

CICS Transaction Server for z/  
OSバージョン 5 リリース 6

ユーザー出口リファレンス



## 注記

本書および本書で紹介する製品をご使用になる前に、[製品の特記事項](#)に記載されている情報をお読みください。

本書は、IBM® CICS® Transaction Server for z/OS®, バージョン 5 リリース 6 (製品番号 5655-Y305655-BTA)、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

### 原典：

CICS Transaction Server for z/OS  
Version 5 Release 5  
User Exit Reference

### 発行：

日本アイ・ビー・エム株式会社

### 担当：

トランスレーション・サービス・センター

© Copyright International Business Machines Corporation 1974, 2020.

# 目次

この PDF について.....	vii
------------------	-----

第 1 章グローバル・ユーザー出口点 (機能別).....	1
-------------------------------	---

グローバル・ユーザー出口点 (アルファベット順).....	1
活動キーポイントプログラム出口 (XAKUSER).....	10
AP ドメイン内のアプリケーション関連データ出口 (XAPADMGR).....	11
基本基本マッピング・サポート出口 (XBMIN、XBMOU).....	12
出口 XBMIN.....	13
出口 XBMOU.....	13
フィールド・エレメント・テーブルの構造.....	14
XBMIN 出口のプログラミング.....	15
XBMOU 出口のプログラミング.....	15
ブリッジ機能出口 XFAINTU.....	16
データ・テーブル管理出口 XDTRD、XDTAD、および XDTLC.....	17
出口 XDTRD.....	18
出口 XDTAD.....	20
出口 XDTLC.....	21
DBCTL インターフェース制御プログラム出口 (XXDFA).....	22
DBCTL トラッキング・プログラム出口 (XXDFB、XXDTO).....	22
出口 XXDFB.....	22
出口 XXDTO.....	23
ディスパッチャー・ドメイン出口 XDSBWT および XDSAWT.....	24
出口 XDSBWT.....	24
出口 XDSAWT.....	24
DL/I インターフェース・プログラム出口 XDLIPRE および XDLIPOST.....	25
出口 XDLIPRE.....	25
出口 XDLIPOST.....	27
グローバル・ユーザー出口ルーチン XDLIPRE の使用例.....	29
ダンプ・ドメイン出口 XDUREQ、XDUREQC、XDUCLSE、および XDUOUT.....	34
出口 XDUREQ.....	34
出口 XDUREQC.....	36
出口 XDUCLSE.....	39
出口 XDUOUT.....	39
エンキュー EXEC インターフェース・プログラム出口 XNQEREQ および XNQEREQC.....	40
出口 XNQEREQ.....	41
出口 XNQEREQC.....	42
コマンド・レベルのパラメーター構造.....	43
イベント・キャプチャー出口 XEPCAP.....	46
EXEC インターフェース・プログラム出口 XEIIN、XEIOUT、XEISPIN、および XEISPOUT.....	47
コマンド・パラメーター・リスト.....	48
コマンドのバイパス.....	48
出口の XEIIN.....	48
出口 XEISPIN.....	49
出口 XEIOUT.....	50
出口 XEISPOUT.....	51
フロントエンド・プログラミング・インターフェース出口 XSZARQ および XSZBRQ.....	51
XSZBRQ.....	52
XSZARQ.....	53
出口固有のパラメーター UEPSZACT および UEPSZACN.....	53
メッセージ出力を制御するための XMEOU の使用.....	55

ファイル制御ドメイン出口 XFCFRIN および XFCFROUT.....	55
出口の XFCFRIN.....	56
出口 XFCFROUT.....	62
ファイル制御 EXEC インターフェース API 出口 XFCREQ および XFCREQC.....	67
コマンド・レベルのパラメーター構造.....	68
コマンド・レベルのパラメーター構造のフィールドの変更.....	72
EID の変更.....	74
パラメーター UEPFSHIP の使用.....	75
EIB (EXEC インターフェース・ブロック).....	75
XFCREQ および XFCREQC の使用方法の例.....	76
出口 XFCREQ.....	76
出口 XFCREQC.....	77
ファイル制御 EXEC インターフェース SPI 出口 XFCAREQ および XFCAREQC.....	78
出口 XFCAREQ.....	79
出口 XFCAREQC.....	80
コマンド・レベルのパラメーター構造.....	81
コマンド・レベルのパラメーター構造のフィールドの変更.....	88
EID の変更.....	92
ユーザー引数の変更.....	92
ファイル制御ファイル状態プログラム出口 XFCSREQ および XFCSREQC.....	93
出口 XFCSREQ.....	94
出口 XFCSREQC.....	97
ファイル制御オープン/クローズ・プログラム出口 XFCNREC.....	101
一致しないバックアウト・リカバリー設定に対する XFCNREC 出口.....	102
BWO の不一致がある場合の XFCNREC の使用.....	102
ファイル制御静止受信出口 XFCVSDS.....	102
出口 XFCVSDS.....	103
ファイル制御静止送信出口 XFCQUIS.....	104
ファイル制御リカバリー・プログラム出口 XFCBFAIL、XFCBOUT、XFCBOVER、および XFCLDEL..	106
呼び出し順序.....	106
出口プログラムの使用可能化.....	107
出口 XFCBFAIL、ファイル制御バックアウト障害出口.....	107
出口 XFCBOUT、ファイル制御バックアウト出口.....	110
出口 XFCBOVER、ファイル制御バックアウト・オーバーライド出口.....	111
出口 XFCLDEL、ファイル制御論理削除出口.....	112
ファイル制御 RLS 共存プログラム出口 XFCRLSCO.....	113
Good morning メッセージ・プログラム出口 (XGMTEXT).....	115
HTTP クライアントのオープンおよび送信出口: XWBAUTH、XWBOPEN、および XWBSNDO.....	115
HTTP クライアントの送信出口 XWBAUTH.....	116
XWBAUTH による LDAP XPI 関数の一般的な使用方法.....	118
HTTP クライアントのオープン出口 XWBOPEN.....	119
HTTP クライアントの送信出口 XWBSNDO.....	120
システム間連絡プログラム出口 XISCONA、XISLCLQ、および XISQLCL.....	121
XISCONA 出口.....	121
XISLCLQ 出口.....	124
XISQLCL 出口.....	125
インターバル制御プログラム出口 XICREQ、XICEXP、および XICTENF.....	126
出口 XICREQ.....	127
出口 XICEXP.....	128
出口 XICTENF.....	128
インターバル制御 EXEC インターフェース・プログラム出口 (XICREQ、XICERES、および XICEREQC).....	128
それぞれの出口に渡されるパラメーター.....	130
出口 XICREQ.....	130
出口 XICERES.....	131
出口 XICEREQC.....	132
コマンド・レベルのパラメーター構造.....	134
ローダー・ドメイン出口 XLDLOAD および XLDELETE.....	146

出口 XLDLOAD.....	146
出口 XLDELETE.....	147
ログ・マネージャー・ドメイン出口 XLGSTRM.....	148
出口 XLGSTRM.....	149
XLGSTRM 出口の使用方法的例.....	150
メッセージ・ドメイン出口 XMEOUT.....	151
出口 XMEOUT.....	152
モニター・ドメイン出口 (XMNOUT).....	154
出口 XMNOUT.....	154
パイプライン・ドメイン出口.....	156
出口 XWSPRRWI.....	157
出口 XWSPRROI.....	158
出口 XWSPRROO.....	159
出口 XWSPRRWO.....	160
出口 XWSRQRWO.....	161
出口 XWSRQROO.....	162
出口 XWSRQROI.....	163
出口 XWSRQRWI.....	164
出口 XWSSRRWO.....	165
出口 XWSSRROO.....	166
出口 XWSSRROI.....	167
出口 XWSSRRWI.....	168
プログラム制御プログラム出口 (XPCREQ、XPCERES、XPCREQC、XPCFTCH、XPCHAIR、XPCTA、 および XPCABND).....	169
プログラム制御出口 XPCREQ、XPCERES、XPCREQC.....	169
出口 XPCFTCH.....	178
出口 XPCHAIR.....	180
出口 XPCTA.....	181
出口 XPCABND.....	183
リソース・マネージャー・インターフェース・プログラム出口 (XRMIIN、XRMIOUT).....	184
出口の XRMIIN.....	184
出口 XRMIOUT.....	185
リソース管理のインストールおよび破棄出口 XRSINDI.....	186
出口 XRSINDI.....	187
サインオン出口およびサインオフ出口 XSNON、XSNOFF、XSNEX.....	192
出口 XSNON.....	192
出口 XSNOFF.....	193
出口 XSNEX.....	194
統計ドメイン出口 XSTOUT.....	195
出口 XSTOUT.....	195
システム・リカバリー・プログラム出口 XSRAB.....	196
システム終了プログラム出口 XSTERM.....	199
一時記憶域ドメイン出口 (XTSQRIN、XTSQROUT、XTSPTIN、XTSPTOUT).....	199
出口 XTSQRIN.....	200
出口 XTSQROUT.....	201
出口 XTSPTIN.....	202
出口 XTSPTOUT.....	203
一時ストレージ EXEC インターフェース・プログラム出口 XTSEREQ および XTSEREQC.....	204
出口 XTSEREQ.....	205
出口 XTSEREQC.....	206
コマンド・レベルのパラメーター構造.....	207
端末割り振りプログラム出口 XALCAID.....	213
端末管理プログラム出口 (XTCIN、XTCOUT、XTCATT).....	214
出口 XTCIN.....	214
出口 XTCOUT.....	215
出口 XTCATT.....	215
「端末未認識」状態出口 XALTENF および XICTENF.....	216
出口.....	216

出口 XALTENF.....	217
出口 XICTENF.....	219
トランザクション・マネージャー・ドメイン出口 XXMATT.....	221
一時データ・プログラム出口 (XTDREQ、XTDIN、XTDOUT).....	222
出口 XTDREQ.....	222
出口 XTDIN.....	223
出口 XTDOUT.....	224
一時データ EXEC インターフェース・プログラム出口 XTDEREQ および XTDEREQC.....	224
出口 XTDEREQ.....	225
出口 XTDEREQC.....	226
コマンド・レベルのパラメーター構造.....	227
ユーザー・ログ・レコード・リカバリー・プログラム出口 XRCINIT および XRCINPT.....	231
出口プログラムのコーディング.....	232
出口プログラムの使用可能化.....	233
出口 XRCINIT.....	233
出口 XRCINPT.....	233
SNA LU 管理プログラム出口 (XZCATT).....	234
SNA 実効ページ・セット・モジュール出口 (XZCIN、XZCOUT、XZCOUT1、および XZIQUE).....	235
出口 XZCIN.....	235
出口 XZCOUT.....	236
出口 XZCOUT1.....	236
MRO および APPC システム間キューの管理用 XZIQUE 出口.....	236
XZIQUE グローバル・ユーザー出口プログラムの設計.....	242
IPIC システム間キュー管理用の XISQUE 出口.....	243
出口 XISQUE.....	243
XISQUE グローバル・ユーザー出口プログラムの使用.....	245
DFHISRDS の統計フィールド.....	246
XISQUE グローバル・ユーザー出口プログラムの設計.....	246
XRF 要求処理プログラム出口 XXRSTAT.....	247
XXRSTAT 出口.....	247
<b>特記事項.....</b>	<b>249</b>
<b>索引.....</b>	<b>255</b>

## この PDF について

---

この PDF は、作成されたグローバル・ユーザー出口プログラムへ CICS が制御を転送できるようにするために用意されているグローバル・ユーザー出口ポイントの解説書です。プログラムでこれらのグローバル・ユーザー出口ポイントを使用する方法を確認するには、*CICS システム・プログラムの開発*という PDF を参照してください。CICS TS V5.4 より前は、この PDF の情報は「カスタマイズ・ガイド」に収録されていました。

本書で使用する用語および表記の詳細については、IBM Knowledge Center の [CICS 資料で使用されている表記規則および用語](#)を参照してください。

### この PDF の作成日

この PDF は、2020 年 5 月 28 日に作成されました。





# 第 1 章 グローバル・ユーザー出口点 (機能別)

出口点が機能関係に応じてグループ分けされています。

グループ分けは通常、出口点が発生する CICS モジュールまたはドメインがベースとされます。ただし、異なるモジュールの出口点と同じ機能を果たす場合、これらの出口は総称名によるグループとされます。出口のグループは、モジュール名または総称名のアルファベット順で示されます。

それぞれのグローバル・ユーザー出口点について、以下の情報が提供されます。

- 出口 ID
- 出口の位置
- 出口に固有の DFHUEPAR パラメーター (存在する場合)
- 有効な戻りコード
- 呼び出し可能な XPI 呼び出し

## グローバル・ユーザー出口点 (アルファベット順)

この表は、各出口の出口名、モジュールまたはドメイン、出口を呼び出す場所とタイミングを示しています。また、追加情報へのリンクも含まれています。

表 1. グローバル・ユーザー出口点のアルファベット順リスト			
出口ルーチン名	モジュールまたはドメイン	呼び出される場面	トピック
XAKUSER	活動キーポイント・プログラム	「end of keypoint」レコードが書き込まれる直前。	<a href="#">10 ページの『活動キーポイントプログラム出口 (XAKUSER)』</a>
XALCAID	端末割り振りプログラム	データを伴う AID がキャンセルされるとき。	<a href="#">213 ページの『端末割り振りプログラム出口 XALCAID』</a>
XALTENF	端末割り振りプログラム	一時データやインターバル制御機能からの ATI 要求で、このシステムで認識されていない端末が要求されるとき。	<a href="#">217 ページの『出口 XALTENF』</a>
XAPADMGR	アプリケーション・ドメイン	継承した関連データ発信元記述子データを持たない非システム・タスクが接続されるとき。	<a href="#">11 ページの『AP ドメイン内のアプリケーション関連データ出口 (XAPADMGR)』</a>
XBMIN	基本マッピング・サポート	入力マッピング操作が正常に完了するとき。	<a href="#">13 ページの『出口 XBMIN』</a>
XBMOUT	基本マッピング・サポート	出力のページが正常に作成されたとき。	<a href="#">13 ページの『出口 XBMOUT』</a>
XDLIPOST	DL/I インターフェース・プログラム	DL/I インターフェース・プログラムからの出口で。	<a href="#">27 ページの『出口 XDLIPOST』</a>
XDLIPRE	DL/I インターフェース・プログラム	DL/I インターフェース・プログラムへの入り口で。	<a href="#">25 ページの『出口 XDLIPRE』</a>

表 1. グローバル・ユーザー出口点のアルファベット順リスト (続き)			
出口ルーチン名	モジュールまたはドメイン	呼び出される場面	トピック
XDSAWT	ディスパッチャー・ドメイン	オペレーティング・システムが待機した後。	<a href="#">24 ページの『出口 XDSAWT』</a>
XDSBWT	ディスパッチャー・ドメイン	オペレーティング・システムが待機する前。	<a href="#">24 ページの『出口 XDSBWT』</a>
XDTAD	データ・テーブル管理	データ・テーブルへの書き込み要求が発行される時。	<a href="#">20 ページの『出口 XDTAD』</a>
XDTLC	データ・テーブル管理	データ・テーブルのロードが完了したとき。	<a href="#">21 ページの『出口 XDTLC』</a>
XDTRD	データ・テーブル管理	データ・テーブルのロード中、ソース・データ・セットからレコードが取り出されているとき。	<a href="#">18 ページの『出口 XDTRD』</a>
XDUCLSE	ダンプ・ドメイン	ドメインがトランザクション・ダンプ・データ・セットを閉じた後。	<a href="#">39 ページの『出口 XDUCLSE』</a>
XDUOUT	ダンプ・ドメイン	ドメインがトランザクション・ダンプ・データ・セットにレコードを書き込む前。	<a href="#">39 ページの『出口 XDUOUT』</a>
XDUREQ	ダンプ・ドメイン	ドメインがシステムまたはトランザクションのダンプを取る前。	<a href="#">34 ページの『出口 XDUREQ』</a>
XDUREQC	ダンプ・ドメイン	システムまたはトランザクションのダンプが取られた後 (あるいは失敗または抑制された後)。	<a href="#">36 ページの『出口 XDUREQC』</a>
XEIIN	EXEC インターフェース・プログラム	EXEC CICS API または SPI コマンドが実行される前。	<a href="#">48 ページの『出口 XEIIN』</a>
XEIOUT	EXEC インターフェース・プログラム	EXEC CICS API または SPI コマンドが実行された後。	<a href="#">50 ページの『出口 XEIOUT』</a>
XEISPIN	EXEC インターフェース・プログラム	EXEC CICS ENABLE、EXEC CICS DISABLE、EXEC CICS EXTRACT EXIT、EXEC CICS PERFORM DUMP、または EXEC CICS RESYNC ENTRYNAM を除く EXEC CICS SPI コマンドが実行される前。	<a href="#">49 ページの『出口 XEISPIN』</a>
XEISPOUT	EXEC インターフェース・プログラム	<b>EXEC CICS ENABLE、EXEC CICS DISABLE、EXEC CICS EXTRACT EXIT、EXEC CICS PERFORM DUMP、または EXEC CICS RESYNC ENTRYNAME</b> を除く EXEC CICS SPI コマンドが実行された後。	<a href="#">51 ページの『出口 XEISPOUT』</a>
XEPCAP	イベント・キャプチャー	イベントが CICS イベント処理によってキャプチャーされる前。	<a href="#">46 ページの『イベント・キャプチャー出口 XEPCAP』</a>
XFAINTU	3270 ブリッジ機能管理プログラム	ブリッジ機能が作成または削除される時。	<a href="#">16 ページの『ブリッジ機能出口 XFAINTU』</a>
XFCAREQ	ファイル制御 EXEC インターフェース・プログラム	CICS がファイル制御 SPI 要求を処理する前。	<a href="#">78 ページの『ファイル制御 EXEC インターフェース SPI 出口 XFCAREQ および XFCAREQC』</a>

表 1. グローバル・ユーザー出口点のアルファベット順リスト (続き)			
出口ルーチン名	モジュールまたはドメイン	呼び出される場面	トピック
XFCAREQC	ファイル制御 EXEC インターフェース・プログラム	ファイル制御 SPI 要求が完了した後。	<a href="#">78 ページの『ファイル制御 EXEC インターフェース SPI 出口 XFCAREQ および XFCAREQC』</a>
XFCBFAIL	ファイル制御リカバリ管理プログラム	UOW のバックアウト中にエラーが発生するとき。	<a href="#">107 ページの『出口 XFCBFAIL、ファイル制御バックアウト障害出口』</a>
XFCBOUT	ファイル制御リカバリ管理プログラム	CICS がファイル更新をバックアウトしようとしているとき。	<a href="#">110 ページの『出口 XFCBOUT、ファイル制御バックアウト出口』</a>
XFCBOVER	ファイル制御リカバリ管理プログラム	バッチ・プログラムが RLS 保持ロック保護をオーバーライドし、バッチ処理用にデータ・セットを開いたために、CICS が UOW のバックアウトをスキップしようとしているとき。	<a href="#">111 ページの『出口 XFCBOVER、ファイル制御バックアウト・オーバーライド出口』</a>
XFCFRIN	ファイル制御ドメイン	ファイル制御要求が実行される前。	<a href="#">56 ページの『出口の XFCFRIN』</a>
XFCFROUT	ファイル制御ドメイン	ファイル制御要求が実行された後。	<a href="#">62 ページの『出口 XFCFROUT』</a>
XFCLDEL	ファイル制御リカバリ管理プログラム	VSAM ESDS または BDAM データ・セットへの書き込みがバックアウトされるとき。	<a href="#">112 ページの『出口 XFCLDEL、ファイル制御論理削除出口』</a>
XFCNREC	ファイル制御オープン/クローズ・プログラム	ファイル・オープン処理の途中でファイルのバックアウト・リカバリ設定とそれに関連付けられているデータ・セットの間に不一致が検出されたとき。	<a href="#">101 ページの『ファイル制御オープン/クローズ・プログラム出口 XFCNREC』</a>
XFCQUIS	ファイル制御静止送信プログラム	SET DSNAME QUIESCESTATE コマンドの完了時 (成功か失敗かを問わず)。	<a href="#">104 ページの『ファイル制御静止送信出口 XFCQUIS』</a>
XFCREQ	ファイル制御 EXEC インターフェース・プログラム	CICS がファイル制御 API 要求を処理する前。	<a href="#">76 ページの『出口 XFCREQ』</a>
XFCREQC	ファイル制御 EXEC インターフェース・プログラム	ファイル制御 API 要求が完了した後。	<a href="#">77 ページの『出口 XFCREQC』</a>

表 1. グローバル・ユーザー出口点のアルファベット順リスト (続き)			
出口ルーチン名	モジュールまたはドメイン	呼び出される場面	トピック
XFCRLSCO	ファイル制御 RLS 共存プログラム	他の方法では RLS 共存障害によって失敗する、VSAM RLS ファイルまたは非 RLS の読み取り専用ファイルを開くとき。	<a href="#">113 ページの『ファイル制御 RLS 共存プログラム 出口 XFCRLSCO』</a>
XFCSREQ	ファイル制御ファイル状態プログラム	ファイルの OPEN、CLOSE、ENABLE、または DISABLE コマンドが試行される前。	<a href="#">93 ページの『ファイル制御ファイル状態プログラム 出口 XFCSREQ および XFCSREQC』</a>
XFCSREQC	ファイル制御ファイル状態プログラム	ファイルの OPEN、CLOSE、CANCEL CLOSE、ENABLE、または DISABLE コマンドが完了した後。	<a href="#">93 ページの『ファイル制御ファイル状態プログラム 出口 XFCSREQ および XFCSREQC』</a>
XFCVSDS	ファイル制御静止受信プログラム	シスプレックス内で発生しているデータ・セット関連アクションの結果として処理が必要であることを RLS が CICS に通知した後。	<a href="#">102 ページの『ファイル制御静止受信 出口 XFCVSDS』</a>
XGMTTEXT	「Good morning」メッセージ・プログラム	"Good morning" メッセージが送信される前。	<a href="#">115 ページの『Good morning メッセージ・プログラム 出口 (XGMTTEXT)』</a>
XICEREQ	インターバル制御 EXEC インターフェース・プログラム	CICS がインターバル制御 API 要求を処理する前。	<a href="#">130 ページの『出口 XICEREQ』</a>
XICEREQC	インターバル制御 EXEC インターフェース・プログラム	インターバル制御 API 要求が完了した後。	<a href="#">132 ページの『出口 XICEREQC』</a>
XICERES	インターバル制御 EXEC インターフェース・プログラム	ルーティング領域が「リソース使用不可」(RESUNAVAIL) 状態をサポートしている場合に、この領域に動的にルーティングされている非端末関連 EXEC CICS START 要求を CICS が処理する前。	<a href="#">131 ページの『出口 XICERES』</a>
XICEXP	インターバル制御プログラム	インターバル制御の時間間隔が満了した後。	<a href="#">128 ページの『出口 XICEXP』</a>
XICREQ	インターバル制御プログラム	インターバル制御プログラム開始時の要求分析の前。	<a href="#">127 ページの『出口 XICREQ』</a>
XICTENF	インターバル制御プログラム	EXEC CICS START コマンドで、このシステムで認識されていない端末が要求されるとき。	<a href="#">219 ページの『出口 XICTENF』</a>

表 1. グローバル・ユーザー出口点のアルファベット順リスト (続き)			
出口ルーチン名	モジュールまたはドメイン	呼び出される場面	トピック
XISCONA	システム間連絡プログラム	すぐに使用可能なリモート領域へのセッションがないために、機能シブまたは DPL 要求がキューに入れられようとしているとき。	<a href="#">121 ページの『システム間連絡プログラム出口 XISCONA、XISLCLQ、および XISQLCL』</a>
XISLCLQ	システム間連絡プログラム	リモート・システムがサービス中でない、リモート・システムへの接続が確立できない、またはすぐに使用できるセッションがなく、XISCONA 出口プログラムが発行側の領域でその要求をキューに入れないことを指定していたために、機能シブされた START NOCHECK 要求へのセッションの割り振りの試行が失敗した後。	<a href="#">124 ページの『XISLCLQ 出口』</a>
XISQLCL	システム間連絡プログラム	IPIC 接続がサービス休止の状態であるか、IPIC 接続が獲得されていないか、セッションが使用不可で CICS が新規セッションの要求をキューに入れないために、IPIC 接続のためにスケジュールに入れている、START NOCHECK 要求に対するセッションの割り振りの試行が失敗した後。	<a href="#">125 ページの『XISQLCL 出口』</a>
XISQUE	IPCONN 上でのセッションのためにキューに入れる要求の数を制御するため	次のような場合: 1. IPCONN 上でのセッションに対する割り振り要求がキューに入れられようとしている 2. 直前のキューイング抑止の後に IP 割り振り要求が成功する	<a href="#">243 ページの『IPIC システム間キュー管理用の XISQUE 出口』</a>
XLDELETE	ローダー・ドメイン	プログラムのインスタンスが CICS によって解放された後、プログラムが記憶域から解放される直前。	<a href="#">147 ページの『出口 XLDELETE』</a>
XLDLOAD	ローダー・ドメイン	プログラムのインスタンスが記憶域に入れられた後、プログラムが使用可能にされる前。	<a href="#">146 ページの『出口 XLDLOAD』</a>
XLGSTRM	ログ・マネージャー・ドメイン	ログ・ストリームが存在しないことを CICS ログ・マネージャーが検出した後、ログ・ストリームを定義するために MVS™ システム・ロガーを呼び出す前。	<a href="#">148 ページの『ログ・マネージャー・ドメイン出口 XLGSTRM』</a>
XMEOUT	メッセージ・ドメイン	メッセージ・ドメインからその宛先にメッセージが送信される前。	<a href="#">152 ページの『出口 XMEOUT』</a>
XMNOUT	モニター・ドメイン	レコードが SMF に書き込まれる前、または SMF への書き込みの前にバッファーに入れられる前。	<a href="#">154 ページの『出口 XMNOUT』</a>
XNQEREQ	エンキュー EXEC インターフェース・プログラム	CICS がエンキュー API 要求を処理する前。	<a href="#">41 ページの『出口 XNQEREQ』</a>
XNQEREQC	エンキュー EXEC インターフェース・プログラム	エンキュー API 要求が完了した後。	<a href="#">42 ページの『出口 XNQEREQC』</a>
XPCABND	プログラム管理プログラム	トランザクションが異常終了した後、ダンプ呼び出しが行われる前。	<a href="#">183 ページの『出口 XPCABND』</a>

表 1. グローバル・ユーザー出口点のアルファベット順リスト (続き)			
出口ルーチン名	モジュールまたはドメイン	呼び出される場面	トピック
XPCERES	プログラム管理プログラム	ルーティング領域が「リソース使用不可」(RESUNAVAIL) 状態をサポートしている場合に、この領域に動的にルーティングされているプログラム・リンクまたは Link3270 ブリッジ要求を CICS が処理する前。	<a href="#">171 ページの『出口 XPCERES』</a>
XPCFTCH	プログラム管理プログラム	アプリケーション・プログラムに制御が渡される前。	<a href="#">178 ページの『出口 XPCFTCH』</a>
XPCHAIR	プログラム管理プログラム	HANDLE ABEND ルーチンに制御が渡される前。	<a href="#">180 ページの『出口 XPCHAIR』</a>
XPCREQ	プログラム管理プログラム	LINK 要求が処理される前。	<a href="#">170 ページの『出口の XPCREQ』</a>
XPCREQC	プログラム管理プログラム	LINK 要求が完了した後。	<a href="#">172 ページの『出口 XPCREQC』</a>
XPCTA	プログラム管理プログラム	異常終了が発生した後、環境が変更される前。	<a href="#">181 ページの『出口 XPCTA』</a>
XRCINIT	ユーザー・ログ・レコード・リカバリー・プログラム	ウォーム・リスタートおよび緊急時再始動中に、CICS システム・ログにユーザー・リカバリー・ログが検出されたとき。	<a href="#">233 ページの『出口 XRCINIT』</a>
XRCINPT	ユーザー・ログ・レコード・リカバリー・プログラム	ウォーム・リスタートおよび緊急時再始動中に、それぞれについて CICS システム・ログにユーザー・リカバリー・ログ・レコードが検出されたとき。	<a href="#">233 ページの『出口 XRCINPT』</a>
XRMIIN	リソース・マネージャー・インターフェース・プログラム	EXEC DLI、EXEC SQL、または RMI コマンドが実行される前。	<a href="#">184 ページの『出口の XRMIIN』</a>
XRMIOUT	リソース・マネージャー・インターフェース・プログラム	EXEC DLI、EXEC SQL、または RMI コマンドが実行された後。	<a href="#">184 ページの『出口の XRMIIN』</a>
XRSINDI	リソース管理モジュール	リソースを正常にインストールまたは破棄した直後。	<a href="#">186 ページの『リソース管理のインストールおよび破棄出口 XRSINDI』</a>
XSNEX	セキュリティ・マネージャー・ドメイン	以前の CICS のサインオンおよびサインオフの動作 (CICS TS 2.1 より前) を復元する	<a href="#">194 ページの『出口 XSNEX』</a>
XSNOFF	セキュリティ・マネージャー・ドメイン	端末ユーザーのサインオフの後。	<a href="#">193 ページの『出口 XSNOFF』</a>
XSNON	セキュリティ・マネージャー・ドメイン	端末ユーザーのサインオンの後。	<a href="#">192 ページの『出口 XSNON』</a>
XSRAB	システム・リカバリー・プログラム	システム・リカバリー・プログラムが SRT 内に MVS 異常終了コードの一致を検出するとき。	<a href="#">196 ページの『システム・リカバリー・プログラム出口 XSRAB』</a>



表 1. グローバル・ユーザー出口点のアルファベット順リスト (続き)			
出口ルーチン名	モジュールまたはドメイン	呼び出される場面	トピック
XSTERM	システム終了プログラム	通常のシステム・シャットダウンにおいて、TD バッファがクリアされる直前。	<a href="#">199 ページの『システム終了プログラム出口 XSTERM』</a>
XSTOUT	統計ドメイン	統計レコードが SMF に書き込まれる前。	<a href="#">195 ページの『出口 XSTOUT』</a>
XSZARQ	フロントエンド・プログラミング・インターフェース	FEPI 要求が完了した後。	<a href="#">51 ページの『フロントエンド・プログラミング・インターフェース出口 XSZARQ および XSZBRQ』</a>
XSZBRQ	フロントエンド・プログラミング・インターフェース	FEPI 要求のアクションが実行される前。	<a href="#">51 ページの『フロントエンド・プログラミング・インターフェース出口 XSZARQ および XSZBRQ』</a>
XTCATT	端末管理プログラム	タスク接続の前。	<a href="#">215 ページの『出口 XTCATT』</a>
XTCIN	端末管理プログラム	入力イベントの後。	<a href="#">214 ページの『出口 XTCIN』</a>
XTCOUT	端末管理プログラム	出力イベントの前。	<a href="#">215 ページの『出口 XTCOUT』</a>
XTDEREQ	一時データ EXEC インターフェース・プログラム	CICS が一時データ API 要求を処理する前。	<a href="#">225 ページの『出口 XTDEREQ』</a>
XTDEREQC	一時データ EXEC インターフェース・プログラム	一時データ API 要求が完了した後。	<a href="#">226 ページの『出口 XTDEREQC』</a>
XTDIN	一時データ・プログラム	QSAM (区画外) または VSAM (区画内) からのデータの受信後。	<a href="#">223 ページの『出口 XTDIN』</a>
XTDOUT	一時データ・プログラム	QSAM (区画外) または VSAM (区画内) ユーザー定義一時データ・キューにデータを渡す前。	<a href="#">224 ページの『出口 XTDOUT』</a>
XTDREQ	一時データ・プログラム	要求分析の前。	<a href="#">222 ページの『出口 XTDREQ』</a>
XTSREQ	一時記憶域 EXEC インターフェース・プログラム	CICS が一時記憶域 API 要求を処理する前。	<a href="#">205 ページの『出口 XTSREQ』</a>
XTSREQC	一時記憶域 EXEC インターフェース・プログラム	一時記憶域 API 要求が完了した後。	<a href="#">206 ページの『出口 XTSREQC』</a>
XTSPTIN	一時記憶域ドメイン	TSPT 機能の呼び出しの前。	<a href="#">202 ページの『出口 XTSPTIN』</a>

表 1. グローバル・ユーザー出口点のアルファベット順リスト (続き)			
出口ルーチン名	モジュールまたはドメイン	呼び出される場面	トピック
XTSPTOUT	一時記憶域ドメイン	TSPT 機能の呼び出しの後。	<a href="#">203 ページの『出口 XTSPTOUT』</a>
XTSQRIN	一時記憶域ドメイン	TSQR 機能の呼び出しの前。	<a href="#">200 ページの『出口 XTSQRIN』</a>
XTSQROUT	一時記憶域ドメイン	TSQR 機能の呼び出しの後。	<a href="#">201 ページの『出口 XTSQROUT』</a>
XWBAUTH	Web ドメイン	<b>EXEC CICS WEB SEND</b> コマンドまたは <b>EXEC CICS WEB CONVERSE</b> コマンドの処理中。	<a href="#">115 ページの『HTTP クライアントのオープンおよび送信出口: XWBAUTH、XWBOPEN、および XWBSNDO』</a>
XWBOPEN	Web ドメイン	<b>EXEC CICS WEB OPEN</b> コマンドまたは <b>EXEC CICS INVOKE SERVICE</b> コマンドの処理中。	<a href="#">115 ページの『HTTP クライアントのオープンおよび送信出口: XWBAUTH、XWBOPEN、および XWBSNDO』</a>
XWBSNDO	Web ドメイン	<b>EXEC CICS WEB SEND</b> コマンドまたは <b>EXEC CICS WEB CONVERSE</b> コマンドの処理中。	<a href="#">115 ページの『HTTP クライアントのオープンおよび送信出口: XWBAUTH、XWBOPEN、および XWBSNDO』</a>
XWSPRROI	パイプライン・ドメイン	XWSPRRWI 出口のインスタンスが呼び出された後、かつ Web サービス・プロバイダー・ビジネス・アプリケーションの前。	<a href="#">158 ページの『出口 XWSPRROI』</a>
XWSPRROO	パイプライン・ドメイン	Web サービス・プロバイダー・アプリケーションが戻った後、かつ CICS が応答メッセージの本体を作成する前。	<a href="#">159 ページの『出口 XWSPRROO』</a>
XWSPRRWI	パイプライン・ドメイン	CICS が Web サービス要求本体を言語構造に変換した後、かつ XWSPRROI 出口のインスタンスが呼び出される前。	<a href="#">157 ページの『出口 XWSPRRWI』</a>
XWSPRRWO	パイプライン・ドメイン	XWSPRROO 出口のインスタンスの後、かつ CICS が応答メッセージの本体を作成する前。	<a href="#">160 ページの『出口 XWSPRRWO』</a>
XWSRQROI	パイプライン・ドメイン	CICS がアウトバウンド Web サービス応答を処理した後、かつ XWSRQRWI 出口のインスタンスの前。	<a href="#">163 ページの『出口 XWSRQROI』</a>
XWSRQROO	パイプライン・ドメイン	XWSRQRWO 出口のインスタンスが処理された後、かつ Web サービス・トランスポートのデータがアウトバウンドに流れる前。	<a href="#">162 ページの『出口 XWSRQROO』</a>



表 1. グローバル・ユーザー出口点のアルファベット順リスト (続き)			
出口ルーチン名	モジュールまたはドメイン	呼び出される場面	トピック
XWSRQRWI	パイプライン・ドメイン	CICS がアウトバウンド Web サービス応答を処理した後、かつ XWSRQROI 出口のインスタンスの後。	<a href="#">164 ページの『出口 XWSRQRWI』</a>
XWSRQRWO	パイプライン・ドメイン	CICS がアプリケーションの言語構造を Web サービス要求の本体に変換した後、かつ CICS がオプションの XWSRQROO 出口点を処理する前。	<a href="#">161 ページの『出口 XWSRQRWO』</a>
XWSSRROI	パイプライン・ドメイン	CICS がアウトバウンド Web サービス応答を処理した後、かつ XWSSRRWI 出口のインスタンスの前。	<a href="#">167 ページの『出口 XWSSRROI』</a>
XWSSRROO	パイプライン・ドメイン	XWSSRRWO 出口のインスタンスが処理された後、かつ Web サービス・トランスポートのアウトバウンドに流れるデータを暗号化する前。	<a href="#">166 ページの『出口 XWSSRROO』</a>
XWSSRRWI	パイプライン・ドメイン	CICS がアウトバウンド Web サービス応答を処理した後、かつ XWSSRROI 出口のインスタンスの後。	<a href="#">168 ページの『出口 XWSSRRWI』</a>
XWSSRRWO	パイプライン・ドメイン	CICS がアプリケーションの言語構造を Web サービス要求の本体に変換した後、かつ CICS がオプションの XWSSRRWO 出口点を処理する前、また、パイプラインのセキュリティ・ハンドラーによって暗号化される前。	<a href="#">165 ページの『出口 XWSSRRWO』</a>
XXDFA	DBCTL インターフェース制御プログラム	アクティブ CICS で、CICS-DBCTL 接続が失敗するとき。	<a href="#">22 ページの『DBCTL インターフェース制御プログラム出口 (XXDFA)』</a>
XXDFB	DBCTL トラッキング・プログラム	代替 CICS で、DBCTL が失敗するとき。	<a href="#">22 ページの『出口 XXDFB』</a>
XXDTO	DBCTL トラッキング・プログラム	代替 CICS で、アクティブ DBCTL が失敗するとき。	<a href="#">23 ページの『出口 XXDTO』</a>
XXMATT	トランザクション・マネージャー・ドメイン	ユーザー・トランザクションが接続されるとき。	<a href="#">221 ページの『トランザクション・マネージャー・ドメイン出口 XXMATT』</a>
XXRSTAT	XRF 要求処理プログラム	z/OS Communications Server の障害または略奪的なテークオーバーの発生後。	<a href="#">247 ページの『XXRSTAT 出口』</a>
XZCATT	z/OS Communications Server 端末管理プログラム	タスク接続の前。	<a href="#">234 ページの『SNA LU 管理プログラム出口 (XZCATT)』</a>
XZCIN	z/OS Communications Server 作業セット・モジュール	入力イベントの後。	<a href="#">235 ページの『出口 XZCIN』</a>

表 1. グローバル・ユーザー出口点のアルファベット順リスト (続き)			
出口ルーチン名	モジュールまたはドメイン	呼び出される場面	トピック
XZCOUT	z/OS Communications Server 作業セット・モジュール	出力イベントの前。	236 ページの『 <a href="#">出口 XZCOUT</a> 』
XZCOUT1	z/OS Communications Server 作業セット・モジュール	メッセージが RU に割り込む前。	236 ページの『 <a href="#">出口 XZCOUT1</a> 』
XZIQUE	z/OS Communications Server 作業セット・モジュール	<ol style="list-style-type: none"> <li>セッションに対する割り振り要求がキューに入れられようとしているとき。</li> <li>直前のキューイング抑止の後に割り振り要求が成功したとき。</li> </ol>	236 ページの『 <a href="#">MRO および APPC システム間キューの管理用 XZIQUE 出口</a> 』

## 活動キーポイントプログラム出口 (XAKUSER)

XAKUSER 出口は、活動キーポイント処理中に呼び出されます。この出口を使用して、緊急時再始動後に復元する必要のあるユーザー・データをシステム・ログに記録することができます。

最良のパフォーマンスを得るためには、ジャーナル制御要求に WAIT を指定しないでください。CICS は、出口プログラムから戻るときに、キーポイント・レコードの同期終了を書き込むことによって記録を強制します。

NOEDF オプションを使用して、出口プログラムを変換する必要があります。この出口プログラムのリンク先のすべてのプログラムも、このオプションを使用して変換する必要があります。PL/I で作成されたプログラムにリンクすることはできません。

出口プログラムがすべてのキーポイントで確実に呼び出されるようにするため、第 1 フェーズの PLTPI プログラムを使用して出口プログラムを使用可能にする必要があります。[初期設定プログラムの作成](#)を参照してください。ただし、プログラムをこのステージで使用可能にする場合、プログラムが COBOL または C でコーディングされたプログラムへのリンクを試行しないようにしてください。所定の初期設定が行われる前に呼び出されてしまう可能性があるからです。

**注:** 出口プログラムは、重要な CICS システム活動の一部を形成します。出口プログラムが失敗すると、CICS は終了します。XAKUSER 出口で利用できるのは、リストされた EXEC CICS コマンドのみです。出口がリンクするのは、同じ制約を持つ他のプログラムのみでなければなりません。

### 出口 XAKUSER

#### 呼び出される状況

活動キーポイント処理中。

#### 出口固有のパラメーター

##### UEPAKTYP

出口が呼び出されるキーポイントのタイプを示す 1 バイト・フィールドのアドレス。可能な値は次のとおりです。

##### UEPAKPER

活動キーポイント

##### UEPAKWSD

ウォーム・シャットダウン・キーポイント

#### 戻りコード

##### UERCNORM

処理が続行される。

### **XPI 呼び出し**

XPI は使用しないでください。

### **API および SPI 呼び出し**

以下のコマンドがサポートされています。

- ADDRESS CWA
- ADDRESS EIB
- LINK (ただし、ローカル・プログラムへのリンクのみ。分散プログラム・リンクは使用できません。)
- RETURN
- WRITE JOURNALNAME

## **AP ドメイン内のアプリケーション関連データ出口 (XAPADMGR)**

XAPADMGR 出口は、分散トランザクションに使用します。XAPADMGR を使用すると、分散トランザクションの起点で、ユーザー情報をタスクの関連データに追加することができます。この情報は後で、例えば、CICSplex SM を介して実行される処理の検索キーとして使用できます。

使用可能になっている場合、この出口プログラムは、入力発信元記述子レコードが提供されていない非システム・タスクに接続したときに呼び出されます。

出口プログラムは、入力時にタスクの関連データを受け渡されます。CICS コマンドを使用することにより、この出口が他のソースから他の関連情報を検出し、関連データに含めるという可能性もあります。

**注:** IPIC による接続を介して DPL を使用する分散トランザクションは、そのトランザクション・グループ ID と発信元データ (ユーザー関係子を含む) を渡して、ターゲット領域のミラー・タスクにより継承されます。

出口プログラムは、関連データ内で見つかった情報のロギングなどの他のアクティビティーを、ワークロードの監査やアカウントिंगの目的で実行できます。関連データと発信元データについて詳しくは、関連データを参照してください。

### **出口 XAPADMGR**

#### **呼び出し時**

継承する関連データが渡されていない、非システム・タスクに接続したとき。

#### **出口固有のパラメーター**

##### **UEPADCB**

選択可能な関連データ制御ブロックのアドレス。このアドレスは、DFHMNADS DSECT によってマップされます。

##### **UEPADCBL**

関連データ制御ブロックの長さ (バイト単位)。

##### **UEPUCD**

出口プログラムがユーザー相関データを配置できる 64 バイトの出力域のアドレス。

この領域は、XAPADMGR ユーザー出口点でアクティブな最初の出口プログラムを呼び出す前に、CICS によってゼロにクリアされますが、複数の出口プログラムがアクティブである場合は、プログラム間では CICS によってリセットされません。同じストレージ域が、同じ出口点内の後続のアクティブな出口プログラムに渡されます。

#### **戻りコード**

##### **UERCNORM**

処理が続行される。

### **XPI 呼び出し**

すべて使用できます。

### **API および SPI 呼び出し**

以下以外は、すべて使用できます。

- EXEC CICS ABEND

- EXEC CICS PERFORM SHUTDOWN

## サンプル出口プログラム

DFH\$APAD

## 基本基本マッピング・サポート出口 (XBMIN、XBMOUT)

XBMOUT および XBMIN の 2 つの基本マッピング・サポート出口が提供されています。XBMIN 出口を使用すると、BMS が要求を正常に処理した後に RECEIVE MAP 要求をインターセプトできます。XBMOUT 出口を使用すると、BMS が要求を正常に処理した後、あるいは、累積マッピングが進行中の場合は各ページの出力の完了後に、SEND MAP 要求をインターセプトできます。

XBMIN 出口は、有効になっていれば、以下のすべての条件が満たされた場合に呼び出されます。

- RECEIVE MAP コマンドが正常に処理された。
- コマンドで参照されているマップに、VALIDN=USEREXIT として指定されたフィールドが少なくとも 1 つ含まれている。
- インバウンド・データ・ストリームで少なくとも 1 つの USEREXIT フィールドが返され、アプリケーション・データ構造にマップされた。

XBMIN を使用して、以下のことができます。

- この要求でアプリケーションにマップされた VALIDN=USEREXIT として定義されている各フィールドを分析する。
- マップに定義されたマップ・セット名、マップ名、およびフィールド長と、インバウンド・データ・ストリームで返された実際のフィールド・データの長さを使用する。
- 各フィールド内のデータを変更する。

XBMOUT 出口は、有効になっていれば、以下のすべての条件が満たされた場合に呼び出されます。

- SEND MAP コマンドが正常に処理された。
- コマンドで参照されているマップに、VALIDN=USEREXIT として指定されたフィールドが少なくとも 1 つ含まれている。
- アウトバウンド・データ・ストリームで少なくとも 1 つの USEREXIT フィールドが生成された。

XBMOUT を使用して、以下のことができます。

- アウトバウンド・データ・ストリームで生成された VALIDN=USEREXIT として定義されている各フィールドを分析する。
- マップに定義されたマップ・セット名、マップ名、およびフィールド長と、アウトバウンド・データ・ストリームに入れられた実際のフィールド・データの長さを使用する。
- 各フィールド内のデータを変更する。
- 各フィールドと共に送信される属性を変更する。

両方の出口に、以下の 4 つの出口固有パラメーターが渡されます。

1. マッピング要求に関連付けられている TCTTE のアドレス
2. マッピング要求を発行したタスクに関連付けられているシステム EIB のアドレス。
3. フィールド・エレメント・テーブル内のエレメント数を示すハーフワード・バイナリー・カウントアドレス。
4. フィールド・エレメント・テーブルのアドレス。

## サンプル・プログラム DFH\$BMXT

CICS では、サンプル・プログラム DFH\$BMXT を提供します。このサンプル・プログラムは、マップされた入出力データを「フィールド・エレメント」テーブルで提供される情報を参照して変更する方法を示しています。また、コピーブック DFHXBMD5 も提供されます。このコピーブックは、フィールド・エレメントの構造を定義する DSECT です。

## 出口 XBMIN

この出口は、基本マッピング・サポート (BMS) が入力マッピング操作を正常に処理した後で呼び出されます。

### 呼び出される状況

BMS が入力マッピング操作を正常に処理した後。

### 出口固有のパラメーター

#### UEPBMCTCT

マッピング要求に関連付けられた TCTTE のアドレス。

#### UEPEXECB

タスクに関連付けられたシステム EIB のアドレス。

#### UEPBMCNT

フィールド・エレメント・テーブル内の「フィールド・エレメント」のハーフワードの 2 進数のアドレス。

#### UEPBMTAB

フィールド・エレメント・テーブルのアドレス。

### 戻りコード

#### UERCNORM

処理が続行される。

#### UERCPURG

タスクは XPI 呼び出し中にページされます。

### XPI 呼び出し

すべて使用できます。

## 出口 XBMOU

この出口は、基本マッピング・サポート (BMS) が出力マッピング操作時に 1 ページの出力を正常に完了した後で呼び出されます。

### 呼び出される状況

BMS が出力マッピング操作時に 1 ページの出力を正常に完了した後。

### 出口固有のパラメーター

#### UEPBMCTCT

マッピング要求に関連付けられた TCTTE のアドレス。

#### UEPEXECB

タスクに関連付けられたシステム EIB のアドレス。

#### UEPBMCNT

フィールド・エレメント・テーブル内の「フィールド・エレメント」のハーフワードの 2 進数のアドレス。

#### UEPBMTAB

フィールド・エレメント・テーブルのアドレス。

### 戻りコード

#### UERCNORM

処理が続行される。

#### UERCPURG

タスクは XPI 呼び出し中にページされます。

### XPI 呼び出し

すべて使用できます。

## フィールド・エレメント・テーブルの構造

フィールド・エレメント・テーブルには、出口に渡される個々の「対象フィールド」に関する情報を提供する1つ以上のエレメントが含まれています。

「対象フィールド」とは、マッピング操作で参照されるマップ・セットを作成するために使用されるマップ・ソース・ファイルで、VALIDN=USEREXITとして定義されたフィールドです。

各フィールド・エレメントは以下の構造を備えています。

### BMXMAPST

このフィールドに関連付けられたマップ・セットの名前が含まれた8バイトの領域。CICS インストール済み環境のマップ・セット名で端末接尾部または代替接尾部が使用される場合、マップ・セット名は、マッピング要求で指定された名前に接尾部が付加されている場合があります。

### BMXMAP

このフィールドに関連付けられたマップの名前が含まれた7バイトの領域。

### BMXFDFB

マップ・ロード・モジュールのフィールド指定からコピーされる1バイトのフィールド。ここには、以下の標識が含まれます。

**X'80'**

CASE=MIXED

**X'40'**

グループ・フィールド・エントリー

**X'20'**

グループ・フィールド記述子

**X'10'**

ATTRB=DET

**X'08'**

JUSTIFY=ZERO

**X'04'**

JUSTIFY=RIGHT

**X'02'**

INITIAL、XINIT、またはGINITを指定

**X'01'**

名前フィールド (DSECT エントリーが存在)

### BMXMAPLN

DFHMDL マクロのLENGTH オプションで定義されているフィールド長が含まれたハーフワード・バイナリー値。

### BMXACTLN

このフィールドで受信または送信されるデータの実際の長さが含まれたハーフワード・バイナリー値。

### BMXDATA

フィールド・データのアドレス。

XBMIN 出口では、BMXDATA は、BMS が入力マッピングのために取得した作業域を指します。出口が制御を返すとき、この作業域はこのマップに関連付けられたアプリケーション・データ構造にコピーされます。

XBMOU 出口では、BMXDATA は、BMS が生成した出力データ・ストリームを入れた端末入出力領域を指します。出口が制御を返すとき、TIOA は、SEND MAP 要求で指定されている TERMINAL (デフォルト)、SET、または PAGING のいずれかのオプションの処理に従って破棄されます。

### BMXATTR

XBMOU 出口でのみ使用できます。これは、BMS がこのフィールドに先行する出力データ・ストリームに配置した属性がある場合に、その属性のアドレスです。

### BMXMAPOF

マップ内のフィールドのオフセット。例えば、マップが以下のように定義されているとします。

```
MYMAP DFHMDI SIZE=(12,40)
```

そして、このマップ内のフィールドが以下のように定義されているとします。

```
FLDA DFHMDF POS=(5,1)
```

このフィールドのオフセット (ゼロに対する相対位置) は、10 進表記で 160 です。この例では、BMXMAPOF に値 X'00A0' が含まれています。

#### BMXBUF

装置バッファ内のフィールドのオフセットです。一般に (つまり、マップの寸法が装置で使用されている現在の画面サイズと同じ場合)、この値は BMXMAPOF の場合と同じになります。ただし、BMXMAPOF の説明で示した例を使用すると、MYMAP が現在 24 x 80 の画面際を使用する装置に送られた場合、装置バッファ内のフィールドのオフセット (ここでもゼロに対する相対位置) は 10 進数表記で 320 になります。この例では、BMXBUF に含まれる値は X'0140' になります。

## XBMIN 出口のプログラミング

XBMIN 出口をプログラミングするときは、データ長を考慮することが重要です。

実際のデータ長 (BMXACTLN 内) が、マップで定義された長さ (BMXMAPLN 内) より短いことがあります。これが起こるのは、例えば、端末オペレーターがデータ・エン트리・フィールドを完全に埋めなかった場合です。その場合、BMS はフィールド内のデータを右寄せまたは左寄せにし、フィールドにブランクまたはゼロの文字を埋め込みます。この位置調整と埋め込みは、データベースが呼び出される前に行われます。出口プログラムでは、BMXFDFB フィールドのビット設定を検査することによって、BMS がフィールドの位置調整と埋め込みをどのように実行したかを判別することができます。

実際のデータ長 (BMXACTLN 内) が、マップで定義された長さ (BMXMAPLN 内) より長いことがあります。これが起こるのは、例えば、マップに無保護フィールドが含まれていて、その直後に続く別のフィールドがない場合です。この場合、端末オペレーターはフィールドの終わりを超えてデータを入力することができます。これが起こると、マップ内のフィールドに対して定義された長さに従って、データ・フィールドが BMS によって切り捨てられます。ただし、BMXACTLN には、インバウンド・データ・ストリーム内で検出されたデータの長さが含まれています。

XBMIN 出口でデータを変更する場合、最も安全な方法は BMXMAPLN で提供される長さを使用することです。ただし、BMS によって追加された埋め込み文字がすべて保持されるようにします。

BMXATTR は、XBMIN 出口では無視する必要があります。これには常に 2 進ゼロが含まれています。

## XBMOU 出口のプログラミング

XBMOU 出口をプログラミングするときは、実際のデータ長を考慮することが重要です。

実際のデータ長 (BMXACTLN 内) は、マップで定義された長さ (BMXMAPLN 内) より短いことがあります。これは、各出力フィールドに対して BMS が末尾のヌルを圧縮するために起こる現象です。

実際のデータ長を出口プログラムで変更することはできません。この出口は、出力データ・ストリームが生成された後で呼び出されます。そのため、データ長を変更しようとすると、無効なデータ・ストリームが生じることがあります。したがって、XBMOU 出口プログラムでデータを変更する場合は、BMXACTLN 内の長さ値を参照して変更する必要があります。

BMXDATA にはヌル値が含まれていることがあります。これは、SEND MAP 要求に MAPONLY オプションが指定されていて、マップにデフォルト・データを持たないフィールドがあった場合に発生することがあります。これが原因で、BMS がフィールドの属性シーケンスを送信したが、データは送信しなかったためです。

BMXATTR にはヌル値が含まれていることがあります。これは、SEND MAP 要求に DATAONLY オプションが指定されていて、アプリケーションがフィールド内のデータを更新し、属性を更新しないことが原因で発生することがあります。

## 累積マッピング操作

アプリケーションが累積マッピング (つまり、ACCUM オプションを指定した一連の SEND MAP コマンドの発行) を実行している場合、BMS は、複合表示を作成します。この表示では、単一ページの出力が複数の SEND MAP 要求から構成される場合があります。

累積マッピングが発生すると、各 SEND MAP 要求が処理される時点ではなく、ページの作成が完了した時点で XBMOUT 出口が呼び出されます。

## メッセージ・ルーティング

アプリケーションがルーティング・メッセージを作成する場合 (例えば、ROUTE コマンドに続いて、SET オプションまたは PAGING オプションを指定した 1 つ以上の SEND MAP コマンドを発行する場合)、XBMOUT 出口は、ルーティング指定のないマッピング要求の場合と同じ方法で呼び出されます。

ただし、ルーティング・メッセージの場合には、UEPBMTCT パラメーターはヌル値として渡されます。これは、ルーティング・メッセージは複数の装置にあてて送られる場合があります、BMS がルーティング・メッセージのターゲットとなる装置でサポートされる機能を最適化したためです。XBMOUT でルーティング・メッセージを処理する場合は、この種のどの装置について TCTTE を参照しても、多くの場合は関係ありません。

## ブリッジ機能出口 XFAINTU

ブリッジ機能出口は、新しいブリッジ機能が作成された直後、およびブリッジ機能が削除される直前に呼び出されます。

保持時間がゼロに指定されていた場合、またはこの機能が再使用される前に保持時間が満了した場合には、タスクの終了時にブリッジ機能が削除されます。

### 出口 XFAINTU

#### 呼び出される状況

ブリッジ機能が作成された直後、およびこの機能が削除される直前。

#### 出口固有のパラメーター

##### UEPFAREQ

出口が呼び出される理由を示す 1 バイト・フィールドのアドレス。可能な値は次のとおりです。

##### UEPFAIN

初期化。

##### UEPFATU

タイディアップ (整理)。

##### UEPFATUT

必要なタイディアップのタイプを示す 1 バイト・フィールドのアドレス。可能な値は次のとおりです。

##### UEPFANTU

通常のタイディアップ。

##### UEPFAETU

満了タイディアップ。

##### UEPFANAM

ブリッジ機能名のアドレス。

##### UEPFATYP

機能タイプを示す 1 バイト・フィールドのアドレス。値は常に以下のとおりです。

##### UEPFABR

3270 ブリッジ機能。

##### UEPFAUAA

ブリッジ機能ユーザー領域 (TCTUA) のアドレス。

##### UEPFAUAL

ブリッジ機能ユーザー領域の長さが含まれた 1 バイト・フィールドのアドレス。



#### **UEPFATK**

8 バイトの機能トークンのアドレス。

#### **UEPFAMCH**

このブリッジ機能を使ってブリッジ・トランザクションを開始する場合に使用するメカニズムを示す 1 バイト・フィールドのアドレス。可能な値は次のとおりです。

#### **UEPFASTA**

START BREXIT を使って開始されます。

#### **UEPFALNK**

DFHL3270 へのリンクを使って開始されます。

#### **UEPFAREG**

領域がブリッジ機能を使用しているか、またはそれがリモートかを示す 1 バイト・フィールドのアドレス。ブリッジ機能は AOR によって所有されます。この場合、ブリッジ機能はローカルで、ルーター領域に対してリモートです。AOR とルーターが同じ領域の場合は、XFAINTU を同じ領域内で 2 回呼び出すことができることに注意してください。可能な値は次のとおりです。

#### **UEPFAROU**

この領域はこのブリッジ機能のルーターです。

#### **UEPFAAOR**

この領域はこのブリッジ機能の AOR です。

### **戻りコード**

#### **UERCNORM**

処理が続行される。

#### **XPI 呼び出し**

リカバリー可能リソースを使用するもの以外は、すべて使用できます。

#### **API 呼び出し**

タスク関連ユーザー出口を呼び出すもの、およびリカバリー可能リソースを使用するもの以外は、すべて使用できます。

## **データ・テーブル管理出口 XDTRD、XDTAD、および XD TLC**

データ・テーブル管理出口は、CICS 共用データ・テーブルと CICS カップリング・ファシリティ・データ・テーブルの両方に適用されます。

XDTRD および XDTAD を使用して、データ・テーブルに組み込むレコードの選択を制御することができます。XDTRD は、ロード時にこのような選択を行うために使用されます。XDTAD は、その後、ロード済みのデータ・テーブル (またはロードを必要としない CFDT) にレコードを追加するときに呼び出されます。

XDTRD を使用して、ユーザー保守テーブルまたはカップリング・ファシリティ・データ・テーブルに組み込まれたレコードの内容を、それらのレコードを追加する前に変更することもできます。

CICS 共用データ・テーブルの場合、XD TLC を使用して、データ・テーブルがロードを完了したという事実に基づきアクションと実行することができます。これにより、ロード中のデータ・テーブルへのアクセスに対して課すことにしてあった制約を終了したり、ロードの完了に失敗した場合の対応をとったりすることができます。

カップリング・ファシリティ・データ・テーブルの場合、XD TLC を使用すると、ロードに失敗したカップリング・ファシリティ・データ・テーブルを受け入れるかどうかをグローバル・ユーザー出口プログラムが決定できるようになります。ユーザー出口プログラムがテーブルの受け入れを決定すると、そのテーブルは開いたままになり、アクセスできる状態のままとなりますが、CICS はロード完了を示すマークを付けません。これは、XD TLC 出口が使用可能になっていない場合のデフォルトのアクションでもあります。これは、アプリケーション・プログラムが、テーブルに正常にロードされたレコードのキー範囲外にあるすべてのレコードに関して LOADING 条件を受信し続けることを意味します。これにより、予期されたデータが必ずしもすべて使用可能ではないことをアプリケーション・プログラムが認識できるようになります。また、障害の原因が修正された後に、該当のロードを開始したファイルを閉じてから再び開くことによって、ロードを再試行することもできます。あるいは、同じデータ・テーブルを参照する別のロード可能ファイルを開くこともできます。出口プログラムがテーブルの拒否を決定すると、そのテーブルは閉じられ、既にロード済みのレコードはテーブル内に残ります。障害の原因が修正されると、そのデータ・テーブル

に対する後続のオープンによってロードを完了することができます。ソース・データ・セットからロードされないカップリング・ファシリティ・データ・テーブルの場合は、XDTLC は呼び出されません。

これらの出口点のいずれかから呼び出されるプログラムでは、フィールド UEPDTPL が指すデータ・テーブル・ユーザー制御パラメーター・リストを定義した DSECT を宣言する必要があることに注意してください。(UEPDTPL は DFHUEXIT 呼び出しによって定義されますが、UEPDTPL でアドレス指定するパラメーター・リストは定義されません。)この宣言をするには、DT\_UE\_PLIST DSECT を定義するコピーブック DFHXDTDS をプログラムで組み込むことができます。

テーブルで OPENTIME=STARTUP を指定している場合、またはテーブルが暗黙的に開かれる場合は、プログラム・リスト・テーブル初期設定後 (PLTPI) プログラムを提供して、ユーザー出口をアクティブにする必要があります。そうしないと、出口が使用可能になる前にデータ・テーブルがロードを開始する可能性があります。PLTPI プログラムについて詳しくは、[Writing initialization and shutdown programs](#) を参照してください。

注: CICS 共用データ・テーブル・サポートでこれらの出口を使用する場合の追加情報については、[共用データ・テーブルの概要](#)を参照してください。

## 出口 XDTRD

XDTRD ユーザー出口は、CICS がソース・データ・セットから取得したレコードをデータ・テーブルに追加しようとする直前に呼び出されます。

この処理は通常、ロード処理がソース・データ・セットの順次コピー中にレコードを取得するときに発生します。ただし、アプリケーションがデータ・テーブルにはないレコードを取得するときに、以下のいずれかの条件が当てはまる場合にも発生することがあります。

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

注: カップリング・ファシリティ・データ・テーブルの場合、XDTRD 出口はソース・データ・セットからロードされたテーブルについてのみ呼び出されます。

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

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

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

### 呼び出される状況

CICS がソース・データ・セットから取得したレコードをデータ・テーブルに追加しようとする直前。

### 出口固有のパラメーター

#### UEPDTPL

データ・テーブル・ユーザー出口パラメーター・リスト (コピーブック DFHXDTDS の DSECT DT\_UE\_PLIST によってマップされるリスト) のアドレス。データ・テーブル・ユーザー出口パラメーター・リストには以下のものが入っています。

#### UEPDNAM

8 文字のデータ・テーブル名。

**UEPDFTLG**

1 バイトのフラグ・フィールド。可能なビット設定は以下のとおりです。

**UEPDTSMT (X'80')**

出口は、CICS 共用データ・テーブル・サポートによって呼び出されました。

**UEPDTCMT (X'40')**

これは CICS 保守テーブルです。UEPDTSMT がオンの場合にのみ意味を持ちます。

**UEPDTOPT (X'20')**

出口は、テーブルをロードするために呼び出されました。つまり、スキップによる最適化を要求することができます。

**UEPDTCFT(X'10')**

出口は、カップリング・ファシリティー・データ・テーブル・サポートによって呼び出されました。

**UEPDUMT (X'08')**

これはユーザー保守テーブルです。UEPDTSMT がオンの場合にのみ意味を持ちます。

**UEPDTRA**

データ・レコードのアドレス。

**UEPDTRBL**

データ・テーブル・バッファのフルワード長。

**UEPDTRL**

データ・レコードのフルワード長。

ユーザー保守テーブルの場合、出口プログラムは、レコードを修正すると、このフィールドに新しい長さを設定することができます。

**UEPDTKA**

データ・テーブル・キーのアドレス。

**UEPDTKL**

データ・テーブル・キーのフルワード長。

**UEPDDSL**

ソース・データ・セットの名前のフルワード長。UEPDTSMT または UEPDTCFT のいずれかがオンの場合のみ意味を持ちます。

**UEPDTSN**

ソース・データ・セットの名前が含まれた 44 文字のフィールド。UEPDTSMT または UEPDTCFT のいずれかがオンの場合のみ意味を持ちます。

**UEPDTSKA**

スキップ・キー域のアドレス。テーブルをロードするために呼び出された場合、出口プログラムはこの領域に長さ UEPDTKL のキーを返すことができ、UERCDSOP の戻りコードを設定することによってロードの最適化を要求することができます。UEPDTSMT または UEPDTCFT のいずれかがオンの場合のみ意味を持ちます。

**戻りコード****UERCDSAC**

レコードをデータ・テーブルに追加します。

**UERCDSRJ**

レコードを拒否します。つまり、レコードをテーブルに追加しません。

**UERCDSOP**

スキップ・キー域に指定しキー以上のキーが見つかるまで、このレコードと後続のレコードをスキップします。UEPDTSMT または UEPDTCFT のいずれかがオンの場合のみ意味を持ちます。

**XPI 呼び出し**

すべて使用できます。

## 出口 XDTAD

出口 XDTAD は、データ・テーブルへの書き込み要求が発行されるときに呼び出されます。

ユーザー保守データ・テーブルおよびカップリング・ファシリティ・データ・テーブルの場合、レコードがデータ・テーブルに追加される前に、このユーザー出口が 1 回呼び出されます。CICS 保守データ・テーブルの場合、2 回 (レコードがソース・データ・セットに追加される前とその後レコードがデータ・テーブルに追加される前) ユーザー出口が呼び出されます。

**注:** カップリング・ファシリティ・データ・テーブルの場合、出口がオープン TCB で呼び出される可能性があります。そのため、出口がスレッド・セーフであることと、過度の TCB 切り替えを回避するため CICS に対してスレッド・セーフとして有効になっていることを確認してください。

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

XDTAD 出口では、レコード内のデータを変更してはなりません。XDTRD を使用して、データ・テーブルがロードされたときにデータを切り捨てる場合は、そのデータ・テーブル用の正しい形式でのみレコードの書き込みを試行するようにアプリケーションをコーディングする必要があります。

サンプルの XDTAD 出口プログラムは、[共用データ・テーブルの概要](#)にリストされています。

### 呼び出される状況

データ・テーブルへの書き込み要求の処理中に 1 回以上。

### 出口固有のパラメーター

#### UEPDTRL

データ・テーブル・ユーザー出口パラメーター・リスト (コピーブック DFHXDTDS の DSECT DT UE\_PLIST によってマップされるリスト) のアドレス。データ・テーブル・ユーザー出口パラメーター・リストには以下のものが入っています。

#### UEPDTRNAM

8 文字のデータ・テーブル名。

#### UEPDTRFLG

1 バイトのフラグ・フィールド。可能なビット設定は以下のとおりです。

#### UEPDTSDDT (X'80')

出口は、CICS 共用データ・テーブル・サポートによって呼び出されました。

#### UEPDTCMT (X'40')

これは CICS 保守テーブルです。UEPDTSDDT がオンの場合にのみ意味を持ちます。

#### UEPDTCFT (X'10')

出口は、カップリング・ファシリティ・データ・テーブル・サポートによって呼び出されました。

#### UEPDТУMT (X'08')

これはユーザー保守テーブルです。UEPDTSDDT がオンの場合にのみ意味を持ちます。

#### UEPDTRA

データ・レコードのアドレス。

#### UEPDTRBL

データ・テーブル・バッファのフルワード長。

#### UEPDTRL

データ・レコードのフルワード長。

#### UEPDTKA

データ・テーブル・キーのアドレス。

#### UEPDTKL

データ・テーブル・キーのフルワード長。

**UEPDTDSL**

ソース・データ・セットの名前のフルワード長。UEPDTSMT または UEPDTCFT のいずれかがオンの場合のみ意味を持ちます。

**UEPDTSN**

ソース・データ・セットの名前が含まれた 44 文字のフィールド。UEPDTSMT または UEPDTCFT のいずれかがオンの場合のみ意味を持ちます。

**戻りコード****UERC DTAC**

レコードをデータ・テーブルに追加します。

**UERC DTRJ**

レコードを拒否します。つまり、レコードをテーブルに追加しません。

**XPI 呼び出し**

すべて使用できます。

**出口 XDTLC**

XDTLC ユーザー出口は、データ・テーブルのロードの完了時に (ロードに成功したかどうかに関係なく) 呼び出されます。何らかの理由でロードの完了前にデータ・テーブルが閉じた場合、このユーザー出口は呼び出されません。カップリング・ファシリティ・データ・テーブルの場合、テーブルがソース・データ・セットからロードされる場合のみ、XDTLC 出口が呼び出されます。

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

**呼び出される状況**

テーブルのロードが完了したとき。データ・テーブルが閉じたためにロード処理が終了した場合、この出口は呼び出されません。

**出口固有のパラメーター****UEPD TPL**

データ・テーブル・ユーザー出口パラメーター・リスト (コピーブック DFHXDTDS の DSECT DT\_UE\_PLIST によってマップされるリスト) のアドレス。データ・テーブル・ユーザー出口パラメーター・リストには以下のものが入っています。

**UEPD TNAM**

8 文字のデータ・テーブル名。

**UEPD TFLG**

1 バイトのフラグ・フィールド。可能なビット設定は以下のとおりです。

**UEPD TSMT (X'80')**

出口は、CICS 共用データ・テーブル・サポートによって呼び出されました。

**UEPD TCMT (X'40')**

これは CICS 保守テーブルです。UEPDTSMT がオンの場合にのみ意味を持ちます。

**UEPD TCFT (X'10')**

出口は、カップリング・ファシリティ・データ・テーブル・サポートによって呼び出されました。

**UEPD TUMT (X'08')**

これはユーザー保守テーブルです。UEPDTSMT がオンの場合にのみ意味を持ちます。

**UEPD TORC**

データ・テーブルのオープン結果コード。可能な値は次のとおりです。

**UEPD TLC S**

ロードに成功しました。

**UEPD TFL**

ロードに失敗しました。

#### **UEPDTDSL**

ソース・データ・セットの名前のフルワード長。UEPDTSDT または UEPDTCFT のいずれかがオンの場合のみ意味を持ちます。

#### **UEPDTDSN**

ソース・データ・セットの名前が含まれた 44 文字のフィールド。UEPDTSDT または UEPDTCFT のいずれかがオンの場合のみ意味を持ちます。

### **戻りコード**

#### **UERCDTOK**

現在の状態のデータ・テーブルを受け入れます。

#### **UERCDTCL**

データ・テーブルを閉じます。

### **XPI 呼び出し**

すべて使用できます。

## **DBCTL インターフェース制御プログラム出口 (XXDFA)**

この出口は、DBCTL への接続が失敗した場合に、アクティブ CICS によって呼び出されます。

### **呼び出される状況**

DBCTL への接続が失敗した場合に、アクティブ CICS によって。出口プログラムは、アクティブ CICS が代替 CICS に失敗を通知した後で呼び出されます。

### **出口固有のパラメーター**

#### **UEPDBXR**

DBCTL で使用するための CICS XRF 情報のアドレス。CICS XRF 情報は、DSECT DFHDXUEP を使用してマップすることができます。

### **戻りコード**

#### **UERCNOAC**

何のアクションもしません。

#### **UERC SWCH**

代替 DBCTL に切り替えます。

#### **UERCABNO**

ダンプなしで CICS を異常終了します。

#### **UERCABDU**

ダンプありで CICS を異常終了します。

### **XPI 呼び出し**

TRANSACTION\_DUMP は使用しないでください。

## **DBCTL トラッキング・プログラム出口 (XXDFB、XXDTO)**

これらの出口は、DBCTL への接続に失敗した場合、または CICS がテークオーバーを実行する場合に呼び出されます。

### **出口 XXDFB**

XXDFB 出口は、DBCTL への接続が失敗したことを示すメッセージをアクティブ CICS から受信したときに呼び出されます。

### **呼び出される状況**

代替 CICS がアクティブ CICS から DBCTL への接続が失敗したことを示すメッセージを受信したときに、代替 CICS によって。代替およびアクティブ CICS システムは、異なる MVS イメージ内で (多くの場合は異なる中央演算処理複合システム (CPC) 内で) 実行されています。これらの出口の詳細については、[データベース制御 \(DBCTL\) の概要](#)を参照してください。

## 出口固有のパラメーター

### UEPDBXR

DBCTL で使用するための CICS XRF 情報のアドレス。CICS XRF 情報は、DSECT DFHDXUEP を使用してマップすることができます。

## 戻りコード

### UERCNOAC

何のアクションもしません。

### UERCSWCH

代替 DBCTL に切り替えます。

### UERCABNO

ダンプなしで CICS を異常終了します。

### UERCABDU

ダンプありで CICS を異常終了します。

戻りコード「UERCNORM」はこの出口点では使用できません。

## XPI 呼び出し

以下は使用しないでください。

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

## 出口 XXDTO

出口 XXDTO は、テークオーバーの実行時に代替 CICS によって呼び出されます。

## 呼び出される状況

以下の条件下でテークオーバーを実行するときに、代替 CICS によって。

- アクティブおよび代替 CICS システムが、異なる MVS イメージ内に (多くの場合は異なるプロセッサ内) に存在する。
- アクティブ CICS が DBCTL サブシステムに接続されていないか、または DBCTL サブシステムに接続しようとしていた。(これには、DBCTL から切断して別の DBCTL に再接続する場合は含まれません。)
- テークオーバーが XXDFB 出口によって開始されなかった。または、テークオーバーは XXDFB によって開始されたが、テークオーバーの発生前にアクティブ・サブシステムが DBCTL 接続を再確立し、新しい DBCTL テークオーバー決定のための XXDTO が呼び出された。

## 出口固有のパラメーター

### UEPDBXR

DBCTL で使用するための CICS XRF 情報のアドレス。CICS XRF 情報は、DSECT DFHDXUEP を使用してマップすることができます。

## 戻りコード

### UERCNOAC

何のアクションもしません。

### UERCSWCH

代替 DBCTL に切り替えます。

### UERCABNO

ダンプなしで CICS を異常終了します。

### UERCABDU

ダンプありで CICS を異常終了します。

戻りコード UERCNORM はこの出口点では使用できません。

### **XPI 呼び出し**

以下は使用しないでください。

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

## **ディスパッチャー・ドメイン出口 XDSBWT および XDSAWT**

XDSBWT 出口点および XDSAWT 出口点は、オペレーティング・システムの待機の前および後に配置します。これらの出口点から呼び出される出口プログラムでは、CICS サービスは使用できません。

XDSBWT 出口および XDSAWT 出口を使用して、CICS アドレス・スペースのスワップ状態を制御することができます。ただし、アドレス・スペースのデフォルトの状態がスワップ不能になっている場合は、これらの出口を使用してこの状態をオーバーライドすることはできません。

CICS は、SYSEVENT DONTSWAP 要求ごとに増加し、SYSEVENT OKSWAP 要求ごとに最小 0 まで減少するカウンタを使用します。SYSEVENT DONTSWAP 要求は、このカウンタが 0 から 1 に増加するときに発行されます。SYSEVENT OKSWAP 要求は、このカウンタが 1 から 0 に減少するときに発行されます。その他のすべての状況では、SYSEVENT は発行されません。

### **出口 XDSBWT**

この出口は、準再入可能 CICS TCB によって発行されるオペレーティング・システム待機の前に呼び出されます。

#### **呼び出される状況**

準再入可能 CICS TCB によって発行されるオペレーティング・システム待機の前。

#### **出口固有のパラメーター**

なし

#### **戻りコード**

##### **UERCNORM**

処理が続行される。

##### **UERC\_SWAP**

SYSEVENT を発行して、アドレス・スペースのスワッピングができるようにします。

### **XPI 呼び出し**

これは使用しないでください。

### **出口 XDSAWT**

この出口は、準再入可能 CICS TCB によって発行されるオペレーティング・システム待機の後で呼び出されます。

#### **呼び出される状況**

準再入可能 CICS TCB によって発行されるオペレーティング・システム待機の後。

#### **出口固有のパラメーター**

##### **UEPSYSRC**

オペレーティング・システム待機の前に行われた SYSEVENT 要求からの 4 バイトの戻りコードのアドレス。この戻りコードは次の 2 種類の形式のいずれかです。

1. SYSEVENT OKSWAP 戻りコード、または
2. SYSEVENT 要求が CICS によって拒否された場合は、以下のいずれかの 10 進数値を取る特殊な CICS 戻りコード

#### **17**

SYSEVENT OKSWAP は発行されませんでした。未処理の SYSEVENT OKSWAP 要求の数が SYSEVENT DONTSWAP 要求の数を超えました。SYSEVENT OKSWAP を発行するには、その前に SYSEVENT DONTSWAP を要求する必要があります。



SYSEVENT OKSWAP は発行されませんでした。未処理の SYSEVENT DONTSWAP 要求の数がまだ SYSEVENT OKSWAP の数を超えています。CICS で SYSEVENT OKSWAP を発行する前に、さらに SYSEVENT OKSWAP を要求する必要があります。

## 戻りコード

### UERCNORM

処理が続行される。

### UERCNOSW

SYSEVENT を発行して、アドレス・スペースのスワッピングを抑止します。

### XPI 呼び出し

これは使用しないでください。

## DL/I インターフェース・プログラム出口 XDLPRE および XDLIPOST

XDLPRE 出口点および XDLIPOST 出口点は、EXEC DLI コマンドまたは DL/I 呼び出しの発行に続いて呼び出されます。要求が処理される前に出口 XDLPRE が呼び出され、要求が処理された後に XDLIPOST が呼び出されます。

要求が機能シップされると、これらの出口が、アプリケーション専有領域とデータベース所有領域の両方から呼び出されます。ただし、データベース所有領域で呼び出される場合は、以下の制約があります。

1. 出口の説明は、アプリケーションのパラメーター・リストの一般形式を示しています。CALL レベルの DL/I パラメーター・リストの形式について詳しくは、「[IMS 製品資料内の『DL/I 呼び出しリファレンス』](#)」を参照してください。
2. すべての EXEC DLI 呼び出しの場合について、アプリケーションのパラメーター・リストはアセンブラ言語形式です。つまり、UEPLANG が指すプログラム言語バイトの値は常に UEPASM であり、UEPAPLIST が指すパラメーター・リストは常にアセンブラ言語形式です。この形式が使用されるのは、EXEC DLI 呼び出しがすべてアセンブラ言語の CALL レベル要求に変換されるためです。

EXEC DLI オンライン要求は、DFHEDP によって DFHDLI 用の CALL レベル要求に変換されます。IMS は、EXEC レベルのパラメーター・リストを直接扱いません。CALL パラメーター・リストの最初のパラメーターには、パラメーター・カウンターのアドレスが含まれています。CALL パラメーター・リストの 2 番目のパラメーターには、関数のアドレスが含まれています。他のすべてのパラメーターは、関数によって異なります。

3. XDLPRE 出口プログラムでは、PSB 名および SYSID 名を変更することができます。名前の変更は、本来指定されていた SYSID が失敗した場合に利用できます。

以下のような方法で SYSID を変更できます。

- ・ リモート値から別のリモート値へ
- ・ ローカル値からリモート値へ
- ・ リモート値からローカル値へ

SYSID の変更は、関連する PSB に PDIR エントリーがある場合にのみ有効です。この SYSID は、ローカル CICS (つまり、CICS 領域に指定された SYSIDNT) でも、リモート接続名でも構いません。新しい SYSID を使用するためには、PSB 名に PDIR エントリーが含まれている必要があります。PDIR エントリーが含まれていないと、ローカル CICS が DBCTL に接続されているものと見なされ、そこで IMS 要求を実行する試みが行われます。IMS スケジュール障害は、存在しない接続へのルーティングの失敗と同様に処理されます。SYSID がローカル CICS の SYSIDNT と同じ値またはブランク (16 進数 '40404040') に変更されると、CICS は IMS 要求をローカル・システムで実行しようとします。

## 出口 XDLPRE

出口 XDLPRE は、DL/I インターフェース・プログラムへの入り口で呼び出されます。

この出口で実行するプログラムは、スレッド・セーフ標準に従ってコーディングされ、CICS にスレッド・セーフとして定義されている必要があります。

## 出口固有のパラメーター

### UEPCTYPE

要求タイプ・バイトのアドレス。値は次のとおりです。

### UEPCEXEC

元の要求は EXEC DLI 要求でした。

### UEPCCALL

元の要求は CALL レベルの要求でした。

### UEPCSHIP

要求は別の領域から機能シッパされたものです。この値が設定されると、出口パラメーターの残り部分の設定および使用に、説明されている制限が適用されます。

### UEPAPLIST

アプリケーションのパラメーター・リストのアドレス。COBOL およびアセンブラー言語の一般的な形式は次のとおりです。

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

PL/I の一般的な形式は次のとおりです。

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

UEPCTYPE が UEPCSHIP ではない場合、出口プログラムでアプリケーション・パラメーター・リスト内の任意のパラメーターを変更できます。UEPCSHIP 要求の場合、出口プログラムではいずれのパラメーターも **変更できません**。さらに、UEPCSHIP 要求の場合、UEPAPLIST は上記の形式のパラメーター・リストのコピーを指しますが、最初の 2 つのパラメーター parm1 および parm2 のみが含まれています。

注: PL/I アプリケーションの場合、parm1 にパラメーター・カウントが含まれている場合もあり、含まれていない場合もあります。出口プログラムでは、このフィールドを使用前に検査する必要があります。

### UEPLANG

プログラム言語バイトのアドレス。値は次のとおりです。

### UEPPLI

PL/I

### UEPCBL

COBOL

### UEPASM

アセンブラー言語

UEPCSHIP 要求の場合、言語は常にアセンブラーです。

### UEPIOAX

入出力域存在フラグ・バイトのアドレス。

### UEPIOA1

入出力域が存在します。

UEPCSHIP 要求の場合、入出力域存在フラグは常にオフです。

### UEPIOA

入出力域のアドレス。これはアプリケーションの IOAREA です。EXEC DLI の場合、これは DFHEDP の IOAREA です。IOAREA の内容は出口で上書きすることができます。新しい内容は

DL/I 要求の処理時に使用されます。ただし、IOAREA はプログラムの静的ストレージ内にある場合があります。その場合は上書きしてはならないことに注意してください。

UEPCSHIP 要求の場合、UEPIOA は常にゼロです。

#### **UEPPSBNX**

PSB 存在フラグ・バイトのアドレスで、次のとおりです。

#### **UEPPSB1**

PSB が存在します。

#### **UEPPSBNM**

8 文字の PSB 名が含まれた領域のアドレス。UEPCSHIP を含めてすべてのタイプの要求の場合に、この領域の内容を出口で上書きすることができます。新しい内容は DL/I 要求の処理時に使用されます。

#### **UEPSYSDX**

SYSID 存在フラグ・バイトのアドレスで、次のとおりです。

#### **UEPSYS1**

SYSID が存在します。

#### **UEPSYSID**

4 文字の SYSID 名が含まれた領域のアドレス。UEPCSHIP を含めてすべてのタイプの要求の場合に、この領域の内容を出口で上書きすることができます。新しい内容は DL/I 要求の処理時に使用されます。

### 戻りコード

#### **UERCNORM**

処理を続行します。

#### **UERCBYP**

DL/I 要求をバイパスし、戻ります。

#### **UERCPURG**

タスクは XPI 呼び出し中にページされます。

### XPI 呼び出し

すべて使用できます。

## 出口 XDLIPOST

出口 XDLIPOST は、DL/I インターフェース・プログラムからの出口で呼び出されます。

この出口で実行するプログラムは、スレッド・セーフ標準に従ってコーディングされ、CICS にスレッド・セーフとして定義されている必要があります。

### 出口固有のパラメーター

#### **UEPCTYPE**

要求タイプ・バイトのアドレス。値は次のとおりです。

#### **UEPCEXEC**

EXEC DLI 要求。

#### **UEPCCALL**

CALL レベルの要求。

#### **UEPCSHIP**

要求は別の領域から機能シッパされたものです。この値が設定されると、出口パラメーターの残り部分の設定および使用に、説明されている制限が適用されます。

#### **UEAPLIST**

アプリケーションのパラメーター・リストのアドレス。COBOL およびアセンブラー言語の一般的な形式は次のとおりです。

```
plist address --> parm1 address --> parm1
                parm2 address --> parm2
                parm3 address --> parm3
                .....
```

up to a maximum of 18 parameters  
excluding the optional parmcount.

PL/I の一般的な形式は次のとおりです。

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

UEPCTYPE が UEPCSHIP ではない場合、出口プログラムでアプリケーション・パラメーター・リスト内の任意のパラメーターを変更できます。UEPCSHIP 要求の場合、出口プログラムではいずれのパラメーターも **変更できません**。さらに、UEPCSHIP 要求の場合、UEAPLIST は前の形式のパラメーター・リストのコピーを指しますが、最初の 2 つのパラメーター parm1 および parm2 のみが含まれています。25 ページの『[DL/I インターフェース・プログラム出口 XDLPRE および XDLPPOST](#)』も参照してください。

**注:** PL/I アプリケーションの場合は、parm1 にパラメーター・カウントが含まれている場合もあり、含まれていない場合もあります。出口プログラムでは、このフィールドを使用前に検査する必要があります。

#### UEPLANG

プログラム言語バイトのアドレス。値は次のとおりです。

##### UEPPLI

PL/I

##### UEPCBL

COBOL

##### UEPASM

アセンブラー言語

UEPCSHIP 要求の場合、言語は常にアセンブラーです。

#### UEPIOAX

入出力域存在フラグ・バイトのアドレス。

##### UEPIOA1

入出力域が存在します。

UEPCSHIP 要求の場合、入出力域存在フラグは常にオフです。

#### UEPIOA

入出力域のアドレス。これはアプリケーションの IOAREA です。EXEC DLI の場合、これは DFHEDP の IOAREA です。IOAREA の内容は出口で上書きすることができ、新しい形式でアプリケーション・プログラムに返されます。ただし、アプリケーションの IOAREA はプログラムの静的ストレージ内にある場合があります。その場合は上書きしてはならないことに注意してください。

UEPCSHIP 要求の場合、UEPIOA は常にゼロです。

#### UEPUIBX

UIB 存在フラグ・バイトのアドレスで、次のとおりです。

##### UEPUIB1

UIB が存在します。

#### UEPUIB

UIB のアドレスで、モジュール DFHDBCOP の DFHUIB によってマップされます。UEPCSHIP を含めたすべてのタイプの要求の場合に、UIB の内容は上書きすることができます。新しい内容がアプリケーションか、またはその要求を機能シップした領域に返されます。

#### 戻りコード

##### UERCNORM

処理が続行される。

## UERC PURG

タスクは XPI 呼び出し中にパージされます。

## XPI 呼び出し

すべて使用できます。

## グローバル・ユーザー出口ルーチン XDLIPRE の使用例

グローバル・ユーザー出口ルーチン XDLIPRE を使用して、実行時にアプリケーション・プログラムによってスケジュールされた PSB 名を変更することができます。XDLIPRE 出口を使用して、SYSID の ID を変更し、使用不可になる SYSID から使用可能な SYSID に作業を切り換えるようにすることもできます。

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

次の図に示す XDLIPRE の例は、コピーして変更することができます。この例は、ガイダンスの目的でのみ提供されています。グローバル・ユーザー出口に関するプログラミング情報については、[DFHZNEPI TYPE=INITIAL: デフォルト・ルーチンの指定](#)を参照してください。

```
*****
* 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
```

図 1. PSB 名を変更する XDLIPRE ユーザー出口の例 1/6

```

*
*      COPY  DFHTRPTY          DEFINE PARAMETER LIST FOR
*                               USE BY DFHTRPTX MACRO
*
*      COPY  DFHSMCMCY        DEFINE PARAMETER LIST FOR
*                               USE BY DFHSMCMCX 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

```

図 2. PSB 名を変更する XDLIPRE ユーザー出口の例 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?
*       B0      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?

```

図 3. PSB 名を変更する XDLIPRE ユーザー出口の例 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),DUMMYPSB
        MVC      MESS1,MESS2T      SET UP MESSAGE
        MVC      OLDPSB,0(R6)      SUPPLIED PSB NAME
        MVC      NEWPSB,=CL8''     CLEAR FIELD
        LA       R1,USERCNORM      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
*

```

図 4. PSB 名を変更する XDLIPRE ユーザー出口の例 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(*)                            X

*
*****
*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(*)                             X

*
*****
*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                                  *
*****
*

```

図 5. PSB 名を変更する XDLIPRE ユーザー出口の例 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'
DUMMYP	DC	CL8' '
	LTORG	
	END	DLIPR

図 6. PSB 名を変更する XDLIPRE ユーザー出口の例 6/6

## ダンプ・ドメイン出口 XDUREQ、XDUREQC、XDUCLSE、および XDUOUT

ダンプ・ドメイン出口を使用して、トランザクション・ダンプまたはシステム・ダンプの前後に情報を取得することができます。

### 出口 XDUREQ

出口 XDUREQ は、システム・ダンプまたはトランザクション・ダンプが取られる直前に呼び出されます。

#### 出口固有のパラメーター

##### UEPTRANID

4 バイトのトランザクション ID のアドレス。

##### UEPUSER

8 バイトのユーザー ID のアドレス。

##### UEPTERM

4 バイトの端末 ID のアドレス。

##### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス、または NULL (現行アプリケーションが存在しない場合)。

##### UEPDUMPC

8 バイトのダンプ・コードのコピーのアドレス。

##### UEPABCDE

xxx/yyyy 形式の 8 バイトのカーネル・エラー・コードのコピーのアドレス。xxx は 3 桁の 16 進数 MVS 完了コード (例えば OC1 や D37) を示します。MVS 完了コードが適用外の場合、xxx は 3 つのハイフンです。4 桁のコード yyyy は、CICS またはシステム上の別の製品によって作成されたユーザー異常終了コードです。UEPABCDE は、次のダンプ・コードに対応する異常終了コードの場合にのみ完了します。

- AP0001
- SR0001
- ASRA
- ASRB
- ASRD

その他の場合、このフィールドにはヌル文字が含まれています。

##### UEPDUMPT

以下のいずれかの値が含まれた 1 バイトのダンプ・タイプ ID のアドレス。

##### UEPDTRAN

トランザクション・ダンプが要求されました。

##### UEPDSYST

システム・ダンプが要求されました。

注: ダンプ・タイプ ID は、ダンプ・ドメインに渡されたダンプ要求のタイプを示します。ダンプ・テーブル内のユーザー・エントリーによって元の要求に対しておこなわれた可能性のある変更は、これには反映されません。

##### UEPXDSCP

現行ダンプ・テーブルの DUMPSCOPE の設定を示す 1 バイト・フィールドのアドレス。これには、以下のいずれかの値が含まれています。

##### UEPXDLOC

システム・ダンプはローカル MVS イメージのみで取られます。

##### UEPXDREL

ローカル MVS イメージと、シスプレックス内の関連の MVS イメージの両方でシステム・ダンプが取られます。

このフィールドは、ダンプ・テーブルの DUMPSCOPE の設定を更新するために出口で変更することができます。

#### **UEPXD TXN**

現行ダンプ・テーブルの TRANDUMP の設定を示す 1 バイト・フィールドのアドレス。これには、以下のいずれかの値が含まれています。

##### **UEPXDYES**

トランザクション・ダンプが取られます。

##### **UEPXDNO**

トランザクション・ダンプは取られません。

このフィールドは、ダンプ・テーブルの TRANDUMP の設定を更新するために出口で変更することができます。

注：このフィールドは、UEPDUMPT に値 UEPDTRAN が含まれている場合にのみ有効です。

#### **UEPXD SYS**

現行ダンプ・テーブルの SYSDUMP の設定を示す 1 バイト・フィールドのアドレス。これには、以下のいずれかの値が含まれています。

##### **UEPXDYES**

システム・ダンプが取られます。

##### **UEPXDNO**

システム・ダンプは取られません。

このフィールドは、ダンプ・テーブルの SYSDUMP の設定を更新するために出口で変更することができます。

#### **UEPXD TRM**

現行ダンプ・テーブルの SHUTDOWN の設定を示す 1 バイト・フィールドのアドレス。これには、以下のいずれかの値が含まれています。

##### **UEPXDYES**

CICS システムは、まもなくシャットダウンします。

##### **UEPXDNO**

CICS システムはシャットダウンしません。

このフィールドは、ダンプ・テーブルの SHUTDOWN の設定を更新するために出口で変更することができます。

#### **UEPXD MAX**

現行ダンプ・テーブルの MAXIMUM の設定が含まれた 4 バイト・フィールドのアドレス。このフィールドは、現行ダンプ・テーブルの MAXIMUM の設定を変更するために出口で変更することができます。MAXIMUM の設定への変更がこのダンプ要求を抑止することはありません。UERCYP の戻りコードを使用して、現行のダンプ要求を抑止することができます。

#### **UEPD XDCNT**

現行ダンプ・テーブルの CURRENT の設定が含まれた 4 バイト・フィールドのアドレス。

#### **UEPD TST**

このダンプ・コードの現行ダンプ・テーブルの統計が含まれた 16 バイト・フィールドのアドレス。アドレス指定されたフィールドは、2 進整数が含まれた以下の 4 つの 4 バイト・フィールドで構成されます。

- トランザクション・ダンプの実施回数
- トランザクション・ダンプの抑止回数
- システム・ダンプ実施回数
- システム・ダンプの抑止回数

注：トランザクション・ダンプの統計は、UEPDUMPT に値 UEPDTRAN が含まれている場合にのみ有効です。

**UEPXDDAE**

現行ダンプ・テーブルの DAEOPTION の設定を表す 1 バイト・フィールドのアドレスです。これには、以下のいずれかの値が含まれています。

**UEPXDYES**

ダンプは DAE 抑止に適格です。

**UEPXDNO**

ダンプは DAE によって抑止されません。

このフィールドは、ダンプ・テーブルの DAEOPTION の設定を更新するために出口で変更することができます。

**UEPDMPID**

ダンプ ID が含まれた xxxx/xxxx 形式の 9 文字のフィールドのアドレス。ダンプ ID は、対応するダンプ・メッセージによって出力される ID と同じです。

**UEPFMOD**

8 バイト領域のアドレス。この領域には、ダンプ・コードが AP0001 の場合は障害が発生したモジュールの名前が含まれ、それ以外の場合はヌル文字が含まれています。

フィールド UEPPROG は、障害が発生した場所に関係なく、常に現行のアプリケーションの名前をアドレス指定することに注意してください。障害が発生したモジュールが既知の場合、UEPFMOD はそのモジュールの名前をアドレス指定します。

ダンプ・コードが AP0001 の場合、以下の 3 つの可能性があります。

1. UEPFMOD によってアドレス指定されたフィールドに、UEPPROG によってアドレス指定されたフィールドと同じ名前が含まれている場合。障害はアプリケーション・コードで発生しました。
2. UEPFMOD によってアドレス指定されたフィールドに、UEPPROG によってアドレス指定されたフィールドとは別の名前が含まれている場合。障害はアプリケーション・コード以外で発生しました。
3. UEPFMOD によってアドレス指定されたフィールドに「????????」が含まれている場合。障害はアプリケーション・コード内にはないが、CICS は障害が発生したモジュールの名前を判別できませんでした。

**UEPDLISI**

4 バイトの DSPLIST のアドレス。

**UEPJLISI**

4 バイトの JOBLIST のアドレス。

**戻りコード****UERCNORM**

処理が続行される。

**UERCBYD**

ダンプを抑止します。

**UERCPUFG**

タスクは XPI 呼び出し中にページされます。

**XPI 呼び出し**

UEDUMPT がトランザクション・ダンプの取得中であることを示している場合のみ、WAIT\_MVS を使用できます。その他の呼び出しを使用しないでください。

**出口 XDUREQC**

出口 XDUREQC は、システムまたはトランザクションのタスクが取られた直後、またはそのいずれかのダンプが失敗したか抑止された直後に呼び出されます。

**出口固有のパラメーター****UEPTRANID**

4 バイトのトランザクション ID のアドレス。

**UEPUSER**

8 バイトのユーザー ID のアドレス。

**UEPTERM**

4 バイトの端末 ID のアドレス。

**UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。

**UEPDUMPC**

8 バイトのダンプ・コードのコピーのアドレス。

**UEPABCDE**

xxx/yyyy 形式の 8 バイトのカーネル・エラー・コードのコピーのアドレス。xxx は 3 桁の 16 進数 MVS 完了コード (例えば X'0C1' や X'D37') を示します。MVS 完了コードが適用外の場合、xxx は 3 つのハイフンです。4 桁のコード yyyy は、CICS またはシステム上の別の製品によって作成されたユーザー異常終了コードです。UEPABCDE は、次のダンプ・コードに対応する異常終了コードの場合にのみ完了します。

- AP0001
- SR0001
- ASRA
- ASRB
- ASRD

その他の場合、このフィールドにはヌル文字が含まれています。

**UEPDUMPT**

以下のいずれかの値が含まれた 1 バイトのダンプ・タイプ ID のアドレス。

**UEPDTRAN**

トランザクション・ダンプが要求されました。

**UEPDSYST**

システム・ダンプが要求されました。

注: ダンプ・タイプ ID は、ダンプ・ドメインに渡されたダンプ要求のタイプを示します。ダンプ・テーブル内のユーザー・エントリーによって元の要求に対しておこなわれた可能性のある変更は、これには反映されません。

**UEPXDSCP**

現行ダンプ・テーブルの DUMPSCOPE の設定を示す 1 バイト・フィールドのアドレス。これには、以下のいずれかの値が含まれています。

**UEPXDLOC**

システム・ダンプはローカル MVS イメージのみで取られます。

**UEPXDREL**

ローカル MVS イメージと、シスプレックス内の関連の MVS イメージの両方でシステム・ダンプが取られます。

このフィールドは、ダンプ・テーブルの DUMPSCOPE の設定を更新するために出口で変更することができます。

**UEPDXTXN**

現行ダンプ・テーブルの TRANDUMP の設定を示す 1 バイト・フィールドのアドレス。これには、以下のいずれかの値が含まれています。

**UEPXDYES**

トランザクション・ダンプが取られます。

**UEPXDNO**

トランザクション・ダンプは取られません。

このフィールドは、ダンプ・テーブルの TRANDUMP の設定を更新するために出口で変更することができます。

注: このフィールドは、UEPDUMPT に値 UEPDTRAN が含まれている場合にのみ有効です。

#### **UEPXDSYS**

現行ダンプ・テーブルの SYSDUMP の設定を示す 1 バイト・フィールドのアドレス。これには、以下のいずれかの値が含まれています。

#### **UEPXDYES**

システム・ダンプが取られます。

#### **UEPXDNO**

システム・ダンプは取られません。

このフィールドは、ダンプ・テーブルの SYSDUMP の設定を更新するために出口で変更することができます。

#### **UEPXDTRM**

現行ダンプ・テーブルの SHUTDOWN の設定を示す 1 バイト・フィールドのアドレス。これには、以下のいずれかの値が含まれています。

#### **UEPXDYES**

CICS システムは、まもなくシャットダウンします。

#### **UEPXDNO**

CICS システムはシャットダウンしません。

このフィールドは、ダンプ・テーブルの SHUTDOWN の設定を更新するために出口で変更することができます。

#### **UEPXDMAX**

現行ダンプ・テーブルの MAXIMUM の設定が含まれた 4 バイト・フィールドのアドレス。このフィールドは、現行ダンプ・テーブルの MAXIMUM の設定を変更するために出口で変更することができます。

#### **UEPDXDCNT**

現行ダンプ・テーブルの CURRENT の設定が含まれた 4 バイト・フィールドのアドレス。

#### **UEPXDST**

このダンプ・コードの現行ダンプ・テーブルの統計が含まれた 16 バイト・フィールドのアドレス。アドレス指定されたフィールドは、2 進整数が含まれた以下の 4 つの 4 バイト・フィールドで構成されます。

- ・ トランザクション・ダンプの実施回数
- ・ トランザクション・ダンプの抑止回数
- ・ システム・ダンプ実施回数
- ・ システム・ダンプの抑止回数。

注: トランザクション・ダンプの統計は、UEPDUMPT に値 UEPDTRAN が含まれている場合にのみ有効です。

#### **UEPXDDAE**

現行ダンプ・テーブルの DAEOPTION の設定を表す 1 バイト・フィールドのアドレスです。これには、以下のいずれかの値が含まれています。

#### **UEPXDYES**

ダンプは、DAE によって抑制されました。

#### **UEPXDNO**

ダンプは DAE によって抑止されませんでした。

このフィールドは、ダンプ・テーブルの DAEOPTION の設定を更新するために出口で変更することができます。

#### **UEPDMPID**

ダンプ ID が含まれた xxxx/xxxx 形式の 9 文字のフィールドのアドレス。ダンプ ID は、対応するダンプ・メッセージによって出力される ID と同じです。

**UEPDRESP**

2 バイトのダンプ応答コードのアドレス。

**UEPDREAS**

2 バイトのダンプ理由コードのアドレス。

**UEPDLISO**

4 バイトの DSPLIST のアドレス。

**UEPJLISO**

4 バイトの JOBLIST のアドレス。

**戻りコード****UERCNORM**

処理が続行される。

**XPI 呼び出し**

UEPDUMPT がトランザクション・ダンプの取得中であることを示している場合のみ、WAIT\_MVS を使用できます。その他の呼び出しを使用しないでください。

**出口 XDUCLSE**

この出口は、トランザクション・ダンプ・データ・セットが閉じた直後に呼び出されます。

**呼び出される状況**

トランザクション・ダンプ・データ・セットが閉じた直後。

**出口固有のパラメーター****UEPTRANID**

4 バイトのトランザクション ID のアドレス。

**UEPUSER**

8 バイトのユーザー ID のアドレス。

**UEPTERM**

4 バイトの端末 ID のアドレス。

**UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。

**UEPDMPDD**

8 バイトのダンプ・データ・セット DD 名のアドレス。

**UEPDMPSN**

44 バイトのダンプ・データ・セット DS 名のアドレス。

**戻りコード****UERCNORM**

処理が続行される。

**UERCSWCH**

自動スイッチ・フラグがオンに設定されます (DUMPSW=ALL がまだ設定されていない場合)。

**XPI 呼び出し**

WAIT\_MVS を使用できます。その他の呼び出しを使用しないでください。

**出口 XDUOUT**

この出口は、トランザクション・ダンプ・データ・セットにレコードを書き込む前に呼び出されます。

**呼び出される状況**

トランザクション・ダンプ・データ・セットにレコードを書き込む前。

**出口固有のパラメーター****UEPTRANID**

4 バイトのトランザクション ID のアドレス。

**UEPUSER**

8 バイトのユーザー ID のアドレス。

**UEPTERM**

4 バイトの端末 ID のアドレス。

**UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。

**UEPDMPFC**

1 バイトの機能コードのアドレス。同等の値は以下のとおりです。

**UEPDMPWR**

バッファに書き込もうとしています。

**UEPDMPRE**

自動スイッチの後でダンプを再始動しようとしています。

**UEPDMPAB**

ダンプの異常終了。

**UEPDMPDY**

バッファに書き込もうしていますが、CICS ダンプ・データ・セットがダミー・ファイルであるか、閉じています。

**UEPDMPBF**

ダンプ・バッファのアドレス。この長さは、パラメーター UEPDMPLEN によってアドレス指定されます。

**UEPDMPLEN**

2 バイトのダンプ・バッファ長のアドレス。

**戻りコード****UERCNORM**

処理が続行される。

**UERCBYD**

ダンプ・レコード出力を抑止します。

**XPI 呼び出し**

WAIT\_MVS を使用できます。その他の呼び出しを使用しないでください。

## エンキュー EXEC インターフェース・プログラム出口 XNQEREQ および XNQEREQC

XNQEREQ 出口を使用して、エンキュー API 要求に対するアクションが実行される前に、この要求をインターセプトできます。XNQEREQC 出口を使用して、エンキュー API 要求が完了した後に、この要求の応答をインターセプトできます。

影響を受ける API 要求は、次のとおりです。

- EXEC CICS ENQ
- EXEC CICS DEQ

**XNQEREQ**

XNQEREQ を使用して、以下のことができます。

- API パラメーター・リストの分析 (関数、キーワード、引数値、および応答)
- 要求実行前の入力パラメーター値の変更
- 要求実行の回避これにより、CICS 機能をユーザー独自の処理で置き換えることができます。

**XNQEREQ を使用して ENQ または DEQ のスコープを変更する場合の注：**

1. XNQEREQ により、既存のアプリケーションを、アプリケーションを変更せずにシスプレックス・エンキューを使用するように変換することができます。



注: 多くの場合、これは ENQMODEL リソース定義またはユーザー出口のいずれかを使用して行うことができます。ただし、リソース名が動的に決定されるため事前に認識されていないアプリケーションの場合には、この出口を使用した場合のみこの変換を行うことができます。

2. シスプレックス・スコープ・エンキューと領域スコープ・エンキューでは、別個の名前空間を使用します。領域スコープ・エンキューがシスプレックス・エンキューを待機することではなく、シスプレックス・スコープ・エンキューが領域エンキューを待機することもあります。

注: この状況は、この出口を使用する場合にのみ起こります。ENQ または DEQ の SCOPE を定義するための唯一の方法として ENQMODEL リソース定義を使用することで、このリスクが発生する可能性を回避することができます。

3. スtring ENQ については領域スコープとシスプレックス・スコープの両方がサポートされていますが、アドレス ENQ についてはシスプレックス・スコープはサポートされていません。

## **XNQEREQC**

**XNQEREQC** を使用して、以下のことができます。

- API パラメーター・リストの分析
- XNQEREQ 出口プログラムと XNQEREQC 出口プログラムが同じ要求のために呼び出された場合、それらのプログラム間でのデータの受け渡し
- 複数のエンキュー出口プログラムが同じタスク内で呼び出された場合、それらのプログラム間でのデータの受け渡し。

CICS は、エンキュー EXEC インターフェース用のサンプル出口プログラム DFH\$XNQE を提供します。詳しくは、[エンキュー EXEC インターフェース・サンプル出口プログラム: DFH\\$XNQE](#) を参照してください。

## **出口 XNQEREQ**

この出口は、CICS が EXEC CICS ENQ または DEQ 要求を処理する前、またはその要求をインストール済みの ENQMODEL リソース定義と突き合わせようとする前に呼び出されます。

### **呼び出される状況**

CICS が EXEC CICS ENQ または DEQ 要求を処理する前、またはその要求をインストール済みの ENQMODEL リソース定義と突き合わせようとする前。

### **出口固有のパラメーター**

#### **UEPCLPS**

コマンド・パラメーター・リストのコピーのアドレス。43 ページの『コマンド・レベルのパラメーター構造』を参照してください。

#### **UEPNQTOK**

単一エンキュー要求での XNQEREQ と XNQEREQC の間の情報の受け渡しに使用できる 4 バイト領域のアドレス。

#### **UEPRCODE**

EIB 戻りコード EIBRCODE の 6 バイトの 16 進数コピーのアドレス。EIB 戻りコードの詳細については、以下を参照してください。[EIB フィールド](#)。

#### **UEPRES**

EIB 応答コード EIBRESP の 4 バイトのバイナリー・コピーのアドレス。

#### **UEPRES2**

EIB 応答コード EIBRESP2 の 4 バイトのバイナリー・コピーのアドレス。

#### **UEPTSTOK**

同じタスク内の連続するエンキュー要求間 (例えば、XNQEREQ 出口の連続呼び出し間) での情報の受け渡しに使用できる 4 バイトのトークンのアドレス。

#### **UEPRECUR**

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

#### **UEPScope**

使用する 4 バイトの ENQSCOPE 名のアドレス。

## 戻りコード

### UERCBYBYP

この要求をバイパスします。

### UERCNORM

処理が続行される。

### UERCPURG

タスクは XPI 呼び出し中にページされます。

### UERCSCPE

ENQSCOPE 名が指定されました。

## XPI 呼び出し

すべて使用できます。

## API コマンドおよび SPI コマンド

次のコマンドを除き、すべてを使用できます。

EXEC CICS SHUTDOWN

EXEC CICS XCTL

注：再帰的コマンドを発行するときには注意してください。例えば、XNQEREQ 出口からエンキュー要求を発行するときには、ループに入らないようにする必要があります。再帰カウンター UEPRECUR を使用することをお勧めします。

## 出口 XNQEREQC

出口 XNQEREQC は、エンキュー API 要求が完了した後で、エンキュー EXEC インターフェース・プログラムから戻る前に呼び出されます。

### 出口固有のパラメーター

#### UEPCLPS

コマンド・パラメーター・リストのコピーのアドレス。[43 ページの『コマンド・レベルのパラメーター構造』](#)を参照してください。

#### UEPNQTOK

単一エンキュー要求での XNQEREQ と XNQEREQC の間の情報の受け渡しに使用できる 4 バイト領域のアドレス。

#### UEPRCODE

EIB 戻りコード EIBRCODE の 6 バイトの 16 進数コピーのアドレス。EIB 戻りコードの詳細については、[以下を参照してください。EIB フィールド。](#)

#### UEPRES

EIB 応答コード EIBRESP の 4 バイトのバイナリー・コピーのアドレス。

#### UEPRES2

EIB 応答コード EIBRESP2 の 4 バイトのバイナリー・コピーのアドレス。

#### UEPTSTOK

同じタスク内の連続するエンキュー要求間 (例えば、XNQEREQC 出口の連続呼び出し間) での情報の受け渡しに使用できる 4 バイトのトークンのアドレス。

#### UEPRECUR

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

#### UEPSCOPE

使用する 4 バイトの ENQSCOPE 名のアドレス。

## 戻りコード

### UERCNORM

処理が続行される。

### UERCPURG

タスクは XPI 呼び出し中にページされます。

## XPI 呼び出し

すべて使用できます。

## API コマンドおよび SPI コマンド

次のコマンドを除き、すべてを使用できます。

```
EXEC CICS SHUTDOWN  
EXEC CICS XCTL
```

パラメーター・リストに指定した EIBRCODE、EIBRESP、および EIBRESP2 のコピーを更新することができます。値を更新した場合、XNQEREQC の完了後に、または XNQEREQ に戻りコード UERCBYP を指定してある場合に、CICS は新しい値をアプリケーション・プログラムの EIB にコピーします。

有効なエンキュー応答を設定する必要があります。EIBRCODE、EIBRESP、および EIBRESP2 の 3 つすべてを整合性のある値セット (有効な完了を記述するためにエンキュー・ドメインによって設定される値セットなど) に設定する必要があります。CICS では、EIBRCODE、EIBRESP、および EIBRESP2 の整合性は検査されません。EIBRCODE がゼロ以外の値に設定され、EIBRESP がゼロに設定されると、CICS はゼロ以外の値で EIBRESP をオーバーライドします。EIBRCODE、EIBRESP、および EIBRESP2 の値の設定に役立つように、エンキュー・ドメインで使用する値が DSECT DFHNQUED に指定されます。

**注:** 再帰的コマンドを発行する際には、ループが生じないように注意してください。例えば、XNQEREQC 出口からエンキュー要求を発行する場合に、ループに入らないようにユーザーが注意する必要があります。再帰カウンター UEPRECUR を使用することをお勧めします。

## コマンド・レベルのパラメーター構造

コマンド・レベルのパラメーター構造は、一連のアドレスで構成されます。最初のアドレスは、EXEC インターフェース記述子 (EID) を指します。これは、要求のタイプを記述するビット・ストリングからなり、要求で指定された各キーワードを識別します。残りのアドレスは、その要求に関連付けられたデータ部分を指します。

### パラメーター・リスト終了標識

EID を調べて、要求のタイプおよび指定されたキーワードを判別することができます。リスト内のその他のパラメーターを調べて、キーワードの値を判別することができます。要求で指定されたキーワードの値を変更することもできます。

パラメーター・リスト内の最後のアドレス・セットの高位ビットが、リスト内の最後のアドレスであることを示すために設定されます。ユーザー出口プログラムから戻ったときに、CICS は、パラメーター・リストをスキャンして高位ビットを探し、最終パラメーターを検出します。したがって、パラメーター・リストの長さを変更した場合は、高位ビットをリセットしてどれが新しい最終アドレスかを示す必要もあります。

### UEPCLPS 出口固有パラメーター

UEPCLPS 出口固有パラメーターは、出口 XNQEREQ と出口 XNQEREQC の両方に組み込まれ、コマンド・レベルのパラメーター構造を格納します。

コマンド・レベルのパラメーター構造には、NQ\_ADDR0 から NQ\_ADDR3 までの 4 つのアドレスが含まれています。この構造は DSECT NQ\_ADDR\_LIST で定義されます。DSECT NQ\_ADDR\_LIST は、COPY DFHNQUED ステートメントを組み込むことによって出口プログラムにコピーする必要があります。

コマンド・レベルのパラメーター・リストは、以下のように構成されています。

### NQ\_ADDR0

EID という 9 バイトの領域のアドレスのことで、以下のように構成されています。

- NQ\_GROUP
- NQ\_FUNCT
- NQ\_BITS1
- NQ\_BITS2
- NQ\_EIDOPT5
- NQ\_EIDOPT6

- **NQ\_EIDOPT7**

- **NQ\_EIDOPT8**

#### **NQ\_GROUP**

常に X'12' で、これがタスク制御要求であることを示します。

#### **NQ\_FUNCT**

以下の要求のタイプを定義する 1 バイトの領域です。

**X'04'**

ENQ

**X'06'**

DEQ

#### **NQ\_BITS1**

どの引数が指定されたかを定義する存在ビット。キーワードに関連付けられた引数を取得するには、コマンド・レベルのパラメーター構造から適切なアドレスを使用する必要があります。このアドレスを使用する前に、関連付けられた存在ビットを確認する必要があります。存在ビットがオフに設定されている場合は、引数が要求に指定されていないため、このアドレスを使用してはなりません。

**X'80'**

要求に RESOURCE キーワードの引数が含まれている場合に設定されます。設定された場合は、**NQ\_ADDR1** が意味を持ちます。

**X'40'**

要求に LENGTH キーワードの引数が含まれている場合に設定されます。設定された場合は、**NQ\_ADDR2** が意味を持ちます。

**X'20'**

要求に MAXLIFETIME キーワードの引数が含まれている場合に設定されます。設定された場合は、**NQ\_ADDR3** が意味を持ちます。

#### **NQ\_BITS2**

エンキュー・ドメインによって使用されない 2 バイト。

#### **NQ\_EIDOPT5**

エンキュー・ドメインによって使用されない 1 バイト。

#### **NQ\_EIDOPT6**

エンキュー・ドメインによって使用されない 1 バイト。

#### **NQ\_EIDOPT7**

エンキュー・ドメインによって使用されない 1 バイト。

#### **NQ\_EIDOPT8**

特定のキーワードが要求に指定されたかどうかを示します。

**X'04'**

NOSUSPEND が指定されています。

**X'02'**

DEQ が指定されています。

**X'01'**

ENQ が指定されています。

#### **NQ\_ADDR1**

RESOURCE からの値が含まれた領域のアドレスです。

#### **NQ\_ADDR2**

LENGTH のハーフワード値のアドレスです。

#### **NQ\_ADDR3**

MAXLIFETIME のフルワード値のアドレスです。

エンキュー・コマンドの場合の引数、キーワード、データ型、および入出力タイプの関係が [45 ページの表 2](#) に要約されています。

## コマンド・レベルのパラメーター構造のフィールドの変更

エンキュー・ドメインに渡されるフィールドは要求への入力として使用されます。入力フィールドを変更する場合の正しい方法は、そのフィールドの新規コピーを作成し、コマンド・レベル・パラメーター・リスト内のアドレスを変更して新規データを指すようにすることです。

### 重要:

1. コマンド・レベルのパラメーター・リストによって指定されたデータを変更することによって、入力フィールドを変更してはなりません。これを行うと、アプリケーション・プログラムに属しているストレージが破壊され、プログラムがフィールドを再使用しようとしたときに障害が発生することになります。
2. EXEC CICS ENQ 要求および DEQ 要求には出力フィールドはありません。

## EID の変更

EID を変更することで要求に対して大きな変更を行うことはできません。例えば、ENQ 要求を DEQ 要求に変更することはできません。ただし、LENGTH の存在ビットをオンにするなど、要求に対する小さな変更を行うことはできます。

以下のリストは、EID 内の変更可能なビットを示しています。EID のその他の部分を変更しようとしても、すべて無視されます。

### NQ\_BITS1

**X'40'**

LENGTH の存在ビット

**X'20'**

MAXLIFETIME の存在ビット。

### NQ\_EIDOPT7

XNQEREQ のユーザー出口プログラムは、ENQ コマンドについて、以下をオンまたはオフに設定できます。

**X'04'**

NOSUSPEND の存在ビット。

EID は、アプリケーション・プログラムに戻る前に、元の値にリセットされます。つまり、EID に対して行われた変更は、エンキュー要求の期間中のみ保持されます。

注: ユーザー出口プログラムは、EID に大きな変更を加えることはできません。ただし、許容されている小さい変更を加える場合でも、十分に注意する必要があります。

## タスク・トークン UEPTSTOK の使用

タスク・トークン UEPTSTOK は、同じタスク内の連続するエンキュー要求間で情報の受け渡しをするために使用できる 4 バイトの領域のアドレスを提供します。例えば、必要に応じてこのタスク・トークンを使用して、XNQEREQ 出口の連続呼び出し間で情報の受け渡しをすることができます。

これに対して、UEPNQTOK は、単一のエンキュー要求の期間中だけ使用可能です。これは、その内容が要求の終了時に破棄される可能性があるからです。

### 注:

1. UEPTSTOK によって指示される領域の存続期間が、タスクの存続期間です。
2. UEPTSTOK の値は、このタスクの存続期間中にこの値が渡されるすべての出口で共用されます。

表 2. ユーザー引数および関連のキーワード、データ型、および入出力タイプ			
引数	キーワード	データ型	入出力タイプ
Arg1	RESOURCE	DATA-AREA	入力
Arg2	LENGTH	BIN(15)	入力
Arg3	MAXLIFETIME	CVDA	入力

## ユーザー引数の変更

ユーザー出口プログラムは、ストレージを取得して設定するか、またはポインターを設定することによって、ユーザー入力引数を変更することができます。

ユーザー出口プログラムは、以下の方法でユーザー入力引数を変更することができます。

1. 変更済みの引数を保持するために十分なストレージを取得する。
2. ストレージに必要な値に設定する。
3. パラメーター・リスト内の関連ポインターを新規獲得した領域のアドレスに設定する。

注：

1. CICS は、引数値に対する変更を検査しません。そのため、変更はすべて、その変更を行うユーザー出口プログラムで検証する必要があります。
2. XNQEREQC で入力引数を変更することはお勧めできません。

## ユーザー引数の追加

グローバル・ユーザー出口プログラムで、MAXLIFETIME キーワードおよび LENGTH キーワードに関連付けられた引数を追加することができます。出口プログラムで指定または変更する引数が有効なことをユーザーが確認する必要があります。

MAXLIFETIME の有効な値は、DFHVALUE(TASK) および DFHVALUE(UOW) で、前者は 233、後者は 246 です。

追加する引数がまだ存在していないことを前提として、ユーザー出口プログラムで以下を実行する必要があります。

1. 追加する引数用のストレージを取得する。
2. そのストレージに必要な値に初期設定する。
3. パラメーター・リストから適切なポインターを選択してセットアップする。
4. EID 内で適切な引数存在ビットを選択してセットアップする。
5. 新しいリスト終了標識を反映するようにパラメーター・リストを変更する。

## ユーザー引数の削除

ユーザー出口プログラムで、LENGTH キーワードおよび MAXLIFETIME キーワードに関連付けられた引数を削除することができます (処理についての全責任はプログラムにあります)。

削除する引数が存在することを前提として、ユーザー出口プログラムで以下を実行する必要があります。

1. 該当の引数の存在ビットを EID 内で '0'b に切り替える。
2. 新しいリスト終了標識を反映するようにパラメーター・リストを変更する。

# イベント・キャプチャー出口 XEPCAP

XEPCAP 出口は、イベントが CICS イベント処理にキャプチャーされる直前に呼び出されます。イベントがいつキャプチャーされたかを検出するには、XEPCAP 出口を使用します。

出口固有のパラメーター

### UEPEPCX

EPCX (イベント・コンテキスト・データ構造) のアドレス。このパラメーターには、キャプチャー中のイベントに関する情報が含まれています。EPCX について詳しくは、[EPCX イベント処理コンテキスト・コンテナー](#)を参照してください。

### UEPEPTASK

タスク番号を含む 4 バイト (パック 10 進数) フィールドのアドレス。

### UEPLOAD

キャプチャー中のトランザクションのプログラム・ロード・ポイントのアドレス。

### UEPRSA

キャプチャー中のトランザクションのレジスター保管域のアドレス。このアドレスには、プログラムが EXEC CICS コマンドを発行した時点のレジスターの内容が含まれています。イベントが API

呼び出しではなくプログラム初期化によってキャプチャーされる場合、このパラメーターは 0 に設定されます。

## 戻りコード

### UERCNORM

処理が続行される。

### XPI 呼び出し

すべての XPI 呼び出しを使用できます。

### API コマンドおよび SPI コマンド

使用できる EXEC CICS コマンドはありません。

## EXEC インターフェース・プログラム出口 XEIIIN、XEIOUT、XEISPIN、および XEISPOUT

API または SPI 呼び出しの前または後に、EXEC インターフェース・プログラム内で 4 つのグローバル・ユーザー出口点を使用できます。

### XEIIIN

いずれかの EXEC CICS アプリケーション・プログラム・インターフェース (API) コマンド、またはシステム・プログラミング・インターフェース (SPI) コマンドの実行前に呼び出されます。

### XEISPIN

以下を除く、いずれかの EXEC CICS SPI コマンドの実行前に呼び出されます。

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

順序は以下のとおりです。

```
TRACE - XEIIIN - XEISPIN - EDF - command
```

### XEIOUT

いずれかの EXEC CICS API コマンドまたは SPI コマンドの実行後に呼び出されます。

### XEISPOUT

XEISPIN に対してリストされているものを除く、いずれかの EXEC CICS SPI コマンドの実行後に呼び出されます。

順序は以下のとおりです。

```
command - EDF - XEISPOUT - XEIOUT - TRACE
```

**注：**トランザクションが中断状態になると (例えば、ファイル入出力の待機)、これらの出口の非同期処理が行われる可能性があります。この状況は、CEDF のもとでも発生する場合があります。これは、CEDF がアプリケーションの XEISPIN 出口と XEISPOUT 出口の間で独自の EXEC CICS コマンドを発行するためです。

例えば、XEIIIN 出口と XEIOUT 出口の間で同じ GWA が共用される場合は、データの保全性を保証し、予測不能な結果を防ぐために、非同期処理が行われる可能性を考慮しておく必要があります。

出口に入る際に、出口固有パラメーター UEPARG には、コマンド・パラメーター・リストのアドレスが格納されます。



## コマンド・パラメーター・リスト

リスト内の最初のパラメーターは、引数 0 として認識されるデータ・ストリングを指します。その他のパラメーターは、コマンドで渡されるパラメーターに指定された値を指します。

引数 0 は、コマンドを識別する 2 バイトの機能コードで始まります。機能コードについては、資料 [EIB フィールド](#) および [EXEC インターフェース・ブロック \(EIB\)](#) の応答コードと機能コードに説明があります。機能コードの後に、「存在ビット」が含まれた 2 バイトのフィールドが続きます。存在ビットは、コマンドで引数が渡されるかどうかを示します。例えば、以下のコマンドの場合を考えてください。

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

ここで、引数 0 は機能コード X'0E02' (LINK) で始まっています。存在ビット 1 が設定され、引数 1 (つまり 'MYPROG') が存在することを示しています。

コマンド・パラメーター (PROGRAM など) とパラメーター・リスト内でのその位置および値 (この場合は引数 1、'MYPROG') との対応は、特定のコマンドの変換済みコードから推論できます。

### 重要:

引数 0 を変更して CICS コマンドを変更する処理はサポートされていないため、予期しないエラーや結果が生じることになります。

例えば、アプリケーション・プログラムがアセンブラーまたは PL/I で作成されている場合に、引数 0 を変更すると、プログラム・ストレージ (つまり、プログラム自体が占めているストレージ) に書き込むことになり、0C4 異常終了を引き起こす可能性があります。さらに、引数 0 を変更すると、アプリケーション・プログラム内でこのコマンド実行用の CICS コマンドを変更するだけでなく、アプリケーション・プログラムの仮想ストレージ・コピー内の CICS コマンドを変更することになります。これは、プログラムの同じコピーを呼び出す次のタスクでも変更されたコマンドを実行するということです。

引数 0 の変更の危険性を示すこの特殊例は、COBOL や C のアプリケーション・プログラムには適用されません。そうではあっても、サポートされるどの言語で作成されたアプリケーション・プログラムの場合も、CICS コマンドを変更してはなりません。

## コマンドのバイパス

XEIIN 出口プログラムまたは XEISPIN 出口プログラムは、UERCBYP 戻りコードを設定することによって、コマンドの実行をバイパスできます。これを行うと、EDF は起動されず、XEISPOUT、XEIOUT、および出口トレースがアクティブであれば呼び出されます。

EXEC CICS コマンドをバイパスすることによって、出口プログラムは、CICS 機能を出口プログラム独自の処理などで置き換えることができます。

UERCBYP を設定する前に、UEPPGM が指す値をプログラムで検査して、CICS が発行した EXEC CICS コマンドをバイパスするものではないことを確認する必要があります。

## 出口の XEIIN

出口の XEIIN は、いずれかの EXEC CICS API または SPI コマンドの実行の前に呼び出されます。

### 出口固有のパラメーター

#### UEPARG

EXEC コマンドのパラメーター・リストのアドレス。

#### UEPEXECB

システム EIB のアドレス。

#### UEPUSID

8 文字のユーザー ID のアドレス。

#### UEPPGM

8 文字のアプリケーション・プログラム名のアドレス。

#### UEUPLOAD

アプリケーション・プログラムのロード・ポイントのアドレス。



## **UEPRSA**

アプリケーションのレジスター 保管域のアドレス。このアドレスには、プログラムが EXEC CICS コマンドを発行した時点のレジスターの内容が含まれています。

## **UEP\_EI\_PBTOK**

z/OS ワークロード・マネージャー (WLM) の Performance Block Token を含む 4 バイトのフィールドのアドレス。出口プログラムは、このトークンを使用して、WLM Performance Block の情報 (サービス・クラス・トークン SERVCLS など) にアクセスすることができます。このアクセスを行うためには、出口プログラムは、Performance Block Token を MONTKN の入力パラメーターとして受け渡す WLM EXTRACT マクロの IWMMEXTR を使用する必要があります。IWMMEXTR マクロについて詳しくは、[z/OS MVS プログラミング: ワークロード管理サービス](#)を参照してください。

出口プログラムで、Performance Block の変更を試みないでください。そのような変更を試みた場合の結果は、予測不能です。

## 戻りコード

### **UERCNORM**

処理が続行される。

### **UERCBYR**

このコマンドの実行を迂回します。

### **UERCPRG**

タスクは XPI 呼び出し中にページされます。

## **XPI 呼び出し**

すべて使用できます。

## 出口 **XEISPIN**

出口 XEISPIN は、一部の SPI コマンドの実行前に呼び出されます。**ENABLE**、**DISABLE**、**EXTRACT EXIT**、**PERFORM DUMP**、および **RESYNC ENTRYNAME** の各コマンドに対しては、この出口は呼び出されません。

## 出口固有のパラメーター

### **UEPARG**

EXEC コマンドのパラメーター・リストのアドレス。

### **UEPEXECB**

システム EIB のアドレス。

### **UEPUSID**

8 文字のユーザー ID のアドレス。

### **UEPPGM**

8 文字のアプリケーション・プログラム名のアドレス。

### **UEPLOAD**

アプリケーション・プログラムのロード・ポイントのアドレス。

## **UEPRSA**

アプリケーション・プログラムのレジスター保管域のアドレス。この領域には、プログラムが EXEC CICS コマンドを発行した時点のレジスターの内容が含まれています。

## **UEP\_EI\_PBTOK**

z/OS ワークロード・マネージャー (WLM) の Performance Block Token を含む 4 バイトのフィールドのアドレス。出口プログラムは、このトークンを使用して、WLM Performance Block 内の情報 (例えばサービス・クラス・トークン SERVCLS) にアクセスすることができます。このアクセスを行うためには、出口プログラムは、Performance Block Token を MONTKN の入力パラメーターとして受け渡す WLM EXTRACT マクロの IWMMEXTR を使用する必要があります。IWMMEXTR マクロについて詳しくは、[z/OS MVS プログラミング: ワークロード管理サービス](#)を参照してください。

出口プログラムで、Performance Block の変更を試みないでください。そのような変更を試みた場合の結果は、予測不能です。

## 戻りコード

### **UERCNORM**

処理が続行される。

### **UERCBYD**

このコマンドの実行を迂回します。

### **UERPURG**

タスクは XPI 呼び出し中にページされます。

## **XPI 呼び出し**

すべて使用できます。

## 出口 XEIOU

出口 XEIOU は、任意の EXEC CICS API または SPI コマンドが実行された後で呼び出されます。

### 出口固有のパラメーター

#### **UEPARG**

EXEC コマンドのパラメーター・リストのアドレス。

#### **UEPEXECB**

システム EIB のアドレス。

#### **UEPUSID**

8 文字のユーザー ID のアドレス。

#### **UEPPGM**

8 文字のアプリケーション・プログラム名のアドレス。

#### **UEPLOAD**

アプリケーション・プログラムのロード・ポイントのアドレス。

#### **UEPRSA**

アプリケーションのレジスター 保管域のアドレス。このアドレスには、プログラムが EXEC CICS コマンドを発行した時点のレジスターの内容が含まれています。

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

z/OS ワークロード・マネージャー (WLM) の Performance Block Token を含む 4 バイトのフィールドのアドレス。出口プログラムは、このトークンを使用して、WLM Performance Block の情報 (サービス・クラス・トークン SERVCLS など) にアクセスすることができます。このアクセスを行うためには、出口プログラムは、Performance Block Token を MONTKN の入力パラメーターとして受け渡す WLM EXTRACT マクロの IWMMECTR を使用する必要があります。IWMMECTR マクロについて詳しくは、[z/OS MVS プログラミング: ワークロード管理サービスを参照してください](#)。

出口プログラムで、Performance Block の変更を試みないでください。そのような変更を試みた場合の結果は、予測不能です。

## 戻りコード

### **UERCNORM**

処理が続行される。

### **UERPURG**

タスクは XPI 呼び出し中にページされます。

## **XPI 呼び出し**

すべて使用できます。

## 出口 XEISPOUT

出口 XEISPOUT は、一部の SPI コマンドの実行後に呼び出されます。**ENABLE**、**DISABLE**、**EXTRACT**、**EXIT**、**PERFORM DUMP**、および **RESYNC ENTRYNAME** の各コマンドに対しては、この出口は呼び出されません。

### 出口固有のパラメーター

#### UEPARG

EXEC コマンドのパラメーター・リストのアドレス。

#### UEPEXECB

システム EIB のアドレス。

#### UEPUSID

8 文字のユーザー ID のアドレス。

#### UEPPGM

8 文字のアプリケーション・プログラム名のアドレス。

#### UEUPLOAD

アプリケーション・プログラムのロード・ポイントのアドレス。

#### UEPRSA

アプリケーションのレジスター 保管域のアドレス。このアドレスには、プログラムが EXEC CICS コマンドを発行した時点のレジスターの内容が含まれています。

#### UEP\_EI\_PBTOK

z/OS ワークロード・マネージャー (WLM) の Performance Block Token を含む 4 バイトのフィールドのアドレス。出口プログラムは、このトークンを使用して、WLM Performance Block の情報 (サービス・クラス・トークン SERVCLS など) にアクセスすることができます。このアクセスを行うためには、出口プログラムは、Performance Block Token を MONTKN の入力パラメーターとして受け渡す WLM EXTRACT マクロの IWMMEXTR を使用する必要があります。IWMMEXTR マクロについて詳しくは、[z/OS MVS プログラミング: ワークロード管理サービス](#)を参照してください。

出口プログラムで、Performance Block の変更を試みないでください。そのような変更を試みた場合の結果は、予測不能です。

### 戻りコード

#### UERCNORM

処理が続行される。

#### UERCPURG

タスクは XPI 呼び出し中にパージされます。

### XPI 呼び出し

すべて使用できます。

## フロントエンド・プログラミング・インターフェース 出口 XSZARQ および XSZBRQ

フロントエンド・プログラミング・インターフェース (FEPI) をインストールした場合は、グローバル・ユーザー出口 XSZARQ および XSZBRQ を FEPI コマンドの前と後に使用できます。

### XSZBRQ

FEPI コマンドが実行される前 (ただし、コマンドの構文が検証された後、したがって EDF 処理の後) に呼び出されます。

### XSZARQ

FEPI コマンドが完了した直後に (EDF 処理の前に) 呼び出されます。

FEPI アプリケーション・プログラミング・コマンドとシステム・プログラミング・コマンドは、両方とも XSZBRQ と XSZARQ を呼び出す原因になることに注意してください。ただし、後者は出口プログラムに意味のある情報を提供しません。

これらの出口から呼び出されるプログラムでは、出口プログラミング・インターフェース (XPI) 呼び出しも EXEC CICS コマンドも使用できません。これらの出口を使用して、FEPI コマンドおよび処理されるデータ

をモニターできます。コマンドを抑制したり、特定のコマンド・オプションを変更したりすることができます。これらの出口は以下の目的で使用できます。

- FEPI コマンドの発行のモニター
- ワークロード・ルーティング
- アプリケーション・プログラミング・コマンドに対する外部セキュリティ

## XSZBRQ

XSZBRQ は、FEPI コマンドが実行される前に呼び出されます。このコマンドの入力パラメーターが出口プログラムに渡されます。

渡される情報の大部分は読み取り専用ですが、特定のパラメーターを更新するプログラムを作成することができます。FEPI では、更新されたパラメーターの新しい値の妥当性は検査されません。さらに、出口プログラムで、要求を処理するかバイパスするかを決定することができます。例えば、XSZBRQ を使用して、コマンドをログに記録したり、インストール済み環境の規則に違反するコマンドをバイパスしたり、指定したターゲットまたはプールを変更することによってコマンドを転送したりすることができます。

合わせて、UEPSZALP および UEPSZALT には会話を開始するために必要な情報が含まれています。

### 呼び出される状況

FEPI コマンドの実行前、ただし構文検査およびセマンティック検査の後に、FEPI によって呼び出されます。

### 出口固有のパラメーター

#### UEPSZACT

コマンドを識別する 2 バイト・フィールド。値は [54 ページの表 3](#) に示されています。

#### UEPSZCNV

コマンドの会話 ID (CONVID) が含まれた 8 文字のフィールド。FEPI ALLOCATE、SEND、RECEIVE、CONVERSE、EXTRACT、ISSUE、START、および FREE の各コマンドに適用できます。

#### UEPSZALP

プール (POOL) の名前が含まれた 8 文字のフィールド。変更可能で、FEPI ALLOCATE コマンドおよび CONVERSE コマンドに適用できます。

#### UEPSZALT

ターゲット (TARGET) の名前が含まれた 8 文字のフィールド。変更可能で、FEPI ALLOCATE コマンドおよび CONVERSE コマンドに適用できます。

#### UEPSZTIM

タイムアウト値 (TIMEOUT) が含まれたフルワードの 2 進数フィールド。変更可能で、FEPI ALLOCATE、RECEIVE、CONVERSE、および START の各コマンドに適用できます。

#### UEPSZSND

「送信」データ域 (FROM) のアドレス。FEPI CONVERSE コマンドおよび SEND コマンドに適用できます。

#### UEPSZSNL

「送信」データの長さ (FROMLENGTH、FLENGTH) が含まれたフルワードの 2 進数フィールド。FEPI CONVERSE コマンドおよび SEND コマンドに適用できます。

#### UEPSZSTT

トランザクション ID (TRANSID) が含まれた 4 文字のフィールド。変更可能で、FEPI START コマンドに適用できます。

#### UEPSZSTM

端末 ID (TERMID) が含まれた 4 文字のフィールド。変更可能で、FEPI START コマンドに適用できます。

#### UEPSZSNK

データがキー・ストローク形式 (KEYSTROKE) かそうかを示す 1 ビットのフラグ・フィールド。FEPI CONVERSE FORMATTED コマンドおよび SEND FORMATTED コマンドに適用できます。次のような値をとります。

**UEPSZSNK\_OFF**

キー・ストローク形式ではありません。

**UEPSZSNK\_ON**

キー・ストローク形式です。

**UEPSZSNE**

キー・ストロークのエスケープ文字 (ESCAPE) が含まれた 1 文字のフィールド。FEPI CONVERSE FORMATTED コマンドおよび SEND FORMATTED コマンドに適用できます。

**戻りコード****UERCNORM**

処理が続行される。

**UERCBYR**

要求を処理しません。アプリケーションに INVREQ を返します。

注: 出口プログラムでイベント (CICS シャットダウンやタスク終了など) をバイパスすることはありません。

**XPI 呼び出し**

XPI 呼び出しを使用しないでください。

**XSZARQ**

XSZARQ は、FEPI コマンドの実行直後に呼び出されます。この出口プログラムに、コマンドの出力であるパラメーターが渡されます。渡される情報はすべて読み取り専用です。

**呼び出される状況**

FEPI コマンドの処理が完了した直後に FEPI によって呼び出されます。

**出口固有のパラメーター****UEPSZACN**

コマンドを識別する 2 バイト・フィールド。値は [54 ページの表 3](#) に示されています。

**UEPSZCON**

コマンドの会話 ID (CONVID) が含まれた 8 文字のフィールド。FEPI ALLOCATE、SEND、RECEIVE、CONVERSE、EXTRACT、ISSUE、START、および FREE の各コマンドに適用できます。

**UEPSZRP2**

コマンドの応答コード (RESP2) が含まれたフルワード。

**UEPSZRVD**

「受信」データ域 (INTO) のアドレス。FEPI RECEIVE、CONVERSE、および EXTRACT FIELD の各コマンドに適用できます。

**UEPSZRVL**

受信データの長さ (FLENGTH、TOFLENGTH) が含まれたフルワードのバイナリー・データ。FEPI RECEIVE、CONVERSE、および EXTRACT FIELD の各コマンドに適用できます。

**戻りコード****UERCNORM**

処理が続行される。

**XPI 呼び出し**

XPI 呼び出しを使用しないでください。

**出口固有のパラメーター UEPSZACT および UEPSZACN**

XSZBRQ と XSZARQ の両方に、処理中のコマンドまたはイベントを示すパラメーター (XSZBRQ の場合は **UEPSZACT**、XSZARQ の場合は **UEPSZACN**) が渡されます。

54 ページの表 3. は、UEPSZACT および UEPSZACN で渡される 16 進数値と、それらが表す FEPI コマンドとの関連付けを示しています。

表 3. 出口 XSZARQ の場合の UEPSZACT、および出口 XSZARQ の場合の UEPSZACT の設定

名前	設定 (16 進数)	FEPI コマンドまたはイベント
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	CICS 通常シャットダウン 1
UEPSZSDI	8404	CICS 即時シャットダウン 1
UEPSZSDF	8406	CICS 強制シャットダウン 1
UEPSZEOT	8408	CICS ダンプ終了 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
UEPSZOIP	8468	INSTALL POOL
UEPSZODP	8470	DISCARD POOL
UEPSZOQT	8482	INQUIRE TARGET
UEPSZOTT	8484	SET TARGET
UEPSZOIT	8488	INSTALL TARGETLIST
UEPSZODT	8490	DISCARD TARGETLIST

表 3. 出口 XSZARQ の場合の UEPSZACT、および出口 XSZARQ の場合の UEPSZACT の設定 (続き)		
名前	設定 (16 進数)	FEPI コマンドまたはイベント
UEPSZOQC	84A2	INQUIRE CONNECTION
UEPSZOTC	84A4	SET CONNECTION

注記:

- 1 これらのイベントは CICS 内部で生成されるもので、バイパスすることはできません。

## メッセージ出力を制御するための XMEOUT の使用

XMEOUT グローバル・ユーザー出口を CICS メッセージ・ドメインで使用して、FEPI メッセージを抑止または転送することができます。

ただし、メッセージを生成するエラー条件は一時データ・キュー・レコードも生成することに注意してください。この種のイベントを処理するには、メッセージの重複を招いてそれに対処するより、TD キュー経由でモニター・プログラムを使用する方が効率的です。

## ファイル制御ドメイン出口 XFCFRIN および XFCFROUT

XFCFRIN 出口は、メインファイル制御要求ゲート FCFR に入った時点で呼び出され、XFCFROUT 出口はファイル制御要求の完了後に実行されます。XFCFRIN 出口および XFCFROUT 出口は、スレッド・セーフ標準に沿ってコーディングする必要があり、スレッド・セーフ・リモート・ファイル・サポートを利用するためにスレッド・セーフであることを宣言する必要があります。

### XFCFRIN

XFCFRIN を使用すると、以下のタスクの 1 つ以上を実行するプログラムを作成することができます。

- ファイル制御要求をモニターし、それらの要求が引き続き CICS ファイル制御によって処理されるようにする。
- ファイル制御要求をインターセプトし、CICS ファイル制御処理も一緒にバイパスする。
- 要求をリモート領域にリダイレクトする。

出口プログラムが CICS ファイル制御に (リモート領域へのリダイレクトを選択せずに) 要求を渡す場合、パラメーターのいずれにも変更を加えることはできません。出口プログラムが要求をインターセプトしてファイル制御バイパスする場合は以下のようになります。

- ファイル制御によって返されるはずであった応答および出力パラメーターをすべて返す必要があります。これらは、出口固有のパラメーターの説明で、**出力**としてのマークで示されています。
- 要求が機能シッパされたものの場合、ミラー・トランザクションの終了が許可されるかどうかを示す必要があります。ある種のファイル制御要求の場合、同じトランザクション内で別の要求が事前に実行されていることが必要になります。(例えば、READNEXT の前には対応する STARTBR が必要であり、REWRITE の前には UPDATE を指定した対応する READ、READNEXT、または READPREV が必要です。このような 2 つの要求の間でミラー・トランザクションが終了すると、2 番目の要求は失敗する可能性が高くなります。逆に、不必要に保持されたミラー・トランザクションは、CICS リソースを保持することになり、ストレージやロックの問題の原因となる可能性があります。
- CICS は、同期点処理の一部として、ファイル参照および未処理の更新を終了します。ただし、同期点の場合は XFCFRIN 出口は呼び出されません。CICS 動作のこの側面を正確にエミュレートする場合、またはリカバリー可能リソースをサポートする場合は、同期点マネージャーをスケジュールするタスク関連ユーザー出口プログラムを呼び出す必要があります。CICS 同期点マネージャーによって開始されるプログラムのコーディングを参照してください。

要求をリモート領域にリダイレクトするには、出口プログラムで、SYSID パラメーターの値を追加または変更する必要があります。この場合、キー長およびレコード長の値も指定する必要があります。その他のパラメーターのいずれにも変更を加えることはできません。

### XFCFROUT

XFCFROUT を使用すると、完了したファイル制御要求の結果をモニターすることができます。例えば、CICS ファイル制御処理のバイパスを選択しなかった場合、(CICS 内部の) ファイル制御要求を分析し



て、そのタイプ、ファイル制御に渡されるパラメーター、および返される値を判別することができます。この出口は、以下の両方の場合に呼び出されます。

- CICS ファイル制御がその処理を完了した後 (正常の場合もエラーがある場合も含む)
- XFCFRIN 出口プログラムで CICS ファイル処理のバイパスを選択した場合

すべてのパラメーターは入力専用です。出口プログラムで値を変更することはできません。

機能シップのファイル制御要求で IPIC 接続を使用するには、XFCFRIN および XFCFROUT が、それをアドレスとして使用することを試行する前に UEPTERM パラメーターがゼロ以外の値であることを確認する必要があります。UEPTERM パラメーターは、IPIC 接続を介して機能シップされたファイル制御要求に対しては、ゼロになります。

## 出口の XFCFRIN

出口の XFCFRIN は、ファイル制御要求の実行前に呼び出されます。

この要求は、以下のものから生じた可能性があります。

- ユーザーのファイルを処理するためのアプリケーションの要求の実行
- 機能シップされた要求の受信
- システム・ファイルを処理するための CICS の内部要求

### 出口固有のパラメーター

#### UEPTRANID

4 バイトのトランザクション ID のアドレス。

#### UEPUSER

8 バイトのユーザー ID のアドレス。

#### UEPTERM

ゼロ、または 4 バイトの端末 ID のアドレス。戻されるアドレスがない場合は、この要求が IPIC 接続を介して機能シップされた可能性があります。

#### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス。

#### UEP\_FC\_FUNCTION

関数を含む 1 バイトのアドレス。FCFR 関数は、EXEC CICS インターフェースで使用可能な関数から派生し、いくつかの要求オプション (SET、INTO、UPDATE) は関数値に組み込まれています。例えば、UEP\_FC\_FUN\_DELETE は、RIDFLD オプションを指定した EXEC CICS DELETE から派生します。UEP\_FC\_FUN\_REWRITE\_DELETE は RIDFLD を指定しない EXEC CICS DELETE から派生します。可能な値は次のとおりです。

- 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

タスクの存続中に有効な 4 バイトのトークンのアドレス。 [タスク・トークン UEPTSTOK の使用](#)を参照してください。

#### UEP\_FC\_FILE\_NAME

ファイル名が含まれた 8 バイトの変更可能フィールドのアドレス。

#### UEP\_FC\_BUFFER\_P

要求の発信元によって提供されたバッファのアドレスが含まれたフルワードのアドレス。このアドレスの中に、INT0 オプションを持つ READ、READ NEXT、または READ PREV 要求の完了時に (出力) レコードが返されます。

#### UEP\_FC\_BUFFER\_L

(READ、READ NEXT、および READ PREV 要求のための) バッファ (その中にレコードが読み取られる) の LENGTH の値が含まれたフルワードのアドレス。

#### UEP\_FC\_RECORD\_P

次のいずれかのアドレスです。

- 要求が SET オプションを持つ READ、READ NEXT、または READ PREV の場合は、中にバッファ (このバッファの中にレコードが配置される) のアドレス (出力) が返されるフルワード。バッファ自体は、CICS ファイル制御、または出口プログラムがファイル制御を迂回する場合は、出口プログラムのいずれかによって提供されます。
- 要求が WRITE または REWRITE の場合は、書き込まれるレコードのアドレスが含まれたフルワードです。

#### UEP\_FC\_RECORD\_L

(READ、WRITE、REWRITE、READ NEXT、および READ PREV 要求のための) LENGTH の値が含まれるフルワードのアドレス。

すべての READ、READ NEXT、または READ PREV 要求の場合、これは出力フィールドです。このフィールド内に、読み取られたレコードの実際の長さが戻される際に配置されます。

**Warning:** INT0 を指定する要求の場合は、LENGTH の値を UEP\_FC\_BUFFER\_L フィールドで指定された値よりも大きい値に変更しないでください。この値を大きい値に変更すると、アプリケーション内でストレージ・オーバーレイが発生するおそれがあります。

WRITE または REWRITE の場合は、これはオプションのフィールドで、このフィールドが存在する場合、書き込まれるレコードの長さが含まれています。フィールドが指定されない場合は、このフルワードには 2 進数のゼロが含まれます。この場合に、リモート・ファイルのレコード長を変更するには、代わりに UEP\_FC\_M\_RECORD\_L を使用してください。

また、UEP\_FC\_SYSID パラメーターの説明も参照してください。

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

ファイルの (出力) の最大レコード長が含まれたフルワードのアドレス。(CICS の機能シップでは、この値を使用してリモートのファイル管理テーブル内のファイルのエントリを更新します。)

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

RIDFLD (レコード ID) の値のアドレスが含まれたフルワードのアドレス。レコード ID が入力または出力フィールドの場合の説明は、[73 ページの表 4](#) を参照してください。

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

KEYLENGTH のハーフワード値のアドレス。このアドレスは、レコード ID の長さ (場合によっては部分的な長さ) です。

KEYLENGTH は、READ、WRITE、DELETE、START BR、READ NEXT、READ PREV、および RESET BR 要求上のオプションの入力パラメーターです。フィールドが指定されない場合は、このハーフワードには 2 進数のゼロが含まれます。この場合に、リモート・ファイルのキー長を変更するには、代わりに UEP\_FC\_M\_RECORD\_ID\_L を使用してください。

また、UEP\_FC\_SYSID パラメーターの説明も参照してください。

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

レコード ID のフルサイズのハーフワード 値のアドレス。

レコード ID のフルサイズは、READ NEXT および READ PREV 要求上の必須の (出力) フィールドとして返されます。この値は、CICS の機能シップによって使用されます。

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

(READ、WRITE、DELETE、START BR、READ NEXT、READ PREV、および RESET BR 要求のための) RIDFLD タイプが含まれた 1 バイトのアドレス。出口への入力の場合、このパラメーターは以下の 1 つに対して設定されます。

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

VSAM KSDS または AIX® PATH のアクセス

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

RBA アクセスを介した VSAM ESDS または KSDS

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

VSAM RRDS アクセス

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

キーによる BDAM の非ブロック化 (READ、DELETE、START BR、および RESET BR 要求のみ)

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

相対レコードによる BDAM の非ブロック化 (READ、DELETE、START BR、および RESET BR 要求のみ)

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

VSAM 拡張 ESDS アクセス

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

(START BR、READ NEXT、READ PREV、RESET BR、および END BR 要求のための) REQID のハーフワード値のアドレス。

#### **UEP\_FC\_NUMREC**

NUMREC (出力) のフルワード値のアドレス。このアドレスには、(要求が RIDFLD を持つ DELETE の場合) 削除されたレコード数が返されます。

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

(READ、START BR、および RESET BR 要求のための) キー比較の設定が含まれた 1 バイトのアドレス。出口への入力の場合、このパラメーターは以下の 1 つに対して設定されます。

#### **UEP\_FC\_EQUAL**

「等しい」のキー比較が使用されます。

#### **UEP\_FC\_GTEQ**

「より大か等しい」のキー比較が使用されます。

#### **UEP\_FC\_GENERIC**

(READ、DELETE、START BR、および RESET BR 要求のための) 総称キーの設定が含まれた 1 バイトのアドレス。出口への入力の場合、このパラメーターは以下の 1 つに対して設定されます。

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

総称キーは、キー検索に使用されます。

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

完全キーは、キー検索に使用されます。

#### **UEP\_FC\_MASS\_INSERT**

(WRITE 要求のための) 大量挿入の設定が含まれた 1 バイトのアドレス。出口への入力の場合、このパラメーターは以下の 1 つに対して設定されます。

#### **UEP\_FC\_SEQUENTIAL\_WRITE**

レコードは、順次モードで書き込まれます。

#### **UEP\_FC\_DIRECT\_WRITE**

レコードは、直接モードで書き込まれます。

#### **UEP\_FC\_READ\_INTEGRITY**

(非更新 READ、READ NEXT、および READ PREV 要求のための) 読み取り保全性設定が含まれた 1 バイトのアドレス。(CICS の現行バージョンでは、この設定は VSAM RLS のみに当てはまります。) 出口への入力の場合、このパラメーターは以下の 1 つに対して設定されます。

#### **UEP\_FC\_CR**

整合した読み取り保全性が使用されます。

#### **UEP\_FC\_FCT\_VALUE**

読み取り保全性は、FILE 定義内の設定に従います。

#### **UEP\_FC\_NRI**

レコードは、読み取り保全性なしで読み取られます。

#### **UEP\_FC\_RR**

反復可能な読み取り保全性が使用されます。

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

TOKEN の値が含まれたフルワードのアドレス。

要求が更新のある READ、READ NEXT、または READ PREV で、アドレスがヌル以外の場合は、この領域はトークンが返される出力フィールドです。

要求が REWRITE、RIDFLD のない DELETE、または UNLOCK の場合は、この領域は入力フィールドです。

#### **UEP\_FC\_SYSID**

リモート領域を識別する SYSID が含まれる 4 バイトの領域のアドレス。出口への入力の場合、この領域には以下のいずれかが含まれます。

- API 呼び出しの SYSID オプションの値、または
- ブランク (SYSID が指定されなかった場合)。

要求を別の領域にリダイレクトするには、出口プログラムは、ターゲット領域の SYSID をこの出力域内に配置する必要があります。

このパラメーターが出口プログラムによって設定された場合は、要求は、ファイルの属性への参照なしでファイル制御によって機能シップされます。キー長が要求上に含まれていない場合は、出口プログラムは、UEP\_FC\_RECORD\_ID\_L パラメーターを設定してその値を設定する必要があります。

同様に、要求が WRITE または REWRITE で、レコード長が要求上に指定されていない場合は、出口プログラムは、UEP\_FC\_RECORD\_L パラメーターを設定してその値を設定する必要があります。

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

要求の完了後に返される長さのエラー・コードが含まれた 1 バイトの出力域のアドレス。可能な値は次のとおりです。

- 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**

要求が指定されたキーに対して複数のレコードを検出したかどうかを示す 1 バイトの出力域のアドレス。可能な値は次のとおりです。

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

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

4 バイトの出力域のアドレスで、UEP\_FC\_REASON\_ACCMETH\_REQUEST\_ERROR または UEP\_FC\_REASON\_IO\_ERROR のいずれかの応答が返された場合に、アクセス方式に依存した情報がこのアドレス内に返されます。

戻り値は、EIBRCODE の 2 から 5 バイトに配置されます。

## **UEP\_FC\_RESPONSE**

要求の完了後の以下の応答が含まれた 1 バイトの出力域のアドレス。

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

処理はエラーなしで完了しました。

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

処理はエラー状態で完了しました。この理由は UEP\_FC\_REASON 内に設定されています。

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

処理の完了を妨げるエラーが発生しました。このエラーは、通常 XPI 機能呼び出しからの DISASTER 応答、または UEPGAA か UEPTSTOK からアドレス指定されたデータの破損の結果として発生します。

この応答を設定した場合は、ファイル制御の呼び出し元は、First Failure Data Capture (初期障害データ収集) が実行されていると想定します。XPI 要求からの DISASTER 応答を回復機能委任する場合は、First Failure Data Capture が既に実行されています。一方、回復機能委任をしない場合、エラーを正常に診断するための十分な情報の収集を試みる必要があります。DFHDUDUX SYSTEM\_DUMP XPI 機能が、この目的のために適している場合があります。

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

出口プログラムが、無効なパラメーター・リストで呼び出されました。これは、CICS の内部論理エラーを示しています。アプリケーション・エラーを示す無効なパラメーター・リストは、EXCEPTION で応答する必要があることに注意してください。

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

XPI 機能呼び出しが PURGED 応答を受信しました。この応答を設定することは、パラメーター・リストへのすべての変更を受け付けることを除いて、UERCPURG 戻りコードを設定することと同じです。

## **UEP\_FC\_REASON**

要求が EXCEPTION 応答で完了した後の理由が含まれた 1 バイトの出力域のアドレス。想定される理由は、以下のとおりです。

- 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**

XFCFROUT に受け渡される 4 バイトのトークンのアドレス。これにより、例えば、作業域を出口の XFCFROUT に受け渡すことができます。

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

変更されたレコードの長さフィールドが含まれたフルワードのアドレス。これを使用して、LENGTH フィールドが API 呼び出し上に指定されていない場合に、リモート・ファイルのためにのみ LENGTH を変更できます。API 呼び出しで LENGTH フィールドが指定されている場合は、UEP\_FC\_M\_RECORD\_L によって格納されたアドレスの値を変更することは効果がないため、LENGTH を変更するために代わりに UEP\_FC\_RECORD\_L を使用してください。

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

変更されたキーの長さフィールドが含まれたハーフワードのアドレス。これを使用して、KEYLENGTH フィールドが API 呼び出し上に指定されていない場合に、リモート・ファイルのためにのみ KEYLENGTH を変更できます。KEYLENGTH フィールドが指定されている場合は、UEP\_FC\_M\_RECORD\_ID\_L によって格納されたアドレスの値を変更することは効果がないため、KEYLENGTH を変更するために代わりに UEP\_FC\_RECORD\_ID\_L を使用してください。

### 戻りコード

#### **UERCNORM**

処理が続行される。

#### **UERCBYPL**

この要求の CICS 処理を迂回します。出口プログラムが機能シップされた要求のために呼び出されている場合は、ミラー・トランザクションが終了することが許可されます。

#### **UERCBYPL**

この要求の CICS 処理を迂回します。出口プログラムが機能シップされた要求のために呼び出されている場合は、ミラー・トランザクションは終了してはいけません。

#### **UERCPURG**

タスクは XPI 呼び出し中にページされます。

### **XPI 呼び出し**

すべてを使用できます。

### **API および SPI 呼び出し**

何も使用できません。

## 出口 XFCFROUT

出口 XFCFROUT は、ファイル制御要求の完了後に呼び出されます。

### 出口固有のパラメーター

#### UEPTRANID

4 バイトのトランザクション ID のアドレス。

#### UEPUSER

8 バイトのユーザー ID のアドレス。

#### UEPTERM

ゼロ、または 4 バイトの端末 ID のアドレス。戻されるアドレスがない場合は、この要求が IPIC 接続を介して機能シップされた可能性があります。

#### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス。

#### UEP\_FC\_FUNCTION

関数を含む 1 バイトのアドレス。可能な値は次のとおりです。

- 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

タスクの存続中に有効な 4 バイトのトークンのアドレス。 [タスク・トークン UEPTSTOK の使用](#)を参照してください。

#### UEP\_FC\_FILE\_NAME

ファイル名が含まれた 8 バイト・フィールドのアドレス。

#### UEP\_FC\_BUFFER\_P

要求の発信元によって提供されたバッファのアドレスが含まれたフルワードのアドレス。このアドレスの中に、INT0 オプションを持つ READ、READ NEXT、または READ PREV 要求の完了時にレコードが返されます。

#### UEP\_FC\_BUFFER\_L

(READ、READ NEXT、および READ PREV 要求のための) レコードが読み込まれたバッファの LENGTH の値が含まれたフルワードのアドレス。

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

次のいずれかのアドレスです。

- 要求が、SET オプションを持つ READ、READ NEXT、または READ PREV 要求の場合、レコードが置かれたバッファのアドレスが返されるフルワード。
- 要求が WRITE または REWRITE の場合は、書き込まれたレコードのアドレスが含まれたフルワード。

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

(READ、WRITE、REWRITE、READ NEXT、および READ PREV 要求のための) LENGTH の値が含まれるフルワードのアドレス。

すべての READ、READ NEXT、または READ PREV 要求の場合、これは、読み取られたレコードの実際の長さが含まれた出力フィールドです。これらのタイプ要求の場合、EXEC CICS API 呼び出しに LENGTH オプションが指定されていない場合でも、このレコード長値は常に存在しています。

WRITE または REWRITE の場合、これはオプション・フィールドです。このフィールドが存在する場合、書き込まれたレコードの長さが含まれています。

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

ファイルの最大レコード長が含まれたフルワードのアドレス。

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

RIDFLD (レコード ID) の値のアドレスが含まれたフルワードのアドレス。レコード ID が入力または出力フィールドの場合の説明は、[73 ページの表 4](#) を参照してください。

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

KEYLENGTH のハーフワード値のアドレス。このアドレスは、レコード ID の長さ (場合によっては部分的な長さ) です。

レコード ID の長さは、READ、WRITE、DELETE、START BR、READ NEXT、READ PREV、および RESET BR 要求のオプションの入力パラメーターです。

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

レコード ID のフルサイズのハーフワード値のアドレス。(レコード ID のフルサイズは、EXEC CICS インターフェースの KEYLENGTH キーワードに対応します。

レコード ID のフルサイズは、READ NEXT および READ PREV 要求上の必須の出力フィールドとして返されます。

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

(READ、WRITE、DELETE、START BR、READ NEXT、READ PREV、および RESET BR 要求のための) RIDFLD タイプが含まれた 1 バイトのアドレス。

## **UEP\_FC\_KEY**

VSAM KSDS または AIX PATH のアクセス

## **UEP\_FC\_RBA**

RBA アクセスを介した VSAM ESDS または KSDS

## **UEP\_FC\_RRN**

VSAM RRDS アクセス

## **UEP\_FC\_DEBKEY**

キーによる BDAM の非ブロック化 (READ、DELETE、START BR、および RESET BR 要求のみ)

## **UEP\_FC\_DEBREC**

相対レコードによる BDAM の非ブロック化 (READ、DELETE、START BR、および RESET BR 要求のみ)

## **UEP\_FC\_XRBA**

VSAM 拡張 ESDS アクセス

## **UEP\_FC\_REQID**

(START BR、READ NEXT、READ PREV、RESET BR、および END BR 要求のための) REQID のハーフワード値のアドレス。

**UEP\_FC\_NUMREC**

NUMREC のフルワード値のアドレス。このアドレスには、(要求が RIDFLD を持つ DELETE の場合) 削除されたレコードの数が返されます。

**UEP\_FC\_KEY\_COMPARISON**

(READ、START BR、および RESET BR 要求のための) キー比較の設定が含まれた 1 バイトのアドレス。

**UEP\_FC\_EQUAL**

「等しい」のキー比較

**UEP\_FC\_GTEQ**

「より大か等しい」のキー比較

**UEP\_FC\_GENERIC**

(READ、DELETE、START BR、および RESET BR 要求のための) 総称キーの設定が含まれた 1 バイトのアドレス。

**UEP\_FC\_GENERIC\_KEY**

総称キーがキー検索に使用されます。

**UEP\_FC\_FULL\_KEY**

完全キーがキー検索に使用されます。

**UEP\_FC\_MASS\_INSERT**

(WRITE 要求のための) 大量挿入の設定が含まれた 1 バイトのアドレス。

**UEP\_FC\_SEQUENTIAL\_WRITE**

順次モード。

**UEP\_FC\_DIRECT\_WRITE**

直接モード。

**UEP\_FC\_READ\_INTEGRITY**

(非更新 READ、READ NEXT、および READ PREV 要求のための) 読み取り保全性設定が含まれた 1 バイトのアドレス。(CICS の現行バージョンでは、この設定は VSAM RLS のみに当てはまります。)

**UEP\_FC\_CR**

整合した読み取り保全性。

**UEP\_FC\_FCT\_VALUE**

FILE 定義の設定に従った読み取り保全性。

**UEP\_FC\_NRI**

読み取り保全性なし。

**UEP\_FC\_RR**

反復可能な読み取り保全性。

**UEP\_FC\_TOKEN**

TOKEN の値が含まれたフルワードのアドレス。

要求が更新のある READ、READ NEXT、または READ PREV で、アドレスがヌル以外の場合は、この領域はトークンが返される出力フィールドです。

要求が REWRITE、RIDFLD のない DELETE、または UNLOCK の場合は、この領域は入力フィールドです。

**UEP\_FC\_SYSID**

リモート領域を識別する SYSID が含まれた 4 バイト領域のアドレス。XFCFROUT 出口への入力の場合、この領域には以下のいずれかが含まれます。

- API 呼び出しの SYSID オプションの値、または
- ブランク (SYSID が指定されなかった場合)、または
- XFCFRIN 出口で指定された SYSID 値の値。

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

要求の完了後に返される長さエラー・コードが含まれた 1 バイト領域のアドレス。可能な値は次のとおりです。



- 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**

要求が指定されたキーに対して複数のレコードを検出したがどうかを示す 1 バイト領域のアドレス。可能な値は次のとおりです。

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

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

4 バイト領域のアドレスで、UEP\_FC\_REASON\_ACCMETH\_REQUEST\_ERROR または UEP\_FC\_REASON\_IO\_ERROR のいずれかの応答が返された場合に、アクセス方式に依存した情報がこのアドレス内に返されます。

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

要求の完了後の応答が含まれた 1 バイト領域のアドレス。

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

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

要求が EXCEPTION 応答で完了した場合、その理由が含まれた 1 バイト領域のアドレス。想定される理由は、以下のとおりです。

- 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\_CFDT\_CONNECT\_ERROR
- UEP\_FC\_REASON\_CFDT\_DISCONNECT\_ERROR
- UEP\_FC\_REASON\_CFDT\_INVALID\_CONTINUATION
- UEP\_FC\_REASON\_CFDT\_POOL\_FULL
- UEP\_FC\_REASON\_CFDT\_REOPEN\_ERROR
- UEP\_FC\_REASON\_CFDT\_SERVER\_NOT\_AVAILABLE
- UEP\_FC\_REASON\_CFDT\_SERVER\_NOT\_FOUND
- UEP\_FC\_REASON\_CFDT\_SYSIDERR
- UEP\_FC\_REASON\_CFDT\_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**

XFCFRIN から渡された 4 バイトのトークンのアドレス。

#### **戻りコード**

##### **UERCNORM**

処理が続行される。

##### **UERCPUrg**

タスクは XPI 呼び出し中にページされます。

#### **XPI 呼び出し**

すべてを使用できます。

#### **API および SPI 呼び出し**

何も使用できません。

## **ファイル制御 EXEC インターフェース API 出口 XFCREQ および XFCREQC**

XFCREQ 出口により、ファイル制御アプリケーション・プログラミング・インターフェース (API) 要求に対して何らかのアクションが実行される前に、その要求をインターセプトできます。また、XFCREQC 出口を使用して、ファイル制御機能が処理を完了した後に、ファイル制御 API 要求をインターセプトすることができます。

#### **重要**

機能シップされた要求の場合、XFCREQ 出口および XFCREQC 出口はターゲット領域では呼び出されません。つまり、ファイル制御 API 要求がリモート領域に機能シップされる場合、この出口はリモート領域で呼び出されません。機能シップされたファイル制御 API 要求をターゲット領域でインターセプトするに

は、XFCFRIN 出口を使用します。[55 ページの『ファイル制御ドメイン出口 XFCFRIN および XFCFROUT』](#)を参照してください。

インターセプトされるファイル制御 API コマンドは、以下のとおりです。

- READ
- WRITE
- REWRITE
- DELETE
- UNLOCK
- STARTBR
- READNEXT
- READPREV
- ENDBR
- RESETBR

XFCREQ 出口および XFCREQC 出口は、アセンブラー言語でのみ作成できます。

XFCREQ を使用して、以下のことができます。

- 要求を分析して、そのタイプ、指定するキーワード、および値を決定する。
- コマンドが実行される前に、要求で指定されている値を変更する。
- 戻りコードを設定して、次のいずれかを指定する。
  - CICS は (多くの場合は修正された) 要求を使用して処理を続行する。
  - CICS は要求を迂回する。(この戻りコードを設定すると、自分自身で要求を処理したかのように、EXEC インターフェース・ブロック (EIB) の戻りコードも設定する必要があることに注意してください。)

XFCREQC を使用して、以下のことができます。

- 要求を分析して、そのタイプ、指定するキーワード、および値を決定する。
- EIB の戻りコードを設定する。

両方の出口に、以下のように 9 つのパラメーターが渡されます。

- コマンド・レベルのパラメーター構造のアドレス
- XFCREQ から XFCREQC に 4 バイトのデータを渡すために使用されるトークン (UEPFCTOK) のアドレス。
- EIB からの 4 つの戻りコードおよびリソース情報のコピーのアドレス。
- タスクの存続中に有効な 4 バイトのトークン (UEPTSTOK) のアドレス。
- 再帰カウント・フィールドのアドレス。
- 要求が機能シッパされた場合に使用される、16 バイト領域のアドレス。

## コマンド・レベルのパラメーター構造

コマンド・レベルのパラメーター構造は、一連のアドレスで構成されます。最初のアドレスは、EXEC インターフェース記述子 (EID) を指します。これは、要求のタイプを記述するビット・ストリングからなり、要求で指定された各キーワードを識別します。残りのアドレスは、その要求に関連付けられたデータ部分を指します。例えば、2 番目のアドレスは常にファイル名を指します。

最初の 8 つのアドレスおよび最後のアドレスのみが、ユーザー出口で参照されます。9 番目から 11 番目のアドレスは、CICS の内部使用のために予約されています。

EID を調べて、要求のタイプおよび指定されたキーワードを判別することができます。リスト内のその他のパラメーターを調べて、キーワードの値を判別することができます。要求で指定されたキーワードの値を変更することもできます。(例えば、要求に関連するファイルの名前を変更できます。)

## パラメーター・リスト終了標識

パラメーター・リスト内の最後のアドレス・セットの高位ビットが、リスト内の最後のアドレスであることを示すために設定されます。ユーザー出口プログラムから戻ったときに、CICS は、パラメーター・リストをスキャンして高位ビットを探し、最終パラメーターを検出します。したがって、パラメーター・リストの長さを変更した場合は、高位ビットをリセットしてどれが新しい最終アドレスかを示す必要もあります。

XFCREQC の完了後に、XFCREQ が呼び出される前の元のパラメーター・リストが復元されます。その後に、実行診断機能 (EDF) により、実行前および実行後の元のコマンドが表示されます。

注: EDF は、出口によって行われた変更は表示しません。

## UEPCLPS 出口固有パラメーター

UEPCLPS 出口固有のパラメーターは、コマンド・レベルのパラメーター構造のアドレスで、XFCREQ および XFCREQC に含まれています。

コマンド・レベルのパラメーター構造には、FC\_ADDRO から FC\_ADDRB までの 12 のアドレスが含まれています。この構造は、DSECT FC\_ADDR\_LIST 内に定義されますが、それは COPY DFHFCEDS というステートメントを組み込むことによって、出口プログラムにコピーする必要があります。

コマンド・レベルのパラメーター・リストは、以下のように構成されています。

### FC\_ADDRO

EID という 9 バイトの領域のアドレスのことで、以下のように構成されています。

- **FC\_GROUP**
- **FC\_FUNC**
- **FC\_BITS1**
- **FC\_BITS2**
- **FC\_EIDOPT5**
- **FC\_EIDOPT6**
- **FC\_EIDOPT7**
- **FC\_EIDOPT8**

EID をマップする DSECT の名前は FC\_EID です。

### FC\_GROUP

常に X'06' で、これがファイル制御要求であることを示します。

### FC\_FUNC

以下の要求のタイプを定義する 1 バイトの領域です。

- 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**

値を含むどのキーワードが指定されたかを定義する存在ビット。キーワードに関連付けられた値を取得するには、コマンド・レベルのパラメーター構造から適切なアドレスを使用する必要があります。このアドレスを使用する前に、関連付けられた存在ビットを確認する必要があります。存在ビットがオフに設定されている場合は、キーワードが要求に指定されなかったため、このアドレスを使用してはなりません。

**X'80'**  
要求にキーワード FILE が含まれているかどうかを設定します。設定された場合は、**FC\_ADDR1** が意味を持ちます。

**X'40'**  
要求にキーワードの INTO、SET、または FROM のいずれかが含まれているかどうかを設定します。設定された場合は、**FC\_ADDR2** が意味を持ちます。

**X'20'**  
要求に LENGTH または NUMREC が指定されているかどうか、または STARTBR、RESETBR、ENDBR の各要求に REQID が指定されているかどうかを設定します。設定された場合は、**FC\_ADDR3** が意味を持ちます。

**X'10'**  
要求に RIDFLD が指定されているかどうかを設定します。設定された場合は、**FC\_ADDR4** が意味を持ちます。

**X'08'**  
要求に KEYLENGTH が指定されているかどうかを設定します。設定された場合は、**FC\_ADDR5** が意味を持ちます。

**X'04'**  
要求が READNEXT または READPREV で、REQID が指定されているかどうかを設定します。設定された場合は、**FC\_ADDR6** が意味を持ちます。

**X'02'**  
要求に SYSID が指定されているかどうかを設定します。設定された場合は、**FC\_ADDR7** が意味を持ちます。

**X'01'**  
ファイル制御では使用されません。

#### **FC\_BITS2**

存在ビットの 2 番目のセット。

**X'20'**  
要求に TOKEN が指定されているかどうかを設定します。設定された場合は、**FC\_ADDR8** が意味を持ちます。

#### **FC\_EIDOPT5**

値を取らない特定のキーワードが、要求上に指定されたかどうかを示します。

**X'04'**  
MASSINSERT が指定されています。

**X'02'**  
RRN が指定されています。

**X'01'**  
SET が指定されています (INTO の指定はなし)。

注：プログラムは、これらのキーワードが複数存在している可能性があるため、キーワードが存在するかどうかビット・レベルで検査する必要があります。

#### **FC\_EIDOPT6**

値を取らない特定のキーワードが、要求上に指定されたかどうかを示します。

**X'80'**

RBA が指定されています。

**X'40'**

GENERIC が指定されています。

**X'20'**

GTEQ が指定されています。

**X'10'**

UNCOMMITTED が指定されています。

**X'08'**

CONSISTENT が指定されています。

**X'04'**

REPEATABLE が指定されています。

**X'01'**

NOSUSPEND が (READ、READNEXT、READPREV、WRITE、DELETE、または REWRITE 上に) 指定されています。

注：

1. (UNCOMMITTED、CONSISTENT、および REPEATABLE に対する) 読み取り保全性ビットがコマンド上オフ (ゼロ) になっている場合は、ファイルのリソース定義上に指定された読み取り保全性オプションが使用されます。これらのオプションが何であるかを知る必要がある場合は、EXEC CICS INQUIRE FILE コマンドを発行することができます。
2. プログラムは、これらのキーワードが複数存在している可能性があるため、キーワードが存在するかどうかビット・レベルで検査する必要があります。

**FC\_EIDOPT7**

値を取らない特定のキーワードが、要求上に指定されたかどうかを示します。

**X'04'**

UPDATE が指定されています。この設定は READ 要求にのみ意味を持ちます。他の要求の場合は、X'04' が設定される場合も設定されない場合もあります。

**X'01'**

DEBREC または DEBKEY が指定されています (**FC\_EIDOPT8** を参照)。この設定は READ 要求にのみ意味を持ちます。他の要求の場合は、X'01' が設定される場合も設定されない場合もあります。

注：プログラムは、これらのキーワードが複数存在している可能性があるため、キーワードが存在するかどうかビット・レベルで検査する必要があります。

**FC\_EIDOPT8**

値を取らない特定のキーワードが、要求上に指定されたかどうかを示します。

**X'80'**

DEBKEY が指定されています。

**X'40'**

DEBREC が指定されています。

**X'20'**

TOKEN が指定されています。

**X'08'**

XRBA が指定されています。XRBA ビットがオンになっている場合は、FC\_RIDFLD (DSECT DFHFCEDS で説明) は、8 バイトの拡張相対バイト・アドレス (XRBA) を指します。

**FC\_ADDR1**

FILE キーワードに指定された名前が含まれている 8 バイトの領域のアドレスです。

**FC\_ADDR2**

次のアドレスのいずれかです。

- SET に返された 4 バイトのアドレス (要求が READ、READNEXT、または READPREV の場合で、**FC\_EIDOPT5** がキーワードが SET であることを示す場合)。
- INTO に返されたデータ (要求が READ、READNEXT、または READPREV の場合で **FC\_EIDOPT5** がキーワードが SET でないことを示す場合)。
- FROM からのデータ (要求が WRITE または REWRITE の場合)。

### **FC\_ADDR3**

次のアドレスのいずれかです。

- LENGTH のハーフワード値 (要求が READ、WRITE、REWRITE、READNEXT、または READPREV の場合)。

**警告:** INTO を指定する要求の場合は、LENGTH の値をアプリケーションで指定された値よりも大きい値に変更しないでください。この値を大きい値に変更すると、アプリケーション内でストレージ・オーバーレイが発生します。

- NUMREC の返されたハーフワード値 (要求が DELETE の場合)。
- REQID のハーフワード値 (要求が STARTBR、RESETBR、または ENDBR の場合)。

### **FC\_ADDR4**

RIDFLD キーワードの値が含まれている領域のアドレスです。

### **FC\_ADDR5**

KEYLENGTH のハーフワード値のアドレスです。

### **FC\_ADDR6**

REQID のハーフワード値のアドレスです (要求が READNEXT または READPREV の場合)。

### **FC\_ADDR7**

SYSID の値が含まれている領域のアドレスです。

### **FC\_ADDR8**

CICS の内部使用専用の値のアドレスです。このアドレスは使用できません。

### **FC\_ADDR9**

CICS の内部使用専用の値のアドレスです。このアドレスは使用できません。

### **FC\_ADDRA**

CICS の内部使用専用の値のアドレスです。このアドレスは使用できません。

### **FC\_ADDRB**

TOKEN のフルワード値のアドレスです (要求が READ、READNEXT、READPREV、REWRITE、DELETE、または UNLOCK の場合)。

## **コマンド・レベルのパラメーター構造のフィールドの変更**

ファイル制御に渡される一部のフィールドは要求への入力として使用され、一部は出力フィールドとして使用され、一部は入力と出力の両方に使用されます。ユーザー出口プログラムでフィールドを変更するために使用する方法は、フィールドの使用方法に応じて異なります。

### **入力フィールドおよび出力フィールドのリスト**

以下は、常に入力フィールドです。

- FILE
- FROM
- KEYLENGTH
- REQID
- SYSID

以下は、常に出力フィールドです。

- INTO
- NUMREC
- SET



LENGTH および RIDFLD が入力フィールドになるか出力フィールドになるかは、73 ページの表 4 に示すように、要求に応じて異なります。ダッシュ (–) は、そのキーワードを要求に指定できないことを意味します。

表 4. 入力フィールドおよび出力フィールドとしての LENGTH および RIDFLD		
Request	LENGTH	RIDFLD
READ	出力	注 73 ページの『1』を参照。
WRITE	入力	注 73 ページの『2』を参照。
REWRITE	入力	–
DELETE	–	注 73 ページの『3』を参照。
UNLOCK	–	–
STARTBR	–	入力
READNEXT	出力	出力
READPREV	出力	出力
ENDBR	–	–
RESETBR	–	入力

注：

1. 通常、これは入力フィールドです。ただし、UPDATE が指定されていて、ファイルが拡張キー検索を使用する BDAM ファイルである場合は、RIDFLD は入力と出力の両方に使用されます。
2. WRITE 要求での RIDFLD の用途はファイル・タイプに応じて異なります。VSAM KSDS または RRDS の場合、あるいは固定形式の BDAM ファイルの場合は、RIDFLD は入力フィールドです。その他のすべてのファイル・タイプの場合は、出力専用を使用されるか、入力および出力の両方に使用されるかのいずれかであり、出力フィールドと同様に処理する必要があります。
3. RIDFLD は、先行する READ UPDATE がない DELETE 要求では、入力フィールドです。先行する READ UPDATE がある要求では、このフィールドは指定されません。

### 入力フィールドの変更

入力フィールドを変更する場合の正しい方法は、そのフィールドの新規コピーを作成し、コマンド・レベル・パラメーター・リスト内のアドレスを変更して新規データを指すようにすることです。

注：コマンド・レベルのパラメーター・リストによって指定されたデータを変更することによって、入力フィールドを変更してはなりません。これを行うと、アプリケーション・プログラムに属しているストレージが破壊され、プログラムがフィールドを再使用しようとしたときに障害が発生することになります。

### 出力フィールドの変更

73 ページの『入力フィールドの変更』で説明されている手法は、出力フィールドの変更には適していません。(結果が、アプリケーションの領域ではなく新しい領域に返されるため、アプリケーションで表示できません。)

出力フィールドの変更は、コマンド・レベル・パラメーター・リストが指すデータを変更することによって行います。出力フィールドの場合は、アプリケーションのデータを所定の位置で変更することができます。これは、アプリケーションが、そのフィールドが何らかの方法で変更されることを予期しているためです。

## 入力と出力の両方に使用されるフィールドの変更

入力と出力の両方に使用されるフィールドの例として、INTO を指定する READ 要求の LENGTH があります。この種のフィールドは、出力フィールドと同様の方法で取り扱うことができ、同じものと見なされます。

## EID の変更

EID を変更することで要求に対して大きな変更を行うことはできません。例えば、WRITE 要求を READ 要求に変更することはできません。ただし、SYSID の存在ビットをオンにして要求をリモート・システムに送信できるようにするなど、要求に対する小さな変更を行うことはできます。

以下のリストは、EID 内の変更可能なビットを示しています。EID のその他の部分を変更しようとしても、すべて無視されます。

### FC\_BITS1

#### X'20'

LENGTH、NUMREC、または (要求が STARTBR、RESETBR、または ENDBR の場合は) REQID の存在ビット。

#### X'08'

KEYLENGTH の存在ビット。

#### X'04'

要求が READNEXT または READPREV の場合に、REQID の存在ビット。

#### X'02'

SYSID の存在ビット。

### FC\_BITS2

#### X'20'

Token が指定されています。

### FC\_EIDOPT5

#### X'04'

MASSINSERT が指定されています。

### FC\_EIDOPT6

#### X'40'

GENERIC が指定されています。

#### X'20'

GTEQ が指定されています。

#### X'10'

UNCOMMITTED が指定されています。

#### X'08'

CONSISTENT が指定されています。

#### X'04'

REPEATABLE が指定されています。

#### X'02'

READNEXT または READPREV で UPDATE が指定されています。

#### X'01'

NOSUSPEND が (READ、READNEXT、READPREV、WRITE、DELETE、または REWRITE 上に) 指定されています。

EID 内のビットは所定の位置で変更する必要があります。EID へのポインターを変更しないで下さい。そうしようとすると、CICS によって無視されます。

EID は、アプリケーション・プログラムに戻る前に、元の値にリセットされます。つまり、EID に対して行われた変更は、ファイル制御要求の期間中のみ保持されます。

複数の UNCOMMITTED、CONSISTENT、または REPEATABLE が指定されていると、CONSISTENT が UNCOMMITTED より優先され、REPEATABLE が CONSISTENT および UNCOMMITTED より優先されます。

## 読み取り保全性ビットの変更例

特定のファイルに対するすべてのプログラムからのすべての RLS 読み取り要求で、CONSISTENT 読み取りを指定したい場合があります。出口プログラムをコーディングする際に、そのファイルに対するすべての要求で CONSISTENT に対するビットをオンにし、他の 2 つの読み取り保全性ビットをオフにすることができます。FILE 定義で CONSISTENT を指定することによって、この効果を部分的に達成することができます。ただし、これは読み取り保全性のレベルを明示的に指定していない要求をオーバーライドするだけです。この目的でグローバル・ユーザー出口プログラムを使用することでも、UNCOMMITTED または REPEATABLE を明示指定するプログラムをオーバーライドすることができます。

### 警告：

1. グローバル・ユーザー出口プログラムがファイル要求を変更して高いレベルの読み取り保全性を要求した場合 (例えば、要求を UNCOMMITTED から REPEATABLE に変更した場合)、CICS が余分な読み取りロックを獲得したり、保持する読み取りロックの時間を長くしたりすることがあります。これにより、他のトランザクションの待機が発生したりデッドロックが生じたりして、システムのスループットが低下する場合があります。
2. グローバル・ユーザー出口プログラムが要求を変更して、低いレベルの読み取り保全性を要求するようになった場合 (例えば、要求を REPEATABLE から UNCOMMITTED に変更した場合)、要求を発信したプログラムでアプリケーション論理エラーが発生する可能性があります。これは、アプリケーション・プログラムが一連の他の関連レコードを読み取る間にレコードが未変更のままになっていることに依存している可能性があるためです。未変更状態は REPEATABLE では保証できますが、オプションが UNCOMMITTED に変更された場合には保証できません。
3. ユーザー出口プログラムは、EID に大きな変更を加えることはできません。ただし、許容されている小さい変更を加える場合でも、十分に注意する必要があります。例えば、DELETE を GENERIC DELETE に変更することは可能ですが、このような変更は危険な場合があります。

## パラメーター UEPFSHIP の使用

UEPFSHIP には、16 バイト領域のアドレスが格納されます。この領域は、4 文字と、それに続く 3 つのフルワードで構成されます。

最初のバイトに「Y」が含まれている場合、この要求はこの領域に機能シッパされたものです。この場合、出口プログラムで (戻りコード UERCBYP を設定することによって) ファイル制御をバイパスするには、3 つのフルワードを以下のように設定する必要があります。

### フルワード 1

バッファ領域の長さ

### フルワード 2

レコードの長さ

### フルワード 3

変更済みの RIDFLD の長さ

こうすることによって、データと RIDFLD が正しく返されます。

## EIB (EXEC インターフェース・ブロック)

EIBRSRCE、EIBRCODE、EIBRESP、および EIBRESP2 のコピーを出口に渡して、XFCREQ および XFCREQC 内の完了情報およびリソース情報の変更や設定を行ったり、XFCREQC 内の完了情報およびリソース情報を検査したりすることができます。

パラメーター・リストで指定されている EIBRSRCE、EIBRCODE、EIBRESP、および EIBRESP2 のコピーを更新できます。XFCREQC の完了後に、または XFCREQ に戻りコード「bypass」を指定してある場合に、ファイル制御が値を実 EIB にコピーします。

有効なファイル制御応答を設定する必要があります。EIBRCODE、EIBRESP、および EIBRESP2 の 3 つすべてを、整合性のある値セット (有効な完了を記述するためにファイル制御によって設定されるような値セット) に設定する必要があります。ファイル制御では、**EIBRCODE、EIBRESP、および EIBRESP2 の整合性は監視されません**。EIBRCODE、EIBRESP、および EIBRESP2 の値の設定に役立つように、ファイル制御で使用される値が DFHFCEDS に指定されます。

## XFCREQ および XFCREQC の使用方法の例

この例では、XFCREQ および XFCREQC を使用して、圧縮データが含まれたレコードを取得し、そのデータを解凍して、ユーザー・プログラムで INTO として指定された領域にそのデータを返します。この例では、出口の機能のみを示します。この例は、機能を実行するのに理想的な方法を示すものではありません。

**XFCREQ** では以下のようにします。

1. EXEC CICS GETMAIN を発行して、解凍済みデータを十分保持できる大きさの領域を確保します。
2. この新しい領域を指すように INTO ポインターを変更し、ファイル制御が要求を処理するときそのポインターを使用できるようにします。(解凍済みデータは、ユーザーの INTO 領域にコピーされます。すると、INTO ポインターは、アプリケーション・プログラムに戻る前にリセットされます。XFCREQC によって実行される処理のステージ [76 ページの『4』](#) および [76 ページの『7』](#) を参照してください。)
3. UEPFCTOK を新しい領域のアドレスに設定し、XFCREQC もこの領域を使用できるようにします。
4. CICS に戻ります。

**XFCREQC** では以下のようにします。

1. 「UEPRCODE」を調べて、ファイル制御要求がエラーなしで完了したことを確認します。
2. UEPFCTOK を使用して、領域のアドレスを見つけます。この領域には圧縮データが保持されています。
3. データを所定の位置に解凍します。
4. 新しい領域からユーザーの INTO 領域にデータをコピーします。ユーザー指定の LENGTH (コマンド・レベル・パラメーター・リストからの) を使用して、データが適合し、コピーが記憶保護違反を起こさないものであることを確認します。
5. データが適合しない場合は、UEPRES、UEPRES2、および UEPRCODE の「LENGERR」を設定します。
6. EXEC CICS FREEMAIN を使用して、UEPFCTOK が指す作業域を解放します。
7. この時点で、コマンド・レベル・パラメーター・リストはこのフリー域を INTO のアドレスとして指しています。これは問題ではありません。XFCREQC の完了後に、ユーザー・プログラムで指定された領域を指すように、ファイル制御がこのポインターを復元するからです。
8. CICS に戻ります。

## 出口 XFCREQ

出口 XFCREQ は、CICS がファイル制御 API 要求を処理する前に呼び出されます。機能シッパされた要求の場合、この出口はターゲット領域では呼び出されません。

### 出口固有のパラメーター

#### UEPCLPS

コマンド・レベルのパラメーター構造のアドレス。 [69 ページの『UEPCLPS 出口固有パラメーター』](#) を参照してください。

#### UEPFCTOK

XFCREQC に渡される 4 バイトのトークンのアドレス。これにより、例えば、作業域を出口の XFCREQC に渡すことができます。

#### UEPRCODE

EIB 戻りコード「EIBRCODE」の 6 バイトの 16 進数コピーのアドレス。EIB 戻りコードの詳細については、[EIB フィールド](#) を参照してください。

#### UEPRES

EIB 応答コード「EIBRESP」の 4 バイトのバイナリー・コピーのアドレス。

#### UEPRES2

EIB 応答コード「EIBRESP2」の 4 バイトのバイナリー・コピーのアドレス。

#### UEPTSTOK

タスクの存続中に有効な 4 バイトのトークンのアドレス。 [タスク・トークン UEPTSTOK の使用](#) を参照してください。

## UEPRECUR

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

## UEPFSHIP

16 バイト領域のアドレス。[75 ページの『パラメーター UEPFSHIP の使用』](#)を参照してください。

## UEPRSRCE

EIB リソース値の EIBRSRCE の 8 文字のコピーのアドレス。

## 戻りコード

### UERCNORM

処理が続行される。

### UERCBYP

ファイル制御 EXEC インターフェース・プログラムはこの要求を無視する必要があります。

### UERCPURG

タスクは XPI 呼び出し中にパージされます。

## XPI 呼び出し

すべてを使用できます。

この出口では、XPI GETMAIN 呼び出しおよび FREEMAIN 呼び出しの使用が許可されていますが、代わりに EXEC CICS GETMAIN コマンドおよび FREEMAIN コマンドを使用することをお勧めします。

## API および SPI 呼び出し

次のコマンドを除き、すべてを使用できます。

EXEC CICS SHUTDOWN

EXEC CICS XCTL

## 注：

1. 再帰的コマンドを発行する際には、ループが生じないように注意してください。例えば、XFCREQ 出口からファイル制御要求を発行する場合に、ループに入らないようにユーザーが注意する必要があります。再帰カウンター UEPRECUR を使用することをお勧めします。
2. EXEC CICS コマンドを発行する出口プログラムでは、最初に EIB をアドレス指定する必要があります。CICS サービスの使用を参照してください。
3. EXEC CICS コマンドを発行する出口プログラム、および DFHEIENT マクロを使用する出口プログラムでは、DFHEIRET マクロを使用して戻りコードを設定し、CICS に戻る必要があります。[ストレージの使用法を示すグローバル・ユーザー出口 XPI の例](#)を参照してください。

## 出口 XFCREQC

出口 XFCREQC は、ファイル制御 API 要求が完了した後で、ファイル制御 EXEC インターフェース・プログラムから戻る前に呼び出されます。機能シップされた要求の場合、この出口はターゲット領域では呼び出されません。

## 出口固有のパラメーター

### UEPCLPS

コマンド・レベルのパラメーター構造のアドレス。[69 ページの『UEPCLPS 出口固有パラメーター』](#)を参照してください。

### UEPFCTOK

XFCREQ から渡される 4 バイトのトークンのアドレス。

### UEPRCODE

EIB 戻りコード EIBRCODE の 6 バイトの 16 進数コピーのアドレス。EIB 戻りコードの詳細については、[EIB フィールド](#)を参照してください。

### UEPRES

EIB 応答コード EIBRESP の 4 バイトのバイナリー・コピーのアドレス。

注：アクセスされたばかりのファイルがリモートである場合、アドレス指定されたフィールドには (UEPRCODE がゼロ以外の場合でも) ゼロが入っています。

## UEPRES2

EIB 応答コード EIBRESP2 の 4 バイトのバイナリー・コピーのアドレス。

注: アクセスされたばかりのファイルがリモートである場合、アドレス指定されたフィールドには (UEPRCODE がゼロ以外の場合でも) ゼロが入っています。

## UEPTSTOK

タスクの存続中に有効な 4 バイトのトークンのアドレス。 タスク・トークン UEPTSTOK の使用を参照してください。

## UEPRECUR

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

## UEPRSRCE

EIB リソース値の EIBRSRCE の 8 文字のコピーのアドレス。

## 戻りコード

### UERCNORM

処理が続行される。

### UERCPUrg

タスクは XPI 呼び出し中にページされます。

## XPI 呼び出し

すべてを使用できます。

この出口では、XPI GETMAIN 呼び出しおよび FREEMAIN 呼び出しの使用が許可されていますが、代わりに EXEC CICS GETMAIN コマンドおよび FREEMAIN コマンドを使用することをお勧めします。

## API および SPI 呼び出し

次のコマンドを除き、すべてを使用できます。

EXEC CICS SHUTDOWN

EXEC CICS XCTL

## 注:

1. 再帰的コマンドを発行する際には、ループが生じないように注意してください。例えば、XFCREQC 出口からファイル制御要求を発行する場合に、ループに入らないようにユーザーが注意する必要があります。再帰カウンター UEPRECUR を使用することをお勧めします。
2. EXEC CICS コマンドを発行する出口プログラムでは、最初に EIB をアドレス指定する必要があります。CICS サービスの使用を参照してください。
3. EXEC CICS コマンドを発行する出口プログラム、および DFHEIENT マクロを使用する出口プログラムでは、DFHEIRET マクロを使用して戻りコードを設定し、CICS に戻る必要があります。ストレージの使用法を示すグローバル・ユーザー出口 XPI の例を参照してください。

## プログラム例

CICS では、EXEC インターフェース 出口に渡されるコマンド・レベルのパラメーター構造内のフィールドを変更する方法を示すプログラム例 DFH\$XTSE を CICSTS56.CICS.SDFHSAMP で提供しています。

## ファイル制御 EXEC インターフェース SPI 出口 XFCAREQ および XFCAREQC

XFCAREQ 出口により、ファイル制御システム・プログラミング・インターフェース (SPI) 要求に対して何らかのアクションが実行される前に、その要求をインターセプトできます。また、XFCAREQC を使用して、ファイル制御 SPI 要求が完了した後で応答をインターセプトすることができます。

インターセプトされるファイル制御 SPI 要求は、以下のとおりです。

- EXEC CICS INQUIRE FILE
- EXEC CICS SET FILE.

XFCAREQ を使用して、以下のことができます。

- SPI パラメーター・リスト (関数、キーワード、引数値、および応答) の分析
- 要求の実行前の入力パラメーターの変更
- 要求実行の回避および適切な応答の設定

XFCAREQC を使用して、以下のことができます。

- SPI パラメーター・リストの分析
- 実行後の出力パラメーター値の変更および応答の設定

また、以下のことも可能です。

- XFCAREQ 出口プログラムと XFCAREQC 出口プログラムが同じ要求のために呼び出された場合、それらのプログラム間でのデータの受け渡し。
- 複数のファイル制御出口プログラムが同じタスク内で呼び出された場合、それらのプログラム間でのデータの受け渡し。XFCAREQ および XFCAREQC の間の連続呼び出しの間、および他の EXEC 対応ユーザー出口の呼び出し間でもデータを渡すことができます。

ファイル状態に変更を加える場合 (つまり、ファイルのオープン、クローズ、有効化、または無効化を行う場合)、XFCAREQ でセットアップされた状況がファイル状態変更プログラム (XFCSREQ および XFCSREQC) 内の出口によって変更される可能性があります。したがって、出口を呼び出す順序を考慮する必要があります。4 つの出口がすべて使用可能になっている場合、呼び出しの順序は以下のようになります。

- **SET FILE** コマンドの場合:

1. XFCAREQ
2. XFCSREQ
3. XFCSREQC
4. XFCAREQC

- **INQUIRE FILE** コマンドの場合:

1. XFCAREQ
2. XFCAREQC

## 出口 XFCAREQ

この出口は、CICS がファイル制御 SPI 要求を処理する前に呼び出されます。

### 呼び出される状況

CICS がファイル制御 SPI 要求を処理する前。

### 出口固有のパラメーター

#### UEPCLPS

SPI コマンド・パラメーター・リストのコピーのアドレス。81 ページの『コマンド・レベルのパラメーター構造』を参照してください。

#### UEPFATOK

単一のファイル制御 SPI 要求での XFCAREQ と XFCAREQC の間の情報の受け渡しに使用できる 4 バイト領域のアドレス。

#### UEPRCODE

EIB 戻りコード EIBRCODE の 6 バイトの 16 進数コピーのアドレス。EIB 戻りコードの詳細については、[EIB フィールド](#)を参照してください。

#### UEPRES

EIB 応答コード EIBRESP の 4 バイトのバイナリー・コピーのアドレス。

#### UEPRES2

EIB 応答コード EIBRESP2 の 4 バイトのバイナリー・コピーのアドレス。



## **UEPTSTOK**

同じタスク内の連続するファイル制御要求間 (例えば、XFCAREQC 出口の連続呼び出し間) での情報の受け渡しに使用できる 4 バイトのトークンのアドレス。[タスク・トークン UEPTSTOK の使用](#)を参照してください。

## **UEPRECUR**

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際にゼロに設定され、再帰呼び出しごとに増加します。

## 戻りコード

### **UERCBYBYP**

この要求をバイパスします。

### **UERCNORM**

処理が続行される。

### **UERCPURG**

タスクは XPI 呼び出し中にページされます。

## **XPI コマンド**

すべて使用できます。

## **API コマンドおよび SPI コマンド**

次のコマンドを除き、すべてを使用できます。

EXEC CICS SHUTDOWN

EXEC CICS XCTL

注：再帰的コマンドを使用するときには注意してください。例えば、XFCAREQ 出口からファイル制御 SPI 要求を発行するときには、ループに入らないようにする必要があります。再帰カウンター UEPRECUR を使用することをお勧めします。

## 出口 XFCAREQC

出口 XFCAREQC は、ファイル制御 SPI 要求が完了した後で、ファイル制御 EXEC インターフェース・プログラムから戻る前に呼び出されます。

出口固有のパラメーターは以下のとおりです。

### **UEPCLPS**

API コマンド・パラメーター・リストのコピーのアドレス。[81 ページの『コマンド・レベルのパラメーター構造』](#)を参照してください。

### **UEPFATOK**

単一のファイル制御 SPI 要求での XFCAREQ と XFCAREQC の間の情報の受け渡しに使用できる 4 バイト領域のアドレス。

### **UEPRCODE**

EIB 戻りコード EIBRCODE の 6 バイトの 16 進数コピーのアドレス。EIB 戻りコードの詳細については、[EIB フィールド](#)を参照してください。

### **UEPRESB**

EIB 応答コード EIBRESP の 4 バイトのバイナリー・コピーのアドレス。

### **UEPRESB2**

EIB 応答コード EIBRESP2 の 4 バイトのバイナリー・コピーのアドレス。

### **UEPTSTOK**

同じタスク内の連続するファイル制御要求間 (例えば、XFCAREQC 出口の連続呼び出し間) での情報の受け渡しに使用できる 4 バイトのトークンのアドレス。[タスク・トークン UEPTSTOK の使用](#)を参照してください。

### **UEPRECUR**

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際にゼロに設定され、再帰呼び出しごとに増加します。



## 戻りコード

### UERCNORM

処理が続行される。

### UERCPURG

タスクは XPI 呼び出し中にページされます。

## XPI コマンド

すべて使用できます。

## API コマンドおよび SPI コマンド

次のコマンドを除き、すべてを使用できます。

EXEC CICS SHUTDOWN

EXEC CICS XCTL

パラメーター・リストに指定した EIBRCODE、EIBRESP、および EIBRESP2 のコピーを更新することができます。値を更新した場合、XFCAREQC の完了後に、または XFCAREQ に戻りコード UERCBYP を指定してある場合に、ファイル制御が新しい値をアプリケーション・プログラムの EXEC インターフェース・ブロック (EIB) にコピーします。

有効なファイル制御応答を設定する必要があります。EIBRCODE、EIBRESP、および EIBRESP2 の 3 つすべてを、整合性のある値セット (有効な完了を記述するためにファイル制御によって設定されるような値セット) に設定する必要があります。CICS では、ユーザーが設定した値の整合性は検査されません。

EIBRCODE がゼロ以外の値に設定され、EIBRESP がゼロに設定されると、CICS はゼロ以外の値で EIBRESP をオーバーライドします。EIBRCODE、EIBRESP、および EIBRESP2 の値の設定に役立つように、ファイル制御で使用する値が DSECT DFHFAUED に指定されます。

注: 再帰的コマンドを使用するときには注意してください。例えば、XFCAREQ 出口からファイル制御 SPI 要求を発行するときには、ループに入らないようにする必要があります。再帰カウンター UEPRECUR を使用することをお勧めします。

## コマンド・レベルのパラメーター構造

コマンド・レベルのパラメーター構造は、一連のアドレスで構成されます。最初のアドレスは、EXEC インターフェース記述子 (EID) を指します。これは、要求のタイプを記述するビット・ストリングからなり、要求で指定された各キーワードを識別します。残りのアドレスは、その要求に関連付けられたデータ部分を指します。

EID を調べて、要求のタイプおよび指定されたキーワードを判別することができます。リスト内のその他のパラメーターを調べて、キーワードの値を判別することができます。要求で指定されたキーワードの値を変更することもできます。

注: ファイル制御 SPI コマンドの引数、キーワード、データ型、および入出力タイプの関係が、次の表に要約されています。

- INQUIRE FILE については、[88 ページの表 5](#) を参照してください。
- SET FILE については、[90 ページの表 6](#) を参照してください。

### UEPCLPS 出口固有パラメーター

UEPCLPS 出口固有パラメーターは、XFCAREQ と XFCAREQC の両方に渡され、コマンド・レベルのパラメーター構造のアドレスを格納します。

コマンド・レベルのパラメーター・リストには、FCIS\_ADDR0 から FCIS\_ADDR63 までの 64 個のアドレスが含まれています。これらは DSECT DFHFAUED に記述されます。DSECT DFHFAUED は、COPY DFHFAUED ステートメントを組み込むことによって、プログラムにコピーする必要があります。

コマンド・レベルのパラメーター・リストは、以下のように構成されています。

### FCIS\_ADDR0

EID と呼ばれる 13 バイトの領域のアドレスであり、以下のように構成されています。

- FCIS\_GROUP
- FCIS\_FUNC

- **FCIS\_EIDOPT2**
- **FCIS\_EIDOPT3**
- **FCIS\_EIDOPT4**
- **FCIS\_BITS1**
- **FCIS\_BITS2**
- **FCIS\_BITS3**
- **FCIS\_BITS4**
- **FCIS\_BITS5**
- **FCIS\_BITS6**
- **FCIS\_BITS7**
- **FCIS\_BITS8**

#### **FCIS\_GROUP**

常に X'4C' で、これがファイル制御 SPI 要求であることを示します。

#### **FCIS\_FUNC**

以下の要求のタイプを定義する 1 バイトの領域です。

**X'02'**

INQUIRE FILE

**X'04'**

SET FILE

#### **FCIS\_EIDOPT2**

ファイル制御では使用されません。

#### **FCIS\_EIDOPT3**

ファイル制御では使用されません。

#### **FCIS\_EIDOPT4**

ファイル制御では使用されません。

#### **FCIS\_BITS1**

どの引数が指定されたかを示す存在ビット。キーワードに関連付けられた引数を取得するには、コマンド・レベルのパラメーター構造から適切なアドレスを取得する必要があります。このアドレスを使用する前に、関連付けられた存在ビットを確認する必要があります。存在ビットがオフに設定されている場合は、引数が要求に指定されていないため、このアドレスを使用してはなりません。

**X'80'**

要求に FILE キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR1 が意味を持ちます。

**X'40'**

要求に DSNAME キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR2 が意味を持ちます。

**X'20'**

要求に FWDRECSTATUS キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR3 が意味を持ちます。

**X'10'**

要求に STRINGS キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR4 が意味を持ちます。

**X'08'**

要求に BASEDSNAME キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR5 が意味を持ちます。

**X'04'**

要求に LSRPOOLID キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR6 が意味を持ちます。

**X'02'**

要求に READ キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR7 が意味を持ちます。

**X'01'**

要求に UPDATE キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR8 が意味を持ちます。

**FCIS\_BITS2**

どの引数が指定されたかを示す存在ビット。FCIS\_BITS1 のコメントは、FCIS\_BITS2 にも適用されます。

**X'80'**

要求に BROWSE キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR9 が意味を持ちます。

**X'40'**

要求に ADD キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR10 が意味を持ちます。

**X'20'**

要求に DELETE キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR11 が意味を持ちます。

**X'10'**

要求に DISPOSITION キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR12 が意味を持ちます。

**X'08'**

要求に EMPTYSTATUS キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR13 が意味を持ちます。

**X'04'**

要求に OPENSTATUS キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR14 が意味を持ちます。

**X'02'**

要求に ENABLESTATUS キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR15 が意味を持ちます。

**X'01'**

要求に RECOVSTATUS キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR16 が意味を持ちます。

**FCIS\_BITS3**

どの引数が指定されたかを示す存在ビット。FCIS\_BITS1 のコメントは、FCIS\_BITS3 にも適用されます。

**X'80'**

要求に ACCESSMETHOD キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR17 が意味を持ちます。

**X'40'**

要求に TYPE キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR18 が意味を持ちます。

**X'20'**

要求に OBJECT キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR19 が意味を持ちます。

**X'10'**

要求に REMOTESYSTEM キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR20 が意味を持ちます。

**X'08'**

要求に REMOTENAME キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR21 が意味を持ちます。

**X'04'**

要求に RECORDFORMAT キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR22 が意味を持ちます。

**X'02'**

要求に BLOCKFORMAT キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR23 が意味を持ちます。

**X'01'**

要求に KEYLENGTH キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR24 が意味を持ちます。

**FCIS\_BITS4**

どの引数が指定されたかを示す存在ビット。FCIS\_BITS1 のコメントは、FCIS\_BITS4 にも適用されます。

**X'80'**

要求に KEYPOSITION キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR25 が意味を持ちます。

**X'40'**

要求に RECORDSIZE キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR26 が意味を持ちます。

**X'20'**

要求に RELTYPE キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR27 が意味を持ちます。

**X'10'**

要求に EXCLUSIVE キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR28 が意味を持ちます。

**X'08'**

要求に BLOCKKEYLEN キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR29 が意味を持ちます。

**X'04'**

要求に BLOCKSIZE キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR30 が意味を持ちます。

**X'02'**

ファイル制御では使用されません。

**X'01'**

ファイル制御では使用されません。

**FCIS\_BITS5**

どの引数が指定されたかを示す存在ビット。FCIS\_BITS1 のコメントは、FCIS\_BITS5 にも適用されます。

**X'80'**

要求に TABLE キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR33 が意味を持ちます。

**X'40'**

要求に MAXNUMRECS キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR34 が意味を持ちます。

**X'20'**

要求に READINTEG キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR35 が意味を持ちます。

**X'10'**

要求に RLSACCESS キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR36 が意味を持ちます。

**X'08'**

要求に DEFINESOURCE キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR37 が意味を持ちます。

**X'04'**

要求に INSTALLAGT キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR38 が意味を持ちます。

**X'02'**

要求に INSTALLUSR キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR39 が意味を持ちます。

**X'01'**

要求に CHANGEAGENT キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR40 が意味を持ちます。

**FCIS\_BITS6**

特定のキーワードがファイル制御 SPI コマンドに指定されたかどうかを示します。

**X'80'**

要求に START キーワードが含まれている場合に設定されます。

**X'40'**

要求に NEXT キーワードが含まれている場合に設定されます。

**X'20'**

要求に END キーワードが含まれている場合に設定されます。

**X'10'**

要求に WAIT キーワードが含まれている場合に設定されます。

**X'08'**

要求に NOWAIT キーワードが含まれている場合に設定されます。

**X'04'**

要求に FORCE キーワードが含まれている場合に設定されます。

**X'02'**

要求に ENABLED キーワードが含まれている場合に設定されます。

**X'01'**

要求に DISABLED キーワードが含まれている場合に設定されます。

**FCIS\_BITS7**

特定のキーワードがファイル制御 SPI コマンドに指定されたかどうかを示します。JOURNALNUM の存在ビットも含まれています。

**X'80'**

要求に OPEN キーワードが含まれている場合に設定されます。

**X'40'**

要求に CLOSED キーワードが含まれている場合に設定されます。

**X'20'**

要求に EMPTY キーワードが含まれている場合に設定されます。

**X'10'**

要求に JOURNALNUM キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR52 が意味を持ちます。

**X'08'**

要求に LOADTYPE キーワードが含まれている場合に設定されます。

**X'04'**

要求に POOL キーワードが含まれている場合に設定されます。

**X'02'**

要求に TABLENAME キーワードが含まれている場合に設定されます。

**X'01'**

要求に UPDATEMODEL キーワードが含まれている場合に設定されます。

**FCIS\_BITS8****X'80'**

要求に REMOTETABLE キーワードが含まれている場合に設定されます。

**X'40'**

ファイル制御では使用されません。

**X'20'**

要求に CHANGEUSRID キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR59 が意味を持ちます。

**X'10'**

要求に CHANGEAGREL キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR60 が意味を持ちます。

**X'08'**

要求に DEFINETIME キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR61 が意味を持ちます。

**X'04'**

要求に CHANGETIME キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR62 が意味を持ちます。

**X'02'**

要求に INSTALLTIME キーワードの引数が含まれている場合に設定されます。設定された場合は、FCIS\_ADDR63 が意味を持ちます。

**X'01'**

ファイル制御では使用されません。

**FCIS\_ADDR1**

FILE からの名前が含まれた 8 バイト領域のアドレスです。

**FCIS\_ADDR2**

DSNAME からの名前が含まれた 44 バイト領域のアドレスです。

**FCIS\_ADDR3**

FWDRECOVSTATUS からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR4**

STRINGS からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR5**

BASEDSNAME からの名前が含まれた 44 バイト領域のアドレスです。

**FCIS\_ADDR6**

LSRPOOLID からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR7**

READ からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR8**

UPDATE からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR9**

BROWSE からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR10**

ADD からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR11**

DELETE からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR12**

DISPOSITION からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR13**

EMPTYSTATUS からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR14**

OPENSTATUS からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR15**

ENABLESTATUS からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR16**

RECOVSTATUS からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR17**

ACCESSMETHOD からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR18**

TYPE からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR19**

OBJECT からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR20**

REMOTESYSTEM からの名前が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR21**

REMOTENAME からの名前が含まれた 8 バイト領域のアドレスです。

**FCIS\_ADDR22**

RECORDFORMAT からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR23**

BLOCKFORMAT からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR24**

KEYLENGTH からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR25**

KEYPOSITION からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR26**

RECORDSIZE からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR27**

RELTYPE からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR28**

EXCLUSIVE からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR29**

BLOCKKEYLEN からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR30**

BLOCKSIZE からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR31**

ファイル制御では使用されません。

**FCIS\_ADDR32**

BUSY からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR33**

TABLE からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR34**

MAXNUMRECS からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR35**

READINTEG からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR36**

RLSACCESS からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR37**

DEFINESOURCE からのデータが含まれた 8 バイト領域のアドレスです。

**FCIS\_ADDR38**

INSTALLAGENT からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR39**

INSTALLUSRID からのデータが含まれた 8 バイト領域のアドレスです。

**FCIS\_ADDR40**

CHANGEAGENT からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR41 から FCIS\_ADDR51**

ファイル制御では使用されません。

**FCIS\_ADDR52**

JOURNALNUM からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR53**

LOADTYPE からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR54**

CFDTPPOOL からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR55**

TABlename からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR56**

UPdatEMODEL からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR57**

REMOtETABLE からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR58**

RBATYPE からの CVDA が含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR59**

CHANGeUSRID からのデータが含まれた 8 バイト領域のアドレスです。

**FCIS\_ADDR60**

CHANGeAGREL からのデータが含まれた 4 バイト領域のアドレスです。

**FCIS\_ADDR61**

DEFINETIME からのデータが含まれた 8 バイト領域のアドレスです。

**FCIS\_ADDR62**

CHANGETIME からのデータが含まれた 8 バイト領域のアドレスです。

**FCIS\_ADDR63**

INSTALLTIME からのデータが含まれた 8 バイト領域のアドレスです。

## コマンド・レベルのパラメーター構造のフィールドの変更

ファイル制御 SPI 要求に渡される一部のフィールドは要求への入力として使用され、一部のフィールドは要求の出力として使用されます。ユーザー出口プログラムでフィールドを変更するために使用する方法は、フィールドの使用方法に応じて異なります。

一般的な規則として、以下が当てはまります。

- INQUIRE FILE 要求では、FILE 以外のすべてのフィールドが出力フィールドです。
- SET FILE 要求では、すべてのフィールドが入力フィールドです。

INQUIRE FILE のパラメーターの詳細な説明については、88 ページの表 5 を参照してください。SET FILE のパラメーターの詳細な説明については、90 ページの表 6 を参照してください。

表 5. INQUIRE FILE 要求. 引数、キーワード、データ型、および入出力タイプの関係。			
引数	キーワード	データ型	入出力
Arg1	FILE	CHAR(8)	注を参照してください。
Arg2	DSNAME	CHAR(44)	出力
Arg3	FWDRECSTATUS	BIN(31)	出力
Arg4	STRINGS	BIN(31)	出力
Arg5	BASEDSNAME	CHAR(44)	出力
Arg6	LSRPOOLNUM	BIN(31)	出力
Arg7	READ	BIN(31)	出力



表 5. INQUIRE FILE 要求. 引数、キーワード、データ型、および入出力タイプの関係。(続き)

引数	キーワード	データ型	入出力
Arg8	UPDATE	BIN(31)	出力
Arg9	BROWSE	BIN(31)	出力
Arg10	ADD	BIN(31)	出力
Arg11	DELETE	BIN(31)	出力
Arg12	DISPOSITION	BIN(31)	出力
Arg13	EMPTYSTATUS	BIN(31)	出力
Arg14	OPENSTATUS	BIN(31)	出力
Arg15	ENABLESTATUS	BIN(31)	出力
Arg16	RECOVSTATUS	BIN(31)	出力
Arg17	ACCESSMETHOD	BIN(31)	出力
Arg18	TYPE	BIN(31)	出力
Arg19	OBJECT	BIN(31)	出力
Arg20	REMOTESYSTEM	CHAR(4)	出力
Arg21	REMOTENAME	CHAR(8)	出力
Arg22	RECORDFORMAT	BIN(31)	出力
Arg23	BLOCKFORMAT	BIN(31)	出力
Arg24	KEYLENGTH	BIN(31)	出力
Arg25	KEYPOSITION	BIN(31)	出力
Arg26	RECORDSIZE	BIN(31)	出力
Arg27	RELTYPE	BIN(31)	出力
Arg28	EXCLUSIVE	BIN(31)	出力
Arg29	BLOCKKEYLEN	BIN(31)	出力
Arg30	BLOCKSIZE	BIN(31)	出力
Arg31	*	*	*
Arg32	BUSY	BIN(31)	出力
Arg33	TABLE	BIN(31)	出力
Arg34	MAXNUMRECS	BIN(31)	出力
Arg35	READINTEG	BIN(31)	出力
Arg36	RLSACCESS	BIN(31)	出力
Arg37	DEFINESOURCE	CHAR(8)	出力
Arg38	INSTALLAGENT	BIN(31)	出力
Arg39	INSTALLUSRID	CHAR(8)	出力
Arg40	CHANGEAGENT	BIN(31)	出力
Arg41 から Arg51	*	*	*

表 5. INQUIRE FILE 要求. 引数、キーワード、データ型、および入出力タイプの関係。(続き)

引数	キーワード	データ型	入出力
Arg52	JOURNALNUM	BIN(15)	出力
Arg53	LOADTYPE	BIN(31)	出力
Arg54	CEDTPOOL	CHAR(8)	出力
Arg55	TABlename	CHAR(8)	出力
Arg56	UPDATEMODEL	BIN(31)	出力
Arg57	REMOTETABLE	BIN(31)	出力
Arg58	RBATYPE	BIN(31)	出力
Arg59	CHANGEUSRID	CHAR(8)	出力
Arg60	CHANGEAGREL	BIN(31)	出力
Arg61	DEFINETIME	CHAR(8)	出力
Arg62	CHANGETIME	CHAR(8)	出力
Arg63	INSTALLTIME	CHAR(8)	出力

注: INQUIRE FILE コマンドのファイル・パラメーターは以下のとおりです。

- 要求で START、NEXT、または END を指定していない場合は、入力フィールド
- 要求で NEXT を指定している場合は、出力フィールド
- 要求で START または END を指定している場合は、省略されます。

表 6. SET FILE 要求. 引数、キーワード、データ型、および入出力タイプの関係。

引数	キーワード	データ型	入出力
Arg1	FILE	CHAR(8)	入力
Arg2	DSNAME	CHAR(44)	入力
Arg3	FWDRECSTATUS	BIN(31)	入力
Arg4	STRINGS	BIN(31)	入力
Arg5	*	*	*
Arg6	LSRPOOLNUM	BIN(31)	入力
Arg7	READ	BIN(31)	入力
Arg8	UPDATE	BIN(31)	入力
Arg9	BROWSE	BIN(31)	入力
Arg10	ADD	BIN(31)	入力
Arg11	DELETE	BIN(31)	入力
Arg12	DISPOSITION	BIN(31)	入力
Arg13	EMPTYSTATUS	BIN(31)	入力
Arg14	OPENSTATUS	BIN(31)	入力
Arg15	ENABLESTATUS	BIN(31)	入力
Arg16	RECOVSTATUS	BIN(31)	入力

表 6. SET FILE 要求. 引数、キーワード、データ型、および入出力タイプの関係。(続き)

引数	キーワード	データ型	入出力
Arg17	*	*	*
Arg18	*	*	*
Arg19	*	*	*
Arg20	*	*	*
Arg21	*	*	*
Arg22	*	*	*
Arg23	*	*	*
Arg24	*	*	*
Arg25	*	*	*
Arg26	*	*	*
Arg27	*	*	*
Arg28	EXCLUSIVE	BIN(31)	入力
Arg29	*	*	*
Arg30	*	*	*
Arg31	*	*	*
Arg32	*	*	*
Arg33	TABLE	BIN(31)	入力
Arg34	MAXNUMRECS	BIN(31)	入力
Arg35	READINTEG	BIN(31)	入力
Arg36	RLSACCESS	BIN(31)	入力
Arg37	*	*	*
Arg38	*	*	*
Arg39	*	*	*
Arg40	*	*	*
Arg58	*	*	*
Arg59	*	*	*
Arg60	*	*	*
Arg61	*	*	*
Arg62	*	*	*
Arg63	*	*	*

**入力フィールドの変更**

入力フィールドを変更する場合の正しい方法は、そのフィールドの新規コピーを作成し、コマンド・レベル・パラメーター・リスト内のアドレスを変更して新規データを指すようにすることです。

コマンド・レベルのパラメーター・リストが指すデータを変更することによって入力フィールドを変更してはなりません。これを行うと、アプリケーション・プログラムに属しているストレージが破壊され、プログラムがフィールドを再使用しようとしたときに障害が発生することになります。

## 出力フィールドの変更

91 ページの『[入力フィールドの変更](#)』で説明されている手法は、出力フィールドの変更には適していません。(結果が、アプリケーションの領域ではなく新しい領域に返されるため、アプリケーションで表示できません。)

出力フィールドの変更は、コマンド・レベル・パラメーター・リストが指すデータを変更することによって行います。出力フィールドの場合は、アプリケーションのデータを所定の位置で変更することができます。これは、アプリケーションが、そのフィールドが何らかの方法で変更されることを予期しているためです。

## EID の変更

EID を変更することで要求に対して大きな変更を行うことはできません。例えば、INQUIRE FILE 要求を SET FILE 要求に変更することはできません。ただし、現行要求で指定されていない変数の存在ビットをオンにするなど、要求に対する小さな変更を行うことはできます。

以下の段落では、EID 内の変更可能なビットをリストしています。EID のその他の部分を変更しようとしても、すべて無視されます。

出口プログラムでは、以下の場合を除き、FCIS\_BITS1、FCIS\_BITS2、FCIS\_BITS3、FCIS\_BITS4、FCIS\_BITS5、FCIS\_BITS6、および FCIS\_BITS7 の任意のビットを変更することができます。

- FILE キーワードの存在ビット。
- START、NEXT、END、DEFINESOURCE、INSTALLAGENT、INSTALLUSRID、および CHANGEAGENT の各キーワードのビット。
- 「ファイル制御では使用されない」と説明されているビット。
- 実行中のコマンドに適用されないキーワードに対応するビット。例えば、CLOSED キーワードのビットは、SET FILE 要求では変更できますが、INQUIRE FILE 要求では変更できません。これは、CLOSED が意味を持つのは SET FILE 要求の場合のみだからです。[88 ページの表 5](#) および [90 ページの表 6](#) の説明を参照してください。

ユーザーのプログラムで、独自のコマンド・レベル・パラメーター構造および EID を提供することができます。その場合、UEPCLPS および TS\_ADDR0 を変更して、それぞれが新しい構造を指すようにする必要があります。

EID は、アプリケーション・プログラムに戻る前に、元の値にリセットされます。つまり、EID に対して行われた変更は、ファイル制御 SPI 要求の期間中のみ保持されます。

**注：**EID を変更する場合は、不整合なパラメーターを作成しないように注意する必要があります。例えば、元の要求で SET FILE OPEN が指定されていて、出口で CLOSED 用の EID ビットをオンにしたとすると、結果の SET FILE 要求には OPEN と CLOSED の両方が指定されます。この場合、コマンドの結果は予測不能です。

## ユーザー引数の変更

ユーザー出口プログラムがユーザー引数を変更できる方法は、引数が入力または出力のどちらであるかによって異なります。

- 入力引数の場合、出口プログラムは変更後の引数を保持するために十分なストレージを獲得し、必要な値をセットアップし、パラメーター・リスト内の関連ポインターを新しく獲得された領域のアドレスに設定する必要があります。
- 出力引数および入出力引数の場合、出口プログラムは所定の位置の引数を更新することができます。これは、ストレージの領域が CICS から値を受け取ると予期されているアプリケーション内の変数によって表されるためです。

### ユーザー引数の追加

出口プログラムは、EID 内の対応する存在ビットの変更が許されている場合に限り、ユーザー引数を追加することができます。

追加する引数がまだ存在していないことを前提として、出口プログラムで以下を実行する必要があります。

1. 追加する引数用のストレージを取得する。

2. そのストレージを必要な値に初期設定する。
3. パラメーター・リストから適切なポインターを選択してセットアップする。
4. Arg0 の適切な存在ビットを選択してセットアップする。
5. 必要に応じて、新しいリスト終了標識を反映するようにパラメーター・リストを変更する。

### ユーザー引数の削除

出口プログラムは、EID 内の対応する存在ビットの変更が許されている場合に限り、ユーザー引数を削除することができます。

削除する引数が存在することを前提として、出口プログラムで以下を実行する必要があります。

1. 対応する引数の存在ビットを Arg0 内でゼロに切り替える。
2. 新しいリスト終了標識を反映するようにパラメーター・リストを変更する。

## ファイル制御ファイル状態プログラム出口 XFCSREQ および XFCSREQC

ファイル制御状態プログラムでは、ファイル要求の前および後に呼び出すことができる 2 つのユーザー出口が提供されています。

### XFCSREQ

この出口は、ファイルの ENABLE、DISABLE、OPEN、CLOSE、または CANCEL CLOSE のいずれかの要求が実行される前に呼び出されます。XFCSREQ を使用して、ファイルの状態に関する情報 (例えば、どのファイル要求 (SERVREQ) が有効か、どのジャーナル処理オプションが設定されているかなど) を収集することができます。この情報に基づいて、必要に応じて要求を抑止することができます。詳しくは、[戻りコード UERCBYP](#) を参照してください。

### XFCSREQC

この出口は、ファイル要求が実行された後で呼び出されます。XFCSREQC を使用して、ファイルと関連付けられたデータ・セットに関する情報 (例えば、どのリカバリー・オプションが設定されているか) を収集することができます。ファイル要求を抑止するために XFCSREQ を使用した場合でも、XFCSREQC が呼び出されます。

ENABLE、DISABLE、OPEN、および CANCEL CLOSE の各要求の場合、各出口は一度だけ呼び出されます。しかし、CLOSE 要求の場合、実際のクローズの前にファイルが静止されることがあるため、出口は複数回呼び出される場合があります。クローズ要求時にユーザー出口 XFCSREQ および XFCSREQC が呼び出されないのは、以下の 2 つの場合です。

1. CICS の制御下の非即時シャットダウンで、CICS がすべてのファイルを閉じる場合。
2. ユーザー保守データ・テーブルをロードした後。データ・テーブルのロードが完了すると、ソース・データ・セットはそれ以上は必要ではなくなります。その後、CICS はファイルを閉じて割り振り解除し、データ・テーブルを開いたままにします。

### 単一の CLOSE 要求

単一の CLOSE 要求の場合、他のタスクがファイルにアクセスしているときにクローズが試行されると、XFCSREQ および XFCSREQC は複数回呼び出されます。例えば、このような状況での CLOSE NOWAIT コマンドの結果として、クローズが試行される前に XFCSREQ が呼び出されます。ファイルのユーザーがまだ存在しているので、クローズは遅延されます。しかし、NOWAIT が指定されているため、CLOSE 要求は完了し、UEPFSRSP を「UEFSPEND」に設定して (クローズが保留中であることを示し) XFCSREQC が呼び出されます。そのファイルに対するすべてのアクティビティが完了したら、ファイルが閉じられ、それを閉じたタスクの下で XFCSREQ および XFCSREQC が呼び出されます。

### CLOSE WAIT 要求

CLOSE WAIT 要求の場合、出口は、次のように呼び出されます。XFCSREQ が呼び出され、タスクがファイルのクローズを要求し、クローズが発生するのを待ちます。そのファイルに対するすべてのアクティビティが完了したら、ファイルが閉じられ、それを閉じたタスクの下で XFCSREQ および XFCSREQC が呼び出されます。これでクローズは完了しているので、最後に、CLOSE WAIT を発行したタスクが再開され、CLOSE 要求を完了して XFCSREQC を呼び出します。

## CANCEL CLOSE 要求

CANCEL CLOSE 要求は、保留中の QUIESCE コマンドを取り消す UNQUIESCE コマンドに応答して CICS によって発行されます。QUIESCE データ・セット・コマンドは、指定されたデータ・セットに対して開いているすべてのファイルをただちに使用不可に設定し、新しいタスクがそのデータ・セットにアクセスできないようにします。ただし、操作のクローズ部分は、ファイルが閉じられる前に最後のユーザー・タスクが終了するまで待機します。(これは、ファイルに対する他のクローズ操作の場合と同じです)。クローズがまだ待機中の間に UNQUIESCE が発行されると、CANCEL CLOSE 要求が出て、XFCSREQ および XFCSREQC の呼び出しが終了します。CANCEL CLOSE は、QUIESCE コマンドによって開始されたクローズ要求に対してのみ発行されることに注意してください。その他のクローズ要求に対しては発行されません。

## 出口 XFCSREQ

この出口は、ファイルの ENABLE、DISABLE、OPEN、CLOSE、または CANCEL CLOSE を試行する前に呼び出されます。

### 呼び出される状況

ファイルの ENABLE、DISABLE、OPEN、CLOSE、または CANCEL CLOSE を試行する前。

注：機能シップされた要求の場合、この出口はファイルがローカルになっているシステムで呼び出されます。

### 出口固有のパラメーター

#### UEPFSREQ

ファイル要求のタイプを示す 2 バイト・フィールドのアドレス。最初のバイトには、以下のいずれかの値が含まれています。

#### UEPFSOPN

オープン要求

#### UEPFSCLS

クローズ要求

#### UEPFSENB

有効化要求

#### UEPFSDIS

無効化要求

#### UEPFSCAN

ファイル・クローズ取り消し要求

最初のバイトがオープン要求 (UEPFSOPN) を示している場合、2 番目のバイトは以下のようにオープンのタイプを示します。

#### UEPFSNOP

通常のオープン

#### UEPFSOFB

バックアウトのためのオープン

最初のバイトがクローズ要求 (UEPFSCLS) を示している場合、2 番目のバイトは以下のようにクローズのタイプを示します。

#### UEPFSNC

通常のクローズ

#### UEPFSCP

クローズ保留

#### UEPFSELM

ロード・モード終了時クローズ

#### UEPFSIMM

即時クローズ

**UEPFSICP**

即時クローズ保留

**UEPFSQU**

RLS 静止クローズ

**UEPFILE**

8 バイトのファイル名のアドレス。

**UEPFINFO**

ファイルに関する情報が含まれたストレージ領域のアドレス。DSECT DFHUEFDS を使用してこの領域をマップすることができます。この領域には以下の情報が含まれています。

**UEFLNAME**

8 文字のファイル名。

**UEDSNAME**

ファイルに関連付けられたデータ・セットの 44 文字の DS 名 (ファイル要求が発行される前に設定されている場合)。

**UEFSERV**

このファイルの SERVREQ 設定を示す 1 バイト。可能な値は次のとおりです。

**UEFRDIM**

読み取り有効

**UEFUPDIM**

更新有効

**UEFADDIM**

追加有効

**UEFDELIM**

削除有効

**UEFBRZIM**

参照有効

**UEFDSJL**

このファイルに設定された自動ジャーナリング・オプションを示す 1 バイト。可能な値は次のとおりです。

**UEFJRO**

ジャーナル読み取り専用

**UEFJRU**

更新用ジャーナル読み取り

**UEFJWU**

ジャーナル書き込み更新

**UEFJWA**

ジャーナル書き込み追加

**UEFJDSN**

DS 名のジャーナル完了

**UEFJSYN**

ジャーナル同期読み取り

**UEFJASY**

ジャーナル非同期書き込み

**UEFDSVJL**

VSAM ファイルのみに適用される追加の自動ジャーナリング・オプションを示す 1 バイト。値は以下のとおりです。

**UEFJWAC**

書き込み追加完了

**UEFDSJID**

自動ジャーナリングに使用するジャーナルの番号がある場合、その番号が含まれた 1 バイト。

**UEFDSACC**

ファイルのアクセス方式を示す 1 バイト。可能な値は次のとおりです。

**UEFVSAM**

VSAM ファイル

**UEFBDAM**

BDAM ファイル

**UEFCFDT**

カップリング・ファシリティ・データ・テーブル

**UEFBCRV**

この出口の場合はヌルに設定されます。

**UEFFRLOG**

この出口の場合はヌルに設定されます。

**UEFFRCLG**

この出口の場合はブランクに設定されます。

**UEFCDATE**

この出口の場合はヌルに設定されます。

**UEFCTIME**

この出口の場合はヌルに設定されます。

**UEFBCAS**

この出口の場合はヌルに設定されます。

**UEFACBCP**

このフィールドはこの出口でヌルに設定されます。

注: UEPFINFO の最初の 7 つのフィールドのみがこの出口用に設定されます。残りのフィールドのうち、URFFRCLG はブランクに設定され、その他はヌルに設定されます。

**UEPRECUR**

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

**戻りコード****UERCNORM**

処理が続行される。

**UERCBYYP**

ファイル要求を抑止します。UERCBYYP は、以下の場合には使用できません。

- UEPFSREQ の 2 番目のバイトが以下のいずれかのクローズ・タイプを示している CLOSE 要求の抑止。
  - ロード・モード・クローズの終わり (UEPFSELM)
  - 即時クローズ (UEPFSIMM)
  - 即時クローズ保留 (UEPFSICP)
- バックアウト処理を実行するために開かれているファイルへの OPEN 要求の抑止 (この要求はバックアウト障害の原因となるため)。ファイルがバックアウト用に開かれる場合、UEPFSREQ の 2 番目のバイトは UEPFSOFB に設定されます。

有効な抑止の場合、CICS はメッセージ DFHFC0996 を出します。

```
Open/Close/Enable/Disable/Cancel of close of file
filename suppressed due to intervention of user exit
```

**UERCPURG**

タスクは XPI 呼び出し中にページされます。



## XPI 呼び出し

すべてを使用できます。

## API および SPI 呼び出し

EXEC CICS SHUTDOWN および EXEC CICS XCTL を除くすべてを使用できます。

注：

1. 再帰的コマンドを発行する際には、ループが生じないように注意してください。例えば、XFCSREQ 出口からファイル制御要求を発行する場合に、ループに入らないようにユーザーが注意する必要があります。再帰カウンタ UEPRECUR を使用することをお勧めします。
2. EXEC CICS コマンドを発行する出口プログラムでは、最初に EIB をアドレス指定する必要があります。[CICS サービスの使用](#)を参照してください。
3. EXEC CICS コマンドを発行する出口プログラム、および DFHEIENT マクロを使用する出口プログラムでは、DFHEIRET マクロを使用して戻りコードを設定し、CICS に戻る必要があります。[ストレージの使用法](#)を示すグローバル・ユーザー出口 XPI の例を参照してください。
4. ファイルの状態変更の結果としてこの出口が呼び出されるようなファイルがある場合、出口プログラムでそのようなファイルに対して EXEC CICS SET コマンドを呼び出すことができます。ただし、CICS システム内の他の並行アクティビティによっては、そのような SPI コマンドを使用してファイルの状態を操作する複数のタスク間でデッドロックが発生する可能性があります。

## 出口 XFCSREQ

この出口は、ファイルの ENABLE、DISABLE、OPEN、CLOSE、または CANCEL CLOSE のいずれかのコマンドの完了後に呼び出されます。

### 呼び出し時

ファイルの ENABLE、DISABLE、OPEN、CLOSE、または CANCEL CLOSE のいずれかのコマンドの完了後。

注：機能シップされた要求の場合、この出口はファイルがローカルになっているシステムで呼び出されます。

### 出口固有のパラメーター

#### UEPFSREQ

ファイル要求のタイプを示す 2 バイト・フィールドのアドレス。最初のバイトには、以下のいずれかの値が含まれています。

#### UEPFSOPN

オープン要求

#### UEPFSCLS

クローズ要求

#### UEPFSENB

有効化要求

#### UEPFSDIS

無効化要求

#### UEPFSCAN

ファイル・クローズ取り消し要求

最初のバイトがオープン要求 (UEPFSOPN) を示している場合、2 番目のバイトは以下のようにオープンのタイプを示します。

#### UEPFSNOP

通常のオープン

#### UEPFSOFB

バックアウトのためのオープン

最初のバイトがクローズ要求 (UEPFSCLS) を示している場合、2 番目のバイトは以下のようにクローズのタイプを示します。

**UEPFSNC**

通常のクローズ

**UEPFSCP**

クローズ保留

**UEPFSELM**

ロード・モード終了時クローズ

**UEPFSIMM**

即時クローズ

**UEPFSICP**

即時クローズ保留

**UEPFSQU**

RLS 静止クローズ

**UEPFILE**

8 バイトのファイル名のアドレス。

**UEPFINFO**

ファイルに関する情報が含まれたストレージ領域のアドレス。DSECT DFHUEFDS を使用してこの領域をマップすることができます。この領域には以下の情報が含まれています。

**UEFLNAME**

8 文字のファイル名。

**UEDSNAME**

ファイルに関連付けられたデータ・セットの 44 文字の DS 名。

**UEFSERV**

このファイルの SERVREQ 設定を示す 1 バイト。可能な値は次のとおりです。

**UEFRDIM**

読み取り有効

**UEFUPDIM**

更新有効

**UEFADDIM**

追加有効

**UEFDELIM**

削除有効

**UEFBRZIM**

参照有効

**UEFDSJL**

このファイルに設定された自動ジャーナリング・オプションを示す 1 バイト。可能な値は次のとおりです。

**UEFJRO**

ジャーナル読み取り専用

**UEFJRU**

更新用ジャーナル読み取り

**UEFJWU**

ジャーナル書き込み更新

**UEFJWA**

ジャーナル書き込み追加

**UEFJDSN**

DS 名のジャーナル完了

**UEFJSYN**

ジャーナル同期読み取り

**UEFJASY**

ジャーナル非同期書き込み

**UEFDSVJL**

VSAM ファイルのみに適用される追加の自動ジャーナリング・オプションを示す 1 バイト。値は以下のとおりです。

**UEFJWAC**

書き込み追加完了

**UEFDSJID**

自動ジャーナリングに使用するジャーナルの番号がある場合、その番号が含まれた 1 バイト。

**UEFDSACC**

ファイルのアクセス方式を示す 1 バイト。可能な値は次のとおりです。

**UEFVSAM**

VSAM ファイル

**UEFBDAM**

BDAM ファイル

**UEFCFDT**

カップリング・ファシリティ・データ・テーブル

**UEFBCRV**

このファイルに関連付けられたデータ・セットのリカバリー属性を示す 1 バイト。可能な値は次のとおりです。

**UEFBCFR**

順方向リカバリーが指定されています

**UEFBCLOG**

ロギングが指定されています

**UEFBCVAL**

リカバリー属性が有効であることを示すフラグ。

**UEFFRLOG**

CICS ファイル・リソース定義のリカバリー属性から取得される 1 から 99 までの範囲の順方向リカバリー・ログ ID が含まれた 1 バイト・フィールド。この番号は、DFHJnn の形式 (nn は順方向リカバリー・ログ番号) の CICS 内部ジャーナル名に対応します。CICS は、このジャーナル名を順方向リカバリー・ログ・ストリームにマップします。

ファイルに順方向リカバリー・ロギングが指定されていない場合、または順方向リカバリー・ログ・ストリーム名が ICF カタログから取得された場合は、フィールドはゼロに設定されます。

**UEFFRCLG**

ICF カタログから取得された順方向リカバリー・ログ・ストリームの名前が含まれた 26 バイト・フィールド。これは順方向リカバリーに使用されます。ICF カタログに指定されていない場合、または順方向リカバリーがファイルに使用されていない場合は、ブランクに設定されます。

**UEFCDATE**

パック 10 進数形式の日付 (YYYYDDD+)。このフィールドは、ファイルに関連付けられている VSAM 範囲に対して閉じる最後のファイルである場合にのみ設定されます。ここには、VSAM 範囲に対するアクティビティーが終了 (静止) した日付が含まれています。

**UEFCTIME**

パック 10 進数形式の時刻 (HHMMSS+)。このフィールドは、ファイルに関連付けられている VSAM 範囲に対して閉じる最後のファイルである場合にのみ設定されます。ここには、VSAM 範囲に対するアクティビティーが終了 (静止) した時刻が含まれています。

**UEFBCAS**

このデータ・セットの使用可能性を示すフラグ・バイト。設定された場合、値は以下のとおりです。

**UEPFBCAS**

データ・セットには使用不可のマークが付いています。

## **UEFACBCP**

VSAM ファイルの場合は ACB の、BDAM ファイルの場合は DCB の読み取り専用コピーのアドレス。正常オープンの完了後にのみ設定されます。

## **UEPFSRSP**

要求の戻りコードが含まれたバイトのアドレス。以下のいずれかの値を取ります。

### **UEFSNORM**

通常応答。

### **UEFSWARN**

警告応答。

### **UEFSFAIL**

障害応答。

### **UEFSPEND**

保留応答。「保留」応答は、CLOSE 要求の後でのみ返されることがあります。これは、CLOSE 要求の結果として、ファイルのクローズが保留されていて、ファイルが静止されていることを示します。ファイルに対するすべてのアクティビティーが完了すると、ファイルは閉じられます。これを有効にすると、XFCSREQ 出口および XFCSREQC 出口が再び呼び出され、そのときに実際のクローズが実行されることに注意してください。

## **UEPRECUR**

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとに増加します。

### **注：**

1. UEPFINFO の最初の 7 つのフィールド (UEFLNAME から UEFDSACC まで) は、すべての要求 (つまり OPEN、CLOSE、ENABLE、または DISABLE の各要求) の場合に設定されます。
2. 次の 3 つのフィールド (UEFBCRV、UEFFRLOG、および UEFFRCLG) は、OPEN 要求が成功した後でのみ有効です。
3. UEFCDATE から UEFCBCAS までのフィールドは、CLOSE 要求が成功した後でのみ設定されます。他のすべての要求の後では、ファイルが既に閉じていたり、クローズが失敗したり、クローズが保留になったりしていた場合、これらのフィールドがヌルに設定されます。
4. ファイルの状態変更の結果としてこの出口が呼び出されるようなファイルがある場合、出口プログラムでそのようなファイルに対して EXEC CICS SET コマンドを呼び出すことができます。ただし、CICS システム内の他の並行アクティビティーによっては、そのような SPI コマンドを使用してファイルの状態を操作する複数のタスク間でデッドロックが発生する可能性があります。

## **戻りコード**

### **UERCNORM**

処理が続行される。

### **UERCPUrg**

タスクは XPI 呼び出し中にページされます。

## **XPI 呼び出し**

すべてを使用できます。

## **API および SPI 呼び出し**

EXEC CICS SHUTDOWN および EXEC CICS XCTL を除くすべてを使用できます。

### **注：**

1. 再帰的コマンドを発行する際には、ループが生じないように注意してください。例えば、XFCSREQC 出口からファイル制御要求を発行する場合に、ループに入らないようにユーザーが注意する必要があります。再帰カウンター UEPRECUR を使用することをお勧めします。
2. EXEC CICS コマンドを発行する出口プログラムでは、最初に EIB をアドレス指定する必要があります。CICS サービスの使用を参照してください。
3. EXEC CICS コマンドを発行する出口プログラム、および DFHEIENT マクロを使用する出口プログラムでは、DFHEIRET マクロを使用して戻りコードを設定し、CICS に戻る必要があります。[ストレージの使用法を示すグローバル・ユーザー出口 XPI の例](#)を参照してください。

## XFCSREQC 用のグローバル・ユーザー出口のサンプル・プログラム

DFH\$REQC では、ファイル制御状態プログラム・グローバル・ユーザー出口 XFCSREQC の処理例を提供しています。詳細については、[ファイル制御状態サンプル出口プログラム: DFH\\$REQC](#) を参照してください。

## ファイル制御オープン/クローズ・プログラム出口 XFCNREC

XFCNREC を使用して、非 RLS データ・セットのオープン失敗を抑止することができます。

RLS データ・セットの場合、リカバリーはデータ・セットのプロパティです。そのため、ファイルとその基本データ・セットが一致しないリカバリー属性を持つことはあり得ません。XFCNREC 出口プログラムの作成についての詳細は、[CICS 管理リソースのリカバリーの構成](#)を参照してください。

### 出口 XFCNREC

#### 呼び出される状況

ファイルのオープン前に、以下のような不一致が検出された場合

1. ファイルのバックアウト・リカバリー設定とそれに関連付けられている非 RLS データ・セットの間の不一致。
2. BWO が必須であるにもかかわらず、リカバリー属性が、関連付けられた順方向リカバリーファイルが指定されていないことを示している。

#### 出口固有のパラメーター

##### UEFILE

8 バイトのファイル名のアドレス。ファイル名の長さが 8 文字未満の場合は、空白で埋め込まれます。

##### UEDSETN

44 バイトの基本データ・セット名のアドレス。データ・セット名の長さが 44 文字未満の場合は、空白で埋め込まれます。

##### UEPFRCV

FILE 定義で指定された、ファイルのバックアウト・リカバリー設定が含まれた 1 バイト・フィールドのアドレス。以下の値を指定できます。

##### UEPFLOG

バックアウト・ロギングが指定されています。

RECOV(NONE) が FILE 定義で指定されている場合、アドレス指定されたフィールドには 16 進数ゼロが含まれています。

BWO 不一致のために出口が呼び出された場合には、このフィールドは意味を持ちません。

##### UEPFAIL

不一致の理由が含まれた 1 バイト・フィールドのアドレス。可能な値は次のとおりです。

##### UEPBWOF

BWO の不一致

##### UEPATTF

バックアウト・リカバリー設定の不一致

##### UEPOPEN

1 バイト・フィールドのアドレス。デフォルト値は N です。BWO の不一致によるオープン失敗をバイパスするには、アドレス指定されたフィールドを Y に設定します。

#### 戻りコード

##### UERCNORM

ファイルは正常に開かれました。

##### UERCBYP

オープン失敗をバイパス。不一致を受け入れます。

#### **XPI 呼び出し**

これは使用しないでください。

#### **SPI 呼び出し**

これは使用しないでください。

#### **API および SPI 呼び出し**

これは使用しないでください。

### **一致しないバックアウト・リカバリー設定に対する XFCNREC 出口**

同じ基本データ・セットに関連付けられた異なるファイルのバックアウト・リカバリー設定が整合していない場合でも、オープン処理を継続する必要がある場合は、XFCNREC グローバル・ユーザー出口を使用します。

オープン障害が抑制された後、CICS はデータ・セットの保全性を保証できなくなり、そのことを示すマークをデータ・セットに付けます。オープン障害が抑制された後で、**EXEC CICS INQUIRE DSNAME** コマンドまたは **CEMT INQUIRE DSNAME RECOVSTATUS** コマンドを使用すると、NOTRECOVERABLE という応答が返されます。要求についてのロギングは、このデータ・セットに対して続行されます。ロギングは、BACKOUT が定義されているファイルのみを使用します。

**EXEC CICS** コマンドまたは **CEMT SET DSNAME REMOVE** コマンドが発行されるまで、または CICS の初期始動またはコールド・スタートまで (関連付けられているデータ・セットがバックアウト障害状態にない場合)、データ・セットの不一致状態は継続します。

不一致が受け入れられた時点で、CICS は保全性が保証できなくなることを警告するメッセージを発行します。同じ基本データ・セットに対するファイルが開かれる順序によって、受け取るメッセージの内容が決まります。

基本クラスター・ブロックがリカバリー不能として設定されている場合に、不一致が発生すると、データ・セットが完全にリカバリーされる前に、リカバリー不能ファイルによるデータ・セットへのアクセスが許可されます。

受け入れる不一致と拒否する不一致を選択する手段として、XFCNREC 出口に 3 つのパラメーターが渡されます。これらのパラメーターは、ファイル名のアドレス、基本データ・セット名のアドレス、およびファイル・バックアウト標識が入っているバイトのアドレスです。出口は不一致がある場合のみ呼び出されるので、データ・セットのバックアウト標識はファイルの設定から取得されることがあります。

注: XFCNREC を使用して、不一致によるオープン障害を抑止する場合、グローバル・ユーザー出口 XFCNREC は、基本データ・セットのバックアウト設定を出口パラメーター UEFCRV として渡し、ファイルのバックアウト設定は渡しません。これらのバックアウト設定は異なる場合があります。

### **BWO の不一致がある場合の XFCNREC の使用**

出口 XFCNREC を使用してファイルを開くことができ、CICS は通常どおり実行を継続します。ただし、開いたデータ・セットに対して順方向リカバリーを使用することはできません。

### **ファイル制御静止受信出口 XFCVSDS**

XFCVSDS 出口は、シスプレックス内で発生している一部のデータ・セット関連イベントの結果として処理が必要であることを VSAM RLS が CICS に通知したときに呼び出されます。

XFCVSDS は、データ・セットにデータ・セット名ブロック (DSNB) が存在する場合に限り、CICS 処理が実行される前に呼び出されます。XFCVSDS が呼び出される要因となるアクションは以下のとおりです。

- シスプレックス全体にわたってデータ・セットが静止中である。

CICS は、データ・セットに対して RLS モードで開かれたファイルがある場合のみ、このアクションに関する通知を受けます。

CICS が静止アクションに関する通知を受けた場合、XFCVSDS グローバル・ユーザー出口プログラムはデータ・セット静止を取り消すことができます。その場合、この出口プログラムはシスプレックス全体にわたる静止を取り消し、データ・セットは静止解除状態のままになります。

- データ・セットがシスプレックス全体にわたって静止解除中である。

VSAM RLS 制御 ACB に登録されているシスプレックス内のすべての CICS 領域が、静止解除アクションに関する通知を受けます。

- **DFSMSdss がデータ・セットの非 BWO バックアップの開始を予定している。**

CICS は、データ・セットに対して RLS モードで開かれたファイルがある場合のみ、非 BWO バックアップ開始アクションに関する通知を受けます。

CICS が非 BWO バックアップ開始アクションに関する通知を受けた場合、XFCVSDS を使用してバックアップを取り消すことができます。

- **DFSMS がデータ・セットの非 BWO バックアップを完了した。**

VSAM RLS 制御 ACB に登録されているシスプレックス内のすべての CICS 領域が、非 BWO バックアップ完了アクションに関する通知を受けます。

- **DFSMS がデータ・セットの BWO バックアップの開始を予定している。**

CICS は、データ・セットに対して RLS モードで開かれたファイルがある場合のみ、BWO バックアップ開始アクションに関する通知を受けます。

CICS が BWO バックアップ開始アクションに関する通知を受けた場合、XFCVSDS を使用してバックアップを取り消すことができます。

- **DFSMS がデータ・セットの BWO バックアップを完了した。**

VSAM RLS 制御 ACB に登録されているシスプレックス内のすべての CICS 領域が、BWO バックアップ完了アクションに関する通知を受けます。

## 出口 XFCVSDS

出口 XFCVSDS は、シスプレックス内で発生しているデータ・セット関連アクションの結果として処理が必要であることを RLS が CICS に通知した後に呼び出されます。

### 呼び出される状況

シスプレックス内で発生しているデータ・セット関連アクションの結果として処理が必要であることを、RLS が CICS に通知した後に呼び出されます。

### 出口固有のパラメーター

#### UEPDSNAM

アクションが適用されるデータ・セットの名前を格納する 44 バイト・フィールドのアドレス

#### UEPVSACT

CICS が通知を受けた対象の RLS アクションを示す 1 バイト・フィールドのアドレス。可能な値は次のとおりです。

#### UEQUIES

静止

#### UEUNQUIS

静止解除

#### UENBWST

非 BWO バックアップの開始

#### UENBWCMP

非 BWO バックアップの完了

#### UEBWOST

BWO バックアップの開始

#### UEBWOCMP

BWO バックアップの完了

#### UEPQUCLS

RLS モードで開いたファイルを閉じる方法を示す 1 バイト・フィールドのアドレス (UEQUIES の場合のみ)。可能な値は次のとおりです。

**UEORDCLO**

閉じる前に、データ・セットにアクセスしている実行中の UOW すべてが同期点を完了するまで待ちます。

**UEIMMCLO**

閉じる前に、データ・セットにアクセスしている実行中の UOW すべてを異常終了します。

**UEPCPTEC**

バックアップが並行コピー技法を使用するかどうかを示す 1 バイト・フィールドのアドレス (UENBWST および UEBWOST の場合のみ)。可能な値は次のとおりです。

**UEORDCOP**

並行コピーは使用されません。

**UECONCOP**

並行コピーが使用されます。

**戻りコード****UERCNORM**

処理を続行し、VSAM RLS 操作をサポートするために必要なアクションを完了します。

**UERCBYR**

これは、アクション UEQUIES、UENBWST、および UEBWOST のみに適用されます。CICS は、VSAM RLS アクションに必要な処理を実行せず、シスプレックス全体でアクションを取り消します。

UERCPLRG の戻りコードは使用できません。

**XPI 呼び出し**

すべて使用できます。

**API および SPI 呼び出し**

この出口では、CICS API コマンドと SPI コマンドを使用できます。一般的には、すべてを使用できますが、以下の制約事項があります。

- 発行タスクを中断するコマンドの使用を避ける必要があります。
- EXEC CICS SHUTDOWN または EXEC CICS XCTL を使用してはなりません。
- データ・セット UEPDSNAM に対して EXEC CICS SET DSNAME の QUIESCESTATE オプションを使用してはなりません。
- データ・セット UEPDSNAM を参照するファイルに対して、EXEC CICS SET FILE の OPENSTATUS オプションを使用してはならず、ファイル制御要求を発行することもできません。

## ファイル制御静止送信出口 XFCQUIS

XFCQUIS グローバル・ユーザー出口は、CEMT コマンドまたは **EXEC CICS SET DSNAME QUIESCESTATE** コマンドのいずれかによって要求されたデータ・セットの VSAM RLS 静止または静止解除の完了時に呼び出されます。

この出口は、QUIESCESTATE アクションが正常に完了したか正常に完了しなかったかに関係なく呼び出されます。これにより、静止処理または静止解除処理が完了するまで実行できない処理を実行したり、スケジュールしたりすることができます。

**呼び出される状況**

SET DSNAME QUIESCESTATE コマンドの完了時 (成功か失敗かを問わず)。

**出口固有のパラメーター****UEPQDSNM**

静止または静止解除の対象になっていたデータ・セットの名前が含まれた 44 バイト・フィールドのアドレス。

**UEPQSTAT**

データ・セットが静止または静止解除の対象となっていたかどうかを示す 1 バイト・フィールドのアドレス。可能な値は次のとおりです。



**UEQSD**

データ・セットは QUIESCESTATE(QUIESCED) によって静止中でした。データ・セットにアクセスしていた未完了の UOW は、RLS モードで開いたファイルが閉じられる前の同期点で完了しました。

**UEIMQSD**

データ・セットは QUIESCESTATE(IMMQUIESCED) によって静止中でした。データ・セットにアクセスしていた未完了の UOW は、RLS モードで開いたファイルが閉じられる前に異常終了しました。

**UEUNQSD**

データ・セットは QUIESCESTATE(UNQUIESCED) によって静止解除中でした。

**UEPQRCDE**

静止または静止解除の結果を示す 1 バイト・フィールドのアドレス。可能な値は次のとおりです。

**UEQOK**

正常に終了

**UEQREJEC**

拒否。理由コードについて UEPQCONF を参照してください。

**UEQUNKNO**

DFSMS に VSAM データ・セットとして認識されていないデータ・セットのため、失敗しました。

**UEQIOERR**

RLS エラーのため、または SMSVSAM サーバーが使用できないために失敗しました。

**UEQCANCL**

ユーザーによって静止が取り消されたために失敗しました (UEQSD および UEIMQSD のみ)。

**UEQTIMED**

タイムアウトが原因で静止が取り消されたために失敗しました (UEQSD および UEIMQSD のみ)。

**UEQMIGRT**

データ・セットがマイグレーションされたために失敗しました。

**UEPQCONF**

静止または静止解除が拒否された理由を示す 1 バイト・フィールドのアドレス (UEQREJEC の場合のみ)。可能な値は次のとおりです。

**UEQUIINP**

静止が進行中です (UEQSD 状況および UEIMQSD 状況の場合のみ)。

**UEUNQINP**

静止解除が進行中です。

**UENBWINP**

非 BWO コピーが進行中です。

**UEBWOINP**

BWO コピーが進行中です。

**UEUNKINP**

不明イベントが進行中です。

**戻りコード****UERCNORM**

処理が続行される。

UERCPURG の戻りコードは使用できません。

**XPI 呼び出し**

すべて使用できます。

## API および SPI 呼び出し

CICS の API コマンドおよび SPI コマンドをこの出口で使用できます。一般に、EXEC CICS SHUTDOWN および EXEC CICS XCTL を除くすべてを使用できますが、EXEC CICS SET DSNAME の QUIESCESTATE キーワードを使用することはできません。

## ファイル制御リカバリー・プログラム出口 XFCBFAIL、XFCBOUT、XFCBOVER、および XFCLDEL

CICS では、ファイル制御リカバリー操作と関連付けて使用できる 4 つのグローバル・ユーザー出口を提供します。

### XFCBFAIL

バックアウト中にエラーが発生したときに呼び出されます。

### XFCBOUT

CICS がファイル更新をバックアウトしようとしているときに呼び出されます。

### XFCBOVER

RLS が保持しているロック保護をバッチ・プログラムがオーバーライドしてバッチ処理用にデータ・セットを開いたために、CICS が作業単位 (UOW) のバックアウトをスキップしようとしているときに呼び出されます。

### XFCLDEL

BDAM または VSAM の ESDS データ・セットへの書き込みをバックアウトするときに呼び出されます。

## ファイル制御リカバリー出口プログラムのサンプル

CICS には、ファイル制御グローバル・ユーザー出口プログラムのサンプルとして、DFH\$FCBF、DFH\$FCBV、および DFH\$FCLD の 3 つが組み込まれています。

## 呼び出し順序

ファイル制御リカバリー・プログラムの個々の出口は、ファイル更新をバックアウトしようとしているときに呼び出されることがあります。バックアウトが失敗すると、バックアウトの再試行時に各出口が再び呼び出される可能性があります。出口プログラムは、元のバックアウト試行の際に呼び出されたか、再試行時に呼び出されているのかを判別する必要がある場合、XPI INQUIRE\_TRANSACTION 呼び出しから返された RE\_ATTACHED\_TRANSACTION フィールドの値を検査することができます。

出口が情報をやり取りする方法、および出口が呼び出される順序を以下のリストに示します。各バックアウト試行または再試行されたバックアウト試行についてすべての出口が有効になっているとすると、以下のようになります。

1. バックアウト中にオープンが失敗した場合は、XFCBFAIL が呼び出されます。その他の出口は何も呼び出されません。
2. バックアウト中のデータ・セットに対して SHCDS PERMITNONRLSUPDATE コマンドが発行された場合は、XFCBOVER が呼び出されます。この出口から **UERCNORM** (バックアウトを実行しない) が返されると、それ以降は出口は呼び出されません。
3. 項目 106 ページの『1』が適用されない場合、または XFCBOVER が呼び出されて UERCNORM を返した場合は、XFCBOUT が呼び出されます。
4. バックアウトでは、バックアウトされるレコードに対する読み取り更新要求を発行します。  
この読み取り更新が失敗すると、XFCBFAIL が呼び出されます。その後に呼び出される出口はありません。
5. バックアウトされる更新が物理的削除をサポートしないデータ・セット (つまり、BDAM データ・セットまたは VSAM ESDS) に書き込まれた場合、XFCLDEL が呼び出されます。
6. この時点以降に障害が発生した場合は、XFCBFAIL が呼び出されます。

## 出口プログラムの使用可能化

出口プログラムを使用可能にするには、以下の可能な 2 つのアクションのうち、どちらかを実行する必要があります。

これらの出口を使用可能にするには、以下のどちらかを行う必要があります。

- システム初期設定パラメーター TBEXITS=(name1,name2,name3,name4,name5,name6) を指定する。  
ここで、name1 から name6 は、XRCINIT、XRCINPT、XFCBFAIL、XFCLDEL、XFCBOVER、および XFCBOUT のユーザー出口プログラムの名前です。
- PLTPI プログラムを使用した初期設定の最初の段階で、出口を使用可能にする。

TBEXITS パラメーターを使用して出口を使用可能にすると、4 バイトのグローバル作業域が提供されます。PLTPI プログラムを使用する場合は、グローバル作業域のサイズを選択できます。また、それぞれの出口点で複数の出口プログラムを使用可能にすることもできます。TBEXITS パラメーターは、それぞれの出口点でただ 1 つの出口プログラムを使用可能にすることができます。PLTPI 処理については、[Writing initialization and shutdown programs](#) に説明があります。

## 出口 XFCBFAIL、ファイル制御バックアウト障害出口

出口 XFCBFAIL は、ファイル・レコードに対する更新のバックアウト中に障害が発生すると必ず呼び出されます。

特定の UOW において、同じファイル内の複数のレコードで、または複数のファイル内のレコードで、バックアウト障害が発生した場合は、以下のように出口が呼び出されます。

- バックアウト障害が発生した各データ・セット内の最初のレコードに対して。

複数のファイルが単一のデータ・セットに関連付けられている場合は、UOW 内のバックアウト障害が発生した最初のファイルにある最初のレコードによってのみ、CICS による出口の呼び出しが生じます。それ以降のレコードはすべて、同じエラーで失敗しますが、出口が再び呼び出されることはありません。

- この UOW のバックアウトの再試行中に障害が発生した各データ・セットの最初のレコードに対して。

UOW 内の他の (非ファイル制御) リソースに対するバックアウト障害の場合は、出口は呼び出されません。

VSAM データ・セットの場合は、バックアウトを後から再試行できるように、バックアウト障害処理によって情報が保存されます。

BDAM データ・セットの場合は、バックアウトを再試行できません。BDAM データ・セットに対するバックアウトが失敗した場合は、XFCBFAIL 出口を使用すると、CICS を即時に終了させることによって、データ保全性を保つことができます。XFCBFAIL 出口が有効でない場合、および CICS を終了しない場合は、BDAM データがコミットされ、ロックが解放されます。出口が有効である場合は、XFCBFAIL グローバル・ユーザー出口プログラムを使用すると、データを手動で修正するために使用できる情報を保存できます。ただしこの操作を行うときには、バックアウト障害が発生してから手動のリカバリー・アクションを実行するまでの間に加えられたその他の変更はバックアウトされないことに注意してください。

### 呼び出される状況

ファイルに対して加えられた変更のバックアウト中にエラーが発生した場合 (ファイルに関連付けられたデータ・セットの UOW 内の最初の障害)。

### 出口固有のパラメーター

#### UEPBLOGR

ファイル制御障害が発生したときにバックアウトされていた更新を表すログ・レコードのファイル制御部分のアドレス。ログ・レコードをマップするには、DSECT DFHFCLGD を使用します。

#### UEPTRANS

バックアウト中の更新が実行されたトランザクションの ID を含む 4 バイト・フィールドのアドレス。

#### UEPTRMNL

バックアウト中の更新が実行された端末または基本機能の 4 バイトの端末 ID のアドレス。

**UEPTASK**

バックアウト中の更新が実行されたタスクのタスク番号を含む 4 バイト (パック 10 進数) のフィールドのアドレス。

**UEPFCRSP**

ファイル制御応答バイトのアドレス。これには、以下のいずれかの値を含めることができます。

**UEAIXFUL**

非固有代替索引にスペースがありません。

**UECACHE**

RLS キャッシュ障害またはキャッシュ接続障害。

**UENBWBK**

非 BWO バックアップが進行中。

**UEDLOCK**

デッドロックが検出されました。

**UEDUPREC**

固有の代替索引の重複キー。

**UEIOEROR**

入出力エラー。

**UELCKFUL**

RLS ロック構造がフルです。

**UENOLDEL**

実行されなかった論理削除 (XFCLDEL 出口点が有効になっていないか、または XFCLDEL グローバル・ユーザー出口プログラムが論理削除実行しないように選択された)。

**UENOSPAC**

ストレージ外部のデータ・セット。

**UEOPENER**

ファイル・オープン中のエラー。

**UERLSERR**

SMSVSAM RLS サーバー障害。

**UERLSDIS**

RLS アクセスが現在無効化されています。

**UERLSCON**

SMSVSAM RLS サーバーの新規インスタンスを使用してスレッド続行を試行します。

**UEUNEXP**

予期しないエラーです。

**UEPERR**

エラー・タイプを含む 1 バイト・フィールドのアドレス。エラー・バイトおよびそれらの意味の値については、『[109 ページの『UEPERR によって参照されるエラー・タイプ・バイトの値』](#)』で説明しています。

**戻りコード****UERCNORM**

処理を継続し、CICS バックアウト障害制御を呼び出します。

これにより、バックアウト障害のエラー・メッセージが出されます (VSAM データ・セットの場合は DFHFC4701、BDAM データ・セットの場合は DFHFC4702)。VSAM データ・セットの場合は、CICS によってレコード・ロックが保存ロックに変換され、後からバックアウトを再試行できるようにログ・レコードが保存されます。

**UERCBYBYP**

エラーを無視し (CICS バックアウト障害制御を呼び出さず)、続行します。この戻りコードを設定すると、データの保全性が保たれなくなる可能性があります。

UERCPU RG の戻りコードは使用できません。この出口が呼び出された条件は、ページされた条件を XPI または API 呼び出しで戻すことができないことを表しているため、UERCPU RG 戻りコードを設定する必要はありません。

### **XPI 呼び出し**

すべて使用できますが、API および SPI 呼び出しの場合と同じ注意が必要です。

### **API および SPI 呼び出し**

この出口では API および SPI 呼び出しを発行できますが、同期点フェーズ 2 の一部であるファイルのバックアウト中に出口が呼び出されるため、使用するコマンドは注意深く決定する必要があります。

EXEC CICS コマンドを照会のみで制限すること、およびリソースはリカバリー状態である可能性があるため、CICS リソースを更新するコマンドを使用しないようにすることをお勧めします。特に、以下の制約事項が適用されます。

1. リカバリー可能操作を実行しないでください。
2. ターゲット・リソースがリカバリー不能である場合でも、この CICS の外部のシステムまたはリソースの所有者にアクセスする操作を使用しないでください。
3. ファイルを無効にしたり、クローズしたりすると、さらにエラー条件が発生する可能性があるため、このような操作は行わないでください。
4. バックアウト中の更新が最初に行われたトランザクション環境とは異なるトランザクション環境で、この出口を呼び出すことができます。出口プログラムが、元のトランザクション環境で実行することが求められるアクション (例えば、端末へのメッセージの書き込み) を実行するには、トランザクション・マネージャー INQUIRE\_TRANSACTION XPI 呼び出しの RE\_ATTACHED\_TRANSACTION パラメーターに返された値を最初に確認する必要があります。

### **XFCBFAIL 用のグローバル・ユーザー出口のサンプル・プログラム**

DFH\$FCBF では、ファイル制御バックアウト障害グローバル・ユーザー出口 XFCBFAIL の処理例を提供しています。詳細については、[ファイル制御リカバリー・サンプル出口プログラム: DFH\\$FCBF](#) を参照してください。

### **UEPERR によって参照されるエラー・タイプ・バイトの値**

XFCBFAIL パラメーター・リスト内の UEPERR フィールドはエラー・タイプ・バイトを指します。エラー・タイプ・バイトにはいくつかの可能な値のうちの 1 つが入ります。

可能な値は次のとおりです。

### **XBFERU**

READ UPDATE 要求の処理中に、ファイル制御ファイル要求ハンドラー・プログラムからエラー応答が返されました。この要求は、バックアウトの前にレコードの既存コピーを取得するために、ファイル制御バックアウトによって発行されます。

UEPFCRSP をレコードのタイプと組み合わせて使用して、具体的な問題を判別してください。レコードのタイプは、パラメーター UEPBLOGR によってアドレス指定されたログ・レコードのファイル制御部分に示されています。UEPBLOGR によってアドレス指定されたストレージ域には、「read-update」レコードの変更前イメージ、または削除する「write-add」の新規コピーが含まれています。レコード・タイプ・フィールド FLJB\_RECORD\_TYPE は、DSECT DFHFCLGD で定義されます。

### **XBFERE**

REWRITE 要求の処理中に、ファイル制御ファイル要求ハンドラー・プログラムからエラー応答が返されました。この要求は、UEPBLOGR によってアドレス指定されるログ・レコード内に保持された「変更前イメージ」で設定されているデータのレコードの既存コピーを置き換えるために、ファイル制御バックアウトによって発行されます。発生したエラーを判別するには、パラメーター UEPFCRSP を使用します。

### **XBFEWR**

WRITE 要求の処理中に、ファイル制御ファイル要求ハンドラー・プログラムからエラー応答が返されました。この要求は、削除するレコードの「変更前イメージ」を追加するために、ファイル制御バックアウトによって発行されます。発生したエラーを判別するには、パラメーター UEPFCRSP を使用します。

## **XBFDL**

REWRITE DELETE 要求の処理中に、ファイル制御ファイル要求ハンドラー・プログラムからエラー応答が返されました。この要求は、VSAM データ・セットに追加された新規レコードを削除するために、ファイル制御バックアウトによって発行されます。発生したエラーを判別するには、パラメーター UEPFCRSP を使用します。

## **XBFENO**

ファイル制御バックアウト中に発生した障害は、ファイル制御ファイル要求ハンドラー・プログラムからのエラー応答の結果ではありません。発生したエラーを判別するには、パラメーター UEPFCRSP を使用します。

## **出口 XFCBOUT、ファイル制御バックアウト出口**

XFCBOUT は、ファイル制御更新がバックアウトされようとしたときに呼び出されます。バックアウトされるレコードの変更前イメージが含まれたログ・レコードが出口プログラムに渡されます。

XFCBOUT は、出口プログラムが更新のバックアウトをバイパスできるようにするための戻りコードを提供しません。これを行うと、データ破壊を招くことになるからです。しかし、ファイル名がログ・レコードに含まれているため、出口プログラムは **EXEC CICS INQUIRE FILE** コマンドを使用してファイルに関する情報を取得することができます。

### **呼び出される状況**

ファイル制御によって更新 (ログ・レコードの変更前イメージにより表される) がバックアウトされるときに呼び出されます。

### **出口固有のパラメーター**

#### **UEPFLOGR**

バックアウトに関して表示されるログ・レコードのファイル制御部分のアドレス。これは、DSECT DFHFCLGD によってマップされます。

### **戻りコード**

#### **UERCNORM**

処理が続行される。

UERCPURG の戻りコードは使用できません。UERCPURG 戻りコードを設定する必要はありません。この出口は同期点フェーズ 2 の間に呼び出され、したがってこの出口で実行する呼び出しからはページされた応答は取得できないためです。

### **XPI 呼び出し**

すべて使用できますが、API および SPI 呼び出しの場合と同じ注意が必要です。

### **API および SPI 呼び出し**

この出口では API および SPI 呼び出しを発行できますが、同期点フェーズ 2 の一部であるファイルのバックアウト中に出口が呼び出されるため、使用するコマンドは注意深く決定する必要があります。

EXEC CICS コマンドを照会のみで制限すること、およびリソースはリカバリー状態である可能性があるため、CICS リソースを更新するコマンドを使用しないようにすることをお勧めします。特に、以下の制約事項が適用されます。

1. リカバリー可能操作を実行しないでください。
2. ターゲット・リソースがリカバリー不能である場合でも、この CICS の外部のシステムまたはリソースの所有者にアクセスする操作を使用しないでください。
3. ファイルを無効にしたり、クローズしたりすると、さらにエラー条件が発生する可能性があるため、このような操作は行わないでください。
4. バックアウト中の更新が最初に行われたトランザクション環境とは異なるトランザクション環境で、この出口を呼び出すことができます。出口プログラムが、元のトランザクション環境で実行することが求められるアクション (例えば、端末へのメッセージの書き込み) を実行するには、トランザクション・マネージャー INQUIRE\_TRANSACTION XPI 呼び出しの RE\_ATTACHED\_TRANSACTION パラメーターに返された値を最初に確認する必要があります。



XFCBOUT は特定のアプリケーションに使用されると予想されるため、汎用のサンプル出口プログラムは提供されません。他のファイル制御リカバリー出口 (DFH\$FCBF、DFH\$FCBV、または DFH\$FCLD) のサンプルをどれでも、XFCBOUT 出口プログラムのベースとして使用することができます。

## 出口 XFCBOVER、ファイル制御バックアウト・オーバーライド出口

出口 XFCBOVER は、CICS ファイル制御が VSAM RLS 環境でバッチ・ウィンドウ 用に提供するサポートの一部です。

VSAM RLS は、データ・セット内の個々のレコードをロックします。バックアウトまたは未確定の障害が原因で完了していない UOW の場合、これらのロックが保持ロックに変換され、この方法でデータ保全性が保持されます。保持レコード・ロックを認識しない非 RLS バッチ・ジョブによってデータ・セットが破壊されるのを防ぐため、ロック・レコードが含まれている場合は、通常、データ・セットを非 RLS モードで更新するために開くことはできません。

### XFCBOVER 用のグローバル・ユーザー出口のサンプル・プログラム

DFH\$FCBV では、ファイル制御バックアウト・オーバーライド・グローバル・ユーザー出口 XFCBOVER の処理例を提供しています。詳しくは、[ファイル制御リカバリー・サンプル出口プログラム: DFH\\$FCBV](#) を参照してください。

### バッチの保持ロックのオーバーライド

この種のロックをオーバーライドして、バッチ処理用にデータ・セットを強制的に開くことが必要な場合があります。

例えば、次のような場合です。

- レコードがロックされる原因となった状況を解決するためのバッチ・ジョブを実行する前に 使用可能な時間が不十分である。または
- バッチ・ジョブはデータ・セット内の既存のレコードを更新しないため、または CICS が更新した可能性のあるレコードを更新しないため、そのバッチ・ジョブがデータ保全性を破壊しないことが判明している。

開かれている制約をオーバーライドするために、VSAM RLS では SHCDS PERMITNONRLSUPDATE コマンドを提供して、保持ロックが存在する場合でも、非 RLS バッチ・ジョブで更新用の範囲を開くことができますようにしています。

### CICS での保持ロック・オーバーライドの影響

VSAM は、保持ロックをオーバーライドするオプションの使用状況を記録して、CICS 領域が次回そのデータ・セットを開く際に、その領域に通知できるようにします。データは非 RLS バッチ・ジョブによって変更された可能性があるため、CICS が (バッチ・ジョブの時点でバックアウト失敗または未確定失敗の状態であった UOW に対して) リカバリーを実行すると、結果は予測不能です。したがってこの状況では、デフォルトの CICS アクションでは、ロックがオーバーライドされた時点で未処理であった更新のバックアウトは行われず、CSFL 一時データ・キューで無視された各バックアウトに関する診断情報も書き込まれません。

XFCBOVER グローバル・ユーザー出口は、バックアウトが無視される各 UOW ログ・レコードについて、以下の操作が可能になるように用意されています。

- CICS 提供の診断を補足するために、アプリケーション関連の診断を書き込む。
- アプリケーション関連のリカバリー・アクションを実行する。
- いずれにしてもバックアウトを実行する必要があることを要求することによって、デフォルトを逆転させる。このオプションが必要になるのは、バッチ・ジョブがデータ保全性を破壊しないことが判明している場合 (例えば、レコードの挿入のみを行う場合) です。

### 呼び出される状況

バックアウトが必要な UOW ログ・レコードを CICS が無視しようとする都度 (これは、更新済みレコードを保護していたロックが非 RLS バッチ・プログラムによってオーバーライドされた可能性があるためです。)

## 出口固有のパラメーター

### UEPOLOGR

保持ロックがオーバーライドされた可能性のあるデータ・セットに対する更新を表す中断ログ・レコードのファイル制御部分のアドレス。ログ・レコードのファイル制御部分は、DSECT DFHFCLGD を使用してマップすることができます。

### UEPODSN

ロックがオーバーライドされたデータ・セットの名前が含まれた 44 バイトのストレージ領域のアドレス。

## 戻りコード

### UERCNORM

このログ・レコードのバックアウトは実行しないでください。バッチ実行によって実行される更新を優先する必要があります。

### UERCBACKO

バックアウトを実行します。バッチ・ジョブのアクションはこの更新に影響しなかったことが判明しています。

UERCPCURG の戻りコードは使用できません。UERCPCURG 戻りコードを設定する必要はありません。このグローバル・ユーザー出口は同期点フェーズ 2 の間に呼び出され、したがってこの出口で実行する呼び出しからはパージされた応答は取得できないためです。

## XPI 呼び出し

すべて使用できますが、API および SPI 呼び出しの場合と同じ注意が必要です。

## API および SPI 呼び出し

この出口では API および SPI 呼び出しを発行できますが、同期点フェーズ 2 の一部であるファイルのバックアウト中に出口が呼び出されるため、使用するコマンドは注意深く決定する必要があります。

EXEC CICS コマンドを照会のみで制限すること、およびリソースはリカバリー状態である可能性があるため、CICS リソースを更新するコマンドを使用しないようにすることをお勧めします。特に、以下の制約事項が適用されます。

1. リカバリー可能操作を実行しないでください。
2. ターゲット・リソースがリカバリー不能である場合でも、この CICS の外部のシステムまたはリソースの所有者にアクセスする操作を使用しないでください。
3. ファイルを無効にしたり、クローズしたりすると、さらにエラー条件が発生する可能性があるため、このような操作は行わないでください。
4. バックアウト中の更新が最初に行われたトランザクション環境とは異なるトランザクション環境で、この出口を呼び出すことができます。出口プログラムが、元のトランザクション環境で実行することが求められるアクション (例えば、端末へのメッセージの書き込み) を実行するには、トランザクション・マネージャー INQUIRE\_TRANSACTION XPI 呼び出しの RE\_ATTACHED\_TRANSACTION パラメーターに返された値を最初に確認する必要があります。

## 出口 XFCLDEL、ファイル制御論理削除出口

出口 XFCLDEL は、VSAM ESDS または BDAM データ・セットに対する WRITE がバックアウトされるときに必ず呼び出されます。これらのタイプのデータ・セットは削除をサポートしていないため、XFCLDEL を使用して、削除のフラグを立てる何らかの方法でレコードを修正することによって、論理削除を実行できます。

## 出口固有のパラメーター

### UEPBLOGR

論理削除によってバックアウトされている更新を表すログ・レコードのファイル制御部分のアドレス。ログ・レコードをマップするには、DSECT DFHFCLGD を使用します。

### UEPTRANS

バックアウト中の更新が実行された 4 バイトのトランザクションの ID のアドレス。

### UEPTRMNL

バックアウト中の更新が実行された端末または基本機能の 4 バイトの端末 ID のアドレス。



## UEPTASK

バックアウト中の更新が実行されたタスクの 4 バイト (パック 10 進数) のタスク番号のアドレス。

## UEPFDATA

ファイル制御要求のデータを含む可変長フィールドのアドレス。出口プログラムは、このフィールドでアドレス指定されたレコード・データに、論理的に削除されたレコードを表すものとしてアプリケーションに認識される何らかの方法でマークを付けることによって、そのデータを修正できます。

## UEPFLEN

ファイル制御要求のデータの長さを含むフルワードのアドレス。

## 戻りコード

### UERCFAIL

論理削除を実行せず、これをバックアウト障害として扱います。これは、出口が無効になっている場合に実行されるデフォルト・アクションです。

### UERCLDEL

更新されたレコードを再度適用して、論理削除を実行します。

UERCPURG の戻りコードは使用できません。この出口が呼び出された条件は、「ページ」された条件を XPI または API 呼び出しで戻すことができないことを表しているため、UERCPURG 戻りコードを設定する必要はありません。

## XPI 呼び出し

すべて使用できますが、API および SPI 呼び出しの場合と同じ注意が必要です。

## API および SPI 呼び出し

この出口では API および SPI 呼び出しを発行できますが、同期点フェーズ 2 の一部であるファイルのバックアウト中に出口が呼び出されるため、使用するコマンドは注意深く決定する必要があります。

EXEC CICS コマンドを照会のみ制限すること、およびリソースはリカバリー状態である可能性があるため、CICS リソースを更新するコマンドを使用しないようにすることをお勧めします。特に、以下の制約事項が適用されます。

1. リカバリー可能操作を実行しないでください。
2. ターゲット・リソースがリカバリー不能である場合でも、この CICS の外部のシステムまたはリソースの所有者にアクセスする操作を使用しないでください。
3. ファイルを無効にしたり、クローズしたりすると、さらにエラー条件が発生する可能性があるため、このような操作は行わないでください。
4. バックアウト中の更新が最初に行われたトランザクション環境とは異なるトランザクション環境で、この出口を呼び出すことができます。出口プログラムが、元のトランザクション環境で実行することが求められるアクション (例えば、端末へのメッセージの書き込み) を実行するには、トランザクション・マネージャー INQUIRE\_TRANSACTION XPI 呼び出しの RE\_ATTACHED\_TRANSACTION パラメーターに返された値を最初に確認する必要があります。

CICS ファイル定義では、DFH\$FCLD のような XFCLDEL ユーザー出口によって設定された論理削除フラグを使用して CICS がレコードを更新するようにするために、UPDATE=YES を指定する必要はありません。バックアウトが実行されているとき、UPDATE などの SERVREQS の検査はバイパスされます。

## XFCLDEL 用のグローバル・ユーザー出口のサンプル・プログラム

DFH\$FCLD では、ファイル制御論理削除グローバル・ユーザー出口 XFCLDEL の処理例を提供しています。詳しくは、[ファイル制御リカバリー・サンプル出口プログラム: DFH\\$FCLD](#) を参照してください。

## ファイル制御 RLS 共存プログラム出口 XFCRLSCO

XFCRLSCO 出口は、ファイルのオープンの要求時に呼び出すことができます。この出口を使用することにより、アプリケーションは RLS と読み取り専用の非 RLS との間でモードを切り替えて、特定のデータ・セットにアクセスすることができます。

RLS ファイルが同じデータ・セットに対して既にオープンされている場合、またはデータ・セットがシステム内に未処理の RLS 作業を持つ場合、CICS は非 RLS ファイルに対して行われるオープン要求を許可しま

せん。さらに、既存の非 RLS ファイルが既に基本データ・セットに対してオープンされている場合も、CICS は RLS ファイルに対して行われるオープン要求を許可しません。これらの状態では、オープン要求が行われ、読み取り専用アクセスに対して非 RLS ファイルがオープンになっている場合、XFCRLSCO 出口が駆動されます。この出口を使用して、オープン要求を続行するか、あるいはメッセージ DFHFC0511 または DFHFC0512 で通常の方法で失敗させるかを決定することができます。

アクセス・モードを切り替えるには、アプリケーションは、別のアクセス・モードを持つ新規ファイルを使用してデータ・セットをオープンすることができます。両方のアクセス方式を同時に使用して、長期間にわたって同じデータ・セットをオープンにしたままにしないでください。RLS ファイルと非 RLS ファイルの両方を使用して同時にアクセスすると、CICS はデータ・セットの一貫したビューを受信できないためです。特に、CICS は、データ・セットが非 RLS ファイルによって読み取られているのと同時に RLS ファイルによっても更新されている場合、一貫したビューを受信することができません。

RLS および非 RLS ファイルの両方を使用してデータ・セットがオープンになっている間に、VSAM アップグレード・セット処理が行われる場合、基本クラスターまたは関連した代替索引のいずれかでアップグレード処理が完了していないため、読み取りエラーが発生するリスクが高まります。

#### 注：

1. 非 RLS ファイルに更新可能な SERVREQS セットが含まれている (つまり、更新、追加、または削除が許可されている) 場合、この出口は駆動されません。
2. オープンされるデータ・セットは、VSAM 基本クラスターで共用オプション SHAREOPTION(2) を指定する必要があります。より低い共用オプションが指定されている場合、VSAM は 2 番目のオープンに失敗します。
3. 静的割り振りが使用されている場合は、必ず DISP=SHR を指定してください。そうしないと、VSAM はオープンに失敗します。

#### 呼び出される状況

ファイル・オープン要求時に、オープン要求が VSAM に発行される前。

#### 出口固有のパラメーター

##### UEPFILEN

ファイル名が含まれた 8 バイトのフィールドのアドレス。ファイル名の長さが 8 文字より少ない場合は、ブランクが埋め込まれます。

##### UEPDSNAME

基本データ・セット名を格納する 44 バイト・フィールドのアドレス。データ・セット名の長さが 44 文字より少ない場合は、ブランクが埋め込まれます。

##### UEPFSERV

ファイル SERVREQ 標識を格納する 1 バイト・フィールドのアドレス。可能な値は次のとおりです。

##### UEPFRDIM

有効な標識を読み取ります。

##### UEPFUPDIM

有効な標識を更新します。

##### UEPFADDIM

有効な標識を追加します。

##### UEPFDELIM

有効な標識を削除します。

##### UEPFBRZIM

有効な標識をブラウズします。

##### UEPFDSACC

ファイル・アクセス方式フラグを格納する 1 バイト・フィールドのアドレス。可能な値は次のとおりです。

##### UEPFVSAM

VSAM ファイル標識。

**UEPFDTBL**

データ・テーブル・ファイル標識。

**UEPFDTUM**

ユーザー・データ・テーブル・ファイル標識。

**UEPFRLS**

RLS ファイル標識。

**UEPFCFDT**

CFDT ファイル標識。

**UEPRECUR**

ハーフワードの再帰レベルのアドレス。

## 戻りコード

**UERCNORM**

通常どおりに処理を続行します。オープン要求は失敗します。

**UERCBYBYP**

オープンを実行することを許可し、共存の障害を迂回します。

**XPI 呼び出し**

XPI コマンドを使用できますが、コマンドの結果いずれかのファイルの状態が変更されてはなりません。

**API および SPI 呼び出し**

EXEC CICS API および SPI コマンドを使用できますが、コマンドによっていずれかのファイルの状態が変更されてはなりません。例えば、**EXEC CICS INQUIRE FILE** は使用できますが、**EXEC CICS SET FILE** やいずれかのファイルの状態が変更されることになるファイル制御に対する EXEC CICS API コマンドは使用できません。

**Good morning メッセージ・プログラム出口 (XGMTEXT)**

この出口は、「Good morning」メッセージが送信される前に呼び出されます。

## 呼び出される状況

Good morning メッセージが送信される前。

## 出口固有のパラメーター

**UEPTCTTE**

端末制御テーブル・エントリー (TCTTE) のアドレス。DSECT DFHTCTTE を使用して TCTTE をマップすることができます。

**UEPTIOA**

端末入出力域 (TIOA) のアドレス。DSECT DFHTIOA を使用して TIOA をマップすることができます。ただし、フィールド TIOASAL および TIOASCA はプログラミング・インターフェースではありません。

## 戻りコード

**UERCNORM**

処理が続行される。

**UERCPUFG**

タスクは XPI 呼び出し中にページされます。

**XPI 呼び出し**

すべて使用できます。

**HTTP クライアントのオープンおよび送信出口: XWBAUTH、XWBOPEN、および XWBSNDO**

出口 XWBAUTH、XWBOPEN、および XWBSNDO は、**EXEC CICS WEB CONVERSE**、**EXEC CICS WEB OPEN**、**EXEC CICS INVOKE SERVICE**、および **EXEC CICS WEB SEND** の各コマンドの処理中に呼び出

されます。これらの出口は、HTTP クライアントとしての CICS から HTTP クライアント 要求を作成する場合に使用されます。これは、CICS Web サポートによって提供される機能です。

## HTTP クライアントの送信出口 XWBAUTH

XWBAUTH を使用すると、ターゲット・サーバーまたはサービス・プロバイダー用に基本認証の資格情報 (ユーザー名およびパスワード) を指定できます。XWBAUTH は、HTTP を使って転送される許可ヘッダーを作成するために、要求に応じてこれらを CICS に渡します。

**EXEC CICS WEB SEND (Client)** コマンドまたは **WEB CONVERSE** コマンドで

**AUTHENTICATE(BASICAUTH)** が指定されている場合は、ユーザー名およびパスワードをアプリケーションから提供することができます。提供されていない場合は、こうした資格情報を指定する別の方法として XWBAUTH が呼び出されます。アプリケーションがユーザー名とパスワードを **EXEC CICS WEB SEND (Client)** コマンドまたは **WEB CONVERSE** コマンドで提供する場合を除き、XWBAUTH は、**USAGE(CLIENT)** の **URIMAP** リソース定義で **AUTHENTICATE(BASIC)** を指定した場合に呼び出されます。また、urimap が **AUTHENTICATE(BASIC)** を指定した場合、XWBAUTH は、**EC INVOKE SERVICE() URIMAP()** を指定して呼び出されます。

ユーザー名とパスワードは、通常はリモート・サーバー環境に特定のもので、**RACF®** システムで使用される標準の 8 文字よりも長いことがあります。ユーザー名フィールドおよびパスワード・フィールドの長さは、最長 256 文字まで可能です。これらのフィールドの構文の妥当性検査は行われません。

ホストは **UEPHOST** パラメーターとしてユーザー出口プログラムに渡され、パスは **UEPPATH** パラメーターとして渡されます。レルムはオプションで、**UEPREALM** パラメーターとして渡されます。応答では、ユーザー出口プログラムによってユーザー名は **UEPUSNM** パラメーターとして、パスワードは **UEPPSWD** パラメーターとして戻されます。提供されたユーザー ID とパスワードをエンコードするとき、CICS は **EBCDIC** コード・ページを使用します。

次に紹介する出口プログラムのサンプルが、CICS サンプル・ライブラリーの **SDFHSAMP** に同梱されています。

- **DFH\$WBPI**
- **DFH\$WBEX**
- **DFH\$WBX1**
- **DFH\$WBX2**
- **DFH\$WBGA**。DFH\$WBPI、DFH\$WBX1、DFH\$WBX2、および DFH\$WBEX の各サンプル が使用するグローバル作業域をマップした、コピーブックです。

クライアントのサンプル 出口プログラムについて詳しくは、[HTTP クライアント・サンプル出口プログラム \(DFH\\$WB\\*\)](#)を参照してください。LDAP プロファイルのセットアップについて詳しくは、[CICS で使用するための LDAP の構成](#)を参照してください。

### 出口 XWBAUTH

#### 呼び出される状況

**EXEC CICS WEB SEND** コマンドまたは **WEB CONVERSE** コマンドで **AUTHENTICATE(BASICAUTH)** は指定されているが、**USERNAME** および **PASSWORD** は指定されていない場合。

#### 出口固有のパラメーター

##### **UEPHOST (CICS 提供の入力)**

ホスト名のアドレス (接続の **WEB OPEN** コマンドの **HOST** オプションで指定された **IPv4** または **IPv6** アドレス) が含まれるフィールドのアドレス。ホスト名は、このフィールドに保管されるときに小文字に変換されます。ユーザー出口プログラムは、ホスト名とのマッチング時に、この変換を考慮に入れる必要があります。

##### **UEPHOSTL (CICS 提供の入力)**

ホスト名のハーフワードの長さが入っているフィールドのアドレス。

**UEPPATH (CICS 提供の入力)**

WEB SEND コマンドまたは WEB CONVERSE コマンドの PATH オプションで指定されたパスのアドレスが含まれるフィールドのアドレス。パスは、指定どおりの大/小文字混合です。

**UEPPATHL (CICS 提供の入力)**

パスのハーフワードの長さが入っているフィールドのアドレス。

**UEPREALM (CICS 提供の入力)**

宛先に関連付けられているレルム名のアドレスが含まれるフィールドのアドレス (サーバーからの前回の HTTP 401 応答で、レルム名が返された場合)。

**UEPREALML (CICS 提供の入力)**

レルム名のハーフワードの長さが入っているフィールドのアドレス。

**UEPAUTHT (CICS 提供の入力)**

認証タイプを示す、1 バイト・コードのアドレス。これはバイナリーの 01 であり、基本認証を示します。

**UEPUSNM (ユーザー出口提供の出力)**

HTTP サーバーへのアクセスに必要なユーザー名のアドレスが含まれるフルワード・フィールドのアドレス。ユーザー名を保管するために、定義済みのアドレスと 64 バイトの領域が CICS によって作成されます。この 64 バイトの領域にユーザー名を入れて、UEPUSNM にあるアドレスは変更せずにそのままにすることができます。また、ユーザー名を独自の領域に入れて、UEPUSNM にあるアドレスを独自のユーザー名のアドレスで置き換えることもできます。独自のユーザー名領域を作成する場合、そのフィールドの長さは 256 バイトまで可能です。

**UEPUSNML (CICS 提供の入力とユーザー出口提供の出力)**

ハーフワード・フィールドのアドレス。最初、このフィールドには、UEPUSNM で提供されるバッファー・アドレスの長さが入ります。ユーザー出口プログラムは、このバッファーの長さを、UEPUSNM で提供されるユーザー名の長さに設定する必要があります。

**UEPPSWD (ユーザー出口提供の出力)**

HTTP サーバーへのアクセスに必要なパスワードのアドレスが含まれるフルワード・フィールドのアドレス。パスワードまたはパスワード・フレーズを保管するために、事前定義されたアドレスおよび 100 バイト領域が CICS によって作成されます。この 100 バイトの領域にパスワードを入れて、UEPPSWD にあるアドレスは変更せずにそのままにすることができます。また、パスワードを独自の領域に入れて、UEPPSWD にあるアドレスを独自のパスワードのアドレスで置き換えることもできます。独自のパスワード領域を作成する場合、そのフィールドの長さは 256 バイトまで可能です。

**UEPPSWDL (CICS 提供の入力とユーザー出口提供の出力)**

ハーフワード・フィールドのアドレス。最初、このフィールドには、UEPPSWD で提供されるバッファー・アドレスの長さが入ります。ユーザー出口プログラムは、このバッファーの長さを、UEPPSWD で提供される実際のパスワードの長さに設定する必要があります。

**UEPHOSTT (CICS 提供の入力)**

UEPHOST パラメーターに含まれているホスト・タイプを示す 1 バイト・コードのアドレス。

2 進数 01 はホスト名を、2 進数 02 は IPv4 アドレスを、2 進数 03 は IPv6 アドレスを示します。

**戻りコード****UERCNORM**

出口は正常にユーザー名およびパスワードを戻しました。

**UERCBYP**

出口がユーザー名およびパスワードを識別できません。Authorization ヘッダーは送信されません。

**UERCERR**

出口がユーザー名およびパスワードを識別できません。WEB SEND (Client) コマンドまたは WEB CONVERSE コマンドは停止される必要があります。

**XPI 呼び出し**

すべての XPI 呼び出しが使用可能です。

## API コマンドおよび SPI コマンド

**EXEC CICS SHUTDOWN** および **EXEC CICS XCTL** を除く、すべての API コマンドおよび SPI コマンドを使用できます。

## XWBAUTH による LDAP XPI 関数の一般的な使用方法

DFHDDAPX 関数の想定される使用方法 (XWBAUTH グローバル・ユーザー出口との関連における) には、LDAP セッションの開始と終了、資格情報の結果の表示、結果のスキャンと探索、表示の終了、正しい値の戻し、および検索の終了が含まれます。

### BIND\_LDAP

LDAP サーバーとのセッションを確立します。グローバル・ユーザー出口 XWBAUTH への最初の呼び出しで一度使用されます。LDAP セッション・トークンは、XWBAUTH のグローバル作業域が用意されている場合はそこに保管され、それ以降の LDAP\_SEARCH に対する呼び出しで使用されます。

### UNBIND\_LDAP

LDAP サーバーとの接続を解放します。この関数は、CICS シャットダウン処理時にのみ必要になります。この関数は、XSTERM (システム終了) グローバル・ユーザー出口の際に使用できます。

### SEARCH\_LDAP

必要なユーザー情報の URL およびレلمを識別する LDAP 識別名を指定して、資格情報を検索します。識別名は次の形式で指定されます。

```
racfcid=uuuuuuuu, ibm-httprealm=rrrrrrrrr, labeledURI=xxxxxxx, cn=BasicAuth
```

ここで、

- uuuuuuuu は、XWBAUTH パラメーター UEPUSER から取得された現在のユーザー ID です。
- rrrrrrrrrr には、XWBAUTH パラメーター UEPREALM から取得される HTTP 401 レルムが入ります (これが存在する場合)。
- xxxxxxxx には、http:// とホスト名を連結することで XWBAUTH パラメーター UEPHOST から取得されるターゲット URL、および、XWBAUTH パラメーター UEPPATH から取得されるパスが入ります。
- cn=BasicAuth は、基本認証資格情報を保管するために、LDAP サーバー内に構成される任意の接尾部です。

### START\_BROWSE\_RESULTS

SEARCH\_LDAP から戻される結果のスキャンを開始します。

### GET\_NEXT\_ENTRY

SEARCH\_LDAP から戻された一連のエントリー内の次の結果エントリーを見つけます。通常は、SEARCH\_LDAP で指定される URL で固有の入力が位置指定されるため、GET\_NEXT\_ENTRY 関数は使用されません。

### GET\_NEXT\_ATTRIBUTE

現在の結果エントリー内の次の属性を見つけます。通常は、固有の属性が選択されるため、GET\_NEXT\_ATTRIBUTE 関数は使用されません。

### END\_BROWSE\_RESULTS

SEARCH\_LDAP で開始された表示セッションを終了します。

### GET\_ATTRIBUTE\_VALUE

ターゲット識別名の各種属性の値を戻します。XWBAUTH の場合、この属性の値は、属性 uid に保管されているユーザー名、および userpassword に保管されているパスワードです。XWBAUTH はこれらの属性値を資格情報として戻します。

### FREE\_SEARCH\_RESULTS

SEARCH\_LDAP によって開始された検索を終了し、関連付けられているストレージを解放します。

## HTTP クライアントのオープン 出口 XWBOPEN

XWBOPEN を使用して、HTTP クライアントとしての CICS による HTTP 要求に使用されるプロキシ・サーバーを指定できます。これらの要求に対して指定されるホスト名にセキュリティ・ポリシーを適用することもできます。

XWBOPEN は、アプリケーション・プログラムがサーバーとの接続を開くために使用する **EXEC CICS WEB OPEN** コマンドの処理中に呼び出されます。XWBOPEN は、**EXEC CICS INVOKE SERVICE** コマンドの処理中にも呼び出されます。

CICS には、HTTP クライアントとしての CICS による HTTP 要求のためのプロキシ・サーバーの使用 (およびその他) に関する要件は一切ありません。また、CICS がこれらの要求にセキュリティ・ポリシーを適用することはありません。ご使用のシステムまたは組織でこのような機能が必要な場合は、ユーザーがセットアップする必要があります。

**EXEC CICS WEB OPEN** コマンドは、CICS Web ドメインに、サーバーとの接続を開くように指示します。XWBOPEN は接続が開かれる前に呼び出されます。**EXEC CICS WEB OPEN** コマンドの HOST オプションで指定される接続のホスト名 (例えば `www.example.com`) は、UEPHOST パラメーターとしてユーザー出口プログラムに渡され、検査されます。この時点で、ユーザー出口プログラムを次の 2 つの目的で使用できます。

- HTTP 要求にプロキシ・サーバーが使用されるかどうかを判別し、必要なプロキシ・サーバーの名前を返す。プロキシ・サーバーが必要な場合は、戻りコード UERCPROX が使用され、プロキシ・サーバーの名前が CICS Web ドメイン (UEPPROXY によって識別されるバッファ内) に返されて、サーバーとの接続を作成するために使用されます。プロキシ・サーバーが必要でない場合は、戻りコード UERCNORM が使用されます。
- セキュリティ・ポリシーをホスト名に適用する。戻りコード UERCBARR は、ホストへのアクセスが許可されていないことを示し、NOTAUTH 応答が WEB OPEN コマンドに返されます。アプリケーション・プログラマーは、その接続を開こうとするのを止めなければなりません。ホストだけでなく (またはホストではなく) 個々のリソースにセキュリティ・ポリシーを適用する場合には、**EXEC CICS WEB SEND** コマンドおよび **EXEC CICS WEB CONVERSE** コマンド上の XWBSNDO ユーザー出口を使用して、その URL のパス構成要素にセキュリティ・ポリシーを適用することができます。

XWBOPEN ユーザー出口は、**EXEC CICS** コマンドの使用をサポートしていません。

関連するコピーブックの DFH\$WBGA を使用するサンプル・プログラム DFH\$WBPI および DFH\$WBEX は、グローバル作業域でプロキシ・サーバー情報またはセキュリティ・ポリシーをセットアップする方法を示しています。例えば、CICS システムからのすべての要求に単一のプロキシ・サーバーを使用しなければならない場合には、そのプロキシ・サーバー名を初期設定パラメーターとして指定することができます。多数のプロキシ・サーバーを使用しているか、またはセキュリティ・ポリシーを異なるホスト名に適用する場合、ホスト名を適切なプロキシ・サーバーに一致させるか、禁止のマークを付けるテーブルをロードまたは作成できます。これにより、**EXEC CICS WEB OPEN** コマンドの処理の間に、検索テーブルとして使用されます。サンプル・プログラムは、プログラム・リスト・テーブルの初期設定後 (PLTPI) 処理中に実行するか、または **EXEC CICS WEB OPEN** コマンドの使用が予期される前の任意の時点で実行することができます。

### 出口 XWBOPEN

#### 呼び出される状況

**EXEC CICS WEB OPEN** コマンドまたは **EXEC CICS INVOKE SERVICE** コマンドの処理中。

#### 出口固有のパラメーター

##### UEPHOST (CICS 提供の入力)

ホスト名 (WEB OPEN コマンドの HOST オプションで指定された IPv4 または IPv6 アドレス) が含まれるフィールドのアドレス。

注: ホスト名は、このフィールドに保管されるときに小文字に変換されます。ユーザー出口プログラムは、ホスト名とのマッチング時に、この変換を考慮に入れる必要があります。

##### UEPHOSTL (CICS 提供の入力)

ホスト名のハーフワードの長さが入っているフィールドのアドレス。

### UEPPROXY (ユーザー出口提供の出力)

プロキシ・サーバー名を指し示すアドレスが入っているフィールドのアドレス。プロキシ・サーバー名は、URL 形式で指定する必要があります。ユーザー出口プログラムへの入力時、このパラメーターは、2046 バイト領域のアドレスを含んだフィールドのアドレスに設定されます。この領域にプロキシ・サーバー名を入れ、UEPPROXY にあるアドレスは変更しないまま残すことができます。また、独自の領域にプロキシ・サーバー名を入れ、UEPPROXY にあるアドレスを、独自の領域のアドレスが含まれているフィールドのアドレスで置き換えることもできます。

### UEPPROXYL (ユーザー出口提供の出力)

プロキシ・サーバー名のハーフワードの長さが入っているフィールドのアドレス。

### UEPHOST (CICS 提供の入力)

UEPHOST パラメーターに含まれているホスト・タイプを示す 1 バイト・コードのアドレス。

注: 2 進数 01 はホスト名を、2 進数 02 は IPv4 アドレスを、2 進数 03 は IPv6 アドレスを示します。

## 戻りコード

### UERCNORM

この HTTP 要求には、プロキシ・サーバーは必要ではなく、ホスト名は制限されません。

### UERCPROX

この HTTP 要求にはプロキシ・サーバーが必要です。UEPPROXY は必要なプロキシ・サーバーの名前に設定され、UEPPROXYL はプロキシ・サーバー名の長さに設定されます。

### UERCBARR

そのサーバーのホスト名は制限されます。

### UERCERR

終了処理中にエラーが発生しました。

## XPI 呼び出し

すべての XPI 呼び出しが使用可能です。

## API コマンドおよび SPI コマンド

**EXEC CICS** コマンドは使用できません。

## HTTP クライアントの送信出口 XWBSNDO

XWBSNDO を使用して、HTTP クライアントとしての CICS による HTTP 要求に対するセキュリティ・ポリシーを指定できます。XWBSNDO は、**EXEC CICS WEB SEND** コマンドまたは **EXEC CICS WEB CONVERSE** コマンドの処理中に呼び出されます。ホスト名とパスの情報が出口に渡されます。セキュリティ・ポリシーは、これらの構成要素のどちらか、または両方に適用することができます。

CICS は、HTTP クライアントとしての CICS による HTTP 要求に対して、どのようなセキュリティ・ポリシーも適用しません。ご使用のシステムまたは組織でこの機能が必要な場合は、ユーザーがセットアップする必要があります。

WEB OPEN コマンドの XWBOPEN 出口を使用して、ホストへのアクセスを全面的に制限することができます。XWBSNDO 出口を使用して、同じ操作をすることも、ホスト内の特定のパスへのアクセスを制限することもできます。ホストへのアクセスを全面的に制限するには、XWBOPEN 出口を使用すると時間の節約になります。アプリケーション・プログラムは接続を開くことができないので、送信する必要のある要求を作成することに時間を浪費しないからです。異なるホストで使用されるまったく同じパスを区別できるように、XWBSNDO 出口にホスト名が提供されます。

HTTP 要求に対してチャック転送コーディングが使用されている場合、XWBSNDO は、そのチャック化されているメッセージ用の最初の WEB SEND コマンドでのみ呼び出されます。

XWBSNDO ユーザー出口は、**EXEC CICS** コマンドの使用をサポートしていません。

ホストは UEPHOST パラメーターとしてユーザー出口プログラムに渡され、パスは UEPPATH パラメーターとして渡されます。戻りコード UERCNORM はそのパスが許可されていることを示し、戻りコード UERCBARR はそのパスが許可されていないことを示します。パスが許可されていない場合は、WEB SEND



コマンドまたは WEB CONVERSE コマンドに NOTAUTH 応答が返されます。アプリケーション・プログラマーは、WEB CLOSE コマンドで接続を閉じることで、この応答に対処します。

## 出口 XWBSNDO

### 呼び出される状況

HTTP クライアントとしての CICS による HTTP 要求に対する、**EXEC CICS WEB SEND** コマンドまたは **EXEC CICS WEB CONVERSE** コマンドの処理中。クライアント要求は、WEB SEND コマンドで SESSTOKEN パラメーターを使用することで示されます。

### 出口固有のパラメーター

#### UEPHOST

ホスト名 (接続の WEB OPEN コマンドの HOST オプションで指定された IPv4 または IPv6 アドレス) が含まれるフィールドのアドレス。

注: ホスト名は、このフィールドに保管されるときに小文字に変換されます。ユーザー出口プログラムは、ホスト名とのマッチング時に、この変換を考慮に入れる必要があります。

#### UEPHOSTL

ホスト名のハーフワードの長さが入っているフィールドのアドレス。

#### UEPPATH

WEB SEND コマンドの PATH オプションで指定されたパスが入っているフィールドのアドレス。パスは、指定どおりの大/小文字混合です。

#### UEPPATHL

パスのハーフワードの長さが入っているフィールドのアドレス。

#### UEPHOSTT

UEPHOST パラメーターに含まれているホスト・タイプを示す 1 バイト・コードのアドレス。

注: 2 進数 01 はホスト名を、2 進数 02 は IPv4 アドレスを、2 進数 03 は IPv6 アドレスを示します。

### 戻りコード

#### UERCNORM

そのパスは許可されています。

#### UERCBARR

そのパスは許可されていません。

### XPI 呼び出し

すべての XPI 呼び出しが使用可能です。

### API コマンドおよび SPI コマンド

**EXEC CICS** コマンドは使用できません。

## システム間連絡プログラム出口 XISCONA、XISLCLQ、および XISQLCL

システム間連絡プログラムの 3 つの出口を使用して、システム間キューの長さを制御できます。

システム間キューの長さは、いくつかの方法で制御することができます。使用可能な方式の説明については、[システム間のセッション・キューの管理](#)を参照してください。

## XISCONA 出口

機能シップ要求または DPL 要求が非 IPIC 接続のための空きセッションを待っている間に要求側の領域でキューに入れられると、パフォーマンス上の問題が起こる可能性があります。XISCONA の目的は、このような問題の回避に役立つようにすることです。この出口を使用すると、キューに入れられる可能性のある機能シップ要求または DPL 要求をユーザーが拒否できるようになるため、未処理の ALLOCATE 要求の数を制御することができます。

**重要:** XISCONA でなく、XZIQUE 出口を z/OS Communications Server のワーキング・セット・モジュールで使用して、システム間キューの長さを制御してください。XZIQUE の方が、XISCONA より提供する機能も多く、一般的な使用に適しています (XZIQUE は、機能シップ、DPL、トランザクション・ルーティング、

および分散トランザクション処理の各要求に対して呼び出されますが、XISCONA は機能シップと DPL に対してしか呼び出されません)。両方の出口を使用可能にすると、機能シップおよび DPL 要求に対して XZIQUE と XISCONA の両方が呼び出される可能性があるため、両方の出口を使用可能にすることはお勧めしません。

XISCONA 出口プログラムが既にある場合は、これを修正することによって XZIQUE 出口点で利用できる場合があります。

「競合勝者」は、LU6.2 接続で使用される用語です。XISCONA 出口は、MRO 接続および LU6.1 接続にも適用されます。これらの接続では、SEND セッション (セッション定義で定義される) が最初に ALLOCATE 要求に使用されます。すべての SEND セッションが使用中になると、キューイングが始まります。

バインドされたすべてのコンテンション勝者セッションが使用中のために、すぐに使用できるセッションがない場合、リソース所有領域に対する機能シップ要求および DPL 要求はデフォルトでキューに入れられます。リソース所有領域からの応答が悪いと (例えば、ファイル所有領域である場合、システム・ジャーナルがアーカイブされるまで待機することがあります)、キューが長くなり、発行側領域のパフォーマンスが著しく損なわれます。さらに、要求側の領域がアプリケーション専有領域である場合は、パフォーマンスの低下は端末専有領域に及びます。

機能シップ要求および DPL 要求のキューイングを制御するには、XISCONA 出口を使用して、セッションの即時割り振りができなくなるたびに、要求をキューに入れるかどうか、要求をキューに入れるかどうか、またはアプリケーションに「SYSIDERR」を返すのかを CICS に通知します。この出口の仕組みは以下のとおりです。

1. XISCONA 出口プログラムがアクティブでない場合、CICS は必要に応じて要求をキューに入れます。
2. この出口プログラムがアクティブな場合、バインドされたすべてのコンテンション勝者セッションが使用中の場合のみ この出口プログラムが呼び出されます。他の障害 (例えば、「モード名が見つかりません」や「サービス休止」) の場合は、CICS は出口をバイパスしてアプリケーションに戻ります。
3. 出口プログラムは、呼び出されると、ユーザー出口パラメーター・リストで提供された統計を分析することによって、要求をキューに入れるかどうかを決定する必要があります。出口プログラムは以下を行うことができます。

- キューイングを使用しないように要求する。これは、出口をコーディングする最も簡単な方法で、調整の複雑さを回避できます。接続のピーク・トランザクション・ロードを処理するために十分なコンテンション勝者セッションを定義するには、これが効果的です。すべてのキューイングを抑止する場合、キューイング・メカニズムは以後はセッションをバインドしなくなるため、SESSIONS 定義で AUTOCONNECT(YES) を指定する必要があります。

この方法では、必要なセッション数を平均条件に基づいて見積もり、その後トランザクション・ロードが大きく変動した場合に、危険が生じます。CICS がキューイングを使用してその変動を処理できない場合、リソース所有領域に大きな問題がないのに、トランザクションが異常終了する可能性があります。

- 現在キューに入っている要求の数を調べてください。プログラムは、例えば、その数がセッションの最大数の 120% を超えたとき、キューイングを停止することができます。この方法を使用して、リソース所有領域での断続的な停止を処理することができます。

キューイングの問題に関する過去の経験から決定された値を示す、システム内の接続のしきい値の表を使用することができます。あるいは、別のプログラムの EXEC CICS インターフェースを使用して接続の状態について照会し、その情報を作業域に入れて XISCONA 出口プログラムに渡すこともできます。

- 要求のタイプおよびアクセス対象のリソースを調べてください (要求パラメーター・リストを調べることによって見つけることができます)。プログラムで、例えば、ファイル読み取り要求は拒否するが、ファイル更新はキューにいれることができます。

**注:** 出口プログラムの障害はシステムの可用性に影響を及ぼす可能性があるため、プログラムのロジックをできるだけ単純にしてエラーの可能性を軽減することをお勧めします。

XISCONA では解決できない問題がいくつかあります。例えば、多数のセッションと MXT の大きな値の両方を指定した場合、それ以上使用可能なセッションがなくなると、XISCONA が呼び出される前に、CICS でストレージ不足 (SOS) 状態が発生することがあります。

## XISCONA 用のグローバル・ユーザー出口のサンプル・プログラム

DFHXIS は、プログラムに渡される情報に基づいて ALLOCATE 要求のキューを制限する方法を示すサンプル出口プログラムです。詳しくは、[機能シップ/DPL キュー制御サンプル出口プログラム: DFHXIS](#) を参照してください。

### 出口

出口 XISCONA は、リモート領域にバインドされているコンテンション勝者セッションがすべて使用中であるために、機能シップまたは DPL 要求がキューに入れられようとしているときに呼び出されます。

### 呼び出される状況

リモート領域にバインドされているコンテンション勝者セッションがすべて使用中であるために、機能シップまたは DPL 要求がキューに入れられようとしているとき。

**注:** 動的ルーティングが行われる DPL 要求の場合は、動的ルーティング・プログラムが XISCONA の前に呼び出されます。空きセッションがない場合、ルーティング・プログラムは DPL 要求をキューに入れない場合があります。このような状況では、XISCONA は呼び出されません。DPL 要求の動的ルーティングに関する情報は、[DPL 要求の動的ルーティング](#) を参照してください。

### 出口固有のパラメーター

#### UEPISPCA

以下のフィールドを含むパラメーター・リストのアドレス。DSECT DFHXISDS を使用してパラメーター・リストをマップできます。

#### UEPCONST

接続統計レコードのアドレス。

接続統計レコードは、タイプ STICONSR (STID 値 52) のものです。出口プログラムは、DSECT DFHA14DS を使用してレコードをマップできます。注を参照してください。

#### UEPMODST

モード・エン트리統計レコードのアドレス、またはゼロ。モード・エン트리統計レコードは、以下の場合のみ作成されます。

- ・ 接続タイプが LU6.2 である (フィールド UEPCONTY を参照)。
- ・ プロファイル DFHCICSF (常に機能シップに使用される) が、LU6.2 セッションの割り振りに使用される特定の MODENAME を定義している。

モード・エン트리統計レコードは、タイプ STICONMR (STID 値 76) のものです。出口プログラムは、DSECT DFHA20DS を使用してレコード (存在する場合) をマップできます。

#### UEPEIPPL

要求パラメーター・リストのアドレス。

#### UEPCONTY

接続タイプを示す 1 バイト・フィールド。可能な値は次のとおりです。

##### UEPMRO (X'80')

MRO 接続の要求

##### UEPLU6 (X'40')

LU6.1 接続の要求

##### UEPLUC (X'20')

LU6.2 接続の要求

#### UEPNETNM

接続の NETNAME を含む 8 文字のフィールド。つまり、リモート CICS 領域またはシステムの ID (アプリケーション ID)。

### 注:

1. 統計レコードの一般的な形式は、[CICS 統計レコードの形式](#)で説明されています。
2. 統計レコード・タイプおよび関連したコピーブックのリストについては、[CICS 統計データ・セクション](#)を参照してください。

3. 接続統計レコードとモード・エン트리統計レコードのフィールドについては、[DSECTS および DFHSTUP レポートの CICS 統計](#)を参照してください。

## 戻りコード

### UERCAQUE

要求をキューに入れます。これはデフォルトです。

### UERCAPUR

ローカル・キューイングが可能でない限り、要求をキューに入れません。

## XPI 呼び出し

すべて使用できます。

## 重要

出口は障害後に呼び出されるため、この出口点において UERCNORM 戻りコードはありません。システムのデフォルト・アクションである要求のキューイングを行うか、行わないかを選択できます。

## XISLCLQ 出口

XISLCLQ は、非 IPIC 接続用にスケジュールされる EXEC CICS START NOCHECK 要求で使用します。

XISLCLQ は、有効になっていると、以下のいずれの状況でも呼び出されます。

- リモート・システムがサービス中でない。
- リモート・システムへの接続を確立できない。
- すぐに使用可能なセッションがなく、XISCONA 出口プログラムで、要求を発行側領域でキューに入れなことが指定されている。

この出口は送信される要求のタイプが EXEC CICS START NOCHECK である場合のみ呼び出されることに注意してください。NOCHECK オプション (START コマンドでのみ使用可能) が指定された要求以外の EXEC CICS 要求の場合は、アプリケーション・プログラムで SYSIDERR 条件が生成されます。

この出口を使用して、失敗した要求をローカル・キューに入れるか、接続が再確立されたときに実行するかを指定することができます。

システム再始動を実行すると、ローカル・キューがリカバリーされます。

## 出口 XISLCLQ

出口 XISLCLQ は、リモート・システムがサービス中でないか、リモート・システムへの接続を確立できないか、またはすぐに使用可能なセッションがなく、発行側の領域で要求をキューに入れずに XISCONA 出口プログラムで指定されているために、タイプ EXEC CICS START NOCHECK の機能シップ要求に失敗した後に呼び出されます。

## 呼び出される状況

リモート・システムがサービス中でないか、リモート・システムへの接続を確立できないか、またはすぐに使用可能なセッションがなく、発行側の領域で要求をキューに入れずに XISCONA 出口プログラムで指定されているために、タイプ EXEC CICS START NOCHECK の機能シップ要求に失敗した後。

## 出口固有のパラメーター

### UEPISPP

以下を含むパラメーター・リストのアドレス。

### UEPTCTSE

関連する端末管理テーブル・システム・エントリーのアドレス。TCT システム・エントリーは、DSECT DFHTCTTE を使用してマップできます。

### UEPXXTE

ローカル・トランザクション名のアドレス、または 0 (コマンドに SYSID が指定されていた場合)

**注:** プログラムでは、トランザクション・マネージャー XPI 呼び出し INQUIRE\_TRANDEF を使用して、ローカル・トランザクションの詳細を取得できます ([INQUIRE\\_TRANDEF 呼び出し](#)を参照)。

## UEPPLIST

コマンドのパラメーター・リストのアドレス。

**注:** このパラメーター・リストに DSECT は提供されていません。指定されたフィールドにアクセスするためには、独自の DSECT をコーディングする必要があります。

## 戻りコード

### UERC SYS

システム処置を行います。この処置は、リモート・トランザクションのローカル TRANSACTION 定義における、以下の LOCALQ 属性値によって決まります。

#### LOCALQ(YES)

要求はローカルでキューに入れられます。

#### LOCALQ(NO)

「SYSIDERR」がアプリケーション・プログラムに返されます。

### UERC QUE

要求をローカル・キューに入れます (指定されていた場合は、LOCALQ(NO) 属性をオーバーライドします)。

### UERC IGN

LOCALQ(YES) 属性をオーバーライドして (指定されていた場合)、「SYSIDERR」を返します。

### UERC PURG

タスクは XPI 呼び出し中にページされます。

## XPI 呼び出し

すべて使用できます。

## 重要

出口は失敗後に呼び出されるので、この出口点に「UERC NORM」戻りコードはありません。システムのデフォルト・アクションを実行するか、他の何らかの方法でエラーを処理するかを選択できます。

## XISQLCL 出口

XISQLCL 出口は、IPIC 接続にスケジュールされている EXEC CICS START NOCHECK コマンドに使用できます。

有効化されている場合、この出口は次のいずれかの場合に呼び出されます。

- IPIC 接続が獲得されなかった場合。
- セッションが使用できないため、CICS で新しいセッションの要求をキューに入れることができなかった場合。

XISQLCL を使用して、要求をローカル・キューに追加するかどうか、または要求をエラー応答とともに返すかどうかを指定することができます。

システム再始動を実行すると、ローカル・キューがリカバリーされます。

## サンプル XISQLCL 出口プログラム

DFHEXISL は、IPIC 接続用にスケジュールに入れられる START NOCHECK 要求のキューイングを制御するためのサンプル XISQLCL 出口プログラムです。DFHEXISL について詳しくは、[IPIC キュー制御サンプル出口プログラム: DFHEXISL](#) を参照してください。

## 出口 XISQLCL

出口 XISQLCL が呼び出されるのは、リモート・システムがサービス中でない、リモート・システムへの接続を確立できない、または直ちに使用可能なセッションがないことが理由で、IPIC を介した **START NOCHECK** または **START NOCHECK PROTECT** コマンドの機能シブ要求が失敗した後です (しかも発行側の領域で要求をキューに入れないよう XISQUE 出口プログラムが指定している場合です)。

## 出口固有のパラメーター

このパラメーター・リストのために DSECT の DFHXILDS が設けられます。

**UEPISQPL**

以下のフィールドを格納するパラメーター・リストのアドレスです。

**UEPPLIST**

コマンドのパラメーター・リストのアドレス。

**UEPQLEN**

キューに現在ある項目の数を格納するハーフワード 2 進数フィールド。

**UEPIPCNM**

IPCONN の 8 バイトの名前。

**UEPTRID**

ローカル・トランザクション名の 4 バイトの ID、またはブランク (コマンドに SYSID を指定した場合)。プログラムでは、トランザクション・マネージャー XPI 呼び出しの INQUIRE\_TRANDEF を使用して、ローカル・トランザクションの詳細を取得できます。  
INQUIRE\_TRANDEF 呼び出しを参照してください。

**戻りコード****UERC SYS**

システム処置を行います。この処置は、リモート・トランザクションのローカル TRANSACTION 定義における、以下の LOCALQ 属性値によって決まります。

**LOCALQ(YES)**

要求はローカルでキューに入れられます。

**LOCALQ(NO)**

SYSIDERR エラー・メッセージがアプリケーション・プログラムに返されます。

**UERC QUE**

指定すると、要求をローカルでキューに入れて、LOCALQ(NO) 属性をオーバーライドします。

**UERC IGN**

指定すると、LOCALQ(YES) 属性をオーバーライドし、SYSIDERR 応答を返します。

**UERC PURG**

タスクは XPI 呼び出し中にパージされます。

**XPI 呼び出し**

すべて使用できます。

**重要**

出口は障害後に呼び出されるため、この出口点において UERCNORM 戻りコードはありません。システム・デフォルトのアクションを取るか、または別の方法でエラーを処置するかを決める必要があります。

XISQLCL グローバル・ユーザー出口プログラムのサンプルとして DFH\$XISL が用意されています。

**インターバル制御プログラム出口 XICREQ、XICEXP、および XICTENF**

インターバル制御プログラムから呼び出される出口プログラムで、一部の XPI 呼び出しを使用することができます。ただし、これらの出口のいずれかが満了分析のために呼び出された場合、インターバル制御プログラムの実行を遅らせるアクションは、間隔の満了を待っている他のトランザクションに悪影響を及ぼす可能性があります。

要求タイプ・フィールド TCAICTR (このコピーが UEPICRQ1 出口固有パラメーターによって指示されている) を調べることによって、満了分析のために出口が呼び出されたかどうかを判別することができます。

XICREQ 出口は、CICS コードで作成された内部要求と、アプリケーションで作成された要求によって呼び出されます。DFHXRSP は、インターバル制御 WAIT を 2 秒ごとに発行します。これは、どのインターバル制御出口プログラムでも 2 秒ごとに呼び出されることを意味します。



## 出口 XICREQ

この出口は、インターバル制御プログラム開始時に、要求分析の前に呼び出されます。

### 呼び出される状況

インターバル制御プログラム開始時の要求分析の前。

### 出口固有のパラメーター

#### UEPICQID

要求の要求 ID パラメーターが含まれた 8 バイト・フィールドのアドレス。注 1 および注 2 を参照してください。

#### UEPICTID

EXEC CICS START コマンドに指定された端末 ID が存在する場合に、その端末 ID が含まれた 4 バイト・フィールドのアドレス。注 1 および注 2 を参照してください。

#### UEPICTI

EXEC CICS START コマンドに指定されたトランザクション ID が含まれた 4 バイトのアドレス。注 1 および注 2 を参照してください。

#### UEPICRQ1

TCAICTR (インターバル制御プログラムへの要求の最初の要求コード・フィールド) のコピーが含まれた 1 バイト・フィールドのアドレス。

#### UEPICRQ2

TCAICTR2 (インターバル制御プログラムへの要求の 2 番目の要求コード・フィールド) のコピーが含まれた 1 バイト・フィールドのアドレス。

#### UEPICRT

有効期限時刻または間隔がパック 10 進数形式で含まれている 4 バイト・フィールドのアドレス。この値は OHHMMSSF の形式です。ここで、H= 時、M= 分、S= 秒、F は正記号です。

### 注:

1. UEPICQID および UEPICTID によってアドレス指定されたフィールドの内容は、関連データ項目が要求に指定されていない場合は予測不能です。TCAICTR のコピーをテストして、意味のある値が含まれているかどうかを判断する必要があります。
2. 出口プログラムで、UEPICQID、UEPICTID、UEPICTI、および UEPICRT によってアドレス指定されたフィールドの値を変更することができます。UEPICRQ1 または UEPICRQ2 によってアドレス指定されたフィールドの値を変更しても効果はありません。

### 戻りコード

#### UERCNORM

処理が続行される。

#### UERCPURG

タスクは XPI 呼び出し中にページされます。

### XPI 呼び出し

以下は使用しないでください。

- ADD\_SUSPEND
- DELETE\_SUSPEND
- DEQUEUE
- ENQUEUE
- RESUME
- SUSPEND
- WAIT\_MVS

## 出口 XICEXP

この出口は、インターバル制御時間間隔が満了した後で呼び出されます。

### 呼び出される状況

インターバル制御時間間隔が満了した後。

### 出口固有のパラメーター

#### UEPICE

満了したばかりのインターバル制御エレメント (ICE) のアドレス。DSECT DFHICEDS を使用して ICE をマップすることができます。

### 戻りコード

#### UERCNORM

処理が続行される。

#### UERCPURG

タスクは XPI 呼び出し中にページされます。

### XPI 呼び出し

以下は使用しないでください。

- ADD\_SUSPEND
- DELETE\_SUSPEND
- DEQUEUE
- ENQUEUE
- RESUME
- SUSPEND
- WAIT\_MVS

## 出口 XICTENF

この出口は、インターバル制御プログラムから呼び出されます。

### 呼び出される状況

この出口は「端末未認識」状態に関連しています。詳しくは、[216 ページの『「端末未認識」状態出口 XALTENF および XICTENF』](#)を参照してください。

## インターバル制御 EXEC インターフェース・プログラム出口 (XICEREQ、XICERES、および XICEREQC)

これらの出口は、インターバル制御プログラムとの相互作用が発生したときに呼び出されます。

### XICEREQ

XICEREQ は、CICS がインターバル制御機能要求を処理する前に、インターバル制御機能プログラムへの入力時に呼び出されます。XICEREQ を使用することにより、ユーザーは以下の処理ができます。

- 要求を分析して、そのタイプ、指定するキーワード、および値を決定する。
- コマンドが実行される前に、要求により指定される任意の値を変更する。
- 戻りコードを設定して、次のいずれかを指定する。
  - CICS は修正された要求または、修正されない要求をそのまま継続する。
  - CICS は要求を迂回する。(この戻りコードを設定すると、自分自身で要求を処理したかのように、EXEC インターフェース・ブロック (EIB) の戻りコードも設定する必要があることに注意してください。)

注：XICEREQ 出口は、CICS コードで作成された内部要求と、アプリケーションで作成された要求により、呼び出されます。



## XICERES

XICERES は、CICS がこの領域に動的にルーティングされた非端末関連 **EXEC CICS START** 要求を処理する前に、インターバル制御プログラムにより呼び出されます。

XICERES は以下の場合に呼び出されることに注意してください。

- 出口 XICEREQ の後および XICEREQC の前 (これらの出口が使用可能になっている場合)。これは、以下のことを意味します。
  - XICEREQ 出口プログラムが要求を迂回することを選択した場合、XICERES は使用可能になっていても呼び出されません。
  - XICEREQ 出口プログラムが要求を変更した場合、XICERES は修正された要求を処理する必要があります。
- ターゲット 領域上、つまり、START 要求がルーティングされた領域。
- ルーティング領域 (ルーティング・プログラムが実行されている領域) が「リソース使用不可」状態 (RESUNAVAIL) をサポートしている場合のみ。「リソース使用不可」状態をサポートするには、ルーティング領域が CICS TS のサポートされるリリースである必要があります。
- 使用可能になっている場合のみ。非端末関連 **EXEC CICS START** 要求が動的にルーティングされる可能性のあるアプリケーション専有領域でのみ、この出口を使用可能にすることを強くお勧めします。
- CICS コードで作成された内部要求と、アプリケーションで作成された要求による場合。

XICERES 出口は以下の場合、呼び出されません。

- 静的ルーティング要求の場合。
- 端末関連 **EXEC CICS START** 要求の場合。(これらは常に端末専有領域で実行され、ルーティングできません。)
- 動的ルーティングされたトランザクションの場合。この場合、動的ルーティングされた (非端末関連) **START** 要求によって出口が呼び出されます。すなわち、端末関連 **EXEC CICS START** コマンドにより開始され、動的ルーティングされたトランザクションは、出口を呼び出しません。
- 使用不可の場合。
- XICEREQ 出口プログラムが要求を迂回することを選択した場合。

XICERES を使用して、開始するトランザクションに必要なすべてのリソースが、ターゲット領域で使用可能であることを確認することができます。例えば、トランザクションが使用不可の場合、または必要なファイルがない場合は、出口プログラムは分散ルーティング・プログラムに対し、要求を別の領域にルーティングする機会を与えることができます。これを行うには、戻りコード UERCRESU を設定します。こうすることにより、CICS は次のことを実行します。

1. 分散ルーティング・プログラムの通信域の DYRERROR フィールドを「F」(リソース使用不可) に設定します。
2. 経路選択に失敗した場合、ルーティング領域上でルーティング・プログラムを再度呼び出します。
3. ミラーによりターゲット領域に対して実行された **EXEC CICS START** コマンドの RESUNAVAIL 状態を返します。(この状態はアプリケーション・プログラムには返されません。)

CICS は、出口パラメーターの任意の値に対して出口プログラムにより加えられたすべての変更を無視します。出口プログラムは戻りコードを設定できますが、どのパラメーターも変更しません。

非端末関連 **EXEC CICS START** 要求の動的ルーティングのガイダンス情報については、[非端末関連の START コマンド](#)を参照してください。非端末関連 **EXEC CICS START** 要求をルーティングする分散ルーティング・プログラムの作成については、[非端末関連 START 要求のルーティング](#)を参照してください。

## XICEREQC

XICEREQC は、インターバル制御機能プログラム要求が完了してから呼び出されます。XICEREQC を使用することにより、ユーザーは以下の処理ができます。

- 要求を分析して、そのタイプ、指定するキーワード、および値を決定する。

- EIB の戻りコードを設定する。戻りコードを調べる際は、EIBRCODE 値を参照します。

注：XICEREQC 出口は、CICS コードで作成された内部要求と、アプリケーションで作成された要求により呼び出されます。

## それぞれの出口に渡されるパラメーター

CICS は、10 種類のアドレス・パラメーターを出口に渡します。

- コマンド・レベル・パラメーター構造 (UEPCLPS) のアドレス
- XICEREQ から XICEREQC に 4 バイトのデータを渡すために使用されるトークン (UEPICTOK) のアドレス
- EIB に格納されている 6 つの戻りコード・セグメント、リソース、日付、および時刻情報のコピーのアドレス
- タスクの存続中に有効なトークン (UEPTSTOK) のアドレス
- 出口再帰カウント (UEPRECUR) のアドレス

## 出口 XICEREQ

XICEREQ を使用することにより、要求を分析してそのタイプ、指定されているキーワード、および値を判別すること、要求で指定された値をコマンドの実行前に変更すること、および CICS が要求を続行すべきか、または迂回すべきかを指定する戻りコードを設定することができます。

### 出口が呼び出される時点

CICS がインターバル制御 API 要求を処理する前の、インターバル制御プログラムへの入力時。

### 出口固有のパラメーター

#### UEPCLPS

コマンド・レベルのパラメーター構造のアドレス。[135 ページの『UEPCLPS 出口固有パラメーター』](#)を参照してください。

#### UEPICTOK

XICEREQC に渡される 4 バイトのトークンのアドレス。これにより、例えば、作業域を出口の XICEREQC に渡すことができます。

#### UEPRCODE

EIB 戻りコード「EIBRCODE」の 6 バイトの 16 進数コピーのアドレス。EIB 戻りコードの詳細については、[EIB フィールド](#)を参照してください。

#### UEPRES

EIB 応答コード「EIBRESP」の 4 バイトのバイナリー・コピーのアドレス。

#### UEPRES2

EIB 応答コード「EIBRESP2」の 4 バイトのバイナリー・コピーのアドレス。

#### UEPTSTOK

タスクの存続中に有効な 4 バイトのトークンのアドレス。[タスク・トークン UEPTSTOK の使用](#)を参照してください。

#### UEPRECUR

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

#### UEPRSRCE

EIB リソース値の EIBRSRCE の 8 文字のコピーのアドレス。

#### UEPDATE

EIB 日付値 EIBDATE のフルワード・コピーのアドレス。

#### UEPTIME

EIB 時刻値 EIBTIME のフルワード・コピーのアドレス。

### 戻りコード

#### UERCNORM

処理が続行される。

## **UERCBYP**

インターバル制御 EXEC インターフェース・プログラムはこの要求を無視します。

## **UERCPURG**

タスクは XPI 呼び出し中にページされます。

## **XPI 呼び出し**

すべてを使用できます。このユーザー出口で EXEC CICS API コマンドを使用することもできます。

この出口では XPI GETMAIN 呼び出しおよび FREEMAIN 呼び出しの使用が許可されていますが、代わりに EXEC CICS GETMAIN コマンドおよび FREEMAIN コマンドを使用することをお勧めします。

## **API コマンドおよび SPI コマンド**

次のコマンドを除き、すべてを使用できます。

### **EXEC CICS SHUTDOWN**

### **EXEC CICS XCTL**

注：再帰的コマンドを発行する際には、ループが生じないように注意してください。例えば、XICEREQ 出口からインターバル制御要求を発行する場合に、ループに入らないようにユーザーが注意する必要があります。再帰カウンタ UEPRECUR を使用することをお勧めします。

## **出口 XICERES**

出口 XICERES は、インターバル制御プログラムによって呼び出されます。XICERES を使用して、開始するトランザクションに必要なすべてのリソースが、ターゲット領域で使用可能であることを確認することができます。

### **出口が呼び出される時点**

ルーティング領域が「リソース使用不可」(RESUNAVAIL) 状態をサポートしている場合に、この領域に動的にルーティングされている非端末関連 **EXEC CICS START** 要求の処理の前。

### **出口固有のパラメーター**

注：CICS は、出口パラメーターの任意の値に対して出口プログラムにより加えられたすべての変更を無視します。出口プログラムは戻りコードを設定できますが、どのパラメーターも変更しません。

### **UEPCLPS**

コマンド・レベルのパラメーター構造のアドレス。135 ページの『UEPCLPS 出口固有パラメーター』を参照してください。

### **UEPICTOK**

XICEREQC に渡される 4 バイトのトークンのアドレス。

### **UEPRCODE**

EIB 戻りコード「EIBRCODE」の 6 バイトの 16 進数コピーのアドレス。EIB 戻りコードについて詳しくは、[EIB フィールド](#)を参照してください。

### **UEPRES**

EIB 応答コード「EIBRESP」の 4 バイトのバイナリー・コピーのアドレス。

### **UEPRES2**

EIB 応答コード「EIBRESP2」の 4 バイトのバイナリー・コピーのアドレス。

### **UEPTSTOK**

タスクの存続中に有効な 4 バイトのトークンのアドレス。[タスク・トークン UEPTSTOK の使用](#)を参照してください。

### **UEPRECUR**

ハーフワードの再帰カウンタのアドレス。XICERES 出口は同じトランザクションで再帰的に呼び出すことができないため、このフィールドの値は常に 0 です。

### **UEPRSRCE**

EIB リソース値の EIBRSRCE の 8 文字のコピーのアドレス。

### **UEPDATE**

EIB 日付値 EIBDATE のフルワード・コピーのアドレス。

## **UEPTIME**

EIB 時刻値 EIBTIME のフルワード・コピーのアドレス。

## 戻りコード

### **UERCNORM**

処理が続行される。

### **UERPURG**

タスクは XPI 呼び出し中にパージされます。

### **UERCRESU**

必要なリソースが使用不可です。この値を設定すると、CICS がルーティングされた要求を拒否し、ルーティング・プログラムの通信域の DYRERROR フィールドに値「F」(リソース使用不可)を返すことになります。

## **XPI 呼び出し**

すべてを使用できます。このユーザー出口で EXEC CICS API コマンドを使用することもできます。

この出口では XPI GETMAIN 呼び出しおよび FREEMAIN 呼び出しの使用が許可されていますが、代わりに EXEC CICS GETMAIN コマンドおよび FREEMAIN コマンドを使用することをお勧めします。

## **API コマンドおよび SPI コマンド**

**EXEC CICS SHUTDOWN** および **EXEC CICS XCTL** を除くすべてを使用できます。

## 出口 **XICEREQC**

XICEREQC を使用して、要求を分析し、そのタイプ、指定されているキーワード、および値を判別できます。また、EIB の戻りコードを設定することもできます。

戻りコードを調べる際は、EIBRCODE 値を参照します。詳細については、[145 ページの『EXEC インターフェース・ブロック \(EIB\)』](#)を参照してください。

## 出口が呼び出される時点

インターバル制御 API 要求が完了した後で、インターバル制御 EXEC インターフェース・プログラムから戻る前。

## 出口固有のパラメーター

### **UEPCLPS**

コマンド・レベルのパラメーター構造のアドレス。[135 ページの『UEPCLPS 出口固有パラメーター』](#)を参照してください。

### **UEPICTOK**

XICEREQC から渡された 4 バイトのトークンのアドレス。これにより XICEREQC は、例えば、作業域を XICEREQC に渡すことができます。

### **UEPRCODE**

EIB 戻りコード「EIBRCODE」の 6 バイトの 16 進数コピーのアドレス。EIB 戻りコードについて詳しくは、[EIB フィールド](#)を参照してください。

### **UEPRES P**

EIB 応答コード「EIBRESP」の 4 バイトのバイナリー・コピーのアドレス。

### **UEPRES P2**

EIB 応答コード「EIBRESP2」の 4 バイトのバイナリー・コピーのアドレス。

### **UEPTSTOK**

タスクの存続中に有効な 4 バイトのトークンのアドレス。[タスク・トークン UEPTSTOK の使用](#)を参照してください。

### **UEPRECUR**

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとに増加します。

### **UEPRSRCE**

EIB リソース値の EIBRSRCE の 8 文字のコピーのアドレス。

### **UEPDATE**

EIB 日付値 EIBDATE のフルワード・コピーのアドレス。

## **UEPTIME**

EIB 時刻値 EIBTIME のフルワード・コピーのアドレス。

## **UEP\_IC\_REMOTE\_SYSTEM**

要求をリモート領域に送信する場合は、そのリモート領域の 4 バイトの名前が含まれた領域のアドレス。(リモート領域は、例えば、START コマンドの SYSID オプション、ワークロード管理、または TRANSACTION 定義の REMOTESYSTEM オプションで指定された可能性があります。)

要求をローカル領域で実行する場合、このパラメーターはブランクが含まれた 4 バイト領域のアドレスです。

## **UEP\_IC\_REMOTE\_NAME**

トランザクションをリモート・システムで実行する場合、リモート・システムで認識されているおりのトランザクションの名前が含まれた領域のアドレス。

リモート・システムは、別の CICS 領域の場合もあり、IMS システムの場合もあります。

UEP\_IC\_REMOTE\_SYSTEM が CICS 領域を示している場合、この名前は 1 文字から 4 文字の長さになります。UEP\_IC\_REMOTE\_SYSTEM が IMS システムを示している場合、この名前は 1 文字から 8 文字の長さになります。IMS は 8 文字の名前を使用します。UEP\_IC\_REMOTE\_NAME が 8 文字未満の場合、IMS はそれを使用可能な形式に変換します。

## 戻りコード

### **UERCNORM**

処理が続行される。

### **UERCPURG**

タスクは XPI 呼び出し中にページされます。

## **XPI 呼び出し**

すべてを使用できます。

この出口では XPI GETMAIN 呼び出しおよび FREEMAIN 呼び出しの使用が許可されていますが、代わりに EXEC CICS GETMAIN コマンドおよび FREEMAIN コマンドを使用することをお勧めします。

## **API コマンドおよび SPI コマンド**

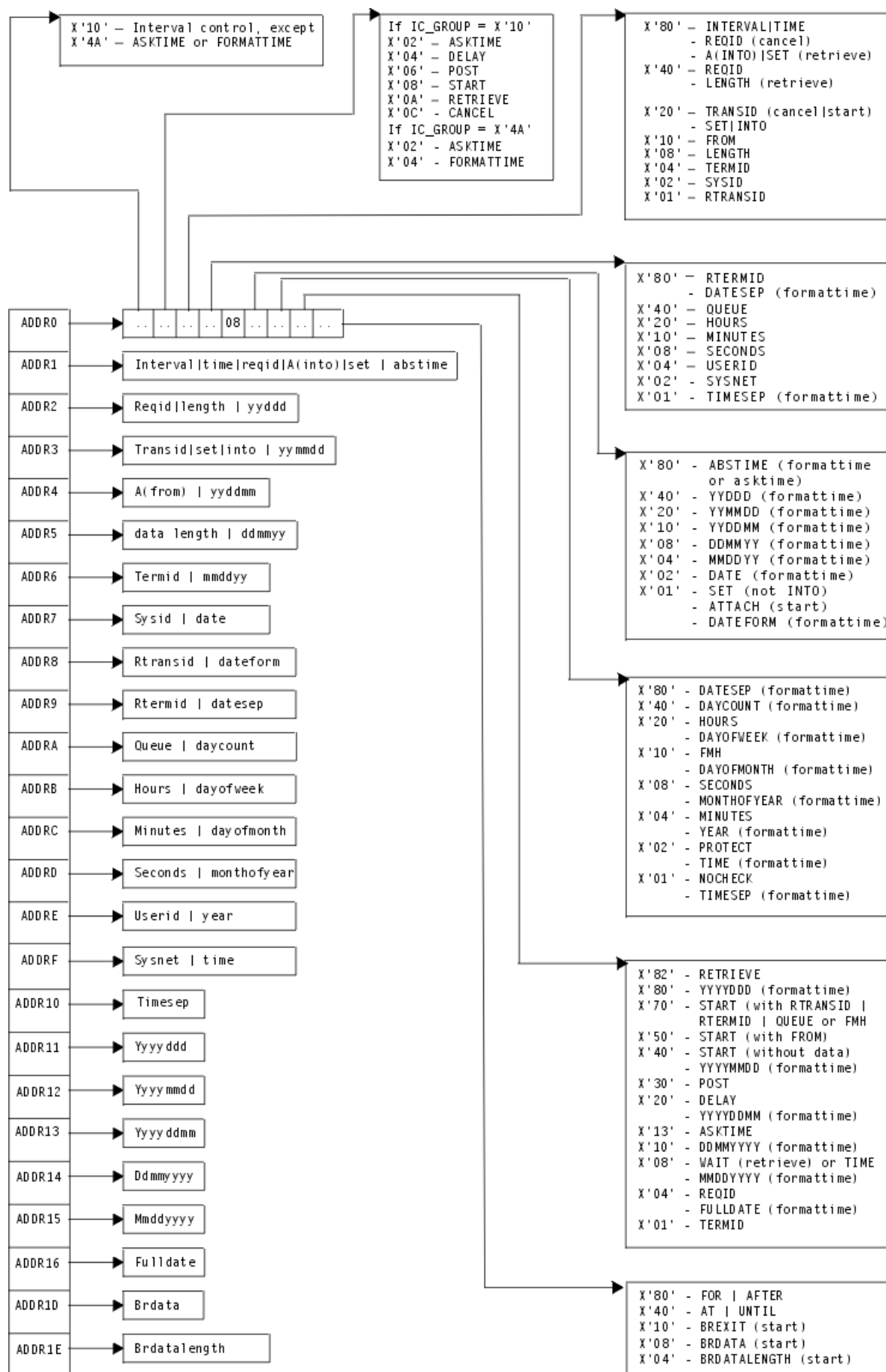
次のコマンドを除き、すべてを使用できます。

**EXEC CICS SHUTDOWN**

**EXEC CICS XCTL**

**注：**再帰的コマンドを発行するときには注意してください。例えば、XICEREQC 出口からインターバル制御要求を発行するときには、ループに入らないようにする必要があります。再帰カウンタ UEPPRECUR を使用することをお勧めします。

## コマンド・レベルのパラメーター構造



コマンド・レベルのパラメーター構造は、一連のアドレスで構成されます。最初のアドレスは、EXEC インターフェース記述子 (EID) を指します。これは、要求のタイプを記述する 9 バイト領域からなり、要求で指定された各キーワードを識別します。残りのアドレスは、その要求に関連付けられたデータ部分を指します。例えば、2 番目のアドレスは START 要求の間隔を指します。

EID を調べて、要求のタイプおよび指定されたキーワードを判別することができます。リスト内のその他のパラメーターを調べて、キーワードの値を判別することができます。要求で指定されたキーワードの値を変更することもできます。例えば、要求で指定された SYSID を変更できます。

### パラメーター・リスト終了標識

パラメーター・リスト内の最後のアドレス・セットの高位ビットが、リスト内の最後のアドレスであることを示すために設定されます。ユーザー出口プログラムから戻ったときに、CICS は、パラメーター・リストをスキャンして高位ビットを探し、最終パラメーターを検出します。したがって、パラメーター・リストの長さを変更した場合は、高位ビットをリセットしてどれが新しい最終アドレスかを示す必要もあります。

例えば、パラメーター・リストで最初の 4 つのアドレスのみ (EID のアドレスである IC\_ADDR0 から START 要求で指定されたトランザクションの名前のアドレスである IC\_ADDR3 まで) を指定した場合、IPC\_ADDR3 内の高位ビットがオンに設定されます。IC\_ADDR7 に SYSID のアドレスを設定してパラメーター・リストを拡張した場合、IC\_ADDR3 の高位ビットを設定解除し、代わりに IC\_ADDR7 の高位ビットをオンに設定する必要があります。

パラメーター・リストの最大サイズを出口に対して指定します。これによって、出口プログラムは、事前に追加のストレージを取得せずに、まだ指定されていないパラメーターを追加することができます。

XICEREQC の完了後に、XICEREQ が呼び出される前の元のパラメーター・リストが復元されます。その後に、実行診断機能 (EDF) により、実行前および実行後の元のコマンドが表示されます。**EDF は、出口によって行われた変更は表示しません。**

### UEPCLPS 出口固有パラメーター

UEPCLPS 出口固有パラメーターは、出口 XICEREQ と出口 XICEREQC の両方に組み込まれています。このパラメーターは、コマンド・レベルのパラメーター構造のアドレスです。

コマンド・レベルのパラメーター構造には、IC\_ADDR0 から IC\_ADDR1F までの 26 のアドレスが含まれています。この構造は DSECT IC\_ADDR\_LIST で定義されます。DSECT IC\_ADDR\_LIST は、COPY DFHICUED ステートメントを組み込むことによって出口プログラムにコピーする必要があります。

コマンド・レベルのパラメーター・リストは、以下のように構成されています。

#### IC\_ADDR0

EXEC インターフェース記述子 (EID) と呼ばれる 9 バイトの領域のアドレスであり、以下のように構成されています。

- IC\_GROUP
- IC\_FUNC
- IC\_BITS1
- IC\_BITS2
- IC\_BITS3
- IC\_EIDOPT5
- IC\_EIDOPT6
- IC\_EIDOPT7
- IC\_EIDOPT8

#### IC\_GROUP

X'10'

これは、インターバル制御要求です。

**X'4A'**

これは、ASKTIME コマンドまたは FORMATTIME コマンドです。

**IC\_FUNCT**

以下の要求のタイプを定義する 1 バイトの領域です。

IC\_GROUP = X'10' の場合:

**X'02'**

ASKTIME

**X'04'**

DELAY

**X'06'**

POST

**X'08'**

START

**X'0A'**

RETRIEVE

**X'0C'**

CANCEL

IC\_GROUP = X'4A' の場合:

**X'02'**

ASKTIME

**X'04'**

FORMATTIME

**IC\_BITS1**

どの引数が指定されたかを定義する存在ビット。キーワードに関連付けられた引数を取得するには、コマンド・レベルのパラメーター構造から適切なアドレスを使用する必要があります。このアドレスを使用する前に、関連付けられた存在ビットを確認する必要があります。存在ビットがオフに設定されている場合は、引数が要求に指定されていないため、このアドレスを使用してはなりません。

**X'80'**

要求に INTERVAL 引数または TIME 引数が含まれている場合、CANCEL 要求に REQID を指定している場合、または RETRIEVE 要求に SET または INTO を指定している場合に設定されます。設定された場合は、**IC\_ADDR1** が意味を持ちます。

**X'40'**

CANCEL 以外の要求に REQID を指定している場合、または RETRIEVE 要求に LENGTH を指定している場合に設定されます。設定された場合は、**IC\_ADDR2** が意味を持ちます。

**X'20'**

要求に TRANSID を指定している場合、または RETRIEVE 以外の要求に SET または INTO を指定している場合に設定されます。設定された場合は、**IC\_ADDR3** が意味を持ちます。

**X'10'**

要求に FROM を指定している場合に設定されます。設定された場合は、**IC\_ADDR4** が意味を持ちます。

**X'08'**

RETRIEVE 以外の要求に LENGTH が指定されている場合に設定されます。設定された場合は、**IC\_ADDR5** が意味を持ちます。

**X'04'**

要求に TERMID を指定している場合に設定されます。設定された場合は、**IC\_ADDR6** が意味を持ちます。

**X'02'**

要求に SYSID が指定されているかどうかを設定します。設定された場合は、**IC\_ADDR7** が意味を持ちます。



**X'01'**

要求に RTRANSID を指定している場合に設定されます。設定された場合は、**IC\_ADDR8** が意味を持ちます。

**IC\_BITS2**

後続の引数存在ビット。

**X'80'**

要求に RTERMID を指定している場合、または FORMATTIME 要求に DATESEP を指定している場合に設定されます。設定された場合は、**IC\_ADDR9** が意味を持ちます。

**X'40'**

要求に QUEUE を指定している場合に設定されます。設定された場合は、**IC\_ADDRA** が意味を持ちます。

**X'20'**

要求に HOURS を指定している場合に設定されます。設定された場合は、**IC\_ADDRB** が意味を持ちます。

**X'10'**

要求に MINUTES を指定している場合に設定されます。設定された場合は、**IC\_ADDRC** が意味を持ちます。

**X'08'**

要求に SECONDS を指定している場合に設定されます。設定された場合は、**IC\_ADDRD** が意味を持ちます。

**X'04'**

要求に USERID を指定している場合に設定されます。設定された場合は、**IC\_ADDRE** が意味を持ちます。

**X'02'**

要求に SYSNET を指定している場合に設定されます。設定された場合は、**IC\_ADDRF** が意味を持ちます。

**X'01'**

FORMATTIME 要求に TIMESEP を指定している場合に設定されます。設定された場合は、**IC\_ADDR10** が意味を持ちます。

**IC\_BITS3**

インターバル制御機能によって使用されない 1 バイト。

**IC\_EIDOPT5**

特定のキーワードが要求に指定されたかどうかを示します。

**X'80'**

FORMATTIME コマンドまたは ASKTIME コマンドで ABSTIME が指定されています。

**X'40'**

FORMATTIME コマンドで YYDDD が指定されています。

**X'20'**

FORMATTIME コマンドで YYMMDD が指定されています。

**X'10'**

FORMATTIME コマンドで YYDDMM が指定されています。

**X'08'**

FORMATTIME コマンドで DDMMYY が指定されています。

**X'04'**

FORMATTIME コマンドで MMDDYY が指定されています。

**X'02'**

FORMATTIME コマンドで DATE が指定されています。

**X'01'**

RETRIEVE コマンドで、SET が指定されています (INTO の指定はなし)。START コマンドでは、ATTACH が指定されています。FORMATTIME コマンドでは、DATEFORM が指定されています。このフィールドをユーザー 出口で変更することはできません。

## **IC\_EIDOPT6**

特定のキーワードが要求に指定されたかどうかを示す存在ビット。

**X'80'**

FORMATTIME コマンドで DATESEP が指定されています。

**X'40'**

FORMATTIME コマンドで DAYCOUNT が指定されています。

**X'20'**

FORMATTIME コマンドで DAYOFWEEK が指定されているか、HOURS が指定されています。

**X'10'**

FORMATTIME コマンドで DAYOFMONTH が指定されているか、FMH が指定されています。

**X'08'**

FORMATTIME コマンドで MONTHOFYEAR が指定されているか、SECONDS が指定されています。

**X'04'**

FORMATTIME コマンドで YEAR が指定されているか、MINUTES が指定されています。

**X'02'**

FORMATTIME コマンドで TIME が指定されているか、PROTECT が指定されています。

**X'01'**

FORMATTIME コマンドで TIMESEP が指定されているか、NOCHECK が指定されています。

## **IC\_EIDOPT7**

特定の関数またはキーワードが要求に指定されたかどうかを示します。

**X'F0'**

CANCEL が指定されています。

**X'82'**

RETRIEVE が指定されています。

**X'80'**

FORMATTIME コマンドで YYYYDD が指定されています。

**X'40'**

FORMATTIME コマンドで YYYYMMDD が指定されているか、START が指定されています。

**X'30'**

POST が指定されています。

**X'20'**

FORMATTIME コマンドで YYYYDDMM が指定されているか、DELAY、RTRANSID、RTERMID、または QUEUE が指定されているか、あるいは FMH (またはその両方) が指定されています。

**X'13'**

ASKTIME が指定されています。

**X'10'**

FORMATTIME コマンドで DDMMYYYY が指定されているか、FROM、RTRANSID、または RTERMID が指定されているか、あるいは QUEUE (またはその両方) が指定されています。

**X'08'**

FORMATTIME コマンドで MMDDYYYY が指定されているか、TIME または WAIT が指定されています。

**X'04'**

FORMATTIME コマンドで FULLDATE が指定されているか、REQID が指定されています。

**X'01'**

TERMID が指定されています。

## **IC\_EIDOPT8**

特定のキーワードが要求に指定されたかどうかを示します。

**X'80'**

FOR または AFTER が指定されています。

**X'40'**

AT または UNTIL が指定されています。

**X'10'**

BREXIT が指定されています。

**X'08'**

BRDATA が指定されています。

**X'04'**

BRDATALENGTH が指定されています。

**X'02'**

START コマンドで CHANNEL が指定されています。

**IC\_ADDR1**

次のアドレスのいずれかです。

- INTERVAL キーワードの値が含まれた 8 バイトの領域 (または、TIME が指定されていることが **IC\_EIDOPT7** で示されている場合は、TIME キーワード)。
- REQID の値が含まれた 8 バイトの領域 (要求が CANCEL の場合)。
- ABSTIME キーワードの値が含まれた 8 バイトの領域。
- INTO に返されたデータ (要求が RETRIEVE であって、かつ **IC\_EIDOPT5** がこれが SET でないことを示している場合)。
- SET に返された 4 バイトのアドレス (要求が RETRIEVE であって、かつ **IC\_EIDOPT5** がこれが SET であることを示す場合)。

**IC\_ADDR2**

次のアドレスのいずれかです。

- REQID の値が含まれた 8 バイトの領域 (要求が DELAY、POST、または START の場合)。
- LENGTH の値が含まれたハーフワード (要求が RETRIEVE の場合)。

**警告:** INTO を指定する要求の場合は、LENGTH の値をアプリケーションで指定された値よりも大きい値に変更しないでください。この値を大きい値に変更すると、アプリケーション内でストレージ・オーバーレイが発生します。

- YYDD の値が含まれた領域。

**IC\_ADDR3**

次のアドレスのいずれかです。

- TRANSID の値が含まれた領域 (要求が CANCEL または START の場合)。
- SET に返された 4 バイトのアドレス (要求が START または POST であって、かつ **IC\_EIDOPT5** がこれが SET であることを示す場合)。
- YYMMDD の値が含まれた領域。

**IC\_ADDR4**

次のアドレスのいずれかです。

- FROM からのデータが含まれた領域。
- YYDDMM の値が含まれた領域。

**IC\_ADDR5**

次のアドレスのいずれかです。

- LENGTH のハーフワード値が含まれた領域。

**警告:** INTO を指定する要求の場合は、LENGTH の値をアプリケーションで指定された値よりも大きい値に変更しないでください。この値を大きい値に変更すると、アプリケーション内でストレージ・オーバーレイが発生します。

- DDMMYY の値が含まれた領域。

**IC\_ADDR6**

次のアドレスのいずれかです。

- TERMID の値が含まれた領域。
- MMDDYY の値が含まれた領域。

**IC\_ADDR7**

次のアドレスのいずれかです。

- SYSID の値が含まれた領域。
- DATE の値が含まれた領域。

**IC\_ADDR8**

次のアドレスのいずれかです。

- RTRANSID の値が含まれた領域。
- DATEFORM の値が含まれた領域。

**IC\_ADDR9**

次のアドレスのいずれかです。

- RTERMID の値が含まれた領域。
- DATESEP の値が含まれた領域。

**IC\_ADDRA**

次のアドレスのいずれかです。

- QUEUE の値が含まれた領域。
- DAYCOUNT の値が含まれたフルワード。

**IC\_ADDRB**

次のアドレスのいずれかです。

- HOURS の値が含まれた領域。
- DAYOFWEEK の値が含まれたフルワード。

**IC\_ADDRRC**

次のアドレスのいずれかです。

- MINUTES の値が含まれた領域。
- DAYOFMONTH の値が含まれたフルワード。

**IC\_ADDRD**

次のアドレスのいずれかです。

- SECONDS の値が含まれた領域。
- MONTHOFYEAR の値が含まれたフルワード。

**IC\_ADDRE**

次のアドレスのいずれかです。

- USERID の値が含まれた領域。
- YEAR の値が含まれたフルワード。

**IC\_ADDRF**

次のアドレスのいずれかです。

- SYSNET の値が含まれた 8 バイトの領域。
- TIME の値が含まれた領域。

**IC\_ADDR10**

TIMESEP の値が含まれた 1 バイトの領域のアドレスです。

**IC\_ADDR11**

YYYYDDD の値が含まれた領域のアドレスです。

**IC\_ADDR12**

YYYYMMDD の値が含まれた領域のアドレスです。

**IC\_ADDR13**

YYYYDDMM の値が含まれた領域のアドレスです。

**IC\_ADDR14**

DDMMYYYY の値が含まれた領域のアドレスです。

**IC\_ADDR15**

MMDDYYYY の値が含まれた領域のアドレスです。

**IC\_ADDR16**

FULLDATE の値が含まれた領域のアドレスです。

**IC\_ADDR1D**

BRDATA の値が含まれた領域のアドレスです。

**IC\_ADDR1E**

BRDATALENGTH の値が含まれたフルワードのアドレスです。

**IC\_ADDR1F**

CHANNEL の値が含まれた 16 バイトの領域のアドレスです。

**コマンド・レベルのパラメーター構造のフィールドの変更**

インターバル制御機能に渡される一部のフィールドは要求への入力として使用され、一部は出力フィールドとして使用され、さらに一部は入力および出力の両方に使用されます。ユーザー出口プログラムでフィールドを変更するために使用する方法は、フィールドの使用方法に応じて異なります。

以下は、常に入力フィールドです。

- INTERVAL
- TIME
- REQID
- FROM
- TERMID
- SYSID
- HOURS
- MINUTES
- SECONDS
- USERID
- CHANNEL

以下は、常に出力フィールドです。

- DATE
- DATEFORM
- DAYCOUNT
- DAYOFMONTH
- DAYOFWEEK
- DDMMYY
- DDMMYYYY
- FULLDATE
- INTO
- MMDDYY
- MMDDYYYY
- MONTHOFYEAR

- SET
- TIME
- YEAR
- YYDDD
- YYDDMM
- YYMMDD
- YYYYDDD
- YYYYDDMM
- YYYYMMDD

以下は、START 要求では入力フィールドであり、RETRIEVE 要求では出力フィールドです。

- RTRANSID
- RTERMID
- QUEUE

LENGTH は、START 要求では入力フィールドで、SET が指定された RETRIEVE では出力フィールド、INTO が指定された RETRIEVE では入出力フィールドです。

ABSTIME は、FORMATTIME 要求では入力フィールドであり、ASKTIME 要求では出力フィールドです。DATESEP および TIMESEP は、FORMATTIME 要求では入力フィールドとして使用できます。

### 入力フィールドの変更

入力フィールドを変更する場合の正しい方法は、そのフィールドの新規コピーを作成し、コマンド・レベル・パラメーター・リスト内のアドレスを変更して新規データを指すようにすることです。

注：コマンド・レベルのパラメーター・リストによって指定されたデータを変更することによって、入力フィールドを変更してはなりません。これを行うと、アプリケーション・プログラムに属しているストレージが破壊され、プログラムがフィールドを再使用しようとしたときに障害が発生することになります。

### 出力フィールドの変更

コマンド・レベルのパラメーター・リストが指すデータを変更することによって、出力フィールドを変更します。

142 ページの『[入力フィールドの変更](#)』で説明する手法は、出力フィールドの変更には適していません。結果が、アプリケーションの領域ではなく新しい領域に返されるため、アプリケーションで表示できなくなるからです。

出力フィールドの場合は、アプリケーションのデータを所定の位置で変更することができます。これは、アプリケーションが、そのフィールドが何らかの方法で変更されることを予期しているためです。

### EID の変更

EID を変更することで要求に対して大きな変更 (DELAY 要求から START 要求への変更など) を行うことはできません。ただし、SYSID の存在ビットをオンにして要求をリモート・システムに送信できるようにするなど、要求に対する小さな変更を行うことはできます。

一部のインターバル制御コマンドでは、EID 内の 2 ビットを使用して単一のキーワードを示します。例えば、EXEC CICS START では、TERMID を示すために 2 ビットを使用します。IC\_BITS1 に入っている先頭ビットは、コマンド・パラメーター・リスト内の ADDR6 が有効であることを示します (ADDR6 は TERMID を指します)。IC\_EIDOPT7 に入っている 2 番目のビットは、TERMID キーワードがコマンドで指定されていることを示すキーワード存在ビットです。

この状況に該当する場合は、ユーザー出口プログラム内部からこのようなコマンドを変更しようとする場合、必ず両方のビット設定が (一貫性のある形で) 変更されるようにする必要があります。そうでないと、結果は予測不能になります。

以下のリストは、EID 内の**変更可能な**ビットを示しています。EID のその他の部分を変更しようとしても、すべて無視されます。

## **IC\_BITS1**

### **X'80'**

REQID の存在ビット (要求が CANCEL の場合)

### **X'40'**

LENGTH の存在ビット (要求が RETRIEVE の場合) または REQID の存在ビット

### **X'10'**

FROM の存在ビット

### **X'08'**

LENGTH の存在ビット

### **X'04'**

TERMID の存在ビット

### **X'02'**

SYSID の存在ビット

### **X'01'**

RTRANSID の存在ビット

## **IC\_BITS2**

### **X'80'**

RTERMID の存在ビット

### **X'40'**

QUEUE の存在ビット

### **X'20'**

HOURS の存在ビット

### **X'10'**

MINUTES の存在ビット

### **X'08'**

SECONDS の存在ビット

## **IC\_EIDOPT6**

### **X'20'**

HOURS の 2 次存在ビット

### **X'10'**

FMH の存在ビット

### **X'08'**

SECONDS の 2 次存在ビット

### **X'04'**

MINUTES の 2 次存在ビット

### **X'02'**

PROTECT の存在ビット

### **X'01'**

NOCHECK の存在ビット

## **IC\_EIDOPT7**

IC\_EIDOPT7 内のビットは、同じ機能グループ内でのみ変更する必要があります。つまり、START 要求では、START 要求について有効と定義された存在ビットのみを設定する必要があります。

### **ASKTIME 要求**

### **X'13'**

ASKTIME 要求。この値はすべての ASKTIME 要求の場合に固定です。変更してはなりません。

### **DELAY 要求**

### **X'20'**

DELAY 要求

**X'08'**

TIME が指定されています。

**X'04'**

REQID が指定されています。

**POST 要求**

**X'30'**

POST 要求

**X'08'**

TIME が指定されています。

**X'04'**

REQID が指定されています。

**START 要求**

**X'40'**

START 要求 (DATA なし)

**X'50'**

DATA 付きの START 要求

**X'70'**

1 つ以上の RTRANSID、RTERMID、QUEUE、または FMH が指定された START。

**X'08'**

TIME が指定されています。

**X'04'**

REQID が指定されています。

**X'01'**

TERMID が指定されています。

**RETRIEVE 要求**

**X'82'**

RETRIEVE 要求

**CANCEL 要求**

**X'F0'**

CANCEL 要求

**X'04'**

REQID が指定されています。

**IC\_EIDOPT8**

**X'20'**

CICS では使用しません。

EID は、アプリケーション・プログラムに戻る前に、元の値にリセットされます。つまり、EID に対して行われた変更は、インターバル制御要求の期間中のみ保持されます。

**注：**ユーザー出口プログラムは、EID に大きな変更を加えることはできません。ただし、許容されている小さい変更を加える場合でも、十分に注意する必要があります。

**インターバル制御要求トークン UEPICTOK の使用**

UEPICTOK は、同じインターバル制御要求に対して XICEREQ ユーザー出口と XICEREQC ユーザー出口の間で情報を渡すために使用できる、4 バイト領域のアドレスを提供します。

例えば、XICEREQ ユーザー出口によって取得されたストレージの部分 (XICEREQC 出口によって解放される) のアドレスは、UEPICTOK フィールドで渡すことができます。

UEPICTOK は、その内容が要求の終了時に破棄される可能性があるため、単一のインターバル制御要求の期間のみ使用可能です。グローバル・ユーザー出口の連続する呼び出しの間で情報を渡す必要がある場合



は、タスク・トークン UEPTSTOK を使用してこれを行うことができます。UEPTSTOK について詳しくは、[タスク・トークン UEPTSTOK の使用](#)を参照してください。

### EXEC インターフェース・ブロック (EIB)

EIBRSRCE、EIBRCODE、EIBRESP、および EIBRESP2 のコピーを出口に渡して、XICEREQ および XICEREQC 内の完了情報およびリソース情報の変更や設定を行ったり、XICEREQC 内の完了情報およびリソース情報を検査したりすることができます。

パラメーター・リストで指定されている EIBRSRCE、EIBRCODE、EIBRESP、および EIBRESP2 のコピーを更新できます。XICEREQC の完了後に、または XICEREQ に戻りコード「bypass」を指定してある場合に、インターバル制御機能が値を実 EIB にコピーします。

有効なインターバル制御応答を設定する必要があります。EIBRCODE、EIBRESP、および EIBRESP2 の 3 つすべてを、整合性のある値セット (有効な完了を記述するために CICS インターバル制御機能によって設定されるような値セット) に設定する必要があります。**CICS では、EIBRCODE、EIBRESP、および EIBRESP2 の整合性は監視されません。**ただし、EIBRCODE がゼロ以外の値に設定され、EIBRESP がゼロに設定されると、CICS はゼロ以外の値で EIBRESP をオーバーライドします。EIBRCODE、EIBRESP、および EIBRESP2 の値の設定に役立つように、インターバル制御機能で使用する値が DFHICUED に指定されます。

XICEREQC 出口を使用して戻りコードを調べる場合には、EIBRCODE 値を参照してください。

### XICEREQ および XICEREQC の使用方法の例

この例では、XICEREQ および XICEREQC を使用して START 要求を複数の異なる CICS 領域にルーティングし、簡便なロード・バランシング・メカニズムを提供します。この例では、出口の機能のみを示します。この例は、機能を実行するのに理想的な方法を示すものではありません。

**XICEREQ では以下のようにします。**

1. グローバル作業域 (GWA) をスキャンして、適切な CICS 領域 (例えば、現在処理中の START 要求の数が最も少ない領域) を見つけます。
2. 要求をどのシステムにルーティングするかを決めたら、そのシステムの使用回数を増やします。
3. この要求の SYSID を格納するための 4 バイト領域を取得します。これは、GWA から割り振ると、GETMAIN を発行しなくても済みます。GETMAIN を発行してこの領域を取得する場合は、UEPICTOK を取得したストレージのアドレスに設定してください。
4. IC\_ADDR7 を 4 バイト領域のアドレスに設定し、XICEREQC もこの領域を使用できるようにします。
5. この時点で IC\_ADDR7 を設定するとそれが最終アドレスになる場合は、アドレスの高位ビットを設定し、以前の最終アドレスであった高位ビットをリセットします。
6. SYSID が指定されていることを示すために、IC\_BITS1 内の X'02' 存在ビットをオンに設定します。
7. CICS に戻ります。

**XICEREQC では以下のようにします。**

1. グローバル作業域 (GWA) をスキャンし、SYSID パラメーターで指定された CICS 領域のエントリーを見つけてます。
2. このシステムの使用回数を減らします。
3. SYSID を保持する領域を取得するために XICEREQ で GETMAIN が発行されていた場合は、UEPICTOK に保持されているアドレスに対して FREEMAIN を発行します。
4. CICS に戻ります。

### 例およびサンプル・プログラム

CICS では、XICEREQ 出口で使用するためのプログラムを 2 つ提供しています。

- DFH\$XTSE (ソフトコピー・リストとしてのみ提供され、ソース・コード・ファイルとしての提供ではありません)。これは、すべての EXEC インターフェース 出口に渡されるコマンド・レベルのパラメーター構造のフィールドを変更する方法を示すプログラム例です。

- DFH\$ICCN。これは、分散ルーティング環境で使用するサンプル・プログラムです。分散ルーティング環境では、前に発行したインターバル制御要求の取り消しが必要になっても、CANCEL をどの領域にあてて発行すればよいかわかる方法がありません。DFH\$ICCN を設計して処理する状況の例については、[以下を参照してください。](#) [インターバル制御要求の取り消し。](#)

## ローダー・ドメイン出口 XLDLOAD および XLDELETE

ローダー・ドメインには 2 つのグローバル・ユーザー出口があります。XLDLOAD は、プログラムのインスタンスがストレージに入れられた後、プログラムが使用可能にされる前に呼び出されます。

XLDELETE は、プログラムのインスタンスが CICS によって解放された後、プログラムがストレージから解放される前に呼び出されます。

LPA 常駐プログラムの場合、プログラムが物理的にロードまたは解放されていなくても、プログラムが獲得または解放されるときにこの出口がやはり呼び出されます。

これらはどちらも通知専用出口です。出口プログラムによって出口パラメーターに加えられた変更は、出口プログラムが設定したすべての戻りコードと同様、CICS によって無視されます。

## 出口 XLDLOAD

この出口は、プログラム・インスタンスがストレージに入れられた場合に、そのプログラムが使用可能になる前に呼び出されます。

### 呼び出される状況

プログラムのインスタンスが記憶域に入れられた後、プログラムが使用可能にされる前。

### 出口固有のパラメーター

#### UEPPROGN

ロードされるプログラムの名前が含まれた 8 文字のフィールドのアドレス。

#### UEPPROGL

ロードされるプログラムの長さ (バイト単位) が含まれた 4 バイト・フィールドのアドレス。

#### UEPLDPT

プログラムがロードされた場所のアドレスが含まれた 4 バイト・フィールドのアドレス。

#### UEPENTRY

プログラムの入り口点のアドレスが含まれた 4 バイト・フィールドのアドレス。

#### UEPTRANID

ゼロ、またはこの出口が呼び出されたときに適用されたトランザクション ID が含まれた 4 バイト・フィールドのアドレス。

#### UEPUSER

ゼロ、またはこの出口が呼び出されたときに制御権を持っていたユーザー ID が含まれた 8 バイト・フィールドのアドレス。

#### UEPTERM

ゼロ、またはこの出口が呼び出されたときに実行中だったトランザクションに関連付けられた端末名が含まれた 4 バイト・フィールドのアドレス。

#### UEPPROG

ゼロ、またはこの出口が呼び出されたときに制御権を持っていたプログラムの名前が含まれた 8 文字のフィールドのアドレス。

#### UEPLDCTXT

ゼロ、またはアプリケーションに属する専用プログラムがロードされたときのアプリケーション・コンテキストが含まれた 140 バイト・フィールドのアドレス。このフィールドには、以下のものが入っています。

1. プラットフォーム名。スペースを埋め込んで 64 文字。
2. アプリケーション名。スペースを埋め込んで 64 文字。
3. アプリケーションのメジャー・バージョン番号。フルワード・バイナリー値。
4. アプリケーションのマイナー・バージョン番号。フルワード・バイナリー値。

5. アプリケーションのマイクロ・バージョン番号。フルワード・バイナリー値。

CICS は、この情報にマップする **DFHUEACD** という名前の DSECT を提供します。**DFHUEACD** についての詳細は、『Data Areas』の『UEACD - User exit application context』を参照してください。

## 戻りコード

### **UERCNORM**

処理が続行される。

### **XPI** 呼び出し

これは使用しないでください。

### **API** および **SPI** 呼び出し

これは使用しないでください。

## 出口 **XLDELETE**

この出口は、プログラム・インスタンスが CICS によって解放されたときに、そのプログラムがストレージから解放される前に呼び出されます。

### 呼び出される状況

プログラムのインスタンスが CICS によって解放された後、プログラムがストレージから解放される前。

### 出口固有のパラメーター

#### **UEPPROGN**

解放されるプログラムの名前が含まれた 8 文字のフィールドのアドレス。

#### **UEPPROGL**

解放されるプログラムの長さ (バイト単位) が含まれた 4 バイト・フィールドのアドレス。

#### **UEPLDPT**

プログラムがストレージに常駐している場所のアドレスが含まれた 4 バイト・フィールドのアドレス。

#### **UEPENTRY**

プログラムの入り口点のアドレスが含まれた 4 バイト・フィールドのアドレス。

#### **UEPTRANID**

ゼロ、またはこの出口が呼び出されたときに適用されたトランザクション ID が含まれた 4 バイト・フィールドのアドレス。

#### **UEPUSER**

ゼロ、またはこの出口が呼び出されたときに制御権を持っていたユーザー ID が含まれた 8 バイト・フィールドのアドレス。

#### **UEPTERM**

ゼロ、またはこの出口が呼び出されたときに実行中だったトランザクションに関連付けられた端末名が含まれた 4 バイト・フィールドのアドレス。

#### **UEPPROG**

ゼロ、またはこの出口が呼び出されたときに制御権を持っていたプログラムの名前が含まれた 8 文字のフィールドのアドレス。

#### **UEPLDCTXT**

ゼロ、またはアプリケーションに属する専用プログラムが削除されたときのアプリケーション・コンテキストが含まれた 140 バイト・フィールドのアドレス。このフィールドには、以下のものが入っています。

1. プラットフォーム名。スペースを埋め込んで 64 文字。
2. アプリケーション名。スペースを埋め込んで 64 文字。
3. アプリケーションのメジャー・バージョン番号。フルワード・バイナリー値。
4. アプリケーションのマイナー・バージョン番号。フルワード・バイナリー値。
5. アプリケーションのマイクロ・バージョン番号。フルワード・バイナリー値。

CICS は、この情報にマップする **DFHUEACD** という名前の DSECT を提供します。**DFHUEACD** についての詳細は、『Data Areas』の『UEACD - User exit application context』を参照してください。

## 戻りコード

### UERCNORM

処理が続行される。

### XPI 呼び出し

これは使用しないでください。

### API および SPI 呼び出し

これは使用しないでください。

## ログ・マネージャー・ドメイン出口 XLGSTRM

ログ・マネージャー・ドメインには XLGSTRM という出口点が 1 つあります。XLGSTRM を使用して、MVS に対する要求を変更し、新規ログ・ストリームを作成することができます。モデル・ログ・ストリーム名パラメーターおよび他のパラメーターが MVS システム・ロガーに渡される前に、それらのパラメーターを変更することができます。

ログ・ストリームが MVS に定義されていないために CICS から MVS システム・ロガーへのログ・ストリーム接続要求が失敗した場合、CICS は、MVS システム・ロガーに対して、モデル・ログ・ストリーム定義を使用してログ・ストリームを動的に作成するための要求を発行します。

CICS が MVS に渡すモデル・ログ・ストリーム名は、以下のように、ジャーナル名がシステム・ログまたは CICS 一般ログのいずれを参照しているかによって異なります。

### CICS システム・ログ

`&sysname.LSN_last_qualifier.MODEL`

`&sysname` は MVS イメージのシステム名に解決される MVS シンボルです。`LSN_last_qualifier` は、JOURNALMODEL リソース定義で指定された、ログ・ストリーム名の最終修飾子です。

DFHLOG および DFHSHUNT に JOURNALMODEL リソース定義を指定しない場合、または DFHLGMOD で指定された CICS 定義を使用する場合は、モデル・ログ・ストリーム名はデフォルトで `&sysname.DFHLOG.MODEL` および `&sysname.DFHSHUNT.MODEL` になります。

例えば、CICS 領域がその 1 次システム・ログのためのログ・ストリームの作成要求を発行し、CICS が MV10 というシステム ID の MVS イメージで実行されていて、デフォルトの JOURNALMODEL 定義を使用している場合、MVS システム・ロガーは、MV10.DFHLOG.MODEL という名前のモデル・ログ・ストリームを見つけようとしています。

MVS イメージのシステム名が数字で始まり、長さが 8 文字未満の場合は、CICS はその名前に「C」という接頭部を付けて、モデル・ログ・ストリーム名が `C&sysname.LSN_last_qualifier` になるようにします。これは、数字で始まるログ・ストリーム名は MVS システム・ロガーで拒否されるためです。MVS イメージのシステム名が数字で始まっていて、長さが既に 8 文字 (最大) に達している場合は、CICS は「C」という接頭部を追加しません。つまり、MVS システム・ロガーはそのデフォルトのモデル・ログ・ストリーム名を拒否します。ただし、グローバル・ユーザー出口プログラムでモデル・ログ・ストリーム名を変更することができます。

### CICS 一般ログ

`LSN_qualifier_1.LSN_qualifier2.MODEL`。これら 2 つの修飾子のデフォルトは CICS 領域ユーザー ID と CICS 領域アプリケーション ID ですが、JOURNALMODEL リソース定義で指定したユーザー定義値にすることもできます。

例えば、CICS 領域ユーザー ID が CICSHT## で、アプリケーション ID が CICSHTA1 の場合、デフォルトのモデル名は `CICSHT##.CICSHTA1.MODEL` になります。

XLGSTRM グローバル・ユーザー出口プログラムに以下の情報が渡されます。

- ・ 定義するログ・ストリームの名前
- ・ デフォルトのモデル・ログ・ストリーム名
- ・ システム・ログ・フラグ
- ・ MVS システム・ロガーの IXGINVNT パラメーター・リスト

UEPMLSN 出口固有パラメーターが指すフィールドを更新することによって、出口プログラムでモデル・ストリーム名を変更することができます。以下の例は、出口プログラムでモデル・ストリーム名を変更する方法を示します。

```
L      R3,UEPMLSN      R3 = address of stream name
MVC    0(26,R3),=CL26'NEW.MODEL.NAME'
```

UEPIXG パラメーターが指すフィールドを更新することによって、出口プログラムは、ログ・ストリームを定義するために MVS システム・ロガーで使用する IXGINVNT マクロ・パラメーター・リストを修正することができます。マクロの IXGINVNT MF=M 形式を使用してください。これにより、使用するログ・ストリーム属性を出口で指定できるようになります。以下の例は、出口プログラムで構造名を変更する方法を示します。

```
L      R9,UEPIXG
IXGINVNT REQUEST=DEFINE,
        TYPE=LOGSTREAM,
        STRUCTNAME=NEW_STRUCTURE,
        MF=(M,(R9),NOCHECK)

NEW_STRUCTURE DC CL16'LOG_SYSTEST_009'
```

IXGINVNT マクロのリスト形式および実行形式をコーディングする必要も、出口に IXGCON マクロまたは IXGANSAA マクロを組み込む必要もありません。これらは、DEFINE 要求を発行する CICS コードによって提供されます。

IXGINVNT サービスについては、「[z/OS MVS プログラミング: アセンブラー・サービスガイド](#)」を参照してください。

XLGSTRM グローバル・ユーザー出口プログラムでは、ログ・ストリーム定義の明示属性を設定でき、ログ・ストリーム定義をバイパスするための戻りコードも設定できます。

注: XLGSTRM で CICS システム・ログの接続をインターセプトするには、第 1 フェーズの PLT プログラムで出口プログラムを使用可能にする必要があります。

### XLGSTRM 用のグローバル・ユーザー出口のサンプル・プログラム

DFH\$LGLS は、XLGSTRM 出口点のためのグローバル・ユーザー出口のサンプル・プログラムです。この例では、XLGSTRM 出口プログラムに渡されるパラメーターの一部にアクセスして変更する方法を示しています。詳しくは、[ログ・マネージャー・ドメイン・サンプル出口プログラム: DFH\\$LGLS](#) を参照してください。

## 出口 XLGSTRM

出口 XLGSTRM が呼び出されるのは、ログ・ストリームが存在しないことを CICS ログ・マネージャーが検出した後、ログ・ストリームを動的に定義するために CICS が MVS システム・ロガーを呼び出す前です。

### 出口固有のパラメーター

#### UEPTRANID

4 バイトのトランザクション ID のアドレス。

#### UEPUSER

現行タスクがユーザー・タスクである場合は、トランザクションに関連付けられている 8 バイトのユーザー ID のアドレス。

#### UEPTERM

トランザクションに関連付けられている 4 バイトの端末 ID のアドレス (存在する場合)。

#### UEPPROG

このトランザクション用の 8 バイトのアプリケーション・プログラム名のアドレス (存在する場合)。

#### UEPLSN

定義対象のログ・ストリームの名前が含まれた 26 文字のフィールドのアドレス。

出口プログラムでログ・ストリームの名前を変更しないでください。出口から戻ったときに、CICS は、UEPLSN によってアドレス指定されているフィールドの内容に対する変更をすべて無視します。ログ・ストリーム名を選択するために、JOURNALMODEL 定義が提供されています。

## UEPMLSN

新規ログ・ストリームの属性を提供するために使用されるモデル・ログ・ストリームの名前を指定する 26 文字のフィールドのアドレス。このフィールドは変更可能で、グローバル・ユーザー出口プログラムで、CICS で生成されたモデル・ログ・ストリーム名とは異なるモデル・ログ・ストリーム名を指定するために使用できます。

## UEPIXG

MVS システム・ロガーがログ・ストリームを定義するために使用する IXGINVNT マクロ・パラメーター・リストのアドレス。IXGINVNT マクロの MF=M 形式を使用して、グローバル・ユーザー出口プログラムは使用するログ・ストリーム属性を指定できます。

IXGINVNT マクロの詳細については、「[z/OS MVS Programming: Authorized Assembler Services Guide](#)」を参照してください。

## UEPLGTYP

作成されるログ・ストリームがシステム・ログ用を一般ログ用のどちらであるかを示す 1 バイト・フィールドのアドレス。有効な値は次のとおりです。

### UEPSYSLG

ログ・ストリームは CICS システム・ログ用です。

### UEPGENLG

ログ・ストリームは一般ログ (順方向リカバリー・ログ、ユーザー・ジャーナル、または自動ジャーナル) 用です。

## 戻りコード

### UERCNORM

CICS は処理を続行し、ログ・ストリームを定義しようとします。

### UERCBYR

CICS はログ・ストリームを定義しようとしません。ログ・ストリームを使用しようとするプロセス (例えば、データ・セットのオープン) は失敗します。

## XPI 呼び出し

すべて使用できます。

## API コマンドおよび SPI コマンド

これは使用しないでください。

## XLGSTRM 出口の使用方法的例

XLGSTRM 出口は、代替モデル・ログ・ストリームを選択するために使用します。

200 の CICS 領域が 20 の MVS イメージで実行されているとします。各 CICS 領域で使用される個々のログ・ストリームを明示的に定義する必要がなくなるように、モデル定義を使用することにしました。ログ・ストリームは、その最初の使用時に MVS に動的に定義されます。その際、代替モデル・ログ・ストリームから選択するために、XLGSTRM 出口プログラムを使用します。この流れをまとめると、以下のようになります。

1. CICS 領域の初回始動時に、INITPARM システム初期設定パラメーターで以下を指定します。

```
INITPARM=(Exit_enabler_pgmname=nnn)
```

ここで、

- Exit\_enabler\_pgmname は、XLGSTRM ユーザー出口プログラムを使用可能にするプログラムの名前です。
  - nnn は、同一セットのログ・ストリーム・モデルを共用する CICS 領域のグループを識別する番号です。
2. XLGSTRM ユーザー出口プログラムを使用可能にするプログラムは、EXEC CICS ASSIGN INITPARM コマンドを発行して値 nnn を取得し、それを出口プログラムのグローバル作業域に置きます。
  3. ログ・ストリームが定義されていないために領域がそのシステム・ログに接続しようとする、XLGSTRM 出口プログラムが呼び出されます。出口プログラムがモデル CICS.DFHLOG.MODELnnn を選択します。



## メッセージ・ドメイン出口 XMEOUT

XMEOUT 出口を使用して、CICS メッセージ・ドメインを使用する CICS と、CICS メッセージ・ドメインを使用する CICSplex SM メッセージを抑止または転送することができます。

出口プログラムには以下の制限があります。

- システム・コンソールまたは一時データ・キューに送信されたメッセージのみを抑止または転送することができます。端末オペレーターに送信されるメッセージを抑止または転送することはできません。端末オペレーターに送信されたメッセージに対しては、XMEOUT は呼び出されません。
- メッセージ・ドメインを使用するメッセージのみを抑止または転送することができます。どの CICS メッセージにこれが適用されるかは CICS メッセージ で特定できます。XMEOUT を呼び出すことができる各メッセージの説明に、「XMEOUT パラメーター/メッセージ挿入」というタイトルのリストが含まれています。メッセージにリストされている XMEOUT パラメーターがない場合、メッセージは出口を呼び出すことはできません。例えば、メッセージ DFHDX8320 は XMEOUT を呼び出すことができますが、メッセージ DFHDU0205 はできません。

CICSplex SM の場合、XMEOUT が呼び出されるのは、EYULOG という宛先を持つメッセージの場合のみです。これは、このようなメッセージがメッセージ・ドメインを使用するメッセージであるからです。どのメッセージにこれが適用されるかは CICSplex SM メッセージ から特定できます。

**注：**XMEOUT 出口を呼び出す CICSplex SM メッセージを転送または抑止できるのは、ジョブ・ログまたはコンソールからのみです。EYULOG から抑止することはできません。

- CICSplex SM Web ユーザー・インターフェース・メッセージを転送または抑止することはできません。
- メッセージのテキストを変更したり、メッセージの挿入を変更したりすることはできません。これを行おうとすると、CICS はその変更を無視します。
- CICS 初期設定の初期段階で出されたメッセージを抑止または転送することはできません (その段階では出口を使用可能にすることができないため)。
- 元のメッセージ宛先に 1 つ以上の一時データ (TD) キューが組み込まれていない限り、CICS シャットダウン時にメッセージを一時データ・キューに転送することはできません。これを行おうとすると、メッセージは元の宛先にルーティングされ、コンソールに対してメッセージ DFHME0120I が発行されます。ユーザー出口プログラムはメッセージ DFHME0120I を転送することはできませんが、このメッセージを抑止することはできます。

この制限が必要なのは、一時データ・キュー機能が終了した後であっても、メッセージ・ドメインは CICS のシャットダウン中にメッセージを処理する必要があるためです。

CICS のシャットダウンが開始されたかどうかを調べるには、出口プログラムでメッセージ DFHME0120 の最初のインスタンスの存在を確認することができます。DFHME0120 の発行後に、TD キューへのメッセージの転送を停止することができます。

**注：**一時データ・キューへのメッセージの転送中に一時データ要求が失敗すると、メッセージは失われます。MEME 例外トレース・ポイント ID X'0328' が書き込まれます。このトレース・エントリーの解釈ストリングに、一時データ要求が失敗した理由が示されています。

### 重要

再帰の危険性があるため、XMEOUT 出口プログラムで以下のメッセージを転送しようとししないでください。

- 一時データ・プログラムによって生成されるすべての DFHTDxxxx メッセージ。
- DFHUS0002 から DFHUS0006 の範囲のユーザー・ドメイン・メッセージ、およびメッセージ DFHUS0150。
- トランザクション・マネージャー・メッセージ DFHXM0212、DFHXM0213、DFHXM0304、および DFHXM0308。
- アプリケーション・メッセージ DFHAP0001、DFHAP0002、DFHAP0004、DFHAP0601、DFHAP0602、および DFHAP0603。
- ゼロ以外の TRIGLEV 値を使用して定義された区画内キューに対するすべてのユーザー・ドメイン (DFHUSxxxx) メッセージ (ユーザー・ドメインがエラー・リカバリー処理を実行中に生成されたメッセージの場合)。

メッセージ定義テンプレートには、*noroute* と呼ばれる標識が含まれています。発行されているメッセージが XMEOUT 出口プログラムによって一時データキューに転送できない場合、この標識がオンに設定されます。標識のアドレスは UEPNRTE 出口固有パラメーターの XMEOUT に渡されます。出口プログラムは、この標識の値を検査してから、特定のメッセージを転送するかどうかを決定することができます。

**注：** 出口プログラムが不適格メッセージを転送しようとする、メッセージ・ドメインは転送を禁止し、代わりにそのメッセージを、メッセージ DFHME0137 と一緒にコンソールに送ります。

この制限によって影響を受ける各メッセージは、[CICS メッセージ](#)内の注で識別されています。

APPLID (アプリケーション ID) をオプション・パラメーターとしてメッセージに入れて渡すことができます。ただし、メッセージに挿入されたアプリケーション ID は、現行 CICS システムのアプリケーション ID ではない可能性があります。例えば、CICSplex SM MAS が CMAS に転送される場合、MAS システムのアプリケーション ID が渡されることがあります。この場合、メッセージに含まれているのは MAS システムのアプリケーション ID であり、現行システム (CMAS) のアプリケーション ID ではありません。

出口プログラムでは、パラメーター・リストの以下のフィールドが指すアドレスに保持されている値を変更することによって、メッセージを抑止または転送することができます。プログラムでその他の値セットを変更することはできません。

- UEPMROU (経路コード)
- UEPMNRC (経路コードの数)
- UEPMTDQ (一時データ・キュー名)
- UEPMNTD (TDQ の数)

### DFHAP1900 メッセージをモニターするための XMEOUT の使用

特定のシステム・プログラミング・インターフェース・コマンドによって CICS システム構成に変更が加えられると、メッセージ DFHAP1900 が発行されます。該当するコマンドは、**SET**、**PERFORM**、**ENABLE**、**DISABLE**、または **RESYNC** です。コマンドは、一時データ・キュー CADS に書き込まれます。DFHAP1900 メッセージは、動的構成変更の監査を提供することができ、モニター判別にも役立ちます。詳しくは、[SPI commands that can be audited](#) を参照してください。

**注：** この出口で SPI コマンドを発行しないでください。これを発行すると、出口で再帰が発生するため、DFHAP1900 メッセージが発行されることになります。

## 出口 XMEOUT

この出口は、メッセージ・ドメインが CICS メッセージを送信する前に呼び出されます。

### 呼び出される状況

メッセージ・ドメインが CICS メッセージをその宛先に送信する前。

### 出口固有のパラメーター

#### UEPTRANID

4 バイトのトランザクション ID のアドレス。

#### UEPUSER

8 バイトのユーザー ID のアドレス。

#### UEPTERM

現行トランザクションが実行されている端末の 4 バイト ID のアドレス。現行トランザクションが端末と関連付けられていない場合、アドレス指定されたフィールドには 16 進ゼロが含まれます。

#### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス、または NULL (現行アプリケーションが存在しない場合)。

#### UEPMNUM

メッセージ番号を格納する 4 バイト・フィールドのアドレス。CICSplex SM メッセージの場合、このフィールドには 2 進ゼロが含まれます。



## UEPMDOM

CICS メッセージのドメイン ID を格納する 2 バイト・フィールドのアドレス。CICSplex SM メッセージの場合、このフィールドには 2 進ゼロが含まれます。

## UEPMROU

最大 28 の宛先コードの配列のアドレス。宛先コードは、範囲が 1 から 28 の数でなければなりません。

## UEPMNRC

宛先コード配列内の宛先コードの数が含まれたハーフワードのアドレス。

## UEPMTDQ

メッセージの送信先である、最大 25 の一時データ・キュー名の配列のアドレス。TD キュー名は 4 つの英数字で構成されている必要があります。

## UEPMNTD

キュー配列内の TDQ の数が含まれたハーフワードのアドレス。

## UEPINSN

メッセージ挿入の数が含まれた 2 バイト・フィールドのアドレス。

## UEPINSA

配列のアドレス。配列の各エレメントには単一メッセージ挿入に関する情報が含まれます。配列のサイズは挿入の数によって異なります。各配列エレメントには、以下の構造が含まれます。

```
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.
```

配列内の挿入の順序は、[CICS メッセージ](#)の特定のメッセージのエントリに示されています。以下に例を示します。

DFHFC0531 *date time applid* ファイル *filename* に対して開かれた自動ジャーナル *journal journalname* は、タイプが MVS モジュール *module* ではありません。

XMEOUT 挿入は、*date*、*time*、*applid*、*journal*、*journalname*、*filename*、および *module* です。4 番目の挿入 (*journal*) は、ファイル定義の際に JOURNAL に指定される数値です。

## UEPNRTE

XMEOUT によってメッセージを転送できるかどうかを示す 1 文字のフラグのアドレス。可能な値は次のとおりです。

**C'0:'**

メッセージを送付できます。

**C'1:'**

メッセージを送付できません。

## UEPCPID

3 バイト製品 ID のアドレス。可能な値は次のとおりです。

**DFH**

CICS メッセージ。

**EYU**

CICSplex SM メッセージ。

## UEPCPDOM

メッセージのドメイン ID を格納する 2 バイト・フィールドのアドレス。

## **UEPCPNUM**

メッセージ番号を格納する 4 バイト・フィールドのアドレス。

## **UEPCPSEV**

メッセージ重大度コードのアドレス。

## **戻りコード**

### **UERCNORM**

処理が続行される。

### **UERCBYBYP**

すべての宛先に対するメッセージを抑制します。

CICSplex SM メッセージは抑止できません。これらのメッセージの場合、UERCBYBYP の応答は UERCNORM として処理されます。

戻りコード UERCBYBYP を指定しても、メッセージ用に定義された MESSAGE システム・イベントの発行を抑止することにはなりません。

## **XPI 呼び出し**

WAIT\_MVS を使用できます。その他の呼び出しを使用しないでください。

## **モニター・ドメイン出口 (XMNOUT)**

この出口は、モニター・レコードが SMF またはレコード・バッファーに書き込まれる前に呼び出されます。

XMNOUT は、以下のイベント・ポイントで呼び出されます。

- クラス・モニター・レコードが SMF に渡される前
- パフォーマンス・クラス・モニター・レコードがパフォーマンス・レコード・バッファーに書き込まれる前
- トランザクション・リソース・モニター・レコードがトランザクション・リソース・レコード・バッファーに書き込まれる前

**注:** パフォーマンス・クラスとトランザクション・リソース・モニターの両方が CICS 領域でアクティブになっている場合は、XMNOUT を同一イベントについて 2 回呼び出すことができます。例えば、イベントがタスク終了イベントであり、CICS には適切なバッファーに移動するためにパフォーマンス・クラス・データとトランザクション・リソース・データの両方がある場合、XMNOUT はそれぞれのモニター・レコード・タイプについて 1 回ずつ呼び出されます。

この出口を使用して、レコードを調べたり SMF への出力を抑止したりすることができます。すべてのデータは読み取り専用であるため、含まれているデータを変更することはできません。

パフォーマンス・クラス・データ・レコードにデータを追加することもできます。それには、モニター管理テーブル (MCT) にダミーのユーザー・イベント・モニター・ポイント (EMP) を定義して、必要なサイズとリサイクルのデータ・フィールドを確保する必要があります。

## **出口 XMNOUT**

この出口は、モニター・レコードが SMF に書き込まれる前、または SMF への後続の書き込み用にバッファーに入れられる前に呼び出されます。

### **呼び出される状況**

XMNOUT は、以下の状況で呼び出されます。

- 例外クラス・モニター・レコードが SMF に書き込まれる前
- パフォーマンス・クラス・モニター・レコードが後ほど SMF に書き込まれるためにバッファーに入れられる前
- トランザクション・リソース・モニター・レコードが後ほど SMF に書き込まれるためにバッファーに入れられる前
- ID クラス・モニター・レコードが後ほど SMF に書き込まれるためにバッファーに入れられる前

## 出口固有のパラメーター

### UEPTRANID

4 バイトのトランザクション ID のアドレス。このフィールドは、タスク終了時には使用できません。

### UEPUSER

8 バイトのユーザー ID のアドレス。このフィールドは、タスク終了時には使用できません。

### UEPTERM

4 バイトの端末 ID のアドレス。このフィールドは、タスク終了時には使用できません。

### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス。このフィールドは、タスク終了時には使用できません。

### UEPDICT

ディクショナリーのアドレス。ディクショナリー項目のシーケンスは、マクロ DFHMCTDR から生成される DSECT によってマップされます。このフィールドは、パフォーマンス・クラス・レコードに関してのみ意味を持ちます。このモニター・レコード・タイプが例外クラス (タイプ 4)、トランザクション・リソース (タイプ 5)、または ID クラス (タイプ 6) の場合、このフィールドは 0 に設定されます。パラメーター UEPMRTYP を参照してください。

### UEPDICTE

フルワードのディクショナリー項目数のアドレス。このフィールドは、パフォーマンス・クラス・レコードに関してのみ意味を持ちます。このモニター・レコード・タイプが例外クラス (タイプ 4)、トランザクション・リソース (タイプ 5)、または ID クラス (タイプ 6) の場合、このフィールドは 0 に設定されます。パラメーター UEPMRTYP を参照してください。

### UEPFCL

一連のハーフワード・コネクター値が含まれるフィールド・コネクター・リストのアドレス。このフィールドは、パフォーマンス・クラス・レコードに関してのみ意味を持ちます。このモニター・レコード・タイプが例外クラス (タイプ 4)、トランザクション・リソース (タイプ 5)、または ID クラス (タイプ 6) の場合、このフィールドは 0 に設定されます。パラメーター UEPMRTYP を参照してください。

### UEPFCLNO

フルワードのフィールド・コネクター数のアドレス。このフィールドは、パフォーマンス・クラス・レコードに関してのみ意味を持ちます。このモニター・レコード・タイプが例外クラス (タイプ 4)、トランザクション・リソース (タイプ 5)、または ID クラス (タイプ 6) の場合、このフィールドは 0 に設定されます。パラメーター UEPMRTYP を参照してください。

### UEPMRTYP

ハーフワードのモニター・レコード・タイプのアドレス。モニター・レコード・タイプ値としては、以下の表に示されているいずれかの数値が可能です。

表 7. モニター・レコード・タイプ値およびその意味	
レコード・タイプ値	意味
3	パフォーマンス・クラス・モニター・レコード
4	例外クラス・モニター・レコード
5	トランザクション・リソース・モニター・レコード
6	ID クラス・モニター・レコード

### UEPMRLN

フルワードのモニター・レコード長のアドレス。

### UEPMREC

モニター・レコードのアドレス。UEPMREC の長さは、パラメーター UEPMRLN で示されます。

### UEPSRCTK

現行トランザクションの z/OS ワークロード・マネージャー・サービス報告クラス・トークン。z/OS ワークロード・マネージャーの CICS サポートが使用できない場合、このトークンはヌルです。

## UEMPREC

モニター・パフォーマンス・レコードのアドレス。このフィールドは、パフォーマンス・クラス・レコードに関してのみ意味を持ちます。このモニター・レコード・タイプが例外クラス (タイプ 4)、トランザクション・リソース (タイプ 5)、または ID クラス (タイプ 6) の場合、このフィールドは 0 に設定されます。パラメーター UEMRTYP を参照してください。このパラメーターによってアドレス指定されるパフォーマンス・レコードは DFHMNTDS DSECT を使用してマップする必要があり、UEPDICT および UEPDICTE ディクショナリー・パラメーターを使用してマップしないでください。

## 戻りコード

### UERCNORM

処理が続行される。

### UERCBYP

モニター・レコード出力を抑制します。

### UERCPUrg

タスクは XPI 呼び出し中にページされます。

## XPI 呼び出し

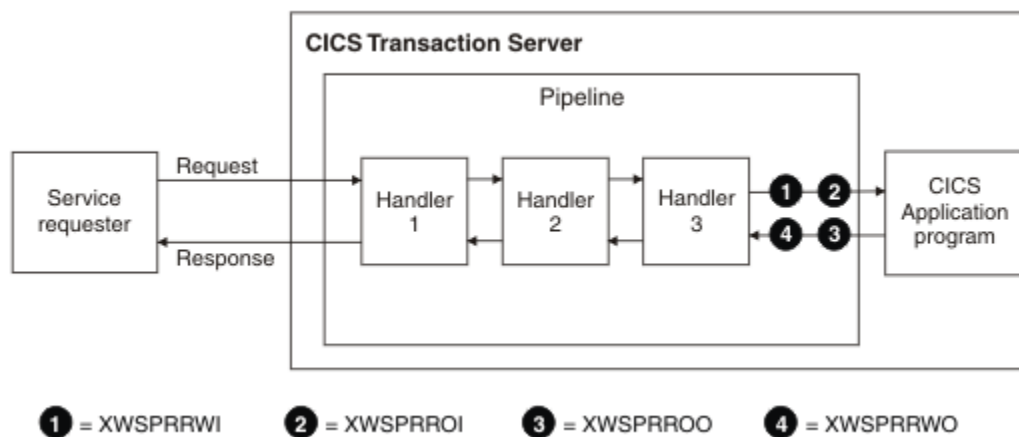
WAIT\_MVS を使用できます。その他の呼び出しを使用しないでください。

## パイプライン・ドメイン出口

パイプライン・ドメイン出口を使用して、パイプライン内のインバウンドおよびアウトバウンド Web サービスに対して行われる処理をカスタマイズできます。パイプライン・ドメイン出口を使用すると、Web サービス・プロバイダー・パイプライン、Web サービス・リクエスター・パイプライン、またはセキュリティー・メッセージ・ハンドラーを備えた Web サービス・リクエスター・パイプライン上のコンテナにアクセスできます。

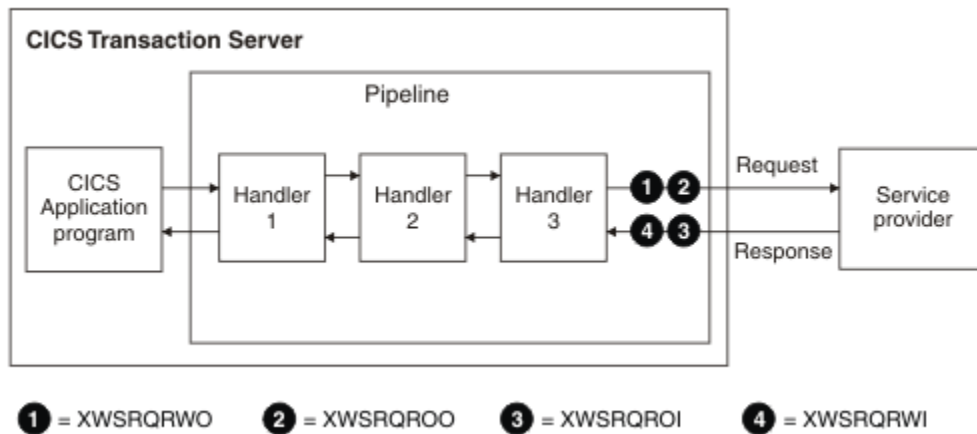
### プロバイダー・パイプライン内の GLUE ポイント

プロバイダー・パイプラインまたはセキュア・プロバイダー・パイプラインで使用するグローバル・ユーザー出口 (GLUE) ポイントの接頭部は、XWSPR です。この図は、GLUE ポイントを使用できる順序を示しています。

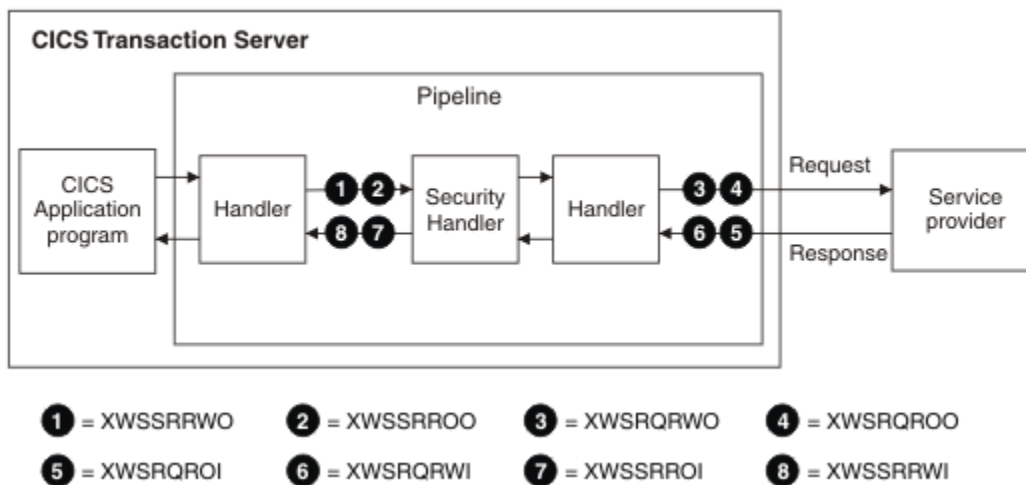


### リクエスター・パイプライン内の GLUE ポイント

リクエスター・パイプラインで使用する GLUE ポイントの接頭部は、XWSRQ です。この図は、GLUE ポイントを使用できる順序を示しています。



セキュア・リクエスター・パイプラインで使用できる GLUE ポイントの接頭部は、XWSSR です。セキュリティー・ハンドラーを含むパイプライン内で使用できる GLUE ポイントは 8 つあります。このうちの 4 つはセキュア・リクエスター・パイプライン内でのみ使用でき、4 つはすべてのリクエスター・パイプライン内で使用できます。この図は、GLUE ポイントを使用できる順序を示しています。



## 出口 XWSPRRWI

CICS が Web サービス要求の本体を言語構造に変換した後で、かつ XWSPRROI 出口のインスタンスが呼び出される前に、Web サービス・プロバイダー・アプリケーションによって処理される現行チャンネル上のコンテナにアクセスするには、XWSPRRWI 出口を使用します。

この出口を使用して、API および SPI コマンドを発行して、コンテナの情報の調査や更新を行い、SOAP 障害を発行することができます。

### 出口固有のパラメーター

#### UEPTRANID

4 バイトのトランザクション ID のアドレス。出口またはタスクはこのトランザクション ID の下で稼働します。

#### UEPUSER

8 バイトのユーザー ID のアドレス。このユーザー ID は、トランザクション ID に関連付けられています。

#### UEPTERM

4 バイトの端末 ID のアドレス。Web サービス・プロバイダーの場合は、この値はヌルです。

## **UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。

## **UEPCHANN**

現行チャンネルの名前を含める 16 バイト・フィールドのアドレス。このパラメーターを使用して、チャンネルを明示的に識別できます。

## **UEPCONTR**

UEPCHANN で名前指定されたチャンネル上のデータ・コンテナの名前を含める 16 バイト・フィールドのアドレス。このコンテナは、アプリケーション・データ構造を保持します。

## **戻りコード**

### **UERCNORM**

処理が続行される。

### **UERCRRIP**

このパイプラインで継続しないでください。

## **XPI 呼び出し**

使用可能な XPI インターフェースはありません。

## **API コマンドおよび SPI コマンド**

以下のコマンドがサポートされています。

- **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**

## **出口 XWSPRROI**

コンテナが Web サービス・プロバイダー・アプリケーションによって処理される前、ただし XWSPRRWI 出口のいずれかのインスタンスが呼び出された後に、現行チャンネル上のコンテナにアクセスするには、XWSPRROI 出口を使用します。

この出口を使用して、API および SPI コマンドを発行し、Web サービスのビジネス・アプリケーションによって処理される情報を調べることができます。SOAP 障害を発行したり、いずれかの情報を更新したりすることはできません。グローバル・ユーザー出口プログラムの終了後、CICS はレジスター 15 に指定された戻りコードを無視します。

## **出口固有のパラメーター**

### **UEPTRANID**

4 バイトのトランザクション ID のアドレス。出口またはタスクはこのトランザクション ID の下で稼働します。

### **UEPUSER**

8 バイトのユーザー ID のアドレス。このユーザー ID は、トランザクション ID に関連付けられています。

### **UEPTERM**

4 バイトの端末 ID のアドレス。Web サービス・プロバイダーの場合は、この値はヌルです。

#### **UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。

#### **UEPCHANN**

現行チャンネルの名前を含める 16 バイト・フィールドのアドレス。このパラメーターを使用して、チャンネルを明示的に識別できます。

#### **UEPCONTR**

UEPCHANN で名前指定されたチャンネル上のデータ・コンテナの名前を含める 16 バイト・フィールドのアドレス。このコンテナは、アプリケーション・データ構造を保持します。

#### **戻りコード**

#### **UERCNORM**

処理が続行される。

#### **XPI 呼び出し**

使用可能な XPI インターフェースはありません。

#### **API コマンドおよび SPI コマンド**

以下のコマンドを使用できます。

- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**

### **出口 XWSPRROO**

Web サービス・プロバイダー・アプリケーションが Web サービス応答メッセージを発行した後で、かつ CICS が応答メッセージの本文を作成する前に、現行チャンネル上のコンテナにアクセスするには、XWSPRROO 出口を使用します。

この出口を使用して、API および SPI コマンドを発行し、現行チャンネル上のコンテナを調べることができます。SOAP 障害を発行したり、いずれかのコンテナを更新したりすることはできません。グローバル・ユーザー出口プログラムの終了後、CICS はレジスター 15 に指定された戻りコードを無視します。

#### **出口固有のパラメーター**

#### **UEPTRANID**

4 バイトのトランザクション ID のアドレス。出口またはタスクはこのトランザクション ID の下で稼働します。

#### **UEPUSER**

8 バイトのユーザー ID のアドレス。このユーザー ID は、トランザクション ID に関連付けられています。

#### **UEPTERM**

4 バイトの端末 ID のアドレス。Web サービス・プロバイダーの場合は、この値はヌルです。

#### **UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。

#### **UEPCHANN**

現行チャンネルの名前を含める 16 バイト・フィールドのアドレス。このパラメーターを使用して、チャンネルを明示的に識別できます。

#### **UEPCONTR**

UEPCHANN で名前指定されたチャンネル上のデータ・コンテナの名前を含める 16 バイト・フィールドのアドレス。このコンテナは、アプリケーション・データ構造を保持します。

## **UEPAPAB**

Web サービス・プロバイダー・アプリケーションが処理を正常に完了したかどうかを示す 1 バイト・フィールド。有効な値は、以下のとおりです。

### **UEPAPABY (X'80')**

Web サービス・プロバイダー・アプリケーションが異常終了しました。

### **UEPAPABN (X'40')**

Web サービス・プロバイダー・アプリケーションが処理を正常に完了しました。

## **UEPAPSF**

Web サービス・プロバイダー・アプリケーションが SOAP 障害を設定したかどうかを示す 1 バイト・フィールド。有効な値は、以下のとおりです。

### **UEPAPSFY (X'80')**

Web サービス・プロバイダー・アプリケーションは SOAP 障害を返しています。

### **UEPAPSFN (X'40')**

Web サービス・プロバイダー・アプリケーションは SOAP 障害を返していません。

## **戻りコード**

### **UERCNORM**

処理が続行される。

## **XPI 呼び出し**

使用可能な XPI インターフェースはありません。

## **API コマンドおよび SPI コマンド**

以下のコマンドがサポートされています。

- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**

## **出口 XWSPRRWO**

XWSPRRWO 出口の任意のインスタンスの後に、Web サービス・プロバイダー・アプリケーションによって処理される現行チャンネル上のコンテナにアクセスするには、XWSPRRWO 出口を使用します。

この出口を使用して、API および SPI コマンドを発行して、コンテナの情報の調査や更新を行い、SOAP 障害を発行することができます。例えば、追加の SOAP ヘッダーをアウトバウンド SOAP 応答に追加できます。現行のチャンネル・コンテナ・データに対する更新はすべて CICS によって処理され、リクエスターに返されます。グローバル・ユーザー出口プログラムの終了後、CICS はレジスター 15 に指定された戻りコードを無視します。

## **出口固有のパラメーター**

### **UEPTRANID**

4 バイトのトランザクション ID のアドレス。出口またはタスクはこのトランザクション ID の下で稼働します。

### **UEPUSER**

8 バイトのユーザー ID のアドレス。このユーザー ID は、トランザクション ID に関連付けられています。

### **UEPTERM**

4 バイトの端末 ID のアドレス。Web サービス・プロバイダーの場合は、この値はヌルです。

### **UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。



## **UEPCHANN**

現行チャンネルの名前を含める 16 バイト・フィールドのアドレス。このパラメーターを使用して、チャンネルを明示的に識別できます。

## **UEPCONTR**

UEPCHANN で名前指定されたチャンネル上のデータ・コンテナの名前を含める 16 バイト・フィールドのアドレス。このコンテナは、アプリケーション・データ構造を保持します。

## **UEPAPAB**

Web サービス・プロバイダー・アプリケーションが処理を正常に完了したかどうかを示す 1 バイト・フィールド。有効な値は、以下のとおりです。

### **UEPAPABY (X'80')**

Web サービス・プロバイダー・アプリケーションが異常終了しました。

### **UEPAPABN (X'40')**

Web サービス・プロバイダー・アプリケーションが処理を正常に完了しました。

## **UEPAPSF**

Web サービス・プロバイダー・アプリケーションが SOAP 障害を設定したかどうかを示す 1 バイト・フィールド。有効な値は、以下のとおりです。

### **UEPAPSFY (X'80')**

Web サービス・プロバイダー・アプリケーションは SOAP 障害を返しています。

### **UEPAPSFN (X'40')**

Web サービス・プロバイダー・アプリケーションは SOAP 障害を返していません。

## **戻りコード**

### **UERCNORM**

処理が続行される。

## **XPI 呼び出し**

使用可能な XPI インターフェースはありません。

## **API コマンドおよび SPI コマンド**

以下のコマンドを使用できます。

- 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

## **出口 XWSRQRWO**

処理されるためにトランスポートに渡される前に現行チャンネル上のコンテナにアクセスするには、XWSRQRWO 出口を使用します。この出口は CICS がアプリケーションの言語構造を Web サービス要求の本体に変換した後で、かつ CICS がオプションの XWSRQROO 出口点を処理する前に実行されます。

この出口を使用して、API および SPI コマンドを発行して、現在のチャンネルのコンテナ情報の調査や更新を行うことができます。この情報は XWSRQRWO 出口の任意のインスタンスおよびアウトバウンド Web サービス・プロバイダーの両方で使用可能です。SOAP 障害は発行できません。グローバル・ユーザー出口プログラムが終了した後に、CICS がパイプラインを続けないことを示すため、戻りコードをレジスター 15 に入れることができます。

## 出口固有のパラメーター

### UEPTRANID

4 バイトのトランザクション ID のアドレス。出口またはタスクはこのトランザクション ID の下で稼働します。

### UEPUSER

8 バイトのユーザー ID のアドレス。このユーザー ID は、トランザクション ID に関連付けられています。

### UEPTERM

4 バイトの端末 ID のアドレス。

### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。

### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。

### UEPCHANN

現行チャンネルの名前を含める 16 バイト・フィールドのアドレス。このパラメーターを使用して、チャンネルを明示的に識別できます。

### UEPCONTR

UEPCHANN で名前指定されたチャンネル上のデータ・コンテナの名前を含める 16 バイト・フィールドのアドレス。このコンテナは、アプリケーション・データ構造を保持します。

## 戻りコード

### UERCNORM

処理が続行される。

### UERCRRIP

このパイプラインで継続しないでください。

## XPI 呼び出し

使用可能な XPI インターフェースはありません。

## API コマンドおよび SPI コマンド

以下のコマンドを使用できます。

- **EXEC CICS DELETE CONTAINER**
- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**
- **EXEC CICS PUT CONTAINER**

## 出口 XWSRQROO

処理されるためにトランスポートに渡される前に現行チャンネル上のコンテナにアクセスするには、XWSRQROO 出口を使用します。この出口は XWSRQRWO 出口の任意のインスタンスが処理された後、しかも Web サービス・トランスポートのデータがアウトバウンドに流れる前に実行されます。

この出口を使用して、API および SPI コマンドを発行し、コンテナの情報を調べることができます。この情報はアウトバウンド Web サービス・プロバイダーによって処理されます。SOAP 障害は発行できません。グローバル・ユーザー出口プログラムの終了後、CICS はレジスター 15 に指定された戻りコードを無視します。

## 出口固有のパラメーター

### UEPTRANID

4 バイトのトランザクション ID のアドレス。出口またはタスクはこのトランザクション ID の下で稼働します。

### UEPUSER

8 バイトのユーザー ID のアドレス。このユーザー ID は、トランザクション ID に関連付けられています。

### UEPTERM

4 バイトの端末 ID のアドレス。

### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。

### UEPCHANN

現行チャンネルの名前を含める 16 バイト・フィールドのアドレス。このパラメーターを使用して、チャンネルを明示的に識別できます。

### UEPCONTR

UEPCHANN で名前指定されたチャンネル上のデータ・コンテナの名前を含める 16 バイト・フィールドのアドレス。このコンテナは、アプリケーション・データ構造を保持します。

## 戻りコード

### UERCNORM

処理が続行される。

### UERCRIIP

このパイプラインで継続しないでください。

## XPI 呼び出し

使用可能な XPI インターフェースはありません。

## API コマンドおよび SPI コマンド

以下のコマンドを使用できます。

- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**

## 出口 XWSRQROI

コンテナが Web サービス応答としてトランスポートに処理された後に現行チャンネル上のコンテナにアクセスするには、XWSRQROI 出口を使用します。XWSRQROI 出口は、CICS がアウトバウンド Web サービス・プロバイダーを処理した直後に呼び出されます。XWSRQRWI 出口のどのインスタンスより前に呼び出されるようにすることもできます。

この出口を使用して、API および SPI コマンドを発行し、コンテナの情報を調べることができます。この情報はアウトバウンド Web サービス・プロバイダーによって処理されます。SOAP 障害は発行できません。グローバル・ユーザー出口プログラムの終了後、CICS はレジスター 15 に指定された戻りコードを無視します。

## 出口固有のパラメーター

### UEPTRANID

4 バイトのトランザクション ID のアドレス。出口またはタスクはこのトランザクション ID の下で稼働します。

### UEPUSER

8 バイトのユーザー ID のアドレス。このユーザー ID は、トランザクション ID に関連付けられています。

#### **UEPTERM**

4 バイトの端末 ID のアドレス。

#### **UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。

#### **UEPCHANN**

現行チャンネルの名前を含める 16 バイト・フィールドのアドレス。このパラメーターを使用して、チャンネルを明示的に識別できます。

#### **UEPCONTR**

UEPCHANN で名前指定されたチャンネル上のデータ・コンテナの名前を含める 16 バイト・フィールドのアドレス。このコンテナは、アプリケーション・データ構造を保持します。

### **戻りコード**

#### **UERCNORM**

処理が続行される。

### **XPI 呼び出し**

使用可能な XPI インターフェースはありません。

### **API コマンドおよび SPI コマンド**

以下のコマンドを使用できます。

- **EXEC CICS GET CONTAINER**
- **EXEC CICS INQUIRE WEBSERVICE**

## **出口 XWSRQRWI**

コンテナが Web サービス応答としてトランスポートに処理された後に現行チャンネル上のコンテナにアクセスするには、XWSRQRWI 出口を使用します。XWSRQRWI 出口は、CICS がインバウンド Web サービス応答を処理した直後に呼び出されます。XWSRQROI 出口の任意のインスタンスの後にも呼び出されます。

この出口を使用して、API および SPI コマンドを発行し、コンテナの情報を調べたり更新したりすることができます。この情報はアウトバウンド Web サービス・プロバイダーによって処理され、関連する Web サービス・リクエスターのアプリケーションによって受け取られます。SOAP 障害は発行できません。グローバル・ユーザー出口プログラムの終了後、CICS はレジスター 15 に指定された戻りコードを無視します。

### **出口固有のパラメーター**

#### **UEPTRANID**

4 バイトのトランザクション ID のアドレス。出口またはタスクはこのトランザクション ID の下で稼働します。

#### **UEPUSER**

8 バイトのユーザー ID のアドレス。このユーザー ID は、トランザクション ID に関連付けられています。

#### **UEPTERM**

4 バイトの端末 ID のアドレス。

#### **UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。

#### **UEPCHANN**

現行チャンネルの名前を含める 16 バイト・フィールドのアドレス。このパラメーターを使用して、チャンネルを明示的に識別できます。

## UEPCONTR

UEPCHANN で名前指定されたチャンネル上のデータ・コンテナの名前を含める 16 バイト・フィールドのアドレス。このコンテナは、アプリケーション・データ構造を保持します。

## 戻りコード

### UERCNORM

処理が続行される。

## XPI 呼び出し

使用可能な XPI インターフェースはありません。

## API コマンドおよび SPI コマンド

以下のコマンドを使用できます。

- EXEC CICS DELETE CONTAINER
- EXEC CICS GET CONTAINER
- EXEC CICS INQUIRE WEBSERVICE
- EXEC CICS PUT CONTAINER

## 出口 XWSSRRWO

コンテナがトランスポートに渡されて処理される前に、CICS が保護された Web サービス・リクエストーとして動作して、現行チャンネル上のコンテナにアクセスするには、XWSSRRWO 出口を使用します。この出口は、CICS がアプリケーションの言語構造を Web サービス要求の本体に変換した後で、かつ CICS がオブションの XWSSRROO 出口点进行处理する前、また、パイプラインのセキュリティ・ハンドラーによって暗号化される前に実行されます。

この出口を使用して、API および SPI コマンドを発行し、コンテナの情報を調べたり更新したりすることができます。この情報は XWSSRRWO 出口の任意のインスタンスおよびアウトバウンド Web サービス・プロバイダーの両方で使用可能です。SOAP 障害は発行できません。グローバル・ユーザー出口プログラムが終了した後に、CICS がパイプラインを続けないことを示すため、戻りコードをレジスター 15 に指定することができます。

パイプラインにセキュリティ・ハンドラーがない場合、この出口は駆動されません。セキュリティ・ハンドラーがないパイプラインのインスタンスについては、[161 ページの『出口 XWSRQRWO』](#)トピックを参照してください。

## 出口固有のパラメーター

### UEPTRANID

4 バイトのトランザクション ID のアドレス。出口またはタスクはこのトランザクション ID の下で稼働します。

### UEPUSER

8 バイトのユーザー ID のアドレス。このユーザー ID は、トランザクション ID に関連付けられています。

### UEPTERM

4 バイトの端末 ID のアドレス。

### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。

### UEPCHANN

現行チャンネルの名前を含める 16 バイト・フィールドのアドレス。このパラメーターを使用して、チャンネルを明示的に識別できます。

## UEPCONTR

UEPCHANN で名前指定されたチャネル上のデータ・コンテナーの名前を含める 16 バイト・フィールドのアドレス。このコンテナーは、アプリケーション・データ構造を保持します。

## 戻りコード

### UERCNORM

処理が続行される。

### UERCRIPI

このパイプラインで継続しないでください。

## XPI 呼び出し

使用可能な XPI インターフェースはありません。

## API コマンドおよび SPI コマンド

以下のコマンドを使用できます。

- EXEC CICS DELETE CONTAINER
- EXEC CICS GET CONTAINER
- EXEC CICS INQUIRE WEBSERVICE
- EXEC CICS PUT CONTAINER

## 出口 XWSSRROO

コンテナーがトランスポートに渡されて処理される前に、CICS が保護された Web サービス・リクエスターとして動作して、現行チャネル上のコンテナーにアクセスするには、XWSSRROO 出口を使用します。この出口は XWSSRRWO 出口の任意のインスタンスが処理された後、かつ Web サービス・トランスポートのアウトバウンドに流れるデータが暗号化される前に実行されます。

パイプラインにセキュリティー・ハンドラーがない場合、この出口は駆動されません。セキュリティー・ハンドラーがないパイプラインのインスタンスについては、[162 ページの『出口 XWSRQROO』トピック](#)を参照してください。

## 出口固有のパラメーター

### UEPTRANID

4 バイトのトランザクション ID のアドレス。出口またはタスクはこのトランザクション ID の下で稼働します。

### UEPUSER

8 バイトのユーザー ID のアドレス。このユーザー ID は、トランザクション ID に関連付けられています。

### UEPTERM

4 バイトの端末 ID のアドレス。

### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。

### UEPCHANN

現行チャネルの名前を含める 16 バイト・フィールドのアドレス。このパラメーターを使用して、チャネルを明示的に識別できます。

### UEPCONTR

UEPCHANN で名前指定されたチャネル上のデータ・コンテナーの名前を含める 16 バイト・フィールドのアドレス。このコンテナーは、アプリケーション・データ構造を保持します。

## 戻りコード

### UERCNORM

処理が続行される。

## XPI 呼び出し

使用可能な XPI インターフェースはありません。

## API コマンドおよび SPI コマンド

以下のコマンドを使用できます。

- EXEC CICS GET CONTAINER
- EXEC CICS INQUIRE WEBSERVICE

## 出口 XWSSRROI

コンテナが Web サービス応答としてトランスポートに処理された後に、CICS が保護された Web サービス・リクエストとして動作して、現行チャネル上のコンテナにアクセスするには、XWSRQROI 出口を使用します。この出口は、CICS が Web サービス応答を処理した後で、かつ XWSSRRWI 出口のいずれのインスタンスよりも前に実行されます。

この出口を使用して、API および SPI コマンドを発行し、コンテナの情報を調べることができます。この情報はアウトバウンド Web サービス・プロバイダーによって処理されます。SOAP 障害は発行できません。グローバル・ユーザー出口プログラムの終了後、CICS はレジスター 15 に指定された戻りコードを無視します。

パイプラインにセキュリティー・ハンドラーがない場合、この出口は駆動されません。セキュリティー・ハンドラーがないパイプラインのインスタンスについては、[163 ページの『出口 XWSRQROI』トピック](#)を参照してください。

## 出口固有のパラメーター

### UEPTRANID

4 バイトのトランザクション ID のアドレス。出口またはタスクはこのトランザクション ID の下で稼働します。

### UEPUSER

8 バイトのユーザー ID のアドレス。このユーザー ID は、トランザクション ID に関連付けられています。

### UEPTERM

4 バイトの端末 ID のアドレス。

### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。

### UEPCHANN

現行チャネルの名前を含める 16 バイト・フィールドのアドレス。このパラメーターを使用して、チャネルを明示的に識別できます。

### UEPCONTR

UEPCHANN で名前指定されたチャネル上のデータ・コンテナの名前を含める 16 バイト・フィールドのアドレス。このコンテナは、アプリケーション・データ構造を保持します。

## 戻りコード

### UERCNORM

処理が続行される。

## XPI 呼び出し

使用可能な XPI インターフェースはありません。

## API コマンドおよび SPI コマンド

以下のコマンドを使用できます。

- EXEC CICS GET CONTAINER
- EXEC CICS INQUIRE WEBSERVICE

## 出口 XWSSRRWI

コンテナが Web サービス応答としてトランスポートに処理された後に、CICS が保護された Web サービス・リクエスターとして動作して、現行チャネル上のコンテナにアクセスするには、XWSSRRWI 出口を使用します。この出口は、CICS が Web サービス応答を処理した後、さらに XWSSRROI 出口の任意のインスタンスの後に実行されます。

この出口を使用して、API および SPI コマンドを発行し、コンテナの情報を調べたり更新したりすることができます。この情報は Web サービス・リクエスター・アプリケーションによって処理されます。SOAP 障害は発行できません。グローバル・ユーザー出口プログラムの終了後、CICS はレジスター 15 に指定された戻りコードを無視します。

パイプラインにセキュリティー・ハンドラーがない場合、この出口は駆動されません。セキュリティー・ハンドラーがないパイプラインのインスタンスについては、[164 ページの『出口 XWSRQRWI』トピック](#)を参照してください。

### 出口固有のパラメーター

#### UEPTRANID

4 バイトのトランザクション ID のアドレス。出口またはタスクはこのトランザクション ID の下で稼働します。

#### UEPUSER

8 バイトのユーザー ID のアドレス。このユーザー ID は、トランザクション ID に関連付けられています。

#### UEPTERM

4 バイトの端末 ID のアドレス。

#### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス。このアプリケーション・プログラム名は、**INVOKE SERVICE** または **INVOKE WEBSERVICE** コマンドを発行したプログラムの名前になります。

#### UEPCHANN

現行チャネルの名前を含める 16 バイト・フィールドのアドレス。このパラメーターを使用して、チャネルを明示的に識別できます。

#### UEPCONTR

UEPCHANN で名前指定されたチャネル上のデータ・コンテナの名前を含める 16 バイト・フィールドのアドレス。このコンテナは、アプリケーション・データ構造を保持します。

### 戻りコード

#### UERCNORM

処理が続行される。

### XPI 呼び出し

使用可能な XPI インターフェースはありません。

## API コマンドおよび SPI コマンド

以下のコマンドを使用できます。

- EXEC CICS DELETE CONTAINER
- EXEC CICS GET CONTAINER
- EXEC CICS INQUIRE WEBSERVICE



## プログラム制御プログラム出口 (XPCREQ、XPCERES、XPCREQC、XPCFTCH、XPCHAIR、XPCTA、および XPCABND)

これらの出口は、CICS プログラム制御操作 (プログラム・リンク要求、プログラム受信制御、トランザクション異常終了を含む) の前または後に呼び出されます。

### プログラム制御出口 XPCREQ、XPCERES、XPCREQC

これらの出口が EXEC インターフェース・プログラムで呼び出されるのは、リンク要求が処理される前、動的にルーティングされたリンク要求を CICS が処理する前、またはリンク要求が完了した後です。

#### XPCREQ

XPCREQ は、リンク要求が処理される前に EXEC インターフェース・プログラムによって呼び出されます。要求が分散プログラム・リンクである場合、XPCREQ 出口ルーチンは、リンクの両側 (つまり、クライアント領域とサーバー領域の両方) で呼び出されます。この出口プログラムは、アプリケーションのパラメーター・リスト (UEPCLPS 内) のアドレスを受け取り、このリストを必要に応じて変更することができます。例えば、この出口を使用して、分散プログラム・リンク要求の時点での SYSID を変更することができます。グローバル作業域 (GWA) 内で SYSID のリストを管理するアプリケーション・プログラムを作成することができます。グローバル・ユーザー出口プログラムは、GWA へのアクセスを取得し、そこに格納されている情報を使用して DPL 要求をリダイレクトすることができます。

注:

1. ローカル PROGRAM リソースの属性は、出口プログラムには渡されません。属性の値が必要な場合、出口プログラムは **EXEC CICS INQUIRE PROGRAM** コマンドを発行することができます。
2. XPCREQ を使用してターゲット SYSID を変更する場合は、以下に注意してください。
  - a. SYSID でリモート領域を指定すると、ローカル PROGRAM リソースへの参照は行われません。リモート領域では、プログラムは、クライアント領域の PROGRAM リソースで指定されている TRANSID の下ではなく、トランザクションの TRANSID の下で実行されます。
  - b. SYSID でローカル領域を指定すると、CICS は、SYSID が指定されたなかった場合と同様にリンク要求を処理します。ローカル PROGRAM リソースが受け入れられます。
  - c. XPCREQ 出口は、CICS コードで作成された内部要求と、アプリケーションで作成された要求によって呼び出されます。

#### XPCERES

XPCERES は、動的にルーティングされる以下のいずれかの種類のリンク要求を CICS が処理する前に、EXEC インターフェース・プログラムによって呼び出されます。

- 分散プログラム・リンク (DPL) 呼び出し
- Link3270 ブリッジ要求

XPCERES が呼び出されるのは以下の場合です。

- 以下の出口が使用可能になっている場合は、出口 XPCREQ の後で、XPCREQC の前。
  - XPCREQ 出口プログラムが要求をバイパスすることを選択した場合、XPCERES は呼び出されません。
  - XPCREQ 出口プログラムがコマンド・パラメーター・リストを変更した場合、XPCERES は変更された要求を処理する必要があります。
- 要求がルーティングされたターゲット領域。
- 使用可能になっている場合のみ。この出口は、DPL 要求および Link3270 ブリッジ要求の動的ルーティングが可能な場合にアプリケーション専有領域でのみ使用可能にしてください。
- CICS コードで作成された内部と、アプリケーションで作成された要求による場合。

以下の場合、XPCERES 出口は呼び出されません。

- 静的ルーティング要求の場合。

- 使用不可の場合。
- XPCREQ 出口プログラムが要求をバイパスすることを選択した場合。

XPCERES を使用して、リンク先プログラムが必要とするすべてのリソースがターゲット領域で使用可能であることを確認できます。プログラムが使用不可の場合、または必要なファイルが欠落している場合は、出口プログラムは動的ルーティング・プログラムに対し、要求を別の領域にルーティングする機会を与えることができます。戻りコード UERCRESU を設定します。CICS は以下の処理を実行します。

1. ルーティング・プログラムの COMMAREA で、CICS は DYRERROR フィールドを「F」(リソースが使用不可)に設定します。
2. ルート選択に失敗した場合、CICS はルーティング領域でルーティング・プログラムを呼び出します。
3. CICS は、ターゲット領域のミラーによって実行された EXEC CICS LINK コマンドに RESUNAVAIL 状態を返します。この状態はアプリケーション・プログラムには返されません。

CICS は、出口パラメーターの任意の値に対して出口プログラムにより加えられたすべての変更を無視します。出口プログラムは戻りコードを設定できますが、どのパラメーターも変更しません。

DPL 要求の動的ルーティングのガイダンス情報については、[DPL 要求の動的ルーティング](#)を参照してください。Link3270 ブリッジ要求の動的ルーティングのガイダンス情報については、[Link3270 ブリッジ・ロード・バランシングの使用](#)を参照してください。DPL 要求をルーティングするための動的ルーティング・プログラムの作成に関するプログラミング情報については、[DPL 要求の動的ルーティング](#)を参照してください。Link3270 ブリッジ要求をルーティングするための動的ルーティング・プログラムの作成に関するプログラミング情報については、[ブリッジ要求の動的ルーティング](#)を参照してください。

## XPCREQC

XPCREQC は、リンク要求が完了した後で呼び出されます。この出口により、EIBRESP フィールドまたは EIBRESP2 フィールドを使用して、アプリケーションに返す応答を渡すことができます。このような応答は、リンク要求に関する状況情報を最新状態に保つために使用される場合があります。例えば、接続が使用不可能なためにリンク要求が失敗した場合、XPCREQC は EIBRESP=500 (CICS では使用されない応答コード)を設定して失敗を示し、他の出口 XPCREQ を使用して、アプリケーションが適切なアクションを決定できるようにすることができます。

注：XPCREQC 出口は、CICS コードで作成された内部要求と、アプリケーションで作成された要求によって呼び出されます。

## 出口の XPCREQ

出口の XPCREQ は、リンク要求が処理される前に EXEC インターフェース・プログラムによって呼び出されます。

### 呼び出される状況

EXEC インターフェース・プログラムによってリンク要求が処理される前。

### 出口固有のパラメーター

#### UEPCLPS

コマンドのパラメーター・リストのアドレス。

#### UEPPCTOK

XPCREQC に受け渡される 4 バイトのトークンのアドレス。これにより、例えば、作業域を出口の XPCREQC に受け渡すことができます。

#### UEPRCODE

EIBRCODE の 6 バイトの 16 進数コピーのアドレス。

#### UEPRECUR

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

#### UEPRES

EIBRESP の 4 バイトのコピーのアドレス。

#### UEPRES2

EIBRESP2 の 4 バイトのコピーのアドレス。

**UEPTSTOK**

タスクの存続中に有効な 4 バイトのトークンのアドレス。 タスク・トークン UEPTSTOK の使用を参照してください。

**UEPRSRCE**

EIB リソース値の EIBRSRCE の 8 文字のコピーのアドレス。

**UEP\_PC\_PBTOK**

z/OS ワークロード・マネージャー (WLM) の Performance Block Token を含む 4 バイトのフィールドのアドレス。 出口プログラムは、このトークンを使用して、WLM Performance Block の情報 (サービス・クラス・トークン SERVCLS など) にアクセスすることができます。 このアクセスを行うためには、出口プログラムは、Performance Block Token を MONTKN の入力パラメーターとして受け渡す WLM EXTRACT マクロの IWMMEXTR を使用する必要があります。 IWMMEXTR マクロについて詳しくは、z/OS MVS プログラミング: ワークロード管理サービスを参照してください。

出口プログラムで、Performance Block の変更を試みないでください。 そのような変更を試みた場合の結果は、予測不能です。

**戻りコード****UERCIBYP**

プログラム制御は、この要求を無視します。

**UERCNORM**

処理が続行される。

**UERCPURG**

タスクは XPI 呼び出し中にページされます。

**XPI 呼び出し**

すべてを使用できます。

この出口では、XPI GETMAIN 呼び出しおよび FREEMAIN 呼び出しの使用が許可されていますが、代わりに EXEC CICS GETMAIN コマンドおよび FREEMAIN コマンドを使用することをお勧めします。

**API および SPI 呼び出し**

次のコマンドを除き、すべてを使用できます。

EXEC CICS SHUTDOWN

EXEC CICS XCTL

**出口 XPCERES**

出口 XPCERES は、ルーティング領域が「リソース使用不可」状態をサポートしている場合に、この領域に動的にルーティングされたプログラム・リンク要求または Link3270 ブリッジ要求の処理の前に、EXEC インターフェース・プログラムによって呼び出されます。

**出口固有のパラメーター**

注: CICS は、出口パラメーターの任意の値に対して出口プログラムにより加えられたすべての変更を無視します。 出口プログラムは戻りコードを設定できますが、どのパラメーターも変更しません。

**UEPCLPS**

コマンドのパラメーター・リストのアドレス。

**UEPPCTOK**

XPCREQC に受け渡される 4 バイトのトークンのアドレス。

**UEPRCODE**

EIBRCODE の 6 バイトの 16 進数コピーのアドレス。

**UEPRECUR**

ハーフワードの再帰カウンターのアドレス。 XPCERES 出口は同じトランザクションで再帰的に呼び出すことができないため、このフィールドの値は常に 0 です。

**UEPRES P**

EIBRESP の 4 バイトのコピーのアドレス。

**UEPRES P2**

EIBRESP2 の 4 バイトのコピーのアドレス。

**UEPTSTOK**

タスクの存続中に有効な 4 バイトのトークンのアドレス。 [タスク・トークン UEPTSTOK の使用](#)を参照してください。

**UEPRSRCE**

EIB リソース値の EIBRSRCE の 8 文字のコピーのアドレス。

**UEP\_PC\_PBTOK**

z/OS ワークロード・マネージャー (WLM) の Performance Block Token を含む 4 バイトのフィールドのアドレス。 出口プログラムは、このトークンを使用して、WLM Performance Block の情報 (サービス・クラス・トークン SERVCLS など) にアクセスすることができます。 このアクセスを行うためには、出口プログラムは、Performance Block Token を MONTKN の入力パラメーターとして受け渡す WLM EXTRACT マクロの IWMMECTR を使用する必要があります。 IWMMECTR マクロについて詳しくは、[z/OS MVS プログラミング: ワークロード管理サービス](#)を参照してください。

出口プログラムで、Performance Block の変更を試みないでください。 そのような変更を試みた場合の結果は、予測不能です。

**戻りコード****UERCNORM**

処理が続行される。

**UERCPUrg**

タスクは XPI 呼び出し中にページされます。

**UERCRESU**

必要なリソースが使用不可です。 この値を設定すると、CICS がルーティングされた要求を拒否し、ルーティング・プログラムの通信域の DYRError フィールドに値「F」(リソース使用不可) を返すことになります。

**XPI 呼び出し**

すべてを使用できます。

この出口では、XPI GETMAIN 呼び出しおよび FREEMAIN 呼び出しの使用が許可されていますが、代わりに EXEC CICS GETMAIN コマンドおよび FREEMAIN コマンドを使用することをお勧めします。

**API および SPI 呼び出し**

EXEC CICS SHUTDOWN および EXEC CICS XCTL を除くすべてを使用できます。

**出口 XPCREQC**

出口 XPCREQC は、プログラム制御リンク要求の完了時に呼び出されます。

**呼び出される状況**

プログラム制御リンク要求の完了時。

**出口固有のパラメーター****UEPCLPS**

コマンドのパラメーター・リストのアドレス。

**UEPPCTOK**

XPCREQ から渡された 4 バイトのトークンのアドレス。 これにより XPCREQ は、例えば、作業域を XPCREQC に渡すことができます。

**UEPRCODE**

EIBRCODE の 6 バイトの 16 進数コピーのアドレス。

**UEPRECUR**

ハーフワードの再帰カウンターのアドレス。 このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

**UEPRES**

EIBRESP の 4 バイトのコピーのアドレス。

**UEPRES2**

EIBRESP2 の 4 バイトのコピーのアドレス。

## **UEPTSTOK**

タスクの存続中に有効な 4 バイトのトークンのアドレス。 タスク・トークン UEPTSTOK の使用を参照してください。

## **UEPRSRCE**

EIB リソース値の EIBRSRCE の 8 文字のコピーのアドレス。

## **UEP\_PC\_REMOTE\_SYSTEM**

要求をリモート領域に送信する場合は、そのリモート領域の 4 バイトの名前が含まれた領域のアドレス。(リモート領域は、例えば、EXEC CICS LINK コマンドの SYSID オプション、機能シップ、ワークロード管理、または PROGRAM 定義の REMOTESYSTEM オプションで指定された可能性があります。)

要求をローカル領域で実行する場合、このパラメーターはブランクが含まれた 4 バイト領域のアドレスです。

## **UEP\_PC\_REMOTE\_NAME**

プログラムをリモート・システムで実行する場合、リモート・システムで認識されているとおりのプログラムの名前が含まれた領域のアドレス。

## **UEP\_PC\_PBTOK**

z/OS ワークロード・マネージャー (WLM) の Performance Block Token を含む 4 バイトのフィールドのアドレス。 出口プログラムは、このトークンを使用して、WLM Performance Block の情報 (サービス・クラス・トークン SERVCLS など) にアクセスすることができます。 このアクセスを行うためには、出口プログラムは、Performance Block Token を MONTKN の入力パラメーターとして受け渡す WLM EXTRACT マクロの IWMMEXTR を使用する必要があります。 IWMMEXTR マクロについて詳しくは、z/OS MVS プログラミング: ワークロード管理サービスを参照してください。

出口プログラムで、Performance Block の変更を試みないでください。 そのような変更を試みた場合の結果は、予測不能です。

## **戻りコード**

### **UERCNORM**

処理が続行される。

### **UERCPURG**

タスクは XPI 呼び出し中にページされます。

## **XPI 呼び出し**

すべてを使用できます。

この出口では、XPI GETMAIN 呼び出しおよび FREEMAIN 呼び出しの使用が許可されていますが、代わりに EXEC CICS GETMAIN コマンドおよび FREEMAIN コマンドを使用することをお勧めします。

## **API および SPI 呼び出し**

次のコマンドを除き、すべてを使用できます。

EXEC CICS SHUTDOWN

EXEC CICS XCTL

**注:** 再帰的コマンドを発行する際には、ループが生じないように注意してください。 例えば、XPCREQ 出口または XPCREQC 出口からファイル制御要求を発行する場合に、ループに入らないようにユーザーが注意する必要があります。

再帰カウンター UEPRECUR を使用することをお勧めします。

## **コマンド・パラメーター構造**

コマンド・パラメーター構造は、一連のアドレスで構成されます。 最初のアドレスは、EXEC インターフェース記述子 (EID) を指します。 これは、要求のタイプを記述するビット・ストリングからなり、要求で指定された各キーワードを識別します。

残りのアドレスは、要求に関連付けられたデータ部分を指します。 例えば、2 番目のアドレスは常にプログラム名を指しています。 リスト内のパラメーターを調べて、キーワードの値を判別することができます。 要求で指定されたパラメーターの値を変更することもできます。 例えば、要求に関連するプログラムの名

前を変更したり、リンク要求をリモート・システムにルーティングするために SYSID を追加したりすることができます。

### パラメーター・リスト終了標識

パラメーター・リスト内の最後のアドレス・セットの高位ビットが、リスト内の最後のアドレスであることを示すために設定されます。ユーザー出口プログラムから戻ったときに、CICS は、パラメーター・リストをスキャンして高位ビットを探し、最終パラメーターを検出します。したがって、パラメーター・リストの長さを変更した場合は、高位ビットをリセットしてどれが新しい最終アドレスかを示す必要もあります。

例えば、パラメーター・リストで最初の 2 つのアドレスのみ (EID のアドレスである PC\_ADDR0 と、リンク要求に指定されたプログラムの名前のアドレスである PC\_ADDR1) を指定した場合、PC\_ADDR1 内の高位ビットがオンに設定されます。PC\_ADDR7 に SYSID のアドレスを設定してパラメーター・リストを拡張した場合、PC\_ADDR1 の高位ビットを設定解除し、代わりに PC\_ADDR7 の高位ビットをオンに設定する必要があります。

XPCREQC の完了後に、XPCREQ が呼び出される前の元のパラメーター・リストが復元されます。その後に、EDF により、実行前および実行後の元のコマンドが表示されます。**EDF は、出口によって行われた変更は表示しません。**

### UEPCLPS 出口固有パラメーター

UEPCLPS 出口固有パラメーターは、コマンド・レベルのパラメーター構造のアドレスで、出口 XPCREQ および XPCREQC に含まれています。

コマンド・レベルのパラメーター構造には、PC\_ADDR0 から PC\_ADDRA までの 11 のアドレスが含まれています。この構造は、DSECT PC\_ADDR\_LIST 内に定義されますが、それは COPY DFHPCEDS というステートメントを組み込むことによって、出口プログラムにコピーする必要があります。

コマンド・レベルのパラメーター・リストは、以下のように構成されています。

#### PC\_ADDR0

EXEC インターフェース 記述子 (EID) と呼ばれる 7 バイト領域のアドレス。以下のように構成されます。

- PC\_GROUP
- PC\_FUNC
- PC\_BITS1
- PC\_BITS2
- PC\_EIDOPT4
- PC\_EIDOPT5
- PC\_EIDOPT6

#### PC\_GROUP

常に 'X'0E' で、これがプログラム制御要求であることを示します。

#### PC\_FUNC

要求のタイプを定義する 1 バイト。XPCREQ および XPCREQC の場合は常に 'X'02' であり、LINK 要求を示します。

#### PC\_BITS1

値を含むどのキーワードが指定されたかを定義する存在ビット。キーワードに関連付けられた値を取得するには、コマンド・レベルのパラメーター・リストから適切なアドレスを使用する必要があります。このアドレスを使用する前に、アドレスが有効であるかどうか判断するために、関連付けられた存在ビットを確認する必要があります。存在ビットがオフに設定されている場合は、キーワードが要求に指定されなかったため、このアドレスを使用してはなりません。存在ビットのシンボリック値と 16 進値は、以下のとおりです。



**PC\_EXIST1 (X'80')**

要求にキーワード PROGRAM が含まれている場合に設定されます。設定される場合は、**PC\_ADDR1** が意味を持ちます。(LINK 要求の場合は、このビットを常に設定する必要があります。)

**PC\_EXIST2 (X'40')**

要求に COMMAREA パラメーターが指定されている場合に設定されます。設定される場合は、**PC\_ADDR2** が意味を持ちます。

**PC\_EXIST3 (X'20')**

要求に LENGTH パラメーターが指定されている場合に設定されます。設定される場合は、**PC\_ADDR3** が意味を持ちます。

**PC\_EXIST4 (X'10')**

要求に INPUTMSG パラメーターが指定されている場合に設定されます。設定される場合は、**PC\_ADDR4** が意味を持ちます。

**PC\_EXIST5 (X'08')**

要求に INPUTMSGLEN パラメーターが指定されている場合に設定されます。設定される場合は、**PC\_ADDR5** が意味を持ちます。

**PC\_EXIST6 (X'04')**

要求に DATALENGTH パラメーターが指定されている場合に設定されます。設定される場合は、**PC\_ADDR6** が意味を持ちます。

**PC\_EXIST7 (X'02')**

要求に SYSID パラメーターが指定されている場合に設定されます。設定される場合は、**PC\_ADDR7** が意味を持ちます。

**PC\_EXIST8 (X'01')**

要求に TRANSID パラメーターが指定されている場合に設定されます。設定される場合は、**PC\_ADDR8** が意味を持ちます。

**PC\_BITS2**

以下のいずれかの値を含む 1 バイト。

**PC\_EXIST9 (X'80')**

使用されません。

**PC\_EXISTA (X'40')**

要求に CHANNEL パラメーターが指定されている場合に設定されます。設定される場合は、**PC\_ADDRA** が意味を持ちます。

**PC\_EIDOPT4**

プログラム制御では使用されません。

**PC\_EIDOPT5**

プログラム制御では使用されません。

**PC\_EIDOPT6**

要求が SYNCONRETURN オプションを指定するかどうかを示します。指定する場合は、X'80' が設定されます。

**PC\_ADDR1**

PROGRAM パラメーターからのプログラム名を含む 8 バイトの領域のアドレス。

**PC\_ADDR2**

COMMAREA データのアドレス。

**PC\_ADDR3**

ハーフワード・バイナリー値として COMMAREA の長さを含む 2 バイト領域のアドレス。

**PC\_ADDR4**

INPUTMSG データのアドレス。

**PC\_ADDR5**

ハーフワード・バイナリー値として INPUTMSG の長さを含む 2 バイト領域のアドレス。

#### **PC\_ADDR6**

COMMAREA から送信されるデータの量を定義する、DATALENGTH パラメーターに指定された長さを  
含む 2 バイト領域のアドレス。この長さは、ハーフワード・バイナリー値として保管されます。

#### **PC\_ADDR7**

SYSID パラメーターに指定された、LINK 要求のシップ先であるリモート・システムの名前 (4 バイト)  
のアドレス。

#### **PC\_ADDR8**

TRANSID パラメーターに指定された、リモート・システム内で接続されるミラー・トランザクション  
の名前 (4 バイト) のアドレス。

#### **PC\_ADDR9**

使用されません。

#### **PC\_ADDRA**

CHANNEL パラメーターに指定された、16 バイトのチャンネル名のアドレス。

### **コマンド・パラメーター構造内のフィールドの変更**

プログラム制御に渡される一部のフィールドは要求への入力として使用され、一部は出力フィールドとし  
て使用され、さらに一部は入力および出力の両方に使用されます。ユーザー出口プログラムでフィールド  
を変更するために使用する方法は、フィールドの使用方法に応じて異なります。

#### **入力フィールドの変更**

入力フィールドを変更する正しい方法としては、そのフィールドの新規コピーを作成し、新しいデータを  
指すようにコマンド・パラメーター・リスト内のアドレスを変更します。

**注:** コマンド・パラメーター・リストによって指定されたデータを変更することによって、入力フィールド  
を変更してはなりません。このようにすると、アプリケーション・プログラムに属するストレージが破壊  
され、プログラムがフィールドの再使用を試みると障害が発生する可能性があります。

#### **出力フィールドの変更**

176 ページの『[入力フィールドの変更](#)』で説明されている手法は、出力フィールドの変更には適していま  
せん。(結果が、アプリケーションの領域ではなく新しい領域に返されるため、アプリケーションで表示で  
きません。)

出力フィールドの変更は、コマンド・レベル・パラメーター・リストが指すデータを変更することによっ  
て行います。出力フィールドの場合は、アプリケーションのデータを所定の位置で変更することができま  
す。これは、アプリケーションが、そのフィールドが何らかの方法で変更されることを予期しているため  
です。

#### **EID の変更**

EID を変更することで要求に対して大きな変更を行うことはできません。例えば、LINK 要求を別のタイプ  
のプログラム制御要求に変更することはできません。ただし、SYSID の存在ビットをオンにして要求をリ  
モート・システムに送信できるようにするなど、要求に対する小さな変更を行うことはできます。

以下のリストは、EID 内の変更可能なビットを示しています。EID のその他の部分を変更しようとしても、  
すべて無視されます。

#### **PC\_BITS1**

##### **X'40'**

COMMAREA の存在ビット

##### **X'20'**

LENGTH の存在ビット

##### **X'10'**

INPUTMSG の存在ビット

##### **X'08'**

INPUTMSGLEN の存在ビット

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

DATALENGTH の存在ビット



**X'02'**

SYSID の存在ビット

**X'01'**

TRANSID の存在ビット

**PC\_BITS2**

**X'40'**

CHANNEL の存在ビット

**PC\_EIDOPT5**

PC リンク要求では使用されません。

EID 内のビットは所定の位置で変更する必要があります。EID へのポインターを変更しないでください。  
(このような試みは CICS ですべて無視されます。)

EID は、アプリケーション・プログラムに戻る前に、元の値にリセットされます。つまり、EID に対して行われた変更は、プログラム制御要求の期間中のみ保持されます。

ユーザー出口プログラムは、EID に大きな変更を加えることはできません。

### プログラム制御要求トークン UEPPCTOK の使用

UEPPCTOK は、同じプログラム制御要求に対して XPCREQ ユーザー出口と XPCREQC ユーザー出口の間で情報を渡すために使用できる、4 バイト領域のアドレスを提供します。

例えば、XPCREQ ユーザー出口によって取得されたストレージの部分 (XPCREQC ユーザー出口によって解放する必要がある) のアドレスは、UEPPCTOK フィールドで渡すことができます。

UEPPCTOK によってアドレス指定された領域を使用して、文字データを直接格納しないでください。  
UEPPCTOK によってアドレス指定された 4 バイトの領域は常に、受け渡すデータとストレージを識別するための目印を含むストレージ域のアドレスを格納している必要があります。CICS IA® は UEPPCTOK を使用して、自身のストレージ域のアドレスを格納します。CICS IA グローバル・ユーザー出口は、実行時に UEPPCTOK 内のアドレスを検査して、既に出口に対して存在するストレージ域を特定します。

UEPPCTOK は、その内容が要求の終了時に破棄される可能性があるため、単一のプログラム制御要求の期間のみ使用可能です。グローバル・ユーザー出口の連続する呼び出しの間で情報を渡す必要がある場合は、タスク・トークン UEPTSTOK を使用してこれを行うことができます。UEPTSTOK について詳しくは、[タスク・トークン UEPTSTOK の使用](#)を参照してください。

## EIB

EIBRSRCE、EIBRCODE、EIBRESP、および EIBRESP2 が出口に渡されるので、以下のことができます。

- XPCREQ および XPCREQC 内で、完了情報またはリソース情報を変更または設定する。
- XPCREQC 内の完了情報を調べる。

パラメーター・リストに指定した EIBRSRCE、EIBRCODE、EIBRESP および EIBRESP2 のコピーを更新することができます。XPCREQC の完了後に、または XPCREQ に戻りコード「bypass」を指定してある場合に、プログラム制御が値を実 EIB にコピーします。

有効なプログラム制御応答を設定する必要があります。EIBRCODE、EIBRESP、および EIBRESP2 の 3 つすべてを、整合性のある値セット (有効な完了を記述するためにプログラム制御によって設定されるような値セット) に設定する必要があります。**プログラム制御では、EIBRCODE、EIBRESP、および EIBRESP2 の整合性は監視されません。**EIBRCODE、EIBRESP、および EIBRESP2 の値の設定に役立つように、プログラム制御で使用される値が DFHPCEDS に指定されます。

### XPCREQ および XPCREQC の使用方法の例

この例では、XPCREQ および XPCREQC を使用して LINK 要求を複数の異なる CICS 領域にルーティングし、簡便なロード・バランシング・メカニズムを提供します。

この例では、出口の機能のみを示します。この例は、ロード・バランシング機能を実行するのに理想的な方法を示すものではありません。この例の目的上、グローバル作業域 (GWA) が既に存在し、そこに使用可能な SYSID のリストおよび各 SYSID で現在処理中の LINK 要求の数が既に格納されていることが前提となっています。

## XPCREQ の場合

1. グローバル作業域 (GWA) をスキャンして、適切な CICS 領域、例えば、現在処理中の LINK 要求が最も少ない領域) を見つけます。
2. 要求をどのシステムにルーティングするかを決めたら、そのシステムの使用回数を増やします。
3. この要求の SYSID を格納するための 4 バイト領域を取得します (これは、GWA から割り振ると、GETMAIN を発行しなくても済みます)。GETMAIN を発行してこの領域を取得する場合は、UEPPCTOK を取得したストレージのアドレスに設定してください。
4. PC\_ADDR7 を取得した 4 バイト領域のアドレスに設定します。
5. この時点で PC\_ADDR7 を設定するとそれが最終アドレスになる場合は、アドレスの高位ビットを設定し、以前の最終アドレスであった高位ビットを設定解除します。
6. SYSID が指定されていることを示すために、PC\_BITS1 内の X'02' 存在ビットをオンに設定します。
7. CICS に戻ります。

## XPCREQC の場合

1. グローバル作業域 (GWA) をスキャンし、SYSID パラメーターで指定された CICS 領域のエントリーを見つけてみます。
2. このシステムの使用回数を減らします。
3. SYSID を保持する領域を取得するために XPCREQ で GETMAIN が発行されていた場合は、UEPPCTOK に保持されているアドレスに対して FREEMAIN を発行します。
4. CICS に戻ります。

## 出口 XPCFTCH

CICS に対して定義されているプログラム (内部 CICS モジュールを含む) が、トランザクション内の最初のプログラムとして、あるいは LINK、XCTL、または HANDLE ABEND PROGRAM 要求の結果として制御を受け取る前に、XPCFTCH が呼び出されます。

この出口を使用して、プログラムにリンクするときに使用するエントリー・アドレスを変更できます。出口がゼロの戻りコードを設定している場合、またはアドレスをゼロに変更した場合には、元のアプリケーション・プログラムのエントリー・アドレスが使用されます。

元のプログラムが呼び出される前に、制御を AMODE(64)、AMODE(31)、または AMODE(24) アセンブラー・アプリケーション・プログラムまたはルーチンに渡すには、この出口を使用します。このアセンブラー・プログラムは、処理を完了した後に、分岐命令を使用して制御を元のプログラムのエントリー・ポイントに返す必要があります。出口を使用して、元のプログラム以外のプログラムを呼び出さないでください。元のプログラム以外のプログラムを呼び出した場合、その結果は予測不能となるためです。

XPLINK オプションを使用してコンパイルされた C または C++ プログラムに対して XPCFTCH が呼び出されると、ユーザー出口で行われる可能性のあるエントリー・ポイント・アドレスの変更を無視することを示すフラグが設定されます。

変更されたエントリー・アドレスが指定されている場合、呼び出されたプログラムは、元のアプリケーション・プログラムが制御を受け取ることになっていた実行キー (つまり、元のプログラムのリソース定義の EXECKEY オプションに指定された実行キー) 内で制御を受け取ります。

### 呼び出される状況

アプリケーション・プログラムが制御を受け取る前。

### 出口固有のパラメーター

#### UEPPCDS

プログラム関連および端末関連の情報が格納され、DSECT DFHPCUE を使用してマッピングできるストレージ域のアドレス。XPCFTCH が呼び出される時には、以下の DFHPCUE フィールドが意味を持ちます。

## **PCUE\_CONTROL\_BITS**

- 1 バイト・フラグ・フィールド。PCUECBTE の設定は、トランザクションが端末にリンクされていることを示します。
- PCUENOTX (X'40') の設定は、コマンド・レベルのプログラムではないことを示します。
- PCUE\_CONTROL\_BITS 内のフラグ PCUE\_NO\_MODIFY は、変更されたエントリー・アドレスがサポートされていないことを示します。設定されている場合、XPCFTCH からの UERCMEA の戻りコードはすべて無視されます。CICS は、XPLINK オプションを使用してコンパイルされた C および C++ プログラムに対して XPCFTCH を呼び出す前に、このフラグを設定します。
- PCUE\_REAL (X'20') の設定は、実際のエントリー・ポイントが PCUE\_REAL\_ENTRY で設定されていることを示します。

## **PCUE\_TASK\_NUMBER**

タスク番号を含む 3 文字パック 10 進数フィールド。

## **PCUE\_TRANSACTION\_ID**

元のトランザクションの ID を含む 4 文字のフィールド。この ID は、現在のトランザクション ID とは異なる場合があります。

## **PCUE\_TERMINAL\_ID**

端末 ID (該当する場合) を含む 4 文字のフィールド。

## **PCUE\_PROGRAM\_NAME**

制御を受け取るプログラムの名前を含む 8 文字のフィールド。

## **PCUE\_PROGRAM\_LANGUAGE**

制御を受け取るプログラムの言語を含む 3 文字のフィールド。

## **PCUE\_LOAD\_POINT**

プログラムのロード・ポイント。

## **PCUE\_ENTRY\_POINT**

プログラムのエントリー・ポイント。

## **PCUE\_AMOD**

プログラムのアドレッシング・モードは AMODE(31) です。このフィールドは、既存の出口プログラムとの互換性のために提供されています。

## **PCUE\_AMOD\_31**

プログラムのアドレッシング・モードは AMODE(31) です。このフィールドを PCUE\_AMOD に優先して使用してください。

## **PCUE\_AMOD\_64**

プログラムのアドレッシング・モードは AMODE(64) です。

## **PCUE\_PROGRAM\_SIZE**

プログラムのサイズ (バイト数) を含むフルワード。

## **PCUE\_COMMAREA\_ADDRESS**

プログラムに通信域がある場合、その通信域のアドレス。

## **PCUE\_COMMAREA\_SIZE**

プログラムに通信域がある場合、その通信域の長さを含むフルワード。

## **PCUE\_LOGICAL\_LEVEL**

プログラムの論理レベルを含むフルワード。

## **PCUE\_BRANCH\_ADDRESS**

フルワード。このフィールドを使用して、代替エントリー・アドレスを指定します。代替プログラムが AMODE (31) を実行することを指定するには、最上位ビットを設定します。

## **PCUE\_REAL\_ENTRY**

z/OS 1.7 以降、このフィールドは Language Environment® 規格適合プログラムの実際のエントリー・ポイントを指定するようになっています。それ以前に使用可能だったのは PCUE\_ENTRY\_POINT だけでしたが、Language Environment 規格適合プログラムの場合に把握する必要のあるエントリー・ポイントは、このフィールドに含まれていませんでした。

注: z/OS 1.7 では、このフィールドが、APAR PQ43992 で提起された問題に対するソリューションになります。

#### **PCUE\_CHANNEL\_NAME**

アプリケーション・プログラムを起動するために使用するチャンネル (つまり、プログラムの現行のチャンネル) の名前を含む 16 バイト・フィールドのアドレス。チャンネルがない場合、このフィールドはブランクに設定されます。

#### **PCUE\_INVOKING\_PROGRAM\_NAME**

現在のプログラムを呼び出したプログラムの名前を含む 8 文字のフィールド。

### **戻りコード**

#### **UERCNORM**

処理が続行される。

#### **UERCPURG**

タスクは XPI 呼び出し中にページされます。

#### **UERCMEA**

エントリー・アドレスが変更されました。

### **XPI 呼び出し**

すべて使用できます。

### **サンプル出口プログラム**

DFH\$PCEX

## **出口 XPCHAIR**

出口 XPCHAIR は、HANDLE ABEND LABEL ルーチンに制御を渡す前に呼び出されます。

この状態が起こるのは、プログラム異常終了が原因で内部異常終了ルーチンへの分岐が発生した場合のみです。HANDLE ABEND 要求に PROGRAM が指定されていると、出口 XPCFTCH が呼び出されます (178 ページの『[出口 XPCFTCH](#)』を参照)。XPCHAIR の出口を使用して、代替ハンドル異常終了アドレスを提供することができます。この出口でゼロの戻りコードが設定されている場合、またはゼロの代替アドレスが設定されている場合は、CICS はアプリケーション・プログラムの指定の内部ルーチンに制御を渡します。

XPLINK オプションを使用してコンパイルされた C または C++ プログラム、または AMODE(64) プログラムに対して、この出口が呼び出されることはありません。

変更されたエントリー・アドレスが提供される場合は、以下のようになります。

- 呼び出されたコードは、内部異常終了ルーチンが受け取るようになっていた実行キー (つまり EXEC CICS HANDLE ABEND LABEL コマンドが発行されたときに有効だったキー) 内で制御を受け取ります。
- 再開アドレスは、COBOL アプリケーションの場合はレジスター 14 に、アセンブラー・アプリケーションの場合はレジスター 15 に置かれます。アプリケーションが混合環境で実行されている場合、出口プログラムで独自の基底レジスターをセットアップしなければならない場合があります。例えば、以下のコードを使用してアドレッシング機能をセットアップすることができます。

```
BASSM 15,0  
USING *,15
```

### **呼び出される状況**

HANDLE ABEND ルーチンに制御が渡される前。

### **出口固有のパラメーター**

#### **UEPPCDS**

プログラム関連および端末関連の情報が格納され、DSECT DFHPCUE を使用してマッピングできるストレージ域のアドレス。XPCHAIR が呼び出されるときには、以下の DFHPCUE フィールドが意味を持ちます。

**PCUE\_CONTROL\_BITS**

1 バイト・フラグ・フィールド。PCUECBTE の設定は、トランザクションが端末にリンクされていることを示します。

**PCUE\_TASK\_NUMBER**

タスク番号を含む 3 文字パック 10 進数フィールド。

**PCUE\_TRANSACTION\_ID**

トランザクション ID が含まれた 4 文字のフィールド。

**PCUE\_TERMINAL\_ID**

端末 ID (該当する場合) を含む 4 文字のフィールド。

**PCUE\_PROGRAM\_NAME**

HANDLE ABEND LABEL コマンドを発行したプログラムの名前が含まれている 8 文字のフィールド。

**PCUE\_LOGICAL\_LEVEL**

プログラムの論理レベルを含むフルワード。

**PCUE\_BRANCH\_ADDRESS**

フルワード。このフィールドを使用して、代替異常終了ルーチンのアドレスを指定します。代替異常終了ルーチンが AMODE (31) を実行することを指定するには、最上位ビットを設定します。

**UEPTACB**

異常終了のトランザクション異常終了制御ブロック (TACB) のアドレス。プログラム・チェックが原因で異常終了が発生した場合、TACB 内の情報には以下が含まれています。

- プログラム状況ワード (PSW)。
- 異常終了時のレジスター。
- 異常終了時に現行であったサブスペース、アクセス・レジスター、およびベクトル・レジスターについての詳細。
- 中断イベント・アドレス・レジスター (BEAR)。
- 変換例外アドレス (TEA)。

DFHTACB TYPE=DSECT マクロを使用して TACB をマップすることができます。

**戻りコード****UERCNORM**

処理が続行される。

**UERCPURG**

タスクは XPI 呼び出し中にページされます。

**UERCMEA**

代替異常終了ルーチンのアドレスが指定されています。

**XPI 呼び出し**

すべて使用できます。

**出口 XPCTA**

出口 XPCTA は、トランザクションの異常終了の直後で、既存の環境を変更してタスクを再開できないようにする処理の前に呼び出されます。

XPCTA 出口を使用して、以下のことを行うことができます。

- CICS に異常終了の処理を任せずに、再開アドレスを設定する。
- 制御を入れて渡すサブスペースを指定する。

再開アドレスが返されると、レジスター 0 から 13 および 15 は異常終了時の値に復元されます。レジスター 14 は、再開アドレスに分岐するために使用されます。この出口でゼロの戻りコードが設定されている場合、または再開アドレスがゼロの場合は、CICS は異常終了を処理します。

プログラム・チェックまたはオペレーティング・システムの異常終了の結果としてトランザクション異常終了が発生した場合は、XPCTA ダンプ・ドメイン出口が XPCTA の前に呼び出されることがあります (34 ページの『[出口 XDUREQ](#)』を参照)。また、再開アドレスが返されると、レジスター 0 から 15 は異常終了時の値に復元されます。プログラム状況ワード (PSW) は、再開アドレスに分岐するために使用されます。

一部の状況では、CICS は、通常は UERCMEA 戻りコードから取得する再開アドレスを無視するためのフラグを設定します。

- 以下の状況では、CICS は、出口が提供する再開アドレスを無視するための PCUE\_NO\_RESUME フラグを設定します。
  - XPLINK オプションを使用してコンパイルされた C または C++ プログラムに対して XPCTA が呼び出される場合。
  - アプリケーションのタスク制御ブロック (TCB) が既に使用可能ではない場合。
  - トランザクション異常終了が AKKD および AKKE 以外の AKxx 異常終了 (KILL 要求に関連するもの) である場合。
- AMODE(64) プログラムの場合、異常終了時に 64 ビット・レジスターが使用不可であれば、CICS は出口が提供する再開アドレスを無視するための PCUE\_NO\_RESUME\_AMODE64 フラグを設定します。

### 呼び出される状況

異常終了が発生した後、環境が変更される前。

### 出口固有のパラメーター

#### UEPPCDS

プログラム関連および端末関連の情報が格納され、DSECT DFHPCUE を使用してマッピングできるストレージ域のアドレス。XPCTA が呼び出されるときには、以下の DFHPCUE フィールドが意味を持ちます。

#### PCUE\_CONTROL\_BITS

1 バイト・フラグ・フィールド。PCUECBTE の設定は、トランザクションが端末にリンクされていることを示します。

PCUE\_CONTROL\_BITS 内のフラグ PCUE\_NO\_RESUME および PCUE\_NO\_RESUME\_AMODE64 は、再開アドレスがサポートされていないことを示します。設定されている場合、XPCTA からの UERCMEA の戻りコードはすべて無視されます。CICS は、XPLINK オプションを使用してコンパイルされた C および C++ プログラムに対して XPCTA を呼び出す前、アプリケーションの TCB が既に使用不可になっている場合、および AKKD または AKKE 以外の AKxx 異常終了の場合に、PCUE\_NO\_RESUME を設定します。CICS は、再開アドレスがサポートされていない場合に、AMODE(64) プログラムに対して PCUE\_NO\_RESUME\_AMODE64 を設定します。

#### PCUE\_TASK\_NUMBER

タスク番号を含む 3 文字パック 10 進数フィールド。

#### PCUE\_TRANSACTION\_ID

トランザクション ID が含まれた 4 文字のフィールド。

#### PCUE\_TERMINAL\_ID

端末 ID (該当する場合) を含む 4 文字のフィールド。

#### PCUE\_PROGRAM\_NAME

障害が発生したプログラムの名前が含まれている 8 文字のフィールド。

#### PCUE\_LOGICAL\_LEVEL

プログラムの論理レベルを含むフルワード。

#### PCUE\_BRANCH\_ADDRESS

フルワード。このフィールドを使用して、再開アドレスを提供できます。再開タスクが AMODE(31) を実行することを指定するには、最上位ビットを設定します。再開タスクが AMODE(64) を実行することを指定するには、最下位ビットを指定します。

#### PCUE\_BRANCH\_EXECKEY

ストレージ保護がアクティブになっている場合は、この 1 バイト・フィールドを使用して、再開タスクの実行キーを指定することができます。可能な値は次のとおりです。



## **PCUE\_BRANCH\_USER**

ユーザー・キー

## **PCUE\_BRANCH\_CICS**

CICS キー

ストレージ保護がアクティブになっていて、値を指定しなかった場合は、再開タスクはユーザー・キーで実行されます。

ストレージ保護がアクティブでない場合は、再開タスクは CICS キーで実行されます。

## **UEPTACB**

異常終了のトランザクション異常終了制御ブロック (TACB) のアドレス。プログラム・チェックが原因で異常終了が発生した場合、TACB 内の情報には以下が含まれています。

- プログラム状況ワード (PSW)。
- 異常終了時のレジスター。
- 異常終了時に現行であったサブスペース、アクセス・レジスター、およびベクトル・レジスターについての詳細。
- 中断イベント・アドレス・レジスター (BEAR)。
- 変換例外アドレス (TEA)。

DFHTACB TYPE=DSECT マクロを使用して TACB をマップすることができます。

## 戻りコード

### **UERCNORM**

処理が続行される。

### **UERCPURG**

タスクは XPI 呼び出し中にページされます。

### **UERCMEA**

再開アドレスが提供されています。

## **XPI 呼び出し**

すべて使用できます。

## **XPCTA 出口プログラムの例**

DFH\$PCTA は、XPCTA 出口点のためのグローバル・ユーザー出口プログラムの例です。この例では、異常終了の原因がストレージ保護例外条件であったかどうかをテストします。DFH\$PCTA について詳しくは、[トランザクション異常終了サンプル出口プログラム: DFH\\$PCTA](#) を参照してください。

## 出口 **XPCABND**

出口 XPCABND は、トランザクションが異常終了した後で、トランザクション・ダンプが呼び出される前に呼び出されます。この出口を使用してダンプを抑止できます。

### 呼び出される状況

トランザクションが異常終了した後で、トランザクション・ダンプ呼び出しが行われる前。

### 出口固有のパラメーター

#### **UEPPCDS**

プログラム関連情報および端末関連情報が含まれたストレージ域のアドレス。このストレージ域は、DSECT DFHPCUE によってマップされます。

XPCABND が呼び出されるときには、以下の DFHPCUE フィールドが意味を持ちます。

#### **PCUE\_CONTROL\_BITS**

1 バイトのフラグ・フィールド。PCUECBTE の設定は、トランザクションが端末にリンクされていることを示します。

#### **PCUE\_TASK\_NUMBER**

タスク番号が含まれた 3 文字のパック 10 進数フィールド。

**PCUE\_TRANSACTION\_ID**

トランザクション ID が含まれた 4 文字のフィールド。

**PCUE\_TERMINAL\_ID**

端末 ID (該当する場合) が含まれた 4 文字のフィールド。

**PCUE\_PROGRAM\_NAME**

異常終了中のプログラムの名前が含まれた 8 文字のフィールド。

**PCUE\_LOGICAL\_LEVEL**

プログラムの論理レベルを含むフルワード。

**UEPTACB**

異常終了のトランザクション異常終了制御ブロック (TACB) のアドレス。プログラム・チェックが原因で異常終了が発生した場合、TACB 内の情報には以下が含まれています。

- プログラム状況ワード (PSW)。
- 異常終了時のレジスター。
- 異常終了時に現行であったサブスペース、アクセス・レジスター、およびベクトル・レジスターについての詳細。
- 中断イベント・アドレス・レジスター (BEAR)。
- 変換例外アドレス (TEA)。

DFHTACB TYPE=DSECT マクロを使用して TACB をマップすることができます。

**戻りコード****UERCNORM**

処理を続行し、ダンプ呼び出しを行います。

**UERCBYD**

ダンプ呼び出しを抑止します。

**UERCPUFG**

タスクは XPI 呼び出し中にページされます。

**XPI 呼び出し**

すべて使用できます。

## リソース・マネージャー・インターフェース・プログラム出口 (XRMIIN、XRMIOUT)

これらの出口は RMI API 要求が処理されるときに呼び出されます。

### 出口の XRMIIN

出口の XRMIIN は、アプリケーション・プログラムが RMI API 要求を出してタスク関連ユーザー出口プログラムが呼び出される前に、呼び出されます。

#### 出口固有のパラメーター

**UEPTRUEN**

タスク関連ユーザー出口プログラムの名前のアドレス。

**UEPTRUEP**

タスク関連ユーザー出口プログラムに受け渡されるパラメーター・リストのアドレス。

**UEP\_RM\_PBTOK**

z/OS ワークロード・マネージャー (WLM) の Performance Block Token を含む 4 バイトのフィールドのアドレス。出口プログラムは、このトークンを使用して、WLM Performance Block の情報 (サービス・クラス・トークン SERVCLS など) にアクセスすることができます。このアクセスを行うためには、出口プログラムは、Performance Block Token を MONTKN の入力パラメーターとして受け渡す WLM EXTRACT マクロの IWMMESTR を使用する必要があります。IWMMESTR マクロについて詳しくは、[z/OS MVS プログラミング: ワークロード管理サービス](#)を参照してください。



出口プログラムで、Performance Block の変更を試みないでください。そのような変更を試みた場合の結果は、予測不能です。

#### **UEPRECUR**

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

**注:** タスク関連ユーザー出口プログラムのパラメーター・リストは、グローバル・ユーザー出口プログラムの DFHUEPAR パラメーター・リストと共通のフィールド名を共用する DFHUEPAR DSECT によってマップされます。両方の DSECT 定義を出口プログラムに組み込むには、次のようにコーディングする必要があります。

```
DFHUEEXIT TYPE=EP,ID=XRMIIN
DFHUEEXIT TYPE,TYPE=RM
```

各ステートメントは、この順序でコーディングしてください。

2 つの DFHUEPAR パラメーター・リスト (グローバル・ユーザー出口ルーチンのものとタスク関連ユーザー出口のもの) は、別々のストレージ域を占有します。タスク関連ユーザー出口のパラメーター・リストは、情報提供専用であるため、決して変更しないでください。

#### **戻りコード**

##### **UERCNORM**

処理が続行される。

##### **UERCPUrg**

タスクは XPI 呼び出し中にページされます。

#### **XPI 呼び出し**

すべて使用できます。

#### **API および SPI コマンド**

EXEC CICS SHUTDOWN および EXEC CICS XCTL を除くすべてを使用できます。ただし、CALLDLI、EXEC DLI、または EXEC SQL コマンドを使用しないでください。

### **出口 XRMIOU**

出口 XRMIOU は、タスク関連ユーザー出口プログラムが RMI API 要求の処理から戻った後で呼び出されます。

#### **出口固有のパラメーター**

##### **UEPTRUEN**

タスク関連ユーザー出口プログラムの名前のアドレス。

##### **UEPTRUEP**

タスク関連ユーザー出口プログラムに受け渡されるパラメーター・リストのアドレス。

このパラメーター・リストの UEPHMSA パラメーターには、呼び出し元のレジスター保管域 (RSA) が含まれています。EXEC CICSplex SM 呼び出しの場合、この RSA のレジスター 1 には、CPSM コマンドのパラメーター・リストのアドレスが含まれ、CICSplex SM コマンドおよび指定されたリソースの識別に使用できます。詳しくは、[CICSplex SM API コマンド引数リスト](#)を参照してください。

##### **UEP\_RM\_PBTOK**

z/OS ワークロード・マネージャー (WLM) の Performance Block Token を含む 4 バイトのフィールドのアドレス。出口プログラムは、このトークンを使用して、WLM Performance Block の情報 (サービス・クラス・トークン SERVCLS など) にアクセスすることができます。このアクセスを行うためには、出口プログラムは、Performance Block Token を MONTKN の入力パラメーターとして受け渡す WLM EXTRACT マクロの IWMMEXTR を使用する必要があります。IWMMEXTR マクロについて詳しくは、[z/OS MVS プログラミング: ワークロード管理サービス](#)を参照してください。

出口プログラムで、Performance Block の変更を試みないでください。そのような変更を試みた場合の結果は、予測不能です。

## UEPRECUR

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

**注:** タスク関連ユーザー出口プログラムのパラメーター・リストは、グローバル・ユーザー出口プログラムの DFHUEPAR パラメーター・リストと共通のフィールド名を共用する DFHUEPAR DSECT によってマップされます。両方の DSECT 定義を出口プログラムに組み込むには、次のようにコーディングする必要があります。

```
DFHUEXIT TYPE=EP,ID=XRMIOU  
DFHUEXIT TYPE,TYPE=RM
```

各ステートメントは、この順序でコーディングしてください。

グローバル・ユーザー出口の DFHUEPAR パラメーター・リストおよびタスク関連ユーザー出口の DFHUEPAR パラメーター・リストは、ストレージ内でそれぞれ別個の領域を占めます。タスク関連ユーザー出口のパラメーター・リストは、情報提供専用であるため、決して変更しないでください。

## 戻りコード

### UERCNORM

処理が続行される。

### UERCPURG

タスクは XPI 呼び出し中にパージされます。

## XPI 呼び出し

すべてを使用できます。

## API コマンドおよび SPI コマンド

EXEC CICS SHUTDOWN および EXEC CICS XCTL を除くすべてを使用できます。ただし、CALLDLI、EXEC DLI、または EXEC SQL コマンドを使用しないでください。

**注:** 出口プログラムで RMI を使用する他の外部リソース・マネージャーに対する呼び出しを実行することはお勧めできません。これによって再帰が発生し、ループになる可能性があるためです。ループに入らないようにするのは出口プログラムの責任です。この可能性に対するガードとして、出口プログラムで再帰カウンター UEPRECUR を使用することができます。

## リソース管理のインストールおよび破棄出口 XRSINDI

XRSINDI グローバル・ユーザー出口は、使用可能に設定されていると、CICS がリソース定義のインストールまたは破棄を正常に実行した直後に呼び出されます。

出口を呼び出すインストール・アクティビティーおよび破棄アクティビティーは以下のとおりです。

- CICS の初期スタートまたはコールド・スタート時のグループ・リストのインストール機能
- **CEDA INSTALL** コマンド
- CICSplex SM BAS **INSTALL** コマンド
- 以下のようなすべての自動インストール操作
  - 端末、接続、プログラム、マップ・セット、区分セット、またはジャーナルの自動インストール
  - AILDELAY システム初期設定パラメーターおよび TYPETERM リソース定義の SIGNOFF 属性によって制御された未使用端末の自動破棄
- MVS ログ・ストリームへの接続、およびこのストリームからの切断
- **CEMT DISCARD** コマンドおよび **EXEC CICS DISCARD** コマンド
- **EXEC CICS CREATE** コマンド
- フロントエンド・プログラミング・インターフェース (FEPI) のインストールおよび破棄操作: **EXEC CICS FEPI INSTALL** コマンドおよび **EXEC CICS FEPI DISCARD** コマンド

パラメーター・リストは、インストールまたは破棄された複数のリソースの名前をフィールド UEPIDNAM に入れて渡すように設計されています。グローバル・ユーザー出口プログラムを設計する際に、渡される

リソース名が常に1つだと想定しないでください。UEPIDNUMによって参照される値に基づいて、ループ内のリソースを分析してください。

モードグループの名前の前に、対応する接続名が付きます。2つの名前の間には分離文字がありません。最初の4文字は接続名を形成し、その後にモードグループを表す8文字が続きます。連結された名前の部分は固定長です。接続名が4文字未満で定義されている場合は、連結された名前に空白が埋め込まれます。同様に、フロントエンド・プログラミング・インターフェース (FEPI) 接続の接続名は、FEPI ノード名と FEPI ターゲット名を連結したものです。ノード名とターゲット名はそれぞれ8文字の長さ (固定長) で、分離文字はありません。

CICS 初期スタートまたはコールド・スタート時にインストールされたグループ・リスト内の個別リソースごとに、この出口が1回呼び出されます。初期スタートまたはコールド・スタート時のパフォーマンスが心配な場合は、グループ・リストのインストールが終わるまで、出口を使用可能にしないでください。出口を使用可能にする前にインストールストリームのリソースに関する情報を取得するには、**EXEC CICS INQUIRE resource\_name** ブラウズ機能を使用して、インストール済みのリソースのテーブルをスキャンするプログラムを作成することができます。

## 出口 XRSINDI

XRSINDI グローバル・ユーザー出口は、CICS がリソース定義をインストールまたは破棄するときに呼び出されます。

### 出口 XRSINDI のパラメーターおよび戻りコード

XRSINDI 出口点で使用可能にされたプログラムで異常終了すると、CICS がシャットダウンする可能性があります。なぜなら、一部のリソースでは出口が同期点で実行されるためです。出口により、これらのリソースの同期点でコード UERCPURG が返されると、異常終了コード AUPE が生成され、CICS はシャットダウンします。

パラメーター、戻りコード、および XPI 情報は以下のとおりです。

#### 出口固有のパラメーター

##### UEPTRANID

4 バイトのトランザクション ID のアドレス。

##### UEPUSER

8 バイトのユーザー ID のアドレス。

##### UEPTERM

4 バイトの端末 ID のアドレス。

##### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス。

##### UEPIDREQ

1 バイトのインストールまたは破棄 ID のアドレス。値は以下のとおりです。

##### UEIDINS

この要求はインストール操作のためのものです。あるいは、ログ・ストリームの場合、これはログ・ストリームへの接続になります。

##### UEIDDIS

この要求は破棄操作のためのものです。あるいは、ログ・ストリームの場合、これはログ・ストリームからの切断になります。

##### UEPIDTYP

1 バイトのリソース・タイプのアドレス。値は以下のとおりです。

##### UEIDATOM

ATOMSERVICE リソース

##### UEIDAITM

自動インストール端末モデル

##### UEIDBNL

BUNDLE リソース

**UEIDCONN**

接続

**UEIDDB2C**

CICS と Db2® 間の接続の DB2CONN リソース定義

**UEIDDB2E**

DB2ENTRY リソース定義

**UEIDDB2T**

DB2TRAN リソース定義

**UEIDDOCT**

DOCTEMPLATE

**UEIDDMPC**

DUMPCODE リソース定義

**UEIDEARB**

CICS バンドルの一部である EAR ファイル。

**UEIDEBAB**

CICS バンドルの一部である EBA ファイル。

**UEIDEPAD**

EPADAPTER リソース

**UEIDEPAS**

EPADAPTERSET リソース

**UEIDEVCS**

イベント・キャプチャー・リソース

**UEIDEVNT**

EVENTBINDING リソース

**UEIDFECO**

FEPI 接続

**UEIDFENO**

FEPI ノード

**UEIDFEPO**

FEPI プール

**UEIDFEPS**

FEPI プロパティ・セット

**UEIDFETA**

FEPI ターゲット

**UEIDFILE**

ファイル。

**UEIDIPCO**

IPCONN リソース

**UEIDJNMD**

ジャーナル・モデル

**UEIDJNNM**

ジャーナル名

**UEIDJSRV**

JVM サーバー・リソース

**UEIDLTRY**

LIBRARY リソース

**UEIDMAP**

マップ・セット

**UEIDMODE**  
モードグループ

**UEIDMPPP**  
ポリシー・リソース。

**UEIDMQCN**  
CICS と IBM MQ の間の接続の MQCONN リソース定義

**UEIDMQIN**  
MQINI リソース

**UEIDMQMN**  
MQMONITOR リソース

**UEIDNAPP**  
NODEJSAPP リソース。

**UEIDNQRN**  
ENQMODEL

**UEIDOSGB**  
OSGi バンドル

**UEIDPART**  
パートナー

**UEIDPIPE**  
パイプライン (PIPELINE)

**UEIDPKST**  
Db2 PACKAGESET リソース

**UEIDPROF**  
プロファイル

**UEIDPROG**  
プログラム

**UEIDPRTY**  
BTS プロセス・タイプ

**UEIDPSET**  
区分セット

**UEIDSESS**  
セッション

**UEIDSTRM**  
MVS ログ・ストリーム

**UEIDTCLS**  
トランザクション・クラス

**UEIDTCPS**  
TCP/IP サービス

**UEIDTDQU**  
一時データ・キュー

**UEIDTERM**  
端末

**UEIDTRAN**  
トランザクション

**UEIDTSMD**  
一時記憶域キュー・モデル。

**UEIDURIM**  
URIMAP リソース

**UEIDWARB**  
CICS バンドルの一部である WAR ファイル。

**UEIDWEBS**

Web サービス (WEBSERVICE)

**UEIDXMLT**

XMLTRANSFORM リソース

**UEPIDLEN**

公用リソースの場合、このパラメーターは、個々のリソース名のフルワード・バイナリー値としての長さのアドレスです。

OSGi バンドルの場合、このパラメーターは、CICS の **UEPIDNAM** パラメーターに入る、OSGi バンドルを一意的に識別する情報の長さのアドレスです。最大長は 526 バイトです。

**UEPIDNUM**

この呼び出しで報告される、フルワード・バイナリー値としてのリソース数のアドレスです。

**UEPIDNAM**

この呼び出しで報告された個別のリソース名が入っている可変長リストのアドレス。

OSGi バンドルの場合は、CICS の OSGi バンドルを一意的に識別する情報がこのパラメーターに入ります。この情報は、以下の順序でリストされます。

1. JVM サーバー名が入る 8 バイト。
2. OSGi バンドルのシンボル名の長さが入るフルワード。
3. OSGi バンドルのバージョンの長さが入るフルワード。
4. OSGi バンドルのシンボル名とバージョンの連結 (文字ストリング)。

**UEPIDREC**

リソースをウォーム・リスタートでリカバリーするか、あるいは緊急リスタートでリカバリーするかを示す 1 バイトの ID のアドレス。値は以下のとおりです。

**UEIDKEEP**

リソースはウォーム・リスタートあるいは緊急リスタートでリカバリー可能です。

**UEIDLOSE**

リソースはリカバリー可能ではありません。

注: 出口は、CICS 再始動中には呼び出されません。

**UEPDEFTM**

個別のリソースの定義時刻を、8 文字の STCK 値として収容する可変長リスト (UEPIDNAM のリストに対応) のアドレス。

注: パラメーター UEPDEFTM、UEPCHUSR、UEPCHAGT、UEPCHREL、UEPCHTIM、UEPDEFSRC、UEPINUSR、UEPINTIM、および UEPINAGT は、以下のリソースの場合に有効です。  
ATOMSERVICE、BUNDLE、CONNECTION、DB2CONN、DB2ENTRY、DB2TRAN、DOCTEMPLATE、DUMPCODE、ENQMODEL、EPADAPTER、EPADAPTERSET、EVENTBINDING、FILE、IPCONN、JOURNALMODEL、JVMSEVER、LIBRARY、MQCONN、MQINI、OSGIBUNDLE、PIPELINE、PROFILE、PROCESSTYPE、PROGRAM、TCPIPSERVICE、TDQUEUE、TRANCLASS、TRANSACTION、TSMODEL、URIMAP、WEBSERVICE、XMLTRANSFORM。その他のすべてのリソースでは、パラメーター値はゼロです。

**UEPCHUSR**

個別のリソースを最後に変更したエージェントを実行した、8 文字のユーザー ID を収容する可変長リスト (UEPIDNAM のリストに対応) のアドレス。

**UEPCHAGT**

個別のリソースを最後に変更したエージェントを表す 2 バイトの ID の可変長リスト (UEPIDNAM のリストに対応) のアドレス。有効な値は、以下のとおりです。

**UEPUNKAGT**

リソースは、不明エージェントによって変更されました。

**UEPCSDAPI**

CSD API または CEDA を使用して、リソースが変更されました。

**UEPCSDBAT**

CSD バッチ・プログラム DFHCSDUP を使用してリソースが変更されました。

**UEPDRPAPI**

CICSplex SM BAS API を使用して、リソースが変更されました。

**UEPAUTOIN**

自動インストールを使用してリソースが変更されました。

**UEPSYSTEM**

実行中の CICS 領域によって、リソースが変更されました。

**UEPDYNAMC**

リソースが動的に変更されました。

**UEPTABLE**

テーブルを使用してリソースが変更されました。

**UEPCHREL**

個別のリソースが最後に変更された際に実行中であった、4 文字の CICS リリース・レベルを収容する可変長リスト (UEPIDNAM のリストに対応) のアドレス。

**UEPCHTIM**

個別のリソースの CSD レコード・タイム・スタンプ変更を、8 文字の STCK 値として収容する可変長リスト (UEPIDNAM のリストに対応) のアドレス。

**UEPDEFSRC**

個別のリソースに対応する、8 文字の CSD グループ名またはソースを収容する可変長リスト (UEPIDNAM のリストに対応) のアドレス。

**UEPINUSR**

個別のリソースをインストールした 8 文字のユーザー ID を収容する可変長リスト (UEPIDNAM のリストに対応) のアドレス。

**UEPINTIM**

個別のリソースのインストールにおいてドメインが呼び出された時刻を、8 文字の STCK 値として収容する可変長リスト (UEPIDNAM のリストに対応) のアドレス。

**UEPINAGT**

個別のリソースをインストールしたエージェントを表す、2 バイトの ID の可変長リスト (UEPIDNAM のリストに対応) のアドレス。有効な値は、以下のとおりです。

**UEPCSDAPI**

CSD API または CEDA を使用して、リソースがインストールされました。

**UEPCRESPI**

EXEC CICS CREATE SPI コマンドを使用して、リソースがインストールされました。

**UEPGRPLST**

GRPLIST インストールを使用して始動時にリソースがインストールされました。

**UEPAUTOIN**

リソースは自動インストールされました。

**UEPSYSTEM**

CICS システムを実行してリソースがインストールされました。

**UEPDYNAMC**

リソースが動的にインストールされました。

**UEPBUNDLE**

リソースは、バンドルのデプロイメントによってインストールされました。

**UEPTABLE**

テーブルを使用してリソースがインストールされました。

**UEPAPPTK**

このリソースが属するアプリケーション・インスタンスを表す 8 文字のトークンを含む可変長リストのアドレス。公用リソースの場合、このアドレスはゼロになります。

## UEPAPCTXT

プラットフォームにデプロイされたアプリケーションの専用リソースの場合、このパラメーターには、リソースに関するアプリケーション・コンテキスト情報を含む可変長リスト (UEPIDNAM のリストに対応) のアドレスが入ります。この情報は、以下の順序でリストされます。

1. プラットフォーム名。スペースを埋め込んで 64 文字。
2. アプリケーション名。スペースを埋め込んで 64 文字。
3. アプリケーションのメジャー・バージョン番号。フルワード・バイナリー値。
4. アプリケーションのマイナー・バージョン番号。フルワード・バイナリー値。
5. アプリケーションのマイクロ・バージョン番号。フルワード・バイナリー値。

CICS は、この情報にマップする **DFHUEACD** という名前の DSECT を提供します。**DFHUEACD** についての詳細は、『Data Areas』の『UEACD - User exit application context』を参照してください。

## UEPPLATTK

このリソースが属するプラットフォーム・インスタンスを表す 8 文字のトークンを含む可変長リストのアドレス。公用リソースの場合、このアドレスはゼロになります。

## 戻りコード

### UERCNORM

処理を続行します。このコードはデフォルトです。

### UERCPURG

タスクは XPI 呼び出し中にパージされます。

## XPI 呼び出し

すべての XPI 呼び出しを使用できます。

## サインオン出口およびサインオフ出口 XSNON、XSNOFF、XSNEX

出口 XSNON は端末ユーザーがサインオンするときに呼び出され、出口 XSNOFF は端末ユーザーがサインオフした後で (サインオンまたはサインオフが成功したかどうかに関係なく) 呼び出されます。XSNON および XSNOFF は、セキュリティ決定は何も行いません。これらの出口は、CICS システムでのユーザーのログオンおよびログオフを追跡するための手段でしかありません。

この出口を呼び出すアクティビティは、以下のとおりです。

- 端末に対する EXEC CICS SIGNON コマンドの呼び出し (例えば、端末ユーザーが、CICS 提供の CESN、または同等のユーザー作成のサインオン・トランザクションを入力した場合)
- 代理端末 (つまり、CRTE ルーティング・トランザクションによって、または動的トランザクション・ルーティングによって接続された端末) に対する EXEC CICS SIGNON コマンドの呼び出し
- 端末に対する EXEC CICS SIGNOFF コマンドの呼び出し
- CRTE ルーティング・セッションを終了するために「CANCEL」コマンドが入力されたとき
- タイムアウトによるサインオフ

XSNEX は、IBM 提供のグローバル・ユーザー出口プログラム DFH\$SNEX でのみ使用するための特殊目的のグローバル・ユーザー・ポイントです。

## 出口 XSNON

### 呼び出される状況

ユーザーがサインオンしたとき。

### 出口固有のパラメーター

#### UEPUSRID

端末ユーザー ID のアドレス。

#### UEPUSRLN

端末ユーザー ID の長さのアドレス。



**UEPGRPID**

グループ ID のアドレス。サインオンが正常に実行された場合、このグループ ID は、ユーザーがこのサインオン・セッションに関連付けられる ID です。サインオンが失敗した場合は、ユーザーがサインオンしようとしたときに指定した ID です。

**UEPGRPLN**

グループ ID の長さのアドレス。

**UEPNETN**

端末のネット名のアドレス。

**UEPTRMID**

端末 ID のアドレス。

**UEPTCTUA**

TCT ユーザー域のアドレス。

**UEPTCTUL**

TCT ユーザー域の長さのアドレス。

**UEPTRMTY**

端末タイプ・バイトのアドレス。

**UEPSNFLG**

以下のフラグが含まれた 2 バイト・フィールドのアドレス。

表 1. XSNON の UEPSNFLG フィールドで設定されるフラグ		
フラグ	等価の値	意味
UEPSNOK	0	サインオンが正常に終了しました。
UEPSNFL	1	サインオンが失敗しました。
UEPSNPSS	2	持続セッションのサインオンが正常に終了しました。
UEPSNPSF	3	持続セッションのサインオンが失敗しました。

**UEPSGTYP**

サインオン・タイプ・バイトのアドレス。

このパラメーターには 2 つの等価の値があります。

UEPSGUID	SIGNON USERID
UEPSGKER	SIGNON KERBEROS

**戻りコード****UERCNORM**

処理が続行される。

**UERCPURG**

タスクは XPI 呼び出し中にページされます。

**XPI 呼び出し**

すべて使用できます。

**出口 XSNOFF****呼び出される状況**

ユーザーがサインオフしたとき。

**出口固有のパラメーター****UEPUSRID**

端末ユーザー ID のアドレス。

**UEPUSRLN**

端末ユーザー ID の長さのアドレス。

**UEPGRPID**

グループ ID のアドレス。

**UEPGRPLN**

グループ ID の長さのアドレス。

**UEPNETN**

端末のネット名のアドレス。

**UEPTRMID**

端末 ID のアドレス。

**UEPTCTUA**

TCT ユーザー域のアドレス。

**UEPTCTUL**

TCT ユーザー域の長さのアドレス。

**UEPTRMTY**

端末タイプ・バイトのアドレス。

**UEPSNFLG**

以下のフラグが含まれた 2 バイト・フィールドのアドレス。

**UEPSNOK**

サインオフが正常に終了しました

**UEPSNFL**

サインオフが失敗しました

**UEPSNNML**

正常なサインオフ

**UEPSNTIM**

サインオフがタイムアウトになりました

**戻りコード****UERCNORM**

処理が続行される。

**UERCPUrg**

タスクは XPI 呼び出し中にページされます。

**XPI 呼び出し**

すべて使用できます。

**出口 XSSEX**

XSSEX の目的は、それをサポートするサンプル・プログラムと共に、アップグレードのための短期支援を提供することです。この出口は、CICS TS 2.1 でアプリケーション・プログラムを現行の動作と関係できるようにする前に、CICS が **EXEC CICS SIGNON** および **SIGNOFF** を処理する方法に依存するアプリケーション・プログラムを変更するための時間をユーザーに提供するために設計されています。

**注: XSSEX はアップグレード目的のみで使用します。** 古いサインオン/サインオフ動作に対するアプリケーションの依存関係をすべて削除してください。

このグローバル・ユーザー出口には出口固有のパラメーターはありません。この出口は、アプリケーション・プログラムが **EXEC CICS SIGNON** コマンドまたは **EXEC CICS SIGNOFF** コマンドを発行するたびに呼び出されます。この出口点のために独自のグローバル・ユーザー出口プログラムを作成する必要はありません。IBM が DFH\$SEX を用意しています。このプログラムの唯一の目的は、CICS で **EXEC CICS SIGNON** および **SIGNOFF** コマンドを CICS TS 1.3 以前と同じ方法で処理することです。

提供されるプログラムは以下のとおりです。

**DFH\$SEX**

このユーザー出口プログラムは SDFHSAMP で提供されます。このプログラムが実行する機能は、戻りコード UERCPREV を設定することだけです。この戻りコードによって、セキュリティー・ドメインで

CICS TS 1.3 以前と同じ CICS 動作が復元されます。DFH\$SNPI を使用して、このユーザー出口プログラムを有効にすることができます。

### DFH\$SNPI

この初期設定後処理プログラムは SDFHSAMP で提供されます。これは、EXEC CICS ENABLE PROGRAM('DFH\$SNEX') EXIT('XSNE') コマンドを発行して、CICS 初期設定の最終段階で IBM 提供のユーザー出口プログラム DFH\$SNEX を有効にします。

このプログラムを使用するには、PLTPI テーブルの最初のセクション (つまり、DFHDELIM ステートメントの前に、エントリーを 1 つ追加します。例:

```
          DFHPLT TYPE=INITIAL,SUFFIX=SN
DFHPLT TYPE=ENTRY,PROGRAM=DFH$SNPI
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
DFHPLT TYPE=FINAL
END
```

## 統計ドメイン出口 XSTOUT

XSTOUT は、呼び出されると、1 つ以上の統計レコードが含まれたバッファのアドレスを受け取ります。このバッファは、接続やモード名など、さまざまなリソース・タイプのレコードを格納できます。また、このバッファは、固有情報とグローバル情報 (例えば、個別プログラムのローダー統計とすべてのプログラムのローダー統計) を格納することもできます。

出口プログラムは、バッファ内のレコードのタイプをその STID 値によって識別できます。(STID 値については、[CICS 統計データ・セクション](#)で説明されています。)

XSTOUT を使用して、統計データ・バッファの内容が SMF に書き込まれるのを防止することができます。バッファ内の個々のレコードを選択的に抑止することはできないことに注意してください。出口プログラムで、出口固有のパラメーターの値を変更しないでください。

CICS の初期設定の初期段階では、一部の統計レコードは生成されない場合があります、それらのレコードは XSTOUT に渡されません。グローバル・ユーザー出口を使用可能にできる最も早い時点は、PLT 処理中です。その前は、出口を呼び出すことはできません。

## 出口 XSTOUT

### 呼び出される状況

統計レコードが SMF に書き込まれる前。

### 出口固有のパラメーター

フィールド UEPPROG、UEPTERM、UEPTRANID、および UEPUSER は、要求された統計についてのみ意味を持ちます (CICS Explorer® 「領域」操作ビュー、**CEMT PERFORM STATISTICS RECORD** コマンド、または **EXEC CICS PERFORM STATISTICS RECORD** コマンドを使用する場合)。

#### UEPPROG

8 バイトのアプリケーション・プログラム名のアドレス。

#### UEPSCLD

収集日 (MMDDYYYY) が含まれた 8 バイト文字フィールドのアドレス。

#### UEPSDATE

収集日 (MMDDYY) が含まれた 6 バイト文字フィールドのアドレス。

#### UEPSIVAL

間隔時間 (HHMMSS) が含まれた 6 バイト文字フィールドのアドレス。このフィールドは、間隔統計に関してのみ意味を持ちます。

#### UEPSIVN

4 バイトの間隔番号のアドレス。このフィールドは、間隔統計に関してのみ意味を持ちます。

#### UEPSRLEN

統計レコードの 4 バイトの 16 進数の長さのアドレス。

#### UEPSTATS

1 つ以上の統計レコードが含まれたバッファのアドレス。非送信請求統計の場合、バッファには常に 1 つのレコードが含まれています。その他のタイプの統計の場合、バッファには複数のレ

コードが含まれていることがあります。バッファの長さは、UEPSRLLEN パラメーターによってアドレス指定されます。

#### **UEPSTIME**

収集時刻 (HHMMSS) が含まれた 6 バイト文字フィールドのアドレス。

#### **UEPSTYPE**

3 バイトの文字フィールド統計タイプのアドレス。タイプの値は、以下のとおりです。

##### **INT**

間隔統計。

##### **EOD**

1 日の終わり統計。

##### **REQ**

要求された統計。

##### **RRT**

要求されたリセット統計。

##### **USS**

非送信請求統計。

#### **UEPTERM**

4 バイトの端末 ID のアドレス。

#### **UEPTRANID**

4 バイトのトランザクション ID のアドレス。

#### **UEPUSER**

8 バイトのユーザー ID のアドレス。

### 戻りコード

#### **UERCBYBYP**

SMF への統計データ・バッファの出力を抑止します。

#### **UERCNORM**

処理を続行します。

### **XPI 呼び出し**

WAIT\_MVS を使用できます。ただし、CEMT コマンドまたは SPI コマンドを使用して待機をパージすることはできません。その他の呼び出しを使用しないでください。

## システム・リカバリー・プログラム出口 XSRAB

システム・リカバリー・プログラム (DFHSRP) がシステム・リカバリー・テーブル (SRT) 内でオペレーティング・システム異常終了コードに関する一致を検出すると、出口 XSRAB が呼び出されます。

### 呼び出される状況

出口 XSRAB は、システム・リカバリー・プログラム (DFHSRP) が SRT 内でオペレーティング・システム異常終了コードに関する一致を検出したときに呼び出されます。SRT 内のエントリーの定義に関する情報は、[システム・リカバリー・テーブル \(SRT\)](#) を参照してください。

SRT テーブルが処理され、出口が実行されるのは、CICS の必須 TCB、すなわち QR、RO、CO、SZ、RP、FO のいずれかの下で MVS 異常終了が発生した場合に限られます。L8、SL、SO、または S8 などの重要でない TCB タイプの場合、出口は実行されません。

### 出口固有のパラメーター

#### **UEPERROR**

以下のフィールドを含む、エラー・データ構造 SRP\_ERROR\_DATA のアドレス。

##### **SRP\_ERROR\_TYPE**

4 文字のエラー・タイプ。これは常に、ASRB となります。

##### **SRP\_SYS\_ABCODE**

バイナリー形式のシステム異常終了コード XXX を含む 2 バイト (D37 など)。

**SRP\_USER\_ABCODE**

バイナリー形式のユーザー異常終了コード NNNN を含む 2 バイト (0999 など)。

**SRP\_ERROR\_TRANID**

異常終了したトランザクションの ID を含む 4 文字のフィールド。

**SRP\_ERROR\_STACK\_NAME**

異常終了時のトランザクションに関する現行のカーネル・スタック・エントリーの名前を含む 8 文字のフィールド。

**SRP\_ERROR\_PPT\_NAME**

トランザクションに関する現行のプログラムの名前を含む 8 文字のフィールド。このフィールドに値が含まれるのは、フラグ SRP\_PPT\_ENTRY が設定されている場合のみです。

**SRP\_ERROR\_OFFSET**

以下のような、異常終了したプログラムへのオフセットを含むフルワード。

- ・フラグ SRP\_PPT\_ENTRY が設定されている場合は、SRP\_ERROR\_PPT\_NAME 内のオフセットを付与します。
- ・設定されていない場合は、SRP\_ERROR\_STACK\_NAME 内のオフセットを付与します。

このフィールドに値が含まれるのは、フラグ SRP\_VALID\_OFFSET が設定されている場合のみです。

**SRP\_ERROR\_FLAGS**

以下のフラグを含む 1 バイト：

**SRP\_CICS\_CODE**

CICS コードの実行中に異常終了が発生しました。

**SRP\_USER\_CODE**

ユーザー・アプリケーション・コードの実行中に異常終了が発生しました。

**SRP\_PPT\_ENTRY**

SRP\_ERROR\_PPT\_NAME の実行中に異常終了が発生しました。このフラグが設定されていない場合は、SRP\_ERROR\_STACK\_NAME の実行中に異常終了が発生しました。

**SRP\_VALID\_OFFSET**

意味のあるオフセットを判別できました。

**SRP\_VALID\_REASON**

MVS が異常終了に関する理由コードを提供しました。

**SRP\_NOT\_CICS\_RB**

異常終了の時点で、CICS RB が制御下にありませんでした (つまり、異常終了は CICS によって呼び出されたシステム・サービス内で発生しました)。

**SRP\_CICS\_ERROR\_REASON**

MVS 異常終了理由コードを含む 4 文字のフィールド。このフィールドに値が含まれるのは、フラグ SRP\_VALID\_REASON が設定されている場合のみです。

**SRP\_CICS\_ERROR\_DATA**

異常終了の前に、CICS が最後に行ったことを記述する領域。以下のものが含まれます。

**SRP\_CICS\_EC\_PSW**

拡張制御 (EC) モードのプログラム状況ワード (PSW) を含む 8 文字のフィールド

**SRP\_CICS\_PSW16**

128 ビット PSW を含む 16 文字のフィールド。

**SRP\_CICS\_EC\_INT**

中断コードと ILC を含む 8 文字のフィールド

**SRP\_CICS\_REGST**

汎用 (GP) レジスタの内容を含む 64 文字のフィールド

**SRP\_CICS\_EXEC\_KEY**

形式 X'On' の PSW キーを含む 1 バイト。

**SRP\_SYSTEM\_ERROR\_DATA**

異常終了の前にシステムが最後に行ったことを記述する領域。以下のものが含まれます。

**SRP\_SYSTEM\_EC\_PSW**

EC モード PSW を含む 8 文字のフィールド

**SRP\_SYSTEM\_PSW16**

128 ビット PSW を含む 16 文字のフィールド。

**SRP\_SYSTEM\_EC\_INT**

中断コードと ILC を含む 8 文字のフィールド

**SRP\_SYSTEM\_REGST**

GP レジスターの内容を含む 64 文字のフィールド

**SRP\_SYSTEM\_EXEC\_KEY**

形式 X'On' の PSW キーを含む 1 バイト。

**SRP\_ERROR\_FP\_REGS**

異常終了の時点の浮動小数点レジスターの内容を記述する領域。内容は次のとおりです。

**SRP\_FP\_REG\_0**

FP レジスター 0

**SRP\_FP\_REG\_2**

FP レジスター 2

**SRP\_FP\_REG\_4**

FP レジスター 4

**SRP\_FP\_REG\_6**

FP レジスター 6

**SRP\_ADDITIONAL\_REG\_INFO**

追加のレジスター情報を含む領域。

**SRP\_ADDITIONAL\_REGS\_FLAG**

以下のフラグを含む 1 バイト:

**SRP\_CICS\_GPR64\_AVAIL**

64 ビットの CICS GP レジスターが使用可能です。

**SRP\_SYSTEM\_GPR64\_AVAIL**

64 ビットのシステム GP レジスターが使用可能です。

**SRP\_ADDITIONAL\_FPR\_AVAIL**

追加の FP レジスターが使用可能です。

**SRP\_CICS\_GP64\_REGS**

異常終了の時点の CICS 64 ビット GP レジスターを含む 128 バイトの領域。

**SRP\_SYSTEM\_GP64\_REGS**

異常終了の時点のシステム 64 ビット GP レジスターを含む 128 バイトの領域。

**SRP\_ADDITIONAL\_FPR\_REGS**

異常終了の時点の追加の FP レジスターを含む 132 バイトの領域。

**SRP\_FP\_REGS**

異常終了の時点のすべての FP レジスターを含む 128 バイトの領域。

**SRP\_FPC\_REG**

異常終了の時点の FPC レジスターを含む 4 バイトのフィールド。

**SRP\_VR\_REGS**

異常終了の時点のすべてのベクトル・レジスターを含む 512 バイトの領域。

フラグ SRP\_NOT\_CICS\_RB が設定されている場合、SRP\_CICS\_ERROR\_DATA は異常終了の前に CICS が最後に行ったことを記述します。SRP\_SYSTEM\_ERROR\_DATA はシステム・サービス (z/OS Communications Server、VSAM、または MVS など) が最後に行ったことを記述します。

DFHSRED TYPE=DSECT マクロを使用することで、XSRAB に渡された SRP\_ERROR\_DATA をマップできます。SRP\_ERROR\_DATA の形式については、[SRED - システム・リカバリー・エラー・データ](#)を参照してください。

## 戻りコード

### UERCNOCA

異常終了コード ASRB でタスクを異常終了します。このタスクに関連したプログラム・レベルの異常終了出口を取り消さないでください。

### UERCCANC

異常終了コード ASRB でタスクを異常終了します。このタスクに関連したプログラム・レベルの異常終了出口を取り消してください。

### UERCCICS

CICS を異常終了します。

## XPI 呼び出し

CICS はエラー環境で出口 XSRAB を呼び出すため、XPI 呼び出しのサブセットのみを使用できます。

一般の用途には TRACE\_PUT のみを使用できます。

WAIT\_MVS を使用できます。ただし使用できるのは、出口プログラムによって、(SRP\_CICS\_CODE フィールドおよび SRP\_USER\_CODE フィールドをもとに) 異常終了が CICS コード内ではなく、ユーザー・アプリケーション・コード内で発生したことが判明した後に限られます。

## 重要:

- XSRAB 出口点で実行するようプログラムをコーディングする際には注意してください。出口プログラムによってシステム・リカバリー・プログラムに再入する結果となる場合 (例えば、プログラム・チェックが行われる場合)、CICS は異常終了し、DFHSR06xx メッセージが発行されます。
- デフォルトの戻りコードは UERCNOCA です。したがって、出口がエラー状態の場合には、確実にタスクが異常終了します。
- 出口は障害後に呼び出されるため、この出口点において UERCNORM 戻りコードはありません。
- 出口が戻りコード UERCPURG を設定してはなりません。

## システム終了プログラム出口 XSTERM

XSTERM 出口は、統計 SMF データ・セットに最終統計を出力し、SMF データ・セットを閉じるために使用できます。CICS VSAM データ・セットおよび BDAM データ・セットは、この出口が呼び出される前に CICS ファイル制御によって既に閉じられていることに注意してください。

## 呼び出される状況

通常のシステム・シャットダウンの第 2 静止ステージ内で、一時データ・バッファおよび一時ストレージ・バッファがクリアされる直前。IMMEDIATE シャットダウン時には、この出口は呼び出されません。

## 出口固有のパラメーター

なし。

## 戻りコード

### UERCNORM

処理が続行される。

## XPI 呼び出し

WRITE\_JOURNAL\_DATA 以外のすべての XPI 呼び出しが使用可能です。ただし、タスクが制御を失い、他のタスクがさらに多くのモニター・データを書き込めるようになる可能性があるため、使用はお勧めできません。

## 一時記憶域ドメイン出口 (XTSQRIN、XTSQROUT、XTSPTIN、XTSPTOUT)

これらの出口は、一時記憶域を制御またはモニターする必要があるときに呼び出されます。

以下のタスクを実行するように、一時記憶域ドメイン出口 XTSQRIN、XTSQROUT、XTSPTIN、および XTSPTOUT を変更できます。

- キューを作成する要求の場合に、キューを主記憶域または補助記憶域のどちらに格納するか、およびキューのリカバリー可能性を指定する

- 一時記憶域の使用状況をモニターする
- 一時記憶域キューのセキュリティーを制御する

UEPTERM パラメーターは、IPIC 接続を介して機能シッパされた一時ストレージ要求ではゼロの値です。一時ストレージ要求で IPIC 接続を使用するには、XTSQRIN、XTSQROUT、XTSPTIN、および XTSPTOUT が、それをアドレスとして使用することを試行する前に UEPTERM パラメーターがゼロ以外の値であることを確認するようにします。

XTSQRIN、XTSQROUT、XTSPTIN、および XTSPTOUT で、IPIC 接続を使用したスレッド・セーフの利点を活用するには、スレッド・セーフ標準に沿ってコード化して、スレッド・セーフを宣言する必要があります。

一時記憶域ドメインには、以下の関数をサポートする TSQR と TSPT の 2 つのメインゲートがあります。

#### **TSQR**

Write、Rewrite、Read\_into、Read\_set、Read\_next\_into、Read\_next\_set、Delete

#### **TSPT**

Put、Put\_replace、Get、Get\_set、Get\_release、Get\_release\_set、Release

TSQR 関数は、EXEC CICS インターフェース (または DFHTS TYPE=PUTQ、GETQ、または PURGE) で使用可能なものに対応します。TSPT 関数は、START 関数と RETRIEVE 関数 (または DFHTS TYPE=PUT、GET、または RELEASE) をサポートするために、インターバル制御プログラムによって使用されます。

## **出口 XTSQRIN**

出口 XTSQRIN は、ユーザー TS キューに対するユーザー一時ストレージ・インターフェース 要求 (例えば WRITEQ TS 要求または READQ TS 要求) の実行前に呼び出されます。

### **出口固有のパラメーター**

#### **UEPTRANID**

4 バイトのトランザクション ID のアドレス。

#### **UEPUSER**

8 バイトのユーザー ID のアドレス。

#### **UEPTERM**

ゼロ、または 4 バイトの端末 ID のアドレス。戻されるアドレスがない場合は、この要求が IPIC 接続を介して機能シッパされた可能性があります。

#### **UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。

#### **UEP\_TS\_FUNCTION**

関数を含む 1 バイトのアドレス。

- 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**

キュー名が含まれた 16 バイト・フィールドのアドレス。

#### **UEP\_TS\_DATA\_P**

データのアドレスが含まれたフルワードのアドレス。(書き込み要求および再書き込み要求)。

#### **UEP\_TS\_DATA\_L**

データの長さが含まれたフルワードのアドレス。(書き込み要求および再書き込み要求)。



### **UEP\_TS\_ITEM\_NUMBER**

項目番号が含まれたフルワードのアドレス。(再書き込み要求、read\_into 要求、および read\_set 要求)。

### **UEP\_TS\_STORAGE\_TYPE**

ストレージ・タイプが含まれた 1 バイトのアドレス。(書き込み要求)。

出口への入力で、このパラメーターは UEP\_TS\_STORAGE\_TYPE\_MAIN または UEP\_TS\_STORAGE\_TYPE\_AUX\_TST のいずれかに設定されます。このパラメーターは、出口によって以下のいずれかの値に変更できます。

CICS® が TS メインのみのサポートで初期設定された場合は、このパラメーターを設定しても効果がないことに注意してください。詳しくは、[TS システム 初期設定パラメーター](#)の説明を参照してください。

### **UEP\_TS\_STORAGE\_TYPE\_MAIN**

主記憶域。

### **UEP\_TS\_STORAGE\_TYPE\_AUX\_TST**

補助記憶域 (リカバリー可能性はリソース定義により決定)。

### **UEP\_TS\_STORAGE\_TYPE\_AUX\_RECOV\_YES**

補助記憶域 (リカバリー可能)。

### **UEP\_TS\_STORAGE\_TYPE\_AUX\_RECOV\_NO**

補助記憶域 (リカバリー不能)。

## 戻りコード

### **UERCNORM**

正常。

### **UERCPURG**

ページされました。

## XPI 呼び出し

すべて使用できます。

## API および SPI 呼び出し

何も使用できません。

## 出口 XTSQROUT

出口 XTSQROUT は、ユーザー TS キューに対するユーザー一時ストレージ・インターフェース 要求 (例えば WRITEQ TS 要求または READQ TS 要求) の実行後に呼び出されます。

### 出口固有のパラメーター

### **UEPTRANID**

4 バイトのトランザクション ID のアドレス。

### **UEPUSER**

8 バイトのユーザー ID のアドレス。

### **UEPTERM**

ゼロ、または 4 バイトの端末 ID のアドレス。戻されるアドレスがない場合は、この要求が IPIC 接続を介して機能シップされた可能性があります。

### **UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。

### **UEP\_TS\_FUNCTION**

関数を含む 1 バイトのアドレス。

- 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**

キュー名が含まれた 16 バイト・フィールドのアドレス。

#### **UEP\_TS\_DATA\_P**

データのアドレスが含まれたフルワードのアドレス。(削除要求以外のすべての要求)。

#### **UEP\_TS\_DATA\_L**

データの長さが含まれたフルワードのアドレス。(削除要求以外のすべての要求)。

#### **UEP\_TS\_ITEM\_NUMBER**

項目番号が含まれたフルワードのアドレス。(再書き込み要求、read\_into 要求、および read\_set 要求)。

#### **UEP\_TS\_TOTAL\_ITEMS**

キュー内の項目の合計数が含まれたフルワードのアドレス。(削除要求以外のすべての要求)。

#### **UEP\_TS\_RESPONSE**

要求の完了後の応答が含まれた 1 バイトのアドレス。

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

### 戻りコード

#### **UERCNORM**

通常応答。

#### **UERCPURG**

ページされた応答を XPI 要求から受け取りました。

### **XPI 呼び出し**

すべて使用できます。

### **API および SPI 呼び出し**

何も使用できません。

## 出口 XTSPTIN

出口 XTSPTIN は、CICS 内部キュー (例えばインターバル制御キューまたは BMS キュー) に対する一時ストレージ・インターフェース 要求の実行前に呼び出されます。

### 出口固有のパラメーター

#### **UEPTRANID**

4 バイトのトランザクション ID のアドレス。

#### **UEPUSER**

8 バイトのユーザー ID のアドレス。

#### **UEPTERM**

ゼロ、または 4 バイトの端末 ID のアドレス。戻されるアドレスがない場合は、この要求が IPIC 接続を介して機能シップされた可能性があります。

#### **UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。

#### **UEP\_TS\_FUNCTION**

関数を含む 1 バイトのアドレス。

- 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**

キュー名が含まれた 16 バイト・フィールドのアドレス。

#### **UEP\_TS\_DATA\_P**

データのアドレスが含まれたフルワードのアドレス。(put および put\_replace)。

#### **UEP\_TS\_DATA\_L**

データの長さが含まれたフルワードのアドレス。(put および put\_replace)。

#### **UEP\_TS\_STORAGE\_TYPE**

ストレージ・タイプが含まれた 1 バイトのアドレス。(put 要求)。

出口への入力で、このパラメーターは UEP\_TS\_STORAGE\_TYPE\_MAIN または UEP\_TS\_STORAGE\_TYPE\_AUX\_TST のいずれかに設定されます。このパラメーターは、出口によって以下のいずれかの値に変更できます。

CICS が TS メインのみのサポートで初期設定された場合は、このパラメーターを設定しても効果がないことに注意してください。

#### **UEP\_TS\_STORAGE\_TYPE\_MAIN**

主記憶域。

#### **UEP\_TS\_STORAGE\_TYPE\_AUX\_TST**

補助記憶域 (リカバリー可能性はリソース定義により決定)。

#### **UEP\_TS\_STORAGE\_TYPE\_AUX\_RECOV\_YES**

補助記憶域 (リカバリー可能)。

#### **UEP\_TS\_STORAGE\_TYPE\_AUX\_RECOV\_NO**

補助記憶域 (リカバリー不能)。

### 戻りコード

#### **UERCNORM**

正常。

#### **UERCPURG**

タスクは XPI 呼び出し中にページされます。

### **XPI 呼び出し**

すべて使用できます。

### **API および SPI 呼び出し**

何も使用できません。

## 出口 XTSPROUT

出口 XTSPROUT は、CICS 内部キュー (例えばインターバル制御キューまたは BMS キュー) に対する一時ストレージ・インターフェース 要求の実行後に呼び出されます。TSPT 要求の実行後。パラメーターを変更することはできません。

### 出口固有のパラメーター

#### **UEPTRANID**

4 バイトのトランザクション ID のアドレス。

#### **UEPUSER**

8 バイトのユーザー ID のアドレス。

**UEPTERM**

ゼロ、または 4 バイトの端末 ID のアドレス。戻されるアドレスがない場合は、この要求が IPIC 接続を介して機能シップされた可能性があります。

**UEPPROG**

8 バイトのアプリケーション・プログラム名のアドレス。

**UEP\_TS\_FUNCTION**

関数を含む 1 バイトのアドレス。

- 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**

キュー名が含まれた 16 バイト・フィールドのアドレス。

**UEP\_TS\_DATA\_P**

データのアドレスが含まれたフルワードのアドレス。(解放要求以外のすべての要求)。

**UEP\_TS\_DATA\_L**

データの長さが含まれたフルワードのアドレス。(解放要求以外のすべての要求)。

**UEP\_TS\_RESPONSE**

要求の完了後の応答が含まれた 1 バイトのアドレス。

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

**戻りコード****UERCNORM**

通常応答。

**UERCPURG**

パージされた応答を XPI 要求から受け取りました。

**XPI 呼び出し**

すべて使用できます。

**API および SPI 呼び出し**

何も使用できません。

## 一時ストレージ EXEC インターフェース・プログラム出口 XTSEREQ および XTSEREQC

XTSEREQ 出口により、要求に対して何かアクションが起こる前に一時ストレージの API 要求を代行受信できます。また、XTSEREQC 出口では、一時ストレージの API 要求が完了してから応答を代行受信できます。

影響を受ける API 要求は、次のとおりです。

- EXEC CICS WRITEQ TS
- EXEC CICS READQ TS
- EXEC CICS DELETEQ TS

XTSREQ を使用して、以下のことができます。

- API パラメーター・リストの分析 (関数、キーワード、引数値、および応答)
- 要求実行前の入力パラメーター値の変更
- 要求実行の回避

XTSREQC を使用して、以下のことができます。

- API パラメーター・リストの分析
- 要求完了後の出力パラメーター値の変更

また、以下のことも可能です。

- XTSREQ 出口プログラムと XTSREQC 出口プログラムが同じ要求のために呼び出されたときに、プログラム間でデータを渡します
- 複数の一時ストレージ出口プログラムが同じタスク内で呼び出されたときに、そのプログラム間でデータを渡します。

一時ストレージ・ドメイン内で出口から呼び出される各プログラム (XTSQRIN、XTSQROUT、XTSPTIN、および XTSPTOUT) は、XTSREQ によりセットアップされたシチュエーションを変更する可能性があります。このため、出口が呼び出される順番を考慮する必要があります。

すべての一時ストレージ出口が使用可能になっている場合、呼び出しの順番は次のようになります。

1. XTSREQ
2. XTSQRIN
3. XTSQROUT
4. XTSREQC

## 出口 XTSREQ

XTSREQ 出口により、要求に対して何かアクションが起こる前に一時ストレージの API 要求を代行受信できます。

### 呼び出される状況

CICS が一時記憶域 API 要求を処理する前。

### 出口固有のパラメーター

#### UEPCLPS

コマンド・パラメーター・リストのコピーのアドレス。207 ページの『[コマンド・レベルのパラメーター構造](#)』を参照してください。

#### UEPTQTOK

単一の一時記憶域要求について、XTSREQ と XTSREQC の間で情報を受け渡すために使用できる 4 バイト領域のアドレス。

#### UEPRCODE

EIB 戻りコード EIBRCODE の 6 バイト 16 進数コピーのアドレス。EIB 戻りコードの詳細については、[EIB フィールド](#)を参照してください。

#### UEPRES

EIB 応答コード EIBRESP の 4 バイトのバイナリー・コピーのアドレス。

#### UEPRES2

EIB 応答コード EIBRESP2 の 4 バイトのバイナリー・コピーのアドレス。

#### UEPTSTOK

同じタスク内の連続した一時記憶域要求の間で情報を受け渡すために使用できる 4 バイト・トークンのアドレス (例えば、XTSREQ 出口の連続した呼び出しの間)。

#### UEPRECUR

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

**UEPRSRCE**

EIB リソース値の EIBRSRCE の 8 文字のコピーのアドレス。

**戻りコード****UERCMBYP**

この要求をバイパスします。

**UERCNORM**

処理が続行される。

**UERCPURG**

タスクは XPI 呼び出し中にページされます。

**XPI 呼び出し**

すべてを使用できます。

**API および SPI コマンド**

次のコマンドを除き、すべてを使用できます。

EXEC CICS SHUTDOWN

EXEC CICS XCTL

**注:** 再帰的コマンドを発行するときには注意してください。例えば、XTSREQ 出口から一時記憶域要求を発行するときにループに入らないようにする必要があります。再帰カウンタ UEPRECUR を使用することをお勧めします。

**出口 XTSREQC**

また、XTSREQC 出口では、一時ストレージの API 要求が完了してから応答を代行受信できます。

**呼び出される状況**

CICS が一時記憶域 API 要求を処理した後、一時記憶域 EXEC インターフェース・プログラムから戻る前。

**出口固有のパラメーター****UEPCLPS**

コマンド・パラメーター・リストのコピーのアドレス。207 ページの『コマンド・レベルのパラメーター構造』を参照してください。

**UEPTQTOK**

単一の一時記憶域要求について、XTSREQ と XTSREQC の間で情報を受け渡すために使用できる 4 バイト領域のアドレス。

**UEPRCODE**

EIB 戻りコード EIBRCODE の 6 バイトの 16 進数コピーのアドレス。EIB 戻りコードの詳細については、[EIB フィールド](#)を参照してください。

**UEPRES**

EIB 応答コード EIBRESP の 4 バイトのバイナリー・コピーのアドレス。

**UEPRES2**

EIB 応答コード EIBRESP2 の 4 バイトのバイナリー・コピーのアドレス。

**UEPTSTOK**

同じタスク内の連続した一時記憶域要求の間で情報を受け渡すために使用できる 4 バイト・トークンのアドレス (例えば、XTSREQC 出口の連続した呼び出しの間)。

**UEPRECUR**

ハーフワードの再帰カウンタのアドレス。このカウンタは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

**UEPRSRCE**

EIB リソース値の EIBRSRCE の 8 文字のコピーのアドレス。

**UEP\_TS\_REMOTE\_SYSTEM**

要求をリモート領域に送信する場合は、そのリモート領域の 4 バイトの名前が含まれた領域のアドレス。リモート領域は、例えばコマンドの SYSID オプション、機能シップ、またはワークロード管理によって指定されます。

要求がローカル領域で実行される場合、このパラメーターはブランクを含む 4 バイト領域のアドレスです。

## 戻りコード

### **UERCNORM**

処理が続行される。

### **UERCPUrg**

タスクは XPI 呼び出し中にページされます。

## **XPI 呼び出し**

すべてを使用できます。

## **API および SPI コマンド**

次のコマンドを除き、すべてを使用できます。

### **EXEC CICS SHUTDOWN**

### **EXEC CICS XCTL**

パラメーター・リストで指定されている EIBRSRCE、EIBRCODE、EIBRESP、および EIBRESP2 のコピーを更新できます。これらの値を更新すると、XTSREQC の完了後、または戻りコード UERCBYP を XTSREQ に指定した場合に、新しい値が一時記憶域によってアプリケーション・プログラムの EIB にコピーされます。

有効な一時記憶域の応答を設定する必要があります。EIBRCODE、EIBRESP、および EIBRESP2 の 3 つすべてを、整合性のある値のセットに設定する必要があります (例えば、有効な完了を示すために一時記憶域によって設定される値に)。CICS は、EIBRCODE、EIBRESP、および EIBRESP2 の整合性を検査しません。EIBRCODE がゼロ以外の値に設定され、EIBRESP がゼロに設定されると、CICS はゼロ以外の値を使用して EIBRESP をオーバーライドします。EIBRCODE、EIBRESP、および EIBRESP2 の値の設定を支援するために、一時記憶域によって使用される値は DSECT DFHTSUED で指定されます。

注: 再帰的コマンドを発行する際には、ループが生じないように注意してください。例えば、XTSREQC 出口から一時記憶域要求を発行するときにループに入らないようにする必要があります。再帰カウンター UEPRECUR を使用することをお勧めします。

## コマンド・レベルのパラメーター構造

コマンド・レベルのパラメーター構造は、一連のアドレスで構成されます。最初のアドレスは、EXEC インターフェース記述子 (EID) を指します。これは、要求のタイプを記述するビット・ストリングからなり、要求で指定された各キーワードを識別します。残りのアドレスは、その要求に関連付けられたデータ部分を指します。

### パラメーター・リスト終了標識

EID を調べて、要求のタイプおよび指定されたキーワードを判別することができます。リスト内のその他のパラメーターを調べて、キーワードの値を判別することができます。要求で指定されたキーワードの値を変更することもできます。

パラメーター・リスト内の最後のアドレス・セットの高位ビットが、リスト内の最後のアドレスであることを示すために設定されます。ユーザー出口プログラムから戻ったときに、CICS は、パラメーター・リストをスキャンして高位ビットを探し、最終パラメーターを検出します。したがって、パラメーター・リストの長さを変更した場合は、高位ビットをリセットしてどれが新しい最終アドレスかを示す必要もあります。

### **UEPCLPS 出口固有パラメーター**

UEPCLPS 出口固有パラメーターは、出口 XTSREQ と出口 XTSREQC の両方に組み込まれています。このパラメーターは、コマンド・レベルのパラメーター構造のアドレスです。

コマンド・レベルのパラメーター構造には、TS\_ADDR0 から TS\_ADDR7 までの 8 つのアドレスが含まれています。この構造は、DSECT TS\_ADDR\_LIST 内に定義されますが、それは COPY DFHTSUED というステートメントを組み込むことによって、出口プログラムにコピーする必要があります。

コマンド・レベルのパラメーター・リストは、以下のように構成されています。

注：以下の表では、一時記憶域コマンドの引数、キーワード、データ・タイプ、および入出力タイプの関係を要約しています。

表 8. 一時記憶域コマンドの引数、キーワード、データ・タイプ、および入出力タイプの関係	
コマンド	参照
WRITEQ TS	<a href="#">211 ページの表 9</a>
READQ TS	<a href="#">211 ページの表 10</a>
DELETEQ TS	<a href="#">212 ページの表 11</a>

## TS\_ADDR0

EID という 9 バイトの領域のアドレスのことで、以下のように構成されています。

- TS\_GROUP
- TS\_FUNC
- TS\_BITS1
- TS\_BITS2
- TS\_EIDOPT5
- TS\_EIDOPT6
- TS\_EIDOPT7
- TS\_EIDOPT8

## TS\_GROUP

常に X'0A' で、これが一時記憶域要求であることを示します。

## TS\_FUNC

以下の要求のタイプを定義する 1 バイトの領域です。

X'02'

WRITEQ

X'04'

READQ

X'06'

DELETEQ

## TS\_BITS1

どの引数が指定されたかを定義する存在ビット。キーワードに関連付けられた引数を取得するには、コマンド・レベルのパラメーター構造から適切なアドレスを使用する必要があります。このアドレスを使用する前に、関連付けられた存在ビットを確認する必要があります。存在ビットがオフに設定されている場合は、引数が要求に指定されていないため、このアドレスを使用してはなりません。

X'80'

要求に QUEUE キーワードまたは QNAME キーワードの引数が含まれている場合に設定されます。設定される場合は、**TS\_ADDR1** が意味を持ちます。

X'40'

要求に FROM、INTO、または SET のいずれかのキーワードの引数が含まれている場合に設定されます。設定される場合は、**TS\_ADDR2** が意味を持ちます。

X'20'

要求に LENGTH キーワードの引数が含まれている場合に設定されます。設定される場合は、**TS\_ADDR3** が意味を持ちます。

X'10'

要求に NUMITEMS キーワードの引数が含まれている場合に設定されます。設定される場合は、**TS\_ADDR4** が意味を持ちます。



**X'08'**

要求に NUMITEMS キーワードまたは ITEM キーワードの引数が含まれている場合に設定されます。設定される場合は、**TS\_ADDR5** が意味を持ちます。

**X'02'**

要求に SYSID キーワードの引数が含まれている場合に設定されます。設定される場合は、**TS\_ADDR7** が意味を持ちます。

**TS\_BITS2**

一時記憶域によって使用されない 2 バイト。

**TS\_EIDOPT5**

特定のキーワードが要求に指定されたかどうかを示します。

**X'80'**

QNAME が指定されました (これ以外の場合は QUEUE)。このビットは、必要に応じてユーザー 出口内で変更できます。

**TS\_EIDOPT6**

一時記憶域によって使用されない 1 バイト。

**TS\_EIDOPT7**

特定の機能またはキーワード (あるいはその両方) が要求に指定されたかどうかを示します。

**X'10'**

WRITEQ NOSUSPEND が指定されました。

**X'80'**

WRITEQ MAIN または READQ ITEM が指定されました。

**X'04'**

WRITEQ REWRITE または READQ NUMITEMS が指定されました。

**TS\_EIDOPT8**

特定のキーワードが要求に指定されたかどうかを示します。

**X'80'**

ITEM が指定されました (これ以外の場合は NUMITEMS)。

**TS\_ADDR1**

QUEUE からの 8 バイトの名前、または QNAME からの 16 バイトの名前を含む領域のアドレス。これらのどちらが適用されるかを判別するには、**TS\_BITS2** フィールドを参照してください。

**TS\_ADDR2**

次のアドレスのいずれかです。

- SET からの 4 バイトのアドレス (要求が READQ で、これが SET であることが **TS\_EIDOPT5** に示されている場合)。
- INTO からのデータ (要求が READQ で、これが SET でないことが **TS\_EIDOPT5** に示されている場合)。
- FROM からのデータ (要求が WRITEQ の場合)。

**TS\_ADDR3**

LENGTH のハーフワード値のアドレス (要求が READQ または WRITEQ の場合)。

**Warning :** INTO を指定する要求の場合は、LENGTH の値をアプリケーションで指定された値よりも大きい値に変更しないでください。この値を大きい値に変更すると、アプリケーション内でストレージ・オーバーレイが発生します。

**TS\_ADDR4**

NUMITEMS のハーフワード値のアドレス (要求が READQ の場合)。

**TS\_ADDR5**

次のアドレスのいずれかです。

- NUMITEMS のハーフワード値 (要求が WRITEQ の場合)
- ITEM のハーフワード値 (要求が READQ または WRITEQ の場合)

## TS\_ADDR6

CICS の内部使用専用の値のアドレスです。このアドレスは使用できません。

## TS\_ADDR7

SYSID の値が含まれている領域のアドレスです。

### コマンド・レベルのパラメーター構造のフィールドの変更

一時ストレージに渡される一部のフィールドは要求への入力として使用され、一部は出力フィールドとして使用され、さらに一部は入力および出力の両方に使用されます。ユーザー出口プログラムでフィールドを変更するために使用する方法は、フィールドの使用方法に応じて異なります。

以下は、常に入力フィールドです。

- QUEUE|QNAME
- FROM
- SYSID

以下は、常に出力フィールドです。

- INTO
- NUMITEMS
- SET

LENGTH は、WRITEQ 要求では入力フィールドで、SET が指定された READQ 要求では出力フィールドです。INTO が指定された READQ 要求では、入力フィールドと出力フィールドの両方になります。

ITEM は、READQ 要求および REWRITE が指定された WRITEQ 要求では入力フィールドです。REWRITE が指定されていない WRITEQ 要求では、入力フィールドと出力フィールドの両方になります。

### 入力フィールドの変更

入力フィールドを変更する場合の正しい方法は、そのフィールドの新規コピーを作成し、コマンド・レベル・パラメーター・リスト内のアドレスを変更して新規データを指すようにすることです。

コマンド・レベルのパラメーター・リストによって指定されたデータを変更することによって、入力フィールドを変更してはなりません。これを行うと、アプリケーション・プログラムに属しているストレージが破壊され、プログラムがフィールドを再使用しようとしたときに障害が発生することになります。

### 出力フィールドの変更

210 ページの『入力フィールドの変更』で説明されている手法は、出力フィールドの変更には適していません。(結果が、アプリケーションの領域ではなく新しい領域に返されるため、アプリケーションで表示できません。)

出力フィールドの変更は、コマンド・レベル・パラメーター・リストが指すデータを変更することによって行います。出力フィールドの場合は、アプリケーションのデータを所定の位置で変更することができます。これは、アプリケーションが、そのフィールドが何らかの方法で変更されることを予期しているためです。

### 入力と出力の両方に使用されるフィールドの変更

入力と出力の両方に使用されるフィールドの例として、INTO を指定する READQ 要求の LENGTH があります。この種のフィールドは、出力フィールドと同様の方法で取り扱うことができ、同じものと見なされます。

### EID の変更

EID を変更することで要求に対して大きな変更を行うことはできません。例えば、READQ 要求を WRITEQ 要求に変更することはできません。ただし、要求に対する小さな変更を行うことはできます。

以下のリストは、EID 内の変更可能なビットを示しています。EID のその他の部分を変更しようとしても、すべて無視されます。

## TS\_BITS1

### X'02'

SYSID の存在ビット。

## TS\_EIDOPT7

XTSREQ のユーザー出口プログラムは、すべての WRITEQ TS コマンドについて、以下をオンまたはオフに設定できます。

### X'10'

NOSUSPEND の存在ビット。

### X'08'

MAIN の存在ビット。

EID は、アプリケーション・プログラムに戻る前に、元の値にリセットされます。つまり、EID に対して行われた変更は、一時ストレージ要求の期間中のみ保持されます。

注：ユーザー出口プログラムは、EID に大きな変更を加えることはできません。ただし、許容されている小さい変更を加える場合でも、十分に注意する必要があります。

## タスク・トークン UEPTSTOK の使用

タスク・トークン UEPTSTOK は、同じタスク内の連続する一時ストレージ要求間で情報を渡すために使用できる 4 バイト領域のアドレスを提供します。

例えば、UEPTSTOK を使用して、XTSREQ 出口の連続した呼び出し間で情報を渡すことができます。これに対して、UEPTQ TOK は、単一の一時記憶域要求の期間中だけ使用可能です。これは、その内容が要求の終了時に破棄される可能性があるからです。

表 9. WRITEQ TS: ユーザー引数および関連のキーワード、データ型、および入出力タイプ			
引数	キーワード	データ型	入出力タイプ
Arg1	QUEUE	CHAR(8)	入力
Arg1	QNAME	CHAR(16)	入力
Arg2	FROM	DATA-AREA	入力
Arg3	LENGTH	BIN(15)	入力
Arg4	*	*	*
Arg5	ITEM	BIN(15)	入出力
Arg5	NUMITEMS	BIN(15)	出力
Arg6	*	*	*
Arg7	SYSID	CHAR(4)	入力

注：Arg5 には異なる使用法が示されています。これは、Arg5 が代替的な ITEM キーワードと NUMITEMS キーワードで使用され、REWRITE の使用時には ITEM キーワードの引数は入力フィールドになるためです。

表 10. READQ TS: ユーザー引数および関連のキーワード、データ型、および入出力タイプ			
引数	キーワード	データ型	入出力タイプ
Arg1	QUEUE	CHAR(8)	入力
Arg1	QNAME	CHAR(16)	入力
Arg2	SET	DATA-AREA、PTR	出力
Arg2	INTO	DATA-AREA	出力
Arg3	LENGTH	BIN(15)	入出力
Arg4	NUMITEMS	BIN(15)	出力

表 10. READQ TS: ユーザー引数および関連のキーワード、データ型、および入出力タイプ (続き)			
引数	キーワード	データ型	入出力タイプ
Arg5	ITEM	BIN(15)	入力
Arg6	*	*	
Arg7	SYSID	CHAR(4)	入力

表 11. DELETEQ TS: ユーザー引数および関連のキーワード、データ型、および入出力タイプ			
引数	キーワード	データ型	入出力タイプ
Arg1	QUEUE	CHAR(8)	入力
Arg1	QNAME	CHAR(16)	入力
Arg2	*	*	*
Arg3	*	*	*
Arg4	*	*	*
Arg5	*	*	*
Arg6	*	*	*
Arg7	SYSID	CHAR(4)	入力

### ユーザー引数の変更

ユーザー出口プログラムは、以下の方法でユーザー引数を変更することができます。

入力引数の場合、出口プログラムは変更後の引数を保持するために十分なストレージを獲得し、そのストレージを必要な値にセットアップし、パラメーター・リスト内の関連ポインターを新しく獲得された領域のアドレスに設定する必要があります。

出力引数および入出力引数の場合、出口プログラムは所定の位置の引数を更新することができます。これは、ストレージの領域が CICS から値を受け取ると予期されているアプリケーション内の変数によって表されるためです。

注：

1. CICS は、引数値に対する変更を検査しません。そのため、変更はすべて、その変更を行うユーザー出口プログラムで検証する必要があります。
2. XTSEREQ による出力引数の変更や XTSEREQC による入力引数の変更はお勧めできません。

### ユーザー引数の追加

グローバル・ユーザー出口プログラムで、SYSID キーワードに関連付けられた引数を追加することができます。出口プログラムで指定または変更する引数が有効なことをユーザーが確認する必要があります。

追加する引数がまだ存在していないことを前提として、ユーザー出口プログラムで以下を実行する必要があります。

1. 追加する引数用のストレージを取得する。
2. そのストレージを必要な値に初期設定する。
3. パラメーター・リストから適切なポインターを選択してセットアップする。
4. EID 内で適切な引数存在ビットを選択してセットアップする。
5. 新しいリスト終了標識を反映するようにパラメーター・リストを変更する。

## ユーザー引数の削除

ユーザー出口プログラムで、SYSID キーワードに関連付けられた引数を削除することができます (処理についての全責任はプログラムにあります)。

削除する引数が存在することを前提として、ユーザー出口プログラムで以下を実行する必要があります。

1. 該当の引数の存在ビットを EID 内で '0'b に切り替える。
2. 新しいリスト終了標識を反映するようにパラメーター・リストを変更する。

## プログラム例

CICS では、一時ストレージ要求を変更する方法を示すプログラム例として、DFH\$XTSE を提供しています (これは、ソフトコピー・リストとしてのみの提供で、ソース・コード・ファイルとして提供されているものではありません)。

## 端末割り振りプログラム出口 XALCAID

XALCAID は、データを伴う自動開始記述子 (AID) が、SET TERMINAL コマンドまたは SET CONNECTION コマンドの実行中か、あるいは端末または接続の再インストール中に、CEMT トランザクションによって取り消された場合に呼び出されます。

XALCAID が呼び出されるのは、AID に関連したデータがある場合だけです。

### 呼び出される状況

データを伴う AID がキャンセルされる時。

注: 要求の取り消しを出口で防止することはできません。

### 出口固有のパラメーター

#### UEPALTSD

この要求によって開始するトランザクションのシンボル ID が含まれた 4 バイト・フィールドのアドレス。

#### UEPALTRM

この要求の宛先の端末または接続の ID が含まれた 4 バイト・フィールドのアドレス。

#### UEPALDAT

FROM オプションに指定されたデータが含まれているか、または以下のいずれかの場合に 16 進数ゼロが含まれているストレージ域のアドレス。

- AID が、FROM オプション指定のない START 要求で作成された。
- AID がチャンネルに関連付けられている (この場合、UEPALCHN が指すフィールドはブランク以外の名前に設定されます)。

#### UEPALLEN

FROM データの長さが含まれているか、または以下のいずれかの場合に 16 進数ゼロが含まれているフルワード 2 進数フィールドのアドレス。

- AID が、FROM オプション指定のない START 要求で作成された。
- AID がチャンネルに関連付けられている (この場合、UEPALCHN が指すフィールドはブランク以外の名前に設定されます)。

#### UEPALRQD

FROM データに関連付けられている REQID の値が含まれた 8 バイト・フィールドのアドレス。データは、この名前を使用して一時ストレージ・キューに保管されています。この値は、START コマンドで REQID オプションを使用して明示的に指定されているか、CICS で内部的に作成されたものです。

#### UEPALQUE

START コマンドの QUEUE オプションで指定された値が含まれているか、または QUEUE が指定されなかった場合は 16 進数ゼロが含まれた 8 バイト・フィールドのアドレス。

#### **UEPALRTE**

START コマンドの RTERMID オプションで指定された値が含まれているか、または RTERMID が指定されなかった場合は 16 進数ゼロが含まれた 4 バイト・フィールドのアドレス。

#### **UEPALRTA**

START コマンドの RTRANSID オプションで指定された値が含まれているか、または RTRANSID が指定されなかった場合は 16 進数ゼロが含まれた 4 バイト・フィールドのアドレス。

#### **UEPALFMH**

データに FMH (関連の START コマンドの FMH オプションで指定) が含まれている場合は値 X'FF' が含まれ、それ以外の場合は 16 進数ゼロが含まれている 1 場合・フィールドのアドレス。

#### **UEPALSTC**

開始コードが含まれた 2 バイト・フィールドのアドレス。これは、FEPI 開始の場合は「SZ」で、それ以外の場合は「SD」です。

#### **UEPALCHN**

AID に関連付けられたチャンネルの名前が含まれた 16 バイト・フィールドのアドレス。AID に関連付けられたチャンネルがない場合、このフィールドはブランクに設定されます。

### 戻りコード

#### **UERCNORM**

その他の戻りコードは提供されません。戻りコードの値は検査されません。

### XPI 呼び出し

次のものを使用することができます。

- INQ\_APPLICATION\_DATA
- INQUIRE\_SYSTEM

その他の XPI 呼び出しは使用しないでください。

### API コマンドおよび SPI コマンド

使用できる EXEC CICS コマンドはありません。

注 : XALTENF 出口は、「端末未認識」状態を処理するために使用され、端末割り振りプログラムからも呼び出されます。XALTENF については、[216 ページの『「端末未認識」状態出口 XALTENF および XICTENF』](#)を参照してください。

## 端末管理プログラム出口 (XTCIN、XTCOUT、XTCATT)

これらの出口は、順次装置の入出力イベントの前、またはタスク接続の前に呼び出されます。

### 出口 XTCIN

#### 呼び出される状況

順次装置に対する入力イベントの後。

#### 出口固有のパラメーター

#### **UEPTCTTE**

端末制御テーブル・エントリー (TCTTE) のアドレス。DSECT DFHTCTTE を使用して TCTTE をマップすることができます。

#### **UEPTIOA**

端末入出力域 (TIOA) のアドレス。出口プログラムでアドレスを変更しないでください。DSECT DFHTIOA を使用して TIOA をマップすることができます。ただし、フィールド TIOASAL および TIOASCA はプログラミング・インターフェースではありません。

#### **UEPTCTLE**

端末制御テーブル行エントリー (TCTLE) のアドレス。TCTLE は、DSECT DFHTCTLE を使用してマップすることができます。

### 戻りコード

#### **UERCNORM**

処理が続行される。

### **XPI 呼び出し**

すべてを使用できます。ただし、GETMAIN 呼び出しを使用して、置換 TIOA として使用するための端末クラス・ストレージを取得することはできないことに注意してください。

## **出口 XTCOUT**

### **呼び出される状況**

順次装置に対する出力イベントの前。

### **出口固有のパラメーター**

#### **UEPTCTTE**

端末制御テーブル・エントリー (TCTTE) のアドレス。DSECT DFHTCTTE を使用して TCTTE をマップすることができます。

#### **UEPTIOA**

端末入出力域 (TIOA) のアドレス。出口プログラムでアドレスを変更しないでください。DSECT DFHTIOA を使用して TIOA をマップすることができます。ただし、フィールド TIOASAL および TIOASCA はプログラミング・インターフェースでは **ありません**。

#### **UEPTCTLE**

端末制御テーブル行エントリー (TCTLE) のアドレス。TCTLE は、DSECT DFHTCTLE を使用してマップすることができます。

### **戻りコード**

#### **UERCNORM**

処理が続行される。

### **XPI 呼び出し**

すべてを使用できます。ただし、GETMAIN 呼び出しを使用して、置換 TIOA として使用するための端末クラス・ストレージを取得することはできないことに注意してください。

## **出口 XTCATT**

### **呼び出される状況**

タスク接続の前。

### **出口固有のパラメーター**

#### **UEPTCTTE**

端末制御テーブル・エントリー (TCTTE) のアドレス。DSECT DFHTCTTE を使用して TCTTE をマップすることができます。

#### **UEPTIOA**

端末入出力域 (TIOA) のアドレス。DSECT DFHTIOA を使用して TIOA をマップすることができます。ただし、フィールド TIOASAL および TIOASCA はプログラミング・インターフェースでは **ありません**。

#### **UEPTCTLE**

端末制御テーブル行エントリー (TCTLE) のアドレス。TCTLE は、DSECT DFHTCTLE を使用してマップすることができます。

#### **UEPTRAN**

4 バイトのトランザクション ID のアドレス。

### **戻りコード**

#### **UERCNORM**

処理が続行される。

### **XPI 呼び出し**

すべて使用できます。

## 「端末未認識」状態出口 XALTENF および XICTENF

「端末未認識」状態は、相互通信している CICS 領域が SHIPPABLE 端末定義と自動トランザクション開始 (ATI) の両方を使用する場合に発生することがあります。この状態は、特に自動インストールが使用される場合に起こる可能性が高くなります。

### SHIPPABLE 属性

端末専有領域 (TOR) の SHIPPABLE 属性により定義された端末は、接続されたアプリケーション所有領域 (AOR) での定義が必要ありません。トランザクション・ルーティングのサポートが必要な場合は、CICS が定義のコピーを TOR から AOR にシップします。詳細については、[端末定義と接続定義のシップ](#)を参照してください。

### 自動トランザクション開始 (ATI)

ATI は、内部で生成された要求によってトランザクションが開始される場合に発生します。例えば、次のような場合です。

- ・アプリケーションが EXEC CICS START コマンドを発行する。または、
- ・一時データのトリガー・レベルに達する。

2 つの CICS モジュールが ATI 要求を処理します。

**インターバル制御機能プログラム**は START コマンドを処理し、端末がローカル・システムで認識されていることを確認し、それから (任意の START 時間間隔が経過したときに) 端末割り振りプログラムを呼び出します。

**端末割り振りプログラム**は、インターバル制御機能プログラムまたは一時データ・トリガー・メカニズムにより呼び出され、その端末がローカル・システムで認識されていることを確認します。要求端末がリモートの場合、端末割り振りプログラムは ATI 要求をリモート・システムへシップします。リモート・システムは、トランザクション・ルーティングをローカル・システムに戻す作業を開始します。

ATI に関するガイダンス情報については、[ATI によって開始されたトランザクションの従来のルーティング](#)を参照してください。

### 「端末未認識」状態

「端末未認識」状態は、領域で認識されていない端末に ATI 要求が行われた場合に起こります。ATI 要求は、その端末に任意のトランザクション・ルーティングが起こる前、つまり、端末の定義が TOR から AOR にシップされる前に、SHIPPABLE 端末の AOR に発生することがあります。

「端末未認識」状態が起こると、インターバル制御機能プログラムと端末割り振りプログラムの両方がトランザクション開始要求を「TERMIDERR」として拒否します。

## 出口

「端末未認識」状態に対処するために、CICS はインターバル制御プログラムおよび端末割り振りプログラムでグローバル・ユーザー出口を提供しています。

### XICTENF

インターバル制御プログラム内

### XALTENF

端末割り振りプログラム内

EXEC CICS START コマンドによってインターバル制御プログラムが呼び出された後で「端末未認識」状態が発生すると、CICS は XICTENF 出口を呼び出します。一時データ・トリガー・レベルまたはインターバル制御プログラムで端末割り振りプログラムが呼び出された後で「端末未認識」状態が発生すると、CICS は XALTENF 出口を呼び出します。EXEC CICS START コマンドの結果、両方の出口が呼び出される場合があることに注意してください。

出口プログラムでは、端末が別のシステムに存在しているかどうか、そうであればどのシステムに存在しているのかを示す必要があります。CICS は、この情報の設定に役立つデータを出口プログラムに渡します。両方の出口点で同じ出口プログラムを使用することができます。CICS では、サンプル出口プログラム [DFHXTENF](#) を提供します。このプログラムは両方の出口で使用でき、変更を加えずにいくつかの代表的な状況を取り扱うことができます。



このデータベースは、TOR 以外の CICS 領域で発生した「端末未認識」状態に対処するために設計されています。TOR/AOR ペアの場合は、AOR で出口プログラムを使用可能にします。出口は TOR の「端末未認識」状態には対処できないため、通常は TOR では出口プログラムを使用可能にしないでください。ただし、複数の TOR が存在する場合、各 TOR で出口プログラムを使用可能にして、他の TOR で所有される端末の要求に対処することが必要な場合があります。この場合、出口プログラムでは、このシステムが所有して要求を拒否すべき端末を認識している必要があります（「UERCTEUN」）。出口はできるだけ多くのデータを提供しますが、プログラムのロジックはシステム設計に完全に依存します。最も複雑なケースに対する単純な解決策は、各端末の名前にその所有領域のネット名またはシステム ID を反映させることです。

### 出口から返されるデータ

出口プログラムでは、レジスター 15 に以下のように戻りコードを設定する必要があります。

#### UERCTEUN

端末が存在しません

#### UERCNETN

TOR のネット名が返されました

#### UERCYSYSI

TOR のシステム ID が返されました

戻りコード UERCNETN および UERCYSYSI の場合、出口プログラムは、端末専有領域のネット名またはシステム ID をフィールド UEPxxNTO または UEPxxSYO（ここで xx は AL または IC）に入れる必要があります。

端末専有領域が z/OS Communications Server 汎用リソースのメンバーである場合、出口プログラムは、端末のネット名をフィールド UEPxxNNO に入れる必要があります。z/OS Communications Server 汎用リソースで ATI を使用する場合は、以下を参照してください。[ATI での総称リソースの使用](#)。

## 出口 XALTENF

出口 XALTENF は、一時データやインターバル制御要求からの ATI 要求が必要とする端末がこのシステムでは認識されていないときに、端末割り振りプログラムによって呼び出されます。この出口プログラムは、端末が別の接続済み CICS システムに存在しているかどうか、もしそうであればどのシステムに存在しているのかを示す戻りコードを返すことが想定されています。

### 出口固有のパラメーター

#### UEPALEVT

要求のタイプが含まれた 2 バイトのアドレス。タイプの等価の値は次のとおりです。

#### UEPALESD

データ付きの START コマンド

#### UEPALES

データなしの START コマンド

#### UEPALETD

一時データ・トリガー・レベルに達しました。

#### UEPALTR

START コマンドを発行するタスクがトランザクション・ルーティングによって開始されたかどうかを示す標識が含まれた 1 バイトのアドレス。等価の値は次のとおりです。

#### UEPALTY

START コマンドは処理中で、コマンドを発行したタスクはルーティング先のトランザクションでした。

#### UEPALTN

START コマンドが処理されていないか、または START コマンドは処理されたが、コマンドを発行したタスクがルーティング先のトランザクションではありませんでした。

#### UEPALFS

START コマンドが機能シッパされたものかどうかを示す標識が含まれた 1 バイトのアドレス。同等の値は以下のとおりです。

#### UEPALFY

START コマンドが処理され、START は機能シッパされたものでした。

#### **UEPALFN**

START コマンドが処理されていないか、または START は処理されたが機能シッパされたものではありませんでした。

#### **UEPALTRN**

実行するトランザクションの名前が含まれた 4 バイトのアドレス。

#### **UEPALTR**

トランザクションを実行する端末の名前が含まれた 4 バイトのアドレス。(一時データ・トリガー・レベルに達し、一時データ・キュー定義にシステムが指定されている場合は、ここにシステム ID が含まれています。)

#### **UEPALCTR**

トランザクション・ルーティングされた START コマンドの場合は現行端末の名前が含まれており、機能シッパされた START コマンドの場合はセッションの名前が含まれた、4 バイトのアドレス。

その他の START コマンドおよび一時データ・トリガー・イベントの場合、指示先のフィールドにはブランクが入ります。

#### **UEPALNTI**

機能シッパされた START コマンドの場合は、要求の発行元となった最後のシステムのネット名が含まれた 8 バイトのアドレス。

このシステム内でタスクにルーティングされるトランザクションによって発行された START コマンドの場合は、タスクのルーティング元となった最後のシステムのネット名。

その他の START コマンド状況および一時データ・トリガー・レベル・イベントの場合は、指示先のフィールドにはブランクが入ります。

#### **UEPALSXI**

UEPALNTI にネット名が含まれている場合は、対応するシステム ID が含まれた 4 バイトのアドレス。

UEPALNTI にネット名が含まれていない場合、指示先のフィールドはブランクです。

#### **UEPALNTO**

UEPALNTI の内容が含まれた 8 バイトのアドレス。

これで「**UERCNETN**」の戻りコードが設定される場合、出口プログラムで、**ATI** 要求の送信先となるシステムのネット名をこのフィールドに置く必要があります。

#### **UEPALSXI**

UEPALSXI の内容が含まれた 4 バイトのアドレス。

これで「**UERCSSXI**」の戻りコードが設定される場合、出口プログラムで、**ATI** 要求の送信先となるシステムのシステム ID をこのフィールドに置く必要があります。

#### **UEPALNNI**

トランザクションを実行する端末のネット名を CICS が認識している場合は、そのネット名が含まれた 4 バイトの入力フィールドのアドレス。CICS がネット名を認識していない場合、アドレス指定されたフィールドにはブランクが入ります。

#### **UEPALNNO**

呼び出し時に、UEPALNNI の内容が含まれた 4 バイトの入出力フィールドのアドレス。出口プログラムはこのフィールドを使用して、トランザクションを実行する端末のネット名を提供することができます。ATI 要求の送信先となる TOR が z/OS Communications Server の汎用リソースのメンバーである場合、出口プログラムが端末のネット名を提供することは重要です。

### 戻りコード

#### **UERCETUN**

端末が不明で、要求は拒否されます。

#### **UERCNETN**

端末は既知のもので、UEPALNTO にネット名が返されます。

#### **UERCSSXI**

端末は既知のもので、UEPALSXI にシステム ID が返されます。

## XPI 呼び出し

次のものを使用することができます。

- INQ\_APPLICATION\_DATA
- INQUIRE\_SYSTEM

その他の XPI 呼び出しは使用しないでください。

## サンプル出口プログラム

DFHXTENF は、XALTENF 出口および XICTENF 出口に使用できるサンプル・プログラムです。DFHXTENF の詳細については、[端末未認識サンプル出口プログラム: DFHXTENF](#) を参照してください。

## 出口 XICTENF

出口 XICTENF は、EXEC CICS START コマンドが必要とする端末がこのシステムで認識されていない場合に、インターバル制御プログラムによって呼び出されます。

### 呼び出される状況

EXEC CICS START コマンドが必要とする端末がこのシステムで認識されていない場合に、インターバル制御プログラムによって。この出口プログラムは、端末が別の接続済み CICS システムに存在しているかどうか、もしそうであればどのシステムに存在しているのかを示す戻りコードを返すことが想定されています。

### 出口固有のパラメーター

#### UEPICEVT

要求のタイプが含まれた 2 バイトのアドレス。タイプの等価の値は次のとおりです。

#### UEPICESD

データ付きの START コマンド

#### UEPICES

データなしの START コマンド

#### UEPICTR

START コマンドを発行するタスクがトランザクション・ルーティングによって開始されたかどうかを示す標識が含まれた 1 バイトのアドレス。等価の値は次のとおりです。

#### UEPICTY

START コマンドは処理中で、コマンドを発行したタスクはルーティング先のトランザクションでした。

#### UEPICTN

START コマンドが処理されていないか、または START コマンドは処理されたが、コマンドを発行したタスクがルーティング先のトランザクションではありませんでした。

#### UEPICFIS

START コマンドが機能シップされたものかどうかを示す標識が含まれた 1 バイトのアドレス。同等の値は以下のとおりです。

#### UEPICFY

START コマンドが処理され、START は機能シップされたものでした。

#### UEPICFN

START コマンドが処理されていないか、または START は処理されたが機能シップされたものではありませんでした。

#### UEPICTRN

実行するトランザクションの名前が含まれた 4 バイトのアドレス。

#### UEPICRTR

トランザクションを実行する端末の名前が含まれた 4 バイトのアドレス。

#### UEPICCTR

トランザクション・ルーティングされた START コマンドの場合は現行端末の名前が含まれており、機能シップされた START コマンドの場合はセッションの名前が含まれた、4 バイトのアドレス。

その他の START コマンドの場合、指示先のフィールドにはブランクが入ります。

#### **UEPICNTI**

機能シッパされた START コマンドの場合は、要求の発行元となった最後のシステムのネット名が含まれた 8 バイトのアドレス。

このシステム内でタスクにルーティングされるトランザクションによって発行された START コマンドの場合は、タスクのルーティング元となった最後のシステムのネット名。

その他の START コマンドの場合、指示先のフィールドにはブランクが入ります。

#### **UEPICSYI**

UEPICNTI にネット名が含まれている場合、対応する SYSID が含まれた 4 バイトのアドレス。

UEPICNTI にネット名が含まれていない場合、指示先のフィールドはブランクです。

#### **UEPICNTO**

UEPICNTI の内容が含まれた 8 バイトのアドレス。

これで「**UERCNETN**」の戻りコードが設定される場合、出口プログラムで、**ATI** 要求の送信先となるシステムのネット名をこのフィールドに置く必要があります。

#### **UEPICSYO**

UEPICSYI の内容が含まれた 4 バイトのアドレス。

これで「**UERCYSYI**」の戻りコードが設定される場合、出口プログラムで、**ATI** 要求の送信先となるシステムのシステム ID をこのフィールドに置く必要があります。

#### **UEPICNNI**

トランザクションを実行する端末のネット名を CICS が認識している場合は、そのネット名が含まれた 4 バイトの入力フィールドのアドレス。CICS がネット名を認識していない場合、アドレス指定されたフィールドにはブランクが入ります。

#### **UEPICNNO**

呼び出し時に、UEPICNNI の内容が含まれた 4 バイトの入出力フィールドのアドレス。出口プログラムはこのフィールドを使用して、トランザクションを実行する端末のネット名を提供することができます。ATI 要求の送信先となる TOR が z/OS Communications Server の汎用リソースのメンバーである場合、出口プログラムが端末のネット名を提供することは重要です。

### 戻りコード

#### **UERCTEUN**

端末が不明で、要求は拒否されます。

#### **UERCNETN**

端末は既知のもので、UEPICNTO にネット名が返されます。

#### **UERCYSYI**

端末は既知のもので、UEPICSYO にシステム ID が返されます。

#### **UERCPURG**

タスクは XPI 呼び出し中にパージされます。

### XPI 呼び出し

以下は使用しないでください。

- ADD\_SUSPEND
- DELETE\_SUSPEND
- DEQUEUE
- ENQUEUE
- RESUME
- SUSPEND
- WAIT\_MVS

## サンプル出口プログラム

DFHXTENF は、XALTENF 出口および XICTENF 出口に使用できるサンプル・プログラムです。DFHXTENF の詳細については、[端末未認識サンプル出口プログラム: DFHXTENF](#) を参照してください。

## トランザクション・マネージャー・ドメイン出口 XXMATT

出口 XXMATT は、トランザクション接続時に呼び出され、接続対象のトランザクションの一部の属性を変更することができます。

この出口は、UEPATPTI によってアドレス指定されたフィールドを変更することによって、トランザクションの接続トランザクション ID を変更できます。この出口から **EXEC CICS** コマンドを使用することはできません。

### 出口固有のパラメーター

#### UEPTRANID

トランザクション ID のアドレス (注を参照)。

#### UEPUSER

現行タスクがユーザー・タスクの場合に、トランザクションに関連付けられているユーザー ID のアドレス (注を参照)。

#### UEPTERM

トランザクションに関連付けられている端末 ID のアドレス (存在する場合)(注を参照)。

#### UEPPROG

このトランザクションのアプリケーション・プログラム名のアドレス (存在する場合)(注を参照)。

#### UEPATPTI

1 次トランザクション ID が含まれた 4 バイト・フィールドのアドレス。アドレス指定されたフィールドを変更することによって、1 次トランザクション ID を変更できます。

#### UEPATOTI

4 バイトの接続トランザクション ID のアドレス。トランザクション ID X'00000000' は、接続にトランザクション ID が指定されていないことを示します。

#### UEPATTPPL

接続 TPName の長さが含まれた領域のアドレス。長さゼロは、接続に TPName が指定されていないことを示します。

#### UEPATTPA

接続 TPName のアドレスが含まれたフルワードのアドレス。接続 TPName は、1 から 64 バイトの長さにすることができ、UEPTTPL によって定義されます。

#### UEPATLOC

トランザクションが見つかったかどうかを示す 1 バイト・フィールドのアドレス。トランザクションが見つからず、システム初期設定パラメーター DTRTRAN および DTRPGM が指定されている場合、DTRTRAN で指定されたトランザクションが接続され、CICS はトランザクションが検出されたものとみなします。

等価の値は次のとおりです。

#### UEATFND

トランザクションが見つかりました。

#### UEATNFND

トランザクションが見つかりませんでした。

#### UEPATST

1 バイトのトランザクション定義状態のアドレス。定義状態の等価の値は次のとおりです。

#### UEATENAB

トランザクションが使用可能になります。

#### UEATDISA

トランザクションは使用不可になります。

## UEPATTTK

トランザクション・トークンが含まれたダブルワードのアドレス。一部のトランザクション・マネージャー XPI 呼び出しでは、接続中のトランザクションを識別するためにこのトークンが必要なことに注意してください。

## 戻りコード

### UERCNORM

接続処理を続行します。

## XPI 呼び出し

このユーザー出口は、UEPATTTK トランザクション・トークンを XMIQ INQUIRE\_TRANSACTION XPI 呼び出しに対する入力として使用して、接続中のトランザクションについて照会することができます。

この出口では、XMIQ SET\_TRANSACTION XPI 呼び出しを使用して、合計優先順位および TCLASS も設定できます。

ほとんどの XPI 呼び出しを使用できますが、一般的にこの出口は TCP タスクの下で呼び出されるので、慎重に使用してください。したがって、TCP タスクの待機を引き起こす可能性のある XPI 呼び出しは発行しないようお勧めします。

## 注:

1. 以下の XPI 呼び出しは、トランザクションの接続を変更するために使用できる情報を取得する場合に役立ちます。
  - INQUIRE\_TRANSACTION
  - INQUIRE\_MXT
  - INQUIRE\_TCLASS
  - INQUIRE\_TRANDEF
  - INQUIRE\_SYSTEM
2. フィールド UEPTRANID、UEPUSER、UEPTERM、および UEPPROG は、多くのドメインのグローバル・ユーザー出口点に共通で、通常は、現行ユーザー・タスクに関連付けられた値を返します。ただし、XXMATT の場合は、出口が呼び出されるときに接続されるユーザー・タスクは現行タスクではありません。タスク接続が完了するまでは、現行タスクは接続を実行している CICS タスクです。

接続中のタスクが即時 START コマンド (つまり、間隔指定のない START) によって開始されるタスクの場合、現行タスクは START コマンドを発行したタスクであり、フィールドにはそのタスクに関連付けられた値が含まれています。

## 一時データ・プログラム出口 (XTDREQ、XTDIN、XTDOUT)

XTDREQ 出口は、要求分析の前に一時データ要求をインターセプトします。XTDOUT および XTDIN 出口はそれぞれ、QSAM または VSAM とのデータ交換の前、データ交換の後に呼び出されます。

CICS 一時データ機能はスレッド・セーフであるため、CICS はオープン TCB で一時データ要求を処理することができます。IPIC 接続経由でリモート領域への一時データ要求の機能シップを行う場合にも、一時データ要求はスレッド・セーフです。TCB の切り替えを最適化してオープン・トランザクション環境のパフォーマンスを高めるには、XTDREQ、XTDIN、および XTDOUT で実行されるプログラムをスレッド・セーフな規格にコーディングし、スレッド・セーフなものとして CICS に定義する必要があります。

## 出口 XTDREQ

出口 XTDREQ は、要求分析の前に呼び出されます。

### 出口固有のパラメーター

#### UEPTDQUE

4 バイトの TD キュー名のアドレス。

#### UEPTDTYP

1 バイトの TD 要求タイプのアドレス。値は次のとおりです。

**UEPTDPUT**

PUT 要求

**UEPTDGET**

GET 要求

**UEPTDPUR**

PURGE 要求

**戻りコード****UERCNORM**

TD 処理を続行します。

**UERCTDOK**

TD 処理を終了し、「NORMAL」を呼び出し元に返します。

**UERCTDNA**

TD 処理を終了し、「NOTAUTH」を呼び出し元に返します。

**UERCPURG**

タスクは XPI 呼び出し中にページされます。

**XPI 呼び出し**

次のものを使用することができます。

- INQ\_APPLICATION\_DATA
- INQUIRE\_SYSTEM
- WAIT\_MVS

その他の呼び出しを使用しないでください。

**出口 XTDIN**

出口 XTDIN は、CICS が QSAM (区画外の場合) または VSAM (区画内の場合) からデータを受け取った後で呼び出されます。

**出口固有のパラメーター****UEPTDQUE**

4 バイトの TD キュー名のアドレス。

**UEPTDAUD**

未変更の TD データのアドレス。

**UEPTDLUD**

未変更の TD データのフルワード長のアドレス。

**UEPTDAMD**

出口プログラムによって変更済みの TD データのアドレス。

**UEPTDLMD**

出口プログラムによって変更済みの TD データのフルワード長のアドレス。

**戻りコード****UERCNORM**

TD 処理を続行します。

**UERCPURG**

タスクは XPI 呼び出し中にページされます。

**XPI 呼び出し**

次のものを使用することができます。

- INQ\_APPLICATION\_DATA
- INQUIRE\_SYSTEM
- WAIT\_MVS

その他の呼び出しを使用しないでください。

## 出口 XTDOUT

出口 XTDOUT は、CICS が QSAM (区画外の場合) または VSAM (区画内の場合) ユーザー定義一時データ・キューにデータを渡す前に呼び出されます。

### 出口固有のパラメーター

#### UEPTDQUE

4 バイトの TD キュー名のアドレス。

#### UEPTDAUD

未変更の TD データのアドレス。

#### UEPTDLUD

未変更の TD データのフルワード長のアドレス。

#### UEPTDAMD

出口プログラムによって変更済みの TD データのアドレス。

#### UEPTDLMD

出口プログラムによって変更済みの TD データのフルワード長のアドレス。

#### UEPTDNUM

リスト内の項目の数が含まれたフルワードのアドレス。

#### UEPTDCUR

現在の項目の番号が含まれたフルワードのアドレス。

### 戻りコード

#### UERCNORM

TD 処理を続行します。

#### UERCTDOK

TD 処理を終了し、「NORMAL」を呼び出し元に返します。

注: UERCTDOK を返して複数行メッセージの先頭行を抑止した場合、メッセージの残りの XTDOUT に返されず、同様に抑止されます。

#### UERCPURG

タスクは XPI 呼び出し中にページされます。

### XPI 呼び出し

次のものを使用することができます。

- INQ\_APPLICATION\_DATA
- INQUIRE\_SYSTEM
- WAIT\_MVS

その他の呼び出しを使用しないでください。

## 一時データ EXEC インターフェース・プログラム出口 XTDEREQ および XTDEREQC

XTDEREQ 出口は、一時データ要求に対して一時データによるアクションが行われる前に、その要求をインターセプトします。XTDEREQC 出口は、一時データがその処理を完了した後で、一時データ要求をインターセプトします。

XTDEREQ 出口を変更して、以下のタスクを実行することができます。

- 要求を分析して、そのタイプ、指定するキーワード、および値を決定する。
- コマンドが実行される前に、要求により指定される任意の値を変更する。
- 以下のいずれかの命令を指定する戻りコードを設定する。
  - CICS は、行った変更をすべて使用して要求の処理を続ける必要がある。
  - CICS は要求を迂回する。この戻りコードを設定する場合、自分自身で要求を処理した場合と同様に、EXEC インターフェース・ブロック (EIB) の戻りコードもセットアップする必要があります。



XTDEREQC 出口を変更して、以下のタスクを行うことができます。

- 要求を分析して、そのタイプ、指定するキーワード、および値を決定する。
- EIB の戻りコードを設定する。

CICS 一時データ機能はスレッド・セーフであるため、CICS はオープン TCB で一時データ要求を処理することができます。IPIC 接続経由でリモート領域への一時データ要求の機能シップを行う場合にも、一時データ要求はスレッド・セーフです。TCB 切り替えを最適化し、オープン・トランザクション環境によるパフォーマンスの利益を得るためには、XTDEREQ および XTDEREQC で実行されるプログラムをスレッド・セーフ標準に従ってコーディングし、CICS にスレッド・セーフとして定義する必要があります。

両方の出口に、以下のように 8 つのパラメーターが渡されます。

- コマンド・レベルのパラメーター構造のアドレス。
- XTDEREQ から XTDEREQC に 4 バイトのデータを渡すために使用されるトークン (UEPTDTOK) のアドレス。
- EIB からの 4 つの戻りコードおよびリソース情報のコピーのアドレス。
- タスクの存続中に有効な 4 バイトのトークン (UEPTSTOK) のアドレス。
- 出口再帰カウント (UEPRECUR) のアドレス。

## プログラム例

CICS では、ソース・コード・ファイルとしてではなくソフトコピー・リストとしてのみ、プログラム例 `DFH$XTSE` を提供しています。このプログラム例は、EXEC インターフェース 出口に渡されるコマンド・レベルのパラメーター構造内のフィールドを変更する方法を示しています。

## 出口 XTDEREQ

出口 XTDEREQ は、CICS が一時データ API 要求を処理する前に呼び出されます。

### 出口固有のパラメーター

#### UEPCLPS

コマンド・レベルのパラメーター構造のアドレス。228 ページの『[UEPCLPS 出口固有パラメーター](#)』を参照してください。

#### UEPTDTOK

XTDEREQC に渡される 4 バイトのトークンのアドレス。UEPTDTOK を使用して、例えば、作業域を出口の XTDEREQC に渡すことができます。

#### UEPRCODE

EIB 戻りコード「EIBRCODE」の 6 バイトの 16 進数コピーのアドレス。EIB 戻りコードの詳細については、[EIB フィールド](#)を参照してください。

#### UEPRES

EIB 応答コード「EIBRESP」の 4 バイトのバイナリー・コピーのアドレス。

#### UEPRES2

EIB 応答コード「EIBRESP2」の 4 バイトのバイナリー・コピーのアドレス。

#### UEPTSTOK

タスクの存続中に有効な 4 バイトのトークンのアドレス。[タスク・トークン UEPTSTOK の使用](#)を参照してください。

#### UEPRECUR

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

#### UEPRSRCE

EIB リソース値の EIBRSRCE の 8 文字のコピーのアドレス。

### 戻りコード

#### UERCNORM

処理が続行される。

## UERCBYP

一時データ EXEC インターフェース・プログラムはこの要求を無視します。

## UERCPURG

タスクは XPI 呼び出し中にページされます。

## XPI 呼び出し

すべてを使用できます。

この出口では、XPI GETMAIN 呼び出しおよび FREEMAIN 呼び出しの使用が許可されていますが、代わりに EXEC CICS GETMAIN コマンドおよび FREEMAIN コマンドを使用することをお勧めします。

## API および SPI コマンド

次のコマンドを除き、すべてを使用できます。

EXEC CICS SHUTDOWN

EXEC CICS XCTL

**注：**再帰的コマンドを発行するときには注意してください。例えば、XTDEREQ 出口から一時データ要求を発行するときには、ループに入らないようにする必要があります。再帰カウンター UEPRECUR を使用することをお勧めします。

## 出口 XTDEREQC

出口 XTDEREQC は、一時データ API 要求が完了した後で、一時データ EXEC インターフェース・プログラムから戻る前に呼び出されます。

### 出口固有のパラメーター

#### UEPCLPS

コマンド・レベルのパラメーター構造のアドレス。228 ページの『UEPCLPS 出口固有パラメーター』を参照してください。

#### UEPTDTOK

XTDEREQC に渡される 4 バイトのトークンのアドレス。これにより、例えば、作業域を出口の XTDEREQC に渡すことができます。

#### UEPRCODE

EIB 戻りコード「EIBRCODE」の 6 バイトの 16 進数コピーのアドレス。EIB 戻りコードの詳細については、[EIB フィールド](#)を参照してください。

#### UEPRES

EIB 応答コード「EIBRESP」の 4 バイトのバイナリー・コピーのアドレス。

#### UEPRES2

EIB 応答コード「EIBRESP2」の 4 バイトのバイナリー・コピーのアドレス。

#### UEPTSTOK

タスクの存続中に有効な 4 バイトのトークンのアドレス。[タスク・トークン UEPTSTOK の使用](#)を参照してください。

#### UEPRECUR

ハーフワードの再帰カウンターのアドレス。このカウンターは、出口が最初に呼び出された際に 0 に設定され、再帰呼び出しごとにインクリメントされます。

#### UEPRSRCE

EIB リソース値の EIBRSRCE の 8 文字のコピーのアドレス。

#### UEP\_TD\_REMOTE\_SYSTEM

要求をリモート領域に送信する場合は、そのリモート領域の 4 バイトの名前が含まれた領域のアドレス。(リモート領域は、例えば、コマンドの SYSID オプション、機能シップ、または TDQUEUE 定義の REMOTESYSTEM オプションによって指定された可能性があります。)

要求をローカル領域で実行する場合、このパラメーターはブランクが含まれた 4 バイト領域のアドレスです。

#### UEP\_TD\_REMOTE\_NAME

要求をリモート領域に送信する場合は、そのリモート領域がキューを認識する 4 文字の名前が含まれた領域のアドレス。

## 戻りコード

### UERCNORM

処理が続行される。

### UERCPURG

タスクは XPI 呼び出し中にページされます。

### XPI 呼び出し

すべてを使用できます。

この出口では、XPI GETMAIN 呼び出しおよび FREEMAIN 呼び出しの使用が許可されていますが、代わりに EXEC CICS GETMAIN コマンドおよび FREEMAIN コマンドを使用することをお勧めします。

### API および SPI コマンド

次のコマンドを除き、すべてを使用できます。

EXEC CICS SHUTDOWN

EXEC CICS XCTL

注：再帰的コマンドを発行するときには注意してください。例えば、XTDEREQC 出口から一時データ要求を発行するときには、ループに入らないようにする必要があります。再帰カウンター UEPRECUR を使用することをお勧めします。

## コマンド・レベルのパラメーター構造

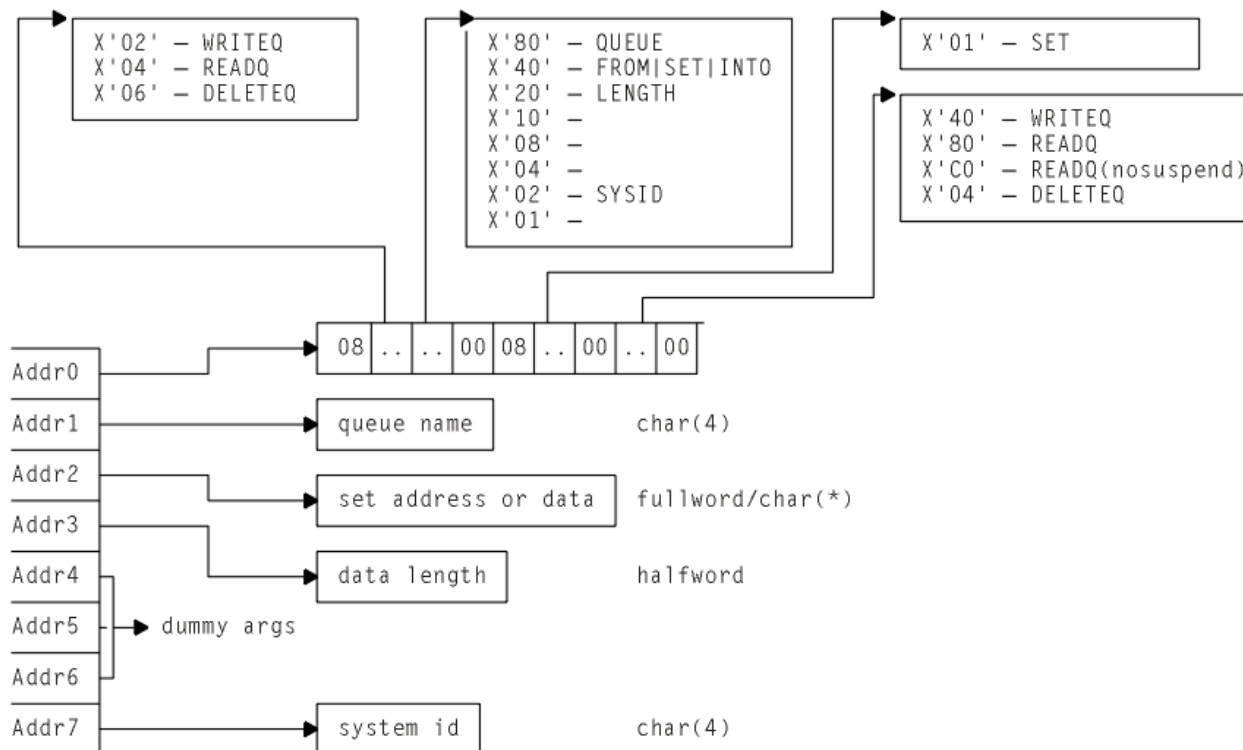


図 8. 一時データ用のコマンド・レベルのパラメーター構造

コマンド・レベルのパラメーター構造は、一連のアドレスで構成されます。最初のアドレスは、EXEC インターフェース記述子 (EID) を指します。これは、要求のタイプを記述する 8 バイト領域からなり、要求で指定された各キーワードを識別します。残りのアドレスは、その要求に関連付けられたデータ部分を指します。(例えば、2 番目のアドレスはキュー名を指します。)

EID を調べて、要求のタイプおよび指定されたキーワードを判別することができます。リスト内のその他のパラメーターを調べて、キーワードの値を判別することができます。要求で指定されたキーワードの値を変更することもできます。例えば、要求で指定された sysid を変更できます。

## パラメーター・リスト終了標識

パラメーター・リスト内の最後のアドレス・セットの高位ビットが、リスト内の最後のアドレスであることを示すために設定されます。ユーザー出口プログラムから戻ったときに、CICS は、パラメーター・リストをスキャンして高位ビットを探し、最終パラメーターを検出します。したがって、パラメーター・リストの長さを変更した場合は、高位ビットをリセットしてどれが新しい最終アドレスかを示す必要もあります。

例えば、パラメーター・リストで最初の 2 つのアドレスのみ (EID のアドレスである TD\_ADDR0 と、DELETEQ 要求に指定されたキューの名前のアドレスである TD\_ADDR1) を指定した場合、TD\_ADDR1 内の高位ビットが設定されます。TD\_ADDR7 に SYSID のアドレスを指定してパラメーター・リストを拡張した場合、TD\_ADDR1 の高位ビットをリセットし、代わりに TD\_ADDR7 の高位ビットをオンに設定する必要があります。

パラメーター・リストの最大サイズを出口に対して指定します。これによって、出口プログラムは、事前に追加のストレージを取得せずに、まだ指定されていないパラメーターを追加することができます。

XTDEREQC の完了後に、XTDEREQ が呼び出される前の元のパラメーター・リストが復元されます。その後に、実行診断機能 (EDF) により、実行前および実行後の元のコマンドが表示されます。**EDF は、出口によって行われた変更は表示しません。**

## UEPCLPS 出口固有パラメーター

UEPCLPS 出口固有パラメーターは、出口 XTDEREQ と出口 XTDEREQC の両方に組み込まれています。このパラメーターには、コマンド・レベルのパラメーター構造のアドレスが入ります。

コマンド・レベルのパラメーター構造には、TD\_ADDR0 から TD\_ADDR7 までの 8 つのアドレスが含まれています。この構造は DSECT TD\_ADDR\_LIST で定義されます。DSECT TD\_ADDR\_LIST は、COPY DFHTDUED ステートメントを組み込むことによって出口プログラムにコピーする必要があります。

コマンド・レベルのパラメーター・リストは、以下のように構成されています。

### TD\_ADDR0

EID と呼ばれる 8 バイトの領域のアドレスであり、以下のように構成されています。

- TD\_GROUP
- TD\_FUNCT
- TD\_BITS1
- TD\_BITS2
- TD\_EIDOPT5
- TD\_EIDOPT6
- TD\_EIDOPT7

### TD\_GROUP

常に X'08' で、これが一時データ要求であることを示します。

### TD\_FUNCT

以下の要求のタイプを定義する 1 バイトの領域です。

X'02'

WRITEQ

X'04'

READQ

X'06'

DELETEQ。

### TD\_BITS1

どの引数が指定されたかを定義する存在ビット。キーワードに関連付けられた引数を取得するには、コマンド・レベルのパラメーター構造から適切なアドレスを使用する必要があります。このアドレスを使用する前に、関連付けられた存在ビットを確認する必要があります。存在ビットがオフに設定されている場合は、引数が要求に指定されていないため、このアドレスを使用してはなりません。

**X'80'**

要求に QUEUE キーワードの引数が含まれている場合に設定されます。設定された場合は、**TD\_ADDR1** が意味を持ちます。

**X'40'**

要求に INTO、SET、または FROM のいずれかのキーワードの引数が含まれている場合に設定されます。設定された場合は、**TD\_ADDR2** が意味を持ちます。

**X'20'**

要求に LENGTH キーワードの引数が含まれている場合に設定されます。設定された場合は、**TD\_ADDR3** が意味を持ちます。

**X'02'**

要求に SYSID キーワードの引数が含まれている場合に設定されます。設定された場合は、**TD\_ADDR7** が意味を持ちます。

**TD\_BITS2**

一時データによって使用されない 2 バイト。

**TD\_EIDOPT5**

特定のキーワードが要求に指定されたかどうかを示します。

**X'01'**

SET が指定されています (INTO の指定はなし)。

**TD\_EIDOPT6**

一時データによって使用されない 1 バイト。

**TD\_EIDOPT7**

特定の関数、キーワードまたはその両方が要求上に指定されたかどうかを示します。

**X'40'**

WRITEQ が指定されています。

**X'80'**

READQ が指定されています。

**X'C0'**

READQ(nosuspend) が指定されています。

**X'04'**

DELETEQ が指定されています。

**TD\_ADDR1**

QUEUE からの名前が含まれた 4 バイト領域のアドレスです。

**TD\_ADDR2**

次のアドレスのいずれかです。

- SET からの 4 バイトのアドレス (要求が READQ であって、かつ **TD\_EIDOPT5** がこれが SET であることを示す場合)。
- INTO からのデータ (要求が READQ であって、かつ **TD\_EIDOPT5** がこれが SET でないことを示している場合)。このビットをユーザー出口で変更することはできません。
- FROM からのデータ (要求が WRITEQ の場合)。

**TD\_ADDR3**

次のアドレスのいずれかです。

- LENGTH のハーフワード値 (要求が READQ または WRITEQ の場合)。警告: INTO を指定する要求の場合は、LENGTH の値をアプリケーションで指定された値よりも大きい値に変更しないでください。この値を大きい値に変更すると、アプリケーション内でストレージ・オーバーレイが発生します。

**TD\_ADDR4**

CICS の内部使用専用の値のアドレスです。このアドレスは使用できません。

**TD\_ADDR5**

CICS の内部使用専用の値のアドレスです。このアドレスは使用できません。

## TD\_ADDR6

CICS の内部使用専用の値のアドレスです。このアドレスは使用できません。

## TD\_ADDR7

SYSID の値が含まれている領域のアドレスです。

## TD\_ADDR8

CICS の内部使用専用の値のアドレスです。このアドレスは使用できません。

## コマンド・レベルのパラメーター構造のフィールドの変更

一時データに渡される一部のフィールドは要求への入力として使用され、一部は出力フィールドとして使用され、さらに一部は入力および出力の両方に使用されます。ユーザー出口プログラムでフィールドを変更するために使用する方法は、フィールドの使用方法に応じて異なります。

以下は、常に入力フィールドです。

- QUEUE
- FROM
- SYSID

以下は、常に出力フィールドです。

- INTO
- SET

LENGTH は、WRITEQ 要求では入力フィールドで、SET が指定された READQ 要求では出力フィールドです。INTO が指定された READQ 要求では、入力フィールドと出力フィールドの両方になります。

## 入力フィールドの変更

入力フィールドを変更する場合の正しい方法は、そのフィールドの新規コピーを作成し、コマンド・レベル・パラメーター・リスト内のアドレスを変更して新規データを指すようにすることです。

**注：**コマンド・レベルのパラメーター・リストによって指定されたデータを変更することによって、入力フィールドを変更してはなりません。これを行うと、アプリケーション・プログラムに属しているストレージが破壊され、プログラムがフィールドを再使用しようとしたときに障害が発生することになります。

## 出力フィールドの変更

230 ページの『[入力フィールドの変更](#)』で説明されている手法は、出力フィールドの変更には適していません。(結果が、アプリケーションの領域ではなく新しい領域に返されるため、アプリケーションで表示できません。)

出力フィールドの変更は、コマンド・レベル・パラメーター・リストが指すデータを変更することによって行います。出力フィールドの場合は、アプリケーションのデータを所定の位置で変更することができます。これは、アプリケーションが、そのフィールドが変更されることを予期しているためです。

## 入力と出力の両方に使用されるフィールドの変更

入力と出力の両方に使用されるフィールドの例として、INTO を指定する READQ 要求の LENGTH があります。この種のフィールドは、出力フィールドと同様の方法で取り扱うことができ、同じものと見なされます。

## EID の変更

EID を変更することで要求に対して大きな変更 (READQ 要求から WRITEQ 要求への変更など) を行うことはできません。ただし、SYSID の存在ビットをオンにして要求をリモート・システムに送信できるようにするなど、要求に対する小さな変更を行うことはできます。

以下のリストは、EID 内の変更可能なビットを示しています。EID のその他の部分を変更しようとしても、すべて無視されます。

## TD\_BITS1

X'20'

LENGTH の存在ビット。

**X'02'**

SYSID の存在ビット。

#### **TD\_EIDOPT5**

**X'01'**

SET キーワードの存在ビット。このビットをユーザー出口から変更することはできません。

#### **TD\_EIDOPT7**

TD\_EIDOPT7 に対する変更は、READQ 要求に制限されています。X'80'-READQ は X'C0'-READQ(nosuspend) と交換可能です。このバイトにその他の変更を行うことはできません。

EID は、アプリケーション・プログラムに戻る前に、元の値にリセットされます。つまり、EID に対して行われた変更は、一時データ要求の期間中のみ保持されます。

注：ユーザー出口プログラムは、EID に大きな変更を加えることはできません。ただし、許容されている小さい変更を加える場合でも、十分に注意する必要があります。

#### **EIB**

EIBRSRCE、EIBRCODE、EIBRESP、および EIBRESP2 が出口に渡されるので、以下のことができます。

- XTDEREQ および XTDEREQC 内で、完了情報およびリソース情報を変更または設定する。
- XTDEREQC 内の完了情報およびリソース情報を調べる。

パラメーター・リストで指定されている EIBRSRCE、EIBRCODE、EIBRESP、および EIBRESP2 のコピーを更新できます。XTDEREQC の完了後に、または XTDEREQ に戻りコード「bypass」を指定してある場合に、一時データが値を実 EIB にコピーします。

有効な一時データ応答を設定する必要があります。EIBRCODE、EIBRESP、および EIBRESP2 の 3 つすべてを、整合性のある値セット (有効な完了を記述するために CICS 一時データによって設定されるような値セット) に設定する必要があります。**CICS では、EIBRCODE、EIBRESP、および EIBRESP2 の整合性は監視されません。**ただし、EIBRCODE がゼロ以外の値に設定され、EIBRESP がゼロに設定されると、CICS はゼロ以外の値で EIBRESP をオーバーライドします。EIBRCODE、EIBRESP、および EIBRESP2 の値の設定に役立つように、一時データで使用する値が DFHTDUEd に指定されます。

## **ユーザー・ログ・レコード・リカバリー・プログラム出口 XRCINIT および XRCINPT**

ウォーム・リスタートおよび緊急時再始動の際に、リカバリー可能 CICS リソースに対して行われ、システム終了時にコミットされなかった更新内容はバックアウトする必要があります。XRCINIT と XRCINPT がユーザー・ログ・レコード・リカバリー・プログラムから呼び出されます。これは、必要に応じて、ユーザー作成のシステム・ログ・エントリーをバックアウトするために使用されます。

XRCINIT は、次のように、ウォーム・リスタートおよび緊急時再始動時に呼び出されます。

- 最初のユーザー・リカバリー・レコードが XRCINPT に送信される前
- 該当するすべてのレコードが XRCINPT に送信されたとき

XRCINPT は、システム・ログからユーザー・ログ・レコードが読み取られたときに呼び出されます。

XRCINPT を使用して、特定のユーザー・ジャーナル・レコードの緊急時再始動の際に CICS が取るデフォルトのアクションを変更できます。XRCINPT に渡されるレコードは、UOW 内で以下に該当するものです。

- 最後の完全なアクティビティー・キーポイントに出現した
- CICS の終了時に未完了であった
- 最後の完全なアクティビティー・キーポイントの開始後に、コミットされたか、バックアウトされたか、未確定になった。(ただし、これは WRITE JOURNALNAME(DFHLOG) 要求内で指定された JTYPEID の左端ビットが 1 であったレコードのみに適用されます。)

アクティビティー・キーポイント出口 XAKUSER によって書き込まれたレコードは、最後の完全なアクティビティー・キーポイントに出現する場合のみ渡されます。これらは、他のすべてのレコードより後に渡さ



れます。このため、逆のログ・ストリーム・シーケンスでの順序とは、レコードの提示順序が異なる場合があります。

出口に渡されるレコードのフォーマットは、以下のとおりです。

#### オフセット

##### フィールドの内容

0	JTYPEID
2	予約
4	接頭部データ (L) の長さ。(接頭部がない場合はゼロ)
8	接頭部データ (ある場合)
8 + L	ログ・データ

このレコードは、コピーブック DFHLGGFD の DSECT CL\_USER\_HEADER によってマップされます。

XRCINIT および XRCINPT を使用する際には、一時記憶域と一時データ・リソースのリカバリーが完了する前に、これらの出口が呼び出される場合があることに注意する必要があります。

## 出口プログラムのコーディング

出口プログラム内で CICS サービスを使用できます。これらのサービスは、XPI コマンドまたは EXEC CICS コマンドを使用して、これらの出口から呼び出されます。

以下のことを考慮する必要があります。

- 初期設定中の早い段階で XPI を使用する場合には、制限があります。PLTPI の第 2 フェーズまでは、XPI 関数 TRANSACTION\_DUMP、WRITE\_JOURNAL\_DATA、MONITOR、および INQUIRE\_MONITOR\_DATA を使用する出口プログラムを呼び出さないでください。
- これらの出口での EXEC CICS コマンドの使用についても、以下の制限があります。

- EXEC CICS コマンドを使用して、端末制御サービスにアクセスすることはできません。
- 一時記憶域、一時データ、ファイル制御、ジャーナル制御、または DL/I サービスは使用しないことを強くお勧めします。これは、アクセスを試みるリソースもリカバリー状態になるために「活動を開始していない」可能性があるからです。このような状況にあるリソースにアクセスしようとすると、最良の場合でもリカバリー・タスクの逐次化が発生し、最悪の場合はデッドロックが生じます。

これらの出口から呼び出されるプログラム内でファイル制御要求を発行する場合は、以下のことに注意してください。

- 出口プログラムがファイル制御要求の結果として領域を獲得する場合、その領域の解放はそのプログラムが行う必要があります。
- 出口プログラムは、ストリング番号 1 の VSAM データ・セットを参照するファイルへのファイル制御要求を試行してはなりません (出口の初期設定中に、そのファイルに対するアクションが指定されていない場合を除いて)。
- リカバリーが EXEC CICS SYNCPOINT ROLLBACK 要求の結果として行われる場合、出口プログラムは EXEC CICS コマンドを発行してはなりません。
- EXEC CICS コマンドを発行する出口プログラムでは、最初に EIB をアドレス指定する必要があります。CICS サービスの使用を参照してください。
- EXEC CICS コマンドを発行する出口プログラム、および DFHEIENT マクロを使用する出口プログラムでは、DFHEIRET マクロを使用して戻りコードを設定し、CICS に戻る必要があります。[ストレージの使用法を示すグローバル・ユーザー出口 XPI](#) の例を参照してください。
- これらの出口から呼び出される出口プログラムは、NOEDF オプションを使用して変換する必要があります (EXEC CICS コマンドを発行する場合)。[EDF とグローバル・ユーザー出口](#)を参照してください。



- 出口プログラムで獲得されたタスク・チェーン・ストレージは、緊急時再始動処理の完了時に解放されます。ただし、出口プログラムは、ストレージの内容が不要になった直後に、ストレージの解放を試みる必要があります。
- 出口プログラムは、ファイル制御バックアウト・プログラムによって設定された不在標識またはアクションなし標識をリセットしてはなりません。
- 再帰的コマンドを発行する際には、ループが生じないように注意してください。例えば、これらの出口から RC 要求を発行する際に、ループに入らないようにユーザーが注意する必要があります。

## 出口プログラムの使用可能化

これらの出口を使用可能にするには、以下のどちらかを行う必要があります。

- システム初期設定パラメーター TBEXITS=(name1,name2,name3,name4,name5,name6) を指定する。ここで、name1 から name6 は、XRCINIT、XRCINPT、XFCBFAIL、XFCLDEL、XFCBOVER、および XFCBOUT のユーザー出口プログラムの名前です。
- PLTPI プログラムを使用した初期設定の最初の段階で、出口を使用可能にする。

TBEXITS パラメーターを使用して出口を使用可能にすると、4 バイトのグローバル作業域が提供されます。PLTPI プログラムを使用する場合は、グローバル作業域のサイズを選択できます。また、それぞれの出口点で複数の出口プログラムを使用可能にすることもできます。TBEXITS パラメーターは、それぞれの出口点でただ 1 つの出口プログラムを使用可能にすることができます。PLTPI 処理については、[Writing initialization and shutdown programs](#) に説明があります。

## 出口 XRCINIT

### 呼び出される状況

ウォーム・リスタートおよび緊急時再始動の際に、以下の時点。

- 最初のユーザー・リカバリー・レコードが XRCINPT に送信される前
- 該当するすべてのレコードが XRCINPT に送信されたとき

### 出口固有のパラメーター

#### UEPTREQ

呼び出しの理由を示す 1 バイト・フラグのアドレス。UEPTREQ の値が UEUSINIT である場合、出口はユーザー・リカバリーの開始時に呼び出されており、UEPTREQ の値が UEUSTERM である場合、出口はユーザー・リカバリーの終了時に呼び出されています。

#### UEPRSTRT

CICS がどのように再始動されたかを示す 1 バイト・フラグのアドレス。

#### UEPRWARM

ウォーム・スタート

#### UEPREMER

緊急時始動

### 戻りコード

#### UERCNORM

処理を続行します。これ以外の戻りコードはサポートされていません。

### XPI 呼び出し

すべてを使用できます。制約事項については、『[231 ページの『ユーザー・ログ・レコード・リカバリー・プログラム出口 XRCINIT および XRCINPT』](#)』を参照してください。

## 出口 XRCINPT

### 呼び出される状況

ウォーム・リスタート時、および緊急リスタート時に、システム・ログで検出されたユーザー・ログ・レコードごとに 1 回ずつ。

## 出口固有のパラメーター

### UEPUOWST

作業単位の後処理を示す 1 バイト・フラグのアドレス。可能な値は次のとおりです。

### UEPUOWAK

活動キーポイント・レコード

### UEPUOWCM

コミットされた作業単位

### UEPUOWBO

バックアウトされた作業単位

### UEPUOWIF

未完了の作業単位

### UEPUOWID

未確定の作業単位

### UEPLGREC

読み取られたばかりのログ・レコードのアドレス。ジャーナル管理レコードをマップするには、『[CICS ロギングおよびジャーナリング](#)』に記載されている情報を使用します。

### UEPLGLEN

ログ・レコードの長さを含むフルワードのアドレス。

### UEPTAID

タスク ID を含む 4 バイト・フィールドのアドレス。

### UEPTRID

トランザクション ID を含む 4 バイト・フィールドのアドレス。

### UEPTEID

端末 ID を含む 4 バイト・フィールドのアドレス。

注：UEPTAID、UEPTRID、および UEPTEID でアドレス指定されたフィールドの値は、活動キーポイント・レコードの場合 (つまり、UEPUOWST でアドレス指定されたフィールドに UEPUOWAK が含まれている場合) は作用しません。

## 戻りコード

### UERCNORM

処理が続行される。

### UERCBYP

このレコードをバイパスします。

## XPI 呼び出し

すべてを使用できます。制約事項については、『[231 ページの『ユーザー・ログ・レコード・リカバリ・プログラム出口 XRCINIT および XRCINPT』](#)』を参照してください。

## SNA LU 管理プログラム出口 (XZCATT)

この出口は、LU 端末タスク用のタスク接続の前に呼び出されます。

### 呼び出される状況

端末タスク用のタスク接続の前。

## 出口固有のパラメーター

### UEPTCTTE

端末制御テーブル・エントリー (TCTTE) のアドレス。DSECT DFHTCTTE を使用して TCTTE をマップすることができます。

### UEPTIOA

端末入出力域 (TIOA) のアドレス。DSECT DFHTIOA を使用して TIOA をマップすることができます。ただし、フィールド TIOASAL および TIOASCA はプログラミング・インターフェースでは **ありません**。

**UEPTPN**

APPC トランザクション・プロセス名 (TPN) または LU6.1 プロセス名 (DPN) のアドレス。長さはパラメーター UEPTPNL によってアドレス指定されます。

**UEPTPNL**

TPN または DPN の長さが含まれた 4 バイト・フィールドのアドレス。

**UEPTRAN**

4 バイトのトランザクション ID のアドレス。

注：出口プログラムで、自動トランザクション開始 (ATI) によって開始されたタスクの TRANSID を変更しないでください。(CICS では、プログラム制御テーブルでこの TRANSID を AOR で作成された自動開始記述子 (AID) 内の TRANSID と突き合わせる必要があるためです。)

**戻りコード****UERCNORM**

処理が続行される。

**XPI 呼び出し**

すべて使用できます。

**サンプル出口プログラム**

DFH\$ZCAT

## **SNA 実効ページ・セット・モジュール出口 (XZCIN、XZCOUT、XZCOUT1、および XZIQUE)**

これらの出口は、入出力イベントの後、またはメッセージが要求単位 (RU) に分解される前に呼び出されます。

### **出口 XZCIN**

この出口は、入力イベントの後に呼び出されます。

**呼び出される状況**

入力イベントの後。

**出口固有のパラメーター****UEPTCTTE**

端末制御テーブル・エントリー (TCTTE) のアドレス。DSECT DFHTCTTE を使用して TCTTE をマップすることができます。

**UEPTIOA**

端末入出力域 (TIOA) のアドレス。出口プログラムでアドレスを変更しないでください。DSECT DFHTIOA を使用して TIOA をマップすることができます。ただし、フィールド TIOASAL および TIOASCA はプログラミング・インターフェースではありません。

**戻りコード****UERCNORM**

処理が続行される。

**XPI 呼び出し**

すべてを使用できます。ただし、置換 TIOA として使用するための端末クラス・ストレージを取得するために、GETMAIN 呼び出しを使用しないでください。なぜなら、TIOA への複数の内部ポインターがあり、そのいずれかが更新されなかった場合にはアプリケーションに問題が発生する可能性があるからです。

**API コマンドおよび SPI コマンド**

**EXEC CICS** コマンドは使用できません。

## 出口 XZCOUT

この出口は、出力イベントの前に呼び出されます。

### 呼び出される状況

出力イベントの前。

### 出口固有のパラメーター

#### UEPTCTTE

端末制御テーブル・エントリー (TCTTE) のアドレス。DSECT DFHTCTTE を使用して TCTTE をマップすることができます。

#### UEPTIOA

端末入出力域 (TIOA) のアドレス。出口プログラムでアドレスを変更しないでください。DSECT DFHTIOA を使用して TIOA をマップすることができます。ただし、フィールド TIOASAL および TIOASCA はプログラミング・インターフェースでは **ありません**。

注: ある種の状況では (例えば NULL RU の送信前に XZCOUT が呼び出された場合)、UEPTIOA にはゼロが入っています。

### 戻りコード

#### UERCNORM

処理が続行される。

### XPI 呼び出し

すべてを使用できます。ただし、GETMAIN 呼び出しを使用して、置換 TIOA として使用するための端末クラス・ストレージを取得することはお勧めできません。なぜなら、TIOA への複数の内部ポインターがあり、そのいずれかが更新されなかった場合にはアプリケーションに問題が発生する可能性があるからです。

## 出口 XZCOUT1

この出口は、メッセージが RU に分解される前に呼び出されます。

### 呼び出される状況

メッセージが RU に割り込む前。

### 出口固有のパラメーター

#### UEPTCTTE

端末制御テーブル・エントリー (TCTTE) のアドレス。DSECT DFHTCTTE を使用して TCTTE をマップすることができます。

#### UEPTIOA

端末入出力域 (TIOA) のアドレス。出口プログラムでアドレスを変更しないでください。DSECT DFHTIOA を使用して TIOA をマップすることができます。ただし、フィールド TIOASAL および TIOASCA はプログラミング・インターフェースでは **ありません**。

### 戻りコード

#### UERCNORM

処理が続行される。

### XPI 呼び出し

すべてを使用できます。ただし、GETMAIN 呼び出しを使用して、置換 TIOA として使用するための端末クラス・ストレージを取得することはお勧めできません。なぜなら、TIOA への複数の内部ポインターがあり、そのいずれかが更新されなかった場合にはアプリケーションに問題が発生する可能性があるからです。

## MRO および APPC システム間キューの管理用 XZIQUE 出口

XZIQUE 出口を使用すると、MRO および APPC 接続上のセッションについて、キューに入れられた要求の数を制御できます。

### 注:

- セッションについてキューに入れられた要求は、「割り振りキュー」と呼ばれます。

- IP 相互接続 (IPIC) 接続上のセッションについてキューに入れられた要求の数を制御するための、それに相当するグローバル・ユーザー出口は XISQUE です。243 ページの『IPIC システム間キュー管理用の XISQUE 出口』を参照してください。
- システム間キューの長さの制御には、いくつかの方法が使用できます。各種の方法の説明については、以下を参照してください。システム間のセッション・キューの管理。

XZIQUE 出口を使用すると、キューイングの問題点 (ボトルネック) を早期発見できます。これは、XISCONA グローバル・ユーザー出口 (121 ページの『システム間連絡プログラム出口 XISCONA、XISLCLQ、および XISQLCL』の説明を参照) によって提供される機能を拡張します。XISCONA は、機能シップおよび DPL 要求の場合にのみ呼び出されます。XZIQUE は機能シップと DPL に加えて、トランザクション・ルーティング、非同期処理、および分散トランザクション処理の要求の場合に呼び出されます。XISCONA と比べ、さらに詳細な情報を受け取って、それに基づいて決定します。

XZIQUE を使用すると、キューの長さに応じて割り振り要求をキューに入れるかまたは拒否することが可能になります。また、ボトルネックの存在する接続を終了してから再確立することも可能です。

### **XISCONA 出口との対話**

XZIQUE と XISCONA のグローバル・ユーザー出口の間で対話することはありません。両方の出口を使用可能にした場合、機能シップと DPL 要求に対して XISCONA と XZIQUE を両方とも呼び出すことができますが、これは推奨されません。

このため、これらの出口のうち 1 つのみが使用可能になるようにしてください。提供する機能が増えて柔軟性が強化されているので、XISCONA ではなく XZIQUE を使用することをお勧めします。

XISCONA グローバル・ユーザー出口プログラムがすでにある場合には、おそらくこの出口を修正すれば、XZIQUE 出口点で 사용할 ことができます。

### **XZIQUE 出口が呼び出される時点**

XZIQUE グローバル・ユーザー出口は、使用可能になっていれば、次の時点で呼び出されます。

- CICS がリモート・システムとのセッションの獲得を試行したときに、使用可能な空きセッションがない場合。この出口は、CONNECTION 定義に QUEUELIMIT オプションが指定されているかどうかや、その限度を超えているかどうかにかかわらず、呼び出されます。割り振り要求に NOQUEUE か NOSUSPEND が指定されている場合には、呼び出されません。

セッションに対する要求は、DTP プログラムによって出された明示的な EXEC CICS ALLOCATE コマンドや、トランザクション・ルーティングや機能シップの要求など、いくつかの方法で行われます。

- 接続のキューが出口プログラムの前の呼び出しによって除去されたあとに、割り振り要求が空きセッションを見つかることができた場合。この場合には、CICS が通常どおり処理を続け、必要ならキューイングを再開するように、出口プログラムで指定できます。

### **XZIQUE グローバル・ユーザー出口プログラムの使用**

この出口が使用可能になっていれば、XZIQUE グローバル・ユーザー出口プログラムによって、ローカル・システムの特定の接続に対する割り振りキューの状態を検査することができます。

この情報はパラメーター・リストで出口プログラムに渡されます。このリストは、不特定の割り振り要求のデータ用か、特定のモードグループのデータ用に構造化されます (どちらになるかは、そのセッション要求による)。不特定の割り振り要求は、モードグループを指定しない MRO、LU6.1、APPC の各セッションに対するものです。

パラメーター・リストで渡される情報を使用して、グローバル・ユーザー出口プログラムは、CICS が割り振り要求をキューに入れるかどうかを決定できます (例えば、キューの長さに基づいて)。プログラムは、CICS が提供する戻りコードの 1 つを使用して、決定を CICS に通知します。それらは以下のとおりです。

### **UERCAQUE**

この戻りコードは、CICS で割り振り要求をキューに入れることを示します。

接続に対してキューに入れられる割り振り要求の総数は、システム・エントリー統計のフィールド A14ESTAQ (すべての非特定割り振りの場合) またはモード・エントリー統計の A20ESTAQ (特定モードグループ割り振りの場合) で指定します。詳しくは、DFHA14DS または DFHA20DS の DSECT を参照してください。CICS は、出口固有パラメーター UEPQUELIM に入れて、接続定義からの QUEUELIMIT を出口プログラムに渡します。

限度に達していなければ、戻りコード UERCAQUE を使用して制御を CICS に返すことができます。

## UERCAPUR

この戻りコードは、CICS が割り振り要求を拒否し、SYSIDERR をアプリケーション・プログラムに返すことを示します。既存のキューは変更されません。

キューに入れられた割り振り要求の数が接続の QUEUELIMIT パラメーターに設定された限度に達した場合は、CICS に要求拒否を要求することができます。ただし、まずリンクの状態が正常であるかどうかを確認する必要があります。つまり、セッションの割り振り速度が許容可能なかどうかを確認することです。キューの開始時刻、現在時刻、およびキューの開始以降処理された割り振りの総数を使用して、CICS が要求を処理する速度を判別します。関連フィールドは、非特定割り振り要求の場合は UEPSAQTS および UEPSACNT、特定モードグループ要求の場合は UEPMAQTS および UEPMACNT です。

CICS がこの接続上のセッションに対する要求を許容速度で割り振っているかどうかを判別するには、計算した時間を次のいずれかと比較することができます。

1. 出口固有のパラメーター UEPEMXQT で渡される、接続定義からのパラメーター MAXQTIME。
2. その他の事前設定された時間値。

この種の公式を使用して得た処理時間が許容可能であれば、戻りコード UERCAPUR と共に制御を CICS に返し、この要求のみをページします。

## UERCAKLL または UERCAKLM

これらの戻りコードは、CICS で以下のように要求を処理することを指示します。

- UERCAKLL – この要求を拒否し、この接続上でキューに入れられた他のすべての割り振り要求をページし、情報メッセージをオペレーター・コンソールに送ります。
- UERCAKLM – この要求を拒否し、この接続上でキューに入れられた他のすべてのモードグループ割り振り要求をページし、情報メッセージをオペレーター・コンソールに送ります。

キュー限度に達したが、キューに対する割り振り処理のパフォーマンスがユーザー出口プログラムで定義された許容可能限度を下回っている場合は、以下のように CICS に制御を返すことができます。

- 非特定割り振り要求の場合は、戻りコード UERCAKLL を使用してください。UERCAKLL は、ページされた割り振り要求上で待機しているすべてのアプリケーション・プログラムにも SYSIDERR を返します。CICS は、XZIQUE 出口プログラムに対する後続の呼び出しで UEPFLAG パラメーターを UEPRC8 に設定して、キューのページの前に UERCAKLL が返されたことを示します。
- 特定モードグループ割り振り要求の場合は、戻りコード UERCAKLM を使用してください。UERCAKLM は、ページされた割り振り要求上で待機しているすべてのアプリケーション・プログラムにも SYSIDERR を返します。CICS は、XZIQUE 出口プログラムに対する後続の呼び出しで UEPFLAG パラメーターを UEPRC12 に設定して、キューのページの前に UERCAKLM が返されたことを示します。

タスクのフローを混雑させているキューをページすることによって、システムの停滞を防ぐために必要なタスク・スロットが解放されます。セッション・キューの拡大を可能にする限度が大きいほど、MAXT パラメーターによって設定されたタスクの上限に達する可能性が高くなり、その後は接続できないローカル領域内で着信タスクのキューが発生します。一部の内部 CICS 要求 (LU サービス・モデル・トランザクション CLS1、CLS2、および CLS3 に対する要求など) は、戻りコード UERCAKLL および UERCAKLM ではページされないことに注意してください。

キューが (UERCAKLL または UERCAKLM によって) 事前にページされたが、現在キューに入っている要求がない場合には、キューが最後にページされてからの成功した割り振りの数を確認してください。この数は、非特定割り振り要求の場合は UEPSARC8 に、特定モードグループ要求の場合は UEPMAR12 に入っています。キューが最後にページされてからこの接続上で割り振られたこのタイプの要求がない場合、前のページの原因となった問題は解決されていないため、UERCAPUR を使用してこの要求を拒否する必要があります。

割り振られた UEPSARC8 パラメーターまたは UEPMAR12 パラメーターが処理中の場合、UERCAQUE を使用して要求のキューイングを再開する必要があります。このケースで UERCAQUE を返すと、CICS は、コンソールに対して情報メッセージを発行してキューイングが再開されたことを通知します。



注：非特定割り振り要求の場合も特定モードグループ割り振り要求の場合も、システム・エントリー統計レコード UEPCONST のアドレスが提供されます。

非特定割り振り要求の場合は、モードグループ統計レコード UEPMODST のアドレスはゼロに設定されます。このアドレスは、要求が特定のモードグループの場合にのみ提供されます。

キューイングの抑止後に割り振りが正常に行われた後でこの出口ルーチンを呼び出す場合は、次の戻りコードを使用できます。

#### **UERCNORM**

この戻りコードは、CICS が要求のキューイングを含めてリンク上で通常の処理を再開することを示します。

#### **DFHA14DS および DFHA20DS の統計フィールド**

XZIQUE グローバル・ユーザー出口プログラムがキューを制御するために使用できる、いくつかの統計フィールドがあります。

#### **A14EALRJ**

XZIQUE グローバル・ユーザー出口プログラムが戻る際に要求の拒否が要求されるたびに、CICS はシステム・エントリー接続統計のフィールド A14EALRJ を増分します。

フィールド A14EALRJ (割り振り拒否) は DSECT DFHA14DS にあり、キュー制限の調整を支援するために提供されています。通常、セッションの数とリンクに対して定義されたキュー制限の正しい均衡が保たれ、リンク上で異常な輻輳 (ふくそう) が発生していない場合、A14EALRJ はゼロになります。拒否された割り振りのフィールドがゼロ以外の場合は、何らかのアクションが必要であることを示している可能性があります。

#### **A14EQPCT および A20EQPCT**

XZIQUE グローバル・ユーザー出口プログラムが戻る際にキューのページが要求されるたびに、CICS はシステム・エントリー接続統計 (フィールド A14EQPCT) またはモード・エントリー接続統計 (フィールド A20EQPCT) の新規フィールドを増分します。

#### **A14EQPCT**

接続全体でキューがページされた回数のカウント。

#### **A20EQPCT**

モード・グループ・キューがページされた回数のカウント。

統計フィールドの詳細、フィールドの内容、および更新方法について詳しくは、[ISC/IRC システム・エントリー: リソース統計](#)を参照してください。

#### **出口 XZIQUE**

出口 XZIQUE が呼び出されるのは、セッションの割り振り要求がキューに入れられるとき、および以前のキューイング抑止の後に割り振り要求が成功したときです。

#### **呼び出される状況**

以下の場合に必ず呼び出されます。

1. セッションに対する割り振り要求がキューに入れられようとしているとき
2. 直前のキューイング抑止の後に割り振り要求が成功したとき

#### **出口固有のパラメーター**

##### **UEPZDATA**

リストされた情報が入っている 70 バイト域のアドレス。この領域は、コピーブック DFHXZIDS 内の DSECT によってマップされます。

#### **UEPZDATA によってアドレス指定される領域**

##### **UEPSYSID**

接続の 4 バイトの SYSID。

##### **UEPREQ**

2 バイトの要求起点コード。以下のいずれかの値をとります。

**TR**

トランザクション・ルーティング

**FS**

機能シップ (分散プログラム・リンクを含む)

**AL**

その他の種類の相互通信 (例えば、分散トランザクション処理 (DTP) または CPI コミュニケーション)

**UEPREQTR**

要求側トランザクションの 4 バイト ID (要求起点コードが FS または AL の場合にのみ適用可能)

**UEPTRAN**

経路指定されるトランザクションの 4 バイト ID (要求起点が TR の場合にのみ適用可能)

**UEPFLAG**

最後に出口が呼び出されたときに、戻りコード 8 または戻りコード 12 が発行されたかどうかを示す 1 バイトのフラグ。

**UEPRC8**

前の呼び出しで、出口プログラムが戻りコード 8 を出して CICS に制御を戻しました。

**UEPRC12**

前の呼び出しで、出口プログラムが戻りコード 12 を出して CICS に制御を戻しました。

**UEPPAD**

1 バイトの埋め込みフィールド。

**UEPFSPL**

10 バイトの機能シップ・パラメーター・リストのアドレス。

**UEPCONST**

158 バイトのシステム・エントリー統計レコードのアドレス (これは、DSECT DFHA14DS を使用してマップできます)。

**UEPMODST**

関連する CICS プロファイルに指定されているモードグループに対する、84 バイトのモードグループ統計レコードのアドレス。このフィールドは、特定割り振りの APPC 接続にのみ適用されます。LU61、IRC、または非特定 APPC 割り振りの場合は、ゼロが入ります。

統計レコードは、DSECT DFHA20DS を使用してマップできます。モードグループ名フィールド (A20MODE) にはブランクが入る場合があります。このレコードの後に、フルワード X'FFFFFFFF' が続きます。

**UEPSTEX**

モードグループ統計レコード (DFHA20DS) にまだ含まれていない、APPC の追加現行統計が格納される 6 バイトの領域。特定割り振りの場合、数値は指定されたモードグループのみを参照します。非特定割り振りの場合は、接続全体を参照します。つまり、数値はそれぞれのモードグループの合計です。

6 バイト領域には、以下が含まれています。

**UEPEBND**

バインド済みセッションの数を含むハーフワード 2 進数フィールド

**UEPEWWT**

タスクのコンテンション勝者の数を格納するハーフワード 2 進数フィールド

**UEPELWT**

タスクのコンテンション敗者の数を格納するハーフワード 2 進数フィールド

**UEPEMXQT**

接続に対して指定された最大キューイング時間を格納するハーフワード 2 進数フィールド (CONNECTION リソース定義の MAXQTIME)。



## UEPMDGST

モードグループ統計レコード (84 バイト) のセットのアドレス (接続のユーザー・モードグループごとに 1 つずつ)。このフィールドは、非特定割り振りの APPC 接続にのみ適用されます。LU61、IRC、および APPC の特定割り振りの場合は、ゼロが入ります。

それぞれの統計レコードは、DSECT DFHA20DS を使用してマップできます。モードグループ名フィールド (A20MODE) にはブランクが入る場合があります。レコードのセットの終わりは、フルワード X'FFFFFFFF' によって示されます。

**非特定割り振りデータ:** 以下の 3 つのフィールドには、MRO、LU6.1、および非特定 APPC 割り振りに関連するデータが格納されます。

## UEPSAQTS

非特定要求のキューが開始された時刻を示す、TCT システム・エントリーからのタイム・スタンプを格納するダブルワード 2 進数フィールド。

## UEPSACNT

キューの開始後に処理された、すべての非特定割り振り要求の数を格納するハーフワード 2 進数フィールド (開始時刻については、UEPSAQTS を参照)。

## UEPSARC8

CICS に対する UEPCAKLL 戻りコードの結果としてキューが最後にページされた後、解放されたセッションの数を格納するハーフワード 2 進数フィールド。

**特定割り振りデータ:** 以下の 3 つのフィールドには、特定のモードグループ割り振りに関連するデータが格納されます。これらは、UEPMODST がゼロ以外である (つまり、関連するモードグループ統計のアドレスを含んでいる) 場合にのみ該当します。

## UEPMAQTS

この特定のモードグループに対してモードグループ・キューが開始された時刻を示す、TCT モード・エントリーからのタイム・スタンプを格納するダブルワード 2 進数フィールド。

## UEPMACNT

キューの開始後に処理された、このモードグループに対するすべての特定割り振り要求の数を格納するハーフワード 2 進数フィールド (開始時刻については、UEPMAQTS を参照)。

## UEPMAR12

CICS に対する UEPCAKLL 戻りコードの結果としてキューが最後にページされた後、解放されたモードグループ・セッションの数を格納するハーフワード 2 進数フィールド。

## UEPQUELM

この接続に対して指定されたキュー制限を格納するハーフワード 2 進数フィールド (CONNECTION 定義の QUEUELIMIT)。

## 戻りコード

キューに入れられようとしている割り振りの場合は、以下のいずれかを使用します。

## UERCAQUE

割り振り要求をキューに入れる。

## UERCAPUR

SYSIDERR によってこの割り振り要求を拒否します。

## UEPCAKLL

SYSIDERR によってこの割り振り要求を拒否します。キューに入れられた他の割り振り要求をすべてページし、オペレーター・コンソールに情報メッセージを送信します。また CICS は、ページされた割り振り要求を待機しているすべてのアプリケーション・プログラムに SYSIDERR を返します。

## UEPCAKLM

モードグループに対するこの割り振り要求を拒否して、SYSIDERR を返します。この割り振り要求で指定されたモードグループに対する、キューに入れられた他の割り振り要求をすべてページし、オペレーター・コンソールに情報メッセージを送信します。間隔の後で、モードグループを再試行します。

## UERCPURG

タスクは XPI 呼び出し中にページされます。

前の出口の呼び出しで UERCAKLL または UERCAKLM を使用した後に割り振りが成功した場合には、以下のいずれかを使用します。

#### **UERCNORM**

リンクまたはモードグループの通常の操作を再開します。

#### **UERCAPUR**

SYSIDERR によってこの割り振り要求を拒否します。

#### **XPI 呼び出し**

すべて使用できます。

## **XZIQUE グローバル・ユーザー出口プログラムの設計**

XZIQUE 出口の機能は、以下のように設計する必要があります。

1. 空きのシステム間セッションをキュー内で待機するタスクの数 (および関連したリソースの量) を制御します。待機中のタスクが、ローカル・システムのパフォーマンスを低下させる可能性があります。
2. 受信側 (リモート) システムからの応答が悪化していることを検出し、オペレーター (または自動操作プログラム) に通知します。
3. リンクが通常操作を再開したときに CICS からメッセージを発行します。

XZIQUE グローバル・ユーザー出口パラメーター・リストは、これらの目標をサポートするように設計されます。

### **サンプル XZIQUE 出口プログラム**

サンプル XZIQUE 出口プログラム DFH\$XZIQ は、ユーザー独自のグローバル・ユーザー出口プログラムを設計するための基礎として CICS Transaction Server for z/OS, バージョン 5 リリース 6 に付属しています。このサンプルは、CICSTS56.CICS.SDFHSAMP ライブラリー内で提供されています。UEPZDATA によってアドレス指定される領域をマップするためにサンプル・プログラムによって使用される DSECT オブジェクト DFHXZIDS は、CICSTS56.CICS.SDFHMAC ライブラリー内で提供されています。

DFH\$XZIQ について詳しくは、[MRO および APPC セッション・キュー管理サンプル出口プログラム: DFH\\$XZIQ](#) を参照してください。

### **設計上の考慮事項**

XZIQUE グローバル・ユーザー出口プログラムで以下のことが可能になるように、XZIQUE に渡される情報を設計します。

- 応答時間の悪化と完全なボトルネックを区別することによって、接続の問題の誤診断を防ぐ
- リモート・システムの問題が解決した後、リンクの通常操作を迅速に再開し、オペレーター介入が必要ないようにする

### **IRC/ISC 統計の使用に関するガイダンス**

CICS は、満たされない割り振り要求に関するエントリーを、非特定 (汎用) 割り振りキュー と特定モードグループ割り振りキュー に追加します。

#### **非特定 (汎用) 割り振りキュー**

すべての非特定の割り振り要求がこの単一キューに入れられます。CICS は、システム・エントリー統計フィールド A14ESTAQ に、このキュー内のエントリーの総数を格納します。グローバル・ユーザー出口プログラムは、UEPCONST で渡されるシステム・エントリー統計のアドレスを使用して、このフィールドにアクセスできます。

#### **特定モードグループ割り振りキュー**

特定割り振り要求は、適切なモードグループ・キューに入れられます。特定モードグループ名ごとに、キューが 1 つずつあります。CICS は、モード・エントリー統計フィールド A20ESTAQ に、これらすべてのキュー内のエントリーの総数を格納します。グローバル・ユーザー出口プログラムは、UEPMODST で渡されるモード・エントリー統計のアドレスを使用して、このフィールドにアクセスできます。

## IPIC システム間キュー管理用の XISQUE 出口

XISQUE 出口を使用すると、IP 相互接続性 (IPIC) 接続でのキューイングを制御できます。

XISQUE 出口は、IPIC 接続上でキューに入れられる以下の要求やコマンドを制御します。

- セッションに関する分散プログラム・リンク (DPL) の要求
- トランザクション・ルーティングの要求
- 機能シッパ要求
- START または CANCEL コマンド

キューイングの問題 (ボトルネック) を早期に検出するには、XISQUE 出口を使用します。

XISQUE により、割り振り要求をキューの長さに応じてキューに入れるか、または拒否することができます。また、ボトルネックが存在する IPCONN を終了してから、再確立することもできます。

### 出口 XISQUE

CICS がセッションの獲得を試行したが使用可能な空きセッションがなかったときや、以前にキューイングが抑止された後で割り振り要求によって空きセッションが検出されたとき、XISQUE グローバル・ユーザー出口が呼び出されます (使用可能な場合)。出口に固有のパラメーター、戻りコード、および XPI 呼び出しについての情報が説明されています。

#### 呼び出し時

XISQUE 出口は、以下の状況で呼び出されます。

1. CICS は、リモート・システムへの IPIC 接続上でセッションの獲得を試みていますが、使用可能な空きセッションがありません。この出口は、IPCONN 定義に QUEUELIMIT オプションが指定されているかどうかや、その限度を超えているかどうかにかかわらず呼び出されます。

IPIC セッションの要求は、IPIC 接続を経由して以下のいずれかの要求またはコマンドが使用されると発生します。

- 分散プログラム・リンク (DPL) 要求
  - START コマンドまたは CANCEL コマンド
  - トランザクション・ルーティング要求
  - 機能シッパのファイル制御、一時データ、または一時記憶域の要求
2. IPIC 接続に対するキューが出口プログラムの前の呼び出しによってページされた後に、IPIC 割り振り要求が空きセッションを見つけることができた。この場合には、CICS が通常どおり処理を続け、必要ならキューイングを再開するように、出口プログラムで指定することができます。

#### 出口固有のパラメーター

##### UEPISDATA

78 バイト領域のアドレス。この領域は、コピーブック DFHXIQDS 内の DSECT によってマップされます。

**UEPISDATA** によってアドレス指定される領域:

##### UEPREQ

2 バイトの要求起点コード。以下のいずれかの値をとります。

##### AL

その他の種類の相互通信 (例: START)。

##### FS

機能シッパと分散プログラム・リンク

##### TR

トランザクション・ルーティング

##### UEPIPCNM

IPCONN の 8 バイトの名前。

**UEPREQTR**

要求側トランザクションの 4 バイトの ID。

**UEPFLAG**

最後に出口が呼び出されたときに、戻りコード 8 が発行されたかどうかを示す 1 バイトのフラグ。

**UEPRC8**

前の呼び出しで、出口プログラムが戻りコード 8 を出して CICS に制御を戻しました。

**UEPFSPL**

DPL 要求に対する 10 バイトのパラメーター・リストのアドレス。

**UEPCONST**

504 バイトの IPCONN 統計レコードのアドレス。このレコードは、DSECT DFHISRDS を使用してマップできます。

**UEPEMXQT**

IPCONN リソース定義に指定された最大キューイング時間 MAXQTIME を含むハーフワード 2 進数フィールド。

**UEPSAQTS**

インストールされた IPCONN リソース定義からのタイム・スタンプを含むダブルワード 2 進数フィールドであり、割り振り要求のキューが開始された時刻を示します。

**UEPSACNT**

キューの開始後に処理された割り振り要求の数を含むハーフワード 2 進数フィールド。開始時刻については、UEPSAQTS を参照してください。

**UEPSARC8**

UEPCAKLL 戻りコードの結果としてキューが最後にページされた後、解放されたセッションの数を含むハーフワード 2 進数フィールド。

**UEPQUELM**

IPCONN リソース定義に指定されたキュー制限 QUEUELIMIT を含むハーフワード 2 進数フィールド。

**戻りコード**

キューに入れられようとしている割り振りの場合は、以下のいずれかの戻りコードを使用します。

**UERCAQUE**

割り振り要求をキューに入れる。

**UERCALL**

SYSIDERR によってこの割り振り要求を拒否します。キューに入れられた他の割り振り要求をすべてページし、オペレーター・コンソールに情報メッセージを送信します。また CICS は、ページされた割り振り要求を待機しているすべてのアプリケーション・プログラムに SYSIDERR を返します。

**UERCAPUR**

SYSIDERR によってこの割り振り要求を拒否します。

**UERPURG**

タスクは XPI 呼び出し中にページされます。

前の出口の呼び出しで UERCALL を使用した後で割り振りが成功した場合には、以下に示す通常の戻りコード (UERCNORM) を使用するか、または SYSIDERR 戻りコード (UERCAPUR) を使用できます。

**UERCNORM**

IPCONN の通常操作を再開します。

**UERCAPUR**

SYSIDERR によってこの割り振り要求を拒否します。

**XPI 呼び出し**

すべて使用できます。

## XISQUE グローバル・ユーザー出口プログラムの使用

この出口が使用可能になっていれば、XISQUE グローバル・ユーザー出口プログラムによって、ローカル・システムの IPCONN に対する割り振りキューの状態を検査することができます。呼び出し時に出口プログラムに渡されるパラメーター・リストは、特定の割り振り要求と IPCONN に関するデータを提供します。

パラメーター・リストで渡される情報を使用して、グローバル・ユーザー出口プログラムは、CICS が割り振り要求をキューに入れるかどうかを決定できます (例えば、キューの長さに基づいて)。プログラムは、以下のいずれかの戻りコードを設定して、決定内容を CICS に伝達します。

### UERCAQUE

CICS は割り振り要求をキューに入れます。

接続に対して現在キューに入れられている割り振り要求の合計数は、IPCONN 統計レコードのフィールド `ISR_CURRENT_QUEUED_ALLOCATES` で提供されています。この統計レコードのアドレスは、UEPCONST 出口固有パラメーターによって指定されます。詳しくは、DSECT DFHISRDS を参照してください。

また CICS は、IPCONN リソース定義の `QUEUELIMIT` オプションの値を、フィールド `UEPQUELM` に格納して出口プログラムに渡します。キュー限度に達していない場合は、戻りコード `UERCAQUE` を出して CICS に制御を戻すことができます。

### UERCAPUR

CICS は割り振り要求を拒否し、`SYSIDERR` をアプリケーション・プログラムに返して、既存のキューはそのまま変更しません。

キューに入れられた割り振り要求の数が、IPCONN 定義の `QUEUELIMIT` オプションで設定された限度に達した場合は、要求を拒否するように CICS に要求できます。ただし、まずリンクの状態が正常であるかどうかを確認する必要があります。つまり、セッションの割り振りの速度が許容できるかどうかを検査します。キューが開始された時刻、現在時刻、およびキューの開始以降に処理された割り振りの総数を使用して、CICS が要求を処理する速度を判別します。該当するフィールドは、`UEPSAQTS` と `UEPSACNT` です。

CICS がこの IPCONN 上でセッションの要求を許容できる速度で割り振っているかどうか判別するには、計算した時間を以下のいずれかと比較できます。

1. IPCONN リソース定義の `MAXQTIME` オプションの値 (`UEPEMXQT` 出口固有パラメーターで渡される)
2. その他の事前設定された時間値。

このような公式を使用して、処理時間が許容できると判断した場合は、戻りコード `UERCAPUR` を指定して CICS に制御を戻します。この戻りコードは、この要求のみをパージします。

### UERCALL

この要求を拒否し、この IPCONN に対するキューに入れられた他の割り振り要求をすべてパージして、オペレーター・コンソールに情報メッセージを送信します。

キュー限度に達したときに、割り振り処理のパフォーマンスがユーザー出口プログラムに定義されている許容限度を下回っている場合は、戻りコード `UERCALL` を指定して、キューに入れられたすべての割り振り要求をパージできます。

また、`UERCALL` によって CICS は以下のことを行います。

- パージされた割り振り要求を待機しているすべてのアプリケーション・プログラムに `SYSIDERR` を返します。
- XISQUE 出口プログラムの後続の呼び出しでは、`UEPFLAG` パラメーターを `UEPRC8` に設定して、以前にキューをパージするために `UERCALL` が返されたことを示します。

タスクのフローを混雑させているキューをパージすることによって、システムの停滞を防ぐために必要なタスク・スロットが解放されます。セッション・キューの拡大を可能にする限度が大きいほど、`MAXT` パラメーターによって設定されたタスクの上限に達する可能性が高くなり、その後は接続できないローカル領域内で着信タスクのキューが発生します。

キューが以前にパージされていた場合 (`UERCALL` によって) に、現在はキューに入れられた要求がない場合は、キューが最後にパージされた後に解放されたセッションの数を確認します。この数は

UEPSARC8 に格納されています。キューが最後にパージされた後にこの IPCONN 上で解放されたセッションがない場合は、以前にパージの原因になった問題が解決しておらず、この要求は UERCAPUR によって拒否する必要があります。

セッションが解放されていることを UEPSARC8 パラメーターが示している場合は、UERCAQUE を使用して要求のキューイングを再開する必要があります。この場合に UERCAQUE を使用して戻ると、CICS はキューイングが再開されたことを示す情報メッセージをコンソールに発行します。

#### **UERCNORM**

CICS は、要求のキューイングを含む通常の処理を接続上で再開します。

キューイングの抑止に続いて、割り振りが正常に行われた後で出口を呼び出す場合は、UERCNORM を使用します

## **DFHISRDS の統計フィールド**

IPCONN 統計の以下のフィールドは、XISQUE グローバル・ユーザー出口プログラムがキューの割り振りを効率的に制御するために役立ちます。

#### **ISR\_XISQUE\_ALLOC\_REJECTS**

XISQUE グローバル・ユーザー出口プログラムが戻る際に要求の拒否が要求されるたびに、CICS はこのフィールドを増分します。これは、キュー制限を調整するために提供されています。通常、セッションの数と IPCONN 定義で指定されたキュー制限の正しい均衡が保たれ、リンク上で異常な輻輳(ふくそう)が発生していない場合、ISR\_XISQUE\_ALLOC\_REJECTS はゼロになります。拒否された割り振りのフィールドがゼロ以外の場合は、アクションが必要になる可能性があることを示しています。

#### **ISR\_XISQUE\_ALLOC\_QPURGES**

XISQUE グローバル・ユーザー出口プログラムが戻る際にキューのパージが要求されるたびに、CICS はこのフィールドを増分します。

IPCONN 統計のフィールドについて詳しくは、[IPCONN レポート](#)を参照してください。IPCONN 統計は DSECT DFHISRDS によってマップされます。

## **XISQUE グローバル・ユーザー出口プログラムの設計**

XISQUE 出口プログラムは、以下のことを行うように設計する必要があります。

1. キュー内で空きの IPIC セッションを待機しているタスクの数 (および関連したりソースの量) の制御。待機中のタスクが、ローカル・システムのパフォーマンスを低下させる可能性があります。
2. リモート・システムからの応答が悪化していることを検出し、オペレーター (または自動操作プログラム) に通知する。
3. IPCONN が通常操作を再開したときに CICS からメッセージを発行する。

XISQUE パラメーター・リストは、これらの目標をサポートするように設計されます。含まれる情報を出口プログラムで使用するにより、以下のことが可能になります。

- 応答時間の悪化と完全なボトルネックを区別することによって、接続の問題の誤診断を防ぐ
- リモート・システムの問題が解決した後、リンクの通常操作を迅速に再開し、オペレーター介入が必要ないようにする

#### **IPCONN 統計の使用**

どの要求を拒否し、どれをキューに入れ、どのキューをパージするか決定に至る過程で、出口プログラムは接続に対してキューに現在入れられている割り振り要求の数を考慮することが考えられます。特定の IPCONN に対する割り振り要求はすべて、その IPCONN に固有の単一のキューに入れられます。CICS は、IPCONN 統計フィールド `ISR_CURRENT_QUEUED_ALLOCATES` に、このキュー内のエントリーの合計数を格納します。出口プログラムは、IPCONN 統計のアドレスを使用してこのフィールドにアクセスできます。このアドレスは、UEPCONST 出口固有パラメーターで渡されます。

## サンプル XISQUE 出口プログラム

CICS には、ユーザー独自のプログラムの基礎として使用できる、サンプル XISQUE 出口プログラム DFH\$XISQ が用意されています。これは、CICSTS56.CICS.SDFHSAMP ライブラリー内で提供されています。UEPISDATA によってアドレス指定される領域をマップするためにサンプル・プログラムによって使用される DSECT は DFHXIQDS と呼ばれ、CICSTS56.CICS.SDFHMAC ライブラリー内で提供されています。

DFH\$XISQ について詳しくは、[IPIC 接続用のセッション・キュー管理サンプル出口プログラム: DFH\\$XISQ](#) を参照してください。

## XRF 要求処理プログラム出口 XXRSTAT

この出口は、z/OS Communications Server 障害または z/OS Communications Server の略奪的なテークオーバーが発生したときに呼び出されます。

XXRSTAT を使用すると、以下のいずれかが発生した場合に CICS を終了するかどうかを決定することができます。

- TPEND 出口により CICS に z/OS Communications Server の障害が通知された。
- 略奪的なテークオーバーが発生した。略奪的なテークオーバーが発生する可能性があるのは、リリース 3.4.0 以降を使用して、実行中の CICS システムと同じアプリケーション ID を持つ z/OS Communications Server アプリケーションが、実行中の CICS システムのすべてのセッションの制御を引き継いだ場合です。

XXRSTAT では、テークオーバーの対象となるシステムを続行するか終了するかを選択することができます。

整合性が損なわれる可能性を避けるため、略奪的なテークオーバー後の CICS のデフォルト・アクションは、ダンプなしで終了することです。CICS をダンプ付きで終了するには、出口プログラムで UERCABDU を返す必要があります。CICS は、出口プログラムによって指定された異常終了コードで終了します。

略奪的なテークオーバー後に CICS で処理を続行するには、出口プログラムで UERCCOIG を返す必要があります。メッセージ DFHZC0101 が発行され、CICS は、z/OS Communications Server のサポートなしで処理を続行します。略奪側アプリケーションは、すべての z/OS Communications Server セッションの制御を引き継ぎます。

**注:** 略奪的なテークオーバー後に CICS の続行を許すことは、整合性の問題を引き起こす可能性があるため、お勧めできません。RACF を使用して、CICS アプリケーション ID を保護してください。

## XXRSTAT 出口

この出口は、z/OS Communications Server の障害または略奪的なテークオーバーが発生したときに呼び出されます。

### 呼び出される状況

以下のいずれかの後:

- TPEND 出口により CICS に z/OS Communications Server の障害が通知された。
- 略奪的なテークオーバー。

### 出口固有のパラメーター

#### UEPERRA

以下を含むパラメーター・リストのアドレス:

#### UEPGAPLD

8 バイトの汎用アプリケーション ID のアドレス

#### UEPSAPLD

8 バイトの固有アプリケーション ID のアドレス

#### UEPDOMID

4 バイトのドメイン ID のアドレス

#### UEPERRID

4 バイトのエラー ID のアドレス

注:

1. このパラメーター・リスト用の DSECT は提供されていません。指定されたフィールドにアクセスするためには、独自の DSECT をコーディングする必要があります。
2. z/OS Communications Server が失敗した場合、ドメイン ID は「ZC」(大文字の Z、大文字の C、および 2 つのブランク) となり、エラー ID は文字ストリング「3443」となります。

## 戻りコード

### **UERCNORM**

システム処置を行います。システム・アクションは、出口が呼び出された理由によって異なります。

- XRF で z/OS Communications Server の障害が発生した場合: CICS は出口プログラムが呼び出されなかった場合と同様に処理を続行します。
- z/OS Communications Server 持続セッションで略奪的なテークオーバーが発生した場合: CICS はダンプなしで異常終了します。

### **UERCCOIG**

無視します。

### **UERCABNO**

ダンプなしで CICS を異常終了します。

### **UERCABDU**

ダンプありで CICS を異常終了します。

### **UERCPURG**

タスクは XPI 呼び出し中にページされます。

## **XPI 呼び出し**

すべて使用できます。



## 特記事項

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

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

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス涉外

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

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

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

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

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

*IBM Director of Licensing*

*IBM Corporation*

*North Castle Drive, MD-NC119 Armonk,*

*NY 10504-1785*

*United States of America*

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

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

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関す

る実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

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

著作権使用許諾:

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

## プログラミング・インターフェース情報

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

オンライン製品資料の以下のセクションには、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のサービスを取得するプログラムをお客様が作成するためのプログラミング・インターフェースが含まれています。

- [アプリケーションの開発](#)
- [システム・プログラムの開発](#)
- [CICS TS セキュリティー](#)
- [外部インターフェースに向けた開発](#)
- [アプリケーション開発のリファレンス](#)
- [リファレンス: システム・プログラミング](#)
- [リファレンス: 接続](#)

オンライン製品資料の以下のセクションには、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のプログラミング・インターフェースとして意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が含まれています。

- [トラブルシューティングおよびサポート](#)
- [CICS TS 診断参照](#)

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のサービスを取得するプログラムをお客様が作成するためのプログラミング・インターフェースが以下のマニュアルに含まれています。

- [アプリケーション・プログラミング・ガイドおよびアプリケーション・プログラミング・リファレンス](#)
- [Business Transaction Services](#)
- [Customization Guide](#)
- [C++ OO Class Libraries](#)
- [Debugging Tools Interfaces Reference](#)
- [Distributed Transaction Programming Guide](#)
- [External Interfaces Guide](#)
- [Front End Programming Interface Guide](#)

- IMS Database Control Guide
- インストール・ガイド
- セキュリティー・ガイド
- Supplied Transactions
- CICSplex SM Managing Workloads
- CICSplex SM Managing Resource Usage
- CICSplex SM アプリケーション・プログラミング・ガイドおよび CICSplex SM アプリケーション・プログラミング・リファレンス
- CICS における Java™ アプリケーション

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 6 のプログラミング・インターフェースとして意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が以下のマニュアルに含まれています。

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

## 商標

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

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

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

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

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

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

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

## 製品資料に関するご使用条件

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

### 適用範囲

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

### 個人使用

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

## 商用使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

## 権利

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

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

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

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

## IBM オンラインでのプライバシー・ステートメント

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

### CICSplex SM Web ユーザー・インターフェース (メイン・インターフェース) の場合:

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

### CICSplex SM Web ユーザー・インターフェース (データ・インターフェース) の場合:

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

### CICSplex SM Web ユーザー・インターフェース (「Hello World」ページ) の場合:

このソフトウェア・オファリングは、展開される構成に応じて、個人情報を収集しないセッションごとの Cookie を使用する場合があります。これらの Cookie を無効にすることはできません。

### CICS Explorer の場合:

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

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

このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、『IBM オンラインでのプライバシー・ステートメント』 (<http://www.ibm.com/privacy/details/jp/ja/>) の『クッキー、ウェブ・ビー

コン、その他のテクノロジー』および『IBM Software Products and Software-as-a-Service Privacy Statement』 (<http://www.ibm.com/software/info/product-privacy>) を参照してください。





# 索引

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

## [ア行]

アルファベット順リスト、GLUE の [1](#)

## [カ行]

機能リスト、GLUE [1](#)

基本マッピング・サポート (BMS)

グローバル・ユーザー出口点 [12](#)

キュー、システム間セッションの

制御、長さの

使用、XISCONA グローバル・ユーザー出口の [121](#)

XZIQUE グローバル・ユーザー出口の使用 [236](#)

グローバル・ユーザー出口

サンプル・プログラム

DFH\$ICCN [145](#)

DFH\$WBGA [116](#)

出口点

アプリケーション関連データの [11](#)

一時記憶域ドメイン内 [199](#)

一時ストレージ EXEC インターフェース・プログラムの [204](#)

一時データ EXEC インターフェース・プログラム [224](#)

一時データ・プログラム [222](#)

インターバル制御 EXEC インターフェース・プログラムの [128](#)

インターバル制御プログラムの [126](#)

エンキュー EXEC インターフェース・プログラムの [40](#)

活動キーポイント・プログラムの [10](#)

システム間通信プログラムの [121](#)

システム終了プログラムの [199](#)

システム・リカバリー・プログラム内 [196](#)

セキュリティ・マネージャー・ドメイン [192](#)

ダンプ・ドメイン [34](#)

端末管理プログラム [214](#)

「端末未認識」状態の [216](#)

端末割り振りプログラムの [213](#)

ディスパッチャー・ドメインの [24](#)

データ・テーブル管理 [17](#)

データ・テーブル・プログラム [17](#)

統計ドメイン [195](#)

トランザクション・マネージャー・ドメインの [221](#)

パイプライン・ドメイン内の [156](#)

ファイル制御 EXEC インターフェース・プログラム [67, 78](#)

ファイル制御 RLS 共存での [113](#)

ファイル制御オープン/クローズ・プログラムの [101](#)

ファイル制御静止受信プログラム [102](#)

ファイル制御静止送信プログラムの [104](#)

ファイル制御ドメインでの [55](#)

ファイル制御ファイル状態プログラムの [93](#)

グローバル・ユーザー出口 (続き)

出口点 (続き)

ファイル制御リカバリー・プログラムの [106](#)

ブリッジ機能の削除 [16](#)

ブリッジ機能の作成 [16](#)

プログラム管理プログラムの [169](#)

フロントエンド・プログラミング・インターフェースの [51](#)

メッセージ・ドメイン [151](#)

モニター・ドメイン [154](#)

ユーザー・ログ・レコード・リカバリー・プログラム内 [231](#)

リソース管理モジュール [186](#)

リソース・マネージャー・インターフェース・プログラムの [184](#)

ローダー・ドメイン [146](#)

ログ・マネージャー・ドメイン [148](#)

BMS [12](#)

CICS Web サポートの [115](#)

DBCTL インターフェース 制御プログラム [22](#)

DBCTL トラッキング・プログラム [22](#)

DL/I インターフェース・プログラム [25](#)

EXEC インターフェース・プログラム内 [47](#)

Good morning メッセージ・プログラムの [115](#)

SNA LU 管理プログラム [234](#)

SNA 実効ページ・セット・モジュール内の [235](#)

XRF 要求処理プログラム [247](#)

プログラム例

TS 要求の変更 [213](#)

XFCREQ の場合 [76, 78](#)

XFCREQC の場合 [76, 78](#)

XICREQ の場合 [145](#)

XICREQC の場合 [145](#)

XPCREQ の場合 [177](#)

XPCREQC の場合 [177](#)

XTDEREQ の場合 [224](#)

XTDEREQC の場合 [224](#)

XTSEREQ の場合 [213](#)

XTSEREQC の場合 [213](#)

XDLPRE

例 [29](#)

XSZARQ

概要 [53](#)

出口固有のパラメーター [53](#)

UEPSZACN パラメーター [53](#)

XSZBRQ

概要 [52](#)

UEPSZACT パラメーター [53](#)

## [サ行]

サンプル・プログラム

グローバル・ユーザー出口用

DFH\$WBGA、XWBOPEN 出口用 [116](#)

システム間キュー

制御、長さの

使用、XISCONA グローバル・ユーザー出口の [121](#)

システム間キュー (続き)  
制御、長さの (続き)  
XZIQUE グローバル・ユーザー出口の使用 [236](#)  
システム間セッション用のキュー  
長さの制御  
XISQUE グローバル・ユーザー出口の使用 [243](#)  
システム初期設定パラメーター  
AILEDAY [186](#)  
TBEXITS [107](#), [233](#)

## [タ行]

データ・テーブル [17](#)  
出口 XTSPTOUT [203](#)  
出口 XTSQRIN [200](#)

## [ラ行]

ローカル DL/I と DBCTL の共存  
スケジュールに入れる PSB を変更するための XDLPRE  
[29](#)

## [ワ行]

割り振り、キューの  
制御、長さの  
使用、XISCONA グローバル・ユーザー出口の [121](#)  
XISQUE グローバル・ユーザー出口の使用 [243](#)  
割り振りキュー  
制御、長さの  
XZIQUE グローバル・ユーザー出口の使用 [236](#)

## A

AILEDAY、システム初期設定パラメーター [186](#)  
APPC 接続  
システム間キュー [236](#)  
XZIQUE グローバル・ユーザー出口  
システム間キューの制御用の [236](#)

## C

CICS Web サポート  
ユーザー出口 XWBAUTH、XWBOPEN、XWBSNDO [116](#)  
ユーザー出口 XWBOPEN、XWBSNDO [119](#), [120](#)

## D

DFH\$ICCN、グローバル・ユーザー出口のサンプル・プログラム [145](#)  
DFH\$WBGA、サンプルのグローバル・ユーザー出口プログラム [116](#)

## G

GLUE、アルファベット順リスト [1](#)  
GLUE、機能リスト [1](#)

## H

HTTP クライアントのオープン 出口 XWBOPEN [119](#)

## I

IPCONN  
システム間キュー [243](#)  
XISQUE グローバル・ユーザー出口  
システム間キューの制御用の [243](#)  
ISC over TCP/IP のシステム間キュー  
制御、長さの  
XISQUE グローバル・ユーザー出口の使用 [243](#)  
ISC over TCP/IPCONN  
システム間キュー [243](#)  
XISQUE グローバル・ユーザー出口  
システム間キューの制御用の [243](#)

## M

MRO 接続  
システム間キュー [236](#)  
XZIQUE グローバル・ユーザー出口  
システム間キューの制御用の [236](#)

## P

PSB (プログラム仕様ブロック)  
スケジュールに入れる PSB を変更するための XDLPRE  
[29](#)

## T

TBEXITS、システム初期設定パラメーター [107](#), [233](#)

## U

UEPSZACN、XSZARQ の出口固有のパラメーター [53](#)  
UEPSZACT、XSZBRQ の出口固有のパラメーター [53](#)

## X

XALCAID、グローバル・ユーザー出口 [213](#)  
XALTENF、グローバル・ユーザー出口 [217](#)  
XAPADMGR、グローバル・ユーザー出口 [11](#)  
XBMIN、グローバル・ユーザー出口 [13](#)  
XBMOU、グローバル・ユーザー出口 [13](#)  
XDLPPOST、グローバル・ユーザー出口 [27](#)  
XDLPRE、グローバル・ユーザー出口  
スケジュールに入れる PSB を変更するための [29](#)  
XDSAWT、グローバル・ユーザー出口 [24](#)  
XDSBWT、グローバル・ユーザー出口 [24](#)  
XDTAD、グローバル・ユーザー出口 [20](#)  
XDTLC、グローバル・ユーザー出口 [21](#)  
XDTRD、グローバル・ユーザー出口 [18](#)  
XDUCLSE、グローバル・ユーザー出口 [39](#)  
XDUOUT、グローバル・ユーザー出口 [39](#)  
XDUREQ、グローバル・ユーザー出口 [34](#)  
XDUREQC、グローバル・ユーザー出口 [36](#)  
XEIIN、グローバル・ユーザー出口 [48](#)  
XEIOUT、グローバル・ユーザー出口 [50](#)  
XEISPIN、グローバル・ユーザー出口 [49](#)  
XEISPOUT、グローバル・ユーザー出口 [51](#)  
XEPCAP、グローバル・ユーザー出口 [46](#)  
XFAINTU、グローバル・ユーザー出口 [16](#)  
XFCAREQ、グローバル・ユーザー出口  
説明 [78](#)



XFCAREQ、グローバル・ユーザー出口 (続き)  
   パラメーター・リストおよび戻りコード [79](#)  
 XFCAREQC、グローバル・ユーザー出口  
   説明 [78](#)  
   パラメーター・リストおよび戻りコード [80](#)  
 XFCBFAIL、グローバル・ユーザー出口 [107](#)  
 XFCBOUT、グローバル・ユーザー出口 [110](#)  
 XFCBOVER、グローバル・ユーザー出口 [111](#)  
 XFCFRIN、グローバル・ユーザー出口  
   パラメーター・リストおよび戻りコード [56](#)  
 XFCFROUT、グローバル・ユーザー出口  
   パラメーター・リストおよび戻りコード [62](#)  
 XFCLDEL、グローバル・ユーザー出口 [112](#)  
 XFCNREC、グローバル・ユーザー出口  
   説明 [101](#)  
   パラメーター・リストおよび戻りコード [101](#)  
 XFCQUIS、グローバル・ユーザー出口  
   説明 [104](#)  
 XFCREQ、グローバル・ユーザー出口  
   コマンド・パラメーター構造 [68](#)  
   使用例 [76](#)  
   説明 [67](#)  
   パラメーター・リストおよび戻りコード [76](#)  
   UEPCLPS パラメーター [69](#)  
 XFCREQC、グローバル・ユーザー出口  
   コマンド・パラメーター構造 [68](#)  
   使用例 [76](#)  
   説明 [67](#)  
   パラメーター・リストおよび戻りコード [77](#)  
   UEPCLPS パラメーター [69](#)  
 XFCRLSCO、グローバル・ユーザー出口  
   説明 [113](#)  
 XFCSREQ、グローバル・ユーザー出口 [94](#)  
 XFCSREQC、グローバル・ユーザー出口 [97](#)  
 XFCVSDS、グローバル・ユーザー出口  
   説明 [102](#)  
 XGMTEXT、グローバル・ユーザー出口 [115](#)  
 XICEREQ、グローバル・ユーザー出口  
   コマンド・パラメーター構造 [135](#)  
   使用例 [145](#)  
   パラメーター・リストおよび戻りコード [130](#)  
   UEPCLPS パラメーター [135](#)  
 XICEREQC、グローバル・ユーザー出口  
   コマンド・パラメーター構造 [135](#)  
   使用例 [145](#)  
   パラメーター・リストおよび戻りコード [132](#)  
   UEPCLPS パラメーター [135](#)  
 XICERES、グローバル・ユーザー出口  
   パラメーター・リストおよび戻りコード [131](#)  
 XICEXP、グローバル・ユーザー出口 [128](#)  
 XICREQ、グローバル・ユーザー出口 [127](#)  
 XICTENF、グローバル・ユーザー出口 [219](#)  
 XISCONA、グローバル・ユーザー出口 [121](#), [123](#)  
 XISLCLQ グローバル・ユーザー出口 [121](#), [124](#)  
 XISQLCL グローバル・ユーザー出口 [121](#), [125](#)  
 XISQUE、グローバル・ユーザー出口  
   概要 [243](#)  
   使用方法 [245](#)  
   出口プログラムの設計 [246](#)  
 XLDELETE、グローバル・ユーザー出口 [147](#)  
 XLDLOAD、グローバル・ユーザー出口 [146](#)  
 XLGSTRM、グローバル・ユーザー出口 [148](#)  
 XMEOUT [151](#)  
 XMEOUT、グローバル・ユーザー出口 [152](#)  
 XMNOUT、グローバル・ユーザー出口 [154](#)  
 XNQEREQ、グローバル・ユーザー出口  
   コマンド・パラメーター構造 [43](#)  
   UEPCLPS パラメーター [43](#)  
 XNQEREQC、グローバル・ユーザー出口  
   コマンド・パラメーター構造 [43](#)  
   UEPCLPS パラメーター [43](#)  
 XPCABND、グローバル・ユーザー出口 [183](#)  
 XPCERES、グローバル・ユーザー出口  
   説明 [169](#)  
   パラメーター・リストおよび戻りコード [171](#)  
 XPCFTCH、グローバル・ユーザー出口 [178](#)  
 XPCHAIR、グローバル・ユーザー出口 [180](#)  
 XPCREQ、グローバル・ユーザー出口  
   コマンド・パラメーター構造 [173](#)  
   使用例 [177](#)  
   説明 [169](#)  
   パラメーター・リストおよび戻りコード [170](#)  
   UEPCLPS パラメーター [174](#)  
 XPCREQC、グローバル・ユーザー出口  
   コマンド・パラメーター構造 [173](#)  
   使用例 [177](#)  
   説明 [170](#)  
   パラメーター・リストおよび戻りコード [172](#)  
   UEPCLPS パラメーター [174](#)  
 XPCTA、グローバル・ユーザー出口 [181](#)  
 XRCINIT、グローバル・ユーザー出口 [233](#)  
 XRCINPT、グローバル・ユーザー出口 [233](#)  
 XRMIOUT、グローバル・ユーザー出口 [185](#)  
 XRMMI、グローバル・ユーザー出口 [184](#)  
 XRSINDI [187](#)  
 XRSINDI、グローバル・ユーザー出口 [187](#)  
 XSNEX、グローバル・ユーザー出口 [194](#)  
 XSNOFF、グローバル・ユーザー出口 [193](#)  
 XSNON、グローバル・ユーザー出口 [192](#)  
 XSRAB、グローバル・ユーザー出口 [196](#)  
 XSTERM、グローバル・ユーザー出口 [199](#)  
 XSTOUT、グローバル・ユーザー出口 [195](#)  
 XSZARQ、グローバル・ユーザー出口  
   概要 [53](#)  
   UEPSZACN パラメーター [53](#)  
 XSZBRQ、グローバル・ユーザー出口  
   概要 [52](#)  
   UEPSZACT パラメーター [53](#)  
 XTCATT、グローバル・ユーザー出口 [215](#)  
 XTCIN、グローバル・ユーザー出口 [214](#)  
 XTCOUT、グローバル・ユーザー出口 [215](#)  
 XTDEREQ、グローバル・ユーザー出口  
   コマンド・パラメーター構造 [227](#)  
   パラメーター・リストおよび戻りコード [225](#)  
   UEPCLPS パラメーター [228](#)  
 XTDEREQC [224](#)  
 XTDEREQC、グローバル・ユーザー出口  
   コマンド・パラメーター構造 [227](#)  
   パラメーター・リストおよび戻りコード [226](#)  
   UEPCLPS パラメーター [228](#)  
 XTDIN [222](#)  
 XTDIN、グローバル・ユーザー出口 [223](#)  
 XTDOUT [222](#)  
 XTDOUT、グローバル・ユーザー出口 [224](#)  
 XTDREQ [222](#)  
 XTDREQ、グローバル・ユーザー出口 [222](#)  
 XTSEREQ、グローバル・ユーザー出口  
   コマンド・パラメーター構造 [207](#)

XTSREQ、グローバル・ユーザー出口 (続き)  
    プログラム例 [213](#)  
    UEPCLPS パラメーター [207](#)  
XTSREQC、グローバル・ユーザー出口  
    コマンド・パラメーター構造 [207](#)  
    プログラム例 [213](#)  
    UEPCLPS パラメーター [207](#)  
XTSPTIN グローバル・ユーザー出口 [202](#)  
XTSQROUT、グローバル・ユーザー出口 [201](#)  
XWBAUTH ユーザー出口 [116](#)  
XWBAUTH、グローバル・ユーザー出口 [115](#)  
XWBOPEN ユーザー出口 [119](#)  
XWBOPEN、グローバル・ユーザー出口 [115](#)  
XWBSNDO ユーザー出口 [120](#)  
XWBSNDO、グローバル・ユーザー出口 [115](#)  
XWSPRROI、グローバル・ユーザー出口 [158](#)  
XWSPRROO、グローバル・ユーザー出口 [159](#)  
XWSPRRWI、グローバル・ユーザー出口 [157](#)  
XWSPRRWO、グローバル・ユーザー出口 [160](#)  
XWSRQROI、グローバル・ユーザー出口 [163](#)  
XWSRQROO、グローバル・ユーザー出口 [162](#)  
XWSRQRWI、グローバル・ユーザー出口 [164](#)  
XWSRQRWO、グローバル・ユーザー出口 [161](#)  
XWSSRROI、グローバル・ユーザー出口 [167](#)  
XWSSRROO、グローバル・ユーザー出口 [166](#)  
XWSSRRWI、グローバル・ユーザー出口 [168](#)  
XWSSRRWO、グローバル・ユーザー出口 [165](#)  
XXDFA、グローバル・ユーザー出口 [22](#)  
XXDFB、グローバル・ユーザー出口 [22](#)  
XXDTO、グローバル・ユーザー出口 [23](#)  
XXMATT、グローバル・ユーザー出口 [221](#)  
XXRSTAT、グローバル・ユーザー出口 [247](#)  
XZCATT、グローバル・ユーザー出口 [234](#)  
XZCIN、グローバル・ユーザー出口 [235](#)  
XZCOUT、グローバル・ユーザー出口 [236](#)  
XZCOUT1、グローバル・ユーザー出口 [236](#)  
XZIQUE、グローバル・ユーザー出口  
    概要 [236](#)  
    使用方法 [237](#)  
    出口プログラムの設計 [242](#)  
    呼び出される時点 [237](#)  
    IRC/ISC 統計の使用 [242](#)  
    XISCONA との対話 [237](#)

## [特殊文字]

XTDEREQ [224](#)



