

**IMS**  
バージョン 13

**システム・プログラミング API**

**IBM**



**IMS**  
バージョン 13

システム・プログラミング **API**

**IBM**

お願い

本書および本書で紹介する製品をご使用になる前に、577 ページの『特記事項』に記載されている情報をお読みください。

本書は、IMS バージョン 13 (プログラム番号 5635-A04)、IMS Database Value Unit Edition V13.1 (プログラム番号 5655-DSM)、IMS Transaction Manager Value Unit Edition V13.1 (プログラム番号 5655-TM2)、および新しい版で明記されていない限り、以降のすべてのリソ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： SC19-3661-04

IMS

Version 13

System Programming APIs

(November 5, 2017 edition)

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

© Copyright IBM Corporation 1974, 2017.

# 目次

本書について . . . . .	vii
前提知識 . . . . .	vii
本書で使用されるIMS 機能名 . . . . .	vii
新規および変更された情報の識別方法 . . . . .	vii
構文図の読み方 . . . . .	viii
IMS バージョン 13 のアクセシビリティ機能 . . . . .	x

## 第 1 部 Common Queue Server (CQS) . . . . . 1

第 1 章 CQS クライアントの作成 . . . . .	3
CQS クライアント要求の要約 . . . . .	3
クライアントの出す、キュー構造に対する CQS 要求のシーケンス . . . . .	4
CQS 要求をコーディングする際の考慮事項 . . . . .	5
CQS における環境要件 . . . . .	9
CQS 要求の戻りコードと理由コード . . . . .	10
CQS クライアントおよび特殊イベントの処理 . . . . .	12
第 2 章 CQS クライアント要求 . . . . .	15
CQSBRWSE 要求 . . . . .	17
CQSCHKPT 要求 . . . . .	27
CQSCONN 要求 . . . . .	30
CQSDEL 要求 . . . . .	38
CQSDEREG 要求 . . . . .	43
CQSDISC 要求 . . . . .	44
CQSINFRM 要求 . . . . .	49
CQSMOVE 要求 . . . . .	55
CQSPUT 要求 . . . . .	59
CQSQUERY 要求 . . . . .	67
CQSREAD 要求 . . . . .	77
CQSRECVR 要求 . . . . .	84
CQSREG 要求 . . . . .	89
CQSRSYNC 要求 . . . . .	92
CQSSHUT 要求 . . . . .	100
CQSUNLCK 要求 . . . . .	102
CQSUPD 要求 . . . . .	108

## 第 2 部 共通サービス層 (CSL) (Common Service Layer (CSL)) . . . . . 115

第 3 章 CSL クライアントの作成 . . . . .	117
CSL 要求でのイベント制御ブロック . . . . .	117
SCI 要求の環境要件 . . . . .	117
CSL 要求の戻りコードおよび理由コードの解釈方法 . . . . .	119
CSL 用のクライアント作成の計画に関する考慮事項 . . . . .	119
CSL マネージャーの SCI への登録 . . . . .	121

SCI への登録 . . . . .	121
ODBM クライアントの登録 . . . . .	122
OM コマンド処理クライアントの登録 . . . . .	123
RM クライアントの登録 . . . . .	124
SCI 作動可能状態を有効にする方法 . . . . .	124
CSL 要求をコーディングする順序 . . . . .	125
全 CSL コンポーネントに共通の要求 . . . . .	125
CSLZQRY: 照会要求 . . . . .	125
CSLZSHUT: シャットダウン要求 . . . . .	127

## 第 4 章 CSL 自動化操作プログラム要求 131

CSL OM 自動化操作プログラム・クライアント . . . . .	157
ホストで実行される AOP クライアントが CSL OM と通信する方法 . . . . .	158
ワークステーションで実行される AOP クライアントが CSL OM と通信する方法 . . . . .	158
コマンド処理クライアントによる AOP コマンドの処理 . . . . .	159
CSL OM XML 出力の解釈 . . . . .	160
CSL OM 自動化操作プログラム要求 . . . . .	131
CSL OM API 要求 . . . . .	140
CSL OM 照会要求 . . . . .	152

## 第 5 章 CSL ODBM クライアントの作成 . . . . . 161

ODBM クライアント要求の順序 . . . . .	161
CSL ODBM クライアント要求 . . . . .	162
CSLDMDRG: ODBM クライアント登録解除要求 . . . . .	163
CSLDMI: ODBM アプリケーション・プログラム・インターフェース . . . . .	164
CSLDMREG: ODBM クライアント登録要求 . . . . .	177

## 第 6 章 CSL OM クライアントの作成 183

CSL OM コマンド処理クライアントの要求 . . . . .	183
CSL OMBLD: コマンド登録作成 . . . . .	183
CSL OMDRG: コマンド登録解除要求 . . . . .	186
CSL OMOUT: 非送信請求出力要求 . . . . .	187
CSL OMRDY: レディー要求 . . . . .	189
CSL OMRREG: コマンド登録要求 . . . . .	191
CSL OMRSP: コマンド応答要求 . . . . .	194
CSL OMSUB: 非送信請求メッセージへのサブスクライブ . . . . .	197
CSL OMUSB: 非送信請求メッセージからのアンサブスクライブ . . . . .	201
CSL OM ディレクティブ . . . . .	202

## 第 7 章 CSL RM クライアントの作成 207

RM クライアント要求の順序 . . . . .	207
グローバル・リソースを管理するための CSL RM 要求の発行 . . . . .	208

全 IMSplex プロセスを調整するための CSL RM 要求の発行	209
CSLRMDEL: リソースの削除	210
CSLRMDRG: クライアントを登録解除する	215
CSLRMPRI: プロセス開始	216
CSLRMPRR: プロセス応答	219
CSLRMPRS: プロセス・ステップ	221
CSLRMPRT: プロセス終了	228
CSLRMQRY: リソース照会	231
CSLRMREG: クライアントを登録する	237
CSLRMUPD: リソースを更新する	241
CSL RM ディレクティブ	247
CSL RM 構造再移植ディレクティブ	248
CSL RM 構造障害ディレクティブ	248
CSL RM プロセス・ステップ・ディレクティブ	249
CSL RM プロセス・ステップ応答ディレクティブ	250
<b>第 8 章 CSL SCI クライアントの作成</b>	<b>253</b>
CSL SCI 要求の順序	253
拡張 CSL SCI 要求	254
CSL SCI 要求	255
CSLSCBFR: バッファ戻し要求	255
CSLSCDRG: 登録解除要求	257
CSLSCMSG: メッセージ送信要求	259
CSLSCQRY: 照会要求	266
CSLSCQSC: 静止要求	270
CSLSCRDY: レディー要求	271
CSLSCREG: 登録要求	273
CSLSCRQR 要求戻し要求	280
CSLSCRQS: 要求送信	284
<b>第 9 章 CSL Operations Manager</b>	
<b>XML 出力</b>	<b>291</b>
CSLOMI XML 出力の例	291
CSLOMCMMD 出力	294
CSLOMQRY 出力	295
CSLOMOUT 出力	297
CSL OM 応答として返される XML タグ	297
<b>第 10 章 REXX SPOC API および CSL</b>	<b>307</b>
CSL OM での REXX SPOC API 環境	307
CSL 内での REXX 環境のセットアップ	307
IMSplex 環境のセットアップ	307
タイプ-2 IMS コマンドの発行	309
CSLULGTS: XML 形式のコマンド応答の取得	309
CSLULOPT: フォーマット ID のコマンド応答への組み込み	310
CSLULGTP: REXX ステム変数でのコマンド応答の直接取得	310
トランザクション内での REXX SPOC API	317
IMS SPOC 環境の終了	317
非送信請求メッセージの検索	318
CSLULSUB 要求	318
CSLULUSB 要求	318

CSLULGUM 要求	319
OM にサブスクライブする場合のサンプル・プログラム	319
REXX のサンプルと例	320
サンプル REXX SPOC プログラム	320
REXX SPOC バッチ・ジョブ例	321
/DISPLAY コマンドの例およびフォーマット ID	323
オートノミック・コンピューティングの例	323

## 第 3 部 非同期データ伝搬 . . . . . 325

### 第 11 章 変更データ・ログ・レコード 327

取り込みデータの要素	327
取り込みデータ量の削減	329
ログに記録されたデータ・要素の例	329

### 第 12 章 ジョブ終了 (EOJ) 呼び出しログ・レコード . . . . . 331

### 第 13 章 SETS および ROLS 呼び出しログ・レコード . . . . . 333

### 第 14 章 データ・キャプチャー・ログ・レコードのフォーマット . . . . . 335

データ・キャプチャー・ログ・レコードの接頭部	335
変更データ・ログ・レコードのフォーマット	335
データ・要素・ヘッダーのフォーマット	337
CAPD ブロックのフォーマット (LOGID=X'00')	338
CAPD_DATA のフォーマット (LOGID=X'0C')	341
ジョブ終了呼び出しログ・レコードのフォーマット	342
SETS および ROLS 呼び出しログ・レコードのフォーマット	342

## 第 4 部 データベース・リソース・アダプター (DRA) . . . . . 345

### 第 15 章 スレッド概念 . . . . . 347

スレッドの処理	347
マルチスレッドの処理	348
CCTL マルチスレッドの例	349

### 第 16 章 同期点 . . . . . 355

2 フェーズ・コミット・プロトコル	356
2 フェーズ同期処理における未確定状態	359

第 17 章 DRA 始動テーブル . . . . .	361
第 18 章 CCTL に対する DRA の有効化 . . . . .	365
第 19 章 ODBA インターフェースに対する DRA の有効化 . . . . .	367
第 20 章 CCTL DRA 要求の処理 . . . . .	369
第 21 章 ODBA 呼び出しの処理 . . . . .	371
第 22 章 CCTL が働きかける DRA 機能要求. . . . .	373
INIT 要求 . . . . .	373
RESYNC 要求 . . . . .	377
TERM 要求 . . . . .	378
SCHED 要求 . . . . .	379
IMS 要求. . . . .	382
SYNTERM 要求 . . . . .	383
PREP 要求 . . . . .	385
COMTERM 要求 . . . . .	386
ABTTERM 要求 . . . . .	387
TERMTHRD 要求. . . . .	388
第 23 章 DRA の終了 . . . . .	391
第 24 章 CCTL リカバリー処理の設計	393
第 25 章 CCTL のパフォーマンス: DRA スレッド TCB のモニター . . . . .	395
DRA スレッド統計 . . . . .	396
DRA 統計 . . . . .	397
DRA トレース . . . . .	398
IMS DB へのコマンド送信. . . . .	399
問題診断 . . . . .	399
<hr/>	
第 5 部 データベース・リカバリー管理 (DBRC) . . . . .	401
第 26 章 DBRC API . . . . .	403
DBRC API にアクセスするためのアプリケーションの構造 . . . . .	404
アプリケーション・プログラムによる DBRC API 環境の確立方法 . . . . .	404
アプリケーション・プログラムによる DBRC API 環境の終了方法 . . . . .	404
アドレッシング・モードと常駐モード . . . . .	404
アドレス・スペース制御 (ASC) モードおよび状態 . . . . .	405
DBRC API によるレジスターの使用方法 . . . . .	405
DBRC API アプリケーションへの等価 (EQU) ステートメントの組み込み方法 . . . . .	405
API アプリケーション . . . . .	405
DBRC API マクロのバージョン . . . . .	406

DBRC API トークン. . . . .	406
DSPAPI マクロのマクロ形式 . . . . .	407
照会出力ブロック・ヘッダー . . . . .	409
DBRC API の実行時の考慮事項 . . . . .	410
DSPAPI マクロ・アクセス . . . . .	410
RECON データ・セット・アクセス . . . . .	410
RECON アクセス権限 . . . . .	411
DBRC 要求のタイム・スタンプ・フォーマット . . . . .	412
DBRC による出力データ・セットの使用方法 . . . . .	412
照会要求の名前パラメーターに対するワイルドカード・サポート. . . . .	413
第 27 章 DBRC API セキュリティー・フィーチャー . . . . .	415
第 28 章 DBRC 許可要求 (AUTH). . . . .	417
AUTH 要求の構文 . . . . .	418
AUTH 要求のパラメーター . . . . .	418
AUTH の戻りコードと理由コード . . . . .	420
AUTH 出力ブロックの APAUB_RsnCode. . . . .	422
AUTH 出力ブロックのマッピング . . . . .	423
AUTH 出力ブロック . . . . .	424
第 29 章 DBRC コマンド要求 (COMMAND). . . . .	425
COMMAND 要求の構文 . . . . .	425
コマンド要求のパラメーター . . . . .	425
COMMAND 要求の戻りコードと理由コード . . . . .	427
コマンド出力ブロックのマッピング . . . . .	428
第 30 章 DBRC 照会要求 . . . . .	431
照会要求からの出力 . . . . .	432
バックアウト照会要求 (TYPE=BACKOUT) . . . . .	432
データベース照会要求 (TYPE=DB) . . . . .	437
DBDS 照会要求 (TYPE=DBDS) . . . . .	461
グループ照会要求 (TYPE=*GROUP). . . . .	467
ログ照会要求 (TYPE=LOG) . . . . .	477
OLDS 照会要求 (TYPE=OLDS) . . . . .	486
HALDB 区画照会要求 (TYPE=PART) . . . . .	491
RECON 状況照会要求 (TYPE=RECON) . . . . .	499
サブシステム照会要求 (TYPE=SUBSYS) . . . . .	504
第 31 章 DBRC バッファ解放要求	511
第 32 章 DBRC 開始要求 (STARTDBRC) . . . . .	515
第 33 章 DBRC 停止要求 (STOPDBRC). . . . .	521
第 34 章 DBRC 無許可要求 (UNAUTH) 525	
UNAUTH の戻りコードと理由コード . . . . .	528
UNAUTH 出力ブロック用の APAUB_RsnCode . . . . .	530
UNAUTH 出力ブロックのマッピング . . . . .	531
UNAUTH 出力ブロック . . . . .	531

---

第 6 部 リポジトリ・サーバー・  
バッチ・インターフェース  
(FRPBATCH) . . . . . 533

第 35 章 FRPBATCH でのコマンド . . . . . 535

FRPBATCH での ADD コマンド . . . . .	537
FRPBATCH での DELETE コマンド . . . . .	539
FRPBATCH での DSCHANGE コマンド . . . . .	540
FRPBATCH での LIST コマンド . . . . .	541
FRPBATCH での RENAME コマンド . . . . .	542
FRPBATCH での START コマンド . . . . .	543
FRPBATCH での STOP コマンド . . . . .	544
FRPBATCH での UPDATE コマンド . . . . .	545

---

第 7 部 VTAM および SNA の参照  
情報 . . . . . 549

第 36 章 SLU P と LU 6.1 のバイン  
ド・パラメーター . . . . . 551

金融機関通信システム・バインド・パラメーター	551
1 次ハーフセッションとしての IMS . . . . .	553
2 次ハーフセッションとしての IMS . . . . .	558

第 37 章 SLU 1 および SLU 2 のバイン  
ド・パラメーター . . . . . 563

SLU 1 バインド・パラメーター . . . . .	563
SLU 2 バインド・パラメーター . . . . .	565

第 38 章 CINIT ユーザー・データ・パ  
ラメーターの形式 . . . . . 569

第 39 章 SNA 文字ストリング制御 . . . 573

フォーマット設定制御 . . . . .	573
制御機能コード割り当て . . . . .	574

---

第 8 部 付録 . . . . . 575

特記事項 . . . . . 577

プログラミング・インターフェース情報 . . . . .	579
商標 . . . . .	579
製品資料に関するご使用条件 . . . . .	579
IBM オンライン・プライバシー・ステートメント	580

参考文献 . . . . . 583

索引 . . . . . 585

---

## 本書について

これらのトピックでは、IMS™ システ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 (API) 呼び出しについての参照情報を提供します。IMS Common Queue Server (CQS)、IMS Common Service Layer (CSL)、IMS DataPropagator for z/OS® による IMS データ伝搬、IMS Database Resource Adapter (DRA)、IMS Database Recovery Control (DBRC) API、IMS Repository Server (FRPBATCH)、および VTAM® と SNA に関する情報があります。

この情報は、IBM® Knowledge Center で参照できます。

---

## 前提知識

本書を使用するには、IMS Database Manager (DB) または IMS Transaction Manager (TM) の知識が必要です。また、z/OS およびIMS の基本概念、インストールされている IMS システムを理解しており、プロジェクト計画に関する作業の一般的な知識を持っていることが必要です。

z/OS の詳細については、IBM Knowledge Center の「z/OS basic skills」トピックを参照してください。

IMS の基本概念を理解するには、「*An Introduction to IMS*」(IBM Press 出版) をお読みになると役立ちます。

IBM では、IMS の学習に役立つような講習会や自習講座を数多く提供しています。利用可能な講習の詳しいリストについては、IBM Skills Gateway にアクセスして、IMS を検索してください。

---

## 本書で使用されるIMS 機能名

本書では、「HALDB オンライン再編成」という用語は、特に断りがない限り、IMS バージョン 13 の一部として組み込まれた HALDB オンライン再編成機能を指しています。

---

## 新規および変更された情報の識別方法

IMS ライブラリーの PDF 資料のほとんどの新規および変更された情報は、左マージン内の文字 (改訂マーカー) によって示されています。「リリース計画」、ならびに「*Program Directory*」および「*Licensed Program Specifications*」の第 1 版 (-00) には、改訂マーカーは含まれていません。

改訂マーカーは、以下の一般的な規則に従っています。

- 技術的な変更のみにマークが付けられています。形式上の変更や文法的な変更には、マークは付けられていません。

- 段落、構文図、リスト項目、操作手順、または図などの要素の一部が変更された場合、その要素の一部だけの変更であっても、要素全体に改訂マークが付けられています。
- トピックの変更が 50% を超えた場合には、そのトピック全体に改訂マークが付けられています (そのため、新規トピックではなくても、新規トピックのように見えることがあります)。

改訂マークは情報に加えられたすべての変更を示しているとは限りません。削除されたテキストとグラフィックスには、改訂マークでマークを付けることはできないためです。

---

## 構文図の読み方

本書で使用されている構文図には、以下の規則が適用されています。

- 構文図は、経路を示す線に沿って、左から右、上から下に読み取ります。以下の規則が使用されます。
  - >>--- 記号は、構文図の始まりを示します。
  - ---> 記号は、構文図が次の行に続くことを示します。
  - >--- 記号は、この構文図が直前の行から続いていることを示します。
  - --->< 記号は、構文図の終わりを示します。
- 必須項目は、水平線 (メインパス) 上に表示されます。



- オプション項目は、メインパスより下に示されます。

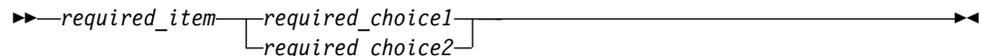


メインパスより上にオプション項目が示されている場合は、その項目が構文エレメントの実行に影響することはない、読みやすくするためのみの表記です。

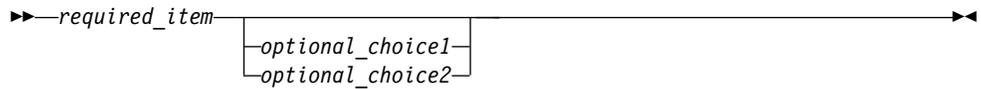


- 複数の項目から選択できる場合は、縦方向に並べて (スタック) 示されます。

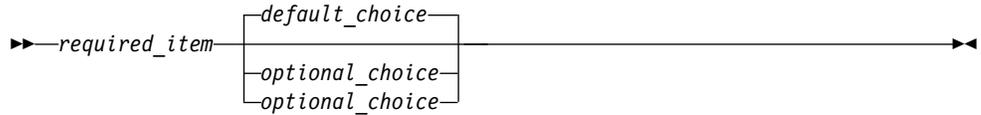
それらの項目の中から 1 つを選択する必要がある場合は、スタックの中の 1 つの項目がメインパス上に表示されます。



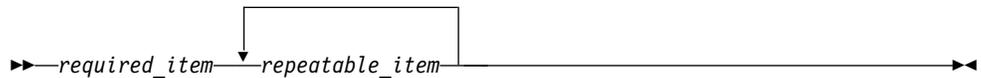
それらの項目から 1 つを選択することがオプションである場合は、スタック全体がメインパスの下に表示されます。



デフォルト項目が含まれている場合、その項目はメインパスより上に示され、他の選択項目はメインパスより下に示されます。



- メインパスの上方にある左に戻る矢印線は、項目が反復可能であることを示します。

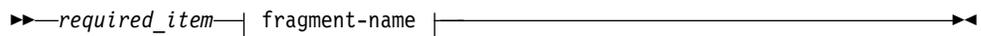


反復矢印線にコンマが含まれている場合は、反復項目をコンマで区切る必要があります。



スタック上方の反復矢印線は、スタック内の項目を反復できることを示しています。

- 1 つの構文図を複数のフラグメントに分割しなければならない場合もあります。構文フラグメントはメインの構文図とは別に示されますが、フラグメントの内容は、図のメインパス上にあるものとして読む必要があります。



#### fragment-name:



- IMS では、b 記号は、該当位置に空白が 1 つあることを示します。
- キーワード、および該当する場合はキーワードの最小の省略語は、大文字で表されます。これらは、示されているとおりに入力する必要があります。変数は、すべて小文字のイタリック文字で示されます (例えば、`column-name`)。これらは、ユーザーが指定する名前または値を表します。
- キーワードとパラメーターは、構文図で間に句読点が表示されていない場合は、少なくとも 1 つのスペースで分離します。
- 句読記号、括弧、算術演算子、およびその他の記号は、構文図で示されたとおりに入力します。

- 脚注は、例えば (1) のように、数字を括弧で囲んで示してあります。

---

## IMS バージョン 13 のアクセシビリティ機能

アクセシビリティ機能は、運動障害または視覚障害など身体に障害を持つユーザーが情報技術製品を快適に使用できるようにサポートします。

### アクセシビリティ機能

以下のリストは、IMS バージョン 13 を含む z/OS 製品の主なアクセシビリティ機能を示しています。これらの機能は、以下をサポートしています。

- キーボードのみの操作。
- スクリーン・リーダー (読み上げソフトウェア) およびスクリーン拡大鏡によって通常使用されるインターフェース。
- 色、コントラスト、フォント・サイズなど表示属性のカスタマイズ。

### キーボード・ナビゲーション

IMS バージョン 13 ISPF パネル機能には、キーボードまたはキーボード・ショートカット・キーを使用してアクセスできます。

TSO/E または ISPF を使用して IMS バージョン 13 ISPF パネルをナビゲートする詳細については、「z/OS TSO/E 入門」、「z/OS TSO/E ユーザーズ・ガイド」、および「z/OS 対話式システム生産性向上機能 (ISPF) ユーザーズ・ガイド 第 1 巻」を参照してください。上記の資料には、キーボード・ショートカットまたはファンクション・キー (PF キー) の使用方法を含む、各インターフェースのナビゲート方法が記載されています。それぞれの資料では、PF キーのデフォルトの設定値とそれらの機能の変更方法についても説明しています。

### 関連のアクセシビリティ情報

IMS バージョン 13 のオンライン資料は、IBM Knowledge Center で参照できます。

### IBM におけるアクセシビリティ

IBM のアクセシビリティに対する取り組みについて詳しくは、*IBM Human Ability and Accessibility Center* ([www.ibm.com/able](http://www.ibm.com/able)) を参照してください。

---

## 第 1 部 Common Queue Server (CQS)

ここに記載する情報から、CQS クライアントおよび CQS クライアント要求の記述方法を学べます。



---

## 第 1 章 CQS クライアントの作成

CQS クライアントは、要求を使用して CQS と通信します。CQS を使用して製品やサービスのためのキューおよびキュー構造を管理するには、1 つ以上の CQS クライアントを作成する必要があります。

Common Queue Server (CQS) クライアントを作成する際には、念頭に置いておかなければならない、さまざまな考慮事項があります。以降のトピックの内容は主に、クライアントを作成するプログラマー用ですが、CQS クライアントの設計および作成に伴ういくつかの問題を認識する必要のある CQS 管理者またはシステム・プログラマーも対象にしています。

このトピックには汎用プログラミング・インターフェース情報が含まれています。

関連概念:

15 ページの『第 2 章 CQS クライアント要求』

---

### CQS クライアント要求の要約

CQS クライアント要求により、クライアントが CQS またはカップリング・ファシリティー・リスト構造上の共用キューへアクセスできるようになります。CQS クライアント作成のための基本ツールは、CQS が提供する一組のクライアント要求マクロです。クライアントが CQS またはカップリング・ファシリティー・リスト構造上の共用キューへアクセスできるようにするには、これらの要求を使用します。

以下に、CQS 要求の要約を挙げます。

#### **CQSBRWSE**

データ・オブジェクトのコピーをキューから検索する

#### **CQSCHKPT**

内部テーブルまたは構造上のすべてのデータ・オブジェクトのチェックポイントをとる

#### **CQSCONN**

クライアントを 1 つ以上の構造へ接続する

#### **CQSDEL**

1 つ以上のデータ・オブジェクトをキューから削除する

#### **CQSDEREG**

クライアントの登録をその CQS から取り消して、そのクライアントとの通信を終了する

#### **CQSDISC**

クライアントを 1 つ以上の構造から切断する

#### **CQSINFRM**

1 つ以上のキューに対するクライアントのインタレストを登録し、そのキューに作業が存在するときはクライアントに通知する

## CQSMOVE

1 つ以上のデータ・オブジェクトをあるキューから別のキューへ移動する

## CQSPUT

データ・オブジェクトをキュー上に置く

## CQSQUERY

キューまたは構造に関する情報を要求する

## CQSREAD

キューからのデータ・オブジェクトのコピーを検索およびロックする

## CQSRECVR

クライアントまたは CQS コールド・スタートの後にコールド・キューへ移動されたデータ・オブジェクトをリカバリーする

## CQSREG

CQS にクライアントを登録し、通信を確立する

## CQSRSYNC

障害発生後にクライアントとその CQS の間の未確定データの再同期を行う

## CQSSHUT

CQS をシャットダウンする

## CQSUNLCK

データ・オブジェクトをアンロックして、どのクライアントも使用できるようにする

## CQSUPD

リソース構造上にある、一意的な名前の付いた、1 つ以上のリソースを更新する

---

## クライアントの出す、キュー構造に対する CQS 要求のシーケンス

クライアントは、CQS 要求を使用して CQS のサービスおよびリソースを使用します。CQS サービスに対するクライアント要求は、この表に概略する特定の順序で行う必要があります。

クライアントが CQS サービスを要求するには、特定の要求を発行する必要があります。これらの要求の一部は、特定の順序で発行しなければなりません。以下の表に、CQS 要求の順序を示します。そのほかの要求は、クライアントの処理要件に基づいて、複数回、どのような順序でも出すことができます。

表 1. CQS 要求のシーケンス

注文	要求	要求の使用目的
1	CQSREG	CQS との通信を確立する。
2	CQSCONN	個々の構造へ接続する。
3	CQSRSYNC	CQS での未確定の作業を解決する。
4	CQSRECVR <sup>1</sup>	CQS コールド・スタート後に特定のデータ・オブジェクトをリカバリーする。
5	CQSINFRM	特定のキュー名に関するインタレストを登録する。

表 1. CQS 要求のシーケンス (続き)

注文	要求	要求の使用目的
6	その他の CQS 要求	作業を処理する。その他の要求の例としては、CQSBrowse、CQSPUT、および CQSREAD があります。
7	CQSDISC	構造から切断する。
8	CQSSHUT	CQS にシャットダウンを要求する。クライアントでは、構造から切断して CQS シャットダウンを要求するために、CQSSHUT 要求のみを出す代わりに、CQSDISC ... CQSSHUT=YES を使用することもできます。
9	CQSDEREG	CQS との通信を終了する。

注:

1. クライアントは、CQSRECVR 要求と CQSINFRM 要求を、CQSRSYNC 要求の後であれば、いつでもどのような順序でも発行することができます。ただし、クライアントは、CQS での実際の作業を開始する前に、これらの要求を両方とも出す必要があります。

## CQS 要求をコーディングする際の考慮事項

CQS 要求には、さまざまなキーワード、パラメーター、および変数を使用できます。クライアントの状態として選択するインターフェースによって、その後のすべての CQS 要求、および CQS が駆動するすべてのクライアント出口ルーチンに使用できる環境が決まります。

各要求の使用法に関するトピックでは、CQS 要求のそれぞれのキーワード、パラメーター、および変数について詳しく説明していますが、すべての要求に適用される全体的な使用上の考慮事項は少数です。

### CQS における許可

CQS には、そのクライアントに関する 2 つのインターフェース、許可インターフェースと非許可インターフェースが用意されています。クライアントが CQSREG 要求を出すと、CQS は、自動的にそのクライアントの状態に基づいて正しいインターフェース環境を選択および初期化します。クライアントが許可されている場合 (PSW キー 0 から 7 の監視プログラム状態)、CQS は許可インターフェース環境を初期化します。クライアントが許可されていない場合 (キー 8 以上の問題プログラム状態)、CQS は非許可インターフェース環境を初期化します。

CQS がどのインターフェースをクライアントに割り当てるかによって、その後のすべての CQS 要求および CQS によって駆動されるすべてのクライアント出口ルーチンに許可される環境が決まります。一般に、クライアントが CQS 要求を行うとき、その PSW 状態とキーは、CQSREG 要求を出したときと同じでなければなりません。

### CQS 要求によるレジスターの使用法

すべての CQS 要求は、作業レジスターとして R0、R1、R14、および R15 のレジスターを使用することができます。CQS 要求が呼び出し元に制御を戻すとき、これらのレジスターの内容はマクロが呼び出される前とは異なっています。R15 には戻

りコードが含まれ、R0 には CQS インターフェースからの理由コードが含まれています。レジスター R2 から R13 の内容は、特定の要求の出力パラメーターとして指定されたレジスターを除いて、CQS 要求の後もそのまま変更されません。

すべての CQS 要求は、レジスター R13 が標準 72 バイト保管域を指すことを要求しています。他のレジスターは、CQS 要求が出されたときに特定の値を含んでいる必要はありません (特定の要求の入力パラメーターに指定されたレジスターを除く)。

## CQS 要求用のパラメーターのコーディング

リテラルでないすべてのパラメーター (構文図で、例えば *parameter* のように示す) の場合、CQS はアドレスか値のどちらかを期待しています。例えば、CQSREAD 要求に *cqstoken* があると、CQS は 16 バイト CQS トークンのアドレスを期待しますが、*buffersize* の場合、CQS は 4 バイトのバッファー・サイズを期待します。

アドレスまたはパラメーターの値を CQS へ渡すための CQS 要求のパラメーターは、次に示す 3 通りの方法の 1 つでコーディングすることができます。

### 1. レジスターを使用する

レジスターを使用するには、アドレスまたはパラメーターの値を汎用レジスターの 1 つにロードして、そのレジスター (括弧で囲まれている) を CQS 要求でパラメーターに使用する必要があります。

図 1. 登録のアドレスの引き渡し

```
LA      5,TOKEN
CQSREAD FUNC=READ,CQSTOKEN=(5),...
:
TOKEN   DS      XL16
```

図 2. 登録のための値の引き渡し

```
L       4,MYBUFLEN
CQSREAD FUNC=READ,BUFSIZE=(4),...
:
MYBUFLEN DC    F'00000024'
```

### 2. シンボルを使用する

シンボル名を使用するには、アドレスまたはパラメーターの値を含むシンボルを定義した上で、そのシンボルを CQS 要求のパラメーターとして使用する必要があります。

図 3. シンボルのアドレスの引き渡し

```
CQSREAD FUNC=READ,CQSTOKEN=TOKENADR,...
:
TOKENADR DC    A(TOKEN) TOKEN   DS      XL16
```

図 4. シンボルの値の引き渡し

```
CQSREAD FUNC=READ,BUFSIZE=MYBUFLEN,...  
:  
MYBUFLEN DC F'00000024'
```

### 3. シンボル値を使用する

シンボル値を使用するには、パラメーター値を含むシンボルまたは EQU を定義した上で、そのシンボル (アットマーク (@) を前に付け、括弧で囲む) を CQS 要求内でパラメーターに使用します。

図 5. シンボル値の値の引き渡し

```
CQSREAD FUNC=READ,CQSTOKEN=@(TOKEN),...  
:  
TOKEN DC XL16'0000A765B55CFF00'
```

図 6. シンボル値と等価のもの引き渡し

```
CQSREAD FUNC=READ,BUFSIZE=@(MYBUFLEN),...  
:  
MYBUFLEN EQU 24
```

## CQS 要求のためのリテラル

多くの CQS 要求マクロには、リテラルを使用するパラメーターがあります (例: CQSREAD 要求マクロの LOCAL パラメーター)。マクロ呼び出しでは、リテラル・パラメーターの組み合わせまたは OPTWORD1 パラメーターを使用して、リテラルを表す フラグを含む 4 バイトを渡すことができます。OPTWORD1 パラメーターを使用する場合、各要求マクロの DSECT 機能を使用することによって、リテラル EQU を入手することができます。リテラル値を表す EQU は、通常のストレージの場所に一緒に追加されます。

要件: 1 つのマクロ呼び出しで使用できるのは、リテラル・パラメーターか OPTWORD1 パラメーターのどちらか一方です。両方を使用することはできません。マクロ呼び出しに OPTWORD1 パラメーターが含まれている場合、このパラメーターで渡される値は、マクロがサポートするリテラル・パラメーターごとに 1 つの EQU を含んでいる必要があります。例えば、CQSREAD 要求に LOCAL、PARTIAL、および QPOS の 3 つのリテラル・パラメーターがあります。したがって OPTWORD1 パラメーターで渡す値には、LOCAL パラメーター用に 1 つの EQU、PARTIAL パラメーター用に 1 つの EQU、そして QPOS パラメーター用に 1 つの EQU を含んでいる必要があります。

一連のリテラル・パラメーターを使用して CQSREAD 要求をコーディングするには、CQSREAD FUNC=READ,...,QPOS=FIRST,LOCAL=YES... を使用します。

## OPTWORD1 パラメーターを使用した CQSREAD のコーディング

同じ CQSREAD 要求を、OPTWORD1 パラメーターを使用してコーディングするには、以下に記載する例を使用してください。

```
      L      R2,=A(CQSREAD_QPOSF+CQSREAD_LCLY+CQSREAD_PRTLY)
CQSREAD FUNC=READ,...,OPTWORD1=(R2),...
      .
      .
      .
CQSREAD FUNC=DSECT  GENERATE CQSREAD EQUs
```

## CQS 要求でのイベント制御ブロック

一部の要求では、z/OS イベント制御ブロック (ECB) を使用することができます。ECB (ECB=*ecbaddress*) を指定すると、クライアントは要求を発行した後にすぐ制御を受け取りますが、時として、要求が ECB を通知するのを必ず待たなければならない場合があります。ECB を指定しないと、CQS は要求の処理が完了するまでクライアントに制御を戻しません。

## CQS 要求内のリスト

一部の CQS 要求には LIST キーワードがあり、これはパラメーター・リスト項目のアドレスを指定するものです。このキーワードは、最初のリスト項目のアドレスを指定します。複数のリスト項目を渡したい場合は、それらがすべて連続するストレージに常駐している、すなわち、次の項目が現行項目の後に続く最初のバイトから始まるようにする必要があります。すべてのリストが、ワードまたはフルワード境界に位置合わせされていない場合でも、連続していることが必要です。

## CQS 要求を含んだプログラムのアセンブル

CQS 要求マクロは IMS と共に出荷され、IMS.ADFSMAC データ・セットの中に組み込まれています。CQS 要求マクロが含まれているプログラムをアセンブルするときは、アセンブラーにこのデータ・セット内でマクロを探すように指示する必要があります。IMS データ・セットから別のデータ・セットに、必要に応じてメンバーをコピーすることもできます。

CQS 要求を含むプログラムのリンク・エディットに特別な要件はありませんが、IMS.SDFSRESL データ・セットがクライアント・ジョブのための JOB または STEPLIB DD ステートメントと確実に連結されるようにする必要があります。

## IMS.SDFSRESL を連結するための STEPLIB DD ステートメント

IMS.SDFSRESL データ・セットを MYPROGS.SDFSRESL データ・セットの後に連結するには、以下の例に示すように、STEPLIB DD ステートメントをコーディングします。

```
STEPLIB DD DSN=MYPROGS.SDFSRESL,DISP=SHR
           DSN=IMS.SDFSRESL,DISP=SHR
```

関連概念:

15 ページの『第 2 章 CQS クライアント要求』

関連資料:

## CQS における環境要件

環境要件は、CQSREG 要求および CQSDEREG 要求以外の CQS 要求のクライアントに割り当てられた CQS インターフェースによって異なります。

以下の表に、許可 CQS インターフェースを使用しているクライアント用の環境を説明します。

表 2. 許可インターフェースを使用する CQS 要求 (CQSREG および CQSDEREG を除く) の場合の環境

環境	状態
許可	監視プログラム状態および PSW キー 0 ~ 7 (PSW キーは、CQSREG 要求が出されたときの PSW キーと一致している必要がある)
ディスパッチ可能単位モード	タスク
仮想記憶間モード	任意、ただし、CQSREG 要求が出された 1 次アドレス・スペースと PASN が同じであることが必要
AMODE	31
ASC モード	1 次
ホーム・アドレス・スペース	任意
ロック	ロックは保持されない
割り込み状況	割り込み可能
制御パラメーター	1 次アドレス・スペース内

以下の表に、非許可 CQS インターフェースを使用しているクライアント用の環境を説明します。

表 3. 非許可インターフェースを使用する CQS 要求 (CQSREG および CQSDEREG を除く) の場合の環境

環境アспект	状態
許可	問題プログラム状態または PSW キー 8 (PSW キーは、CQSREG 要求が出されたときの PSW キーと一致している必要がある)
ディスパッチ可能単位モード	タスク
仮想記憶間モード	なし (PASN=SASN=HASN)
AMODE	31
ASC モード	1 次
ホーム・アドレス・スペース	CQSREG が出されたアドレス・スペース
ロック	ロックは保持されない
割り込み状況	割り込み可能
制御パラメーター	1 次アドレス・スペース内

CQS の登録と登録解除の要求 (CQSREG および CQSDEREG を除く) の環境要件は、他のすべての CQS 要求とは異なります。許可クライアントは、以下の表に説明する環境で CQSREG 要求および CQSDEREG 要求を発行する必要があります。

表 4. 許可インターフェースを使用する CQSREG および CQSDEREG 要求の環境

環境アспект	状態
許可	監視プログラム状態および PSW キー 0 から 7
ディスパッチ可能単位モード	タスク
仮想記憶間モード	なし (PASN=SASN=HASN)
AMODE	31
ASC モード	1 次
ロック	ロックは保持されない
割り込み状況	割り込み可能
制御パラメーター	1 次アドレス・スペース内

非許可クライアントは、以下の表に説明する環境で CQSREG 要求および CQSDEREG 要求を発行する必要があります。

表 5. 非許可インターフェースを使用する CQSREG および CQSDEREG 要求の環境

環境アспект	状態
許可	問題プログラム状態または PSW キー 8
ディスパッチ可能単位モード	タスク
仮想記憶間モード	なし (PASN=SASN=HASN)
AMODE	31
ASC モード	1 次
ロック	ロックは保持されない
割り込み状況	割り込み可能
制御パラメーター	1 次アドレス・スペース内

## CQS 要求の戻りコードと理由コード

CQS の戻りコードおよび理由コードは、CQS アドレス・スペースへの要求送信の成功または失敗を示し、行われている特定の CQS 要求の失敗または成功を反映します。

CQSREG と CQSDEREG を除いて、各 CQS 要求は 2 組の戻りコードと理由コードを戻します。1 組は CQS インターフェースによって戻されて、CQS アドレス・スペースへの要求の送信の成功または失敗を示します (これらのコードは R15 および R0 で戻されます)。もう 1 組は CQS アドレス・スペースによって戻され、行われようとしている特定の CQS 要求の成功または失敗を反映します (これらは CQS 要求マクロの RETCODE および RSNCODE パラメーターの示すフィールドに入れて戻されます)。

CQS 要求を行うとき、その要求は、クライアント・アドレス・スペースから CQS アドレス・スペースまで CQS インターフェースを通して移動する必要があります。CQS インターフェースは、レジスター R15 と R0 に要求の送信の成功または

失敗に関する情報を戻します。CQS 要求マクロを出した後は、ユーザー・コードでまず R15 の値をチェックしてください。R15 の値がゼロであれば、要求は CQS インターフェースにより正常に CQS アドレス・スペースに送信されています。R15 がゼロでない場合、CQS インターフェースは CQS アドレス・スペースに要求を送信することができなかつたので、R0 にエラーを説明する理由コードが入っています。

CQS 要求自体からの戻りコードと理由コードは、CQS 要求マクロにコーディングされた RETCODE および RSNCODE パラメーターで指定されたフィールドに戻されます。これらのフィールドに戻される値は、CQS インターフェースの戻りコード (R15) がゼロの場合のみ有効です。CQS 要求マクロを出した後で、R15 のインターフェース戻りコードがゼロでない場合、RETCODE および RSNCODE フィールドの値は予測可能でないで、これらの値は使用しないでください。

同期要求 (すなわち、ECB パラメーターがコーディングされていない要求) の場合、ユーザー・モジュールが要求マクロから制御を戻された後で RETCODE および RSNCODE フィールドがセットされて、すぐに使用できるようになります。非同期要求 (すなわち、ECB パラメーターがコーディングされている要求) の場合、RETCODE および RSNCODE フィールドは、ECB が CQS によって POST (通知) された後にのみセットされます。RETCODE フィールドと RSNCODE フィールドは、要求で指定した ECB で WAIT を出し、その WAIT が戻るまでチェックしないでください。

CQSREG 要求と CQSDEREG 要求は、この場合の例外です。CQSREG と CQSDEREG は、CQS インターフェースでクライアントを登録または登録解除しますが、実際にインターフェースを通して CQS アドレス・スペースへ要求を送信するわけではありません。CQSREG と CQSDEREG には 1 組の戻りコードと理由コードがあるだけで、これらが登録要求または登録解除要求から戻るとすぐに使用可能になります。戻りコードは、レジスター 15 と要求マクロの RETCODE で指定されたフィールドの両方にセットされます。理由コードは、レジスター 0 と要求マクロの RSNCODE で指定されたフィールドの両方にセットされます。

CQS インターフェースは、以下の表に示されている戻りコードと理由コードを発行します。どの CQS 要求でも、これらの戻りコードと理由コードを受け取ることができます。CQS インターフェースは非許可クライアントに対してより広範囲に渡る検査を実行するので、クライアントが非許可クライアントである場合、以下の戻りおよび理由コードの一部については、受信のみ可能です。

表 6. CQS インターフェースによって検出されたエラーの戻りコードと理由コード

戻りコード	理由コード	意味
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000214'	<i>connecttoken</i> が無効。
X'00000010'	X'00000430'	CQS アドレス・スペースが利用不能。
X'00000014'	X'00000600'	CQS インターフェースが内部ブロックにアクセスできない。
X'00000014'	X'00000604'	クライアントが問題プログラム状態で実行しているか、または誤った PSW キーを使用している。
X'00000014'	X'00000608'	クライアントが CQS インターフェースに無効な機能コードを渡した。

表 6. CQS インターフェースによって検出されたエラーの戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'00000014'	X'0000060C'	クライアントが無効な CQS 要求タイプを指定した。
X'00000014'	X'00000610'	CQS が要求パラメーターをコピーするためのストレージを割り振ることができなかった。
X'00000014'	X'00000614'	渡された全要求パラメーターの全長が全パラメーター長の合計より小さい。
X'00000014'	X'00000618'	全パラメーターの合計長に関してインターフェースに渡された値がゼロまたは負。
X'00000014'	X'0000061C'	合計パラメーター数に関してインターフェースに渡された値がゼロまたは負。
X'00000014'	X'00000620'	要求のパラメーターの 1 つの長さが負。
X'00000014'	X'00000624'	構造呼び出しパラメーター・リストに関して渡された長さが無効。
X'00000014'	X'00000628'	無効要求機能コード。
X'00000014'	X'0000062C'	無効要求パラメーター・リスト・バージョン番号。
X'00000014'	X'00000630'	誤ったパラメーターの番号が要求された機能に対して渡された。
X'00000014'	X'00000634'	パラメーターが誤った長さとともに渡された。
X'00000014'	X'00000638'	パラメーターが、アドレスではなく値によって渡された。
X'00000014'	X'0000063C'	パラメーターが、値ではなくアドレスによって渡された。
X'00000014'	X'00000640'	CQS 要求は、CQS へ送信される前に異常終了した。
X'00000014'	X'00000644'	CQS が要求パラメーターをコピーしているときに、CQS 要求が異常終了した。このエラーは、通常、クライアントが誤ったパラメーター・データを渡したために起こります。
X'00000014'	X'00000648'	CQS 要求マクロによってパスされたインターフェース・パラメーター・リスト・バージョンが有効ではなかった。このエラーは、CQS クライアントおよびクライアントが使用しようとしている CQS アドレス・スペース間のバージョンの差が原因と考えられます。

すべての CQS 要求には、要求のすべての戻りコードおよび理由コードのために、プログラム内に EQU ステートメントを組み込むための DSECT 機能があります。

推奨事項: すべての CQS 要求に対して FUNC=DSECT を指定するプログラムを作成して、戻りコード値と理由コード値に使用できる記号変数名を判別できるようにしてください。

## CQS クライアントおよび特殊イベントの処理

CQS クライアントは、多種多様なイベントを開始するか、またはそれに参加できなければなりません。これらのイベントを適切に処理するためには、CQS クライアントが処理可能なイベントを認識している必要があります。

CQS クライアントは、多くのさまざまなタイプのイベントを開始するかまたはそれに参加できなければなりません。このトピックでは、これらの特殊イベントの一部について説明するとともに、CQS クライアントはこれらのイベントに対して何ができるか、あるいは何をしなければならないかを説明します。

## CQS コールド・スタート

データを含む構造に接続した後、CQS は、コールド・スタートすると CQSMOVE または CQSDEL 要求から未解決の作業を探します。CQS は、CQSMOVE 要求をバックアウトし、CQSDEL 要求を完了します。その後、CQS がシステム・チェックポイントを実行すると、再始動は完了します。

CQS は、CQSREAD 要求を使用して開始された作業は解決しません。その結果、データ・オブジェクトがキューに残される場合があります。クライアントは、CQSRSYNC 要求を出して、CQS がこれらのデータ・オブジェクトをコールド・キューに移動し、クライアントにその存在を知らせるようにさせることができます。クライアントは、その後で CQSRECVR 要求を出して、このデータ・オブジェクトにアクセスすることができます。

推奨事項: CQS は上記のデータ・オブジェクトを認識しないため、CQSPUT 要求を使用して開始されたすべての作業を完了してください。

## キューに関するインタレストの CQSINFRM への登録

キューにデータ・オブジェクトが存在するとき、またはキューが空でなくなったときに、CQS がクライアントに通知できるようにするには、CQSINFRM 要求を使用します。クライアントがキュー上の作業の存在の通知を受けるには、前もってそのキューのインタレストを登録しておく必要があります。

## CQS 要求の使用によるコールド・キュー上のオブジェクトの処理

アクティブ構造にオブジェクトがあるときに CQS またはクライアントがコールド・スタートされると、CQS はオブジェクトをコールド・キューに入れます。クライアントは、CQSBRWSE 要求を使用してコールド・キュー上のオブジェクトを調べ、その上で、この要求によって戻されるコールド・キュー・トークンと UOW を使用し、CQSRECVR 要求を使用してコールド・キューからオブジェクトを検索または削除することができます。

CQS クライアントを作成するときは、以下の要求を使用することによって、コールド・キュー内の各オブジェクトに関する情報 (例えば、qname、データ・オブジェクト・カウント、最も古いデータ・オブジェクトのタイム・スタンプ、最も新しいデータ・オブジェクトのタイム・スタンプ) を取得することができます。

```
CQSQUERY FUNC=QTYPE,QTYPENM=COLDQ
```

## CQS 要求の使用によるチェックポイントの開始

CQS クライアントは、CQSCHKPT FUNC=CHKPTSYS 要求を出すことによって、システム・チェックポイントを開始することができます。CQS クライアントは、CQSCHKPT FUNC=CHKPTSTR 要求を出すことによって、構造チェックポイントを開始することができます。

## CQS のシャットダウン

CQS をシャットダウンする場合、クライアントは CQSSHUT 要求または CQSDISC 要求に CQSSHUT=YES を指定して出すことができます。どちらの場合も、構造接続がなくなった時点で CQS が終了します。CQS は、進行中の作業が完了できるように、入力と出力の要求の受け入れを続けます。構造チェックポイントは出すことが認められます。新しい接続は、CQSDISC 要求が CQSSHUT=YES と指定して出される場合は許可されますが、CQSSHUT 要求が出される場合は許可されません。

## CQS パフォーマンス改善のためのチューニング

CQSQUERY、CQSDEL、および CQSINFRM 要求で使用するパラメーターを注意深く選ぶことで CQS のパフォーマンスを改善することができます。

関連概念:

 CQS の管理 (システム管理)

関連資料:

67 ページの『CQSQUERY 要求』

38 ページの『CQSDEL 要求』

49 ページの『CQSINFRM 要求』

## 第 2 章 CQS クライアント要求

CQS クライアントは、CQS 要求と呼ばれるいくつかのアセンブラー・マクロからなる汎用インターフェースを使用して、CQS アドレス・スペースと通信します。CQS クライアントはこれらの要求を使用して、CQS と通信するとともに、共用カップリング・ファシリティ構造上のクライアント・データを操作します。CQS クライアントを作成または保守するには、これらの要求を使用できます。

IMS 制御領域などの IBM 提供のクライアントを使用している場合は、これらの要求を使用する必要はありません。

CQS 要求の一部はワイルドカード・パラメーターをサポートします。ワイルドカード・パラメーターによって、名前がワイルドカード・パラメーター・マスクと一致する複数のリソースを指定することができます。ワイルドカード・パラメーターのサイズは、1 文字から、そのリソースに対してサポートされている文字の最大数まで可能です。英数字名には、1 つ以上の専用文字およびアスタリスクまたは % 記号を含めることができます。アスタリスクを、0 個、1 個または複数の文字で置き換えて、有効リソース名を作成することができます。% 記号を 1 個の文字で置き換えて、有効リソース名を作成することができます。ワイルドカード・パラメーター・アスタリスク (\*) は「すべて」を意味しています。ただし、インストール・システムによっては、その他のワイルドカード・パラメーターがすべてを意味する場合があります。例えば、ワイルドカード・パラメーター %%%% は、リソース名がすべて 4 文字の長さをもつインストールに対しては、「すべて」を意味します。

このトピックには汎用プログラミング・インターフェース情報が含まれています。

### CQS 要求の使用例: CQSREAD

以下の例に、クライアント・サブシステムで CQSREAD 要求を使用する方法を示します。

```
*****
* FUNCTION:  USE CQSREAD REQUEST TO RETRIEVE A MESSAGE FROM SHARED *
* QUEUES. *
* *
* THE CALLER OF THIS MODULE PASSES THE ADDRESS AND SIZE OF *
* A BUFFER. IF THIS MODULE ENDS WITH RC=0, THAT BUFFER *
* HOLDS THE DATA OBJECT OR PARTIAL DATA. IF THIS MODULE *
* ENDS WITH A NON-ZERO RC, THE BUFFER'S CONTENTS ARE *
* UNPREDICTABLE. *
* *
* REGISTERS ON ENTRY: *
* *
* R2 - READ OBJECT BUFFER ADDRESS (BUFFER TO READ OBJECT INTO) *
* R3 - SIZE OF READ OBJECT BUFFER *
* R4 - CQS REGISTRATION TOKEN ADDRESS *
* R5 - CQS CONNECT TOKEN ADDRESS *
* R9 - ECB ADDRESS *
* R13 - SAVE AREA ADDRESS *
* R14 - RETURN ADDRESS *
* R15 - GETDOBJ ENTRY POINT ADDRESS *
* *
* REGISTERS DURING EXECUTION: *
* *
* R0 - WORK REGISTER *
* R1 - WORK REGISTER *
* R2 - CQSREAD PARMLIST AREA ADDRESS *
* R3 - WORK REGISTER *
* R4 - WORK REGISTER *
* R5 - WORK REGISTER *
* R6 - WORK REGISTER *
* R7 - WORK REGISTER *
* R8 - WORK REGISTER *
* R9 - ECB ADDRESS *
* R10 - WORK REGISTER *
* R11 - WORK REGISTER *
* R12 - BASE REGISTER *
```

```

* R13 - SAVE AREA ADDRESS
* R14 - WORK REGISTER
* R15 - WORK REGISTER
*
* MACROS REFERENCED:
* WAIT
* CQSREAD
*
* RETURN CODES:
* R15 - RETURN CODE
* X'00' CQSREAD SUCCESSFUL/PARTIAL DATA RETURNED
* X'08' INTERFACE PROBLEM
* X'0C' NO MESSAGE FOR QNAME
* X'10' REQUEST IS UNSUCCESSFUL, UNEXPECTED RETURN OR REASON
* CODE
*
*****
STM R14,R12,12(R13) SAVE THE REGS
LR R12,R15 R12 = PROGRAM BASE REGISTER
USING GETDOBJ,R12 GETDOBJ CSECT
LA R14,SAVEAREA CHAIN SAVE AREAS
ST R13,4(R14) THIS SAVEAREA BACKWARD PTR
ST R14,8(R13) LAST SAVEAREA FORWARD PTR
LA R13,SAVEAREA THIS ROUTINE'S SAVEAREA
ST R2,RDRBUFA SAVE A(BUFFER TO READ INTO)
ST R3,RDRBUFSZ SAVE READ BUFFER SIZE
MVC RDRRQTK,0(R4) SAVE CQS REGISTRATION TOKEN
MVC RDRCONTK,0(R5) SAVE CQS CONNECT TOKEN
ST R9,RDRECBA SAVE A(ECB)
LA R2,RDRPARAM LOAD A(PARAMETER AREA) INTO R2
XC RDRLCKTK,RDRLCKTK LOCKTOKEN=0 FOR FIRST CQSREAD
XC 0(4,R9),0(R9) CLEAR CALLER'S ECB

****
* RETRIEVE RECORD FROM IMS SHARED QUEUES
****
CQSREAD FUNC=READ, X
CQSTOKEN=@(RDRRQTK), A(REGISTRATION TOKEN) X
PARM=(R2), A(CQSREAD PARMLIST AREA) X
CONTOKEN=@(RDRCONTK), A(CONNECT TOKEN) X
ECB=RDRECBA, A(ECB) X
LCKTOKEN=@(RDRLCKTK), A(LOCK TOKEN) - RETURNED X
UOW=@(RDRUOW), A(UOW) - RETURNED X
LOCAL=NO, READ OBJECT FROM SHARED QUEUE X
QNAME=@(RDRQNAME), A(QUEUE NAME) X
QPOS=FIRST, READ FIRST OBJECT ON QUEUE X
OBSIZE=@(RDROBJSZ), A(DATA OBJECT SIZE) - RETURNED X
RSNCODE=@(RDRRSN), A(REASON CODE) - RETURNED X
RETCODE=@(RDRRC), A(RETURN CODE) - RETURNED X
BUFFER=RDRBUFA, A(CLIENT'S READ BUFFER) X
BUFSIZE=@(RDRBUFSZ) CLIENT'S READ BUFFER SIZE

LTR R15,R15 TEST RETURN CODE FROM CQS INTERFACE
BZ CHECKRC ZERO - CQSREAD OK
* LA R15,RC08 OTHER - RETURN R0, R15 IN PARM LIST
B GOEXIT CQS INTERFACE PROBLEM
RETURN TO CALLER

****
* CHECK CQSREAD RETURN CODE
****
CHECKRC DS 0H
WAIT ECB=(R9) WAIT FOR CQSREAD TO COMPLETE

L R15,RDRRC RETURN CODE
LTR R15,R15 CQSREAD REQUEST SUCCESSFUL?
BZ GOEXIT YES - RETURN TO CALLER****
* CHECK FOR CQS WARNING RETURN CODE
****
CLC RDRRC,=AL4(RQRWARN) CQSREAD WARNING?
BNE UNEXPECT NO - SET RC AND RETURN TO CALLER

****
* CQSREAD: WARNING RETURN CODE - CHECK WARNING REASON CODE
* CHECK FOR DATA OBJECT
****
CLC RDRRSN,=AL4(RRDNOOBJ) NO DATA OBJECT?
BNE PARTIAL NO, CHECK NEXT REASON CODE
LA R15,RC0C SET NO DATA OBJECT RETURN CODE
B GOEXIT RETURN TO CALLER

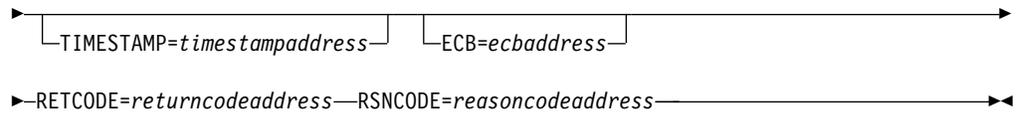
****
* CHECK PARTIAL DATA RETURNED
* PARTIAL DATA RETURNED - RETURN DATA OBJECT - RETURN CODE 0
****
PARTIAL DS 0H
CLC RDRRSN,=AL4(RRDPARTL) PARTIAL DATA RETURNED?
BNE UNEXPECT NO - SET RC AND RETURN TO CALLER
LA R15,RC00 SET RETURN CODE
B GOEXIT RETURN TO CALLER

****
* UNEXPECTED RETURN OR REASON CODE
****
UNEXPECT DS 0H
LA R15,RC10 UNEXPECTED RETURN OR REASON CODE
B GOEXIT RETURN TO CALLER

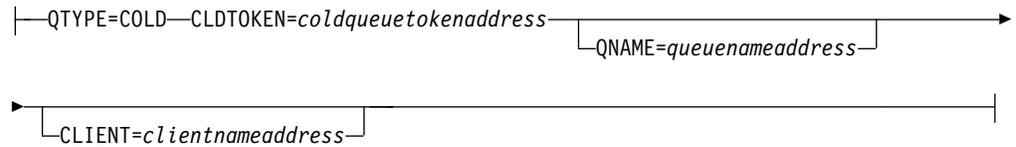
*****
* STANDARD EXIT
*****
GOEXIT DS 0H
L 13,4(13) GET PREVIOUS SAVE LEVEL
L 14,12(13) A(RETURN-TO-CALLER)
LM 0,12,20(13) RESTORE REGS
OI 15(13),X'01' SET RETURN FLAG IN CALLER SAVE AREA
BR 14 RETURN TO CALLER

```



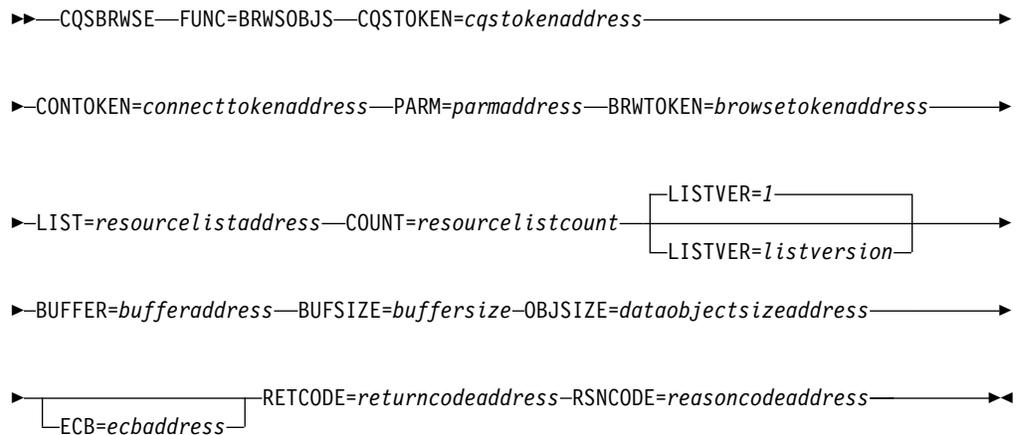


**A:**



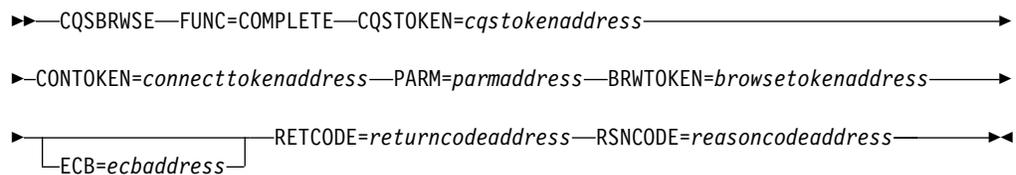
### BRWSOBSJ 機能

CQSBRWSE 要求の BRWSOBSJ 機能を使用して、リソース構造からの特定タイプのリソース・データ・オブジェクトを 1 つ以上ブラウズしてください。



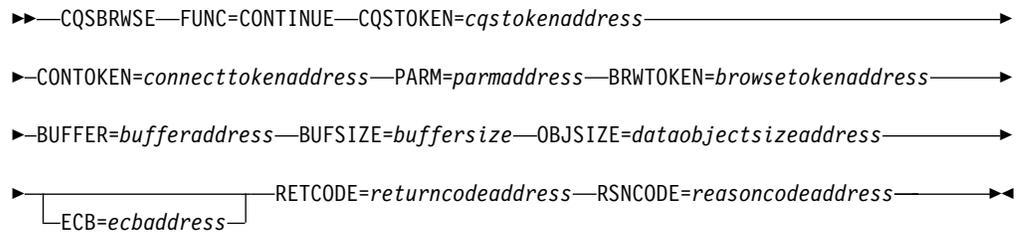
### COMPLETE 機能

CQS に対して、特定のブラウズ・トークンに関する CQSBRWSE 要求が完了したことを示すには、CQSBRWSE 要求の COMPLETE 機能を使用します。



### CQSBRWSE の CONTINUE 機能

前の CQSBRWSE 要求で部分的なデータが検索されており、データ・オブジェクトの残りの部分を検索したい場合は、CQSBRWSE 要求の CONTINUE 機能を使用します。



## DSECT 機能

CQSBRWSE パラメーター・リストの長さ、および CQSBRWSE の戻りコードと理由コードのために、プログラムの中に等価 (EQU) ステートメントを組み込むには、CQSBRWSE 要求の DSECT 機能を使用します。



## 使用上の注意

CQSBRWSE FUNC=BROWSE 要求は、キュー構造上の特定のキューからのデータ・オブジェクトのコピーを検索します。最初の CQSBRWSE FUNC=BROWSE 要求は、選択基準を満たすデータ・オブジェクトのスナップショットを実行し、最初のデータ・オブジェクトのコピーを返します。データ・オブジェクトは、削除もロックもされていません。および以降の CQS 要求によってアクセスすることが可能です。以降の CQSBRWSE FUNC=BROWSE 要求はそれぞれ、次のデータ・オブジェクトのコピーを検索します。データ・オブジェクトは、CQSBRWSE 要求で指定されたクライアント・バッファに返されます。データ・オブジェクトのサイズはクライアントに渡されます。

ブラウズ・トークンは、ブラウズされているデータ・オブジェクトのカーソル位置を維持します。ゼロのブラウズ・トークンを指定した CQSBRWSE FUNC=BROWSE 要求は、最初のデータ・オブジェクトを返します。ゼロ以外のブラウズ・トークンを指定した CQSBRWSE FUNC=BROWSE 要求は、そのブラウズ・トークンと関連付けられたキュー上の次のデータ・オブジェクトのコピーを取得します。返されたデータ・オブジェクトがキュー上の最後のデータ・オブジェクトである場合、CQS はブラウズ・トークンを無効にして、そのブラウズ・トークンに関連したデータ構造をすべて解放します。

CQSBRWSE FUNC=BROWSE 要求が発行され、渡されたバッファのサイズが次のデータ・オブジェクトを保持するには足りない場合には、データの一部が返されます。バッファは、可能な限りフィットするだけのデータ・オブジェクトで充てんされています。CQSBRWSE FUNC=CONTINUE 要求は、データ・オブジェクトの残りを検索します。

CQSBRWSE FUNC=BRWSOBS 要求は、1 つ以上のデータ・オブジェクトに関する情報をリソース構造から取得します。最初の CQSBRWSE FUNC=BRWSOBS 要求は、選択基準を満たすデータ・オブジェクトのスナップショットを取り、これらのデータ・オブジェクトのうち 1 つ以上のオブジェクトに関する情報を返します。この要求は、CQSBRWSE 要求で指定されたクライアント・バッファに収まるだけの数のデータ・オブジェクトを返します。以降の CQSBRWSE

FUNC=BRWSOBS 要求はそれぞれ、データ・オブジェクト・エントリーの次のセットを検索します。ブラウズ・トークンは、ブラウズされているデータ・オブジェクトのカーソル位置を維持します。ゼロのブラウズ・トークンを指定した CQSBWSE FUNC=BRWSOBS 要求は、バッファに収まるだけの数のデータ・オブジェクトに関する情報を検索します。非ゼロのブラウズ・トークンを指定した CQSBWSE FUNC=BRWSOBS 要求は、データ・オブジェクト・エントリーの次のグループを検索します。バッファが、ブラウズされている最後のデータ・オブジェクトに関する情報を含んでいる場合、CQS はブラウズ・トークンを無効にして、そのブラウズ・トークンに関連付けられたデータ構造をすべて解放します。

CQSBWSE FUNC=COMPLETE 要求は、ブラウズ・トークンに関連付けられた CQSBWSE 要求が完了したことを CQS に示します。直前の CQSBWSE 要求からのブラウズ・トークンが必要です。CQS は、そのブラウズ・トークンを無効にして、それに関連付けられたデータ構造をすべて解放します。指定されたキュー上のすべてのデータ・オブジェクトを検索するのであれば、クライアントは CQSBWSE FUNC=COMPLETE 要求を出すべきです。

CQSBWSE FUNC=BRWSOBS 要求は一部のデータを返さないため、CQSBWSE FUNC=CONTINUE 要求は、リソース構造に対してはサポートされません。

重要:

- CQSBRWSE FUNC=BROWSE 要求または CQSBRWSE FUNC=CONTINUE 要求のカーソル位置は、CQS の再始動、クライアントの再始動、構造リカバリー、構造のコピー、またはブラウズ・テーブルのタイムアウトによって、失われる場合があります。(ブラウズ・テーブルは、およそ 1 時間後にタイムアウトになります。)
- CQSBRWSE 要求は、CQS またはクライアントの障害があった後はリカバリーできません。このような障害の後には、クライアントが CQSBRWSE 要求を出し直す必要があります。
- データ・オブジェクトは CQSBRWSE 要求ではロックされないため、最初の CQSBRWSE FUNC=BROWSE 要求によってスナップされたオブジェクトの 1 つ以上が、別の CQSREAD 要求または CQSDEL 要求により使用できなくなる可能性があります。
- 最初の CQSBRWSE FUNC=BROWSE 要求の後に、オーバーフローしきい値処理が行われて、キューがオーバーフロー構造に移動された場合、ブラウズ・トークンを指定した以降の CQSBRWSE FUNC=BROWSE 要求はすべて、オブジェクトが見つからないことを示すエラーになります。ブラウズ・トークンがゼロの CQSBRWSE FUNC=BROWSE 要求を再度出して、CQS がオーバーフロー構造上のキューのスナップショットを取れるようにしてください。QSMOVE 要求、またはオーバーフローしきい値処理。CQSBRWSE FUNC=BROWSE は、使用できなくなったオブジェクトを単にスキップするだけです。
- ブラウズ・テーブルがタイムアウトしたために現在位置が失われた場合、CQSBRWSE FUNC=CONTINUE 要求はリジェクトされます。

## パラメーター

### **BRWTKEN=browsetokenaddress**

16 バイトのブラウズ・トークンのアドレスを指定する入出力パラメーター。ブラウズ・トークンは、ブラウズされているデータ・オブジェクトのカーソル位置を維持します。

ブラウズ・トークンは、初期 CQSBRWSE 要求上でゼロに設定してください。CQSBRWSE FUNC=BROWSE または FUNC=BRWSOBS 要求で CQS によって返されたブラウズ・トークンを、以降の CQSBRWSE=BROWSE、CQSBRWSE=CONTINUE、CQSBRWSE=COMPLETE、または CQSBRWSE=BRWSOBS 要求で入力として渡します。

出力では、ブラウズ・トークンはブラウズされている現行データ・オブジェクトを一意的に識別します。このデータ・オブジェクトが、BUFFER パラメーターによって識別されるバッファーに返されます。

CQSBRWSE FUNC=CONTINUE 要求、CQSBRWSE FUNC=COMPLETE 要求、またはそれに続く CQSBRWSE FUNC=BROWSE 要求で、前の CQSBRWSE FUNC=BROWSE 要求で CQS によって返されたブラウズ・トークンを識別するための入力パラメーターは、BRWTKEN パラメーターです。

### **BUFFER=bufferaddress**

1 つ以上のデータ・オブジェクトに関して取得された情報を保持するクライアント・バッファーのアドレスを指定する 4 バイトの入力パラメーター。

CQSBrowse FUNC=BROWSE 要求の場合、クライアント・バッファには、キュー構造上のキューから取得されたデータ・オブジェクトのコピーが入れられます。

CQSBrowse FUNC=BRWSOBSJ 要求の場合、クライアント・バッファには、データ・オブジェクト・エントリーの数と、1 つ以上のデータ・オブジェクト・エントリーが入れられます。各データ・オブジェクト・エントリーには、リソース構造から取得した 1 つのリソース・データ・オブジェクトに関する情報が含まれます。各データ・オブジェクト・エントリーに含まれるブラウザ・データ・オブジェクトに関する情報は、リソース ID、完了コード、リソース ID 状況、バージョン、所有者、クライアント・データ 1、オプション・クライアント・データ 2、および入力リストに渡されたユーザー・データなどです。情報のサイズが、クライアントによって渡されたバッファ・サイズより大きい場合、バッファは、バッファに入る限りのリソース・エントリーで充てんされます。

BUFFER は CQSBrowse DSECT によってマップされます。

リソース ID 状況は、データ・オブジェクト・エントリー内のリソース ID が、入力パラメーターとどのように関連しているかを示します。この情報を使用して、入力パラメーターを、出力バッファで生成されるデータ・オブジェクト・エントリーにタイすることができます。リソース ID 状況には、以下のものがあります。

#### 特定パラメーター

特定のリソース ID。このデータ・オブジェクト・エントリーには、入力パラメーターと一致するリソース ID が含まれます。

#### ワイルドカード・パラメーター

ワイルドカード・パラメーターが指定されました。このデータ・オブジェクト・エントリーには、ワイルドカード・パラメーターおよび完了コードが含まれます。このデータ・オブジェクト・エントリーには、特定のリソース ID に関する情報は含まれていません。完了コードがゼロの場合、1 つ以上のワイルドカード・マッチ・リスト項目が続きます。

#### ワイルドカード・マッチ

ワイルドカード・パラメーターが指定されました。このデータ・オブジェクトには、入力ワイルドカード・パラメーターと一致する 1 つのリソース ID に関する情報が含まれています。ワイルドカード・パラメーター・リスト項目の後に、全ワイルドカード・マッチ・リスト項目が連続して続きます。

完了コードには、以下のものがあります。

**X'00000000'**

要求が正常に完了した。

**X'00000020'**

*Resourceid* パラメーターが無効です。名前のタイプは、1 から 255 までの 10 進数である必要があります。

**X'00000024'**

CQS 内部エラー。

**X'00000040'**

リソース ID、リソース・タイプ、所有者、あるいはそれらの組み合わせと一致するリソースが見つからない。

**BUFSIZE=buffer size**

クライアント・バッファのサイズを指定する 4 バイトの入力パラメーター。

**CLDTOKEN=coldqueuetoken address**

UOW と共にコールド・キュー上のオブジェクトを識別する、データ・オブジェクトの 16 バイトのコールド・キュー・トークンのアドレスを指定する出力パラメーター。

CQSRECVR 要求でコールド・キュー・トークンと作業単位 (UOW) を使用して、コールド・キュー上のオブジェクトを取得または削除することができます。

**CLIENT=client name address**

CQSREAD 要求でデータ・オブジェクトをロックしたクライアントの名前が含まれる 8 バイト・フィールドのアドレスを指定する、4 バイトの出力パラメーター。このパラメーターは、QTYPE=COLD パラメーターが指定されている場合のみ有効です。

**CONTOKEN=connect token address**

この CQS が管理する特定のカップリング・ファシリティ構造に対するクライアントの接続を一意的に識別する、16 バイトの接続トークンのアドレスを指定する入力パラメーター。接続トークンは、CQSCONN 要求によって戻されません。

**COUNT=resource list count**

リソース・リスト内の項目の数を指定する、4 バイトの入力パラメーター。

**CQSTOKEN=cqstoken address**

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

**ECB=ecb address**

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

**LIST=resource list address**

1 つ以上のエントリーを含むリソース・リストを指定する可変サイズ入力パラメーターのアドレス。各項目が、それぞれ別々のブラウズ要求になります。クライアントは、CQSBRWSE 要求を発行する前に、各エントリー内のフィールドの一部を初期化する必要があります。他のフィールドは、要求の完了時に CQS から返されます。

CQSBRWSL リスト項目の DSECT は、リスト項目をマップし、クライアントが使用することができます。複数のリスト項目は、連続するストレージに常駐させる必要があります。

各リスト項目には以下のパラメーターが含まれています。

**resource id**

ブラウズ対象のリソースの固有 ID を含む、12 バイトの入力フィールド。リソース ID はワイルドカード・パラメーターにすることができます。リソース ID は IMSplex 内で固有です。リソース ID は、11 バイトのクライアント定義の名前と、それに続く 1 バイトの名前タイプから構成されます。名前タイプは、同じ名前タイプを持つリソースに対するクライアント定

義の名前の固有性を保証します。異なるリソース・タイプのリソースが同じ名前タイプを持つ場合もあります。名前タイプに対して有効な値は、1 から 255 までの 10 進数です。クライアント定義の名前は、クライアントに対して意味を持ち、英数字から構成されます。ワイルドカード・パラメーターを使用してリソース ID を指定する場合は、パフォーマンスを向上させるために、リソース・タイプも指定してください。リソース ID、リソース・タイプ、またはその両方を指定する必要があります。

#### **resourcetype**

リソース・タイプを指定する 1 バイトの入力フィールド。リソース・タイプは、リソース構造上のリソースの、クライアント定義の物理的グループ化です。リソース・タイプに対して有効な値は、1 から 255 までの 10 進数です。リソース・タイプが、CQS によって定義されたリソース・タイプの最大数 (11) より大きい場合、折り返されて既存のリソース・タイプのうちの 1 つになります。リソース・タイプ、リソース ID、またはその両方を指定する必要があります。

#### **予約済み**

3 バイトの予約フィールド。

#### **owner**

ブラウズ対象のリソース・データ・オブジェクトの所有者を識別する、8 バイトの入力パラメーター。CQSBRWSE 要求は、特定の所有者によって所有されているリソース・データ・オブジェクトのみを戻します。owner はオプション・パラメーターです。

#### **options**

ブラウズ・オプションを指定する 4 バイトの入力パラメーター。可能なオプションは以下のとおりです。

#### **X'80000000'**

ブラウズされたデータ・オブジェクトに対して data2 を返します。

#### **userdata**

ユーザー・データを指定する 4 バイトの入力パラメーター。このユーザー・データは、入力リソース ID パラメーターに一致する各データ・オブジェクトに対する出力上で渡されます。

#### **LISTVER=1 | listversion**

このリスト・バージョンに対する等価を指定する入力パラメーター。デフォルトは 1 です。CQSBRWSE リスト・バージョンのためにプログラムの中に等価 (EQU) ステートメントを組み込むには、CQSBRWSE 要求の DSECT 機能を使用します。

#### **OBJSIZE=dataobjectsizaddress**

データ・オブジェクトまたはデータ・オブジェクト・エントリーのサイズを格納する、4 バイトのエリアのアドレスを指定する出力パラメーター。

CQSBRWSE FUNC=BROWSE 要求が発行され、かつデータ・オブジェクトのサイズがクライアントによって渡されたバッファー・サイズよりも大きい場合、バッファーは、バッファーに入る限りのデータ・オブジェクトで充てんされます。この要求は、一部のデータが戻されたことを示す、戻りコードおよび理由コードを受信します。データ・オブジェクトのサイズは、OBJSIZE パラメーターによって指定されたロケーション内に戻されます。データ・オブジェクトのサイ

ズがバッファのサイズと等しいかそれ以下である場合、そのデータ・オブジェクトがバッファ内に移動され、バッファの残りの部分は変更されません。

CQSBRWSE FUNC=BRWSOBS 要求が発行された場合、フィットする限りのデータ・オブジェクト・エントリがバッファに移動されます。その場合、クライアントは次の CQSBRWSE FUNC=BRWSOBS 要求を発行して、次のデータ・オブジェクト・エントリを検索する必要があります。バッファが、次のデータ・オブジェクト・エントリを保留できるほど大規模でない場合、この要求は、バッファが小さすぎることを示す、戻りコードおよび理由コードを受信します。戻される次のデータ・オブジェクト・エントリのサイズは、OBJSIZE パラメーターによって指定されたロケーション内に保管されます。

#### **PARM=parmaddress**

要求が CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値である CQSBRWSE\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

#### **QNAME=queuenameaddress**

16 バイトのキュー名フィールドのアドレスを指定する 4 バイトの出力パラメーター。

QTYPE=COLD および CLDTOKEN パラメーターを指定する CQSBRWSE 要求の場合、キュー名フィールドが出力フィールドとなり、このフィールドに、返されたデータ・オブジェクトの元のクライアント・キュー名が含まれます。このクライアント・キュー名には、コールド・キューに移動される前のデータ・オブジェクトが入っていました。

他のすべての CQSBRWSE 要求の場合、キュー名フィールドはキュー名を指定する入力フィールドであり、すべての CQSBRWSE 要求に対するデータ・オブジェクトがそこから検索されます。

#### **QTYPE=COLD**

データ・オブジェクトを検索するキューのタイプを指定する入力パラメーター。COLD は、コールド・キューからデータ・オブジェクトを取得することを示します。

#### **RETCODE=returncodeaddress**

CQSBRWSE の戻りコードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 の戻りコードがゼロ以外の場合は、CQS インターフェースがエラーを検出して CQS に要求を送信できなかったことを示すため、戻りコードと理由コードのフィールド値は無効です。

#### **RSNCODE=reasoncodeaddress**

CQSBRWSE の理由コードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

#### **TIMESTAMP=timestampaddress**

データ・オブジェクトが各キューに配置されたときのタイム・スタンプが含まれる 8 バイト・フィールドのアドレスを指定する 4 バイト出力パラメーター。

#### **UOW=uowaddress**

キューから検索されたデータ・オブジェクトの作業単位 (UOW) を保持する 32

バイト・エリアのアドレスを指定する出力パラメーター。UOW は、キューにデータ・オブジェクトを保管した (CQSPUT 要求) クライアントによって生成された固有 ID です。

## 戻りコードおよび理由コード

以下の表は、CQSBRWSE 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に等価 EQU ステートメントを組み込むには、CQSBRWSE FUNC=DSECT 要求を使用します。

表 7. CQSBRWSE 要求の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000004'	X'00000120'	バッファー・サイズ ( <i>buffersize</i> ) がデータ・オブジェクト・サイズ ( <i>dataobjectsize</i> ) より小さい。部分的なデータが戻される。
X'00000004'	X'00000124'	バッファー・サイズ ( <i>buffersize</i> ) が小さすぎて、次のリソース・データ・オブジェクト・エントリーを含むことができない。戻された部分的データはない。
X'00000004'	X'00000128'	指定されたキュー名 ( <i>queuename</i> ) に検索すべきデータ・オブジェクトがない。
X'00000004'	X'0000012C'	戻すべき部分的なデータがない。
X'00000004'	X'00000138'	要求は完了し、最後のデータ・オブジェクトが戻される。
X'00000004'	X'0000013C'	データ・オブジェクトはもう存在しない。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000214'	<i>connecttoken</i> が無効。
X'00000008'	X'00000218'	FUNC が無効。
X'00000008'	X'0000021C'	<i>browsetoken</i> が無効。
X'00000008'	X'00000220'	<i>queuename</i> が無効。
X'00000008'	X'00000224'	<i>buffer</i> が無効。
X'00000008'	X'00000228'	<i>buffersize</i> が無効。
X'00000008'	X'0000022C'	<i>dataobjectsize</i> が無効。
X'00000008'	X'00000230'	<i>uow</i> が無効。
X'00000008'	X'00000234'	<i>browsetoken</i> が無効。
X'00000008'	X'00000250'	<i>count</i> が無効。
X'00000008'	X'00000254'	<i>listaddress</i> が無効。
X'00000008'	X'0000027C'	CQSBRWSE FUNC=BROWSE 要求は、リソース構造に対して許可されない。CQSBRWSE FUNC=CONTINUE 要求は、リソース構造に対して許可されない。リソース構造から戻された部分的データはない。
X'00000008'	X'00000280'	CQSBRWSE FUNC=BRWSOBSJ 要求は、キュー構造に対して許可されない。
X'00000008'	X'00000284'	Parm <i>listversion</i> が無効です。
X'00000008'	X'00000288'	<i>listversion</i> が無効です。

表 7. CQSBRWSE 要求の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'00000010'	X'00000400'	この構造には CQSRSYNC 要求が必須です。
X'00000010'	X'00000404'	構造がアクセス不能。後で要求を再試行してください。
X'00000010'	X'00000408'	現在位置が失われた。CQSBRWSE 要求を再発行してください。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。
X'00000014'	X'00000500'	CQS 内部エラー。

## CQSCHKPT 要求

CQSCHKPT 要求を使用して、CQS のシステム・チェックポイントまたは構造チェックポイントのいずれかを開始します。

### CQSCHKPT の形式

#### CQSCHKPT の CHKPTSTR 機能

キュー構造に対する CQS 構造チェックポイントを開始するには、CQSCHKPT 要求の CHKPTSTR 機能を使用します。構造チェックポイントは、リソース構造に対してサポートされていません。

▶▶—CQSCHKPT—FUNC=CHKPTSTR—CQSTOKEN=*cqstokenaddress*—PARAM=*parmaddress*————▶▶

▶▶—COUNT=*count*—LIST=*listaddress*————▶▶  
└─ECB=*ecbaddress*—┘ RETCODE=*returncodeaddress*————▶▶

▶▶—RSNCODE=*reasoncodeaddress*—CQSCHKPT—FUNC=CHKPTSTR—CQSTOKEN=*cqstokenaddress*————▶▶

▶▶—PARAM=*parmaddress*—COUNT=*count*—LIST=*listaddress*————▶▶  
└─ECB=*ecbaddress*—┘

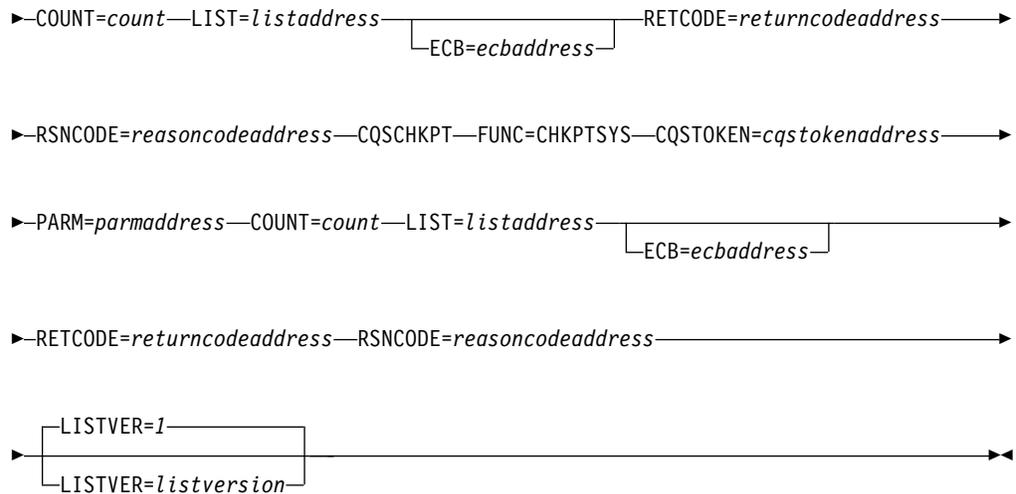
▶▶—RETCODE=*returncodeaddress*—RSNCODE=*reasoncodeaddress*————▶▶

▶▶—LISTVER=*1*————▶▶  
└─LISTVER=*listversion*—┘

#### CQSCHKPT の CHKPTSYS 機能

CQS システム・チェックポイントを開始するには、CQSCHKPT 要求の CHKPTSYS 機能を使用します。

▶▶—CQSCHKPT—FUNC=CHKPTSYS—CQSTOKEN=*cqstokenaddress*—PARAM=*parmaddress*————▶▶



### CQSCHKPT の DSECT 機能

CQSCHKPT パラメーター・リストの長さ、および CQSCHKPT の戻りコードと理由コードのために、プログラムの中に等価 (EQU) ステートメントを組み込むには、CQSCHKPT 要求の DSECT 機能を使用します。



## CQSCHKPT の使用法

構造チェックポイントの場合、CQS は、チェックポイント・リストに指定されている各構造のキューを DASD にダンプします。現時点で構造がオーバーフロー・モードの場合は、オーバーフロー構造も DASD にダンプされます。

システム・チェックポイントの場合、CQS は、チェックポイント・リストに指定されている各構造の内部テーブルをログに記録します。構造が現在オーバーフロー・モードである場合、CQS はオーバーフロー構造の内部テーブルもログに記録します。

### パラメーターの説明

#### COUNT=count

チェックポイント・リスト内の項目の数を指定する、4 バイトの入力パラメーター。

#### CQSTOKEN=cqstokenaddress

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

#### ECB=ecbaddress

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

**LIST=listaddress**

チェックポイント・リストのアドレスを指定する、4 バイトの入力パラメーター。チェックポイント・リストには、クライアントがチェックポイントを要求する各構造の項目が入っている必要があります。

CQSCHKPL リスト項目の DSECT は、リスト項目をマップし、クライアントが使用することができます。複数のリスト項目は、連続するストレージに常駐させる必要があります。

各リスト項目には以下のパラメーターが含まれています。

**connecttoken**

CQSCONN 要求によって戻される接続トークンを指定する、16 バイトの入力パラメーター。接続トークンは、この CQS が管理する特定のカップリング・ファシリティ構造に対するクライアントの接続を一意的に識別します。このパラメーターは必須です。

**compcode**

要求からの完了コードを受け取るための 4 バイトの出力フィールド。完了コードには、以下のものがあります。

X'00000000'

正常に完了した。

X'00000004'

接続トークンが無効。

X'00000008'

CQS チェックポイント要求は、CQS 再始動によるシステム・チェックポイントが正常に完了するまで許可されない。

X'0000000C'

この構造には CQSRSYNC が必須。

X'00000010'

チェックポイントはすでに構造に対して進行中。

X'00000014'

構造がアクセス不能。後で要求を再試行してください。

X'00000018'

CQS 内部エラー。

X'00000020'

CQSCHKPT FUNC=CHKPTSTR は、リソース構造に対して無効。

**LISTVER=1 | listversion**

このリスト・バージョンに対する等価を指定する入力パラメーター。

CQSCHKPT リスト・バージョンのためにプログラムの中に等価 (EQU) ステートメントを組み込むには、CQSCHKPT 要求の DSECT 機能を使用します。

**PARM=parmaddress**

要求が CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU ステートメントの値 CQSCHKPT\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

**RETCODE=returncodeaddress**

CQSCHKPT の戻りコードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

**RSNCODE=reasoncodeaddress**

CQSCHKPT の理由コードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

**CQSCHKPT の戻りコードと理由コード**

以下の表は、CQSCHKPT 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSCHKPT FUNC=DSECT 要求を使用します。

表 8. CQSCHKPT の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000218'	FUNC が無効。
X'00000008'	X'00000250'	<i>count</i> が無効。
X'00000008'	X'00000254'	<i>listaddress</i> が無効。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000008'	X'00000288'	リスト・バージョンが無効。
X'0000000C'	X'00000300'	要求は少なくとも 1 つのリスト項目については成功したが、すべてではない。個々のエラーについては、 <i>compcode</i> を参照してください。
X'0000000C'	X'00000304'	要求はすべてのリスト項目について失敗した。個々のエラーについては、 <i>compcode</i> を参照してください。
X'00000010'	X'0000040C'	CQS シャットダウンが保留中。クライアントが開始したチェックポイント要求は許可されない。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。

関連概念:

 CQS システム・チェックポイントの使用 (システム管理)

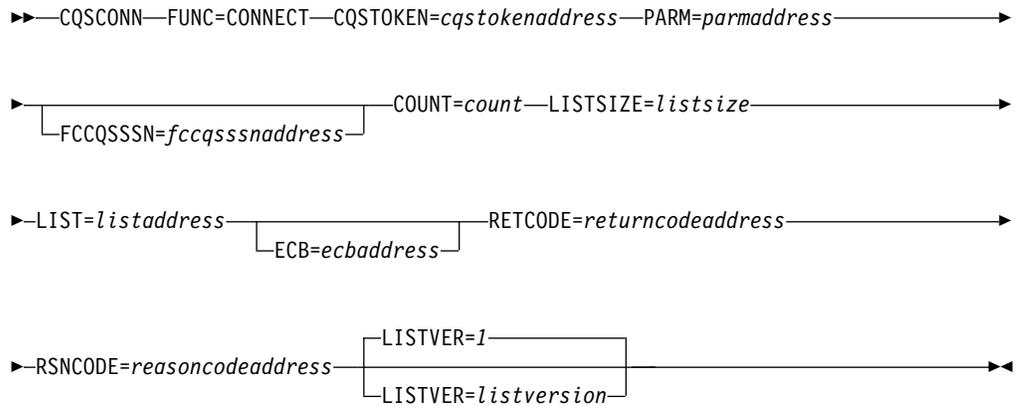
**CQSCONN 要求**

CQSCONN 要求は、1 つ以上のカップリング・ファシリティー構造へクライアントを接続します。

**CQSCONN の形式**

CQSCONN の CONNECT 機能

1 つ以上のカップリング・ファシリティ構造に接続するには、CQSCONN 要求の CONNECT 機能を使用します。カップリング・ファシリティ構造は、キュー構造またはリソース構造である場合があります。



### CQSCONN の DSECT 機能

CQSCONN パラメーター・リストの長さ、および CQSCONN の戻りコードと理由コードのために、プログラムの中に等価 (EQU) ステートメントを組み込むには、CQSCONN 要求の DSECT 機能を使用します。



### CQSCONN の使用法

CQSCONN 要求は、1 つ以上のカップリング・ファシリティ構造へクライアントを接続します。クライアントは、1 つ以上のリスト項目がある接続リストを指定しますが、その中の各項目は別々の接続要求です。構造への接続が成功すると、クライアントに構造に対する接続を表す接続トークンが戻されます。クライアントは、その構造に対する以後すべての CQS 要求で、このトークンを指定する必要があります。最大 32 のクライアントが、CQS アドレス・スペースを使用して 1 つのカップリング・ファシリティ構造に接続することができます。

制約事項: リソース構造に対する CQSCONN 要求はログに記録されず、この要求は FCCQSSSN キーワードをサポートしません。CQSCONN 要求はリソース構造に対する以下の接続リスト・パラメーターをサポートしません。

- *structureattributes*
- *overflowstructurename*
- *structureinformexit*
- *structureinformparm*
- *qtypecnt*
- *qtypelist*

CQSCONN FUNC=CONNECT 要求は、CQSREG FUNC=REGISTER 要求の後で、かつ他のどの CQS 要求よりも前に出す必要があります。また、CQS 異常終了

と再始動の後、およびクライアントが CQS に登録された後で、クライアントが他の CQS 要求を出すには、CQSCONN FUNC=CONNECT 要求が必要です。

パラメーター説明

**COUNT=count**

接続リストのリスト項目の数を指定する、4 バイトの入力パラメーター。

**CQSTOKEN=cqstokenaddress**

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

**ECB=ecbaddress**

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

**FCCQSSSN=fccqssnaddress**

障害が起きたクライアント CQS サブシステムのアドレスを指定する 4 バイトの入力パラメーター。あるクライアントが別のクライアントを引き継いだ場合、これは障害が起きたクライアントに接続された CQS の SSN です。

このキーワードは、リソース構造に対して適用できません。

**LIST=listaddress**

1 つ以上の項目を含む接続リストのアドレスを指定する 4 バイトの入力パラメーター。各項目は、クライアントをカップリング・ファシリティ構造へ接続する独立した要求です。各項目の一部のフィールドは、CQSCONN 要求の前にクライアントによって初期化されている必要があります。他のフィールドは、CQSCONN 要求の完了時に CQS によって戻されます。

CQSCONNL リスト項目の DSECT は、リスト項目をマップし、クライアントが使用することができます。複数のリスト項目は、連続するストレージに常駐させる必要があります。

注: 現在「使用されない」と記載されている、CQSCONNL DSECT 内のすべてのフィールドは、CQSCONN の呼び出し元がゼロにセットしておく必要があります。

各リスト項目には以下のパラメーターが含まれています。

**comcode**

要求からの完了コードを受け取るための 4 バイトの出力フィールド。完了コードには、以下のものがあります。

**X'00000000'**

クライアント接続に成功した。接続トークンがクライアントに戻されます。

**X'00000004'**

クライアントはすでにこの CQS を通して構造に接続されている。接続トークンがクライアントに戻されます。

**X'00000008'**

*structurename* が無効。

**X'0000000C'**

構造イベント出口ルーチンのアドレスが指定されていなかった。

**X'00000010'**

クライアントはすでに別の CQS を通して構造に接続されている。クライアントが一定の構造に接続できるのは、1 つの CQS を介してのみです。クライアントは、この CQS を介しては構造に接続されていません。このことは、他の CQS によるクライアント接続の状況には影響しません。

**X'00000014'**

CQS 内部エラー。

**X'00000018'**

クライアントが、障害が起きたクライアントの作業を引き継ぐために構造に接続しようとして、FCCQSSSN= パラメーターを指定した。しかし、CQS は、障害が起きたクライアントに接続されていた CQS に有効なシステム・チェックポイント・ログ・トークンを検出できなかった。CQS はメッセージ CQS0033A を出し、それに対してオペレーターが REJECT を応答した。

**X'0000001C'**

クライアント・アドレス・スペースのユーザー ID は、構造への接続の許可を受けていない。

**X'00000020'**

structureinformexit は指定されていましたが、リソース構造に対しては許可されていません。

**X'00000024'**

structureinformparm は指定されていましたが、リソース構造に対しては許可されていません。

**X'0000002C'**

structureattributes は指定されていましたが、リソース構造に対しては許可されていません。

**X'00000030'**

Qtype は指定されていましたが、リソース構造に対しては許可されていません。

**X'00000034'**

FCCQSSSN は指定されていましたが、リソース構造に対しては許可されていません。

**structureattributes**

構造属性が入った 4 バイトの入出力パラメーター・フィールド。

+0 フラグ・バイト 1 で、ビットの定義は以下の通りです。

**X'80'** 構造の仕様が『再作成待機』属性であることを示します。シスプレックス内で最初に構造へ接続するクライアントが、この属性をすべてのクライアントに代わって定義します。これは接続要求の際に戻され、この属性がそれぞれのクライアント

トのニーズに合った正しい設定であるかどうかをチェックすることができます (この属性が前のクライアント接続によって定義されたままになっている可能性があるため)。

*structureattributes* で指定された値は、構造が持続する限り有効で、変更はできません。

0 にセットされている場合、構造にデータ・オブジェクトを書き込んだり検索するクライアント要求が、再作成が完了するまで待たないことを示します。

1 にセットされている場合、構造にデータ・オブジェクトを書き込んだり検索するクライアント要求は、再作成の完了を待たなければならないことを示します。

**X'40'** CQS によって戻される出力フラグ。このフラグは、キュー構造の場合にのみ、構造がリカバリー不能構造であるかどうか (RECOVERABLE=NO が CQSSGxxx PROCLIB メンバーで指定されたかどうか) を示します。構造がリカバリー不能構造である場合にこのフラグが 1 にセットされ、それ以外の場合は 0 にセットされます。

このフラグは、リソース構造に対しては適用できません。

このバイトのその他のビットは使用されておらず、ゼロにセットしておく必要があります。

**+1** 次の 3 バイトは使用されておらず、ゼロに設定しておく必要があります。

#### **structuretype**

キュー構造またはリソース構造のどちらかとして構造タイプを指定する、1 バイトの出力パラメーター・フィールド。

#### **structureversion**

クライアントが接続した構造の構造バージョンを指定する、8 バイトの出力パラメーター・フィールド。

#### **structurename**

クライアントが接続を希望する構造の名前が入った 16 バイトの入力パラメーター・フィールド。このパラメーターは必須です。

#### **overflowstructurename**

オーバーフロー構造の名前が CQS グローバル構造定義 PROCLIB のメンバーである CQSSGxxx 内で CQS に対して定義されていた場合に、オーバーフロー構造の名前を受け取るための 16 バイトの出力パラメーター・フィールド。

このパラメーターは、リソース構造に対して適用できません。

#### **connecttoken**

この CQS が管理する特定のカップリング・ファシリティ構造に対するクライアントの接続を一意的に識別する接続トークンを受け取るための、16 バイトの出力パラメーター・フィールド。

### **structureeventexit**

構造イベント出口ルーチンのアドレスが入った 4 バイトの入力パラメーター・フィールド。このパラメーターは必須です。

### **structureeventparm**

出口が呼び出されるたびに CQS が構造イベント出口ルーチンに渡すクライアント・データが入った、4 バイトの入力パラメーター・フィールド。このパラメーターの指定はオプションです。出口ルーチンに渡すデータがない場合は、これをゼロにセットしてください。

### **structureinformexit**

構造通知出口ルーチンのアドレスが入った 4 バイトの入力パラメーター・フィールド。このパラメーターの指定はオプションです。構造通知出口ルーチンがない場合は、これをゼロにセットしてください。

このパラメーターは、リソース構造に対して適用できません。

### **structureinformparm**

出口が呼び出されるたびに CQS が構造通知出口ルーチンに渡すクライアント・データが入った、4 バイトの入力パラメーター・フィールド。このパラメーターの指定はオプションです。出口ルーチンに渡すデータがない場合は、これをゼロにセットしてください。

このパラメーターは、リソース構造に対して適用できません。

### **qtypecnt**

キュー・タイプ・リスト内のキュー・タイプ項目の数が入った、4 バイトの入力パラメーター・フィールド。このパラメーターの指定はオプションです。キュー・タイプ・リストに項目がない場合は、これをゼロにセットしてください。

このパラメーターは、リソース構造に対して適用できません。

### **qtypelst**

キュー・タイプ・リストの可変長入力域。

このパラメーターは、リソース構造に対して適用できません。

このエリアの長さは、*qtypecnt* に指定された値と同じです。各キュー・タイプ項目は 1 バイトのキュー・タイプの値であり、基本構造がオーバーフロー・モードになっても、オーバーフロー構造に移動させてはいけません。このパラメーターはオプションです。

CQSCONN パラメーター・リストのバージョン 1 を使用している場合 (これはデフォルトです)、CQSCONNL DSECT 内のラベル CNLQTYPL で始まるキュー・タイプ・リストを作成し、これによりリスト項目をマップします。CQSCONN パラメーター・リストのバージョン 16 を使用している場合は、ラベル CNLQTYPL\_V16 で始まるキュー・タイプ・リストを作成します。

キュー・タイプは、定義された後は構造が存続する限り有効なので、オーバーフロー構造に移動されません。

キュー・タイプがリストされていない場合、デフォルトで、すべてのキュー・タイプについてオーバーフロー可能になります。このリストは、クライアントが移動するべきでないことを認識している (おそらくクラ

イアントのキュー・タイプの使用法に基づいて) キュー・タイプがある場合だけ組み込むようにしてください。

推奨事項: クライアントは、複数のオブジェクトが同じキュー名と UOW を持つことを許可するキュー・タイプを処理から除外する必要があります。CQS は、同じキュー名と UOW を持つ複数のオブジェクトがオーバーフロー構造への移動を許可された場合、リカバリーできません。

#### **logstreamname**

CQS 構造に関連付けられている z/OS ログ・ストリームの名前を受け取る、26 バイトの出力パラメーター・フィールド。このフィールドは、リカバリー不能キュー構造およびリソース構造の場合、すべてブランクにセットされます。

このフィールドは、バージョン 16 以降の CQSCONN リストの場合のみ存在します。

#### **logstreamstructurename**

CQS 構造に関連付けられている z/OS ログ・ストリームと関連付けられる CF 構造の名前を入れる、16 バイトの出力パラメーター・フィールド。このフィールドは、リカバリー不能キュー構造、リソース構造、および DASD-only z/OS ログ・ストリームを持つ構造の場合、すべてブランクにセットされます。

このフィールドは、バージョン 16 以降の CQSCONN リストの場合のみ存在します。

#### **LISTSIZE=*listsize***

接続リストのサイズを指定する 4 バイトの入力パラメーター。 *listsize* は、1 つの項目の長さではなく、リスト内のすべての項目の合計の長さを指定します。

#### **LISTVER=1 | *listversion***

パラメーター・リスト・バージョンを指定する入力パラメーター。CQSCONN リスト・バージョンおよび長さに合わせてプログラムに等価 (EQU) ステートメントを組み込むには、CQSCONN 要求の DSECT 機能を使用します。サポートされるパラメーター・リスト・バージョンは、以下のとおりです。

- 1** EQU シンボルは CNL\_LVER1 です。これは、デフォルトのパラメーター・リスト・バージョンです。パラメーター・リストのこのバージョンには、より高いリスト・バージョンでのみ存在することが特に注記されているフィールドを除き、LIST= パラメーターの下に記載されているすべてのフィールドが組み込まれています。バージョン 1 パラメーター・リスト項目の最小の長さは、CNL\_MINLNV1 バイトです。キュー・タイプ項目は、存在する場合、CQSCONNL DSECT 内のラベル CNLQTYPL で始まり、これによりリスト項目をマッピングします。
- 16** EQU シンボルは CNL\_LVER16 です。バージョン 16 パラメーター・リストには、バージョン 1 パラメーター・リストに存在するフィールドを上回る追加の出力フィールドが含まれています。これらの追加フィールドは LIST= パラメーターの下に記載され、バージョン 16 フォーマットのパラメーター・リストが渡された場合にのみ戻されます。バージョン 16 パラメーター・リスト項目の最小の長さは、CNL\_MINLNV16 バイトです。キュー

ー・タイプ項目は、存在する場合、CQSCONNL DSECT 内のラベル CNLQTYPL\_V16 で始まり、これによりリスト項目をマッピングします。

**PARM=parmaddress**

要求が、CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSCONN\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

**RETCODE=returncodeaddress**

CQSCONN の戻りコードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

**RSNCODE=reasoncodeaddress**

CQSCONN の理由コードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

**CQSCONN の戻りコードと理由コード**

以下の表は、CQSCONN 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSCONN FUNC=DSECT 要求を使用します。

表 9. CQSCONN の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000004'	X'00000100'	クライアントは、以前はこの CQS を介して指定された構造の中の 1 つ以上に接続されていた。クライアントはすべての構造に接続される。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000218'	<i>FUNC</i> が無効。
X'00000008'	X'00000250'	<i>count</i> が無効。
X'00000008'	X'00000254'	<i>listaddress</i> が無効。
X'00000008'	X'00000258'	<i>listsize</i> が無効。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000008'	X'00000288'	リスト・バージョンが無効。
X'0000000C'	X'00000300'	要求は 1 つのリスト項目については成功したが、すべてのリスト項目についてではない。個々のエラーについては、 <i>compcode</i> を参照してください。
X'0000000C'	X'00000304'	要求はすべてのリスト項目について失敗した。個々のエラーについては、 <i>compcode</i> を参照してください。
X'00000010'	X'0000040C'	CQS シャットダウンが進行中 (CQSSHUT)。CQS はすべてのクライアントが切断されるのを待っているため、新規クライアントの接続は許可されません。

表 9. CQSCONN の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'00000010'	X'00000410'	クライアントの最大数がこの CQS に接続された。この要求で、クライアント接続の限界を超える可能性がある。これ以上のクライアント接続は許可されない。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。
X'00000014'	X'00000500'	CQS 内部エラー。

## CQSDEL 要求

CQSDEL 要求は、1 つ以上のデータ・オブジェクトを、キュー構造またはリソース構造から削除します。

### CQSDEL の形式

#### CQSDEL の DELETE 機能

1 つ以上のデータ・オブジェクトを、キュー構造またはリソース構造から削除するには、CQSDEL 要求の DELETE 機能を使用します。

▶▶—CQSDEL—FUNC=DELETE—CQSTOKEN=*cqstokenaddress*—————▶▶

▶—CONTOKEN=*connecttokenaddress*—PARAM=*parmaddress*—COUNT=*count*————▶

▶—LIST=*listaddress*—————▶  
                                   └—ECB=*ecbaddress*—┘                                  ▶—RETCODE=*returncodeaddress*————▶

▶—RSNCODE=*reasoncodeaddress*—————▶  
   └—LISTVER=*listversion*—┘  
   └—LISTVER=1—————┘

#### CQSDEL の DSECT 機能

CQSDEL パラメーター・リストの長さ、および CQSDEL の戻りコードと理由コードのために、プログラムの中に等価 (EQU) ステートメントを組み込むには、CQSDEL 要求の DSECT 機能を使用します。

▶▶—CQSDEL—FUNC=DSECT—————▶▶

### CQSDEL の使用法

CQSDEL 要求は、1 つ以上のデータ・オブジェクトを、キュー構造またはリソース構造から削除します。クライアントは、1 つ以上のリスト項目を含んだ削除リストを指定しますが、その中の各項目は、別々の削除要求 (ロック・トークン、キュー

名、キュー名および UOW、リソース ID、またはリソース・タイプおよび所有者のいずれかによる削除) です。各リスト項目は別々に処理され、それぞれ独自の完了コードを受け取ります。

パラメーターの説明:

**CONTOKEN=connecttokenaddress**

この CQS が管理する特定のカップリング・ファシリティ構造に対するクライアントの接続を一意的に識別する、16 バイトの接続トークンのアドレスを指定する入力パラメーター。接続トークンは、CQSCONN 要求によって戻されません。

**COUNT=count**

削除リスト内のリスト項目の数を指定する、4 バイトの入力パラメーター。

**CQSTOKEN=cqstokenaddress**

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

**ECB=ecbaddress**

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

**LIST=listaddress**

1 つ以上の項目が入っている削除リストのアドレスを指定する 4 バイトの入力パラメーター。各項目が、それぞれ別々の削除要求になります。各項目の一部のフィールドは、CQSDEL 要求の前にクライアントが初期化しておく必要があります。他のフィールドは、要求の完了時に CQS によって戻されます。

CQSDELL リスト項目の DSECT は、リスト項目をマップし、クライアントが使用することができます。複数のリスト項目は、連続するストレージに常駐させる必要があります。

各リスト項目には以下のパラメーターが含まれています。

**deletetype**

削除タイプが入った 1 バイトの入力パラメーター・フィールド。これは必須パラメーターです。deletetype は次の値のいずれか 1 つが可能です。

- 1      ロック・トークンによる削除
- 2      キュー名による削除
- 3      キュー名と作業単位による削除
- 4      リソース ID およびバージョンによる削除
- 5      特定の所有者を持つリソース・タイプによる削除

推奨事項: パフォーマンスを向上させるためには、削除タイプ 1 または削除タイプ 2 を使用してください。これらのタイプは、削除タイプ 3 より効率的です。

**deleteqpos**

すべてのデータ・オブジェクトを削除することを指定するか、削除すべ

きデータ・オブジェクトのキュー上の位置を指定する、1 バイトの入力パラメーター・フィールド。このパラメーターは削除タイプ 2 だけに使用されます。*deleteqpos* は次のいずれか 1 つが可能です。

- 1 キュー上のすべてのデータ・オブジェクトを削除する
- 2 キュー上の最初のデータ・オブジェクトを削除する
- 3 キュー上の最後のデータ・オブジェクトを削除する

*locktoken*、*deleteqpos*、および *uow* フィールドは互いに排他的です。

***reserved***

2 バイトの予約フィールド。

***objdelcnt***

削除されたデータ・オブジェクトの数を受け取る 4 バイトの出力パラメーター・フィールド。

***compcode***

要求からの完了コードを受け取るための 4 バイトの出力フィールド。完了コードには、以下のものがあります。

**X'00000000'**

要求が正常に完了した。

**X'00000004'**

無効な *deleteqpos* (削除タイプ 2)。

**X'00000008'**

無効な *deletetype*。

**X'0000000C'**

無効な *locktoken* (削除タイプ 1)。

**X'00000010'**

無効な *queuename* (削除タイプ 2 またはタイプ 3)。

**X'00000014'**

無効な *uow* (削除タイプ 3)。

**X'0000001C'**

構造がアクセス不能。後で、この要求を再試行してください。

**X'00000020'**

CQS 内部エラー。

**X'00000024'**

キュー上 (削除タイプ 2 の場合)、または UOW の *queuename* 上 (削除タイプ 3 の場合)、またはリソース構造上 (削除タイプ 4 の場合) でデータ・オブジェクトが検出できない。このケースをエラーとして扱うかどうかは、クライアントの判断です。

**X'00000028'**

削除タイプ 1、2、または 3 は、リソース構造に対して無効。

**X'00000032'**

削除タイプ 4 または 5 はキュー構造に対して無効。

X'00000036'

*Resourceid* は無効。名前のタイプは、1 から 255 までの 10 進数である必要があります。

X'00000040'

*Version* は無効。バージョンは、ゼロより大きい数である必要があります。

X'00000044'

*Version* は、既存のリソースのそれと一致しません。

X'00000048'

*Resourcetype* は無効。リソース・タイプは、1 から 255 までの 10 進数である必要があります。

### **locktoken**

ロック・トークンが入った 16 バイトの入力パラメーター・フィールド。ロック・トークンは、CQSREAD 要求によって戻されます。このパラメーターは、削除タイプ 1 だけに使用されます。

*locktoken*、*deleteqpos*、および *uow* フィールドは互いに排他的です。  
*locktoken* フィールドと *queuename* フィールドも互いに排他的です。

### **queuename**

キュー名が入った 16 バイトの入力パラメーター・フィールド。このパラメーターは、削除タイプ 2 および 3 だけに使用されます。

*locktoken* フィールドと *queuename* フィールドは互いに排他的です。

### **uow**

作業単位を含む 32 バイトの入力パラメーター。このパラメーターは、削除タイプ 3 だけに使用されます。

*locktoken*、*deleteqpos*、および *uow* フィールドは互いに排他的です。

### **resourceid**

削除するリソース・データ・オブジェクトの固有 ID を含む 12 バイトの入力パラメーター。このパラメーターは、削除タイプ 4 には必須です。*resourceid*、*locktoken*、*queuename*、および *resourcetype* フィールドは、互いに排他的です。

### **バージョン**

削除されるリソースのバージョンを含む 8 バイトの入力および出力パラメーター。削除要求が成功するためには、指定されたバージョンはリソースのバージョンと一致する必要があります。バージョンは、リソースが更新された回数のカウントです。このパラメーターは、削除タイプ 4 には必須です。バージョンのミスマッチにより削除が失敗した場合、そのバージョンは出力として戻されます。

### **resourcetype**

リソース・タイプを含んでいる 1 バイトの入力パラメーター。リソース・タイプは、リソース構造上のリソースの、クライアント定義の物理的グループ化です。リソース・タイプに対して有効な値は、1 から 255 までの 10 進数です。リソース・タイプが、CQS によって定義されたリソース・タイプの最大数 (11) より大きい場合、折り返されて既存のリソース・タイプのうちの 1 つになります。このパラメーターは、削

除タイプ 4 および 5 には必須です。ゼロを指定して、所有されていないリソース・タイプの、すべてのリソースを削除してください。

**reserved**

3 バイトの予約フィールド。

**owner** 指定されたリソース・タイプのリソースを削除する所有者を指定する 8 バイトの入力パラメーター。このパラメーターは、削除タイプ 5 に対して必要です。

**LISTVER=1 | listversion**

このリスト・バージョンに対する等価を指定する入力パラメーター。CQSDEL リスト・バージョンのためにプログラムの中に等価 (EQU) ステートメントを組み込むには、CQSDEL 要求の DSECT 機能を使用します。

**PARM=parmaddress**

要求が CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSDEL\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

**RETCODE=returncodeaddress**

CQSDEL の戻りコードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

**RSNCODE=reasoncodeaddress**

CQSDEL の理由コードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

## CQSDEL の戻りコードと理由コード

以下の表は、CQSDEL 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSDEL FUNC=DSECT 要求を使用します。

表 10. CQSDEL の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000008'	X'00000210'	cqstoken が無効。
X'00000008'	X'00000214'	connecttoken が無効。
X'00000008'	X'00000218'	FUNC が無効。
X'00000008'	X'00000250'	count が無効。
X'00000008'	X'00000254'	listaddress が無効。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000008'	X'00000288'	リスト・バージョンが無効。
X'0000000C'	X'00000300'	要求は少なくとも 1 つのリスト項目については成功したが、すべてではない。個々のエラーについては、compcode を参照してください。

表 10. CQSDEL の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'0000000C'	X'00000304'	要求はすべてのリスト項目について失敗した。個々のエラーについては、 <i>compcode</i> を参照してください。
X'00000010'	X'00000400'	この構造には CQSRSYNC が必須。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。
X'00000014'	X'00000500'	CQS 内部エラー。

## CQSDEREG 要求

CQSDEREG 要求は、クライアントを CQS から登録解除して、CQSTOKEN を無効にします。

### CQSDEREG の形式

#### CQSDEREG の DEREGISTER 機能

CQSDEL 要求は、1 つ以上のデータ・オブジェクトを、キュー構造またはリソース構造から削除します。

▶▶—CQSDEREG—FUNC=DEREGISTER—CQSTOKEN=*cqstokenaddress*—PARM=*parmaddress*————▶▶

▶▶—RETCODE=*returncodeaddress*—RSNCODE=*reasoncodeaddress*————▶▶

#### CQSDEREG 要求の DSECT 機能

CQSDEREG パラメーター・リストの長さ、および CQSDEREG の戻りコードと理由コードのために、プログラムの中に等価 (EQU) ステートメントを組み込むには、CQSDEREG 要求の DSECT 機能を使用します。

▶▶—CQSDEREG—FUNC=DSECT————▶▶

### CQSDEREG の使用法

CQSDEREG 要求は、クライアントを CQS から登録解除して、CQSTOKEN を無効にします。この要求を出す前に、クライアントは CQSDISC 要求を出して、クライアントが接続しているすべての構造から切断しておく必要があります。この要求が正常に完了すると、新しい CQSTOKEN を入手する CQSREG 要求が作成されるまで、CQS に対して後続の要求を行うことができません。

#### パラメーター説明

##### CQSTOKEN=*cqstokenaddress*

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

##### PARM=*parmaddress*

要求が、CQS へパラメーターを渡すために使用するパラメーター・リストのア

ドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSDEREG\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

**RETCODE=returncodeaddress**

CQSDEREG の戻りコードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。CQSDEREG の戻りコードは、このフィールドとレジスタ - 15 の両方に戻されます。

**RSNCODE=reasoncodeaddress**

CQSDEREG の理由コードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。CQSDEREG の理由コードは、このフィールドとレジスタ - 0 の両方に戻されます。

**CQSDEREG の戻りコードと理由コード**

以下の表は、CQSDEREG 要求で返される戻りコードと理由コードの組み合わせの一覧です。

表 11. CQSDEREG の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000004'	X'00000104'	クライアントのアドレス・スペース内の CQS のストレージを解放できない。cqstoken は現在無効。
X'00000004'	X'00000108'	z/OS リソース・マネージャー・ルーチンを削除できない。cqstoken は現在無効。
X'00000008'	X'00000210'	cqstoken が無効。
X'00000008'	X'00000248'	CQSDEREG パラメーター・リスト・バージョンが無効です。このエラーは、CQS クライアントおよびクライアントが使用しようとしている CQS アドレス・スペース間のバージョンの差が原因と考えられます。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000010'	X'00000434'	要求がアクティブ。
X'00000014'	X'00000500'	CQS 内部エラー。cqstoken は現在無効。
X'00000014'	X'00000504'	作業域のストレージ割り振りエラー。
X'00000014'	X'00000518'	CQS 内部エラー (ESTAE を作成できない)。
X'00000014'	X'0000053C'	CQS 登録解除モジュール CQSREG10 をロードできない。

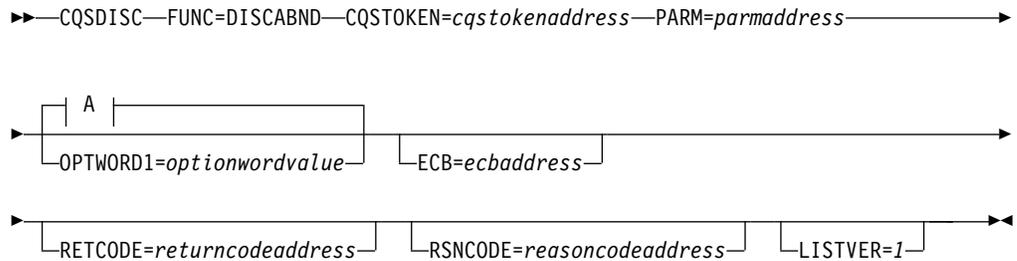
**CQSDISC 要求**

CQSDISC 要求を使用して、クライアントは、1 つ以上のカップリング・ファシリティ構造から切断することができます。

**CQSDISC の形式**

CQSDISC の DISCABND 機能

クライアントが異常終了している間に、すべてのカップリング・ファシリティ構造に対するクライアント接続を終了させるには、CQSDISC 要求の DISCABND 機能を使用します。

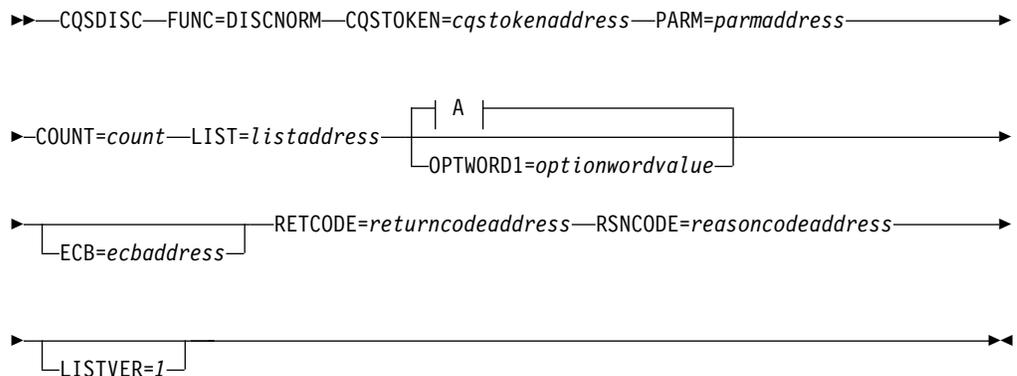


**A:**



### CQSDISC の DISCNORM 機能

クライアントが正常終了している間に、1 つ以上のカップリング・ファシリティ構造に対するクライアント接続を終了させるには、CQSDISC 要求の DISCNORM 機能を使用します。



**A:**



### CQSDISC の DSECT 機能

CQSDISC パラメーター・リストの長さ、CQSDISC の戻りコードと理由コード、および OPTWORD1 パラメーターの作成に使用できるリテラルのために、プログラムに等価 (EQU) ステートメントを組み込むには、CQSDISC 要求の DSECT 機能を使用します。

▶▶—CQSDISC—FUNC=DSECT—◀◀

## CQSDISC の使用法

制約事項: CQSDISC 要求は、リソース構造に対する構造属性をサポートしません。

CQSDISC 要求を使用して、クライアントは、1 つ以上のカップリング・ファシリティ構造から切断することができます。CQS は、構造に関連したクライアント・リソースを切断します。クライアントは、CQS から完全に切断するには、CQSDEREG 要求を出す必要があります。

CQSDISC FUNC=DISCABND 要求は、クライアントが異常終了するときに使用され、すべてのカップリング・ファシリティ構造に対するクライアント接続を終了させます。

CQSDISC FUNC=DISCNORM 要求は、クライアントが正常終了するときに使用され、1 つ以上のカップリング・ファシリティ構造に対するクライアント接続を終了させます。クライアントは、1 つ以上のリスト項目が入っている切断リストを指定しますが、その中の各項目は、別々の切断要求になります。各構造の切断が完了すると、その構造の接続トークンが無効になり、クライアントによる使用ができなくなります。

パラメーター説明

### COUNT=*count*

切断リスト内のリスト項目の数を指定する、4 バイトの入力パラメーター。

### CQSSHUT=YES | NO

すべてのクライアントが切断された後に CQS アドレス・スペースをシャットダウンするかどうかを指示する入力パラメーター。

CQSSHUT=YES が指定されている場合、新しいクライアントが継続して CQSCONN 要求を出すことができます。CQSSHUT FUNC=QUIESCE 要求を使用すれば、新しいクライアントが CQSCONN 要求を出すのを防ぐことができます。

CQSSHUT パラメーターは、OPTWORD1 パラメーターが指定されているときは使用できません。CQSSHUT の代わりに OPTWORD1 を指定する場合、次の等価 (EQU) 記号を使用して、OPTWORD1 パラメーターの値を生成することができます。

```
CQSDISC_SHUTYEQX    CQSSHUT=YES
CQSDISC_SHUTNEQX    CQSSHUT=NO
```

### CQSTOKEN=*cqstokenaddress*

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

### **ECB=ecbaddress**

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

### **LIST=listaddress**

1 つ以上の項目が入っている切断リストのアドレスを指定する 4 バイトの入力パラメーター。各項目は、クライアントをカップリング・ファシリティ構造から切断する独立した要求です。各項目の一部のフィールドは、CQSDISC 要求の前にクライアントが初期化しておく必要があります。他のフィールドは、CQSDISC 要求の完了時に CQS によって戻されます。

CQSDISCL リスト項目の DSECT は、リスト項目をマップし、クライアントが使用することができます。複数のリスト項目は、連続するストレージに常駐させる必要があります。

各リスト項目の内容は次の通りです。

#### **connecttoken**

この CQS が管理する特定のカップリング・ファシリティ構造に対するクライアントの接続を一意的に識別する接続トークンを指定する、16 バイトの入力パラメーター。接続トークンは、CQSCONN 要求によって戻されます。このパラメーターは必須です。

#### **structureattributes**

構造属性が入っている 4 バイトの入力パラメーター・フィールド。

**+0** フラグ・バイト 1 で、ビットの定義は以下の通りです。

**X'80'** 0 に設定した場合、CQS はその構造の構造チェックポイントを実行するべきでないことを指示します。

1 に設定した場合、CQS はその構造の構造チェックポイントを実行するべきであることを指示します。

**X'40'** 0 に設定した場合、未了作業 (ロック・オブジェクト) のある構造については、CQS が切断処理を行うべきでないことを指示します。未了作業が見つかった場合、CQS は、要求に関して *compcode* フィールドに完了コード X'00000008' を設定し、X'0000000C' の戻りコードと X'00000300' または X'00000304' の理由コードを返します。

1 に設定した場合、CQS は、未了作業 (ロック・オブジェクト) のある構造についても、構造からの切断をおこなうべきであることを指示します。未了作業が見つかった場合、切断処理で他にエラーが起これなければ、CQS は、要求に関して *compcode* フィールドに完了コード X'00000008' を設定し、X'00000004' の戻りコードと X'00000140' の理由コードを返します。戻りコードと理由コードは警告のみであり、切断処理はまだ実行中であることに注意してください。

このバイトのその他のビットは使用されておらず、ゼロにセットしておく必要があります。

**+1** 次の 3 バイトは使用されておらず、ゼロに設定しておく必要があります。

### **compcode**

要求からの完了コードを受け取るための 4 バイトの出力フィールド。  
完了コードには、以下のものがあります。

**X'00000000'**

要求が正常に完了した。

**X'00000004'**

*connecttoken* が無効。

**X'00000008'**

クライアントには構造に関する未了作業がある。 *structureattributes* パラメーターの最初のバイトの X'40' ビットが 1 に設定されている場合、その構造の切断処理は正常に実行されており、この完了コードは通知のためです。

X'40' ビットがゼロの場合、この構造に対する切断処理は行われておらず、CQS クライアントは未了作業を完了させてから先に進む必要があります。

**X'0000000C'**

構造属性は、リソース構造に対して許可されていません。

### **LISTVER=1 | listversion**

このリスト・バージョンに対する等価を指定する入力パラメーター。CQSDISC リスト・バージョンのためにプログラムの中に等価 (EQU) ステートメントを組み込むには、CQSDISC 要求の DSECT 機能を使用します。

### **OPTWORD1=optionwordvalue**

この要求に対するリテラルを指定する 4 バイトの入力パラメーター。このパラメーターは、CQSSHUT の代わりとして使用できます。このリテラル値に対する等価 (EQU) ステートメントは、CQSSHUT パラメーターの説明のところにリストされています。EQU ステートメントは、DSECT 機能を使用しても生成できます。OPTWORD1 パラメーターは、CQSSHUT が指定されている場合は使用できません。

要件: OPTWORD1 パラメーターをコーディングする場合、このマクロがサポートする各リテラル値ごとに 1 つの等価値で構成される値を渡す必要があります。

### **PARM=parmaddress**

要求が、CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSDISC\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

### **RETCODE=returncodeaddress**

CQSDISC の戻りコードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

### RSNCODE=reasoncodeaddress

CQSDISC の理由コードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

## CQSDISC の戻りコードと理由コード

以下の表は、CQSDISC 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSDISC FUNC=DSECT 要求を使用します。

表 12. CQSDISC の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000004'	X'00000130'	要求された構造に関しては、要求が正常に完了した。クライアントはまだ追加カップリング・ファシリティ構造に接続されています。
X'00000004'	X'00000140'	要求された構造に関しては、要求が正常に完了した。少なくとも 1 つの構造にこのクライアントに関する未了作業がありましたが、クライアントは、CQSDISC で未了作業があっても切断処理が可能であることを指示しています。このような構造の完了コード・フィールドには、X'00000008' が入ります。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000218'	FUNC が無効。
X'00000008'	X'00000250'	<i>count</i> が無効。
X'00000008'	X'00000254'	<i>listaddress</i> が無効。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000008'	X'00000288'	リスト・バージョンが無効。
X'0000000C'	X'00000300'	要求は少なくとも 1 つのリスト項目については成功したが、すべてではない。個々のエラーについては、 <i>compcode</i> を参照してください。
X'0000000C'	X'00000304'	要求はすべてのリスト項目について失敗した。個々のエラーについては、 <i>compcode</i> を参照してください。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。

## CQSINFRM 要求

CQSINFRM 要求は、特定のカップリング・ファシリティ構造上の 1 つ以上のキューに関するインタレストを登録、あるいは登録解除します。

### CQSINFRM の形式

#### CQSINFRM の DSECT 機能

CQSINFRM パラメーター・リストの長さ、および CQSINFRM の戻りコードと理由コードのために、プログラムに等価 (EQU) ステートメントを組み込むには、CQSINFRM 要求の DSECT 機能を使用します。

▶▶—CQSINFRM—FUNC=DSECT—▶▶

### CQSINFRM の INFORM 機能

特定のカップリング・ファシリティ構造上の 1 つ以上のキューに関するクライアントのインタレストを登録するには、CQSINFRM 要求の INFORM 機能を使用します。

▶▶—CQSINFRM—FUNC=INFORM—CQSTOKEN=cqstokenaddress—PARAM=parmaddress—▶▶

▶—COUNT=count—LISTSIZE=listsize—LIST=listaddress—▶  
└—ECB=ecbaddress—┘

▶—RETCODE=returncodeaddress—RSNCODE=reasoncodeaddress—▶

└—LISTVER=1—┘  
└—LISTVER=listversion—┘

### CQSINFRM の UNINFORM 機能

以前にインタレストを登録した特定のカップリング・ファシリティ構造上の 1 つ以上のキューに関するクライアントのインタレストの登録を解除するには、CQSINFRM 要求の UNINFORM 機能を使用します。

▶▶—CQSINFRM—FUNC=UNINFORM—CQSTOKEN=cqstokenaddress—PARAM=parmaddress—▶▶

▶—COUNT=count—LISTSIZE=listsize—LIST=listaddress—▶  
└—ECB=ecbaddress—┘

▶—RETCODE=returncodeaddress—RSNCODE=reasoncodeaddress—▶

└—LISTVER=1—┘  
└—LISTVER=listversion—┘

### CQSINFRM の使用法

特定のカップリング・ファシリティ構造上の 1 つ以上のキューに関するインタレストを登録、あるいは登録解除するには、クライアントは CQSINFRM 要求を使用します。キューが空から空でない状態になると、CQS は、構造通知クライアント出口ルーチンをスケジューリングすることによって、そのキューに対してインタレストを登録したすべてのクライアントに、状況の変化を通知します。

制約事項: CQSINFRM 要求は、リソース構造に対してサポートされていません。

クライアントは、CQSREAD または CQSBRWSE 要求を出して、キューからデータを検索することができます。クライアントは、CQSPUT、CQSMOVE、または CQSUNLCK 要求を使用して、キュー上でデータ・オブジェクトを使用できるようにすることができます。

あるキューについてのインタレストを登録したクライアントは、そのキューが空から空ではない状態へ変化した場合、または CQSINFRM 要求が出されたときにデータ・オブジェクトがそのキュー上で使用可能な場合だけ、通知を受けます。クライアントは、追加データ・オブジェクトが非空キューに入れられたときは通知を受けません。

クライアントは、キューのインタレストの登録を取り消した後は、キューの中の 1 つが空から空ではない状態へ変化しても通知を受けなくなります。クライアント通知は CQSINFRM 要求によって非同期的に行われるので、クライアントは、クライアントが CQSINFRM FUNC=UNINFORM 要求を出した時から CQS が要求を処理する時までの間に到着した新しいデータ・オブジェクトに関しては、通知されることを予期しておく必要があります。

パラメーター説明

**COUNT=***count*

構造リスト内の構造リスト項目の数を指定する 4 バイトの入力パラメーター。

**CQSTOKEN=***cqstokenaddress*

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

**ECB=***ecbaddress*

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

**LIST=***listaddress*

構造リストのアドレスを指定する 4 バイトの入力パラメーター。構造リストは連続したストレージに作成され、リストのサイズは LISTSIZE パラメーターを使用して指定する必要があります。構造リストには、クライアントがインタレストを登録または登録解除する各カップリング・ファシリティ構造の項目が入っている必要があります。各構造リスト項目には、クライアントがインタレストを登録または登録解除するキューのリストが入っている必要があります。

構造リスト項目内の各接続トークンおよびキュー・リスト項目内のキュー名は、要求の前に初期化されている必要があります。要求が完了すると、CQS は、構造リストについては構造完了コードを戻し、キュー・リストについてはキュー完了コードを戻します。

CQSINFL リスト項目の DSECT は、キュー・リスト項目と構造リスト項目をマップし、クライアントが使用することができます。複数のリスト項目は、連続するストレージに常駐させる必要があります。

各構造リスト項目には以下のパラメーターが含まれています。

**connecttoken**

CQS および特定のカップリング・ファシリティ構造に対するクライ

アントの接続を一意的に識別する接続トークンを指定する、16 バイトの入力パラメーター。接続トークンは、CQSCONN 要求によって戻されます。このパラメーターは必須です。

#### **structurecompletioncode**

構造に関する CQSINFRM 要求の完了コードを受け取るための、4 バイトの出力フィールド。構造完了コードには、以下のものがあります。

**X'00000000'**

要求が正常に完了した。

**X'00000004'**

すべてのキューに関して、要求が正常に完了した。少なくとも 1 つのキューに作業が存在します。キュー完了コードを参照して、どのキューに作業が存在するかを判別してください。

**X'00000010'**

*connecttoken* が無効。

**X'00000014'**

*queuelistcount* が無効。

**X'00000018'**

通知出口ルーチンが存在しない。構造に対する CQSCONN 要求で、構造通知出口ルーチンが指定されていませんでした。

**X'00000020'**

要求は少なくとも 1 つのリスト項目については正常に完了したが、*queelist* 内のすべてのキューについてではない。個々のエラーについては、*queuecompletioncode* を参照してください。

**X'00000024'**

要求は *queelist* 内のすべてのキューについて失敗した。個々のエラーまたは成功については、*queuecompletioncode* を参照してください。

**X'00000030'**

この構造には CQSRSYNC が必須。

**X'00000034'**

CQSINFRM は、リソース構造に対して許可されていない。

#### **queelistcount**

キュー・リスト内のキューの数を指定する、4 バイトの入力パラメーター。このパラメーターは必須です。

推奨事項: 最適なパフォーマンスを得るためには、多数のキューにインタレストを登録するクライアントが複数の CQSINFRM 要求を発行する場合に、それぞれの要求がリストするキューが、1024 を超えないようにする必要があります。

#### **queelist**

1 つ以上のキュー・リストが入った可変長入力域。クライアントによって作成されたキュー・リストには、クライアントがインタレストを登録

または登録解除する構造上の各キューの項目が入っている必要があります。キュー名は、要求の前に初期化しておく必要があります。このパラメーターは必須です。

各キュー・リスト項目の内容は、次の通りです。

**queuename**

クライアントがインタレストを登録するキューの名前が入った 16 バイトの入力フィールド。このパラメーターは必須です。

**queuerequestflag**

この CQSINFRM 要求用としてセットできる、このキュー固有のフラグを含む 1 バイトの入力フィールド。

**X'80'** クライアントが CQSINFRM FUNC=INFORM 要求を出す時点で、キュー上にデータ・オブジェクトがある場合は、クライアント通知出口ルーチン呼び出します。CQSINFRM FUNC=INFORM 要求のみに適用されません。

**queuecompletioncode**

指定されたキューの完了コードを受け取るための 4 バイトの出力フィールド。完了コードには、以下のものがあります。

**X'00000000'**

要求が正常に完了した。

**X'00000040'**

キュー上に作業が存在する。

**X'00000044'**

*queuename* が無効。

**X'00000048'**

CQS 内部エラー。

**X'00000050'**

構造がいっぱい。キュー登録用として使用可能なイベント・モニター制御 (EMC) が無い。

**X'00000054'**

構造がアクセス不能。要求を再試行してください。

**LISTSIZE=*listsiz***

構造リストのサイズを指定する 4 バイトの入力パラメーター。クライアントは構造リストを作成し、このフィールドにその構造リストのサイズを指定する必要があります。

**LISTVER=1 | *listversion***

このリスト・バージョンに対する等価を指定する入力パラメーター。CQSINFRM リスト・バージョンのためにプログラムの中に等価 (EQU) ステートメントを組み込むには、CQSINFRM 要求の DSECT 機能を使用します。

**PARM=*parmaddress***

要求が、CQS へパラメーターを渡すために使用するパラメーター・リストの A

ドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSINFRM\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

**RETCODE=returncodeaddress**

CQSINFRM の戻りコードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

**RSNCODE=reasoncodeaddress**

CQSINFRM の理由コードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

**CQSINFRM の戻りコードと理由コード**

以下の表は、CQSINFRM 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSINFRM FUNC=DSECT 要求を使用します。

表 13. CQSINFRM の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000004'	X'00000134'	要求が正常に完了しました。1 つ以上のキューに作業が存在します。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000218'	FUNC が無効。
X'00000008'	X'00000250'	<i>count</i> が無効。
X'00000008'	X'00000254'	<i>listaddress</i> が無効。
X'00000008'	X'00000258'	<i>listsize</i> が無効。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000008'	X'00000288'	リスト・バージョンが無効。
X'0000000C'	X'00000300'	要求は少なくとも 1 つのリスト項目については成功したが、すべてではない。個々のエラーまたは成功については、 <i>structurecompletioncode</i> をチェックしてください。
X'0000000C'	X'00000304'	要求はすべてのリスト項目について失敗した。個々のエラーについては、 <i>structurecompletioncode</i> を参照してください。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。

関連資料:

 [CQS クライアント構造通知出口ルーチン \(出口ルーチン\)](#)

## CQSMOVE 要求

CQSMOVE 要求は、1 つまたはすべてのデータ・オブジェクトを、あるキューから別のキューへ移動します。データ・オブジェクトは、古いキューの最初または最後の位置から、新しいキューの最初または最後の位置へ移動することができます。

### CQSMOVE の形式

#### CQSMOVE の DSECT 機能

CQSMOVE パラメーター・リストの長さ、CQSMOVE の戻りコードと理由コード、および OPTWORD1 パラメーターの作成に使用できるリテラルのために、プログラムに等価 (EQU) ステートメントを組み込むには、CQSMOVE 要求の DSECT 機能を使用します。

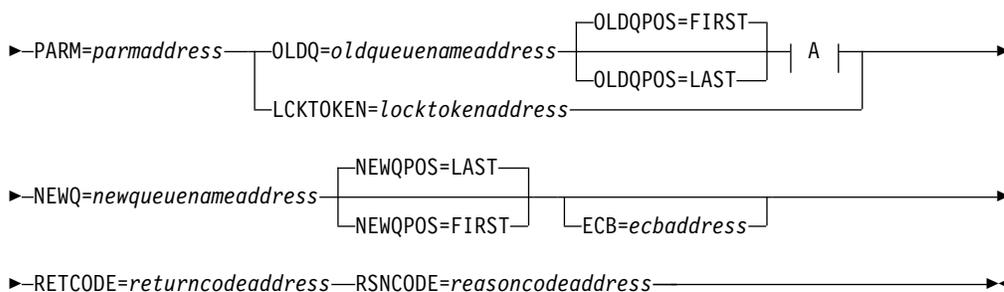
▶▶—CQSMOVE—FUNC=DSECT—▶▶

#### CQSMOVE の MOVE 機能

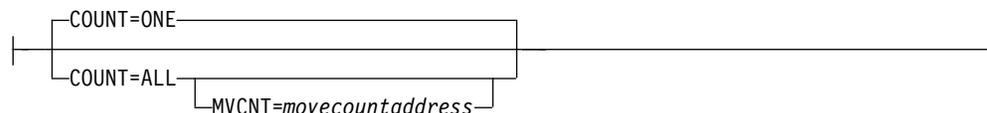
1 つまたはすべてのデータ・オブジェクトをあるキューから別のキューへ移動するには、CQSMOVE 要求の MOVE 機能を使用します。リテラル・パラメーターの各組み合わせごとに、マクロ呼び出しをコーディングする必要があります。

#### リテラル・パラメーターを使用した CQSMOVE の MOVE 機能

▶▶—CQSMOVE—FUNC=MOVE—CQSTOKEN=*cqstokenaddress*—CONTOKEN=*connecttokenaddress*—▶▶

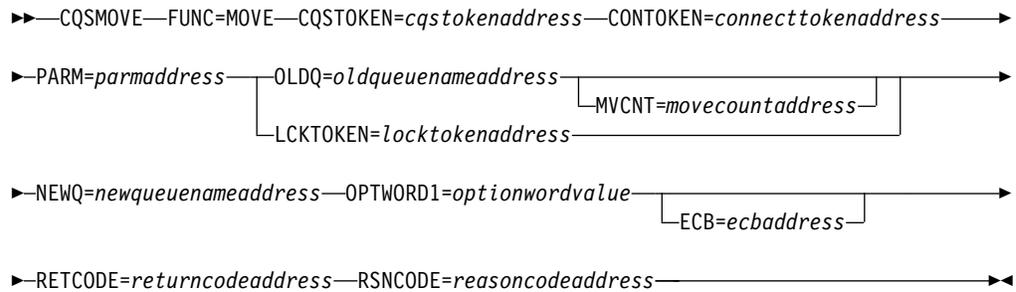


A:



OPTWORD1 パラメーターを使用すれば、単一のマクロ呼び出しをコーディングして、オプションは実行時に設定することができます。ただし、OPTWORD1 パラメーターを使用する場合は、COUNT、NEWQPOS、および OLDQPOS パラメーターは使用できません。

## OPTWORD1 パラメーターを使用した CQSMOVE の MOVE 機能



### CQSMOVE の使用法

制約事項: CQSMOVE 要求は、リソース構造に対してサポートされていません。

CQSMOVE 要求は、1 つまたはすべてのデータ・オブジェクトを、あるキューから別のキューへ移動します。データ・オブジェクトは、古いキューの最初または最後の位置から、新しいキューの最初または最後の位置へ移動することができます。クライアントは、移動させるデータ・オブジェクトを、古いキュー名とキュー位置、またはロック・トークンのどちらかによって識別します。同じキュー名と UOW を持つ複数のオブジェクトを移動しないでください。CQS がオブジェクトをリカバリーできなくなる恐れがあります。

CQS がクライアントに応答する前に CQS またはクライアントに障害が起きると、CQSMOVE 要求が完了しない可能性があります。障害の後には、クライアントは CQS と再接続しなければならず、また移動がコミットされる前に障害が起きたのであれば、CQSMOVE 要求を再び出したり、COUNT=ALL を使用して移動を再開しなければならない場合があります。

パラメーター説明

#### CONTOKEN=connecttokenaddress

この CQS が管理する特定のカップリング・ファシリティ構造に対するクライアントの接続を一意的に識別する、16 バイトの接続トークンのアドレスを指定する入力パラメーター。接続トークンは、CQSCONN 要求によって戻されます。

#### COUNT=ONE | ALL

古いキュー上の移動するデータ・オブジェクトの数を指定する入力パラメーター。クライアントはその中の 1 つまたは全部を移動することができます。

COUNT パラメーターは、OPTWORD1 パラメーターが指定されているときは使用できません。COUNT パラメーターの代わりに OPTWORD1 を指定する場合、次の等価 (EQU) 記号を使用して、OPTWORD1 パラメーターの値を生成することができます。

```
CQSMOVE_CNT1EQUX    COUNT=ONE
CQSMOVE_CNT1EQUX    COUNT=ALL
```

**CQSTOKEN=cqstokenaddress**

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

**ECB=ecbaddress**

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

**LCKTOKEN=locktokenaddress**

移動するロック・データ・オブジェクトの 16 バイトのロック・トークンのアドレスを指定する入力パラメーター。ロック・トークンは、CQSREAD 要求によってロックされたデータ・オブジェクトを一意的に識別します。

**MVCNT=movecountaddress**

移動されたデータ・オブジェクトの数を受け取る 4 バイトのフィールドのアドレスを指定する出力パラメーター。戻りコードまたは理由コードが非ゼロであっても、CQS がいくつかのデータ・オブジェクトを移動している場合があります。

**NEWQ=newqueueaddress**

データ・オブジェクトの移動先である新しいキューの 16 バイトの名前のアドレスを指定する入力パラメーター。

**NEWQPOS=FIRST | LAST**

データ・オブジェクトを移動する新しいキュー上の位置が最初か最後かを指定する入力パラメーター。

OPTWORD1 パラメーターが指定されているときは、NEWQPOS パラメーターを使用できません。NEWQPOS の代わりに OPTWORD1 を指定する場合、次の等価 (EQU) 記号を使用して、OPTWORD1 パラメーターの値を生成することができます。

```
CQSMOVE_NEWQFEQUX  NEWQPOS=FIRST
CQSMOVE_NEWQLEQUX  NEWQPOS=LAST
```

**OLDQ=oldqueueaddress**

データ・オブジェクトの移動元の古いキューの 16 バイトの名前のアドレスを指定する入力パラメーター。

**OLDQPOS=FIRST | LAST**

移動されるデータ・オブジェクトの位置が古いキュー上の最初か最後かを指定する入力パラメーター。

OLDQPOS パラメーターは、OPTWORD1 パラメーターが指定されているときは使用できません。OLDQPOS の代わりに OPTWORD1 を指定する場合、次の等価 (EQU) 記号を使用して、OPTWORD1 パラメーターの値を生成できます。

```
CQSMOVE_OLDQFEQUX  OLDQPOS=FIRST
CQSMOVE_OLDQLEQUX  OLDQPOS=LAST
```

**OPTWORD1=optionwordvalue**

この要求に対するリテラルを指定する 4 バイトの入力パラメーター。このパラメーターは、COUNT、NEWQPOS、および OLDQPOS の代わりとして使用できます。このリテラル値の等価 (EQU) ステートメントは、COUNT、

NEWQPOS、および OLDQPOS パラメーターの説明のところにリストされています。EQU ステートメントは、DSECT 機能を使用しても生成できます。OPTWORD1 パラメーターは、COUNT、NEWQPOS、または OLDQPOS が指定されている場合は使用できません。

要件: OPTWORD1 パラメーターをコーディングする場合、このマクロがサポートする各リテラル値ごとに 1 つの等価値で構成される値を渡す必要があります。

#### **PARM=parmaddress**

要求が、CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSMOVE\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

#### **RETCODE=returncodeaddress**

CQSMOVE の戻りコードを含む 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

#### **RSNCODE=reasoncodeaddress**

CQSMOVE の理由コードが入った 4 バイト・フィールドのアドレスを指定する出力パラメーター。

### **CQSMOVE の戻りコードと理由コード**

以下の表は、CQSMOVE 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSMOVE FUNC=DSECT 要求を使用します。

表 14. CQSMOVE の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000004'	X'00000128'	指定されたキュー名には移動するデータ・オブジェクトがない。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000214'	<i>connecttoken</i> が無効。
X'00000008'	X'00000218'	FUNC が無効。
X'00000008'	X'0000021C'	<i>locktoken</i> が無効。
X'00000008'	X'00000220'	キュー名が無効です。
X'00000008'	X'00000224'	バッファー・アドレスが無効。
X'00000008'	X'0000027C'	CQSMOVE は、リソース構造に対して許可されていない。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000010'	X'00000400'	この構造には CQSRSYNC が必須。
X'00000010'	X'00000404'	構造がアクセス不能。後で要求を再試行してください。

表 14. CQSMOVE の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'00000010'	X'00000414'	宛先キューがいっぱいなために、データ・オブジェクトを移動できない。他のキューについての CQSMOVE 要求は許可されます。
X'00000010'	X'0000041C'	要求は保留中。完了するには、構造リカバリーまたは CQS 再始動が必要である可能性があります。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。
X'00000010'	X'00000440'	ロックされた (リカバリー不能) データ・オブジェクトが再作成のために失われた。
X'00000014'	X'00000500'	CQS 内部エラー。

## CQSPUT 要求

CQSPUT 要求を使用して、クライアントは、データ・オブジェクトをキューに入れることができます。このデータ・オブジェクトは、1 作業単位について 1 つだけあるものにするか、または 1 作業単位について一連のデータ・オブジェクトのうちの 1 つにすることが可能です。

### CQSPUT の形式

#### CQSPUT の ABORT 機能

リカバリー可能作業単位に関連付けられた、コミットされていないデータ・オブジェクトをキューからすべて除去するには、CQSPUT 要求の ABORT 機能を使用します。

```

▶▶—CQSPUT—FUNC=ABORT—CQSTOKEN=cqstokenaddress—CONTOKEN=connecttokenaddress—▶▶
▶—PARAM=paramaddress—PUTTOKEN=puttokenaddress—▶▶
└──ECB=ecbaddress──┘
▶—RETCODE=returncodeaddress—RSNCODE=reasoncodeaddress—▶▶
  
```

#### CQSPUT の DSECT 機能

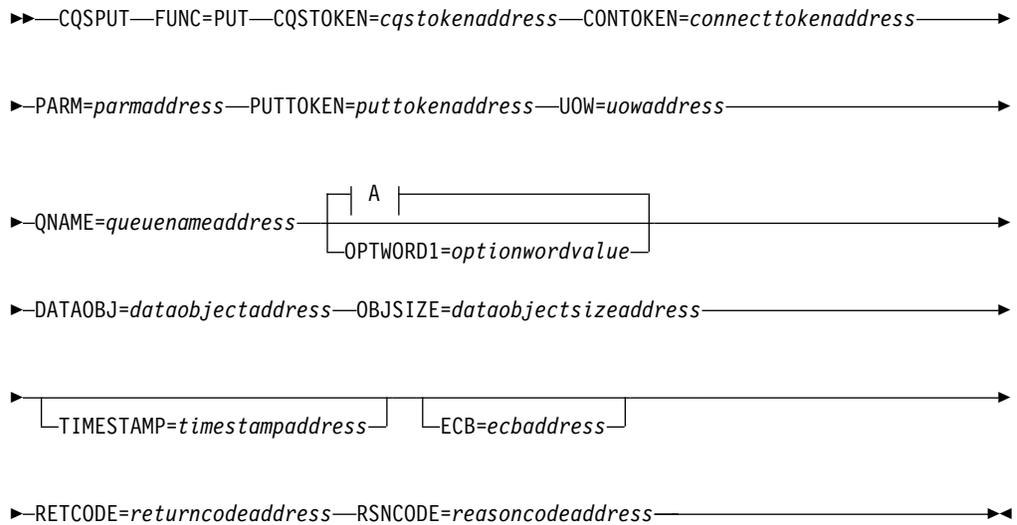
CQSPUT パラメーター・リストの長さ、CQSPUT の戻りコードと理由コード、および OPTWORD1 パラメーターの作成に使用できるリテラルのために、プログラムに等価 (EQU) ステートメントを組み込むには、CQSPUT 要求の DSECT 機能を使用します。

```

▶▶—CQSPUT—FUNC=DSECT—▶▶
  
```

#### CQSPUT の PUT 機能

データ・オブジェクトをキューに入れるには、CQSPUT 要求の PUT 機能を使用します。



**A:**



**B:**



**C:**



**D:**



## CQSPUT の使用法

制約事項: CQSPUT 要求は、リソース構造に対してサポートされていません。

CQSPUT 要求を使用して、クライアントは、データ・オブジェクトをキューに入れることができます。このデータ・オブジェクトは、1 作業単位について 1 つだけあるものにするか、または 1 作業単位について一連のデータ・オブジェクトのうちの 1 つにすることが可能です。データ・オブジェクトは、キューの先頭または最後に

追加できます。データ・オブジェクトは、キューに入れられた後、そのキューに対するアクセス権を持つどのクライアントでも使用可能になります。

複数のオブジェクトを、作業単位のための同じキューに入れることができます。これらのオブジェクトは、移動したり (CQSMOVE 要求) オーバーフロー構造への移動を許可したり (CQSCONN 要求) しないでください。CQS がオブジェクトをリカバリーできなくなります。

1 つの作業単位 (UOW) が複数のデータ・オブジェクトで構成されている場合、それらがすべて同一キュー上にあるなら、その UOW に対するデータ・オブジェクトがまだそのキュー上に一部しか無く、UOW がまだコミットされていなくても、CQS は、最初のデータ・オブジェクトをそのキューに入れるときに、他のクライアントに対して、そのキューにインタレストを登録したことを通知します。

推奨事項: クライアントが不完全なデータを検索しないようにするためには、ある作業単位 (UOW) に対する最後のデータ・オブジェクトはその UOW に対するそれまでのデータ・オブジェクトとは異なるキューに置いて、クライアントがそのキューだけにインタレストを登録するようにしてください。

ある作業単位に対するキュー上にデータ・オブジェクトを置く最初の要求が、その作業単位がリカバリー可能かリカバリー不能かを決定します。クライアントの障害時、CQS 障害時、構造のコピー時、あるいは構造リカバリー時にデータ・オブジェクトに対してとられる処置は、その作業単位がリカバリー可能かどうか、リカバリー可能であれば、コミット済みかどうかによって決まります。以下の表に、それぞれの場合の処置を示しています。

データ・オブジェクトがキューに配置されると、そのデータ・オブジェクトのタイム・スタンプが格納されます。タイム・スタンプのソースは、CQSPUT= 要求に TIMESTAMP= が使用されているかどうかで決まります。TIMESTAMP= が CQSPUT 要求に指定された場合、TIMESTAMP= に指定された値は、データ・オブジェクトと一緒に保管されます。CQSPUT 要求に TIMESTAMP= が指定されていない場合は、現在時刻を表すタイム・スタンプが生成され、データ・オブジェクトと一緒に格納されます。タイム・スタンプがキュー内の最も古いデータ・オブジェクトに関連付けられている場合や、キュー内の最新のデータ・オブジェクトに関連付けられている場合は、そのタイムスタンプは、CQSQUERY FUNC=QTYPE 要求時に戻されます。

表 15. 障害または構造アクティビティの結果、データ・オブジェクトに行われる処置

	リカバリー不能	リカバリー可能でコミットされていない	リカバリー可能でコミット済み
クライアント障害	リカバリー不能作業単位のキューにあるすべてのデータ・オブジェクトがキューに残される。	コミットされていない作業単位に属するキューにあるすべてのデータ・オブジェクトが、クライアントの終了時に削除される。	その作業単位のキューにあるすべてのデータ・オブジェクトはキューに残される。

表 15. 障害または構造アクティビティの結果、データ・オブジェクトに行われる処置 (続き)

	リカバリー不能	リカバリー可能でコミットされていない	リカバリー可能でコミット済み
CQS 障害	リカバリー不能作業単位のデータ・オブジェクトで、キューに正常に入れられたものは、すべてキューに残される。障害発生時に CQS がデータ・オブジェクトをキューに入れる処理中だった場合、そのデータ・オブジェクトは CQS の再始動にリカバリーされません。	コミットされていない作業単位の属するキューにあるすべてのデータ・オブジェクトが、CQS の再始動時に削除される。	コミット済み作業単位の属するキューにあるすべてのデータ・オブジェクトが、キューに残される。障害発生時に CQS がキュー上に作業単位の最後のデータ・オブジェクトを置く処理を実行中だった場合、CQS の再始動で、データ・オブジェクトが確実にキュー上にあるようになります。
構造のコピー	リカバリー不能作業単位のデータ・オブジェクトが新しい構造にコピーされる。	リカバリー可能作業単位のすべてのデータ・オブジェクトが、作業単位がコミット済みであるかないかにかかわらず、新しい構造にコピーされる。	リカバリー可能作業単位のすべてのデータ・オブジェクトが、新しい構造にコピーされる。
構造リカバリー	リカバリー不能作業単位のキューにあるデータ・オブジェクトは、新しい構造でリカバリーされない。	リカバリー可能作業単位のキューにあるすべてのデータ・オブジェクトが、作業単位がコミット済みであるかないかにかかわらず、新しい構造でリカバリーされる。	リカバリー可能作業単位のキューにあるすべてのデータ・オブジェクトが、新しい構造でリカバリーされる。

CQSPUT FUNC=FORGET 要求を実行すると、CQSPUT FUNC=PUT 要求はすべて終了され、CQS は作業単位に関して所有している内部情報を廃棄します。作業単位は PUT トークンによって識別されます。クライアントは、その作業単位に関して出された最後の CQSPUT FUNC=PUT 要求からの応答を受け取った後、この要求を出す必要があります。作業単位がリカバリー可能ではあるがコミット済みでない場合、CQSPUT FUNC=FORGET 要求はリジェクトされます。

CQSPUT FUNC=ABORT 要求は、リカバリー可能作業単位に関連したコミットされていないデータ・オブジェクトをすべてキューから除去します。作業単位は PUT トークンによって識別されます。作業単位がリカバリー不能な場合、または作業単位がリカバリー可能だがすでにコミット済みの場合、要求はリジェクトされます。

例: キュー上の 1 つの作業単位について、単一のオブジェクトを書き込むには、次の要求を出します。

```
CQSPUT FUNC=PUT,COMMIT=YES,...
⋮
CQSPUT FUNC=FORGET,...
```

キュー上の 1 つの作業単位について、複数のオブジェクトを書き込むには、次の要求を出します。

```
CQSPUT FUNC=PUT,COMMIT=NO,...
⋮
CQSPUT FUNC=PUT,COMMIT=NO,...
⋮
CQSPUT FUNC=PUT,COMMIT=YES,...
```

⋮  
CQSPUT FUNC=FORGET,...

パラメーターの説明:

#### **COMMIT=**YES | NO

リカバリー可能作業単位をコミットするかどうかを指示する入力パラメーター。  
1 つ以上のデータ・オブジェクトを、リカバリー可能作業単位のキュー上に置く  
ことができます。

COMMIT= パラメーターは、リカバリー可能作業単位のみ適用され、  
RECOVERABLE=YES が指定されている場合のみ有効です。  
RECOVERABLE=NO が指定された場合、パラメーターは無視されます。

ある作業単位に対して出される最後の (または唯一の) CQSPUT FUNC=PUT  
要求には、COMMIT=YES を指定する必要があります (単独または  
OPTWORD1 の一部として)。複数のデータ・オブジェクトが 1 つの作業単位  
のキュー上にある場合、一連の CQSPUT FUNC=PUT 要求のうちの最後を除く  
すべての要求に COMMIT=NO を指定する必要があります。 COMMIT=YES  
は、最後の CQSPUT FUNC=PUT 要求に指定する必要があります。

COMMIT パラメーターは、OPTWORD1 パラメーターが指定されているとき  
は使用できません。COMMIT の代わりに OPTWORD1 を指定する場合、次の  
等価 (EQU) 記号を使用して OPTWORD1 パラメーターの値を生成することが  
できます。

```
CQSPUT_CMTYEQUX  COMMIT=YES  
CQSPUT_CMTNEQUX  COMMIT=NO
```

#### **CONTOKEN=***connecttokenaddress*

この CQS が管理する特定のカップリング・ファシリティー構造に対するクラ  
イアントの接続を一意的に識別する、16 バイトの接続トークンのアドレスを指  
定する入力パラメーター。接続トークンは、CQSCONN 要求によって戻されま  
す。

#### **CQSTOKEN=***cqstokenaddress*

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登  
録トークンのアドレスを指定する入力パラメーター。登録トークンは、  
CQSREG 要求によって戻されます。

#### **DATAOBJ=***dataobjectaddress*

指定されたキューに入れるクライアント・データ・オブジェクトのアドレスを指  
定する 4 バイトの入力パラメーター。

#### **ECB=***ecbaddress*

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指  
定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期  
的に処理され、それ以外の場合は同期的に処理されます。

#### **LOCAL=**NO | N | YES | Y

クライアントがデータのローカル・コピーを保持すべきかどうかを指示する入力  
パラメーター。

**NO** クライアントが、指定されたクライアント・キューに CQS がデータ・オブ  
ジェクトを入れ、他の CQS も使用できるようにすることを希望しているこ  
とを示す。

## YES

クライアントが、共用キューに CQS がデータ・オブジェクトを入れ、そのオブジェクトをロックするように希望していることを示す。LOCAL=YES は、クライアントがローカル・バッファにデータ・オブジェクトのローカル・コピーを保持することも指示します。

データ・オブジェクトのローカル・コピーを保持することにより、クライアントは、共用キューを使用した場合のパフォーマンスへの影響を低減できます。共用キューにデータ・オブジェクトを保持した場合は、クライアントの障害の場合もリカバリーできます。データ・オブジェクトをロックすると、他のクライアントは使用できません。

クライアントは、データを処理する (データ・オブジェクトのロック・トークンを検索し、CQS にクライアントがデータを処理中であることを通知する) には、CQSREAD LOCAL=YES 要求を出す必要があります。

CQSREAD 要求では、クライアントにローカル・コピーがあるので、データ・オブジェクトがクライアントに戻されることはありません。クライアントが CQSREAD LOCAL=YES 要求を出さない場合、クライアントと CQS の間の接続が失われると、CQS はデータ・オブジェクトをアンロックしてどのクライアントも使用できるようにします。

LOCAL パラメーターは、OPTWORD1 パラメーターが指定されているときは使用できません。LOCAL の代わりに OPTWORD1 を使用する場合、次の等価 (EQU) ステートメントを使用して OPTWORD1 パラメーターの値を生成することができます。

```
CQSPUT_LCLYEQUX LOCAL=YES  
CQSPUT_LCLNEQUX LOCAL=NO
```

## **OBJSIZE=***dataobjectsizedress*

キューに入れるクライアント・データ・オブジェクトのサイズを保持する 4 バイト・エリアのアドレスを指定する入力パラメーター。指定できる最大サイズは 61312 バイト (X'EF80') です。

## **OPTWORD1=***optionwordvalue*

この要求に対するリテラルを指定する 4 バイトの入力パラメーター。このパラメーターは、COMMIT、LOCAL、QPOS、および RECOVERABLE の代わりとして使用できます。このリテラル値の等価 (EQU) ステートメントは、COMMIT、LOCAL、QPOS、および RECOVERABLE パラメーターの説明のところにリストされています。EQU ステートメントは、DSECT 機能を使用しても生成できます。OPTWORD1 パラメーターは、COMMIT、LOCAL、QPOS、または RECOVERABLE が指定されている場合は使用できません。

要件: OPTWORD1 パラメーターをコーディングする場合、このマクロがサポートする各リテラル値ごとに 1 つの等価値で構成される値を渡す必要があります。

## **PARM=***parmaddress*

要求が CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSPUT\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

### **PUTTOKEN=puttokenaddress**

CQS が、1 つの作業単位に対する一連の CQSPUT 要求の関連付けの際に使用する 16 バイト・トークンのアドレスを指定する、4 バイトの入出力パラメーター。一連の CQSPUT 要求のうちの最初の要求の場合、トークンはゼロでなければなりません。更新されたトークンは、CQS によってそれぞれの CQSPUT 要求に戻されます。更新済みトークンは、その作業単位に対する次の CQSPUT 要求で CQS へ戻す必要があります。どの CQSPUT FUNC=FORGET または CQSPUT FUNC=ABORT 要求についても、*puttoken* を CQS へ戻す必要があります。

### **QNAME=queuenameaddress**

データ・オブジェクトを入れるキューの 16 バイトの名前のアドレスを指定する入力パラメーター。キュー・タイプの判別に使用される為、キュー名の最初のバイトはゼロにすることができません。最初のバイトの値が CQS の定義するキュー・タイプの最大値より大きい場合、折り返されて既存のキュー・タイプの中の 1 つになります。ある作業単位に対する最後のデータ・オブジェクトを構造に書き込む場合、その作業単位のそれまでのどのデータ・オブジェクトとも異なるキューに書き込む必要があります。

### **QPOS=LAST | FIRST**

クライアント・データ・オブジェクトを置くキュー上の位置を指定する入力パラメーター。

#### **FIRST**

データ・オブジェクトは、キューの先頭に追加されます。

#### **LAST**

データ・オブジェクトは、キューの終わりに追加されます。

OPTWORD1 パラメーターが指定されているときは、QPOS パラメーターを使用できません。QPOS の代わりに OPTWORD1 を指定する場合、次の等価 (EQU) ステートメントを使用して OPTWORD1 パラメーターの値を生成することができます。

```
CQSPUT_QPOSFEQUX  QPOS=FIRST
CQSPUT_QPOSLEQUX  QPOS=LAST
```

### **RECOVERABLE=YES | NO**

作業単位が CQS によってリカバリー可能かどうかを指定する入力パラメーター。RECOVERABLE=NO は、作業単位がリカバリー不能であることを示します。リカバリー不能作業単位の場合、キューにはデータ・オブジェクトを 1 つだけ置くことができます。RECOVERABLE=YES は、作業単位がリカバリー可能であることを示します。1 つ以上のデータ・オブジェクトを、リカバリー可能作業単位のキュー上に置くことができます。

RECOVERABLE=YES パラメーターは、その作業単位に対して CQSPUT FUNC=PUT 要求が出されるたびに指定する必要があります。その作業単位は、その一連のデータ・オブジェクトのうちの最後のもの (または唯一のもの) がキュー (COMMIT=YES が指定されている) に置かれるまでコミットされません。

RECOVERABLE パラメーターは、OPTWORD1 パラメーターが指定されているときは使用できません。RECOVERABLE の代わりに OPTWORD1 を指定する場合、次の等価 (EQU) ステートメントを使用して OPTWORD1 パラメーターの値を生成することができます。

```
CQSPUT_RECVYEQUX RECOVERABLE=YES
CQSPUT_RECNEQUX RECOVERABLE=NO
```

#### **RETCODE=returncodeaddress**

CQSPUT の戻りコードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

#### **RSNCODE=reasoncodeaddress**

CQSPUT の理由コードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

#### **TIMESTAMP=timestampaddress**

データ・オブジェクトがキューに置かれるときに、データ・オブジェクトと一緒に保管する 8 バイトの STCK 値のアドレスを指定する、4 バイトの入力パラメーター。TIMESTAMP パラメーターが省略された場合、現在時刻がデータ・オブジェクトと一緒に保管されます。

#### **UOW=uowaddress**

作業単位を保持する 32 バイト・エリアのアドレスを指定する入力パラメーター。このパラメーターは、ある作業単位に対して最初に出される（または唯一の）CQSPUT FUNC=PUT 要求の場合は必須です。その作業単位に対して、それ以降 CQSPUT FUNC=PUT 要求が出されても、すべて無視されます。

UOW= パラメーターに値が指定されている場合、PUTTOKEN=0 も指定する必要があります。UOW= パラメーターに指定される値は、すべてゼロであってはならず、また共用キューの中で固有な値でなければなりません。クライアントは、その値が固有であることを確認する責任があります。

## **CQSPUT の戻りコードと理由コード**

以下の表は、CQSPUT 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSPUT FUNC=DSECT 要求を使用します。

表 16. CQSPUT の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000214'	<i>connecttoken</i> が無効。
X'00000008'	X'00000218'	FUNC が無効。
X'00000008'	X'0000021C'	<i>puttoken</i> が無効。
X'00000008'	X'00000220'	<i>queuename</i> が無効。
X'00000008'	X'00000224'	<i>dataobject</i> が無効。
X'00000008'	X'00000228'	<i>dataobjectsize</i> が無効。
X'00000008'	X'00000230'	<i>uow</i> が無効。

表 16. CQSPUT の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'00000008'	X'00000238'	キュー名が固有でない。1 つの作業単位に対するキューに複数のデータ・オブジェクトがある場合、最後のデータ・オブジェクトに割り当てられたキュー名はその作業単位で固有なものでなければならない。
X'00000008'	X'00000260'	CQSPUT FUNC=PUT 要求が出されたが、その作業単位はすでにコミットされている。
X'00000008'	X'00000264'	リカバリー可能作業単位に対して CQSPUT FUNC=FORGET 要求が出されたが、その作業単位はコミットされていなかった。
X'00000008'	X'00000268'	リカバリー不能作業単位に対して CQSPUT FUNC=ABORT 要求が出された。
X'00000008'	X'0000026C'	リカバリー可能作業単位に対して CQSPUT FUNC=ABORT 要求が出されたが、その作業単位はすでにコミットされていた。
X'00000008'	X'00000270'	リカバリー不能であることが CQS にすでに認識されている作業単位に対して、続けて CQSPUT FUNC=PUT 要求が発行された。リカバリー不能作業単位の場合、キューにはデータ・オブジェクトを 1 つだけ置くことができます。
X'00000008'	X'00000274'	前の CQSPUT FUNC=PUT 要求でリカバリー可能であることが示されていた作業単位に対して、RECOVERABLE=NO が指定された。
X'00000008'	X'0000027C'	CQSPUT は、リソース構造に対して許可されていない。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000010'	X'00000400'	この構造には CQSRSYNC が必須。
X'00000010'	X'00000404'	構造がアクセス不能。後で、この要求を再試行してください。
X'00000010'	X'00000414'	queuname のキューがいっぱい。このキュー名については、構造にこれ以上データ・オブジェクトを挿入することはできない。他のキュー名であれば、CQSPUT 要求は許可される。
X'00000010'	X'00000418'	構造がいっぱい。CQSPUT 要求はすべてリジェクトされる。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。
X'00000014'	X'00000500'	CQS 内部エラー。

## CQSQUERY 要求

CQSQUERY 要求は、CQS が管理する 1 つ以上の構造に関する情報または状況を検索します。

### CQSQUERY の形式

#### CQSQUERY の DSECT 機能

CQSQUERY パラメーター・リストの長さ、および CQSQUERY の戻りコードと理由コードのために、プログラムの中に等価 (EQU) ステートメントを組み込むには、CQSQUERY 要求の DSECT 機能を使用します。

▶▶—CQSQUERY—FUNC=DSECT—▶▶

### CQSQUERY の QNAME 機能

CQS が管理する特定のキューに関する情報を検索するには、CQSQUERY 要求の QNAME 機能を使用します。

▶▶—CQSQUERY—FUNC=QNAME—CQSTOKEN=*cqstokenaddress*—▶▶

▶—CONTOKEN=*connecttokenaddress*—PARM=*parmaddress*—COUNT=*count*—▶

▶—LIST=*listaddress*—ECB=*ecbaddress*—RETCODE=*returncodeaddress*—▶

▶—RSNCODE=*reasoncodeaddress*—LISTVER=*1*—▶▶

### CQSQUERY 要求の QRYOBS 機能

キュー名に関する指定されたりストに対するキュー・カウントを検索するには、CQSQUERY 要求の QRYOBS 機能を使用します。

▶▶—CQSQUERY—FUNC=QRYOBS—CQSTOKEN=*cqstokenaddress*—▶▶

▶—CONTOKEN=*connecttokenaddress*—PARM=*parmaddress*—COUNT=*count*—▶

▶—LIST=*listaddress*—BUFFER=*bufferaddress*—BUFSIZE=*buffersize*—▶

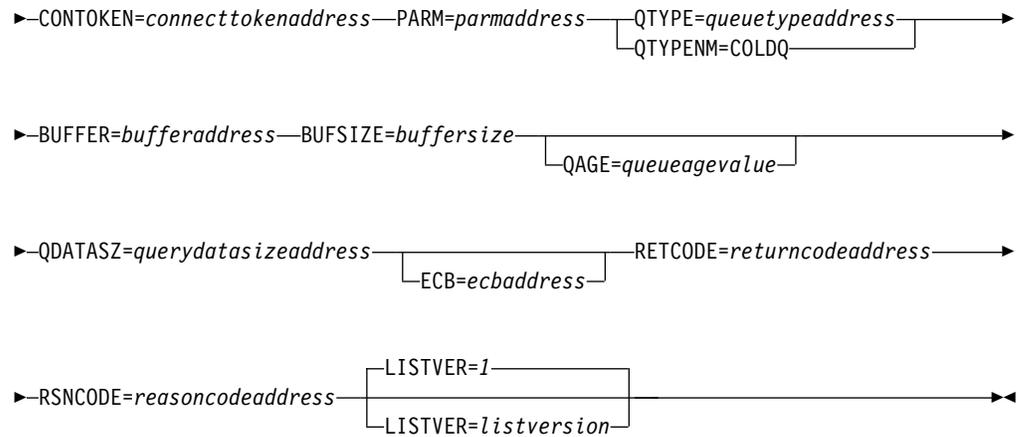
▶—QDATASZ=*querydatasizeaddress*—ECB=*ecbaddress*—RETCODE=*returncodeaddress*—▶

▶—RSNCODE=*reasoncodeaddress*—LISTVER=*1*  
LISTVER=*listversion*—▶▶

### CQSQUERY の QTYPE 機能

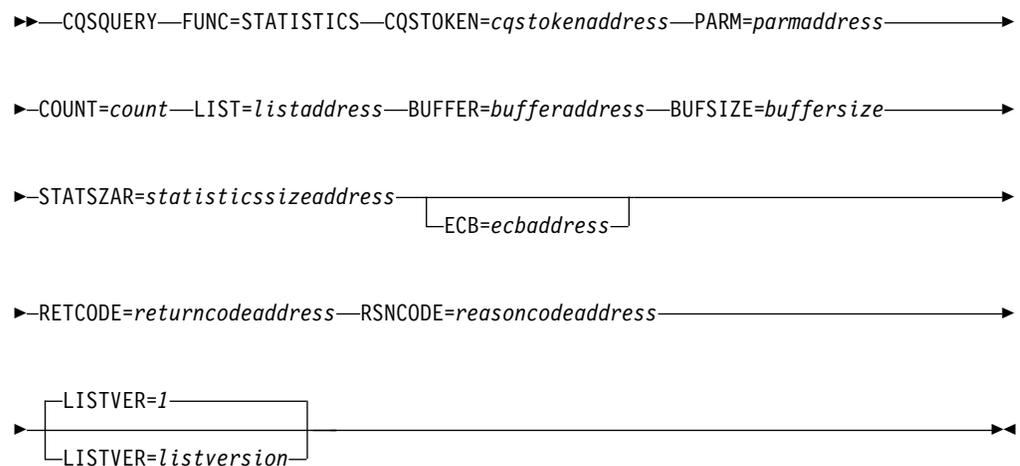
指定されたキュー・タイプ内のすべてまたは一部のキューに関する情報を検索するには、CQSQUERY 要求の QTYPE 機能を使用します。

▶▶—CQSQUERY—FUNC=QTYPE—CQSTOKEN=*cqstokenaddress*—▶▶



### CQSQUERY の STATISTICS

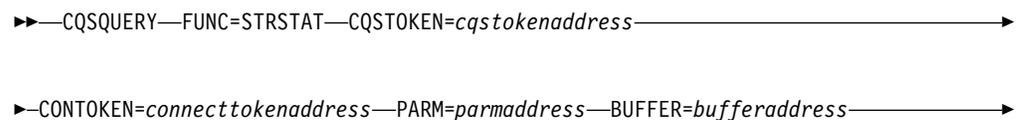
CQS が管理するすべてのキューに関する状況情報を検索するには、CQSQUERY 要求の STATISTICS 機能を使用します。

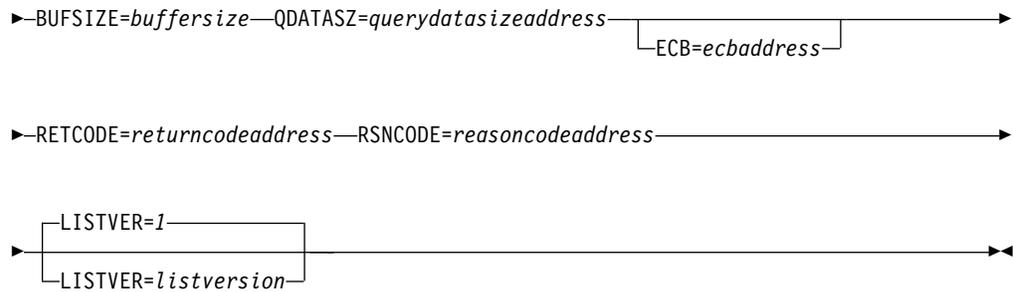


### CQSQUERY の STRSTAT 機能

構造関連の統計を検索するには、CQSQUERY 要求の STRSTAT 機能を使用します。STRSTAT 機能は、構造統計ユーザー出力ルーチンで得られるのと同じ統計データを戻します。

重要: 要求を処理している CQS が構造チェックポイントの最中である場合、現行の構造チェックポイントで戻されるデータは不完全な可能性があります。





## CQSQUERY の使用法

CQSQUERY 要求は、CQS が管理する 1 つ以上の構造に関する情報または状況を検索します。CQSQUERY FUNC=QNAME 要求は、CQS が管理する 1 つ以上の特定のキューに関する情報を検索します。CQSQUERY FUNC=QRYOBS 要求は、1 つ以上の特定キュー、または名前がワイルドカード・パラメーターに一致するキューに対するキュー・カウントを検索します。CQSQUERY FUNC=QTYPE 要求は、指定されたキュー・タイプ内のすべてまたは一部のキューに関する情報を検索します。CQSQUERY FUNC=STATISTICS 要求は、CQS が管理するすべてのキューの状況情報を検索します。CQSQUERY FUNC=STRSTAT 要求は、チェックポイントや再作成などの構造統計を検索しますが、ユーザー出口をコーディングする必要はありません。

制約事項: CQSQUERY FUNC=QNAME、CQSQUERY FUNC=QRYOBS、および CQSQUERY FUNC=QTYPE 要求は、リソース構造に対してはサポートされていません。

### CQSQUERY FUNC=QNAME

CQSQUERY FUNC=QNAME の場合、LIST= で指定された *queuename* のデータ・オブジェクトの数が戻されます。

QAGE パラメーターが指定されている場合、指定されているキューの経過日数より古いキューの情報だけが戻されます。キュー・カウントだけに興味がある場合は、QAGE パラメーターを省略すれば、CQSQUERY 要求のパフォーマンスを上げることができます。

### CQSQUERY FUNC=QRYOBS

CQSQUERY FUNC=QRYOBS の場合、LIST= で指定された *queuename* のデータ・オブジェクトの数が戻されます。リスト内の各キュー名は、最大 16 バイト長まで可能です。キュー名の最初のバイトは、QTYPE として扱われます。各キュー名に対する入力リストには、入力キュー名に一致する各エントリに関して出力にコピーされる、8 バイトのユーザー・データが含まれています。

CQSQUERY FUNC=QRYOBS 出力は、入力リストおよび出力バッファの両方に戻されます。入力リストには、キュー名に対する完了コードが含まれています。完了コードが 0 の場合、入力キュー名およびそれらのキュー・カウントに一致するキュー名は、出力バッファに戻されます。完了コードが非ゼロの場合、出力バッファ内のそのキュー名に対しては、データは渡されません。入力リストには、キュー名に対して検索された合計キュー・カウントが含まれています。キュー名がワイ

ルドカード・パラメーターの場合、このキュー・カウントは、ワイルドカード・パラメーターに一致するすべてのキュー名の合計キュー・カウントです。一致した各キュー名に対するエントリーは、そのキュー名に対するキュー・カウントとともに、出力バッファーにパスされます。指定されたバッファー・サイズが小さすぎる場合、そのバッファーにフィットするデータはパス・バックされ、必要な実際の長さは QDATASZ フィールドにパス・バックされます。

推奨事項: CQSQUERY FUNC=QRYOBS 要求は注意して使用してください。CQS がそのキュー・タイプ上にあるすべてのデータ・オブジェクトを読み取ることになるため、パフォーマンスに重大な影響がもたらされる可能性があります。

### **CQSQUERY FUNC=QTYPE**

CQSQUERY FUNC=QTYPE の場合、キュー・タイプ内のすべてのキューに関する情報が戻されます。この中には、キュー名、データ・オブジェクト数、一番古いデータ・オブジェクトのタイム・スタンプ、および最新のオブジェクトのタイム・スタンプが含まれます。

推奨事項: CQSQUERY FUNC=QTYPE 要求は注意して使用してください。CQS がそのキュー・タイプ上にあるすべてのデータ・オブジェクトを読み取ることになるため、パフォーマンスに重大な影響がもたらされる可能性があります。

CQSQUERY FUNC=QTYPE に関して、バッファー域が要求されたデータのすべてを保留することができるほど大規模でなければ、CQS は以下のことを行います。

- バッファー域に入れることができるだけの完全なレコードを戻す
- 統計データ全体を入れるために必要な長さに QDATASZ をセットする
- 「部分的なデータが戻された」ことを示す理由コードをセットする

これによって、クライアント・プログラムはこの後の要求でもっと大きいバッファーを使用することができます。

### **CQSQUERY FUNC=STATISTICS**

CQSQUERY FUNC=STATISTICS の場合、CQS はクライアント・バッファーに次の情報を戻します。

- 基本構造の現行容量に関する状況
- 基本構造の最大容量 (XES 動的再構成が使用可能な場合)
- 現在の動作モード (通常モード、オーバーフロー・モード、または再作成モード)
- エレメントと項目の比率 (この要求でクライアントによって渡されたバッファーに入れて戻される)

オーバーフロー構造が定義されていて、基本構造の現行動作モードがオーバーフロー・モードの場合、CQS は関連するオーバーフロー構造の現行および最大の容量も戻します。基本構造がオーバーフロー・モードでなく、オーバーフロー構造が定義されている場合、CQS はオーバーフロー構造名とオーバーフロー構造が使用中でないことを示す状況を戻します。

バッファー域が要求されたすべての構造の統計データを入れるだけの十分な大きさを持っていない場合、CQSQUERY FUNC=STATISTICS は、STATSZAR フィールド

ドを単一の統計項目の長さにセットし、「バッファ・サイズが小さすぎる」ことを示す理由コードをセットします。要求の完了に必要なバッファのサイズは、STATSZAR に入れて戻される値に、要求に指定されたリスト項目の数を乗算することによって求めることができます。

### **CQSQUERY FUNC=STRSTAT**

CQSQUERY FUNC=STRSTAT の場合、CQS は次の情報を戻します。

- 構造プロセスの統計
- CQS 要求の統計
- データ・オブジェクトの統計
- キュー名の統計
- z/OS 要求統計
- 構造再作成の統計
- 構造チェックポイントの統計

この機能に関して、バッファ領域が要求されたデータのすべてを保留することができるほど大規模でなければ、CQS は以下のことを行います。これによって、クライアント・プログラムはこの後の要求でもっと大きいバッファを使用することができます。

- バッファ領域に入れることができるだけの完全なレコードを戻す
- 統計データ全体を入れるために必要な長さに QDATASZ をセットする
- 「部分的なデータが戻された」ことを示す理由コードをセットする

以下のキーワードは、CQSQUERY マクロに対して適用されます。ここに提供されている情報の一部が、特定の CQSQUERY 機能に対して適用されることに注意してください。

#### **BUFFER=bufferaddress**

クライアントに渡された情報を保持するバッファのアドレスを指定する 4 バイトの入力パラメーター。

CQSQUERY FUNC=QTYPE の場合、バッファは CQSQRQT DSECT によってマップされます。CQSQUERY FUNC=STATISTICS の場合、バッファは CQSQRYST DSECT によってマップされます。CQSQUERY FUNC=STRSTAT の場合、バッファは CQSQSTAT DSECT によってマップされます。

CQSQUERY FUNC=QRYOBS の場合、バッファは CQSQRQO DSECT によってマップされます。

#### **BUFSIZE=buffer size**

クライアントが渡すバッファのサイズを指定する 4 バイトの入力パラメーター。

#### **CONTOKEN=connecttokenaddress**

この CQS が管理する特定のカップリング・ファシリティー構造に対するクライアントの接続を一意的に識別する、16 バイトの接続トークンのアドレスを指定する入力パラメーター。接続トークンは、CQSCONN 要求によって戻されません。

**COUNT=count**

リスト内の項目の数を指定する、4 バイトの入力パラメーター。

**CQSTOKEN=cqstokenaddress**

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

**ECB=ecbaddress**

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

**LIST=listaddress**

1 つ以上の項目が入っているリストのアドレスを指定する 4 バイトの入力パラメーター。CQSQUERY FUNC=QNAME および CQSQUERY FUNC=QRYOBS 要求では、このリストには情報検索を行うキュー名が含まれます。このリストは、入力および出力パラメーターで構成されています。少なくとも 1 つのリスト項目が必要です。

CQSQRYL リスト項目の DSECT は、リスト項目をマップし、クライアントが使用することができます。複数のリスト項目は、連続するストレージに常駐させる必要があります。

CQSQUERY FUNC=QNAME 要求では、各リスト項目には以下の内容が含まれています。

**comcode**

要求からの完了コードを受け取るための 4 バイトの出力フィールド。完了コードには、以下のものがあります。

**X'00000000'**

要求が正常に完了した。

**X'00000004'**

*queuename* が無効。

**X'00000020'**

構造がアクセス不能。要求を再試行してください。

**X'00000024'**

CQS 内部エラー。

**clientdata**

クライアント・データ・フィールドを指定する 8 バイトの入力パラメーター。このパラメーターはオプションです。CQS は、この項目に保管されたデータを使用しません。

**queuename**

データ・オブジェクト・カウント情報を検索するキュー名を指定する 16 バイトの入力パラメーター。このパラメーターは必須です。

**qcnt** 指定されたキュー名のデータ・オブジェクト・カウントを入れるフィールドを指定する 4 バイトの出力パラメーター。

CQSQUERY FUNC=STATISTICS 要求では、各リスト項目には以下のパラメーターが入ります。

### **comPCODE**

要求からの完了コードを受け取るための 4 バイトの出力フィールド。  
完了コードには、以下のものがあります。

**X'00000000'**

要求が正常に完了した。

**X'00000008'**

*connecttoken* が無効。

**X'0000000C'**

この構造には CQSRSYNC が必須。

**X'00000020'**

構造がアクセス不能。要求を再試行してください。

**X'00000024'**

CQS 内部エラー。

### **clientdata**

クライアント・データ・フィールドを指定する 8 バイトの入力パラメーター。このパラメーターはオプションです。CQS は、この項目に保管されたデータを使用しません。

### **connecttoken**

この CQS が管理する特定のカップリング・ファシリティ構造に対するクライアントの接続を一意的に識別する接続トークンを指定する、16 バイトの入力パラメーター。接続トークンは、CQSCONN 要求によって戻されます。このパラメーターは必須です。

### **outputoffset**

この項目の出力データ域の出力バッファー内でのオフセットを指定する 4 バイトの出力パラメーター。

CQSQUERY FUNC=QRYOBS 要求では、各リスト項目には以下のパラメーターが含まれています。

### **comPCODE**

要求からの完了コードを受け取るための 4 バイトの出力フィールド。  
完了コードには、以下のものがあります。

**X'0000'**

要求が正常に完了した。キュー名およびそれらのキュー・カウントが一致するリソースのリストは、出力バッファーに戻されます。

**X'0004'**

キュー名が無効。

**X'0010'**

キュー名がオブジェクトを持っていません。キュー・カウントがゼロです。

**X'0020'**

キュー名に対するエラーを再試行してください。CQSQUERY

FUNC=QRYOBSJ を再試行して、キュー・カウントを取得してください。出力バッファーに戻された出力が無効である可能性があります。

#### **X'0024'**

CQS 内部エラー。CQSQUERY FUNC=QRYOBSJ を再試行して、キュー・カウントを取得してください。出力バッファーに戻された出力が無効である可能性があります。

#### **clientdata**

クライアント・データ・フィールドを指定する 8 バイトの入力パラメーター。このパラメーターはオプションです。CQS は、この項目に保管されたデータを使用しません。

#### **queuename**

データ・オブジェクト・カウント情報を検索するキュー名を指定する 16 バイトの入力パラメーター。このパラメーターは必須です。キュー名はワイルドカード・パラメーターにすることができます。

**qcnt** 指定されたキュー名のデータ・オブジェクト・カウントを入れるフィールドを指定する 4 バイトの出力パラメーター。キュー名がワイルドカード・パラメーターの場合、このパラメーターはフィールドを指定して、ワイルドカード・パラメーターに一致するすべてのキュー名の合計キュー・カウントをそのフィールドに含ませます。

#### **LISTVER=1 | listversion**

このリスト・バージョンに対する等価を指定する入力パラメーター。CQSQUERY リスト・バージョンのためにプログラムの中に等価 (EQU) ステートメントを組み込むには、CQSQUERY 要求の DSECT 機能を使用します。

#### **PARM=parmaddress**

要求が、CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSQUERY\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

#### **QAGE=queueageaddress**

キューの経過日数を入れるための 4 バイト・フィールドのアドレスを指定する入力パラメーター。queueage の有効な値は X'0' ~ X'16D'(10 進で 0 ~ 365) です。

**定義:** キューの経過日数は、その最も古いメッセージの経過日数で決まります。

このパラメーターは、CQSQUERY FUNC=QTYPE 要求でどのキューを処理するかを判別のためのフィルターとして使用されます。CQSQUERY 要求は、指定された queueage より古いデータ・オブジェクトが入っているキューの情報を戻します。queueage に 0 を指定するか、QAGE パラメーターを省略すると、CQSQUERY 要求はそのキュー・タイプのすべてのキューを処理します。

**重要:** QAGE を指定すると、キュー内のすべてのデータ・オブジェクトが読み取られるために、さらにパフォーマンス・オーバーヘッドがかかる原因になります。

**QDATASZ=querydatasizeaddress**

クライアントに戻される情報のサイズが入る 4 バイト・フィールドを指定する出力パラメーター。部分的なデータがバッファーに入れて戻される場合、このフィールドには、その情報を保持するために必要な実際のバッファー・サイズが含まれます。

**QTYPE=queuetypeaddress**

キュー・タイプが入った 4 バイト・フィールドのアドレスを指定する入力パラメーター。キュー・タイプの有効な値は、1 ~ 255 (10 進) です。

**QTYPENM=COLDQ**

CQSQUERY 要求が COLDQ に関する情報のためであることを示す入力パラメーター。

このパラメーターを使用すると、クライアントが、QTYPE=queuetypeaddress を指定した CQSQUERY FUNC=QTYPE 要求を使用してクライアント・キューに関して得られるのと同じタイプの情報を、コールド・キューに関して得ることができます。

**RETCODE=returncodeaddress**

CQSQUERY の戻りコードが入る 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

**RSNCODE=reasoncodeaddress**

CQSQUERY の理由コードが入る 4 バイト・フィールドのアドレスを指定する出力パラメーター。

**STATSZAR=statisticssizeaddress**

出力バッファーに入れて戻される CQSQUERY FUNC=STATISTICS 要求の単一の統計項目の長さが入る、4 バイト・フィールドのアドレスを指定する出力パラメーター。

部分的なデータが戻される場合、必要なバッファーのサイズは、このフィールドに入れて戻される値に、指定されたリスト項目の数を乗算することによって求めることができます。

**CQSQUERY の戻りコードと理由コード**

以下の表は、CQSQUERY 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSQUERY FUNC=DSECT 要求を使用します。

表 17. CQSQUERY の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000004'	X'00000120'	バッファー・サイズ (buffersize) が照会データ・サイズ (querydatasize) より小さい。部分的なデータが戻される。querydatasize は、すべてのデータを入れるために必要な実際のバッファー・サイズを指す。

表 17. CQSQUERY の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'00000004'	X'00000124'	<i>buffer</i> size が小さすぎるために、 <i>list</i> に指定された項目の数のデータを入れることができない。
X'00000004'	X'00000128'	そのキュー・タイプにデータ・オブジェクトがない。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000214'	<i>connecttoken</i> が無効。
X'00000008'	X'00000218'	FUNC が無効。
X'00000008'	X'00000224'	<i>bufferaddress</i> が無効。
X'00000008'	X'00000228'	<i>buffer</i> size が無効。
X'00000008'	X'0000022C'	<i>statisticssize</i> または <i>querydatasize</i> が無効。
X'00000008'	X'0000023C'	<i>queueage</i> が無効。
X'00000008'	X'00000240'	<i>queuetype</i> が無効。
X'00000008'	X'00000250'	<i>count</i> が無効。
X'00000008'	X'00000254'	<i>listaddress</i> が無効。
X'00000008'	X'0000027C'	CQSQUERY FUNC=QNAME、CQSQUERY FUNC=QTYPE、または CQSQUERY FUNC=QOBSJ はリソース構造に対して許可されていない。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000008'	X'00000288'	リスト・バージョンが無効。
X'0000000C'	X'00000300'	要求は少なくとも 1 つのリスト項目については成功したが、すべてではない。個々のエラーについては、 <i>compcode</i> を参照してください。
X'0000000C'	X'00000304'	要求はすべてのリスト項目について失敗した。個々のエラーについては、 <i>compcode</i> を参照してください。
X'00000010'	X'00000400'	この構造には CQSRSYNC が必須。
X'00000010'	X'00000404'	構造がアクセス不能。後で要求を再試行してください。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。
X'00000014'	X'00000500'	CQS 内部エラー。

## CQSREAD 要求

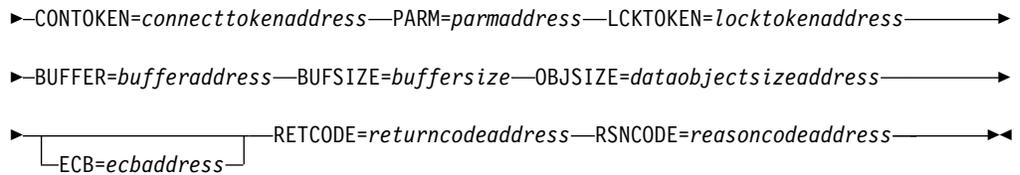
CQSREAD 要求は、特定のキューからクライアント・データ・オブジェクトのコピーを検索します。

### CQSREAD の形式

#### CQSREAD の CONTINUE 機能

前の CQSREAD 要求で部分的なデータが返された後で、残りのデータ・オブジェクトを取得するには、CQSREAD 要求の CONTINUE 機能を使用します。

▶—CQSREAD—FUNC=CONTINUE—CQSTOKEN=*cqstokenaddress*—▶



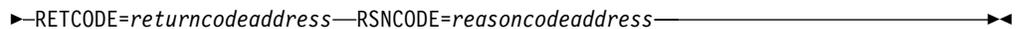
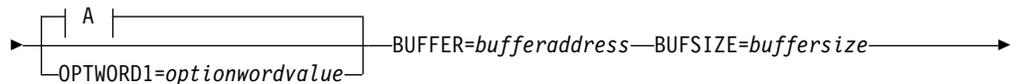
### CQSREAD の DSECT 機能

CQSREAD パラメーター・リストの長さ、CQSREAD の戻りコードと理由コード、および OPTWORD1 パラメーターの作成に使用できるリテラルのために、プログラムの中に等価 (EQU) ステートメントを組み込むには、CQSREAD 要求の DSECT 機能を使用します。

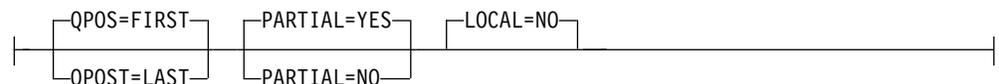


### LOCAL=NO を指定した CQSREAD の READ 機能

特定のキューからクライアント・データ・オブジェクトのコピーを検索してそれをロックするには、LOCAL=NO パラメーターを指定した CQSREAD 要求を使用します。



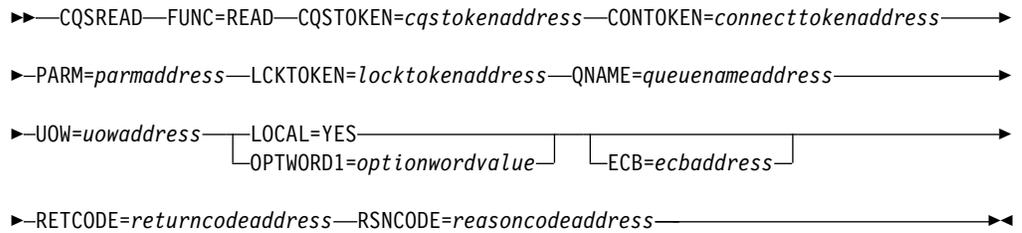
**A:**



### LOCAL=YES を指定した CQSREAD の READ 機能

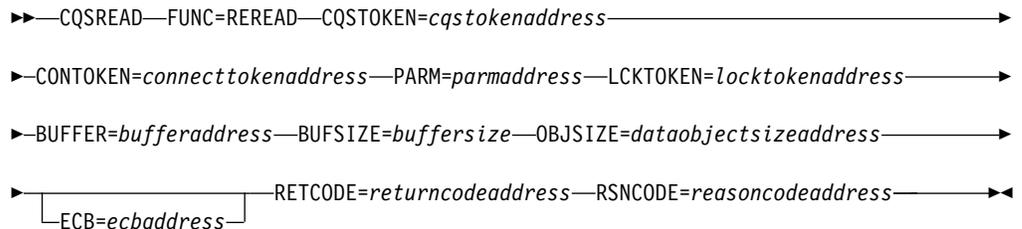
以前に CQSPUT LOCAL=YES 要求によって共用キューに保管したデータ・オブジェクトのロック・トークンを検索するには、LOCAL=YES パラメーターを指定した

CQSREAD 要求を使用します。この要求を使用することによって、クライアント障害、構造再作成、または CQS 再始動の場合にも、データ・オブジェクトをロックしたままにすることができます。



### CQSREAD の REREAD 機能

以前の CQSREAD FUNC=READ 要求で読み取られてロックされた、ロック・データ・オブジェクトを再読み取りするには、CQSREAD 要求の REREAD 機能を使用します。



### CQSREAD の使用法

CQSREAD 要求は、特定のキューからクライアント・データ・オブジェクトのコピーを検索します。データ・オブジェクトはキューから削除されませんが、CQSREAD FUNC=READ 要求の場合は、ロックされるので、後の CQS 要求でデータ・オブジェクトがアクセスされるのを防止します（適切なロック・トークンを使用した場合を除く）。データ・オブジェクトは、キューの先頭または最後から検索できます。データ・オブジェクトは、CQSREAD 要求で提供されたクライアント・バッファーに戻されます。

制約事項: CQSREAD 要求は、リソース構造に対してサポートされていません。

ロック・トークンがクライアントへ戻され、これがデータ・オブジェクトを識別します。ロックされたデータ・オブジェクトに作用するすべての要求（例: CQSDEL、CQSMOVE、CQSREAD、または CQSUNLCK）の場合に、このトークンを CQS に渡す必要があります。

検索されたデータ・オブジェクトのサイズがクライアント・バッファーのサイズより大きい場合、PARTIAL=YES が指定されているなら、クライアント・バッファーに入るだけの量のデータがクライアントに戻されます。戻りコードまたは理由コードも戻され、部分的なデータ・オブジェクトが実際のデータ・オブジェクト・サイズのとおりに戻されたことを示します。

検索されたデータ・オブジェクトのサイズがクライアント・バッファのサイズより大きい場合、PARTIAL=NO が指定されているなら、データ・オブジェクトは戻されません。戻りコードと理由コードが戻され、クライアント・バッファ・サイズが小さすぎるためにデータ・オブジェクトが戻されないことを示します。実際のデータ・オブジェクト・サイズもクライアントに戻されます。

検索されたデータ・オブジェクトのサイズがクライアント・バッファと同じサイズかそれよりも小さい場合、データ・オブジェクト全体がバッファに移動され、バッファの残りの部分に変更されません。データ・オブジェクト・サイズもクライアントに戻されます。

前の CQSREAD 要求で部分的なデータが戻された場合、CQSREAD FUNC=CONTINUE 要求は残りのデータ・オブジェクトを検索します。

**重要:** この要求は、現在位置が失われることがあるため、CQS 再始動後にエラーになる可能性があります。

CQSREAD FUNC=REREAD 要求は、(前の CQSREAD FUNC=READ 要求で) 以前読み取ってロックしたロック・データ・オブジェクトを再読み取りします。データ・オブジェクトはロックされたままです。

関連資料: CQS クライアントに対して CQSREAD 要求を使用する方法については、以下の例を参照してください。

パラメーター説明

**BUFFER=bufferaddress**

キューから検索されたデータ・オブジェクトを入れておくクライアント・バッファのアドレスを指定する、4 バイトの入力パラメーター。

**BUFSIZE=buffer size**

クライアント・バッファのサイズを指定する 4 バイトの入力パラメーター。

**CONTOKEN=connecttokenaddress**

この CQS が管理する特定のカップリング・ファシリティー構造に対するクライアントの接続を一意的に識別する、16 バイトの接続トークンのアドレスを指定する入力パラメーター。接続トークンは、CQSCONN 要求によって戻されません。

**CQSTOKEN=cqstokenaddress**

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

**ECB=ecbaddress**

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

**LCKTOKEN=locktokenaddress**

CQSREAD 要求によってロックされたデータ・オブジェクトの 16 バイトのロック・トークンのアドレスを指定する入出力パラメーター。

CQSREAD FUNC=READ 要求の場合、ロック・トークンは入力時にはゼロになっています。これはクライアントに戻されるロック・トークンを入れる出力域としても使用されます。CQSREAD FUNC=REREAD または FUNC=CONTINUE 要求の場合、このフィールドは前の CQSREAD 要求で戻されたロック・トークンを入れる入力域です。

#### **LOCAL=NO | YES**

クライアント・アドレス・スペースのデータ・オブジェクトのローカル・コピーをクライアントが処理するべきかどうかを指示する入力パラメーター。

**NO** CQS が、指定されたクライアント・キューからデータ・オブジェクトを戻して、ロックするようにクライアントが希望していることを示す。この場合、CQS はカップリング・ファシリティーにアクセスしてデータ・オブジェクトを検索します。

#### **YES**

クライアントがそのローカル・バッファーからのデータ・オブジェクトのローカル・コピーを処理していることを示す。この要求は、クライアントが共用キュー上のデータ・オブジェクトのコピーへのアクセスに使用できる、データ・オブジェクトのロック・トークンを戻します。このデータ・オブジェクトは、CQSPUT LOCAL=YES 要求によって共用キューに置かれたものです。

データ・オブジェクトのローカル・コピーを使用すれば、クライアントは、共用キューを使用した場合のパフォーマンスのオーバーヘッドを削減することができます。データ・オブジェクトは、共用キューにある限りは、クライアントに障害が発生した場合にもリカバリーできます。データ・オブジェクトは、ロックされたままである間は、他のクライアントには利用不能です。

CQSREAD 要求では、クライアントにローカル・コピーがあるので、データ・オブジェクトがクライアントに戻されることはありません。クライアントが CQSREAD LOCAL=YES 要求を出さない場合、クライアントと CQS の間の接続が失われると、CQS はデータ・オブジェクトをアンロックしてどのクライアントも使用できるようにします。

制約事項: LOCAL=YES を指定する場合は、TIMESTAMP パラメーターを使用できません。

LOCAL パラメーターは、OPTWORD1 パラメーターが指定されているときは使用できません。LOCAL の代わりに OPTWORD1 を指定する場合、次の等価 (EQU) ステートメントを使用して OPTWORD1 パラメーターの値を生成することができます。

```
CQSREAD_LCLYEQUX LOCAL=YES  
CQSREAD_LCLNEQUX LOCAL=NO
```

#### **OBJSIZE=dataobjectsizaddress**

データ・オブジェクトのサイズを入れる 4 バイト・フィールドのアドレスを受け取るための出力パラメーター。データ・オブジェクト・サイズがクライアント・バッファー・サイズより大きい場合、このフィールドには実際のデータ・オブジェクト・サイズが入ります。部分的なデータが戻される場合、戻されるデータ・オブジェクトのサイズは指定されたクライアント・バッファーのサイズになります。

**OPTWORD1=optionwordvalue**

この要求に対するリテラルを指定する 4 バイトの入力パラメーター。このパラメーターは LOCAL、PARTIAL、および QPOS の代わりとして使用できます。このリテラル値の等価 (EQU) ステートメントは、LOCAL、PARTIAL、および QPOS パラメーターの説明のところにリストされています。EQU ステートメントは、DSECT 機能を使用しても生成できます。OPTWORD1 パラメーターは、LOCAL、PARTIAL、または QPOS が指定されている場合は使用できません。

要件: OPTWORD1 パラメーターをコーディングする場合、このマクロがサポートする各リテラル値ごとに 1 つの等価値で構成される値を渡す必要があります。

**PARM=parmaddress**

要求が、CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSREAD\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

**PARTIAL=YES | NO**

部分的なデータを検索するかどうか、およびデータ・オブジェクト・サイズがクライアント・バッファ・サイズより大きい場合にデータ・オブジェクトをロックするかどうかを指定する入力パラメーター。

**YES**

データ・オブジェクト・サイズがクライアント・バッファ・サイズより大きい場合、データ・オブジェクトをロックして、部分的なデータがクライアント・バッファに戻されます。データ・オブジェクトの実際のサイズは *dataobjectsize* に入れて戻されます。

**NO** データ・オブジェクト・サイズがクライアント・バッファ・サイズより大きい場合、データ・オブジェクトにはロックも検索も行われません。データ・オブジェクトの実際のサイズは *dataobjectsize* に入れて戻されます。

PARTIAL パラメーターは、OPTWORD1 パラメーターが指定されているときは使用できません。PARTIAL の代わりに OPTWORD1 を指定する場合、次の等価 (EQU) ステートメントを使用して OPTWORD1 パラメーターの値を生成することができます。

```
CQSREAD_PRTLNEQUX PARTIAL=NO  
CQSREAD_PRTLNEQUX PARTIAL=YES
```

**QNAME=queuenameaddress**

データ・オブジェクトの検索を行うキューの 16 バイトの名前のアドレスを指定する入力パラメーター。キュー名の最初のバイトは、キュー・タイプを識別します。

**QPOS=FIRST | LAST**

データ・オブジェクトの検索を行うキュー上の位置を指定する入力パラメーター。

**FIRST**

データ・オブジェクトはキューの先頭から検索される。

## LAST

データ・オブジェクトはキューの終わりから検索される。

QPOS パラメーターは、OPTWORD1 パラメーターが指定されているときは使用できません。QPOS の代わりに OPTWORD1 を指定する場合、次の等価 (EQU) ステートメントを使用して OPTWORD1 パラメーターの値を生成することができます。

```
CQSREAD_QPOSLEQUX  QPOS=LAST
CQSREAD_QPOSFEQUX  QPOS=FIRST
```

## RETCODE=returncodeaddress

CQSREAD の戻りコードを含む 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

## RSNCODE=reasoncodeaddress

CQSREAD の理由コードを含む 4 バイト・フィールドのアドレスを指定する出力パラメーター。

## TIMESTAMP=timestampaddress

データ・オブジェクトが各キューに配置されたときのタイム・スタンプが含まれる 8 バイト・フィールドのアドレスを指示する 4 バイト出力パラメーター。

重要: LOCAL=YES が指定されている場合は、CQS は、構造からデータ・オブジェクトを読み取らず、タイム・スタンプは取得できません。

## UOW=uowaddress

キューから検索されたデータ・オブジェクトの作業単位 (UOW) を保持する 32 バイト・エリアのアドレスを指定する出力パラメーター。UOW は、CQSPUT 要求を使用してキューにデータ・オブジェクトを書き込んだクライアントによって生成されました。

## CQSREAD の戻りコードと理由コード

以下の表は、CQSREAD 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSREAD FUNC=DSECT 要求を使用します。

表 18. CQSREAD の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000004'	X'00000120'	バッファー・サイズ (buffersize) がデータ・オブジェクト・サイズ (dataobjectsize) より小さい。部分的なデータが戻される。dataobjectsize には実際のデータ・オブジェクト・サイズが含まれる。
X'00000004'	X'00000124'	バッファー・サイズ (buffersize) がデータ・オブジェクト・サイズ (dataobjectsize) より小さい。PARTIAL=NO が指定されていたため、データは戻されない。dataobjectsize には実際のデータ・オブジェクト・サイズが含まれる。

表 18. CQSREAD の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'00000004'	X'00000128'	指定されたキュー名に検索すべきデータ・オブジェクトがない。
X'00000004'	X'0000012C'	戻すべき部分的なデータがない。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000214'	<i>connecttoken</i> が無効。
X'00000008'	X'00000218'	FUNC が無効。
X'00000008'	X'0000021C'	<i>locktoken</i> が無効。
X'00000008'	X'00000220'	<i>queuename</i> が無効。
X'00000008'	X'00000224'	<i>bufferaddress</i> が無効。
X'00000008'	X'00000228'	<i>buffersize</i> が無効。
X'00000008'	X'0000022C'	<i>dataobjectsize</i> が無効。
X'00000008'	X'00000230'	<i>uow</i> が無効。
X'00000008'	X'00000234'	ロック・トークン・アドレスが無効。
X'00000008'	X'00000278'	要求では LOCAL=YES と指定しているが、要求されたオブジェクトは LOCAL=NO を使用して構造に置かれていた。
X'00000008'	X'0000027C'	CQSREAD は、リソース構造に対して許可されていない。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000010'	X'00000400'	この構造には CQSRSYNC が必須。
X'00000010'	X'00000404'	構造がアクセス不能。後で要求を再試行してください。
X'00000010'	X'00000408'	現在位置が失われ、CQSREAD FUNC=CONTINUE 要求を処理できない。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。
X'00000010'	X'00000440'	再作成のためにオブジェクトが失われた。
X'00000014'	X'00000500'	CQS 内部エラー。

## CQSRECVR 要求

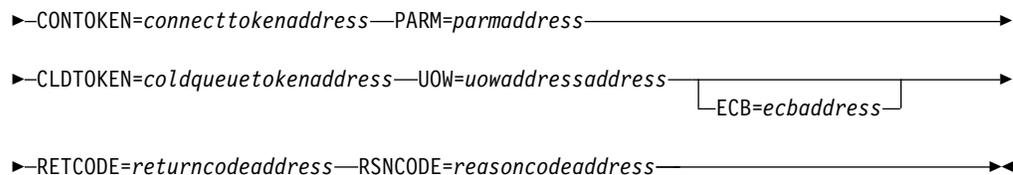
CQSRECVR 要求を使用すると、クライアントは、CQS またはクライアントがコールド・スタートされたために CQS コールド・キュー (CQS 専用キュー) に移動されたロック・データ・オブジェクトをリカバリーすることができます。

### CQSRECVR の形式

#### CQSRECVR の DELETE 機能

ある UOW に関連した 1 つのデータ・オブジェクトをコールド・キューから削除するには、CQSRECVR 要求の DELETE 機能を使用します。

▶—CQSRECVR—FUNC=DELETE—CQSTOKEN=*cqstokenaddress*—▶



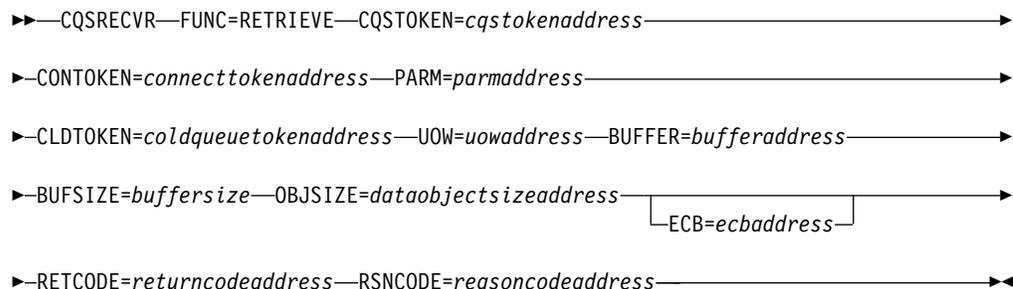
### CQSRECVR の DSECT 機能

CQSRECVR パラメーター・リストの長さ、CQSRECVR の戻りコードと理由コード、および OPTWORD1 パラメーターの作成に使用できるリテラルのために、プログラムの中に等価 (EQU) ステートメントを組み込むには、CQSRECVR 要求の DSECT 機能を使用します。



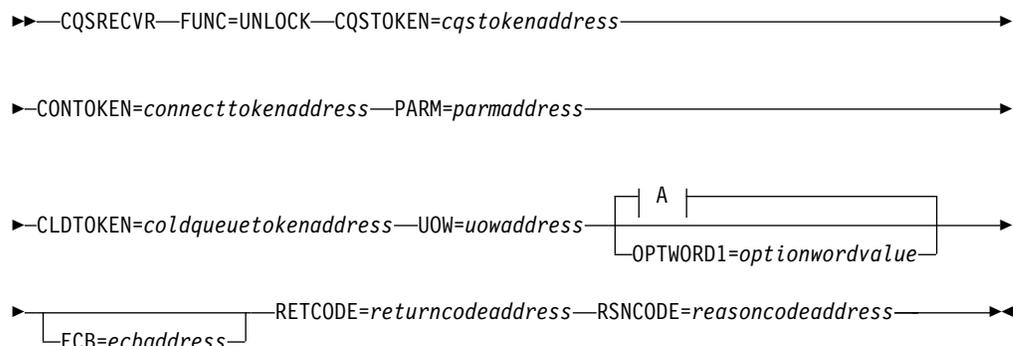
### CQSRECVR の RETRIEVE 機能

ある UOW に関連したデータ・オブジェクトのコピーをコールド・キューから検索するには、CQSRECVR 要求の RETRIEVE 機能を使用します。



### CQSRECVR の UNLOCK 機能

コールド・キュー上のある UOW に関連したデータ・オブジェクトをアンロックするには、CQSRECVR 要求の UNLOCK 機能を使用します。



A:



## CQSRECVR の使用法

制約事項: CQSRECVR 要求は、リソース構造に対してサポートされていません。

ある UOW に関連した 1 つのデータ・オブジェクトをコールド・キューから削除するには、CQSRECVR FUNC=DELETE 要求を使用します。データ・オブジェクトが 1 つだけ削除されます。

CQSRECVR FUNC=RETRIEVE 要求は、ある UOW に関連したデータ・オブジェクトのコピーをコールド・キューから検索します。そのデータ・オブジェクトはコールド・キュー上にそのまま残されるので、他の CQSRECVR 要求で使用可能です。データ・オブジェクトは、CQSRECVR FUNC=RETRIEVE 要求に指定されたクライアント・バッファーに入れて戻されます。

データ・オブジェクトが提供されたクライアント・バッファーのサイズと等しいかそれよりも小さい場合、データ・オブジェクトはバッファーに入れて戻され、バッファーの残りの部分に変更されません。データ・オブジェクトのサイズがクライアントに戻されます。

データ・オブジェクトのサイズがクライアント・バッファーより大きい場合、データ・オブジェクトは戻されません。データ・オブジェクトのサイズがクライアントに戻されます。

コールド・キュー上のある UOW に関連した 1 つのデータ・オブジェクトをアンロックするには、CQSRECVR FUNC=UNLOCK 要求を使用します。そのデータ・オブジェクトはコールド・キューから元のクライアント・キューに移動されて、他の CQS 要求で使用することができます。データ・オブジェクトを移動すべき位置は、クライアントが指定することができます。

パラメーター説明

### **BUFFER=bufferaddress**

キューから検索されたデータ・オブジェクトを入れておくクライアント・バッファーのアドレスを指定する、4 バイトの入力パラメーター。

### **BUFSIZE=buffer size**

クライアント・バッファーのサイズを指定する 4 バイトの入力パラメーター。

### **CLDTOKEN=coldqueuetokenaddress**

UOW と組み合わせて、CQS コールド・キュー (COLDQ) からリカバリーすべきデータ・オブジェクトを識別する、16 バイトのコールド・キュー・トークンのアドレスを指定する入力パラメーター。

コールド・キュー・トークンは、CQS 構造イベント出口ルーチン内の Resync 項目の SEVX\_RETOKEN フィールドに入れてクライアントに渡されます。この出口ルーチンは、UOW 状況が COLD のときに、CQS が開始した再同期のために呼び出されます。

**CONTOKEN=connecttokenaddress**

この CQS が管理する特定のカップリング・ファシリティ構造に対するクライアントの接続を一意的に識別する、16 バイトの接続トークンのアドレスを指定する入力パラメーター。接続トークンは、CQSCONN 要求によって戻されません。

**CQSTOKEN=cqstokenaddress**

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

**ECB=ecbaddress**

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

**OBJSIZE=dataobjectsizedataaddress**

データ・オブジェクトのサイズを入れるための 4 バイトのエリアのアドレスを指定する出力パラメーター。データ・オブジェクト・サイズがクライアント・バッファー・サイズより大きい場合、このフィールドには実際のデータ・オブジェクト・サイズが入ります。部分的なデータが戻される場合、戻されるデータ・オブジェクトのサイズは指定されたクライアント・バッファーのサイズになります。

**OPTWORD1=optionwordvalue**

この要求に対するリテラルを指定する 4 バイトの入力パラメーター。このパラメーターは、QPOS の代わりとして使用できます。このリテラル値の等価 (EQU) ステートメントは、QPOS パラメーターの説明のところにリストされています。EQU ステートメントは、DSECT 機能を使用しても生成できます。OPTWORD1 パラメーターは QPOS が指定されている場合は使用できません。

要件: OPTWORD1 パラメーターをコーディングする場合、このマクロがサポートする各リテラル値ごとに 1 つの等価値で構成される値を渡す必要があります。

**PARM=parmaddress**

要求が、CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSRECVR\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

**QPOS=SYSTEM | FIRST | LAST**

アンロックされたデータ・オブジェクトを追加するキュー上の位置を指定する入力パラメーター。デフォルトは SYSTEM です。

**FIRST**

データ・オブジェクトがアンロックされ、キューの先頭に追加されることを指示する。

## LAST

データ・オブジェクトがアンロックされ、キューの最後に追加されることを指示する。

## SYSTEM

データ・オブジェクトがアンロックされ、元の位置に応じてキューの先頭または最後に追加されることを指示する。このデータ・オブジェクトをロックした CQSREAD 要求がそのデータ・オブジェクトをキューの先頭から取得した場合、そのデータ・オブジェクトはアンロックされてキューの先頭に追加されます。CQSREAD 要求がそのデータ・オブジェクトをキューの最後から取得した場合、そのデータ・オブジェクトはアンロックされてキューの最後に追加されます。

QPOS パラメーターは、OPTWORD1 パラメーターが指定されているときは使用できません。QPOS の代わりに OPTWORD1 を指定する場合、次の等価 (EQU) ステートメントを使用して OPTWORD1 パラメーターの値を生成することができます。

```
CQSRECVR_QPOSSEQX  QPOS=SYSTEM
CQSRECVR_QPOSFEQUX QPOS=FIRST
CQSRECVR_QPOSLEQUX QPOS=LAST
```

## RETCODE=returncodeaddress

CQSRECVR の戻りコードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

## RSNCODE=reasoncodeaddress

CQSRECVR の理由コードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

## UOW=uowaddress

データ・オブジェクトの作業単位 (UOW) を保持する 32 バイト・エリアのアドレスを指定する入力パラメーター。UOW は、*coldqueuetoken* と共に、コールド・キューからリカバリーするデータ・オブジェクトを識別します。

UOW は、CQS 構造イベント出口ルーチン内の Resync 項目の SEVX\_REUOW フィールドに入れてクライアントに渡されます。この出口ルーチンは、UOW 状況が COLD のときに、CQS が開始した再同期のために呼び出されます。

## CQSRECVR の戻りコードと理由コード

以下の表は、CQSRECVR 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSRECVR FUNC=DSECT 要求を使用します。

表 19. CQSRECVR の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000004'	X'00000124'	<i>bufferize</i> が小さすぎる。

表 19. CQSRECVR の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'00000004'	X'00000128'	UOW のデータ・オブジェクトがコールド・キュー上に見つからない。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000214'	<i>connecttoken</i> が無効。
X'00000008'	X'00000218'	FUNC が無効。
X'00000008'	X'00000224'	<i>bufferaddress</i> が無効。
X'00000008'	X'00000228'	<i>buffersize</i> が無効。
X'00000008'	X'0000022C'	<i>dataobjectsize</i> が無効。
X'00000008'	X'00000230'	<i>uow</i> が無効。
X'00000008'	X'00000234'	<i>coldqueuetoken</i> が無効。
X'00000008'	X'0000027C'	CQSRECVR は、リソース構造に対して許可されていない。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000010'	X'00000400'	この構造には CQSRSYNC が必須。
X'00000010'	X'00000404'	構造がアクセス不能。後で要求を再試行してください。
X'00000010'	X'00000414'	元のキューがいっぱいなために、データ・オブジェクトをアンロックできない。このキューにはこれ以上データ・オブジェクトを移動できない。他のキューについての CQSRECVR FUNC=UNLOCK 要求は許可される。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。
X'00000014'	X'00000500'	CQS 内部エラー。

## CQSREG 要求

CQSREG 要求はクライアントを CQS に登録します。

### CQSREG の形式

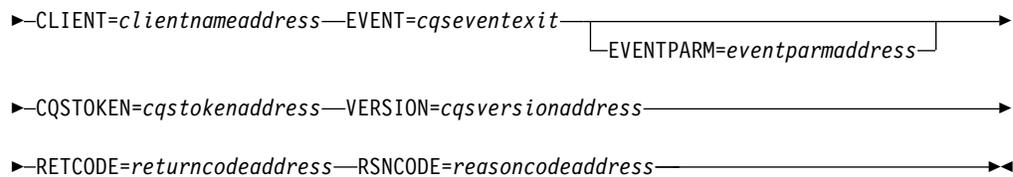
#### CQSREG の DSECT 機能

CQSREG パラメーター・リストの長さ、および CQSREG の戻りコードと理由コードのために、プログラムの中に等価 (EQU) ステートメントを組み込むには、CQSREG 要求の DSECT 機能を使用します。

▶—CQSREG—FUNC=DSECT—▶

クライアントを CQS に登録するには、CQSREG 要求の REGISTER 機能を使用します。

▶—CQSREG—FUNC=REGISTER—PARM=parmaddress—CQSSSN=cqssystemnameaddress—▶



## CQSREG の使用法

CQSREG 要求はクライアントを CQS に登録します。登録が正常に完了すると、CQS トークンが戻されます。このトークンは、クライアントが CQS に登録されたことを表しており、それ以後すべての CQS 要求では、クライアント識別にはこれを使用する必要があります。

CQSREG FUNC=REGISTER 要求は、クライアントが最初に行う CQS 要求でなければなりません。また、CQS が異常終了して再始動した後は、クライアントが CQS 要求の発行を再開するには、その前に CQSREG FUNC=REGISTER 要求が必要です。

### **CLIENT=clientnameaddress**

CQS に登録されるクライアントの 8 バイトの名前のアドレスを指定する入力パラメーター。クライアント名は、同一 CQS に登録されているすべてのクライアント、および同一キューを共用しているすべての CQS で固有であることが必要です。

### **CQSTOKEN=cqstokenaddress**

クライアントの CQS に対する接続を一意的に識別する CQS 登録トークンを受け取る、16 バイト・エリアのアドレスを指定する出力パラメーター。登録トークンは、正常に行われた CQSREG 要求によって戻されます。

### **CQSSN=cqssystemnameaddress**

クライアントが接続したい CQS の 4 バイトのサブシステム名のアドレスを指定する入力パラメーター。このパラメーターは、クライアントが接続したい CQS の CQSIPxxx PROCLIB メンバーの SSN= パラメーターと一致していなければなりません。

### **EVENT=cqseventexit**

CQS イベント出口ルーチンのアドレスを指定する 4 バイトの入力パラメーター。

### **EVENTPARM=eventparmaddress**

CQS が出口が呼び出されるたびに CQS イベント出口ルーチンに渡すクライアント・データが入っている 4 バイト・エリアのアドレスを指定する入力パラメーター。

### **PARM=parmaddress**

要求が、CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSREG\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

**RETCODE=returncodeaddress**

CQSREG の戻りコードが入る 4 バイト・フィールドのアドレスを指定する出力パラメーター。CQSREG の戻りコードは、このフィールドとレジスター 15 の両方に戻されます。

**RSNCODE=reasoncodeaddress**

CQSREG の理由コードが入る 4 バイト・フィールドのアドレスを指定する出力パラメーター。CQSREG の理由コードは、このフィールドとレジスター 0 の両方に戻されます。

**VERSION=cqsversionaddress**

CQS のバージョン番号が入る 4 バイト・エリアのアドレスを指定する出力パラメーター。バージョン番号のフォーマットは、00vvrrmm です。

**00** このバイトは将来の利用のために予約済み。現在は常に 00。

**vv** バージョン番号。

**rr** リリース番号。

**mm** 修正レベルまたはサブリリース番号。

例えば、バージョン 1.1.0 は X'00010100' のように示されます。

**CQSREG の戻りコードと理由コード**

以下の表は、CQSREG 要求で返される戻りコードと理由コードの組み合わせの一覧です。

表 20. CQSREG の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000004'	X'00000100'	クライアントはすでに CQS に登録済み。
X'00000008'	X'00000244'	<i>clientname</i> が無効。
X'00000008'	X'00000284'	CQSREG パラメーター・リスト・バージョンが無効。このエラーは、CQS クライアントおよびクライアントが使用しようとしている CQS アドレス・スペース間のバージョンの差が原因と考えられます。
X'00000010'	X'0000040C'	CQS シャットダウンが保留中。
X'00000010'	X'00000430'	CQS アドレス・スペースがアクティブではない。 CQS アドレス・スペースは開始済みでなければならない。
X'00000010'	X'00000438'	別のアドレス・スペースがすでにこのクライアント ID (CQSREG 要求で渡された) を使用して CQS に登録されている。
X'00000010'	X'00000440'	クライアント・アドレス・スペースのユーザー ID は、この CQS に登録する許可を受けていない。
X'00000010'	X'00000448'	登録済みクライアント・アドレス・スペースが、CQS に対して 2 回目の登録を試みた。
X'00000014'	X'00000500'	CQS 内部エラー。
X'00000014'	X'00000504'	クライアントのアドレス・スペースに CQS が使用するためのストレージを獲得できない。

表 20. CQSREG の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'00000014'	X'00000508'	ストレージを獲得できない (CCIB)。
X'00000014'	X'0000050C'	ストレージを獲得できない (CRET)。
X'00000014'	X'00000510'	CQS 内部エラー (ASCB ロケーション)。
X'00000014'	X'00000514'	クライアントを登録するために CQS をモニターする、z/OS リソース・マネージャー・ルーチンを確立できない。
X'00000014'	X'00000518'	CQS 内部エラー (ESTAE アドレス)。
X'00000014'	X'0000051C'	CQS 内部エラー (NmTkn 検索)。
X'00000014'	X'00000520'	CQS 内部エラー (CGCT エラー)。
X'00000014'	X'00000524'	CQS 内部エラー (TTKN エラー)。
X'00000014'	X'00000528'	CQS 内部エラー (ALESERV エラー)。
X'00000014'	X'0000052C'	CQS 内部エラー (BPESVC エラー)。
X'00000014'	X'00000530'	CQS のためにクライアントをモニターする z/OS リソース・マネージャー・ルーチンを確立できない。
X'00000014'	X'00000534'	CQSREG 処理中に異常終了が起きた。
X'00000014'	X'0000053C'	CQS 登録モジュール CQSREG00 をロードできない。

## CQSRSYNC 要求

クライアントは、CQSRSYNC 要求を使用して、1 つの構造に対する未確定データと CQS を再同期することができます。この要求は、クライアントが CQSCONN 要求の後に最初に出す要求でなければなりません。

### CQSRSYNC の形式

#### CQSRSYNC の DSECT 機能

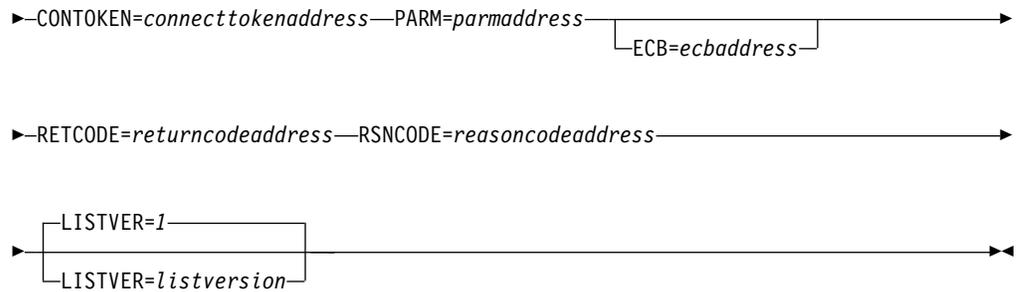
CQSRSYNC パラメーター・リストの長さ、および CQSRSYNC の戻りコードと理由コードのために、プログラムの中に等価 (EQU) ステートメントを組み込むには、CQSRSYNC 要求の DSECT 機能を使用します。

▶▶—CQSRSYNC—FUNC=DSECT—▶▶

#### CQSRSYNC の RSYNCCOLD 機能

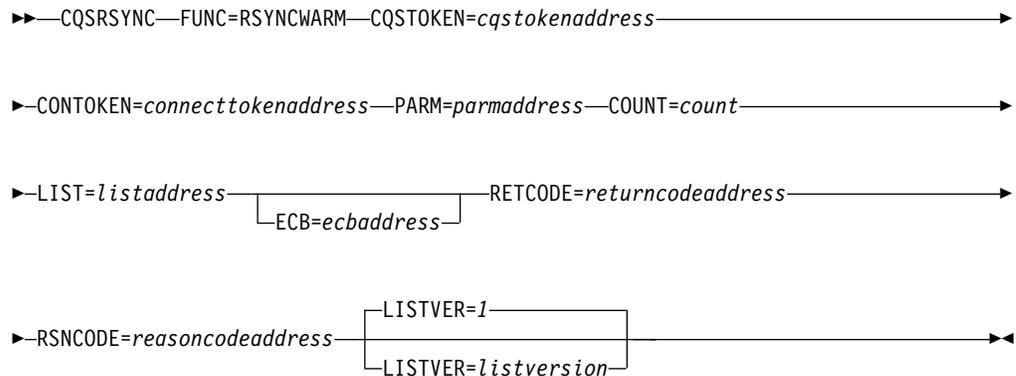
クライアントがコールド・スタートを実行しているときに未解決の UOW に関する情報がない場合には、CQSRSYNC 要求の RSYNCCOLD 機能を使用します。

▶▶—CQSRSYNC—FUNC=RSYNCCOLD—CQSTOKEN=cqstokenaddress—▶▶



### CQSRSYNC の RSYNCWARM 機能

クライアントがウォーム・リスタートまたは緊急リスタートを実行しているときに、CQS が解決する必要のある未解決の UOW に関する情報がある場合は、CQSRSYNC 要求の RSYNCWARM 機能を使用します。



### CQSRSYNC の使用法

クライアントは、CQSRSYNC 要求を使用して、1 つの構造に対する未確定データと CQS を再同期することができます。この要求は、クライアントが CQSCONN 要求の後に最初に出す要求でなければなりません。

制約事項: CQSRSYNC 要求は、リソース構造に対してサポートされていません。

CQSRSYNC 要求は、クライアントが解決すべき未確定作業単位 (UOW) を持っていない場合でも、例えばクライアントが正常終了の後にコールド・スタートやウォーム・スタートを行うときには必須です。この要求が必須なのは、CQS が接続に関する情報を持っていて、かつ未解決の処理すべき UOW を持っている場合があるためです。

未解決の UOW がある場合、CQS はクライアントの構造イベント出口ルーチンを再同期の一部として呼び出します。CQS は、このルーチンを呼び出して、CQS が認識している UOW と、クライアントが CQSRSYNC 要求で渡さなかった UOW について、クライアントに通知します。このプロセスは、CQS 開始の再同期と呼ばれます。

出口ルーチンは、CQS に未解決の UOW がある場合に限り、クライアントのコールド・スタートまたはリスタート時に呼び出されます。CQS 開始の再同期の場合、

構造イベント出口ルーチンは複数回呼び出される可能性があります。クライアントには、出口ルーチンに渡される各 UOW のために、CQS の戻り状況に基づいて UOW を解決する正しいアクションを行えるようにする責任があります。

CQS がコールド・スタートした場合、CQS にはクライアント UOW についての認識がありません。このような場合、再同期リストは処理されません。CQS は、CQS が終了したときに完了していなかった CQSREAD 要求を探します。未完了の作業がある場合、データ・オブジェクトはコールド・キューに移動され、未解決の UOW のクライアントにデータ・オブジェクトについて知らせるために、構造イベント出口ルーチンが呼び出されます。

CQSRSYNC 要求が完了した後で、いくつかの UOW が据え置き再同期状況になっている場合があります。この状況は、CQS がまだ UOW の再同期中であることを示しています。CQS が再同期を完了すると、UOW の状態を示すために構造イベント出口ルーチンが呼び出されます。据え置き再同期は、CQSRSYNC 要求の間に CQS が再同期できない UOW に適用されるだけで、クライアント・コールド・スタートでは起こりません。出口ルーチンはそれぞれの据え置き UOW に 1 回呼び出されます。したがって、据え置き再同期の場合は出口ルーチンが複数回呼び出される可能性があります。

#### パラメーター説明

##### **CONTOKEN=connecttokenaddress**

この CQS が管理する特定のカップリング・ファシリティ構造に対するクライアントの接続を一意的に識別する、16 バイトの接続トークンのアドレスを指定する入力パラメーター。接続トークンは、CQSCONN 要求によって戻されません。

##### **COUNT=count**

resync リスト内の項目の数を指定する、4 バイトの入力パラメーター。

##### **CQSTOKEN=cqstokenaddress**

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

##### **ECB=ecbaddress**

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

##### **LIST=listaddress**

再同期リストのアドレスを指定する 4 バイトの入力パラメーター。各項目には、クライアントによる解決が必要な未確定 UOW が入っています。各項目の一部のフィールドは、CQSRSYNC 要求の前にクライアントが初期化しておく必要があります。他のフィールドは、CQSRSYNC 要求の完了時に CQS によって戻されます。

CQSRSYNL リスト項目の DSECT は、リスト項目をマップし、クライアントが使用することができます。複数のリスト項目は、連続するストレージに常駐させる必要があります。

各リスト項目の内容は次の通りです。

### clientdata

クライアント・データ・フィールドを指定する 4 バイトの入力パラメーター。このパラメーターはオプションです。CQS は、この項目に保管されたデータを使用しません。

**uow** キューの作業単位 ID を指定する 32 バイトの入力パラメーター。このパラメーターは必須で、CQSRSYNC 要求の前にクライアントが初期化しておく必要があります。

### clientstatus

UOW の状況が入った 2 バイトの入力パラメーター。この状況は、この UOW にクライアントが最後に行った処置を表します。このパラメーターは必須で、CQSRSYNC 要求の前にクライアントが初期化しておく必要があります。

以下の表は、状況の値の一覧です。

表 21. クライアントからの UOW 状況

状況	意味
X'0010'	書き込み完了  一連の CQSPUT 要求のうちの最後の (または唯一の) CQSPUT 要求が、UOW に対して出されています。UOW のすべてのデータ・オブジェクトはカップリング・ファシリティにあるものと想定されます。
X'0020'	読み取り  UOW のデータ・オブジェクトはカップリング・ファシリティでロックされているものと想定されます。
X'0030'	アンロック  ロック・トークン付き CQSUNLCK 要求が UOW に対して出されました。データ・オブジェクトはアンロックされ、カップリング・ファシリティ上の作業キューで使用可能にされたものと想定されます。
X'0040'	移動  ロック・トークン付き CQSMOVE 要求が UOW に対して出されました。データ・オブジェクトはカップリング・ファシリティ上の新しいキューに移動されたものと想定されます。
X'0050'	削除  ロック・トークン付き CQSDEL 要求が UOW に対して出されました。データ・オブジェクトはカップリング・ファシリティから削除されたものと想定されます。

### cqsstate

結果としての UOW の状態を CQS から受け取るための 2 バイトの出力パラメーター。このパラメーターは、CQS によって CQSRSYNC 要求の結果として戻されます。

以下の表は、状況の値の一覧です。

表 22. CQS からの UOW 状況

状況	意味
X'0010'	書き込み同期  クライアント状況は「書き込み完了」です。CQS 状況は「書き込み完了」です。CQS は UOW を認識し、その UOW に関するすべてのデータ・オブジェクトはカップリング・ファシリティーにあります。PUT トークンが UOW のために戻されます。クライアントは、PUT トークンを使用して CQSPUT FUNC=FORGET 要求を出す必要があります。
X'0012'	再同期据え置き  クライアント状況は「書き込み完了」です。CQS 状況は「未確定」です。この状況はリカバリー可能 UOW のためだけに戻されます。CQS はその UOW について認識していますが、まだその状況の処理中です。クライアントは、その構造イベント出口ルーチンが CQS によって呼び出されるまで待つ必要があります。CQS は、クライアントの構造イベント出口ルーチンに通知して、UOW および UOW の状況を渡します。状況が「書き込み同期」であれば、UOW の PUT トークンも戻されます。クライアントは、PUT トークンを使用して CQSPUT FUNC=FORGET 要求を出す必要があります。  状況が「書き込み障害」である場合、クライアントは CQSPUT FUNC=PUT 要求を再度出す必要があります。状況が「不明」である場合、データ・オブジェクトはカップリング・ファシリティー上にある可能性もあり、ない可能性もあります。
X'0020'	読み取り同期  クライアント状況は「読み取り」です。CQS 状況は「読み取り完了」です。CQS はロックすべき UOW のデータ・オブジェクトを検出しました。ロック・トークンが UOW のために戻されます。クライアントは、後続の CQS 要求で、この UOW を持つデータ・オブジェクトについてこのロック・トークンを使用する必要があります。
X'0030'	アンロック同期  クライアント状況は「読み取りアンロック」です。CQS 状況は「アンロック同期」です。CQS はロックすべき UOW のデータ・オブジェクトを検出し、それをアンロックしました。クライアントによるこれ以上の処置は必要ありません。
X'0050'	削除同期  クライアント状況は「削除」です。CQS 状況は「削除同期」です。CQS はロックすべき UOW のデータ・オブジェクトを検出し、それを削除しました。クライアントによるこれ以上の処置は必要ありません。

表 22. CQS からの UOW 状況 (続き)

状況	意味
X'00F1'	<p>ロック</p> <p>以下の条件の 1 つが存在します。</p> <ul style="list-style-type: none"> <li>クライアント状況は「削除」です。CQS 状況は「ロック」です。CQS はロックすべき UOW を見つけましたが、構造からデータ・オブジェクトを削除できませんでした。データ・オブジェクトはロックされたままです。ロック・トークンが UOW のために戻されます。クライアントはこのロック・トークンを使用して CQSDDEL 要求を再度出す必要があります。</li> <li>クライアント状況は「移動」です。CQS 状況は「ロック」です。CQS は UOW のデータ・オブジェクトがロック状態であることを検出しました。CQSMOVE は、新しいキュー名が利用不能であるため、完了できませんでした。ロック・トークンが UOW のために戻されます。クライアントはこのロック・トークンを使用して CQSMOVE 要求を再度出す必要があります。</li> <li>クライアント状況は「アンロック」です。CQS 状況は「ロック」です。CQS はロックすべき UOW を検出しましたが、データ・オブジェクトをアンロックできませんでした。データ・オブジェクトはロックされたままです。ロック・トークンが UOW のために戻されます。クライアントはこのロック・トークンを使用して CQSUNLCK 要求を再度出す必要があります。</li> </ul>
X'00F2'	<p>不明</p> <p>クライアント状況は、有効ないずれかのクライアント状況です。UOW は CQS には不明です。</p> <p>UOW が「書き込み完了」状況であるとクライアントが確信している場合、クライアントは、CQSPUT 要求を再度出すかどうかを判断する必要があります。</p> <p>UOW が「削除」、「移動」、「読み取り」、または「アンロック」の状況であるとクライアントが確信している場合、前の要求は完了している可能性があります。</p>

### resynctoken

クライアントが UOW の処理を完了させるために使用するトークンを受け取るために使用する、16 バイトの出力パラメーター。状態が「書き込み同期」である場合、このフィールドには PUT トークンが入ります。状態が「ロック」である場合、このフィールドにはロック・トークンが入ります。このフィールドは、CQS によって CQSRSYNC 要求の結果として戻されます。

### compcode

要求からの完了コードを受け取るための 4 バイトの出力フィールド。完了コードには、以下のものがあります。

**X'00000000'**

CQS はこの UOW を正常に処理した。クライアントおよび CQS はこの UOW に関して同期状態です。「同期」状態がこの UOW に戻されます。

**X'00000004'**

CQS はこの UOW を正常に処理した。クライアントおよび

CQS はこの UOW に関して同期状態ではありません。CQS はこの UOW に関して認識している状態を戻します。

**X'00000008'**

*clientstatus* が無効。CQS はこの UOW を再同期できませんでした。*cqsstate* は戻されません。

**X'0000000C'**

*uow* が無効。CQS はこの UOW を再同期できませんでした。*cqsstate* は戻されません。

**X'00000010'**

CQS 内部エラー。CQS はこの UOW を再同期できませんでした。*cqsstate* は戻されません。

#### **LISTVER=1 | listversion**

このリスト・バージョンに対する等価を指定する入力パラメーター。

CQSRSYNC リスト・バージョンのためにプログラムの中に等価 (EQU) ステートメントを組み込むには、CQSRSYNC 要求の DSECT 機能を使用します。

#### **PARM=parmaddress**

要求が、CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSRSYNC\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

#### **RETCODE=returncodeaddress**

CQSRSYNC の戻りコードが入る 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

#### **RSNCODE=reasoncodeaddress**

CQSRSYNC の理由コードが入る 4 バイト・フィールドのアドレスを指定する出力パラメーター。

### **CQSRSYNC の戻りコードと理由コード**

以下の表は、CQSRSYNC 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSRSYNC FUNC=DSECT 要求を使用します。

表 23. CQSRSYNC の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求は正常に完了し、すべてのリスト項目は同期している。構造イベント出口ルーチンが CQS 再同期のために呼び出されます。その結果、クライアントはこの構造のためにデータを書き込むかまたは検索する CQS 要求を出すことができるようになります。

表 23. CQSRSYNC の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'00000004'	X'00000110'	CQS はコールド・スタートされた。リスト項目は処理されませんでした。CQS は未解決の UOW を検出されませんでした。構造イベント出口ルーチンは呼び出されません。その結果、クライアントはこの構造のためにデータを書き込むかまたは検索する CQS 要求を出すことができるようになります。
X'00000004'	X'00000114'	クライアントはコールド・スタートされた。CQS は未解決の UOW を検出されませんでした。構造イベント出口ルーチンは呼び出されません。その結果、クライアントはこの構造のためにデータを書き込むかまたは検索する CQS 要求を出すことができるようになります。
X'00000004'	X'00000118'	CQS はコールド・スタートされた。リスト項目は処理されませんでした。CQS はいくつかの未解決 UOW を検出し、それらにコールド状況であるとするマークを付けました。構造イベント出口ルーチンが呼び出されて、未解決 UOW のクライアントに通知を行います。その結果、クライアントはこの構造のためにデータを書き込むかまたは検索する CQS 要求を出すことができるようになります。
X'00000004'	X'0000011C'	クライアントはコールド・スタートされた。CQS はいくつかの未解決 UOW を検出しました。構造イベント出口ルーチンが呼び出されて、未解決 UOW のクライアントに通知を行います。その結果、クライアントはこの構造のためにデータを書き込むかまたは検索する CQS 要求を出すことができるようになります。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。リスト項目は処理されませんでした。構造イベント出口ルーチンは呼び出されません。クライアントは CQSRSYNC 要求を再度出す必要があります。
X'00000008'	X'00000214'	<i>connecttoken</i> が無効。リスト項目は処理されませんでした。構造イベント出口ルーチンは呼び出されません。クライアントは CQSRSYNC 要求を再度出す必要があります。
X'00000008'	X'00000218'	<i>FUNC</i> が無効。クライアントは CQSRSYNC 要求を再度出す必要があります。
X'00000008'	X'00000254'	<i>listaddress</i> が無効。リスト項目は処理されませんでした。構造イベント出口ルーチンは呼び出されません。クライアントは CQSRSYNC 要求を再度出す必要があります。
X'00000008'	X'0000027C'	CQSRSYNC は、リソース構造に対して許可されていない。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000008'	X'00000288'	リスト・バージョンが無効。

表 23. CQSRSYNC の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'0000000C'	X'00000300'	要求は少なくとも 1 つのリスト項目については成功したが、すべてではない。少なくとも 1 つのリスト項目が同期状態です。個々のエラーの各リスト項目については、 <i>compcode</i> を参照してください。構造イベント出口ルーチンが CQS 再同期のために呼び出されます。その結果、クライアントはこの構造のためにデータを書き込むかまたは検索する CQS 要求を出すことができるようになります。
X'0000000C'	X'00000304'	要求はすべてのリスト項目について失敗した。どのリスト項目も同期状態ではありません。個々のエラーの各リスト項目については、 <i>compcode</i> を参照してください。構造イベント出口ルーチンが CQS 再同期のために呼び出されます。その結果、クライアントはこの構造のためにデータを書き込むかまたは検索する CQS 要求を出すことができるようになります。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。
X'00000014'	X'00000500'	CQS 内部エラー。

## CQSSHUT 要求

CQSSHUT 要求は、CQS に、すべてのクライアントを切断した後に終了するように通知します。

### CQSSHUT の形式

#### CQSSHUT の DSECT 機能

CQSSHUT パラメーター・リストの長さ、および CQSSHUT の戻りコードと理由コードのために、プログラムに等価 (EQU) ステートメントを組み込むには、CQSSHUT 要求の DSECT 機能を使用します。

▶▶—CQSSHUT—FUNC=DSECT—▶▶

#### CQSSHUT の QUIESCE 機能

CQS を終了するには、CQSSHUT 要求の QUIESCE 機能を使用します。

▶▶—CQSSHUT—FUNC=QUIESCE—CQSTOKEN=*cqstokenaddress*—PARAM=*parmaddress*—▶▶

▶▶—  
 RETCODE=*returncodeaddress*—RSNCODE=*reasoncodeaddress*—▶▶  
 ECB=*ecbaddress*—

### CQSSHUT の使用法

CQSSHUT 要求は、CQS に、すべてのクライアントを切断した後に終了するように通知します。CQSSHUT 要求が出された後、CQS は CQSCONN 要求の受け入れを停止します。CQS は、入力または出力の要求の受け入れは継続し、クライアント

が処理中の作業を完了できるようにします。シャットダウン・プロセスを完了させるには、クライアントが作業を停止し、CQSDISC 要求を出して CQS から切断する必要があります。すべてのクライアントが切断された後、CQS はすべてのタスクを終了して、z/OS へ制御を戻します。

#### パラメーター説明

##### **CQSTOKEN=cqstokenaddress**

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

##### **ECB=ecbaddress**

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

##### **PARM=parmaddress**

要求が、CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSSHUT\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

##### **RETCODE=returncodeaddress**

CQSSHUT の戻りコードを含む 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

##### **RSNCODE=reasoncodeaddress**

CQSSHUT の理由コードを含む 4 バイト・フィールドのアドレスを指定する出力パラメーター。

## **CQSSHUT の戻りコードと理由コード**

以下の表は、CQSSHUT 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSSHUT FUNC=DSECT 要求を使用します。

表 24. CQSSHUT の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000218'	FUNC が無効。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000010'	X'00000444'	CQS 初期設定が進行中。初期設定の完了後に CQSSHUT 要求を再度出してください。

## CQSUNLCK 要求

CQSUNLCK 要求は、1 つ以上のデータ・オブジェクトをアンロックし、それらのデータ・オブジェクトをキューの最初または最後の位置に移動します。FUNC=FORCE を指定して、アンロックを強制することもできます。

### CQSUNLCK の形式

#### CQSUNLCK の DSECT 機能

CQSUNLCK パラメーター・リストの長さ、および CQSUNLCK の戻りコードと理由コードのために、プログラムの中に等価 (EQU) ステートメントを組み込むには、CQSUNLCK 要求の DSECT 機能を使用します。

▶▶—CQSUNLCK—FUNC=DSECT—▶▶

#### CQSUNLCK の UNLOCK 機能

1 つ以上のデータ・オブジェクトをアンロックして、それらのデータ・オブジェクトをキューの最後または最初に移動するには、CQSUNLCK の UNLOCK 機能を使用します。

▶▶—CQSUNLCK—FUNC=UNLOCK—CQSTOKEN=cqstokenaddress—▶▶

▶—CONTOKEN=connecttokenaddress—PARM=parmaddress—COUNT=count—▶

▶—LIST=listaddress—

LISTVER=1
LISTVER=listversion

—ECB=ecbaddress—▶

▶—RETCODE=returncodeaddress—RSNCODE=reasoncodeaddress—▶▶

#### CQSUNLCK の FORCE 機能

CQSUNLCK 要求の FORCE 機能を使用して、指定された障害のあった CQS クライアントによって指定されていたキュー・タイプから読み取られたデータ・オブジェクトを強制的にアンロックし、そのデータ・オブジェクトの CQS 認識をクリーンアップしてください。

▶▶—CQSUNLCK—FUNC=FORCE—CQSTOKEN=cqstokenaddress—▶▶

▶—CONTOKEN=connecttokenaddress—PARM=parmaddress—CLIENT=clientnameaddress—▶

▶—COUNT=count—QTYPE=queuetype—

ECB=ecbaddress
----------------

—RETCODE=returncodeaddress—▶

▶—RSNCODE=reasoncodeaddress—▶▶

## CQSUNLCK の使用法

制約事項: CQSUNLCK 要求は、リソース構造に対してサポートされていません。

CQSUNLCK FUNC=UNLOCK 要求は、1 つ以上のデータ・オブジェクトをアンロックし、それらのデータ・オブジェクトをキューの最初または最後の位置に移動します。クライアントは、1 つ以上のリスト項目を含むアンロック・リストを渡しますが、このリストの各項目は別々のアンロック要求です。CQSUNLCK 要求が正常に終了すると、ロック・トークンが無効にされ、CQSBRWSE、CQSDDEL、CQSMOVE、または CQSREAD 要求では、どのクライアントもそのデータ・オブジェクトを使用できるようになります。

CQSUNLCK FUNC=FORCE 要求を使用すると、CQS クライアントが、指定された障害のあった CQS クライアントが指定しているキュー・タイプから読み取られたデータ・オブジェクトを強制的にアンロックすることができます。そのため、その障害 CQS クライアントが再始動するまで、そのデータ・オブジェクトは LOCKQ 上に残りません。この CQS が、データ・オブジェクトをロックした CQSREAD 要求を処理した場合、強制アンロックは、ロック状態のデータ・オブジェクトの CQS 認識も除去します。

CQS クライアントに障害が起こった場合、そのロック状態のデータ・オブジェクトは、CQS クライアントが再始動して CQS と再同期し、そのロック状態のデータ・オブジェクトをどうするかについて決定するまで、あるいは別の CQS クライアントがそのデータ・オブジェクトを強制的にアンロックするまで、LOCKQ 上に残ります。ロック状態のデータ・オブジェクトは、他の CQS クライアントからはアクセス可能ではありません。

重要: CQS クライアントは、CQSUNLCK FUNC=FORCE 要求を慎重に使用しなければなりません。IMSplex 内の CQS クライアントは、以下の強制アンロック・ルールを一貫して適用する必要があります。一貫して使用しなかった場合、CQSRSYNC 要求は失敗する可能性があり、データ・オブジェクトがロック・キュー上に残る、読み取りテーブルが CQS 内に残る、あるいはデータ・オブジェクトが COLDQ に移動する場合があります。CQSUNLCK FUNC=FORCE を使用する場合、以下のルールを適用してください。

- CSL を使用して IMSplex を定義する。

IMSplex は、共通サービス層によって定義する必要があります。そうすることで、CQS クライアントに障害が起こったときに、CQS クライアントに通知されます。

- キュー・タイプ候補を選択する。

強制的にアンロックされる候補となるデータ・オブジェクトを持つ 1 つ以上のキュー・タイプを選択する。指定されたキュー・タイプを持つデータ・オブジェクトのすべてがその候補です。強制的にアンロックされるキュー・タイプの特定データ・オブジェクトを選択することはできません。

- CQS クライアントに障害が起こった場合は、別の CQS クライアントのデータ・オブジェクトを強制的にアンロックする。

CQS クライアントに障害が起こった場合、LOCKQ 上にロックされたデータ・オブジェクトが残される場合があります。障害 CQS が再始動するまでデータ・

オブジェクトが LOCKQ 上に残らないようにするため、別の CQS クライアントが CQSUNLCK FUNC=FORCE 要求を発行する必要があります。

CQSUNLCK FUNC=FORCE 要求を発行し、現時点ではアクティブではない CQS クライアントのデータ・オブジェクトを強制的にアンロックしてください。宛先 CQS クライアントがアクティブではないことを確認するために、CQS クライアントが CQSUNLCK FUNC=FORCE 要求を発行する必要があります。

IMSpdex 内の CQS クライアントが、CQSUNLCK FUNC=FORCE 要求を発行している CQS クライアントは 1 つだけであることを確認する必要があります。メンバーの 1 つに障害が起こった場合、CSL によって定義された IMSplex 内のすべてのメンバーに通知されます。CQSUNLCK FUNC=FORCE 要求を複数発行することで、以下の望ましくない結果になる場合があります。

- 不必要な CF アクセス

CQSUNLCK FUNC=FORCE 要求は、複数の CF アクセスが候補キュー・タイプ上のデータ・オブジェクトを検索する状態を引き起こします。複数の CQSUNLCK FUNC=FORCE 要求が発行された場合、それぞれの要求が同じ CF アクセスを無数に作成します。これら余分の CF アクセスは不必要で、さらなるパフォーマンス・オーバーヘッドを引き起こします。不必要な CF アクセスによるパフォーマンス・オーバーヘッドが受諾不能な場合、IMSpdex 内の CQS クライアントは、CQSUNLCK FUNC=FORCE 要求を発行している CQS クライアントは 1 つだけであることを確認する必要があります。

IMSpdex 内の CQS クライアントは、CQSUNLCK FUNC=FORCE 要求を正常に発行している CQS クライアントはただ 1 つだけであることを確認する必要があります。CQS クライアントが CQSUNLCK FUNC=FORCE 要求を発行し、かつ CQSUNLCK エラー、構造障害、リンクの逸失、その他の障害が起こった場合、IMSpdex 内の CQS クライアントは、そのエラーが訂正された後に CQSUNLCK FUNC=FORCE 要求が正常に発行されることを確実にする必要があります。

- 誤ってアンロックされたデータ・オブジェクト

障害 CQS クライアントが直ちに初期化する場合、別の CQS クライアントが障害 CQS クライアントのデータ・オブジェクトを強制的にアンロックしようとする前に、その障害 CQS がそれ自身のデータ・オブジェクトを強制的にアンロックし、CQS と再同期し、キュー構造上に新規データ・オブジェクトを置く場合があります。もう 1 つの CQS クライアントが、処理中の UOW に対するデータ・オブジェクトを誤ってアンロックする場合があります。IMSpdex 内の CQS クライアントは、指定されたクライアントに対するデータ・オブジェクトを強制的にアンロックする CQS クライアントはただ 1 つだけであることを確認する必要があります。

- CQS クライアントが初期化される時、CQS クライアント自身のデータ・オブジェクトを強制的にアンロックする

CQS クライアントが初期化される時、CQSRSYNC を発行する前にそれ自身のデータ・オブジェクトを強制的にアンロックする必要があります。これにより、障害時に強制アンロックを行う他の CQS クライアントが使用可能ではなかった場合、CQS クライアントのデータ・オブジェクトは、再同期の前に確実にアン

ロックされます。この CQS が、データ・オブジェクトをロックした CQSREAD 要求を処理したため、強制アンロックは、IMS クライアントのロック状態のデータ・オブジェクトの CQS 認識もクリーンアップします。

- 強制アンロックの候補の UOW を処理して、CQS と再同期する

CQSRSYNC 要求で CQS にパスする再同期リストを再作成する場合、強制の CQS クライアント状況での UNLOCK FORCE に対する候補のすべてにマークを付けてください。CQS 再同期は、強制のクライアント状況をチェックし、同期のアンロックの CQS 状況に対する UOW をセットします。

- CQS クライアントが初期化される時に、その他の障害 CQS クライアントのデータ・オブジェクトを強制的にアンロックする

CQS クライアントが初期化される時、障害 CQS クライアントのデータ・オブジェクトを強制的にアンロックする必要があります。これは CQS クライアントに障害が起こったときに、使用可能なその他の CQS クライアントが存在せず、強制を実行することができなかったためです。初期化している CQS クライアントが CQS と再同期した後で、1 つの障害 CQS クライアントに対して CQSUNLCK FUNC=FORCE 要求を 1 つ発行し、候補キュー・タイプ上のデータ・オブジェクトを強制的にアンロックする必要があります。

#### パラメーター説明

##### **CLIENT=clientnameaddress**

データ・オブジェクトを強制的にアンロックする CQS クライアントを指定する 8 バイトの入力フィールド。クライアント名は、そのクライアントが CQS に登録されたときに CQSREG 上で指定された名前と同じです。CQS クライアントは、CQSRSYNC 要求が発行される前に、それ自身のロック状態データ・オブジェクトを強制的にアンロックすることができます。CQS クライアントは、CQSRSYNC 要求が発行された後で、別の CQS クライアントのロック状態データ・オブジェクトを強制的にアンロックすることができます。

##### **CONTOKEN=connecttokenaddress**

この CQS が管理する特定のカップリング・ファシリティー構造に対するクライアントの接続を一意的に識別する、16 バイトの接続トークンのアドレスを指定する入力パラメーター。接続トークンは、CQSCONN 要求によって戻されません。

##### **COUNT=count**

アンロック・リスト内のリスト項目の数を指定する、4 バイトの入力パラメーター、または強制的にアンロックされたデータ・オブジェクトの数を受信する、4 バイトの出力パラメーター。

##### **CQSTOKEN=cqstokenaddress**

クライアントの CQS に対する接続を一意的に識別する、16 バイトの CQS 登録トークンのアドレスを指定する入力パラメーター。登録トークンは、CQSREG 要求によって戻されます。

##### **ECB=ecbaddress**

非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定する 4 バイトの入力パラメーター。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

### **LIST=listaddress**

アンロック・リストのアドレスを指定する 4 バイトの入力パラメーター。各項目が別々の CQSUNLCK 要求になります。各項目の一部のフィールドは、CQSUNLCK 要求の前にクライアントによって初期化される必要があります。他のフィールドは CQSUNLCK 要求の完了時に CQS によって戻されます。

CQSUNLL リスト項目の DSECT は、リスト項目をマップし、クライアントが使用することができます。複数のリスト項目は、連続するストレージに常駐させる必要があります。

各リスト項目の内容は次の通りです。

#### **compcode**

要求からの完了コードを受け取るための 4 バイトの出力フィールド。完了コードには、以下のものがあります。

**X'00000000'**

要求が正常に完了した。

**X'00000004'**

*locktoken* が無効。

**X'00000008'**

構造がアクセス不能。後で要求を再試行してください。

**X'0000000C'**

データ・オブジェクト用の元のキューがいっぱいなために、データ・オブジェクトをアンロックできない。指定されたキューにデータ・オブジェクトを移動することはできませんが、他のキューの CQSUNLCK 要求は許可されています。

**X'00000010'**

CQS 内部エラー。

**X'00000014'**

構造が再作成されたためにデータ・オブジェクトが失われた。データ・オブジェクトはリカバリー不能だったので、データ・オブジェクトがロックされた後に再作成が起きました。現在データ・オブジェクトは失われています。

**qpos** アンロックされたエレメントを追加するキュー上の位置を指示する 1 バイトの入力パラメーター。

**X'00'** 元のクライアント・キュー位置。このデータ・オブジェクトをロックした CQSREAD 要求が最初のデータ・オブジェクトを読み取る場合、この要求はデータ・オブジェクトをアンロックし、キューの最初にそれを追加します。CQSREAD 要求が最後のデータ・オブジェクトを読み取る場合、この要求はデータ・オブジェクトをアンロックし、キューの最後にそれを追加します。

**X'01'** キューの最後。

**X'02'** キューの最初。

### locktoken

CQSREAD 要求によってロックされたデータ・オブジェクトを一意的に識別するロック・トークンを指定する、6 バイトの入力パラメーター。このパラメーターは必須です。

### LISTVER=1 | listversion

このリスト・バージョンに対する等価を指定する入力パラメーター。  
CQSUNLCK リスト・バージョンのためにプログラムの中に等価 (EQU) ステートメントを組み込むには、CQSUNLCK 要求の DSECT 機能を使用します。

### PARM=parmaddress

要求が、CQS へパラメーターを渡すために使用するパラメーター・リストのアドレスを指定する、4 バイトの入力パラメーター。ストレージ域の長さは、最低でも EQU 値の CQSUNLCK\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

### QTYPE=queuetype

読み取られたロック状態データ・オブジェクトからのキュー・タイプを指定する、4 バイトの入力パラメーター。キュー・タイプの有効な値は、1 ~ 255 (10 進) です。

### RETCODE=returncodeaddress

CQSUNLCK 要求の戻りコードを含む 4 バイト・フィールドのアドレスを指定する出力パラメーター。

レジスター 15 に入っている戻りコードがゼロ以外の場合、CQS インターフェースがエラーを検出したために CQS に要求を送信できなかったため、戻りコードと理由コードのフィールドにある値は無効です。

### RSNCODE=reasoncodeaddress

CQSUNLCK 要求の理由コードを入れる 4 バイト・フィールドのアドレスを指定する出力パラメーター。

## CQSUNLCK の戻りコードと理由コード

以下の表は、CQSUNLCK 要求で返される戻りコードと理由コードの組み合わせの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSUNLCK FUNC=DSECT 要求を使用します。

表 25. CQSUNLCK の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000214'	<i>connecttoken</i> が無効。
X'00000008'	X'00000218'	FUNC が無効。
X'00000008'	X'00000240'	<i>queuetype</i> が無効。
X'00000008'	X'00000244'	<i>clientname</i> が無効。
X'00000008'	X'00000250'	<i>count</i> が無効。
X'00000008'	X'00000254'	<i>listaddress</i> が無効。
X'00000008'	X'0000027C'	CQSUNLCK は、リソース構造に対して許可されていない。



## CQSUPD の使用法

CQSUPD は、リソース構造上にある、一意的な名前の付いた、1 つ以上のリソースを作成または更新します。CQSUPD は、リソースが存在しない場合はリソースを作成し、存在する場合はリソースを更新します。クライアント・データがあってもなくても、リソースは作成または更新できます。リソースの例には、トランザクションおよび制御ブロックが含まれます。

パラメーター説明

### **CONTOKEN=connecttokenaddress**

この CQS が管理する特定のカップリング・ファシリティ構造に対するクライアントの接続を一意的に識別する接続トークンを指定する、16 バイトの入力パラメーターのアドレス。接続トークンは、CQSCONN 要求によって戻されません。

### **COUNT=resourcecount**

リスト内の項目の数を指定する、4 バイトの入力パラメーター。

### **CQSTOKEN=cqstokenaddress**

クライアントの CQS に対する接続を一意的に識別する CQS 登録トークンを指定する、16 バイトの入力パラメーターのアドレス。登録トークンは、CQSREG 要求によって戻されます。

### **ECB=ecbaddress**

非同期要求に使用される z/OS イベント制御ブロック (ECB) を指定する 4 バイトの入力パラメーターのアドレス。ECB が指定されていると、要求は非同期的に処理され、それ以外の場合は同期的に処理されます。

### **LISTSIZE=resourcesize**

リソース・リストのサイズを指定する 4 バイトの入力パラメーター。リスト内の各エントリーは可変長である場合があるので、リスト・サイズを指定する必要があります。

### **LISTVER=1 | listversion**

このリスト・バージョンに対する等価を指定する入力パラメーター。CQSUPD リスト・バージョンのためにプログラムの中に等価 (EQU) ステートメントを組み込むには、CQSUPD 要求の DSECT 機能を使用します。

### **LIST=resourceaddress**

1 つ以上のエントリーを含む、可変サイズのリソース・リストを指定する入力パラメーターのアドレス。各項目が別々の更新要求になります。各項目の一部のフィールドは、CQSUPD 要求の前にクライアントによって初期化される必要があります。他のフィールドは、要求の完了時に CQS によって戻されます。

CQSUPDL リスト項目の DSECT は、リスト項目をマップし、クライアントが使用することができます。複数のリスト項目は、連続するストレージに常駐させる必要があります。

各リスト項目には以下のフィールドが含まれています。

#### **listentrylength**

リスト項目の長さを指定する 4 バイトの入力フィールド。リスト項目の長さは可変で、指定されている場合は data2 により変化します。このパラメーターは必須です。

### ***resourceid***

リソース構造上で作成または更新されるリソースの固有 ID を含む、12 バイトの入力フィールド。リソース ID は IMSplex 内で固有です。リソース ID は、11 バイトのクライアント定義のリソース名を伴った、1 バイトの名前タイプから構成されます。名前タイプは、同じ名前タイプを持つリソースに対するクライアント定義の名前の固有性を保証します。異なるリソース・タイプのリソースが同じ名前タイプを持つ場合があります。名前タイプに対して有効な値は、1 から 255 までの 10 進数です。クライアント定義の名前は、クライアントに対して意味を持ち、英数字から構成されます。このパラメーターは必須です。

### ***resourcetype***

リソース・タイプを指定する 1 バイトのフィールド。リソース・タイプは、リソース構造上のリソースの、クライアント定義の物理的グループ化です。リソース・タイプに対して有効な値は、1 から 255 までの 10 進数です。リソース・タイプが、CQS によって定義されたリソース・タイプの最大数 (11) より大きい場合、折り返されて既存のリソース・タイプのうちの 1 つになります。このパラメーターは必須です。

### ***reserved***

3 バイトの予約フィールド。

### ***options***

更新オプションを指定する 4 バイトの入力フィールド。このパラメーターはオプションです。可能なオプションは以下のとおりです。

#### **X'80000000'**

バージョンのミスマッチにより更新が失敗した場合、*data1* および *owner* を戻す。これは、追加 CF アクセスによるパフォーマンス・オーバーヘッドを引き起こします。

#### **X'40000000'**

バージョンのミスマッチにより更新が失敗した場合、*data2*、*data1*、および *owner* を戻す。*data2buffer* および *data2buffersize* が指定されている場合、*data2* が戻される。これは、追加 CF アクセスによるパフォーマンス・オーバーヘッドを引き起こします。

#### **X'20000000'**

*data2* を削除する。

### ***compcode***

要求からの完了コードを受け取るための 4 バイトの出力フィールド。完了コードには、以下のものがあります。

#### **X'00000000'**

要求が正常に完了した。

#### **X'00000004'**

要求が正常に完了したが、部分的データのみが *data2buffer* に戻された。

#### **X'00000020'**

*Resourceid* は無効。名前タイプは、1 から 255 までの 10 進数である必要があります。

**X'00000024'**

CQS 内部エラー。

**X'00000028'**

バージョン が既存のリソースと一致しない。

**X'00000030'**

リソースが別の名前タイプとしてすでに存在している。

**X'00000034'**

構造がいっぱい。

**X'00000038'**

*Resourcetype* は無効。リソース・タイプは、1 から 255 までの 10 進数である必要があります。

**X'0000003C'**

*Listentrylength* は無効。リスト項目の長さは、最小リスト項目の長さより大きいか等しい非ゼロの数でなければなりません。  
CQSUPDL DSECT を参照してください。

**X'00000040'**

構造がアクセス不能。

**X'00000044'**

CQS アドレス・スペースがない。

#### バージョン

リソースのバージョンを指定する 8 バイトの入力および出力フィールド。バージョンは、リソースが更新された回数です。初期 CQSUPD 要求がリソースを作成するためには、*version* は入力上でゼロである必要があります。以降の CQSUPD 要求が既存のリソースを更新するためには、*version* が既存のリソースのバージョンと一致する必要があります。CQSUPD 要求は *version* を 1 ずつ増加させ、リソースを新しいバージョンに更新し、新規 *version* を出力として戻します。*version* のミスマッチのために、既存のリソースを更新する CQSUPD 要求が失敗した場合、CQS は正しいバージョンをクライアントに出力として戻します。このパラメーターは必須です。データ・オブジェクトが作成される場合、入力上では *version* は無視され、1 の *version* が出力として戻されます。

#### owner

リソースの所有者を指定する 8 バイトの入力および出力フィールド。入力上では、*owner* はリソースに対してセットされます。リソースに所有者をセットしない場合はゼロを指定してください。所有者が 1 つだけ許可されます。バージョンのミスマッチのために更新要求が失敗し、所有者へ戻すオプションが指定されている場合、既存リソースの所有者は出力として戻されます。このパラメーターは必須です。

#### data1

更新されるリソースに対する少量のクライアント・データ *data1* を指定する 24 バイトの入力および出力フィールド。*data1* にクライアント・データをセットしない場合はゼロを指定してください。バージョンのミスマッチのために CQSUPD 要求が失敗し、*data1* を戻すオプションが指定されている場合、既存リソースの *data1* は出力として戻されます。*data1* によって指定されたクライアント・データにアクセスするパフォ

ーマンスは、*data2* によって指定されたクライアント・データへのアクセスより高速です。このパラメーターは必須です。

#### *data2size*

更新されるリソースに対する *data2buffer* 内のクライアント・データ *data2* のサイズを指定する 4 バイトの入力および出力フィールド。更新する *data2* が存在しない場合、入力時にゼロを指定してください。バージョンのミスマッチのために CQSUPD が失敗し、*data2* を戻すオプションがセットされている場合、既存のリソースの *data2* サイズは出力として戻されます。このパラメーターはオプションです。

#### *data2buffersize*

更新される、または出力として戻されるリソースに対するクライアント・データ *data2* を含んでいる *data2buffer* のサイズを指定する 4 バイトの入力フィールド。指定できる最大サイズは 61312 バイト (X'EF80') です。*data2* を更新する、または出力として戻す必要がない場合は、ゼロを指定してください。このパラメーターはオプションです。

#### *data2buffer*

更新されるリソースに対する大量のクライアント・データ *data2* を指定する、可変サイズの入力および出力バッファ。バージョンのミスマッチのために CQSUPD が失敗し、かつ *data2* が指定されている場合、既存リソースの *data2* は、*data2buffer* に入る限り戻されます。このパラメーターはオプションです。

#### **PARM=parmaddress**

要求が、CQS へパラメーターを渡すために使用する入力パラメーター・リストのアドレス。ストレージ域の長さは、最低でも EQU 値の CQSUPD\_PARM\_LEN (FUNC=DSECT 要求を使用して定義される) と等しくなければなりません。

#### **RETCODE=returncodeaddress**

CQSUPD 戻りコードを含んでいる 4 バイトの出力フィールドのアドレス。レジスター 15 の戻りコードが非ゼロの場合、CQS がエラーを検出し、要求を処理しなかったために、*returncodeaddress* および *reasoncodeaddress* に対して戻される値は有効ではありません。

#### **RSNCODE=reasoncodeaddress**

CQSUPD 理由コードを含んでいる 4 バイトの出力フィールドのアドレス。

### **CQSUPD の戻りコードと理由コード**

以下の表は、CQSUPD 要求で返される戻りコードと理由コードの一覧です。戻りコードと理由コードのために、プログラムの中に EQU ステートメントを組み込むには、CQSUPD=DSECT 要求を使用します。

表 26. CQSUPD の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'00000008'	X'00000210'	<i>cqstoken</i> が無効。
X'00000008'	X'00000214'	<i>contoken</i> が無効。
X'00000008'	X'00000218'	FUNC が無効。

表 26. CQSUPD の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'00000008'	X'00000250'	<i>resourcelistcount</i> が無効。
X'00000008'	X'00000254'	<i>listaddress</i> が無効。
X'00000008'	X'00000280'	要求はキュー構造に対して許可されていない。
X'00000008'	X'00000284'	パラメーター・リスト・バージョンが無効。
X'00000008'	X'00000288'	リスト・バージョンが無効。
X'0000000C'	X'00000300'	要求は少なくとも 1 つのリスト項目については成功したが、すべてではない。個々のエラーについては、 <i>compcode</i> を参照してください。
X'0000000C'	X'00000304'	要求は、すべての項目について失敗した。個々のエラーについては、 <i>compcode</i> を参照してください。
X'0000000C'	X'00000308'	バージョンのミスマッチのために、要求が、1 つ以上のリスト項目に対して失敗した。これらのリソースは、 <i>resourcetype</i> が指定したとおりにすでに存在している。他のすべての項目は成功した。
X'00000010'	X'00000430'	CQS アドレス・スペースがない。
X'00000014'	X'00000500'	内部エラー。



---

## 第 2 部 共通サービス層 (CSL) (Common Service Layer (CSL))

本書に含まれるトピックでは、CSL に関する情報を記載しています。



---

## 第 3 章 CSL クライアントの作成

以降のトピックでは、CSL クライアントの作成時の考慮事項について説明します。このセクションの内容は、クライアントを作成するプログラマー用ですが、CSL クライアントの設計および作成に伴う問題を認識する必要のある CSL 管理者または IMS システム・プログラマーも対象にしています。

---

### CSL 要求でのイベント制御ブロック

イベント制御ブロック (ECB) は CSL 要求に含まれるオプション・パラメーターで、z/OS ECB のアドレスを指定するために使用できます。

ほとんどの CSL 要求は、ECB の指定を容認しています。ECB パラメーターはオプションであり、z/OS ECB のアドレスを指定します。CSL 要求が完了すると、ECB パラメーターで指定された ECB は通知を受け取ります。このパラメーターが指定されていないと、要求側モジュールは、要求が完了するまで制御を受け取りません。

ECB が指定された場合、要求の起動側は、要求から制御を受け取った後、この要求から戻されたデータ (RETCODE および RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を出す必要があります。WAIT が出されない場合、データが無効である可能性があります。

---

### SCI 要求の環境要件

SCI 要求の環境要件は、そのクライアントに割り当てられた SCI インターフェースによって異なります。

以下の表では、許可 SCI 要求の環境について説明します。

表 27. 許可 SCI インターフェースを使用する SCI 要求の環境

環境特性	要件
許可	監視プログラム状態 (PSW キーは、CSLSCREG 要求が出されたときの PSW キーと一致していることが必要)
ディスパッチ可能単位モード	タスク
仮想記憶間モード	任意。ただし、CSLSCREG 要求が出された 1 次アドレス・スペース と PASN が同じであることが必要。
AMODE	31
ASC モード	1 次
ホーム・アドレス・スペース	任意
ロック	ロックは保持されない
割り込み状況	割り込み可能
制御パラメーター	1 次アドレス・スペース内

以下の表では、非許可 SCI 要求の環境について説明します。

表 28. 非許可インターフェースを使用する SCI 要求の環境

環境特性	要件
許可	問題プログラム状態 (PSW キーは、CSLSCREG 要求が出されたときの PSW キーと一致していることが必要)
ディスパッチ可能単位モード	タスク
仮想記憶間モード	なし (PASN=SASN=HASN)
AMODE	31
ASC モード	1 次
ホーム・アドレス・スペース	CSLSCREG が出されたアドレス・スペース
ロック	ロックは保持されない
割り込み状況	割り込み可能
制御パラメーター	1 次アドレス・スペース内

SCI の登録および登録解除要求 (CSLSCREG および CSLSCDRG) の環境要件は、他のすべての SCI 要求と異なります。許可クライアントは、以下の表に説明する環境で CSLSCREG 要求および CSLSCDRG 要求を発行する必要があります。

表 29. 許可インターフェースを使用する CSLSCREG および CSLSCDRG 要求の環境

環境特性	要件
許可	監視プログラム状態
ディスパッチ可能単位モード	タスク
仮想記憶間モード	なし (PASN=SASN=HASN)
AMODE	31
ASC モード	1 次
ロック	ロックは保持されない
割り込み状況	割り込み可能
制御パラメーター	1 次アドレス・スペース内

非許可クライアントは、以下の表に説明する環境で CSLSCREG 要求および CSLSCDRG 要求を発行する必要があります。

表 30. 非許可インターフェースを使用する CSLSCREG および CSLSCDRG 要求の環境

環境特性	要件
許可	問題プログラム状態
ディスパッチ可能単位モード	タスク
仮想記憶間モード	なし (PASN=SASN=HASN)
AMODE	31
ASC モード	1 次
ロック	ロックは保持されない
割り込み状況	割り込み可能
制御パラメーター	1 次アドレス・スペース内

---

## CSL 要求の戻りコードおよび理由コードの解釈方法

Common Service Layer (CSL) 戻りコードおよび理由コードは、CSL アドレス・スペースへの要求送信の成功または失敗を示し、行われている特定の CSL 要求の成功または失敗を反映します。

各 CSL 要求は、指定した要求の結果を示す戻りコードおよび理由コードを受け取ります。大部分の要求には複数のコンポーネントが関係しているため、それらの関係するコンポーネントのいずれかから、戻りコードおよび理由コードが発信される可能性があります。例えば、Structured Call Interface (SCI) はこのような要求に応える通信手段であるため、SCI の戻りコードと理由コードは受信することができます。戻りコードと理由コードを設定したコンポーネントを識別しやすくするために、高位バイトが使用されます。

上位バイトの取り得る値、およびそれぞれの値の意味は、次のとおりです。

X'00' IMS が戻りコードと理由コードを設定

X'01' SCI が戻りコードと理由コードを設定

X'02' Operations Manager (OM) が戻りコードと理由コードを設定

X'03' Resource Manager (RM) が戻りコードと理由コードを設定

X'04' ODBM が戻りコードと理由コードを設定

各 CSL 要求は、戻りコードと理由コードのテーブルを持ちます。ユーザーが出した要求に対する戻りコードと理由コードが検出できない場合は、戻りコードの高位バイトを使用すると、戻りコードと理由コードを設定したコンポーネントを識別しやすくなります。例えば、理由コードが X'01' (SCI) の場合は、CSLSCMSG マクロおよび CSLSCRQS マクロに対する戻りコードと理由コードを参照することから始めてください。

ODBM の理由コードは CSLDRR マクロで、OM の理由コードは CSLORR マクロで、RM の理由コードは CSLRRR マクロで定義されています。また、SCI の理由コードは CSLSRR マクロで定義されています。これらのマクロは IMS.SDFS MAC データ・セット内にあります。

関連資料:

125 ページの『CSLZQRY: 照会要求』

127 ページの『CSLZSHUT: シャットダウン要求』

---

## CSL 用のクライアント作成の計画に関する考慮事項

タスクの計画とは、CSL マネージャーおよび CSL 要求をどのように使用するかを決定することです。

タスクの計画とは、CSL マネージャーをどのように使用するかを決定することです。次のような決定をします。

- どの許可レベルを使用するか

プログラムを許可状態 (監視プログラム状態) で実行するのか、非許可状態 (問題プログラム状態) で実行するのかを決定する必要があります。SCI は、SCI に登録されたときのプログラムの状態と PSW キーに基づいて適切な環境を初期化します。

注: 非許可クライアントは、RM に登録できず、RM 要求を出せず、OM にコマンドを登録できず、CSLSCRQS を使用して発行された要求を処理することができません。

- **SCI** 出口ルーチンを使用するかどうか

SCI 入力出口ルーチンおよび通知出口ルーチンを使用するかどうかを決定する必要があります。例えば、OM コマンド処理クライアントは、SCI 入力出口がなければ、OM ディレクティブを処理できません。SCI 通知出口がなければ、IMSplex に新しい OM が結合されたとき、OM コマンド処理クライアントに通知できません。通知されなければ、そのクライアントは新しい OM に登録できません。

- **TCB** アソシエーション

SCI 登録 (CSLSCREG 要求を使います) は、IMSplex メンバーを特定の個別の TCB に関連付けることができます。使用する許可レベルは、TCB アソシエーションについても考慮が必要です。SCI は、指定された TCB に、登録を内部的に関連付けます。TCB が指定されていない場合、SCI は登録を発行した TCB にその登録を関連付けます。関連付けられた TCB が、登録解除を発行せずに終了すると、SCI は登録を異常終了して、SCI がメンバー・アドレス・スペースで割り振りを行った関連ストレージを解放します。これに続いて SCI 要求が出されると、異常終了が発生する場合があります。

- **RM** サービス、**OM** サービス、または **ODBM** サービスを使用するかどうか

ユーザーは、自分に固有のグローバル・リソースを自分で管理することができます。しかし、IMS グローバル・リソースをアクセスしたい場合、RM クライアントをコーディングする必要があります。

固有のコマンド・セットと固有の (コマンド登録とセキュリティーを整合させるための) コマンド処理クライアントの作成を計画している場合、OM コマンド処理クライアントを書くことができます。固有のコマンドを入力するための固有の SPOC または AOP の作成を計画している場合、OM AOP クライアントを書くことができます。OM の役割は、IMSplex 全体にわたってコマンドをトランスポートすること、およびコマンド応答を SPOC または AOP 用に XML タグで統合することです。

DBCTL または DB/DC 環境のいずれかで IMS DB が管理する IMS データベースにアクセスする場合、IMS トランザクションを使用しない ODBM クライアント・アプリケーション・プログラムを作成できます。ODBM は IMSplex 内のデータベース接続を管理し、アプリケーション・プログラムが IMS Universal ドライバーまたは ODBA インターフェースのいずれかを使用して IMSplex 内のデータベースにアクセスできるようにします。ODBM は、ODBA インターフェースを使用するアプリケーション・プログラムの予期しない終了からも、IMS 制御領域を保護します。

- 要求を出すときにメッセージ・プロトコルを使用するのか、要求プロトコルを使用するのか

同期応答が必要ない場合、または非同期応答が必要な場合、メッセージ・プロトコルを使用してください。メッセージ・プロトコルで送信される IMSplex コマンド応答は、非同期で送信されます。

- **CSL OM 監査証跡**

CSL OM 監査証跡には、RM、OM、および SCI などのアクティブ CSL アドレス・スペースが作成する通常の操作メッセージだけでなく、CSL コンポーネントを通じて IMS コンポーネントが作成する操作メッセージも含まれます。また、CSL クライアント・アクティビティーもキャプチャーされます。監査証跡は、コンプライアンス監査タスクや診断タスクで使用できます。

関連概念:

 CSL OM 監査証跡 (システム管理)

関連資料:

 BPE ベースの CSL SCI ユーザー出口ルーチン (出口ルーチン)

## CSL マネージャーの SCI への登録

ODBM、OM、および RM クライアントは、Structured Call Interface (SCI) に登録する必要があります。SCL マネージャーのいずれかを使用するためには、複数の登録ステップを完了しなければなりません。

どの CSL マネージャーを使用する場合も、まず最初に登録ステップを完了する必要があります。ODBM、OM、および RM クライアントは、SCI に登録する必要があります。このトピックでは、SCI 登録、ODBM、OM および RM クライアントの SCI での登録方法、SCI を作動可能状態にする方法、および、CSL 要求を発行する順序について説明します。

### SCI への登録

後続のすべての要求で使用される IMSplex メンバーと SCI との接続を一意的に識別するには、SCI に登録する必要があります。

SCI に登録するとき、次のものを示す必要があります。

- IMSplex の名前
- クライアント名。許可クライアントの場合、この名前は固有でなければなりません。
- 出口ルーチンを使う場合、出口ルーチン。
- アドレス・スペースのタイプ

アドレス・スペースのタイプには、AOP または OTHER を使用してください。AOP または OTHER でないタイプのアドレス・スペースを定義すると、IMS アドレス・スペースに干渉するおそれがあります。SUBTYPE パラメーターを使用することにより、クライアントをさらに詳しく識別できます。

SCI に登録すると、SCI トークンが戻されます。トークンは、IMSplex メンバーの SCI への接続を一意的に識別し、後続のすべての要求で使用されます。このトークンは、将来の ODBM、OM、RM、および SCI 要求に備えて保管しておいてください。

## ODBM クライアントの登録

ODBM に登録するには、クライアントはまず CSL SCI に登録してから、IMSplex 内のすべてのアクティブ ODBM に登録する必要があります。

IMS ODBA インターフェースを使用して IMS データベースにアクセスするアプリケーション・プログラムのアプリケーション・サーバーは、ODBM を子として登録することができます。ODBM は、IMSplex 内で IMS DB システムが管理するデータベースとの接続を管理し、アプリケーション・プログラムの予期しない終了から、IMS 制御領域を保護します。

ODBM クライアントを登録するには、次のようにします。

1. CSL SCI とクライアント間の対話を駆動するために必要な出口ルーチンを識別する。SCI は、IMSplex 内のすべての CSL マネージャーとそれらのクライアント・アプリケーションとの間の通信を扱います。
  - a. クライアントの SCI 通知出口を識別する。データベース接続は IMSplex 内の任意のアクティブ ODBM に経路指定される場合があるため、ODBM クライアントをすべての ODBM に登録する必要があります。SCI 通知出口は、新規 ODBM が IMSplex 内でアクティブになった時に駆動され、新規 ODBM に関する情報をクライアントに通知します。その後、クライアントは新規 ODBM に登録することができます。SCI 通知出口は、CSLSCREG 要求の NOTIFYEXIT パラメーターで識別されます。
  - b. CSLSCREG 要求の INPUTEXIT パラメーターで、クライアントの SCI 入力出口を識別する。ODBM クライアントは、IMSplex 内のどの ODBM から送信される ODBM ディレクティブも受信できなければなりません。SCI 入力出口は、クライアントに対するメッセージまたは要求がある時に駆動され、これによりクライアントはそれらを受信して処理することができます。
2. 手順 1 で収集した情報を使用して、SCI に対して CSLSCREG 要求を発行する。クライアントは、IMSplex 内の CSL SCI に登録してからでないと、ODBM などの CSL マネージャーに登録することができません。
3. SCI に対して CSLSCQRY 要求を出し、ODBM のどのインスタンスが IMSplex 内でアクティブか判別する。クライアントは、初めて ODBM に登録する前に、この要求を使用してすべてのアクティブ ODBM を手動で識別する必要があります。一度登録すれば、新規 ODBM がアクティブになった時に SCI 通知出口が自動的にクライアントに通知します。
4. 到達可能で、登録要求を受け付ける準備ができている IMSplex 内のすべての ODBM に対して、CSLDMREG 要求を発行する。

ODBM クライアントの CSL 登録を管理することについて詳しくは、一連の ODBM クライアント要求を参照してください。

関連概念:

161 ページの『ODBM クライアント要求の順序』

関連資料:

273 ページの『CSLSCREG: 登録要求』

177 ページの『CSLDMREG: ODBM クライアント登録要求』

## OM コマンド処理クライアントの登録

Operations Manager (OM) コマンド処理クライアントは、そのコマンドを OM に登録する必要がありますが、自動化操作プログラム (AOP) クライアントを OM に登録する必要はありません。

IMSplex 内で OM コマンド処理クライアントを登録するには、以下の手順を実行します。

1. CSL SCI とクライアント間の対話を駆動するために必要な出口ルーチンを識別する。SCI は、IMSplex 内のすべての CSL マネージャーとそれらのクライアント・アプリケーションとの間の通信を扱います。
  - a. クライアントの SCI 通知出口を識別する。データベース接続は IMSplex 内の任意のアクティブ OM に経路指定される場合があるため、OM コマンド処理クライアントをすべての OM に登録する必要があります。SCI 通知出口は、新規 OM が IMSplex 内でアクティブになった時に駆動され、新規 OM に関する情報をクライアントに通知します。その後、クライアントは新規 OM に登録することができます。SCI 通知出口は、CSLSCREG 要求の NOTIFYEXIT パラメーターで識別されます。
  - b. CSLSCREG 要求の INPUTEXIT パラメーターで、クライアントの SCI 入力出口を識別する。OM コマンド処理クライアントは、IMSplex 内のどの OM から送信される OM ディレクティブも受信できなければなりません。SCI 入力出口は、クライアントに対するメッセージまたは要求がある時に駆動され、これによりクライアントはそれらを受信して処理することができます。
2. 手順 1 で収集した情報を使用して、SCI に対して CSLSCREG 要求を発行する。クライアントは、IMSplex 内の CSL SCI に登録してからでないと、OM などの CSL マネージャーに登録することができません。
3. SCI に対して CSLSCQRY 要求を発行し、OM のどのインスタンスが IMSplex 内でアクティブか判別する。クライアントは、初めて OM に登録する前に、この要求を使用してすべてのアクティブ OM を手動で識別する必要があります。一度登録すれば、新規 OM がアクティブになった時に SCI 通知出口が自動的にクライアントに通知します。
4. CSLOMBLD 要求を発行して、OM に渡すコマンド・リストを作成する。
5. 到達可能で、登録要求を受け付ける準備ができている、IMSplex 内のすべての OM に対して CSLOMREG 要求を発行する。
6. CSLOMRDY 要求を発行して、クライアントが OM からのコマンドの処理を開始する準備ができていることを示す。

OM コマンド処理クライアントの CSL 登録を管理することについて詳しくは、一連の OM コマンド処理クライアント要求を参照してください。

関連資料:

183 ページの『CSL OM コマンド処理クライアントの要求』

## RM クライアントの登録

リソースを管理して、全 IMSplex プロセスにアクセスするように、RM クライアントを登録します。

以下の手順では、RM クライアントをまず IMSplex Structured Call Interface (SCI) に登録し、次に IMSplex 内のアクティブ RM に登録する方法を、説明します。

1. CSL SCI とクライアント間の対話を駆動するために必要な出口ルーチンを識別する。SCI は、IMSplex 内のすべての CSL マネージャーとそれらのクライアント・アプリケーションとの間の通信を扱います。
  - a. クライアントの SCI 通知出口を識別する。データベース接続は IMSplex 内の任意のアクティブ RM に経路指定される場合があるため、RM クライアントをすべての RM に登録する必要があります。SCI 通知出口は、新規 RM が IMSplex 内でアクティブになった時に駆動され、新規 RM に関する情報をクライアントに通知します。その後、クライアントは新規 RM に登録することができます。SCI 通知出口は、CSLSCREG 要求の NOTIFYEXIT パラメーターで識別されます。
  - b. CSLSCREG 要求の INPUTEXIT パラメーターで、クライアントの SCI 入力出口を識別する。RM クライアントは、IMSplex 内のどの RM から送信される RM ディレクティブも受信できなければなりません。SCI 入力出口は、クライアントに対するメッセージまたは要求がある時に駆動され、これによりクライアントはそれらを受信して処理することができます。
2. 手順 1 で収集した情報を使用して、SCI に対して CSLSCREG 要求を発行する。クライアントは、IMSplex 内の CSL SCI に登録してからでないと、RM などの CSL マネージャーに登録することができません。
3. SCI に対して CSLSCQRY 要求を発行し、RM のどのインスタンスが IMSplex 内でアクティブか判別する。クライアントは、初めて RM に登録する前に、この要求を使用してすべてのアクティブ RM を手動で識別する必要があります。一度登録すれば、新規 RM がアクティブになった時に SCI 通知出口が自動的にクライアントに通知します。
4. 到達可能で、登録要求を受け付ける準備ができていて、IMSplex 内のすべての RM に対して CSLRMREG 要求を発行する。グローバル・リソースを管理するには、リソース・タイプおよび関連する名前タイプを登録します。

## SCI 作動可能状態を有効にする方法

IMSplex メンバーがタイプ別に経路指定されたメッセージおよび要求を受信できるようにするには、CSLSCRDY 要求を使用します。

SCI については、「登録済み」と「作動可能」という 2 つの状態があります。CSLSCRDY 要求は、タイプで経路指定されたメッセージおよび要求を IMSplex メンバーが受信できるようにします。登録されているが、CSLSCRDY 要求を出していない IMSplex メンバーは、そのメンバーに明示的に送信されたメッセージおよび要求しか処理できません。

## CSL 要求をコーディングする順序

ほとんどの Common Service Layer (CSL) 要求は、特定の順序で発行する必要があります。

各種の CSL クライアントから要求を発行する順序について詳しくは、以下のトピックのそれぞれに記載されている表を参照してください。

- 158 ページの『ホストで実行される AOP クライアントが CSL OM と通信する方法』
- 158 ページの『ワークステーションで実行される AOP クライアントが CSL OM と通信する方法』
- 159 ページの『コマンド処理クライアントによる AOP コマンドの処理』
- 207 ページの『RM クライアント要求の順序』
- 161 ページの『ODBM クライアント要求の順序』

関連概念:

157 ページの『CSL OM 自動化操作プログラム・クライアント』

---

## 全 CSL コンポーネントに共通の要求

CSLZSHUT 要求と CSLZQRY の 2 つの要求は、すべての CSL コンポーネント (OM、RM、および SCI) で処理できる共通の要求です。

### CSLZQRY: 照会要求

IMSpIex 内で、CSL 内の 1 つ以上のコンポーネントについて統計を照会したい場合があります。CSLZQRY 要求を使って統計を入手する IMSpIex メンバー・プログラム、例えば、自動化オペレーション・プログラム (AOP) を作成することができます。IMSpIex のどのメンバーでも、CSLZQRY 要求を出すことができます。

#### CSLZQRY の構文

##### CSLZQRY DSECT の構文

CSLZQRY パラメーター・リストの長さ、および CSLZQRY 戻りコードおよび理由コードのための等価 (EQU) ステートメントをプログラムの中に組み込むには、FUNC=DSECT を使用します。

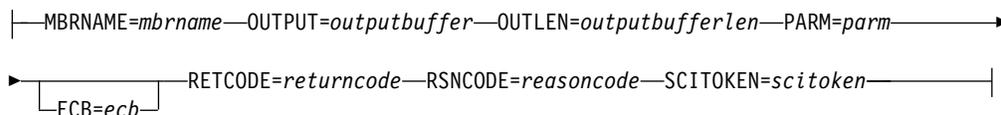
▶▶—CSLZQRY—FUNC=DSECT—▶▶

##### CSLZQRY STATS の構文

ODBM、OM、RM、または SCI に統計を要求するには、FUNC=STATS を使用します。CSLZQRY 要求から返される情報は、その ODBM、OM、RM、または SCI の STATS 出口に渡される情報と同じです。

▶▶—CSLZQRY—FUNC=STATS—| A |—▶▶

A:



## CSLZQRY パラメーター

**ECB=***symbol*

**ECB=(r2-r12)**

(オプション) - 非同期要求に使用される z/OS イベント制御ブロック (ECB) を指定します。要求が完了したとき、指定された ECB は通知されます。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、要求の起動側は、CSLZQRY から制御を受け取った後、この要求から戻されたデータ (RETCODE および RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を出す必要があります。

**MBRNAME=***symbol*

**MBRNAME=(r2-r12)**

(必須) -- 照会の送信先である 8 バイト CSL メンバー名のアドレスを指定する、4 バイトの入力パラメーター。

**OUTLEN=***symbol*

**OUTLEN=(r2-r12)**

(必須) -- 出力バッファの長さを受け取るために使用される、4 バイトの出力パラメーター。要求が戻ったとき、このワードには、OUTPUT= パラメーターで示されたバッファの長さが入っています。出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力長はゼロです。呼び出し元がこのストレージの処理を終っている場合、呼び出し元は、CSLSCBFR 要求を出してストレージを解放する必要があります。

**OUTPUT=***outputbuffer*

**OUTPUT=(r2-r12)**

(必須) -- CSLZQRY 要求から戻される可変長出力のアドレスを受け取る 4 バイト・フィールドを指定します。この出力には、CSLZQRY の結果が入っています。出力長は、OUTLEN= フィールドに入れて戻されます。出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力アドレスはゼロです。このバッファは、呼び出し元によって事前割り振りされません。呼び出し元がこのストレージの処理を終っている場合、呼び出し元は、CSLSCBFR 要求を出してストレージを解放する必要があります。

**PARM=***symbol*

**PARM=(r2-r12)**

(必須) -- CSLZQRY パラメーター・リストを指定します。パラメーター・リストの長さは、ZQRY\_PARMLN でパラメーター・リストの長さとして EQU 値と同じでなければなりません。

**RETCODE=***symbol*

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。こ

これは、ODBM、OM、RM、または SCI から返されます。 ODBM 戻りコードは、CSLDRR で定義されています。 OM 戻りコードは、CSLORR で定義されています。 RM 戻りコードは、CSLRRR で定義されています。 SCI 戻りコードは、CSLSRR で定義されています。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。これは、ODBM、OM、RM、または SCI から返されます。 ODBM 理由コードは、CSLDRR で定義されています。 OM 理由コードは、CSLORR で定義されています。 RM 理由コードは、CSLRRR で定義されています。 SCI 理由コードは、CSLSRR で定義されています。

**SCITOKEN=symbol**

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。 SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

## CSLZQRY の戻りコードと理由コード

以下の表は、CSLZQRY マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 31. CSLZQRY の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'xx000008'	X'00002050'	サービスの呼び出し元が無効なパラメーター・リストを渡そうとした。要求はリジェクトされます。「xx」は戻りコードを設定するコンポーネントを識別します。

関連概念:

119 ページの『CSL 要求の戻りコードおよび理由コードの解釈方法』

## CSLZSHUT: シャットダウン要求

CSLZSHUT は、IMSpIex 許可メンバーから 1 つ以上の CSL コンポーネントをシャットダウンするために使用できるプログラミング・インターフェースです。CSLZSHUT はメッセージとして送信されるため、メッセージの送信後、制御は、要求を出したプログラムに戻ります。

CSLZSHUT で、以下を終了させることができます。

- 単一の CSL マネージャー (ODBM、OM、RM、または SCI)
- 単一 z/OS イメージ上の CSL とその全コンポーネント
- 複数の z/OS イメージにまたがる IMSpIex 用の CSL とその全コンポーネント

CSLZSHUT 要求はメッセージとして送信され、そのため、要求の送信後、制御は、その要求を出したプログラムに戻ります。

単一の CSL コンポーネントをシャットダウンするには、シャットダウンしたいコンポーネントに対して CSLZSHUT FUNC=QUIESCE,SCOPE=CSLMEMBER メッセージを送信します。

単一 z/OS イメージ上の CSL とその全コンポーネントをシャットダウンするには、次のいずれかを行います。

- シャットダウンする CSL が含まれる z/OS イメージ上でアクティブな SCI に、CSLZSHUT FUNC=QUIESCE,SCOPE=CSLLOCAL メッセージを送信する。
- IMSplex 内でアクティブな任意の SCI に、CSLZSHUT FUNC=QUIESCE,SCOPE=CSLLOCAL,OSNAME=xxxx メッセージを送信する (ここで、xxxx は、シャットダウンする CSL がアクティブになっている z/OS イメージです)。SCI は、シャットダウンされる全 CSL コンポーネントに CSLZSHUT 要求を送信します。

IMSplex 全体にわたる CSL をシャットダウンするには、IMSplex 内でアクティブな任意の SCI に CSLZSHUT FUNC=QUIESCE,SCOPE=CSLPLEX メッセージを送信します。SCI は、IMSplex 内の全 CSL コンポーネントに CSLZSHUT 要求を送信します。

## CSLZSHUT の構文

CSLZSHUT パラメーター・リストの長さ、および CSLZSHUT 戻りコードおよび理由コードのための等価 (EQU) ステートメントをプログラムの中に組み込むには、FUNC=DSECT を使用します。

▶▶ CSLZSHUT — FUNC=DSECT —————▶▶

## CSLZSHUT QUIESCE 構文

CSL コンポーネントが通常にシャットダウンされるように要求するには、FUNC=QUIESCE を使用します。CSL コンポーネントは、そのコンポーネントが現在処理中のすべての作業が完了してから、シャットダウンされます。この要求の処理後、そのコンポーネントは新規作業を受け入れません。

▶▶ CSLZSHUT — FUNC=QUIESCE —| A |————▶▶

**A:**

|— SCITOKEN=scitokenaddress — SCOPE=CSLMEMBER — MBRNAME=mbrnameaddress —▶  
 |— SCOPE=CSLLOCAL — OSNAME=osnameaddress —▶  
 |— SCOPE=CSLPLEX —▶

▶▶ PARM=parmaddress — RETCODE=returncodeaddress — RSNCODE=reasoncodeaddress —▶▶

シャットダウンされているコンポーネントが SCI の場合は、現在その SCI に登録されている各 IMSplex メンバーは、SCI が終了されるまで登録解除されません。これは、イベント通知に影響するおそれがあります。このような IMSplex メンバー

は、その SCI がシャットダウンされたので、他の IMSplex メンバーと通信できません。「孤立した」メンバーのうちの 1 つ以上がシャットダウンされたか、失敗した場合、SCI がオンラインに戻るまで、他の IMSplex メンバーには、そのシャットダウンまたは失敗が通知されません。

シャットダウンまたは障害の通知は、メンバーの許可レベルにより異なります。終了する側のメンバーが非許可メンバーの場合、他のメンバーは、SCI の再始動時に通知を受けます。終了する側のメンバーが許可メンバーの場合、他の許可メンバー（オーファンの許可メンバーを含む）は、SCI が再始動する前に通知を受けます。

## CSLZSHUT パラメーター

**MBRNAME=***symbol* **MBRNAME=(r2-r12)**

(SCOPE=CSLMEMBER の場合は必須) - シャットダウン要求の送信先である 8 バイト CSL メンバー名を指定します。

**OSNAME=***symbol*

**OSNAME=(r2-r12)**

(SCOPE=CSLLOCAL の場合はオプション) - z/OS イメージ上で実行中の、シャットダウンする CSL の 8 バイトの名前を指定します。OSNAME パラメーターが指定され、しかも指定された z/OS イメージ上で SCI がアクティブでない場合は、コマンドは処理されません。

**PARM=***symbol*

**PARM=(r1-r12)**

(必須) -- CSLZSHUT パラメーター・リストを指定します。パラメーター・リストの長さは、ZSHUT\_PARMLN でパラメーター・リストの長さとして EQU 値と同じでなければなりません。

**RETCODE=***symbol*

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。SCI 戻りコードは、CSLSRR で定義されています。返される戻りコードについては、以下の表に説明があります。

**RSNCODE=***symbol*

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。SCI 理由コードは、CSLSRR で定義されています。返される戻りコードについては、以下の表に説明があります。

**SCITOKEN=***symbol*

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

**SCOPE=CSLMEMBER | CSLLOCAL | CSLPLEX**

(必須) -- CSL 終了の有効範囲を指定します。SCOPE パラメーターの有効な値は次のとおりです。

### CSLMEMBER

これは、この要求を受信中の CSL コンポーネントがそれ自身をシャットダウンすることを要求します。CSLMEMBER は、どの CSL コンポーネントにも処理できます。

### CSLLOCAL

これは、単一 z/OS イメージ上の CSL コンポーネントをシャットダウンすることを要求します。OSNAME パラメーターも指定された場合は、その特定の z/OS イメージ上の CSL コンポーネントがシャットダウンされます。OSNAME パラメーターが指定され、しかも指定された z/OS イメージ上で SCI がアクティブでない場合は、コマンドは処理されません。OSNAME パラメーターが指定されていない場合は、メッセージを受信中の SCI は、ローカル z/OS イメージ上の CSL をシャットダウンします。SCI だけが SCOPE=CSLLOCAL 要求を処理することができます。この要求が他の CSL コンポーネントに送られた場合、要求は無視されます。

### CSLPLEX

これは、IMSpIex 全体の中の CSL コンポーネントをシャットダウンすることを要求します。SCI だけが SCOPE=CSLPLEX 要求を処理することができます。この要求が他の CSL コンポーネントに送られた場合、要求は無視されます。

## CSLZSHUT の戻りコードと理由コード

以下の表は、CSLZSHUT マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 32. CSLZSHUT の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'xx000008'	X'00002050'	サービスの呼び出し元が無効なパラメーター・リストを渡そうとした。要求はリジェクトされます。「xx」は戻りコードを設定するコンポーネントを識別します。

関連概念:

119 ページの『CSL 要求の戻りコードおよび理由コードの解釈方法』

---

## 第 4 章 CSL 自動化操作プログラム要求

TSO SPOC などの AOP クライアントは、特定の CSL 要求を使用して、操作プログラムの一部を自動化することができます。これらの要求について、以下のトピックで詳しく説明します。

---

### CSLOMCMD: コマンド要求

ホスト上で実行中の AOP クライアント・アプリケーションが OM に対して要求を発行し、コマンドを送信するには、CSLOMCMD 要求を使用します。

OM API または REXX SPOC API を介して実行依頼されるコマンドは、アドレス・スペース ID (ASID) USERID を許可に使用します。

TSO セッションで実行中に OM API を使用するプログラムから実行依頼されるコマンドは、TSO USERID を許可に使用します。

メッセージ処理プログラム (MPP) 領域またはバッチ・メッセージ処理プログラム (BMP) 領域で実行中に OM API を使用するプログラムから実行依頼されるコマンドは、IMS MPP/BMP 従属領域 USERID を許可に使用します。この環境では、ユーザーのインストール済み環境が IMS セキュリティー環境構築出口ルーチン (DFSBSEX0) または OTMA/APPC SECURITY FULL を使用する (例えば、ユーザーのインストール済み環境が /SECURE OTMA/APPC FULL コマンドを発行する) 場合は、実際のトランザクション・ユーザー ID を許可に使用できます。

#### CSLOMCMD の構文

CSLOMCMD の構文は、自動化操作プログラム・クライアントが何を実行するかによって異なります。

##### DSECT 構文

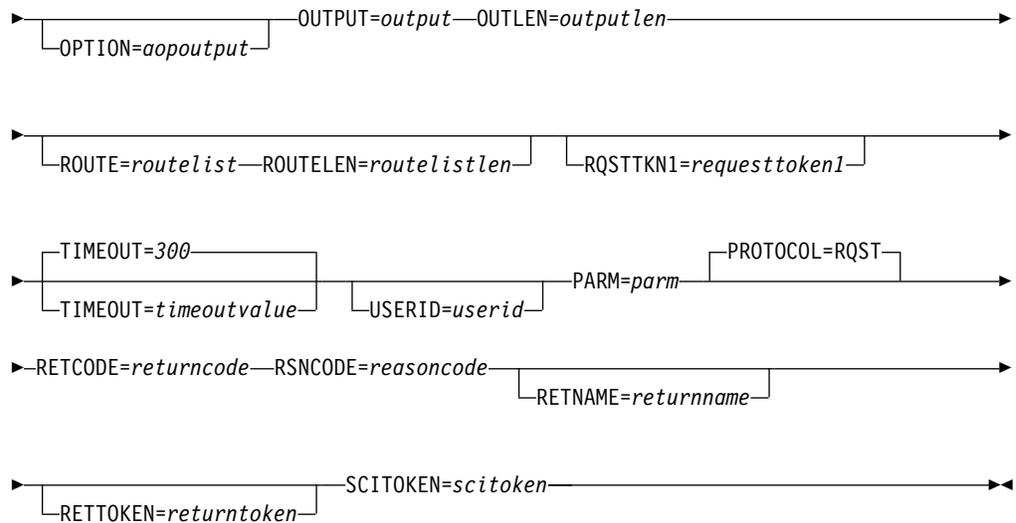
CSLOMCMD 要求の DSECT 機能は、CSLOMCMD パラメーター・リストの長さ、および戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶—CSLOMCMD—FUNC=DSECT—▶▶

##### 要求プロトコルの構文

OM 要求からの出力を待ちたい自動化クライアントの場合、この構文を使用します。

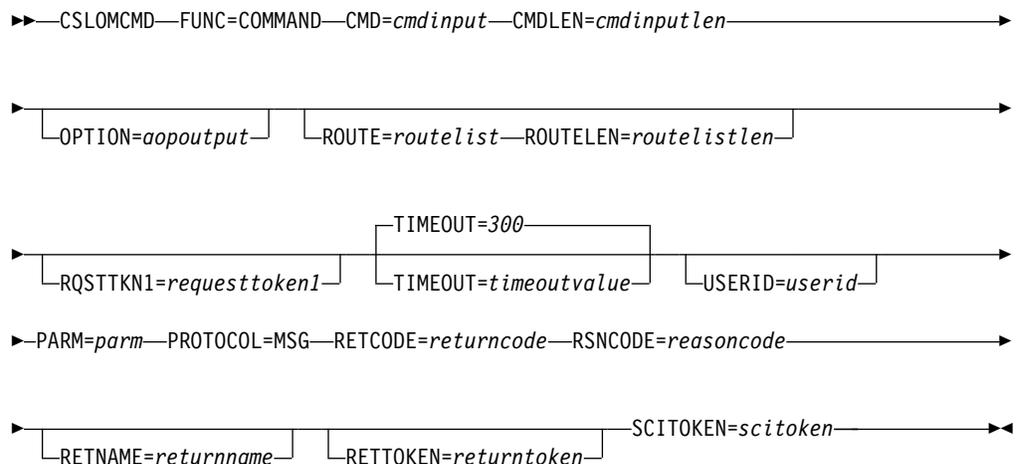
▶▶—CSLOMCMD—FUNC=COMMAND—ECB=ecb—CMD=cmdinput—CMDLEN=cmdinputlen—▶▶



応答は、要求の完了後、クライアントに戻されます。

メッセージ・プロトコルの構文

ユーザー出口からコマンド出力を受け取りたい自動化クライアントの場合、この構文を使用します。



応答は、SCI 入力出口を使ってクライアントに戻されます。クライアントは、応答を受け取るためには、SCI 登録要求 (CSLSCREG) に SCI 入力出口 (INPUTEXIT=) を指定しておく必要があります。

## CSLOMCMD のパラメーター

**CMD**=symbol

**CMD**=(r2-r12)

(必須) -- コマンド入力バッファを指定します。これは、OM API から指定できる任意の IMS コマンドが可能です。コマンドの先頭文字はコマンド認識文字 (例えば、/) である必要はありません。コマンド認識文字は OM 内でコマンド

経路指定を制御しません。どの IMSplex メンバーがコマンドを受け取るかは、ROUTE= キーワードで制御します。コマンド・ストリング中にコマンド認識文字が入力されていても、それは無視されます。コマンドの先頭文字は、それが A から Z (大文字でも、小文字でも) の文字でない場合、コマンド認識文字と見なされます。

**CMDLEN=***symbol*

**CMDLEN=**(*r2-r12*)

(必須) -- コマンド入力バッファの長さを指定します。

**ECB=***symbol*

**ECB=**(*r2-r12*)

(オプション) -- 非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定します。要求が完了したとき、指定された ECB は通知されず。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、マクロの発行元は、CSLONCMMD から制御を受け取った後、このマクロから戻されたデータ (RETCODE および RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を出す必要があります。

**OPTION=***aopoutput*

**OPTION=**(*r2-r12*)

(オプション) - OPTION を使用し、各コマンド処理クライアントからの出力の中にフォーマット ID (FID) を戻します。例えば、type-1 /DISPLAY コマンドを IMS コマンド処理クライアントに送信するときに、FID が各出力行に戻されるように要求することができます。FID は、AOI プログラムに対して、出力の行のマップの仕方を指示します。FID は、既存の AOI プログラムを OM AOI プログラムに変換するときに、役に立つことがあります。

OPTION がレジスターとして指定されている場合は、レジスターには、オプション値が入っている必要があります。例えば、AOPOUTPUT の値は 1 です。したがって、レジスターには、1 が入っている必要があります。

CSLONCMMD 要求には、AOPOUTPUT 値に等価のものが含まれます。

OPTION=AOPOUTPUT の場合、CSLONCMMD の出力用の DSECTS は IMS.SDFSMAC データ・セット内の DISPLAY マクロに記述されます。

**OUTLEN=***symbol*

**OUTLEN=**(*r2-r12*)

(RQST の場合、必須) -- CSLONCMMD 要求から戻される出力の長さを受け取る 4 バイト・フィールドを指定します。OUTLEN には、OUTPUT= パラメーターで示された出力の長さが入っています。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力長はゼロです。

**OUTPUT=***output*

**OUTPUT=**(*r2-r12*)

(RQST の場合、必須) -- CSLONCMMD 要求から戻される可変長出力のアドレスを受け取る 4 バイト・フィールドを指定します。この出力には、コマンド応答出力が入っています。出力長は、OUTLEN= フィールドに入れて戻されます。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力アドレスはゼロです。

出力バッファは、呼び出し元によって事前割り振りされることはありません。要求がこのワードを戻した場合、そこには、更新出力を含むバッファのアドレスが入っています。ストレージの使用が終了したとき、呼び出し元は、CSLSCBFR FUNC=RELEASE 要求を出すことにより、このストレージを解放する必要があります。出力の長さは、OUTLEN= フィールドに入れて戻されます。

**PARM=***symbol*

**PARM=(***r1-r12***)**

(必須) -- CSLOMCMMD パラメーター・リストを指定します。パラメーター・リストの長さは、OCMD\_PARMLN で定義されたパラメーター・リスト長 EQU 値と同じでなければなりません。

**PROTOCOL=**RQST

**PROTOCOL=**MSG

(オプション) -- 要求を OM に送信するための SCI プロトコルを指定します。

- RQST -- SCI 要求プロトコルを使ってコマンドを OM に送信する
- MSG -- SCI メッセージ・プロトコルを使ってコマンドを OM に送信する

**RETCODE=***symbol*

**RETCODE=(***r2-r12***)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。OM 戻りコードは、CSLORR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

戻りコードは、OM (CSLOMCMMD) または SCI (CSLSCMSG または CSLSCRQS) から返されます。ECB が指定されている場合、ECB が通知されるまで、RETCODE は無効です。戻りコードにはすべて、高位バイトに SCI、OM、または RM を表す SCI メンバー・タイプ標識が含まれます (X'01' は SCI、X'02' は OM、X'03' は RM を表します)。

**RETNAME=***symbol*

**RETNAME=(***r2-r12***)**

(オプション) -- OM 名を受け取る 8 バイト出力フィールドを指定します。これは、SCI の要求送信先のターゲット・アドレス・スペースの CSL メンバー名です。

**RETTOKEN=***symbol*

**RETTOKEN=(***r2-r12***)**

(オプション) -- 呼び出し元に戻される OM SCI トークンを受け取る 16 バイト出力フィールドを指定します。これは、要求の送信先のターゲット・アドレス・スペースの OM SCI トークンです。

**ROUTE=***symbol*

**ROUTE=(***r2-r12***)**

(オプション) - コマンドの送信先となる、IMSplex 内の OM クライアント (例えば、IMS 制御領域) を識別する経路リストを指定します。ROUTE を指定しないと、OM は、登録されていて、コマンド処理の準備ができていないすべてのクライアントに送信します。指定された経路リストが、指定されたコマンド用に登録されていない OM クライアントの SYSID で構成されている場合、コマン

ドは、クライアントがそのコマンド用に登録されていないことを示す戻りコードと理由コードを出して失敗します。例えば、ROUTE=IMS1 を指定した QUERY IMSPLEX コマンドが OM アドレス・スペースで処理される場合に、IMS 制御領域 IMS1 がこのコマンド用に登録されていなければ、コマンドは失敗します。

- 登録済みで、コマンド処理の準備ができていないすべてのコマンド処理クライアントにコマンドを明示的に送信するためには、ROUTE=\* を使用してください。
- READY 状態で MASTER 機能がある最初のコマンド処理クライアントにコマンドを送信するためには、ROUTE=% を使用してください。ROUTE=% の場合、OM は OM が選択する 1 つのコマンド処理クライアントのみにコマンドを送信します。

注: クライアント名のリストを分離するには、コンマを使用してください。

**ROUTELEN=***symbol*

**ROUTELEN=**(*r2-r12*)

(オプション) -- ROUTE= パラメーターで指定されたリストの長さを指定します。

**RQSTTKN1=***symbol*

**RQSTTKN1=**(*r2-r12*)

(オプション) -- 非同期処理の要求応答を要求に関連付けるために使用する 16 バイトのユーザー生成要求トークンを指定します。RQSTTKN1 は、A から Z、0 から 9、または &、<、および > を除く印刷可能文字 (大/小文字の区別なし) を使用することができます。OM は、XML 出力の中で、<rqsttkn1></rqsttkn1> タグでカプセル化された要求トークンを戻します。OM は、XML 出力をクライアントに戻す前に、無効なデータをすべてピリオド (.) に変換します。PROTOCOL=MSG 要求の場合、OM はこのトークンのアドレスも、OM ディレクティブ・パラメーター・リスト (CSLMDIR マクロによってマップされます) のフィールド ODIR\_CQRT1PTR に入れて戻します。このパラメーターは 16 バイトでなければならない、必要な場合、空白が埋め込まれます。

**RSNCODE=***symbol*

**RSNCODE=**(*r2-r12*)

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。OM 理由コードは、CSLORR で定義されています。SCI 理由コードは、CSLSRR で定義されています。

**SCITOKEN=***symbol*

**SCITOKEN=**(*r2-r12*)

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されます。

**TIMEOUT=***timeoutvalue*

**TIMEOUT=***symbol*

**TIMEOUT=**(*r2-r12*)

(オプション) -- 4 バイトのコマンド・タイムアウト値 (秒) を指定します。OM コマンド処理中に TIMEOUT 値に達したとき、すべてのクライアントがコ

マンドに応答し終わっていない場合、OM はコマンドを終了し、すべての使用可能な応答を戻します。指定された値が小さすぎる場合、不完全な応答が戻されます。TIMEOUT 値は、コマンドを処理しているクライアントが応答不能の場合でも、応答が必ず戻されるようにします。CMD キーワードが指定されていない場合、TIMEOUT キーワードは無視されます。コマンドが要求されたが、タイムアウト値が指定されていない場合、5 分のタイムアウト値が使用されま

す。  
TIMEOUT をシンボルとして指定する場合、そのシンボルは、タイムアウト値と等価にされた EQU シンボルでなければなりません。TIMEOUT を数値として指定する場合、数値はタイムアウト値でなければなりません。

**USERID=***symbol*

**USERID=**(*r2-r12*)

(オプション) - RACF® または同等のセキュリティー製品が使用する 8 バイトのユーザー ID を指定します。このパラメーターは、クライアント・アドレス・スペースがこの要求に対して許可されている場合にのみ使用します。ご使用のクライアントが無許可の場合、ユーザー ID が z/OS 制御ブロックから自動的に取得されます。このユーザー ID は、他のアドレス・スペースまたはリモート・クライアントに代わって OM 要求を出すことができる許可システム管理アドレス・スペースが使用するためのものです。この場合、クライアント・アドレス・スペースのユーザー ID は実際のクライアントのユーザー ID ではないので、このユーザー ID は OM に渡さなければなりません。このパラメーターは 8 バイトで、左寄せでなければならず、必要な場合、ブランクが埋め込まれます。

表 33. CSLOMCMD 戻りコードおよび理由コード :

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。

表 33. CSLOMCMMD 戻りコードおよび理由コード (続き):

戻りコード	理由コード	意味
X'02000004'	任意のコード	この戻りコードは警告を表します。要求の全部または一部が正常に完了した。追加情報は要求への応答で戻されます。
	X'00001000'	<p>コマンド応答情報のすべてが収集される前に、コマンドがタイムアウトになった。1 つ以上のクライアントが応答していない可能性があります。または、いずれかのクライアントがコマンド処理時間を長くする必要があった可能性もあります。TIMEOUT オプションを指定した場合、インターバルが、コマンドを処理できるだけ十分に長いことを確認してください。タイムアウトまでに収集された全コマンド応答情報が戻されます。</p> <p>SCI または RM などの CSL メンバーがローカルまたはリモート z/OS イメージ上でアクティブでなくて、要求を処理できないか、または応答を返せない場合も、この理由コードが戻されます。さらに詳しい情報入手するには、QUERY IMSPLEX を発行して、どの CSL メンバーが非アクティブであるかを判別してください。そのメンバーを再始動して、要求を再発行してください。</p> <p>INIT OLC または TERM OLC コマンドの後、この理由コードが戻された場合、QUERY MEMBER を発行して、オンライン変更に関与している IMS システムのオンライン変更状況を判別し、その状況に基づいて処置をとってください。</p>
	X'00001004'	CMD フィールドに指定されたコマンドを INPUT 出口がリジェクトした。コマンドは処理されませんでした。
	X'00001008'	クライアント (<cmderr> セクション内の対応する XML <mbr></mbr> タグで指定される) が、CMD フィールドで指定されたコマンドの ROUTE リストに指定された。クライアントがマスターではないため、コマンドはコマンド処理クライアントに経路指定されませんでした。
	x'00001014'	<p>コマンドは警告を出して完了した。戻りコードを確認してください。Operations Manager に、1 つ以上のクライアント・メンバーが戻りコード 4 を戻しました。その他のすべてのクライアントは、4 以下の戻りコードを戻しました。詳しくは、要求で戻された完了コードを参照してください。</p> <p>設計どおりに 1 つのコマンド処理クライアントがコマンドを正常に処理したものの、そのコマンドの明示的な経路指定先 (つまり、ユーザーが経路リストを指定した) または暗黙的な経路指定先 (つまり、ユーザーが経路リストを指定しなかったためコマンドの経路指定先が ALL になった) であるその他のすべてのコマンド処理クライアントが、戻りコード x'00000004' に対して理由コード x'00001000' (IRSN_NOTMSTR) を受信した場合、OM 全体としての戻りコードおよび理由コードは、それぞれ WARNING (02000004) と WARNING (00001014) になります。</p>

表 33. CSLOMCMD 戻りコードおよび理由コード (続き) :

戻りコード	理由コード	意味
X'02000008'	任意のコード	この戻りコードはパラメーター・エラーを表します。エラーのために、要求は処理されませんでした。
	X'00002000'	CMD フィールドに指定されたコマンドが無効。
	X'00002004'	CMD フィールドに指定されたコマンドに、そのコマンドでは無効なキーワードが入っていた。
	X'00002028'	CMD フィールドに指定されたコマンド・ストリングに、無効なキーワードが入っていた。
	X'0000202C'	CMD フィールドに指定されたコマンド・ストリングに、BPE が不明な位置パラメーターを検出した。
	X'00002030'	CMD フィールドに指定されたコマンド・ストリングに、サブリストが期待されているときに、等号の付いたキーワードが入っていた。例えば、 <code>keyword()</code> ではなく、 <code>keyword=</code> が指定されていました。
	X'00002034'	CMD フィールドに指定されたコマンド・ストリングに、不完全なキーワードまたはキーワード・パラメーターが入っていた。
	X'00002038'	CMD フィールドに指定されたコマンド・ストリングに、キーワードが欠落していた。
	X'0000203C'	CMD フィールドに指定されたコマンド・ストリングに、無効なキーワード・パラメーターが入っていた。
	X'00002040'	CMD フィールドに指定されたコマンド・ストリングに、重複したキーワードが入っていた。
	X'00002044'	コマンドに無効な構文が含まれている。エラー・ログの中で、構文エラーを含むテキストが <code>&lt;message&gt;&lt;/message&gt;</code> XML タグに入れられて戻される。
X'00002050'	サービスの呼び出し元が無効なパラメーター・リストを渡そうとした。要求はリジェクトされます。	
X'0200000C'	任意のコード	この戻りコードはリスト・エラーを表します。要求は処理されたかもしれないし、処理されなかったかもしれません。 <code>&lt;cmderr&gt;</code> のセクションと、 <code>&lt;cmdrspdata&gt;</code> セクションにリストされている、各コマンド処理クライアントの完了コードを参照してください。
	X'00003000'	コマンドは複数のクライアントに経路指定されました。少なくとも 1 つのクライアントが要求を正常に処理して、コマンド応答データまたは応答メッセージを SPOC に戻すことができました。詳しくは、CC フィールドの完了コードを参照してください。
	X'00003004'	コマンドは複数のクライアントに経路指定されました。どのクライアントも要求を正常に処理することができませんでした。コマンド応答データも、応答メッセージも戻されませんでした。
	X'00003008'	コマンドは複数のクライアントに経路指定されました。コマンドを処理したどのクライアントも、OM に戻りコードおよび理由コードを戻しませんでした。少なくとも 1 つのコマンド・クライアントがコマンド応答データまたは応答メッセージを戻しました。
	X'0000300C'	コマンドは複数のクライアントに経路指定されました。コマンドを処理したすべてのクライアントが、戻りコード 0 および理由コード 0 を OM に戻したわけではありません。また、少なくとも 1 つのクライアントが戻りコード 4 を戻しました。追加情報については、要求で戻された完了コードを参照してください。
	X'00003010'	経路リストにアクティブ・クライアントがない。メッセージ CSLN022I を参照してください。

表 33. CSLOMCMC 戻りコードおよび理由コード (続き):

戻りコード	理由コード	意味
X'02000010'	任意のコード	この戻りコードは環境関係のエラーを表します。要求は、現在の環境が原因で処理できませんでした。この条件は一時的の可能性がありません。
	X'00004000'	CMD フィールドに指定されたコマンドは、<cmderr> セクションの対応する <mbr></mbr> タグで示されたクライアントで処理できませんでした。そのクライアントは、コマンドを処理する準備がまだできていませんでした。
	X'00004004'	CMD フィールドに指定されたコマンドは、<cmderr> セクションの対応する XML <mbr></mbr> タグで示されたクライアントで処理できませんでした。そのクライアントはこのコマンドを登録していませんでした。
	X'00004008'	CMD フィールドに指定されたコマンドは、<cmderr> セクションの対応する XML <mbr></mbr> タグで示されたクライアントで処理できませんでした。そのクライアントは IMSplex 内でアクティブではありません。
	X'0000400C'	CMD フィールドに指定されたコマンドは、<cmderr> セクションの対応する XML <mbr></mbr> タグで示されたクライアントで処理できませんでした。そのクライアントは、無効な PADEF 文法でこのコマンドを登録していました。
	X'00004010'	CMD フィールドに指定されたコマンドを処理できませんでした。コマンドを発行したクライアントが許可されていません。エラー・ログ内の <cmdsecerr> セクションを調べて、このクライアントがなぜ許可されていないかを判別してください。
	X'00004020'	パラメーター・リストの、このバージョンが無効。

表 33. CSLOMCMD 戻りコードおよび理由コード (続き):

戻りコード	理由コード	意味
X'02000014'	任意のコード	この戻りコードはシステム・エラーを表します。内部エラーが発生しました。コマンドは処理されませんでした。
	X'00005000'	CMD フィールドに指定されたコマンドの処理用の CMD ブロックの割り振り中に OM 内部エラーが発生した。IBM ソフトウェア・サポートに連絡してください。
	X'00005004'	CMD フィールドに指定されたコマンドを処理するための CRSP ブロックの割り振り中に OM 内部エラーが発生した。IBM ソフトウェア・サポートに連絡してください。
	X'00005008'	CMD フィールドに指定されたコマンドを処理するためのコマンド入力バッファの割り振り中に OM 内部エラーが発生した。IBM ソフトウェア・サポートに連絡してください。
	X'0000500C'	CMD フィールドに指定されたコマンドの処理中に OM 内部エラーが発生した。IBM ソフトウェア・サポートに連絡してください。
	X'00005010'	CMD フィールドに指定されたコマンドを構文解析するための解析済み出力ブロック用のストレージを取得中に OM 内部エラーが発生した。IBM ソフトウェア・サポートに連絡してください。
	X'00005014'	CMD フィールドに指定されたコマンドの処理中に CMD ブロックをコマンド・インスタンス・ハッシュ・テーブルに追加していたとき、OM 内部エラーが発生した。IBM ソフトウェア・サポートに連絡してください。
	X'00005018'	CMD フィールドに指定されたコマンドの処理中にコマンド・インスタンス・ハッシュ・テーブル内の CMD ブロックにアクセスしていたとき、OM 内部エラーが発生した。IBM ソフトウェア・サポートに連絡してください。
	X'0000501C'	CMD フィールドに指定されたコマンドの処理中にコマンド・インスタンス・ハッシュ・テーブル内で CMD ブロックをスキャンしていたとき、OM 内部エラーが発生した。IBM ソフトウェア・サポートに連絡してください。
	X'00005020'	CMD フィールドに指定されたコマンドを処理中に OM 内部エラーが発生した。コマンドは、コマンド処理クライアントで処理されませんでした。コマンド処理クライアントのメンバー名についてエラー・ログの <cmderr> セクションを参照し、IBM ソフトウェア・サポートに連絡してください。
	X'00005024'	CMD フィールドに指定されたコマンドを処理中に OM 内部エラーが発生した。コマンドは、コマンド処理クライアントで処理されませんでした。コマンド処理クライアントのメンバー名について <cmderr> セクションを参照し、IBM ソフトウェア・サポートに連絡してください。
	X'00005028'	CMD フィールドに指定されたコマンドを構文解析中に OM 内部エラーが発生した。IBM ソフトウェア・サポートに連絡してください。

関連資料:

273 ページの『CSLSCREG: 登録要求』

291 ページの『第 9 章 CSL Operations Manager XML 出力』

## CSLOMI: API 要求

AOP クライアントは CSLOMI 要求を使用して、OM AOP クライアントとしての役割を果たす z/OS アドレス・スペースと通信します。これにより、OM 要求を発行し、OM に QUERY コマンドを送信することができます。

CSLOMI 要求を使用することにより、z/OS 自動化操作プログラム・クライアントは、OM に対して IMS コマンドを発行するか、OM から OM 固有情報を要求することができます。CSLOMI マクロ・インターフェースは、ワークステーションまたは他の z/OS アドレス・スペースから入力を受け取って、要求を OM に渡さな

なければならないシステム管理アドレス・スペースが使用するように設計されています。この場合には、ワークステーション・アプリケーションが入力ストリングを作成し、それを z/OS アドレス・スペースに渡します。z/OS アドレス・スペースは、INPUT= パラメーターで入力ストリングを OM に渡します。

OM API または REXX SPOC API を介して実行依頼されるコマンドは、アドレス・スペース ID (ASID) USERID を許可に使用します。

TSO セッションで実行中に OM API を使用するプログラムから実行依頼されるコマンドは、TSO USERID を許可に使用します。

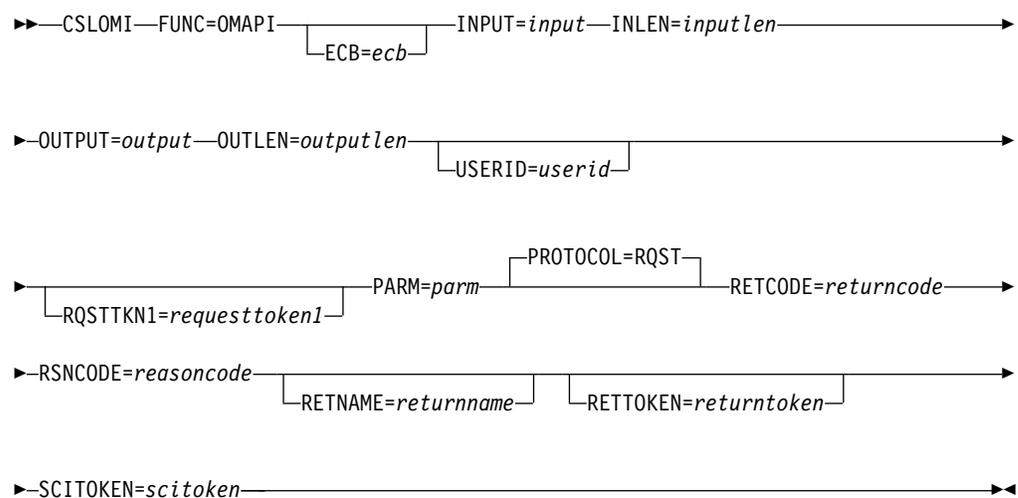
メッセージ処理プログラム (MPP) 領域またはバッチ・メッセージ処理プログラム (BMP) 領域で実行中に OM API を使用するプログラムから実行依頼されるコマンドは、IMS MPP/BMP 従属領域 USERID を許可に使用します。この環境では、ユーザーのインストール済み環境が IMS セキュリティー環境構築出口ルーチン (DFSBSEX0) または OTMA/APPC SECURITY FULL を使用する (例えば、ユーザーのインストール済み環境が /SECURE OTMA/APPC FULL コマンドを発行する) 場合は、実際のトランザクション・ユーザー ID を許可に使用できます。

## CSLOMI の構文

CSLOMI の構文は、自動化操作プログラム・クライアントがコマンド応答をどのように受信したいかによって異なります。クライアントが入力出口を持っておらず、コマンド出力を応答として受け取りたい場合、要求の構文を使用してください。クライアントが入力出口を持っていて、コマンド出力をメッセージとして受け取りたい場合、メッセージの構文を使用してください。

### CSLOMI 要求プロトコルの構文

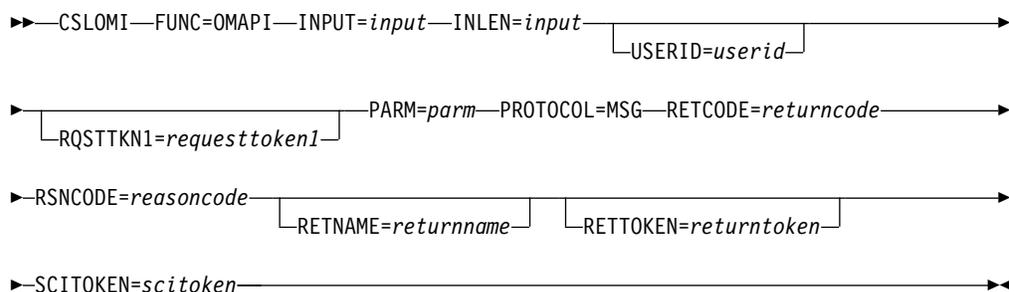
OM 要求からの出力を待ちたい自動化クライアントの場合、この構文を使用します。



制御がクライアントに戻されたら (ECB が指定されていない場合)、または ECB が通知されたら (ECB が指定されている場合)、応答はクライアントで使用可能です。

## CSLOMI メッセージ・プロトコルの構文

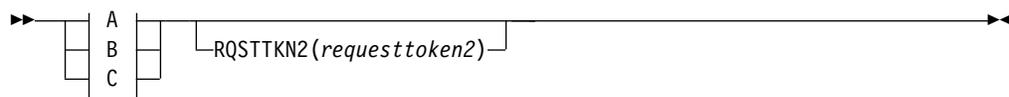
ユーザー出口からコマンド出力を受け取りたい自動化クライアントの場合、この構文を使用します。



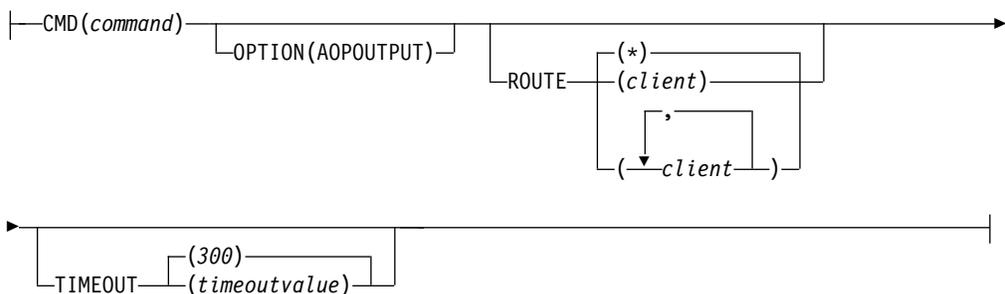
応答は、SCI 入力出口を使ってクライアントに戻されます。クライアントは、応答を受け取るためには、SCI 登録要求 (CSLSCREG) に SCI 入力出口 (INPUTEXIT=) を指定しておく必要があります。

## CSLOMI *Input=* パラメーターの構文

OM と直接には通信しない他のアプリケーションまたはワークステーションの場合、この構文を使用します。



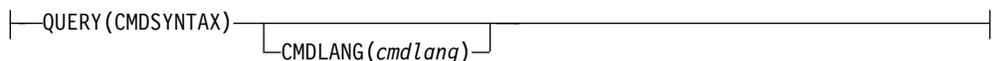
### A:



### B:



### C:



この構文は INPUT= パラメーター用です。アプリケーションがコマンドまたは照会を作成し、それを、OM と直接に通信する z/OS アドレス・スペースに渡します。

## CSLOMI 要求および CSLOMI メッセージのパラメーター

**ECB=***symbol*

**ECB=(r2-r12)**

(オプション) -- 非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定します。要求が完了したとき、指定された ECB は通知されません。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、マクロの発行元は、CSLOMI から制御を受け取った後、このマクロから戻されたデータ (RETCODE および RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を出す必要があります。

**INLEN=***symbol*

**INLEN=(r2-r12)**

(必須) -- 入力バッファの長さを指定します。

**INPUT=***symbol*

**INPUT=(r2-r12)**

(必須) -- 入力バッファのアドレスを指定します。

以下に、CSLOMI に渡される入力バッファの例を示します。入力バッファは文字フィールド MYINPUT であって、3 つのパラメーター、すなわち、コマンド・ストリング QRY TRAN SHOW(ALL)、タイムアウト値 360 秒、および 1 つのエレメントで構成される経路リスト IMSA を指定しています。

```
CSLOMI FUNC=OMAPI,INPUT=MYINPUT,INLEN=INPUTLEN
INPUTLEN DC      A(MYINPUTL)
MYINPUT DC      C'CMD (QRY TRAN SHOW(ALL) TIMEOUT(360) ROUTE(IMSA)
MYINPUTL EQU    *-MYINPUT
```

**OUTLEN=***symbol*

**OUTLEN=(r2-r12)**

(RQST の場合、必須) -- CSLOMI 要求から戻される出力の長さを受け取る 4 バイト・フィールドを指定します。OUTLEN には、OUTPUT= パラメーターで示された出力の長さが入っています。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力長はゼロです。

**OUTPUT=***symbol*

**OUTPUT=(r2-r12)**

(必須) -- CSLOMI 要求から戻される可変長出力を受け取るフィールドを指定します。この出力には、コマンド応答出力が入っています。出力長は、OUTLEN= フィールドに入れて戻されます。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力アドレスはゼロです。

出力バッファは、呼び出し元によって事前割り振りされることはありません。要求がこのワードを戻した場合、そこには、更新出力を含むバッファのアドレスが入っています。ストレージの使用が終了したとき、呼び出し元は、

CSLSCBFR FUNC=RELEASE 要求を出すことにより、このストレージを解放する必要があります。出力の長さは、OUTLEN= フィールドに入れて戻されます。

**PARM=symbol**

**PARM=(r2-r12)**

(必須) -- CSLOMI パラメーター・リストを指定します。パラメーター・リストの長さは、OI\_PARMLN でパラメーター・リストの長さとして定義された EQU 値と同じでなければなりません。

**PROTOCOL=RQST**

**PROTOCOL=MSG**

(オプション) -- 要求を OM に送信するための SCI プロトコルを指定します。

- RQST -- SCI 要求プロトコルを使ってコマンドを OM に送信する
- MSG -- SCI メッセージ・プロトコルを使ってコマンドを OM に送信する

**RETCODE=symbol**

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。OM 戻りコードは、CSLORR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

戻りコードは、OM (CSLOMI) または SCI (CSLSCMSG または CSLSCRQS) から返されます。ECB が指定されている場合、ECB が通知されるまで、RETCODE は無効です。戻りコードにはすべて、高位バイトに SCI、OM、または RM を表す SCI メンバー・タイプ標識が含まれます (X'01' は SCI、X'02' は OM、X'03' は RM を表します)。

**RETNAME=symbol**

**RETNAME=(r2-r12)**

(オプション) -- OM 名を受け取る 8 バイト出力フィールドを指定します。これは、SCI の要求送信先のターゲット・アドレス・スペースの CSL メンバー名です。

**RETTOKEN=symbol**

**RETTOKEN=(r2-r12)**

(オプション) -- 呼び出し元に戻される OM SCI トークンを受け取る 16 バイト出力フィールドを指定します。これは、要求の送信先のターゲット・アドレス・スペースの OM SCI トークンです。

**RQSTTKN1=symbol**

**RQSTTKN1=(r2-r12)**

(オプション) -- 非同期処理の要求応答を要求に関連付けるために使用する 16 バイトのユーザー生成要求トークンを指定します。RQSTTKN1 は、A から Z、0 から 9、または &、<、および > を除く印刷可能文字 (大/小文字の区別なし) を使用することができます。OM は、XML 出力の中で、<rqsttkn1></rqsttkn1> タグでカプセル化された要求トークンを戻します。OM は、XML 出力をクライアントに戻す前に、無効なデータをすべてピリオド (.) に変換します。PROTOCOL=MSG 要求の場合、OM はこのトークンのアドレスも、OM ディレクティブ・パラメーター・リスト (CSLOMDIR マクロに

よってマップされます) のフィールド ODIR\_CQRT1PTR に入れて戻します。  
このパラメーターは 16 バイトでなければならず、必要な場合、空白が埋め込まれます。

#### **RSNCODE=***symbol*

##### **RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。  
OM 理由コードは、CSLORR で定義されています。SCI 理由コードは、CSLSRR で定義されています。返される理由コードについては、以下の表に説明があります。

#### **SCITOKEN=***symbol*

##### **SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されます。

#### **USERID=***symbol*

##### **USERID=(r2-r12)**

(オプション) -- コマンドとキーワードの組み合わせのセキュリティ検査に使用する 8 バイトのユーザー ID を指定します。このユーザー ID は、クライアント・アドレス・スペースが許可呼び出し元である場合にのみ、使用されます。クライアント・アドレス・スペースが無許可の場合、ユーザー ID z/OS 制御ブロックから取得されます。このユーザー ID は、他のアドレス・スペースまたはリモート・クライアントに代わって OM 要求を出すことができる許可システム管理アドレス・スペースが使用するためのものです。この場合、クライアント・アドレス・スペースのユーザー ID は実際のクライアントのユーザー ID ではないので、このユーザー ID は OM に渡さなければなりません。このパラメーターは 8 バイトで、左寄せでなければならず、必要な場合、空白が埋め込まれます。

## **CSLOMI Input=** パラメーター

CSLOMI 入力オプションのパラメーターは、OM と直接には通信しないアプリケーションおよびワークステーション用です。

#### **CMD**(*command*)

(QUERY が指定されていない場合、必須) -- コマンド入力バッファを指定します。これは、OM API から指定できる任意の IMS コマンドが可能です。コマンドの先頭文字はコマンド認識文字 (例えば、/) である必要はありません。コマンド認識文字は OM 内でコマンド経路指定を制御しません。どの IMSplex メンバーがコマンドを受け取るかを制御するには、ROUTE キーワードを使用します。コマンド・ストリング中にコマンド認識文字が入力されていても、それは無視されます。コマンドの先頭文字は、それが A から Z (大文字でも、小文字でも) の文字でない場合、コマンド認識文字と見なされます。

#### **CMDLANG**(*cmdlang*)

要求に応じて戻される、IMS コマンド・テキストに使用される言語。この値を指定しない場合、デフォルト値は、OM 始動パラメーター CMDLANG= で OM システム用に設定された値です。現在、受け入れられる値は ENU (米国英語) だけです。無効な言語が指定されると、OM デフォルト言語のテキストが戻されます。

## OPTION(AOPOUTPUT)

(オプション、CMD() でのみ有効) - AOPOUTPUT オプションを指定して、コマンド処理クライアントからの出力の中にフォーマット ID (FID) を戻します。例えば、type-1 /DISPLAY コマンドを IMS コマンド処理クライアントに送信するときに、FID が各出力行に戻されるように要求することができます。FID は、AOI プログラムに対して、出力の行のマップの仕方を指示します。FID は、既存の AOI プログラムを OM AOI プログラムに変換するときに、役に立つことがあります。

## QUERY(querytype)

OM が実行する照会のタイプ。

### CMDCLIENTS

コマンド処理が OM に登録されている全クライアント (例えば、IMS 制御領域) のリストを戻すように OM に要求します。

クライアントのリストは、<cmdclients> </cmdclients> タグでカプセル化されて戻されます。querytype は、次のいずれか 1 つが可能です。

- <mbr name=membername>

メンバー名は、クライアント・アドレス・スペースの名前です。

- <typ> </typ>

メンバー・タイプは、クライアント・アドレス・スペースのタイプです。

- <styp> </styp>

メンバー・サブタイプは、クライアント・アドレス・スペースのサブタイプです。

- <vsn> </vsn>

メンバー・バージョンは、クライアント・アドレス・スペースのバージョンです。

- <jobname> </jobname>

クライアント・ジョブ名は、クライアント・アドレス・スペースのジョブ名または開始タスクです。

- </mbr>

### CMDSYNTAX

OM に登録されている選択済みコマンドのコマンド構文を表す XML のリストを戻すように OM に要求します。それに加えて、コマンド構文に関連付けられた変換可能テキストが戻されます。

コマンド構文 XML は、<cmdsyntax> </cmdsyntax> タグでカプセル化されて戻されます。コマンド構文 DTD は、<cmdddtd> </cmdddtd> タグでカプセル化されて戻されます。コマンド構文変換可能テキストは、<cmdtext> </cmdtext> タグでカプセル化されて戻されます。

## ROUTE(routelist)

(オプション) - コマンドの送信先となる、IMSplex 内の OM クライアント (例えば、IMS 制御領域) を識別する経路リストを指定します。リスト内で、クラ

クライアントはコマンドで分離されています。ROUTE を指定しないと、OM は、登録されていて、コマンド処理の準備ができていないすべてのクライアントに送信します。

- 登録済みで、コマンド処理の準備ができていないすべてのコマンド処理クライアントにコマンドを明示的に送信するためには、ROUTE(\*) を使用してください。
- READY 状態で MASTER 機能がある最初のコマンド処理クライアントにコマンドを送信するためには、ROUTE(%) を使用してください。ROUTE(%) の場合、OM は OM が選択する 1 つのコマンド処理クライアントのみにコマンドを送信します。

#### **RQSTTKN2(requesttoken2)**

(オプション) -- 非同期処理の要求応答を要求に関連付けるために使用する 16 バイトのユーザー生成要求トークンを指定します。RQSTTKN2 は、A から Z、0 から 9、または &、<、および > を除く印刷可能文字 (大/小文字の区別なし) を使用することができます。OM は、XML 出力の中で、`<rqsttkn2></rqsttkn2>` タグでカプセル化された要求トークンを戻します。OM は、XML 出力を発行元のクライアントに戻す前に、無効なデータをすべてピリオド (.) に変換します。PROTOCOL=MSG 要求の場合、OM はこのトークンのアドレスも、OM ディレクティブ・パラメーター・リスト (CSLOMDIR マクロによってマップされます) のフィールド ODIR\_CQRT2PTR に入れて戻します。

#### **TIMEOUT(timeoutvalue )**

(オプション) -- 4 バイトのコマンド・タイムアウト値 (秒) を指定します。コマンド処理中に、すべてのクライアントがコマンドに応答し終わる前に TIMEOUT 値に達した場合、OM はコマンドを終了し、すべての使用可能な応答を戻します。指定された値が小さすぎる場合、不完全な応答が戻されます。TIMEOUT 値は、コマンドを処理しているクライアントが応答不能の場合でも、応答が必ず戻されるようにします。CMD キーワードが指定されていない場合、TIMEOUT キーワードは無視されます。コマンドが要求されたが、タイムアウト値が指定されていない場合、5 分のタイムアウト値が使用されます。

### **CSLOMI の戻りコードと理由コード**

以下の表は、CSLOMI マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 34. CSLOMI の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'02000004'	任意のコード	この戻りコードは警告を表します。要求の全部または一部が正常に完了した。追加情報は要求への応答で戻されます。

表 34. CSLOMI の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'00001000'	コマンド応答情報のすべてが収集される前に、指定されたコマンドがタイムアウトになった。1 つ以上のクライアントが応答していない可能性があります。または、いずれかのクライアントがコマンド処理時間を長くする必要があったかもしれません。TIMEOUT 値が指定されている場合、その値が、コマンドが十分に処理できる時間であるかを確認してください。タイムアウトまでに収集された全コマンド応答情報が戻されます。
	X'00001004'	CMD フィールドに入っているコマンドを INPUT 出口がリジェクトした。コマンドは処理されませんでした。
	X'00001008'	クライアント (<cmderr> セクション内の対応する XML <mbr></mbr> タグで指定される) が、CMD フィールドで指定されたコマンドの ROUTE リストに指定された。クライアントがマスターではないため、コマンドはコマンド処理クライアントに経路指定されませんでした。
	X'00001010'	CMDLANG パラメーターで指定された言語でテキスト・ファイルがロードできなかった。デフォルト言語が使用されます。
	X'00001014'	<p>コマンドは警告を出して完了した。戻りコードを確認してください。Operations Manager に、1 つ以上のクライアント・メンバーが戻りコード 4 を戻しました。その他のすべてのクライアントは、4 以下の戻りコードを戻しました。</p> <p>設計どおりに 1 つのコマンド処理クライアントがコマンドを正常に処理したものの、そのコマンドの経路指定先であるその他のすべてのコマンド処理クライアントが、戻りコード X'00000004' に対して理由コード X'00001000' (IRSN_NOTMSTR) を受信した場合、OM 全体としての戻りコードおよび理由コードは、以下のようになります。</p> <ul style="list-style-type: none"> <li>• WARNING (X'02000004')</li> <li>• WARNING (X'00001014')</li> </ul> <p>詳しくは、要求で戻された完了コードを参照してください。</p>
X'02000008'	任意のコード	この戻りコードはパラメーター・エラーを表します。エラーのために、要求は処理されませんでした。
	X'00002000'	CMD フィールドに指定されたコマンドが無効。
	X'00002004'	CMD フィールドに指定された基本キーワードは、指定されたコマンドでは無効。
	X'00002028'	CMD フィールドに無効なキーワードが指定されていた。
	X'0000202C'	BPE が CMD フィールドのコマンドに不明な定位置パラメーターを検出した。

表 34. CSLOMI の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'00002030'	CMD フィールドのコマンドにサブリストが期待されている (KEYWORD()) ときに、等号を使ってキーワードが指定された (KEYWORD=)。
	X'00002034'	CMD フィールドのコマンドに不完全なキーワードまたはキーワード・パラメーターが指定されていた。
	X'00002038'	CMD フィールドのコマンドからキーワードが欠落している。
	X'0000203C'	コマンドに指定されたキーワード・パラメーターの値が無効だった。
	X'00002040'	CMD フィールドのコマンドに重複キーワードが指定されていた。
	X'00002044'	構文エラーを含むテキストが XML <message></message> タグに入れられて戻される。
	X'00002048'	複数のフィルターが指定されていた。
	X'00002050'	サービスの呼び出し元が無効なパラメーター・リストを渡そうとした。要求はリジェクトされます。
X'0200000C'	任意のコード	この戻りコードはリスト・エラーを表します。要求は処理されているかもしれないし、エラーのために処理されていないかもしれません。XML タグ <cmderr> のセクションと、XML タグ <cmdrspdata> にリストされている、各コマンド処理クライアントの完了コードを参照してください。
	X'00003000'	コマンドは複数のクライアントに経路指定されました。少なくとも 1 つのクライアントが要求を正常に処理して、コマンド応答データまたは応答メッセージを戻すことができました。詳しくは、要求で戻された完了コードを参照してください。
	X'00003004'	コマンドは複数のクライアントに経路指定されました。どのクライアントも要求を正常に処理することができませんでした。どのクライアントからも、コマンド応答データも応答メッセージも戻されませんでした。
	X'00003008'	コマンドは複数のクライアントに経路指定されました。コマンドを処理したどのクライアントも、OM に戻りコード 0 および理由コード 0 を戻しませんでした。少なくとも 1 つのコマンド・クライアントがコマンド応答データまたは応答メッセージを戻しました。
	X'0000300C'	コマンドは複数のクライアントに経路指定されました。コマンドを処理したすべてのクライアントが、戻りコード 0 および理由コード 0 を OM に戻したわけではありません。また、少なくとも 1 つのクライアントが戻りコード 4 を戻しました。追加情報については、要求で戻された完了コードを参照してください。
X'02000010'	任意のコード	この戻りコードは環境関係のエラーを表します。要求は、この時点では、現在の環境が原因で処理できませんでした。この条件は一時的の可能性があります。

表 34. CSLOMI の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'00004000'	CMD フィールドに入っていたコマンドは、<cmderr> セクションの対応する XML <mbr></mbr> タグで示されたクライアントで処理できませんでした。そのクライアントは、コマンドを処理する準備がまだできていませんでした。
	X'00004004'	CMD フィールドに入っていたコマンドは、<cmderr> セクションの対応する XML <mbr></mbr> タグで示されたクライアントで処理できませんでした。そのクライアントはこのコマンドを登録していませんでした。
	X'00004008'	CMD フィールドに入っていたコマンドは、<cmderr> セクションの対応する XML <mbr></mbr> タグで示されたクライアントで処理できませんでした。そのクライアントは IMSplex 内でアクティブではありません。
	X'0000400C'	CMD フィールドに入っていたコマンドは、<cmderr> セクションの対応する XML <mbr></mbr> タグで示されたクライアントで処理できませんでした。そのクライアントは、無効な PADEF 文法でこのコマンドを登録していました。
	X'00004010'	CMD フィールドに入っているコマンドを処理できませんでした。コマンドを発行したクライアントが許可されていません。XML ファイル内の <cmdsecerr> セクションを調べて、このクライアントがなぜ許可されていないかを判別してください。
	X'00004014'	データ・セット割り振りエラーが発生した。 CMDTEXT= DSN パラメーターで指定されたデータ・セットを割り振れませんでした。
	X'00004018'	データ・セット読み取りエラーが発生した。 CMDTEXT= DSN で指定されたデータ・セット内のメンバーを読み取れませんでした。メンバー名は、CSLOT に 3 文字の CMDLANG 値が連結された値です。
	X'00004020'	パラメーター・リスト・バージョンが無効。
X'02000014'	任意のコード	この戻りコードはシステム・エラーを表します。内部エラーが発生して、コマンドは処理されませんでした。
	X'00005000'	OM 内部エラーが発生した。ストレージ不足のために、OM は、CMD フィールドのコマンドを処理するための CMD ブロックを割り振ることができませんでした。
	X'00005004'	OM 内部エラーが発生した。ストレージ不足のために、OM は、CMD フィールドのコマンドを処理するための CRSP ブロックを割り振ることができませんでした。
	X'00005008'	OM 内部エラーが発生した。ストレージ不足のために、OM は、CMD フィールドのコマンドを処理するためのコマンド入力バッファを割り振ることができませんでした。

表 34. CSLOMI の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'0000500C'	OM 内部エラーが発生した。 OM は、CMD フィールドのコマンドを処理中に VERB ラッチを取得できませんでした。
	X'00005010'	OM 内部エラーが発生した。ストレージ不足のために、OM は、CMD フィールドのコマンドを解析するための解析出力ブロック用のストレージを取得できませんでした。
	X'00005014'	OM 内部エラーが発生した。 OM は、CMD フィールドのコマンドを処理中に、コマンド・インスタンス・ハッシュ・テーブルに CMD ブロックを追加できませんでした。
	X'00005018'	OM 内部エラーが発生した。 OM は、CMD フィールドのコマンドを処理中に、コマンド・インスタンス・ハッシュ・テーブルの中で CMD ブロックを見つけられませんでした。
	X'0000501C'	OM 内部エラーが発生した。 OM は、CMD フィールドのコマンドを処理中に、コマンド・インスタンス・ハッシュ・テーブルの中で CMD ブロックをスキャンできませんでした。
	X'00005020'	OM 内部エラーが発生した。 OM は、CMD フィールドのコマンドを処理中にシステム AWE を取得できませんでした。コマンドは、コマンド処理クライアントで処理されませんでした。コマンド処理クライアントのメンバー名については、XML ファイルの <cmderr> セクションを参照してください。
	X'00005024'	OM 内部エラーが発生した。 OM は、CMD フィールドのコマンドを処理中に、システム AWE をキューに入れることができませんでした。コマンドは、コマンド処理クライアントで処理されませんでした。コマンド処理クライアントのメンバー名については、XML ファイルの <cmderr> セクションを参照してください。
	X'00005028'	OM 内部エラーが発生した。 OM は、BPEPARSE 内部エラーのために、CMD フィールドに入っているコマンドを解析できませんでした。
	X'0000502C'	OM 内部エラーが発生した。コマンド出力ヘッダー割り振りが失敗しました。
	X'00005030'	OM 内部エラーが発生した。コマンド出力応答割り振りが失敗しました。
	X'00005034'	OM 内部エラーが発生した。 OUTPUT バッファ割り振りが失敗しました。
	X'00005038'	OM 内部エラーが発生した。 VERB ハッシュ・テーブル追加が失敗しました。
	X'0000503C'	OM 内部エラーが発生した。 CLNT ブロックが取得できませんでした。
	X'00005040'	OM 内部エラーが発生した。 CSLSCQRY 要求が失敗しました。

表 34. CSLOMI の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'00005044'	OM 内部エラーが発生した。OM は、コマンド文法のコピーを BPEPARSE サービスに渡すためのストレージを取得できませんでした。

関連資料:

273 ページの『CSLSCREG: 登録要求』

291 ページの『第 9 章 CSL Operations Manager XML 出力』

202 ページの『CSL OM ディレクティブ』

## CSLOMQR: 照会要求

CSLOMQR 要求により、ホストで実行されているどの AOP クライアントでも、OM 固有の情報を要求することができます。

### CSLOMQR の構文

CSLOMQR の構文は、自動化操作プログラム・クライアントが何を発行するかによって異なります。それぞれの構文例のパラメーターについての説明は、以下のセクションに記載されています。

#### DSECT 構文

CSLOMQR 要求の DSECT 機能は、CSLOMQR パラメーター・リストの長さ、および戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶—CSLOMQR—FUNC=DSECT—▶▶

#### 要求プロトコルの構文

OM 要求からの出力を待ちたい自動化クライアントの場合、この構文を使用します。

▶▶—CSLOMQR—FUNC=QUERY—OUTPUT=*output*—OUTLEN=*outputlen*—  
└—ECB=*ecb*—┘▶▶

▶▶—└—RQSTTKN1=*requesttoken1*—┘TYPE=└—CMDCLIENTS—  
└—CMDSYNTAX—  
└—CMDLANG=*cmdlang*—┘▶▶

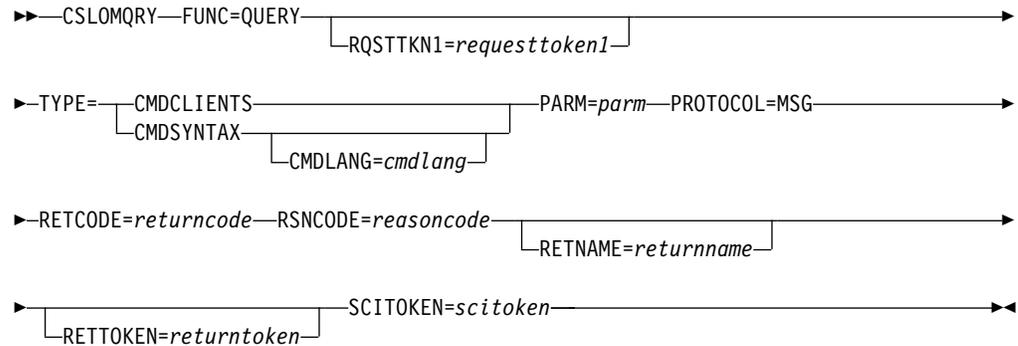
▶▶—└—PARM=*parm*—┘└—PROTOCOL=RQST—┘RETCODE=*returncode*—RSNCODE=*reasoncode*—▶▶

▶▶—└—RETNAME=*returnname*—┘└—RETTOKEN=*returntoken*—┘SCITOKEN=*scitoken*—▶▶

応答は、要求の完了後、クライアントに戻されます。

メッセージ・プロトコルの構文

OM 要求を処理するために OM にメッセージを送りたい自動化クライアントの場合、この構文を使用します。



応答は、SCI 入力出口を使ってクライアントに戻されます。クライアントは、応答を受け取るためには、SCI 登録要求 (CSLSCREG) に SCI 入力出口 (INPUTEXIT=) を指定しておく必要があります。

## CSLOMQRy のパラメーター

### **CMDLANG=cmdlang**

(オプション) - 要求に応じて戻される IMS コマンド・テキストに使用される言語。デフォルト値は、OM 始動パラメーター CMDLANG= で OM システム用に設定された値です。現在、受け入れられる値は ENU (米国英語) だけです。OM 内で無効な言語が指定されると、デフォルト言語が戻されます。

### **ECB=symbol**

#### **ECB=(r2-r12)**

(オプション) -- 非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定します。要求が完了したとき、指定された ECB は通知されず。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、マクロの発行元は、CSLOMQRy から制御を受け取った後、このマクロから戻されたデータ (RETCODE および RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を出す必要があります。

### **OUTLEN=symbol**

#### **OUTLEN=(r2-r12)**

(RQST の場合、必須) -- CSLOMQRy 要求から戻される出力の長さを受け取る 4 バイト・フィールドを指定します。OUTLEN には、OUTPUT= パラメーターで示された出力の長さが入っています。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力長はゼロです。

### **OUTPUT=output**

#### **OUTPUT=(r2-r12)**

(必須) -- CSLOMQRy 要求から戻される可変長出力を受け取るフィールドを指

定します。この出力には、コマンド応答出力が入っています。出力長は、OUTLEN= フィールドに入れて戻されます。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力アドレスはゼロです。

出力バッファは、呼び出し元によって事前割り振りされることはありません。要求がこのワードを戻した場合、そこには、更新出力を含むバッファのアドレスが入っています。ストレージの使用が終了したとき、呼び出し元は、CSLSCBFR FUNC=RELEASE 要求を出すことにより、このストレージを解放する必要があります。出力の長さは、OUTLEN= フィールドに入れて戻されます。

#### **PARM=symbol**

##### **PARM=(r2-r12)**

(必須) -- SCI にパラメーターを渡すために要求が使用するストレージのアドレスを指定する、4 バイトの入力パラメーター。パラメーター・リストの長さは、OQRY\_PARMLN で定義されたパラメーター・リスト長 EQU 値と同じでなければなりません。

#### **PROTOCOL=RQST**

##### **PROTOCOL=MSG**

(オプション) -- 要求を OM に送信するための SCI プロトコルを指定します。

- RQST -- SCI 要求プロトコルを使ってコマンドを OM に送信する
- MSG -- SCI メッセージ・プロトコルを使ってコマンドを OM に送信する

#### **RETCODE=symbol**

##### **RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。OM 戻りコードは、CSLORR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

戻りコードは、OM (CSLQMRY) または SCI (CSLSCMSG または CSLSCRQS) から返されます。ECB が指定されている場合、ECB が通知されるまで、RETCODE は無効です。戻りコードにはすべて、高位バイトに SCI、OM、または RM を表す SCI メンバー・タイプ標識が含まれます (X'01' は SCI、X'02' は OM、X'03' は RM を表します)。

#### **RETNAME=symbol**

##### **RETNAME=(r2-r12)**

(オプション) -- OM 名を受け取る 8 バイト出力フィールドを指定します。これは、SCI の要求送信先のターゲット・アドレス・スペースの CSL メンバー名です。

#### **RETTOKEN=symbol**

##### **RETTOKEN=(r2-r12)**

(オプション) -- 呼び出し元に戻される OM SCI トークンを受け取る 16 バイト出力フィールドを指定します。これは、要求の送信先のターゲット・アドレス・スペースの OM SCI トークンです。

#### **RQSTTKN1=symbol**

##### **RQSTTKN1=(r2-r12)**

(オプション) -- 非同期処理の要求応答を要求に関連付けるために使用する 16 バイトのユーザー生成要求トークンを指定します。RQSTTKN1 は、A から

Z、0 から 9、または &、<、および > を除く印刷可能文字 (大/小文字の区別なし) を使用することができます。OM は、XML 出力の中で、`<rqsttkn1></rqsttkn1>` タグでカプセル化された要求トークンを戻します。OM は、XML 出力をクライアントに戻す前に、無効なデータをすべてピリオド (.) に変換します。PROTOCOL=MSG 要求の場合、OM はこのトークンのアドレスも、OM ディレクティブ・パラメーター・リスト (CSLOMDIR マクロによってマップされます) のフィールド ODIR\_CQRT1PTR に入れて戻します。このパラメーターは 16 バイトでなければならず、必要な場合、空白が埋め込まれます。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。OM 理由コードは、CSLORR で定義されています。SCI 理由コードは、CSLSRR で定義されています。返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol**

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されます。

**TYPE=CMDCLIENTS**

**TYPE=CMDSYNTAX**

(必須) -- OM が実行する照会のタイプを指定する、4 バイトの入力パラメーター。

**CMDCLIENTS**

コマンド処理が OM に登録されている全クライアント (例えば、IMS 制御領域) のリストを戻すように OM に要求します。

クライアントは、`<cmdclients> </cmdclients>` タグでカプセル化されて戻されます。

- `<mbr name=membername>`

メンバー名は、クライアント・アドレス・スペースの名前です。

- `<typ> </typ>`

メンバー・タイプは、クライアント・アドレス・スペースのタイプです。

- `<styp> </styp>`

メンバー・サブタイプは、クライアント・アドレス・スペースのサブタイプです。

- `<vsn> </vsn>`

メンバー・バージョンは、クライアント・アドレス・スペースのバージョンです。

- `<jobname> </jobname>`

クライアント・ジョブ名は、クライアント・アドレス・スペースのジョブ名または開始タスクです。

• </mbr>

### CMSYNTAX

OM に登録されている選択済みコマンドのコマンド構文を表す XML のリストを戻すように OM に要求します。それに加えて、コマンド構文に関連付けられた変換可能テキストが戻されます。

コマンド構文 XML は、<cmdsyntax> </cmdsyntax> タグでカプセル化されて戻されます。コマンド構文 DTD は、<cmdddtd> </cmdddtd> タグでカプセル化されて戻されます。コマンド構文変換可能テキストは、<cmdtext> </cmdtext> タグでカプセル化されて戻されます。

CSLQMRY QUERY TYPE(CMSYNTAX) 要求の結果として戻されるコマンド構文と変換可能テキストには、タイプ-2 コマンドの情報が含まれません。

### CSLQMRY の戻りコードと理由コード

以下の表は、CSLQMRY 要求で返される、CSLQMRY 要求に固有の戻りコードと理由コードの組み合わせの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 35. CSLQMRY の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'02000004'	任意のコード	この戻りコードは警告を表します。要求の全部または一部が正常に完了した。追加情報は要求への応答で戻されます。
	X'00001010'	CMDLANG パラメーターで指定された言語でテキスト・ファイルがロードできなかった。デフォルト言語が使用されます。
X'02000008'	X'00002050'	サービスの呼び出し元が無効なパラメーター・リストを渡そうとした。要求はリジェクトされます。
X'02000010'	任意のコード	この戻りコードは環境関係のエラーを表します。要求は、この時点では、現在の環境が原因で処理できませんでした。この条件は一時的の可能性ががあります。
	X'00004014'	データ・セット割り振りエラーが発生した。OM 初期設定 PROCLIB メンバー (CSLOIxxx) の中の CMDTEXTDSN= パラメーターで指定されたデータ・セットを割り振れませんでした。
	X'00004018'	データ・セット読み取りエラーが発生した。OM 初期設定 PROCLIB メンバー (CSLOIxxx) の中の CMDTEXTDSN= パラメーターで指定されたデータ・セットの中のメンバーを読み取れませんでした。メンバー名は、'CSLOT' に 3 文字の CMDLANG 値が連結された値です。
	X'00004020'	パラメーター・リスト・バージョンが無効。

関連資料:

- 273 ページの『CSLSCREG: 登録要求』
- 291 ページの『第 9 章 CSL Operations Manager XML 出力』
- 202 ページの『CSL OM ディレクティブ』
- 295 ページの『CSLQMRY 出力』

---

## CSL OM 自動化操作プログラム・クライアント

OM は、オペレーター・アクションを自動化するアプリケーション・プログラムの API インターフェースを提供します。これらのプログラムは、自動化操作プログラム (AOP) として知られています。AOP を使用することで、OM への OM API 要求に埋め込まれたコマンドを発行することができます。

OM は、アプリケーション・プログラム用の、オペレーター・アクションを自動化するアプリケーション・プログラミング・インターフェース (API) を提供します。これらのプログラムは、自動化操作プログラム (AOP) と呼ばれます。AOP は、OM への OM API 要求に埋め込まれるコマンドを発行します。コマンドへの応答は、XML タグに埋め込まれて AOP に戻されます。

OM を使用して、ユーザー自身の製品またはサービスについて、IMSplex でのコマンドおよびコマンド応答を管理したい場合は、以下のような各 AOP クライアントを使用できます。

- IMS 提供の AOP クライアントである、TSO 単一制御点 (SPOC)。これは、ホスト上で実行されます。TSO SPOC では、自動化操作プログラムは、IMSplex にコマンドを出して、そのコマンドに対する応答を対話式に受け取ります。
- ワークステーション上で実行される AOP クライアント (ワークステーション SPOC と呼ばれます)。
- IMS などのコマンド処理クライアント。

OM クライアントは、OM との通信に OM 要求を使います。OM クライアントは、SCI に登録されなければ、OM 要求を出すことはできません。

AOP は、アセンブラーまたは REXX で作成することができます。アセンブラー・アプリケーションは、要求を OM API に出します。REXX アプリケーションは、REXX ホスト・コマンドを出して OM と通信します。

IMS は REXX SPOC API を提供しますが、これは、SPOC アプリケーションとの REXX プログラム・インターフェースです。既存の REXX アプリケーションは、この REXX SPOC API を使って OM と対話することができます。

関連概念:

- 255 ページの『CSL SCI 要求』
- 125 ページの『CSL 要求をコーディングする順序』

関連資料:

- 291 ページの『第 9 章 CSL Operations Manager XML 出力』

## ホストで実行される AOP クライアントが CSL OM と通信する方法

ホストで実行される自動化操作プログラム (AOP) クライアントは、Operations Manager (OM) と直接通信することができます。z/OS AOP は、SCI に登録された後で、OM コマンド (CSLSCMD) または照会 (CSLSCQRY) 要求を発行することができます。z/OS AOP は、終了の準備ができたなら、CSLSCDRG マクロを使って SCI から登録を解除する必要があります。各要求は、OM または SCI へ直接送信することができます。

以下の表は、ホストで実行される AOP から発行される要求のシーケンスの一覧です。要求の目的も併記します。

表 36. ホストで実行される AOP OM クライアントの場合の要求のシーケンス

要求	目的
CSLSCREG	SCI に登録する。これにより、クライアントは、SCI を介して OM に OM 要求を送信できます。
CSLSCRDY	SCI に対して OM クライアントを作動可能にする。これにより、メッセージは、クライアント・タイプ別にクライアントへ経路指定されます。
CSLSCxxx	OM 要求 (CSLSCMD、CSLSCQRY) を発行して、コマンドを OM に送信する。
CSLSCBFR	要求によって出力バッファが戻された場合、その出力バッファを解放する。
CSLSCQSC	SCI に対して静止する。
CSLSCDRG	SCI への登録を抹消します。

注: ホストで実行される AOP に必須ではありませんが、TYPE で経路指定されるメッセージを受け取りたいクライアントは、CSLSCRDY および CSLSCQSC を使用するようお勧めします。

OM クライアントは、OM サービスおよび OM リソースにアクセスするには、OM 要求を使います。OM サービスを要求するには、クライアントからいくつかの SCI 要求と OM 要求を出さなければなりません。一部の要求は、表 36 に示したように、特定の順序で出さなければなりません。そのほかの要求は、クライアントの処理要件に基づいて、複数回、どのような順序でも出すことができます。

## ワークステーションで実行される AOP クライアントが CSL OM と通信する方法

ワークステーション自動化操作プログラム (AOP) クライアントは、Operations Manager (OM) とは直接通信できません。ワークステーション AOP クライアントは、OM とではなく、OM AOP クライアントとしての役割を果たす z/OS アドレス・スペースと通信する必要があります。

z/OS アドレス・スペースは、CSLSCMD または CSLSCQRY 要求を出す代わりに CSLSOMI を出します。その CSLSOMI は、ワークステーションから受け取った、事前に作成されたストリングを OM に渡します。例えば、ワークステーションが IMSplex 内にいくつかのコマンド処理クライアントがあるかを照会する場合、

z/OS アドレス・スペースにストリング QUERY(CMDCLIENTS) を送信すると、z/OS アドレス・スペースが CSLOMI を使用して、コマンドを処理する OM に照会を送信することになります。

IMSpIex に QRY TRAN コマンドを出したい場合、ワークステーションは、z/OS アドレス・スペースに以下のストリングを送ることができます。

CMD(QUERY TRAN NAME) ROUTE(IMS) TIMEOUT(10) RQSTTKN2(QTRANCMD)

すると、z/OS アドレス・スペースは、CSLOMI を使用して OM にストリングを送り、コマンド処理してもらいます。z/OS アドレス・スペースは、OM で正しい許可処理が行われるように、ワークステーション・アプリケーションに関連付けられたユーザー ID を渡さなければなりません。

以下の表で、z/OS で実行され、ワークステーション AOP の代わりに OM と通信する、プロキシ AOP クライアントが発行する要求のシーケンスを説明します。要求の目的も併記します。

表 37. ワークステーションで実行される AOP の場合の要求のシーケンス

要求	目的
CSLSCREG	SCI に登録する。これにより、クライアントは、SCI を介して OM に OM 要求を送信できます。
CSLSCRDY	SCI に対して OM クライアントを作動可能にする。これにより、メッセージは、クライアント・タイプ別にクライアントへ経路指定されます。
CSLLOMI	OM 要求 (CMD(), QUERY()) を発行して、コマンドを OM に送信する。
CSLSCBFR	要求によって出力バッファが戻された場合、その出力バッファを解放する。
CSLSCQSC	SCI に対して静止する。
CSLSCDRG	SCI への登録を抹消します。

注: ワークステーションで実行される AOP に必須ではありませんが、TYPE で経路指定されるメッセージを受け取りたいクライアントは、CSLSCRDY および CSLSCQSC を使用するようにお勧めします。

## コマンド処理クライアントによる AOP コマンドの処理

IMS 制御領域などのコマンド処理クライアントは、自動化操作プログラム (AOP) によって入力されたコマンドを受け付けて処理するためのコマンド・プロセッサを提供するシステムです。

コマンド処理クライアントは、SCI への登録のほかに、OM にも登録しなければなりません。コマンド処理クライアントは、処理できるコマンドのリストを OM に渡すことにより、OM に登録します。

コマンド登録が正常に完了したら、クライアントは、コマンド処理の準備ができたことを OM に知らせる必要があります。

AOP コマンドは IMSpIex 内の任意のアクティブ OM を介して経路指定できるため、コマンド処理クライアントは、自身のコマンド・リストおよび作動可能状況をすべてのアクティブ OM に登録しておく必要があります。IMSpIex 内のすべての

OM に登録することにより、AOP コマンドを経路指定する OM にかかわらず、コマンド処理クライアントを対象とする任意の AOP コマンドが確実に正しく経路指定されることになります。

AOP クライアントと同じように、コマンド処理クライアントは、特定の順序で要求を出さなければなりません。この順序と要求の目的を、表 38 に示します。

表 38. コマンド処理クライアントの要求のシーケンス

要求	目的
CSLSCREG	SCI に登録する。これにより、クライアントは、SCI を介して OM に OM 要求を送信できます。
CSLSCRDY	SCI に対して OM クライアントを作動可能にする。これにより、メッセージは、クライアント・タイプ別にクライアントへ経路指定されます。
CSLLOMREG	コマンド・リストを OM に登録する。
CSLLOMRDY	OM クライアントを OM に対して作動可能にする。これで、クライアントはコマンド処理の準備ができました。
CSLLOMRSP	OM からのコマンドを受け取って処理した後、コマンド応答出力を OM に送り返す。
CSLLOMDRG	OM から登録解除する。クライアントは、この後、コマンドを処理しません。
CSLSCQSC	SCI に対して静止する。
CSLSCDRG	SCI への登録を抹消します。

## CSL OM XML 出力の解釈

OM API を介して戻されるコマンド応答は、XML タグに埋め込まれています。XML 出力は、CSLOMI、CSLOMCMMD、および CSLOMQRY 要求に対する応答として生成されます。OPTION=AOPOUTPUT の場合、CSLOMCMMD および CSLOMI の出力用の DSECTS は、IMS.SDFSMAC データ・セット内の DISPLAY マクロに記述されます。

例えば、CSLOMI 要求の場合、QUERY パラメーターにより、OM に登録されているすべてのクライアントについて照会することができます。クライアントは、`<cmdclients></cmdclients>` タグに埋め込まれて戻されます。

XML タグのリストと各タグの説明は、297 ページの『CSL OM 応答として返される XML タグ』にあります。

---

## 第 5 章 CSL ODBM クライアントの作成

独自の ODBM クライアントを作成して Open Database Manager (ODBM) に登録し、そのクライアントが、IMSplex 内の DBCTL システムおよび DB/TM システムの IMS DB が管理するデータベースに、CSLDMI インターフェースを使用して DL/I 呼び出しを渡すようにすることができます。

IMS Universal ドライバーの TCP/IP 接続の管理サービスおよびルーティング・サービスを提供する IMS Connect は、ODBM クライアントの一例です。IMS Connect および IMS Universal ドライバーは IMS に付属しているため、この 2 つを ODBM で使用するための CSL ODBM 要求をコーディングする必要はありません。

その他の ODBM クライアントは、IMS ODBM API を介して IMS データベースにアクセスするアプリケーション・プログラムを実行する、WebSphere® Application Server for z/OS や Db2® for z/OS などのアプリケーション・サーバーにできます。

ODBM クライアントを作成するには、ODBM で提供される一連のクライアント要求を使用します。これらの要求を使用して、ODBM クライアント (アセンブラー・プログラミング言語で作成された z/OS アプリケーション・プログラム) は、IMS DBCTL または DB/TM 環境用に構成された IMS DB システムが管理する IMSplex 内の IMS データベースにアクセスすることができます。ODBM クライアントの一例は、IMS Connect です。

ODBM クライアントは、ODBM に登録するための CSLDMI 要求をサブミットし、IMS データベースと対話して、ローカルまたはグローバル・トランザクションの同期点処理を管理します。

ODBM CSLDMI API は、IMS ODBA インターフェースを使用して IMS DB と通信するため、ODBA インターフェースがサポートする DL/I 呼び出しのみをサポートします。ODBM クライアントは、CSLDMI API の ODBMCI 機能を使用して、DL/I 呼び出しを IMS DB に渡します。

---

### ODBM クライアント要求の順序

例えば ODBM との通信を有効または無効にする場合など、ODBM から Open Database Manager (ODBM) への要求のなかには、特定の順序で発行しなければならないものがあります。

ODBM クライアントは、SCI 要求および ODBM 要求を発行して、ODBM サービスを要求します。一部の要求は、特定の順序で出さなければなりません。そのほかの要求は、クライアントの処理要件に基づいて、複数回、どのような順序でも出すことができます。

ODBM クライアントが ODBM 要求を発行できるようになるためには、IMSplex 内の SCI およびすべてのアクティブ ODBM に登録する必要があります。

以下の表に、ODBM クライアントが発行する要求の基本的な順序を記載します。CSLSCREG、CSLSCRDY、および CSLDMREG 要求は、記載されている順序で発行する必要があります。CSLSCBFR、CSLDMDRG、および CSLSCDRG 要求についても、記載されている順序で発行しなければなりません。CSLDMI 要求の順序は、ODBM クライアントおよび同期点処理の要件によって異なります。

表 39. ODBM クライアントの要求の順序

要求	目的
CSLSCREG	SCI に登録する。これにより、クライアントは SCI を介して ODBM に ODBM 要求を送信できるようになります。
CSLSCRDY	SCI に対して ODBM クライアントを作動可能にする。これにより、メッセージは、クライアント・タイプ別にクライアントへ経路指定されます。
CSLDMREG	クライアントを ODBM に登録して、ODBM との通信を可能にする。
CSLDMI FUNC=ODBMCI DLIFUNC	PSB の割り振りおよび割り振り解除を行います。ODBM を介して、DL/I 呼び出しを IMS DB に渡します。
CSLDMI FUNC=COMMIT	ローカル作業単位の場合、単一の APSB 呼び出しに関連付けられた更新をコミットする。
CSLDMI FUNC=BACKOUT	ローカル作業単位の場合、単一の APSB 呼び出しに関連付けられた更新をバックアウトする。
CSLDMI FUNC=READYSYNCP	グローバル作業単位の場合、複数の APSB 呼び出しのうちのいずれかを同期点処理用に準備する。
CSLSCBFR	要求によって出力バッファが戻された場合、その出力バッファを解放する。
CSLDMDRG	ODBM からクライアントを登録解除して、ODBM との通信を終了させる。
CSLSCQSC	ODBM クライアントを SCI に対して静止する。SCI は、クライアント・タイプ別にクライアントへ経路指定しなくなります。
CSLSCDRG	SCI への登録を抹消します。

関連タスク:

122 ページの『ODBM クライアントの登録』

関連資料:

177 ページの『CSLDMREG: ODBM クライアント登録要求』

## CSL ODBM クライアント要求

ODBM クライアントは、ODBM への登録要求をサブミットし、IMS データベースと対話して、データベース更新をコミットまたはバックアウトします。

## CSLDMDRG: ODBM クライアント登録解除要求

CSLDMDRG 要求は、ODBM での ODBM クライアントの登録を解除します。

ODBM クライアントが ODBM を介した ODBM 呼び出しの送信を終了すると、その ODBM クライアントは CSLDMDRG 要求を発行します。登録解除要求により、ODBM がその ODBM クライアント用に保管している内部制御ブロックがクリーンアップされます。

### CSLDMDRG DSECT の構文

CSLDMDRG パラメーター・リストの長さ、および戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込むには、CSLDMDRG 要求の DSECT 機能を使用します。

```
▶▶—CSLDMDRG—FUNC=DSECT—▶▶
```

### CSLDMDRG 要求プロトコルの構文

```
▶▶—CSLDMDRG—FUNC=DEREGISTER—ECB=ecb—ODBMNAME=odbmname—PARAM=parm—▶▶
```

```
▶—RETCODE=returncode—RSNCODE=reasoncode—SCITOKEN=scitoken—▶
```

### CSLDMDRG パラメーター

CSLDMDRG パラメーターは、ODBM での登録解除に必要な ODBM 値を指定します。

アドレスは、シンボルとして指定することも、2 から 12 までのレジスターとして指定することもできます。

CSLDMDRG 要求には、以下のパラメーターが含まれます。

**ODBMNAME**=*symbol*

**ODBMNAME**=(*r2-r12*)

(必須) - コマンド登録解除要求の送信先とする 8 バイトの ODBM 名を指定します。

ODBMNAME の値をシンボルとして指定する場合、そのシンボルは ODBM フィールドのラベルでなければなりません。ODBMNAME の値をレジスターとして指定する場合、そのレジスターには ODBM 名フィールドのアドレスが含まれていなければなりません。

**PARAM**=*symbol*

**PARAM**=(*r1-r12*)

(必須) - CSLDMDRG パラメーター・リストを指定します。パラメーター・リストの長さは、DDRG\_PARMLN でパラメーター・リストの長さとして定義された EQU 値と同じでなければなりません。

PARAM の値をシンボルとして指定する場合、そのシンボルは、このパラメーター・リスト・ストレージの先頭を示さなければなりません。PARAM の値をレジスターとして指定する場合、そのレジスターには、このパラメーター・リストのアドレスが含まれていなければなりません。

**RETCODE**=symbol

**RETCODE**=(r2-r12)

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。ODBM 戻りコードは、CSLDRR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

戻りコードは、ODBM (CSLDMDRG) または SCI (CSLSCMSG または CSLSCRQS) から返されます。ECB が指定されている場合、ECB が通知されるまで、RETCODE は無効です。戻りコードにはすべて、高位バイトに SCI または ODBM を表す SCI メンバー・タイプ標識が含まれます (X'01' は SCI、X'04' は ODBM を表します)。

**RSNCODE**=symbol

**RSNCODE**=(r2-r12)

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。ODBM 理由コードは、CSLDRR で定義されています。SCI 理由コードは、CSLSRR で定義されています。

**SCITOKEN**=symbol

**SCITOKEN**=(r2-r12)

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されます。

## CSLDMDRG の戻りコードと理由コード

以下の表は、CSLDMDRG マクロ要求で返される戻りコードおよび理由コードの一覧です。

表 40. CSLDMDRG の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	CSLDMDRG 要求が正常に完了しました。
X'04000010'	X'00004004'	クライアントは登録されません。
	X'00004020'	PARM= パラメーターに指定されたパラメーター・リストのパラメーター・リスト・バージョンが無効です。

## CSLDMI: ODBM アプリケーション・プログラム・インターフェース

アセンブラーで作成され、z/OS 上で実行されるアプリケーション・プログラムは、CSL Open Database Manager (ODBM) の CSLDMI API を使用して、IMSplex の DBCTL および DB/DC システム内で IMS DB によって管理されている IMS データベースにアクセスできます。

CSLDMI API は、IMS ODBA インターフェースがサポートするすべての DL/I 呼び出し、グローバルおよびローカル・トランザクション処理、およびセキュリティをサポートします。

CSLDMI 要求を発行する前に、ODBM クライアントは CSLDMREG 要求を発行して、ODBM への登録を行う必要があります。

CSLDMI API には、以下の関数呼び出しが組み込まれています。

#### **BACKOUT**

ローカル・トランザクションのローカル作業単位をバックアウトします。ローカル作業単位は、その作業が APSB スレッドに関連付けられた単一の APSB 呼び出し (DL/I 呼び出しなど) で構成されます。

#### **COMMIT**

ローカル・トランザクションのローカル作業単位をコミットします。ローカル作業単位は、その作業が APSB スレッドに関連付けられた単一の APSB 呼び出し (DL/I 呼び出しなど) で構成されます。

#### **DSECT**

アプリケーション・プログラムに、CSLDMI パラメーター・リストおよび CSLDMI 戻りコードと理由コードのための等価 (EQU) ステートメントを組み込みます。

#### **ODBMCI**

DL/I 呼び出しを IMS データベースに対して発行するには、ODBMCI 関数を使用します。 ODBMCI 関数の DLIFUNC パラメーターを使用して、DL/I 呼び出しを IMS に渡します。

ODBMCI 関数呼び出しを発行する前に、DLIFUNC パラメーターで渡される DL/I 呼び出しに対して適切に AIB フィールドがコード化されていることを確認する必要があります。

以下のパラメーターは ODBMCI 関数に固有であるため、他の CSLDMI 関数呼び出しでは指定できません。

- AIB
- CLIENTID
- CLIENTIDLEN
- CTXTOKEN
- DLIFUNC
- GROUPNAME=
- GROUPNAMELEN=
- IOAREA
- IOAREALEN
- PCB
- PCBLEN
- SECTKNLEN
- SECTOKEN
- SSA1 から SSA15
- SSA1LEN から SSA15LEN
- URTOKEN
- USERID=
- USERIDLEN=

## READYSYNCP

グローバル作業単位に含まれる複数の APSB 呼び出しのそれぞれについて、同期点処理を準備します。

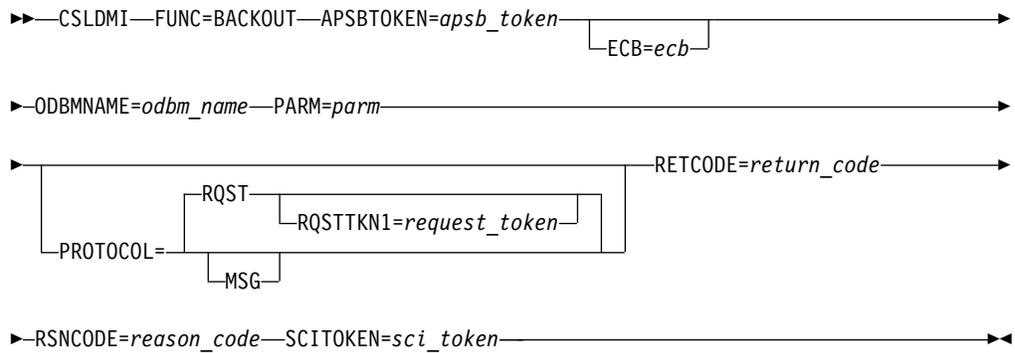
CSLDMI API の READYSYNCP 関数は、グローバル・トランザクション (単一の作業単位内に複数の APSB スレッドが含まれるトランザクションなど) のための同期点処理を開始する前に呼び出さなければなりません。 ODBM クライアントは、グローバル作業単位 (UOW) で APSBTOKEN によって表される各 APSB に対し、FUNC=READYSYNCP を発行する必要があります。これは、URTOKEN パラメーターに関連します。詳しくは、URTOKEN パラメーターを参照してください。

同期点処理を開始する前に、呼び出し元は、グローバル UOW の各 APSB 呼び出しに対し、FUNC=READYSYNCP を発行する必要があります。 CSLDMI は、APSB トークン (APSBTOKEN) を使用して各 APSB 呼び出しを表します。

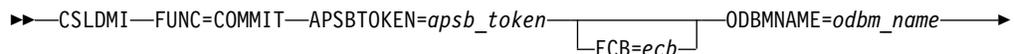
以下のサブセクションがあります。

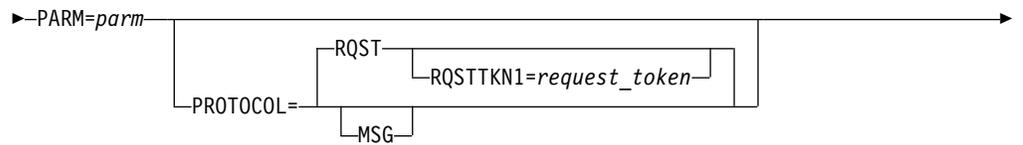
- 『CSLDMI FUNC=BACKOUT の構文』
- 『CSLDMI FUNC=COMMIT の構文』
- 167 ページの『CSLDMI FUNC=DSECT の構文』
- 167 ページの『CSLDMI FUNC=ODBMCI の構文』
- 168 ページの『CSLDMI FUNC=READYSYNCP の構文』
- 168 ページの『CSLDMI 機能パラメーター』
- 176 ページの『CSLDMI の戻りコードと理由コード』

## CSLDMI FUNC=BACKOUT の構文



## CSLDMI FUNC=COMMIT の構文





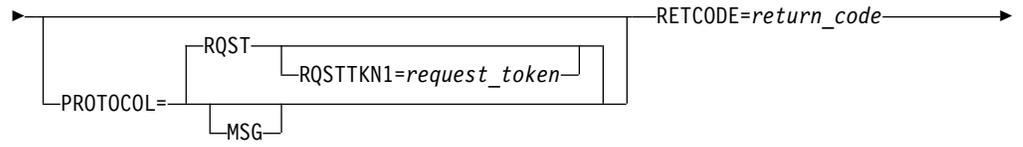
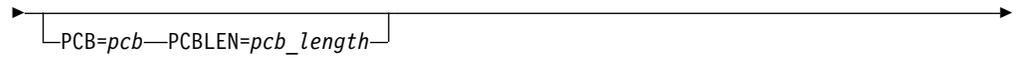
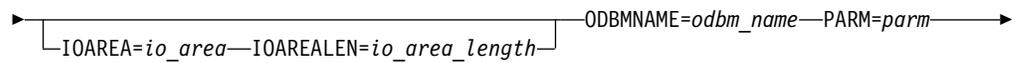
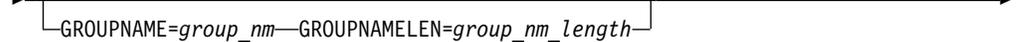
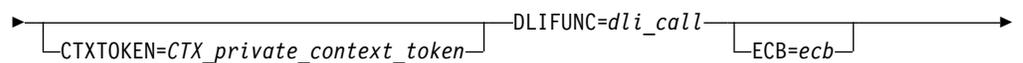
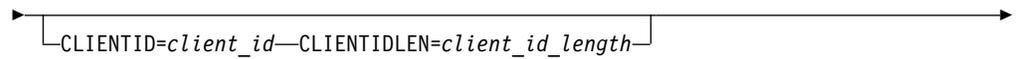
▶ RETCODE=return\_code—RSNCODE=reason\_code—SCITOKEN=sci\_token

### CSLDMI FUNC=DSECT の構文

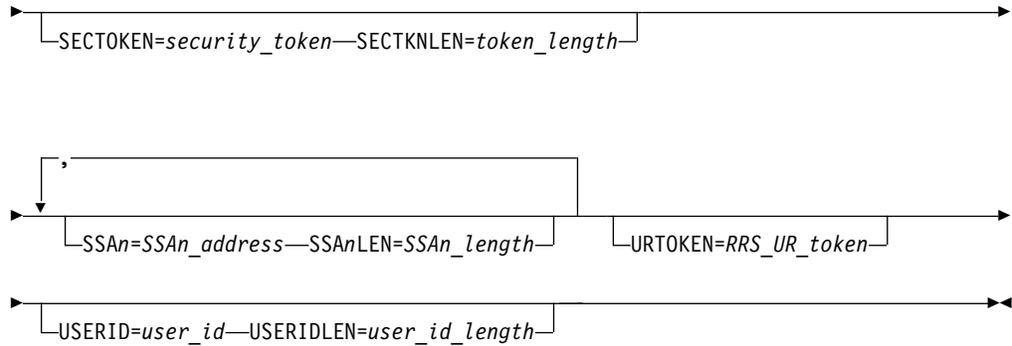
▶ CSLDMI—FUNC=DSECT

### CSLDMI FUNC=ODBMCI の構文

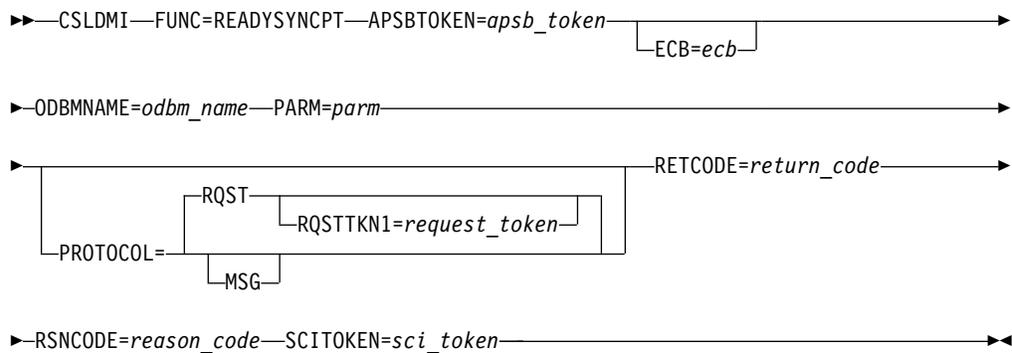
▶ CSLDMI—FUNC=ODBMCI—AIB=aib—APSBTOKEN=apsb\_token



▶ RSNCODE=reason\_code—SCITOKEN=sci\_token



### CSLDMI FUNC=READYSYNCPT の構文



### CSLDMI 機能パラメーター

CSLDMI パラメーターは、ODBM と通信して IMS データベースにアクセスするために必要な ODBM 値を指定します。

アドレスは、シンボルとして指定することも、2 から 12 までのレジスターとして指定することもできます。

ODBMCI 関数を指定する場合にのみサポートされるパラメーターは、パラメーターの説明のなかで注記が示されています。 ODBMCI の場合にのみサポートされるとして注記されていないパラメーターは、DSECT 以外のすべての CSLDMI 関数で指定できます。

以下のパラメーターは、CSLDMI API の 1 つ以上の関数で指定できます。

#### AIB=symbol | (r2-r12)

(必須) - アプリケーション・インターフェース・ブロック (AIB) のアドレスを指定します。 AIB パラメーターは、ODBM に対する入力および出力の両方に必須です。このパラメーターは、ODBMCI 機能呼び出しでのみサポートされません。

CSLDMI 要求で SCI メッセージ・プロトコル (PROTOCOL=MSG) が指定されている場合、ODBM は、CSLDMDIR マクロによってマップされた ODBM ディレクティブ・パラメーター・リストの DDIR\_DMIRAIBPTR フィールドに AIB のアドレスを返します。

CSLDMI FUNC=ODBMCI を発行する前に、DLIFUNC パラメーターで渡される各 DL/I 呼び出しの必須 AIB フィールドを設定する必要があります。AIB のフィールドは、DFSAIB マクロによってマップされています。これについては、「IMS V13 アプリケーション・プログラミング」の『Specifying the AIB mask for ODBA applications』を参照してください。

AIB をレジスターとして指定する場合、そのレジスターには、この AIB 域のアドレスが含まれていなければなりません。AIB をシンボルとして指定する場合、そのシンボルは、この AIB 域のラベルでなければなりません。

#### **APSBTOKEN=symbol | (r2-r12)**

(必須) - 16 バイトの ODBM APSB トークンのアドレスを指定します。APSB トークンは、初期 APSB スレッド要求で CSLDMI によって返されます。このトークンは、このスレッドをターゲットとした以降の呼び出しのすべてで必要となります。

APSB 要求では、APSBTOKEN によって、ODBM からのトークンを受信するフィールドのアドレスが指定されます。APSBTOKEN フィールドの長さは、CSLDMI マクロの DMI\_APSBTKNLEN によって定義されているように、16 バイトでなければなりません。

このスレッドに関連付けられた以降のすべての要求では、APSBTOKEN で、この 16 バイトのトークンのアドレスを指定します。

CSLDMI 要求で SCI メッセージ・プロトコル (PROTOCOL=MSG) が指定されている場合、ODBM は、CSLDMDIR マクロによってマップされた ODBM ディレクティブ・パラメーター・リストの DDIR\_DMIRAPSBTPTR フィールドに APSB トークンのアドレスを返します。

APSBTOKEN パラメーターをレジスターとして指定する場合、そのレジスターには、このトークン・フィールドのアドレスが含まれていなければなりません。APSBTOKEN パラメーターをシンボルとして指定する場合、そのシンボルは、このトークン・フィールドのラベルでなければなりません。

#### **CLIENTID=symbol | (r2-r12)**

(オプション) - エンド・ユーザー・クライアントによって定義されたエンド・ユーザー・クライアント・アプリケーション ID を指定します。

CLIENTID パラメーターをレジスターとして指定する場合、そのレジスターには、このクライアント ID フィールドのアドレスが含まれていなければなりません。CLIENTID パラメーターをシンボルとして指定する場合、そのシンボルは、このクライアント ID フィールドのラベルでなければなりません。

#### **CLIENTIDLEN=symbol | (r2-r12)**

(CLIENTID を指定する場合は必須) - クライアント ID の長さを指定します。

CLIENTIDLEN パラメーターをレジスターとして指定する場合、そのレジスターには、このクライアント ID の長さが含まれていなければなりません。

CLIENTIDLEN パラメーターをシンボルとして指定する場合、そのシンボルは、このクライアント ID の長さが含まれるフルワードのラベルでなければなりません。

#### **CTXTOKEN=symbol | (r2-r12)**

(オプション) - z/OS リカバリー可能リソース管理サービス (RRMS) コンテキスト・サービスの Begin\_Context サービス (CTXBEGC) 呼び出しによって取得

された 16 バイトの専用コンテキスト・トークンを指定します。このパラメーターは、ODBMCI 機能呼び出しでのみサポートされます。

単一の LPAR での 1 つのコミット・スコープに含まれる複数の APSB 呼び出しからなるグローバル作業単位をセットアップするには、CTXTOKEN を使用できます。これを指定する場合には、グローバル作業単位に含まれる初期 APSB 呼び出しごとに、同じ CTXTOKEN を指定する必要があります。

ODBM クライアントは、CTXTOKEN に指定された専用コンテキスト・トークンを ODBM に渡す前に、そのトークンとディスパッチ可能単位 (TCB など) との関連付けが解除されていることを確認する必要があります。ODBM クライアントはまた、

ATRPRP、ATRACMT、ATRABCK、ATRCMIT、ATRBCK などの適切な z/OS リソース・リカバリー・サービス サービスを使用して、同期点処理を開始する必要もあります。

ODBM クライアントは、必要に応じて RRS 作業単位 ID (UWID) を設定する必要があります。通常、UWID は XID です。作業 ID を設定するには、ATRSWID2 を使用します。UWID は、この CTXTOKEN パラメーターに指定されたコンテキスト・トークンに基づいて設定されます。

CTXTOKEN パラメーターをレジスターとして指定する場合、そのレジスターには、この専用コンテキスト・トークン・フィールドのアドレスが含まれていなければなりません。CTXTOKEN パラメーターをシンボルとして指定する場合、そのシンボルは、この専用コンテキスト・トークン・フィールドのラベルでなければなりません。

#### **DLIFUNC=symbol | (r2-r12)**

(必須) - 4 バイトの DL/I 呼び出しを指定します。オープン・データベース・アクセス (ODBA) 呼び出し可能インターフェースでサポートされる任意の DL/I 呼び出しを指定できます。このパラメーターは、ODBMCI 機能呼び出しでのみサポートされます。

CSLDMI FUNC=ODBMCI DLIFUNC を発行する前に、DLIFUNC パラメーターで渡される各 DL/I 呼び出しの必須 AIB フィールドを設定する必要があります。AIB のフィールドは、DFSAIB マクロによってマップされています。これについては、「IMS V13 アプリケーション・プログラミング」の『Specifying the AIB mask for ODBA applications』を参照してください。

DLIFUNC パラメーターをレジスターとして指定する場合、そのレジスターには、この DLI 機能コードのアドレスが含まれていなければなりません。DLIFUNC パラメーターをシンボルとして指定する場合、そのシンボルは、この DLI 機能コードのラベルでなければなりません。

#### **ECB=symbol | (r2-r12)**

(オプション) - 非同期要求に使用される MVS™ イベント制御ブロック (ECB) を指定します。要求が完了したとき、指定された ECB は通知されます。

ECB が指定されていないと、そのタスクは、要求が完了するまで中断されません。ECB が指定された場合、CSLDMI マクロを発行する ODBM クライアントは、CSLDMI から制御を受け取った後、CSLDMI が返すデータ (RETCODE フィールドおよび RSNCODE フィールドを含む) を使用または調査する前に、z/OS WAIT マクロ (または同等のもの) を呼び出す必要があります。

ECB パラメーターをレジスターとして指定する場合、そのレジスターには、この ECB のアドレスが含まれていなければなりません。 ECB パラメーターをシンボルとして指定する場合、そのシンボルは、この ECB ストレージの先頭を示さなければなりません。

#### **GROUPNAME=symbol | (r2-r12)**

(オプション) - RACF または同等のセキュリティー製品のグループ名を指定します。このグループ名は、APSB 呼び出し (DLIFUNC=APSB) にのみ関係するため、他のすべての DL/I 呼び出しでは無視されます。

GROUPNAME パラメーターをレジスターとして指定する場合、そのレジスターには、このグループ名フィールドのアドレスが含まれていなければなりません。 GROUPNAME パラメーターをシンボルとして指定する場合、そのシンボルは、このグループ名フィールドのラベルでなければなりません。

#### **GROUPNAMELEN=symbol | (r2-r12)**

(GROUPNAME を指定する場合は必須) - グループ名の長さを指定します。

GROUPNAMELEN パラメーターをレジスターとして指定する場合、そのレジスターには、このグループ名の長さが含まれていなければなりません。

GROUPNAMELEN パラメーターをシンボルとして指定する場合、そのシンボルは、このグループ名の長さが含まれるフルワードのラベルでなければなりません。

#### **IOAREA=symbol | (r2-r12)**

(条件によっては必須) - データベース DL/I 呼び出しに関する入力または出力データに使用する入出力域を指定します。 IOAREA パラメーターおよび IOAREALEN パラメーターは、入力データを必要とする DL/I 呼び出し、または出力データを返す DL/I 呼び出しが DLIFUNC パラメーターで指定された場合にのみ必須です。このパラメーターは、ODBMCI 機能呼び出しでのみサポートされます。

CSLDMI 要求で SCI メッセージ・プロトコル (PROTOCOL=MSG) が指定されている場合、ODBM は、CSLDMDIR マクロによってマップされた ODBM ディレクティブ・パラメーター・リストの DDIR\_DMIRIOAPT フィールドに入出力域のアドレスを返します。

IOAREA パラメーターをレジスターとして指定する場合、そのレジスターには、この入出力域のアドレスが含まれていなければなりません。 IOAREA パラメーターをシンボルとして指定する場合、そのシンボルは、この入出力域のラベルでなければなりません。

#### **IOAREALEN=symbol | (r2-r12)**

(条件によっては必須) - IOAREA パラメーターによって指定された入出力域の長さを指定します。 IOAREA パラメーターおよび IOAREALEN パラメーターは、入力を必要とする DL/I 呼び出し、または出力を返す DL/I 呼び出しが DLIFUNC パラメーターで指定された場合にのみ必須です。このパラメーターは、ODBMCI 機能呼び出しでのみサポートされます。

IOAREALEN パラメーターに指定した長さは、AIB マスクの AIBOALEN フィールドにも指定する必要があります。 AIB マスクについては、「AIB マスクの指定 (アプリケーション・プログラミング)」を参照してください。

IOAREALEN パラメーターをレジスターとして指定する場合、そのレジスターには、この入出力域の長さが含まれていなければなりません。IOAREALEN パラメーターをシンボルとして指定する場合、そのシンボルは、この入出力域の長さが含まれるフルワードのラベルでなければなりません。

**ODBMNAME=symbol | (r2-r12)**

(必須) - CSLDMI 要求の送信先とする 8 バイトの ODBM 名を指定します。

ODBMNAME パラメーターをシンボルとして指定する場合、そのシンボルは、この ODBM フィールドのラベルでなければなりません。ODBMNAME パラメーターをレジスターとして指定する場合、そのレジスターには、この ODBM 名フィールドのアドレスが含まれていなければなりません。

**PARM=symbol | (r2-r12)**

(必須) -- アドレス CSLDMI パラメーター・リストを指定します。パラメーター・リストの長さは、CSLDMI マクロで DMI\_PARMLN に割り当てられた値以上の長さでなければなりません。

CSLDMI パラメーター・リストの長さ用にアプリケーション・プログラムの中に等価 (EQU) ステートメントを組み込むには、CSLDMI FUNC=DSECT を使用します。

PARM の値をレジスターとして指定する場合、そのレジスターには、このパラメーター・リストのアドレスが含まれていなければなりません。PARM の値をシンボルとして指定する場合、そのシンボルは、このパラメーター・リスト・ストレージの先頭を示さなければなりません。

**PCB=symbol | (r2-r12)**

(オプション) - DL/I 呼び出しの処理後に IMS によって返されたプログラム連絡ブロック (PCB) のアドレスを受け取るための、フルワードのストレージ域のアドレスを指定する出力パラメーター。PCB には、DL/I 呼び出しおよびその他のフィールドに関連する状況コードが含まれます。このパラメーターは、ODBMCI 機能呼び出しでのみサポートされます。

PCB ストレージは、呼び出し元によって事前割り振りされることはありません。要求から返された時点で、PCB パラメーターのアドレスには、PCB が含まれるストレージ・バッファのアドレスが入れられます。ODBM クライアントが PCB での処理を終了した後、ODBM クライアントは SCI 要求 CSLSCBFR FUNC=RELEASE を発行して、PCB ストレージ・バッファを解放する必要があります。

CSLDMI 要求で SCI メッセージ・プロトコル (PROTOCOL=MSG) が指定されている場合、ODBM は、CSLDMDIR マクロによってマップされた ODBM ディレクティブ・パラメーター・リストの DDIR\_DMIRPCBPTR フィールドに PCB ストレージのアドレスを返します。

レジスターとして指定する場合、そのレジスターには、PCB のアドレスが含まれるフルワードのアドレスが含まれていなければなりません。シンボルとして指定する場合、そのシンボルは、PCB のアドレスが含まれるフルワードのラベルでなければなりません。

**PCBLEN=symbol | (r2-r12)**

(条件によっては必須) - DL/I 呼び出し処理によって返される PCB の長さを指

定します。PCBLEN パラメーターは、PCB パラメーターを指定する場合に必須です。このパラメーターは、ODBMCI 機能呼び出しでのみサポートされません。

レジスターとして指定する場合、そのレジスターには、PCB の長さが含まれるフルワードのアドレスが含まれていなければなりません。シンボルとして指定する場合、そのシンボルは、PCB の長さが含まれるフルワードのラベルでなければなりません。

#### **PROTOCOL=MSG | RQST**

(オプション) -- 要求を ODBM に送信するための SCI プロトコルを指定します。

##### **MSG**

CSLDMI が SCI メッセージ・プロトコルを使用して ODBM に入力要求を送信するように指定します。このプロトコルは他の IMSpIex メンバーに対してデータを片方向で送信するため、出力パラメーターをサポートしません。ODBM クライアントが ODBM からの出力を待機することなく、MSG プロトコルを使用する要求によって生成される出力は、非同期で処理されます。

##### **RQST**

CSLDMI が SCI 要求プロトコルを使用して ODBM に入力要求を送信するように指定します。SCI 要求プロトコルは、入力パラメーターおよび出力パラメーターをサポートします。ODBM クライアントは ODBM からの出力を待機し、出力を同期処理します。RQST がデフォルトです。

#### **RETCODE=symbol | (r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。ODBM 戻りコードは、CSLDRR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

戻りコードは、ODBM (CSLDMI) または SCI (CSLSCMSG または CSLSCRQS) から返されます。ECB が指定されている場合、ECB が通知されるまで、RETCODE の値は無効です。戻りコードにはすべて、高位バイトに SCI または ODBM を表す SCI メンバー・タイプ標識が含まれます (X'01' は SCI、X'04' は ODBM を表します)。

#### **RQSTTKN1=symbol | (r2-r12)**

(条件付きオプション) - 非同期処理の場合、出力応答をそれに関連付けられた入力要求に相関させる、16 バイトのユーザー生成要求トークンを指定します。ODBM はこのトークンのアドレスを、CSLDMDIR によってマップされた ODBM ディレクティブ・パラメーター・リストの DDIR\_DMIRQT1PTR フィールドに返します。

RQSTTKN1 は、PROTOCOL=MSG で SCI メッセージ・プロトコルが指定されている場合のみサポートされます。

レジスターとして指定する場合、そのレジスターには、この要求トークン・フィールドのアドレスが含まれていなければなりません。シンボルとして指定する場合、そのシンボルは、この要求トークン・フィールドのラベルでなければなりません。要求トークン・フィールドは、長さ 16 バイトで、左寄せされ、空白で埋め込まれる必要があります。

**RSNCODE=symbol | (r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。ODBM 理由コードは、CSLDRR マクロで定義されています。SCI 理由コードは、CSLSRR で定義されています。

**SCITOKEN=symbol | (r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されます。

SCITOKEN パラメーターをレジスターとして指定する場合、そのレジスターには、この SCI トークン・フィールドのアドレスが含まれていなければなりません。SCITOKEN パラメーターをシンボルとして指定する場合、そのシンボルは、この SCI トークン・フィールドのラベルでなければなりません。

**SECTKNLEN=symbol | (r2-r12)**

(条件によっては必須) - セキュリティー・トークンの長さを指定します。SECTKNLEN パラメーターは、SECTOKEN パラメーターを指定する場合に必須です。このパラメーターは、ODBMCI 機能呼び出しでのみサポートされません。

SECTKNLEN パラメーターをレジスターとして指定する場合、そのレジスターには、このセキュリティ・トークンの長さが含まれていなければなりません。SECTKNLEN パラメーターをシンボルとして指定する場合、そのシンボルは、このセキュリティ・トークンの長さが含まれるフルワードのラベルでなければなりません。

**SECTOKEN=symbol | (r2-r12)**

(条件付きオプション) - RACF または同等のセキュリティ製品がセキュリティ検査に使用する、可変長のセキュリティ・トークンのアドレスを指定します。このセキュリティ・トークンは、DLIFUNC=APSB で指定された APSB DL/I 呼び出しにのみ適用されます。他のすべての DL/I 呼び出しでは、このセキュリティ・トークンは無視されます。このパラメーターは、ODBMCI 機能呼び出しでのみサポートされます。このセキュリティ・トークンは、クライアント・アドレス・スペースが許可された呼び出し元である場合にのみ、使用されます。クライアント・アドレス・スペースが無許可の場合、ユーザー ID が z/OS 制御ブロックから自動的に取得されます。

このセキュリティ・トークンは、セキュリティ・オブジェクトでなければなりません。例えば、RACF を使用している場合には、セキュリティ・トークンは RACO (RACF オブジェクト) でなければなりません。ODBM は、ENVRIN= を設定した RACROUTE REQUEST=VERIFY,ENVIR=CREATE を呼び出して、APSB スレッド用に RACF アクセサー・エレメント (ACEE) を使用するセキュリティ環境を設定します。IMS は、PSB の ODBA または RAS セキュリティーを許可する際に、この ACEE を使用します。

SECTOKEN パラメーターをレジスターとして指定する場合、そのレジスターには、このセキュリティ・トークン・フィールドのアドレスが含まれていなければなりません。SECTOKEN パラメーターをシンボルとして指定する場合、そのシンボルは、このセキュリティ・トークン・フィールドのラベルでなければなりません。

**SSAn=symbol | (r2-r12)**

(オプション) - DL/I 呼び出しのセグメント検索指数 (SSA) を指定します。SSA1 から SSA15 までの最大 15 個の SSA を指定できます。このパラメーターは、ODBMCI 機能呼び出しでのみサポートされます。

SSA1 から SSA15 までのパラメーターをレジスターとして指定する場合、そのレジスターには、SSA のアドレスが含まれていなければなりません。SSA1 から SSA15 までのパラメーターをシンボルとして指定する場合、それぞれのシンボルは、SSA のアドレスが含まれるフルワードのラベルでなければなりません。

**SSAnLEN=symbol | (r2-r12)**

(条件によっては必須) - 対応する SSA リスト域の長さを指定します。指定したすべての SSAn パラメーターには、SSA1LEN から SSA15LEN までの対応する SSAnLEN パラメーターを指定する必要があります。このパラメーターは、ODBMCI 機能呼び出しでのみサポートされます。

SSA1LEN から SSA15LEN までのパラメーターをレジスターとして指定する場合、そのレジスターには、SSA リスト域の長さが含まれていなければなりません。SSA1LEN から SSA15LEN までのパラメーターをシンボルとして指定する場合、そのシンボルは、SSA リスト域の長さが含まれるフルワードのラベルでなければなりません。

**URTOKEN=symbol | (r2-r12)**

(オプション) - 16 バイトの RRS 親リカバリー単位 (UR) トークンを指定します。このトークンは、RRS Express\_UR\_Interest サービス (ATREINT5) の呼び出しによって取得されます。複数の LPAR での単一のコミット・スコープに含まれる複数の APSB 呼び出しからなるグローバル作業単位をセットアップするには、URTOKEN が必須です。これを指定する場合には、グローバル作業単位に含まれる初期 APSB 呼び出しごとに、同じ URTOKEN を含める必要があります。このパラメーターは、ODBMCI 機能呼び出しでのみサポートされます。

ODBM クライアントは、スーパーバイザー状態、または PSW キー 0 から 7 で実行されているか、この両方で実行されている z/OS 認可プログラムである必要があります。ODBM クライアントを問題状態で実行することはできません。

ODBM クライアントは、

ATRAPRP、ATRACMT、ATRABCK、ATRCMIT、ATRBCK などの適切な RRS サービスを使用して、同期点処理を開始する必要があります。同期点処理を開始する前に、ODBM クライアントは、グローバル UOW の各 APSB 呼び出しに対し、FUNC=READYSYNCPPT を発行する必要があります。CSLDMI は、APSB トークンを使用して各 APSB 呼び出しを表します。

ODBM クライアントは、必要に応じて RRS 作業単位 ID (UWID) を設定する必要があります。通常、UWID は XID です。UWID を設定するには、ATRSWID2 を使用します。UWID は、この URTOKEN パラメーターで指定された親 UR トークンに基づきます。

URTOKEN パラメーターをレジスターとして指定する場合、そのレジスターには、この RRS UR トークン・フィールドのアドレスが含まれていなければなりません。URTOKEN パラメーターをシンボルとして指定する場合、そのシンボルは、この RRS UR トークン・フィールドのラベルでなければなりません。

### USERID=symbol | (r2-r12)

(オプション) - RACF または同等のセキュリティー製品が使用するユーザー ID を指定します。このパラメーターは、クライアント・アドレス・スペースがこの要求に対して許可されている場合にのみ使用します。クライアントが許可されていない場合は、ユーザー ID が z/OS 制御ブロックから自動的に取得されます。このユーザー ID は、APSB 呼び出し (DLIFUNC=APSB) にのみ関係するため、他のすべての DL/I 呼び出しでは無視されます。

USERID パラメーターをレジスターとして指定する場合、そのレジスターには、このユーザー ID フィールドのアドレスが含まれていなければなりません。USERID パラメーターをシンボルとして指定する場合、そのシンボルは、このユーザー ID フィールドのラベルでなければなりません。

### USERIDLEN=symbol | (r2-r12)

(USERID を指定する場合は必須) - ユーザー ID の長さを指定します。

USERIDLEN パラメーターをレジスターとして指定する場合、そのレジスターには、このユーザー ID の長さが含まれていなければなりません。

USERIDLEN パラメーターをシンボルとして指定する場合、そのシンボルは、このユーザー ID の長さが含まれるフルワードのラベルでなければなりません。

## CSLDMI の戻りコードと理由コード

以下の表は、CSLDMI マクロ要求で返される戻りコードおよび理由コードの一覧です。

表 41. CSLDMI の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	CSLDMI 要求が正常に完了した。
X'04000004'	X'00001004'	入力ユーザー出口が要求をリジェクトした。
X'04000008'	X'00002018'	無効な AIB パラメーター。
	X'0000201C'	AIB マスクの AIBRSNM1 フィールドに指定した値が無効。
	X'00002020'	サポートされない DL/I 機能。
	X'00002024'	入力ユーザー出口が誤って、IAB マスクの AIBOALEN フィールドに、IOAREALEN パラメーターで指定されている値よりも大きい値を設定した。
X'04000010'	X'00004000'	別名が見つからない。
	X'00004004'	クライアントは登録されません。
	X'00004008'	データ・ストアを獲得できない。

表 41. CSLDMI の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'04000014'	X'00005004'	APSB 制御ブロックを獲得できない。
	X'00005008'	AIB 制御ブロックを獲得できない。
	X'00005014'	APSB ブロック用のハッシュ・テーブル ADD に失敗。
	X'00005034'	OUTPUT バッファ割り振りに失敗。
	X'00005040'	RRS0_ASSOCCTX に失敗。
	X'00005044'	RRS0_DISCTX に失敗。
	X'00005048'	RRS0_COMMIT3_DMIR に失敗。
	X'0000504C'	RRS0_SUSI に失敗。
	X'00005050'	RRS0_BACKOUT3_DMIR に失敗。
	X'00005054'	APSB トークンの長さにエラーがある。
	X'00005058'	APSB ハッシュ・テーブル RELEASE に失敗。
	X'0000505C'	RRS0_ASSOCCTX3 に失敗。
	X'00005060'	RRS0_CASCADE3 に失敗。
	X'00005064'	RRS0_GETCTX2 に失敗。
	X'00005068'	RRS0_ENDCTX2 に失敗。
	X'0000506C'	SEC0_CREATE 呼び出しに失敗。
	X'00005070'	SEC0_DELETE 呼び出しに失敗。
	X'00005074'	PSB をスケジュールするための APSB 呼び出しに失敗。
	X'00005078'	PSB 名にエラーがある。
	X'00005080'	PAPL ブロック割り振りに失敗。
	X'00005084'	RRS がグローバル・トランザクションにアクティブでない。
X'0000507C'	DMI IOA ストレージを獲得できない。	
X'00005094'	ORRS TCB への BPETCBSW が失敗しました。	
X'00005098'	ODRA TCB への BPETCBSW が失敗しました。	
X'0000509C'	ORRS TTE の取得に失敗しました。	

関連資料:

-  [ODBA アプリケーションの AIB マスクの指定 \(アプリケーション・プログラミング\)](#)
-  [データベース管理 \(アプリケーション・プログラミング API\)](#)
-  [データベース管理呼び出しの要約 \(アプリケーション・プログラミング API\)](#)

## CSLDMREG: ODBM クライアント登録要求

CSLDMREG 要求は、ODBM クライアントを ODBM に登録します。

WebSphere Application Server for z/OS や Db2 for z/OS などの ODBM クライアントを ODBM のインスタンスに登録するには、CSLDMREG 要求を使用します。

CSLDMREG 要求は、アプリケーション・プログラムを ODBM クライアントとして ODBM に登録します。この登録コマンドは、クライアントが ODBM に発行する最初の要求でなければなりません。

IMSplex 内の全 ODBM の名前を取得するには、CSLSCQRY 要求を使用してください。

## CSLDMREG DSECT の構文

CSLDMREG 要求の DSECT 機能は、CSLDMREG パラメーター・リストの長さ、および戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶ CSLDMREG—FUNC=DSECT

## 要求プロトコルの構文

▶▶ CSLDMREG—FUNC=REGISTER—ECB=*ecb*—ODBMNAME=*odbmname*—OUTLEN=*outlen*

▶—OUTPUT=*output*—PARM=*parm*—RETCODE=*returncode*—RETNAME=*odbmname*

▶—RSNCODE=*reasoncode*—SCITOKEN=*scitoken*

## CSLDMREG パラメーター

CSLDMREG パラメーターは、ODBM での登録に必要な ODBM 値を指定します。

アドレスは、シンボルとして指定することも、2 から 12 までのレジスターとして指定することもできます。

CSLDMREG コマンドには、以下のパラメーターが含まれます。

**ECB**=*symbol*

**ECB**=(*r2-r12*)

(オプション) -- 非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定します。要求が完了したとき、指定された ECB は通知されません。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、マクロの呼び出し側は、CSLDMREG から制御を受け取った後、このマクロから返されたデータ (RETCODE フィールドおよび RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を発行する必要があります。

ECB の値をシンボルとして指定する場合、そのシンボルは、この ECB ストレージの先頭を示さなければなりません。ECB の値をレジスターとして指定する場合、そのレジスターには、この ECB のアドレスが含まれていなければなりません。

**ODBMNAME=***symbol*

**ODBMNAME=**(*r2-r12*)

(オプション) -- コマンド登録要求の送信先とする 8 バイトの ODBM 名を指定します。

CSLDMREG 要求では、ODBMNAME パラメーターまたは RETNAME パラメーターのいずれか 1 つを指定する必要があります。両方のパラメーターを指定することはできません。

ODBM クライアントを ODBM の具体的な既知のインスタンスに接続するには、ODBMNAME パラメーターを使用します。

ODBMNAME の値をシンボルとして指定する場合、そのシンボルは ODBM フィールドのラベルでなければなりません。ODBMNAME の値をレジスターとして指定する場合、そのレジスターには ODBM 名フィールドのアドレスが含まれていなければなりません。

**OUTLEN=***symbol*

**OUTLEN=**(*r2-r12*)

(必須) -- CSLDMREG 要求から戻される出力の長さを受け取る 4 バイト・フィールドを指定します。OUTLEN には、OUTPUT= パラメーターで示された出力の長さが入っています。

出力が作成されない場合、出力の長さはゼロになります。出力を作成する前にエラーが検出された場合も同じくゼロになります。

OUTLEN の値をレジスターとして指定する場合、そのレジスターには出力の長さフィールドのアドレスが含まれていなければなりません。OUTLEN の値をシンボルとして指定する場合、そのシンボルは出力の長さフィールドのラベルでなければなりません。

**OUTPUT=***symbol*

**OUTPUT=**(*r2-r12*)

(必須) -- CSLDMREG 要求から戻される可変長出力を受け取る 4 バイト・フィールドを指定します。出力長は、OUTLEN= フィールドに入れて戻されます。

出力が作成されない場合、出力アドレスはゼロになります。出力を作成する前にエラーが検出された場合も同じくゼロになります。

CSLDREGO マクロは、戻される出力をマップします。出力バッファには、ODBM バージョン、別名の数、および 4 バイトの別名のリストが含まれます。

出力バッファは、呼び出し元によって事前割り振りされることはありません。要求から返された時点で、このワードには、更新出力が入ったバッファのアドレスが含まれます。

ODBM クライアントがバッファ・ストレージを必要としなくなったら、ODBM クライアントは CSLSCBFR FUNC=RELEASE 要求を発行して、ストレージを解放する必要があります。

OUTPUT の値をレジスターとして指定する場合、そのレジスターには出力アドレスを収容するフィールドのアドレスが含まれていなければなりません。

OUTPUT の値をシンボルとして指定する場合、そのシンボルは出力を収容するフィールドのラベルでなければなりません。

**PARM=***symbol*

**PARM=(***r1-r12***)**

(必須) -- CSLDMREG パラメーター・リストのアドレスを指定します。パラメーター・リストの長さは、DREG\_PARMLN でパラメーター・リストの長さとして定義された EQU 値と同じでなければなりません。

PARM の値をシンボルとして指定する場合、そのシンボルは、このパラメーター・リスト・ストレージの先頭を示さなければなりません。PARM の値をレジスターとして指定する場合、そのレジスターには、このパラメーター・リストのアドレスが含まれていなければなりません。

**RETCODE=***symbol*

**RETCODE=(***r2-r12***)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。ODBM 戻りコードは、CSLDRR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

戻りコードは、ODBM (CSLDMREG) または SCI (CSLSCMSG または CSLSCRQS) から返されます。ECB が指定されている場合、ECB が通知されるまで、RETCODE は無効です。戻りコードにはすべて、高位バイトに SCI または ODBM を表す SCI メンバー・タイプ標識が含まれます (X'01' は SCI、X'04' は ODBM を表します)。

RETCODE の値をシンボルとして指定する場合、そのシンボルは戻りコード・フィールドのラベルでなければなりません。RETCODE の値をレジスターとして指定する場合、そのレジスターには戻りコード・フィールドのアドレスが含まれていなければなりません。

**RETNAME=***symbol*

**RETNAME=(***r2-r12***)**

(オプション) -- ODBM 名を受け取る 8 バイト出力フィールドを指定します。これは、SCI の要求送信先のターゲット・アドレス・スペースの CSL メンバー名です。

CSLDMREG 要求では、RETNAME パラメーターまたは ODBMNAME パラメーターのいずれか 1 つを指定する必要があります。両方のパラメーターを指定することはできません。

ODBM インスタンスの名前がわからない場合、または ODBM クライアントに ODBM の特定インスタンスとの接続が必要ない場合には、RETNAME パラメーターを使用します。RETNAME を指定すると、SCI は使用可能な任意の ODBM のインスタンスに接続し、その ODBM の名前を RETNAME 出力フィールドに返します。

OUTPUT の値をレジスターとして指定する場合、そのレジスターには出力アドレスを収容するフィールドのアドレスが含まれていなければなりません。

OUTPUT の値をシンボルとして指定する場合、そのシンボルは出力を収容するフィールドのラベルでなければなりません。

**RSNCODE=***symbol*

**RSNCODE=(***r2-r12***)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。ODBM 理由コードは、CSLDRR で定義されています。SCI 理由コードは、CSLSRR で定義されています。

RSNCODE の値をシンボルとして指定する場合、そのシンボルは理由コード・フィールドのラベルでなければなりません。RSNCODE の値をレジスターとして指定する場合、そのレジスターには理由コード・フィールドのアドレスが含まれていなければなりません。

**SCITOKEN**=*symbol*

**SCITOKEN**=(*r2-r12*)

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されます。

SCITOKEN の値をレジスターとして指定する場合、そのレジスターには、この SCI トークン・フィールドのアドレスが含まれていなければなりません。

SCITOKEN の値をシンボルとして指定する場合、そのシンボルは、この SCI トークン・フィールドのラベルでなければなりません。

## CLSDMREG の戻りコードと理由コード

以下の表は、CLSDMREG マクロ要求で返される戻りコードおよび理由コードの一覧です。16 進値 04 は、ODBM の SCI メンバー・タイプを表します。

表 42. CLSDMREG の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	CLSDMREG 要求が正常に完了した。
X'04000010'	X'0000401C'	登録済みのクライアント。
	X'00004020'	PARM= パラメーターに指定されたパラメーター・リストのパラメーター・リスト・バージョンが無効です。
X'04000014'	X'0000502C'	CLSDREGO フィードバック域を獲得できない。
	X'0000503C'	CLNT 制御ブロックを獲得できない。

関連概念:

161 ページの『ODBM クライアント要求の順序』

関連タスク:

122 ページの『ODBM クライアントの登録』



---

## 第 6 章 CSL OM クライアントの作成

以降のトピックでは、OM クライアントを作成するためのクライアント要求とディレクティブについて説明します。

---

### CSL OM コマンド処理クライアントの要求

以下のトピックでは、コマンド処理クライアントによって行われる要求について説明します。

ユーザー自身がコマンド処理クライアントを作成する場合は、RACF PERMIT コマンドに指定するアクセス権限が、コマンドが登録されたときに指定されたアクセス権限と一致していることを確認してください。

関連タスク:

123 ページの『OM コマンド処理クライアントの登録』

### CSLOMBLD: コマンド登録作成

CSLOMBLD 要求を使用して、CSLOMREG 要求で OM に渡されるコマンド・リストを作成することができます。

このリストは、IMS システムの呼び出しが可能なコマンドを識別します。このリストには、CSLOMBLD FUNC=BEGIN ステートメントで始まり、CSLOMBLD FUNC=END ステートメントで終わる一連のステートメントが記載されています。

それぞれがコマンド verb を定義する CSLOMBLD FUNC=DEFVRB ステートメントをいくつでも記述することができます。各 DEFVRB ステートメントの後に CSLOMBLD FUNC=DEFKEY ステートメントが続き、この DEFKEY ステートメントは、直前に定義したコマンド verb に有効なキーワードを識別します。

一連の CSLOMBLD ステートメントは、独立したデータ専用アセンブラー・モジュールに定義することもできるし、実行可能アセンブラー・モジュールの静的データ・セクションに定義することもできます。CSLOMBLD マクロ内の説明を参照してください。

CSLOMBLD は、コマンド登録リストを作成するためのものです。入力パラメーター・リストは指定しません。

#### CSLOMBLD の構文

##### CSLOMBLD BEGIN

BEGIN 機能ステートメントは、一連のコマンド・ステートメントの先頭を表します。

▶▶—CSLOMBLD—FUNC=BEGIN—◀◀

### ***CSLOMBLD DEFVRB***

DEFVRB 機能ステートメントは、OM クライアントまたは IMS システムがサポートするコマンドを識別するときに使用します。 コマンド `verb` の短縮形を指定することもできます。

▶▶—CSLOMBLD—FUNC=DEFVRB—VERB=verbname—NORM=shortverbname————▶▶

### ***CSLOMBLD DEFKEY***

DEFKEY 機能ステートメントは、直前に定義したコマンドに有効なキーワードを識別します。このステートメントで、コマンド・ルーティングならびに必要な RACF 許可を指定することもできます。

▶▶—CSLOMBLD—FUNC=DEFKEY—KEYW=keyword—ROUTE=ANY|ALL—SEC=READ|UPDATE————▶▶

### ***CSLOMBLD DEFGMR***

DEFGMR 機能ステートメントは、出力の構文解析文法を記述するステートメントの先頭を表します。

注: この機能は、IBM 内部使用専用です。

▶▶—CSLOMBLD—FUNC=DEFGMR————▶▶

### ***CSLOMBLD ENDGMR***

ENDGMR 機能ステートメントは、出力の構文解析文法を記述するステートメントの終わりを指定します。

注: この機能は、IBM 内部使用専用です。

▶▶—CSLOMBLD—FUNC=ENDGMR————▶▶

### ***CSLOMBLD END***

END 機能ステートメントは、コマンド・ステートメントのリストの終わりを指定します。

▶▶—CSLOMBLD—FUNC=END————▶▶

## **CSLOMBLD パラメーター**

### **KEYW=keyword**

このパラメーターの直前のコマンド `verb` 用の有効なキーワードを指定します。ヌル・キーワードにはブランクを使用します。例えば、`'KEYW= '`。このパラメーターは、FUNC=DEFKEY のとき、必須です。

**NORM=shortverbname**

定義するコマンドの短縮形を指定します。このパラメーターは、  
FUNC=DEFVRB のとき、必須です。

**ROUTE=ANY | ALL**

定義するコマンドのオーバーライド・ルーティングを指定します。このパラメーターは、FUNC=DEFKEY のとき、必須です。

**SEC=READ | UPDATE**

KEYW に対する、必須の RACF 許可を指定します。このパラメーターは、  
FUNC=DEFKEY のとき、必須です。

**VERB=verbname**

定義するコマンドの長形式を指定します。このパラメーターは、  
FUNC=DEFVRB のとき、必須です。

**CSLOMBLD の例**

以下に、一連の CSLOMBLD ステートメントの例を示します。

図 7. CSLOMBLD ステートメントの例

```
CSLOMBLD FUNC=BEGIN
CSLOMBLD FUNC=DEFVRB,VERB=ACTIVATE,NORM=ACT
CSLOMBLD FUNC=DEFKEY,KEYW=LINK,SEC=UPDATE
CSLOMBLD FUNC=DEFKEY,KEYW=NODE,SEC=UPDATE
CSLOMBLD FUNC=END
```

**ROUTE パラメーターによる CSL OM コマンド経路指定のオーバーライド**

CSLOMBLD では、コマンド処理クライアントは、ユーザーがコマンドを入力するときに指定した経路指定をオーバーライドすることができます。コマンド経路指定のオーバーライドを指定するコマンドがいくつかあります。2つのコマンド処理クライアントが同じコマンドに対して異なる経路指定オーバーライドを指定した場合、OM がコマンド経路指定をオーバーライドします。

- 少なくとも1つのコマンド処理クライアントが ROUTE=ALL のオーバーライドを指定した場合、OM は、登録されているすべてのコマンド処理クライアントにコマンドを経路指定します。
- 少なくとも1つのコマンド処理クライアントが ROUTE=ANY のオーバーライドを指定していて、どのコマンド処理クライアントも ROUTE=ALL を指定していない場合、OM は、登録されているコマンド処理クライアントの1つにコマンドを経路指定します。
- どのコマンド処理クライアントも ROUTE=ALL も ROUTE=ANY も指定していない場合、OM は、コマンドを入力したユーザーの指定どおりにコマンドを経路指定します。

OM コマンドに ROUTE パラメーターが指定されている場合、IMS は、経路リスト内で最も高いレベルの IMS をコマンド・マスターとして選択します。例えば、IMS バージョン 13 CQS を持つ IMS バージョン 13 システム、および以前のバージョンの SCI を持つ別の IMS システムが含まれる IMSplex 構成で、INIT OLC



を作成したのが SCI (X'01') か OM (X'02') かを識別します。OM 戻りコードは、CSLORR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。OM 理由コードは、CSLORR で定義されています。SCI 理由コードは、CSLSRR で定義されています。返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol**

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されます。

## CSLOMDRG の戻りコードと理由コード

以下の表は、CSLOMDRG マクロ要求で返される戻りコードおよび理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 43. CSLOMDRG の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。

## CSLOMOUT: 非送信請求出力要求

コマンドへの応答として間接的にメッセージを送信する必要があるコマンド処理クライアントは、CSLOMOUT 要求を発行します。メッセージは、初期のコマンド応答が OM に返された後に発行されたコマンドの結果としての追加情報である場合、またはシステム内のイベントの結果としての通知メッセージの場合があります。OM は、非送信請求メッセージを OM 出力ユーザー出口に送ります。

### CSLOMOUT の構文

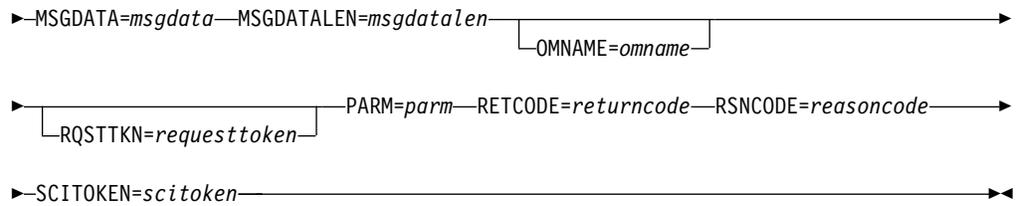
#### DSECT 構文

CSLOMOUT 要求の DSECT 機能は、CSLOMOUT パラメーター・リストの長さ、および戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶CSLOMOUT—FUNC=DSECT—▶▶

#### 要求プロトコルの構文

▶▶CSLOMOUT—FUNC=MESSAGE—CMD=cmdinput—CMDLEN=cmdinputlen—▶▶



## CSLOMOUT のパラメーター

**CMD=***symbol*

**CMD=(r2-r12)**

(オプション) -- コマンド入力バッファを指定します。これは、OM API から指定できる任意の IMS コマンドが可能です。このパラメーターは、オリジナルのコマンド入力を表します。

**CMDLEN=***symbol*

**CMDLEN=(r2-r12)**

(オプション) -- コマンド入力バッファの長さを指定します。

**MSGDATA=***symbol*

**MSGDATA=(r2-r12)**

(必須) -- コマンド応答メッセージ・バッファを指定します。

**MSGDATALEN=***symbol*

**MSGDATALEN=(r2-r12)**

(必須) -- コマンド応答メッセージ・バッファの長さを指定します。

**OMNAME=***symbol*

**OMNAME=(r2-r12)**

(オプション) -- 非送信請求出力メッセージがコマンドへの非同期応答である場合、そのメッセージの送信先である 8 バイトの OM 名を指定します。

**RQSTTKN=***symbol*

**RQSTTKN=(r2-r12)**

(オプション) -- OM コマンド・ディレクティブでコマンド・プロセッサ・クライアントに渡された、32 バイトの要求トークンを指定します。これは、CSLOMI と CSLOMCMMD 要求の一方または両方で入力される、RQSTTKN1 および RQSTTKN2 フィールドを表します。

**PARM=***symbol*

**PARM=(r2-r12)**

(必須) -- CSLOMOUT パラメーター・リストを指定します。パラメーター・リストの長さは、OOUT\_PARMLN で定義されたパラメーター・リスト長 EQU 値と同じでなければなりません。

**RETCODE=***symbol*

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。返される戻りコードについては、以下の表に説明があります。

戻りコードは、OM (CSLOMOUT) または SCI (CSLSCMSG または CSLSCRQS) から返されます。ECB が指定されている場合、ECB が通知されるまで、RETCODE は無効です。戻りコードの上位バイトの値は、戻りコード

を作成したのが SCI (X'01') か OM (X'02') かを識別します。OM 戻りコードは、CSLORR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。OM 理由コードは、CSLORR で定義されています。SCI 理由コードは、CSLSRR で定義されています。返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol**

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されます。

## CSL OMOUT の戻りコードと理由コード

以下の表は、CSL OMOUT マクロ要求で返される戻りコードおよび理由コードの一覧です。以下の表は、CSL OMI マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 44. CSL OMOUT の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。

## CSL OMRDY: レディー要求

CSL OMRDY 要求により、IMS 制御領域のようなコマンド処理クライアントは、コマンドを処理する準備が整ったことを OM に知らせることができます。OM は、この要求が処理されるまで、クライアントにコマンドを送信しません。

### CSL OMRDY の構文

**CSL OMRDY DSECT** の構文

CSL OMRDY 要求の DSECT 機能は、CSL OMRDY パラメーター・リストの長さ、および戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶ CSL OMRDY FUNC=DSECT ◀◀

**CSL OMRDY** 要求プロトコルの構文

▶▶ CSL OMRDY FUNC=READY [OMNAME=omname] [MASTER=NO|YES] PARM=parm ▶▶

▶▶ RETCODE=returncode RSNCODE=reasoncode SCITOKEN=scitoken ▶▶

## CSLOMRDY のパラメーター

**MASTER=NO**

**MASTER=YES**

(オプション) -- このクライアントをコマンド・マスターとして選択するかどうかを指定します。クライアントが MASTER=YES を指定すると、OM は、そのクライアントをコマンド・マスターにすることを選択します。クライアントが MASTER=NO を指定すると、OM は、他のどのクライアントも MASTER=YES を指定していない場合のみ、そのクライアントをコマンド・マスターとして選択します。

**OMNAME=symbol**

**OMNAME=(r2-r12)**

(オプション) -- コマンド・レディー要求の送信先である 8 バイトの OM 名を指定します。OM 名が指定されないと、作業可能要求は、IMSpIex 内に登録されているすべての OM アドレス・スペースに送られます。

**PARM=symbol**

**PARM=(r2-r12)**

(必須) -- CSLOMRDY パラメーター・リストを指定します。パラメーター・リストの長さは、ORDY\_PARMLN で定義されたパラメーター・リスト長 EQU 値と同じでなければなりません。

**RETCODE=symbol**

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。返される戻りコードについては、以下の表に説明があります。

戻りコードは、OM (CSLOMRDY) または SCI (CSLSMSG または CSLSRQS) から返されます。ECB が指定されている場合、ECB が通知されるまで、RETCODE は無効です。戻りコードの上位バイトの値は、戻りコードを作成したのが SCI (X'01') か OM (X'02') かを識別します。OM 戻りコードは、CSLORR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。OM 理由コードは、CSLORR で定義されています。SCI 理由コードは、CSLSRR で定義されています。返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol**

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSREG FUNC=REGISTER 要求によって戻されます。

## CSLOMRDY の戻りコードと理由コード

以下の表は、CSLOMRDY マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 45. CLSOMRDY の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。

関連概念:

 CSL OM コマンド・ルーティング (システム管理)

## CSL OM REG: コマンド登録要求

IMS 制御領域などのコマンド処理クライアントは、CSL OM REG 要求によって、OM にコマンドを登録することができます。コマンドを登録することにより、OM に、クライアントが処理可能なコマンドが通知されます。

CSL OM REG は、コマンド処理クライアントが OM に発行する最初の要求でなければなりません。コマンド処理クライアントは、IMSplex 内の全 OM アドレス・スペースに登録しなければなりません。クライアントが IMSplex 内の 1 つの OM にしか登録されていないと、その OM がダウンしたとき、そのクライアントのコマンドは IMSplex 内の別の OM に経路指定されません。IMSplex 内の全 OM の名前を入手するには、CSL SCQRY 要求を使用してください。クライアントは、CSL OM REG 要求を出すための許可が必要です。この許可は SCI から取得され、それにより、そのクライアントは要求を出せることを OM に知らせます。

### CSL OM REG の構文

#### CSL OM REG DSECT の構文

CSL OM REG 要求の DSECT 機能は、CSL OM REG パラメーター・リストの長さ、および戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶ CSL OM REG—FUNC=DSECT—▶▶

#### CSL OM REG 要求プロトコルの構文

CSL OM REG 要求の構文は次のとおりです。

▶▶ CSL OM REG—FUNC=REGISTER—CMDLIST=cmdlistaddr—CMDLISTLEN=cmdlistlen—▶▶

▶▶ ECB=ecbaddress—OMNAME=omnameaddr—▶▶

▶▶ OUTPUT=outputaddr—OUTLEN=outputlen—PARAM=paramaddr—▶▶

▶▶ RETCODE=returncodeaddr—RSNCODE=reasoncodeaddr—SCITOKEN=scitokenaddr—▶▶

### CSL OM REG のパラメーター

**CMDLIST=***symbol*

**CMDLIST=***(r2-r12)*

(必須) -- コマンド定義リストを指定します。

コマンド・リストは、CSLOMBLD マクロを使って作成します。

**CMDLISTLEN=***symbol*

**CMDLISTLEN=(r2-r12)**

(必須) -- コマンド定義リスト・バッファの長さを指定します。

**ECB=***symbol*

**ECB=(r2-r12)**

(オプション) -- 非同期要求に使用される z/OS イベント制御ブロック (ECB) のアドレスを指定します。要求が完了したとき、指定された ECB は通知されません。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、マクロの発行元は、CSLOMREG から制御を受け取った後、このマクロから戻されたデータ (RETCODE および RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を出す必要があります。

**OMNAME=***symbol*

**OMNAME=(r2-r12)**

(必須) -- コマンド登録要求の送信先である 8 バイトの OM 名を指定します。

**OUTLEN=***symbol*

**OUTLEN=(r2-r12)**

(オプション) -- CSLOMREG 要求から戻される出力の長さを受け取る 4 バイト・フィールドを指定します。OUTLEN には、OUTPUT= パラメーターで示された出力の長さが入っています。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力長はゼロです。

**OUTPUT=***outputaddr*

**OUTPUT=(r2-r12)**

(必須) -- CSLOMREG 要求から戻される可変長出力を受け取るフィールドを指定します。出力長は、OUTLEN= フィールドに入れて戻されます。

出力は、CSLOREGO マクロによってマップされ、1 つ以上のコマンドの登録でエラーがあった場合にのみ、作成されます。その出力には、ヘッダーと 1 つ以上のリスト項目が含まれます。出力フィールドについては、CSLOREGO マクロを参照してください。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力アドレスはゼロです。

出力バッファは、呼び出し元によって事前割り振りされることはありません。要求がこのワードを戻した場合、そこには、更新出力を含むバッファのアドレスが入っています。ストレージの使用が終了したとき、呼び出し元は、CSLSCBFR FUNC=RELEASE 要求を出すことにより、このストレージを解放する必要があります。出力の長さは、OUTLEN= フィールドに入れて戻されます。

**PARM=***symbol*

**PARM=(r2-r12)**

(必須) -- SCI にパラメーターを渡すために要求が使用するストレージのアドレ

スを指定する、4 バイトの入力パラメーター。パラメーター・リストの長さは、OREG\_PARMLN で定義されたパラメーター・リスト長 EQU 値と同じでなければなりません。

**RETCODE=symbol**

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。返される戻りコードについては、以下の表に説明があります。

戻りコードは、OM (CSLORMREG) または SCI (CSLSCMSG または CSLSCRQS) から返されます。ECB が指定されている場合、ECB が通知されるまで、RETCODE は無効です。戻りコードの上位バイトの値は、戻りコードを作成したのが SCI (X'01') か OM (X'02') かを識別します。OM 戻りコードは、CSLORR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。

OM 理由コードは、CSLORR で定義されています。SCI 理由コードは、CSLSRR で定義されています。返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol**

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されます。

**CLSOMREG 戻りコードおよび理由コード**

以下の表は、CSLORMREG マクロ要求で返される戻りコードおよび理由コードの一覧です。

表 46. CLSOMREG 戻りコードおよび理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'0200000C'	X'00003000'	要求は完全に処理されたかもしれないし、処理されなかったかもしれません。要求で OUTPUT パラメーターを指定した場合、出力バッファの完了コードを参照してエラー条件を調べてください。完了コードは、エラーの理由とリソース名を示します。返される可能性のある完了コードについては、以下の表に説明があります。
X'02000010'	X'00004010'	コマンドを発行したクライアントが許可されていません。
	X'00004020'	パラメーター・リスト・バージョンが無効。
X'02000014'	X'00005034'	OM 内部エラーが発生した。OM は、出力バッファ用にストレージを取得できませんでした。
	X'00005038'	OM 内部エラーが発生した。OM は、コマンド処理中に、コマンド verb ハッシュ・テーブルに VERB ブロックを追加できませんでした。

表 46. CLSOMREG 戻りコードおよび理由コード (続き)

戻りコード	理由コード	意味
	X'0000503C'	OM 内部エラーが発生した。OM は、コマンド処理中に、クライアントに CLNT ブロックを割り振ることができませんでした。

以下の表は、CSLQMREG 要求で返される完了コードの一覧です。コマンド登録中にエラーが起きても、この完了コードは、CSLOREGO マクロのフィールド ORGE\_CC に入れて戻され、これが OUTPUT= 域をマップします。

表 47. CLSOMREG の完了コード

完了コード	意味
X'00000104'	OM は、VERB ブロックをリソースに割り振ることができませんでした。
X'00000108'	OM は、KWD ブロックをリソースに割り振ることができませんでした。
X'0000010C'	OM は、MUID ブロックをリソースに割り振ることができませんでした。
X'00000160'	OM は、リソース用にラッチを取得できませんでした。

関連資料:

183 ページの『CSLOMBLD: コマンド登録作成』

## CSLQMRSP: コマンド応答要求

コマンド処理クライアントはコマンドへの応答として、CSLQMRSP 要求を発行します。コマンド応答情報が、まとめて OM に送信されます。

コマンド処理クライアントは、コマンドへの応答として CSLQMRSP 要求を出します。個々のコマンド処理クライアントからのすべてのコマンド応答情報は、クライアントによって統合され、1 つの要求に入れられて OM に送られます。OM は、複数のクライアントからの応答を自動化操作プログラム・クライアント用に 1 つの応答に統合します。

### CSLQMRSP の構文

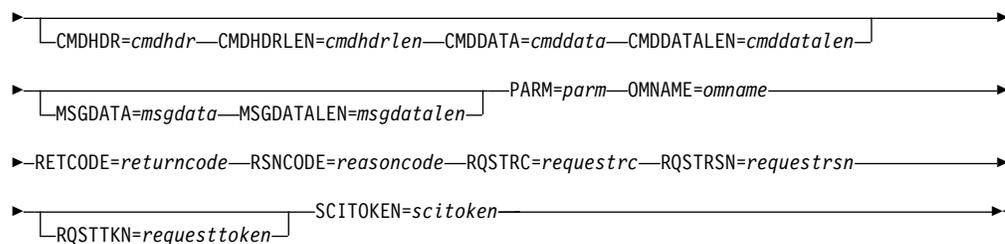
#### CSLQMRSP DSECT の構文

CSLQMRSP 要求の DSECT 機能は、CSLQMRSP パラメーター・リストの長さ、および戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶—CSLQMRSP—FUNC=DSECT—▶▶

#### CSLQMRSP 要求プロトコルの構文

▶▶—CSLQMRSP—FUNC=RESPOND—  
└─CMD=cmdinput—CMDLEN=cmdinputlen─┘ CMDTOKEN=cmdtoken—▶▶



## CSLOMRSP のパラメーター

**CMD=***symbol*

**CMD=(r2-r12)**

(オプション) -- コマンド入力バッファーを指定します。これは、OM API から指定できる任意の IMS コマンドが可能です。

このパラメーターはオプションです。ここで指定したものは、入力タグに入れられて、XML 出力として戻されます。

**CMDDATA=***symbol*

**CMDDATA=(r2-r12)**

(オプション) -- コマンド応答データ・バッファーを指定します。

**CMDDATALEN=***symbol*

**CMDDATALEN=(r2-r12)**

(オプション) -- コマンド応答データ・バッファーの長さを指定します。

**CMDHDR=***symbol*

**CMDHDR=(r2-r12)**

(オプション) -- コマンド応答ヘッダー・バッファーを指定します。

**CMDHDRLEN=***symbol*

**CMDHDRLEN=(r2-r12)**

(オプション) -- コマンド応答ヘッダー・バッファーの長さを指定します。

**CMDLEN=***symbol*

**CMDLEN=(r2-r12)**

(オプション) -- コマンド入力バッファーの長さを指定します。

**CMDTOKEN=***symbol*

**CMDTOKEN=(r2-r12)**

(必須) -- コマンド・トークンが入っている 32 バイト・フィールドを指定します。このトークンは、クライアントが処理したコマンド・インスタンスを一意的に識別します。コマンド・トークンは、OM コマンド・ディレクティブでクライアントに渡されます。トークンのアドレスは、OM コマンド・ディレクティブ・パラメーター・リストの フィールド ODIR\_CMDTKPTR 入れてクライアントに渡されます。

**MSGDATA=***symbol*

**MSGDATA=(r2-r12)**

(オプション) -- コマンド応答メッセージ・バッファーを指定します。

**MSGDATALEN=***symbol*

**MSGDATALEN=(r2-r12)**

(オプション) -- コマンド応答メッセージ・バッファーの長さを指定します。

**PARM=symbol**

**PARM=(r2-r12)**

(必須) -- SCI にパラメーターを渡すために要求が使用するストレージのアドレスを指定する、4 バイトの入力パラメーター。パラメーター・リストの長さは、ORSP\_PARMLN で定義されたパラメーター・リスト長 EQU 値と同じでなければなりません。

**OMNAME=symbol**

**OMNAME=(r2-r12)**

(必須) -- コマンド登録要求の送信先である 8 バイトの OM 名を指定します。

**RETCODE=symbol**

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。返される戻りコードについては、以下の表に説明があります。

戻りコードは、OM (CSLORMRSP) または SCI (CSLSCMSG または CSLSCRQS) から返されます。ECB が指定されている場合、ECB が通知されるまで、RETCODE は無効です。戻りコードの上位バイトの値は、戻りコードを作成したのが SCI (X'01') か OM (X'02') かを識別します。OM 戻りコードは、CSLORR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

**RQSTRC=symbol**

**RQSTRC=(r2-r12)**

(必須) -- コマンドの発信元に渡す理由コードを入れる 4 バイト・フィールドを指定します。この理由コードは、コマンド処理クライアントによって定義され、コマンドの結果を示します。非ゼロの理由コードは、コマンド応答の <cmderr> セクションに入れてクライアントに戻されます。

**RQSTRSN=symbol**

**RQSTRSN=(r2-r12)**

(必須) -- コマンドの発信元に渡す理由コードを入れる 4 バイト・フィールドを指定します。この理由コードは、コマンド処理クライアントによって定義され、コマンドの結果を示します。非ゼロの理由コードは、コマンド応答の <cmderr> セクションに入れてクライアントに戻されます。

**RQSTTKN=symbol**

**RQSTTKN=(r2-r12)**

(オプション) -- OM コマンド・ディレクティブでコマンド・プロセッサ・クライアントに渡された、32 バイトの要求トークンを指定します。このパラメーターは、CSLOMI と CSLOMCMD 要求の一方または両方で入力される、RQSTTKN1 および RQSTTKN2 フィールドを表します。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。OM 理由コードは、CSLORR で定義されています。SCI 理由コードは、CSLSRR で定義されています。返される理由コードについては、以下の表に説明があります。

**RSNTEXT=symbol**

**RSNTEXT=(r2-r12)**

(オプション) -- 理由テキスト・バッファのアドレスを入れる必要のあるレジスターを指定します。シンボルとして指定される場合、理由テキスト・バッファのラベルである必要があります。バッファは 2 バイトの長さフィールドと、その後続く理由テキストから成ります。このトークンにより、OM クライアントはコマンド応答で理由コードのテキスト記述を渡すことができます。

**RSNTEXTLEN=symbol****RSNTEXTLEN=(r2-r12)**

(オプション) -- 理由テキスト・バッファのアドレスを入れる必要のあるレジスターを指定します。シンボルとして指定される場合、理由テキスト・バッファのラベルである必要があります。バッファは 2 バイトの長さフィールドと、その後続く理由テキストから成ります。このトークンにより、OM クライアントはコマンド応答で理由コードのテキスト記述を渡すことができます。

**SCITOKEN=symbol****SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されます。

**CSLORMRSP の戻りコードと理由コード**

以下の表は、CSLORMRSP マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 48. CSLORMRSP の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。

関連資料:

291 ページの『第 9 章 CSL Operations Manager XML 出力』

**CSLORMSUB: 非送信請求メッセージへのサブスクライブ**

Single Point of Control (SPOC) クライアントは、IMS などのコマンド処理クライアントから非送信請求出力メッセージを受け取るために、CSLORMSUB 要求を使用して OM にサブスクライブします。

メッセージは、それが入力メッセージへの応答として生成されたものでない場合は、非送信請求と定義されます。例えば、システムの通知メッセージは非送信請求メッセージです。IMS バージョン 13 IMSplex 環境では、コマンド応答がコマンドの発信元でない SPOC に返される場合、そのコマンド応答は非送信請求メッセージと見なされます。

すべての非送信請求出力メッセージは、出力の変更が可能な OM の出力出口ルーチンに経路指定されます。SPOC は CSLORMSUB 要求を使用して OM へサブスクライブし、非送信請求出力メッセージを、変更済みのものもそうでないものも SPOC に経路指定されるようにできます。OM に送信すべきでない非送信請求出力メッセージを識別するために、OM、RM、SCI、CQS、および無視する必要のある IMS 制御領域からのメッセージを含むテーブルを編集できます。

非送信請求出力メッセージを受け取るためには、SCI 登録要求 (CSLSCREG) で SCI 入力出口ルーチンを指定しておく必要があります。

OM が終了しているかどうかを判断するには、SCI 登録要求で SCI 通知出口ルーチンを指定して RETNAME または RETTOKEN パラメーターを指定します。SCI 通知出口は、IMSpIex のいずれかのメンバーの状況に変更があったときに呼び出されます。SCI 登録からの RETNAME または RETTOKEN の値を保管して、OM が終了していることを示す一致があるかどうかを調べるために使用します。そうすると、SPOC クライアントは別の OM ヘサブスクライブすることができます。

この要求はアセンブラーおよび PLX でサポートされます。

## CSLOMSUB の構文

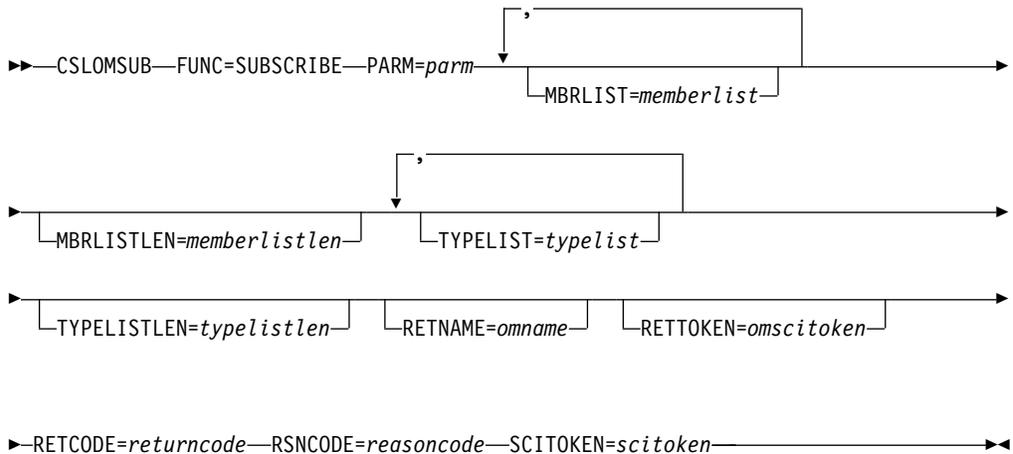
### CSLOMSUB FUNC-DSECT の構文

CSLOMSUB 要求の DSECT 機能を使用して、CSLOMSUB パラメーター・リストの長さ、および戻りコードと理由コード用にロジックの中に等価 (EQU) ステートメントを組み込みます。

▶▶CSLOMSUB—FUNC—=—DSECT—▶▶

### CSLOMSUB 要求の構文

この構文では、SPOC クライアントは非送信請求出力メッセージを受け取るために OM ヘサブスクライブできます。



## CSLOMSUB 要求のパラメーター

**PARAM=parm**

**PARAM=(r1-r12)**

CSLOMSUB パラメーター・リストを指定します。パラメーター・リストの長さは、OSUB\_PARMLN で定義されたパラメーター・リスト長等価 (EQU) 値と同じでなければなりません。

**MBRLIST=memberlist**

**MBRLIST=(r1-r12)**

IMSpIex 内の、受け取る非送信請求出力メッセージの発信元のコマンド処理クライアント (IMS 制御領域など) を識別するメンバー・リストを指定します。TYPELIST パラメーターを指定した場合、MBRLIST= パラメーターは指定しないでください。TYPELIST パラメーターと MBRLIST パラメーターのいずれも指定しない場合、デフォルトのメンバー・リストは IMSpIex 内のすべてのコマンド処理クライアントになります。

クライアント名を区切るには、コンマを使用してください。

**MBRLISTLEN=memberlistlen**

**MBRLISTLEN=(r1-r12)**

MBRLIST パラメーターで指定されたメンバー・リストの長さを指定します。

**TYPELIST=typelist**

**TYPELIST=(r1-r12)**

受け取る非送信請求出力メッセージの発信元である、有効な IMSpIex メンバー・タイプを識別するタイプ・リストを指定します。MBRLIST パラメーターを指定した場合、TYPELIST パラメーターは指定しないでください。有効な IMSpIex メンバー・タイプについては、273 ページの『CSLSCREG: 登録要求』の TYPE= パラメーターの説明を参照してください。TYPELIST パラメーターと MBRLIST パラメーターのいずれも指定しない場合、デフォルトのリストは IMSpIex 内のすべてのコマンド処理クライアントになります。

クライアント名を区切るには、コンマを使用してください。

**TYPELISTLEN=typelistlen**

**TYPELISTLEN=(r1-r12)**

TYPELIST= パラメーターで指定されたタイプ・リストの長さを指定します。

**RETNAME=omname**

**RETNAME=(r1-r12)**

サブスクリプション要求の送信先の OM の名前を受け取る 8 バイト出力フィールドを指定します。

**RETTOKEN=omscitoken**

**RETTOKEN=(r1-r12)**

呼び出し元に戻される OM SCI トークンを受け取る 16 バイト出力フィールドを指定します。これは、SCI の要求送信先のターゲット OM アドレス・スペースの SCI トークンです。

**RETCODE=returncode**

**RETCODE=(r1-r12)**

出力の戻りコードを受け取る 4 バイト・フィールドを指定します。OM 戻りコードは、CSLORR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

OM または SCI が戻りコードを作成します。戻りコードの上位バイトの値は、戻りコードを作成したのが SCI (X'01') か OM (X'02') かを識別します。

**RSNCODE=reasoncode**

**RSNCODE=(r1-r12)**

出力の理由コードを受け取る 4 バイト・フィールドを指定します。OM 戻りコードは、CSLORR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

**SCITOKEN=scitoken****SCITOKEN=(r1-r12)**

この SCI への接続を一意的に識別する SCI トークンが入っている 16 バイト・フィールドを指定します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されます。

**CSLOMSUB の戻りコードと理由コード**

以下の表は、CSLOMSUB 要求で返される戻りコードと理由コードの一覧です。

表 49. CSLOMSUB 要求の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'02000014'	X'00005044'	OSUB ブロックを取得できません。
	X'00005048'	メンバー・リスト用のストレージを獲得できません。
	X'0000504C'	タイプ・リスト用のストレージを獲得できません。

**使用上の注意**

SPOC が非送信請求出力メッセージを受け取るために OM にサブスクライブする場合、デフォルトで、すべてのメッセージがそのサブスクライバーに送信されるように設定されます。IMS、CQS、および CSL メッセージが非送信請求出力として送信されないようにするには、以下のマクロを使用して、メッセージ・テーブルにメッセージを追加し、保守してください。

## マクロ 説明

**CSLZUMT**

CSLZUMTU テーブルに、サブスクライバーに非送信請求メッセージとして経路指定しない CSL メッセージを追加するには、このマクロを使用します。

**CQSUOMT**

CQSUOMTU テーブルに、サブスクライバーに非送信請求メッセージとして経路指定しない CQS メッセージを追加するには、このマクロを使用します。

**DFSUOMT**

DFSUOMTU テーブルに、サブスクライバーに非送信請求メッセージとして経路指定しない IMS メッセージを追加するには、このマクロを使用します。

CQSUOMTU、CSLZUMTU、および DFSUOMTU テーブルの例は、IMS サンプル・ライブラリーに提供されています。

以下の例に、CSLZUMT マクロの例を示します。この例では、メッセージ CSL2020I および CSL2021E を CSLZUMTU テーブルに追加し、それによって、これらのメッセージが非送信請求出力メッセージとしてサブスクライブしたクライアントに送信されないようにします。

```
CSLZUMT MESSAGE=CSL2020I
CSLZUMT MESSAGE=CSL2021E
```

CQSUOMTU、CSLZUMTU、および DFSUOMTU テーブルには、MESSAGE=SUPPRESS を指定することもできます。例えば、メッセージ・テーブル DFSUOMTU に以下のコードを作成すると、DFSxxxxx および DFSyyyyy は OM に送信され、メッセージ DFSzzzzz は OM に送信されません。また、MESSAGE=SUPPRESS ステートメントにより、その他すべてのメッセージは OM に送信されません。

```
DFSUOMT MESSAGE=DFSxxxxx,SEND=YES
DFSUOMT MESSAGE=DFSyyyyy,SEND=YES
DFSUOMT MESSAGE=DFSzzzzz,SEND=NO
DFSUOMT MESSAGE=SUPPRESS
```

IMS から OM に送信される非送信請求出力メッセージを制御するもう 1 つの方法は、DFSDFxxx および DFSCGxxx IMS PROCLIB メンバーに、新規 UOM=MTO | NONE | ALL パラメーターを使用することです。

- UOM=MTO を指定すると、MTO のみ、またはシステム・コンソールのみ (またはその両方) を宛先とするメッセージが、OM に送信されます。この場合、エンド・ユーザー端末などの他の宛先に送信されるメッセージは、OM に送信されません。
- UOM=NONE を指定すると、どのメッセージも OM に送信されません。
- UOM=ALL を指定すると、すべてのメッセージが OM に送信されます。

---

## CSL OM USB: 非送信請求メッセージからのアンサブスクライブ

SPOC クライアントは CSL OM USB 要求を使用して OM からアンサブスクライブし、IMS などのコマンド処理クライアントからの非送信請求出力メッセージの受け取りを停止することができます。

この要求はアセンブラーおよび PLX でサポートされます。

### CSL OM USB の構文

#### CSL OM USB FUNC=DSECT の構文

CSL OM USB 要求の DSECT 機能を使用して、CSL OM USB パラメーター・リストの長さ、および戻りコードと理由コード用にロジックの中に等価 (EQU) ステートメントを組み込みます。

▶▶ CSL OM USB—FUNC=DSECT—◀◀

#### CSL OM USB 要求の構文

この構文では、SPOC クライアントは非送信請求出力メッセージの受け取りを停止するために OM からアンサブスクライブできます。

▶▶—CSL0MUSB—FUNC=UNSUBSCRIBE—PARM=*parm*—RETCODE=*returncode*————▶▶

▶—RSN0CODE=*reasoncode*—SCITOKEN=*scitoken*————▶▶

## CSL0MUSB パラメーター

**PARM=*parm***

**PARM=(*r1-r12*)**

CSL0MUSB パラメーター・リストを指定します。パラメーター・リストの長さは、OUSB\_PARMLN で定義されたパラメーター・リスト長等価 (EQU) 値と同じでなければなりません。

**RETCODE=*returncode***

**RETCODE=(*r1-r12*)**

出力の戻りコードを受け取る 4 バイト・フィールドを指定します。OM 戻りコードは、CSLORR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

OM または SCI が戻りコードを作成します。戻りコードの上位バイトの値は、戻りコードを作成したのが SCI (X'01') か OM (X'02') かを識別します。

**RSN0CODE=*reasoncode***

**RSN0CODE=(*r1-r12*)**

出力の理由コードを受け取る 4 バイト・フィールドを指定します。OM 戻りコードは、CSLORR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。

**SCITOKEN=*scitoken***

**SCITOKEN=(*r1-r12*)**

この SCI への接続を一意的に識別する SCI トークンが入っている 16 バイト・フィールドを指定します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されます。

## CSL0MUSB の戻りコードと理由コード

以下の表に示した戻りコードと理由コードは、CSL0MUSB 要求で戻されます。

表 50. CSL0MUSB 要求の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。

## CSL OM ディレクティブ

OM ディレクティブは、OM が定義する機能で、OM クライアントに処理対象の作業を通知するためのメッセージとして送信できます。OM にコマンドを登録したコマンド処理クライアントは、選択されて、OM ディレクティブを実行することができます。

OM ディレクティブは、常にメッセージ・プロトコル (PROTOCOL=MSG) で、すなわち、非同期で発行されます。したがって、OM は OM クライアントからの応

答を期待せず、応答を待たずに処理を続行します。OM クライアントは、ディレクティブに対してアクションをとるべきかどうかを自分で決定しなければなりません。

クライアントが `PROTOCOL=MSG` を発行すると、SCI は、OM からの XML 出力をクライアントの SCI 入力出口に送ります。SCI 入力出口ルーチンのフィールド `INXP_MBRPLPTR` は、`CSLOMDIR` パラメーター・リストを指し示しています。

クライアントが `CSLOMI PROTOCOL=RQST` を出した場合、OM からの XML 出力ストリームは、`OUTPUT=` パラメーターのクライアントに直接送信されます。

SCI 入力出口ルーチンは、クライアントにディレクティブのことを知らせる責任があります。クライアントは、各自の SCI 入力出口ルーチンを、OM ディレクティブをサポートするようにコーディングする必要があります。クライアントの責任で、機能および機能コードをどこに定義するかを決めてください。クライアントは、`CSLOMDIR` パラメーター・リストを使い終えたら、`CSLSCBFR` を発行してストレージを解放しなければなりません。

OM ディレクティブは `CSLOMDIR` マクロ内で定義されます。次のような OM ディレクティブがあります。

- コマンド・ディレクティブ (`ODIR_CMDD`)
- `CSLOMI` 応答ディレクティブ (`ODIR_OMIRESPD`)
- コマンド応答ディレクティブ (`ODIR_CMDRESPD`)
- 照会応答ディレクティブ (`ODIR_QRYRESPD`)

ディレクティブとそのパラメーターについて、以下で説明します。

## CSL OM コマンド・ディレクティブ

OM コマンド・ディレクティブ `ODIR_CMDD` は、処理するコマンドがあるときにコマンド処理クライアントに送信されます。

OM コマンド・ディレクティブのパラメーターは次のとおりです。このパラメーターは、SCI 入力出口に渡されます。

### **ODIR\_COMMAND**

コマンド・ディレクティブの先頭を識別します。

### **ODIR\_CMDTKLEN=length**

OM コマンド・トークンの長さが入っています。これは、コマンド・インスタンスを識別するために OM で使用されるだけです。

### **ODIR\_CMDTKPTR=address**

OM コマンド・トークンのアドレスが入っています。

### **ODIR\_INPUTLEN=length**

ユーザーが入力するコマンド入力ストリングの長さが入っています。

### **ODIR\_INPUTPTR=address**

コマンド入力ストリングのアドレスが入っています。

**ODIR\_VERBLEN=length**

正規形のコマンド verb の長さが入っています。

**ODIR\_VERBPTR=address**

コマンド verb のアドレスが入っています。

**ODIR\_KWDLEN=length**

コマンド・キーワードの長さが入っています。

**ODIR\_KWDPTR=address**

コマンド・キーワードのアドレスが入っています。

**ODIR\_PARSELEN=length**

構文解析されたコマンド・ブロックの長さが入っています。

**ODIR\_PARSEPTR=address**

構文解析されたコマンド・ブロックのアドレスが入っています。

**ODIR\_CUIDLEN=length**

コマンドを発信したユーザー ID の長さが入っています。

**ODIR\_CUIDPTR=address**

コマンドを発信したユーザー ID のアドレスが入っています。

**ODIR\_CNAMELEN=length**

コマンドを発信したクライアントの名前 (SCI に登録された名前) の長さが入っています。

**ODIR\_CNAMEPTR=address**

コマンドを発信したクライアントの名前 (SCI に登録された名前) のアドレスが入っています。

**ODIR\_CTYPE=client type**

コマンドを発信したクライアントのタイプが入っています。これは、SCI に定義された TYPE= パラメーターからの値です。このパラメーターは、値で渡されます。長さフィールドは常にゼロです。

**ODIR\_CSTYPLEN=length**

コマンドを発信したクライアントのサブタイプが入っています。これは、SCI に定義された SUBTYPE= パラメーターからの値です。

**ODIR\_CSTYPPTR=length**

コマンドを発信したクライアントのサブタイプのアドレスが入っています。

**ODIR\_CFLAGS=flags**

OM コマンド処理フラグが入っています。これらのパラメーターは、値で渡されます。長さフィールドは常にゼロです。

**ODIR\_CRQTKLEN=length**

ユーザー要求トークンの長さが入っています。このパラメーターは、コマンド・インスタンスを識別するために、コマンドを発信したプログラムで使用されるだけです。

**ODIR\_CRQTKPTR=address**

ユーザー要求トークンのアドレスが入っています。このパラメーターは、コマンド・インスタンスを識別するために、コマンドを発信したプログラムで使用されるだけです。

**ODIR\_TIMEOUT=timeoutvalue**

コマンドで指定されたコマンド・タイムアウト値が入っています。このパラメーターは、値で渡されます。長さフィールドは常にゼロです。

**ODIR\_CMDLN**

コマンド・ディレクティブ長 EQU。

**CSL OM 応答ディレクティブ**

CSLOMDIR には、3 つの応答ディレクティブがあります。

## • CSLOMI 応答 (ODIR\_OMIRESPD)

CSLOMI 応答ディレクティブは、CSLOMI 呼び出しに関する応答をクライアントに戻します。エラーが起きたときに応答が送信され、その応答が CSLOMI CMD に関するものなのか、CSLOMI QUERY 呼び出しに関するものなのか明確ではありません。

## • コマンド応答 (ODIR\_CMDRESPD)

コマンド応答ディレクティブは、CSLOMI CMD または CSLOMCMMD 呼び出しの結果としてのコマンド応答をクライアントに戻します。

## • 照会応答 (ODIR\_QRYRESPD)

照会応答ディレクティブは、CSLOMI QUERY または CSLOMQRV 呼び出しの結果としての照会応答をクライアントに戻します。

OM 応答ディレクティブのパラメーターは同一です。

**ODIR\_CQRESP**

コマンド応答または照会応答の先頭を識別します。

**ODIR\_CQRSPRC=returncode**

コマンド応答または照会応答の戻りコードが入っています。

**ODIR\_CQRSPRSN=reasoncode**

コマンド応答または照会応答の理由コードが入っています。

**ODIR\_CQXMLEN=length**

戻される XML 出力の長さが入っています。

**ODIR\_CQXMLPTR=address**

戻される XML 出力のアドレスが入っています。

**ODIR\_CQRT1LEN=length**

要求トークン 1 (RQSTTKN1) の長さが入っています。

**ODIR\_CQRT1PTR=address**

要求トークン 1 (RQSTTKN1) のアドレスが入っています。

**ODIR\_CQRT2LEN=length**

要求トークン 2 (RQSTTKN2) の長さが入っています。

**ODIR\_CQRT2PTR=address**

要求トークン 2 (RQSTTKN2) のアドレスが入っています。

**ODIR\_CQRSPLN**

応答ディレクティブ長 EQU。

## CSL UOM ディレクティブ

OM 非送信請求出力メッセージ (UOM) ディレクティブ ODIR\_UOM は、OM へサブスクライブするすべての OM クライアントに送信されます。このディレクティブは、CSLOMDIR で定義します。

UOM ディレクティブのパラメーターは OM クライアントの SCI 入力出口ルーチンに渡されます。

### ODIR\_UOM

UOM ディレクティブ・パラメーターの先頭を識別します。

#### ODIR\_UOMXMLLEN=*length*

送られる XML 出力の長さが入っています。

#### ODIR\_UOMXMLPTR=*address*

XML タグで囲まれた、非送信請求出力メッセージのアドレスが入っています。

#### ODIR\_UOMLN

UOM ディレクティブ長の等価 (EQU) 値。

### ODIR\_UOMCR

コマンド応答 UOM ディレクティブ・パラメーターの先頭を識別します。このパラメーターは、サブスクライブしているクライアントに、このコマンド応答が非送信請求メッセージとして送信されたことを通知します。

#### ODIR\_UOMXMLLEN=*length*

送られる XML 出力の長さが入っています。

#### ODIR\_UOMXMLPTR=*address*

XML タグで囲まれた、非送信請求出力メッセージのアドレスが入っています。

#### ODIR\_UOMLN

UOM ディレクティブ長の等価 (EQU) 値。

関連資料:

140 ページの『CSLOMI: API 要求』

152 ページの『CSLOMQRY: 照会要求』

 BPE ベースの CSL SCI ユーザー出口ルーチン (出口ルーチン)

## 第 7 章 CSL RM クライアントの作成

このセクションのトピックでは、RM クライアントを作成するためのクライアント要求とディレクティブについて説明します。

### RM クライアント要求の順序

RM を使用して、自分のプロダクトまたはサービス用に IMSplex 内のグローバル・リソースを管理したい場合、RM クライアントを 1 つ以上作成する必要があります。RM クライアントは、RM 要求を特定の順序で発行して、RM と通知します。

RM クライアントを作成するには、RM で提供される一連のクライアント要求を使用します。この要求を使用することによって、クライアントは、RM またはリソース構造上のリソースにアクセスしたり、全 IMSplex プロセスを調整することができます。RM クライアントの 1 例は、IMS です。RM クライアントはアセンブラ言語で作成することができます。

RM クライアントは、RM 要求を使用して RM のサービスおよびリソースを使用します。クライアントは、SCI および RM 要求を出して RM サービスを要求します。一部の要求は、特定の順序で出さなければなりません。そのほかの要求は、クライアントの処理要件に基づいて、複数回、どのような順序でも出すことができます。

RM クライアントは、RM 要求を発行する前に、次のような登録が必要です。

- SCI に登録する。
- IMSplex 内の各アクティブ RM に登録する。これにより、どの RM でも、RM 要求を処理できるようになります。
- その固有のリソース・タイプおよび関連の名前タイプを RM に登録する。

以下の表に、RM クライアントが発行する要求の順序を示します。要求の目的も併記します。

表 51. RM クライアントの要求のシーケンス

要求	目的
CSLSCREG	SCI に登録する。これにより、クライアントは、SCI を介して RM に RM 要求を送信できます。
CSLSCRDY	SCI に対して RM クライアントを作動可能にする。これにより、メッセージは、クライアント・タイプ別にクライアントへ経路指定されます。
CSLRMREG	クライアントを RM に登録して、RM との通信を可能にする。クライアントは IMSplex 内の各アクティブ RM に登録しなければなりません。これにより、どの RM でも、RM 要求を処理できるようになります。クライアントは、その固有のリソース・タイプおよび関連の名前タイプを RM に登録しなければなりません。

表 51. RM クライアントの要求のシーケンス (続き)

要求	目的
CSLRMxxx	CSLRMUPD、CSLRMDEL、CSLRMQRY などの RM リソース要求を出して、リソース構造上のリソースを操作する。
CSLRMPxx	CSLRMPRI、CSLRMPRS、CSLRMPRR、CSLRMPRT などの RM 要求を出して、全 IMSplex プロセスに参加する。
CSLSCBFR	要求によって出力バッファが戻された場合、その出力バッファを解放する。
CSLRMDRG	RM からクライアントを登録解除して、RM との通信を終了させる。
CSLSCDRG	SCI への登録を抹消します。

以下の表に、全 IMSplex プロセスに参加している RM クライアントが発行する要求の順序を示します。要求の目的も併記します。

表 52. 全 IMSplex プロセスに参加している RM クライアントの要求のシーケンス

要求	目的
CSLRMPRI	全 IMSplex プロセスを開始する。
CSLRMPRS	全 IMSplex プロセス内のステップを処理する。プロセスには、プロセス・ステップはゼロ個、1 個、または複数個あります。プロセス・ステップを開始するクライアントは、そのステップのマスターです。
CSLRMPRR	プロセス・ステップに応答する。
CSLRMPRT	全 IMSplex プロセスを終了させる。

関連資料:

237 ページの『CSLRMREG: クライアントを登録する』

## グローバル・リソースを管理するための CSL RM 要求の発行

クライアントがグローバル・リソース情報にアクセスしたり、この情報を変更したりするには、事前に CSLSCREG 要求を使用して SCI への登録を行う必要があります。SCI に登録した後、CSLRMREG 要求を使って RM に登録しなければなりません。クライアントは、通信する予定の個々の IMSplex に関して SCI 登録要求を発行する必要があります。

クライアントがグローバル・リソース情報にアクセスしたり、この情報を変更したりするには、事前に CSLSCREG 要求を使用して SCI への登録を行う必要があります。

SCI に登録した後、CSLRMREG 要求を使って RM に登録しなければなりません。

クライアントは、終了の準備ができたなら、CSLRMDRG 要求を使って RM から登録を解除し、CSLSCDRG 要求を使って SCI から登録を解除する必要があります。

関連資料:

273 ページの『CSLSCREG: 登録要求』

210 ページの『CSLRMDEL: リソースの削除』

- 215 ページの『CSLRMDRG: クライアントを登録解除する』
- 216 ページの『CSLRMPRI: プロセス開始』
- 219 ページの『CSLRMPRR: プロセス応答』
- 221 ページの『CSLRMPRS: プロセス・ステップ』
- 228 ページの『CSLRMPRT: プロセス終了』
- 231 ページの『CSLRMQRY: リソース照会』
- 237 ページの『CSLRMREG: クライアントを登録する』
- 241 ページの『CSLRMUPD: リソースを更新する』

---

## 全 IMSplex プロセスを調整するための CSL RM 要求の発行

RM 提供の要求を使って、全 IMSplex プロセスを調整することができます。プロセスに参加するすべてのクライアントは、RM クライアント登録要求 (CSLRMREG) を使って RM に登録します。クライアントが登録された後、いくつかの異なる要求を使用して、プロセスを調整できます。

1 つのクライアントが、RM プロセス開始要求 (CSLRMPRI) を使ってプロセスを開始します。同じクライアントまたは別のクライアントが、RM プロセス・ステップ要求 (CSLRMPRS) を使ってステップを開始します。開始するクライアントは、そのステップの マスター と呼ばれます。1 つの RM がその要求を処理し、他のクライアントに、プロセス・ステップを実行するようという RM ディレクティブを送ります。他のすべてのクライアントは、そのステップを処理し、出力を作成し、RM プロセス応答要求 (CSLRMPRR) を使ってステップに応答します。RM は、全クライアントからの応答を 1 つの出力に統合し、その出力をプロセス・ステップのマスターに戻します。プロセス内に、まだステップがある場合、いずれかのクライアントがステップを開始し、他のクライアントが処理を行い、応答します。いずれかのクライアントが、RM プロセス終了要求 (CSLRMPRT) を使ってプロセスを終了させます。必要ならば、クライアントは、RM クライアント登録解除要求 (CSLRMDRG) を使って登録解除することができます。

障害の種類によっては、RM は全 IMSplex プロセスに関する全知識を失います。そのような障害としては、リソース構造 (複製があれば、複製との両方) の障害と全 RM 障害があります。このタイプの障害が発生した場合、各 RM クライアントはローカルでプロセスの知識をクリーンアップしなければならず、マスター RM はプロセスを終了しなければなりません。問題を検出した最初の RM クライアントは、強制オプションを指定した CSLRMPRS 要求を発行することにより、クリーンアップ・プロセス・ステップを開始して、エラーのタイプに関係なく、RM がクリーンアップ・プロセス・ステップを強制することができるようにします。このプロセス・ステップに参加しているクライアントは、プロセスをローカルにクリーンアップします。次に、このプロセス・ステップのマスターは、CSLRMPRT 要求でプロセスを終了します。

全 IMSplex プロセスを調整するには、CSLRMPRI、CSLRMPRR、CSLRMPRS、および CSLRMPRT 要求を使用できます。

関連資料:

- 216 ページの『CSLRMPRI: プロセス開始』
- 219 ページの『CSLRMPRR: プロセス応答』

## CSLRMDEL: リソースの削除

CSLRMDEL 要求を発行することで、リソース構造上の 1 つ以上の固有な名前付きリソース、あるいは所有者ごとに特定のリソース・タイプのすべてのリソースを削除できます。

この要求はアセンブラー言語でサポートされます。

### CSLRMDEL 構文

#### CSLRMDEL DSECT の構文

CSLRMDEL 要求の DSECT 機能は、プログラムに次のリソースを組み込みます。

- CSLRMDEL パラメーター・リスト長の等価 (EQU) ステートメント
- CSLRMDEL の戻りコード、理由コード、および完了コード
- 入力削除リストをマップするための CSLRDELL DSECT
- 削除出力をマップするための CSLRDELO DSECT

▶▶CSLRMDEL—FUNC=DSECT—▶▶

#### CSLRMDEL DELETE の構文

CSLRMDEL 要求の DELETE 機能は、リソース構造上の 1 つ以上の固有な名前付きリソースを削除します。

▶▶CSLRMDEL—FUNC=DELETE—PARM=*parm*—LIST=*deletelist*—▶▶

▶—LISTLEN=*deletelistlength*—OUTPUT=*output*—OUTLEN=*outputlength*—▶

▶—ECB=*ecb*—RETNAME=*returnname*—▶

▶—RETTOKEN=*returntoken*—RETCODE=*returncode*—▶

▶—RSNCODE=*reasoncode*—SCITOKEN=*scitoken*—▶▶

### CSLRMDEL パラメーター

**ECB=*symbol***

**ECB=(*r2-r12*)**

(オプション) - 非同期要求に使用される z/OS ECB のアドレスを指定します。要求が完了したとき、指定された ECB は通知されます。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、要求の起動側は、CSLRMDEL から制御を受け取った後で、この要求から戻されたデータ (RETCODE および RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を出す必要があります。

**LIST=symbol****LIST=(r2-r12)**

(必須) -- 呼び出し元が作成した削除リソース・リストを指定します。各リスト項目が、それぞれ別々の削除要求です。リスト長は、リスト項目の数に応じて異なります。

CSLRDELL は、削除リソース・リスト項目をマップします。そのリストには、ヘッダーと 1 つ以上のリスト項目が含まれます。リスト項目は、連続するストレージに置かなければなりません。各削除リスト項目には、削除対象についての情報が入っています。

リソース名による削除で、固有な名前付きリソースを削除する場合は、以下のようになります。

- リソース名 -- リソースのクライアント定義の名前。
- リソース・タイプ -- リソース構造上のリソースのクライアント定義の物理的グループ化。有効な値は 1 から 255 です。
- バージョン -- リソース・バージョンであって、リソースが更新された回数。

所有者による削除で、あるリソース・タイプについて、リソース・バージョンにかかわらず特定の所有者が所有するリソースをすべて削除する場合は、以下のようになります。

- リソース・タイプ -- リソース構造上のリソースのクライアント定義の物理的グループ化。有効な値は 1 から 255 です。
- 所有者 - リソース所有者。

**LISTLEN=symbol****LISTLEN=(r2-r12)**

(必須) -- 4 バイトの削除リソース・リスト長を指定します。

**OUTLEN=symbol****OUTLEN=(r2-r12)**

(必須) -- CSLRMDEL 要求から戻される出力の長さを受け取る 4 バイト・フィールドを指定します。OUTLEN には、OUTPUT= パラメーターで示された出力の長さが入っています。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力長はゼロです。

**OUTPUT=output****OUTPUT=(r2-r12)**

(必須) -- CSLRMDEL 要求から戻される可変長出力のアドレスを受け取る 4 バイト・フィールドを指定します。その出力には、試みられたリソース削除に対するヘッダーと 1 つ以上の削除項目が含まれます。出力長は、OUTLEN= フィールドに入れて戻されます。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力アドレスはゼロです。

CSLRDELO マクロは、戻される出力をマップします。その出力には、ヘッダーと 1 つ以上のリスト項目が含まれます。

出力ヘッダーの内容は次のパラメーターが含まれています。

- 目印

- 出力長
- CSLRDELO のバージョン
- CSLRDELO のヘッダーの長さ (項目の先頭までのオフセット)
- CSLRDELO 項目の長さ
- リソース項目カウント

各出力項目は、失敗したリソース削除を表します。各項目には以下のパラメーターが含まれています。

- 出力項目長 -- リスト項目長
- 名前タイプ -- 名前タイプ内のクライアント定義リソース名の一意性を保証する、リソース・タイプに関連付けられた、クライアント定義の値。有効な値は 1 から 255 です。
- リソース名
- リソース・タイプ
- 削除タイプ
- バージョン -- バージョン不一致のために削除要求が失敗した場合、既存リソースのリソース・バージョン。
- 所有者 -- バージョン不一致のために削除が失敗し、所有者を読み取るオプションが設定されていた場合、既存リソースのリソース所有者。
- 削除要求の完了コード。完了コードは、CSLRRRR によってマップされます。

完了コードには、以下のものがあります。

**X'00000008'**

リソース・タイプが無効。

**X'00000010'**

バージョン不一致。入力に指定されたバージョンはリソースのバージョンと一致しておらず、そのため、削除は失敗します。

**X'00000018'**

リソース・タイプは登録されません。リソース・タイプは、CSLRMREG 要求を使って登録しなければなりません。

**X'00000024'**

リソース構造が利用不能。

**X'00000038'**

CQS 内部エラーのために、削除が失敗しました。

**X'0000003C'**

RM が作成した CQSDEL リスト項目が正しくなかったため、削除は失敗しました。

出力バッファは、呼び出し元によって事前割り振りされることはありません。要求からこのワードが戻された場合、そこには、削除出力が入ったバッファのアドレスが入っています。ストレージの使用が終了したとき、呼び出し元は、CSLSCBFR FUNC=RELEASE 要求を出すことにより、このストレージを解放する必要があります。出力の長さは、OUTLEN= フィールドに入れて戻されます。

**PARM=symbol**

**PARM=(r2-r12)**

(必須) -- CSLRMDEL パラメーター・リストを指定します。パラメーター・リストの長さは、RDEL\_PARMLN でパラメーター・リストの長さとして EQU 値と同じでなければなりません。

**RETCODE=symbol****RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。RM 戻りコードは、CSLRRR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。返される戻りコードについては、以下の表に説明があります。

**RETNAME=symbol****RETNAME=(r2-r12)**

(オプション) -- 呼び出し元に戻される RM 名を受け取る 8 バイト・フィールドを指定します。これは、SCI の要求送信先のターゲット RM アドレス・スペースの CSL メンバー名です。

**RETTOKEN=symbol****RETTOKEN=(r2-r12)**

(オプション) -- 呼び出し元に戻される RM の SCI トークンを受け取る 16 バイト・フィールドを指定します。これは、SCI の要求送信先のターゲット RM アドレス・スペースの SCI トークンです。

**RSNCODE=symbol****RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。RM 理由コードは、CSLRRR で定義されています。SCI 理由コードは、CSLSRR で定義されています。返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol****SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

**CSLRMDEL の戻りコードと理由コード**

以下の表は、CSLRMDEL 要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 53. CSLRMDEL の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'03000008'	X'00002000'	クライアントは登録されません。
	X'00002100'	削除リスト長が無効。
	X'00002108'	削除リスト・アドレスが無効。

表 53. CSLRMDEL の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'00002110'	リスト・ヘッダー (DELL_PVER) 内のバージョンがゼロであって、これは無効です。リスト・ヘッダー内のリスト・バージョンは、最大リスト・バージョン (DELL_PVERMAX) に設定されていなければなりません。
	X'00002114'	リスト・ヘッダー長が無効。リスト・ヘッダー長は、ゼロ、または渡されたリスト長より大きくてはなりません。リスト・ヘッダー内のリスト・ヘッダー長 (DELL_HDRLEN) は、リスト・ヘッダー長に設定されていなければなりません。
	X'00002200'	リスト項目の 1 つに、ゼロなどの無効なリソース・タイプが入っています。RM は、リストの残り部分は無効であると想定します。
	X'0000220C'	リスト項目の 1 つに、1 つ以上の無効な削除オプションが入っています。RM は、リストの残り部分は無効であると想定します。
	X'00002210'	リソース名または所有者は必須です。
	X'00002214'	バージョンが無効。
	X'00002404'	リソース構造が定義されていない。
X'0300000C'	X'00003000'	要求は少なくとも 1 つのリスト項目については成功しましたが、すべてについて成功したわけではありません。OUTPUT バッファ内の各リスト項目の完了コードを検査して、個々のエラーを調べてください。
	X'00003004'	要求はすべてのリスト項目について失敗しました。OUTPUT バッファ内の各リスト項目の完了コードを検査して、個々のエラーを調べてください。
	X'00003008'	要求は 1 つ以上のリスト項目について失敗し、すべての失敗はバージョン不一致でした。OUTPUT バッファ内の各リスト項目の完了コードを検査して、個々のエラーを調べてください。
X'03000010'	X'00004000'	CQS アドレス・スペースが利用不能。要求を再試行してください。再試行では、使用可能な CQS のある、別の RM に要求を経路指定しようとしています。
	X'00004100'	要求されたバージョンはサポートされていません。クライアントがコンパイルした CSLRMDEL のバージョンは、RM でサポートされていません。新しい CSLRMDEL 機能を使用する新しいリリースに IMS をマイグレーションする前に、すべての RM を新しいリリースにマイグレーションしなければなりません。
	X'00004104'	リストのバージョンはサポートされていません。クライアントが削除リストを作成したバージョンは、RM でサポートされていません。新しい CSLRMDEL 機能を使用する新しいリリースにクライアントをマイグレーションする前に、すべての RM を新しいリリースにマイグレーションしなければなりません。

表 53. CSLRMDEL の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'03000014'	X'00005000'	削除出力バッファのストレージ割り振りが失敗しました。
	X'00005120'	CQSDEL バッファのストレージ割り振りが失敗しました。
	X'00005200'	CQS 要求が予期しないエラーになりました。
	X'00005204'	RM が作成した要求入力为正しくなかったため、CQS 要求は失敗しました。

関連概念:

208 ページの『グローバル・リソースを管理するための CSL RM 要求の発行』

## CSLRMDRG: クライアントを登録解除する

登録解除要求は、それ以降、クライアントが RM からのリソース要求または全 IMSplex プロセスを処理したくない場合、クライアントから出されます。登録解除要求は、RM からクライアント情報を除去し、RM からクライアントへの新規リソース要求の送信を止めさせます。クライアントに関する一部の情報は保持され、全 IMSplex プロセスに影響があるかもしれません。

この要求は、IMS 制御領域などのリソース処理クライアントによって発行されます。

この要求はアセンブラでサポートされます。

### CSLRMDRG の構文

CSLRMDRG DSECT の構文

CSLRMDRG Deregister の構文

CSLRMDRG 要求の DEREGISTER 機能は、RM から登録解除します。

```

▶▶—CSLRMDRG—FUNC=DEREGISTER—PARM=parm—┬──────────────────────────────────┐
└──────────────────────────────────┘ OPTWORD1=deregisteroptions
▶—RETCODE=returncode—RSNCODE=reasoncode—SCITOKEN=scitoken──────────────────▶▶
    
```

### CSLRMDRG のパラメーター

**OPTWORD1=***symbol*

**OPTWORD1=(***r2-r12***)**

(オプション) - 登録解除オプションが入っている 4 バイト・フィールドを指定します。CSLRMDRG FUNC=DSECT は、登録解除オプション用の等価ステートメントを生成します。

**X'80000000'**

IMSplex からクライアントを除去してください。クライアントの全知識を削除してください。

**PARM=***symbol*

**PARM=(r2-r12)**

(必須) -- CSLRMDRG パラメーター・リストを指定します。パラメーター・リストの長さは、RDRG\_PARMLN でパラメーター・リストの長さとして EQU 値と同じでなければなりません。

**RETCODE=***symbol*

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。RM 戻りコードは、CSLRRR で定義されています。RM は、CSLRMDRG 要求に対して応答を戻しません。

**RSNCODE=***symbol*

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。SCI 理由コードは、CSLSRR で定義されています。RM は、CSLRMDRG 要求に対して応答を戻しません。

**SCITOKEN=***symbol*

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

関連概念:

208 ページの『グローバル・リソースを管理するための CSL RM 要求の発行』

---

## CSLRMPRI: プロセス開始

クライアントは CSLRMPRI 要求を使用して、IMSpIex でのプロセスを開始することができます。RM では、あるタイプの全 IMSpIex プロセスを一度に 1 つだけ進行させることができます。そのタイプの他の全 IMSpIex プロセスが進行中の場合、プロセス開始は失敗します。

この要求はアセンブラー言語でサポートされます。

### CSLRMPRI の構文

#### CSLRMPRI DSECT の構文

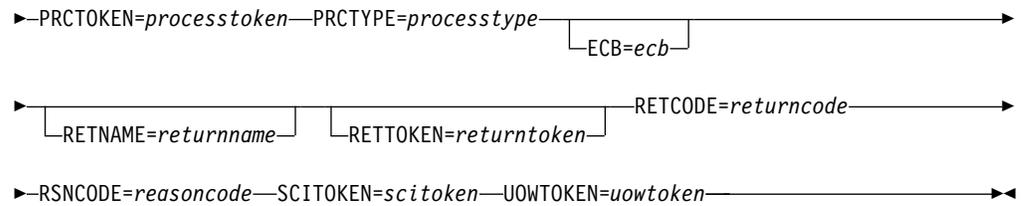
CSLRMPRI 要求の DSECT 機能は、CSLRMPRI パラメーター・リストの長さ用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶CSLRMPRI—FUNC=DSECT◀◀

#### CSLRMPRI INITIATE の構文

CSLRMPRI 要求の INITIATE 機能は、全 IMSpIex プロセスを開始します。

▶▶CSLRMPRI—FUNC=INITIATE—PARM=*parm*—PRCNAME=*processname*▶▶



## CSLRMPRI のパラメーター

### **ECB=***symbol*

#### **ECB=(r2-r12)**

(オプション) - 非同期要求に使用される z/OS ECB のアドレスを指定します。要求が完了したとき、指定された ECB は通知されます。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、要求の呼び出し側は、CSLRMPRI から制御を受け取った後で、WAIT (あるいはそれと同等のもの) を発行してから、この要求によって戻された何らかのデータ (RETCODE フィールドや RSNCODE フィールドを含む) を使用したり調べたりする必要があります。

### **PARM=***symbol*

#### **PARM=(r2-r12)**

(必須) -- CSLRMPRI パラメーター・リストを指定します。パラメーター・リストの長さは、RPRI\_PARMLN でパラメーター・リストの長さとして EQU 値と同じでなければなりません。

### **PRCNAME=***symbol*

#### **PRCNAME=(r2-r12)**

(必須) -- プロセス名が入った 8 バイト・フィールドを指定します。プロセス名はクライアントが定義するものであり、RM に対して意味はありません。RM は、一度に特定のプロセス・タイプのプロセスの 1 つのインスタンスだけを進行させるために、プロセス名とプロセス・タイプを使用します。

### **PRCTOKEN=***symbol*

#### **PRCTOKEN=(r2-r12)**

(必須) -- 呼び出し元に戻されるプロセス・トークンを受け取る 16 バイト・フィールドを指定します。プロセス・トークンは、プロセス・インスタンスを一意的に識別します。IMSpIex がリソース構造を定義されている場合、戻されるプロセス・トークンはゼロです。IMSpIex がリソース構造を定義されていない場合、プロセス・トークンは非ゼロです。後続の CSLRMPRS、CSLRMPRR、または CSLRMPRT 要求で、プロセス・トークンは入力として指定されなければなりません。

### **PRCTYPE=***symbol*

#### **PRCTYPE=(r2-r12)**

(必須) -- 1 バイトのクライアント定義プロセス・タイプを指定します。一度には、特定のタイプの 1 つのプロセスだけが進行可能です。プロセス・タイプは 1 から 255 が可能です。

### **RETCODE=***symbol*

#### **RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。

RM 戻りコードは、CSLRRR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。返される戻りコードについては、以下の表に説明があります。

**RETNAME=symbol**

**RETNAME=(r2-r12)**

(オプション) -- 呼び出し元に戻される RM 名を受け取る 8 バイト・フィールドを指定します。これは、SCI の要求送信先のターゲット RM アドレス・スペースの CSL メンバー名です。

**RETTOKEN=symbol**

**RETTOKEN=(r2-r12)**

(オプション) -- 呼び出し元に戻される RM の SCI トークンを受け取る 16 バイト・フィールドを指定します。これは、SCI の要求送信先のターゲット RM アドレス・スペースの SCI トークンです。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。RM 理由コードは、CSLRRR で定義されています。SCI 理由コードは、CSLSRR で定義されています。返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol**

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

**UOWTOKEN=symbol**

**UOWTOKEN=(r2-r12)**

(必須) -- 作業単位トークンが入っている 16 バイト・フィールドを指定します。UOW トークンは、このプロセスのインスタンスを一意的に識別し、プロセス・ステップのすべてを結び付けます。RM プロセス・ステップ要求 CSLRMPRS では、UOW トークンの指定が必要です。UOW トークンはクライアントが定義するものであり、RM に対して意味はありません。

## CSLRMPRI の戻りコードと理由コード

以下の表は、CSLRMPRI 要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 54. CSLRMPRI の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'03000008'	X'00002000'	クライアントは登録されません。
	X'00002208'	プロセス・タイプが無効。
	X'00002310'	UOW トークンが無効。
X'03000010'	X'00004000'	CQS アドレス・スペースが利用不能。要求を再試行して、使用可能な CQS のある、別の RM に要求を経路指定してみてください。

表 54. CSLRMPRI の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'00004100'	要求されたバージョンはサポートされていません。クライアントがコンパイルした CSLRMPRI のバージョンは、RM でサポートされていません。新しい CSLRMPRI 機能を使用する新しいリリースに IMS をマイグレーションする前に、すべての RM を新しいリリースにマイグレーションしなければなりません。
	X'00004120'	同じタイプのプロセスがすでに進行中です。このプロセス開始要求はリジェクトされます。後で要求を再試行してください。
	X'0000412C'	同じタイプの別のプロセスがすでに進行中です。このプロセス開始要求はリジェクトされます。後で要求を再試行してください。
X'03000014'	X'00005114'	処理ブロック割り振りは失敗しました。
	X'00005200'	CQS 要求が予期しないエラーになりました。
	X'00005204'	RM が作成した要求入力为正しくなかったため、CQS 要求は失敗しました。
	X'00005208'	リソース構造が利用不能。
	X'0000520C'	リソース構造が満杯です。
	X'00005210'	RM は、ハッシュ・テーブルに処理ブロックを追加できません。
	X'00005218'	RM は、ハッシュ・テーブル内で処理ブロックをスキャンできません。
	X'00005220'	RM は、プロセス・ラッチを取得できません。

関連概念:

- 208 ページの『グローバル・リソースを管理するための CSL RM 要求の発行』
- 209 ページの『全 IMSplex プロセスを調整するための CSL RM 要求の発行』

## CSLRMPRR: プロセス応答

クライアントは CSLRMPRR 要求を発行して、全 IMSplex プロセス内のステップに応答することができます。

この要求はアセンブラー言語でサポートされます。

### CSLRMPRR 構文

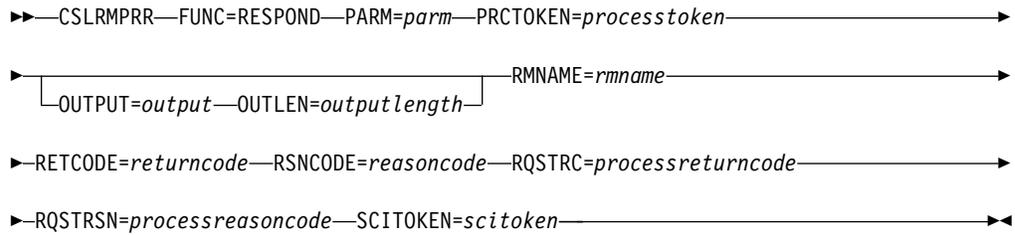
#### CSLRMPRR DSECT の構文

CSLRMPRR 要求の DSECT 機能は、CSLRMPRR パラメーター・リストの長さ用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶—CSLRMPRR—FUNC=DSECT—▶

#### CSLRMPRR RESPOND の構文

CSLRMPRR 要求の RESPOND 機能は、全 IMSplex プロセス内のステップに応答します。



## CSLRMPRR パラメーター

**OUTLEN=***symbol*

**OUTLEN=(r2-r12)**

(必須) -- プロセス・ステップ出力バッファの長さが入っている 4 バイト入力フィールドを指定します。OUTLEN= には、OUTPUT= パラメーターで示された出力の長さが入っています。

**OUTPUT=***output*

**OUTPUT=(r2-r12)**

(必須) -- 呼び出し元で作成される出力バッファのアドレスが入っている 4 バイト・フィールドを指定します。出力はクライアント定義されるものであって、このクライアントのステップ処理結果が入っています。出力長は、OUTLEN= フィールドに入れて戻されます。

**PARAM=***symbol*

**PARAM=(r2-r12)**

(必須) -- CSLRMPRR パラメーター・リストを指定します。パラメーター・リストの長さは、RPRR\_PARMLN でパラメーター・リストの長さとして EQU 値と同じでなければなりません。

**PRCTOKEN=***symbol*

**PRCTOKEN=(r2-r12)**

(必須) -- プロセスを一意的に識別するプロセス・トークンが入っている 16 バイト・フィールドを指定します。このトークンは、CSLRMPRI FUNC=INITIATE 要求が成功したので、戻されました。

IMSplex がリソース構造を定義されている場合、プロセス・トークンはゼロです。

**RETCODE=***symbol*

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。SCI 戻りコードは、CSLSRR で定義されています。RM は、CSLRMPRR に対して応答を戻しません。

**RMNAME=***symbol*

**RMNAME=(r2-r12)**

(必須) -- プロセス・ステップ応答の送信先である RM 名が入った 8 バイト・フィールドを指定します。これは、プロセス・ステップを開始した RM です。

**RQSTRC=***symbol*

**RQSTRC=(r2-r12)**

(必須) -- 出力でプロセス・ステップの開始元に渡す戻りコードを入れる 4 バイト・フィールドを指定します。戻りコードは、プロセス・ステップ開始元クライアントによって定義され、プロセス・ステップの結果を示します。

**RQSTRSN=symbol****RQSTRSN=(r2-r12)**

(必須) -- 出力でプロセス・ステップの開始元に渡す理由コードを入れる 4 バイト・フィールドを指定します。理由コードは、プロセス・ステップ開始元クライアントによって定義され、プロセス・ステップの結果を示します。

**RSNCODE=symbol****RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。RM 理由コードは、CSLRRRR で定義されています。RM は、CSLRMPRR に対して応答を戻しません。

**SCITOKEN=symbol****SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

**CSLRMPRR の戻りコードと理由コード**

CSLRMPRR は、SCI メッセージ・プロトコルを使って、ターゲット・クライアント・アドレス・スペースに送られます。RM は、コードを CSLRMPRR に戻しません。CSLSCMSG 要求に適用できる戻りコードおよび理由コードはすべて、CSLRMPRR 要求で戻すことができます。

関連概念:

208 ページの『グローバル・リソースを管理するための CSL RM 要求の発行』

209 ページの『全 IMSplex プロセスを調整するための CSL RM 要求の発行』

---

**CSLRMPRS: プロセス・ステップ**

クライアントは CSLRMPRS 要求を発行して、全 IMSplex プロセスのステップを実行できます。プロセスは、ゼロ個、1 個、または複数個のステップで構成されています。

この要求はアセンブラー言語でサポートされます。

**CSLRMPRS の構文****CSLRMPRS DSECT の構文**

CSLRMPRS 要求の DSECT 機能は、CSLRMPRS パラメーター・リストの長さおよびプロセス・ステップ要求オプション用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶—CSLRMPRS—FUNC=DSECT—▶▶

## CSLRMPRS PROCESS の構文

CSLRMPRS 要求の PROCESS 機能は、全 IMSplex プロセス内のステップを実行します。



## CSLRMPRS のパラメーター

**CDATA=***symbol*

**CDATA=(***r2-r12***)**

(オプション) -- 全 IMSplex プロセス・ステップに参加しているクライアントに送信するクライアント・データが入っている可変長域を指定します。クライアント・データはクライアントに対しては意味がありますが、RM に対してはありません。

**CDATALEN=***symbol*

**CDATALEN=(***r2-r12***)**

(オプション) -- クライアント・データ長が入っている 4 バイト入力フィールドを指定します。このパラメーターを指定する場合、CDATA= も指定しなければなりません。

**ECB=***symbol*

**ECB=(***r2-r12***)**

(オプション) - 非同期要求に使用される z/OS ECB のアドレスを指定します。要求が完了したとき、指定された ECB は通知されます。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、要求の起動側は、CSLRMPRS から制御を受け取った後、この要求から

戻されたデータ (RETCODE フィールドおよび RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を出す必要があります。

**LIST=symbol**

**LIST=(r2-r12)**

(必須) -- プロセス・ステップの送信先であるクライアントのリストが入った可変長入力リストを指定します。

そのプロセス・ステップ・リストには、リスト・ヘッダーと 1 つ以上のリスト項目が含まれます。リスト・ヘッダーには、リスト・ヘッダー長、パラメーター・リスト・バージョン、リスト項目長、リスト項目カウント、およびユーザー・データ域が入っています。リスト・ヘッダー・ユーザー・データ域は、プロセス・ステップ出力のリスト・ヘッダーに入れられて要求元に渡されます。各リスト項目には、クライアント名とオプションのユーザー・データ域が入っています。ユーザー・データ域は、プロセス・ステップ出力のリスト項目に入れられて要求元に戻されます。リスト項目は、連続するストレージに置かなければなりません。

CSLRPRSL マクロは、プロセス・ステップ・リストをマップします。

**LISTLEN=symbol**

**LISTLEN=(r2-r12)**

(必須) -- プロセス・ステップ・リスト長が入っている 4 バイト入力フィールドを指定します。

**OPTWORD1=symbol**

**OPTWORD1=(r2-r12)**

(オプション) - プロセス・ステップ・オプションが入っている 4 バイト・フィールドを指定します。CSLRMPRS FUNC=DSECT は、プロセス・ステップ・オプションをマップします。

**X'80000000'**

エラー後のプロセス・ステップを強制してください。アクティブでない IMSplex メンバーに対してプロセス・ステップがすでに進行中の場合、進行中のプロセス・ステップを引き継いでください。リソース構造の障害のようなエラーのために進行中のプロセスがないことが分かっている場合、プロセスを開始し、プロセス・ステップを実行してください。

**OUTLEN=symbol**

**OUTLEN=(r2-r12)**

(必須) -- CSLRMPRS 要求から戻される出力バッファの長さを受け取る 4 バイト・フィールドを指定します。要求によって戻された後、このワードには、OUTPUT= パラメーターで示されたバッファの長さが入っています。出力が作成されない場合、出力バッファ長はゼロです。出力が作成される前にエラーが検出されると、出力は作成されません。

ストレージの使用が終了したとき、呼び出し元は、CSLSCBFR FUNC=RELEASE 要求を出すことにより、このストレージを解放する必要があります。

**OUTPUT=outputaddress**

**OUTPUT=(r2-r12)**

(必須) -- CSLRMPRS 要求から戻される可変長出力バッファのアドレスを受け取る 4 バイト・フィールドを指定します。出力バッファには、参加してい

る各クライアントからのクライアント定義データが入っていて、プロセス・ステップの結果を示します。出力バッファ長は、OUTLEN= フィールドに入れて戻されます。

出力が作成されない場合、出力バッファ・アドレスはゼロです。出力が作成される前にエラーが検出されると、出力は作成されません。

CSLRPRSO マクロは、戻される出力バッファをマップします。出力バッファ・ヘッダーには、目印、出力バッファ長、CSLRPRSO バージョン、ヘッダー長 (プロセス・リスト項目の先頭までのオフセット)、リスト項目最小サイズ、プロセス・リスト項目カウント、ユーザー・データ域、および CSLRPRSO 作成タイム・スタンプが入っています。ユーザー・データ域には、入力プロセス・ステップ・リスト・ヘッダーに入れて渡されるユーザー・データが入っています。

各出力バッファ項目は、プロセス・ステップに参加したクライアントからの結果を表します。各項目の内容は、次のとおりです。

- 項目の長さ
- クライアント名
- ユーザー・データ -- 入力プロセス・ステップ・リストに入れて渡されるユーザー・データ
- プロセス・ステップ応答の長さ
- プロセス・ステップ応答
- 完了コード (CSLRPRR) -- 完了コードには、以下のものがあります。

**X'00000000'**

クライアントはステップを正常に処理します。

**X'00000044'**

クライアントは、プロセス・ステップがタイムアウトになる前に応答しませんでした。

**X'00000048'**

クライアントが RM に登録されていないために、クライアントにプロセス・ステップ要求が届きませんでした。

このバッファは、呼び出し元によって事前割り振りされません。要求がこのワードを戻した場合、そこには、プロセスに参加した IMSplex メンバーからの情報を含むバッファのアドレスが入っています。ストレージの使用が終了したとき、呼び出し元は、CSLSCBFR FUNC=RELEASE 要求を出すことにより、このストレージを解放する必要があります。

**PARM=***symbol*

**PARM=(r2-r12)**

(必須) -- CSLRMPRS パラメーター・リストを指定します。パラメーター・リストの長さは、RPRS\_PARMLN でパラメーター・リストの長さとして EQU 値と同じでなければなりません。

**PRCNAME=***symbol*

**PRCNAME=(r2-r12)**

(必須) -- プロセス名が入った 8 バイト・フィールドを指定します。プロセス名はクライアントが定義するものであり、RM に対して意味はありません。RM

は、一度に特定のプロセス・タイプのプロセスの 1 つのインスタンスだけを進行させるために、プロセス名とプロセス・タイプを使用します。

**PRCTOKEN=symbol**

**PRCTOKEN=(r2-r12)**

(必須) -- プロセスを一意的に識別するプロセス・トークンが入っている 16 バイト・フィールドを指定します。このトークンは、CSLRMPRI  
FUNC=INITIATE 要求が成功したので、戻されました。

IMSpIex がリソース構造を定義されている場合、プロセス・トークンはゼロです。

**PRCTYPE=symbol**

**PRCTYPE=(r2-r12)**

(必須) -- 1 バイトのクライアント定義プロセス・タイプを指定します。一度には、特定のタイプの 1 つのプロセスだけが進行可能です。プロセス・タイプは 1 から 255 が可能です。

**RETCODE=symbol**

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。SCI 戻りコードは、CSLSRR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。返される戻りコードについては、以下の表に説明があります。

**RETNAME=symbol**

**RETNAME=(r2-r12)**

(オプション) -- 呼び出し元に戻される RM 名を受け取る 8 バイト・フィールドを指定します。これは、SCI の要求送信先のターゲット RM アドレス・スペースの CSL メンバー名です。

**RETNAME=symbol**

**RETNAME=(r2-r12)**

(オプション) -- 呼び出し元に戻される RM 名を受け取る 8 バイト・フィールドを指定します。これは、SCI の要求送信先のターゲット RM アドレス・スペースの CSL メンバー名です。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。RM 理由コードは、CSLRRR で定義されています。SCI 理由コードは、CSLSRR で定義されています。返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol**

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

**STEPNAME=symbol**

**STEPNAME=(r2-r12)**

(必須) -- プロセス・ステップ名が入った 4 バイト・フィールドを指定します。

プロセス・ステップ名はクライアントが定義するものであり、RM に対して意味はありません。各プロセス・ステップは、異なる名前であればなりません。

**TIMEOUT=timeoutvalue**

**TIMEOUT=symbol**

**TIMEOUT=(r2-r12)**

(オプション) -- プロセス・ステップ・タイムアウト値 (秒) が入っている 4 バイト・フィールドを指定します。すべての参加者がプロセス・ステップに応答し終わる前に、ステップの処理の途中でタイムアウト値に達すると、RM はプロセス・ステップを終了し、用意できている応答を戻します。指定されたタイムアウト値が小さすぎると、不完全な応答が戻されます。TIMEOUT 値は、ステップを処理しているクライアントが応答不能の場合でも、応答が必ず戻されるようにします。

デフォルトのタイムアウト値は 5 分 (300 秒) です。その要求にタイムアウトが必要ない場合、負の 1 (-1) を指定してください。

TIMEOUT 値は、プロセス・ステップをタイムアウトにする、可能な最小のタイム値です。RM は、内部的に、タイマーが 5 秒ごとにポップするように設定します。RM タイマーがポップすると、RM は、いずれかのプロセス・ステップ・タイムアウト値が満了していないかを検査します。プロセス・ステップ・タイムアウト値が RM タイマー値より小さい場合、ステップ処理の実際の長さは、ユーザー指定の TIMEOUT 値より長くすることができます。

**UOWTOKEN=symbol**

**UOWTOKEN=(r2-r12)**

(必須) -- 作業単位トークンが入っている 16 バイト・フィールドを指定します。UOW トークンは、このプロセスのインスタンスを一意的に識別し、プロセス・ステップのすべてを結び付けます。この UOW トークンは、CSLRMPRI FUNC=INITIATE 要求で指定された UOW トークンと一致しなければなりません。UOW トークンはクライアントが定義するものであり、RM に対して意味はありません。

## CSLRMPRS の戻りコードと理由コード

以下の表は、CSLRMPRS 要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 55. CSLRMPRS の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'03000008'	X'00002000'	クライアントは登録されません。
	X'00002110'	リスト・ヘッダー (PRSL_PVER) 内のリスト・バージョンはゼロであり、これは無効です。リスト・ヘッダー内のリスト・バージョンは、最大リスト・バージョン (PRSL_PVERMAX) に設定されていなければなりません。
	X'00002114'	リスト・ヘッダー長は、ゼロ、または渡されたリスト長より大きくてはなりません。リスト・ヘッダー内のリスト・ヘッダー長 (PRSL_HDRLEN) は、リスト・ヘッダー長に設定されていなければなりません。

表 55. CSLRMPRS の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'00002140'	クライアント・データ長はゼロ、または 256 より大きくすることはできません。
	X'00002208'	プロセス・タイプが無効。
	X'0000220C'	プロセス・ステップ・オプションが無効。
	X'00002300'	プロセス・トークンが無効。
	X'00002310'	UOW トークンが無効。
X'0300000C'	X'00003000'	プロセス・ステップは少なくとも 1 つのクライアントについては成功しましたが、すべてについて成功したわけではありません。OUTPUT バッファ内の各リスト項目の完了コードを検査して、個々のエラーを調べてください。
	X'00003004'	要求はすべてのクライアントについて失敗しました。OUTPUT バッファ内の各リスト項目の完了コードを検査して、個々のエラーを調べてください。
X'03000010'	X'00004000'	CQS アドレス・スペースが利用不能。要求を再試行して、使用可能な CQS のある、別の RM に要求を経路指定してみてください。
	X'00004100'	要求されたバージョンはサポートされていません。クライアントがコンパイルした CSLRMPRS のバージョンは、RM でサポートされていません。新しい CSLRMPRS 機能を使用する新しいリリースに IMS をマイグレーションする前に、すべての RM を新しいリリースにマイグレーションしなければなりません。
	X'00004104'	リストのバージョンはサポートされていません。クライアントが作成したプロセス・ステップ・リストのバージョンは、RM でサポートされていません。新しい CSLRMPRS 機能を使用する新しいリリースにクライアントをマイグレーションする前に、すべての RM を新しいリリースにマイグレーションしなければなりません。
	X'00004108'	SCI アドレス・スペースが利用不能。SCI は使用可能であって、CSLRMPRS 要求を RM に送信しました。RM は、SCI メッセージをアクティブ・クライアントに送信することにより、プロセス・ステップを調整しようとしていました。メッセージを SCI に送る SCI 要求は、システム上で SCI がアクティブになっていないアクティブ・クライアントが 1 つ以上あるために、失敗しました。一部のクライアントはステップを正常に処理した可能性もあります。
	X'00004124'	プロセスは進行中ではありません。プロセス・ステップはリジェクトされます。

表 55. CSLRMPRS の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'00004128'	プロセス・ステップがすでに進行中です。プロセス・ステップはリジェクトされます。前のプロセス・ステップが進行中のときにエラーが発生し、そのプロセス・ステップの所有者がアクティブのままであるためにプロセス・ステップがすでに進行中の場合、次のプロセス・ステップは、そのプロセス・ステップの所有者が FORCE オプションを使って指定しなければなりません。
X'03000014'	X'00005000'	出力バッファのストレージ割り振りが失敗しました。プロセス・ステップが成功したかはわかりません。
	X'00005114'	処理ブロック割り振りは失敗しました。
	X'00005118'	プロセス・ステップ応答ブロック割り振りは失敗しました。
	X'00005200'	CQS 要求が予期しないエラーになりました。
	X'00005204'	RM が作成した要求入力为正しくなかったため、CQS 要求は失敗しました。
	X'00005208'	リソース構造が利用不能。
	X'00005210'	RM は、ハッシュ・テーブルに処理ブロックを追加できません。
	X'00005214'	RM は、ハッシュ・テーブル内で処理ブロックを見つけられません。
	X'00005218'	RM は、ハッシュ・テーブル内で処理ブロックをスキャンできません。
	X'00005300'	SCI エラーが検出されました。SCI は使用可能であっても、CSLRMPRS 要求を RM に送信しました。RM は、SCI メッセージをアクティブ・クライアントに送信することにより、プロセス・ステップを調整しようとしました。メッセージを SCI に送る SCI 要求は、アクティブ・クライアントが 1 つ以上あるためのエラーで失敗しました。一部のクライアントはステップを正常に処理した可能性もあります。

関連概念:

208 ページの『グローバル・リソースを管理するための CSL RM 要求の発行』

209 ページの『全 IMSplex プロセスを調整するための CSL RM 要求の発行』

## CSLRMPRT: プロセス終了

CSLRMPRT 要求を発行することで、全 IMSplex プロセスを終了できます。プロセスに参加しているクライアントはどれでも、CSLRMPRT FUNC= TERMINATE 要求を出してプロセスを終了させることができます。

この要求はアセンブラー言語でサポートされます。

## CSLRMPRT の構文

CSLRMPRT 要求の構文は次のとおりです。

### DSECT 構文

CSLRMPRT 要求の DSECT 機能は、CSLRMPRT パラメーター・リストの長さ用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

```
▶▶—CSLRMPRT—FUNC=DSECT—▶▶
```

### TERMINATE の構文

CSLRMPRT 要求の TERMINATE 機能は、全 IMSplex プロセスを終了させます。

```
▶▶—CSLRMPRT—FUNC=TERMINATE—  
└─OPTWORD1=processteroptions─┐ PARM=parm ─▶▶  
▶—PRCNAME=processname—PRCTOKEN=processtoken—PRCTYPE=processtype—▶▶  
▶—UOWTOKEN=uowtoken—  
└─RETNAME=returnname─┐ └─RETTOKEN=returntoken─┐  
▶—RETCODE=returncode—RSNCODE=reasoncode—SCITOKEN=scitoken—▶▶
```

## CSLRMPRT のパラメーター

**OPTWORD1=symbol**

**OPTWORD1=(r2-r12)**

(オプション) -- プロセス終了オプションが入っている 4 バイト・フィールドを指定します。CSLRMPRT FUNC=DSECT は、プロセス終了オプションをマップします。OPTWORD1 には、値 X'80000000' を指定できます。

**PARM=symbol**

**PARM=(r2-r12)**

(必須) -- CSLRMPRT パラメーター・リストを指定します。パラメーター・リストの長さは、RPRT\_PARMLN でパラメーター・リストの長さとして EQU 値と同じでなければなりません。

**PRCNAME=symbol**

**PRCNAME=(r2-r12)**

(必須) -- プロセス名が入った 8 バイト・フィールドを指定します。プロセス名はクライアントが定義するものであり、RM に対して意味はありません。RM は、一度に特定のプロセス・タイプのプロセスの 1 つのインスタンスだけを進行させるために、プロセス名とプロセス・タイプを使用します。

**PRCTOKEN=symbol**

**PRCTOKEN=(r2-r12)**

(必須) -- プロセスを一意的に識別するプロセス・トークンが入っている 16 バイト・フィールドを指定します。このトークンは、CSLRMPRI FUNC=INITIATE 要求が成功したので、戻されました。

IMSplex がリソース構造を定義されている場合、プロセス・トークンはゼロです。

**PRCTYPE=symbol**

**PRCTYPE=(r2-r12)**

(必須) -- 1 バイトのクライアント定義プロセス・タイプを指定します。一度には、特定のタイプの 1 つのプロセスだけが進行可能です。プロセス・タイプは 1 から 255 が可能です。

**RETCODE=symbol**

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。SCI 戻りコードは、CSLSRR で定義されています。RM は、CSLRMPRT に対して応答を戻しません。

**RETNAME=symbol**

**RETNAME=(r2-r12)**

(オプション) -- SCI がプロセス終了要求を送った先の RM アドレス・スペースの名前を受け取る 8 バイト・フィールドを指定します。

**RETTOKEN=symbol**

**RETTOKEN=(r2-r12)**

(オプション) -- SCI がプロセス終了要求を送った先の RM アドレス・スペースの SCI トークンを受け取る 16 バイト・フィールドを指定します。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。SCI 理由コードは、CSLSRR で定義されています。RM は、CSLRMPRT 要求に対して応答を戻しません。

**SCITOKEN=symbol**

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

**UOWTOKEN=symbol**

**UOWTOKEN=(r2-r12)**

(必須) -- 作業単位トークンが入っている 16 バイト・フィールドを指定します。UOW トークンは、このプロセスのインスタンスを一意的に識別し、プロセス・ステップのすべてを結び付けます。この UOW トークンは、CSLRMPRI FUNC=INITIATE 要求で指定された UOW トークンと一致しなければなりません。UOW トークンはクライアントが定義するものであり、RM に対して意味はありません。

## CSLRMPRT の戻りコードと理由コード

CSLRMPRT は、SCI メッセージ・プロトコルを使って、ターゲット・クライアント・アドレス・スペースに送られます。CSLSCMSG 要求に適用できる戻りコードおよび理由コードはすべて、CSLRMPRT 要求で戻すことができます。CSLRMPRT は、それ以上の戻りコードおよび理由コードを出しません。

関連概念:

## CSLRMQRY: リソース照会

CSLRMQRY 要求を発行することで、リソース構造上の 1 つ以上の固有な名前付きリソースを照会できます。

この要求はアセンブラー言語でサポートされます。

### CSLRMQRY の構文

#### CSLRMQRY DSECT の構文

CSLRMQRY 要求の DSECT 機能は、プログラムに次の入力および出力を組み込みます。

- CSLRMQRY パラメーター・リストの長さ用の等価 (EQU) ステートメント
- CSLRMQRY の戻りコード、理由コード、および完了コード
- 入力照会リストをマップするための CSLRQRYL DSECT
- 照会出力をマップするための CSLRQRYO DSECT

▶▶CSLRMQRY—FUNC=DSECT—————▶▶

#### CSLRMQRY QUERY の構文

CSLRMQRY 要求の QUERY 機能は、リソース構造上の 1 つ以上の固有な名前付きリソースを照会します。

▶▶CSLRMQRY—FUNC=QUERY—PARAM=parm—LIST=querylist—————▶▶

▶▶LISTLEN=querylistlength—————▶▶  
└─RETNAME=returnname─┘

▶▶└─RETTOKEN=returntoken─┘—————▶▶  
OUTPUT=output—OUTLEN=outputlength—————▶▶

▶▶└─ECB=ecb─┘—————▶▶  
└─PROTOCOL=RQST─┘ RETCODE=returncode—————▶▶

▶▶RSNCODE=reasoncode—SCITOKEN=scitoken—————▶▶

### CSLRMQRY のパラメーター

**ECB=symbol**

**ECB=(r2-r12)**

(オプション) - 非同期要求に使用される z/OS ECB のアドレスを指定します。要求が完了したとき、指定された ECB は通知されます。ECB が指定されてい

ないと、そのタスクは、要求が完了するまで中断されます。 ECB が指定された場合、要求の起動側は、CSLRM QRY から制御を受け取った後、この要求から戻されたデータ (RETCODE フィールドおよび RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を出す必要があります。

#### **LIST=symbol**

##### **LIST=(r2-r12)**

(必須) -- 呼び出し元が作成した照会リソース・リストを指定します。各リスト項目が、それぞれ別々の照会要求です。リスト長は、リスト項目の数に応じて異なります。

そのリストには、ヘッダーと 1 つ以上のリスト項目が含まれます。リスト項目は、連続するストレージに置かなければなりません。各照会リスト項目の内容は、次のとおりです。

- リソース名 -- リソースのクライアント定義の名前。リソース名は、ワイルドカード名にすることができます。ワイルドカード名の場合は、そのワイルドカード名に合致するあらゆるリソースが戻されます。
- リソース・タイプ -- リソース・タイプは、リソース構造上のリソースのクライアント定義の物理的グループ化です。有効な値は 1 から 255 です。
- 照会オプション (オプション) - 照会について実行する特殊な処理を表すオプション。
- 所有者 (オプション) - リソースの所有者。所有者を指定する場合、そのリソースが戻されるのは、リソース名および所有者がリソース構造体にあるリソースに合致する場合だけです。所有者を省略するには、2 進ゼロを指定します。こうすると、照会は、出力リスト・エントリー内の RQYO\_OWNER フィールドに所有者名を戻します。
- ユーザー (オプション) - 呼び出し元によって設定されたユーザー・フィールド。これは、入力リスト・エントリーに関連付けられた出力リスト・エントリー内に渡し戻されます。

#### **LISTLEN=symbol**

##### **LISTLEN=(r2-r12)**

(必須) -- 4 バイトの照会リソース・リスト長を指定します。

#### **OUTLEN=symbol**

##### **OUTLEN=(r2-r12)**

(必須) -- CSLRMQRY 要求から戻される出力バッファの長さを受け取る 4 バイト・フィールドを指定します。 OUTLEN には、OUTPUT パラメーターで示された出力バッファの長さが入っています。出力データ (ヘッダーと項目) の長さは、出力ヘッダー・データに入れて渡され、 CSLRQRYO によってマップされています。

#### **OUTPUT=output**

##### **OUTPUT=(r2-r12)**

(必須) -- CSLRMQRY 要求から戻される可変長出力のアドレスを受け取る 4 バイト・フィールドを指定します。その出力には、試みられたリソース照会に対するヘッダーと 1 つ以上の照会項目が含まれます。出力長は、OUTLEN フィールドに入れて戻されます。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力アドレスはゼロです。

CSLRQRYO マクロは、戻される出力をマップします。その出力には、ヘッダーと 1 つ以上のリスト項目が含まれます。ヘッダーの内容は次のとおりです。

- 目印
- 出力の長さ
- CSLRQRYO のバージョン
- CSLRQRYO のヘッダーの長さ (項目の先頭までのオフセット)
- 最小項目長 (DATA2 までのオフセット)
- リソース項目カウント
- タイム・スタンプ

各出力項目は、試みられたリソース照会を表します。各項目には以下のパラメーターが含まれています。

- 出力項目長 -- リスト項目長は、DATA2 が戻されるかどうかによって異なります。
- 名前タイプ -- 名前タイプは、名前タイプ内のクライアント定義リソース名の一意性を保証する、リソース・タイプに関連付けられた、クライアント定義の値。有効な値は 1 から 255 です。
- リソース名 -- リソースのクライアント定義の名前。
- リソース・タイプ -- リソース・タイプは、リソース構造上のリソースのクライアント定義の物理的グループ化です。有効な値は 1 から 255 です。
- バージョン -- リソース・バージョンであって、リソースが更新された回数。
- DATA2 フラグ・バイト -- DATA2 が読み取られたかどうかを示すフラグ・バイト。
- リソース名状況フラグ -- リソース名状況は、照会出力リスト項目内のリソース名がどのように入力リソース・パラメーターと関連付けられているかを示します。これにより、入力リソース・パラメーターを、生成される出力照会リスト項目に結び付けることができます。可能なリソース名状況を以下に示します。

#### 特定パラメーター

特定のリソース名が指定されました。この照会リスト項目には、入力パラメーターと一致するリソース名が入れられます。

#### ワイルドカード・パラメーター

ワイルドカード・パラメーターが指定されました。この照会リスト項目には、ワイルドカード・パラメーターと完了コードが入れられます。この照会リスト項目には、特定のリソースに関する情報は入れられません。完了コードがゼロの場合、1 つ以上のワイルドカード・マッチ・リスト項目が続きます。

#### ワイルドカード・マッチ

ワイルドカード・パラメーターが指定されました。この項目には、入力ワイルドカード・パラメーターと一致する 1 つのリソースに関する情報が入れられます。ワイルドカード・パラメーター・リスト項目の後に、全ワイルドカード・マッチ・リスト項目が連続して続きます。

- 所有者 -- リソースの所有者。
- DATA1 -- 既存のリソースに関連付けられている、少量のクライアント・データ (固定長で、データ項目の付属域に含まれる)。

- DATA2 の長さ -- DATA2 が存在していて、DATA2 を読み取るオプションが設定されていた場合に、既存のリソースに関連付けられた多量のクライアント・データの長さ。
- オプション・ユーザー・フィールド -- 入力リスト項に関連付けられた出力リスト項目に入れて呼び出し元に渡される、オプションの 4 バイト・ユーザー・フィールド。
- DATA2 -- DATA2 が存在していて、DATA2 を読み取るオプションが設定されていた場合、既存のリソースに関連付けられた多量のクライアント・データ (可変長で、データ項目の 1 つ以上のデータ・エレメントに含まれる)。DATA2 の最大サイズは 61312 バイト (X'EF80') です。
- 照会要求の完了コード -- 完了コードは、CSLRRR によってマップされます。完了コードには、以下のものがあります。

**X'00000000'**

照会要求は成功しました。照会パラメーターと一致する少なくとも 1 つのリソースが、OUTPUT= で指定された出力バッファーに入れて戻されます。

**X'00000004'**

リソースが見つからなかった。

**X'00000008'**

リソース・タイプが無効。

**X'0000000C'**

名前タイプが無効。

**X'00000024'**

リソース構造が利用不能。

**X'00000034'**

指定されたオプションが無効。

**X'00000038'**

CQS 内部エラーのために、照会が失敗しました。

**X'0000003C'**

RM が作成した CQSBWSE リスト項目が正しくなかったため、照会は失敗しました。

出力バッファーは、呼び出し元によって事前割り振りされることはありません。要求がこのワードを戻すと、そこに照会出力を含むバッファーのアドレスがあります。ストレージの使用が終了したとき、呼び出し元は、CSLSCBFR FUNC=RELEASE 要求を出すことにより、このストレージを解放する必要があります。バッファーの長さは、OUTLEN= フィールドに入れて戻されます。

**PARM=***symbol*

**PARM=(***r2-r12***)**

(必須) -- CSLRMQRY パラメーター・リストを指定します。パラメーター・リストの長さは、RQRY\_PARMLN でパラメーター・リストの長さとして EQU 値と同じでなければなりません。

**PROTOCOL=RQST**

(オプション) -- 要求を RM に送信するための SCI プロトコル。RQST は、SCI 要求インターフェースを使って照会要求を送信します。

**RETCODE=symbol****RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。RM 戻りコードは、CSLRRR で定義されています。SCI 戻りコードは、CSLSRR で定義されています。返される戻りコードについては、以下の表に説明があります。

**RETNAME=symbol****RETNAME=(r2-r12)**

(オプション) -- 呼び出し元に戻される RM 名を受け取る 8 バイト・フィールドを指定します。これは、SCI の要求送信先のターゲット RM アドレス・スペースの CSL メンバー名です。

**RETTOKEN=symbol****RETTOKEN=(r2-r12)**

(オプション) -- 呼び出し元に戻される RM の SCI トークンを受け取る 16 バイト・フィールドを指定します。これは、SCI の要求送信先のターゲット RM アドレス・スペースの SCI トークンです。

**RSNCODE=symbol****RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。RM 理由コードは、CSLRRR で定義されています。SCI 理由コードは、CSLSRR で定義されています。返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol****SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

**CSLRMQRY の戻りコードと理由コード**

以下の表は、CSLRMQRY 要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 56. CSLRMQRY の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'03000004'	X'00001000'	リソースが見つからなかった。
X'03000008'	X'00002000'	クライアントは登録されません。
	X'00002100'	照会リストの長さが無効。
	X'00002108'	照会リストのアドレスが無効。

表 56. CSLRMQRY の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'00002110'	リスト・ヘッダー (QRYL_PVER) 内のリスト・バージョンはゼロであり、これは無効です。リスト・ヘッダー内のリスト・バージョンは、最大リスト・バージョン (QRYL_PVERMAX) に設定されていなければなりません。
	X'00002114'	リスト・ヘッダー長は、ゼロ、または渡されたリスト長より大きくてはなりません。リスト・ヘッダー内のリスト・ヘッダー長 (QRYL_HDRLLEN) は、リスト・ヘッダー長に設定されていなければなりません。
	X'00002404'	リソース構造が定義されていない。
X'0300000C'	X'00003000'	要求は少なくとも 1 つのリスト項目については成功しましたが、すべてについて成功したわけではありません。OUTPUT バッファ内の各照会リスト項目の完了コードを検査して、個々のエラーを調べてください。
	X'00003004'	要求はすべてのリスト項目について失敗しました。OUTPUT バッファ内の各照会リスト項目の完了コードを検査して、個々のエラーを調べてください。
X'03000010'	X'00004000'	CQS アドレス・スペースが利用不能。要求を再試行して、使用可能な CQS のある、別の RM に要求を経路指定してみてください。
	X'00004100'	要求されたバージョンはサポートされていません。クライアントがコンパイルした CSLRMQRY のバージョンは、RM でサポートされていません。新しい CSLRMQRY 機能を使用する新しいリリースに IMS をマイグレーションする前に、すべての RM を新しいリリースにマイグレーションしなければなりません。
	X'00004104'	リストのバージョンはサポートされていません。クライアントが照会リストを作成したバージョンは、RM でサポートされていません。新しい CSLRMQRY 機能を使用する新しいリリースにクライアントをマイグレーションする前に、すべての RM を新しいリリースにマイグレーションしなければなりません。
X'03000014'	X'00005000'	照会出力バッファのストレージ割り振りが失敗しました。
	X'00005108'	CQSBRWSE バッファのストレージ割り振りが失敗しました。
	X'00005200'	CQS 要求が予期しないエラーになりました。
	X'00005204'	RM が作成した要求入力为正しくなかったため、CQS 要求は失敗しました。

関連概念:

208 ページの『グローバル・リソースを管理するための CSL RM 要求の発行』

## CSLRMREG: クライアントを登録する

CSLRMREG 要求を使用して、クライアントを RM に登録できます。オプションで、クライアントのリソース・タイプおよび関連名前タイプを登録することもできます。RM に登録するクライアントは、CSLRMREG 要求を発行するための許可が必要です。ただし、全 IMSplex プロセスが進行中の場合には、クライアントを登録できません。

クライアントが RM 要求を発行するには、その前にクライアントを RM に登録する必要があります。クライアントは、登録されてから、実行される全 IMSplex プロセスに参加しなければなりません。IMSplex 内でアクティブな全 RM にクライアントを登録する必要があります。RM への登録が失敗した場合、正常に登録された RM からクライアントを登録解除する必要があります。RM に障害が起こった場合、RM が再始動して登録してください。

同じクライアントを複数回登録することができます。例えば、クライアントがすでに登録された後で、そのクライアント用のリソース・リストを指定する必要があるかもしれません。オプションですが、リソース・タイプを RM に定義し、名前タイプを各リソース・タイプに関連付けるには、クライアントと一緒にリソース・タイプを RM に登録してください。他の要求でリソース・タイプを指定するには、その前にリソース・タイプを登録する必要があります。リソース・タイプと名前タイプの関連付けが、前にすでに登録されているものと一致しない場合、クライアントを登録することはできません。

IMS 制御領域などのリソース処理クライアントがこの要求を出します。

この要求はアセンブラ言語でサポートされます。

### CSLRMREG 構文

#### CSLRMREG DSECT の構文

CSLRMREG 要求の DSECT 機能は、プログラムに次の入力および出力を組み込みます。

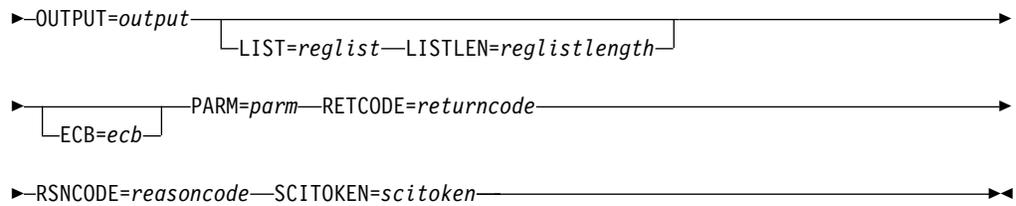
- CSLRMREG パラメーター・リストの長さ用の等価 (EQU) ステートメント
- CSLRMREG の戻りコード、理由コード、および完了コード
- 入力登録リストをマップするための CSLRREGL DSECT
- 登録出力をマップするための CSLRREGO DSECT

▶▶—CSLRMREG—FUNC=DSECT—▶▶

#### CSLRMREG REGISTER の構文

CSLRMREG 要求は、クライアントを RM に登録し、オプションでクライアントのリソース・タイプおよび 関連名前タイプを RM に登録するためのものです。

▶▶—CSLRMREG—FUNC=REGISTER—RMNAME=*rmname*—OUTLEN=*output length*—▶▶



## CSLRMREG のパラメーター

### ECB=*symbol*

#### ECB=(*r2-r12*)

(オプション) - 非同期要求に使用される z/OS ECB のアドレスを指定します。要求が完了したとき、指定された ECB は通知されます。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、要求の呼び出し側は、CSLRMREG から制御を受け取った後で、WAIT (あるいはそれと同等のもの) を発行してから、この要求によって戻された何らかのデータ (RETCODE フィールドや RSNCODE フィールドを含む) を使用したり調べたりする必要があります。

### LIST=*symbol*

#### LIST=(*r2-r12*)

(オプション) -- 呼び出し元が作成した登録リストを指定します。各リスト項目が、それぞれ別々のリソース・タイプ登録です。リソース構造が定義されていないときに登録リストが指定されると、それは無視されます。

CSLRREGL マクロは、登録リスト項目をマップします。そのリストには、ヘッダーと 1 つ以上のリスト項目が含まれます。リスト項目は、連続するストレージに置かなければなりません。各登録リスト項目の内容は、次のとおりです。

- リソース・タイプ
- 名前タイプ

### LISTLEN=*symbol*

#### LISTLEN=(*r2-r12*)

(オプション) -- 4 バイトの登録リスト長を指定します。LIST が指定された場合、LISTLEN は必須です。

### OUTLEN=*symbol*

#### OUTLEN=(*r2-r12*)

(必須) -- CSLRMREG 要求から戻される出力の長さを受け取る 4 バイト・フィールドを指定します。OUTLEN には、OUTPUT= パラメーターで示された出力の長さが入っています。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力長はゼロです。

### OUTPUT=*output*

#### OUTPUT=(*r2-r12*)

(必須) -- CSLRMREG 要求から戻される可変長出力のアドレスを受け取る 4 バイト・フィールドを指定します。その出力には、試みられた登録に対するヘッダーのみ、またはヘッダーの他、1 つ以上の登録項目が含まれます。出力長は、OUTLEN= フィールドに入れて戻されます。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力アドレスはゼロです。

CSLRREGO マクロは、戻される出力をマップします。その出力には、ヘッダーのみ、またはヘッダーの他、1 つ以上のリスト項目が含まれます。出力ヘッダーの内容は次のとおりです。

- 目印
- 出力長
- CSLRREGO のバージョン
- CSLRREGO のヘッダーの長さ (項目の先頭までのオフセット)
- CSLRREGO 項目の長さ
- 登録リスト・カウント
- タイム・スタンプ
- 登録の状況
- 構造のバージョン

各出力項目は、試みられた登録要求を表します。各項目の内容は、次のとおりです。

- リソース・タイプ
- 名前タイプ
- 登録要求の完了コード。完了コードは、CSLRRRR によってマップされます。完了コードには、以下のものがあります。

**X'00000000'**

登録は成功しました。

**X'00000008'**

リソース・タイプが無効。リソース・タイプは、ゼロにすることはできません。

**X'0000000C'**

名前タイプが無効。名前タイプは、ゼロにすることはできません。またはリソース・タイプが、すでに別の名前タイプで定義されています。

**PARM=symbol**

**PARM=(r2-r12)**

(必須) -- CSLRMREG パラメーター・リストを指定します。パラメーター・リストの長さは、RREG\_PARMLN でパラメーター・リストの長さとして EQU 値と同じでなければなりません。

**RETCODE=symbol**

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。RM 戻りコードは、CSLRRRR で定義されています。SCI 戻りコードは、CSLSRRR で定義されています。返される戻りコードについては、以下の表に説明があります。

**RMNAME=symbol**

**RMNAME=(r2-r12)**

(必須) -- 登録要求の送信先である 8 バイトの RM 名を指定します。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。RM 理由コードは、CSLRRR で定義されています。SCI 理由コードは、CSLSRR で定義されています。返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol**

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

## CSLRMREG の戻りコードと理由コード

以下の表は、CSLRMREG 要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 57. CSLRMREG の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'03000004'	X'00001100'	要求は正常に完了しましたが、LIST は無視されます。リソース構造が定義されていない。
X'03000008'	X'00002100'	登録リストの長さが無効。
	X'00002108'	登録リストのアドレスが無効。
	X'00002110'	リスト・ヘッダー (REGL_PVER) 内のリスト・バージョンはゼロであり、これは無効です。リスト・ヘッダー内のリスト・バージョンは、最大リスト・バージョン (REGL_PVERMAX) に設定されていなければなりません。
X'0300000C'	X'00002114'	リスト・ヘッダー長は、ゼロ、または渡されたリスト長より大きくてはなりません。リスト・ヘッダー内のリスト・ヘッダー長 (REGL_HDRLEN) は、リスト・ヘッダー長に設定されていなければなりません。
	X'00003000'	要求は少なくとも 1 つのリスト項目については有効ですが、すべてについて有効なわけではありません。有効なリスト項目の登録が実行されず、クライアントの登録はリジェクトされます。OUTPUT バッファ内の各リスト項目の完了コードを検査して、個々のエラーを調べてください。
X'03000010'	X'00003004'	要求はすべてのリスト項目について失敗しました。OUTPUT バッファ内の各リスト項目の完了コードを検査して、個々のエラーを調べてください。
	X'00004010'	このクライアントは許可されていません。

表 57. CSLRMREG の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'00004100'	要求されたバージョンはサポートされていません。クライアントがコンパイルした CSLRMREG のバージョンは、RM でサポートされていません。新しい CSLRMREG 機能を使用する新しいリリースに IMS をマイグレーションする前に、すべての RM を新しいリリースにマイグレーションしなければなりません。
	X'00004104'	リストのバージョンはサポートされていません。クライアントが登録リストを作成したバージョンは、RM でサポートされていません。新しい CSLRMREG 機能を使用する新しいリリースにクライアントをマイグレーションする前に、すべての RM を新しいリリースにマイグレーションしなければなりません。
X'03000014'	X'00005000'	登録出力バッファのストレージ割り振りが失敗しました。
	X'00005100'	CQSUPD バッファのストレージ割り振りが失敗しました。
	X'00005200'	CQS 要求が予期しないエラーになりました。
	X'00005204'	RM が作成した要求入力为正しくなかったため、CQS 要求は失敗しました。
	X'00005110'	クライアント・ブロック割り振りは失敗しました。

関連概念:

207 ページの『RM クライアント要求の順序』

208 ページの『グローバル・リソースを管理するための CSL RM 要求の発行』

## CSLRMUPD: リソースを更新する

CSLRMUPD 要求を発行することで、リソースが存在していなければ、リソースを作成し、存在していれば、それを更新することができます (指定されたバージョンがリソースのバージョンと一致していなければなりません)。クライアント・データがあってもなくても、リソースは作成または更新できます。

この要求はアセンブラ言語でサポートされます。

### CSLRMUPD の構文

#### CSLRMUPD DSECT の構文

CSLRMUPD 要求の DSECT 機能は、プログラムに次の入力および出力を組み込みます。

- CSLRMUPD パラメーター・リストの長さ用の等価 (EQU) ステートメント
- CSLRMUPD の戻りコード、理由コード、および完了コード
- 入力更新リストをマップするための CSLRUPDL DSECT
- 更新出力をマップするための CSLRUPDO DSECT

▶—CSLRMUPD—FUNC=DSECT—▶

## CSLRMUPD UPDATE の構文

CSLRMUPD 要求は、リソース構造上の固有な名前付きリソースを作成および更新するためのものです。

▶—CSLRMUPD—FUNC=UPDATE—PARAM=parm—LIST=updlist—LISTLEN=updlistlength—▶

▶—OUTPUT=output—OUTLEN=outputlength—  
└—ECB=ecb—┘ └—RETNAME=returnname—┘

▶—  
└—RETTOKEN=returntoken—┘ RETCODE=returncode—RSNCODE=reasoncode—▶

▶—SCITOKEN=scitoken—▶

## CSLRMUPD のパラメーター

**ECB=symbol**

**ECB=(r2-r12)**

(オプション) - 非同期要求に使用される z/OS ECB のアドレスを指定します。要求が完了したとき、指定された ECB は通知されます。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、要求の起動側は、CSLRM UPD から制御を受け取った後、この要求から戻されたデータ (RETCODE フィールドおよび RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を出す必要があります。

**LIST=symbol**

**LIST=(r2-r12)**

(必須) -- 呼び出し元が作成した更新リソース・リストを指定します。各リスト項目が、それぞれ別々の更新要求です。リスト長は、リスト項目の数と、リスト項目に DATA2 が入っているかに応じて異なります。

CSLRUPDL マクロは、更新リソース・リスト項目をマップします。そのリストには、ヘッダーと 1 つ以上のリスト項目が含まれます。リスト項目は、連続するストレージに置かなければなりません。各更新リスト項目の内容は、次のとおりです。

- 項目の長さ -- 更新リスト項目の長さ。リスト項目長は、DATA2 が指定されているかどうかによって異なります。
- リソース名 -- リソースのクライアント定義の名前。
- リソース・タイプ -- リソース・タイプは、リソース構造上のリソースのクライアント定義の物理的グループ化です。有効な値は 1 から 255 です。
- 更新オプション -- 更新について実行する特殊な処理を示すオプション。
- バージョン -- リソース・バージョンであって、リソースが更新された回数。更新が成功するためには、バージョンは、既存のリソースの場合、リソースのバージョンと一致しなければなりません。リソースを作成するには、バージョンはゼロでなければなりません。
- 所有者 -- リソースの所有者。

- DATA1 -- リソースが更新されるための、少量のクライアント・データ (固定長で、データ項目の付属域に含まれる)。
- DATA2 の長さ -- DATA2 が指定されている場合、DATA2 の長さ。
- DATA2 -- 更新されるリソースに関連付けられた多量のクライアント・データ (可変長で、データ項目の 1 つ以上のデータ・エレメントに含まれる)。DATA2 はオプションです。DATA2 の最大サイズは 61312 バイト (X'EF80') です。

**LISTLEN=***symbol*

**LISTLEN=(r2-r12)**

(オプション) -- 4 バイトの更新リソース・リスト長を指定します。LIST が指定された場合、LISTLEN は必須です。

**OUTLEN=***symbol*

**OUTLEN=(r2-r12)**

(必須) -- CSLRMUPD 要求から戻される出力の長さを受け取る 4 バイト・フィールドを指定します。OUTLEN には、OUTPUT= パラメーターで示された出力の長さが入っています。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力長はゼロです。

**OUTPUT=***output*

**OUTPUT=(r2-r12)**

(必須) -- CSLRMUPD 要求から戻される可変長出力のアドレスを受け取る 4 バイト・フィールドを指定します。その出力には、試みられたリソース更新に対するヘッダーと 1 つ以上の更新項目が含まれます。出力長は、OUTLEN= フィールドに入れて戻されます。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力アドレスはゼロです。

CSLRUPDO マクロは、戻される出力をマップします。その出力には、ヘッダーと 1 つ以上のリスト項目が含まれます。出力ヘッダーの内容は次のとおりです。

- 目印
- 出力長
- CSLRUPDO のバージョン
- タイム・スタンプ
- リソース項目カウント
- CSLRUPDO のヘッダーの長さ (項目の先頭までのオフセット)
- 最小項目長 (DATA2 までのオフセット)

各出力項目は、試みられたリソース更新を表します。各項目の内容は、次のとおりです。

- 出力項目長 -- リスト項目長は、DATA2 が戻されるかどうかによって異なります。
- リソース・タイプ

- 名前タイプ -- 名前タイプは、名前タイプ内のクライアント定義リソース名の一意性を保証する、リソース・タイプに関連付けられた、クライアント定義の値。有効な値は 1 から 255 です。
- リソース名
- バージョン -- 更新が成功した場合、新しいリソース・バージョン。またはバージョン不一致のために更新が失敗した場合、既存リソースのリソース・バージョン。
- 所有者 -- バージョン不一致のために更新が失敗し、所有者を読み取るオプションが設定されていた場合、既存リソースのリソース所有者。
- DATA1 -- バージョン不一致のために更新が失敗し、DATA1 を読み取るオプションが設定されていた場合、既存リソースに関連付けられた少量のクライアント・データ (固定長で、データ項目の付属域に含まれる)。
- DATA2 の長さ -- バージョン不一致のために更新が失敗し、DATA2 が存在していて、DATA2 を読み取るオプションが設定されていた場合、既存のリソースに関連付けられた多量のクライアント・データの長さ。
- DATA2 -- バージョン不一致のために更新が失敗し、DATA2 が存在していて、DATA2 を読み取るオプションが設定されていた場合、既存のリソースに関連付けられた多量のクライアント・データ (可変長で、データ項目の 1 つ以上のデータ・エレメントに含まれる)。DATA2 の最大サイズは 61312 バイト (X'EF80') です。
- 更新要求の完了コード -- 完了コードは、CSLRRR によってマップされます。完了コードには、以下のものがあります。

**X'00000000'**

更新要求は成功しました。

**X'00000008'**

リソース・タイプが無効。

**X'00000010'**

バージョン不一致。リソースはすでに存在していて、入力に指定されたバージョンが一致しませんでした。

**X'00000014'**

リソースは、別のリソース・タイプとしてすでに存在しています。

**X'00000018'**

リソース・タイプは登録されません。リソース・タイプは、CSLRMREG 要求を使って登録しなければなりません。

**X'0000001C'**

リソース構造が満杯です。

**X'00000024'**

リソース構造が利用不能。

**X'00000038'**

CQS 内部エラーのために、更新が失敗しました。

**X'0000003C'**

RM が作成した CQSUPD リスト項目が正しくなかったため、更新は失敗しました。

**X'00000040'**

バージョン不一致。リソースはすでに存在していて、入力に指定されたバージョンが一致しませんでした。要求者は、DATA2 を戻すように要求しましたが、RM は、DATA2 を読み取り中にエラーを検出しました。

出力バッファは、呼び出し元によって事前割り振りされることはありません。要求がこのワードを戻した場合、そこには、更新出力を含むバッファのアドレスが入っています。ストレージの使用が終了したとき、呼び出し元は、CSLSCBFR FUNC=RELEASE 要求を出すことにより、このストレージを解放する必要があります。出力の長さは、OUTLEN= フィールドに入れて戻されません。

**PARM=symbol**

**PARM=(r2-r12)**

(必須) -- CSLRMUPD パラメーター・リストを指定します。パラメーター・リストの長さは、RUPD\_PARMLN でパラメーター・リストの長さとして EQU 値と同じでなければなりません。

**RETCODE=symbol**

**RETCODE=(r2-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。RM 戻りコードは、CSLRRR で定義されています。SCI 戻りコードは、CSLSRRR で定義されています。返される戻りコードについては、以下の表に説明があります。

**RETNAME=symbol**

**RETNAME=(r2-r12)**

(オプション) -- 呼び出し元に戻される RM 名を受け取る 8 バイト・フィールドを指定します。これは、SCI の要求送信先のターゲット RM アドレス・スペースの CSL メンバー名です。

**RETTOKEN=symbol**

**RETTOKEN=(r2-r12)**

(オプション) -- 呼び出し元に戻される RM の SCI トークンを受け取る 16 バイト・フィールドを指定します。これは、SCI の要求送信先のターゲット RM アドレス・スペースの SCI トークンです。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。RM 理由コードは、CSLRRR で定義されています。SCI 理由コードは、CSLSRRR で定義されています。返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol**

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

## CSLRMUPD の戻りコードと理由コード

以下の表は、CSLRMUPD 要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 58. CSLRMUPD の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'03000008'	X'00002000'	クライアントは登録されません。
	X'00002100'	更新リストの長さが無効。
	X'00002108'	更新リストのアドレスが無効。
	X'0000210C'	リスト項目の 1 つに、以下の無効なリスト項目長の 1 つが入っています。 <ul style="list-style-type: none"> <li>長さがゼロ</li> <li>最小リスト項目長より小さい</li> <li>渡されたリストの終わりを超えている</li> <li>フルワード境界上にはない</li> </ul> RM は、リストの残り部分は無効であると想定します。
	X'00002110'	リスト・ヘッダー (UPDL_PVER) 内のリスト・バージョンはゼロであり、これは無効です。リスト・ヘッダー内のリスト・バージョンは、最大リスト・バージョン (UPDL_PVERMAX) に設定されていなければなりません。
	X'00002114'	リスト・ヘッダー長は、ゼロ、または渡されたリスト長より大きくはなりません。リスト・ヘッダー内のリスト・ヘッダー長 (UPDL_HDRLEN) は、リスト・ヘッダー長に設定されていなければなりません。
	X'00002200'	リスト項目の 1 つに、ゼロなどの無効なリソース・タイプが入っています。RM は、リストの残り部分は無効であると想定します。
	X'0000220C'	リスト項目の 1 つに、1 つ以上の無効な更新オプションが入っています。RM は、リストの残り部分は無効であると想定します。
	X'00002404'	リソース構造が定義されていない。
X'0300000C'	X'00003000'	要求は少なくとも 1 つのリスト項目については成功しましたが、すべてについて成功したわけではありません。OUTPUT バッファ内の各リスト項目の完了コードを検査して、個々のエラーを調べてください。
	X'00003004'	要求はすべてのリスト項目について失敗しました。OUTPUT バッファ内の各リスト項目の完了コードを検査して、個々のエラーを調べてください。
	X'00003008'	要求は 1 つ以上のリスト項目について失敗し、すべての失敗はバージョン不一致でした。OUTPUT バッファ内の各リスト項目の完了コードを検査して、個々のエラーを調べてください。

表 58. CSLRMUPD の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
X'03000010'	X'00004000'	CQS アドレス・スペースが利用不能。要求を再試行して、使用可能な CQS のある、別の RM に要求を経路指定してみてください。
	X'00004100'	要求されたバージョンはサポートされていません。クライアントがコンパイルした CSLRMUPD のバージョンは、RM でサポートされていません。新しい CSLRMUPD 機能を使用する新しいリリースに IMS をマイグレーションする前に、すべての RM を新しいリリースにマイグレーションしなければなりません。
X'03000014'	X'00004104'	リストのバージョンはサポートされていません。クライアントが更新リストを作成したバージョン・レベルは、RM でサポートされていません。新しい CSLRMUPD 機能を使用する新しいリリースにクライアントをマイグレーションする前に、すべての RM を新しいリリースにマイグレーションしなければなりません。
	X'00005000'	出力バッファのストレージ割り振りが失敗しました。リソースの更新が成功したかは、わかりません。
	X'00005100'	CQSUPD バッファのストレージ割り振りが失敗しました。
	X'00005200'	CQS 要求が予期しないエラーになりました。
	X'00005204'	RM が作成した要求入力が正しくなかったため、CQS 要求は失敗しました。

関連概念:

208 ページの『グローバル・リソースを管理するための CSL RM 要求の発行』

## CSL RM ディレクティブ

RM ディレクティブは、RM が、RM クライアントへのメッセージとして送信できると定義した機能であって、処理すべき作業を RM クライアントに通知します。リソース処理クライアントが RM に登録された後、RM は CSLSCMSG 要求を発行することにより、そのクライアントに対し RM 機能、すなわち、RM ディレクティブを実行するように指示することができます。リソース処理クライアントは、リソースを管理し、RM を使って、そのようなリソースに関するグローバル情報を管理するシステムです。

RM ディレクティブは、常にメッセージ・プロトコル (PROTOCOL=MSG) で、すなわち、非同期で発行されます。したがって、RM は RM クライアントからの応答を期待せず、応答を待たずに処理を続行します。RM クライアントは、ディレクティブに対してアクションをとるべきかどうかを自分で決定しなければなりません。クライアントが応答しないと、ディレクティブはタイムアウトになります。

CSLRMDIR マクロは RM ディレクティブをマップします。SCI 入力出口ルーチンのフィールド INXP\_MBRPLPTR は、CSLRMDIR パラメーター・リストを指し示しています。

以下の RM ディレクティブは CSLRMDIR マクロ内で定義されます。

- 構造再移植 (RDIR\_STRPOPD)
- 構造障害 (RDIR\_STRFAILD)
- プロセス・ステップ (RDIR\_PRSTEPD)
- プロセス・ステップ応答 (RDIR\_PRESPD)

関連資料:

 [CSL SCI 入力出口ルーチン \(出口ルーチン\)](#)

## CSL RM 構造再移植ディレクティブ

RM が構造の障害を検出し、構造を際割り振りすると、すべてのリソース処理クライアントに構造再移植ディレクティブが送信されます。これにより、クライアントは構造を再移植します。クライアントはこのディレクティブを、登録先のすべての RM から受け取る可能性があります。

クライアントがすでに再移植した後で、構造再移植のディレクティブを複数の RM から受け取った場合、そのディレクティブが同じ構造名と同じバージョンに適用されることを確認した後、その要求を無視してください。

### 構造再移植パラメーター

#### **RDIR\_STRPOP**

構造再移植ディレクティブの先頭を識別します。

#### **RDIR\_STNAMLEN=length**

構造名の長さが入っています。

#### **RDIR\_STNAMPTR=address**

構造名のアドレスが入っています。

#### **RDIR\_STVERLEN=length**

構造バージョンの長さが入っています。

#### **RDIR\_STVERPTR=address**

構造バージョンのアドレスが入っています。

#### **RDIR\_STRPOPLN=length**

構造再移植の長さが入っています。

## CSL RM 構造障害ディレクティブ

リソース構造に障害があって、再割り振りできないときには、リソース処理クライアントに構造障害ディレクティブが送られます。この問題が訂正されるまで、クライアントは、それ以上のリソース要求を発行することはできません。

クライアントは、それが登録されているすべての RM から、このディレクティブを受け取る可能性があります。クライアントが複数の RM からディレクティブを受け取った場合、そのディレクティブが同じ構造名と同じバージョンに適用されることを確認した後、重複した要求を無視することができます。

## 構造障害パラメーター

### **RDIR\_STRFAIL**

構造障害ディレクティブの先頭を識別します。

### **RDIR\_SFNAMELEN=length**

構造名の長さが入っています。

### **RDIR\_SFNAMEPTR=address**

構造名のアドレスが入っています。

### **RDIR\_SFVERLEN=length**

構造バージョンの長さが入っています。

### **RDIR\_SFVERPTR=address**

構造バージョンのアドレスが入っています。

### **RDIR\_STRFAILN=length**

構造障害ディレクティブの長さが入っています。

## CSL RM プロセス・ステップ・ディレクティブ

プロセス・ステップの実行が必要なとき、リソース処理クライアントにプロセス・ステップ (Process Step) ディレクティブが送られます。

## プロセス・ステップ・パラメーター

### **RDIR\_PRSTEP**

プロセス・ステップ・ディレクティブの先頭を識別します。

### **RDIR\_PSTKNLEN=length**

プロセス・トークン (PRCTOKEN) の長さが入っています。プロセス・トークンは、全 IMSplex プロセスを一意的に識別します。PRCTOKEN は、CSLRMPRI FUNC=INITIATE 要求が正常に完了した後、戻されます。PRCTOKEN は、CSLRMPRS FUNC=PROCESS、CSLRMPRR FUNC=RESPOND、および CSLRMPRT FUNC=TERMINATE 要求に指定できます。

### **RDIR\_PSTKNPTR=address**

PRCTOKEN のアドレスが入っています。

### **RDIR\_PSUOWLEN=length**

UOWTOKEN の長さが入っています。プロセス・インスタンスを一意的に識別する、クライアント定義の UOW です。また、UOWTOKEN は、PROCESS INITIATE、PROCESS RESPOND、および PROCESS TERMINATE ステップを結合します。UOWTOKEN は CSLRMPRI FUNC=INITIATE 要求で定義され、CSLRMPRS FUNC=PROCESS 要求で指定できます。

### **RDIR\_PSUOWPTR=address**

UOWTOKEN のアドレスが入っています。

### **RDIR\_PRCNMLEN=length**

プロセス名 (PRCNAME) の長さが入っています。プロセス名は、CSLRMPRI FUNC=INITIATE 要求で定義されます。プロセス名は、CSLRMPRS FUNC=PROCESS および CSLRMPRT FUNC=TERMINATE 要求でも指定できます。

**RDIR\_PRCNMPTR=address**

PRCNAME のアドレスが入っています。

**RDIR\_PRCTYPE**

プロセス・タイプは、CSLRMPRI FUNC=INITIATE 要求で定義されます。プロセス・タイプは、CSLRMPRS FUNC=PROCESS および CSLRMPRT FUNC=TERMINATE 要求でも指定できます。このパラメーターは、値で渡されます。長さフィールドは常にゼロです。

**RDIR\_PSNAME**

プロセス・ステップ名が入っています。プロセス・ステップ名は、CSLRMPRS FUNC=PROCESS 要求で定義されます。このパラメーターは、値で渡されます。長さフィールドは常にゼロです。

**RDIR\_PSDATLEN=length**

プロセス・ステップ・クライアント・データの長さ (CDATALEN) が入っています。クライアント・データは、プロセス・ステップの参加者に渡されます。CDATALEN は CSLRMPRS FUNC=PROCESS 要求で指定されます。

**RDIR\_PSDATPTR=address**

プロセス・ステップ・クライアント・データのアドレス (CDATA) が入っています。

**RDIR\_CNAMLEN=length**

プロセス・ステップを開始したクライアント (プロセス・ステップ・マスター) が SCI に登録したクライアント名の長さが入っています。

**RDIR\_CNAMPTR=address**

プロセス・ステップを開始したクライアント (プロセス・ステップ・マスター) が SCI に登録したクライアント名のアドレスが入っています。

**RDIR\_CTYPE**

プロセス・ステップを開始したクライアント (プロセス・ステップ・マスター) が SCI に登録したクライアント・タイプを識別します。このパラメーターは、値で渡されます。長さフィールドは常にゼロです。

**RDIR\_CSTYPLEN**

プロセス・ステップを開始したクライアント (プロセス・ステップ・マスター) が SCI に登録したクライアント・サブタイプの長さが入っています。

**RDIR\_CSTYPPTR**

プロセス・ステップを開始したクライアント (プロセス・ステップ・マスター) が SCI に登録したクライアント・サブタイプのアドレスが入っています。

**RDIR\_PRSTEPLN**

プロセス・ステップ・ディレクティブの長さが入っています。

## CSL RM プロセス・ステップ応答ディレクティブ

プロセス・ステップ応答 (Process Step Response) ディレクティブは、CSLRMPRR 要求でプロセス・ステップに応答しているクライアントによって RM に送られます。

## プロセス・ステップ応答パラメーター

### **RDIR\_PRESP**

プロセス・ステップ応答ディレクティブの先頭を識別します。

### **RDIR\_PRTKNLEN=length**

プロセス・トークン (PRCTOKEN) の長さが入っています。プロセス・トークンは、全 IMSplex プロセスを一意的に識別します。PRCTOKEN は、CSLRMPRI FUNC=INITIATE 要求が正常に完了した後、戻されます。PRCTOKEN は、CSLRMPRS FUNC=PROCESS、CSLRMPRR FUNC=RESPOND、および CSLRMPRT FUNC=TERMINATE 要求に指定できます。

### **RDIR\_PRTKNPTR=address**

PRCTOKEN のアドレスが入っています。

### **RDIR\_PROUTLEN=length**

プロセス・ステップ応答出力 (OUTPUT) の長さが入っています。応答出力は、プロセス・ステップの開始元に渡されます。OUTPUT は、CSLRMPRR FUNC=RESPOND 要求で指定されます。

### **RDIR\_PROUTPTR=address**

応答出力 (OUTPUT) のアドレスが入っています。

### **RDIR\_PRRCLN=length**

プロセス・ステップ応答戻りコード (RQSTRC) が入っています。戻りコードは、CSLRMPRR FUNC=RESPOND 要求で指定されます。

### **RDIR\_PRRCPTR=address**

プロセス・ステップ応答戻りコード (RQSTRC) のアドレスが入っています。

### **RDIR\_PRRSNLEN=length**

プロセス・ステップ応答理由コード (RQSTRSN) の長さが入っています。これは、CSLRMPRR FUNC=RESPOND 要求で指定されます。

### **RDIR\_PRRSNPTR=address**

プロセス・ステップ応答理由コード (RQSTRSN) のアドレスが入っています。



---

## 第 8 章 CSL SCI クライアントの作成

IMSplex メンバーとして IMSplex に参加して、他の IMSplex メンバーとの通信を可能にするプログラムを作成するには、SCI (Structured Call Interface) との接続を確立する必要があります。SCI に接続されなくては、プログラムは IMSplex に参加することはできず、他の IMSplex メンバーと通信することはできません。

SCI との接続を確立するには、SCI 要求のサブセットを使用できます。これらの要求は、SCI との接続を確立または終了させ、オプションで、IMSplex メンバーが作動可能状態であることを SCI に示します。メンバーは、作動可能状態のとき、要求およびメッセージをタイプで送信してもらうことができます。

SCI 要求によって、IMSplex メンバーは他の IMSplex メンバーと通信することもできるし、他の IMSplex メンバーに関する情報を得ることもできます。IMSplex メンバーは、メッセージ、要求、および要求への応答を送信する SCI 要求を使用することにより、他のメンバーと通信します。照会要求を使用して、IMSplex の他のメンバーに関する情報を入手できます。

関連資料:

266 ページの『CSLSCQRY: 照会要求』

---

### CSL SCI 要求の順序

SCI への登録、メンバーの準備、メンバーに割り振られたストレージの解放、メンバーの静止、および SCI からのメンバーの登録解除を正常に行うためには、Structured Call Interface (SCI) 要求を特定の順序で発行する必要があります。

最初に行う SCI 要求は CSLSCREG です。次に、このメンバーは CSLSCRDY を出して、メンバー・タイプで経路指定されるメッセージおよび要求を受け取る準備ができたことを SCI に伝えます。SCI によって割り振られたストレージがある場合 (例えば、メッセージまたは SCI 割り振り済み出力パラメーターが受信されます)、そのメンバーは、SCI バッファ解放要求 CSLSCBFR を出すことにより、ストレージを解放します。

終了準備ができたら、SCI 静止要求 CSLSCQSC を出して、そのメンバーは、メンバー・タイプで経路指定されたメッセージおよび要求を受け取らないことを SCI に伝えます。SCI 登録解除要求 CSLSCDRG を発行して、SCI との接続を終了させると、それ以降、そのメンバーは IMSplex に参加できなくなります。

以下の表に、SCI クライアントが発行する要求の順序を示します。要求の目的も併記します。

表 59. SCI クライアントの要求のシーケンス

要求	目的
CSLSCREG	SCI に登録する。これにより、SCI との接続が確立され、メンバーは IMSplex 内で通信できるようになります。

表 59. SCI クライアントの要求のシーケンス (続き)

要求	目的
CSLSCRDY	SCI に対してメンバーを作動可能にする。これにより、SCI は、メンバー・タイプで経路指定されたメッセージおよび要求をこのメンバーに経路指定できるようになります。
CSLSCBFR	SCI によってメンバーに割り振られているストレージを解放する (例えば、SCI によって要求から割り振られたメッセージ・データまたはパラメーター)。
CSLSCQSC	SCI に対してメンバーを静止させる。これは SCI に、メンバー・タイプで経路指定されたメッセージおよび要求をこのメンバーに経路指定しないように伝えます。
CSLSCDRG	メンバーを SCI から登録解除する。これにより、このメンバーと SCI の接続は終了します。

## 拡張 CSL SCI 要求

Structured Call Interface (SCI) との接続後、IMSplex メンバーは拡張 SCI 要求を使用して、他の IMSplex メンバーにサービスを要求したり、これらのメンバーに関する情報を調べたりすることができます。拡張要求には、それぞれ固有の目的があります。

SCI との接続が確立できたら、IMSplex メンバーは、拡張 SCI 要求を使用することにより次のことができます。

- 他の IMSplex メンバーとの間で通信したり、サービスを要求したりする。

IMSplex メンバー間の通信用にメッセージ・プロトコルと要求プロトコルが提供されています。メッセージは、他の IMSplex メンバーとの片方向通信です。要求は、要求側メンバーに応答が戻されなければなりません。

- IMSplex 内の他のメンバーに関する情報を入手する。

照会要求 CSLSCQRY により、IMSplex メンバーは、IMSplex の他のメンバーが誰であるかを知り、これらの IMSplex メンバーに関する情報を入手することができます。

以下の表に、拡張 SCI 要求と、その目的を併記します。これらの要求は、順序にこだわらずに発行できます。ただし、要求を出す IMSplex メンバーは SCI に登録済みでなければなりません。

表 60. IMSplex メンバー用の拡張 SCI 要求

要求	目的
CSLSCMSG	他の IMSplex メンバーに片方向メッセージを送信する。
CSLSCRQS	他の IMSplex メンバーに要求を送信する。SCI は、要求への応答を期待します。
CSLSCRQR	前に発行された要求への応答を送信する。
CSLSCQRY	SCI に照会を出し、IMSplex のメンバーに関する情報を入手する。

## CSL SCI 要求

SCI 要求は、IMSpIex メンバーから発行できます。接続が確立された後は、どのメンバーでも、他のすべての IMSpIex メンバーからのメッセージを受信できます。

関連概念:

157 ページの『CSL OM 自動化操作プログラム・クライアント』

### CSLSCBFR: バッファース返し要求

CSLSCBFR 要求は、Structured Call Interface (SCI) が IMSpIex メンバーに割り振ったストレージを解放します。このストレージは、CSLSCMSG 要求で他の IMSpIex メンバーから送られた入力メッセージ、または CSLSCRQS 要求で生成された出力パラメーターを受け取るために割り振られます。

別のマクロが、そのマクロが生成するコードの一部として CSLSCRQS 要求を呼び出し、CSLSCRQS に SCI データ・タイプを返させることができます。これらのパラメーター用に割り振られたストレージは、CSLSCBFR マクロで解放しなければなりません。これを行う SCI マクロの例には、CSLSCQRY マクロがあります。OUTPUT パラメーターは、ストレージ内の、CSLSCQRY マクロからの出力を入れるバッファのアドレスを受け取るためのアドレスを指定します。このストレージを解放するには、CSLSCBFR 要求を使用します。

#### 構文

##### DSECT 構文

CSLSCBFR 要求の DSECT 機能は、CSLSCBFR パラメーター・リストの長さ、および CSLSCBFR 要求の戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

```
▶▶—CSLSCBFR—FUNC=DSECT—▶▶
```

##### RELEASE の構文

SCI メッセージ・バッファまたは SCI データ・タイプ・バッファを解放するには、CSLSCBFR 要求の RELEASE 機能を使用します。SCI データ・タイプ・バッファは、SCI がストレージを割り振った CSLSCRQS 要求の、選択された出力パラメーター用に使用されます。

```
▶▶—CSLSCBFR—FUNC=RELEASE—PARM=parm—SCITOKEN=scitoken—  
└─BUFFER=buffer—  
└─BUFFERPTR=buffer—▶▶
```

```
▶▶—RETCODE=returncode—RSNCODE=reasoncode—▶▶
```

CSLSCMSG 要求で生成されたメッセージの場合、バッファ・アドレスは、入力出口パラメーター・リストの INXP\_MBRPLPTR フィールドでメンバー入力出口に対して指定されたメンバー・パラメーター・リストのアドレスとなります。

SCI データ・タイプ・バッファを使用する CSLSCRQS 要求で生成される応答の場合、ストレージは、要求が元の要求を開始した IMSpIex メンバーに戻されるとき

に割り振られます。バッファ・アドレスはこのストレージのアドレスであり、それは、メンバーが要求に指定したフィールドに入れて戻されます。

CSLSCBFR 要求の完了後は、メッセージ・バッファまたは要求応答に入っていたストレージは、それ以降、IMSpIex メンバーからアクセスすることはできません。

無許可メンバーの場合、ストレージを、SCI 登録呼び出し元の JOBSTEP TCB の下の TCB から解放する必要があります。登録済み JOBSTEP TCB の下でない TCB から解放しようとしても、失敗します。

## パラメーター

**BUFFER=***symbol*

**BUFFER=(r1-r12)**

解放されるバッファのアドレスが入っている 4 バイト・パラメーター。

BUFFER または BUFFERPTR のいずれかが必要です。

**BUFFERPTR=***symbol*

**BUFFERPTR=(r1-r12)**

解放されるバッファのアドレスが入っているワード (ストレージ内) へのアドレスが入っている、4 バイト・パラメーター。

BUFFER または BUFFERPTR のいずれかが必要です。

**PARM=***symbol*

**PARM=(r1-r12)**

CSLSCBFR パラメーター・リストを指定します。パラメーター・リストの長さは、SBFR\_PARMLN でパラメーター・リストの長さとして定義された EQU 値と同じでなければなりません。

**RETCODE=***symbol*

**RETCODE=(r1-r12)**

出力の戻りコードを受け取る 4 バイト・フィールドを指定します。SCI 戻りコードは、CSLSRR で定義されています。CSLSCBFR で返される戻りコードについては、以下の表に説明があります。

**RSNCODE=***symbol*

**RSNCODE=(r1-r12)**

出力の理由コードを受け取る 4 バイト・フィールドを指定します。SCI 理由コードは、CSLSRR で定義されています。CSLSCBFR で返される理由コードについては、以下の表に説明があります。

**SCITOKEN=***symbol*

**SCITOKEN=(r1-r12)**

SCITOKEN が入っている 16 バイト・フィールドを指定します。このトークンは、この IMSpIex メンバーと SCI の接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

## 戻りコードおよび理由コード

以下の表は、CSLSCBFR マクロ要求で返される戻りコードと理由コードの一覧です。

表 61. CSLSCBFR 要求の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'01000008'	X'00002014'	解放しようとしたバッファーは SCI バッファーではなかった。
	X'00002018'	SCI トークンが無効。
	X'00002038'	パラメーター・リスト・バージョンが無効。
	X'00002054'	解放しようとしたバッファーは割り振られたバッファーではなかった。
X'01000010'	X'00004FFF'	機能がサポートされていない。
X'01000014'	X'00005000'	SCI 内部エラーが発生した。
	X'00005074'	CSLSCBFR 呼び出しでバッファー接頭部が損傷している。
	X'00005078'	CSLSCBFR 要求で、SCI バッファーの STORAGE RELEASE が失敗した。
	X'00005500'	CSLSCBFR 処理中に異常終了が起きた。

## CSLSCDRG: 登録解除要求

CSLSCDRD 要求を発行することで、IMSplex メンバーと SCI 間の接続を終了できます。この要求の正常終了後は、それ以降、SCI トークンは無効です。

それ以降の SCI 要求を行うには、IMSplex メンバーは、CSLSCREG 要求によって、SCI との間に新規接続を作成する必要があります。

### CSLSCDRG の構文

#### CSLSCDRG DSECT の構文

CSLSCDRG 要求の DSECT 機能は、CSLSCDRG パラメーター・リストの長さ、および戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

```
▶▶—CSLSCDRG—FUNC=DSECT—▶▶
```

#### CSLSCDRG DEREGISTER の構文

CSLSCDRG FUNC=DEREGISTER 要求は、SCI から IMSplex メンバーを登録解除します。CSLSCDRG FUNC=DEREGISTER 要求の正常終了後は、SCITOKEN は無効です。

```
▶▶—CSLSCDRG—FUNC=DEREGISTER—PARM=parm—SCITOKEN=scitoken—▶▶
```

```
▶—RETCODE=returncode—RSNCODE=reasoncode—▶▶
```

### CSLSCDRG のパラメーター

PARM=symbol

**PARM=(r1-r12)**

(必須) -- CSLSCDRG パラメーター・リストを指定します。パラメーター・リストの長さは、SDRG\_LN で定義されたパラメーター・リスト長 EQU 値と同じでなければなりません。

**RETCODE=symbol****RETCODE=(r1-r12)**

(必須) -- CSLSCDRG の戻りコードを受け取る 4 バイト・フィールドのアドレスを指定します。SCI 戻りコードは、CSLSRR で定義されています。CSLSCDRG で返される戻りコードについては、以下の表に説明があります。

**RSNCODE=symbol****RSNCODE=(r1-r12)**

(必須) -- CSLSCDRG の理由コードを受け取る 4 バイト・フィールドのアドレスを指定します。SCI 理由コードは、CSLSRR で定義されています。CSLSCDRG で返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol****SCITOKEN=(r1-r12)**

(必須) -- SCITOKEN が入っている 16 バイト・フィールドを指定します。このトークンは、この IMSplex メンバーと SCI の接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

**CSLSCDRG の戻りコードと理由コード**

以下の表は、CSLSCDRG マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 62. CSLSCDRG の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'01000004'	X'00001010'	メンバーの z/OS システム間カップリング・ファシリティー leave が失敗した。
X'01000008'	X'00002018'	SCI トークンが無効。
	X'00002038'	パラメーター・リスト・バージョンが無効。
X'01000010'	X'00004000'	SCI がアクティブでない。
	X'00004014'	CSLSDR00 をロードできなかった。
	X'00004018'	未解決の登録解除中の要求があります。
	X'00004FFF'	機能がサポートされていない。
X'01000014'	X'00005000'	SCI 内部エラーが発生した。
	X'00005004'	SCI は ESTAE ルーチンを追加できなかった。
	X'00005008'	BPE SVC エラーが発生した。
	X'00005020'	ENQ リソース・エラーが発生した。
	X'00005500'	CSLSCDRG 処理中に異常終了が起きた。

## CSLSCMSG: メッセージ送信要求

CSLSCMSG 要求を発行することで、1 つ以上の他の IMSplex メンバーにメッセージを送信することができます。ターゲット・メンバーは、SCITOKEN、メンバー名、またはメンバー・タイプによって指定されます。

### CSLSCMSG の構文

#### CSLSCMSG DSECT の構文

CSLSCMSG 要求の DSECT 機能は、CSLSCMSG パラメーター・リストの長さ、IMSplex タイプ、および CSLSCMSG 戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶ CSLSCMSG—FUNC=DSECT

#### CSLSCMSG SEND MESSAGE の構文

CSLSCMSG FUNC=SEND 要求の構文を以下に示します。

▶▶ CSLSCMSG—FUNC=SEND—SCITOKEN=*scitoken*—PARAM=*parm*—MBRPARAM=*mbrparmlist*

▶ MBRPCNT=*mbrparmcount*—MBRFUNC=*mbrfunctioncode*

┌ MBRFUNC=*mbrsubfunctioncode* ─┐ ┌ FUNCTYPE=DEST ─┐  
└────────────────────────────────┘ └──────────────────────────┘  
┌──────────────────────────────────┐  
└──────────────────────────────────┘ FUNCTYPE=SENDER

┌ LISTLEN=*listlength* ─┐ ┌ TOKENLIST=*tokenlist* ─┐ ┌ RETCODE=*returncode* ─┐  
└──────────────────────────┘ └──────────────────────────┘ └──────────────────────────┘  
┌ NAMELIST=*namelist* ─┐ ┌ TYPELIST=*typelist* ─┐ B  
└──────────────────────────┘ └──────────────────────────┘  
┌ TOKEN=*scitoken* ─┐  
└──────────────────┘  
┌ NAME=*membername* ─┐  
└──────────────────┘  
A

▶ RSNCODE=*reasoncode* ─┐ ┌ RETNAME=*returnname* ─┐ ┌ RETTOKEN=*returntoken* ─┐

#### A:

┌ TYPE=*membertypecode* ─┐ B ┌──┐  
└──────────────────┘  
┌ TYPE='AOP' ─┐  
└──────────┘  
┌ TYPE='BATCH' ─┐  
└──────────┘  
┌ TYPE='CQS' ─┐  
└──────────┘  
┌ TYPE='DBRC' ─┐  
└──────────┘  
┌ TYPE='IMS' ─┐  
└──────────┘  
┌ TYPE='IMSCON' ─┐  
└──────────┘  
┌ TYPE='ODBM' ─┐  
└──────────┘  
┌ TYPE='OM' ─┐  
└──────────┘  
┌ TYPE='OTHER' ─┐  
└──────────┘  
┌ TYPE='RM' ─┐  
└──────────┘  
┌ TYPE='SCI' ─┐  
└──────────┘

**B:**



## CSLSCMSG のパラメーター

**FUNCTYPE=SENDER**

**FUNCTYPE=DEST**

(オプション) -- このメッセージの DEST (宛先) またはメッセージの SENDER によって MBRFUNC および MBRSFUNC が定義されたことを指定します。この標識は、SCI 入力出口パラメーター・リストに入れられてメッセージの受信者に渡されます。

**LISTLEN=<numeric literal>**

**LISTLEN=symbol**

**LISTLEN=(r1-r12)**

(NAMELIST、TOKENLIST、または TYPELIST が指定されている場合、必須) -- 経路指定リストの長さを指定します。経路指定リストは、ヘッダーと 1 つ以上のリスト項目で構成されます。各項目は、単一のメッセージ宛先 (NAMELIST および TOKENLIST) または宛先セット (TYPELIST) を記述しています。

LISTLEN が数値リテラルの場合、すべての文字が数字でなければなりません。いずれかの文字が英字の場合、パラメーターはシンボルと見なされます。

**MBRFUNC=symbol**

**MBRFUNC=(r1-r12)**

(必須) -- SCI 入力出口パラメーター・リストに入れられてメッセージの宛先に渡される 4 バイト・メンバー機能コードを指定します。この機能コードは、MBRSFUNC とともに、送信されるメッセージを識別します。

MBRFUNC がシンボルの場合、そのシンボルは、機能コードが入ったストレージの 4 バイト域を指します。

**MBRPARM=symbol**

**MBRPARM=(r1-r12)**

(必須) -- 事前に作成されたパラメーター・リストのアドレスを指定します。このパラメーター・リストはメッセージング・モジュールで作成する必要があり、いくつかのペアで構成されます。個々のペアは、メンバー・パラメーター・リスト内の 1 つのパラメーターの記述であり、次のもので構成されます。

パラメーター長

メンバー・パラメーターの長さを指定する 4 バイト・パラメーター。

パラメーター・アドレス

メンバー・パラメーターのアドレスを指定する 4 バイト・パラメーター。

パラメーター・リストでパラメーターを渡す方法は、アドレスによる方法 と、値による方法 の 2 通りがあります。このどちらの方法も、CSLSCMSG 要求で

パラメーターを渡すときに使用できます。長さとおアドレスのペアを、SCI がパラメーターを正しく扱えるようにセットアップしなければなりません。

- アドレスによる方法

パラメーターをアドレスによって渡すには、パラメーターのアドレスを *parameteraddress* に入れて渡し、パラメーターの長さを *parameterlength* に入れて渡します。SCI は、*parameteraddress* からパラメーターを入手します。

- 値による方法

パラメーターを値によって渡すには、パラメーターを *parameteraddress* に入れて渡し、*parameterlength* にゼロを入れて渡します。長さがゼロの場合、SCI は、*parameteraddress* に入っている値を宛先にコピーします。

メンバー・パラメーター・リスト: ここに指定されるユーザー・パラメーターは、メンバー・パラメーター・リスト内の、メッセージを受け取る IMSplex メンバーに提示され、そのアドレスは、入力出力パラメーター域のフィールド INXP\_MBRPLPTR に入っています。各パラメーターは 8 バイトで構成され、先頭 4 バイトは *parameterlength* を示し、2 番目の 4 バイトは *parameteraddress* を示します (*parameteraddress* がアドレスの場合、2 番目の 4 バイトは、ローカル・アドレス・スペース内のストレージを示します。要求側のアドレス・スペースではありません)。

**NULL** パラメーター: 場合によっては、メッセージ処理モジュールは、定められた数のパラメーターを、定義された順で期待することがあります。すべてのパラメーターが入っていないメッセージを送る場合、期待されているすべてがデータ・バッファーに入っているように NULL パラメーターを送らなければなりません。NULL パラメーターは、*parameterlength* および *parameteraddress* にゼロを指定することにより、送ることができます。データ・バッファー内のパラメーターを表す 8 バイトには、ゼロが入ります。

**MBRPCNT=symbol**

**MBRPCNT=(r1-r12)**

(必須) -- MBRPARG に含まれるメンバー・パラメーターの数が入っている 4 バイト・フィールドを指定します。

**MBRSFUNC=symbol**

**MBRSFUNC=(r1-r12)**

(オプション) -- SCI 入力出力パラメーター・リストに入れられてメッセージの宛先に渡される 4 バイト・メンバー副次機能コードを指定します。この副次機能コードは、MBRFUNC とともに、送信されるメッセージを識別します。

MBRSFUNC がシンボルの場合、そのシンボルは、副次機能コードが入ったストレージの 4 バイト域を指します。

**NAME=symbol**

**NAME=(r1-r12)**

(オプション) -- このメッセージの宛先の 8 バイト・メンバー名のアドレスを指定します。この名前は、通知出口から入手できます (メンバーが IMSplex に参加している場合)。または、CSLSCQRY メッセージを出すことによっても入手できます。

注: 経路指定パラメーターの 1 つ (NAME、TOKEN、TYPE、NAMELIST、TOKENLIST、または TYPELIST) を指定しなければなりません。

NAME で経路指定する場合、宛先メンバーは許可されている必要があります。メンバーが許可されていない場合、メッセージは送信されません。

**NAMELIST=symbol**

**NAMELIST=(r1-r12)**

(オプション) -- このメッセージの経路指定先のメンバー名のリストのアドレスを指定します。このリストには、ヘッダーと 1 つ以上のリスト項目が含まれます。各項目は、1 つのメンバー名を定義します。NAMELIST を指定する場合、LISTLEN も指定する必要があります。

注: 経路指定パラメーターの 1 つ (NAME、TOKEN、TYPE、NAMELIST、TOKENLIST、または TYPELIST) を指定しなければなりません。

リスト・ヘッダー DSECT は CSLSMGLH であり、リスト項目 DSECT は CSLSNMLE です。これらの DSECT は、CSLSCMAP で定義されます。

NAMELIST を使ってメッセージをメンバーに経路指定する場合、そのメンバーは許可メンバーでなければなりません。NAMELIST に非許可メンバーのメンバー名が含まれていると、その名前は見つからず、メッセージは、そのメンバーに送信されません。

NAMELIST は SCI に送られて処理されます。その後、制御はプログラムに戻されます。「Request completed successfully (要求は正常に完了した)」という応答は、メッセージがリストのすべての名前に送られたことを意味するものではありません。それは、リストが SCI に正常に送られたことを意味します。リストが処理されて、メッセージが送られるときに、エラーが起こる可能性があります。次のようなエラーが考えられます。

- 名前が見つからなかった
- 名前は見つかったが、メッセージが送信される前にメンバーが終了した
- SCI が異常終了した

これらのエラーは、プログラムには戻されません。

**PARM=symbol**

**PARM=(r1-r12)**

(必須) -- パラメーターを SCI に渡すためにメッセージが使用するパラメーター・リストのアドレスを指定します。ストレージの長さは、少なくとも SMSG\_LN の値と等しくなければなりません。ストレージは、ワード境界から始まらなければなりません。

**RETCODE=symbol**

**RETCODE=(r1-r12)**

(必須) -- CSLSCMSG の戻りコードを受け取る 4 バイト・フィールドのアドレスを指定します。SCI 戻りコードは、CSLSRR で定義されています。

**RETNAME=symbol**

**RETNAME=(r1-r12)**

(オプション) -- このメッセージの送信先の IMSplex メンバーの名前を受け取る 8 バイト・フィールドのアドレスを指定します。メッセージが複数の宛先に送られる場合、このフィールドには何も戻されません。

**RETTOKEN=***symbol*

**RETTOKEN=(r1-r12)**

(オプション) -- このメッセージの送信先の IMSplex メンバーのトークンを受け取る 8 バイト・フィールドのアドレスを指定します。メッセージが複数の宛先に送られる場合、このフィールドには何も戻されません。

**ROUTE=**ANY

**ROUTE=**ALL

**ROUTE=**LOCAL

(オプション) -- TYPE パラメーターで指定されたタイプ、または TYPELIST パラメーターで指定されたタイプにメッセージをどのようにして経路指定すべきかを指定します。このパラメーターは、TYPE または TYPELIST が指定されている場合のみ有効です。

**ANY**

指定されたタイプの 1 つのメンバーにメッセージを送ります。SCI が、メッセージを受け取るメンバーを選択します。TYPE=ANY は、TYPELIST が指定された場合、無効です。

**ALL**

メッセージを、指定されたタイプの全メンバーに経路指定する。

**LOCAL**

メッセージを、ローカル z/OS イメージ上でアクティブな、指定されたタイプの全メンバーに経路指定する。

**RSNCODE=***symbol*

**RSNCODE=(r1-r12)**

(必須) -- CSLSCMSG の理由コードを受け取る 4 バイト・フィールドのアドレスを指定します。SCI 理由コードは、CSLSRR で定義されています。

**SCITOKEN=***symbol*

**SCITOKEN=(r1-r12)**

(必須) -- 要求を出すメンバーの SCI トークンが入っている 16 バイト・フィールドのアドレスを指定します。トークンは CSLSCREG 要求で戻されました。

**TOKEN=***symbol*

**TOKEN=(r1-r12)**

(オプション) -- このメッセージの宛先の 16 バイト SCI トークンのアドレスを指定します。このトークンは、通知出口から入手できます (メンバーが IMSplex に参加している場合)。または、CSLSCQRY メッセージを出すことによっても、入手できます。

注: 経路指定パラメーターの 1 つ (NAME、TOKEN、TYPE、NAMELIST、TOKENLIST、または TYPELIST) を指定しなければなりません。

**TOKENLIST=***symbol*

**TOKENLIST=(r1-r12)**

(オプション) -- このメッセージの経路指定先のメンバーを表す、SCI トークンのリストのアドレスを指定します。このリストには、ヘッダーと 1 つ以上のリスト項目が含まれます。各項目は、1 つの SCI トークンを定義します。TOKENLIST を指定する場合、LISTLEN も指定する必要があります。

注: 経路指定パラメーターの 1 つ (NAME、TOKEN、TYPE、NAMELIST、TOKENLIST、または TYPELIST) を指定しなければなりません。

リスト・ヘッダー DSECT は CSLSMGLH であり、リスト項目 DSECT は CSLSTKLE です。これらの DSECT は、CSLSCMAP で定義されます。

TOKENLIST は SCI に送られて処理されます。その後、制御はプログラムに戻されます。「Request completed successfully (要求は正常に完了した)」という応答は、メッセージがリストのすべての SCI トークンに送られたことを意味するものではありません。それは、リストが SCI に正常に送られたことを意味します。リストが処理されて、メッセージが送られるときに、エラーが起こる可能性があります。次のようなエラーが考えられます。

- トークンが見つからなかった
- トークンは見つかったが、メッセージが送信される前にメンバーが終了した
- SCI が異常終了した

これらのエラーは、プログラムには戻されません。

**TYPE=***symbol*

**TYPE='AOP'**

**TYPE='BATCH'**

**TYPE='CQS'**

**TYPE='DBRC'**

**TYPE='IMS'**

**TYPE='IMSCON'**

**TYPE='ODBM'**

**TYPE='OM'**

**TYPE='OTHER'**

**TYPE='RM'**

**TYPE='SCI'**

(オプション) -- TYPE は、このメッセージの宛先の SCI タイプを指定します。SCI は、指定されたタイプの 1 つ以上のメンバーにメッセージを経路指定します (経路指定パラメーターの値に依存します)。指定されたタイプのメンバーがない場合、エラーが戻されます。

注: 経路指定パラメーターの 1 つ (NAME、TOKEN、TYPE、NAMELIST、TOKENLIST、または TYPELIST) を指定しなければなりません。

このパラメーターをリテラルとして渡す場合は、そのリテラルを単一引用符で囲む必要があります。このパラメーターをシンボルまたはレジスターとして渡す場合、そのシンボルまたはレジスターは、メンバー・タイプ・コードを表していなければなりません。メンバー・タイプ・コードは、CSLSTPIX マクロを使って入手することができます。

**TYPELIST=***symbol*

**TYPELIST=(r1-r12)**

(オプション) -- このメッセージの経路指定先のメンバー・タイプのリストのアドレスを指定します。このリストには、ヘッダーと 1 つ以上のリスト項目が含まれます。各項目は、1 つの SCI トークンを定義します。TYPELIST を指定する場合、LISTLEN も指定する必要があります。

注: 経路指定パラメーターの 1 つ (NAME、TOKEN、TYPE、NAMELIST、TOKENLIST、または TYPELIST) を指定しなければなりません。

リスト・ヘッダー DSECT は CSLSMGLH であり、リスト項目 DSECT は CSLSTPLE です。これらの DSECT は、CSLSCMAP で定義されます。

TYPELIST は SCI に送られて処理されます。その後、制御はプログラムに戻されます。「Request completed successfully (要求は正常に完了した)」という応答は、メッセージがリストのすべてのタイプに送られたことを意味するものではありません。それは、リストが SCI に正常に送られたことを意味します。リストが処理されて、メッセージが送られるときに、エラーが起こる可能性があります。次のようなエラーが考えられます。

- 指定されたタイプの、アクティブなメンバーがない
- 指定されたタイプのメンバーが見つかったが、メッセージが送信される前に終了した
- SCI が異常終了した

これらのエラーは、プログラムには戻されません。

## CSLSCMSG の戻りコードと理由コード

以下の表は、CSLSCMSG マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 63. CSLSCMSG の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'01000008'	X'00002004'	SCI インターフェース PC ルーチンに無効な関数が渡された。
	X'00002008'	渡されたパラメーターの数は、ゼロ以下か、または許容最大値より大きい。
	X'00002010'	無効なタイプが渡された。
	X'00002018'	SCI トークンが無効。
	X'00002024'	PHDR の長さが無効。
	X'00002028'	経路指定データの長さが無効。
	X'00002034'	非許可呼び出し元としては、パラメーターの長さが大きすぎる。
	X'00002038'	パラメーター・リスト・バージョンが無効。
X'01000010'	X'00004000'	SCI がアクティブでない。
	X'0000400C'	宛先 IMSplex メンバーがアクティブでない。要求されたメンバーは、名前、トークン、またはタイプにより指定済みの可能性があります。
	X'0000401C'	呼び出し元のメンバーは、SCI からの登録解除処理中です。
	X'00004FFF'	機能がサポートされていない。
X'01000014'	X'00005000'	SCI 内部エラーが発生した。
	X'00005004'	ESTAE 追加エラーが発生した。
	X'00005024'	SRB ルーチンのエラーが発生した。

表 63. CSLSCMSG の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'00005028'	経路指定タイプが無効。
	X'0000502C'	内部 BPE ハッシュ・テーブル・サービス・エラーのためにメンバーが見つからなかった。
	X'00005030'	SCI バッファが取得できなかった。
	X'00005034'	SCI アドレス・スペース内のキー 7 バッファが PHDR とパラメーターのコピー用に取得できなかった。
	X'00005038'	IEAMSCHD エラーが発生した。SRB はターゲット・アドレス・スペースに対してスケジュールできなかった。
	X'0000504C'	メッセージ SRB キー 7 パラメーター域が取得できなかった。
	X'00005500'	CSLSCMSG 処理中に異常終了が起きた。
	X'00005504'	メンバー・パラメーターをターゲット・アドレス・スペースにコピーしたとき、異常終了が起きた。

関連資料:

259 ページの『CSLSCMSG: メッセージ送信要求』

271 ページの『CSLSCRDY: レディー要求』

## CSLSCQRY: 照会要求

CSLSCQRY 要求を発行することで、IMSplex メンバーは、IMSplex のメンバーに関する情報を入手することができます。

### CSLSCQRY の構文

#### CSLSCQRY DSECT の構文

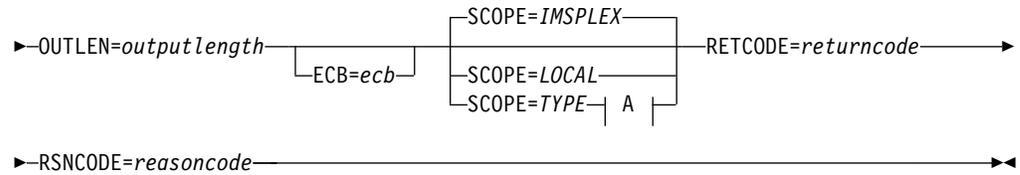
CSLSCQRY 要求の DSECT 機能は、CSLSCQRY パラメーター・リストの長さ、IMSplex タイプ、および CSLSCQRY 戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶—CSLSCQRY—FUNC=DSECT—▶▶

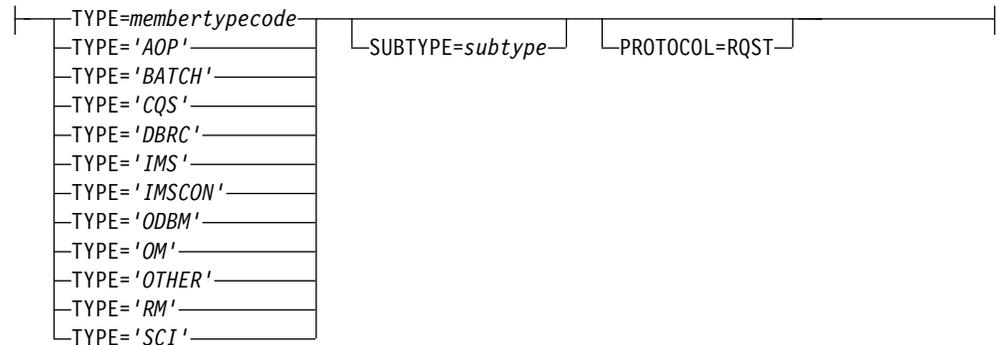
#### CSLSCQRY QUERY の構文

CSLSCQRY サービス要求を発行するための構文は次のとおりです。要求が完了すると、出力は呼び出し元に戻されます。

▶▶—CSLSCQRY—FUNC=QUERY—SCITOKEN=scitoken—PARM=parm—OUTPUT=output—▶▶



**A:**



## CSLSCQRY のパラメーター

**ECB=***symbol*

**ECB=(r1-r12)**

(オプション) - 非同期要求に使用される z/OS ECB のアドレスを指定します。要求が完了したとき、指定された ECB は通知されます。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、マクロの発行元は、CSLSCQRY から制御を受け取った後、このマクロから戻されたデータ (RETCODE フィールドおよび RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を発行する必要があります。

**OUTLEN=***symbol*

**OUTLEN=(r1-r12)**

(必須) -- CSLSCQRY 要求から戻される出力の長さを受け取る 4 バイト・フィールドを指定します。OUTLEN は、OUTPUT= パラメーターで示された出力の長さを受け取ります。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力長はゼロです。

**OUTPUT=***output*

**OUTPUT=(r1-r12)**

(必須) -- CSLSCQRY 要求から戻される可変長出力へのポインターを受け取るフィールドを指定します。出力長は、OUTLEN= フィールドに入れて戻されず。

出力が作成されなかった場合、例えば、出力が作成される前にエラーが検出された場合などは、出力アドレスはゼロです。

CSLSQRYO マクロは、戻される出力をマップします。その出力には、ヘッダーと 1 つ以上のリスト項目が含まれます。

出力バッファは、呼び出し元によって事前割り振りされることはありません。要求からこのワードが戻された場合、そこには、照会出力が入ったバッファのアドレスが入っています。ストレージの使用が終了したとき、呼び出し元は、CSLSCBFR FUNC=RELEASE 要求を出すことにより、このストレージを解放する必要があります。

**PARM=symbol**

**PARM=(r1-r12)**

(必須) -- CSLSCQRY パラメーター・リストを指定します。パラメーター・リストの長さは、SQRY\_PARMLN で定義されたパラメーター・リスト長 EQU 値と同じでなければなりません。

**PROTOCOL=RQST**

(オプション) -- 要求を SCI に送信するための SCI プロトコル。RQST は、この要求には、SCI 要求インターフェース・プロトコルを使用することを表します。

**RETCODE=symbol**

**RETCODE=(r1-r12)**

(必須) -- CSLSCQRY の戻りコードを受け取る 4 バイト・フィールドのアドレスを指定します。SCI 戻りコードは、CSLSRR で定義されています。CSLSCQRY で返される戻りコードについては、以下の表に説明があります。

**RSNCODE=symbol**

**RSNCODE=(r1-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。SCI 理由コードは、CSLSRR で定義されています。CSLSCQRY で返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol**

**SCITOKEN=(r1-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

**SCOPE=IMSPLEX**

**SCOPE=LOCAL**

**SCOPE=TYPE**

(オプション) -- 要求された情報の有効範囲を指定します。

**IMSPLEX**

このオプションは、IMSpIex 内の全メンバーに関するデータを戻します。

**LOCAL**

このオプションは、ローカル z/OS イメージ上の全メンバーに関する情報を戻します。

**TYPE**

このオプションは、指定された IMSpIex メンバー・タイプ (および、オプションでサブタイプ) の全メンバーに関する情報を戻します。

**SUBTYPE=symbol**

**SUBTYPE=(r1-r12)**

(オプション) -- 情報が要求された IMSpIex メンバー・タイプをさらに限定す

る、 8 バイト・サブタイプのアドレスを指定する 4 バイト入力パラメーター。このサブタイプは IMSplex メンバーで定義され、CSLSCREG 要求で指定されます。

このパラメーターは、SCOPE=TYPE の場合のみ、有効です。

TYPE=symbol  
TYPE='AOP'  
TYPE='BATCH'  
TYPE='CQS'  
TYPE='DBRC'  
TYPE='IMS'  
TYPE='IMSCON'  
TYPE='ODBM'  
TYPE='OM'  
TYPE='OTHER'  
TYPE='RM'  
TYPE='SCI'

(オプション) -- 照会の発行対象の IMSplex メンバー・タイプを指定します。SCI は、指定された IMSplex メンバー・タイプ (および、オプションでサブタイプ) の全メンバーに関する情報を戻します。このパラメーターは、SCOPE=TYPE のとき、必須です。

このパラメーターをリテラルとして渡す場合は、そのリテラルを単一引用符で囲む必要があります。このパラメーターをシンボルとして渡す場合、そのシンボルは、ストレージ内の、メンバー・タイプを表すコードが入ったワードを指します。このパラメーターをレジスターとして渡す場合、そのレジスターは、下位ハーフワードにメンバー・タイプ・コードが入っています。

メンバー・タイプのコードは、CSLSTPIX マクロを使って入手することができます。

## CSLSCQRY の戻りコードと理由コード

以下の表は、CSLSCQRY マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。さらに、CSLSCQRY は以下の表にリストされた戻りコードのどれでも返すことができます。

表 64. CSLSCQRY の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'01000008'	X'00002050'	サービスの呼び出し元が無効なパラメーター・リストを渡そうとした。要求はリジェクトされます。
X'0100000C'	X'00003004'	この要求に対してメンバー・データは戻されませんでした。
X'01000014'	X'00005048'	SCI は、要求の出力域用にストレージを取得できませんでした。

関連資料:

253 ページの『第 8 章 CSL SCI クライアントの作成』

273 ページの『CSLSCREG: 登録要求』

## CSLSCQSC: 静止要求

SCI 静止要求は、TYPE で経路指定されたメッセージおよび要求を、発行元の IMSplex メンバーに経路指定するのを停止するように SCI に指示します。この要求が正常に完了すると、その後は、そのメンバーに経路指定されるメッセージおよび要求は、SCITOKEN または NAME で直接に経路指定されたものだけです。

注: IMSplex および z/OS 内でプロセスが非同期であるため、TYPE で経路指定されたメッセージおよび要求は、CSLSCQSC FUNC=QUIESCE 要求が正常に完了した後でも、IMSplex メンバーによって受信される可能性があります。この可能性はわずかですが、起こり得るものです。IMSplex メンバーは、CSLSCQSC FUNC=QUIESCE が正常に完了した後で入ってくるメッセージまたは要求を扱えなければなりません。

### CSLSCQSC の構文

#### CSLSCQSC DSECT の構文

CSLSCQSC 要求の DSECT 機能は、CSLSCQSC パラメーター・リストの長さ、および戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

```
▶▶—CSLSCQSC—FUNC=DSECT—▶▶
```

#### CSLSCQSC QUIESCE の構文

CSLSCQSC FUNC=QUIESCE 要求は、SCI と IMSplex メンバー間の接続を静止させます。この要求が正常に終了した後は、この IMSplex メンバーに対しては、SCITOKEN または NAME で直接に経路指定されたメッセージおよび要求だけが送信されます。

```
▶▶—CSLSCQSC—FUNC=QUIESCE—PARAM=parm—SCITOKEN=scitoken—RETCODE=returncode—▶▶
```

```
▶—RSNCODE=reasoncode—▶▶
```

### CSLSCQSC のパラメーター

**PARAM=symbol**

**PARAM=(r1-r12)**

(必須) -- CSLSCQSC パラメーター・リストを指定します。パラメーター・リストの長さは、SQSC\_PARMLN で定義されたパラメーター・リスト長 EQU 値と同じでなければなりません。

**RETCODE=symbol**

**RETCODE=(r1-r12)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。SCI 戻りコードは、CSLSRR で定義されています。CSLSCQSC で返される戻りコードについては、以下の表に説明があります。

**RSNCODE=symbol**

**RSNCODE=(r1-r12)**

(必須) -- 出力の理由コードを受け取る 4 バイト・フィールドを指定します。

SCI 理由コードは、CSLSRR で定義されています。CSLSCQSC で返される理由コードについては、以下の表に説明があります。

**SCITOKEN=***symbol*

**SCITOKEN=(***r1-r12***)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

## CSLSCQSC の戻りコードと理由コード

以下の表は、CSLSCQSC マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 65. CSLSCQSC の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'01000008'	X'01002038'	パラメーター・リスト・バージョンが無効。
X'01000010'	X'00004000'	SCI がアクティブでない。
	X'00004FFF'	機能がサポートされていない。
X'01000014'	X'00005000'	SCI 内部エラーが発生した。

関連資料:

259 ページの『CSLSCMSG: メッセージ送信要求』

## CSLSCRDY: レディー要求

SCI レディー要求は、TYPE で経路指定されたメッセージおよび要求を IMSplex メンバーが受信できるようにします。CSLSCREG 要求が出された後で、CSLSCRDY が出されるまでは、IMSplex メンバーは、単一のターゲット・アドレス・スペースに直接に経路指定される要求を受信できるだけです。この IMSplex メンバーが送信するメッセージおよび要求は、どの方法で経路指定してもかまいません。

注: IMSplex メンバーは、CSLSCRDY が出されたとき、TYPE で経路指定されているメッセージおよび要求を処理できる準備ができていなければなりません。

IMSplex が非同期であるため、CSLSCRDY を出した後、制御が戻される前に、メンバーは、TYPE で経路指定されたメッセージまたは要求を受信する可能性があります。

### CSLSCRDY の構文

**DSECT** 構文

CSLSCRDY 要求の DSECT 機能は、CSLSCRDY パラメーター・リストの長さ、および戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶—CSLSCRDY—FUNC=DSECT—◀◀

**READY** の構文

CSLSCRDY FUNC=READY 要求は、その IMSplex メンバーは、IMSplex メンバー・タイプで経路指定されるメッセージおよび要求を受信する準備ができたことを SCI に知らせます。

▶—CSLSCRDY—FUNC=READY—SCITOKEN=*scitoken*—PARAM=*parm*—RETCODE=*returncode*—▶

▶—RSNCODE=*reasoncode*—▶

## CSLSCRDY のパラメーター

**PARAM=***symbol*

**PARAM=(***r1-r12***)**

(必須) -- CSLSCRDY パラメーター・リストを指定します。パラメーター・リストの長さは、SRDY\_PARMLN で定義されたパラメーター・リスト長 EQU 値と同じでなければなりません。

**RETCODE=***symbol*

**RETCODE=(***r1-r12***)**

(必須) -- 出力の戻りコードを受け取る 4 バイト・フィールドを指定します。SCI 戻りコードは、CSLSRR で定義されています。CSLSCRDY で返される理由コードについては、以下の表に説明があります。

**RSNCODE=***symbol*

**RSNCODE=(***r1-r12***)**

(必須) -- CSLSCRDY の理由コードを受け取る 4 バイト・フィールドのアドレスを指定します。SCI 理由コードは、CSLSRR で定義されています。CSLSCRDY で返される理由コードについては、以下の表に説明があります。

**SCITOKEN=***symbol*

**SCITOKEN=(***r1-r12***)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

## CSLSCRDY の戻りコードと理由コード

以下の表は、CSLSCRDY マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 66. CSLSCRDY の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
	X'00002038'	パラメーター・リスト・バージョンが無効。
X'01000010'	X'00004000'	SCI がアクティブでない。
	X'00004FFF'	機能がサポートされていない。
X'01000014'	X'00005000'	SCI 内部エラーが発生した。

関連資料:

259 ページの『CSLSCMSG: メッセージ送信要求』

## CSLSCREG: 登録要求

Structured Call Interface (SCI) 登録要求は、IMSplex メンバーと SCI 間の接続を作成するために使用されます。IMSplex 内の通信用に SCI を使用するには、IMSplex メンバーは、SCI に認識されるために、事前に CSLSCREG 要求を発行し、その要求の完了時に SCI トークンを受け取っておく必要があります。

このトークンは、それ以降のすべての SCI 要求で使用されます。IMSplex メンバーがアクティブのときに SCI が終了しても、SCI が再びアクティブになったとき、そのメンバーの登録は残っています。また、そのメンバーが初期の CSLSCREG 要求で受け取った SCI トークンは有効です。

### 制約事項:

- 呼び出し元のアドレス・スペースが SYSEVENT DONTSWAP 呼び出しによってスワップ不能とマークされている場合、CSLSCREG はサポートされません。この環境で CSLSCREG を発行すると、結果は予測不能です。SYSEVENT DONTSWAP を出した呼び出し元は、SCI に登録する前に SYSEVENT OKSWAP を出す必要があります。
- 単一のアドレス・スペースを、SCI に複数回登録することが可能です。ただし、単一のアドレス・スペースからのすべての登録は、最初のアクティブ登録の時と同じ PSW キーおよび状態 (スーパーバイザーまたは問題) で行われる必要があります。

## CSLSCREG の構文

### CSLSCREG DSECT の構文

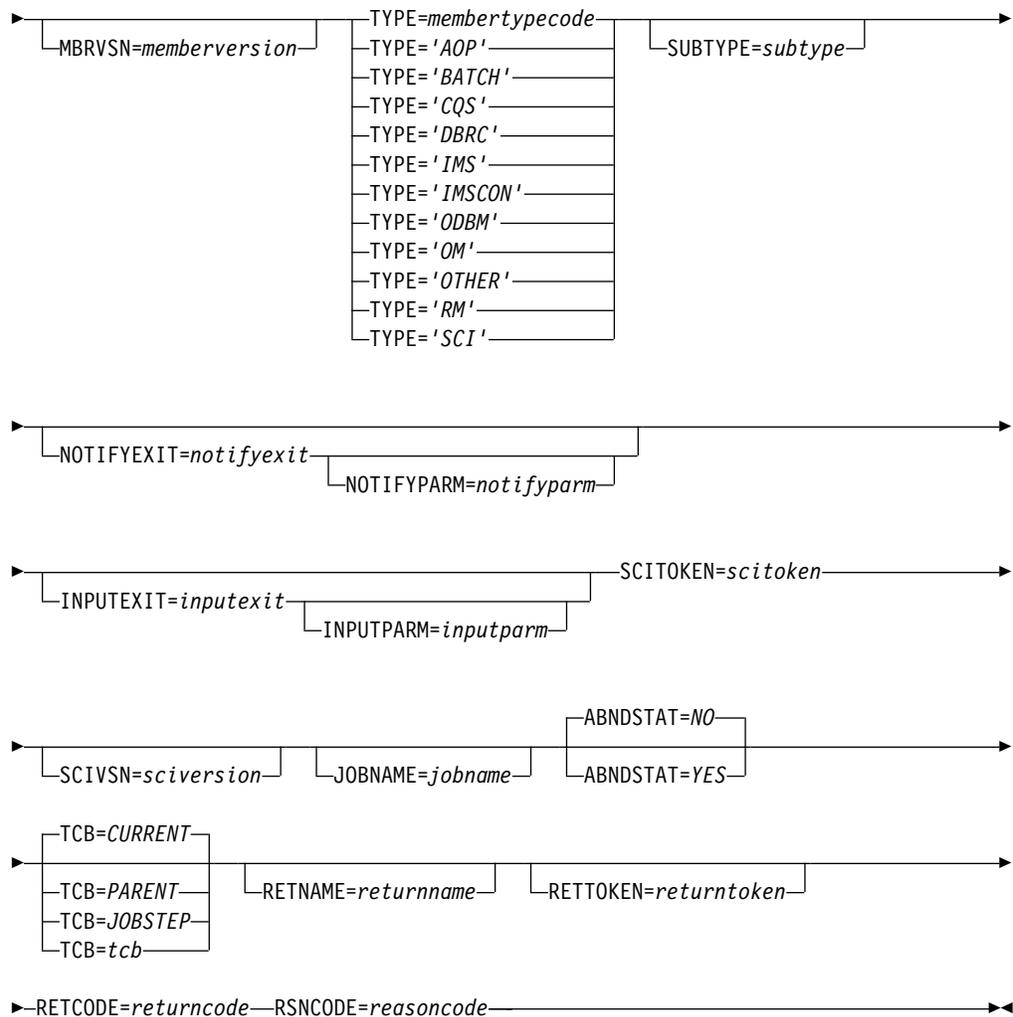
CSLSCREG 要求の DSECT 機能は、CSLSCREG パラメーター・リストの長さ、IMSplex タイプ、および CSLSCREG 戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶—CSLSCREG—FUNC=DSECT—▶▶

### CSLSCREG REGISTER の構文

CSLSCREG FUNC=REGISTER 要求は、IMSplex メンバーと SCI 間を接続します。この要求が正常に完了すると、SCI トークンが戻されます。このトークンを、それ以降のすべての SCI 要求で使用しなければなりません。CSLSCRDY FUNC=READY 要求が発行されるまで、その IMSplex メンバーは、そのメンバーに (SCITOKEN または NAME で) 直接に経路指定されるメッセージおよび要求を受信するだけです。TYPE で経路指定されるメッセージおよび要求は、このメンバーに経路指定されません。CSLSCREG FUNC=READY 要求が正常に完了すると、このメンバーからは、どの方法で経路指定してもメッセージおよび要求を送信することができます。CSLSCREG 要求の REGISTER 機能の構文は次のとおりです。

▶▶—CSLSCREG—FUNC=REGISTER—PARAM=*parm*—IMSPLEX=*imsplex*—MBRNAME=*membername*—▶▶



## CSLSCREG のパラメーター

**ABNDSTAT=NO**

**ABNDSTAT=YES**

(オプション) -- メンバーが異常終了した場合、SCI はメンバーを追跡するかどうかを指定します。ABNDSTAT=YES が指定されると、SCI は、状況 ABTERM と記した項目をそのメンバー用に保存します。メンバーが正常に終了した場合、または正常な CSLSCDRG の後にメンバーが異常終了した場合、SCI に、そのメンバーの記録は残りません。

このパラメーターは、非許可 IMSplex メンバーに対しては無視されます。

**IMSPLEX=symbol**

**IMSPLEX=(r2-r12)**

(必須) -- 1 から 5 文字の IMSplex 名のアドレスを指定します。IMSPlex 名は、この要求の宛先である SCI を識別します。シンボルとして指定された場合、そのシンボルは、IMSPlex 名が入っているストレージを指します。

**INPUTEXIT=symbol**

**INPUTEXIT=(r2-r12)**

(オプション) -- SCI 入力出口ルーチンのアドレスを指定します。入力出口は、このメンバーに対してメッセージまたは要求があるたびに呼び出されます。

**INPUTPARM=symbol****INPUTPARM=(r2-r12)**

(オプション) -- メンバー・データが入っている 8 バイト域のアドレスを指定します。このデータは入力出口ルーチンに、それが呼び出されるたびに渡されます。シンボルとして指定された場合、そのシンボルは、メンバー・データが入っているストレージを指します。

**JOBNAME=symbol****JOBNAME=(r2-r12)**

(オプション) -- SCI ジョブ名を受け取る 8 バイト域のアドレスを指定します。

**MBRNAME=symbol****MBRNAME=(r2-r12)**

(必須) -- SCI に登録している IMSplex メンバーの 8 バイトの名前のアドレスを指定します。許可メンバーの場合、この名前は、IMSplex 内で固有でなければなりません。非許可メンバーの場合、この名前は固有である必要はありません。シンボルとして指定された場合、そのシンボルは、IMSplex メンバー名が入っているストレージを指します。この名前の有効な文字は、A から Z、0 から 9、および特殊文字 @、#、\$ です。

**MBRVSN=symbol****MBRVSN=(r2-r12)**

(オプション) -- SCI に登録している IMSplex メンバーの 4 バイト・バージョンのアドレスを指定します。このバージョン番号は、この IMSplex メンバーがイベントのサブジェクトであるとき、SCI 通知出口のパラメーター・リストに入れて渡されます。このメンバーから発信されるメッセージおよび要求についても、SCI 入力出口のパラメーター・リストに入れて渡されます。MBRVSN が指定されないと、出口パラメーター・リスト内のバージョン番号はゼロに設定されます。シンボルとして指定された場合、そのシンボルは、IMSplex メンバー・バージョンが入っているストレージを指します。

SCI はこのフィールドを検証しません。しかし、このフィールドは、QRY IMSPLEX コマンドで出力できます。その形式は、X'00vvrrmm' となるはずですが。

- 00 -- このバイトは無視されます
- vv -- バージョン番号
- rr -- リリース番号
- mm -- 修正レベルまたはサブリリース番号

例えば、X'00080100' は、8.1.0 と出力されます。

**NOTIFYEXIT=symbol****NOTIFYEXIT=(r2-r12)**

(オプション) -- SCI 通知出口ルーチンのアドレスを指定します。通知出口は、IMSplex メンバーの状況に変更があったときに駆動されます。

**NOTIFYPARM=symbol****NOTIFYPARM=(r2-r12)**

(オプション) -- メンバー・データが入っている 8 バイト域のアドレスを指定し

ます。このデータは通知出口ルーチンに、それが呼び出されるたびに渡されます。シンボルとして指定された場合、そのシンボルは、メンバー・データが入っているストレージを指します。

**PARM=symbol**

**PARM=(r2-r12)**

(必須) -- パラメーターを SCI に渡すために要求が使用するパラメーター・リストのアドレスを指定します。ストレージの長さは、少なくとも SREG\_LN の値と等しくなければなりません。

**RETCODE=symbol**

**RETCODE=(r2-r12)**

(必須) -- CSLSCREG の戻りコードを受け取る 4 バイト・フィールドのアドレスを指定します。SCI 戻りコードは、CSLSRR で定義されています。CSLSCREG で返される戻りコードについては、以下の表に説明があります。

**RETNAME=symbol**

**RETNAME=(r2-r12)**

(オプション) -- 登録要求を処理する SCI の名前を受け取る 8 バイト・フィールドのアドレスを指定します。

**RETTOKEN=symbol**

**RETTOKEN=(r2-r12)**

(オプション) -- 登録要求を処理する SCI の SCI トークンを受け取る 16 バイト・フィールドのアドレスを指定します。

**RSNCODE=symbol**

**RSNCODE=(r2-r12)**

(必須) -- CSLSCREG の理由コードを受け取る 4 バイト・フィールドのアドレスを指定します。SCI 理由コードは、CSLSRR で定義されています。CSLSCREG で返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol**

**SCITOKEN=(r2-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

**SCIVSN=symbol**

**SCIVSN=(r2-r12)**

(オプション) -- SCI バージョン番号を受け取る 4 バイト・フィールドのアドレスを指定します。バージョン番号のフォーマットは、00vvrmm です。

**00** このバイトは将来の利用のために予約済み。現在は常に 00。

**vv** バージョン番号。

**rr** リリース番号。

**mm** 修正レベルまたはサブリリース番号。

例: SCI バージョン 1.1.0 は X'00010100' のように示されます。

**SUBTYPE=symbol**

**SUBTYPE=(r2-r12)**

(オプション) -- SCI に登録しているメンバーの 8 バイトのサブタイプアドレス

スを指定します。サブタイプはユーザーによって定義され、任意の 8 文字を使用できます。シンボルとして指定された場合、そのシンボルは、サブタイプが入っているストレージを指します。指定されない場合、このパラメーターはブランクに設定されます。SUBTYPE が指定されない場合、ブランクに設定されます。

サブタイプは、英数字 (A から Z、0 から 9)、または文字 &、<、および > を除く印刷可能文字 (大/小文字の区別なし) を使用することができます。OM は、クライアントに XML 出力を送信する前に、このフィールドに入っている無効なデータをすべてピリオド (.) に変換します。

**TCB=**CURRENT

**TCB=***JOBSTEP*

**TCB=***PARENT*

**TCB=***symbol*

**TCB=***(r2-r12)*

(オプション) -- 新規 SCI 接続が関連付けられる TCB を指定します。SCI 接続は、以下のいずれかが起こるまで存続します。

- メンバーが CSLSCDRG を使って登録解除する。
- 接続に関連付けられていた TCB が終了する。

CSLSCREG のすべての呼び出し元は、TCB パラメーターに以下の値を指定することができます。

#### **CURRENT**

SCI 接続を、現在実行中の TCB に関連付ける。これはデフォルトです。

#### **JOBSTEP**

SCI 接続を、CSLSCREG 要求が出された TCB の JOBSTEP TCB に関連付ける。この TCB は TCBJSTCB で指定されます。

#### **PARENT**

SCI 接続を、現在実行中の TCB を接続した TCB に関連付ける。

非許可呼び出し元の場合、示された TCB は、それに関連付けられたストレージ・キーが、呼び出し元の現 PSW キーと同じでなければなりません (すなわち、TCBPKF は、現 PSW キーと一致しなければなりません)。

それに加えて、許可呼び出し元は、シンボルまたはレジスターを指定することにより、明示 TCB を識別することができます。シンボルとして指定された場合、そのシンボルは、TCB のアドレスが入っている 1 ワード・ストレージ上のラベルでなければなりません。レジスターとして指定された場合、レジスターには TCB アドレスが入っていなければなりません。

**TYPE=***membertypecode*

**TYPE=**'AOP'

**TYPE=**'BATCH'

**TYPE=**'CQS'

**TYPE=**'DBRC'

**TYPE=**'IMS'

**TYPE=**'IMSCON'

**TYPE=**'ODBM'

**TYPE=**'OM'

**TYPE='OTHER'**

**TYPE='RM'**

**TYPE='SCI'**

(必須) -- SCI に登録中のアドレス・スペースの SCI メンバー・タイプを指定します。

このパラメーターをリテラルとして渡す場合は、そのリテラルを単一引用符で囲む必要があります。このパラメーターをシンボルまたはレジスターとして渡す場合、そのシンボルまたはレジスターは、メンバー・タイプ・コードを表していません。

メンバー・タイプのコードは、CSLSTPIX マクロを使って入手することができます。次のようなメンバー・タイプがあります。

**AOP** この SCI タイプは自動化操作プログラムです。これは、コマンドを送信し、コマンドに対する応答を受信することにより、OM と対話します。

**バッチ**

この SCI タイプは、IMS バッチ領域です。これは、IMS DL/I バッチ領域またはユーティリティ領域として対話します。

**CQS** この SCI タイプは、IMS Common Queue Server です。これは、IMSplex 内の一連の共通キューにアクセスできるようにします。

**DBRC**

この SCI タイプは、IMS データベース・リカバリー管理領域です。

**IMS** この SCI タイプは、IMS 領域です。これには、データベース・マネージャー、トランザクション・マネージャー、および FDBR (IMS データベース・マネージャーに障害が発生したときにデータベース・リソースをリカバリーする IMS 制御領域) が含まれます。SUBTYPE を使用して、特定の制御領域 (例えば、DBDC、DBCTL、DCCTL、または FDBR) をさらに限定することができます。

**IMSCON**

この SCI タイプは、IMS とのコネクターです。これは、IMS と、IMS によって直接にはサポートされないプロトコル (TCP/IP など) との間のインターフェースとして機能します。

**ODBM**

この SCI タイプは IMS Open Database Manager であり、CSL の一部です。これは、クライアントから IMSConnect や ODBA アプリケーションなどの IMS データベース・アクセスおよび操作要求を受け取り、その要求を、データベースを管理する IMSplex 内の IMS DB システムに経路指定します。IMS Connect および IMS Universal ドライバーをサポートするために ODBM が使用されている場合、ODBM は、下位 DRDA プロトコルからの着信データベース・アクセス要求を、出力で IMS が使用する DL/I 呼び出しに再び変換します。

**OM** この SCI タイプは IMS Operations Manager であって、CSL の一部です。これは、AOP からコマンドを受信し、そのコマンドを登録してある IMSplex の他のメンバーに送ります。そして、コマンドへの応答を統合し、出力を発信元の AOP に送り返します。

その他

この SCI タイプは、定義された SCI タイプのいずれにも該当しない、他のアドレス・スペースです。

**RM** この SCI タイプは、IMS Resource Manager であって、これは CSL の一部です。これは、IMSplex 内のリソースを管理し、全 IMSplex プロセスを調整します。SUBTYPE を使用して、IMSplex 内に単一の RM があるのか (SNGLRM)、あるいは IMSplex 内に複数の RM があるのか (MULTRM) をさらに限定します。

**SCI** この SCI タイプは、IMS SCI であって、CSL の一部です。これは、IMSplex 内の通信を管理します。

## CSLSCREG の戻りコードと理由コード

以下の表は、CSLSCREG マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 67. CSLSCREG の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'01000004'	X'00001000'	このメンバーは、すでに登録されている。このメンバーが許可されている場合、メンバー名はすでにこの SCI に登録されており、SCI トークンが戻されます。メンバーが許可されていない場合、現行の TCB から 3 つの登録が存在しており、これ以上の登録は許可されません。SCI トークンのいずれかが戻されます。
X'01000008'	X'00002010'	無効なタイプが渡された。
	X'00002038'	パラメーター・リスト・バージョンが無効。
X'01000010'	X'00004000'	SCI がアクティブでない。
	X'00004004'	CSLSRG00 をロードできなかった。
	X'00004008'	メンバー・アドレス・スペースのユーザー ID は、この SCI への登録が許可されていない。
	X'00004010'	メンバー名 <i>membername</i> は、許可クライアントに対して固有でない。登録はリジェクトされます。
	X'00004028'	非許可メンバーが、許可システム SCI タイプとして登録しようとした。
	X'0000402C'	呼び出し元キーまたは状態が、既存の登録のキーまたは状態と一致しない。
	X'00004FFF'	機能がサポートされていない。
X'01000014'	X'00005000'	SCI 内部エラーが発生した。
	X'00005004'	ESTAE 追加エラーが発生した。
	X'00005008'	BPE SVC エラーが発生した。
	X'0000500C'	z/OS Name/Token 検索エラーが発生した。
	X'00005010'	ResMgr を確立中にエラーが発生した。
	X'00005014'	ストレージを取得中にエラーが発生した。
	X'00005018'	TTOKEN を取得中にエラーが発生した。
	X'0000501C'	ALESERV エラーが発生した。

表 67. CSLSCREG の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'00005020'	ENQ リソース・エラーが発生した。
	X'00005050'	CSLSRGS0 内で BPECGBET エラーが発生した。
	X'00005054'	CSLSRGS0 内で ALESERV エラーが発生した。
	X'00005058'	CSLSRGS0 内で BPEHTADD エラーが発生した。
	X'00005064'	CSLSRGS0 内で BPEHTFND トークン・エラーが発生した。
	X'00005070'	SCI バッファ・マネージャーを初期化できなかった。
	X'00005080'	メンバーの z/OS システム間カップリング・ファシリティー 結合が失敗した。
	X'00005084'	非許可メンバーが明示接続 TCB を指定した。
	X'00005088'	接続 TCB キーが CSLSCREG 呼び出し元のキーと一致しない。
	X'0000508C'	CSLSCREG 要求で渡された TCB タイプ・コードが無効。
	X'00005090'	登録 AWE のエンキュー中にエラー。これは内部エラーです。
	X'00005094'	SRB を SCI にスケジューリング中にエラー。これは内部エラーです。
	X'00005500'	CSLSCREG 処理中に異常終了が起きた。

関連概念:

208 ページの『グローバル・リソースを管理するための CSL RM 要求の発行』

関連タスク:

122 ページの『ODBM クライアントの登録』

関連資料:

131 ページの『CSLQMCMD: コマンド要求』

140 ページの『CSLQMI: API 要求』

152 ページの『CSLQMQRY: 照会要求』

266 ページの『CSLSCQRY: 照会要求』

 CSL SCI 通知クライアント出口ルーチン (出口ルーチン)

284 ページの『CSLSCRQS: 要求送信』

## CSLSCRQR 要求戻し要求

要求をその発行元である IMSplex メンバーに返すには、CSLSCRQR 要求を発行します。返された要求は、サーバーが要求を完了し、要求からの出力を返す準備ができた時点で発行する必要があります。

CSLSCRQR は、要求が発信された IMSplex メンバーに要求を戻します。この要求は、サーバーが要求を完了して、要求からの出力を返す準備ができたときに発行されなければなりません。この要求は、出力を要求側のアドレス・スペースにコピーします。

IMSpIex メンバーは、まず要求を受け取らなければ、このマクロを発行することができないので、要求サーバーだけが CSLSCRQR を発行することができます。要求サーバーは許可されている必要があり、キー 7 を実行していなければなりません。

## CSLSCRQR の構文

### CSLSCRQR DSECT の構文

CSLSCRQR 要求の DSECT 機能は、CSLSCRQR パラメーター・リストの長さ、および戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶ CSLSCRQR—FUNC=DSECT—▶▶

### CSLSCRQR RETURN の構文

CSLSCRQR FUNC=RETURN 要求の構文は次のとおりです。

▶▶ CSLSCRQR—FUNC=RETURN—SCITOKEN=*scitoken*—PARAM=*parm* NOCPYABN=N  
NOCPYABN=Y▶▶

▶▶ RQSTTKN=*requesttoken* RQSTRC=*requestreturncode*▶▶

▶▶ RQSTRSN=*requestreasoncode* RETCODE=*returncode*—RSNCODE=*reasoncode*▶▶

## CSLSCRQR のパラメーター

**PARAM=***symbol*

**PARAM=(***r1-r12***)**

(必須) -- CSLSCRQR パラメーター・リストを指定します。パラメーター・リストの長さは、少なくとも SRQR\_PARMLN の基本パラメーター・リストの長さと同じであることが必要です。ただし、CSLSCRQR に特定のパラメーターをコーディングする場合は、それより長いパラメーター・リストを提供する必要があります。より長いパラメーター・リストを提供する必要があるかどうかを判断するには、次の表を使用します。複数のパラメーターをリストにして使用する場合、しかもそれらのパラメーターの長さの最小値が異なる場合は常に、最も大きな数値となる値を使用してください。

表 68. SRQR\_PARMLN より長いパラメーター・リストを必要とするパラメーター

長さの EQU	この長さを必要とするパラメーター
SRQR_PRMLV2	NOCPYABN

長さが SRQR\_PRMLMX バイトであるパラメーター・リストを定義することもできます。この EQU は、最も長い CSLSCRQR パラメーター・リスト・バージョンの長さに設定され、パラメーター・リストは常に、マクロ・パラメーター

のあらゆる組み合わせに対しても十分な長さになります。ただし、この最大長は IMS リリースの保守のために、または IMS リリース間で、変わる場合があります。

**NOCPYABN=N | Y**

(オプション) - 要求戻りプロセスのコピー・フェーズでの異常終了を、SCI がどのように処理するかを指定します。

**N** SCI はダンプを取得し、ARR または FRR のロジックによって決定されたメッセージを発行します。NOCPYABDN=N がデフォルトです。

**Y** SCI は、要求側ヘデータをコピーして返そうとしたときに異常終了を検出しても、ダンプを取得せず、メッセージも発行しません。SCI は、単に LOGREC データ・セットに項目を書き込むだけです。

このパラメーターは、バージョン 2 のパラメーター・リストの一部として追加されました。このパラメーターを組み込んだ場合、パラメーター・リストは少なくとも SQRQ\_PRMLV2 によって指定された値に等しくなります。

**RQSTRC=symbol**

**RQSTRC=(r1-r12)**

(オプション) -- 戻される要求に関連付けられる戻りコードを指定します。この戻りコードは、この要求を発信した CSLSCRQS の RETCODE パラメーターで示されたストレージに入れて要求元のメンバーに渡されます。このパラメーターが指定されていないと、要求元メンバーに戻りコード・ゼロが渡されます。

シンボルとして指定された場合、そのシンボルは、戻りコードが入っているストレージを指します。

**RQSTRSN=symbol**

**RQSTRSN=(r1-r12)**

(オプション) -- 戻される要求に関連付けられる理由コードを指定します。この理由コードは、この要求を発信した CSLSCRQS の RSNCODE パラメーターで示されたストレージに入れて要求元のメンバーに渡されます。このパラメーターが指定されていないと、要求元メンバーに理由コード・ゼロが渡されます。

シンボルとして指定された場合、そのシンボルは、戻りコードが入っているストレージを指します。

**RQSTTKN=symbol**

**RQSTTKN=(r1-r12)**

(必須) -- 戻される要求に関連付けられる要求トークンを指定します。この要求トークンは、要求が要求処理メンバーに提示されたときに入力出口パラメーター・リスト (INXP\_RQSTTKN) から取得できます。

シンボルとして指定された場合、そのシンボルは、戻りコードが入っているストレージを指します。

**RETCODE=symbol**

**RETCODE=(r1-r12)**

(必須) -- CSLSCRQR の戻りコードを受け取る 4 バイト・フィールドのアドレスを指定します。SCI 戻りコードは、CSLSRR で定義されています。

CSLSCRQR で返される戻りコードについては、以下の表に説明があります。

**RSNCODE=symbol**

**RSNCODE=(r1-r12)**

(必須) -- CSLSCRQR の理由コードを受け取る 4 バイト・フィールドのアドレスを指定します。SCI 理由コードは、CSLSRR で定義されています。CSLSCRQR で返される理由コードについては、以下の表に説明があります。

**SCITOKEN=symbol****SCITOKEN=(r1-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

**CSLSCRQR の戻りコードと理由コード**

以下の表は、CSLSCRQR マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 69. CSLSCRQR の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'01000008'	X'00002004'	SCI インターフェース PC ルーチンに渡された関数が無効。
	X'00002018'	SCI トークンが無効。
	X'00002038'	パラメーター・リスト・バージョンが無効。
X'01000010'	X'00004000'	SCI がアクティブでない。
	X'0000400C'	ターゲット・メンバーがアクティブになっていません。
	X'00004FFF'	機能がサポートされていない。
X'01000014'	X'00005000'	SCI 内部エラーが発生した。
	X'0000502C'	内部 BPE ハッシュ・テーブル・サービス・エラーのためにメンバーが見つからなかった。
	X'00005030'	SCI バッファーが取得できなかった。
	X'00005034'	SCI アドレス・スペース内のキー 7 バッファーが PHDR とパラメーターのコピー用に取得できなかった。
	X'00005038'	IEAMSCHD エラーが発生した。SRB はターゲット・アドレス・スペースに対してスケジュールできなかった。
	X'00005040'	要求は未解決ではなく、戻すことはできない。
	X'00005044'	SCI 割り振り出力バッファーが取得できなかった。
	X'00005500'	SCI 要求の処理中に異常終了が起きた。
	X'00005504'	メンバー・パラメーターをターゲット・アドレス・スペースにコピーしたとき、異常終了が起きた。
	X'00005508'	メンバー・パラメーターがターゲット・アドレス・スペースにコピーされたときに異常終了が発生しましたが、NOCPYABN=Y が指定されていたためにダンプとメッセージは抑止されました。

## CSLSCRQS: 要求送信

IMSpIex メンバーが IMSpIex 内の別のメンバーに要求を送信するには、CSLSCRQS 要求を発行します。ターゲット・メンバーは、SCITOKEN、メンバー名、またはメンバー・タイプによって指定できます。

IMSpIex 内の要求には、入力データと出力データ (ターゲット・メンバーにとっての) の両方を含めることができます。これは、入力データ (ターゲット・メンバーにとっての) しか含めることのできないメッセージとは対照的です。要求のデータは、ターゲット・メンバーのアドレス・スペースにコピーされます。関数は処理され、出力は、要求側のアドレス・スペースに戻されます。要求に ECB が含まれていた場合、SCI が要求を処理し終わったら、制御は要求側モジュールに戻されます。要求者は ECB で待たなければなりません。

要求の処理が完了すると、ECB が通知されます。要求者は RETCODE および RSNCODE フィールドを見て、要求の結果を判別します。要求に ECB が含まれていない場合、要求側モジュールが SCI から制御を戻されたときに、RETCODE および RSNCODE フィールドを使用して要求の結果を判別できます。

注: CSLSCRQS を出す前に、要求者は、SCI 割り振り出力パラメーターのアドレスと長さを受け取るフィールドをクリアしなければなりません。エラーのために要求が宛先に送られない場合、または出力するデータがない場合、SCI は、長さフィールドおよびアドレス・フィールドを更新しません。

### CSLSCRQS の構文

#### DSECT 構文

CSLSCRQS 要求の DSECT 機能は、CSLSCRQS パラメーター・リストの長さ、IMSpIex タイプ、および CSLSCRQS 戻りコードと理由コード用にプログラムの中に等価 (EQU) ステートメントを組み込みます。

▶▶—CSLSCRQS—FUNC=DSECT—▶▶

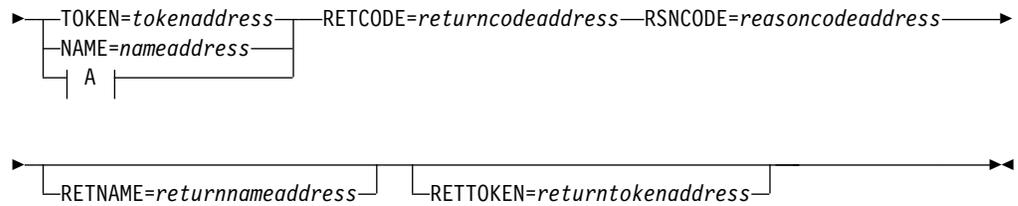
#### SEND REQUEST の構文

CSLSCRQS FUNC=SEND 要求の構文は次のとおりです。

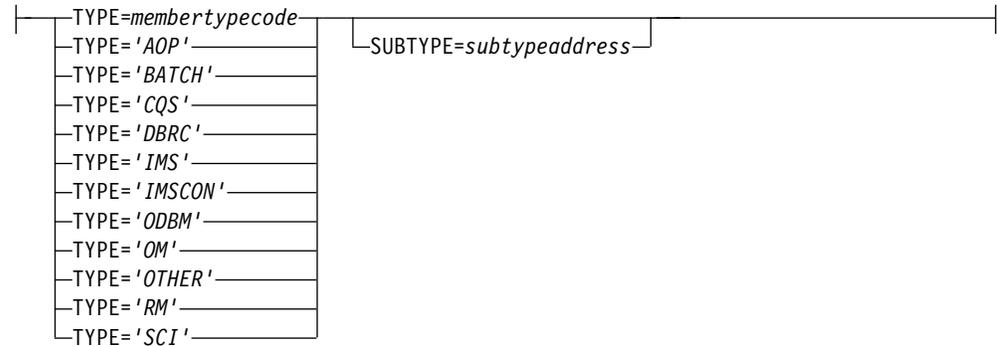
▶▶—CSLSCRQS—FUNC=SEND—SCITOKEN=*scitokenaddress*—PARM=*parmaddress*—▶▶

▶—MBRPARM=*mbrparmlistaddress*—MBRPCNT=*mbrparmcount*—MBRFUNC=*mbrfunctioncode*—▶

▶—MBRSFUNC=*mbrsubfunctioncode*—  
┌—FUNCTYPE=DEST—┐  
└—FUNCTYPE=SENDER—┘  
└—ECB=*ecbaddress*—▶



**A:**



## CSLSCRQS のパラメーター

**ECB=***symbol*

**ECB=(***r1-r12***)**

(オプション) - 非同期要求に使用される z/OS ECB のアドレスを指定します。要求が完了したとき、指定された ECB は通知されます。ECB が指定されていないと、そのタスクは、要求が完了するまで中断されます。ECB が指定された場合、マクロの発行元は、CSLSCRQS から制御を受け取った後、このマクロから戻されたデータ (RETCODE および RSNCODE フィールドを含む) を使用または調査する前に WAIT (または同等のもの) を出す必要があります。

**FUNCTYPE=**DEST

**FUNCTYPE=**SENDER

(オプション) -- この要求の DEST (宛先) または要求の SENDER によって MBRFUNC および MBRSFUNC が定義されたことを指定します。この標識は、SCI 入力出力パラメーター・リストに入れられて要求の受信者に渡されます。

**MBRFUNC=** *symbol*

**MBRFUNC=** (*r1-r12*)

(必須) -- SCI 入力出力パラメーター・リストに入れられて要求の宛先に渡される 4 バイト・メンバー機能コードを指定します。この機能コードは、MBRSFUNC とともに、送信される要求を識別します。

MBRFUNC がシンボルの場合、そのシンボルは、機能コードが入ったストレージの 4 バイト域を指します。

**MBRPARM=** *symbol*

**MBRPARM=** (*r1-r12*)

(必須) -- 事前に作成されたパラメーター・リストのアドレスを指定します。こ

のパラメーター・リストは要求側モジュールで作成する必要があり、いくつかのトリプレットで構成されます。個々のトリプレットは、メンバー・パラメーター・リスト内の 1 つのパラメーターの記述であり、次のもので構成されます。

#### **parameterlength**

メンバー・パラメーターの長さを指定する 4 バイト・パラメーター。

#### **parameteraddress**

メンバー・パラメーターのアドレスを指定する 4 バイト・パラメーター。

#### **datatype**

このパラメーターが SCI でどのように扱われるかを指定する 4 バイト・パラメーター。各タイプに対して等価な値が提供されています (CSLSCODE で組み込まれます)。これらの等価な値を使って、データ・タイプの値を設定することができます。指定できる値は、次のとおりです。

- IN** このパラメーターは入力パラメーターです。これは、要求で宛先アドレス・スペースにコピーされます。
- OUT** このパラメーターは出力パラメーターです。これは、サーバーが要求を完了したとき、要求側アドレス・スペースに戻されてコピーされます。パラメーター用のストレージは、要求が発行される前に割り振っておかなければなりません。
- IO** このパラメーターは入力パラメーターと出力パラメーターの両方です。これは、要求でターゲット・アドレス・スペースにコピーされ、要求が完了したとき、要求側アドレス・スペースに戻されてコピーされます。
- SCI** このパラメーターは SCI 割り振り出力パラメーターです。要求が完了したとき、このパラメーター用のストレージが要求者のアドレス・スペース内に割り振られます。このストレージのアドレスは、パラメーター・アドレス・フィールドに入れて戻され、長さはパラメーター長フィールドに入れて戻されます。ストレージは、要求者が CSLSCBFR 要求を使って解放しなければなりません。SCI 割り振り出力パラメーター用に戻されたアドレスの直前の 8 バイトは、要求者が使用できます。この 8 バイトはクリアされず、バッファの前回使用の残留データが入っている可能性があります。

パラメーター・リストでパラメーターを渡す方法は、アドレスによる方法 と、値による方法 の 2 通りがあります。このどちらの方法も、CSLSCRQS 要求でパラメーターを渡すときに使用できます。長さ、アドレス、タイプのトリプレットを、SCI がパラメーターを正しく扱えるようにセットアップしなければなりません。

- アドレスによる方法

パラメーターをアドレスによって渡すには、パラメーターのアドレスを *parameteraddress* に入れて渡し、パラメーターの長さを *parameterlength* に入れて渡します。SCI は、データ・タイプ IN および IO の場合、*parameteraddress* からパラメーターを取得し、データ・タイプ OUT および

IO の場合、*parameteraddress* にパラメーターを保管します。データ・タイプ SCI の場合、パラメーターが保管されるアドレスとパラメーターの長さが戻されます。

- 値による方法

パラメーターを値によって渡すには、パラメーターを *parameteraddress* に入れて渡し、*parameterlength* にゼロを入れて渡します。データ・タイプ IN では、長さがゼロのとき、SCI は、*parameteraddress* に入っている値を宛先にコピーします。他のデータ・タイプはすべて、「アドレスによる方法」で渡さなければなりません。これは、SCI は、出力パラメーターを保管するためにアドレスを必要とするためです。

メンバー・パラメーター・リスト: ここに指定されるユーザー・パラメーターは、メンバー・パラメーター・リスト内の、要求を受け取るプログラムに提示され、そのアドレスは、入力出口パラメーター域のフィールド INXP\_MBRPLPTR に入っています。各パラメーターは 8 バイトで構成され、先頭 4 バイトは *parameterlength* を示し、2 番目の 4 バイトは *parameteraddress* を示します (*parameteraddress* がアドレスの場合、2 番目の 4 バイトは、ローカル・アドレス・スペース内のストレージを示します。要求側のアドレス・スペースではありません)。パラメーターのデータ・タイプが SCI の場合、先頭の 4 バイトは長さ 4 を示し、2 番目のワードの値は予測不能です。

NULL パラメーター: 場合によっては、要求処理モジュールは、定められた数のパラメーターを、定義された順で期待することがあります。すべてのパラメーターが入っていない要求を送る場合、期待されているすべてがデータ・バッファに入っているように NULL パラメーターを送らなければなりません。

NULL パラメーターは、*parameterlength* および *parameteraddress* にゼロを指定することにより、送ることができます。データ・バッファ内のパラメーターを表す 8 バイトには、ゼロが入ります。これは、すべてのデータ・タイプ (IN、OUT、IO、または SCI) についても、どの方法 (アドレスによる方法、または値による方法) でパラメーターを渡す場合にも、同様です。

**MBRPCNT=symbol**

**MBRPCNT=(r1-r12)**

(必須) -- MBRPARG に含まれるメンバー・パラメーターの数が入っている 4 バイト・フィールドを指定します。

**MBRSFUNC=symbol**

**MBRSFUNC=(r1-r12)=**

(オプション) -- SCI 入力出口パラメーター・リストに入れられて要求の宛先に渡される 4 バイト・メンバー副次機能コードを指定します。この副次機能コードは、MBRFUNC とともに、送信される要求を識別します。

MBRSFUNC がシンボルの場合、そのシンボルは、副次機能コードが入ったストレージの 4 バイト域を指します。

**NAME=symbol**

**NAME=(r1-r12)**

(オプション) -- この要求の宛先の 8 バイト・メンバー名のアドレスを指定します。この名前は、通知出口から入手できます (メンバーが IMSplex に参加している場合)。または、CSLSCQRY 要求を出すことによっても入手できます。

注: 経路指定パラメーターの 1 つ (NAME、TOKEN または TYPE) を指定しなければなりません。

**PARM=***symbol*

**PARM=(r1-r12)**

(必須) -- パラメーターを SCI に渡すために要求が使用するパラメーター・リストのアドレスを指定します。ストレージの長さは、少なくとも SRQS\_LN の値と等しくなければなりません。

**RETCODE=***symbol*

**RETCODE=(r1-r12)**

(必須) -- CSLSCRQS の戻りコードを受け取る 4 バイト・フィールドのアドレスを指定します。SCI 戻りコードは、CSLSRR で定義されています。CSLSCRQS で返される戻りコードについては、以下の表に説明があります。

**RETNAME=***symbol*

**RETNAME=(r1-r12)**

(オプション) -- 要求を処理する SCI の名前を受け取る 8 バイト・フィールドのアドレスを指定します。

**RETTOKEN=***symbol*

**RETTOKEN=(r1-r12)**

(オプション) -- 要求を処理する SCI の SCI トークンを受け取る 16 バイト・フィールドのアドレスを指定します。

**RSNCODE=***symbol*

**RSNCODE=(r1-r12)**

(必須) -- CSLSCRQS の理由コードを受け取る 4 バイト・フィールドのアドレスを指定します。SCI 理由コードは、CSLSRR で定義されています。CSLSCRQS で返される理由コードについては、以下の表に説明があります。

**SCITOKEN=***symbol*

**SCITOKEN=(r1-r12)**

(必須) -- SCI トークンが入っている 16 バイト・フィールドを指定します。このトークンは、SCI との接続を一意的に識別します。SCI トークンは、正常な CSLSCREG FUNC=REGISTER 要求によって戻されました。

**TOKEN=***symbol*

**TOKEN=(r1-r12)**

(オプション) -- この要求の宛先の 16 バイト SCI トークンのアドレスを指定します。このトークンは、通知出口からも入手できるし (メンバーが IMSplex に参加している場合)、SLSCQRY メッセージを出すことによっても入手できます。

注: 経路指定パラメーターの 1 つ (NAME、TOKEN または TYPE) を指定しなければなりません。

**TYPE=***symbol*

**TYPE=(r1-r12)**

**TYPE='AOP'**

**TYPE='BATCH'**

**TYPE='CQS'**

**TYPE='DBRC'**

**TYPE='IMS'**  
**TYPE='IMSCON'**  
**TYPE='ODBM'**  
**TYPE='OM'**  
**TYPE='OTHER'**  
**TYPE='RM'**  
**TYPE='SCI'**

要求の経路指定先の IMSplex メンバーのタイプを指定する入力パラメーター。IMSplex メンバー・タイプによる経路指定は、SUBTYPE パラメーターを使用することにより、さらに限定することができます。TYPE が指定されている場合、SCI は、要求されたタイプの、要求送信先の IMSplex メンバーを選択します。

メンバー・タイプをリテラルとして指定する場合、そのリテラルは単一引用符で囲まなければなりません。このパラメーターをシンボルまたはレジスターとして渡す場合、そのシンボルまたはレジスターは、メンバー・タイプ・コードを表していなければなりません。メンバー・タイプ・コードは、CSLSTPIX マクロを使って入手することができます。

注: 経路指定パラメーターの 1 つ (NAME、TOKEN または TYPE) を指定しなければなりません。

## CSLSCRQS の戻りコードと理由コード

以下の表は、CSLSCRQS マクロ要求で返される戻りコードと理由コードの一覧です。理由コードの意味 (すなわち、推定原因) も示してあります。

表 70. CSLSCRQS の戻りコードと理由コード

戻りコード	理由コード	意味
X'00000000'	X'00000000'	要求が正常に完了した。
X'01000008'	X'00002004'	SCI インターフェース PC ルーチンに渡された関数が無効。
	X'00002008'	渡されたパラメーターの数は、ゼロ以下か、または許容最大値より大きい。
	X'00002010'	無効なタイプが渡された。
	X'00002018'	この SCI トークンが無効。
	X'00002024'	PHDR の長さが無効。
	X'00002028'	経路指定データの長さが無効。
	X'0000202C'	要求のターゲット・メンバーがキー 7 でない。
	X'00002030'	要求のターゲット・メンバーが許可されていない。
	X'00002034'	非許可呼び出し元としては、パラメーターの長さが大きすぎる。
	X'00002038'	パラメーター・リスト・バージョンが無効。
	X'0000203C'	CSLSCRQS が同期要求のために SRB モードで呼び出された (ECB= はコーディングされていない)。
X'01000010'	X'00004000'	SCI がアクティブでない。

表 70. CSLSCRQS の戻りコードと理由コード (続き)

戻りコード	理由コード	意味
	X'0000400C'	宛先メンバーがアクティブでない。宛先メンバーは、名前、トークン、またはタイプにより指定済みの可能性があります。
	X'0000401C'	呼び出し元のメンバーは、SCI からの登録解除処理中です。
	X'00004020'	要求がタイムアウトになりました。
X'01000014'	X'00005000'	SCI 内部エラーが発生した。
	X'00005004'	ESTAE 追加エラーが発生した。
	X'00005024'	SRB ルーチン・エラーが発生した。
	X'00005028'	経路指定タイプが無効。
	X'0000502C'	内部 BPE ハッシュ・テーブル・サービス・エラーのためにメンバーが見つからなかった。
	X'00005030'	宛先メンバーのアドレス・スペース内のバッファークが取得できなかった。
	X'00005034'	SCI アドレス・スペース内のキー 7 バッファークが PHDR とパラメーターのコピー用に取得できなかった。
	X'00005038'	IEAMSCHD エラーが発生した。SRB はターゲット・アドレス・スペースに対してスケジュールできなかった。
	X'0000503C'	MRT が展開できなかった。
	X'00005044'	SCI 割り振り出力バッファークが取得できなかった。
	X'0000504C'	メッセージ SRB キー 7 パラメーター域が取得できなかった。
	X'0000507C'	IXCMMSGO エラーが発生した。
	X'00005500'	CSLSCRQS 処理中に異常終了が起きた。
	X'00005504'	メンバー・パラメーターをターゲット・アドレス・スペースにコピーしていたとき、異常終了が起きた。

関連資料:

273 ページの『CSLSCREG: 登録要求』

---

## 第 9 章 CSL Operations Manager XML 出力

OM API 経由で戻されるコマンド応答は、コード・ページ 037 を使用して XML タグ内に埋め込まれます。XML 出力は、CSLOMI、CSLOMCMD、および CSLOMQRY 要求に対する応答として生成されます。

注: OM の応答はプログラミング・インターフェースとして提供されるのであって、画面に表示される事前作成のメッセージを生成するインターフェースとして提供されるものではありません。OM 要求の場合、出力は、OUTPUT= バッファに入れて戻されます。メッセージの場合、出力は SCI 入力出口に戻されます。OM 応答は、XML タグにカプセル化されて戻されます。

関連概念:

157 ページの『CSL OM 自動化操作プログラム・クライアント』

関連資料:

131 ページの『CSLOMCMD: コマンド要求』

140 ページの『CSLOMI: API 要求』

152 ページの『CSLOMQRY: 照会要求』

194 ページの『CSLOMRSP: コマンド応答要求』

---

### CSLOMI XML 出力の例

以下のコマンド構文例には、CSLOMI XML 出力の例が含まれます。これらの例はそれぞれに異なるシナリオを表し、その例で使用されているコマンドに基づく XML 出力を生成します。

#### CSLOMI XML 出力

それぞれの CSLOMI 要求で、以下の出力例に示されたタグのセットが 1 つ以上返されます。

```
<imsout>
  <ctl>
    <omname> </omname>
    <omvsn> </omvsn>
    <xmlvsn> </xmlvsn>
    <statime> </statime>
    <stotime> </stotime>
    <staseq> </staseq>
    <stoseq> </stoseq>
    <rqsttkn1> </rqsttkn1>
    <rqsttkn2> </rqsttkn2>
    <rc> </rc>
    <rsn> </rsn>
  <rsnmsg> </rsnmsg>
</ctl>
<cmdclients>
  <mbr name="membername">
    <typ> </typ>
    <styp> </styp>
    <vsn> </vsn>
  </mbr>
</cmdclients>
</imsout>
```

```

</mbr>
</cmdclients>
<cmdsyntax> </cmdsyntax>
<cmddtd> </cmddtd>
<cmdtext> </cmdtext>
<cmderr>
  <mbr name="membername">
    <typ> </typ>
    <styp> </styp>
    <rc> </rc>
    <rsn> </rsn>
  </mbr>
</cmderr>
<cmdsecerr>
  <exit>
    <rc> </rc>
    <userdata> </userdata>
  </exit>
  <saf>
    <rc> </rc>
    <racfrc> </racfrc>
    <racfrsn> </racfrsn>
  </saf>
</cmdsecerr>
<cmd>
  <master> </master>
  <userid> </userid>
  <verb> </verb>
  <kwd> </kwd>
  <input> </input>
</cmd>
<cmdrsphdr>
  <hdr ... />
</cmdrsphdr>
<cmdrspdata>
  <rsp> </rsp>
</cmdrspdata>
<msgdata>
  <mbr name="membername">
    <msg> </msg>
  </mbr>
</msgdata>
</imsout>

```

## IMS コマンドの発行例

以下に、CSLOMI XML 出力の例を示します。以下のコマンドの例では、QUERY TRAN コマンドがタイムアウト値 10 秒で IMSA に経路指定されました。

OM API Input:  
 CMD(QUERY TRAN) NAME(SKS\*) ROUTE(IMSA) TIMEOUT(10) RQSTTKN2(QTRANCMD)

OM API Output:  
 <imsout>  
 <ctl>  
 <omname>OM1</omname>  
 <omvsn>1.1.0</omvsn>  
 <xmlvsn>1</xmlvsn>  
 <statime>1999.341 12:52:44.46</statime>  
 <stotime>1999.341 12:52:44.46</stotime>  
 <staseq>B342BCC72A34D206</staseq>  
 <stoseq>B342BCC75CD52208</stoseq>  
 <rqsttkn2>QTRANCMD</rqsttkn2>  
 <rc>0</rc> <rsn>0</rsn>  
 </ctl>  
 <cmd>

```

<master>IMS1</master>
<verb>QRY</verb>
<kwd>TRAN</kwd>
<input>QUERY TRAN</input>
</cmd>
<cmdrsphdr>
<hdr s1b1="TRAN" l1b1="TranCode" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="4" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" key="0" scroll="YES" len="4" dtype="INT"
  align="right" />
</cmdrsphdr>
<cmdrspdata>
<rsp> TRAN(SKS1) MBR(IMSA) CC(0) </rsp>
<rsp> TRAN(SKS2) MBR(IMSA) CC(0) </rsp>
<rsp> TRAN(SKS3) MBR(IMSA) CC(0) </rsp>
<rsp> TRAN(SKS4) MBR(IMSA) CC(0) </rsp>
<rsp> TRAN(SKS5) MBR(IMSA) CC(0) </rsp>
</cmdrspdata>
</imsout>

```

## クライアント・リストの照会例

以下のクライアント・リストの例では、Operations Manager (OM) が、コマンド処理用に現在登録されているクライアント名のリストを返します。

OM API Input:  
 QUERY (CMDCLIENTS) RQSTTKN2 (CLIENTLIST)

OM API Output:

```

<imsout>
<ctl>
<omname>OM1</omname>
<omvsn>1.1.0</omvsn>
<xmlvsn>1</xmlvsn>
<statime>1999.341 12:52:44.46</statime>
<stotime>1999.341 12:52:44.46</stotime>
<staseq>B342BCC72A34D206</staseq>
<stoseq>B342BCC75CD52208</stoseq>
<rqsttkn2>CLIENTLIST</rqsttkn2>
<rc>0</rc> <rsn>0</rsn>
</ctl>
<cmdclients>
<mbr name=IMSA>
<typ>DBDC</typ>
<vsn>0800</vsn>
<jobname>IMSJOB01</jobname>
</mbr>
<mbr name=IMSB>
<typ>DBDC</typ>
<vsn>0800</vsn>
<jobname>IMSJOB02</jobname>
</mbr>
</cmdclients>
</imsout>

```

## コマンド構文の照会例

以下のコマンド構文の例は、現在登録されているコマンドのコマンド構文を返します。この例では、QUERY TRAN コマンドが OM に登録されている唯一のコマンドであり、このコマンドにはキーワード NAME が関連付けられています。

OM API Input:  
 QUERY (CMDSYNTAX) RQSTTKN2 (CMDLIST)  
 OM API Output:

```

<imsout>
  <ctl>
    <omname>OM1</omname>
    <omvsn>1.1.0</omvsn>
    <xmlvsn>1</xmlvsn>
    <statime>1999.341 12:52:44.46</statime>
    <stotime>1999.341 12:52:44.46</stotime>
    <staseq>B342BCC72A34D206</staseq>
    <stoseq>B342BCC75CD52208</stoseq>
    <rqsttkn2>CMDLIST</rqsttkn2>
    <rc>0</rc> <rsn>0</rsn>
  </ctl>

  <cmdsyntax>
    <root>
      <resource name="TRAN">
        <verb name="QUERY">
          <keyword name="NAME">
            <var name="trannname*" />
          </keyword>
        </verb>
      </resource>
    </root>
  </cmdsyntax>

  <cmdtext>
    NEXT "Next"
    BACK "Back"
    FINISH "Finish"
    CANCEL "Cancel"
    SUMMARY "Summary"
    TRAN_NAME "Transaction"
    TRAN_QUERY_NAME "Query"
    TRAN_QUERY_NAME_NAME "Name"
    TRAN_QUERY_NAME_TEXT "Name of transaction."
    TRAN_QUERY_NAME_VAR "trannname*"
  </cmdtext>
</imsout>

```

関連資料:

297 ページの『CSL OM 応答として返される XML タグ』

 [QUERY TRAN コマンド \(コマンド\)](#)

---

## CSL OM CMD 出力

以下のコマンド構文例に、CSL OM CMD XML 出力の例を記載します。この例は、使用されているコマンドに基づいて XML 出力を生成するシナリオを表します。

### CSL OM CMD XML 出力

CSL OM CMD 要求の結果として、以下の出力例に示されたタグが返されます。

```

<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
  <ctl>
    <omname> </omname>
    <omvsn> </omvsn>
    <xmlvsn> </xmlvsn>
    <statime> </statime>
    <stotime> </stotime>
    <staseq> </staseq>
    <stoseq> </stoseq>
  </ctl>

```

```

        <rqsttkn1> </rqsttkn1>
        <rc> </rc>
        <rsn> </rsn>
</ctl>
<cmderr>
  <mbr name="membername">
    <typ> </typ>
    <styp> </styp>
    <rc> </rc>
    <rsn> </rsn>
  </mbr>
</cmderr>
<cmdsecerr>
  <exit>
    <rc> </rc>
    <userdata> </userdata>
  </exit>
  <saf>
    <rc> </rc>
    <racfrc> </racfrc>
    <racfrsn> </racfrsn>
  </saf>
</cmdsecerr>
<cmd>
  <master> </master>
  <userid> </userid>
  <verb> </verb>
  <kwd> </kwd>
  <input> </input>
</cmd>
<cmdrsphdr>
  <hdr ... /hdr>
</cmdrsphdr>
<cmdrspdata>
  <rsp> </rsp>
</cmdrspdata>
<msgdata>
  <mbr name="membername">
    <msg> </msg>
  </mbr>
</msgdata>
</imsout>

```

---

## CSLQMRY 出力

以下のコマンド構文例ごとに、CSLQMRY XML 出力の例を記載します。これらの例はそれぞれに異なるシナリオを表し、その例で使用されているコマンドに基づく XML 出力を生成します。

### CSLQMRY XML 出力

CSLQMRY 要求の結果として、以下の出力例に示されたタグが返されます。

CSLQMRY QUERY TYPE(CMDSYNTAX) 要求の結果として戻されるコマンド構文と変換可能テキストには、タイプ-2 コマンドの情報が含まれます。

```

<imsout>
  <ctl>
    <omname> </omname>
    <omvsr> </omvsr>
    <xmvsr> </xmvsr>
    <statime> </statime>
    <stotime> </stotime>
    <staseq> </staseq>
  </ctl>
</imsout>

```

```

    <stoseq> </stoseq>
    <rqsttkn1> </rqsttkn1>
    <rc> </rc>
    <rsn> </rsn>
</ctl>
<cmdclients>
  <mbr name="membername">
    <typ> </typ>
    <styp> </styp>
    <vsn> </vsn>
    <jobname> </jobname>
  </mbr>
</cmdclients>
<cmdsyntax> </cmdsyntax><cmdddtd>
  <!ELEMENT imsout ( ctl, cmdclients?, cmdsyntax?, cmdddtd?,
  cmdtext?, cmderr?, cmd, cmdrsphdr, cmdrspdata?, msgdata? )>
  <!ELEMENT ctl (omname?, omvsn?, xmlvsn?, statime, stotime,
  staseq, stoseq, rqsttkn1?, rqsttkn 2?, rc, rsn )>
  <!ELEMENT omname (#PCDATA) >
  <!ELEMENT omvsn (#PCDATA) >
  <!ELEMENT xmlvsn (#PCDATA) >
  <!ELEMENT statime (#PCDATA) >
  <!ELEMENT stotime (#PCDATA) >
  <!ELEMENT staseq (#PCDATA) >
  <!ELEMENT stoseq (#PCDATA) >
  <!ELEMENT rqsttkn1 (#PCDATA) >
  <!ELEMENT rqsttkn2 (#PCDATA) >
  <!ELEMENT rc (#PCDATA) >
  <!ELEMENT rsn (#PCDATA) >
  <!ELEMENT cmdclients ( mbr+ )>
  <!ELEMENT cmdsyntax (#PCDATA) >
  <!ELEMENT cmdddtd (#PCDATA) >
  <!ELEMENT cmdtext (#PCDATA) >
  <!ELEMENT cmderr ( mbr* )>
  <!ELEMENT MBR ( (TYP, STYP, ((VSN, JOBNAME) | (rc, rsn))) | msg)>

  <!ELEMENT typ (#PCDATA) >
  <!ELEMENT styp (#PCDATA) >
  <!ELEMENT vsn (#PCDATA) >
  <!ELEMENT jobname (#PCDATA) >
  <!ELEMENT msg (#PCDATA) >
  <!ELEMENT cmdsecerr ( exit, saf )>
  <!ELEMENT exit ( rc, userdata ) >
  <!ELEMENT saf ( rc, racfc, racfrsn )>
  <!ELEMENT userdata (#PCDATA) >
  <!ELEMENT racfc (#PCDATA) >
  <!ELEMENT racfrsn (#PCDATA) >
  <!ELEMENT cmd ( master?, userid?, verb, kwd, input )>
  <!ELEMENT master (#PCDATA) >
  <!ELEMENT userid (#PCDATA) >
  <!ELEMENT verb (#PCDATA) >
  <!ELEMENT kwd (#PCDATA) >
  <!ELEMENT input (#PCDATA) >
  <!ELEMENT cmdrsphdr ( hdr* ) >
  <!ELEMENT hdr (#PCDATA) >
  <!ELEMENT cmdrspdata ( rsp* ) >
  <!ELEMENT rsp (#PCDATA) >
  <!ELEMENT msgdata ( mbr ) >
  <!ATTLIST hdr s1b1 CDATA #REQUIRED >
  <!ATTLIST hdr l1b1 CDATA #REQUIRED >
  <!ATTLIST hdr scope CDATA #REQUIRED >
  <!ATTLIST hdr sort CDATA #REQUIRED >
  <!ATTLIST hdr key CDATA #REQUIRED >
  <!ATTLIST hdr scroll CDATA #REQUIRED >
  <!ATTLIST hdr len CDATA #REQUIRED >
  <!ATTLIST hdr dtype CDATA #REQUIRED >

```

```
<!ATTLIST hdr align CDATA #REQUIRED >
</cmdtd>
<cmdtext> </cmdtext>
</imsout>
```

関連資料:

152 ページの『CSLQMRY: 照会要求』

---

## CSL OM 出力

以下のコマンド構文例に、CSL OM XML 出力の例を記載します。この例は、使用されているコマンドに基づいて XML 出力を生成するシナリオを表します。

### XML タグでカプセル化された非送信請求出力メッセージ

CSL OM 要求の結果として、以下の出力例に示されたタグが返されます。

```
<imsout>
  <ctl>
    <omname></omname>
    <omvsno></omvsno>
    <xm1vsno></xm1vsno>
    <statime></statime>
    <staseq></staseq>
    <uom>UOM</uom>
  </ctl>
  <msgdata>
    <mbr name="membername">
      <typ></typ>
      <styp></styp>
      <msg></msg>
    </mbr>
  </msgdata>
</imsout>
```

---

## CSL OM 応答として返される XML タグ

CSL OM 応答として、さまざまな XML タグが返されます。各タグ名は、文字「<」および「>」によって区切られます。タグを親タグの中にネストして、関連情報をカプセル化することができます。

データまたはその他のタグのセットは、これらの開始タグと終了タグで囲まれています。タグのリストにおいて、インデントは、タグが親タグ内でネストされていることを表します。

**<?xml version "1.0"?>**

この出力で使用される XML のバージョン。

**<!DOCTYPE imsout SYSTEM "imsout.dtd">**

DOCTYPE タグは、文書タイプ定義 (DTD) を含むファイルを識別します。DTD は、このタイプの XML 文書用にサポートされている構造を記述します。z/OS のユーザーは、IMS.SDFSRESL データ・セットにある CSLOMDTD メンバーの中で、DTD 情報を見つけることができます。

**<imsout> </imsout>**

<imsout> </imsout> タグは、OM からの出力をカプセル化します。このタグは、各要求で戻されます。

**<ctl> </ctl>**

**<ctl> </ctl>** タグは、OM が返す制御情報をカプセル化します。これらのタグは、すべての要求で返され、以下の制御情報を含んでいます。

**<omname>om name</omname>**

この要求を処理した OM の名前を示します。名前は、CSLOIxxx PROCLIB メンバーの OMNAME= 実行パラメーター上で指定します。

**<omvsn>om version number</omvsn>**

OM バージョン番号を示します。

**<xmlvsn>xml version number</xmlvsn>**

XML バージョン番号を示します。

**<statime>starttime</statime>**

OM が要求の処理を開始した時刻を示します。このフィールドの形式は、yyyy.ddd hh:mm:ss.th です。

**<stotime>stoptime</stotime>**

OM が要求の処理を完了した時刻を示します。このフィールドの形式は、yyyy.ddd hh:mm:ss.th です。

**<staseq>startsequence</staseq>**

OM が要求の処理を開始したときのシーケンス値を示します。この値は、ソートに使用できます。これは、印刷可能な EBCDIC 16 進形式の値です。

**<stoseq>stopsequence</stoseq>**

OM が要求の処理を停止したときのシーケンス値を示します。この値は、ソートに使用できます。これは、印刷可能な EBCDIC 16 進形式の値です。

**<rqstkn1>requesttoken1</rqstkn1>**

応答に関連付けられた、ユーザー指定の RQSTTKN1 値を示します。OM は、出力の中の印刷不能文字をピリオド (.) に変換します。

**<rqstkn2>requesttoken2</rqstkn2>**

応答に関連付けられた、ユーザー指定の RQSTTKN2 値を示します。OM は、出力の中の印刷不能文字をピリオド (.) に変換します。

**<rc>returncode</rc>**

印刷可能な EBCDIC 16 進形式の、要求に対する戻りコード。

**<rsn>reasoncode</rsn>**

印刷可能な EBCDIC 16 進形式の、要求に対する理由コード。

**<uom>unsolicited output message</uom>**

XML は非送信請求出力出力メッセージ用であることを示します。

**<cmdclients> </cmdclients>**

OM クライアントに関する情報をカプセル化します。このタグは、QUERY(CMDCLIENTS) 要求で戻されます。

**<mbr name="membername"></mbr>**

コマンドに関して登録される IMSplex メンバーの名前を示します。

**<typ>membertype</typ>**

IMSplex メンバーのタイプを示します。

**<styp>membersubtype</styp>**

IMSplex メンバーのサブタイプを示します。OM は、出力の中の印刷不能文字をピリオド (.) に変換します。

**<vsn>memberversion</vsn>**

メンバー・バージョン番号を示します。

**<jobname>memberjobname</jobname>**

メンバー・ジョブ名を示します。

**<cmdddtd> </cmdddtd>**

OM がコマンド構文用および OM 出力 XML 用に定義した文書タイプ定義 (DTD) をカプセル化します。このタグは、QUERY(CMDSYNTAX) 要求で戻されます。

**<cmdsyntax> </cmdsyntax>**

全クライアントから OM に登録されたコマンドに関する XML 定義をカプセル化します。このタグは、QUERY(CMDSYNTAX) 要求で戻されます。

**<cmdtext> </cmdtext>**

XML コマンド構文タグに関連付けられた変換可能なテキスト・ストリングをカプセル化します。このタグは、QUERY(CMDSYNTAX) 要求で戻されます。

**<cmd> </cmd>**

OM に渡されたコマンド情報をカプセル化します。このタグは、コマンド要求で戻されます。このタグで戻される出力は、CSLOMBLD マクロの CMD= パラメーターで指定されたものです。以下のタグは、<cmd> タグの中に入れられます。

**<master> </master>**

コマンドを送信するときにマスターのタグが付けられたコマンド処理クライアントの名前をカプセル化します。この情報は、コマンドが少なくとも 1 つのコマンド処理クライアントに正常に送られなければ、示されません。

**<userid> </userid>**

コマンドの発信元のユーザー ID をカプセル化します。

**<verb> </verb>**

OM が処理したコマンド verb の短縮形をカプセル化します。OM に渡された verb は、長いフォームだったかもしれませんが。

**<kwd> </kwd>**

OM が処理したコマンド・キーワードをカプセル化します。

### <input> </input>

OM に渡された実際の入力コマンド・ストリングをカプセル化します。以下の文字は、出力 XML の妥当性を維持するために、ピリオド (.) に変換されます。

- より大なり記号 (>)
- より小なり記号 (<)
- アンパーサンド (&)
- 印刷不能文字

### <cmdrsphdr> </cmdrsphdr>

コマンド応答で返されたデータ・フィールドに関するコマンド・ヘッダー情報をカプセル化します。このタグは、コマンド要求で戻されます。

#### <hdr ... />

データ・フィールドの列の属性を定義します。

コマンド応答ヘッダー情報の形式は、以下の例に示された形式となります。

```
<hdr s1bl="ss" l1bl="l1l1" scope="c" sort="d" key="e" scroll="f" len="g" dtype="h" align="i" skipb="no"/>
```

**s1bl** データ記述を、<cmdrspdata> タグで返されるデータ値と照合するために使用される短ラベル。

短ラベル値は、コマンドによって異なります。どのコマンドにはどんな値が戻されるかを判別するには、各コマンドの資料を参照してください。

**l1bl** テーブル列ヘッダーとして使用できる長ラベル。

長ラベル値は、コマンドによって異なります。どのコマンドにはどんな値が戻されるかを判別するには、各コマンドの資料を参照してください。

#### 有効範囲

データがグローバルまたはローカルであるかどうかを示します。

**GBL** データがグローバルであることを示します。照会出力の場合、グローバル・データは同じ名前のすべてのリソースに適用されますが、コマンド応答では、特定のリソース名について 1 回返されるだけです。グローバル情報は、異なる IMSplex メンバー名について、同じリソース名の他の行に適用されません。リソース名は、KEY="1" 属性によって識別されるデータ・フィールドです。アプリケーションで、コマンド応答データを、ユーザーに表示するテーブルに変換する場合、グローバル・データ値を、同じリソース名の他の行に伝搬することができます。

**LCL** データがローカルであることを示します。照会出力の場合、ローカル・データは、特定の IMS 内の特定のリソース名のみ適用されます。異なる IMS システムは、ローカル・データ・フィールドに対し

て異なる値を戻す可能性があります。各 IMS は、そのローカル値があれば、それを戻します。アプリケーションで、コマンド応答データを、ユーザーに表示するテーブルに変換する場合、ローカル・データ値を同じリソース名の他の行に伝搬してはなりません。

**sort** このフィールドをソートかどうか、またはソートの方向を示します。

**A** 昇順でソートする。

**D** 降順でソートする。

**N** フィールドをソートしない。

**キー** このフィールドのソート優先順位を示します。

**0** フィールドはソートされない。

**1** 優先順位 1 番目のソート・フィールド。

**2** 優先順位 2 番目のソート・フィールド。

**n** 優先順位 *n* 番目のソート・フィールド。

<cmdrsphdr> タグの KEY= で表されている優先度の値は、事前に決定されています。コマンド応答によっては、複数のソート・フィールドを指定できます。<cmdrsphdr> タグの中に、以下のソート優先順位を持ついくつかのフィールドがリストされています。

- Trancode - 1
- MbrName - 4
- CC - 0
- PSBname - 0
- QCnt - 2
- LCLs - 0
- LQCnt - 3

以下のコード例は、コマンドの結果を以下の優先順位でソートします。

1. Trancode
2. Qcnt
3. LQcnt
4. MbrName

```
<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.1.0</omvsn>
<xm1vsn>1 </xm1vsn>
<statime>2002.261 18:33:56.425140</statime>
<stotime>2002.261 18:33:56.487941</stotime>
<staseq>B8400987409B4A0E</staseq>
<stoseq>B84009874FF05409</stoseq>
<rqsttkn1>USRT002 10113356</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
```

```

</ctl>
<cmd>
<master>IMS2 </master>
<userid>USRT002 </userid>
<verb>QRY </verb>
<kwd>TRAN </kwd>
<input>QRY TRAN NAME(ADD*) SHOW(PSB,QCNT,CLASS) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="TRAN" l1b1="Trancode" scope="LCL" sort="a" key="1"
scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="4"
scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0"
scroll="yes" len="4" dtype="INT" align="right" />
<hdr s1b1="PSB" l1b1="PSBname" scope="LCL" sort="n" key="0"
scroll="yes" len="8" dtype="CHAR" align="left" />
<hdr s1b1="Q" l1b1="QCnt" scope="GBL" sort="d" key="2"
scroll="yes" len="8" dtype="INT" align="right" />
<hdr s1b1="LCLS" l1b1="LC1s" scope="LCL" sort="n" key="0"
scroll="yes" len="3" dtype="INT" align="right" />
<hdr s1b1="LQ" l1b1="LQCnt" scope="LCL" sort="d" key="3"
scroll="yes" len="8" dtype="INT" align="right" /></cmdrsphdr>
<cmdrspdata>
<rsp>TRAN(ADDPART ) MBR(IMS2 ) CC( 0) PSB(DFSSAM04) LCLS( 4)
LQ( 0) </rsp>
<rsp>TRAN(ADDINV ) MBR(IMS2 ) CC( 0) PSB(DFSSAM04) LCLS( 4)
LQ( 0) </rsp>
<rsp>TRAN(ADDPART ) MBR(IMS2 ) CC( 0) Q( 0) </rsp>
<rsp>TRAN(ADDINV ) MBR(IMS2 ) CC( 0) Q( 0) </rsp>
<rsp>TRAN(ADDPART ) MBR(SYS3 ) CC( 0) PSB(DFSSAM04) LCLS( 4)
LQ( 0) </rsp>
<rsp>TRAN(ADDINV ) MBR(SYS3 ) CC( 0) PSB(DFSSAM04) LCLS( 4)
LQ( 3) </rsp>
</cmdrspdata>
</imsout>

```

2つのレコードに同じ Trancode がある場合、これらのレコードは Qcnt でソートされます。これらのレコードに同じ Qcnt がある場合、LQCnt でソートされます。これらのレコードの LQCnt 値が同じ場合、MbrName でソートされます。このソートは、*n* 番目のソート・フィールドが使用されるまで続けられます。

前のコードの XML の結果を、以下のコード・サンプルに示します。

Response for: QRY TRAN NAME(ADD\*) SHOW(PSB,QCNT,CLASS)

Trancode	MbrName	CC	PSBname	QCnt	LC1s	LQCnt
ADDINV	IMS2	0		0		
ADDINV	SYS3	0	DFSSAM04		4	3
ADDINV	IMS2	0	DFSSAM04		4	0
ADDPART	IMS2	0		0		
ADDPART	IMS2	0	DFSSAM04		4	0
ADDPART	SYS3	0	DFSSAM04		4	0

QUERY コマンドの SHOW パラメーターを使用したフィールドの選択によっては、中間優先度の値のフィールドのすべてが表示されるわけではありません。すなわち、ソート結果で、優先度の値が 1 と 4 にセットされたフィールドを表示し、2 と 3 にセットされたフィールドを表示しないよう

にできます。また、プログラムで、レコードをオリジナルの順序のままにしておく、レコードを事前決定された優先度の値を使用してソートする、あるいは、ユーザーによってローカルにセットされた基準を使用するその他のフィールドでソートするようにもできます。

**scroll** TSO SPOC が画面を右にシフトしたときに、このフィールドを画面からスクロールオフするかどうかを示します。

**NO** フィールドをスクロールしない。

**YES** フィールドを画面からスクロールオフすることを許可する。

**len** データの最大長 (戻されるデータは、文字数が少ないかもしれません)。出力応答からデータ・テーブルを作成している場合、この属性に対して表示される列の幅を、この値から決めることができます。このフィールドの値が '\*' の場合、これは可変長フィールドです。

**dtype** 元のデータ・タイプを記述します。すべてのデータが、文字フォーマットで戻されます。ただし、一部のフィールドは数値データを表します。整数として発信されたデータは、数値計算を行うためには、文字から整数へ変換する必要があるかもしれません。

**CHAR**

出力フィールドは文字データを表します。

**INT** 出力フィールドは、整数データの文字表現です。

**align** データを列に定様式で入れる場合、推奨される列位置合わせを示します。

**RIGHT**

データは右寄せです (数値データなど)。

**CENTER**

データは中央そろえです。

**LEFT** データは左寄せです (文字データなど)。

**skipb**

**no** クライアントがこの列に情報を戻さない場合でも、列が TSO SPOC 出力上に表示されます。これはデフォルトです。

**yes** クライアントがこの列に情報を戻さない場合は、列は TSO SPOC 出力上に表示されません。

**<cmdrspdata> </cmdrspdata>**

コマンド応答の詳細情報をカプセル化します。このタグは、コマンド要求で戻されます。タグには、<cmdrsphdr> </cmdrsphdr> タグで示される実際のデータが含まれます。

どのコマンドにはどんな値が戻されるかを判別するには、各コマンドの資料を参照してください。

### **<rsp>response data</rsp>**

特定のリソースに関するコマンド応答出力の論理行が入っています。応答データには、さまざまなタグが *name(value)* の形式で含まれています。名前は、<hdr> タグ内の短ラベル (slbl=) 値にマップされます。これは以下の例で、値 TRAN および PSB で示されています。

```
<cmdrsphdr>
<hdr slbl="TRAN"      llbl="Trancode"... />
<hdr slbl="PSB"       llbl="PSBname" ... />
</cmdrsphdr>
<cmdrspdata>
<rsp>TRAN(A          ) PSB(A11          ) </rsp>
<rsp>TRAN(B          ) PSB(B22          ) </rsp>
<rsp>TRAN(C          ) PSB(C33          ) </rsp>
</cmdrspdata>
```

<hdr> タグには、長ラベル値 (llbl=) もあります。これは列見出しとして使用することができます。これは以下の出力例で、Trancode および PSBname として示されています。

Trancode	PSBname
A	A11
B	B22
C	C33

応答データに含まれた値は、SPOC 出力のデータ列を伝搬します。<hdr> タグの中の残りのタグは、その列の値のフォーマット設定属性を示します。

### **<msgdata> </msgdata>**

事前に作成された IMS メッセージをカプセル化します。メッセージのタイプは任意です。例えば、通知メッセージ、警告メッセージ、またはエラー・メッセージなどです。このタグは、コマンド要求で戻されます。

### **<mb name="membername"></mb>**

メッセージを戻した IMSplex メンバーの名前を示します。

### **<msg>message data</msg>**

リソースに関する論理コマンド応答出力がメッセージ・フォーマットで入っています。メッセージは、メッセージ番号 (例えば、DFSnnnnI) で始まります。LL フィールドも X'15' 改行文字ありません。

### **<cmderr> </cmderr>**

OM またはコマンド処理クライアントが返した戻りコードおよび理由コード情報をカプセル化します。このタグは、コマンド処理クライアントに固有のエラーを戻さなければならないとき、コマンド要求で戻されます。エラーが起きた各 IMSplex メンバーについて、以下の情報が戻されます。

### **<mb name="membername"></mb>**

エラーが検出された IMSplex メンバーの名前を示します。

### **<typ>membertype</typ>**

IMSplex メンバーのタイプを示します。

**<styp>membersubtype</styp>**

IMSpIex メンバーのサブタイプを示します。OM は、出力の中の印刷不能文字をピリオド (.) に変換します。

**<rc>returncode</rc>**

IMSpIex メンバーに関する戻りコードを印刷可能な EBCDIC 16 進形式で示します。

**<rsn>reasoncode</rsn>**

IMSpIex メンバーに関する理由コードを印刷可能な EBCDIC 16 進形式で示します。

**<cmdsecerr> </cmdsecerr>**

OM セキュリティー出口、SAF および RACF (または同等のもの) が返した戻りコードおよび理由コード情報をカプセル化します。OM セキュリティー出口が何らかの理由でコマンドをリジェクトした場合、セキュリティ出口からのユーザー・データもここでカプセル化されます。

**<exit> </exit>**

OM セキュリティー出口からの戻りコードとユーザー・データをカプセル化します。

**<rc>returncode</rc>**

OM セキュリティー出口からの戻りコードを、印刷可能な EBCDIC 16 進形式で示します。

**<userdata>userdata</userdata>**

OM コマンド・セキュリティ・ユーザー出口パラメーター・リスト (CSLOSCX) の OSCX\_USERDATA フィールドに入っている、OM セキュリティー出口から返されたユーザー・データを示します。OM は、出力の中の印刷不能文字をピリオド (.) に変換します。

**<saf> </saf>**

SAF および RACF (または同等のもの) からの戻りコードと理由コードをカプセル化します。

**<rc>returncode</rc>**

SAF からの戻りコードを印刷可能な EBCDIC 16 進形式で示します。

**<racfrc>racfreturncode</racfrc>**

RACF (または同等のセキュリティ製品) からの戻りコードを印刷可能な EBCDIC 16 進形式で示します。

**<racfrsn>racfreasoncode</racfrsn>**

RACF (または同等のセキュリティ製品) からの理由コードを印刷可能な EBCDIC 16 進形式で示します。

関連資料:

291 ページの『CSLOMI XML 出力の例』



---

## 第 10 章 REXX SPOC API および CSL

REXX SPOC API では、REXX プログラムが OM にコマンドをサブミットし、コマンド応答を検索することができます。

---

### CSL OM での REXX SPOC API 環境

REXX SPOC API では、REXX プログラムによって IMSplex 環境をセットアップし、OM にコマンドをサブミットして、コマンド応答を取得することができます。

CSL OM での REXX SPOC API の使用に関連するフェーズには、以下の 3 つがあります。

1. REXX 環境のセットアップ
2. IMSplex 環境のセットアップとコマンドの発行
3. コマンド応答の検索

### CSL 内での REXX 環境のセットアップ

ADDRESS コマンドを発行することにより、REXX 環境をセットアップする CSLULXSB プログラムを呼び出します。このプログラムは、REXX SPOC API 用の REXX サブコマンド環境を確立します。

▶▶—ADDRESS—LINK—'CSLULXSB'—▶▶

注: その他のフォームの ADDRESS コマンドは、Tivoli® NetView® for z/OS 環境では作動しない場合があります。

### IMSplex 環境のセットアップ

デフォルト・ホスト・コマンドを IMSSPOC に切り替え、ADDRESS コマンドを発行することにより、IMSplex 環境をセットアップします。デフォルトのホスト・コマンドを IMSSPOC に設定したら、IMSSPOC は、実行中の REXX プログラムから発行される、それ以降のホスト・コマンドを実行します。

▶▶—ADDRESS—IMSSPOC—▶▶

デフォルトのホスト・コマンドを IMSSPOC に設定したら、IMSSPOC は、実行中の REXX プログラムから発行される、それ以降のホスト・コマンドを実行します。他のホストを指定した ADDRESS コマンドを使用して、他のホストへ切り替えることができます。例えば、次のようにします。

```
ADDRESS TSO
ADDRESS MVS
ADDRESS ISPEXEC
```

その後、それぞれの環境に固有のコマンドを発行することができます。

注: 以下で述べるサブコマンド以外のコマンドを REXX 環境で実行すると、そのコマンドは OM に送られて処理されます。

## IMS サブコマンド

IMS サブコマンドは、IMSpIex の名前を確立します。他のコマンドを出す前に、IMS サブコマンドを出して、IMSpIex 名を確立しなければなりません。指定された名前に自動的に接頭部 "CSL" が付加されます。

▶▶IMS—*IMSpIex\_name*◀◀

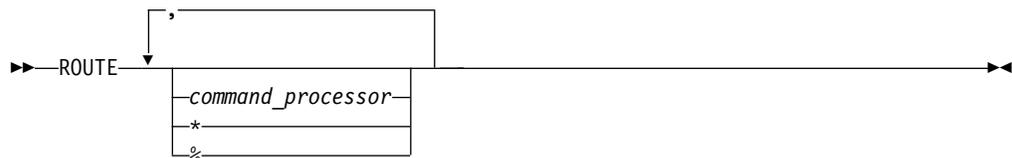
## ROUTE サブコマンド

ROUTE サブコマンドは、コマンド・プロセッサーの名前を設定します。コマンド・プロセッサーは、後続の IMS コマンドを実行する特定のシステムです。コマンド・プロセッサーを指定しないと、次のようになります。

- 直前の経路指定値が除去される。
- コマンドは、IMSpIex の全メンバーに経路指定される。これはデフォルトです。

「\*」を指定すると、コマンドは IMSpIex 内のすべての登録済みコマンド処理クライアントに経路指定されます。「%」を指定すると、そのコマンドに登録されていて MASTER 機能がある、IMSpIex 内の 1 つのコマンド処理クライアントのみに経路指定されます。Operations Manager がそのコマンド処理クライアントを選択します。

ROUTE サブコマンドはオプションです。



## CART サブコマンド

コマンド/応答トークン (CART) サブコマンドは、コマンド/応答トークンの名前を設定します。この 16 文字テキスト・ストリング・トークンは、コマンド応答を検索するために使用されます。

IMS コマンドを出す前に、CART サブコマンドを発行しなければなりません。

▶▶CART—*token\_name*◀◀

## WAIT サブコマンド

WAIT サブコマンドは、OM に対してタイムアウト値を指定するためのものです。タイムアウト値は、形式 MMM:SS または ssss でなければなりません。指定できる最大値は、999:59 です。WAIT サブコマンドはオプションです。

## タイプ-2 IMS コマンドの発行

IMS コマンド (タイプ-2 コマンドを含む) は、REXX プログラム・ストリームの中に引用符付きストリングとして、または引用符付きストリングに解釈される REXX 変数として組み込むことにより、発行してください。

### タイプ-2 コマンドの例

```
"QUERY IMSPLEX SHOW(ALL)"
"DIS ACT"
tranlist = "PETER1,MATT1"
"QUERY TRAN NAME("tranlist")"
```

## CSLULGTS: XML 形式のコマンド応答の取得

CSLULGTS 要求を発行することによって、コマンド応答を取得できます。CSLULGTS コマンドは、REXX がアクセスできるように、コマンド応答をステム変数に入れます。

▶—CSLULGTS(*stem\_name*,*token\_name*,-"*wait\_time*")—▶

### *stem\_name*

CSLULGTS コマンドが正常に完了したら、ステム変数に XML ステートメントが入っています。ステム変数の各行に、XML ステートメントが 1 つずつ入っています。開始 XML タグと終了 XML タグが隣接している場合 (すなわち、その間に他の XML タグが存在していない)、その 2 つのタグはステム変数の同じ行に置かれます。ステム変数の 1 行は、次のようになります。

```
<rsp>TRAN(VIDB ) MBR (IMS2 ) CC( 0) </rsp>
```

### *token\_name*

コマンド/応答トークン (CART) の名前です。これは、CART サブコマンドで指定された名前と一致しなければなりません。

### *wait\_time*

CSLULGTS コマンドのタイムアウト値です。CSLULGTS コマンドは、コマンドが完了するまで待ちますが、待機は、指定された時間だけ持続します。待ち時間は、形式 MMM:SS または ssss です。最大タイムアウト値は 999:59 です。この値は引用符で囲んでください。

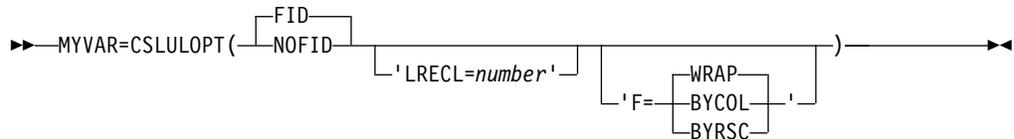
注: このタイムアウト値は、WAIT サブコマンドのタイムアウト値と同じではありません。ただし、この *wait\_time* は、少なくとも、WAIT サブコマンドで指定された値と同じ長さにしてください。そうしないと、実行時間の長いコマンドではコマンド応答が受け取れません。

初めて応答が受信されなかった場合、CSLULGTS を再発行することができます。

## CSLULOPT: フォーマット ID のコマンド応答への組み込み

コマンド応答の中にフォーマット ID を含めるかどうかを指定するには、CSLULOPT 要求を発行します。自動化操作プログラム (AOP) は、フォーマット ID (FID) を使うことによって、コマンド応答の個々の行のレコード・フォーマットを識別します。

CSLULOPT は、IMS オペレーター・コマンドを発行する前に呼び出してください。CSLULOPT で選択される設定は、それが明示的に変更されるまでは、この REXX プログラムに関して有効です。



**F** 出力オプションを指定します。可能なオプションは以下のとおりです。

### BYCOL

行を列によってグループ化します。

### BYRSC

行をリソースによってグループ化します。

### WRAP

個々の行を折り返します (デフォルト)。

### FID

コマンド応答の中に FID が含まれることを指定します。デフォルトは、FID です。

### LRECL

論理レコード長を数値として指定します。

### NOFID

コマンド応答の中に FID が含まれないことを意味します。

### MYVAR

戻りコードを包む、指定できる変数。

関連資料:

 REXX SPOC の戻りコードおよび理由コード (メッセージおよびコード)

## CSLULGTP: REXX ステム変数でのコマンド応答の直接取得

CSLULGTP 要求を発行することで、OM からコマンド応答を検索し、そのコマンド応答を REXX ステム変数に取り込むことができます。それにより REXX プログラムは、CSLULGTS 要求の場合のように XML ステートメントを構文解析するのではなく、ステム変数内の情報を直接参照します。

```
CSLULGTP(—stem_name—, —token_name—, —"wait_time"—)
```

### stem\_name

CSLULGTP 要求が正常に完了すると、REXX ステム変数には、OM から戻さ

れたコマンド応答が取り込まれます。これにより、REXX プログラムはコマンド応答を参照して適切な処置を取ることができます。

#### **token\_name**

コマンド/応答トークン (CART) の名前です。トークンの名前は、CART サブコマンドで指定された名前と一致しなければなりません。

#### **wait\_time**

CSLULGTP コマンドのタイムアウト値です。CSLULGTP コマンドは、コマンドが完了するまで待ちますが、待機は、指定された時間だけ持続します。待ち時間は、形式 MMMMM:SS または ssssss です。この値は引用符で囲んでください。

最大タイムアウト値は 99999:59 です。このパラメーターの値を指定しない場合は、1 秒の 1/10 未満の非常に短時間の遅延の後でこのコマンドはタイムアウトになります。

IMS は、このコマンドが完了したか、または 0.01 秒ごとにタイムアウトになる (**wait\_time** の値が 10 秒未満である場合) かどうかを検査します。この値が 10 秒を超える場合は、IMS の検査は毎秒行われます。

このタイムアウト値は、WAIT サブコマンドのタイムアウト値と同じではありません。ただし、**wait\_time** の値は、少なくとも、WAIT サブコマンドで指定された値と同じ長さにしてください。そうしないと、実行時間の長いコマンドではコマンド応答が受け取れません。

初めて応答が受信されなかった場合、CSLULGTP を再発行することができます。

パラメーター **stem\_name**、**token\_name**、および **wait\_time** が CSLULGTP 呼び出しで指定されていない場合、再現性の低い結果になる可能性があります。

### **CSLULGTP 要求を使用してコマンド応答を検索するためのサンプル・コード**

以下の例に、CSLULGTP 要求のサンプル・コードを記載します。

例 #1: この例では、IMS コマンド QRY TRAN NAME(A) を発行し、CSLULGTP 要求を使用してコマンド応答を検索しています。

```
Address LINK 'CSLULXSB'  
Address IMSSPOC  
"ims PLEX1"  
"wait 5:00"  
cartid = 'PROD12'  
"CART" cartid  
"QRY TRAN NAME(A*)"  
results = cslulgtpl('qinfo.', cartid,"5:00")  
If qinfoctl.rc = 0 Then  
  Do  
    say "OM name      =("qinfoctl.omname")"  
    say "command master =("qinfo.cmd.master")"  
  End
```

例 #2: このプログラムは、IMS UPD PGM コマンドを出し、コマンド応答を処理します。完了コードに基づいて、以下のアクションが実行されます。

```

0 - OK - say command complete for program 'program name'
10 - not found - issue CRE PGM command
73 - PSB scheduled - issue /DIS ACT REG command
other completion code - say invalid CC for program

```

```

/*-----
| Establish IMS rexx environment.
|-----*/
Address LINK 'CSLULXSB'
Address IMSSPOC
"IMS PLEX1"
"WAIT 5:00"
"CART CMDTOKEN"

/*-----
| Issue an IMS command
|-----*/
"UPD PGM NAME(APOL1,BMP255,PGMX,PGMY) SET(SCHDTYPE(SERIAL))"

/*-----
| Get command responses
|-----*/
spoc_rc = CSLULGTP('stem1.','CMDTOKEN','59")

/*-----
| Get some data for diag
|-----*/
Say '*DATA: '
Say '*CMD issued = ' stem1.cmd.input
Say '*SPOC rc = ' spoc_rc
Say '*Command rc = ' stem1.cntl.rc
Say '*Command rsn= ' stem1.cntl.rsn
row = 1
if stem1.cmderr.0 > 0 then do
  Say '*Num of errors = ' stem1.cmderr.0
  Say '*CMD error RC = ' stem1.cmderr.row.rc
  Say '*CMD error RSN = ' stem1.cmderr.row.rsn
end /* if */
if stem1.rsp.0 > 0 then do
  Say '*Num of rsp rows = ' stem1.rsp.0
end /* if */
if stem1.rsp.0 > 0 then do
  /* print each line of the stem.report */
  Do n = 1 to stem1.report.0
  Say stem1.report.n
  End
end /* if */
say ' '

/*-----
| Process responses
|-----*/
spoc_rc = left(spoc_rc,2)
if spoc_rc >< '08' then do /* process response if GTP ran ok */

  /*-----
  | Process the stem response if IMS return code is
  | X'00000000' or X'00000004' or X'0000000C'.
  |-----*/
  row = 1
  imsrc = left(stem1.cmderr.row.rc,2)
  If (imsrc = 00) | (imsrc = 04) | (ims.rc = 0C) then do

    if stem1.RSP.0 > 0 then do
      cccol = 3 /* set col num for CC */
      cctxtcol = 4 /* set col num for CCtext */
    end
  end
end

```

```

do row = 1 to stem1.RSP.0
  stem_cc.row = stem1.RSP.row.cccol
  stem_cctxt.row = stem1.RSP.row.cctxtcol
  say '*Completion code on row 'row ': 'stem_cc.row
  say '*Completion code text on row 'row ': 'stem_cctxt.row
  stem_cc.row = stem1.RSP.row.cccol
  stem_cctxt.row = stem1.RSP.row.cctxtcol
select
  /*-----
  | Process case 1
  -----*/
when left(stem_cc.row,2) = 0 then do
  say 'Command complete for program' stem1.RSP.row.1
  say ' '
end

  /*-----
  | Process case 2.
  | When completion code is X'10', resource not found, issue
  | a CREATE PGM command to create a program like the default
  | descriptor.
  -----*/
when left(stem_cc.row,2) = 10 then do
  say 'Issue CREATE PGM command for' stem1.RSP.row.1
  arg1 = stem1.rsp.row.1 /* extract pgmname to arg1 */
  "CREATE PGM NAME("arg1")" /* issue CRE cmd with arg1 */
  spoc_rc1 = CSLULGTP('stem2.','CMDTOKEN',"59")
  if stem2.rsp.0 > 0 then do
    /* print each line of the stem.report */
    Do n = 1 to stem2.report.0
      Say stem2.report.n
    End
    say ' '
  end /* if */
end /* when */

  /*-----
  | Process case 3
  | When completion code is X'73', program scheduled, issue
  | a DISPLAY ACTIVE REGION command to see the active regions.
  | The region needs to be stopped and the UPD command retried.
  -----*/
when left(stem_cc.row,2) = 73 then do
  say 'Program ' stem1.RSP.row.1 'is scheduled'
  "/DISPLAY ACT REGION"
  spoc_rc2 = CSLULGTP('stem3.','CMDTOKEN',"59")
  /* print the response from each IMS */
  if stem3.MSGDATA.MSG.1.0 > 0 then do
    do x = 1 TO stem3.MSGDATA.NAME.0
      SAY stem3.MSGDATA.NAME.X
      do y = 1 TO stem3.MSGDATA.MSG.X.0
        SAY stem3.MSGDATA.MSG.X.Y
      end/*end do y loop */
    end /* end do x loop */
  end /* end if */
  say 'Issue /STOP REGION commnd and retry UPD cmd'
  say ' '
end /* when */

  /*-----
  | For all other completion codes print the error compcode.
  -----*/
otherwise do
  say 'Invalid CC for program' stem1.RSP.row.1
  say ' '
end /* otherwise */

```

```

        end /* select */
        end /* do loop */

end /* if (imsrc = 00) | . . .*/

/*-----
| Print IMS rc and rsn for all other error rc/rsn.
|-----*/
Else do
    say '*IMS RC & RSN = ' stem1.ctl.rc stem1.ctl.rsn
end /* else */

end /* if spoc_rc >> '08' */

/*-----
| Exit program
|-----*/
"END" /* SPOC */

EXIT /* REXX */

```

上記の例の **say** 命令は、REXX ステム変数のエレメントを参照します。CSLULGTP 要求は、ステム変数の接尾部を設定します。以下の表に、CSLULGTP 要求がステム変数を作成する際に設定される可能性がある接尾部変数名を示します。

表 71. CSLULGTP コマンドによって設定される接尾部変数名

XML タグ	変数名
<?xml version="1.0"?>	stem.xmlversion
<!DOCTYPE imsout SYSTEM "imsout.dtd">	stem.dtd
<imsout>	N/A
<ctl>	N/A
<omname> </omname>	stem.ctl.omname
<omvsn> </omvsn>	stem.ctl.omvsn
<xmlvsn> </xmlvsn>	stem.ctl.xmlvsn
<statime> </statime>	stem.ctl.statime
<stotime> </stotime>	stem.ctl.stotime
<rqsttkn1> </rqsttkn1>	stem.ctl.rqsttkn1
<rc> </rc>	stem.ctl.rc
<rsn> </rsn>	stem.ctl.rsn
<rsnmsg> </rsnmsg>	stem.ctl.rsnmsg
<rsntxt> </rsntxt>	stem.ctl.rsntxt
</ctl>	N/A
<cmderr>	stem.cmderr.0
<mbr name="membername">	stem.cmderr.x.name
<typ> </typ>	stem.cmderr.x.typ
<styp> </styp>	stem.cmderr.x.styp
<rc> </rc>	stem.cmderr.x.rc
<rsn> </rsn>	stem.cmderr.x.rsn
<rsntxt> </rsntxt>	stem.cmderr.x.rsntxt
</mbr>	N/A

表 71. CSLULGTP コマンドによって設定される接尾部変数名 (続き)

XML タグ	変数名
</cmderr>	N/A
<cmdsecerr>	N/A
<exit>	N/A
<rc> </rc>	stem.cmdsecerr.exit.rc
<userdata> </userdata>	stem.cmdsecerr.exit.userdata
</exit>	N/A
<saf>	N/A
<rc> </rc>	stem.cmdsecerr.saf.r
<racfrc> </racfrc>	stem.cmdsecerr.saf.racfrc
<racfrsn> </racfrsn>	stem.cmdsecerr.saf.racfrsn
</saf>	N/A
</cmdsecerr>	N/A
<cmd>	N/A
<master> </master>	stem.cmd.master
<userid> </userid>	stem.cmd.userid
<verb> </verb>	stem.cmd.verb
<kwd> </kwd>	stem.cmd.kwd
<input> </input>	stem.cmd.input
</cmd>	N/A
<cmdrsphdr>	N/A
<hdr></hdr>	stem.hdr.0 (number of columns) stem.hdr.x.s1b1 stem.hdr.x.l1b1 stem.hdr.x.scope stem.hdr.x.sort stem.hdr.x.key stem.hdr.x.scroll stem.hdr.x.len stem.hdr.x.dtype stem.hdr.x.align
</cmdrsphdr>	N/A
<cmdrspdata>	N/A
<rsp> </rsp>	stem.rsp.0 (number of rows) stem.rsp.x.0 (number of cols) stem.rsp.x.y stem.rsp.x.y stem.rsp.x.y stem.rsp.x.y stem.rsp.x.y stem.rsp.x.y stem.rsp.x.y stem.rsp.x.y stem.rsp.x.y
</cmdrspdata>	N/A
<msgdata>	N/A
<mbr name="membername">	stem.msgdata.name.0 (num of systems) stem.msgdata.name.y (1 member name)

表 71. CSLULGTP コマンドによって設定される接尾部変数名 (続き)

XML タグ	変数名
<msg> </msg>	stem.msgdata.msg.y.0 (num of msgs /sys) stem.msgdata.msg.y.x (1 message line)
</mbr>	N/A
</msgdata>	N/A
</imsout>	N/A
N/A	stem.report.0 (number of lines) stem.report.x (1 line of report)

ここで、接尾語変数は以下のとおりです。

**stem** ユーザー定義ステム名  
**x** コマンド応答の行番号  
**y** コマンド応答の列番号

CSLULGTP 関数は、ステム変数の一部として報告書を作成します。ステムは、ユーザーが任意で指定する値です。接尾部は「report」です。

```
"QRY TRAN SHOW(PSB,QCNT)"
results = cslulgtp('friday_status.', cartid,"1:30")
If friday_status.report.0 > 0 Then
Do
say friday_status.report.0
Do x = 1 to friday_status.report.0
say friday_status.report.x
End
End
```

プログラムは以下のような結果を出力します。ステムの各行には、フォーマット済みの報告書の 1 行が含まれます。

```
6
Response for: QRY TRAN SHOW(PSB,QCNT)
Trancode MbrName CC PSBname QCnt LQCnt
ADDINV IMS2 0 0
ADDINV IMS2 0 DFSSAM04 2
ADDINV SYS3 0 DFSSAM04 1
ADDPART IMS2 0 0
```

## CSLULGTP 関数使用時のエラーの処理

CSLULGTP 関数は、関数自体の中でエラーが検出された場合、「ctl.rc」ステム変数および「ctl.rsn」ステム変数を設定しません。このため、CSLULGTP 関数を使用するすべての REXX プログラムでは、関数が正常に完了したかどうかを判別するための他の処理を続行する前に、最初に IMSRC および IMSREASON の REXX 変数を検査することを強くお勧めします。

IMSRC 変数がゼロ以外であり、エラーが CSLULGTP 自体の中で検出された場合、IMSRC 内の値は「08」で始まります。その場合、「ctl.rc」ステム変数および「ctl.rsn」ステム変数は設定されず、ステム変数にデータは返されません。

返される可能性がある IMSRC エラーおよび IMSREASON エラーは、マクロ CSLUXRR で説明されています。

それ以外のエラーの場合 (IMSRC 変数がゼロ以外であり、「08」で始まっていない場合)、「ctl.rc」ステム変数および「ctl.rsn」ステム変数にはコマンド戻りコードと理由コードが含まれ、さらに、コマンド応答に基づいて他のいくつかのステム変数も設定されます。例えば、無効な verb が入力された場合、コマンド応答データは返されませんが、REXX の IMSRC 変数と IMSREASON 変数だけでなく、「ctl.rc」ステム変数および「ctl.rsn」ステム変数が設定されます。

## トランザクション内での REXX SPOC API

トランザクションは REXX 言語で記述できます。REXX EXEC は、IMS アプリケーションとして実行され、他の IMS サポート・プログラミング言語と同様の特性を持ちます。REXX SPOC API を使用して、トランザクションからコマンドを発行してコマンド応答を調べることができます。

IMS アダプター (REXX 版) 環境では、OM API コマンド権限は、MPP ユーザー ID を使用して与えられます。MPP ユーザー ID ではなくトランザクションの起点ユーザー ID を使用するには、ユーザー出口を使用する必要があります。非 OTMA/非 APPC 入力メッセージに対して、セキュリティー環境構築ユーザー出口 (BSEX) が呼び出されます。出口は、メッセージがスケジュールに入れられる際に、MPP 領域にセキュリティー環境を構築するよう IMS に要求することができます (アクセス機能環境エレメント (ACEE) は TCBSENV にアンカーされます)。出口の使用後、トランザクションの起点ユーザー ID は OM API のセキュリティー検査に使用されます。SECURITY=FULL または SECURITY=PROFILE で入力メッセージの選択が FULL でない限り、OTMA/APPC メッセージはサポートされません。

関連概念:

 IMS アダプター (REXX 版) の参照情報 (アプリケーション・プログラミング API)

## IMS SPOC 環境の終了

IMS コマンドを実行する必要がなくなったことから、SPOC 環境が不要になったと伝えるには、END サブコマンドを発行して、IMS SPOC 環境を終了します。

SPOC 環境が必要なくなったことを伝えるには、END サブコマンドを使用します。END サブコマンドが出されると、SPOC 環境に関連付けられていた制御ブロックは解放されます。

END は有効な IMS コマンドです。オペランドなしの END コマンドが指定されると、それは IMS SPOC サブコマンドとして扱われます。END コマンドにパラメーターが指定されていると、その END は IMSplex へ送られて、IMS コマンドとして処理されます。

▶—END—▶

## 非送信請求メッセージの検索

CSLULXSB によって環境をセットアップした後、CSLULSUB、CSLULUSB、および CSLULGUM関数を使用して、非送信請求出力メッセージを取得するために OM にサブスクライブできます。

### CSLULSUB 要求

CSLULSUB 要求を発行することで、サブスクリプションを開始し、OM で非送信請求メッセージをモニターすることができます。

▶▶—CSLULSUB—(—*IMS plex\_name*—,—*member\_list*—,—*type\_list*—)————▶▶

#### CSLULSUB 要求のパラメーター

##### *IMS plex\_name*

IMSplexの名前。「CSL」接頭部は使用しないでください。

##### *member\_list*

IMSplex 内の、この OM クライアントが受け取る非送信請求出力メッセージの発信元であるコマンド処理クライアント (IMS 制御領域など) のメンバー名のリストを、指定します。タイプ・リスト・パラメーターを指定する場合は、このパラメーターは指定しないでください。メンバー・リストもタイプ・リストも指定しない場合は、この OM クライアントは、IMSplex 内のすべてのコマンド処理クライアントにサブスクライブします。リストは引用符で囲んで、名前をコンマで区切ります。

##### *type\_list*

この OM クライアントが受け取る非送信請求出力メッセージの発信元であるコマンド処理クライアント・タイプ (OM、RM、および IMS など) のタイプ・リストを指定します。メンバー・リスト・パラメーターを指定する場合は、このパラメーターは指定しないでください。メンバー・リストまたはタイプ・リストのいずれも指定しない場合は、この OM クライアントは、IMSplex 内のすべてのコマンド処理クライアントにサブスクライブします。リストは引用符で囲んで、名前をコンマで区切ります。

### CSLULUSB 要求

CSLULUSB 要求を発行することで、サブスクリプションを終了できます。サブスクリプションがない場合は、OM は非送信請求メッセージについてモニターされなくなります。

▶▶—CSLULUSB—(—*IMS plex\_name*—)————▶▶

#### CSLULUSB 要求のパラメーター

##### *IMS plex\_name*

IMSplexの名前。「CSL」接頭部は使用しないでください。

## CSLULGUM 要求

CSLULGUM 要求を発行することで、非送信請求出力メッセージを *stem\_name* という名前の REXX プログラム変数にコピーできます。これにより、REXX プログラムはメッセージを調べて適切な処置を取ることができます。

▶▶—CSLULGUM—(—IMS plex\_name—,—stem\_name—)————▶▶

### CSLULGUM 要求のパラメーター

#### *IMS plex\_name*

IMSplexの名前。「CSL」接頭部は使用しないでください。

#### *stem\_name*

受信した非送信請求メッセージを含む REXX 変数の名前。ステム変数は値の配列であり、「.0」エレメント（「mystem.0」など）によって、配列内のエレメント数が示されます。

## OM にサブスクライブする場合のサンプル・プログラム

この REXX SPOC API サンプル・プログラムは、CSLULSUB、CSLULBGUM、および CSLULUSB を使用して OM にサブスクライブし、非送信請求メッセージを取得します。このプログラムに、非送信請求出力メッセージ関数を呼び出す方法が示されています。

```
/* rexx */
/*-----*/
| REXX SPOC API example to invoke unsolicited output message |
| functions. |
|-----*/

/*-----*/
| tuning parameter: check every 10 seconds |
|-----*/
interval = 10

/*-----*/
| We want to make syscalls, that is, sleep |
|-----*/
Call syscalls 'ON'

/*-----*/
| Establish IMS rexx environment |
|-----*/
Address LINK 'CSLULXSB'
If rc = 0 Then
  Do
    Address IMSSPOC
  /*-----*/
  | Subscribe to messages from IMSplex named 'PLEX1' |
  |-----*/
  continu = 1
  Do while(continu)
    subrc = CSLULSUB('PLEX1')
    say 'subrc=('subrc')'
    If subrc = '01000010X' Then
      Do
        Say time()
        Address syscall "sleep" interval
      End
    Else
```

```

        continu = 0
    End

    Do a = 1 To 25
/*-----
| wait a little before checking for new messages
|-----*/
        Address syscall "sleep" interval

/*-----
| Check if any unsolicited messages are present.
|-----*/
        results = CSLULGUM('PLEX1','xml.')
        say 'a='a 'results=('results')

        If xml.0 /= '' Then
            Do
/*-----
| Display any messages in unsolicited message array.
|-----*/
                say 'xml.'0' = ('xml.'0')
                Do idx = 1 To xml.0
                    say 'xml.'idx'=('xml.idx')
                End
            End
        End

    End
End
/*-----
| Unsubscribe to unsolicited messages.
|-----*/
    usbrc = CSLULUSB('PLEX1' )

/*-----
| clean up REXX SPOC API
|-----*/
    "END"
End
Exit

```

---

## REXX のサンプルと例

以降のトピックに、REXX SPOC 環境のサンプル・プログラムと例を記載します。

### サンプル REXX SPOC プログラム

以下のサンプル REXX プログラムは、IMS オペレーター・コマンド /DIS TRAN PART を発行し、コマンド応答を表示します。

#### サンプルの REXX プログラム

```

Address LINK 'CSLULXSB'
Address IMSSPOC
/*-----
| 'ims' defines the IMSplex that receives the commands
|
| 'route' defines which IMSplex members in the IMSplex
| receives the commands.  If ROUTE is not specified or if
| ROUTE * is specified, commands are routed to all IMSplex
| members.
|
| 'wait' provides a timeout value to OM.  The time is in
| mmm:ss format (or ssss if no colon is specified).
|
| 'cart' establishes the command response token for subsequent

```

```

| commands.
|
| 'end' frees control blocks
|-----*/
"IMS IPLX4"
"ROUTE IMS1,IMSB"

"WAIT 5:00"

"CART DISTRAN"
"/DIS TRAN PART"

/*-----*/
| The cslulgts function retrieves data associated with a
| a specific token and fills in a REXX stem variable. In
| this example, it waits 59 seconds.
|
| The XML statements returned are put in the stem variable
| specified by the user.
|-----*/
spoc_rc = cslulgts('DISINFO.', 'DISTRAN', "59")
do z1 = 1 to DISINFO.0
  /* display each line of XML information */
  Say disinfo.z1
end
"END"

```

## REXX SPOC バッチ・ジョブ例

以下に、サンプル・バッチ・ジョブ、サンプル REXX SPOC プログラム、および REXX SPOC の例によるジョブ出力を示します。

### サンプルの REXX SPOC バッチ・ジョブ

以下の図に示したバッチ・ジョブは、文字 V で始まるすべてのトランザクションを含む応答を得るために、バッチ TSO コマンド・プロセッサを呼び出します。

```

//REXXSPOC JOB ,
//  MSGCLASS=H,NOTIFY=USRT002,USER=USRT002,TIME=(,30)
//*
//SPOC      EXEC PGM=IKJEFT01,DYNAMNBR=45
//STEPLIB  DD DISP=SHR,DSN=IMS.SDFSRESL
//SYSPROC  DD DISP=SHR,DSN=LOCAL.IMS.CLIST
//SYSTSPRT DD SYSOUT=A
//SYSTSIN  DD *
           %REXXSPOC QRY TRAN NAME(V*)
/*EOF

```

このバッチ・ジョブ内の DD 名は次のとおりです。

#### STEPLIB

ロード・モジュールが入っています。

#### SYSPROC

REXX プログラムが入っています。

#### SYSTSPRT

REXX プログラムが作成した出力が入っています。

#### SYSTSIN

パラメーターを含めて、実行すべきコマンドが入っています。

JCL 内の QRY TRAN コマンドは、以下のサンプル REXX プログラムへ引数として渡されます。コマンドが発行され、応答は、SYSTSPRT ファイルに送られます。

## REXX SPOC sample program

以下の図は、サンプル REXX プログラム REXXSPOC を示します。

```
/* rexx */
parse upper arg theIMScmd
Address LINK 'CSLULXSB'
if rc = 0 then
do
  Address IMSSPOC
  "IMS plex1" ; if rc > 0 then say 'rc='imsrc 'reason='imsreason
  "route ims2"; if rc > 0 then say 'rc='imsrc 'reason='imsreason
  cartid = "TEST13"
  "cart" cartid ; if rc > 0 then say 'rc='imsrc 'reason='imsreason
  "WAIT 1:00" ; if rc > 0 then say 'rc='imsrc 'reason='imsreason
  theIMScmd
  if rc > 0 then say 'rc='rc 'imsrc='imsrc 'reason='imsreason
  do
    results = cslulgts('TEMP.', cartid,"1:30")
    say 'results='results ' imsrc='imsrc ' reason='imsreason
    if temp.0 /= '' then
      do
        say 'temp.'0'=('temp.0')'
        do idx = 1 to temp.0
          say 'temp.'idx'= 'temp.idx'
        end
      end
    end
  end
end
"END"
End
exit
```

## 出力例

以下の出力例に、REXXSPOC サンプル・プログラムからの出力を示します。

```
READY
%REXXSPOC QRY TRAN NAME(V*)
results=00000000X imsrc=00000000X reason=00000000X
temp.0=(30)
temp.1= <imsout>
temp.2= <ctl>
temp.3= <omname>OM10M </omname>
temp.4= <omvsn>1.1.0</omvsn>
temp.5= <xm1vsn>1 </xm1vsn>
temp.6= <stotime>2001.198 16:08:39.944953</stotime>
temp.7= <stotime>2001.198 16:08:40.271944</stotime>
temp.8= <staseq>B625CACD49AF914A</staseq>
temp.9= <stoseq>B625CACD99848CC6</stoseq>
temp.10= <rqsttkn1>TEST13 </rqsttkn1>
temp.11= <rc>00000000</rc>
temp.12= <rsn>00000000</rsn>
temp.13= </ctl>
temp.14= <cmd>
temp.15= <master>IMS2 </master>
temp.16= <userid>USRT002 </userid>
temp.17= <verb>QRY </verb>
temp.18= <kwd>TRAN </kwd>
temp.19= <input>QRY TRAN NAME(V*)</input>
temp.20= </cmd>
temp.21= <cmdsphdr>
temp.22= <hdr slb1="TRAN" llb1="Trancode" scope="LCL" sort="a"
key="1" scroll="no" len="8" dtype="CHAR" align="left" />
```

```

temp.23= <hdr slb1="MBR" llb1="MbrName" scope="LCL" sort="a"
key="4" scroll="no" len="8" dtype="CHAR" align="left" />
temp.24= <hdr slb1="CC" llb1="CC" scope="LCL" sort="n"
key="0" scroll="yes" len="4" dtype="INT" align="right" />
temp.26= </cmdrsphdr>
temp.26= <cmdrspdata>
temp.27= <rsp>TRAN(VIDB ) MBR(IMS2 ) CC( 0) </rsp>
temp.28= <rsp>TRAN(VIDA ) MBR(IMS2 ) CC( 0) </rsp>
temp.29= </cmdrspdata>
temp.30= </imsout>
READY
END

```

## /DISPLAY コマンドの例およびフォーマット ID

以下の例で、REXX プログラムがオペレーター・コマンド /DISPLAY ACT および /DISPLAY STATUS を発行したときに、フォーマット ID (FID) がどのように返されるのかを示します。

この例は、オペレーター・コマンドでフォーマット ID (FID) がどのようにして返されるのかを示します。この REXX プログラムでは、2 つのコマンドを発行します。/DISPLAY ACT および /DISPLAY STATUS です。

```

ADDRESS LINK 'CSLULXSB'
ADDRESS IMSSPOC
"IMS PLEX1"
"CART DISCARD1"
"DISPLAY ACT"
RSP_RC1= CSLULGTS('DISCARD1','ACT1.')

```

この例では、/DISPLAY ACT を使用します。コマンド応答には FID が含まれません。これは、デフォルトでは、FID が与えられるからです。

次の例では、/DISPLAY STATUS を使用することによって、CSLULOPT 関数が呼び出された後に、コマンドが発行されます。CSLULOPT 関数は、コマンド応答の中に FID を含めないことを指定します。コマンド応答の中に、FID は含まれません。

```

OPT_RC = CSLULOPT('NOFID')
"CART DISCARD2"
"DISPLAY STATUS"
RSP_RC2= CSLULGTS('DISCARD1','STAT1.')

```

## オートノミック・コンピューティングの例

以下の例で、REXX SPOC API に関連するオートノミック・コンピューティング機能について説明します。オートノミック は、コードが反動的であり、誤りと判断したときには、それを訂正するための特定の処置を取ることができることを示します。

### オートノミックの例 1

以下の例では、あるトランザクションを照会します。トランザクションが停止すると、REXX SPOC API はそのトランザクションを始動しようとしています。REXX SPOC API は、問題のトランザクションを参照している行を見つけるために、CSLULGTS から戻された情報を調べます。

```

/* autonomic computing example 1 */
"CART qrytran12"
"qry tran name(CDEBTRN3) show(status)"
results = cslulgts("resp.", "qrytran12", "3:15")

Do idx = 1 to resp.0
  /* find which IMS and the status of tran */
  parse var resp.idx . "TRAN(CDEBTRN3" . ,
                    "MBR(" imsname ")" . ,
                    "LSTT(" status ")" .

  /* if tran is stoppped, try to start it */
  If pos('STOSCHD', status) > 0 Then
    Do
      /* send command to IMS that needs to restart tran */
      "ROUTE" imsname
      "UPD TRAN NAME(CDEBTRN3) START(SCHD)"
    End
  End
End

```

## オートノミックの例 2

以下の例では、QUERY コマンドを使用して、トランザクションのキュー・カウン  
ト (qcnt) を判別します。値が 5 より大きい qcnt は問題と見なされます。REXX  
SPOC API は、この問題を解決するために、別の領域を開始して、当該トランザク  
ションを別のクラスに変更しようとします。

```

/* autonomic computing example 2 */
"CART qrytran13"
"qry tran name(sks1) show(qcnt)"
results = cslulgts("resp.", "qrytran13", "3:15")
Do idx = 1 to resp.0
  parse var resp.idx . "TRAN(SKS1" . "Q(" count ")" .
  If count ^= '' &,
    count > 5 Then
    Do
      "CART strtrgn05"
      "START REGION IMSRG5"
      start? = cslulgts("strt.", "strtrgn05", "10:00")
      if imsrc = '00000000X' then
        Do
          "CART updtran14"
          "update tran name(SKS1) set(class(5))"
        End
      End
    End
  End
End
"END"

```

---

## 第 3 部 非同期データ伝搬

データ・キャプチャー出口ルーチンを使用する代わりに、取り込みデータを非同期で伝搬することができます。このオプションを使用すると、選択したセグメント・タイプおよび環境情報に対するデータベース変更内容を、IMS データ・キャプチャー・ログ・レコードに取り込むことができます。

この代替手段を使用するには、DBDGEN ユーティリティの EXIT= パラメーターにロギング・オプションを追加します。このロギング・オプションを使用して、選択したセグメント・タイプおよび環境情報に対するデータベース変更内容を、IMS データ・キャプチャー・ログ・レコードに取り込むことができます。取り込まれた情報は、IMS オンライン・データ・セット (OLDS) への取り込みデータの格納に必要なスペースを最小限にするために、z/OS 圧縮サービスを使用して圧縮されます。いったん格納された取り込みデータは、例えば Db2 for z/OS 環境へ伝搬するために使用できます。

データ・キャプチャー・ログ・レコードは、X'99' ログ・コードを使用し、下記のデータ・キャプチャー・サブコードによってログに記録されているレコードのタイプを示します。

X'04' 変更データ

X'28' ジョブ終了 (EOJ)

X'30' SETS 呼び出し

X'34' ROLS 呼び出し

すべてのデータ・キャプチャー・ログ・レコードには、共通接頭部として、データ・キャプチャー・ログ接頭部が含まれます。この接頭部には、ログに記録されているレコードのタイプを定義するサブコードが含まれます。ジョブ終了レコードと、SETS および ROLS 呼び出しレコードは、このログ・レコード接頭部だけで構成され、その他の情報を含んでいません。しかし、変更データ・ログ・レコードには、接頭部のほかに情報も含まれており、すべての取り込みデータを含めるために複数の物理ログ・レコードにまたがる場合があります。

関連概念:

 [IMS Configuration Manager for z/OS V2.2](#)

 [データ・キャプチャー出口ルーチン \(データベース管理\)](#)

関連資料:

 [DBD ステートメント \(システム・ユーティリティ\)](#)



---

## 第 11 章 変更データ・ログ・レコード

特定の呼び出しのためにログに記録されたデータをすべて取り出すには、物理ログ・レコードを調べて、論理ログ・レコードの始まりと終わりを判別しなければなりません。変更データ・ログ・レコードを調べることで、DL/I 呼び出し用に取り込まれたデータを確認できます。

各 DL/I 呼び出しごとに、その呼び出しのために要求されたすべてのデータから成る論理ログ・レコードが 1 つずつあります。ただし、すべてのデータを 1 つのログ・バッファに収容できないために、1 つの論理ログ・レコードが実際には複数の物理ログ・レコードで構成される場合もあります。特定の呼び出しのためにログに記録されたデータをすべて取り出すには、物理ログ・レコードを調べて、論理ログ・レコードの始まりと終わりを判別しなければなりません。

DL/I 呼び出し用に取り込まれたデータは、変更データ・ログ・レコード内のエレメントとして保管されます。それぞれの DL/I 呼び出しごとに、複数のデータ・エレメントが変更データ・ログ・レコードに記録されます。データ・エレメントの詳細は、以降のトピックで説明します。

---

### 取り込みデータのエレメント

各データ・エレメントに含まれる 4 バイトのヘッダーには、ヘッダーに続くデータのタイプおよび長さを記述するメタデータがあります。

取り込みデータ・エレメントは、変更データ・ログ・レコード内の DCAP\_DATA フィールドに記録されます。データ・エレメントのフォーマットは、ヘッダー情報と実際のログ・データとに分類されます。ログ・レコード・ヘッダーのフォーマットについては、337 ページの『データ・エレメント・ヘッダーのフォーマット』を参照してください。

各データ・エレメントには 4 バイトのヘッダーが含まれています。ヘッダーの内容は、このエレメントに記録されるデータのタイプを記述する 1 バイトの LOGID フィールド、1 バイトの LOG\_FLAG フィールド、および 2 バイトの LOG\_LL フィールドです。LOG\_LL フィールドには、ヘッダーの後に続くデータの長さが含まれます。この長さは、次のエレメントのヘッダーを見つけるために使用されます。ヘッダーの後には、LOGID フィールドに定義されている実データが続きます。このデータは、z/OS/ESA 圧縮サービスの使用が可能であれば、それを使用して圧縮されます。圧縮された場合は、LOG\_DCAP\_DATA 内の COMPR\_ALGORITHM フィールドが、データの圧縮に使用されたアルゴリズムを示します。

変更データ・ログ・レコードの FLAG\_1 に X'60' または X'E0' が含まれている (FIRST\_RECORD フラグおよび FIRST\_CALL フラグがセットされている) 場合、次のデータ・エレメントは LOG\_INQY\_DATA となります。このエレメントは、論理ログ・レコードの最初の物理レコードにのみ含まれます。

ヘッダーの LOGID フィールドで識別されるデータ・エレメントのタイプは、下記のいずれかの値となります。

#### LOGID X'00'

それぞれの EXIT= ロギング要求ごとに CAPD ブロックが 1 つあります。

#### LOGID X'04'

データベースの DBD のバージョン (DBDGEN の DBD マクロの VERSION= パラメーターで指定されているもの)。63 文字までの文字ストリングです。

#### LOGID X'08'

CAPD 中のセグメントへの連結キー。物理連結キーのフォーマットは、3,824 (255 x 15) 文字までの文字ストリングです。

#### LOGID X'0C'

取り込みデータ・セグメント・データ・ブロック (CAPD\_DATA)。このブロックは、取り込みセグメント・データ (パス・データ、セグメント・データ、または変更前イメージ・データのいずれか) 用に使用されます。呼び出しに取り込まれたそれぞれのセグメントごとに、CAPD\_DATA ブロックがあります。ブロックは、パス・データ、セグメント・データ、変更前イメージ・データ (置き換え)、ツイン・データ (挿入) の順にログに記録されます。

#### LOGID X'10'

パスに関するセグメント・データか、削除、挿入、または置換されたセグメント・データ (置き換えの場合、このデータは変更後イメージを表します)。セグメント・データは、長さが 30,700 文字までの文字ストリングです。

#### LOGID X'14'

置き換え操作で変更されたセグメント・データの変更前イメージ。最初に変更されたバイトから最後に変更されたバイトまでしかログに記録されません。セグメントの未変更の始めと終わりの部分は、セグメント・データに含まれている変更後イメージから再構成されます。変更前イメージ・データは、ハーフワードのオフセット値 (セグメント内の変更データのオフセット) と、それに続く 30,700 文字までの文字ストリングです。実際の変更前イメージだけが圧縮されます。ハーフワードのオフセット値は圧縮されません。変更データ・ログ・レコード内には、各変更セグメントごとに変更前イメージが 1 つずつ含まれることがあります。

データ・エレメントの全部が単一のログ・レコードに収容されない場合は、次のログ・レコード (同じ PST\_NUMBER をもつ) に残りのデータ・エレメントが (前のレコードからはみ出した場所から) 入れられます。しかし、変更前イメージ・データは、セグメント内のデータのオフセット (2 バイト) が変更データの前に記録されるという点で独特です。したがって、DCAP\_DATA (LOG\_LL) 内のデータの長さは、ログに記録されたデータの実際の長さより 2 バイト長くなります。オフセット・フィールドは圧縮されません。

#### 関連資料:

341 ページの『CAPD\_DATA のフォーマット (LOGID=X'0C')』

338 ページの『CAPD ブロックのフォーマット (LOGID=X'00')』

---

## 取り込みデータ量の削減

変更データ・ログ・レコードの内部域がすべてのデータを入れるのに十分な大きさでない場合は、ログ・レコードに `ERROR_LOG` フラグが設定され、完全なデータではないことが示されます。カスケードが必要な場合には、`DBDGEN` の `EXIT=` パラメーターに `CASCADE,KEY,NODATA,NOPATH` を指定してください。

変更データ・ログ・レコードには、カスケード削除用の大量のデータを入れることがあります (特に、パス・データが要求された場合)。データは、ロギングされる前に、内部域に段階的に書き込まれます。この内部域がすべてのデータを入れるのに十分な大きさでない場合は、ログ・レコードに `ERROR_LOG` フラグがセットされて、データが完了していないことを示します。ただし、その場合でも、IMS データはコミットされ、呼び出しは正常に完了します。

ログ・レコードへの大量データの書き込みと未完成ログ・レコードを回避するために、カスケードが必要な場合は、`DBDGEN` の `EXIT=` パラメーターに `CASCADE,KEY,NODATA,NOPATH` を指定してください。これを指定すると、`DLET` 操作では連結キーだけしかログに書き込まれないので、カスケード削除の際に取り込まれるデータの量が大幅に削減されます。1 次キーを連結キーから取り出した場合の `Db2 for z/OS` へのデータ伝搬では、`Db2 for z/OS` での削除にセグメント・データは不要です。

---

## ログに記録されたデータ・エレメントの例

変更データ・ログ・レコードには、呼び出しの結果としての多数のデータ・エレメントが含まれています。このトピックでは、第 3 レベル・セグメントが置き換えられ、パス・データをログに記録するように要求された場合について説明します。

変更データ・ログ・レコードには、呼び出しの結果としての多数のデータ・エレメントが含まれています。第 3 レベル・セグメントが置き換えられ、パス・データをログに記録するよう要求された場合を考えてみます (ルート・セグメントは A、第 2 レベル・セグメントは B、そして置き換えられたセグメントは C です)。この状況では、データ・エレメントは以下の順序でログに記録されます。

1. CAPD
2. DBD バージョン ID
3. 物理連結キー
4. A に関する CAPD\_DATA ブロック
5. A に関するセグメント・データ
6. B に関する CAPD\_DATA ブロック
7. B に関するセグメント・データ
8. C に関する CAPD\_DATA ブロック
9. C に関するセグメント・データ
10. C に関する CAPD\_DATA ブロック (変更前データ)
11. C に関する変更前イメージ・データ

パス・データが要求されていなかった場合は、A および B に関する `CAPD_DATA` ブロックとセグメント・データ・エレメントはログに記録されません。



---

## 第 12 章 ジョブ終了 (EOJ) 呼び出しログ・レコード

EOJ 呼び出しログ・レコード (X'28' サブコード) は、変更データ・ログ・レコードを書き込んだバッチ DL/I プログラムが正常に終了したときに書き込まれます。バッチ・ジョブが終了したときはコミット・レコードがログに書き込まれないので、更新がコミットされたことを示すためにこのレコードが書き込まれます。

関連資料:

342 ページの『ジョブ終了呼び出しログ・レコードのフォーマット』



---

## 第 13 章 SETS および ROLS 呼び出しログ・レコード

データの取り込みを発生させる可能性のあるアプリケーションがトークンを使用して SETS または ROLS 呼び出しを発行するたびに、SETS (X'30' サブコード) および ROLS (X'34' サブコード) 呼び出しログ・レコードが書き込まれます。

ROLS 呼び出しで使用されるトークンに対する SETS 呼び出しの後に書き込まれた変更データ・ログ・レコードは、打ち切られて (バックアウトされて) しまうことを示すために、このログ・レコードが書き込まれます。このレコードは、出口がロギング要求を持たずに定義されている場合でも書き込まれます。

関連資料:

342 ページの『SETS および ROLS 呼び出しログ・レコードのフォーマット』



## 第 14 章 データ・キャプチャー・ログ・レコードのフォーマット

このセクションの以降のトピックでは、データ・キャプチャー・ログ・レコードのフォーマットについて説明します。

### データ・キャプチャー・ログ・レコードの接頭部

データ・キャプチャー・ログ・レコードの接頭部には、ログ・レコードの長さ、データ・キャプチャー・サブコード、リカバリー・トークン、および CPU ストア・クロックのタイム・スタンプを記述するフィールドがあります。

以下の表は、データ・キャプチャー・ログ・レコードの接頭部の一覧です。

表 72. データ・キャプチャー・ログ・レコードの接頭部

フィールド名	データ・タイプ	フィールドの説明
LL	H	ログ・レコードの長さ。IMS によってレコードの最後に追加された 8 バイトのログ・シーケンス番号を含む。
ZZ	H	常にゼロ
LOGCODE	XL1	X'99' ログ・レコード・コード。
SUBCODE	XL1	データ・キャプチャー・サブコード。
PST_NUMBER	H	PST 番号。
RECOVTKN	XL16	リカバリー単位に対するリカバリー・トークン。これは、このリカバリー単位に関するコミット・ログ・レコードまたは打ち切りログ・レコードを関連付けるために使用される。
STORE_CLOCK	XL8	呼び出しが完了し、このログ・レコードが書き込まれたときの CPU ストア・クロック・タイム・スタンプ。

### 変更データ・ログ・レコードのフォーマット

変更データ・ログ・レコードには、呼び出しに関して書き込まれた最初と最後のログ・レコード、呼び出しのユーザー ID、ロギング対象のデータ・エレメント、およびログ・レコードに関連するその他の情報を記述するフィールドがあります。

以下の表は、変更データ・ログ・レコードのレコード・フォーマットの一覧です。

表 73. 変更データ・ログ・レコードのフォーマット

フィールド名	データ・タイプ	フィールドの説明
FLAG_1	XL1	フラグ 1。ビット定義は次のとおり。
LAST_RECORD	X'80'	この呼び出しで書き込まれた最後のログ・レコード。このビットがオンでない場合は、この DL/I 呼び出しでの残りのデータが後続のログ・レコードに入っています。
FIRST_RECORD	X'40'	この呼び出しで書き込まれた最初のログ・レコード。このビットがオンでない場合は、LOG_DATA_OFFSET に記録されているデータは、前のレコードに収まらないデータです。

表 73. 変更データ・ログ・レコードのフォーマット (続き)

フィールド名	データ・タイプ	フィールドの説明
FIRST_CALL	X'20'	このリカバリ単位に関して書き込まれた最初のログ・レコード。したがって、このログ・レコードには LOG_INQY_DATA が含まれています。(FIRST_RECORD がオンになっています。)
	X'10'	将来の利用のために予約済み。
	X'08'	将来の利用のために予約済み。
ERROR_LOG	X'04'	この呼び出しのログ・レコードの処理中にエラーがあったために、このログ・レコードは完成していない。呼び出しに関するデータが記録されていない可能性があります。
DBLEWORD_SEQNUM	X'02'	ログ・シーケンス番号がダブルワード。
STCK_AT_END	X'01'	ログ・レコードの終わりでのストア・クロック。
FLAG_2	XL1	フラグ 2。ビット定義は次のとおり。
V11_9904_FORMAT	X'80'	このログ・レコードの CAPD_DATA ブロックには SEGMENT_RBA_64 および SEGMENT_RBA_64H が入っています。
V11_9904_PARTNM	X'40'	X'9904' ログ・レコードの区画名。
V13_9904_DBVER	X'20'	データベース・バージョン番号は、この X'9904' ログ・レコードに記録されます。
V13_9904_POSDATA	X'10'	位置決めデータは、この X'9904' ログ・レコードに取り込まれます。
LOG_DATA_OFFSET	H	DCAP_DATA エlementが始まる、ログ・レコード内のオフセット。FIRST_RECORD がオンになっていない (つまり、分割ログ・レコードを意味する) ときには、このフィールドが、前のログ・レコードからのデータの続きまでのオフセットを示します。
COMPR_ALGORITHM	XL1	DCAP_DATA を圧縮するために使用された z/OS 圧縮アルゴリズム。
	XL1	将来の利用のために予約済み。
LOCK_SEQ_NUM	CL6	ブロック・レベルのデータ共用に IRLM SCOPE=GLOBAL が使用されたときに用いられた IRLM ロック・シーケンス番号。
LOG_INQY_DATA		FIRST_CALL がオンになっているときには、以下の LOG_INQY_DATA フィールドがログに記録される。
PSBNAME	CL8	アプリケーションの PSB 名。
TRANNAME	CL8	アプリケーションのトランザクション名。
USERID	CL8	呼び出しに対するユーザー ID。バッチ・ジョブでは、JOB ステートメントに USER= キーワードがあれば、USERID は RACF ID になる。オンライン・アプリケーションでは、USERID は PSBNAME または RACFID のいずれかである。
RACF ID	CL8	JOB ステートメントの USER= キーワードの値。
RACFID または PSB 名	CL8	オンライン・アプリケーションの場合の値。
DCAP_DATA	CL0	ログに記録されているデータ・Element。

関連概念:

 z/OS: データ圧縮展開サービス

## データ・エレメント・ヘッダーのフォーマット

データ・エレメント・ヘッダーには、ログに記録されたデータのタイプと長さを記述するフィールド、および z/OS 圧縮サービスのオプションがあります。

以下の表は、データ・エレメント・ヘッダーのフォーマットの一覧です。

表 74. ログ・レコード・ヘッダーのフォーマット

フィールド名	データ・タイプ	フィールドの説明
LOGID	XL1	ログに記録されているデータのタイプ。 X'00'- CAPD ブロック X'04'- DBD バージョン ID X'08'- 物理連結キー X'0C'- CAPD_DATA X'10'- セグメント・データ X'14'- 変更前イメージ・データ
LOG_FLAGS	XL1	フラグ。ビット定義は次のとおり。
COMPR_DATA	X'20'	データは、z/OS 圧縮サービスを使用して圧縮される。
	X'10'	将来の利用のために予約済み。
	X'08'	将来の利用のために予約済み。
	X'04'	将来の利用のために予約済み。
	X'02'	将来の利用のために予約済み。
	X'01'	将来の利用のために予約済み。
LOG_LL	H	LOG_DATA の長さ。
LOG_DATA	CL0	z/OS サービスが使用可能な場合に圧縮されるログ・データ。4 バイトのデータ・エレメント・ヘッダーの後に、ログに記録されているデータのタイプが示される。  X'00' CAPD ブロック X'04' ログ・データは DBD バージョン ID である。 X'08' ログ・データは物理連結キーである。 X'0C' CAPD_DATA X'10' ログ・データはセグメント・データである。 X'14' ログ・データは変更前イメージ・データである。  変更前イメージ・データの場合は、データの前に、セグメント内の変更データのハーフワード・オフセットが示される。ハーフワード・オフセットは圧縮されない。

### 関連資料:

338 ページの『CAPD ブロックのフォーマット (LOGID=X'00')』

341 ページの『CAPD\_DATA のフォーマット (LOGID=X'0C')』

## CAPD ブロックのフォーマット (LOGID=X'00')

CAPD ブロックには、制御を与える出口ルーチンの名前、物理データベースの名前、セグメント・コード、およびログ・レコードに関連するその他の情報を記述するフィールドが含まれます。

以下の表は、CAPD ブロックのフォーマットの一覧です。

表 75. CAPD ブロックのフォーマット (LOGID = X'00')

フィールド名	データ・タイプ	フィールドの説明
NEXT_PTR	AL4	内部使用のみ。無効。
PREVIOUS_PTR	AL4	内部使用のみ。無効。
USER_EXIT_NAME	CL8	ロギングに加えて出口ルーチンが要求されたときに、制御を与えられる出口ルーチンの名前。出口ルーチンが定義されていないか、DBCTL によってログに記録されている場合は、ブランク。
DATABASE_NAME	CL8	物理データベースの名前。
DBD_VERSION_PTR	AL4	内部使用のみ。無効。
SEGMENT_NAME	CL8	更新されて、そのログ・データが要求された物理セグメントの名前。
SEGMENT_CODE	XL1	比較のためのセグメント・コード。
FLAGS	XL1	フラグ。ビット定義は次のとおり。
DEDB_DB	X'80'	DEDB データベース。
KEY_NEEDED	X'40'	連結キーが要求されている。
DATA_NEEDED	X'20'	セグメント・データが要求されている。
PATH_NEEDED	X'10'	パス・データが要求されている。
	X'08'	IMS 用に予約済み。
	X'04'	IMS 用に予約済み。
	X'02'	将来の利用のために予約済み。
	X'01'	将来の利用のために予約済み。
	XL1	IMS 用に予約済み。
CALL_SEGMENT_LEVEL	XL1	将来の利用のために予約済み。

表 75. CAPD ブロックのフォーマット (LOGID = X'00') (続き)

フィールド名	データ・タイプ	フィールドの説明
CALL_FUNCTION	CL4	要求の呼び出し機能。
		<b>REPL</b> 置き換え呼び出し。
		<b>ISRT</b> 挿入呼び出し。
		<b>DLET</b> 削除呼び出し。
		<b>FLD</b> この更新は、フィールド呼び出しにより行われました。物理機能は「REPL」になります。
		<b>CASC</b> アプリケーション削除呼び出しの結果としてのカスケード削除。
		<b>DLLP</b> 以下の理由による、その論理パスを通しての論理親セグメントの削除。
		<ul style="list-style-type: none"> <li>• 以前にその物理パスから削除されたものとしてマークされている。</li> <li>• 物理パスと論理パスの両方から削除されやすい。</li> <li>• 最後の論理子セグメントを削除予定。</li> </ul>
		<b>Gxxx</b> サブセット・ポインタの更新が取り込まれる場合、これは、実行された Get 呼び出し (例えば、GHU、GHNP) です。
PHYS_FUNCTION	CL4	DL/I が行った物理機能。
		<b>REPL</b> セグメントが物理的に置き換えられた。
		<b>ISRT</b> セグメントが物理的に挿入された。
		<b>DLET</b> セグメントが物理的に削除された。
		<b>DLPP</b> このセグメントは、物理パス上では削除するが、論理子からそのセグメントへの論理パスがまだ存在しているので、物理的に除去してはならない。
		<b>SSPU</b> SSP は、セグメントが物理的に取得された場合、またはセグメントがパス ISRT または REPL で更新されない場合に更新されます。
CALL_TIMESTAMP	XL8	呼び出し完了の STCK タイム・スタンプ。
AREA_NAME	CL8	DEDB データベースに関する AREA 名。
LOWEST_LVL_KEY_OR_DATA	XL1	パス・データまたは連結キーの一部が追加された最低レベル番号。
	XL39	将来の利用のために予約済み。
COMMAND_CODES	0XL12	コマンド・コード。
CMD_CODE_SNGL	XL1	単一文字のコマンド・コード。
CMD_CODE_F	X'80'	コマンド・コード F。
CMD_CODE_L	X'40'	コマンド・コード L。
	XL1	将来の利用のために予約済み。
CMD_CODE_DBL	0XL10	二文字のコマンド・コード。
CMD_CODE_M	XL1	サブセット・ポインタのコマンド・コード M1 から M8 まで。(*)

表 75. CAPD ブロックのフォーマット (LOGID = X'00') (続き)

フィールド名	データ・タイプ	フィールドの説明
CMD_CODE_R	XL1	サブセット・ポインタの命令・コード R1 から R8 まで。(*)
CMD_CODE_S	XL1	サブセット・ポインタの命令・コード S1 から S8 まで。(*)
CMD_CODE_W	XL1	サブセット・ポインタの命令・コード W1 から W8 まで。(*)
CMD_CODE_Z	XL1	サブセット・ポインタの命令・コード Z1 から Z8 まで。(*)
	XL5	将来の利用のために予約済み
DB_VERSION	AL4	データベース・バージョン番号。
PART_NAME	CL8	HALDB 区画の名前。
SAVED_DLTWA	AL4	比較のために保管された DLTWA。
PHYSICAL_ROOT_RBA	AL4	ルート・セグメントの RBA。
STORAGE_SIZE	F	内部使用のみ。無効。
CONC_KEY_PTR	AL4	内部使用のみ。無効。
CONC_KEY_LENGTH	H	物理連結キーの長さ。
ROOT_KEY_LENGTH	XL1	ルート・キーの長さ。
CAPD_DATA_DIMENSION	XL1	データ配列の次元。この値は、呼び出しに関してログに記録される CAPD_DATA ブロック・エレメントの数を反映している。パス・データが要求されている場合、最初の CAPD_DATA ブロック・エレメントはルート・セグメントに関するもので、後続の CAPD_DATA ブロック・エレメントはパス上の各セグメントに関するものである。次の (パス・データでない場合は、最初の) CAPD_DATA ブロック・エレメントは、この要求に関連したセグメントに関するもので、後続の CAPD_DATA ブロック・エレメント (もしあれば) は変更前イメージ・データに関するものである (物理置き換え操作)。
CAPD_DATA_PTR(16)	16AL4	内部使用のみ。無効。

(\*): 各ビットは、対応する命令・コード番号が指定されたかどうかを表します。例えば、CMD\_CODE\_S が X'A0' の場合、S1S3 が SSA に指定されたことを意味します。

注: PATH が EXIT= ステートメントで指定された場合、これは、FLD 呼び出しのパス・データが信頼できないために、FLD 呼び出しから生成されるあらゆる X'9904' ログ・レコードに対しては無視されます。

関連概念:

327 ページの『取り込みデータのエレメント』

## CAPD\_DATA のフォーマット (LOGID=X'0C')

CAPD\_DATA ブロックには、取り込まれる物理セグメントの名前、取り込まれるデータの型、セグメントのキーの長さとおフセット、およびログ・レコードに関連するその他の情報を記述するフィールドが含まれます。

以下の表は、CAPD\_DATA のフォーマットの一覧です。

表 76. CAPD\_DATA のフォーマット (LOGID = X'0C')

フィールド名	データ・タイプ	フィールドの説明
NEXT_PTR	AL4	内部使用のみ。無効。
	CL4	将来の利用のために予約済み。
	AL8	IMS 用に予約済み。
SEGMENT_NAME	CL8	物理セグメントの名前。
SEGMENT_LEVEL	XL1	物理セグメントのレベル。
CMD_CODE_R	XL1	サブセット・ポインターのコマンド・コード R1 から R8。(*)
DATA_TYPE	XL1	取り込まれたデータのタイプ。 <b>X'00'</b> セグメント <b>X'01'</b> 変更前データ <b>X'02'</b> カスケード・データ <b>X'03'</b> セグメント・パス・データ <b>X'04'</b> 挿入されるセグメントに続くツインのセグメント・データ。
DATA_FLAGS	XL1	フラグ・バイト。ビット定義は次のとおり。
DATA_USER_IO	X'80'	ユーザー入出力域内のデータ。
DELETED_ON_PHYS_PATH	X'40'	セグメントは物理パス上で削除されている。
	X'20'	将来の利用のために予約済み。
	X'10'	将来の利用のために予約済み。
	X'08'	将来の利用のために予約済み。
	X'04'	将来の利用のために予約済み。
	X'02'	将来の利用のために予約済み。
	X'01'	将来の利用のために予約済み。
LP_KEY_LENGTH	H	セグメント・データの前に連結されている、セグメントの論理親キーの長さ。セグメントが論理子でない場合は、ゼロ。
KEY_LENGTH	H	セグメントのキーの長さ。セグメントがキーを持たない場合は、ゼロ。
KEY_OFFSET	H	セグメントのキーのおフセット。セグメントがキーを持たない場合は、ゼロ。
SEGMENT_LENGTH	H	セグメント・データの長さ。
SEGMENT_PTR	AL4	内部使用のみ。無効。

(\*): 各ビットは、対応するコマンド・コード番号が指定されたかどうかを表します。例えば、CMD\_CODE\_R が X'20' の場合、R3 が SSA に指定されていることを意味します。

関連概念:

## ジョブ終了呼び出しログ・レコードのフォーマット

ジョブ終了呼び出しログ・レコードには、レコードの長さ、リカバリー・トークン、および CPU ストア・クロックのタイム・スタンプを記述するさまざまなフィールドがあります。

以下の表に、ジョブ終了呼び出しログ・レコードのフォーマットを示します。

表 77. EOJ 呼び出しのフォーマット

フィールド名	データ・タイプ	フィールドの説明
LL	H	ログ・レコードの長さ。IMS によってレコードの最後に追加された 4 バイトのログ・シーケンス番号を含む。
ZZ	XL2	常にゼロ。
LOGCODE	XL1	X'99' ログ・レコード・コード。
SUBCODE	XL1	X'28' ログ・サブコード。
PST_NUMBER	H	PST 番号。
RECOVTKN	XL16	リカバリー単位に対するリカバリー・トークン。これは、このリカバリー単位に関するコミット・ログ・レコードまたは打ち切りログ・レコードを関連付けるために使用される。
STORE_CLOCK	XL8	プログラムが終了し、ログ・レコードが書き込まれたときの CPU ストア・クロック・タイム・スタンプ。

関連概念:

331 ページの『第 12 章 ジョブ終了 (EOJ) 呼び出しログ・レコード』

## SETS および ROLS 呼び出しログ・レコードのフォーマット

SETS および ROLS 呼び出しログ・レコードには、レコードの長さ、リカバリー・トークン、および CPU ストア・クロックのタイム・スタンプを記述するさまざまなフィールドがあります。

以下の表に、SETS および ROLS 呼び出しログ・レコードのフォーマットを示します。

表 78. SETS および ROLS 呼び出しのフォーマット

フィールド名	データ・タイプ	フィールドの説明
LL	H	ログ・レコードの長さ。IMS によってレコードの最後に追加された 4 バイトのログ・シーケンス番号を含む。
ZZ	XL2	常にゼロ。
LOGCODE	XL1	X'99' ログ・レコード・コード。
SUBCODE	XL1	SETS 呼び出しの X'30' ログ・サブコード。ROLS 呼び出しの X'34' ログ・サブコード。
PST_NUMBER	H	PST 番号。
RECOVTKN	XL16	リカバリー単位に対するリカバリー・トークン。これは、このリカバリー単位に関するコミット・ログ・レコードまたは打ち切りログ・レコードを関連付けるために使用される。

表 78. SETS および ROLS 呼び出しのフォーマット (続き)

フィールド名	データ・タイプ	フィールドの説明
STORE_CLOCK	XL8	呼び出しが完了し、ログ・レコードが書き込まれたときの CPU ストア・クロック・タイム・スタンプ。
TOKEN	CL4	呼び出しで使用された SETS/ROLS トークン。

関連概念:

333 ページの『第 13 章 SETS および ROLS 呼び出しログ・レコード』



---

## 第 4 部 データベース・リソース・アダプター (DRA)

DRA は、IMS 全機能データベースと高速処理データベース (DEDB) へのインターフェースです。アプリケーションの設計担当者は、オープン・データベース・アクセス (ODBA) インターフェースを使用するコーディネーター・コントローラー (CCTL) または z/OS アプリケーション・プログラムが DRA を使用できるように、プログラムを作成することができます。

以降のトピックは、CCTL または z/OS アプリケーション・プログラムの設計担当者向けです。IMS DB または DB/DC と CCTL との具体的な相互作用についての詳細は、CCTL 関連の資料を参照してください。

関連概念:

 ODBA アプリケーション・プログラムの作成 (アプリケーション・プログラミング)

関連タスク:

 ODBA インターフェースを介した IMS データベースへのアクセス (コミュニケーションおよびコネクション)



---

## 第 15 章 スレッド概念

DRA スレッドとは、CCTL タスクまたは z/OS アプリケーション・プログラムのタスクを、これらの呼び出しを処理可能な IMS DB タスクと結び付ける DRA 構造のことです。1 つの DRA スレッドが、すべての CCTL スレッドまたは ODBA スレッドと関連付けられます。CCTL スレッドは、一度に複数の DRA スレッドと接続することはできません。

DRA スレッド とは、次のものを結び付ける DRA 構造を言います。

- IMS DB に対してデータベース呼び出しを行う CCTL タスクと、その呼び出しを処理する IMS DB タスク。CCTL スレッドは、DRA を使って IMS DB 要求を出す CCTL タスクです。
- IMS DB に対してデータベース呼び出しを行う z/OS アプリケーション・プログラムと、その呼び出しを処理する IMS DB タスク。ODBA スレッドは、DRA を使って IMS DB 呼び出しを行う z/OS タスクです。

1 つの DRA スレッドが、すべての CCTL スレッドまたは ODBA スレッドと関連付けられます。CCTL スレッドは、一度に複数の DRA スレッドと接続することはできません。

DRA オープン・スレッド・オプションが使用された場合、スレッドは要求に対してスケジュールされると、IMS にサインオンします。この動作が原因となり、IMS コマンド /DISPLAY CCTL、/DISPLAY ACTIVE REGION、および /DISPLAY ACTIVE THREAD は、MINTHRD 出力フィールドに非オープン・スレッド・システム構成とは異なる値を出力します。スレッド TCB は接続されないため、MINTHRD は、DRA 初期設定時に接続されるスレッドの数には適用されません。しかし、MINTHRD は初期設定中に IMS が割り振るスレッド構造の数を示します。IMS が PSB のスケジューリングを開始した後、それらのスレッドは正常にコマンド出力に表示されます。

---

### スレッドの処理

CCTL スレッドと ODBA スレッドでは、DRA による処理方法が異なります。CCTL または z/OS アプリケーション・プログラムはそれぞれ、DRA スレッドの作成、DRA スレッド・ブロックの割り振りによって、要求を発行します。

#### CCTL スレッドの処理

CCTL アプリケーション・プログラムで IMS DB データベースからのデータが必要になると、CCTL タスクが PSB に対し SCHED 要求を出します。この SCHED 要求を処理するため、DRA は DRA スレッドを作成しなければなりません。それには、DRA が使用可能な DRA スレッド TCB を選択し、それに CCTL スレッド・トークン (一意のトークン。CCTL はこれを SCHED PAPL PAPLTTOK に置きます) と自身の IMS DB タスクを割り当てます。このタスクが PSB をスケジュー

ールします。スケジューリングに成功すると、その DRA スレッド接続は完成と見なされ、特定の DRA スレッド TCB を用いて、CCTL スレッドを IMS DB タスクに接続できます。

この CCTL スレッドから出されるそれ以後の DRA 要求は、どれも同じ CCTL スレッド・トークンを使わなければなりません。同じトークンを使うことで、要求が正しい DRA スレッドに行くことが保証されます。アプリケーション・プログラムが終了し、CCTL スレッドがもはや DRA スレッドのサービスを必要としなくなると、CCTL は TERMTHRD (スレッド終了) 要求を出し、DRA スレッド TCB から CCTL スレッド・トークンを取り除き、DRA スレッドを終了させます。これにより、そのスレッド TCB は新しい DRA スレッドの一部になることができます。

## ODBA スレッドの処理

z/OS アプリケーション・プログラムで IMS データベースからのデータが必要になると、ODBA タスクが APSB 呼び出しを行い、ODBA 環境を初期化します。APSB 呼び出しを処理するために、DRA は DRA スレッド・ブロックを割り振り、それに ODBA スレッドと自身の IMS DB タスクを割り当てます。このタスクが PSB をスケジューリングします。スケジューリングに成功すると、その DRA スレッド接続は完成と見なされ、特定の DRA スレッド・ブロックを用いて ODBA スレッドを IMS DB タスクに接続できます。

アプリケーション・プログラムが終了し、ODBA スレッドがもはや DRA スレッドのサービスを必要としなくなると、ODBA アプリケーションは DPSB 呼び出しを行い、DRA スレッドを終了させます。これにより、そのスレッド・ブロックは新しい DRA スレッドの一部になることができます。

---

## マルチスレッドの処理

一度に複数のスレッドを処理する DRA の機能は、マルチスレッド化 として知られています。マルチスレッド化は、複数の CCTL スレッドまたは ODBA スレッドが同時に DRA を使用できることを意味します。DRA 要求と ODBA 呼び出しは、すべてマルチスレッド化できます。

### 複数 CCTL スレッドの処理

マルチスレッド化機能を使用するには、

- DRA を複数のスレッド TCB で初期化しなければなりません。DRA を複数のスレッド TCB で初期化するには、MAXTHRD パラメーター (DRA 始動テーブル) に 1 より大きな値を指定してください。
- CCTL が複数の CCTL スレッドを並行処理できなければなりません。
- CCTL に中斷・再開出口ルーチンがなければなりません。DRA はこれらのルーチンを用いて、スレッド処理の状況を CCTL に通知します。

### 複数 ODBA スレッドの処理

マルチスレッド機能を使用するには、DRA を複数の DRA スレッドで初期化しなければなりません。それには、MAXTHRD パラメーター (DRA 始動テーブル) に 1 より大きな値を指定してください。

## CCTL マルチスレッドの例

以下の例で、マルチスレッド・システムでの並行処理の概念を説明します。

以下の例に、マルチスレッド化システムで起こるイベントを上から下へ発生順に示します。並行処理の概念を分かりやすく示すために、図を 2 つの欄に分けてあります。

図には、2 つの CCTL スレッドと 2 つの DRA スレッドが示されています。CCTLRTNA は、この例における CCTL ルーチンのモジュール名です。この CCTL ルーチンが PAPL を作成し、DFSPRRC0 を呼んで DRA 要求を処理させます。

表 79. マルチスレッド・システムにおけるイベントの例

CCTL TCB イベント	DRA TCB イベント
<p>アプリケーション・プログラム 1 で PSB が必要となり、CCTL スレッド 1 が作られる。 CCTL スレッド 1 のイベント：</p> <ul style="list-style-type: none"><li>• CCTLRTNA が SCHED PAPL を作成し、DFSPRRC0 を呼ぶ。</li><li>• DFSPRRC0 が DRA スレッドを作成し、スレッド・トークン (PAPLTOK) が DRA スレッド TCB1 に割り当てられる。</li><li>• DFSPRRC0 がスレッド TCB1 を活動化する。</li><li>• DFSPRRC0 が延期出口ルーチンと呼ぶ。</li><li>• 延期出口ルーチンが CCTL スレッド 1 を中断させる。</li></ul> <p>CCTL は、当該 CCTL TCB に対して他の CCTL スレッドをディスパッチできる。 アプリケーション・プログラム 2 で PSB が必要となり、CCTL スレッド 2 が作られる。 CCTL スレッド 2 のイベント：</p> <ul style="list-style-type: none"><li>• CCTLRTNA が SCHED PAPL を作成し、DFSPRRC0 を呼ぶ。</li><li>• DFSPRRC0 が DRA スレッドを作成し、当該スレッド・トークン (PAPLTOK) が DRA スレッド TCB2 に割り当てられる。</li><li>• DFSPRRC0 がスレッド TCB2 を活動化する。</li><li>• DFSPRRC0 が延期出口ルーチンと呼ぶ。 延期出口ルーチンが CCTL スレッド 2 を中断させる。</li></ul>	<p>DRA スレッド TCB1 のイベント：</p> <ul style="list-style-type: none"><li>• DRA が SCHED 要求を処理し、スケジューラ・プロセスを実行するよう IMS DB に依頼する。</li><li>• スケジューリングが進行する。</li></ul> <p>DRA スレッド TCB2 のイベント：</p>

表 79. マルチスレッド・システムにおけるイベントの例 (続き)

CCTL TCB イベント	DRA TCB イベント
<p>2 つのスレッドがともに中断されている。 CCTL TCB は、いずれかのスレッドが実行を再開するまで非アクティブ状態を続ける。</p> <p>CCTL TCB がアイドルなので、スレッド 2 はただちに実行を再開できる。延期出口ルーチンによってスレッドが中断された地点の直後から実行が再開される。</p> <p>スレッド 2 の実行が再開されたばかりのため、スレッド 1 は再開出口ルーチンが使用可能になるまで待たなければならない。</p> <p>CCTL スレッド 2 のイベント :</p> <ul style="list-style-type: none"> <li>• 延期出口ルーチンが呼び出し元の DFSPRRC0 に制御を返す。</li> <li>• DFSPRRC0 が CCTLRINA に制御を返す。</li> <li>• CCTLRINA が SCHED PAPL から結果を受け取り、それをアプリケーション・プログラム 2 に渡す。</li> <li>• CCTLRINA がスレッド 2 の SCHED 要求を終了させる。</li> </ul> <p>CCTL TCB でスレッド 2 が完了した後、CCTL は現在待機中のスレッド 1 をディスパッチできる。</p> <p>CCTL スレッド 1 のイベント :</p>	<p>DRA スレッド TCB2 のイベント :</p> <ul style="list-style-type: none"> <li>• DRA が SCHED 要求を処理し、スケジュール・プロセスを実行するよう IMS DB に依頼する。</li> <li>• スケジューリングが進行する。</li> </ul> <p>TCB2 のスケジューリングが先に終わる。</p> <p>DRA スレッド TCB1 のイベント :</p> <ul style="list-style-type: none"> <li>• IMS DB でスケジューリングが完了し、PAPL に結果が置かれる。</li> <li>• DRA が再開出口ルーチン呼び、PAPL を CCTL に返す。</li> <li>• 再開出口ルーチンがスレッド・トークン (PAPLTTOK) を見て、CCTL スレッド 2 に「再開可能」のフラグを立てる。</li> <li>• 再開出口ルーチンが DRA に制御を返し、TCB2 が非アクティブになる。</li> </ul> <p>TCB1 スケジューリングが完了する。</p> <p>DRA スレッド TCB1 のイベント :</p> <ul style="list-style-type: none"> <li>• IMS DB でスケジューリングが完了し、PAPL に結果が置かれる。</li> <li>• DRA が再開出口ルーチン呼び、PAPL を CCTL に返す。</li> <li>• 再開出口ルーチンがスレッド・トークン (PAPLTTOK) を見て、CCTL スレッド 1 に「再開可能」のフラグを立てる。</li> <li>• 再開出口ルーチンが DRA に制御を返し、TCB1 が非アクティブになる。</li> </ul>

表 79. マルチスレッド・システムにおけるイベントの例 (続き)

CCTL TCB イベント	DRA TCB イベント
<ul style="list-style-type: none"> <li>• 延期出口ルーチンが呼び出し元の DFSPRRC0 に制御を返す。</li> <li>• DFSPRRC0 が CCTLRTNA に制御を返す。</li> <li>• CCTLRTNA が SCHED PAPL から結果を受け取り、それをアプリケーション・プログラム 1 に渡す。</li> <li>• CCTLRTNA がスレッド 1 の SCHED 要求を終了させる。</li> </ul> <p>CCTL スレッド 2 のイベント :</p> <ul style="list-style-type: none"> <li>• CCTLRTNA は DL/I PAPL を作成し、DFSPRRC0 を呼び出します。</li> <li>• DFSPRRC0 は正しい DRA スレッドを検出します。</li> <li>• DFSPRRC0 がスレッド TCB2 を活動化する。</li> <li>• DFSPRRC0 が延期出口ルーチンを呼ぶ。</li> <li>• 延期出口ルーチンは CCTL スレッド 2 を中断します。</li> </ul> <p>CCTL スレッド 1 のイベント :</p> <ul style="list-style-type: none"> <li>• CCTLRTNA は DL/I PAPL を作成し、DFSPRRC0 を呼び出します。</li> <li>• DFSPRRC0 は正しい DRA スレッドを検出します。</li> <li>• DFSPRRC0 がスレッド TCB1 を活動化する。</li> <li>• DFSPRRC0 が延期出口ルーチンを呼ぶ。</li> <li>• 延期出口ルーチンが CCTL スレッド 1 を中断させる。</li> </ul> <p>application program2 が完了します。 CCTL はこの UOR の処理をコミットするために同期点要求を IMS DB に発行します。 CCTL は、フェーズ 1 要求を発行する前に、application program2 の UOR に未確定フラグを立てます。 CCTL は、IMS DB へのフェーズ 2 呼び出しを正常に実行できるようになるまで、この未確定 UOR の記憶を保持します。</p>	<p>DRA スレッド TCB1 のイベント :</p> <ul style="list-style-type: none"> <li>• DL/I 呼び出しが進行中です。</li> <li>• DRA は DL/I PAPL を処理し、IMS DB に DL/I 処理の実行を要求します。</li> </ul>
<p>CCTL スレッド 2 のイベント :</p>	

表 79. マルチスレッド・システムにおけるイベントの例 (続き)

CCTL TCB イベント	DRA TCB イベント
<ul style="list-style-type: none"> <li>• CCTLRTNA は PREP 要求を発行し、DFSPRRC0 を呼び出します。</li> <li>• DFSPRRC0 がスレッド TCB2 を活動化する。</li> <li>• DFSPRRC0 が延期出口ルーチンを呼ぶ。</li> <li>• 延期出口ルーチンは CCTL スレッド 2 を中断します。</li> </ul>	<p>DRA スレッド TCB2 のイベント :</p> <ul style="list-style-type: none"> <li>• DRA は PREP 要求を IMS DB に送信します。</li> <li>• IMS DB はフェーズ 1 完了をログに記録します。この IMS DB UOR は現在未確定です。</li> <li>• PREP 要求が正常に完了し、再開出口ルーチンを呼び出します。</li> </ul>
<p>PREP 要求の結果が CCTLRTNA に戻され (PREP PAPL によって)、thread2 が非アクティブになります。</p>	<p>DRA スレッド TCB1 のイベント :</p> <ul style="list-style-type: none"> <li>• IMS DB が DL/I 障害を検出します。この障害により、この DRA スレッドは終了し、このスレッド TCB も終了します。</li> <li>• このスレッドはスケジュール状態にあったため、DRA は UPSTOR 情報を DL/I PAPL に入れてから、状況出口ルーチンを呼び出し、DL/I PAPL を CCTL に渡します。</li> <li>• 状況出口ルーチンは UPSTOR を CCTL スレッドに関連付け、PAPL を DRA に渡します。</li> <li>• DRA は再開出口ルーチンを呼び出します。</li> <li>• DRA は SDUMP を取り、スレッド TCB1 を終了させます。</li> </ul>
<p>CCTL スレッド 1 のイベント :</p> <ul style="list-style-type: none"> <li>• thread1 が再開されたあと、制御が CCTLRTNA に渡されます。</li> <li>• CCTL は、DL/I 要求が失敗したことを通知し (PAPLRETC=4 を戻す)、UPSTOR を含んでいる診断ダンプを取ります。</li> <li>• thread1 が失敗したため、CCTL はこの CCTL スレッドを終了させ、UPSTOR を解放します。</li> </ul> <p>CCTL が thread2 についてのコミット要求を送信する前に、IMS DB が失敗します。</p>	<p>DRA TCB イベント:</p>

表 79. マルチスレッド・システムにおけるイベントの例 (続き)

CCTL TCB イベント	DRA TCB イベント
<p>CCTL は、IMS DB 未確定 UOR の項目が再同期リストにあるため、この未確定状況を解決するために新規タスクを作成します。</p> <p>CCTL thread3 イベント:</p> <ul style="list-style-type: none"> <li>• CCTL は、再同期リスト内の未確定 UOR を CCTL 独自のリスト内の未確定 UOR と突き合わせます。CCTL 未確定 UOR には、CCTL のフェーズ 2 のアクションとしてコミット処理のフラグが立てられています。</li> <li>• CCTLRTNA は RESYNC を DFSPRRC0 に発行して、コミット処理を要求します。RESYNC は DRA 要求であり、スレッド要求ではありません。</li> <li>• DFSPRRC0 は DRA TCB を活動化して要求を処理し、延期出口ルーチン呼び出します。</li> <li>• thread3 が再開されたあと、CCTLRTNA は戻りコード PAPLRETC=0 を受け取ります。</li> </ul>	<ul style="list-style-type: none"> <li>• DRA は制御出口ルーチン呼び出して、IMS DB が失敗したことを CCTL に通知します。</li> <li>• 制御出口ルーチンは、IMS DB との再接続を DRA に指示するコード (PAPLRETC=8) を戻します。</li> <li>• DRA は既存のすべてのスレッド TCB を終了させます。CCTL がこれらの終了済み DRA スレッドに対して要求を出すと、DRA は、要求を処理できないことを示す戻りコードで応答します。</li> <li>• IMS DB が再始動された後、DRA は IMS DB に正常に接続します。</li> <li>• DRA は制御出口ルーチン呼び出して、DRA が IMS DB に正常に接続したことを CCTL に通知します。</li> <li>• DRA は再同期リスト (PAPLRST) のアドレスを CCTL に渡します。このリストには、CCTL thread2 用の IMS DB 未確定 UOR の項目があります。</li> <li>• 制御出口ルーチンは、実行の続行を DRA に指示するコード (PAPLRETC=0) を戻します。</li> <li>• DRA は新規 DRA スレッド TCB を作成することにより、セットアップ・プロセスを完了します。</li> <li>• DRA は IMS DB を呼び出して IMS DB UOR をコミットします。</li> <li>• 処理が正常に終了すると、DRA は再開出口ルーチン呼び出します。</li> </ul>

表 79. マルチスレッド・システムにおけるイベントの例 (続き)

CCTL TCB イベント	DRA TCB イベント
<ul style="list-style-type: none"><li>• RESYNC 要求が正常に実行されたため、CCTL は CCTL 未確定 UOR を廃棄します。</li></ul>	

---

## 第 16 章 同期点

リソースに対する変更は、同期点処理で最終的に確定します。同期点要求は、変更されたリソースに対してとるべき処置 (例えば、コミットか、アボートか、など) を指定します。同期点とは、IMS DB が実際にその要求を処理する時点をいいます。

同期点は、どれもリカバリー単位 (UOR) に基づいて決められます。データベース・リソースが割り振られ、更新可能となってから、変更のコミット要求もしくはアボート要求が出されるときまでが UOR です。普通、UOR は CCTL SCHED (PSB のスケジューリング) 要求か ODBA APSB 呼び出しで始まり、同期点要求で終わります。他の DRA スレッド要求も、同様に UOR の起点と終点を定義できます。

CCTL UOR は、リカバリー・トークン (PAPLR TOK) で特定されます。これは、新しい UOR を作成するスレッド要求の一部として渡されるもので、長さは 16 バイトです。最初の 8 バイトには、CCTL ID が入っています。この ID は、INIT 要求の USERID または PAPLUSID から確定された最後の DRA 始動パラメーター、CCTL ID と同じものです。次の 8 バイトは、UOR ごとに CCTL から指定される固有 ID です。

関連資料 : DRA スレッド要求の詳細については、373 ページの『第 22 章 CCTL が働きかける DRA 機能要求』にある要求の説明を参照してください。

IMS DB は、CCTL または ODBA アプリケーションが同期点を決定し、それに基づいて要求を行うものと想定しています。CCTL について見ると、CCTL が同期点マネージャーであり、当該 UOR と関連付けられているすべての他リソース・マネージャー (IMS DB 以外のマネージャーも含みます) との間で同期点処理を調整します。一方、ODBA アプリケーションでは、z/OS リソース・リカバリー・サービス が同期点マネージャーであり、当該 UOR と関連付けられているすべての他リソース・マネージャー (IMS 以外のマネージャーも含みます) との調整に当たります。

扱うリソース・マネージャーが 1 つだけなら、CCTL は同期点の要求に 1 フェーズと 2 フェーズのどちらの同期点プロトコルでも使用できます。しかし、複数のリソース・マネージャーを扱うときは、2 フェーズの同期点プロトコルが必要です。単一フェーズ同期点要求を出せるのは、CCTL が UOR のコミットを決定したとき、その UOR によって変更されたリソースの全部を IMS DB が所有している場合です。

ODBA アプリケーションは、IMS データベースに変更をコミットするのに必ず 2 フェーズの同期点プロトコルを使用しなければなりません。

関連資料:

379 ページの『SCHED 要求』

383 ページの『SYNTERM 要求』

385 ページの『PREP 要求』

386 ページの『COMTERM 要求』

## 2 フェーズ・コミット・プロトコル

2 フェーズ同期点プロトコルでは、当該 UOR に関係するリソース・マネージャーのそれぞれに対し、同期点マネージャーが 2 つの要求を発行します。未完了状態と未確定状態というそれぞれの UOR 状態によって、スレッドで障害が発生したときに UOR に何が起こるかが決まります。

### フェーズ 1

同期点マネージャーからすべての参加者に対し、UOR をコミットする用意があるかどうかを問い合わせます。

### フェーズ 2

フェーズ 1 で出した問い合わせへの応答に基づいて、同期点マネージャーが各参加者に対し、コミットするよう、もしくは打ち切るよう伝えます。

UOR には、飛行中と未確定という 2 つの状態があります。UOR は、作成されてから、IMS DB がフェーズ 1 の終了をログに記録する時点 (以下の 2 つの表の C 点) までは、飛行中状態に置かれます。C 点から、IMS DB がフェーズ 2 をログに記録する時点 (以下の 2 つの表の D 点) までは、UOR は未確定状態となります。

単一フェーズ同期点要求の未確定状態は、表 1 の C 点から D 点までの瞬間的な状態です。

スレッドで障害が発生したとき、UOR に何が起こるかは、飛行中状態と未確定状態で異なります。その意味で、この 2 つの状態は重要です。IMS DB UOR が飛行中のときスレッドで障害が発生すると、UOR データベース変更はバックアウトされます。IMS DB UOR が未確定のときスレッドで障害が発生すると、単一フェーズ・コミットでは、個々のスレッドの異常終了なら UOR データベース変更が保持され、システム異常終了なら保持されません。IMS DB UOR が未確定のときスレッドで障害が発生すると、2 フェーズ・コミットではデータベース変更が保持されます。

スレッドの障害とは、次のいずれかを意味します。

- 個々のスレッドの異常終了。
- システムの異常終了：IMS DB の障害、CCTL の障害、ODBA アプリケーションの障害、z/OS の障害 (すべてのスレッドが異常終了します)。

次のコードは、単一フェーズ同期点処理に CCTL を使用したときに起こるシステム・イベントを示しています。

時間 →  
—A—B—C—D—E—

表 80. CCTL 単一フェーズ同期点処理

時点	システム・イベント
A	CCTL フェーズ 1 送信
B	IMS DB フェーズ 1 受信
C	IMS DB ログ・フェーズ 1 終了
D	IMS DB ログ・フェーズ 2

表 80. CCTL 単一フェーズ同期点処理 (続き)

時点	システム・イベント
E	CCTL フェーズ 2 受信

次の表に、2 フェーズ同期点処理に CCTL を使用したときに起こるシステム・イベントを示します。



表 81. CCTL 2 フェーズ同期点処理

時点	システム・イベント
A	CCTL フェーズ 1 送信
B	IMS DB フェーズ 1 受信
C	IMS DB ログ・フェーズ 1 終了
D	IMS DB フェーズ 1 応答
E	CCTL フェーズ 1 受信
F	CCTL フェーズ 2 送信
G	IMS DB フェーズ 2 受信
H	IMS DB ログ・フェーズ 2
J	IMS DB フェーズ 2 応答
K	CCTL フェーズ 2 受信

以下の図に、ODBA を使って 2 フェーズ同期点処理を行ったときに起こるシステム・イベントを示します。

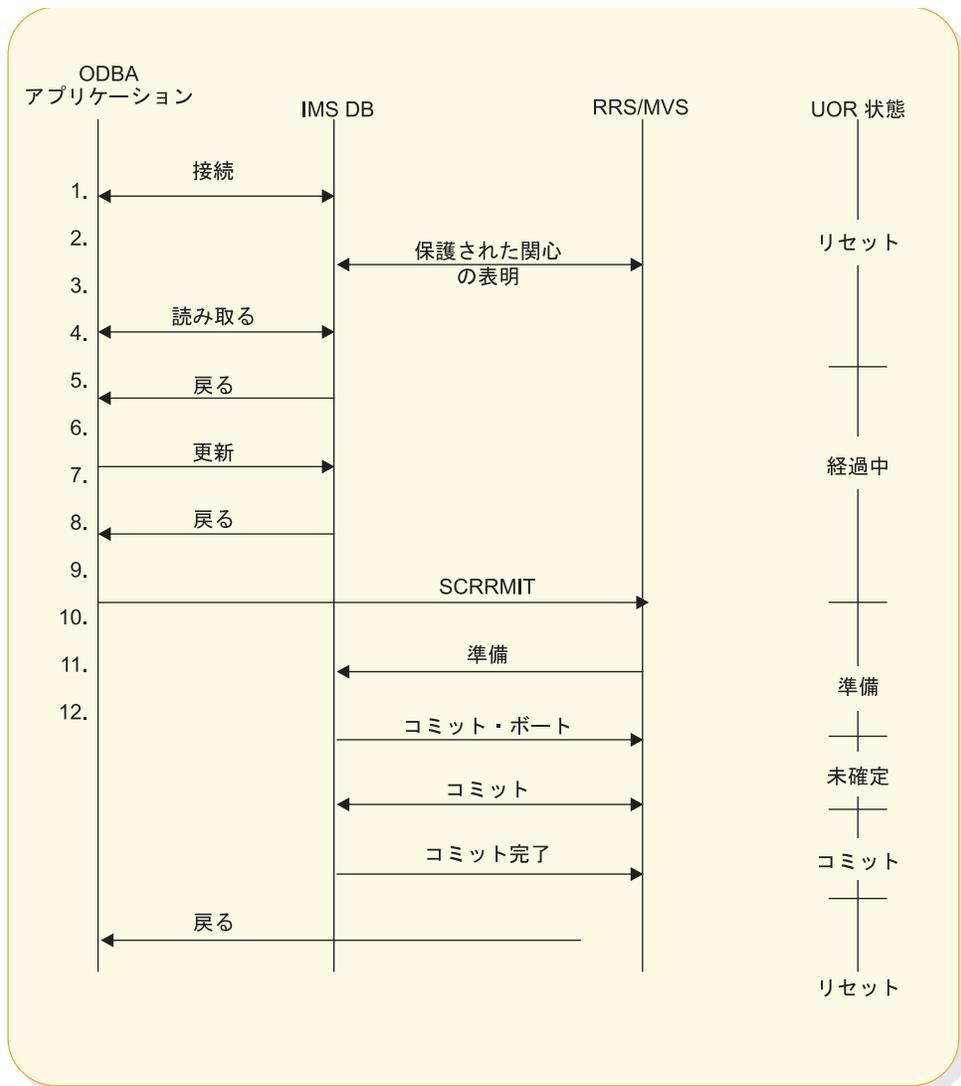


図 8. ODBA 2 フェーズ同期点処理

注:

1. ODBA アプリケーションと IMS DB が ODBA インターフェースを使用して接続します。
2. ODBA アプリケーションによって開始された作業に対し、IMS が保護された関心を表明します。この表明により、z/OS リソース・リカバリー・サービスは IMS が 2 フェーズ・コミット・プロセスに参加するものと見なします。
3. ODBA アプリケーションから IMS リソースに対して読み取り要求が出されず。
4. ODBA アプリケーションが保護リソースを更新します。
5. 制御が ODBA アプリケーションに戻り、更新要求の直後に渡されます。
6. ODBA アプリケーションが SRRCMIT を呼び、更新を定着させるよう要求します。
7. RRS が、準備 (フェーズ 1) 処理を行うために、IMS を呼び出す。
8. IMS が RRS にコミットについてのポートを戻す。

9. RRS が、コミット (フェーズ 2) 処理を行うために、IMS を呼び出す。
10. IMS が RRS に、フェーズ 2 を完了したことを知らせる。
11. 制御が ODBA アプリケーションに戻り、コミット要求の直後に渡されます。

---

## 2 フェーズ同期処理における未確定状態

IMS DB UOR は、フェーズ 2 要求が受信されるまで未確定状態にとどまります。このプロセスを「未確定状態の解決」と呼びます。UOR が未確定状態にある間、その UOR によって所有されているデータベース・リソースは、他の要求にとってアクセス不能です。未確定状態は、ただちに解消することが重要です。

### CCTL の例

IMS DB の障害のために未確定 UOR が作成された場合には、未確定 UOR を解消するために、以下の手順が行われなければなりません。

1. IMS DB の再始動後、CCTL は INIT 要求によって IMS DB に自己を証明しなければなりません。
2. この自己証明に成功すると、DRA が CCTL 制御出口に連絡し、未確定状態にある IMS DB UOR のリストを引き渡します。
3. CCTL は RESYNC 呼び出しを行い、未確定 UOR を 1 つずつ解消していかねばなりません。RESYNC 呼び出しでは、コミットまたはアボートというフェーズ 2 アクションが起こります。
4. CCTL が IMS DB の未確定 UOR を解消するには、その UOR のレコードと、それに対して取るべき適切なフェーズ 2 アクションが必要です。この例では、IMS DB 未確定 UOR の CCTL レコードを推移 UOR と呼んでいます。
5. CCTL は、A から K 間隔 (357 ページの表 81 を参照) の推移 UOR を定義しなければなりません。この間隔には IMS DB 未確定期間の C から H が含まれていますから、CCTL はすべての未確定状態を解消できます。

CCTL が推移 UOR を E から K 間隔と定義し、スレッドが C と D の間にあるときに IMS DB で障害が発生すると、フェーズ 1 の呼び出しで起こった障害であるにもかかわらず、CCTL が該当する推移 UOR を持たない未確定 UOR IMS DB に生じることになります。CCTL は、識別処理でこの UOR を解消することができません。この未確定状態を解消するには、IMS DB コマンドの /CHANGE CCTL を使用する以外に方法がありません。

ODBA では、すべての未確定状態は z/OS リソース・リカバリー・サービスを通じて解消されます。



---

## 第 17 章 DRA 始動テーブル

データベース・リソース・アダプター (DRA) 始動テーブルには、DRA の特性定義に使用される値が含まれています。これらのモジュールに変更を加えて、正しく DRA パラメーターを指定するようにならなければなりません。

DRA 始動テーブルは、次のモジュールをアセンブルして作成します。

- DFSPZPxx モジュール (CCTL 用)
- DFSxxxx0 モジュール (ODBA 用)

CCTL または ODBA のシステム・プログラマーは、必要な変更をこれらのモジュールに加え、DRA パラメーターを正しく指定しなければなりません。DRA パラメーターは、DFSPRP マクロ呼び出しでキーワードとして指定されます。

### サンプル DFSPZP00 ソース・コード

```
DFSPZP00 CSECT
          DFSPRP DSECT=NO,FPBUF=10,FPBOF=5,CNBA=60,MINTHRD=3,MAXTHRD=6,DSNAME=IMS.SDFSRESL
          END
```

### DFSPRP マクロ・キーワード

キーワード  
説明

#### AGN=

このキーワードは使用されなくなりました。指定すると無視されます。

#### CNBA=

CCTL または ODBA が使用する高速機能 NBA バッファの総数。

#### DBCTLID=

IMS DB または DB/DC 領域の 4 文字名。これは、DBC プロシージャの IMSID パラメーターと同じです。デフォルト名は SYS1 です。

#### DDNAME=

IMS DB 実行ライブラリーの動的割り振りに使用される 1 から 8 文字の DD 名。デフォルトの DD 名は CCTLDD です。

#### DSNAME=

IMS DB 実行ライブラリーの 1 から 44 文字のデータ・セット名。このライブラリーは DRA モジュールを取めていなければならない、z/OS で許可されていないなければなりません。デフォルトの DSNAME は、IMS.SDFSRESL です。このライブラリーに DRA モジュールがなければなりません。

#### FPBOF=

スレッドごとに割り振られる高速機能 DEDB オーバーフロー・バッファの数。デフォルトは 00 です。

#### FPBUF=

スレッドごとに割り振られ、固定される高速機能 DEDB バッファの数。デフォルトは 00 です。

#### **FUNCLV=**

CCTL または ODBA がサポートする DRA レベルを指定します。デフォルトは 3 です。

#### **GENSNAP=**

スレッド終了時に DFSPAT20 で SNAP 出力を生成するか抑止するかを指定します。

**YES** スレッド終了時に DFSPAT20 で SNAP 出力を生成します。YES がデフォルトです。

**NO** スレッド終了時に DFSPAT20 で SNAP 出力の生成を抑止します。

#### **IDRETRY=**

最初の IDENTIFY の試みが失敗に終わった後、z/OS アプリケーション領域が IMS に対して IDENTIFY (もしくは ATTACH) しようと試みる回数。最大値は 255、デフォルトは 0 です。

#### **IMSPLEX=**

1 文字から 5 文字のユーザー指定の ID。この ID が「CSL」に連結されて、z/OS システム間カップリング・ファシリティ CSL IMSPLEX グループ名が作成されます。ここに指定する値は、SCI 始動プロシージャで指定された IMSPLEX NAME= の値と一致していなければなりません。IMS ODBA インターフェースは、CSLSCREG インターフェースを使用して、この XCF CSL IMSPLEX グループを SCI に登録します。そのため、ターゲット ODBM アドレス・スペースは、SCI に登録するときに、これと同じ XCF CSL IMSPLEX グループ名を指定する必要があります。このパラメータを指定すると、IMS ODBA インターフェースは、呼び出しを直接 IMS に経路指定する代わりに、ODBM アドレス・スペースに経路指定します。IMSPLEX= を指定しない場合、ODBA 呼び出しは直接 IMS に経路指定されます。

このパラメータは、ODBA 呼び出しを ODBA アドレス・スペースを介して経路指定するアプリケーションの場合には必須です。デフォルト値はありません。

#### **MAXTHRD=**

一度に使用できる DRA スレッド TCB の最大数。最大数は 4095 です。デフォルトは、数値 1 です。

#### **MINTHRD=**

一度に使用できる DRA スレッド TCB の最小数。最大数は 4095 です。デフォルトは、数値 1 です。

DRA オープン・スレッド TCB オプションがアクティブである場合、この値は IMS にサインオンしている DRA スレッドの数を示します。

DRA オープン・スレッド TCB オプションがアクティブでない場合、この値は IMS に接続してサインオン状態のままである DRA スレッド TCB の数を示します。

#### **ODBMNAME=**

ODBM アドレス・スペースの名前を指定します。ODBA インターフェースは CSLDMREG 要求を使用して、ここに指定されたアドレス・スペースに登録し、CSLDMI インターフェースを使用して ODBA 呼び出しをこのア

ドレス・スペースに経路指定します。これは、オプション・パラメーターです。デフォルトはありません。 ODBMNAME= を指定しない場合、ODBA インターフェイスが、すべての ODBA 呼び出しを経路指定する ODBM アドレス・スペースを選択します。

#### OPENTHRD=

DRA オープン・スレッド・サポート処理を使用可能にするかどうかを指定します。使用可能にした場合、しかも CICS® 4.2 以上を使用している場合、このオプションは DRA に対して、専用の IMS DRA スレッド・タスク制御ブロック (TCB) を接続しないよう指示します。代わりに、CICS TCB が使用されます。これは、CICS/DRA 環境での並列処理をより多くするためです。

CCTL オープン・スレッド処理が使用可能に設定されます。

#### DISABLE

オープン・スレッド処理が使用不可に設定されます。これはデフォルトです。

#### SOD=

異常なスレッド終了の SNAP DUMP に使用される出力クラス。デフォルトは A です。

#### TIMEOUT=

(CCTL のみ) DRA TERM 要求が正常に完了するまで CCTL が待つ時間 (秒数)。 CCTL アプリケーションがこのキーワードを使用するようにコーディングされている場合のみ、指定してください。 INIT 要求の完了時には、この値が CCTL に返されます。

#### TIMER=

INIT 要求で、DRA が IMS DB または DB/DC に対する自己証明に失敗したとき、次に自己証明を試みるまでの時間 (秒数)。デフォルトは 60 秒です。

#### TIMETHREADCPU=

DRA オープン・スレッド・サポートが使用可能である場合に、DRA が、DRA スレッドに関連した CPU 使用統計をモニターして報告するかどうかを指定します。時間は、IMS 07 ログ・レコードの DLRTIME フィールドで報告されるほか、DRA スレッド統計フィールド PAPLCTM1 内の CCTL に対しても報告されます。

**YES** DRA は、CPU 使用統計をモニターして報告します。値 YES は、CCTL によって指定されたあらゆる値をオーバーライドします。YES がデフォルトです。

**NO** DRA は、CPU 使用統計をモニターおよび報告しません。値 NO は、CCTL によって指定されたあらゆる値をオーバーライドします。

#### CLIENT

DRA は、INIT 呼び出しを使用して DRA クライアントによって指定された設定 (YES または NO) を使用します。クライアントが設定を指定していない場合、YES がデフォルトです。

**USERID=**

CCTL 領域または ODBA 領域の 8 文字からなる名前。このキーワードは、ODBA 領域には無視されます。

## 関連概念:

 [DEDB または MSDB バッファークラスの設計 \(データベース管理\)](#)

## 関連資料:

 [DBC プロシージャ \(システム定義\)](#)

---

## 第 18 章 CCTL に対する DRA の有効化

CCTL に対して DRA を有効にするには、以下の主要な 2 つのステップが必要です。両方のステップを完了すると、DRA が他の要求を処理できるようになります。

この節では、2 つのステップからなる DRA の有効化手順を説明します。

1. コーディネーター・コントローラー (CCTL) システムは、DRA 始動/ルーター・ルーチン (DFSPRRC0) を CCTL ロード・ライブラリーにロードしなければなりません。DFSPRRC0 は IMS 製品の一部として出荷されますが、CCTL アドレス・スペースで稼動します。また、CCTL が使用する IMS DRA モジュールのバージョンは、CCTL の通信先の IMS と同じバージョンであることが必要です。

推奨事項:

- IMS.SDFSRESL ライブラリーを CCTL ステップ・ライブラリーに連結して、DRA 始動/ルーター・ルーチン (DFSPRRC0) の正しいバージョンが CCTL ロード・ライブラリーにロードされるようにします。
  - DRA 始動テーブル (DFSPZPxx) が必ず正しいバージョンの IMS.SDFSRESL を指すようにしてください。
2. システム・プログラマーは、DFSPZPxx ロード・モジュールをロード・ライブラリーに収めなければなりません。これで DRA を初期化する準備が整いました。

CCTL アプリケーションが初期設定 (INIT) 要求を出すと、CCTL が初期設定プロセスを開始します。この時点で、CCTL は DFSPRRC0 をロードし、INIT 要求を処理するよう DRA に呼びかけます。

CCTL アプリケーションは、初期設定要求の一部として、始動テーブル名接尾部 (xx) を指定します。デフォルトのロード・モジュール DFSPZP00 は IMS.SDFSRESL ライブラリーにあります。

INIT 要求の処理後、DRA は IMS DB に対して自己証明を行い、これで、他の要求を処理できるようになります。

DFSPZP00 には、DRA 初期設定パラメーターのデフォルト値が含まれています。デフォルト以外の値を指定したいときは、独自のモジュール (名前は DFSPZPxx) を書き、アセンブルして、CCTL ロード・ライブラリーにロードしてください。用意されている DFSPZP00 モジュールを例としてお使いください。

残りの DRA モジュール群は、DFSPRRC0 によって動的に割り振られるロード・ライブラリーに置かれます。このロード・ライブラリーの DDNAME と DSNAME は、始動テーブルで指定されます。デフォルト DSNAME (IMS.SDFSRESL) には、すべての DRA コードが収められています。デフォルト始動テーブル DFSPZP00 には、これが指定されています。

関連資料:

 CCTL 領域のデータベース・リソース・アダプター始動テーブル (システム定義)

373 ページの『INIT 要求』

## 第 19 章 ODBA インターフェースに対する DRA の有効化

ODBA インターフェースで DRA を使用できるようにするには、4 つのステップを実行して、DRA を有効にする必要があります。最初のステップは、ODBA DRA 始動テーブルを作成することです。

ODBA インターフェースで DRA を使用できるようにするには、次の 4 つのステップからなる有効化手順を実行します。

1. ODBA DRA 始動テーブルを作成します。
2. ODBA と DRA のモジュール群が、z/OS 領域の STEPLIB または JOBLIB に収められていることを確認します。
3. ODBA アプリケーション・プログラムを DFSCDLI0 とリンクします。
4. セキュリティーを構成します。

ODBA アプリケーション・プログラムが CIMS INIT 要求、または CIMS CONNECT 要求 (ODBA アプリケーションが複数の IMS システムと接続する必要がある場合) を発行すると、ODBA インターフェースが初期設定処理を開始します。CIMS INIT 要求および CIMS CONNECT 要求は、アドレス・スペースに ODBA 環境を確立します。その後 APSB 要求は DRA を呼び出し、DFSxxxx0 に指定された IMS DB を使用して PSB スケジュール要求を処理します。ここで、xxxx は、APSB 呼び出しで AIB の AIBRSNM2 フィールドに指定された DRA 始動テーブルの名前です。

オプションの xxxx 値が CIMS INIT または CIMS CONNECT 呼び出しで渡された場合、CIMS INIT 要求または CIMS CONNECT 要求の処理後、DRA インターフェースが IMS DB に対して自己証明を行います。これで、他の要求を処理できるようになります。以下の図に、この時点での DRA の構造を示します。

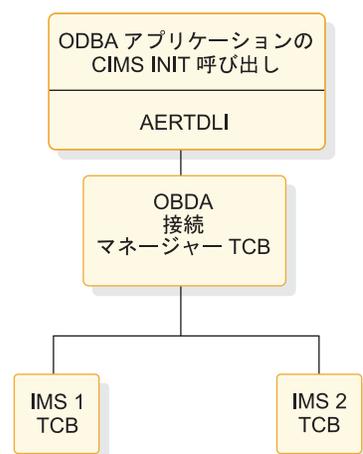


図 9. ODBA インターフェースでの DRA コンポーネント構造

残りの DRA モジュール群は、DFS AERA0 によって動的に割り振られるロード・ライブラリーに置かれます。このロード・ライブラリーの DDNAME と DSNAME

は、始動テーブルで指定されます。デフォルト DSNNAME (IMS.SDFSRESL) には、すべての DRA コードが取められています。

関連資料:

 CCTL 領域のデータベース・リソース・アダプター始動テーブル (システム定義)

---

## 第 20 章 CCTL DRA 要求の処理

CCTL が IMS DB と通信するには、DRA 要求を使用します。DRA 要求は、参加者のアダプター・パラメーター・リスト (PAPL) を使用して、CCTL から DRA に渡されます。DRA 要求を行うとき、CCTL はまず DRA 始動/ルーター・ルーチンの DFSPRRC0 に制御を渡し、レジスター 1 で PAPL を指し示します。

もちろん、DFSPRRC0 に制御を渡す前に、CCTL は PAPL を作成しておかなければなりません。要求の指定には PAPLFUNC フィールドを使い、そこに機能コードを指定します。

スレッド機能要求を指定するには、PAPLTFUN 値を PAPLFUNC フィールドに指定します。

機能要求は、さらにいくつもの副次機能に細分されます。スレッド機能要求は、その副次機能名で呼ばれます (例えば、スケジュール副次機能を持つスレッド要求は、SCHED 要求と呼ばれます)。非スレッド機能要求は、機能名で呼ばれます (例えば、初期設定要求は INIT 要求と呼ばれます)。

「DRA 要求」という用語は、スレッド機能要求と非スレッド機能要求の両方に使われます。

PAPL が作成され、DRA 始動/ルーター・ルーチンがロードされると、CCTL が DFSPRRC0 に制御を渡します。DFSPRRC0 に制御が渡った時点でのレジスター内容は、次のとおりです。

レジスター

内容

- 1 PAPL のアドレス
- 13 標準の 18 ワード保管域のアドレス
- 14 呼び出しルーチンのリターン・アドレス

DRA 始動/ルーター・ルーチンは、自身を 31 ビット・アドレッシング・モードに切り替えますが、呼び出しルーチンに戻るときは、すべてのレジスターを復元して、そのルーチンのアドレッシング・モードに戻します。レジスター 15 は、常にゼロのまま返されます。

要求の戻りコードは、PAPL の PAPLRETC フィールドに入っています。



---

## 第 21 章 ODBA 呼び出しの処理

ODBA アプリケーション・プログラムは、AERTDLI インターフェースを使用して IMS DB と通信します。AERTDLI 呼び出しインターフェースは ODBA アプリケーションからの DL/I 呼び出しを処理します。また、AIB を用いて、その呼び出し結果を ODBA に返します。

CCTL が PAPL を使用するのとは異なり、ODBA アプリケーション・プログラムは AERTDLI インターフェースを使用して IMS DB と通信します。AERTDLI 呼び出しインターフェースは ODBA アプリケーションからの DL/I 呼び出しを処理します。また、AIB を用いて、その呼び出し結果を ODBA に返します。

関連資料:

 ODBA アプリケーションの AIB マスクの指定 (アプリケーション・プログラミング)



---

## 第 22 章 CCTL が働きかける DRA 機能要求

CCTL には、DBCTL との通信に使用できる特定の要求があります。どの DRA 要求についても PAPL フィールドがあり、CCTL はそこに情報を書き込まなければなりません。PAPLUSER フィールドは、CCTL が任意に使用してよいフィールドです。例えば、出口ルーチンにデータを引き渡す目的に使用できます。

このトピックには汎用プログラミング・インターフェース情報が含まれています。

どの DRA 要求についても PAPL フィールドがあり、CCTL はそこに情報を書き込まなければなりません。DRA 要求が完了すると、こんどは DRA がいくつかの出力 PAPL フィールドに値を入れて返します。返されてくる PAPL には、もとの入力値をそのまま含んでいるフィールドもあります。

(PAPLTTOK フィールドと PAPLUSER フィールドには、もとの入力値が残っています。)

PAPLUSER フィールドは、CCTL が任意に使用してよいフィールドです。例えば、出口ルーチンにデータを引き渡す目的に使用できます。

DRA は、DRA 要求を処理した後、CCTL にコードを (PAPLRETC フィールド経由で) 返します。これは要求の状況を表すコードであり、IMS コード、DRA コード、z/OS コードのいずれかです。DRA が失敗したときは、PAPLRETC フィールドにゼロ以外の値が返されます。

デフォルトの中断出口ルーチンおよび再開出口ルーチンを使用するには、各 DRA 要求の PAPLTECB フィールドに、スレッドが待機または通知されるときに使用される CCTL ECB のアドレスを設定する必要があります。CCTL が中断出口ルーチンや再開出口ルーチンを提供しない場合、DRA のデフォルトの出口ルーチンが使用されます。

関連概念:

399 ページの『問題診断』

関連資料:



DBCTL 戻りコード (メッセージおよびコード)

---

### INIT 要求

DRA を初期化するには、INIT 要求を使用します。DRA 始動パラメーター・テーブルには、DRA 定義に必要なすべての必要パラメーターが含まれています。デフォルト・モジュール DFSPZP00 に用意されているパラメーターを使用するか、または独自のモジュールを作成し、それを IMS.SDFSRESL データ・セットにバインドすることができます。

INIT PAPL にも、DRA の初期化に必要なパラメーターがいくつか含まれています。INIT PAPL と DRA 始動パラメーターの両方に同じパラメーターが指定されているときは、INIT PAPL の指定が始動テーブルに優先します。

INIT PAPL の必須パラメーターの他に、以下のオプション・パラメーターがあります。

フィールド  
内容

#### **PAPLFUNC**

PAPLINIT

#### **PAPLSUSP**

延期出口ルーチンのアドレス

#### **PAPLRESM**

再開出口ルーチンのアドレス

#### **PAPLTECB**

デフォルトの中断出口ルーチンおよび再開出口ルーチンを使用するには、各 DRA 要求の PAPLTECB フィールドに、スレッドが待機または通知される  
ときに使用される CCTL ECB のアドレスを設定する必要があります。

CCTL が中断出口ルーチンや再開出口ルーチンを提供しない場合、DRA の  
デフォルトの出口ルーチンが使用されます。

#### **PAPLCNTL**

制御出口ルーチンのアドレス

#### **PAPLTSTX**

状況出口ルーチンのアドレス

INIT 要求と始動テーブルの処理が終わると、DRA は INIT PAPL 経由で次のデー  
タを CCTL に返します。

フィールド  
内容

#### **PAPLDBCT**

IMS DB ID (これはシステム定義からの IMSID パラメーターです)

#### **PAPLCTOK**

DRA に対して CCTL を明示するためのトークン

#### **PAPLTIMO**

DRA TERM 要求タイムアウト値 (秒数)

#### **PAPLRETC**

要求の状況を表す、CCTL に返されるコード

#### **PAPLDLEV**

DRA がどの機能をサポートしているかを CCTL に示すフラグ (PAPL マ  
ッピング形式の最新バージョンについては、IMS. ライブラリーを参照して  
ください。メンバー名は DFSPAPL です)

### **INIT 要求、DBCTL に対する自己証明**

DRA は、まず IMS DB に対して自己を証明し、IMS DB と CCTL の間にリンク  
を確立しないと、機能できません。この識別処理は、次の 2 つの場合に発生しま  
す。

- INIT 要求の直接的な結果として。

- 制御出口ルーチン呼び出しから要求された終了/自己再証明の一部として。

DRA は、最後の DRA 始動パラメーターで指定されている IMS DB サブシステムに対して自己を証明します。この識別処理は、INIT プロセスとは非同期的に実行されます。したがって、INIT 要求が正常に完了する一方で、識別処理が失敗に終わることもありえます。その場合は、制御出口ルーチンから CCTL に対し、IMS DB への接続に失敗したという通知が送られます。

IMS DB がアクティブでないと、コンソール・オペレーターに DFS690 メッセージが送られます (PAPLRETC フィールドにはコード 0 が返されます)。これに CANCEL または WAIT と応答してください。WAIT と応答すると、DRA は指定されている時間間隔だけ待ってから、再度、自己証明を試みます。待ち時間が必要なのは、DBCTL 再始動プロセスが完了しないと、識別処理の成功はないからです。待ち時間の長さは、TIMER DRA 始動パラメーターで指定します。自己証明を繰り返しても失敗が続くときは、コンソール・オペレーターにメッセージ DFS691、WAITING FOR IMS DB が送られます。

サブシステムが再始動完了状態に達せず、DRA が IMS DB に対して自己を証明できないときは、識別処理を終了させなければなりません。これには 2 通りの方法があります。

- 自己証明が失敗するたびに、制御出口ルーチンを呼ぶ。これで PAPL 戻りコードの 4 または 8 が設定され、識別処理が終了します。
- CCTL が TERM 要求を出す。

メッセージ DFS690 に CANCEL と応答すると、制御出口ルーチンに制御が渡り、DRA はそのルーチンの決定に従って行動します。

識別処理が正常に終了すると、DRA は CCTL アドレス・スペースをスワップ不能にし、制御出口ルーチンを呼んで、未確定 UOR のリストを渡します。未確定 UOR がまったくないときは、ヌル・リストが渡されます。未確定 UOR があれば、CCTL は RESYNC 要求でそれを解消します。

INIT 要求は、MINTHRD に指定された数だけのスレッド TCB を作成しようとしています。ストレージに余裕がないときは、実際に作成される TCB の数がそれより小さくなることもあります。

### 先行 DRA セッションの終了後における INIT 要求

先の DRA セッションが TERM 要求で終わり、それに対して PAPL 戻りコードの 0 が返されていたときは、この INIT 要求で PAPLCTOK=0 を指定しなければなりません。0 以外の PAPLCTOK を指定すると、INIT 要求が失敗します。

DRA セッションが次のいずれかの理由で終了した場合、INIT 要求で、前のセッションの PAPLCTOK 値を現行 PAPLCTOK フィールドに引き渡さなければなりません。

- TERM 要求からの戻りコードがゼロ以外。
- 制御出口ルーチン要求から内部 TERM 要求が出された。
- DRA の障害。

## **DRA オープン・スレッド TCB オプションを使用するための INIT 要求**

DRA オープン・スレッド TCB により、CCTL は DRA に対して専用 DRA スレッド TCB への接続をしないように指示できます。代わりに、DRA スレッド要求は CCTL アプリケーション TCB 上で処理されます。

DRA インスタンスが存続する間、DRA オープン・スレッド TCB オプションは、使用中、または使用中でないのいずれかです。

DRA オープン・スレッド TCB を活動化するように要求するには、INIT 要求で以下のフィールドを設定します。

### **PALPFNCL**

3 (PAPLFNC3)

### **PAPLOOTT**

オープン・スレッド TCB を活動化するには、このビット (X'08') をセットします。

INIT 呼び出し後に DRA が戻す INIT PAPL 内の CCTL にある PAPLDLEV フラグ・フィールドを調べることにより、TCB オープン・スレッドが DRA に対して使用中になっているかどうか検証することができます。PAPLOTCB (X'08') フラグが 1 にセットされていれば、オープン・スレッド TCB オプションが使用中です。

## **DRA オープン・スレッド TCB オプションを使用するための INIT 要求**

DRA オープン・スレッド処理が使用可能である場合、DRA は、DFSPZPxx メンバの DFSPRP マクロの TIMETHREADCPU= に指定された値に基づく DRA スレッド処理に関連した CPU 時間使用統計のモニターおよび報告を行いません。

DFSPZPxx メンバに TIMETHREADCPU=CLIENT 以外のものが指定されている場合、DRA は、INIT 呼び出しで指定されているものを無視します。

CPU 時間使用統計をモニターしないよう DRA に要求するには、INIT 要求で以下のフィールドを設定します。

### **PALPFNCL**

3 (PAPLFNC3)

### **PAPLOTCN**

DRA スレッド CPU 時間使用統計のモニタリングを使用不可にするためには、フィールド PAPLDROP にこのビット (X'20') を設定します。

DRA スレッド処理について CPU 使用統計がモニター対象であるかどうかを判別するには、INIT 呼び出し後に DRA が戻す CCTL INIT PAPL 内の PAPLDLEV フラグ・フィールドを調べます。PAPLOTCF (x'40') フラグが 1 に設定されている場合、CPU 使用統計のモニターおよび報告は行われません。

関連概念:

365 ページの『第 18 章 CCTL に対する DRA の有効化』

---

## RESYNC 要求

IMS DB に未確定 UOR への対処方法を指示するには、RESYNC 要求を使用します。4 つの異なる副次機能の値により、IMS DB が UOR に対して可能なアクションが示されます。

以下の副次機能の値で、可能なアクションを指示します。

### PAPLRCOM

未確定 UOR をコミットします。

### PAPLRABT

未確定 UOR を打ち切ります。リカバリー可能なリソースに対して加えられた変更は、バックアウトされます。

### PAPLSCLD

コールド・スタートが行われたため、Transaction Manager に対して UOR が失われました。

### PAPLSUNK

CCTL にとって不明の未確定 UOR です。CCTL の未確定期間にフェーズ 1 の起点が含まれていないと、この状況が起こります。( 357 ページの表 81 に未確定期間が図示されていますから、参照してください。)

PAPL の次の入力フィールドに値を指定してください。

フィールド

内容

### PAPLCTOK

要求トークン

このトークンは、DRA に対して CCTL を証明する働きをします。DRA はトークンを確認したのち、始動 INIT 要求のパラメーター・リストを通じてそれを CCTL に返します。要求トークンは、あらゆる RESYNC 要求で DRA に渡さなければなりません。

### PAPLR TOK

リカバリー・トークン

この 16 バイト・トークンは、UOR と関連付けられています。最初の 8 バイトは、Transaction Manager のサブシステム ID です。次の 8 バイトは、CCTL スレッドごとに一意でなければなりません。これは、制御出口ルーチンに引き渡される未確定リカバリー・トークンの 1 つです。

### PAPLFUNC

PAPLRSYN

### PAPLSFNC

このフィールドには、PAPLRCOM、PAPLRABT、PAPLSCLD、または PAPLSUNK を含める必要があります。

必須の入力パラメーターの他に、以下のオプションの入力パラメーターがあります。

フィールド

内容

### PAPLTECB

デフォルトの中断出口ルーチンおよび再開出口ルーチンを使用するには、各 DRA 要求の PAPLTECB フィールドに、スレッドが待機または通知される時に使用される CCTL ECB のアドレスを設定する必要があります。CCTL が中断出口ルーチンや再開出口ルーチンを提供しない場合、DRA のデフォルトの出口ルーチンが使用されます。

---

## TERM 要求

IMS DB/CCTL 接続を終了し、CCTL 環境から DRA を取り除くには、TERM 要求を使用します。まず、すべてのスレッドが解決され、その後、DRA が終了します。新しい DRA 要求やスレッド要求は認められず、現在処理中の要求は完了しなければなりません。

PAPL の次の入力フィールドに値を指定してください。

フィールド  
内容

### PAPLFUNC

PAPLTERM。DRA 終了機能コード

### PAPLCTOK

DRA 要求トークン (INIT 要求からの出力)

必須の入力パラメーターの他に、以下のオプションの入力パラメーターがあります。

フィールド  
内容

### PAPLTECB

デフォルトの中断出口ルーチンおよび再開出口ルーチンを使用するには、各 DRA 要求の PAPLTECB フィールドに、スレッドが待機または通知される時に使用される CCTL ECB のアドレスを設定する必要があります。CCTL が中断出口ルーチンや再開出口ルーチンを提供しない場合、DRA のデフォルトの出口ルーチンが使用されます。

TERM 要求の結果を受け取った後、CCTL が DFSPPRC0 を取り除く場合もあります。

PAPL を通じて CCTL に返されるフィールドは次のとおりです。

フィールド  
内容

### PAPLRETC

戻りコード

### PAPLMXNB

この DRA セッション中に最大スレッド数に達した回数

### PAPLMTNB

この DRA セッション中に最小スレッド数に達した回数

## PAPLHITH

この DRA セッション中にスケジュールされたスレッド TCB の最大数

## PAPLTIMX

この DRA セッション中、最大スレッド数のまま経過した時間

---

## SCHED 要求

IMS DB 内で PSB をスケジュールするには、SCHED 要求を使用します。CCTL スレッドが出す最初の SCHED 要求では、新しい DRA スレッドが必要とされます。既存の DRA スレッド TCB が、現在 DRA スレッドを処理していない場合は、それが使用されます。

使用できる TCB がないと、DRA は新しいスレッド TCB を作成するか (INIT 要求の MAXTHRD パラメーターで指定されたスレッドの最大数に達するまで)、使用できるスレッドが現れるまで SCHED 要求を待機させます。

PAPLWCMD フィールドの値は、SCHED 要求の対象スレッドが短スレッドか長スレッドかを示します。どちらのスレッドであるかにより、そのスレッドでスケジュールされているデータベースに対してデータベース・コマンドが出されたとき、IMS のとるアクションが異なります。短スレッドでスケジュールされているデータベースに対して /STOP DATABASE、/DBDUMP DATABASE、/DBRECOVERY DATABASE のいずれかのコマンドが出された場合、そのコマンドは、そのデータベースがスケジュールから解放されるまで待ちます。一方、データベースが長スレッドでスケジュールされていると、IMS はこれらのコマンドを拒否します。

PAPL の次の入力フィールドに値を指定してください。

フィールド  
内容

## PAPLFUNC

PAPLTFUN。スレッド機能コード

## PAPLSFNC

PAPLSCHE。スケジュール要求の副次機能コード

## PAPLCTOK

DRA 要求トークン (INIT 要求からの出力)

## PAPLTTOK

CCTL によってセットアップされたスレッド・トークン

## PAPLRTOK

16 バイトの UOR トークン (RTOKEN)。

最初の 8 バイトには、CCTL ID が入っています。この ID は、INIT 要求の USERID または PAPLUSID から確定された最後の DRA 始動パラメーター CCTL ID と同じものです。USERID パラメーターは、DFSPZPxx モジュールを生成するために使用される DFSPRP マクロに入っています。次の 8 バイトには、UOR ごとに CCTL によって指定された固有 ID が含まれています。

## PAPLPSB

PSB 名

## PAPLWRTH

デッドロック加重値

このスレッドが他の DRA スレッドまたはいずれかの IMS 領域とデッドロック状態になったとき、DBCTL はデッドロック加重値が小さいほうのスレッドを消滅させます。

## PAPLWCMD

このビットは、スレッドを短スレッドと長スレッドのいずれかに定めます。どちらであるかにより、そのスレッドでスケジュールされているデータベースに対して /STOP DATABASE、/DBDUMP DATABASE、/DBRECOVERY DATABASE のいずれかのコマンドが出されたとき、IMS のとるアクションが異なります。

このビットがオン (X'80') の場合はデータベースは短スレッドでスケジュールされ、オフの場合は長スレッドでスケジュールされます。

## PAPLFTRD

高速機能トレース・オプション

このビットがオン (X'40') の場合、IMS DB の高速機能トレースがアクティブになります。

## PAPLKEYP

公開鍵オプション

このビットが設定されている (X'10') 場合、DBCTL は特殊サブプール内に UPSTOR 域を作成して、公開鍵で実行されるアプリケーションが UPSTOR 域を取り出せるようにします。

## PAPLLKGV

最大ロック・オプション

このビットがオン (X'08') の場合、DBCTL は PAPLLKMX の値を、UOR が保持できるロックの最大数と見なします。最大値を超えると、U3301 異常終了が起きます。

## PAPLLKMX

最大ロック値。0 から 255

この値は、SCHED 要求で参照されている PSB に対して PSBGEN で指定されている LOCKMAX パラメーターに優先します。

## PALPUFXT

DRA オープン・スレッド標識

PAPLUFXT を SCHED 呼び出しの場合に 1 にセットして、スレッド TCB がスレッド存続中に同一の TCB であるとは限らないことを、DRA に対して示します。TCB は、PSB がスケジュールされる時点と、PSB がスケジュールから解放されてスレッドが強制終了される時点の間に、変わる可能性があります。

## PAPLALAN

アプリケーション言語タイプ

次の入力フィールドへの指定は任意です。

フィールド  
内容

#### **PAPLSTAT**

スケジュール統計データを CCTL に返すときに使用されるエリアのアドレス

アドレスを割り振りたくない場合は、0 を入力します。

#### **PAPLPBTK**

CCTL が取得した z/OS ワークロード・マネージャー・パフォーマンス・ブロックのトークンのアドレス

DRA スレッドを z/OS ワークロード・マネージャーでサポートしたければ、このフィールドにアドレスを指定しなければなりません。

#### **PAPLTECB**

デフォルトの中断出口ルーチンおよび再開出口ルーチンを使用するには、各 DRA 要求の PAPLTECB フィールドに、スレッドが待機または通知されるときに使用される CCTL ECB のアドレスを設定する必要があります。CCTL が中断出口ルーチンや再開出口ルーチンを提供しない場合、DRA のデフォルトの出口ルーチンが使用されます。

PAPL を通じて CCTL に返される出力フィールドは次のとおりです。

フィールド  
内容

#### **PAPLRETC**

戻りコード

#### **PAPLCTK2**

スレッド要求トークン番号 2。このスレッドから発する将来の DRA 要求で必要となる DRA トークンです。

#### **PAPLPCBL**

PCB リストのアドレス。スケジュールされた PSB に存在するどの PCB についても、IMS DB で使用されるかどうかにかかわらず、1 項目ずつが必要です。

DFSPRP マクロに PCBLOC=31 が設定されているか、INIT 要求に PAPLLPSO=31 が指定されている場合、PCB リストのアドレスが 16 MB 境界より上になります。

#### **PAPL1PCB**

最初のデータベース PCB を指し示している PCBLIST 項目のアドレス

#### **PAPLIOSZ**

最大入出力域のサイズ

#### **PAPLPLAN**

PSB の言語種

#### **PAPLMKEY**

最大キー長

## PAPLSTAT

スケジュール統計データ域のアドレス。このアドレスは、入力フィールドに指定されていなければなりません。

現在 IMS データベース・マネージャーを使用している CCTL にとっては、DBCTL にマイグレーションすると、スケジュール要求にある PCBLIST とユーザー PCB 域が変化します。PCBLIST の最初の PCB ポインターには、入出力 PCB のアドレスが含まれています。この入出力 PCB は、DBCTL 環境におけるスケジュール・プロセスで内部的に割り振られたものです。入出力 PCB は、普通、出力メッセージに使用されるか、制御型機能の処理を要求する目的に使用されます。PCBLIST と PCB は、UPSTOR と呼ばれる 1 つの連続域に収められています。PSB の生成に LANG=PLI が使用されていれば、PCBLIST は PCB へのポインターを指し示します。LANG=PLI でないときは、PCB を直接指し示します。

関連概念:

355 ページの『第 16 章 同期点』

398 ページの『DRA トレース』

395 ページの『第 25 章 CCTL のパフォーマンス: DRA スレッド TCB のモニター』

---

## IMS 要求

現在スケジュールされている PSB に対して IMS または高速機能データベース要求を行うには、IMS 要求を使用できます。

PAPL の次の入力フィールドに値を指定してください。

フィールド  
内容

### PAPLFUNC

PAPLTFUN

### PAPLSFNC

PAPLDLI。DL1 要求の副次機能コード

### PAPLCTOK

DRA 要求トークン (INIT 要求からの出力)

### PAPLCTK2

スレッド・トークン番号 2。これは、SCHED 要求からの出力の一部になっている DRA 要求トークンです。

### PAPLTTOK

CCTL によってセットアップされたスレッド・トークン

### PAPLR TOK

RTOKEN

16 バイト UOR トークン。

### PAPLCLST

IMS 呼び出しリストのアドレス。

## PAPLALAN

アプリケーション言語タイプ。これは、呼び出しリストがどうセットアップされているかを反映していなければなりません。 PAPLALAN='PLI' の場合、DRA は、PCB のポインターへのポインターが呼び出しリストに含まれているものと想定します。他のプログラミング言語なら、DRA は直接ポインターを想定します。

PAPLALAN は、要求からの戻りをスケジュールする PAPLPLAN と一致していなくてかまいません。例えば、PAPLPLAN=PLI なら、UPSTOR の PCBLIST は間接リストを指し示します。指定した場合、CCTL はこれを利用してアプリケーション・プログラムが使用する PCBLIST を作成することができます。例えば、アプリケーション・プログラムが COBOL で書かれているとすれば、CCTL はポインターなしの新しい PCBLIST を作成し、その新しいリストに UPSTOR にある PCB を指し示させることができます。アプリケーション・プログラムの IMS 呼び出しリストで PAPLALAN=COBOL を指定しておけば、DRA が呼び出しリスト中にポインターを求めることはありません。

必須の入力パラメーターの他に、以下のオプションの入力パラメーターがあります。

フィールド

内容

## PAPLTECB

デフォルトの中断出口ルーチンおよび再開出口ルーチンを使用するには、各 DRA 要求の PAPLTECB フィールドに、スレッドが待機または通知される時に使用される CCTL ECB のアドレスを設定する必要があります。CCTL が中断出口ルーチンや再開出口ルーチンを提供しない場合、DRA のデフォルトの出口ルーチンが使用されます。

PAPL を通じて CCTL に返される出力フィールドは次のとおりです。

フィールド

内容

## PAPLRETC

返されたコード

## PAPLSEGL

返されたデータの長さ

関連概念:

355 ページの『第 16 章 同期点』

関連資料:

 プログラム連絡ブロック (PCB) リスト (アプリケーション・プログラミング)

---

## SYNTERM 要求

単一フェーズの同期点要求により、UOR をコミットするか、または PSB を解放するには、SYNTERM 要求を使用できます。

PAPL の次の入力フィールドに値を指定してください。

フィールド  
内容

**PAPLFUNC**

PAPLTFUN

**PAPLSFNC**

PAPLSTRM。同期点コミット/終了の副次機能コード

**PAPLCTOK**

DRA 要求トークン (INIT 要求からの出力)

**PAPLCTK2**

スレッド要求トークン #2。この DRA トークンは、SCHED 要求からの出力です。

**PAPLTTOK**

CCTL によってセットアップされたスレッド・トークン

**PAPLR TOK**

16 バイトの UOR トークン (RTOKEN)。

次の入力フィールドを指定できます。この指定は任意です。

フィールド  
内容

**PAPLSTAT**

トランザクション統計データを CCTL に返すときに使用されるエリアのアドレス

**PAPLTECB**

デフォルトの中断出口ルーチンおよび再開出口ルーチンを使用するには、各 DRA 要求の PAPLTECB フィールドに、スレッドが待機または通知されるときに使用される CCTL ECB のアドレスを設定する必要があります。CCTL が中断出口ルーチンや再開出口ルーチンを提供しない場合、DRA のデフォルトの出口ルーチンが使用されます。

PAPL を通じて CCTL に返される出力フィールドは次のとおりです。

フィールド  
内容

**PAPLRETC**

返されたコード

**PAPLSSCC**

スレッドの障害発生時における単一フェーズ同期点要求の状態。このフィールドは、PAPLRETC がゼロでないときにセットされます。

**PAPLSTAT**

トランザクション統計データ域のアドレス。このアドレスは、入力フィールドに指定されていなければなりません。

関連概念:

355 ページの『第 16 章 同期点』

---

## PREP 要求

フェーズ 1 の同期点要求により、IMS DB に、UOR をコミットする準備ができて  
いるかどうかを尋ねるには、PREP 要求を使用できます。

PAPL の次の入力フィールドに値を指定してください。

フィールド  
内容

**PAPLFUNC**  
PAPLTFUN

**PAPLSFNC**  
PAPLPPREP。同期点準備の副次機能コード

**PAPLCTOK**  
DRA 要求トークン (INIT 要求からの出力)

**PAPLCTK2**  
スレッド・トークン #2。これは、SCHED 要求から出力される DRA 要求  
トークンです。

**PAPLTTOK**  
CCTL によってセットアップされたスレッド・トークン

**PAPLR TOK**  
16 バイトの UOR トークン (RTOKEN)。

**PAPLSDPL**  
1 ビットのフラグ。このビットを 1 にセットして、このスレッドが  
DUOW (分散作業単位) の一部であることを IMS に対して示します。

必須の入力パラメーターの他に、以下のオプションの入力パラメーターがありま  
す。

フィールド  
内容

**PAPLTECB**  
デフォルトの中断出口ルーチンおよび再開出口ルーチンを使用するには、各  
DRA 要求の PAPLTECB フィールドに、スレッドが待機または通知される  
ときに使用される CCTL ECB のアドレスを設定する必要があります。  
CCTL が中断出口ルーチンや再開出口ルーチンを提供しない場合、DRA の  
デフォルトの出口ルーチンが使用されます。

PAPL には以下の出力フィールドが返されます。

フィールド  
内容

**PAPLRETC**  
返されたコード

**PAPLSTCD**  
高速機能状況コード

PAPLRETC フィールドの値が 10 進値の 35 なら、PAPLSTCD フィールドには、エラーをさらに詳しく記述する状況コードが入っています。

関連概念:

355 ページの『第 16 章 同期点』

---

## COMTERM 要求

フェーズ 2 同期点要求により、UOR をコミットするか、または PSB を解放するには、COMTERM 要求を使用できます。COMTERM 要求を出す前には、必ず PREP 要求を出さなければなりません。

PAPL の次の入力フィールドに値を指定してください。

フィールド  
内容

**PAPLFUNC**  
PAPLTFUN

**PAPLSFNC**  
PAPLCTRM。同期点コミット/終了の副次機能コード

**PAPLCTOK**  
DRA 要求トークン (INIT 要求からの出力)

**PAPLCTK2**  
スレッド・トークン番号 2。これは、SCHED 要求から出力される DRA 要求トークンです。

**PAPLTTOK**  
CCTL によってセットアップされたスレッド・トークン

**PAPLRTOK**  
16 バイトの UOR トークン (RTOKEN)。

次の入力フィールドへの指定は任意です。

フィールド  
内容

**PAPLSTAT**  
トランザクション統計データを CCTL に返すときに使用されるエリアのアドレス

**PAPLTECB**  
デフォルトの中断出口ルーチンおよび再開出口ルーチンを使用するには、各 DRA 要求の PAPLTECB フィールドに、スレッドが待機または通知されるときに使用される CCTL ECB のアドレスを設定する必要があります。CCTL が中断出口ルーチンや再開出口ルーチンを提供しない場合、DRA のデフォルトの出口ルーチンが使用されます。

PAPL を通じて CCTL に返される出力フィールドは次のとおりです。

フィールド  
内容

## PAPLRETC

返されたコード

## PAPLSTAT

トランザクション統計データ域のアドレス。このアドレスは、入力フィールドに指定されていなければなりません。

関連概念:

355 ページの『第 16 章 同期点』

---

## ABTTERM 要求

フェーズ 2 の同期点要求により、処理を中止して PSB を解放するには、ABTTERM 要求を使用できます。ABTTERM 要求の前には PREP 要求を出す必要がありません。

PAPL の次の入力フィールドに値を指定してください。

フィールド  
内容

### PAPLFUNC

PAPLTFUN

### PAPLSFNC

PAPLATRM。同期点コミット/終了の副次機能コード

### PAPLCTOK

DRA 要求トークン (INIT 要求からの出力)

### PAPLCTK2

スレッド・トークン番号 2。これは、SCHED 要求から出力される DRA 要求トークンです。

### PAPLTTOK

CCTL によってセットアップされたスレッド・トークン

### PAPLRTOK

16 バイトの UOR トークン (RTOKEN)。

次の入力フィールドへの指定は任意です。

フィールド  
内容

### PAPLSTAT

トランザクション統計データを CCTL に返すときに使用されるエリアのアドレス

### PAPLTECB

デフォルトの中断出口ルーチンおよび再開出口ルーチンを使用するには、各 DRA 要求の PAPLTECB フィールドに、スレッドが待機または通知されるときに使用される CCTL ECB のアドレスを設定する必要があります。CCTL が中断出口ルーチンや再開出口ルーチンを提供しない場合、DRA のデフォルトの出口ルーチンが使用されます。

PAPL を通じて CCTL に返される出力フィールドは次のとおりです。

フィールド  
内容

**PAPLRETC**

返されたコード

**PAPLSTAT**

トランザクション統計データ域のアドレス。このアドレスは、入力フィールドに指定されていなければなりません。

関連概念:

355 ページの『第 16 章 同期点』

---

## TERMTHRD 要求

DRA スレッドを終了するには、TERMTHRD 要求を使用します。

PAPL の次の入力フィールドに値を指定してください。

フィールド  
内容

**PAPLFUNC**

PAPLTFUN

**PAPLSFNC**

PAPLTTHD。スレッド終了の副次機能コード

**PAPLCTOK**

DRA 要求トークン (INIT 要求からの出力)

**PAPLCTK2**

スレッド・トークン番号 2。これは、SCHED 要求から出力される DRA 要求トークンです。

**PAPLTTOK**

CCTL によってセットアップされたスレッド・トークン

次の入力フィールドへの指定は任意です。

フィールド  
内容

**PAPLSTAT**

トランザクション統計データを CCTL に返すときに使用されるエリアのアドレス

**PAPLTECB**

デフォルトの中断出口ルーチンおよび再開出口ルーチンを使用するには、各 DRA 要求の PAPLTECB フィールドに、スレッドが待機または通知されるときに使用される CCTL ECB のアドレスを設定する必要があります。CCTL が中断出口ルーチンや再開出口ルーチンを提供しない場合、DRA のデフォルトの出口ルーチンが使用されます。

PAPL を通じて CCTL に返される出力フィールドは次のとおりです。

フィールド  
内容

**PAPLRETC**

返されたコード

**PAPLSTAT**

トランザクション統計データ域のアドレス。このアドレスは、入力フィールドに指定されていなければなりません。



---

## 第 23 章 DRA の終了

終了の分離とは、IMS DB サブシステムでの障害発生が、ATTACH されている CCTL サブシステムまたは ODBA アプリケーションの障害に直結しないこと、また、その逆が起こらないことを意味します。

CCTL サブシステムまたは ODBA アプリケーションを設計するときは、終了の分離を設計の主目標の 1 つにしなければなりません。

IMS DB は、接続しているサブシステム間の障害の伝染を阻止するように設計されていますが、CCTL サブシステムが終了すると、それが IMS DB に障害を引き起こすことがあります。IMS DB が CCTL のためにスレッド DL/I 呼び出しを処理しているときに、DRA スレッド TCB が終了すると、IMS DB で U0113 異常終了が起こります。

IMS DB が DL/I 呼び出しを処理しているとき、次の状態が発生するとスレッド TCB が終了します。

- コードの不具合による DRA スレッドの異常終了。これはコードの不具合を直すことで修正できます。
- スレッド TCB がまだ存在しているのに、CCTL TCB が消滅した。S13E または S33E の異常終了となり、スレッド TCB が消滅します。この原因としては、CCTL の異常終了、取り消しコマンド、シャットダウンの 3 通りが考えられます。
- CCTL が出す IMS DB /STOP REGION CANCEL コマンドに起因する DRA スレッドの異常終了。

IMS DB U0113 異常終了は、CCTL リカバリー処理の設計を工夫することで阻止できます。TERM 要求を出し、この要求の完了まで待つようにしてください。これで、CCTL TCB より先に DRA とスレッド TCB が終了します。

関連概念:

393 ページの『第 24 章 CCTL リカバリー処理の設計』



## 第 24 章 CCTL リカバリー処理の設計

CCTL を設計する際には、CCTL の操作およびインストール要件を考慮しなければなりません。例えば、CCTL に自己シャットダウンの手段を持たせることはできますが、CCTL スレッドまたは BMP に実行が長時間にわたる UOR を持たせるべきではありません。

回復不能な z/OS の異常終了という状況では、DRA TERM 要求を出してください。これですべてのスレッドが消滅し、U0113 が可能となります。CCTL のリカバリー不能異常終了の数を減らすために、IMS DB は、IMS DB に接続している CCTL のオペレーター CANCEL をインターセプトし、それを CCTL の S08E リカバリー可能異常終了に変換します。CANCEL を S08E 異常終了に変換する場合は、IMS 始動パラメーターに CCTCVCAN=Y を指定する必要があります。

最後の手段として、CCTL の強制シャットダウンという選択肢もあります。CANCEL コマンドの入力後 (そして、S08E に変換された後)、オペレーターが FORCE コマンドを入力すると、IMS DB はその FORCE を z/OS の取り消しコマンドに変換します。それ以後の FORCE の試みは、IMS DB によってインターセプトされません。これらの回復不能な異常終了の場合は、U0113 が可能です。

CCTL に自己シャットダウンの手段を持たせることもできます。CCTL シャットダウン論理に DRA TERM 要求を出させ、要求の完了を待たせれば、IMS DB での U0113 異常終了を阻止できます。DRA TERM 要求は、現在のスレッド要求が完了するのを待ちます。現在のスレッド DL/I 呼び出しが正常に完了するのを妨げる事態が起こるとすれば、それは、データベース・セグメントが使用可能になるのを、呼び出しが IMS DB の中で待たなければならない場合です。セグメントが使用可能でない理由として考えられるのは、別の UOR によって拘束されていることです。それは、別の CCTL に属するスレッドの UOR かもしれませんし、IMS とは従属領域 (例えば、BMP) にある UOR かもしれません。解決策は、実行が長時間続くような CCTL スレッドや BMP を作らないことです。

**推奨事項:** BMP では頻繁にチェックポイントを設けてください。

どのような方法で長時間 CCTL スレッドをやめさせ、減らそうとするにしても、DRA TERM 要求の完了をどれだけの時間待つか (TIMEOUT) を決めておくことが必要です。ほとんどの場合、CCTL 終了の途中で IMS DB に U113 異常終了が起こることは望ましくありません。したがって、タイムアウト値は、考えられる最長の UOR より長く設定しておかなければなりません。CCTL に UOR 時間を制限する手段、もしくはこの時間制限を指定する手段があれば、DRA TERM タイムアウト値を指定できます。このタイムアウト値は、DRA 始動テーブルで指定でき、INIT PAPL を通じて CCTL に返されてきます。

**推奨事項:** DRA TERM 要求の完了を待つときは、CCTL でこの DRA TERM タイムアウト値を使用してください。DRA TERM タイムアウト値を使用すれば、少なくとも、ある IMS DB/CCTL セッションで実行されるアプリケーションの UOR 時間長に関連して、CCTL の終了が IMS DB の障害をもたらすかを制御できます。

CCTL オペレーションの指針：

- IMS DB に接続している CCTL 領域に対して CANCEL コマンドまたは FORCE コマンドを出すことを避けてください。

CCTL 設計の指針：

- リカバリー可能異常終了の処理の中で、CCTL から DRA TERM 要求を出してください。
- CCTL シャットダウン機能で DRA TERM 要求を出してください。
- DRA TERM 要求を出すときは、それが完了するのを待ってください。このタイムアウトに上限を設けるときは、DRA 始動テーブルに指定されている TIMEOUT 値を使用してください。
- CCTL では、IMS DB を使用するスレッドの中で実行が長時間にわたる UOR を避けてください。

ユーザー・インストールの推奨事項：

- BMP では頻繁にチェックポイントを設けてください。
- 実行が長時間にわたる UOR アプリケーションを制限してください。
- TIMEOUT 始動パラメーターをできるだけ大きく設定してください。できれば、最長の UOR より長くします。

関連概念:

391 ページの『第 23 章 DRA の終了』

---

## 第 25 章 CCTL のパフォーマンス: DRA スレッド TCB のモニター

DRA スレッド TCB の状況は、ほとんどの場合、/DISPLAY CCTL ALL コマンドの出力から評価できます。スレッド障害がまったく発生していないと、一時的な内部状態のために、出力に MINTHRD 値より小さなスレッド TCB 数が示されることがあります。

要件: DRA 初期設定処理には、DRA スレッド TCB の最小値と最大値 (MINTHRD と MAXTHRD) が必要です。MINTHRD と MAXTHRD にどのような値を指定するかによって、並行的に進められるマルチスレッド実行の数が決まります。また、DRA がスレッド障害のない正常な状態で維持するスレッド TCB の数の範囲も、これらの値によって決まります。次のスレッド障害が発生すると、TCB の数が MINTHRD 値を下回ることがあります。

- 異常終了。
- DRA スレッド要求からゼロ以外の戻りコード。スレッド TCB が消滅します。
- IMS DB /STOP REGION コマンドを使用しての終了。

不首尾だったスレッド TCB は、自動的に再作成されません。SCHED 要求を処理するために新しいスレッドを作成すると、スレッド TCB 番号が再び大きくなります。スレッド TCB の番号が MINTHRD 値より大きくなり、すべてのスレッド・アクティビティーが正常に終了すると、DRA には MINTHRDD 値に等しい数のスレッド TCB が残されます。

CCTL 処理の際、スレッド TCB を占拠するアクティブな DRA スレッドの数は、0 から MAXTHRD 値まで変動します。アクティブな DRA スレッドとは、少なくとも 1 個の SCHED 要求を行っていて、まだ TERMTHRD 要求を行っていないことを意味します。アクティブでないスレッド TCB の数が大きくなりすぎると、DRA は自動的にいくつかのスレッド TCB を消滅させ、IMS DB リソースを解放します。

DRA スレッド TCB の状況は、/DISPLAY CCTL ALL コマンドの出力から確認できますが、次の例外があります。

スレッド障害がまったく発生していないと、内部で発生する瞬間的な状態のため、出力に MINTHRD 値より小さなスレッド TCB 数が示されることがあります。実際には、スレッド TCB の数が MINTHRD 値と等しくなっています。

関連概念:

 z/OS: STIMER マクロの説明

関連資料:

379 ページの『SCHED 要求』

 /DISPLAY CCTL コマンド (コマンド)

## DRA スレッド統計

SCHEM 要求のほか、UOR を終了させる DRA 要求では、DRA スレッド統計が返されてきます。この統計値は、PAPLSTAT フィールドで指し示される CCTL 域に置かれています。

PAPL リストは、以下の表に示す領域をマッピングしています。これらの統計値は、IMS DB ログ・レコード X'08' (SCHEM)と X'07' (UOR 終了) にも記録されます。

表 82. スケジュール・プロセスで提供される情報

PAPL フィールド	フィールド長 (16 進)	内容
PAPLNPSB	8	PSB 名
PAPLPOOL	8	経過したプール・スペース待ち時間 (パック : マイクロ秒)
PAPLINTC	8	経過した待ち時間 - 意図の対立 (パック : マイクロ秒)
PAPLSCHT	8	スケジュール・プロセスの経過時間 (パック : マイクロ秒)
PAPLTIMO	8	DB 入出力の経過時間 (パック : マイクロ秒)
PAPLTLOC	8	DI ロックの経過時間 (パック : マイクロ秒)
PAPLDBIO	4	DB 入出力の数

表 83. UOR 終了時に提供される情報

PAPL フィールド	フィールド長 (16 進)	内容
PAPLGU1	4	出されたデータベース GU 呼び出しの数
PAPLGN	4	出されたデータベース GN 呼び出しの数
PAPLGNP	4	出されたデータベース GNP 呼び出しの数
PAPLGHU	4	出されたデータベース GHU 呼び出しの数
PAPLGHN	4	出されたデータベース GHN 呼び出しの数
PAPLGHNP	4	出されたデータベース GHNP 呼び出しの数
PAPLISRT	4	出されたデータベース ISRT 呼び出しの数
PAPLDLET	4	出されたデータベース DLET 呼び出しの数
PAPLREPL	4	出されたデータベース REPL 呼び出しの数
PAPLTOTC	4	DL/I データベース呼び出しの総数
PAPLTENQ	4	テスト・エンキューの数
PAPLWTEQ	4	テスト・エンキューでの WAIT の数
PAPLTSQ	4	テスト・デキューの数
PAPLUENQ	4	更新エンキューの数
PAPLWUEQ	4	更新とエンキューでの WAIT の数
PAPLUPDQ	4	更新デキューの数
PAPLESEQ	4	排他的エンキューの数
PAPLWEXQ	4	排他的エンキューでの WAIT の数

表 83. UOR 終了時に提供される情報 (続き)

PAPL フィールド	フィールド長 (16 進)	内容
PAPLEXDQ	4	排他的デキューの数
PAPLDATS	8	STCK タイム・スケジュールの開始
PAPLDATN	8	STCK タイム・スケジュールの完了
PAPLDECL	2	DEDB 呼び出しの数
PAPLDERD	2	DEDB 読み取り操作の数
PAPLMSCL	2	高速機能用に予約
PAPLOVFN	2	使用されたオーバーフロー・バッファースの数
PAPLUOWC	2	UOW 競合の数
PAPLBFWT	2	DEDB バッファース WAIT の数
PAPLUSSN	4	一意のスケジュール・シーケンス番号
PAPLCTM1	4	UOR の経過 CPU 時間 (スレッド TCB)

## DRA 統計

DRA 統計は、DRA TERM 要求の結果として返されてくる PAPL に含まれています。あるいは、DRA 終了時制御出口ルーチンが呼ばれたときは、同ルーチンの PAPL に含まれています。このルーチンが呼ばれるのは、DRA で障害が発生したか、先の制御出口ルーチン呼び出しで戻りコード 4 が返されてきた場合です。

返されてくる PAPL に含まれている DRA 統計値は、次のとおりです。

1. MAXTHRD 値に達した回数
2. MINTHRD 値に達した回数 (スレッド TCB の数が減少に向かい、その結果、この値に達した回数だけが含まれます)
3. この DRA セッション中に到達したスレッド TCB の最大数 (これは TCB の数であって、DRA スレッドの数ではありません。したがって、少なくとも最小スレッド値以上です)
4. DRA スレッド TCB の数が MAXTHRD 値を維持していた時間 (秒数)

以前の統計値のフィールド名は、TERM PAPL と制御出口ルーチン PAPL の PAPL エクステンションに見つかります。

DRA パフォーマンス統計値を評価するにあたっては、次の点に注意してください。

- DRA がすでに最大数のスレッド (MAXTHRD) を使用している場合、そこへ新しい SCHED 要求が出されると、DRA は使用可能なスレッドが現れるまでその要求を待機させます。
- アクティブなスレッドが使用可能になるにつれ (例えば、TERMTHRD 呼び出しの結果)、使用可能なスレッドのなかには消滅していくものもあります。

これらの要因は、パフォーマンスには悪影響を与えることがありますが、DRA スレッドが少なくなれば IMS DB リソースが少なくてすむので、IMS DB のリソース可用性は向上します。つまり、他の BMP や他の CCTL にとっては、使用できる IMS DB リソース (PST) が増えることを意味します。

統計値 1、2、4 は、2 つの要因の尺度として活用でき、パフォーマンスとリソース使用量の間のバランスを取る方法を決めるときに役立ちます。これらの統計値を論じるときは、パフォーマンスの観点からのみ論じていることに注意してください (例えば、IMS DB にただ 1 つの CCTL だけが接続していると見なしています)。

## DRA 統計の評価

統計値 1 と 4 が大きいということは、使用可能なスレッドが現れるのを SCHED 要求が何度も待たなければならなかったことを意味します。パフォーマンスを向上させるには、MAXTHRD 値を大きくしてください。

統計値 2 がパフォーマンスに与える影響は、スレッドの活動歴がわからないと評価できません (この活動歴は DRA から得られませんが、CCTL から得られます)。アクティビティーのレベルが安定していれば、スレッドの消滅はほとんど起こらず、統計値 2 は無意味です。アクティビティーのレベルが大きく変動していれば、統計値 2 は大いに参考になります。

- 統計値 2 が 0 なら、スレッドの消滅が起こったかもしれませんが、MINTHRD 値に達したことはないことを意味します。
- 統計値 2 がゼロ以外なら、スレッド数が MINTHRD 値に達したことがあり、その時点でスレッドの消滅が停止され、パフォーマンスが強化されたことを意味します。したがって、スレッド・アクティビティーに大きな変動が見られるときは、統計値 2 がゼロ以外の値になるまで MINTHRD を大きくすれば、パフォーマンスを向上させられます。

最後に、統計値 3 は、MAXTHRD 値の調整に役立ちます。

注: これらの統計値は、MINTHRD と MAXTHRD の定義の決定に役立ちます。MINTHRD=MAXTHRD では、統計値の意味がなくなります。

---

## DRA トレース

アクティビティーのトレース (ロギング) は、DRA では行われません。ただし、IMS DB では、DL/I と高速機能がトレースされます。IMS DB で DL/I トレースをセットアップし、呼び出す方法は、IMS の場合と同じです。CCTL スレッドの出力トレース・レコードには、リカバリー・トークンが含まれています。

IMS DB での高速機能トレースは、IMS でのそれと異なります。IMS DB では、DRA に対する SCHED 要求で PAPLFTRD を ON (この UOR では高速機能トレースが必要) に設定しておくこと、高速機能トレースがアクティブになります。

この UOR が完了すると、トレース出力ファイルが閉じられ、SYSOUT クラス A に送られます。

高速機能処理の途中でスレッドに障害が発生した場合、DRA から返される PAPL で、PAPLFTRR フィールドが ON になっていることがあります。これは、この (不首尾だった) トランザクションを CCTL が再度実行する場合、SCHED PAPL の PAPLFTRD フィールドを ON にしてほしい (高速機能トレースが必要) という、CCTL への要望を表します。

関連資料:

379 ページの『SCHED 要求』

---

## IMS DB へのコマンド送信

IMS DB ウォーム・スタンバイや IMS XRF 環境で、IMS 代替システムを主 IMS システムに切り替えるには、CCTL が z/OS SVC 34 を使用して、緊急時再始動コマンドを IMS DB 代替システムにブロードキャストするか、または SWITCH コマンドを IMS XRF 代替システムにブロードキャストします。

このインターフェースを使用して実行できる IMS コマンドは、これ以外にありません。コマンド verb の前には、コマンド認識文字か、または 4 文字の IMS ID を付けます。IMS ID は、INIT PAPL の PAPLDBCT フィールドにあります。

---

## 問題診断

DRA は、SDUMP または SNAP データ・セット出力という形で診断情報を提供します。X'80' では、最初に SDUMP が試みられます。それに失敗すると、SNAP が実行されます。

失敗に終わった DRA 要求では、CCTL に返されてくる PAPL の PAPLRETC フィールドにゼロ以外の値が入っています。PAPLRETC の形式は次のとおりです。

HHSSUUU

ここで: HH= X'00'- 出力なし

UUU IMS DB 戻りコード

X'88'- 出力なし

SSS すべての z/OS やり直し不能異常終了コード (例えば、222、13E) あるいは、

UUU IMS 異常終了コード (775、777、844、849、2478、2479、3303)

X'84'- SNAP のみ

UUU IMS 異常終了コード (260、261、263)

X'80'- SDUMP/SNAP あり

SSS 再試行可能なすべての z/OS 異常終了コード

UUU PAPLRETC の形式にリストされているものを除く、すべての IMS 異常終了コード

DRA は、SDUMP または SNAP データ・セット出力という形で診断情報を提供します。X'80' では、最初に SDUMP が試みられます。それに失敗すると、SNAP が実行されます。X'84' では SDUMP は試みられず、SNAP が試みられます。

z/OS または IMS の異常終了コードを伴う障害では、DRA スレッドが終了し、スレッド TCB が消滅します。IMS DB 戻りコードは、DRA 自体またはスレッド TCB に何の影響も及ぼしません。

スレッド要求を処理していないときに DRA スレッド TCB の障害が発生すると、SDUMP/SNAP プロセスが実行されます。DRA 要求を処理していないときに DRA 制御 TCB の障害が発生すると、SDUMP/SNAP プロセスが実行され、制御

出口ルーチンが呼ばれます。タイプ SCHED のスレッド要求では、X'80' または X'84' を伴う障害が発生すると、SNAP か SDUMP が実行されます。

## SDUMP

SDUMP 出力には次のものが含まれます。

- IMS 制御領域
- DLISAS アドレス・スペース
- キー 0 およびキー 7 CSA
- DRA 専用ストレージの選択部分 (アドレス・スペース制御ブロック (ASCB)、タスク制御ブロック (TCB)、および要求ブロック (RB) を含む)

IMS 制御ブロックは、オフライン・ダンプ・フォーマッター (ODF) で書式設定できます。

ODF は、DRA ストレージをフォーマットしません。IPCS を使用して CCTL の専用ストレージで z/OS ブロックをフォーマットできます。

DRA SDUMPS には、独自の SDUMP オプションがあります。その結果、CHNGDUMP 指定では、DRA SDUMP のセクションを省くことができません。これらの指定が DRA のオプション・リストにないときは、DRA SDUMP に付加的效果をもたらします。

## SNAP

SNAP ダンプ・データ・セットは、SNAP ダンプが必要とされる都度、動的に割り振られます。SYSOUT クラスは、DRA 始動テーブルのパラメーターで定義されます。

SNAP 出力には次のものが含まれます。

- DRA 専用ストレージの特定部分 (ASCB、TCB、RB を含みます)
- IMS DB の制御ブロック

関連資料:

 オフライン・ダンプ・フォーマッター・ユーティリティー (DFSOFMD0) (システム・ユーティリティー)

---

## 第 5 部 データベース・リカバリー管理 (DBRC)

総称してリカバリー管理 (RECON) データ・セットと呼ばれる、一連の VSAM データ・セットに保管された情報を記録および管理するには、DBRC を使用できます。この情報を基に、DBRC を使用して、特定の IMS アクションへの対処方法を IMS にアドバイスすることができます。



---

## 第 26 章 DBRC API

ユーザーのアプリケーションは、DBRC アプリケーション・プログラミング・インターフェース (API) を介してデータベース・リカバリー管理 (DBRC) からのサービスを受けられます。この API は、リリースに依存しない、アセンブラ・マクロ・インターフェースです。アプリケーションは DBRC に DBRC API 要求を発行することによりサービスを受け、DBRC はその結果をアプリケーションが検索できるストレージ内のエリアに戻します。

この DBRC API は DSPAPI マクロに入れて IMS と一緒に提供されます。DBRC API を使用するサンプル・アプリケーション・プログラム (DSPAPSMPL) は、IMS.ADFSSMPL (IMS.SDFSSMPL と同じ) ライブラリーの中に組み込まれています。

**重要:** すべての DBRC API 要求は、DBRC 開始要求 (STARTDBRC) が発行された TCB と同じ TCB で発行する必要があります。別の TCB で発行された要求は、理由コード X'C90000A' により失敗します。

DBRC API を使用するプログラムを作成するには、次の項目に関する実用的な知識が必要です。

- アセンブラ言語のプログラミング
- z/OS と、z/OS が提供するサービス
- IMS
- DBRC

関連概念:

 z/OS: HLASM の言語解説書

 z/OS: HLASM プログラマーズ・ガイド

432 ページの『照会要求からの出力』

関連資料:

417 ページの『第 28 章 DBRC 許可要求 (AUTH)』

425 ページの『第 29 章 DBRC コマンド要求 (COMMAND)』

431 ページの『第 30 章 DBRC 照会要求』

511 ページの『第 31 章 DBRC バッファ解放要求』

515 ページの『第 32 章 DBRC 開始要求 (STARTDBRC)』

521 ページの『第 33 章 DBRC 停止要求 (STOPDBRC)』

525 ページの『第 34 章 DBRC 無許可要求 (UNAUTH)』

437 ページの『データベース照会要求 (TYPE=DB)』

461 ページの『DBDS 照会要求 (TYPE=DBDS)』

---

## DBRC API にアクセスするためのアプリケーションの構造

アプリケーションが DBRC API にアクセスするためには、一般的なアプリケーションの構成に従う必要があります。

DBRC API を使用するアプリケーションの一般的な構成は、次のとおりです。

1. API DSECTS を組み込む (DSPAPI FUNC=DSECT)
2. API を初期化する、DBRC を開始する、RECON データ・セットへの接続を確立する、および API トークンを受け取る (DSPAPI FUNC=STARTDBRC)
3. 1 つ以上の照会要求を発行する (DSPAPI FUNC=QUERY)
4. 照会要求から戻された情報を処理する
5. バッファ・ストレージを戻す (DSPAPI FUNC=RELBUF)
6. DBRC と DBRC API を終了する (DSPAPI FUNC=STOPDBRC)

### アプリケーション・プログラムによる DBRC API 環境の確立方法

アプリケーション・プログラムは、DSPAPI FUNC=STARTDBRC マクロを使用してアプリケーション・プログラム・インターフェース (API) 環境を確立します。このマクロ機能は、API を初期化して、データベース・リカバリー管理 (DBRC) RECON データ・セットとの接続を作成し、API トークンを返します。

要件:

- API トークンは、それ以降のすべての API 呼び出しに指定する必要があります。
- すべての API 要求は、FUNC=STARTDBRC が発行されたタスク制御ブロック (TCB) と同じ TCB で発行する必要があります。

関連概念:

406 ページの『DBRC API トークン』

### アプリケーション・プログラムによる DBRC API 環境の終了方法

ユーザーのアプリケーション・プログラムは、DBRC での作業を完了した後、DSPAPI FUNC=STOPDBRC マクロを使用して API 環境を終了する必要があります。このマクロ機能によって、DBRC は終了することができ、さらに DBRC が割り振っていたすべてのデータ・セットを割り振り解除し、API 環境に割り振られているストレージを解放します。

### アドレッシング・モードと常駐モード

アプリケーションは DBRC API を 31 ビット・アドレッシング・モードで呼び出す必要があります。16MB 境界の上下どちらにでも常駐できます。

API に渡されるパラメーター・アドレスは、マクロ記述に特に示されていない限り、16MB 境界の上下どちらにあってもかまいません。同様に、API によってユーザーに戻されるパラメーター・アドレスも、16MB 境界の上下どちらでもかまいません。

## アドレス・スペース制御 (ASC) モードおよび状態

ユーザーのアプリケーションは API サービスを呼び出すには 1 次 ASC モードでなければなりません。

仮想記憶間モードはサポートされていません。このアプリケーションは、問題プログラム状態でも監視プログラム状態でも実行でき、さらに APF 許可プログラムとしても実行できます。

## DBRC API によるレジスタの使用方法

汎用レジスタ 0、1、14、および 15 は、DBRC API によって変更できます。レジスタ 13 には標準 (18 ワード) 保管域のアドレスが含まれていなければなりません。レジスタ 2 から 13 の内容は、DBRC API によって変更されません。

## DBRC API アプリケーションへの等価 (EQU) ステートメントの組み込み方法

DSPAPI マクロで使用される等価 (EQU) ステートメントおよび定数をユーザーのプログラムに組み込むには、DSPAPI 要求の DSECT 機能を使用します。

DSPAPI の DSECT 機能の構文は、次のとおりです。

▶▶—DSPAPI—FUNC=DSECT—▶▶

## API アプリケーション

DSPAPI マクロに指定するパラメーターは、リテラル、アドレス、または値のいずれかです。アドレスまたはパラメーター値は、レジスタ、シンボル、またはリテラルを使用して渡すことができます。リテラル値は大/小文字混合で指定することができます。

レジスタを使用する

レジスタを使用するには、シンボルまたは値のアドレスを汎用レジスタの 1 つにロードし、そのレジスタ (括弧で囲む) を DSPAPI 要求のパラメーターとして使用する必要があります。2 から 12 の範囲のレジスタのみを使用してください。レジスタ表記は MF=L では機能しません。この形式では実行可能コードが作成されないためです。

次に示すのは、レジスタを使用してアドレスを渡す場合の例です。

```
LA 5,OUTPUTAD
DSPAPI FUNC=QUERY,OUTPUT=(5),...
.
.
OUTPUTAD DS A
```

上記の例から生成される命令は、次のとおりです。

```
ST 5,DSPAPI_Plist_Output
```

シンボル名を使用する場合

シンボル名を使用するには、必要な値を含む名前付きストレージ、または戻されるアドレスまたは値を受け取る名前付きストレージ域を定義し、そのシンボル名を DSPAPI 要求のパラメーターとして使用する必要があります。

次に示すのは、シンボル名を使用してアドレスを渡す場合の例です。

```
DSPAPI FUNC=QUERY,OUTPUT=OUTPUTAD,..  
.  
.  
OUTPUTAD DS A
```

上記の例から生成される命令は、以下のとおりです。

```
LA 0,OUTPUTAD  
ST 0,DSPAPI_Plist_Output
```

リテラルを使用する場合

リテラルは、特定のパラメーター値、例えば、タイム・スタンプなどに使用することができます。リテラルを使用するには、DSPAPI 要求のパラメーターとしてリテラルを 16 進数ストリングで渡します。リテラルは大/小文字混合で指定することもできます。

次に示すのは、リテラルを使用して値を渡す場合の例です。

```
DSPAPI FUNC=QUERY,TYPE=LOG, 72  
STARTIME==XL12'1980030F191212009999028D' C
```

上記の例から生成される命令は、以下のとおりです。

```
LA 0,=XL12'198030F191212009999028D'  
ST 0,DSPAPI_Plist_Startime
```

特に注記のない限り、名前フィールドは 8 文字のフィールドで、左寄せされ、ブランクが埋め込まれます。

## DBRC API マクロのバージョン

パラメーター・リストは IMS のリリースによって変更される場合があるため、DSPAPI マクロが提供する関数にはそれぞれバージョンが関連付けられています。あるバージョンに関連付けられた関数またはパラメーターを使用するには、そのバージョン番号かそれ以降のバージョンを指定する必要があります。

バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要な最小バージョン番号が示されます。

DBRC API の出力ブロック・バージョン番号は以下のとおりです。

- 5.0 (IMS バージョン 13 の場合)
- 4.0 (IMS バージョン 12 の場合)
- 3.0 (IMS バージョン 11 の場合)

## DBRC API トークン

DBRC API トークンは、一連の API 要求を関連付けるために使用される 4 バイトのフィールドです。

DSPAPI FUNC=STARTDBRC マクロが発行されると、プログラムはこのトークンを受け取ります。このトークンは、STARTDBRC マクロのこのインスタンスに関連する他のすべてのマクロ呼び出しで提供される必要があります。DSPAPI FUNC=STOPDBRC マクロ呼び出しの後、このトークンは無効になります。マクロ

で重大エラー (戻りコード X'0000000C') が発生した場合は、プログラムはトークンを受け取りません。機能が警告エラー (戻りコード X'00000004') を受け取った場合は、付随する理由コードの意味を 515 ページの『第 32 章 DBRC 開始要求 (STARTDBRC)』で検索して、必要なアクションを判別します。

関連概念:

404 ページの『アプリケーション・プログラムによる DBRC API 環境の確立方法』

## DSPAPI マクロのマクロ形式

DSPAPI マクロには、標準 (S)、リスト (L)、変更 (M)、および実行 (E) の 4 つの形式があり、COMPLETE と NOCHECK という 2 つのバリエーションがあります。リスト形式、変更形式、および実行形式は、通常、再入可能プログラムの作成時、またはアプリケーションが多重要要求を発行するときに、組み合わせて使用されます。

必要に応じてデフォルトが使用されます。標準形に比べて、行われる妥当性検査が少なくなります。次に、各形式をいつ、どのような理由で使用するかについて説明します。

標準形 (デフォルト)

標準形のマクロ (MF=S または MF=S,list) は、インライン・パラメーター・リストを生成および変更するときに使用します。ユーザーのプログラムが再入可能の場合、標準形のマクロは使用しないでください。再入可能コードが変更できないためです。いくつかの例外がありますが、再入可能コードの作成に標準形のマクロを使用した場合、そのコードを実行すると異常終了が起きます。標準形のマクロは、次の 3 つのを行います。

- インライン・パラメーター・リストを作成する
- 要求に指定されたパラメーターによってパラメーター・リストを変更する
- API に要求を送信する

標準形を使用する場合、MF=(S,list) を指定することにより、生成されたパラメーター・リストにオプションでラベルを割り当てることができます。この list は、この形式のマクロによって作成されたパラメーター・リストに割り当てられたラベル名を指定します。標準の MF=S 形式のマクロがデフォルトです。

リスト形式

リスト形式のマクロ (MF=L,list) は、ラベル付きのインライン・パラメーター・リストを生成するときに使用します。このリストには、マクロに指定されたパラメーター値が取り込まれます。リスト形式は既存のリストを変更せず、API に要求を送信しません。要するに、リスト形式は、変更形式または実行形式 (使用しようとしている実際のリスト) のターゲットとして使用できるテンプレートを作成します。パラメーター・リストが再入可能コードで生成されている場合、変更できません。そのため、パラメーター・リストを保持するために十分な大きさのストレージを取得し、このストレージを変更形式またはリスト形式のターゲットとして使用する必要があります。

*list* は、この形式のマクロによって作成されたパラメーター・リストに割り当てられたラベル名を指定します。

レジスター表記にはマクロのリスト形式との互換性がありません。その代わりに、ADCON ゼロが生成されます。

#### 変更形式

変更形式のマクロ (MF=M,*list*,COMPLETE | NOCHECK) は、リスト・パラメーターによって指定されたパラメーター・リストで、マクロに指定された値を変更するときに使用します。変更形式のマクロはパラメーター・リストを生成せず、DBRC に要求を出しません。

#### リスト

変更されるパラメーター・リストのアドレスを指定する、シンボルまたは 2 から 12 の範囲の汎用レジスター。

#### COMPLETE

DBRC がリストに含まれないすべてのパラメーターに対してデフォルト (デフォルトがあるパラメーター用) を使用し、リストに指定されているパラメーターに対して妥当性検査を行うように指定します。

#### NOCHECK

デフォルトが設定されず、既存のパラメーター・リストが使用されるように指定します。妥当性検査は最小限になります。ただし、無効なキーワードの組み合わせには、エラーとしてフラグが立てられます。

#### 実行形式

実行形式のマクロ (MF=E,*list*,COMPLETE | NOCHECK) は、次の操作に使用します。

- リスト形式で指定されなかった可能性のある新規および追加の許容パラメーターを使用して、(リスト形式で生成された) パラメーター・リストを変更する
- API に要求を出す

これ以降、実行形式のマクロを呼び出すたびに、そのマクロのパラメーターを変更することができます。

#### リスト

シンボルまたは 2 から 12 の範囲の汎用レジスターを指定でき、使用されるパラメーター・リストのアドレスを指定します。

#### COMPLETE

DBRC がリストに含まれないすべてのパラメーターに対してデフォルト (デフォルトがあるパラメーター用) を使用し、リストに指定されているパラメーターに対して妥当性検査を行うように指定します。

#### NOCHECK

デフォルトが設定されず、既存のパラメーター・リストが使用されるように指定します。妥当性検査は最小限になります。ただし、無効なキーワードの組み合わせには、エラーとしてフラグが立てられます。

以下の構文に、マクロ形式の要約を示します。

**MF=S | L | M | E**

マクロ形式を指定します。

**MF=S | MF=(S,list)**

マクロの標準形を指定します。MF=S がデフォルトです。

**MF=(L,list)**

マクロのリスト形式を指定します。

**MF=(M,list,COMPLETE | NOCHECK)**

マクロの変更形式を指定します。

**MF=(E,list,COMPLETE | NOCHECK)**

マクロの実行形式を指定します。

関連資料:

- 418 ページの『AUTH 要求のパラメーター』
- 425 ページの『コマンド要求のパラメーター』
- 432 ページの『バックアウト照会要求 (TYPE=BACKOUT)』
- 477 ページの『ログ照会要求 (TYPE=LOG)』
- 486 ページの『OLDS 照会要求 (TYPE=OLDS)』
- 499 ページの『RECON 状況照会要求 (TYPE=RECON)』
- 504 ページの『サブシステム照会要求 (TYPE=SUBSYS)』
- 511 ページの『第 31 章 DBRC バッファ解放要求』
- 521 ページの『第 33 章 DBRC 停止要求 (STOPDBRC)』
- 515 ページの『第 32 章 DBRC 開始要求 (STARTDBRC)』
- 525 ページの『第 34 章 DBRC 無許可要求 (UNAUTH)』

## 照会出力ブロック・ヘッダー

以下の例に、照会要求からの出力の一般的なフォーマットを示します。この図の後に記載する DSECT の例で、ストレージ・ブロックのフィールドとそれらのフィールドの相互関係についての詳細を説明します。

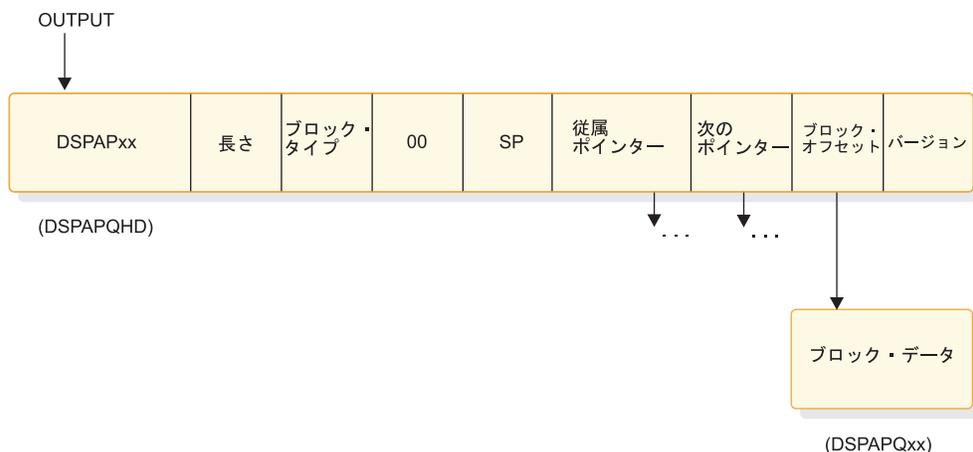


図 10. 照会出力要求の一般的なフォーマット

STARTDBRC の場合、SSID はオプションで指定されるサブシステム ID です。STOPDBRC の場合、SSID は STARTDBRC 要求で指定されたサブシステム ID です。

### DSPAPQHD - QUERY 出力ブロック・ヘッダー

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	32	DSPAPQHD	
0	(0)	CHARACTER	8	APQHD_EYECATCHER	Output area eyecatcher
8	(8)	SIGNED	4	APQHD_LENGTH	Block length, hdr + data
12	(C)	SIGNED	2	APQHD_BLKTYPE	Block type
14	(E)	UNSIGNED	2	*	Reserved
15	(F)	UNSIGNED	1	APQHD_SUBPOOL	Subpool ID
16	(10)	ADDRESS	4	APQHD_DEPPTR	Ptr to block dependent
20	(14)	ADDRESS	4	APQHD_NEXTPTR	Ptr to next block of the same type
24	(18)	UNSIGNED	4	APQHD_BLKOFFSET	Offset to block data
28	(1C)	SIGNED	4	APQHD_VERSION	Version of output block

## DBRC API の実行時の考慮事項

DBRC API の実行中には、DSPAPI および RECON アクセス、IMSplex 内での API の動作方法など、対処しなければならないさまざまな考慮事項があります。

### DSPAPI マクロ・アクセス

DBRC API (DSPAPI) は IMS と一緒に配布されるため、アプリケーションでは IMS.SDFSRESL を JOBLIB または STEPLIB のいずれかとしてプログラムの JCL に割り振り、DSPAPI マクロへのアクセスを提供する必要があります。

### RECON データ・セット・アクセス

DBRC API はその実行中、DD 名 RECON1、RECON2、および RECON3、あるいはそれぞれの代替 DD 名 によって割り振られた最大 3 つの RECON データ・セットを使用します。DBRC API が RECON を割り振るには、その前に、データ・セットに MDA メンバーを作成する必要があります。

RECON は、次の方法で割り振ることができます。

- DSPAPI FUNC=STARTDBRC 要求が発行されたときに API によって動的に (推奨方法)
- ユーザーのアプリケーション・プログラムによって動的に
- 使用中の JCL から

DBRC API によって RECON が割り振られた場合、それらの RECON は DSPAPI FUNC=STOPDBRC 要求が出されたときに割り振り解除されます。DBRC API によって RECON を割り振るには、事前に MDA メンバーを RECON1、RECON2、および RECON3 用、あるいはそれぞれの代替 DD 名用に作成しておく必要があります。MDA メンバーは、//IMSDALIB として割り振られたライブラリーか、//JOB LIB または //STEPLIB ライブラリーの中に作成する必要があります。DBRC API はまず最初に、MDA メンバーを //IMSDALIB (もしあれば) で検索します。

JOB LIB または STEPLIB と連結された IMS ライブラリーは、通常 APF 許可ライブラリーです。ユーザーのプログラムが APF 許可ライブラリーを実行し、MDA メンバーを含むライブラリーが APF 許可でない場合は、DD 名として IMSDALIB を使用してそのライブラリーを割り振ります。

要件: 1 つのセットになっている RECON データ・セットにアクセスするすべてのジョブは、同じデータ・セットを同じ DD 名で割り振る必要があります。例えば、RECON1=dsn1、RECON2=dsn2、および RECON3=dsn3 などのように指定します。この規則に従わない場合、RECON に重大な損傷が発生する原因となります。

API は、FUNC=STARTDBRC 要求とそれに関連した FUNC=STOPDBRC 要求の間では、1 セットの RECON しか使用しません。ただし、FUNC=STOPDBRC 要求の後に RECON を割り振り解除してから、次の FUNC=STARTDBRC 要求を出す前に新たな RECON のセットを割り振ると、ユーザーのプログラムが複数の RECON セットを使用できます。この処理を機能させるには、ユーザーのプログラムが RECON を動的に割り振る必要があります。

関連資料:

 DFSMDA マクロ (システム定義)

## RECON アクセス権限

DBRC には、3 つのレベルのアクセス制御を設定できます。それぞれのレベルによって、RECON データ・セットにアクセスするユーザーに与えられる権限は異なります。

DBRC では、以下の 3 つのレベルのアクセス制御が可能です。

- RECON データ・セットの情報の削除と定義を行うユーザーには、ALTER 権限が必要です。
- RECON データ・セットの更新のみを行うユーザーには、UPDATE 権限が必要です。
- RECON データ・セットの更新は行わないが、RECON データ・セットへの情報の照会を行うユーザーには、READ 権限が必要です。

DBRC 開始 API 要求 (STARTDBRC) のための READONLY キーワードを使用すると、READONLY オプションを指定できます。READONLY はまた、データベース・リカバリー管理ユーティリティ (DSPURX00) では JCL EXEC PARM です。READONLY は、そのジョブによって RECON データ・セットの更新ができないことを示します。READONLY は、ジョブを実行するユーザーがデータ・セットに対して UPDATE (またはそれ以上の) 権限を持っていない場合に必要です。READONLY を指定すると、RECON データ・セットにエラーが発生しても、リカバリー・アクションは実行されません。スペア・データ・セットにスワッピングするのではなく、ジョブが強制終了します。

## DBRC 要求のタイム・スタンプ・フォーマット

DBRC 要求 (入力または出力) に関連付けられるタイム・スタンプは、次のパック 10 進 UTC 時刻形式に従います。DBRC は照会要求に指定されたタイム・スタンプのオフセット・フィールドの値を無視します。

タイム・スタンプの形式は、次のとおりです。

yyyydddFhhmmssthmijufqqqs

ここで、

**yyyy** 年 (0000 から 9999)

**ddd** 日 (000 から 366)

**F** 埋め込みの目的で使用する 16 進文字 (X'F')

**hh** 時 (0 から 23)

**mm** 分 (0 から 59)

**ss** 秒 (0 から 59)

**thmiju**

100 万分の 1 秒 (マイクロ秒、000000 から 999999)

**fqqqs** 時間帯オフセット。

**f** フラグ・ビット。UTC 表記の場合、通常は 0

**qq** 1 時間の 4 分の 1 (32/4=8)

**s** 符号 (D は負、C は正)

## DBRC による出力データ・セットの使用法

DBRC API で実行中、DBRC が、メッセージおよびその他の情報を DD 名 SYSPRINT、またはユーザーが指定した DD 名によって定義されたデータ・セットに出力する場合があります。ユーザーのプログラムがすでに DD 名 SYSPRINT を使用している場合、API が使用する代替 DD 名を指定することができます。

この出力データ・セットは、テープ、DASD、プリンター上、または出力ストリーム (SYSOUT) を通じて経路指定される場合があります。このデータ・セットは、JCL を使用してユーザーのプログラムによって割り振られる場合、または DBRC API の呼び出し前に動的に割り振られる場合があります。このデータ・セットの属性は、RECFM=FBA、LRECL=121 です。DBRC はシステムが決定したブロック・サイズを使用するため、ブロック・サイズは指定しないでください (システムは、その装置に最適なブロック・サイズを判断します)。

## 照会要求の名前パラメーターに対するワイルドカード・サポート

より柔軟な照会要求を行うために、名前を指定するいくつかのキーワード・パラメーターにワイルドカード (アスタリスク) を使用できます。このワイルドカード・サポートを使用すると、照会に対する名前パターンを指定することにより、照会に応じてその拡張やフィルター操作を行うことができます。

アスタリスクは以下の 2 つの形式で使用できます。

- 照会タイプに応じてすべての DB 名、グループ、またはサブシステムの情報を要求するには、アスタリスクを単独で使用します。
- パターンと一致する名前を持つ DB 名、グループ、またはサブシステムの情報を要求するには、名前の最後にアスタリスクを使用します。この場合、アスタリスクの前には、英字が少なくとも 1 つ必要です。

ワイルドカードの使用は、以下のパラメーターでサポートされています。

表 84. ワイルドカードがサポートされる DBRC QUERY のパラメーター

パラメーター名	照会タイプ	構文
DBNAME	DB	DBNAME= <i>dbname</i>   <i>dbname</i> *
GROUP	All xxxxGROUP	GROUP= *   <i>name</i>   <i>name</i> *
NAME	All xxxxGROUP	NAME= *   <i>name</i>   <i>name</i> *
		NAME=*
SSID	<ul style="list-style-type: none"><li>• BACKOUT</li><li>• OLDS</li><li>• SUBSYS</li></ul>	SSID= *   <i>symbol</i>   <i>symbol</i> *



## 第 27 章 DBRC API セキュリティー・フィーチャー

RECON データ・セットへのアクセスを特定のユーザーに制限することが必要な場合があります。DBRC API では、ユーザーが出す個別の DBRC API 要求にインストール制御を与えることができます。

以下の表は、DBRC API 要求と、各要求を保護するためにセキュリティ製品が使用するリソース・プロファイルを示しています。記号 \* は、ワイルドカード値を示します。

表 85. DBRC API 要求とリソース・プロファイル：

機能	タイプ	パラメーター	リソース
STARTDBRC または STOPDBRC	N/A	パラメーターの指定なし	hlq.STDBRC  このリソースは、ssid が指定されていない場合に使用されます。
		SSID=ssid	hlq.STDBRC.ssid  STARTDBRC の場合、ssid はオプションで指定されるサブシステム ID です。  STOPDBRC の場合、ssid は STARTDBRC 要求で指定されるサブシステム ID です。
RELBUF	N/A	N/A	N/A
QUERY	RECON	N/A	hlq.LIST.RECON
QUERY	DB	DBNAME=name	hlq.LIST.DB.name
		DBNAME=name*	hlq.LIST.DB.ALL
		DBLIST=dblist	hlq.LIST.DB.ALL
		LOC=FIRST   NEXT	hlq.LIST.DB.ALL
QUERY	PART	DBNAME=name	hlq.LIST.DB.name
		PARTNAME=name	hlq.LIST.DB.name
QUERY	DBDS	DBNAME=name	hlq.LIST.DBDS.name
		GROUP=grpname	hlq.LIST.DBDS.grpname
QUERY	LOG	STARTIME	hlq.LIST.LOG.STARTIME
		FROMTIME   TOTIME	hlq.LIST.LOG.ALL
QUERY	OLDS	SSID=ssid   ssid*   *	hlq.LIST.LOG.ALLOLDS

表 85. DBRC API 要求とリソース・プロファイル (続き):

機能	タイプ	パラメーター	リソース
QUERY	SUBSYS	SSID=ssid	hlq.LIST.SUBSYS.ssid
		SSID=ssid*	hlq.LIST.SUBSYS.ALL
		SSID=*	hlq.LIST.SUBSYS.ALL
		SSTYPE=ALL	hlq.LIST.SUBSYS.ALL
		SSTYPE=BATCH	hlq.LIST.SUBSYS.BATCH
		SSTYPE=ONLINE	hlq.LIST.SUBSYS.ONLINE
		SSTYPE=DBRCAPI	hlq.LIST.SUBSYS.DBRCAPI
QUERY	BACKOUT	SSID=ssid	hlq.LIST.BKOUT.ssid
		SSID=ssid*	hlq.LIST. BKOUT.ALL
		SSID=*	hlq.LIST. BKOUT.ALL
QUERY	DBDSGROUP	GROUP=grpname	hlq.LIST.DBDSGRP.grpname
		GROUP=*   grpname*	hlq.LIST.DBDSGRP.ALL
QUERY	DBGROUP	GROUP=grpname	hlq.LIST.DBDSGRP.grpname
		GROUP=*   grpname*	hlq.LIST.DBDSGRP.ALL
QUERY	RECOVGROUP	GROUP=grpname	hlq.LIST.DBDSGRP.grpname
		GROUP=*   grpname*	hlq.LIST.DBDSGRP.ALL
QUERY	CAGROUP	GROUP=grpname	hlq.LIST.CAGRP.grpname
		GROUP=*   grpname*	hlq.LIST.CAGRP.ALL
QUERY	GSGROUP	GROUP=grpname	hlq.LIST.GSG.grpname
		GROUP=*   grpname*	hlq.LIST.GSG.ALL
COMMAND	N/A	N/A	現行コマンド・リソースを使用します。
AUTH UNAUTH	N/A	AUHLIST=list	hlq.AUTH.dbname  リスト内の各 dbname が検査されます。

## 第 28 章 DBRC 許可要求 (AUTH)

無効なデータ共用環境が作成されないようにするには、DBRC 許可要求を使用します。データベース許可は、データベースへのアクセスとその使用の許可を要求するプロセスです。このコンテキストでは、データベースは DL/I データベースまたは高速機能 DEDB エリアです。

要求されるアクセスのタイプは、以下のとおりです。

- リソースの排他的 (EX) 制御。排他的アクセス権限が付与されるのは、データベースが現在許可されていない場合です。この権限が付与されたら、他のすべての許可要求は許可されません。このタイプの許可は、通常、リカバリー・ユーティリティーで使用されます。
- データベースに対する読み取り (RD) 権限 (保全性を維持した読み取りとも呼ばれます)。読み取り権限が付与されるのは、データベースが排他的アクセスまたは更新の対象として現在許可されていない場合です。このアクセス権限が付与されたら、排他的アクセスまたは更新のための他の許可要求は失敗します。このタイプの許可は、通常、非並行型 (クリーン) イメージ・コピーをとるユーティリティーで使用されます。
- データベースに対する読み取り専用 (RO) 権限 (ダーティー読み取りとも呼ばれます)。読み取り専用権限が付与されるのは、データベースが排他的アクセスの対象として現在許可されていない場合です。このアクセス権限が付与されたら、排他的アクセスのための他の許可要求は失敗します。読み取り専用権限は、アプリケーションがデータベースを読み取っているときに、そのデータベースが更新されるのを防止するものではありません。このタイプのアクセスは、通常、並行イメージ・コピーをとるユーティリティーで使用されます。

この機能を使用するためには、アプリケーションは初期化時に DBRC に登録する必要があります。要求するデータベースは、DBRC に登録されている必要があります。

データベースは、DSPAPI FUNC=UNAUTH マクロによって明示的に、あるいは、DSPAPI FUNC=STOPDBRC マクロによって暗黙的に無許可になっています。

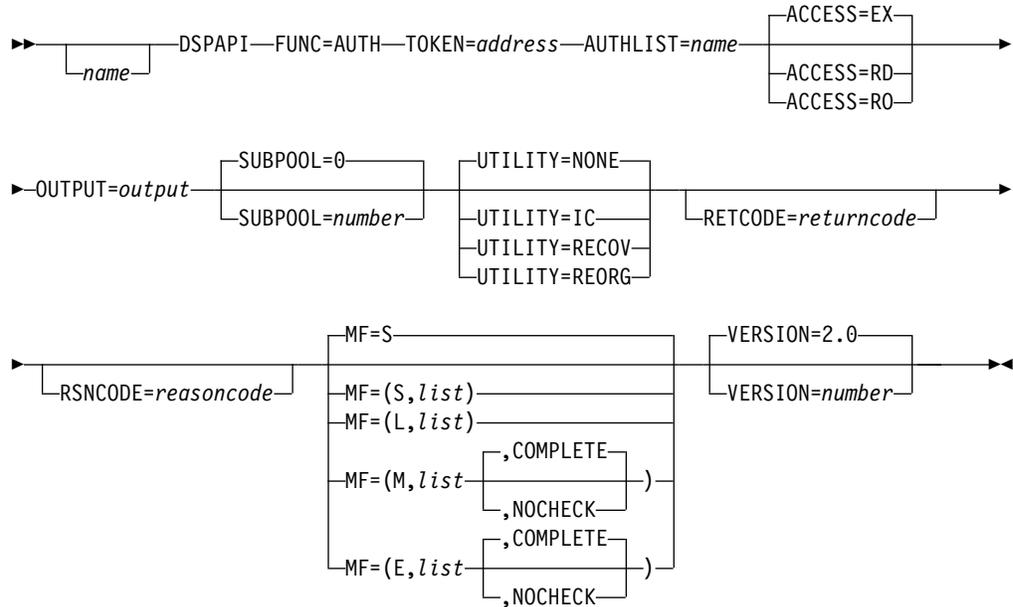
許可機能または無許可機能には 1 つ以上のデータベース名を指定できます。データベースは、許可されたときと同じ順序で無許可にする必要はありません。例えば、リスト内の複数のデータベースが許可されている場合、一度に 1 つずつ任意の順序でそれらが無許可にすることができます。アクセス権限のタイプを変更するには、データベースを無許可にしてから、もう一度そのデータベースを許可します。

標準データベース許可が拒否される場合もあります。例えば、データベースの「Prohibit Authorization」フラグがオンの場合や、そのデータベースの 1 つ以上の DBDS がイメージ・コピーを必要としている場合などです。アプリケーションの意図がデータベースのイメージ・コピー、リカバリー、または再編成である場合は、FUNC=AUTH 要求で UTILITY キーワードを使用することで、権限を付与できます。

関連概念:

## AUTH 要求の構文

以下の構文図から、DBRC AUTH 要求のフォーマットを理解することができます。



## AUTH 要求のパラメーター

以下の情報から、DBRC AUTH 要求のパラメーターについて理解できます。各パラメーターは、AUTH 要求の構文図との関連で説明されています。

### name name

1 桁目から name を始めます。

### TOKEN=address | (2-12)

FUNC=STARTDBRC マクロで返された API トークンのアドレスを指定します。

### AUTHLIST=name | (2 - 12)

許可するデータベース名または高速機能エリアのリストを指定します。このリストは、リスト内のエレメント数を含むフルワード、エレメントの長さを含むフルワード、およびそれに続く 1 つ以上のエレメントで構成されます。各エレメントは、8 文字の DB 名または高速機能 DEDB 名、および 8 文字のブランク (X'40') または高速機能エリア名で構成されます。

### ACCESS=EX | RD | RO

排他的 (EX)、読み取り (RD)、または読み取り専用 (RO) 許可が要求されることを指定します。デフォルトは EX です。

### OUTPUT=output | (2 - 12)

許可出力ブロック DSPAPAUB を示すポインタを受け取るフィールドを指定します。

出力が作成されなかった場合、出力アドレスはゼロです。これは、RECON データ・セット内に要求を満たす情報がない場合、または出力を作成できるようになる前にエラーが発生した場合に起こります。

出力ブロックのストレージは、呼び出し元によって事前割り振りされません。DBRC はこれらのブロック用に指定されたサブプールからストレージを取得します。呼び出し元は、バッファ解放サービス (DSPAPI FUNC=RELBUF) を使用してこのストレージを解放し、返された出力アドレスを指定する必要があります。

**SUBPOOL= 0 | *number***

取得されるストレージのサブプール番号を指定します。このパラメーターを指定しない場合、FUNC=STARTDBRC 要求によって指定されたサブプールがデフォルトになります。それ以外の場合は、サブプール 0 がデフォルトになります。

**UTILITY=IC | RECOV | REORG | NONE**

アプリケーションの目的のユーティリティー機能を指定します。機能は、イメージ・コピー (IC)、データベース・リカバリー (RECOV)、データベース再編成 (REORG)、または NONE です。デフォルトは NONE です。これは、標準のデータベース許可要求を示します。

**RETCODE=*return code* | (2-12)**

シンボルとして指定する場合、戻りコードを受け取るために、ストレージのワードを示すラベルを指定します。レジスターとして指定する場合、戻りコードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RETCODE が指定されたかどうかにかかわらず、レジスター 15 は戻りコードを含んでいます。

**RSNCODE=*reason code* | (2-12)**

シンボルとして指定する場合、理由コードを受け取るために、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、理由コードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RSNCODE が指定されたかどうかにかかわらず、レジスター 0 は理由コードを含んでいます。

**MF=S | L | M | E**

要求のマクロ形式を指定します。

**VERSION=2.0 | *number***

このマクロによって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターはマクロ処理によって受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

FUNC=AUTH 要求の有効なバージョン番号は、2.0 です (デフォルト)。

関連概念:

407 ページの『DSPAPI マクロのマクロ形式』

関連資料:

424 ページの『AUTH 出力ブロック』

## AUTH の戻りコードと理由コード

以下の表で、DBRC AUTH 要求の戻りコードと理由コードを検索できます。この表には、各コードのタイプおよび説明も記載されています。

### AUTH 要求の戻りコードと理由コード

表 86. DSPAPI FUNC=AUTH の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
警告	X'00000008'	X'C1000001'	AUTHLIST の 1 つ以上の項目を処理できません。理由コードは、AUTH 出力ブロック内の対応する項目に設定する必要があります。
重大エラー - 返される AUTH ブロッ クがありません	X'0000000C'	X'C1000001'	アプリケーションは DBRC にサインオンされていません。
	X'0000000C'	X'C1000002'	アプリケーションがサインオンされていないため DBRC AUTH 処理が完了できず、サブシステム・レコードが検出されません。以前の検査ではサブシステムがサインオンされたことが示されていたため、通常の状態ではこのエラーが発生することはありません。
	X'0000000C'	X'C1000003'	RSR 許可処理エラー。トラッカーとしての許可はサポートされていないため、このエラーが発生することはありません。
	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。
ストレージ・ エラー	X'00000028'	X'C1000001'	AUTH 出力ブロック用のストレージの取得時にエラーが発生しました。
内部エラー	X'0000002C'	X'C1000001'	RECON の複数更新処理を開始しようとしてエラーが発生しました。
	X'0000002C'	X'C1000002'	RECON の複数更新処理を終了しようとしてエラーが発生しました。
	X'0000002C'	X'C1000003'	AUTH 出力ブロック内に項目を検出できませんでした。このエラーが発生することはありません。

表 86. DSPAPI FUNC=AUTH の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'0000002C'	X'C1000004'	DBRC 許可処理中に内部エラーが発生しました。
	X'0000002C'	X'C1000005'	DBRC 許可処理中に内部エラー (無効なパラメーター) が発生しました。
パラメーター・エラー	X'00000030'	X'C1000001'	AUTHLIST が渡されませんでした。
	X'00000030'	X'C1000002'	項目を持たない AUTHLIST が渡されました。
	X'00000030'	X'C1000003'	AUTHLIST 内のエレメントが重複しています。
	X'00000030'	X'C1000004'	OUTPUT パラメーターが欠落しているか無効です。
	X'00000030'	X'C9000001'	API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000003'	RETCODE アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000005'	OUTPUT アドレスが無効です。API OUTPUT を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C900000A'	要求する機能 (FUNC) に対して、不正な VERSION 値が指定されました。
	X'00000030'	X'C900001A'	AUTHLIST アドレスが無効です。API AUTHLIST を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

関連資料:

422 ページの『AUTH 出力ブロックの APAUB\_RsnCode』

## AUTH 出力ブロックの APAUB\_RsnCode

以下の表で、AUTH 要求の戻りコードと理由コードに対応する APAUB\_RsnCode 値を検索できます。この表には、各コードの説明も記載されています。

AUTH 出力ブロック (DSPAPAUB) が返される時、以下の理由コードのいずれかが、要求内の DB またはエリアのリストの各エレメントの APAUB\_RsnCode フィールドに設定されます。

表 87. AUTH 要求の戻りコードと理由コードに対応する APAUB\_RsnCode 値

APAUB_RsnCode	意味
X'00000000'	要求が正常に完了した。
X'C1000100'	セキュリティー・エラー。SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーはこのデータベースまたはエリアに対して、要求を実行することが許可されていないと判断しました。
X'C1000201'	要求した状態と現行の許可状態が非互換です。データベースは、別のアクティブなまたは異常終了した IMS サブシステムによって許可されており、その許可状態が現行の許可要求と非互換です。
X'C1000202'	データベースが RECON データ・セットに登録されていません。また、AUTHLIST を不正にセットアップしている可能性もあります。高速機能の場合、DEDB 名およびエリア名を指定する必要があります。高速機能以外の場合は、エレメントに 8 文字の DB 名と 8 文字のブランクを指定します。
X'C1000203'	データベースには、これ以上の許可を禁止する対象としてマークが付けられています。その理由は、グローバル /DBR、グローバル /STOP、UPDATE DB STOP、または CHANGE.DB(NOAUTH) コマンドのいずれかです。
X'C1000204'	グローバル /DBDUMP または UPDATE DB STOP(UPDATES) コマンドにより、データベースが許可されるのは、要求した状態が「READ」または「READ-GO」である場合のみです。
X'C1000205'	データベースに、イメージ・コピーが必要というマークが付けられています。
X'C1000206'	データベースに、リカバリーが必要というマークが付けられています。
X'C1000207'	データベースに、バックアウトが必要というマークが付けられています。
X'C100020A'	データベースは、サブシステムに対して既に許可済みです。
X'C100020B'	DB 使用互換性評価プロセス中に無効なパラメーターが検出されました。RECON データ・セット内のデータベース・レコードが不正であることが考えられます。
X'C100020C'	無許可エラーのため、DBRC の現行の許可状態が無効です。
X'C100020D'	許可要求処理の試行中に、DBRC でエラーが発生しました。
X'C100020F'	データベースは、アクティブな IMS サブシステムに対して既に許可済みです。すべてのサブシステムを異常終了する必要があります。

表 87. AUTH 要求の戻りコードと理由コードに対応する APAUB\_RsnCode 値 (続き)

APAUB_RsnCode	意味
X'C1000214'	DB が DBRC に登録されておらず、すべてのデータベースの登録が必要な RECON FORCER オプションが有効であるため、DB 許可が失敗しました。
X'C100021E'	DB またはエリアが対象に含まれていますが、それを対象として含んでいる GSG に要求元のサブシステムがサインオンしていません。
X'C1000220'	HALDB は、初期設定する必要があります。
X'C1000221'	HALDB マスターの許可が試行されました。許可を要求できるのは、区画レベルのみです。
X'C1000223'	高位キーが定義されるまで、DB 区画は許可できません。HALDB マスターは区画選択ルーチンを使用しないため、キーが必要になります。
X'C1000224'	HALDB OLR 処理の間、イメージ・コピーは許可されません。
X'C1000225'	区画データベースの M-V DBDS へのロードが許可されていません。
X'C1000226'	HALDB OLR がアクティブであり、その HALDB OLR を IMS サブシステムが所有している場合、オフライン再編成は許可されません。
X'C1000228'	データベースが再編成中。
X'C1000229'	バッチ更新、リカバリー・ユーティリティー、および再編成ユーティリティーは、DB 静止の進行中には許可されません。
X'C100022A'	イメージ・コピー・ユーティリティーは、DB 静止が進行中であり、まだ DB 静止保留状態になっていない場合には許可されません。
X'C10003xx'	DBRC アプリケーションに適用されない許可の理由コードを受け取りました。xx は、返される FxFx 理由コードに対応する 16 進数です。この理由コードは、診断のために利用できます。

## AUTH 出力ブロックのマッピング

以下の図に、TYPE=AUTH 要求からの出力フォーマットを示します。

TYPE=AUTH 要求の出力ブロックは、DSPAPQHD によってマップされた標準ヘッダーから始まります。この出力ブロックのデータ部分は、DSPAPAUB によってマップされます。

以下の図に、TYPE=AUTH 要求からの出力フォーマットを示します。

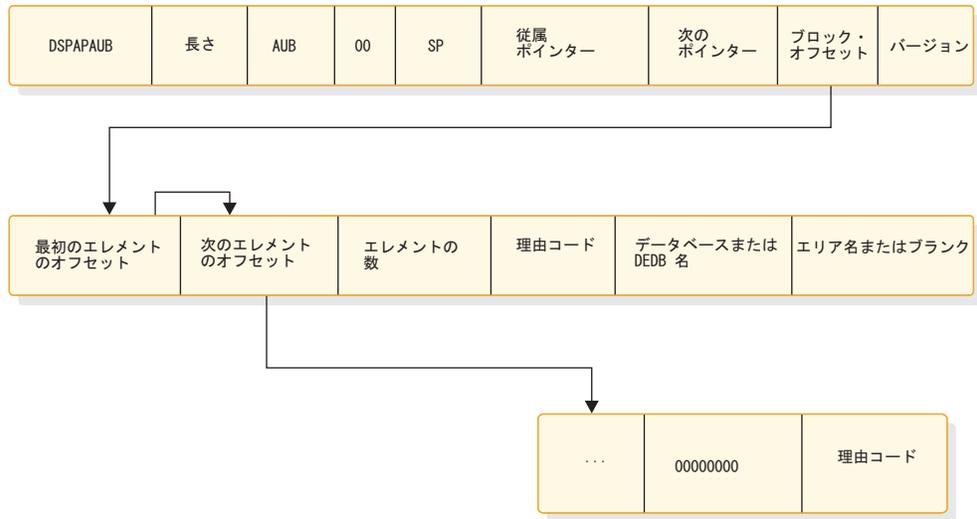


図 11. TYPE=AUTH の出力のフォーマット

## AUTH 出力ブロック

以下の例に、AUTH 要求によって返される出力ブロックを記載します。出力ブロックは、許可されたデータベースの配列を含み、AUTH 要求が正常に実行されたかどうかを示します。

### DSPAPAUB でマップされた出力ブロックの例

```
=====
DSPAPAUB
```

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	8	DSPAPAUB	AUTH/UNAUTH block
0	(0)	UNSIGNED	4	APAUB_OFFSET	Offset to first element
4	(4)	SIGNED	4	APAUB_ELCOUNT	Number of elements in list

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	24	APAUB_ELEMENT	
0	(0)	UNSIGNED	4	APAUB_OFFNEXT	Offset to next element
4	(4)	SIGNED	4	APAUB_RSNCODE	Reason code
8	(8)	CHARACTER	8	APAUB_DBNAME	Database or DEDB name
16	(10)	CHARACTER	8	APAUB_AREANAME	Area name or blanks

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPAUB	APAUB_EYECATCHER	

関連資料:

418 ページの『AUTH 要求のパラメーター』

## 第 29 章 DBRC コマンド要求 (COMMAND)

アプリケーション・プログラムから DBRC ユーティリティー・コマンドを発行するには、DSPAPI FUNC=COMMAND 要求を使用します。LIST コマンドを除くすべての DBRC コマンドは、この要求で受け入れられます。

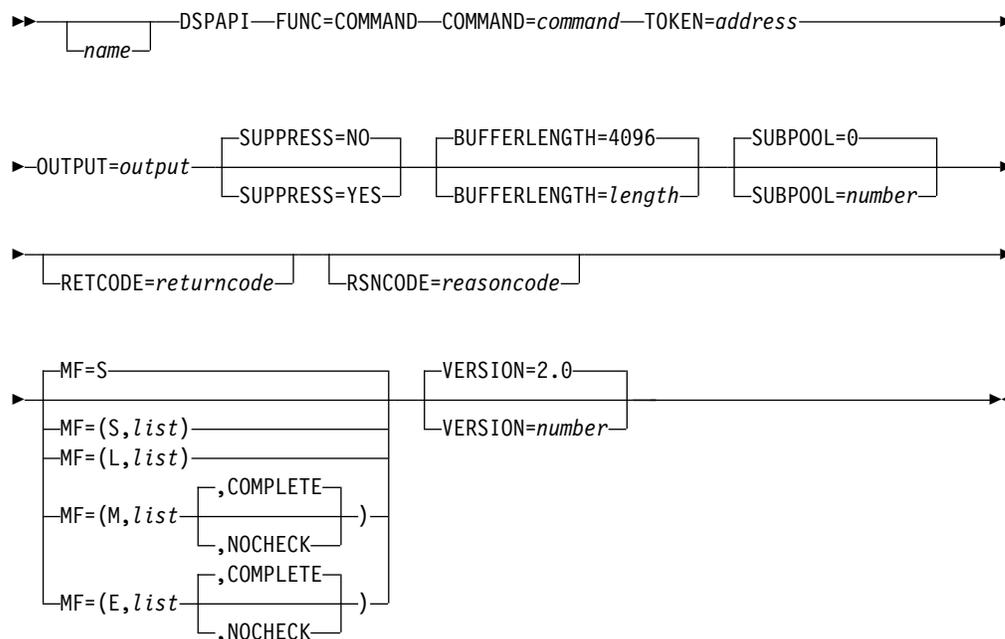
DBRC コマンド要求によって生成されるすべての出力は、SYSPRINT に送信される代わりに、API 出力ブロックで返されます。DBRC コマンド権限が使用可能な場合は、API を介して入力されるコマンドは、DBRC ユーティリティーを介して実行されるコマンドと同じ結果になります。IMS DD ステートメントが必要な場合があります。GENJCL コマンドを実行する場合、JCLPDS DD ステートメント (JCLPDS パラメーターで指定する DD 名) と JCLOUT DD ステートメント (JCLOUT パラメーターで指定する DD 名) の両方が必要です。

関連概念:

403 ページの『第 26 章 DBRC API』

### COMMAND 要求の構文

以下の構文図から、データベース・リカバリー管理 (DBRC) コマンド要求のフォーマットを理解できます。



### コマンド要求のパラメーター

以下の情報から、DBRC コマンド要求のパラメーターについて理解できます。各パラメーターは、コマンド要求の構文図との関連で説明されています。

名前 名前パラメーターを指定します。使用する場合は、1 桁目から始めます。

**COMMAND=***symbol* | (2 - 12)

実行する DBRC ユーティリティ・コマンドのアドレスを指定します。このコマンドは、ヘッダーとその後続く DBRC コマンドで構成されます。ヘッダーは、後続のコマンドの長さ (バイト単位) を含むフルワードです。DBRC コマンドは、DBRC コマンドの構文に準拠する必要があります。準拠していない場合、コマンドの続行は不可能です。区切り記号は、ブランク、コンマ、またはコメントです。これは、コマンド内の必要な任意の場所に使用でき、コマンドの先頭に使用することもできます。

**TOKEN=***symbol* | (2 - 12)

API トークンを受信する 4 バイト・フィールドのアドレスを指定します。このトークンは、この DSPAPI 要求に関連する後続のすべての要求に組み込む必要があります。

**OUTPUT=***output* | (2 - 12)

コマンドの情報を含むブロックを示すポインターを受け取るフィールドを指定します。

**SUPPRESS=** | **NO** | **YES**

コマンド出力を返すかどうかを指定します。SUPPRESS=YES を指定すると、コマンド出力は、コマンドが正常に完了しなかった場合 (戻りコードがゼロでない場合) のみ返されます。SUPPRESS=NO を指定すると、コマンド出力は常に返されます。SUPPRESS=NO がデフォルトです。

**BUFFERLENGTH=**4096 | *number* | (2 - 12)

コマンドの実行によって生成される出力を受け取るバッファの長さを指定します。レジスターを指定する場合、そのレジスターに目的の長さが含まれている必要があります。指定可能な最大長は 32760 です。必要に応じて、長さはダブルワード境界に切り上げられます。デフォルト値および最小値は 4096 です。

**SUBPOOL=** 0 | *number*

取得されるストレージのサブプール番号を指定します。このパラメーターを指定しない場合、FUNC=STARTDBRC 要求によって指定されたサブプールがデフォルトになります。それ以外の場合は、サブプール 0 がデフォルトになります。

**RETCODE=***return code* | (2-12)

シンボルとして指定する場合、戻りコードを受け取るために、ストレージのワードを示すラベルを指定します。レジスターとして指定する場合、戻りコードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RETCODE が指定されたかどうかにかかわらず、レジスター 15 は戻りコードを含んでいます。

**RSNCODE=***reason code* | (2-12)

シンボルとして指定する場合、理由コードを受け取るために、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、理由コードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RSNCODE が指定されたかどうかにかかわらず、レジスター 0 は理由コードを含んでいます。

## MF=S | L | M | E

要求のマクロ形式を指定します。

## VERSION=2.0 | number

この要求によって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターの処理は受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

デフォルトのバージョンは 2.0 です。

関連概念:

407 ページの『DSPAPI マクロのマクロ形式』

関連資料:

 DBRC コマンド構文 (コマンド)

 DBRC コマンドの概要 (コマンド)

---

## COMMAND 要求の戻りコードと理由コード

以下の表で、DBRC COMMAND 要求の戻りコードと理由コードを検索できます。この表には、各コードのタイプおよび説明も記載されています。

表 88. DSPAPI FUNC=COMMAND の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了しました。DBRC コマンドは、戻りコード 0 で完了しました。
部分的な正常完了	X'00000004'	X'C3000001'	DBRC コマンドは、ゼロ以外の戻りコードで完了しました。
重大エラー	X'0000000C'	X'C3000001'	LIST コマンドは許可されません。照会機能を使用してください。
	X'0000000C'	X'C9000001'	TOKEN が無効です。TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。
ストレージ・エラー	X'00000028'	X'C3000001'	コマンド・ブロック用のストレージの取得時にエラーが発生しました。
パラメーター・エラー	X'00000030'	X'C3000001'	COMMAND パラメーターが欠落しているか無効です。
	X'00000030'	X'C3000002'	コマンド長の誤り。ゼロより大きい値を指定する必要があります。
	X'00000030'	X'C3000003'	OUTPUT パラメーターが欠落しているか無効です。

表 88. DSPAPI FUNC=COMMAND の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'C3000004'	BUFFERLENGTH 値が無効です。0 以上かつ 32760 以下の値を指定する必要があります。
	X'00000030'	X'C9000001'	パラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000003'	RETCODE アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000005'	OUTPUT アドレスが無効です。API OUTPUT を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C900000A'	要求された機能 (FUNC) に対して、不正な VERSION 値が指定されました。
	X'00000030'	X'C9000019'	COMMAND アドレスが無効です。API COMMAND を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

## コマンド出力ブロックのマッピング

以下の例に、DBRC コマンド要求による出力ブロックのマッピングを示します。これらの例から、DSPAPCMD 出力ブロックのヘッダーがどのように構成されているか、そして出力がどのように表示されるかを理解することができます。

以下の図は、コマンド出力ブロックのマッピングを示します。

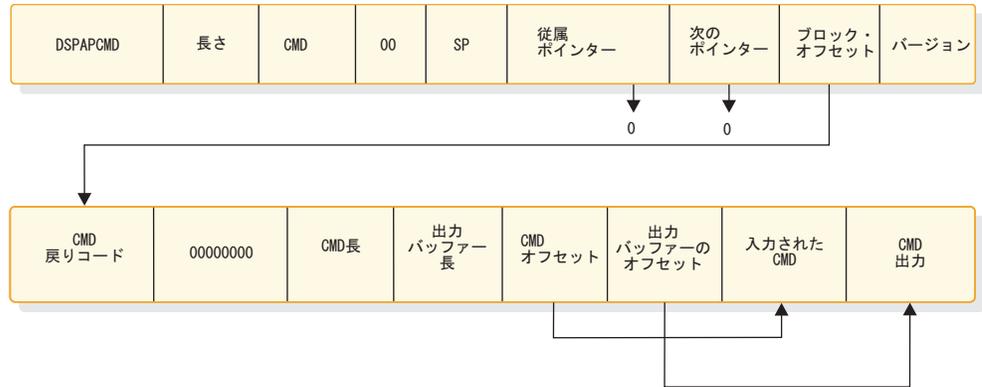


図 12. コマンド出力ブロックのマッピング

## DSPAPCMD によってマップされたストレージ・ブロックの例

各ストレージ・ブロックは、DSPAPQHD によってマップされた標準ヘッダーから始まります。この出力ブロックのデータ部分は、以下の例に示されているように、DSPAPCMD によってマップされます。

DSPAPCMD

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	24	DSPAPCMD	
0	(0)	SIGNED	4	APCMD_RETCODE	DBRC command return code
4	(4)	SIGNED	4	*	Reserved
8	(8)	SIGNED	4	APCMD_CMDLEN	Length of command entered
12	(C)	SIGNED	4	APCMD_BUFFLEN	Length of command output
16	(10)	UNSIGNED	4	APCMD_CMDOFF	Offset to command buffer
20	(14)	UNSIGNED	4	APCMD_BUFFOFF	Offset to command output

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	*	APCMD_COMMAND	Command as entered

## DSPAPCMD 出力ブロック・ヘッダーの例

以下の例では、DSPAPCMD 出力ブロック・ヘッダーに要求の状況が含まれます。このヘッダーが、コマンドのエコーおよびコマンド出力を返します。以下の例は、DBRC コマンド出力の DSECT について示したものです。

Each output line is mapped by the following structure

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	5	APCMD_OUTPUT_LINES	
0	(0)	SIGNED	2	APCMD_OUTPUT_LINELEN	Length of line
2	(2)	SIGNED	2	*	Reserved
4	(4)	CHAR VARY	1	APCMD_OUTPUT_DATA	Output data

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPCMD	APCMD_EYECATCHER	

関連資料:

461 ページの『DBDS 照会要求 (TYPE=DBDS)』

---

## 第 30 章 DBRC 照会要求

DBRC 照会要求 (DSPAPI FUNC=QUERY) を TYPE パラメーターと一緒に使用することで、RECON データ・セットから、以下のさまざまなタイプの情報を取得できます。

- バックアウト (TYPE=BACKOUT)
- データベース (TYPE=DB) - この QUERY のバリエーションでは、以下のデータベースに関するデータベース登録情報と状況情報が返されます。
  - 全機能データベース
  - 高速機能データベース
  - HALDB データベース
  - DBDS 情報またはエリア情報、およびDBDS またはエリアごと (割り振り、イメージ・コピー、リカバリー、および再編成) のサポートするリカバリー関連情報
- データベース区画化 (TYPE=PART)
- DBDS またはエリア情報 (TYPE=DBDS)
- 以下のグループ・タイプに関するグループおよびメンバー情報
  - 変更累積 (TYPE=CAGROUP)。CA 実行情報も戻されます。
  - DBDS (TYPE=DBDSGROUP)
  - データベース (TYPE=DBGROUP)
  - リカバリー (TYPE=RECOVGROUP)
  - グローバル・サービス・グループ (TYPE=GSGROUP)
- ログ、リカバリーおよびシステム・ログ・データ・セット (TYPE=LOG)
- オンライン・ログ・データ・セット (TYPE=OLDS)
- RECON 状況 (TYPE=RECON) - この QUERY のバリエーションでは、RECON ヘッダー情報、および RECON 構成の状況が返されます。
- サブシステム (TYPE=SUBSYS)

照会要求からの出力は時間に整合性があり、要求の処理中は RECON データ・セットへのアクセスは制限されます。

並列 RECON アクセスを使用可能にすると、照会 API 要求は、LIST.xxx STATIC を指定した場合と同様の出力を返します (DBRC による 照会要求の処理中には、RECON データ・セットがアクセスできなくなります)。

関連概念:

403 ページの『第 26 章 DBRC API』

関連資料:

432 ページの『バックアウト照会要求 (TYPE=BACKOUT)』

437 ページの『データベース照会要求 (TYPE=DB)』

491 ページの『HALDB 区画照会要求 (TYPE=PART)』

- 461 ページの『DBDS 照会要求 (TYPE=DBDS)』
- 467 ページの『グループ照会要求 (TYPE=\*GROUP)』
- 477 ページの『ログ照会要求 (TYPE=LOG)』
- 486 ページの『OLDS 照会要求 (TYPE=OLDS)』
- 499 ページの『RECON 状況照会要求 (TYPE=RECON)』
- 504 ページの『サブシステム照会要求 (TYPE=SUBSYS)』

## 照会要求からの出力

要求された情報は、呼び出し側アプリケーションに、ストレージ内の 1 つ以上のブロックのチェーンとして返されます。このチェーンの先頭を示すポインタは、照会要求の OUTPUT パラメーターによって指定されたエリアに戻されます。

これらのブロックのストレージは、呼び出し側アプリケーションによって事前割り振りされません。DBRC はこれらのブロック用に専用ストレージを獲得します。呼び出し側アプリケーションは、バッファ解放要求 (DSPAPI FUNC=RELBUF) を使用してこのストレージを解放する責任を負います。

各ストレージ・ブロックは、DSPAPQHD によってマップされた標準ヘッダーで始まります。

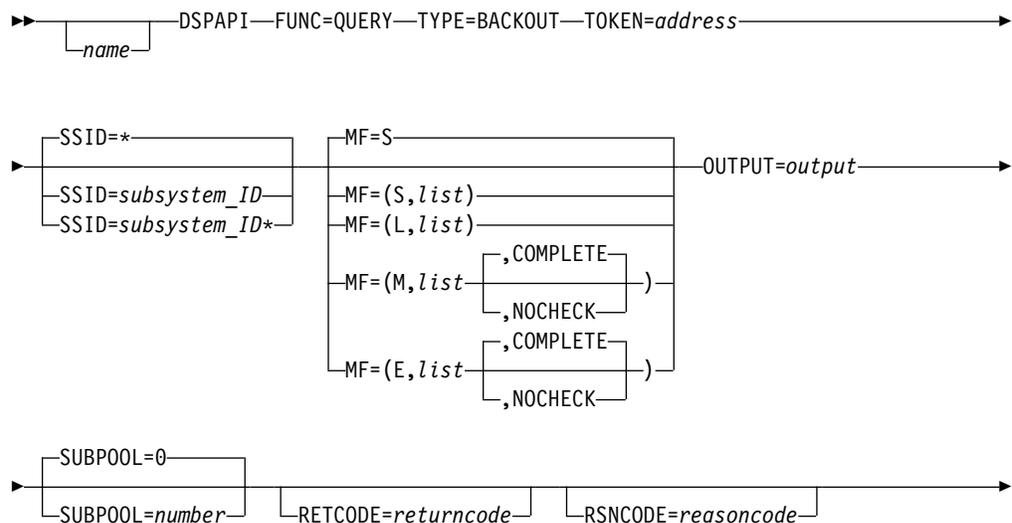
関連概念:

- 403 ページの『第 26 章 DBRC API』

## バックアウト照会要求 (TYPE=BACKOUT)

特定のサブシステムまたはすべてのサブシステムのバックアウト情報を RECON データ・セットから取得するには、バックアウト照会 (DSPAPI FUNC=QUERY TYPE=BACKOUT) 要求を使用します。

### TYPE=BACKOUT 照会要求の構文





## TYPE=BACKOUT 照会要求のパラメーター

名前 使用する場合は、1 桁目から始めます。

### TYPE=BACKOUT

バックアウト情報が要求されることを指定します。

### TOKEN=*symbol* | (2 - 12)

FUNC=STARTDBRC 要求で戻された 4 バイト・フィールドのアドレスを指定します。

### SSID=\_ | *symbol* | *symbol*\* | (2 - 12)

照会されるバックアウトのサブシステム名を指定します。ワイルドカード・キーワード \* (アスタリスク) のみを使用して、すべてのグループに関する情報を要求することができます (SSID=\*, これはデフォルトです)。また、これを名前の最後に追加して、そのパターンと一致する名前を持つすべてのサブシステムを照会することもできます。この場合、アスタリスクの前には、英字が少なくとも 1 つ必要です。

### MF=S | L | M | E

要求のマクロ形式を指定します。

### OUTPUT=*output* | (2 - 12)

バックアウト情報ブロックの最初のブロックを示すポインターを受け取るフィールドを指定する、必須パラメーターです。

出力が作成されなかった場合、出力アドレスはゼロです。この状態は、RECON 内に要求を満たす情報がない場合、または出力を作成できるようになる前にエラーが発生した場合に起こります。

出力ブロックのストレージは、呼び出し元によって事前割り振りされません。DBRC はこれらのブロック用に指定されたサブプールからストレージを取得します。呼び出し元は、バッファ解放サービス (DSPAPI FUNC=RELBUF) を使用してこのストレージを解放し、戻された出力アドレスを指定する必要があります。

### SUBPOOL= 0 | *number*

取得されるストレージのサブプール番号を指定します。このパラメーターを指定しない場合、FUNC=STARTDBRC 要求によって指定されたサブプールがデフォルトになります。それ以外の場合は、サブプール 0 がデフォルトになります。

### RETCODE=*returncode* | (2-12)

シンボルとして指定する場合、戻りコードを受け取るために、ストレージのワードを示すラベルを指定します。レジスターとして指定する場合、戻りコードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RETCODE が指定されたかどうかにかかわらず、レジスター 15 は戻りコードを含んでいます。

### RSNCODE=*reasoncode* | (2-12)

シンボルとして指定する場合、理由コードを受け取るために、そのシンボル

はストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、理由コードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RSNCODE が指定されたかどうかにかかわらず、レジスター 0 は理由コードを含んでいます。

#### VERSION=2.0 | *number*

このマクロによって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターはマクロ処理によって受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

FUNC=QUERY TYPE=BACKOUT 要求の有効なバージョン番号は、1.0 および 2.0 です。

### TYPE=BACKOUT 照会要求の戻りコードと理由コード

表 89. TYPE=BACKOUT 照会要求の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
警告	X'00000008'	X'D87000001'	バックアウト・レコードが存在しません。
重大エラー	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。
	X'0000000C'	X'D8700100'	セキュリティ・エラー。SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーは要求の実行を許可されていないと判断しました。
ストレージ・エラー	X'00000028'	X'D87000001'	BACKOUT ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D91000001'	要求の処理中にエラーが発生しました。DBRC はこの時点までに取得したストレージを解放します。しかし、ストレージを解放しようとして別のエラーが検出されました。
内部エラー	X'0000002C'	X'D8000001'	RECON を開くときに障害が発生しました。
	X'0000002C'	X'D8700001'	最初のバックアウト・レコードまたは指定されたバックアウト・レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8700002'	次のバックアウト・レコードの検索時に障害が発生しました。

表 89. TYPE=BACKOUT 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
パラメーター・エラー	X'00000030'	X'C9000001'	API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN フィールド・アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000003'	RETCODE フィールド・アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE フィールド・アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000005'	OUTPUT フィールド・アドレスが無効です。OUTPUT アドレスを含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000010'	SSID フィールド・アドレスが無効です。SSID を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'D80000001'	OUTPUT パラメーターが欠落しているか無効です。
	X'00000030'	X'D80000002'	TYPE パラメーターに指定された値が無効です。
	X'00000030'	X'D8700100'	ワイルドカードを使用するとき、アスタリスクの前に英字が少なくとも 1 つ必要です。
X'00000030'	X'D8700101'	ワイルドカードを使用するとき、最後の文字はアスタリスクである必要があります。	

## TYPE=BACKOUT QUERY 要求の出力

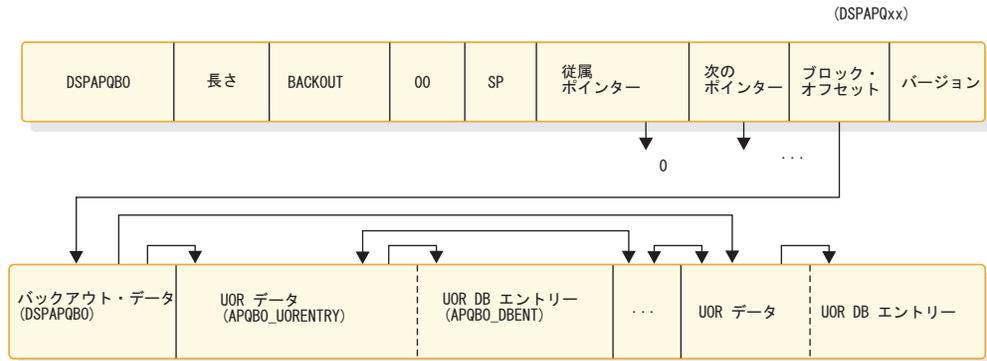


図 13. QUERY TYPE=BACKOUT 出力要求からの出力のフォーマット

### DSPAPQBO の DSECT

以下の例のサンプル DSECT が、ストレージ・ブロックのフィールドとそれらのフィールドの相互関係を詳細に説明します。

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	48	DSPAPQBO	
0	(0)	CHARACTER	8	APQBO_SSID	Subsystem identifier
8	(8)	UNSIGNED	4	APQBO_FIRSTUOR	Offset of first UOR entry
12	(C)	UNSIGNED	4	APQBO_LASTUOR	Offset of last UOR entry
16	(10)	CHARACTER	12	APQBO_TIMEFIRST	Earliest UOR time
28	(1C)	CHARACTER	12	APQBO_TIMELAST	Latest UOR time
40	(28)	BIT(8)	1	APQBO_FLAGS	Backout flags
		1... ....		APQBO_SAVER	SAVUOR call during restart
41	(29)	CHARACTER	3	*	Reserved
44	(2C)	SIGNED	4	APQBO_UORCOUNT	Number of UORs

```
=====
The following structure maps the unit of recovery entries.
There is one such entry for each unit of recovery (that is, there
are apqbo_UORcount entries). Each unit of recovery entry
contains the offset within the backout block to the previous
and following entries. Field apqbo_PrevUOR is the offset of
the previous entry and apqbo_NextUOR is the offset of the
following entry. For the first unit of recovery (UOR) entry,
apqbo_PrevUOR will be zero. Similarly, apqbo_NextUOR will be
zero for the last entry. Addressability to the first UOR entry
is given by:
```

```
    rfy apqbo_UORentry based(addr(DSPAPQBO) + apqbo_FirstUOR)
```

```
Addressability to the last UOR entry is given by:
```

```
    rfy apqbo_UORentry based(addr(DSPAPQBO) + apqbo_LastUOR)
```

```
Addressability to the next UOR entry, if one exists (that is:
apqbo_NextUOR not equal to 0), is given by:
```

```
    rfy apqbo_UORentry based(addr(DSPAPQBO) + apqbo_NextUOR)
```

```
Similarly, to address the previous entry (when apqbo_PrevUOR
not equal to 0):
```

```
    rfy apqbo_UORentry based(addr(DSPAPQBO) + apqbo_PrevUOR)
```

```
Once addressability has been established to a UOR entry,
addressability to the ith database for this UOR is given by:
```

```
    rfy apqbo_DBent
        based(addr(apqbo_UORentry) + apqbo_DBOffset
            + (i-1) apqbo_DBLength)
```

```
=====
```

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	64	APQBO_UOENTRY	Unit of Recovery entry
0	(0)	CHARACTER	64	APQBO_PREFIX	Prefix section
0	(0)	UNSIGNED	4	APQBO_NEXTUOR	Offset of next UOR entry
4	(4)	UNSIGNED	4	APQBO_PREVUOR	Offset of previous UOR entry
8	(8)	UNSIGNED	4	APQBO_DBOFFSET	Offset to DB entries
12	(C)	CHARACTER	12	APQBO_UORTIME	Time stamp for this UOR
24	(18)	CHARACTER	8	APQBO_UORPSB	PSB name
32	(20)	BIT(16)	2	APQBO_UORFLAGS	
		1... ....		APQBO_DEFBO	Deferred backout - dynamic backout failure
		.1.. ....		APQBO_INFLT	Inflight UOR
		..1. ....		APQBO_INDOU	Indoubt UOR
		...1 ....		APQBO_BMP	BMP UOR
		.... 1...		APQBO_BOCAN	BBO identified candidate
		.... .1..		APQBO_COLDN	Cold start ended for UOR
		.... ..1.		APQBO_BBOK	Backed out OK by BBO
		.... ...1		APQBO_CMD	UOR modified by command
33	(21)	1... ....		APQBO_BATCH	Batch IMS UOR
34	(22)	CHARACTER	6	*	Reserved
40	(28)	CHARACTER	16	APQBO_RTOKN	Recovery token
40	(28)	CHARACTER	8	APQBO_RTSSID	SSID for this token
48	(30)	CHARACTER	8	APQBO_UORID	Unique UOR ID
56	(38)	SIGNED	4	APQBO_DBCOUNT	Number of DBs for this UOR
60	(3C)	UNSIGNED	2	APQBO_DBLENGTH	Length of each DB entry
62	(3E)	CHARACTER	2	*	Reserved

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	16	APQBO_DBENT	Database entry
0	(0)	CHARACTER	8	APQBO_DBNAME	Database name
8	(8)	BIT(8)	1	APQBO_DBFLAGS	Flags
		1... ....		APQBO_DBOUT	UOR backed out for this DB
		.1.. ....		APQBO_DBDEF	Dyn backout failure this DB
9	(9)	CHARACTER	7	*	Reserved

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQBO	APQBO_EYECATCHER	

関連概念:

407 ページの『DSPAPI マクロのマクロ形式』

関連資料:

431 ページの『第 30 章 DBRC 照会要求』

## データベース照会要求 (TYPE=DB)

1 つ以上の登録済みデータベースに関する情報を RECON から取得するには、データベース照会要求 (DSPAPI FUNC=QUERY TYPE=DB) を使用します。

この情報には、以下のデータベース・タイプが含まれます。

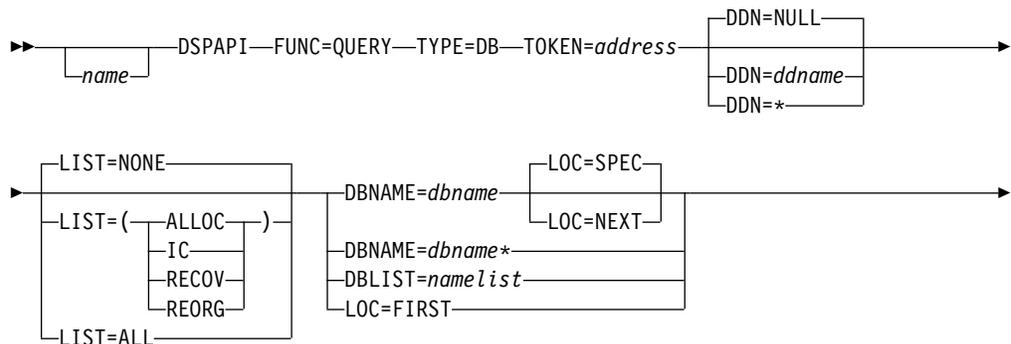
- 全機能
- 高速機能 DEDB
- HALDB (HALDB マスターとそのすべての区画の情報を戻します)

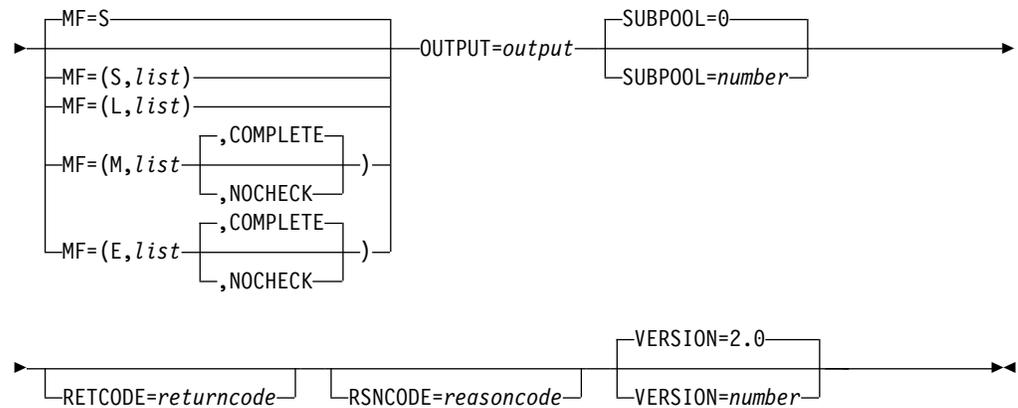
オプションで、データ・セットとエリアの情報を要求することもできます。この情報を要求する場合、割り振り、イメージ・コピー、リカバリー、および再編成の情報を含むデータ・セットまたはエリアのリカバリー関連情報も要求できます。

以下のサブセクションがあります。

- 『TYPE=DB 照会要求の構文』
- 439 ページの『TYPE=DB 照会要求のパラメーター』
- 441 ページの『TYPE=DB 照会要求の戻りコードと理由コード』
- 445 ページの『TYPE=DB 照会要求の出力』
- 445 ページの『全機能の出力』
- 446 ページの『DSPAPQDB の DSECT』
- 447 ページの『DSPAPQSL の DSECT』
- 448 ページの『高速機能 DEDB の出力』
- 448 ページの『DSPAPQFD の DSECT』
- 449 ページの『DSPAPQAR の DSECT』
- 450 ページの『DSPAPQEL の DSECT』
- 451 ページの『HALDB (マスターおよびすべての区画) の出力』
- 451 ページの『DSPAPQHB の DSECT』
- 452 ページの『DSPAPQHP の DSECT』
- 454 ページの『DBDS の出力』
- 455 ページの『DSPAPQDS の DSECT』
- 456 ページの『リカバリー情報 (RCVINFO) の出力』
- 457 ページの『DSPAPQRI の DSECT』
- 457 ページの『DSPAPQAL の DSECT』
- 457 ページの『DSPAPQIC の DSECT』
- 459 ページの『DSPAPQRV の DSECT』
- 459 ページの『DSPAPQRR の DSECT』
- 460 ページの『「データベース未検出」の出力』
- 460 ページの『DSPAPQNF の DSECT』

## TYPE=DB 照会要求の構文





## TYPE=DB 照会要求のパラメーター

名前 使用する場合は、1 桁目から始めます。

### TYPE=DB

1 つ以上のデータベースに関する RECON 情報が要求されることを指定します。

### TOKEN=address | (2 - 12)

FUNC=STARTDBRC 要求で戻された 4 バイト・フィールドのアドレスを指定します。

### DBNAME=symbol | symbol\* | (2 - 12)

情報が照会されるデータベースの名前を指定します。この名前には、全機能データベース、高速機能 DEADB、または HALDB の名前を指定できます。名前の最後にワイルドカード・キーワード \* (アスタリスク) を使用すると、そのパターンと一致する名前を持つデータベースを照会できます。この場合、アスタリスクの前には、英字が少なくとも 1 つ必要です。

LOC=FIRST を指定しない場合は、DBNAME または DBLIST を指定する必要があります。

### DBLIST=namelist | (2 - 12)

情報が照会されるデータベース名のリストを指定します。リストのそれぞれの名前には、全機能データベース、高速機能 DEADB、または HALDB の名前を指定できます。

このリストは、見出しと、8 文字からなる 1 つ以上のリスト項目で構成されます。このヘッダーは、リスト内のエンタリー数を含む 1 つのフルワードで成り立ちます。名前項目は左寄せされ、ブランクで埋め込まれます。

LOC=FIRST を指定しない場合は、DBNAME または DBLIST を指定する必要があります。

### DDN=ddname | (2 - 12) | NULL

データ・セットまたはエリアの DD 名を指定します。アスタリスク (\*) を指定すると、データベースのすべての DBDS またはエリアに関する情報を返すことができます。DBLIST によって、または HALDB を指定する DBNAME によって特定の DD 名を指定すると、その特定の DD 名は無視され、DDN=\* が指定された場合と同様に扱われます。

DDN=NULL を指定すると、DBDS またはエリアの情報は戻されません。  
DDN=NULL はデフォルトです。

**LIST=(ALLOC | IC | RECOV | REORG) | ALL | LIST=NONE**

指定した DBDS またはエリアの照会出力に組み込むサポート情報のタイプ (1 つ以上) を指定します。

DDN を指定しない場合、データベース内のすべての DBDS またはエリアに関してこの情報が戻されます。

リストには、1 つ以上の特定の値をコンマで区切って組み込むことができます。それらの値には、ALLOC (割り振りレコード)、IC (イメージ・コピー・レコード)、RECOV (リカバリー・レコード)、または REORG (再編成レコード) があります。LIST=ALL を指定すると、すべてのサポート情報が要求されます。

HALDB オンライン変更処理に関する情報を表示するには、LIST(REORG) を指定します。変更処理に関する情報には、他のすべての再編成処理に関する情報が含まれています。

LIST=NONE を指定すると、サポート情報は戻されません。LIST=NONE はデフォルトです。

**LOC=FIRST | NEXT | SPEC**

その要求が、RECON 内で定義済みの指定されたデータベース、最初のデータベース、または次のデータベースを対象とすることを指定します。

DBNAME=*dbname* は、LOC=NEXT または LOC=SPEC を指定する場合には必須ですが、LOC=FIRST の場合には指定できません。

データベースは英数字順です。次のデータベースが、DBNAME パラメーターに指定されたデータベース名と必ずしも同じタイプとは限りません。

DBNAME パラメーターの値は検索のベースとして使用されますが、RECON に登録済みのデータベースの名前でなくともかまいません。

LOC=SPEC はデフォルトです。

**MF=S | L | M | E**

要求のマクロ形式を指定します。

**OUTPUT=*output* | (2 - 12)**

データベース情報ブロック (おそらくチェーンになっている) の最初のブロックを示すポインターを受け取るフィールドを指定します。

出力が作成されなかった場合、出力アドレスはゼロです。この状態は、RECON 内に要求を満たす情報がない場合、または出力を作成できるようになる前にエラーが発生した場合に起こります。

出力ブロックのストレージは、呼び出し元によって事前割り振りされません。DBRC はこれらのブロック用に指定されたサブプールからストレージを取得します。呼び出し元は、バッファ解放サービス (DSPAPI FUNC=RELBUF) を使用してこのストレージを解放し、戻された出力アドレスを指定する必要があります。

**SUBPOOL= 0 | *number***

取得されるストレージのサブプール番号を指定します。このパラメーターを

指定しない場合、FUNC=STARTDBRC 要求によって指定されたサブプールがデフォルトになります。それ以外の場合は、サブプール 0 がデフォルトになります。

**RETCODE=returncode | (2-12)**

シンボルとして指定する場合、戻りコードを受け取るために、ストレージのワードを示すラベルを指定します。レジスターとして指定する場合、戻りコードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RETCODE が指定されたかどうかにかかわらず、レジスター 15 は戻りコードを含んでいます。

**RSNCODE=reasoncode | (2-12)**

シンボルとして指定する場合、理由コードを受け取るために、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、理由コードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RSNCODE が指定されたかどうかにかかわらず、レジスター 0 は理由コードを含んでいます。

**VERSION=2.0 | number**

この要求によって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターはマクロ処理によって受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

FUNC=QUERY TYPE=DB 要求の有効なバージョン番号は、1.0 および 2.0 です。

**TYPE=DB 照会要求の戻りコードと理由コード**

表 90. TYPE=DB 照会要求の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
部分的な正常完了	X'00000004'	X'D8200001'	DBLIST 内の 1 つ以上のデータベースが RECON に登録されていません。検出されなかったデータベースごとに、DSPAPQNF の目印が付いた「DB 未検出」データ・ブロックが作成されます。このブロックは、戻されるブロックのチェーンに含まれます。
警告	X'00000008'	X'D8220001'	RECON データ・セットに HALDB の区画が登録されていません。情報ブロックは戻されません。
	X'00000008'	X'D8200001'	DBLIST 内のデータベースがどれも RECON に登録されていません。情報ブロックは戻されません。

表 90. TYPE=DB 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000008'	X'D8220002'	指定された区画が RECON に登録されていません。情報ブロックは戻されません。
	X'00000008'	X'D8200002'	指定されたデータベースが RECON に登録されていません。情報ブロックは戻されません。要求で LOC=NEXT が指定された場合、RECON に登録されたデータベース・リストの終わりに到達しました。
	X'00000008'	X'D8200003'	指定されたワイルドカードの名前パターンを持つ DB が、RECON に登録されていません。情報ブロックは戻されません。
	X'00000008'	X'D8210002'	指定された DBDS またはエリアが RECON に登録されていません。情報ブロックは戻されません。
	X'00000008'	X'D8220003'	指定された区画より前の区画が、RECON データ・セットに存在しません。情報ブロックは戻されません。
	X'00000008'	X'D8220004'	指定された区画より後の区画が、RECON データ・セットに存在しません。情報ブロックは戻されません。
重大エラー	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。
	X'0000000C'	X'D8200100'	セキュリティエラー。SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーは要求の実行を許可されていないと判断しました。
ストレージ・エラー	X'00000028'	X'D8200001'	IMSDB ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8200002'	HALDB ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8200003'	FPDEDB ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8200004'	「DB 未検出」ブロック (DBNOTFND) 用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210001'	DBDS ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210002'	AREA ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210003'	RCVINFO ブロック用のストレージの取得時にエラーが発生しました。

表 90. TYPE=DB 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000028'	X'D8210004'	ALLOC ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210005'	IC ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210006'	REORG ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210007'	RECOV ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8220001'	PART ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D9100001'	要求の処理中にエラーが発生しました。DBRCはこの時点までに取得したストレージを解放します。しかし、ストレージを解放しようとして別のエラーが検出されました。
内部エラー	X'0000002C'	X'D8000001'	RECON を開くときに障害が発生しました。
	X'0000002C'	X'D8200001'	DBLIST 処理中に DB レコードの検索で障害が発生しました。
	X'0000002C'	X'D8200002'	単一データベース要求の処理中に DB レコードの検索で障害が発生しました。
	X'0000002C'	X'D8200003'	ワイルドカードを使用したデータベース要求の処理中に DB レコードの検索で障害が発生しました。
	X'0000002C'	X'D82021xx'	内部照会 DBDS 呼び出しが RC=X'30' RSN=X'D82100xx' のパラメーター・エラーを返しました。
	X'0000002C'	X'D82022xx'	内部照会 PART 呼び出しが RC=X'30' RSN=X'D82200xx' のパラメーター・エラーを返しました。
	X'0000002C'	X'D8210001'	指定された DBDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210002'	次の DBDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210003'	最初の DBDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210004'	最初のエリアの AUTH レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210005'	最初の ALLOC レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210006'	次の ALLOC レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210007'	最初の IC レコードの検索時に障害が発生しました。

表 90. TYPE=DB 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'0000002C'	X'D8210008'	次の IC レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210009'	最初の REORG レコードの検索時に障害が発生しました。
	X'0000002C'	X'D821000A'	次の REORG レコードの検索時に障害が発生しました。
	X'0000002C'	X'D821000B'	最初の RECOV レコードの検索時に障害が発生しました。
	X'0000002C'	X'D821000C'	次の RECOV レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8220001'	最初の HALDB 区画レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8220002'	処理中の HALDB 区画に関連付けられた DB レコードを検索しようとして障害が発生しました。
	X'0000002C'	X'D8220003'	次の HALDB 区画レコードの検索時に障害が発生しました。
パラメータ ・エラー	X'00000030'	X'C9000001'	API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN フィールド・アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000003'	RETCODE フィールド・アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE フィールド・アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000005'	OUTPUT フィールド・アドレスが無効です。OUTPUT アドレスを含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000008'	DBNAME または DBLIST アドレスが無効です。DBNAME または DBLIST を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

表 90. TYPE=DB 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'C9000009'	DDN アドレスが無効です。DDN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'D8000001'	OUTPUT パラメーターが欠落しているか無効です。
	X'00000030'	X'D8200001'	LOC パラメーターは DBLIST の場合は指定できません。
	X'00000030'	X'D8220001'	DBNAME または PARTNAME は必須です。
	X'00000030'	X'D8000002'	TYPE パラメーターに指定された値が無効です。
	X'00000030'	X'D8200002'	DBNAME パラメーターは LOC=FIRST の場合は指定できません。
	X'00000030'	X'D8220002'	LOC=FIRST   LAST は DBNAME の場合は必須です。
	X'00000030'	X'D8200003'	DBNAME パラメーターは LOC=NEXT の場合は必須です。
	X'00000030'	X'D8220003'	LOC=FIRST   LAST は PARTNAME の場合は指定できません。
	X'00000030'	X'D8200004'	DBNAME または DBLIST は必須です。
	X'00000030'	X'D8200005'	DBLIST 内のデータベースの数がゼロです。
	X'00000030'	X'D8200006'	HALDB 区画に関してデータベース情報が要求されています。DBNAME または DBLIST に区画名が含まれています。
	X'00000030'	X'D8200007'	DBNAME のワイルドカードは LOC=NEXT の場合は指定できません。
	X'00000030'	X'D8200100'	ワイルドカードを使用するとき、アスタリスクの前に英字が少なくとも 1 つ必要です。
	X'00000030'	X'D8200101'	ワイルドカードを使用するとき、最後の文字はアスタリスクである必要があります。

## TYPE=DB 照会要求の出力

以下の図に、QUERY TYPE=DB 要求からの出力のフォーマットを示します。図の後のサンプル DSECT では、ストレージ・ブロックのフィールドとそれらのフィールドの関係を詳細に説明します。

### 全機能の出力

以下に、DSPAPQDB ブロックのフィールドを示します。

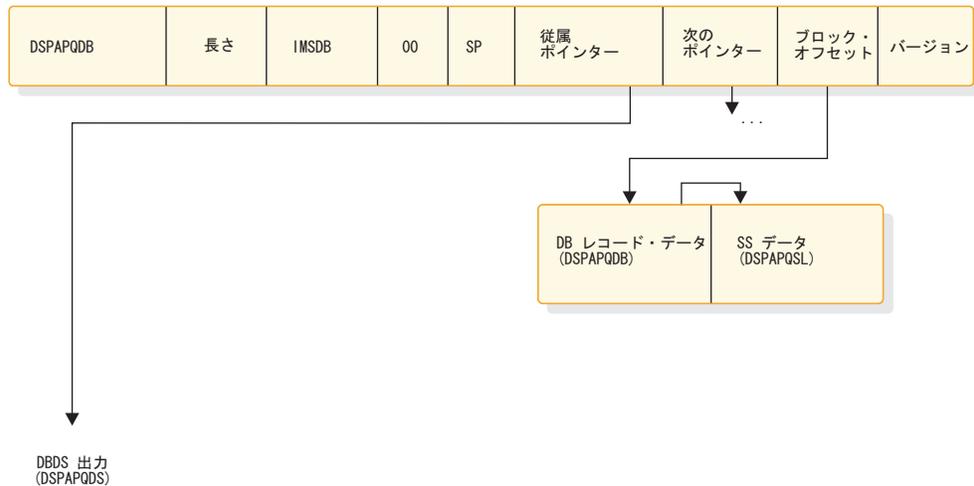


図 14. QUERY TYPE=DB (全機能) の出力のフォーマット

DBDS 情報は、DDN が指定されている場合にのみ戻されます。

### DSPAPQDB の DSECT

以下の例で、図 14 に示す DSPAPQDB ブロックに含まれるフィールドについて説明します。

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	96	DSPAPQDB	
0	(0)	CHARACTER	8	APQDB_DBNAME	Database name
8	(8)	UNSIGNED	4	APQDB_SSLIST	Offset to SS list (DSPAPQSL), zero is no SS auth'd
12	(C)	SIGNED	4	* (3)	Reserved
24	(18)	UNSIGNED	2	APQDB_IRCNT	IC receive needed counter
26	(1A)	BIT(8)	1	APQDB_AUFLAG	Authorization flags
		1... ..		APQDB_BKFLG	Backout needed flag
		.1.. ..		APQDB_PAFLG	Prohibit authorization
		..1. ....		APQDB_RDFLG	Read only SS auth
		...1 ....		APQDB_NONRV	nonrecoverable
		.... 1...		APQDB_DBREORGI	Reorg intent
		.... .1..		APQDB_DBQUI	Quiesce in progress
		.... ..1.		APQDB_DBQUIH	Quiesce held
27	(1B)	CHARACTER	5	APQDB_IRLMAU	IRLMID of auth SS
32	(20)	SIGNED	2	APQDB_RCVCTR	Recovery needed count
34	(22)	SIGNED	2	APQDB_ICCTR	IC needed count
36	(24)	SIGNED	2	APQDB_ICRECCTR	IC recommended counter
38	(26)	UNSIGNED	1	APQDB_SHRLVL	Share level of DB
39	(27)	UNSIGNED	1	APQDB_HELDAU	Held auth state
		1... ..		APQDB_HAUBIT	High order bit flag
40	(28)	UNSIGNED	2	APQDB_DMBNUM	Global DMB number
42	(2A)	SIGNED	2	APQDB_SSNUM	# of SS auth DB
44	(2C)	UNSIGNED	2	APQDB_SSENTLEN	Length of each SS entry
46	(2E)	UNSIGNED	1	APQDB_CACCSS	Access state for chg auth
47	(2F)	UNSIGNED	1	APQDB_CANCDD	Encode state for chg auth
48	(30)	UNSIGNED	1	APQDB_CAHELD	Held state for chg auth
49	(31)	CHARACTER	3	*	Reserved
52	(34)	UNSIGNED	2	APQDB_EQECNT	Total EQE count
54	(36)	BIT(16)	2	APQDB_RSRFLG	Flags
		1... ..		APQDB_RCVTRK	Recovery level tracking
		.1.. ..		APQDB_TRKSPN	Tracking is suspended
		..1. ....		APQDB_PURBIT	Suspended by time
		...1 ....		APQDB_ICNDIS	IC needed disabled option

56	(38)	CHARACTER	8	APQDB_GSGNAME	Global Service Group name
64	(40)	UNSIGNED	4	APQDB_USID	Last alloc USID
68	(44)	UNSIGNED	4	APQDB_AUSID	Last authorized USID
72	(48)	UNSIGNED	4	APQDB_RUSID	Last received USID
76	(4C)	UNSIGNED	4	APQDB_HUSID	Hardened by tracker USID
80	(50)	UNSIGNED	4	APQDB_RNUSID	Receive needed USID
84	(54)	CHARACTER	8	APQDB_RECOVGRP	Recovery Group name
92	(5C)	CHARACTER	4	*	Reserved

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQDB	APQDB_EYECATCHER	

### DSPAPQSL の DSECT

以下の図で、SS データ (DSPAPQSL) に含まれるフィールドについて説明します。

DSPAPQSL

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	16	DSPAPQSL	
0	(0)	CHARACTER	16	APQSL_SSINFO	Subsystem list entry
0	(0)	CHARACTER	8	APQSL_SSNAME	Subsystem ID
8	(8)	UNSIGNED	1	APQSL_ACCESS	Access intent
9	(9)	UNSIGNED	1	APQSL_NCDDST	Encoded state
10	(A)	BIT(8)	1	APQSL_SSFLGS	Flags
		1... ....		APQSL_SSROLE	0 - Active SS, 1 - Tracking SS
		.1.. ....		APQSL_SSXRFC	1 - XRF Capable
		..1. ....		APQSL_SSBAT	1 - Batch SS
		...1 ....		APQSL_SSIC	1 - IC SS
11	(B)	BIT(8)	1	*	Reserved
12	(C)	CHARACTER	4	APQSL_BKINFO	Backout information
12	(C)	SIGNED	2	APQSL_BKCTR	Backout needed count
14	(E)	SIGNED	2	APQSL_BKNUM	Backout done count

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
Possible access intent values (apqsl_ACCESS)				
1	HEX	01	APQSL_ACCRO	READ-GO
1	HEX	02	APQSL_ACCRD	READ
1	HEX	03	APQSL_ACCUP	UPDATE
1	HEX	04	APQSL_ACCEX	EXCLUSIVE

## 高速機能 DEDB の出力

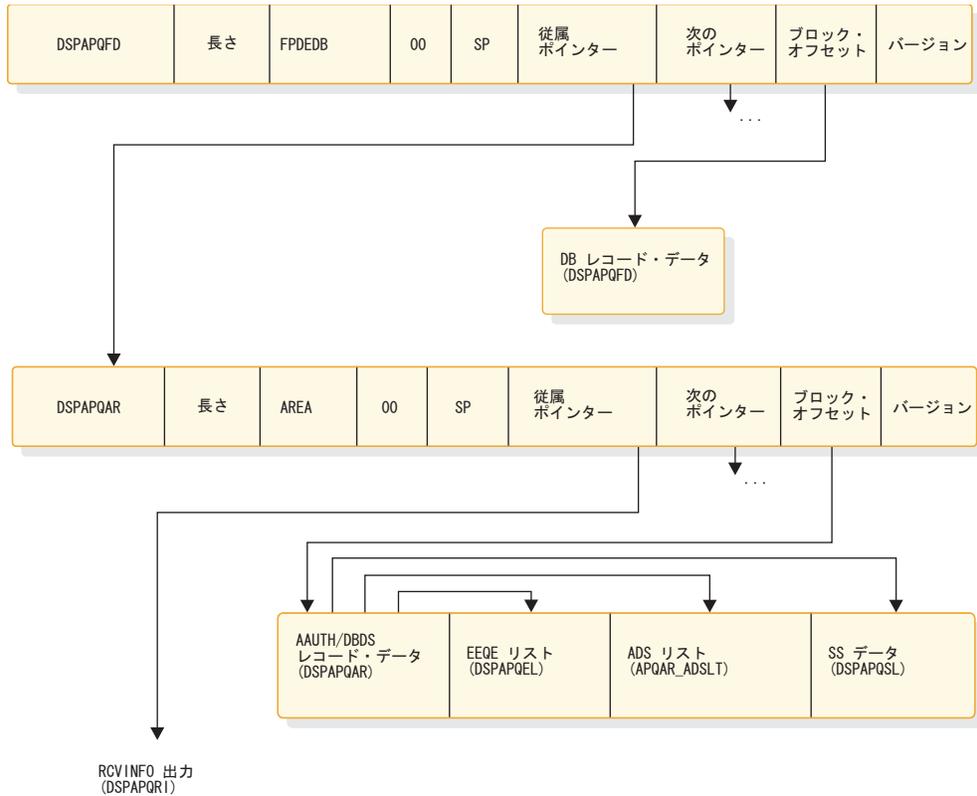


図 15. QUERY TYPE=DB (高速機能 DEDB) の出力のフォーマット

エリア情報は、DDN が指定されている場合にのみ戻されます。リカバリー情報 (RCVINFO) は、LIST パラメーターが指定されている場合にのみ戻されます。

## DSPAPQFD の DSECT

以下の例で、図 15 に示す DSPAPQFD ブロックおよび DSPAPQAR ブロックに含まれるフィールドについて説明します。

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	38	DSPAPQFD	
0	(0)	CHARACTER	8	APQFD_DBNAME	Database name
8	(8)	SIGNED	4	*(4)	Reserved
24	(18)	SIGNED	2	APQFD_RCVCTR	Recovery Needed Counter
26	(1A)	SIGNED	2	APQFD_ICCTR	IC Needed Counter
28	(1C)	SIGNED	2	APQFD_ICRECCTR	IC Recommended Counter
30	(1E)	UNSIGNED	2	APQFD_DMBNUM	Global DMB number
32	(20)	UNSIGNED	2	APQFD_EQECNT	Total EEQE count
34	(22)	SIGNED	2	APQFD_AUTHDAREAS	Number of Areas authorized
36	(24)	UNSIGNED	1	APQFD_SHRLVL	Share Level
37	(25)	BIT(8)	1	APQFD_FLAGS	Flags
		1... ..		APQFD_PAFLG	Prohibit authorization
		.1.. ..		APQFD_NONRV	nonrecoverable
		..1. ....		APQFD_ICNDIS	IC needed disabled option
		...1 ....		APQFD_USRRV	user-recoverable (VERSION=1.01)
		.... 1...		APQFD_FULLSEG_DEFAULT	Default full segment logging setting for areas (VERSION=4.0)
38	(26)	UNSIGNED	2	APQFD_ALTER#	FP DEDB alter status

40 (28) CHARACTER 8 APQFD\_RANDOMIZER (VERSION=5.0)  
Randomizer name  
(VERSION=5.01)

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQFD	APQFD_EYECATCHER	

### DSPAPQAR の DSECT

以下の例で、448 ページの図 15 に示す DSPAPQFD ブロックおよび DSPAPQAR ブロックに含まれるフィールドについて説明します。

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	196	DSPAPQAR	
0	(0)	CHARACTER	8	APQAR_DBNAME	Database name
8	(8)	CHARACTER	8	APQAR_AREANAME	Area name
16	(10)	UNSIGNED	4	APQAR_EEQELIST	Offset to EEQE list (DSPAPQEL), zero if no EEQEs
20	(14)	UNSIGNED	4	APQAR_SSLIST	Offset to SS list (DSPAPQSL), zero if no SS auth'd
24	(18)	UNSIGNED	4	APQAR_ADSLIST	Offset to ADS list, zero if none registered
28	(1C)	SIGNED	4	*	Reserved
32	(20)	UNSIGNED	1	APQAR_SHRLVL	Share level of DB
33	(21)	UNSIGNED	1	APQAR_HELDAU	Held auth state
		1... ..		APQAR_HAUBIT	High order bit flag
34	(22)	UNSIGNED	2	APQAR_DMBNUM	Global DMB number
36	(24)	SIGNED	2	APQAR_SSNUM	# subsystems auth'd to Area
38	(26)	UNSIGNED	2	APQAR_SSENTLEN	Length of each SS entry
40	(28)	UNSIGNED	1	APQAR_CACCSS	Access state for CHG AUTH
41	(29)	UNSIGNED	1	APQAR_CANCDD	Encoded state for CHG AUTH
42	(2A)	UNSIGNED	1	APQAR_CAHELD	Held state for CHG AUTH
43	(2B)	CHARACTER	5	APQAR_IRLMAU	IRLMID of auth SS
48	(30)	BIT(16)	2	APQAR_FLAGS	
		1... ..		APQAR_RECYC	REUSE image copies
		.1.. ..		APQAR_ICREC	Image Copy Recommended
		..1. ....		APQAR_IC	Image Copy Needed
		...1 ....		APQAR_ICNDIS	IC needed disabled option 1 = IC Needed Disabled
		.... 1..		APQAR_RECOV	Recovery needed
		.... .1..		APQAR_INPRO	HSSP CIC in progress
		.... ..1.		APQAR_GT240	M/C FP GT240 area DEDB
		.... ...1		APQAR_VSO	VSO flag
49	(31)	1... ..		APQAR_PREOP	PREOPEN flag
		.1.. ....		APQAR_PRELD	PRELOAD flag
		..1. ....		APQAR_LKASD	VSO CF buffer lookaside
		...1 ....		APQAR_MAS	VSO area resides in multi-area CF structure
		.... 1..		APQAR_RRGAL	REORG since last ALLOC, only set if RSR-covered
		.... .1..		APQAR_TSRAL	TS recov since last ALLOC, only set if RSR-covered
		.... ..1.		APQAR_FULLSEG	Full segment logging in effect (VERSION=4.0)
50	(32)	BIT(8)	1	APQAR_RSRFLAGS	Remote Site Recovery flags
		1... ..		APQAR_RCVTRK	Recovery Level Tracking
		.1.. ....		APQAR_TRKSPN	Tracking was suspended
		..1. ....		APQAR_PURBIT	Suspended by time
		...1 ....		APQAR_RCVRQ	Receive Required
51	(33)	BIT(8)	1	APQAR_AUFLAG	Authorization flags
		1... ..		APQAR_PAFLG	Prohibit authorization
		.1.. ....		APQAR_NONRV	nonrecoverable

	...1. ....		APQAR_DBQUI	Quiesce in progress
	...1 ....		APQAR_DBQUIH	Quiesce held
	.... 1...		APQAR_DBQUICMD	HALDB/DEDB on command
52	(34) BIT(8)	1	APQAR_DSORG	Data set organization
	1... ....		APQAR_VSAM	1 = VSAM, 0 = NON-VSAM
	.1.. ....		APQAR_INDEX	0 = Non-indexed (OSAM or ESDS), 1 = Indexed (ISAM or KSDS)
	..11 1111	*		Reserved - zeroes

以下の例で、448 ページの図 15 に示す DSPAPQFD ブロックおよび DSPAPQAR ブロックに含まれるフィールドについて説明します。

53	(35) CHARACTER	1	APQAR_DBORG	IMS DB organization
54	(36) CHARACTER	8	APQAR_GSGNAME	GSG Name
62	(3E) CHARACTER	2	*	Reserved
64	(40) UNSIGNED	4	APQAR_USID	Last ALLOC USID
68	(44) UNSIGNED	4	APQAR_AUSID	Last authorized USID
72	(48) UNSIGNED	4	APQAR_RUSID	Last received USID
76	(4C) UNSIGNED	4	APQAR_HUSID	Hardened USID
80	(50) UNSIGNED	4	APQAR_RNUSID	Receive needed USID
84	(54) CHARACTER	8	APQAR_RECOVGRP	Recovery Group name
92	(5C) CHARACTER	8	APQAR_CAGRPNAME	Change Accum group name
100	(64) UNSIGNED	2	APQAR_GENMX	Max number of ICs that may be predefined for this area
102	(66) UNSIGNED	2	APQAR_GENNO	Number of available ICs for this area
104	(68) UNSIGNED	2	APQAR_USDIC	Number of ICs used
106	(6A) SIGNED	2	APQAR_EEQECOUNT	EEQE count
108	(6C) UNSIGNED	2	APQAR_EEQELENGTH	EEQE entry length
110	(6E) UNSIGNED	1	APQAR_NOADS	# of ADS in the area
111	(6F) UNSIGNED	1	APQAR_AVADS	# of available ADS
112	(70) UNSIGNED	2	APQAR_ADLENGTH	ADS entry length
114	(72) CHARACTER	2	*	Reserved
116	(74) CHARACTER	40	APQAR_JCL	GENJCL members
116	(74) CHARACTER	8	APQAR_ICJCL	Image copy member
124	(7C) CHARACTER	8	APQAR_OIJCL	Online IC member
132	(84) CHARACTER	8	APQAR_RCJCL	Recovery member
140	(8C) CHARACTER	8	APQAR_DFJCL	DEFLTJCL member
148	(94) CHARACTER	8	APQAR_RVJCL	Receive JCL member
156	(9C) UNSIGNED	2	APQAR_RTPRD	IC retention period
158	(9E) UNSIGNED	2	APQAR_DSID	IMS data set ID
160	(A0) UNSIGNED	4	APQAR_DSSN	Data set sequence number
164	(A4) CHARACTER	16	APQAR_CFST1	VSO CF Structure 1
180	(B4) CHARACTER	16	APQAR_CFST2	VSO CF Structure 2

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	56	APQAR_ADSL	Area Data Set List
0	(0)	CHARACTER	8	APQAR_ADSDD	DDNAME of the ADS
8	(8)	CHARACTER	44	APQAR_ASDSN	DSN of the ADS
52	(34)	BIT(8)	1	APQAR_ADSBT	
		1... ....		APQAR_ADSAV	Avail status of ADS
		.1.. ....		APQAR_ADSFM	Format status of create util
		..1. ....		APQAR_ADSCP	Copy status of create util
53	(35)	CHARACTER	3	*	Reserved

#### CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQAR	APQAR_EYECATCHER	

### DSPAPQEL の DSECT

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
-------------------	---------------	------	--------	------------	-------------

0	(0)	STRUCTURE	13	DSPAPQEL	
0	(0)	CHARACTER	13	APQEL_EEQENTRY	EEQE descriptor entry
0	(0)	BIT(8)	1	APQEL_EQEFG	EEQE flags
	1...	....		APQEL_ERTL	Toleration error
	.1..	....		APQEL_ERRD	Read error
	..1.	....		APQEL_ERWT	Write error
	...1	....		APQEL_ERUS	DBRC user modified
	....	1...		APQEL_ERPM	DBRC permanent error
	....	.1..		APQEL_INDT	Indoubt EEQE
	....	..1.		APQEL_CIIND	Index CI indicator
1	(1)	CHARACTER	4	APQEL_EQE	EEQE
5	(5)	CHARACTER	8	APQEL_SSID	SSID which owns the EEQE

## HALDB (マスターおよびすべての区画) の出力

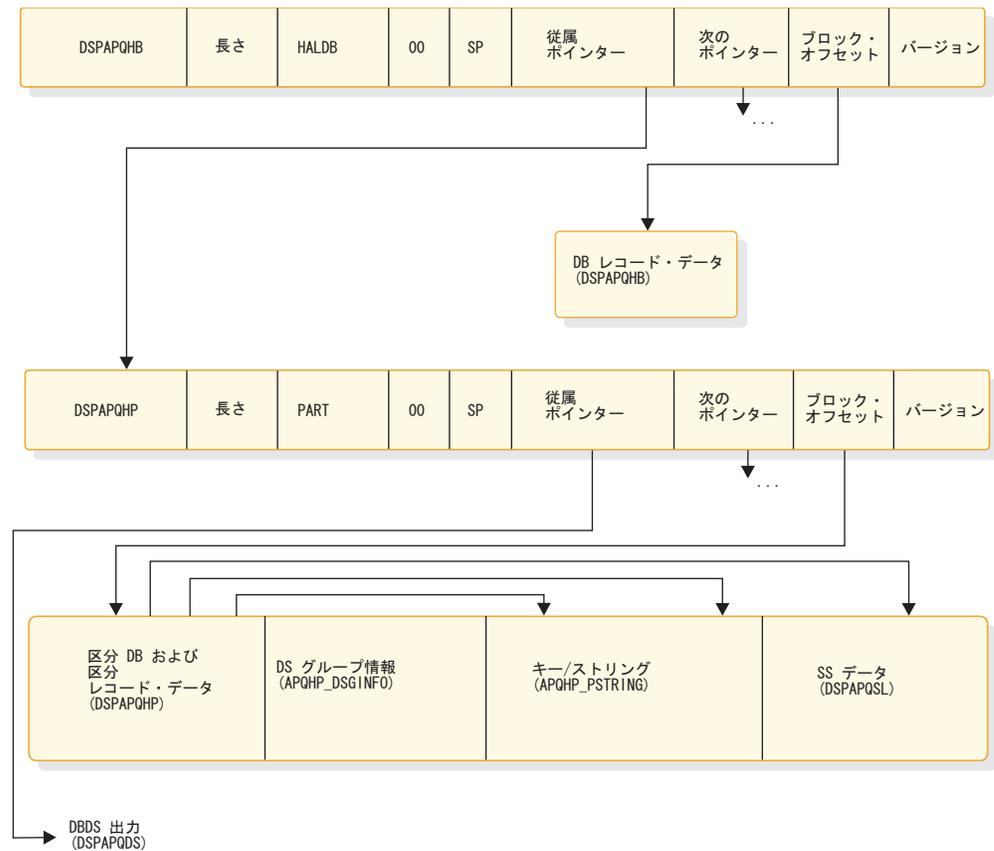


図 16. QUERY TYPE=DB (HALDB マスターおよび区画) の出力のフォーマット

DBDS 情報は、DDN が指定されている場合にのみ戻されます。

## DSPAPQHB の DSECT

以下の 2 つの例で、図 16 に示す DSPAPQHB ブロックおよび DSPAPQHP ブロックに含まれるフィールドについて説明します。DBDS 出力のフィールドの図については、454 ページの図 17 を参照してください。

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	60	DSPAPQHB	
0	(0)	CHARACTER	8	APQHB_DBNAME	HALDB name
8	(8)	SIGNED	4	*(4)	Reserved

24	(18) BIT(8)	1	APQHB_FLAGS	Flags
	1... ..		APQHB_NONRV	nonrecoverable
	.1.. ..		APQHB_ICNDIS	IC needed disabled
	..1. ....		APQHB_OLRCAP	HALDB is OLR capable
	...1 ....		APQHB_OSAM8G	HALDB OSAM is 8GB addressability
25	(19) BIT(8)	1	APQHB_ORG	DB organization
	1... ..		APQHB_PSINDEX	PSINDEX DB
	.1.. ..		APQHB_PHIDAM	PHIDAM DB
	..1. ....		APQHB_PHDAM	PHDAM DB
	...1 ....		APQHB_OSAM	OSAM DB
	.... 1111		*	Reserved
26	(1A) UNSIGNED	1	APQHB_SHRLVL	Share level
27	(1B) UNSIGNED	1	APQHB_DSGCNT	# DS Group members
28	(1C) UNSIGNED	2	APQHB_DMBNUM	Global DMB number
30	(1E) UNSIGNED	2	APQHB_PARTID	Current Partition ID
32	(20) SIGNED	2	APQHB_PART#	Number of parts in HALDB
34	(22) UNSIGNED	2	APQHB_VERSION#	Version number
36	(24) CHARACTER	8	APQHB_PSNAME	Name of Part Sel Routine
44	(2C) CHARACTER	8	APQHB_GSGNAME	GSG name
52	(34) CHARACTER	8	APQHB_RECOVGRP	Recovery Group name
60	(3C) UNSIGNED	2	APQHB_ALTER#	The total number of partitions to be altered in an online HALDB データベース
62	(3E) UNSIGNED	2	APQHB_ALTCMP#	The number of partitions that an active alter operation has completed at the time the query call was processed

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQHB	APQHB_EYECATCHER	

DSPAPQHP の DSECT

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	192	DSPAPQHP	
0	(0)	CHARACTER	8	APQHP_HALDBNAME	HALDB name
8	(8)	CHARACTER	8	APQHP_PARTNAME	Partition name
16	(10)	UNSIGNED	4	APQHP_SSLIST	Offset to SS list (DSPAPQSL), zero if no SS auth'd
20	(14)	UNSIGNED	4	APQHP_KEYSTRING	Offset to KEYSTRING (apqhp_PString), zero is no key/string
24	(18)	UNSIGNED	4	APQHP_DSGINFOFFSET	Offset to data set group information
28	(1C)	SIGNED	4	APQHP_ALTERINFOFFSET	Offset of HALDB Alter information structure, APQHP_ALTERINFO, if present. Value is 0 if not present.
32	(20)	CHARACTER	44	*	
32	(20)	CHARACTER	37	APQHP_DSNBASE	Base Partition DSN
76	(4C)	CHARACTER	18	APQHP_HDAM	PHDAM fields
76	(4C)	CHARACTER	8	APQHP_RMNAME	Randomizing module name
84	(54)	SIGNED	4	APQHP_RBN	Max relative block number
88	(58)	SIGNED	4	APQHP_BYTES	Max # of bytes
92	(5C)	UNSIGNED	2	APQHP_ANCHR	# of root anchor points
94	(5E)	UNSIGNED	1	APQHP_FBFF	Free block frequency factor
95	(5F)	UNSIGNED	1	APQHP_FSPF	Free space percentage factor
96	(60)	UNSIGNED	2	APQHP_PARTID	Partition ID
98	(62)	SIGNED	2	APQHP_PSTLN	Length of Part Key/String, apqhp_PString
100	(64)	UNSIGNED	2	APQHP_DSGINFOLEN	Length of each

102	(66)	UNSIGNED	1	APQHP_DSGCNT	aphp_DSGinfo entry DSG count
103	(67)	BIT(8)	1	APQHP_FLAGS	Flags
		1... ..		APQHP_PINIT	Partition must be initialized
		.1.. ..		APQHP_ORDBDS	0=A-J/1=M-V DBDS active
		..1. ....		APQHP_OLRON	OLR active
		...1 .....		APQHP_DISAB	Partition Disabled
		.... 1...		APQHP_MVDBDS	1 = M-V DBDS exist
		..... .1..		APQHP_OLRCAP	Partition is OLR capable
		..... ..1.		APQHP_OLRREL	1 = RELEASE OLR OWNER
		..... ...1		APQHP_OSAM8G	8GB OSAM addressability

=====  
If the Partitioned DB uses high keys, that is, no Partition Selection routine, the next two fields are used to sort the partitions in key sequence.  
=====

104	(68)	CHARACTER	8	APQHP_PREV	DDN of previous partition
112	(70)	CHARACTER	8	APQHP_NEXT	DDN of next partition
120	(78)	CHARACTER	8	APQHP_OLRIMS	Owning IMS for OLR
128	(80)	UNSIGNED	2	APQHP_IRCNT	IC receive needed counter
130	(82)	BIT(8)	1	APQHP_AUFLAG	Authorization flags
		1... ..		APQHP_BKFLG	Backout needed
		.1.. ..		APQHP_PAFLG	Prohibit authorization
		..1. ....		APQHP_RDFLG	Read only SS auth
		...1 .....		APQHP_NONRV	nonrecoverable
		.... 1...		APQHP_DBREORGI	Reorg intent
		..... .1..		APQHP_DBQUI	Quiesce in progress
		..... ..1.		APQHP_DBQUIH	Quiesce held
		..... ...1		APQAR_DBQUICMD	HALDB/DEDB on command
131	(83)	CHARACTER	5	APQHP_IRLMAU	IRLM ID of auth SS
136	(88)	SIGNED	2	APQHP_RCVCTR	Recovery needed count
138	(8A)	SIGNED	2	APQHP_ICCTR	IC needed count
140	(8C)	UNSIGNED	1	APQHP_SHRLVL	Share level of DB
141	(8D)	UNSIGNED	1	APQHP_HELDAU	Held auth state
		1... ..		APQHP_HAUBIT	High order bit flag
142	(8E)	UNSIGNED	2	APQHP_DMBNUM	Global DMB number
144	(90)	SIGNED	2	APQHP_SSNUM	# of SS auth DB
146	(92)	UNSIGNED	2	APQHP_SSENTLEN	Length of each SS entry
148	(94)	UNSIGNED	1	APQHP_CACCSS	Access state for chg auth
149	(95)	UNSIGNED	1	APQHP_CANCDD	Encode state for chg auth
150	(96)	UNSIGNED	1	APQHP_CAHELD	Held state for chg auth
151	(97)	CHARACTER	1	*	Reserved
152	(98)	UNSIGNED	2	APQHP_EQECNT	Total EQE count
154	(9A)	BIT(16)	2	APQHP_RSRFLG	Flags
		1... ..		APQHP_RCVTRK	Only recov level tracking
		.1.. ..		APQHP_TRKSPN	Tracking is suspended
		..1. ....		APQHP_PURBIT	Suspended by time
		...1 .....		APQHP_ICNDIS	IC needed disabled option
		.... 1...		APQHP_NOHKEY	High key required
		..... .1..		APQHP_ALTER	Partition being altered
		..... ...1		APQHP_ALTCMP	Partition alter completed; partition ready for online change
156	(9C)	CHARACTER	8	APQHP_GSGNAME	GSG name
164	(A4)	UNSIGNED	4	APQHP_USID	Last alloc USID
168	(A8)	UNSIGNED	4	APQHP_AUSID	Last authorized USID
172	(AC)	UNSIGNED	4	APQHP_RUSID	Last received USID
176	(B0)	UNSIGNED	4	APQHP_HUSID	Hardened by tracker USID
180	(B4)	UNSIGNED	4	APQHP_RNUSID	Receive needed USID
184	(B8)	SIGNED	2	APQHP_ICRECCTR	IC Recommended Counter
186	(BA)	UNSIGNED	2	APQHP_VERSION#	Version number
188	(BC)	UNSIGNED	1	APQHP_OLRACTHARDCTR	OLR curs active count
189	(BD)	UNSIGNED	1	APQHP_OLRINACTHARDCTR	OLR curs inact count
190	(BE)	UNSIGNED	2	APQHP_REORG#	Partition reorg #
192	(C0)	CHARACTER	8	APQHP_OLRBytes	OLR Bytes moved

200	(C8)	CHARACTER	8	APQHP_OLRSegs	OLR Segments moved
208	(D0)	CHARACTER	4	APQHP_OLRRoots	OLR Root Segments
212	(D4)	UNSIGNED	2	APQHP_ALTERINFOLEN	Length of each apqhp_AlterInfo entry

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	2	APQHP_DSGINFO	Data set group information
0	(0)	UNSIGNED	2	APQHP_BLKSZ	DS block size, OSAM only

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	*	APQHP_PSTRING	Partition Key/String

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	2	APQHP_ALTERINFO	DB Alter information Structure present only when an alter operation is in progress
0	(0)	UNSIGNED	2	APQHP_ALTERSZ	If alter changes block or CI sizes, ALTERSZ contains the OSAM block size or VSAM CI size for the output data sets of an alter operation

#### CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQHP	APQHP_EYECATCHER	

### DBDS の出力

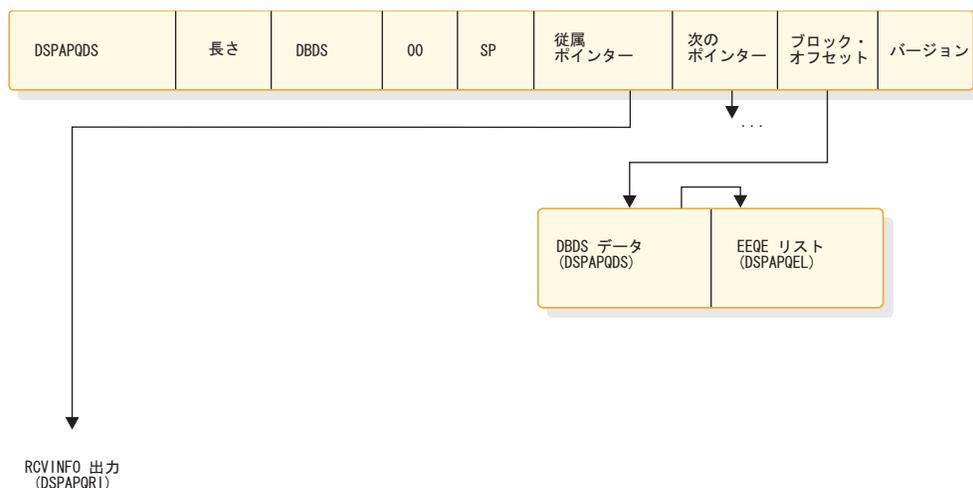


図 17. QUERY TYPE=DB (DBDS) の出力のフォーマット

リカバリー情報 (RCVINFO) は、LIST パラメーターが指定されている場合にのみ戻されます。

## DSPAPQDS の DSECT

以下の例および 450 ページの『DSPAPQEL の DSECT』で、454 ページの図 17 に示す DSPAPQDS ブロックに含まれるフィールドについて説明します。リカバリ情報の出力フィールドの図については、456 ページの図 18 を参照してください。

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	160	DSPAPQDS	
0	(0)	CHARACTER	8	APQDS_DBNAME	Database name
8	(8)	CHARACTER	8	APQDS_DDNAME	DD name
16	(10)	UNSIGNED	4	APQDS_EEQELIST	Offset to EEQE list (DSPAPQEL), zero if no EEQEs
20	(14)	SIGNED	4	* (3)	Reserved
32	(20)	CHARACTER	44	APQDS_DSN	Data set name
76	(4C)	UNSIGNED	2	APQDS_RTPRD	IC retention period
78	(4E)	UNSIGNED	2	APQDS_DSID	Data set ID number
80	(50)	UNSIGNED	4	APQDS_DSSN	Data set sequence number
84	(54)	UNSIGNED	4	APQDS_RUSID	Recovered-to USID(trkr)
88	(58)	BIT(8)	1	APQDS_FLAGS	BINARY ZEROS
		1... ..		APQDS_RECYC	REUSE image copies
		.1.. ..		APQDS_ICREC	Image Copy Recommended
		..1. ..		APQDS_RCVRQ	Receive required
		...1 ..		APQDS_IC	Image Copy Needed
		.... 1..		APQDS_RECOV	Recovery Needed
		.... .1..		APQDS_NONRV	nonrecoverable
89	(59)	BIT(8)	1	APQDS_DSORG	Data set organization
		1... ..		APQDS_VSAM	1 = VSAM, 0 = NON-VSAM
		.1... ..		APQDS_INDEX	0 = Non-indexed (OSAM or ESDS), 1 = Indexed (ISAM or KSDS)
		..11 1111	*		Reserved - zeroes
90	(5A)	CHARACTER	1	APQDS_DBORG	IMS DB organization
91	(5B)	UNSIGNED	1	*	Reserved
92	(5C)	UNSIGNED	2	APQDS_GENMX	Max number of ICs that may be predefined for this area
94	(5E)	UNSIGNED	2	APQDS_AVAILIC#	Number of available ICs for this area
96	(60)	UNSIGNED	2	APQDS_USEDIC#	Number of ICs used
98	(62)	SIGNED	2	APQDS_EEQECOUNT	EEQE count
100	(64)	UNSIGNED	2	APQDS_EEQELENGTH	EEQE entry length
102	(66)	BIT(8)	1	APQDS_FLG1	Flags
		1... ..		APQDS_RRGAL	REORG since last ALLOC, only set if RSR-covered
		.1... ..		APQDS_TSRAL	TS recov since last ALLOC, only set if RSR-covered
103	(67)	BIT(8)	1	APQDS_FLG2	DBDS type flags
		1... ..		APQDS_PART	TYPEPART record
		.1... ..		APQDS_PDATA	TYPEPART subtype DATA
		..1. ....		APQDS_PILE	TYPEPART subtype ILE
		...1 ....		APQDS_PINDX	TYPEPART subtype Index
104	(68)	CHARACTER	8	APQDS_CAGRPNAME	Change Accum group name
112	(70)	CHARACTER	40	APQDS_JCL	GENJCL members
112	(70)	CHARACTER	8	APQDS_ICJCL	Image copy member
120	(78)	CHARACTER	8	APQDS_OIJCL	Online IC member
128	(80)	CHARACTER	8	APQDS_RCJCL	Recovery member
136	(88)	CHARACTER	8	APQDS_DFJCL	DEFLTJCL member
144	(90)	CHARACTER	8	APQDS_RVJCL	Receive JCL member
152	(98)	CHARACTER	8	APQDS_ODDN	OLR partner DBDS

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQDS	APQDS_EYECATCHER	

### リカバリー情報 (RCVINFO) の出力

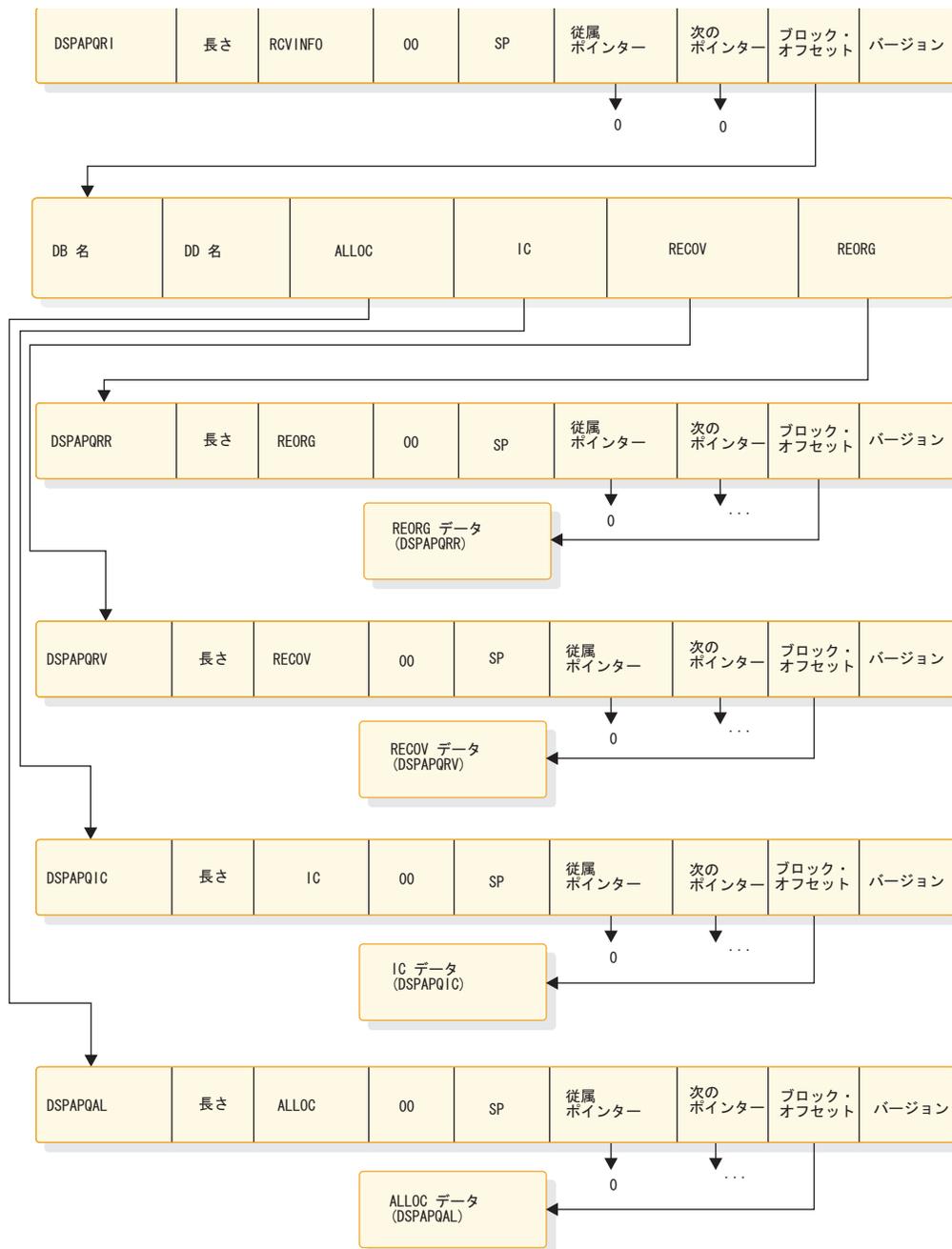


図 18. QUERY TYPE=DB (RCVINFO) の出力のフォーマット

リカバリー情報 (RCVINFO) は、LIST パラメーターが指定されている場合にのみ戻されます。特定の情報が存在しない場合、または要求されなかった場合、ポインタはゼロになります。

## DSPAPQRI の DSECT

以下の DSECT の例で、456 ページの図 18 に示す DSPAPQRI ブロックに含まれるフィールドについて説明します。

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	32	DSPAPQRI	
0	(0)	CHARACTER	8	APQRI_DBNAME	Database name
8	(8)	CHARACTER	8	*	
8	(8)	CHARACTER	8	APQRI_DDNAME	DD name
8	(8)	CHARACTER	8	APQRI_AREANAME	Area name
16	(10)	ADDRESS	4	APQRI_ALLOCPTR	ptr to ALLOC chain
20	(14)	ADDRESS	4	APQRI_ICPTR	ptr to IC chain
24	(18)	ADDRESS	4	APQRI_RECOVPTR	ptr to RECOV chain
28	(1C)	ADDRESS	4	APQRI_REORGPTR	ptr to REORG chain

### CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQRI	APQRI_EYECATCHER	

## DSPAPQAL の DSECT

以下の DSECT の例で、456 ページの図 18 に示す DSPAPQAL ブロックに含まれるフィールドについて説明します。

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	88	DSPAPQAL	
0	(0)	CHARACTER	8	APQAL_DBNAME	Database name
8	(8)	CHARACTER	8	*	
8	(8)	CHARACTER	8	APQAL_DDNAME	DD name or
8	(8)	CHARACTER	8	APQAL_AREANAME	Area name
16	(10)	CHARACTER	12	APQAL_ALLOCTM	Allocation time
28	(1C)	CHARACTER	12	APQAL_DALTM	Deallocation time
40	(28)	CHARACTER	12	APQAL_STRTM	Log start time
52	(34)	UNSIGNED	4	APQAL_DSSN	Field for DSSN value
56	(38)	UNSIGNED	4	APQAL_USID	Update set identifier
60	(3C)	CHARACTER	8	APQAL_ALRID	LRID of begin-upd rec
68	(44)	CHARACTER	8	APQAL_DLRID	LRID of end-upd rec
76	(4C)	CHARACTER	8	APQAL_SLRID	Last LRID applied if suspended
84	(54)	BIT(8)	1	APQAL_FLAGS	Flags
		1... ....		APQAL_TSUSP	Tracking is suspended
		.1.. ....		APQAL_NAPPL	No records applied
		..1. ....		APQAL_CICPT	Fuzzy ic purge time
		...1 ....		APQAL_DBQUI	Quiesce caused deallocation
85	(55)	CHARACTER	3	*	Reserved

### CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQAL	APQAL_EYECATCHER	

## DSPAPQIC の DSECT

以下の DSECT の例で、456 ページの図 18 に示す DSPAPQIC ブロックに含まれるフィールドについて説明します。

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
-------------------	---------------	------	--------	------------	-------------

0	(0)	STRUCTURE	64	DSPAPQIC	
0	(0)	CHARACTER	8	APQIC_DBNAME	Database name
8	(8)	CHARACTER	8	*	
8	(8)	CHARACTER	8	APQIC_DDNAME	DD name or
8	(8)	CHARACTER	8	APQIC_AREANAME	Area name
16	(10)	CHARACTER	12	APQIC_STARTIME	IC start time, packed decimal
28	(1C)	CHARACTER	12	APQIC_STOPTIME	IC stop time, packed decimal
40	(28)	BIT(8)	1	APQIC_TYPE	IMAGE COPY TYPE
		1... ..		APQIC_BATCH	BATCH
		.1.. ..		APQIC_CIC	CONCURRENT
		..1. ....		APQIC_USERIC	USER IMAGE COPY
		...1 ....		APQIC_ONLINE	ONLINE
		.... 1...		APQIC_SMSIC	SMS IC w/ DB exclusive
		.... .1..		APQIC_SMSCC	SMS IC w/ DB shared
		.... ..1.		APQIC_SMSOF	FastRep IC w/ DB exclusive (VERSION=2.00)
		.... ..1		APQIC_SMSON	FastRep IC w/ DB shared (VERSION=2.00)
41	(29)	BIT(8)	1	APQIC_STATUS	IC status flags
		1... ..		APQIC_AVAIL	Available IC
		.1.. ..		APQIC_IC1	Image Copy 1 exists
		..1. ....		APQIC_IC2	Image Copy 2 exists
		...1 ....		APQIC_ERR1	Error on image 1
		.... 1...		APQIC_ERR2	Error on image 2
		.... .1..		APQIC_EMP2	Image 2 defined and unused
42	(2A)	BIT(8)	1	APQIC_FLAGS	
		1... ..		APQIC_HSINP	HSSP CIC in progress
		.1.. ..		APQIC_CAT	Catalogued IC (HSSP)
43	(2B)	CHARACTER	1	APQIC_MoreTYPEs	More Image Copy types
		1... ..		APQIC_UserCIC	User Concurrent Image Copy (VERSION=2.0)
44	(2C)	CHARACTER	2	*	
44	(2C)	UNSIGNED	2	APQIC_OFF1	Offset to image 1 data
44	(2C)	UNSIGNED	2	APQIC_OFFU	Offset to user IC data
46	(2E)	UNSIGNED	2	APQIC_OFF2	Offset to image 2 data
48	(30)	UNSIGNED	4	APQIC_CNT12	Record count
52	(34)	UNSIGNED	4	APQIC_USID	Update set ID
56	(38)	CHARACTER	2	*	
56	(38)	UNSIGNED	2	APQIC_LEN12	Length of image 1/2 data
56	(38)	UNSIGNED	2	APQIC_LENU	Length of user IC data
58	(3A)	CHARACTER	6	*	Reserved
64	(40)	UNSIGNED	2	APQIC_OFFUD	Offset to user data (Version=4.0)
66	(42)	UNSIGNED	2	APQIC_LENUD	Length of user data (Version=4.0)

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	64	APQIC_IC12	Data for image 1 or 2
0	(0)	CHARACTER	44	APQIC_DSN12	Data set name
44	(2C)	UNSIGNED	2	APQIC_FILE	File sequence number
46	(2E)	CHARACTER	8	APQIC_RUT12	Unit device type
54	(36)	UNSIGNED	2	APQIC_VOLCT	# of volumes predefined
56	(38)	UNSIGNED	2	APQIC_VOLUS	# of volumes used
58	(3A)	UNSIGNED	2	APQIC_VOLLISTLEN	Length of each volume list entry in apqic_VOLS
60	(3C)	UNSIGNED	4	APQIC_VOLLISTOFFSET	Offset to volume list

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	CHARACTER	6	APQIC_VOLS	List of VOLSERS

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	80	APQIC_USER	Data for user IC
0	(0)	CHARACTER	80	APQIC_UDATA	User supplied data

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
1	DECIMAL	255	APQIC_MAXV	max # volumes
8	CHARACTER	DSPAPQIC	APQIC_EYECATCHER	

**DSPAPQRV の DSECT**

以下の DSECT の例で、456 ページの図 18 に示す DSPAPQRV ブロックに含まれるフィールドについて説明します。

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	49	DSPAPQRV	
0	(0)	CHARACTER	8	APQRV_DBNAME	Database name
8	(8)	CHARACTER	8	*	
8	(8)	CHARACTER	8	APQRV_DDNAME	DD name or
8	(8)	CHARACTER	8	APQRV_AREANAME	Area name
16	(10)	CHARACTER	12	APQRV_RUNTIME	The time at which the DBDS was recovered
28	(1C)	CHARACTER	12	APQRV_ENDTIME	Partial recovery only, the time to which the DBDS was restored
40	(28)	UNSIGNED	4	APQRV_FUSID	First undone USID
44	(2C)	UNSIGNED	4	APQRV_LUSID	Last undone USID
48	(30)	BIT(8)	1	APQRV_FLAGS	Flags
		1... ..	1	APQRV_PITR	Point In Time Recovery
		.1.. ..	1	APQRV_EXTCM	External command (Version=4.0)
49	(31)	UNSIGNED	1	*	Reserved
50	(32)	UNSIGNED	2	APQRV_OFFFUD	Offset to user data (Version=4.0)
52	(34)	UNSIGNED	2	APQRV_LENUD	Length of user data (Version=4.0)
54	(33)	UNSIGNED	2	APQRV_PREORG	Prior reorg number (Version=4.0)
56	(36)	UNSIGNED	2	APQRV_NREORG	New reorg number (Version=4.0)

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	80	APQRV_USER	Data for user data
0	(0)	CHARACTER	80	APQRV_UDATA	User supplied data (VERSION=4.0)

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQRV	APQRV_EYECATCHER	

**DSPAPQRR の DSECT**

以下の DSECT の例で、456 ページの図 18 に示す DSPAPQRR ブロックに含まれるフィールドについて説明します。

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	60	DSPAPQRR	
0	(0)	CHARACTER	8	APQRR_DBNAME	Database name
8	(8)	CHARACTER	8	*	
8	(8)	CHARACTER	8	APQRR_DDNAME	DD name or
8	(8)	CHARACTER	8	APQRR_AREANAME	Area name
16	(10)	CHARACTER	12	APQRR_RUNTIME	The time at which the DBDS was reorganized
28	(1C)	CHARACTER	12	APQRR_STOPTIME	Stoptime of online reorg
40	(28)	BIT(8)	1	APQRR_FLAGS	
		1... ..		APQRR_ONL	1=ONLINE/0=OFFLINE reorg
		.1.. ..		APQRR_RECOV	1=May be used for recovery
		..1. ....		APQRR_ALTER	1=HALDB structure altered by

41	(29)	CHARACTER	3	*	an online reorganization
					Reserved
44	(2C)	UNSIGNED	4	APQRR_USID	Associated USID
48	(30)	CHARACTER	12	APQRR_PITR	Stoptime moved - PITR
60	(3C)	UNSIGNED	4	APQRR_PRAPs	Total number of RAPs processed
64	(40)	UNSIGNED	4	APQRR_Roots	Total number of roots processed
68	(44)	UNSIGNED	2	APQRR_OFFUD	Offset to user data (VERSION=4.0)
70	(46)	UNSIGNED	2	APQRR_LENUD	Length of user data (VERSION=4.0)

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	80	APQRR_USER	Data for user data
0	(0)	CHARACTER	80	APQRR_UDATA	User supplied data (VERSION=4.0)

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQRR	APQRR_EYECATCHER	

### 「データベース未検出」の出力



図 19. QUERY TYPE=DB (データベース未検出) の出力のフォーマット

この出力ブロックは、DBLIST ブロックに指定されたデータベースの一部が RECON 内で検出できなかった場合に戻されます。検出できなかったデータベースごとに、ブロックが 1 つずつ戻されます。データベース名は、このブロックのデータ域に含まれています。

データベースが検出されないと、マクロ呼び出しで 4 (RC=4) の戻りコードを受け取ります。リスト内のデータベースがどれも検出されない場合 (RC=8)、出力ブロックは戻されません。

### DSPAPQNF の DSECT

以下の例で、図 19 に示す DSPAPQNF ブロックに含まれるフィールドについて説明します。

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	8	DSPAPQNF	
0	(0)	CHARACTER	8	APQNF_DBNAME	DB name

関連概念:

403 ページの『第 26 章 DBRC API』

関連資料:

431 ページの『第 30 章 DBRC 照会要求』

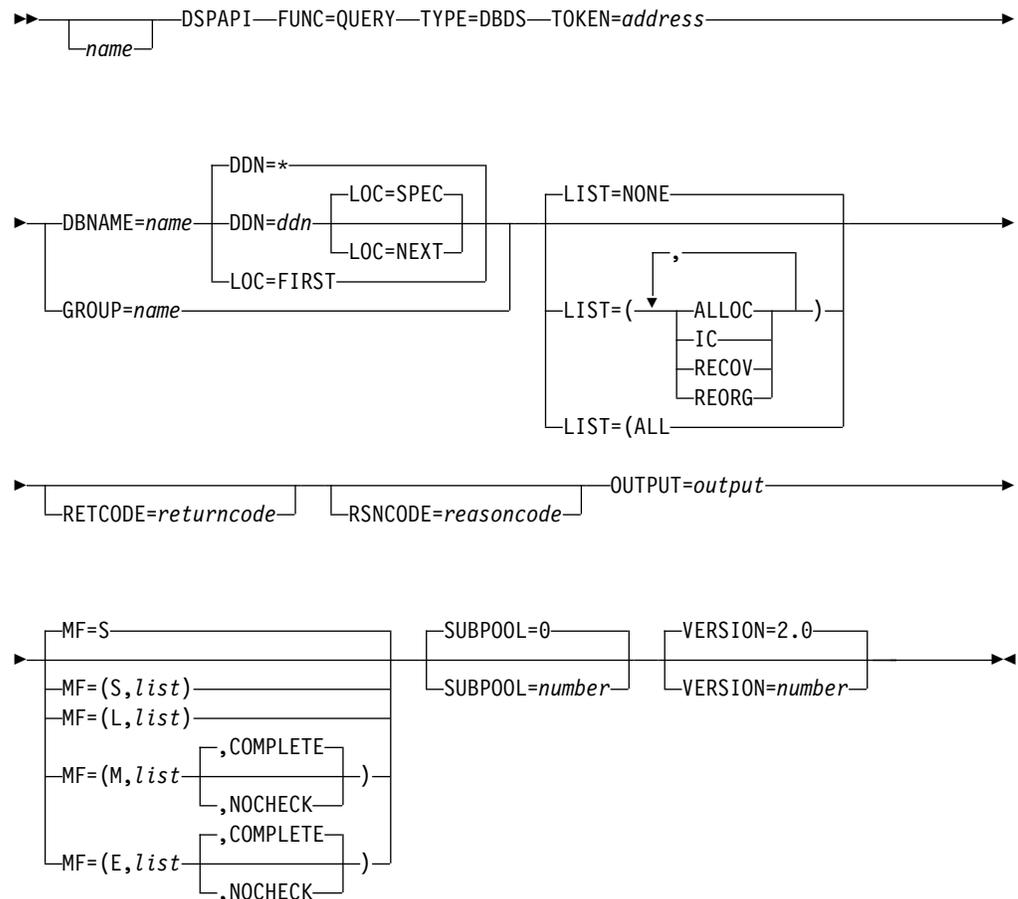
『DBDS 照会要求 (TYPE=DBDS)』

491 ページの『HALDB 区画照会要求 (TYPE=PART)』

## DBDS 照会要求 (TYPE=DBDS)

HALDB 以外のデータベース、HALDB 区画、DBDS グループ、または CA グループの 1 つ以上の DBDS に関する情報を RECON データ・セットから取得するには、DSPAPI FUNC=QUERY TYPE=DBDS 要求を使用します。割り振り、イメージ・コピー、リカバリー、および再編成の情報を含むデータ・セットのリカバリー関連情報も要求できます。

### TYPE=DBDS 照会要求の構文



### TYPE=DBDS 照会要求のパラメーター

名前 使用する場合は、1 桁目から始めます。

#### TYPE=DBDS

DBDS またはエリアに関する RECON 情報が要求されることを指定します。

TYPE パラメーターで DBDS を指定する場合、最小バージョン番号 VERSION=2.0 も指定する必要があります。

**TOKEN=address | (2 - 12)**

API トークンを受信する 4 バイト・フィールドのアドレスを指定します。このトークンは、この FUNC=STARTDBRC 要求に関連する後続のすべての要求に組み込む必要があります。

**DBNAME=name | (2 -12)**

照会する DBDS のデータベース名 (HALDB 以外) または区画名を指定します。このパラメーターは、データベースまたは区画の特定の DBDS を対象とする場合に使用します。HALDB の名前は指定できません。

DBDNAME または PARTNAME を指定する必要があります。

**GROUP=name | (2 -12)**

照会する DBDS の名前を含む DBDS グループまたは CA グループの名前を指定します。このパラメーターを指定する場合、LOC および DDN パラメーターは指定できません。

DBNAME または GROUP を指定する必要があります。データベース名およびリカバリー・グループ名は指定できません。

ワイルドカード GROUP=\* は、すべての TYPE=xxxxGROUP 照会で使用できますが、TYPE=DBDS では使用できません。ただし、このワイルドカードは TYPE=DBDSGROUP では使用できます。

**DDN=ddname | (2 -12)**

データ・セットまたはエリアの DD 名を指定します。このパラメーターは、データベースまたは区画の特定のデータ・セット、次のデータ・セット、または前のデータ・セットを照会する場合に、DBNAME パラメーターおよび LOC パラメーターと一緒に使用します。DDN=\* を指定すると、データベースのすべてのデータ・セットまたはエリアに関する情報が返されます。LOC=FIRST を使用せずに DBNAME を指定する場合、DDN=\* はデフォルトです。

DDN は LOC=SPEC|NEXT の場合は指定する必要があります。

次のデータ・セットまたは前のデータ・セットを照会する場合、DDN パラメーターの値は検索のベースとして使用され、データベースまたは区画について RECON に登録済みの DBDS の DD 名である必要はありません。

**LIST=NONE | LIST=({ALLOC},{IC},{RECOV},{REORG}) | LIST=ALL**

返される DBDS の照会出力に組み込むサポート情報のタイプを指定します。リストには、1 つ以上の特定の値が組み込まれます。それらの値には、ALLOC (割り振りレコード)、IC (イメージ・コピー・レコード)、RECOV (リカバリー・レコード)、または REORG (再編成レコード) があります。すべてのサポート情報を要求する場合は、LIST=ALL を指定します。サポート情報を要求しない場合は、LIST=NONE を指定します。

LIST=NONE はデフォルトです。

**LOC=FIRST | NEXT | SPEC**

(オプション) - その要求が、データベースまたは区画について RECON 内で定義済みの指定された区画 (SPEC)、最初の DBDS、または次の DBDS を対象とすることを指定します。DBNAME は LOC パラメーターの場合は必須です。LOC=NEXT|SPEC の場合は、ワイルドカードなしの DDN が

必須です。DDN は LOC=FIRST の場合は指定できません。特定の DDN を指定する場合、LOC=SPEC がデフォルトです。LOC は GROUP の場合は指定できません。

データ・セットまたはエリアの情報も要求するには DDN=\* を指定します。特定の DDN (DDN=ddn) を指定した場合、指定した DDN は無視され、すべてのデータ・セットまたはエリアの情報が返されます。

HALDB が高位キーを使用する場合、区画は高位キーの順序で返されます。それ以外の場合は、区画はアルファベット順で返されます。

#### **RETcode=returncode | (2-12)**

シンボルとして指定する場合、戻りコードを受け取るために、ストレージのワードを示すラベルを指定します。レジスターとして指定する場合、戻りコードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RETcode が指定されたかどうかにかかわらず、レジスター 15 は戻りコードを含んでいます。

#### **RSNcode=reasoncode | (2-12)**

シンボルとして指定する場合、理由コードを受け取るために、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、理由コードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RSNcode が指定されたかどうかにかかわらず、レジスター 0 は理由コードを含んでいます。

#### **OUTPUT=output | (2 - 12)**

区画の情報を含むブロック (おそらくチェーンになっている) の最初のブロックを示すポインターを受け取るフィールドを指定します。

出力が作成されなかった場合、出力アドレスはゼロです。この結果は、RECON 内に要求を満たす情報がない場合、または出力を作成できるようになる前にエラーが発生した場合に起こります。

出力ブロックのストレージは、呼び出し元によって事前割り振りされません。DBRC はこれらのブロック用に指定されたサブプールからストレージを取得します。呼び出し元は、バッファ解放サービス (DSPAPI FUNC=RELBUF) を使用してこのストレージを解放し、戻された出力アドレスを指定する必要があります。

#### **SUBPOOL= 0 | number**

取得されるストレージのサブプール番号を指定するオプション・パラメータです。このパラメータを指定しない場合、FUNC=STARTDBRC 要求によって指定されたサブプールがデフォルトになります。それ以外の場合は、サブプール 0 がデフォルトになります。

#### **MF=S | L | M | E**

要求のマクロ形式を指定します。

#### **VERSION=2.0 | number**

このマクロによって生成されるパラメータ・リストのバージョン番号を指定するオプション・パラメータ。

1 つのバージョンに関連するパラメータを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバ

バージョン・レベルを指定すると、パラメーターはマクロ処理によって受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

デフォルトのバージョンは 2.0 です。

注: TYPE=DBDS を指定する場合、API の最小バージョン番号 VERSION=2.0 を指定する必要があります。

## TYPE=DBDS 照会要求の戻りコードと理由コード

表 91. DSPAPI FUNC=QUERY TYPE=DBDS 照会の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
警告	X'00000008'	X'D8210001'	DB の DBDS またはエリアが RECON データ・セットに登録されていません。情報ブロックは戻されません。
	X'00000008'	X'D8210002'	指定された DBDS またはエリアが RECON データ・セットに登録されていません。情報ブロックは戻されません。
	X'00000008'	X'D8210003'	指定された DBNAME が HALDB です。DBNAME には HALDB 以外または HALDB 区画名を指定する必要があります。情報ブロックは戻されません。
	X'00000008'	X'D8210004'	指定された グループが RECON データ・セットに登録されていません。情報ブロックは戻されません。
	X'00000008'	X'D8210005'	指定されたグループが DBDS または CA グループではありません。情報ブロックは戻されません。
	X'00000008'	X'D8210006'	指定された DBNAME が RECON に登録されていません。情報ブロックは戻されません。
重大エラー	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。
	X'0000000C'	X'D8210001'	SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーは要求の実行を許可されていないと判断しました。
ストレージ・エラー	X'00000028'	X'D8210001'	DBDS ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210002'	AREA ブロック用のストレージの取得時にエラーが発生しました。

表 91. DSPAPI FUNC=QUERY TYPE=DBDS 照会の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000028'	X'D8210003'	RCVINFO ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210004'	ALLOC ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210005'	IC ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210006'	REORG ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210007'	RECOV ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D9100001'	要求の処理中にエラーが発生しました。DBRCはこの時点までに取得したストレージを解放します。しかし、ストレージを解放しようとして別のエラーが検出されました。
内部エラー	X'0000002C'	X'D8000001'	RECON データ・セットを開くときに障害が発生しました。
	X'0000002C'	X'D8210001'	最初の DBDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210002'	指定された DBDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210003'	次の DBDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210004'	最初のエリアの AUTH レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210005'	最初の ALLOC レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210006'	次の ALLOC レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210007'	最初の IC レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210008'	次の IC レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210009'	最初の REORG レコードの検索時に障害が発生しました。
	X'0000002C'	X'D821000A'	次の REORG レコードの検索時に障害が発生しました。
	X'0000002C'	X'D821000B'	最初の RECOV レコードの検索時に障害が発生しました。
	X'0000002C'	X'D821000C'	次の RECOV レコードの検索時に障害が発生しました。
	X'0000002C'	X'D821000D'	指定されたグループ・レコード (DBDS グループ) の検索時に障害が発生しました。

表 91. DSPAPI FUNC=QUERY TYPE=DBDS 照会の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'0000002C'	X'D821000E'	指定されたグループ・レコード (CA グループ) の検索時に障害が発生しました。
	X'0000002C'	X'D821000F'	指定されたグループ・レコードからの DBDS の検索時に障害が発生しました。
	X'0000002C'	X'D8210010'	指定された DBNAME による DB レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210011'	最初の AVAIL IC レコードを検索しようとして障害が発生しました。
	X'0000002C'	X'D8210012'	次の AVAIL IC レコードを検索しようとして障害が発生しました。
パラメータ・エラー	X'00000030'	X'C9000001'	API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN フィールド・アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000003'	RETCODE フィールド・アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE フィールド・アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000005'	OUTPUT フィールド・アドレスが無効です。OUTPUT アドレスを含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000008'	DBNAME または GROUP アドレスが無効です。DBNAME または GROUP を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000009'	DDN アドレスが無効です。DDN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'D8000001'	OUTPUT パラメーターが欠落しているか無効です。

表 91. DSPAPI FUNC=QUERY TYPE=DBDS 照会の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'D8000002'	TYPE パラメーターに指定された値が無効です。
	X'00000030'	X'D8000003'	照会 TYPE=DBDS には、API の最小バージョン 2.0 が必要です。
	X'00000030'	X'D8210001'	DBNAME または GROUP は必須です。
	X'00000030'	X'D8210002'	無効な LOC 値で DDN が指定されました。入力できるのは、LOC=NEXT SPEC のみです。
	X'00000030'	X'D8210003'	LOC 値が無効です。DDN を指定しない場合、使用できるのは LOC=FIRST のみです。
	X'00000030'	X'D8210004'	DDN パラメーターは GROUP パラメーターの場合は指定できません。
	X'00000030'	X'D8210005'	LOC は GROUP パラメーターの場合は指定できません。

以下のブロック・マッピングは、TYPE=DB および TYPE=DBDS 要求に関連するものです。

- DSPAPQAR – 高速機能 AREA ブロック
- DSPAPQDS – DBDS ブロック
- DSPAPQEL – EEQE リスト
- DSPAPQSL – サブシステム・リスト (高速機能のみ)
- DSPAPQRI – リカバリー情報 (RCVINFO) ブロック
- DSPAPQAL – 割り振りブロック
- DSPAPQIC – イメージ・コピー・ブロック
- DSPAPQRV – リカバリー・ブロック
- DSPAPQRR – 再編成ブロック

関連概念:

403 ページの『第 26 章 DBRC API』

関連資料:

431 ページの『第 30 章 DBRC 照会要求』

437 ページの『データベース照会要求 (TYPE=DB)』

428 ページの『コマンド出力ブロックのマッピング』

## グループ照会要求 (TYPE=\*GROUP)

RECON データ・セットに登録済みの以下のグループ・タイプの 1 つに関するグループとメンバーの情報を取得するには、グループ照会 (DSPAPI FUNC=QUERY TYPE=\*GROUP) 要求を使用します。

以下のリストに、グループ照会 (DSPAPI FUNC=QUERY TYPE=\*GROUP) 要求を使用して情報を取得できるグループを概略します。

- DBDS グループ (TYPE=DBDSGROUP)
- DB グループ (TYPE=DBGROUP)
- リカバリー・グループ (TYPE=RECOVGROUP)
- CA グループ (TYPE=CAGROUP)
- グローバル・サービス・グループ (TYPE=GSGROUP)

以下のサブセクションがあります。

- 『TYPE=\*GROUP 照会要求の構文』
- 469 ページの『TYPE=\*GROUP 照会要求のパラメーター』
- 470 ページの『TYPE=\*GROUP 照会要求の戻りコードと理由コード』
- 472 ページの『TYPE=\*GROUP 照会要求の出力』
- 472 ページの『QUERY TYPE=DBDSGROUP、DBGROUP、および RECOVGROUP の出力』
- 473 ページの『DSPAPQDG の DSECT』
- 474 ページの『QUERY TYPE=CAGROUP の出力』
- 474 ページの『DSPAPQCG の DSECT』
- 475 ページの『DSPAPQCA の DSECT』
- 476 ページの『QUERY TYPE=GSGROUP の出力』
- 476 ページの『DSPAPQGG の DSECT』

## TYPE=\*GROUP 照会要求の構文



## TYPE=\*GROUP 照会要求のパラメーター

名前 ユーザーが指定できるオプションのシンボルです。使用する場合は、1 桁目から始めます。

### TYPE=DBDSGROUP | DBGROUP | RECOVGROUP | CAGROUP | GSGROUP

情報を要求するグループのタイプを指定します。

### CAINFO= YES | NO

CA 実行情報を CAGROUP 情報に組み込むかどうかを指定します。CAINFO は TYPE=CAGROUP の場合にのみ有効です。CAINFO は、TYPE=CAGROUP が指定されている場合はデフォルトの NO に設定され、CA グループ・メンバー情報のみが要求されることを示します。

CAINFO=NO が指定されている場合、または CAINFO=YES が指定されていても CA 実行情報が存在しない場合は、CAGROUP ブロックのヘッダーのブロック従属ポインター (apqhd\_depptr) は 0 になります。

### TOKEN=symbol | (2 - 12)

FUNC=STARTDBRC 要求で戻された 4 バイト・フィールドのアドレスを指定します。

### GROUP= \_ | symbol | symbol\* | (2 - 12)

照会されるグループの名前を指定します。ワイルドカード・キーワード \* (アスタリスク) のみを使用して、すべてのグループに関する情報を要求することができます。名前の最後にワイルドカードを使用することもできます。この場合、アスタリスクの前には、英字が少なくとも 1 つ必要です。デフォルトは GROUP=\* です。

GROUP キーワードまたは NAME キーワードを使用できます。GROUP が優先されます。NAME は互換性の理由で受け入れられます。

### NAME= \_ | symbol | symbol\* | (2 - 12)

GROUP パラメーターを参照してください。

### MF=S | L | M | E

要求のマクロ形式を指定します。

### OUTPUT=output | (2 - 12)

グループ情報ブロック (おそらくチェーンになっている) の最初のブロックを示すポインターを受け取るフィールドを指定します。戻される情報ブロックの詳細については、472 ページの『TYPE=\*GROUP 照会要求の出力』を参照してください。

出力アドレスは、出力が作成されなかった場合はゼロになります。これは、RECON 内に要求を満たす情報がない場合、または出力を作成できるようになる前にエラーが発生した場合に、起こる可能性があります。

出力ブロックのストレージは、呼び出し元によって事前割り振りされません。DBRC はこれらのブロック用に指定されたサブプールからストレージを取得します。呼び出し元は、バッファ解放サービス (DSPAPI FUNC=RELBUF) を使用してこのストレージを解放し、戻された出力アドレスを指定する必要があります。

**SUBPOOL= 0 | number**

取得されるストレージのサブプール番号を指定します。このパラメーターを指定しない場合、FUNC=STARTDBRC 要求によって指定されたサブプールがデフォルトになります。それ以外の場合は、サブプール 0 がデフォルトになります。

**RETCODE=returncode | (2-12)**

シンボルとして指定する場合、戻りコードを受け取るために、ストレージのワードを示すラベルを指定します。レジスターとして指定する場合、戻りコードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RETCODE が指定されたかどうかにかかわらず、レジスター 15 は戻りコードを含んでいます。

**RSNCODE=reasoncode | (2-12)**

シンボルとして指定する場合、理由コードを受け取るために、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、理由コードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RSNCODE が指定されたかどうかにかかわらず、レジスター 0 は理由コードを含んでいます。

**VERSION=2.0 | number**

この要求によって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターはマクロ処理によって受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

FUNC=QUERY TYPE=xxxxGROUP 要求の有効なバージョン番号は、1.0 および 2.0 です。

**TYPE=\*GROUP 照会要求の戻りコードと理由コード**

以下の表には、TYPE=\*GROUP 照会要求の戻りコードと理由コードのほとんどを示しています。

表 92. TYPE=\*GROUP 照会要求の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
警告	X'00000008'	X'D8300001'	要求されたタイプのグループ・レコードが RECON データ・セット内に存在しません。
重大エラー	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。

表 92. TYPE=\*GROUP 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'0000000C'	X'D8300100'	セキュリティー・エラー。SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーは要求の実行を許可されていないと判断しました。
ストレージ・エラー	X'00000028'	X'D8300001'	CAGROUP ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8300002'	GSG ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8300003'	DBDSGRP、DBGGRP、または RECOVGRP ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8310001'	CA ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D9100001'	要求の処理中にエラーが発生しました。DBRC は、この時点までに取得したストレージを解放します。しかし、ストレージを解放しようとして別のエラーが検出されました。
内部エラー	X'0000002C'	X'D8000001'	RECON を開くときに障害が発生しました。
	X'0000002C'	X'D8300001'	要求されたグループ・タイプの特定のグループ・レコードまたは最初のグループ・レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8300002'	要求されたグループ・タイプの次のグループ・レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8300003'	CA レコードの検索時に障害が発生しました。
パラメーター・エラー	X'00000030'	X'C9000001'	API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN フィールド・アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000003'	RETCODE フィールド・アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE フィールド・アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

表 92. TYPE=\*GROUP 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'C9000005'	OUTPUT フィールド・アドレスが無効です。OUTPUT アドレスを含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000008'	GROUP または NAME フィールド・アドレスが無効です。グループ名を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'D8000001'	OUTPUT パラメーターが欠落しているか無効です。
	X'00000030'	X'D8000002'	TYPE パラメーターに指定された値が無効です。
	X'00000030'	X'D8300001'	CAINFO=YES は TYPE=CAGROUP の場合にしか指定できません。
	X'00000030'	X'D8300100'	ワイルドカードを使用するとき、アスタリスクの前に英字が少なくとも 1 つ必要です。
	X'00000030'	X'D8300101'	ワイルドカードを使用するとき、最後の文字はアスタリスクである必要があります。

## TYPE=\*GROUP 照会要求の出力

この後のいくつかの図に、QUERY TYPE=\*GROUP 要求からの出力フォーマットを示します。出力のレイアウトを示す図の後のサンプル DSECT では、ストレージ・ブロックのフィールドとそれらのフィールド相互の関係を詳細に説明します。

## QUERY TYPE=DBDSGROUP、DBGROUP、および RECOVGROUP の出力

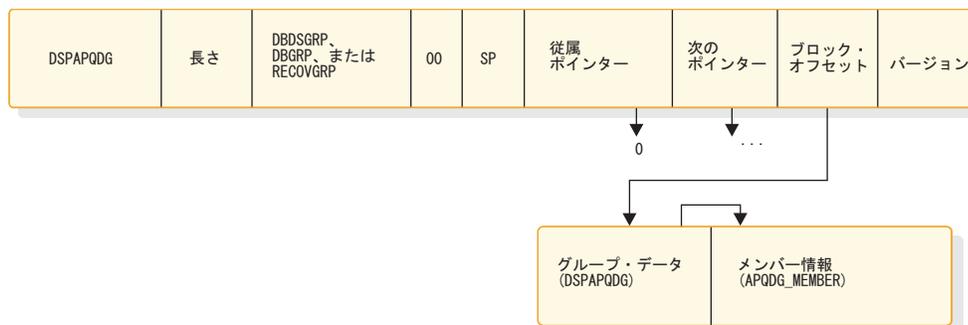


図 20. QUERY TYPE=DBDSGROUP、DBGROUP、RECOVGROUP の出力のフォーマット

## DSPAPQDG の DSECT

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	32	DSPAPQDG	
0	(0)	CHARACTER	8	APQDG_GROUPNAME	Group name
8	(8)	UNSIGNED	4	APQDG_MEMBERINFO	Offset to group member list
12	(C)	UNSIGNED	2	APQDG_MEMBERLEN	Length of group member entry
14	(E)	SIGNED	2	APQDG_MEMBERCOUNT	Number of group members
16	(10)	SIGNED	4	*(3)	Reserved
28	(1C)	CHARACTER	4	*	Reserved

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	16	APQDG_MEMBER	List of group members
0	(0)	CHARACTER	16	*	
0	(0)	CHARACTER	16	APQDG_DBDSG	DBDS group
0	(0)	CHARACTER	8	APQDG_DBDSG_DBNAME	Database name
8	(8)	CHARACTER	8	*	
8	(8)	CHARACTER	8	APQDG_DBDSG_DDNAME	DD name or
8	(8)	CHARACTER	8	APQDG_DBDSG_AREANAME	AREA name
0	(0)	CHARACTER	16	APQDG_DBG	DB group
0	(0)	CHARACTER	8	*	
0	(0)	CHARACTER	8	APQDG_DBG_DBNAME	Database name or
0	(0)	CHARACTER	8	APQDG_DBG_AREANAME	AREA name
8	(8)	CHARACTER	8	*	Not used
0	(0)	CHARACTER	16	APQDG_RECOVG	Recovery group
0	(0)	CHARACTER	8	APQDG_RECOVG_DBNAME	Database name
8	(8)	CHARACTER	8	APQDG_RECOVG_AREANAME	AREA name, null if not Fast Path

### CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQDG	APQDG_EYECATCHER	

## QUERY TYPE=CAGROUP の出力

RCVINF-U 出力  
(DSPAPQRI)

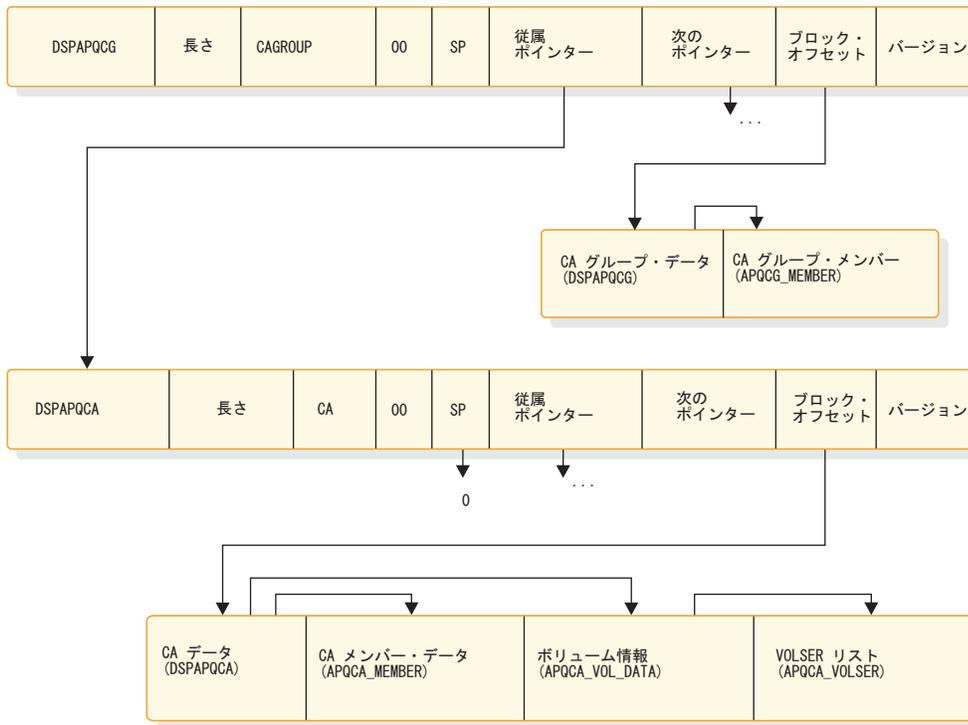


図 21. QUERY TYPE=CAGROUP の出力のフォーマット

CA ブロックは、CAINFO=YES が指定され、変更累積のレコードが RECON 内に存在する場合にのみ戻されます。

## DSPAPQCG の DSECT

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	48	DSPAPQCG	
0	(0)	CHARACTER	8	APQCG_GROUPNAME	Group name
8	(8)	UNSIGNED	4	APQCG_MEMBERINFO	Offset to group member list
12	(C)	UNSIGNED	2	APQCG_MEMBERLEN	Length of group member entry
14	(E)	SIGNED	2	APQCG_MEMBERCOUNT	Number of group members
16	(10)	SIGNED	4	*(2)	Reserved
24	(18)	SIGNED	2	APQCG_GRPMAX	Maximum number of CAs that may be predefined for this CA group
26	(1A)	SIGNED	2	APQCG_AVAILCA#	Number of available CA data sets for this group
28	(1C)	SIGNED	2	APQCG_USEDCA#	Number of used CA data sets
30	(1E)	BIT(8)	1	APQCG_FLAGS	Flags
		1... ..		APQCG_REUSE	Reuse CA data sets
31	(1F)	CHARACTER	1	*	Reserved
32	(20)	CHARACTER	8	APQCG_CAJCL	GENJCL CAJCL member name
40	(28)	CHARACTER	8	APQCG_DFJCL	DEFLTJCL member name
48	(30)	SIGNED	2	APQCG_RECOVPD	Retention Period (Version=4.0)

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	16	APQCG_MEMBER	List of group members

0	(0)	CHARACTER	8	APQCG_DBNAME	Database name
8	(8)	CHARACTER	8	*	
8	(8)	CHARACTER	8	APQCG_DDNAME	DD name or
8	(8)	CHARACTER	8	APQCG_AREANAME	AREA name

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQCG	APQCG_EYECATCHER	

**DSPAPQCA の DSECT**

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	96	DSPAPQCA	
0	(0)	CHARACTER	8	APQCA_GROUPNAME	Group name
8	(8)	UNSIGNED	4	APQCA_MEMBERINFO	Offset to group member list
12	(C)	UNSIGNED	2	APQCA_MEMBERLEN	Length of each member entry
14	(E)	SIGNED	2	APQCA_MEMBERCOUNT	Number of group members
16	(10)	UNSIGNED	4	APQCA_VOLINFO	Offset to volume information
20	(14)	SIGNED	4	*	Reserved
24	(18)	CHARACTER	44	APQCA_DSN	Data set name
68	(44)	CHARACTER	12	APQCA_STOPTIME	Packed decimal date/time - for predefined datasets, represents record creation time. Otherwise, it is the stoptime of the last logtape volume used as input to the Change Accumulation utility that produced this CA as output. If the CA run included an 'incomplete log subset' it is the start time of the first truncated log volume.
80	(50)	CHARACTER	12	APQCA_RUNTIME	CA run time
92	(5C)	BIT(8)	1	APQCA_FLAGS	Flags
		1... ..		APQCA_ERROR	Error on data set
		.1.. ..		APQCA_SUBSET	Subset of logs used for CA
		..1. ....		APQCA_COMMAND	SUBSET/COMP has been set or reset with an external cmd
		...1 ....		APQCA_AVAIL	Available CA indicator
93	(5D)	CHARACTER	3	*	Reserved
96	(60)	UNSIGNED	4	APQCA_OFFUD	Offset to user data (VERSION=4.0)
100	(64)	UNSIGNED	2	APQCA_LENUD	Length of user data (VERSION=4.0)

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	80	APQIC_USER	Data for user IC
0	(0)	CHARACTER	80	APQRV_UDATA	User supplied data (VERSION=4.0)

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	56	APQCA_MEMBER	List of group members
0	(0)	CHARACTER	8	APQCA_MEM_DBNAME	Database name
8	(8)	CHARACTER	8	*	
8	(8)	CHARACTER	8	APQCA_MEM_DDNAME	DD name or
8	(8)	CHARACTER	8	APQCA_MEM_AREANAME	AREA name
16	(10)	UNSIGNED	4	APQCA_MEM_DSSN	Data Set sequence number
20	(14)	UNSIGNED	4	APQCA_MEM_USID	USID of last change accumulated
24	(18)	CHARACTER	8	APQCA_MEM_LRID	LRID of last change accumulated
32	(20)	CHARACTER	12	APQCA_MEM_PURGETIME	Purge time
44	(2C)	CHARACTER	6	APQCA_MEM_LSN	Lock sequence number

50	(32)	BIT(8)	1	APQCA_MEM_FLAGS	Member flags
		1... ..		APQCA_MEM_NOCHG	No changes accumulated
		.1.. ..		APQCA_MEM_INDOUBT	Indoubt EEQEs accumulated
		..1. ....		APQCA_MEM_INCOMP	Incomplete CA
51	(33)	CHARACTER	5	*	Reserved

## QUERY TYPE=GSGROUP の出力

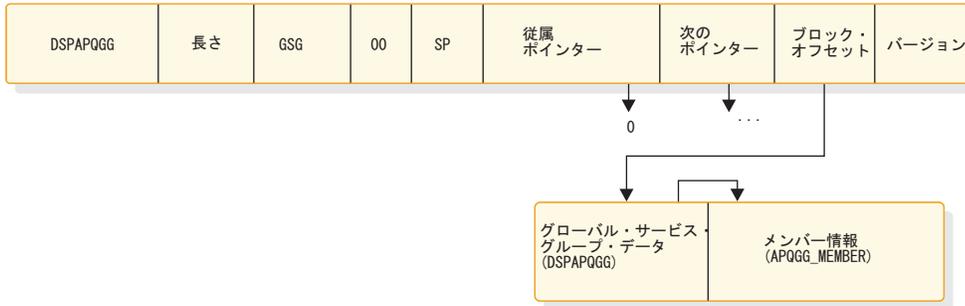


図 22. QUERY TYPE=GSGROUP の出力のフォーマット

## DSPAPQGG の DSECT

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	80	DSPAPQGG	
0	(0)	CHARACTER	8	APQGG_GROUPNAME	Group name
8	(8)	UNSIGNED	4	APQGG_MEMBERINFO	Offset to group member list
12	(C)	UNSIGNED	2	APQGG_MEMBERLEN	Length of group member entry
14	(E)	SIGNED	2	APQGG_MEMBERCOUNT	Number of group members
16	(10)	SIGNED	4	*(2)	Reserved
24	(18)	UNSIGNED	4	APQGG_SDSN#	SLDS DSN sequence number
28	(1C)	UNSIGNED	4	APQGG_PTOKEN	Current PRILOG token
32	(20)	UNSIGNED	4	APQGG_MPTOK	Min required PRILOG token
36	(24)	UNSIGNED	4	APQGG_TTOKN	Planned takeover token
40	(28)	CHARACTER	12	APQGG_LOGTIME	Current log start time
52	(34)	CHARACTER	12	APQGG_HITIME	Highest time ever received from the active site
64	(40)	CHARACTER	8	APQGG_TSNAME	Tracking subsystem ID
72	(48)	BIT(8)	1	APQGG_FLAGS	
		1... ..		APQGG_ACTTKO	Active takeover in progress
		.1.. ..		APQGG_1STSIGNON	Indicates the first signon of an active subsystem after an RSR takeover
		..1. ....		APQGG_TRKTKO	Tracker takeover in progress
		...1 .....		APQGG_RESET	RESET.GSG issued
73	(49)	CHARACTER	7	*	Reserved

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	16	APQGG_MEMBER	List of group members
0	(0)	CHARACTER	8	APQGG_SGNAME	Service group name
8	(8)	CHARACTER	1	APQGG_SGROLE	Role 'A' OR 'T'
9	(9)	BIT(8)	1	APQGG_SGFLAGS	Flags
		1... ..		APQGG_SGLOCAL	1 = local SG
10	(A)	CHARACTER	6	*	Reserved

### CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQGG	APQGG_EYECATCHER	

関連概念:

407 ページの『DSPAPI マクロのマクロ形式』

関連資料:

431 ページの『第 30 章 DBRC 照会要求』

➡ DBRC 要求戻りコード (メッセージおよびコード)

## ログ照会要求 (TYPE=LOG)

サブシステムの特定のインスタンスのログ情報を RECON から取得するには、ログ照会 (DSPAPI FUNC=QUERY TYPE=LOG) 要求を使用します。

要求が非 RSR ログまたは RSR アクティブ・サブシステム・ログに関するものである場合、以下の RECON データ・セットからの情報が戻されます。

- PRILOG
- LOGALL
- SECLOG (該当する場合)
- PRISLDS (該当する場合)
- SECSLDS (該当する場合)

要求が RSR トラッキング・サブシステム・ログに関するものである場合、以下の RECON データ・セットからの情報が戻されます。

- PRITSLDS
- SECTSLDS (該当する場合)

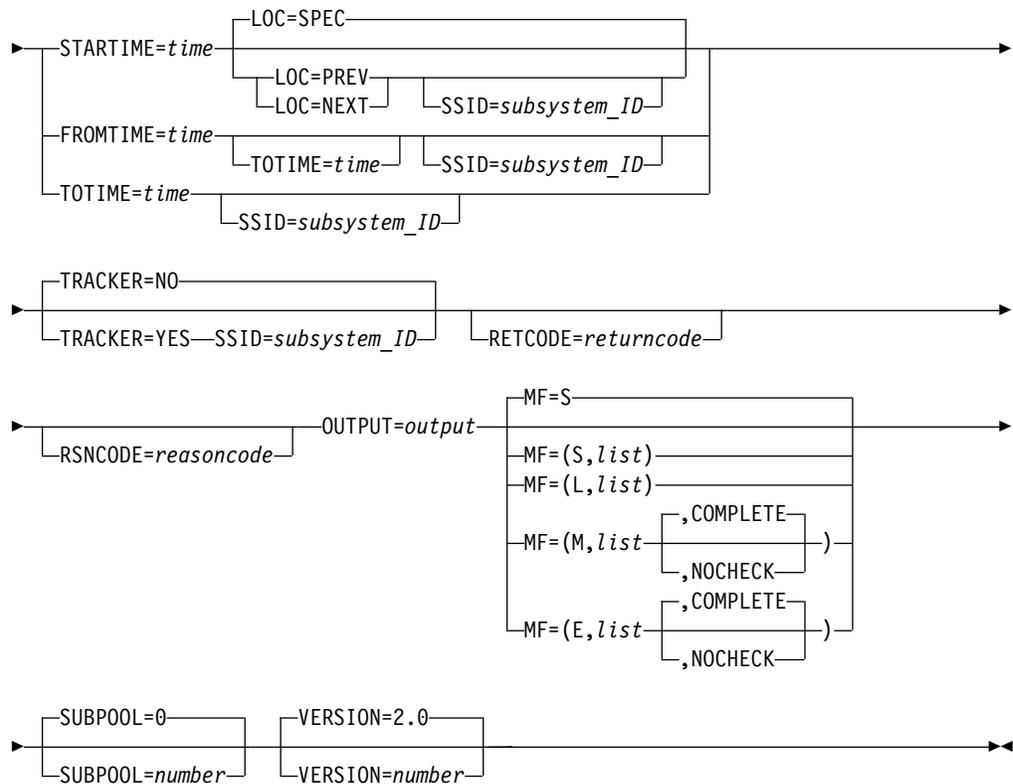
ログ照会要求では、指定された時刻範囲内で開始されたサブシステムのログ情報を返すこともできます。この要求は、特定のサブシステムに対しても使用できます。

以下のサブセクションがあります。

- 『TYPE=LOG 照会要求の構文』
- 478 ページの『TYPE=LOG 照会要求のパラメーター』
- 480 ページの『TYPE=LOG 照会要求の戻りコードと理由コード』
- 483 ページの『TYPE=LOG 照会要求の出力』
- 483 ページの『ログ情報の出力』
- 483 ページの『DSECT の DSPAPQLI』
- 484 ページの『PRILOG、SECLOG、PRISLDS、SECSLDS、PRITSLDS、および SECTSLDS の TYPE=LOG 出力』
- 484 ページの『DSECT の DSPAPQLG』
- 486 ページの『LOGALL の TYPE=LOG 出力』
- 486 ページの『DSECT の DSPAPQLA』

### TYPE=LOG 照会要求の構文

→ name DSPAPI—FUNC=QUERY—TYPE=LOG—TOKEN=address →



## TYPE=LOG 照会要求のパラメーター

名前 ユーザーが指定できるオプションのシンボルです。使用する場合は、1 桁目から始めます。

### TYPE=LOG

ログ情報が要求されることを指定します。

### TOKEN=address | (2 - 12)

FUNC=STARTDBRC 要求で戻された 4 バイト・フィールドのアドレスを指定します。

### STARTIME=time | (2-12)

要求されたログの開始時刻を含むタイム・スタンプ・フィールドを指定します。この時刻は、UTC フォーマットのパック 10 進数のタイム・スタンプです。

### FROMTIME=time | (2-12)

要求するログを、そのサブシステムが指定時刻またはそれより後に開始されたものに制限するタイム・スタンプ・フィールドを指定します。この時刻は、UTC フォーマットのパック 10 進数のタイム・スタンプです。

このパラメーターを指定する場合、DBRC API の最小バージョン番号 VERSION=2.0 も指定する必要があります。

### TOTIME=time | (2-12)

要求するログを、そのサブシステムが指定時刻またはそれより前に開始されたものに制限するタイム・スタンプ・フィールドを指定します。この時刻は、UTC フォーマットのパック 10 進数のタイム・スタンプです。このパラメーターは、FROMTIME パラメーターと一緒に使用できます。

このパラメーターを指定する場合、API の最小バージョン番号  
VERSION=2.0 も指定する必要があります。

**SSID=***subsystem\_ID* | (2-12)

照会されるログのサブシステム名を指定します。

SSID は、RSR トラッキング・サブシステム・ログ照会 (TRACKER=YES) に必要です。それ以外の場合は、特定のサブシステムの前のログまたは次のログを要求する場合のみ、SSID パラメーターを指定できます。例えば、LOC で前のログまたは次のログを指定した場合や、FROMTIME または TOTIME パラメーターを使用して特定のサブシステムに対してログ範囲を指定した場合などです。

**LOC=**PREV | NEXT | SPEC

その要求が、指定された開始時刻のログ (LOC=SPEC)、指定された開始時刻より前の開始時刻のログ (LOC=PREV)、指定された開始時刻より後の開始時刻のログ (LOC=NEXT) を対象とすることを指定します。STARTIME パラメーターは検索のベースとして使用されますが、RECON 内のログの開始時刻でなくてもかまいません。

LOC=SPEC はデフォルトです。

**TRACKER=**NO | YES

RSR トラッキング・サブシステムに関するログ情報が要求されているかどうかを指定します。TRACKER=YES を指定すると、RSR トラッカーのログ情報が要求されます。TRACKER=NO を指定すると、非 RSR サブシステム、または RSR システムのアクティブ・サブシステムからのログ情報が要求されます。TRACKER=NO がデフォルトです。

**MF=**S | L | M | E

要求のマクロ形式を指定します。

**OUTPUT=***output* | (2 - 12)

ログ情報ブロック (おそらくチェーンになっている) の最初のブロックを示すポインターを受け取るフィールドを指定します。

出力が作成されなかった場合、出力アドレスはゼロです。この状態は、RECON 内に要求を満たす情報がない場合、または出力を作成できるようになる前にエラーが発生した場合に起こります。

出力ブロックのストレージは、呼び出し元によって事前割り振りされません。DBRC はこれらのブロック用に指定されたサブプールからストレージを取得します。呼び出し元は、バッファ解放サービス (DSPAPI FUNC=RELBUF) を使用してこのストレージを解放し、戻された出力アドレスを指定する必要があります。

**SUBPOOL=** 0 | *number*

取得されるストレージのサブプール番号を指定します。このパラメーターを指定しない場合、FUNC=STARTDBRC 要求によって指定されたサブプールがデフォルトになります。それ以外の場合は、サブプール 0 がデフォルトになります。

**RETCODE=***returncode* | (2-12)

シンボルとして指定する場合、戻りコードを受け取るために、ストレージのワードを示すラベルを指定します。レジスターとして指定する場合、戻りコ

ードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RETCODE が指定されたかどうかにかかわらず、レジスター 15 は戻りコードを含んでいます。

#### RSNCODE=*reasoncode* | (2-12)

シンボルとして指定する場合、理由コードを受け取るために、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、理由コードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RSNCODE が指定されたかどうかにかかわらず、レジスター 0 は理由コードを含んでいます。

#### VERSION=2.0 | *number*

このマクロによって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターはマクロ処理によって受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

FUNC=QUERY TYPE=LOG 要求の有効なバージョン番号は、1.0 および 2.0 です。

### TYPE=LOG 照会要求の戻りコードと理由コード

表 93. TYPE=LOG 照会要求の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
警告	X'00000008'	X'D8400001'	要求されたログ・タイプのログ・レコード (PRILOG または PRITSLDS) が存在しません。要求は、FROMTIME または TOTIME で指定された時刻範囲内の前のログまたは次のログに対するものでした。
	X'00000008'	X'D8400002'	要求されたログ・タイプの指定されたログ・レコード (PRILOG または PRITSLDS) が存在しません。
重大エラー	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。
	X'0000000C'	X'D8400100'	セキュリティ・エラー。SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーは要求の実行を許可されていないと判断しました。

表 93. TYPE=LOG 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
ストレージ・エラー	X'00000028'	X'D8400001'	LOGINFO ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8400002'	PRILOG、SECLOG、PRISLDS、SECSLDS、PRITSLDS、または SECTSLDS ブロック用のストレージの取得中にエラーが発生しました。
	X'00000028'	X'D8400003'	LOGALL ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D9100001'	要求の処理中にエラーが発生しました。DBRCはこの時点までに取得したストレージを解放します。しかし、ストレージを解放しようとして別のエラーが検出されました。
内部エラー	X'0000002C'	X'D8000001'	RECON を開くときに障害が発生しました。
	X'0000002C'	X'D8400001'	要求されたログ・タイプの前のログ・レコードまたは次のログ・レコード (PRILOG または PRITSLDS) の検索時に障害が発生しました。
	X'0000002C'	X'D8400002'	要求されたログ・タイプの指定されたログ・レコード (PRILOG または PRITSLDS) の検索時に障害が発生しました。
	X'0000002C'	X'D8400003'	対応する SECTSLDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8400004'	PRILOG レコードに対応する LOGALL レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8400005'	PRILOG レコードに対応する LOGALL レコードが存在しません。
	X'0000002C'	X'D8400006'	対応する SECLOG レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8400007'	対応する PRISLDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8400008'	オンライン・ログの PRISLDS レコードが存在しません。
	X'0000002C'	X'D8400009'	対応する SECSLDS レコードの検索時に障害が発生しました。
X'0000002C'	X'D840000A'	DBRC タイム・サービスによる FROMTIME の処理時に障害が発生しました。	
パラメーター・エラー	X'00000030'	X'C9000001'	API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN フィールド・アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

表 93. TYPE=LOG 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'C9000003'	RETCODE フィールド・アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE フィールド・アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000005'	OUTPUT フィールド・アドレスが無効です。OUTPUT アドレスを含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000010'	SSID フィールド・アドレスが無効です。SSID を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000011'	STARTIME フィールド・アドレスが無効です。STARTIME を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000012'	FROMTIME フィールド・アドレスが無効です。FROMTIME パラメーターを含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000013'	TOTIME フィールド・アドレスが無効です。TOTIME パラメーターを含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'D8000001'	OUTPUT パラメーターが欠落しているか無効です。
	X'00000030'	X'D8000002'	TYPE パラメーターに指定された値が無効です。
	X'00000030'	X'D8400001'	STARTIME、FROMTIME、または TOTIME パラメーターは必須です。
	X'00000030'	X'D8400002'	SSID は TRACKER=YES の場合は必須です。

表 93. TYPE=LOG 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'D8400003'	特定のアクティブ・ログの照会時 (LOC=SPEC および TRACKER=NO) に、SSID パラメーターは指定できません。
	X'00000030'	X'D8400006'	FROMTIME パラメーター値は、TOTIME パラメーター値より小さい値にする必要があります。
	X'00000030'	X'D8400007'	STARTIME は、FROMTIME   TOTIME を使用する場合は指定できません。
	X'00000030'	X'D8400008'	LOC は、FROMTIME   TOTIME を使用する場合は指定できません。
	X'00000030'	X'D8400010'	FROMTIME パラメーターに渡された値が無効な時刻です。

## TYPE=LOG 照会要求の出力

以下の図に、QUERY TYPE=LOG 要求からの出力フォーマットを示します。図の後のサンプル DSECT では、ストレージ・ブロックのフィールドとそれらのフィールド相互の関係を詳細に説明します。

### ログ情報の出力

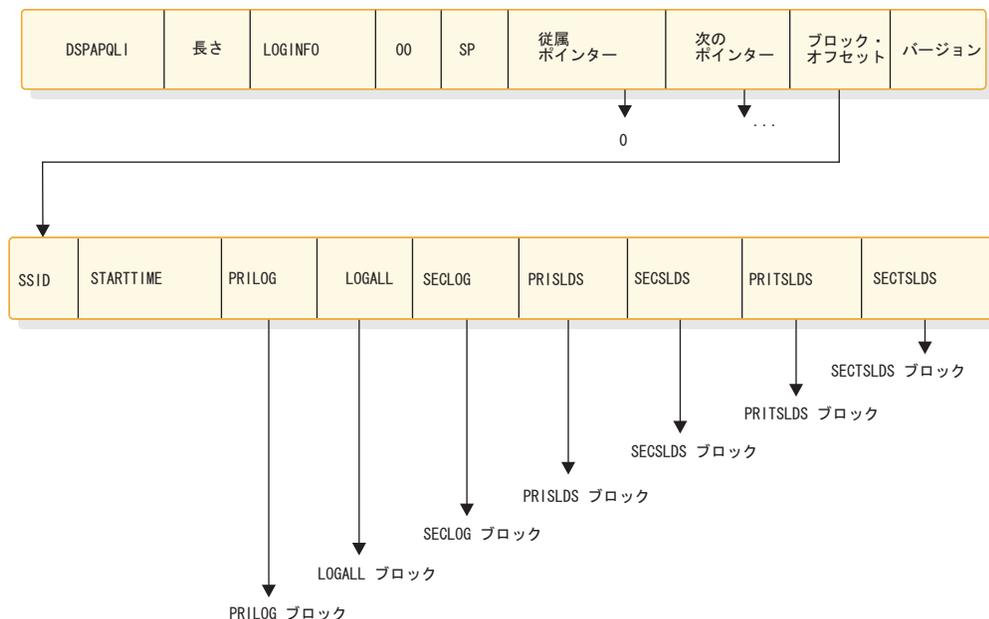


図 23. QUERY TYPE=LOG ログ情報の出力のフォーマット

## DSECT の DSPAPQLI

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	48	DSPAPQLI	

0	(0)	CHARACTER	8	APQLI_SSID	Log SSID
8	(8)	CHARACTER	12	APQLI_STARTTIME	Log start time
20	(14)	ADDRESS	4	APQLI_PRILOGPTR	ptr to PRILOG block
24	(18)	ADDRESS	4	APQLI_LOGALLPTR	ptr to LOGALL block
28	(1C)	ADDRESS	4	APQLI_SECLGPTR	ptr to SECLG block
32	(20)	ADDRESS	4	APQLI_PRISLDSPTR	ptr to PRISLDS block
36	(24)	ADDRESS	4	APQLI_SECSLDSPTR	ptr to SECSLDS block
40	(28)	ADDRESS	4	APQLI_PRITSLDSPTR	ptr to PRITSLDS block
44	(2C)	ADDRESS	4	APQLI_SECTSLDSPTR	ptr to SECTSLDS block

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQLI	APQLI_EYECATCHER	

**PRILOG、SECLOG、PRISLDS、SECSLDS、PRITSLDS、および SECTSLDS の TYPE=LOG 出力**

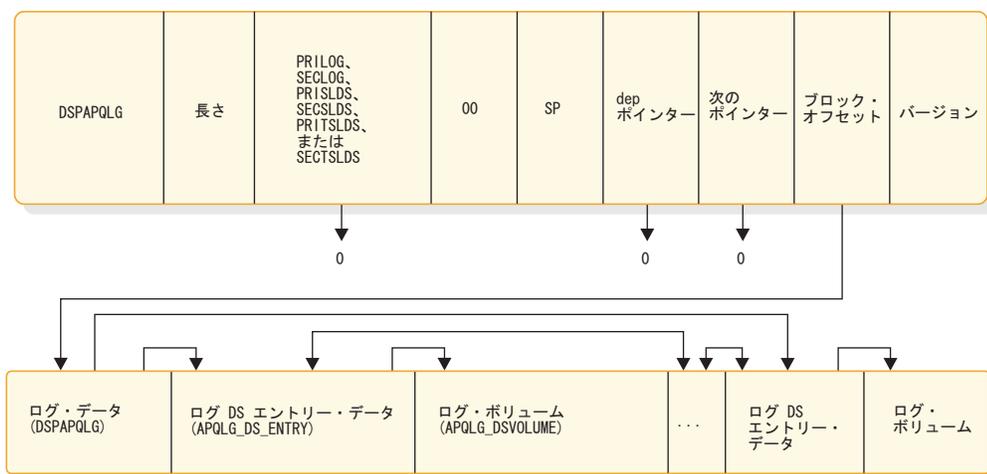


図 24. PRILOG、SECLOG、PRISLDS、SECSLDS、PRITSLDS、および SECTSLDS の TYPE=LOG 出力の形式

**DSECT の DSPAPQLG**

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	96	DSPAPQLG	
0	(0)	UNSIGNED	4	APQLG_FIRSTLOGDS	Offset to first log DS entry
4	(4)	UNSIGNED	4	APQLG_LASTLOGDS	Offset to last log DS entry
8	(8)	SIGNED	4	*(2)	Reserved
16	(10)	CHARACTER	8	APQLG_SSID	Subsystem ID
24	(18)	CHARACTER	12	APQLG_STARTTIME	Log start time
36	(24)	CHARACTER	12	APQLG_ENDTIME	Log end time
48	(30)	SIGNED	4	APQLG_DSNCOUNT	Number of data sets
52	(34)	UNSIGNED	1	APQLG_RELVL	Log Release Level
		1... ..		APQLG_ONLINE	Online log - PRILOG and SECLOG only
		.1.. ..		APQLG_LSTAR	Last OLDS has been archived - PRILOG and PRITSLDS only
		..1. ....		APQLG_LSTNA	Last OLDS has not been archived - PRILOG and PRITSLDS only
		...1 ....		APQLG_GAP	There is a gap in this log
		.... 1...		APQLG_PRGAP	There is a gap in a prev log
53	(34)	UNSIGNED	1	APQLG_FLAGS1	Flags
		1... ..		APQLG_ONLINE	Online log - PRILOG and SECLOG only
		.1.. ..		APQLG_LSTAR	Last OLDS has been archived - PRILOG and PRITSLDS only
		..1. ....		APQLG_LSTNA	Last OLDS has not been archived - PRILOG and PRITSLDS only
		...1 ....		APQLG_GAP	There is a gap in this log
		.... 1...		APQLG_PRGAP	There is a gap in a prev log
		.... .1..		APQLG_BPE	BPE-based subsystem (VERSION=4.0)
54	(36)	BIT(8)	1	APQLG_FLAGS2	Flags

DECIMAL	HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
		1... ....		APQLG_TRKNG	Tracking log data set
		.1.. ....		APQLG_NTERM	IMS subsystem has terminated normally
		..1. ....		APQLG_BKLOG	Batch backout log
55	(37)	CHARACTER	1	*	Reserved
56	(38)	CHARACTER	8	APQLG_FIRSTLRID	Id of first log record
64	(40)	UNSIGNED	4	APQLG_PTOKEN	PRIOLOG token
68	(44)	CHARACTER	8	APQLG_GSGNAME	GSG name
76	(4C)	CHARACTER	12	APQLG_CHKPT0	Checkpoint 0 time
88	(58)	CHARACTER	8	*	Reserved OFFSET OFFSET
0	(0)	STRUCTURE	120	APQLG_DS_ENTRY	Data set entry
0	(0)	UNSIGNED	4	APQLG_DS_NEXT	Offset to next log DS entry
4	(4)	UNSIGNED	4	APQLG_DS_PREV	Offset to prev log DS entry
8	(8)	UNSIGNED	4	APQLG_DS_VOLINFO	Offset to DS volume info
12	(C)	CHARACTER	44	APQLG_DS_DSNAME	Data set name
56	(38)	CHARACTER	12	APQLG_DS_STARTTIME	DS start time
68	(44)	CHARACTER	12	APQLG_DS_ENDTIME	DS end time
80	(50)	BIT(8)	1	APQLG_DS_FLAGS1	Flags
		1... ....		APQLG_DS_ERR I/O	Error
		.1.. ....		APQLG_DS_DUMMY	Log compressed, 1st DS dummy
		..1. ....		APQLG_DS_RSTBG	Restart begin
		.... 1...		APQLG_DS_RSTEN	Restart end
		.... 1..		APQLG_DS_COLD	Cold start
		.... .1..		APQLG_DS_NOBMP	ERE NOBMP
		.... ..1.		APQLG_DS_SAVER	Backout UORs saved
		.... ...1		APQLG_DS_NOID	Backouts not identified
81	(51)	BIT(8)	1	APQLG_DS_FLAGS2	Flags
		1... ....		APQLG_DS_TRKARCH	Tracking log DS archived
		.1.. ....		APQLG_DS_TRKFEOV	Tracking log closed FEOV
82	(52)	CHARACTER	2	APQLG_DS_DFLG3	Reserved
84	(54)	CHARACTER	8	APQLG_DS_FLRID	First log record ID
92	(5C)	CHARACTER	8	APQLG_DS_LLRID	Last log record ID
100	(64)	UNSIGNED	4	APQLG_DS_LASTBLKSEQNO	Last block seq number
104	(68)	CHARACTER	8	APQLG_DS_UNITTYPE	Unit type name
112	(70)	UNSIGNED	2	APQLG_DS_FILESEQ	File sequence no
114	(72)	UNSIGNED	2	APQLG_DS_VOLCOUNT	Number of volumes
116	(74)	UNSIGNED	1	APQLG_DS_CKPTCOUNT	Number of chkpts on DSN
117	(75)	BIT(8)	1	APQLG_DS_CHKPTTYPES	CHKPT types.
		1... ....		APQLG_DS_SIMPL	SIMPLE checkpoint
		.1.. ....		APQLG_DS_SNAPQ	SNAPQ checkpoint
		..1. ....		APQLG_DS_DUMPQ	DUMPQ checkpoint
		.... 1...		APQLG_DS_PURGE	PURGE checkpoint
		.... ..1..		APQLG_DS_FREEZ	FREEZE checkpoint
118	(76)	UNSIGNED	2	*	Reserved
OFFSET	OFFSET				
DECIMAL	HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	48	APQLG_DSVOLUME	Volume information
0	(0)	UNSIGNED	4	APQLG_DSVOL_NEXT	Offset to next volume
4	(4)	CHARACTER	6	APQLG_DSVOL_SER	VOLSER
10	(A)	UNSIGNED	1	APQLG_DSVOL_CKPTCT	Volume chkpt count
11	(B)	CHARACTER	1	*	Reserved
12	(C)	CHARACTER	12	APQLG_DSVOL_ENDTIME	Volume end time
24	(18)	CHARACTER	12	APQLG_DSVOL_CPTID	Checkpoint ID
36	(24)	CHARACTER	6	APQLG_DSVOL_LOCKSN	Lock Sequence Number
42	(2A)	CHARACTER	6	*	Reserved

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQLG	APQLG_EYECATCHER	

## LOGALL の TYPE=LOG 出力

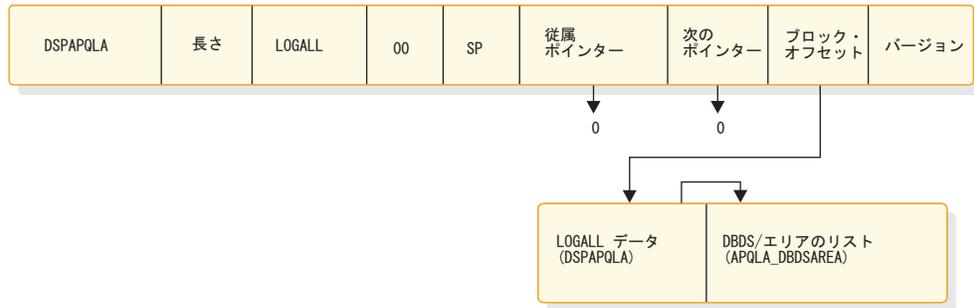


図 25. LOGALL の QUERY TYPE=LOG 出力のフォーマット

## DSECT の DSPAPQLA

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	48	DSPAPQLA	
0	(0)	UNSIGNED	4	APQLA_DBDSAREAINFO	Offset to allocated DBDS/Area list
4	(4)	SIGNED	4	* (3)	Reserved
16	(10)	CHARACTER	12	APQLA_PRILOGTIME	PRILOG time
28	(1C)	BIT (8)	1	APQLA_FLAGS	Flags
29	(1D)	UNSIGNED	3	APQLA_DBDSAREAACCOUNT	Number of DBDS/Areas allocated on this log
32	(20)	UNSIGNED	4	APQLA_DBDSAREALEN	Length of DBDS/Area entry
36	(24)	CHARACTER	12	APQLA_EARLIESTALLOC	Earliest ALLOC time for this log

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	32	APQLA_DBDSAREA	List of allocated DBDSs and areas
0	(0)	CHARACTER	8	APQLA_DBNAME	Database name
8	(8)	CHARACTER	8	*	
8	(8)	CHARACTER	8	APQLA_DDNAME	DD name or
8	(8)	CHARACTER	8	APQLA_AREANAME	AREA name
16	(10)	CHARACTER	12	APQLA_FIRSTALLOC	Earliest ALLOC time for this DBDS/Area on this log
28	(1C)	SIGNED	2	APQLA_ALLNO	Number of ALLOCs for this DBDS/Area on this log
30	(1E)	CHARACTER	2	*	Reserved

### CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQLA	APQLA_EYECATCHER	

### 関連概念:

407 ページの『DSPAPI マクロのマクロ形式』

### 関連資料:

431 ページの『第 30 章 DBRC 照会要求』

## OLDS 照会要求 (TYPE=OLDS)

特定のサブシステムまたは全サブシステムのオンライン・ログ・データ・セット情報を RECON から取得するには、OLDS 照会 (DSPAPI FUNC=QUERY TYPE=OLDS) 要求を使用します。



出力ブロックのストレージは、呼び出し元によって事前割り振りされません。DBRC はこれらのブロック用に指定されたサブプールからストレージを取得します。呼び出し元は、バッファ解放サービス (DSPAPI FUNC=RELBUF) を使用してこのストレージを解放し、戻された出力アドレスを指定する必要があります。

**SUBPOOL= 0 | number**

取得されるストレージのサブプール番号を指定します。このパラメーターを指定しない場合、FUNC=STARTDBRC 要求によって指定されたサブプールがデフォルトになります。それ以外の場合は、サブプール 0 がデフォルトになります。

**RETCODE=returncode | (2-12)**

シンボルとして指定する場合、戻りコードを受け取るために、ストレージのワードを示すラベルを指定します。レジスターとして指定する場合、戻りコードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RETCODE が指定されたかどうかにかかわらず、レジスター 15 は戻りコードを含んでいます。

**RSNCODE=reasoncode | (2-12)**

シンボルとして指定する場合、理由コードを受け取るために、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、理由コードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RSNCODE が指定されたかどうかにかかわらず、レジスター 0 は理由コードを含んでいます。

**VERSION=2.0 | number**

この要求によって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターはマクロ処理によって受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

FUNC=QUERY TYPE=OLDS 要求の有効なバージョン番号は、1.0 および 2.0 (デフォルト) です。

**TYPE=OLDS 照会要求の戻りコードと理由コード**

表 94. TYPE=OLDS 照会要求の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
警告	X'00000008'	X'D8500001'	PRIOLDS レコードが存在しません。
重大エラー	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。

表 94. TYPE=OLDS 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。
	X'0000000C'	X'D8500100'	セキュリティー・エラー。SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーは要求の実行を許可されていないと判断しました。
ストレージ・エラー	X'00000028'	X'D8500001'	OLDS ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D9100001'	要求の処理中にエラーが発生しました。DBRC はこの時点までに取得したストレージを解放します。しかし、ストレージを解放しようとして別のエラーが検出されました。
内部エラー	X'0000002C'	X'D8000001'	RECON を開くときに障害が発生しました。
	X'0000002C'	X'D8500001'	最初の PRIOLDS レコードまたは指定された PRIOLDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8500002'	対応する SECOLDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8500003'	次の PRIOLDS レコードの検索時に障害が発生しました。
パラメーター・エラー	X'00000030'	X'C9000001'	API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN フィールド・アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000003'	RETCODE フィールド・アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE フィールド・アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000005'	OUTPUT フィールド・アドレスが無効です。OUTPUT アドレスを含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

表 94. TYPE=OLDS 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'C9000010'	SSID フィールド・アドレスが無効です。SSID を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'D8000001'	OUTPUT パラメーターが欠落しているか無効です。
	X'00000030'	X'D8500100'	ワイルドカードを使用するとき、アスタリスクの前に英字が少なくとも 1 つ必要です。
	X'00000030'	X'D8500101'	ワイルドカードを使用するとき、最後の文字はアスタリスクである必要があります。

## TYPE=OLDS 照会要求の出力

以下の図に、QUERY TYPE=OLDS 要求からの出力フォーマットを示します。以下のサンプル DSECT が、ストレージ・ブロックのフィールドとそれらのフィールドの相互関係を詳細に説明します。

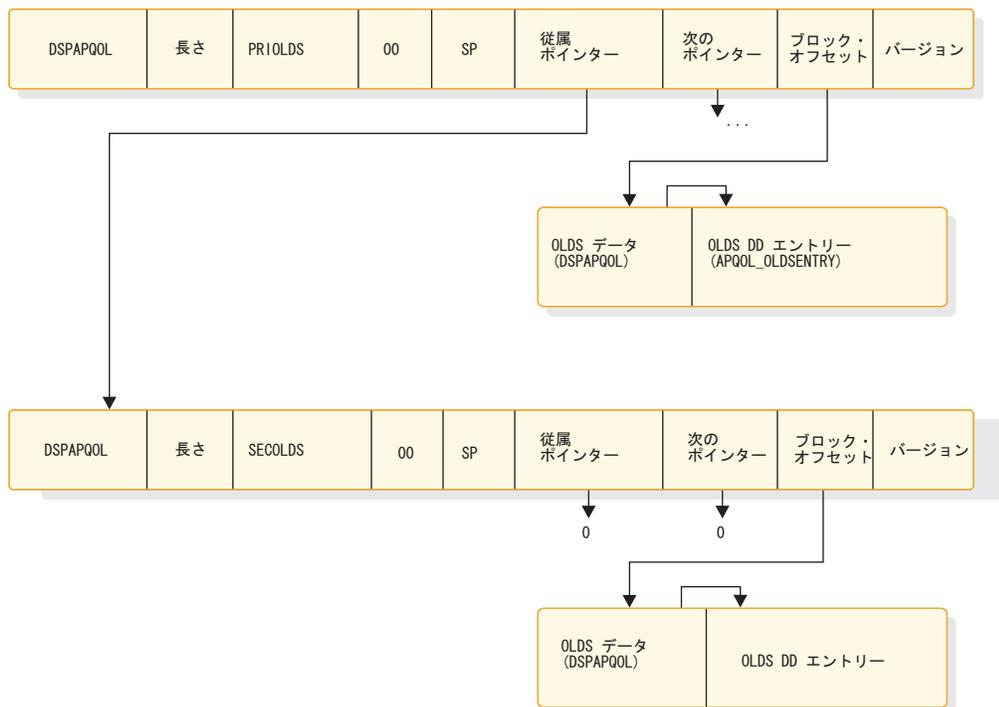


図 26. QUERY TYPE=OLDS の出力のフォーマット

## DSPAPQOL の DSECT

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	48	DSPAPQOL	
0	(0)	UNSIGNED	4	APQOL_OLDSINFO	Offset to OLDS list

4	(4)	SIGNED	4	*(3)	Reserved
16	(10)	CHARACTER	8	APQOL_SSID	Subsystem ID
24	(18)	UNSIGNED	2	APQOL_OLDSLEN	Length of OLDS entry
26	(1A)	SIGNED	2	APQOL_OLDSCOUNT	Number of OLDS entries
28	(1C)	CHARACTER	12	APQOL_CHKPT0	Checkpoint 0 time
40	(28)	CHARACTER	8	*	Reserved

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	128	APQOL_OLDSETRY	OLDS entry
0	(0)	CHARACTER	8	APQOL_DDNAME	OLDS DD name
8	(8)	CHARACTER	44	APQOL_DSNAM	OLDS data set name
52	(34)	CHARACTER	12	APQOL_OPENTIME	OLDS open time
64	(40)	CHARACTER	12	APQOL_CLOSETIME	Close time
76	(4C)	CHARACTER	12	APQOL_PRILOGTIME	Start time of associated PRILOG
88	(58)	CHARACTER	8	APQOL_FLSN	LSN of first record in OLDS
96	(60)	CHARACTER	8	APQOL_LLSN	LSN of last record in OLDS
104	(68)	BIT(8)	1	APQOL_FLAG1	
		1... ..		APQOL_RSTBG	Restart begin
		.1.. ..		APQOL_RSTEN	Restart end
		..1. ....		APQOL_COLD	Cold start
		...1 ....		APQOL_NOBMP	ERE NOBMP
		.... 1...		APQOL_SAVER	Backout UORs saved
		.... .1..		APQOL_NOID	Backouts not identified
		.... .1.		APQOL_TRKNG	OLDS created by tracking SS
105	(69)	BIT(8)	1	APQOL_FLAG2	OLDS flags
		1111 ....		APQOL_STAT	OLDS status
		1... ..		APQOL_INUSE	OLDS is in use
		.1.. ....		APQOL_ARNEED	Archive needed
		..1. ....		APQOL_ARSCH	Archive scheduled (GENJCL)
		...1 ....		APQOL_ARSTD	Archive job started
		.... 1...		APQOL_CLERR	Close error on OLDS
		.... .1..		APQOL_FEOV	Force EOVS at archive
		.... .1.		APQOL_DUMMY	OLDS not used due to I/O err
		.... .1.		APQOL_PRVCE	Close error on previous OLDS
106	(6A)	UNSIGNED	1	APQOL_RELVL	Log release level
107	(6B)	UNSIGNED	1	APQOL_GAVER	GENJCL.ARCHIVE version
108	(6C)	BIT(32)	4	APQOL_BLOCKSEQNO	Block sequence number
112	(70)	CHARACTER	8	APQOL_ARJOB	Name of the archive job
					Generated by GENJCL.ARCHIVE
120	(78)	CHARACTER	6	APQOL_LOCKSEQNO	Lock sequence number
126	(7E)	CHARACTER	2	*	Reserved

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQOL	APQOL_EYECATCHER	

関連概念:

407 ページの『DSPAPI マクロのマクロ形式』

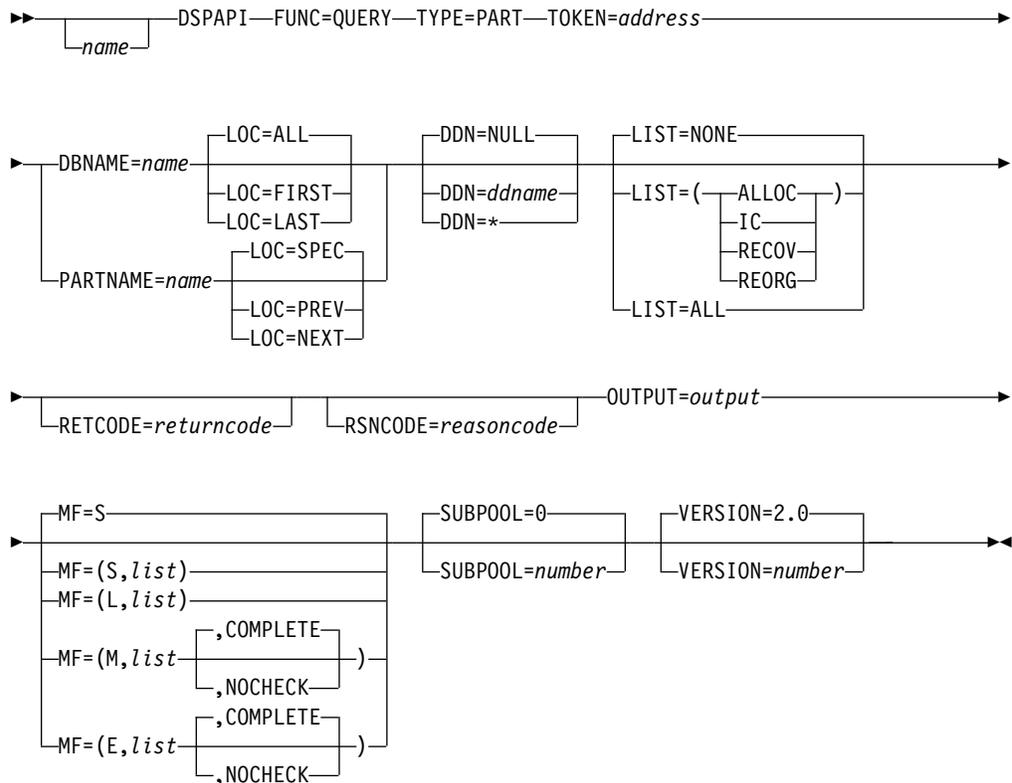
関連資料:

431 ページの『第 30 章 DBRC 照会要求』

## HALDB 区画照会要求 (TYPE=PART)

特定の HALDB 区画の情報を RECON データ・セットから検索するには、DSPAPI FUNC=QUERY TYPE=PART 要求を使用します。特定の DBDS、または区画内のすべての DBDS に関するデータ・セット情報を要求できます。さらに、オプションで、データ・セットのリカバリー関連の情報 (割り振り、イメージ・コピー、リカバリー、再編成情報など) を要求することもできます。

## TYPE=PART 照会要求の構文



## TYPE=PART 照会要求のパラメーター

名前 ユーザーが指定できるオプションのシンボルです。使用する場合は、1 桁目から始めます。

### TYPE=PART

HALDB 区画の RECON データ・セット情報を要求することを指定する必須パラメーターです。

TYPE=PART を指定する場合、DBRC API の最小バージョン番号 VERSION=2.0 も指定する必要があります。

### DBNAME=name | (2 - 12)

照会される区画の HALDB の名前を指定します。オプションで LOC パラメーターを指定できます。このパラメーターでは、指定する HALDB の最初の区画、最後の区画、またはすべての区画のいずれかを対象とすることを示します。HALDB が高位キーを使用する場合、要求は最低の高位キーまたは最高の高位キーを持つ区画を対象とします。それ以外の場合は、アルファベット順で最初または最後の区画が返されます。LOC=SPEC|PREV|NEXT は指定できません。

### PARTNAME=name | (2 - 12)

照会される HALDB 区画の名前を指定する必須パラメーターです。

LOC パラメーターは、以下のように対象の区画を示します。

- デフォルトの LOC=SPEC は、指定された区画に対する要求を示します。この区画は、使用可能または使用不可場合があります。

- LOC=PREV は、指定された区画より前の HALDB 内のアクティブ区画に対する要求を示します。HALDB が高位キーを使用する場合、要求はより低い高位キーを持つ区画を対象とします。それ以外の場合は、HALDB 内のアルファベット順で前の区画が返されます。指定された区画名は、RECON に登録済みのアクティブ区画の名前であることが必要です。これは、前の区画が同じ HALDB 内に存在することを保証するために必要です。
- LOC=NEXT は、指定された区画より後の HALDB 内のアクティブ区画に対する要求を示します。HALDB が高位キーを使用する場合、要求はより高い高位キーを持つ区画を対象とします。それ以外の場合は、HALDB 内のアルファベット順で次の区画が返されます。指定された区画名は、RECON に登録済みのアクティブ区画の名前であることが必要です。これは、次の区画が同じ HALDB 内に存在することを保証するために必要です。
- LOC=FIRST|LAST は、PARTNAME を使用する場合は指定できません。

このパラメーターを使用する場合、API の最小バージョン番号 VERSION=2.0 も指定する必要があります。

#### **DDN=NULL | (2 - 12) | \* | *ddname***

区画内の DBDS セットの DD 名を指定します。アスタリスク (引用符で囲まない \*) を指定することもできます。これにより、すべての DBDS の情報を要求することが示されます。特定の DDN が要求されるときに、要求が特定の区画を対象としない場合、区画のすべての DBDS に関する情報が返されます。

DDN=NULL は、DBDS 情報を要求しないことを示します。これはデフォルトです。

#### **LIST=( {ALLOC},{IC},{RECOV},{REORG}) | LIST=ALL | LIST=NONE**

指定した DBDS の照会出力に組み込むサポート情報のタイプを指定します。DDN を指定しない場合、区画内のすべての DBDS に関してこの情報が返されます。リストには、1 つ以上の特定の値が組み込まれます。それらの値には、ALLOC (割り振りレコード)、IC (イメージ・コピー・レコード)、RECOV (リカバリー・レコード)、または REORG (再編成レコード) があります。LIST=ALL は、すべてのサポート情報を要求します。

LIST=NONE は、サポート情報を要求しません。

#### **LOC=ALL | FIRST | LAST | PREV | NEXT | SPEC**

オプション・パラメーターは、その要求が、HALDB に対して RECON 内で定義済みの指定された区画 (SPEC)、あるいは最初のアクティブ区画、最後のアクティブ区画、前のアクティブ区画、次のアクティブ区画、またはすべてのアクティブ区画を対象とすることを指定します。LOC=SPEC は、アクティブまたは使用不可の区画を返す場合があります。DBNAME は、LOC=FIRST|LAST|ALL の場合は必須ですが、LOC=PREV|NEXT|SPEC の場合は指定できません。逆に、PARTNAME は LOC=PREV|NEXT|SPEC の場合は必須ですが、LOC=FIRST|LAST|ALL

の場合は指定できません。DBNAME を使用する場合、LOC=ALL がデフォルトです。PARTNAME を使用する場合は、LOC=SPEC がデフォルトです。

HALDB が高位キーを使用する場合、区画は高位キーの順序で返されます。それ以外の場合は、区画はアルファベット順で返されます。

**TOKEN=address | (2 - 12)**

API トークンを受信する 4 バイト・フィールドのアドレスを指定します。このトークンは、この FUNC=STARTDBRC 要求に関連する後続のすべての要求に組み込む必要があります。

**RETCODE=returncode | (2 - 12)**

戻りコードを受け取る、ストレージ内の場所を指定します。シンボルとして指定する場合、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。このパラメーターを指定しない場合、戻りコードはレジスター 15 に入れられます。

**RSNCODE=reasoncode | (2 - 12)**

理由コードを受け取る、ストレージ内の場所を指定します。シンボルとして指定する場合、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。このパラメーターを指定しない場合、理由コードはレジスター 0 に入れられます。

**OUTPUT=output | (2 - 12)**

区画の情報を含むブロック (おそらくチェーンになっている) の最初のブロックを示すポインターを受け取るフィールドを指定する必須パラメーターです。

出力が作成されなかった場合、出力アドレスはゼロです。これは、RECON データ・セット内に要求を満たす情報がない場合、または出力が作成される前にエラーが発生した場合に起こります。

出力ブロックのストレージは、呼び出し元によって事前割り振りされません。DBRC はこれらのブロック用に指定されたサブプールからストレージを取得します。呼び出し元は、バッファ解放サービス (DSPAPI FUNC=RELBUF) を使用してこのストレージを解放し、戻された出力アドレスを指定する必要があります。

**SUBPOOL= 0 | number**

取得されるストレージのサブプール番号を指定するオプション・パラメーターです。このパラメーターを指定しない場合、FUNC=STARTDBRC 要求によって指定されたサブプールがデフォルトになります。それ以外の場合は、サブプール 0 がデフォルトになります。

**MF=S | L | M | E**

要求のマクロ形式を指定します。

**VERSION=2.0 | number**

この要求によって生成されるパラメーター・リストのバージョン番号を指定するオプション・パラメーター。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターの処理は受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

デフォルトのバージョンは 2.0 です。

注: TYPE=PART を指定する場合、API の最小バージョン番号 VERSION=2.0 を指定する必要があります。

## TYPE=PART 照会要求の戻りコードと理由コード

以下の表には、TYPE=PART 照会要求の戻りコードと理由コードのほとんどを示しています。これ以外に出される可能性のある戻りコードと理由コードは DBRC 関連のもので、この照会要求に関連したものではありません。

表 95. TYPE=PART 照会要求の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
警告	X'00000008'	X'D8210002'	指定された DBDS またはエリアが RECON に登録されていません。情報ブロックは戻されません。
	X'00000008'	X'D8220001'	RECON に HALDB の 区画が登録されていません。情報ブロックは戻されません。
	X'00000008'	X'D8220002'	指定された区画が RECON に登録されていません。情報ブロックは戻されません。
	X'00000008'	X'D8220003'	指定された区画より前の高位キー区画が、RECON に存在しません。情報ブロックは戻されません。
	X'00000008'	X'D8220004'	指定された区画より後の高位キー区画が、RECON に存在しません。情報ブロックは戻されません。
	X'00000008'	X'D8220005'	指定された DBNAME が RECON に登録されていません。情報ブロックは戻されません。
	X'00000008'	X'D8220006'	HALDB の最初の区画または最後の区画の検索時に、アクティブ区画が検出されませんでした。
	X'00000008'	X'D8220007'	前の区画または次の区画の検索時に、アクティブ区画が検出されませんでした。
重大エラー	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。

表 95. TYPE=PART 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'0000000C'	X'D8220100'	セキュリティー・エラー。SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーは要求の実行を許可されていないと判断しました。
ストレージ・エラー	X'00000028'	X'D8210001'	DBDS ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210002'	AREA ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210003'	RCVINFO ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210004'	ALLOC ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210005'	IC ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210006'	REORG ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8210007'	RECOV ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D8220001'	PART ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D9100001'	要求の処理中にエラーが発生しました。DBRC はこの時点までに取得したストレージを解放します。しかし、ストレージを解放しようとして別のエラーが検出されました。
内部エラー	X'0000002C'	X'D8000001'	RECON を開くときに障害が発生しました。
	X'0000002C'	X'D8210001'	最初の DBDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210002'	指定された DBDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210003'	次の DBDS レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210004'	最初のエリアの AUTH レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210005'	最初の ALLOC レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210006'	次の ALLOC レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210007'	最初の IC レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210008'	次の IC レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8210009'	最初の REORG レコードの検索時に障害が発生しました。

表 95. TYPE=PART 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'0000002C'	X'D821000A'	次の REORG レコードの検索時に障害が発生しました。
	X'0000002C'	X'D821000B'	最初の RECOV レコードの検索時に障害が発生しました。
	X'0000002C'	X'D821000C'	次の RECOV レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8220001'	最初の HALDB 区画レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8220002'	処理中の HALDB 区画に関連付けられた DB レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8220003'	次の HALDB 区画レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8220004'	指定された HALDB (DBNAME) の検索時に障害が発生しました。
	X'0000002C'	X'D8220005'	最初のアクティブ区画または最後のアクティブ区画の検索時に障害が発生しました。
	X'0000002C'	X'D8220006'	最初の高位キー区画または最後の高位キー区画の検索時に障害が発生しました。
	X'0000002C'	X'D8220007'	前の高位キー区画または次の高位キー区画の検索時に障害が発生しました。
	X'0000002C'	X'D8220008'	既存の Part DB レコードに対応する PART レコードの検索時に障害が発生しました。
	X'0000002C'	X'D8220009'	部分選択ルーチンを使用する HALDB の前の区画または次の区画の検索時に障害が発生しました。
	X'0000002C'	X'D822000A'	指定された区画に対する HALDB レコードの検索時に障害が発生しました。
	X'0000002C'	X'D822000B'	指定された PARTNAME に対する Part DB レコードの検索時に障害が発生しました。
	X'0000002C'	X'D822000C'	PART レコードも Part DB レコードも ProcessPART に渡されていません。
	X'0000002C'	X'D822000D'	処理中の HALDB 区画に関連付けられた PART レコードの検索時に障害が発生しました。
	X'0000002C'	X'D82221xx'	内部照会 DBDS 呼び出しが RC=X'30' RSN=X'D82100xx' のパラメーター・エラーを返しました。
パラメーター・エラー	X'00000030'	X'C9000001'	API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN フィールド・アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

表 95. TYPE=PART 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'C9000003'	RETCODE フィールド・アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE フィールド・アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000005'	OUTPUT フィールド・アドレスが無効です。OUTPUT アドレスを含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000008'	DBNAME または PARTNAME アドレスが無効です。DBNAME または PARTNAME を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000009'	DDN アドレスが無効です。DDN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'D8000001'	OUTPUT パラメーターが欠落しているか無効です。
	X'00000030'	X'D8000002'	TYPE パラメーターに指定された値が無効です。
	X'00000030'	X'D8000003'	TYPE=PART では、少なくとも VERSION=2.0 が必要です。
	X'00000030'	X'D8220002'	LOC=FIRST LAST ALL は、PARTNAME を使用する場合は指定できません。
	X'00000030'	X'D8220003'	LOC=PREV NEXT は、DBNAME を使用する場合は指定できません。
	X'00000030'	X'D8220004'	DBNAME または PARTNAME には、明示的な名前を指定する必要があります。アスタリスクは使用できません。
	X'00000030'	X'D8220005'	指定された DBNAME が HALDB ではありません。
	X'00000030'	X'D8220006'	LOC=PREV NEXT を使用する場合、区画名にはアクティブ区画を指定する必要があります。

表 95. TYPE=PART 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'D8220007'	PARTNAME には、区画の名前を指定する必要があります。

## TYPE=PART 照会の出力

以下のマッピングは、TYPE=PART 照会で使用されます。

- DSPAPQHP - HALDB 区画ブロック
- DSPAPQSL - サブシステム・リスト

関連概念:

403 ページの『第 26 章 DBRC API』

関連資料:

431 ページの『第 30 章 DBRC 照会要求』

437 ページの『データベース照会要求 (TYPE=DB)』

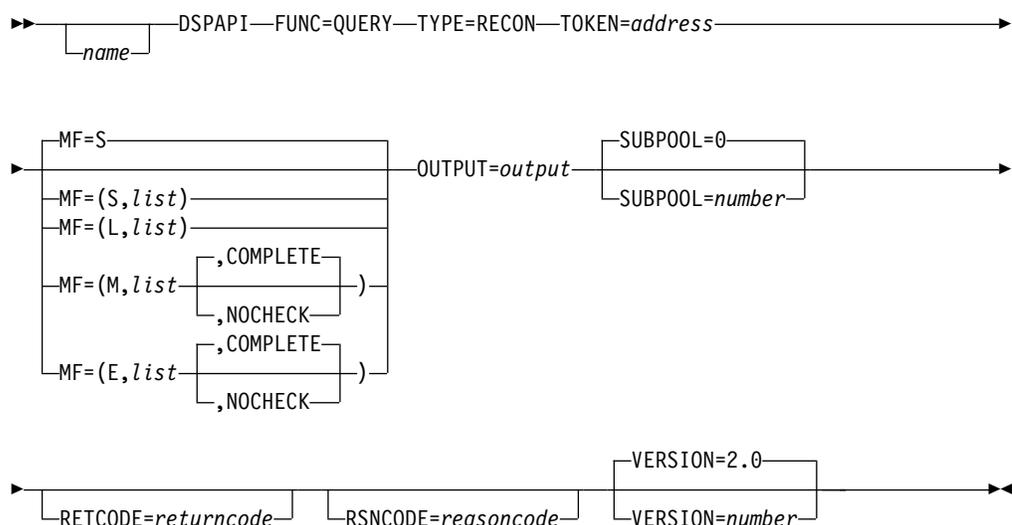
461 ページの『DBDS 照会要求 (TYPE=DBDS)』

➡ DBRC 要求戻りコード (メッセージおよびコード)

## RECON 状況照会要求 (TYPE=RECON)

RECON ヘッダー情報および各 RECON データ・セットの状況を取得するには、RECON 状況照会 (DSPAPI FUNC=QUERY TYPE=RECON) 要求を使用します。

### TYPE=RECON 照会要求の構文



### TYPE=RECON 照会要求のパラメーター

名前 ユーザーが指定できるオプションのシンボルです。使用する場合は、1 桁目から始めます。

**TYPE=RECON**

RECON 状況情報が要求されることを指定します。

**TOKEN=***symbol* | (2 - 12)

FUNC=STARTDBRC 要求で戻された 4 バイト・フィールドのアドレスを指定する必須パラメーター。

**MF=**S | L | M | E

要求のマクロ形式を指定するオプション・パラメーターです。

**OUTPUT=***output* | (2 - 12)

RECON 状況情報ブロックを示すポインターを受け取るフィールドを指定します。

出力が作成されなかった場合、出力アドレスはゼロです。この状態は、RECON 内に要求を満たす情報がない場合、または出力を作成できるようになる前にエラーが発生した場合に起こります。

出力ブロックのストレージは、呼び出し元によって事前割り振りされません。DBRC はこれらのブロック用に指定されたサブプールからストレージを取得します。呼び出し元は、バッファ解放サービス (DSPAPI FUNC=RELBUF) を使用してこのストレージを解放し、戻された出力アドレスを指定する必要があります。

**SUBPOOL= 0** | *number*

取得されるストレージのサブプール番号を指定するオプション・パラメーターです。このパラメーターを指定しない場合、FUNC=STARTDBRC 要求によって指定されたサブプールがデフォルトになります。それ以外の場合は、サブプール 0 がデフォルトになります。

**RETCODE=***returncode* | (2-12)

シンボルとして指定する場合、戻りコードを受け取るために、ストレージのワードを示すラベルを指定します。レジスターとして指定する場合、戻りコードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RETCODE が指定されたかどうかにかかわらず、レジスター 15 は戻りコードを含んでいます。

**RSNCODE=***reasoncode* | (2-12)

シンボルとして指定する場合、理由コードを受け取るために、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、理由コードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RSNCODE が指定されたかどうかにかかわらず、レジスター 0 は理由コードを含んでいます。

**VERSION=2.0** | *number*

この要求によって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターはマクロ処理によって受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョン

の従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

有効なバージョン番号は 1.0 および 2.0 です。

## TYPE=RECON 照会要求の戻りコードと理由コード

表 96. TYPE=RECON 照会の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
重大エラー	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。
	X'0000000C'	X'D8100100'	セキュリティ・エラー。SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーは要求の実行を許可されていないと判断しました。
ストレージ・エラー	X'00000028'	X'D8100001'	RECON ブロック用のストレージの取得時にエラーが発生しました。
内部エラー	X'0000002C'	X'D8000001'	RECON を開くときに障害が発生しました。
	X'0000002C'	X'D8100001'	RECON ヘッダー・レコードを検索しようとして障害が発生しました。
パラメーター・エラー	X'00000030'	X'C9000001'	API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN フィールド・アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000003'	RETCODE フィールド・アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE フィールド・アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000005'	OUTPUT フィールド・アドレスが無効です。OUTPUT アドレスを含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

表 96. TYPE=RECON 照会の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'D8000001'	OUTPUT パラメーターが欠落しているか無効です。
	X'00000030'	X'D8000002'	TYPE パラメーターに指定された値が無効です。

## TYPE=RECON 照会要求の出力

以下の図に、QUERY TYPE=RECON 要求からの出力フォーマットを示します。以下のサンプル DSECT が、ストレージ・ブロックのフィールドとそれらのフィールドの相互関係を詳細に説明します。

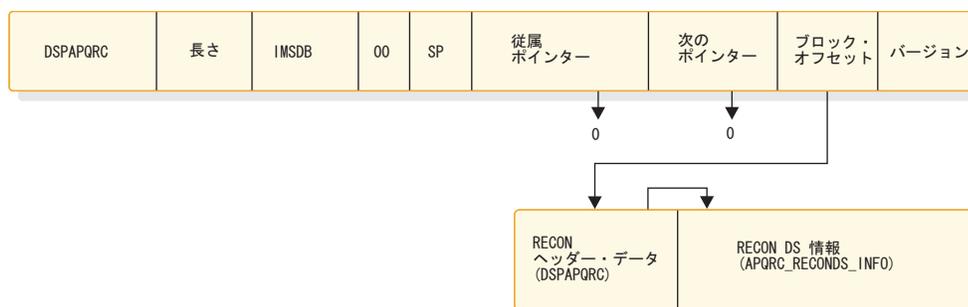


図 27. QUERY TYPE=RECON の出力のフォーマット

## DSPAPQRC の DSECT

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	560	DSPAPQRC	
0	(0)	CHARACTER	44	APQRC_DATA	Initialized with "RECOVERY CONTROL DATASET"
44	(2C)	UNSIGNED	4	APQRC_RECONINFO	Offset to RECON data set info
48	(30)	SIGNED	4	*(2)	Reserved
56	(38)	UNSIGNED	2	APQRC_RECONINFOLEN	Length of each RECON dataset info element
58	(3A)	UNSIGNED	1	APQRC_RECONCOUNT	# of RECON dataset elements
59	(3B)	CHARACTER	1	*	Reserved
60	(3C)	BIT(8)	1	APQRC_FLAGS	Process flags...
		1... ..		APQRC_NOCHK	1= NOCHECK log tape dsn check
		.1.. ..		APQRC_CHK17	1= CHECK17 log tape dsn check
		..1. ....		APQRC_CHK44	1= CHECK44 log tape dsn check
		...1 ....		APQRC_LSTLG	1= list log DSN
		.... 1...		APQRC_INUPG	Upgrade in progress.
		.... .1..		APQRC_REORGV	Reorg verification
61	(3D)	BIT(8)	1	APQRC_FLAG2	More flags...
		1... ..		APQRC_FORCE	1 = FORCER, 0 = NOFORCER
		.1.. ..		APQRC_CATDS	1=CA IC LOGS cataloged
		..1. ....		APQRC_TRACE	1 = ext. GTF trace on
		...1 ....		APQRC_CMDSAF	SAF enabled
		.... 1...		APQRC_CMDEXIT	Cmd auth exit enabled
		.... .1..		APQRC_PRA	Parallel RECON Access in use (VERSION=2.0)
		.... ..1.		APQRC_LISTFUZZY	PRA Concurrent LIST active (VERSION=2.0)
62	(3E)	CHARACTER	2	*	Reserved
64	(40)	CHARACTER	136	APQRC_CLEAN	Fields needed for cleanup
64	(40)	SIGNED	4	APQRC_CSET	0 = updates not in progress,

68	(44)	SIGNED	4	APQRC_TYPE	>0 = update in progress Type of update
72	(48)	CHARACTER	32	APQRC_OKEY	Key of original record
104	(68)	CHARACTER	32	APQRC_BKEY	Key of base record that is in the process of being changed
136	(88)	CHARACTER	32	APQRC_NKEY	Key of new record
168	(A8)	CHARACTER	16	APQRC_DBID	DBID of DBDS
168	(A8)	CHARACTER	8	APQRC_DBD	DBD name
176	(B0)	CHARACTER	8	APQRC_DDN	DD name
184	(B8)	CHARACTER	8	APQRC_CAGRP	CA group name
192	(C0)	CHARACTER	8	APQRC_DDNEW	New DBDS DD name
200	(C8)	UNSIGNED	2	APQRC_DMBNO	DMB sequence number
202	(CA)	UNSIGNED	2	APQRC_LASTREUSEDDB#	Last reused DMB number, valid only when apqrc_DMBNO is 32767
204	(CC)	CHARACTER	7	APQRC_INITTOKEN	Recon init. トークン
211	(D3)	CHARACTER	8	APQRC_CMDHLQ	Cmd auth high lvl qual
219	(DB)	UNSIGNED	1	APQRC_MVERS	Minimum IMS version
220	(DC)	BIT(8)	1	APQRC_NWFLG	Fields needed for RECON I/O error(s)
		1... ..		APQRC_NEW	1=STARTNEW, 0=NONEW
221	(DD)	CHARACTER	3	*	Reserved
224	(E0)	CHARACTER	8	APQRC_SSIDN	SSID for DASD
232	(E8)	CHARACTER	8	APQRC_DASDU	Unit type for DASD
240	(F0)	CHARACTER	8	APQRC_TAPEU	Unit type for tape
248	(F8)	CHARACTER	24	APQRC_TIME	
248	(F8)	CHARACTER	2	APQRC_TZDEF	Input offset default
250	(FA)	CHARACTER	5	APQRC_TMFMT	Time format options
255	(FF)	CHARACTER	1	*	Reserved
256	(100)	SIGNED	2	APQRC_TPREC	Time-stamp precision
258	(102)	CHARACTER	12	APQRC_LOGRT	Minimum log retention period
270	(10E)	SIGNED	2	APQRC_TZNUM	Number of entries in time zone label table
272	(110)	CHARACTER	8	APQRC_TZTBL(32)	Time zone label table
528	(210)	BIT(16)	2	APQRC_TROPT	DBRC trace options
530	(212)	CHARACTER	5	APQRC_IMSPLEX	IMSpIex name
535	(217)	CHARACTER	5	*	Reserved
540	(21C)	UNSIGNED	4	APQRC_SIZW_DSNUM	SIZEALERT dsnum
544	(220)	UNSIGNED	4	APQRC_SIZW_VOLNUM	SIZEALERT volnum
548	(224)	UNSIGNED	4	APQRC_SIZW_PERCENT	SIZEALERT percent
552	(228)	UNSIGNED	4	APQRC_LOGW_DSNUM	LOGALERT dsnum
556	(22C)	UNSIGNED	4	APQRC_LOGW_VOLNUM	LOGALERT volnum
560	(230)	UNSIGNED	44	APQRC_CMDRNQ	CMDAUTH RECON qual
604	(25C)	UNSIGNED	8	APQRC_DBCOUNT	REGISTERED DBCOUNT

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	53	APQRC_RECONDS_INFO	
0	(0)	CHARACTER	8	APQRC_RECONDS_DDNAME	RECON DD name
8	(8)	CHARACTER	44	APQRC_RECONDS_DSNAME	RECON DS name
52	(34)	BIT(8)	1	APQRC_RECONDS_STATUS	RECON DS status
		1... ..		APQRC_RECONDS_COPY1	COPY 1
		.1.. ..		APQRC_RECONDS_COPY2	COPY 2
		..1. ....		APQRC_RECONDS_SPARE	Spare
		...1 ....		APQRC_RECONDS_DISCARDED	Discarded
		.... 1...		APQRC_RECONDS_UNAVAIL	Unavailable

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQRC	APQRC_EYECATCHER	

関連概念:

407 ページの『DSPAPI マクロのマクロ形式』

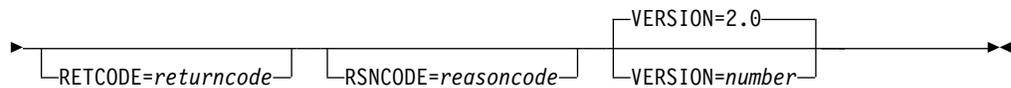
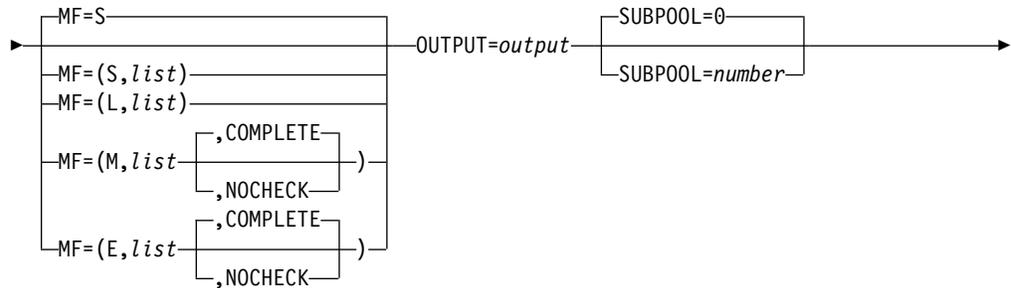
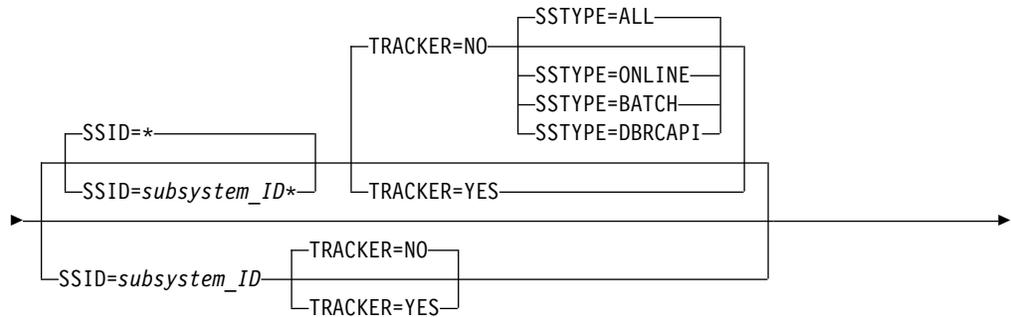
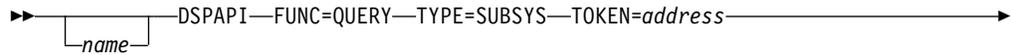
関連資料:

431 ページの『第 30 章 DBRC 照会要求』

## サブシステム照会要求 (TYPE=SUBSYS)

特定のサブシステムまたはすべてのサブシステムのサブシステム情報を RECON データ・セットから取得するには、サブシステム照会 (DSPAPI FUNC=QUERY TYPE=SUBSYS) 要求を使用します。

### TYPE=SUBSYS 照会要求の構文



## TYPE=SUBSYS 照会要求のパラメーター

名前 ユーザーが指定できるオプションのシンボルです。使用する場合は、1 桁目から始めます。

### TYPE=SUBSYS

サブシステム情報が要求されることを指定します。

### TOKEN=*address* | (2 - 12)

FUNC=STARTDBRC 要求で戻された 4 バイト・フィールドのアドレスを指定します。

### SSID= \* | *name*\* | *name* | (2 - 12)

照会されるサブシステムの名前を指定します。ワイルドカード・キーワード \* (アスタリスク) のみを使用して、すべてのサブシステムに関する情報を要求することができます (SSID=\*, これはデフォルトです)。また、これを名前の最後に使用して、そのパターンと一致する名前を持つサブシステムを照会することもできます。この場合、アスタリスクの前には、英字が少なくとも 1 つ必要です。

### SSYTYPE=ALL | ONLINE | BATCH | DBRCAPI

情報を要求するサブシステムのタイプを指定します。このパラメーターは、それと一緒に SSID に対して特定のサブシステム名を指定している場合、または TRACKER=YES を指定する場合には指定できません。

SSTYPE=DBRCAPI を指定する場合、最小バージョン番号 VERSION=2.0 を指定する必要があります。SSTYPE=ALL がデフォルトです。

### TRACKER=YES | NO

RSR トラッキング・サブシステムに関するログ情報が要求されているかどうかを指定します。TRACKER=YES を指定すると、RSR トラッカーのログ情報が要求されます。TRACKER=NO を指定すると、非 RSR サブシステム、または RSR システムのアクティブ・サブシステムからのログ情報が要求されます。TRACKER=NO がデフォルトです。

### MF=S | L | M | E

要求のマクロ形式を指定するオプション・パラメーターです。

### OUTPUT=*output* | (2 - 12)

サブシステム情報ブロック (おそらくチェーンになっている) の最初のブロックを示すポインターを受け取るフィールドを指定します。

出力アドレスは、出力が作成されなかった場合は 0 になります。これは、RECON 内に要求を満たす情報がない場合、または出力が作成される前にエラーが発生した場合に、起こる可能性があります。

出力ブロックのストレージは、呼び出し元によって事前割り振りされません。DBRC はこれらのブロック用に指定されたサブプールからストレージを取得します。呼び出し元は、バッファ解放サービス (DSPAPI FUNC=RELBUF) を使用してこのストレージを解放し、戻された出力アドレスを指定する必要があります。

### SUBPOOL= 0 | *number*

取得されるストレージのサブプール番号を指定します。このパラメーターを

指定しない場合、FUNC=STARTDBRC 要求によって指定されたサブプールがデフォルトになります。それ以外の場合は、サブプール 0 がデフォルトになります。

**RETCODE=returncode | (2 - 12)**

戻りコードを受け取る、ストレージ内の場所を指定します。シンボルとして指定する場合、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。このパラメーターを指定しない場合、戻りコードはレジスター 15 に入れられます。

**RSNCODE=reasoncode | (2 - 12)**

理由コードを受け取る、ストレージ内の場所を指定します。シンボルとして指定する場合、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。このパラメーターを指定しない場合、理由コードはレジスター 0 に入れられます。

**VERSION=2.0 | number**

この要求によって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターはマクロ処理によって受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

FUNC=QUERY TYPE=SUBSYS 要求の有効なバージョン番号は、1.0 および 2.0 です。

**TYPE=SUBSYS 照会要求の戻りコードと理由コード**

表 97. TYPE=SUBSYS 照会要求の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
警告	X'00000008'	X'D8600001'	要求されたタイプ (アクティブまたはトラッカー) のサブシステム・レコードが存在しません。
	X'00000008'	X'D8600002'	バッチまたはオンライン・サブシステム・レコードが存在しません。
重大エラー	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。

表 97. TYPE=SUBSYS 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'0000000C'	X'D8600100'	セキュリティー・エラー。SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーは要求の実行を許可されていないと判断しました。
ストレージ・エラー	X'00000028'	X'D8600001'	SUBSYS ブロック用のストレージの取得時にエラーが発生しました。
	X'00000028'	X'D9100001'	要求の処理中にエラーが発生しました。DBRC はこの時点までに取得したストレージを解放します。しかし、ストレージを解放しようとして別のエラーが検出されました。
内部エラー	X'0000002C'	X'D8000001'	RECON を開くときに障害が発生しました。
	X'0000002C'	X'D8600001'	要求されたタイプ (アクティブまたはトラッカー) の最初のサブシステム・レコードまたは指定されたサブシステム・レコードを検索しようとして障害が発生しました。
	X'0000002C'	X'D8600002'	要求されたタイプ (アクティブまたはトラッカー) の次のサブシステム・レコードを検索しようとして障害が発生しました。
パラメーター・エラー	X'00000030'	X'C9000001'	API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN フィールド・アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000003'	RETCODE フィールド・アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE フィールド・アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000005'	OUTPUT フィールド・アドレスが無効です。OUTPUT アドレスを含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000010'	SSID フィールド・アドレスが無効です。SSID を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

表 97. TYPE=SUBSYS 照会要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'D8000001'	OUTPUT パラメーターが欠落しているか無効です。
	X'00000030'	X'D8000002'	TYPE パラメーターに指定された値が無効です。
	X'00000030'	X'D8600001'	SSTYPE=ONLINE BATCH DBRCAPI は、特定の SSID に対する要求、またはトラッキング・サブシステム要求 (TRACKER=YES) の場合は指定できません。
	X'00000030'	X'D8600100'	ワイルドカードを使用するとき、アスタリスクの前に英字が少なくとも 1 つ必要です。
	X'00000030'	X'D8600101'	ワイルドカードを使用するとき、最後の文字はアスタリスクである必要があります。

## TYPE=SUBSYS 照会要求の出力

以下の図に、QUERY TYPE=SUBSYS 要求からの出力フォーマットを示します。以下のサンプル DSECT が、ストレージ・ブロックのフィールドとそれらのフィールドの相互関係を詳細に説明します。

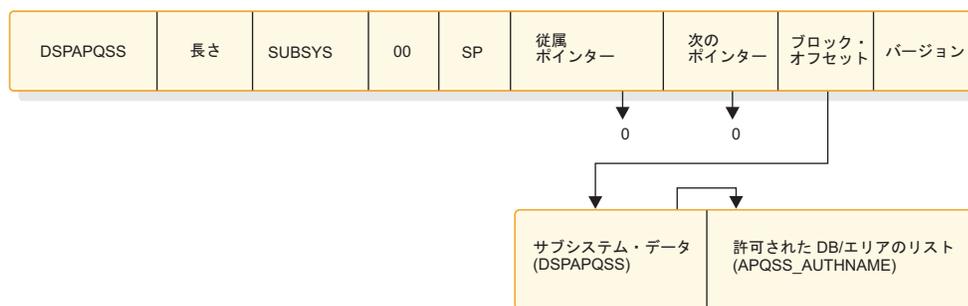


図 28. QUERY TYPE=SUBSYS の出力のフォーマット

## DSPAPQSS の DSECT

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	64	DSPAPQSS	
0	(0)	CHARACTER	8	APQSS_SSID	Subsystem identifier
8	(8)	UNSIGNED	4	APQSS_AUTHLIST	Offset to authd DB/Area list
12	(C)	SIGNED	4	APQSS_AUTHCOUNT	Number of authorized DB/Areas
16	(10)	UNSIGNED	2	APQSS_AUTHLEN	Length of auth entry
18	(12)	CHARACTER	6	*	Reserved
24	(18)	CHARACTER	12	APQSS_LOGTIME	Start time of log
36	(24)	UNSIGNED	1	APQSS_RELLVL	Subsystem release level '71'X=V7R1,'81'X=V8R1 '91'X=V9R1, etc.
37	(25)	CHARACTER	1	APQSS_COEXLVL	Coexistence level
38	(26)	UNSIGNED	1	APQSS_IRLMCT	IRLM status count
39	(27)	CHARACTER	1	*	Reserved
40	(28)	CHARACTER	8	APQSS_GSGNAME	Global Service Group name
48	(30)	CHARACTER	5	APQSS_IRLMID	IRLM ID of SS

53	(35)	CHARACTER	5	APQSS_IRLMBK	IRLM ID of backup SS
58	(3A)	BIT(8)	1	APQSS_FLAGS	Flags
		1... ..		APQSS_TYPE	1=Online   0=batch
		.1.. ..		APQSS_ABTERM	Abnormal termination
		..1. ....		APQSS_RCVPRC	Recovery processing started
		...1 ....		APQSS_BKSIGN	Backup SS signed on
		.... 1...		APQSS_IRLMFL	IRLM failure
		.... .1..		APQSS_COMMFL	COMM failure
		.... ..1.		APQSS_SYSFL	SYS failure
		.... ...1		APQSS_ACTVST	Status of active SS when backup exists, 1=abterm
59	(3B)	BIT(8)	1	APQSS_FLAGS2	FLAGS 2
		1... ..		APQSS_SHRING	Sharing covered DBs
		.1.. ..		APQSS_TRKER	Subsystem is a Tracker
		..1. ....		APQSS_TRKTRM	TRACKER has terminated
		...1 ....		APQSS_TRCKED	SSID is tracked
		.... 1...		APQSS_FRSTSO	1st signon after RSR takeover is in progress
		.... .1..		APQSS_XRFCAP	SS is XRF capable
		.... ..1.		APQSS_DBRCAPI	SS is a DBRC application (VERSION=2.0)
		.... ...1		APQSS_BPE	BPE-based subsystem (VERSION=4.0)
		.... ..11		*	Reserved
60	(3C)	SIGNED	2	APQSS_BCKTKN	Backup recovery token
62	(3E)	CHARACTER	2	*	Reserved

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	32	APQSS_AUTHNAME	Names of authd DB/Areas
0	(0)	CHARACTER	8	APQSS_DBNAME	DB name
8	(8)	CHARACTER	8	APQSS_AREANM	If FP, Area name
16	(10)	UNSIGNED	1	APQSS_SHRLVL	Share level
17	(11)	UNSIGNED	1	APQSS_DBACCS	Access intent
18	(12)	UNSIGNED	1	APQSS_DBNCOD	Encoded state
19	(13)	UNSIGNED	1	APQSS_DBSTAT	DB status flags
20	(14)	UNSIGNED	2	APQSS_DBEQCT	DB EQE count
22	(16)	SIGNED	2	APQSS_GLBDMB	Global DMB number
24	(18)	BIT(8)	1	APQSS_AUTHFLAGS	Flags
		1... ..		APQSS_NRDBUP	Nonrecov DB/Area updated
		.1.. ..		APQSS_COVRD	DB covered by GSG
		..1. ....		APQSS_NRECV	nonrecoverable DB/Area
		...1 ....		APQSS_ORDBDS	0=A-J/1=M-V ACTIVE
		.... 1...		APQSS_OLRON	0 = no OLR active
		.... .1..		APQSS_OLROWR	0 = no OLR owner
		.... ..1.		APQSS_OLROWD	0 OLR not owned by SS
25	(19)	CHARACTER	7	*	Reserved

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPQSS	APQSS_EYECATCHER	

関連概念:

407 ページの『DSPAPI マクロのマクロ形式』

関連資料:

431 ページの『第 30 章 DBRC 照会要求』





**RSNCODE=reasoncode | (2 - 12)**

理由コードを受け取る、ストレージ内の場所を指定します。シンボルとして指定する場合、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。このパラメーターを指定しない場合、理由コードはレジスター 0 に入れます。

**MF=S | L | M | E**

要求のマクロ形式を指定します。

**VERSION=2.0 | number**

この要求によって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターの処理は受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

**RELBUF の戻りコードと理由コード**

表 98. RELBUF の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
部分的な正常完了	X'00000004'	X'D9000001'	解放するストレージがありませんでした。BUFFER パラメーターにアドレス 0 (ゼロ) が渡されました。
重大エラー	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC 要求の発行元の TCB アドレスとは異なります。
ストレージ・エラー	X'00000028'	X'D9100001'	ストレージを解放しようとしてエラーが発生しました。
パラメーター・エラー	X'00000030'	X'C9000001'	パラメーター・エラー。API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN フィールド・アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

表 98. RELBUF の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'C9000003'	RETCODE フィールド・アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE フィールド・アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000007'	BUFFER アドレスが無効です。解放する BUFFER のアドレスが妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'D9100001'	解放するストレージ・ブロックのヘッダーが無効です。
	X'00000030'	X'D9100002'	解放するストレージ・ブロックの長さが 0 より大きくありません。

関連概念:

403 ページの『第 26 章 DBRC API』

407 ページの『DSPAPI マクロのマクロ形式』





**READONLY=NO | YES**

アプリケーションで DBRC 情報への読み取り専用アクセスが必要か (YES) 不要か (NO) を指定します。READONLY=NO を指定する場合、アプリケーションには RECON データ・セットに対する最小限の更新レベル権限が必要です。

このパラメーターを使用するには、API VERSION=2.0 以降を指定する必要があります。

**SSID=NULL | name**

DBRC への登録に使用するサブシステム名を含む 8 バイト・フィールドのアドレスを指定します。SSID=NULL を指定する場合、DBRC への登録は実行されません。READONLY=YES と SSID=symbol の両方を指定しないでください。

デフォルトは SSID=NULL です。このパラメーターを使用するには、API VERSION=2.0 以降を指定する必要があります。

**GSGNAME=name | (2 - 12)**

グローバル・サービス・グループ (GSG) の名前を含む 8 バイト・フィールドのアドレスを指定します。GSG 名は、DBRC に認識されている必要があります。デフォルトは NULL で、GSG 名は指定しません。

このパラメーターを使用する場合、API VERSION=2.0 以降も指定する必要があります。

**RECON1=ddname | (2 - 12)**

デフォルト DD 名 RECON1 の代わりに使用する DD 名を含む 8 バイト・フィールドのアドレスを指定します。

省略すると、デフォルト名 RECON1 が使用されます。このパラメーターを使用するには、API VERSION=2.0 以降を指定する必要があります。

**RECON2=ddname | (2 - 12)**

デフォルト DD 名 RECON2 の代わりに使用する DD 名を含む 8 バイト・フィールドのアドレスを指定します。

省略すると、デフォルト名 RECON2 が使用されます。このパラメーターを使用するには、API VERSION=2.0 以降を指定する必要があります。

**RECON3=ddname | (2 - 12)**

デフォルト DD 名 RECON3 の代わりに使用する DD 名を含む 8 バイト・フィールドのアドレスを指定します。

省略すると、デフォルト名 RECON3 が使用されます。このパラメーターを使用するには、API VERSION=2.0 以降を指定する必要があります。

**IMS=IMS | ddname**

デフォルト DD 名 IMS の代わりに使用する DD 名を含む 8 バイト・フィールドのアドレスを指定します。

デフォルトは IMS=IMS です。このパラメーターを使用する場合、API VERSION=2.0 以降も指定する必要があります。

**SUBPOOL= 0 | number**

ストレージを返すすべての要求 (QUERY や AUTH など) に使用するデフ

オルト・サブプール番号を指定します。使用するプログラムに有効なサブプールについては、「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。

ここでまたは要求で SUBPOOL を指定しない場合、その要求のデフォルトは SUBPOOL=0 です。

**MF=S | L | M | E**

要求のマクロ形式を指定します。

**RETCODE=returncode | (2 - 12)**

戻りコードを受け取る、ストレージ内の場所を指定します。シンボルとして指定する場合、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。このパラメーターを指定しない場合、戻りコードはレジスター 15 に入れます。

**RSNCODE=reasoncode | (2 - 12)**

理由コードを受け取る、ストレージ内の場所を指定します。シンボルとして指定する場合、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。このパラメーターを指定しない場合、理由コードはレジスター 0 に入れます。

**VERSION=2.0 | number**

この要求によって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターの処理は受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

**STARTDBRC の戻りコードと理由コード**

表 99. STARTDBRC 要求の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
警告	X'00000004'	X'E2000001'	RECON データ・セットは初期設定されません。使用できる機能は INIT.RECON コマンドのみです。
	X'00000004'	X'E2000002'	RECON データ・セットはアップグレードされません。使用できる機能は CHANGE.RECON UPGRADE コマンドのみです。
重大エラー	X'0000000C'	X'C9000002'	DBRC は、DBRC 初期設定のフェーズ 0 で、初期設定を正常に行えませんでした。
	X'0000000C'	X'C9000003'	DBRC は、DBRC 初期設定のフェーズ 1 で、初期設定を正常に行えませんでした。

表 99. STARTDBRC 要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'0000000C'	X'C9000004'	z/OS LOAD が、DBRC モジュール DSPCRTR0 で失敗しました。
	X'0000000C'	X'C9000005'	STORAGE 要求が失敗しました。API は、要求を完了するために必要なストレージを取得できません。
	X'0000000C'	X'C9000012'	STARTDBRC は前に発行されていますが、介入する STOPDBRC は出されていません。
	X'0000000C'	X'C9D40001'	DSPAPI マクロの実行時に、DBRC モジュール DSPAPI00 のロードで障害が発生しました。
	X'0000000C'	X'E2000001'	RECON データ・セットは初期設定されません。SSID が指定されており、SIGNON を実行できないため、STARTDBRC は失敗します。
	X'0000000C'	X'E2000002'	RECON データ・セットはアップグレードされません。SSID が指定されており、SIGNON を実行できないため、STARTDBRC は失敗します。
	X'0000000C'	X'E2000003'	STARTDBRC で READONLY が指定されました。ただし、SSID が指定されたため、SIGNON を実行できません。
	X'0000000C'	X'E200000A'	RECON が初期化されていないか、アップグレードされていません。V1 は呼び出し元です。
	X'0000000C'	X'E2000100'	セキュリティー・エラー。SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーは要求の実行を許可されていないと判断しました。
	X'0000000C'	X'E210nnnn'	アプリケーションにサインオンしようとして発生したエラー。「nnnn」は、DBRC サインオン・モジュール DSPSSIGN からの戻りコードです。
内部エラー	X'00000028'	X'C9000001'	DBRC の内部エラー。サブシステム・レコードの書き込み時に障害が発生しました。
パラメーター・エラー	X'00000030'	X'C9000001'	パラメーター・エラー。API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN フィールド・アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

表 99. STARTDBRC 要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'C9000003'	RETCODE フィールド・アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE フィールド・アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000006'	SYSPRINT フィールド・アドレスが無効です。SYSPRINT DD 名を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C900000A'	IMS フィールド・アドレスが無効です。IMS DD 名を含むフィールドのアドレスが、妥当性検査に失敗しました。
	X'00000030'	X'C900000B'	GDGNAME フィールド・アドレスが無効です。GSGNAME 名を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C900000C'	RECON1 フィールド・アドレスが無効です。RECON1 DD 名を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C900000D'	RECON2 フィールド・アドレスが無効です。RECON2 DD 名を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C900000E'	RECON3 フィールド・アドレスが無効です。RECON3 DD 名を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'E2100001'	SSID にゼロまたはブランクが指定されました。

## 関連概念:

403 ページの『第 26 章 DBRC API』

407 ページの『DSPAPI マクロのマクロ形式』

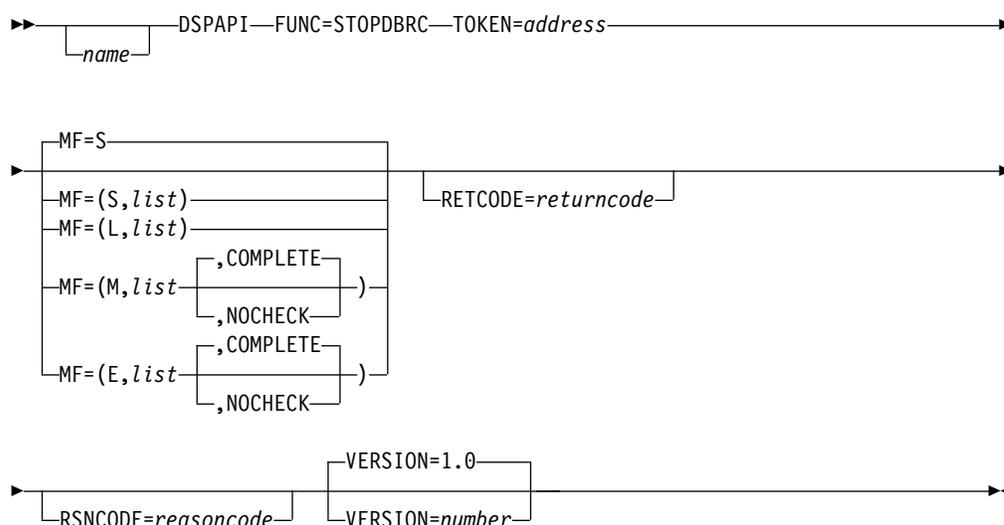


## 第 33 章 DBRC 停止要求 (STOPDBRC)

DBRC アプリケーションを終了し、DBRC を停止するには、STOPDBRC 要求を使用します。

STARTDBRC 要求で SSID を指定して、アプリケーションを DBRC に登録した場合、そのアプリケーションのサブシステム・レコードが RECON に記録されます。STOPDBRC 要求は、自動的にアプリケーションを登録解除し、許可されたデータベースを無許可にして、サブシステム・レコードを削除します。

### STOPDBRC 要求の構文



### STOPDBRC のパラメーター

**名前** ユーザーが指定できるオプションのシンボルです。使用する場合は、1 桁目から始めます。

#### TOKEN=*symbol* | (2 - 12)

FUNC=STARTDBRC 要求で戻された 4 バイト・フィールドのアドレスを指定します。

#### RETCODE=*returncode* | (2-12)

シンボルとして指定する場合、戻りコードを受け取るために、ストレージのワードを示すラベルを指定します。レジスターとして指定する場合、戻りコードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RETCODE が指定されたかどうかにかかわらず、レジスター 15 は戻りコードを含んでいます。

#### RSNCODE=*reasoncode* | (2-12)

シンボルとして指定する場合、理由コードを受け取るために、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、理由コードを受け取るために、そのレジスターはストレージ

ジのワードを示すアドレスを含んでいなければなりません。RSNCODE が指定されたかどうかにかかわらず、レジスター 0 は理由コードを含んでいます。

**MF=S | L | M | E**

要求のマクロ形式を指定します。

**VERSION=1.0 | number**

この要求によって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターの処理は受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

## STOPDBRC の戻りコードと理由コード

表 100. STOPDBRC 要求の戻りコードと理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
重大エラー	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC 要求の発行元の TCB アドレスとは異なります。
	X'0000000C'	X'E220nnnn'	アプリケーションからサインオフしようとして発生したエラー。「nnnn」は、DBRC サインオフ・モジュール DSPSSIGN からの戻りコードです。
	X'0000000C'	X'E2000100'	セキュリティ・エラー。SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーは要求の実行を許可されていないと判断しました。
パラメーター・エラー	X'00000030'	X'C9000001'	パラメーター・エラー。API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN フィールド・アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000003'	RETCODE フィールド・アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

表 100. STOPDBRC 要求の戻りコードと理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'C9000004'	RSNCODE フィールド・アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

関連概念:

403 ページの『第 26 章 DBRC API』

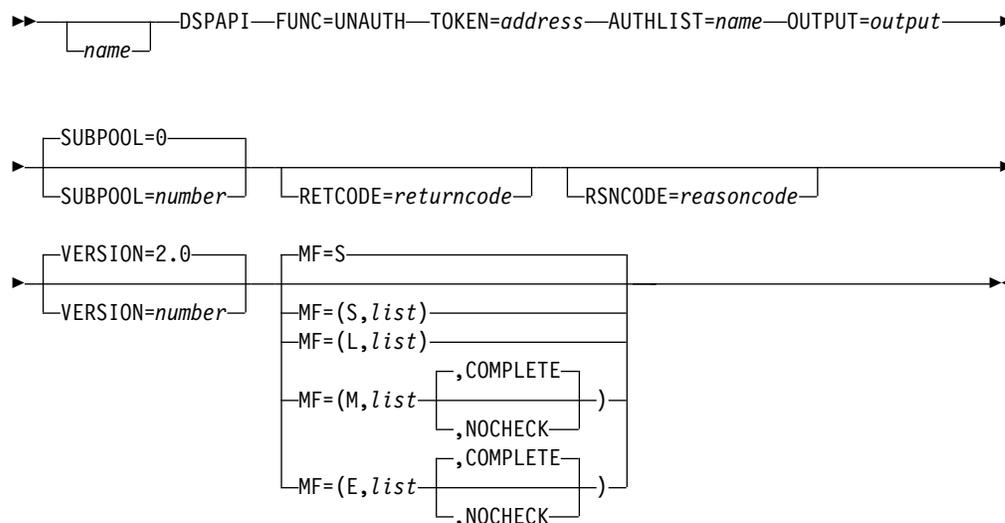
407 ページの『DSPAPI マクロのマクロ形式』



## 第 34 章 DBRC 無許可要求 (UNAUTH)

データベースまたはエリアへの許可を明示的に取り消すには、UNAUTH 要求を使用します。アプリケーションによる許可は、STOPDBRC 要求により暗黙的に取り消されます。UNAUTH は、FUNC=AUTH のオプションです。

### UNAUTH 要求の構文



### UNAUTH 要求のパラメーター

#### name name

オプションのパラメーターです。1 桁目から name を始めます。

#### TOKEN=address | (2-12)

FUNC=STARTDBRC マクロで返された API トークンのアドレスを指定します。

#### AUTHLIST=name | (2 - 12)

無許可にするデータベース名または高速機能エリアのリストを指定します。このリストは、リスト内の要素数を含むフルワード、要素の長さを含むフルワード、および、それに続く 1 つ以上の要素から成ります。各エレメントは 8 文字の DB 名または高速機能 DEDB 名、および、8 文字のブランク (X'40') または高速機能エリア名から成ります。

#### OUTPUT=output | (2-12)

許可出力ブロック DSPAPAUB を示すポインターを受け取るフィールドを指定します。

出力が作成されなかった場合、出力アドレスはゼロです。この状態は、出力を作成できるようになる前にエラーが発生した場合に起こります。

出力ブロックのストレージは、呼び出し元によって事前割り振りされません。DBRC はこれらのブロック用に指定されたサブプールからストレージ

を取得します。呼び出し元は、バッファ解放サービス (DSPAPI FUNC=RELBUF) を使用してこのストレージを解放し、戻された出力アドレスを指定する責任を負います。

#### **SUBPOOL= 0 | number**

取得されるストレージのサブプール番号を指定します。ご使用のプログラムで有効なサブプールについては、「z/OS MVS プログラミング: アセンブラ・サービスガイド」を参照してください。このパラメーターを指定しない場合、FUNC=STARTDBRC 要求によって指定されたサブプールがデフォルトになります。それ以外の場合は、サブプール 0 がデフォルトになります。

#### **RETCODE=returncode | (2-12)**

シンボルとして指定する場合、戻りコードを受け取るために、ストレージのワードを示すラベルを指定します。レジスターとして指定する場合、戻りコードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RETCODE が指定されたかどうかにかかわらず、レジスター 15 は戻りコードを含んでいます。

#### **RSNCODE=reasoncode | (2-12)**

シンボルとして指定する場合、理由コードを受け取るために、そのシンボルはストレージのワードを示すラベルでなければなりません。レジスターとして指定する場合、理由コードを受け取るために、そのレジスターはストレージのワードを示すアドレスを含んでいなければなりません。RSNCODE が指定されたかどうかにかかわらず、レジスター 0 は理由コードを含んでいます。

#### **VERSION=2.0 | number**

このマクロによって生成されるパラメーター・リストのバージョン番号を指定します。

1 つのバージョンに関連するパラメーターを使用するには、そのバージョンかそれ以降のバージョンの番号を指定する必要があります。それより前のバージョン・レベルを指定すると、パラメーターはマクロ処理によって受け入れられず、アセンブリー時にエラー・メッセージが出されます。バージョンの従属関係を持つパラメーターの場合、各要求タイプのパラメーターの説明で、必要なバージョン番号が示されます。

FUNC=UNAUTH 要求で有効なバージョン番号は 2.0 です (デフォルト)。

#### **MF=S | L | M | E**

要求のマクロ形式を指定します。

### **UNAUTH の戻りコードと理由コード**

表 101. DSPAPI FUNC=UNAUTH 要求の戻りコードおよび理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
警告	X'00000008'	X'C1000001'	AUTHLIST の 1 つ以上の項目が処理できませんでした。UNAUTH 出力ブロックの対応する項目に理由コードが設定されています。

表 101. DSPAPI FUNC=UNAUTH 要求の戻りコードおよび理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
重大エラー	X'0000000C'	X'C1000001'	アプリケーションは DBRC にサインオンされていません。
	X'0000000C'	X'C1000004'	アプリケーションがサインオンされていないため、UNAUTH 処理は完了できませんでした。SS レコードが見つかりませんでした。GDB の以前の検査では SS がサインオンされていたことが示されていたため、通常の状態ではこのエラーが発生することはありません。
	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。
ストレージ・エラー	X'00000028'	X'C1000001'	UNAUTH 出力ブロック用のストレージの取得時にエラーが発生しました。
内部エラー	X'0000002C'	X'C1000001'	RECON の複数更新処理を開始しようとしてエラーが発生しました。
	X'0000002C'	X'C1000002'	RECON の複数更新処理を終了しようとしてエラーが発生しました。
	X'0000002C'	X'C1000006'	UNAUTH 出力ブロック内の項目が見つかりませんでした。このエラーが発生することはありません。
	X'0000002C'	X'C1000007'	DBRC 無許可処理中に内部エラーが発生しました。
	X'0000002C'	X'C1000008'	DBRC 無許可処理中に内部エラー (無効なパラメーター) が発生しました。
パラメーター・エラー	X'00000030'	X'C1000001'	AUTHLIST が渡されませんでした。
	X'00000030'	X'C1000002'	項目を持たない AUTHLIST が渡されました。
	X'00000030'	X'C1000003'	AUTHLIST 内のエレメントが重複しています。
	X'00000030'	X'C1000004'	OUTPUT パラメーターが欠落しているか無効です。
	X'00000030'	X'C9000001'	API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

表 101. DSPAPI FUNC=UNAUTH 要求の戻りコードおよび理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'C9000003'	RETCODE アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000005'	OUTPUT アドレスが無効です。API OUTPUT を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C900000A'	要求する機能 (FUNC) に対して、不正な VERSION 値が指定されました。
	X'00000030'	X'C900001A'	AUTHLIST アドレスが無効です。API AUTHLIST を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

関連概念:

403 ページの『第 26 章 DBRC API』

407 ページの『DSPAPI マクロのマクロ形式』

関連資料:

530 ページの『UNAUTH 出力ブロック用の APAUB\_RsnCode』

 DBRC 要求戻りコード (メッセージおよびコード)

## UNAUTH の戻りコードと理由コード

以下の表で、DBRC UNAUTH 要求の戻りコードと理由コードを検索できます。この表には、各コードのタイプおよび説明も記載されています。

表 102. DSPAPI FUNC=UNAUTH 要求の戻りコードおよび理由コード

コード・タイプ	戻りコード	理由コード	意味
	X'00000000'	X'00000000'	要求が正常に完了した。
警告	X'00000008'	X'C1000001'	AUTHLIST の 1 つ以上の項目が処理できませんでした。UNAUTH 出力ブロックの対応する項目に理由コードが設定されています。
重大エラー	X'0000000C'	X'C1000001'	アプリケーションは DBRC にサインオンされていません。

表 102. DSPAPI FUNC=UNAUTH 要求の戻りコードおよび理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'0000000C'	X'C1000004'	アプリケーションがサインオンされていないため、UNAUTH 処理は完了できませんでした。SS レコードが見つかりませんでした。GDB の以前の検査では SS がサインオンされていたことが示されていたため、通常の状態ではこのエラーが発生することはありません。
	X'0000000C'	X'C9000001'	TOKEN が無効です。API に渡された TOKEN ブロックが、FUNC=STARTDBRC 呼び出しによって作成された TOKEN として認識されません。
	X'0000000C'	X'C900000A'	TCB アドレスは、STARTDBRC サービスの発行元の TCB アドレスとは異なります。
ストレージ・エラー	X'00000028'	X'C1000001'	UNAUTH 出力ブロック用のストレージの取得時にエラーが発生しました。
内部エラー	X'0000002C'	X'C1000001'	RECON の複数更新処理を開始しようとしてエラーが発生しました。
	X'0000002C'	X'C1000002'	RECON の複数更新処理を終了しようとしてエラーが発生しました。
	X'0000002C'	X'C1000006'	UNAUTH 出力ブロック内の項目が見つかりませんでした。このエラーが発生することはありません。
	X'0000002C'	X'C1000007'	DBRC 無許可処理中に内部エラーが発生しました。
	X'0000002C'	X'C1000008'	DBRC 無許可処理中に内部エラー (無効なパラメーター) が発生しました。
パラメーター・エラー	X'00000030'	X'C1000001'	AUTHLIST が渡されませんでした。
	X'00000030'	X'C1000002'	項目を持たない AUTHLIST が渡されました。
	X'00000030'	X'C1000003'	AUTHLIST 内のエレメントが重複しています。
	X'00000030'	X'C1000004'	OUTPUT パラメーターが欠落しているか無効です。
	X'00000030'	X'C9000001'	API に渡されたパラメーター・リストに指定された機能 (FUNC) が無効です。
	X'00000030'	X'C9000002'	TOKEN アドレスが無効です。API TOKEN を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

表 102. DSPAPI FUNC=UNAUTH 要求の戻りコードおよび理由コード (続き)

コード・タイプ	戻りコード	理由コード	意味
	X'00000030'	X'C9000003'	RETCODE アドレスが無効です。API RETCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000004'	RSNCODE アドレスが無効です。API RSNCODE を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C9000005'	OUTPUT アドレスが無効です。API OUTPUT を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。
	X'00000030'	X'C900000A'	要求する機能 (FUNC) に対して、不正な VERSION 値が指定されました。
	X'00000030'	X'C900001A'	AUTHLIST アドレスが無効です。API AUTHLIST を含むフィールドのアドレスが、妥当性検査に失敗しました。そのアドレスは、呼び出し側プログラムによって所有されないストレージを指定しています。

関連資料:

 DBRC 要求戻りコード (メッセージおよびコード)

## UNAUTH 出力ブロック用の APAUB\_RsnCode

以下の表で、UNAUTH 要求の戻りコードと理由コードに対応する APAUB\_RsnCode 値を検索できます。この表には、各コードの説明も記載されています。

UNAUTH 出力ブロック (DSPAPAU) が返されると、要求に含まれる DB またはエリアのリストの各エレメントについて、次のいずれかの理由コードがフィールド APAUB\_RsnCode に設定されます。

表 103. UNAUTH 要求の戻りコードおよび理由コードに対する APAUB\_RsnCode 値

APAUB_RsnCode	意味
X'00000000'	要求が正常に完了した。
X'C1000100'	セキュリティー・エラー。SAF または DBRC コマンド許可出口ルーチン (DSPDCAX0) が、ユーザーはこのデータベースまたはエリアに対して、要求を実行することが許可されていないと判断しました。
X'C1000408'	データベースが RECON に登録されていません。
X'C1000410'	サブシステムがデータベースの使用を許可されていません。

表 103. UNAUTH 要求の戻りコードおよび理由コードに対する APAUB\_RsnCode 値 (続き)

APAUB_RsnCode	意味
X'C1000414'	以下のいずれかのアクションが発生した。 <ul style="list-style-type: none"> <li>DBRC 無許可の内部エラー。データベースとサブシステムのレコードが一致しません。</li> <li>テークオーバーの後に、前のアクティブ・サブシステムが無許可処理を要求しました。</li> </ul>
X'C1000418'	DBRC または IMS 無許可の内部エラー。保持されている状態が IMS 互換性評価ルーチンで計算できませんでした。

関連資料:

525 ページの『第 34 章 DBRC 無許可要求 (UNAUTH)』

## UNAUTH 出力ブロックのマッピング

以下の図に、TYPE=UNAUTH 要求からの出力フォーマットを示します。TYPE=UNAUTH 要求の出力ブロックは、DSPAPQHD によってマップされている標準ヘッダーで始まります。この出力ブロックのデータ部分は、DSPAPAUB によってマップされます。

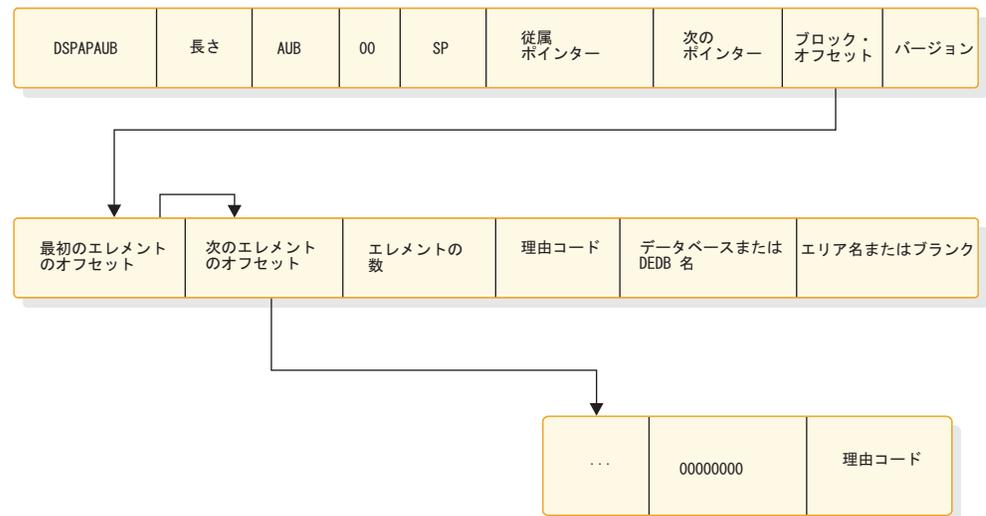


図 29. TYPE=UNAUTH の出力のフォーマット

## UNAUTH 出力ブロック

以下の図に、UNAUTH 要求で返される出力ブロックを示します。出力ブロックは、許可データベースの配列を含み、UNAUTH 要求が正常に実行されたかどうかを示します。

## DSPAPAU8 でマップされた出力ブロックの例

=====

DSPAPAU8

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	8	DSPAPAU8	AUTH/UNAUTH block
0	(0)	UNSIGNED	4	APAU8_OFFSET	Offset to first element
4	(4)	SIGNED	4	APAU8_ELCOUNT	Number of elements in list

OFFSET DECIMAL	OFFSET HEX	TYPE	LENGTH	NAME (DIM)	DESCRIPTION
0	(0)	STRUCTURE	24	APAU8_ELEMENT	
0	(0)	UNSIGNED	4	APAU8_OFFNEXT	Offset to next element
4	(4)	SIGNED	4	APAU8_RSNCODE	Reason code
8	(8)	CHARACTER	8	APAU8_DBNAME	Database or DEDB name
16	(10)	CHARACTER	8	APAU8_AREANAME	Area name or blanks

CONSTANTS

LEN	TYPE	VALUE	NAME	DESCRIPTION
8	CHARACTER	DSPAPAU8	APAU8_EYECATCHER	

---

## 第 6 部 リポジトリ・サーバー・バッチ・インターフェース (FRPBATCH)

リポジトリ・サーバー (RS) アドレス・スペース・バッチ・インターフェース (FRPBATCH) は、RS およびリポジトリを管理するためのバッチ・インターフェースです。



## 第 35 章 FRPBATCH でのコマンド

リポジトリー・サーバー (RS) アドレス・スペース・バッチ・インターフェース (FRPBATCH) は、実行可能ジョブ・ステップ・プログラムとして JCL (ジョブ制御言語) から起動され、SYSIN 入力ストリームを介してコマンドを受け付けます。

一部の MODIFY (F)コマンドと FRPBATCH コマンドは、同等です。

表 104. 同等な Modify (F) コマンドと FRPBATCH コマンド

MODIFY (F)	FRPBATCH	注記
--	ADD	
ADMIN DISPLAY	LIST	
ADMIN START	START	
ADMIN STOP	STOP	IMSRSC リポジトリーを停止する
--	RENAME	
--	DELETE	
ADMIN DSCHANGE	DSCHANGE	
--	UPDATE	
AUDIT	--	監査レベルを変更する
SECURITY	--	ストレージ内にあるプロファイルをリフレッシュする
SHUTDOWN	--	RS を停止する。z/OS STOP (P) インターフェースを介した STOP コマンドに類似。

ジョブ制御ステートメントは以下のとおりです。

### EXEC

プログラム名 (PGM=FRPBATCH) およびプログラム・パラメーターを指定します。パラメーターはコンマで区切る必要がありますが、任意の順序で指定できます。各パラメーターは、parameter=value というフォーマットになります。

### XCFGROUP

RS が置かれている XCF グループの名前。このパラメーターの値は、IMS PROCLIB データ・セットの FRPCFG メンバー内の XCF\_GROUP\_NAME パラメーターの値と同じです。

### LANG

出力メッセージの言語。ENU のみサポートされます。このパラメーターを省略した場合、ENU が使用されます。

以下の例は、EXEC ステートメントです。

```
EXEC PGM=FRPBATCH,PARM=('XCFGROUP=FRPGRUP1','LANG=ENU')
```

## SYSPRINT DD

汎用メッセージおよび情報のデータ・セットを定義します。このデータ・セットの DCB 属性は、RECFM=FBM および LRECL=133 です。

## FRPLIST DD

LIST コマンド出力のデータ・セットを定義します。このステートメントを省略した場合、LIST コマンドの出力は SYSPRINT DD 出力データ・セットに書き込まれます。このデータ・セットの DCB 属性は、RECFM=FBM および LRECL=133 です。

処理対象のコマンドは、SYSIN 制御カードで指定されます。SYSIN 制御カードは、入力ストリームの 1 桁目から 72 桁目までに入力する必要があります。各ステートメントは、以下のような汎用フォームになります。

```
command_name parameter1 parameter2(value) /*inline comment  
*Full-line comment
```

以下の規則が適用されます。

- *command\_name* がコマンドの先頭項目です。これに 1 スペース ( ) と、1 つ以上のパラメーターが続きます。
- コマンドに複数のパラメーターが含まれる場合、各パラメーターは 1 つ以上のスペース ( ), 1 つのコンマ (,), あるいはその両方で区切る必要があります。
- パラメーターには値を入れることができます。パラメーターの値は括弧 ( ) で囲みます。
- コマンド名、パラメーター、および値は、プログラムによって大文字に変換されます。
- インライン・コメントを入力するには、スラッシュの後にアスタリスクを続けて入力します (/\*)。それ以降の同一行上に文字があっても、プログラムによって無視されます。
- フルライン・コメントを入力するには、1 桁目にアスタリスク (\*) を入力します。その行のすべての文字は、プログラムによって無視されます。
- 複数行にまたがる 1 つのコマンドをブレイクするには、以下のいずれかの継続文字を使用します。

- 複数行にまたがる同一コマンドの複数のパラメーターを分離するには、ハイフン (-) を使用します。ハイフンがあれば、継続行の先頭の区切り記号は削除されません。例えば、次のようにします。

```
RENAME REPOSITORY(REPOSITORY_NAME) -  
REPOSITORYNEW(REPNEWNAME)
```

- 単一のパラメーターとその値を複数行に入力するには、正符号 (+) を使用します。これにより、継続行の先頭の区切り記号が削除されます。正符号は、行の最後の文字の直後に続ける必要があります。例えば、次のようにします。

```
START REPOSITORY  
(REPOSIT+  
ORY_NAME)
```

以下のサンプルは、さまざまな FRPBATCH コマンドを指定した FRPBATCH インターフェイスを実行する JCL です。

```

//FRPBAT EXEC PGM=FRPBATCH,PARM='XCFGROUP=FRPGRP1'
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
//*
ADD REPOSITORY(IMSRSC_REPOSITORY)+
  REPDSN1RID(IMSTESTS.REPO.IMPRI.RID)+
  REPDSN1RMD(IMSTESTS.REPO.IMPRI.RMD)+
  REPDSN2RID(IMSTESTS.REPO.IMSSEC.RID)+
  REPDSN2RMD(IMSTESTS.REPO.IMSSEC.RMD)+
  REPDSN3RID(IMSTESTS.REPO.IMSSPR.RID)+
  REPDSN3RMD(IMSTESTS.REPO.IMSSPR.RMD)+
  AUTOOPEN(YES)
//*
START REPOSITORY(IMSRSC_REPOSITORY)+
  MAXWAIT(30,CONTINUE)

//*
LIST REPOSITORY(IMSRSC_REPOSITORY)

//*
STOP REPOSITORY(IMSRSC_REPOSITORY)+
  MAXWAIT(30,CONTINUE)

//*
RENAME REPOSITORY(IMSRSC_REPOSITORY) REPOSITORYNEW(IMSRSC_TEST_REPOSITORY)

//*
UPDATE REPOSITORY (IMSRSC_TEST_REPOSITORY) -
  REPDSN1RID(IMSTESTS.TESTREPO.IMPRI.RID) -
  REPDSN1RMD(IMSTESTS.TESTREPO.IMPRI.RMD) -
  AUTOOPEN(NO)

//*
DELETE REPOSITORY(IMSRSC_TEST_REPOSITORY)

```

FRPBATCH コマンドは、以下の機能を提供します。

関連概念:

-  [IMSRSC リポジトリーの概要 \(システム定義\)](#)
-  [IMSRSC リポジトリーの管理 \(システム管理\)](#)
-  [IMSRSC リポジトリーの開始および停止 \(オペレーションおよびオートメーション\)](#)
-  [IMSRSC リポジトリーのオープン \(オペレーションおよびオートメーション\)](#)

関連資料:

-  [F reposervername,ADMIN \(コマンド\)](#)
-  [IMS PROCLIB データ・セットの FRPCFG メンバー \(システム定義\)](#)

---

## FRPBATCH での ADD コマンド

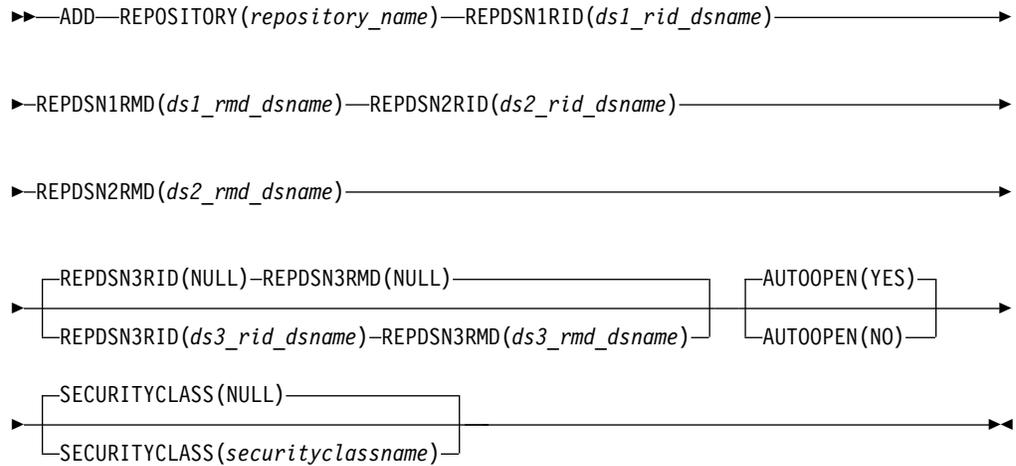
ADD FRPBATCH コマンドは、IMSRSC リポジトリーをリポジトリー・サーバー (RS) カタログ・リポジトリー・データ・セットに追加するために使用します。

以下のサブセクションがあります。

- 538 ページの『構文』

- 『キーワード』

## 構文



## キーワード

### REPOSITORY ( )

これは必須キーワードです。リポジトリの名前。名前は 44 文字まで指定できます。有効な文字は、A から Z (大文字のみ)、0 から 9、そして以下の記号、すなわち、番号記号(#)、ドル記号(\$)、アットマーク(@)、ピリオド(.)、および下線(\_)です。すべての小文字は、大文字に変換されます。

「CATALOG」というリポジトリ名は、RS が使用するために予約されており、使用できません。

### REPDSN1RID ( )

これは必須キーワードです。1 次リポジトリ索引データ・セット (RID)。リポジトリを初期設定するには、有効な既存の VSAM キー順データ・セット (KSDS) 名が必要です。

### REPDSN1RMD ( )

これは必須キーワードです。1 次リポジトリ・メンバー・データ・セット (RMD)。リポジトリを初期設定するには、有効な既存の VSAM KSDS 名が必要です。

### REPDSN2RID ( )

これは必須キーワードです。副次 RID。リポジトリを初期設定するには、有効な既存の VSAM KSDS 名が必要です。

### REPDSN2RMD ( )

これは必須キーワードです。副次 RMD。リポジトリを初期設定するには、有効な既存の VSAM KSDS 名が必要です。

### REPDSN3RID ( )

このキーワードはオプションです。予備 RID。このパラメーターが指定されていないと、予備データ・セットは NONE にセットされます。指定する場合は、有効な VSAM KSDS データ・セット名もしくは NULL であることが必要です。REPDSN3 データ・セット・ペアを NULL にセットすると、その状況は

NONE にセットされることとなります。予備は、データ・セット・リカバリー処理が開始されるまで、割り振られません。

#### **REPDSN3RMD()**

このキーワードはオプションです。予備 RMD。このパラメーターが指定されていないと、予備データ・セットは NONE にセットされます。指定する場合は、有効な VSAM KSDS データ・セット名もしくは NULL であることが必要です。REPDSN3 データ・セット・ペアを NULL にセットすると、その状況は NONE にセットされることとなります。予備は、データ・セット・リカバリー処理が開始されるまで、割り振られません。

#### **AUTOOPEN(YES | NO)**

このキーワードはオプションです。リポジトリ・データ・セットが割り振られる時点を指定します。

##### **YES**

リポジトリ・データ・セットは、リポジトリが開始される時に割り振られます。これはデフォルトです。

**NO** リポジトリ・データ・セットは、ユーザーが最初にリポジトリに接続する時に割り振られます。

#### **SECURITYCLASS(NULL | *securityclassname*)**

このキーワードはオプションです。ユーザー・リポジトリに使用するセキュリティ・クラスの名前を指定します。名前は左寄せされた 8 バイト名で、連続スペースが後続する必要があります。先頭文字は英字で、英数字の名前文字が後続する必要があります。

このパラメーターが省略された場合、または NULL が指定された場合は、このユーザー・リポジトリに対するセキュリティは存在しません。指定する名前は、IMS PROCLIB データ・セットの FRPCFG メンバー内の RS アドレス・スペースで指定された SAF\_CLASS と同じ名前にすることができます。

関連概念:

 [IMSRSC リポジトリの概要 \(システム定義\)](#)

 [IMSRSC リポジトリおよび RS カタログ・リポジトリ・データ・セット \(システム定義\)](#)

関連タスク:

 [RS カタログ・リポジトリおよび IMSRSC リポジトリへのアクセスの制限 \(システム管理\)](#)

 [IMSRSC リポジトリ・データ・セットの割り振り \(システム定義\)](#)

関連資料:

 [IMS PROCLIB データ・セットの FRPCFG メンバー \(システム定義\)](#)

---

## **FRPBATCH での DELETE コマンド**

DELETE FRPBATCH コマンドは、IMSRSC リポジトリをリポジトリ・サーバー (RS) カタログ・リポジトリ・データ・セットから除去するために使用します。

注: リポジトリを RS カタログ・リポジトリから除去する場合、リポジトリのデータ・セットは削除されません。データ・セットを削除するには、リポジトリを RS カタログ・リポジトリから除去した後で、z/OS Access Method Services (IDCAMS) ユーティリティまたは類似の手段を使用します。

以下のサブセクションがあります。

- 『構文』
- 『キーワード』

## 構文

▶▶—DELETE—REPOSITORY(*repository\_name*)—▶▶

## キーワード

### REPOSITORY()

これは必須キーワードです。除去するリポジトリの名前。「CATALOG」は内部使用のために予約されており、削除することはできません。

関連概念:

📄 IMSRSC リポジトリの概要 (システム定義)

関連タスク:

📄 RS カタログ・リポジトリからの IMSRSC リポジトリの除去 (システム管理)

関連資料:

📄 z/OS: DFSMS カタログのためのアクセス方式サービス・プログラム

---

## FRPBATCH での DSCHANGE コマンド

DSCHANGE FRPBATCH コマンドは、IMSRSC リポジトリ・データ・セットペアの状況を、DISCARD または SPARE に変更するために使用します。

以下のサブセクションがあります。

- 『構文』
- 『キーワード』

## 構文

▶▶—DSCHANGE—REPOSITORY(*repository\_name*)—RDS(

1
2
3

)—ACTION(

SPARE
DISCARD

)—▶▶

## キーワード

### REPOSITORY()

これは必須キーワードです。変更するリポジトリの名前。「CATALOG」は予約されており、変更することはできません。

### RDS(1 | 2 | 3)

これは必須キーワードです。範囲 1 から 3 までの数値。これにより、要求された DSCHANGE アクションが適用されるリポジトリー・データ・セット・ペアを識別します。

- 1 次リポジトリー・データ・セット・ペア (COPY1)。
- 2 副次リポジトリー・データ・セット・ペア (COPY2)。
- 3 予備リポジトリー・データ・セット・ペア (SPARE)。

### ACTION(SPARE | DISCARD)

このキーワードは必須です。RDS パラメーターで指定されたリポジトリー・データ・セットに対して適用するアクション。

#### SPARE

リポジトリー・データ・セット・ペアのファイル属性指定を SPARE 状況に変更することを要求します。SPARE アクションは、DISCARD 状況になっているリポジトリー・データ・セット・ペアに対してのみ、実行することができます。SPARE リポジトリー索引データ・セット (RID) および SPARE リポジトリー・メンバー・データ・セット (RMD) は、空であることが必要です。

#### DISCARD

リポジトリー・データ・セット・ペアのファイル属性指定を DISCARD 状況に変更することを要求します。DISCARD アクションは、アクティブなりポジトリー・データ・セット (COPY1 または COPY2) もしくは SPARE リポジトリー・データ・セットのいずれかに対して、実行できます。データ・セットが DISCARD 状況に設定されないと、新規データ・セットを定義することはできません。この要求が COPY1 または COPY2 に対する DISCARD アクションを処理するには、リポジトリーを停止状態にする必要があります。SPARE に対する DISCARD を処理するために、リポジトリーを停止する必要はありません。

関連概念:

-  IMSRSC リポジトリーの概要 (システム定義)
-  IMS リポジトリー・データ・セットの状態 (システム定義)

---

## FRPBATCH での LIST コマンド

LIST FRPBATCH コマンドは、1 つの IMSRSC リポジトリー、またはリポジトリー・サーバー (RS) カタログ・リポジトリー・データ・セットに定義されているすべてのリポジトリーについての情報をリストするために使用します。

以下のサブセクションがあります。

- 542 ページの『構文』
- 542 ページの『キーワード』

## 構文

```
►►—LIST—REPOSITORY(repository_name)—STATUS—◄◄
```

## キーワード

### REPOSITORY()

RS カタログ・リポジトリ・データ・セットに定義されている特定のリポジトリの詳細をリストします。情報をリストする対象となるリポジトリの名前。

### STATUS

RS カタログ・リポジトリ・データ・セットに定義されているすべてのリポジトリの詳細をリストします。

以下の情報が戻されます。

- リポジトリの名前
- リポジトリの状況
- リポジトリが最後に更新された日付と、更新したユーザーのユーザー ID

関連概念:

 [IMSRSC リポジトリの概要 \(システム定義\)](#)

関連タスク:

 [IMSRSC リポジトリの定義と状況の表示 \(システム管理\)](#)

---

## FRPBATCH での RENAME コマンド

RENAME FRPBATCH コマンドは、IMSRSC リポジトリの名前を変更するために使用します。

以下のサブセクションがあります。

- 『構文』
- 『キーワード』

## 構文

```
►►—RENAME—REPOSITORY(repository_name)—REPOSITORYNEW(repository_new_name)—◄◄
```

## キーワード

### REPOSITORY()

これは必須キーワードです。名前変更するリポジトリの現行名です。

「CATALOG」は内部使用のために予約されており、名前変更することはできません。

### REPOSITORYNEW()

これは必須キーワードです。名前変更するリポジトリの新規名です。名前は 44 文字まで指定できます。有効な文字は、A から Z (大文字のみ)、0 から

9、以下の記号、すなわち、ピリオド (.)、下線 (\_)、番号記号 (#)、ドル記号 (\$)、およびアットマーク (@) です。すべての小文字は、大文字に変換されます。

「CATALOG」というリポジトリ名は、リポジトリ・サーバー (RS) が使用するために予約されており、使用できません。

関連概念:

➡ IMSRSC リポジトリの概要 (システム定義)

➡ RS カタログ・リポジトリ内の IMSRSC リポジトリ指定の更新 (システム管理)

---

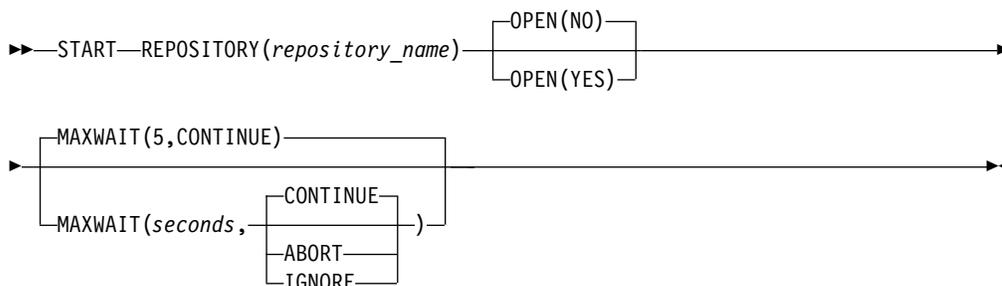
## FRPBATCH での START コマンド

START FRPBATCH コマンドは、IMSRSC リポジトリを開始するために使用します。

以下のサブセクションがあります。

- 『構文』
- 『キーワード』

### 構文



### キーワード

#### REPOSITORY()

これは必須キーワードです。開始するリポジトリ・データ・セットの名前。リポジトリ・サーバー (RS) カタログ・リポジトリ・データ・セットは、START コマンドを使用して開始することはできません。

#### OPEN(NO | YES)

このキーワードはオプションです。リポジトリが開始された後に、即時にリポジトリ・データ・セットを開くかどうか、指定します。

**NO** リポジトリ・データ・セットは、ユーザーが最初にこのリポジトリに接続した時に開かれます。あるいは、リポジトリに対して ADD FRPBATCH コマンドで AUTOOPEN=Y が定義されている場合は、即時に開かれます。これはデフォルトです。

#### YES

リポジトリ・データ・セットは、リポジトリが開始される時に割り振られ、開かれます。

### MAXWAIT(5 | seconds , CONTINUE | IGNORE | ABORT)

このキーワードはオプションです。START 操作が完了するのをユーティリティーが待機する最大秒数 (0 から9999)。デフォルトは 5 秒です。値 0 は、ユーティリティーは待機せずに即時に戻りコードを出すという意味です。

MAXWAIT では、MAXWAIT 期間が満了した時点で START 操作が完了していない時 (サーバーに対するユーザー・リポジトリ開始要求は成功したが、リポジトリが要求した状態になったことをバッチ・ユーティリティーが確認できない場合) に取るべきアクションも、指定します。

#### CONTINUE

コマンド処理は、MAXWAIT 期間が満了しても続行されます。戻りコードは 4 にセットされます。

#### IGNORE

コマンド処理は、MAXWAIT 期間が満了しても続行されます。戻りコードはセットされません。

#### ABORT

コマンド処理は、MAXWAIT 期間が満了した時は強制終了されます。戻りコードは 8 にセットされます。

関連概念:

 IMSRSC リポジトリの概要 (システム定義)

---

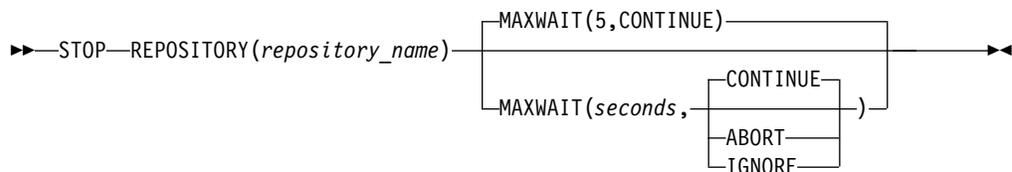
## FRPBATCH での STOP コマンド

STOP FRPBATCH コマンドは、リポジトリ・サーバー (RS) カタログ・リポジトリ・データ・セットに定義されている IMSRSC リポジトリを停止するために使用します。

以下のサブセクションがあります。

- 『構文』
- 『キーワード』

### 構文



### キーワード

#### REPOSITORY()

これは必須キーワードです。停止するリポジトリ・データ・セットの名前。RS カタログ・リポジトリ・データ・セットは、STOP コマンドを使用して停止することはできません。

このリポジトリを停止するよう RS に要求し、以降はこのリポジトリに接続しないようにします。リポジトリ・データ・セットが割り振られている場合は、サーバーはすべての書き込み操作が完了した後にデータ・セットを割り振り解除して、リポジトリを閉じます。

#### MAXWAIT(5 | *seconds* , CONTINUE | IGNORE | ABORT)

このキーワードはオプションです。ユーティリティーが STOP 操作の完了を待機する、秒単位の最大時間 (0 から 9999)。デフォルトは 5 秒です。値 0 は、ユーティリティーは待機せずに即時に戻りコードを出すという意味です。MAXWAIT では、MAXWAIT 期間が満了した時点で STOP 操作が完了していない時 (サーバーに対するユーザー・リポジトリ停止要求は成功したが、リポジトリが要求した状態になったことをバッチ・ユーティリティーが確認できない場合) に取るべきアクションも、指定します。

##### CONTINUE

コマンド処理は、MAXWAIT 期間が満了しても続行されます。戻りコードは 4 にセットされます。

##### IGNORE

コマンド処理は、MAXWAIT 期間が満了しても続行されます。戻りコードはセットされません。

##### ABORT

コマンド処理は、MAXWAIT 期間が満了した時は強制終了されます。戻りコードは 8 にセットされます。

関連概念:

➡ IMSRSC リポジトリの概要 (システム定義)

---

## FRPBATCH での UPDATE コマンド

UPDATE FRPBATCH コマンドは、リポジトリ・サーバー (RS) カタログ・リポジトリ・データ・セット内の IMSRSC リポジトリ定義を更新するために使用します。このコマンドを使用して、IMSRSC リポジトリ・データ・セットまたはリポジトリの AUTOOPEN 指定を変更します。

以下のサブセクションがあります。

- 『構文』
- 546 ページの『キーワード』

### 構文

```

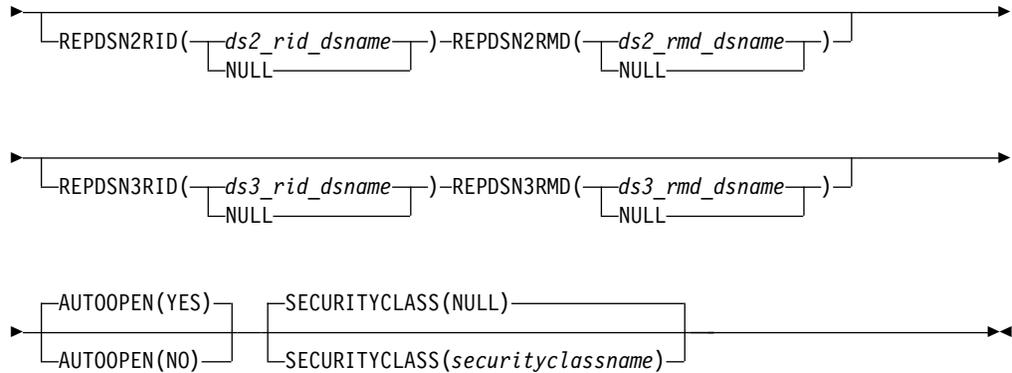
▶▶—UPDATE—REPOSITORY(repository_name)————▶▶

```

```

▶——┌──REPDSN1RID(┌──ds1_rid_dsname──┐)──REPDSN1RMD(┌──ds1_rmd_dsname──┐)──┐
      └──NULL──┘                                └──NULL──┘

```



## キーワード

### REPOSITORY()

これは必須キーワードです。更新するリポジトリの名前。「CATALOG」は予約されており、更新することはできません。

### REPDSN1RID()

これは、オプション・キーワードです。1 次リポジトリ索引データ・セット (RID) の名前。このパラメーターが省略された場合、または NULL が指定された場合は、データ・セットは更新されません。

REPDSN1RMD を指定したが、このパラメーターを更新したくない場合は、このパラメーターを (NULL を使用して) 指定する必要があります。

### REPDSN1RMD()

これは、オプション・キーワードです。1 次リポジトリ・メンバー・データ・セット (RMD) の名前。このパラメーターが省略された場合、または NULL が指定された場合は、データ・セットは更新されません。

REPDSN1RID() を指定したが、このパラメーターを更新したくない場合は、このパラメーターを (NULL を使用して) 指定する必要があります。

### REPDSN2RID()

これは、オプション・キーワードです。副次 RID の名前。このパラメーターが省略された場合、または NULL が指定された場合は、データ・セットは更新されません。

REPDSN2RMD() を指定したが、このパラメーターを更新したくない場合は、このパラメーターを (NULL を使用して) 指定する必要があります。

### REPDSN2RMD()

これは、オプション・キーワードです。副次 RMD の名前。このパラメーターが省略された場合、または NULL が指定された場合は、データ・セットは更新されません。

REPDSN2RID() を指定したが、このパラメーターを更新したくない場合は、このパラメーターを (NULL を使用して) 指定する必要があります。

### REPDSN3RID()

このキーワードはオプションです。予備 RID の名前。このパラメーターが省略された場合、または NULL が指定された場合は、データ・セットは更新されません。

REPDSN3RMD() を指定したが、このパラメーターを更新したくない場合は、このパラメーターを (NULL を使用して) 指定する必要があります。

#### **REPDSN3RMD()**

このキーワードはオプションです。予備 RMD の名前。このパラメーターが省略された場合、または NULL が指定された場合は、データ・セットは更新されません。

REPDSN3RID() を指定したが、このパラメーターを更新したくない場合は、このパラメーターを (NULL を使用して) 指定する必要があります。

#### **AUTOOPEN(YES | NO)**

このキーワードはオプションです。リポジトリ・データ・セットが割り振られる時点を指定します。

##### **YES**

リポジトリ・データ・セットは、リポジトリが開始される時に割り振られます。これはデフォルトです。

**NO** リポジトリ・データ・セットは、ユーザーが最初にリポジトリに接続する時に割り振られます。

#### **SECURITYCLASS(NULL | *securityclassname*)**

このキーワードはオプションです。ユーザー・リポジトリに使用するセキュリティ・クラスの名前を指定します。名前は左寄せされた 8 バイト名で、連続スペースが後続する必要があります。先頭文字は英字で、英数字の名前文字が後続する必要があります。

このパラメーターが省略された場合、または NULL が指定された場合は、このユーザー・リポジトリに対するセキュリティは存在しません。指定する名前は、IMS PROCLIB データ・セットの FRPCFG メンバー内の RS アドレス・スペースで指定された SECURITYCLASS と同じ名前にすることができます。

関連概念:

 [IMSRSC リポジトリの概要 \(システム定義\)](#)

 [RS カタログ・リポジトリ内の IMSRSC リポジトリ指定の更新 \(システム管理\)](#)

関連資料:

 [IMS PROCLIB データ・セットの FRPCFG メンバー \(システム定義\)](#)



---

## 第 7 部 VTAM および SNA の参照情報

このセクションのトピックでは、VTAM および SNA に関する参照情報について説明しています。



## 第 36 章 SLU P と LU 6.1 のバインド・パラメーター

以下のトピックでは、VTAM OPNDST マクロ命令を使用して接続を設定するときに IMS が指定するセッション・パラメーターを検索できます。これらのパラメーターは、IMS との通信時に論理装置が従わなければならない規則を定義します。

制約事項: IMS からのバインドを必要とする論理装置の場合には、OUTBUF および RECANY のバッファー・サイズに関する VTAM 制約事項があります。

次のアウトバウンド (IMS から SLU まで) のバインド形式は、以降のトピックに示された装置タイプに適用されます。

関連資料:

569 ページの『第 38 章 CINIT ユーザー・データ・パラメーターの形式』

## 金融機関通信システム・バインド・パラメーター

以下の表で、TYPE マクロまたは拡張端末オプション (ETO) ログオン記述子上で UNITYPE=FINANCE または UNITYPE=SLUTYPEP として定義された金融機関通信システムのセッション・パラメーターを検索できます。

表 105. 金融機関通信システム・バインド・パラメーター:

バイト	ビット	内容	BIND FORMAT の説明
0		X'31'	
1		X'01'	Format=0; BINDTYPE=COLD <sup>1</sup>
2		X'04'	FM プロファイル 4
3		X'04'	TS プロファイル 4
4			基本 NAU プロトコル
	0	B'1'	複数の RU チェーン
	1	B'0'	即時要求
	2-3	B'11'	応答
	4-5	B'00'	予約済み
	6	B'0'	圧縮なし
	7	B'1'	基本は EB を送信することができる
5		B'1'	2 次 NAU プロトコル
	0	B'1'	複数の RU チェーン
	1	B'0'	即時要求
	2-3	B'11'	応答
	4-5	B'00'	予約済み
	6	B'0'	圧縮なし
	7	B'1'	2 次は EB を送信することができる

表 105. 金融機関通信システム・バインド・パラメーター (続き):

バイト	ビット	内容	BIND FORMAT の説明
6		X'60'	共通 NAU プロトコル
	0	B'0'	予約済み
	1	B'1'	メッセージ・ヘッダーが許可される
	2	B'1'	使用されるブラケット
	3	B'0'	ブラケット終了規則 2 (無条件)
	4	B'0'	代替コードなし
	5-7	B'000'	予約済み
7		X'80'	共通 NAU プロトコル
	0-1	B'10'	FM 伝送モード - HDX フリップフロップ
	2	B'0'	基本 ERP 責任
	3	B'0'	2 次ステーションがファースト・スピーカー
	4-5	B'00'	予約済み
	6	B'0'	関連チェーンなし
	7	B'0'	コンテンション解消
8			予約済み: X'00' 指定する必要がある
9			SLU 受信ペーシング・カウント: IMS によって変更されない <sup>2</sup>
10			SLU 最大 RU 送信サイズ: IMS システム定義からの任意のバッファー・サイズの最大受信に設定する 1, 3, 4
11			PLU 最大 RU 送信サイズ: IMS システム定義で指定された出力バッファー・サイズから設定する <sup>1, 4</sup>
12-13			予約済み: X'0000' を指定する必要がある
14			LU TYPE: X'00' を指定する必要がある
15-26			予約済み : XL11'00' を指定する必要がある
27			PLU 名の長さ <sup>2</sup>
28-35			PLU 名: IMS ACB 名 <sup>2</sup>
36		X'0B'	ユーザー・データ長
37-47			ユーザー・データ <sup>5</sup>
37		X'00'	構造化データ標識
38		X'09'	USERVAR セグメントの長さ
39		X'03'	USERVAR セグメント標識
40-47		XRF USERVAR	USERVAR

表 105. 金融機関通信システム・バインド・パラメーター (続き):

バイト	ビット	内容	BIND FORMAT の説明
注:			
1		1 バイトから 7 バイトまでと、10、および 11 バイトは、IMS によって設定された値に設定されます。ユーザーが、これらの値を変更することはできません。ペーシング・パラメーターは、VTAM リスト LU 定義で、または LU (VTAM DLOGMOD パラメーター) のモード・テーブル項目によって定義されます。	
2		0 バイト、9 バイト、および 27 から 35 バイトまでは、VTAM によって設定されます。	
3		受信バッファー・サイズは、28 バイト未満の COMM マクロ・ステートメントの RECANY キーワード・パラメーター上のユーザー提供のサイズの値によって判別されません。	
4		IMS は、4701/4702 に対してバッファー・サイズを設定しません。	
5		発行された BIND データが MNPS ではなく USERVAR を使用する XRF システムである場合には、次の構造化ユーザー・セグメントが組み込まれています。	
		USERVAR セグメントの長さ - X'09'	
		USERVAR セグメント標識 - X'03'	
		USERVAR (8 バイト) - XRF システムの USERVAR	

関連資料:

 z/OS: 要求単位 (RU) のフォーマット

## 1 次ハーフセッションとしての IMS

IMS より他へのセッション開始時に、IMS が指定されたバインド形式を送信します。モード・テーブル項目が交渉バインドを示している場合には、IMS は、バインドを送信する前に、指示された値でモード・テーブル基本 NAU プロトコル・フィールドをオーバーライドします。

IMS を使用すると、VTAM モード・テーブル項目または交渉バインド応答によって一部のパラメーターを任意に設定することができます。その後で、IMS は指示された制約範囲内で操作します。非交渉バインドの場合には、IMS がバインドを送信する前にパラメーターを検査して、妥当性を調べます。交渉バインドの場合には、BIND 要求の送信前およびバインド応答の受信時に IMS がパラメーターの妥当性を検査します。その理由は、2 次ハーフセッションが、以下の表に示されている制約範囲内でパラメーターを変更することがあるからです。

表 106. 論理装置タイプ 6.1 バインド・パラメーター:

バイト	ビット	内容	BIND FORMAT の説明 <sup>1</sup>
0		X'31'	バインド要求コード
1	0-3	B'0000'	FORMAT: TYPE 0
	4-7	B'0000', B'0001'	BIND タイプ 0000 - 交渉 0001 - 非交渉
2	0-7	X'12'	FM プロファイル 18
3	0-7	X'04'	TS プロファイル 4

表 106. 論理装置タイプ 6.1 バインド・パラメーター (続き):

バイト	ビット	内容	BIND FORMAT の説明 <sup>1</sup>
4			基本 NAU プロトコル
	0	B'1'	複数の RU チェーン
	1	B'0'	即時要求
	2-3		10 - 確定応答チェーン
		B'10', B'11' (交渉 BIND 用に設定された B'11')	11- 例外時 / 確定応答チェーン
	4	B'0'	作成を送信できない
	5	B'0'	予約済み
	6	B'0'	圧縮なし
5	7	B'1'	0- 基本がブラケットを終了することができない  1- 基本が終了ブラケットを送信することができる
			2 次 NAU プロトコル
	0	B'1'	0- 単一 RU チェーン、1- 複数 RU チェーン
	1	B'0'	即時要求
			01 - 例外時応答チェーン
			10 - 確定応答チェーン
	2-3	B'10', B'11'	11- 例外時 / 確定応答チェーン
	4	B'0'	作成を送信できない
6	5	B'0'	予約済み
	6	B'0'	圧縮なし
			0 - 2 次はブラケットを終了することができない
	7	B'0', B'1'	1 - 2 次はブラケット終了を送信することができる
			共通 NAU プロトコル
	0	B'0'	予約済み
	1	B'1'	機能管理ヘッダーが許可される  ブラケット状態
			0 - ブラケット状態マネージャーのイン・ブラケット状態へのリセット
2	B'0', B'1'	1 - ブラケット状態マネージャーのブラケット間状態へのリセット <sup>2</sup>	
7	3	B'1'	条件付きブラケット終了
	4	B'0'	代替コードなし
	5		STSN 要求フラグ
		B'0' シーケンス 使用不能 B'1' シーケンス番号 使用可能	
	6		BIS がフラグを送信
		B'0' BIS が送信しなかった B'1' BIS が送信した	
	7	B'0'	予約済み
	0-1	B'10'	FM トランザクション・モード、HDX-FF
8	2	B'1'	送信側 ERP
	3	B'0'	2 次がファースト・スピーカーである
	4-6	B'0000'	予約済み
			その時バイト 6 のビット 2 が 0 の場合  0 - 2 次が、データ・トラフィックを アクティブ状態にした後で 最初に発信する
	7	B'0', B'1'	1 - 基本が、データ・トラフィックをアクティブ状態にした後で最初に発信する <sup>2</sup>
8	0-7	未変更	2 次送信ベージング・カウント

表 106. 論理装置タイプ 6.1 バインド・パラメーター (続き):

バイト	ビット	内容	BIND FORMAT の説明 <sup>1</sup>
9	0-7	未変更	2 次受信ペーシング・カウント
10	0-7	ユーザー定義の受信からセットされる	SLU 最大送信 RU サイズ <sup>3</sup>
11	0-7	ユーザー定義出力バッファ・サイズからセットされる	PLU 最大送信 RU サイズ
12	0-7	未変更	基本送信ペーシング・カウント
13	0-7	未変更	基本受信ペーシング・カウント
			表示サービス
14	0-7	X'06'	LU プロファイル (LUTYPE6)
15	0-7	X'00' 予約済み	機能管理ヘッダー・サブセット
16	0	予約済み	1 次ハーフセッション・フラグ
	1	予約済み	
	2	1 - FMH6: 受信	
		SYSMSG がサポートされる	
	3	1 - 受信 SCHEDULER	
		モデルがサポートされる	
	4	1 - 受信 QMODEL がサポートされる	
	5	0 - 線形ファイル・モデルが無視される	
	6	0 - DL/I モデルが無視される	
	7	予約済み	
17		予約済み	
18-19		予約済み	
20	0	予約済み	2 次ハーフセッション・フラグ
	1	予約済み	
	2	1 - 受信 SYSMSG がサポートされる	
	3	0 - 受信 SCHEDULER	
		サポートされない <sup>4</sup>	
		1 - 受信 SCHEDULER	
		モデルがサポートされる	
	4	0 - 受信 QMODEL が	
		無視される1 - 受信 QMODEL が	
		サポート	
	5	0 - 線形ファイル・モデルが無視される	
	6	0 - DL/I モデルが無視される	
	7	予約済み	
21		予約済み	
22-26		予約済み	
27	0-7		PLU 名の長さ
28-M			PLU 名
M+1	0-7	X'00' ユーザー・データなし	ユーザー・データの長さ
M+2-N		X'00' 構造化フィールドが続く	ユーザー・データ
		X'00' 未構造化ユーザー・データの最初のバイト <sup>5</sup>	
M+3-N		未構造化 ユーザー・データの 残り	未構造化ユーザー・データの場合、
M+3-N		構造化フィールド	構造化ユーザー・データの場合 <sup>6</sup>

表 106. 論理装置タイプ 6.1 バインド・パラメーター (続き):

バイト	ビット	内容	BIND FORMAT の説明 <sup>1</sup>
N+1	0-7	構造化フィールド、要求相関 (URC) フィールド	URC の長さ <sup>7</sup>
		X'00' = URC が存在しない	
N+2-P		エンド・ユーザー定義 ID	URC <sup>4</sup>
P+1	0-7	X'00' = 2 次名なし	2 次 LU 名の長さ <sup>4</sup>
P+2-R	0-7	2 次 LU 名	2 次 LU 名 <sup>4</sup>

表 106. 論理装置タイプ 6.1 バインド・パラメーター (続き) :

バイト	ビット	内容	BIND FORMAT の説明 <sup>1</sup>
-----	-----	----	------------------------------

注:

1. BIND RU の長さは 256 バイトを超えることはできません。超えた場合は否定応答が戻されます。
2. 可能なブラケット内またはプロセス再始動を指示するために設定します。応答モード出力がキューに残っている場合、あるいは IMS が会話型モードになっている場合は、バインド時に IMS が設定します。また、交渉バインド応答時にもう一方のハーフセッションが送信することもできます。
3. 受信用バッファ・サイズは、28 バイト未満の COMM マクロ・ステートメントの RECANY キーワード・パラメーター上のユーザー提供のサイズの値によって判別されます。
4. もう一方のハーフセッションが SCHEDULER プロセスをサポートしていないことがバインドによって示された場合は、IMS は ATTACH を使用してすべての非送信請求出力および非同期出力を送信します。
5. 未構造化ユーザー・データが無視され、IMS によって提供されません。
6. 構造化ユーザー・データ形式。構造化フィールドには、設計済みまたはサブシステム定義の情報が含まれていて、そのフィールドによってサブシステムがデータを通信する方法が提供されます。各構造化フィールドには、フィールド ID (サブフィールド番号) および長さが含まれています。構造化データ・フィールドには、未構造化データが含まれていることがあります。

構造化フィールド (M+3-N) が X'00' の場合は、次の未構造化データが含まれます。

- 1 (サブフィールド・キー・フィールドを含む) 未構造化データ・フィールドの長さ。ゼロの場合には、このフィールド全体を省略することができます。
- 2 サブフィールド・キー: X'00'
- 3-N 未構造化データ。構造化サブフィールド番号が X'03' の場合には、8 バイトの USERVAR 名がサブフィールドの後に続きます。

構造化フィールド (M+3-N) が X'01' の場合は、次のセッション修飾子が含まれます。

- 1 (サブフィールド・キー・フィールドを含む) セッション修飾子フィールドの長さゼロの場合には、このフィールド全体を省略することができます。
- 2 サブフィールド・キー: X'01'
- 3 1 次リソース修飾子の長さ (X'00' は 1 次リソース修飾子が存在しないことを意味します)。0 から 8 までの値は有効です。
- 4-N 1 次リソース修飾子
- N+1 2 次リソース修飾子の長さ (X'00' は 2 次リソース修飾子が存在しないことを意味します)。0 から 8 までの値は有効です。
- N+2-M 2 次リソース修飾子
- M+1 パスワードの長さ (X'00' はパスワードが存在しないことを意味します)。0 から 8 までの値は有効です。
- M+2-P パスワード。バインド時または IMS からのバインド応答時には無視され、バインド時またはバインド応答時に送信されることはありません。並列セッションを使用するセッション開始にこれらのパラメーターが必要な場合は、IMS は X'01' のフィールド (セッション修飾子フィールド) を検出するために構造化フィールドを索引付けします。
- M+3-N 構造化フィールドに X'02' のサブフィールドがある場合には、IMS はフィールドを MSC パートナー ID と解釈します。MNPS ではなく USERVAR を使用する XRF 環境でバインドが発行された場合には、追加の構造化セグメントがユーザー・データに組み込まれます。このセグメントの形式は次のとおりです。
  - 1 USERVAR セグメントの長さ、X'09'
  - 2 サブフィールド・キー、X'03'
  - 3-10 XRF 複合システムの USERVAR を含む 8 バイトのフィールド

7. URC および 2 次 LU 名は IMS によって使用されませんが、互換性のために表示されます。

関連資料:

 z/OS: 要求単位 (RU) のフォーマット

## 2 次ハーフセッションとしての IMS

IMS より他へのセッション開始時に、IMS は指定されたバインド形式を受信することがあります。受信したバインド・パラメーターが交渉 BIND 要求を指示している場合には、バインド応答を送信する前に、IMS は指示された値で 2 次 NAU プロトコル・フィールドをオーバーライドします。

IMS では、バインドを使用して一部のパラメーターを任意に設定することができます。IMS は、これによって指示された制約範囲内で操作を行います。

表 107. IMS より他への 2 次ハーフセッション

バイト	ビット	内容	BIND FORMAT の説明 <sup>1</sup>
0		X'31'	バインド要求コード
1	0-3	B'0000'	形式 : TYPE 0
	4-7	B'0000', B'0001'	BIND タイプ 0000 - 交渉 0001 - 非交渉
2	0-7	X'12'	FM プロファイル 18
3	0-7	X'04'	TS プロファイル 4
4			基本 NAU プロトコル
	0	B'0', B'1'	B'0' 単一 RU チェーン
			B'1' 複数の RU チェーン
	1	B'01', B'10'	即時要求:
			B'01' 例外時応答チェーン
			B'10' 確定応答チェーン
	2-3	B'01', B'10', B'11'	11- 例外時 / 確定応答チェーン
	4	B'0'	作成を送信できない
	5	B'0'	予約済み
	6	B'0'	圧縮なし
7	B'0', B'1'	0- 基本がブラケットを終了することができない	
		1- 基本が終了ブラケットを送信することができる	
5			2 次 NAU プロトコル
	0	B'1'	複数の RU チェーン
	1	B'0'	即時要求
	2-3	B'10', B'11' (交渉 BIND 用に設定された B'11')	10 - 確定応答チェーン
			11- 例外時 / 確定応答チェーン
	4	B'0'	作成を送信できない
	5	B'0'	予約済み
	6	B'0'	圧縮なし
7	B'1'	0 - 2 次はブラケット終了を送信することができない	
		1 - 2 次はブラケット終了を送信することができる	
6			共通 NAU プロトコル
	0	B'0'	予約済み
	1	B'1'	機能管理ヘッダーが許可される
	2	B'0', B'1'	ブラケット状態:
			B'0' ブラケット状態マネージャーのイン・ ブラケット状態へのリセット
		B'1' ブラケット状態マネージャーのブラケ ット間状態へのリセット <sup>2</sup>	

表 107. IMS より他への 2 次ハーフセッション (続き)

バイト	ビット	内容	BIND FORMAT の説明 <sup>1</sup>
	3	B'1'	条件付きブラケット終了
	4	B'0'	代替コードなし
	5	B'0', B'1'	STSN 必須フラグ:  B'0' シーケンス番号は使用不能  B'1' シーケンス番号使用可能
	6	B'0', B'1'	BIS がフラグを送信:  B'0' BIS 未送信  B'1' BIS 送信済み
	7	B'000'	予約済み
7	0-1	B'10'	FM トランザクション・モード、HDX-FF
	2	B'1'	送信側 ERP
	3-6	B'0000'	予約済み
	7	B'0', B'1'	バイト 6 のビット 2 が 0 の場合、 0 - 2 次が、データ・トラフィックをアクティブ状態にした状態  1 - 基本が、データ・トラフィックをアクティブ状態にした後で最初に発信する <sup>2</sup>
8	0-7	未変更	2 次送信ベージング・カウント
9	0-7	未変更	2 次受信ベージング・カウント
10	0-7	定義された出力バッファ・サイズと等しいか、それより大きくなければならない	SLU 最大送信 RU サイズ
11	0-7	定義された受信サイズと等しいか、それより小さくなければならない	PLU 最大送信 RU サイズ <sup>3</sup>
12	0-7	未変更	基本送信ベージング・カウント
13	0-7	未変更	基本受信ベージング・カウント
14	0-7	X'06'	LU プロファイル (LUTYPE6)
15	0-7	X'00' 予約済み	機能管理ヘッダー・サブセット
16	0	予約済み	1 次ハーフセッション・フラグ
	1	予約済み	
	2	1 - 受信 SYMSMSG がサポートされる	
	3	0 - 受信 SCHEDULER  モデルがサポートされない <sup>4</sup>  1 - 受信 SCHEDULER	
	4	0 - 受信 QMODEL が  サポートされない  1 - 受信 QMODEL がサポートされる	
	5	0 - 線形ファイル・モデルが  サポートされない	
	6	0 - DL/I モデルが  サポートされない	
7	予約済み		
17		予約済み	
18-19		予約済み	

表 107. IMS より他への 2 次ハーフセッション (続き)

バイト	ビット	内容	BIND FORMAT の説明 <sup>1</sup>
20	0	予約済み	2 次ハーフセッション・フラグ
	1	予約済み	
	2	1 - FMH6: 受信  SYSMSG がサポートされる	
	3	1 - 受信スケジュール・  モデルがサポートされる	
	4	1 - 受信 QMODEL が  サポート	
	5	0 - 線形ファイル・モデルが  サポートされない	
	6	0 - DL/I モデルが  サポートされない	
7	予約済み		
21		予約済み	
22-26		予約済み	
27	0-7		PLU 名の長さ
28-M			PLU 名
M+1	0-7	X'00' ユーザー・データなし	ユーザー・データの長さ
M+2-N		X'00' 構造化フィールドが続く  X'00' 未構造化ユーザー・データの最初のバイト <sup>5</sup>	ユーザー・データ
M+3-N		未構造化 ユーザー・データの 残り	未構造化ユーザー・データの場合、
M+3-N		構造化フィールド <sup>6</sup>	構造化ユーザー・データの場合
N+1	0-7	ユーザー要求相関 (URC) フィールドの長さ X'00' = URC なし	URC の長さ <sup>7</sup>
N+2-P		URC: エンド・ユーザー定義 ID	URC <sup>7</sup>
P+1	0-7	X'00' = 2 次名なし	2 次 LU 名の長さ <sup>7</sup>
P+2-R	0-7	2 次 LU 名	2 次 LU 名 <sup>7</sup>

表 107. IMS より他への 2 次ハーフセッション (続き)

バイト	ビット	内容	BIND FORMAT の説明 <sup>1</sup>
-----	-----	----	------------------------------

注:

1. BIND RU の長さは 256 バイトを超えることはできません。超えた場合は否定応答が戻されます。
2. 可能なブラケット内またはプロセス再始動を指示するために設定します。応答モード出力がキューに残っている場合、あるいは IMS が会話型モードになっている場合は、バインド時に IMS が設定します。また、交渉バインド応答時にもう一方のハーフセッションが送信することもできます。
3. 受信用バッファ・サイズは、28 バイト未満の COMM マクロ・ステートメントの RECANY キーワード・パラメータ上のユーザー提供のサイズの値によって判別されます。
4. もう一方のハーフセッションが SCHEDULER プロセスをサポートしていないことがバインドによって示された場合は、IMS は ATTACH を使用してすべての非送信請求出力および非同期出力を送信します。
5. 未構造化ユーザー・データが無視され、IMS によって提供されません。
6. 構造化ユーザー・データ形式。構造化フィールドには、設計済みまたはサブシステム定義の情報が含まれていて、そのフィールドによってサブシステムがデータを通信する方法が提供されます。各構造化フィールドには、フィールド ID (サブフィールド番号) および長さが含まれています。構造化データ・フィールドには、未構造化データが含まれていることがあります。

構造化フィールド (M+3-N) が X'00' の場合は、次の未構造化データが含まれます。

- 1 (サブフィールド・キー・フィールドを含む) 未構造化データ・フィールドの長さ。ゼロの場合には、このフィールド全体を省略することができます。
- 2 サブフィールド・キー: X'00'
- 3-N 未構造化データ。構造化サブフィールド番号が X'03' の場合には、8 バイトの USERVAR 名がサブフィールドの後に続きます。

構造化フィールド (M+3-N) が X'01' の場合は、次のセッション修飾子が含まれます。

- 1 (サブフィールド・キー・フィールドを含む) セッション修飾子フィールドの長さゼロの場合には、このフィールド全体を省略することができます。
- 2 サブフィールド・キー: X'01'
- 3 1 次リソース修飾子の長さ (X'00' は 1 次リソース修飾子が存在しないことを意味します)。0 から 8 までの値は有効です。
- 4-N 1 次リソース修飾子
- N+1 2 次リソース修飾子の長さ (X'00' は 2 次リソース修飾子が存在しないことを意味します)。0 から 8 までの値は有効です。
- N+2-M 2 次リソース修飾子
- M+1 パスワードの長さ (X'00' はパスワードが存在しないことを意味します)。0 から 8 までの値は有効です。
- M+2-P パスワード (バインドまたは IMS によって受信されたバインド応答が無視され、バインドまたはバインド応答で送信されません。並列セッションを使用するセッション開始にこれらのパラメーターが必要な場合は、IMS は X'01' のフィールド (セッション修飾子フィールド) を検出するために構造化フィールドを索引付けします。

発行された BIND データが MNPS ではなく USERVAR を使用する XRF システムである場合には、次の構造化ユーザー・セグメントが組み込まれています。

USERVAR セグメントの長さ - X'09'

USERVAR セグメント標識 - X'03'

USERVAR (8 バイト) - XRF システムの USERVAR

7. URC および 2 次 LU 名は IMS によって使用されませんが、互換性のために表示されます。

関連資料:

 z/OS: 要求単位 (RU) のフォーマット



## 第 37 章 SLU 1 および SLU 2 のバインド・パラメーター

IMS は、SLU 1 および SLU 2 との接続を確立するときに、さまざまなセッション・パラメーターを指定します。これらのパラメーターは、IMS との通信時に論理装置が従わなければならない規則を定義します。

次のアウトバウンド (IMS から SLU まで) のバインド形式は、このセクションに指示してある装置タイプに適用されます。

関連資料:

569 ページの『第 38 章 CINIT ユーザー・データ・パラメーターの形式』

### SLU 1 バインド・パラメーター

以下の表で、SLU 1 バインド・パラメーターを検索できます。

表 108. SLU 1 バインド・パラメーター

バイト	ビット	内容	説明
0		X'01'	Format=0; BINDTYPE=COLD
1		X'03'	FM プロファイル 3
2		X'03'	TS プロファイル 3
3			基本 LU プロトコル
	0	B'1'	複数の RU チェーン
	1	B'0'	即時要求モード
	2-3	B'11'	チェーン応答プロトコル: 任意
	4-5	B'00'	予約済み
	6	B'0'	圧縮なし
	7	B'1'	基本が終了ブラケットを送信することができる
4			2 次 LU プロトコル
	0	B'1'	複数の RU チェーン
	1	B'0'	即時要求モード
	2-3	注 <sup>1</sup>	チェーン応答プロトコル
	4-5	B'00'	予約済み
	6	B'0'	圧縮なし
	7	B'0'	2 次は終了ブラケットを送信しない
5			共通 LU プロトコル (最初のバイト)
	0	B'0'	予約済み
	1	注 <sup>2</sup>	FM ヘッダー
	2	B'1'	ブラケットを使用できる
	3	B'1'	ブラケット終了規則 1 (条件付き)
	4	B'0'	EBCDIC (代替コードなし)
	5-7	B'000'	予約済み

表 108. SLU 1 バインド・パラメーター (続き)

バイト	ビット	内容	説明
6			共通 LU プロトコル (2 番目のバイト)
	0-1	B'10'	半二重フリップフロップ
	2	B'0'	基本 ERP 責任
	3	B'0'	2 次がファースト・スピーカーである
	4-6	B'000'	予約済み
	7	B'0'	2 次が競合勝者である
7	0-1	B'00'	予約済み
	2 から 7		SLU 送信ペーシング・カウント (VTAM によって設定) <sup>3</sup>
8	0-1	B'00'	予約済み
	2 から 7		SLU 受信ペーシング・カウント (VTAM によって設定) <sup>3</sup>
9	0-7	ユーザー定義受信から設定される	SLU から PLU への RU サイズ
10	0-7	ユーザー定義出力バッファ・サイズから設定される	PLU から SLU への RU サイズ
11	0-1	B'00'	予約済み
	2 から 7		PLU CPMGR 送信ペーシング・カウント (VTAM によって設定) <sup>3</sup>
12	0-1	B'00'	予約済み
	2 から 7		PLU CPMGR 受信ペーシング・カウント (VTAM によって設定) <sup>3</sup>
13	0-7	X'01'	LU プロファイル (LUTYPE1)
14 から 35		注を参照	バインド区域の残り

表 108. SLU 1 バインド・パラメーター (続き)

バイト	ビット	内容	説明
-----	-----	----	----

注:

1. IMS がフリップフロップ・モードを強制します。
2. 前のバインド (0 から 6 および 13 バイト) は、ログモード入力に入っている可能性のあるものをオーバーライドします。
3. VTAM リスト LU 定義上の VPACING パラメーターをコーディングするか、または VTAM DLOGMOD パラメーターを使用して LU 定義上に該当するモード・テーブル項目を指定することによって、これを実行することができます。
4. 装置が必要とする場合には、バインドの残り (バイト 14 から 35) を指定することができます。これはログモード入力から入手されます。
5. 不在操作は、以下のビットを使用して、ログモード入力に指定する必要があります。

**BYTE 18**

**ビット 0**

- |   |           |
|---|-----------|
| 0 | 在席操作を開始する |
| 1 | 不在操作を開始する |

**ビット 1**

- |   |                          |
|---|--------------------------|
| 0 | セッション時に在席操作 / 不在操作を交替しない |
| 1 | セッション時に在席操作 / 不在操作を交替する  |

ノードがマスター端末として定義された場合には、IMS が在席モードを強制します。

6. IMS ユーザーは、IMS 定義に従ってログモード入力を行う必要があります。
7. 端末装置が SCS2 透過フィールド (X'35' によって識別され、その後にはフィールド長が入っている 1 バイトが続き、さらにその後には透過フィールドが続く) を送信した場合には、バインド・イメージ・ビット BINPDSB1=BINTRNDS (オフセット 17=01) を設定しなければなりません。IMS は、X'35' および長さバイトを削除し、さらにこの透過フィールドを更新せずにそのまま編集ルーチンに渡すことによって、これらのフィールドを処理します。
8. IMS は、B'10' (確定応答)、B'01' (例外応答)、または B'11' (そのいずれかの応答) の設定値を受け入れます。B'00' の設定値が見つかった場合、IMS は、最初のコンポーネントだけが定義されたときに B'01' を設定し、複数のコンポーネントが定義されたときに B'10' を設定します。
9. IMS はこのビットをオンのままにしておきます (FM ヘッダーは許可されます)。オフの場合には、IMS は、コンポーネントが 1 つだけ定義されたときにこれをオフのままにしておき (FM ヘッダーは許可されません)、複数のコンポーネントが定義されたときにこれをオンに設定します。

## SLU 2 バインド・パラメーター

以下の表で、SLU 2 デバイスのバインド・パラメーターを検索できます。

表 109. SLU 2 バインド・パラメーター :

バイト	ビット	内容	説明
0		X'01'	Format=0; BINDTYPE=COLD
1		X'03'	FM プロファイル 3

表 109. SLU 2 バインド・パラメーター (続き):

バイト	ビット	内容	説明
2		X'03'	TS プロファイル 3
3			基本 NAU プロトコル
	0	B'1'	複数の RU チェーン
	1	B'0'	即時要求
	2-3	B'11'	応答
	4-5		予約済み
	6	B'0'	圧縮なし
	7	B'1'	基本は EB を送信することができる
4			2 次 NAU プロトコル
	0	B'1'	複数の RU チェーン
	1	B'0'	即時要求
	2-3	B'01'	例外応答
	4-5		予約済み
	6	B'0'	圧縮なし
	7	B'0'	2 次は EB を送信できない
5			共通 NAU プロトコル
	0		予約済み
	1	B'0'	メッセージ・ヘッダーは許可されない
	2	B'1'	使用されるブラケット
	3	B'1'	ブラケット終了規則 1 (条件付き)
	4	B'0'	代替コードなし
	5-7		予約済み
6			共通 NAU プロトコル
	0-1	B'10'	FM 伝送モード: HDX フリップフロップ
	2	B'0'	基本 ERP 責任
	3	B'0'	2 次ステーションがファースト・スピーカー
	4-6	xxx	予約済み
	7	B'0'	HDX-FF モードの場合には、データ・トラフィック・リセット状態の終了時に、2 次が最初に送信する
7			SLU 送信ベージング・カウント: IMS によって変更されない
8			SLU 受信ベージング・カウント: IMS によって変更されない
9			SLU MAX RU 送信サイズ: IMS システム定義から最大受信バッファ・サイズに設定
10			PLU MAX RU 送信サイズ: IMS システム定義時に指定された出力バッファ・サイズから設定する

表 109. SLU 2 バインド・パラメーター (続き):

バイト	ビット	内容	説明
11 から 12			予約済み: X'0000' を指定する必要がある
13			LU TYPE: IMS によって X'02' に設定される
14 から 18			予約済み: X'00' を指定する必要がある
19 から 20			3274/3276 装置の場合、ユーザーが指定した画面サイズ。それ以外の場合、変更されない。
21 から 22			代替画面サイズ。X'00' を指定する必要がある
23			3274/3276 NDS 装置の場合は X'7E' に設定し、非 NDS 3270 マスター端末の場合は X'02' に設定する。それ以外の場合は X'00' でなければならない
24 から 25			ユーザー指定: X'00' でなければならない
26			PLU 名の長さ
27 から 34			PLU 名: IMS ACB 名
35			ユーザー・データ長: サポートされない。X'00' を指定する必要がある

注:

1. バイト 0 から 6、9、および 10 は、IMS によって指示された値に設定され、ユーザーは変更することができません。
2. バイト 8 およびバイト 26 から 34 は VTAM によって設定されます。残りのバイトはユーザーが 0 (ゼロ) に設定する必要がありますが、この設定は必須ではありません。



## 第 38 章 CINIT ユーザー・データ・パラメーターの形式

このトピックでは、CINIT ユーザー・データ・パラメーターとその構文規則について説明します。

非 MSC VTAM 端末タイプの場合にはすべて、IMS は次の発信元からユーザー・データ・パラメーターを受信することができます。

- IMS (/OPNDST コマンド)
- IMS 自動ログオン要求
- ユーザー・ログオン (SNA INITSELF コマンドなど)
- インストール・ログオン出口ルーチン (DFSLGNX0)
- サインオン出口ルーチン (DFSSGNX0)
- 宛先作成出口ルーチン (DFSINSX0)

マイグレーションが目的である場合は、ユーザー・データ・パラメーターはオプションです。ただし、セッション開始のために端末とユーザーとの間で突き合わせを行わなければならない場合は、以下に関して、ユーザー・データ・パラメーターが必要となります。

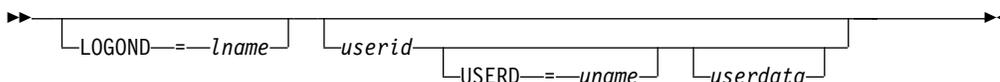
- ISC 並列セッション (LU 6.1 アーキテクチャー)
- 金融機関用端末 (3601) および SLU P 端末 (ETO と一緒に使用される場合)

オプションであれ、必須であれ、すべてのパラメーターは CINIT ユーザー・データ・フィールドに現れ、VTAM ログオン出口ルーチンが予定されているときに IMS に使用できます。

ログオンおよびサインオン・プロセス時に、IMS は、インストール・ログオン出口ルーチン (DFSLGNX0) およびサインオン出口ルーチン (DFSSGNX0) を呼び出す前に、CINIT ユーザー・データ・パラメーターで最小限のプロセスを実行します。ログオン出口ルーチンまたはサインオン出口ルーチンが提供されない場合には、IMS はデフォルトのユーザー・データ形式を取ります。

### ユーザー・データ形式

以下の構文図に、CINIT ユーザー・データの形式を示します。



これらのパラメーターでブランクは必須であり、また認識される唯一の区切り文字になっています。パラメーターを区切るために複数のブランクを使用することはできません。

## パラメーター

### **LOGOND=*lname***

端末から IMS へのログオンを試みる際に使用するログオン記述子名を指定します。*lname* は 1 から 8 バイトの長さです。LOGOND パラメーターおよび USERD パラメーターは、拡張端末操作 (ETO) にのみ有効です。

### ***userid***

IMS へログオンするユーザーのユーザー ID を 1 バイトから 8 バイトの長さで指定します。*userid* パラメーターは、この ID に関連付けられたユーザーが IMS にもサインオンすることを示します。

金融機関、SLU P、ISC、および出力専用の装置 (プリンターなど) の場合、*userid* は必須パラメーターです。

制約事項: ISC は、ISC SUBPOOL 名 (SNA PHS/SHS 修飾子) に変換されるユーザー ID だけに制限されます。ISC 並列セッションの場合、DFSLGNX0 出口ルーチンは、ISC サブプール名 (SNA PHA/SHA 修飾子) に変換されるユーザー ID だけを受信します。

### **USERD=*uname***

サインオン時にユーザー制御ブロック構造を作成するために使用するユーザー記述子名を 1 バイトから 8 バイトの長さで指定します。

### ***userdata***

サインオン出口ルーチン (DFSSGNX0) またはサインオン / オフ・セキュリティー出口ルーチン (DFSCSGN0)、および RACF などのセキュリティー製品のための追加データを指定します。出口ルーチンの場合、ユーザー・データ・フィールドの形式はインストール先で定義します。

RACF では、ユーザー・データの形式は次のとおりです。

### ***userpw***

前に入力した *userid* と関連付ける 1 バイトから 8 バイトの長さのユーザー・パスワードを識別します。ユーザー・パスワードの前にキーワードはありません。

### **GROUP *groupname***

*groupname* を、*userid* パラメーターと関連付ける 1 バイトから 8 バイトの長さのグループ名として識別します。GROUP キーワードおよび関連パラメーターはオプションです。

### **NEWPW *nuserpw***

*nuserpw* を、現行パスワード *userpw* を置き換える新規ユーザー・パスワードとして識別します。

ETO ログオン記述子名 *lname* は、IMS ログオン・プロセスに適用されます。残りのオプション・パラメーターは、提供したとしても、すべて ETO ユーザー割り振り、サインオン・プロセス、および RACF などのセキュリティー製品に渡されます。セキュリティー製品が使用された場合には、その製品に適用できないパラメーターはすべて、呼び出す前に削除しなければなりません。パラメーターは、DFSLGNX0、DFSSGNX0、および DFSCSGN0 出口ルーチンで削除できます。

関連概念:

 z/OS Communications Server SNA Programmer's LU 6.2 Guide

関連資料:

551 ページの『第 36 章 SLU P と LU 6.1 のバインド・パラメーター』

563 ページの『第 37 章 SLU 1 および SLU 2 のバインド・パラメーター』



---

## 第 39 章 SNA 文字ストリング制御

SNA 文字ストリング制御 (SCS) は、EBCDIC 制御コードの特定の機能を記述します。記述された基本機能は、印刷ページまたは英数字の表示画面をフォーマット設定するものです。

さらに、装置操作のモードを設定、固有の方法で使用されるデータを定義、または装置オペレーターとアプリケーション・プログラムの間の通信に使用されるコードのために機能が定義されます (ここで、コードに関連した特定の機能は、プログラムとオペレーターの間を設定されたプロトコルに定義されています)。

SCS データ・ストリームは、制御およびデータ文字の順次ストリングからなっています。SCS 定義の制御コードの形式の制御機能文字は、図形データ文字と混合することができます。他のデータ・タイプ (2 進数、パック 10 進数など) は、透過 (TRN) 制御と結合する場合のみ使用することができます。

SCS 制御コードは、要求単位 (RU) のデータ部分の範囲内に現れます。機能管理 (FM) ヘッダーは、RU 内部の SCS データの前に置くことができます。コンポーネント選択などの機能は FM ヘッダー機能によって実行され、SCS 機能として組み込まれません。

データ・フロー制御機能は、たとえキーボード・オペレーターがキーボード上のキーを介してそれらを使用できる場合でも、SCS 機能には組み込まれません。例えば、CANCEL は、キーボード上のキーによって開始することができるデータ・フロー制御要求です。

---

### フォーマット設定制御

フォーマット設定制御機能は、行およびページに基づいて装置の出力メディアをフォーマット設定します。これらの制御の他に、SCS を使用する装置も自動的に文字ストリングをフォーマット設定して、装置の回線の長さを調整します。

自動改行生成機能は、特定の出力形式を必要としないこれらのアプリケーションから装置行の長さ依存性を排除します。したがって、文字ストリングを再形式設定する必要なしに、各種の行の長さ機能をもつ装置に同じ文字ストリングが送信されます。

特定の行およびページ形式が必要な場合には、フォーマット設定制御機能が使用されます。自動改行機能は常にアクティブになっています。しかし、指定された行の長さに合わせてフォーマット設定された文字ストリングを、表示できる行の長さがそれよりも短い装置で表示する場合には、データを失うことなく表示することはできませんが、形式は失われます。逆に、最大表示位置に合わせて構成された文字ストリングの長さが、その文字ストリングを送られる装置の行の長さより短い場合には、その装置の最大表示位置の値を小さくすることにより、形式を失うことなくそのストリングを表示することができます。

## 制御機能コード割り当て

以下の表で、EBCDIC コードに割り当てられた SCS 制御機能を検索できます。

表 110. 制御機能コード割り当て：

EBCDIC コード	機能	機能 省略語
04	垂直チャンネル選択	VCS <sup>1</sup>
05	水平タブ	HT
0B	垂直タブ	VT
0C	用紙送り	FF
0D	改行	CR
14	表示可能	ENP
15	改行	NL
16	後退	BS
17	プログラム・オペレーター	POC <sup>1</sup>
24	表示禁止	INP
25	行送り	LF
2B	フォーマット	FMT <sup>1</sup>
34	表示位置	PP <sup>1</sup>
35	透過	TRN <sup>1</sup>

注:

1. パラメーターをとる機能

SCS のパラメーターは、次の 2 つのタイプのいずれかとなります。

- 機能パラメーター は、機能コードによって定義された機能を拡張します。例えば、PP 制御機能には、実行された位置付け機能を明示的に定義するための機能パラメーターが入っています。機能パラメーターの形式は、単一の EBCDIC 文字です。
- 値パラメーター は、機能に数値を指定します。例えば、PP 機能にもそれ用の値パラメーターが入っています。移動が現在位置に相対したものである場合には、値パラメーターは、表示位置がその現在位置から移動される列または行の数を指定します。移動が絶対的な場合には、値パラメーターは、表示位置が移動する先の絶対列番号または行番号を指定します。値パラメーターの形式は 1 バイトの 2 進数です。

---

## 第 8 部 付録



---

## 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。本書の他言語版を IBM から入手できる場合があります。ただし、ご利用にはその言語版の製品もしくは製品のコピーを所有していることが必要な場合があります。

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。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町19番21号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、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*  
*US*

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せんが、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。

記載されている性能データとお客様事例は、例として示す目的でのみ提供されています。実際の結果は特定の構成や稼働条件によって異なります。

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。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

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

著作権使用許諾:

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

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (年).

このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。

© Copyright IBM Corp. \_年を入れる\_.

---

## プログラミング・インターフェース情報

これらのトピックの情報の目的は、IMS 環境のカスタマイズを支援することです。この情報では、IMS が提供する汎用プログラミング・インターフェースと関連ガイダンス情報を記述しています。

汎用プログラミング・インターフェースにより、お客様は IMS のサービスを取得するプログラムを作成することができます。汎用プログラミング・インターフェースと関連ガイダンス情報は、それらが記載されているセクションまたはトピックの冒頭で示されているか、「汎用プログラミング・インターフェース」というラベルで示されています。

---

## 商標

IBM、IBM ロゴおよび [ibm.com](http://www.ibm.com)<sup>®</sup> は、世界の多くの国で登録された International Business Machines Corporation の商標です。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 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。

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

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。

Java<sup>™</sup> 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。

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。

### 適用される条件

このご使用条件は、IBM Web サイトのすべてのご利用条件に追加して適用されません。

## 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。

## 商業的使用

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 は、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を提供され、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・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。

この「ソフトウェア・オファリング」は、Cookie もしくはその他のテクノロジーを使用して個人情報を収集することはありません。

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 の『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/privacy/details>) を参照してください。



## 参考文献

この参考文献のリストには、IMS バージョン 13 ライブラリーのすべての資料、補足資料、資料コレクション、および IMS バージョン 13 ライブラリーで引用されているアクセシビリティ関連の資料が記載されています。

表題	頭字語	資料番号
IMS V13 アプリケーション・プログラミング	APG	SA88-5465
IMS V13 アプリケーション・プログラミング API	APR	SA88-5463
IMS V13 コマンド 第 1 巻: IMS コマンド A-M	CR1	SA88-5466
IMS V13 コマンド 第 2 巻: IMS コマンド N-V	CR2	SA88-5467
IMS V13 コマンド第 3 巻: IMS コンポーネント および z/OS コマンド	CR3	SA88-5472
IMS V13 コミュニケーションおよびコネクション	CCG	SA88-5482
IMS V13 データベース管理	DAG	SA88-5485
IMS V13 データベース・ユーティリティー	DUR	SA88-5492
IMS Version 13 Diagnosis	DGR	GC19-3654
IMS V13 出口ルーチン	ERR	SA88-5494
IMS V13 インストール	INS	GA88-5697
IMS Version 13 Licensed Program Specifications	LPS	GC19-3663
IMS V13 メッセージおよびコード 第 1 巻: DFS メッセージ	MC1	GC43-1535
IMS V13 メッセージおよびコード 第 2 巻: DFS 以外メッセージ	MC2	GC43-1536
IMS V13 メッセージおよびコード 第 3 巻: IMS 異常終了コード	MC3	GC43-1537
IMS V13 メッセージおよびコード 第 4 巻: IMS コンポーネント・コード	MC4	GC43-1538
IMS V13 オペレーションおよびオートメーション	OAG	SA88-7059
IMS V13 リリース計画	RPG	GA88-7071
IMS V13 システム管理	SAG	SA88-7072
IMS V13 システム定義	SDG	GA88-7073
IMS V13 システム・プログラミング API	SPR	SA88-7077
IMS V13 システム・ユーティリティー	SUR	SA88-7100

## 補足資料

表題	資料番号
Program Directory for Information Management System Transaction and Database Servers V13.0	GI10-8914
Program Directory for Information Management System Transaction and Database Servers V13.0 Database Value Unit Edition V13R1	GI10-8966
Program Directory for Information Management System Transaction and Database Servers V13.0 Transaction Manager Value Unit Edition V13R1	GI10-9001
IRLM メッセージおよびコード	GC19-2666

## 資料コレクション

表題	フォーマット	資料番号
IMS バージョン 13 製品キット	CD	SK5T-8864

## IMS バージョン 13 ライブラリーで引用されているアクセシビリティ ー関連の資料

表題	資料番号
z/OS TSO/E 入門	SA88-8632
z/OS TSO/E ユーザーズ・ガイド	SA88-8638
z/OS 対話式システム生産性向上機能 (ISPF) ユーザーズ・ガイド 第 1 巻	SC88-8965

# 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]

アクセシビリティ

キーボード・ショートカット x  
機能 x

アセンブル、クライアント・プログラムの 5

値の引き渡し

シンボル値の 5  
シンボルのための 5  
登録のための 5

アドレスの引き渡し

シンボルのための 5  
登録のための 5

アプリケーション・プログラミング・インターフェース

DBRC (データベース・リカバリー管理)

アクセス、RECON データ・セットへの 410

アクセス方法 404

アドレッシングおよび常駐 405

概要 403

環境の確立 404

環境の終了 404

コーディング、パラメーターの 405

コマンド要求 425

サービス、使用可能な 403

実行時の考慮事項 410

使用、レジスターの 405

使用、EQU ステートメントの 405

セキュリティ 415

トークン 406

必須前提知識 403

ワイルドカード・サポート 413

DSPAPI マクロ 403

QUERY 要求 431

READONLY アクセス 411

インターフェース

許可 5

オートノミック・コンピューティング 323

## [カ行]

開始する、プロセスを 216

環境

クライアント要求 9

CQS 登録解除要求 9

CQS 登録要求 9

CQS 要求、許可インターフェース 9

CQS 要求、非許可インターフェース 9

環境要件 117

監視プログラム状態 117

キーボード・ショートカット x

キュー

インタレストの登録 13

コールド・キュー上のオブジェクト 13

共通キュー・サーバー 3

共通サービス層 (CSL) (Common Service Layer (CSL))

クライアント 117

要求

順序、発行する 125

許可

要求 5

許可クライアント

環境要件 117

許可レベル 119

金融機関通信システム

セッション・パラメーター 551

クライアント

インターフェース

許可された 5

許可されていない 5

計画に関する考慮事項 119

コマンド処理 157

作成、ユーザー独自の 119

作成、CSL 用の 119

登録、RM クライアントの 124

ホストで実行される 157

ワークステーション 157, 158

ワークステーション SPOC 157

AOP 157

ODBM クライアントの登録 122

OM コマンド処理クライアントの登録 123

TSO SPOC 157

クライアント要求 131

概要 3

環境要件 9

許可 5

コーディング 5

サンプル 15

の順序付け 4

パラメーターのコーディング 5

クライアント要求 (続き)

プログラムのアセンブル 5

戻りコードと理由コード 10  
要求

CQSCONN 30

リスト、使用 5

リテラル、コーディング 5

リテラル、使用 5

例 15

CQSBWSE 17

CQSCHKPT 27

CQSCONN 30

CQSDEL 38

CQSDEREG 43

CQSDISC 44

CQSINFRM 49

CQSMOVE 55

CQSPUT 59

CQSQUERY 67

CQSREAD 77

CQSRECVR 84

CQSREG 89

CQSRSYNC 92

CQSSHUT 100

CQSUNLCK 102

CQSUPD 108

DSECT、使用 10

ECB、使用 5

クライアント・プログラム

アセンブル 5

作成 3

クリーンアップ・プロセス 209

グローバル・リソース

管理、ユーザーの 119

グローバル・リソース情報

保守 208

マクロ 208

計画に関する考慮事項 119

コーディング要求 5

更新、リソースの 241

構造障害ディレクティブ 248

構文図

読み方 viii

コマンド

タイプ 2 309

ADDRESS 307

CSLULGTP 309, 310

IMS DB への送信 399

REXX サブコマンド 307

CART 307

END 317

コマンド (続き)

REXX サブコマンド (続き)

IMS 307

ROUTE 307

WAIT 307

コマンド応答

フォーマット ID の組み込み 310

コマンド応答ディレクティブ 202

コマンド応答要求 194

コマンド出力ブロックのマッピング 428

コマンド処理クライアント 159

コマンド処理クライアントの要求 183

コマンド登録解除要求 186

コマンドのオーバーライド 183

コマンド要求

概要 425

コマンド/応答トークン 307

コマンド・ディレクティブ 202

コマンド・ヘッダー

XML 出力 297

## [サ行]

再移植ディレクティブ、構造 248

削除、リソースの 210

作成、CQS クライアントの 3

作動可能状態 124

自動化操作プログラム要求 131

自動化操作プログラム・クライアント 157

終了させる、プロセスを 228

障害、Resource Manager に関する 209

照会応答ディレクティブ 202

照会リソース 231

商標 577, 579

ジョブ終了ログ・レコード 329, 342

シンボル値、使用 5

シンボル値と等価のもの引き渡し 5

シンボル名、使用 5

ステム変数 309, 310

静止要求 270

前提知識 vii

## [タ行]

タイプ-2 IMS コマンド 309

タイム・スタンプ

フォーマット、DBRC 要求の 412

チェックポイント (checkpoint)

クライアントの開始 13

調整、全 IMSplex プロセスの 209

データベース・リカバリー管理 (DBRC)

アプリケーション・プログラミング・

インターフェース

マクロ・バージョン 406

データベース・リソース・アダプター

(DRA)

使用可能化

CCTL 365

初期設定

CCTL 365

データベース・リソース・アダプター

(DRA) (database resource adapter

(DRA)) 345

データ・エレメント・ヘッダー 337

データ・キャプチャー、非同期サポート

325

データ・キャプチャー・ログ・レコード

335

接頭部 335

データ・セット

IMS.ADFSMAC 5

ディレクティブ

OM 202

RM 247

構造障害 248

構造を再移植する 248

プロセス・ステップ 249

プロセス・ステップ応答 251

統計の照会 125

登録、クライアントの 237

登録解除、クライアントの 215

登録解除要求

ODBM (Open Database Manager) ク

ライアント要求 163, 164

Open Database Manager

(ODBM) 163, 164

登録済み状態 124

登録要求

ODBM (Open Database Manager) ク

ライアント要求 178

Open Database Manager

(ODBM) 178

特殊イベントの処理 13

特記事項

商標 577, 579

特記事項 577

取り込みデータ

量の削減 329

取り込みデータ・エレメント 327

トレース

DRA (データベース・リソース・アダ

プター) 398

## [ハ行]

バインド

パラメーター 558

金融機関通信システム 551

ISC、1 次ハーフセッションとして

の IMS 553

バインド (続き)

パラメーター (続き)

ISC、2 次ハーフセッションとして  
の IMS 553

LU 6.1 551

SLU 1 563

SLU 2 565

SLU P 551

バッファ戻し要求 255

パフォーマンスの チューニング 13

パラメーター

コーディング、DSPAPI マクロでの

405

割り振り出力 284

DBRC アプリケーション・プログラミ

ング・インターフェース 405

OPTWORD1 5

非許可クライアント

環境要件 117

非送信請求出力要求 187

非同期データ・キャプチャー 325

ジョブ終了呼び出しログ・レコード

331

変更データ・ログ・レコード 327

非同期伝搬、取り込みデータの

IMS サポート 325

フォーマット ID

コマンド応答への組み込み 310

複写、DL/I 更新の 325

プログラム

CSLULXCB 307

プログラムのアSEMBル 5

プロセスに回答する 219

プロセス・ステップ応答ディレクティブ

251

プロセス・ステップ・ディレクティブ 249

プロトコル

メッセージ (message) 119

要求 119

変更データ・ログ・レコード 325

形式 335

編集オプション

MFS-SCS1 573

## [マ行]

マクロ

CSLOREGO 191

DSPAPI

アクセス 410

概要 403

形式 407

バージョン 406

メッセージ

TYPE で経路指定 157, 158

メッセージ (message)  
 CQS0033A 30  
 メッセージ・プロトコル 119  
 文字ストリング制御 573  
 戻りコード  
 CSLRMDEL 210  
 CSLRMPRI 216  
 CSLRMPRR 219  
 CSLRMPRS 221  
 CSLRMPRT 228, 231  
 CSLRMQRY 231  
 CSLRMREG 237  
 CSLRMUPD 241  
 CSLSCBFR 255  
 CSLSCDRG 257  
 CSLSCMSG 259  
 CSLSCQRY 266  
 CSLSCQSC 270  
 CSLSCRDY 271  
 CSLSCREG 273  
 CSLSCRQR 280  
 CSLSCRQS 284  
 CSLZQRY 125  
 CSLZSHUT 127  
 戻りコードと理由コード  
 クライアント要求 10  
 CQSBWSE 要求 17  
 CQSCHKPT 要求 27  
 CQSCONN 要求 30  
 CQSDEL 要求 38  
 CQSDEREG 要求 43  
 CQSDISC 要求 44  
 CQSINFRM 要求 49  
 CQSMOVE 要求 55  
 CQSPUT 要求 59  
 CQSQUERY 要求 67  
 CQSREAD 要求 77  
 CQSRECVR 要求 84  
 CQSREG 要求 89  
 CQSRSYNC 要求 92  
 CQSSHUT 要求 100  
 CQSUNLCK 要求 102  
 CQSUPD 要求 108  
 問題プログラム状態 117

## [ヤ行]

要求  
 環境要件 9, 117  
 許可 5  
 計画に関する考慮事項 119  
 順序 157  
 順序、発行する 125  
 シンボル名、使用 5  
 プロトコル 119  
 リテラル、コーディング 5

要求 (続き)  
 AOP クライアントの場合のシーケンス 158  
 CQSUPD 108  
 CSLZQRY  
 構文 125  
 説明 125  
 パラメーター 125  
 CSLZSHUT 127  
 構文 127  
 説明 127  
 パラメーター 127  
 DBDS 照会  
 グループ 461  
 DBRC AUTH 要求 418  
 戻りコード 420  
 DBRC 開始要求  
 概要 515  
 構文 515  
 パラメーター 515  
 戻りコード 515  
 DBRC コマンド 425  
 DBRC 照会 431  
 グループ 467, 492  
 サブシステム 504  
 データベース 437  
 バックアウト (backout) 432  
 ログ 477  
 OLDS 487  
 RECON の状況 499  
 DBRC 停止要求  
 概要 521  
 構文 521  
 パラメーター 521  
 DBRC バッファ解放 417, 418, 511, 525  
 構文 511, 525  
 戻りコード 511, 525, 528  
 ODBM (Open Database Manager) クライアント要求  
 クライアント登録 178  
 クライアント登録解除 163, 164  
 CSLDMDRG 163  
 CSLDMI 164  
 CSLDMREG 178  
 Open Database Manager (ODBM)  
 クライアント登録 178  
 クライアント登録解除 163, 164  
 CSLDMDRG 163  
 CSLDMI 164  
 CSLDMREG 178  
 Operations Manager  
 コマンド応答 194  
 コマンド登録 191  
 コマンド登録解除 186  
 非送信請求出力 187

要求 (続き)  
 Operations Manager (続き)  
 CSLOMCMD 131  
 CSLOMI 140  
 CSLOMQRY 152  
 CSLOMREG 191  
 Resource Manager  
 削除、リソースの 210  
 順序、発行する 207  
 照会リソース 231  
 CSLRMDRG 215  
 CSLRMPRI 216  
 CSLRMPRS 221  
 CSLRMPRT 228  
 CSLRMQRY 231  
 Structured Call Interface  
 照会 266  
 登録解除 257  
 バッファ戻し 255  
 メッセージ送信 259  
 CSLSCQSC 270  
 CSLSCRDY 271  
 CSLSCREG 273  
 CSLSCRQR 280  
 CSLSCRQS 284  
 要求のシーケンス 4  
 要求プロトコル 119

## [ラ行]

リスト、クライアント要求での使用 5  
 リソース・マネージャー (RM)  
 障害 209  
 リテラル  
 使用 5  
 リポジトリ・サーバー  
 バッチ・インターフェース  
 (FRPBATCH) 535  
 FRPBATCH コマンド 535  
 ADD 537  
 DELETE 540  
 DSCHANGE 540  
 LIST 541  
 RENAME 542  
 START 543  
 STOP 544  
 UPDATE 545  
 理由コード  
 CSLRMDEL 210  
 CSLRMPRI 216  
 CSLRMPRR 219  
 CSLRMPRS 221  
 CSLRMPRT 228, 231  
 CSLRMQRY 231  
 CSLRMREG 237  
 CSLRMUPD 241

理由コード (続き)

CSLSCBFR 255  
CSLSCDRG 257  
CSLSCMSG 259  
CSLSCQRY 266  
CSLSCQSC 270  
CSLSCRDY 271  
CSLSCREG 273  
CSLSCRQR 280  
CSLSCRQS 284  
CSLZQRY 125  
CSLZSHUT 127

例

値の引き渡し

シンボル値の 5  
シンボルのための 5  
登録のための 5

アドレスの引き渡し  
シンボルのための 5  
登録のための 5

シンボル値と等価のもの引き渡し 5

CQSREAD 要求 15

OPTWORD1 を使用した CQSREAD  
のコーディング 5

REXX SPOC API  
オートノミック 323

STEPLIB DD statement to  
concatenate IMS.SDFSRESL 5

レジスター

クライアント要求 5  
使用 5

レディー要求 271

ロギング

DRA (データベース・リソース・アダ  
プター) 398

## [ワ行]

ワークステーション SPOC 157

## [数字]

2 次ハーフセッション 558

## A

ADD バッチ・コマンド 537

AOP クライアント 157  
ホストで実行される 158

AOP コマンド 159

AUTH 照会

概要 417

AUTH 要求

構文 418  
出力ブロック 423, 424

AUTH 要求 (続き)

パラメーター 418, 525  
戻りコード 420  
理由コード 422

## B

BACKOUT 照会

構文 432  
出力 432  
説明 432  
パラメーター 432  
戻りコード 432

## C

CAPD

ブロックのフォーマット 338  
DATA フォーマット 341

CART 307

CCTL 機能要求

INIT 373

CCTL (コーディネーター・コントロー  
ー)

設計の指針 393  
パフォーマンスに関する考慮事項  
スレッドのモニター 395  
マルチスレッドの例 349

CINIT ユーザー・データ・パラメーター  
569

COMMAND 要求

戻りコード 427

CQS (Common Queue Server) 3

クライアント 15

CQS のシャットダウン 13

CQSBrowse 要求

機能 17  
構文 17  
使用法 17  
パラメーター 17  
戻りコードと理由コード 17

BROWSE 機能 17

BRWSOBS 機能 17

COMPLETE 機能 17

CONTINUE 機能 17

DSECT 機能 17

CQSCHKPT 要求

形式 27  
構文 27  
使用法 27  
パラメーター 27  
戻りコードと理由コード 27

CHKPTSTR 機能 27

CHKPTSYS 機能 27

DSECT 機能 27

CQSCONN 要求

形式 30  
構文 30  
使用法 30  
制約事項 30  
パラメーター 30  
戻りコードと理由コード 30  
CONNECT 機能 30

DSECT 機能 30

CQSDEL 要求

形式 38  
構文 38  
使用法 38  
パラメーター 38  
戻りコードと理由コード 38

DELETE 機能 38

DSECT 機能 38

CQSDEREG 要求

形式 43  
構文 43  
使用法 43  
パラメーター 43  
戻りコードと理由コード 43

DEREGISTER 機能 43

DSECT 機能 43

CQSDISC 要求

形式 44  
構文 44  
使用法 44  
パラメーター 44  
戻りコードと理由コード 44

DISCABND 機能 44

DISCNORM 機能 44

DSECT 機能 44

CQSINFRM 要求

形式 49  
構文 49  
パラメーター 49  
戻りコードと理由コード 49  
DSECT 機能 49

INFORM 機能 49

UNINFORM 機能 49

CQSMOVE 要求

形式 55  
構文 55  
使用法 55  
パラメーター 55  
戻りコードと理由コード 55

DSECT 機能 55

MOVE 機能 55

CQSPUT 要求

アクション 59  
形式 59  
構文 59  
使用法 59  
パラメーター 59

- CQSPUT 要求 (続き)
  - 戻りコードと理由コード 59
  - ABORT 機能 59
  - DSECT 機能 59
  - PUT 機能 59
- CQSQUERY 要求
  - 形式 67
  - 構文 67
  - 使用法 67
  - パラメーター 67
  - 戻りコードと理由コード 67
  - DSECT 機能 67
  - QNAME 機能 67
  - QRYOBS 機能 67
  - QTYPE 機能 67
  - STATISTICS 機能 67
  - STRSTAT 機能 67
- CQSREAD 要求
  - 機能 77
  - 形式 77
  - 構文 77
  - 使用法 77
  - パラメーター 77
  - 戻りコードと理由コード 77
  - 例 15
  - CONTINUE 機能 77
  - DSECT 機能 77
  - READ 機能 77
  - REREAD 機能 77
- CQSRECVR 要求
  - 機能 84
  - 形式 84
  - 構文 84
  - 使用法 84
  - パラメーター 84
  - 戻りコードと理由コード 84
  - DELETE 機能 84
  - DSECT 機能 84
  - RETRIEVE 機能 84
  - UNLOCK 機能 84
- CQSREG 要求
  - 機能 89
  - 構文 89
  - 使用法 89
  - パラメーター 89
  - 戻りコードと理由コード 89
  - DSECT 機能 89
  - REGISTER 機能 89
- CQSRSYNC 要求
  - 機能 92
  - 形式 92
  - 構文 92
  - 使用法 92
  - パラメーター 92
  - 戻りコードと理由コード 92
  - DSECT 機能 92
- CQSRSYNC 要求 (続き)
  - RSYNCCOLD 機能 92
  - RSYNCWARM 機能 92
- CQSSHUT 要求
  - 機能 100
  - 形式 100
  - 構文 100
  - 使用法 100
  - パラメーター 100
  - 戻りコードと理由コード 100
  - DSECT 機能 100
  - QUIESCE 機能 100
- CQSUNLCK 要求
  - 機能 102
  - 形式 102
  - 構文 102
  - パラメーター 102
  - 戻りコードと理由コード 102
  - DSECT 機能 102
  - FORCE 機能 102
  - UNLOCK 機能 102
- CQSUPD 要求
  - 機能 108
  - 形式 108
  - 構文 108
  - 使用法 108
  - パラメーター 108
  - 戻りコードと理由コード 108
  - DSECT 機能 108
  - UPDATE 機能 108
- csl
  - 戻りおよび理由 119
- CSL (Common Service Layer) 117
  - クライアント 117
  - 要求 183
    - 順序、発行する 125
    - RM クライアントの作成 207
    - SCI クライアントの作成 253
- CSL OM API
  - XML 出力 297
- CSL マネージャー
  - SCI への登録 121
- csl 要求
  - コード 119
- CSLDMDRG 163
- CSLDMI 164
- CSLDMREG 178
  - 構文 178
  - パラメーター 178
  - 戻りコードと理由コード 181
- CSLOMBLD 183
- CSLOMBLD コマンドのオーバーライド 183
- CSLDMCMD 131
- CSLDMCMD 出力 294
- CSLOMI
  - 応答ディレクティブ 202
  - 出力 291
  - 入力バッファ、例 140
- CSLOMOUT 187
- CSLOMOUT 出力 297
- CSLOMQRY 152
- CSLOMQRY 出力 295
- CSLOMRDY 要求 189
- CSLOMRSP 194
- CSLOMSUB 197
- CSLOMUSB 201
- CSLOREGO 191
- CSLRMDEL 210
- CSLRMDRG 215
- CSLRMPRI 216
- CSLRMPRR 219
- CSLRMPRS 221
- CSLRMQRY 231
- CSLRMREG 237
- CSLRMUPD 241
- CSLSCBFR 255
- CSLSCDRG 257
  - 環境要件 117
- CSLSCMSG 259
- CSLSCQRY 266
- CSLSCQSC 270
- CSLSCREG 273
  - 環境要件 117
  - 制約事項 273
- CSLSCRQR 280
- CSLSCRQS 284
- CSLULGTP 310
- CSLULGTS 309
- CSLULGUM 要求 319
- CSLULOPT 310
- CSLULSUB 要求 318
- CSLULUSB 要求 318
- CSLULXCB 307
- CSLZQRY 要求 125
  - 構文 125
  - 説明 125
  - パラメーター 125
- CSLZSHUT 要求 125
  - 構文 127
  - 説明 127
  - パラメーター 127

## D

- DB 照会
  - 概要 437
  - 構文 437
  - 出力 437
  - パラメーター 437
  - 戻りコード 437

DB2 への DL/I 更新の伝搬 325	DBRC (データベース・リカバリー管理)	DBRC (データベース・リカバリー管理)
DBDS 照会	(続き)	(続き)
概要 461	DBDS 照会 (続き)	UNAUTH 照会 (続き)
構文 461	戻りコード 461	構文 525
パラメーター 461	GROUP 照会	戻りコード 525, 528
戻りコード 461	概要 467	DELETE バッチ・コマンド 540
DBRC (データベース・リカバリー管理)	構文 467	DFSPRP マクロ・キーワード 361
403	出力 467	DFSPSP00 (DRA 始動テーブル) 361
アプリケーション・プログラミング・	パラメーター 467	DRA (データベース・リソース・アダプ
インターフェース	戻りコード 467	ター)
アクセス、RECON データ・セッ	LOG 照会	開始テーブル
トへの 410	概要 477	説明 361
アクセス方法 404	構文 477	DFSPZPxx 361
アドレッシングおよび常駐 405	出力 477	終了 391
概要 403	パラメーター 477	使用可能化
環境の確立 404	戻りコード 477	CCTL 365
環境の終了 404	OLDS 照会	ODBA 367
コーディング、パラメーターの 405	概要 487	初期設定
コマンド要求 425	構文 487	CCTL 365
サービス、使用可能な 403	出力 487	ODBA 367
実行時の考慮事項 410	パラメーター 487	処理
使用、レジスターの 405	戻りコード 487	CCTL 要求 369
使用、EQU ステートメントの 405	PART 照会	ODBA 呼び出し 371
セキュリティ 415	概要 492	スレッド
トークン 406	構文 492	構造 347
マクロ・バージョン 406	パラメーター 492	処理 347
ワイルドカード・サポート 413	戻りコード 492	ODBA 347
DSPAPI マクロ 403	QUERY 要求	スレッド機能要求
QUERY 要求 431	BACKOUT 432	ABTTERM 387
READONLY アクセス 411	RECON 状況照会	COMTERM 386
データ・セット	概要 499	IMS 382
出力データ・セットの使用 412	構文 499	PREP 385
RECON への READONLY アクセ	出力 499	SCHED 379
ス 411	パラメーター 499	SYNTERM 383
要求タイム・スタンプのフォーマット	戻りコード 499	TERMTHRD 388
412	RELBUF 照会	スレッド統計 396
AUTH 照会	概要 511	説明 345
概要 417	構文 511	同期点処理
構文 418, 525	戻りコード 511	説明 355
戻りコード 420	STARTDBRC 要求	同期点の処理
BACKOUT 照会	概要 515	プロトコル 356
出力 432	構文 515	未確定状態 359
戻りコード 432	パラメーター 515	トレース 398
COMMAND 要求 425	戻りコード 515	マクロ・キーワード 361
パラメーター 425	STOPDBRC 要求	マルチスレッド化 348
戻りコード 427	概要 521	問題判別 399
DB 照会	構文 521	CCTL 機能要求
概要 437	パラメーター 521	説明 373
構文 437	SUBSYS 照会	INIT 373
出力 437	概要 504	RESYNC 377
パラメーター 437	構文 504	TERM 378
戻りコード 437	出力 504	CCTL リカバリー処理 393
DBDS 照会	パラメーター 504	DRA 統計 397
概要 461	戻りコード 504	DSCHANGE バッチ・コマンド 540
構文 461	UNAUTH 照会	DSECT
パラメーター 461	概要 525	DSPAPCMD 428

## DSECT (続き)

DSPAPQAL 437  
DSPAPQAR 437  
DSPAPQCG 467  
DSPAPQDB 437  
DSPAPQDG 467  
DSPAPQDS 437  
DSPAPQEL 437  
DSPAPQFD 437  
DSPAPQHB 437  
DSPAPQHP 437  
DSPAPQIC 437  
DSPAPQLA 477  
DSPAPQLG 477  
DSPAPQLI 477  
DSPAPQOL 487  
DSPAPQRC 499  
DSPAPQRI 437  
DSPAPQRR 437  
DSPAPQRV 437  
DSPAPQSL 437  
DSPAPQSS 504

## DSECTS

DSPAPQCA 467  
DSPAPQGG 467

## DSPAPI

アクセス 410  
形式

実行 407  
標準 407  
変更 407  
リスト 407

バージョン 406

## DSPAPI マクロ

概要 403

## E

### ECB 117

ECB (z/OS イベント制御ブロック)、クライアント要求での使用 5

## F

### FID

コマンド応答への組み込み 310

### FRPBATCH コマンド 535

ADD 537  
DELETE 540  
DSCHANGE 540  
LIST 541  
RENAME 542  
START 543  
STOP 544  
UPDATE 545

## G

### GROUP 照会

概要 467  
構文 467  
出力 467  
パラメーター 467  
戻りコード 467

## I

### IMSpIex

統計の照会 125  
マクロを使ったプロセスの調整 209  
REXX SPOC API の準備 307

### IMSSPOC 環境 317

### IMS.ADFSMAC データ・セット 5

## L

### LIST バッチ・コマンド 541

### LOG 照会

概要 477  
構文 477  
出力 477  
パラメーター 477  
戻りコード 477

## O

### ODBM (Open Database Manager)

クライアント要求 163  
登録、クライアントの 122  
ODBM クライアントの作成 161  
ODBM 要求、順序 161

### ODBM (Open Database Manager) クライアント要求

要求  
クライアント登録 178  
クライアント登録解除 163, 164  
CSLDMDRG 163  
CSLDMI 164  
CSLDMREG 178

### ODBM クライアント

作成、CSL 用の 119

### OLDS 照会

概要 487  
構文 487  
出力 487  
パラメーター 487  
戻りコード 487

### OM

クライアント 157  
ディレクティブ 202

### OM (Operations Manager)

登録、クライアントの 123  
AOP クライアント 158  
AOP クライアントから発行される要求 158  
参照: Operations Manager (OM)

### OM クライアント

作成、CSL 用の 119

### OM ディレクティブ

および SCI 入力出口ルーチン 202  
コマンド 202  
コマンド応答 202  
照会応答 202  
CSLOMI 応答 202  
UOM 202

### Open Database Manager (ODBM)

クライアント要求 163  
登録、クライアントの 122  
要求  
クライアント登録 178  
クライアント登録解除 163, 164  
CSLDMDRG 163  
CSLDMI 164  
CSLDMREG 178  
ODBM クライアントの作成 161  
ODBM 要求、順序 161

### Operations Manager

要求  
コマンド応答 194  
コマンド登録解除 186  
CSLOMCMD 131  
CSLOMQRY 152

### Operations Manager (OM)

クライアント要求 131  
出力 160  
出力の解釈 160  
登録、クライアントの 123  
要求  
非送信請求出力 187  
CSLOMI 140  
CSLOMREG 191  
AOP クライアント 158  
AOP クライアントから発行される要求 158  
XML 出力 160, 291

### OPTWORD1 パラメーター 5

## P

### PART 照会

概要 492  
構文 492  
パラメーター 492  
戻りコード 492

## Q

### QUERY 要求

- 概要 431
  - SUBSYS 504
- 構文
  - BACKOUT 432
  - DB 437
  - DBDS 461
  - GROUP 467
  - LOG 477
  - OLDS 487
  - PART 492
  - RECON の状況 499
  - SUBSYS 504

### 出力 432

- BACKOUT 432
- DB 437
- GROUP 467
- LOG 477
- OLDS 487
- RECON の状況 499
- SUBSYS 504

### 出力の一般的なフォーマット 409

#### タイプ

- BACKOUT 432
- DB 437
- DBDS 461
- GROUP 467
- LOG 477
- OLDS 487
- PART 492
- RECON の状況 499

#### パラメーター

- BACKOUT 432
- DB 437
- DBDS 461
- GROUP 467
- LOG 477
- OLDS 487
- PART 492
- RECON の状況 499
- SUBSYS 504

#### 戻りコード

- BACKOUT 432
- DB 437
- DBDS 461
- GROUP 467
- LOG 477
- OLDS 487
- PART 492
- RECON の状況 499
- SUBSYS 504

## R

### RECON 状況照会

- 概要 499
- 構文 499
- 出力 499
  - パラメーター 499
  - 戻りコード 499

### RECON データ・セット

- アクセス、DSPAPI による 410

### RELBUF 照会

- 概要 511

### RELBUF 要求

- 構文 511
  - パラメーター 511
  - 戻りコード 511

### RENAME バッチ・コマンド 542

#### Resource Manager

- クリーンアップ・プロセス 209
- 調整、全 IMSplex プロセスの 209
- マスター 209
- 要求

- 順序、発行する 207
- 登録、クライアントの 237
- CSLRMPRR 219
- CSLRMPRS 221
- CSLRMPRT 228
- CSLRMUPD 241

#### Resource Manager (RM)

- 登録、クライアントの 124
- 登録解除、クライアントの 215
- 要求
  - 更新、リソースの 241
  - 削除、リソースの 210
  - 終了させる、プロセスを 228
  - プロセス応答 219
  - プロセス・ステップ 221
  - 保守、グローバル・リソース情報の 208
  - CSLRMDRG 215
  - CSLRMPRI 216
  - CSLRMQRY 231
  - CSLRMREG 237

### REXX SPOC API 157, 307

- オートノミック・コンピューティング 323
- 環境を準備する 307
- コマンド応答の検索 309
- サブコマンド 307
- サンプル 320
- サンプル・プログラム 319
- トランザクション内 317
- バッチ・ジョブ 321
- 非送信請求メッセージの検索 318
- 例 320
- IMSplex の設定 307

### REXX SPOC プログラム、サンプル 320

#### RM (Resource Manager)

- 登録、クライアントの 124
- RM クライアント
  - 作成、CSL 用の 119

## S

### SCI (Structured Call Interface)

- 環境要件 117
- 作動可能状態 124
- 出口ルーチン
  - 使用するかどうか 119
- 登録済み状態 124
- への登録 121
- 要求、拡張 254
- 要求のシーケンス 253
- CSL マネージャーの登録 121
- TCB アソシエーション 119

### SCS (SNA 文字ストリング) 制御

- 機能コード割り当て 574
- フォーマット設定制御 573

### SETS および ROLS 呼び出しログ・レコード 331, 333, 342

#### SLU 1

- バインド・パラメーター 563

#### SLU 2

- バインド・パラメーター 565

#### SLU P

- セッション・パラメーター 551

### SNA (システム・ネットワーク体系) 文字ストリング (SCS) 制御 573

#### SNA の参照情報 551

#### START バッチ・コマンド 543

#### STARTDBRC 要求

- 概要 515
- 構文 515
- パラメーター 515
- 戻りコード 515

#### STEPLIB DD statement to concatenate

#### IMS.SDFSRESL 5

#### STOP バッチ・コマンド 544

#### STOPDBRC 要求

- 概要 521
- 構文 521
- パラメーター 521

#### Structured Call Interface

- 要求
  - 拡張 254
  - メッセージ送信 259
  - 要求送信 284
  - レディー要求 271
  - CSLSCDRG 257
  - CSLSMSG 259
  - CSLSCQSC 270
  - CSLSCREG 273

Structured Call Interface (続き)

要求 (続き)

CSLSCRQR 280

要求のシーケンス 253

Structured Call Interface (SCI)

環境要件 117

作動可能状態 124

出口ルーチン

使用するかどうか 119

登録済み状態 124

への登録 121

要求 255

照会 266

登録 273

登録解除 257

バッファ戻し 255

CSLSCRDY 271

割り振り出力パラメーター 284

CSL マネージャーの登録 121

TCB アソシエーション 119

SUBSYS 照会

概要 504

構文 504

出力 504

パラメーター 504

戻りコード 504

SWITCH コマンド 399

## X

XML 出力 291

および OM ディレクティブ 202

コマンド・ヘッダー 297

タグの説明 297

CSLQMCMD 294

CSLQMOU 297

CSLQMQRY 295

## [特殊文字]

/DISPLAY 323

## T

TCB アソシエーション 119

Tivoli NetView 環境 307

TSO

CSLULXCB プログラムの始動 307

TSO SPOC 157

## U

UNAUTH 照会

概要 525

UNAUTH 要求

構文 525

出力ブロック 531, 532

戻りコード 525, 528

理由コード 530

UOM ディレクティブ 202

UPDATE バッチ・コマンド 545

## V

VTAM の参照情報 551







プログラム番号: 5635-A04  
5655-DSM  
5655-TM2

Printed in Japan

SA88-7077-04



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

Spine information:

IMS バージョン 13

システム・プログラミング API

