefelder:

- DATE
- DATEFORM
- DAYCOUNT
- DAYOFMONTH
- DAYOFWEEK
- DDMMYY
- DDMMYYYY
- FULLDATE
- INTO
- MMDDYY
- MMDDYYYY
- MONTHOFYEAR
- SET
- TIME
- YEAR
- YYDDD
- YYDDMM
- YYMMDD
- YYYYDDD
- YYYYDDMM
- YYYYMMDD

Die folgenden Felder sind bei START-Anforderungen Eingabefelder und bei RETRIEVE-Anforderungen Ausgabefelder:

- RTRANSID
- RTERMID
- QUEUE

LENGTH ist in Anforderungen vom Typ START ein Eingabefeld, ein Ausgabefeld in Anforderungen vom Typ RETRIEVE mit Angabe von SET und ein Ein-/Ausgabefeld in Anforderungen vom Typ RETRIEVE mit Angabe von INTO.

ABSTIME ist ein Eingabefeld in Anforderungen vom Typ FORMATTIME und ein Ausgabefeld in Anforderungen vom Typ ASKTIME. DATESEP und TIMESEP können Eingabefelder in FORMATTIME-Anforderungen sein.

### Ändern von Eingabefeldern

Die richtige Methode zum Ändern eines Eingabefelds besteht darin, eine neue Kopie desselben zu erstellen und die Adresse in der Parameterliste der Befehlsebene so zu ändern, dass sie auf die neuen Daten verweist.

**Anmerkung:** Sie dürfen ein Eingabefeld niemals ändern, indem Sie die Daten ändern, auf die durch die Parameterliste der Befehlsebene verwiesen wird. Dies würde zu einer Beschädigung von Speicher führen, der zum Anwendungsprogramm gehört; dies wiederum würde zu einem Fehler führen, wenn das Programm versucht, das Feld wiederzuverwenden.

## Ändern von Ausgabefeldern

Sie können ein Ausgabefeld ändern, indem Sie die Daten ändern, auf die die Parameterliste der Befehlsebene verweist.

Das unter „Ändern von Eingabefeldern“ auf Seite 173 beschriebene Verfahren ist für das Ändern von Ausgabefeldern nicht geeignet, da die Ergebnisse an den neuen Bereich und nicht an den Bereich der Anwendung zurückgegeben würden und daher für die Anwendung nicht sichtbar wären.

Bei einem Ausgabefeld können Sie die vorhandenen Anwendungsdaten ändern, da die Anwendung ohnehin erwartet, dass das Feld geändert wird.

## Ändern des EXEC-Schnittstellendesktors (EID)

Es ist nicht möglich, den EXEC-Schnittstellendesktors (EXEC Interface Descriptor, EID) zu ändern, um bedeutende Änderungen an Anforderungen vorzunehmen. So kann beispielsweise eine Anforderung vom Typ DELAY nicht in eine Anforderung vom Typ START geändert werden. Es können jedoch geringfügige Änderungen an Anforderungen vorgenommen werden. So ist es beispielsweise möglich, das Existenzbit für die System-ID (SYSID) zu aktivieren, sodass die betreffende Anforderung in eine Anforderung geändert werden kann, die an ein fernes System übertragen wird.

Einige Intervallsteuerungsbefehle verwenden 2 Bit im EID, um ein einzelnes Schlüsselwort anzugeben. Der Befehl EXEC CICS START beispielsweise verwendet 2 Bit zur Angabe der Terminal-ID (TERMINID). Das erste Bit (in IC\_BITS1) gibt an, dass ADDR6 in der Befehlsparameterliste gültig ist (ADDR6 verweist auf TERMINID), und das zweite Bit (in IC\_EIDOPT7) ist das Existenzbit für das Schlüsselwort, um anzuzeigen, dass das Schlüsselwort TERMINID im Befehl angegeben wurde.

In solchen Fällen müssen Sie unbedingt beide Biteinstellungen (konsistent) ändern, wenn Sie die betreffenden Befehle über ein Benutzerexitprogramm ändern wollen, da die Ergebnisse andernfalls unvorhersehbar sind.

In der nachstehenden Liste werden diejenigen Bits im EID aufgeführt, die geändert werden **können**. Versuche, andere Bestandteile des EID zu ändern, werden ignoriert.

### IC\_BITS1

- X'80' Existenzbit für REQID (wenn die Anforderung CANCEL ist).
- X'40' Existenzbit für LENGTH (wenn die Anforderung RETRIEVE ist) oder REQID.
- X'10' Existenzbit für FROM.
- X'08' Existenzbit für LENGTH.
- X'04' Existenzbit für TERMINID.
- X'02' Existenzbit für SYSID.
- X'01' Existenzbit für RTRANSID.

### IC\_BITS2

- X'80' Existenzbit für RTERMINID.
- X'40' Existenzbit für QUEUE.
- X'20' Existenzbit für HOURS.



X'10' Existenzbit für MINUTES.

X'08' Existenzbit für SECONDS.

#### **IC\_EIDOPT6**

X'20' Sekundäres Existenzbit für HOURS.

X'10' Existenzbit für FMH.

X'08' Sekundäres Existenzbit für SECONDS.

X'04' Sekundäres Existenzbit für MINUTES.

X'02' Existenzbit für PROTECT.

X'01' Existenzbit für NOCHECK.

#### **IC\_EIDOPT7**

Bits in IC\_EIDOPT7 sollten nur innerhalb derselben Funktionsgruppe geändert werden. Beispiel: Für eine Anforderung vom Typ START sollten nur diejenigen Existenzbits gesetzt werden, die für START-Anforderung als gültig definiert worden sind.

##### **ASKTIME-Anforderungen**

X'13' ASKTIME-Anforderung. Dieser Wert ist für alle ASKTIME-Anforderungen festgelegt und sollte nicht geändert werden.

##### **DELAY-Anforderungen**

X'20' DELAY-Anforderung.

X'08' TIME wurde angegeben.

X'04' REQID wurde angegeben.

##### **POST-Anforderungen**

X'30' POST-Anforderung.

X'08' TIME wurde angegeben.

X'04' REQID wurde angegeben.

##### **START-Anforderungen**

X'40' START-Anforderung (ohne DATA).

X'50' START-Anforderung (mit DATA).

X'70' START mit mindestens einer der folgenden Angaben: RTRANSID, RTERMID, QUEUE oder FMH.

X'08' TIME wurde angegeben.

X'04' REQID wurde angegeben.

X'01' TERMID wurde angegeben.

##### **RETRIEVE-Anforderungen**

X'82' RETRIEVE-Anforderung.

##### **CANCEL-Anforderungen**

X'F0' CANCEL-Anforderung.

X'04' REQID wurde angegeben.

#### **IC\_EIDOPT8**

X'20' Wird nicht von CICS verwendet.

Der EID wird vor der Rückkehr zum Anwendungsprogramm auf seinen ursprünglichen Wert zurückgesetzt. Dies bedeutet, dass Änderungen, die am EID vorgenommen werden, nur für die Dauer der Intervallsteueranforderung beibehalten werden.

**Anmerkung:** Das Benutzerexitprogramm wird zwar daran gehindert, bedeutende Änderungen am EID vorzunehmen. Dennoch müssen Sie auch bei den geringfügigen Änderungen, die **zulässig sind**, sehr sorgfältig vorgehen.

### **Verwendung des Tokens für Intervallsteueranforderungen, UEPICTOK**

UEPICTOK stellt die Adresse eines 4-Byte-Bereichs bereit, mit dem Sie für dieselbe Intervallsteueranforderung Informationen zwischen den Benutzerexits XICEREQ und XICEREQC übergeben können.

So kann beispielsweise die Adresse eines Speicherteils, der vom Benutzerexit XICEREQ abgerufen wird und vom Exit XICEREQC freigegeben werden soll, im Feld UEPICTOK übergeben werden.

Das Token UEPICTOK kann nur für die Dauer einer einzelnen Intervallsteueranforderung verwendet werden, da sein Inhalt am Ende der Anforderung unter Umständen gelöscht werden kann. Wenn Sie Informationen zwischen aufeinanderfolgenden Aufrufen eines globalen Benutzerexits übergeben müssen, können Sie zu diesem Zweck das Task-Token UEPTSTOK verwenden. Weitere Informationen zu UEPTSTOK finden Sie unter Using the task token UEPTSTOK.

### **EXEC-Schnittstellenblock (EIB)**

Es werden Kopien von EIBRSRCE, EIBRCODE, EIBRESP und EIBRESP2 an den Exit übergeben, sodass Sie Ausführungs- und Ressourceninformationen in XICEREQ und XICEREQC ändern/festlegen oder Ausführungs- und Ressourceninformationen in XICEREQC untersuchen können.

Sie können die Kopien von EIBRSRCE, EIBRCODE, EIBRESP und EIBRESP2, die Sie in der Parameterliste erhalten, aktualisieren. Die Intervallsteuerung kopiert Ihre Werte in den realen EIB, nachdem der Exit XICEREQC ausgeführt wurde oder wenn Sie den Rückgabecode für Umgehen in XICEREQ angeben.

Sie müssen gültige Antworten für die Intervallsteuerung festlegen. Sie müssen für die drei Codes EIBRCODE, EIBRESP und EIBRESP2 eine konsistente Gruppe von Werten festlegen, so wie sie auch von der CICS-Intervallsteuerung festgelegt würden, um eine gültige Ausführung zu beschreiben. **Die Konsistenz der Werte für EIBRCODE, EIBRESP und EIBRESP2 wird von CICS nicht überprüft.** Wenn jedoch EIBRCODE auf einen Wert ungleich null gesetzt wird und EIBRESP auf null gesetzt wird, überschreibt CICS diesen Wert für EIBRESP durch einen Wert, der ungleich null ist. Als Hilfestellung beim Festlegen von Werten für EIBRCODE, EIBRESP und EIBRESP2 können Sie die im Pseudobereich (DSECT) DFHICUED angegebenen Werte heranziehen, die von der Intervallsteuerung verwendet werden.

Wenn Sie die Rückgabecodes mithilfe des Exits XICEREQC untersuchen wollen, verwenden Sie den Wert für EIBRCODE.

## Beispiel für die Verwendung von XICERREQ und XICEREQC

In diesem Beispiel werden XICERREQ und XICEREQC verwendet, um START-Anforderungen an eine Reihe von CICS-Regionen weiterzuleiten, um einen einfachen Lastausgleichsmechanismus bereitzustellen. Das Beispiel veranschaulicht lediglich, das Leistungsspektrum der Exits; es ist nicht beabsichtigt aufzuzeigen, wie diese Funktionalität bestmöglich erreicht werden kann.

### Verarbeitungsschritte in XICERREQ:

1. Durchsuchen Sie den globalen Arbeitsbereich (Global Work Area, GWA) nach einer geeigneten CICS-Region (beispielsweise nach der Region, die momentan die wenigsten START-Anforderungen verarbeitet).
2. Nachdem Sie entschieden haben, an welches System die Anforderung weitergeleitet werden soll, erhöhen Sie den Nutzungszähler für dieses System.
3. Fordern Sie einen 4-Byte-Bereich an, in dem die System-ID (SYSID) für diese Anforderung gespeichert werden soll. Dieser Bereich kann aus dem globalen Arbeitsbereich (GWA) zugeordnet werden, um die Ausgabe einer GETMAIN-Anforderung zu vermeiden. Wird der Bereich durch Ausgabe einer GETMAIN-Anforderung abgerufen, setzen Sie UEPIC TOK auf die Adresse des angeforderten Speichers.
4. Setzen Sie IC\_ADDR7 auf die Adresse des 4-Byte-Bereichs, damit XICEREQC diesen Bereich ebenfalls verwenden kann.
5. Wenn das Festlegen von IC\_ADDR7 dazu führt, dass es sich nun um die letzte Adresse handelt, setzen Sie das höchstwertige Bit in dieser Adresse und setzen Sie das höchstwertige Bit in der Adresse, die zuvor die letzte Adresse war, entsprechend zurück.
6. Aktivieren Sie das Existenzbit X'02' in IC\_BITS1, um anzuzeigen, dass eine System-ID (SYSID) angegeben worden ist.
7. Kehren Sie zu CICS zurück.

### Verarbeitungsschritte in XICEREQC:

1. Durchsuchen Sie den globalen Arbeitsbereich (Global Work Area, GWA) nach dem Eintrag für die CICS-Region, die im Parameter SYSID angegeben ist.
2. Verringern Sie den Nutzungszähler für dieses System.
3. Wurde in XICERREQ eine GETMAIN-Anforderung ausgegeben, um einen Bereich zum Speichern der SYSID abzurufen, geben Sie einen Aufruf vom Typ FREEMAIN für die in UEPIC TOK enthaltene Adresse aus.
4. Kehren Sie zu CICS zurück.

## Beispiel und Beispielprogramme

CICS stellt zwei Programme zur Verwendung am Exit XICERREQ bereit:

- DFH\$XTSE: Dieses Programm wird nur als Softcopy-Liste (nicht als Quellcode-datei) bereitgestellt. Es handelt sich um ein Beispielprogramm, das veranschaulicht, wie Felder in der Parameterstruktur der Befehlsebene, die an alle EXEC-Schnittstellenexits übergeben wird, geändert werden.
- DFH\$ICCN: Hierbei handelt es sich um ein Beispielprogramm zur Verwendung in einer Umgebung für verteiltes Routing, in der Sie eine zuvor ausgegebene Intervallsteueranforderung abbrechen wollen, aber keine Möglichkeit haben zu ermitteln, an welche Region die Anforderung CANCEL übertragen werden soll. Beispiele für Situationen, die mithilfe von DFH\$ICCN bewältigt werden können, finden Sie unter Canceling interval control requests.

---

## Exits XLDLOAD und XLDELETE für die Ladedomäne

Für die Ladedomäne gibt es zwei globale Benutzerexits. XLDLOAD wird aufgerufen, wenn eine neue Instanz eines Programms in den Speicher geladen wird, und zwar bevor das Programm zur Verwendung bereitgestellt wird.

XLDELETE wird aufgerufen, nachdem eine Instanz eines Programms durch CICS freigegeben wurde und bevor das Programm aus dem Speicher freigegeben wird.

Bei LPA-residenten Programmen werden die Exits weiterhin aufgerufen, wenn ein Programm angefordert oder freigegeben wird, obwohl das betreffende Programm nicht physisch geladen oder freigegeben wird.

Diese beiden Exits dienen lediglich zu Informationszwecken. Sämtliche Änderungen, die das Exitprogramm an den Exitparametern vornimmt, werden von CICS ebenso ignoriert wie sämtliche Rückgabecodes, die das Programm festlegt.

### Exit XLDLOAD

Dieser Exit wird für eine Programminstanz aufgerufen, die in den Speicher überführt wird, bevor das Programm verfügbar wird.

#### Zeitpunkt des Aufrufs

Nach der Überführung einer Instanz eines Programms in den Speicher und vor der Bereitstellung des Programms zur Verwendung.

#### Exitspezifische Parameter

##### UEPPROGN

Adresse eines Felds aus 8 Zeichen mit dem Namen des Programms, das geladen wird.

##### UEPPROGL

Adresse eines 4-Byte-Felds mit der Länge (in Byte) des Programms, das geladen wird.

##### UEPLDPT

Adresse eines 4-Byte-Felds mit der Adresse, an der das Programm geladen worden ist.

##### UEPENTRY

Adresse eines 4-Byte-Felds mit der Adresse des Einstiegspunkts des Programms.

##### UEPTRANID

Null oder die Adresse eines 4-Byte-Felds mit der Transaktions-ID, die beim Aufruf des Exits gültig war.

##### UEPUSER

Null oder die Adresse eines 8-Byte-Felds mit der Benutzer-ID, die zum Zeitpunkt des Exitaufrufs die Steuerung hatte.

##### UEPTERM

Null oder die Adresse eines 4-Byte-Felds mit dem Terminalnamen, der der Transaktion zugeordnet ist, unter der der Exit aufgerufen wurde.

##### UEPPROG

Null oder die Adresse eines Felds aus 8 Zeichen mit dem Namen des Programms, das zum Zeitpunkt des Exitaufrufs die Steuerung hatte.

## UEPLDCTXT

Null oder die Adresse eines 140-Byte-Felds mit dem Anwendungskontext, wenn ein privates Programm, das zu einer Anwendung gehört, geladen wird. Das Feld enthält Folgendes:

1. Plattformname, mit Leerzeichen auf 64 Zeichen aufgefüllt.
2. Anwendungsname, mit Leerzeichen auf 64 Zeichen aufgefüllt.
3. Hauptversionsnummer für die Anwendung, angegeben als Vollwort-Binärwert.
4. Nebenversionsnummer für die Anwendung, angegeben als Vollwort-Binärwert.
5. Mikroversionsnummer für die Anwendung, angegeben als Vollwort-Binärwert.

CICS stellt einen Pseudobereich (DSECT) namens **DFHUEACD** für die Zuordnung dieser Informationen bereit. Weitere Informationen zu **DFHUEACD** finden Sie unter UEACD - User exit application context in Data Areas.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

### XPI-Aufrufe

Dürfen nicht verwendet werden.

### API- und SPI-Aufrufe

Dürfen nicht verwendet werden.

## Exit XLDELETE

Dieser Exit wird aufgerufen, wenn eine Programminstanz durch CICS freigegeben wird, und bevor das Programm aus dem Speicher freigegeben wird.

### Zeitpunkt des Aufrufs

Nach der Freigabe einer Instanz eines Programms durch CICS und vor der Freigabe des Programms aus dem Speicher.

### Exitspezifische Parameter

#### UEPPROGN

Adresse eines Felds aus 8 Zeichen mit dem Namen des Programms, das freigegeben wird.

#### UEPPROGL

Adresse eines 4-Byte-Felds mit der Länge (in Byte) des Programms, das freigegeben wird.

#### UEPLDPT

Adresse eines 4-Byte-Felds mit der Adresse, an der sich das Programm im Speicher befindet.

#### UEPENTRY

Adresse eines 4-Byte-Felds mit der Adresse des Einstiegspunkts des Programms.

#### UEPTRANID

Null oder die Adresse eines 4-Byte-Felds mit der Transaktions-ID, die beim Aufruf des Exits gültig war.

#### **UEPUSER**

Null oder die Adresse eines 8-Byte-Felds mit der Benutzer-ID, die zum Zeitpunkt des Exitaufrufs die Steuerung hatte.

#### **UEPTERM**

Null oder die Adresse eines 4-Byte-Felds mit dem Terminalnamen, der der Transaktion zugeordnet ist, unter der der Exit aufgerufen wurde.

#### **UEPPROG**

Null oder die Adresse eines Felds aus 8 Zeichen mit dem Namen des Programms, das zum Zeitpunkt des Exitaufrufs die Steuerung hatte.

#### **UEPLDCTXT**

Null oder die Adresse eines 140-Byte-Felds mit dem Anwendungskontext, wenn ein privates Programm, das zu einer Anwendung gehört, gelöscht wird. Das Feld enthält Folgendes:

1. Plattformname, mit Leerzeichen auf 64 Zeichen aufgefüllt.
2. Anwendungsname, mit Leerzeichen auf 64 Zeichen aufgefüllt.
3. Hauptversionsnummer für die Anwendung, angegeben als Vollwort-Binärwert.
4. Nebenversionsnummer für die Anwendung, angegeben als Vollwort-Binärwert.
5. Mikroversionsnummer für die Anwendung, angegeben als Vollwort-Binärwert.

CICS stellt einen Pseudobereich (DSECT) namens **DFHUEACD** für die Zuordnung dieser Informationen bereit. Weitere Informationen zu **DFHUEACD** finden Sie unter UEACD - User exit application context in Data Areas.

#### **Rückgabecodes**

##### **UERCNORM**

Verarbeitung fortsetzen.

#### **XPI-Aufrufe**

Dürfen nicht verwendet werden.

#### **API- und SPI-Aufrufe**

Dürfen nicht verwendet werden.

---

## **Exit XLGSTRM für die Protokollmanagerdomäne**

In der Protokollmanagerdomäne gibt es einen einzigen Exitpunkt: XLGSTRM. Mit Hilfe von XLGSTRM können Sie eine Anforderung an MVS ändern, um einen neuen Protokolldatenstrom zu erstellen. Sie können den Namen des Modellprotokolldatenstroms und andere Parameter ändern, bevor diese an die MVS-Systemprotokollfunktion übergeben werden.

Wenn die Anforderung einer Protokolldatenstromverbindung von CICS an die MVS-Systemprotokollfunktion fehlschlägt, weil der Protokolldatenstrom nicht für MVS definiert ist, gibt CICS eine Anforderung an die MVS-Systemprotokollfunktion aus, um den betreffenden Protokolldatenstrom anhand der Definition eines Modellprotokolldatenstroms dynamisch zu erstellen.

Der Name des Modellprotokolldatenstroms, den CICS an MVS übergibt, hängt wie nachstehend dargestellt davon ab, ob der Journalname auf das Systemprotokoll oder ein allgemeines CICS-Protokoll verweist:

### CICS-Systemprotokolle

*&systemname.letztes\_LSN-qualifikationsmerkmal.MODEL*

Hierbei gilt Folgendes: *&systemname* ist das MVS-Symbol, das in den Systemnamen des MVS-Image aufgelöst wird. *letztes\_LSN-qualifikationsmerkmal* ist das letzte Qualifikationsmerkmal des Namens des Protokolldatenstroms (Log Stream Name, LSN) gemäß Angabe in der JOURNALMODEL-Ressourcendefinition.

Wenn Sie keine JOURNALMODEL-Ressourcendefinition für DFHLOG und DFHSHUNT angeben oder wenn Sie die in der Gruppe DFHLGMOD bereitgestellten CICS-Definitionen verwenden, werden als Namen der Modellprotokolldatenströme standardmäßig '*&systemname.DFHLOG.MODEL*' und '*&systemname.DFHSHUNT.MODEL*' verwendet.

Beispiel: Wenn eine CICS-Region eine Anforderung zum Erstellen eines Protokolldatenstroms für das Protokoll ihres primären Systems ausgibt und CICS in einem MVS-Image mit der System-ID MV10 ausgeführt wird und die standardmäßige JOURNALMODEL-Definition verwendet, geht die MVS-Systemprotokollfunktion davon aus, dass ein Modellprotokolldatenstrom namens MV10.DFHLOG.MODEL vorhanden ist.

Wenn der Systemname des MVS-Image mit einem numerischen Zeichen beginnt und weniger als 8 Zeichen lang ist, stellt CICS diesem Namen das Präfix 'C' voran, sodass der Name des Modellprotokolldatenstroms das folgende Format erhält: *C&systemname.letztes\_LSN-qualifikationsmerkmal*. Dies liegt daran, dass die MVS-Systemprotokollfunktion Protokolldatenstromnamen zurückweist, die mit einem numerischen Zeichen beginnen. Wenn der Systemname des MVS-Image mit einem numerischen Wert beginnt, aber bereits 8 Zeichen lang ist (das Maximum), fügt CICS das Präfix 'C' nicht hinzu. Dies bedeutet, dass die MVS-Systemprotokollfunktion den Standardnamen des Modellprotokolldatenstroms zurückweist. Ihr globales Benutzerexitprogramm kann den Namen des Modellprotokolldatenstroms jedoch ändern.

### Allgemeine CICS-Protokolle

*LSN-qualifikationsmerkmal\_1.LSN-qualifikationsmerkmal\_2.MODEL*. Die Standardwerte für diese beiden Qualifikationsmerkmale sind die Benutzer-ID der CICS-Region und die Anwendungs-ID der CICS-Region. Es können jedoch auch benutzerdefinierte Werte verwendet werden, die in einer JOURNALMODEL-Ressourcendefinition angegeben sind.

Beispiel: Wenn die Benutzer-ID der CICS-Region 'CICSHT##' und die Anwendungs-ID 'CICSHTA1' ist, dann lautet der Standardmodellname 'CICSHT##.CICSHTA1.MODEL'.

Die folgenden Informationen werden an ein globales Benutzerexitprogramm für XLGSTRM übergeben:

- Der Name des Protokolldatenstroms, der definiert werden soll.
- Der Standardname des Modellprotokolldatenstroms.
- Ein Systemprotokollflag.
- Die Parameterliste IXGINVNT der MVS-Systemprotokollfunktion.

Das Exitprogramm kann den Namen des Modelldatenstroms ändern, indem das Feld aktualisiert wird, auf das vom exitspezifischen Parameter UEPMLSN verwiesen wird. Es folgt ein Beispiel, das zeigt, wie das Exitprogramm den Modelldatenstromnamen ändern kann:

```
L      R3,UEPMLSN          R3 = address of stream name
MVC    0(26,R3),=CL26'NEW.MODEL.NAME'
```

Durch die Aktualisierung des Felds, auf das durch den Parameter UEPIXG verwiesen wird, kann das Exitprogramm die Parameterliste des Makros IXGINVNT ändern, die von der MVS-Systemprotokollfunktion verwendet wird, um den Protokolldatenstrom zu definieren. Verwenden Sie im Makro IXGINVNT das Format MF=M; dies ermöglicht es dem Exit, die Protokolldatenstromattribute anzugeben, die verwendet werden sollen. Es folgt ein Beispiel, das zeigt, wie das Exitprogramm den Strukturnamen ändern kann:

```
L      R9,UEPIXG
IXGINVNT REQUEST=DEFINE,
        TYPE=LOGSTREAM,
        STRUCTNAME=NEW_STRUCTURE,
        MF=(M,(R9),NOCHECK)

...
NEW_STRUCTURE DC CL16'LOG_SYSTEST_009'
```

Es ist nicht notwendig, die Liste zu codieren und Formate des Makros IXGINVNT auszuführen oder die Makros IXGCON oder IXGANSAA in den Exit einzuschließen. Dafür sorgt der CICS-Code, der die Anforderung DEFINE ausgibt.

Informationen zum Service IXGINVNT finden Sie in der Veröffentlichung z/OS MVS Programming: Assembler Services Guide.

Ein globales Benutzerexitprogramm für XLGSTRM kann explizite Attribute für die Protokolldatenstromdefinition festlegen und darüber hinaus auch einen Rückgabecode festlegen, der bewirkt, dass die Protokolldatenstromdefinition umgangen wird.

**Anmerkung:** Wenn XLGSTRM die Verbindung der CICS-Systemprotokolle abfangen soll, müssen Sie das Exitprogramm in einem PLT-Programm der ersten Phase aktivieren.

## Beispiel eines globalen Benutzerexitprogramms für XLGSTRM

DFH\$LGLS ist ein Beispiel eines globalen Benutzerexitprogramms für den Exitpunkt XLGSTRM. Es zeigt, wie auf einige der an das Exitprogramm für XLGSTRM übergebenen Parameter zugegriffen wird und wie diese Parameter geändert werden können. Weitere Informationen finden Sie unter Log manager domain sample exit program: DFH\$LGLS.

## Exit XLGSTRM

Der Exit XLGSTRM wird aufgerufen, nachdem der CICS-Protokollmanager feststellt, dass ein Protokolldatenstrom nicht vorhanden ist, und bevor der CICS-Protokollmanager die MVS-Systemprotokollfunktion aufruft, um den Protokolldatenstrom dynamisch zu definieren.

### Exitspezifische Parameter

#### UEPTRANID

Adresse der 4-Byte-Transaktions-ID.



**UEPUSER**

Adresse der 8-Byte-Benutzer-ID, die der Transaktion zugeordnet ist, wenn es sich bei der aktuellen Task um eine Benutzertask handelt.

**UEPTERM**

Adresse der 4-Byte-Terminal-ID, die der Transaktion zugeordnet ist (sofern vorhanden).

**UEPPROG**

Adresse des 8-Byte-Anwendungsprogrammnamens für diese Transaktion (sofern vorhanden).

**UEPLSN**

Adresse eines aus 26 Zeichen bestehenden Felds mit dem Namen des zu definierenden Protokolldatenstroms.

Das Exitprogramm sollte den Namen des Protokolldatenstroms nicht ändern. Bei der Rückkehr vom Exit ignoriert CICS sämtliche Änderungen, die am Inhalt des durch UEPLSN adressierten Felds vorgenommen wurden. Zum Auswählen des Namens eines Protokolldatenstroms können die dafür bereitgestellten JOURNALMODEL-Definitionen verwendet werden.

**UEPMLSN**

Adresse eines aus 26 Zeichen bestehenden Felds mit dem Namen des Modellprotokolldatenstroms, der zur Bereitstellung der Attribute für den neuen Protokolldatenstrom verwendet werden soll. Dieses Feld kann geändert werden, damit das globale Benutzerexitprogramm anstelle des von CICS generierten Namens einen anderen Namen für den Modellprotokolldatenstrom angeben kann.

**UEPIXG**

Adresse der Parameterliste des Makros IXGINVNT, die von der MVS-Systemprotokollfunktion verwendet werden soll, um den Protokolldatenstrom zu definieren. Anhand des Formats MF=M im Makro IXGINVNT kann das globale Benutzerexitprogramm die Protokolldatenstromattribute angeben, die verwendet werden sollen.

Details zum Makro IXGINVNT finden Sie in der Veröffentlichung z/OS MVS Programming: Authorized Assembler Services Guide.

**UEPLGTYP**

Adresse eines 1-Byte-Felds, das angibt, ob der gerade erstellte Protokolldatenstrom für ein Systemprotokoll oder ein allgemeines Protokoll verwendet werden soll. Gültige Werte:

**UEPSYSLG**

Der Protokolldatenstrom wird für ein CICS-Systemprotokoll erstellt.

**UEPGENLG**

Der Protokolldatenstrom wird für ein allgemeines Protokoll (ein Protokoll für aktualisierende Wiederherstellung, ein Benutzerjournal oder ein automatisches Journal) erstellt.

**Rückgabecodes****UERCNORM**

CICS wird fortgesetzt und versucht, den Protokolldatenstrom zu definieren.

#### UERCBYP

CICS versucht nicht, den Protokolldatenstrom zu definieren. Es ist möglich, dass der Prozess, der den Protokolldatenstrom verwenden wollte (beispielsweise eine Operation zum Öffnen eines Datasets), daraufhin fehlschlägt.

#### XPI-Aufrufe

Es können alle verwendet werden.

#### API- und SPI-Befehle

Dürfen nicht verwendet werden.

## Beispiel für die Verwendung des Exits XLGSTRM

Der Exit XLGSTRM wird für die Auswahl alternativer Modellprotokolldatenströme verwendet.

Nehmen Sie an, dass 200 CICS-Regionen auf 20 MVS-Images ausgeführt werden. Um nicht jeden einzelnen Protokolldatenstrom, der von den einzelnen CICS-Regionen verwendet wird, explizit definieren zu müssen, entscheiden Sie sich für die Verwendung von Modelldefinitionen. Protokolldatenströme werden bei ihrer erstmaligen Verwendung für MVS dynamisch definiert. Hierbei wird ein Exitprogramm für XLGSTRM verwendet, um aus alternativen Modellprotokolldatenströmen eine entsprechende Auswahl zu treffen. So könnte es funktionieren:

1. Beim ersten Start einer CICS-Region gibt der Systeminitialisierungsparameter INITPARM Folgendes an:

```
INITPARM=(Exit_enabler_pgmname=nnn)
```

Hierbei gilt Folgendes:

- Exit\_enabler\_pgmname ist der Name des Programms, welches das Benutzerexitprogramm für XLGSTRM aktiviert.
  - nnn ist eine Zahl, die eine Gruppe von CICS-Regionen identifiziert, die dieselbe Gruppe von Protokolldatenstrommodellen gemeinsam nutzen.
2. Das Programm, welches das Benutzerexitprogramm für XLGSTRM aktiviert, gibt den Befehl EXEC CICS ASSIGN INITPARM aus, um den Wert nnn abzurufen, und stellt ihn in den globalen Arbeitsbereich des Exitprogramms.
  3. Wenn die Region versucht, eine Verbindung zu ihrem Systemprotokoll herzustellen, weil der Protokolldatenstrom nicht definiert ist, wird das Exitprogramm für XLGSTRM aufgerufen. Das Exitprogramm wählt daraufhin das Modell CICS.DFHLOG.MODELnnn aus.

---

## Exit XMEOUT für die Nachrichtendomäne

Sie können den Exit XMEOUT verwenden, um CICS- und CICSplex SM-Nachrichten, die die CICS-Nachrichtendomäne verwenden, zu unterdrücken oder umzuleiten.

Das Exitprogramm unterliegt den folgenden Einschränkungen:

- Es kann nur solche Nachrichten unterdrücken oder umleiten, die an die Systemkonsole oder an Warteschlangen mit transienten Daten gesendet werden. Nachrichten, die an Terminalbediener gesendet werden, können hingegen nicht unterdrückt oder umgeleitet werden. Daher wird XMEOUT nicht für Nachrichten aufgerufen, die an Terminalbediener gesendet werden.
- Es kann nur solche Nachrichten unterdrücken oder umleiten, die die Nachrichtendomäne verwenden. Anhand von CICS messages können Sie ermitteln, auf

welche CICS-Nachrichten dies zutrifft. Die Beschreibung der einzelnen Nachrichten, die XMEOUT aufrufen können, enthält eine Liste mit dem Titel „XMEOUT parameter/Message inserts“ (XMEOUT-Parameter/Nachrichteneinfügungen); wenn für eine Nachricht keine XMEOUT-Parameter aufgelistet sind, dann kann die betreffende Nachricht den Exit nicht aufrufen. So kann beispielsweise die Nachricht DFHDX8320 den Exit XMEOUT aufrufen, die Nachricht DFHDX0205 hingegen nicht.

Bei CICSplex SM wird XMEOUT nur für Nachrichten mit dem Ziel EYULOG aufgerufen, da dies die Nachrichten sind, die die Nachrichtendomäne verwenden. Anhand von CICSplex SM messages können Sie ermitteln, auf welche Nachrichten dies zutrifft.

**Anmerkung:** CICSplex SM-Nachrichten, die den Exit XMEOUT aufrufen, können nur über das Jobprotokoll oder die Konsole umgeleitet oder unterdrückt werden, nicht jedoch über EYULOG.

- Es kann keine Nachrichten der CICSplex SM-Webbenutzerschnittstelle umleiten oder unterdrücken.
- Es kann weder den Text einer Nachricht noch die Nachrichteneinfügungen ändern. Bei entsprechenden Versuchen werden die Änderungen von CICS ignoriert.
- Es kann keine Nachrichten unterdrücken oder umleiten, die in den frühen Phasen der CICS-Initialisierung ausgegeben werden (da das Exitprogramm in diesen Phasen nicht aktiviert werden kann).
- Es kann während eines CICS-Systemabschlusses keine Nachrichten an Warteschlangen mit transienten Daten (TDQs) umleiten, es sei denn, das ursprüngliche Nachrichtenziel umfasste bereits mindestens eine Warteschlange mit transienten Daten. Bei einem entsprechenden Versuch, wird die Nachricht an ihr ursprüngliches Ziel weitergeleitet, und die Nachricht DFHME0120I wird an die Konsole ausgegeben. Das Benutzerexitprogramm kann die Nachricht DFHME0120I nicht umleiten, aber unterdrücken.

Diese Einschränkung ist erforderlich, da die Nachrichtendomäne Nachrichten während des Systemabschlusses von CICS auch dann noch verarbeiten muss, nachdem die Funktion der Warteschlangen mit transienten Daten beendet worden ist.

Um zu ermitteln, ob der CICS-Systemabschluss bereits gestartet wurde, kann das Exitprogramm nach der ersten Instanz der Nachricht DFHME0120 suchen. Es kann die Umleitung von Nachrichten an Warteschlangen mit transienten Daten stoppen, nachdem DFHME0120 ausgegeben wurde.

**Anmerkung:** Wenn eine Nachricht an eine Warteschlange mit transienten Daten umgeleitet wird und die Anforderung für transiente Daten fehlschlägt, geht die Nachricht verloren. Für den MEME-Ausnahmebedingungstracepunkt wird die ID X'0328' geschrieben. Die Interpretationszeichenfolge dieses Traceeintrags erläutert, warum die Anforderung für transiente Daten fehlgeschlagen ist.

## Wichtig

Aufgrund der Gefahr einer Rekursion darf das Exitprogramm für XMEOUT nicht versuchen, die folgenden Nachrichten umzuleiten:

- Alle Nachrichten des Typs DFHTDxxxx, die vom Programm für transiente Daten generiert werden.
- Benutzerdomänennachrichten im Bereich DFHUS0002 bis DFHUS0006, zuzüglich Nachricht DFHUS0150.

- Transaktionsmanagernachrichten DFHXM0212, DFHXM0213, DFHXM0304 und DFHXM0308.
- Anwendungsnachrichten DFHAP0001, DFHAP0002, DFHAP0004, DFHAP0601, DFHAP0602 und DFHAP0603.
- Alle Benutzerdomänennachrichten (DFHUSxxxx) an eine partitionsinterne Warteschlange, die mit einem anderen TRIGLEV-Wert als null definiert ist, wenn die Nachrichten generiert werden, während die Benutzerdomäne eine Fehlerbehebungsverarbeitung ausführt.

Die Vorlage für Nachrichtendefinitionen enthält einen Indikator namens *noreroute*. Dieser Indikator wird aktiviert, wenn die Nachricht, die gerade ausgegeben wird, vom Exitprogramm für XMEOUT nicht an eine Warteschlange mit transienten Daten umgeleitet werden kann. Die Adresse des Indikators wird im exitspezifischen Parameter UEPNRTE an XMEOUT übergeben. Das Exitprogramm kann den Wert des Indikators überprüfen, bevor es entscheidet, ob eine bestimmte Nachricht umgeleitet werden soll.

**Anmerkung:** Wenn das Exitprogramm versucht, eine unzulässige Nachricht umzuleiten, verhindert die Nachrichtendomäne diese Umleitung und gibt die betreffende Nachricht stattdessen an die Konsole aus, zusammen mit der Nachricht DFHME0137.

Jede Nachricht, die von dieser Einschränkung betroffen ist, wird in CICS messages durch eine entsprechende Anmerkung gekennzeichnet.

Es ist möglich, APPLID (die Anwendungs-ID) als optionalen Parameter in einer Nachricht zu übergeben. Allerdings ist die APPLID, die in eine Nachricht eingefügt wird, möglicherweise nicht die APPLID des aktuellen CICS-Systems. Wenn beispielsweise eine CICSplex SM-MAS-Nachricht an einen CMAS (CICSplex SM-Adressraum) weitergeleitet wird, wird unter Umständen die APPLID des MAS-Systems übergeben, sodass die Nachricht die APPLID des MAS-Systems und nicht die des aktuellen Systems (des CMAS) enthält.

Ihr Exitprogramm kann Nachrichten unterdrücken oder umleiten, indem es die Werte in den Adressen ändert, auf die von den nachstehenden Feldern der Parameterliste verwiesen wird. Andere Gruppen von Werten können von dem Programm nicht geändert werden.

- UEPMROU (Weiterleitungscodes)
- UEPMNRC (Anzahl der Weiterleitungscodes)
- UEPMTDQ (Namen der Warteschlangen mit transienten Daten)
- UEPMNTD (Anzahl der Warteschlangen mit transienten Daten)

## Verwendung von XMEOUT zur Überwachung der Nachricht DFHAP1900

Die Nachricht DFHAP1900 wird ausgegeben, wenn durch bestimmte Befehle der Systemprogrammierschnittstelle eine Änderung an der CICS-Systemkonfiguration vorgenommen wird. Hierbei handelt es sich um folgende Befehle: **SET**, **PERFORM**, **ENABLE**, **DISABLE** und **RESYNC**. Die Befehle werden in die Warteschlange mit transienten Daten namens CADS geschrieben. Nachrichten vom Typ DFHAP1900 können eine Möglichkeit zur Überwachung dynamischer Konfigurationsänderungen bieten und auch bei der Problembestimmung hilfreich sein. Weitere Informationen finden Sie unter SPI commands that can be audited .

**Anmerkung:** Geben Sie im Exit keine SPI-Befehle aus, die zu einer Nachricht vom Typ DFHAP1900 führen, da in diesem Fall eine Rekursion im Exit auftreten kann.

## Exit XMEOUT

Dieser Exit wird aufgerufen, bevor eine Nachrichtendomäne eine CICS-Nachricht sendet.

### Zeitpunkt des Aufrufs

Bevor die Nachrichtendomäne eine CICS-Nachricht an das entsprechende Ziel sendet.

### Exitspezifische Parameter

#### UEPTRANID

Adresse der 4-Byte-Transaktions-ID.

#### UEPUSER

Adresse der 8-Byte-Benutzer-ID.

#### UEPTERM

Adresse der 4-Byte-ID des Terminals, unter dem die aktuelle Transaktion ausgeführt wird. Wenn die aktuelle Transaktion keinem Terminal zugeordnet ist, enthält das adressierte Feld hexadezimale Nullen.

#### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens oder Nullen, wenn keine aktuelle Anwendung vorhanden ist.

#### UEPMNUM

Adresse eines 4-Byte-Felds mit der Nachrichtennummer. Bei CICSplex SM-Nachrichten enthält dieses Feld binäre Nullen.

#### UEPMDOM

Adresse eines 2-Byte-Felds mit der Domänen-ID der CICS-Nachricht. Bei CICSplex SM-Nachrichten enthält dieses Feld binäre Nullen.

#### UEPMROU

Adresse einer Gruppe von bis zu 28 Routing-Codes. Routing-Codes müssen Zahlen im Bereich von 1 bis 28 sein.

#### UEPMNRC

Adresse eines Halbworts mit der Anzahl der Routing-Codes, die in der Gruppe der Routing-Codes enthalten sind.

#### UEPMTDQ

Adresse einer Gruppe von bis zu 25 Namen von Warteschlangen mit transienten Daten, an die die Nachricht gesendet werden soll. Namen von Warteschlangen mit transienten Daten müssen aus 4 alphanumerischen Zeichen bestehen.

#### UEPMNTD

Adresse eines Halbworts mit der Anzahl der Warteschlangen mit transienten Daten, die in der Gruppe der Warteschlangen enthalten sind.

#### UEPINSN

Adresse eines 2-Byte-Felds mit der Anzahl der Nachrichteneinfügungen.

## UEPINS A

Adresse einer Gruppe, in der jedes Element Informationen zu einer einzelnen Nachrichteneinfügung enthält. Die Größe der Gruppe hängt von der Anzahl der Einfügungen ab. Jedes Gruppenelement hat die folgende Struktur:

INSERT_FORMAT_P	DS	A	Address of the 1-byte insert type-code, which has one of the following hexadecimal values:
		0	Not present
		1	Character
		2	Hexadecimal
		3	Decimal
		4	The insert is a number representing one item in a list of options. (See the example below.)
		7	Integer (no reformatting of numbers > 999)
INSERT_P	DS	A	Address of the message insert
INSERT_LENGTH_P	DS	A	Address of a fullword containing the length of the insert
INSERT_TYPE_P	DS	A	Reserved.

Sie können die Reihenfolge der Einfügungen in der Gruppe anhand des Eintrags für die jeweilige Nachricht in den CICS messages finden. Beispiel:

DFHFC0531 *datum uhrzeit anwendungs-id* Automatic journal  
*journal journalname*, opened for file *dateiname* is not of type  
MVS Module *modul*

Bei den XMEOUT-Einfügungen handelt es sich um *datum*, *uhrzeit*, *anwendungs-id*, *journal*, *journalname*, *dateiname* und *modul*. Die vierte Einfügung (*journal*) ist die Nummer, die in der Dateidefinition für JOURNAL angegeben wurde.

## UEPNRTE

Adresse des Flags aus einem Zeichen, das angibt, ob die Nachricht von XMEOUT umgeleitet werden kann oder nicht. Mögliche Werte:

C'0:' Die Nachricht kann weitergeleitet werden.

C'1:' Die Nachricht kann nicht weitergeleitet werden.

## UEPCPID

Adresse einer 3-Byte-Produkt-ID. Mögliche Werte:

DFH CICS-Nachrichten.

EYU CICSplex SM-Nachrichten.

## UEPCPDOM

Adresse eines 2-Byte-Felds mit der Domänen-ID der Nachricht.

## UEPCPNUM

Adresse eines 4-Byte-Felds mit der Nachrichtennummer.

## UEPCPSEV

Adresse des Nachrichtenbewertungscodes.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

### UERCBYP

Nachricht für alle Ziele unterdrücken.

CICSplex SM-Nachrichten können nicht unterdrückt werden. Bei diesen Nachrichten wird eine Antwort vom Typ UERCBYP als UERCNORM behandelt.

Die Angabe eines Rückgabecodes vom Typ UERCBYP führt nicht zur Unterdrückung der Ausgabe von Systemereignissen des Typs MESSAGE, die für die Nachricht definiert sind.

#### **XPI-Aufrufe**

WAIT\_MVS kann verwendet werden. **Verwenden Sie keine anderen Aufrufe.**

---

## **Exit für die Überwachungsdomäne (XMNOUT)**

Dieser Exit wird aufgerufen, bevor Überwachungssätze in SMF oder in die Datensatzpuffer geschrieben werden.

XMNOUT wird an folgenden Ereignispunkten aufgerufen:

- Bevor ein Ausnahmeklassenüberwachungssatz an SMF übergeben wird.
- Bevor ein Leistungsklassenüberwachungssatz in den Puffer für Leistungsdatensätze geschrieben wird.
- Bevor ein Transaktionsressourcenüberwachungssatz in den Puffer für Transaktionsressourcendatensätze geschrieben wird.

**Anmerkung:** Wenn in Ihrer CICS-Region sowohl die Leistungsklassenüberwachung als auch die Transaktionsressourcenüberwachung aktiv ist, kann es sein, dass XMNOUT für dasselbe Ereignis zweimal aufgerufen wird. Wenn beispielsweise das Ereignis 'Ende der Task' eintritt und in CICS sowohl Leistungsklassendaten als auch Transaktionsressourcendaten vorliegen, die in den entsprechenden Puffer verschoben werden sollen, dann wird XMNOUT für jeden Überwachungssatztyp jeweils einmal aufgerufen.

Sie können diesen Exit verwenden, um den Datensatz zu untersuchen, um seine Ausgabe an SMF zu unterdrücken oder um die in ihm enthaltenen Daten zu ändern. Sie müssen sicherstellen, dass sämtliche Änderungen, die Sie vornehmen, nicht mit der Wörterverzeichnisbeschreibung der Daten in Konflikt stehen.

Sie können auch Daten zu Leistungsklassendatensätzen hinzufügen. Dazu müssen Sie in der Überwachungssteuertabelle (MCT) Pseudopunkte für die Benutzerereignisüberwachung definieren, um entsprechende Datenfelder mit der erforderlichen Größe und dem erforderlichen Typ zu reservieren.

## **Exit XMNOUT**

Dieser Exit wird aufgerufen, bevor Überwachungssätze in SMF geschrieben oder für einen anschließenden Schreibvorgang in SMF gepuffert werden.

#### **Zeitpunkt des Aufrufs**

Der Exit XMNOUT wird in folgenden Situationen aufgerufen:

- Bevor ein Ausnahmeklassenüberwachungssatz in SMF geschrieben wird.
- Bevor ein Leistungsklassenüberwachungssatz für einen späteren Schreibvorgang in SMF gepuffert wird.
- Bevor ein Transaktionsressourcenüberwachungssatz für einen späteren Schreibvorgang in SMF gepuffert wird.
- Bevor ein Identitätsklassenüberwachungssatz für einen späteren Schreibvorgang in SMF gepuffert wird.

#### **Exitspezifische Parameter**

**UEPTRANID**

Adresse der 4-Byte-Transaktions-ID. Dieses Feld ist bei Taskbeendigung nicht verfügbar.

**UEPUSER**

Adresse der 8-Byte-Benutzer-ID. Dieses Feld ist bei Taskbeendigung nicht verfügbar.

**UEPTERM**

Adresse der 4-Byte-Terminal-ID. Dieses Feld ist bei Taskbeendigung nicht verfügbar.

**UEPPROG**

Adresse des 8-Byte-Anwendungsprogrammnamens. Dieses Feld ist bei Taskbeendigung nicht verfügbar.

**UEPDICT**

Adresse des Wörterverzeichnisses. Die Reihenfolge der Wörterverzeichniseinträge wird durch den Pseudobereich (DSECT) zugeordnet, der über das Makro DFHMCTDR generiert wird. Dieses Feld ist nur für Leistungsklassendatensätze aussagekräftig. Wenn es sich beim Überwachungssatztyp um Ausnahmeklasse (Typ 4), Transaktionsressource (Typ 5) oder Identitätsklasse (Typ 6) handelt, wird dieses Feld auf 0 gesetzt. (Siehe Parameter UEPMRTYP.)

**UEPDICTE**

Adresse der Anzahl von Wörterverzeichniseinträgen als Vollwort. Dieses Feld ist nur für Leistungsklassendatensätze aussagekräftig. Wenn es sich beim Überwachungssatztyp um Ausnahmeklasse (Typ 4), Transaktionsressource (Typ 5) oder Identitätsklasse (Typ 6) handelt, wird dieses Feld auf 0 gesetzt. (Siehe Parameter UEPMRTYP.)

**UEPFCL**

Adresse der Feldconnectorliste mit einer Reihe von Connectorwerten als Halbwort. Dieses Feld ist nur für Leistungsklassendatensätze aussagekräftig. Wenn es sich beim Überwachungssatztyp um Ausnahmeklasse (Typ 4), Transaktionsressource (Typ 5) oder Identitätsklasse (Typ 6) handelt, wird dieses Feld auf 0 gesetzt. (Siehe Parameter UEPMRTYP.)

**UEPFCLNO**

Adresse der Anzahl von Feldconnectors als Vollwort. Dieses Feld ist nur für Leistungsklassendatensätze aussagekräftig. Wenn es sich beim Überwachungssatztyp um Ausnahmeklasse (Typ 4), Transaktionsressource (Typ 5) oder Identitätsklasse (Typ 6) handelt, wird dieses Feld auf 0 gesetzt. (Siehe Parameter UEPMRTYP.)

**UEPMRTYP**

Adresse des Überwachungssatztyps als Halbwort. Der Wert des Überwachungssatztyps kann eine der Zahlen aus folgender Tabelle sein:

*Tabelle 7. Werte für Überwachungssatztypen und ihre Bedeutung*

Satztypwert	Bedeutung
3	Leistungsklassenüberwachungssatz
4	Ausnahmeklassenüberwachungssatz
5	Transaktionsressourcenüberwachungssatz



Tabelle 7. Werte für Überwachungssatztypen und ihre Bedeutung (Forts.)

Satztypwert	Bedeutung
6	Identitätsklassenüberwachungssatz

#### UEPMRLN

Adresse der Überwachungssatzlänge als Vollwort.

#### UEPMREC

Adresse des Überwachungssatzes. Die Länge von UEPMREC wird durch den Parameter UEPMRLN adressiert.

#### UEPSRCTK

Adresse des Tokens der z/OS Workload Manager-Serviceberichterstellungs-klasse für die aktuelle Transaktion. Wenn keine CICS-Unterstützung für z/OS Workload Manager verfügbar ist, ist dieses Token null.

#### UEMPREC

Adresse des Überwachungsleistungssatzes. Dieses Feld ist nur für Leistungsklassendatensätze aussagekräftig. Wenn es sich beim Überwachungssatztyp um Ausnahmeklasse (Typ 4), Transaktionsressource (Typ 5) oder Identitätsklasse (Typ 6) handelt, wird dieses Feld auf 0 gesetzt. (Siehe Parameter UEPMRTYP.) Der von diesem Parameter adressierte Leistungsdatensatz muss mithilfe des Pseudobereichs (DSECT) DFHMNTDS zugeordnet werden und darf nicht mit den Wörterverzeichnisparametern UEPDICT und UEPDICTE zugeordnet werden.

#### Rückgabecodes

##### UERCNORM

Verarbeitung fortsetzen.

##### UERCBYT

Ausgabe von Überwachungssätzen unterdrücken.

##### UERCPURG

Task während des XPI-Aufrufs bereinigt.

#### XPI-Aufrufe

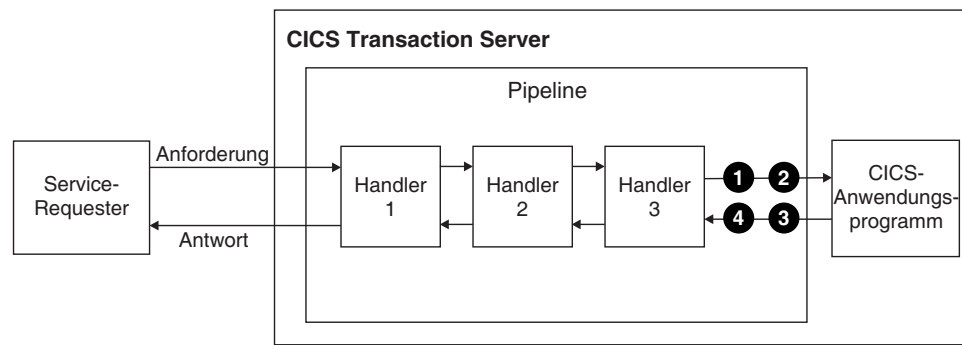
WAIT\_MVS kann verwendet werden. **Verwenden Sie keine anderen Aufrufe.**

## Exits für die Pipelinedomäne

Verwenden Sie Pipelinedomänenexits zum Anpassen der Verarbeitung, die für eingehende und abgehende Web-Services in der Pipeline ausgeführt wird. Mithilfe der Pipelinedomänenexits können Sie auf Container in einer Web-Service-Provider-Pipeline, in einer Web-Service-Requester-Pipeline oder in einer Web-Service-Requester-Pipeline mit Sicherheitsnachrichtenhandler zugreifen.

### Globale Benutzerexitpunkte in der Provider-Pipeline

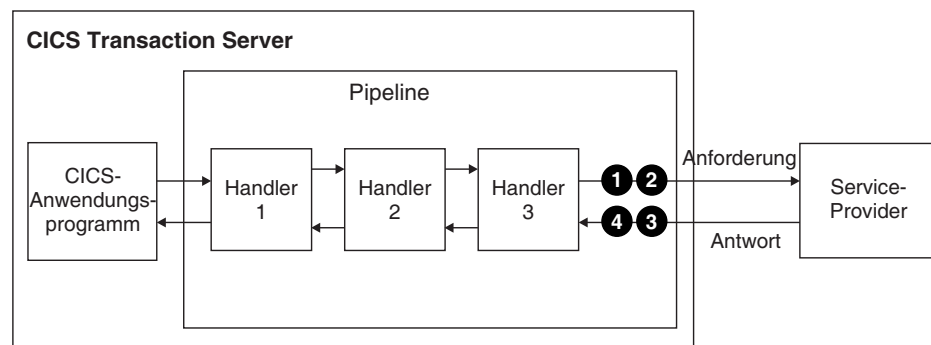
Die globalen Benutzerexitpunkte, die Sie in einer Provider-Pipeline oder einer gesicherten Provider-Pipeline verwenden können, haben das Präfix XWSPR vorangestellt. Das folgende Diagramm zeigt die Reihenfolge, in der die globalen Benutzerexitpunkte verwendet werden können:



❶ = XWSPRRWI    ❷ = XWSPRROI    ❸ = XWSPRROO    ❹ = XWSPRRWO

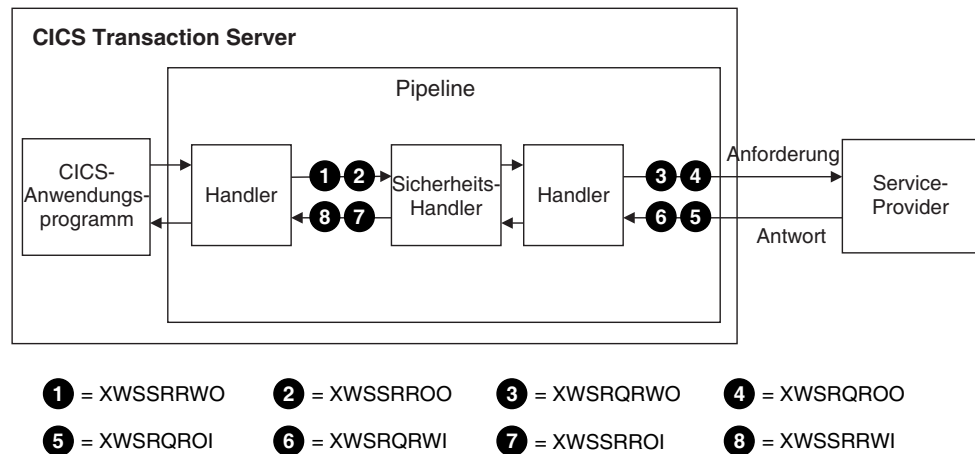
## Globale Benutzerexitpunkte in der Requester-Pipeline

Die globalen Benutzerexitpunkte, die Sie in einer Requester-Pipeline verwenden können, haben das Präfix XWSRQ vorangestellt. Das folgende Diagramm zeigt die Reihenfolge, in der die globalen Benutzerexitpunkte verwendet werden können:



❶ = XWSRQRWO    ❷ = XWSRQROO    ❸ = XWSRQROI    ❹ = XWSRQRWI

Die globalen Benutzerexitpunkte, die Sie in einer gesicherten Requester-Pipeline verwenden können, haben das Präfix XWSSR vorangestellt. Es gibt acht globale Benutzerexitpunkte, die in einer Pipeline verwendet werden können, die einen Sicherheitshandler enthält; vier davon können nur in einer gesicherten Requester-Pipeline verwendet werden, und vier können in jeder beliebigen Requester-Pipeline verwendet werden. Das folgende Diagramm zeigt die Reihenfolge, in der die globalen Benutzerexitpunkte verwendet werden können:



## Exit XWSPRRWI

Verwenden Sie den Exit XWSPRRWI, um im aktuellen Kanal auf Container zuzugreifen, die von der Web-Service-Provider-Anwendung verarbeitet werden sollen, nachdem CICS den Hauptteil der Web-Service-Anforderung in eine Sprachstruktur konvertiert hat und bevor eine beliebige Instanz des Exits XWSPRROI aufgerufen wird.

Sie können diesen Exit verwenden, um API- und SPI-Befehle zum Untersuchen und Aktualisieren beliebiger Informationen in den Containern auszugeben und um einen SOAP-Fehler auszugeben.

### Exitspezifische Parameter

#### UEPTRANID

Adresse der 4-Byte-Transaktions-ID. Der Exit oder die Task wird unter dieser Transaktions-ID ausgeführt.

#### UEPUSER

Adresse der 8-Byte-Benutzer-ID. Die Benutzer-ID ist die ID, die der Transaktions-ID zugeordnet ist.

#### UEPTERM

Adresse der 4-Byte-Terminal-ID. Für Web-Service-Provider ist der Wert null.

#### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungsprogrammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

#### UEPCHANN

Adresse eines 16-Byte-Felds mit dem Namen des aktuellen Kanals. Mithilfe dieses Parameters können Sie den Kanal explizit angeben.

#### UEPCONTR

Adresse eines 16-Byte-Felds mit dem Namen des Datencontainers in dem Kanal, der durch UEPCHANN benannt wird. Dieser Container enthält die Anwendungsdatenstruktur.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

### UERCRIPT

Verarbeitung für die Pipeline nicht fortsetzen.

## XPI-Aufrufe

Es sind keine XPI-Schnittstellen verfügbar.

## API- und SPI-Befehle

Die folgenden Befehle werden unterstützt:

- **EXEC CICS DELETE CONTAINER**
- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**
- **EXEC CICS PUT CONTAINER**
- **EXEC CICS SOAPFAULT ADD**
- **EXEC CICS SOAPFAULT CREATE**
- **EXEC CICS SOAPFAULT DELETE**

## Exit XWSPRROI

Verwenden Sie den Exit XWSPRROI, um im aktuellen Kanal auf Container zuzugreifen, bevor diese von einer Web-Service-Provider-Anwendung verarbeitet werden, aber nachdem eine beliebige Instanz des Exits XWSPRRWI aufgerufen wurde.

Mithilfe dieses Exits können Sie API- und SPI-Befehle ausgeben, um beliebige Informationen zu untersuchen, die von der Web-Service-Geschäftsanwendung verarbeitet werden. Sie können jedoch keinen SOAP-Fehler ausgeben und die Informationen auch nicht aktualisieren. Nach Beendigung des globalen Benutzerexitprogramms werden alle in Register 15 angegebenen Rückgabecodes von CICS ignoriert.

## Exitspezifische Parameter

### UEPTRANID

Adresse der 4-Byte-Transaktions-ID. Der Exit oder die Task wird unter dieser Transaktions-ID ausgeführt.

### UEPUSER

Adresse der 8-Byte-Benutzer-ID. Die Benutzer-ID ist die ID, die der Transaktions-ID zugeordnet ist.

### UEPTERM

Adresse der 4-Byte-Terminal-ID. Für Web-Service-Provider ist der Wert null.

### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungsprogrammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

### UEPCHANN

Adresse eines 16-Byte-Felds mit dem Namen des aktuellen Kanals. Mithilfe dieses Parameters können Sie den Kanal explizit angeben.

#### UEPCONTR

Adresse eines 16-Byte-Felds mit dem Namen des Datencontainers in dem Kanal, der durch UEPCHANN benannt wird. Dieser Container enthält die Anwendungsdatenstruktur.

### Rückgabecodes

#### UERCNORM

Verarbeitung fortsetzen.

### XPI-Aufrufe

Es sind keine XPI-Schnittstellen verfügbar.

### API- und SPI-Befehle

Sie können die folgenden Befehle verwenden:

- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**

## Exit XWSPRROO

Verwenden Sie den Exit XWSPRROO, um auf Container im aktuellen Kanal zuzugreifen, nachdem die Anwendung des Web-Service-Providers die Web-Service-Antwortnachricht ausgegeben hat und bevor CICS den Hauptteil der Antwortnachricht erstellt.

Mithilfe dieses Exits können Sie API- und SPI-Befehle ausgeben, um die Container im aktuellen Kanal zu untersuchen. Sie können jedoch keinen SOAP-Fehler ausgeben und auch keinen der Container aktualisieren. Nach Beendigung des globalen Benutzerexitprogramms werden alle in Register 15 angegebenen Rückgabecodes von CICS ignoriert.

### Exitspezifische Parameter

#### UEPTRANID

Adresse der 4-Byte-Transaktions-ID. Der Exit oder die Task wird unter dieser Transaktions-ID ausgeführt.

#### UEPUSER

Adresse der 8-Byte-Benutzer-ID. Die Benutzer-ID ist die ID, die der Transaktions-ID zugeordnet ist.

#### UEPTERM

Adresse der 4-Byte-Terminal-ID. Für Web-Service-Provider ist der Wert null.

#### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungsprogrammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

#### UEPCHANN

Adresse eines 16-Byte-Felds mit dem Namen des aktuellen Kanals. Mithilfe dieses Parameters können Sie den Kanal explizit angeben.

## **UEPCONTR**

Adresse eines 16-Byte-Felds mit dem Namen des Datencontainers in dem Kanal, der durch UEPCHANN benannt wird. Dieser Container enthält die Anwendungsdatenstruktur.

## **UEPAPAB**

Ein 1-Byte-Feld, das angibt, ob die Anwendung des Web-Service-Providers ihre Verarbeitung erfolgreich abgeschlossen hat oder nicht. Gültige Werte:

### **UEPAPABY (X'80')**

Die Anwendung des Web-Service-Providers wurde abgebrochen.

### **UEPAPABN (X'40')**

Die Anwendung des Web-Service-Providers hat ihre Verarbeitung erfolgreich abgeschlossen.

## **UEPAPSF**

Ein 1-Byte-Feld, das angibt, ob die Anwendung des Web-Service-Providers einen SOAP-Fehler gesetzt hat oder nicht. Gültige Werte:

### **UEPAPSFY (X'80')**

Die Anwendung des Web-Service-Providers gibt einen SOAP-Fehler zurück.

### **UEPAPSFN (X'40')**

Die Anwendung des Web-Service-Providers gibt keinen SOAP-Fehler zurück.

## **Rückgabecodes**

### **UERCNORM**

Verarbeitung fortsetzen.

## **XPI-Aufrufe**

Es sind keine XPI-Schnittstellen verfügbar.

## **API- und SPI-Befehle**

Die folgenden Befehle werden unterstützt:

- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**

## **Exit XWSPRRWO**

Verwenden Sie den Exit XWSPRRWO, um im aktuellen Kanal auf Container zuzugreifen, die von einer Web-Service-Provider-Anwendung nach einer beliebigen Instanz des Exits XWSPRROO verarbeitet worden sind.

Sie können diesen Exit verwenden, um API- und SPI-Befehle zum Untersuchen und Aktualisieren beliebiger Informationen in den Containern auszugeben und um einen SOAP-Fehler auszugeben. So können Sie beispielsweise zusätzliche SOAP-Header zu einer abgehenden SOAP-Antwort hinzufügen. Sämtliche Aktualisierungen der aktuellen Kanalcontainerdaten werden von CICS verarbeitet und an den Anforderer (Requester) zurückgegeben. Nach Beendigung des globalen Benutzerexitprogramms werden alle in Register 15 angegebenen Rückgabecodes von CICS ignoriert.

## Exitspezifische Parameter

### UEPTRANID

Adresse der 4-Byte-Transaktions-ID. Der Exit oder die Task wird unter dieser Transaktions-ID ausgeführt.

### UEPUSER

Adresse der 8-Byte-Benutzer-ID. Die Benutzer-ID ist die ID, die der Transaktions-ID zugeordnet ist.

### UEPTERM

Adresse der 4-Byte-Terminal-ID. Für Web-Service-Provider ist der Wert null.

### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungsprogrammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

### UEPCHANN

Adresse eines 16-Byte-Felds mit dem Namen des aktuellen Kanals. Mithilfe dieses Parameters können Sie den Kanal explizit angeben.

### UEPCONTR

Adresse eines 16-Byte-Felds mit dem Namen des Datencontainers in dem Kanal, der durch UEPCHANN benannt wird. Dieser Container enthält die Anwendungsdatenstruktur.

### UEPAPAB

Ein 1-Byte-Feld, das angibt, ob die Anwendung des Web-Service-Providers ihre Verarbeitung erfolgreich abgeschlossen hat oder nicht. Gültige Werte:

#### UEPAPABY (X'80')

Die Anwendung des Web-Service-Providers wurde abgebrochen.

#### UEPAPABN (X'40')

Die Anwendung des Web-Service-Providers hat ihre Verarbeitung erfolgreich abgeschlossen.

### UEPAPSF

Ein 1-Byte-Feld, das angibt, ob die Anwendung des Web-Service-Providers einen SOAP-Fehler gesetzt hat oder nicht. Gültige Werte:

#### UEPAPSFY (X'80')

Die Anwendung des Web-Service-Providers gibt einen SOAP-Fehler zurück.

#### UEPAPSFN (X'40')

Die Anwendung des Web-Service-Providers gibt keinen SOAP-Fehler zurück.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

## XPI-Aufrufe

Es sind keine XPI-Schnittstellen verfügbar.

## API- und SPI-Befehle

Sie können die folgenden Befehle verwenden:

- **EXEC CICS DELETE CONTAINER**
- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**
- **EXEC CICS PUT CONTAINER**
- **EXEC CICS SOAPFAULT ADD**
- **EXEC CICS SOAPFAULT CREATE**
- **EXEC CICS SOAPFAULT DELETE**

## Exit XWSRQRWO

Mithilfe des Exits XWSRQRWO können Sie auf Container im aktuellen Kanal zugreifen, bevor diese zur Verarbeitung an den Transport übergeben werden. Dieser Exit wird ausgeführt, nachdem CICS die Sprachstruktur der Anwendung in den Hauptteil einer Web-Service-Anforderung konvertiert hat und bevor CICS den optionalen Exitpunkt XWSRQROO verarbeitet.

Mithilfe dieses Exits können Sie API- und SPI-Befehle ausgeben, um beliebige Informationen in den Containern im aktuellen Kanal zu untersuchen und zu aktualisieren. Diese Informationen sind für alle Instanzen des Exits XWSRQRWO sowie auch für den abgehenden Web-Service-Provider verfügbar. Sie können jedoch keinen SOAP-Fehler ausgeben. Es besteht die Möglichkeit, in Register 15 einen Rückgabecode anzugeben, um anzuzeigen, dass CICS die Pipeline nach Beendigung des globalen Benutzerexitprogramms nicht fortsetzt.

### Exitspezifische Parameter

#### UEPTRANID

Adresse der 4-Byte-Transaktions-ID. Der Exit oder die Task wird unter dieser Transaktions-ID ausgeführt.

#### UEPUSER

Adresse der 8-Byte-Benutzer-ID. Die Benutzer-ID ist die ID, die der Transaktions-ID zugeordnet ist.

#### UEPTERM

Adresse der 4-Byte-Terminal-ID.

#### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungsprogrammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

#### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungsprogrammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

#### UEPCHANN

Adresse eines 16-Byte-Felds mit dem Namen des aktuellen Kanals. Mithilfe dieses Parameters können Sie den Kanal explizit angeben.

#### UEPCONTR

Adresse eines 16-Byte-Felds mit dem Namen des Datencontainers in dem Kanal, der durch UEPCHANN benannt wird. Dieser Container enthält die Anwendungsdatenstruktur.



## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

### UERCRIIP

Verarbeitung für die Pipeline nicht fortsetzen.

## XPI-Aufrufe

Es sind keine XPI-Schnittstellen verfügbar.

## API- und SPI-Befehle

Sie können die folgenden Befehle verwenden:

- **EXEC CICS DELETE CONTAINER**
- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**
- **EXEC CICS PUT CONTAINER**

## Exit XWSRQROO

Mithilfe des Exits XWSRQROO können Sie auf Container im aktuellen Kanal zugreifen, bevor diese zur Verarbeitung an den Transport übergeben werden. Dieser Exit wird ausgeführt, nachdem eine beliebige Instanz des Exits XWSRQRWO verarbeitet wurde und vor den abgehenden Datenflüssen im Web-Service-Transport.

Mithilfe dieses Exits können Sie API- und SPI-Befehle ausgeben, um beliebige Informationen in den Containern zu untersuchen. Diese Informationen werden vom abgehenden Web-Service-Provider verarbeitet. Sie können jedoch keinen SOAP-Fehler ausgeben. Nach Beendigung des globalen Benutzerexitprogramms werden alle in Register 15 angegebenen Rückgabecodes von CICS ignoriert.

## Exitspezifische Parameter

### UEPTRANID

Adresse der 4-Byte-Transaktions-ID. Der Exit oder die Task wird unter dieser Transaktions-ID ausgeführt.

### UEPUSER

Adresse der 8-Byte-Benutzer-ID. Die Benutzer-ID ist die ID, die der Transaktions-ID zugeordnet ist.

### UEPTERM

Adresse der 4-Byte-Terminal-ID.

### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungsprogrammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

### UEPCHANN

Adresse eines 16-Byte-Felds mit dem Namen des aktuellen Kanals. Mithilfe dieses Parameters können Sie den Kanal explizit angeben.

### UEPCONTR

Adresse eines 16-Byte-Felds mit dem Namen des Datencontainers in dem Kanal, der durch UEPCHANN benannt wird. Dieser Container enthält die Anwendungsdatenstruktur.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

### UERCRIIP

Verarbeitung für die Pipeline nicht fortsetzen.

## XPI-Aufrufe

Es sind keine XPI-Schnittstellen verfügbar.

## API- und SPI-Befehle

Sie können die folgenden Befehle verwenden:

- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**

## Exit XWSRQROI

Mithilfe des Exits XWSRQROI können Sie auf Container im aktuellen Kanal zugreifen, nachdem diese vom Transport als Web-Service-Antwort verarbeitet worden sind. Der Exit XWSRQROI wird direkt aufgerufen, nachdem CICS den ausgehenden Web-Service-Provider verarbeitet hat. Er kann auch vor einer beliebigen Instanz des Exits XWSRQRWI aufgerufen werden.

Mithilfe dieses Exits können Sie API- und SPI-Befehle ausgeben, um beliebige Informationen in den Containern zu untersuchen. Diese Informationen werden vom abgehenden Web-Service-Provider verarbeitet. Sie können jedoch keinen SOAP-Fehler ausgeben. Nach Beendigung des globalen Benutzerexitprogramms werden alle in Register 15 angegebenen Rückgabecodes von CICS ignoriert.

## Exitspezifische Parameter

### UEPTRANID

Adresse der 4-Byte-Transaktions-ID. Der Exit oder die Task wird unter dieser Transaktions-ID ausgeführt.

### UEPUSER

Adresse der 8-Byte-Benutzer-ID. Die Benutzer-ID ist die ID, die der Transaktions-ID zugeordnet ist.

### UEPTERM

Adresse der 4-Byte-Terminal-ID.

### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungsprogrammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

### UEPCHANN

Adresse eines 16-Byte-Felds mit dem Namen des aktuellen Kanals. Mithilfe dieses Parameters können Sie den Kanal explizit angeben.

### UEPCONTR

Adresse eines 16-Byte-Felds mit dem Namen des Datencontainers in dem Kanal, der durch UEPCHANN benannt wird. Dieser Container enthält die Anwendungsdatenstruktur.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

## XPI-Aufrufe

Es sind keine XPI-Schnittstellen verfügbar.

## API- und SPI-Befehle

Sie können die folgenden Befehle verwenden:

- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**

## Exit XWSRQRWI

Mithilfe des Exits XWSRQRWI können Sie auf Container im aktuellen Kanal zugreifen, nachdem diese vom Transport als Web-Service-Antwort verarbeitet worden sind. Der Exit XWSRQRWI wird direkt aufgerufen, nachdem CICS die eingehende Web-Service-Antwort verarbeitet hat. Darüber hinaus wird er nach jeder Instanz des Exits XWSRQROI aufgerufen.

Mithilfe dieses Exits können Sie API- und SPI-Befehle ausgeben, um beliebige Informationen in den Containern zu untersuchen und zu aktualisieren. Diese Informationen werden vom abgehenden Web-Service-Provider verarbeitet und von der zugehörigen Web-Service-Requester-Anwendung empfangen. Sie können jedoch keinen SOAP-Fehler ausgeben. Nach Beendigung des globalen Benutzerexitprogramms werden alle in Register 15 angegebenen Rückgabecodes von CICS ignoriert.

## Exitspezifische Parameter

### UEPTRANID

Adresse der 4-Byte-Transaktions-ID. Der Exit oder die Task wird unter dieser Transaktions-ID ausgeführt.

### UEPUSER

Adresse der 8-Byte-Benutzer-ID. Die Benutzer-ID ist die ID, die der Transaktions-ID zugeordnet ist.

### UEPTERM

Adresse der 4-Byte-Terminal-ID.

### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungsprogrammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

### UEPCHANN

Adresse eines 16-Byte-Felds mit dem Namen des aktuellen Kanals. Mithilfe dieses Parameters können Sie den Kanal explizit angeben.

### UEPCONTR

Adresse eines 16-Byte-Felds mit dem Namen des Datencontainers in dem Kanal, der durch UEPCHANN benannt wird. Dieser Container enthält die Anwendungsdatenstruktur.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

## XPI-Aufrufe

Es sind keine XPI-Schnittstellen verfügbar.

## API- und SPI-Befehle

Sie können die folgenden Befehle verwenden:

- **EXEC CICS DELETE CONTAINER**
- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**
- **EXEC CICS PUT CONTAINER**

## Exit XWSSRRWO

Mithilfe des Exits XWSSRRWO können Sie auf Container im aktuellen Kanal zugreifen, bevor diese zur Verarbeitung an den Transport übergeben werden; hierbei fungiert CICS als gesicherter Web-Serviceanforderer. Dieser Exit wird ausgeführt, nachdem CICS die Sprachstruktur der Anwendung in einen Web-Service-Anforderungshauptteil konvertiert hat und bevor CICS den optionalen Exitpunkt XWSSRROO verarbeitet, und bevor die Verschlüsselung durch den Sicherheitshandler der Pipeline erfolgt.

Mithilfe dieses Exits können Sie API- und SPI-Befehle ausgeben, um beliebige Informationen in den Containern zu untersuchen und zu aktualisieren. Diese Informationen sind für alle Instanzen des Exits XWSSRRWO sowie auch für den abgehenden Web-Service-Provider verfügbar. Sie können jedoch keinen SOAP-Fehler ausgeben. Es besteht die Möglichkeit, in Register 15 einen Rückgabecode anzugeben, um anzuzeigen, dass CICS die Pipeline nach Beendigung des globalen Benutzerexitprogramms nicht fortsetzt.

Wenn die Pipeline keinen Sicherheitshandler enthält, wird dieser Exit nicht gesteuert. Informationen zu Instanzen der Pipeline, die keinen Sicherheitshandler enthalten, finden Sie unter Abschnitt „Exit XWSRQRWO“ auf Seite 198.

## Exitspezifische Parameter

### UEPTRANID

Adresse der 4-Byte-Transaktions-ID. Der Exit oder die Task wird unter dieser Transaktions-ID ausgeführt.

### UEPUSER

Adresse der 8-Byte-Benutzer-ID. Die Benutzer-ID ist die ID, die der Transaktions-ID zugeordnet ist.

### UEPTERM

Adresse der 4-Byte-Terminal-ID.

### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungsprogrammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

## **UEPCHANN**

Adresse eines 16-Byte-Felds mit dem Namen des aktuellen Kanals. Mithilfe dieses Parameters können Sie den Kanal explizit angeben.

## **UEPCONTR**

Adresse eines 16-Byte-Felds mit dem Namen des Datencontainers in dem Kanal, der durch UEPCHANN benannt wird. Dieser Container enthält die Anwendungsdatenstruktur.

## **Rückgabecodes**

### **UERCNORM**

Verarbeitung fortsetzen.

### **UERCRIPIP**

Verarbeitung für die Pipeline nicht fortsetzen.

## **XPI-Aufrufe**

Es sind keine XPI-Schnittstellen verfügbar.

## **API- und SPI-Befehle**

Sie können die folgenden Befehle verwenden:

- **EXEC CICS DELETE CONTAINER**
- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**
- **EXEC CICS PUT CONTAINER**

## **Exit XWSSRROO**

Mithilfe des Exits XWSSRROO können Sie auf Container im aktuellen Kanal zugreifen, bevor diese zur Verarbeitung an den Transport übergeben werden; hierbei fungiert CICS als gesicherter Web-Serviceanforderer. Dieser Exit wird ausgeführt, nachdem eine beliebige Instanz des Exits XWSSRRWO verarbeitet wurde und vor der Verschlüsselung abgehender Datenflüssen im Web-Service-Transport.

Wenn die Pipeline keinen Sicherheitshandler enthält, wird dieser Exit nicht gesteuert. Informationen zu Instanzen der Pipeline, die keinen Sicherheitshandler enthalten, finden Sie unter „Exit XWSRQROO“ auf Seite 199.

## **Exitspezifische Parameter**

### **UEPTRANID**

Adresse der 4-Byte-Transaktions-ID. Der Exit oder die Task wird unter dieser Transaktions-ID ausgeführt.

### **UEPUSER**

Adresse der 8-Byte-Benutzer-ID. Die Benutzer-ID ist die ID, die der Transaktions-ID zugeordnet ist.

### **UEPTERM**

Adresse der 4-Byte-Terminal-ID.

### **UEPPROG**

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungsprogrammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

## **UEPCHANN**

Adresse eines 16-Byte-Felds mit dem Namen des aktuellen Kanals. Mithilfe dieses Parameters können Sie den Kanal explizit angeben.

## **UEPCONTR**

Adresse eines 16-Byte-Felds mit dem Namen des Datencontainers in dem Kanal, der durch UEPCHANN benannt wird. Dieser Container enthält die Anwendungsdatenstruktur.

## **Rückgabecodes**

### **UERCNORM**

Verarbeitung fortsetzen.

## **XPI-Aufrufe**

Es sind keine XPI-Schnittstellen verfügbar.

## **API- und SPI-Befehle**

Sie können die folgenden Befehle verwenden:

- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**

## **Exit XWSSRROI**

Mithilfe des Exits XWSRQROI können Sie auf Container im aktuellen Kanal zugreifen, nachdem diese vom Transport als Web-Service-Antwort verarbeitet worden sind; hierbei fungiert CICS als gesicherter Web-Serviceanforderer. Dieser Exit wird ausgeführt, nachdem CICS die Web-Service-Antwort verarbeitet hat, sowie vor jeder Instanz des Exits XWSSRRWI.

Mithilfe dieses Exits können Sie API- und SPI-Befehle ausgeben, um beliebige Informationen in den Containern zu untersuchen. Diese Informationen werden vom abgehenden Web-Service-Provider verarbeitet. Sie können jedoch keinen SOAP-Fehler ausgeben. Nach Beendigung des globalen Benutzerexitprogramms werden alle in Register 15 angegebenen Rückgabecodes von CICS ignoriert.

Wenn die Pipeline keinen Sicherheitshandler enthält, wird dieser Exit nicht gesteuert. Informationen zu Instanzen der Pipeline, die keinen Sicherheitshandler enthalten, finden Sie unter „Exit XWSRQROI“ auf Seite 200.

## **Exitspezifische Parameter**

### **UEPTRANID**

Adresse der 4-Byte-Transaktions-ID. Der Exit oder die Task wird unter dieser Transaktions-ID ausgeführt.

### **UEPUSER**

Adresse der 8-Byte-Benutzer-ID. Die Benutzer-ID ist die ID, die der Transaktions-ID zugeordnet ist.

### **UEPTERM**

Adresse der 4-Byte-Terminal-ID.

### **UEPPROG**

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungsprogrammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

## **UEPCHANN**

Adresse eines 16-Byte-Felds mit dem Namen des aktuellen Kanals. Mithilfe dieses Parameters können Sie den Kanal explizit angeben.

## **UEPCONTR**

Adresse eines 16-Byte-Felds mit dem Namen des Datencontainers in dem Kanal, der durch UEPCHANN benannt wird. Dieser Container enthält die Anwendungsdatenstruktur.

## **Rückgabecodes**

### **UERCNORM**

Verarbeitung fortsetzen.

## **XPI-Aufrufe**

Es sind keine XPI-Schnittstellen verfügbar.

## **API- und SPI-Befehle**

Sie können die folgenden Befehle verwenden:

- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**

## **Exit XWSSRRWI**

Mithilfe des Exits XWSSRRWI können Sie auf Container im aktuellen Kanal zugreifen, nachdem diese vom Transport als Web-Service-Antwort verarbeitet worden sind; hierbei fungiert CICS als gesicherter Web-Serviceanforderer. Dieser Exit wird ausgeführt, nachdem CICS die Web-Service-Antwort verarbeitet hat, sowie nach jeder Instanz des Exits XWSSRROI.

Mithilfe dieses Exits können Sie API- und SPI-Befehle ausgeben, um beliebige Informationen in den Containern zu untersuchen und zu aktualisieren. Diese Informationen werden von der Anwendung des Web-Serviceanforderers verarbeitet. Sie können jedoch keinen SOAP-Fehler ausgeben. Nach Beendigung des globalen Benutzerexitprogramms werden alle in Register 15 angegebenen Rückgabecodes von CICS ignoriert.

Wenn die Pipeline keinen Sicherheitshandler enthält, wird dieser Exit nicht gesteuert. Informationen zu Instanzen der Pipeline, die keinen Sicherheitshandler enthalten, finden Sie unter „Exit XWSRQRWI“ auf Seite 201.

## **Exitspezifische Parameter**

### **UEPTRANID**

Adresse der 4-Byte-Transaktions-ID. Der Exit oder die Task wird unter dieser Transaktions-ID ausgeführt.

### **UEPUSER**

Adresse der 8-Byte-Benutzer-ID. Die Benutzer-ID ist die ID, die der Transaktions-ID zugeordnet ist.

### **UEPTERM**

Adresse der 4-Byte-Terminal-ID.

### **UEPPROG**

Adresse des 8-Byte-Anwendungsprogrammnamens. Der Anwendungspro-

grammname ist der Name des Programms, das den Befehl **INVOKE SERVICE** oder **INVOKE WEBSERVICE** ausgegeben hat.

#### **UEPCHANN**

Adresse eines 16-Byte-Felds mit dem Namen des aktuellen Kanals. Mithilfe dieses Parameters können Sie den Kanal explizit angeben.

#### **UEPCONTR**

Adresse eines 16-Byte-Felds mit dem Namen des Datencontainers in dem Kanal, der durch UEPCHANN benannt wird. Dieser Container enthält die Anwendungsdatenstruktur.

### **Rückgabecodes**

#### **UERCNORM**

Verarbeitung fortsetzen.

### **XPI-Aufrufe**

Es sind keine XPI-Schnittstellen verfügbar.

### **API- und SPI-Befehle**

Sie können die folgenden Befehle verwenden:

- **EXEC CICS DELETE CONTAINER**
- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**
- **EXEC CICS PUT CONTAINER**

---

## **Exits für das Programmsteuerprogramm (XPCREQ, XPCERES, XPCREQC, XPCFTCH, XPCHAIR, XPCTA und XPCABND)**

Diese Exits werden vor oder nach CICS-Programmsteueroperationen aufgerufen. Dies gilt auch für Programmverbindungsanforderungen, Programme, die die Steuerung erhalten, und Transaktionsabbrüche.

## **Programmsteuerungsexits XPCREQ, XPCERES und XPCREQC**

Diese Exits werden vom EXEC-Schnittstellenprogramm aufgerufen, bevor eine Verknüpfungsanforderung verarbeitet wird, bevor CICS dynamisch weitergeleitete Verknüpfungsanforderungen verarbeitet oder nachdem eine Verknüpfungsanforderung abgeschlossen wurde.

#### **XPCREQ**

XPCREQ wird vom EXEC-Schnittstellenprogramm aufgerufen, bevor eine Verknüpfungsanforderung verarbeitet wird. Wenn es sich bei der Anforderung um eine Verbindung zu verteilten Programmen (Distributed Program Link, DPL) handelt, wird der Exit XPCREQ auf beiden Seiten der Verbindung gesteuert, d. h. sowohl in der Clientregion als auch in der Serverregion. Die Adresse der Parameterliste der Anwendung wird (in UEPCLPS) an das Exitprogramm übergeben, das diese Liste erforderlichenfalls auch ändern kann. Sie können diesen Exit beispielsweise verwenden, um die SYSID zum Zeitpunkt einer DPL-Anforderung zu ändern. Sie können ein Anwendungsprogramm schreiben, um eine Liste der SYSIDs in einem globalen Arbeitsbereich (Global Work Area, GWA) zu verwalten. Das globale Benutzerexitprogramm kann Zugriff auf den GWA erhalten und die dort gespeicherten Informationen verwenden, um DPL-Anforderungen umzuleiten.



**Anmerkung:**

1. Die Attribute der lokalen PROGRAM-Ressource werden nicht an das Exitprogramm übergeben. Wenn das Exitprogramm den Wert eines Attributs benötigt, kann das Programm einen Befehl vom Typ **EXEC CICS INQUIRE PROGRAM** ausgeben.
2. Wenn Sie XPCREQ verwenden, um die Ziel-SYSID zu ändern, bedenken Sie Folgendes:
  - a. Wenn mit SYSID eine ferne Region angegeben wird, wird keine Referenz zur lokalen PROGRAM-Ressource angegeben. In der fernen Region wird das Programm unter der Transaktions-ID (TRANSID) der Transaktion in der Clientregion ausgeführt und nicht unter der TRANSID, die in der PROGRAM-Ressource in der Clientregion angegeben ist.
  - b. Wenn mit SYSID die lokale Region angegeben wird, behandelt CICS die Verknüpfungsanforderung so, als wäre SYSID nicht angegeben worden. Die lokale PROGRAM-Ressource wird berücksichtigt.
  - c. Der Exit XPCREQ wird sowohl durch interne Anforderungen aufgerufen, die vom CICS-Code ausgegeben werden, als auch durch Anforderungen, die von Anwendungen ausgegeben werden.

**XPCERES**

XPCERES wird vom EXEC-Schnittstellenprogramm aufgerufen, bevor CICS eine der folgenden Arten von dynamisch weitergeleiteten Verknüpfungsanforderungen verarbeitet:

- DPL-Aufrufe (Distributed Program Link).
- Link3270-Brückenansforderungen.

XPCERES wird in folgenden Situationen aufgerufen:

- Nach dem Exit XPCREQ und vor dem Exit XPCREQC, sofern diese Exits aktiviert sind. Die bedeutet Folgendes:
  - Wenn ein Exitprogramm für XPCREQ die Anforderung umgeht, wird XPCERES nicht aufgerufen.
  - Wenn ein Exitprogramm für XPCREQ die Befehlsparameterliste ändert, muss XPCERES die geänderte Anforderung bearbeiten.
- In der Zielregion, an die die Anforderung weitergeleitet wurde.
- Nur dann, wenn der Exit aktiviert ist. Aktivieren Sie diesen Exit nur dann in Anwendungsverwaltungsregionen (AORs), wenn DPL- und Link3270-Brückenansforderungen dynamisch weitergeleitet werden können.
- Sowohl durch interne Anforderungen, die vom CICS-Code ausgegeben werden, als auch durch Anforderungen, die von Anwendungen ausgegeben werden.

Der Exit XPCERES wird in folgenden Situationen nicht aufgerufen:

- Bei statisch weitergeleiteten Anforderungen.
- Wenn der Exit inaktiviert ist.
- Wenn ein Exitprogramm für XPCREQ die Anforderung umgeht.

Mithilfe von XPCERES können Sie überprüfen, ob alle Ressourcen, die für das verbundene Programm erforderlich sind, in der Zielregion verfügbar sind. Wenn das Programm inaktiviert ist oder eine erforderliche Datei fehlt, kann Ihr Exitprogramm dem Programm für dynamisches Routing die Gelegenheit geben, die Anforderung an eine andere Region weiterzuleiten. Setzen Sie hierfür den Rückgabecode UERCRESU. CICS führt dann die folgenden Verarbeitungsschritte aus:

1. Im Kommunikationsbereich (COMMAREA) des Routing-Programms setzt CICS das Feld DYRError auf 'F' (Ressource nicht verfügbar).
2. Bei einer fehlgeschlagenen Routing-Auswahl ruft CICS das Routing-Programm in der Routing-Region auf.
3. CICS gibt eine Bedingung vom Typ RESUNAVAIL in dem vom Spiegel in der Zielregion ausgeführten EXEC CICS LINK-Befehl zurück. Diese Bedingung wird nicht an das Anwendungsprogramm zurückgegeben.

CICS ignoriert sämtliche Änderungen, die das Exitprogramm an den Werten beliebiger Exitparameter vorgenommen hat. Das Exitprogramm kann einen Rückgabecode festlegen, aber keine Parameter ändern.

Informationen für den Benutzer zum dynamischen Routing von DPL-Anforderungen finden Sie unter Dynamisches Routing für DPL-Anforderungen ausführen. Informationen für den Benutzer zum dynamischen Routing von Link3270-Brückenforderungen finden Sie unter Lastausgleich mit Link3270-Brücke verwenden. Programmierinformationen zum Schreiben eines Programms für dynamisches Routing zur Weiterleitung von DPL-Anforderungen finden Sie unter Routing DPL requests dynamically. Programmierinformationen zum Schreiben eines Programms für dynamisches Routing zur Weiterleitung von Link3270-Brückenforderungen finden Sie unter Routing bridge requests dynamically.

#### **XPCREQC**

XPCREQC wird nach Abschluss einer Verknüpfungsanforderung aufgerufen. Sie können diesen Exit verwenden, um mithilfe des Felds EIBRESP oder EIBRESP2 eine Antwort an die Anwendung zurückzugeben. Solche Antworten können verwendet werden, um Statusinformationen zu einer Verknüpfungsanforderung auf dem aktuellen Stand zu halten. Wenn beispielsweise eine Verknüpfungsanforderung fehlschlägt, weil eine Verbindung nicht verfügbar ist, kann XPCREQC EIBRESP=500 (einen von CICS nicht verwendeten Antwortcode) festlegen, um auf den Fehler hinzuweisen, sodass die Anwendung zusammen mit dem anderen Exit XPCREQ eine geeignete Vorgehensweise ermitteln kann.

**Anmerkung:** Der Exit XPCREQC wird sowohl durch interne Anforderungen aufgerufen, die vom CICS-Code ausgegeben werden, als auch durch Anforderungen, die von Anwendungen ausgegeben werden.

#### **Exit XPCREQ**

Der Exit XPCREQ wird vom EXEC-Schnittstellenprogramm aufgerufen, bevor eine Verknüpfungsanforderung verarbeitet wird.

#### **Zeitpunkt des Aufrufs**

Durch das EXEC-Schnittstellenprogramm, bevor eine Verknüpfungsanforderung verarbeitet wird.

#### **Exitspezifische Parameter**

##### **UEPCLPS**

Adresse der Befehlsparameterliste.

##### **UEPPCTOK**

Adresse eines 4-Byte-Tokens, das an XPCREQC übergeben werden soll. Dies ermöglicht es Ihnen beispielsweise, einen Arbeitsbereich an den Exit XPCREQC zu übergeben.

##### **UEPRCODE**

Adresse einer 6 Byte langen hexadezimalen Kopie von EIBRCODE.

**UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

**UEPRES**

Adresse einer 4 Byte langen Kopie von EIBRESP.

**UEPRES2**

Adresse einer 4 Byte langen Kopie von EIBRESP2.

**UEPTSTOK**

Adresse eines 4-Byte-Tokens, das während des gesamten Lebenszyklus einer Task gültig ist. Siehe Using the task token UEPTSTOK.

**UEPRSRCE**

Adresse einer 8 Zeichen langen Kopie des EIB-Ressourcenwerts, EIBRSRCE.

**UEP\_PC\_PBTOK**

Adresse eines 4-Byte-Felds mit dem Leistungsbloktoken (Performance Block Token, PBTOK) von z/OS Workload Manager (WLM). Mithilfe dieses Tokens kann ein Exitprogramm auf Informationen (wie beispielsweise das Serviceklassetoken SERVCLS) im WLM-Leistungsblock zugreifen. Hierzu muss das Programm das WLM EXTRACT-Makro IWMMEXTR verwenden und dabei das Leistungsbloktoken als Eingabeparameter MONTKN übergeben. Weitere Informationen zum Makro IWMMEXTR finden Sie unter z/OS MVS Programming: Workload Management Services.

Exitprogramme dürfen nicht versuchen, den Leistungsblock zu ändern, da die Ergebnisse eines solchen Versuchs unvorhersehbar sind.

**Rückgabecodes****UERCBY**

Die Anforderung soll von der Programmsteuerung ignoriert werden.

**UERCNORM**

Verarbeitung fortsetzen.

**UERCPU**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

Wenngleich der Exit die Verwendung der XPI-Aufrufe GETMAIN und FREEMAIN zulässt, wird empfohlen, stattdessen die Befehle EXEC CICS GETMAIN und EXEC CICS FREEMAIN zu verwenden.

**API- und SPI-Aufrufe**

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

EXEC CICS SHUTDOWN  
EXEC CICS XCTL

**Exit XPCERES**

Der Exit XPCERES wird vom EXEC-Schnittstellenprogramm aufgerufen, bevor eine Programmverknüpfungsanforderung oder eine Link3270-Brückenanforderung verarbeitet wird, die dynamisch an diese Region weitergeleitet worden ist, wobei die Routing-Region die Bedingung "Ressource nicht verfügbar" unterstützt.

## Exitspezifische Parameter

**Anmerkung:** CICS ignoriert sämtliche Änderungen, die das Exitprogramm an den Werten beliebiger Exitparameter vorgenommen hat. Das Exitprogramm kann einen Rückgabecode festlegen, aber keine Parameter ändern.

### UEPCLPS

Adresse der Befehlsparameterliste.

### UEPPCTOK

Adresse eines 4-Byte-Tokens, das an XPCREQC übergeben werden soll.

### UEPRCODE

Adresse einer 6 Byte langen hexadezimalen Kopie von EIBRCODE.

### UEPRECUR

Adresse eines Halbwortzählers für Rekursionen. Da der Exit XPCE-RES in derselben Transaktion niemals rekursiv aufgerufen werden kann, ist der Wert dieses Feldes immer 0.

### UEPRES P

Adresse einer 4 Byte langen Kopie von EIBRESP.

### UEPRES P2

Adresse einer 4 Byte langen Kopie von EIBRESP2.

### UEPTSTOK

Adresse eines 4-Byte-Tokens, das während des gesamten Lebenszyklus einer Task gültig ist. Siehe Using the task token UEPTSTOK.

### UEPRSRCE

Adresse einer 8 Zeichen langen Kopie des EIB-Ressourcenwerts, EIBRSRCE.

### UEP\_PC\_PBTOK

Adresse eines 4-Byte-Felds mit dem Leistungsbloktoken (Performance Block Token, PBTOK) von z/OS Workload Manager (WLM). Mithilfe dieses Tokens kann ein Exitprogramm auf Informationen (wie beispielsweise das Serviceklassetoken SERVCLS) im WLM-Leistungsblock zugreifen. Hierzu muss das Programm das WLM EXTRACT-Makro IWMMEXTR verwenden und dabei das Leistungsbloktoken als Eingabeparameter MONTKN übergeben. Weitere Informationen zum Makro IWMMEXTR finden Sie unter z/OS MVS Programming: Workload Management Services.

Exitprogramme dürfen nicht versuchen, den Leistungsblock zu ändern, da die Ergebnisse eines solchen Versuchs unvorhersehbar sind.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

### UERCPURG

Task während des XPI-Aufrufs bereinigt.

### UERCRESU

Eine erforderliche Ressource ist nicht verfügbar. Wenn Sie diesen Wert festlegen, wird CICS die weitergeleitete Anforderung zurück-

weisen und den Wert 'F' (Ressource nicht verfügbar) im Feld DYRError im Kommunikationsbereich des Routing-Programms zurückgeben.

#### **XPI-Aufrufe**

Es können alle verwendet werden.

Wenngleich der Exit die Verwendung der XPI-Aufrufe GETMAIN und FREEMAIN zulässt, wird empfohlen, stattdessen die Befehle EXEC CICS GETMAIN und EXEC CICS FREEMAIN zu verwenden.

#### **API- und SPI-Aufrufe**

Es können alle API- und SPI-Aufrufe mit Ausnahme von EXEC CICS SHUTDOWN und EXEC CICS XCTL verwendet werden.

### **Exit XPCREQC**

Der Exit XPCREQC wird beim Abschluss einer Verknüpfungsanforderung (LINK) zur Programmsteuerung aufgerufen.

#### **Zeitpunkt des Aufrufs**

Bei Abschluss einer Verknüpfungsanforderung (LINK) zur Programmsteuerung.

#### **Exitspezifische Parameter**

##### **UEPCLPS**

Adresse der Befehlsparameterliste.

##### **UEPPCTOK**

Adresse eines 4-Byte-Tokens, das von XPCREQ übergeben wird. Dadurch kann XPCREQ beispielsweise einen Arbeitsbereich an XPCREQC übergeben.

##### **UEPRCODE**

Adresse einer 6 Byte langen hexadezimalen Kopie von EIBRCODE.

##### **UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

##### **UEPRES P**

Adresse einer 4 Byte langen Kopie von EIBRESP.

##### **UEPRES P2**

Adresse einer 4 Byte langen Kopie von EIBRESP2.

##### **UEPTSTOK**

Adresse eines 4-Byte-Tokens, das während des gesamten Lebenszyklus einer Task gültig ist. Siehe Using the task token UEPTSTOK.

##### **UEPRSRCE**

Adresse einer 8 Zeichen langen Kopie des EIB-Ressourcenwerts, EIBRSRCE.

##### **UEP\_PC\_REMOTE\_SYSTEM**

Wenn die Anforderung an eine ferne Region gesendet werden soll, ist dies die Adresse eines Bereichs mit dem 4-Byte-Namen der fernen Region. (Die ferne Region kann beispielsweise durch die Option SYSID des Befehls EXEC CICS LINK, durch die Funktionsverlagerung, durch das Workload-Management oder durch die Option REMOTESYSTEM der Programmdefinition angegeben worden sein.)

Wenn die Anforderung in der lokalen Region ausgeführt werden soll, ist dieser Parameter die Adresse eines 4-Byte-Bereichs mit Leerzeichen.

#### **UEP\_PC\_REMOTE\_NAME**

Wenn das Programm in einem fernen System ausgeführt werden soll, ist dies die Adresse eines Bereichs mit dem Namen des Programms, wie er im fernen System bekannt ist.

#### **UEP\_PC\_PBTOK**

Adresse eines 4-Byte-Felds mit dem Leistungblocktoken (Performance Block Token, PBTOK) von z/OS Workload Manager (WLM). Mithilfe dieses Tokens kann ein Exitprogramm auf Informationen (wie beispielsweise das Serviceklassetoken SERVCLS) im WLM-Leistungsblock zugreifen. Hierzu muss das Programm das WLM EXTRACT-Makro IWMMEXTR verwenden und dabei das Leistungsblocktoken als Eingabeparameter MONTKN übergeben. Weitere Informationen zum Makro IWMMEXTR finden Sie unter z/OS MVS Programming: Workload Management Services.

Exitprogramme dürfen nicht versuchen, den Leistungsblock zu ändern, da die Ergebnisse eines solchen Versuchs unvorhersehbar sind.

### **Rückgabecodes**

#### **UERCNORM**

Verarbeitung fortsetzen.

#### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

### **XPI-Aufrufe**

Es können alle verwendet werden.

Wenngleich der Exit die Verwendung der XPI-Aufrufe GETMAIN und FREEMAIN zulässt, wird empfohlen, stattdessen die Befehle EXEC CICS GETMAIN und EXEC CICS FREEMAIN zu verwenden.

### **API- und SPI-Aufrufe**

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

EXEC CICS SHUTDOWN

EXEC CICS XCTL

**Anmerkung:** Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor, um keine Schleife zu verursachen. So sind Sie beispielsweise dafür verantwortlich, das Auftreten einer Schleife zu vermeiden, wenn eine Programmsteueranforderung über den Exit XPCREQC oder XPCREQC ausgegeben wird.

In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.

### **Die Befehlsparameterstruktur**

Die Befehlsparameterstruktur besteht aus einer Reihe von Adressen. Die erste Adresse verweist auf den EXEC-Schnittstellendeskriptor (EID), der aus einer Bitfolge besteht, die den Typ der Anforderung beschreibt und jedes mit der Anforderung angegebene Schlüsselwort identifiziert.

Die übrigen Adressen verweisen auf Teile von Daten, die der Anforderung zugeordnet sind. So verweist beispielsweise die zweite Adresse immer auf den Programmnamen. Anhand der Parameter in der Liste können Sie die Werte der

Schlüsselwörter ermitteln. Sie können auch Werte von Parametern ändern, die in der Anforderung angegeben sind. So könnten Sie beispielsweise den Namen des an der Anforderung beteiligten Programms ändern oder die SYSID hinzufügen, um die Verknüpfungsanforderung an ein fernes System weiterzuleiten.

### Indikator für Ende der Parameterliste

Das höchstwertige Bit wird für die letzte Adresse in der Parameterliste gesetzt, um anzuzeigen, dass es sich um die letzte Adresse in der Liste handelt. Bei der Rückkehr vom Benutzerexitprogramm sucht CICS in der Parameterliste nach dem höchstwertigen Bit, um den letzten Parameter ausfindig zu machen. Daher gilt: Wenn Sie die Länge der Parameterliste ändern, müssen Sie auch das höchstwertige Bit neu setzen, um so die aktuelle letzte Adresse zu markieren.

Beispiel: Wenn in der Parameterliste nur die ersten beiden Adressen angegeben werden (PC\_ADDR0, die Adresse des EID, und PC\_ADDR1, die Adresse des in der Verknüpfungsanforderung angegebenen Namens des Programms), dann wird das höchstwertige Bit in PC\_ADDR1 gesetzt. Wenn Sie dann die Parameterliste erweitern, indem Sie beispielsweise die Adresse einer SYSID in PC\_ADDR7 festlegen, müssen Sie das höchstwertige Bit in PC\_ADDR1 zurücksetzen und stattdessen in PC\_ADDR7 setzen.

Die ursprüngliche Parameterliste wird nach Beendigung von XPCREQC in der Version wiederhergestellt, die vor dem Aufruf von XPCREQ vorlag. Daraus folgt, dass die Ausführungsdiagnosefunktion (Execution Diagnostic Facility, EDF) vor **und** nach der Ausführung den ursprünglichen Befehl anzeigt: **EDF zeigt keine Änderungen an, die durch den Exit vorgenommen wurden.**

### Exitspezifischer Parameter UEPCLPS:

Der exitspezifische Parameter UEPCLPS ist die Adresse der Parameterstruktur der Befehlsebene und ist in den Exits XPCREQ und XPCREQC enthalten.

Die Parameterstruktur der Befehlsebene enthält 11 Adressen: PC\_ADDR0 bis PC\_ADDRA. Die Struktur wird im Pseudobereich (DSECT) PC\_ADDR\_LIST definiert, den Sie in Ihr Exitprogramm kopieren sollten, indem Sie die Anweisung COPY DFHPCEDS einschließen.

Die Parameterliste der Befehlsebene ist wie folgt aufgebaut:

#### PC\_ADDR0

Adresse eines 7-Byte-Bereichs, der als EID (EXEC Interface Descriptor) bezeichnet wird und wie folgt aufgebaut ist:

- PC\_GROUP
- PC\_FUNCT
- PC\_BITS1
- PC\_BITS2
- PC\_EIDOPT4
- PC\_EIDOPT5
- PC\_EIDOPT6

#### PC\_GROUP

Immer X'0E'; hierdurch wird angegeben, dass es sich um eine Programmsteueranforderung handelt.

## PC\_FUNCT

Ein Byte zum Definieren des Anforderungstyps. Bei XPCREQ und XPCREQC ist der Wert immer X'02'; hierdurch wird angegeben, dass es sich um eine Anforderung vom Typ LINK handelt.

## PC\_BITS1

Existenzbits zum Definieren der angegebenen Schlüsselwörter mit Werten. Um den Wert abzurufen, der einem Schlüsselwort zugeordnet ist, müssen Sie die entsprechende Adresse aus der Parameterliste der Befehlsebene verwenden. Bevor Sie diese Adresse verwenden, müssen Sie das zugehörige Existenzbit überprüfen, um sicherzustellen, dass die Adresse gültig ist. Wenn das Existenzbit inaktiviert ist, wurde das Schlüsselwort in der Anforderung nicht angegeben, und die Adresse sollte nicht verwendet werden. Die symbolischen und hexadezimalen Werte der Existenzbits lauten wie folgt:

### PC\_EXIST1 (X'80')

Wird gesetzt, wenn die Anforderung das Schlüsselwort PROGRAM enthält. Ist dieses Existenzbit gesetzt, dann ist **PC\_ADDR1** aussagekräftig. (Dieses Bit sollte stets für eine LINK-Anforderung festgelegt werden.)

### PC\_EXIST2 (X'40')

Wird gesetzt, wenn die Anforderung den Parameter COMMA-REA angibt. Ist dieses Existenzbit gesetzt, dann ist **PC\_ADDR2** aussagekräftig.

### PC\_EXIST3 (X'20')

Wird gesetzt, wenn die Anforderung den Parameter LENGTH angibt. Ist dieses Existenzbit gesetzt, dann ist **PC\_ADDR3** aussagekräftig.

### PC\_EXIST4 (X'10')

Wird gesetzt, wenn die Anforderung den Parameter INPUTMSG angibt. Ist dieses Existenzbit gesetzt, dann ist **PC\_ADDR4** aussagekräftig.

### PC\_EXIST5 (X'08')

Wird gesetzt, wenn die Anforderung den Parameter INPUTMSGLEN angibt. Ist dieses Existenzbit gesetzt, dann ist **PC\_ADDR5** aussagekräftig.

### PC\_EXIST6 (X'04')

Wird gesetzt, wenn die Anforderung den Parameter DATALENGTH angibt. Ist dieses Existenzbit gesetzt, dann ist **PC\_ADDR6** aussagekräftig.

### PC\_EXIST7 (X'02')

Wird gesetzt, wenn die Anforderung den Parameter SYSID angibt. Ist dieses Existenzbit gesetzt, dann ist **PC\_ADDR7** aussagekräftig.

### PC\_EXIST8 (X'01')

Wird gesetzt, wenn die Anforderung den Parameter TRANSID angibt. Ist dieses Existenzbit gesetzt, dann ist **PC\_ADDR8** aussagekräftig.

## PC\_BITS2

Ein Byte, das einen der folgenden Werte enthält:



**PC\_EXIST9 (X'80')**

Wird nicht verwendet.

**PC\_EXISTA (X'40')**

Wird gesetzt, wenn die Anforderung den Parameter CHANNEL angibt. Ist dieses Existenzbit gesetzt, dann ist PC\_ADDRA aussagekräftig.

**PC\_EIDOPT4**

Wird nicht von der Programmsteuerung verwendet.

**PC\_EIDOPT5**

Wird nicht von der Programmsteuerung verwendet.

**PC\_EIDOPT6**

Gibt an, ob die Anforderung die Option SYNCONRETURN angibt. Ist dies der Fall, wird X'80' festgelegt.

**PC\_ADDR1**

Adresse eines 8-Byte-Bereichs, der den Programmnamen aus dem Parameter PROGRAM enthält.

**PC\_ADDR2**

Adresse der Daten des Kommunikationsbereichs (COMMAREA).

**PC\_ADDR3**

Adresse eines 2-Byte-Bereichs, der die Länge des Kommunikationsbereichs (COMMAREA) als Halbwort-Binärwert enthält.

**PC\_ADDR4**

Adresse der Daten aus INPUTMSG.

**PC\_ADDR5**

Adresse eines 2-Byte-Bereichs, der die Länge von INPUTMSG als Halbwort-Binärwert enthält.

**PC\_ADDR6**

Adresse eines 2-Byte-Bereichs, der die im Parameter DATALENGTH angegebene Länge enthält und definiert, wie viele Daten aus dem Kommunikationsbereich (COMMAREA) gesendet werden sollen. Die Länge wird als Halbwort-Binärwert gespeichert.

**PC\_ADDR7**

Adresse des 4-Byte-Namens des fernen Systems, an das die LINK-Anforderung übertragen werden soll (wie im Parameter SYSID angegeben).

**PC\_ADDR8**

Adresse des 4-Byte-Namens der Spiegeltransaktion, die auf dem fernen System angehängt werden soll (wie im Parameter TRANSID angegeben).

**PC\_ADDR9**

Wird nicht verwendet.

**PC\_ADDRA**

Adresse des 16-Byte-Kanalnamens (wie im Parameter CHANNEL angegeben).

**Ändern von Feldern in der Befehlsparameterstruktur**

Von den Feldern, die an die Programmsteuerung übergeben werden, werden einige als Eingabefelder für die Anforderung, andere als Ausgabefelder und wieder andere sowohl als Eingabe- als auch als Ausgabefelder verwendet. Die Methode, die Ihr Benutzerexitprogramm zum Ändern eines Felds verwendet, hängt von der Verwendung des betreffenden Felds ab.

## Ändern von Eingabefeldern:

Die richtige Methode zum Ändern eines Eingabefelds besteht darin, eine neue Kopie desselben zu erstellen und die Adresse in der Befehlsparameterliste so zu ändern, dass sie auf die neuen Daten verweist.

**Anmerkung:** Sie dürfen ein Eingabefeld niemals ändern, indem Sie die Daten ändern, auf die von der Befehlsparameterliste verwiesen wird. Dies würde zu einer Beschädigung von Speicher führen, der zum Anwendungsprogramm gehört; dies wiederum könnte zu einem Fehler führen, wenn das Programm versucht, das Feld wiederzuverwenden.

**Ändern von Ausgabefeldern:** Das unter „Ändern von Eingabefeldern“ beschriebene Verfahren ist für das Ändern von Ausgabefeldern nicht geeignet. (Die Ergebnisse würden an den neuen Bereich und nicht an den Bereich der Anwendung zurückgegeben und wären daher für die Anwendung nicht sichtbar.)

Ein Ausgabefeld wird geändert, indem die Daten geändert werden, auf die von der Parameterliste der Befehlsebene verwiesen wird. Bei einem Ausgabefeld können Sie die vorhandenen Anwendungsdaten ändern, da die Anwendung ohnehin erwartet, dass das Feld geändert wird.

## Ändern des EXEC-Schnittstellendesktors (EID)

Es ist nicht möglich, den EXEC-Schnittstellendesktor (EXEC Interface Descriptor, EID) zu ändern, um bedeutende Änderungen an Anforderungen vorzunehmen. So ist es beispielsweise nicht möglich, eine Anforderung vom Typ LINK in eine Programmsteueranforderung eines anderen Typs zu ändern. Es können jedoch geringfügige Änderungen an Anforderungen vorgenommen werden. So ist es beispielsweise möglich, das Existenzbit für die System-ID (SYSID) zu aktivieren, sodass die betreffende Anforderung in eine Anforderung geändert werden kann, die an ein fernes System übertragen wird.

In der nachstehenden Liste werden diejenigen Bits im EID aufgeführt, die geändert werden können. Versuche, andere Bestandteile des EID zu ändern, werden ignoriert.

### PC\_BITS1

- X'40' Existenzbit für COMMAREA.
- X'20' Existenzbit für LENGTH.
- X'10' Existenzbit für INPUTMSG.
- X'08' Existenzbit für INPUTMSGLEN.
- X'04' Existenzbit für DATALENGTH.
- X'02' Existenzbit für SYSID.
- X'01' Existenzbit für TRANSID.

### PC\_BITS2

- X'40' Existenzbit für CHANNEL.

### PC\_EIDOPT5

Wird für PC-Verknüpfungsanforderungen nicht verwendet.

Bits im EID sollten an Ort und Stelle geändert werden. Es sollte nicht der Verweis auf den EID geändert werden. (Jeder Versuch, dies zu tun, wird von CICS ignoriert.)

Der EID wird vor der Rückkehr zum Anwendungsprogramm auf seinen ursprünglichen Wert zurückgesetzt. Dies bedeutet, dass Änderungen, die am EID vorgenommen werden, nur für die Dauer der Programmsteueranforderung beibehalten werden.

Das Benutzerexitprogramm wird daran gehindert, bedeutende Änderungen am EID vorzunehmen.

### **Verwendung des Tokens für Programmsteueranforderungen, UEPPCTOK**

UEPPCTOK stellt die Adresse eines 4-Byte-Bereichs bereit, mit dem Sie für dieselbe Programmsteueranforderung Informationen zwischen den Benutzerexits XPCREQ und XPCREQC übergeben können.

So kann beispielsweise die Adresse eines Speicherteils, der vom Benutzerexit XPCREQ abgerufen wird und vom Benutzerexit XPCREQC freigegeben werden muss, im Feld UEPPCTOK übergeben werden.

Verwenden Sie den von UEPPCTOK adressierten Bereich nicht, um dort Zeichendaten direkt zu speichern. Der von UEPPCTOK adressierte 4-Byte-Bereich muss immer die Adresse eines Speicherbereichs enthalten, der die Daten enthält, die Sie übergeben wollen, und eine beliebige Signalmarkierung, um den Speicher zu identifizieren. CICS IA verwendet UEPPCTOK, um die Adresse des eigenen Speicherbereichs zu speichern. Wenn der globale CICS IA-Benutzerexit ausgeführt wird, überprüft er die Adresse in UEPPCTOK, um alle Speicherbereiche zu identifizieren, die bereits für den Exit vorhanden sind.

Das Token UEPPCTOK kann nur für die Dauer einer einzelnen Programmsteueranforderung verwendet werden, da sein Inhalt am Ende der Anforderung unter Umständen gelöscht werden kann. Wenn Sie Informationen zwischen aufeinanderfolgenden Aufrufen eines globalen Benutzerexits übergeben müssen, können Sie zu diesem Zweck das Task-Token UEPTSTOK verwenden. Weitere Informationen zu UEPTSTOK finden Sie unter Using the task token UEPTSTOK.

### **Der EXEC-Schnittstellenblock (EIB)**

Es werden Kopien von EIBRSRCE, EIBRCODE, EIBRESP und EIBRESP2 an den Exit übergeben, damit Sie folgende Aktionen ausführen können:

- Ändern oder Festlegen von Ausführungs- oder Ressourceninformationen in XPCREQ und XPCREQC.
- Untersuchen von Ausführungsinformationen in XPCREQC.

Sie können die Kopien von EIBRSRCE, EIBRCODE, EIBRESP und EIBRESP2, die Sie in der Parameterliste erhalten, aktualisieren. Die Programmsteuerung kopiert Ihre Werte in den realen EIB, nachdem der Exit XPCREQC ausgeführt wurde oder wenn Sie den Rückgabecode für Umgehen in XPCREQ angeben.

Sie müssen gültige Antworten für die Programmsteuerung festlegen. Sie müssen für die drei Codes EIBRCODE, EIBRESP und EIBRESP2 eine konsistente Gruppe von Werten festlegen, so wie sie auch von der Programmsteuerung festgelegt würden, um eine gültige Ausführung zu beschreiben. **Die Konsistenz der Werte für EIBRCODE, EIBRESP und EIBRESP2 wird von der Programmsteuerung nicht überprüft.** Als Hilfestellung beim Festlegen von Werten für EIBRCODE, EIBRESP und EIBRESP2 können Sie die im Pseudobereich (DSECT) DFHPCEDS angegebenen Werte heranziehen, die von der Programmsteuerung verwendet werden.

## Beispiel für die Verwendung von XPCREQ und XPCREQC

In diesem Beispiel werden XPCREQ und XPCREQC verwendet, um LINK-Anforderungen an eine Reihe von verschiedenen CICS-Regionen weiterzuleiten, um einen einfachen Lastausgleichsmechanismus bereitzustellen.

Das Beispiel veranschaulicht lediglich das Leistungsspektrum der Exits; es ist nicht beabsichtigt aufzuzeigen, wie die Lastausgleichsfunktion bestmöglich erreicht werden kann. Für den Zweck dieses Beispiels wird vorausgesetzt, dass bereits ein globaler Arbeitsbereich (GWA) vorhanden ist und dass dieser eine Liste der verfügbaren System-IDs (SYSIDs) enthält, zusammen mit der Anzahl der LINK-Anforderungen, die momentan von den einzelnen SYSIDs verarbeitet werden.

### Verarbeitungsschritte in XPCREQ:

1. Durchsuchen Sie den globalen Arbeitsbereich (Global Work Area, GWA) nach einer geeigneten CICS-Region (beispielsweise nach der Region, die momentan die wenigsten LINK-Anforderungen verarbeitet).
2. Nachdem Sie entschieden haben, an welches System die Anforderung weitergeleitet werden soll, erhöhen Sie den Nutzungszähler für dieses System.
3. Fordern Sie einen 4-Byte-Bereich an, in dem die SYSID für diese Anforderung gespeichert werden soll (dieser Bereich kann aus dem GWA zugeordnet werden, um die Ausgabe einer GETMAIN-Anforderung zu vermeiden). Wird der Bereich durch Ausgabe einer GETMAIN-Anforderung abgerufen, setzen Sie UEPPCTOK auf die Adresse des angeforderten Speichers.
4. Setzen Sie PC\_ADDR7 auf die Adresse des 4-Byte-Bereichs.
5. Wenn das Festlegen von PC\_ADDR7 dazu führt, dass es sich nun um die letzte Adresse handelt, setzen Sie das höchstwertige Bit in dieser Adresse und setzen Sie das höchstwertige Bit in der Adresse, die zuvor die letzte Adresse war, entsprechend zurück.
6. Aktivieren Sie das Existenzbit X'02' in PC\_BITS1, um anzuzeigen, dass eine System-ID (SYSID) angegeben worden ist.
7. Kehren Sie zu CICS zurück.

### Verarbeitungsschritte in XPCREQC:

1. Durchsuchen Sie den globalen Arbeitsbereich (Global Work Area, GWA) nach dem Eintrag für die CICS-Region, die im Parameter SYSID angegeben ist.
2. Verringern Sie den Nutzungszähler für dieses System.
3. Wurde in XPCREQ eine GETMAIN-Anforderung ausgegeben, um einen Bereich zum Speichern der SYSID abzurufen, geben Sie einen Aufruf vom Typ FREE-MAIN für die in UEPPCTOK enthaltene Adresse aus.
4. Kehren Sie zu CICS zurück.

## Exit XPCFTCH

XPCFTCH wird aufgerufen, bevor ein Programm, das für CICS definiert ist (einschließlich interner CICS-Module), die Steuerung erhält. Die Übergabe der Steuerung kann darauf zurückzuführen zu sein, dass es sich um das erste Programm in einer Transaktion handelt, oder auf eine Anforderung vom Typ LINK, XCTL oder HANDLE ABEND PROGRAM.

Sie können diesen Exit verwenden, um die Eingangsadresse zu ändern, die beim Herstellen einer Verbindung mit dem Programm verwendet wird. Wenn der Exit den Rückgabecode null oder eine geänderte Adresse mit null setzt, wird die Eingangsadresse des ursprünglichen Anwendungsprogramms verwendet.

Mithilfe dieses Exits wird die Steuerung an ein(e) Assembleranwendungsprogramm oder -routine vom Typ AMODE(64), AMODE(31) oder AMODE(24) übergeben, bevor das ursprüngliche Programm aufgerufen wird. Nachdem das Assemblerprogramm seine Verarbeitung beendet hat, sollte es die Steuerung an den Einstiegspunkt des ursprünglichen Programms zurückgeben, indem es eine Verzweigungsanweisung verwendet. Verwenden Sie den Exit ausschließlich zum Aufrufen des ursprünglichen Programms und nicht zum Aufrufen eines anderen Programms, da die Ergebnisse sonst unvorhersehbar sind.

Wenn XPCFTCH für ein C- oder C++-Programm aufgerufen wird, das mit der Option XPLINK kompiliert ist, wird ein Flag so gesetzt, dass jedwede Änderung der Einstiegspunktadresse ignoriert wird, die der Benutzerexit eventuell vornimmt.

Wird eine geänderte Eingangsadresse angegeben, erhält das aufgerufene Programm die Steuerung in dem Ausführungsschlüssel, in dem auch das ursprüngliche Anwendungsprogramm die Steuerung erhalten hätte; das heißt, wie in der Option EXECKEY der Ressourcendefinition des ursprünglichen Programms angegeben.

## **Zeitpunkt des Aufrufs**

Bevor ein Anwendungsprogramm die Steuerung erhält.

## **Exitspezifische Parameter**

### **UEPPCDS**

Adresse eines Speicherbereichs, der programm- und terminalbezogene Informationen enthält und mithilfe des Pseudobereichs (DSECT) DFHPCUE zugeordnet werden kann. Wenn XPCFTCH aufgerufen wird, sind die folgenden DFHPCUE-Felder von Bedeutung:

#### **PCUE\_CONTROL\_BITS**

- 1-Byte-Flagfeld. Die Einstellung PCUECBTE gibt an, dass die Transaktion mit einem Terminal verbunden ist.
- Die Einstellung PCUENOTX (X'40') gibt an, dass sich das Programm nicht auf der Befehlsebene befindet.
- Ein Flag (PCUE\_NO\_MODIFY) in PCUE\_CONTROL\_BITS gibt an, dass eine geänderte Eingangsadresse nicht unterstützt wird. Wenn dieses Flag gesetzt ist, wird der Rückgabecode UERCMEA von XPCFTCH stets ignoriert. CICS setzt dieses Flag, bevor XPCFTCH für C- und C++-Programme aufgerufen wird, die mit der Option XPLINK kompiliert sind.
- Die Einstellung PCUE\_REAL (X'20') gibt an, dass ein realer Einstiegspunkt in PCUE\_REAL\_ENTRY festgelegt ist.

#### **PCUE\_TASK\_NUMBER**

Gepacktes Dezimalfeld aus 3 Zeichen mit der Tasknummer.

#### **PCUE\_TRANSACTION\_ID**

Feld aus 4 Zeichen mit der ID der ursprünglichen Transaktion. Diese ID weicht unter Umständen von der ID der aktuellen Transaktion ab.

#### **PCUE\_TERMINAL\_ID**

Feld aus 4 Zeichen mit der Terminal-ID (sofern vorhanden).

#### **PCUE\_PROGRAM\_NAME**

Feld aus 8 Zeichen mit dem Namen des Programms, das die Steuerung erhalten soll.

**PCUE\_PROGRAM\_LANGUAGE**

Feld aus 3 Zeichen mit der Sprache des Programms, das die Steuerung erhalten soll.

**PCUE\_LOAD\_POINT**

Der Ladepunkt des Programms.

**PCUE\_ENTRY\_POINT**

Der Einstiegspunkt des Programms.

**PCUE\_AMOD**

Der Adressierungsmodus des Programms ist AMODE(31). Dieses Feld wird aus Gründen der Kompatibilität mit vorhandenen Exitprogrammen bereitgestellt.

**PCUE\_AMOD\_31**

Der Adressierungsmodus des Programms ist AMODE(31). Verwenden Sie vorzugsweise dieses Feld anstelle von PCUE\_AMOD.

**PCUE\_AMOD\_64**

Der Adressierungsmodus des Programms ist AMODE(64).

**PCUE\_PROGRAM\_SIZE**

Vollwort mit der Größe des Programms (in Byte).

**PCUE\_COMMAREA\_ADDRESS**

Adresse des Kommunikationsbereichs des Programms, sofern das Programm über einen solchen Bereich verfügt.

**PCUE\_COMMAREA\_SIZE**

Vollwort mit der Länge des Kommunikationsbereichs des Programms, sofern das Programm über einen solchen Bereich verfügt.

**PCUE\_LOGICAL\_LEVEL**

Vollwort mit der logischen Programmebene.

**PCUE\_BRANCH\_ADDRESS**

Vollwort. Verwenden Sie dieses Feld, um eine alternative Eingangsadresse anzugeben. Legen Sie anhand des oberen Bit fest, dass das alternative Programm den Adressierungsmodus AMODE(31) ausführen soll.

**PCUE\_REAL\_ENTRY**

Ab z/OS 1.7 gibt dieses Feld den realen Einstiegspunkt für Programme an, die mit Language Environment konform sind. Zuvor stand Ihnen nur PCUE\_ENTRY\_POINT zur Verfügung; bei Programmen, die mit Language Environment konform sind, enthielt dieses Feld jedoch nicht den Einstiegspunkt, den Sie kennen mussten.

**Anmerkung:** Mit z/OS 1.7 stellt dieses Feld eine Lösung für das Problem bereit, das durch APAR PQ43992 gemeldet wurde.

**PCUE\_CHANNEL\_NAME**

Adresse eines 16-Byte-Felds mit dem Namen des Kanals, mit dem das Anwendungsprogramm aufgerufen werden soll. Hierbei handelt es sich um den aktuellen Kanal des Programms. Ist kein Kanal vorhanden, wird dieses Feld auf Leerzeichen gesetzt.

**PCUE\_INVOKING\_PROGRAM\_NAME**

Feld aus 8 Zeichen mit dem Namen des Programms, das das aktuelle Programm aufgerufen hat.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

### UERCPURG

Task während des XPI-Aufrufs bereinigt.

### UERCMEA

Eingangsadresse wurde geändert.

## XPI-Aufrufe

Es können alle verwendet werden.

## Beispielexitprogramm

DFH\$PCEX

## Exit XPCHAIR

Der Exit XPCHAIR wird aufgerufen, bevor eine Routine vom Typ HANDLE ABEND LABEL die Steuerung erhält.

Diese Situation tritt nur auf, wenn ein Programmabbruch eine Verzweigung zu einer internen Abbruchroutine verursacht. Wenn in einer Anforderung vom Typ HANDLE ABEND die Option PROGRAM angegeben ist, wird der Exit XPCFTCH aufgerufen (siehe „Exit XPCFTCH“ auf Seite 218). Mithilfe des Exits XPCHAIR können Sie eine alternative Adresse für die Abbruchverarbeitung (HANDLE ABEND) angeben. Wenn der Exit den Rückgabecode null oder eine alternative Adresse mit null setzt, übergibt CICS die Steuerung an die angegebene interne Routine des Anwendungsprogramms.

Dieser Exit wird niemals für ein C- oder C++-Programm aufgerufen, das mit der Option XPLINK kompiliert ist, und auch niemals für ein AMODE(64)-Programm.

Bei Angabe einer geänderten Eintragsadresse gilt Folgendes:

- Der aufgerufene Code erhält die Steuerung in dem Ausführungsschlüssel, in dem die interne Abbruchroutine die Steuerung erhalten hätte, also in dem Schlüssel, der in Kraft war, als der Befehl EXEC CICS HANDLE ABEND LABEL ausgegeben wurde.
- Die Adresse für Fortsetzung (RESUME) wird für COBOL-Anwendungen in Register 14 und für Assembleranwendungen in Register 15 gestellt. Wird Ihre Anwendung in einer heterogenen Umgebung ausgeführt, muss Ihr Exitprogramm möglicherweise ein eigenes Basisadressregister einrichten. So könnten Sie beispielsweise den folgenden Code verwenden, um die Adressierbarkeit einzurichten:

```
BASSM 15,0  
USING *,15
```

### Zeitpunkt des Aufrufs

Bevor eine Routine vom Typ HANDLE ABEND die Steuerung erhält.

### Exitspezifische Parameter

#### UEPPCDS

Adresse eines Speicherbereichs, der programm- und terminalbezogene Informationen enthält und mithilfe des Pseudobereichs

(DSECT) DFHPCUE zugeordnet werden kann. Wenn XPCHAIR aufgerufen wird, sind die folgenden DFHPCUE-Felder von Bedeutung:

**PCUE\_CONTROL\_BITS**

1-Byte-Flagfeld. Die Einstellung PCUECBTE gibt an, dass die Transaktion mit einem Terminal verbunden ist.

**PCUE\_TASK\_NUMBER**

Gepacktes Dezimalfeld aus 3 Zeichen mit der Tasknummer.

**PCUE\_TRANSACTION\_ID**

Feld aus 4 Zeichen mit der Transaktions-ID.

**PCUE\_TERMINAL\_ID**

Feld aus 4 Zeichen mit der Terminal-ID (sofern vorhanden).

**PCUE\_PROGRAM\_NAME**

Feld aus 8 Zeichen mit dem Namen des Programms, das den Befehl HANDLE ABEND LABEL ausgegeben hat.

**PCUE\_LOGICAL\_LEVEL**

Vollwort mit der logischen Programmebene.

**PCUE\_BRANCH\_ADDRESS**

Vollwort. Verwenden Sie dieses Feld, um die Adresse einer alternativen Abbruchroutine anzugeben. Legen Sie anhand des oberen Bit fest, dass die alternative Abbruchroutine den Adressierungsmodus AMODE(31) ausführen soll.

**UEPTACB**

Adresse des Transaktionsabbruchsteuerblocks (TACB) für den betreffenden Abbruch. Wenn der Abbruch infolge eines Programmfehlers auftrat, enthält der TACB folgende Informationen:

- Das Programmstatuswort (PSW).
- Die Register zum Zeitpunkt des Abbruchs.
- Die Details des Unterbereichs, der Zugriffsregister und der Vektorregister zum Zeitpunkt des Abbruchs.
- Das Register für Unterbrechungseignisadressen (Breaking Event Address Register, BEAR).
- Die Adresse der Umsetzungsausnahmebedingung (Translation Exception Address, TEA).

Sie können den TACB mithilfe des Makros DFHTACB TYPE=DSECT zuordnen.

**Rückgabecodes**

**UERCNORM**

Verarbeitung fortsetzen.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**UERCMEA**

Adresse einer alternativen Abbruchroutine wird angegeben.

**XPI-Aufrufe**

Es können alle verwendet werden.



## Exit XPCTA

Der Exit XPCTA wird unmittelbar nach dem Abbruch einer Transaktion aufgerufen sowie vor jeder Art von Verarbeitung, bei der die vorhandene Umgebung so verändert werden könnte, dass sich die Task nicht fortsetzen ließe.

Sie können den Exit XPCTA verwenden, um folgende Aktionen auszuführen:

- Festlegen einer Adresse für Fortsetzung (RESUME), anstatt zuzulassen, dass CICS den Abbruch verarbeitet.
- Angabe des Unterbereichs, in dem die Steuerung übergeben wird.

Wenn eine Adresse für Fortsetzung zurückgegeben wird, dann werden die Register 0 bis 13 und 15 mit den Werten wiederhergestellt, die zum Zeitpunkt des Abbruchs vorlagen. Register 14 wird für die Verzweigung zur Adresse für Fortsetzung verwendet. Wenn der Exit den Rückgabecode null oder eine Adresse für Fortsetzung mit null setzt, wird der Abbruch von CICS verarbeitet.

Wenn der Transaktionsabbruch infolge eines Programmfehlers oder Betriebssystemabbruchs auftritt, wird der Exit XDUREQ der Speicherauszugsdomäne möglicherweise vor XPCTA aufgerufen (siehe „Exit XDUREQ“ auf Seite 40). Ebenso gilt: Wenn eine Adresse für Fortsetzung zurückgegeben wird, dann werden die Register 0 bis 15 mit den Werten wiederhergestellt, die zum Zeitpunkt des Abbruchs vorlagen. Das Programmstatuswort (PSW) wird für die Verzweigung zur Adresse für Fortsetzung verwendet.

In bestimmten Situationen setzt CICS ein Flag, um die Adresse für Fortsetzung zu ignorieren, die normalerweise aus dem Rückgabecode UERCMEA abgerufen wird.

- In den folgenden Situationen setzt CICS das Flag PCUE\_NO\_RESUME, um eventuell vom Exit bereitgestellte Adressen für Fortsetzung zu ignorieren:
  - XPCFTCH wird für ein C- oder C++-Programm aufgerufen, das mit der Option XPLINK kompiliert ist.
  - Der Tasksteuerblock (Task Control Block, TCB) der Anwendung ist nicht mehr verfügbar.
  - Bei dem Transaktionsabbruch handelt es sich um einen Abbruch vom Typ 'AKxx' (im Zusammenhang mit einer Beendigungsanforderung), aber nicht um den Typ AKKD oder AKKE.
- Bei einem AMODE(64)-Programm setzt CICS das Flag PCUE\_NO\_RESUME\_AMODE64, um eventuell vom Exit bereitgestellte Adressen für Fortsetzung zu ignorieren, wenn die 64-Bit-Registers zum Zeitpunkt des Abbruchs nicht verfügbar sind.

### Zeitpunkt des Aufrufs

Nach dem Auftreten eines Abbruchs und vor dem Ändern der Umgebung.

### Exitspezifische Parameter

#### UEPPCDS

Adresse eines Speicherbereichs, der programm- und terminalbezogene Informationen enthält und mithilfe des Pseudobereichs (DSECT) DFHPCUE zugeordnet werden kann. Wenn XPCTA aufgerufen wird, sind die folgenden DFHPCUE-Felder von Bedeutung:

#### PCUE\_CONTROL\_BITS

1-Byte-Flagfeld. Die Einstellung PCUECBTE gibt an, dass die Transaktion mit einem Terminal verbunden ist.

Die Flags PCUE\_NO\_RESUME und PCUE\_NO\_RESUME\_AMODE64 in PCUE\_CONTROL\_BITS geben an, dass kei-

ne Adressen für Fortsetzung unterstützt werden. Wenn diese Flags gesetzt sind, wird der Rückgabecode UERCMEA von XPCTA stets ignoriert. In den folgenden Situationen setzt CICS das Flag PCUE\_NO\_RESUME vor dem Aufruf von XPCTA: für C- und C++-Programme, die mit der Option XPLINK kompiliert sind; wenn der TCB der Anwendung nicht mehr verfügbar ist; und bei Abbrüchen vom Typ 'AKxx', bei denen es sich nicht um AKKD oder AKKE handelt. CICS setzt das Flag PCUE\_NO\_RESUME\_AMODE64 für AMODE(64)-Programme, wenn keine Adressen für Fortsetzung unterstützt werden.

**PCUE\_TASK\_NUMBER**

Gepacktes Dezimalfeld aus 3 Zeichen mit der Tasknummer.

**PCUE\_TRANSACTION\_ID**

Feld aus 4 Zeichen mit der Transaktions-ID.

**PCUE\_TERMINAL\_ID**

Feld aus 4 Zeichen mit der Terminal-ID (sofern vorhanden).

**PCUE\_PROGRAM\_NAME**

Feld aus 8 Zeichen mit dem Namen des fehlerhaften Programms.

**PCUE\_LOGICAL\_LEVEL**

Vollwort mit der logischen Programmebene.

**PCUE\_BRANCH\_ADDRESS**

Vollwort. Sie können dieses Feld verwenden, um eine Adresse für Fortsetzung anzugeben. Legen Sie anhand des oberen Bit fest, dass die fortgesetzte Task den Adressierungsmodus AMODE(31) ausführen soll. Legen Sie anhand des unteren Bit fest, dass die fortgesetzte Task den Adressierungsmodus AMODE(64) ausführen soll.

**PCUE\_BRANCH\_EXECKEY**

Wenn der Speicherschutz aktiv ist, können Sie dieses 1-Byte-Feld verwenden, um den Ausführungsschlüssel der fortgesetzten Task anzugeben. Mögliche Werte:

**PCUE\_BRANCH\_USER**

Benutzerschlüssel.

**PCUE\_BRANCH\_CICS**

CICS-Schlüssel.

Wenn der Speicherschutz aktiv ist und Sie keinen Wert angeben, wird die fortgesetzte Task im Benutzerschlüssel ausgeführt.

Wenn der Speicherschutz nicht aktiv ist, wird die fortgesetzte Task im CICS-Schlüssel ausgeführt.

**UEPTACB**

Adresse des Transaktionsabbruchsteuerblocks (TACB) für den betreffenden Abbruch. Wenn der Abbruch infolge eines Programmfehlers auftrat, enthält der TACB folgende Informationen:

- Das Programmstatuswort (PSW).
- Die Register zum Zeitpunkt des Abbruchs.

- Die Details des Unterbereichs, der Zugriffsregister und der Vektorregister zum Zeitpunkt des Abbruchs.
- Das Register für Unterbrechungsereignisadressen (Breaking Event Address Register, BEAR).
- Die Adresse der Umsetzungsausnahmebedingung (Translation Exception Address, TEA).

Sie können den TACB mithilfe des Makros DFHTACB TYPE=DSECT zuordnen.

### **Rückgabecodes**

#### **UERCNORM**

Verarbeitung fortsetzen.

#### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

#### **UERCMEA**

Eine Adresse für Fortsetzung wird angegeben.

### **XPI-Aufrufe**

Es können alle verwendet werden.

## **Beispielexitprogramm für XPCTA**

DFH\$PCTA ist ein Beispiel eines globalen Benutzerexitprogramms für den Exitpunkt XPCTA. Mithilfe dieses Beispiels wird überprüft, ob der Abbruch durch eine Speicherschutz Ausnahmebedingung verursacht wurde. Weitere Informationen zu DFH\$PCTA finden Sie unter Transaction Abend sample exit program: DFH\$PCTA.

## **Exit XPCABND**

Der Exit XPCABND wird nach einem Transaktionsabbruch und vor einem Transaktionspeicherauszugsaufruf aufgerufen. Sie können diesen Exit verwenden, um den Speicherauszug zu unterdrücken.

### **Zeitpunkt des Aufrufs**

Nach dem Abbruch einer Transaktion und vor Absetzen eines Transaktionspeicherauszugsaufrufs.

### **Exitspezifische Parameter**

#### **UEPPCDS**

Adresse eines Speicherbereichs, der programmbezogene und terminalbezogene Informationen enthält. Der Speicherbereich wird durch den Pseudobereich (DSECT) DFHPCUE zugeordnet.

Wenn XPCABND aufgerufen wird, sind die folgenden DFHPCUE-Felder von Bedeutung:

#### **PCUE\_CONTROL\_BITS**

1-Byte-Flagfeld. Die Einstellung PCUECBTE gibt an, dass die Transaktion mit einem Terminal verbunden ist.

#### **PCUE\_TASK\_NUMBER**

Gepacktes Dezimalfeld aus 3 Zeichen mit der Tasknummer.

#### **PCUE\_TRANSACTION\_ID**

Feld aus 4 Zeichen mit der Transaktions-ID.

**PCUE\_TERMINAL\_ID**

Feld aus 4 Zeichen mit der Terminal-ID (sofern vorhanden).

**PCUE\_PROGRAM\_NAME**

Feld aus 8 Zeichen mit dem Namen des Programms, das abgebrochen wird.

**PCUE\_LOGICAL\_LEVEL**

Vollwort mit der logischen Programmebene.

**UEPTACB**

Adresse des Transaktionsabbruchsteuerblocks (TACB) für den betreffenden Abbruch. Wenn der Abbruch infolge eines Programmfehlers auftrat, enthält der TACB folgende Informationen:

- Das Programmstatuswort (PSW).
- Die Register zum Zeitpunkt des Abbruchs.
- Die Details des Unterbereichs, der Zugriffsregister und der Vektorregister zum Zeitpunkt des Abbruchs.
- Das Register für Unterbrechungsereignisadressen (Breaking Event Address Register, BEAR).
- Die Adresse der Umsetzungsausnahmebedingung (Translation Exception Address, TEA).

Sie können den TACB mithilfe des Makros DFHTACB TYPE=DSECT zuordnen.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen und Speicherauszugsaufruf absetzen.

**UERCBYB**

Speicherauszugsaufruf unterdrücken.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

---

## **Exits für das Ressourcenmanagerschnittstellenprogramm (XRMIIN, XRMIOU)**

Diese Exits werden aufgerufen, wenn API-Anforderungen der Ressourcenmanagerschnittstelle (RMI) verarbeitet werden.

### **Exit XRMIIN**

Der Exit XRMIIN wird aufgerufen, bevor ein taskbezogener Benutzerexitprogramm aufgerufen wird, wenn ein Anwendungsprogramm eine RMI-API-Anforderung ausgibt.

**Exitspezifische Parameter****UEPTRUEN**

Adresse des Namens des taskbezogenen Benutzerexitprogramms.

#### **UEPTRUEP**

Adresse der Parameterliste, die an das taskbezogene Benutzerexitprogramm übergeben werden soll.

#### **UEP\_RM\_PBTOK**

Adresse eines 4-Byte-Felds mit dem Leistungsbloktoken (Performance Block Token, PBTOK) von z/OS Workload Manager (WLM). Mithilfe dieses Tokens kann ein Exitprogramm auf Informationen (wie beispielsweise das Serviceklassetoken SERVCLS) im WLM-Leistungsblock zugreifen. Hierzu muss das Programm das WLM EXTRACT-Makro IWMMECTR verwenden und dabei das Leistungsbloktoken als Eingabeparameter MONTKN übergeben. Weitere Informationen zum Makro IWMMECTR finden Sie unter z/OS MVS Programming: Workload Management Services.

Exitprogramme dürfen nicht versuchen, den Leistungsblock zu ändern, da die Ergebnisse eines solchen Versuchs unvorhersehbar sind.

#### **UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

**Anmerkung:** Die Parameterliste des taskbezogenen Benutzerexitprogramms wird durch den Pseudobereich (DSECT) DFHUEPAR zugeordnet, der allgemeine Feldnamen mit der Parameterliste für DFHUEPAR des globalen Benutzerexitprogramms gemeinsam nutzt. Um beide DSECT-Definitionen in Ihr Exitprogramm aufzunehmen, müssen Sie den folgenden Code verwenden:

```
DFHUEXIT TYPE=EP,ID=XRMIIN  
DFHUEXIT TYPE,TYPE=RM
```

Die Anweisungen müssen in der angegebenen Reihenfolge codiert werden.

Die beiden Parameterlisten für DFHUEPAR (die des globalen Benutzerexits und die des taskbezogenen Benutzerexits) belegen separate Speicherbereiche. Die Parameterliste der taskbezogenen Benutzerexits wird nur zur Information bereitgestellt; Sie sollten diese Liste in keiner Weise ändern.

#### **Rückgabecodes**

##### **UERCNORM**

Verarbeitung fortsetzen.

##### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

#### **XPI-Aufrufe**

Es können alle verwendet werden.

#### **API- und SPI-Befehle**

Es können alle API- und SPI-Befehle mit Ausnahme von EXEC CICS SHUTDOWN und EXEC CICS XCTL verwendet werden. Befehle vom Typ CALLDLI, EXEC DLI und EXEC SQL dürfen hingegen **nicht** verwendet werden.

## Exit XRMIOUT

Der Exit XRMIOUT wird aufgerufen, nachdem ein taskbezogenes Benutzerexitprogramm von der Verarbeitung einer RMI-API-Anforderung zurückgekehrt ist.

### Exitspezifische Parameter

#### UEPTRUEN

Adresse des Namens des taskbezogenen Benutzerexitprogramms.

#### UEPTRUEP

Adresse der Parameterliste, die an das taskbezogene Benutzerexitprogramm übergeben werden soll.

Der Parameter UEPHMSA in dieser Parameterliste enthält den Registersicherungsbereich (RSA) des Aufrufenden. Bei einem EXEC CPSM-Aufruf enthält Register 1 in diesem RSA die Adresse der Parameterliste für den CICSPlex SM-Befehl und kann verwendet werden, um den CICSPlex SM-Befehl und die angegebene Ressource zu identifizieren. Ausführliche Informationen hierzu finden Sie unter CICSPlex SM API command argument list.

#### UEP\_RM\_PBTOK

Adresse eines 4-Byte-Felds mit dem Leistungsblocktoken (Performance Block Token, PBTOK) von z/OS Workload Manager (WLM). Mithilfe dieses Tokens kann ein Exitprogramm auf Informationen (wie beispielsweise das Serviceklassetoken SERVCLS) im WLM-Leistungsblock zugreifen. Hierzu muss das Programm das WLM EXTRACT-Makro IWMMECTR verwenden und dabei das Leistungsblocktoken als Eingabeparameter MONTKN übergeben. Weitere Informationen zum Makro IWMMECTR finden Sie unter z/OS MVS Programming: Workload Management Services.

Exitprogramme dürfen nicht versuchen, den Leistungsblock zu ändern, da die Ergebnisse eines solchen Versuchs unvorhersehbar sind.

#### UEPRECUR

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

**Anmerkung:** Die Parameterliste des taskbezogenen Benutzerexitprogramms wird durch den Pseudobereich (DSECT) DFHUEPAR zugeordnet, der allgemeine Feldnamen mit der Parameterliste für DFHUEPAR des globalen Benutzerexitprogramms gemeinsam nutzt. Um beide DSECT-Definitionen in Ihr Exitprogramm aufzunehmen, müssen Sie den folgenden Code verwenden:

```
DFHUEXIT TYPE=EP, ID=XRMIOUT
DFHUEXIT TYPE, TYPE=RM
```

Die Anweisungen müssen in der angegebenen Reihenfolge codiert werden.

Die Parameterliste für DFHUEPAR des globalen Benutzerexits und die Parameterliste für DFHUEPAR des taskbezogenen Benutzerexits belegen jeweils einen separaten Speicherbereich. Die Parameterliste des taskbezogenen Benutzerexits wird nur zu Informationszwecken bereitgestellt und darf auf keinen Fall geändert werden.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

### UERPURG

Task während des XPI-Aufrufs bereinigt.

## XPI-Aufrufe

Es können alle verwendet werden.

## API- und SPI-Befehle

Es können alle API- und SPI-Befehle mit Ausnahme von EXEC CICS SHUTDOWN und EXEC CICS XCTL verwendet werden. Befehle vom Typ CALLDLI, EXEC DLI und EXEC SQL dürfen hingegen **nicht** verwendet werden.

**Anmerkung:** Es ist nicht empfehlenswert, das Exitprogramm Aufrufe an andere externe Ressourcenmanager ausgeben zu lassen, die die RMI verwenden, da dies zu einer Rekursion und somit möglicherweise zu einer Schleife führt. Es liegt in der Verantwortung des Exitprogramms, den Eintritt in eine Schleife zu vermeiden. Sie können das Rekursionszählerfeld UEPRECUR im Exitprogramm verwenden, um sich gegen dieses mögliche Problem zu schützen.

---

## Exit XRSINDI für das Ressourcenmanagement beim Installieren und Verwerfen

Der globale Benutzerexit XRSINDI (sofern aktiviert) wird unmittelbar nach dem erfolgreichen Installieren oder Verwerfen einer Ressourcendefinition durch CICS gesteuert.

Der Exit wird bei folgenden Aktivitäten im Bereich Installieren und Verwerfen gesteuert:

- Installationsfunktion der Gruppenliste bei einem Erststart oder Kaltstart von CICS.
- Befehl **CEDA INSTALL**.
- CICSplex SM BAS-Befehl **INSTALL**.
- Alle folgenden Operationen für automatische Installation:
  - Automatische Installation eines Terminals, einer Verbindung, eines Programms, einer Maskengruppe, einer Partitionsgruppe oder eines Journals.
  - Automatisches Verwerfen eines nicht verwendeten Terminals, das vom Systeminitialisierungsparameter AILDELAY und dem Attribut SIGNOFF in der TY-PETERM-Ressourcendefinition gesteuert wird.
- Aufbau und Abbau einer Verbindung zu einem MVS-Protokolldatenstrom.
- Befehl **CEMT DISCARD** und **EXEC CICS DISCARD**.
- Befehl **EXEC CICS CREATE**.
- Operationen zum Installieren und Verwerfen von FEPI (Front End Programming Interface): Befehl **EXEC CICS FEPI INSTALL** und **EXEC CICS FEPI DISCARD**.

Die Parameterliste ist so konzipiert, dass die Namen mehrerer (installierter oder verworfener) Ressourcen im Feld UEPIDNAM übergeben werden. Bei der Gestaltung Ihres globalen Benutzerexitprogramms dürfen Sie nicht davon ausgehen, dass die Anzahl der übergebenen Ressourcennamen immer eins ist. Analysieren Sie die Ressourcen in einer Schleife, basierend auf dem Wert, auf den UEPIDNUM verweist.

Den Namen der Modusgruppen wird der entsprechende Verbindungsname als Präfix vorangestellt. Zwischen den beiden Namen gibt es kein Trennzeichen: die ersten 4 Zeichen bilden den Verbindungsnamen, gefolgt von 8 Zeichen für die Modusgruppe. Die Teile des verketteten Namens haben eine feste Länge; werden Verbindungsnamen mit weniger als 4 Zeichen definiert, werden diese Namen in den verketteten Namen mit Leerzeichen aufgefüllt. Ebenso gilt: Die Verbindungsnamen für eine FEPI-Verbindung (Front End Programming Interface) sind eine Verkettung aus einem FEPI-Knotennamen und einem FEPI-Zielnamen, wobei jeder einzelne Name jeweils 8 Zeichen lang ist (feste Länge) und es kein Trennzeichen gibt.

Der Exit wird für jede einzelne Ressource in einer Gruppenliste, die während eines Erststarts oder Kaltstarts von CICS installiert wird, jeweils einmal gesteuert. Wenn Sie Bedenken hinsichtlich der Leistung bei einem Erst- oder Kaltstart haben, sollten Sie den Exit erst nach der Installation der Gruppenliste aktivieren. Zum Abrufen von Informationen zu Ressourcen, die vor dem Aktivieren des Exits installiert wurden, können Sie ein Programm zum Durchsuchen der Tabellen installierter Ressourcen schreiben, indem Sie die Anzeigefunktion **EXEC CICS INQUIRE** *ressourcennamen* verwenden.

## Exit XRSINDI

Der globale Benutzerexit XRSINDI wird aufgerufen, wenn CICS eine Ressourcendefinition installiert oder verwirft.

### Exit XRSINDI - Parameter und Rückgabecodes

Abbrüche in einem Programm, das am Exitpunkt XRSINDI aktiviert ist, können unter Umständen dazu führen, dass CICS beendet wird, da der Exit für einige Ressourcen während der Synchronisationspunktverarbeitung gesteuert wird. Wenn der Exit während der Synchronisationspunktverarbeitung für diese Ressourcen den Code UERCPURG zurückgibt, wird der Abbruchcode AUEP erstellt und CICS wird beendet.

Die Parameter, Rückgabecodes und XPI-Informationen lauten wie folgt:

#### Exitspezifische Parameter

##### UEPTRANID

Adresse der 4-Byte-Transaktions-ID.

##### UEPUSER

Adresse der 8-Byte-Benutzer-ID.

##### UEPTERM

Adresse der 4-Byte-Terminal-ID.

##### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens.

##### UEPIDREQ

Adresse der 1-Byte-Kennung für Installation oder Verwerfung. Die Werte lauten wie folgt:

##### UEIDINS

Diese Anforderung erfolgt für eine Installationsaktion bzw. im Fall eines Protokolldatenstroms für eine Verbindung zu einem Protokolldatenstrom.



**UEIDDIS**

Diese Anforderung erfolgt für eine Verwerfungsaktion bzw. im Fall eines Protokolldatenstroms für einen Abbau der Verbindung zu einem Protokolldatenstrom.

**UEPIDTYP**

Adresse des 1-Byte-Ressourcentyps. Die Werte lauten wie folgt:

**UEIDATOM**

Eine Ressource vom Typ ATOMSERVICE.

**UEIDAITM**

Ein Terminalmodell für die automatische Installation.

**UEIDBNDL**

Eine Ressource vom Typ BUNDLE.

**UEIDCONN**

Eine Verbindung.

**UEIDDB2C**

Eine DB2CONN-Ressourcendefinition für die Verbindung zwischen CICS und Db2.

**UEIDDB2E**

Eine DB2ENTRY-Ressourcendefinition.

**UEIDDB2T**

Eine DB2TRAN-Ressourcendefinition.

**UEIDDOCT**

Eine Ressource vom Typ DOCTEMPLATE.

**UEIDEARB**

Eine EAR-Datei, die Teil eines CICS-Bundles ist.

**UEIDEBAB**

Eine EBA-Datei, die Teil eines CICS-Bundles ist.

**UEIDEPAD**

Eine Ressource vom Typ EPADAPTER.

**UEIDEPAS**

Eine Ressource vom Typ EPADAPTERSET.

**UEIDEVCS**

Eine Ereigniserfassungsressource.

**UEIDEVNT**

Eine Ressource vom Typ EVENTBINDING.

**UEIDFECO**

Eine FEPI-Verbindung.

**UEIDFENO**

Ein FEPI-Knoten.

**UEIDFEPO**

Ein FEPI-Pool.

**UEIDFEPS**

Eine FEPI-Eigenschaftengruppe.

**UEIDFETA**

Ein FEPI-Ziel.

**UEIDFILE**  
Eine Datei.

**UEIDIPCO**  
Eine Ressource vom Typ IPCONN.

**UEIDJNMD**  
Ein Journalmodell.

**UEIDJNNM**  
Ein Journalname.

**UEIDJSRV**  
Eine JVM-Serverressource.

**UEIDLBRY**  
Eine Ressource vom Typ LIBRARY.

**UEIDMAP**  
Eine Maskengruppe.

**UEIDMODE**  
Eine Modusgruppe.

**UEIDMPPP**  
Eine Richtlinienressource.

**UEIDMQCN**  
Eine MQCONN-Ressourcendefinition für die Verbindung zwischen CICS und IBM® MQ.

**UEIDMQIN**  
Eine Ressource vom Typ MQINI.

**UEIDMQMN**  
Eine Ressource vom Typ MQMONITOR.

**UEIDNAPP**  
Eine Ressource vom Typ NODEJSAPP.

**UEIDNQRN**  
Eine Ressource vom Typ ENQMODEL.

**UEIDOSGB**  
Ein OSGi-Bundle.

**UEIDPART**  
Ein Partner.

**UEIDPIPE**  
Eine Pipeline (PIPELINE).

**UEIDPKST**  
Eine Db2-Ressource vom Typ PACKAGESET.

**UEIDPROF**  
Ein Profil.

**UEIDPROG**  
Ein Programm.

**UEIDPRTY**  
Ein BTS-Prozesstyp.

**UEIDPSET**  
Eine Partitionsgruppe.

**UEIDSESS**

Eine Sitzung.

**UEIDSTRM**

Ein MVS-Protokolldatenstrom.

**UEIDTCLS**

Eine Transaktionsklasse.

**UEIDTCPS**

Ein TCP/IP-Service.

**UEIDTDQU**

Eine Warteschlange mit transienten Daten.

**UEIDTERM**

Ein Terminal.

**UEIDTRAN**

Eine Transaktion.

**UEIDTSMD**

Ein Modell für temporäre Speicherwarteschlangen.

**UEIDURIM**

Eine Ressource vom Typ URIMAP.

**UEIDWARB**

Eine WAR-Datei, die Teil eines CICS-Bundles ist.

**UEIDWEBS**

Ein Web-Service (WEBSERVICE).

**UEIDXMLT**

Eine Ressource vom Typ XMLTRANSFORM.

**UEPIDLEN**

Bei öffentlichen Ressourcen handelt es sich bei diesem Parameter um die Adresse der Länge eines einzelnen Ressourcennamens; die Angabe erfolgt als Vollwort-Binärwert.

Bei OSGi-Bundles ist dieser Parameter die Adresse der Länge der Informationen, die ein OSGi-Bundle in CICS im Parameter **UEPIDNAM** eindeutig identifizieren. Die maximale Länge beträgt 526 Byte.

**UEPIDNUM**

Adresse der Anzahl der Ressourcen, die von diesem Aufruf gemeldet werden; die Angabe erfolgt als Vollwort-Binärwert.

**UEPIDNAM**

Adresse einer Liste variabler Länge, die die Namen der einzelnen Ressourcen enthält, die von diesem Aufruf gemeldet werden.

Bei OSGi-Bundles enthält dieser Parameter die Informationen, die ein OSGi-Bundle in CICS eindeutig identifizieren. Die Informationen werden in der folgenden Reihenfolge aufgelistet:

1. 8 Byte, die den JVM-Servernamen enthalten.
2. Ein Vollwort, das die Länge des symbolischen Namens des OSGi-Bundles enthält.
3. Ein Vollwort, das die Länge der Version des OSGi-Bundles enthält.

4. Eine Verkettung des symbolischen Namens und der Version des OSGi-Bundles als Zeichenfolge.

#### **UEPIDREC**

Adresse einer 1-Byte-Kennung, die angibt, ob Ressourcen bei einem Warmstart oder einem Wiederanlauf nach Systemabsturz wiederhergestellt werden. Die Werte lauten wie folgt:

#### **UEIDKEEP**

Die Ressourcen sind bei einem Warmstart oder einem Wiederanlauf nach Systemabsturz wiederherstellbar.

#### **UEIDLOSE**

Die Ressourcen sind nicht wiederherstellbar.

**Anmerkung:** Der Exit wird während eines Neustarts von CICS nicht gesteuert.

#### **UEPDEFTM**

Die Adresse einer Liste variabler Länge, die der Liste in UEPIDNAM entspricht, die die Definitionszeit der Einzelressource als Wert aus 8 Zeichen für STCK enthält.

**Anmerkung:** Die Parameter UEPDEFTM, UEPCHUSR, UEPCHAGT, UEPCHREL, UEPCHTIM, UEPDEFSRC, UEPINUSR, UEPINTIM und UEPINAGT sind für die folgenden Ressourcen gültig: ATOMSERVICE, BUNDLE, CONNECTION, DB2CONN, DB2ENTRY, DB2TRAN, DOCTEMPLATE, ENQMODEL, EPADAPTER, EPADAPTERSET, EVENTBINDING, FILE, IPCONN, JOURNALMODEL, JVMSERVER, LIBRARY, MQCONN, MQINI, OSGI-BUNDLE, PIPELINE, PROFILE, PROCESSTYPE, PROGRAM, TCPIPSERVICE, TDQUEUE, TRANCLASS, TRANSACTION, TS-MODEL, URIMAP, WEBSERVICE und XMLTRANSFORM. Für alle anderen Ressourcen ist der Parameterwert null.

#### **UEPCHUSR**

Adresse einer Liste variabler Länge, die der Liste in UEPIDNAM entspricht, die die Benutzer-ID aus 8 Zeichen enthält, mit der der Agent ausgeführt wurde, der die Einzelressource zuletzt geändert hat.

#### **UEPCHAGT**

Adresse einer Liste variabler Länge, die der Liste in UEPIDNAM entspricht, mit einer 2-Byte-Kennung zur Darstellung des Agenten, der die Einzelressource zuletzt geändert hat. Mögliche Werte:

#### **UEPUNKAGT**

Die Ressource wurde von einem unbekannten Agenten geändert.

#### **UEPCSDAPI**

Die Ressource wurde mithilfe der CSD-API oder CEDA geändert.

#### **UEPCSDBAT**

Die Ressource wurde mithilfe des CSD-Stapelverarbeitungsprogramms DFHCSDUP geändert.

#### **UEPDRPAPI**

Die Ressource wurde mithilfe der CICSplex SM BAS-API geändert.

**UEPAUTOIN**

Die Ressource wurde mithilfe der Funktion für automatische Installation geändert.

**UEPSYSTEM**

Die Ressource wurde durch die aktive CICS-Region geändert.

**UEPDYNAMC**

Die Ressource wurde dynamisch geändert.

**UEPTABLE**

Die Ressource wurde mithilfe einer Tabelle geändert.

**UEPCHREL**

Adresse einer Liste variabler Länge, die der Liste in UEPIDNAM entspricht, die den aus 4 Zeichen bestehenden CICS-Release-Level enthält, der gerade ausgeführt wurde, als die Einzelressource zuletzt geändert wurde.

**UEPCHTIM**

Adresse einer Liste variabler Länge, die der Liste in UEPIDNAM entspricht, die die Änderung der Zeitmarke des CSD-Datensatzes für die Einzelressource als Wert aus 8 Zeichen für STCK enthält.

**UEPDEFSRC**

Adresse einer Liste variabler Länge, die der Liste in UEPIDNAM entspricht, die den/die aus 8 Zeichen bestehende(n) Namen bzw. Quelle der CSD-Gruppe enthält, der/die der Einzelressource entspricht.

**UEPINUSR**

Adresse einer Liste variabler Länge, die der Liste in UEPIDNAM entspricht, die die Benutzer-ID aus 8 Zeichen enthält, mit der die Einzelressource installiert wurde.

**UEPINTIM**

Adresse einer Liste variabler Länge, die der Liste in UEPIDNAM entspricht, die den Zeitpunkt enthält, zu dem die Domäne für die Installation der Einzelressource aufgerufen wurde; die Angabe erfolgt als Wert aus 8 Zeichen für STCK.

**UEPINAGT**

Adresse einer Liste variabler Länge, die der Liste in UEPIDNAM entspricht, mit einer 2-Byte-Kennung zur Darstellung des Agenten, der die Einzelressource installiert hat. Mögliche Werte:

**UEPCSDAPI**

Die Ressource wurde mithilfe der CSD-API oder CEDA installiert.

**UEPCRESPI**

Die Ressource wurde mithilfe der SPI-Befehle vom Typ EXEC CICS CREATE installiert.

**UEPGRPLST**

Die Ressource wurde beim Start mithilfe der Installation einer Gruppenliste (GRPLIST) installiert.

**UEPAUTOIN**

Die Ressource wurde automatisch installiert.

#### **UEPSYSTEM**

Die Ressource wurde durch das aktive CICS-System installiert.

#### **UEPDYNAMC**

Die Ressource wurde dynamisch installiert.

#### **UEPBUNDLE**

Die Ressource wurde im Rahmen einer Bundleimplementierung installiert.

#### **UEPTABLE**

Die Ressource wurde mithilfe einer Tabelle installiert.

#### **UEPAPPTK**

Adresse einer Liste variabler Länge, die ein Token aus 8 Zeichen zur Darstellung der Anwendungsinstanz enthält, zu der diese Ressource gehört. Bei öffentlichen Ressourcen ist diese Adresse null.

#### **UEPAPCTXT**

Bei privaten Ressourcen für Anwendungen, die auf Plattformen implementiert sind, enthält dieser Parameter die Adresse einer Liste variabler Länge, die der Liste in UEPIDNAM entspricht, die die Anwendungskontextinformationen für die Ressource enthält. Die Informationen werden in der folgenden Reihenfolge aufgelistet:

1. Plattformname, mit Leerzeichen auf 64 Zeichen aufgefüllt.
2. Anwendungsname, mit Leerzeichen auf 64 Zeichen aufgefüllt.
3. Hauptversionsnummer für die Anwendung, angegeben als Vollwort-Binärwert.
4. Nebenversionsnummer für die Anwendung, angegeben als Vollwort-Binärwert.
5. Mikroversionsnummer für die Anwendung, angegeben als Vollwort-Binärwert.

CICS stellt einen Pseudobereich (DSECT) namens **DFHUEACD** für die Zuordnung dieser Informationen bereit. Weitere Informationen zu **DFHUEACD** finden Sie unter UEACD - User exit application context in Data Areas.

#### **UEPPLATTK**

Adresse einer Liste variabler Länge, die ein Token aus 8 Zeichen zur Darstellung der Plattforminstanz enthält, zu der diese Ressource gehört. Bei öffentlichen Ressourcen ist diese Adresse null.

#### **Rückgabecodes**

##### **UERCNORM**

Verarbeitung fortsetzen. Dieser Code ist der Standardwert.

##### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

#### **XPI-Aufrufe**

Es können alle XPI-Aufrufe verwendet werden.

---

## Exits XSNON, XSNOFF und XSNEX für Anmeldungen und Abmeldungen

Der Exit XSNON wird aufgerufen, nachdem sich ein Terminalbenutzer angemeldet hat, und der Exit XSNOFF wird aufgerufen, nachdem sich ein Terminalbenutzer abgemeldet hat, und zwar unabhängig davon, ob die Anmeldung bzw. Abmeldung erfolgreich ist oder nicht. Die Exits XSNON und XSNOFF treffen keine Sicherheitsentscheidungen; sie stellen lediglich eine Möglichkeit dar, Benutzer zu überwachen, die sich bei einem CICS-System anmelden und abmelden.

Die Exits werden durch folgende Aktivitäten gesteuert:

- Aufruf eines Befehls vom Typ EXEC CICS SIGNON für ein Terminal (wenn beispielsweise der Terminalbenutzer in die von CICS bereitgestellte Transaktion CESN oder eine funktional entsprechende benutzerdefinierte Anmeldetransaktion eintritt).
- Aufruf eines Befehls vom Typ EXEC CICS SIGNON für ein Ersatzterminal (also ein Terminal, das durch die Routing-Transaktion CRTE oder durch dynamisches Transaktionsrouting angehängt wird).
- Aufruf eines Befehls vom Typ EXEC CICS SIGNOFF für ein Terminal.
- Eingabe eines Befehls vom Typ CANCEL zum Beenden einer CRTE-Routing-Sitzung.
- Abmeldung wegen Zeitlimitüberschreitung.

XSNEX ist ein globaler Benutzerpunkt mit besonderem Zweck, der nur mit dem von IBM gelieferten globalen Benutzerexitprogramm DFH\$SNEX verwendet werden soll.

### Exit XSNON

#### Zeitpunkt des Aufrufs

Wenn sich ein Benutzer anmeldet.

#### Exitspezifische Parameter

##### UEPUSRID

Adresse der Terminalbenutzer-ID.

##### UEPUSRLN

Adresse der Länge der Terminalbenutzer-ID.

##### UEPGRPID

Adresse der Gruppen-ID. Bei erfolgreicher Anmeldung handelt es sich bei der Gruppen-ID um diejenige, der der Benutzer in der betreffenden Anmeldesitzung zugeordnet ist. Bei nicht erfolgreicher Anmeldung handelt es sich um die Gruppen-ID, die vom Benutzer beim Anmeldeversuch angegeben wurde.

##### UEPGRPLN

Adresse der Länge der Gruppen-ID.

##### UEPNETN

Adresse des Netznamens des Terminals.

##### UEPTRMID

Adresse der Terminal-ID.

##### UEPTCTUA

Adresse des TCT-Benutzerbereichs.

**UEPTCTUL**

Adresse der Länge des TCT-Benutzerbereichs.

**UEPTRMTY**

Adresse des Byte für den Terminaltyp.

**UEPSNFLG**

Adresse eines 2-Byte-Felds mit Flags:

Tabelle 1. Flags für Feld UEPSNFLG von XSNON

Flag	Entsprechung	Bedeutung
UEPSNOK	0	Die Anmeldung war erfolgreich.
UEPSNFL	1	Die Anmeldung ist fehlgeschlagen.
UEPSNPSS	2	Die Anmeldung für persistente Sitzungen war erfolgreich.
UEPSNPSF	3	Die Anmeldung für persistente Sitzungen ist fehlgeschlagen.

**UEPSGTYP**

Adresse des Byte für den Anmeldetyp.

Für diesen Parameter gibt es zwei Gleichsetzungen:

UEPSGUID    SIGNON USERID  
UEPSGKER    SIGNON KERBEROS

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

**Exit XSNOFF****Zeitpunkt des Aufrufs**

Wenn ein Benutzer sich abmeldet.

**Exitspezifische Parameter****UEPUSRID**

Adresse der Terminalbenutzer-ID.

**UEPUSRLN**

Adresse der Länge der Terminalbenutzer-ID.

**UEPGRPID**

Adresse der Gruppen-ID.

**UEPGRPLN**

Adresse der Länge der Gruppen-ID.

**UEPNETN**

Adresse des Netznamens des Terminals.

**UEPTRMID**

Adresse der Terminal-ID.



**UEPTCTUA**

Adresse des TCT-Benutzerbereichs.

**UEPTCTUL**

Adresse der Länge des TCT-Benutzerbereichs.

**UEPTRMTY**

Adresse des Byte für den Terminaltyp.

**UEPSNFLG**

Adresse eines 2-Byte-Felds mit Flags:

**UEPSNOK**

Die Abmeldung war erfolgreich.

**UEPSNFL**

Die Abmeldung ist fehlgeschlagen.

**UEPSNNML**

Normale Abmeldung.

**UEPSNTIM**

Abmeldung wegen Zeitlimitüberschreitung.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

**Exit XSSEX**

Der Zweck von XSSEX besteht darin, zusammen mit den zugehörigen unterstützenden Beispielprogrammen, kurzfristige Hilfe beim Durchführen von Upgrades bereitzustellen. Der Exit wurde entwickelt, um Ihnen Zeit zu geben, diejenigen Anwendungsprogramme, die davon abhängen, wie CICS Anforderungen vom Typ **EXEC CICS SIGNON** und **EXEC CICS SIGNOFF** vor CICS TS 2.1 verarbeitet, entsprechend anzupassen, damit diese Anwendungsprogramme mit dem aktuellen Verhalten funktionieren.

**Anmerkung: XSSEX dient nur zur Durchführung von Upgrades.** Ändern Sie sämtliche Anwendungen so, dass keine Abhängigkeit mehr vom alten Anmelde- und Abmeldeverhalten besteht.

Es gibt keine exitspezifischen Parameter für diesen globalen Benutzerexit, der immer dann aufgerufen wird, wenn ein Anwendungsprogramm einen Befehl vom Typ **EXEC CICS SIGNON** oder **EXEC CICS SIGNOFF** ausgibt. Es ist nicht vorgesehen, dass Sie ein eigenes globales Benutzerexitprogramm für diesen Exitpunkt schreiben. IBM stellt das Programm DFH\$SEX bereit, dessen einziger Zweck darin besteht, CICS zu veranlassen, die Befehle **EXEC CICS SIGNON** und **EXEC CICS SIGNOFF** in derselben Weise wie in CICS TS 1.3 und früher zu verarbeiten.

Folgende Programme werden bereitgestellt:

**DFH\$SEX**

Dieses Benutzerexitprogramm wird in SDFHSAMP bereitgestellt. Die einzige von diesem Programm ausgeführte Funktion besteht darin, den Rückgabecode

UERCPCREX festzulegen. Dieser Rückgabecode bewirkt, dass die Sicherheitsdomäne das Verhalten von CICS wie in CICS TS 1.3 und früher wiederherstellt. Sie können dieses Benutzerexitprogramm mithilfe von DFH\$SNPI aktivieren.

#### **DFH\$SNPI**

Dieses Postinitialisierungsprogramm wird in SDFHSAMP bereitgestellt. Es gibt den Befehl EXEC CICS ENABLE PROGRAM('DFH\$SNEX') EXIT('XSNE') aus, um das von IBM gelieferte Benutzerexitprogramm DFH\$SNEX in den letzten Phasen der CICS-Initialisierung zu aktivieren.

Um dieses Programm verwenden zu können, müssen Sie einen entsprechenden Eintrag zum ersten Abschnitt Ihrer PLTPI-Tabelle hinzufügen (also vor der Anweisung für DFHDELIM). Beispiel:

```
DFHPLT TYPE=INITIAL,SUFFIX=SN
DFHPLT TYPE=ENTRY,PROGRAM=DFH$SNPI
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
DFHPLT TYPE=FINAL
END
```

---

## **Exit XSTOUT für die Statistikdomäne**

Bei Aufruf wird XSTOUT die Adresse eines Puffers übergeben, der mindestens einen Statistikdatensatz enthält. Der Puffer kann Datensätze für verschiedene Ressourcentypen enthalten, beispielsweise für Verbindungen und Modusnamen. Darüber hinaus kann der Puffer sowohl spezifische als auch globale Informationen enthalten, beispielsweise Ladeprogrammstatistikdaten für einzelne Programme und Ladeprogrammstatistikdaten für alle Programme.

Das Exitprogramm kann die Typen von Datensätzen im Puffer anhand den ihnen zugeordneten STID-Werten identifizieren. (Die STID-Werte werden unter CICS statistics data section beschrieben.)

Mithilfe von XSTOUT können Sie verhindern, dass der Inhalt des Statistikdatenpuffers in SMF geschrieben wird. Hierbei ist zu beachten, dass der Exit nicht dazu verwendet werden kann, einzelne Datensätze im Puffer selektiv zu unterdrücken. Die Werte der exitspezifischen Parameter sollten vom Exitprogramm keinesfalls geändert werden.

Einige Statistikdatensätze werden unter Umständen bereits in einem sehr frühen Stadium der CICS-Initialisierung erstellt; diese Datensätze werden nicht an XSTOUT übergeben. Der früheste Zeitpunkt, zu dem ein globaler Benutzerexit aktiviert werden kann, ist während der PLT-Verarbeitung. Davor ist das Aktivieren von Exits nicht möglich.

## **Exit XSTOUT**

### **Zeitpunkt des Aufrufs**

Bevor ein Statistikdatensatz in System Management Facilities (SMF) geschrieben wird.

### **Exitspezifische Parameter**

Die Felder UEPPROG, UEPTERM, UEPTRANID und UEPUSER sind nur für angeforderte Statistiken von Bedeutung (bei Verwendung der CICS Explorer-Betriebsansicht für **Regionen**, des Befehls **CEMT PERFORM STATISTICS RECORD** oder des Befehls **EXEC CICS PERFORM STATISTICS RECORD**).

#### **UEPPROG**

Adresse des 8-Byte-Anwendungsprogrammnamens.

**UEPSCLD**

Adresse eines aus 8 Zeichen bestehenden Felds mit dem Erfassungsdatum (MMTTJJJJ).

**UEPSDATE**

Adresse eines 6-Byte-Zeichenfelds mit dem Erfassungsdatum (MMTTJJ).

**UEPSIVAL**

Adresse eines 6-Byte-Zeichenfelds mit der Intervallzeit (HHMMSS). Dieses Feld ist nur für Intervallstatistiken von Bedeutung.

**UEPSIVN**

Adresse der 4-Byte-Intervallnummer. Dieses Feld ist nur für Intervallstatistiken von Bedeutung.

**UEPSRLEN**

Adresse der hexadezimalen Länge (4 Byte) des Statistikdatensatzes.

**UEPSTATS**

Adresse eines Puffers mit mindestens einem Statistikdatensatz. Bei nicht angeforderten Statistikdaten enthält der Puffer immer einen einzelnen Datensatz; bei anderen Statistiktypen kann er mehrere Datensätze enthalten. Die Länge des Puffers wird durch den Parameter UEPSRLEN adressiert.

**UEPSTIME**

Adresse eines 6-Byte-Zeichenfelds mit der Erfassungszeit (HHMMSS).

**UEPSTYPE**

Adresse des 3-Byte-Zeichenfelds für den Statistiktyp. Mögliche Werte:

**INT** Intervallstatistik.

**EOD** Tagesabschlussstatistik.

**REQ** Angeforderte Statistik.

**RRT** Angeforderte Zurücksetzungsstatistik.

**USS** Nicht angeforderte Statistik.

**UEPTERM**

Adresse der 4-Byte-Terminal-ID.

**UEPTRANID**

Adresse der 4-Byte-Transaktions-ID.

**UEPUSER**

Adresse der 8-Byte-Benutzer-ID.

**Rückgabecodes****UERCBYP**

Ausgabe des Statistikdatenpuffers für SMF unterdrücken.

**UERCNORM**

Verarbeitung fortsetzen.

**XPI-Aufrufe**

WAIT\_MVS kann verwendet werden. Hierbei ist jedoch zu beachten, dass der Wartestatus nicht durch die Verwendung von CEMT- oder SPI-Befehlen bereinigt werden kann. Verwenden Sie keine anderen Aufrufe.

---

## Exit XSRAB für das Systemwiederherstellungsprogramm

Der Exit XSRAB wird aufgerufen, wenn das Systemwiederherstellungsprogramm (DFHSRP) in der Systemwiederherstellungstabelle (SRT) eine Übereinstimmung für einen Abbruchcode des Betriebssystems findet.

### Zeitpunkt des Aufrufs

Der Exit XSRAB wird aufgerufen, wenn das Systemwiederherstellungsprogramm (DFHSRP) in der Systemwiederherstellungstabelle (SRT) eine Übereinstimmung für einen Abbruchcode des Betriebssystems findet. Informationen zum Definieren von Einträgen in der SRT finden Sie unter Systemwiederherstellungstabelle (SRT).

Die Systemwiederherstellungstabelle wird nur dann verarbeitet, und der Exit wird nur dann gesteuert, wenn ein MVS-Abbruch unter einem grundlegenden Tasksteuerblock (TCB) von CICS auftritt, d. h. unter QR, RO, CO, SZ, RP oder FO. Bei nicht grundlegenden TCB-Typen (wie beispielsweise L8, SL, SO oder S8) wird der Exit nicht gesteuert.

### Exitspezifische Parameter

#### UEPERROR

Adresse der Fehlerdatenstruktur (SRP\_ERROR\_DATA), die die folgenden Felder enthält:

#### SRP\_ERROR\_TYPE

Fehlertyp aus 4 Zeichen; dieser Typ ist immer ASRB.

#### SRP\_SYS\_ABCODE

2 Byte, die den Systemabbruchcode XXX im Binärformat enthalten (beispielsweise D37).

#### SRP\_USER\_ABCODE

2 Byte, die den Benutzerabbruchcode NNNN im Binärformat enthalten (beispielsweise 0999).

#### SRP\_ERROR\_TRANID

Feld aus 4 Zeichen, das die ID der Transaktion enthält, die abgebrochen wurde.

#### SRP\_ERROR\_STACK\_NAME

Feld aus 8 Zeichen, das den Namen des aktuellen Kernel-Stackeintrags für die Transaktion zum Zeitpunkt des Abbruchs enthält.

#### SRP\_ERROR\_PPT\_NAME

Feld aus 8 Zeichen, das den Namen des aktuellen Programms für die Transaktion enthält. Dieses Feld enthält nur dann einen Wert, wenn das Flag SRP\_PPT\_ENTRY gesetzt ist.

#### SRP\_ERROR\_OFFSET

Vollwort, das wie folgt den Offset in das Programm enthält, das abgebrochen wurde:

- Ist das Flag SRP\_PPT\_ENTRY gesetzt, wird der Offset in SRP\_ERROR\_PPT\_NAME angegeben.
- Andernfalls wird der Offset in SRP\_ERROR\_STACK\_NAME angegeben.

Dieses Feld enthält nur dann einen Wert, wenn das Flag SRP\_VALID\_OFFSET gesetzt ist.

## **SRP\_ERROR\_FLAGS**

1 Byte, das Flags enthält:

### **SRP\_CICS\_CODE**

Der Abbruch trat während der Ausführung von CICS-Code auf.

### **SRP\_USER\_CODE**

Der Abbruch trat während der Ausführung von Benutzeranwendungscode auf.

### **SRP\_PPT\_ENTRY**

Der Abbruch trat während der Ausführung von SRP\_ERROR\_PPT\_NAME auf. Wenn dieses Flag nicht gesetzt ist, trat der Abbruch während der Ausführung von SRP\_ERROR\_STACK\_NAME auf.

### **SRP\_VALID\_OFFSET**

Es konnte ein aussagekräftiger Offset ermittelt werden.

### **SRP\_VALID\_REASON**

MVS hat einen Ursachencode für den Abbruch bereitgestellt.

### **SRP\_NOT\_CICS\_RB**

Die Steuerung lag zum Zeitpunkt des Abbruchs nicht bei CICS RB (dies bedeutet, dass der Abbruch in einem von CICS aufgerufenen Systemservice auftrat).

## **SRP\_CICS\_ERROR\_REASON**

Feld aus 4 Zeichen, das den Ursachencode für Abbruch von MVS enthält. Dieses Feld enthält nur dann einen Wert, wenn das Flag SRP\_VALID\_REASON gesetzt ist.

## **SRP\_CICS\_ERROR\_DATA**

Ein Bereich, der die letzte Aktion beschreibt, die vor dem Abbruch von CICS ausgeführt wurde. Der Bereich enthält Folgendes:

### **SRP\_CICS\_EC\_PSW**

Feld aus 8 Zeichen, das das Programmstatuswort (PSW) für den erweiterten Betriebsmodus (Extended Control, EC) enthält.

### **SRP\_CICS\_PSW16**

Feld aus 16 Zeichen, das das 128-Bit-Programmstatuswort (PSW) enthält.

### **SRP\_CICS\_EC\_INT**

Feld aus 8 Zeichen, das den Interrupt-Code und ILC enthält.

### **SRP\_CICS\_REGST**

Feld aus 64 Zeichen, das den Inhalt des Mehrzweckregisters (General Purpose, GP) enthält.

### **SRP\_CICS\_EXEC\_KEY**

1 Byte, das den PSW-Schlüssel im Format X'0n' enthält.

### **SRP\_SYSTEM\_ERROR\_DATA**

Ein Bereich, der die letzte Aktion beschreibt, die vor dem Abbruch vom System ausgeführt wurde. Der Bereich enthält Folgendes:

#### **SRP\_SYSTEM\_EC\_PSW**

Feld aus 8 Zeichen, das das Programmstatuswort (PSW) für den erweiterten Betriebsmodus (Extended Control, EC) enthält.

#### **SRP\_SYSTEM\_PSW16**

Feld aus 16 Zeichen, das das 128-Bit-Programmstatuswort (PSW) enthält.

#### **SRP\_SYSTEM\_EC\_INT**

Feld aus 8 Zeichen, das den Interrupt-Code und ILC enthält.

#### **SRP\_SYSTEM\_REGST**

Feld aus 64 Zeichen, das den Inhalt des Mehrzweckregisters (General Purpose, GP) enthält.

#### **SRP\_SYSTEM\_EXEC\_KEY**

1 Byte, das den PSW-Schlüssel im Format X'0n' enthält.

### **SRP\_ERROR\_FP\_REGS**

Ein Bereich, der den Inhalt der Gleitkommaregister zum Zeitpunkt des Abbruchs beschreibt. Der Bereich enthält Folgendes:

#### **SRP\_FP\_REG\_0**

Gleitkommaregister 0.

#### **SRP\_FP\_REG\_2**

Gleitkommaregister 2.

#### **SRP-FP\_REG\_4**

Gleitkommaregister 4.

#### **SRP-FP\_REG\_6**

Gleitkommaregister 6.

### **SRP\_ADDITIONAL\_REG\_INFO**

Ein Bereich, der zusätzliche Registerinformationen enthält.

### **SRP\_ADDITIONAL\_REGS\_FLAG**

1 Byte, das Flags enthält:

#### **SRP\_CICS\_GPR64\_AVAIL**

Die 64-Bit-Mehrzweckregister von CICS sind verfügbar.

#### **SRP\_SYSTEM\_GPR64\_AVAIL**

Die 64-Bit-Mehrzweckregister des Systems sind verfügbar.

#### **SRP\_ADDITIONAL\_FPR\_AVAIL**

Zusätzliche Gleitkommaregister sind verfügbar.

### **SRP\_CICS\_GP64\_REGS**

128-Byte-Bereich, der die 64-Bit-Mehrzweckregister von CICS zum Zeitpunkt des Abbruchs enthält.

**SRP\_SYSTEM\_GP64\_REGS**

128-Byte-Bereich, der die 64-Bit-Mehrzweckregister des Systems zum Zeitpunkt des Abbruchs enthält.

**SRP\_ADDITIONAL\_FPR\_REGS**

132-Byte-Bereich, der die zusätzlichen Gleitkommaregister zum Zeitpunkt des Abbruchs enthält.

**SRP\_FP\_REGS**

128-Byte-Bereich, der alle Gleitkommaregister zum Zeitpunkt des Abbruchs enthält.

**SRP\_FPC\_REG**

4-Byte-Feld, das das FPC-Register zum Zeitpunkt des Abbruchs enthält.

**SRP\_VR\_REGS**

512-Byte-Bereich, der alle Vectorregister zum Zeitpunkt des Abbruchs enthält.

Ist das Flag `SRP_NOT_CICS_RB` gesetzt, dann beschreibt `SRP_CICS_ERROR_DATA` die letzte Aktion, die vor dem Abbruch von CICS ausgeführt wurde, und `SRP_SYSTEM_ERROR_DATA` beschreibt die letzte Aktion, die vom Systemservice (beispielsweise z/OS Communications Server, VSAM oder MVS) ausgeführt wurde.

Sie können die Fehlerdatenstruktur (`SRP_ERROR_DATA`), die an den Exit `XSAB` übergeben wird, mithilfe des Makros `DFHSRED TYPE=DSECT` zuordnen. Das Format von `SRP_ERROR_DATA` finden Sie unter `SRED - System recovery error data`.

**Rückgabecodes****UERCNOCA**

Die Task wird mit Abbruchcode `ASRB` abgebrochen. Es werden keine Abbruchexits auf Programmebene abgebrochen, die der betreffenden Task zugeordnet sind.

**UERCCANC**

Die Task wird mit Abbruchcode `ASRB` abgebrochen. Es werden auch alle Abbruchexits auf Programmebene abgebrochen, die der betreffenden Task zugeordnet sind.

**UERCCICS**

CICS wird abgebrochen.

**XPI-Aufrufe**

Da CICS den Exit `XSAB` in einer Fehlerumgebung aufruft, können Sie nur einen Teil der XPI-Aufrufe verwenden.

Nur `TRACE_PUT` ist zur allgemeinen Verwendung verfügbar.

Sie können `WAIT_MVS` verwenden, aber erst, nachdem das Exitprogramm (anhand der Felder `SRP_CICS_CODE` und `SRP_USER_CODE`) ermittelt hat, dass der Abbruch im Benutzeranwendungscode und nicht im CICS-Code aufgetreten ist.

**Important:**

- Gehen Sie mit Sorgfalt vor, wenn Sie ein Programm codieren, das am Exitpunkt `XSAB` ausgeführt werden soll. Wenn das Exitprogramm dazu führt, dass das

Systemwiederherstellungsprogramm erneut gestartet wird (beispielsweise wenn ein Programmfehler auftritt), wird CICS mit einer Nachricht vom Typ DFHSR06xx abgebrochen.

- Der Standardrückgabecode lautet UERCNOCA; dadurch wird sichergestellt, dass die Task abgebrochen wird, wenn der Exit fehlerhaft ist.
- An diesem Exitpunkt ist der Rückgabecode UERCNORM nicht vorhanden, da der Exit nach einem Fehler aufgerufen wird.
- Der Rückgabecode UERCPURG sollte vom Exit nicht gesetzt werden.

---

## Exit XSTERM für das Systemabschlussprogramm

Der Exit XSTERM kann beispielsweise verwendet werden, um endgültige Statistikdaten an Ihre SMF-Statistikdatasets auszugeben und diese zu schließen. Hierbei ist zu beachten, dass CICS-VSAM-Datasets und BDAM-Datasets bereits vor dem Aufruf des Exits durch die Dateisteuerung von CICS geschlossen wurden.

### Zeitpunkt des Aufrufs

Während der zweiten Quiesce-Phase eines normalen Systemabschlusses, unmittelbar vor dem Löschen des Inhalts der Puffer für transiente Daten und temporären Speicher. Während einer sofortigen Beendigung (Typ IMMEDIATE) wird der Exit nicht aufgerufen.

### Exitspezifische Parameter

Keine.

### Rückgabecodes

UERCNORM

Verarbeitung fortsetzen.

### XPI-Aufrufe

Es können alle anderen XPI-Aufrufe mit Ausnahme von WRITE\_JOURNAL\_DATA verwendet werden. Die Verwendung von XPI-Aufrufen wird jedoch nicht empfohlen, da sie dazu führen könnten, dass die Task die Steuerung verliert, wodurch es unter Umständen einer anderen Task ermöglicht wird, mehr Überwachungsdaten zu schreiben.

---

## Exits für die temporäre Speicherdomäne (XTSQRIN, XTSQROUT, XTSP-TIN, XTSP-OUT)

Diese Exits werden aufgerufen, wenn der temporäre Speicher gesteuert oder überwacht werden muss.

Sie können die Exits XTSQRIN, XTSQROUT, XTSP-TIN und XTSP-OUT für die temporäre Speicherdomäne ändern, um folgende Tasks auszuführen:

- Für eine Anforderung zum Erstellen einer Warteschlange: Angeben, ob die Warteschlange in den Hauptspeicher oder in den Zusatzspeicher gestellt werden soll, und Angeben der Wiederherstellbarkeit.
- Überwachen der Verwendung des temporären Speichers.
- Steuern der Sicherheit für temporäre Speicherwarteschlangen.

Bei Anforderungen für temporären Speicher, die über Funktionen über eine IPIC-Verbindung übertragen worden sind, hat der Parameter UEPTERM den Wert null. Um IPIC-Verbindungen für Anforderungen für temporären Speicher zu verwenden, muss von XTSQRIN, XTSQROUT, XTSP-TIN und XTSP-OUT unbedingt über-



prüft werden, ob der Parameter UEPTERM einen Wert ungleich null aufweist, bevor versucht wird, ihn als Adresse zu verwenden.

XTSQRIN, XTSQROUT, XTSPIN und XTSPOUT müssen gemäß threadsicheren Standards codiert und als threadsicher deklariert werden, um die Vorteile von Threadsicherheit bei der Verwendung einer IPIC-Verbindung nutzen zu können.

Die temporäre Speicherdomäne verfügt über zwei Hauptzugänge, TSQR und TSPT, die folgende Funktionen unterstützen:

**TSQR** Write, Rewrite, Read\_into, Read\_set, Read\_next\_into, Read\_next\_set und Delete.

**TSPT** Put, Put\_replace, Get, Get\_set, Get\_release, Get\_release\_set und Release.

Die TSQR-Funktionen entsprechen den Funktionen, die über die EXEC CICS-Schnittstelle (oder über DFHTS TYPE=PUTQ, GETQ oder PURGE) verfügbar sind. Die TSPT-Funktionen werden vom Intervallsteuerprogramm zur Unterstützung von START- und RETRIEVE-Funktionen (oder DFHTS TYPE=PUT, GET oder RELEASE) verwendet.

## Exit XTSQRIN

Der Exit XTSQRIN wird vor Ausführung einer Benutzeranforderung der Schnittstelle für temporären Speicher an eine Benutzerwarteschlange für temporären Speicher (beispielsweise eine Anforderung vom Typ WRITEQ TS oder READQ TS) aufgerufen.

### Exitspezifische Parameter

#### UEPTRANID

Adresse der 4-Byte-Transaktions-ID.

#### UEPUSER

Adresse der 8-Byte-Benutzer-ID.

#### UEPTERM

Null oder die Adresse der 4-Byte-Terminal-ID. Wird keine Adresse zurückgegeben, so kann dies bedeuten, dass diese Anforderung mittels einer Funktion über eine IPIC-Verbindung übertragen wurde.

#### UEPPROG

Adresse des 8-Byte-Anwendungsprogrammnamens.

#### UEP\_TS\_FUNCTION

Adresse eines Byte, das die Funktion enthält:

- UEP\_TS\_FUN\_WRITE
- UEP\_TS\_FUN\_REWRITE
- UEP\_TS\_FUN\_READ\_INT0
- UEP\_TS\_FUN\_READ\_SET
- UEP\_TS\_FUN\_READ\_NEXT\_INT0
- UEP\_TS\_FUN\_READ\_NEXT\_SET
- UEP\_TS\_FUN\_DELETE

#### UEP\_TS\_QUEUE\_NAME

Adresse eines 16-Byte-Felds mit dem Warteschlangennamen.

**UEP\_TS\_DATA\_P**

Adresse eines Vollworts, das die Adresse der Daten enthält. (Anforderungen vom Typ WRITE und REWRITE.)

**UEP\_TS\_DATA\_L**

Adresse eines Vollworts, das die Länge der Daten enthält. (Anforderungen vom Typ WRITE und REWRITE.)

**UEP\_TS\_ITEM\_NUMBER**

Adresse eines Vollworts, das die Elementnummer enthält. (Anforderungen vom Typ REWRITE, READ\_INT0 und READ\_SET.)

**UEP\_TS\_STORAGE\_TYPE**

Adresse eines Byte, das den Speichertyp enthält. (Anforderungen vom Typ WRITE.)

Bei Eingabe für den Exit wird der Parameter entweder auf UEP\_TS\_STORAGE\_TYPE\_MAIN oder auf UEP\_TS\_STORAGE\_TYPE\_AUX\_TST gesetzt. Dieser Parameter kann vom Exit in einen der nachstehenden Werte geändert werden.

Bitte beachten Sie, dass die Einstellung dieses Parameters ohne Wirkung bleibt, wenn CICS® ausschließlich mit Unterstützung für den temporären Speichertyp MAIN (Hauptspeicher) initialisiert wurde. Weitere Informationen hierzu finden Sie in der Beschreibung zum TS system initialization parameter.

**UEP\_TS\_STORAGE\_TYPE\_MAIN**

Hauptspeicher.

**UEP\_TS\_STORAGE\_TYPE\_AUX\_TST**

Zusatzspeicher (die Wiederherstellbarkeit wird durch die Ressourcendefinition festgelegt).

**UEP\_TS\_STORAGE\_TYPE\_AUX\_RECOV\_YES**

Zusatzspeicher (wiederherstellbar).

**UEP\_TS\_STORAGE\_TYPE\_AUX\_RECOV\_NO**

Zusatzspeicher (nicht wiederherstellbar).

**Rückgabecodes****UERCNORM**

Normal.

**UERCPURG**

Bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

**API- und SPI-Aufrufe**

Es können keine verwendet werden.

**Exit XTSQROUT**

Der Exit XTSQROUT wird nach Ausführung einer Benutzeranforderung der Schnittstelle für temporären Speicher an eine Benutzerwarteschlange für temporären Speicher (beispielsweise eine Anforderung vom Typ WRITEQ TS oder READQ TS) aufgerufen.

**Exitspezifische Parameter**

**UEPTRANID**

Adresse der 4-Byte-Transaktions-ID.

**UEPUSER**

Adresse der 8-Byte-Benutzer-ID.

**UEPTERM**

Null oder die Adresse der 4-Byte-Terminal-ID. Wird keine Adresse zurückgegeben, so kann dies bedeuten, dass diese Anforderung mittels einer Funktion über eine IPIC-Verbindung übertragen wurde.

**UEPPROG**

Adresse des 8-Byte-Anwendungsprogrammnamens.

**UEP\_TS\_FUNCTION**

Adresse eines Byte, das die Funktion enthält:

- UEP\_TS\_FUN\_WRITE
- UEP\_TS\_FUN\_REWRITE
- UEP\_TS\_FUN\_READ\_INT0
- UEP\_TS\_FUN\_READ\_SET
- UEP\_TS\_FUN\_READ\_NEXT\_INT0
- UEP\_TS\_FUN\_READ\_NEXT\_SET
- UEP\_TS\_FUN\_DELETE

**UEP\_TS\_QUEUE\_NAME**

Adresse eines 16-Byte-Felds mit dem Warteschlangennamen.

**UEP\_TS\_DATA\_P**

Adresse eines Vollworts, das die Adresse der Daten enthält. (Alle Anforderungen außer DELETE).

**UEP\_TS\_DATA\_L**

Adresse eines Vollworts, das die Länge der Daten enthält. (Alle Anforderungen außer DELETE).

**UEP\_TS\_ITEM\_NUMBER**

Adresse eines Vollworts, das die Elementnummer enthält. (Anforderungen vom Typ REWRITE, READ\_INT0 und READ\_SET.)

**UEP\_TS\_TOTAL\_ITEMS**

Adresse eines Vollworts, das die Gesamtzahl der Elemente in der Warteschlange enthält. (Alle Anforderungen außer DELETE).

**UEP\_TS\_RESPONSE**

Adresse eines Byte, das nach Ausführung einer Anforderung die entsprechende Antwort enthält:

- UEP\_TS\_RESPONSE\_OK
- UEP\_TS\_RESPONSE\_PURGED
- UEP\_TS\_RESPONSE\_EXCEPTION
- UEP\_TS\_RESPONSE\_DISASTER
- UEP\_TS\_RESPONSE\_INVALID

**Rückgabecodes****UERCNORM**

Antwort 'Normal'.

#### **UERC PURG**

Es wurde eine Antwort vom Typ 'Bereinigt' von einer XPI-Anforderung empfangen.

#### **XPI-Aufrufe**

Es können alle verwendet werden.

#### **API- und SPI-Aufrufe**

Es können keine verwendet werden.

## **Exit XTSPTIN**

Der Exit XTSPTIN wird vor Ausführung einer Anforderung der Schnittstelle für temporären Speicher an eine interne CICS-Warteschlange (beispielsweise Intervallsteuerungs- oder BMS-Warteschlangen) aufgerufen.

#### **Exitspezifische Parameter**

##### **UEPTRANID**

Adresse der 4-Byte-Transaktions-ID.

##### **UEPUSER**

Adresse der 8-Byte-Benutzer-ID.

##### **UEPTERM**

Null oder die Adresse der 4-Byte-Terminal-ID. Wird keine Adresse zurückgegeben, so kann dies bedeuten, dass diese Anforderung mittels einer Funktion über eine IPIC-Verbindung übertragen wurde.

##### **UEPPROG**

Adresse des 8-Byte-Anwendungsprogrammnamens.

##### **UEP\_TS\_FUNCTION**

Adresse eines Byte, das die Funktion enthält:

- UEP\_TS\_FUN\_PUT
- UEP\_TS\_FUN\_PUT\_REPLACE
- UEP\_TS\_FUN\_GET
- UEP\_TS\_FUN\_GET\_SET
- UEP\_TS\_FUN\_GET\_RELEASE
- UEP\_TS\_FUN\_GET\_RELEASE\_SET
- UEP\_TS\_FUN\_RELEASE

##### **UEP\_TS\_QUEUE\_NAME**

Adresse eines 16-Byte-Felds mit dem Warteschlangennamen.

##### **UEP\_TS\_DATA\_P**

Adresse eines Vollworts, das die Adresse der Daten enthält. (PUT und PUT\_REPLACE).

##### **UEP\_TS\_DATA\_L**

Adresse eines Vollworts, das die Länge der Daten enthält. (PUT und PUT\_REPLACE).

##### **UEP\_TS\_STORAGE\_TYPE**

Adresse eines Byte, das den Speichertyp enthält. (Anforderungen vom Typ PUT).

Bei Eingabe für den Exit wird der Parameter entweder auf UEP\_TS\_STORAGE\_TYPE\_MAIN oder auf UEP\_TS\_STORAGE\_TY-

PE\_AUX\_TST gesetzt. Dieser Parameter kann vom Exit in einen der nachstehenden Werte geändert werden.

Bitte beachten Sie, dass die Einstellung dieses Parameters ohne Wirkung bleibt, wenn CICS ausschließlich mit Unterstützung für den temporären Speichertyp MAIN (Hauptspeicher) initialisiert wurde.

**UEP\_TS\_STORAGE\_TYPE\_MAIN**  
Hauptspeicher.

**UEP\_TS\_STORAGE\_TYPE\_AUX\_TST**  
Zusatzspeicher (die Wiederherstellbarkeit wird durch die Ressourcendefinition festgelegt).

**UEP\_TS\_STORAGE\_TYPE\_AUX\_RECOV\_YES**  
Zusatzspeicher (wiederherstellbar).

**UEP\_TS\_STORAGE\_TYPE\_AUX\_RECOV\_NO**  
Zusatzspeicher (nicht wiederherstellbar).

#### **Rückgabecodes**

**UERCNORM**  
Normal.

**UERCPURG**  
Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**  
Es können alle verwendet werden.

**API- und SPI-Aufrufe**  
Es können keine verwendet werden.

## **Exit XTSPTOUT**

Der Exit XTSPTOUT wird nach Ausführung einer Anforderung der Schnittstelle für temporären Speicher an eine interne CICS-Warteschlange (beispielsweise Intervallsteuerungs- oder BMS-Warteschlangen) aufgerufen. Nach Ausführung einer TSPT-Anforderung. Es können keine Parameter geändert werden.

#### **Exitspezifische Parameter**

**UEPTRANID**  
Adresse der 4-Byte-Transaktions-ID.

**UEPUSER**  
Adresse der 8-Byte-Benutzer-ID.

**UEPTERM**  
Null oder die Adresse der 4-Byte-Terminal-ID. Wird keine Adresse zurückgegeben, so kann dies bedeuten, dass diese Anforderung mittels einer Funktion über eine IPIC-Verbindung übertragen wurde.

**UEPPROG**  
Adresse des 8-Byte-Anwendungsprogrammnamens.

**UEP\_TS\_FUNCTION**  
Adresse eines Byte, das die Funktion enthält:

- UEP\_TS\_FUNCTION\_PUT
- UEP\_TS\_FUN\_PUT\_REPLACE
- UEP\_TS\_FUN\_GET

- UEP\_TS\_FUN\_GET\_SET
- UEP\_TS\_FUN\_GET\_RELEASE
- UEP\_TS\_FUN\_GET\_RELEASE\_SET
- UEP\_TS\_FUN\_RELEASE

#### **UEP\_TS\_QUEUE\_NAME**

Adresse eines 16-Byte-Felds mit dem Warteschlangennamen.

#### **UEP\_TS\_DATA\_P**

Adresse eines Vollworts, das die Adresse der Daten enthält. (Alle Anforderungen außer RELEASE).

#### **UEP\_TS\_DATA\_L**

Adresse eines Vollworts, das die Länge der Daten enthält. (Alle Anforderungen außer RELEASE).

#### **UEP\_TS\_RESPONSE**

Adresse eines Byte, das nach Ausführung einer Anforderung die entsprechende Antwort enthält:

- UEP\_TS\_RESPONSE\_OK
- UEP\_TS\_RESPONSE\_PURGED
- UEP\_TS\_RESPONSE\_EXCEPTION
- UEP\_TS\_RESPONSE\_DISASTER
- UEP\_TS\_RESPONSE\_INVALID

#### **Rückgabecodes**

##### **UERCNORM**

Antwort 'Normal'.

##### **UERCPURG**

Es wurde eine Antwort vom Typ 'Bereinigt' von einer XPI-Anforderung empfangen.

#### **XPI-Aufrufe**

Es können alle verwendet werden.

#### **API- und SPI-Aufrufe**

Es können keine verwendet werden.

---

## **Exits XTSEREQ und XTSEREQC für das EXEC-Schnittstellenprogramm für temporären Speicher**

Mit dem Exit XTSEREQ können Sie API-Anforderungen für temporären Speicher abfangen, bevor entsprechende Aktionen für diese Anforderungen ausgeführt werden. Mithilfe des Exits XTSEREQC können Sie die entsprechende Antwort abfangen, nachdem eine API-Anforderung für temporären Speicher abgeschlossen worden ist.

Dies betrifft die folgenden API-Anforderungen:

- **EXEC CICS WRITEQ TS**
- **EXEC CICS READQ TS**
- **EXEC CICS DELETEQ TS**

Mithilfe von XTSEREQ können Sie folgende Aktionen ausführen:

- Analysieren der API-Parameterliste (Funktion, Schlüsselwörter, Argumentwerte und Antworten).

- Ändern beliebiger Eingabeparameterwerte vor Ausführung einer Anforderung.
- Verhindern der Ausführung einer Anforderung.

Mithilfe von XTSEREQC können Sie folgende Aktionen ausführen:

- Analysieren der API-Parameterliste.
- Ändern beliebiger Ausgabeparameterwerte nach Anforderungsausführung.

Darüber hinaus ist Folgendes möglich:

- Übergeben von Daten zwischen den Exitprogrammen für XTSEREQ und XTSEREQC, wenn diese Programme für dieselbe Anforderung aufgerufen werden.
- Übergeben von Daten zwischen den Exitprogrammen für temporären Speicher, wenn diese Programme innerhalb derselben Task aufgerufen werden.

Es kann sein, dass Programme, die über die Exits in der temporären Speicherdomäne (XTSQRIN, XTSQROUT, XTSPTIN und XTSPTOUT) aufgerufen werden, die von XTSEREQ eingerichteten Situationen ändern. Daher müssen Sie die Reihenfolge berücksichtigen, in der die Exits aufgerufen werden.

Wenn alle Exits für temporären Speicher aktiviert sind, lautet die Reihenfolge für das Aufrufen der Exits wie folgt:

1. XTSEREQ
2. XTSQRIN
3. XTSQROUT
4. XTSEREQC

## Exit XTSEREQ

Mit dem Exit XTSEREQ können Sie API-Anforderungen für temporären Speicher abfangen, bevor entsprechende Aktionen für diese Anforderungen ausgeführt werden.

### Zeitpunkt des Aufrufs

Bevor CICS eine API-Anforderung für temporären Speicher verarbeitet.

### Exitspezifische Parameter

#### UEPCLPS

Adresse einer Kopie der Befehlsparameterliste. Siehe „Parameterstruktur der Befehlsebene“ auf Seite 256.

#### UEPTQTOK

Adresse eines 4-Byte-Bereichs, der verwendet werden kann, um für eine einzelne Anforderung für temporären Speicher Informationen zwischen XTSEREQ und XTSEREQC zu übergeben.

#### UEPRCODE

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

#### UEPRES

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

#### UEPRES2

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

**UEPTSTOK**

Adresse eines 4-Byte-Tokens, das verwendet werden kann, um innerhalb derselben Task Informationen zwischen aufeinanderfolgenden Anforderungen für temporären Speicher zu übergeben (beispielsweise zwischen aufeinanderfolgenden Aufrufen des Exits XTSEREQ).

**UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

**UEPRSRCE**

Adresse einer 8 Zeichen langen Kopie des EIB-Ressourcenwerts, EIBRSRCE.

**Rückgabecodes****UERCBYP**

Diese Anforderung umgehen.

**UERCNORM**

Verarbeitung fortsetzen.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

**API- und SPI-Befehle**

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

EXEC CICS SHUTDOWN  
EXEC CICS XCTL

**Anmerkung:** Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor. So müssen Sie beispielsweise den Eintritt in eine Schleife vermeiden, wenn Sie eine Anforderung für temporären Speicher über den Exit XTSEREQ ausgeben. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.

**Exit XTSEREQC**

Mithilfe des Exits XTSEREQC können Sie die entsprechende Antwort abfangen, nachdem eine API-Anforderung für temporären Speicher abgeschlossen worden ist.

**Zeitpunkt des Aufrufs**

Nach der Verarbeitung einer API-Anforderung für temporären Speicher durch CICS, vor der Rückkehr vom EXEC-Schnittstellenprogramm für temporären Speicher.

**Exitspezifische Parameter****UEPCLPS**

Adresse einer Kopie der Befehlsparameterliste. Siehe „Parameterstruktur der Befehlsebene“ auf Seite 256.

**UEPTQTOK**

Adresse eines 4-Byte-Bereichs, der verwendet werden kann, um für eine einzelne Anforderung für temporären Speicher Informationen zwischen XTSEREQ und XTSEREQC zu übergeben.



**UEPRCODE**

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

**UEPRES**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

**UEPRES2**

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

**UEPTSTOK**

Adresse eines 4-Byte-Tokens, das verwendet werden kann, um innerhalb derselben Task Informationen zwischen aufeinanderfolgenden Anforderungen für temporären Speicher zu übergeben (beispielsweise zwischen aufeinanderfolgenden Aufrufen des Exits XTSEREQC).

**UEPRECUR**

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

**UEPRSRCE**

Adresse einer 8 Zeichen langen Kopie des EIB-Ressourcenwerts, EIBRSRCE.

**UEP\_TS\_REMOTE\_SYSTEM**

Wenn die Anforderung an eine ferne Region gesendet werden soll, ist dies die Adresse eines Bereichs mit dem 4-Byte-Namen der fernen Region. Die ferne Region könnte möglicherweise durch die Option SYSID des Befehls, durch die Funktionsverlagerung oder durch das Workload-Management angegeben worden sein.

Wenn die Anforderung in der lokalen Region ausgeführt werden soll, ist dieser Parameter die Adresse eines 4-Byte-Bereichs mit Leerzeichen.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.

**API- und SPI-Befehle**

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

**EXEC CICS SHUTDOWN**

**EXEC CICS XCTL**

Sie können die Kopien von EIBRSRCE, EIBRCODE, EIBRESP und EIBRESP2, die Sie in der Parameterliste erhalten, aktualisieren. Wenn Sie die Werte aktualisieren, kopiert der temporäre Speicher die neuen Werte in den EIB des Anwendungsprogramms, nachdem XTSEREQC beendet wurde oder wenn Sie den Rückgabecode UERCBYP in XTSEREQ angeben.

Sie müssen gültige Antworten für den temporären Speicher festlegen. Sie müssen für die drei Codes EIBRCODE, EIBRESP und EIBRESP2 eine konsistente Gruppe von Werten festlegen, so wie sie auch vom temporären Speicher festgelegt würden, um eine gültige Ausführung zu beschreiben. Die Konsistenz der Werte für EIBRCODE, EIBRESP und EIBRESP2 wird von CICS nicht überprüft. Wenn EIBRCODE auf einen Wert ungleich null gesetzt wird und EIBRESP auf null gesetzt wird, überschreibt CICS diesen Wert für EIBRESP durch einen Wert, der ungleich null ist. Als Hilfestellung beim Festlegen von Werten für EIBRCODE, EIBRESP und EIBRESP2 können Sie die im Pseudobereich (DSECT) DFHTSUED angegebenen Werte heranziehen, die vom temporären Speicher verwendet werden.

**Anmerkung:** Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor, um keine Schleife zu verursachen. So sind Sie beispielsweise dafür verantwortlich, das Auftreten einer Schleife zu vermeiden, wenn eine Anforderung für temporären Speicher über den Exit XTSEREQC ausgegeben wird. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.

## Parameterstruktur der Befehlsebene

Die Parameterstruktur der Befehlsebene besteht aus einer Reihe von Adressen. Die erste Adresse verweist auf den EXEC-Schnittstellendeskriptor (EID), der aus einer Bitfolge besteht, die den Typ der Anforderung beschreibt und jedes mit der Anforderung angegebene Schlüsselwort identifiziert. Die übrigen Adressen verweisen auf Teile von Daten, die der Anforderung zugeordnet sind.

### Indikator für Ende der Parameterliste

Anhand des EXEC-Schnittstellendeskriptors können Sie den Typ der Anforderung sowie die angegebenen Schlüsselwörter ermitteln. Anhand der anderen Parameter in der Liste können Sie die Werte der Schlüsselwörter ermitteln. Sie können auch Werte von Schlüsselwörtern ändern, die in der Anforderung angegeben sind.

Das höchstwertige Bit wird für die letzte Adresse in der Parameterliste gesetzt, um anzuzeigen, dass es sich um die letzte Adresse in der Liste handelt. Bei der Rückkehr vom Benutzerexitprogramm sucht CICS in der Parameterliste nach dem höchstwertigen Bit, um den letzten Parameter ausfindig zu machen. Daher gilt: Wenn Sie die Länge der Parameterliste ändern, müssen Sie auch das höchstwertige Bit neu setzen, um so die aktuelle letzte Adresse zu markieren.

### Exitspezifischer Parameter UEPCLPS

Der exitspezifische Parameter UEPCLPS ist sowohl in Exit XTSEREQ als auch in Exit XTSEREQC enthalten. Es handelt sich um die Adresse der Parameterstruktur der Befehlsebene.

Die Parameterstruktur der Befehlsebene enthält 8 Adressen: TS\_ADDR0 bis TS\_ADDR7. Die Struktur wird im Pseudobereich (DSECT) TS\_ADDR\_LIST definiert, den Sie in Ihr Exitprogramm kopieren sollten, indem Sie die Anweisung COPY DFHTSUED einschließen.

Die Parameterliste der Befehlsebene ist wie folgt aufgebaut:

**Anmerkung:** Die Beziehungen zwischen Argumenten, Schlüsselwörtern, Datentypen und den Typen 'Eingabe/Ausgabe' werden in den folgenden Tabellen für die Befehle des temporären Speichers zusammengefasst:

*Tabelle 8. Beziehungen zwischen Argumenten, Schlüsselwörtern, Datentypen und den Typen 'Eingabe/Ausgabe' für die Befehle des temporären Speichers*

Befehl	Siehe
WRITEQ TS	Tabelle 9 auf Seite 261
READQ TS	Tabelle 10 auf Seite 261
DELETEQ TS	Tabelle 11 auf Seite 261

#### **TS\_ADDR0**

Adresse eines 9-Byte-Bereichs, der als EID (EXEC Interface Descriptor) bezeichnet wird und wie folgt aufgebaut ist:

- **TS\_GROUP**
- **TS\_FUNCT**
- **TS\_BITS1**
- **TS\_BITS2**
- **TS\_EIDOPT5**
- **TS\_EIDOPT6**
- **TS\_EIDOPT7**
- **TS\_EIDOPT8**

#### **TS\_GROUP**

Immer X'0A'; hierdurch wird angegeben, dass es sich um eine Anforderung für temporären Speicher handelt.

#### **TS\_FUNCT**

Ein Byte, das den Typ der Anforderung definiert:

- X'02' WRITEQ
- X'04' READQ
- X'06' DELETEQ

#### **TS\_BITS1**

Existenzbits zum Definieren der Argumente, die angegeben wurden. Um das Argument abzurufen, das einem Schlüsselwort zugeordnet ist, müssen Sie die entsprechende Adresse aus der Parameterstruktur der Befehlsebene verwenden. Bevor Sie diese Adresse verwenden, müssen Sie das zugehörige Existenzbit überprüfen. Wenn das Existenzbit inaktiviert ist, wurde das Argument in der Anforderung nicht angegeben, und die Adresse sollte nicht verwendet werden.

- X'80'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort QUEUE oder das Schlüsselwort QNAME enthält. Ist dieses Existenzbit gesetzt, dann ist **TS\_ADDR1** aussagekräftig.
- X'40'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort FROM, INTO oder SET enthält. Ist dieses Existenzbit gesetzt, dann ist **TS\_ADDR2** aussagekräftig.
- X'20'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort LENGTH enthält. Ist dieses Existenzbit gesetzt, dann ist **TS\_ADDR3** aussagekräftig.
- X'10'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort NUMITEMS enthält. Ist dieses Existenzbit gesetzt, dann ist **TS\_ADDR4** aussagekräftig.

**X'08'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort NUMITEMS oder ITEM enthält. Ist dieses Existenzbit gesetzt, dann ist **TS\_ADDR5** aussagekräftig.

**X'02'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort SYSID enthält. Ist dieses Existenzbit gesetzt, dann ist **TS\_ADDR7** aussagekräftig.

#### **TS\_BITS2**

Zwei Byte, die nicht vom temporären Speicher verwendet werden.

#### **TS\_EIDOPT5**

Gibt an, ob bestimmte Schlüsselwörter in der Anforderung angegeben wurden.

**X'80'** QNAME wurde angegeben (andernfalls QUEUE). Sie können dieses Bit in Ihrem Benutzerexit ändern, sofern gewünscht.

#### **TS\_EIDOPT6**

Ein Byte, das nicht vom temporären Speicher verwendet wird.

#### **TS\_EIDOPT7**

Gibt an, ob bestimmte Funktionen und/oder Schlüsselwörter in der Anforderung angegeben wurden.

**X'10'** WRITEQ NOSUSPEND wurde angegeben.

**X'80'** WRITEQ MAIN oder READQ ITEM wurde angegeben.

**X'04'** WRITEQ REWRITE oder READQ NUMITEMS wurde angegeben.

#### **TS\_EIDOPT8**

Gibt an, ob bestimmte Schlüsselwörter in der Anforderung angegeben wurden.

**X'80'** ITEM wurde angegeben (andernfalls NUMITEMS).

#### **TS\_ADDR1**

Adresse eines Bereichs mit dem 8-Byte-Namen aus QUEUE oder dem 16-Byte-Namen aus QNAME. Informationen dazu, welcher Name angewendet wird, finden Sie im Feld TS\_BITS2.

#### **TS\_ADDR2**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse einer 4-Byte-Adresse aus SET (wenn die Anforderung READQ ist und **TS\_EIDOPT5** anzeigt, dass dies SET ist).
- Adresse der Daten aus INTO (wenn die Anforderung READQ ist und **TS\_EIDOPT5** anzeigt, dass es sich nicht um SET handelt).
- Adresse der Daten aus FROM (wenn die Anforderung WRITEQ ist).

#### **TS\_ADDR3**

Adresse des Halbwortwerts für LENGTH (wenn die Anforderung READQ oder WRITEQ ist).

**Warnung:** Bei Anforderungen mit Angabe von INTO darf der Wert für LENGTH nicht in einen Wert geändert werden, der größer ist als der von der Anwendung angegebene Wert. Wird ein größerer Wert angegeben, führt dies zu einer Speicherüberlagerung in der Anwendung.

#### **TS\_ADDR4**

Adresse des Halbwortwerts für NUMITEMS (wenn die Anforderung READQ ist).

#### **TS\_ADDR5**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse des Halbwortwerts für NUMITEMS (wenn die Anforderung WRITEQ ist).
- Adresse des Halbwortwerts für ITEM (wenn die Anforderung READQ oder WRITEQ ist).

#### **TS\_ADDR6**

Adresse eines Werts, der nur zur internen Verwendung durch CICS bestimmt ist. Dieser Wert darf nicht verwendet werden.

#### **TS\_ADDR7**

Adresse eines Bereichs, der den Wert für SYSID enthält.

### **Ändern von Feldern in der Parameterstruktur der Befehlsebene**

Von den Feldern, die an temporären Speicher übergeben werden, werden einige als Eingabefelder für die Anforderung, andere als Ausgabefelder und wieder andere sowohl als Eingabe- als auch als Ausgabefelder verwendet. Die Methode, die Ihr Benutzerexitprogramm zum Ändern eines Felds verwendet, hängt von der Verwendung des betreffenden Felds ab.

Die folgenden Felder sind immer Eingabefelder:

- QUEUE|QNAME
- FROM
- SYSID

Die folgenden Felder sind immer Ausgabefelder:

- INTO
- NUMITEMS
- SET

LENGTH ist in Anforderungen vom Typ WRITEQ ein Eingabefeld und ein Ausgabefeld in Anforderungen vom Typ READQ mit Angabe von SET. In Anforderungen vom Typ READQ mit Angabe von INTO ist LENGTH sowohl ein Eingabe- als auch ein Ausgabefeld.

ITEM ist in Anforderungen vom Typ READQ sowie in Anforderungen vom Typ WRITEQ mit Angabe von REWRITE ein Eingabefeld. In Anforderungen vom Typ WRITEQ ohne Angabe von REWRITE ist ITEM sowohl ein Eingabe- als auch ein Ausgabefeld.

### **Ändern von Eingabefeldern**

Die richtige Methode zum Ändern eines Eingabefelds besteht darin, eine neue Kopie desselben zu erstellen und die Adresse in der Parameterliste der Befehlsebene so zu ändern, dass sie auf die neuen Daten verweist.

Ändern Sie ein Eingabefeld nicht durch Ändern der Daten, auf die von der Parameterliste der Befehlsebene verwiesen wird. Dies würde zu einer Beschädigung von Speicher führen, der zum Anwendungsprogramm gehört; dies wiederum würde zu einem Fehler führen, wenn das Programm versucht, das Feld wiederzuverwenden.

## Ändern von Ausgabefeldern

Das unter „Ändern von Eingabefeldern“ auf Seite 259 beschriebene Verfahren ist für das Ändern von Ausgabefeldern nicht geeignet. (Die Ergebnisse würden an den neuen Bereich und nicht an den Bereich der Anwendung zurückgegeben und wären daher für die Anwendung nicht sichtbar.)

Ein Ausgabefeld wird geändert, indem die Daten geändert werden, auf die von der Parameterliste der Befehlsebene verwiesen wird. Bei einem Ausgabefeld können Sie die vorhandenen Anwendungsdaten ändern, da die Anwendung ohnehin erwartet, dass das Feld geändert wird.

## Ändern von Feldern für Eingabe und Ausgabe

Ein Beispiel eines Felds, das sowohl für Eingabe als auch für Ausgabe verwendet wird, ist das Längensfeld LENGTH in einer Anforderung vom Typ READQ mit Angabe von INTO. Derartige Felder können in derselben Weise wie Ausgabefelder behandelt werden, und sie werden auch als solche angesehen.

## Ändern des EXEC-Schnittstellendesktors (EID)

Es ist nicht möglich, den EXEC-Schnittstellendesktors (EXEC Interface Descriptor, EID) zu ändern, um bedeutende Änderungen an Anforderungen vorzunehmen. So ist es beispielsweise nicht möglich, eine Anforderung vom Typ READQ in eine Anforderung vom Typ WRITEQ zu ändern. Es können jedoch geringfügige Änderungen an Anforderungen vorgenommen werden.

In der nachstehenden Liste werden diejenigen Bits im EID aufgeführt, die geändert werden können. Versuche, andere Bestandteile des EID zu ändern, werden ignoriert.

### TS\_BITS1

X'02' Existenzbit für SYSID.

### TS\_EIDOPT7

Benutzerexitprogramme am Exit XTSEREQ können für alle WRITEQ TS-Befehle die folgenden Existenzbits aktivieren bzw. inaktivieren:

X'10' Existenzbit für NOSUSPEND.

X'08' Existenzbit für MAIN.

Der EID wird vor der Rückkehr zum Anwendungsprogramm auf seinen ursprünglichen Wert zurückgesetzt. Dies bedeutet, dass Änderungen, die am EID vorgenommen werden, nur für die Dauer der Anforderung für temporären Speicher (TS) beibehalten werden.

**Anmerkung:** Das Benutzerexitprogramm wird zwar daran gehindert, bedeutende Änderungen am EID vorzunehmen. Dennoch müssen Sie auch bei den geringfügigen Änderungen, die **zulässig sind**, sehr sorgfältig vorgehen.

## Verwendung des Task-Tokens UEPTSTOK

Das Task-Token UEPTSTOK stellt die Adresse eines 4-Byte-Bereichs bereit, mit dem Sie Informationen zwischen aufeinanderfolgenden temporären Speicheranforderungen in derselben Task übergeben können.

So können Sie UEPTSTOK beispielsweise verwenden, um Informationen zwischen aufeinanderfolgenden Aufrufen des Exits XTSEREQ zu übergeben. Im Gegensatz dazu kann das Token UEPTQ TOK nur für die Dauer einer einzelnen temporären Speicheranforderung verwendet werden, da sein Inhalt am Ende der Anforderung möglicherweise gelöscht werden kann.

*Tabelle 9. WRITEQ TS: Benutzerargumente und zugehörige Schlüsselwörter, Datentypen und Typen 'Eingabe/Ausgabe'*

Argument	Schlüsselwort	Datentyp	Eingabe/Ausgabe
Arg1	QUEUE	CHAR(8)	Eingabe
Arg1	QNAME	CHAR(16)	Eingabe
Arg2	FROM	DATA-AREA	Eingabe
Arg3	LENGTH	BIN(15)	Eingabe
Arg4	*	*	*
Arg5	ITEM	BIN(15)	Eingabe/ Ausgabe
Arg5	NUMITEMS	BIN(15)	Ausgabe
Arg6	*	*	*
Arg7	SYSID	CHAR(4)	Eingabe

**Anmerkung:** Die verschiedenen Verwendungen von Arg5 werden angezeigt, da Arg5 von den Schlüsselwörtern ITEM und NUMITEMS verwendet wird, bei denen es sich um Alternativen handelt, wobei das Argument für das Schlüsselwort ITEM als Eingabefeld fungiert, wenn REWRITE angegeben ist.

*Tabelle 10. READQ TS: Benutzerargumente und zugehörige Schlüsselwörter, Datentypen und Typen 'Eingabe/Ausgabe'*

Argument	Schlüsselwort	Datentyp	Eingabe/Ausgabe
Arg1	QUEUE	CHAR(8)	Eingabe
Arg1	QNAME	CHAR(16)	Eingabe
Arg2	SET	DATA-AREA, PTR	Ausgabe
Arg2	INTO	DATA-AREA	Ausgabe
Arg3	LENGTH	BIN(15)	Eingabe/ Ausgabe
Arg4	NUMITEMS	BIN(15)	Ausgabe
Arg5	ITEM	BIN(15)	Eingabe
Arg6	*	*	
Arg7	SYSID	CHAR(4)	Eingabe

*Tabelle 11. DELETEQ TS: Benutzerargumente und zugehörige Schlüsselwörter, Datentypen und Typen 'Eingabe/Ausgabe'*

Argument	Schlüsselwort	Datentyp	Eingabe/Ausgabe
Arg1	QUEUE	CHAR(8)	Eingabe
Arg1	QNAME	CHAR(16)	Eingabe
Arg2	*	*	*
Arg3	*	*	*
Arg4	*	*	*
Arg5	*	*	*
Arg6	*	*	*
Arg7	SYSID	CHAR(4)	Eingabe

## Ändern von Benutzerargumenten

Benutzerexitprogramme können Benutzerargumente wie folgt ändern:

Für Eingabeargumente sollte das Benutzerexitprogramm ausreichend Speicher abrufen, um das geänderte Argument zu speichern, diesen Speicher für den erforderlichen Wert einrichten und den zugehörigen Verweis in der Parameterliste auf die Adresse des neu angeforderten Bereichs setzen.

Bei Ausgabeargumenten sowie bei Eingabe-/Ausgabeargumenten kann das Benutzerexitprogramm das betreffende Argument vor Ort aktualisieren, da der Speicherbereich in der Anwendung durch eine Variable dargestellt wird, an die erwartungsgemäß von CICS ein entsprechender Wert übertragen wird.

### Anmerkung:

1. Änderungen an Argumentwerten werden von CICS nicht überprüft. Sämtliche Änderungen müssen demzufolge von dem Benutzerexitprogramm verifiziert werden, das die Änderungen vornimmt.
2. Das Ändern von Ausgabeargumenten durch XTSEREQ sowie das Ändern von Eingabeargumenten durch XTSEREQC ist nicht empfehlenswert.

## Hinzufügen von Benutzerargumenten

Globale Benutzerexitprogramme können Argumente hinzufügen, die dem Schlüsselwort SYSID zugeordnet sind. Sie müssen sicherstellen, dass die Argumente, die Sie in Ihren Exitprogrammen angeben oder ändern, gültig sind.

Unter der Voraussetzung, dass das hinzuzufügende Argument bisher nicht vorhanden ist, muss das Exitprogramm folgende Aktionen ausführen:

1. Abrufen von Speicher für das hinzuzufügende Argument.
2. Initialisieren des Speichers anhand des erforderlichen Werts.
3. Auswählen und Einstellen des entsprechenden Verweises in der Parameterliste.
4. Auswählen und Einstellen des Existenzbits des entsprechenden Arguments im EXEC-Schnittstellendeskriptor (EXEC Interface Descriptor, EID).
5. Ändern der Parameterliste zur Berücksichtigung des neuen Indikators für das Ende der Liste.

## Entfernen von Benutzerargumenten

Benutzerexitprogramme können Argumente entfernen, die dem Schlüsselwort SYSID zugeordnet sind. Hierbei liegt die vollständige Verantwortung für das Entfernen von Argumenten beim Programm.

Unter der Voraussetzung, dass das zu entfernende Argument vorhanden ist, muss das Exitprogramm folgende Aktionen ausführen:

1. Umschalten des Existenzbits des entsprechenden Arguments im EID auf '0'b.
2. Ändern der Parameterliste zur Berücksichtigung des neuen Indikators für das Ende der Liste.

## Beispielprogramm

CICS stellt ein Beispielprogramm namens DFH\$XTSE bereit, das veranschaulicht, wie Anforderungen für temporären Speicher (TS) geändert werden können. Dieses Beispielprogramm wird nur als Softcopyliste (nicht als Quellcodedatei) bereitgestellt.



---

## Exit XALCAID für das Terminalzuordnungsprogramm

XALCAID wird gesteuert, wenn ein automatischer Initialisierungsdeskriptor (AID) mit Daten entweder durch eine CEMT-Transaktion, die den Befehl SET TERMINAL oder SET CONNECTION ausführt, oder während der Neuinstallation eines Terminals oder einer Verbindung abgebrochen wird.

XALCAID wird nur dann aufgerufen, wenn dem automatischen Initialisierungsdeskriptor (AID) Daten zugeordnet sind.

### Zeitpunkt des Aufrufs

Sobald ein AID mit Daten abgebrochen wird.

**Anmerkung:** Der Exit kann den Abbruch der Anforderung nicht verhindern.

### Exitspezifische Parameter

#### UEPALTSD

Adresse eines 4-Byte-Felds mit der symbolischen ID der Transaktion, die von dieser Anforderung gestartet werden sollte.

#### UEPALTRM

Adresse eines 4-Byte-Felds mit der ID des Terminals oder der Verbindung, an das bzw. die die Anforderung übertragen wurde.

#### UEPALDAT

Adresse eines Speicherbereichs mit den Daten, die in der Option FROM angegeben wurden; in einem der beiden folgenden Fälle enthält der Speicherbereich jedoch hexadezimale Nullen:

- Der AID wurde von einer START-Anforderung ohne die Option FROM erstellt.
- Der AID ist einem Kanal zugeordnet (in diesem Fall wird das Feld, auf das von UEPALCHN verwiesen wird, auf einen anderen Namen als Leerzeichen gesetzt).

#### UEPALLEN

Adresse eines Vollwort-Binärfelds mit der Länge der Daten aus FROM; in einem der beiden folgenden Fälle enthält das Feld jedoch hexadezimale Nullen:

- Der AID wurde von einer START-Anforderung ohne die Option FROM erstellt.
- Der AID ist einem Kanal zugeordnet (in diesem Fall wird das Feld, auf das von UEPALCHN verwiesen wird, auf einen anderen Namen als Leerzeichen gesetzt).

#### UEPALRQD

Adresse eines 8-Byte-Felds mit dem Wert der Anforderungs-ID (REQID), die den Daten aus FROM zugeordnet ist. Die Daten wurden in einer temporären Speicherwarteschlange mit diesem Namen gespeichert. Dieser Wert wurde entweder explizit mit der Option REQID im START-Befehl angegeben oder intern von CICS erstellt.

#### UEPALQUE

Adresse eines 8-Byte-Felds mit dem Wert, der in der Option QUEUE im START-Befehl angegeben wurde, oder mit hexadezimalen Nullen, wenn QUEUE nicht angegeben wurde.

#### UEPALRTE

Adresse eines 4-Byte-Felds mit dem Wert, der in der Option RTER-

MID im START-Befehl angegeben wurde, oder mit hexadezimalen Nullen, wenn RTERMID nicht angegeben wurde.

#### **UEPALRTA**

Adresse eines 4-Byte-Felds mit dem Wert, der in der Option RTRANSID im START-Befehl angegeben wurde, oder mit hexadezimalen Nullen, wenn RTRANSID nicht angegeben wurde.

#### **UEPALFMH**

Adresse eines 1-Byte-Felds, das den Wert X'FF' enthält, wenn die Daten FMHs enthalten, wie durch die Option FMH im zugeordneten START-Befehl angegeben; andernfalls enthält das Feld hexadezimale Nullen.

#### **UEPALSTC**

Adresse eines 2-Byte-Felds mit dem Startcode. Bei FEPI-Starts lautet dieser Startcode "SZ"; andernfalls lautet der Startcode "SD".

#### **UEPALCHN**

Adresse eines 16-Byte-Felds mit dem Namen des Kanals, der dem AID zugeordnet ist. Ist dem AID kein Kanal zugeordnet, wird dieses Feld auf Leerzeichen gesetzt.

### **Rückgabecodes**

#### **UERCNORM**

Es werden keine anderen Rückgabecodes angegeben. Der Wert des Rückgabecodes wird nicht überprüft.

### **XPI-Aufrufe**

Folgende Aufrufe können verwendet werden:

- INQ\_APPLICATION\_DATA
- INQUIRE\_SYSTEM

Es dürfen keine anderen XPI-Aufrufe verwendet werden.

### **API- und SPI-Befehle**

Es können keine EXEC CICS-Befehle verwendet werden.

**Anmerkung:** Der Exit XALTENF zur Verarbeitung der Bedingung "Terminal unbekannt" wird ebenfalls über das Terminalzuordnungsprogramm aufgerufen. XALTENF wird unter „Exits XALTENF und XICTENF für die Bedingung "Terminal unbekannt"“ auf Seite 266 beschrieben.

---

## **Exits für das Terminalsteuerprogramm (XTCIN, XTCOUT, XTCATT)**

Diese Exits werden entweder vor E/A-Ereignissen für sequenzielle Einheiten oder vor dem Anhängen von Tasks aufgerufen.

### **Exit XTCIN**

#### **Zeitpunkt des Aufrufs**

Nach einem Eingabeereignis für eine sequenzielle Einheit.

#### **Exitspezifische Parameter**

##### **UEPTCTTE**

Adresse des Terminaleintrags in der Terminalsteuertabelle (TCTTE). Der TCTTE kann mithilfe des Pseudobereichs (DSECT) DFHTCTTE zugeordnet werden.

#### **UEPTIOA**

Adresse des Terminal-E/A-Bereichs (TIOA). Das Exitprogramm sollte die Adresse nicht ändern. Der TIOA kann mithilfe des Pseudobereichs (DSECT) DFHTIOA zugeordnet werden. Bei den Feldern TIOASAL und TIOASCA handelt es sich jedoch **nicht** um Programmierschnittstellen.

#### **UEPTCTLE**

Adresse des Leitungseintrags in der Terminalsteuertabelle (TCTLE). Der TCTLE kann mithilfe des Pseudobereichs (DSECT) DFHTCTLE zugeordnet werden.

#### **Rückgabecodes**

##### **UERCNORM**

Verarbeitung fortsetzen.

#### **XPI-Aufrufe**

Es können alle verwendet werden. Hierbei ist jedoch zu beachten, dass Sie keinen GETMAIN-Aufruf verwenden können, um Terminalklassenspeicher zur Verwendung als Ersatz für den Terminal-E/A-Bereich (TIOA) abzurufen.

## **Exit XTCOUT**

#### **Zeitpunkt des Aufrufs**

Vor einem Ausgabeereignis für eine sequenzielle Einheit.

#### **Exitspezifische Parameter**

##### **UEPTCTTE**

Adresse des Terminaleintrags in der Terminalsteuertabelle (TCTTE). Der TCTTE kann mithilfe des Pseudobereichs (DSECT) DFHTCTTE zugeordnet werden.

##### **UEPTIOA**

Adresse des Terminal-E/A-Bereichs (TIOA). Das Exitprogramm sollte die Adresse nicht ändern. Der TIOA kann mithilfe des Pseudobereichs (DSECT) DFHTIOA zugeordnet werden. Bei den Feldern TIOASAL und TIOASCA handelt es sich jedoch **nicht** um Programmierschnittstellen.

##### **UEPTCTLE**

Adresse des Leitungseintrags in der Terminalsteuertabelle (TCTLE). Der TCTLE kann mithilfe des Pseudobereichs (DSECT) DFHTCTLE zugeordnet werden.

#### **Rückgabecodes**

##### **UERCNORM**

Verarbeitung fortsetzen.

#### **XPI-Aufrufe**

Es können alle verwendet werden. Hierbei ist jedoch zu beachten, dass Sie keinen GETMAIN-Aufruf verwenden können, um Terminalklassenspeicher zur Verwendung als Ersatz für den Terminal-E/A-Bereich (TIOA) abzurufen.

## **Exit XTCATT**

#### **Zeitpunkt des Aufrufs**

Vor dem Anhängen einer Task.

## Exitspezifische Parameter

### UEPTCTTE

Adresse des Terminaleintrags in der Terminalsteuertabelle (TCTTE). Der TCTTE kann mithilfe des Pseudobereichs (DSECT) DFHTCTTE zugeordnet werden.

### UEPTIOA

Adresse des Terminal-E/A-Bereichs (TIOA). Der TIOA kann mithilfe des Pseudobereichs (DSECT) DFHTIOA zugeordnet werden. Bei den Feldern TIOASAL und TIOASCA handelt es sich jedoch **nicht** um Programmierschnittstellen.

### UEPTCTLE

Adresse des Leitungseintrags in der Terminalsteuertabelle (TCTLE). Der TCTLE kann mithilfe des Pseudobereichs (DSECT) DFHTCTLE zugeordnet werden.

### UEPTRAN

Adresse der 4-Byte-Transaktions-ID.

## Rückgabecodes

### UERCNORM

Verarbeitung fortsetzen.

## XPI-Aufrufe

Es können alle verwendet werden.

---

## Exits XALTENF und XICTENF für die Bedingung "Terminal unbekannt"

Die Bedingung "Terminal unbekannt" kann auftreten, wenn miteinander kommunizierende CICS-Regionen sowohl Terminaldefinitionen vom Typ SHIPPABLE als auch automatische Transaktionsinitialisierung (ATI) verwenden. Die Wahrscheinlichkeit eines Auftretens dieser Bedingung ist insbesondere dann gegeben, wenn die Funktion für automatische Installation verwendet wird.

### Attribut SHIPPABLE

Für Terminals, die mit dem Attribut SHIPPABLE in einer Terminalverwaltungsregion (Terminal-Owning Region, TOR) definiert sind, ist keine Definition in einer verbundenen Anwendungsverwaltungsregion (Application-Owning Region, AOR) erforderlich. Sofern dies für die Unterstützung von Transaktionsrouting erforderlich ist, überträgt CICS eine Kopie der Definition von der TOR an die AOR. Weitere Informationen finden Sie unter Shipping terminal and connection definitions.

### Automatische Transaktionsinitialisierung (ATI)

ATI tritt auf, wenn eine intern generierte Anforderung zur Initialisierung einer Transaktion führt. Dies ist beispielsweise in folgenden Situationen der Fall:

- Eine Anwendung gibt einen Befehl vom Typ EXEC CICS START aus. ODER
- Die Auslöserebene für transiente Daten ist erreicht.

ATI-Anforderungen werden von zwei CICS-Modulen verarbeitet:

Das **Intervallsteuerprogramm** verarbeitet einen Startbefehl (START), überprüft, ob das Terminal im lokalen System bekannt ist, und ruft (nach Ablauf eines eventuell angegebenen Zeitintervalls für den Startbefehl) das Terminalzuordnungsprogramm auf.

Das **Terminalzuordnungsprogramm** wird durch das Intervallsteuerprogramm oder durch den Auslösemechanismus für transiente Daten aufgerufen und

überprüft, ob das Terminal im lokalen System bekannt ist. Handelt es sich bei dem angeforderten Terminal um ein fernes Terminal, überträgt das Terminalzuordnungsprogramm eine ATI-Anforderung an das ferne System, das daraufhin das Transaktionsrouting zurück zum lokalen System initialisiert.

Information für den Benutzer zum Thema ATI finden Sie unter Traditional routing of transactions started by ATI.

#### **Bedingung "Terminal unbekannt"**

Die Bedingung "Terminal unbekannt" tritt ein, wenn eine ATI-Anforderung für ein Terminal ausgegeben wird, das in der Region nicht bekannt ist. Eine ATI-Anforderung kann in der AOR für ein Terminal vom Typ SHIPPABLE auftreten, bevor ein Transaktionsrouting für das Terminal stattgefunden hat und somit bevor die Definition des Terminals von der TOR an die AOR übertragen werden konnte.

Wenn die Bedingung "Terminal unbekannt" auftritt, wird die Transaktionsinitialisierungsanforderung sowohl vom Intervallsteuerprogramm als auch vom Terminalzuordnungsprogramm als TERMIDERR zurückgewiesen.

## **Die Exits**

Zur Verarbeitung der Bedingung "Terminal unbekannt" stellt CICS die folgenden globalen Benutzerexits im Intervallsteuerprogramm bzw. im Terminalzuordnungsprogramm bereit:

#### **XICTENF**

Im Intervallsteuerprogramm.

#### **XALTENF**

Im Terminalzuordnungsprogramm.

Der Exit XICTENF wird von CICS gesteuert, wenn die Bedingung "Terminal unbekannt" eintritt, nachdem das Intervallsteuerprogramm durch einen Befehl vom Typ EXEC CICS START aufgerufen worden ist. Der Exit XALTENF wird von CICS gesteuert, wenn die Bedingung "Terminal unbekannt" eintritt, nachdem das Terminalzuordnungsprogramm durch die Auslöseebene für transiente Daten oder durch das Intervallsteuerprogramm aufgerufen worden ist. Hierbei ist zu beachten, dass ein Befehl vom Typ EXEC CICS START dazu führen kann, dass beide Exits aufgerufen werden.

Das Exitprogramm muss angeben, ob das Terminal auf einem anderen System vorhanden ist, und wenn ja, auf welchem. Um diese Informationen zu ermitteln, werden von CICS Daten an das Exitprogramm übergeben. Sie können an beiden Exitpunkten dasselbe Exitprogramm verwenden. CICS stellt ein Beispiexitprogramm namens DFHXTENF bereit, das an beiden Exits verwendet werden kann und in der Lage ist, ohne Änderungen eine Reihe von typischen Situationen zu verarbeiten.

Die Exits sind so konzipiert, dass sie Bedingungen vom Typ "Terminal unbekannt" verarbeiten, die in anderen CICS-Regionen als der Terminalverwaltungsregion (TOR) auftreten. Bei einem Paar aus Terminalverwaltungsregion (TOR) und Anwendungsverwaltungsregion (AOR) müssen Sie das Exitprogramm in der AOR aktivieren. Die Exits können keine Bedingungen vom Typ "Terminal unbekannt" in der TOR verarbeiten, sodass das Exitprogramm dort normalerweise nicht aktiviert werden sollte. Wenn jedoch mehr als eine TOR vorhanden ist, müssen Sie möglicherweise das Exitprogramm in jeder einzelnen TOR aktivieren, um Anforderungen für Terminals zu verarbeiten, die von anderen TORs verwaltet werden. In diesem Fall muss das Exitprogramm Terminals erkennen, die von den entsprechenden

Systemen verwaltet werden sollten, und die betreffenden Anforderungen zurückweisen (UERCTEUN). Obwohl der Exit so viele Daten wie möglich bereitstellt, hängt die Logik Ihres Programms vollständig von Ihrem Systemdesign ab. Eine einfache Lösung für den komplexesten Fall bestünde darin, den Namen jedes Terminals so festzulegen, dass er den Netznamen oder die System-ID der entsprechenden Verwaltungsregion widerspiegelt.

### **Vom Exit zurückgegebene Daten**

Das Exitprogramm muss in Register 15 wie folgt einen Rückgabecode festlegen:

#### **UERCTEUN**

Terminal ist nicht vorhanden.

#### **UERCNETN**

Netzname der Terminalverwaltungsregion (TOR) zurückgegeben.

#### **UERCYSI**

System-ID der Terminalverwaltungsregion (TOR) zurückgegeben.

Bei den Rückgabecodes UERCNETN und UERCYSI muss das Exitprogramm den Netznamen bzw. die System-ID der Terminalverwaltungsregion (TOR) in die Felder UEPxxNTO bzw. UEPxxSYO stellen (wobei xx für AL oder IC steht).

Wenn die Terminalverwaltungsregion Member einer generischen Ressource von z/OS Communications Server ist, sollte das Exitprogramm den Netznamen des Terminals in das Feld UEPxxNNO stellen. Informationen zur Verwendung von ATI mit generischen Ressourcen von z/OS Communications Server finden Sie unter *Using ATI with generic resources*.

## **Exit XALTENF**

Der Exit XALTENF wird vom Terminalzuordnungsprogramm aufgerufen, wenn das Terminal, das eine ATI-Anforderung von transienten Daten oder der Intervallsteuerung benötigt, in diesem System unbekannt ist. Das Exitprogramm soll einen Rückgabecode liefern, der angibt, ob das Terminal auf einem anderen verbundenen CICS-System vorhanden ist, und wenn ja, auf welchem.

### **Exitspezifische Parameter**

#### **UEPALEVT**

2-Byte-Adresse mit dem Anforderungstyp. Die Typen werden durch folgende Werte entsprechend angegeben:

#### **UEPALESD**

Befehl START mit Daten.

#### **UEPALES**

Befehl START ohne Daten.

#### **UEPALETD**

Auslöserebene für transiente Daten erreicht.

#### **UEPALTR**

1-Byte-Adresse mit Hinweis, ob die den Befehl START ausgebende Task über Transaktionsrouting gestartet wurde. Die Hinweise werden durch folgende Werte entsprechend angegeben:

#### **UEPALTY**

Ein START-Befehl wurde verarbeitet, und die den Befehl ausgebende Task wurde über Transaktionsrouting gestartet.

**UEPALTN**

Es wurde kein START-Befehl verarbeitet, **oder** es wurde ein START-Befehl verarbeitet, aber die den Befehl ausgebende Task wurde nicht über Transaktionsrouting gestartet.

**UEPALFS**

1-Byte-Adresse mit Hinweis, ob der Befehl START über Funktionen übertragen wurde. Die Hinweise werden durch folgende Werte entsprechend angegeben:

**UEPALFY**

Ein START-Befehl wurde verarbeitet, und dieser Befehl wurde über Funktionen übertragen.

**UEPALFN**

Es wurde kein START-Befehl verarbeitet, **oder** es wurde ein START-Befehl verarbeitet, aber dieser Befehl wurde nicht über Funktionen übertragen.

**UEPALTRN**

4-Byte-Adresse mit dem Namen der auszuführenden Transaktion.

**UEPALRTR**

4-Byte-Adresse mit dem Namen des Terminals, auf dem die Transaktion ausgeführt werden soll. (Wurde eine Auslöseebene für transiente Daten erreicht und in der Definition der Warteschlange mit transienten Daten ein System angegeben, dann enthält diese Adresse eine System-ID.)

**UEPALCTR**

4-Byte-Adresse, die für START-Befehle den Namen des aktuellen Terminals enthält, wenn der Befehl über Transaktionsrouting gestartet wurde, oder den Namen der Sitzung, wenn der Befehl über Funktionen übertragen wurde.

Bei anderen START-Befehlen und bei Auslöserereignissen für transiente Daten enthält das Feld, auf das verwiesen wird, Leerzeichen.

**UEPALNTI**

8-Byte-Adresse, die für START-Befehle, die über Funktionen übertragen wurden, den Netznamen des letzten Systems enthält, aus dem die Anforderung stammte.

Für START-Befehle, die in diesem System durch Transaktionsrouting an eine Task ausgegeben wurden, enthält die Adresse den Netznamen des letzten Systems, von dem aus die Task weitergeleitet wurde.

Bei anderen START-Befehlssituationen und bei Ereignissen auf Auslöseebene für transiente Daten enthält das Feld, auf das verwiesen wird, Leerzeichen.

**UEPALSXI**

4-Byte-Adresse mit der System-ID (SYSID) des in UEPALNTI angegebenen Netznamens, sofern UEPALNTI einen Netznamen enthält.

Wenn UEPALNTI keinen Netznamen enthält, ist das Feld, auf das verwiesen wird, leer.

**UEPALNTO**

8-Byte-Adresse mit dem Inhalt von UEPALNTI.

Wird der Rückgabecode UERCNETN gesetzt, muss das Exitprogramm in dieses Feld den Netznamen des Systems stellen, an das die ATI-Anforderung gesendet werden soll.

#### **UEPALSYO**

4-Byte-Adresse mit dem Inhalt von UEPALSYI.

Wird der Rückgabecode UERCSYSI gesetzt, muss das Exitprogramm in dieses Feld die System-ID des Systems stellen, an das die ATI-Anforderung gesendet werden soll.

#### **UEPALNNI**

Adresse eines 4-Byte-Eingabefelds mit dem Netznamen des Terminals, auf dem die Transaktion ausgeführt werden soll, sofern CICS diesen Namen kennt. Wenn CICS den Netznamen nicht kennt, enthält das adressierte Feld Leerzeichen.

#### **UEPALNNO**

Adresse eines 4-Byte-Eingabefelds/-Ausgabefelds, das bei Aufruf den Inhalt von UEPALNNI enthält. Das Exitprogramm kann dieses Feld verwenden, um den Netznamen des Terminals zu liefern, auf dem die Transaktion ausgeführt werden soll. Es ist wichtig, dass das Exitprogramm einen Terminalnetznamen liefert, wenn die Terminalverwaltungsregion (TOR), an die es die ATI-Anforderung überträgt, ein Member einer generischen Ressource von z/OS Communications Server ist.

### **Rückgabecodes**

#### **UERCTEUN**

Terminal unbekannt, Anforderung zurückweisen.

#### **UERCNETN**

Terminal bekannt, Netzname wird in UEPALNTO zurückgegeben.

#### **UERCSYSI**

Terminal bekannt, System-ID wird in UEPALSYO zurückgegeben.

### **XPI-Aufrufe**

Folgende Aufrufe können verwendet werden:

- INQ\_APPLICATION\_DATA.
- INQUIRE\_SYSTEM.

Es dürfen keine anderen XPI-Aufrufe verwendet werden.

### **Beispielexitprogramm**

DFHXTENF ist ein Beispielprogramm, das für die Exits XALTENF und XICTENF verwendet werden kann. Weitere Informationen zu DFHXTENF finden Sie unter Terminal-not-known sample exit program: DFHXTENF.

## **Exit XICTENF**

Der Exit XICTENF wird durch das Intervallsteuerprogramm aufgerufen, wenn das Terminal, das ein EXEC CICSSTART-Befehl benötigt, in diesem System unbekannt ist.

#### **Zeitpunkt des Aufrufs**

Durch das Intervallsteuerprogramm, wenn das Terminal, das ein EXEC CICSSTART-Befehl benötigt, in diesem System unbekannt ist. Das Exitpro-



gramm soll einen Rückgabecode liefern, der angibt, ob das Terminal auf einem anderen verbundenen CICS-System vorhanden ist, und wenn ja, auf welchem.

### **Exitspezifische Parameter**

#### **UEPICEVT**

2-Byte-Adresse mit dem Anforderungstyp. Die Typen werden durch folgende Werte entsprechend angegeben:

#### **UEPICESD**

Befehl START mit Daten.

#### **UEPICES**

Befehl START ohne Daten.

#### **UEPICTR**

1-Byte-Adresse mit Hinweis, ob die den Befehl START ausgebende Task über Transaktionsrouting gestartet wurde. Die Hinweise werden durch folgende Werte entsprechend angegeben:

#### **UEPICTY**

Ein START-Befehl wurde verarbeitet, und die den Befehl ausgebende Task wurde über Transaktionsrouting gestartet.

#### **UEPICTN**

Es wurde kein START-Befehl verarbeitet, **oder** es wurde ein START-Befehl verarbeitet, aber die den Befehl ausgebende Task wurde nicht über Transaktionsrouting gestartet.

#### **UEPICFS**

1-Byte-Adresse mit Hinweis, ob der Befehl START über Funktionen übertragen wurde. Die Hinweise werden durch folgende Werte entsprechend angegeben:

#### **UEPICFY**

Ein START-Befehl wurde verarbeitet, und dieser Befehl wurde über Funktionen übertragen.

#### **UEPICFN**

Es wurde kein START-Befehl verarbeitet, **oder** es wurde ein START-Befehl verarbeitet, aber dieser Befehl wurde nicht über Funktionen übertragen.

#### **UEPICTRN**

4-Byte-Adresse mit dem Namen der auszuführenden Transaktion.

#### **UEPICRTR**

4-Byte-Adresse mit dem Namen des Terminals, auf dem die Transaktion ausgeführt werden soll.

#### **UEPICCTR**

4-Byte-Adresse, die für START-Befehle den Namen des aktuellen Terminals enthält, wenn der Befehl über Transaktionsrouting gestartet wurde, oder den Namen der Sitzung, wenn der Befehl über Funktionen übertragen wurde.

Bei anderen START-Befehlen enthält das Feld, auf das verwiesen wird, Leerzeichen.

#### UEPICNTI

8-Byte-Adresse, die für START-Befehle, die über Funktionen übertragen wurden, den Netznamen des letzten Systems enthält, aus dem die Anforderung stammte.

Für START-Befehle, die in diesem System durch Transaktionsrouting an eine Task ausgegeben wurden, enthält die Adresse den Netznamen des letzten Systems, von dem aus die Task weitergeleitet wurde.

Bei anderen START-Befehlssituationen enthält das Feld, auf das verwiesen wird, Leerzeichen.

#### UEPICSYI

4-Byte-Adresse mit der System-ID (SYSID) des in UEPICNTI angegebenen Netznamens, sofern UEPICNTI einen Netznamen enthält.

Wenn UEPICNTI keinen Netznamen enthält, ist das Feld, auf das verwiesen wird, leer.

#### UEPICNTO

8-Byte-Adresse mit dem Inhalt von UEPICNTI.

**Wird der Rückgabecode UERCNETN gesetzt, muss das Exitprogramm in dieses Feld den Netznamen des Systems stellen, an das die ATI-Anforderung gesendet werden soll.**

#### UEPICSYO

4-Byte-Adresse mit dem Inhalt von UEPICSYI.

**Wird der Rückgabecode UERCYSYI gesetzt, muss das Exitprogramm in dieses Feld die System-ID des Systems stellen, an das die ATI-Anforderung gesendet werden soll.**

#### UEPICNNI

Adresse eines 4-Byte-Eingabefelds mit dem Netznamen des Terminals, auf dem die Transaktion ausgeführt werden soll, sofern CICS diesen Namen kennt. Wenn CICS den Netznamen nicht kennt, enthält das adressierte Feld Leerzeichen.

#### UEPICNNO

Adresse eines 4-Byte-Eingabefelds/-Ausgabefelds, das bei Aufruf den Inhalt von UEPICNNI enthält. Das Exitprogramm kann dieses Feld verwenden, um den Netznamen des Terminals zu liefern, auf dem die Transaktion ausgeführt werden soll. Es ist wichtig, dass das Exitprogramm einen Terminalnetznamen liefert, wenn die Terminalverwaltungsregion (TOR), an die es die ATI-Anforderung überträgt, ein Member einer generischen Ressource von z/OS Communications Server ist.

#### Rückgabecodes

##### UERCTEUN

Terminal unbekannt, Anforderung zurückweisen.

##### UERCNETN

Terminal bekannt, Netzname wird in UEPICNTO zurückgegeben.

##### UERCYSYI

Terminal bekannt, System-ID wird in UEPICSYO zurückgegeben.

##### UERCPURG

Task während des XPI-Aufrufs bereinigt.

### XPI-Aufrufe

Folgende Aufrufe dürfen nicht verwendet werden:

- ADD\_SUSPEND
- DELETE\_SUSPEND
- DEQUEUE
- ENQUEUE
- RESUME
- SUSPEND
- WAIT\_MVS

### Beispielexitprogramm

DFHXTENF ist ein Beispielprogramm, das für die Exits XALTENF und XICTENF verwendet werden kann. Weitere Informationen zu DFHXTENF finden Sie unter Terminal-not-known sample exit program: DFHXTENF.

---

## Exit XXMATT für die Transaktionsmanagerdomäne

Der Exit XXMATT wird beim Zuordnen (Anhängen) von Transaktionen aufgerufen und kann einige Attribute der Transaktion ändern, die zugeordnet werden soll.

Der Exit kann die Transaktions-ID in der Zuordnung für die betreffende Transaktion ändern, indem er das von UEPATPTI adressierte Feld ändert. Über diesen Exit können keine **EXEC CICS**-Befehle verwendet werden.

### Exitspezifische Parameter

#### UEPTRANID

Adresse der Transaktions-ID (siehe Anmerkungen).

#### UEPUSER

Adresse der Benutzer-ID, die der Transaktion zugeordnet ist, wenn es sich bei der aktuellen Task um eine Benutzertask handelt (siehe Anmerkungen).

#### UEPTERM

Adresse der Terminal-ID, die der Transaktion zugeordnet ist (sofern vorhanden); (siehe Anmerkungen).

#### UEPPROG

Adresse des Anwendungsprogrammnamens für diese Transaktion (sofern vorhanden); (siehe Anmerkungen).

#### UEPATPTI

Adresse eines 4-Byte-Felds mit der primären Transaktions-ID. Sie können die primäre Transaktions-ID ändern, indem Sie das adressierte Feld ändern.

#### UEPATOTI

Adresse der 4-Byte-Transaktions-ID in der Zuordnung. Der Wert X'00000000' gibt an, dass in der Zuordnung keine Transaktions-ID angegeben wurde.

#### UEPATTPL

Adresse eines Bereichs mit der Länge von 'TPName' in der Zuordnung. Der Wert null gibt an, dass in der Zuordnung kein TPName angegeben wurde.

**UEPATTPA**

Adresse eines Vollworts mit der Adresse von TPName in der Zuordnung. Der Wert für TPName in der Zuordnung kann eine Länge von 1 bis 64 Byte haben, wie durch UEPTTPL definiert.

**UEPATLOC**

Adresse eines 1-Byte-Felds, das angibt, ob die Transaktion gefunden wurde. Anmerkung: Wenn die Transaktion nicht gefunden wurde, aber die Systeminitialisierungsparameter DTRTRAN und DTRPGM angegeben sind, wird die in DTRTRAN angegebene Transaktion angehängt, und in CICS gilt die Transaktion als gefunden.

Mögliche Werte und ihre Entsprechungen:

**UEATFND**

Die Transaktion wurde gefunden.

**UEATNFND**

Die Transaktion wurde nicht gefunden.

**UEPATTST**

Adresse eines Transaktionsdefinitionsstatus aus 1 Byte. Die Definitionsstatus werden durch folgende Werte entsprechend angegeben:

**UEATENAB**

Die Transaktion ist aktiviert.

**UEATDISA**

Die Transaktion ist inaktiviert.

**UEPATTTK**

Adresse eines Doppelworts mit einem Transaktionstoken. Anmerkung: Für bestimmte XPI-Aufrufe des Transaktionsmanagers ist dieses Token erforderlich, um die Transaktion zu identifizieren, die angehängt wird.

**Rückgabecodes****UERCNORM**

Verarbeitung der Zuordnung fortsetzen.

**XPI-Aufrufe**

Der Benutzerexit kann Abfragen für die angehängte Transaktion durchführen, indem er das Transaktionstoken aus UEPATTTK als Eingabe im XPI-Aufruf XMIQ INQUIRE\_TRANSACTION verwendet.

Darüber hinaus kann der Exit auch die Gesamtpriorität sowie TCLASS festlegen. Hierfür wird der XPI-Aufruf XMIQ SET\_TRANSACTION verwendet.

Grundsätzlich können die meisten XPI-Aufrufe verwendet werden. Hierbei ist jedoch Vorsicht geboten, da dieser Exit normalerweise unter der TCP-Task aufgerufen wird. Daher ist es ratsam, keine XPI-Aufrufe auszugeben, die dazu führen könnten, dass die TCP-Task warten muss.

**Anmerkung:**

1. Die folgenden XPI-Aufrufe können nützlich sein, wenn es darum geht, Informationen abzurufen, die zum Ändern der Zuordnung einer Transaktion verwendet werden können:
  - INQUIRE\_TRANSACTION
  - INQUIRE\_MXT

- INQUIRE\_TCLASS
  - INQUIRE\_TRANDEF
  - INQUIRE\_SYSTEM
2. Die Felder UEPTRANID, UEPUSER, UEPTERM und UEPPROG kommen in vielen globalen Benutzerexitpunkten der Domäne vor und geben normalerweise Werte zurück, die der aktuellen Benutzertask zugeordnet sind. Bei XXMATT hingegen ist die Benutzertask, die angehängt wird, **nicht** die aktuelle Task, wenn der Exit aufgerufen wird. Bis zum Abschluss der Taskzuordnung handelt es sich bei der aktuellen Task um die CICS-Task, die die betreffende Zuordnung ausführt.
- Wenn die Task, die zugeordnet wird, für eine Task verwendet werden soll, die von einem START-Befehl für sofortige Ausführung gestartet wird (also von einem START-Befehl ohne Intervall), dann handelt es sich bei der aktuellen Task um die Task, die den START-Befehl ausgibt, und die Felder enthalten Werte, die dieser Task zugeordnet sind.

---

## Exits für das Programm für transiente Daten (XTDREQ, XTDIN, XTDOUT)

Der Exit XTDREQ fängt eine Anforderung für temporäre Daten vor der Anforderungsanalyse ab. Die Exits XTDOUT und XTDIN werden vor und nach dem Austausch von Daten mit QSAM oder VSAM aufgerufen.

Die CICS-Einrichtung für transiente Daten ist threadsicher, sodass CICS Anforderungen für transiente Daten in einem offenen Tasksteuerblock (TCB) verarbeiten kann. Anforderungen für transiente Daten sind auch dann threadsicher, wenn sie über Funktionen mittels einer IPIC-Verbindung an eine ferne Region übertragen werden. Um das Umschalten zwischen TCBs zu optimieren und die Leistungsvorteile der offenen Transaktionsumgebung nutzen zu können, müssen Programme, die an den Exits XTDREQ, XTDIN und XTDOUT ausgeführt werden, gemäß threadsicheren Standards codiert und für CICS als threadsicher definiert werden.

### Exit XTDREQ

Der Exit XTDREQ wird vor der Anforderungsanalyse aufgerufen.

#### Exitspezifische Parameter

##### UEPTDQUE

Adresse des 4-Byte-Namens der Warteschlange mit transienten Daten.

##### UEPTDTYP

Adresse des 1-Byte-Typs der Anforderung für transiente Daten.  
Mögliche Werte:

##### UEPTDPUT

PUT-Anforderung.

##### UEPTDGET

GET-Anforderung.

##### UEPTDPUR

PURGE-Anforderung.

#### Rückgabecodes

##### UERCNORM

Verarbeitung der transienten Daten fortsetzen.

**UERCTDOK**

Verarbeitung der transienten Daten beenden; Rückgabe von NORMAL an den Aufrufenden.

**UERCTDNA**

Verarbeitung der transienten Daten beenden; Rückgabe von NOTAUTH an den Aufrufenden.

**UERC PURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Folgende Aufrufe können verwendet werden:

- INQ\_APPLICATION\_DATA
- INQUIRE\_SYSTEM
- WAIT\_MVS

**Verwenden Sie keine anderen Aufrufe.**

## Exit XTDIN

Der Exit XTDIN wird aufgerufen, nachdem CICS Daten von QSAM (partitionsübergreifend) oder VSAM (partitionsintern) empfängt.

**Exitspezifische Parameter****UEPTDQUE**

Adresse des 4-Byte-Namens der Warteschlange mit transienten Daten.

**UEPTDAUD**

Adresse der nicht geänderten transienten Daten.

**UEPTDLUD**

Adresse der Vollwortlänge der nicht geänderten transienten Daten.

**UEPTDAMD**

Adresse der transienten Daten, die durch das Exitprogramm geändert wurden.

**UEPTDLMD**

Adresse der Vollwortlänge der transienten Daten, die durch das Exitprogramm geändert wurden.

**Rückgabecodes****UERC NORM**

Verarbeitung der transienten Daten fortsetzen.

**UERC PURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Folgende Aufrufe können verwendet werden:

- INQ\_APPLICATION\_DATA
- INQUIRE\_SYSTEM
- WAIT\_MVS

**Verwenden Sie keine anderen Aufrufe.**

## Exit XTDOUT

Der Exit XTDOUT wird aufgerufen, bevor CICS Daten an eine benutzerdefinierte Warteschlange mit transienten Daten unter QSAM (partitionsübergreifend) oder VSAM (partitionsintern) übergibt.

### Exitspezifische Parameter

#### UEPTDQUE

Adresse des 4-Byte-Namens der Warteschlange mit transienten Daten.

#### UEPTDAUD

Adresse der nicht geänderten transienten Daten.

#### UEPTDLUD

Adresse der Vollwortlänge der nicht geänderten transienten Daten.

#### UEPTDAMD

Adresse der transienten Daten, die durch das Exitprogramm geändert wurden.

#### UEPTDLMD

Adresse der Vollwortlänge der transienten Daten, die durch das Exitprogramm geändert wurden.

#### UEPTDNUM

Adresse des Vollworts, das die Anzahl der Elemente in der Liste enthält.

#### UEPTDCUR

Adresse des Vollworts, das die Nummer des aktuellen Elements enthält.

### Rückgabecodes

#### UERCNORM

Verarbeitung der transienten Daten fortsetzen.

#### UERCTDOK

Verarbeitung der transienten Daten beenden; Rückgabe von NORMAL an den Aufrufenden.

**Anmerkung:** Wenn Sie UERCTDOK zurückgeben, um die erste Zeile einer mehrzeiligen Nachricht zu unterdrücken, wird der Rest der Nachricht nicht an XTDOUT übergeben, sondern ebenfalls unterdrückt.

#### UERCPURG

Task während des XPI-Aufrufs bereinigt.

### XPI-Aufrufe

Folgende Aufrufe können verwendet werden:

- INQ\_APPLICATION\_DATA
- INQUIRE\_SYSTEM
- WAIT\_MVS

Verwenden Sie keine anderen Aufrufe.

---

## Exits XTDEREQ und XTDEREQC des EXEC-Schnittstellenprogramms für transiente Daten

Der Exit XTDEREQ fängt Anforderungen für transiente Daten ab, bevor entsprechende Aktionen im Rahmen der Verarbeitung transienter Daten ausgeführt werden. Der Exit XTDEREQC fängt Anforderungen für transiente Daten ab, nachdem die Verarbeitung der transienten Daten abgeschlossen worden ist.

Sie können den Exit XTDEREQ ändern, um die folgenden Tasks auszuführen:

- Analysieren der Anforderung zwecks Ermittlung ihres Typs sowie der angegebenen Schlüsselwörter und zugehörigen Werte.
- Ändern der in der Anforderung angegebenen Werte vor Ausführung des Befehls.
- Festlegen von Rückgabecodes zum Angeben einer der beiden folgenden Anweisungen:
  - CICS soll die Verarbeitung der Anforderung einschließlich aller von Ihnen vorgenommenen Änderungen fortsetzen.
  - CICS soll die Anforderung umgehen. Wenn Sie diesen Rückgabecode festlegen, müssen Sie auch Rückgabecodes für den EXEC-Schnittstellenblock (EIB) so festlegen, als hätten Sie die Anforderung selbst verarbeitet.

Sie können den Exit XTDEREQC ändern, um die folgenden Tasks auszuführen:

- Analysieren der Anforderung zwecks Ermittlung ihres Typs sowie der angegebenen Schlüsselwörter und zugehörigen Werte.
- Festlegen von Rückgabecodes für den EXEC-Schnittstellenblock (EIB).

Die CICS-Einrichtung für transiente Daten ist threadsicher, sodass CICS Anforderungen für transiente Daten in einem offenen Tasksteuerblock (TCB) verarbeiten kann. Anforderungen für transiente Daten sind auch dann threadsicher, wenn sie über Funktionen mittels einer IPIC-Verbindung an eine ferne Region übertragen werden. Um das Umschalten zwischen TCBs zu optimieren und die Leistungsvorteile der offenen Transaktionsumgebung nutzen zu können, müssen Programme, die an den Exits XTDEREQ und XTDEREQC ausgeführt werden, gemäß threadsicheren Standards codiert und für CICS als threadsicher definiert werden.

An beide Exits werden acht Parameter wie folgt übergeben:

- Adresse der Parameterstruktur der Befehlsebene.
- Adresse eines Tokens (UEPTDTOK) zur Übergabe von 4 Byte Daten von XTDEREQ an XTDEREQC.
- Adressen von Kopien von vier Rückgabecodes und Ressourceninformationen vom EXEC-Schnittstellenblock (EIB).
- Adresse eines Tokens (UEPTSTOK), das während der gesamten Lebensdauer einer Task gültig ist.
- Adresse eines Rekursionszählers (UEPRECUR) für die Exits.

### Beispielprogramm

CICS stellt ein Beispielprogramm namens DFH\$XTSE bereit, das veranschaulicht, wie Felder in der Parameterstruktur der Befehlsebene, die an Exits für die EXEC-Schnittstellenexits übergeben wird, geändert werden können. Dieses Beispielprogramm wird nur als Softcopyliste und nicht als Quellcodedatei bereitgestellt.



## Exit XTDEREQ

Der Exit XTDEREQ wird aufgerufen, bevor CICS eine API-Anforderung für transiente Daten verarbeitet.

### Exitspezifische Parameter

#### UEPCLPS

Adresse der Parameterstruktur der Befehlsebene. Siehe „Exitspezifischer Parameter UEPCLPS“ auf Seite 283.

#### UEPTDTOK

Adresse des 4-Byte-Tokens, das an XTDEREQC übergeben werden soll. UEPTDTOK ermöglicht es Ihnen beispielsweise, einen Arbeitsbereich an den Exit XTDEREQC zu übergeben.

#### UEPRCODE

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

#### UEPRES

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

#### UEPRES2

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

#### UEPTSTOK

Adresse eines 4-Byte-Tokens, das während des gesamten Lebenszyklus einer Task gültig ist. Siehe Using the task token UEPTSTOK.

#### UEPRECUR

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

#### UEPRSRCE

Adresse einer 8 Zeichen langen Kopie des EIB-Ressourcenwerts, EIBRSRCE.

### Rückgabecodes

#### UERCNORM

Verarbeitung fortsetzen.

#### UERCBYP

Diese Anforderung wird vom EXEC-Schnittstellenprogramm für transiente Daten ignoriert.

#### UERCPURG

Task während des XPI-Aufrufs bereinigt.

### XPI-Aufrufe

Es können alle verwendet werden.

Wenngleich der Exit die Verwendung der XPI-Aufrufe GETMAIN und FREEMAIN zulässt, wird empfohlen, stattdessen die Befehle EXEC CICS GETMAIN und EXEC CICS FREEMAIN zu verwenden.

### API- und SPI-Befehle

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:  
EXEC CICS SHUTDOWN

**Anmerkung:** Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor. So müssen Sie beispielsweise den Eintritt in eine Schleife vermeiden, wenn Sie eine Anforderung für transiente Daten über den Exit XTDEREQ ausgeben. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.

## Exit XTDEREQC

Der Exit XTDEREQC wird nach Abschluss einer API-Anforderung für transiente Daten und vor der Rückkehr vom EXEC-Schnittstellenprogramm für transiente Daten aufgerufen.

### Exitspezifische Parameter

#### UEPCLPS

Adresse der Parameterstruktur der Befehlsebene. Siehe „Exitspezifischer Parameter UEPCLPS“ auf Seite 283.

#### UEPTDTOK

Adresse des 4-Byte-Tokens, das an XTDEREQC übergeben werden soll. Dies ermöglicht es Ihnen beispielsweise, einen Arbeitsbereich an den Exit XTDEREQC zu übergeben.

#### UEPRCODE

Adresse einer 6 Byte langen hexadezimalen Kopie des EIB-Rückgabecodes EIBRCODE. Details zu EIB-Rückgabecodes finden Sie unter EIB fields.

#### UEPRES P

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP.

#### UEPRES P2

Adresse einer 4 Byte langen binären Kopie des EIB-Antwortcodes EIBRESP2.

#### UEPTSTOK

Adresse eines 4-Byte-Tokens, das während des gesamten Lebenszyklus einer Task gültig ist. Siehe Using the task token UEPTSTOK.

#### UEPRECUR

Adresse eines Halbwortzählers für Rekursionen. Der Zähler wird auf 0 gesetzt, wenn der Exit erstmals aufgerufen wird, und dann bei jedem rekursiven Aufruf erhöht.

#### UEPRSRCE

Adresse einer 8 Zeichen langen Kopie des EIB-Ressourcenwerts, EIBRSRCE.

#### UEP\_TD\_REMOTE\_SYSTEM

Wenn die Anforderung an eine ferne Region gesendet werden soll, ist dies die Adresse eines Bereichs mit dem 4-Byte-Namen der fernen Region. (Die ferne Region kann beispielsweise durch die Option SYSID des Befehls, durch die Funktionsverlagerung durch die Option REMOTESYSTEM der TDQUEUE-Definition angegeben worden sein.)

Wenn die Anforderung in der lokalen Region ausgeführt werden soll, ist dieser Parameter die Adresse eines 4-Byte-Bereichs mit Leerzeichen.

#### **UEP\_TD\_REMOTE\_NAME**

Wenn die Anforderung an eine ferne Region gesendet werden soll, ist dies die Adresse eines Bereichs mit dem aus 4 Zeichen bestehenden Namen, unter dem die Warteschlange in der fernen Region bekannt ist.

#### **Rückgabecodes**

##### **UERCNORM**

Verarbeitung fortsetzen.

##### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

#### **XPI-Aufrufe**

Es können alle verwendet werden.

Wenngleich der Exit die Verwendung der XPI-Aufrufe GETMAIN und FREEMAIN zulässt, wird empfohlen, stattdessen die Befehle EXEC CICS GETMAIN und EXEC CICS FREEMAIN zu verwenden.

#### **API- und SPI-Befehle**

Es können alle Befehle verwendet werden, mit Ausnahme der folgenden:

EXEC CICS SHUTDOWN

EXEC CICS XCTL

**Anmerkung:** Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor. So müssen Sie beispielsweise den Eintritt in eine Schleife vermeiden, wenn Sie eine Anforderung für transiente Daten über den Exit XTDEREQC ausgeben. In diesem Zusammenhang wird die Verwendung des Rekursionszählers UEPRECUR empfohlen.

## **Parameterstruktur der Befehlsebene**

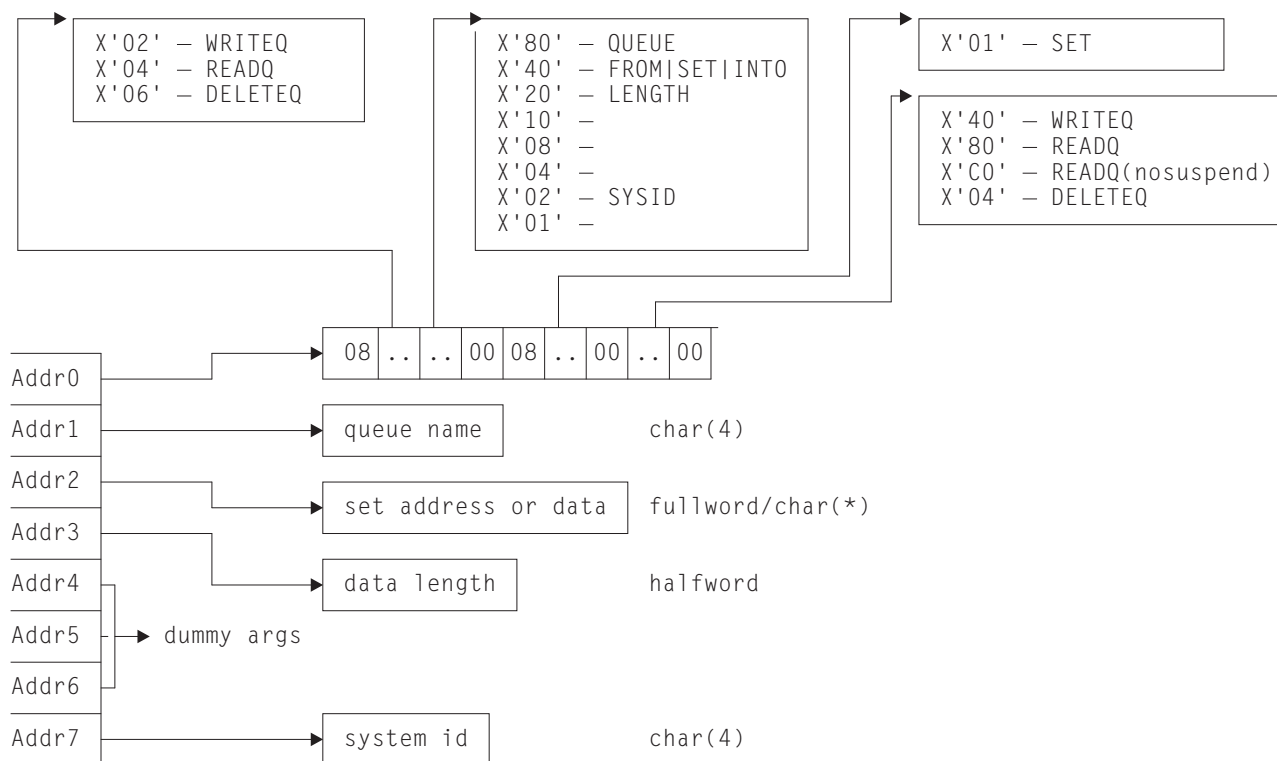


Abbildung 8. Parameterstruktur der Befehlsebene für transiente Daten

Die Parameterstruktur der Befehlsebene besteht aus einer Reihe von Adressen. Die erste Adresse verweist auf den EXEC-Schnittstellendeskriptor (EID), der aus einem 8-Byte-Bereich besteht, der den Typ der Anforderung beschreibt und jedes mit der Anforderung angegebene Schlüsselwort identifiziert. Die übrigen Adressen verweisen auf Teile von Daten, die der Anforderung zugeordnet sind. (So verweist beispielsweise die zweite Adresse auf den Namen der Warteschlange.)

Anhand des EXEC-Schnittstellendeskriptors können Sie den Typ der Anforderung sowie die angegebenen Schlüsselwörter ermitteln. Anhand der anderen Parameter in der Liste können Sie die Werte der Schlüsselwörter ermitteln. Sie können auch Werte von Schlüsselwörtern ändern, die in der Anforderung angegeben sind. (So könnten Sie beispielsweise die in der Anforderung angegebene System-ID (SYSID) ändern.)

## Indikator für Ende der Parameterliste

Das höchstwertige Bit wird für die letzte Adresse in der Parameterliste gesetzt, um anzuzeigen, dass es sich um die letzte Adresse in der Liste handelt. Bei der Rückkehr vom Benutzerexitprogramm sucht CICS in der Parameterliste nach dem höchstwertigen Bit, um den letzten Parameter ausfindig zu machen. Daher gilt: Wenn Sie die Länge der Parameterliste ändern, müssen Sie auch das höchstwertige Bit neu setzen, um so die aktuelle letzte Adresse zu markieren.

Beispiel: Wenn in der Parameterliste nur die ersten beiden Adressen angegeben werden (TD\_ADDR0, die Adresse des EID, und TD\_ADDR1, die Adresse des in einer DELETEQ-Anforderung angegebenen Warteschlangennamens), dann wird das höchstwertige Bit in TD\_ADDR1 gesetzt. Wenn Sie dann die Parameterliste erwei-

tern, indem Sie beispielsweise die Adresse einer SYSID in TD\_ADDR7 festlegen, müssen Sie das höchstwertige Bit in TD\_ADDR1 zurücksetzen und stattdessen in TD\_ADDR7 neu setzen.

Es wird die maximale Größe der Parameterliste an den Exit übergeben, sodass das Exitprogramm alle bisher noch nicht angegebenen Parameter hinzufügen kann, ohne zuvor mehr Speicher abrufen zu müssen.

Die ursprüngliche Parameterliste wird nach Beendigung von XTDEREQC in der Version wiederhergestellt, die vor dem Aufruf von XTDEREQ vorlag. Daraus folgt, dass die Ausführungsdiagnosefunktion (Execution Diagnostic Facility, EDF) vor **und** nach der Ausführung den ursprünglichen Befehl anzeigt. **EDF zeigt keine Änderungen an, die durch den Exit vorgenommen wurden.**

### Exitspezifischer Parameter UEPCLPS

Der exitspezifische Parameter UEPCLPS ist sowohl in Exit XTDEREQ als auch in Exit XTDEREQC enthalten. Er enthält die Adresse der Parameterstruktur der Befehlsebene.

Die Parameterstruktur der Befehlsebene enthält 8 Adressen: TD\_ADDR0 bis TD\_ADDR7. Die Struktur wird im Pseudobereich (DSECT) TD\_ADDR\_LIST definiert, den Sie in Ihr Exitprogramm kopieren sollten, indem Sie die Anweisung COPY DFHTDUEd einschließen.

Die Parameterliste der Befehlsebene ist wie folgt aufgebaut:

#### TD\_ADDR0

Adresse eines 8-Byte-Bereichs, der als EID (EXEC Interface Descriptor) bezeichnet wird und wie folgt aufgebaut ist:

- TD\_GROUP
- TD\_FUNCT
- TD\_BITS1
- TD\_BITS2
- TD\_EIDOPT5
- TD\_EIDOPT6
- TD\_EIDOPT7

#### TD\_GROUP

Immer X'08'; hierdurch wird angegeben, dass es sich um eine Anforderung für transiente Daten handelt.

#### TD\_FUNCT

Ein Byte, das den Typ der Anforderung definiert:

X'02'    WRITEQ  
X'04'    READQ  
X'06'    DELETEQ

#### TD\_BITS1

Existenzbits zum Definieren der Argumente, die angegeben wurden. Um das Argument abzurufen, das einem Schlüsselwort zugeordnet ist, müssen Sie die entsprechende Adresse aus der Parameterstruktur der Befehlsebene verwenden. Bevor Sie diese Adresse verwenden, müssen Sie das zugehörige Existenzbit überprüfen. Wenn das Existenzbit inaktiviert ist, wurde das Argument in der Anforderung nicht angegeben, und die Adresse sollte nicht verwendet werden.

- X'80'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort **QUEUE** enthält. Ist dieses Existenzbit gesetzt, dann ist **TD\_ADDR1** aussagekräftig.
- X'40'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort **INTO**, **SET** oder **FROM** enthält. Ist dieses Existenzbit gesetzt, dann ist **TD\_ADDR2** aussagekräftig.
- X'20'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort **LENGTH** enthält. Ist dieses Existenzbit gesetzt, dann ist **TD\_ADDR3** aussagekräftig.
- X'02'** Wird gesetzt, wenn die Anforderung ein Argument für das Schlüsselwort **SYSID** enthält. Ist dieses Existenzbit gesetzt, dann ist **TD\_ADDR7** aussagekräftig.

#### **TD\_BITS2**

Zwei Byte, die nicht von transienten Daten verwendet werden.

#### **TD\_EIDOPT5**

Gibt an, ob bestimmte Schlüsselwörter in der Anforderung angegeben wurden.

- X'01'** SET (und nicht INTO) wurde angegeben.

#### **TD\_EIDOPT6**

Ein Byte, die nicht von transienten Daten verwendet wird.

#### **TD\_EIDOPT7**

Gibt an, ob bestimmte Funktionen, Schlüsselwörter oder beides in der Anforderung angegeben wurden:

- X'40'** WRITEQ wurde angegeben.
- X'80'** READQ wurde angegeben.
- X'C0'** READQ(nosuspend) wurde angegeben.
- X'04'** DELETEQ wurde angegeben.

#### **TD\_ADDR1**

Adresse eines 4-Byte-Bereichs mit dem Namen aus **QUEUE**.

#### **TD\_ADDR2**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse einer 4-Byte-Adresse aus **SET** (wenn die Anforderung **READQ** ist und **TD\_EIDOPT5** anzeigt, dass dies **SET** ist).
- Adresse der Daten aus **INTO** (wenn die Anforderung **READQ** ist und **TD\_EIDOPT5** anzeigt, dass es sich nicht um **SET** handelt). Sie können dieses Bit in Ihrem Benutzerexit nicht ändern.
- Adresse der Daten aus **FROM** (wenn die Anforderung **WRITEQ** ist).

#### **TD\_ADDR3**

Hierbei handelt es sich um eine der folgenden Adressen:

- Adresse des Halbwortwerts für **LENGTH** (wenn die Anforderung **READQ** oder **WRITEQ** ist). **Warnung:** Bei Anforderungen mit Angabe von **INTO** darf der Wert für **LENGTH** nicht in einen Wert geändert werden, der größer ist als der von der Anwendung angegebene Wert. Wird ein größerer Wert angegeben, führt dies zu einer Speicherüberlagerung in der Anwendung.

#### **TD\_ADDR4**

Adresse eines Werts, der nur zur internen Verwendung durch CICS bestimmt ist. Dieser Wert darf nicht verwendet werden.

**TD\_ADDR5**

Adresse eines Werts, der nur zur internen Verwendung durch CICS bestimmt ist. Dieser Wert darf nicht verwendet werden.

**TD\_ADDR6**

Adresse eines Werts, der nur zur internen Verwendung durch CICS bestimmt ist. Dieser Wert darf nicht verwendet werden.

**TD\_ADDR7**

Adresse eines Bereichs, der den Wert für SYSID enthält.

**TD\_ADDR8**

Adresse eines Werts, der nur zur internen Verwendung durch CICS bestimmt ist. Dieser Wert darf nicht verwendet werden.

**Ändern von Feldern in der Parameterstruktur der Befehlsebene**

Von den Feldern, die an transiente Daten übergeben werden, werden einige als Eingabefelder für die Anforderung, andere als Ausgabefelder und wieder andere sowohl als Eingabe- als auch als Ausgabefelder verwendet. Die Methode, die Ihr Benutzerexitprogramm zum Ändern eines Felds verwendet, hängt von der Verwendung des betreffenden Felds ab.

Die folgenden Felder sind immer Eingabefelder:

- QUEUE
- FROM
- SYSID

Die folgenden Felder sind immer Ausgabefelder:

- INTO
- SET

LENGTH ist in Anforderungen vom Typ WRITEQ ein Eingabefeld und ein Ausgabefeld in Anforderungen vom Typ READQ mit Angabe von SET. In Anforderungen vom Typ READQ mit Angabe von INTO ist LENGTH sowohl ein Eingabe- als auch ein Ausgabefeld.

**Ändern von Eingabefeldern**

Die richtige Methode zum Ändern eines Eingabefelds besteht darin, eine neue Kopie desselben zu erstellen und die Adresse in der Parameterliste der Befehlsebene so zu ändern, dass sie auf die neuen Daten verweist.

**Anmerkung:** Sie dürfen ein Eingabefeld niemals ändern, indem Sie die Daten ändern, auf die durch die Parameterliste der Befehlsebene verwiesen wird. Dies würde zu einer Beschädigung von Speicher führen, der zum Anwendungsprogramm gehört; dies wiederum würde zu einem Fehler führen, wenn das Programm versucht, das Feld wiederzuverwenden.

**Ändern von Ausgabefeldern**

Das unter „Ändern von Eingabefeldern“ beschriebene Verfahren ist für das Ändern von Ausgabefeldern nicht geeignet. (Die Ergebnisse würden an den neuen Bereich und nicht an den Bereich der Anwendung zurückgegeben und wären daher für die Anwendung nicht sichtbar.)

Ein Ausgabefeld wird geändert, indem die Daten geändert werden, auf die von der Parameterliste der Befehlsebene verwiesen wird. Bei einem Ausgabefeld können Sie die vorhandenen Anwendungsdaten ändern, da die Anwendung erwartet, dass das Feld geändert wird.

### Ändern von Feldern für Eingabe und Ausgabe

Ein Beispiel eines Felds, das sowohl für Eingabe als auch für Ausgabe verwendet wird, ist das Längensfeld LENGTH in einer Anforderung vom Typ READQ mit Angabe von INTO. Derartige Felder können in derselben Weise wie Ausgabefelder behandelt werden, und sie werden auch als solche angesehen.

### Ändern des EXEC-Schnittstellendeskriptors (EID)

Es ist nicht möglich, den EXEC-Schnittstellendeskriptor (EXEC Interface Descriptor, EID) zu ändern, um bedeutende Änderungen an Anforderungen vorzunehmen. So kann beispielsweise eine Anforderung vom Typ READQ nicht in eine Anforderung vom Typ WRITEQ geändert werden. Es können jedoch geringfügige Änderungen an Anforderungen vorgenommen werden. So ist es beispielsweise möglich, das Existenzbit für die System-ID (SYSID) zu aktivieren, sodass die betreffende Anforderung in eine Anforderung geändert werden kann, die an ein fernes System übertragen wird.

In der nachstehenden Liste werden diejenigen Bits im EID aufgeführt, die geändert werden können. Versuche, andere Bestandteile des EID zu ändern, werden ignoriert.

#### TD\_BITS1

X'20' Existenzbit für LENGTH.

X'02' Existenzbit für SYSID.

#### TD\_EIDOPT5

X'01' Existenzbit für das Schlüsselwort SET. Sie können dieses Bit über Ihren Benutzerexit nicht ändern.

#### TD\_EIDOPT7

Änderungen an TD\_EIDOPT7 sind auf READQ-Anforderungen beschränkt. X'80' für READQ ist austauschbar mit X'C0' für READQ(nosuspend). Es können keine weiteren Änderungen an diesem Byte vorgenommen werden.

Der EID wird vor der Rückkehr zum Anwendungsprogramm auf seinen ursprünglichen Wert zurückgesetzt. Dies bedeutet, dass Änderungen, die am EID vorgenommen werden, nur für die Dauer der Anforderung für transiente Daten beibehalten werden.

**Anmerkung:** Das Benutzerexitprogramm wird zwar daran gehindert, bedeutende Änderungen am EID vorzunehmen. Dennoch müssen Sie auch bei den geringfügigen Änderungen, die **zulässig sind**, sehr sorgfältig vorgehen.

### Der EXEC-Schnittstellenblock (EIB)

Es werden Kopien von EIBRSRCE, EIBRCODE, EIBRESP und EIBRESP2 an den Exit übergeben, damit Sie folgende Aktionen ausführen können:

- Ändern oder Festlegen von Ausführungs- und Ressourceninformationen in XTDEREQ und XTDEREQC.
- Untersuchen von Ausführungs- und Ressourceninformationen in XTDEREQC.



Sie können die Kopien von EIBRSRCE, EIBRCODE, EIBRESP und EIBRESP2, die Sie in der Parameterliste erhalten, aktualisieren. Die transienten Daten kopieren Ihre Werte in den realen EIB, nachdem der Exit XTDEREQC ausgeführt wurde oder wenn Sie den Rückgabecode für Umgehen in XTDEREQ angeben.

Sie müssen gültige Antworten für transiente Daten festlegen. Sie müssen für die drei Codes EIBRCODE, EIBRESP und EIBRESP2 eine konsistente Gruppe von Werten festlegen, so wie sie auch von transienten CICS-Daten festgelegt würden, um eine gültige Ausführung zu beschreiben. **Die Konsistenz der Werte für EIBRCODE, EIBRESP und EIBRESP2 wird von CICS nicht überprüft.** Wenn jedoch EIBRCODE auf einen Wert ungleich null gesetzt wird und EIBRESP auf null gesetzt wird, überschreibt CICS diesen Wert für EIBRESP durch einen Wert, der ungleich null ist. Als Hilfestellung beim Festlegen von Werten für EIBRCODE, EIBRESP und EIBRESP2 können Sie die im Pseudobereich (DSECT) DFHTDUEU angegebenen Werte heranziehen, die von transienten Daten verwendet werden.

---

## Exits XRCINIT und XRCINPT des Wiederherstellungsprogramms für Benutzerprotokolldatensätze

Bei einem Warmstart sowie bei einem Wiederanlauf nach Systemabsturz müssen Aktualisierungen an wiederherstellbaren CICS-Ressourcen, die zum Zeitpunkt der Beendigung des Systems noch nicht festgeschrieben waren, zurückgesetzt werden. Die Exits XRCINIT und XRCINPT werden über das Wiederherstellungsprogramm für Benutzerprotokolldatensätze aufgerufen. Dieses Programm wird verwendet, um vom Benutzer geschriebene Systemprotokolleinträge bei Bedarf zurückzusetzen.

XRCINIT wird bei einem Warmstart bzw. Wiederanlauf nach Systemabsturz wie folgt aufgerufen:

- Bevor der erste Benutzerwiederherstellungsdatensatz an XRCINPT übergeben wird.
- Nachdem alle derartigen Datensätze an XRCINPT übergeben worden sind.

XRCINPT wird immer dann aufgerufen, wenn ein Benutzerprotokollsatz aus dem Systemprotokoll gelesen wird.

Mithilfe von XRCINPT können Sie die Standardaktionen ändern, die CICS bei einem Wiederanlauf nach Systemabsturz für bestimmte Datensätze in Benutzerjournalen ausführen soll. Bei den an XRCINPT übergebenen Datensätzen handelt es sich um Datensätze in Arbeitseinheiten (UOWs), für die Folgendes gilt:

- UOWs, die im letzten vollständigen Aktivitätsschlüsselpunkt aufgeführt wurden.
- UOWs, die gerade ausgeführt wurden, als CICS beendet wurde.
- UOWs, die nach dem Start des letzten vollständigen Aktivitätsschlüsselpunkts festgeschrieben, zurückgesetzt oder in den Status 'unbestätigt' versetzt wurden. (Dies gilt jedoch nur für Datensätze, bei denen in der Anforderung WRITE JOURNALNAME(DFHLOG) das Bit ganz links im Wert von JTYPEID eine Eins war.

Datensätze, die vom Exit XAKUSER für Aktivitätsschlüsselpunkte geschrieben wurden, werden nur dann übergeben, wenn sie im letzten vollständigen Aktivitätsschlüsselpunkt aufgeführt wurden. Diese Datensätze werden im Anschluss an alle anderen Datensätze übergeben. Die Reihenfolge, in der Datensätze dargestellt werden, kann daher von der Reihenfolge in der zurückgesetzten Datenstromsequenz abweichen.

Das Format von Datensätzen, die an den Exit übergeben werden, lautet wie folgt:

**Offset   Feldinhalt**

0	JTYPEID
2	Reserviert
4	Länge der Präfixdaten (L). (Null, wenn kein Präfix vorhanden ist.)
8	Präfixdaten (sofern vorhanden)
8 + L	Protokolldaten

Der Datensatz wird durch den Pseudobereich (DSECT) CL\_USER\_HEADER im Copybook DFHLGGFD zugeordnet.

Wenn Sie XRCINIT und XRCINPT verwenden, sollten Sie beachten, dass es möglich ist, dass die Exits aufgerufen werden, bevor die Wiederherstellung von Ressourcen für temporären Speicher und transiente Daten abgeschlossen ist.

## Codierung der Exitprogramme

Mithilfe der XPI- oder EXEC CICS-Befehle können Sie CICS-Services in Exitprogrammen verwenden, die über diese Exits aufgerufen werden.

Hierbei ist Folgendes zu beachten:

- Es gibt eine Einschränkung hinsichtlich der Verwendung von XPI in einem frühen Stadium der Initialisierung: Rufen Sie bis zur zweiten PLTPI-Phase keine Exitprogramme auf, die die XPI-Funktionen TRANSACTION\_DUMP, WRITE\_JOURNAL\_DATA, MONITOR und INQUIRE\_MONITOR\_DATA verwenden.
- Ebenso gibt es Einschränkungen hinsichtlich der Verwendung von EXEC CICS-Befehlen in diesen Exits:
  - Es können keine EXEC CICS-Befehle für den Zugriff auf Terminalsteuerservices verwendet werden.
  - Von der Verwendung von temporärem Speicher, transienten Daten, Dateisteuerung, Journalsteuerung und DL/I-Services wird dringend abgeraten, da die Ressourcen, auf die zugegriffen werden soll, auch in einem Status der Wiederherstellung vorliegen können und daher möglicherweise nicht einsatzbereit sind. Der Versuch, unter diesen Umständen auf Ressourcen zuzugreifen, führt im besten Fall zu einer Serialisierung der Wiederherstellungstasks und im schlimmsten Fall zu einem Deadlock.

Wenn Sie dennoch Dateisteueranforderungen in Programmen ausgeben, die über diese Exits aufgerufen werden, beachten Sie Folgendes:

- Wenn ein Exitprogramm aufgrund einer Dateisteueranforderung einen Bereich anfordert, liegt es in der Zuständigkeit des Programms, diesen Bereich auch wieder freizugeben.
- Ein Exitprogramm darf nicht versuchen, Dateisteueranforderungen an eine Datei auszugeben, die sich auf ein VSAM-Dataset mit der Zeichenfolgennummer 1 bezieht, es sei denn, es wird während der Verarbeitung des Initialisierungsexits keine Aktion für diese Datei angegeben.
- Das Exitprogramm darf keine EXEC CICS-Befehle ausgeben, wenn die Wiederherstellung als Ergebnis einer Anforderung vom Typ EXEC CICS SYNCPOINT ROLLBACK ausgeführt wird.
- Exitprogramme, die EXEC CICS-Befehle ausgeben, müssen zunächst den EXEC-Schnittstellenblock (EIB) adressieren. Siehe Using CICS services.

- Exitprogramme, die EXEC CICS-Befehle ausgeben und das Makro DFHEIENT verwenden, sollten das Makro DFHEIRET verwenden, um einen Rückgabecode festzulegen und zu CICS zurückzukehren. Siehe Global user exit XPI examples, showing the use of storage.
- Exitprogramme, die von diesen Exits aufgerufen werden, müssen mit der Option NOEDF umgesetzt werden, wenn sie EXEC CICS-Befehle ausgeben. Siehe EDF and global user exits.
- Der in einem Exitprogramm mittels Taskverkettung erworbene Speicher wird bei Beendigung der Verarbeitung eines Wiederanlaufs nach Systemabsturz freigegeben. Das Exitprogramm sollte jedoch versuchen, den Speicher freizugeben, sobald dessen Inhalt nicht mehr benötigt wird.
- Kein Exitprogramm sollte die vom Programm für Dateisteuerungsbackout festgelegten Indikatoren 'absent' (abwesend) und 'no-action' (keine Aktion) zurücksetzen.
- Gehen Sie beim Ausgeben von rekursiven Befehlen mit Sorgfalt vor, um keine Schleife zu verursachen. So sind Sie beispielsweise dafür verantwortlich, das Auftreten einer Schleife zu vermeiden, wenn eine RC-Anforderung über diese Exits ausgegeben wird.

## Aktivierung der Exitprogramme

Um diese Exits zu aktivieren, müssen Sie einen der folgenden Schritte ausführen:

- Geben Sie den Systeminitialisierungsparameter  
TBEXITS=(name1,name2,name3,name4,name5,name6) an. Hierbei sind 'name1' bis 'name6' die Namen Ihrer Benutzerexitprogramme für XRCINIT, XRCINPT, XFCBFAIL, XFCLDEL, XFCBOVER und XFCBOUT.
- Aktivieren Sie die Exits während der ersten Phase der Initialisierung mithilfe eines PLTPI-Programms.

Wenn Sie den Parameter TBEXITS verwenden, um die Exits zu aktivieren, wird ein globaler Arbeitsbereich mit 4 Byte bereitgestellt. Wenn Sie ein PLTPI-Programm verwenden, können Sie die Größe des globalen Arbeitsbereichs selbst auswählen. Darüber hinaus können Sie auch mehrere Exitprogramme zur Verwendung an den einzelnen Exitpunkten auswählen, während der Parameter TBEXITS an jedem Exitpunkt jeweils immer nur ein Exitprogramm erlaubt. Informationen zur PLTPI-Verarbeitung finden Sie unter Writing initialization and shutdown programs.

## Exit XRCINIT

### Zeitpunkt des Aufrufs

Bei einem Warmstart und einem Wiederanlauf nach Systemabsturz:

- Bevor der erste Benutzerwiederherstellungsdatensatz an XRCINPT übergeben wird.
- Nachdem alle derartigen Datensätze an XRCINPT übergeben worden sind.

### Exitspezifische Parameter

#### UEPTREQ

Adresse eines 1-Byte-Flags, das die Ursache für den Aufruf angibt. Wenn UEPTREQ den Wert UEUSINIT aufweist, dann wurde der Exit beim Start der Benutzerwiederherstellung aufgerufen; wenn UEPTREQ den Wert UEUSTERM aufweist, dann wurde der Exit am Ende der Benutzerwiederherstellung aufgerufen.

**UEPRSTRT**

Adresse eines 1-Byte-Flags, das angibt, wie CICS erneut gestartet wurde:

**UEPRWARM**

Warmstart.

**UEPREMER**

Start nach Systemabsturz.

**Rückgabecodes****UERCNORM**

Verarbeitung fortsetzen. Es werden keine anderen Rückgabecodes unterstützt.

**XPI-Aufrufe**

Es können alle verwendet werden. Informationen zu Einschränkungen finden Sie unter „Exits XRCINIT und XRCINPT des Wiederherstellungsprogramms für Benutzerprotokolldatensätze“ auf Seite 287.

**Exit XRCINPT****Zeitpunkt des Aufrufs**

Bei einem Warmstart und einem Wiederanlauf nach Systemabsturz; einmal für jeden Benutzerprotokolldatensatz, der im Systemprotokoll gefunden wird.

**Exitspezifische Parameter****UEPUOWST**

Adresse eines 1-Byte-Flags, das die Disposition der Arbeitseinheit (UOW) angibt. Mögliche Werte:

**UEPUOWAK**

Aktivitätsschlüsselpunktsatz.

**UEPUOWCM**

Arbeitseinheit festgeschrieben.

**UEPUOWBO**

Arbeitseinheit zurückgesetzt.

**UEPUOWIF**

Arbeitseinheit wurde gerade ausgeführt.

**UEPUOWID**

Arbeitseinheit war unbestätigt.

**UEPLGREG**

Adresse des gerade gelesenen Protokollsatzes. Der Journalsteuerungssatz kann mithilfe der unter CICS logging and journaling bereitgestellten Informationen zugeordnet werden.

**UEPLGLEN**

Adresse eines Vollworts, das die Länge des Protokollsatzes enthält.

**UEPTAID**

Adresse eines 4-Byte-Felds mit der Task-ID.

**UEPTRID**

Adresse eines 4-Byte-Felds mit der Transaktions-ID.

**UEPTEID**

Adresse eines 4-Byte-Felds mit der Terminal-ID.

**Anmerkung:** Die Werte der Felder, die von UEPTAID, UEPTRID und UEP-TEID adressiert werden, sind für Aktivitätsschlüsselpunktsätze bedeutungslos (also dann, wenn das von UEPUOWST adressierte Feld UEPUOWAK enthält).

#### Rückgabecodes

##### UERCNORM

Verarbeitung fortsetzen.

##### UERCBYP

Diesen Datensatz umgehen.

#### XPI-Aufrufe

Es können alle verwendet werden. Informationen zu Einschränkungen finden Sie unter „Exits XRCINIT und XRCINPT des Wiederherstellungsprogramms für Benutzerprotokolldatensätze“ auf Seite 287.

---

## SNA-LU-Verwaltungsprogrammexit (XZCATT)

Dieser Exit wird aufgerufen, bevor eine Task für eine LU-Terminaltask angehängt wird.

#### Zeitpunkt des Aufrufs

Vor dem Anhängen von Tasks für Terminaltasks.

#### Exitspezifische Parameter

##### UEPTCTTE

Adresse des Terminaleintrags in der Terminalsteuertabelle (TCTTE). Der TCTTE kann mithilfe des Pseudobereichs (DSECT) DFHTCTTE zugeordnet werden.

##### UEPTIOA

Adresse des Terminal-E/A-Bereichs (TIOA). Der TIOA kann mithilfe des Pseudobereichs (DSECT) DFHTIOA zugeordnet werden. Bei den Feldern TIOASAL und TIOASCA handelt es sich jedoch **nicht** um Programmierschnittstellen.

##### UEPTPN

Adresse des APPC-Transaktionsprozessnamens (TPN) oder des LU6.1-Prozessnamens (DPN), dessen Länge durch den Parameter UEPTPNL adressiert wird.

##### UEPTPNL

Adresse eines 1-Byte-Felds mit der Länge des TPN oder DPN.

##### UEPTRAN

Adresse der 4-Byte-Transaktions-ID.

**Anmerkung:** Das Exitprogramm darf die Transaktions-ID von Tasks, die durch die automatische Transaktionsinitialisierung (ATI) gestartet werden, nicht ändern. (Dies liegt daran, dass CICS die Transaktions-ID in der eigenen Programmsteuertabelle mit der Transaktions-ID in dem in der Anwendungsverwaltungsregion (AOR) erstellten automatischen Initialisierungsdeskriptor (AID) abgleichen muss.

#### Rückgabecodes

##### UERCNORM

Verarbeitung fortsetzen.

### **XPI-Aufrufe**

Es können alle verwendet werden.

### **Beispielexitprogramm**

DFH\$ZCAT

---

## **Exits für SNA-Arbeitsbereichsmodule (XZCIN, XZCOUT, XZCOUT1 und XZIQUE)**

Diese Exits werden nach E/A-Ereignissen aufgerufen oder bevor Nachrichten in Anforderungseinheiten (RUs) disassembliert werden.

### **Exit XZCIN**

Dieser Exit wird nach einem Eingabeereignis aufgerufen.

#### **Zeitpunkt des Aufrufs**

Nach einem Eingabeereignis.

#### **Exitspezifische Parameter**

##### **UEPTCTTE**

Adresse des Terminaleintrags in der Terminalsteuertabelle (TCTTE). Der TCTTE kann mithilfe des Pseudobereichs (DSECT) DFHTCTTE zugeordnet werden.

##### **UEPTIOA**

Adresse des Terminal-E/A-Bereichs (TIOA). Das Exitprogramm sollte die Adresse nicht ändern. Der TIOA kann mithilfe des Pseudobereichs (DSECT) DFHTIOA zugeordnet werden. Bei den Feldern TIOASAL und TIOASCA handelt es sich jedoch nicht um Programmierschnittstellen.

#### **Rückgabecodes**

##### **UERCNORM**

Verarbeitung fortsetzen.

### **XPI-Aufrufe**

Es können alle verwendet werden. Verwenden Sie jedoch keinen GET-MAIN-Aufruf, um Terminalklassenspeicher zur Verwendung als Ersatz für den Terminal-E/A-Bereich (TIOA) abzurufen. Dies liegt daran, dass es mehrere interne Verweise auf den TIOA gibt, und wenn auch nur einer dieser internen Verweise nicht aktualisiert wird, kann die Anwendung möglicherweise auf Probleme stoßen.

### **API- und SPI-Befehle**

Es können keine **EXEC CICS**-Befehle verwendet werden.

### **Exit XZCOUT**

Dieser Exit wird vor einem Ausgabeereignis aufgerufen.

#### **Zeitpunkt des Aufrufs**

Vor einem Ausgabeereignis.

#### **Exitspezifische Parameter**

##### **UEPTCTTE**

Adresse des Terminaleintrags in der Terminalsteuertabelle (TCTTE). Der TCTTE kann mithilfe des Pseudobereichs (DSECT) DFHTCTTE zugeordnet werden.

### **UEPTIOA**

Adresse des Terminal-E/A-Bereichs (TIOA). Das Exitprogramm sollte die Adresse nicht ändern. Der TIOA kann mithilfe des Pseudobereichs (DSECT) DFHTIOA zugeordnet werden. Bei den Feldern TIOASAL und TIOASCA handelt es sich jedoch **nicht** um Programmierschnittstellen.

**Anmerkung:** Unter bestimmten Umständen, beispielsweise wenn XZCOUT vor dem Senden von NULL RU aufgerufen wird, enthält UEPTIOA Nullen.

### **Rückgabecodes**

#### **UERCNORM**

Verarbeitung fortsetzen.

### **XPI-Aufrufe**

Es können alle verwendet werden. Es wird jedoch nicht empfohlen, einen GETMAIN-Aufruf zu verwenden, um Terminalklassenspeicher zur Verwendung als Ersatz für den Terminal-E/A-Bereich (TIOA) abzurufen. Dies liegt daran, dass es mehrere interne Verweise auf den TIOA gibt, und wenn auch nur einer dieser internen Verweise nicht aktualisiert wird, kann die Anwendung möglicherweise auf Probleme stoßen.

## **Exit XZCOUT1**

Dieser Exit wird aufgerufen, bevor eine Nachricht in RUs zerlegt wird.

### **Zeitpunkt des Aufrufs**

Bevor eine Nachricht in RUs aufgeteilt wird.

### **Exitspezifische Parameter**

#### **UEPTCTTE**

Adresse des Terminaleintrags in der Terminalsteuertabelle (TCTTE). Der TCTTE kann mithilfe des Pseudobereichs (DSECT) DFHTCTTE zugeordnet werden.

#### **UEPTIOA**

Adresse des Terminal-E/A-Bereichs (TIOA). Das Exitprogramm sollte die Adresse nicht ändern. Der TIOA kann mithilfe des Pseudobereichs (DSECT) DFHTIOA zugeordnet werden. Bei den Feldern TIOASAL und TIOASCA handelt es sich jedoch **nicht** um Programmierschnittstellen.

### **Rückgabecodes**

#### **UERCNORM**

Verarbeitung fortsetzen.

### **XPI-Aufrufe**

Es können alle verwendet werden. Es wird jedoch nicht empfohlen, einen GETMAIN-Aufruf zu verwenden, um Terminalklassenspeicher zur Verwendung als Ersatz für den Terminal-E/A-Bereich (TIOA) abzurufen. Dies liegt daran, dass es mehrere interne Verweise auf den TIOA gibt, und wenn auch nur einer dieser internen Verweise nicht aktualisiert wird, kann die Anwendung möglicherweise auf Probleme stoßen.

## Exit XZIQUE zum Verwalten systemübergreifender MRO- und APPC-Warteschlangen

Sie können den Exit XZIQUE verwenden, um die Anzahl der in die Warteschlange gestellten Anforderungen für Sitzungen in MRO- und APPC-Verbindungen zu steuern.

### Anmerkung:

- Anforderungen in der Warteschlange für Sitzungen werden als „*Zuordnungswarteschlangen*“ bezeichnet.
- Der entsprechende globale Benutzerexit zum Steuern der Anzahl von Anforderungen in der Warteschlange für Sitzungen in Verbindungen vom Typ IP Interconnectivity (IPIC) ist XISQUE. Siehe „Exit XISQUE zum Verwalten systemübergreifender IPIC-Warteschlangen“ auf Seite 302.
- Es gibt mehrere Methoden, mit denen Sie die Länge von systemübergreifenden Warteschlangen steuern können. Eine Beschreibung der verschiedenen Methoden finden Sie unter *Intersystem session queue management*.

Mithilfe des Exits XZIQUE können Sie Probleme bei der Warteschlangensteuerung (Engpässe) bereits in einem frühen Stadium erkennen. Dieser Exit erweitert die Funktionalität des globalen Benutzerexits XISCONA, der unter „Exits XISCONA, XISLCLQ und XISQLCL für das Programm für systemübergreifende Kommunikation“ auf Seite 148 beschrieben wird. XISCONA wird lediglich für Funktionsverlagerungs- und DPL-Anforderungen aufgerufen. XZIQUE hingegen wird nicht nur für Funktionsverlagerungs- und DPL-Anforderungen, sondern auch für Anforderungen für Transaktionsrouting, asynchrone Verarbeitung und verteilte Transaktionsverarbeitung aufgerufen. Im Vergleich zu XISCONA erhält XZIQUE ausführlichere Informationen als Grundlage für die Entscheidungsfindung.

Mithilfe von XZIQUE können Zuordnungsanforderungen in Abhängigkeit von der Warteschlangenlänge entweder eingereiht oder zurückgewiesen werden. Darüber hinaus ermöglicht der Exit auch das Beenden und erneute Herstellen von Verbindungen, bei denen ein Engpass auftritt.

### Interaktion mit dem Exit XISCONA

Es gibt keine Interaktion zwischen den globalen Benutzerexits XZIQUE und XISCONA. Wenn Sie beide Exits aktivieren, dann könnten XISCONA und XZIQUE gleichermaßen für Funktionsverlagerungs- und DPL-Anforderungen aufgerufen werden, was jedoch nicht empfohlen wird.

Daher sollten Sie sicherstellen, dass jeweils immer nur einer dieser Exits aktiviert ist. Es wird empfohlen, eher XZIQUE als XISCONA zu verwenden, da XZIQUE mehr Funktionen und größere Flexibilität bietet.

Wenn Sie bereits über ein globales Benutzerexitprogramm für XISCONA verfügen, können Sie es möglicherweise entsprechend für die Verwendung am Exitpunkt XZIQUE ändern.

### Zeitpunkt des Aufrufs des Exits XZIQUE

Sofern er aktiviert ist, wird der globale Benutzerexit XZIQUE wie folgt aufgerufen:

- Sobald CICS versucht, eine Sitzung mit einem fernen System anzufordern, aber keine freie Sitzung verfügbar ist. Der Exit wird unabhängig davon aufgerufen, ob Sie die Option QUEUELIMIT in der Verbindungsdefinition (CONNECTION)



angegeben haben und ob der Grenzwert überschritten wurde. Der Exit wird nicht aufgerufen, wenn in der Zuordnungsanforderung NOQUEUE oder NOSUSPEND angegeben ist.

Anforderungen für Sitzungen können auf verschiedene Art und Weise entstehen. Beispielsweise können explizite Befehle vom Typ EXEC CICS ALLOCATE durch DTP-Programme ausgegeben werden, oder es können solche Anforderungen mittels Anforderungen für Transaktionsrouting oder Funktionsverlagerung übergeben werden.

- Sobald eine Zuordnungsanforderung eine freie Sitzung findet, nachdem die Warteschlange für die Verbindung durch einen vorherigen Aufruf des Exitprogramms bereinigt worden ist. In diesem Fall kann das Exitprogramm angeben, dass CICS die Verarbeitung normal fortsetzen und bei Bedarf die Warteschlangensteuerung wiederaufnehmen soll.

## **Verwendung eines globalen Benutzerexitprogramms für XZIQUE**

Wenn der Exit XZIQUE aktiviert ist, kann das globale Benutzerexitprogramm für XZIQUE verwendet werden, um den Status der Zuordnungswarteschlange für eine bestimmte Verbindung im lokalen System zu überprüfen.

Informationen werden in einer Parameterliste an das Exitprogramm übergeben. Diese Parameterliste ist so strukturiert, dass sie in Abhängigkeit von der Sitzungsanforderung Daten entweder zu unspezifischen Zuordnungsanforderungen oder zu Anforderungen für bestimmte Modusgruppen bereitstellen kann. Unspezifische Zuordnungsanforderungen gelten für MRO, LU6.1 und APPC-Sitzungen ohne Angabe einer Modusgruppe.

Anhand der in der Parameterliste übergebenen Informationen kann Ihr globales Benutzerexitprogramm (beispielsweise auf Grundlage der Warteschlangenlänge) entscheiden, ob CICS die Zuordnungsanforderung in die Warteschlange stellen soll. Das Programm kommuniziert die getroffene Entscheidung an CICS mithilfe eines der von CICS bereitgestellten Rückgabecodes. Folgende Rückgabecodes sind möglich:

### **UERCAQUE**

Dieser Rückgabecode gibt an, dass CICS die Zuordnungsanforderung in die Warteschlange stellen soll.

Die Gesamtzahl der Zuordnungsanforderungen, die für die Verbindung in die Warteschlange gestellt wurden, wird im Feld A14ESTAQ der Systemeintragsstatistik (für alle unspezifischen Zuordnungen) bzw. im Feld A20ESTAQ der Moduseintragsstatistik (für spezifische Modusgruppenzuordnungen) angegeben. Ausführliche Informationen hierzu finden Sie in den Pseudobereichen (DSECTs) DFHA14DS bzw. DFHA20DS. CICS übergibt im exitspezifischen Parameter UEPQUELIM den Parameter QUEUELIMIT aus der Verbindungsdefinition an das Exitprogramm.

Wenn der Grenzwert nicht erreicht wurde, können Sie die Steuerung mit dem Rückgabecode UERCAQUE an CICS zurückgeben.

### **UERCAPUR**

Dieser Rückgabecode gibt an, dass CICS die Zuordnungsanforderung zurückweisen und SYSIDERR an das Anwendungsprogramm zurückgeben soll, aber die vorhandene Warteschlange unverändert lassen soll.

Wenn die Anzahl der Zuordnungsanforderungen in der Warteschlange den im Parameter QUEUELIMIT für die Verbindung festgelegten Grenzwert erreicht hat, können Sie anfordern, dass CICS die Anforderung zurückweisen soll. Sie sollten jedoch zunächst prüfen, ob der Status der Verbindung zufriedenstellend

ist. Es sollte also überprüft werden, ob die Geschwindigkeit bei der Zuordnung von Sitzungen akzeptabel ist. Verwenden Sie die Uhrzeit, zu der die Warteschlange gestartet wurde, die aktuelle Uhrzeit und die Gesamtzahl der seit dem Start der Warteschlange verarbeiteten Zuordnungen, um die Geschwindigkeit zu ermitteln, mit der CICS Anforderungen verarbeitet. Folgende Felder sind relevant: UEPQAQTS und UEPQACNT für unspezifische Zuordnungsanforderungen sowie UEPMAQTS und UEPMACNT für spezifische Modusgruppenanforderungen.

Um zu ermitteln, ob CICS Sitzungsanforderungen über diese Verbindung mit akzeptabler Geschwindigkeit zuordnet, können Sie die berechnete Zeit mit einem der folgenden Werte vergleichen:

1. Mit dem Wert des Parameters MAXQTIME aus der Verbindungsdefinition, der im exitspezifischen Parameter UEPQMXQT übergeben wird.
2. Mit einem anderen voreingestellten Zeitwert.

Wenn die anhand dieser Formel ermittelte Verarbeitungszeit akzeptabel ist, geben Sie die Steuerung mit Rückgabecode UERCAPUR an CICS zurück, um nur diese eine Anforderung zu bereinigen.

#### **UERCAKLL oder UERCAKLM**

Diese Rückgabecodes geben an, dass CICS die Anforderung wie folgt bearbeiten soll:

- UERCAKLL: Diese Anforderung zurückweisen, alle anderen Zuordnungsanforderungen in der Warteschlange für diese Verbindung bereinigen und eine Informationsnachricht an die Bedienerkonsole senden.
- UERCAKLM: Diese Anforderung zurückweisen, alle anderen Modusgruppenzuordnungsanforderungen in der Warteschlange für diese Verbindung bereinigen und eine Informationsnachricht an die Bedienerkonsole senden.

Wenn der Warteschlangengrenzwert erreicht ist, die Leistung der Zuordnungsverarbeitung für die Warteschlange jedoch unterhalb der in Ihrem Benutzerexitprogramm definierten akzeptablen Grenzwerte liegt, können Sie die Steuerung wie folgt an CICS zurückgeben:

- Bei unspezifischen Zuordnungsanforderungen: Verwenden Sie den Rückgabecode UERCAKLL. Mit UERCAKLL wird auch SYSIDERR an alle Anwendungsprogramme zurückgegeben, die auf die bereinigten Zuordnungsanforderungen warten. Bei nachfolgenden Aufrufen an das Exitprogramm für XZIQUE setzt CICS den Parameter UEPFLAG auf UEPRC8, um anzugeben, dass zuvor UERCAKLL zurückgegeben wurde, um die Warteschlange zu bereinigen.
- Bei spezifischen Modusgruppenzuordnungsanforderungen: Verwenden Sie den Rückgabecode UERCAKLM. Mit UERCAKLM wird auch SYSIDERR an alle Anwendungsprogramme zurückgegeben, die auf die bereinigten Zuordnungsanforderungen warten. Bei nachfolgenden Aufrufen an das Exitprogramm für XZIQUE setzt CICS den Parameter UEPFLAG auf UEPRC12, um anzugeben, dass zuvor UERCAKLM zurückgegeben wurde, um die Warteschlange zu bereinigen.

Beim Bereinigen einer Warteschlange, die einen Engpass im Taskablauf verursacht, werden entsprechende Zeitfenster für Tasks freigegeben, die benötigt werden, um eine Überlastung des Systems zu verhindern. Je mehr Sie eine Sitzungswarteschlange anwachsen lassen, desto größer ist die Wahrscheinlichkeit, dass die im Parameter MAXT festgelegte Obergrenze für Tasks erreicht wird, was dann dazu führen kann, dass eine Warteschlange mit eingehenden Tasks in der lokalen Region nicht angehängt werden kann. Hierbei ist zu beachten, dass bestimmte interne CICS-Anforderungen (wie beispielsweise solche für die

Transaktionen CLS1, CLS2 und CLS3 des LU-Servicemodells) von den Rückgabecodes UERCAKLL und UERCAKLM nicht bereinigt werden.

Wenn eine Warteschlange zuvor bereinigt worden ist (mit UERCAKLL oder UERCAKLM), zum gegenwärtigen Zeitpunkt jedoch keine Anforderungen in der Warteschlange vorhanden sind, überprüfen Sie die Anzahl der erfolgreichen Zuordnungen seit der letzten Warteschlangenbereinigung. Für unspezifische Zuordnungsanforderungen befindet sich die Anzahl in UEPSARC8; für spezifische Modusgruppenanforderungen befindet sich die Anzahl in UEPMAR12. Wenn seit der letzten Bereinigung der Warteschlange keine Anforderungen dieses Typs für diese Verbindung zugeordnet worden sind, ist das Problem, das zuvor zur Bereinigung geführt hat, weiterhin vorhanden, und diese Anforderung sollte mit UERCAPUR zurückgewiesen werden.

Ist anhand der Parameter UEPSARC8 bzw. UEPMAR12 ersichtlich, dass Zuordnungen verarbeitet werden, sollten Sie den Rückgabecode UERCAQUE verwenden, um die Einreihung von Anforderungen in die Warteschlange wieder aufzunehmen. Wird in diesem Fall die Steuerung mit UERCAQUE zurückgegeben, gibt CICS eine entsprechende Informationsnachricht an die Konsole aus, um zu signalisieren, dass Anforderungen nun wieder in die Warteschlange eingereiht werden.

**Anmerkung:** Die Adresse des Statistikdatensatzes für Systemeinträge (UEPCONST) wird sowohl für unspezifische Zuordnungsanforderungen als auch für spezifische Modusgruppenzuordnungsanforderungen bereitgestellt.

Die Adresse des Statistikdatensatzes für Modusgruppen (UEPMODST) wird bei unspezifischen Zuordnungsanforderungen auf Nullen gesetzt. Diese Adresse wird nur dann bereitgestellt, wenn die Anforderung für eine bestimmte Modusgruppe gilt.

Wenn der Exit nach einer erfolgreichen Zuordnung aufgerufen wird, nachdem die Warteschlangensteuerung davor unterdrückt worden war, können Sie den folgenden Rückgabecode verwenden:

#### **UERCNORM**

Dieser Rückgabecode gibt an, dass CICS die normale Verarbeitung für die Verbindung wiederaufnehmen soll. Dies gilt auch für das Einreihen von Anforderungen in die Warteschlange.

### **Statistikfelder in DFHA14DS und DFHA20DS**

Es gibt eine Reihe von Statistikfeldern, die Ihr globales Benutzerexitprogramm für XZIQUE für die Steuerung von Warteschlangen verwenden kann.

#### **A14EALRJ:**

Jedes Mal, wenn ein globales Benutzerexitprogramm für XZIQUE mit einer Anforderung zum Zurückweisen einer Anforderung zurückkehrt, wird der Wert des Felds A14EALRJ in der Verbindungsstatistik für Systemeinträge von CICS erhöht.

Das Feld A14EALRJ (Zuordnung zurückgewiesen) befindet sich im Pseudobereich (DSECT) DFHA14DS und soll Ihnen bei der Optimierung des Warteschlangengrenzwerts helfen. Normalerweise sollte der Wert des Felds A14EALRJ null sein, wenn die Anzahl der Sitzungen und der für eine Verbindung definierte Warteschlangengrenzwert im richtigen Gleichgewicht sind und es zu keinem abnormalen Engpass in der Verbindung gekommen ist. Wenn das Feld für zurückgewiesene Zuordnungen ungleich null ist, deutet dies wahrscheinlich darauf hin, dass eine entsprechende Maßnahme erforderlich ist.

## **A14EQPCT und A20EQPCT:**

Jedes Mal, wenn ein globales Benutzerexitprogramm für XZIQUE mit einer Anforderung zum Bereinigen einer Warteschlange zurückkehrt, erhöht CICS den Wert eines neuen Felds: entweder in der Verbindungsstatistik für Systemeinträge (Feld A14EQPCT) oder in der Verbindungsstatistik für Moduseinträge (Feld A20EQPCT).

### **A14EQPCT**

Zähler, der angibt, wie häufig die Warteschlange für die Verbindung insgesamt bereinigt worden ist.

### **A20EQPCT**

Zähler, der angibt, wie häufig die Modusgruppenwarteschlange bereinigt worden ist.

Ausführliche Informationen zu Statistikfeldern, zu den in ihnen enthaltenen Informationen und dazu, wie sie aktualisiert werden, finden Sie unter ISC/IRC system entry: Resource statistics.

## **Exit XZIQUE**

Der Exit XZIQUE wird aufgerufen, wenn eine Zuordnungsanforderung für eine Sitzung im Begriff ist, in die Warteschlange gestellt zu werden, und wenn eine Zuordnungsanforderung nach vorheriger Unterdrückung der Warteschlangensteuerung erfolgreich ausgeführt wird.

### **Zeitpunkt des Aufrufs**

Sobald Folgendes eintritt:

1. Eine Zuordnungsanforderung für eine Sitzung ist im Begriff, in die Warteschlange gestellt zu werden.
2. Eine Zuordnungsanforderung wird nach vorheriger Unterdrückung der Warteschlangensteuerung erfolgreich ausgeführt.

### **Exitspezifische Parameter**

#### **UEPZDATA**

Adresse des 70-Byte-Bereichs mit den aufgelisteten Informationen. Dieser Bereich wird durch den Pseudobereich (DSECT) im Copybook DFHXZIDS zugeordnet.

### **Von UEPZDATA adressierter Bereich**

#### **UEPSYSID**

4-Byte-System-ID (SYSID) der Verbindung.

#### **UEPREQ**

2-Byte-Code für den Anforderungsursprung mit folgenden möglichen Werten:

- |           |  |
|-----------|--|
| <b>TR</b> | Transaktionsrouting.   |
| <b>FS</b> | Funktionsverlagerung (einschließlich Distributed Program Link).  |
| <b>AL</b> | Andere Arten der übergreifenden Kommunikation (beispielsweise verteilte Transaktionsverarbeitung (DTP) oder CPI Communications). |

#### **UEPREQTR**

4-Byte-ID der anfordernden Transaktion (gilt nur dann, wenn der Code für den Anforderungsursprung FS oder AL lautet).

**UEPTRAN**

4-Byte-ID der weitergeleiteten Transaktion (gilt nur dann, wenn der Code für den Anforderungsursprung TR lautet).

**UEPFLAG**

1-Byte-Flag, das angibt, ob beim letzten Aufruf des Exits der Rückgabecode 8 oder der Rückgabecode 12 ausgegeben wurde.

**UEPRC8**

Beim vorherigen Aufruf hat das Exitprogramm die Steuerung mit dem Rückgabecode 8 an CICS zurückgegeben.

**UEPRC12**

Beim vorherigen Aufruf hat das Exitprogramm die Steuerung mit dem Rückgabecode 12 an CICS zurückgegeben.

**UEPPAD**

1-Byte-Feld zum Auffüllen.

**UEPFSPL**

Adresse der 10-Byte-Parameterliste für die Funktionsverlagerung.

**UEPCONST**

Adresse des 158-Byte-Statistikdatensatzes für Systemeinträge (kann mithilfe von DSECT DFHA14DS zugeordnet werden).

**UEPMODST**

Adresse des 84-Byte-Statistikdatensatzes für Modusgruppen für die im entsprechenden CICS-Profil angegebene Modusgruppe. Dieses Feld gilt nur für APPC-Verbindungen für eine spezifische Zuordnung. Bei LU61, IRC oder unspezifischen APPC-Zuordnungen enthält dieses Feld den Wert null.

Der Statistikdatensatz kann mithilfe von DSECT DFHA20DS zugeordnet werden. Das Feld für den Namen der Modusgruppe (A20MODE) kann Leerzeichen enthalten. Auf den Datensatz folgt ein Vollwort vom Typ 'X'FFFFFFF'.

**UEPSTEX**

6-Byte-Bereich mit zusätzlichen aktuellen Statistikdaten für APPC, die im Statistikdatensatz für Modusgruppen (DFHA20DS) bisher nicht enthalten sind. Bei spezifischen Zuordnungen beziehen sich die Zahlen nur auf die angegebene Modusgruppe. Bei unspezifischen Zuordnungen beziehen sich die Zahlen hingegen auf die gesamte Verbindung; sie geben also die Gesamtergebnisse für jede Modusgruppe wider.

Der 6-Byte-Bereich enthält Folgendes:

**UEPEBND**

Halbwort-Binärfeld mit der Anzahl der gebundenen Sitzungen.

**UEPEWWT**

Halbwort-Binärfeld mit der Anzahl der Konfliktgewinner mit Tasks.

**UEPELWT**

Halbwort-Binärfeld mit der Anzahl der Konfliktverlierer mit Tasks.

#### **UEPEMXQT**

Halbwort-Binärfeld mit der für die Verbindung angegebenen maximalen Warteschlangensteuerungszeit (MAXQTIME in der CONNECTION-Ressourcendefinition).

#### **UEPMDGST**

Adresse einer Gruppe von 84-Byte-Statistikdatensätzen für Modusgruppen - ein Datensatz pro Benutzermodusgruppe für die Verbindung. Dieses Feld gilt nur für APPC-Verbindungen für eine unspezifische Zuordnung. Bei LU61, IRC und spezifischen APPC-Zuordnungen enthält dieses Feld den Wert null.

Jeder Statistikdatensatz kann mithilfe von DSECT DFHA20DS zugeordnet werden. Das Feld für den Namen der Modusgruppe (A20MODE) kann Leerzeichen enthalten. Das Ende der Datensatzgruppe wird durch ein Vollwort vom Typ X'FFFFFFFF' angezeigt.

**Daten für unspezifische Zuordnungen:** Die folgenden drei Felder enthalten Daten im Zusammenhang mit MRO, LU6.1 und unspezifischen APPC-Zuordnungen:

#### **UEPSAQTS**

Doppelwort-Binärfeld mit der Zeitmarke des TCT-Systemeintrags, die angibt, zu welchem Zeitpunkt die Warteschlange für unspezifische Anforderungen gestartet wurde.

#### **UEPSACNT**

Halbwort-Binärfeld mit der Anzahl aller unspezifischen Zuordnungen, die seit dem Start der Warteschlange verarbeitet worden sind (die Startzeit finden Sie in UEPSAQTS).

#### **UEPSARC8**

Halbwort-Binärfeld mit der Anzahl der Sitzungen, die seit der letzten Bereinigung der Warteschlange infolge eines Rückgabecodes vom Typ UEPCAKLL an CICS freigegeben worden sind.

**Daten für spezifische Zuordnungen:** Die nachstehenden drei Felder enthalten Daten im Zusammenhang mit spezifischen Modusgruppenzuordnungen. Sie gelten nur dann, wenn UEPMODST ungleich null ist (also dann, wenn die Adresse der entsprechenden Modusgruppenstatistiken enthalten ist).

#### **UEPMAQTS**

Doppelwort-Binärfeld mit der Zeitmarke des TCT-Moduseintrags, die angibt, zu welchem Zeitpunkt die Modusgruppenwarteschlange für diese spezielle Modusgruppe gestartet wurde.

#### **UEPMACNT**

Halbwort-Binärfeld mit der Anzahl aller spezifischen Zuordnungen für diese Modusgruppe, die seit dem Start der Warteschlange verarbeitet worden sind (die Startzeit finden Sie in UEPMAQTS).

#### **UEPMAR12**

Halbwort-Binärfeld mit der Anzahl der Modusgruppensitzungen, die seit der letzten Bereinigung der Warteschlange infolge eines Rückgabecodes vom Typ UEPCAKLL an CICS freigegeben worden sind.

### **UEPQUELM**

Halbwort-Binärfeld mit dem für diese Verbindung angegebenen Warteschlangengrenzwert (QUEUELIMIT in der CONNECTION-Definition).

### **Rückgabecodes**

Verwenden Sie bei Zuordnungen, die im Begriff sind, in die Warteschlange gestellt zu werden, einen der folgenden Rückgabecodes:

#### **UERCAQUE**

Zuordnungsanforderung in die Warteschlange stellen.

#### **UERCAPUR**

Zuordnungsanforderung mit SYSIDERR zurückweisen.

#### **UERCAKLL**

Diese Zuordnungsanforderung mit SYSIDERR zurückweisen. Alle anderen Zuordnungsanforderungen in der Warteschlange bereinigen und eine Informationsnachricht an die Bedienerkonsole senden. Darüber hinaus wird SYSIDERR von CICS auch an alle Anwendungsprogramme zurückgegeben, die auf die bereinigten Zuordnungsanforderungen warten.

#### **UERCAKLM**

Diese Zuordnungsanforderung für die Modusgruppe zurückweisen und SYSIDERR zurückgeben. Alle anderen in die Warteschlange eingereichten Zuordnungsanforderungen für die in dieser Zuordnungsanforderung angegebene Modusgruppe bereinigen und eine Informationsnachricht an die Bedienerkonsole senden. Modusgruppe nach einem Intervall wiederholen.

#### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

Wenn nach der Verwendung von UERCAKLL oder UERCAKLM in einem vorherigen Aufruf des Exits eine erfolgreiche Zuordnung stattgefunden hat, verwenden Sie einen der folgenden Rückgabecodes:

#### **UERCNORM**

Wiederaufnahme des normalen Betriebs der Verbindung oder der Modusgruppe.

#### **UERCAPUR**

Zuordnungsanforderung mit SYSIDERR zurückweisen.

### **XPI-Aufrufe**

Es können alle verwendet werden.

## **Entwerfen eines globalen Benutzerexitprogramms für XZIQUE**

Die Funktionen des Exits XZIQUE sollten mit folgender Zielsetzung gestaltet werden:

1. Zur Steuerung der Anzahl der Tasks (und der Menge der zugeordneten Ressourcen), die in einer Warteschlange auf eine freie systemübergreifende Sitzung warten. Tasks im Wartestatus können zu einer Leistungsver schlechterung auf dem lokalen System führen.
2. Zur Erkennung einer unzureichenden Antwort vom empfangenden (fernen) System und zur entsprechenden Benachrichtigung des Bedieners (oder des automatischen Betriebsprogramms).
3. Zur Veranlassung, dass CICS eine Nachricht ausgibt, nachdem die Verbindung den Normalbetrieb wiederaufgenommen hat.

Die Parameterliste für den globalen Benutzerexit XZIQUE wurde entsprechend erstellt, um diese Ziele zu unterstützen.

## Beispielexitprogramm für XZIQUE

Für den Exit XZIQUE wird das Beispielexitprogramm DFH\$XZIQ mit CICS Transaction Server for z/OS, Version 5 Release 5 bereitgestellt. Dieses Beispielprogramm dient als Grundlage für die Gestaltung Ihres eigenen globalen Benutzerexitprogramms. Das Beispiel wird in der Bibliothek CICSTS55.CICS.SDFHSAMP bereitgestellt. Das DSECT-Objekt DFHXZIDS, das vom Beispielprogramm zum Zuordnen des von UEPZDATA adressierten Bereichs verwendet wird, wird in der Bibliothek CICSTS55.CICS.SDFHMAC bereitgestellt.

Weitere Informationen zu DFH\$XZIQ finden Sie unter MRO and APPC session queue management sample exit program: DFH\$XZIQ.

## Aspekte beim Programmmentwurf

Die am Exit XZIQUE übergebenen Informationen sollen Ihrem globalen Benutzerexitprogramm für XZIQUE Folgendes ermöglichen:

- Vermeidung falscher Diagnosen von Problemen mit der Verbindung durch Unterscheidung schlechter Antwortzeiten von einem vollständigen Engpass.
- Sicherstellen, dass eine Programmverbindung nach Lösung eines Problems auf einem fernen System den normalen Betrieb schnell und ohne Bedienereingriff wiederaufnimmt.

## Hinweise zur Verwendung von IRC/ISC-Statistiken

CICS fügt einen Eintrag für nicht erfüllte Zuordnungsanforderungen zur *unspezifischen (generischen) Zuordnungswarteschlange* und zu den *spezifischen Zuordnungswarteschlangen für Modusgruppe* hinzu.

### unspezifische (generische) Zuordnungswarteschlange

In diese einzelne Warteschlange werden alle unspezifischen Zuordnungsanforderungen eingereiht. CICS gibt die Gesamtzahl der Einträge in dieser Warteschlange im Feld A14ESTAQ der Systemeintragsstatistik an. Auf dieses Feld hat Ihr globales Benutzerexitprogramm über die Adresse der Systemeintragsstatistik Zugriff. Diese Adresse wird in UEPCONST übergeben.

### Spezifische Zuordnungswarteschlangen für Modusgruppen

Spezifische Zuordnungsanforderungen werden in die entsprechenden Modusgruppenwarteschlangen eingereiht, wobei es für jeden Modusgruppennamen eine eigene Warteschlange gibt. CICS gibt die Gesamtzahl der Einträge in allen diesen Warteschlangen als einzelne Summe im Feld A20ESTAQ der Moduseintragsstatistik an. Auf dieses Feld hat Ihr globales Benutzerexitprogramm über die Adresse der Moduseintragsstatistik Zugriff. Diese Adresse wird in UEPMODST übergeben.

---

## Exit XISQUE zum Verwalten systemübergreifender IPIC-Warteschlangen

Mithilfe des Exits XISQUE können Sie die Warteschlangensteuerung für IPIC-Verbindungen (IP-Interkonnektivität) steuern.

Der Exit XISQUE steuert die folgenden Anforderungen bzw. Befehle in der Warteschlange für die IPIC-Verbindung:

- DPL-Anforderungen (Distributed Program Link) für Sitzungen.
- Anforderungen für Transaktionsweiterleitung.



- Anforderungen für Funktionsverlagerung.
- Befehle vom Typ START oder CANCEL.

Mithilfe des Exits XISQUE können Sie Probleme bei der Warteschlangensteuerung (Engpässe) bereits in einem frühen Stadium erkennen.

Mithilfe von XISQUE können Zuordnungsanforderungen in Abhängigkeit von der Warteschlangenlänge entweder eingereiht oder zurückgewiesen werden. Darüber hinaus ermöglicht der Exit auch das Beenden und erneute Herstellen von IPIC-Verbindungen, bei denen ein Engpass auftritt.

## Exit XISQUE

Sofern er aktiviert ist, wird der globale Benutzerexit XISQUE aufgerufen, wenn CICS versucht, eine Sitzung anzufordern, aber keine freie Sitzung verfügbar ist, oder wenn eine Zuordnungsanforderung eine freie Sitzung findet, nachdem die Warteschlangensteuerung zuvor unterdrückt worden war. Die exitspezifischen Parameter, Rückgabecodes und Informationen zu XPI-Aufrufen werden erläutert.

### Zeitpunkt des Aufrufs

Der Exit XISQUE wird in folgenden Situationen aufgerufen:

1. CICS versucht, eine Sitzung für eine IPIC-Verbindung zu einem fernen System anzufordern, aber es ist keine freie Sitzung verfügbar. Der Exit wird unabhängig davon aufgerufen, ob Sie die Option QUEUELIMIT in der IPCONN-Definition angegeben haben und ob der Grenzwert überschritten wurde.

Anforderungen für IPIC-Sitzungen treten auf, wenn eine der folgenden Anforderungen bzw. einer der folgenden Befehle über eine IPIC-Verbindung verwendet wird:

- DPL-Anforderung (Distributed Program Link).
  - Befehl START oder CANCEL
  - Transaktionsrouting-Anforderung
  - Funktionsverlagerungsanforderung für Dateisteuerung, transiente Daten oder temporären Speicher
2. Eine IPIC-Zuordnungsanforderung findet eine freie Sitzung, nachdem die Warteschlange für die IPIC-Verbindung durch einen vorherigen Aufruf des Exitprogramms bereinigt worden ist. In diesem Fall kann das Exitprogramm angeben, dass CICS die Verarbeitung normal fortsetzen und bei Bedarf die Warteschlangensteuerung wiederaufnehmen soll.

### Exitspezifische Parameter

#### UEPISDATA

Adresse des 78-Byte-Bereichs. Dieser Bereich wird vom Pseudobereich (DSECT) im Copybook DFHXIQDS zugeordnet.

#### Von UEPISDATA adressierter Bereich:

#### UEPREQ

2-Byte-Code für den Anforderungsursprung mit folgenden möglichen Werten:

- |           |   |
|-----------|---|
| <b>AL</b> | Andere Arten der übergreifenden Kommunikation (beispielsweise START-Befehle). |
| <b>FS</b> | Funktionsverlagerung und Distributed Program Link (DPL).                      |

**TR** Transaktionsrouting.

**UEIPCNM**

8-Byte-Name der IPIC-Verbindung (IPCONN).

**UEPREQTR**

4-Byte-ID der anfordernden Transaktion.

**UEPFLAG**

1-Byte-Flag, das angibt, ob beim letzten Aufruf des Exits der Rückgabecode 8 ausgegeben wurde.

**UEPRC8**

Beim vorherigen Aufruf hat das Exitprogramm die Steuerung mit dem Rückgabecode 8 an CICS zurückgegeben.

**UEPFSPL**

Adresse der 10-Byte-Parameterliste für die DPL-Anforderung.

**UEPCNST**

Adresse des 504-Byte-IPCONN-Statistikdatensatzes. Dieser Statistikdatensatz kann mithilfe von DSECT DFHISRDS zugeordnet werden.

**UEPEMXQT**

Halbwort-Binärfeld mit der in der IPCONN-Ressourcendefinition angegebenen maximalen Warteschlangensteuerungszeit (MAXQTIME).

**UEPSAQTS**

Doppelwort-Binärfeld mit der Zeitmarke der installierten IPCONN-Ressourcendefinition, die angibt, zu welchem Zeitpunkt die Warteschlange für Zuordnungsanforderungen gestartet wurde.

**UEPSACNT**

Halbwort-Binärfeld mit der Anzahl der Zuordnungsanforderungen, die seit dem Start der Warteschlange verarbeitet worden sind. Die Startzeit wird in UEPSAQTS angegeben.

**UEPSARC8**

Halbwort-Binärfeld mit der Anzahl der Sitzungen, die seit der letzten Bereinigung der Warteschlange infolge eines Rückgabecodes vom Typ UEPCAKLL freigegeben worden sind.

**UEPQUELM**

Halbwort-Binärfeld mit dem in der IPCONN-Ressourcendefinition angegebenen Warteschlangengrenzwert (QUEUELIMIT).

**Rückgabecodes**

Verwenden Sie bei Zuordnungen, die im Begriff sind, in die Warteschlange gestellt zu werden, einen der folgenden Rückgabecodes:

**UERCAQUE**

Zuordnungsanforderung in die Warteschlange stellen.

**UEPCAKLL**

Diese Zuordnungsanforderung mit SYSIDERR zurückweisen. Alle anderen Zuordnungsanforderungen in der Warteschlange bereini-

gen und eine Informationsnachricht an die Bedienerkonsole senden. Darüber hinaus wird SYSIDERR von CICS auch an alle Anwendungsprogramme zurückgegeben, die auf die bereinigten Zuordnungsanforderungen warten.

#### **UERCAPUR**

Zuordnungsanforderung mit SYSIDERR zurückweisen.

#### **UERCPURG**

Task während des XPI-Aufrufs bereinigt.

Wenn nach der Verwendung von UERCAKLL in einem vorherigen Aufruf des Exits eine erfolgreiche Zuordnung stattgefunden hat, können Sie einen normalen Rückgabecode (UERCNORM) oder den Rückgabecode mit SYSIDERR (UERCAPUR) verwenden:

#### **UERCNORM**

Wiederaufnahme des normalen Betriebs der IPIC-Verbindung (IPCONN).

#### **UERCAPUR**

Zuordnungsanforderung mit SYSIDERR zurückweisen.

#### **XPI-Aufrufe**

Es können alle verwendet werden.

## **Verwendung eines globalen Benutzerexitprogramms für XISQUE**

Wenn der Exit XISQUE aktiviert ist, kann das globale Benutzerexitprogramm für XISQUE verwendet werden, um den Status von Zuordnungswarteschlangen für IPIC-Verbindungen (IPCONN) im lokalen System zu überprüfen. Die Parameterliste, die beim Aufruf an das Exitprogramm übergeben wird, enthält Daten zu einer bestimmten Zuordnungsanforderung und IPIC-Verbindung (IPCONN).

Anhand der in der Parameterliste übergebenen Informationen kann Ihr globales Benutzerexitprogramm (beispielsweise auf Grundlage der Warteschlangenlänge) entscheiden, ob CICS die Zuordnungsanforderung in die Warteschlange stellen soll. Das Programm kommuniziert die getroffene Entscheidung an CICS mithilfe eines der folgenden Rückgabecodes:

#### **UERCAQUE**

CICS soll die Zuordnungsanforderung in die Warteschlange stellen.

Die Gesamtzahl der momentan in der Warteschlange befindlichen Zuordnungsanforderungen für die Verbindung wird im Feld ISR\_CURRENT\_QUEUE\_D\_ALLOCATES des IPCONN-Statistikdatensatzes angegeben, der durch den exitspezifischen Parameter UEPCONST adressiert wird. Ausführliche Informationen hierzu finden Sie im Pseudobereich (DSECT) DFHISRDS.

CICS übergibt außerdem im Feld UEPQUELM den Wert der Option QUEUELIMIT der IPCONN-Ressourcendefinition an das Exitprogramm. Wenn der Warteschlangengrenzwert nicht erreicht wurde, können Sie die Steuerung mit dem Rückgabecode UERCAQUE an CICS zurückgeben.

#### **UERCAPUR**

CICS soll die Zuordnungsanforderung zurückweisen, SYSIDERR an das Anwendungsprogramm zurückgeben und die vorhandene Warteschlange unverändert lassen.

Wenn die Anzahl der Zuordnungsanforderungen in der Warteschlange den in der Option QUEUELIMIT der IPCONN-Definition festgelegten Grenzwert er-

reicht hat, können Sie anfordern, dass CICS die Anforderung zurückweisen soll. Sie sollten jedoch zunächst prüfen, ob der Status der Verbindung zufriedenstellend ist. Es sollte also überprüft werden, ob die Geschwindigkeit bei der Zuordnung von Sitzungen akzeptabel ist. Verwenden Sie die Uhrzeit, zu der die Warteschlange gestartet wurde, die aktuelle Uhrzeit und die Gesamtzahl der seit dem Start der Warteschlange verarbeiteten Zuordnungen, um die Geschwindigkeit zu ermitteln, mit der CICS Anforderungen verarbeitet. Folgende Felder sind relevant: UEPSAQTS und UEPSACNT.

Um zu ermitteln, ob CICS Sitzungsanforderungen über diese IPIC-Verbindung (IPCONN) mit akzeptabler Geschwindigkeit zuordnet, können Sie die berechnete Zeit mit einem der folgenden Werte vergleichen:

1. Mit dem Wert der Option MAXQTIME der IPCONN-Ressourcendefinition, der im exitspezifischen Parameter UEPEMXQT übergeben wird.
2. Mit einem anderen voreingestellten Zeitwert.

Wenn die anhand dieser Formel ermittelte Verarbeitungszeit für Sie akzeptabel ist, geben Sie die Steuerung mit Rückgabecode UERCAPUR an CICS zurück; auf diese Weise wird nur diese eine Anforderung bereinigt.

### **UERCAKLL**

Diese Anforderung zurückweisen, alle anderen Zuordnungsanforderungen in der Warteschlange für diese IPIC-Verbindung (IPCONN) bereinigen und eine Informationsnachricht an die Bedienerkonsole senden.

Wenn der Warteschlangengrenzwert erreicht ist und die Leistung der Zuordnungsverarbeitung unterhalb der in Ihrem Benutzerexitprogramm definierten akzeptablen Grenzwerte liegt, können Sie alle Zuordnungsanforderungen in der Warteschlange bereinigen, indem Sie den Rückgabecode UERCAKLL angeben.

Der Rückgabecode UERCAKLL bewirkt außerdem, dass CICS folgende Aktionen ausführt:

- Rückgabe von SYSIDERR an alle Anwendungsprogramme, die auf die bereinigten Zuordnungsanforderungen warten.
- Bei nachfolgenden Aufrufen an das Exitprogramm für XISQUE: Festlegen des Parameters UEFLAG auf UEPRC8, um anzugeben, dass zuvor UERCAKLL zurückgegeben wurde, um die Warteschlange zu bereinigen.

Beim Bereinigen einer Warteschlange, die einen Engpass im Taskablauf verursacht, werden entsprechende Zeitfenster für Tasks freigegeben, die benötigt werden, um eine Überlastung des Systems zu verhindern. Je mehr Sie eine Sitzungswarteschlange anwachsen lassen, desto größer ist die Wahrscheinlichkeit, dass die im Parameter MAXT festgelegte Obergrenze für Tasks erreicht wird, was dann dazu führen kann, dass eine Warteschlange mit eingehenden Tasks in der lokalen Region nicht angehängt werden kann.

Wenn eine Warteschlange zuvor bereinigt worden ist (mit UERCAKLL), zum gegenwärtigen Zeitpunkt jedoch keine Anforderungen in der Warteschlange vorhanden sind, überprüfen Sie die Anzahl der Sitzungen, die seit der letzten Warteschlangenbereinigung freigegeben worden sind. Diese Anzahl wird in UEPSARC8 angegeben. Wenn seit der letzten Bereinigung der Warteschlange keine Sitzungen für diese IPIC-Verbindung (IPCONN) freigegeben worden sind, ist das Problem, das zuvor zur Bereinigung geführt hat, weiterhin vorhanden, und diese Anforderung sollte mit UERCAPUR zurückgewiesen werden.

Ist anhand des Parameters UEPSARC8 ersichtlich, dass Sitzungen freigegeben werden, sollten Sie den Rückgabecode UERCAQUE verwenden, um die Einrei-

hung von Anforderungen in die Warteschlange wiederaufzunehmen. Wird in diesem Fall die Steuerung mit UERCAQUE zurückgegeben, gibt CICS eine entsprechende Informationsnachricht an die Konsole aus, um zu signalisieren, dass Anforderungen nun wieder in die Warteschlange eingereicht werden.

#### **UERCNORM**

CICS soll die normale Verarbeitung für die Verbindung wiederaufnehmen. Dies gilt auch für das Einreihen von Anforderungen in die Warteschlange.

Wenn der Exit nach einer erfolgreichen Zuordnung aufgerufen wird, nachdem die Warteschlangensteuerung davor unterdrückt worden war, verwenden Sie den Rückgabecode UERCNORM.

## **Statistikfelder in DFHISRDS**

Die folgenden Felder in der IPCONN-Statistik können das globale Benutzerexitprogramm für XISQUE dabei unterstützen, Zuordnungswarteschlangen effizient zu steuern:

#### **ISR\_XISQUE\_ALLOC\_REJECTS**

Jedes Mal, wenn ein globales Benutzerexitprogramm für XISQUE mit einer Anforderung zum Zurückweisen einer Anforderung zurückkehrt, erhöht CICS den Wert dieses Felds, das bereitgestellt wird, um Sie dabei zu unterstützen, den Warteschlangengrenzwert zu optimieren. Normalerweise sollte der Wert des Felds ISR\_XISQUE\_ALLOC\_REJECTS null sein, wenn die Anzahl der Sitzungen und der in der IPCONN-Definition angegebene Warteschlangengrenzwert im richtigen Gleichgewicht sind und es zu keinem abnormalen Engpass in der Verbindung gekommen ist. Wenn das Feld für zurückgewiesene Zuordnungen ungleich null ist, deutet dies darauf hin, dass wahrscheinlich eine entsprechende Maßnahme erforderlich ist.

#### **ISR\_XISQUE\_ALLOC\_QPURGES**

Jedes Mal, wenn ein globales Benutzerexitprogramm für XISQUE mit einer Anforderung zum Bereinigen einer Warteschlange zurückkehrt, erhöht CICS den Wert dieses Felds.

Ausführliche Informationen zu den Feldern in der IPCONN-Statistik finden Sie unter IPCONN report. Die IPCONN-Statistik wird durch den Pseudobereich (DSECT) DFHISRDS zugeordnet.

## **Entwerfen eines globalen Benutzerexitprogramms für XISQUE**

Das Exitprogramm für XISQUE sollte für Folgendes konzipiert werden:

1. Steuern der Anzahl der Tasks (und der zugeordneten Ressourcenmenge), die in einer Warteschlange auf eine freie IPIC-Sitzung warten. Tasks im Wartestatus können zu einer Leistungsver schlechterung auf dem lokalen System führen.
2. Erkennen einer unzureichenden Antwort vom fernen System und entsprechende Benachrichtigung des Bedieners (oder des automatischen Betriebsprogramms).
3. Veranlassen, dass CICS eine Nachricht ausgibt, nachdem die IPIC-Verbindung (IPCONN) den Normalbetrieb wiederaufgenommen hat.

Die Parameterliste für XISQUE ist auf die Unterstützung dieser Ziele ausgelegt. Die darin enthaltenen Informationen ermöglichen dem Exitprogramm Folgendes:

- Vermeidung falscher Diagnosen von Verbindungsproblemen durch Unterscheidung schlechter Antwortzeiten von einem vollständigen Engpass.

- Sicherstellen, dass eine Programmverbindung nach Lösung eines Problems auf einem fernen System den normalen Betrieb schnell und ohne Bedienereingriff wiederaufnimmt.

## Verwendung von IPCONN-Statistiken

Bei der Entscheidung darüber, welche Anforderungen zurückgewiesen werden sollen, welche in die Warteschlange gestellt werden sollen und welche Warteschlangen bereinigt werden sollen, berücksichtigt das Exitprogramm wahrscheinlich die Anzahl der momentan in der Warteschlange befindlichen Zuordnungsanforderungen für die Verbindung. Alle Zuordnungsanforderungen für eine bestimmte IPCONN werden in eine einzelne Warteschlange eingereiht, die für diese IPCONN spezifisch ist. CICS stellt die Gesamtzahl der Einträge in dieser Warteschlange im IPCONN-Statistikfeld `ISR_CURRENT_QUEUED_ALLOCATES` zur Verfügung. Das Exitprogramm kann über die Adresse der IPCONN-Statistik auf dieses Feld zugreifen. Diese Adresse wird im exitspezifischen Parameter `UEPCONST` übergeben.

## Beispielexitprogramm für XISQUE

CICS stellt ein Beispielexitprogramm für XISQUE bereit. Dieses Beispielprogramm heißt `DFH$XISQ` und kann als Grundlage für ein eigenes Programm verwendet werden. Es wird in der Bibliothek `CICSTS55.CICS.SDFHSAMP` bereitgestellt. Der Pseudobereich (DSECT), der vom Beispielprogramm zum Zuordnen des von `UEPISDATA` adressierten Bereichs verwendet wird, heißt `DFHXIQDS` und wird in der Bibliothek `CICSTS55.CICS.SDFHMAC` bereitgestellt.

Weitere Informationen zu `DFH$XISQ` finden Sie unter `Session queue management sample exit program for IPIC connections: DFH$XISQ`.

---

## Exit XXRSTAT für das Programm zur Verarbeitung von XRF-Anforderungen

Dieser Exit wird aufgerufen, wenn ein Fehler bei z/OS Communications Server oder eine unfreiwillige Systemübernahme durch z/OS Communications Server auftritt.

Mithilfe von XXRSTAT können Sie entscheiden, ob CICS beendet werden soll, wenn eine der folgenden beiden Situationen eintritt:

- CICS wird vom Exit `TPEND` über einen Fehler bei z/OS Communications Server benachrichtigt.
- Es hat eine unfreiwillige Systemübernahme stattgefunden. Eine unfreiwillige Systemübernahme kann stattfinden, wenn Sie Release 3.4.0 oder höher verwenden und eine z/OS Communications Server-Anwendung mit derselben Anwendungs-ID (APPLID) wie die Anwendungs-ID des ausführenden CICS-Systems die Steuerung sämtlicher Sitzungen des ausführenden CICS-Systems übernimmt.

Mithilfe von XXRSTAT können Sie entscheiden, ob das System, das auf diese Weise übernommen wurde, fortgesetzt oder beendet werden soll.

Um potenzielle Integritätsrisiken zu vermeiden, wird CICS nach einer solchen Übernahme standardmäßig ohne Speicherauszug beendet. Wenn CICS mit einem Speicherauszug beendet werden soll, muss das Exitprogramm `UERCABDU` zurückgeben. CICS wird mit dem Abbruchcode beendet, der vom Exitprogramm angegeben wurde.

Wenn CICS nach einer unfreiwilligen Systemübernahme fortgesetzt werden soll, muss das Exitprogramm `UERCCOIG` zurückgeben. Die Nachricht `DFHZC0101`

wird ausgegeben, und CICS setzt die Verarbeitung ohne z/OS Communications Server-Unterstützung fort. Die Anwendung, die die unfreiwillige Übernahme durchführen wollte, übernimmt die Steuerung sämtlicher z/OS Communications Server-Sitzungen.

**Anmerkung:** Wird zugelassen, dass CICS nach einer unfreiwilligen Systemübernahme die Verarbeitung fortsetzt, kann dies unter Umständen zu Integritätsproblem führen. Deshalb wird hiervon abgeraten. Verwenden Sie RACF zum Schutz Ihrer CICS-Anwendungs-IDs (APPLIDs).

## Exit XXRSTAT

Dieser Exit wird nach einem Ausfall von z/OS Communications Server oder einer Systemübernahme durch z/OS Communications Server aufgerufen.

### Zeitpunkt des Aufrufs

Nach Eintritt einer der beiden folgenden Situationen:

- CICS wird vom Exit TPEND über einen Ausfall von z/OS Communications Server benachrichtigt.
- Es hat eine Systemübernahme stattgefunden.

### Exitspezifische Parameter

#### UEPERRA

Adresse einer Parameterliste, die Folgendes enthält:

#### UEPGAPLD

Adresse der generischen 8-Byte-Anwendungs-ID.

#### UEPSAPLD

Adresse der spezifischen 8-Byte-Anwendungs-ID.

#### UEPDOMID

Adresse der 4-Byte-Domänen-ID.

#### UEPERRID

Adresse der 4-Byte-Fehler-ID.

### Anmerkung:

1. Für diese Parameterliste wird kein Pseudobereich (DSECT) bereitgestellt. Für den Zugriff auf die benannten Felder müssen Sie Ihren eigenen DSECT codieren.
2. Wenn z/OS Communications Server fehlgeschlagen ist, lautet die Domänen-ID 'ZC ' (Z in Großschreibung, C in Großschreibung und zwei Leerzeichen), und die Fehler-ID ist die Zeichenfolge '3443'.

### Rückgabecodes

#### UERCNORM

Systemaktion ausführen. Die Systemaktion hängt von der Ursache für den Aufruf des Exits ab.

- Für XRF im Falle eines Ausfalls von z/OS Communications Server: CICS setzt die Verarbeitung fort, als wäre das Exitprogramm nicht aufgerufen worden.
- Für persistente Sitzungen von z/OS Communications Server im Falle einer Systemübernahme: CICS wird ohne Speicherauszug abgebrochen.

#### UERCCOIG

Ignorieren.

**UERCABNO**

CICS ohne Speicherauszug abbrechen.

**UERCABDU**

CICS mit Speicherauszug abbrechen.

**UERCPURG**

Task während des XPI-Aufrufs bereinigt.

**XPI-Aufrufe**

Es können alle verwendet werden.



---

## Bemerkungen

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

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

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

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

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

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

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

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

*IBM Director of Licensing  
IBM 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 Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

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

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

#### **COPYRIGHTLIZENZ:**

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmiertechniken 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 5 zu schreiben, sind in folgenden Abschnitten der Online-Produktdokumentation enthalten:

- Developing applications

- Developing system programs
- Securing overview
- Developing for external interfaces
- Reference: application development
- Reference: system programming
- Reference: connectivity

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

- Troubleshooting and support
- Reference: diagnostics

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 5 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 5 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 unter [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

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

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.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

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

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

## 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 ei-

nem 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 Privacy Policy und in der IBM Online Privacy Statement im Abschnitt „Cookies, Web-Beacons und sonstige Technologien“ sowie auf der Seite IBM Software Products and Software-as-a-Service Privacy Statement.

---

# Index

## A

AILDELAY, Systeminitialisierungsparameter 229  
Alphabetische Liste der globalen Benutzerexits 1  
APPC-Verbindungen  
  systemübergreifende Warteschlangen 294  
XZIQUE, globaler Benutzerexit  
  zum Steuern von systemübergreifenden Warteschlangen 294

## B

Basic Mapping Support (BMS)  
  globale Benutzerexitpunkte 12  
Beispielprogramme  
  für globale Benutzerexits  
    DFH\$WBGA, für den Exit XWBO-PEN 141

## C

CICS-Webunterstützung  
  Benutzerexits XWBAUTH, XWBO-PEN, XWBSNDO 140  
  Benutzerexits XWBOPEN, XWBSNDO 144, 146

## D

Datentabellen 19  
DFH\$ICCN, Beispiel für globales Benutzerexitprogramm 177  
DFH\$WBGA, Beispiel für globales Benutzerexitprogramm 141

## F

Funktionsliste der globalen Benutzerexits 1

## G

Globale Benutzerexits  
  Beispielprogramme  
    DFH\$ICCN 177  
    DFH\$WBGA 141  
    für XFCREQ 91, 94  
    für XFCREQC 91, 94  
    für XICEREQ 177  
    für XICEREQC 177  
    für XPCREQ 218  
    für XPCREQC 218  
    für XTDEREQ 278  
    für XTDEREQC 278  
    für XTSEREQ 262  
    für XTSEREQC 262

Globale Benutzerexits (*Forts.*)  
  Beispielprogramme (*Forts.*)  
    zum Ändern von TS-Anforderungen 262  
  Exitpunkte  
    bei RLS-Koexistenz für Dateisteuerung 138  
    Brückeneinrichtung erstellen 17  
    Brückeneinrichtung löschen 17  
    für Bedingung "Terminal unbekannt" 266  
    im Dateisteuerprogramm für Dateistatus 111  
    im Dateisteuerprogramm für die EXEC-Schnittstelle 81, 94  
    im Dateisteuerprogramm für Öffnen/Schließen 121  
    im Dateisteuerprogramm für Wiederherstellung 127  
    im DBCTL-Schnittstellensteuerprogramm 25  
    im DBCTL-Überwachungsprogramm 26  
    im DL/I-Schnittstellenprogramm 29  
    im Einreihungsprogramm der EXEC-Schnittstelle 48  
    im Empfangsprogramm für Quiesce-Anforderungen zur Dateisteuerung 123  
    im EXEC-Schnittstellenprogramm 57  
    im EXEC-Schnittstellenprogramm für temporären Speicher 252  
    im EXEC-Schnittstellenprogramm für transiente Daten 278  
    im Intervallsteuerprogramm 155  
    im Intervallsteuerprogramm für die EXEC-Schnittstelle 157  
    im Programm für Aktivitätsschleispunkte 10  
    im Programm für Guten-Morgen-Nachricht 140  
    im Programm für systemübergreifende Kommunikation 148  
    im Programm für transiente Daten 275  
    im Programm zur Verarbeitung von XRF-Anforderungen 308  
    im Programmsteuerprogramm 206  
    im Ressourcenmanagerschnittstellenprogramm 226  
    im Sendeprogramm für Quiesce-Anforderungen zur Dateisteuerung 126  
    im SNA-Arbeitsbereichsmodul 292  
    im SNA-LU-Verwaltungsprogramm 291

Globale Benutzerexits (*Forts.*)  
  Exitpunkte (*Forts.*)  
    im Systemabschlussprogramm 246  
    im Systemwiederherstellungsprogramm 242  
    im Terminalsteuerprogramm 264  
    im Terminalzuordnungsprogramm 263  
    im Wiederherstellungsprogramm für Benutzerprotokolldatensätze 287  
    in Anwendungsbezugsdaten 11  
    in BMS 12  
    in Datentabellenprogrammen 19  
    in der CICS-Webunterstützung 140  
    in der Dateisteuerungsdomäne 66  
    in der Dispatcherdomäne 28  
    in der Front-End-Programmier-schnittstelle (FEPI) 62  
    in der Ladedomäne 178  
    in der Nachrichtendomäne 184  
    in der Pipelinedomäne 191  
    in der Protokollmanagerdomäne 180  
    in der Sicherheitsmanagerdomäne 237  
    in der Speicherauszugsdomäne 40  
    in der Statistikdomäne 240  
    in der temporären Speicherdomäne 246  
    in der Transaktionsmanagerdomäne 273  
    in der Überwachungsdomäne 189  
    in Ressourcenmanagementmodulen 229  
    zur Datentabellenverwaltung 19  
  XDLIPRE  
    Beispiel 34  
  XSZARQ  
    exitspezifische Parameter 64  
    Übersicht 64  
    UEPSZACN, Parameter 65  
  XSZBRQ  
    Übersicht 63  
    UEPSZACT, Parameter 65  
Globale Benutzerexits, alphabetische Liste 1  
Globale Benutzerexits, Funktionsliste 1

## H

HTTP-Client, Exit XWBOPEN zum Öffnen 144

## I

- IPIC-Verbindungen
  - systemübergreifende Warteschlangen 302
- XISQUE, globaler Benutzerexit
  - zum Steuern von systemübergreifenden Warteschlangen 302

## K

- Koexistenz von Local DL/I und DBCTL
  - XDLIPRE zum Ändern des PSB für Terminierung 34

## M

- MRO-Verbindungen
  - systemübergreifende Warteschlangen 294
- XZIQUE, globaler Benutzerexit
  - zum Steuern von systemübergreifenden Warteschlangen 294

## P

- PSB (Programmspezifikationsblock)
  - XDLIPRE zum Ändern des PSB für Terminierung 34

## S

- Systeminitialisierungsparameter
  - AILDELAY 229
  - TBEXITS 129, 289
- Systemübergreifende Kommunikation (ISC) über systemübergreifende TCP-/IP-Warteschlangen
  - Steuerung der Länge
    - mithilfe des globalen Benutzerexits XISQUE 302
- Systemübergreifende Kommunikation (ISC) über TCP-/IPIC-Verbindungen
  - systemübergreifende Warteschlangen 302
- XISQUE, globaler Benutzerexit
  - zum Steuern von systemübergreifenden Warteschlangen 302
- Systemübergreifende Warteschlangen
  - Steuerung der Länge
    - mithilfe des globalen Benutzerexits XISCONA 148
    - mithilfe des globalen Benutzerexits XZIQUE 294

## T

- TBEXITS, Systeminitialisierungsparameter 129, 289

## U

- UEPSZACN, exitspezifischer Parameter für XSZARQ 65

- UEPSZACT, exitspezifischer Parameter für XSZBRQ 65

## W

- Warteschlangen für systemübergreifende Sitzungen
  - Steuerung der Länge
    - mithilfe des globalen Benutzerexits XISCONA 148
    - mithilfe des globalen Benutzerexits XISQUE 302
    - mithilfe des globalen Benutzerexits XZIQUE 294

## X

- XALCAID, globaler Benutzerexit 263
- XALTENF, globaler Benutzerexit 268
- XAPADMGR, globaler Benutzerexit 11
- XBMIN, globaler Benutzerexit 13
- XBMOUT, globaler Benutzerexit 14
- XDLIPOST, globaler Benutzerexit 32
- XDLIPRE, globaler Benutzerexit 30
  - zum Ändern des PSB für Terminierung 34
- XDSAWT, globaler Benutzerexit 28
- XDSBWT, globaler Benutzerexit 28
- XDTAD, globaler Benutzerexit 22, 23
- XDTLC, globaler Benutzerexit 24
- XDTRD, globaler Benutzerexit 20, 21
- XDUCLSE, globaler Benutzerexit 47
- XDUOUT, globaler Benutzerexit 47
- XDUREQ, globaler Benutzerexit 40
- XDUREQC, globaler Benutzerexit 44
- XEIIN, globaler Benutzerexit 59
- XEIOUT, globaler Benutzerexit 60
- XEISPIN, globaler Benutzerexit 59
- XEISPOUT, globaler Benutzerexit 61
- XEPCAP, globaler Benutzerexit 56
- XFAINTU, globaler Benutzerexit 17
- XFCAREQ, globaler Benutzerexit
  - Beschreibung 94
  - Parameterliste und Rückgabecodes 95
- XFCAREQC, globaler Benutzerexit
  - Beschreibung 94
  - Parameterliste und Rückgabecodes 96
- XFCBFAIL, globaler Benutzerexit 129
- XFCBOUT, globaler Benutzerexit 133
- XFCBOVER, globaler Benutzerexit 134, 135
- XFCFRIN, globaler Benutzerexit
  - Parameterliste und Rückgabecodes 68
- XFCFROUT, globaler Benutzerexit
  - Parameterliste und Rückgabecodes 75
- XFCLDEL, globaler Benutzerexit 136
- XFCNREC, globaler Benutzerexit
  - Beschreibung 121
  - Parameterliste und Rückgabecodes 121
- XFCQUIS, globaler Benutzerexit
  - Beschreibung 126

- XFCREQ, globaler Benutzerexit
  - Befehlsparameterstruktur 82
  - Beispiel für Verwendung 91
  - Beschreibung 81
  - Parameterliste und Rückgabecodes 92
  - UEPCLPS, Parameter 83
- XFCREQC, globaler Benutzerexit
  - Befehlsparameterstruktur 82
  - Beispiel für Verwendung 91
  - Beschreibung 81
  - Parameterliste und Rückgabecodes 93
  - UEPCLPS, Parameter 83
- XFCRLSCO, globaler Benutzerexit
  - Beschreibung 138
- XFCSREQ, globaler Benutzerexit 112
- XFCSREQC, globaler Benutzerexit 116
- XFCVSDS, globaler Benutzerexit
  - Beschreibung 123
- XGMTTEXT, globaler Benutzerexit 140
- XICEREQ, globaler Benutzerexit
  - Befehlsparameterstruktur 165
  - Beispiel für Verwendung 177
  - Parameterliste und Rückgabecodes 160
  - UEPCLPS, Parameter 166
- XICEREQC, globaler Benutzerexit
  - Befehlsparameterstruktur 165
  - Beispiel für Verwendung 177
  - Parameterliste und Rückgabecodes 162
  - UEPCLPS, Parameter 166
- XICERES, globaler Benutzerexit
  - Parameterliste und Rückgabecodes 161
- XICEXP, globaler Benutzerexit 157
- XICREQ, globaler Benutzerexit 155
- XICTENF, globaler Benutzerexit 270
- XISCONA, globaler Benutzerexit 148, 150
- XISLCLQ, globaler Benutzerexit 148, 152
- XISQLCL, globaler Benutzerexit 148, 154
- XISQUE, globaler Benutzerexit 302, 303
  - Exitprogramm entwerfen 307
  - Übersicht 302
  - Verwendung 305
- XLDELETE, globaler Benutzerexit 179
- XLDLOAD, globaler Benutzerexit 178
- XLGSTRM, globaler Benutzerexit 180
- XMEOUT 184
- XMEOUT, globaler Benutzerexit 187
- XMNOUT, globaler Benutzerexit 189
- XNQEREQ, globaler Benutzerexit 49
  - Befehlsparameterstruktur 52
  - UEPCLPS, Parameter 52
- XNQEREQC, globaler Benutzerexit 51
  - Befehlsparameterstruktur 52
  - UEPCLPS, Parameter 52
- XPABND, globaler Benutzerexit 225
- XP CERES, globaler Benutzerexit
  - Beschreibung 207
  - Parameterliste und Rückgabecodes 210
- XPCFTCH, globaler Benutzerexit 218
- XPCHAIR, globaler Benutzerexit 221



XPCREQ, globaler Benutzerexit  
     Befehlsparameterstruktur 212  
     Beispiel für Verwendung 218  
     Beschreibung 206  
     Parameterliste und Rückgabeco-  
     des 208  
     UEPCLPS, Parameter 213  
 XPCREQC, globaler Benutzerexit  
     Befehlsparameterstruktur 212  
     Beispiel für Verwendung 218  
     Beschreibung 208  
     Parameterliste und Rückgabeco-  
     des 211  
     UEPCLPS, Parameter 213  
 XPCTA, globaler Benutzerexit 223  
 XRCINIT, globaler Benutzerexit 289  
 XRCINPT, globaler Benutzerexit 290  
 XRMIOU, globaler Benutzerexit 228  
 XRMML, globaler Benutzerexit 226  
 XRSINDI 230  
 XRSINDI, globaler Benutzerexit 230  
 XSNEX, globaler Benutzerexit 239  
 XSNOFF, globaler Benutzerexit 238  
 XSNON, globaler Benutzerexit 237  
 XSRAB, globaler Benutzerexit 242  
 XSTERM, globaler Benutzerexit 246  
 XSTOUT, globaler Benutzerexit 240  
 XSZARQ, globaler Benutzerexit 62  
     Übersicht 64  
     UEPSZACN, Parameter 65  
 XSZBRQ, globaler Benutzerexit 62  
     Übersicht 63  
     UEPSZACT, Parameter 65  
 XTCATT, globaler Benutzerexit 265  
 XTCIN, globaler Benutzerexit 264  
 XTCOUT, globaler Benutzerexit 265  
 XTDEREQ 278  
 XTDEREQ, globaler Benutzerexit  
     Befehlsparameterstruktur 282  
     Parameterliste und Rückgabeco-  
     des 279  
     UEPCLPS, Parameter 283  
 XTDEREQC 278  
 XTDEREQC, globaler Benutzerexit  
     Befehlsparameterstruktur 282  
     Parameterliste und Rückgabeco-  
     des 280  
     UEPCLPS, Parameter 283  
 XTDIN 275  
 XTDIN, globaler Benutzerexit 276  
 XTDOU 275  
 XTDOU, globaler Benutzerexit 277  
 XTDREQ 275  
 XTDREQ, globaler Benutzerexit 275  
 XTSEREQ, globaler Benutzerexit 253  
     Befehlsparameterstruktur 256  
     Beispielprogramm 262  
     UEPCLPS, Parameter 256  
 XTSEREQC, globaler Benutzerexit 254  
     Befehlsparameterstruktur 256  
     Beispielprogramm 262  
     UEPCLPS, Parameter 256  
 XTSPTIN, globaler Benutzerexit 250  
 XTSPTOUT, Exit 251  
 XTSQRIN, Exit 247  
 XTSQROUT, globaler Benutzerexit 248  
 XWBAUTH, Benutzerexit 140  
 XWBAUTH, globaler Benutzerexit 140  
 XWBOPEN, Benutzerexit 144  
 XWBOPEN, globaler Benutzerexit 140  
 XWBSNDO, Benutzerexit 146  
 XWBSNDO, globaler Benutzerexit 140  
 XWSPRROI, globaler Benutzerexit 194  
 XWSPRROO, globaler Benutzerexit 195  
 XWSPRRWI, globaler Benutzerexit 193  
 XWSPRRWO, globaler Benutzerexit 196  
 XWSRQROI, globaler Benutzerexit 200  
 XWSRQROO, globaler Benutzerexit 199  
 XWSRQRWI, globaler Benutzerexit 201  
 XWSRQRWO, globaler Benutzerexit 198  
 XWSSRROI, globaler Benutzerexit 204  
 XWSSRROO, globaler Benutzerexit 203  
 XWSSRRWI, globaler Benutzerexit 205  
 XWSSRRWO, globaler Benutzerexit 202  
 XXDFA, globaler Benutzerexit 25  
 XXDFB, globaler Benutzerexit 26  
 XXDTO, globaler Benutzerexit 27  
 XXMATT, globaler Benutzerexit 273  
 XXRSTAT, globaler Benutzerexit 309  
 XZCATT, globaler Benutzerexit 291  
 XZCIN, globaler Benutzerexit 292  
 XZCOUT, globaler Benutzerexit 292  
 XZCOUT1, globaler Benutzerexit 293  
 XZIQUE, globaler Benutzerexit 294, 298  
     Exitprogramm entwerfen 302  
     Interaktion mit XISCONA 294  
     Übersicht 294  
     Verwendung 295  
     Verwendung der IRC/ISC-Statistik 302  
     Zeitpunkt des Aufrufs 294

## Z

Zuordnungswarteschlangen  
     Steuerung der Länge  
         mithilfe des globalen Benutzerexits  
         XISCONA 148  
         mithilfe des globalen Benutzerexits  
         XISQUE 302  
         mithilfe des globalen Benutzerexits  
         XZIQUE 294





