CICS Transaction Server for z/OS

リソース解説書

バージョン *5* リリース *5*

CICS Transaction Server for z/OS

リソース解説書

バージョン *5* リリース *5*

注記

本書および本書で紹介する製品をご使用になる前に、 519 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、IBM CICS Transaction Server for z/OS バージョン 5 リリース 5 (製品番号 5655-Y04) および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

原典： CICS Transaction Server for z/OS
Resource Reference
Version 5 Release 5

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 1982, 2018.

目次

この PDF について vii

第 1 章 リソース定義属性 1

ATOMSERVICE リソース定義 1
ATOMSERVICE 属性 1
BUNDLE リソース 4
BUNDLE 属性. 5
CONNECTION リソース 6
CONNECTION 属性 11
DB2CONN リソース 23
DB2CONN 属性. 23
DB2ENTRY リソース 36
DB2ENTRY 属性 36
DB2TRAN リソース 42
DB2TRAN 属性 42
DOCTEMPLATE リソース 44
DOCTEMPLATE 属性. 44
ENQMODEL リソース 48
ENQMODEL 属性 50
FEPI リソース 51
FILE リソース 51
リモート・ファイル 53
カップリング・ファシリティー・データ・テーブル 54
共用データ・テーブル. 55
FILE 属性 55
IPCONN リソース 77
IPCONN 属性 77
JOURNALMODEL リソース 88
JOURNALMODEL 属性 89
デフォルトの JOURNALMODEL 95
例 95
JVMSERVER リソース 96
JVMSERVER 属性 97
LIBRARY リソース. 99
LIBRARY 属性 100
LSRPOOL リソース 105
LSRPOOL 属性 106
MAPSET リソース 111
MAPSET 属性 112
MQCONN リソース 114
MQCONN 属性 115
MQMONITOR リソース 118
MQMONITOR 属性 121
PARTITIONSET リソース 124
PARTITIONSET 属性 125
PARTNER リソース 127
PARTNER 属性 127
PIPELINE リソース 130
PIPELINE の属性 131
PROCESSTYPE リソース 134
PROCESSTYPE 属性 135
PROFILE リソース 137
PROFILE 属性 138
PROGRAM リソース. 145
PROGRAM 属性 146
SESSIONS リソース 159
SESSIONS 属性 160
TCPIPSERVICE リソース 172
TCPIPSERVICE の属性 173
TCPIPSERVICE: 相互関連属性 186
TDQUEUE リソース 188
一時データの二重目的リソース定義 188
TDQUEUE 属性 189
必須の TDQUEUE 定義. 204
ダミー一時データ・キューの使用. 209
TERMINAL ターミナル. 210
印刷用の端末 211
パイプライン端末定義 213
LDC リスト対応の装置 215
APPC (LUTYPE6.2) 単一セッション端末 . . . 215
トランザクション・ルーティング用の端末. . . 215
TERMINAL 属性 221
TRANCLASS リソース 234
TRANCLASS 属性 234
TRANSACTION リソース 236
TRANSACTION 属性 238
TSMODEL リソース 257
TSMODEL 属性 258
TYPETERM リソース 263
TYPETERM属性のデフォルト値 265
サポートされる装置 272
TYPETERM 属性 277
URIMAP リソース 307
URIMAP: 関連リソース. 311
URIMAP の属性 312
WEBSERVICE リソース. 329
WEBSERVICE 属性 331

第 2 章 リソース管理トランザクション CEDA コマンド. 335

CEDA ADD コマンド 335
CEDA ALTER コマンド 336
CEDA APPEND コマンド 339
CEDA CHECK コマンド 340
CEDA COPY コマンド 341
CEDA DEFINE コマンド 344
CEDA DELETE コマンド 346
CEDA DISPLAY コマンド 348
CEDA DISPLAY GROUP コマンド. 350
CEDA DISPLAY LIST コマンド 350
CEDA EXPAND コマンド 351

© Copyright IBM Corp. 1982, 2018

CEDA EXPAND GROUP コマンド 352
CEDA EXPAND LIST コマンド 353
CEDA INSTALL コマンド 353
CEDA LOCK コマンド 356
CEDA MOVE コマンド 357
CEDA REMOVE コマンド 360
CEDA RENAME コマンド 361
CEDA UNLOCK コマンド 363
CEDA USERDEFINE コマンド 365
CEDA VIEW コマンド 368

第 3 章 リソース管理ユーティリティー DFHCSDUP コマンド 371
DFHCSDUP ADD コマンド 371
DFHCSDUP ALTER コマンド 372
DFHCSDUP APPEND コマンド 374
DFHCSDUP COPY コマンド 375
DFHCSDUP DEFINE コマンド 378
DFHCSDUP DELETE コマンド 380
DFHCSDUP EXTRACT コマンド 382
DFHCSDUP INITIALIZE コマンド 384
DFHCSDUP LIST コマンド 385
DFHCSDUP PROCESS コマンド 387
DFHCSDUP REMOVE コマンド 388
DFHCSDUP SCAN コマンド 388
DFHCSDUP SERVICE コマンド 391
DFHCSDUP UPGRADE コマンド 391
DFHCSDUP USERDEFINE コマンド 392
DFHCSDUP VERIFY コマンド 394

第 4 章 DFHCSDUP: 構文規則およびコマンド規則 397

第 5 章 リソース定義マクロ 399
CLT: コマンド・リスト・テーブル 399
制御セクション: DFHCLT TYPE=INITIAL . . 399
代替システムの指定: DFHCLT TYPE=LISTSTART 401
テークオーバー・コマンドの指定: DFHCLT TYPE=COMMAND 401
オペレーターへのメッセージ: DFHCLT TYPE=WTO 402
コマンド・リストのクローズ: DFHCLT TYPE=LISTEND 403
PDIR: DL/I ディレクトリー 403
制御セクション: DFHDLPSB TYPE=INITIAL 403
プログラム仕様ブロック: DFHDLPSB TYPE=ENTRY 403
FCT: ファイル管理テーブル 404
制御セクション: DFHFCT TYPE=INITIAL . . 405
ローカル・ファイル — DFHFCT TYPE=FILE 405
DFHFCT の例 414
MCT: モニター管理テーブル 414
制御セクション: DFHMCT TYPE=INITIAL . . 415
ユーザー・イベント・モニター・ポイント: DFHMCT TYPE=EMP 420
制御データの記録: DFHMCT TYPE=RECORD 425
DFHMCT の例 438
PLT: プログラム・リスト・テーブル 439
制御セクション: DFHPLT TYPE=INITIAL . . 440
プログラム・リスト・テーブル内のエントリー: DFHPLT TYPE=ENTRY 441
DFHPLT の例 443
RST: リカバリー可能サービス・テーブル 444
制御セクション: DFHRST TYPE=INITIAL . . 445
リカバリー可能サービス・エレメント: DFHRST TYPE=RSE 445
DBCTL サブシステム: DFHRST TYPE=SUBSYS 445
DFHRST の例 446
SRT: システム・リカバリー・テーブル 446
制御セクション: DFHSRT TYPE=INITIAL . . 447
異常終了コード: DFHSRT TYPE=SYSTEM|USER 447
DFHSRT の例 448
TCT: 端末管理テーブル 449
DFHTCT マクロ・タイプ 450
DFHTCT 論理装置コード: z/OS Communications Server 非 3270 453
順次装置 462
トランザクション・ルーティング用のリモート端末 468
DFHTCT: CICS 端末リスト 471
TLT: 端末リスト・テーブル 472
制御セクション: DFHTLT TYPE=INITIAL . . 473
端末リスト・テーブル内のエントリー: DFHTLT TYPE=ENTRY 474
DFHTLT の例 475
TST: 一時記憶域テーブル 475
制御セクション: DFHTST TYPE=INITIAL . . 477
リカバリー可能一時記憶域: DFHTST TYPE=RECOVERY 478
ローカル一時記憶域: DFHTST TYPE=LOCAL 480
リモート一時記憶域: DFHTST TYPE=REMOTE 481
一時記憶域セキュリティー検査: DFHTST TYPE=SECURITY 482
一時記憶域データ共用: DFHTST TYPE=SHARED 483
DFHTST の例 484
XLT: トランザクション・リスト・テーブル . . . 485
制御セクション: DFHXLT TYPE=INITIAL . . 486
トランザクション・リスト・テーブル内のエントリー: DFHXLT TYPE=ENTRY 486
DFHXLT の例 488

第 6 章 互換性のために保持されている廃止された属性 489

第 7 章 CICS 提供のリソースの定義、グループ、およびリスト 491
DFHLIST 定義 491
DFHLIST にない CICS 提供グループ 499
CICS 提供の互換性グループ 500

サンプル・アプリケーション・プログラム・グループ 500
グループ DFHTYPE 内の TYPETERM 定義 . . . 505
グループ DFHTERM 内のモデル TERMINAL 定義 511
グループ DFHEP 内の PROFILE 定義 513
グループ DFHISC 内の PROFILE 定義 514
グループ DFHSTAND 内の PROFILE 定義 . . . 515
グループ DFHPGAIP 内のモデル定義 517

特記事項 519

索引 525

この PDF について

この PDF は、CICS リソースの属性の解説書です。これらのリソースを定義する方法を確認するには、*CICS TS for z/OS* の構成という PDF を参照してください。CICS TS V5.4 以前は、この PDF の情報は *Resource Definition Guide* にありました。

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

この PDF の日付

この PDF は、2018 年 12 月 14 日 に作成されました。

© Copyright IBM Corp. 1982, 2018

第 1 章 リソース定義属性

ほとんどのリソースは、CICS RDO または CICSPlex SM ビジネス・アプリケーション・サービス (BAS) を使用して定義することができます。

ATOMSERVICE リソース定義

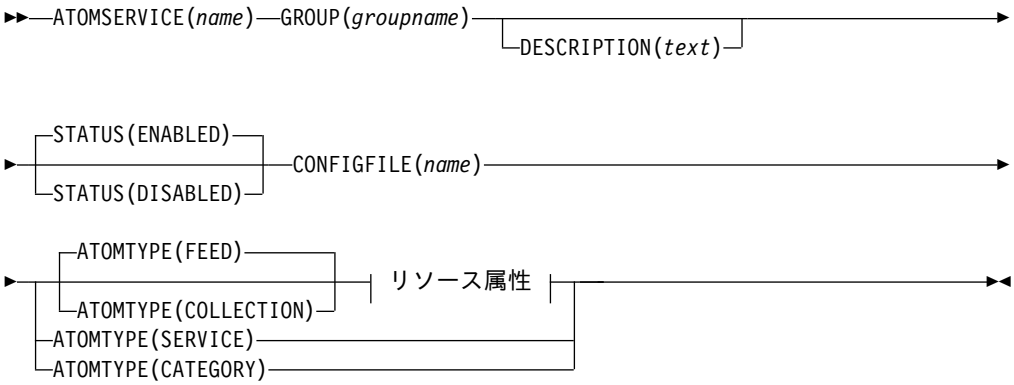
ATOMSERVICE リソースは、Atom サービス、フィード、コレクション、またはカテゴリー文書を定義し、フィードのデータを提供するために使用されるリソースを識別します。

CEMT および CEDA を使用した ATOMSERVICE リソース定義のインストールについては、ATOMSERVICE リソース定義のインストールを参照してください。BAS の作業については、BAS ATOMSERVICE リソース定義の作業を参照してください。

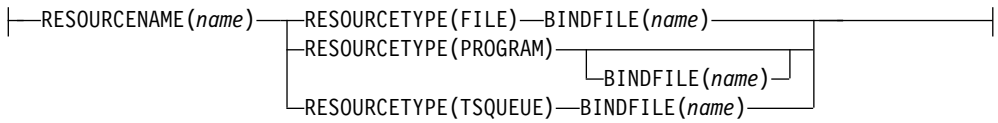
ATOMSERVICE 属性

ATOMSERVICE リソースの構文と属性について記述します。

構文



リソース属性:



属性

ATOMSERVICE(*name*)

このリソース定義の 8 文字の名前を指定します。

© Copyright IBM Corp. 1982, 2018

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

ATOMTYPE({**FEED**|**SERVICE**|**COLLECTION**|**CATEGORY**})

この ATOMSERVICE 定義に対して戻される、Atom 文書のタイプを指定します。

CATEGORY

Atom カテゴリー文書。コレクション内の項目のカテゴリーをリスト表示します。同じカテゴリーを使用して複数のコレクションを定義する場合は、カテゴリー文書をセットアップすることができます。

COLLECTION

Web クライアントによる編集が可能な Atom エントリーのグループを収めた Atom コレクション文書。Web クライアントは、HTTP POST、PUT、および DELETE 要求を使用して編集を行い、HTTP GET 要求を使用して取得を行います。コレクションの Atom 構成ファイルは、ルート・エレメント <cics:atomservice type="collection"> で始まらなければなりません。

FEED フィードのためのメタデータを記述し、フィードにデータを提供する Atom エントリーを収めた Atom フィード文書。Atom フィードは HTTP GET 要求を使用して取得できますが、Web クライアントによって編集することはできません。 Atom フィードの Atom 構成ファイルは、ルート・エレメント <cics:atomservice type="feed"> で始まらなければなりません。

SERVICE

Atom サービス文書。サーバーで使用できる編集可能コレクションに関する情報を提供します。

BINDFILE(*name*)

z/OS® UNIX システム・サービスに保管される XML バインディングの完全修飾 (絶対) 名または相対名を指定します。この属性は、Atom サービスまたはカテゴリー文書には使用されません。 XML バインディングは、CICS® XML アシスタント・プログラム DFHLS2SC を使用して作成します。

リソース・タイプ FILE および TSQUEUE の場合には XML バインディングが必要で、このファイルによって、Atom 文書にデータを提供する RESOURCENAME という名前のリソースが使用するデータ構造が指定されます。

リソース・タイプ PROGRAM の場合、XML バインディングはオプションであり、プログラム自体ではなく、Atom エントリーのデータを取得するためにプログラムがアクセスするリソースを使用して作成します。 DFHATOMPARMS コンテナー内のリソース処理パラメーターを使用して Atom 構成ファイルの情報をプログラムに渡している場合には、リソース・タイプ PROGRAM の XML バインディングを指定しなければなりません。このようにしていない場合には、XML バインディングは指定しないでください。

XML バインディングの名前は絶対パスとして指定できます。絶対パスには、すべてのディレクトリーが含まれ、スラッシュで始まります (例: /u/atom/atomictest.xsdbind)。別の方法としては、CICS 領域のユーザー ID の HOME ディレクトリーに対する相対パスとして指定できます。例えば、先頭にスラッシュを付けることなく、atom/atomictest.xsdbind などとします。最大で 255 文字を入力できます。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

CONFIGFILE(*name*)

z/OS UNIX システム・サービスに保管される Atom 構成ファイルの完全修飾 (絶対) 名または相対名を指定します。Atom 構成ファイルには、このリソース定義に返される Atom 文書のメタデータとフィールド名を指定する XML が含まれます。詳しくは、Atom 構成ファイルの作成を参照してください。

この名前は絶対パスとして指定できます。絶対パスには、すべてのディレクトリーが含まれ、スラッシュで始まります (例: /u/atom/myfeed.xml)。別の方法としては、CICS 領域のユーザー ID の HOME ディレクトリーに対する相対パスとして指定できます。例えば、先頭にスラッシュを付けることなく、atom/myfeed.xml などとします。最大 255 文字を使用できます。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

DESCRIPTION(*text*)

このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)

各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字:
A-Z 0-9 $ @ #
小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

RESOURCENAME(*name*)

この Atom フィードまたはコレクションにデータを提供する CICS リソースの名前 (1 から 16 文字) を指定します。サービス・ルーチンによってアクセスされるリソースに Atom エントリーのデータが保持されている場合、ここでサービス・ルーチンの名前を指定します。この属性は、Atom サービスまたはカテゴリー文書には使用されません。

許容文字: A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

RESOURCETYPE({FILE|PROGRAM|TSQUEUE})
この Atom フィードまたはコレクションのデータを保持する CICS リソースのタイプを指定します。この属性は、Atom サービスまたはカテゴリー文書には使用されません。この属性のデフォルトは TSQUEUE です。

FILE CICS ファイル。ファイル内の単一のレコードが、単一の Atom エントリーのデータを提供します。 Atom エントリーを保持するファイルはレコードの固有キーを持つ必要があり、NONUNIQUEKEY 属性によって定義された代替索引ファイルを使用することはできません。 Atom エントリーを保持する VSAM ファイルは種類を問いませんが、ESDS (入力順データ・セット) ファイルは、編集可能コレクションとしてセットアップするフィードには適していないことにご注意ください。ESDS では、レコードの削除が行えないからです。 BDAM ファイルを使用できません。

PROGRAM
Atom エントリーにコンテンツを提供するために作成された CICS アプリケーション・プログラムであるサービス・ルーチン。

TSQUEUE
一時記憶キュー。 一時記憶域キュー内の単一のレコードが、単一の Atom エントリーのデータを提供します。

STATUS({ENABLED|DISABLED})
このリソース定義によって指定された Atom 文書が使用可能か、使用不可かを示します。

BUNDLE リソース

BUNDLE リソースは、アプリケーションのデプロイメント単位の CICS バンドルを定義します。バンドルは、CICS リソース、成果物、参照、およびマニフェストの集合であり、CICS 領域にデプロイすることで、アプリケーション全体あるいはアプリケーションのコンポーネントを表すことができます。

マニフェストは、バンドルの内容を記述するファイルのことで、CICS 領域およびサポートしている成果物の場所に作成するリソース、アプリケーションを正常に実行するために必要な前提条件、およびアプリケーションが他のアプリケーションに提供できるサービスが含まれています。

バンドルは、z/OS UNIX にデプロイされ、成果物のディレクトリー構造を構成します。 BUNDLE リソースは、z/OS UNIX におけるバンドルのデプロイ場所とその状況を定義します。BUNDLE リソースを使用可能にすると、CICS はマニフェストを読み取り、ユーザー用にマニフェストで定義されているアプリケーション・リソースを動的に作成します。 BUNDLE リソースを正常に使用可能にするには、アプリケーションの前提条件として定義されている他のリソースが CICS 領域になければなりません。

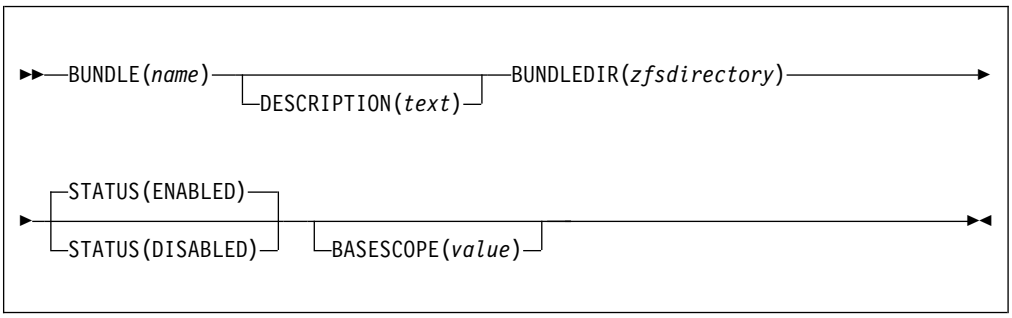
CICS バンドルを使用する場合、インストールされたリソースのライフサイクルは、バンドルのライフサイクルに従います。CICS バンドルを選択可能/選択不可にしたり、使用可能/使用不可にしたり、または破棄したりすると、その CICS バンドルによって動的に作成されたすべてのリソースに対して同じアクションが適用されます。 CICS バンドルでリソースをパッケージ化することの影響については、Characteristics of bundled resourcesを参照してください。

注: CICS Explorer® を使用してバンドルを作成しデプロイした場合は、バンドルのソース・コードが適切に管理されるよう確認する必要があります。zFS 内のエクスポート・データからバンドルを再構成することはできません。また、ワークステーションで障害が発生すると、データが失われる可能性があります。 CICS Explorer のエクスポート機能を使用してバンドルをエクスポートし、それをソース・コード管理システムにチェックインすることができます。

BUNDLE リソース定義のインストールについては、Defining CICS bundlesを参照してください。BAS の作業については、BAS BUNDLE リソース定義の作業を参照してください。

BUNDLE 属性

BUNDLE リソースの構文と属性について記述します。



BASESCOPE(*value*)

バンドルのスコープを定義する 1 文字から 255 文字のストリングを指定します。可能な場合には、URI (Uniform Resource Identifier) を使用してください。 BASESCOPE 属性は、類似するいくつかのバンドルを一緒にグループ化したい場合や、プラットフォームで稼働する特定のアプリケーションに CICS バンドルを関連付けたい場合に使用します。アプリケーションの URI は次のような形式です。

`cicsapplication://Platform/ApplicationID/MajorVersion/MinorVersion/MicroVersion`

Platform はアプリケーションが稼働しているプラットフォームの名前、*ApplicationID* はアプリケーション・バンドルの ID、その後にアプリケーションのバージョン (メジャー、マイナー、マイクロ) が続きます。

デフォルト値は空で、すべての BUNDLE リソースが同じスコープにインストールされます。 詳細については、バンドルのスコープを参照してください。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

BUNDLE(*name*)

BUNDLE の名前を 1 から 8 文字で指定します。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

文字 DFH は、CICS が使用するために予約されているため、DFH で始まる名前は使用しないことが推奨されます。

CICS は、それらを削除または破棄したり、新たに機能や修正をアップグレードあるいは提供する際に CICS リソースに置き換えたりすることができます。

BUNDLEDIR(*zfsdirectory*)

z/OS UNIX 上のバンドルについて、1 から 255 文字のルート・ディレクトリーの完全修飾名を指定します。先頭文字と末尾文字は / にする必要があります。

指定される値は UNIX ファイルに有効な名前でなければなりません。 • 組み込みスペース文字を含めることはできません。 • / 文字の連続インスタンスが含まれていてはなりません。 • 大/小文字の区別があります。 許容文字: A-Z a-z 0-9 . / _ # @ -

DESCRIPTION(*text*)

このフィールドには、定義するリソースの記述を提供することができます。記述テキストは、58 文字以内の長さにすることができます。使用できる文字に制限はありません。しかし、括弧を使用する場合は、左右の括弧が対になるようにしてください。CREATE コマンドを使用する場合、テキスト・コードに含まれる各アポストロフィに 2 つのアポストロフィを使用してください。

STATUS(**ENABLED**|**DISABLED**)

BUNDLE リソースがインストールされるときの初期状況を指定します。

ENABLED

BUNDLE は使用可能です。CICS は、領域にバンドルのすべての前提条件が存在することを検査してから、バンドルで定義されているすべての必須リソースをインストールして使用可能にします。

DISABLED

BUNDLE は使用できません。CICS は、前提条件とインストール用の検査を行わず、バンドルで定義されているリソースを使用可能にしません。

CONNECTION リソース

CONNECTION では、システム間連絡 (ISC) または 複数領域操作 (MRO) を使用して CICS システムと通信するリモート・システムを定義します。

ISC は APPC または LUTYPE6.1 通信プロトコルを使用します。 MRO は IRC、XM、または XCF/MRO アクセス方式を使用します。

また、IPCONN リソースも参照してください。CONNECTION と同様、IPCONN もリモート・システムへの通信リンクを定義しますが、この場合、接続には TCP/IP プロトコルが使用されます。

CONNECTION を定義する場合、システムを識別するのに十分な情報を提供し、その基本的な属性を指定する必要があります。 SESSIONS 定義には、システムとの通信に使用するセッションに関する詳細を指定します。定義がインストールされている場合、CICS は、CONNECTION 名を使用して他方のシステムを識別します。MRO を介して接続されている他の CICS システムでは、通常、この名前は他方の CICS システムで SYSIDNT システム初期設定パラメーターとして指定された名前と同じです。 ISC を介して接続される他のシステムにとっては、この名前は通常、システムのロケーションまたはシステムを所有する組織を記述する頭字語に基づいています (例えば USA1 または IBMC)。

TRANSACTION 定義上または TERMINAL 定義上の REMOTESYSTEM 名では、その CONNECTION 名によって CONNECTION 定義を参照する (あるいは、その IPCONN 名によって IPCONN 定義を参照する) ことができます。これらの属性はトランザクション・ルーティングに使用されます。

PROGRAM 定義上の REMOTESYSTEM 名では、その CONNECTION 名によって CONNECTION 定義を参照する (あるいは、その IPCONN 名によって IPCONN 定義を参照する) ことができます。この属性は、分散プログラム・リンクに使用されます。

CONNECTION 定義では、関連付けられた SESSIONS の名前を指定しません。

相互通信リソースの定義の作成を開始する前に、相互通信リソースの定義を参照して詳しいガイダンスを確認してください。そこには、さまざまなタイプのリンクおよびセッションで指定する必要のある属性の役立つ例が数多く記載されています。

異なる接続タイプについての特別な考慮事項を以下に示します。

MRO リンクおよびセッション

1 つの CONNECTION 定義を使って MRO リンクを定義し、1 つの SESSIONS 定義を使ってそれに関連付けられる並列セッションを定義します。

ACCESSMETHOD

CONNECTION 定義で、これを IRC (領域間通信の場合) または XM (クロスメモリー・サービスの場合) として指定します。 IRC は、リンクのオープンとクローズに使用します。

PROTOCOL

SESSIONS 定義では、PROTOCOL として LU61 を指定します。CONNECTION 定義では、PROTOCOL 値をブランクのままにします。

SENDPFX、SENDCOUNT、RECEIVEPFX、RECEIVECOUNT

1 つの SESSIONS 定義で、複数の送信セッションおよび複数の受信セッションを指定します。これらの属性に指定した値は、定義のインスト

ール時に作成される TCT エントリーの名前を判別するために使用されます。 (CONNECTION リソース定義のインストールを参照してください。)

APPC リンクおよび並列セッション

APPC では、セッションはモードセットにグループ分けされます。各モードセットは SESSIONS 定義を使用して定義するため、必要なモードセットと同じ数の SESSIONS 定義が必要です。 CONNECTION 定義としてリンクを定義します。以下の属性が重要です。

ACCESSMETHOD
CONNECTION 定義で、これを VTAM® として指定します。

MAXIMUM
これを使用して、モードセット内のセッション数を制御します。

MODENAME
各モードセットの SESSIONS 定義で、MODENAME を使用してモードセットの名前を指定します。アクティブ・システムに定義がインストールされると、CICS はこの名前によってモードセットを認識します。

PROTOCOL
CONNECTION 定義と SESSIONS 定義の両方で、プロトコルとして APPC を指定します。

APPC (LUTYPE6.2) 単一セッション端末

APPC 端末を CONNECTION-SESSIONS ペアまたは TERMINAL-TYPETERM ペアとして定義することができます。 TERMINAL-TYPETERM 方式は、 215 ページの『APPC (LUTYPE6.2) 単一セッション端末』で説明されています。 CONNECTION-SESSIONS 方式を使用する場合は、以下の属性が重要です。

ACCESSMETHOD
CONNECTION 定義で、これを VTAM として指定します。

MAXIMUM
単一セッション端末の場合、1,0 を指定しても 1,1 を指定しても効果は同じです。 (詳しくは、 11 ページの『CONNECTION 属性』を参照。)

MODENAME
SESSIONS 定義で、MODENAME を指定します。アクティブ・システムに定義がインストールされると、CICS はこの名前を使用してセッションを識別します。

PROTOCOL
CONNECTION 定義と SESSIONS 定義の両方で、プロトコルとして APPC を指定します。

SINGLESESS
YES は、CONNECTION 定義が単一セッション端末用であることを示します。

LUTYPE6.1 リンクおよびセッション

LUTYPE6.1 リンクおよびセッションは、2 つの方法のいずれかで定義できます。

- 1 つの CONNECTION 定義と 1 つの SESSIONS 定義
- 1 つの CONNECTION 定義といくつかの SESSIONS 定義 (必要なセッションごとに 1 つ)

すべてのセッションが全く同じ属性を持つ場合には、 各リンクを 1 つの CONNECTION 定義に定義し、それに関連付けられるすべてのセッションを 1 つの SESSIONS 定義に定義します。

ACCESSMETHOD
CONNECTION 定義で、これを VTAM として指定します。

PROTOCOL
SESSIONS 定義と CONNECTION 定義で、これを LU61 として指定します。

RECEIVECOUNT、RECEIVEPFX、SENDCOUNT、SENDPFX
これらの属性は、MRO リンクおよびセッションの場合と同様に使用されます。

セッションがそれぞれ異なる属性を持つ場合には、セッションごとに別個の SESSIONS 定義を作成する必要があります。 NETNAMEQ を除いて、この方法は、以下に示す CICS-IMS™ セッション用の方法を同じです。

注**:** CICS-CICS ISC リンクおよびセッションの場合、LUTYPE6.1 ではなく APPC を使用することが推奨されます。

LUTYPE6.1 CICS-IMS リンクおよびセッション
IMS では、各セッションが異なる NETNAMEQ を持つ必要があるため、各セッションを別個の SESSIONS 定義に定義する必要があります。

リンクを CONNECTION 定義として定義し、複数の SESSIONS 定義 (SEND セッションごとに 1 つ、RECEIVE セッションごとに 1 つ) を作成します。

ACCESSMETHOD
CONNECTION 定義で、これを VTAM として指定します。

NETNAMEQ
リモート IMS システムは、この名前を使用してセッションを識別します。

PROTOCOL
CONNECTION 定義と SESSIONS 定義の両方で、プロトコルとして LU61 を指定します。

SESSNAME
アクティブ・システムに定義がインストールされると、CICS はこの名前を使用してセッションを識別します。

RECEIVECOUNT
SENDCOUNT
これらの属性を使用して、セッションが SEND セッションか、RECEIVE セッションかを指定します。

RECEIVE セッションとは、ローカル CICS が 1 次で、競合敗者であるセッションのことをいいます。 これは、RECEIVECOUNT(1) を定義

し、SENDCOUNT をデフォルトのブランクのままにすることで指定されます。 (SENDPFX や RECEIVEPFX を指定する必要はありません。)

SEND セッションとは、ローカル CICS が 2 次で、競合勝者であるセッションのことをいいます。 それは SENDCOUNT(1) を定義し、RECEIVECOUNT をデフォルトのブランクのままにすることによって指定します。

INDIRECT 接続

INDIRECT 接続は、ローカル・システムとの直接リンクを定義していないリモート・システムです。代わりに、2 つのシステムは 1 つ以上の中間システムを使用して相互に通信します。この方式をトランザクション・ルーティングに使用することができます。間接的に接続されたリモート・システムは、常に端末専有領域です。ローカル・システムは、常にアプリケーション専有領域であるか、トランザクション・ルーティング・パス上の中間領域です。

間接接続が必要になるのは、中間システムを介したトランザクション・ルーティングに非 z/OS Communications Server 端末を使用する場合のみです。オプションとして、z/OS Communications Server 端末で間接接続を使用することもできます。その場合、端末専有領域への優先パスを識別するための、いくつかのトランザクション・ルーティング・パスが可能です。間接接続を定義することが推奨される理由について、およびトランザクション・ルーティングに必要なリソースについては、トランザクション・ルーティングのための間接リンクの定義を参照してください。

ローカル・システムには、直接接続先となる中間システム用に、通常の CONNECTION 定義および SESSIONS 定義が必要です。 ACCESSMETHOD は、PROTOCOL(LU61) を指定した IRC または XM、あるいは PROTOCOL(APPC) を指定した VTAM にする必要があります。

INDIRECT 接続 (間接リンクまたは間接システムとも呼ばれる) の場合、ローカルシステムには CONNECTION 定義のみが必要です。 SESSIONS 定義は必要ありません。使用されるセッションは、中間システムのセッションです。 CONNECTION 定義では以下の属性が重要です。

ACCESSMETHOD
これを INDIRECT として指定します。

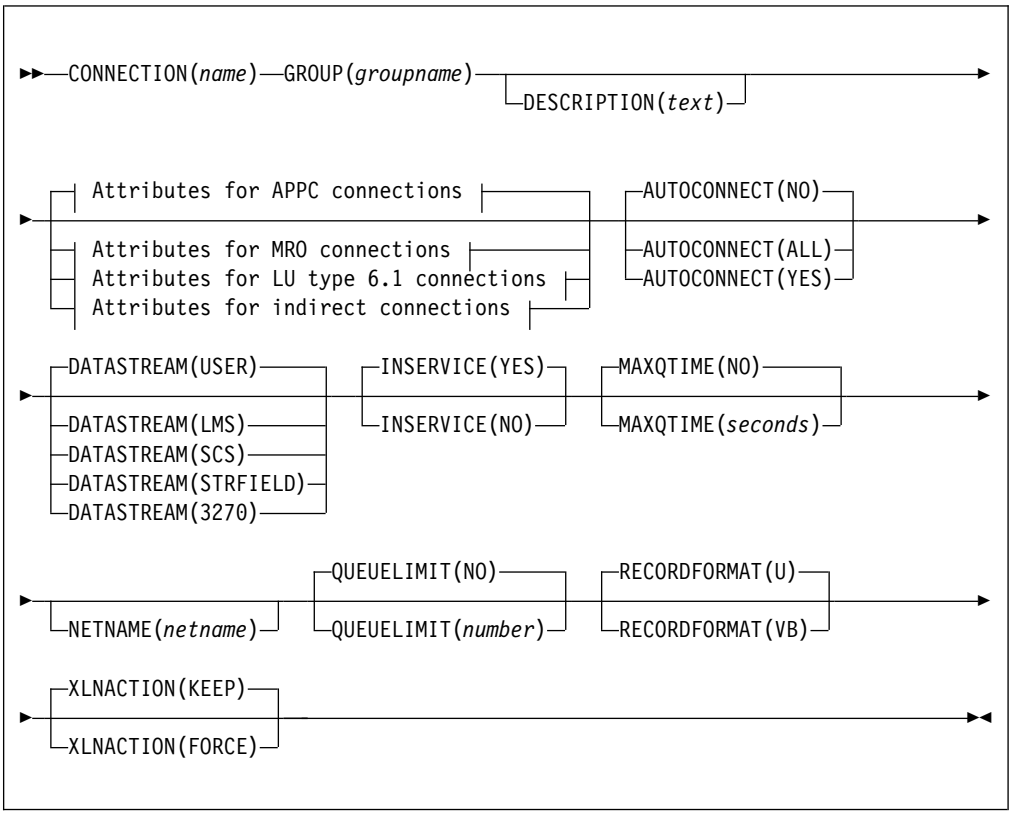
INDSYS
端末専有システムへのパスの起点となる MRO または APPC リンクについて、CONNECTION 定義を指定します。

NETNAME
端末専有システムの APPLID を指定します。

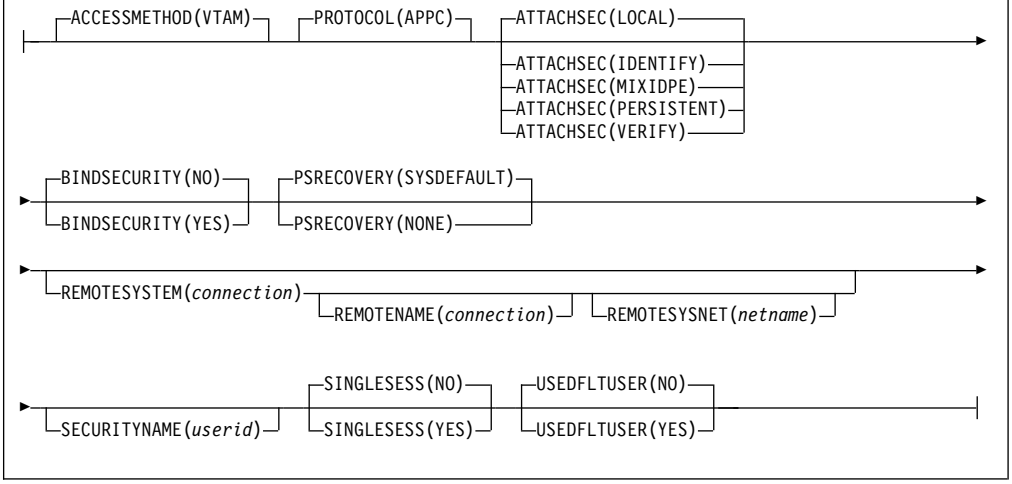
CEMT および CEDA を使用した CONNECTION リソース定義のインストールについては、CONNECTION リソース定義のインストールを参照してください。BAS の作業については、BAS ISC/MRO 接続リソース定義の作業を参照してください。

CONNECTION 属性

CONNECTION リソースの構文と属性について説明します。

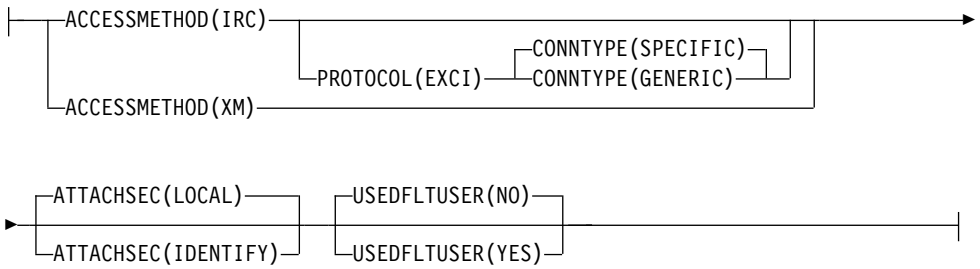


Attributes for APPC connections:

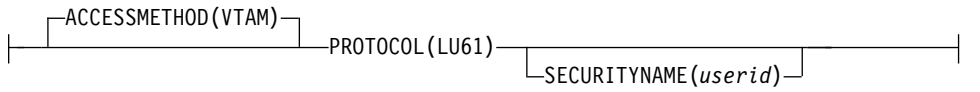


注: VTAM は現在 z/OS Communications Server になっています。

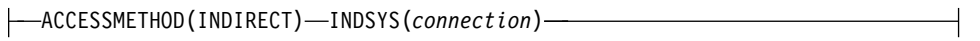
Attributes for MRO connections:



Attributes for LU type 6.1 connections:



Attributes for indirect connections:



ACCESSMETHOD({VTAM|INDIRECT|IRC|XM})
この接続で使用するアクセス方式を指定します。

VTAM
ローカル CICS 領域とこの接続定義で定義されたシステムの間の通信は、z/OS Communications Server を介して行われます。異なる MVS™ イメージ内、あるいは同じ MVS イメージの異なるアドレス・スペース内にあるシステムには、z/OS Communications Server のシステム間通信 (ISC) を使用することができます。

INDIRECT
ローカル CICS システムとこの接続定義で定義されたシステムの間の通信は、INDSYS オペランドで指定されたシステムを介して行われます。

IRC ローカル CICS 領域とこの接続定義で定義された領域の間の通信は、領域間通信 (IRC) プログラム DFHIRP を介して、DFHIRP の SVC (クロスメモリー (XM) の反対) モードを使用して行われます。

注: ここでは、IRC という用語を一般的に使われる意味よりも限定的な意味で使用しています。
IRC は、同じ MVS イメージにある領域や、シスプレックス内の異なる MVS イメージにある領域での複数領域操作 (MRO) に使用できます。

XM ローカル CICS 領域と CONNECTION 定義で定義された領域の間の MRO 通信には、MVS クロスメモリー・サービスを使用します。初期接続は、領域間通信 (IRC) プログラム DFHIRP を介して、DFHIRP のクロスメモリー (XM) (SVC の反対) モードを使用して行われます。XM は、同じ MVS イメージ内の領域、あるいはシスプレックス内の異なる MVS イメージ内の領域に対する複数領域操作に使用することができます。

注**:** DFHIRP はリンクが開かれているときに使用されるため、XM には引き続き CICS タイプ 3 SVC が必要です。 SVC について詳しくは、『インストール』の『CICS SVC のインストール』を参照してください。

MVS クロスメモリー・サービスは、リンク先の ACCESSMETHOD も XM として定義されている場合にのみ使用されます。

MRO パートナーがシスプレックス内の別の MVS イメージにあり、CONNECTION で IRC または XM が指定されている場合、CICS は自動的に XCF をアクセス方式として使用し、IRC や XM の指定を無視します。

注**:** XCF を明示的に定義することはできません。XCF を使用する場合は、IRC または XM を指定する必要があります。 XCF について詳しくは、システム間複数領域操作 (XCF/MRO) を参照してください。

ATTACHSEC({**LOCAL**|**IDENTIFY**|**VERIFY**|**PERSISTENT**| **MIXIDPE**})

接続に必要な接続時ユーザー・セキュリティーのレベルを指定します。

IDENTIFY

着信接続要求には、ユーザー ID を指定する必要があります。接続するシステムにセキュリティー・マネージャーがある場合 (例えば、別の CICS システムに接続する場合) には、IDENTIFY を入力します。

LOCAL

CICS では、クライアントがユーザー ID やパスワードを提供する必要はありません。すべての要求が SECURITYNAME 属性で指定されたユーザー ID で実行されます。 CONNECTION 定義の PROTOCOL 属性が LU6.1 である場合、LOCAL を指定する必要があります。

MIXIDPE

着信接続要求は IDENTIFY と PERSISTENT のいずれかまたは両方のセキュリティー・タイプを使用できます。使用されるセキュリティー・タイプは、着信接続要求によって異なります。

PERSISTENT

着信接続要求の最初の接続要求でユーザー ID およびユーザー・パスワードを指定する必要があります。後続の接続要求では、ユーザー ID のみが必要になります。これは、プログラマブル・ワークステーション (例えば、IBM® パーソナル・コンピューター) と CICS の間でのみ使用する必要があります。

VERIFY

着信接続要求には、ユーザー ID とユーザー・パスワードを指定する必要があります。接続システムにセキュリティー・マネージャーがなく、

信頼できない場合に、VERIFY を入力します。CICS はパスワードを送信しないため、CICS-CICS 間通信では VERIFY は指定しないでください。

AUTOCONNECT({NO|YES|ALL})

ACCESSMETHOD(VTAM) を使用するシステムの場合、AUTOCONNECT (YES) または (ALL) を一緒に指定します。これにより、セッションが確立されます (つまり BIND が実行されます)。これらのセッションは、CICS を初期化する際、あるいは CEMT または EXEC CICS SET VTAM OPEN コマンドを使用して z/OS Communications Server との通信を開始する際にセットアップされます。リモート・システムが使用できないためにこのタイミングで接続を確立できない場合は、リモート・システムが一定期間中に使用可能になって自動的に通信を開始しない限りは、後で CEMT または EXEC CICS SET CONNECTION(sysid) INSERVICE ACQUIRED コマンドを使用して、リンクを獲得する必要があります。

SINGLESESS(NO) が指定されている APPC 接続の場合、CICS は、システム始動時にモード・グループ SNASVCMG で LU サービス管理セッションのバインドを試行します。

SINGLESESS(YES) が指定されている接続定義では、AUTOCONNECT オペランドは無視されます。代わりに、セッション定義の AUTOCONNECT オペランドを使用してください。

ALL この定義の場合、ALL は YES と等価ですが、セッション定義との整合性を保つために ALL を指定することができます。

他の CICS システムへの接続の場合は、バインド競合を引き起こす可能性があるため、AUTOCONNECT(ALL) は指定しないでください。

NO CICS は、接続の確立時にセッションのバインドを試行しません。

YES CICS は、接続の確立時に競合勝者セッションのみのバインドを試行します。

AUTOCONNECT オプションは、LU6.1 接続定義には適用されません。LU6.1 接続の場合、初期化時や CEDA のインストール時に接続を確立させるには、SESSIONS 定義で AUTOCONNECT(YES) を指定します。初期化時や CEDA のインストール時に接続を確立させない場合は、SESSIONS 定義に AUTOCONNECT(NO) を指定してください。

BINDPASSWORD

この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。詳しくは、Obsolete attributesを参照してください。

BINDSECURITY({NO|YES}) (APPC only)

バインド時セキュリティーに ESM が使用されているかどうかを指定します。

NO 外部のバインド時セキュリティーは必要ありません。

YES セキュリティーがアクティブであり、XAPPC システム初期設定パラメーターが YES に設定されている場合、CICS は、バインド時セキュリティーを実行するために RACF® からセッション・キーの抽出を試みます。使用可能な RACF プロファイルがない場合、バインドは失敗します。

CONNECTION(*name*)
この接続定義の名前を指定します。名前の長さは最大 4 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

これは、ファイル、端末、トランザクション、およびプログラムの定義で REMOTESYSTEM として指定される名前です。端末定義と接続定義は同じ名前にしないでください。

CONNTYPE({SPECIFIC|GENERIC})
外部 CICS インターフェース (EXCI) 接続に対して、接続の性質を指定します。

GENERIC
接続は非 CICS クライアント・プログラムから CICS システムへの通信用で、汎用です。汎用接続とは、複数の EXCI ユーザーが共用する複数のセッションが含まれる MRO リンクのことをいいます。汎用接続では NETNAME 属性を指定できません。

SPECIFIC
接続は非 CICS クライアント・プログラムから CICS 領域への通信用で、固有です。固有接続とは、クライアント・プログラムで単一のユーザーに占有される 1 つ以上のセッションが含まれる MRO リンクのことをいいます。固有接続では、NETNAME が必須です。

DATASTREAM({USER|3270|SCS|STRFIELD|LMS})
データ・ストリームのタイプを指定します。

LMS データ・ストリームは、LUTYPE6.1 アーキテクチャーで定義された、FMH4 と FMH8 からなる論理メッセージ・サービス (LMS) データ・ストリームです。

SCS データ・ストリームは、LUTYPE6.1 アーキテクチャーで定義された SCS データ・ストリームです。

STRFIELD
データ・ストリームは、LUTYPE6.1 アーキテクチャーで定義された構造化フィールド・データ・ストリームです。

USER データ・ストリームがユーザー定義の場合に、DATASTREAM のデフォルトを USER にします。複数の CICS システムの間で通信を行っている場合は、必ず DATASTREAM をデフォルトの USER にしてください。

3270 データ・ストリームは、タイプ 6.1 論理装置 (LUTYPE6.1) アーキテクチャーで定義された 3270 データ・ストリームです。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。

CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

INDSYS(*connection*)
このシステムとリモート・システムとの通信を中継するために使用される中間システムを定義する、別の CONNECTION の名前を指定します。名前の長さは最大 4 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

中間システムは、ACCESSMETHOD(INDIRECT) を指定した場合にのみ指定できます。

INSERVICE({**YES**|**NO**})
定義されている接続の状況を指定します。

NO 接続はメッセージを受信することも入力を送信することもできません。

YES トランザクションは開始可能であり、メッセージは接続を通して自動的に送信できます。

MAXQTIME({**NO**|*seconds*})
キューに入れられた割り振り要求が、反応がないように見える接続に空きセッションができるのを待機する待ち時間の時間制御を指定します。最大キュー時間は、QUEUELIMIT にキュー限度が指定されている場合にのみ使用され、その後キューの長さがキュー限度値に達した場合にのみ制限時間が適用されます。

NO CICS は空きセッションを待機する割り振り要求のキューを維持します。要求がキューに入れられている時間の長さに時間制限は設定されません（ただし、個々の要求に DTIMOUT メカニズムを適用することはできます)。この場合、値 X'FFFF' が XZIQUE パラメーター・リスト (フィールド UEPEMXQT) に渡されます。

seconds
反応がないように見える接続において割り振り要求をキューに入れることができる時間のおおよその上限を指定します。数値は 0 から 9999 の範囲の秒数を表します。

CICS は、最大キュー時間属性を使用して、待機している割り振り要求のキューを制御します。キューに入っている割り振り要求の数がキューの上限 (QUEUELIMIT) に達した状態でその接続に対する新しい割り振り要求を受け取った場合、キューの処理速度の平均から新規割り振りにかかる時間が最大キュー時間より長くなることが示されると、キューはパージされ、メッセージ DFHZC2300 が発行されます。キューがパージされると、キューに入っている割り振り要求は SYSIDERR を戻します。

接続がセッションの解放を正常に完了するまで、追加のキューイングは行われません。この時点で、CICS は DFHZC2301 を発行し、通常のキューイングを再開します。

また、XZIQUE グローバル・ユーザー出口プログラムを通して割り振り要求のキューイングを制御することもできます。これにより、CICS が提供する統計を使用し、リンクの状態に関するレポートを受け取ることができます。指定するキュー限度および最大キュー時間の値にこれらの統計を組み合わせて使用することにより、キューに関してさらに特化した決定を下すことができます。

MAXQTIME値は、XZIQUEパラメーター・リストで XZIQUEグローバル・ユーザー出口プログラムに渡されます（出口が使用可能な場合)。 XZIQUE グローバル・ユーザー出口の作成に関するプログラミング情報は、MRO および APPC システム間キューの管理用 XZIQUE 出口を参照してください。

明示的な要求がキューに入れられないようにするために、ALLOCATE コマンドに NOQUEUE|NOSUSPEND オプションを指定することもできます。これらの API オプションに関するプログラミング情報は、CICS コマンド・サマリーを参照してください。

NETNAME(*netname***)**

リモート・システムを識別するネットワーク名を指定します。名前の最大長は 8 文字です。名前はアセンブラー言語の規則に従います。名前の先頭は英字にする必要があります。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

NETNAME は、z/OS Communications Server 汎用リソース・グループへの LUTYPE6.1 または APPC リンクを定義していない限り、リモート・システムまたは領域の APPLID です。

- 汎用リソースへの LUTYPE6.1 リンクを定義している場合、NETNAME は、グループ・メンバーのいずれかの APPLID ではなく、汎用リソース名を指定する必要があります。
- 汎用リソースへの APPC リンクを定義している場合、NETNAME は、グループの汎用リソース名か、グループ・メンバーのいずれかの APPLID (メンバー名) のどちらかを指定できます。ただし、メンバー名を指定する場合

で、この CICS 自体は CICS 汎用リソースのメンバーでない場合は、必ずこの CICS (接続定義がインストールされた CICS 領域) によって接続を獲得する必要があります。

z/OS Communications Server の場合、APPLID は、リモート VTAM VBUILD TYPE=APPL ステートメントのラベルです。

NETNAME を指定しない場合は、デフォルトで CONNECTION 名が使用されます。

NETNAME の重複については、いくつかの規則があります。以下のような重複は許可されません。

- 複数の APPC リンクで同じ NETNAME を使用する
- APPC リンクと LUTYPE6.1 リンクで同じ NETNAME を使用する
- 複数の IRC 接続で同じ NETNAME を使用する
- 複数のリモート APPC 接続で同じ NETNAME を使用する
- リモート APPC 接続で他のいずれかの接続やローカル端末と同じ NETNAME を使用する

以下の重複は許可されます。

- IRC 接続と LUTYPE6.1 接続で同じ NETNAME を使用する
- IRC 接続と APPC 接続で同じ NETNAME を使用する
- 複数の LUTYPE6.1 接続で同じ NETNAME を使用する
- いずれかの接続でリモート端末と同じ NETNAME を使用する

z/OS Communications Server LU 別名機能を使用する接続の場合**:**

- **APPC synclevel 1**: CICS 領域が z/OS Communications Server 動的 LU 別名をサポートしている場合 (つまり、CICS 領域の APPL ステートメントで LUAPFX=*xx* が指定されている場合)、この NETNAME は CICS 領域と同じネットワーク内にあるものと見なされます。そうではない場合は、ローカル z/OS Communications Server CDRSC 定義で LUALIAS=*netname* が定義されている必要があります。この *netname* は、この CONNECTION 定義で定義された NETNAME と一致している必要があります。 Synclevel 1 APPC 接続は、通常、ワークステーションです。

  一部の synclevel 1 リソースは、CICS への接続方法によっては synclevel 2 になる可能性があるので注意してください。例えば、TXSeries® が PPC ゲートウェイを使用していない 場合、接続は synclevel 1 です。しかし、PPC ゲートウェイを使用している場合は、synclevel 2 になります。
- **APPC synclevel 2** および **LUTYPE6.1**: この NETNAME は、固有の名前であると見なされます。CICS は、これを z/OS Communications Server APPL ステートメントで定義されたネットワーク名と突き合わせます。これらは一般に CICS-CICS 接続ですが、例えば、PPC ゲートウェイを介した TXSeries 接続の場合もあります。

NETNAME と **APPLID** に関しては、以下のようないくつかの規則があります。

-

- インストール済みの CONNECTION 定義がインストール済みの IPCONN 定義と同じ名前を持つ場合、IPCONN 定義の APPLID は CONNECTION 定義の NETNAME と一致している必要があります。そうなっていない場合、以下のように状況によって異なるメッセージが出されます。
- DFHIS3009 (IPCONN の自動インストール時にエラーを検出した場合)
- DFHAM4913 (IPCONN のインストール時にエラーを検出した場合)
- DFHZC6312 (CONNECTION のインストール時または自動インストール時にエラーを検出した場合)

• IPCONN 定義は CONNECTION 定義に優先します。つまり、IPCONN と CONNECTION が同じ名前を持っている場合、CICS は IPCONN 接続を使用します。

• 同じ NETNAME と APPLID を持つ CONNECTION と IPCONN が同じ名前を持つ必要はありません。

これによって、IPCONN がある場合に、IPCONN によってすべてのサポート機能をルーティングするという CICS のデフォルトを使用する代わりに、TCP/IP を介した通信で個別のシステム ID を使用できるようになります。

PROTOCOL({APPC|LU61|EXCI|blank})
リンクに使用するプロトコルのタイプを指定します。

APPC (LUTYPE6.2 プロトコル)
拡張プログラム間通信 (APPC) プロトコル。これは、ACCESSMETHOD(VTAM) のデフォルト値です。これは CICS-CICS ISC の場合に指定してください。

ブランク
CICS 領域間の MRO。 MRO の場合は PROTOCOL をブランクのままにしておく必要があります。また、SESSIONS 定義では、PROTOCOL として LU6.1 を指定する必要があります。

EXCI 外部 CICS インターフェース。これを指定すると、この接続が外部 CICS インターフェースを使用する非 CICS クライアント・プログラムによって使用されることが示されます。

LU61 LUTYPE6.1 プロトコル。これは、CICS-CICS ISC または CICS-IMS ISC に指定し、MRO には指定しないでください。

PSRECOVERY({SYSDEFAULT|NONE})
持続セッション・サポートと一緒に実行されている CICS 領域では、LU6.2 セッションが持続セッション遅延間隔内のシステム再始動でリカバリーされるかどうか、およびその方法を指定します。

なし すべてのセッションは、CNOS のリカバリーなしにアンバインドされ、サービス不能となります。

SYSDEFAULT
障害を起こした CICS システムが持続セッション遅延間隔内に再始動した場合、次のアクションが発生します。

- ユーザー・モードグループが SESSIONS RECOVOPTION 値にリカバリーされる。

- SNASVCMG モードグループがリカバリーされる。
- 接続が ACQUIRED 状態に戻され、最後に折衝された CNOS 状態が返されます。

QUEUELIMIT({**NO**|*number*})

空きセッションを待っている間に、CICS がキューイングする割り振り要求の最大数を指定します。

NO 空きセッションを待っている間に CICS がキューに入れることのできる割り振り要求数の設定には制限がありません。この場合、値 X'FFFF' が XZIQUE パラメーター・リスト (フィールド UEPQUELM) に渡されます。

number

空きセッションを待っている間に、CICS がその接続においてキューイングできる割り振り要求の最大数です (0 から 9999 の範囲)。キューに入っている割り振り要求の数がこの制限に達すると、それより後の割り振り要求は、キューが制限を下回るまで SYSIDERR を返します。

このキュー限度は、XZIQUE パラメーター・リストで XZIQUE グローバル・ユーザー出口プログラムに渡されます (出口が使用可能な場合)。

また、MAXQTIME 属性や XZIQUE グローバル・ユーザー出口プログラムを通して割り振り要求のキューイングを制御することもできます。キューの制御について詳しくは、MAXQTIME 属性の説明を参照してください。

注: BIND の再折衝は、使用されていない 2 次セッションがあったとしても、トリガーされません。CEMT SET MODE コマンドを使用して強制的に再折衝を行わない限りは、1 次セッションがすべて使用中になるとすぐにキュー制限の動作が開始されます。

RECORDFORMAT({**U**|**VB**})

SNA チェーンのタイプを指定します。

U SNA チェーンが単一の非ブロック化データ・ストリームである場合は、RECORDFORMAT のデフォルトを U にします。SNA チェーン内に専用のブロック・アルゴリズムを持つことが可能です。複数の CICS システム間で通信を行っている場合は、RECORDFORMAT をデフォルトの U にします。

VB SNA チェーンは、LUTYPE6.1 アーキテクチャーで定義されているように VLVB 標準に従ってフォーマットされます。

REMOTENAME(*connection*)

接続を所有するシステムまたは領域において、トランザクション・ルーティングの APPC 接続を識別するための名前を指定します。名前の長さは最大 4 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

リモートのシステムまたは領域は、APPC 装置にすることができます (トランザクション・ルーティング用の APPC 装置を参照)。

REMOTESYSNET(*netname*)
接続を所有するシステムのネットワーク名 (APPLID) を指定します。名前の最大長は 8 文字です。この名前はアセンブラー言語の規則に従うため、名前の先頭は英字でなければなりません。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

REMOTESYSNET は、リモート APPC システムまたはデバイスへのトランザクション・ルーティングを行う場合で、この定義がインストールされている領域とリモート・デバイスへの接続を所有するシステムの間に直接リンクが存在しない場合に使用します。次のような場合は、REMOTESYSNET を指定する必要はありません。

- ローカル接続を定義している場合 (つまり、REMOTESYSTEM を指定していないか、ローカル・システムのシステム ID を指定している)。
- REMOTESYSTEM が、接続を所有するシステムへの直接リンクの名前を指定している。ただし、1 つ特殊なケースがあります。接続を所有している領域が z/OS Communications Server 汎用リソース・グループのメンバーで、それに対する直接リンクが APPC 接続である場合は、REMOTESYSNET の指定が必要になることがあります。このケースで、直接リンクの CONNECTION 定義で指定されている NETNAME が (APPLID ではなく) 接続を所有している領域の総称リソース名である場合は、REMOTESYSNET が必要です。

REMOTESYSTEM(*connection*)
接続を所有するシステムへの相互通信リンクを識別する名前を指定します。名前の長さは最大 4 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

これは、相互通信リンクの接続定義上の CONNECTION 名です。

リモート APPC システムまたはデバイスへのトランザクション・ルーティングには REMOTESYSTEM が使用されます。この属性が指定されない場合、またはローカル・システムのシステム ID として指定される場合、この接続はこのシステムに対してローカルになります。別のシステムのシステム ID が指定される場合は、接続はリモートになります。したがって、ローカル・システムとリモート・システムの両方で接続に同じ定義を使用することが可能です。

この CICS とデバイス (への接続) を所有する領域との間に中間システムが存在する場合、REMOTESYSTEM は、デバイスを所有する領域へのパスの最初のリンクを指定します。可能なパスが複数ある場合、優先パスの最初のリンクを指定する必要があります。

SECURITYNAME(*userid***)**
APPC および LU6.1 リンクの場合のみ、これはリモート・システムのセキュリティー名です。

セキュリティーが初期化された CICS システム (SEC=YES) では、セキュリティー名はリモート・システムの権限を設定するために使用されます。

注: 接続定義に関連したSESSIONS定義に USERIDが指定されている場合は、SECURITYNAME属性に指定されたユーザーID を無効にして、リンクの 安全保護に使用されます。

セキュリティー名 (またはセッション定義の USERID) は、ご使用のシステムの有効な RACF ユーザー ID でなければなりません。システム上の保護リソースへのアクセスは、RACF ユーザー・プロファイルおよびそのグループ・メンバーシップに基づいています。

SINGLESESS({NO|YES})
定義が CICS への単一セッション APPC リンク上の APPC 端末を対象としたものであるかどうかを指定します。

NO 定義は CICS への単一セッション APPC リンクを対象としたものではありません。

YES 定義は CICS への単一セッション APPC リンク上の APPC 端末を対象としたものです。

SESSIONS 定義の MODENAME 属性は、単一セッション・モードセットにモード名を提供するために使用できます。

APPC 単一セッション端末を TERMINAL-TYPETERM 定義として定義することもできます。TERMINAL-TYPETERM 定義と CONNECTION 定義は、いずれも自動インストールが可能です。自動インストールの使用を検討している場合は、z/OS Communications Server 端末の自動インストールを参照してください。

USEDFLTUSER ({NO|YES}) (APPC と MRO のみ)
ATTACHSEC 属性で暗黙的に指定されたセキュリティー情報がインバウンド FMH5 に含まれない場合に取るアクションを指定します。

NO 接続要求は拒否され、プロトコル違反メッセージが発行されます。

YES 接続は受け入れられ、デフォルト・ユーザー *ID* がトランザクションに関連付けられます。

詳しくは、LU6.2 セキュリティーの実装および MRO セキュリティーの実装を参照してください。

XLNACTION({KEEP|FORCE}) (APPC および MRO のみ)
XLNACTION は、パートナー・システムから新しいログ名を受け取ったときに行うアクションを指定します。新しいログ名の受信は、パートナーがリカバリー情報を削除したことを示します。

注**:** ここで MRO は、ACCESSMETHOD セットを使用して、IRC または XM のいずれかとの接続をカバーします。

FORCE

新しいログ名を使用して何らかの新しい作業を開始する前に、未確定 UOW の事前定義された決定 (トランザクション定義の INDOUBT 属性で定義) をインプリメントします。また、CICS は、パートナー・システムでは未確定であった UOW を解決するために保管されていた情報も削除します。

注意**:** このオプションを使用すると、データ保全性が損なわれる恐れがあります。

KEEP リカバリー情報を保持し、未確定作業単位に対しては何もアクションを実行しません。

IRC の場合、接続は新しい作業で続行されます。CEMT または SPI インターフェースを使用して未確定 UOW を解決してください。

APPC の場合、接続は、パートナーでの未解決のリカバリー可能作業 (未確定 UOW、または古いログ名の下でパートナー・システム上に未確定になっていた UOW に関係する情報) がすべて CEMT または SPI インターフェースを使用して完了されるまで、synclevel 2 プロトコルを必要とする新しい作業を実行できません。

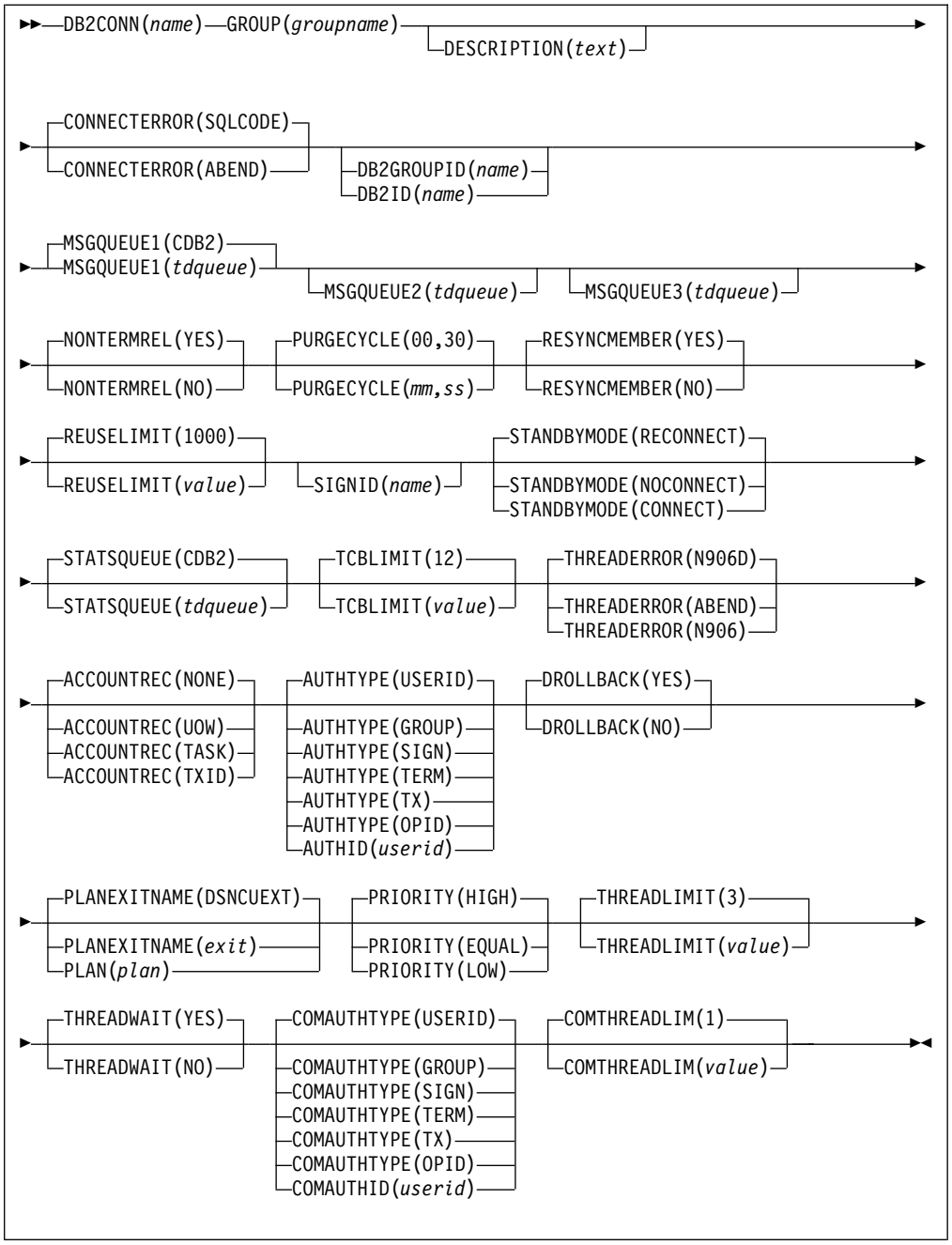
DB2CONN リソース

DB2CONN 定義は、CICS と Db2® の間の接続の属性、および接続に使用されるプール・スレッド とコマンド・スレッド の属性を定義します。

CEMT および CEDA を使用した DB2CONN リソース定義のインストールについては、DB2CONN リソース定義のインストールを参照してください。BAS の作業については、BAS Db2 接続リソース定義の作業を参照してください。

DB2CONN 属性

DB2CONN リソースの構文と属性について記述します。



一般属性

DB2CONN リソースの一般属性について説明します。

DB2CONN(*name*)

Db2 接続定義を識別するための名前。名前の最大長は 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

DESCRIPTION(*text*)

このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)

各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

接続属性

接続に関連する DB2CONN リソースの属性について説明します。

CONNECTERROR({**SQLCODE**|**ABEND**})

接続機能が「待機モード」にあるために CICS が Db2 に接続されないという情報を、SQL 要求を発行したアプリケーションに報告する方法を指定します。

ABEND

アプリケーションは、異常終了コード AEY9で異常終了します。

SQLCODE

アプリケーションは、-923の SQLCODEを受信します。
STANDBYMODE が NOCONNECT に設定されている場合、SQLCODE は指定できません。

DB2GROUPID(*name*)

Db2 サブシステムのデータ共用グループのグループ ID (最大 4 文字) を指定します。グループ接続機能は、CICS をこのデータ共用グループの任意のアクティブ・メンバーに接続します。グループ ID を、Db2 で定義されているグループ接続名と突き合わせます。 DB2® バージョン 10 では、グループ ID を Db2 に定義されるサブグループ接続名にすることができます。この名前は、データ共用グループのサブセットを定義します。DB2GROUPID 属性をブランクのままにすると、グループ接続は使用されません。 DB2GROUPID と DB2ID の両方を指定することはできません。優先順位は以下のとおりです。

1. DB2GROUPIDを指定すると、DB2CONN定義 にすでに設定されているDB2IDが無効に なります。
2. 1つのCEDAパネルで DB2GROUPIDとDB2IDの両方を指定しようと した場合、DB2IDが使用されます。
3. 個々のサブシステムの DB2ID が CEMT または **EXEC CICS SET DB2CONN** コマンド、あるいは DSNC STRT コマンドで指定された場合、この DB2ID は、インストール済みの DB2CONN 定義で設定されているすべて

の DB2GROUPID 属性をオーバーライドします。インストール済みの DB2CONN 定義の DB2GROUPID がブランクになり、グループ接続を使用するには (CEDA コマンドまたは **SET DB2CONN** コマンドを使用して) 再設定する必要があります。

DB2ID(*name*)

CICS Db2 接続機能の接続先となる Db2 サブシステムの名前を指定します。デフォルトでは、このフィールドはブランクです。 グループ接続を使用する場合、DB2CONN 定義に DB2ID ではなく DB2GROUPID を指定します。インストール済みの DB2CONN 定義で設定された DB2ID は、DSNC STRT コマンドで指定された Db2 サブシステム ID、あるいは SET DB2CONN コマンドで指定された DB2ID によってオーバーライドすることができます。インストール済みの DB2CONN 定義で DB2ID をブランクのままにし、DB2GROUPID もブランクのままにしておくと、INITPARM システム初期設定パラメーターで Db2 サブシステム ID を指定することができます。これらのいずれの方法でも Db2 サブシステム ID が指定されておらず、DB2GROUPID も指定されていない場合、接続の試行時にデフォルトの DB2ID であるブランクが DSN に置き換わります。そのため、Db2 サブシステムを判別するための階層は、以下のようになります。

1. DSNC STRT コマンドで指定されている場合は、サブシステム ID を使用します。
2. インストール済みの DB2CONN の DB2ID (ブランクではない場合) を使用します。
3. グループ接続の場合は、インストール済みの DB2CONN の DB2GROUPID (ブランクではない場合) を使用します。
4. 最後にインストールされた DB2CONN の DB2ID および DB2GROUPID がブランクの場合 (あるいは後でブランクに設定された場合) は、サブシステム ID (INITPARM で指定されている場合) を使用します。最後にインストールされた DB2CONN にブランクの DB2ID とブランクの DB2GROUPID が含まれている場合は、SET コマンドを使用して後で DB2ID または DB2GROUPID が変更された場合でも、始動時には必ず INITPARM が使用されます。
5. デフォルトのサブシステムIDであるDSNを使用します。

DB2GROUPID と DB2ID の両方を指定することはできません。この両方を同じ CEDA パネルで指定しようとした場合は、DB2ID が使用されます。DB2GROUPID が CEMT または **EXEC CICS SET DB2CONN** コマンドで指定されている場合、インストールされている DB2CONN 定義で設定された DB2ID がオーバーライドされ、DB2ID がブランクになります。

MSGQUEUE1({**CDB2**|*tdqueue*})

CICS Db2 接続機能からの非送信請求メッセージの送信先となる、最初の一時データ宛先を指定します。この最初の宛先は、ブランクにすることはできません。

MSGQUEUE2(*tdqueue*)

CICS Db2 接続機能からの非送信請求メッセージの送信先となる、2 番目の一時データ宛先を指定します。

MSGQUEUE3(*tdqueue*)
CICS Db2 接続機能からの非送信請求メッセージの送信先となる、3 番目の一時データ宛先を指定します。

NONTERMREL({**YES**|**NO**})
中間の同期点での再使用のために、非端末トランザクションがスレッドを解放するかどうかを指定します。

NO 非端末トランザクションは、中間の同期点でスレッドを再利用のために解放しません。

YES 非端末トランザクションは、中間の同期点でスレッドを再利用のために解放します。

PURGECYCLE({**00**|*mm*},{**30**|*ss*})
保護スレッド・パージ周期の長さ (分単位) を指定します。パージ周期の長さは 5 秒から 59 分 59 秒の範囲です。PURGECYLE の値を指定しない場合、デフォルトは 30 秒 (PURGECYCLE= 00,30) になります。

保護スレッドは、解放されてもすぐには終了しません。その間に再利用されなければ、パージ周期 2 回分の時間が経過した後に終了します。したがって、パージ周期が30秒に設定され ている場合、保護されたスレッドは解放の 30秒から60秒後にパージされます。接続機能 の開始後の最初のパージ周期は、常に5分 です。その後、パージ周期値が適用されます。その DB2ENTRY でスレッドを待機している他のトランザクションがない場合、無保護スレッドは (同期点またはタスクの最後で) 解放されるときに終了します。 DB2ENTRYに属する スレッドのみを保護することができます。プール・ スレッドおよびコマンド・スレッドは保護できません。

RESYNCMEMBER({**YES**|**NO**})
グループ接続を使用する場合、RESYNCMEMBER 属性を使用して、CICS が接続されていた最後の Db2 データ共用グループ・メンバーに対して未解決の作業単位が保持されている場合に、 CICS が採用する戦略を選択します。

YES 未解決の作業単位が保持されている場合に、CICS が接続されていた最後の Db2 データ共用グループ・メンバーとの再同期を要求することを示します。 CICS はグループ接続機能を無視し、CICS-Db2 接続機能は未確定の作業単位を解決するために、その最後に接続されていた Db2 データ共用グループ・メンバーに再接続できるまで待機します。未確定で中断された作業単位は、この時点では CICS 自体がそれらの作業単位を解決できないので、このプロセスに含まれません。これらの作業単位の再同期は、 CICS がそのリモート・コーディネーターと再同期したときに行われます。

NO 再同期を必要としないことを示します。CICS は、最後に接続されていた Db2 データ共用グループ・メンバーへの再接続を 1 回だけ試行します。この試行が成功した場合は、未確定の作業単位 (中断している未確定の UOW を除く) を解決することができます。失敗した場合は、CICS はグループ接続を使用して Db2 データ共用グループ内のいずれかのアクティブ・メンバーに接続し、CICS が接続していたグループの最後のメンバーについて、解決されていない未確定の作業単位がある可能性があることを示す警告メッセージ (DFHDB2064) を発行します。

REUSELIMIT(*value*)

スレッドが強制終了される前に再利用できる最大回数を示す値を 0 から 10000 の範囲で指定します。デフォルトは、1000 です。 0という値を指定すると、スレッドを再使用できる回数に制限 がなくなります。CICS TS 4.2より前は、このような 動作でした。ただし、長期実行される CICS Db2 スレッドが絶え間なく再使用されると、Db2 内にリソースが蓄積してストレージの問題が生じ、異常終了や Db2 サブシステムの障害を引き起こす可能性があります。

再使用制限の適用対象は、非保護スレッド(プール内のスレッドと DB2ENTRY のスレッドの両方)と保護DB2ENTRYスレッドです。 非保護スレッドが再使用されるのは、1つのトランザクションから 解放された時点で新しいトランザクションが既に待機している場合 です。保護スレッドが再使用されるのは、スレッドが終了しない ように保護されている間に、新しいトランザクションでスレッドが 必要になる場合です。どちらの場合も、再使用制限値に達すると、 それより後のトランザクションは、スレッドを使用できません。現在スレッドを使用しているトランザクションがそのスレッドを解放すると、CICS は、そのスレッドを終了して再作成して Db2 リソースを解放した後、スレッドで実行する新規の処理があるか、あるいはスレッドを保護するべきかを判別します。

デフォルトの 1000 を使用することで、Db2 バインド・オプションの RELEASE(DEALLOCATE) を使用する場合でも、パフォーマンスに悪影響を及ぼすことなく、2 GB 境界より下にあるスレッド・ストレージおよび EDM プールを過剰な割り振りから保護することができます。ただし、Db2 のモニターおよび統計で、Db2 スレッド・ストレージまたは EDM プール・ストレージ (あるいはその両方) の過剰な使用が示された場合、この制限を下げることもできます。逆に、CICS-Db2 統計で、プールまたはエントリーのスレッドが頻繁に再使用制限に達していることが示されており、Db2 スレッド・ストレージを増やすのに十分な仮想ストレージおよび実ストレージが使用可能である場合は、この制限を上げることもできます。

再使用制限値を小さく設定すると、プロセッサーの活動 が増えてスループットが減る結果になり、パフォーマンスに悪影響 が及びます。ただし、小さい値をあえて 設定するのが望ましい状況もあり ます。例えば、プランやパッケージのDb2バインド・オプションを RELEASE(COMMIT)からRELEASE (DEALLOCATE)に変更したときの影響を 評価する場合などには、テストのために一時的に小さい値を使用 できます。

SIGNID(*name*)

CICS Db2 接続機能が、プールおよび AUTHTYPE(SIGN) を指定する DB2ENTRY スレッドを使用するために Db2 にサインオンする際に使用する許可 ID を指定します。デフォルトはブランクで、DB2CONN のインストール時に CICS システムのアプリケーション ID に置き換えられます。この ID は、最大 8 文字の長さで指定することができます。

許容文字: `A-Z 0-9 $ @ #` CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

SIGNID 属性にユーザー ID を指定した場合、CICS は、インストールを実行するユーザー ID に対して代理ユーザー検査を実行します。同様に、CICS 領域のユーザー ID は、CICS コールド・スタートまたは初期始動でのグループ・リスト・インストールの際に代理ユーザー検査の対象になります。

指定した ID が CICS 領域ユーザー ID と一致し、いずれかのコマンド、プール、またはエントリー・スレッドで AUTHTYPE(SIGN) を指定した場合、その CICS 領域ユーザー ID の RACF アクセス制御環境エレメント (ACEE) が Db2 に渡されます。

STANDBYMODE({**RECONNECT**|**CONNECT**|**NOCONNECT**})

CICS を Db2 に接続しようとしたときに Db2 がアクティブではない場合に、CICS Db2 接続機能が実行するアクションを指定します。

CONNECT

Db2 がアクティブになるまで CICS Db2 接続機能が「待機モード」で待機することを指定します。接続が確立され、その後で Db2 に障害が発生した場合、CICS Db2 接続機能は終了します。

NOCONNECT

CICS Db2 接続機能を終了することを指定します。

RECONNECT

CICS Db2 接続機能が「待機モード」に入り、Db2 を待つことを指定します。接続が確立された後で Db2 に障害が発生した場合、CICS Db2 接続機能は「待機モード」に戻り、CICS は、Db2 が復旧したら Db2 に再接続します。

STATSQUEUE({**CDB2**|*tdqueue*})

CICS Db2 接続機能がシャットダウンされたときに生成された CICS Db2 接続機能統計の一時データ宛先を指定します。

TCBLIMIT({**12**|*value*})

Db2 要求の処理に使用できる TCB の最大数を指定します。デフォルトは 12 です。 最小数は 4 で、最大数は 2000 です。CICS は、L8 および L9 モードを使用して TCB を開き、Db2 要求を処理します。DB2CONN 定義の TCBLIMIT 属性は、Db2 にアクセスするために使用できるオープン TCB の数、つまり、TCB が Db2 に識別され、Db2 への接続を作成できる数を規定します。

TCB の制限およびオープン TCB モードについて詳しくは、オープン TCB 管理を参照してください。

TCBLIMIT 値は、 CICS 領域のスレッドの総数を制御します。そのため、TCBLIMIT の推奨値は、すべてのスレッド制限値の合計 (つまり、Db2 接続および Db2 エントリー・リソース定義のすべての THREADLIMIT 属性の合計に、Db2 接続定義の COMTHREADLIMIT 値を加算したもの) です (上限は 2000)。

L8 および L9 モードのオープン TCB の数が、CICS によって自動的に設定された制限に達したために追加の TCB を作成することができなくなった場合、タスクは、HTYPE(DISPATCH) および HVALUE(OPEN_TCB) で中断されます。CICS は、数式 (2 * MXT 値) + 32 と CICS 領域の MXT あるいは MAXTASKS の制限を使用して、この制限を設定します。この制限は超えてい

ないが、TCBLIMIT を超えた場合、タスクは HTYPE(CDB2CONN) で中断されます。この状態では、CICS には使用可能なオープン TCB がありますが、(TCBLIMIT で定義されている) 許可される最大数のオープン TCB が Db2 へのアクセスに使用されています。

TCBLIMIT の数値を決定する場合、Db2 インストール・パネル DSNTIPE で MAX USERS パラメーターに指定した量を考慮する必要があります。

THREADERROR({N906D|N906|ABEND})
スレッドの作成エラーの後に発生する 処理を指定します。

ABEND
最初の SQL エラーが検出されると、CICS は、エラーのタイプに応じて、異常終了コード AD2S、AD2T、または AD2U に対するトランザクション・ダンプを取得します。最初のエラーについては、トランザクションは異常終了しません。 2 回目以降の SQL エラーについては、トランザクションは、異常終了コード AD2S、AD2T、または AD2U で異常終了します。トランザクションは終了して再初期化されてからでなければ、別の SQL 要求を発行することはできません。

N906D
トランザクション・ダンプが取られ、トランザクションに関連付けられた DSNCSQL RMI は使用不可になりません。別の SQL 要求が発行された場合、トランザクションは、SYNCPOINT ROLLBACK を発行しない限り、-906 SQLCODE を受け取ります。 ROLLBACK オプションを指定しない SYNCPOINT では、ASP3 または ASP7 の異常終了が発生します。トランザクション・ダンプには、AD2S、AD2T、または AD2U の異常終了が記録されます。

N906 トランザクションに関連付けられた DSNCSQL RMI は使用不可になりません。別の SQL 要求が発行された場合、トランザクションは、SYNCPOINT ROLLBACK を発行しない限り、-906 SQLCODE を受け取ります。 SYNCPOINT に ROLLBACK オプションが指定されていない場合、ASP3 または ASP7 で異常終了します。

プール・スレッド属性

プール・スレッドに関連する DB2CONN リソースの属性について説明します。

ACCOUNTREC({NONE|TASK|TXID|UOW})
プール・スレッドを使用するトランザクションに必要な Db2 アカウンティングの最小量を指定します。以下のオプションで説明するように、指定された最小量を超える場合があります。

NONE
プール・スレッドを使用するトランザクションに必要なアカウンティング・レコードはありません。

Db2 は、スレッドが終了したときに、各スレッドに対して少なくとも 1 つのアカウンティング・レコードを作成します。さらに、権限を変更してもアカウンティング・レコードが作成されます。

TASK
CICS Db2 接続機能により、各 CICS タスクに対して 1 つのアカウンティング・レコードの最小値が作成されます。

複数の作業単位 (UOW) が含まれ、スレッドが同期点で解放されるトランザクションでは、UOW ごとに異なるスレッドを使用することができます。その結果、UOW ごとにアカウンティング・レコードが生成される場合があります。

TXID CICS Db2 接続機能により、スレッドを使用するトランザクション ID が変わったときにアカウンティング・レコードが生成されます。

通常、プール・スレッドは多くの異なるトランザクション ID によって使用されるため、複数の UOW が含まれ、同期点でスレッドが解放されるトランザクションが UOW ごとに異なるスレッドを使用する機会は増えています。この場合、UOW ごとにアカウンティング・レコードが生成されることがあります。

UOW CICS Db2 接続機能により、各 UOW のアカウンティング・レコードが作成されます (このスレッドは UOW の終わりに解放されます)。

AUTHID(*userid*)

プール・スレッドの使用時にセキュリティー検査に使用するユーザー ID を指定します。

Db2 アドレス・スペースのセキュリティー検査の一部または全部に RACF を使用している場合は、AUTHID を指定しないでください。代わりに、GROUP オプション、SIGN オプション、または USERID オプションを指定して AUTHTYPE を使用してください。 AUTHID を使用するスレッドは必要な RACF アクセス制御環境エレメント (ACEE) を Db2 に渡さないため、AUTHTYPE を使用する必要があります。 この ID は、最大 8 文字の長さで指定することができます。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

AUTHTYPE({USERID|OPID|GROUP|SIGN|TERM|TX})

この DB2CONN のスレッドに使用できる ID のタイプを指定します。

Db2 アドレス・スペースのセキュリティー検査の一部または全部に RACF を使用している場合は、GROUP オプション、SIGN オプション、または USERID オプションを使用します。これらのオプションを使用して定義されたスレッドのみが必要な RACF アクセス制御環境エレメント (ACEE) を Db2 に渡すため、これらのオプションのいずれかを使用する必要があります。ただし、SIGN オプションを指定すると、DB2CONN 定義で SIGNID 属性に指定された値が CICS 領域のユーザー ID と一致した場合にのみ ACEE が Db2 に渡されます。

Db2 内部セキュリティーのみを使用している場合は、ACEE は不要です。この場合、任意のオプションを使用することができます。

USERID

CICSトランザクションに関連した ユーザーIDが許可IDとして使用 されます。ユーザー ID の長さが 8 文字未満の場合、右側にブランクが埋め込まれます。

重要: Db2 サンプル・サインオン出口 DSN@SGN を使用する場合は、SQL -922 障害が発生する可能性があるため、AUTHTYPE(USERID) を指定しないでください。代わりにCOMMAUTHTYPE(GROUP)を指定してください。

OPID CICS トランザクションに関連付けられたユーザー ID に関連付けられたオペレーター ID は、許可 ID として使用されます。 3文字のオペレーター識別の右側に ブランクが埋め込まれて、8文字の許可 IDになります。

GROUP

ユーザーIDおよび接続グループ名 を許可IDとして指定します。 以下の表に、これらの 2 つの値が Db2 でどのように解釈されるかを示します。

Db2 へ渡される **ID**	**Db2** が値を解釈する方法
CICS サインオン・ユーザー ID (USERID)	1 次 Db2 許可 ID を表します。
RACF 接続グループ名	グループ・オプションの RACF リストがアクティブではない場合、Db2 は CICS 接続機能が提供する接続されたグループ名を 2 次 Db2 許可 ID として使用します。グループ・オプションの RACF リストがアクティブな場合、Db2 は CICS 接続機能が提供する接続されたグループ名を無視しますが、2 次 Db2 許可 ID の Db2 リストには値が表示されます。

GROUP オプションを使用するには、領域のシステム初期設定パラメーターで SEC=YES を指定する必要があります。

この USERID に対して使用できる RACF グループ ID がない場合、ブランクの 8 文字フィールドがグループ ID として Db2 に渡されます。

SIGN Db2接続定義のSIGNID属性を、資源の 許可IDとして使用することを指定します。

TERM

端末識別名を許可IDとして指定 します。4文字の端末識別名の 右側にブランクが埋め込まれて、 8文字の許可IDになります。

トランザクションが端末に関連付けられ ていない場合（例えば、STARTコマンドで 開始される場合）は、AUTHTYPE (TERM)を 指定しないでください。

TX トランザクション識別 を許可IDとして指定します。 4文字のトランザクション 識別の右側にブランクが 埋め込まれて、8文字の許可IDに なります。

DROLLBACK({YES|NO})

デッドロック解決に関連するトランザクションが選択された場合に、CICS Db2 接続機能が SYNCPOINT ROLLBACK を開始するかどうかを指定します。

YES 制御をアプリケーションに返す前に、接続機能が SYNCPOINT ROLLBACK を開始します。 SQL 戻りコードの -911 がプログラムに返されます。

エンタープライズ Bean を OTS トランザクションの一部として実行しているトランザクションがプールを使用する場合は、YES を指定しないでください。SYNCPOINT ROLLBACK は、OTS トランザクションでは許可されません。エンタープライズ Bean を OTS トランザクションの一部として実行するトランザクションで使用するために、DROLLBACK(NO) を指定した DB2ENTRY を定義することを検討してください。

NO 接続機能で、トランザクションの ロールバックを開始しません。SQL 戻りコードの -913 がアプリケーションに戻されます。

PLAN(*plan***)**
すべてのプール・スレッドで使用するプラン の名前を指定します。

PLANEXITNAME({DSNCUEXT|*exit***})**
プール・スレッドで使用する動的プラン出口 の名前を指定します。 プールに対してアクティブなトランザクションがある間に PLAN および PLANEXITNAME を変更した場合、次にトランザクションがスレッドを解放するときに新規値が適用されます。

PRIORITY({HIGH|EQUAL|LOW})
CICSのメインTCB (QR TCB)に対するプール・スレッド TCBの優先順位を指定します。スレッド TCB は、CICS オープン L8 TCB です。

HIGH
スレッドTCBの優先順位は、CICS QR TCBより 高くなります。

EQUAL
スレッドTCBの優先順位は、 CICS QR TCBと同じです。

LOW スレッドTCBの優先順位は、CICS QR TCB より低くなります。

THREADLIMIT({3|*value***})**
THREADWAIT 属性に従って要求を待機させる、あるいは拒否する前に、CICS Db2 接続機能がアクティブになることを許可するプール・スレッドの現在の最大数を指定します。また、デフォルト値の THREADLIMIT (3) は、指定可能な最小値です。最大値は、TCBLIMIT に指定された値以下でなければなりません。

THREADWAIT({YES|NO})
アクティブ・プール・スレッド数がスレッド制限に達した場合に、トランザクションがプール・スレッドを待つか、異常終了するかを指定します。

THREADWAIT=NO がコーディングされているときにプール・スレッドの数が超過した場合、CICS Db2 接続は、固有の異常終了コード AD3T、メッセージ DFHDB2011 を発行します。

YES すべてのスレッドが使用されている場合、いずれかが使用可能になるまでトランザクションは待機している必要があります。トランザクションは、CICS が許可している限り待機できます。通常はスレッドが使用可能になるまでです。

NO すべてのスレッドが使用中の場合、トランザクションはコード AD2T または AD3T で異常終了します。

コマンド・スレッド属性

コマンド・スレッドに関連する DB2CONN リソースの属性について説明します。
コマンド・スレッドが使用されるのは、DSNC トランザクションを使用してコマンドを Db2 に送信する場合と、CICS が、CICS タスクのパージまたは強制パージ要求の処理の一環として、 Db2 のアクティブなスレッドを取り消すコマンドを発行する場合です。

Db2 接続定義のコマンド・スレッド属性の説明は以下のとおりです。

COMAUTHID(*userid*)
コマンド・スレッドの使用時に、CICS Db2 接続機能がセキュリティー検査に使用する ID を指定します。

Db2 アドレス・スペースのセキュリティー検査の一部または全部に RACF を使用している場合は、COMAUTHID を使用しないでください。代わりに、USERID オプションまたは GROUP オプションを指定して、COMAUTHTYPE を使用してください。 DB2CONN 定義の SIGNID 属性が CICS 領域のユーザー ID と一致する場合は、SIGN オプションを指定して COMMAUTHTYPE を使用することもできます。これは、COMAUTHIDを 使用するスレッドは、必要なRACFアクセス制御 環境エレメント(ACEE)をDb2に渡さないため です。Db2 内部セキュリティーのみを使用している場合は、ACEE は不要です。この場合、COMAUTHID を使用することができます。ID は、最大 8 文字の長さで指定することができます。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

COMAUTHTYPE({USERID|OPID|GROUP|SIGN|TERM|TX})
コマンド・スレッドの使用時に、安全保護 検査で使用できるIDのタイプを指定します。

Db2 アドレス・スペースのセキュリティー検査の一部または全部に RACF を使用している場合は、GROUP オプション、SIGN オプション、または USERID オプションを使用します。これは、これらのオプションで 定義されたスレッドのみが、必要なRACFアクセス 制御環境エレメント(ACEE)をDb2に渡すため です。ただし、SIGN オプションを指定すると、DB2CONN 定義で SIGNID 属性に指定された値が CICS 領域のユーザー ID と一致した場合にのみ ACEE が Db2 に渡されます。

USERID
CICSトランザクションに関連した1文字 から8文字のユーザーIDが、許可ID として使用されます。この名前の長さは最大 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

重要: Db2 サンプル・サインオン出口 DSN@SGN を使用する場合は、SQL -922 障害が発生する可能性があるため、COMMAUTHTYPE (USERID) を指定しないでください。代わりにCOMMAUTHTYPE (GROUP)を指定してください。

OPID CICSのトランザクション・サインオン 機能に関連したユーザーIDに関連した オペレーター識別が、許可IDとして 使用されます(3文字に埋め込みが行わ れ8文字になります）。

GROUP

1文字から8文字のUSERIDを指定し、 接続グループ名を許可IDとして指定 します。以下の表に、これらの 2 つの値が Db2 でどのように解釈されるかを示します。

Db2 へ渡される **ID**	**Db2** が値を解釈する方法
CICS サインオン・ユーザー ID (USERID)	1 次 Db2 許可 ID を表します。
RACF 接続グループ名	グループ・オプションの RACF リストがアクティブではない場合、Db2 は CICS 接続機能が提供する接続されたグループ名を 2 次 Db2 許可 ID として使用します。グループ・オプションの RACF リストがアクティブな場合、Db2 は CICS 接続機能が提供する接続されたグループ名を無視しますが、2 次 Db2 許可 ID の Db2 リストには値が表示されます。

CGROUP オプションを使用するには、CICS システムで、CICS システム初期設定テーブル (SIT) の SEC=YES を指定する必要があります。

この USERID に対して使用できる RACF グループ ID がない場合、ブランクの 8 文字フィールドがグループ ID として Db2 に渡されます。

SIGN DB2CONNのSIGNID属性を、資源の 許可IDとして使用することを指定します。

TERM

許可 ID として端末 ID (8 文字に当てられる 4 文字) を指定します。端末がトランザクションに接続されていない場合、この方法で許可 ID を取得することはできません。

CICS コマンドを使用するトランザクションを開始しても、それに関連付けられた端末がない場合、COMAUTHTYPE(TERM) を使用できません。

TX 許可 ID としてトランザクション ID (8 文字に当てられる 4 文字) を指定します。

COMTHREADLIM({1|value})
この数値は、要求がプールにオーバーフローする前に、CICS Db2 接続機能が、アクティブにすることを許可するコマンド・スレッドの現在の最大数を指定します。

DB2ENTRY リソース

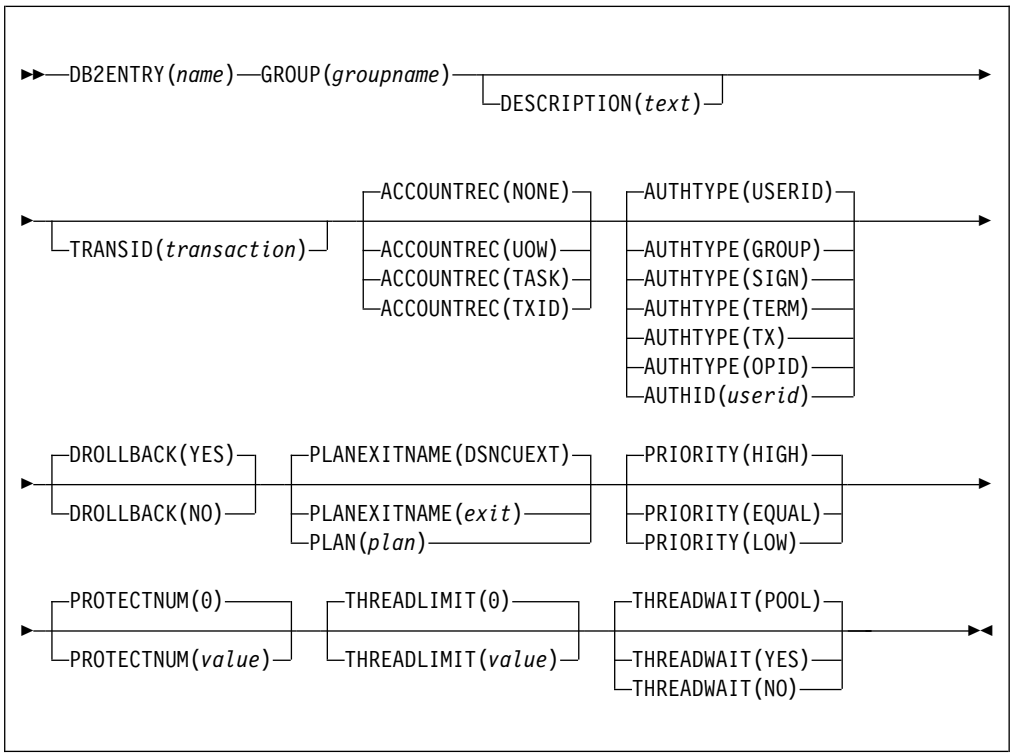
DB2ENTRY は、CICS Db2 接続機能が使用するエントリー・スレッド の属性を定義します。

トランザクション、あるいはトランザクション・グループを DB2ENTRY に関連付けることができます。トランザクション・グループは、1 つ以上のワイルドカード文字を使用して表すことができます (トランザクション ID のワイルドカード文字を参照)。また、DB2TRAN を定義することで、追加のトランザクションを DB2ENTRY に関連付けることもできます。

Db2 エントリー・リソース定義のインストールについては、Db2 エントリー・リソース定義のインストールを参照してください。BAS の作業については、BAS Db2 エントリー・リソース定義の作業を参照してください。

DB2ENTRY 属性

DB2ENTRY リソースの構文と属性について記述します。



一般属性

DB2ENTRY リソースの一般属性について説明します。

DB2ENTRY(*name*)

Db2 エントリー定義を識別する 1 から 8 文字の名前。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

DESCRIPTION(*text*)

このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)

各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字:
A-Z 0-9 $ @ #
小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

スレッド選択属性

スレッド選択に関連する DB2ENTRY リソースの属性について説明します。

TRANSID(*transaction*)

項目に関連したトランザクションIDを指定します。ここで指定できるトランザクションは 1 つのみです。ただし、TRANSID で 1 つ以上のワイルドカード文字を使用することで (43 ページの『トランザクション ID のワイルドカード文字』を参照)、トランザクションのグループを表すことができます。この DB2ENTRY を参照する DB2TRAN を定義することで、この DB2ENTRY に対して追加のトランザクションを定義することができます。 DB2ENTRY では、トランザクション ID はオプションです。代わりに DB2TRAN 定義を使用することで、すべてのトランザクションを DB2ENTRY に関連付けることができます。ただし、DB2ENTRY に関連付けるトランザクションが 1 つだけの場合、DB2ENTRY で指定するほうが容易です。

注: ここでトランザクション ID を指定すると、DB2ENTRY 定義のインストール時に暗黙的に DB2TRAN オブジェクトが作成されます。暗黙 DBTRAN の名前は、指定されたトランザクション ID に DFH が付いたものになります。これらの暗黙 DB2TRAN は、DB2TRAN リソースの CICS ビューおよび CICSPlex® SM ビューに表示されます。別のトランザクション ID を指定するように DB2ENTRY が変更された場合、DB2ENTRY が再インストールされると、以前の暗黙 DB2TRAN は削除され、DB2ENTRY に関連付けられた新規のトランザクション ID を表す新規の暗黙 DB2TRAN が作成されます。

スレッド操作属性

スレッド操作に関連する DB2ENTRY リソースの属性について説明します。

ACCOUNTREC({NONE|TASK|TXID|UOW})

この Db2 エントリーを使用するトランザクションに必要な Db2 アカウンティングの最小量を指定します。以下のオプションで説明するように、指定された最小量を超える場合があります。

NONE

この DB2ENTRY からのスレッドを使用するトランザクションに必要なアカウンティング・レコードはありません。

ただし、Db2は、スレッドの終了後に 各スレッドごとに1つ以上の会計レコード を生成します。 許可を変更すると、追加として レコードが生成されます。

TASK

CICS Db2 接続機能により、各 CICS タスクに対して最小で 1 つのアカウンティング・レコードが生成されます。

複数の作業単位 (UOW) が含まれ、スレッドが同期点で解放されるトランザクションでは、UOW ごとに異なるスレッドを使用することができます。その結果、UOW ごとにアカウンティング・レコードが生成される場合があります。

TXID スレッドを使用するTransIDが 変わったとき、CICS Db2接続機能により、 会計レコードが生成されます。

このオプションは、複数の トランザクションIDが使用するDb2項目 定義に適用されます。通常、スレッドは同期点で解放されるため、複数の UOW が含まれるトランザクションでは、UOW ごとに異なるスレッドを使用する場合があります。その結果、UOW ごとにアカウンティング・レコードが生成される場合があります。

UOW CICS Db2 接続機能により、UOW ごとにアカウンティング・レコードが生成されます (スレッドが UOW の最後に解放されることを前提としています)。

AUTHID(*userid*)

この DB2ENTRY の使用時にセキュリティー検査に使用するユーザー ID を指定します。

Db2 アドレス・スペースのセキュリティー検査の一部または全部に RACF を使用している場合は、AUTHID を指定しないでください。代わりに、GROUP オプション、SIGN オプション、または USERID オプションを指定して AUTHTYPE を使用してください。 AUTHID を使用するスレッドは必要な RACF アクセス制御環境エレメント (ACEE) を Db2 に渡さないため、AUTHTYPE を使用する必要があります。 この ID は、最大 8 文字の長さで指定することができます。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

AUTHTYPE({USERID|OPID|GROUP|SIGN|TERM|TX})

この DB2ENTRY の使用時にセキュリティー検査に使用できる ID のタイプを指定します。

Db2 アドレス・スペースのセキュリティー検査の一部または全部に RACF を使用している場合は、GROUP オプション、SIGN オプション、または USERID オプションを使用します。これらのオプションを使用して定義されたスレッドのみが必要な RACF アクセス制御環境エレメント (ACEE) を Db2 に渡すため、これらのオプションのいずれかを使用する必要があります。ただし、SIGN オプションを指定すると、DB2CONN 定義で SIGNID 属性に指定された値が CICS 領域のユーザー ID と一致した場合にのみ ACEE が Db2 に渡されます。

Db2 内部セキュリティーのみを使用している場合は、ACEE は不要です。この場合、任意のオプションを使用することができます。

USERID

CICSトランザクションに関連した USERIDが、許可IDとして使用されます。 ユーザー ID の長さが 8 文字未満の場合、右側にブランクが埋め込まれます。

AUTHTYPE(USERID) を指定して Db2 サンプル・サインオン出口 DSN3@SGN を使用すると、この出口は、Db2 に 1 次許可 ID としてユーザー ID を、 2 次 ID として接続グループ名を送信します。サンプル・サインオン出口を使用すると、AUTHTYPE(USERID) と AUTHTYPE(GROUP) は同じになります。

OPID CICS トランザクション・サインオン機能に関連付けられたユーザー ID に関連付けられたオペレーター ID は、許可 ID として使用されます。3文字のオペレーター識別の右側に ブランクが埋め込まれて、8文字の許可 IDになります。

GROUP

1 から 8 文字の USERID、および許可 ID として接続グループ名を指定します。以下の表に、これらの 2 つの値が Db2 でどのように解釈されるかを示します。

Db2 へ渡される **ID**	**Db2** が値を解釈する方法
CICS サインオン・ユーザー ID (USERID)	1 次 Db2 許可 ID を表します。

Db2 へ渡される ID	Db2 が値を解釈する方法
RACF 接続グループ名	グループ・オプションの RACF リストがアクティブではない場合、Db2 は CICS 接続機能が提供する接続されたグループ名を 2 次 Db2 許可 ID として使用します。グループ・オプションの RACF リストがアクティブな場合、Db2 は CICS 接続機能が提供する接続されたグループ名を無視しますが、2 次 Db2 許可 ID の Db2 リストには値が表示されます。

GROUP オプションを使用するには、領域のシステム初期設定パラメーターで SEC=YES を指定する必要があります。

この USERID に対して使用できる RACF グループ ID がない場合、ブランクの 8 文字フィールドがグループ ID として Db2 に渡されます。

SIGN DB2CONNのSIGNID属性を、資源の 許可IDとして使用することを指定します。

TERM

端末 ID (4 文字に埋め込み文字が加えられて 8 文字になる) を許可 ID として指定します。端末がトランザクションに接続されていない場合、この方法で許可 ID を取得することはできません。

CICS コマンドを使用してトランザクションを開始しても、それに関連付けられた端末がない場合、AUTHTYPE(TERM) を使用できません。

TX トランザクション識別 を許可IDとして指定します。 4文字のトランザクション 識別の右側にブランクが 埋め込まれて、8文字の許可IDに なります。

DROLLBACK({YES|NO})

トランザクションがデッドロック解決の犠牲として選択された場合に、CICS Db2 接続機能が SYNCPOINT ROLLBACK を開始するかどうかを指定します。

YES 制御をアプリケーションに返す前に、接続機能が SYNCPOINT ROLLBACK を開始します。 SQL 戻りコードの -911 がプログラムに返されます。

エンタープライズ Bean を OTS トランザクションの一部として実行しているトランザクションが DB2ENTRY を使用する場合は、YES を指定しないでください。SYNCPOINT ROLLBACK は、OTS トランザクションでは許可されません。

NO 接続機能は、このトランザクションに対してロールバックを開始しません。 SQL 戻りコードの -913 がアプリケーションに戻されます。

PLAN(*plan*)

このエントリーに使用されるプランの名前を指定します。

PLANEXITNAME(DSNCUEXT|*exit*)

この Db2 エントリー定義に使用される動的プラン出口の名前を指定します。

Db2 エントリー定義に対してアクティブなトランザクションがある間に PLAN および PLANEXITNAME を変更した場合、次にトランザクションがスレッドを解放するときに新規値が適用されます。

PRIORITY({HIGH|EQUAL|LOW})
この DB2ENTRY のスレッド TCB の優先順位を、CICS のメイン TCB (QR TCB) との相対として指定します。CICS が DB2 バージョン 6 以降に接続されている場合、スレッド TCB は、CICS オープン L8 TCB です。

HIGH
スレッドTCBの優先順位は、CICS QR TCBより 高くなります。

EQUAL
スレッドTCBの優先順位は、 CICS QR TCBと同じです。

LOW スレッド TCB の優先順位は、CICS QR TCB より低くなります。

PROTECTNUM({0|*value*})
この Db2 エントリー定義で許可される保護スレッドの最大数を指定します。他の処理がキューに入っていないときにトランザクションによって解放されたスレッドは、保護する ことができます。これは、そのスレッドが即時に終了しないことを意味します。保護されたスレッドは、その間に再利用されなければ、パージ周期 2 回分の時間が経過した後に終了します。例えば、パージ周期が 30 秒に設定されている場合、保護されたスレッドは、解放後 30 秒から 60 秒の間に終了します (その間に再利用されなかった場合)。 CICS Db2接続機能が開始された後の最初の パージ周期は5分です。この後に、PURGECYCLE が適用されます。 スレッドが保護されるのは、そのスレッドが非アクティブの間だけです。トランザクションが保護された スレッドを再使用する場合、スレッドが アクティブになり、保護スレッドの現在の数が 減少します。

THREADLIMIT({0|*value*})
要求を待機させる、異常終了させる、あるいはプールに転送する前に、この Db2 エントリー定義に対して CICS Db2 接続がアクティブになることを許可するスレッドの最大数を指定します。

THREADWAIT({POOL|YES|NO})
アクティブな DB2ENTRY スレッドの数が THREADLIMIT 数に達した場合に、トランザクションが DB2ENTRY スレッドを待つか、異常終了するか、プールにオーバーフローするかを指定します。

POOL
すべてのスレッドが使用中の場合、 トランザクションの送信先が変更され、 スレッドのプールを使用します。プールも使用中であり、Db2 接続定義で THREADWAIT(NO) を指定している場合、トランザクションは異常終了コード AD3T で異常終了します。

NO すべてのスレッドが使用中の場合、トランザクションは異常終了 AD2P で異常終了します。

YES すべてのスレッドが使用中の場合、トランザクションはスレッドが使用可能になるまで待機します。

DB2TRAN リソース

DB2TRAN は、DB2ENTRY と関連付けられたトランザクション、またはトランザクションのグループを定義します。これらは、DB2ENTRY そのもので指定されているトランザクションに追加されたものです。

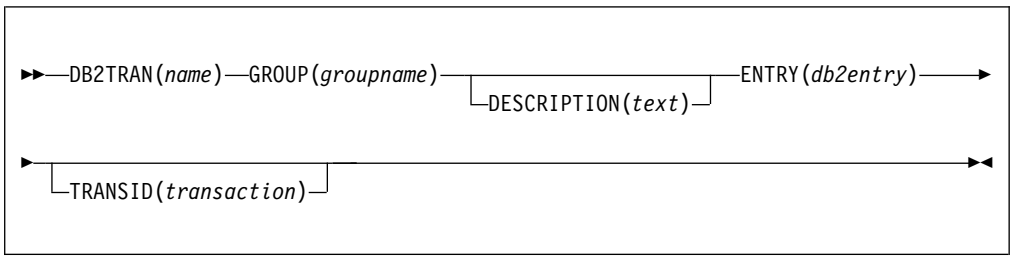
特定のトランザクション用にインストールできる DB2TRAN 定義は 1 つのみです。明示的に同じトランザクション ID を参照する 2 つ目の DB2TRAN 定義をインストールしようすると、そのインストールは失敗します。

DB2TRAN 定義により、DB2ENTRY に、ワイルドカード文字を使用する名前も含めて無制限にトランザクションを関連付けることができます。単一の DB2ENTRY に関連付ける DB2TRAN は、いくつでも定義できます。

DB2 トランザクション・リソース定義のインストールについては、Db2 トランザクション・リソース定義のインストールを参照してください。BAS の作業については、BAS Db2 トランザクション・リソース定義の作業を参照してください。

DB2TRAN 属性

DB2TRAN リソースの構文および属性について説明します。



DB2TRAN(*name*)
この Db2 トランザクション定義を識別する 1 から 8 文字の名前。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

ENTRY(*db2entry*)
この Db2 トランザクション定義が参照する Db2 エントリー定義の名前を指定します。これは、この追加トランザクションを関連付ける必要がある Db2 エントリー定義です。

GROUP(*groupname*)

各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

TRANSID(*transaction*)

エントリーに関連付けるトランザクション ID を指定します。 TRANSID が指定されない場合、デフォルトで Db2 トランザクション定義名の最初の 4 文字になります。トランザクション ID には、ワイルドカード文字を含めることができます (『トランザクション ID のワイルドカード文字』を参照)。

トランザクション ID のワイルドカード文字

DB2TRAN リソースを定義する場合、アスタリスク (*) 記号およびプラス (+) 記号をワイルドカード文字として使用することで、汎用トランザクション ID を指定することができます。

- アスタリスク (*) は、トランザクション名に追加したり、あるいは単独で使用したりして、「ワイルド」に指定される任意の値を生成します。名前の末尾にアスタリスクで指定されたトランザクション名は、トランザクション ID で指定されていない 0 から 3 文字を表します。例えば、『T*』 の TRansid は、トランザクション 『T』、『TA』、『TAB』、『TABE』 などを表します。
- 正符号 (+) は、任意の位置で使用することができ、単一文字を表します。
- 単独のアスタリスクは任意のトランザクションを表し、代替プール定義として機能させることが可能です。これは、スレッド割り振りを使い果たした場合に、プールにオーバーフローする DB2ENTRY の追加属性が存在するという点でプールとは異なります。

突き合わせの規則には、最も具体的な突き合わせを行うということがあります。例えば、トランザクション FRED は、汎用トランザクション ID 『F*』 を指定する DB2ENTRY(2) ではなく、汎用トランザクション ID 『FRE*』 を指定する DB2ENTRY(1) を使用します。さらに、'+' は '*' よりも具体的です。例えば、『FRE+』 は 『FRE*』 よりも具体的です。

AUTHTYPE(TX) を指定した場合、実際の TXID が 1 次許可 ID として Db2 に渡されますが、ワイルドカード文字を使用した名前は渡されません。

ワイルドカードが含まれる汎用トランザクション ID を使用して DB2TRAN を定義した場合、INQUIRE DB2TRAN コマンドは、汎用トランザクション ID に一致する個々のトランザクションを識別することができないので注意してください。例えば、次のコマンドを発行します。

```
CEMT INQUIRE DB2TRAN(*) TRANSID(ABCD)
```

これにより、トランザクションABCDが関連付けられたDB2TRANの詳細が表示されます。しかし、トランザクション ID「ABC*」を使用して DB2TRAN が定義された場合、INQUIRE DB2TRAN コマンドは、DB2TRAN をトランザクション ID「ABCD」と一致させることができず、「非検出」応答を返します。

DOCTEMPLATE リソース

DOCTEMPLATE リソースは、文書テンプレート の属性を定義します。

文書テンプレートは、文書を構成する場合に使用される情報の単位です。文書テンプレートには、固定テキスト、およびアプリケーション・プログラムによって値が指定されるテキストを表すシンボルを含めることができます。文書テンプレートは、CICS アプリケーションにより作成することも、外部ソースから取得することもできます。詳しくは、CICS 文書および文書テンプレートを参照してください。

テンプレートは、以下のいずれかの場所に存在します。

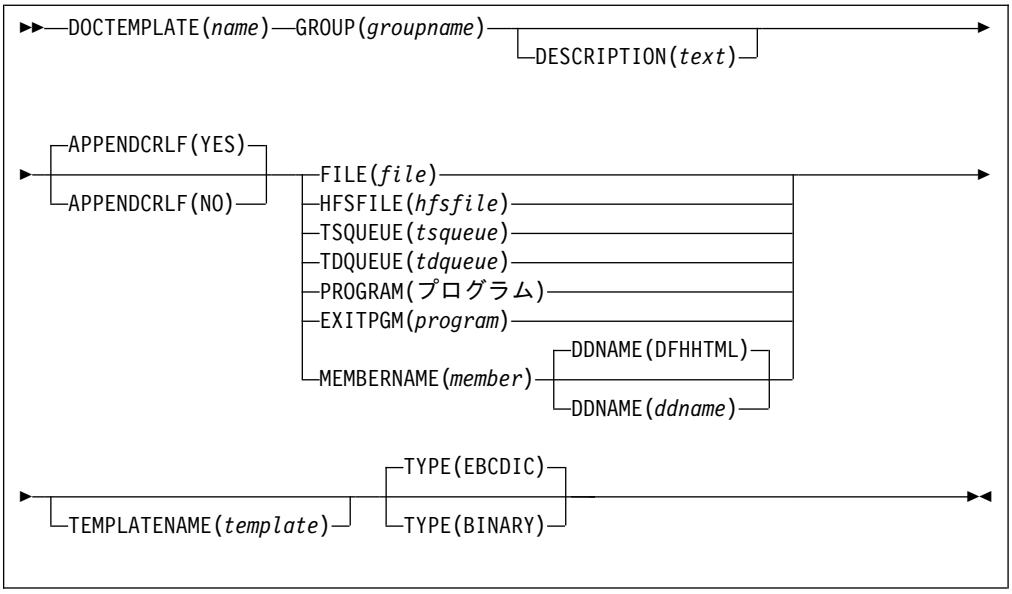
- MVS 区分データ・セット (DDNAME および MEMBERNAME 属性により指定)
- 一時ストレージ・キュー (TSQUEUE 属性により指定)
- 一時データ・キュー (TDQUEUE 属性により指定)
- CICS プログラム (PROGRAM 属性により指定)
- CICS ファイル (FILE 属性により指定)
- z/OS UNIX System Services ファイル (HFSFILE 属性により指定)

テンプレートは、出口プログラムにより戻されることもできます (EXITPGM 属性により指定)。

BAS の作業については、BAS 文書テンプレート・リソース定義の作業を参照してください。

DOCTEMPLATE 属性

DOCTEMPLATE リソースの構文および属性について説明します。



APPENDCRLF(YES|NO)
CICS がテンプレートの各論理レコードからの末尾ブランクの削除、および各論理レコードへの復帰改行の付加を行うかどうかを指定します。

DDNAME(DFHHTML|*ddname*)
テンプレートが MVS 区分データ・セット (PDS) 内にある場合は、PDS の DD 名を指定します。名前の最大長は 8 文字です。

許容文字:
A-Z 0-9 $ @ #
CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

MEMBERNAME 属性の値を指定し、DDNAME の値を指定しなかった場合、DFHHTML のデフォルト値が取られます。

この属性を指定する場合、
EXITPGM、FILE、HFSFILE、PROGRAM、TDQUEUE、および TSQUEUE は、いずれも指定することができません。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

DOCTEMPLATE(*name*)
この文書テンプレート定義の名前を指定します。名前の最大長は 8 文字です。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

EXITPGM(*program*)

テンプレートを生成する出口プログラムの名前を指定します。名前の最大長は 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

この属性を指定する場合、
DDNAME、FILE、HFSFILE、MEMBERNAME、PROGRAM、TDQUEUE、および TSQUEUE は、いずれも指定することができません。

FILE(*file*)

テンプレートが CICS ファイル内にある場合は、そのファイルの名前を指定します。 名前は8文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

この属性を指定する場合、
DDNAME、EXITPGM、HFSFILE、MEMBERNAME、PROGRAM、TDQUEUE、および TSQUEUE は、いずれも指定することができません。

GROUP(*groupname*)

各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

HFSFILE(*hfsfile*)

テンプレートが z/OS UNIX System Services ファイル内にある場合、これは、z/OS UNIX ファイルの完全修飾 (絶対)名または相対名を指定します。この名前は、すべてのディレクトリーが含まれ、スラッシュから始まる絶対名 (例えば、/u/facts/images/bluefish.jpg) として指定することができます。あるいは、CICS 領域ユーザー ID のホーム・ディレクトリーからの相対名 (例えば、facts/images/bluefish.jpg) として指定することもできます。最大 255 文字を使用できます。

許容文字:

A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

この属性を指定した場合、DDNAME, EXITPGM, MEMBERNAME, PROGRAM, TDQUEUE, またはTSQUEUEは指定できません。

注: CICS 領域には、z/OS UNIX へのアクセス権が必要であり、さらにこのファイルが保管されている z/OS UNIX ディレクトリーおよびファイルそのものへのアクセス権も必要です。 これらの権限を付与する方法について詳しくは、CICS 領域に対する z/OS UNIX ディレクトリーおよびファイルへのアクセス権限の付与を参照してください。

MEMBERNAME(*member*)

テンプレートが MVS 区分データ・セット (PDS) 内にある場合は、そのテンプレートが含まれるメンバーの名前を指定します。名前の最大長は 8 文字です。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

この属性を指定する場合、
EXITPGM、FILE、HFSFILE、PROGRAM、TDQUEUE、および TSQUEUE は、いずれも指定することができません。

PROGRAM(プログラム)

テンプレートが CICS プログラム内にある場合は、そのプログラムの名前を指定します。名前の最大長は 8 文字です。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

この属性を指定する場合、
DDNAME、EXITPGM、FILE、HFSFILE、MEMBERNAME、TDQUEUE、および TSQUEUE は、いずれも指定することができません。

TDQUEUE(*tdqueue*)

テンプレートが一時データ・キュー内にある場合は、そのキューの名前を指定します。名前の長さは最大 4 文字です。

許容文字:

A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

この属性を指定する場合、
DDNAME、EXITPGM、FILE、HFSFILE、MEMBERNAME、PROGRAM、および TSQUEUE は、いずれも指定することができません。

TEMPLATENAME(*template*)
テンプレートを使用するアプリケーション・プログラムが、そのテンプレートを認識するための名前を指定します。この名前は最大48文字の長さにできます。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

この属性に値を指定しない場合、DOCTEMPLATE属性が 使用され、右側がブランクで拡張されます。

TSQUEUE(*tsqueue*)
テンプレートが一時記憶域キュー内にある場合は、そのキューの名前を指定します。この名前は最大16文字の長さにできます。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

この属性を指定する場合、
DDNAME、EXITPGM、FILE、HFSFILE、MEMBERNAME、PROGRAM、および TDQUEUE は、いずれも指定することができません。

TYPE({**EBCDIC**|**BINARY**})
テンプレートの内容の形式を指定します。

BINARY
テンプレートをテンプレート・ライブラリーからロードするときに、そのテンプレートの内容の構文解析を実行しません。

EBCDIC
テンプレートをテンプレート・ライブラリーからロードするときに、その内容が EBCDIC テキストとして構文解析されます。

ENQMODEL リソース

ENQMODEL は、**EXEC CICS ENQ** コマンドおよび **EXEC CICS DEQ** コマンドがシスプレックス全体にわたる有効範囲を持つ名前付きのリソースを定義します。

CICS は ENQMODEL リソースを使用する際に z/OS グローバル・リソース逐次化を利用して、複数のアプリケーションが使用するリソースをシスプレックス全体に渡って保護します。単一の CICS 領域内のローカル・エンキューは、CICS アドレス・スペース内で管理されます。複数の CICS 領域に影響を及ぼす、シスプレックス全体にわたるエンキューは、z/OS グローバル・リソースの逐次化によって管理されます。グローバル・リソースの逐次化について詳しくは、z/OS MVS 計画: グローバル・リソース逐次化を参照してください。

EXEC CICS ENQ および **EXEC CICS DEQ** コマンドがリソースに対して発行されると、CICS は、一致するインストール済みの ENQMODEL 定義の有無を確認します。ENQMODEL リソースの ENQSCOPE 属性は、同じエンキュー有効範囲を共用する領域セットを定義します。 ENQSCOPE 属性をブランク (デフォルト値) のままにした場合、CICS は、一致するすべてのエンキュー要求あるいはデキュー要求を、発行元の CICS 領域のローカル要求として処理します。ENQSCOPE が非ブランク

の場合には、CICS は、エンキューあるいはデキューをシスプレックス全体にわたるものとして扱い、キュー名およびリソース名を z/OS グローバル・リソースの逐次化に渡し、エンキューを管理します。

z/OS グローバル・リソースの逐次化は、リング構成またはスター型構成で構成することができます。 パフォーマンス上の理由から、 多重システム環境での実動領域にリング構成は お勧めしません。また、CICS 初期設定パラメーター **NQRNL** を使用して、z/OS グローバル・リソースの逐次化が CICS エンキュー要求に対して RNL 処理を使用するかどうかを指定することもできます。これは、リソースのスコープに影響します。 CICS 用の z/OS グローバル・リソースの逐次化の構成について詳しくは、グローバル CICS エンキューおよびデキュー: パフォーマンスの向上を参照してください。

DISPLAY GRS コマンドを使用して、z/OS グローバル・リソースの逐次化機能を使用しているシスプレックス・エンキューを表示することができます。例えば、次のようにします。

```
D  GRS,RES=(DFHEqname|*,[  rname|,*])
```

ここで、

qname
は、ENQMODEL リソースによって定義された 4 文字の ENQSCOPE を指定します。

rname
は、ENQMODEL リソースによって定義された ENQNAME を指定します。

シスプレックス全体にわたるエンキューはリソースに対してのみサポートされ、アドレス上のエンキューにはサポートされていません。この機能は、リカバリー可能リソースのロックを目的としているわけではありません。

CEMT および CEDA を使用したエンキュー・モデル・リソース定義のインストールについては、エンキュー・モデル・リソース定義のインストールを参照してください。BAS の作業については、BAS エンキュー・モデル・リソース定義の作業を参照してください。

ENQMODEL リソースへのアクセス

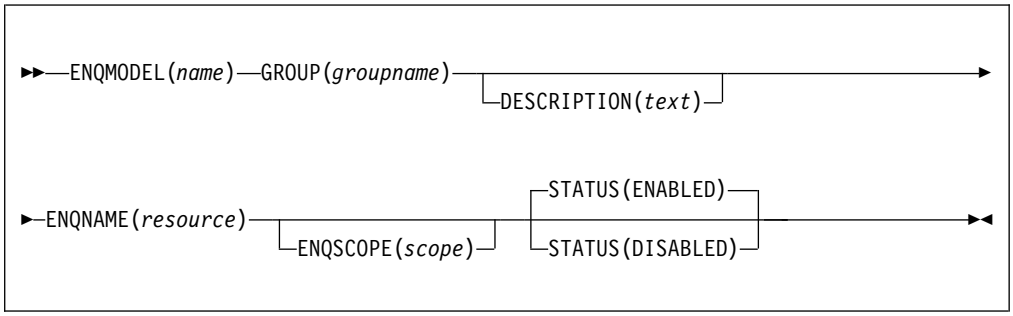
シスプレックス全体にわたるエンキューまたはデキュー機能を使用する必要のある CICS 領域には、必要な ENQMODEL リソースが定義され、インストールされていなければなりません。これを確実に行うために、CICS 領域が CSD を共用し、初期設定グループ・リストが同一の ENQMODEL グループを組み込む方法をお勧めします。

適切な ENQMODEL リソースを定義すると、アプリケーションはシスプレックス・エンキューを使用します。アプリケーション・プログラムを変更する必要はありません。リソース名が動的に構成されるために事前に分からないアプリケーションの場合、エンキュー EXEC インターフェース・プログラム出口 XNQEREQ と XNQEREQC を使用して、リソース名の先頭に、該当する ENQMODEL リソース定義と一致する文字を指定できます。これらのユーザー出口について詳しくは、エ

ンキュー EXEC インターフェース・プログラム出口 XNQEREQ および XNQEREQCを参照してください。

ENQMODEL 属性

ENQMODEL リソースの構文および属性について説明します。



DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

ENQMODEL(*name*)
この ENQMODEL 定義の名前を指定します。名前の最大長は 8 文字です。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

この名前は、CSDファイルでENQMODEL定義を識別 するために使用します。アクティブな CICS システム内では使用されません。

ENQNAME(*resource*)
1 から 255 文字のリソース名を指定します。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

最後の文字に * (アスタリスク) を使用することで、総称名であることを示すこともできます。

ENQSCOPE(*scope*)
オプションの 4 文字のエンキュー・モデル・スコープ名を指定します。省略されるかブランクとして指定 された場合、一致するエンキュー・モデルのスコープは ローカルになります。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字:

A-Z 0-9 $ @ #

小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

STATUS({ENABLED|DISABLED})

エンキュー・モデルを ENABLED 状況でインストールするか DISABLED 状況でインストールするかを指定します。 ENABLEDがデフォルトです。

ENABLED

一致するエンキュー要求は通常の 方法で処理されます。

DISABLED

一致するエンキュー要求はリジェクト され、発行元タスクは異常終了します。マッチング INSTALL CREATE および DISCARD 要求は、処理されます。

FEPI リソース

FEPI リソースでは、ノード・リスト定義、プール定義、プロパティー・セット定義、およびターゲット・リスト定義が使用されます。

FEPI ノード・リスト定義は、FEPI ノードの物理特性および操作特性を記述します。BAS の作業については、FEPI ノード・リスト・リソース定義の作業を参照してください。

FEPI プール定義は、FEPI プールの物理特性および操作特性を記述します。BAS の作業については、FEPI プール・リソース定義の作業を参照してください。

FEPI プロパティー・セット定義は、FEPI プロパティー・セットの物理特性および操作特性を記述します。BAS の作業については、FEPI プロパティー・セット・リソース定義の作業を参照してください。

FEPI ターゲット・リスト定義は、FEPI ターゲットの物理特性および操作特性を記述します。BAS の作業については、FEPI ターゲット・リスト・リソース定義の作業を参照してください。

FILE リソース

FILE リソースは、ファイルの物理的および操作上の特性を定義します。

FILE 定義には、レコード特性、ファイルで実行可能な操作のタイプ、リカバリー属性、およびジャーナル対象の操作についての情報を提供する属性が含まれます。通常、CICS ファイルは物理データ・セットに対応しています。物理データ・セットは、使用する前に VSAM に定義されている必要があります。 CICS ファイルを使用することで、アプリケーションは以下のことが可能です。

- データ・セットのレコードに直接アクセスする
- データ・セットからロードされたデータ・テーブルのレコードにアクセスする

- 関係するデータ・セットのないカップリング・ファシリティー・データ・テーブルのレコードにアクセスする (CFDT ファイル定義に対して LOAD(NO) が指定されているため)

CICS ファイルに関連付けられた以下のリソースは、RDO を使用して管理することができます。
- VSAM ファイル (これには、CICS 保守、ユーザー保守、およびカップリング・ファシリティー・データ・テーブルを参照するファイルと、VSAM データ・セットを参照するファイルが含まれます)
- リモート VSAM ファイル
- リモート BDAM ファイル
- LSRPOOL リソース定義によって定義される VSAM ローカル共有リソース (LSR) プール

アクティブな CICS システムでファイルを使用するには、そのファイルの定義がシステムにインストールされている必要があります。 CICS ファイル制御は、インストールされた定義を使用して、ファイルへのアクセスが必要なときにそのファイルを見つけたり、ファイルを使用するタスク数のカウントを保持したり、処理統計を収集したり、その他のファイル制御情報を保持したりします。

CEMT および CEDA を使用した FILE リソース定義のインストールについては、FILE リソース定義のインストールを参照してください。BAS の作業については、BAS FILE リソース定義の作業を参照してください。

CICS バンドル内の FILE リソース

CICS バンドルを使用して、FILE リソース定義の作成、編集、およびインストールを実行できます。この方法で FILE リソースを作成する場合、CICS バンドルを使用することによって、そのリソースのライフサイクルを管理することができます。

CICS バンドル内の定義では、次のファイル・タイプがサポートされます。
- VSAM ファイル (これには、CICS 保守、ユーザー保守、およびカップリング・ファシリティー・データ・テーブルを参照するファイルと、VSAM データ・セットを参照するファイルが含まれます)
- リモート VSAM ファイル
- リモート BDAM ファイル

動的に生成される FILE リソースの初期状況は、そのリソースを定義するバンドルの初期状況から導出されます。そのため、FILE リソースの STATUS に UNENABLED を定義しても、アプリケーションがそれらのファイルを暗黙的に開くことを禁止することはできません。

EXEC CICS INQUIRE FILE コマンドまたは **CEMT INQUIRE FILE** コマンドを使用することにより、CICS バンドルによって動的に生成される FILE リソースを照会できます。

CICS バンドルによって動的に生成される FILE リソースに対して **DISCARD** コマンドを発行することはできません。 CICS バンドルを破棄する必要があります。そうするなら、CICS により JVMSERVER リソースに操作が適用されます。

CICS バンドルの中で定義され、インストールされた FILE リソースの属性に変更を加えるには、CICS Explorer のリソース・エディターを使用することにより、CICS バンドル内の定義に変更を加え、CICS バンドルまたはそのデプロイ時に使用されたアプリケーションの新しいバージョンをインストールします。動的に生成されたリソースの属性を変更するために **SET FILE** コマンドを使用することは可能ですが、変更内容はカタログされず、CICS のウォーム・リスタートの後に回復されません。

CICS バンドル内で定義され、インストールされている FILE リソースの状況を変更するには、CICS バンドルまたはそのデプロイに使用されるアプリケーションの状況を変更してください。 FILE リソースを定義する CICS バンドルを使用不可にすると問題が発生する場合は、動的に生成されたリソースに対して **EXEC CICS SET FILE** コマンドまたは **CEMT SET FILE** コマンドを使用することができます (このアクションが必要な場合)。 アプリケーション・エラーの診断のトラブルシューティング手順に従って問題を診断し、適切なアクションを実行してください。

CICS バンドル内のリソースの定義について詳しくは、Defining CICS bundlesを参照してください。

リモート・ファイル

複数の CICS システムが接続されている場合、各システムは相互にファイルを共用することができます。そのようなファイルは、他のシステムに属しているファイルも含めて、すべて CICS に定義する必要があります。他のシステム上のファイルは、リモート として定義されます。リモート・ファイルには、ファイルを所有するリモート領域へのファイル制御要求の CICS 機能シップを介してアクセスします。

ただし、これは RLS モードでアクセスされるファイルや、常にローカル・ファイルとして定義されるカップリング・ファシリティー・データ・テーブルには適用されません。

CICSが必要とするリモート・ファイルの資源属性は、使用されるアクセス方式に固有ではありません。そのためRDOを使用してリモートBDAMファイルとリモートVSAMファイルの両方を定義できます。

REMOTESYSTEMを指定すると、REMOTENAMEも入力できます。これはリモート・システムで使用されるファイルの名前です。

ファイル定義がインストールされたCICS領域のSYSIDNTに対応するREMOTESYSTEM名を指定した場合、CICSはローカル・ファイルとして定義をインストールします。それ以外の場合、CICSはリモート・ファイルとして定義をインストールします。

ファイル定義がリモート・ファイルとしてインストールされた場合、以下の属性のみが使用されます。

REMOTESYSTEM
REMOTENAME
RECORDSIZE
KEYLENGTH

同じファイル定義がローカル定義としてインストールされた場合は、他の属性が使用されます。詳しくは、機能シップのためのリモート・リソースの定義を参照してください。

カップリング・ファシリティー・データ・テーブル

カップリング・ファシリティー・データ・テーブルのサポートは、ファイル所有領域や VSAM RLS サポートを必要とせずに、CICS ファイル制御を使用してファイル・データを共用する方法を提供します。データは、カップリング・ファシリティー・リスト構造内の、多くの点で共用ユーザー保守データ・テーブルに類似したテーブルに保持されます。

ユーザー保守データ・テーブルとは異なり、カップリング・ファシリティー・データ・テーブルはソース・データ・セットからプリロードする必要はありません。カップリング・ファシリティー・データ・テーブルのロードは、ファイル資源定義のDSNAMEおよびLOAD属性によって制御されます。このロードにより、CFDTを使用するアプリケーション・プログラムが、 LOAD(NO)を指定して完全にCFDTにデータを取り込むことができます。

LOAD(YES)とDSNAME属性の使用方法によって、以下のようなさまざまな方法でCFDTのロードを制御できます。

1. すべての CICS 領域がカップリング・ファシリティー・データ・テーブルをロードすることができます。 どのCICS領域がオープンを実行したかに関わらず、最初にCFDT用に ファイルを開くとテーブルがロードされます。データ・セットは、 ロードを行うCICSによって読み取り専用で開かれます。テーブルの すべてのファイル定義がLOAD(YES)およびソース・データ・セットの DSNAMEを指定します。この方法を使用する場合、各ファイル定義で 同じデータ・セットを指定してください。そのようにしなければ、 最初の定義で開かれるように指定されたデータ・セットがCFDTに ロードされます。CICSは、同一のCFDTを参照するすべての ファイルでDSNAMEが同じであることを検証しません。
2. 1つのCICS領域に、カップリング・ファシリティー・データ・テーブル のロードを担当させることができます。ロードを担当する領域には、 LOAD(YES)と、CICSのロードによって読み取り専用で開かれたデータ・ セットのDSNAMEを指定する、CFDTのファイル定義が含まれ ます。その他のCICS領域は、ソース・データ・セットに アクセスする必要はありませんが、ロードを担当する領域が CFDTを開くまで、CFDTを開くことはできません。ロードを 担当しない領域のCFDTのファイル定義でもLOAD(YES)を指定する 必要はありますが、DSNAMEは省略します。

   同じカップリング・ファシリティー・データ・テーブルを参照する複数のファイル名を使用してロードされるまで、カップリング・ファシリティー・データ・テーブルへのアクセスを制限することができます。 この方法でアクセスを制御するには、以下の操作を実行します。

   - LOAD(YES) および DSNAME(*datasetname*) を指定することで、データ・テーブルのロードが可能なファイル名を 1 つだけ定義します。このファイル名は、アプリケーション・プログラムから参照しないでください。
   - アプリケーション・プログラムが使用する、別のファイルを 定義し、このファイル定義がテーブルのロードを開始できない ようにしてください。

LOAD(YES)を指定しますが、データ・ セット名がDSNAME属性で指定されていないこと、および このファイルにDDステートメントがないことを確認してください。

- アプリケーション・プログラムがデータ・テーブルへの アクセスを必要とする前に、ロード可能なファイルを開く ようにしてください。例えば、OPENTIME(STARTUP)によって テーブルをロードするファイル名を定義して、CICS初期化の 終了時近くにファイルが自動的に開かれるようにして ください。
- アプリケーション・プログラムが、データをロードしない ファイル名を参照して、データ・テーブルにアクセスする ようにします。

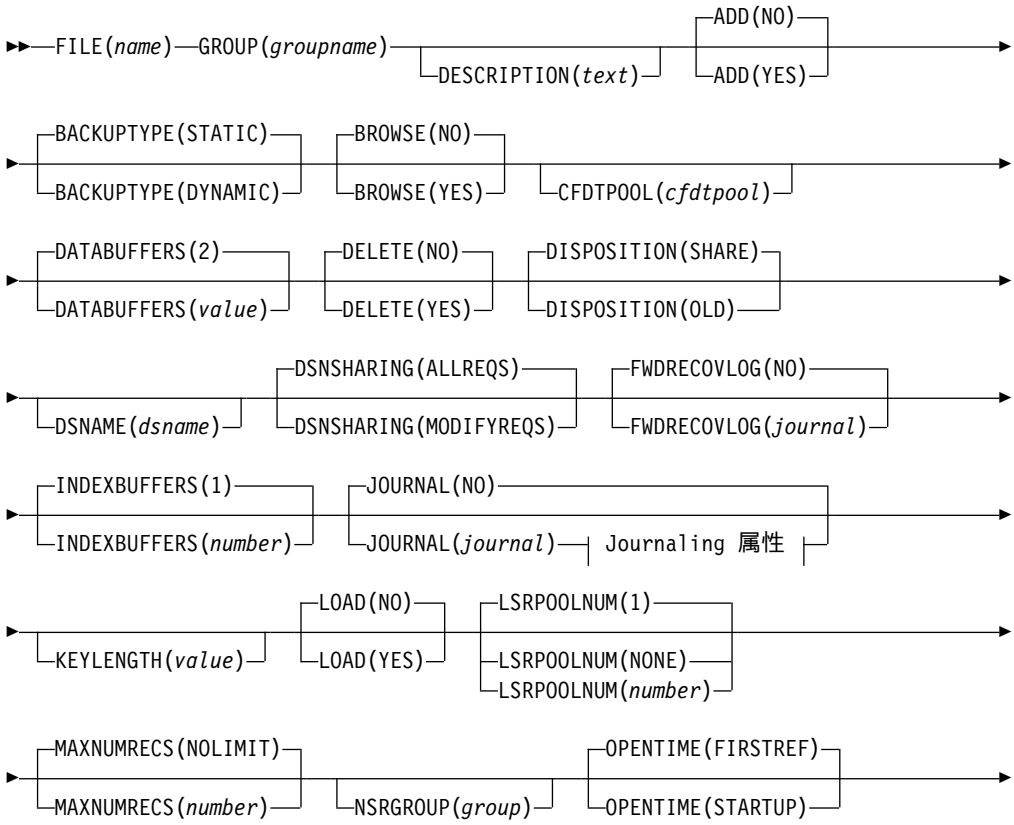
3. 上記のアプローチのうちの 2 つを組み合わせて使用することができます。いくつかの CICS 領域はテーブルをロードすることができ、他の CICS 領域は代わりにロードしてもらう必要があります。

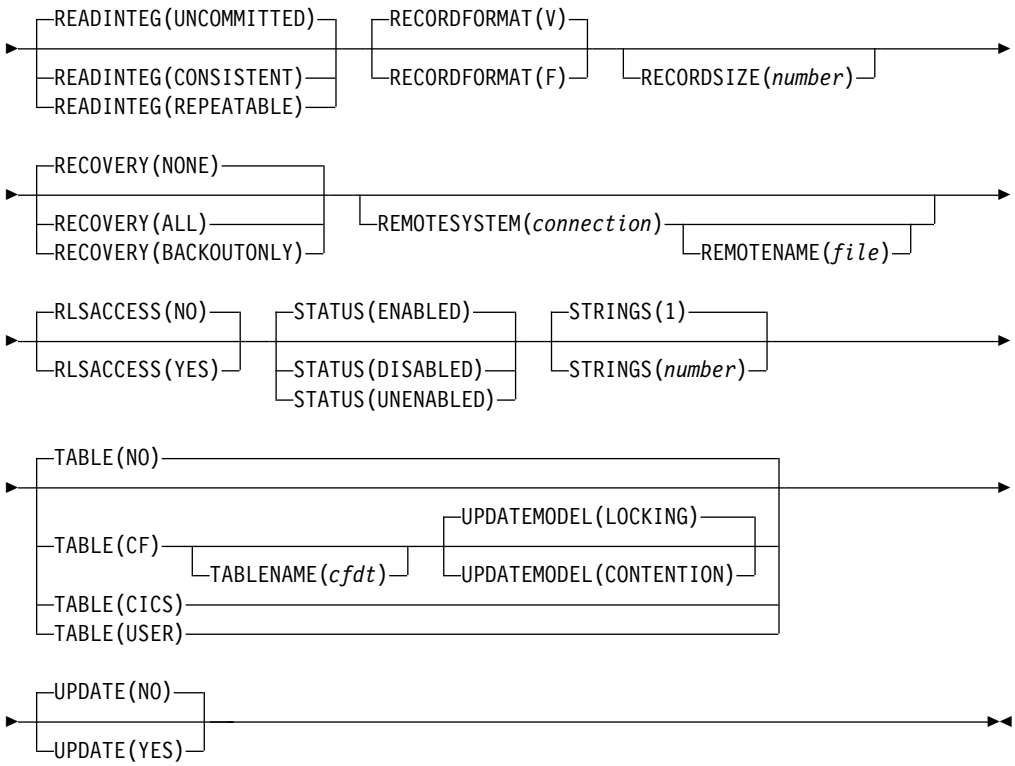
共用データ・テーブル

CICS が保守するデータ・テーブルとユーザーが保守するデータ・テーブルは、まとめて共用データ・テーブル と呼ばれます。

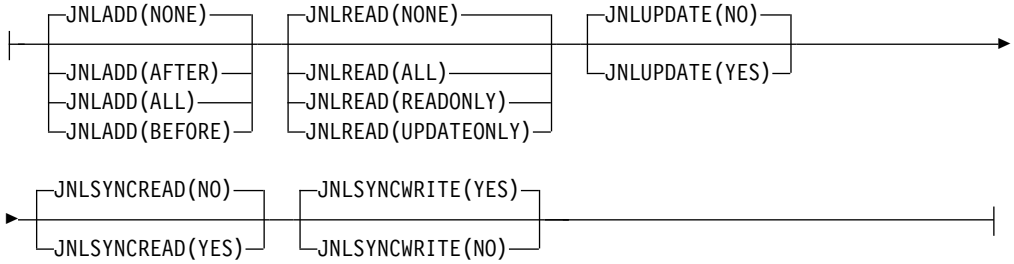
FILE 属性

FILE リソースの構文と属性について説明します。





Journaling 属性:



ADD({**NO**|**YES**})
ファイルにレコードを追加できるかどうかを指定します。

BACKUPTYPE({**STATIC**|**DYNAMIC**})
CICS VSAM ファイルは、更新のためオープンされている間はバックアップ可能として定義することができます。この属性は、RLSACCESS(YES) で定義されたファイルには使用されません。また、リカバリー・オプションが ICF カタログに定義されている場合にも使用されません。 CICS でこのカタログのリカバリー・オプションの代わりにこの属性を使用するには、**NONRLSRECOV** システム初期設定パラメーターに FILEDEF を設定します。 RLS モードでアクセスされるファイルでは、ICF カタログのデータ・セット定義にバックアップ・タイプを指定する必要があります。

この属性は、カップリング・ファシリティー・データ・テーブルでは無視されます。また、リカバリー属性がこのテーブルに関連するソース・データ・セットの ICF カタログで定義されている場合にも、これらの属性は無視されます。
CFDT は、オープン時バックアップ (BWO) は使用できません。

可能な値は以下のとおりです。

DYNAMIC
RECOVERY 属性に ALL を指定しながら DYNAMIC を指定して、更新のためオープンされている間ファイルをバックアップ可能にします。

STATIC
ファイルは、更新のためオープンされている間はバックアップできません。

BROWSE({**NO**|**YES**})
ファイルからレコードを順に取得できるかどうかを指定します。

CFDTPOOL(*cfdtpool*)
このファイル定義で定義されるテーブルを含むカップリング・ファシリティー・データ・テーブル・プールの名前を指定します。この属性は、TABLE(CF) を指定した場合に必要となります。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

カップリング・ファシリティー・データ・テーブルは、アカウンティング、セキュリティー、管理などの目的で、CFDT のグループ (プール) に分けることができます。すべてのカップリング・ファシリティー・データ・テーブル・プールの名前はシスプレックス内で固有でなければなりませんが、タイプが異なるプール (TS データ共用プールなど) の名前とは同じであっても問題ありません。

カップリング・ファシリティー・データ・テーブルを参照するファイルをオープンする場合には、指定したプールのカップリング・ファシリティー・データ・テーブル・サーバーをオープン要求を発行する MVS で実行する必要があります。必要なサーバーが開始されていないと、ファイル・オープン要求は失敗します。

注: CFDTPOOL 属性が意味を持つのは CFDT の場合のみです。TABLE(CF) として定義されていないファイルのプール名を指定できますが、それは CICS により無視されます。カップリング・ファシリティー・データ・テーブルを参照するようにファイル定義を変更すると、CFDTPOOL 名は有効になります。

DATABUFFERS({**2**|*value*})
データに使用されるバッファー数を指定します。範囲が 2 (デフォルト) から 32767 の値を使用します。指定可能な最小値は、STRINGS 属性で定義されているストリング数より 1 多い値です。

DELETE({**NO**|**YES**})
ファイルからレコードを削除できるかどうかを指定します。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。

CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

DISPOSITION({**SHARE**|**OLD**})
このファイルの属性指定を指定します。

OLD JCL の DISP=OLD パラメーターと同等です。

SHARE
JCL の DISP=SHR パラメーターと同等です。

DSNAME(*dsname*)
このファイルに使用されるデータ・セット名 (オペレーティング・システムに認識される名前) を指定します。DSNAME は、1 文字以上 44 文字以下で、MVS データ・セット名の規則に従います (「z/OS MVS JCL 解説書」の DSNAME パラメーターを参照してください)。

許容文字:
A-Z 0-9 $ @ # . -

ファイル・オープン時に、このファイル用の JCL ステートメントが存在しない場合は、オープンに先立ち、この DSNAME を使用するファイルの動的割り振りが行われます。ファイル定義がデータ・テーブル (CICS、USER、または CF) を参照する場合、DSNAME は VSAM ベースの KSDS である必要があります。パスまたは代替索引データ・セットを指定することはできません。

CICS 開始 JCL でこのファイルの DD ステートメントに指定された DSNAME は、このファイル定義で指定された DSNAME に優先します。

カップリング・ファシリティー・データ・テーブル
ファイル定義で LOAD(YES) が指定されていて、ファイルがまだオープンされていない場合には、DSNAME によってテーブルのロード元のソース・データ・セットの名前が指定されます。また、CICS 開始 JCL での DD ステートメントにソース・データ・セットを指定することもできます。指定するデータ・セットは、VSAM ベースの KSDS である必要があります。

ソース・データ・セットに関連付けられているパスまたは代替索引がある場合、ファイルを介する更新は、ソース・データ・セット、またはその関連付けられた代替索引のどちらにおいても反映されません。カップリング・ファシリティー・データ・テーブルは、ロードが完了した後、そのソース・データ・セットから完全に独立しています。

異なるファイル定義で指定された別のファイルのオープンによってテーブルのロードを開始する場合には、この属性を省略します。この場合には、CICS JCL の DD ステートメントでもファイル名が指定されていないことを確認します。ファイルをオープンする試みは、CFDT のロードが開始されるまで失敗します。データ・セットからのカップリング・ファシリティー・データ・テーブルのロードについて詳しくは、カップリング・ファシリティー・データ・テーブルを参照してください。

LOAD(NO) を指定した場合には、この属性は必要なく、無視されます。

DSNSHARING({**ALLREQS**|**MODIFYREQS**})

VSAM データ・セット名共用を VSAM ファイルに対して使用するかどうかを指定します。有効な値は、以下のとおりです。

ALLREQS

データ・セット名共用は、ファイルがオープンされるとき ACB にセットされるので、すべてのファイル要求に対して使用されます。

MODIFYREQS

データ・セット名共用は、ファイルがオープンされたときに ACB にセットされますが、このファイルに対して、DELETE、ADD、または UPDATE の操作が設定された場合に限ります。

FILE(*name*)

ファイルの名前を指定します。名前の長さは、最大で 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

この名前の先頭には数字を使用できません。

FWDRECOVLOG({**NO**|*journal*})

順方向リカバリーに使用される MVS システム・ロガーのログ・ストリームに対応するジャーナルを指定します。

この属性は、カップリング・ファシリティー・データ・テーブルでは無視されます。また、リカバリー属性がこのテーブルに関連するソース・データ・セットの ICF カタログで定義されている場合にも、これらの属性は無視されます。CFDT は順方向にリカバリーできません。

NO このファイルに、順方向リカバリー・ロギングは必要ありません。

journal

CICS が順方向リカバリー・ログに使用するジャーナルを識別する番号。CICS ジャーナル名の形式は DFHJ*nn* で、*nn* の範囲は 01 から 99 です。順方向リカバリーの変更後イメージは、ジャーナル名 DFHJ*nn* に対応する MVS ログ・ストリームに書き込まれます。

注: CICS Transaction Server for z/OS では、DFHJ01 はシステム・ログではありません。

この属性が CICS により使用されるのは、以下の条件を満たした場合のみです。

- RECOVERY(ALL) が指定されている
- RLSACCESS(NO) が指定されている
- ICF カタログでリカバリー属性が定義されていない

対応するデータ・セットの ICF カタログ項目でファイルのリカバリー属性を定義すると、CICS は常に ICF カタログのリカバリー属性を使用し、FILE リソースにあるこれらの属性を無視します。CICS でこのカタログのリカバリー・オプションの代わりに FILE リソース属性を使用するには、**NONRLSRECOV** システム

初期設定パラメーターに FILEDEF を設定します。IDCAMS ALTER コマンドを使用することにより、ICF カタログで定義されたリカバリー属性を変更できます。データ・セット用にオープンされた ACB がある間、この操作が妨げられることはありません。ただし、リカバリー属性を変更する場合には、データ保全性に影響を与える可能性があるので注意してください。

CICS は、データ・セットに対する連続したオープン要求の最初の更新オープン時に、ICF カタログからデータ・セットのリカバリー属性をコピーします。これは、単一 CICS 領域はリカバリー属性の更新によって影響されないことを意味します。ただし、データ・セットが RLS モードでオープンされ、かつ ICF カタログ上の属性が修正された場合には、2 番目の CICS 領域が同じデータ・セットを更新のためにオープンし、異なる属性のセットをコピーする可能性があるので、データ保全性が脅かされることになります。

ICF カタログで定義されているリカバリー属性を変更する必要がある場合 (例えば、順方向リカバリー・ログ・ストリーム名を変更する場合) は、変更を行う前にデータ・セットを静止します。データ・セットを静止すると、ユーザーが変更を終えて、データ・セットを静止解除するまで、RLS モードでの使用ができなくなります。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

INDEXBUFFERS({**1**|*number*})
索引に使用されるバッファー数を指定します。範囲が 1 から 32767 の値を使用します。指定可能な最小値は、STRINGS 属性で定義したストリング数です。

JNLADD({**NONE**|**BEFORE**|**AFTER**|**ALL**})
JOURNAL 属性で指定したジャーナルに記録する追加操作を指定します。可能な値は以下のとおりです。

AFTER
VSAM 入出力操作の後に、ジャーナルへファイル制御書き込み操作を追加します。

ALL VSAM 入出力操作の前と完了後に、ジャーナルへファイル制御書き込み操作を追加します。

BEFORE
VSAM 入出力操作の前に、ジャーナルへファイル制御書き込み操作を追加します。

NONE
ジャーナルに書き込み操作を追加しません。

JNLREAD({NONE|UPDATEONLY|READONLY|ALL})
JOURNAL 属性で指定したジャーナルに記録する読み取り操作を指定します。可能な値は以下のとおりです。

ALL ジャーナルにすべての読み取り操作を追加します。

NONE
ジャーナルに読み取り操作を追加しません。

READONLY
ジャーナルに READ ONLY 操作のみ追加します (READ UPDATE 操作は追加しません)。

UPDATEONLY
ジャーナルに READ UPDATE 操作のみ追加します (READ ONLY 操作は追加しません)。

JNLSYNCREAD({NO|YES})
JOURNAL で指定されたジャーナルへの READ 操作に関して書き込まれる自動ジャーナル・レコードを、同期または非同期のどちらで書き込むかを指定します。

JNLSYNCWRITE({YES|NO})
JOURNAL で指定されたジャーナルへの WRITE 操作に関して書き込まれる自動ジャーナル・レコードを、同期または非同期のどちらで書き込むかを指定します。

JNLUPDATE({NO|YES})
JOURNAL 属性で指定したジャーナルに記録する REWRITE および DELETE 操作を指定します。

JOURNAL({NO|*journal*})
このファイルに関して自動ジャーナリングを行うかどうかを指定します。ジャーナル処理されたデータは VSAM レコード形式であり、ユーザーが制御するジャーナリングに使用されます。

ジャーナル処理されるデータは、
JNLADD、JNLREAD、JNLSYNCREAD、JNLSYNCWRITE、および
JNLUPDATE の各属性で識別されます。

CICS 保守データ・テーブルの場合は、結果として VSAM 入出力要求を生成する要求についてのみジャーナリングが実行されます。

ユーザー保守データ・テーブルまたはカップリング・ファシリティー・データ・テーブルの場合は、ジャーナリングはいかなるファイル制御操作に対しても実行されません。しかし、これらのテーブルの自動ジャーナリングがサポートされていないにもかかわらず、ジャーナル番号を指定すると、CICS はファイルをオープンするときに指定されたジャーナルのログ・ストリームをオープンしようとします。

注**:** 自動ジャーナリングは、RECOVERY 属性および FWDRECOVLOG 属性で指定された、システムおよび順方向リカバリー・ログに対するロギングとは関係ありません。
可能な値は以下のとおりです。

NO このファイルに対して自動ジャーナリングは行われません。

数値　CICS が自動ジャーナルに使用するジャーナルを識別する番号。CICS ジャーナル名の形式は DFHJ*nn* で、*nn* の範囲は 01 から 99 です。

注: CICS Transaction Server for z/OS では、DFHJ01 はシステム・ログではありません。

KEYLENGTH(*value*)

リモート・ファイル内、および LOAD(NO) で指定されたカップリング・ファシリティー・データ・テーブル内のレコードの論理キーの長さをバイト単位で指定します。

ここで KEYLENGTH を定義しない場合、このファイルを参照するアプリケーション・プログラムのファイル制御コマンドで KEYLENGTH オプションを指定する必要があります。 KEYLENGTH がここで定義されず、またアプリケーション・プログラムでも指定されない場合、キーが 4 文字より長いときのデフォルト値は 4 です。

リモート・ファイル

キーの長さの範囲は 1 から 255 です。

カップリング・ファシリティー・データ・テーブル

キーの長さの範囲は 1 から 16 です。キーの長さは、LOAD(NO) を指定した場合のみ必要です。

オプションで、LOAD(YES) が指定されたカップリング・ファシリティー・データ・テーブルに対してもキーの長さを指定できます。その場合には、次の点に注意してください。

- テーブルのロード元であるデータ・セットの ICF カタログ項目からキーの長さを取得します。キーの長さを指定する場合、キーの長さは、ソース・データ・セットで指定した長さと一致する必要があります。一致しない場合、ファイルをオープンする試みは失敗し、エラー・メッセージが表示されます。
- ファイルのオープン時に CICS が既に作成済みの CFDT を検出し、そのキーの長さがデータ・セットのロード時に使用した長さと異なる場合に、オープンは失敗します。

リモート・ファイルでない、または CFDT を参照しないファイルのキーの長さを指定しても、そのファイルをリモート・ファイルとして再定義したり、CFDT を参照するように再定義したりしない限り、その長さは有効となりません。ただし、キーの長さを指定した場合、INQUIRE FILE コマンドにより戻される値は次のようになります。

- ファイルがオープンしている場合、CICS は VSAM から取得した値を戻しますが、この値はファイル定義で指定された値とは異なる可能性があります。
- ファイルがクローズしている場合、CICS はファイル定義で指定されている値を戻します。

この属性の値は、シスプレックス全体で、同じカップリング・ファシリティー・データ・テーブルを参照するすべてのファイル定義において同一である必要があります。

LOAD({NO|YES})

最初にオープンしたときに、ソース・データ・セットからカップリング・ファシリティー・データ・テーブルをロードするかどうか指定します。

NO カップリング・ファシリティー・データ・テーブルをソース・データ・セットからロードする必要がないことを意味します。オープンするとすぐに、アプリケーション・プログラムで完全に使用できます。テーブルは、そのテーブルを使用するアプリケーション・プログラムによりロードされます。これは、カップリング・ファシリティー・データ・テーブルのデフォルト設定方式です。

YES カップリング・ファシリティー・データ・テーブルが完全に使用可能な状態になるには、ソース・データ・セットからロードする必要があることを意味します。このカップリング・ファシリティー・データ・テーブルを使用するアプリケーション・プログラムは、ソース・データ・セットからのレコードを含むカップリング・ファシリティー・データ・テーブルに依存します。ロードを完了しなければデータにアクセスできないというわけではありません。

この属性は、TABLE(CF) 属性を使用して定義されたファイルに対してのみ有用です。 TABLE(CF) として定義されてはいないファイルについて LOAD 属性を指定することは可能ですが、それは CICS により無視されます。 (CICS 保守テーブルとユーザー保守テーブルは、ソース・データ・セットから常に自動的にロードされます。) カップリング・ファシリティー・データ・テーブルを参照するようにファイル定義を変更すると、LOAD 属性は有効になります。

この属性の値が、シスプレックス全体で、同じカップリング・ファシリティー・データ・テーブルを参照するすべてのファイル定義において同一であることを確認します。

この属性の使用について詳しくは、カップリング・ファシリティー・データ・テーブルを参照してください。

LSRPOOLID({1|2|3|4|5|6|7|8|NONE})

この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

既存の定義に指定されている LSRPOOLID の値は、新しいオプション LSRPOOLNUM に移されます。

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

ローカル共用リソース・プールの ID を指定します。NSRGROUP 属性の値が指定されている場合以外は、LSRPOOLNUM のデフォルト値は 1 です。NSRGROUP 属性に値が指定されている場合、LSRPOOLNUM のデフォルト値は NONE です。

なし

このファイルに関連するデータ・セットが、VSAM 非共用リソース (NSR) を使用することを指定します。

CICS 共用データ・テーブル (CICS 保守テーブルまたはユーザー保守テーブル) には、どちらのタイプのデータ・テーブルも LSR プールを使用する

必要があるため、NONE を指定することはできません。ただし、この制限は、NONE を指定できるカップリング・ファシリティー・データ・テーブルには適用されません。

VSAM 非共用リソース (NSR) は、トランザクション分離を使用するトランザクションに対してはサポートされていません。 NSR を使用して VSAM ファイルにアクセスするトランザクションを定義するときに、ISOLATE(NO) を指定します。ファイル要求をリモート領域へ機能シップすることもできます。その要求を実行する DFHMIRS プログラムは、CICS の EXECKEY を指定して定義できます。トランザクション分離がアクティブかどうかにかかわらず、CICS キー・プログラムには、固有のタスクおよび他のすべてのタスクの CICS キー・ストレージおよびユーザー・キー・ストレージへの読み取りおよび書き込みアクセスがあります。

number
このファイルに関連する VSAM データ・セットで使用される VSAM 共用リソース・プールの数を識別します。値の範囲は 1 から 255 までです。VSAM ローカル共用リソース (LSR) を使用するように、データ・セットを定義します。割り当てられた LRPOOLNUM 値に対応する LSRPOOL リソース定義に、バッファー、ストリング、および他のリソースを明示的に定義します。

デフォルトでは、ファイル定義で RLSACCESS(YES) を指定した場合、CICS がファイルをオープンするときに LSRPOOLNUM 値は無視されます。ただし、LSR プールを指定しているファイル定義を RLSACCESS(NO) から RLSACCESS(YES) に変更する場合には、LSRPOOLNUM 値を保持することをお勧めします。 LSRPOOLNUM を保持することにより、ファイルが RLS モードから LSR モードにいつ切り替えられても、ファイルは正しく LSR プールを参照することができます。

MAXNUMRECS({**NOLIMIT**|*number*})
データ・テーブルに収容できる最大レコード (項目) 数を指定します。この属性を使用して、ランナウェイ・トランザクションが次のストレージを使用しないようにします。

- テーブルがカップリング・ファシリティー・データ・テーブルの場合、サーバーのプール内のすべてのストレージ。
- テーブルが CICS 保守テーブルまたはユーザー保守テーブルの場合、MVS データ・スペース内のすべてのストレージ。

この属性が意味を持つのは、TABLE 属性に CICS、USER、または CF を使用して定義されたファイルのみです。TABLE(NO) で定義したファイルに対し MAXNUMRECS を指定できますが、有効にはなりません。データ・テーブルを参照するようにファイル定義を変更すると、MAXNUMRECS 値は有効になります。

NOLIMIT
テーブルに格納可能なレコードの数のユーザー指定制限はありません。CICS では、最大フルワード値として、限界値 2,147,483,647 が適用されます。

number
テーブルに格納できる最大レコード数を指定します。範囲は 1 から

99999999 です。リカバリー可能カップリング・ファシリティー・データ・テーブルに制限を設定する場合は、テーブルに格納する予定の最大レコード数よりも約 5% から 10% 大きい値を指定します。これにより、リカバリー可能要求を処理する際に内部で追加レコードを作成することが可能となります。この内部処理のために許可するマージンは、カップリング・ファシリティー・データ・テーブルの使用レベルと、その使用の性質によって異なります。この内部処理には、指定されている MAXNUMRECS の制限値よりも明らかにレコード数が少ないリカバリー可能カップリング・ファシリティー・データ・テーブルへの WRITE 要求または REWRITE 要求で、RESP2 が 102 の NOSPACE 条件が生じる可能性があります。

NSRGROUP(*group*)

同じ VSAM 基本データ・セットを参照するファイル定義をひとまとめにするシンボル名 (最大 8 文字) を指定します。この値は単なるシンボルであり、特定のファイル定義を参照する必要はありません。必要なのは、同じグループにまとめるファイル定義をすべて同じ名前にすることのみです。正常に処理されるためにこの属性を指定することが必須というわけではありませんが、この属性を指定しない場合、システムのパフォーマンスが低下する可能性があります。

許容文字:
A-Z 0-9 $ @ #
CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

NSRGROUP 属性は、VSAM 非共用リソースを使用するデータ・セットを参照するファイルに対してのみ有効です。データ・テーブルに対して NSRGROUP 属性をコーディングすることはできません。この属性は、データ・セット名共用という VSAM の概念と関係しています。データ・セット名共用では、同じ基本データ・セットに関連するすべてのファイルが必要とするストリングとバッファー用に単一の制御ブロック構造が VSAM により作成されます。

このようなファイル・グループの最初のメンバーをオープンするときに、グループ内のすべてのファイル項目に割り振られるストリングの総数を (アクセス制御ブロックの BSTRNO の値により) VSAM に指定する必要があります。オープンする最初のファイルがパスまたは基本データ・セットのどちらに関連付けられていたとしても、この時点で VSAM 制御ブロック構造が作成されます。BSTRNO の値はこの時点で計算され、同じ NSRGROUP 属性を持つすべてのファイル定義の STRINGS 値が加算されます。グループの最初のファイルがオープンされた後にグループに追加される新規ファイルは、既に作成された VSAM 制御ブロック構造に影響を与えません。基本クラスターに対してオープンされたすべてのファイルがクローズされてから再オープンされた場合にのみ、変更が加えられます。

CICS は、データ・セット名共用をすべての VSAM ファイルのデフォルトとします。DSNSHARING=MODIFYREQS により読み取り専用処理としてファイルをオープンすると、データ・セット名共用は無効となります。ただし、DSNSHARING=MODIFYREQS を指定したファイルは BSTRNO の計算に寄与します。

ユーザーが NSRGROUP 属性を指定せず、ファイルにより VSAM 非共用リソースが使用される場合、VSAM 制御ブロック構造が後続処理には不十分なストリングで作成される可能性があります。この状況が発生した場合、VSAM は動的ストリング追加機能を呼び出し、必要に応じてストリングの追加の制御ブロックを加えます。しかし、このメカニズムは非効率的であり、追加されたストレージは、CICS の実行が終わるまで解放されません。

VSAM ローカル共用リソースを使用するように指定されたファイル (LSRPOOLNUM=n、この n は 1 から 255 の範囲) では、NSRGROUP は無効です。

図 1 では、VSAM 基本データ・セットと代替索引パスに必要なファイル制御定義の指定方法の例を示します。

```
CEDA DEFINE FILE(VSAM10B)  GROUP(xxxxxx)
            DSNAME(DTGCAT.VSAM10B)
            DISPOSITION(SHARE)  ADD(YES)
            BROWSE(YES)  DELETE(YES)  READ(YES)
            UPDATE(NO)  RECORDFORMAT(F)
            STRINGS(8)  LSRPOOLNUM(NONE)
            RECOVERY(NONE)  NSRGROUP(GROUP1)
            INDEXBUFFERS(8)  DATABUFFERS(9)
CEDA DEFINE FILE(VSAM10P)  GROUP(xxxxxx)
            DSNAME(DTGCAT.VSAM10P)
            LSRPOOLNUM(NONE)  DISPOSITION(SHARE)
            STRINGS(5)  NSRGROUP(GROUP1)
            BROWSE(YES)  DELETE(NO)  READ(YES)
            ADD(NO)  UPDATE(NO)  RECORDFORMAT(F)
            RECOVERY(NONE)  INDEXBUFFERS(5)
            DATABUFFERS(6)
```

図 *1. VSAM* 基本データ・セットと代替索引パスの定義

OPENTIME({**FIRSTREF**|**STARTUP**})
ファイルがいつオープンするかを指定します。可能な値は以下のとおりです。

FIRSTREF
以下の方法のいずれかを使用することによってファイルのオープン要求が出されるまで、ファイルはクローズされたままです。

- 発信端末コマンド
- アプリケーション・プログラムでの EXEC CICS SET FILE OPEN コマンド
- 暗黙的なオープン

STARTUP
ファイルが閉じたままでファイルの状況が UNENABLED であるという場合を除き、ファイルは、自動的に開始された CICS トランザクション (CSFU) によって、CICS 初期化の直後に開かれます。

READ({**YES**|**NO**})
このファイルのレコードが読み取れるかどうかを指定します。

READINTEG({**UNCOMMITTED**|**CONSISTENT**|**REPEATABLE**})
RLSACCESS(YES) を指定して定義されたファイルに必要な読み取り保全性のレ

ベルを指定します。読み取り保全性は、非 RLS アクセス・モードのファイル、CICS 共用データ・テーブル、またはカップリング・ファシリティー・データ・テーブルには適用されません。

READINTEG を使用して、ファイルの読み取り保全性のデフォルト・レベルを設定できます。 READ、READNEXT、または READPREV の各コマンドで、API 読み取り保全性オプション UNCOMMITTED、CONSISTENT、または REPEATABLE のいずれも指定していないプログラムに、読み取り保全性のデフォルト・レベルを使用します。ただし、アプリケーション・プログラムがこれらのうちのいずれかを明示的に使用して、読み取り保全性を指定している場合には、その API オプションがこの READINTEG 属性に指定された値を上書きします。

注**:** 読み取り保全性オプションは、CICS ファイル制御 API コマンド、または CICS ファイル・リソース定義にのみ指定できます。CICS によってオープンされるファイルの DD ステートメントに、同等のパラメーターを使用することはできません。

ファイル・リソース定義で CONSISTENT または REPEATABLE を指定すると、これらのオプションが API で利用可能になる前に作成されたプログラムで読み取り保全性を利用可能にすることができ、しかもそれらのプログラムを修正する必要もありません。ただし、そうする場合には、一貫性のあるまたは反復可能な読み取りを実施すると、予期しないデッドロックが生じる可能性があることに注意してください。プログラムが LOCKED 状態に陥る可能性もあります。

CONSISTENT

レコードは、一貫性のある読み取り保全性レベルで読み取られます。別のトランザクションがレコードを修正している場合、READ 要求は更新が完了するまで待機しますが、そのタイミングは、データ・セットがリカバリー可能かリカバリー不能かによって異なります。

- リカバリー可能データ・セットの場合、更新トランザクションが次の同期点まで完了するか、またはロールバックを終了したときに、READ 要求は完了します。
- リカバリー不能データ・セットの場合、更新を実行する VSAM 要求が終了すると、ただちに READ 要求は完了します。

CONSISTENT が有効になるのは、RLSACCESS(YES) が指定されている場合のみです。非 RLS ファイルに CONSISTENT を指定すると、リソース定義は拒否されてエラーになります。

REPEATABLE

レコードは、反復可能な読み取り保全性レベルで読み取られます。別のトランザクションがレコードを修正している場合、READ 要求は更新が完了するまで待機しますが、そのタイミングは、データ・セットがリカバリー可能かリカバリー不能かによって異なります。

- リカバリー可能データ・セットの場合、更新トランザクションが次の同期点まで完了するか、またはロールバックを終了したときに、READ 要求は完了します。
- リカバリー不能データ・セットの場合、更新を実行する VSAM 要求が終了すると、ただちに READ 要求は完了します。

読み取りが完了した後、同期点まで共用ロックが保持されたままになります。これにより、タスクがさらに読み取り要求を実行しても、作業単位内でのレコード読み取りは修正できないことが保障されます。NOTFND などのエラー応答は、反復可能にはできません。

REPEATABLE が有効になるのは、RLSACCESS(YES) が指定されている場合のみです。非 RLS ファイルに REPEATABLE を指定すると、リソース定義は拒否されてエラーになります。

UNCOMMITTED
レコードは読み取り保全性なしで読み取られます。CICS は、VSAM に認識されているレコードの現行値を入手します。この読み取り要求と、同じレコードに対する他の並行更新アクティビティーとの逐次化は行われません。 戻されるレコードには、別のトランザクションにより更新されていてもまだコミットされていない可能性があるため、更新が後でバックアウトされた場合、このレコードが変更する可能性があります。

注:

1. UNCOMMITTED は、READINTEG 属性をサポートしない CICS のリリースで提供される保全性のレベルと同じです。
2. データ・テーブルには、どの種類であっても UNCOMMITTED を指定します。RLSACCESS(YES) が指定されている場合、UNCOMMITTED 以外のどの値も許可されますが、そのファイルに対して TABLE(CF)、TABLE(CICS)、または TABLE(USER) を指定しても無視されます。

RECORDFORMAT({V|F})
ファイル上のレコードの形式を指定します。

F レコードは固定長です。VSAM ファイルの場合は、VSAM アクセス方式のサービス定義が固定サイズ・レコードを指定しており (つまり、平均サイズが最大サイズと等しい)、ファイル内のすべてのレコードが同じサイズである場合にのみ、この値を指定してください。

ユーザー保守データ・テーブルおよびカップリング・ファシリティー・データ・テーブルでは、F は無効です。

V レコードは可変長です。すべてのユーザー保守データ・テーブルおよびカップリング・ファシリティー・データ・テーブルを可変長として指定する必要があります。そうしないと、CICS は RECORDFORMAT(F) が TABLE(CF) または TABLE(USER) オプションと競合して、無視されることを示すエラー・メッセージを戻します。

RECORDSIZE(*number*)
リモート・ファイルまたはカップリング・ファシリティー・データ・テーブルのレコードの最大長をバイト単位で指定します。サイズは 1 から 32767 の範囲で指定できます。

カップリング・ファシリティー・データ・テーブルの場合のみ
この値は、テーブルのファイル定義で LOAD(NO) を指定した場合に必要です。

また、LOAD(YES) を指定した場合にもこの属性を指定することができます (例えば、LOAD(NO) と LOAD(YES) 間のファイル定義切り替え

を簡単にするため)。ただし、LOAD(YES) を指定した場合、レコード・サイズの値は、ソース・データ・セットの値と一致している必要があります。一致していないと、CICS はテーブルのオープンに失敗します。次の 3 つの状況では、LOAD(YES) を指定したときにレコード・サイズが正しくないため CICS がエラーを検出する可能性があります。

1. CICS は、テーブルをオープンする前に、カップリング・ファシリティー・データ・テーブルのロード元であるデータ・セットの VSAM 定義のレコード・サイズがファイル定義内のサイズ (指定されている場合) と同じであることを検証します。レコード・サイズが異なると、CICS はエラー・メッセージ DFHFC7081 を戻します。
2. ファイル定義内のレコード・サイズ (指定されている場合) がデータ・セットの VSAM 定義のレコード・サイズと同じであっても、テーブルが別のレコード・サイズのデータで既にロード済みであることがテーブルのオープン時に CICS により検出されることがあります。これは、恐らくこのファイル定義で指定されたデータ・セットと異なるデータ・セットからデータをロードしたことが原因です。この場合、CICS はエラー・メッセージ DFHFC7082 を戻します。
3. オープンされているテーブルのファイル定義でレコード・サイズが指定されていても、そのテーブルが別のファイルのオープンによってロードされたために、データ・セット名が指定されていないことがあります。テーブルが既に作成されている場合、異なるレコード・サイズを指定してファイルをオープンすると、メッセージ DFHFC7083 で失敗します。

これらのエラーを避けるには、この属性の値が、シスプレックス全体で、同じカップリング・ファシリティー・データ・テーブルを参照するすべてのファイル定義で同じであることを確認するか、または LOAD(YES) を指定するファイルでこの属性をすべて省略します。

リモート・ファイルではない、または CFDT を参照しないファイルのレコード・サイズを指定した場合には、ファイルをリモート・ファイルとして再定義したり、カップリング・ファシリティー・データ・テーブルを参照するように再定義したりしない限り、そのサイズは有効になりません。ただし、レコード・サイズを指定した場合、INQUIRE FILE コマンドにより戻される値は次のようになります。

- ファイルがオープンしている場合、CICS は VSAM から取得した値を戻しますが、この値はファイル定義で指定された値とは異なる可能性があります。
- ファイルがクローズしている場合、CICS はファイル定義で指定されている値を戻します。

注: カップリング・ファシリティー・データ・テーブルでは、レコード・サイズを 63 バイト以下に維持できる場合、レコードをカップリング・ファシリティーに保持する方法により、パフォーマンスが著しく向上します。

RECOVERY({NONE|BACKOUTONLY|ALL})
ファイルに必要なリカバリー・タイプを指定します。

この属性は、RLSACCESS(YES) で定義されたファイルには使用されません。また、リカバリー・オプションが ICF カタログに定義されている場合にも使用されません。 LOG が ICF カタログで定義されている場合、CICS は RLSACCESS(NO) で定義されているファイルであっても、RECOVERY オプションを無視し、ICF カタログから LOG 値を取得します。LOG(ALL) が ICF カタログで指定されている場合には、CICS は LOGSTREAMID および BWO の値も ICF カタログから取得します。 CICS でこのカタログのリカバリー・オプションの代わりにこの属性を使用するには、**NONRLSRECOV** システム初期設定パラメーターに FILEDEF を設定します。

RLS モードでアクセスされるファイルについては、ICF カタログのデータ・セット定義にリカバリー・パラメーターを指定する必要があります。詳しくは、リカバリー可能リソースとしてのファイルの定義を参照してください。

ソース・データ・セットで定義されているカップリング・ファシリティー・データ・テーブルおよびユーザー保守テーブルでは、ICF カタログのリカバリー属性は無視されます。リカバリー属性は、関連するデータ・セットではなく、ファイルのプロパティーになります。

カップリング・ファシリティー・データ・テーブルでは、リカバリー属性は、シスプレックス全体で同じカップリング・ファシリティー・データ・テーブルを参照するすべてのファイル定義において同一である必要があります。

ALL 独自のリカバリーを管理し、ログ・マネージャーまたはリカバリー・マネージャーのサービスを使用しないカップリング・ファシリティー・データ・テーブル以外では、変更前イメージはシステム・ログに記録され、変更後のイメージは FWDRECOVLOG 属性で指定されたジャーナルに記録されます。

FWDRECOVLOG に書き込まれたレコードは、どんな自動ジャーナリング・オプションが設定されていても影響されません。

FWDRECOVLOG と共に指定される RECOVERY=ALL は、自動ジャーナリングの必要性と順方向リカバリー・ユーティリティーの必要性を分離するための手段となります。自動ジャーナリングを通して入手できない追加情報は、FWDRECOVLOG で記録されます。順方向リカバリー・サポートを提供する方法として、RECOVERY=ALL と FWDRECOVLOG を一緒に指定する方法が推奨されています。

JREQ=(WU,WN) と JID=FCT のマクロ設定値を使用する既存の順方向リカバリー・ユーティリティーは、これらの設定値でも使用できます。これらの自動ジャーナリング設定値と同等の RDO は、JNLADD=BEFORE、JNLUPDATE=YES、および JOURNAL 属性です。

CICS 保守データ・テーブルでは、データ・テーブルとそのソース・データ・セットのログへの記録、ジャーナル化、およびリカバリーが共に行われます。

ユーザー保守テーブルでは、ALL を指定すると BACKOUTONLY を指定した場合と同じ効果があり、動的バックアウトのみが行われます。ユーザー保守テーブルには、順方向リカバリー・サポートはありません。

カップリング・ファシリティー・データ・テーブルでは、ALL を指定できません。

注: VSAM ESDS ファイルに対して ALL を指定すると、CICS は ADD のバックアウトを実行できません。この状態を処理するため、エラーのためにファイルがクローズされるのを回避するためのユーザー出口 XFCLDEL をコーディングします。

BACKOUTONLY

独自のリカバリーを管理し、ログ・マネージャーまたはリカバリー・マネージャーのサービスを使用しないカップリング・ファシリティー・データ・テーブル以外では、変更前イメージはシステム・ログに記録されます。

CICS 保守データ・テーブルでは、BACKOUTONLY は、データ・テーブルとそのソース・データ・セットがリカバリー可能であることを指定します。これらは、いずれも段階ごとに更新され、必要に応じて段階ごとにリカバリーされます。

ユーザー保守テーブルでは、BACKOUTONLY は、動的バックアウトのみを指定します。ログ・レコードは作成されないため、緊急時再始動ではリカバリーが行われません。

カップリング・ファシリティー・データ・テーブルでは、BACKOUTONLY は、カップリング・ファシリティー・データ・テーブルが UPDATEMODEL(LOCKING) で定義された場合のみ許可されます。UPDATEMODEL(CONTENTION) には、この属性を指定できません。BACKOUTONLY の指定には、カップリング・ファシリティー・データ・テーブルが UOW リカバリー可能であるという意味があります。つまり、作業単位内で CFDT に加えられた更新は、その作業単位が失敗した場合、作業単位の活動中に CICS や CFDT サーバーで障害が発生した場合、または MVS で障害が発生した場合にバックアウトされます。

注: VSAM ESDS ファイルに対して BACKOUTONLY を指定すると、CICS は ADD のバックアウトを実行できません。この状態を処理するため、エラーのためにファイルがクローズされるのを回避するためのユーザー出口 XFCLDEL をコーディングします。

NONE

このファイルに対しては、リカバリー・ロギングが行われません。

REMOTENAME(*file*)

ファイルがリモート・システムに常駐している場合、このファイルが常駐しているシステムまたは領域の中で、このファイルの識別に使われる名前を指定します。名前の長さは、最大で 8 文字です。 REMOTENAME を指定しない場合、FILE 属性で指定された名前が使用されます。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

リモート名を指定する場合、ファイルを関連システムに割り当てる際に、CICSPlex SM によってその名前が使用されます。リモート名でなく、リモート・システムを指定する場合は、ローカル名 (つまり、このファイル定義の名前) がターゲット・システムと関連システムの両方で使用されます。

REMOTESYSTEM(接続)
ファイルが常駐するリモート・システムへ接続の名前を指定します。

IPIC 接続では、REMOTESYSTEM は、サービス中で取得できる IPCONN 定義の IPCONN 名の最初の 4 文字を指定します。

MRO および APPC 接続では、REMOTESYSTEM は CONNECTION 定義の CONNECTION 名を指定します。REMOTESYSTEM を指定する場合、リモート・システムのファイル名を指定するために、REMOTENAME を提供することもできます。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

注: リソース定義を RLSACCESS(NO) から RLSACCESS(YES) に変更する場合は、リモート・システム名を除去する必要があります。除去しないと、CICS は引き続きファイル要求を機能シップします。

RESSECNUM
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。詳しくは、Obsolete attributesを参照してください。

RLSACCESS({NO|YES})
CICS が RLS モードでファイルをオープンするかどうかを指定します。

NO ファイルは、RLS モードでオープンされません。RLSACCESS(NO) を指定するか、またはデフォルト設定を許可した場合、CICS は、LSRPOOLNUM 属性に応じて LSR アクセス・モードか NSR アクセス・モードでファイルをオープンします。LSRPOOLNUM(NONE) を指定した場合、アクセス・モードは NSR となり、LSRPOOLNUM (*number*) を指定した場合、アクセス・モードは LSR になります。

YES ファイルは、RLS モードでオープンされます。RLSACCESS(YES) を指定した場合、FILE のオープン時には無視される LSRPOOLNUM 属性に優先します。

73 ページの表 1に示されているように、RLSACCESS(YES) を指定すると FILE 定義で定義されているその他のいくつかの属性への影響が変化します。

表 *1.* 他の *FILE* 属性に対する *RLSACCESS(YES)* の影響

属性	**RLSACCESS(YES)** の属性
PASSWORD	無視されます。
LSRPOOLNUM	無視されます。
DSNSHARING	無視されます。
STRINGS	無視されます。RLS アクセス・モードのファイルには、常に 1024 個のストリングが含まれます。
REMOTESYSTEM REMOTENAME RECORDSIZE KEYLENGTH	これらの属性の意味は変わりません。ただし、ローカルとリモートの二重 FILE 定義には、RLS アクセス・モード・ファイルよりも値が少なくなります。 RLS には、1 つではなく多数のファイル所有領域 (FOR) があり、ローカル CICS 領域はいくつかある FOR の中から選択します。
DATABUFFERS INDEXBUFFERS	無視されます。
TABLE	TABLE(CICS) は指定できません。指定するとエラー・メッセージが表示されます。TABLE(USER) または TABLE(CF) は指定できます。ソース・データ・セット (存在する場合) に RLS モードでアクセスして、データ・テーブルをロードします。その後、要求はデータ・テーブル・サービスを使用して直接データ・テーブルにアクセスします。
RECOVERY	無視されます。リカバリー属性は、RLS ファイルの ICF カタログから入手します。
FWDRCOVLOG	無視されます。順方向リカバリー・ログ・ストリーム名は、RLS ファイルの ICF カタログから入手します。
BACKUPTYPE	無視されます。バックアップのタイプは、RLS の DFSMSdss バックアップ・ユーティリティーによって決定されます。

注**:**

1. ファイルが RLS モードでオープンされている場合、PASSWORD、LSRPOOLNUM、DSNSHARING、STRINGS、DATABUFFERS、および INDEXBUFFERS に指定された値は、表 1で説明されているように無視されます。しかし、CEMT、または EXEC CICS コマンドや SET FILE コマンドを使用して、RLSACCESS の値を YES から NO に変更すると、これらの値は無視されなくなります。また、ファイルがいったんクローズされて非 RLS モードで再オープンされると、CICS はそれらの値を使用します。
2. CICS は、常にファイル・リソース定義から RLS アクセス・モードを選択するため、ユーザーが DD ステートメントの RLS=NRI または RLS=CR パラメーターを使ってこれを上書きすることはできません。

STATUS({ENABLED|DISABLED|UNENABLED})

START=COLD または START=INITIAL で CICS の初期設定に続くファイルの初期状況を指定します。クローズされたファイルの状況は、マスター端末トランザクション CEMT で変更できます。 CICS 再始動に続くファイルの状況 (ENABLED、DISABLED、または UNENABLED) は、以前のシャットダウン時の状況にリカバリーされます。

DISABLED
コマンド・レベルのアプリケーション・プログラムからのこのファイルへの要求により、DISABLED 条件が、プログラムに渡されます。

ENABLED
このファイルに対しては、通常の処理が行えます。

UNENABLED
これにより、ファイルは、アプリケーション・プログラムから暗黙的なオープンができなくなります。この方法でファイルへのアクセスを試行すると、NOTOPEN 条件が起こります。逆に、ファイル・オープンの明示的な要求 (例えば、CEMT または EXEC CICS SET FILE OPEN コマンド) は、ファイルのオープンを試行する前に、状況を ENABLED に変更します。

この属性は、CICS バンドルによって動的に生成される FILE リソースでは無視されます。FILE リソースの初期状況は、そのリソースを定義するバンドルの初期状況から導出されます。

STRINGS({**1**|*value*})
ファイルに対して処理できる同時要求数を 1 から 255 の範囲で指定します。要求数がこの値に達すると、アクティブな要求のいずれかが終了するまで、CICS は追加の要求をキューに入れます。これは、共用リソースを使用するファイルにも、共用リソースを使用しないファイルにも適用されます。ファイル定義で RLSACCESS(YES) を指定した場合は、STRINGS 値は無視されます。しかし、RLS モード・アクセスでは常に 1024 個のストリングを取得します。

ローカル共用リソースを使用するファイルの場合は、この数値を VSAM では使用しません。 CICS では、ここで説明されている用途以外に、バッファー・プール定義のデフォルト値を計算するためにも、この数値が使用されます。

注:

1. STRINGS 値を選択する場合には、指定したストリング数の比率 (20%) が、読み取り専用要求用として CICS により予約されるので注意してください。
2. ESDS の STRINGS 値を選択する場合には、次のことを考慮してください。
   - ESDS を「追加専用」ファイル (つまり、書き込みモードでのみ使用するファイル) として使用する場合、ストリング数 1 を使用する必要があります。ストリング数が 1 より大きいと、複数のタスクが同時に ESDS に書き込みを試行するときに生じる排他制御の競合のために、パフォーマンスに著しい影響を及ぼす可能性があります。
   - ESDS を書き込みと読み取りの両方に使用し、アクティビティーのうち 80% は書き込みで使用する場合、2 つのファイル定義を定義し、一方は書き込み用ファイル、他方は読み取り用ファイルとして使用することをお勧めします。
3. ユーザー保守データ・テーブルおよびカップリング・ファシリティー・データ・テーブルの場合、テーブルに対する同時要求数は STRING 値により制限されません。ただし、ユーザー保守テーブルのロード時における同時要求数は、この値によって制限されます。

4. CICS 保守データ・テーブルでは、テーブルを更新する同時要求数が STRINGS 値により制限されます。同時読み取り専用要求数は制限されません。

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

ユーザーが必要とするデータ・テーブルのタイプを指定します。

CF カップリング・ファシリティー・データ・テーブル (CFDT)。これは、そのソース・データ・セットからは独立しているため、テーブルに変更を加えても、対応するソース・データ・セット (存在する場合) には影響しません。 CFDT では、ソース・データ・セットはオプションであり、ファイル定義の LOAD(YES) で指定されます。

CF を指定する場合は、以下も指定します。

- CFDTPOOL。テーブルが常駐するカップリング・ファシリティー・プールの名前を指定します。
- LOAD。ソース・データ・セットからテーブルをロードするかどうかを指定します (または、このデフォルトを NO にします)。
- UPDATEMODEL。テーブルが CONTENTION 更新モデルを使用するか、LOCKING 更新モデルを使用するかを指定します (またはこのデフォルトを LOCKING にします)。
- RECORDFORMAT。V を指定します (またはデフォルトを V にします)。
- MAXNUMRECS。必要な値を指定します。

カップリング・ファシリティー・データ・テーブルには、カップリング・ファシリティー・データ・テーブル・サーバーが必要です。カップリング・ファシリティー・データ・テーブル・サーバーを始動する方法については、カップリング・ファシリティー・データ・テーブル・サーバーの設定と実行を参照してください。

CICS CICS 保守データ・テーブル。これは、対応するソース・データ・セット内のテーブルに加えられたすべての変更を自動的に反映します。
CICS を指定した場合には、次の属性も指定してください。
- LSRPOOLNUM。1 から 255 の値を指定します。
- MAXNUMRECS。必要な値を指定します。

NO データ・テーブルは不要です。

USER ユーザー保守テーブル。これは、ソース・データ・セットを独立したままにし、ユーザー保守テーブルに変更を加えても対応するソース・データ・セットには影響しません。 USER を指定した場合は、次の属性も指定します。
- LSRPOOLNUM。1 から 255 の値を指定します。
- RECORDFORMAT。VARIABLE を指定します (またはこのデフォルトを VARIABLE にします)。
- MAXNUMRECS。必要な値を指定します。

TABLENAME(*cfdt*)

このファイル定義を介してアクセスするカップリング・ファシリティー・データ・テーブルの名前を指定します。名前の長さは、最大で 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

TABLE(CF) を指定したときにこの属性を省略すると、デフォルトにより、FILE に指定した名前となります。 CICS 領域でカップリング・ファシリティー・データ・テーブルを共用できるようにするには、各領域にインストールするファイル定義で、同じ CFDTPOOL 名前と同じ TABLENAME (または TABLENAME を使用していない場合は FILE 名) を指定する必要があります。 TABLENAME は、そのプール内で固有である必要があります。

テーブル名は、そのテーブルの ID としてだけではなく、セキュリティー検査でリソース名としても使用されることに注意してください。

この属性が意味を持つのは、TABLE(CF) 属性を使用して定義されたファイルのみです。 TABLE(CF) として定義されていないファイルのテーブル名を指定できますが、そのテーブル名は CICS により無視されます。カップリング・ファシリティー・データ・テーブルを参照するようにファイル定義を変更すると、TABLENAME 属性は有効になります。

UPDATE({NO|YES})
このファイルのレコードが更新できるかどうかを指定します。

UPDATEMODEL({LOCKING|CONTENTION})
カップリング・ファシリティー・データ・テーブルに使用する更新モデルのタイプを指定します。

LOCKING
CFDT がロック・モデルを使用して更新されるように指定します。これは、更新のためにレコードを読み取るときにレコードがロックされることを意味します。そのため、その更新要求が完了するまで、他の作業単位がそのレコードを変更することはできません。リカバリー可能テーブルの場合、更新要求は同期点で完了されます。リカバリー不能テーブルの場合、更新要求は REWRITE、DELETE、または UNLOCK コマンドが完了した時点で完了されます。LOCKING は、TABLE(CF) を指定するファイルのデフォルトです。LOCKING モデルを指定した場合、CFDT は次のように定義できます。

- リカバリー不能。作業単位が失敗した場合、CFDT の更新はバックアウトされません。また、ロックが保持されるのは、要求の間のみです。 CFDT がリカバリー可能でないことを指定するには、RECOVERY(NONE) を指定します。
- リカバリー可能、または UOW リカバリー可能。作業単位内で CFDT に加えられた更新は、その作業単位が失敗した場合、作業単位の活動中に CICS または CFDT サーバーで障害が発生した場合、または MVS で障害が発生した場合にバックアウトされます。RECOVERY(BACKOUTONLY) を指定して、CFDT をリカバリー可能として指定します。

ロック・モデルを使用するリカバリー可能 CFDT は、常駐しているカップリング・ファシリティーで障害が発生した場合に存続しないという点を除けば、リカバリー可能ファイルまたはリカバリー可能データ・セットに類似します。カップリング・ファシリティー・データ・テーブルには、順方向リカバリーはありません。

CONTENTION
CFDT が競合モデルを使用して更新されるように指定します。これは、更新のためにレコードを読み取るときに、レコードがロックされないということを意味します。更新でレコードを読み取った後に別のタスクによってそのレコードが変更されたり削除されたりすると、後続の REWRITE または DELETE でエラーが戻されます。 CFDT はリカバリー不能 (RECOVERY(NONE)) である必要があります。つまり、作業単位が失敗した場合に、更新はバックアウトされません。

この属性の値は、シスプレックス全体で、同じカップリング・ファシリティー・データ・テーブルを参照するすべてのファイル定義において同一である必要があります。

この属性は、TABLE(CF) 属性を使用して定義されたファイルに対してのみ有用です。 TABLE(CF) として定義されていないファイルの更新モデルを指定できますが、その更新モデルは CICS により無視されます。カップリング・ファシリティー・データ・テーブルを参照するようにファイル定義を変更すると、UPDATEMODEL 属性は有効になります。

IPCONN リソース

IPCONN リソースは、リモート・システムへの伝送制御プロトコル/インターネット・プロトコル (TCP/IP) 通信リンクを定義します。この通信リンクは、IPIC 接続と呼ばれます。

IPIC 接続のインバウンド属性の一部は、IPCONN 定義の TCPIPSERVICE オプションで指定された TCPIPSERVICE 定義によって指定されます。

PROGRAM 定義での REMOTESYSTEM 名は、その IPCONN 名によって IPCONN 定義を参照することができます。この属性は、分散プログラム・リンクに使用されます。

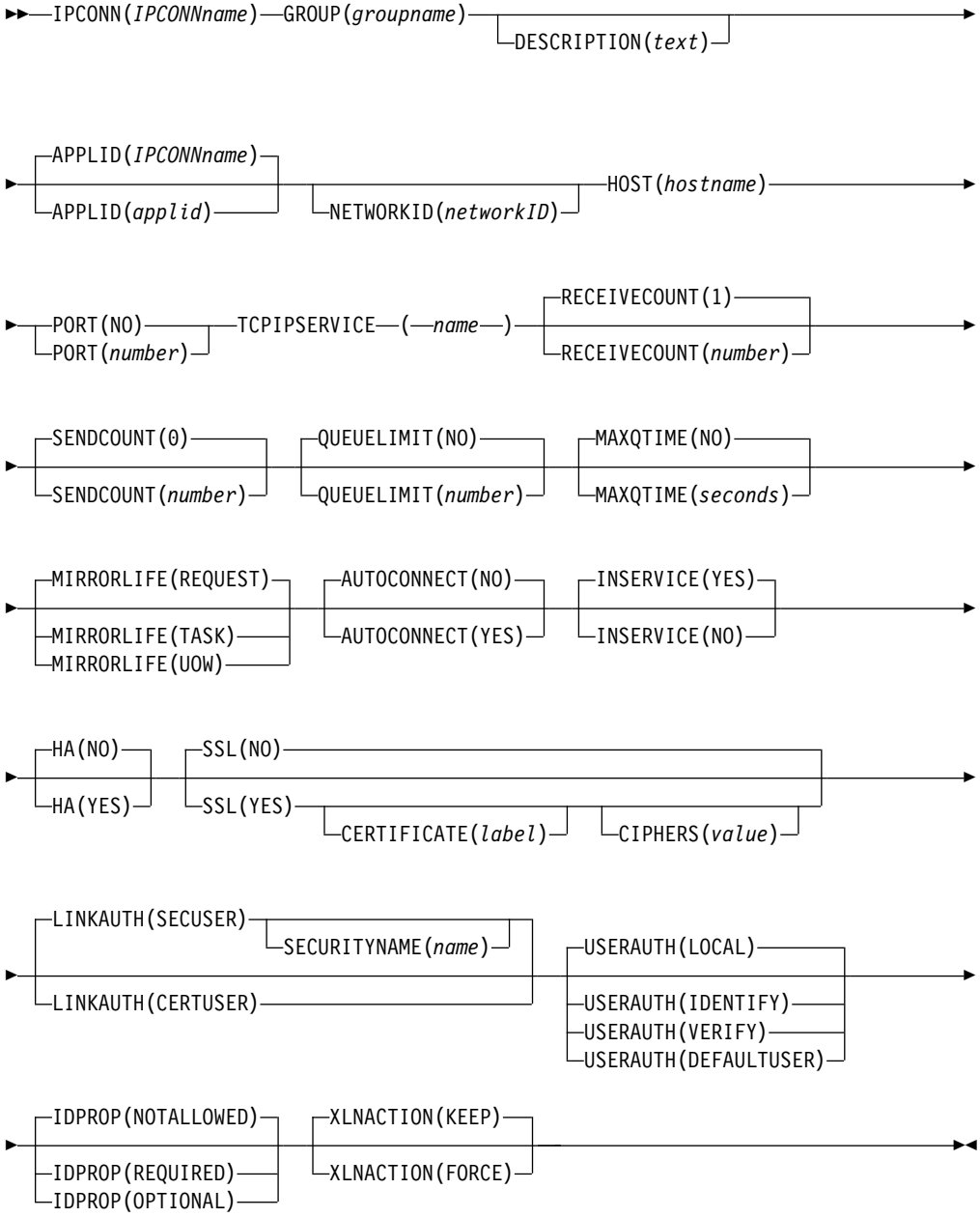
IPCONN リソースの定義に関するガイダンスは、IPIC 接続の定義を参照してください。

CEMT および CEDA を使用した IPCONN リソース定義のインストールについては、IPCONN リソース定義のインストールを参照してください。BAS の作業については、BAS IPCONN リソース定義の作業を参照してください。

IPCONN 属性

IPCONN リソースの構文と属性について説明します。

構文



属性

APPLID({*IPCONNname*|*applid*})

リモート・システムのアプリケーション ID (APPLID) を指定します。リモート・システムが CICS 領域の場合、APPLID は APPLID パラメーターに定義されます。*applid* 値は、その最大長は 8 文字で、先頭文字は英字でなければなりません。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

拡張回復機能 (XRF) CICS 領域への接続の場合は、リモート領域の総称 APPLID を指定します。

APPLID を指定しない場合、CICS は IPCONN 名を使用します。

HA クラスターへの接続に IPCONN を使用する場合、このフィールドには、クラスターを識別する 1 から 8 文字の値を入れる必要があります。この値により、領域にインストールされている他のすべての IPCONN からこの IPCONN を区別します。特定の CICS 領域を指す APPLID の名前を、HA クラスターと同じ名前にすることはできません。

重複する APPLID に関する規則を以下に示します。

- 同一の APPLID と 同一の NETWORKID を指定した IPCONN 定義を 2 つ以上インストールすることはできません。APPLID と NETWORKID を組み合わせることにより、ネットワーク上の各システムに固有の名前を付けることができます。NETWORKID オプションの説明を参照してください。
- インストール済みの MRO、APPC、または LUTYPE6.1 CONNECTION 定義の NETNAME と同じ APPLID を指定した IPCONN 定義をインストールすることができます。
- インストール済みの IPCONN 定義がインストール済みの CONNECTION 定義と同一の名前を持つ場合、IPCONN 定義の APPLID は CONNECTION 定義の NETNAME と一致していなければなりません。一致していない場合、状態に応じて以下のメッセージが出されます。
  - DFHIS3009 (IPCONN の自動インストール時にエラーを検出した場合)
  - DFHAM4913 (IPCONN のインストール時にエラーを検出した場合)
  - DFHZC6312 (CONNECTION のインストール時または自動インストール時にエラーを検出した場合)

  IPCONN 定義は CONNECTION 定義に優先します。つまり、IPCONN と CONNECTION が同じ名前を持っている場合、CICS は IPCONN を使用します。
- CONNECTION 定義と IPCONN 定義の NETNAME と APPLID が同じであれば、名前は同じでなくても、TCP/IP を介した通信で個別の *sysid* を使用できます。この場合、IPCONN によってすべてのサポート機能をルーティングする (IPCONN がある場合) という CICS のデフォルトを使用せずに済みます。

この規則は、インストール時に検証されます。

AUTOCONNECT({NO|YES})

IPCONN 定義をインストールする際にセッションを確立するかどうかを指定します (その後に CEDA INSTALL コマンドを発行するか、CEMT または EXEC CICS SET TCPIP OPEN コマンドを使用して TCP/IP で通信を開始すると、CICS の初期化時にこの処理が行われることがあります)。リモート・システム

が使用不可のためにこの時点で接続できない場合に、しばらく経過してもリモート・システムが使用可能にならず、システム自体が通信を開始しないときには、後で CEMT または EXEC CICS SET IPCONN(*name*) INSERVICE ACQUIRED コマンドを使用することでリンクを獲得することができます。

NO IPCONN がインストールされている場合、CICS はセッションを確立しようとしません。

YES IPCONN がインストールされている場合、CICS はセッションを確立しようとします。

IPCONN 定義をインストールする場合に接続を行うには、以下の条件に注意してください。

1. この IPCONN 定義の TCPIPSERVICE オプションで指定した TCPIPSERVICE 定義も、この領域にインストールし、PROTOCOL(IPIC) を指定する必要があります。
2. 対応する IPCONN 定義と TCPIPSERVICE 定義は、リモート領域にインストールする必要があります。
   a. リモート領域の IPCONN 定義の HOST オプションは、この領域に指定する必要があります。
   b. リモート領域の IPCONN 定義の PORT オプションは、この IPCONN で指定したローカル TCPIPSERVICE 定義の PORTNUMBER オプションに指定したものと同じポート番号を指定する必要があります。
   c. リモート領域の TCPIPSERVICE 定義 (リモート領域の IPCONN 定義で指定) は PROTOCOL(IPIC) を指定し、その PORTNUMBER オプションに、この IPCONN の PORT オプションで指定したものと同じポート番号を指定する必要があります。

PORT(NO) を指定している場合は、AUTOCONNECT(YES) を指定できません。

CERTIFICATE(*label*)

HOST または PORT 定義で識別される TCPIPSERVICE リソースが SSL(CLIENTAUTH) で定義されている場合は、IPIC 接続が獲得される際の SSL ハンドシェーク時に、クライアント証明書として使用する X.509 証明書のラベルを指定します。この属性を省略した場合は、CICS 領域ユーザー ID の鍵リングに定義されているデフォルトの証明書が使用されます。

証明書ラベルは最大32バイトです。

証明書は、外部セキュリティー・マネージャー・データベースの鍵リングに保管する必要があります。詳細については、鍵リングの手動構築を参照してください。

この属性を指定する場合は、SSL(YES) も指定する必要があります。

CIPHERS(*value*)

CIPHERS 属性は、以下の 2 とおりの方法のいずれかで指定できます。

- 最大 28 個の 2 桁の暗号スイート・コードを示すリストとして解釈される 56 桁までの 16 進数字で構成されるストリング。

- SSL 暗号スイート仕様ファイルの名前。この z/OS UNIX ファイルは、**USSCONFIG** システム初期設定パラメーターによって指定されるディレクトリーの security/ciphers サブディレクトリー内にあります。例えば、**USSCONFIG** が /var/cicsts/dfhconfig に設定され、**CIPHERS** が strongciphers.xml に設定されている場合、完全修飾ファイル名は /var/cicsts/dfhconfig/security/ciphers/strongciphers.xml です。詳しくは、SSL 暗号スイート仕様ファイルの作成を参照してください。

CEDA トランザクションを使用してリソースを定義すると、CICS は、デフォルトの許容コード・リストに従ってその属性を自動的に初期化します。この属性を CICS で初期化できるようにするためには、CEDA を実行する CICS 領域で KEYRING システム初期設定パラメーターを指定する必要があります。KEYRING を設定しなかった場合、CICS はその属性を初期化しません。デフォルトのコード・リストは、35363738392F303132330A1613100D15120F0C です。ただし、システム初期設定パラメーターで **NISTSP800131A=CHECK** が設定されている場合は、35363738392F303132330A1613100D です。

暗号コードは、再配列することも、初期リストから除去することもできます。ただし、指定した暗号化レベルのデフォルト・リストに含まれていない暗号コードを追加することはできません。値をコードのデフォルト・リストにリセットするには、すべての暗号スイート・コードを削除します。それにより、フィールドには自動的にデフォルト・リストが取り込まれます。

詳しくは、暗号スイートおよび暗号スイート仕様ファイルを参照してください。

DESCRIPTION(*text*)
このフィールドに定義しているリソースの説明を加えることができます。説明文の長さは最大 58 文字です。使用できる文字に制限はありません。しかし、括弧を使用する場合は、左右の括弧が対になるようにしてください。CREATE コマンドを使用する場合は、テキスト内の単一アポストロフィごとに 2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

HA({**NO**|**YES**})
高可用性クラスターとの接続に IPCONN を使用することができるかどうかを指定します。

NO これはデフォルト値です。 IPCONN を使用して高可用性クラスターに接続することはできません。

YES IPCONN は、高可用性クラスターの一部である領域に接続しなければなりません。

HOST(*hostname*)

リモート・システムのホスト名、あるいはその IPv4 アドレスまたは IPv6 アドレスを指定します。この名前の長さは、最大 116 文字です。 IPv4 アドレスと IPv6 アドレスは、複数のフォーマットで指定することができます。アドレス・フォーマットについて詳しくは、IP アドレスを参照してください。

IPv6 アドレス (または IPv6 アドレスに解決されるホスト名) を指定する場合は、ユーザーが二重モード (IPv4 および IPv6) 環境で操作しており、通信先のクライアントおよびサーバーも二重モード (IPv4 および IPv6) 環境で稼働していることを確認してください。 IPv6 についての詳細は、IPv6 と CICS についてを参照してください。

HOST 属性は、英数字、ハイフン (-)、コロン (:)、およびピリオド (.) のみで構成する必要があります。ただし、IP アドレスではなく、キャラクター・ホスト名を指定する場合は、コロンを使用することはできません。CICS は定義時にホスト名を妥当性検査します。ホスト名は大文字、小文字、または大/小文字混合で入力することができます。ただし、IP アドレスではなく文字のホスト名を指定する場合、そのホスト名は IPCONN 定義で小文字に変換されます。

SENDCOUNT 属性がゼロの場合、HOST は無視されます。SENDCOUNT がゼロより大きい場合、HOST は必要な属性です。

IDPROP({**NOTALLOWED**|**OPTIONAL**|**REQUIRED**})

送信側が分散 ID を接続相手のシステムに送信するかどうかを指定します。IDPROP 属性は、接続がシスプレックスの外に及ぶ場合に限り意味があり、主に企業間で分散 ID が送信されないようにするために使用されます。同じシスプレックスに含まれているシステム同士の接続の場合は、IDPROP(OPTIONAL) を指定した場合と同じ動作になり、他の設定は無視されます。

分散 ID は、接続しているシステムが以下の基準をすべて満たしている場合にのみ流れます。

- 両方のシステムとも ID 伝搬に関係している。ID 伝搬のサポートおよび要件を参照してください。
- 両方のシステムとも同じシスプレックス内にあるか、SSL を使用して接続されている。
- 受信側のシステムで、IPCONN リソース定義内に USERAUTH(IDENTIFY) が指定されている。

NOTALLOWED

この接続を使用する要求では、送信側のトランザクションに関連付けられたユーザー ID が送信されます。NOTALLOWED はデフォルト値です。

REQUIRED

この接続を使用する要求では、分散 ID が必要です。REQUIRED を指定する場合には、受信側のシステムが分散 ID をサポートしていなければなりません。送信側のトランザクションに関連付けられたユーザー ID は送信されません。

OPTIONAL

分散 ID が送信されます (使用可能な場合)。送信側のトランザクションに関連付けられたユーザー ID も送信されます。

INSERVICE({**NO**|**YES**})

インストール時の IPCONN リソースの状況を指定します。

NO この接続は、メッセージを受信することも、出力を送信することもできません。

YES この接続は使用可能です。

IPCONN(*IPCONNname*)

この IPCONN 定義の名前を指定します。名前の長さは、最大で 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

IPIC 接続性を使用する、CICS TS 3.2 以降の領域間の分散プログラム・リンク (DPL)、CICS TS 4.1 以降の領域間のトランザクション・ルーティング、または CICS TS 4.2 以降の領域間のファイル制御、一時データ、一時記憶域の各要求の機能シップにこの IPCONN を使用する場合、その名前は 4 文字の「ローカル名」(SYSID) に一致している必要があります。CICS はこの名前 (4 つの末尾ブランクが埋め込まれている) でリモート・システムを認識します。

注: DPL 要求のリモート・ターゲット領域の名前 (SYSID) は、以下の方法で指定することができます。

- インストール済み PROGRAM 定義の REMOTESYSTEM オプション
- EXEC CICS LINK PROGRAM コマンドの SYSID オプション
- 動的ルーティング・プログラム

IPCONN 名は、インストール済み MRO または APPC CONNECTION 定義の名前と同じにすることができます。

LINKAUTH({**CERTUSER**|**SECUSER**})

セキュリティーを初期設定している (SEC=YES) CICS システムにおける、リンク・セキュリティーのユーザー ID 設定方法を指定します。

CERTUSER

パートナー・システムとの TCP/IP 通信を SSL で構成し、SSL ハンドシェーク時にパートナー・システムから証明書を受け取らなければなりません。

IPCONN リソースは、SSL(CLIENTAUTH) で定義されている TCPIPSERVICE リソースを参照する必要があります。

受け取った証明書は外部セキュリティー・マネージャーに定義し、リンク・セキュリティーを確立する際に使用するユーザー ID と関連付ける必要があります。

SECUSER

SECURITYNAME 属性に指定されたユーザー ID を使用してリンク・セキュリティーを確立するように指定します。

デフォルト値は LINKAUTH(SECUSER) です。

MAXQTIME({NO|*seconds*})

キューに入れられた割り振り要求 (この接続の空きセッションを待機) が、キューがパージされるまでに待機可能な最大時間を指定します。

注:

- 最大キューイング時間を使用できるのは、キューの長さ制限が QUEUELIMIT オプションに指定されている場合のみです。
- 時間制限が適用されるのは、キューの長さが QUEUELIMIT 値に達した場合のみです。

NO CICS は空きセッションを待機する割り振り要求のキューを維持します。要求が待機できる時間の長さに対する制限は設定されません。ただし、DTIMOUT メカニズムを個々の要求に適用することはできます。

seconds

この接続が無応答に見える場合に、空きセッションを待つ割り振り要求をキューイングできるおおよその最大時間 (秒)。*seconds* は、0 から 9999 の範囲でなければなりません。

割り振り要求のキューがその最大長 (QUEUELIMIT 属性で指定) に到達し、さらに新規割り振り要求を接続のために受信した場合、新規割り振りに要する時間が平均で最大キュー時間を超えることをキューの処理速度が示していれば、キューはパージされ、メッセージ DFHIS500 が出されます。キューがパージされると、待機している割り振り要求は SYSIDERR を戻します。

その接続でセッションが正常に解放されるまで、それ以降のキューイングは起こりません。この時点で、CICS は DFHIS5001 メッセージを発行し、通常のキューイングを再開します。

MIRRORLIFE({REQUEST|TASK|UOW})

この領域で受け取られる、機能シップされたファイル制御、一時データ、および一時記憶域要求のミラー・タスクの最短存続時間を指定します。通常、ミラー・タスクは、できるだけ速やかに終了されます。これは、アクティブ・タスクの数を最小にするとともに、そのセッションを長期にわたって保持することを避けるためです。しかし、一部のアプリケーションでは、データ保全性という点では不要であっても、次の同期点までミラー・タスクとセッションの両方を保存した方が効率的な場合があります。例えば、多数の READ FILE 要求をリモート・システムに出すトランザクションは、各要求ごとのミラー・タスクではなく、単一のミラー・タスクによって処理した方が効率的です。このように、送信側におけるセッション割り振り、および受信側におけるミラー・タスクの接続といったオーバーヘッドを減らすことができます。ファイル制御、一時データ、または一時記憶域要求が機能シップされるまで、指定された MIRRORLIFE 値はミラー・タスクの存続時間に反映されません。

REQUEST

ミラー・タスクはできるだけ速やかに終了します。リソース所有領域内でミラーがセッションを保持する必要がないリカバリー不能作業 (非更新 READ FILE 要求など) の場合、これは要求の処理後となります。ミラーがセッションを保持する必要があるリカバリー不能作業 (ファイル・ブラウズ要求など) の場合、保持する必要がなくなるまで (例えば

ENDBR を発行した時点まで) です。リカバリー可能な作業の場合、ミラーは次の同期点まで保持されます。これはデフォルト値です。

TASK
ミラー・タスクは、リモート要求を発行するアプリケーションのタスクが終了するまで、そのアプリケーションから使用可能な状態になります。この値を指定すると、ユーザー同期点の後に続く要求のための、セッションの再割り振りとミラー・タスクの再接続のオーバーヘッドを節約できます。機能シップのファイル制御、一時データ、または一時記憶域要求に長期実行タスクが使用される可能性がある場合は、この値を指定しないでください。セッションがタスクに割り振られると、たとえ使用されていなくてもタスク終了までは解放されないためです。解放されるまでの間は、セッションを他のタスクに使用できません。

UOW ミラー・トランザクションは、次の同期点が発行されるまで、リモート要求を発行するアプリケーションにとって使用可能な状態になります。この値を指定すると、アプリケーションにこの作業単位内でさらに多くの機能シップのファイル制御、一時データ、または一時記憶域要求がある場合に、ミラー・トランザクションとの通信の再確立のオーバーヘッドを節約できます。

このパラメーターは、リソース所有領域の IPCONN で指定される場合に有効になります。特に DPL 要求と SYNCONRETURN または TRANSID を併用している場合は、TASK または UOW 値とこのパラメーターを共に指定する際に注意してください。

NETWORKID(*networkID*)
リモート・システムのネットワーク ID を指定します。リモート・システムのネットワーク ID は、その z/OS Communications Server NETID、またはその UOWNETQL システム初期設定パラメーターの値 (VTAM=NO システムの場合) のいずれかです。

NETWORKID を指定しない場合、CICS では、リモート・システムがローカル・システムと同じネットワークにあると見なされます。この場合、CICS はこの CICS (つまり、この定義がインストールされている CICS) の z/OS Communications Server NETID、または UOWNETQL システム初期設定パラメーターの値を使用します。

別のネットワークにあるリモート・システム、および異なる z/OS Communications Server NETID または UOWNETQL 値を含むリモート・システムに接続する場合は、NETWORKID を指定します。この場合、2 台以上のリモート・システムに同じアプリケーション ID を指定することができます。CICS のアプリケーション ID はシスプレックス内では固有である必要があります。ただし、シスプレックス外または異なるシスプレックスのシステムなどに接続することはできます。 APPLID 属性と NETWORKID 属性の組み合わせにより、リモート・システムが固有の名前によって参照されるようになります。

NETWORKID 値は、リモート・システムのネットワーク ID と一致している必要があります。

指定しない場合、NETWORKID 値は、IPCONN リソースが最初にインストールされる際に派生されます。これは、ローカル NETID 値が変更された場合でも、ウォーム・スタート間において変更されません。

この名前は、最大 8 文字の長さにすることができ、アセンブラー言語規則に従います。名前の先頭は英字にする必要があります。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

PORT(**NO**|*number*)
リモート領域が listen するポート番号を 10 進数で指定します。

NO CICS Transaction Gateway からの接続時に、この IPCONN リソースがアウトバウンド要求に使用されません (つまり、送信セッションがありません)。

CICS から CICS IPCONN リソースの場合、NO は無効になります。

NO を指定すると、AUTOCONNECT の値が強制的に NO になります。

number
値の範囲は 1 から 65535 までです。ポート番号と HOST 値を結合したものが、この IPCONN リソースのアウトバウンド要求の宛先になります。

QUEUELIMIT({**NO**|*number*})
空きセッションを待っている間に、CICS がキューイングする割り振り要求の最大数を指定します。

NO 空きセッションを待っている間に、CICS がキューイングする割り振り要求の数に制限はありません。

number
空きセッションを待っている間に、CICS がその接続においてキューイングできる割り振り要求の最大数です (0 から 9999 の範囲)。キューに入れられた割り振り要求がこの制限に到達すると、以後、割り振り要求は失敗し、キューがこの制限を下回るまで SYSIDERR を返します。

RECEIVECOUNT({**1**|*number*})
受信セッション (つまり、着信要求を受信するセッション) の数 (1 から 999 までの範囲) を指定します。使用される受信セッションの数は、リモート・システムに定義されている送信セッション数にも左右されます。接続が確立すると、これらの値は交換されて低い方の値が使用されます。

SECURITYNAME(*name*)
リンク・セキュリティーに使用する、リモート・システムのセキュリティー名を指定します。

セキュリティーが初期設定してあり (SEC=YES)、かつ LINKAUTH(SECUSER) を使用している CICS システムでは、リモート・システムの権限を確立するためにセキュリティー名が使用されます。

セキュリティー名は、この領域で有効な RACF ユーザー ID である必要があります。この領域の保護リソースに対するアクセスは、RACF ユーザー・プロファイルとそのグループ・メンバーシップに基づいています。IPCONN がインス

トールされたときにセキュリティー名が有効な RACF ユーザー ID でない場合、CICS はセキュリティー名にデフォルトのユーザー ID を使用します。

デフォルト値は LINKAUTH(SECUSER) です。

SENDCOUNT({**0**|*number*})

送信セッション (つまり、発信要求を送信するセッション) の数 (0 から 999 までの範囲) を指定します。使用される送信セッションの数は、リモート・システムに定義されている受信セッション数にも左右されます。接続が確立すると、これらの値は交換されて低い方の値が使用されます。0 を指定した場合、この IPCONN は着信作業のみの処理ができます。接続されたシステムに要求を送信することはできません。

CICS から CICS IPCONN リソースの場合、SENDCOUNT(0) は無効になります。

SENDCOUNT(0) の場合は、PORT(NO) を指定する必要があります。SENDCOUNT 値がゼロより大きい場合、PORT に数値を指定する必要があります。

定義されているポートがないため、SENDCOUNT(0) を含む IPCONN リソースは取得できません。

SSL({**NO**|**YES**})

送信データの暗号化に Secure Sockets Layer (SSL) を使用するかどうかを指定します。SSL を使用する場合は、MAXSSLTCB の値を少なくとも、CICS 領域によって SSL を使用する IPIC 接続の数の 2 倍に設定する必要があるので注意してください。

NO Secure Sockets Layer (SSL) は使用されません。接続を獲得する際にセキュリティー検査は行われません。アウトバウンド・メッセージには、暗号化は適用されません。

YES SEC システム初期設定パラメーターに Yes が設定されている場合は、Secure Sockets Layer (SSL) が使用されます。HOST および PORT 属性によって識別される TCPIPSERVICE リソースが SSL(CLIENTAUTH) で定義されている場合、CICS は CERTIFICATE 属性に指定されているクライアント証明書を抽出し、それを使用してパートナー・システムへの IPIC 接続を獲得します。SSL 暗号化処理は、この IPCONN リソースから送信されるすべてのメッセージに適用されます。暗号化のレベルは、CIPHERS オプションの値によって決まります。

TCPIPSERVICE(*name*)

この IPCONN リソース用のインバウンド処理の属性を定義する、PROTOCOL(IPIC) の TCPIPSERVICE 定義の名前を指定します。

USERAUTH({**LOCAL**|**IDENTIFY**|**VERIFY**|**DEFAULTUSER**})

セキュリティーを初期設定してある (SEC=YES) CICS システムでの、接続時ユーザー・セキュリティーのユーザー ID 設定方法を指定します。

LOCAL

CICS は、クライアントからのユーザー ID またはパスワードを受け入れません。すべての要求は、リンク・ユーザー ID またはデフォルトのユーザー ID (リンク・ユーザー ID がない場合) で実行されます。

IDENTIFY
着信接続要求には、ユーザー ID を指定する必要があります。接続するシステムが別の CICS システムあるいは CICS TG システムである場合など、事前認証ユーザーに信頼されている場合は、IDENTIFY を入力します。

SSL クライアント認証を使用するか、接続するシステムが同じシスプレックス内にある必要があります。

VERIFY
着信接続要求には、ユーザー ID とユーザー・パスワードを指定する必要があります。接続するシステムが不明な場合や信頼できない場合は、VERIFY を指定します。

DEFAULTUSER
CICS は、パートナー・システムからのユーザー ID およびパスワードを受け入れません。すべての要求は、デフォルトのユーザー ID で実行されます。

XLNACTION({KEEP|FORCE})
新規ログ名をパートナーのシステムから受信した場合にとるアクションを指定します。新規ログ名の受信は、そのパートナーがリカバリー情報を削除したことを示します。

KEEP リカバリー情報は保持され、事前定義アクションは未確定作業単位には行われません。

この接続は、パートナーとのすべての未解決のリカバリー可能作業 (すなわち、未確定 UOW、つまり、古いログ名を使用するパートナー・システムでは未確定だった UOW に関する情報) が CEMT または SPI インターフェースを使用して完了するまで、同期レベル 2 プロトコルを必要とする新規作業を実行できません。したがって、常に同期レベル 2 プロトコルを使用する CICS-CICS 間通信に接続が使用されるので、パートナーとの未解決のリカバリー可能作業のすべてが完了するまで接続を獲得することができません。

FORCE
新規ログ名を使用した新しい作業が開始される前に、未確定作業単位 (UOW) に対する事前定義の決定 (その TRANSACTION 定義の未確定属性によって定義された) が実施されます。また、CICS は、パートナー・システムでは未確定だった UOW を解決するために保管されていた情報も削除します。

重要: このオプションを使用すると、データ保全性が損なわれることがあります。

CICS Transaction Gateway で使用される IPCONN 定義の XLNACTION パラメーターは無視されます。

JOURNALMODEL リソース

JOURNALMODEL リソースは、CICS ジャーナル名 (または ID) と MVS システム・ロガーが管理する関連物理ログ・ストリームとの間の接続、またはジャーナル名と SMF ログとの間の接続を定義します。

これらは主にユーザー・ジャーナルを対象としていますが、システム・ログおよび順方向リカバリー・ログ (非 RLS のみ) 用にジャーナル・モデルを定義することもできます。ただし、順方向リカバリー・ログの場合、VSAM カタログ内の順方向リカバリー用にすべてのログ・ストリーム名を定義することが推奨されます。これは、VSAM ファイルが RLS モードで処理される場合には必須です。非 RLS モードのファイルの場合は任意指定で選択できます。

ジャーナル管理テーブルとは異なり、CICS が使用するすべてのジャーナルに対してジャーナル・モデルを定義する必要はありません。代わりに、大部分の CICS ジャーナルのログ・ストリーム名へのマッピングを記述した、汎用モデル定義をいくつか定義します。 CICS で提供されたデフォルト・モデルを使用することができ、独自のモデルを定義する必要がない場合もあります。汎用モデルに加えて、特殊な取り扱いが必要な場合 (たとえば、SMF ロギングや、他のログ・ストリームとのマージなど) には、必要に合わせて特定のモデルを定義できます。

JOURNALMODEL 定義はいつでも変更することができますが、CICS がモデル定義を使用して既に作成したジャーナル・エントリーについては、最初に DISCARD JOURNALNAME() コマンドを使用して既存のエントリーを削除しない限りは、変更は反映されません。

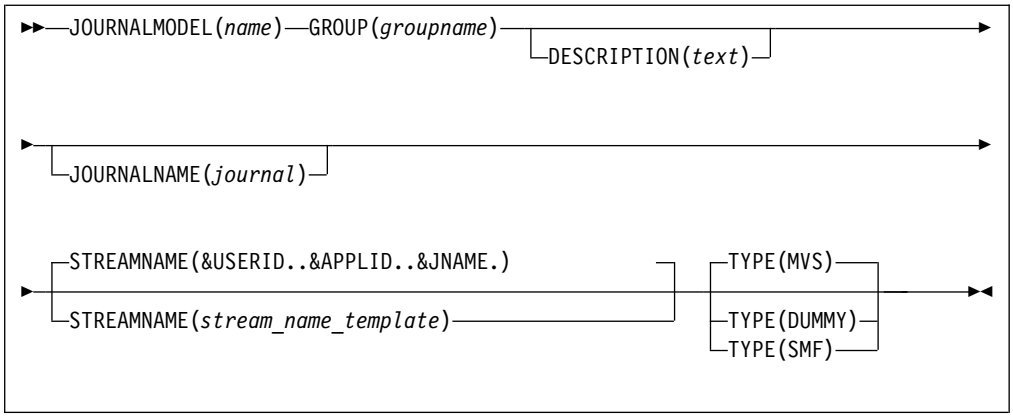
互換性に関する注意事項**:** CICS Transaction Server for z/OS より前のリリースとの API 互換性を確保するために、CICS は、以下の目的で、01 から 99 の範囲の数値ジャーナル ID を引き続きサポートします。

- FILE リソース (あるいは、BDAM ファイルの場合は DFHFCT マクロ・エントリー) で指定されたファイル制御自動ジャーナリングのため
- PROFILE リソース定義で指定された端末制御自動ジャーナリングのため
- FILE リソースで指定された順方向リカバリー・ロギングのため
- API ジャーナル・コマンド (EXEC CICS WRITE JOURNALNUM コマンドなど) を使用するユーザー・ジャーナリングのため

BAS の作業については、BAS ジャーナル・モデル・リソース定義の作業を参照してください。

JOURNALMODEL 属性

JOURNALMODEL リソースの構文および属性について説明します。



DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

JOURNALMODEL(*name*)
この JOURNALMODEL 定義の名前を指定します。

ジャーナル・モデル名は、CSD ファイル内の特定の JOURNALMODEL 定義を参照するために使用されます。CICS ジャーナル名に対応している必要はありません。ただし、 JOURNALNAME属性を省略した場合、JOURNALMODEL名は JOURNALNAMEとしても使用されます。

名前の最大長は 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

JOURNALNAME(*journal*)
この定義を適用するジャーナル名を指定します。 JOURNALNAME属性を 省略した場合、JOURNALMODEL属性で 指定した名前がジャーナル名として 使用

されます。システム・ログと log of logsモデルに総称名を使用しても、 あまり役立つ用途はありませんが、 名前は、ジャーナルの具体的な名前でも 総称名でもかまいません。

この名前の長さは最大 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

定義できる名前の書式は以下のとおり です。

システム・ログの場合
: システム・ログの JOURNALMODELを定義するには、 1次ログ・ストリームは DFHLOG、2次ログ・ストリーム はDFHSHUNTとして名前を指定 します。CICS 領域内のこれらのログ・ストリームごとに、ジャーナル・モデルを 1 つのみインストールします。

  DFHLOG、DFHSHUNT、およびDFHLGLOGの CICS提供の定義は、DFHLIST内の グループDFHLGMODに含まれて います。

log of logs の場合
: log of logsの JOURNALMODELを定義するには、 名前をDFHLGLOGと指定します。log of logs の目的について詳しくは、ログのログ (DFHLGLOG) で使用されるログ・ストリームの計画を参照してください。

自動ジャーナルの場合
: 自動ジャーナル (ファイル制御と端末制御) の場合、名前は DFHJ*nn* の形式にする必要があります。ここで、*nn* は 1 から 99 の範囲の数値です。 名前は、ジャーナルの具体的な名前でも 総称名でもかまいません。

ユーザー・ジャーナルの場合
: ユーザー・ジャーナルの場合、 名前は8文字以内とし、ジャーナルの具体的な名前でも総称名でも かまいません。CICS Transaction Server for z/OS より前のリリースとの互換性が必要な場合、名前は DFHJ*nn* の形式にする必要があります。ここで、*nn* は 1 から 99 の範囲の数値です。

順方向リカバリー・ログ（非**RLS**）の場合
: 非 RLS 順方向リカバリー・ログの場合、名前は、DFHJ*nn* の形式にする必要があります。ここで、*nn* は 1 から 99 の範囲の数値です。 名前は、ジャーナルの具体的な名前でも 総称名でもかまいません。

注**:** VSAM RLS順方向リカバリー・ログと共に 使用するためにジャーナル・モデルを定義 することはできません。CICS は、完全修飾 LSN を VSAM カタログから直接取得するため、LSN を取得するためのジャーナル・モデルを必要としません。

以下のように、特殊記号%、+、および*を 使用して総称名を定義します。

- %記号または+記号を使用して、 ジャーナル名内の任意の単一の文字を 表すことができます。
- 名前の末尾で*記号を使用して、 任意の数の文字を表すことが できます。単一の*のデフォルト名は、 さらに具体的な名前と一致する、任意の ジャーナル名と一致するように使用されます。

ジャーナル名に一致する複数の JOURNALMODEL 定義がインストールされている場合、CICS は、以下のように最も一致するものを選択します。

1. 完全に一致する固有の JOURNALNAME を持つ JOURNALMODEL がある場合、CICS は、このモデルを使用します。
2. 完全一致突き合わせがない場合、 ジャーナル名は一致する総称項目と 比較され、最も具体的な項目が使用 されます。

   どちらが具体的であるかを調べるために 名前を比較するとき、名前は1文字ずつ 比較されます。最初に文字が異なって いる場所で以下のように判断します。
   - 一方が離散的文字(%、+、または*ではない） で、もう一方が総称文字(%、+、または*) である場合、離散的文字を含む方が 使用されます。
   - 一方が%または+で、もう一方が*である場合、 %または+を含む方が使用されます。
3. 重複する JOURNALMODEL 定義 (同じ JOURNALNAME を持つ定義) が存在する場合、CICS は、最後に処理された定義を使用します。

重要: 単一のアスタリスク(*)のみを使用して、 完全な総称ジャーナル名を定義する場合は 注意してください。システム・ログに対して （ジャーナル名DFHLOGを使用して）特定の ジャーナル・モデルを定義しておらず、 ログ・ストリーム名が完全修飾リテラル名である 場合、このことは特に重要です。JOURNALNAME(*) を使用してジャーナル・モデルを定義し、システム・ログのジャーナル・モデルを定義しない場合、CICS は、汎用モデル定義で定義されたログ・ストリーム名を使用します。このようにすると、他のジャーナル および順方向リカバリー・ログが、総称ジャーナル・ モデルによって同じログ・ストリームに割り当て られた場合、問題が発生します。

STREAMNAME({**&USERID..&APPLID..&JNAME.**|*stream_name_template*})

明示的な MVS システム・ロガーのログ・ストリーム名、またはログ・ストリーム名を構成するために使用するテンプレートを指定します。 STREAMNAME は、MVS の LOGSTREAMTYPE をで定義されたジャーナル・モデルにのみ適用できます。

4つのシンボル名を以下に示します。このうち 3つまでを使用できます。

&USERID.
CICS 領域ユーザー ID のシンボル名 (最大 8 文字)。領域がユーザー IDを持たない場合、 ストリング「CICS」が使用されます。

&APPLID.
システム初期設定パラメーターで指定された CICS 領域アプリケーション ID のシンボル名 (最大 8 文字)。

注**:** XRF を使用しており、APPLID システム初期設定パラメーターに APPLID=(generic_applid,specific_applid) を指定した場合、これは、CICS が &APPLID を解決するときに使用する汎用アプリケーション ID です。

&JNAME.
具体的な一致または汎用の一致によってこのジャーナル・モデル定義を参照するジャーナル名のシンボル名。 &JNAME. 長さは最大 8 文字です。

&SYSID.
SYSIDNT システム初期設定パラメーターで指定された CICS 領域 SYSID のシンボル名。 SYSIDNTが指定されない場合、 ストリング「CICS」が使用されます。

シンボル名のデフォルトのセットは以下のとおりです。 &USERID..&APPLID..&JNAME.

例: &USERID..&APPLID..&JNAME. =

CICSHA##.CICSHAA1.DFHJ02

ここで、

CICSHA##
すべての AOR によって使用される CICS 領域ユーザー ID です。

CICSHAA1
1つのAORのアプリケーションIDです。

DFHJ02
自動ジャーナルのジャーナル名です。

代替となるシンボル名のセットを以下に示し ます。

&SYSID..&APPLID..&JNAME. =
SYSA.CICSHAA1.DFHJ02

ここで、

SYSA SYSIDNTシステム初期化パラメーター によって指定された文字ストリング です。

CICSHAA1
1つのAORのアプリケーションIDです。

DFHJ02
自動ジャーナルのジャーナル名です。

CICS は、シンボル名を含め、定義されているように JOURNALMODEL リソースをインストールします。

stream_name_template
ログ・ストリーム名には、MVS データ・セット名で定義されているように、非修飾名または修飾名のいずれかを指定します。

- 非修飾名: 1 から 8 文字の英数字または国別文字 ($ # @)、または ハイフン。名前の先頭文字は英字または国別文字 (A-Z $ # @) でなければなりません。
- 修飾名: ピリオドで結合された複数の名前 (最大 26 文字)。修飾名内のそれぞれの 名前は、非修飾名の規則に従い、 各修飾名の後にはピリオドを 負荷する必要があります （最後の修飾を除く）。その例を次に示します。

  ```
  name_1.name_2...name_n
  ```

  ここで、26文字の制限によって 名前の数は制限されます。

修飾または非修飾のデータ・セット名の規則について詳しくは、 z/OS MVS JCL 解説書を参照してください。

特定の文字（許可されたセットから使用）と、 置換用のシンボル名の混合によって構成 されるログ・ストリーム名を構成する ことができます。置換後の名前が、 修飾および非修飾ログ・ストリーム名の 規則に従い、またピリオドを含めて26文字を 超えないようにする必要があります。 そのため、修飾名の中のそれぞれの名前が 最大の8文字である場合、3つの名前のみに制限 され、最初の名前と2番目の名前、2番目の名前と 3番目の名前は、ピリオドによって区切られ ます。以下に例を示します。

```
CICSDA##.CICSDAA1.FWDRECOV
```

順方向リカバリー・ログ・ストリームの場合。ログ・ ストリーム名は、JOURNALMODEL定義に対して ジャーナル名が最初に解決されたときに、 シンボリック置換によって決まります。

複数の CICS 汎用ログに同じログ・ストリーム名を指定することで、異なる CICS 領域からのログ・ストリームをマージすることができます。ただし、汎用ログ・ストリームを CICS システム・ログとマージすることはできません。また、異なる CICS 領域からのシステム・ログをマージすることもできません。

異なる CICS システムからのログ・ストリームをマージする場合、厳密に MVS システム・ロガーのタイム・スタンプの順序でログ・データ・ブロックがログ・ストリームに書き込まれます。ただし、異なる CICS 領域からの個々のレコードは、異なるブロックにまたがり、厳密にタイム・スタンプの順序にならない場合があります。

CICS ログ・ストリームは、ログ・ストリームを読み取るプログラムによってフォーマットを処理する準備ができている場合を除き、他の製品によって生成されたログ・ストリームとマージしないでください。

セキュリティー上の注記: CICS および MVS システム・ロガーに対してログ・ストリームを定義した場合、RACF (または同等の外部セキュリティー・マネージャー) に必要なセキュリティー許可が定義されている必要があります。このセキュリティー許可は、新規のログ・ストリームを参照する CICS 領域を始動する前に必要です。 RACF は、この目的で、LOGSTRM 一般リソース・クラスをサポートしています。

TYPE({DUMMY|MVS|SMF})

ジャーナル・レコードを書き込む場所を指定します。5文字までの、以下の値を指定できます。

DUMMY
ログ・レコードは書き込まれません。 例えば、これを使用すると、アプリケーションを 変更したり、ファイルまたはプロファイル 資源定義を変更したりすることなく、 不要なログ・レコードを抑制することができます。

システム・ログまたはlog-of-logsを 必要としない場合、必要に応じて、DFHLOG、DFHSHUNT、およびDFHLGLOGの JOURNALMODEL定義でDUMMYを指定 します。

MVS レコードは、MVS システム・ロガーのログ・ストリームに書き込まれます。ログ・ストリームの名前は、STREAMNAME 属性で指定されます。

SMF ジャーナル・レコードは、MVS システム・ロガーのログ・ストリームではなく、MVS SMF ログに SMF フォーマットで書き込まれます。

注**:** CICS システム・ログおよび順方向リカバリー・ログでは、SMF は許可されません。

デフォルトの **JOURNALMODEL**

CICS は、インストール済みの JOURNALMODEL 定義を検出できない場合、デフォルト属性のセットを想定します。

使用される属性は、以下の「組み込み」デフォルト定義の属性です。

```
DEFINE JOURNALMODEL(OTHERS) GROUP(LOGS)
          JOURNALNAME(*)
          STREAMNAME(&USERID..&APPLID..&JNAME.)
          TYPE(MVS)
```

JOURNALNAME(*) は、一致する JOURNALMODEL エントリーがない場合に CICS がジャーナル名に使用するデフォルトのジャーナル・モデルです。

例

CICS JOURNLMODEL のリソース定義コマンドの例。

アプリケーション ID CICSHAA3 と領域ユーザー ID CICSHA## を持つ CICS AOR について、以下の定義のセットがあるとします。

```
1.  DEFINE JOURNALMODEL(USERJNL8) GROUP(LOGS)
          JOURNALNAME(DFHJ08)
          TYPE(SMF)
```

EXEC CICS WRITE JOURNALNAME(DFHJ08)... コマンドあるいは EXEC CICS WRITE JOURNALNUM(08)... コマンドを使用して自動ジャーナル 08 またはユーザー・ジャーナル 08 に書き込まれたレコードは、SMF フォーマットで MVS SMF データ・セットに書き込まれます。

```
2.  DEFINE JOURNALMODEL(USERJNL9) GROUP(LOGS)
          JOURNALNAME(DFHJ09)
          TYPE(DUMMY)
```

EXEC CICS WRITE JOURNALNAME(DFHJ09)... コマンドあるいは EXEC CICS WRITE JOURNALNUM(09)... コマンドを使用して自動ジャーナル 09 またはユー

ザー・ジャーナル 09 に書き込まれたレコードは、どのログ・ストリームにも書き込まれませんが、アプリケーション・プログラムは通常応答を受け取ります。

```
3.  DEFINE JOURNALMODEL(UJ10TO19) GROUP(LOGS)
           JOURNALNAME(DFHJ1%)
           STREAMNAME(&USERID..MERGED.USRJRNLS)
           TYPE(MVS)
```

ユーザー・ジャーナル 10 から 19 (DFHJ10 から DFHJ19) に書き込まれたレコードは、CICSHA## ユーザー ID で実行されている、同じ JOURNALMODEL がインストールされた他の CICS 領域からのレコードと一緒に、ログ・ストリーム CICSHA##.MERGED.USRJRNLS にまとめてマージされます。

```
4.  DEFINE JOURNALMODEL(LOGOFLOG) GROUP(LOGS)
           JOURNALNAME(DFHLGLOG)
           STREAMNAME(CICSVR.SHARED.DFHLGLOG)
           TYPE(MVS)
```

順方向リカバリー製品で使用するためにファイル制御および CICS ログ・マネージャーによってジャーナル DFHLGLOG に書き込まれたファイル結合レコードおよびその他のレコードは、共用ログ・ストリーム CICSVR.SHARED.DFHLGLOG に書き込まれます。このログ・ストリームは、この JOURNALMODEL リソース定義がインストールされているシスプレックス内のすべての CICS 領域によって共用されます。

```
5. DEFINE JOURNALMODEL(JNLMODL1) GROUP(LOGS)
          JOURNALNAME(USERJNL*)
          STREAMNAME(&USERID..ANYCORP.&JNAME..UK)
          TYPE(MVS)
```

文字 USERJNL から始まるいずれかのユーザー・ジャーナルまたは自動ジャーナルに書き込まれたレコードは、ログ・ストリームにまとめてマージされ、&USERID が CICS 領域ユーザー ID に置き換えられ、&JNAME がジャーナル名に置き換えられた名前が付けられます。

これらの例のみがインストールされている場合、他の形式のジャーナル処理 (PROFILE ... JOURNAL(25) で定義された端末制御の自動メッセージ・ジャーナル処理など) は、デフォルトの JOURNALMODEL と、ログ・ストリーム CICSHA##.CICSHAA3.DFHJ25 に書き込まれたレコードを使用します。 95 ページの『デフォルトの JOURNALMODEL』を参照してください。

JVMSERVER リソース

JVMSERVER リソースは、JVM サーバーのランタイム環境を定義します。

JVMSERVER リソースは、JVM プロファイルの場所と、CICS 領域内に Language Environment® エンクレーブおよび JVMサーバーを作成するために必要な Language Environment オプションを定義します。

BAS の作業については、BAS JVMSERVER リソース定義の作業を参照してください。

CICS バンドル内の JVMSERVER リソース

CICS バンドルを使用して、JVMSERVER リソース定義を作成、編集、およびインストールすることができます。この方法で JVMSERVER リソースを作成した場

合、CICS バンドルを使用してそのリソースのライフサイクルを管理する必要があり、リソースを単独で管理することはできません。

EXEC CICS INQUIRE JVMSERVER コマンドまたは **CEMT INQUIRE JVMSERVER** コマンドを使用して、CICS バンドルによって動的に生成される JVMSERVER リソースを照会することができます。

CICS バンドルによって動的に生成された JVMSERVER リソースに対して DISCARD コマンドを発行することはできません。 CICS バンドルを破棄する必要があります。そうするなら、CICS により JVMSERVER リソースに操作が適用されます。

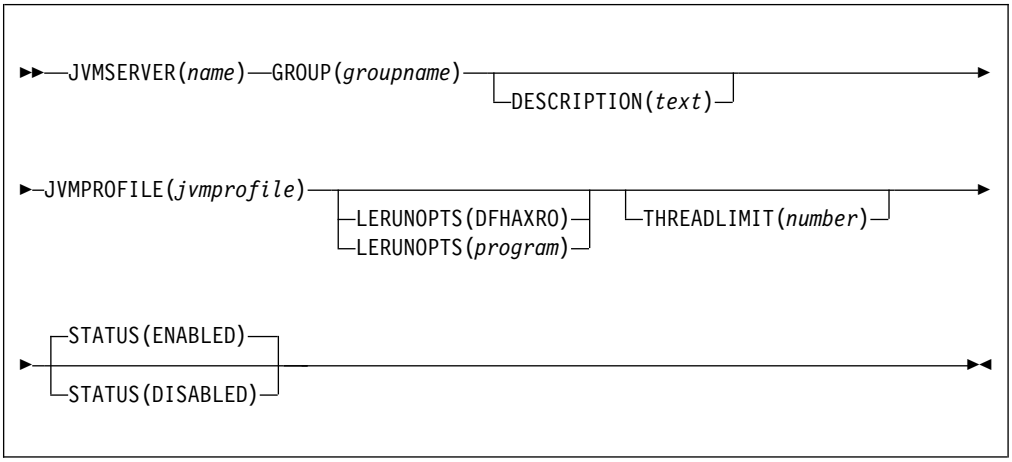
CICS バンドル内で定義されてインストールされた JVMSERVER リソースの属性を変更するには、CICS Explorer のリソース・エディターを使用して CICS バンドル内の定義を変更し、新規バージョンの CICS バンドル、またはそのデプロイに使用したアプリケーションをインストールします。SET JVMSERVER コマンドを使用することで、動的に生成されたリソースの属性を変更することはできますが、これらの変更はカタログされないため、CICS のウォーム・リスタート後には回復されません。

CICS バンドルで定義されてインストールされた JVMSERVER リソースの状況を変更するには、CICS バンドルを使用可能/使用不可にします。CICS バンドルを使用不可にしたが、JVM サーバーでまだ実行中のタスクをパージしたい場合は、動的に生成されたリソースに対して、PURGE、FORCEPURGE、または KILL オプションを指定して SET JVMSERVER DISABLED コマンドを発行することで、タスクをパージすることができます。

CICS バンドルでのリソースの定義について詳しくは、CICS バンドルの定義を参照してください。

JVMSERVER 属性

JVMSERVER リソースの構文と属性について記述します。



JVMSERVER(*name*)

JVMSERVER リソースの名前を 1 から 8 文字で指定します。

許容文字:

A-Z a-z 0-9 $ @ # . - _ % ? ! : | = , ;

文字 DFH は、CICS が使用するために予約されているため、DFH で始まる名前は使用しないことが推奨されます。

CICS は、それらを削除または破棄したり、新たに機能や修正をアップグレードあるいは提供する際に CICS リソースに置き換えたりすることができます。

DESCRIPTION(*text***)**

このフィールドには、定義するリソースの記述を提供することができます。記述テキストは、58 文字以内の長さにすることができます。使用できる文字に制限はありません。しかし、括弧を使用する場合は、左右の括弧が対になるようにしてください。CREATE コマンドを使用する場合、テキスト・コードに含まれる各アポストロフィに 2 つのアポストロフィを使用してください。

GROUP(*groupname***)**

各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字:

A-Z 0-9 $ @ #

小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。

JVMPROFILE(*jvmprofile*)

JVM サーバーを実行するための JVM オプションが含まれる JVM プロファイルの名前を 1 から 8 文字で指定します。 JVM プロファイルが含まれるファイルのファイル拡張子は、.jvmprofile でなければなりません。

- オンライン・リソース定義によって定義された JVMSERVER リソースの JVM プロファイルの場合、ファイル・パスは、システム初期設定パラメーター **JVMPROFILEDIR** によって指定されます。
- CICS バンドルで定義された JVMSERVER リソースの JVM プロファイルの場合、ファイル・パスは、CICS バンドルのルート・ディレクトリーからの相対パスです。JVM プロファイルが含まれるファイルは、JVMSERVER リソースと一緒に CICS バンドルにパッケージ化されている必要があります。詳しくは、Referencing zFS artifacts in a bundleを参照してください。

JVM プロファイル名に、単一あるいは二重のピリオドはいずれも使用することができません。 CICS Explorer を使用して CICS バンドルで JVMSERVER リソースを定義した場合、JVM プロファイル名に使用可能な文字のセットはさらに制限されます。

オンライン **JVMSERVER** リソースの **JVMPROFILE** 名に使用可能な文字:

A-Z a-z 0-9 @ # . - _ % & ? ! : | " = , ; < >

CICS バンドルで定義された **JVMSERVER** リソース定義内の **JVMPROFILE** 名に使用可能な文字**:**

```
A-Z a-z 0-9 @ # - _ % ! =
```

LERUNOPTS(**DFHAXRO**|*program*)

Language Environment エンクレーブのランタイム・オプションを定義するプログラムの名前を 1 から 8 文字で指定します。 DFHAXRO は、デフォルト値をまとめて提供するために用意されたプログラムです。いずれかの Language Environment ランタイム・オプションのデフォルトを変更したい場合、DFHAXRO のソースは *hlq*.SDFHSAMP ライブラリー内にあります。

別のプログラムを使用する場合は、そのプログラムを *hlq*.SDFHLOAD ライブラリーに置き、大文字でプログラム名を指定します。

STATUS(**ENABLED**|**DISABLED**)

JVMSERVER リソースがインストールされるときの初期状況を指定します。

ENABLED

JVM サーバー・ランタイム環境は使用可能です。

DISABLED

JVM サーバー・ランタイム環境は使用できません。

THREADLIMIT(**15**|*number*)

JVM サーバーの Language Environmentエンクレーブで許可されるスレッドの最大数を指定します。各スレッドは、 T8 TCBの下で実行します。1 から 256 スレッドの範囲で限度を指定できます。

その他のすべての使用可能/使用不可状態の JVMSERVER リソースを考慮に入れて、CICS 領域で許可されるスレッドの最大数である 2000 を超えるスレッド制限を指定した場合、CICS は、スレッド制限値の 2000 に達するまで残りのスレッドをリソースに割り振ります。 CICS が既に JVMSERVER スレッドの最大数に達している場合、リソースは、使用不可状態でインストールされます。

ヒント**:** Liberty JVM サーバーの場合は、少なくとも 5 つのスレッドを実行する必要があります。Liberty JVM サーバーに対して 5 未満の threadlimit を指定すると、Liberty が無応答にならないように threadlimit および threadcount が自動的に 5 に引き上げられます。

LIBRARY リソース

LIBRARY リソースは、動的プログラム LIBRARY 連結の物理特性および操作特性を定義するために使用します。

LIBRARY 定義には、LIBRARY リソース内のデータ・セット (複数のこともある) の名前および LIBRARY の操作状況に関するその他の詳細を提供する属性が含まれています。

DFHRPL は、稼働中の CICS システムで変更できない LIBRARY の特殊な例で、CEDA でリソースとして表示されません。

動的プログラム LIBRARY 連結内のデータ・セットは、拡張アドレス・ボリューム (EAV) DASD ボリュームの拡張アドレッシング・スペース (EAS) に常駐できます。

LIBRARY リソース用の「グループ・コミット」は存在しません。CSD グループ内の各 LIBRARY は、グループがインストールされる際に、別々にコミットされます。グループ・インストールに似た効果を得るために、「使用不可」状況を使用して一連の LIBRARY リソースをインストールすることができます。これらのリソースは、すべてのリソースがインストールされた後で、使用可能状況が「使用可能」に設定されます。

CEMT および CEDA を使用した LIBRARY リソース定義のインストールについては、LIBRARY リソース定義のインストールを参照してください。BAS の作業については、BAS LIBRARY リソース定義の作業を参照してください。

CICS バンドル内の LIBRARY リソース

CICS バンドルを使用して、LIBRARY リソース定義を作成、編集、およびインストールすることができます。この方法で LIBRARY リソースを作成した場合、CICS バンドルを使用してそのリソースのライフサイクルを管理する必要があり、リソースを単独で管理することはできません。 **EXEC CICS INQUIRE LIBRARY** コマンドまたは **CEMT INQUIRE LIBRARY** コマンドを使用して、CICS バンドルによって動的に生成される LIBRARY リソースを照会することができます。ただし、CICS バンドルによって動的に生成された LIBRARY リソースに対して SET コマンドや DISCARD コマンドを発行することはできません。これらのコマンドは、BUNDLE リソースに対して発行する必要があり、CICS がそれらを LIBRARY リソースに適用します。

CICS バンドル内のリソースの定義について詳しくは、Defining CICS bundlesを参照してください。

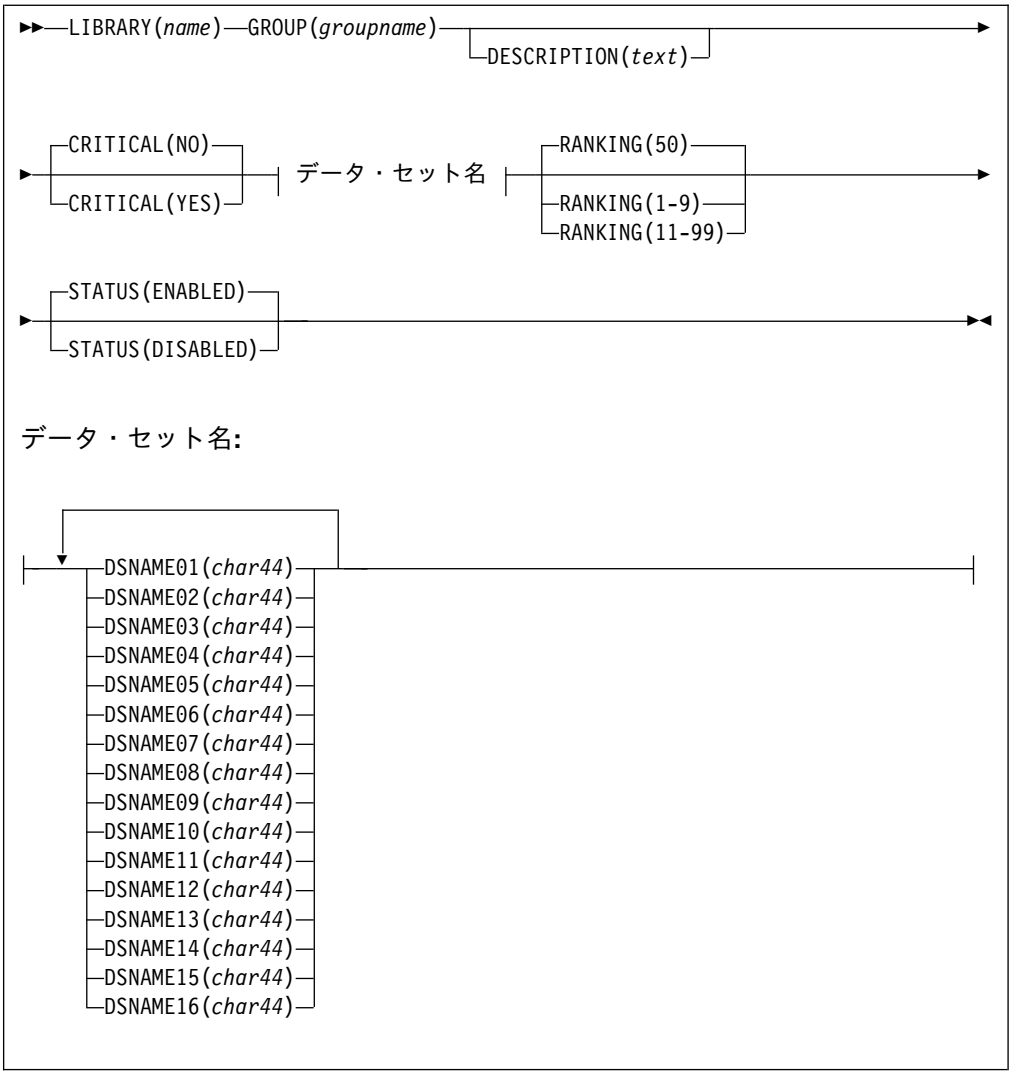
アプリケーションの専用 LIBRARY リソース

LIBRARY リソースは、アプリケーション・バージョンの専用リソースとしてサポートされます。アプリケーションの各バージョンには、そのアプリケーションのロード・モジュールを含むバージョン固有のデータ・セットを表す専用 LIBRARY リソースを組み込む必要があります。

専用リソースについて詳しくは、Characteristics of bundled resourcesを参照してください。

LIBRARY 属性

LIBRARY リソースの構文と属性を説明します。



CRITICAL({**NO**|**YES**})

この LIBRARY が CICS の実行にとってクリティカルであるかどうかを示します。始動中に LIBRARY をインストールできない場合の動作がこの属性によって決まります。インストールできない状態が発生する例として、以下のいずれかの理由で LIBRARY 定義内のデータ・セットが見つからない、または割り振ることができない場合があります。

- LIBRARY が CRITICAL の場合。
  - CICS の始動時に (グループ・リストのインストール、BAS インストール、PLTPI プログラム、またはカタログからの復元によって) LIBRARY が作成される場合、「GO or CANCEL」メッセージが発行され、クリティカル状況を無視して CICS の始動を許可するか、始動を失敗させるかを、オペレーターが決定することができます。始動を続行するという応答であった場合、LIBRARY が NONCRITICAL として再カタログされることはありませんが、将来、LIBRARY を CRITICAL と見なすべきではないと管理者が決定する場合には、クリティカル状況を NONCRITICAL に変更できます。

- CICS の始動時に LIBRARY が作成されない場合、LIBRARY のクリティカル状況はインストールの動作に影響しません。

- LIBRARY が CRITICAL 以外の場合
  - CICS の始動時に (グループ・リストのインストール、BAS インストール、PLTPI プログラム、またはカタログからの復元によって) LIBRARY が作成される場合、LIBRARY はインストールされますが無効になり、警告メッセージが発行され、CICS の始動は続行されます。後で問題の解決を試行し、LIBRARY を有効にすることができます。
  - CICS の始動時に LIBRARY がインストールされない場合、LIBRARY のクリティカル状況はインストールの動作に影響しません。

DFHRPL (静的 LIBRARY) は CRITICAL として事前定義されているため、変更できません。何か問題が発生すると、CICS の始動は失敗し、エラー・メッセージが発行されます。

アプリケーション・バンドルまたはアプリケーション・バインディング・バンドルの一部としてパッケージされてインストールされる CICS バンドルの中で定義されている専用 LIBRARY リソースの場合、CRITICAL 属性は無視されます。

DSNAME01-16(Name)
プログラム成果物が含まれ、LIBRARY 連結を作成するための最大 16 個のデータ・セットの名前。 LIBRARY がインストールされるときには、名前の順序 (例えば、DSNAME01 は DSNAME02 より前) でデータ・セットが連結されます。順番にすべてのデータ・セット属性を使用する必要はありません。例えば、DSNAME01 がブランクであってもかまいません。ただし、少なくとも 1 つのデータ・セット属性に値がなければなりません。詳しくは、データ・セット命名規則の定義を参照してください。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

LIBRARY(*name*)
LIBRARY リソースの名前。

CICS 領域、CICSPlex SM データ・リポジトリー、またはスタンドアロンの CICS バンドルの中で定義する LIBRARY リソースの場合、リソース名はデータ・セットの LIBRARY 連結に対する DD 名として使用されます。そのため、LIBRARY リソースの名前は、有効な DD 名の規則に適合していなければなりません (長さ 1 から 8 文字。先頭文字は A から Z、$、#、または @ のいずれか。残りの文字は A から Z、1 から 9、$、#、または @ のいずれか)。インストール済み環境で使用されている他の DD 名と競合する名前を使用しない

でください。選択した名前が、システム内の別の公用 LIBRARY リソースで既に使用されている DD 名と競合すると、公用 LIBRARY リソースの作成が失敗します。

アプリケーション・バンドルまたはアプリケーション・バインディング・バンドルの一部としてパッケージされてインストールされる CICS バンドルの中で定義されている専用 LIBRARY リソースの場合、LIBRARY リソースの名前は、データ・セットの LIBRARY 連結に対する DD 名としては使用されません。代わりに、CICS は、アプリケーションがプラットフォームにインストールされたときに、データ・セットの LIBRARY 連結用に固有の DD 名を要求します。したがって、リソース名は、インストール済み環境内のどこかで使用されている LIBRARY 名と同じであっても、または異なるバージョンのアプリケーションで使用されている名前と同じであってもかまいません。 CICS は、メッセージ DFHLD0518 を発行して、z/OS が LIBRARY 連結用に生成した DD 名を示します。インストールされたアプリケーションのデータ・セット名は、CICS Explorer で表示することもできます。

LIBRARY リソースに対して以下の予約名を使用することはできません。

- DFHRPL。または DFH で始まるすべての名前
- EYU で始まるすべての名前
- CEEDUMP
- DUMMY
- JOBCAT
- JOBLIB
- STEPCAT
- STEPLIB
- SYSABEND
- SYSIN
- SYSMDUMP
- SYSOUT
- SYSUDUMP

DFH または EYU で始まる LIBRARY リソース名が受け入れられないのは、将来 CICS が使用できる名前を確保するためです。 CEE で始まる LIBRARY リソース名は注意深く使用する必要がありますが、受け入れられます。 SYSIN と SYSOUT と DUMMY は、従来、DD カードとして使用されているため、LIBRARY リソース名としての使用は推奨されません。残りの DD 名は、MVS によって予約済みです。DFHRPL は、CICS 始動 JCL で静的 LIBRARY の名前として使用するために予約済みです。

RANKING(1-99)

RANKING は、LIBRARY の全体的な検索順序の中でこの LIBRARY が出現する位置を決める 10 進数が入ったフルワード・バイナリー値です。値が小さい場合、ロードするプログラムを探すために、ランキング番号が大きい他の LIBRARY リソースの前にこの LIBRARY が検索されることを示します。ランキングは LIBRARY 連結でのデータ・セットの連結番号に似ていますが、いくつかの違いがあります。

RANKING は 1 から 99 の範囲の値をとり、デフォルトは 50 です。値 10 は DFHRPL (静的 LIBRARY)のために予約されており、指定できません。

DFHRPL 連結には事前定義されたランキングとして 10 が割り当てられており、変更できません。この値を利用すると、動的に定義された LIBRARY リソースに 10 より小さいランキング値を指定することにより、それらの LIBRARY リソースを全体的な検索順序で DFHRPL 連結より前に配置することができます。

注:

- LIBRARY リソースを検索順序で DFHRPL より前に配置するランキングにすることは、一時的な状態と考える必要があります。
- 事前定義された DFHRPL ランキングの 10 は、検索順序で LIBRARY リソースの配置が DFHRPL より前にならないようにすることを意図していますが、LIBRARY リソース自体の間のランキングが重要でない場合には、DFHRPL の前に配置できる LIBRARY リソースの総数は制限されません。RANKING 値に 10 より小さい値を指定すると、この LIBRARY が検索順序で DFHRPL より前になることを知らせるメッセージが発行されます。
- ランキングの変更が有効になるのは、NEWCOPY 要求または PHASEIN 要求が発行された後です。または、まだプログラムがロードされていない場合は、プログラムが初めてロードされた時点です。
- ランキングが同じ LIBRARY の検索順序は未定義です。これは、それらの LIBRARY リソースの相対的な検索順序が重要でないという意味になります。実際には、検索順序は LIBRARY がインストールされた順序になります。
- プラットフォームにインストールされているアプリケーションの一部として CICS バンドル内で定義されている LIBRARY リソースは、そのアプリケーションのそのバージョン専用のものになります。これらの LIBRARY 連結は、アプリケーションのそのバージョンのタスクの場合にのみ検索されます。
- 複数の LIBRARY が同じランキングを持つことは可能ですが、2 つの LIBRARY リソースの相対検索位置が重要である場合には、それらに等しいランキング値を使用しないようにしてください。
- ランキングが同じである 2 つの LIBRARY リソースの検索位置は、信頼できません。検索順序は、各リソースがインストールされた時刻に関連し、コールド・スタートまたは初期始動の後、あるいは LIBRARY のランキングの変更時に変更することができます。検索順序では、最初にインストールされたリソースが他の LIBRARY の前に表示されます。

STATUS({ENABLED|DISABLED})

LIBRARY を作成した時点で有効にするか無効にするかを示します。 LIBRARY が使用可能な状態で作成された場合、CICS は、データ・セットの割り振りおよびその後の連結を試行してから、最後に LIBRARY 連結を開きます。これらのステップのいずれかが失敗すると、既に実行されたステップは取り消され、LIBRARY は無効の状態でインストールされます。どのステップで失敗したかを示すメッセージが出されます。

LIBRARY が使用不可の状態で作成された場合、CICS は、データ・セットの割り振りおよび連結を試行しません。

この属性は、CICS バンドルによって動的に生成された LIBRARY リソースでは無視されます。LIBRARY リソースの初期状況は、そのリソースを定義するバンドルの初期状況から導出されます。

LSRPOOL リソース

LSRPOOL リソースは、ローカル共有リソース (LSR) プールのサイズと特性を定義します。LSR プールは、特定のファイルに対するアクセス要求を処理するときに VSAM が使用するデータ・バッファー、ストリング、および Hiperspace™ バッファーの予備です。

Hiperspace バッファーは、MVS イメージにおけるハイパフォーマンス・ストレージ域です。この領域は、4 KB ページの読み取りおよび書き込みに使用されます。VSAM が使用する Hiperspace のタイプは、拡張ストレージ全体に存在しています。拡張ストレージは、実ストレージとの間のページングのみに使用される追加プロセッサー・ストレージです。

システムには、最大 255 個の LSR プールを同時に定義することができ、それぞれが LSRPOOLNUM によって識別されます。この LSRPOOLNUM は、ファイルを共用リソースで使用する場合に、そのファイルを LSR プールに関連付けるために使用されます。

LSRPOOL定義がアクティブなシステムにインストールされた場合、指定されたIDのプールが次に作成されるときに、情報が保管され、使用されます。プールは、特定の LSR プールを使用する最初のファイルが開かれたときに作成され、そのプールに対して現在開かれているファイルがなくなった場合にのみ、動的に割り振り解除されます。これは、LSRPOOL 定義がシステムにインストールされても、即時に有効にならない場合があることを意味します。

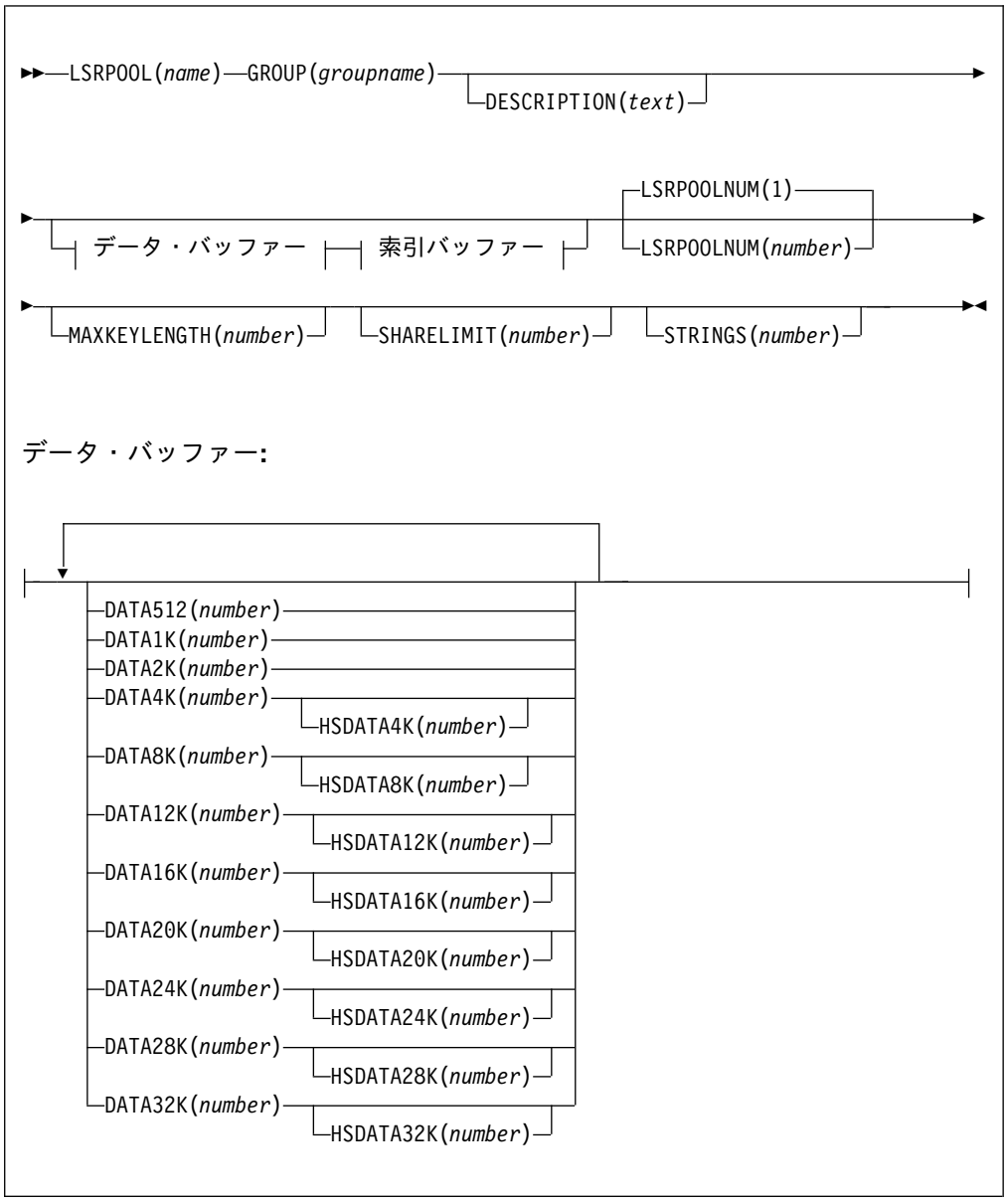
CICS は、LSRPOOL が定義されていない場合にデフォルト属性を設定します。ただし、パフォーマンス上の理由から、このような場合も LSRPOOL プールを定義することが推奨されます。例えば、実動システムでは、CICS によるプール要件の計算中に遅延が発生する場合があります。その他にも、プールの作成時にファイルが割り振られなかった場合に、CICS がデータ・セット名を認識できないという問題が発生する可能性があります。このような場合、使用可能な情報に基づいてプールが作成されますが、その後のシステム・パフォーマンスが低下したり、ファイルのオープンに失敗したりする可能性があります。

CSDLSRNO システム初期設定パラメーターを指定することで、CSD ファイルを特定の LSRPOOL に関連付けることができます。デフォルトはプール 1 です。CSD ファイルの使用を CICS に許可するのに、適切なサイズの十分なバッファーが提供されていることを確認してください。 CSDLSRNO の詳細および CSD ファイルに対するバッファー要件の計算の詳細については、Setting up shared data sets, CSD and SYSINを参照してください。

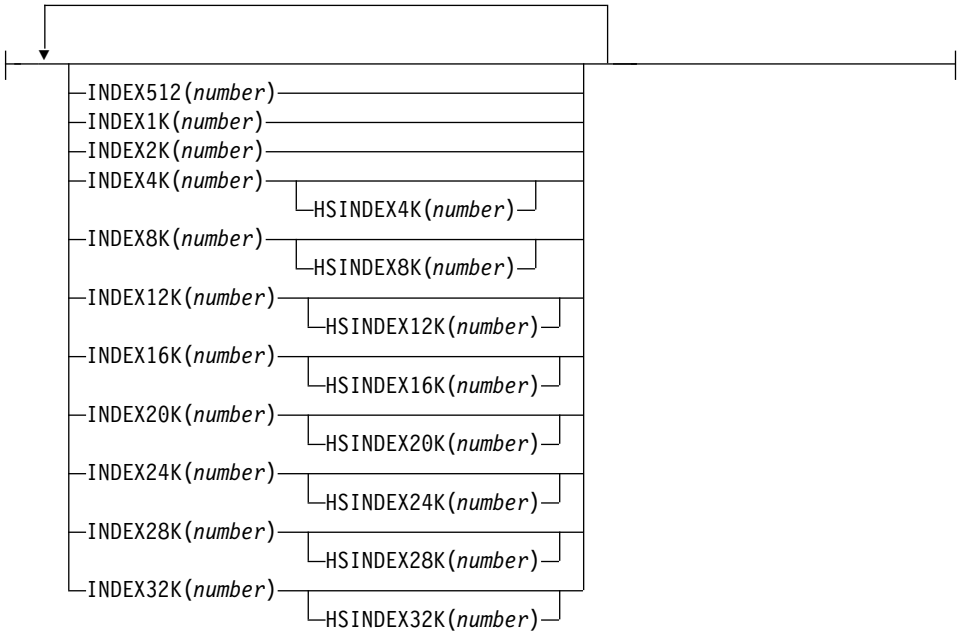
BAS の作業については、BAS LSR プール・リソース定義の作業を参照してください。

LSRPOOL 属性

LSRPOOL リソースの構文と属性を説明します。



索引バッファー:



DATA BUFFERS

必要とする各サイズのデータ・バッファーの数を 3 から 32767 の範囲で指定します。この属性にはデフォルト値がありません。

DATA512(*number*)

必要とする 512 バイトのデータ・バッファーの数を 3 から 32767 の範囲で指定します。

DATA1K(*number*)

必要とする 1 KB のデータ・バッファーの数を 3 から 32767 の範囲で指定します。

DATA2K(*number*)

必要とする 2 KB のデータ・バッファーの数を 3 から 32767 の範囲で指定します。

DATA4K(*number*)

必要とする 4 KB のデータ・バッファーの数を 3 から 32767 の範囲で指定します。

DATA8K(*number*)

必要とする 8 KB のデータ・バッファーの数を 3 から 32767 の範囲で指定します。

DATA12K(*number*)

必要とする 12 KB のデータ・バッファーの数を 3 から 32767 の範囲で指定します。

DATA16K(*number*)
必要とする 16 KB のデータ・バッファーの数を 3 から 32767 の範囲で指定します。

DATA20K(*number*)
必要とする 20 KB のデータ・バッファーの数を 3 から 32767 の範囲で指定します。

DATA24K(*number*)
必要とする 24 KB のデータ・バッファーの数を 3 から 32767 の範囲で指定します。

DATA28K(*number*)
必要とする 28 KB のデータ・バッファーの数を 3 から 32767 の範囲で指定します。

DATA32K(*number*)
必要とする 32 KB のデータ・バッファーの数を 3 から 32767 の範囲で指定します。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

HIPERSPACE DATA BUFFERS
必要とする各サイズの Hiperspace データ・バッファーの数を 0 から 16777215 の範囲で指定します。これらのフィールドをブランクのままにした場合、デフォルト値はありません。

注: 特定のサイズの Hiperspace データ・バッファーに値を指定する場合は、同じサイズのデータ・バッファーの値も指定する必要があります。

HSDATA4K(*number*)
必要とする 4 KB の Hiperspace データ・バッファーの数を 0 から 16777215 の範囲で指定します。

HSDATA8K(*number*)
必要とする 8 KB の Hiperspace データ・バッファーの数を 0 から 16777215 の範囲で指定します。

HSDATA12K(*number*)
必要とする 12 KB の Hiperspace データ・バッファーの数を 0 から 16777215 の範囲で指定します。

HSDATA16K(*number*)
必要とする 16 KB の Hiperspace データ・バッファーの数を 0 から 16777215 の範囲で指定します。

HSDATA20K(*number*)
必要とする 20 KB の Hiperspace データ・バッファーの数を 0 から 16777215 の範囲で指定します。

HSDATA24K(*number*)
必要とする 24 KB の Hiperspace データ・バッファーの数を 0 から 16777215 の範囲で指定します。

HSDATA28K(*number*)
必要とする 28 KB の Hiperspace データ・バッファーの数を 0 から 16777215 の範囲で指定します。

HSDATA32K(*number*)
必要とする 32 KB の Hiperspace データ・バッファーの数を 0 から 16777215 の範囲で指定します。

HIPERSPACE INDEX BUFFERS
必要とする各サイズの Hiperspace 索引バッファーの数を 0 から 16777215 の範囲で指定します。これらのフィールドをブランクのままにした場合、デフォルト値はありません。

注**:** 特定のサイズの Hiperspace 索引バッファーに値を指定する場合は、同じサイズの索引バッファーの値も指定する必要があります。

HSINDEX4K(*number*)
必要とする 4 KB の Hiperspace 索引バッファーの数を 0 から 16777215 の範囲で指定します。

HSINDEX8K(*number*)
必要とする 8 KB の Hiperspace 索引バッファーの数を 0 から 16777215 の範囲で指定します。

HSINDEX12K(*number*)
必要とする 12 KB の Hiperspace 索引バッファーの数を 0 から 16777215 の範囲で指定します。

HSINDEX16K(*number*)
必要とする 16 KB の Hiperspace 索引バッファーの数を 0 から 16777215 の範囲で指定します。

HSINDEX20K(*number*)
必要とする 20 KB の Hiperspace 索引バッファーの数を 0 から 16777215 の範囲で指定します。

HSINDEX24K(*number*)
必要とする 24 KB の Hiperspace 索引バッファーの数を 0 から 16777215 の範囲で指定します。

HSINDEX28K(*number*)
必要とする 28 KB の Hiperspace 索引バッファーの数を 0 から 16777215 の範囲で指定します。

HSINDEX32K(*number*)
必要とする 32 KB の Hiperspace 索引バッファーの数を 0 から 16777215 の範囲で指定します。

INDEX BUFFERS
必要とする各サイズの索引バッファーの数を 3 から 32767 の範囲で指定します。このフィールドをブランクのままにした場合、デフォルト値はありません。

INDEX512(*number*)
必要とする 512 バイトの索引バッファーの数を 3 から 32767 の範囲で指定します。

INDEX1K(*number*)
必要とする 1 KB の索引バッファーの数を 3 から 32767 の範囲で指定します。

INDEX2K(*number*)
必要とする 2 KB の索引バッファーの数を 3 から 32767 の範囲で指定します。

INDEX4K(*number*)
必要とする 4 KB の索引バッファーの数を 3 から 32767 の範囲で指定します。

INDEX8K(*number*)
必要とする 8 KB の索引バッファーの数を 3 から 32767 の範囲で指定します。

INDEX12K(*number*)
必要とする 12 KB の索引バッファーの数を 3 から 32767 の範囲で指定します。

INDEX16K(*number*)
必要とする 16 KB の索引バッファーの数を 3 から 32767 の範囲で指定します。

INDEX20K(*number*)
必要とする 20 KB の索引バッファーの数を 3 から 32767 の範囲で指定します。

INDEX24K(*number*)
必要とする 24 KB の索引バッファーの数を 3 から 32767 の範囲で指定します。

INDEX28K(*number*)
必要とする 28 KB の索引バッファーの数を 3 から 32767 の範囲で指定します。

INDEX32K(*number*)
必要とする 32 KB の索引バッファーの数を 3 から 32767 の範囲で指定します。

LSRPOOL(*name*)
定義するローカル共用リソース・プールの名前を指定します。名前の長さは、最大で 8 文字です。

DATA BUFFERS のみを指定する場合、VSAM KSDS データ・セットの索引コンポーネントとデータ・コンポーネントの両方に使用されるプールとして、1 つのバッファー・セットが作成されます。

データ・バッファーを指定しない場合、CICS によりデータ・コンポーネントと索引コンポーネントの両方に必要なバッファーが計算されます。両コンポーネントとも、同じバッファー・セットを共用します。

INDEX BUFFERS を指定する場合、プールが 2 つに分けられて作成されます。 1 つはデータ用、もう 1 つは索引バッファー用です。 INDEX BUFFERS を指定した場合は、DATA BUFFERS も指定する必要があります。

LSRPOOLID({**1**|*lsrpool*})
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

既存の定義に指定されている LSRPOOLID の値は、新しいオプション LSRPOOLNUM に移されます。

LSRPOOLNUM({**1**|*number*})
定義するローカル共用リソース・プールの ID を指定します。値の範囲は 1 から 255 までです。

MAXKEYLENGTH(*number*)
リソースを共用する任意のファイルの最大キー長を指定します。値の範囲は 0 から 255 までです。この値は、CICS リソース計算の部分を上書きします。ユーザーがこの値を指定しない場合、CICS により最大キー長が決定されます。LSR が再作成される度にこの値は再計算されます。

SHARELIMIT(*number*)
割り振る VSAM リソースの最大量のパーセントを整数で指定します。この数値は 1 から 100 までの値です。

リソースを共用する VSAM ファイルに必要なリソースの最大量を CICS が計算する場合に、SHARELIMIT を指定します。これらのリソースは共用されるので、リソースの最大量の何パーセントかを割り振る必要があります。この属性を省略すると、リソースの最大量の 50% が割り振られます。

STRINGS 属性と SIZE 属性の両方を指定する場合、SHARELIMIT は影響を与えません。

STRINGS(*number*)
プール内のファイルのすべてのストリングの限界を、1 から 255 の範囲で指定します。詳しくは、LSR および NSR プールのバッファーおよびストリングの数を参照してください。

MAPSET リソース

MAPSET リソースは BMS マップ・セットを定義します。

ディスプレイ装置を使用する対話式アプリケーションごとに、固有の画面レイアウトやマップを使用できます。これらはプログラム自体には指定しません。基本マッピング・サポート(BMS) を使用します。BMSは柔軟性に富み、複数の同一プログラムや複数の異なるプログラムでマップを使用することができます。DFHMSD, DFHMDI,およびDFHMDFマクロを使用して、マップとマップ上のフィールドを指定します。詳しくは、マップの作成 を参照してください。

BMS マップ定義マクロを使用する代わりに、表示画面定義機能 II (SDF II) ライセンス・プログラムを使用して対話式にマップを定義することができます。このライセンス・プログラムのプログラム番号は、5665-366 (MVS 用) および 5664-307 (VM 用) です。SDF II を使用することで、画面を対話式に表示設定することができます。その後、画面を生成して、CICS/BMSマップ・セットの画面を取得できます。また、SDF IIのテスト機能では、マップをランタイム表示で参照することもできます。背景情報については、Screen Definition Facility II Libraryを参照してください。

アプリケーションは、ユーザーとの対話中、異なる時間に一連の関連マップを使用できます。また、同じ画面で同時に複数の関連マップを使用して、完全な画面を完成させることもできます。

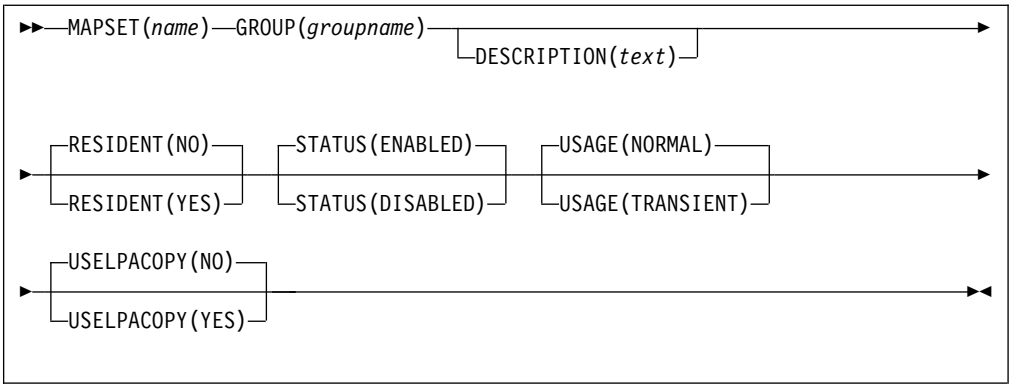
これらの関連マップは、MAPSET定義に指定したマップ・セットに属します。プログラムにマップが1つしか定義されていなくても、マップはマップ・セットに属します。MAPSETを定義するには、CEDA またはDFHCSDUPを使用するか、該当するシステム初期設定パラメーターを自動インストール使用可能に設定します。自動インストールについて詳しくは、『Configuring』の『Autoinstalling programs, map sets, and partition sets』 を参照してください。

プログラムとマップ・セットの間には資源定義へのリンクがありません。したがって、プログラムのBMS SEND MAP およびRECEIVE MAPコマンドでMAPSET名を指定します。

BAS の作業については、BAS マップ・セット・リソース定義の作業を参照してください。

MAPSET 属性

MAPSET リソースの構文および属性について説明します。



DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

MAPSET(*name*)
この MAPSET 定義の名前を指定します。名前の最大長は 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

DFH から始まるマップ・セット名は使用しないでください。これらの文字は、CICS が使用するために予約されているからです。

BMSの装置依存のマップ・セットの場合、 マップ・セット名は、元のマップ・セット名 (1から7文字）にマップ・セット接頭部を追加 して派生したものである必要があります。接尾部は、 マップ・セットを定義したDFHMSDマクロ命令の TERMオペランドに指定されているパラメーター によって異なります。

装置依存の接尾部を使用するには、システム初期設定パラメーターとして BMS=(,,DDS) を指定する必要があります。マップ・セットの接尾部については、装置依存サポート を参照してください。

RESIDENT({NO|YES})
マップ・セットの常駐状況を指定します。

NO マップ・セットは永続的には常駐 しません。

YES マップ・セットは最初の参照時に ロードされ、その後仮想ストレージに 永続的に常駐しますが、オペレーティング・ システムによってページング可能です。

RSL
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。詳しくは、Obsolete attributesを参照してください。

STATUS({ENABLED|DISABLED})
マップ・セットの状況を指定します。

DISABLED
マップ・セットは使用されません。

ENABLED
マップ・セットは使用されます。

USAGE({NORMAL|TRANSIENT})
このマップ・セットのストレージがいつ解放されるかを指定します。

NORMAL
マップ・セットの使用回数がゼロになると、 このマップ・セットのストレージは、通常の 動的ストレージ圧縮処理の一部として ストレージからの削除の対象となります。

TRANSIENT
マップ・セットの使用回数がゼロになると、この マップ・セットのストレージは解放されます。この値は、参照頻度の低いマップ・セットに対して指定してください。

USELPACOPY({NO|YES})
リンク・パック域 (LPA) からマップ・セットを使用するかどうかを指定します。

NO マップ・セットはLPAからは使用されません。これは、CICS 区分画面にロードされます。

YES システム初期設定パラメーターにLPA=YESが指定されている場合、 マップ・セットをLPAから使用できます。LPAからマップ・セット を使用するには、マップ・セットがLPAにインストール済みで、 PRVMOD始動オプションでマップ名が指定されていない必要があります。詳しくは、MVS リンク・パック域からのモジュール使用の制御 を参照してください。

MQCONN リソース

MQCONN リソースは、CICS と IBM MQ の間の接続の属性を定義します。MQCONN リソースをインストールまたは破棄することができるのは、CICS が IBM MQ に接続されていない場合のみです。

CICS 領域に一度にインストールすることができる MQCONN リソースは 1 つのみです。MQCONN リソースを定義する場合、CICS は、グループあるいはリスト内で定義されている MQCONN リソースが 1 つだけであることを確認します。複数見つかると、たとえ名前が異なるとしても警告メッセージが発行されます。2 つ目の MQCONN リソースをインストールする場合、同じ名前の MQCONN リソースを再インストールする場合を除き、CICS は、暗黙的に既存の MQCONN リソースを破棄してからインストールを続行します。

MQCONN リソースのインストールが完了したら、**CEMT SET MQCONN** コマンドまたは **EXEC CICS SET MQCONN** コマンド、CICSPlex SM、あるいは CICS Explorer を使用して、CICS と IBM MQ の間の接続を開始することができます。

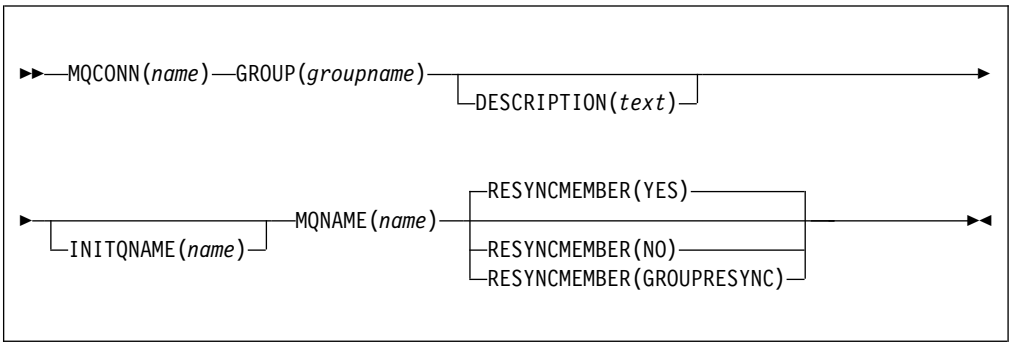
INITQNAME 属性の値を指定する MQCONN リソースをインストールすると、CICS もデフォルトの開始キューを表す予約済みの名前 DFHMQINI を使用して、動的に MQMONITOR リソースを作成してインストールします。

DFHMQINI について詳しくは、MQMONITOR リソースを参照してください。

. BAS の作業については、BAS MQCONN リソース定義の作業を参照してください。

MQCONN 属性

MQCONN リソースの構文と属性について説明します。



属性

DESCRIPTION(*text*)
このフィールドには、定義するリソースの記述を提供することができます。記述テキストは、58 文字以内の長さにすることができます。使用できる文字に制限はありません。しかし、括弧を使用する場合は、左右の括弧が対になるようにしてください。CREATE コマンドを使用する場合、テキスト・コードに含まれる各アポストロフィに 2 つのアポストロフィを使用してください。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。

INITQNAME(*name*)
この CICS-MQ 接続のデフォルトの開始キューの名前 (1 から 48 文字) を指定します。

許容文字: A-Z a-z 0-9 . / _ %

INITQNAME 属性を指定する場合、シンボリック・パラメーター *&APPLID.* を使用して、CICS 領域のアプリケーション ID を識別することができます。例えば、&APPLID..EXAMPLE.QUEUE.NAME のようにします。CICS 領域にリソースをインストールする場合、*&APPLID.* は、ローカル領域のアプリケーション ID に置き換えられます。

MQCONN リソース定義をインストールする際に非ブランクの INITQNAME 属性を指定した場合、CICS は、デフォルトの開始キューを表す予約済みの名前 DFHMQINI を使用して MQMONITOR リソースをインストールします。DFHMQINI の QNAME 属性は、デフォルトの開始キュー名を指定します。QNAME 属性を変更したい場合は、MQCONN リソース定義の INITQNAME 属性を変更してから MQCONN リソース定義を再インストールする必要があります。

CKQC START コマンドを使用して CICS-MQ 接続を開始する際にデフォルト開始キューの名前を指定した場合、インストール済みの MQCONN リソース定義の INITQNAME 値、およびインストール済みの MQMONITOR リソース定義の QNAME 値が、指定した設定に置き換わります。

MQCONN(*name*)
このリソース定義の 8 文字の名前を指定します。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

MQNAME(*name*)
1 つの IBM MQ キュー・マネージャー、または IBM MQ キュー・マネージャーのキュー共用グループの名前 (1 から 4 文字) を指定します。

- 1 つのキュー・マネージャーを指定した場合は、CICS-MQ 接続を開始すると、CICS は、そのキュー・マネージャーだけに接続します。
- キュー共用グループを指定した場合は、接続を開始すると、CICS は、同じ LPAR にあるそのグループのいずれかのアクティブ・メンバーに接続しま

す。 RESYNCMEMBER 属性を使用すると、CICS がキュー共用グループ内から最後に接続したキュー・マネージャーで未処理の作業単位が保留になっている場合の動作を指定できます。

- RESYNCMEMBER(GROUPRESYNC) を指定する場合、MQNAME はキュー共用グループの名前でなければなりません。

許容文字:
A-Z 0-9 $ @ #

キュー・マネージャーまたはキュー共用グループの名前の先頭で数字を使用することはできません。

IBM MQ では、キュー共用グループ名の長さが 4 文字未満である場合、埋め込み文字として @ 記号が使われます。 CICS では、埋め込みが行われた後にキュー共用グループの照会が行われるため、4 文字未満のキュー共用グループ名を指定する場合は、末尾に @ 記号を付けないでください。例えば、キュー共用グループが DEV@ という名前で IBM MQ に対して定義されている場合、MQNAME 属性の値として DEV を指定します。

CEMT または EXEC の CICS SET MQCONN コマンドで IBM MQ の代替キュー・マネージャーまたは代替キュー共用グループの名前を指定したり、CKQC START コマンドを使用して CICS-MQ 接続を開始するときにキュー・マネージャーの名前を指定したりすると、CICS は、その代替キュー・マネージャーまたは代替キュー共用グループに接続します。さらに、インストール済みの MQCONN 定義の MQNAME 属性の設定は、そのコマンドで指定したキュー・マネージャーまたはキュー共用グループの名前に置き換えられます。元のキュー・マネージャーまたはキュー共用グループに戻す場合、MQNAME を再び設定します。

RESYNCMEMBER({**YES**|**NO**|**GROUPRESYNC**})

この属性は、MQNAME 属性を使用して IBM MQ キュー共用グループを指定した場合にのみ適用されます。RESYNCMEMBER では、CICS がキュー共用グループ内から最後に接続したキュー・マネージャーで未処理の作業単位が保留になっている場合に CICS が採用する方針を指定します。

RESYNCMEMBER の設定を変更する操作は、すべてのリソースが整合状態になっている (つまり、未確定の作業単位が保留になっていない) 場合にのみ実行する必要があります。そうでないと、CICS は、IBM MQ の作業単位を再同期できません。 RESYNCMEMBER の値を GROUPRESYNC に変更したり GROUPRESYNC から変更したりする前に、すべてのリソースが整合状態になっていることを確認するのは重要です。

YES CICS は、同じキュー・マネージャーに接続し、必要な場合はそのキュー・マネージャーがアクティブになるまで待機して、未確定の作業単位を解決します。YES がデフォルトです。

NO CICS は同じキュー・マネージャーへの接続を一回試行します。 CICS は、この試行に失敗すると、キュー共用グループの任意のメンバーに接続し、未処理の作業単位に関する警告メッセージ DFHMQ2064 を出します。

GROUPRESYNC

CICS は、キュー共用グループの任意のメンバーに接続します。キュー・マネージャーは IBM MQ によって選択され、キュー共用グループ内のすべての適格キュー・マネージャーの代わりに、未確定の作業単位を解決することを CICS に要求します。この機能のことをグループ・リカバリー単位 といいます。 GROUPRESYNC オプションは、CICS についてグループ・リカバリー単位をサポートしているバージョン 7.1 以降の WebSphere® MQ を使用している場合、かつキュー・マネージャーによってグループ・リカバリー単位の使用を有効化している場合にのみ使用できます。

GROUPRESYNC オプションの主な適用先は共用キューですが、専用キューにも適用できます。共用キューのメッセージは即座に解決されますが、専用キューの解決は、そのキューを所有するキュー・マネージャーが再始動されたときに行われます。専用キューを使用する場合は、QSGDISP(GROUP) でキューを定義すると便利です。
QSGDISP(GROUP) を使用すると、キュー共用グループ内のすべてのマネージャー上に専用キューのインスタンスがそれぞれ 1 つずつ存在することになるため、障害が発生して別のキュー・マネージャーに再接続した場合にも新しいキュー・インスタンスを使用してアプリケーションを実行し続けることができるようになります。アプリケーションは、新しいキュー・インスタンスの使用を許容するように設定しておく必要があります。

IBM MQ で作業単位が未処理になっている場合は、RESYNCMEMBER の設定を変更しないでください。変更すると作業単位を解決できなくなるからです。 CICS で保留になっている作業単位は、リソース・マネージャーの修飾子で特定できます。 RESYNCMEMBER(GROUPRESYNC) を使用する場合は、キュー共用グループの名前が修飾子になり、そうでない場合は、個々のキュー・マネージャーの名前が修飾子として使用されます。

未確定の状態で中断された作業単位はこのプロセスに含まれません。これは、CICS 自体がこの時点でこれらの作業単位を解決できないためです。これらの作業単位の再同期は、 CICS がそのリモート・コーディネーターと再同期したときに行われます。

CICS システム定義データ・セット (CSD) を旧リリースの CICS と共用する場合は、RESYNCMEMBER(GROUPRESYNC) を指定した MQCONN 定義が RESYNCMEMBER(NO) としてインストールされます。

MQMONITOR リソース

MQMONITOR リソースは、トリガー・モニター・トランザクション CKTI など、IBM MQ メッセージ・コンシューマーの属性を定義します。MQMONITOR の使用により、MQ キューにモニタリング・サービスを提供する関連トランザクションは、IBM MQ キュー・マネージャーへの接続が確立されたときに自動的に再始動できます。

MQMONITOR リソースを使用して、MQ キュー (開始キューを含む) をモニターすることができます。複数の MQMONITOR リソースで MQ キューあるいは MQ 開始キューをモニターすることができます。

インストールおよび開始された MQMONITOR リソースは、リソース属性 **QNAME** で指定された MQ キュー、あるいは **MONDATA** 属性によってトランザクションに提供された MQ キューをモニターします。

CICS が MQMONITOR に関連付けられたトランザクションを自動的に始動できるようにするには、属性 AUTOSTART(YES) および STATUS(ENABLED) を指定して MQMONITOR がインストールされている必要があります。 CICS と MQ の接続が確立された後、MQCONN リソースを CONNECTED に設定したタスクに関連付けられているユーザー ID に関連するトランザクションを開始するための十分な権限がある場合、CICS は、属性 AUTOSTART(YES) および STATUS(ENABLED) を指定してインストールされた MQ モニターを開始します。

MQMONITOR は、CICS が IBM MQ キュー・マネージャーから切断されると、自動的に停止します。

z/OS ワークロード・マネージャーの正常性サービスが CICS 領域でアクティブになっている場合、MQMONITOR は、領域の正常性の状態に反応し、z/OS ワークロード・マネージャーの正常性サービスのオープン状況の設定の影響を受ける可能性があります。詳しくは、MQMONITOR に対する z/OS ワークロード・マネージャー正常性サービスの影響を参照してください。

CICS が MXT 状態を検出すると、この状態が続く間は、MQMONITOR が 1 秒間に発行できる MQGET 呼び出しの回数が制限されます。詳細については、アラート・モニター (CKAM)を参照してください。

注**:** MXT ゲートが発生したかどうかを調べるには、トランザクション統計の解釈を参照してください。

MQ 開始キューをモニターするために MQMONITOR リソースを定義する場合は、**TRANSACTION** 属性が CKTI を指定するかブランクにし、**QNAME** 属性でモニター対象の MQ 開始キューの名前を指定する必要があります。

注**:**

- 既に MQCONN がインストールされている場合のみ、MQMONITOR をインストールできます。その方法については、MQMONITOR リソースの定義とインストールを参照してください。
- MQMONITOR、CICS を定義すると、CICS は、同じグループあるいはリスト内の他のリソース定義との整合性を検査します。詳細については、MQMONITOR リソースの定義の検査を参照してください。

BAS の作業については、BAS MQMONITOR リソース定義の作業を参照してください。

セキュリティーの考慮事項

トランザクションのユーザー **ID**

セキュリティー検査がアクティブである (つまり、**SEC** システム初期設定パ

ラメーターが YES に設定されている) 場合、トランザクションに関連付けるユーザー ID は、MQMONITOR リソースの **MONUSERID** 属性から取得されます。

セキュリティー検査が無効にされている (つまり、**SEC** が NO に設定されている) 場合、MQ モニター・トランザクションに関連付けるユーザー ID は、MQMONITOR リソースの状態を「開始」に設定したトランザクションのユーザー ID です。

セキュリティー検査は、**MQ** モニターの状態を「開始」に設定する際に **CICS** によって実行されます。

セキュリティー検査がアクティブである場合、MQ モニターの状態を「開始」に設定する際に、MQ モニターの状態を「開始」に設定しようとしたトランザクションのユーザー ID に対して CICS が以下のようにセキュリティー検査を実行します。

注: CICS Explorer などの CICSPlex SM API インターフェースで MQ モニターの状態を設定する場合は、MQ モニター・トランザクションに関連付けるユーザー ID として、領域ユーザー ID または PLTPIUSR ユーザー ID (指定されている場合) を使用してください。

- CICS は、ユーザー ID が **MONUSERID** で定義されたユーザー ID の代理であるかを検証します。
- CICS は、ユーザー ID が **MONUSERID** で定義されたユーザー ID に関連付けられているトランザクションの開始を許可されているかを検証します。

DFHMQINI

INITQNAME 属性の値を指定する MQCONN リソースをインストールすると、CICS もデフォルトの開始キューを表す予約済みの名前 DFHMQINI を使用して、動的に MQMONITOR リソースを作成してインストールします。

DFHMQINI には以下の属性があります。

QNAME
開始キュー名を指定します。

TRANSACTION
CKTI (デフォルト)

MONUSERID
この属性は、以下のように取得されます。

- **PLTPIUSR** システム初期設定パラメーターから (使用可能な場合)
- それ以外の場合は、CICS 領域ユーザー ID から

USERID
この値は、CICS のデフォルト・ユーザー ID です。

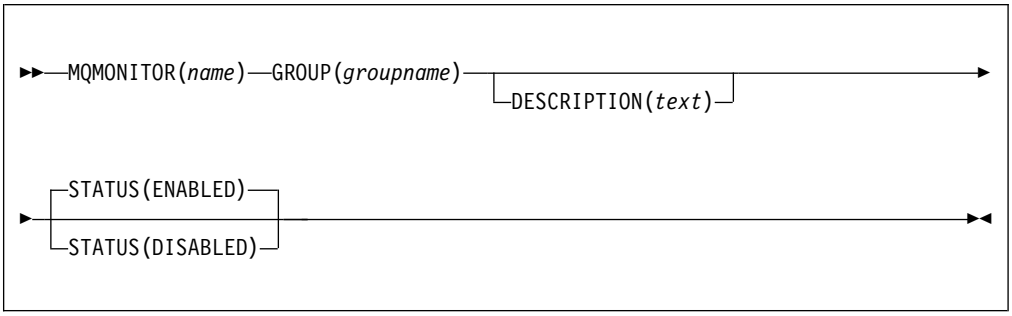
DFHMQINI は、MQ 接続が確立されると自動的に開始されます。セキュリティー検査がアクティブな場合は、CKTI トランザクションに関連付けられたユーザー ID

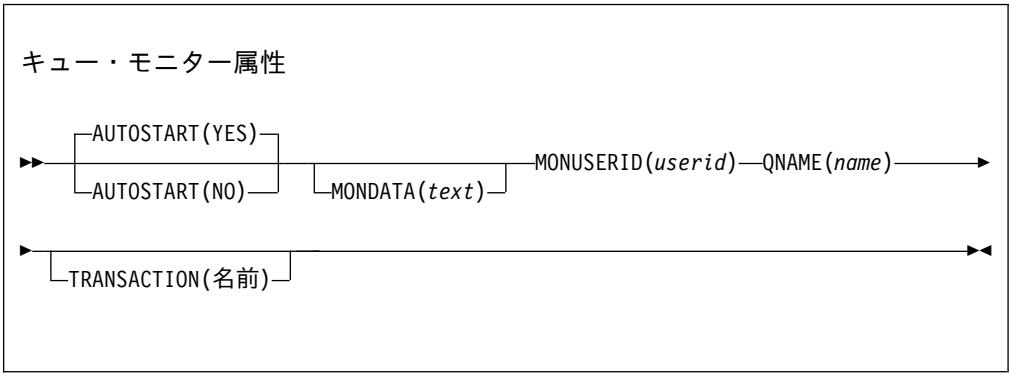
が **MONUSERID** から取得されます。セキュリティー検査が無効にされている場合は、MQMONITOR リソースの状態が「開始」に設定されたトランザクションのユーザー ID から取得されます。

EXEC CICS INQUIRE MQMONITOR コマンドまたは **CEMT INQUIRE MQMONITOR** コマンドを使用して、動的に作成された MQMONITOR リソースの **QNAME** 属性について照会することができます。この **QNAME** 値を変更したい場合は、最初に MQCONN リソースの **INITQNAME** 属性を変更してから、MQCONN リソースを再インストールする必要があります。 **INITQNAME** 属性の設定が含まれている MQCONN リソースを破棄すると、動的に作成された MQMONITOR リソースおよびすべてのユーザー定義 MQMONITOR リソースも破棄されます。

MQMONITOR 属性

MQMONITOR リソースの構文および属性について説明します。





属性

AUTOSTART({**YES**|**NO**})

IBM MQ キューへの接続が確立された場合に、自動的に MQ モニターを開始するかどうかを指定します。

YES IBM MQ キューへの接続が確立されたら、自動的に MQ モニターを開始します。

NO MQ モニターを自動的に開始しません。

DESCRIPTION(*text*)

このフィールドには、定義するリソースの記述を提供することができます。記述テキストは、58 文字以内の長さにすることができます。使用できる文字に制限はありません。しかし、括弧を使用する場合は、左右の括弧が対になるようにしてください。CREATE コマンドを使用する場合、テキスト・コードに含まれる各アポストロフィに 2 つのアポストロフィを使用してください。

GROUP(*groupname*)

各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。

MQMONITOR(*name*)

このリソース定義の 8 文字の名前を指定します。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

MONDATA(*text*)

MQ キューをモニターしているトランザクションに渡すデータを指定することができます。例えば、このフィールドを使用して、MQ ブリッジ・モニター・トランザクション CKBR にパラメーターを渡すことができます。属性には、大/小文字混合で最大 200 文字を使用することができます。

注:

モニター・タスクによって表示され、取得される場合、MONDATA データの前に以下の 18 バイトが付加されます。

Byte 1: < (left chevron)

バイト 2 から 9: *MQMONITOR resource name*

バイト 10 から 17: *USERID*

Byte 18: > (right chevron)

バイト 19 から 218 には、ユーザーが入力した MONDATA が入ります。

したがって、ユーザー作成プログラムでは、**MONDATA** を取得するために最大長 218 バイトを使用可能にする必要があります。また、取得された **MONDATA** からのバイト 2 から 9 で指定された MQMONITOR 名を使用して、MQ モニターの MONSTATUS 属性を設定することにより、現在の状態を示す必要があります。また、セキュリティー検査がアクティブである場合、CICS は、MQ モニター状態を開始済みに設定しようとするトランザクションに関連付けられているユーザー ID でセキュリティー検査を実行することにも注意してください。詳細については、MQMONITOR リソースで説明するセキュリティーの考慮事項を参照してください。

ユーザー定義文字ストリングの `&applid.` または `&APPLID.` は、MQMONITOR がインストールされるとローカル領域の APPLID に置き換えられます。そのため、このリソースを総称で簡単に使用できます。

MONUSERID(*userid*)
MQ キューをモニターしているトランザクションに関連付けるユーザー ID を指定します。 この属性は、セキュリティー検査がアクティブである (つまり、**SEC** システム初期設定パラメーターが YES に設定されている) 場合にのみ有効です。セキュリティー検査が無効にされている (つまり、**SEC** が NO に設定されている) 場合、MQ モニター・トランザクションに関連付けるユーザー ID は、MQMONITOR リソースの状態を「開始」に設定したトランザクションのユーザー ID です。

注**:** セキュリティー検査がアクティブである場合、CICS は、MQ モニターの状態を「開始」に設定するトランザクションに関連付けられたユーザー ID が、**MONUSERID** で定義されたユーザー ID の代理であり、**MONUSERID** に関連付けられたトランザクションの開始を許可されているかを検証します。CICS Explorer などの CICSPlex SM API インターフェースで MQ モニターの状態を設定する場合は、MQ モニター・トランザクションに関連付けるユーザー ID として、領域ユーザー ID または PLTPIUSR ユーザー ID (指定されている場合) を使用してください。

QNAME(*name*)
モニターする MQ キューの名前を指定します。

QNAME 属性を指定する場合、シンボリック・パラメーター *&APPLID.* を値の接頭部として使用して、CICS 領域のアプリケーション ID を識別することができます。例えば、`&APPLID..EXAMPLE.QUEUE.NAME` のようにします。 CICS 領域にリソースをインストールする場合、*&APPLID.* は、ローカル領域のアプリケーション ID に置き換えられます。

QNAME を省略した場合、この値は、リソースのインストール時にデフォルトで `&APPLID..INITIATION.QUEUE` に設定されます。

STATUS({**ENABLED**|**DISABLED**})
MQMONITOR をインストールするときの初期状況を指定します。

DISABLED
MQMONITOR リソースを使用することはできません。

ENABLED
MQMONITOR リソースを使用することができます。

TRANSACTION(名前)
MQ キューをモニターしているタスクが使用する CICS トランザクションの 4 文字の ID を指定します。

TRANSACTION を省略した場合、この値は、デフォルトで CKTI に設定されます。

USERID(*id*)
他のソースに使用可能な適切なユーザー ID がない場合に、アプリケーションに関連付けられたトランザクションを開始するためにデフォルトで使用されるユーザー ID を指定します。

USERID を省略した場合、CICS デフォルト・ユーザー ID が使用されます。

PARTITIONSET リソース

PARTITIONSET リソースは区分セットを定義します。これは、表示画面をどのように区切るかを CICS に対して記述するテーブルです。

区分セットは、一連のコマンドをコード化およびアセンブルして作成します。

一部のディスプレイ端末 (8775 ディスプレイ端末や IBM 3290 パネル表示装置など) の画面区域は、複数の異なる小さなディスプレイとして扱うことができる区分画面に分割することができます。トランザクション内の異なるプログラムまたはプログラム・ステップは、異なる区分からの書き込みや入力の受け取りが可能です。

区分セットは、DFHPSD マクロおよび DFHPDI マクロを使用して設定します。これらは、区分セット定義 にプログラミングの目的で説明されています。

異なる区分構成をPARTITIONSETとして指定します。 PARTITIONSET定義は、CEDAまたはDFHCSDUPを使用して作成されます。または、適切なシステム初期設定パラメーターが設定されている場合、自動インストールされます。詳しくは、『Configuring』の『Autoinstalling programs, map sets, and partition sets』 を参照してください。

トランザクションをTRANSACTION定義で使用する場合、 PARTITIONSETに名前を付けることができます。トランザクションが開始されると、ディスプレイ装置の内部バッファーに情報がロードされます。

あるいは、PARTITIONSET を指定しない場合、CICS は、トランザクションが開始される前にディスプレイ装置をその基本状態に設定します。

トランザクションでは、BMS SEND PARTNSET コマンドを発行して、CICS が PARTITIONSET をロードしたり、新しい PARTITIONSET に変更したりすることを要求する場合があります。これにより、その定義がアクティブ CICS システムにインストール済みの場合は、動的に区分セットがロードされます。

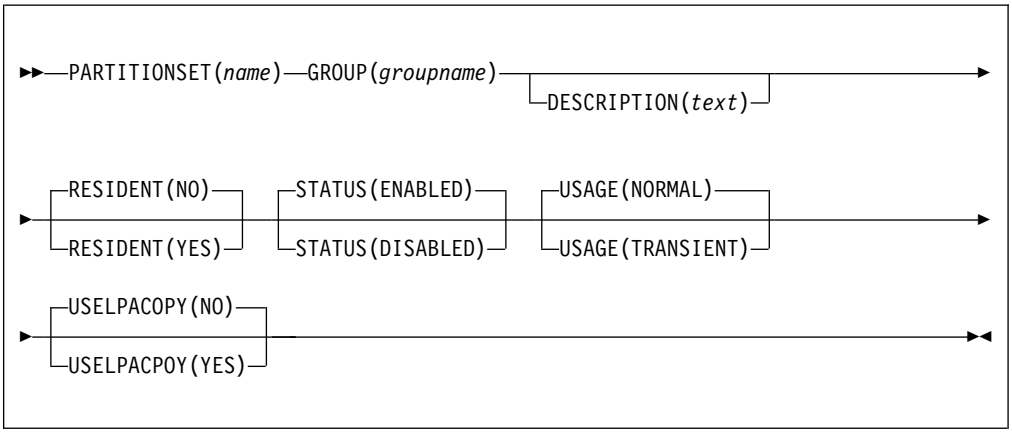
BMS 区分画面定義マクロを使用する代わりに、表示画面定義機能 II (SDF II) ライセンス・プログラムを使用して対話式に区分画面を定義することもできます。このライセンス・プログラムのプログラム番号は、5665-366 (MVS 用) および 5664-307 (VM 用) です。 SDF II を使用することで、画面を対話式に表示設定することができます。その後、画面を生成して、 CICS/BMS区分セットの画面を取得できます。

また、 SDF IIのテスト機能は、区分をランタイム表示で参照することもできます。一般情報については、Screen Definition Facility II Libraryを参照してください。

BAS の作業については、BAS 区分セット・リソース定義の作業を参照してください。

PARTITIONSET 属性

PARTITIONSET リソースの構文および属性について説明します。



DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

PARTITIONSET(*name*)
この PARTITIONSET 定義の名前を指定します。名前の最大長は 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

文字 DFH は、CICS が使用するために予約されているため、DFH で始まる名前は使用しないことが推奨されます。

CICS は、それらを削除または破棄したり、新たに機能や修正をアップグレードあるいは提供する際に CICS リソースに置き換えたりすることができます。

装置依存の区分セットの場合、 区分セット名は、元の区分セット名(1から6文字）に 区分セット接尾部を追加したものである必要があります。 接尾部は、区分セットを定義したDFHPSDマクロ命令の SUFFIXオペランドに指定されているパラメーターによって 異なります。

装置依存の接尾部を使用するには、システム初期設定パラメーターとして BMS=(,,,DDS) を指定する必要があります。

RESIDENT({NO|YES})
区分セットの常駐状況を指定します。

NO 区分セットは永続的には常駐しません。

YES 区分セットは最初の参照時にロードされ、 その後、仮想ストレージに永続的に常駐しますが、 オペレーティング・システムによってページング可能です。

RSL
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。詳しくは、Obsolete attributesを参照してください。

STATUS({ENABLED|DISABLED})
区分セットの状況を指定します。

DISABLED
区分セットは使用できません。

ENABLED
区分セットは使用できます。

USAGE({NORMAL|TRANSIENT})
この区分セットのストレージがいつ解放されるかを指定します。

NORMAL
この区分セットの使用回数がゼロになると、 この区分セットは、通常の動的プログラム圧縮処理の一部として ストレージからの削除が適格になります。

TRANSIENT
この区分セットの使用回数がゼロになると、 この区分セットのストレージは解放されます。この値は、参照頻度の低い区分セットに対して指定してください。

USELPACOPY({NO|YES})
リンク・パック域 (LPA) から区分セットを使用するかどうかを指定します。

NO 区分セットはLPAからは使用されません。これは、CICS 区分画面にロードされます。

YES システム初期設定パラメーターにLPA=YESが指定されている場合、 区分セットはLPAから使用できます。LPA から区分セットを使用するには、それがそこにインストール済みで、区分セットが PRVMOD 開始

オプションで指定されていないことが必要です。詳しくは、MVS リンク・パック域からのモジュール使用の制御 を参照してください。

PARTNER リソース

PARTNER リソースを使用することにより、 CICS アプリケーション・プログラムは、APPC プロトコルを使用して、リモート論理装置上で稼働しているパートナー・アプリケーション・プログラムと通信することができます。

CICS アプリケーション・プログラムとパートナー・アプリケーション・プログラムの間の対話は、会話 と呼ばれます。

PARTNER 定義も、システム・アプリケーション体系 (SAA) の通信要素とのインターフェースに対する呼び出しの使用を容易にしています。 SAA 通信インターフェースについて詳しくは、共通プログラミング・インターフェース・コミュニケーション (CPI-C) 解説書を参照してください。

SAA 通信インターフェースを使用できるようにするには、以下のリソースを指定する必要があります。

- PROFILE 定義 (137 ページの『PROFILE リソース』を参照)
- CONNECTION 定義 (6 ページの『CONNECTION リソース』を参照)
- SESSIONS 定義 (159 ページの『SESSIONS リソース』を参照)

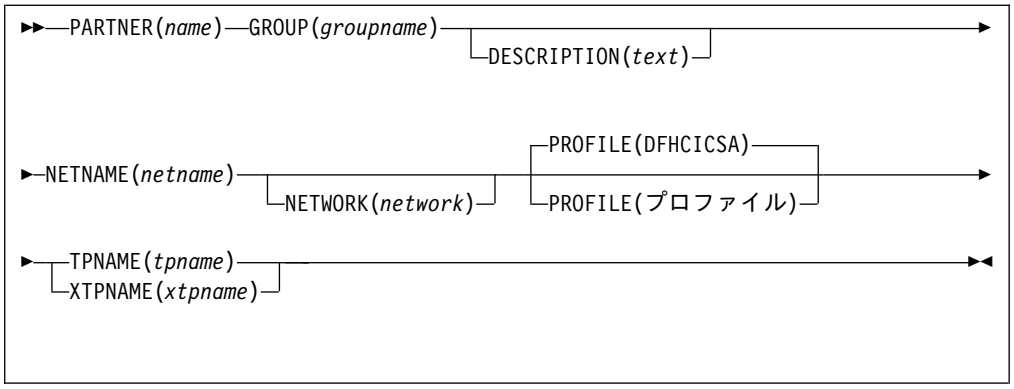
以下の 2 つの方法のどちらかで、CICS パートナー情報を定義することができます。

- PARTNER 定義を作成する
- アプリケーション・プログラムで、SYMDESTNAME をヌル値に設定して適切な CPI SET 呼び出しを発行する。詳細については、「共通プログラミング・インターフェース・コミュニケーション (CPI-C) 解説書」を参照してください。

CEMT および CEDA を使用した PARTNER リソース定義のインストールについては、PARTNER リソース定義のインストールを参照してください。BAS の作業については、BAS パートナー・リソース定義の作業を参照してください。

PARTNER 属性

PARTNER リソースの構文および属性について説明します。



DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

NETNAME(*netname*)
パートナー・アプリケーション・プログラムが実行されている論理装置のネット名。これは、CONNECTION 定義で指定された NETNAME 属性と一致します。名前の最大長は 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

NETWORK(*network*)
パートナーLUが配置されている ネットワークの名前を指定するには、 このオプション属性を使用します。名前の最大長は 8 文字です。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

PARTNER(*name*)

この PARTNER 定義の名前を指定します。名前の最大長は 8 文字です。

許容文字:

A-Z 0-9

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

文字 DFH は、CICS が使用するために予約されているため、DFH で始まるパートナー名は使用しないことが推奨されます。

CICS は、それらを削除または破棄したり、新たに機能や修正をアップグレードあるいは提供する際に CICS リソースに置き換えたりすることができます。

パートナー定義は、パートナー・プログラムとの会話を確立するために必要な SAA 通信インターフェースの情報を指定します。詳しくは、共通プログラミング・インターフェース・コミュニケーション (CPI-C) 解説書 を参照してください。

PROFILE(プロファイル)

セッションおよび会話に使用される PROFILE 定義の名前を指定します。デフォルトの PROFILE は DFHCICSA です。名前の長さは、最大で 8 文字です。

許容文字:

A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

TPNAME(*tpname*)

パートナー LU で実行されるリモート・トランザクション・プログラムの名前を指定します。リモート TP 名の定義は必須です。TPNAME またはその代わりとなる XTPNAME のいずれかを指定する必要があります。この名前の長さは最大 64 文字です。

許容文字:

A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

この範囲の文字では指定したい名前に不十分である場合は、TPNAME の代わりに XTPNAME 属性を使用することができます。

XTPNAME(*xtpname*)

この属性は、TPNAME の代わりとして使用できます。リモート TP 名の定義は必須であるため、これら 2 つの名前のいずれかを指定する必要があります。

128文字までの16進ストリングを入力します。パートナー LU上で実行されているリモート・トランザクション・ プログラムの名前を表します。**X'40'** 以外のすべての 16 進数の組み合わせが可能です。

72 文字より長い XTPNAME を DFHCSDUP に指定するには、列 72 にアスタリスクを指定します。これにより、 後続の行が連結されます。

PIPELINE リソース

PIPELINE リソースは、CICS アプリケーションが Web サービス・プロバイダーまたはリクエスターの役割を果たすときに使用されます。これは、サービス要求および応答に対して作動するメッセージ・ハンドラー・プログラムに関する情報を提供します。普通、単一の PIPELINE 定義により多くのアプリケーションが使用できるインフラストラクチャーが定義されます。

処理ノードに関する情報は、間接的に提供されます。PIPELINE は、ノードおよびその構成の XML 記述が含まれる z/OS UNIX ファイルの名前を指定します。

インバウンド Web サービス (つまり、クライアントが CICS 内で Web サービスを呼び出す要求) は、URIMAP リソースによって PIPELINE リソースに関連付けられます。 URIMAPは、 要求に関連付けられたURIに適用するPIPELINE資源を示します。 PIPELINEは、メッセージに対して実行される処理を指定します。

CICS Web サービス・アシスタントを使用してデプロイされたサービス・プロバイダーの場合、ピックアップ・ディレクトリーのスキャン時に URIMAP リソースが自動的に作成されます。このスキャンは、PIPELINE リソースがインストールされたとき、あるいは **PERFORM PIPELINE SCAN** コマンドの結果として行われます。WEBSERVICE リソースを特定の URI に関連付けるための情報を CICS に提供する URIMAP リソースは、必須リソースです。このリソースの属性は、ピックアップ・ディレクトリー内の Web サービス・バインディング・ファイルによって指定されます。 WSDL アーカイブ・ファイルまたは WSDL 文書を特定の URI に関連付けるための情報を CICS に提供する URIMAP リソースは、オプション・リソースであり、ピックアップ・ディレクトリーに WSDL ファイルまたは WSDL アーカイブ・ファイルが存在する場合に作成されます。

サービス・リクエスターに関しては、PIPELINE リソースがインストールされたとき、あるいは **PERFORM PIPELINE SCAN** コマンドの結果として、CICS が URIMAP リソースを自動的に作成することはありません。

BAS の作業については、BAS パイプライン・リソース定義の作業を参照してください。

CICS バンドル内の PIPELINE リソース

CICS バンドルを使用して、PIPELINE リソース定義を作成、編集、およびインストールすることができます。この方法で PIPELINE リソースを作成した場合、CICS バンドルを使用してそのリソースのライフサイクルを管理する必要があり、リソースを単独で管理することはできません。

EXEC CICS INQUIRE PIPELINE コマンドまたは **CEMT INQUIRE PIPELINE** コマンドを使用して、CICS バンドルによって動的に生成される PIPELINE リソースを照会することができます。ただし、動的に生成された PIPELINE リソースに対して **DISCARD** コマンドを発行したり、**SET** コマンドを使用してリソースの状況を変更し

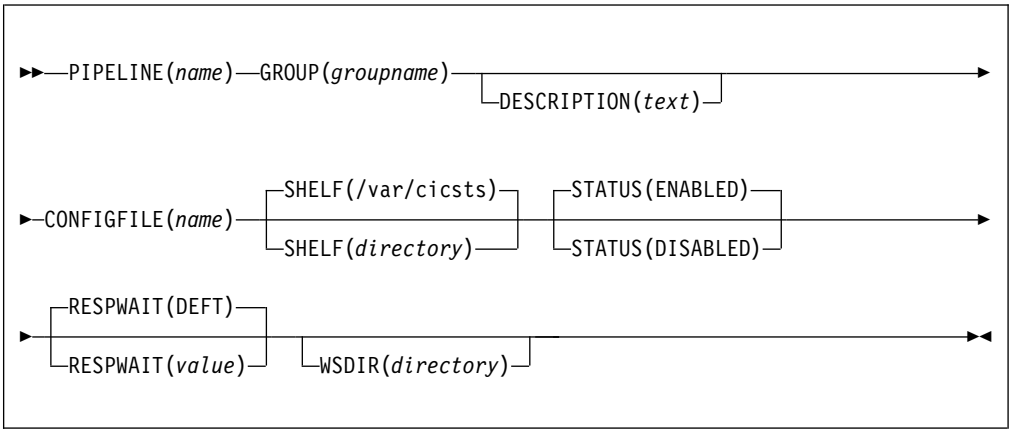
たりすることはできません。これらのコマンドは、BUNDLE リソースに対して発行する必要があり、CICS がそれらを PIPELINE リソースに適用します。

動的に生成された PIPELINE リソースの属性は変更できますが、変更内容はカタログされず、CICS のウォーム・リスタートの後に回復されません。バンドルによってインストールされた PIPELINE リソースの属性を変更したい場合は、CICS バンドルを無効にして破棄し、必要な変更を加えた新規バージョンのバンドルをインストールします。

CICS バンドル内のリソースの定義について詳しくは、Defining CICS bundles を参照してください。

PIPELINE の属性

PIPELINE リソースの構文と属性を説明します。



CONFIGFILE(*name*)

サービス要求および応答に対して作用する処理ノードに関する情報が格納されている z/OS UNIX ファイルの完全修飾名または相対名を指定します。

- オンライン・リソース定義で定義された PIPELINE リソースの構成ファイルでは、ファイル・パスは、先頭にスラッシュ (/) がある場合は完全修飾パスであり、それ以外の場合は CICS 領域ユーザー ID の HOME ディレクトリーからの相対パスです。
- CICS バンドルで定義された PIPELINE リソースの構成ファイルの場合、ファイル・パスは、CICS バンドルのルート・ディレクトリーからの相対パスです。 zFS ファイルが、PIPELINE リソースと一緒に CICS バンドルにパッケージ化されている必要があります。詳しくは、Referencing zFS artifacts in a bundleを参照してください。

指定される値は UNIX ファイルに有効な名前でなければなりません。
- 組み込みスペース文字を含めることはできません。
- / 文字の連続インスタンスが含まれていてはなりません。
- 大/小文字の区別があります。

許容文字:

```
A-Z a-z 0-9 . / _ # @ -
```

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

PIPELINE(*name*)
この PIPELINE の名前を指定します。この名前は、8 文字以内の長さにすることができます。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

RESPWAIT(*value*)
アプリケーション・プログラムがリモート Web サービスからの応答メッセージを待機する秒数を指定します。この値は 0 から 9999 秒の範囲で指定することができます。

トランスポート・プロトコルのデフォルトのタイムアウト値を使用するには、DEFT を指定します。RESPWAIT に値を指定しない場合も、デフォルトのタイムアウト値が使用されます。

- HTTP のデフォルトのタイムアウト値は 10 秒です。
- WebSphere MQ のデフォルトのタイムアウト値は 60 秒です。

SHELF({**/var/cicsts/**|*directory*})
z/OS UNIX 上のディレクトリー (主に Web サービス・バインディング・ファイル用のシェルフ) の完全修飾名を 1 から 255 文字で指定します。

指定される値は UNIX ファイルに有効な名前でなければなりません。 • 組み込みスペース文字を含めることはできません。 • / 文字の連続インスタンスが含まれていてはなりません。 • 大/小文字の区別があります。 許容文字: A-Z a-z 0-9 . / _ # @ -

PIPELINE 定義のインストール先の CICS 領域は、シェルフ・ディレクトリーに対する完全アクセス権 (読み取り、書き込み、およびサブディレクトリーの作成ができること) を持っている必要があります。

単一のシェルフ・ディレクトリーを、複数の CICS 領域および複数の PIPELINE 定義で共用できます。シェルフ・ディレクトリー内では、それぞれの CICS 領域が個別のサブディレクトリーを使用することで、自己のファイルを他の CICS 領域のファイルと分離して保持します。各領域のディレクトリー内では、それぞれの PIPELINE は個別のサブディレクトリーを使用します。

インストール済み PIPELINE 定義によって参照されているシェルフの内容を変更しないでください。変更した場合の影響は予測不能です。

CICS Explorer を使用して CICSバンドル内に定義した PIPELINE リソースは、シェルフ・ディレクトリーを使用しません。

STATUS({**ENABLED**|**DISABLED**})
PIPELINE がインストールされるときの初期状況を指定します。

ENABLED
この PIPELINE の Web サービス要求は通常どおりに処理されます。

DISABLED
この PIPELINE の Web サービス要求は処理できません。

WSDIR(*directory*)
z/OS UNIX 上の *Web* サービス・バインディング・ディレクトリー (ピックアップ・ディレクトリーとも呼ばれる) の完全修飾名を 1 から 255 文字の範囲で指定します。 CICS 領域にインストールされている PIPELINE は、それぞれ別の Web サービス・バインディング・ディレクトリーを指定している必要があります。

指定される値は UNIX ファイルに有効な名前でなければなりません。 • 組み込みスペース文字を含めることはできません。 • / 文字の連続インスタンスが含まれていてはなりません。 • 大/小文字の区別があります。 許容文字: A-Z a-z 0-9 . / _ # @ -

Web サービス・バインディング・ディレクトリーには、PIPELINE に関係付けられた Web サービス・バインディング・ファイルが格納されており、これらのファイルは CICS スキャン・メカニズムによって自動的にインストールされます。 PIPELINE 定義のインストール時に、CICS はディレクトリーをスキャンして、見つかった web サービス・バインディング・ファイルを自動的にインストールします。これは、PIPELINE が使用可能状態でインストールされるか、使用不可状態でインストールされるかに関係なく行われることに注意してください。

WSDIR 属性の値を指定する場合、この値は、CICS 領域が少なくとも読み取りアクセスを持っている有効な z/OS UNIX ディレクトリーを参照している必要があります。そうでない場合、PIPELINE リソースをインストールする試みは失敗します。

WSDIR に値が指定されない場合、PIPELINE のインストール時に自動スキャンは行われません。

PROCESSTYPE リソース

PROCESSTYPE リソースは BTS プロセス・タイプを定義します。物理 VSAM データ・セット (リポジトリー) に関連する CICS ファイルに名前を付けます。このデータ・セットには、このタイプ (およびそのアクティビティー・インスタンス) の全ての処理の詳細情報が保管されます。

CICS ビジネス・トランザクション・サービス (BTS) API を使用して、プロセスと呼ばれる複雑なビジネス・アプリケーションを定義および実行することができます。メモリー内では、プロセスは、その実行に関連する情報を含むストレージのブロックを表します。また、少なくとも 1 つの追加の情報ブロック (アクティビティー・インスタンスと呼ばれます) に関連付けられています。 CICS ビジネス・トランザクション・サービス・ドメインの制御下で実行されていない場合、プロセスおよびそのアクティビティー・インスタンスは、リポジトリー と呼ばれるデータ・セットに書き込まれます。

異なるプロセス・タイプを割り当てて、BTS プロセスをカテゴリー化することができます。このカテゴリー化は、例えばブラウズを行う場合などに便利です。プロセスを構成するアクティビティーのプロセス・タイプは、プロセス自体のプロセス・タイプと同じです。

注: 複数のプロセス・タイプのレコードを同じリポジトリーの データ・セットに書き込むことができます。

135 ページの図 2 は、PROCESSTYPE 定義、FILE 定義、および BTS リポジトリー・データ・セットの間の関係を示しています。

- 複数のPROCESSTYPE資源が同じFILEを参照できる。
- 複数のFILEが同じBTSリポジトリー・データ・セットを参照できる。

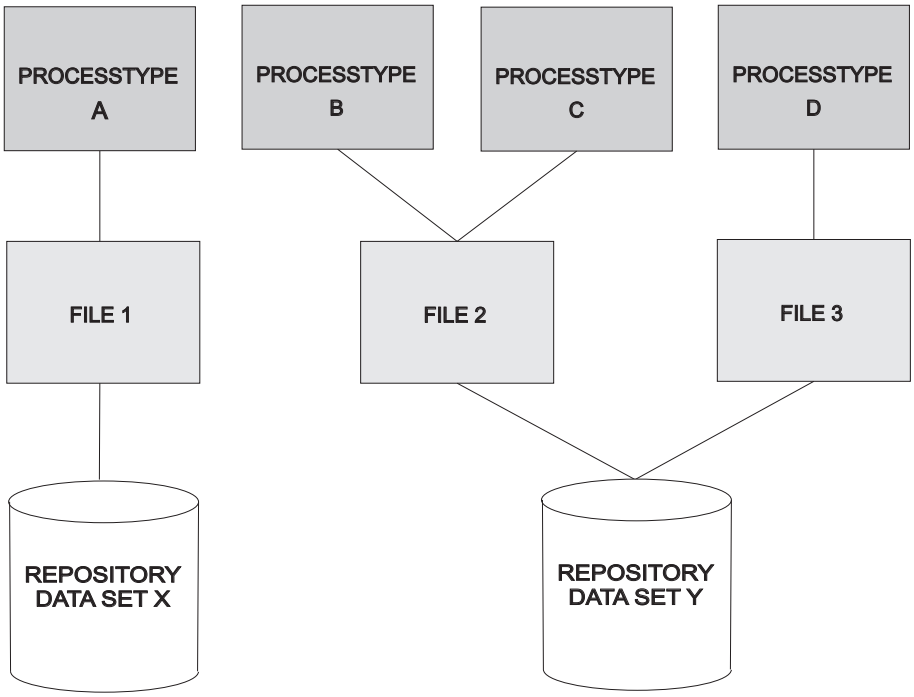


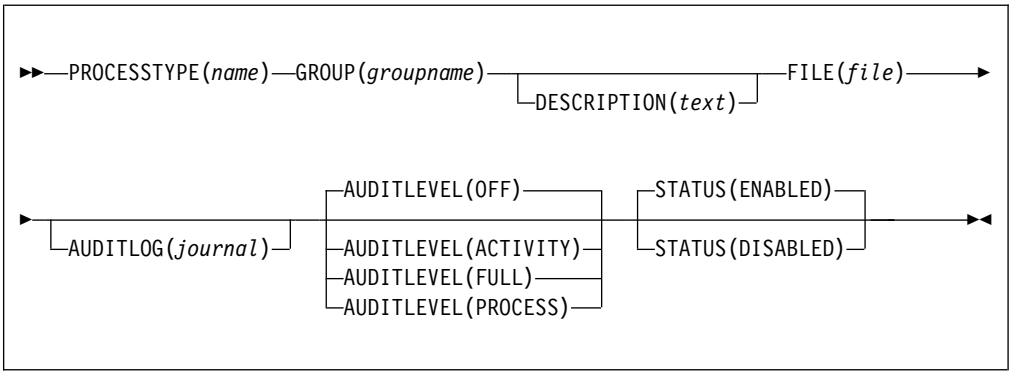
図 2. *PROCESSTYPE* 定義、*FILE* 定義、および *BTS* リポジトリー・データ・セット

監査の目的でBTSのプロセスやアクティビティーの進行を記録し、 BTSアプリケーションにおける診断エラーに役立てることができます。その場合、指定されたタイプのプロセスについて、監査レコードを書き込む CICS ジャーナルと、必要な監査レベルを指定することができます。

BAS の作業については、BAS PROCESSTYPE リソース定義の作業を参照してください。

PROCESSTYPE 属性

PROCESSTYPE リソースの構文および属性について説明します。



AUDITLEVEL({**OFF**|**PROCESS**|**ACTIVITY**|**FULL**})

このタイプのプロセスの監査ログの初期レベルを指定します。 OFF 以外の値を指定する場合、AUDITLOG オプションも指定する必要があります。

ACTIVITY
アクティビティー・レベルの監査。監査レコードの書き込み元は以下のとおりです。
1. プロセス監査ポイント
2. アクティビティー 1 次監査ポイント

FULL 全監査。監査レコードの書き込み元は以下のとおりです。
1. プロセス監査ポイント
2. アクティビティー 1 次監査ポイントおよび 2 次監査ポイント

OFF 監査証跡レコードは書き込まれません。

PROCESS
プロセス・レベルの監査。プロセス監査ポイントからの 監査レコードのみが書き込まれます。

プロセス、アクティビティー 1 次監査ポイント、およびアクティビティー 2 次監査ポイントから書き込まれるレコードの詳細については、監査ロギングのレベルの指定 を参照してください。

AUDITLOG(*journal*)
このタイプのプロセスおよびそれを構成するアクティビティーについて、監査証跡レコードが書き込まれる CICS ジャーナルの名前を指定します。この名前の長さは、最大 8 文字です。監査ログを指定しないと、 このタイプのプロセスの監査レコードは保持されません。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

FILE(*file*)
このプロセス・タイプのプロセス・レコードおよびアクティビティー・レコードを関連リポジトリー・データ・セットに書き込むために使用する、CICS ファイル定義の名前を指定します。この名前の長さは、最大 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

PROCESSTYPE(*name*)
この PROCESSTYPE 定義の名前を指定します。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

名前の最大長は 8 文字です。ブランク文字を名前の先頭に使用したり、名前に組み込んだりすることはできません。指定された名前が 8 文字よりも短い場合は、8 文字になるまで末尾にブランクが埋め込まれます。

STATUS({**ENABLED**|**DISABLED**})
START=COLD または START=INITIAL で、CICS の初期設定に続くプロセス・タイプの初期状況を指定します。初期設定後、CEMT SET PROCESSTYPE コマンドを使用して、プロセス・タイプの状況を変更できます。再始動後のプロセス・タイプの状況は、直前のシャットダウン時の状況にリカバリーされます。

DISABLED
このタイプのプロセスは作成できません。このタイプのプロセスを作成する EXEC CICS DEFINE PROCESS 要求を行うと、INVREQ 条件がアプリケーション・プログラムに返されます。

ENABLED
このタイプのプロセスは作成できます。

PROFILE リソース

PROFILE リソースは、トランザクションとターミナル、またはトランザクションと論理装置間の相互作用を制御するオプションを指定します。PROFILE は、画面サイズやプリンター互換性などのオプションの使用、およびメッセージ・ジャーナリングやノード・エラー・プログラムなどの機能の使用を標準化する手段となっています。

MODENAME
プロファイルは、トランザクションと別のシステムへの LUTYPE6.1 セッションあるいは APPC セッションの間の通信に関連付けられます。 APPC セッションの場合、PROFILE 定義では、SESSIONS 定義でも指定されている MODENAME を参照します。この MODENAME は、セッションが属するモードセットの名前です。 CONNECTION リソースを参照してください。

CICS にインストールされた時点で、PROFILE 定義からの情報により、プロファイル・テーブルの中に 1 つの項目が作成されます。この項目は、後でその PROFILE を参照する各トランザクションにより使用されます。

さまざまな目的に合った PROFILE 定義が CICS から提供されています。使用する PROFILE は、各 TRANSACTION 定義で指定されます。 PROFILE を指定しない場合、トランザクションでは、標準的な方法で端末を使用するために提供されている PROFILE が使用されます。

CICS 相互通信機能 (機能シップなど) を使用する場合、トランザクションとセッションの間の通信には PROFILE が必要です。 CICS 提供プロファイルの属性を、グ

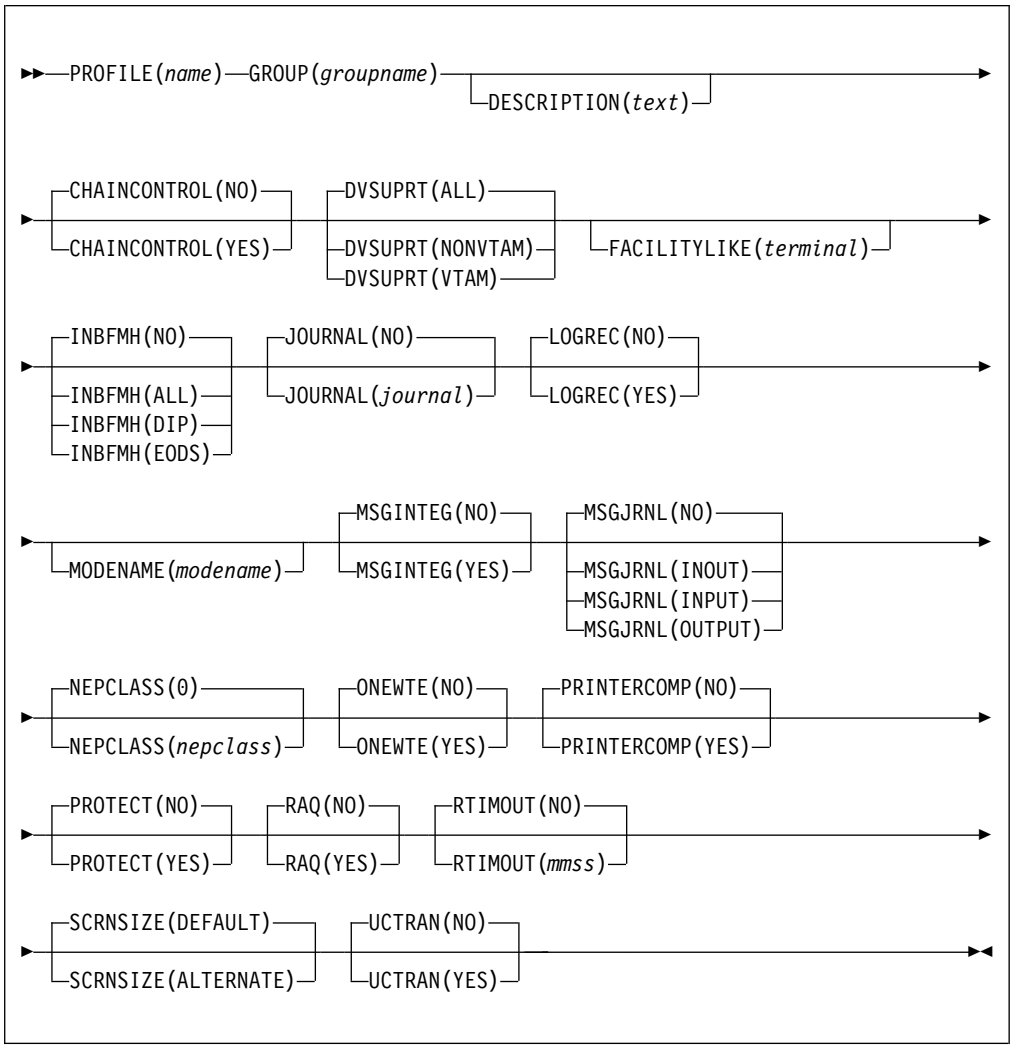
ループ DFHISC 内の PROFILE 定義に示します。 通信プロファイルの定義 では、CICS 提供の PROFILE の詳細、および独自のプロファイルの定義について説明しています。

BAS の作業については、BAS プロファイル・リソース定義の作業を参照してください。

PROFILE 属性

PROFILE リソースの構文および属性について説明します。

注: VTAM は、現在は z/OS Communications Server と呼ばれています。



属性

CHAINCONTROL({NO|YES})

アプリケーション・プログラムで要求単位のアウトバウンド・チェーンを制御できるかどうかを指定します。 CHAINCONTROL(YES) を指定する場合、ONEWTE(YES) は、1 つの端末制御出力要求ではなく、1 つのチェーンを意味します。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

DVSUPRT({**ALL**|**NONVTAM**|**VTAM**})
サポートするデバイス (端末または論理装置) を指定します。特定の端末または論理装置が使用するアクセス方式は、その関連する TCTTE で指定されます。

ALL このプロファイルは、どの端末や論理装置でも使用することができます。

NONVTAM
プロファイルは、z/OS Communications Server 以外の端末でのみ使用することができます。

VTAM
このプロファイルは論理装置でのみ使用できます。

FACILITYLIKE(*terminal*)
これはオプションの属性です。ブリッジ機能のテンプレートとして使用する、既存 (4 文字) の端末リソース定義の名前を指定します。ブリッジ出口ルーチンに FACILITYLIKE を指定することで指定変更できます。

この属性にはデフォルト値がありません。

VTAM=NO システム初期設定テーブル (SIT) パラメーターを使用して始動した CICS システムで実行する場合は、 FACILITYLIKE によって指定するリソース定義をリモート端末として定義する必要があります。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

INBFMH({**NO**|**ALL**|**DIP**|**EODS**}) **(SNA LU のみ)**
論理装置で使用されるプロファイルにおいて、論理装置から受け取る機能管理ヘッダー (FMH) がアプリケーション・プログラムに渡されるかどうかを指定します。

ALL すべての FMH (APPC FMH および LU6.1 ATTACH、CICS によって処理される SYNCPOINT FMH を除く) が、アプリケーション・プログラムに渡されます。この値は、 CSMI などの機能シップ・トランザクションや分散トランザクション処理を使用するトランザクション、分散プログラム・リンク要求には必須です。

DIP バッチ・データ交換プログラム (DFHDIP) が、インバウンド FMH を処理します。 BMS 受信要求が発行された場合、BMS はバッチ・データ交換の受信要求を発行し、バッチ・データ交換の受信要求が端末制御受信要求の代わりに発行されます。

EODS
FMH は、データ・セットの終わり (EODS) を示している場合のみ、アプリケーション・プログラムに渡されます。

NO FMH は廃棄されます。

JOURNAL({**NO**|*journal*})
ジャーナル ID を設定することにより、メッセージの自動ジャーナル処理を行わせることを指定します。

NO メッセージの自動ジャーナル処理は行われません。

journal
自動ジャーナル処理に使用されるジャーナル ID。これは 01 から 99 の範囲の任意の数です。 この数は文字 DFHJ に追加されて DFHJ*nn* という形式のジャーナル ID となり、これが MVS システム・ロガー汎用ログ・ストリームにマップされます。

注: CICS Transaction Server for z/OS では、DFHJ01 はシステム・ログではありません。
トランザクション・ルーティング環境では、メッセージのジャーナル処理はアプリケーション所有領域 (AOR) で実行されます。したがって、JOURNAL 属性は AOR のトランザクション・プロファイルに指定する必要があります。

LOGREC({**NO**|**YES**})
アプリケーションの設計において、各 EXEC CICS RECEIVE 要求が論理レコードにより満たされる必要があるかどうかを指定します。このオプションにより、既存の 2770 および 2780 ベースのアプリケーション・プログラムは、バッチの論理装置 (例えば、3790、8100) にプログラムを変更せずに接続することができます。

MODENAME(*modename*)
APPC 接続で使用するセッション・グループの識別名を指定します。この名前は、最大 8 文字の長さで指定することができ、z/OS Communications Server に定義された z/OS Communications Server LOGMODE エントリーの名前でなければなりません。予約名 SNASVCMG を使用することはできません。モード名を省略した場合、デフォルトはブランクになります。 z/OS Communications Server のモード名について詳しくは、APPC セッション・グループの定義 を参照してください。

このプロファイルを指定するトランザクションが EXEC CICS START コマンドを使用して開始されている場合、MODENAME は基本機能の割り振りに使用されます。トランザクションが EXEC CICS ALLOCATE コマンドを実行してこのプロファイルを指定した場合、MODENAME は代替機能の割り振りに使用されます。

MODENAME を指定しない場合、CICS は、定義済みのモードセットのいずれか 1 つからセッションを選択します。

EXEC CICS ALLOCATE コマンドで PROFILE を指定しない場合、CICS 提供のプロファイル DFHCICSA が使用されます。機能シップの場合は、常にプロファイル DFHCICSF が使用されます。 MODENAME は、これらのプロファイルのいずれの定義にも指定されませんが、独自のコピーを作成する場合には、MODENAME を追加することができます。次に、APPC を使用して通信するすべてのシステムについて、MODENAME を使用するモードセットを、TERMINAL または SESSIONS 定義に定義済みにしておく必要があります。

MODENAMEが指定されていてそれを 削除する場合、EOF消去キーを 押して、前に指定した値を完全に 削除してください。

MSGINTEG({NO|YES}) (SNA LU のみ)
確定応答が論理装置への出力要求で要求されるかどうかを指定します。パイプライン・トランザクションに YES を指定することはできません。

MSGJRNL({NO|INPUT|OUTPUT|INOUT})
ジャーナルに自動的に記録するメッセージを指定します。NO 以外の値を指定する場合は、JOURNAL 属性の値も指定する必要があります。

NO メッセージのジャーナル処理は必要ありません。

INPUT
ジャーナル処理は入力メッセージで必要です。

OUTPUT
ジャーナル処理は出力メッセージで実行されます。

INOUT
ジャーナル処理は入力および出力メッセージで実行されます。

トランザクション・ルーティング環境では、メッセージのジャーナル処理は、ルーティングされた APPC (LU タイプ 6.2) セッションのアプリケーション所有領域 (AOR) で実行されます。したがって、AOR のトランザクション・プロファイルに MSGJRNL 属性を指定する必要があります。ルーティングされたその他のセッションについては、メッセージのジャーナル処理は端末専有領域 (TOR) で実行されます。この場合、TOR のトランザクション・プロファイルに MSGJRNL 属性を指定する必要があります。

NEPCLASS({0|*nepclass*}) (z/OS Communications Server のみ)
ノード・エラー・プログラム・トランザクション・クラスを指定します。この値によって、TYPETERM 定義と SESSION 定義に指定された値がオーバーライドされます。

0 z/OS Communications Server 装置の場合、これは、デフォルト・ノード・エラー・プログラム・モジュールへのリンクとなります。あるいは、z/OS Communications Server 以外の装置の場合は、これがデフォルト値になります。

value (デフォルトではない) ノード・エラー・プログラム・モジュールのトランザクション・クラス。この値は 1 から 255 の範囲です。ノード・エラー・プログラムのプログラミング情報については、ノード・エラー・プログラムの作成 を参照してください。

NEPCLASS 属性は、ユーザー・トランザクションのみに適用され、SNASVCMGR セッションでは無視されます。

ONEWTE({**NO**|**YES**})
トランザクションの実行中に、書き込み操作または EXEC CICS SEND が 1 回だけ許可されるかを指定します。 YES を指定すると、トランザクションの最初の書き込み時に LAST オプションが強制されます。すべての追加書き込み要求はエラーとして処理され、タスクは異常終了するように準備されます。

PIPELINE トランザクションの場合は、YES を指定する必要があります。

PRINTERCOMP({**NO**|**YES**})
データ・ストリームの生成で BMS SEND TEXT コマンドのプリンター互換性オプションをサポートするために必要な互換性レベルを指定します。

NO 出力の各行はブランク文字で始まり、その形式は、属性バイトが各行の前に置かれる 3270 ディスプレイの形式と同等となります。

YES ブランク文字を挿入しないようにすることで、データの先頭文字として組み込まれる用紙送り文字を受け入れ、プリンターの幅全体をデータに使用できるようにします。

BMS 用紙送りオプションを使用する場合には、YES を選択します。

PROFILE(*name*)
この PROFILE 定義の名前を指定します。名前の最大長は 8 文字です。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

文字 DFH は、CICS が使用するために予約されているため、DFH で始まる名前は使用しないことが推奨されます。

CICS は、それらを削除または破棄したり、新たに機能や修正をアップグレードあるいは提供する際に CICS リソースに置き換えたりすることができます。

注: 名前にコンマ (,) を使用すると、以下のようなコマンドを使用できなくなります。

```
CEMT INQUIRE PROFILE(value1,value2)
```

これらのコマンドでは、コンマがリスト区切り文字として機能します。リソース ID のリストの使用については、リソース ID のリスト を参照してください。

プロファイルは、CICS と端末あるいは論理装置の間の対話を制御するオプションを指定します。トランザクション定義でプロファイル名を指定することで、トランザクションとその基本端末の間の通信を制御するオプション・セットを指示します。 EXEC CICS ALLOCATE コマンドでプロファイル名を指定することで、トランザクションと割り振られたセッションの間の通信を制御するオプションを指示することもできます。

CICS は、ほとんどの目的に利用できる多くのプロファイル定義を提供します。詳細については、通信プロファイルの定義を参照してください。

PROTECT
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。詳しくは、Obsolete attributesを参照してください。

RAQ({**NO**|**YES**}) **(SNA 端末のみ)**
「先読みキュー」オプションが必要かどうかを指定します。

NO トランザクションは SNA プロトコルに従い、正しいモードの場合にのみ SEND および RECEIVE を行います。このプロトコルに従わない場合、コード ATCV で異常終了する場合があります。

YES トランザクションは SNA プロトコルに従わない場合があり、CICS は、着信データがトランザクションによって具体的に要求されるまで、そのデータを一時記憶域にキューとして入れます。 RAQ(YES) は 2 進同期デバイスと論理装置の両方をサポートするトランザクションとの互換性のみのために備えられており、それを使用することは推奨されていません。

RTIMOUT({**NO**|*mmss*})

読み取りタイムアウト機能のタイムアウト値を指定します。EXEC CICS WEB API を使用して HTTP クライアント要求を作成するタスクの場合、この期間が終了すると、CICS が TIMEDOUT 応答をアプリケーションに返します。その他のタスクの場合は、失敗したコマンドは TERMERR を返し、タイムアウトになったタスクは、AKCT、AZCT、または AZIG 異常終了を受け取ります。

NO 読み取りタイムアウト機能は不要です。

value これは、端末からの入力を受信していない場合にタスクを終了するようにするまでの間隔 (MMSS は分および秒) です。指定できる最大値は 70 分です。値が指定されており、デフォルトを NO にしたい場合は、以前に指定された値を完全に削除する必要があります。

SCRNSIZE({**DEFAULT**|**ALTERNATE**})

3270 表示装置または印刷装置の DEFAULT または ALTERNATE バッファー・サイズを使用するかどうかを指定します。画面サイズとバッファー・サイズの選択について詳しくは、TYPETERM 定義の ALTSCREEN 属性および DEFSCREEN 属性を参照してください。

TYPETERM 定義に ALTSCREEN(0,0) および DEFSCREEN(0,0) が含まれる場合、SCRNSIZE 値は無視されます。つまり、画面サイズは TYPETERM 定義内の関連した TERMMODEL 属性から引き継がれます。ページ・サイズは PAGESIZE から取得され、ALTPAGE 値は無視されます。出力要求のために 3270 削除書き込み (EW) コマンドに ERASE オプションを付けて挿入されます。

ALTERNATE

TYPETERM 定義に非ゼロの ALTSCREEN がある場合は、代替画面サイズ・モードが適用され、削除書き込み代替 (EWA) コマンドが使用されます。つまり、ERASE オプション付きの端末出力要求が発行されるたびに、3270 EWA コマンドがデータ・ストリームに挿入されます。ALTSCREEN 値は画面サイズとして想定され、BMS はページ・サイズとして ALTPAGE 内の値を使用します。

SCRNSIZE(ALTERNATE) は、すべての CICS サービス・トランザクション (CSMT など) に使用することができます。

DEFAULT

TYPETERM 定義に非ゼロの ALTSCREEN または非ゼロの DEFSCREEN がある場合は、デフォルト画面サイズ・モードが適用され、削除書き込み (EW) コマンドが使用されます。つまり、端末が ERASE オプション付きの端末出力要求を発行するたびに、3270 EW コ

マンドがデータ・ストリームに挿入されます。 DEFSCREEN 属性に指定した画面サイズが想定され、BMS はページ・サイズとして PAGESIZE 属性に指定された値を使用します。

注: DEFAULT と ALTERNATE の両方は、 SEND MAP、SEND TEXT、および SEND CONTROL コマンドの DEFAULT オプションと ALTERNATE オプションで指定変更することができます。

UCTRAN({NO|YES}) (z/OS Communications Server のみ)
このプロファイルを使用するトランザクション用にプログラムに端末入力を引き渡す前に、それを大文字に変換するかどうかを指定します。

また、関連した TYPETERM 定義 (277 ページの『TYPETERM 属性』を参照) の端末レベルで大文字への変換を要求することもできますが、以下の点を認識しておく必要があります。

- TYPETERM UCTRAN(YES) 定義は、PROFILE UCTRAN(NO) 定義をオーバーライドします。したがって、TYPETERM UCTRAN(YES) を指定すると、PROFILE UCTRAN(NO) は無効になります。
- PROFILE UCTRAN(YES) 定義は、TYPETERM UCTRAN(NO) 定義をオーバーライドします。
- TYPETERM UCTRAN(TRANID) を指定すると、トランザクション ID が大文字に変換され、CICS がトランザクション定義を見つけることができるようになります。 アプリケーションが受信するその他すべての入力は、PROFILE UCTRAN に指定された内容に従って変換されます。
- プロファイル定義で UCTRAN(YES) を指定しても、EXEC CICS RECEIVE または CONVERSE が実行されない限り、入力データの変換は行われません。つまり、トランザクションが動的ルーティング・プログラム (例えば、DFHDYP) を介してルーティングされる場合、ルーティング・プログラムに渡される入力データのコピーは、PROFILE 定義の UCTRAN オプションの影響は受けません。

注: z/OS Communications Server 端末の AOR にリモート定義があり、AOR の UCTRAN 値が TOR と異なっているトランザクション・ルーティング環境では、UCTRANST の TOR 値 (EXEC CICS SET TERMINAL コマンドで指定) によって AOR の UCTRANST 値がオーバーライドされます。

表 2は、 PROFILE および TYPETERM リソース定義の UCTRAN 設定に従って、端末入力のどの部分が変換されるかを示しています (トランザクション ID および/またはデータ)。

表 2. *tranid* およびデータ変換での *UCTRAN* 属性の効果

PROFILE の **UCTRAN**	**TYPETERM** の **UCTRAN**	**TRANID** は変換されたか**?**	データは変換されたか**?**
YES	YES	はい	はい
YES	NO	いいえ	はい
YES	TRANID	はい	はい
NO	YES	はい	はい
NO	NO	いいえ	いいえ
NO	TRANID	はい	いいえ

一部の各国語文字は、UCTRAN(YES) が指定されていても自動的に変換されません。そのような場合は、国別文字の大文字への変換 に示されているいずれかの方法を使用することができます。

PROGRAM リソース

PROGRAM リソースは、プログラム・ライブラリーに保管され、トランザクションまたはトランザクションの一部を処理するのに使用されるプログラムの情報を制御します。

CEDA または DFHCSDUP を使用するか、または適切なシステム初期設定パラメーターを設定してプログラムの自動インストールを許可することで、PROGRAM 定義を作成できます。プログラムの自動インストールについては、『Configuring』の『Autoinstalling programs, map sets, and partition sets』 を参照してください。

BAS の作業については、BAS PROGRAM リソース定義の作業を参照してください。

CICS バンドル内の PROGRAM リソース

CICS バンドルを使用して、PROGRAM リソース定義を作成、編集、およびインストールすることができます。この方法で PROGRAM リソースを作成した場合、CICS バンドルを使用してそのリソースのライフサイクルを管理する必要があり、リソースを単独で管理することはできません。 **EXEC CICS INQUIRE PROGRAM** コマンドまたは **CEMT INQUIRE PROGRAM** コマンドを使用して、CICS バンドルによって動的に生成される PROGRAM リソースを照会することができます。ただし、CICS バンドルによって動的に生成された PROGRAM リソースに対して SET コマンドや DISCARD コマンドを発行することはできません。これらのコマンドは、BUNDLE リソースに対して発行する必要があり、CICS がそれらを PROGRAM リソースに適用します。

CICS バンドル内のリソースの定義について詳しくは、Defining CICS bundles を参照してください。

アプリケーション・エントリー・ポイントとしての PROGRAM リソース

PROGRAM リソースは、アプリケーション・エントリー・ポイントとして定義することができます。エントリー・ポイントの定義方法について詳しくは、CICS Explorer 製品資料内の『Defining application entry points』 を参照してください。

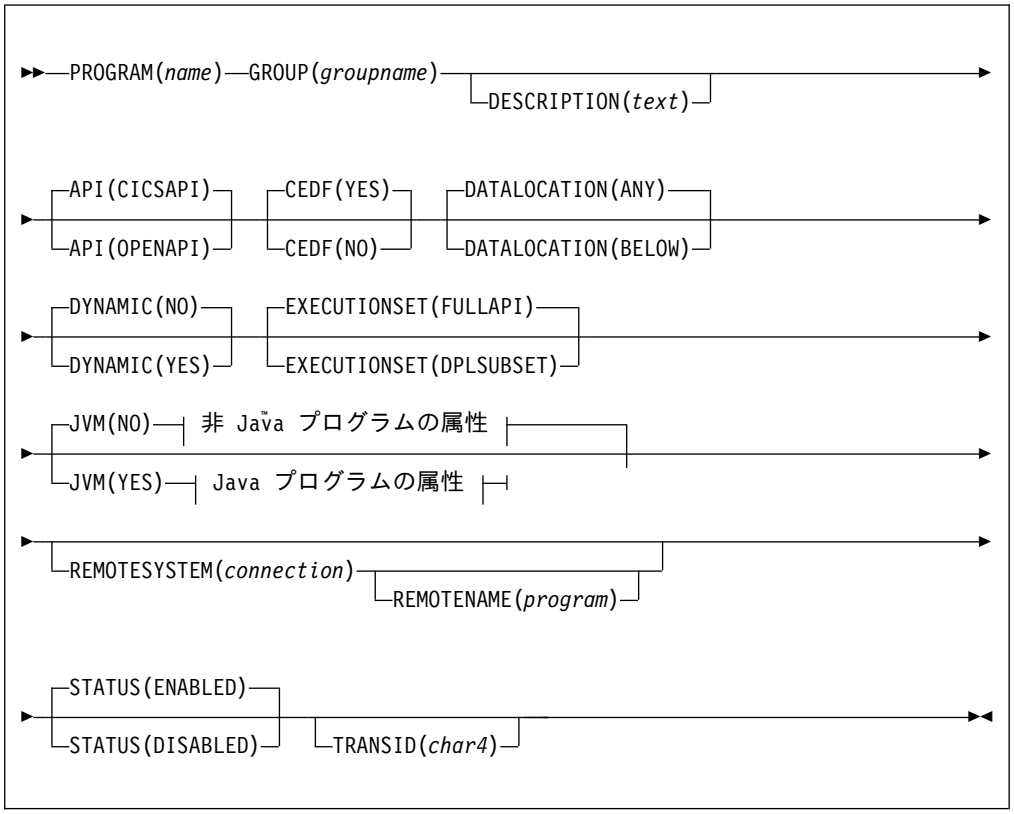
アプリケーションの専用 PROGRAM リソース

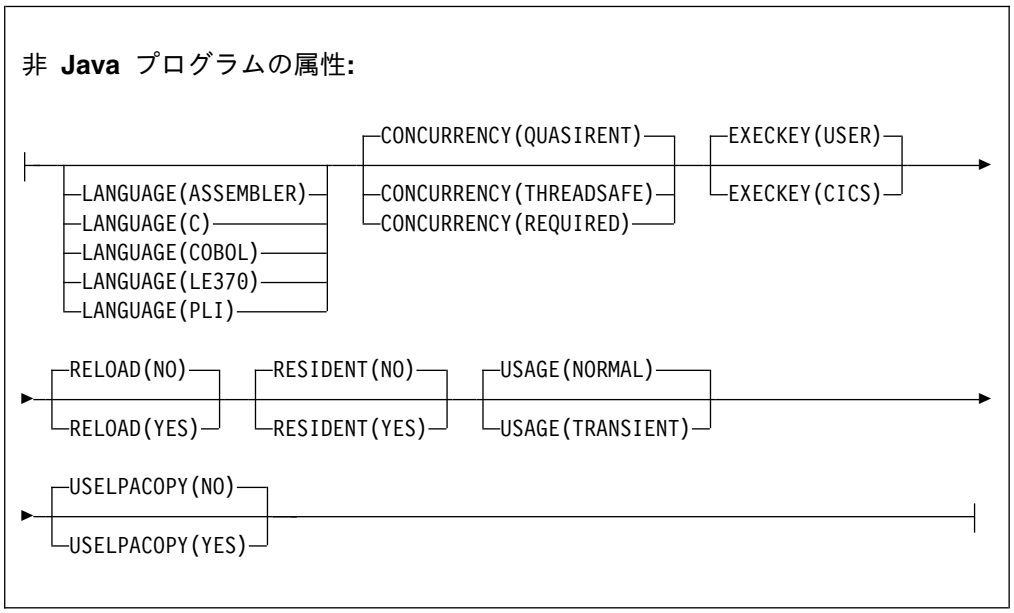
PROGRAM リソースは、アプリケーション・バージョンの専用リソースとしてサポートされます。プラットフォームにデプロイされるアプリケーションのためにタスクによって自動インストールされるプログラムは、そのバージョンのアプリケーションに専用のものでもあります。

専用リソースについて詳しくは、アプリケーション・バージョンの専用リソースを参照してください。

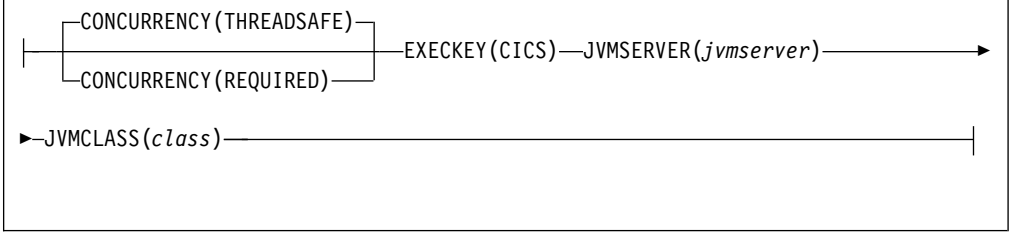
PROGRAM 属性

PROGRAM リソースの構文と属性について説明します。





Java プログラムの属性:



API({**CICSAPI**|**OPENAPI**})

プログラムで使用する API を指定します。API 属性は以下のプログラムに適用されます。

- ユーザー・アプリケーション・プログラム
- PLT プログラム
- ユーザー置換可能プログラム
- タスク関連のユーザー出口プログラム。詳しくは、Task-related user exit programsを参照してください。

API 属性は、グローバルなユーザー出口には適用されません。

CICSAPI

プログラムは CICS が許可したアプリケーション・プログラミング・インターフェースだけを使用するように制限されます。

プログラムを CONCURRENCY(QUASIRENT) で定義した場合、そのプログラムは常に準再入可能 (QR) TCB で実行されます。プログラムを CONCURRENCY(THREADSAFE) で定義した場合、適切と判断されたときに CICS で使用される TCB で、そのプログラムが実行されます。また、プログラムを CONCURRENCY(REQUIRED) で定義した場合は、そのプログラムは常にオープン TCB で実行されます。

OPENAPI

プログラムは、CICS アプリケーション・プログラミング・インターフェースに制限されません。

CICS は独自のオープン TCB 上でプログラムを実行します。使用されるオープン TCB のタイプは、EXECKEY 属性の値およびプログラム言語によって異なります。

コマンドを実行する際に CICS が QR TCB への切り替えを必要とする場合、制御をアプリケーション・プログラムに渡す前にオープン TCB に戻します。

OPENAPI 属性を指定するには、プログラムをスレッド・セーフ標準に適合するようにコーディングし、かつ CONCURRENCY(REQUIRED) で定義する必要があります。OPENAPI プログラムは、主にアプリケーション・ワークロードを QR TCB から複数のオープン TCB へ移動するために使用されます。サーバーのリソースを効率よく活用することができ、スループットを向上させることができます。

注: 以前のリリースでサポートされていた CONCURRENCY (THREADSAFE) と API(OPENAPI) の組み合わせは推奨されていませ

んが、互換性のために残されています。また、この組み合わせは CONCURRENCY(REQUIRED) と API(OPENAPI) の組み合わせと同じ動作をします。

OPENAPI プログラムでその他の (非 CICS) API を使用できます。オペレーティング・システムの待機によりオープン TCB がブロックされている場合、1 つのアプリケーションのみが影響を受け、CICS 全体が影響を受けることはありません。オペレーティング・システムの待機により TCB がブロックされる恐れがあり、この場合 CICS 全体が影響を受けるため、このような OPENAPI プログラムは QR TCB で実行することが許可されていません。それでも、OPENAPI プログラムには CICS システム全体としての役割があります。詳しくは、マルチスレッド化: 再入可能なプログラム、準再入可能なプログラム、およびスレッド・セーフ・プログラムを参照してください。

重要**:** CICS 内でその他の (非 CICS) API を使用する場合、リスクを踏まえたうえ、ユーザーの判断で使用してください。 CICS 内の他の (非 CICS) API のテストは実行されず、そのような API の使用は IBM サービスによってサポートされていません。

CEDF({**YES**|**NO**})

実行診断機能 (EDF) 制御のもとでプログラムを実行しているときの EDF アクションを指定します。

NO EDF 診断画面は表示されません。

YES EDF 診断画面が表示されます。NOEDF オプションを使用してプログラムが変換された場合は、プログラムの EDF 開始画面と EDF 終了画面のみが表示されます。表 3を参照してください。

表 *3. CEDF(NO)* および *NOEDF* が指定されたプログラムでの効果

PROGRAM での **CEDF** オプション	変換プログラムに指定した **EDF**	変換プログラムに指定した **NOEDF**
YES	全 EDF 画面	プログラム開始画面と終了画面のみ
NO	EDF 画面なし	EDF 画面なし
注: この表では、プログラム・リソース定義の CEDF オプションが、変換プログラムに指定された EDF オプションとどのように相互作用するかを示しています。		

CONCURRENCY({**QUASIRENT**|**THREADSAFE**|**REQUIRED**})

プログラムをスレッド・セーフ標準に従って作成するか、または準再入可能のみのプログラムとして作成するかを指定します。 CONCURRENCY 属性は、次のすべての CICS 実行可能プログラム・オブジェクトに対して指定できます。

- ユーザー・アプリケーション・プログラム
- PLT プログラム
- ユーザー置換可能プログラム
- グローバル・ユーザー出口プログラム。詳しくは、Global user exit programsを参照してください。
- タスク関連のユーザー出口プログラム。詳しくは、Task-related user exit programsを参照してください。

QUASIRENT

このプログラムは、準再入可能専用であり、共用リソースにアクセスするときには CICS により提供されるシリアライゼーションに依存します。

プログラムは、CICS で許可されているプログラミング・インターフェースに制限され、CICS の準再入可能規則に準拠している必要があります。詳しくは、マルチスレッド化: 再入可能なプログラム、準再入可能なプログラム、およびスレッド・セーフ・プログラムを参照してください。

この値は、すべての実行可能プログラムでサポートされています。

CICS により、プログラムは、JVM またはオープン API のタスク関連ユーザー出口を開始した後で制御が戻されたときでも、またはスレッド・セーフ・プログラムとやり取りしているときでも、常に QR TCB の下で実行されます。

THREADSAFE

プログラムをスレッド・セーフ標準に従って作成します。プログラムが共用リソースにアクセスするときに、他のプログラムが同時に実行されていて、同じリソースを変更しようとする可能性を考慮します。したがって、プログラムは、共用リソースにアクセスする際には、適切なシリアライゼーション技法を使用します。

JVM プログラム、および XPLink オプションでコンパイルされた C と C++ プログラムは、スレッド・セーフとして定義する必要があります。以前のリリースとの互換性が考慮されて、CONCURRENCY (THREADSAFE) が Java プログラムのデフォルト値ですが、CONCURRENCY(REQUIRED) を使用することが推奨されています。

CICS Db2 アプリケーション・プログラムについて詳しくは、CICS Db2 プログラムの実行および実動の準備を参照してください。

スレッド・セーフ・アプリケーション・プログラムの作成については、スレッド・セーフ・プログラムを参照してください。

この値は、すべての実行可能プログラムでサポートされています。スレッド・セーフなプログラムは、Language Environment に準拠したプログラムであるか、アセンブラー言語プログラムである必要があります。

REQUIRED

プログラムをスレッド・セーフ標準に従って作成します。CICS は、プログラムをオープン TCB で開始し、そのプログラムが常にオープン TCB で実行されるようにします。CICS が CICS コマンドを実行するため QR TCB へ切り替えた場合、制御をアプリケーション・プログラムに渡す前にオープン TCB に戻します。使用されるオープン TCB のタイプは、API 設定およびプログラムの言語によって異なります。

- JVM サーバーで実行される Java プログラムと OSGi バンドルは T8 TCB を使用します。
- C または C++ XPLink プログラムは、OPENAPI プログラムのように作動し、CICS キーが設定されている場合は X8 TCB を、ユーザー・キーが設定されている場合は X9 TCB を使用します。

- API(CICSAPI) を指定してかつ COBOL、PL/I、非 XPLink C または C++、およびアセンブラー言語で作成されたプログラムは、L8 TCB を使用します。これは、プログラムの実行キーに関係なく、CICS コマンドがこの TCB で作動するためです。

REQUIRED は、ユーザー・アプリケーション・プログラム、PLT プログラム、およびユーザー置換可能プログラムに適用でき、Java プログラムに対しては推奨オプションでもあります。REQUIRED 設定は、Db2 や IBM MQ のようなリソース・マネージャーにアクセスするプログラムにも適しています。こうしたプログラムも L8 TCB を必要とするものです。しかし、OPENAPI プログラムの場合、CICS は、MVS 要求などの非 CICS API コマンドが正しく作動するために、ユーザー・キー・プログラムには L9 TCB を使用し、CICS キー・プログラムには L8 TCB を使用する必要があります。 CONCURRENCY(REQUIRED) と API(OPENAPI) を設定して CICS-Db2 タスク関連ユーザー出口プログラムを実行した場合、そのプログラムは L8 TCB を使用することになります。あるいは、CONCURRENCY(REQUIRED) と API(OPENAPI) を設定して実行した場合、そのプログラムは適切なキー 8 のオープン TCB で実行されることになります。

また、プログラムの自動インストールがアクティブの場合には、プログラム自動インストール出口を使用して、プログラムの CONCURRENCY 属性を指定することもできます。

DATALOCATION({ANY|BELOW})

SET オプションを使用するコマンドはアプリケーション・プログラムにデータ・アドレスを返すことができます。このオペランドはデータのロケーションを指定するものです。例えば、コマンド **EXEC CICS RECEIVE SET(ptr-ref)** では、*ptr-ref* は、DATALOCATION(BELOW) が指定されている場合、16 MB より小さくなりますが、DATALOCATION(ANY) が指定されている場合、16 MB より大きくなることがあります。DATALOCATION は GETMAIN コマンドの操作に影響を与えません。このコマンドの要求に応えて CICS がどこからストレージを取得するかについて詳しくは、GETMAINを参照してください。

ANY プログラムは、31 ビット・アドレスを処理できます。これはデフォルトです。データのアドレスは、16 MB 境界の上でも下でも可能です。DATALOCATION 属性に指定された値は、リンク・エディット・プログラムのアドレッシング・モードとは無関係です。アドレッシング・モード AMODE=24 でリンク・エディットしたプログラムは、16 MB を超えるデータにアクセスできません。したがって、ユーザーが指定した値とリンク・エディットしたアプリケーション・プログラムのアドレッシング・モードに必ず互換性があるようにします。

- すべての 31 ビットまたは 64 ビット・プログラムについて ANY を指定してください。ただし、そうしたプログラムが CICS データ・アドレスを他の 24 ビット・プログラムに渡すものである場合はこの限りではありません。
- 16 MB を超えるストレージにアクセスするプログラムにストレージ・アドレスを渡したり、明示的にアドレッシング・モードを切り替えたりしない限り、AMODE=24 プログラムには DATALOCATION(BELOW) を指定します。

BELOW
プログラムは 24 ビット・アドレスのみを処理します。したがって、16 MB 境界より下に位置するデータのみを渡す必要があります。必要なら、アプリケーション・プログラムにアドレスを渡す前に、データを 16 MB 境界より下にコピーします。

DATALOCATION は、GETMAIN64 または GET64 CONTAINER コマンドの操作に影響を与えません。こうしたコマンドについて詳しくは、GETMAIN64およびGET64 CONTAINERを参照してください。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

DYNAMIC({**NO**|**YES**})
プログラムがプログラム・リンク要求の対象である場合に、要求を動的にルーティングできるかどうかを指定します。

NO プログラムがプログラム・リンク要求の対象である場合、動的ルーティング・プログラムは開始されません。

分散プログラム・リンク (DPL) 要求の場合、REMOTESYSTEM 属性か **EXEC CICS LINK** コマンドの SYSID オプションで、プログラムが実行するサーバー領域を明示的に指定しなければなりません。指定しないと、デフォルトのローカル領域になります。

YES プログラムがプログラム・リンク要求の対象である場合、CICS 動的ルーティング・プログラムが開始されます。リモート領域が **EXEC CICS LINK** コマンドの SYSID オプションで名前指定されていない場合は、ルーティング・プログラムはそのプログラムが実行する領域に要求をルーティングできます。

DYNAMIC 属性は REMOTESYSTEM 属性に優先します。
『REMOTESYSTEM』を参照してください。

DPL 要求の動的ルーティングについて詳しくは、DPL 要求の動的ルーティングを参照してください。

EXECKEY({**USER**|**CICS**})
CICS がプログラムに制御を渡すキーを指定し、プログラムで CICS キー・ストレージを変更するかどうかを決定します。再入可能プログラム (つまり、RENT 属性でリンク・エディットされるプログラム) 以外のプログラムでは、EXECKEY は、常駐モードとの併用で、どの DSA に CICS がプログラムをロードするかを定義します。

CICS プログラムが呼び出されると、CICS が CICS キーでプログラムに制御を渡します。 CICS は、プログラムに指定した常駐モードに応じて、CICS キー DSA の 1 つ (CDSA または ECDSA のいずれか) にプログラムをロードします。

ストレージ保護がアクティブな CICS 領域の場合、トランザクション分離がアクティブかどうかに関係なく、CICS キー・プログラムには、固

有のタスクおよび他のすべてのタスクの CICS キー・ストレージおよびユーザー・キー・ストレージへの読み取りおよび書き込みアクセスがあります。

USER プログラムが呼び出されると、CICS はユーザー・キーでプログラムに制御を渡します。 CICS は、プログラムに指定した常駐モードに応じて、ユーザー・キー共用 DSA の 1 つ (SDSA または ESDSA のいずれか) にプログラムをロードします。

ストレージ保護のみがアクティブな CICS 領域の場合、ユーザー・キー・プログラムには、すべてのユーザー・キー・ストレージへの読み取りおよび書き込みアクセスがありますが、CICS キー・ストレージに対しては読み取り専用アクセスです。

ストレージ保護およびトランザクション分離環境では、トランザクションが ISOLATE(YES) で定義されている場合、ユーザー・キー・プログラムには、そのタスク専用のユーザー・キー・タスク存続期間ストレージ、および任意の共用 DSA ストレージへの読み取りおよび書き込みアクセスがあります。

トランザクション分離環境でトランザクションが ISOLATE(NO) で定義されている場合、そのユーザー・キー・プログラムには、ISOLATE(NO) で定義されている他のトランザクションのユーザー・キー・タスク存続期間ストレージへの読み取りおよび書き込みアクセスがあります。

ユーザー・キー・プログラムには、常に CICS キー・ストレージへの読み取り専用アクセスがあります。

以下の場合、EXECKEY 属性は無視されます。

- 第 1 レベルのグローバル・ユーザー出口プログラム、タスク関連ユーザー出口プログラム、ユーザー置換可能プログラム、および PLT プログラムは、常に CICS キーで実行されます。
- プログラムが RENT 属性でリンク・エディットされている場合には、CICS は、プログラムに指定された常駐モードに応じて、プログラムを読み取り専用 DSA の 1 つ (RDSA または ERDSA のいずれか) にロードします。読み取り専用 DSA が読み取り専用ストレージから割り振られるのは、システム初期設定パラメーターとして RENTPGM=PROTECT が指定された場合のみです。
- COBOL 動的 CALL によって呼び出されるプログラムは、常に呼び出し元と同じキーで実行されます。

EXECUTIONSET({FULLAPI|DPLSUBSET})

CICS をプログラムにリンクし、リモート CICS 領域で実行しているかのようにプログラムを実行するかどうかを指定します。

DPLSUBSET

CICS とプログラムをリンクし、そのプログラムをリモート DPL プログラムの API 制限で実行する場合には、DPLSUBSET を指定します。DPL プログラムの API 制限について詳しくは、LINK コマンドの例外条件を参照してください。

FULLAPI
CICS とプログラムをリンクし、そのプログラムを DPL プログラムの API 制限なしで実行する場合は、FULLAPI を指定します。プログラムは、全 CICS API を使用できます。

以下の場合のみ、EXECUTIONSET 属性が適用されます。

- リンクされているプログラム。トランザクションによって最初に制御が渡されるプログラムには適用されません。
- REMOTESYSTEM 名がローカル CICS 領域と同じ名前の場合。その目的は、DPL プログラムとして実行しているかのように、プログラムをローカル CICS 環境でテストすることです。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

JVM({**NO**|**YES**})
このプログラムは、Java 仮想マシン (JVM) で実行する必要がある Java プログラムかどうかを指定します。

NO プログラムは JVM では実行されません。

YES プログラムは JVM で実行されます。 JVM(YES) を指定する場合は、JVMCLASS 属性にクラス名を指定します。

YES および NO に加えて DEBUG を指定することもできますが、これは、互換モードの場合のみです (異なる CICS リリース間での CSD の共用を参照してください)。

JVM が YES に設定されていて、かつ、JVMPROFILE (廃止された属性) にも値が設定されている場合、プログラムのインストールは禁止されます。

JVMCLASS(*class*)
Java プログラムのサービスの名前 (最大 255 文字の長さ) を指定します。

- JVM サーバーで実行される OSGi バンドルでは、この値は OSGi サービス名になります。OSGi バンドルが含まれる BUNDLE リソースをインストールする際に、その OSGi サービスを登録します。 CICS Explorerの「バンドル・パーツ (Bundle Parts)」ビューで OSGi サービスの名前を検索できます。
- JVM サーバーで実行される Java プログラムの場合、この値は、パッケージ名で修飾されたクラス名になります。
- Liberty JVM サーバー内のアプリケーションの場合、この値は `wlp:className#methodName` です。例えば、

wlp:com.ibm.cics.server.examples.tsq.ClassOne#myLinkMethod のようになります。メソッド名はオプションです。

この名前には大/小文字の区別があるため、英大文字と英小文字の正しい組み合わせで入力する必要があります。端末を使用している場合、大文字変換がサポートされているか確認してください。

JVMCLASS の値には、以下の文字を含めることができます。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

この属性は、JVM の制御の下で実行される Java アプリケーションにのみ適用されます。 JVM(NO) を指定した場合、CICS は、JVMCLASS で指定されたすべての値を無視します。

JVMPROFILE(*jvmprofile*)

この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。JVM サーバー上で Java プログラムを実行するときには、この属性ではなく、JVMSERVER 属性を使用してください。

JVMSERVER(*jvmserver*)

OSGi サービスを含む JVMSERVER リソース名 (長さ 8 文字以内) を指定します。 JVMSERVER リソースは、CICS での JVM サーバー実行時環境を表わします。 JVM サーバーは CICS キーにあるすべてのプログラムを実行します。

許容文字:
A-Z a-z 0-9 $ @ # . - _ % ? ! : | = , ;

LANGUAGE({**COBOL**|**ASSEMBLER**|**LE370**|**C**|**PLI**})

プログラム言語を指定します。

ASSEMBLER
LEASM 変換オプションを使用して変換されていないアセンブラー言語プログラム。LEASM は、Language Environment 準拠の MAIN プログラムであるアセンブラー・プログラムの変換に使用されます。

C Language Environment 準拠のコンパイラーでコンパイルされていない C または C++ プログラム。

COBOL
COBOL プログラム。

LE370 複数言語サポートを使用するプログラム、Language Environment 準拠のコンパイラーによってコンパイルされたプログラム、または Language Environment 準拠のプログラムを作成する LEASM オプションを使用して変換されたアセンブラー MAIN プログラム。

PLI PL/I プログラム。

ほとんどの場合、CICS プログラム・マネージャーによって正しい言語が推定され、ユーザーが指定した値は無視されるため、LANGUAGE 属性を指定する必要はありません。ただし、プログラムがアセンブラー言語で作成されたもので、

DFHEAI や DFHEAG スタブが存在しない場合、CICS はその言語を推定できないため、ユーザーが適切な値を指定する必要があります。ユーザーが言語を指定せず、CICS が言語を推定できなかった場合、そのプログラムを使用しようとするトランザクションは、ALIG というコードとともに異常終了します。

通常、この属性に値を指定する必要はありませんが、指定した値は、INQUIRE PROGRAM コマンドの LANGDEDUCED および LANGUAGE オプションで戻されることになります。この属性の値を変更すると、このコマンドを使用するプログラムに影響を与える可能性があります。

この属性は、JVM プログラムには適用されません。CICS は、プログラムが、JVM(YES) を指定したときに JVM の制御の下で実行する Java プログラムであることを判断します。

PROGRAM(*name*)

この PROGRAM 定義の名前を指定します。名前の最大長は 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

文字 DFH は、CICS が使用するために予約されているため、DFH で始まる名前は使用しないことが推奨されます。

CICS は、それらを削除または破棄したり、新たに機能や修正をアップグレードあるいは提供する際に CICS リソースに置き換えたりすることができます。

プログラムをアクティブな CICS 領域で使用するには、DFHRPL または動的 LIBRARY 連結の一部として指定されたライブラリーの 1 つにリンク・エディットする必要があります。再入可能なプログラムの場合、リンク・パック域 (LPA) に配置できます。アプリケーション・プログラムのインストールについて詳しくは、アプリケーション・プログラムのインストール を参照してください。

RELOAD({**NO**|**YES**})

プログラム制御リンク、ロード、または XCTL 要求によって、プログラムの新しいコピーが取り込まれるかどうかを指定します。この属性は、JVM プログラムには適用されません。

NO 現在ストレージにあるプログラムの有効なコピーが、要求に対して再使用されます。

YES 要求ごとにプログラムの新しいコピーがストレージに取り込まれます。さらに、これらのプログラム・コピーが不要になった場合は、トランザクションが終了する前にストレージ管理 FREEMAIN 要求を使用して、それぞれのコピーをストレージから明示的に除去する必要があります。関連する FREEMAIN 要求が発行されないと、DSA/EDSA の領域がアクセス不能プログラム・コピーと関連付けられてしまい、ストレージ不足やフラグメント化が生じる可能性があります。

注: LIBRARY 連結にプログラムの新しいバージョンが配置されている場合、新しいバージョンがロードされる前に、そのプログラムに対して NEWCOPY または PHASEIN を発行する必要があります。

RELOAD(YES) を使用して、関連するプログラムの実行によって変更されるテーブルまたは制御ブロックをロードできます。最初にロードされるタスク・プログラムに対しては、この値を指定しないでください。そのタスク・プログラムには FREEMAIN 要求を発行する方法がないためです。

再入不能プログラムには RELOAD(YES) を指定する必要があります。

RELOAD 属性について詳しくは、MVS リンク・パック域からのモジュール使用の制御 を参照してください。

REMOTENAME(*program*)

リモート CICS 領域にあるプログラム名を指定します。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

REMOTESYSTEM を指定して REMOTENAME を省略した場合、REMOTENAME 属性は、デフォルトでローカル名 (つまり、このリソース定義におけるプログラム名) と同じ名前になります。

REMOTESYSTEM(接続)

IPCONN 名、またはプログラムが常駐するリモート CICS 領域へのリンクを定義する CONNECTION リソース名を指定します。 CICS によって分散プログラム・リンク (DPL) 要求を別の CICS 領域にシップする場合にこの属性を指定します。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

DPL サーバー領域は、プログラム定義の REMOTESYSTEM 属性以外にも、以下のものを使用して指定できます。

- **EXEC CICS LINK PROGRAM** コマンドの SYSID オプションを使用したアプリケーション・プログラム
- 動的ルーティング・プログラム

優先順位の規則は以下のとおりです。

1. アプリケーション・プログラムが DPL 要求を発行し、**LINK** コマンドの SYSID オプションによってリモート CICS 領域が指定されている場合、CICS は要求をリモート領域へシップします。

インストールしたプログラム定義で DYNAMIC(YES) が指定されている場合、または、インストールされているプログラム定義がない場合、動的ルーティング・プログラムは、通知のためだけに呼び出され、要求をルーティングし直すことができません。

2. アプリケーション・プログラムが DPL 要求を発行しても、SYSID がローカル CICS 領域と同じ名前であるか、SYSID オプションが指定されていない場合、

   a. インストールされているプログラム定義で DYNAMIC(YES) が指定されている場合、またはインストールされているプログラム定義がない場合には、動的ルーティング・プログラムが呼び出され、それにより要求の経路指定が可能になります。

      指定されている場合、プログラム定義の REMOTESYSTEM 属性により、動的ルーティング・プログラムに渡すデフォルトのサーバー領域が指定されます。

   b. インストールされているプログラム定義で DYNAMIC(NO) が指定されている場合、CICS は REMOTESYSTEM 属性で指定されたリモート・システムに要求をシップします。 REMOTESYSTEM が指定されていない場合は、CICS はプログラムをローカルで実行します。

リモート・システム名を指定する際の規則は、CONNECTION リソース定義の CONNECTION 属性の規則と同じです。

注**:** 動的トランザクション・ルーティング・プログラム、または自動インストール・ユーザー・プログラムなどのユーザー作成 CICS プログラムにリモート属性を指定することはできません。

RESIDENT({**NO**|**YES**})
プログラムの常駐状況を指定します。この属性は、JVM プログラムには適用されません。

NO プログラムは、永久的には常駐しません。 RELOAD(YES) を指定する場合には、この値を指定する必要があります。

YES プログラムは、最初の参照でロードされた後は、仮想記憶域に永続的に常駐しますが、オペレーティング・システムによってページングすることが可能です。 RESIDENT(YES) を指定すると、CICS は、USAGE(NORMAL) が指定されたと見なします。

RESIDENT 属性の効果について詳しくは、常駐、非常駐、または一時としてのプログラムの定義 を参照してください。

RSL
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。詳しくは、Obsolete attributesを参照してください。

STATUS({**ENABLED**|**DISABLED**})
プログラム状況を指定します。

DISABLED
プログラムを使用できません。

ENABLED
プログラムを使用できます。

この属性は、CICS バンドルによって動的に生成された PROGRAM リソースでは無視されます。 PROGRAM リソースの初期状況は、そのリソースを定義するバンドルの初期状況から導出されます。

TRANSID(*name*)

動的プログラムの場合、この値は分散プログラム・リンク (DPL) 要求に使用されるデフォルトの TRANSID になります。プログラムが動的でない場合、この値には、リモート CICS を接続する、またその下でリモート・プログラムを実行する、トランザクションの名前を指定します。

TRANSID 属性にトランザクション名を指定しない場合、リモート領域は、以下の CICS 提供のデフォルト・ミラー・トランザクションのいずれかの下で DPL プログラムを実行します。リモート領域にユーザー・トランザクションを定義してミラー・プログラムを呼び出す場合は、そのトランザクション ID の下で DPL プログラムを実行します。

CPMI データ変換を必要とする LU6.2 接続、およびマルチプラットフォームの CICS Transaction Gateway からの TCP/IP および IPIC 要求に使用されるミラー・トランザクション。

CSMI 同期レベル 2 による MRO と LU6.2 接続、および z/OS の CICS Transaction Gateway からの EXCI および IPIC 要求に使用される CICS ミラー・トランザクション。

USAGE({**NORMAL**|**TRANSIENT**})

プログラムのストレージをいつ解放するかを指定します。この属性は、JVM プログラムには適用されません。

NORMAL

プログラムの常駐使用回数 (RESCOUNT) がゼロになると、そのプログラムは通常の動的プログラム・ストレージ圧縮処理の一部としてストレージから除去される対象になります。

RELOAD(YES) を指定する場合には、この値を指定する必要があります。

TRANSIENT

プログラムの常駐使用回数 (RESCOUNT) がゼロになると、そのプログラムのストレージを解放します。参照頻度の少ないプログラムにはこの値を指定します。

USELPACOPY({**NO**|**YES**})

プログラムをリンク・パック域 (LPA) から使用するかどうかを指定します。この属性は、JVM プログラムには適用されません。

NO プログラムは LPA からは使用されません。CICS アドレス・スペースにロードされます。

YES プログラムは、システム初期設定パラメーターとして LPA=YES が指定されている場合に、LPA から使用できます。LPA からプログラムを使用するには、プログラムがそこにインストールされていること、またプログラムが PRVMOD システム初期設定パラメーターによって指定されていないことが必要です。詳しくは、MVS リンク・パック域からのモジュール使用の制御を参照してください。

SESSIONS リソース

SESSIONS リソースは、システム間通信 (ISC) または複数領域操作 (MRO) を使用して通信する 2 つの CICS システム間の論理リンクを定義します。

2 つのシステムが ISC または MRO を使用して通信できるようにするには、事前にそれらのシステムが 1 つ以上のセッションによって論理的にリンクされている必要があります。論理リンクの性質によって、通信方法が決まります。アクティブ・システムに定義がインストールされている場合、CICS は、SESSIONS 名を使用しません。この名前が使用されるのは、CSDファイル内で定義を識別する場合だけです。

これらの SESSIONS がアクティブ・システムにインストールされるときに関連付けられる CONNECTION を指定するには、SESSIONS リソースの CONNECTION 属性を使用します。

さまざまなセッション・タイプの特別な考慮事項を次に示します。

MRO リンクおよびセッション
: MRO の SESSIONS 定義をインストールする場合は、この CICS と別の CICS の間の一連の並列セッションについて、CICS に通知してください。セッション数は、SENDCOUNT属性と RECEIVECOUNT属性で決まります。SENDセッションは、 SENDPFX属性とSENDCOUNT属性から作成された名前によって 識別されます。RECEIVEセッションは、RECEIVEPFX属性と RECEIVECOUNT属性から作成された名前によって識別されます。

APPC (LUTYPE6.2)リンクと並行セッション
: SESSIONS 定義をインストールすると、それらのセッションは MODENAME によって識別されるモードセットにグループ化されます (z/OS Communications Server のため)。個々のセッションの名前はカウンターに よって決まります。最初に作成されたセッションの名前は-999, 2番目は-998のようになります。このカウンターの値は、 ウォーム・スタートまたは緊急始動で保存されます。作成される セッション数は、SESSIONS定義のMAXIMUM属性によって制御 されます。

LUTYPE6.1 CICS-CICS ISCリンクとセッション
: CICS がセッションを識別する方法は、それらのセッションを定義した方法によって異なります。例えば、MRO セッションのように SENDPFX、SENDCOUNT、RECEIVEPFX、RECEIVECOUNT を使用したのか、CICS-IMS セッションのように SESSNAME を使用したのかなどです。

  注: すべての新規CICS-CICS ISCリンクにAPPCを使用します。

LUTYPE6.1 CICS-IMS リンクおよびセッション
: アクティブ CICS システムに SESSIONS 定義をインストールすると、CICS は、各セッションを SESSNAME 属性によって識別します。

INDIRECT 接続
: INDIRECTリンクと、それとの通信に使用される 中間システムとの関連はインストール時に作成される ので、INDIRECTリンクを定義する前に中間システムの 定義をインストールします。INDIRECT リンクを 最初にインストールする場合、中間定義が インストールされ、それを参照するその他

の インストール済み接続が解決されるまで、リンクは 休止しています。例えば、システムAがシステムBを介して 間接的にシステムCに接続している場合、システムAで、 次の各定義をこの順序でインストールします。

1. 中間システム：

```
CONNECTION(B) NETNAME(B) ACCESSMETHOD(IRC) ...
```

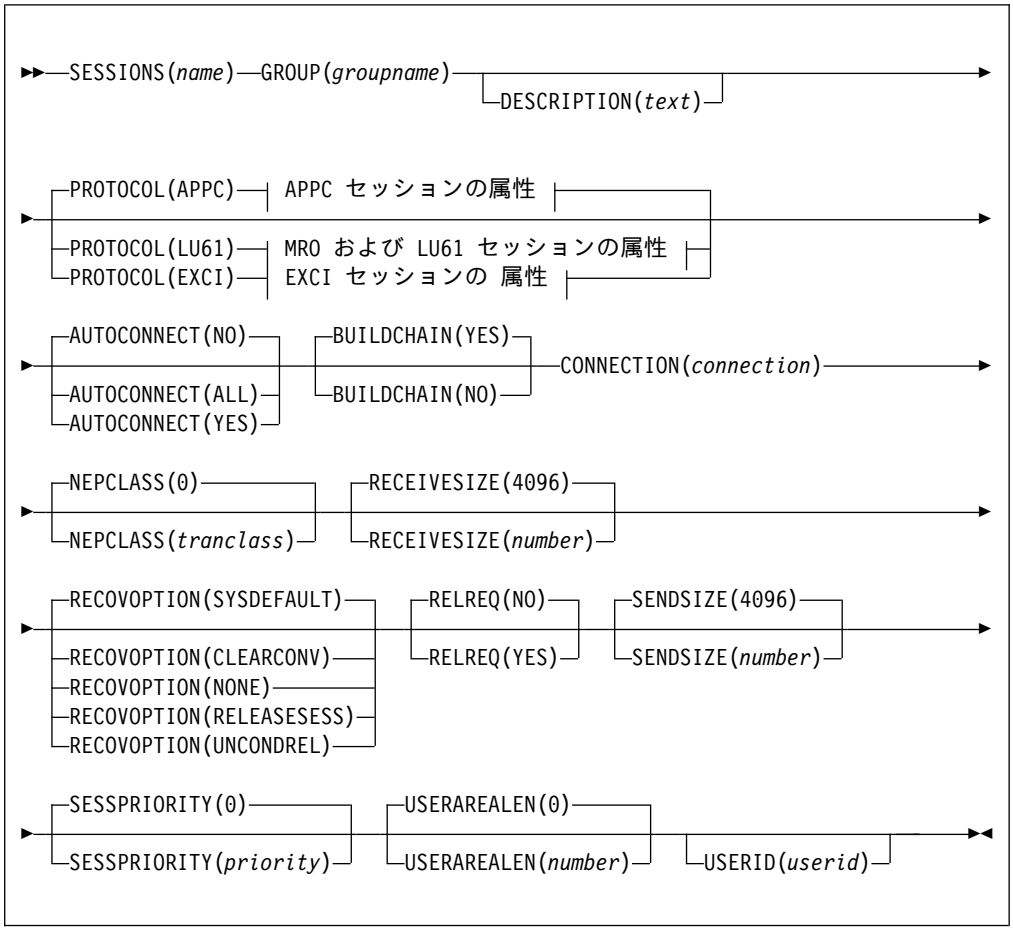
2. INDIRECTリンク

```
CONNECTION(C) NETNAME(C) ACCESSMETHOD(INDIRECT)
   INDSYS(B) ...
```

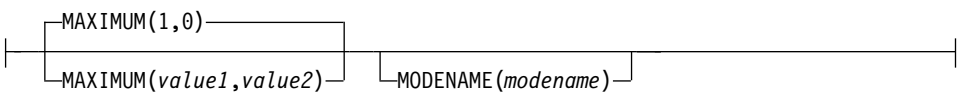
CEMT および CEDA を使用した SESSIONS リソース定義のインストールについては、SESSIONS リソース定義のインストールを参照してください。BAS の作業については、BAS セッション・リソース定義の作業を参照してください。

SESSIONS 属性

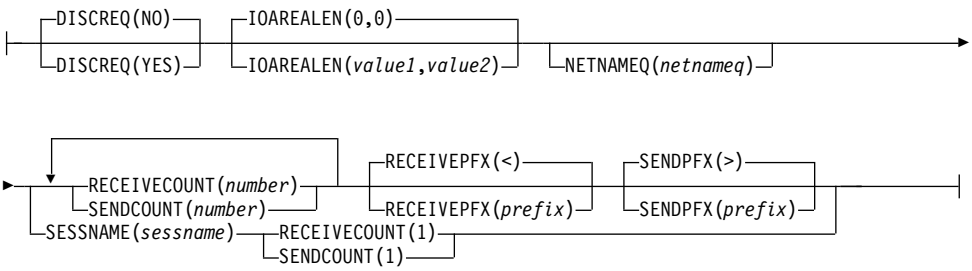
SESSIONS リソースの構文および属性について説明します。



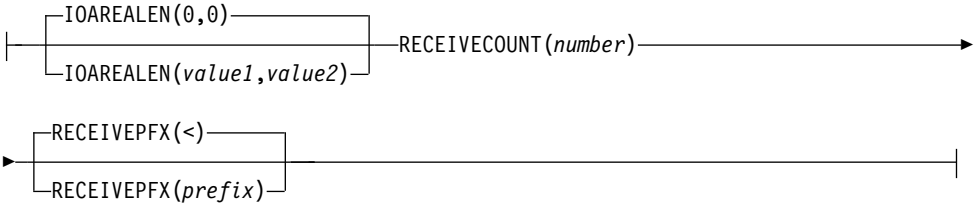
APPC セッションの属性:



MRO および LU61 セッションの属性:



EXCI セッションの属性:



AUTOCONNECT({NO|YES|ALL})

接続を確立する方法を指定します。LU6.1 および APPC セッションで指定する必要がある内容について、以下で説明します。

APPC セッション

z/OS Communications Server に接続されたシステムで、接続定義に AUTOCONNECT(YES) または (ALL) が指定されている場合:

NO CICS は接続の確立時にセッションをバインドしようとしません。ただし、実行される ACQUIRE CONNECTION 処理の一部として、1 つ以上のユーザー・セッションを割り振ることはあります。

YES または **ALL**

CICS の初期設定時、あるいは CEMT SET VTAM OPEN コマンドを使用して z/OS Communications Server との通信を開始するときに、競合勝者セッションが確立されます (つまり、BIND が実行されます)。リモート・システムが使用できないためにこの時点で接続を行えない場合、リモート・システムがそれまでの間に使用可能になって自動的に通信を開始しない限りは、CEMT SET

CONNECTION(sysid) INSERVICE ACQUIRED コマンドを使用して後でリンクを獲得する必要があります。

AUTOCONNECT(ALL) はバインド競合を引き起こす可能性があるため、他の CICS システムへのセッションには指定しないでください。

z/OS Communications Server に接続されたシステムで、CONNECTION 定義に AUTOCONNECT(NO) が指定されている場合:

ALL CEMT SET CONNECTION(name) ACQUIRED の発行により接続を獲得したとき、またはリモート・システムが自発的に通信を開始したときに、競合勝者セッションだけではなくすべてのセッションが確立されます。

NO CICS は接続の確立時にセッションをバインドしようとしません。ただし、実行される ACQUIRE CONNECTION 処理の一部として、1 つ以上のユーザー・セッションを割り振ることはあります。

YES CEMT SET CONNECTION(sysid) ACQUIRED の発行により接続を獲得したとき、またはリモート・システムが自発的に通信を開始したときに、競合勝者セッションが確立されます。

LU6.1 セッション

初期設定時または CEDA のインストール時に接続を確立する場合は、SESSIONS で AUTOCONNECT(YES) を指定します。

初期設定時にも CEDA のインストール時にも接続を確立しない場合は、SESSIONS で AUTOCONNECT(NO) を指定します。

BUILDCHAIN({**YES**|**NO**})

CICS が入力データをアプリケーション・プログラムに渡す前にチェーン・アセンブリーを実行するかどうかを指定します。

NO アプリケーション・プログラムがこの論理装置から受け取る TIOA には、要求単位 (RU) が 1 つ含まれます。

YES アプリケーション・プログラムがこの論理装置から受け取る端末入出力域 (TIOA) には、完全なチェーンが含まれます。

CONNECTION(*connection*)

このセッション定義で使用する接続定義の名前を指定します。名前の長さは最大 4 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

CONNECTION 定義は、SESSIONS 定義と同じ GROUP に属している必要があるので注意してください。

DESCRIPTION(*text*)

このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括

弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

DISCREQ({**NO**|**YES**})
切断要求を受け入れるどうかを指定します。DISCREQ は、LUTYPE6.1 ISC セッションには適用されますが、CICS が z/OS Communications Server 装置を取り扱わない MRO セッションには適用されません。

DISCREQ は、 APPC (LUTYPE6.2) セッションには適用されません。APPC を使用する場合は、トランザクションの必要に応じて個々のセッションが獲得され、順次解放されます。 APPC 論理装置間には複数のセッションが存在できるため、1 つの要求が他の要求を遅延させるという問題が発生することはありません。個々の APPC セッションを切断することはできません。代わりに、CEMT SET CONNECTION RELEASED コマンドを発行することができます。

NO CICS は、z/OS Communications Server 装置に対する切断要求を受け入れません。

YES CICS は、z/OS Communications Server 装置に対する切断要求を受け入れ、z/OS Communications Server CLSDST マクロ命令を発行して、その論理装置で z/OS Communications Server セッションを終了します。

DISCREQ(YES) を指定すると、端末から CESF LOGOFF コマンドまたは GOODNIGHT コマンドを発行しても切断が実行されます。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

INSERVICE
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

IOAREALEN({**0**|*value1*},{**0**|*value2*})
MRO リンクで送信されたメッセージの処理に使用される端末入出力域の長さをバイトで指定します。

value1 *value1* は、RECEIVE コマンドの発行時にアプリケーション・プログラムに渡される端末入出力域の最小サイズを指定します。

value2 *value2* が指定されていない場合、または *value1* より小さい場合は、デフォルトで *value1* の値が使用されます。

value2 には、 *value1* 以上の値を指定できます。この場合、入力メッセージのサイズが value1 を超えると、CICS は、端末入出力域 (TIOA)

の長さに *value2* バイトを使用します。トランザクションが MRO リンクで接続されている場合、CICS は、最初の入力メッセージを入れるのに十分な長さの TIOA を使用します。そうでない場合、入力メッセージのサイズが *value2* も超えると、ノード異常条件プログラムによって端末に例外応答が送信されます。

指定する IOAREALEN の値が MRO リンクで送信されるメッセージのほとんどに対して大きすぎると、実記憶域や仮想記憶域を浪費することになります。一方、指定する IOAREALEN の値がゼロか、ほとんどのメッセージより小さい場合は、FREEMAIN アクティビティーと GETMAIN アクティビティーが頻繁に発生する可能性があります。その結果、追加のプロセッサーが必要になります。

MAXIMUM({**1**|*value1*},{**0**|*value2*}) **(APPC のみ)**
このモードセットでサポートされるセッションの最大数を指定します。value1 は、value2 以上でなければなりません。

1|*value1*
グループ内のセッションの最大数。 この値の範囲は、1から999です。デフォルトは、1 です。

0|*value2*
コンテンション勝者としてサポートされるセッションの最大数。 指定できる値の範囲は、0 から 999 までです。デフォルトは 0 です。このオペランドは、単一セッション接続では意味が無いことに注意してください。

SNA では、いくつかのリソース (例えば交換回線) を、ネットワークで限定リソースと定義することができます。バインド時に、z/OS Communications Server は、バインドが限定リソース上にあるかどうかを CICS に示します。CICS タスクが限定リソース全体でセッションを解放すると、他にそのセッションを使用するタスクがなければ、CICS はセッションをアンバインドします。

セッションが限定リソースを使用する場合は、 **MAXIMUM(***value1***,0)** と指定します。これで、アンバインド済みセッションがすべてリセットされ、次に必要なときには、どちら側でもそのセッションを勝者としてバインドできます。

MAXIMUM オプションの効果、および限定リソースの使用について詳しくは、SESSIONS リソースの MAXIMUM オプションの影響 を参照してください。

MODENAME(*modename*) **(APPC のみ)**
APPC 接続で使用するセッション・グループの識別名を指定します。この名前は、最大 8 文字の長さで指定することができ、z/OS Communications Server に定義された z/OS Communications Server LOGMODE エントリーの名前でなければなりません。予約名 SNASVCMG を使用することはできません。モード名を省略した場合、デフォルトでブランクになります。 z/OS Communications Server のモード名について詳しくは、APPC セッション・グループの定義 を参照してください。

MODENAME は、システム間リンクで定義されたセッション・グループごとに固有でなければなりません。つまり、1 つの CONNECTION 定義に関連した SESSIONS 定義の間では、MODENAME が固有である必要があります。これは、z/OS Communications Server に LOGMODE 名として渡されます。

NEPCLASS({**0**|*tranclass*})

ノード・エラー・プログラムのトランザクション・クラスを指定します。この値は、デフォルトとして機能します。

0 デフォルトのノード・エラー・プログラム・モジュールへのリンクになります。

tranclass

(デフォルトではない) ノード・エラー・プログラム・モジュールのトランザクション・クラス。この値は 1 から 255 の範囲です。ノード・エラー・プログラムのプログラミング情報については、ノード・エラー・プログラムの作成 を参照してください。

NEPCLASS 属性は、 SNASVCMGR セッションでは無視されます。

NETNAMEQ(*netnameq*)

この特定のセッションの、IMS リモート・システムによって認識されている名前を指定します。これは、CICS-IMS セッションに使用されます。 名前の最大長は 8 文字です。許容文字は、A-Z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < and > です。CREATE コマンドを使用する場合を除いて、小文字は大文字に変換されます。

OPERID

OPERPRIORITY

OPERRSL

OPERSECURITY

これらの属性は廃止されましたが、以前のリリースの CICS との互換性を確保するためにサポートされています。

PROTOCOL({**APPC**|**LU61**|**EXCI**})

相互通信リンク (ISC または MRO) で使用するプロトコルのタイプを指定します。

APPC (LUTYPE6.2)
拡張プログラム間通信機能 (APPC) プロトコル。CICS-CICS ISC にはこれを指定します。

EXCI 外部 CICS インターフェース。これを指定すると、セッションが外部 CICS インターフェースを使用する非 CICS クライアント・プログラムによって使用されることが示されます。

LU61 LUTYPE6.1 プロトコル。CICS-CICS ISC、CICS-IMS、または MRO の場合に指定します。

RECEIVECOUNT(*number*)

MRO および z/OS Communications Server LU6.1 セッションの場合、および EXCI クライアントを使用するセッションの場合は、受信セッション (つまり、通常は送信する前に受信するセッション) の数を指定します。

- MRO受信セッション(EXCIクライアントを持つセッションを 含む）は常に送信前に受信します。
- z/OS Communications Server LU6.1 受信セッションは、通常は送信前に受信しますが、適切な送信セッションが不足している場合は、受信前に送信することがあります。

RECEIVECOUNT属性を指定しない場合、 受信セッションはありません。

指定できる受信セッションの数は、RECEIVEPFX 属性で指定した接頭部の長さによって決まります。

- デフォルトの受信接頭部 (<)、または独自の 1 文字の接頭部を使用する場合、指定できる受信セッションの数は 1 から 999 までです。
- 2 文字の接頭部を使用する場合、指定できる受信セッション数は 1 から 99 までです。

指定した値がパートナー・システムの送信セッション数と一致していることも確認する必要があります。

- パートナーが別の CICS システムである場合、この値はパートナー・システムで指定された SENDCOUNT と一致する必要があります。
- パートナーが EXCI クライアントである場合、パートナーの送信セッション数を指定することはできません。ただし、EXCI アドレス・スペースには、送信セッションの上限があります。この限界に達すると、IRPはSYSTEM_ERROR 理由コード608でそれ以上のセッションの要求を拒否します。 CICS システムの制限を指定する方法については、『インストール』の『EXCI パイプ割り振り』 を参照してください。

RECEIVEPFX({**<**|*prefix*})

CICS が受信セッション名 (そのセッションの端末管理テーブルの端末入力 (TCTTE) の名前) の最初の 1 または 2 文字に使用する、1 文字または 2 文字の接頭部を指定します。

接頭部は、既存の接続名や端末名と競合するものであってはなりません。

< (MRO および EXCI セッション)

MRO および EXCI セッションで独自の受信接頭部を指定しない場合、CICS は、デフォルトの接頭部 (より小記号 (<)) を使用します。この接頭部に受信数を結合して、受信セッション名が生成されます。

セッション名の最後の 3 文字は、CICS が作成します。許容文字は A-Z、1-9 です。この 3 文字の ID は文字 AAA から始まり、セッション・エントリーの数が RECEIVECOUNT 値で設定した制限に達するまで、昇順で続きます。受信セッション名が生成されるのは送信セッションの後であり、送信セッション名に続いて同じ順序で生成されます。

例えば、送信セッションで最後に生成されるセッション名が、デフォルトの送信接頭部 (>) を使用した >AAJ である場合、CICS は受信セッション名を <AAK、<AAL、<AAM のように生成します。 (セッション ID の生成方法は、最初の接頭部記号を除いて、APPC セッションと同じです)。

46656 (<AAA から <999) を超えるセッションを使用する場合、CICS は、APPC セッションと同様の方法で、次の範囲 (AAA< から 999<) を再度割り振ります。

領域内のセッション数が 46656 を超えると、パフォーマンスが低下する場合があります。代替案として、CICS 領域の数を増やすことを検討してください。

MRO セッションは最大 93312 まで定義できますが、現在は、一度に 65535 を超えるセッションは獲得できないという制限があります。これは、CICS の始動時、あるいは 65536 を超えるセッションの CEDA イ

ンストール時に、すべてのパートナー領域が稼働中である場合に発生する可能性があります。制限を超えるセッションは、あとで獲得できます。

注: 独自の接頭部を指定した場合、CICS は、LUTYPE6.1 セッションの場合と同じ方法でセッション名を生成します。

prefix **(LUTYPE6.1** セッション**)**
セッションが LUTYPE6.1 ISC 接続上にある場合は、1 文字または 2 文字の接頭部を指定する必要があります。デフォルトの < 記号は LUTYPE6.1 セッションには使用できません。

LUTYPE6.1 セッションでは (独自の 1 文字または 2 文字の接頭部を指定した場合は MRO でも)、CICS は、接頭部に 1 から 99 または 1 から 999 の範囲の番号を接頭部に付加して、セッション名を生成します。番号は 1 から始まり、指定した RECEIVECOUNT に達するまで 1 ずつ増えていきます。

RECEIVESIZE({**4096**|*number*})
これらのセッションが受信可能な z/OS Communications Server 要求単位 (RU) の最大サイズを指定します。値は、LU61 セッションでは 1 から 30720 の間に、または APPC セッションでは 256 から 30720 の間に設定する必要があります。デフォルトは 4096 です。

指定した値は、接続されている論理装置に送信されます。この値は、体系化された形式で送信する必要があるため、指定した値によっては CICS によって端数が切り捨てられる場合があります。この値は、バインド時の折衝によりさらに縮小される場合があります。

CICS が 2 次 LU セッションである場合、この値は、これらのセッションが送信できる z/OS Communications Server 要求単位 (RU) の最大サイズを示します。

RECOVNOTIFY
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

RECOVOPTION({**SYSDEFAULT**|**CLEARCONV**| **RELEASESESS**|**UNCONDREL**|**NONE**})
このオプションは、z/OS Communications Server 持続セッションまたは XRF で実行されている CICS 領域内のセッションのリカバリーに適用されます。

z/OS Communications Server 持続セッション: 持続セッションのサポートありで稼働している CICS 領域内で、このオプションは、CICS がセッションをリカバリーし、持続セッション遅延間隔内のシステム再始動時に端末をサービスに返す方法を指定します。

XRF: XRF のサポートありで稼働している CICS 領域内で、このオプションは、CICS がセッションをリカバリーし、XRF テークオーバー後に端末をサービスに返す方法を指定します。

NONE を除くすべてのリカバリー・オプションでは、実行されるアクションが z/OS Communications Server UNBIND である場合、UNBIND に続いて z/OS Communications Server SIMLOGON が実行されます。

CLEARCONV

z/OS Communications Server 持続セッション**:** APPC セッションでは、CLEARCONV はサポートされません。デフォルトは SYSDEFAULT です。

XRF: AUTOCONNECT(YES) と指定すると、セッションが再開されます。AUTOCONNECT(NO) と指定した場合は、セッションはアンバインドされます。

NONE

z/OS Communications Server 持続セッション: 持続セッションのサポートありで稼働している CICS 領域では、これは、持続セッション遅延期間内の再始動時にセッションのリカバリーを行わないことを指定します。実際に、モードグループ上のセッションには、持続セッションのサポートはありません。 LU6.2 セッションはアンバインドされ、モードグループ CNOS 値はゼロにリセットされます。 AUTOCONNECT (YES) が指定されている場合、セッションはシステムの再始動後に自動的に再接続されます。

XRF: XRF のサポートありで稼働している CICS 領域では、これは、代替システムによってログオン状態が追跡されず、テークオーバー後に端末セッションが自動的にリカバリーされないことを指定します。実際に、端末には XRF サポートはありません。 AUTOCONNECT(YES) を指定すると、代替システムはテークオーバー後に端末を自動で再接続します。

RELEASESESS

z/OS Communications Server 持続セッション**:** APPC セッションでは、RELEASESESS はサポートされません。デフォルトは SYSDEFAULT です。

XRF: AUTOCONNECT(YES) と指定すると、セッションが再開されます。AUTOCONNECT(NO) と指定した場合は、セッションはアンバインドされます。

SYSDEFAULT

z/OS Communications Server 持続セッション: 持続セッションのサポートありで稼働している CICS 領域では、これは、持続セッション遅延期間内のシステム再始動時にセッションをリカバリーするための最適な手順を、セッション・アクティビティーおよび端末の特性に応じて CICS が選択することを指定します。

セッションはリカバリーされますが、障害の発生時に実行中であったトランザクションは異常終了し、リカバリーされません。トランザクションは、リカバリーされたセッションが APPC 接続による別の CICS 領域で使用されている場合にも異常終了します。

CICS は、以下のいずれかの方法によって、考えられる影響が最も小さいセッションをリカバリーします。

- CICS で障害が発生したときにセッションがビジーではなかった場合、アクションは不要です。

- CICS で障害が発生したときにセッションがビジーであった場合、CICS は、その障害の発生時に進行中であった APPC 会話に対して DEALLOCATE(ABEND) (EXEC CICS ISSUE ABEND と同等) を発行します。
- これらのいずれにも当てはまらない場合、セッションはアンバインドされます。

XRF: AUTOCONNECT(YES) と指定すると、セッションが再開されます。AUTOCONNECT(NO) と指定した場合は、セッションはアンバインドされます。

UNCONDREL
UNBIND 要求を送信してアクティブ・セッションを解放するように、CICS に要求します。システム再始動時 (持続セッションがサポートされている場合) またはテークオーバー時 (XRF の場合) にセッションが使用中だったかどうかにかかわりなく、 UNBIND が送信されます。

RELREQ({**NO**|**YES**})
別の z/OS Communications Server アプリケーション・プログラムによる要求時に、CICS が論理装置を解放するかどうかを指定します。

SENDCOUNT(*number*)
MRO セッションおよび z/OS Communications Server LU6.1 セッションの場合にのみ、送信セッション (つまり、通常は受信する前に送信するセッション) の数を指定します。
- MRO 送信セッションは、常に受信前に送信します。
- z/OS Communications Server LU6.1 送信セッションは、通常は受信前に送信しますが、適切な受信セッションが不足している場合は、送信前に受信することがあります。

指定できる送信セッションの数は、SENDPFX 属性で指定した接頭部の長さによって決まります。
- デフォルトの送信接頭部 (>)、または独自の 1 文字の接頭部を使用する場合、指定できる送信セッションの数は 1 から 999 までです。
- 2 文字の接頭部を使用する場合、指定できる送信セッション数は 1 から 99 までです。

指定した値がパートナー・システムの受信セッション数と一致していることも確認する必要があります。
- パートナーが別の CICS システムである場合、この値はパートナー・システムで指定された RECEIVECOUNT と一致する必要があります。

SENDCOUNT 属性を指定しない場合、送信セッションはありません。パートナーが EXCI クライアントである場合は、SENDCOUNT 属性を指定しないでください。

SENDPFX({**>**|*prefix*})
CICS が送信セッション名 (そのセッションの端末管理テーブルの端末入力 (TCTTE) の名前) の最初の 1 または 2 文字に使用する、1 文字または 2 文字の接頭部を指定します。

接頭部は、既存の接続名や端末名と競合するものであってはなりません。

> **(MRO** セッション**)**

MRO セッションで独自の送信接頭部を指定しない場合、CICS は、デフォルトの接頭部 (より大記号 (>)) を使用します。この接頭部に送信数を結合して、送信セッション名が生成されます。

CICS は、セッション名の最後の 3 文字を、英数字 (A から Z、1 から 9) から作成します。この 3 文字の ID は文字 AAA から始まり、セッション・エントリーの数が SENDCOUNT 値で設定した制限に達するまで、昇順で続きます。

例えば、デフォルト接頭部 (>) を使用した場合、CICS は、>AAA、>AAB、>AAC のようにセッション名を生成します。46656 (>AAA から >999) を超えるセッションを使用する場合、CICS は、次の範囲 (AAA> から 999>) を割り振ります。(このセッション ID の生成方法は、最初の記号を除いて、APPC セッションの場合と同じです。)

領域内のセッション数が 46656 を超えると、パフォーマンスが低下する場合があります。代替案として、CICS 領域の数を増やすことを検討してください。

MRO セッションは最大 93312 まで定義できますが、現在は、一度に 65535 を超えるセッションは獲得できないという制限があります。これは、CICS の始動時、あるいは 65536 セッションを超える CEDA のインストール時に、すべてのパートナー領域が稼働している場合に発生する可能性があります。制限を超えるセッションは、あとで獲得できます。

注**:** 独自の接頭部を指定した場合、CICS は、LUTYPE6.1 セッションの場合と同じ方法でセッション名を生成します。

prefix **(LUTYPE6.1** セッションの場合**)**

セッションが LUTYPE6.1 ISC 接続上にある場合は、1 文字または 2 文字の接頭部を指定する必要があります。デフォルトの > 記号は LUTYPE6.1 セッションには使用できません。

LUTYPE6.1 セッションでは (独自の 1 文字または 2 文字の接頭部を指定した場合は MRO でも)、CICS は、接頭部に 1 から 99 または 1 から 999 の範囲の番号を接頭部に付加して、セッション名を生成します。番号は 1 から始まり、指定した SENDCOUNT に達するまで 1 ずつ増えていきます。

SENDSIZE({**4096**|**number**})

これらのセッションが送信可能な z/OS Communications Server 要求単位 (RU) の最大サイズを指定します。値は、LU61 セッションでは 1 から 30720 の間に、または APPC セッションでは 256 から 30720 の間に設定する必要があります。デフォルトは 4096 です。 この値は、バインド時のネゴシエーションによって小さくされる場合があります。 SENDSIZE の値を大きくすると、そのセッションに割り振られるストレージは増えますが、2 つのノード間で送信される物理メッセージの数は減少する場合があります。

CICS が 2 次 LU セッションである場合、この属性は、これらのセッションが受信できる z/OS Communications Server 要求単位 (RU) の最大サイズを示します。値は 256 から 30720 の範囲内である必要があります。

SESSIONS(*name*)
この SESSIONS 定義の名前を指定します。名前の最大長は 8 文字です。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

この名前は、CSD ファイルで SESSIONS 定義を識別するために使用されます。アクティブな CICS システム内では使用されません。

SESSNAME(*sessname*)
CICS 相互通信並列セッションで、セッション修飾子ペアのローカル側の一方として使用されるシンボリック ID を指定します。名前の長さは最大 4 文字です。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

SESSPRIORITY({**0**|*priority*})
端末の優先度を指定します。この 10 進値 (0 から 255) は、全体的なトランザクション処理の優先順位の設定に使用されます。 (トランザクション処理の優先度は、端末優先度、トランザクション優先度、およびオペレーター優先度の合計と等しくなります。この値は 255 を超えることはできません。)

TRANSACTION
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

USERAREALEN({**0**|*number*})
このセッションのユーザー域の長さを 0 から 255 の範囲 (単位はバイト) で指定します。できるだけ小さい値を指定してください。セッションがインストールされると、端末ユーザー域はゼロに初期化されます。

端末ユーザー域は、**TCTUALOC** システム初期設定パラメーターの値に応じて、31 ビット・ストレージあるいは 24 ビット・ストレージに配置されている場合があります。デフォルトでは、端末ユーザー域は、31 ビット・ストレージあるいは 24 ビット・ストレージのいずれに配置することも可能ですが、CICS は、31 ビット・ストレージが使用できる場合は、常にそちらを使用します。 31 ビット・アドレッシングに対応できないアプリケーション・プログラムがあるために、端末ユーザー域を 24 ビット・ストレージに保管する必要がある場合は、CICS 領域にシステム初期設定パラメーター **TCTUALOC=BELOW** を指定してください。

USERID(*userid*)
サインオンに使用され (SEC=YES または MIGRATE)、セキュリティー・エラー・メッセージ、セキュリティー違反メッセージ、および監査証跡で参照されるユーザー ID を指定します。これは、セキュリティー・マネージャーに定義されている有効なユーザー ID でなければなりません。そうでない場合は、オペレーターがサインオンできません。保護リソースへのすべてのアクセスは、USERID に依存します。

この USERID は、 CONNECTION 定義で指定した SECURITYNAME を指定
変更します。

名前の最大長は 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

TCPIPSERVICE リソース

TCPIPSERVICE リソースは、どの TCP/IP サービスが CICS 内部ソケット・サポートを使用するかを定義します。

TCPIPSERVICE リソースを使用して定義することができる CICS サービスは、ECI over TCP/IP (CICS クライアント用)、CICS Web サポート (HTTP)、IPIC、またはユーザー定義のプロトコルです。 TCPIPSERVICE 定義により、これらの内部 CICS インターフェースを、複数のポートで listen する CICS を使って、異なるポート上で ECI、CICS Web サポート、あるいはユーザー定義プロトコルの異なるフレーバーを使用して管理できます。

TCPIPSERVICE 定義は、CICS 提供の TCP/IP サービスでのみ使用することができ、z/OS Communications Server IP CICS ソケット・インターフェースとは関係がありません。 CICS の TCP/IP ソケット・インターフェースは、z/OS Communications Server と共に提供され、z/OS の不可欠な部分であり、CICS SO ドメインを使用しません。

BAS の作業については、BAS TCPIPSERVICE リソース定義の作業を参照してください。

CICS バンドル内の TCPIPSERVICE リソース

CICS バンドルを使用して、TCPIPSERVICE リソース定義を作成、編集、およびインストールすることができます。この方法で TCPIPSERVICE リソースを作成した場合、CICS バンドルを使用してそのリソースのライフサイクルを管理する必要があり、リソースを単独で管理することはできません。

EXEC CICS INQUIRE TCPIPSERVICE コマンドまたは **CEMT INQUIRE TCPIPSERVICE** コマンドを使用して、CICS バンドルによって動的に生成される TCPIPSERVICE リソースを照会することができます。

CICS バンドルによって動的に生成された TCPIPSERVICE リソースに対して DISCARD コマンドを発行することはできません。 CICS バンドルを破棄する必要があります。それにより、CICS が TCPIPSERVICE リソースに操作を適用します。

CICS バンドル内で定義されてインストールされた TCPIPSERVICE リソースの属性を変更するには、CICS Explorer のリソース・エディターを使用して CICS バンドル内の定義を変更し、新規バージョンの CICS バンドル、またはそのデプロイに

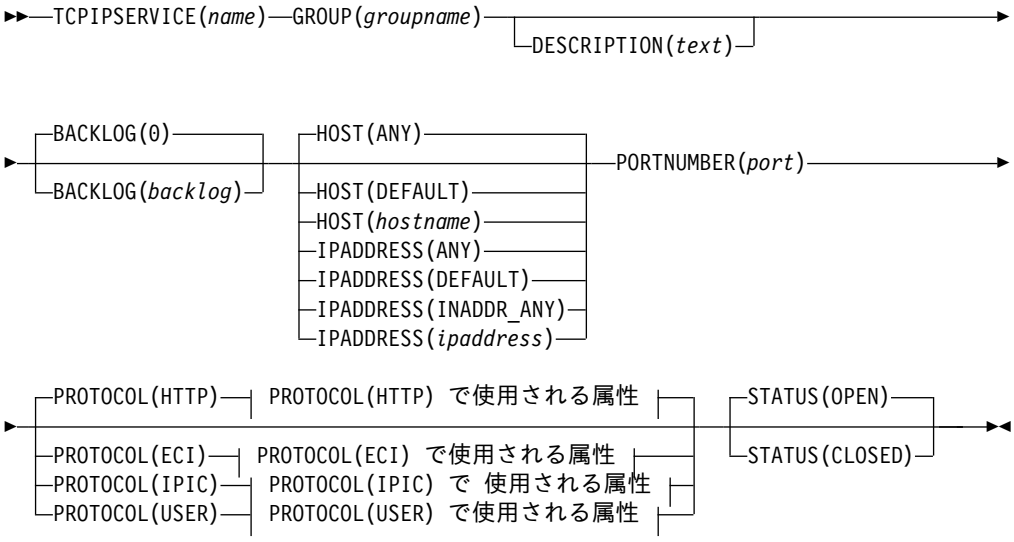
使用したアプリケーションをインストールします。 SET TCPIPSERVICE コマンドを使用することで、動的に生成されたリソースの属性を変更することはできますが、これらの変更はカタログされないため、CICS のウォーム・リスタート後には回復されません。

CICS バンドルで定義されてインストールされた TCPIPSERVICE リソースの状況を変更するには、CICS バンドルを使用可能/使用不可にします。 CICS バンドルを使用不可にしたが、サービスがまだ閉じられていない場合は、動的に生成されたリソースに対して SET TCPIPSERVICE IMMCLOSE コマンドを発行することで、サービスを即時に閉じることができます。

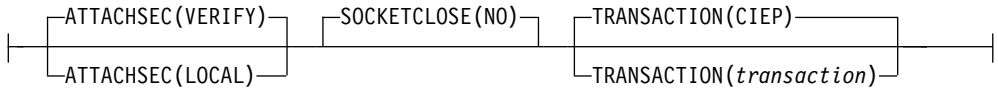
CICS バンドルでのリソースの定義について詳しくは、CICS バンドルの定義を参照してください。

TCPIPSERVICE の属性

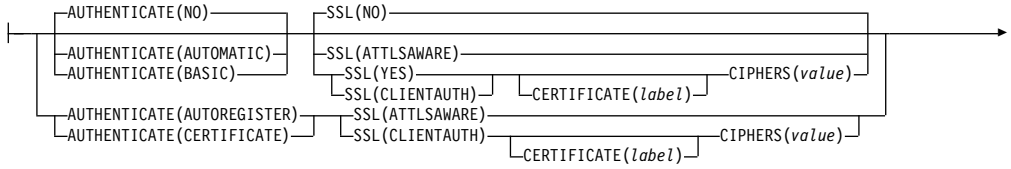
TCPIPSERVICE リソースの構文と属性について説明します。

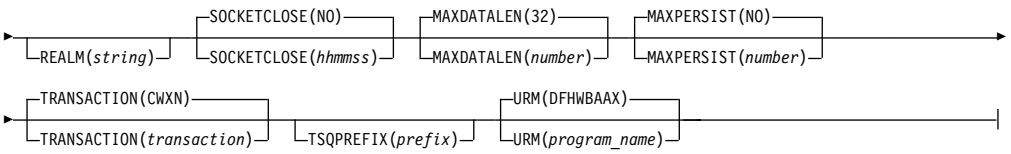


PROTOCOL(ECI) で使用される属性:

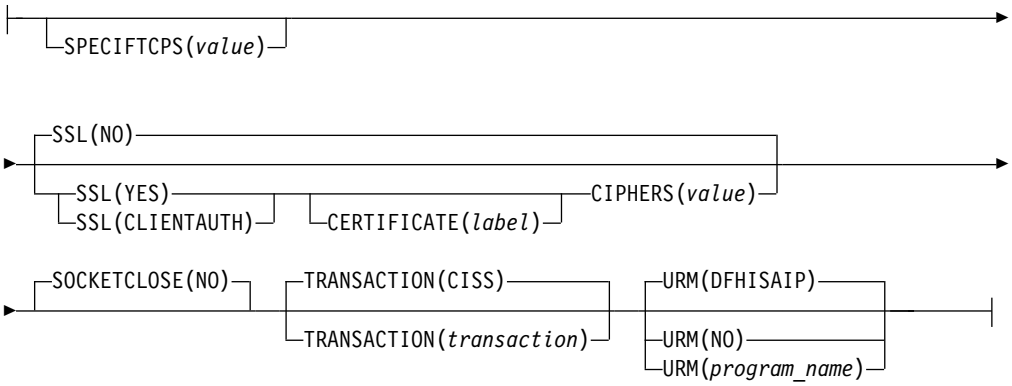


PROTOCOL(HTTP) で使用される属性:

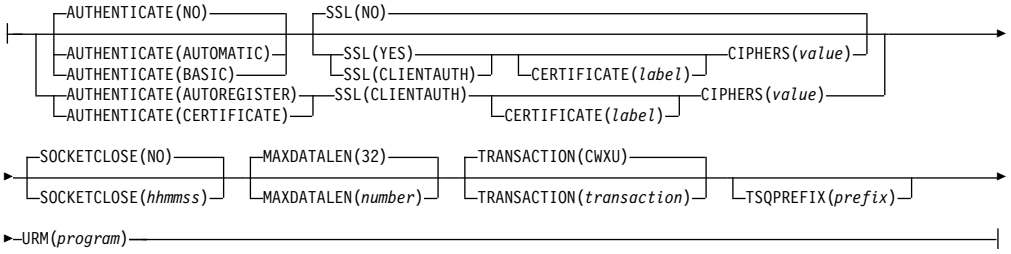




PROTOCOL(IPIC) で使用される属性**:**



PROTOCOL(USER) で使用される属性**:**



ATTACHSEC({LOCAL|VERIFY})

この接続に必要な接続時ユーザー・セキュリティーのレベルを指定します。

このオプションは、PROTOCOL(ECI) でのみ有効です。

LOCAL

CICS がクライアントのユーザー ID またはパスワード (またはパスワード・フレーズ) を必要としないことを指定します。

VERIFY

着信接続要求でユーザー ID とユーザー・パスワードまたはパスワード・フレーズを指定する必要があるということを指定します。接続するシステムが不明な場合や信頼できない場合は、VERIFY を指定します。

AUTHENTICATE({NO|ASSERTED|AUTOMATIC|AUTOREGISTER|BASIC| CERTIFICATE})

HTTP および USER プロトコルのインバウンド TCP/IP 接続で使用する認証および識別方式を指定します。IPIC プロトコルの場合、この属性は適用されません。ECI プロトコルの場合、この属性は無効です。認証について詳しくは、識別と認証 を参照してください。

ASSERTED

TCPIPSERVICE 定義の ASSERTED 属性は廃止されましたが、互換性を確保するために保持されています。

PROTOCOL(HTTP) または **PROTOCOL(USER)** が指定されている場合、以下のようになります。

NO クライアントが認証または識別情報を送信する必要はありません。ただし、クライアントが、セキュリティー・マネージャーにすでに登録済みの、ユーザー ID に関連した有効な証明書を送信した場合、そのユーザー ID はクライアントを識別します。

BASIC

HTTP 基本認証を使用して、クライアントからユーザー ID およびパスワードまたはパスワード・フレーズを取得します。

クライアントが Authorization ヘッダーを送信した場合、その内容がユーザー ID とパスワードまたはパスワード・フレーズとしてデコードされます。そのユーザー ID とパスワードまたはパスワード・フレーズが無効の場合は、WWW-Authenticate ヘッダーとともに HTTP 401 応答が戻され、クライアント・プログラムからユーザーに、新しいユーザー ID とパスワードまたはパスワード・フレーズを求めるプロンプトが出されます。このプロセスは、クライアントが有効なユーザー ID とパスワードまたはパスワード・フレーズを提供するか、接続を取り消すまで続行されます。

ユーザーが正常に認証された場合は、提供されたユーザー ID でクライアントが識別されます。

AUTOREGISTER

SSL クライアントの証明書認証を使用して、クライアントを認証します。

- クライアントが、セキュリティー・マネージャーに既に登録済みの、ユーザー ID に関連付けられた有効な証明書を送信した場合、そのユーザー ID はクライアントを識別します。
- クライアントが、セキュリティー・マネージャーに登録されていない有効な証明書を送信した場合、HTTP 基本認証を使用して、クライアントからユーザー ID およびパスワードまたはパスワード・フレーズを取得します。パスワードまたはパスワード・フレーズが有効な場合、CICS はその証明書をセキュリティー・マネージャーに登録し、ユーザー ID と関連付けます。そのユーザー ID によって、クライアントが識別されます。
- 証明書を登録するには、クライアントについて RACF 内に適切なアクセス権限の定義が必要です。必要なアクセス権限は、FACILITY クラスの IRR.DIGTCERT.ADD プロファイルに対する READ アクセス権限です。

注**:** AUTHENTICATE(AUTOREGISTER) を指定する場合は、SSL(CLIENTAUTH | ATTLSAWARE) も指定する必要があります。

AUTOMATIC
この機能は、AUTOREGISTER と BASIC の機能を組み合わせたものです。

- クライアントが証明書を送信すると、AUTOREGISTER に記載された内容に従って処理が続行されます。
- クライアントが証明書を送信しない場合は、BASIC に記載された内容に従って処理が続行されます。

CERTIFICATE
SSL クライアント証明書認証を使用して、クライアントの認証および識別が行われます。クライアントは、セキュリティー・マネージャーに既に登録済みの、ユーザー ID に関連付けられた有効な証明書を送信する必要があります。有効な証明書を受信しなかったか、または証明書がユーザー ID に関連付けられていない場合、接続はリジェクトされます。

ユーザーが正常に認証されると、証明書に関連付けられたユーザー ID によってクライアントが識別されます。

注: AUTHENTICATE(CERTIFICATE) を指定する場合は、SSL(CLIENTAUTH | ATTLSAWARE) も指定する必要があります。

注: HTTP または USER プロトコルでは、ユーザー置き換え可能モジュール (URM) 属性で指定するアナライザー・プログラムによって、認証プロセスで提供したユーザー ID を変更できます。認証プロセスでユーザー ID が提供されなかった場合は、アナライザー・プログラムまたは URIMAP 定義で提供できます。これらからユーザー ID を提供しない場合は、CICS のデフォルト・ユーザー ID が使用されます。

BACKLOG(10|*backlog*)
CICS 処理を待機する TCP/IP のキューに入れることのできるインバウンド TCP/IP 接続オープン要求の最大数を 1 から 32767 の範囲で指定します。この最大数に達すると、TCP/IP は追加の要求を拒否します。 BACKLOG の値がゼロの場合、TCP/IP の属性 SOMAXCONN の値が使用されます。 BACKLOG の値が SOMAXCONN より大きい場合は、SOMAXCONN の値が使用されます。 IPIC 接続の場合は、この TCPIPSERVICE を使用して確立できる同時接続要求の最大数を、この値に設定します。

CICS 処理で TCP/IP のキューに入れることが可能なインバウンド TCP/IP 接続要求の最大数を指定します。この最大数に達すると、TCP/IP は追加の接続要求を拒否します。この値は、この TCPIPSERVICE を使用して確立できる同時接続要求の最大数に設定します。 BACKLOG の値が SOMAXCONN の TCP/IP 構成値よりも大きい場合、TCP/IP は SOMAXCONN 属性で指定された値を使用します。値がゼロの場合は、着信接続要求が使用不可になります。

HTTP 接続のパフォーマンス・チューニングが有効にされている場合、CICS が最大容量に達すると、すべてのインバウンド HTTP 接続のオープン要求は、CICS 外部の、TCPIPSERVICE のリスニング接続のバックログ・キューに入ります。このキューがいっぱいになると接続要求は拒否されるため、BACKLOG の値は十分に大きい値にしてください。リスニング接続に使用しているバックログ値を表示するには、CICS TCP/IP サービス: リソース統計を使用するか、NETSTAT ALL コマンドを使用して、ローカル・ホストの状況に関する情報

(リスニング接続のバックログに関する情報を含む) を入手します。
接続バランシングが使用されている場合、バックログ・キュー項目数は、最も適したリスナーを判別する処理に含まれるため、ポートを共用する CICS 領域上のすべての TCPIPSERVICE 全体で BACKLOG 属性を考慮してください。

CERTIFICATE(*label*)
接続が獲得される際の SSL ハンドシェーク中にサーバー証明書として使用される X.509 証明書のラベルを指定します。この属性を省略した場合は、CICS 領域ユーザー ID の鍵リングに定義されているデフォルトの証明書が使用されます。

証明書ラベルの長さは最大で 32 バイトです。

証明書は、外部セキュリティー・マネージャーのデータベース内の鍵リングに保管する必要があります。詳しくは、鍵リングの手動構築 を参照してください。

この属性を指定する場合は、SSL(YES) または SSL(CLIENTAUTH) も指定する必要があります。

CIPHERS(*value*)

CIPHERS 属性は、以下の 2 とおりの方法のいずれかで指定できます。

- 最大 28 個の 2 桁の暗号スイート・コードを示すリストとして解釈される 56 桁までの 16 進数字で構成されるストリング。
- SSL 暗号スイート仕様ファイルの名前。この z/OS UNIX ファイルは、**USSCONFIG** システム初期設定パラメーターによって指定されるディレクトリーの security/ciphers サブディレクトリー内にあります。例えば、**USSCONFIG** が /var/cicsts/dfhconfig に設定され、**CIPHERS** が strongciphers.xml に設定されている場合、完全修飾ファイル名は /var/cicsts/dfhconfig/security/ciphers/strongciphers.xml です。詳しくは、SSL 暗号スイート仕様ファイルの作成を参照してください。

CEDA トランザクションを使用してリソースを定義すると、CICS は、デフォルトの許容コード・リストに従ってその属性を自動的に初期化します。この属性を CICS で初期化できるようにするためには、CEDA を実行する CICS 領域で KEYRING システム初期設定パラメーターを指定する必要があります。
KEYRING を設定しなかった場合、CICS はその属性を初期化しません。デフォルトのコード・リストは、35363738392F303132330A1613100D15120F0C です。ただし、システム初期設定パラメーターで **NISTSP800131A=CHECK** が設定されている場合は、35363738392F303132330A1613100D です。

暗号コードは、再配列することも、初期リストから除去することもできます。ただし、指定した暗号化レベルのデフォルト・リストに含まれていない暗号コードを追加することはできません。値をコードのデフォルト・リストにリセットするには、すべての暗号スイート・コードを削除します。それにより、フィールドには自動的にデフォルト・リストが取り込まれます。

詳しくは、暗号スイートおよび暗号スイート仕様ファイルを参照してください。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。

CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

DNSGROUP(*dnsgroup*)
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

GRPCRITICAL({**NO**|**YES**})
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

HOST({**ANY**|**DEFAULT**|*hostname*})
CICS で着信接続を listen する 116 文字の IPv4 または IPv6 アドレス、またはホスト名を指定します。新規リソースを定義する場合は、IPADDRESS の代わりに HOST を使用します。 HOST は常に IPADDRESS より優先されるため、HOST と IPADDRESS の両方を指定しないでください。 IPADDRESS は、IPv4 機能を指定する既存のプログラムでサポートされます。

使用可能な値は以下のとおりです。

ANY ANY オプションには、IPADDRESS の ANY オプションおよび INADDR_ANY オプションと同じ機能があります。ANY オプションは、CICS がホスト・システムの TCP/IP に認識されているすべてのアドレスで listen することを指定します。ホストには複数の IP アドレスを定義できます。ANY を指定すると、CICS サーバー間で TCPIPSERVICE 定義を共用することも可能になります。ANY を指定した場合、CICS は、ポートが定義されているすべてのスタックでポートへのバインドを試行します。加えて、複数の CICS 領域をポートにバインドすることを希望する場合は、ポートが定義されているすべてのスタックで SHAREPORT オプションを指定する必要があります。これを行わないと、SHAREPORT オプションが指定されていないスタックのそのポート番号に 1 つしか CICS 領域をバインドできません。各スタックにバインドしようとする、他の領域による後続の試行は失敗し、CICS はポートが使用中であることを示すメッセージを発行します。

二重モード (IPv4 および IPv6) 環境で ANY オプションを指定した場合、CICS は、最新の IPv4 または IPv6 アドレスの再利用を試行します。これが初めての接続で、CICS がアドレスを取得できない場合は、0.0.0.0 が返され、親和性は割り当てられません。

DEFAULT

DEFAULT オプションは、複数スタック CINET 環境でデフォルトとして定義されている TCP/IP スタックに親和性を割り当てます。

DEFAULT オプションが二重モード (IPv4 および IPv6) 環境で使用される場合には、DEFAULT オプションは IPv4 環境に適用されるため、親和性は IPv4 環境に割り当てられます。

非 CINET 環境で DEFAULT を使用する場合、またはデフォルトの TCP/IP スタックが存在しない場合は、例外トレースが作成され、`0.0.0.0` が返されます。アフィニティーは割り当てられません。

重要: デュアル・モード (IPv4 および IPv6) 環境で作業している場合、HOST(DEFAULT) を指定すると、すべてのトラフィックが強制的に IPv4 ネットワーク接続を介して渡されます。

hostname

hostname には、キャラクター・ホスト名、IPv4 アドレス、または IPv6 アドレスを指定できます。

アドレスには、ドメイン・ネーム・サーバーで検索可能なキャラクター名を指定できます。ホスト名は大文字、小文字、または大/小文字混合で入力することができます。ただし、IP アドレスではなくホスト名を指定する場合、そのホスト名は TCPIPSERVICE 定義で小文字に変換されます。

ドメイン・ネーム・サーバーでアドレスのリストを保持している場合は、キャラクター・ホスト名を使用しないでください。これは、*hostname* はリストの最初の IP アドレスに対してのみ解決するためです。つまり、サーバーは、このホスト名のリストにある IP アドレスでは listen しません。ドメイン・ネーム・サーバーのリストにある特定の IP アドレスが必要な場合は、*hostname* に IP アドレスを明示的に定義します。

IPv6 アドレス (または IPv6 アドレスに解決されるホスト名) を指定する場合は、ユーザーが二重モード (IPv4 および IPv6) 環境で操作しており、通信先のクライアントおよびサーバーも二重モード (IPv4 および IPv6) 環境で稼働していることを確認してください。 IPv6 についての詳細は、IPv6 と CICS についてを参照してください。

IPv4 アドレスと IPv6 アドレスは、複数のフォーマットで指定することができます。アドレス・フォーマットについて詳しくは、IP アドレスを参照してください。

IPADDRESS({**ANY**|**INADDR_ANY**|**DEFAULT**|*ipaddress*})

この TCPIPSERVICE が着信接続で listen する小数点付き 10 進数の IPv4 アドレスを指定します。これは、nnn.nnn.nnn.nnn (nnn は 0 から 255) の形式でなければなりません。HOST 属性を使用して IPADDRESS と同じ情報を指定できますが、HOST は、IPv6 形式のアドレスと文字のホスト名もサポートします。 IPv6 接続を使用する場合、定義には、IPADDRESS 属性ではなく HOST 属性を使用する必要があります。 HOST は、常に IPADDRESS より優先されます。IPADDRESS 属性は、複数の方法で HOST と対話します。

- HOST を指定した場合、常に HOST の内容に応じた値で IPADDRESS が上書きされます。
  - HOST に IPv4 アドレス、ANY、または DEFAULT を指定した場合、IPADDRESS は HOST の内容で上書きされます。
  - HOST で IPv6 アドレスまたはキャラクター・ホスト名を指定すると、IPADDRESS にブランクが上書きされます。
- HOST と IPADDRESS の両方を指定した場合、HOST の値が常に使用されます。
  - HOST に IPv4 アドレス、ANY、または DEFAULT が含まれている場合、IPADDRESS が HOST の内容に入力されます。
  - HOST に IPv6 アドレスが含まれている場合、IPADDRESS にブランクが上書きされます。
- IPADDRESS (HOST を除く) を指定した場合、HOST には IPADDRESS の内容が入力されます。

IP アドレスに `0.0.0.0` を指定し、HOST オプションをブランクに設定した場合、警告が発行され、値 ANY が使用されます。

IPADDRESS は、既存の IPv4 機能のためにのみサポートされています。新規リソースには、HOST オプションを使用してください。

可能な値は以下のとおりです。

ANY または **INADDR_ANY**

TCPIPSERVICE は、ホスト・システムの TCP/IP に認識されているすべてのアドレスを listen します。 1 つのホストに対して、複数の IP アドレスを定義することができます。また、ANY または INADDR_ANY を指定すると、CICS サーバー間で TCPIPSERVICE 定義を共用することも可能になります。

ANY または INADDR_ANY を指定した場合、CICS は、ポートが定義されているすべてのスタックでポートへのバインドを試行します。加えて、複数の CICS 領域をポートにバインドすることを希望する場合は、ポートが定義されているすべてのスタックで SHAREPORT オプションを指定する必要があります。そうしない場合、SHAREPORT オプションが含まれていないそれらのスタックでは、1 つの CICS 領域のみをポート番号にバインドできます。各スタックにバインドしようとする、他の領域による後続の試行は失敗し、CICS はポートが使用中であることを示すメッセージを発行します。SHAREPORT オプションについては、z/OS Communications Server IP 構成解説書 を参照してください。

DEFAULT

複数のスタック CINET 環境でデフォルトに定義されている TCP/IP スタックに、アフィニティーを割り当てます。非 CINET 環境で DEFAULT を使用する場合、またはデフォルトの TCP/IP スタックが存在しない場合は、例外トレースが作成され、アフィニティーは割り当てられません。

ipaddress

TCPIPSERVICE は、この特定のアドレスでの接続を受け入れます。指

定したアドレスがホスト・システム上の TCP/IP で認識されない場合は、TCPIPSERVICE は開きません。特定のアドレスをここに入力した場合、他の領域で実行中の CICS サーバーではこの定義が無効になることがあります。また、これらのサーバーと定義を共用できない場合があります。

MAXDATALEN({**32**|*number*})

HTTP プロトコルまたは USER プロトコルにおいて、CICS が HTTP サーバーとして受け取ることが可能なデータの最大長を指定します (キロバイト単位)。デフォルト値は 32 KB です。最小は 3 KB で、最大は 524,288 KB です。MAXDATALEN は、大量データの送信など、サービス妨害アタックに対する保護のために使用します。

MAXPERSIST({**NO**|*number*})

CICS 領域が常にこのポートに対して許可する Web クライアントからの持続接続の最大数を指定します。この設定は、HTTP プロトコルにのみ適用されます。

- デフォルト値 NO は、持続接続の数に限度がないことを意味します。
- CICS 領域が持続接続によって過負荷になる恐れがある場合、その CICS 領域が同時に処理可能な持続接続の数に基づいて、適切な値 (理論上の最大値は 65535 まで) を指定できます。この制限に達して、さらに Web クライアントがこのポートに接続してくる場合、CICS は、新たに接続してくるクライアントに対して、各応答を受け取った後に接続を閉じるよう要求します。この新たなクライアントが再接続するときに、ポートを共用しかつ制限に達していない別の CICS 領域に接続する場合、クライアントは代わりにそこで持続接続を維持します。通常 HTTP/1.1 サーバーは持続接続を許可している必要があるので、Web クライアントが長い間持続接続するためにパフォーマンス上の問題を抱える CICS 領域にのみこのオプションを設定します。
- このオプションにゼロの値を指定した場合、CICS 領域は持続接続を許可せず、すべての Web クライアントが各応答を受け取った後に接続を閉じるよう要求します。MAXPERSIST のゼロ設定値は HTTP/1.1 の仕様に対応していません。そのため、例えばテスト環境などその時点で外部要求を処理しない CICS 領域で特別な要件がある場合にのみこの設定値を使用します。

PORTNUMBER(*port*)

CICS が着信クライアント要求を listen する 10 進数のポート番号を 1 から 65535 の範囲で指定します。

1 から 1023 まではウェルノウン・ポートです。ポート番号が通常どおり割り当てられるサービスの場合は、ウェルノウン・ポート番号のみを使用することをお勧めします。CICS でサポートされるサービスのウェルノウン・ポートは以下のとおりです。

80 HTTP (SSL 非対応)
443 HTTP (SSL 対応)
1435 ECI (登録済みポート番号)

同じ MVS イメージ上にある、ウェルノウン・ポートを使用する可能性があるその他のサーバーとの競合を解決するように注意する必要があります。

MVS イメージ内の CICS システム間で共用するすべてのポートで、ポート共用を有効にする必要があります。詳細については、z/OS Communications Server IP 構成解説書を参照してください。

PRIVACY
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。詳しくは、Obsolete attributesを参照してください。

PROTOCOL({**HTTP**|**IIOP**|**ECI**|**IPIC**|**USER**})
TCP/IP ポートで使用するアプリケーション・レベル・プロトコルを指定します。

HTTP HTTP プロトコルを使用します。HTTP プロトコルは、CICS Web サポートによって処理されます。CICS はこのプロトコルを使用して、送受信されたメッセージの基本的な受け入れ検査を実行します。このプロトコルは、ウェルノウン・ポート 80 (SSL 非対応の HTTP で使用) および 443 (SSL 対応の HTTP で使用) では必須です。 HTTP 要求は、直接接続ユーザー・トランザクションによって処理され、TCPIPSERVICE の TRANSACTION 属性で定義された Web 接続タスクをバイパスする対象となる場合があります。詳しくは、直接接続されたユーザー・トランザクションにより HTTP 要求が処理されるを参照してください。

IIOP TCPIPSERVICE 定義の IIOP 属性は廃止されましたが、互換性を確保するために保持されています。

ECI CICS ECI プロトコルが使用されます。

IPIC IPIC プロトコルを使用します。IP 相互接続性 (IPIC) 接続に使用される TCPIPSERVICE リソースには IPIC を指定してください。

USER ユーザー定義プロトコルを使用します。メッセージは非 HTTP メッセージとして処理されます。メッセージは非 HTTP のフラグ付きで、TCPIPSERVICE リソースのアナライザー・プログラムにそのままの形で渡されます。CICS Web サポート機能は要求の処理に使用されますが、このプロトコルを使用して送受信されたメッセージに対して受け入れ検査は実行されません。HTTP プロトコルを使用した要求に対して CICS が実行する基本的な受け入れ検査が行われないように、すべての非 HTTP 要求の処理は、USER プロトコルで実行しなければなりません。 HTTP メッセージを USER プロトコルによって処理する場合は、その妥当性の検査をユーザーが行う必要があります。

REALM(*string*)
HTTP 基本認証で使用する領域を指定します。この属性は、HTTP プロトコルにのみ指定することができます。

レルムは、WWW-Authenticate ヘッダー内で CICS によって提供されるもので、ユーザーは基本認証のプロセスの中でこれを見ることができます。これによって、要求された認証情報 (ユーザー ID とパスワードまたはパスワード・フレーズ) が適用されるリソース・セットが識別されます。

レルムが指定されなかった場合に CICS が使用するデフォルトは `CICS application aaaaaaaa` です。*aaaaaaaa* は CICS 領域の APPLID です。

レルムは最大 56 文字で指定することができ、埋め込みブランクを含めることができます。大/小文字混合で指定され、大/小文字は保持されます。開始および終了二重引用符は、CICS が WWW-Authenticate ヘッダーをアセンブルするときに提供するため、指定しないでください。

許容文字: A-Z a-z 0-9 $ @ # . - _ % & ? ! : | ' = ¬ + * , ; < > ( ) スペース文字も使用できます。括弧を使用する場合は、必ず左括弧と右括弧を対にして使用する必要があります。

SOCKETCLOSE({NO|*hhmmss*})

ソケットを閉じるまで CICS が待機するかどうか、および待機する場合の待機時間を指定します。 SOCKETCLOSE 属性は、接続が確立された後に発行される最初の受信要求には適用されません。 ECI プロトコルおよび USER プロトコルの場合、最初の受信要求では、CICS はソケットを閉じるまで 30 秒間データを待ちます。 HTTP プロトコルの場合、最初の受信要求では、CICS は TCPIPSERVICE で指定されたトランザクションに関連付けられた DTIMEOUT 値の間待機します。この DTIMEOUT 値がゼロの場合、CICS は 30 秒間待機します。

この場合の待機時間は、そのソケットに着信データの受信要求が最初に入ったときから測定されます。

NO クライアントまたは CICS 内のユーザー・アプリケーション・プログラムによって閉じられるまで、ソケットは開いたままです。

hhmmss

最初の着信データの受信要求の時間から CICS がソケットをタイムアウトにするまでの間隔 (HHMMSS 形式)。クライアントの応答性およびネットワークの信頼性に適した値を選択してください。 000000 を指定した場合、2 回目以降の受信要求で使用できるデータがなければ、ソケットは即時に閉じられます。

HTTP プロトコルを使用する CICS Web サポートにこの TCPIPSERVICE リソースを使用している場合、SOCKETCLOSE をゼロに設定することは、後続のデータが待機していない限り、Web クライアントからデータを受信した直後に CICS が接続をクローズすることを意味します。この設定は、持続接続を維持できないことを意味しており、HTTP/1.1 仕様に準拠していません。 HTTP プロトコルを使用する SOCKETCLOSE のゼロ設定値は、例えばテスト環境などその時点で外部要求を処理しない CICS 領域で特別な要件がある場合にのみ使用します。

HTTP プロトコルを使用する CICS Web サポートにこの TCPIPSERVICE リソースを使用する場合、TCPIPSERVICE の SOCKETCLOSE 値は、**SET TCPIPSERVICE OPENSTATUS(CLOSED)** コマンドが発行された場合に CICS が HTTP 持続接続を閉じるまでの時間に影響します。 CICS は、以下のように HTTP 持続接続を閉じようと試みます。

- HTTP 持続接続の要求がある場合、CICS は次の要求の処理を許可します。処理が完了すると、CICS はクライアントに close ヘッダーを送信してから、HTTP 持続接続をクローズします。

- HTTP 持続接続の要求がない場合、CICS は、30 秒以内、またはこの値が 30 秒未満である場合は、TCPIPSERVICE の SOCKETCLOSE 属性で指定された時間内に、HTTP 持続接続をクローズします。

PROTOCOL(ECI) または PROTOCOL(IPIC) を指定する場合は、SOCKETCLOSE(NO) を指定する必要があります。

PROTOCOL(USER) を指定した場合、持続セッションはサポートされず、SOCKETCLOSE(000000) を指定する必要があります。

TCPIPSERVICE リソースのインストール後は、CEMT を使用してこの値を変更することはできません。変更する場合は、TCPIPSERVICE リソースをサービス休止に設定してから、変更された定義を使用して TCPIPSERVICE リソースを再インストールする必要があります。

SPECIFTCPS
このパラメーターでは、高可用性クラスターの一部として汎用 TCPIPSERVICE を定義するときに、特定の TCPIPSERVICE の 8 文字の名前を指定します。このパラメーターが有効なのは、プロトコルが IPIC で、属性 PORT と属性 HOSTNAME または IPADDRESS によって定義される IP エンドポイントが CICS 領域の HA クラスターの汎用エンドポイントである場合のみです。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているとき以外は、小文字はすべて大文字に変換されます。

SSL({NO|YES|CLIENTAUTH|ATTLSAWARE})
TCP/IP サービスで、暗号化と認証に Secure Sockets Layer (SSL) を使用するかどうかを指定します。この属性は、HTTP、USER、および IPIC プロトコルの場合に指定できますが、ECI プロトコルの場合は指定できません。注: SSL を使用する場合は、MAXSSLTCB を、CICS 領域によって SSL を使用する IPIC 接続の数の 2 倍以上の値に設定してください。

NO SSL を使用しません。接続を獲得する際にセキュリティー検査は行われません。アウトバウンド・メッセージに 暗号化は適用されません。SSL(NO) 要求および PROTOCOL(HTTP) HTTP 要求は、直接接続ユーザー・トランザクションによって処理され、Web 接続タスクをバイパスする対象になる場合があります。詳しくは、直接接続されたユーザー・トランザクションにより HTTP 要求が処理されるを参照してください。

YES SSL セッションが使用されます。CICS は、クライアントにサーバー証明書を送信します。このポートに到着するすべてのメッセージに対して SSL 暗号化解除の処理を適用します。インバウンド・メッセージに適用されている暗号化のレベルは、CIPHERS 属性の値から知ることができます。

CLIENTAUTH
SSL セッションが使用されます。CICS は、クライアントにサーバー証明書を送信します。CICS は、接続が獲得されるときの SSL ハンドシェークの際に、パートナー・システムからクライアント証明書を受け取ることを予期します。

ATTLSAWARE

CICS は、クライアント接続を照会して、AT-TLS がアクティブであるかどうかを判別します。 CICS は、クライアント証明書を TCP/IP から取得します (パートナーから提供されている場合)。 ATTLSAWARE は、PROTOCOL(HTTP) TCPIPSERVICES と一緒に使用する必要があります。 SSL(ATTLSAWARE) 要求および PROTOCOL(HTTP) HTTP 要求は、直接接続ユーザー・トランザクションによって処理され、Web 接続タスクをバイパスする対象になる場合があります。詳しくは、直接接続されたユーザー・トランザクションにより HTTP 要求が処理されるを参照してください。

STATUS({**OPEN**|**CLOSED**})

インストール後のサービスの初期状況を示します。インストール後に CICS がこのサービス用に listen を開始する場合は、これを OPEN に設定します。インストール後に CICS がこのサービスのために listen を行わない場合は、これを CLOSE に設定します。

この属性は、CICS バンドルによって動的に生成される TCPIPSERVICE リソースでは無視されます。 TCPIPSERVICE リソースの初期状況は、そのリソースを定義するバンドルの初期状況から導出されます。

TCPIPSERVICE(*name*)

このサービスの名前を 8 文字で指定します。

許容文字:
A-Z 0-9 $ @ #
CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

TRANSACTION(*transaction*)

このサービスに対して受け取る新しい要求を処理するよう付加された、CICS トランザクションの 4 文字の ID を指定します。

- ECI over TCP/IP の TCPIPSERVICE リソースでは、CIEP (または、プログラム DFHIEP を実行する別のトランザクション) を指定します。
- HTTP の TCPIPSERVICE リソースでは、CWXN (または、プログラム DFHWBXN を実行する別のトランザクション) を指定します。HTTP 要求は、直接接続ユーザー・トランザクションで処理して Web 接続タスクをバイパスする対象になる場合があります。詳しくは、直接接続されたユーザー・トランザクションにより HTTP 要求が処理されるを参照してください。
- IPIC TCPIPSERVICE リソースでは、CISS (または、プログラム DFHISCOP を実行する別のトランザクション) を指定します。
- USER の TCPIPSERVICE リソースでは、CWXU (または、プログラム DFHWBXN を実行する別のトランザクション) を指定します。

TSQPREFIX(*prefix*)

この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。詳しくは、Obsolete attributesを参照してください。

URM({**NO**|*program_name*})

このサービスで開始されるユーザー置き換え可能プログラムの名前を指定します。

NO この TCPIPSERVICE リソースでは、自動インストールは許可されていません。これは、PROTOCOL(IPIC) にのみ適用可能です。

program_name

URM が必須属性であるようなプロトコルでは、デフォルトのプログラム名は PROTOCOL 属性の値によって決まります。

- HTTP プロトコルでは、この TCPIPSERVICE リソースと関連付けられているアナライザー・プログラムの名前を指定します。デフォルトは、CICS 提供のアナライザー・プログラム DFHWBAAX です。DFHWBAAX は、ポート上のすべての要求 (例えば Web サービス要求) を URIMAP 定義で処理する必要がある場合に、基本的なエラー処理を提供します。URIMAP 定義で処理されない要求のサポートを提供する必要がある場合は、TCPIPSERVICE リソースで指定するアナライザー・プログラムを、CICS 提供のサンプル・アナライザー・プログラム DFHWBADX にするか、あるいはユーザーが独自にカスタマイズしたアナライザー・プログラムにしてください。アナライザー・プログラムについて詳しくは、「システム・プログラムの開発」の「アナライザー・プログラム」 を参照してください。
- IPIC プロトコルの場合は、必要に応じて IPIC 接続用の自動インストール・ユーザー・プログラムの名前を指定します。PROTOCOL(IPIC) では、この属性が指定されていない場合、CICS は CICS 提供のデフォルトの自動インストール・ユーザー・プログラム、DFHISAIP を使用します。
- USER プロトコルでは、この TCPIPSERVICE 定義と関連付けられるアナライザー・プログラムの名前を指定します。アナライザー・プログラムが存在している必要があります。このプロトコルでの要求は、すべてアナライザー・プログラムによって処理されます。 CICS 提供のサンプル・アナライザー・プログラム DFHWBADX をお勧めします。アナライザー・プログラムについて詳しくは、「システム・プログラムの開発」の「アナライザー・プログラム」 を参照してください。

TCPIPSERVICE: 相互関連属性

指定できる属性と値は、すでに指定したその他の属性によって決まる場合があります。

TCPIPSERVICE資源の場合：

- AUTHENTICATE属性に対して指定できる値は、 PROTOCOL属性の値によって決まります。

AUTHENTICATE	**PROTOCOL(ECI)** または **PROTOCOL(IPIC)**	**PROTOCOL(HTTP)** または **PROTOCOL(USER)**
NO	無効	有効
BASIC	無効	有効

AUTHENTICATE	PROTOCOL(ECI) または PROTOCOL(IPIC)	PROTOCOL(HTTP) または PROTOCOL(USER)
CERTIFICATE	無効	有効
AUTOREGISTER	無効	有効
AUTOMATIC	無効	有効

- PROTOCOL(HTTP) または PROTOCOL(USER) を指定する場合、ATTACHSEC はブランクにする必要があります。 ATTACHSEC属性を指定できるのは、PROTOCOL(ECI)を 指定する場合だけです。
- PROTOCOL(HTTP) を指定する場合、URM のデフォルトは、CICS 提供のデフォルトのアナライザー・プログラム DFHWBAAX になります。
- PROTOCOL(IPIC) を指定する場合、URM のデフォルトは、CICS 提供の IP 接続用のデフォルトの自動インストール・プログラム DFHISAIP になります。
- AUTHENTICATE(CERTIFICATE) または AUTHENTICATE(AUTOREGISTER) を指定する場合、SSL(CLIENTAUTH) または SSL(ATTLSAWARE) を指定する必要があります。
- PORTNUMBER 属性で予約済みポート番号を指定すると、CICS は、他の属性の値を以下のように設定します。

PORT	PROTOCOL	TRANSACTION	SSL
80	HTTP	CWXN	NO
443	HTTP	CWXN	YES
1435	ECI	CIEP	NO

- PROTOCOL属性を指定してTRANSACTION属性を指定しないと、 CICSによってTRANSACTION属性が設定されます。同様に、 TRANSACTION属性で次のいずれかの値を指定し、 PROTOCOL属性を指定しないと、CICSによってPROTOCOL属性が設定されます。

PROTOCOL	TRANSACTION
ECI	CIEP
HTTP	CWXN
IPIC	CISS
USER	CWXU

- TRANSACTION 属性に対して CICS が提供する値を変更することができます。PROTOCOL 属性に指定した値によっては、一部の値が許可されない場合があります。

PROTOCOL	TRANSACTION (CWXN)	TRANSACTION (CIEP)	TRANSACTION (CWXU)	TRANSACTION (CISS)
HTTP	デフォルト	無効	無効	無効
USER	無効	無効	デフォルト	無効
ECI	無効	デフォルト	無効	無効
IPIC	無効	無効	無効	デフォルト

- PROTOCOL(ECI) または PROTOCOL(IPIC) を指定する場合は、SOCKETCLOSE(NO) を指定する必要があります。

TDQUEUE リソース

TDQUEUE 定義は、一時データ・キューの属性を定義します。

以下の一時データ・リソースは、RDO を使用して管理することができます。
- 区画内
- 区画外
- 間接
- Remote (リモート)

区画内定義には、リカバリー特性、トリガー・レベル、関連トランザクション、機能、およびユーザー ID に関する情報を提供する属性が含まれます。

区画外定義には、関連 QSAM データ・セット、および使用されるバッファー数に関する情報が含まれます。

間接定義は、基礎となるキュー名を示します。

リモート定義には、リモート・システムの名前、およびそのリモート・システムでキューを認識するための名前が含まれます。

アクティブ CICS システムで一時データ・キューを使用できるようにするには、事前にその定義を稼働中のシステムにインストールする必要があります。 CICS は、その定義を使用してキューに関連付けられたデータ・セットにアクセスし、キュー上の読み取り操作および書き込み操作の数を記録します。

リモート一時データ・キューは、次のいずれかの方法で CEDAトランザクションを使用して定義できます。
- キューTYPEが省略され、データが定義のREMOTE ATTRIBUTES セクションだけに入力される場合は、リモート定義が 作成されます。
- TYPE に INTRA または EXTRA が指定されており、REMOTE ATTRIBUTES セクションが完了すると、ローカルおよびリモートの両方のリソースが同時に確立されます。

  二重目的一時データ・リソース定義を定義する例については、 189 ページの図 3 を参照してください。

TDQUEUE リソース定義のインストール、置換、または使用不可化については、TDQUEUE リソース定義のインストール、TDQUEUE リソース定義の置き換え、およびTDQUEUE リソース定義の無効化を参照してください。BAS の作業については、BAS 一時データ・キュー・リソース定義の作業を参照してください。

一時データの二重目的リソース定義

一時データ・キューに対してTYPE=REMOTEを指定することはできません。

代わりに、二重目的リソース定義について検討することができます (Shared resources for intercommunication を参照)。一時データ定義では、二重の目的を持つ資源定義を使用できます。図 3 は、一時データ・リソースの二重目的リソース定義の例を示しています。

図 3 に示された定義を CICQ という名前のシステムにインストールした場合、その定義は、ローカル区画内キューになります (REMOTESYSTEM の値が CICQ)。

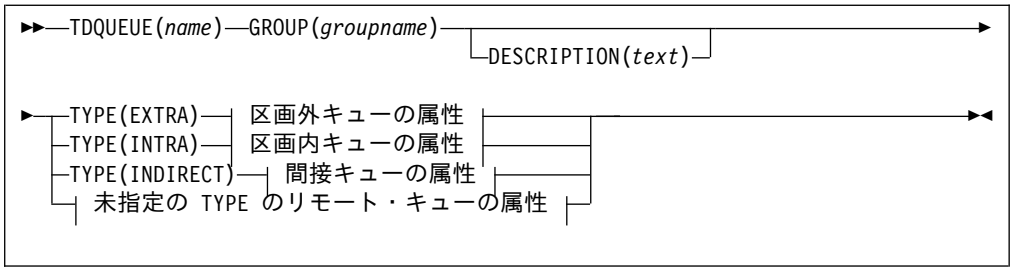
図 3 に示された定義を CICQ 以外のシステムにインストールした場合、その定義はリモート・キューになります。

```
 TDqueue        : TDQ1
 Group          : Example
 DEscription  ==>
 TYPE         ==> Intra                        Extra | INTra | INDirect
EXTRA PARTITION PARAMETERS
 DAtabuffers    :                              1-255
 DD 名:
 DSname         :
 Sysoutclass    :
 Erroroption    :                              Ignore | Skip
 Opentime       :                              Initial | Deferred
 REWind         :                              Leave | Reread
 TYPEFile       :                              Input | Output | Rdback
 RECORDSize     :                              0-32767
 BLOCKSize      :                              0-32767
 RECORDFormat   :                              Fixed | Variable
 BLOCKFormat    :                              Blocked | Unblocked
 Printcontrol   :                              A | M
 DIsposition  :                                Shr | Old |Mod
INTRA PARTITION PARAMETERS
 Atifacility  ==> Terminal          Terminal | File | System
 RECOVstatus  ==> Logical           No | Physical | Logical
 Facilityid   ==> FR1
 TRAnsid      ==>
 TRIggerlevel ==> 00001             0-32767
 Userid       ==>

 INDOUBT ATTRIBUTES
 WAIT         ==> Yes                          Yes|No
 WAITAction   ==> Reject                       Queue|Reject

 INDIRECT PARAMETERS
 Indirectname :

 REMOTE PARAMETERS
 REMOTEName   ==> FR1
 REMOTESystem ==> CICQ
 REMOTELength ==>                                    0-32767
```

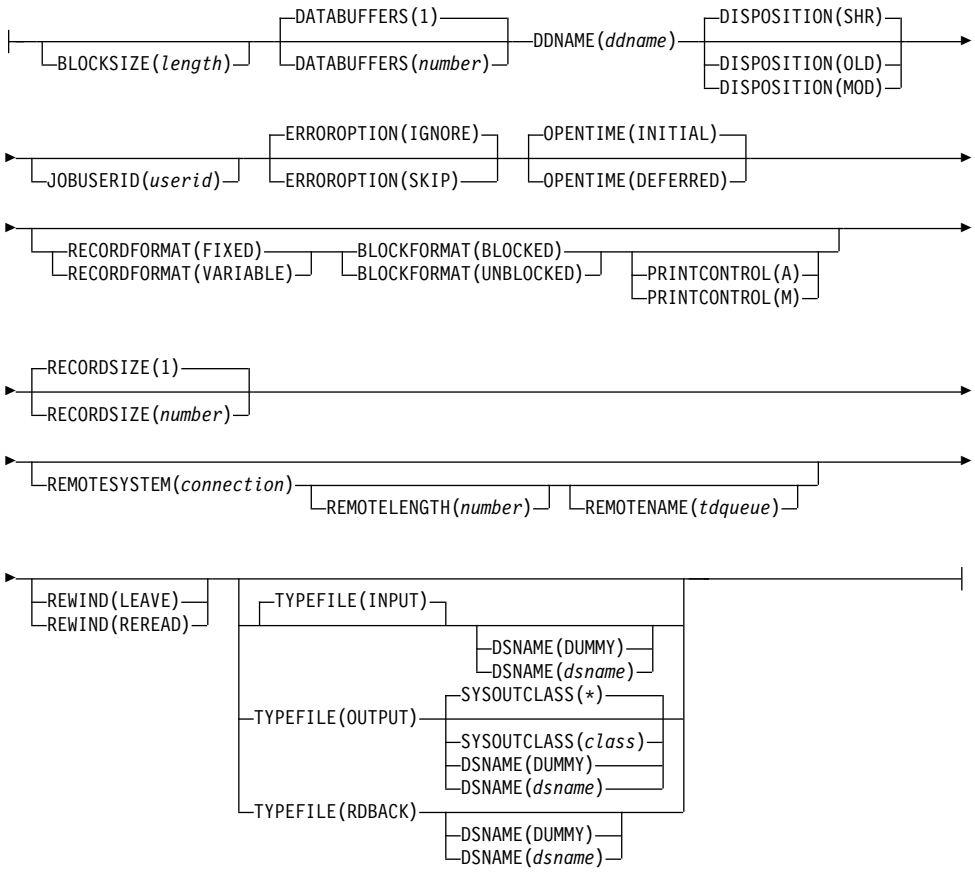
図 3. 一時データの二重目的リソース定義

TDQUEUE 属性

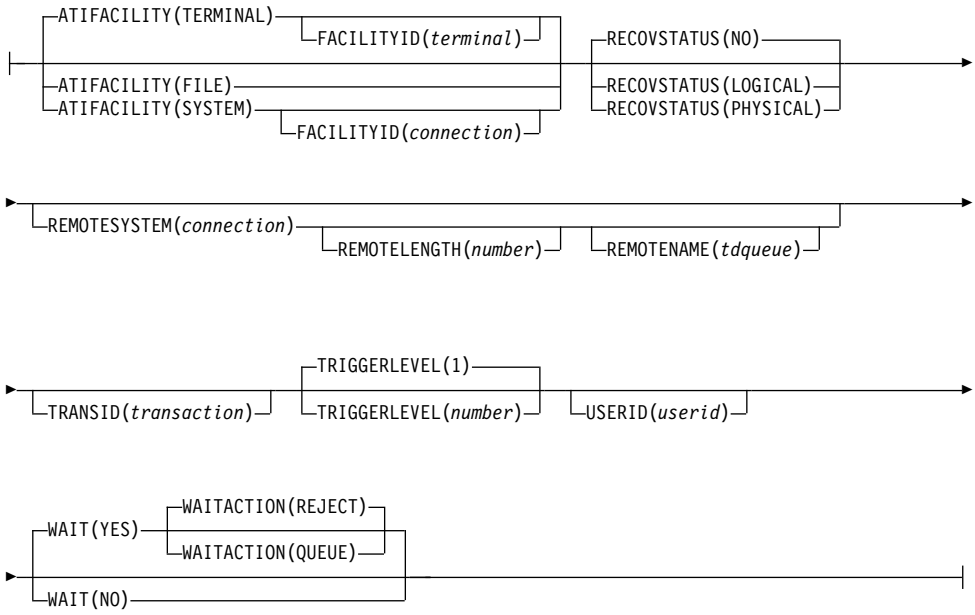
TDQUEUE リソースの構文および属性について説明します。


TDQUEUE(name)
GROUP(groupname)
DESCRIPTION(text)
TYPE(EXTRA)
TYPE(INTRA)
TYPE(INDIRECT)
区画外キューの属性
区画内キューの属性
間接キューの属性
未指定の TYPE のリモート・キューの属性

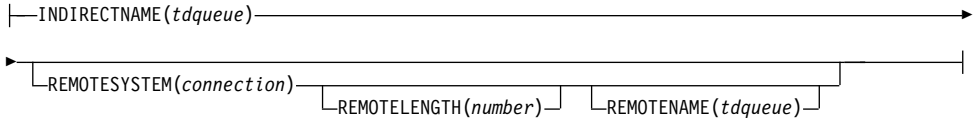
区画外キューの属性:


BLOCKSIZE(length)
DATABUFFERS(1)
DATABUFFERS(number)
DDNAME(ddname)
DISPOSITION(SHR)
DISPOSITION(OLD)
DISPOSITION(MOD)
JOBUSERID(userid)
ERROROPTION(IGNORE)
ERROROPTION(SKIP)
OPENTIME(INITIAL)
OPENTIME(DEFERRED)
RECORDFORMAT(FIXED)
RECORDFORMAT(VARIABLE)
BLOCKFORMAT(BLOCKED)
BLOCKFORMAT(UNBLOCKED)
PRINTCONTROL(A)
PRINTCONTROL(M)
RECORDSIZE(1)
RECORDSIZE(number)
REMOTESYSTEM(connection)
REMOTELENGTH(number)
REMOTENAME(tdqueue)
REWIND(LEAVE)
REWIND(REREAD)
TYPEFILE(INPUT)
DSNAME(DUMMY)
DSNAME(dsname)
TYPEFILE(OUTPUT)
SYSOUTCLASS(*)
SYSOUTCLASS(class)
DSNAME(DUMMY)
DSNAME(dsname)
TYPEFILE(RDBACK)
DSNAME(DUMMY)
DSNAME(dsname)

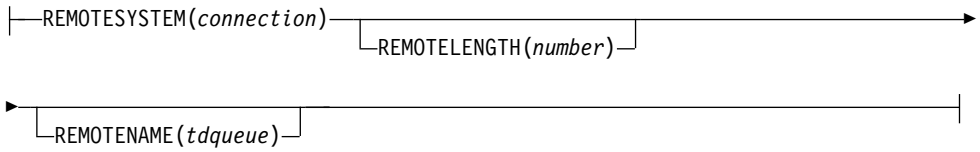
区画内キューの属性:



間接キューの **TDQUEUE** 属性:



未指定の **TYPE** のリモート・キューの **TDQUEUE** 属性:



ATIFACILITY({**TERMINAL**|**FILE**|**SYSTEM**}) **(**区画内キューのみ**)**
キューが表す宛先のタイプを指定します。

FILE 一時データ・キューは、特定の端末またはシステムに 関連しないデータ・レコードのファイルとして 使用されます。ATI は、端末が使用可能であることを必要としません。

SYSTEM
一時データ・キューは指定されたシステムIDに 関連付けられます。システムは、RDO CONNECTION 定義を使用してローカル CICS システムに定義される必要があります。

ATIFACILITY(SYSTEM)を指定すると、分散トランザクション処理 (DTP)セッションが開始されます。アプリケーション・プログラミングにおける DTP の考慮事項について詳しくは、分散トランザクション処理 (DTP) を参照してください。

TERMINAL
一時データ・キューは端末に関連付け られます。この端末は、CICS に定義されている必要があります。 TERMINAL を指定しない場合、これはデフォルト値である FACILITYID になります。 ATI を使用する場合は、TRANSID 属性および TRIGGERLEVEL 属性で指定された内容に従って、開始されたトランザクションが指定端末と関連付けられます。この端末はトランザクションの開始前に使用可能になっている必要があります。

BLOCKFORMAT({**BLOCKED**|**UNBLOCKED**|*blank*}) **(区画外キューのみ)**
データ・セットのブロック・フォーマットを指定します。デフォルトはありません。レコード形式 (RECORDFORMAT 属性) を未定義として指定した場合 (またはデフォルトが適用されるようにした場合)、BLOCKFORMAT 属性には何も指定できません。

blank このデータ・セットに対してブロック形式が定義されていない ことを示します。RECORDFORMAT をブランクにする場合は、このフィールドもブランクにします。

BLOCKED
ブロック化レコード形式。

UNBLOCKED
非ブロック化レコード形式。

JES内部読み取りプログラムへのインターフェースとして 使用される区分外キューには非ブロック化レコード形式を 指定することを強く推奨します。ブロック化レコード形式を使用する場合、ジョブは SYSOUT データ・セット内に保持され、以下のいずれかのアクションが実行されるまで、JES に直接送信されることはありません。

- JES /*EOF 制御ステートメントの後に 2 番目の /*EOF ステートメントを続ける
- アプリケーションによって同じキューに別のジョブが書き込まれる
- ジョブの書き込み後に明示的にキューを閉じる
- CICS を正常シャットダウンする

BLOCKSIZE({**length**}) **(区画外キューのみ)**
ブロックの長さ (バイト数) を 0 から 32767 の範囲で指定します。

指定できる最大値は、SYSOUTCLASS が (明示的に、あるいはデフォルトで) 指定されているか、および RECORDFORMAT が FIXED であるか VARIABLE であるかによって異なります。

- SYSOUTCLASS を指定した場合、RECORDSIZE の最大値は 8968 です。

- 可変長フォーマットのデータ・セット内の各ブロックは、ブロック記述子ワードと、その後に続く 1 つ以上の論理レコードから構成されます。したがって、RECORDFORMAT(VARIABLE) を指定した場合、BLOCKSIZE に指定する値には、ブロック記述子ワード用の 4 バイトと、可能性のある最大の論理レコード用のスペースが含まれている必要があります。

これらの制限の要約は、以下の表に示しています。

SYSOUTCLASS が指定されているか	はい	はい	いいえ	いいえ
RECORDFORMAT	VARIABLE	FIXED	VARIABLE	FIXED
RECORDSIZE の最大値	8968	8968	32763	32767
BLOCKSIZE の最大値	8972	8968	32767	32767

DATABUFFERS({**1**|*number*}) **(区画外キューのみ)**
提供されるバッファーの数 (最大 255) を指定します。

DDNAME(*ddname*)
始動 JCL で定義されたデータ・セットを参照できる 1 から 8 文字の値を指定します。この名前は、文字「DFH」から始めることはできません。この文字は、CICS が使用するために予約されています。ただし、この名前が標準宛先を示している場合は除きます。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

DISPOSITION({**SHR**|**OLD**|**MOD**}) **(区画外キューのみ)**
データ・セットの後処理を指定します。

MOD CICS は、まず、データ・セットが存在すると仮定します。既存の順次データ・セットでは、MOD は読み取り/書き込み機構がデータ・セット内の最終レコードの後に配置されるようにします。 read/writeメカニズムは、データ・セットが 出力のため開かれるたびに最終レコードの後に 置かれます。

CICS は、DD ステートメントのデータ・セットのボリューム情報がカタログにない場合や、直前のステップからのデータ・セットでボリューム情報が渡されなかった場合は、そのデータ・セットがこのジョブ・ステップで作成されると想定します。このように動的に割り振られるデータ・セットは、キューが閉じられると削除され、すべてのレコードが失われます。

新規データ・セットでは、MODはデータ・セットの先頭に read/write メカニズムを配置します。

OLD データ・セットはこのジョブ・ステップの前に存在していました。

SHR データ・セットはこのジョブ・ステップの前に存在しており、 他の並行ジョブはそれを読み取ることができます。

DSNAME({*dsname*|**DUMMY**}) **(区画外キューのみ)**
この区画外キューに書き込まれたレコードを保管するために使用する QSAM データ・セットの名前を指定します。

区画外一時データ・キューを開く要求を CICS が受信すると、データ・セット定義が作成されているかを確認するために始動 JCL が参照されます。見つからない場合は、 DSNAME属性で指定された44文字の名前を使用して、 必要なデータ・セットが動的に割り振られます。

このキューについて DSCNAME 値を DSNAME 値に事前割り振りする JCL がある場合、リソース定義内の DSNAME 値は、JCL の DSNAME 値でオーバーライドされます。 JCL割り振りは常に優先されます。

区分データ・セット(PDS)はDSNAME属性でサポート されません。区分外キュー・データ・セットに PDSメンバーを使用する場合は、JCLで明示的にコード化 します。このキューを照会しても、返される DSNAME 値でメンバー名は示されないので注意してください。

DUMMY
ダミー・データ・セット名。

name 物理データ・セットの44文字の名前。

重要: 区画外キュー・データ・セットにログ・ストリームを使用した場合、予測不能な結果が発生する可能性があります。

ERROROPTION({**IGNORE**|**SKIP**}) **(区画外キューのみ)**
入出力エラーが発生した場合に実行されるアクションを指定します。これは、以下のいずれかを指定することができます。

IGNORE
エラーの原因となったブロックが受け入れられます。

SKIP エラーの原因となったブロックがスキップされます。

FACILITYID(*terminal*|*connection*) **(区画内キューのみ)**
以下のいずれかを含む 4 文字のフィールドを指定します。

- ATIFACILITY(SYSTEM)を指定する区分内キューの システムID。
- ATIFACILITY(TERMINAL)が指定された 端末ID。

FACILITYID フィールドに何も指定しない場合、デフォルトでそれぞれのキューの名前になります。

ATIFACILITY(FILE) が指定された場合は、FACILITYID フィールドをブランクのままにする必要があります。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

INDIRECTNAME(*tdqueue*) **(間接キューのみ)**
この間接キューがデータを送信する一時データ・キューの名前を指定します。一時データ・キューは、区画内、区画外、リモート、あるいは間接のいずれかで、CICS に定義する必要があります。

JOBUSERID(*userid*) **(区画外キューのみ)**
JCL の JOB カードに USER パラメーターが指定されていない場合に、内部読み取りプログラムに実行依頼された JCL ジョブの実行に使用するユーザー ID を指定します。

JOBUSERID が定義されていない場合のデフォルトは、CICS 領域ユーザー ID です。

OPENTIME({**INITIAL**|**DEFERRED**}) **(区画外キューのみ)**
データ・セットの初期状況を指定します。初期状況は、以下のいずれかの値です。

DEFERRED
データ・セットは、CEMT INQUIRE|SET TDQUEUE コマンドを使用して開く指示を出すまで、 閉じられたままです。

INITIAL
データ・セットはインストール時に開かれます。ただし、DSNAME 属性が指定されず、データ・セット名が始動 JCL の DD ステートメントで指定されていない場合、CICS の始動時に一時データ・キューが JES に割り振られます。

PRINTCONTROL({**ASA**|**MACHINE**|*blank*}) **(区画外キューのみ)**
使用する制御文字を指定します。デフォルトはありません。

RECORDFORMAT をデフォルトのブランクにした場合、PRINTCONTROL フィールドには何も指定することができません。以下の文字を使用することができます。

ASA ASA制御文字。

blank 制御文字は使用されません。

MACHINE
マシン制御文字。

RECORDFORMAT({**FIXED**|**VARIABLE**|**blank**) **(区画外キューのみ)**
データ・セットのレコード形式を指定します。

blank RECORDFORMAT を指定しない場合 (つまり、ブランクのまま)、BLOCKFORMAT フィールドおよび PRINTCONTROL フィールドもブランクのままにする必要があります。リソース定義で RECORDFORMAT を指定しない場合、TD は、この属性を CICS 始動 JCL から、あるいはキューを開くときに QSAM データ・セット定義から取得しようと試みます。どちらのソースからもこの情報を導出できない 場合、オープン要求は失敗します。

FIXED

固定レコード。 RECORDFormat(Fixed) を指定する場合、ブロック形式も指定する必要があります。

VARIABLE

可変レコード。 RECORDFormat(Variable)を指定する場合は、 ブロック形式も指定する必要があります。

RECORDSIZE({**1**|*number*}) **(区画外キューおよびリモート・キューのみ)**

レコード長 (バイト数) を 0 から 32767 の範囲で指定します。

1

デフォルトのレコード長は1バイトです。

number

最大32767バイトのレコード長。

指定できる最大値は、SYSOUTCLASS が (明示的に、あるいはデフォルトで) 指定されているか、および RECORDFORMAT が FIXED であるか VARIABLE であるかによって異なります。

- SYSOUTCLASS を指定した場合、RECORDSIZE の最大値は 8968 です。
- V フォーマット・データ・セットの場合、各論理レコードは、データ・レコードが後に続くレコード記述子で構成されます。RECORDSIZE フィールドに入力される値には、レコード記述子ワード (または LLBB) 用の 4 バイトが含まれている必要があり、可能性のある最大のデータ・レコードを許容する値でなければなりません。

これらの制限の要約は、BLOCKSIZE 属性の説明に示されています。

RECOVSTATUS({**NO**|**PHYSICAL**|**LOGICAL**}) **(区画内キューのみ)**

CICS あるいはキューを処理しているトランザクションのいずれかに異常終了が発生した場合の、キューのリカバリー可能性属性を指定します。リカバリー可能性属性は次のとおりです。

LOGICAL

このキューは論理的にリカバリー可能です。アプリケーション・プログラムによるアクセスの追跡を保持するために自動ログ記録が実行されます。このキューにアクセスしたトランザクションが、異常終了時、あるいは、その後の緊急時再始動や動的トランザクション・バックアウト時に実行中であった場合、キューは、実行中の UOW によって変更される前の状態に復元されます。

このキューにアクセスすると、DELETEQ TD, WRITEQ TD または READQ TDコマンドを発行したタスクは、 一時データ・キューの入力端、出力端、または両端にエンキュー されます。アクセス先のデータの整合性を確実にするため、 エンキューは、タスクが終了するまで（またはUOWの終わりを示す 同期点要求を発行するまで）維持されます。これは、エンキューを長時間保持することができ、キューにアクセスするアプリケーション・プログラムが、そのキューに対して複数の UOW を実行した場合に、同期点要求の発行によって CICS に対する個別の UOW の定義が行われていないと、キューのロックアウトが発生する可能性があることを意味します。

さらに、論理的にリカバリー可能なキューに対して DELETEQ要求が発行されると、キューの入力端と出力端の 両方がエンキューされます。これにより、エンキュー・ ロックアウトの可能性が大きくなります。

注: CICS は、リカバリー可能ファイルの場合と同様に、論理的にリカバリー可能な (物理的にリカバリー可能なものは除く) TD キューに対するエンキュー保護機能を提供します。ただし、CICS は、論理的にリカバリー可能な各宛先を、2 つの別個のリカバリー可能リソース (書き込み用と読み取り用) と見なします。

ファイル・レコードの場合、レコードは単一資源として 扱われ、必要なロックは1つだけです。一方で、TD キューには、読み取り側と書き込み側の 2 つの「端」があり、これらは別個にエンキュー (ロック) することができます。この理由は、TD キューの読み取りと書き込みの両方を (同時に) 制御するために、CICS が 2 つのポインター (カーソル) を保持しており (読み取り用と書き込み用に 1 つずつ)、これらを競合トランザクションから保護する必要があるためです。

キュー・レコードは、1 つ以上の制御インターバル (CI) で保持されます。 CI上の最終レコードが読み取られるとすぐ、 各CIに解放のマークが付けられます。ただし、タスクの 終了または次のユーザー同期点の通過まで、解放は 行われません。

NO このキューはリカバリー不能です。 このキューへのアクセスを追跡するための自動ロギングは 実行されません。キュー・レコードは、1 つ以上の制御インターバル (CI) で保持されます。各 CI は、その CI 上の最後のレコードが読み取られると、すぐに解放されます。

PHYSICAL

このキューは物理的にリカバリー可能です。アプリケーション・プログラムによるアクセスの追跡を保持するために自動ログ記録が実行されます。緊急時再始動が実行された場合、このキューは CICS が終了した時点の状態にリカバリーされます。

キューの最後のアクションが READQ 要求であり、関連付けられた作業単位 (UOW) が変更をコミットしなかった場合、そのキューは、CICS の終了時の状況にリカバリーされません 。緊急再始動では、最後の 読み取り操作がバックアウトされ、実行されていないように 見えます。

キュー・レコードは、1 つ以上の制御インターバル (CI) で保持されます。各 CI は、その CI 上の最後のレコードが読み取られると、すぐに解放されます。

REMOTENAME(*tdqueue*) (リモート・キューのみ)

一時データ・キューがリモート・システム上にある場合に、キューが存在するシステムまたは領域がそのキューを認識するために使用する 4 文字の名前を指定します。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

REMOTELENGTH({**1**|*number*}) **(リモート・キューのみ)**
長さ (バイト数) を 1 から 32767 の範囲で指定します。

SYSOUTデータ・セットでは、REMOTELENGTHフィールドに 8968バイトより大きい値を入力することはできません (SYSOUTCLASS属性が指定されている場合）。

1　　長さは1バイトです。

number
　　最大32767バイトの長さ。

キューがTYPE=EXTRAで定義され、REMOTELENGTHの値が 指定されていない場合は、RECORDSIZE属性の値が インストール時に使用されます。

REMOTESYSTEM(接続)
リモート一時データ・キューが存在するシステムまたは領域の 4 文字の英数字名を指定します。入力する名前は、RDO CONNECTION で指定された名前、あるいは RDO IPCONN 定義で指定された名前の最初の 4 文字と同じでなければなりません。接続定義について詳しくは、 11 ページの『CONNECTION 属性』 または 77 ページの『IPCONN 属性』 を参照してください。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

一時データ・キュー定義がインストールされると、 REMOTESYSTEM属性に入力された名前がシステムIDと 比較されます。名前が異なる場合は、 システムまたは領域がリモートです。名前が 同じ場合は、TYPE属性で指定された値が使用 されます。TYPE属性がブランクである場合、 インストールは失敗します。

REWIND({**LEAVE**|**REREAD**}) **(区画外キューのみ)**
テープ・データ・セットの後処理を指定します。後処理は 次のいずれかです。

LEAVE
　　現行テープ位置はデータ・セットの論理終了に 置かれます。

REREAD
　　現行テープ位置はデータ・セットの論理開始に 置かれます。

SYSOUTCLASS({**A..Z**|**0..9**|*****|*blank*}) **(区画外キューのみ)**
区画外キューを物理データ・セットに割り振る代わりに、それをシステム出力データ・セット (SYSOUT と呼ばれる) に割り振ることができます。

SYSOUTCLASS属性を使用して、SYSOUTデータ・セットの クラスを指定します。

A..Z|**0..9**
　　CICS ジョブを実行する MVS システム上でセットアップされた出力クラスを表す単一の英字または数字。

許容文字: A-Z 0-9 CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

* これはデフォルトのクラスです。 DSNAME属性をブランクにし、TypefileフィールドにOUTPUTを指定すると、SYSOUTCLASSは デフォルトでアスタリスク(*)になります。

blank DSNAME属性をブランクにし、Typefile属性にINPUTまたは RDBACKを指定すると、SYSOUTCLASSはデフォルトで ブランク文字になります。後者の場合、TYPEFILE=INPUT または TYPEFILE=RDBACK ではDSNAME を指定する必要があるため、オープン操作は失敗します。

SYSOUTCLASSをDSNAMEの代わりに使用できます。 DSNAMEと同様に、 キューはすでにJCL DDステートメントを使用してSYSOUTに 事前割り振りされている場合があります。JCL DD ステートメントは、TDQUEUE資源定義を使用して作成された仕様を 指定変更します。

区画外一時データ・キューを開く要求を CICS が受信すると、データ・セット定義が作成されているかを確認するために始動 JCL が参照されます。見つからない場合は、 DSNAME属性で指定された44文字の名前を使用して、 必要なデータ・セットが動的に割り振られます。

SYSOUTがJCL内のキューに対して指定されている場合は、 クラス以外の属性も指定できます（例えば、形式タイプ）。

注: RDOを使用してSYSOUTデータ・セットを指定すると、クラス・ パラメーターだけがサポートされます。その他のパラメーターが 必要な場合は、JCLでSYSOUTデータ・セットを指定します。

SYSOUT およびその関連クラスについて詳しくは、z/OS MVS JCL ユーザーズ・ガイドを参照してください。

TDQUEUE(*name*)

一時データ・キューの 1 から 4 文字の名前を指定します。

許容文字: A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

指定された名前が4文字より短い場合は、左寄せにされ、 4文字になるようブランクが埋め込まれます。

注:

1. 名前にコンマ (,) を使用すると、以下のようなコマンドを使用できなくなります。

   ```
   CEMT INQUIRE TDQUEUE(value1,value2)
   CEMT SET     TDQUEUE(value1,value2)
   ```

   これらのコマンドでは、コンマがリスト区切り文字として機能します。リソース ID のリストの使用については、 を参照してください。

2. RACF を使用して一時データ・キューを保護する場合は、% および & を名前に使用しないでください。 RACF コマンドは、これらの文字がプロファイル名で使用された場合に、特殊な意味を割り当てます。 を参照してください。

TRANSID(*transaction*) **(区画内キューのみ)**
トリガー・レベルに達したときに自動的に開始されるトランザクションの名前を指定します。このようにトランザクションを開始し、キューからレコードを読み取ります。 TRANSID属性が指定されていない （またはTRIGGERLEVEL(0)が指定されている）場合は、別の方法を 使用してトランザクションによる一時データ・キューからの レコードの読み取りをスケジュールに入れる必要があります。

指定するトランザクションは、リモート **CICS** システム内にあってはなりません。リモート・システムに置くと、トランザクションの開始が 失敗し、警告メッセージがコンソールに出力されます。

TRIGGERLEVEL({**1**|*number*}) **(区画内キューのみ)**
レコードを処理するためにタスクが自動的に開始される前に集計するレコードの数を指定します。（この数はトリガー・レベルと 呼ばれます）。

TRANSID属性を指定すると、TRIGGERLEVELはデフォルトで 1になります。ATI処理を使用不可にするには、トリガー・レベル 0を指定します。トランザクションIDを指定しない場合、 トリガー・レベルは無視されます。

ATIFACILITY(TERMINAL)を指定した場合は、指定した端末が 使用可能になるまでタスクが開始されません。 ATIFACILITY(FILE)を指定した場合、タスクの開始に端末は不要です。

最大タスクで端末以外の宛先にストレージ不足または スペース不足の条件がある場合、タスクは開始 されません。これは、初期化のステージ1と2, およびシャットダウンの最終ステージにも当てはまり ます。タスクはストレス条件が存在しなくなったときに 開始され、その後にTD WRITEが発生します。

論理的にリカバリー可能な一時データ・キューでは、タスクが 前方にコミットするまでATIタスクは接続されません。つまり、 トリガー・レベルをかなり超えてからATIが発生する場合が あります。

z/OS Communications Server 端末が、トリガー・レベル 1 を持つ 2 つの ISC CICS システム上の CSTL トランザクションの宛先として定義されている場合、両方のシステムが繰り返し端末の取得と解放を行い、セッション開始とセッション終了のメッセージを書き込んだ場合、パフォーマンスの問題が発生する可能性があります。

CEMT トランザクションを使用して CICS が稼働している場合は、トリガー・レベルを変更することができます。これまでに累積された レコードと同じ（またはそれより少ない）数にトリガー・ レベルを下げると、次のレコードが正常にキューに置かれた ときにタスクが開始されます。

1 累積できるレコードは1つだけです。

number
累積レコード数がこの値を超えると、ATIが発生 します（最大値は32767）。

TYPE({EXTRA|INTRA|INDIRECT})

以下のタイプの一時データ・キューを指定します。

EXTRA

CICS 領域外にあるキューが、CICS に割り振られます。

区分外キューの用途：

- CICS 領域外のデータの送信: 例えば、バッチ・プログラムによって処理するためにトランザクションによって作成されたデータ。
- 領域外からのデータの取得。例えば、トランザクションへの 入力として端末から受信されたデータ。

区分外データは順次であり、QSAMによって管理されます。

INDIRECT

間接キューは、実際のデータ・セットではなく別のキューを指す キューです。間接キューは、区分外、区分内、リモート、 または別の間接キューです。

例えば、いくつかの異なるメッセージ・タイプのそれぞれに 異なるシンボル名INDIRECTDESTを付けることができます。 次に、同じ物理キュー(INDIRECTDEST)または別の物理キューへ これらすべてのメッセージ・タイプを送信できます。

DFH$TDWTサンプル・プログラムは、間接キューを使用して 異なるカテゴリーのメッセージを同じ端末へ送信する方法を 示しています。DFH$TDWT に関するプログラミング情報については、一時データを端末に書き込むプログラム (DFH$TDWT) の使用 を参照してください。DFH$TDWT サンプル定義は、「*CICS/ESA 4.1 Sample Applications Guide*」で提供されています。

EXEC CICS WRITEQ TD、EXEC CICS READQ、あるいは EXEC CICS DELETEQ コマンドの QUEUE オペランドで間接キューが指定された場合、アクセス権限は、最終ターゲット・キューのセキュリティー設定によって決まります。

INTRA

一時的に保管されるデータのキュー。

区分内宛先は、端末、ファイル、または別のシステム です。すべての区分内キューのデータを保持するため、 VSAMによって管理される単一のデータ・セットが使用されます。

レコードを処理するトランザクションと区分内キューごとの トリガー・レベルを指定できます。トリガー・レベルは、 指定したトランザクションが開始される前に累積が許される レコード数を示します。 トリガー・レベルについて詳しくは、TRIGGERLEVEL属性の 説明を参照してください。

区分内キューは、論理的にリカバリー可能、物理的にリカバリー可能、またはリカバリー不能として定義できます。

論理的にリカバリー可能なキューは、（個々の トランザクションの失敗またはシステム全体の障害の後） 最後に完了した作業単位(UOW)が

終了した時の状況に復元 されます。(UOWは、タスクの開始時または同期点で開始され、 タスクの終了時または同期点で終了します）。

物理的にリカバリー可能なキューは、 （システム全体の障害の後）システム障害の発生時の 状況に復元されます。

TDQUEUE リソース定義では、TYPE=REMOTE を指定することはできません。リモート一時データ・キューを定義したい場合は、TYPE 属性をブランクのままにし、リモート属性 REMOTELENGTH、REMOTENAME、および REMOTESYSTEM の値を指定します。あるいは、リモート属性を他の一時データ・キュー・タイプのリソース定義の一部として組み込むこともできます。

TYPEFILE({**INPUT**|**OUTPUT**|**RDBACK**})

キューを関連付けるデータ・セットのタイプを指定します。

INPUT

入力データ・セット。

OUTPUT

出力データ・セット。

RDBACK

逆方向読み取りされる入力データ・セット。

注: これは、磁気テープで定義されたデータ・セットにのみ 適しています。

区分外キューは入力または出力ですが、その両方にする ことはできません。

DCB マクロ・フィールドについて詳しくは、z/OS DFSMS Macro Instructions for Data Setsを参照してください。

USERID(*userid*) **(区画内キューのみ)**

TRANSID フィールドで指定されたトリガー・レベル・トランザクションを検査する際に、CICS がセキュリティー検査に使用するユーザー ID を指定します。 ユーザー ID の最大長は 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

USERIDフィールドに入力された値は、ATIFACILITY(FILE)も 指定されている場合のみ有効になります。

安全保護がアクティブである場合、トリガー・レベル・トランザ クションは、指定されたユーザーID権限下で実行されます。この ユーザーIDは、トリガー・レベル・トランザクションで使用される すべての資源に対して許可されている必要があります。

一時データ・キュー定義でユーザー ID を省略した場合、CICS は、DFLTUSER システム初期設定パラメーターで指定された CICS デフォルト・ユーザー ID を使用します。安全保護 検査は、ユーザーIDを含む区分内定義のインストール時に 実行されます。安全保護検査が不合格になると、 その区分内キューの資源定義はインストールされません。

代理ユーザー・セキュリティーについて詳しくは、一時データのトリガー・レベル・トランザクション を参照してください。

WAIT({YES|NO}) (区画内キューのみ)
論理的にリカバリー可能なキューを変更した未確定の作業単位 (UOW) が、変更をコミットするか取り消すかを決めるために、コーディネーターとの再同期を待機する必要があるかどうかを指定します。

NO UOW は待機しません。リカバリー可能リソースに対する変更は、トランザクション・リソース定義の ACTION 属性での指定に応じて、バックアウトされるかまたはコミットされます。

YES UOWは待機し、待機中に必要なアクションはWAITACTION属性に よって決定されます。

この属性は、UOWのトランザクション定義で定義された WAIT属性を指定変更します。TDQUEUE 定義および TRANSACTION 定義の未確定属性の相互作用の説明については、 256 ページの表 4 を参照してください。

WAITACTION({REJECT|QUEUE}) (区画内キューのみ)
このキューの定義で WAIT(YES) が指定されている場合に、CICS が未確定の作業単位 (UOW) に対して実行するアクションを指定します。可能なアクションは以下のとおりです。

QUEUE
UOW は未確定で待機中です。このキューに対して UOW が保持しているすべてのロックは、UOW の最終状態が判明するまでアクティブのまま残ります。つまり、タスクは LOCKED 応答を受け取るのではなく、中断状態になります。 UOW の最終状態が判明したら、その UOW による変更はコミットされるか取り消されます。それまで、アクティブなロックのいずれかを 必要とする次のタイプの今後の要求は待機する必要があります。

- READQ (未確定 UOW が READQ 要求または DELETEQ 要求を発行した場合)
- WRITEQ (未確定 UOW が WRITEQ 要求または DELETEQ 要求を発行した場合)
- DELETEQ (未確定 UOW が READQ 要求、WRITEQ 要求、または DELETEQ 要求を発行した場合)

REJECT
UOW は未確定で待機中です。このキューに対して UOW が保持しているロックは、UOW の最終状態が判明するまで保持されます。最終状態が判明したら、UOW による変更はコミットされるか取り消されます。それまでは、保持されているロックのいずれかを必要とする今後の要求はすべて拒否され、LOCKED 応答が返されます。
WAITACTION=REJECT により、QUEUE によってトランザクションが待機するようになる状況と全く同じ状況で、LOCKED が起きるようになります。

必須の TDQUEUE 定義

CICS サービスは特定の一時データ・キューを使用するため、関連する TDQUEUE 定義は、コールド・スタート中にできるだけ早くインストールされる必要があります。

必要な TDQUEUE 定義ができるだけ早くインストールされるようにするには、以下のいずれかの方法で **GRPLIST** システム初期設定パラメーターを指定します。

- **GRPLIST** で最初のリストとして DFHLIST を指定します。 DFHLIST リストには、グループ DFHDCTG が含まれています。このグループには、該当の TDQUEUE 定義が含まれています。
- グループ DFHDCTG が **GRPLIST** の最初のリストの最初のグループであることを指定します。

グループ DFHDCTG 内の一時データ・キューは、以下のとおりです。

CADL (z/OS Communications Server リソース定義をログに記録するために必要)
z/OS Communications Server リソースの場合、この宛先に、アクティブ CICS システムにインストールされた各 RDO 定義のログが保持されます。ログには、 TCTでの項目のインストールと、TCTからの自動インストール済み 項目の削除の両方が記録されます。 次の方法でインストールされた定義が記録されます。

- 自動インストール
- CEDA INSTALLを使用
- システムの初期化

一時データ・キューのデータ・セットの定義について詳しくは、Defining data sets を参照してください。

CADO (CICS Development Deployment Tool メッセージに必要)
CADO は、CICS Development Deployment Tool からのメッセージをログに記録する一時データ・キューです。

CADS (システム・プログラミング・インターフェース・コマンドの監査によって生成されたメッセージに必要)
CADS は、システム・プログラミング・インターフェース・コマンドの監査時に生成されたメッセージをログに記録する一時データ・キューです。

CAIL (自動インストール端末モデル定義をログに記録するために必要)
自動インストール端末モデル・マネージャー(AITM)は この宛先を使用して、TCTでインストールおよび削除された すべての自動インストール端末モデル項目を記録します。

CCPI (CPI 通信メッセージに必要)
通信の共通プログラミング・インターフェース(CPI通信）は、 この宛先にメッセージを書き込みます。

CCSE (C 言語サポートに必要)
CICS は、C 標準ストリームを一時データ・キューに送信します。 (キュー名は CICS で固定されるため、C 標準ストリームを他のキューにリダイレクトすることはできません。) Cプログラムは stderrへの書き込みによりCCSEキューに書

き込みます。この宛先は、区分外、区分内、 または間接としてコード化できます。 CCSE キュー用の TDQUEUE 定義を提供しない場合、C プログラムでの stderr への書き込みは失敗します。

CCSI (オプション、C 言語サポート用)
CCSIキューは、入力データ用のC標準ストリーム、stdin用に 予約されています。CCSI キュー名は stdin 用に予約されていますが、CICS で stdin からの読み取りを試行すると、EOF が返されます。そのため、この宛先はオプションです。これは、区画外、区画内、あるいは間接としてコーディングすることができます。

CCSO (C 言語サポートに必要)
CCSOは、出力データ用のC標準ストリーム、stdoutに関連付け られています。この宛先は、区分外、区分内、 または間接としてコード化できます。 CCSO キューを定義しない場合、C プログラムでの stdout への書き込みは失敗します。

CCZM (CICS クラス・メッセージに必要)
CCZM は、CICS ファウンデーション・クラスからのメッセージをログに記録する一時データ・キューです。

CDBC (DBCTL DFHDB81xx メッセージに必要)
CDBCは間接キューとして定義され、これはCSML区分外キューを 指します。このデータ・ログを使用するのは DBCTL DFHDB81xxメッセージだけです。その他のメッセージは 端末またはコンソールを使用します。

CDB2 (Db2 を使用する場合に必要)
CDB2 は、CICS Db2 接続機能からのメッセージおよび統計用の一時データ・キューです。

CDEP (非推奨メッセージに必要)
CDEP は、非推奨メッセージ用の一時データ・キューです。

CDUL (トランザクション・ダンプ・メッセージに必要)
CDULはトランザクション・ダンプ・メッセージの宛先です。 トランザクションの異常終了後などにトランザクション・ ダンプが要求されると、ダンプが実行されたことを示すメッセージ またはダンプが抑制された理由を示すメッセージがこの宛先に 書き込まれます。

CECO (イベントの収集および出力のメッセージに必要)
CECO は、イベント収集および出力 (EC) コンポーネントからのメッセージ用の一時データ・キューです。

CEJL (エンタープライズ Java メッセージに必要)
CEJL は、エンタープライズ Java ドメインからのメッセージ用の一時データ・キューです。

CEPO (イベント処理メッセージに必要)
CEPO は、イベント処理 (EP) ドメインからのメッセージ用の一時データ・キューです。

CESE (Language Environment からのランタイム出力に必要)
Language Environment では、すべてのランタイム出力がこの一時データ・キューに書き込まれます。詳しくは、z/OS Language Environment プログラミング・ガイドを参照してください。

CESO (Language Environment からのランタイム出力に必要)
Language Environment では、すべてのランタイム出力がこの一時データ・キューに書き込まれます。詳しくは、z/OS Language Environment プログラミング・ガイドを参照してください。

CIEO (IP ECI メッセージに必要)
CIEO は、IP ECI (IE) ドメインからのインターネット・プロトコル (IP) 外部呼び出しインターフェース (ECI) メッセージ用の一時データ・キューです。

CISL (IPCONN リソース定義をログに記録するために必要)
この宛先は、インストールされた IPCONN リソース定義のログを提供します。 CICS 初期設定中の早い段階で、CISL がまだ使用可能になっていない場合、この宛先に送信される予定であったメッセージは、コンソールに送信することができます。

CISO (IPIC メッセージに必要)
CISO は、システム間 (IS) ドメインからの IP 相互接続 (IPIC) 接続メッセージ用の一時データ・キューです。

CJRM (JRas ロギングおよびトレース機能メッセージに必要)
CJRM は、JRas のロギングおよびトレース機能からのメッセージ用の一時データ・キューです。これらのメッセージは、CICS JVM で実行されている Java クラスによって発行されるメッセージであり、CICS メッセージではありません。

CKQQ (CICS-MQ 接続メッセージに必要)
CKQQ は、CICS-MQ アダプターの始動時およびシャットダウン時に発行された CICS-MQ メッセージをログに記録するための一時データ・キューです。これは、区画内キューとして定義されます。

CMIG (マイグレーション・ログに必要)
CMIG は、CICS でサポートされなくなった機能 (例えば、EXEC CICS ADDRESS CSA コマンド) の使用を報告するメッセージを受信するマイグレーション・ログです。 CMIGは、 区分内、区分外、または間接宛先として定義できます。

CMLO (マークアップ言語メッセージに必要)
CMLO は、マークアップ言語ドメインからのメッセージ用の一時データ・キューです。

CMPO (管理対象プラットフォーム・メッセージに必要)
CMPO は、管理対象プラットフォーム・ドメインからのメッセージ用の一時データ・キューです。

CMQM (CICS-MQ メッセージに必要)
CMQM は、CICS-MQ アダプター、CICS-MQ トリガー・モニター、および CICS-MQ ブリッジによって発行されたすべての CICS-MQ メッセージをログに記録するキューです。

CPIO (CICS SOAP メッセージに必要)
CPIO は、CICS SOAP メッセージをログに記録する一時データ・キューです。

CRDI (プログラム・リソース定義をログに記録するために必要)
この宛先は、インストールされたリソース定義 (プログラム、トランザクション、マップ、マップ・セットなど) のログを提供します。

CRLO (リソース・ライフサイクル・メッセージに必要)
CRLO は、リソース・ライフサイクル・マネージャーからのメッセージ用の一時データ・キューです。

CRPO (Open Network Computing Remote Procedure Call (ONC RPC) メッセージに必要)
CRPO は、区画外キューとして定義されますが、宛先を区画内または間接にすることができます。

CSBA (ビジネス・アプリケーション・マネージャー (BAM) を使用する場合に必要)
CSBA は、ビジネス・アプリケーション・マネージャー・ドメインからのメッセージ用の一時データ・キューです。

CSBR (ブリッジ機能を使用する場合に必要)
CSBR は、ブリッジ機能メッセージをログに記録する一時データ・キューです。

CSCC (CICS クライアント・エラー・メッセージに必要)
CSCC は、CICS クライアント・エラー・メッセージをログに記録する一時データ・キューです。

CSCS (サインオン・トランザクションに必要)
CSCSは、各サインオンとサインオフの詳細を示したメッセージを 受信します。また、サインオンで拒否された試行と資源許可の 失敗に関するメッセージも受信します。 この宛先のタイプは任意です。

CSDH (文書ハンドラー・テンプレートを使用する場合に必要)
CSDH は、文書ハンドラー・テンプレート出口プログラムからの説明メッセージを受信する一時データ・キューです。

CSDL (RDO コマンドをログに記録するために必要)
リソース定義オンライン (RDO) トランザクションは、CICS システム定義 (CSD) ファイルあるいはアクティブ CICS システムを変更するすべてのコマンドをこの宛先に書き込みます。

CSDLが必要なのは、RDOを使用し、コマンドのログを 保持したい場合だけです。

CSDLに書き込まれるデータ・レコードの最大長は128バイト です。CSDLを区分外として定義する場合は、関連するSDSCI またはDDステートメントで最小ブロック・サイズの136バイトで Vフォーマット・レコードを指定する必要があります。

CSFL (ファイル・リソース定義をログに記録するために必要)
CSFL は、アクティブ CICS システムにインストールされたすべてのファイル・リソース定義のログです。ファイル資源項目の削除も ここに記録されます。

CSJE および CSJO (JVM で実行される Java プログラムからリダイレクトされた出力に必要)
CSJEとCSJOは、CICS提供のサンプル・クラス com.ibm.cics.samples.SJMergedStreamによってインターセプト されたJVMからの出力を受信します。このサンプル・クラスは、 JVMプロファイルで USEROUTPUTCLASSオプションを使用して、任意で 指定できます。
CSJEはstderr出力、内部メッセージ、リセット不能 イベント・ログに使用さ

れ、CSJOはstdout出力に使用されます。 com.ibm.cics.samples.SJMergedStream クラスは、両方の タイプの出力を処理し、それらを正しい一時データ・キューに送信 します。CSJEとCSJOは、CSSLキューを指す間接キューとして 定義されます。

JVMによって発行されるメッセージの長さはさまざま です。CSSLキューの最大レコード長(133バイト）は、 受信したメッセージの格納に十分な長さではない場合が あります。これが発生した場合、サンプルの出力リダイレクト・クラスがエラー・メッセージを発行し、メッセージ・テキストが影響を受ける可能性があります。 133 バイトより長いメッセージを JVM から受信していることを検出した場合、CSJO および CSJE を別個の一時データ・キューとして再定義する必要があります。それらのキューを区画外宛先にし、キューのレコード長を増やします。そのキューを物理データ・セットまたはシステム出力データ・セットに割り振ることができます。この場合、システム出力データ・セットの方が便利な場合があります。これは、出力を表示するためにキューをクローズする必要がないからです。 CSJOとCSJEを再定義する場合は、 グループDFHDCTGで定義された一時データ・キューの場合と 同様に、コールド・スタート中にそれらをできるだけ早く インストールしてください。

CSKL (トランザクション・リソース定義およびプロファイル・リソース定義をログに記録するために必要)
CSKL は、アクティブ CICS システムにインストールされたすべてのトランザクション・リソース定義およびプロファイル・リソース定義のログです。削除もここに記録されます。

CSLB (LIBRARY リソース定義をログに記録するために必要)
CSLB は、アクティブ CICS 領域にインストールされた動的 LIBRARY リソースに対するすべての変更のログです。つまり、LIBRARY リソースのインストールおよび破棄、LIBRARY リソースの構成に対するすべての変更 (LIBRARY の使用可能性、ランキング、あるいはクリティカル状況の変更を含む) などです。動的 LIBRARY リソースがインストールされていない場合、CSLB に対する監査ロギングは行われません。

CSML (サインオフ・トランザクションに必要)
CICS サインオフは、この宛先にデータを書き込みます。

CSMT (端末エラー・メッセージおよび異常終了メッセージに必要)
端末異常条件プログラム(DFHTACP)と異常条件プログラム (DFHACP)は、端末エラーとABENDメッセージをそれぞれ この宛先に書き込みます。この宛先は、区分外、区分内、 または間接としてコード化できます。

CSNE (ノード・エラー・メッセージに必要)
ノード異常条件プログラム(DFHZNAC)とノード・エラー・ プログラム (DFHZNEP)は、端末エラー・メッセージとデータを この宛先に書き込みます。この宛先は、区分外、区分内、 または間接としてコード化できます。

CSOO (ソケット・ドメイン・メッセージに必要)
CSOO は、ソケット・ドメインからのメッセージ用の一時データ・キューです。

CSPL (プログラム・リソース定義をログに記録するために必要)
CSPL は、アクティブ CICS システムにインストールされたすべてのプログラム・リソース定義のログです。削除もここに記録されます。

CSQL (一時データ・キュー・メッセージに必要)
CSQL は、一時データ・キュー・マネージャー自身から監査ログ・メッセージを受信する一時データ・キューです。

CSRL (パートナー・リソース定義をログに記録するために必要)
CSRL は、アクティブ CICS システムにインストールされたすべての パートナー・リソース定義のログです。削除もここに記録されます。

CSSH (スケジューラー・サービスに必要)
CSSH は、スケジューラー・サービスからメッセージを受信する一時データ・キューです。

CSSL (リカバリー・ユーティリティー統計に必要)
リカバリー・ユーティリティー・プログラム(DFHRUP)は、 統計をこの宛先に書き込みます。この宛先には、132バイトの 最小論理レコード長と136バイトの最小ブロック・サイズが 必要です。

CSTL (端末入出力エラー・メッセージに必要)
端末異常条件プログラム(DFHTACP)は、端末入出力エラー・ メッセージをこの宛先に書き込みます。この宛先は、区分外、区分内、 または間接としてコード化できます。

CSZL (フロントエンド・プログラミング・インターフェースに必要)
CSZL は、FEPI メッセージをログに記録する一時データ・キューです。

CSZX (フロントエンド・プログラミング・インターフェースに必要)
CSZX は、FEPI の予期しないイベントをログに記録する一時データ・キューです。

CWBO (CICS Web サポート・メッセージに必要)
CWBO は、CICS Web インターフェースから発行された CICS Web サポート・メッセージをログに記録する一時データ・キューです。

CWBW (CICS Web 警告メッセージに必要)
CWBW は、CICS Web サポートが受信したメッセージの HTTP 警告ヘッダーをログに記録する一時データ・キューです。

ダミー一時データ・キューの使用

データの保管が不要な場合、出力をダミー一時データ・キューに送信することができます。

このタスクについて

間接一時データ・キュー (TDQ) は、実行時に使用可能/使用不可にすることはできません。データが不要な場合は、ダミー TDQ にリダイレクトすることができます。ダミー TDQ が作成するには、TDQ を定義するときに **DSNAME(DUMMY)** を指定します。 189 ページの『TDQUEUE 属性』 を参照してください。キューは、永続的にリダイレクトすることも、一時的にリダイレクトすることもできます。

例えば、CADS キューに送信されている監査メッセージが不要であるとします。

このキューを永続的にリダイレクトするには、DFHDCTG グループ内の CADS キューの宛先を変更して、ダミー TDQ にリダイレクトされるようにします。変更は、CICS を再始動すると有効になります。

再始動を行わずにリダイレクトを制御するには、以下のようにします。

1. CADS キューの定義を DFHDCTG グループからユーザー定義グループに移動します。
2. CADS キューの定義を変更して、ダミー・キューにリダイレクトされるようにします。
3. カスタマイズしたキューをインストールします。
4. キューに必要になった場合は、CEDA を使用してグループ DFHDCTG に CADS をインストールします。

TERMINAL ターミナル

TERMINAL リソースは、CICS と通信する端末装置の特性を定義します。端末装置には、ビジュアル・ディスプレイ装置、プリンター、オペレーティング・システム・コンソール、およびファクシミリ (FAX) マシンなどの、さらに特殊な装置が含まれます。

端末装置の固有のプロパティー、また場合によっては動的なプロパティーは、CSD ファイルの TERMINAL 定義の中で定義されます。

しかし、多くの端末装置はプロパティーが同じであり、CICS に対してそれぞれを別個に、そして詳細に定義する必要はありません。各端末の定義にかかる時間と労力を軽減する方法は 2 つあります。それらは、以下のとおりです。

1. **TYPETERM** 定義 (QUERY 機能がある場合とない場合)。各 TERMINAL 定義では、多くの場合より複雑で、通常は静的な共通プロパティーを定義する TYPETERM 定義を参照する必要があります。 TERMINAL 定義の情報と TYPETERM 定義の情報の組み合わせが、TCT 中の 1 つの端末エントリー (TCTTE) となります。

   1 つの TYPETERM で、多くの端末装置の多数のプロパティーを表すことが可能です。それらのプロパティーのいくつかは、TYPETERM 定義を作成する時点で未定義でも問題ありません。それらのプロパティーについては、各端末ごとに、QUERY 構造化フィールドに基づいてログオン時点で決定することが可能です。

   しかし、それらよりさらに多くのプロパティーは、多くの端末装置に共通のものであり、その TERMINAL 定義はすべて同一になります。 CICS には、CICS がアクティブである間ずっと各端末装置ごとに TCT 中に独自のリソース定義がインストールされているという必要性を回避する機能が備わっています。
2. 自動インストール (多くの端末装置を表すために 1 つの TERMINAL 定義を使用)。ログオン時に端末装置が必要になった時点で CICS により動的にリソースを作成し、インストールすることが可能です。そのために CICS は、CSD ファイルに含まれるモデル **TERMINAL** 定義を使用します。この処理は、自動インストールと呼ばれます。

   自動インストールを使用すれば、CICS がアクティブであるときに端末装置のいくつかがログオンされていない場合、端末管理テーブル (TCT) に必要な仮想記憶域の量が少なくなります。

端末装置などの CICS 通信リソースに関する計画および管理についての詳細は、z/OS Communications Server 端末の自動インストールを参照してください。

CEMT および CEDA を使用した TERMINAL リソース定義のインストールについては、TERMINAL リソース定義のインストールを参照してください。BAS の作業については、BAS 端末リソース定義の作業を参照してください。

印刷用の端末

ディスプレイ装置のTERMINAL定義で、PRINTER属性と ALTPRINTER属性を使用してプリンターのTERMINAL定義を指定することができます。このような参照はTERMINAL定義のインストール時には解決されません。そうではなく、ディスプレイ装置でプリンターが必要なときに参照が解決されます。

印刷する出力を作成してプリンターに送信する方法はいくつかあります。

- BMS ページ・ビルド
- 次のいずれかを使用した画面コピー
  - ハードウェア・コピー鍵
  - ローカル・コピー鍵
  - ISSUE PRINTコマンド

TYPETERM定義とTERMINAL定義はプリンターとディスプレイ装置の両方に使用されます。多くの属性はプリンターだけに適用されるか、プリンター用の特別な意味を持ちます。画面コピーに使用するディスプレイ装置について指定すべき属性もいくつかあります。

プリンター

各プリンターに TERMINAL 定義を提供します。

プリンターに自動インストールを使用しない場合は、AUTINSTMODEL に NO を指定します。 (詳しくは、自動インストールおよび出力専用デバイス を参照。)

ALTPAGE

BMS の場合、PAGESIZE 属性により、デフォルトのページ・サイズおよび印刷バッファーのサイズが決まります。ページ内の行数 (長さ) と各行の文字数 (幅) を指定します。

別の属性 ALTPAGE は、代替画面サイズ (ALTSCREEN) が選択された場合に使用されるページ・サイズを示します。 ALTPAGE で指定する幅は、ALTSCREEN 属性で指定した幅と同じでなければなりません。ただし、ALTPAGE と ALTSCREEN の長さは異なっていてもかまいません。これは、表示と印刷に同じ BMS マップを使用する場合に便利です。例えば、画面をページより 1 行長くし、画面の最後の行をエラー・メッセージ用に取っておくことができます。

通常、ALTPAGE、DEFSCREEN、および ALTSCREEN の各属性は、プリンターには適用されません。

AUTOPAGE

プリンターではAUTOPAGEをYESにしなければなりませんが、すべての プリンターDEVICEタイプでRDOがYESを入力するため、ユーザーが行う 必要はあ

りません。自動ページングは、オペレーターの介入なしに、 BMSの複数のページのメッセージが連続して印刷されることを 意味します。これは、プリンターで通常必要とされる機能 です。（オペレーターが、1ページ読み終えてから 次のページを要求する場合の、3270タイプのディスプレイに 表示される複数ページのメッセージの要件とは対照的 です。）

PAGINGオプションを指定したBMS SENDコマンドのみが自動ページング を使用します。 TERMINALまたはSETを指定したBMS SENDは、自動ページングを使用 しません。

使用するプリンターの各タイプごとに、少なくとも1つのTYPETERM 定義が必要です。プリンターが一部の機能のみに使用され、 その他の機能には使用されないようにする場合は、複数の定義が 必要な場合があります。

DEVICE

各プリンターの TERMINAL 定義は、適切な DEVICE タイプを持つ TYPETERM を参照する必要があります。 DEVICE 属性と、場合によって SESSIONTYPE 属性により、TYPETERM がプリンターを定義するかディスプレイ装置を定義するかが決まります。プリンターに指定できる値は、以下のとおりです。

DEVICE	SESSIONTYPE	プリンター
3270P	-	3270 データ・ストリーム (非 SNA 接続) をサポートするすべてのプリンター。
LUTYPE3	-	3270 データ・ストリーム (SNA 接続) をサポートするすべてのプリンター。
SCSPRINT	-	SNA 文字セット (SNA 接続) をサポートするすべてのプリンター。
3790	SCSPRINT	SNA 文字セット (SNA 接続) をサポートする IBM 3793 キーボード・プリンター。

FORMFEED

BMS ページ・ビルドの場合は、FORMFEED を YES として定義します。

PAGESIZE

この端末のデフォルトのページ・サイズを指定します。行と列の積が 32767 を超えてはなりません。ここで、行はページ内の行数、列は各行の文字数です。

PGESIZE がコーディングされていない場合、以下のデフォルトが使用されます。

TW33, TW35	(12,80)
3270 ディスプレイ・モデル 1	(12,40)
3270 ディスプレイ・モデル 2	(24,80)
3270 プリンター	(12,80)

TERMINAL

プリンターの名前は、そのプリンターのリソース定義の TERMINAL 名です。

注: ディスプレイからの出力に使用するプリンター装置を参照するために、PRINTER 属性がディスプレイ装置定義で使用されます。プリンター定義で

PRINTER を使用しないでください。 『プリンターとディスプレイ装置の関連付け』 を参照してください。

プリンターとディスプレイ装置の関連付け

画面の内容を直接プリンターにコピーしたい場合は、TERMINAL 定義内で特定のプリンターをディスプレイ装置に関連付けます。これを行うには、PRINTER 属性と ALTPRINTER 属性を使用します。

ALTPRINTCOPY
代替プリンターでハードウェア・コピー機能を使用する場合は、ALTPRINTCOPY に YES を指定します。

ALTPRINTER
1 次プリンターが使用できない場合に、代替プリンターが使用されます。ディスプレイ装置定義が自動インストールされた場合、ALTPRINTER および ALTPRINTCOPY は、自動インストール制御プログラムによって動的に設定することができます。

PRINTER
1 次プリンターが使用されます。ディスプレイ装置定義が自動インストールされた場合、PRINTER および PRINTERCOPY は、自動インストール制御プログラムによって動的に設定することができます。

PRINTERCOPY
1 次プリンターでハードウェア・コピー機能を使用する場合は、PRINTERCOPY に YES を指定します。

パイプライン端末定義

3600パイプライン論理装置を定義する際に、TCTTEのプールに 関連付けられたTCTTEを生成します。

メッセージが 3600 パイプライン論理装置から CICS に入ると、プールからの TCTTE の 1 つをアンカー・ブロックとして使用して、このメッセージを処理するためのタスクが接続されます。このように、パイプライン論理装置を介して送信された連続メッセージを並行して処理することができます。その際、並行トランザクションの数は、プール内の TCTTE の数によって制限されます。プール内のTCTTEの数は、 照会活動の上限基準点を示します。 このように、パイプライン機能を使用すると、パイプライン 照会端末の総数よりも少ないTCTTEをCICSに対して定義できます。

指定された POOL 内のすべての TERMINAL 定義が、CSD ファイルの同じグループ内になければなりません。グループ内の少なくとも 1 つの定義で TASKLIMIT が指定されている必要があります。これが複数の定義で指定されている場合、使用される値は、グループ内のすべての定義の TASKLIMIT の最大値です。

TERMINAL 定義および TYPETERM 定義は、通常の端末では解決されます。

GROUP を使用して PIPELINE 端末をインストールしない場合、予測不能な結果になる可能性があります。

注**:** CICS は、パイプライン端末での自動トランザクション開始 (ATI) をサポートしません。

パイプライン端末の定義

パイプライン端末を定義する場合、特性の属性を指定します。

以下の属性を指定して、パイプライン端末を定義します。

NETNAME
使用される z/OS Communications Server セッション。

POOL
同じパイプライン・プールに属し、同じ POOL 名を持つすべての TERMINAL 定義。 POOL 属性の値があると、通常の TERMINAL 定義と区別されます。

SESSIONTYPE
パイプライン端末を表す TYPETERM を識別するには、TYPETERM 定義でこの属性を使用します。値として PIPELINE を指定します。

TASKLIMIT
少なくとも 1 つの TERMINAL 定義で、端末のプールでアクティブにすることができる並行タスクの最大数を指定します。

通常は、1 つの TYPETERM がすべての定義に適合します。TYPETERM は、別のグループに入れることができます。

パイプライン・トランザクションは、ONEWTE 属性を持つ PROFILE 定義に関連付けられます。これらのトランザクションに関連付けられたプログラムは、1 つの書き込み操作または EXEC CICS SEND 操作のみを許可されます。それ以外の操作は、ATCC 異常終了コードで終了します。このため、セッションのプール全体で CICS タスクが頻繁に出現と消失を繰り返します。

221 ページの『TERMINAL 属性』 の POOL 属性の説明で、パイプライン端末のプールの定義の例が示されています。

注:

1. CICS に既に存在するパイプラインを指定するパイプライン端末をインストールした場合、古いパイプラインとそのすべての関連端末の両方が削除された後に、新規の定義がインストールされます。
2. 既存のパイプラインを所有するように定義されている端末を破棄する場合、既存のパイプラインおよびそのすべての関連端末が削除されます。
3. パイプライン所有者ではない端末を破棄したり、その端末を別のパイプラインあるいは非パイプライン端末に変更したりしても、残りのパイプライン定義は変更されません。 (パイプラインを所有する端末は、パイプラインのインストール元のグループ内で、パイプラインに関連付けられている端末を英数字順に並べた場合の最初の名前を持つ端末です。)
4. 古いパイプラインも同じグループに再インストールする場合は、既存のパイプライン内の端末を非パイプライン端末に変更したり、その端末を新規パイプラインに変更したりすることはできません。これを行うには、インストールを 2 つのステージに分割します。グループを 2 回インストールする場合は、その間、関連端末をサービス休止状態に設定することを忘れないでください。

LDC リスト対応の装置

3600、3770 バッチ、3770、および 3790 バッチ・データ交換、LUTYPE 4 論理装置の場合、LDC (論理装置コード) リストの名前を指定することができます。

このリストはこの論理装置で有効なLDCを指定し、 オプションで各LDCで有効な装置特性を指定します。このリストの最初の LDC は、CICS が論理装置のデフォルト LDC を選択する必要がある場合のデフォルトです。

LDCLIST 属性が TYPETERM 定義上にあるため、特定の LDC リストを参照する装置のすべての TERMINAL 定義が、同じ TYPETERM を指定する必要があります。

LDC リスト自体は、DFHTCT TYPE=LDCLIST マクロを使用して定義する必要があります。これには、ローカル LDC リストと拡張ローカル LDC リストがあります。それらのコーディングの例が 277 ページの『TYPETERM 属性』 に示されています。また、詳細な説明が 453 ページの『DFHTCT 論理装置コード: z/OS Communications Server 非 3270』 に示されています。マクロ命令でコーディングされたラベルを使用し、LDCLIST 属性でそのラベルを指定することで、TYPETERM 定義の LDC リストを識別します。

APPC (LUTYPE6.2) 単一セッション端末

APPC (LUTYPE6.2) 単一セッション端末は、DEVICE(APPC) を指定した TYPETERM への参照を使用して TERMINAL として定義することができます。これらの定義がインストールされると、そこで定義されているリソースが、RDO で CONNECTION および SESSIONS として定義された場合と同様に、CICS によって接続およびモードセットとして認識されます。接続の名前は TERMINAL で、モードセットの名前は TERMINAL 定義の MODENAME です。

APPC 端末として、PS/2、Application System/400 (AS/400)、System/38 などが可能です。 APPC 端末は、TERMINAL-TYPETERM 定義または CONNECTION-SESSIONS 定義によって定義することができます。どちらの定義も自動インストールすることができます (自動インストールについては、z/OS Communications Server 端末の自動インストール および Autoinstalling APPC connections を参照)。 TERMINAL-TYPETERM 方式を使用する場合は、以下の属性が重要です。

DEVICE
TERMINAL 定義は、DEVICE として APPC (拡張プログラム間通信機能) が指定された TYPETERM を参照します。そのような TYPETERM 定義が 1 つあれば、多くの端末に対応することができます。

MODENAME
アクティブ・システムに定義がインストールされると、CICS はこの名前を使用してセッションを識別します。

トランザクション・ルーティング用の端末

トランザクション・ルーティングにより、1 つの CICS システム内の端末が、別の CICS システム内のトランザクションを開始することが可能になります。

以下のトピックでは、「システム」という単語は、IPIC、MRO、または ISC を使用して他のシステムと通信している CICS システムを示すために使用されます。IPIC、MRO、あるいは APPC リンクによって接続されたシステム間で、トランザクション・ルーティングを使用することができます。詳細については、CICS トランザクション・ルーティングを参照してください。

以下のいずれかのタイプとして、トランザクション・ルーティングに使用する端末を定義することができます。

- ローカル端末
- リモート端末
- 両用端末

定義する端末のタイプは、アプリケーション専有 CICS システム (AOR) に対して定義を使用可能にする方法によって異なります。以下のいずれかの方法を使用して、定義を使用可能にすることができます。

- 端末定義を複写
- 端末定義の共用
- 端末定義のシップ

トランザクション・ルーティングに使用される端末定義

2 つの異なる方法で、トランザクション・ルーティングに使用する端末を定義することができます。端末専有領域にローカル端末定義 をインストールし、アプリケーション専有領域にリモート端末定義 をインストールすることができます。あるいは、両方の領域に二重目的端末定義 をインストールすることもできます。

ローカル端末定義**:**

ローカル端末定義 は、端末専有システムにインストールされた端末の完全な定義です。

ローカル端末定義は、複製方式およびシップ方式で使用されます。

- TERMINAL 定義あるいは CONNECTION 定義では、REMOTESYSTEM、REMOTESYSNET、および REMOTENAME はいずれも不要です。

  TERMINAL 定義で指定された REMOTESYSTEM が (SYSIDNT システム初期設定オペランドで指定された) ローカル・システムの名前である場合、その定義は、インストール時にリモート端末ではなくローカル端末として処理されます。
- 複製定義の場合、TYPETERM の SHIPPABLE は NO です。
- シップ定義の場合、TYPETERM の SHIPPABLE は YES です。
- APPC 装置の場合、TYPETERM の SHIPPABLE は必須です。
- TERMINAL 定義および TYPETERM 定義のその他の必要な属性をすべて指定する必要があります。

リモート端末定義:

1 つのシステムに属する (そのシステムに対してローカルで完全に定義されている) 端末が、別のシステムに属するトランザクションを呼び出した場合、その端末は、アプリケーション専有領域にはリモート端末 として認識されます。

アプリケーション専有システムは、リモート端末の少なくとも部分定義に対するアクセス権限を持っている必要があります。この部分定義は、しばしばリモート定義と呼ばれます。これは、アプリケーション専有領域および中間領域にインストールされた端末の部分定義です。部分定義には、端末がそのシステムのトランザクションにアクセスするために必要な最小限の情報が含まれます。リモート定義を作成するのは、複製方式を使用する場合のみです。

- REMOTESYSTEM 名は、トランザクション・ルーティング・パス内の次の領域から端末専有領域への CONNECTION 定義の名前でなければなりません。
- REMOTESYSNET は、端末専有領域のネット名 (汎用アプリケーション ID) でなければなりません。REMOTESYSTEM で端末専有領域への直接接続が指定されている場合、端末専有領域が z/OS Communications Server の汎用リソース・グループのメンバーであり、直接接続が APPC リンクである場合を除き、REMOTESYSNET は必要ではありません。
- REMOTENAME は、端末専有領域で端末あるいは APPC 装置を認識するための名前です。この名前は、アプリケーション専有システムが端末を認識するための名前でなければならない TERMINAL 名や CONNECTION 名と同じであっても異なっていてもかまいません。 REMOTENAME を指定しない場合、デフォルトで TERMINAL 名になります。
- 非 APPC 装置の場合、TYPETERM の SHIPPABLE は NO です。
- APPC 装置の場合、TYPETERM の SHIPPABLE は必須です。
- TERMINAL 定義および TYPETERM 定義の一部の属性は省略することができます。これらの定義について詳しくは、「構成」の「相互通信リソースの定義」を参照してください。

以下の端末および論理装置は、トランザクション・ルーティングを使用することができないため、リモートとして定義することができません。

- プールされた 3600 または 3650 パイプライン論理装置
- IBM 2260 端末
- MVS コンソール

二重目的端末定義:

二重目的端末定義 は、端末専有システムおよびアプリケーション専有領域と中間領域にインストールされた端末の完全な定義です。

これは、共用方式でのみ使用されます。

- REMOTESYSTEM 名は、端末専有領域の SYSIDNT でなければなりません。これは、アプリケーション専有領域から中間領域 (存在する場合) までの間、および中間領域から端末専有領域までの間に定義する CONNECTION の名前でもなければなりません。アプリケーション専有領域から端末専有領域までのパス上にあるすべての CONNECTION 定義に、この同じ名前が指定されている必要があります。

- REMOTESYSNET は、端末専有領域のネット名 (汎用アプリケーション ID) でなければなりません。
- REMOTENAME は、TERMINAL 名あるいは CONNECTION 名と同じです。
- 非 APPC 装置の場合、TYPETERM の SHIPPABLE は NO です。
- APPC 装置の場合、TYPETERM の SHIPPABLE は必須です。
- TERMINAL 定義および TYPETERM 定義のその他の必要な属性をすべて指定する必要があります。

トランザクション・ルーティングに使用可能な部分定義の作成

いくつかの方法で、アプリケーション専有 CICS 領域で使用可能な端末の部分定義を作成することができます。

端末定義のシップ:

必要な場合に他の領域にシップできるローカル端末定義を、端末専有領域で使用可能な CSD ファイル内に定義できます。

1. 端末専有領域の CSD ファイルに端末のローカル定義を作成します。 (これは、共用 CSD ファイルであっても構いません。)
2. 端末専有領域にローカル定義をインストールします。(この定義は、システムの初期設定時に、または CEDA INSTALL を使用するか、自動インストールによりインストールできます。)
3. 端末が、他の領域に属するトランザクションを呼び出すとき、リモート定義を作成するのに必要な情報がその領域にシップされ、一時定義がそこに自動的にインストールされます。

ローカル定義が自動インストールされた場合、シップされた定義は、端末がログオフされるまで存続します。それ以外の場合、シップされた定義は、ローカル定義が再びインストールされるまで、または領域間のリンクが壊れるまで存続します。

端末定義を複写:

関係する各領域に対して個別の端末定義を、それぞれその領域からアクセス可能な CSD ファイル上に作成することができます。

ローカル定義として 1 つ作成し、リモート定義として 1 つ以上を作成します。同じ端末に対して複数のリソース定義が存在するため、これは複製端末定義と呼ばれます (これらの定義は、必ずしも相互に完全に複製しているわけではありません)。

端末定義を複製するには、以下のようにします。

1. 端末専有領域の CSD ファイルまたは共用 CSD ファイルに端末のローカル定義を作成します。
2. アプリケーション専有領域の CSD ファイルまたは共用 CSD ファイルに端末のリモート定義を作成します。複数のアプリケーション専有領域がある場合は、複数のリモート定義が必要になる場合がありますが、それらの領域が CSD ファイルを共用している場合は、それらすべての領域で同じリモート定義を使用することができます。
3. 端末専有領域にローカル定義をインストールします。この定義は、自動インストールすることができます。

4. アプリケーション専有領域にリモート定義をインストールします。

注**:** ご使用の領域で CSD ファイルを共用している場合は、以下の理由により、これらの定義が異なるグループ内になければなりません。

- それらの定義を異なる領域にインストールしたい
- TERMINAL 名が同じである可能性が高いために、定義を同じグループに入れることができない

利点*:*

複製端末定義には、いくつかの利点があります。

- この方法は、ご使用のシステムが CSD ファイルを共用しているかどうかに関係なく使用することができるため、異なる MVS イメージ間でのトランザクション・ルーティングに使用することができます。
- これは、ご使用の端末が、異なるシステムで異なる名前によって認識されている場合に使用できる唯一の方法です。
- XALTENF 出口および XICTENF 出口をセットアップしなくても、リモート端末のアプリケーション専有システム内で自動トランザクション開始 (ATI) を使用することができます。これは、プリンターでは特に有用です。プリンターは、ATI が正常に実行されるために事前に獲得する必要があるためです。

欠点*:*

端末定義の複製には、いくつかの欠点があります。

- CSD ファイルが定義に使用するディスク・ストレージの必要量が 2 倍以上になります。
- TCT は、仮想ストレージの必要量より多くの量を定義に使用します。
- 定義の保持に関連する不要な作業が増えます。
- 端末専有システムには端末を自動インストールすることができますが、アプリケーション専有システムには自動インストールすることができません。

例*:*

以下の例では、APPC 装置 (ネット名 APPC1) が端末専有領域 (アプリケーション ID CICA) に接続されています。さらに、この端末専有領域は、アプリケーション専有領域 (アプリケーション ID CICB) に接続されています。

端末専有領域にインストールされたリソースは、CSD1 で保持され、アプリケーション専有領域にインストールされたリソースは、CSD2 で保持されます。構成は、220 ページの図 4 に示されています。

以下の表は、各領域に必要なリソース定義を示しています。

定義の目的	端末専有領域内の定義	アプリケーション専有領域内の定義
システム間の接続を定義する	`CONNECTION(AOR)` `NETNAME(CICB)`	`CONNECTION(TOR)` `NETNAME(CICA)`

定義の目的	端末専有領域内の定義	アプリケーション専有領域内の定義
TERMINAL 定義および TYPETERM 定義を使用して、APPC 装置への接続を定義する	TERMINAL(APC1) TYPETERM(APPC0001) NETNAME(APPC1) TYPETERM(APPC0001) DEVICE(APPC)	TERMINAL(BPC1) TYPETERM(APPC0001) NETNAME(APPC1) REMOTESYSTEM(TOR) REMOTENAME(APC1) TYPETERM(APPC0001) DEVICE(APPC)
CONNECTION 定義および SESSIONS 定義を使用して、APPC 装置への接続を定義する	CONNECTION(APC1) NETNAME(APPC1) SESSIONS(APPC0001) CONNECTION(APC1)	CONNECTION(BPC1) NETNAME(APPC1) REMOTESYSTEM(TOR) REMOTENAME(APC1)

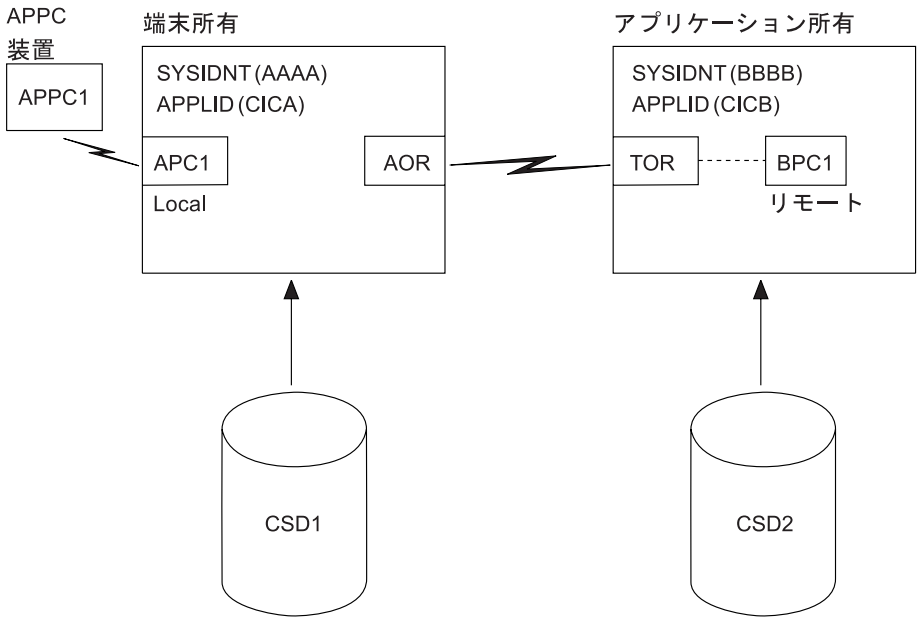


図 4. ローカル定義とリモート定義の別個での保持：

トランザクション・ルーティング用の APPC 装置

APPC トランザクション・ルーティングを使用すると、1 つの CICS システムに属している APPC 装置が、別の CICS システムでトランザクションを呼び出すことができます。

215 ページの『トランザクション・ルーティング用の端末』 で説明されているトランザクション・ルーティング用に端末を定義する 3 つの方法は、APPC 装置にも適用できます。これらの方法では、TERMINAL-TYPETERM 定義または CONNECTION-SESSION 定義の使用を呼び出すことができます。APPC 単一セッション端末の場合、TERMINAL-TYPETERM 定義を使用する方法が推奨されます (215 ページの『APPC (LUTYPE6.2) 単一セッション端末』 を参照)。

トランザクション・ルーティングの要約

トランザクション・ルーティング用に端末を定義する前に、端末を定義するさまざまな方法のうち、どの方法がご使用の CICS 構成に最適であるかを検討する必要があります。

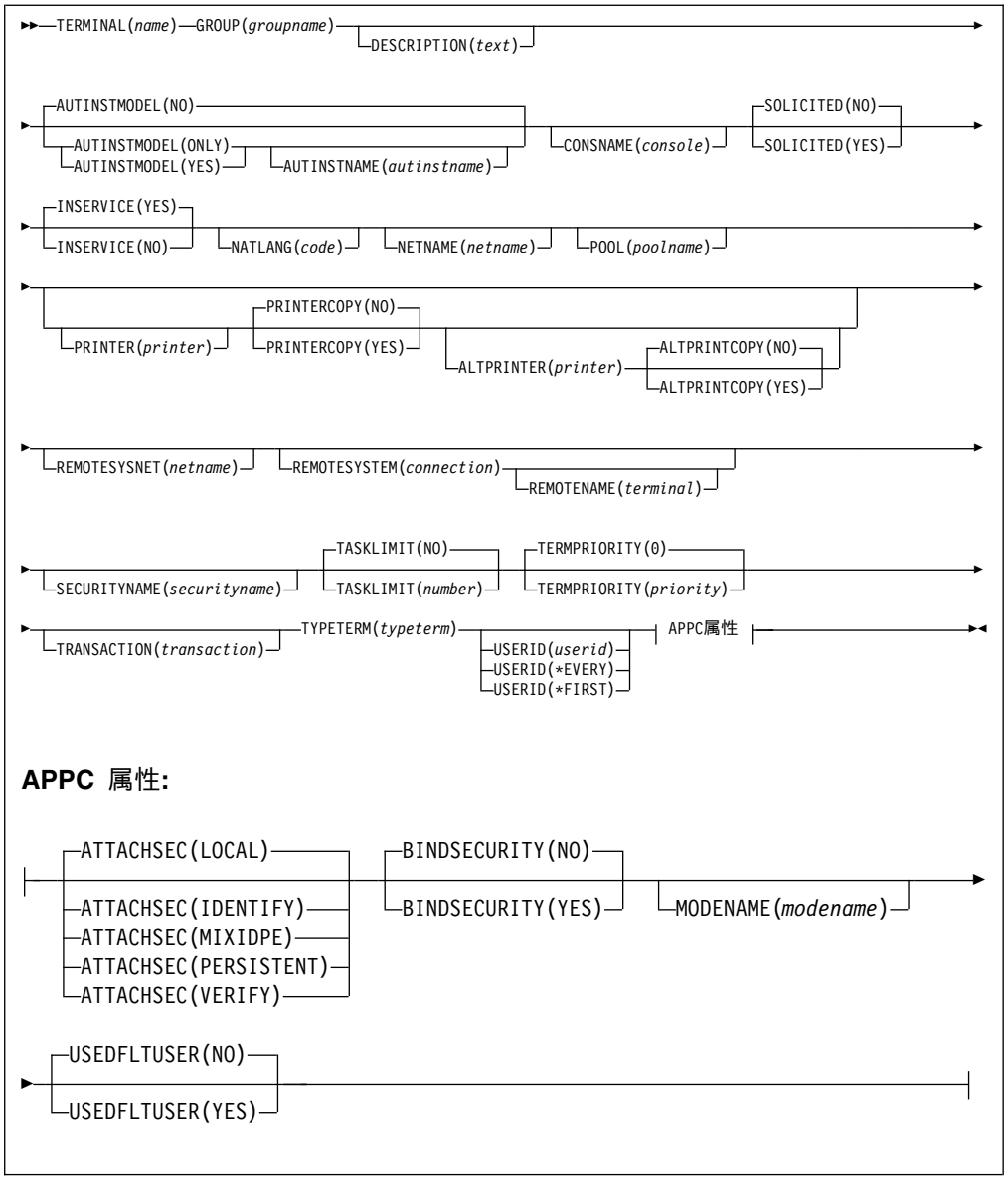
- 異なるシステムで異なる名前によって認識されている端末を使用している場合を除き、シップ方式を使用します。トランザクション所有システムで ATI がシップ方式を行うためには、XALTENF グローバル出口および XICTENF グローバル出口を使用する必要がある場合があります。これらの出口に関するプログラミング情報については、端末未認識状態出口 XALTENF および XICTENF を参照してください。
- 異なるシステムで同じ名前を使用しており、ATI を機能させるためにグローバル出口を使用したくない場合、共用 CSD ファイルを使用するシステムでは、共用方式を使用します。
- 異なるシステムで異なる名前によって認識されている端末を使用している場合、あるいは ATI を使用して端末を獲得しているが共用 CSD ファイルがない場合に、XALTENF グローバル・ユーザー出口および XICTENF グローバル・ユーザー出口を使用したくない場合は、複製方式を使用します。

これらの方式を組み合わせて使用することもできます。例えば、ディスプレイ端末にはシップ方式を使用し、ATI を使用して獲得する必要があるが、XALTENF グローバル・ユーザー出口および XICTENF グローバル・ユーザー出口を使用しないプリンターには複製方式を使用するなどです。

相互通信リソースの定義の作成を開始する前に、リモート・システムへの接続の定義 で詳細を確認してください。ここには、さまざまなタイプのリンクやセッションに指定する必要がある属性について、有用な例が記載されています。

TERMINAL 属性

TERMINAL リソースの構文と属性について記述します。



APPC 属性:

ALTPRINTCOPY({**NO**|**YES**})

ALTPRINTER 属性で指定されたプリンターで、印刷要求を満たすためにハードウェア・コピー機能を使用するかどうかを指定します。詳しくは、PRINTERCOPY 属性を参照してください。

NO CICS は、ハードウェア・コピー機能を使用します。

YES CICS は、ハードウェア・コピー機能を使用しません。

ALTPRINTER(*printer*)

この端末定義の PRINTER 属性で指定されたプリンターが使用できない場合に使用する 3270 プリンターの名前を指定します。 名前は最大4文字です。 詳しくは、PRINTER 属性を参照してください。PRINTER を指定せずに ALTPRINTER を指定した場合、ALTPRINTER は無視されます。

指定するプリンターは、この端末定義を所有している CICS システムが所有している必要があります。

代替プリンターのハードウェア・コピー機能を指定するには、この端末定義の ALTPRINTCOPY に YES を指定します。

ATTACHSEC({LOCAL|IDENTIFY|VERIFY| PERSISTENT|MIXIDPE}) (APPC のみ)
接続に必要な接続時ユーザー・セキュリティーのレベルを指定します。PERSISTENT および MIXIDPE は、アクセス方式として z/OS Communications Server を使用しており、APPC が使用されている場合にのみ有効です。

LOCAL
ユーザーの権限がリンク自体の権限と見なされ、リンク・セキュリティーはリソースのみを保護します。

IDENTIFY
着信接続要求には、ユーザー ID を指定する必要があります。接続している端末にセキュリティー・マネージャーがある場合は、IDENTIFY を指定します。

MIXIDPE
接続は、IDENTIFY および PERSISTENT のいずれかまたは両方のセキュリティー・タイプを使用して、接続をサポートできます。使用されるセキュリティー・タイプは、着信接続によって異なります。

前のリリースと同様に、IDENTIFYは、一方のシステムによる 他方のシステムのサインオン論理の受け入れを可能にする2つの システム間の信頼の度合いを示します。事実上、一方のシステムがサインオン機能を実行し、他方のシステムがセキュリティー検査を実行する分散セキュリティー・マネージャーを持つことになります。

PERSISTENT
このオプションには、リモート・システムへのユーザー・サインオンが含まれています。このサインオンは、ユーザーがリモート・システムをサインオフするまで複数の会話にわたって持続します。この方法では、ユーザー ID とパスワードが渡されるのは、最初の (サインオン) 接続のときだけです。その後の接続要求では、ユーザー ID のみが必要です。

VERIFY
着信接続要求には、ユーザー ID とユーザー・パスワードを指定する必要があります。接続している端末にセキュリティー・マネージャーがないために検査が必要な場合、VERIFY を指定します。

AUTINSTMODEL({NO|YES|ONLY})
この端末定義を自動インストールのモデル端末定義として使用できるかどうかを指定します。自動インストールおよびモデル端末定義について詳しくは、z/OS Communications Server 端末の自動インストール を参照してください。

NO この定義は、自動インストールのモデルとしては使用されません。それは自動インストールされない特定の装置の定義としてのみ使用されます。

ONLY
この定義は、自動インストールのモデルとしてのみ使用されます。特定の装置の定義としては使用されません。

YES この定義は、自動インストールされない特定の装置に対して 使用されます。この定義は、自動インストールのモデルとしても使用されます。

AUTINSTNAME(*autinstname*)
自動インストール制御プログラム内でこのモデル定義を認識するための名前を指定します。名前の長さは、最大で 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

この名前を指定するのは、AUTINSTMODEL が YES または ONLY である場合のみです。自動インストール制御プログラムで許容できる場合は、端末名の後に 4 つのブランクを付けたデフォルトの名前を使用することもできます。自動インストール・モデル、自動インストール名、および自動インストール制御プログラムについて詳しくは、自動インストール制御プログラム を参照してください。

BINDPASSWORD
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

BINDSECURITY({**NO**|**YES**}) **(APPC のみ)**
バインド時のセキュリティーに外部セキュリティー・マネージャー (ESM) を使用するかどうかを指定します。

NO 外部のバインド時セキュリティーは必要ありません。

YES セキュリティーがアクティブであり、**XAPPC** システム初期設定パラメーターが YES に設定されている場合、CICS は、バインド時セキュリティーを実行するために RACF からセッション・キーの抽出を試みます。使用可能な RACF プロファイルがない場合、バインドは失敗します。

CONSOLE
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

CONSNAME(*console*)
CONSNAME を使用することで、既存のコンソールがない場合や、MVS SYS1.PARMLIB の CONSOL*nn* メンバーで以前に定義されていたコンソールがない場合でも、CICS コンソール定義をインストールすることができます。ただし、コンソールを使用するには、事前に SYS1.PARMLIB の CONSOL*nn* メンバーで、あるいは動的割り振りによって、MVS に対して名前を定義する必要があります。 CONSNAME の長さは 2 から 8 文字で、英字または #、@、$ のいずれかで開始しなければなりません。これにより、接続先の MVS イメージに関係なく、CICS 領域内でコンソール装置が一意的に識別されます。つまり、同じ CONSNAME で 2 つのコンソール定義をインストールすることはできません。 CONSNAME は、MVS SYS1.PARMLIB のメンバー CONSOL*nn* でコンソールに対して定義された名前に対応します。

TSO ユーザーをコンソール装置として定義するには、CONSNAME(*name*) を指定します。ここで、*name* は TSO ユーザー ID です。この定義により、TSO CONSOLE コマンドの使用が許可された TSO ユーザーが、CICS トランザクションを開始できるようになります。TSO ユーザー ID が SYS1.PARMLIB メンバーの CONSOL*nn* メンバーで定義されている必要はありません。

CONSOLE(00) に相当するのは、CICS のサービス・レベルと使用する MVS のリリースに応じて、CONSNAME(INTERNAL) または CONSNAME (INSTREAM) になります。このオプションは、CICS トランザクションを開始し、JCL ステートメントでそのトランザクションに対してコマンドを発行したい場合に指定します。JCL を使用して CICS コマンドを発行する方法については、JCL による CICS コマンドの開始 を参照してください。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

INSERVICE({**YES**|**NO**})
定義する端末の状況を指定します。

YES トランザクションを開始することができ、メッセージが自動的に端末に送信されます。

NO 端末は、メッセージを受信したり、入力を送信したりすることができません。

MODENAME(*modename*) **(APPC 単一セッション端末のみ)**
LOGMODE 名として z/OS Communications Server に渡される名前を指定します。この名前は最大 8 文字の長さで指定することができますが、予約名 SNASVCMG を使用することはできません。この名前はアセンブラー言語の規則に従い、先頭文字は英字でなければなりません。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

LOGMODE 名について詳しくは、z/OS Communications Server LOGON モード・テーブルの項目のコーディング を参照してください。

NATLANG(*code*)
この端末ですべてのグローバリゼーション対応メッセージを表示する言語を指定します。以下のいずれかの NATLANG コードを指定します。デフォルトは、NATLANG システム初期設定パラメーター で指定された値です。

E 英語。これはシステム・デフォルトです (つまり、明確にコード E を指定しない場合でも、これが指定されます)。

C 中国語 (簡体字)。これは 2 バイト文字セット言語です。翻訳は IBM によって行われます。

K 日本語。これは 2 バイト文字セット言語です。翻訳は IBM によって行われます。

NETNAME(*netname*)
ACF/Communications Server で端末を識別するネットワーク名を指定します。名前の長さは、最大で 8 文字です。この名前はアセンブラー言語の規則に従い、先頭文字は英字でなければなりません。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

名前を指定しなかった場合、NETNAME はデフォルトの TERMINAL 名になります。

NETNAME は、リモート端末の場合を除いて、固有でなければなりません。つまり、同じ NETNAME を持つ 2 つのローカル端末をインストールしたり、同じ NETNAME を持つローカル端末と接続をインストールしたりすることはできません。ただし、リモート端末のNETNAMEは、他の端末の NETNAMEや接続のNETNAMEと同じにすることができます。

CICS 領域が z/OS Communications Server 動的 LU 別名をサポートしている場合 (つまり、CICS 領域の APPL ステートメントで LUAPFX=*xx* が指定されている場合)、この NETNAME は CICS 領域と同じネットワーク内にあるものと見なされます。端末が別のネットワーク内にある場合は、事前定義された LUALIAS (LUALIAS=*netname*) を使用して CDRSC 定義で z/OS Communications Server に対して定義し、z/OS Communications Server の動的割り振りをオーバーライドする必要があります。この場合、LUALIAS パラメーターの *netname* は、この端末リソース定義で定義された NETNAME に一致している必要があります。

OPERID
OPERPRIORITY
OPERRSL
OPERSECURITY
これらの属性は廃止されましたが、以前のリリースの CICS との互換性を確保するためにサポートされています。

POOL(*poolname*)

他のパイプライン端末と共にプールされる 3600 または 3650 パイプライン端末のプール名を指定します。

3600パイプライン論理装置を定義する際に、TCTTEのプールに 関連付けられたTCTTEを生成します。パイプラインTCTTEの プールは、1つのパイプライン論理装置が使用することもできますし、 いくつものパイプライン論理装置が共有することもできます。

プール名は、CSDファイルで関連するTERMINAL定義を識別する ためにのみ使用されます。これは、アクティブ CICS システムでは使用されません。この名前は、8 文字以内の長さで指定することができます。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

パイプライン端末には、SESSIONTYPE(PIPELINE)を指定して TYPETERMを指定する必要があります。パイプライン端末の最低 1つのプールでTASKLIMITを指定する必要があります。プール内の 各パイプライン端末で同じグループ名を指定する必要があります。

以下は、パイプライン端末のプールの定義の例を示しています。

```
DEFINE TERMINAL(ttt1) GROUP(g) POOL(poolid)
       TYPETERM(xxxxxxxx) NETNAME(nnnnnnn1)...
DEFINE TERMINAL(ttt2) GROUP(g) POOL(poolid)
       TYPETERM(xxxxxxxx) NETNAME(nnnnnnn2)...
DEFINE TERMINAL(ttt3) GROUP(g) POOL(poolid)
       TYPETERM(xxxxxxxx) NETNAME(nnnnnnn3)...
DEFINE TERMINAL(ttt4) GROUP(g) POOL(poolid)
       TASKLIMIT(nn) TYPETERM(xxxxxxxx)
       NETNAME(nnnnnnn4)...
DEFINE TYPETERM(xxxxxxxx) GROUP(g)
       DEVICE(3650) SESSIONTYPE(PIPELINE)
```

PRINTER(*printer*)

ISSUE PRINT コマンドまたはプログラム・アクセス (PA) キーを押したオペレーターによる PRINT 要求に応答するために使用する、1 次 3270 印刷装置の名前を指定します。 名前は最大4文字です。 名前は、プリンターの定義上のTERMINAL名です。ここで PRINTERを指定した場合、このTERMINAL定義によって参照される TYPETERMにはPRINTADAPTER(NO)が必要です。

指定するプリンターは、この TERMINAL 定義を所有しているのと同じ CICS システムが所有している必要があります。

このTERMINAL定義が次のいずれかである場合は、PRINTERを 指定することができます。

- プリンター・アダプター・フィーチャーのない3270ディスプレイ
- 3274制御装置に接続された3270ディスプレイ
- 3276制御装置のディスプレイ装置
- 3270互換モードにおける3790

ハードウェアCOPY機構を指定する場合は、このTERMINAL 定義で PRINTERCOPY(YES)を指定します。

SNA文字ストリング(SCS)プリンターは3790データ・ストリーム のみ受け入れます。3270データ・ストリームは受け入れません。したがって、ディスプレイ装置のバッファー内容の印刷には使用することができません。

プログラム・アテンション・キー (例えば PA1) を使用して、関連付けられた z/OS Communications Server プリンターで画面の内容を印刷する場合、プリンターの画面サイズは、CICS 提供のデフォルト・プロファイル DFHCICST で定義された SCRNSIZE オペランドに従って選択されます。このプロファイルはSCRNSIZE(DEFAULT)で定義され、 プリンターの代替画面サイズを使用する場合はトランザクション CSPPのプロファイル項目を変更する必要があります。

PRINTERCOPY({**NO**|**YES**})

この端末定義の PRINTER 属性で指定されたプリンターで、印刷要求を満たすためにハードウェア・コピー機能を使用するかどうかを指定します。

CICS は、現在タスクがディスプレイに接続されていない限り、3270 のハードウェア・コピー機能を使用して印刷を実行します。

端末定義の PRINTERCOPY(YES) によって暗黙的に指定されているため、TYPETERM 定義で COPY(YES) を指定する必要はありません。

ALTPRINTER と同時に PRINTER も指定した場合、ALTPRINTCOPY(YES) を指定することができます。

COPY機構を使用するには、プリンターとディスプレイ端末 の両方が同じ3270制御装置上にある必要があります。そうしないと、コピーが失敗してエラー状態が発生するか、ディスプレイ装置のアドレスがプリンター制御装置に有効である場合は、別のディスプレイからコピーが実行される可能性があります。

ネットワーク環境で、3270 制御装置が 1 つのドメイン内で TCAM システムに接続されており、別のドメイン内の CICS システムが z/OS Communications Server を介して制御装置にアクセスする権限を持っている場合は、PRINTERCOPY(YES) を指定しないでください。ハードウェア・コピー・アドレスは、CICS で使用することができません。したがって、そのような制御装置に接続された端末で使用することもできません。

COPYコマンドは、3270互換モード・ディスプレイでは無効です。

REMOTENAME(*terminal*)

端末を所有するシステムまたは領域で端末を認識するための名前を指定します。名前は最大4文字です。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

REMOTESYSNET(*netname*)

端末を所有する領域のネットワーク名 (APPLID) を指定します。名前の長さは、最大で 8 文字です。この名前はアセンブラー言語の規則に従い、先頭文字は英字でなければなりません。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

REMOTESYSNETは、この定義がインストールされた領域と 端末専有領域の間に直接リンクがない場合に使用されます。 次のいずれかが当てはまる場合、REMOTESYSNETを 指定する必要はありません。

- ローカル端末を定義している (つまり、REMOTESYSTEM が指定されていないか、ローカル・システムの SYSID を指定している)。
- REMOTESYSTEM が、端末専有領域への直接リンクを指定している。ただし、端末専有領域が z/OS Communications Server の汎用リソース・グループのメンバーであり、直接リンクが APPC 接続である場合は、REMOTESYSNET を指定する必要がある場合があります。この場合は、 直接リンクのCONNECTION定義で指定されたNETNAMEが TORの一般資源名（アプリケーションIDではない） であるときに、REMOTESYSNETが必要です。

REMOTESYSTEM(接続)

端末を所有しているシステムへの相互通信リンクを識別する名前を指定します。名前は最大4文字です。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

REMOTESYSTEM は、以下の ID のいずれかになります。

- IPIC 接続の場合、サービス中で獲得済みの IPCONN 定義における IPCONN 名の最初の 4 文字です。
- MRO および APPC 接続の場合、CONNECTION 定義の CONNECTION 名。CONNECTION 名が指定されていない場合、あるいは、ローカル・システムのシステム ID として名前が指定されている場合、端末はこのシステムのローカル端末です。その名前が別のシステムの名前である場合、端末はリモートです。 したがって、ローカル・システムとリモート・システムの両方で端末に同じ定義を使用することができます。

この CICS と端末専有領域の間に中間システムがある場合、REMOTESYSTEM は、TOR へのパスにおける最初のリンクを指定します。複数のパスが使用可能な場合、REMOTESYSTEM は、優先パスにおける最初のリンクを指定します。

AUTINSTMODEL(YES) または AUTINSTMODEL(ONLY) を指定する場合、REMOTESYSTEM は無視されます。

SECURITYNAME(*securityname*)

リモート・システムのセキュリティー名を指定します。

セキュリティーが初期化された CICS システム (SEC=YES または MIGRATE) では、セキュリティー名はリモート・システムの権限を設定するために使用されます。

接続定義に関連付けられたセッション定義 (DEFINE SESSIONS コマンド) で USERID を指定した場合、このユーザー ID が、SECURITYNAME 属性で指定されたユーザー ID をオーバーライドし、リンク・セキュリティーに使用されます。

セキュリティー名、あるいはセッション定義の USERID は、ご使用のシステムで有効な RACF ユーザー ID でなければなりません。システム上の保護リソースへのアクセスは、RACF ユーザー・プロファイルおよびそのグループ・メンバーシップに基づいています。

MRO、LUTYPE6.1、および APPC の各接続のセキュリティーの定義について詳しくは、Intercommunication security を参照してください。

SOLICITED(**NO**|**YES**)
コンソールに発行された CICS メッセージを NetView® で送信請求として処理するか非送信請求として処理するかを指定します。

NO CICS メッセージは、非送信請求として処理されます。

YES CICS メッセージは、送信請求として処理されます。コンソールに対して SOLICITED(YES) が指定された場合、CICS は、コンソール名またはコンソール識別番号と、コマンド/応答トークンを各コンソール・メッセージに追加します。

SOLICITED属性はコンソールにのみ適用されます。これは、 その他のTERMINAL定義では無視されます。

TASKLIMIT({**NO**|*number*})
パイプライン・セッション内またはパイプライン・セッションのプール内で実行が許可される並行タスクの数を指定します。

NO 並行タスクは許可されません。

number
実行が許可される並行タスクの数 (1 から 32767 の範囲)。

3600パイプライン論理装置を定義する際に、TCTTEのプールに 関連付けられたTCTTEを生成します。メッセージが 3600 パイプライン論理装置から CICS に入ると、プールからの TCTTE の 1 つをアンカー・ブロックとして使用して、このメッセージを処理するためのタスクが接続されます。このように、パイプライン論理装置を使用して送信された連続メッセージを並行して処理することができます。その際、並行トランザクションの数は、プール内の TCTTE の数によって制限されます。プール内の TCTTE の数は、照会活動の最高水準点を表します。このように、パイプライン機能によって、CICS に定義する TCTTE の数をパイプライン照会端末の総数より少なくすることができます。

TERMINAL(*name*)
端末 ID を指定します。端末の場合、名前の最大長は 4 文字です。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

APPC LU6.2 単一セッション端末の場合、有効な文字は以下のとおりです。

許容文字:
A-Z a-z 0-9 $ @ #

指定された名前が 4 文字未満の場合、左寄せされて、4 文字になるまでブランクが埋め込まれます。 TERMINAL定義とCONNECTION定義に同じ名前を付けることは できません。

注: 名前にコンマ (,) を使用すると、以下のようなコマンドを使用できなくなります。

```
CEMT INQUIRE TERMINAL(value1,value2)
CEMT SET     TERMINAL(value1,value2)
```

これらのコマンドでは、コンマがリスト区切り文字として機能します。リソース ID のリストの使用については、リソース ID のリスト を参照してください。

また、CICS がコンソールのセッション名や端末 ID を自動的に生成することを許可する場合は、端末名の最初の文字と 4 番目の文字に以下の記号を使用しないでください。

- ハイフンは、CICS が自動生成された APPC セッションの端末名に使用します。

> および <
: > (より大) および < (より小) 記号は、CICS が自動生成された IRC セッションの端末名に使用します。

¬
: ¬ (論理 NOT) 記号は、CICS が自動生成された MVS コンソールの端末名に使用します。

指定された名前は、TERMINAL定義のインストール時に TCT項目の名前になります。このため、TERMINAL 名は固有でなければなりません。値 CERR は、エラー・コンソール用に生成される ID のために予約されているので注意してください。 AUTINSTMODEL(ONLY) を指定した場合、TCT エントリーの名前としては使用されないため、固有の TERMINAL 名は不要です。AUTINSTMODEL(YES)を指定すると、TERMINAL定義のインストール時に、TERMINAL名がTCTにインストールされたTCT項目の名前として 使用されます。自動インストールされた端末のTCT項目の名前は、 自動インストール・ユーザー・プログラムによって決定されます。

端末を一時データ宛先に関連付ける場合、端末名と TDQUEUE リソース定義内の一時データ・キュー名は同じでなければなりません。

TERMPRIORITY({**0**|*priority*})
: 端末の優先度を指定します。この 10 進値 (0 から 255) は、全体的なトランザクション処理の優先順位の設定に使用されます。トランザクション処理の優先度は、端末優先度、トランザクション優先度、およびオペレーター優先度の合計と等しく、255 を超えてはなりません。

TRANSACTION(*transaction*)
: アクティブ・タスクがない場合に、端末から入力を受信するたびに開始されるトランザクションの名前 (1 から 4 文字) を指定します。

許容文字:

A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

z/OS Communications Server 非 3270 装置の場合、4 文字未満の TRANSACTION 名には区切り文字が必要です。

トランザクションの開始時の動作については、端末管理 を参照してください。

3270 ディスプレイでこのオペランドを指定した場合、このトランザクション以外にオペレーターが開始できる CICS 機能は、ページング・コマンドと印刷要求のみです。

TRANSACTION属性を3270ディスプレイまたはSCSプリンターに 対して指定することはあまりありません。これは3601 論理装置ではオプションですが、3614 論理装置では必須です。

このオペランドが3790通信システムに対して指定され、 複数セッションが同じ3791の接続に使用される場合は、 すべてのセッションに対して同じTRANSACTION名を指定します。

TYPETERM(*typeterm*)

この TERMINAL 定義に関連付ける TYPETERM 定義の名前を指定します。名前の長さは、最大で 8 文字です。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

TYPETERM定義は、いくつかの類似の端末に対して多数の 属性を指定します。詳しくは、 277 ページの『TYPETERM 属性』 および Ways of defining CICS resources を参照してください。TYPETERM 属性は、すべての TERMINAL 定義で必須です。

この TERMINAL 定義をインストールする場合、TYPETERM 定義が既にインストールされている必要があります。

USEDFLTUSER ({**NO**|**YES**}) **(APPC のみ)**

各インバウンド接続 FMH に対して実行されるセキュリティー検査の種類を指定します。

NO 各インバウンド接続 FMH で、ATTACHSEC オプションに必要なフィールドが存在するかを検査することを示します。必須フィールドが存在しない場合、プロトコル違反メッセージが発行され、接続は失敗します。デフォルトはNOです。

YES 接続機能管理ヘッダー (FMH) の妥当性検査で一部の検査をバイパスすることを示します。 LU6.2 セキュリティーの実装 を参照してください。

USERID({**name**|***EVERY**|***FIRST**})

サインオンに使用され、セキュリティー・エラー・メッセージ、セキュリティー

違反メッセージ、および監査証跡で参照されるユーザー ID を指定します。これはセキュリティー・マネージャーに定義された有効なユーザー ID でなければなりません。

USERID 属性は、CESN を使用してサインオンできないプリンターなどの装置用にユーザー ID を指定する唯一の方法を提供します。ディスプレイ装置のUSERIDを指定することもできます。 その場合、ディスプレイは永続的にサインオンされます。 オペレーターはサインオンできません。保護リソースへのすべてのアクセスは、指定された USERID に対して RACF が許可した権限に依存します。

APPC 単一セッション端末の場合、USERID は、接続用に指定したすべての SECURITYNAME をオーバーライドします。

name この名前の長さは最大で 8 文字まで可能です。

許容文字:
A-Z 0-9 $ @ #
CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

***EVERY**

この特殊オペランドは、自動インストールされたコンソール専用です。これは、CICS は、MODIFY コマンドを受け取るたびに、MVS MODIFY コマンドで渡されたユーザー ID を使用することを意味します。コンソールへのサインオンでは、自動インストールされるコンソール用の事前設定ユーザー ID として MVS ユーザー ID を使用します。コンソールが削除されるか、別のユーザー ID による新たな MODIFY を受信するまでは、コンソールにはこのユーザー ID がサインオンした状態のままです。ユーザー ID なしで MODIFY コマンドを受信した場合、CICS は、有効なユーザー ID を持つ MODIFY コマンドを受信するまで、デフォルトの CICS ユーザー ID でサインオンします。非コンソール端末の場合、あるいはセキュリティーが有効にされていない場合は、この値は無視されます。

***FIRST**

この特殊オペランドは、自動インストールされたコンソール専用です。これは、CICS は、コンソールの自動インストールを要求する最初の MVS MODIFY コマンドで渡されたユーザー ID を使用することを意味します。コンソールへのサインオンには、事前設定ユーザー ID として MVS ユーザー ID が使用されます。コンソールが削除されるまでは、コンソールにはこのユーザー ID がサインオンした状態のままです。ユーザー ID なしで MODIFY コマンドを受信した場合、CICS は、デフォルトの CICS ユーザー ID でサインオンします。非コンソール端末の場合、あるいはセキュリティーが有効にされていない場合は、この値は無視されます。

この端末定義でコンソールが定義されている場合、ユーザー ID は、CONSOLE 一般リソース・クラス内の適切なプロファイルに対して許可されている必要があります。コンソールおよび端末の事前設定セキュリティーについては、エントリー・ポート・プロファイルの定義 を参照してください。

TRANCLASS リソース

TRANCLASS リソースは、トランザクション・クラスの特性を定義します。

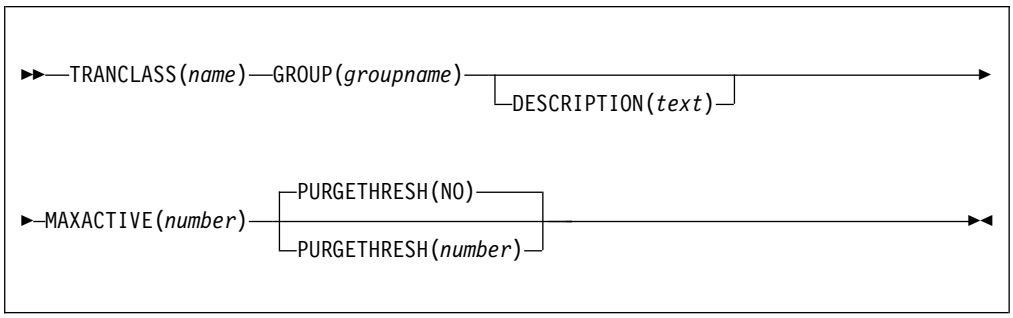
1 つのトランザクション・クラスに属するように定義されたトランザクションは、実行可能になる前に、スケジューリング制約に従います。アクティブなトランザクション・クラスに属するトランザクションが既に実行中の場合、新しいトランザクションはキューに入れられます。実行するトランザクションの最大数を指定するには、MAXACTIVE 属性を使用します。キューのサイズを制限するには、PURGETHRESH 属性を使用できます。

トランザクションをトランザクション・クラスに入れることにより、CICS によるタスクのディスパッチを制御できます。例えば、リソースの消費が激しいユーザーのトランザクションと、「おはようございます」のブロードキャスト・メッセージなど重要度がそれほど高くないトランザクションとに分離することができます。そのようにすれば、TRANCLASS 定義の属性を使用することによって、各トランザクション・クラスごとに許容されるアクティブ・タスクの数を制御できます。

BAS の作業については、BAS トランザクション・クラス・リソース定義の作業を参照してください。

TRANCLASS 属性

TRANCLASS リソースの構文および属性について説明します。



トランザクション・クラス定義属性の説明は以下のとおりです。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

MAXACTIVE(*number*)
このトランザクション・クラス内で、アクティブにすることができるトランザクションの最大数を指定します。トランザクション・クラスを定義する場合、0 から 999 の範囲で MAXACTIVE 値を指定する必要があります。

アクティブ・トランザクションの数が MAXACTIVE 制限に達したときに接続される新しいトランザクションはキューイングの対象となり、PURGETHRESH の制限の対象になります。

トランザクション・クラスの定義で MAXACTIVE 値にゼロを指定すると、すべてのタスクがキューに入れられます。

PURGETHRESH({**NO**|*number*})
これはトランザクション・クラス用のオプションのパージしきい値です。これは、トランザクション・クラスのメンバーシップについてキューイングしているトランザクションがパージされるしきい値を規定します。このトランザクション・クラス内でキューイングするトランザクションの数を制限するには、この値を指定します。これには、以下の値を指定することができます。

NO キューのサイズは無制限です（タスクの接続に使用できる 記憶域以外）。

number
1 から 1 000 000 の範囲のパージしきい値。

この値に 1 を指定した場合、トランザクションをキューに入れることはできません。それ以外の任意の数値 (*n*) を指定した場合、キューのサイズは、*number-1* に制限されます。 *n-1* の制限に達した後に接続された新規トランザクションは、すべてパージされます。

PURGETHRESH の例: 例えば、アクティブ・タスクの最大数 (MAXACTIVE) が 50 に設定され、トランザクション・キューイングを制限するパージしきい値 (PURGETHRESH) が 10 に設定されているトランザクション・クラスの場合、以下の時点で CICS はそのクラスの新しいトランザクションの異常終了を開始します。

- アクティブ・トランザクションの数が50に達し、
- トランザクション・クラスのメンバーシップに対するトランザクション・キューイングの数が 9 に達した場合。

CICS は、キューに入れられた数がキューの最大サイズ (この例では 9) を下回った場合にのみ、このトランザクション・クラスのキューに新しいトランザクションを受け入れます。

TRANCLASS(*name*)
トランザクション・クラスの名前を指定します。トランザクション・クラスに属するトランザクションは、実行可能になる前にスケジューリング制約に従いま

す。予約済み TRANCLASS 名 DFHTCL00 は、トランザクションがどのトランザクション・クラスにも属さないことを示すために使用されます。

TCLASS 属性をサポートするリリースとの互換性のために、CICS は TRANCLASS に相当する以下のものを提供します。

TCLASS	TRANCLASS
NO	DFHTCL00
1	DFHTCL01
2	DFHTCL02
3	DFHTCL03
4	DFHTCL04
5	DFHTCL05
6	DFHTCL06
7	DFHTCL07
8	DFHTCL08
9	DFHTCL09
10	DFHTCL10

これらのトランザクション・クラスのサンプル定義は、 グループDFHTCL内にあり、DFHLISTの一部として提供されます。

注: トランザクションが実行され、それに関連付けられた TRANCLASS 定義がインストールされていない場合、そのトランザクションは、TRANCLASS で指定されたスケジューリング制約なしで実行されます。注意メッセージ DFHXM0212 が発行されます。

TRANCLASSは、最大8文字です。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

TRANSACTION リソース

TRANSACTION リソースは、CICS によって提供される機能に関連するトランザクション属性を定義します。

CICS アプリケーションは、特定の機能を実行するように作成された 1 つ以上のプログラムから構成されます。そのようなアプリケーションの特定の呼び出しはトランザクション と呼ばれ、CICS トランザクション・マネージャーは、それをトランザクション ID (TRANSID) によって識別します。 TRANSACTION リソースをアプリケーション・エントリー・ポイントとして使用することもできます。エントリー・ポイントについては、アプリケーションのエントリー・ポイント で説明しています。

トランザクションの優先順位、セキュリティー・キー、トランザクション作業域 (TWA) の長さなどの情報を提供することによって、主に TRANSACTION 定義でトランザクションをどのように実行するかを CICS に指示します。この定義の名前 (TRANSACTION 名) は TRANSID と同じです。また、他のリソースとトランザクションをリンクさせるには、TRANSACTION 定義で他のリソースの定義の名前をコーディングします。そのようなリソースには、
PROGRAM、PROFILE、PARTITIONSET、REMOTESYSTEM、および TRANCLASS があります。

PROGRAM
PROGRAM 定義では、ご使用のアプリケーションのソフトウェア実装に関連するオプションを指定します。次に、トランザクションを処理するための制御を付与するプログラムが定義されます。 TRANSACTION 定義は、PROGRAM 定義を参照します。

PROFILE
端末や論理装置との対話を制御する属性をトランザクションごとに指定する必要はありません。代わりに、TRANSACTION 定義は、多くのトランザクションで属性を指定する PROFILE 定義を参照します。

REMOTESYSTEM
トランザクション・ルーティングでは、TRANSACTION 定義で PROGRAM 名を指定する代わりに、REMOTESYSTEM の名前を指定します。この値は、同じ名前の CONNECTION 定義および IPCONN 定義内でこの CICS システムに対して定義されている別の CICS システムの名前にすることができます。

CONNECTION 定義で REMOTESYSTEM を指定する場合、リモート・システムで実行するトランザクションの名前である REMOTENAME も指定することができます。リモート・システムは、どのプログラムに制御を行うかを決定します。

定義がインストールされているシステムに対応する REMOTESYSTEM 名を指定した場合、CICS は、ローカル・トランザクション・リソースをインストールします。 REMOTESYSTEM 名が、定義がインストールされているシステムに対応していない場合、CICS は、リモート・トランザクション・リソースをインストールします。 REMOTESYSTEM 名を指定しない場合、CICS は、ローカル・トランザクション・リソースをインストールします。

TRANCLASS
これは、トランザクションが属するトランザクション・クラスの名前を指定します。トランザクション・クラスに属するトランザクションは、実行可能になる前にスケジューリング制約に従います。この制約は、関連する TRANCLASS 定義で指定されます。 TRANCLASS 定義については、「TRANCLASS リソース」で説明します。

BAS の作業については、BAS トランザクション・リソース定義の作業を参照してください。

CICS バンドル内の TRANSACTION リソース

CICS バンドルを使用して、TRANSACTION リソース定義を作成、編集、およびインストールすることができます。この方法で TRANSACTION リソースを作成した場合、CICS バンドルを使用してそのリソースのライフサイクルを管理する必要があり、リソースを単独で管理することはできません。 **EXEC CICS INQUIRE TRANSACTION** コマンドまたは **CEMT INQUIRE TRANSACTION** コマンドを使用して、CICS バンドルによって動的に生成される TRANSACTION リソースを照会することができます。ただし、CICS バンドルによって動的に生成された TRANSACTION リソースに対して SET コマンドや DISCARD コマンドを発行することはできません。これらのコマンドは、BUNDLE リソースに対して発行する必要があり、CICS

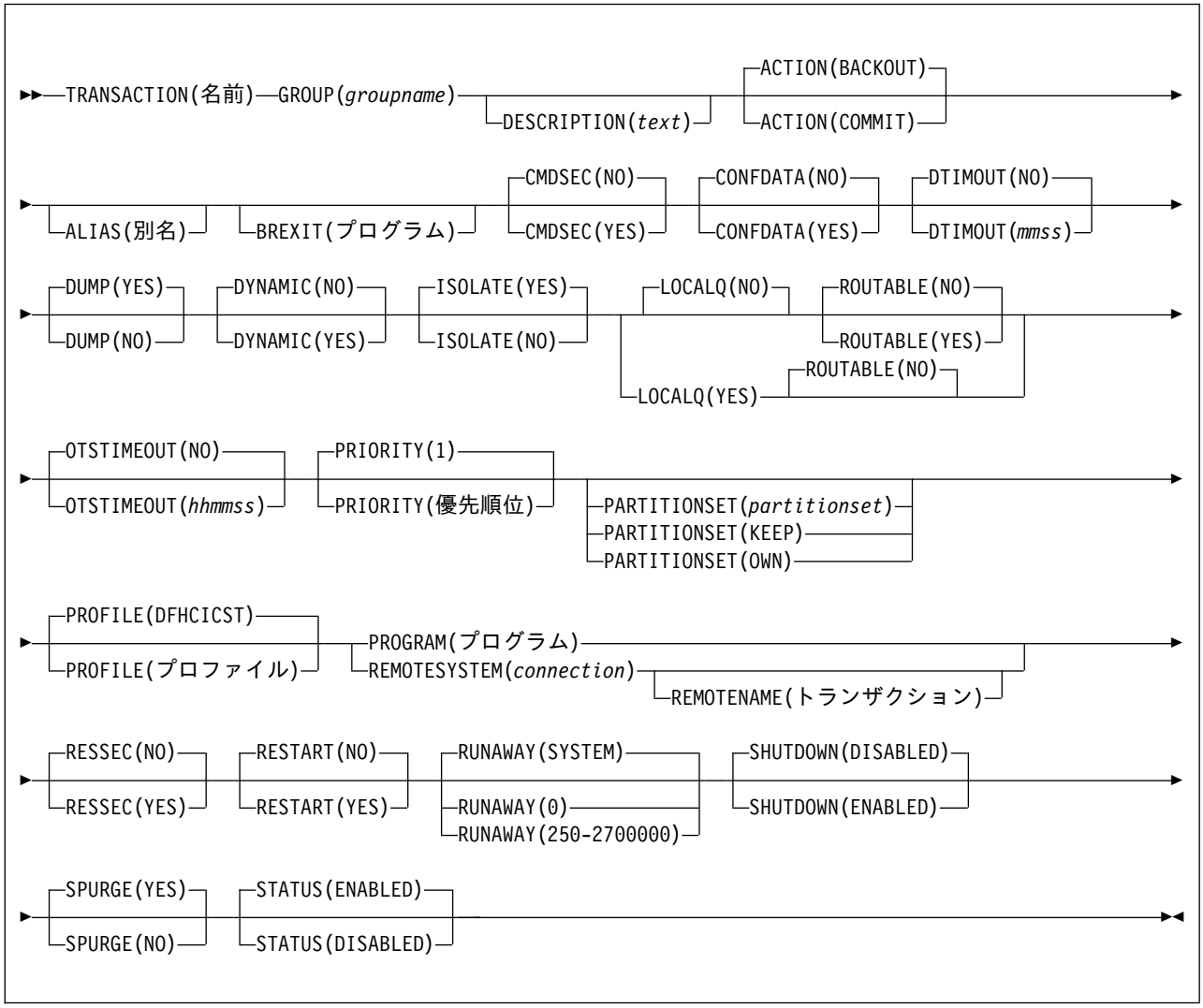
がそれらを TRANSACTION リソースに適用します。CICS バンドルを介して、TRANSACTION リソースをアプリケーション・エントリー・ポイントとして宣言することができます。

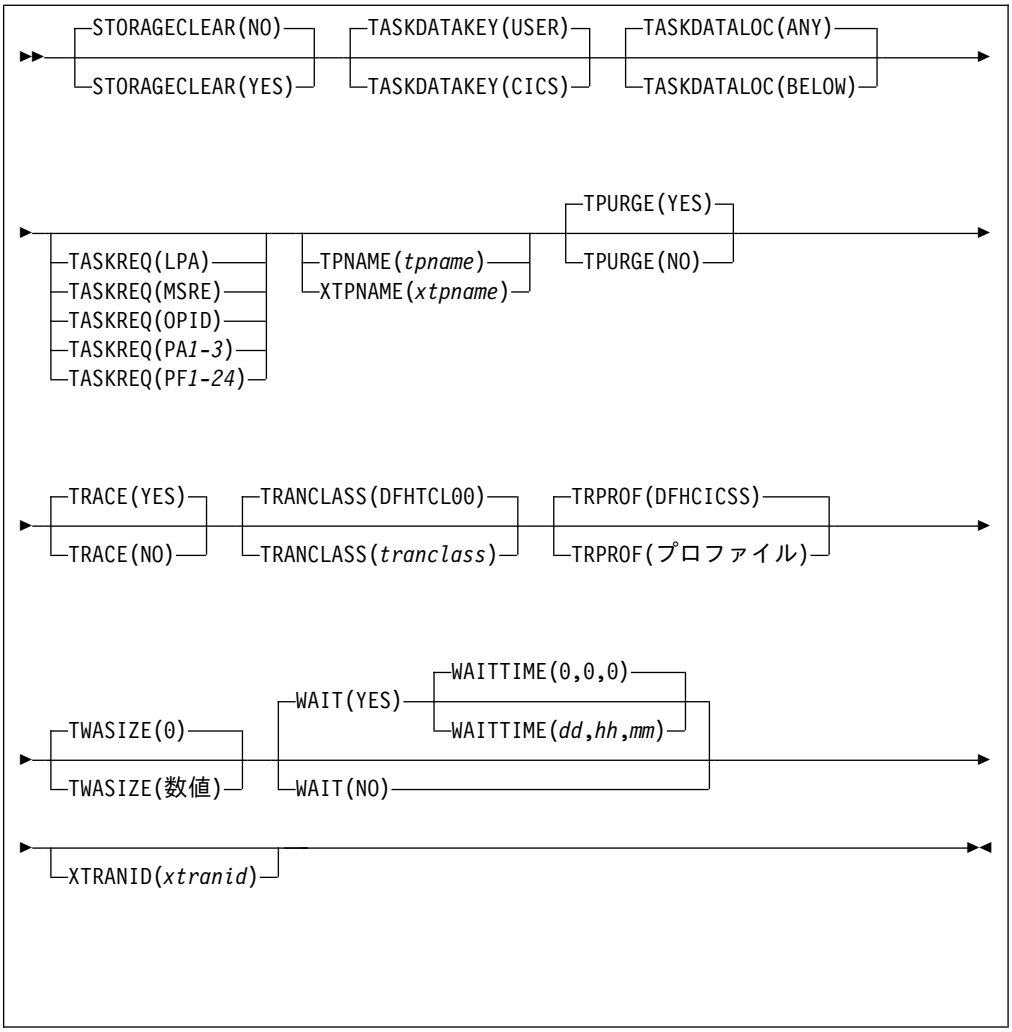
トランザクションには、スケジュールされた時間に **EXEC CICS START** コマンドによって開始される処理が割り振られている場合があります。その処理が、CICS バンドルが使用不可にされた後に開始するようにスケジュールされている場合、スケジュールされた処理は取り消され、そのトランザクションで新規の処理を開始することはできません。その後、CICS バンドルが再び使用可能になっても、取り消された処理のスケジュールが変更されることはなく、取り消されたままです。

CICS バンドル内のリソースの定義について詳しくは、Defining CICS bundlesを参照してください。

TRANSACTION 属性

TRANSACTION リソースの構文と属性を説明します。





ACTION({**BACKOUT**|**COMMIT**})

その作業単位が未確定期間に入った後の 2 フェーズ・コミット処理中に、CICS 領域に障害が発生した場合、あるいはそのコーディネーターとの接続が失われた場合に実行するアクションを指定します。このアクションは、WAIT 属性によって異なります。WAIT に YES を指定した場合、障害からリカバリーする前に WAITTIME が期限切れにならない限りは、ACTION は行われません。

WAIT に NO を指定した場合、実行されるアクションは以下のいずれかです。

BACKOUT
: リカバリー可能リソースのすべての変更がバックアウトされ、それらのリソースは UOW が開始される前の状態に戻されます。

COMMIT
: リカバリー可能リソースのすべての変更がコミットされ、UOW は完了と見なされます。

ALIAS(別名)

このトランザクションのトランザクション名に対して別名を指定します。この名前の最大長は 4 文字です。このオプションは、UCTRAN(NO) で定義した端末や、大/小文字混合入力が可能なトランザクション (PROFILE UCTRAN(NO))

に対して実行したい場合に有用です。例えば、ALIAS(abcd) を使用して ABCD を同じトランザクションを開始することができます。

ALIAS 属性が含まれる TRANSACTION 定義をインストールする場合、その結果は、別名がすでにシステムで使用されているかどうかによって異なります。

- 別名が 1 次トランザクション ID として使用されている場合、ALIAS 属性は無視されます。
- 別名が別のトランザクションの別名として使用されている場合、オリジナルの別名は新規別名によって置き換えられます。つまり、TRANSACTION 定義がインストールされると、その別名は元のトランザクションではなく、新規トランザクションを参照します。

BREXIT(プログラム)

BREXIT オプションで名前を指定しない **START BREXIT** コマンドを使用して 3270 ブリッジ環境でトランザクションが開始された場合に、このトランザクションと関連付けられるデフォルトのブリッジ出口の名前を定義します。名前の長さは、最大で 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

BREXIT が定義されている場合は、
REMOTESYSTEM、REMOTENAME、DYNAMIC(YES)、および
RESTART(YES) は無視されるので指定しないでください。

注: 3270 ブリッジを使用する方法として現在推奨されるのは、Link3270 メカニズムです。

CMDSEC({**NO**|**YES**})

システム・プログラミング・コマンドにセキュリティー検査を適用するかどうかを指定します。

NO 検査は行われません。コマンドは常に実行されます。

YES 外部セキュリティー・マネージャー (ESM) の呼び出しが行われます。CICS は、アクセスを許可または防止します。リソースまたはリソース・タイプを ESM が識別できない場合は、アクセスが回避されます。

CONFDATA({**NO**|**YES**})

CONFDATA システム初期設定パラメーターで HIDETC が指定されている場合に、CICS トレース項目からのユーザー・データを CICS が抑止するかどうかを指定します。システム初期設定パラメーターで CONFDATA=SHOW が指定されている場合、トランザクション定義の CONFDATA は無視されます。

システム初期設定パラメーターで CONFDATA=HIDETC が指定されている場合、以下のオプションが有効です。

NO CICS はユーザー・データを抑止しません。 z/OS Communications Server および MRO の初期ユーザー・データは、トレース・ポイント AP FC92 でトレースされます。 FEPI ユーザー・データは、通常の

CICS FEPI トレース・ポイントでトレースされます。 IPIC ユーザー・データは、通常の CICS IS トレース・ポイントでトレースされます。

YES CICS は、CICS トレース・ポイントからのユーザー・データを抑止します。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

DTIMOUT({**NO**|*mmss*})
デッドロック・タイムアウトをタスクに適用するかどうかを指定します。タスクが (ストレージの不足などによって) 中断状態になった場合、タスクの中断状態が DTIMOUT 値より長く続くと、タスクのパージが開始されます。パージによってトランザクションが異常終了した場合、使用される異常終了コードは、CICS のどの部分がタスクを中断状態にしたかによって異なります。 CEDF を使用する場合、ユーザー・タスクには、可能であれば DTIMOUT(NO) または大きい値を指定します。現在、この値は、DTIMOUT がゼロ以外である場合のすべての RLS ファイル要求に対するタイムアウトとしても使用されます。DTIMOUT がゼロである場合は、要求は SIT FTIMEOUT 値を取得します。FTIMEOUT は、デッドロック・タイムアウト間隔がアクティブではないトランザクションに適用されます。 TRANSACTION 定義の DTIMOUT キーワードに時間値が指定された場合、この値がそのトランザクションのファイル・タイムアウト値として使用されます。

CEDF を使用する場合、ユーザー・タスクに DTIMOUT 値が指定されていても、ユーザー・タスクが中断状態になり、CEDF タスクがアクティブになっている間は、その DTIMOUT 値は無視されます。そのため、中断状態のユーザー・タスクは、CEDF タスクがユーザー応答を待っている間は、デッドロック・タイムアウト (異常終了 AKCS) によって終了することができません。

RLS 以外の使用で DTIMOUT を有効にするには、SPURGE を YES に設定します。

CICS は、特定の時点でデッドロック・タイムアウトを禁止します。

DTIMOUT は、端末入出力の待機ではトリガーされません。中継トランザクションは、セッションの取得後にリソースにアクセスしないため、中断状態の割り振り要求をトラップする以外には DTIMOUT の必要はほとんどありません。ただし、セッションでの入出力待機のために、MRO セッションおよびマップ済み APPC 接続でのトランザクション・ルーティングに対する RTIMOUT 属性を PROFILE 定義で指定することができます。

デッドロック・タイムアウトは、CICS がストレージ不足 (SOS) 状態を処理するメカニズムであるため、一部のトランザクションでは DTIMOUT を定義する必要があります。

NO デッドロック・タイムアウト機能は不要です。

mmss デッドロック・タイムアウトによって中断状態のタスクが終了されるま

での時間の長さ (MMSS は分と秒を表します)。指定できる最大値は 68 分です。この値の精度は 1 秒単位です。

DUMP({**YES**|**NO**})

トランザクションが異常終了した場合に、ダンプ・ドメインに対する呼び出しを行ってトランザクション・ダンプを生成するかどうかを指定します。

このオペランドは、以下のダンプ操作では無効です。

- **EXEC CICS DUMP** コマンド。このコマンドは、常にダンプを生成します。
- CICS が異常終了 ASRA、ARSB、および ASRD で生成するダンプ・コード AP0001 および SR0001 のシステム・ダンプ。トランザクションの DUMP 属性に NO を指定した場合、CICS は、トランザクション・ダンプを抑止しますが、システム・ダンプは抑止しません。

YES CICS は、ダンプ・ドメインを呼び出してトランザクション・ダンプを生成します。トランザクション・ダンプの最終的な生成または抑止は、トランザクション・ダンプ・テーブルによって制御されるので注意してください。ダンプ・テーブルについて詳しくは、指定可能なダンプ・コード・オプション を参照してください。

トランザクションの異常終了時に、指定されたダンプ・コードにトランザクション・ダンプ・テーブル・エントリーが存在しない場合、CICS は、デフォルトでトランザクション・ダンプを生成する一時エントリーを作成します。

トランザクション・ダンプのダンプ・テーブル・エントリーを制御するには、**CEMT SET TRDUMPCODE** コマンドまたは SET TRANDUMPCODE SPI コマンドを使用します。

NO ダンプ・ドメインに対して呼び出しは行われず、すべての潜在的なトランザクション・ダンプが抑制されます。

DYNAMIC({**NO**|**YES**})

CICS 動的トランザクション・ルーティング機能を使用して、トランザクションをリモート領域に動的にルーティングできるかどうかを指定します。

NO REMOTESYSTEM 属性に従って、ローカル定義またはリモート定義を作成します。

YES 動的トランザクション・ルーティング・プログラムに対し、呼び出し時にローカル状況またはリモート状況を動的に決定することを許可します。動的トランザクション・ルーティング・プログラムに関するプログラミング情報については、動的ルーティング・プログラムの書き込み を参照してください。

EXTSEC

この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

GROUP(*groupname*)

各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字:

A-Z 0-9 $ @ #

小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

INDOUBT

この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

ISOLATE({YES|NO})

CICS がトランザクションのユーザー・キーのタスク存続期間ストレージを分離して、トランザクション間保護を提供するかどうかを指定します。 (ユーザー・キー・ストレージの説明については、TASKDATAKEY 属性を参照してください。) 分離とは、ユーザー・キー・ タスク存続時間ストレージが他のトランザクションのユーザー・ キー・プログラム(EXECKEY(USER)で定義されたプログラム）による 読み書きに対して保護されることを意味します。ストレージ・アクセスの ISOLATE 属性の効果に、ISOLATE 属性の効果を示しています。

注:

1. ISOLATE 属性は、CICS キーで実行されているアプリケーション・プログラムに対する (つまり、EXECKEY(CICS) で定義されたプログラムからの) 保護を提供しません。
2. VSAM 非共用リソース (NSR) は、トランザクション分離を使用するトランザクションではサポートされません。 NSR を使用して VSAM ファイルにアクセスするトランザクションを定義する場合は、ISOLATE(NO) を指定する必要があります。ファイル要求をリモート領域に機能シップすることもできます。要求を実行する DFHMIRS プログラムは、CICS の EXECKEY を使用して定義されます。トランザクション分離がアクティブであるかどうかに関係なく、CICS キー・プログラムは、自身のタスクと他のすべてのタスクの CICS キーおよびユーザー・キー・ストレージへの読み取りおよび書き込み権限を持っています。
3. トランザクション分離は、64 ビット・ストレージには適用されません。

YES トランザクションのユーザー・キーのタスク存続期間ストレージは、他のすべてのトランザクションのユーザー・キー・プログラム (つまり EXECKEY(USER) で定義されたプログラム) からは分離されていますが、EXECKEY(CICS) で定義されたプログラムからは分離されていません。

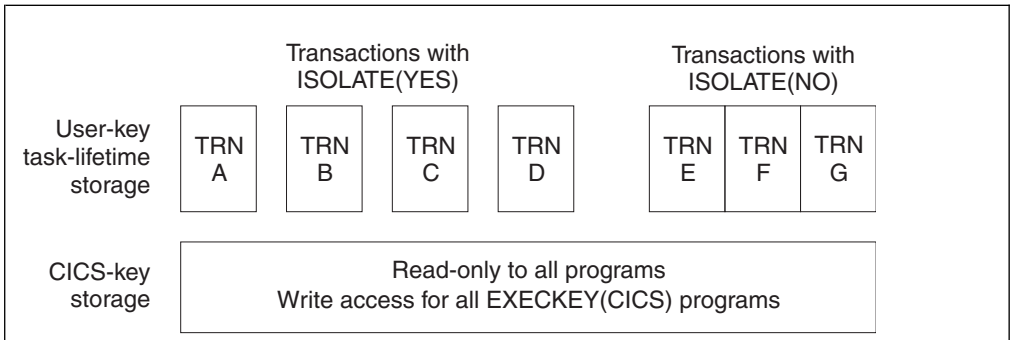
また、他のすべてのトランザクションにおけるユーザー・キーのタスク存続時間ストレージは、ISOLATE(YES) で定義されたトランザクションのユーザー・キー・プログラムから保護されます。

NO ISOLATE(NO) を指定した場合、トランザクションのタスク存続時間ストレージは、ISOLATE(YES) で定義されたトランザクションのユーザー・キー・プログラムから分離されます。ただし、トランザクション・ストレージは、同じく ISOLATE(NO) を指定する他のトランザクショ

ンのユーザー・キー・プログラムからは分離されません。このオプションを使用すると、トランザクションがすべて共通サブスペースに割り振られるためです。

また、ISOLATE(YES) で定義されたトランザクションすべてのユーザー・キーのタスク存続時間ストレージは、ISOLATE(NO) で定義されたトランザクションのユーザー・キー・プログラムから保護されることにも気をつけてください。

ユーザー・キーのタスク存続時間ストレージの一部を共用するトランザクションに、ISOLATE(NO) を指定します。



この図は、ISOLATE(YES) で定義された 4 つのトランザクション (A、B、C、D) と、ISOLATE(NO) で定義された 3 つのトランザクション (E、F、G) を示しています。

- トランザクション A、B、C、D それぞれのユーザー・キーのタスク存続時間ストレージは、他のトランザクションすべてから分離されます。
- トランザクション E、F、G のユーザー・キーのタスク存続時間ストレージは、トランザクション E、F、G のユーザー・キー・アプリケーション・プログラムのすべてからアクセス可能ですが、トランザクション A、B、C、D のユーザー・キー・プログラムからは分離しています。
- すべてのトランザクションが、CICS キー・ストレージに対する読み取り専用アクセス権を持っています。

図 *5.* ストレージ・アクセスの *ISOLATE* 属性の効果

LOCALQ({NO|YES})

ローカル・システムに対するキューイングを実行するかどうかを指定します。

NO ローカル・キューイングは実行されません。

YES システムおよび IPIC 接続が使用可能ではなく、システム名が有効である場合に、NOCHECK オプションを指定した START コマンドの要求に対してローカル・キューイングを試行することができます。

以下の状況では、システムは使用不可として定義されます。

- 要求が開始されたときに、システムがサービス休止である場合。
- リモート・システムへのセッションの開始試行が失敗し、異常条件プログラム (DFHZNAC) またはノード・エラー・プログラム (DFHZNEP) によりとられる修正措置が、システムをサービス休止にした場合。

- リモート・システムへのセッションで直ちに使用可能なものがなく、ご使用の XISCONA グローバル・ユーザー出口プログラムで、その要求を発行領域でキューイングしないと指定している場合。

以下の状況では、IPIC 接続は使用不可として定義されます。

- IPIC 接続が獲得されなかった場合。
- セッションが使用できないため、CICS で新しいセッションの要求をキューに入れることができなかった場合。

ローカル・キューイングは、時間に依存しない要求を表す START コマンドの場合にのみ使用します。ローカル・キューイングによって暗黙的に示される遅延は、要求が開始される時間に影響します。この条件が設定されていることの確認は、ユーザー自身で行うことになります。LOCALQ(YES) を指定する場合、ROUTABLE(YES) は指定できません。

グローバル・ユーザー出口 XISLCLQ または XISQLCL を使用して、LOCALQ 属性の設定をオーバーライドすることができます。システム間通信プログラムにおけるユーザー出口に関するプログラミング情報については、グローバル・ユーザー出口ルーチンのポイント を参照してください。

OTSTIMEOUT({**NO**|*hhmmss*})

エンタープライズ Bean 環境で作成され、この CICS トランザクションの下でタスクとして実行されるオブジェクト・トランザクション・サービス (OTS) トランザクションを、OTS トランザクションのイニシエーターが同期点を取得するか、あるいはトランザクションをロールバックする必要が生じるまでに実行できる時間の長さを、時間、分、秒で指定します。指定された期間が経過すると、CICS はタスクをパージします。

OTS トランザクションのイニシエーターには、以下のいずれかを使用することができます。

- エンタープライズ Bean のクライアント。
- EJB コンテナー。コンテナーは、Bean メソッドの最後に同期点を発行します。
- 独自の OTS トランザクションを管理するセッション Bean。

独自の OTS トランザクションを管理するセッション Bean のメソッドは、`javax.Transaction.UserTransaction` インターフェースの `setTransactionTimeout` メソッドを使用することにより、デフォルトのタイムアウト値を指定変更できます。

NO OTS トランザクションはタイムアウトになりません。デフォルトはNOです。

hhmmss
タスクがパージされるまでの時間 (HHMMSS 形式)。最大期間は 24 時間 (240000) です。

PARTITIONSET({*partitionset*|**KEEP**|**OWN**})

デフォルトのアプリケーション区分セットとなる区分セットの名前を指定します。 名前の長さは、最大で 8 文字です。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

区分セット名およびいずれの予約名も指定しない場合、CICS は、トランザクションから端末への最初の BMS 出力の前に既存の区分画面を破棄します。

区分セット

CICS は、トランザクションから端末への最初の BMS 出力の前に、既存の区分画面を破棄し、指定された区分セットをロードします。 (端末区分セットがアプリケーション区分セットと一致する場合、既存の区分画面は破棄されません。)

この名前は、PROGRAM(名前) で指定される名前と同じものではいけません。

KEEP トランザクションは、この端末にアプリケーション区分セットを使用します。通常、このオプションは、疑似会話型トランザクションのチェーン内の後続トランザクションに使用されます。

OWN トランザクションは自身の区分管理を実行します。

PRIMEDSIZE

この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

PRIORITY({**1**|*priority*})

トランザクションの優先度を指定します。この 1 から 3 桁の 10 進値 (0 から 255) は、トランザクション処理全体の優先順位を設定するために使用されます。トランザクション処理の優先度は、端末優先度、トランザクション優先度、およびオペレーター優先度の合計と等しく、255 を超えてはなりません。数値が高いほど、優先度は高くなります。

PROFILE({**DFHCICST**|プロファイル})

トランザクションを開始した端末と一緒に使用される処理オプションを指定する、PROFILE 定義の名前。

許容文字:

A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

デフォルトは DFHCICST です。

デフォルトの DFHCICST により提供される処理オプションを、グループ DFHISC 内の PROFILE 定義に示します。 DFHCICST は、分散プログラム・リンクでの使用には適していません。代わりに DFHCICSA を指定してください。これには、INBFMH=ALL があります。

PROGRAM(プログラム)

CICS がこのトランザクションを処理するための制御を付与するプログラムの名前を指定します。名前の長さは、最大で 8 文字です。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

この名前は、PARTITIONSET(名前) で指定された名前とは異なるものにしてください。

REMOTESYSTEM に指定された名前が現行システムの名前と異なっている場合、PROGRAM の名前を指定する必要はありません。そのような状況で、PROGRAM の名前を指定した場合、その名前は無視される可能性があります。

このトランザクション定義をリモート・プログラム・リンク要求に使用する場合、この属性で指定するプログラム名は、CICS ミラー・プログラム DFHMIRS の名前でなければなりません。PROGRAM 属性の PROGRAM 定義での TRANSID 属性を参照してください。

REMOTENAME(トランザクション)
このトランザクションがリモート・システム内またはシステム間通信を使用する領域内で実行される場合に、リモート・システム内でこのトランザクションを認識するための名前を指定します。リモート・システムには、別の CICS 領域または IMS システムを使用することができます。REMOTENAME は、REMOTESYSTEM 属性で別の CICS 領域が指定されている場合は 1 から 4 文字の長さ、REMOTESYSTEM で IMS システムが指定されている場合は 1 から 8 文字の長さで指定することができます。IMS は、8 文字の名前を使用します。REMOTENAME が 8 文字未満の場合、IMS はその名前を使用可能な形式に変換します。

許容文字:

A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

REMOTESYSTEM を指定して REMOTENAME を省略した場合、REMOTENAME の値はデフォルトでローカル名 (つまり、この定義の TRANSACTION 名) になります。トランザクションは、リモート・システムあるいはリモート領域にある必要はありません。

REMOTESYSTEM(接続)
トランザクション接続要求の送信に使用する相互通信リンクを示す名前を指定します。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

REMOTESYSTEM は、以下の ID のいずれかになります。

- IPIC 接続の場合、サービス中で獲得済みの IPCONN 定義における IPCONN 名の最初の 4 文字です。

- MRO および APPC 接続の場合、CONNECTION 定義の CONNECTION 名。

REMOTESYSTEM は、CICS 機能要求シップ (非同期処理およびトランザクション・ルーティング) に使用されます。

トランザクションのセットに対して指定された REMOTESYSTEM オプションが、そのセット内の他のトランザクションを参照していない (自己参照ではない) ことを確認してください。そのようになっていると、複数のミラー・トランザクションの開始がループする原因となる可能性があります。

RESSEC({NO|YES})

このトランザクションによってアクセスされるリソースに、リソース・セキュリティー検査を使用するかどうかを指定します。

NO このトランザクションを使用する権限を持つユーザーに対して、すべてのリソースが使用可能になります。

YES 外部セキュリティー・マネージャーを使用します。外部セキュリティー検査について詳しくは、リソース定義のセキュリティー を参照してください。

RESTART({NO|YES})

異常終了して、その後に動的トランザクション・バックアウト機能によってバックアウトされたタスクを、トランザクション再始動機能を使用して再始動するかどうかを指定します。

RESTART(YES) を指定した場合、失敗したタスクは、初期プログラムの最初から再始動されます。動的トランザクション・バックアウトが失敗した場合、あるいは再始動が動的に抑止された場合は、DFHPEP が通常の方法で呼び出されます。トランザクション再始動機能は、タスクを手動で再実行依頼する代わりに自動的に再始動できるプログラム分離デッドロックなどの状態のときに特に役立ちます。端末開始トランザクションは、SHUTDOWN(DISABLED) が指定されている場合でも、CICS のシャットダウン中に再始動することができます。自動トランザクション再始動について詳しくは、アプリケーションをトランザクションに分割する を参照してください。

NO 再始動機能が使用されなくなります。

YES 再始動機能を使用します。

ROUTABLE({NO|YES})

トランザクションが適格な **EXEC CICS START** コマンドの対象である場合に、そのトランザクションが拡張ルーティング方式を使用して経路指定されるかどうかを指定します。

NO トランザクションが START コマンドの対象である場合、従来型方式を使用して経路指定されます。

YES トランザクションが適格な START コマンドの対象である場合に、拡張方式を使用して経路指定されます。

ROUTABLE(YES) を指定する場合、LOCALQ(YES) は指定できません。

EXEC CICS START コマンドによって呼び出されるルーティング・トランザクションの拡張方式および従来型方式について詳しくは、START コマンドで呼び出されたトランザクションのルーティング を参照してください。

RSL
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

RUNAWAY({SYSTEM|0|250-2700000})
このトランザクション定義の下で実行しているタスクが、ランナウェイ状態 (論理ループ) になったと想定されるまで、プロセッサーの制御を保持できる時間 (ミリ秒)。この期間を経過すると、CICS はタスクを異常終了することができます。

SYSTEM
CICS は、このトランザクションのランナウェイ時間制限として、ICVR システム初期設定パラメーター値を使用します。

0
トランザクションに制限は適用されず、ランナウェイ・タスク検出は不要です。

250-2700000
ランナウェイ時間制限 (250 から 2700000 の範囲)。タスクがランナウェイ状態であるかを検査する際に、CICS は、ユーザーが指定した値を 250 の倍数に切り捨てます。

SHUTDOWN({DISABLED|ENABLED})
すべてのトランザクションに適用され、CICS のシャットダウン中にトランザクションを実行できるかどうかを指定します。このオプションは、**PERFORM SHUTDOWN** コマンドの XLT オプションを補足します。シャットダウン中にトランザクションを接続するには、そのトランザクションが SHUTDOWN (ENABLED) として定義されている必要があります。あるいは、端末ベースのトランザクションである場合は、**PERFORM SHUTDOWN** コマンドで指定した XLT で指定されている必要があります。

DISABLED
トランザクションは、CICS のシャットダウン中に実行することができません。

ENABLED
トランザクションは、CICS のシャットダウン中に実行することができます。

SPURGE({YES|NO})
トランザクションの初期状態が**システム・パージ可能**であるかどうかを指定します。

SPURGE=NO の場合、以下の方法でトランザクションのパージが回避されます。

- デッドロック・タイムアウト (DTIMOUT)
- **EXEC CICS ... PURGE** コマンド
- ノード・エラー・プログラム (NEP) での TWAOCT (取り消しタスク) の設定
- **CEMT SET ... PURGE** コマンド

SPURGE=YES を指定すると、ユーザーが関連する範囲内であれば、そのようなパージを行うことができます。ただし、トランザクションが到達した時点でパージを許可することが安全ではない場合は、CICS がパージを防止することもあります。

SPURGE=NO は、読み取りタイムアウト (RTIMOUT) 機能、EXEC CICS SET ... FORCEPURGE コマンド、あるいは CEMT SET TRANSACTION(tranid) FORCEPURGE コマンドによってトランザクションのパージを防止するわけではないので注意してください。 SPURGE は初期値を決定するだけで、この値は、実行中にトランザクションによって変更することができます。

YES トランザクションは、初期設定でシステム・パージ可能になります。

NO トランザクションは、初期設定でシステム・パージ可能にはなりません。

STATUS({ENABLED|DISABLED})
トランザクション状況を指定します。

ENABLED
通常のトランザクションの実行を許可します。

DISABLED
トランザクションの実行を許可しません。

この属性は、CICS バンドルによって動的に生成された TRANSACTION リソースでは無視されます。 TRANSACTION リソースの初期状況は、リソースを定義するバンドルの初期状況から導出されます。

STORAGECLEAR({NO|YES})
このトランザクションのタスク存続期間ストレージを解放時にクリアするかどうかを指定します。 このトランザクションによってタスク存続期間ストレージに保管された機密データや重要データを他のタスクが偶発的に参照することを防ぐには、STORAGECLEAR を使用します。

TASKDATAKEY({USER|CICS})
タスク期間中のタスク初期設定時に CICS が割り振るタスクのストレージである、タスク存続期間ストレージ (アプリケーションからアクセス可能) のストレージ・キーを指定します。これらのストレージ域は、EXEC インターフェース・ブロック (EIB) およびトランザクション作業域 (TWA) です。

TASKDATAKEY は、トランザクションの下で実行されるすべてのプログラムのために CICS が取得するストレージのキーも指定します。指定されたキーに CICS が割り振るプログラム関連のストレージには、以下の項目が含まれます。

- アプリケーション・プログラムの実行ごとに CICS が取得する作業用ストレージのコピー。
- 暗黙的および明示的な GETMAIN 要求に対する応答として CICS がプログラム用に取得するストレージ。例えば、プログラムは、GETMAIN コマンドまたは GETMAIN64 コマンドによって、あるいは他の CICS コマンドでの SET オプションの結果として、ストレージを要求することができます。

トランザクション内のプログラムのいずれかが EXECKEY(USER) で定義されている場合は、TASKDATAKEY(USER) を指定する必要があります。トランザク

ションに TASKDATAKEY(CICS) を指定した場合、このトランザクションにおいてユーザー・キー内でプログラムを実行しようとすると、タスクは異常終了コード AEZD で異常終了します。

USER CICS は、このトランザクション用のユーザー・キー・ストレージを取得します。いずれかのキーで実行されているアプリケーション・プログラムは、これらのストレージ域を読み取ることも変更することもできます。

ISOLATE(YES) で定義されたトランザクションのユーザー・キー・プログラムは、自身のタスクのユーザー・キーのタスク存続時間ストレージに対してのみアクセス権があります。

ISOLATE(NO) で定義されたトランザクションのユーザー・キー・プログラムは、ISOLATE(NO) で定義されたタスクのユーザー・キーのタスク存続時間ストレージに対してもアクセス権があります。

タスク・ストレージ保護について詳しくは、PROGRAM 定義の EXECKEY 属性の説明を参照してください。

CICS CICS は、このトランザクション用の CICS キー・ストレージを取得します。 CICS キーで実行されているアプリケーション・プログラムは、これらのストレージ域を読み取ることも変更することもできます。ユーザー・キーで実行されているアプリケーション・プログラムは、これらのストレージ域を読み取ることも変更することもできます。

TASKDATALOC({ANY|BELOW})

トランザクションの期間中に CICS が取得したタスク存続期間ストレージを、仮想記憶域の 16 MB 境界よりも上に配置できるかどうかを指定します。特定の CICS タスクに関連しているこれらの領域には、EXEC インターフェース・ブロック (EIB) およびトランザクション作業域 (TWA) が含まれます。

トランザクションを形成するプログラムのいずれかが 24 ビットのアドレッシング・モードで実行される場合、TASKDATALOC(BELOW) を指定する必要があります。この制限は、トランザクションに代わって実行しているタスク関連のユーザー出口にも適用されます。

これらの条件のいずれも満たさないトランザクションでは、ANY を指定して、関連付けられた仮想記憶域制約解放を取得できます。

CICS は、TASKDATALOC(ANY) の使用 (特に以下のアクション) をモニターします。

- TASKDATALOC(ANY) で定義されたトランザクションの下で実行されている AMODE 24 プログラムを呼び出そうとすると、AEZC 異常終了が発生します。
- TASKDATALOC(ANY) を指定した AMODE(24) の実行中に、EXEC CICS コマンドの実行またはタスク関連のユーザー出口の呼び出しを行おうとすると、AEZA 異常終了が発生します。
- TASKDATALOC(ANY) を指定したトランザクションとして AMODE 31 プログラムを実行すると、強制的に AMODE(24) を実行するタスク関連ユーザー出口が呼び出され、AEZB 異常終了が発生します。

- AMODE 24 で強制的に実行されているタスク関連ユーザー出口が、タスク開始時に使用可能になった場合、CICS は、残りの CICS 実行のすべてのトランザクションに対して強制的に TASKDATALOC(BELOW) を実行します。

ANY トランザクション用に CICS が取得するストレージ域は、仮想記憶内の 16 MB 境界より上に配置することができます。

BELOW
トランザクション用に CICS が取得するストレージ域は、16 MB 境界より下に配置する必要があります。

TASKREQ(値)
トランザクションを開始する方法として、PF キーを押すこと、ライト・ペンを使用すること、またはカードを使用することのいずれかを指定します。使用可能な値は以下のとおりです。

- PAキーの場合はPA1、PA2、またはPA3。
- ファンクション・キーの場合は、PF1 から PF24 です。
- オペレーター ID カード・リーダーについては、OPID です。
- 3270 装置上のライト・ペン検出可能フィールドについては、LPA です。
- 10/63 文字磁気スロット読取装置については、MSRE です。

PA キーおよび PF キーの場合、以下に注意してください。

- PA キーまたは PF キーを PRINT システム初期設定パラメーターで指定した場合、トランザクションを開始するための TASKREQ として同じ PF キーを使用することはできません。
- PA キーまたは PF キーをページ検索キーとして、SKR*xxxx* システム初期設定パラメーターで指定した場合、ページ検索セッション中にはページ検索キーとして解釈されます。これ以外の場合には、同じキーを使用してトランザクションを開始することができます。以下の値を指定して、キーを定義します。

```
TASKREQ=KEY-ID
PROGRAM=DFHTPR
TWASIZE=1024
TPURGE=NO
SPURGE=NO
```

- PROGRAM(DFHTPR) を指定してトランザクションを定義し、TASKREQ を定義した場合、キーはトランザクションを開始するのと同時にページ検索セッションを開始します。

TCLASS
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

TPNAME(名前)
TRANSACTION 属性の長さに対する 4 文字の制限が制約的すぎる場合に、APPC パートナーが使用できるトランザクションの名前を指定します。この名前の長さは最大 64 文字です。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

この範囲の文字では指定したい名前に不十分である場合は、TPNAME の代わりに XTPNAME 属性を使用することができます。

TPURGE({**YES**|**NO**})
端末エラーによってトランザクションをパージできるかどうかを指定します (非 z/OS Communications Server 端末のみ)。

YES 端末エラーが生じたときに、タスクをパージできます。

NO 端末エラーが生じたときに、タスクをパージできません。これが起こった場合は、マスター端末オペレーターによる手操作による介入が必要です。

TRACE({**YES**|**NO**})
このトランザクションのアクティビティーをトレースするかどうかを指定します。

YES このトランザクションのアクティビティーをトレースします。

NO このトランザクションのアクティビティーをトレースしません。

CICS により提供される CEDF および CSGM 用トランザクション定義では、TRACE(NO) を指定します。

TRANCLASS(**DFHTCL00**|*tranclass*)
トランザクションが属するトランザクション・クラスの名前を指定します。トランザクション・クラスに属するトランザクションは、実行可能になる前にスケジューリング制約に従います。予約済み TRANCLASS 名 DFHTCL00 は、トランザクションがどのトランザクション・クラスにも属さないことを示すために使用されます。

トランザクションが実行され、それに関連付けられた TRANCLASS 定義がインストールされていない場合、そのトランザクションは、TRANCLASS で指定されたスケジューリング制約なしで実行されます。メッセージ DFHXM0212 が、警告として発行されます。

TRANCLASS の長さは、最長で 8 文字となります。

| 許容文字:
| A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

TRANSACTION(名前)
トランザクションの名前またはトランザクション ID (TRANSID) を指定します。 名前は最大4文字です。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

C で始まる名前は避けることをお勧めします。この文字は CICS 用に予約されています。新たに機能やフィックスをアップグレードしたり追加したりしたときに、そのようなリソースが CICS によって削除または破棄され、CICS リソースに置き換えられる可能性があります。

注:

1. 名前にコンマ (,) を使用すると、以下のようなコマンドを使用できなくなります。

   ```
   CEMT INQUIRE TRANSACTION(value1,value2)
   CEMT SET     TRANSACTION(value1,value2)
   ```

   これらのコマンドでは、コンマがリスト区切り文字として機能します。リソース ID のリストの使用については、リソース ID のリスト を参照してください。
2. RACF を使用して一時データ・キューを保護する場合は、% および & を名前に使用しないでください。 RACF コマンドは、これらの文字がプロファイル名で使用された場合に、特殊な意味を割り当てます。 トランザクション・セキュリティー を参照してください。

トランザクションID内で他の特殊文字を使用する 場合、XTRANID属性を使用して、トランザクション の開始に使用できる別の名前を 指定します。CSD ファイル上では TRANSACTION 定義を認識するために TRANSACTION 名を使用するため、この名前も指定する必要があります。

トランザクションを定義する場合は、PROGRAM または REMOTESYSTEM も指定する必要があります。

TRANSEC
この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

TRPROF({DFHCICSS|プロファイル})
ISC トランザクション・ルーティングの際にシステム間フローを送達するセッションのプロファイルの名前を指定します。名前の長さは、最大で 8 文字です。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

このオプションは、リモート・トランザクションに対してのみ指定することができます。

TWASIZE({0|数値})
このトランザクション用に取得するトランザクション作業域のサイズ (バイト数) を指定します。 0 から 32767 の範囲で 1 から 5 桁の 10 進値を指定します。

注:

1. TWASIZE が小さすぎると、ストレージが破損する可能性があります。
2. CICS 提供のトランザクションについては、TWASIZE を変更しないでください。

WAIT({YES|NO})
作業単位 (UOW) が未確定状態に入った後に発生する障害からのリカバリーを保留にして、未確定 UOW を待機させるかどうかを指定します。

CICS は、INDOUBT(WAIT) を使用した古いスタイルのトランザクション定義を受け入れ、WAIT(YES) ACTION(BACKOUT) として解釈します。

YES UOW は、障害のリカバリーを保留しながら待機して、その未確定状態を解決し、リカバリー可能リソースをバックアウトするかコミットするかを決定します。 つまり、UOW は棚上げされるわけです。

リカバリー可能リソースには、以下のいずれかを含めることができます。

- DBCTL データベース
- Db2 データベース
- 一時記憶域キュー
- TDQUEUE 定義で WAIT(YES) が指定された、論理的にリカバリー可能な区画内一時データ・キュー
- VSAM データ・セット
- BDAM データ・セット

WAIT(YES) オプションは、以下のいずれにも該当しない場合に有効になります。

- トランザクションに、バックレベル・システムへの従属 MRO セッションがある。
- トランザクションに LU6.1 従属セッションがある。このコンテキストでは、LU6.1 IMS セッションは従属ではないので注意してください。
- トランザクションに複数のセッションがあり、そのコーディネーター・セッションがバックレベル・システムまたは LU6.1 向けのものである。
- トランザクションに接続されたタスク関連ユーザー出口が、CICS 未確定プロトコルをサポートしていない。

上記のいずれの例外にも該当しないが、CICS Transaction Server for z/OS 未確定アーキテクチャーを使用しない CICS Transaction Server for z/OS システム以外のシステムへの従属 LU6.2 セッションがある場合、CICS は、セッションを強制的に停止することで、従属を待機させるように指示することができます。

コーディネーターによる未確定の解決を待機できないリソースがある場合、ACTION 属性に従って、トランザクションの処理が決定されます。このような方法で実際に強制的な決定が行われるのは、TDQUEUE 定義で WAIT(NO) を指定して一時データ・キューの更新を行う場合と、端末関連リソースをインストールする場合のみです。通常、後者は INSTALL コマンドを使用してインストールされます。

256 ページの表 4 は TRANSACTION 定義に WAIT 属性を定義する方法と、競合した場合に論理的にリカバリー可能な TDQUEUE 定義が解決される方法を示しています。

NO UOW は待機しません。CICS は、ACTION 属性に指定されたアクションが何であっても即時に実行します。

表 4. *TRANSACTION* 定義および *TDQUEUE* 定義の *WAIT* 属性の解決

TDQUEUE 定義の **WAIT** 属性	**TDQUEUE** 定義の **WAITACTION** 属性	**TRANSACTION** 定義の **WAIT** 属性	意味
NO	適用されない	YES	TD WAIT(NO) によって、TRANSACTION 定義の WAIT(YES) を指定変更します。UOW は、トランザクションの ACTION 属性に従って、強制的にコミットまたはバックアウトされます。
NO	適用されない	NO	UOW は、トランザクションの ACTION 属性に従って、強制的にコミットまたはバックアウトされます。
YES	QUEUE	YES	UOW は待機 (中断) します。別のタスクからの TD キューのロック要求は、待機する必要があり、CICS によってキューに入れられます。
YES	QUEUE	NO	トランザクション WAIT(NO) によって、TDQUEUE 定義を指定変更します。 UOW は、トランザクションの ACTION 属性に従って、強制的にコミットまたはバックアウトされます。
YES	REJECT	YES	UOW は待機 (中断) します。別のタスクからの TD キューのロック要求は、LOCKED 状態により拒否されます。
YES	REJECT	NO	トランザクション WAIT(NO) によって、TDQUEUE 定義を指定変更します。 UOW は、トランザクションの ACTION 属性に従って、強制的にコミットまたはバックアウトされます。

UOW が複数の一時データ・キューを参照し、それらのキューの WAIT オプションに不整合があった場合、WAIT(NO) が常に優先され、WAIT(YES) をオーバーライドします。したがって、1 つの TDQUEUE 定義の WAIT(NO) が、失敗した未確定 UOW に対して、UOW の TRANSACTION 定義で定義されている BACKOUT 属性または COMMIT 属性を強制的に実行します。

WAITTIME({**00,00,00**|*dd,hh,mm*})

ACTION 属性に指定された値に基づいて未確定の作業単位に関する決定を行うまでに、トランザクションが待機する時間の長さを指定します。

00,00,00

トランザクションは無制限に待機します。

dd,hh,mm
トランザクションが待機する時間 (日数、時間数、および分数) を示します。最大値は 99,23,59 です。

WAITTIME は、WAIT(YES) を指定している場合にのみ有効です。

XTPNAME(値)
この属性は、TPNAME の代わりに使用することができます。 APPC パートナーが使用できるトランザクションの名前を表す最大 128 文字の長さの 16 進数ストリングを入力します。 X'40' 以外 のすべての 16 進数の組み合わせが使用可能です。 72 文字より長い XTPNAME を DFHCSDUP に指定するには、列 72 にアスタリスクを指定します。アスタリスクを使用すると、以下の行が現在行に連結されます。

XTRANID(*xtranid*)
トランザクションを開始するために TRANSACTION 名の代わりに使用する別の名前を指定します。この名前の最大 8 桁の長さの 16 進数です。 XTRANID は 16 進形式で指定されるので、TRANSACTION 属性には指定できない文字を含む名前を使用できます。

TASKREQ も参照してください。このトランザクション別名も指定することができます。

value 4 バイトのトランザクション ID を 16 進数表記で指定します。したがって、最大 8 桁の 16 進数を ID で使用します。 8 桁より少ない 16 進数を指定した場合、ID は右側にブランクが埋め込まれます。

特定の値は、CICS が使用するために予約されているため、指定できる値には制限があります。

- 最初のバイトは X'C3' にはできません。
- 最初のバイトは X'40' 以下ではいけません。
- この値を X'00000000' にすることはできません。
- 最後の 3 バイトを X'FFFFFF' にすることはできません。

3270 装置として定義された端末から受信した非送信請求データによってトランザクションが接続される場合は、2 番目、3 番目、および 4 番目のバイトに X'00' から X'3F' の範囲の値を使用しないでください。これは、CICS がこれらの値をトランザクション ID の一部ではなく、制御文字として解釈するためです。例えば、EXEC CICS RETURN または EXEC CICS START を発行し、TRANSID(X'41303238') を指定した場合、正しいトランザクションが接続されます。しかし、TRANSID を指定せずに EXEC CICS RETURN を発行し、3270 装置が X'41303238' から始まるデータを送信した場合、CICS は、X'41404040' が送信されたかのようにトランザクションを接続しようとします。

TSMODEL リソース

一時記憶域モデルは、TS キュー・セットのプロパティーを指定します。個々の TS キューは、キュー名の先行文字に一致する文字ストリングである 接頭部 によって TSMODEL に関連付けられます。

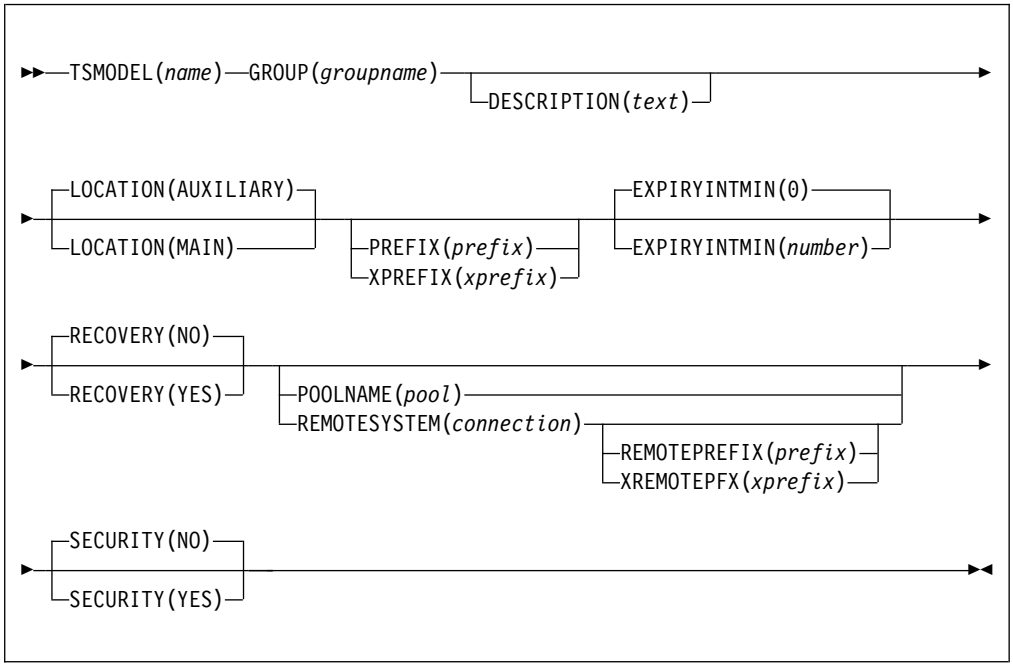
名前を共用 TS プールに直接マップすることもできます (共用システム ID は必要ありません)。

注: TSMODEL が定義されていない領域では、デフォルトのアクションが CICS によって行われます。つまり、AOR、QOR、および QOR への AOR 直接要求で定義されている TSMODEL があり、対応する TSMODEL が QOR に存在しない場合は、一部のキュー属性はデフォルト値から取られます。例えば、キューの位置 (MAIN または AUX) は、CICS 内のデフォルト設定から決定されます。一致するモデルがない場合は、EXEC CICS コマンドで指定された位置が使用されます。一致するモデルがある場合は、この位置が使用されます。

BAS の作業については、BAS 一時記憶域モデル・リソース定義の作業を参照してください。

TSMODEL 属性

TSMODEL リソースの構文と属性を説明します。



DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

EXPIRYINTMIN({**0**|*number*})
このモデルと一致する一時記憶域キューの有効期限間隔を分単位で指定します。CICS は、値を最も近い 10 分の倍数に切り上げて使用します。それぞれの一時

記憶域キューが使用された後、間隔カウントが開始します。有効期限間隔に達する前にそのキューが再度使用されない場合、キューは CICS によって自動的に削除される対象となります。

0
このモデルと一致する一時記憶域キューに適用される有効期限間隔はありません。一時記憶域キューは自動削除の対象になりません。この設定はデフォルトです。

number
1 から 900000 の範囲で指定する分単位の有効期限間隔。この有効期限間隔の後、再度使用されなかった場合、このモデルと一致する一時記憶域キューは自動削除の対象になります。

有効期限間隔は、以下の場所にある一時記憶域キューに適用されます。

- ローカル CICS 領域の主一時記憶域。
- ローカル CICS 領域と関連付けられたリカバリー不能な補助一時記憶域 (DFHTEMP データ・セット)。
- 共用一時記憶域プール内の一時記憶域キュー (CICS TS 5.2 以上)。

有効期限間隔は以下のタイプの一時記憶域キューには適用されないため、CICS により自動的に削除されることはありません。

- リカバリー可能として定義された補助一時記憶域にあるキュー。
- リモート CICS 領域にあるキュー。 CICS にリモート一時記憶域キューを削除させるには、キューがある領域の適切な TSMODEL リソース定義に有効期限間隔を指定します。
- CICS が自分で使用するために作成したキュー。

TSMODEL リソース定義の有効期限間隔を変更する場合、このモデルと一致する既存の一時記憶域キューは影響を受けません。これらのキューは、キューが作成された時点で適用された有効期限間隔を使用し続けます。ゼロ以外の有効期限間隔を指定したすべての TSMODEL リソース定義が CICS 領域から削除された場合、CICS は有効期限が切れた一時記憶域キューのスキャンを終了します。

GROUP(*groupname*)
各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

LOCATION({**AUXILIARY**|**MAIN**})
補助記憶域または主記憶域のどちらにキューを保持するかを指定します。

AUXILIARY
このモデルと一致するキューは補助一時記憶域に保持されます。 API 要求で何を指定しても無視されます。

MAIN

このモデルと一致するキューは主一時記憶域に保持されます。 API 要求で何を指定しても無視されます。

リモート・キューおよび共用一時記憶域プールにあるキューに関連する一時記憶域モデルの LOCATION は無視されます。リモート・キューに対して TSMODEL リソース定義の LOCATION を使用すると、ローカルおよびリモートの領域両方に同じ定義をインストールすることができます。 Shared resources for intercommunicationを参照してください。

POOLNAME(*pool*)

この TSMODEL リソース定義と共に使用する共用 TS プール定義の名前を指定します。名前の最大長は 8 文字です。

許容文字: A-Z 0-9 $ @ # _ 小文字を入力した場合は、大文字に変換されます。

組み込みブランクは認められません。また、すべてブランクからなる名前は、プール名が指定されなかったものとして処理されます。

REMOTESYSTEM を指定いる場合、POOLNAME を指定することはできません。

アプリケーション・プログラムにより EXEC CICS 一時記憶域コマンドで SYSID が指定されている場合、または XTSEREQ グローバル・ユーザー出口プログラムにより SYSID が追加されている場合、CICS は一致する TSMODEL リソース定義を検索しません。アプリケーション・プログラムが明示的に SYSID を指定するときに CICS が一時記憶域データ共用プールの名前を検索するようにするには、適切な TYPE=SHARED 項目がある一時記憶域テーブル (TST) を使用する必要があります。

PREFIX(*prefix*)

CICS が、一致する一時記憶域キューを識別するために使用する文字ストリングを指定します。接頭部は、16 文字以内の長さで指定することができます。

許容文字: A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

最も簡単なケースでは、接頭部がその接頭部で始まる TS キュー名と一致します。例えば、接頭部 ABCD が、キュー名 ABCD、ABCDE、および ABCD99 に一致します。

TS キュー名のどんな文字にも一致させるために、総称文字 (+) を接頭部内に 1 回以上使用できます。例えば、接頭部 A++D は、キュー名 ABCD、A99D、および ABCD99 に一致します。ABD には一致しません。+ 文字を接頭部の末尾に指定する必要はありません。接頭部には + 文字が右側に TS キュー名の最大長 (16 文字) まで埋め込まれるものとして考えることができます。例えば、ABC という接頭部は ABC+ または ABC+++++ という接頭部と同等になります。

+ 文字を接頭部に使用する際に、いくつか考慮すると、複数の接頭部が 1 つの TS キュー名に一致することに気付くかもしれません。この場合、CICS は以下の規則を使用して、一致する TS モデルを選択します。

- 接頭部に正確に指定された文字は、+ 文字よりも強く一致します。
- 文字は左から右に 1 つずつ比較されます。また、一致した文字は左にあるほど強い一致になります。

以下に例を挙げます。

- 接頭部 ABCD (一致する 4 文字を正確に指定) と接頭部 AB+D (正確に指定された文字は 3 文字だけ) の場合、キュー名 ABCD99 は前者にマッチングします。
- 接頭部 ABC+ と接頭部 AB+D の場合、キュー名 ABCD99 は、前者にマッチングします。どちらの接頭部についても正確に一致する文字の数は同じです。しかし、3 番目に正確に一致する文字については、第 1 の接頭部の方がより左に位置します。

接頭部がブランクの TSMODEL 定義を作成すると、予測できない結果が生じることがあります。接頭部がブランクの TSMODEL 定義は、他の TSMODEL 定義とはマッチングしないキュー名すべてとマッチングします。

アプリケーション・プログラムが明示的に SYSID を指定するときに CICS が一時記憶域データ共用プールの名前を検索するようにするには、適切な TYPE=SHARED 項目がある一時記憶域テーブル (TST) を使用する必要があります。

RECOVERY({**NO**|**YES**})

このモデルと一致するキューをリカバリー可能にするかどうかを指定します。

NO このモデルと一致するキューはリカバリー不能です。

YES このモデルと一致するキューはリカバリー可能です。

RECOVERY(YES) は LOCATION(MAIN) と一緒に指定できません。

REMOTEPREFIX(*prefix*)

CICS が、リモート・システムにある一致する一時記憶域キューを識別するために使用する文字ストリングを指定します。接頭部は、16 文字以内の長さで指定することができます。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

REMOTEPREFIX は、REMOTESYSTEM が指定されていないと指定できません。REMOTEPREFIX 属性の長さは、PREFIX 属性の長さと同じでなければなりません。

REMOTEPREFIX 属性で文字を突き合わせる規則は、PREFIX 属性の規則と同じです。 + 文字を総称突き合わせに使用する場合、PREFIX と REMOTEPREFIX で、同じように総称文字を配置する必要があります。以下に例を示します。

PREFIX: A++D

REMOTEPREFIX: X++Y

REMOTESYSTEM(接続)

ローカル・システムを一時記憶域キューが常駐するリモート・システムへリンクする接続の名前を指定します。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

組み込みブランクは認められません。また、すべてブランクからなる名前は、リモート・システム名が指定されなかったものとして処理されます。

REMOTESYSTEM および POOLNAME は相互に排他的です。
REMOTESYSTEM を指定している場合、POOLNAME は無視されます。

SECURITY({**NO**|**YES**})

このモデルと一致するキューに対して、セキュリティー検査を実行するかどうかを指定します。

NO このモデルと一致するキューに対して、セキュリティー検査を実行しません。

YES このモデルと一致するキューに対して、セキュリティー検査を実行します。

詳細については、一時記憶域のセキュリティーを参照してください。

TSMODEL(*name*)

この TSMODEL リソース定義の名前を指定します。名前の最大長は 8 文字です。

許容文字: A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

途中にブランクを含めることはできません。

この名前は、CSD ファイルの TSMODEL リソース定義を識別するために使用されます。アクティブな CICS システム内では使用されません。

XPREFIX(*xprefix*)

PREFIX の代わりに使用できます。このモデルの接頭部として使用する 16 進数ストリング (最大 32 文字の長さ) を入力します。 16 進形式で XPREFIX を指定するため、PREFIX 属性では入力できなかった文字を含む名前を使用できます。

XPREFIX 属性で総称文字を指定するには、コード X'4E' を使用します。

XPREFIX を TS キュー名と突き合わせる規則は、PREFIX 属性の規則と同じです。

XREMOTEPFX(*xprefix*)

REMOTEPREFIX の代わりに使用できます。リモート・システムの接頭部として使用する 16 進数ストリング (最大 32 文字の長さ) を入力します。 16 進形式で XREMOTEPFX を指定するため、REMOTEPREFIX 属性では入力できなかった文字を含む名前を使用できます。

XPREFIX 属性で総称文字を指定するには、コード X'4E' を使用します。

XREMOTEPFX を TS キュー名と突き合わせる規則は、REMOTEPREFIX 属性の規則と同じです。

TYPETERM リソース

TYPETERM リソースは、端末のグループに共通の属性をまとめて定義します。

リソースは、TERMINAL の論理拡張です。同じプロパティーを持つ端末を複数使用している場合は、必須値を指定したTYPETERMを1つ定義し、各TERMINAL定義（自動インストールを使用する場合は自動インストール・モデル定義）でこのTYPETERMを指定します。

各TERMINAL定義ではTYPETERM定義を指定する必要があります。この単一属性は複数の特性を表します。したがって、かなりの労力を節約でき、間違いを犯す可能性が低減します。同じ種類の端末が多数ある場合には、TYPETERMを使用することで端末の定義が容易になります。

端末定義プロセスをさらに単純にする2つのTYPETERM属性について説明します。
DEVICE
QUERY

DEVICE

TYPETERM が表す装置タイプを指定します。 その他の多くの属性のデフォルト値は、この属性に指定する値に 依存しているので、これは重要な属性です。

- 同一タイプのすべての装置では、一部の属性が常に同じです。 これらの属性すべてがRDOにより認識されているので、 ユーザーがこれらの属性を指定する必要はありません。端末の TYPETERMを定義するときにRDOに対し指示する必要があるのは、 端末の装置タイプだけです。 固定属性の値は自動的に提供されます。
- その他の属性では、装置タイプに応じてデフォルト値が 提供されます。ただし、CICS が提供する値を使用する必要はありません。必要に応じて、異なる値を指定することができます。 TYPETERM定義の装置タイプを変更しても、デフォルト値は リセットされません。

DEVICE属性にはデフォルト値がないので、値を指定する必要があります。

RDO でサポートされる端末のリストについては、サポートされる装置を参照してください。 TYPETERM 属性のデフォルト値には、TYPETERM の DEVICE 属性の有効値のリストもあります。 各種装置タイプのその他の属性値が示されています。場合に より、これらの値はSESSIONTYPEとTERMMODELに指定した値に よっても異なります。ただし、これらの属性のデフォルト値も、 指定されているDEVICEによって決まります。

一般的なディスプレイ装置、プリンター、およびその他の特殊入出力装置とは別に、CICS コンソールの TYPETERM 定義を作成することができます。

QUERY

QUERY属性では、端末が接続するまで端末の一部の機能を未定義 にしておくことができます。これにより、CICS 自体が QUERY 構造化フィールドを使用して、これらの属性に関する情報を取得することができます。

QUERYを使用できるすべての属性はTYPETERM属性でもあり ます。それらは、以下のとおりです。
ALTPAGE
ALTSCREEN
APLTEXT

BACKTRANS
CGCSGID
COLOR
EXTENDEDDS
HILIGHT
MSRCONTROL
OBFORMAT
OUTLINE
PARTITIONS
PROGSYMBOLS
SOSI
VALIDATION

QUERY を使用すると、ここでリストされた TYPETERM 属性 (**ALTSCREEN** を除く) に対して明示的に定義された値がオーバーライドされます。 QUERYにより指定された ALTSCREENの値は、TYPETERMでALTSCREEN値が明示的に指定されて いない場合にのみ使用されます。

拡張3270データ・ストリームを備えた3270装置にQUERYを使用 できます。QUERYを使用できるDEVICEタイプを次に示します。

3270
3270P
LUTYPE2
LUTYPE3
SCSPRINT

QUERYを次の2つのうちいずれかの方法で使用することを指定できます。

- QUERY(COLD)は、初期開始またはコールド・スタートの 後に初めて端末を接続する場合にのみQUERYを発行することを 指定します。
- QUERY(ALL)は、端末が接続するたびにQUERYを発行することを 指定します。

QUERY 機能は、IBM Personal System/2 (PS/2) や IBM 3290 などの構成可能装置で特に有用です。この機能を使用して、リソース定義を変更しなくても、CICS へのログオフとログオンの間に装置を再構成することができます。

QUERY機能は自動インストールを組み合わせて使用するときに特に 便利です。

QUERY 機能では、CICS で必要な情報しか取得されないので注意してください。アプリケーション・プログラムが他の装置特性を判別する必要がある場合でも、QUERY構造化フィールドを送信して応答を分析する必要があります。

要約すると、各装置タイプごとに 1 つのTYPETERM定義が必要です。 QUERYにより決定する属性が端末によって異なる場合でも、装置タイプごとに必要なTYPETERMは 1 つだけです。端末の他の属性が異なる場合は、装置タイプに複数のTYPETERM定義が必要になります。

頻繁に使用される端末に適したCICS提供のTYPETERM定義があります。これらについては、グループ DFHTYPE 内の TYPETERM 定義 で説明されています。

すべての端末が基本的に同じである場合には、TYPETERM定義を 1 つと、AUTINSTMODEL(YES)を指定したTERMINAL定義を 1 つだけ使用します。QUERYを使用して、端末が使用する他の機構を処理することができます。

BAS の作業については、BAS Typeterm リソース定義の作業を参照してください。

TYPETERM属性のデフォルト値

TYPETERM 定義で DEVICE、SESSIONTYPE、および TERMMODEL を指定すると、CICS は、その他の多くの属性のデフォルト値を提供します。

デフォルト値は、表 5 に示されています。一部の属性では、提供される値が必須であり、これらの値を変更できない点に注意してください。

表 *5. TYPETERM*属性のデフォルト値

DEVICE	**SESSIONTYPE**	**TERMMODEL**	デフォルト値
3270 3277 L3277 注 1 (272 ページ) を参照		1	DEFSCREEN(12,40) PAGESIZE(12,40) AUTOPAGE(N) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3270		2 (デフォルト値)	DEFSCREEN(24,80) PAGESIZE(24,80) AUTOPAGE(N) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3275		1	DEFSCREEN(12,40) PAGESIZE(12,40) AUTOPAGE(N) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3275		2 (デフォルト値)	DEFSCREEN(24,80) PAGESIZE(24,80) AUTOPAGE(N) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3270P 3284 L3284 3286 L3286 注 2 (272 ページ) を参照		1	DEFSCREEN(12,40) PAGESIZE(12,40) AUTOPAGE(Y) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)

表 5. *TYPETERM*属性のデフォルト値 (続き)

DEVICE	**SESSIONTYPE**	**TERMMODEL**	デフォルト値
3270P		2 (デフォルト値)	DEFSCREEN(24,80) PAGESIZE(24,80) AUTOPAGE(Y) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
APPC			DEFSCREEN(0,0) PAGESIZE(1,40) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(Y) (必須) ROUTEDMSGS(NONE) (必須)
CONSOLE			DEFSCREEN(0,0) (必須) PAGESIZE(1,124) (必須) AUTOPAGE(N) BRACKET(N) (必須) BUILDCHAIN(N) (必須) ROUTEDMSGS(NONE)
LUTYPE2		1	DEFSCREEN(12,40) PAGESIZE(12,40) AUTOPAGE(N) BRACKET(Y) (必須) BUILDCHAIN(Y) (必須) ROUTEDMSGS(ALL)
LUTYPE2		2 (デフォルト値)	DEFSCREEN(24,80) PAGESIZE(24,80) AUTOPAGE(N) BRACKET(Y) (必須) BUILDCHAIN(Y) (必須) ROUTEDMSGS(ALL)
LUTYPE3		1	DEFSCREEN(12,40) PAGESIZE(12,40) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
LUTYPE3		2 (デフォルト値)	DEFSCREEN(24,80) PAGESIZE(24,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)

表 5. *TYPETERM*属性のデフォルト値 (続き)

DEVICE	**SESSIONTYPE**	**TERMMODEL**	デフォルト値
LUTYPE4			DEFSCREEN(0,0) PAGESIZE(50,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
BCHLU	(デフォルト値)		DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
BCHLU	BATCHDI		DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
BCHLU	USERPROG		DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
INTLU			DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
SCSPRINT			DEFSCREEN(0,0) (必須) PAGESIZE(24,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
TLX または TWX	CONTLU (デフォルト値)		DEFSCREEN(0,0) PAGESIZE(1,40) AUTOPAGE(Y) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)

表 5. *TYPETERM*属性のデフォルト値 (続き)

DEVICE	**SESSIONTYPE**	**TERMMODEL**	デフォルト値
TLX または TWX	INTLU		DEFSCREEN(0,0) PAGESIZE(1,40) AUTOPAGE(Y) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3600	(デフォルト値)		DEFSCREEN(0,0) PAGESIZE(1,40) AUTOPAGE(Y) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3600	PIPELINE		DEFSCREEN(0,0) PAGESIZE(6,30) AUTOPAGE(Y) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3614			DEFSCREEN(0,0) PAGESIZE(1,40) AUTOPAGE(Y) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3650	USERPROG (デフォルト値)		DEFSCREEN(0,0) PAGESIZE(3,80) AUTOPAGE(Y) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3650	3270		DEFSCREEN(12,40) PAGESIZE(23,80) AUTOPAGE(Y) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3650	3653		DEFSCREEN(0,0) PAGESIZE(6,30) AUTOPAGE(Y) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)

表 5. *TYPETERM*属性のデフォルト値 (続き)

DEVICE	**SESSIONTYPE**	**TERMMODEL**	デフォルト値
3650	PIPELINE		DEFSCREEN(0,0) PAGESIZE(6,30) AUTOPAGE(Y) BRACKET(Y) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3767			DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3767C			DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3767I			DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3770	(デフォルト値)		DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3770	USERPROG		DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3770	BATCHDI		DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)

表 5. *TYPETERM*属性のデフォルト値 (続き)

DEVICE	SESSIONTYPE	TERMMODEL	デフォルト値
3770B	(デフォルト値)		DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3770B	BATCHDI		DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3770B	USERPROG		DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3770C			DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3770I			DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3790	(デフォルト値)		DEFSCREEN(0,0) PAGESIZE(1,40) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(NONE) (必須)
3790	SCSPRINT		DEFSCREEN(0,0) PAGESIZE(24,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)

表 5. *TYPETERM*属性のデフォルト値 (続き)

DEVICE	**SESSIONTYPE**	**TERMMODEL**	デフォルト値
3790	USERPROG		DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3790	BATCHDI		DEFSCREEN(0,0) PAGESIZE(12,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3790	3277CM 注 3 (272 ページ) を参照	1	DEFSCREEN(12,40) PAGESIZE(12,40) AUTOPAGE(N) BRACKET(Y) (必須) BUILDCHAIN(Y) (必須) ROUTEDMSGS(ALL)
3790	3277CM 注 3 (272 ページ) を参照	2 (デフォルト値)	DEFSCREEN(24,80) PAGESIZE(24,80) AUTOPAGE(N) BRACKET(Y) (必須) BUILDCHAIN(Y) (必須) ROUTEDMSGS(ALL)
3790	3284CM 注 4 (272 ページ) を参照	1	DEFSCREEN(12,40) PAGESIZE(12,40) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3790	3284CM 注 4 (272 ページ) を参照	2 (デフォルト値)	DEFSCREEN(24,80) PAGESIZE(24,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)
3790	3286CM 注 4 (272 ページ) を参照	1	DEFSCREEN(12,40) PAGESIZE(12,40) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)

表 5. *TYPETERM*属性のデフォルト値 (続き)

DEVICE	SESSIONTYPE	TERMMODEL	デフォルト値
3790	3286CM 注 4 を参照	2 (デフォルト値)	DEFSCREEN(24,80) PAGESIZE(24,80) AUTOPAGE(Y) BRACKET(Y) (必須) BUILDCHAIN(N) ROUTEDMSGS(ALL)

注:

1. DEVICE(3277) または DEVICE(L3277) を指定すると、CICS はユーザーが指定した値を DEVICE(3270) に置換します。
2. DEVICE(3284)、DEVICE(3286)、DEVICE(L3284)、または DEVICE(L3286) を指定すると、CICS はユーザーが指定した値を DEVICE(3270P) に置換します。
3. DEVICE(3790) および SESSIONTYPE(3277CM) を指定すると、CICS はユーザーが指定した値を DEVICE(LUTYPE2) に置換します。 SESSIONTYPE の値 DEVICE(LUTYPE2) はありません。
4. DEVICE(3790) と SESSIONTYPE(3284CM) または SESSIONTYPE(3286CM) を指定すると、CICS はユーザーが指定した値を DEVICE(LUTYPE3) に置換します。 SESSIONTYPE の値 DEVICE(LUTYPE3) はありません。

サポートされる装置

このトピックでは、TYPETERM 定義に使用できる装置名をリストしています。

表 6 は、RDO でサポートされる端末タイプのリストです。この表を使用するには、左端の列でご使用の装置のファミリー番号を見つけます (「端末またはシステムのタイプ」の見出し)。次に、2 列目 (「装置」の見出し) で、ご使用の装置のタイプが具体的に示されているかどうかを確認します。次に、3 列目 (「接続」の見出し) で、TYPETERM 定義で使用する装置タイプを確認します。

必要に応じて、表の下の注記に詳細を記載しています。以下の省略形が使用されます。

ローカル **(local)**
: チャネルまたはアダプター接続

s/s : スタート・ストップ伝送

SDLC : 同期データ・リンク制御

sw : 交換

BSC : 2 進データ同期

nonsw
: 非交換通信

表 6. *RDO* でサポートされる *z/OS Communications Server* の端末およびサブシステム

端末またはシステムのタイプ	装置	接続	注
3101		TWX 33/35 としてサポート	9 (276 ページ)
3230		INTLU としてサポート (z/OS Communications Server)	

表 6. *RDO* でサポートされる *z/OS Communications Server* の端末およびサブシステム (続き)

端末またはシステムのタイプ	装置	接続	注
3270	3178, 3179, 3180, 3262, 3271, 3272, 3274, 3276, 3290	ローカル、SDLC、BSC nonsw	1 (275 ページ), 2 (275 ページ)
	3275, 3277, 3278, 3279, 3284, 3286, 3287, 3288, 3289	BSC sw または nonsw	2 (275 ページ)
3270PC	3270PC, 3270PC/G, 3270PC/GX	3270 としてサポート	
3287	モデル 11、12	SDLC (SCSPRT としてサポート)	12 (276 ページ)
3600	3601, 3602, 3690, 3604, 3610, 3612, 3618, 3614, 3624	SDLC、BSC nonsw	3 (275 ページ), 4 (275 ページ), 13 (276 ページ)
3630	3631, 3632, 3643, 3604	3600 として接続	3 (275 ページ), 10 (276 ページ)
3640	3641, 3644, 3646, 3647	SDLC (INTLU として接続)	12 (276 ページ)
	3642	SDLC (SCSPRT として接続)	12 (276 ページ)
	3643	SDLC (LUTYPE2 としてサポート)	12 (276 ページ)
	3645	SDLC (SCSPRT としてサポート)	12 (276 ページ)
3650	3651, 3653, 3275, 3284	SDLC	3 (275 ページ)
3680	3684	3790/3650 としてサポート	3 (275 ページ)
3730	3791	3790 としてサポート	3 (275 ページ)
3767		SDLC s/s (2740/2741 としてサポート)	
3770	3771, 3773, 3774	SDLC	3 (275 ページ), 5 (276 ページ)
	3775, 3776, 3777	BSC (2770 としてサポート)	
3790	3791	SDLC またはローカル	3 (275 ページ), 6 (276 ページ)
4300	4331, 4341, 4361, 4381	BSC または SDLC	3 (275 ページ), 7 (276 ページ)
4700	4701-1	3600 としてサポート	3 (275 ページ), 4 (275 ページ)
5280		3270 としてサポート (z/OS Communications Server)	
5520		SDLC (3790 全機能 LU としてサポート) BSC (2770 としてサポート) SDLC (APPC として接続)	3 (275 ページ)
5550		3270 としてサポート	
5937		SDLC/BSC (3270 として接続)	2 (275 ページ)
6670		SDLC BSC (2770 としてサポート)	

表 6. *RDO* でサポートされる *z/OS Communications Server* の端末およびサブシステム (続き)

端末またはシステムのタイプ	装置	接続	注
8100	DPCX を備えた 8130/8140 プロセッサー	3790 としてサポート	3 (275 ページ)
	ホスト表示サービスまたはホスト・トランザクション機能を使用する DPPX/BASE	3790 として接続	3 (275 ページ)
	DPPX/DSC または DPCX/DSC (8775 接続を含む)	3270 としてサポート	3 (275 ページ), 11 (276 ページ)
8775		SDLC (LUTYPE2 としてサポート)	
8815		APPC としてサポート	
ディスプレイライター		APPC としてサポート EDDS の SNA 3270 としてサポート 2741 として接続 (s/s) または 3780 として接続 (BSC) SDLC (APPC として接続)	
パーソナル・コンピューター		3270 および APPC としてサポート	13 (276 ページ)
PS/2		3270 および APPC としてサポート	13 (276 ページ)
Scanmaster		APPC としてサポート	
シリーズ/1		System/3 として接続; 3650 パイプラインとしてサポート (z/OS Communications Server) または 3790 (全機能 LU)	3 (275 ページ)
System/32	5320	SDLC (3770 としてサポート) BSC (2770 としてサポート)	3 (275 ページ), 8 (276 ページ)
System/34	5340	SDLC (3770 としてサポート) BSC (System/3 として接続)	3 (275 ページ), 8 (276 ページ)
System/36		System/34 としてサポート SDLC (APPC として接続)	
System/38	5381	SDLC (3770 として接続) SDLC (APPC として接続) BSC (System/3 として接続)	3 (275 ページ), 8 (276 ページ)
AS/400	5381	SDLC (3770 として接続) SDLC (APPC として接続) BSC (System/3 として接続)	3 (275 ページ), 8 (276 ページ)
System/370		SDLC (APPC として接続)	3 (275 ページ)
System/390®		SDLC (APPC として接続)	3 (275 ページ)
z Systems®		SDLC (APPC として接続)	3 (275 ページ)

表 6. *RDO* でサポートされる *z/OS Communications Server* の端末およびサブシステム (続き)

端末またはシステムのタイプ	装置	接続	注
TWX 33/35		z/OS Communications Server (NTO 経由) ss sw	9 (276 ページ)
WTTY		z/OS Communications Server (NTO 経由) ss nonsw	9 (276 ページ)

注:

1. CICS は、端末管理と BMS インターフェースの両方を介して、3290 をサポートします。3290の状態はデフォルト、 代替、区分の3つのいずれかです。最大16の区分画面を 定義できます。3290には、全桁入力必須および入力必須の 2種類のデータ妥当性検査機能だけでなく、プログラム式 シンボルと拡張強調表示機能もあります。3290端末は 1つから5つの論理装置として構成できます。3290のセットアップ 時に各論理装置のサイズを定義します。予測不能な結果を回避するために、各論理装置のリソース定義がセットアップ・サイズに一致している必要があります。構成で同時に活動化できる対話式 画面は最大4つですが、常にプログラム式シンボルを持つ画面 として定義できる対話式画面は1つのみです。つまり、すべての プログラム式シンボル・セットをこの対話式画面に割り当てる 必要があります。

   長いデータ行(1行が132文字のCEMT出力など)を表示するため、 デフォルトの画面幅として132文字を指定します。

   大容量のバッファーを使用する予定の場合は、IOAREALEN にさらに大きい値を指定する必要がある場合があります (277 ページの『TYPETERM 属性』を参照)。 その必要があるかどうかは、オペレーターが大量のデータを 変更または入力する可能性があるかどうかによって決まります。 照会のみまたは限られたデータ入力の目的で端末を使用する 場合は、IOAREALENを大きくする必要はありません。
2. SDLC 3270 は、z/OS Communications Server を介してのみサポートされます。ローカルまたは SDLC 3274 および SDLC 3276 に接続されたプリンターは、z/OS Communications Server 経由で、3270 プリンター・データ・ストリームを使用する LU タイプ 3、または SCS データ・ストリーム (SDLC 3767、3770、および 3790 プリンターに使用されるサブセット) を使用する LU タイプ 1としてサポートされます。 3288 は、3286 モデル 2 としてサポートされます。CICS は、3270 コピー機能 (#1550) をサポートします。
3. システムまたはプログラマブル・コントローラーによってサポートされる装置および機能は、通常は CICS に透過的です。場合によっては、CICS が特定の装置サポートを提供します。そのような場合は、装置がリストされています。
4. SDLC は、z/OS Communications Server を介してサポートされます。3614 は、3601への ループ接続機構と3704/3705通信制御装置経由でのホストへのSDLC接続機構の両方でサポートされます。

   3614 は、3601/3602 コントローラーにループ接続されている場合にのみ、BSC として CICS にサポートされます。 3624は3614としてサポート されます。

   3690は3602としてサポートされます。

5. CICS は、3770 データ通信システムの SDLC プログラマブル・モデルのデータ転送機能をサポートします。 この機能を使用する場合には、データ・セットの割り振りと プログラム・ライブラリーの管理はユーザーの責任となります。
6. CICS は、3760 を使用する 3790/データ入力構成をサポートしません。バージョン 1 リリース 3.0 で初めて導入された CICS 機能拡張をサポートするには、#9165 または #9169 構成が必要です。

   3790システムのプリンターは、以下のいずれかでサポート されます。

   - ユーザーが提供する機能プログラム
   - 3270プリンター・データ・ストリーム(LU3)との 3270データ・ストリーム互換性機能
   - SDLC 3767 (LU1)のデータ・ストリームのサブセットをサポート するSCSデータ・ストリーム

   3270モードで作動する場合、3288 Model 2は3286 Model 2として サポートされます。
7. BSC による 4300 接続機構には、CICS に接続されたシステム内に安定した通信プログラム (VSE/3270 Bisync Pass Through Program など) が必要です。CICS は、リモート CPU に適したプログラムではありません。 SDLC による接続機構は、CICS システム間通信によってサポートされます。
8. SNA/SDLCワークステーション・システム・ユーティリティー・ プログラムを持つシステム/32、システム/34、およびシステム/38は、 バッチ論理装置として動作する、適切な機能を持つ3770 通信システムの互換バージョンとしてサポートされています。 システム/34またはシステム/38ユーザー作成プログラムは、 接続されたサブシステムの正しいSNAシーケンスをサポート する責任があります。
9. TWX および WTTY は、NTO によって定義された接続機構を使用するネットワーク端末オプション・ライセンス・プログラム (5735-XX7) により、z/OS Communications Server を介してサポートされます。

   WTTY は通信事業者の交換ネットワーク上で 50 bps の速度で接続されます。このネットワークでサポートされる端末は、IBM World Trade Corporation 電信装置端末制御を介して電信装置回線アダプターと接続する端末です。

   使用される伝送コードはInternational Telegraph Alphabet No. 2 (CCITT No. 2)です。CICS は、自動呼び出しおよび WRITE ブレークによる自動ホスト切断をサポートしません。
10. 3643は3604としてサポートされています。
11. 8775サポートには、全桁入力必須および入力必須のフィールド属性 の検証が含まれます。
12. 接続機構は、4331プロセッサーのループ・アダプター#4830, #4831, およびデータ・リンク・アダプター#4840経由で接続します。
13. 3270 サポートには、パーソナル・コンピューターまたは PS/2 に 3278/3279 エミュレーション・アダプターがインストールされている必要があります。
14. 3600パイプライン論理装置は、特定のタイプのトランザクション （クレジット・カード権限処理または一般照会アプリケーション など）で高スループットを実現するように設計されています。照会メッセージおよびその応答の高スル

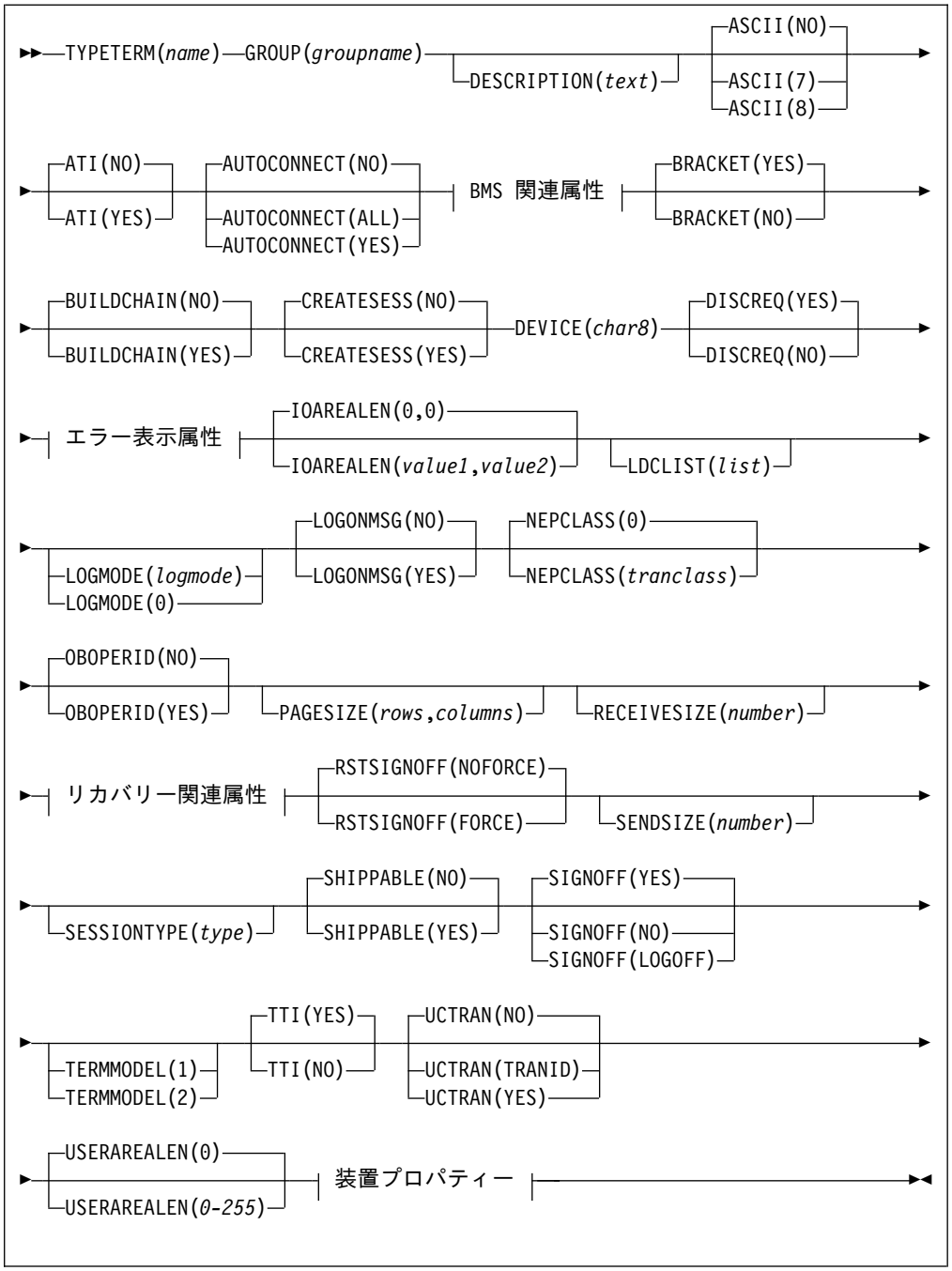
ープットを実現するために、CICS パイプライン・セッションでは、3601 論理装置で使用される制限付きの通信プロトコル・セットを使用します。

これらの制限により、メッセージ・フローが全二重で、多数の 照会メッセージがある時点で未処理の状態となり、応答が元の照会 とは異なる順序で戻る可能性があります。照会端末を制御する 4700/3600アプリケーション・プログラムは、照会と応答の相関、 およびパイプライン論理装置に関連した端末グループへの メッセージ・フローの制御だけでなく、プロトコルの保守にも 関与します。

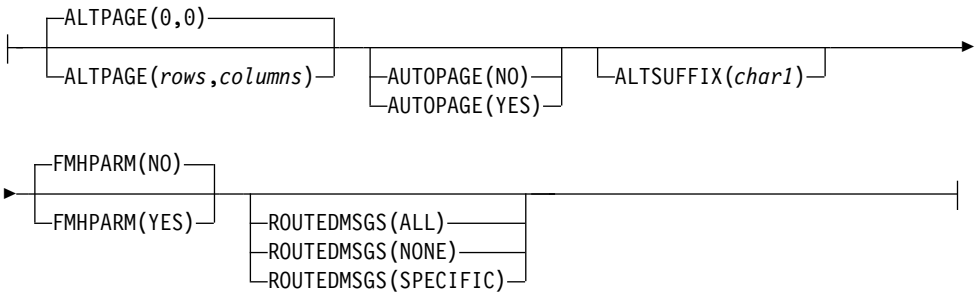
CICS は、パイプライン端末での自動トランザクション開始 (ATI) をサポートしません。

TYPETERM 属性

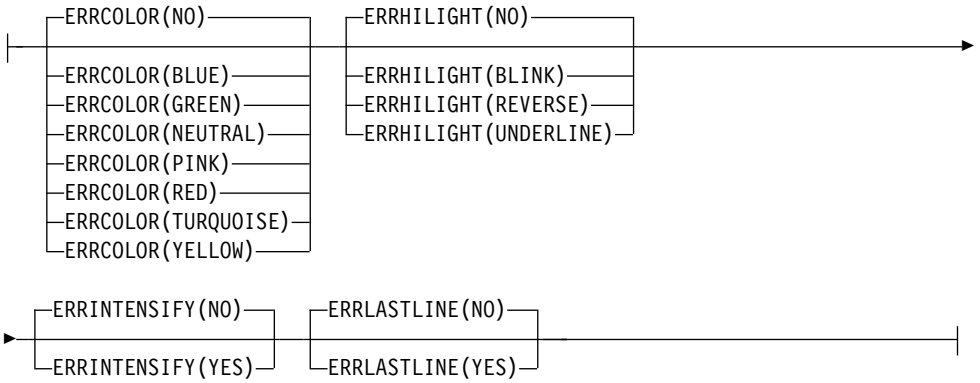
TYPETERM リソースの構文および属性について説明します。


TYPETERM(name)
GROUP(groupname)
DESCRIPTION(text)
ASCII(NO)
ASCII(7)
ASCII(8)
ATI(NO)
ATI(YES)
AUTOCONNECT(NO)
AUTOCONNECT(ALL)
AUTOCONNECT(YES)
BMS 関連属性
BRACKET(YES)
BRACKET(NO)
BUILDCHAIN(NO)
BUILDCHAIN(YES)
CREATESESS(NO)
CREATESESS(YES)
DEVICE(char8)
DISCREQ(YES)
DISCREQ(NO)
エラー表示属性
IOAREALEN(0,0)
IOAREALEN(value1,value2)
LDCLIST(list)
LOGMODE(logmode)
LOGMODE(0)
LOGONMSG(NO)
LOGONMSG(YES)
NEPCLASS(0)
NEPCLASS(tranclass)
OBOPERID(NO)
OBOPERID(YES)
PAGESIZE(rows,columns)
RECEIVESIZE(number)
リカバリー関連属性
RSTSIGNOFF(NOFORCE)
RSTSIGNOFF(FORCE)
SENDSIZE(number)
SESSIONTYPE(type)
SHIPPABLE(NO)
SHIPPABLE(YES)
SIGNOFF(YES)
SIGNOFF(NO)
SIGNOFF(LOGOFF)
TERMMODEL(1)
TERMMODEL(2)
TTI(YES)
TTI(NO)
UCTRAN(NO)
UCTRAN(TRANID)
UCTRAN(YES)
USERAREALEN(0)
USERAREALEN(0-255)
装置プロパティー

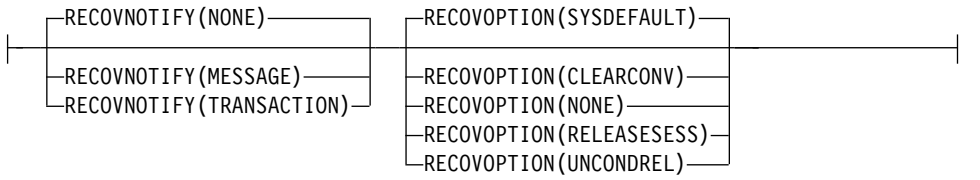
BMS 関連属性:

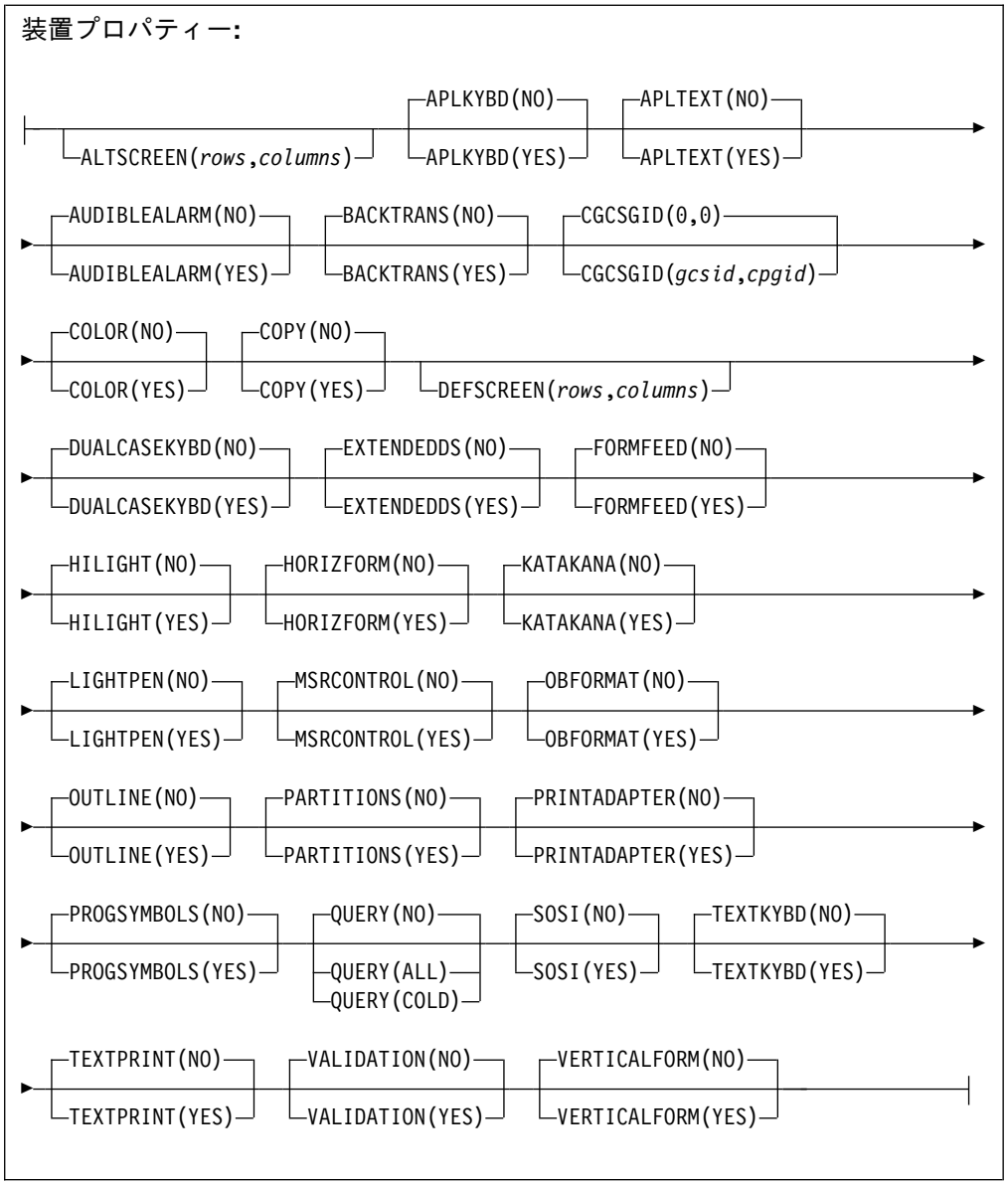


エラー表示属性:



リカバリー関連属性:





入力条件定義属性の説明は以下のとおりです。

ALTPAGE({**0**|*rows*},{**0**|*columns*})

画面サイズとして ALTSCREEN が選択されたときの、この端末入力用に BMS によって使用されるページ・サイズを指定します。デフォルトはPAGESIZEです。 *rows* と *columns* の値は、どちらも 0 から 255 の範囲でなければなりません。*rows* と *columns* の積が 32767 を超えてはなりません。

ALTPAGE の *columns* 値が ALTSCREEN の値と異なる場合、予期しない結果が生じます。 ALTPAGE の *rows* 値は ALTSCREEN の値よりも小さくすることができるので、例えば画面の一番下の行をエラー・メッセージ用に予約しておくことができます。

QUERY 構造化フィールドを使用する場合、使用される代替ページ・サイズは、代替画面サイズとしてセットアップされたサイズです。照会可能な端末の場合、ALTPAGE をゼロに設定し、CINIT BIND によって ALTSCREEN 値を明

示的に定義することができます。 ALTPAGE がゼロ以外の場合、ALTPAGE と ALTSCREEN が異なる値を持つことができます。

ALTSCREEN(*rows*,*columns*)

代替画面サイズがプロファイル定義に指定された、トランザクションに使用される 3270 画面サイズを指定します。 *rows* と *columns* の値は、どちらも 0 から 255 の範囲でなければなりません。 指定可能な値は以下のとおりです。

装置	代替画面サイズ
3276-1, 3278-1	(12,80)
3276-2, 3278-2	(24,80)
3276-3, 3278-3	(32,80)
3276-4, 3278-4	(43,80)
3278-5	(27,132)
3279-2A、3279-2B	(24,80)
3279-3A、3279-3B	(32,80)

選択した画面サイズで妥当性検査は実行されず、 サイズが誤っている場合は予測不能な結果となる 可能性があります。

BSC装置の場合、代替画面サイズとデフォルト画面サイズは どちらも装置のハードウェアによって決まります。代替 画面サイズは最大画面サイズです。3290 ディスプレイの場合、 デフォルト画面サイズと代替画面サイズはどちらも、カスタマーの セットアップ手順によって決まります。

SNA装置(LUTYPE2およびLUTYPE3)の場合、最大物理画面サイズ までの任意の値を、代替画面サイズとデフォルト画面サイズの 両方に指定できます。特に、代替画面サイズと デフォルト画面サイズの両方を最大画面サイズにしたり、 代替画面サイズを指定せずにデフォルト画面サイズを 最大画面サイズにしたりすることができます。SNA バインドは、この情報から CICS によって生成されます。ログモード・テーブル・エントリーを提供したり、装置をカスタマイズしたりする必要はありません。

非SNAの3270およびLUTYPE2装置では、QUERY構造化フィールド を使用して、ディスプレイにセットアップされた代替画面サイズ を判別することができます。QUERYを使用するには、 DEFSCREENをデフォルトの(24,80)とし、ALTSCREENを指定しない ままにしておきます。代替画面サイズは、端末ユーザーが セットアップしたサイズです。そのようにしなければ、QUERY(COLD) またはQUERY(ALL)は代替画面サイズに影響しません。ここで示された条件下で QUERY を使用せず、ALTSCREEN を指定しない場合、代替画面サイズが (00,00) になります。

デュアル画面サイズを使用する場合、関連プロファイルに SCRNSIZE (ALTERNATE) をコーディングすることで、CICS トランザクションで代替画面サイズが使用されるようにすることができます。 アプリケーションが、疑似会話型でリンクされたいくつかの トランザクションで構成されている場合、アプリケーションが代替画面 サイズを使用するのであれば、これらのトランザクションのそれぞれの プロファイルにSCRNSIZE(ALTERNATE)を指定します。

3287および3289プリンターの場合、指定された値は特定の装置の バッファー・サイズと等しくなければなりません。非SNAの3287 および3289プリンターで

は、サイズはモデル番号ではなく注文 された機構によって決まります。SNAプリンターでは、 機構がないため、有効なサイズのリストから任意の2つの サイズを指定できます。関連するTERMINAL定義に PRINTERCOPY(YES)が指定されているプリンターに印刷する場合、 ALTSCREEN値は、画面が印刷される端末の画面サイズと一致して いる必要があります。値が異なる場合、 予測不能な結果になる可能性があります。

ALTSUFFIX({**blank**|*number*})

BMS がマップ・セット名に付加する 1 文字の数値接尾部 (DFHMSD TYPE={DSECT|MAP} マクロの SUFFIX オペランドで指定)。

blank マップ・セットに接尾部を付加しない場合は、この属性を ブランクのままにします。

number

使用されている画面サイズが代替画面サイズと同じ値である場合、BMSはこの接尾部をマップ・セット名に付加します。 すなわち、トランザクションで、PROFILE定義に代替画面サイズ が指定されている場合、またはデフォルト画面サイズと 代替画面サイズが同じである場合です。この場合、 BMSマップ選択ルーチンは、 ALTSUFFIXオペランドで指定 された接尾部を持つマップ・セットのロードを試みます。

そのようなマップ・セットがない場合、BMSはMまたはLの 接尾部が付加されたマップ・セットのロードを試み、このロードが 失敗した場合、BMSは接尾部がないバージョンのマップ・セット のロードを試みます。

トランザクションがデフォルト画面サイズを使用する場合、 BMSはまず、MまたはLの接尾部が付加されたマップ・セットの ロードを試み、このロードが失敗した場合、BMSは接尾部がない バージョンのマップ・セットのロードを試みます。

接尾部の付いたマップ・セットを使用するには、 BMS=(,,,DDS)システム初期化パラメーターを指定する必要があります。

APLKYBD({**NO**|**YES**})

3270 装置に APL キーボード機能があるかどうかを指定します。

YES 3270装置にはAPLキーボード機構があります。

NO 3270装置にはAPLキーボード機構はありません。

APLTEXT({**NO**|**YES**})

3270 装置に APL テキスト機能があるかどうかを指定します。

YES 3270装置にはAPLテキスト機構があります。

NO 3270装置にはAPLテキスト機構はありません。

TEXTPRINT(YES)の指定の有無に関わらず、3288プリンターにYESを 指定しないでください。APLTEXT機構は、TEXTKYBDおよびAPLKYBD オペランドと組み合わせて使用されます。

QUERY構造化フィールドを使用して、APLテキスト機構を使用するように 装置がセットアップされているかどうかを判別できます。

ASCII({NO|7|8})
端末に ASCII 機能があるかどうかを指定します。

NO この端末にはASCII機構がありません。

7 ASCII-7端末と通信するには、この値を指定します。 ASCII-7 機能で構成された装置は、拡張 3270 機能のない LUTYPE2 または LUTYPE3 でなければなりません。以下の 装置のみがサポートされています。

3274モデル1Cおよび51C
3276モデル12
3278
3287

ASCII-7 オプションで構成された端末では、CICS からのアウトバウンド FM データがすべて ASCII-7 に変換され、CICS へのインバウンド FM データがすべて EBCDIC に変換されます。 FM要求データだけが変換されます。 LU 状況やセンス・データなど RU 内の他のすべてのデータは、出力時には EBCDIC であると想定されます。 ASCII-7 は、構造化フィールドや機能管理ヘッダーなどの拡張属性を含むデータ・ストリームをサポートしていません。

ASCII-7 サポートは、標準マイクロコードの構成で、3274-1C でのオプションとして使用できます。 ASCII-7 オプションの使用は、ここで示された TCT 定義の結果として CICS によって設定された BIND パラメーターによって、セッション開始時に判別されます。

8 ASCII-8 端末と通信するには、これを指定します。ASCII-8 機能で構成される装置には、LUTYPE1、LUTYPE2、または LUTYPE3 があり、拡張 3270 および SCS データ・ストリーム機能の有無は問いません。

ASCII-8 オプションで構成された端末では、CICS からのアウトバウンド FM データがすべて ASCII-8 に変換され、CICS へのインバウンド FM データがすべて EBCDIC に変換されます。すべての FM 要求データが変換されます。 これには、AID、カーソル・アドレス、FM ヘッダー、および構造化フィールドが含まれます。 LU 状況やセンス・データなど RU の他の形式は、入力時には EBCDIC であると想定され、出力時には EBCDIC で伝送されます。

ASCII-8 サポートは、この CICS サポートで行われているように、EBCDIC で作動しながら、データ・ストリームを ASCII-8 との間で変換または再変換する装置のみを対象としています。これは、データ・ストリームが文字ストリングとして扱われ、2 進数フィールドが図形文字のようにバイトごとに変換されるためです。そのため、それらは ASCII 形式のときには真の値を表さないことがあります。

ASCII-8 サポートは、3274 上のマイクロコード RPQ として使用可能であり、ASCII-7 オプションとは相互に排他的です。 ASCII-8 オプションの使用は、ここで示された TCT 定義の結果として CICS によって設定された BIND パラメーターによって、セッション開始時に判別されます。

ATI({NO|YES})
トランザクションが自動トランザクション開始によって端末で開始可能かどうかを指定します。

YES 端末で自動トランザクション初期化によって トランザクションを開始できます。

NO 端末で自動トランザクション初期化によって トランザクションを開始できません。

ATI(YES) を指定すると、一時データ管理によって、または別のトランザクションが発行した EXEC CICS START コマンドによって、端末でトランザクションを開始することができます。すでに端末にトランザクションが 存在する場合、それが終了するまでATIトランザクションは保留に なります。ATI(YES)を指定した場合、ATIおよびIOAREALENによって YESおよび0のデフォルト値が強制されたときのDEVICE(APPC)を除き、 少なくとも1バイトのIOAREALENを指定する必要があります。

ATIがYESと指定され、CREATESESSがYESと指定された場合、 および端末がACQUIREDではないときにトランザクションが開始された 場合、端末は自動的に獲得されます。

TTI属性も参照してください。

AUDIBLEALARM({NO|YES})

音響アラーム機能が 3651 制御装置に接続された 3270 ディスプレイまたは 3270 プリンターにインストールされているかどうかを指定します。

YES 音響アラーム機構がインストールされます。

NO 音響アラーム機構がインストールされません。

AUTOCONNECT({NO|YES|ALL})

自動接続処理が端末に対して行われるかどうかを指定します。 AUTOCONNECT(YES) または (ALL) は、CICS の初期設定中に、あるいは **SET VTAM OPEN**コマンドを使用して z/OS Communications Server との通信を開始したときに、その端末を使用するセッションが確立される (つまり、BIND が実行される) ことを指定します。端末が使用できないためにこの時点で接続を行えない場合、端末がそれまでの間に使用可能になって自動的に通信を開始しない限りは、CEMT SET TERMINAL(*termid*) INSERVICE ACQUIRED コマンドを使用して後でリンクを獲得する必要があります。

注**:** z/OS Communications Server LOGAPPL 機能を使用する場合は、AUTOCONNECT(YES) を指定しないでください。これを指定すると、競合状態になり、エラーや論理装置の停止が発生する可能性があるためです。

NO CICS は、接続の確立時にセッションのバインドを試行しません。

YES CICS は、接続の確立時に競合勝者セッションとしてバインドを試行します。

ALL 適用外

AUTOCONNECT の基本的な情報については、「構成」の「相互通信リソースの定義」 を参照してください。

AUTOPAGE({NO|YES})

BMS 自動ページングを使用するかどうかを指定します。 プリンターには YES、ディスプレイ装置には NO を指定してください。デフォルトは、DEVICE 属性に指定した値によって決まります。デフォルト値については、265 ページの『TYPETERM属性のデフォルト値』 に示されています。

BACKTRANS({NO|YES})
装置に背景透過性機能があるかどうかを指定します。

NO 装置には背景透過性機構は ありません。

YES 装置には背景透過性機構があります。

QUERY構造化フィールドを使用して、背景透過性機構を 使用するように装置がセットアップされているか どうかを判別できます。

BRACKET({YES|NO})
ブラケット・プロトコルをこの論理装置に強制適用するかどうかを指定します。デフォルトは、DEVICE 属性に指定した値によって決まります (265 ページの『TYPETERM属性のデフォルト値』 を参照)。

YES ブラケット・プロトコルが使用されます。このオプションは、 3790の照会およびフル機能論理装置で必要です。多くの DEVICE タイプでは、強制的に BRACKET(YES) になります。

NO ブラケット・プロトコルは使用されません。3614論理装置 および3650ホスト・コマンド・プロセッサー(HCP)セッションに BRACKET(NO)を指定する必要があります。

ブラケット・プロトコルは SNA の機能です。SNA 以外の装置に対して BRACKET(YES) を指定すると、CICS は厳密なブラケット・プロトコルに従わず、実行もしません。

BUILDCHAIN({NO|YES})
CICS が入力データをアプリケーション・プログラムに渡す前にチェーン・アセンブリーを実行するかどうかを指定します。

デフォルトは、DEVICE 属性に指定した値によって決まります。

NO この論理装置からアプリケーション・プログラムが 受信した端末入出力域(TIOA)には、1つの要求単位(RU)が 含まれています。

YES この論理装置からアプリケーション・プログラムが受信した TIOAには、完全なチェーンが含まれています。

CGCSGID({0,0|*value1,value2*})
コード化図形文字セット・グローバル ID (CGCSGID) により、アプリケーション・プログラムは装置でサポートされる文字セットを判別できます。

この情報は、いくつかの装置の QUERY 構造化フィールドから取得できます。その他の装置では、この情報をここで提供することで、アプリケーション・プログラムが **EXEC CICS ASSIGN** コマンドを使用して取得できるようにする必要があります。

0,0 CGCSGID は指定されません。

gcsid,cpgid
CGCSGID は、1 から 65535 の範囲の 2 つの 5 桁の 10 進数で構成されます。 *gcsid* は、図形文字セット・グローバル ID (GCSGID) で、*cpgid* は、セットのコード・ポイントの仕様であるコード・ページ・グローバル ID (CPGID) です。

COLOR({**NO**|**YES**})

3270 装置または SCS プリンターに、フィールドまたは文字ごとの色の選択を可能にする拡張カラー機能があるかどうかを指定します。

NO 装置には拡張カラー機構はありません。

YES 装置には拡張カラー機構があります。

QUERY構造化フィールドを使用して、カラー機構を使用するように 装置がセットアップされているかどうかを判別できます。

COPY({**NO**|**YES**})

3270 ディスプレイまたはプリンターが 3270 制御装置に含まれるかどうかを指定します。

NO コピー機構が組み込まれます。

YES コピー機構が組み込まれません。

3270互換モードの論理装置では、COPY(YES)は無視されるため、 デフォルトのCOPY(NO)のままにしてください。

TERMINAL定義のPRINTERCOPY属性とALTPRINTCOPY属性も 参照してください。

画面コピーについて詳しくは、「CICS 3270 Data Stream Device Guide」を参照してください。

CREATESESS({**NO**|**YES**})

セッションを作成するかどうかを指定します。

NO 内部的に生成されたセッション要求がセッションを作成しないようにするには、これを指定します。 CICS の実行中にこれを変更するには、CEMT コマンドを使用する必要があります。

CREATESESS(NO)は、この端末のEXEC START要求と自動 トランザクション初期化(ATI)要求によるセッションの 作成を防ぎます。すなわち、現在確立されている セッションがない場合は、要求はキューに入れられるか、 リジェクトされます。

YES 内部で生成されたセッション要求によるセッションの作成を 許可する状況にするには、これを指定します。CICS 実行中にこの状況を生成するには、CEMT コマンドを使用する必要があります。

CREATESESS(YES)は、この端末のEXEC START要求と自動 トランザクション初期化(ATI)要求によるセッションの 自動作成を許可します。

DEFSCREEN(*rows* , *columns*)

トランザクションに接続している場合、あるいはプロファイル定義でデフォルトの画面サイズが指定されている BMS によって使用されている場合に、その装置で使用する 3270 画面サイズまたは 3270 プリンター・ページ・サイズを指定します。 *rows* と *columns* の値は、どちらも 0 から 255 の範囲でなければなりません。デフォルトは、DEVICE 属性に指定した値によって決まります (265 ページの『TYPETERM属性のデフォルト値』 を参照)。 BSC 3270に指定可能な値は以下のとおりです。

装置	画面サイズ
3278-1	(12,40)
3278-2	(24,80)
3276-3, 3278-3	(24,80)
3276-4, 3278-4	(24,80)
3278-5	(24,80)
3279-2A、3279-2B	(24,80)
3279-3A、3279-3B	(24,80)

BSC装置の場合、デフォルト画面サイズと代替画面サイズは どちらも端末のハードウェアによって決まります。デフォルト 画面サイズは(24,80)です。ただし、3278-1の場合は異なり、 (12,40)となります。

SNA装置(LUTYPE2およびLUTYPE3)の場合、最大物理画面サイズ までの任意の値を選択して、デフォルト画面サイズと 代替画面サイズの両方に指定できます（ALTSCREENを参照）。 特に、デフォルト画面サイズと代替画面サイズの両方を 最大画面サイズにしたり、代替画面サイズを指定せずに デフォルト画面サイズを最大画面サイズにしたりすることが できます。SNA バインドは、この TCT 情報から CICS によって生成されます。ログモード・テーブル・エントリーを提供したり、装置をカスタマイズしたりする必要はありません。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

DEVICE(*device*)
この TYPETERM が定義する装置タイプを指定します。この属性は すべてのTYPETERM定義に必須です。

コマンド行で DEVICE(*xxxx*) を入力する場合 (*xxxx* は有効な装置タイプ)、必要に応じて SESSIONTYPE および TERMMODEL を一緒に指定すると、他の属性には適切なデフォルト値が指定されます。詳しいガイダンスについては、265 ページの『TYPETERM属性のデフォルト値』を参照してください。上書き修正パネル DEVICE、SESSIONTYPE、または TERMMODEL の値を入力あるいは上書きしても、これらのデフォルトは指定されません。

有効な属性および各装置タイプのデフォルトは、 265 ページの『TYPETERM属性のデフォルト値』 にリストされています。SNA z/OS Communications Server 3270 以外の装置に推奨される属性は、ディスプレイの場合は 3270、プリンターの場合は 3270P です。前のリリース との互換性のために、以下の属性を指定して保存することもできます。

- ディスプレイ：3277およびL3277
- プリンター：3284およびL3284, 3286およびL3286

SNA z/OS Communications Server 3270 装置の場合は、必要に応じて LUTYPE2 属性または LUTYPE3 属性を使用します。 LUTYPE2論理装置はSNAによって定義された装置で、 3270ディスプレイ・データ・ストリームを受

信します。 LUTYPE3論理装置はSNAによって定義された装置で、 3270プリンターと同じデータ・ストリームを受信します。

CICS でサポートされる装置タイプのリストについては、 471 ページの『DFHTCT: CICS 端末リスト』 を参照してください。有効な装置名およびそれらが生成するデフォルト属性のリストについて、 265 ページの『TYPETERM 属性のデフォルト値』 も参照してください。

DISCREQ=({YES|NO})
切断要求を受け入れるかどうかを指定します。

YES CICS は、z/OS Communications Server 装置に対する切断要求を受け入れ、z/OS Communications Server CLSDST マクロ命令を発行して、その論理装置で z/OS Communications Server セッションを終了します。

さらに、YESを指定すると、端末からのCESF LOGOFFまたはGOODNIGHTによって切断が実行されます。

TYPETERM定義がAUTINSTMODEL TERMINAL定義によって参照される場合、 自動インストール端末の項目を自動的に削除するためにはYESが必須です。

デフォルトは YES ですが、状況によっては、リソース定義でこの属性が指定されていない場合に、リソース定義内の他のオプションとの互換性を確保するために必要であれば、CICS が強制的に NO 値を指定することがあります。

NO CICS は、z/OS Communications Server 装置に対する切断要求を受け入れません。

DUALCASEKYBD({NO|YES})
3270 表示装置にタイプライター型キーボードまたはオペレーター・コンソール・キーボードのどちらがあるかを指定します。 これらのキーボードのどちらでも、大文字データと小文字データの両方を伝送できます。

NO 装置には大／小文字が使用できるキーボードがありません。

YES 装置には大／小文字が使用できるキーボードがあります。

ERRCOLOR({NO|*color*})
エラー・メッセージをカラーで表示するかどうかを指定します。
ERRCOLOR(*color*) をコーディングすると、暗黙的に ERRLASTLINE(YES) が指定されます。

指定できる色は以下のとおりです。

BLUE
RED
PINK
GREEN
TURQUOISE
YELLOW
NEUTRAL

ERRHILIGHT({NO|BLINK|REVERSE|UNDERLINE})
エラー・メッセージを表示するための強調表示 (存在する場合) を指定します。

ERRINTENSIFY({NO|YES})
エラー・メッセージを高輝度フィールドで表示するかどうかを指定します。 ERRINTENSIFY(YES)をコーディングすると、 ERRLASTLINE(YES)が暗黙指定されます。

ERRLASTLINE({NO|YES})
エラー・メッセージを表示する場所を指定します。

NO エラー・メッセージは現行カーソル位置に追加属性なしで 表示されます。

YES エラー・メッセージは、全メッセージが画面に収まるように、 画面下部に最も近い行の先頭から開始して 表示されます。

すべてのエラー・メッセージは同じ行に表示されるので、複数のメッセージが高速に次々と受け取られる場合、それらは互いに重複して前のメッセージが読まれる前に消去されることがあります。

EXTENDEDDS({NO|YES})
3270 装置または SCS プリンターが、3270 データ・ストリームへの拡張をサポートするかどうかを指定します。

NO 装置は3270データ・ストリーム拡張をサポートしません。

YES 装置は3270データ・ストリーム拡張をサポートします。

COLOR, HILIGHT, PROGSYMBOLS, QUERY,またはVALIDATION (3270のみ） の属性のうちのいずれかをYESに指定すると、EXTENDEDDS(YES)が 暗黙指定されます。

拡張データ・ストリーム(EXTENDEDDS)がYESに設定されている場合、 装置は構造化フィールド書き込みCOMMANDおよび アウトバウンドQUERY構造化フィールドをサポートします。

QUERY構造化フィールドを使用して、装置が拡張データ・ストリームを 使用するようにセットアップされているかどうかを判断できます。 照会が有効な場合、QUERY構造化フィールドを使用すると、 EXTENDEDDSがYESに設定されます。

FMHPARM({NO|YES})
BMS によって作成された機能管理ヘッダーに含めるためのユーザー提供のパラメーターを、BMS が受け入れるかどうかを指定します。

NO BMSによって作成された機能管理ヘッダーに含まれる ユーザー提供パラメーターを受け入れません。

YES BMSによって作成された機能管理ヘッダーに含まれる ユーザー提供パラメーターを受け入れます。

DEVICE タイプが 3650 の場合にのみ、YES を指定します。

FORMFEED({NO|YES})
装置に用紙送り機能があるかどうか、つまり BMS が出力文書をフォーマットする際に用紙送り文字を使用するかどうかを指定します。

NO 装置には用紙送り機能がありません。

YES 装置には用紙送り機能があります。

DEVICE(SCSPRINT)が指定されていると、BMSはデータ・ ストリームの先頭に用紙送り文字を挿入します。これにより 装置は印刷を開始する前に新規ページの上部マージンに スキップします。

上部マージンは垂直方向の書式設定(SVF)データ・ストリーム によって定義されます。行番号は1以上になります。 SVFデータ・ストリームがプリンターに送信されていない場合、 上部マージンは行1になります。オペレーターが 用紙をセットアップするとき、装置の行カウンターは 1に設定されます。

下部余白を越えて印刷しようとする場合も、装置は 自動用紙送りを実行する場合があります。また、 下部余白はSVFデータ・ストリームによっても決定されます。 デフォルト値は最大表示行(MPL)です。MPLは ページの最終行であり、その値はページまたは用紙の長さ を行数で表します（つまり、物理ページ・サイズと 行密度との積です）。MPL と行密度は、どちらも SVF データ・ストリームによって決定することができます。あるいは、オペレーターが装置に対して MPL (行数) をセットアップすることもできます。

DEVICE(3270), DEVICE(3270P), DEVICE(LUTYPE2),または DEVICE(LUTYPE3)が指定されている場合、BMS SENDコマンドの FORMFEEDオプションとともにFORMFEED(YES)を 使用します。ディスプレイ装置で用紙送りを使用すると、 画面データがプリンターにコピーされたときに新規ページに スキップします。

ここで説明した SCSPRINT 操作のオプションは、装置が 3270P 装置または LUTYPE3 装置として作動している場合には適用されません。この場合、存在するのは用紙の 長さという概念のみであり、これを装置に設定できるのは オペレーターのみです。

GROUP(*groupname*)

各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

HILIGHT({**NO**|**YES**})

3270 装置または SCS プリンターに、フィールドまたは文字を反転表示、下線モード、または明滅 (3270 のみ) で表示するための拡張強調表示機能があるかどうかを指定します。

NO 装置には拡張強調表示機能がありません。

YES 装置には拡張強調表示機能があります。

QUERY構造化フィールドを使用して、装置が拡張強調表示機能を 使用するようにセットアップされているかどうかを判断できます。

HORIZFORM({**NO**|**YES**})

装置に水平フォーム機能があるかどうか、つまり BMS が出力文書をフォーマットする際に水平タブを使用する必要があるどうかを指定します。

NO 装置には水平書式機能がありません。

YES 装置には水平書式機能があります。

この機能を使用できる装置は、バッチ、バッチ・データ交換、 対話式、SCSPRT、またはLUTYPE4の論理装置です。

NO BMSマップ定義のHTABオプションは無視されます。

YES BMSは出力文書の書式設定時に水平タブを使用します。

IOAREALEN({**0**|*value1*},{**0**|*value2*})

トランザクションに渡す端末入出力域の長さ (バイト数) を指定します。

ATI(YES)を指定する場合、IOAREALENの値は最低1バイトに 指定する必要があります。

value1 *Value1* は、**RECEIVE** コマンドが発行されたときにアプリケーション・プログラムに渡される端末入出力域の最小サイズを指定します。

value2 *value2* には、 *value1* 以上の値を指定できます。この場合、入力メッセージのサイズが *value1* を超えると、CICS は、端末入出力域の長さに *value2* バイトを使用します。入力メッセージ・サイズも *value2* を超えた場合、ノード異常条件プログラムが例外応答を端末に送信します。

value2 が指定されていない場合、または *value1* より小さい場合は、デフォルトで *value1* の値が使用されます。

IOAREALEN に指定できる最大値は、32767 バイトです。

KATAKANA({**NO**|**YES**})

英数カナ文字モードのサポートが必要かどうかを指定します。 英数カナ文字モード端末は、大/小文字混合の出力を表示できません。大文字は英語の大文字として表示されますが、小文字はカタカナ文字として表示されます。 CICS システムに英数カナ文字モード端末が接続されている場合は、MSGCASE システム初期設定パラメーターに UPPER を指定してください。

NO カタカナのサポートは不要です。

YES カタカナのサポートは必須です。以下の トランザクションから端末に送信されるすべての小文字は 大文字に変換されます。

CBRC CDBC CDBI CEBR CECI CEDA CEDF CEMT CEOT CESN CEST CMSG CRTE CSPG CWTO

重要**:** エミュレートされた英数カナ文字モード端末の場合、使用するコード・ページによって結果が変わります。場合によっては、小文字の英語文字が大文字に変換されないことがあります。

LDCLIST(*list*)

論理装置コード (LDC) リストの名前を指定します。名前の最大長は 8 文字です。名前はアセンブラー言語の規則に従います。名前の先頭は英字にする必要があります。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

LDCLISTとその内容はマクロ命令を使用して定義します。

ローカルLDCリストは以下のように定義します。

```
listname  DFHTCT TYPE=LDCLIST,
LDC(aa=nnn,bb=nnn,....)
```

拡張ローカルLDCリストは以下のように定義します。

```
listname  DFHTCT TYPE=LDC,LOCAL=INITIAL
          DFHTCT TYPE=LDC=(aa=nnn)....
          DFHTCT TYPE=LDC=(bb=nnn)....
          DFHTCT TYPE=LDC,LOCAL=FINAL
```

TYPETERM 定義の LDCLIST 属性の値として、この *listname* を指定します。

この属性は、3600, 3770バッチ、3770および3790バッチ・ データ交換、およびLUTYPE4論理装置にのみ適用されます。 このリストはこの論理装置で有効なLDCを指定し、 オプションで各LDCで有効な装置特性を指定します。 CICSは、論理装置のデフォルト LDC を選択する際に、このリスト内の最初に生成された LDC を使用します。詳しいガイダンスについては、 453 ページの『DFHTCT 論理装置コード: z/OS Communications Server 非 3270』を参照してください。

LIGHTPEN({**NO**|**YES**})

3270 表示装置に選択ペン機能があるかどうかを指定します。

NO 3270ディスプレイに選択ペン機能はありません。

YES 3270ディスプレイに選択ペン機能があります。

LOGMODE({**blank**|**0**|*logmode*})

論理装置に送信される BIND を CICS がビルドする方法を指定します。

ブランク

定義済みの端末定義では、この TYPETERM 定義および関連した端末定義を使用してこの装置用に CICS 定義によって生成された BIND イメージを使用します。自動インストール端末は、 入力CINITに指定されたフィールドを使用します。

name これは、この論理装置で使用するためにセットアップされた z/OS Communications Server ログオン・モード・テーブルの LOGMODE 名です。 名前は最大8文字です。また、アセンブラー言語の規則に 従う必要があります。名前の先頭文字は 英字である必要があります。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

これにより、論理装置用に CICS によって提供された BIND イメージ

をオーバーライドすることができます。詳しくは、該当する CICS サブシステムの資料を参照してください。

端末がクロスドメイン資源である場合、 ログモード名をコーディングすることはできません。

TCTTEは、ログモード・バインド・イメージ・フィールドを反映 するように更新されています。これらのフィールドには、SEND サイズおよび RECEIVE サイズと、デフォルト画面サイズおよび代替画面サイズが含まれます。ログモードが端末照会不能を示している場合、 拡張データ・ストリーム・フィールドはすべてゼロに設定されます。

0（ゼロ）

これを指定すると、CICS は、論理装置から送信された CINIT に含まれる BIND イメージからの情報の一部を使用します。 CINIT の BIND イメージは、CICS へのログオンを要求している論理装置用に定義された LOGMODE エントリーに基づいて、z/OS Communications Server が作成したものです。ノード初期設定ブロック (NIB) は、LOGMODE=0 および BNDAREA=0 を指定して作成されます。TYPETERM の SENDSIZE および RECEIVESIZE にゼロが指定された場合、CICS は、それらを LOGMODE の RUSIZES の値に置換します。

TCTTEは、入力CINITフィールドを反映するように更新されています。これらのフィールドには、SEND サイズおよび RECEIVE サイズと、デフォルト画面サイズおよび代替画面サイズが含まれます。ログモードが 端末照会不能を示している場合、 拡張データ・ストリーム・フィールドはすべてゼロに設定されます。 例外的な環境でのみLOGMODE(0)を使用します。LU は z/OS Communications Server 定義にバインドされますが、CICS は、CICS 定義からの主なセッション特性を保持します。例えば、z/OS Communications Server に対しては LUTYPE1 としてプリンターが定義されているが、CICS に対しては LOGMODE(0) が指定された LUTYPE3 としてプリンターが定義されている場合、CICS は、バインドを受け入れますが、LUTYPE3 制御文字をプリンターに送信し、誤った結果が生じることになります。この制限は パイプライン端末には適用されません。

注**:**

1. この値は、例外的な環境でログモード属性にのみ 使用してください。
2. クロスドメイン環境内の論理装置の場合は、LOGMODE(0) を指定し、z/OS Communications Server(LU) ステートメントの DLOGMOD オペランドおよび MODETAB オペランドに論理装置モード情報を提供します。クロスドメイン環境では、LOGMODE に名前が指定されていると、z/OS Communications Server でエラーが発生します。

LOGMODECOM

この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

LOGONMSG({NO|YES})

GMTRAN システム初期設定パラメーターで指定された「good morning」トランザクションが以下のようになるかどうかを指定します。

- 論理装置が z/OS Communications Server 経由で最初に CICS にログオンしたときに、自動的に開始される。
- 特定の条件下で端末ユーザーの TIMEOUT 期間が経過したときに開始される。

ERRLASTLINE(YES)が指定されていると、 そのトランザクションによって書き込まれたメッセージは エラー・メッセージ行を上書きしません。

NO CICS は、「Good Morning」トランザクションを実行しません。

注: Telnet 3270のような非SNA端末を使用している場合、LOGONMSG(NO)を指定してもキーボード・ロックは自動的に 解放されません。キーボード・ロックを解放するには リセット・キーを押す必要があります。

YES CICS は、OPNDST 出口が正常に完了し、セッションが確立されたときに「Good Morning」トランザクションを実行します。トランザクションは 自動タスク開始(ATI)によって開始され、端末を使用するために他のATIトランザクションと競合します。 このTYPETERMには ATI(YES)を指定してください。

注: Telnet 3270のような非SNA端末を使用している場合、LOGONMSG(YES)を指定してもキーボード・ロックは自動的に解放されます。

MSRCONTROL({NO|YES})

端末 8775 または 3643 に磁気スロット読取装置があるかどうかを指定します。このオプションは、SCS プリンターでは無効です。

QUERY構造化フィールドを使用して、装置が磁気スロット読取装置を 使用するようにセットアップされているかどうかを判断できます。

NEPCLASS({0|*tranclass*})

ノード・エラー・プログラム・トランザクション・クラスを指定します。

0 デフォルトのノード・エラー・プログラム・モジュールへのリンクになります。

tranclass

(デフォルトではない) ノード・エラー・プログラム・モジュールのトランザクション・クラス。 *tranclass* は、1 から 255 の範囲で指定することができます。ノード・エラー・プログラムのプログラミング情報については、ノード・エラー・プログラムの作成 を参照してください。

OBFORMAT({NO|YES})

外部様式設定機能が使用されるかどうかを指定します。 このTYPETERMを定義している装置がBMS外部フォーマット設定を 使用する場合、OBFORMAT(YES)を設定します。OBFORMAT(YES) は以下の2つの装置タイプにのみ指定できます。

- 3650, SESSIONTYPE(3270)

- 表示サービスのために DPPX/DPS バージョン 2 がある DPPX オペレーティング・システムを使用する 8100 情報システムでは、LUTYPE2。

QUERY構造化フィールドを使用して、装置が外部フォーマット設定を 使用するようにセットアップされているかどうかを判断できます。

OBOPERID({**NO**|**YES**})
CICS が外部オペレーター ID を使用してこの端末に必要な BMS ルーティング機能をサポートするかどうかを指定します。このオプションは、3790 および 3770 バッチ・データ交換論理装置にのみ適用されます。

NO CICS は、外部オペレーター ID を使用しません。

YES CICS は、外部オペレーター ID を使用します。

OUTLINE({**NO**|**YES**})
装置がフィールド外形線をサポートするかどうかを指定します。

NO 装置はフィールド・アウトラインをサポートしません。

YES 装置はフィールド・アウトラインをサポートします。

QUERY構造化フィールドを使用して、装置がフィールド・アウトラインを 使用するようにセットアップされているかどうかを判断できます。

PAGESIZE(*rows , columns*)
このプリンターのデフォルトのページ・サイズを指定します。 *rows* と *columns* の値は、どちらも 0 から 255 の範囲でなければなりません。 DEFSCREEN属性でデフォルト画面サイズが選択されている場合、 このデフォルト・ページ・サイズがBMSによって使用されます。

rows ページの行数を示します。印刷と表示の両方で同じ BMS マップを使用する場合、PAGESIZE *rows* 値を DEFSCREEN *rows* 値より小さくすることで、エラー・メッセージ用に画面の一番下の行を予約することができて (ERRLASTLINE 属性を参照) 有用です。

columns
各行の文字数を示します。 PAGESIZE で指定された *columns* 値が DEFSCREEN で指定された *columns* 値と異なっていると、予期しない結果が発生します。

rows と *columns* の積が 32767 を超えてはなりません。

デフォルト値は、DEVICE属性に指定する値によって 異なります。詳しくは、265 ページの『TYPETERM属性のデフォルト値』を参照してください。

BMSは、出力データ・ストリームを準備するときに このページ・サイズ値を使用します。ページの各行に 指定する文字数は、端末の物理的な行の幅を超えないように してください。キャリッジの終わりに達すると 自動的に改行機能を実行するプリンターの場合 （例えば3270プリンターなど）、 ここで指定する行の幅は物理的な行の幅よりも 狭くする必要があります。

これにより、出力データのフォーマット設定は、 装置自体によって実行される改行機能ではなく、 BMSまたはユーザーが指定する改行(NL)文字によって 完全に制御されます。そのため、出力に追加行が発生し、 物理的なページの縦の長さがここで指定したよりも長くなることが あります。

3270 タイプのプリンターの場合、ハードウェアによって BMS が送信できるデータ量が制限されます。マップ要求またはアプリケーション・プログラム要求で L40、L64、または L80 が指定された場合、あるいは SEND MAP コマンドで NLEOM が指定されなかった場合は、PAGESIZE で指定される *lines* と *columns* の積がバッファー・サイズを超えてはなりません。

BMS で NLEOM が指定された場合は、ページの長さを任意の行数にすることができますが、DEFSCREEN 属性あるいは ALTSCREEN 属性で指定される *lines* と *columns* の積が装置のバッファー・サイズを超えてはなりません。つまり、BMS が送信する文字数が、プリンターの物理バッファー・サイズを超えてはなりません。

注**:** BMSは、送信のために大きいページを小さいセグメントに 分割します。したがって、PAGESIZE は、必要な論理ページ・サイズ (*lines* と *columns* の積) に対応している必要があり、DEFSCREEN は、実際のバッファー・サイズに対応している必要があります。

z/OS Communications Server 3600 の場合、論理装置コード (LDC) を指定せずに BMS ページ・ビルド操作が試行されると、指定された PAGESIZE が使用されます。デフォルト装置タイプは 3604です。

3770, LUTYPE4,または3790バッチ・データ交換論理装置の場合、 論理装置コード(LDC)を指定しないで BMSページのビルド操作が要求された場合、 指定されたPAGESIZEが使用されます。デフォルト装置タイプは コンソール・プリンターです。

メッセージを端末リストにルーティングする場合は注意が必要です。 定義した（またはデフォルトの）PAGESIZEが小さすぎて メッセージを格納できない場合、トランザクションは 異常終了します。

累積テキスト処理の場合、許容される最大バッファー・サイズは 32767です。これを超える場合、BMSは内部で強制的に ページの長さを削減し、PAGESIZEが制限を 超えないようにします。

PARTITIONS({NO|YES})

装置が区画化を使用するかどうかを指定します。 このオプションは、SCS プリンターでは無効です。

NO 装置はパーティションを使用しません。

YES 装置はパーティションを使用します。

QUERY構造化フィールドを使用して、装置がパーティションを使用する ようにセットアップされているかどうかを判断できます。

PRINTADAPTER({NO|YES})

3275 の場合: プリンター・アダプター機能および対応する 3284 プリンター・モデル 3 が 3275 ディスプレイ装置にあるかどうかを指定します。 この機能により、ホスト3275のPAキーを使用した印刷要求に対して 3284が適格になります。

NO プリンター・アダプター機能と対応する3284プリンター・ モデル3は使用できません。

YES プリンター・アダプター機能と対応する3284プリンター・ モデル3を使用できます。

LUTYPE2 論理装置の場合: PRINT キーまたは ISSUE PRINT コマンドによって開始された印刷要求に対して、z/OS Communications Server 接続と非 z/OS Communications Server 接続の両方のプリンター許可マトリックスに従って、プリンター割り振りが 3790 によって処理されるか、あるいは 3274 または 3276 によって処理されるかを指定します。

NO 印刷要求は、z/OS Communications Server 接続および非 z/OS Communications Server 接続の両方のプリンター許可マトリックスに従って処理されません。

YES 印刷要求は、z/OS Communications Server 接続および非 z/OS Communications Server 接続の両方のプリンター許可マトリックスに従って処理されます。

さらに、端末制御印刷要求によって3270ディスプレイ論理装置に 送信された印刷要求や、オペレーターによって開始された印刷要求では、 同じ3790に接続された3270プリンターが使用できます。 PRINTADAPTERがNOの場合、プリンター割り振りは TERMINAL定義のPRINTER属性とALTPRINTER属性によって決定します。

PRINTオプション付きBMS要求、NLEOMオプション付きBMS要求、 またはCMSGコマンドによって出力が画面上に作成される場合、 CICS定義PRINTキー（通常PAキー）が押されたかどうかに関わらず、 画面の内容が自動的に3270プリンターにコピーされます。

PROGSYMBOLS({NO|YES})

プログラム式シンボル (PS) 機能をこの 3270 装置または SCS プリンターで使用できるかどうかを指定します。この機能によって保管および アクセスできるのは、ユーザー定義およびプログラム・ロードのフォント およびコードを持つ6つまでの191文字セットです。

NO プログラム式シンボル(PS)機能は使用できません。

YES プログラム式シンボル(PS)機能を使用できます。

QUERY構造化フィールドを使用して、装置がプログラム式シンボルを 使用するようにセットアップされているかどうかを判断できます。

QUERY({NO|COLD|ALL})

CICS が装置の特性を判別するために QUERY 構造化フィールドを使用するかどうかを指定します。

NO CICS は、QUERY 機能を使用しません。

COLD

CICS は、CICS の初期始動またはコールド・スタートの後に装置が初めて接続されたときにのみ、QUERY 機能を使用して装置の特性を判別します。装置特性は、後続のウォーム・スタートおよび緊急時始動で使用するために、CICS グローバル・カタログに保管されます。

ALL CICS は、装置が接続されるたびに、QUERY 機能を使用して装置の特性を判別します。

RECEIVESIZE(*number*)

定義済みの非自動インストール端末の場合、z/OS Communications Server RECEIVE 要求を満たすことができる要求単位の最大サイズを指定します。

RECEIVESIZE 値は、接続された論理装置に送信され、その範囲は 0 から 30720 でなければなりません。この値は、体系化された形式で送信する必要があるため、CICS によって端数が切り捨てられる場合があります。

RECEIVESIZE の効果は、RECEIVE RUSIZE が z/OS Communications Server LOGMODE テーブルに存在しているかどうかによって異なります。 表 7 は、セッションをバインドするために使用される RECEIVE RUSIZE について、TYPETERM と LOGMODE の値の考えられる組み合わせごとに示しています。

表 *7.* 定義済み（非自動インストール）端末の*RECEIVE RUSIZE*

RECEIVE RUSIZE (VTAM)	**TYPETERM RECEIVESIZE**	バインドに使用される **RUSIZE**
0	0	0
0	指定	TYPETERM RECEIVESIZE サイズ
指定	0	この組み合わせは無効であるため、メッセージ DFHZC2403 でバインドは失敗します
指定	指定	TYPETERM RECEIVESIZE サイズ
注: このテーブルに対する例外は LOGMODE(0) です。 TYPETERM 定義でこれを指定した場合、他に指定されている値に関係なく、VTAM 値が使用されます。		

APPC 端末 APPC (LUTYPE6.2) 単一セッション端末の場合、256 が適切な値です。

自動インストール端末: 自動インストール端末の場合、ゼロ以外の RECEIVESIZE 値は、この TYPETERM で定義された論理装置のセッションをバインドするために使用される RECEIVE RUSIZE の最大値あるいは実際の値を指定します。

RECEIVESIZE の効果は、RECEIVE RUSIZE が z/OS Communications Server LOGMODE テーブルに存在しているかどうかによって異なります。 表 8 は、セッションをバインドするために使用される RECEIVE RUSIZE について、TYPETERM と LOGMODE の値の考えられる組み合わせごとに示しています。

表 *8.* 自動インストール端末の*RECEIVE RUSIZE*

RECEIVE RUSIZE (VTAM)	**TYPETERM RECEIVESIZE**	バインドに使用される **RUSIZE**
0	0 および BUILDCHAIN(YES)	256
0	0 および BUILDCHAIN(NO)	0
0	指定	TYPETERM RECEIVESIZE サイズ
指定	0	VTAM RECEIVE RUSIZE サイズ

表 8. 自動インストール端末の*RECEIVE RUSIZE*（続き）

RECEIVE RUSIZE (VTAM)	TYPETERM RECEIVESIZE	バインドに使用される RUSIZE
TYPETERM RECEIVESIZE 以下の値を指定	指定	VTAM RECEIVE RUSIZE サイズ
TYPETERM RECEIVESIZE より大きい値を指定	指定	この組み合わせは無効であるため、メッセージ DFHZC5963 が発行されます

RECOVNOTIFY({NONE|MESSAGE|TRANSACTION})

持続セッション・サポートありで稼働している CICS 領域では、このオプションは、端末セッションがリカバリーされたことを端末エンド・ユーザーに通知する方法を指定します。XRF サポートありで稼働している CICS 領域では、このオプションは、XRF テークオーバーが発生したことを端末ユーザーに通知する方法を指定します。

このオプションはAPPCセッションには適用できません。

NONE
端末セッションがリカバリーされたこと、あるいは XRF テークオーバーが発生したことを通知しません。

MESSAGE
システムがリカバリーされたことを示すメッセージが画面に表示されます。 XRF の場合は 2 つの BMS マップ (DFHXRC1 および DFHXRC2) で、z/OS Communications Server 持続セッションの場合は 1 つの BMS マップ (DFHXRC3) でメッセージが指定されます。これらのマップはマップ・セット DFHXMSG にあります。テークオーバー時間の短縮が重要である場合は、TRANSACTION ではなく MESSAGE を使用してください。

端末は ATI(YES) オプションを指定して定義する必要があり、BMS マップの表示に対応している必要があります。

TRANSACTION
端末でトランザクションが開始されます。トランザクションの名前は、**RMTRAN** システム初期設定パラメーターによって指定されます。 z/OS Communications Server 持続セッションの場合、**RMTRAN** システム初期設定パラメーターで指定された最初のトランザクション名のみが使用されます。

ヒント: RMTRAN のデフォルト・トランザクションは、**GMTRAN** システム初期設定パラメーターで指定された good-morning トランザクションです。

TRANSACTION オプションでは、端末は ATI(YES) オプションを指定して定義する必要があります。テークオーバー時間の短縮が重要である場合は、TRANSACTION ではなく MESSAGE を使用してください。

RECOVOPTION({SYSDEFAULT|CLEARCONV| RELEASESESS|UNCONDREL|NONE})
このオプションは、z/OS Communications Server 持続セッションまたは XRF で実行されている CICS 領域内のセッションのリカバリーに適用されます。

持続セッション・サポートありで稼働している CICS 領域では、このオプションは、CICS がセッションをリカバリーし、持続セッション遅延間隔内のシステム再始動時に端末をサービスに返す方法を指定します。

NONE を除くすべてのリカバリー・オプションでは、実行されるアクションが z/OS Communications Server UNBIND である場合、UNBIND に続いて z/OS Communications Server SIMLOGON が実行されます。

デフォルトは SYSDEFAULT ですが、状況によっては、リソース定義でこの属性が指定されていない場合に、リソース定義内の他のオプションとの互換性を確保するために必要であれば、CICS が強制的に NONE 値を指定することがあります。

SYSDEFAULT

持続セッション・サポートありで稼働している CICS 領域では、このオプションは、持続セッション遅延間隔内のシステム再始動時に、セッション・アクティビティーおよび端末の特性に応じて CICS が最適な手順を選択してセッションをリカバリーすることを指定します。

セッションはリカバリーされますが、障害が発生したときに実行中だったトランザクションは異常終了し、リカバリーされません。トランザクションは、リカバリーされたセッションが APPC 接続による別の CICS 領域で使用されている場合にも異常終了します。

CICS は、以下のいずれかの方法によって、考えられる影響が最も小さいセッションをリカバリーします。

- CICS の障害発生時に端末でトランザクションが実行されていなかった場合、リカバリー・アクションは不要です。CICS は、RECOVNOTIFY 属性で定義された適切なリカバリー通知アクションを実行します。
- CICS の障害発生時に端末が使用中であった (トランザクションを実行していた) 場合、CICS は、最初に z/OS Communications Server ブラケット終了標識を送信してセッションをリカバリーすることを試みます。ブラケット終了でセッションがリカバリーされない場合 (CICS が RECEIVE モードになっている場合など)、CICS は CLEARコマンドを発行します。端末が CLEAR コマンドをサポートしていない場合に実行されるアクションは、z/OS Communications Server UNBIND とその後に続く SIMLOGON です。

  持続セッションについて詳しくは、アプリケーション設計上の考慮事項 を参照してください。

CLEARCONV

CICS は、ブラケット内セッションをクローズするためにブラケット終了標識を送信しません。代わりに、CICS は CLEAR 要求を送信して、会話状態をリセットします。セッションが CLEAR 要求をサポートしない場合、CICS は UNBIND 要求を送信します。 CLEAR または UNBIND が送信されるのは、システム再始動 (持続セッションの場合) またはテークオーバー (XRF の場合) の際にセッションが使用中であった場合のみです。

RELEASESESS

UNBIND 要求を送信してアクティブ・セッションを解放するように、CICS に要求します。 UNBIND が送信されるのは、システム再始動 (持続セッションの場合) またはテークオーバー (XRF の場合) の際にセッションが使用中であった場合のみです。 UNBINDの後、セッションはSIMLOGONのキューに入ります。 セッションが使用中でなければ、要求されたリカバリー通知が 実行されます。

UNCONDREL

UNBIND 要求を送信してアクティブ・セッションを解放するように、CICS に要求します。システム再始動 (持続セッション・サポートの場合) またはテークオーバー (XRF の場合) の際にセッションが使用中であったかどうかに関係なく、UNBIND が送信されます。 UNBINDの後、セッションはSIMLOGONのキューに入ります。

NONE

持続セッション・サポートありで稼働している CICS 領域では、これは、持続セッション遅延間隔内のシステム再始動時に端末をリカバリーしないことを指定します。事実上、端末で持続セッションがサポートされないことになります。 LU6.2 セッションはアンバインドされますが、最後にネゴシエーションされた CNOS 値が、再始動後に CICS システムに返されます。 AUTOCONNECT(YES) が指定されている場合、システムの再始動後に、**AIRDELAY** システム初期設定パラメーターの操作に従って端末が自動的に再接続されます (AIRDELAY=0 が指定されている場合、AUTOCONNECT(YES) がオーバーライドされ、端末は再接続されません)。

この端末あるいは自動インストール・モデルが持続セッション (SIT の **PSDINT** = nnn) ありで使用されているが、端末が **EXEC CICS ISSUE PASS LUNAME() LOGONLOGMODE** の対象である場合は、RECOVOPTION(NONE) を使用します。

RELREQ=({**NO**|**YES**})

別の z/OS Communications Server アプリケーション・プログラムによる要求時に、CICS が論理装置を解放するかどうかを指定します。

NO CICS は論理装置を解放しません。

YES CICS は、論理装置が現在トランザクションの一部ではない場合、論理装置を解放します。

ROUTEDMSGS({**ALL**|**NONE**|**SPECIFIC**})

EXEC CICS ROUTE コマンドによってどのメッセージがこの端末に送信されるかを指定します。デフォルトは、DEVICE 属性に指定した値によって決まります。詳しくは、 265 ページの『TYPETERM属性のデフォルト値』を参照してください。

ALL BMS は、明確にこの端末に宛てられたメッセージだけではなく、すべての端末に宛てられたメッセージもこの端末に送信します。

なし メッセージがすべての端末に宛てられているか 特にこの端末に宛てられているかに関わらず、 BMSはこの端末にメッセージを送付しません。

SPECIFIC

BMS は、明確にこの端末に宛てられている場合にはメッセージをこの端末に送信しますが、すべての端末に宛てられている場合にはこの端末に送信しません。

RSTSIGNOFF({**NOFORCE**|**FORCE**})

持続セッションの再始動あるいは XRF のテークオーバーが発生した場合に、端末ユーザーをサインオフさせるかどうかを指定します。

FORCE

持続セッション再開始またはXRFテークオーバーの後、 端末はサインオフされます。

NOFORCE

RSTSIGNOFF システム初期設定パラメーターと、RACF ユーザー・プロファイルの CICS セグメント内の XRFSOFF エントリーの両方が NOFORCE に設定されている場合は、持続セッションの再始動や XRF テークオーバーが行われた後も、端末はサインオン状態を維持します。

SENDSIZE(*number*)

定義済みの端末 **(**非自動インストール**)**: 非自動インストール端末の場合、これは、z/OS Communications Server **VTAM SEND** 要求を満たすことができる要求単位の最大サイズ (バイト数) を指定します。SENDSIZEの値は接続されている論理装置に 送信されます。この値の範囲は0から30720の範囲である必要が あります。この値は、体系化された形式で送信する必要があるため、CICS によって端数が切り捨てられる場合があります。

SENDSIZE の効果は、SEND RUSIZE が z/OS Communications Server LOGMODE テーブルに存在しているかどうかによって異なります。 表 9 は、セッションをバインドするために使用される SEND RUSIZE について、TYPETERM と LOGMODE の値の考えられる組み合わせごとに示しています。

表 *9.* 定義済み（非自動インストール）端末の*SEND RUSIZE*

SEND RUSIZE (VTAM)	**TYPETERM SENDSIZE**	バインドに使用される **RUSIZE**
0	0	0
0	指定	TYPETERM SENDSIZE サイズ
指定	0	0
指定	指定	TYPETERM SENDSIZE サイズ
注: このテーブルに対する例外は LOGMODE(0) です。 TYPETERM 定義でこれを指定した場合、他に指定されている値に関係なく、VTAM 値が使用されます。 **注:** VTAM は現在 z/OS Communications Server になっています。		

APPC 端末: APPC (LUTYPE6.2) 単一セッション端末の場合、256 が適切な値です。

自動インストール端末: 自動インストール端末の場合、ゼロ以外の SENDSIZE 値は、この TYPETERM で定義された論理装置のセッションをバインドするために使用される SEND RUSIZE の最大値あるいは実際の値を指定します。

SENDSIZE の効果は、SEND RUSIZE が z/OS Communications Server LOGMODE テーブルに存在しているかどうかによって異なります。 表 10 は、セッションをバインドするために使用される SEND RUSIZE について、TYPETERM と LOGMODE の値の考えられる組み合わせごとに示しています。

表 *10.* 自動インストール端末の*SEND RUSIZE*

SEND RUSIZE (VTAM)	**TYPETERM SENDSIZE**	バインドに使用される **RUSIZE**
0	0	0
0	指定	TYPETERM SENDSIZE サイズ
指定	0	VTAM SEND RUSIZE サイズ
TYPETERM SENDSIZE 以下の値を指定	指定	VTAM SEND RUSIZE サイズ
TYPETERM SENDSIZE より大きい値を指定	指定	この組み合わせは無効であるため、メッセージ DFHZC5963 が発行されます

SESSIONTYPE(*type*)

z/OS Communications Server SNA 論理装置に使用可能なセッションのタイプを指定します。詳しくは、 265 ページの『TYPETERM属性のデフォルト値』を参照してください。

SHIPPABLE({**NO**|**YES**})

この装置がリモート・トランザクションの開始を試行する場合に、定義をリモート・システムに送信できるかどうかを指定します。

NO この定義はリモート・システムにシップされません。

YES この定義はリモート・システムにシップできます。

この機能は、自動インストール端末でも独自のTERMINAL定義を持つ 端末でも、どの端末でも使用されます。端末がローカル・システムに これらのメソッドによってインストールされた定義を保持 していない場合、シップは機能しません。

SHIPPABLE(YES)を使用する場合、リモート・システムで トランザクションを開始するために、ローカルで定義された 端末の端末定義がリモート・システムに存在することを 確認する必要はありません。これは、リモート・システムがローカル・システムとCSDファイルを共用できない場合に 便利です。

端末の定義がリモート・システムにすでにインストール されている（またはすでにシップされている）必要があります。

SHIPPABLE(YES) を使用するかどうかの決定については、 215 ページの『トランザクション・ルーティング用の端末』 を参照してください。

SIGNOFF({**YES**|**NO**|**LOGOFF**})

GNTRAN (CESF またはユーザー定義トランザクション) が接続され、端末のサインオフを試行したときに実行されるアクションを指定します。 RACF を使用する場合は、RACF セグメントで TIMEOUT 制限を指定します。

YES オペレーターからの最後の入力の後、指定された時間を経過すると、端末が自動的に CICS からサインオフします。

NO 端末はタイムアウトしません。

LOGOFF
オペレーターからの最後の入力の後、指定された時間を経過すると、端末が自動的に CICS からサインオフし、その後に z/OS Communications Server からログオフします。自動インストール・、モデルの場合LOGOFFが便利です。タイムアウトした端末の エントリーでは仮想ストレージが浪費されません。

端末バッファー内に未処理データがあるために GNTRANが接続に失敗した場合（BIDエラー）、 端末はサインオフおよびログオフされます。GNTRANは 実行されないため影響はありません。

注**:** DEVICE(APPC)が指定されている場合、この属性の値を 変更することはできません。その場合のデフォルト値は SIGNOFF(NO)です。

SOSI({**NO**|**YES**})
装置が混合 EBCDIC および 2 バイト文字セット (DBCS) フィールドをサポートするかどうかを指定します。

NO 装置は、EBCDICフィールドと2バイト文字セット(DBCS) フィールドの混在をサポートしません。

YES 装置は、EBCDICフィールドと2バイト文字セット(DBCS) フィールドの混在をサポートします。

QUERY構造化フィールドを使用して、EBCDICフィールドと DBCSフィールドを混在使用するように装置がセットアップ されているかどうかを判断できます。

TERMMODEL({**1**|**2**})
端末の型式番号を指定します。装置が 3270 情報表示システムのコンポーネントである場合、端末の型式番号を指定します。

1 デフォルト画面またはバッファー・サイズが12x40 (480バイト／文字)の 3270 MODEL 1のディスプレイおよびプリンター (3277 MODEL 1など)の場合、1を指定します。TERMMODEL(1)は、 3270 MODEL 1のプリンターおよびディスプレイのデフォルトです。

3275 ディスプレイ装置モデル 11 では、1 を指定します。 取得される CICS サポートは、TERMMODEL(1) を 3275 ディスプレイ装置モデル 1 にコーディングして取得されるものと同じです。

2 デフォルト画面またはバッファー・サイズが24x80 (1920バイト／文字)の3270のディスプレイおよびプリンター (3278 MODEL 4など)の場合、2を指定します。TERMMODEL(2) は、3270 互換モードでの 3286 プリンターのデフォルトです。

3275ディスプレイ装置モデル12では、2を指定します。 取得される CICS サポートは、TERMMODEL(2) を 3275 ディスプレイ装置モデル 2 にコーディングして取得されるものと同じです。

TEXTKYBD({**NO**|**YES**})
3270 装置にテキスト・キーボード機能があるかどうかを指定します。

NO 3270装置にテキスト・キーボード機能がありません。

YES 3270装置にテキスト・キーボード機能があります。

TEXTPRINT({**NO**|**YES**})

3288 印刷装置にテキスト印刷機能があるかどうかを指定します。

NO 3288プリンターにテキスト印刷機能がありません。

YES 3288プリンターにテキスト印刷機能があります。

TTI({**YES**|**NO**})

端末でユーザーがトランザクションを開始できるかどうかを指定します。

YES ユーザーは、端末でトランザクションを開始できます。 ATI(YES)も指定した場合、トランザクションを自動的に開始する こともできます。この場合、一時データ管理またはインターバル 制御機能による自動トランザクション開始時に、適切な端末管理 テーブル端末入力内の条件が設定されます。ATIとTTIを両方 ともYESに指定し、端末にトランザクションがない場合、端末管理 機能がユーザー定義タスクを開始します。このタスクが端末に メッセージを送信します。

照会や注文入力などのトランザクションの処理で使用される端末の 場合、TTI(YES)とATI(NO)を使用します。これは、ディスプレイ または ハードコピー端末に端末要求付きのメッセージのみが送信され、 それによってトランザクションが入力される場合にも適用されます。 3790照会論理装置ではこの指定のみが許可されます。

NO ユーザーは、端末でトランザクションを開始できません。 NOを指定した場合、ATI(YES)を指定すると、トランザクションを 自動的に開始できます。このタイプの端末の一例は、ウェアハウス などのリモート・ロケーションにあって無人だが、メッセージを受信 する可能性がある端末です。

TYPETERM(*name*)

TERMINAL 定義のこの拡張の名前を指定します。名前の最大長は 8 文字です。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

この名前は、このTYPETERMを使用するすべてのTERMINAL定義で 参照されます。このTYPETERM定義は、TYPETERM定義を参照する TERMINAL定義のインストールの前またはインストールと同時に インストールする必要があります。

UCTRAN({**NO**|**YES**|**TRANID**})

端末からの入力データ・ストリームを大文字に変換するかどうかを指定します。入力データ・ストリームには、 トランザクションIDやプログラム・データが含まれることがあります。 CICS は、大/小文字混合のトランザクション ID 定義

をサポートします。UCTRAN 属性を使用して、正しいトランザクションがあることを確認することができます。大文字変換は、3270と3270以外の両方の データ・ストリームで実行されます。

NO 大文字変換は実行されません。

YES トランザクションID（存在する場合）とプログラム・データの 両方を含む端末からのすべてのデータ入力を、大文字に変換してから 処理します。

TRANID

入力データ・ストリームにトランザクション ID が含まれる場合、CICSはその定義の検索をする前に、それを大文字に変換します。ただし、トランザクションIDとプログラム・データの両方を 含むすべての入力データが、変換なしでプログラムに渡されます。

したがって、YESとTRANIDのどちらのオプションでも、 トランザクションIDを大文字で定義し、端末から大文字または 小文字で入力できます。ただし、TRANIDオプションの場合、 トランザクションIDとプログラム・データが、変換なしで プログラムに渡されます。

PROFILE リソースで、トランザクション・レベルでの大文字への変換を要求することもできますが (138 ページの『PROFILE 属性』 を参照)、TYPETERM UCTRAN(YES) 定義は PROFILE UCTRAN(NO) 定義をオーバーライドするので注意してください。そのため、TYPETERM UCTRAN(YES) を指定すると、PROFILE UCTRAN(NO) が無効になります。最初の要求を除くすべてのRECEIVE要求について、 ASISオプションを使用することにより、アプリケーション・ プログラムが変換をオーバーライドすることがあります。

表 11は、 PROFILE および TYPETERM リソース定義の UCTRAN 設定に従って、端末入力のどの部分が変換されるかを示しています (トランザクション ID および/またはデータ)。

表 *11. tranid* およびデータ変換での *UCTRAN* 属性の効果

PROFILE の **UCTRAN**	**TYPETERM** の **UCTRAN**	**TRANID** は変換されたか**?**	データは変換されたか**?**
YES	YES	はい	はい
YES	NO	いいえ	はい
YES	TRANID	はい	はい
NO	YES	はい	はい
NO	NO	いいえ	いいえ
NO	TRANID	はい	いいえ

一部の各国語文字は、UCTRAN(YES) または UCTRAN(TRANID) が指定されていても自動的に変換されません。そのような場合は、大文字変換 に示されているいずれかの方法を使用することができます。

USERAREALEN({**0**|*number*})

この端末のユーザー域の長さをバイト数 (0 から 255) で指定します。できるだけ小さい値を指定してください。システムの初期設定時に、端末ユーザー域はゼロに初期化されます。

端末ユーザー域は、**TCTUALOC** システム初期設定パラメーターの値に応じて、31 ビット・ストレージあるいは 24 ビット・ストレージに配置されている場合があります。デフォルトでは、端末ユーザー域は、31 ビット・ストレージあるいは 24 ビット・ストレージのいずれに配置することも可能ですが、CICS は、31 ビット・ストレージが使用できる場合は、常にそちらを使用します。 31 ビット・アドレッシングに対応できないアプリケーション・プログラムがあるために、端末ユーザー域を 24 ビット・ストレージに保管する必要がある場合は、CICS 領域にシステム初期設定パラメーター **TCTUALOC=BELOW** を指定してください。

VALIDATION({**NO**|**YES**})

8775 の場合、8775 装置に拡張妥当性検査機能があるかどうかを指定します。この機能は、フィールドを TRIGGER、MANDATORY FILL、または MANDATORY ENTER として定義することを可能にします。

3290 の場合、3290 装置に妥当性検査機能があるかどうかを指定します。この機能は、フィールドを MANDATORY FILL または MANDATORY ENTER として定義することを可能にします。

このオプションは、SCS プリンターでは無効です。 SCSプリンターで VALIDATION(YES)を指定した場合、エラー・メッセージが発生し、 オプションは無視されます。

QUERY構造化フィールドを使用して、検証機構を使用するように 装置がセットアップされているかどうかを判別できます。

VERTICALFORM({**NO**|**YES**})

装置に垂直フォーム機能があるかどうかを指定します。 この機能を使用できる装置は、バッチ、バッチ・データ交換、 対話式、SCSPRT、またはLUTYPE4の論理装置です。

NO 装置に垂直フォーム機能がありません。

YES 装置に垂直フォーム機能があります。

XRFSIGNOFF

この属性は廃止されましたが、以前のリリースの CICS との互換性を保つためにサポートされています。

URIMAP リソース

URIMAP 定義は、HTTP 要求、Atom フィード要求、または Web サービス要求の URI に対応するリソースであり、要求の処理に関する情報を提供します。

HTTP サーバーとしての CICS への Web クライアントからの要求

HTTP サーバーとしての CICS に対する要求に適用される URIMAP 定義の USAGE 属性は、SERVER です。これらの URIMAP 定義は、CICS が Web クライアントから受け取ることを予期している HTTP 要求の URL に対応しており、各要求への応答を CICS がどのように提供するかを定義しています。URIMAP 定義を使用して CICS に以下のことを指示できます。

- 文書テンプレートまたは z/OS UNIX システム・サービス・ファイルを使用して、HTTP 要求に対する静的応答を提供する。

- アプリケーション・プログラムを使用して、HTTP 要求に対するアプリケーション生成応答を提供する。

Web クライアント要求は、直接接続ユーザー・トランザクションで処理して Web 接続タスクをバイパスする対象になる場合があります。詳しくは、直接接続されたユーザー・トランザクションにより HTTP 要求が処理されるを参照してください。

CICS を HTTP サーバーとして構成する方法については、Enabling CICS web support for CICS as a HTTP serverを参照してください。 HTTP サーバーで必要とされる CICS リソースについては、Defining resources for CICS as a HTTP serverを参照してください。

CEMT および CEDA を使用した URIMAP リソース定義のインストールについては、URIMAP リソース定義のインストールを参照してください。BAS の作業については、BAS URIMAP リソース定義の作業を参照してください。

HTTP クライアントとしての CICS からサーバーへの要求

HTTP クライアントとしての CICS からの要求に適用される URIMAP 定義の USAGE 属性は、CLIENT です。これらの URIMAP 定義では、Web クライアントとして動作するユーザー・アプリケーションが CICS Web サポートを介して HTTP サーバーに要求を行うときに使用される URL が指定されます。この目的で URIMAP 定義をセットアップすることは、アプリケーション・プログラムで URL を特定しないようにすることができることを意味します。開かれた接続を同じアプリケーションまたは別のアプリケーションが再使用できるようにプールするようにすることもできます。詳しくは、CICS の URIMAP リソースを HTTP クライアントとして作成を参照してください。

Web サービス・プロバイダーとしての CICS への SOAP クライアントからの要求

インバウンド Web サービス要求に適用される URIMAP 定義の USAGE 属性は、PIPELINE です。これらの URIMAP 定義では、インバウンド Web サービス要求 (つまり、クライアントが CICS 内の Web サービスを開始するために使用する要求) の URI が、実行する処理を指定した PIPELINE および WEBSERVICE リソースに関連付けられます。

CICS Web サービス・アシスタントを使用してデプロイされるサービス・プロバイダーについては、ピックアップ・ディレクトリーをスキャンする際に URIMAP リソースが自動的に作成されます。このスキャンは、PIPELINE リソースがインストールされたとき、あるいは **PERFORM PIPELINE SCAN** コマンドの結果として行われます。WEBSERVICE リソースを特定の URI に関連付けるための情報を CICS に提供する URIMAP リソースは、必須リソースです。このリソースの属性は、ピックアップ・ディレクトリー内の Web サービス・バインディング・ファイルによって指定されます。 WSDL アーカイブ・ファイルまたは WSDL 文書を特定の URI に関連付けるための情報を CICS に提供する URIMAP リソースは、オプション・リソースであり、ピックアップ・ディレクトリーに WSDL ファイルまたは WSDL アーカイブ・ファイルが存在する場合に作成されます。

Web サービスの概要には、CICS 内の Web サービスに関する詳細情報があります。

SOAP クライアント要求は、直接接続ユーザー・トランザクションで処理して Web 接続タスクをバイパスする対象になる場合があります。詳しくは、直接接続されたユーザー・トランザクションにより HTTP 要求が処理されるを参照してください。

Web サービス要求元としての CICS から Web サービス・プロバイダーへの要求

INVOKE WEBSERVICE コマンドを使用するアウトバウンド Web サービス要求に適用される URIMAP 定義はオプションであり、URIMAP 定義の USAGE 属性は CLIENT です。例えば、URIMAP 定義を使用して、Transport Layer Security (TLS) を使用するソケット接続を確立するときに使用する暗号スイートまたは証明書ラベルを指定できます。新しい接続をそのたびに開くのではなく、開かれた接続をプールして、アプリケーションがさらに **INVOKE WEBSERVICE** コマンドで再使用できるようにすることもできます。URIMAP 定義を使用することにより、管理者はリモート Web サービスへの接続を制御できます。つまり、必要に応じてその URI を変更したり、URIMAP を使用可能/使用不可にして接続を許可/不許可にしたりできます。

サービス・リクエスターに関しては、PIPELINE リソースがインストールされたとき、あるいは **PERFORM PIPELINE SCAN** コマンドの結果として、CICS が URIMAP リソースを自動的に作成することはありません。

Web クライアントから Java Web アプリケーションへの要求

Java Web アプリケーションに適用される URIMAP 定義の USAGE 属性は JVMSERVER です。これらの URIMAP 定義は、JVM サーバーで実行中のサーブレットまたは JSP に対する Web クライアントからのインバウンド要求にマッチングします。 Web アプリケーションの URIMAP マッチングは、初期 HTTP 要求時に発生し、転送される要求では発生しません。URIMAP は、CICS トランザクションを指定して、JVM サーバーでオプションでユーザー ID を指定し要求を実行できます。

詳しくは、CICS バンドル内の Java EE アプリケーションの Liberty JVM サーバーへのデプロイを参照してください。

Atom フィード要求

Atom フィード要求に適用される URIMAP 定義の USAGE 属性は、ATOM です。これらの URIMAP 定義は、Atom 文書に対する Web クライアントからのインバウンド要求に対応します。返す Atom 文書を指定した ATOMSERVICE リソース定義が参照されます。

詳しくは、Atom フィード用の CICS 定義のセットアップを参照してください。

Atom フィード要求は、直接接続ユーザー・トランザクションで処理して Web 接続タスクをバイパスする対象になる場合があります。詳しくは、直接接続されたユーザー・トランザクションにより HTTP 要求が処理されるを参照してください。

HTTP EP アダプターからのビジネス・イベント出力

CICS イベント処理に適用される URIMAP 定義の USAGE 属性は、CLIENT です。これらの URIMAP 定義では、HTTP EP アダプターが HTTP POST を使用して HTTP 1.1 準拠サーバーにイベントを出力するために使用する URL が指定されます。開かれた接続を、さらにイベント出力で再使用できるように、プールするようにすることもできます。 イベント処理の概要に、CICS イベント処理に関する詳細情報があります。

HTTP サーバーとしての CICS の場合、TCPIPSERVICE 定義に関連付けられたアナライザー・プログラムが以前提供していた機能のほとんどが、URIMAP 定義に取り込まれています。必要であれば、現在でも処理パスでアナライザー・プログラムを使用できます。

URIMAP の検索順序

複数の URIMAP が使用可能な場合、同じ URI と TCPIPSERVICE を持つ URIMAP が最初に検索されます。ワイルドカードを含んだ URIMAP については、最初に URI が一致するもの、最後に TCPIPSERVICE が一致するものが検索されます。この検索順序により、固有性が最も高い URIMAP が使用されることになります。

CICS バンドルの URIMAP リソース

CICS バンドルを使用して、URIMAP リソース定義の作成、編集、およびインストールを実行できます。この方法で管理 URIMAP リソースを作成する場合、そのリソースのライフサイクルの管理には CICS バンドルを使用する必要があります。リソースを独立して管理することはできません。 **EXEC CICS INQUIRE URIMAP** コマンドまたは **CEMT INQUIRE URIMAP** コマンドを使用することにより、CICS バンドルによって動的に生成される URIMAP リソースを照会できます。ただし、CICS バンドルによって動的に生成される URIMAP リソースに対して、SET コマンドや DISCARD コマンドを発行することはできません。それらのコマンドは BUNDLE リソースに対して発行する必要があります。そうすると、CICS がそれらを URIMAP リソースに適用します。 CICS バンドル内のリソースの定義について詳しくは、Defining CICS bundlesを参照してください。

アプリケーション・エントリー・ポイントとしての URIMAP リソース

以下の使用シナリオは、URIMAP リソースをアプリケーション・エントリー・ポイントとして使用している場合にサポートされます。

Liberty を使用する、**USAGE(JVMSERVER)** を指定した **URIMAP** リソース
: Liberty JVM サーバーで検出された URIMAP リソースがアプリケーション・エントリー・ポイントで定義されている場合、アプリケーション・コンテキスト・データは JVM サーバー要求プロセッサー・タスク、またはトランザクション属性によって指定されたその別名用に設定されます。

WebSphere MQ を使用しない、**USAGE(SERVER)** または **USAGE(PIPELINE)** を指定した **URIMAP** リソース
: CWXN で検出された URIMAP リソースがアプリケーション・エントリー・ポイントで定義されている場合、アプリケーション・コンテキスト・デ

ータは CWXN タスク用に設定されます。次に、CWXN はユーザー・タスクであるため、CWXN によって開始された別名トランザクションにアプリケーション・コンテキスト・データを自動的に伝搬することができます。例えば、USAGE(PIPELINE) 用の CPIH、USAGE(SERVER) 用の CWBA、**TRANSACTION** フィールドによって指定されたトランザクションなどです。

以下の使用シナリオは、URIMAP リソースをアプリケーション・エントリー・ポイントとして使用している場合はサポートされません。

WebSphere MQ を使用する、**USAGE(PIPELINE)** を指定した **URIMAP** リソース　このシナリオはサポートされていません。

USAGE(CLIENT) または **USAGE(ATOM)** を指定した **URIMAP** リソース
これらのシナリオはサポートされていません。URIMAP がアプリケーション・エントリー・ポイントとして定義されている場合、URIMAP および USAGE(CLIENT) が指定されたバンドルまたは URIMAP および USAGE(ATOM) が指定されたバンドルの状況は DISABLED になっています。このバンドルがインストールされていてバンドルを有効にしようとした場合、バンドル内の ENTRYPOINT バンドル・パーツの状況は DISABLED であり、その結果、バンドルの状況が DISABLED になります。

INQUIRE URIMAP コマンドおよび CEMT INQUIRE URIMAP コマンドを使用すると、URIMAP リソースがエントリー・ポイントとして定義されているアプリケーションの詳細が返されます。

アプリケーション・エントリー・ポイントについて詳しくは、アプリケーションのエントリー・ポイントを参照してください。

関連タスク:

データ・セットの暗号化

URIMAP: 関連リソース

URIMAP リソースで指定する属性および値は、他のリソースで指定する属性および値と整合したものでなければなりません。

しかし、CICS は、URIMAP 定義のインストール時にすべての関連リソースの整合性を検査しないため、ほとんどの不整合はインストール時に報告されません。例外は、URIMAP リソース定義のインストール に記載された検査のみです。

- HTTP サーバーおよび Web サービスとしての CICS 用の URIMAP 定義では、TCPIPSERVICE 属性は、この URIMAP 定義が関連するインバウンド・ポートを定義する TCPIPSERVICE リソース定義の名前を指定します。この属性はオプションであり、指定されない場合は、すべての TCPIPSERVICE 定義に対するインバウンド要求に URIMAP 定義が適用されます。 TCPIPSERVICE 属性を指定する場合は、以下のようにします。
  - 選択されたTCPIPSERVICE資源定義はPROTOCOL(HTTP)を 指定する必要があります。
  - TCPIPSERVICE リソース定義で SSL(YES) または SSL(CLIENTAUTH) が指定されている場合、URIMAP 定義の SCHEME 属性は HTTPS にする必要があります。スキームとして HTTPS が指定された URIMAP 定義が、Web

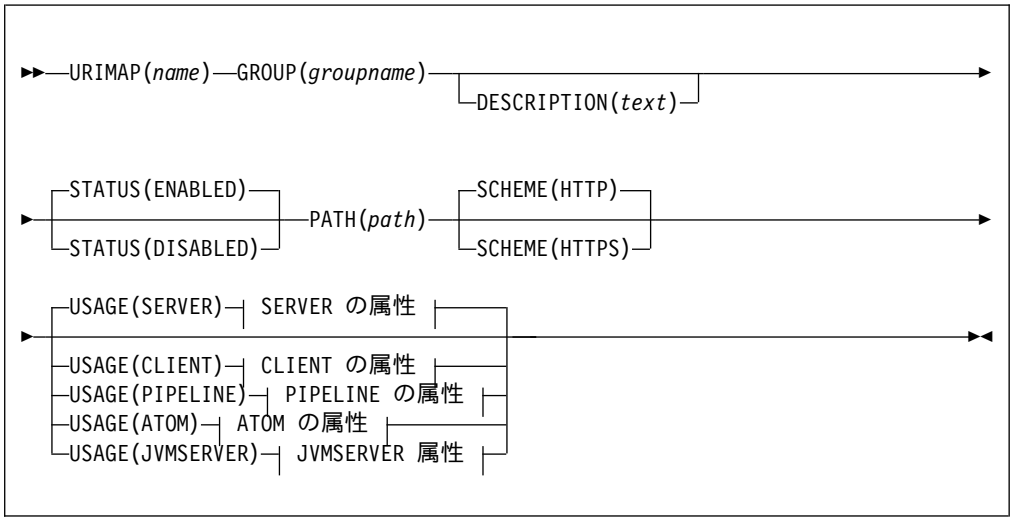
クライアントが行う要求と一致する場合、CICS は、要求で使用されるインバウンド・ポートが SSL を使用しているかを検査します。そのポートで SSL が指定されていない場合、要求は 403 (Forbidden) 状況コードで拒否されます。

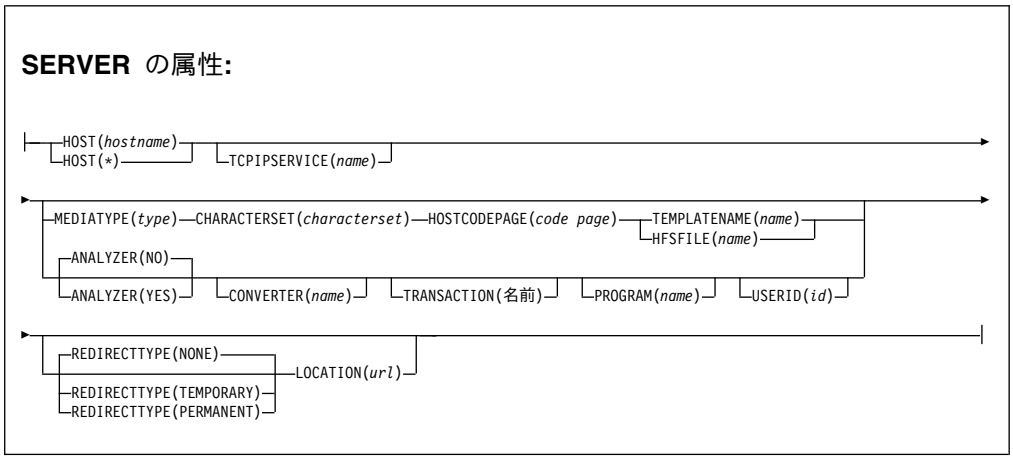
URIMAP 定義を使用するには、選択された TCPIPSERVICE 定義を事前にインストールする必要があります。

- 静的応答が提供される HTTP サーバーとしての CICS 用の URIMAP 定義では、TEMPLATENAME 属性は、静的応答を形成する CICS 文書テンプレートの名前を 1 から 48 文字で指定します。文書テンプレートは DOCTEMPLATE リソース定義を使用して定義する必要があり、その定義の TEMPLATENAME 属性は、URIMAP 定義で使用される名前を指定します。 URIMAP 定義を使用するには、事前に DOCTEMPLATE リソース定義が使用可能になっている必要があります。

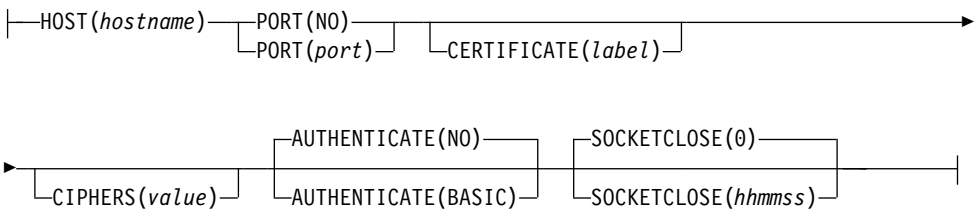
URIMAP の属性

URIMAP リソースの構文と属性について説明します。

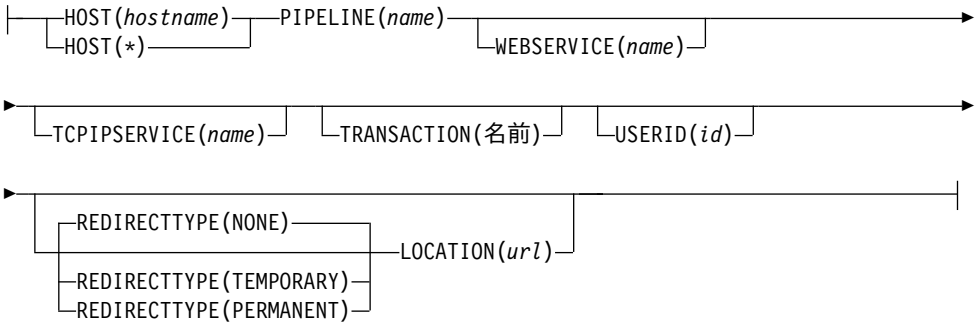




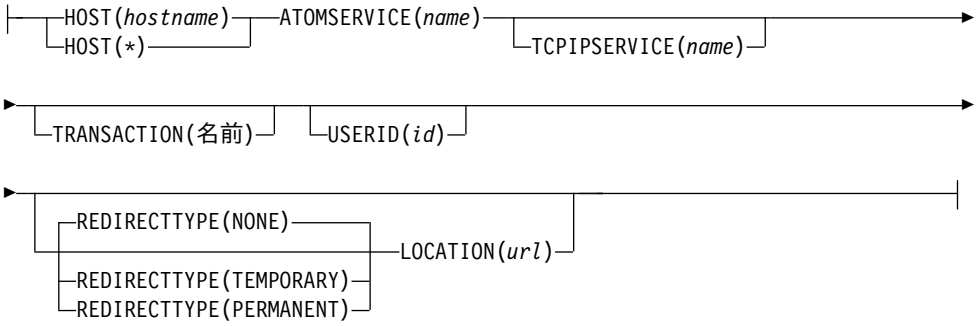
CLIENT の属性:



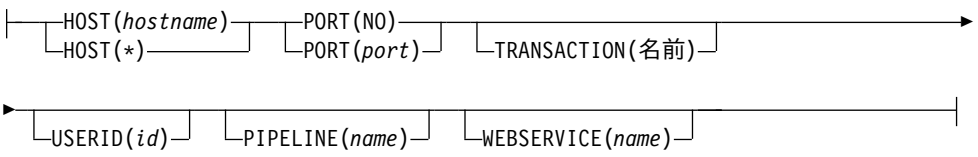
PIPELINE の属性:



ATOM の属性:



JVMSERVER 属性:



ANALYZER({**NO**|**YES**})

この属性は、アプリケーション生成の応答が提供される場合に、USAGE(SERVER) で使用します。そのほかの使用タイプの場合、この属性は強制的に NO に設定されます。

直接接続されたユーザー・トランザクションによって HTTP 要求を処理するには、ANALYZER を NO に設定する必要があります。詳しくは、直接接続されたユーザー・トランザクションにより HTTP 要求が処理されるを参照してください。

ANALYZER は、アナライザー・プログラムを HTTP 要求の処理に使用するかどうかを指定します。アナライザーには、この URIMAP 定義が関連している TCPIPSERVICE 定義 (1 つまたは複数) を関連付ける必要があります。 (アナライザー・プログラムは、ローカルの CICS 領域になければなりません。) YES を指定すると、アナライザーが実行されます。NO は、アナライザーを使用しないことを意味します。

デフォルトのアナライザー DFHWBAAX とサンプル・アナライザー DFHWBADX は、URIMAP に ANALYZER(YES) と指定されていても、要求に一致する URIMAP 定義が見つかった場合であれば、要求を分析しません。

アナライザー・プログラムを使用している場合でも、CONVERTER、TRANSACTION、USERID、および PROGRAM の各属性を指定できます。これらの属性に指定する値は、アナライザー・プログラムへの入力として使用されますが、アナライザー・プログラムによってオーバーライドされる可能性があります。あるいは、こうした属性をブランクのままにして、アナライザー・プログラムが指定するようにできます。

ATOMSERVICE(*name*)

この属性は、USAGE(ATOM) で使用します。

クライアントが、この URIMAP 定義で指定された URI を使用して Atom フィードに関する要求を CICS に対して行うと、ATOMSERVICE は、Atom フィードの ATOMSERVICE リソース定義の名前 (1 から 8 文字) を指定します。 ATOMSERVICE リソース定義は、Atom サービス、フィード、コレクション、またはカテゴリー文書を定義し、Atom 構成ファイル、CICS リソースまたはアプリケーション・プログラム、およびフィード用のデータを提供するために使用される XML バインディングを識別します。

許容文字:

A-Z 0-9 $ @ #

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

AUTHENTICATE({**NO**|**BASIC**})

この属性は USAGE(CLIENT) 用です。

AUTHENTICATE は、HTTP 基本認証情報を HTTP サーバーに送信するかどうかを指定します。AUTHENTICATE(BASIC) には、XWBAUTH グローバル・ユーザー出口によって提供されるユーザー ID とパスワード、または **WEB SEND** や **WEB CONVERSE** のような API コマンドの値としてのユーザー ID とパスワードが必要です。 USERNAME および PASSWORD を API コマンドに指定している場合、XWBAUTH グローバル・ユーザー出口は呼び出されません。 API コマンドで認証値を指定すると、URIMAP リソースで指定された AUTHENTICATE 値の代わりに、この値が使用されます。

CERTIFICATE(*label*)

この属性は、USAGE(CLIENT) で使用します。

CERTIFICATE は、SSL ハンドシェーク時に、SSL クライアント証明書として使用する X.509 証明書のラベルを指定します。証明書ラベルの長さは最大で 32 文字です。この属性は、CICS がクライアントとして作成した HTTPS 要求に、URIMAP 定義で指定された URI を使用する場合にのみ使用します。 SSL クライアント証明書を要求するのはサーバーの責任であり、その場合 CICS が URIMAP 定義で指定される証明書ラベルを提供します。この属性を省略した場合は、CICS 領域ユーザー ID の鍵リングに定義されているデフォルトの証明書が使用されます。証明書は、外部セキュリティー・マネージャーのデータベースの鍵リングに保管する必要があります。 指定された証明書、あるいはデフォルトの証明書に、使用可能な秘密鍵がなければなりません。ない場合は、URIMAP をインストールすることができません。証明書を使用しない場合は、このフィールドをブランクのままにしておき、CICS 領域で使用される鍵リングにデフォルトの証明書がないことを確認してください。詳しくは、鍵リングの手動構築を参照してください。

CHARACTERSET(*characterset*)

この属性は、静的応答が提供される場合に、USAGE(SERVER) で使用します。

CHARACTERSET では、CICS が Web クライアントに送信する応答のエンティティー本体の変換後の文字セットの名前 (1 から 40 文字) を指定します。CICS は、IANA によって指定されるすべての文字セットをサポートするわけではありません。 HTML コード化文字セットに、CICS がサポートしている IANA 文字セットの一覧が含まれています。この属性の値は、応答の Content-Type ヘッダーに含まれています。

静的応答を提供する場合に、MEDIATYPE 属性でテキスト・タイプを指定しているときには、CHARACTERSET を指定する必要があります。

CIPHERS(*value*)

この属性は、USAGE(CLIENT) で使用します。

CIPHERS 属性は、以下の 2 とおりの方法のいずれかで指定できます。

- 最大 28 個の 2 桁の暗号スイート・コードを示すリストとして解釈される 56 桁までの 16 進数字で構成されるストリング。
- SSL 暗号スイート仕様ファイルの名前。この z/OS UNIX ファイルは、**USSCONFIG** システム初期設定パラメーターによって指定されるディレクトリーの `security/ciphers` サブディレクトリー内にあります。例えば、**USSCONFIG** が `/var/cicsts/dfhconfig` に設定され、**CIPHERS** が

strongciphers.xml に設定されている場合、完全修飾ファイル名は /var/cicsts/dfhconfig/security/ciphers/strongciphers.xml です。詳しくは、SSL 暗号スイート仕様ファイルの作成を参照してください。

CEDA トランザクションを使用してリソースを定義すると、CICS は、デフォルトの許容コード・リストに従ってその属性を自動的に初期化します。この属性を CICS で初期化できるようにするためには、CEDA を実行する CICS 領域で KEYRING システム初期設定パラメーターを指定する必要があります。
KEYRING を設定しなかった場合、CICS はその属性を初期化しません。デフォルトのコード・リストは、35363738392F303132330A1613100D15120F0C です。ただし、システム初期設定パラメーターで **NISTSP800131A=CHECK** が設定されている場合は、35363738392F303132330A1613100D です。
暗号コードは、再配列することも、初期リストから除去することもできます。ただし、指定した暗号化レベルのデフォルト・リストに含まれていない暗号コードを追加することはできません。値をコードのデフォルト・リストにリセットするには、すべての暗号スイート・コードを削除します。それにより、フィールドには自動的にデフォルト・リストが取り込まれます。
詳しくは、暗号スイートおよび暗号スイート仕様ファイルを参照してください。

CONVERTER(*name*)
この属性は、アプリケーション生成の応答が提供される場合に、USAGE(SERVER) で使用します。

CONVERTER には、要求と応答を変換するコンバーター・プログラムの名前 (1 から 8 文字) を指定します。通常、コンバーター・プログラムは、HTTP 要求をアプリケーション・プログラムによって使用される COMMAREA に変換し、出力を HTTP 応答に変換します。ローカル CICS 領域で使用可能な任意のコンバーター・プログラムを使用することができます。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

アナライザー・プログラムと TCPIPSERVICE 定義の間の関係とは違って、コンバーター・プログラムと TCPIPSERVICE 定義の間には関連はありません。

ANALYZER 属性に YES が指定されている場合、CONVERTER 属性はアナライザー・プログラムへの入力として使用されます。ただし、アナライザー・プログラムで、この属性を指定変更できます。コンバーター・プログラムを使用している場合でも、URIMAP 定義の PROGRAM 属性を指定できます。ただし、この属性に指定した値は、コンバーター・プログラムで指定変更できます。また、この属性をブランクに設定して、コンバーター・プログラムに指定させることもできます。

DESCRIPTION(*text*)
このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

GROUP(*groupname*)

各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字: A-Z 0-9 $ @ # 小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

HFSFILE(*name*)

この属性は、静的応答が提供される場合に、USAGE(SERVER) で使用します。

HFSFILE は、z/OS UNIX System Services zFS ファイルの完全修飾 (絶対) 名または相対名を指定します。このファイルは、Web クライアントからの HTTP 要求に対して送信される静的応答の本体を形成します。最大 255 文字を使用できます。

この名前は絶対パスとして指定できます。絶対パスには、すべてのディレクトリーが含まれ、スラッシュで始まります (例: `/u/facts/images/bluefish.jpg`)。別の方法として、相対パスを指定することもできます (`facts/images/bluefish.jpg` など)。

- オンライン・リソース定義によって定義された URIMAP リソースの静的応答ファイルの場合、ファイル・パスは、完全修飾パス (先行スラッシュ / が接頭部として付いている場合)、または CICS 領域ユーザー ID の HOME ディレクトリーからの相対パスです。
- CICS バンドルで定義された URIMAP リソースの静的応答ファイルの場合、ファイル・パスは、CICS バンドルのルート・ディレクトリーからの相対パスです。zFS ファイルが、URIMAP リソースと一緒に CICS バンドルにパッケージ化されている必要があります。 CICS にファイル・タイプおよびファイルのエンコードに使用されたコード・ページを通知するには、MEDIATYPE 属性および HOSTCODEPAGE 属性を指定する必要があります。ファイルが CICS バンドルで定義されており、CICS Explorer を使用して zFS にエクスポートされる場合、ファイルのエンコードに使用されるコード・ページは、CICS Explorer が使用するコード・ページと同じです。詳しくは、Referencing zFS artifacts in a bundleを参照してください。

指定される値は UNIX ファイルに有効な名前でなければなりません。 • 組み込みスペース文字を含めることはできません。 • / 文字の連続インスタンスが含まれていてはなりません。 • 大/小文字の区別があります。 許容文字: A-Z a-z 0-9 . / _ # @ -

TEMPLATENAME または HFSFILE を指定する場合は、ANALYZER を NO に設定する必要があります。アプリケーション生成の応答にのみ関連する他の属性 (TRANSACTION、CONVERTER、および PROGRAM) はブランクのままにする必要があります。

パス・マッチングを使用する場合、zFS ファイルのパスの最後と、PATH 属性によって指定されるパスの最後にも、ワイルドカード文字としてアスタリスクを含めてください。 CICS は、ワイルドカード文字でカバーされる各 HTTP 要求パスの一部を取得して、それをファイル・パスの最後の部分として置換します。

例えば、URIMAP 定義を作成する場合、以下のように指定された属性を使用するとします。PATH 属性:

```
findout/pictures/*
```

さらに、以下のように指定された HFSFILE 属性を使用するとします。

```
/u/facts/images/*
```

この URIMAP 定義を使用して、着信 HTTP 要求を処理します。

```
http://www.example.com/findout/pictures/bluefish.jpg
```

CICS はアスタリスクの部分に代えて `bluefish.jpg` を、URIMAP 定義で指定された zFS ファイル・パスに追加するので、zFS ファイル

```
/u/facts/images/bluefish.jpg
```

が静的応答として使用されます。

HFSFILE 仕様にアスタリスクを単独で使用することはできません。ディレクトリー構造のレベルを少なくとも 1 つ指定する必要があります。

Unicode 文字を含む IRI (Internationalized Resource Identifier) を使用する場合、zFS ファイル名およびそれに対応するパスでは、Unicode 文字をパーセント・エンコードされた表記にエスケープする必要があります。支援ツールを Web から入手することができます。 Web で「Unicode パーセント・エスケープ (Unicode percent escaped)」を検索してください。

照会ストリングを zFS ファイルの内容に置き換えることはできません。ただし、zFS ファイルを CICS 文書テンプレートとして定義した上で、HFSFILE 属性ではなく TEMPLATENAME 属性を使用してそれを指定することは可能です。

HOST(*hostname*|*****)

この属性は、すべての USAGE オプションで使用します。

HOST は、URIMAP 定義が適用される URI のホスト名、またはその IPv4 アドレスか IPv6 アドレスを指定します。この名前の長さは最大 116 文字です。URL の構成要素では、各構成要素について、およびそれらがどのように区切られているかについて説明しています。

HOST 属性は常に必要です。HOST 属性には、英数字、ハイフン **-**、コロン **:**、またはピリオド **.** のみを使用することができます。ただし、IP アドレスの代わりに文字でホスト名を指定する場合は、コロンを使用することができません。 CICS は定義時にホスト名を妥当性検査します。ホスト名は大/小文字のどちらでも入力できますが、IP アドレスではなく文字ホスト名を指定する場合、ホスト名は URIMAP 定義では小文字に変換されます。

USAGE(SERVER)、USAGE(PIPELINE)、USAGE(ATOM)、または USAGE(JVMSERVER) を指定するときは、HOST 属性に単一のアスタリスクを使用できます。その場合、URIMAP 定義が任意のホスト名と一致します。HOST 属性では、アスタリスクを他の文字と一緒に指定してワイルドカードとして使用することはできません。これらの USAGE タイプの場合、この属性にポート番号を設定しないでください。

URIMAP リソースは、Internationalized Resource Identifier (IRI) をサポートします。IRI には Unicode 文字を含めることができます。 Unicode 文字を含むホスト名を指定する場合、RFC 3492 で記述されているように、そのホスト名を Punycode フォーマットに変換しなければなりません。この変換を行うためのツールを CICS は提供していませんが、Unicode から Punycode への変換をサポートする無料アプリケーションをインターネットで入手できます。ホスト名としてアスタリスクを使用する場合は、Punycode を使用する必要はありません。IRI について詳しくは、Internationalized Resource Identifier (IRI)を参照してください。

IPv4 アドレスと IPv6 アドレスは、複数のフォーマットで指定することができます。アドレス形式について詳しくは、IP アドレスを参照してください。

CICS が HTTP クライアントであり、USAGE(CLIENT) を指定する場合には、次のようにして、サーバーへの要求の中でポート番号を指定できます。

- PORT 属性を使用して、ポート番号を指定します。HOST 属性の使用に PORT が置き換えられ、ポート番号が指定されます。
- ネイティブ IPv4 アドレスとホスト名を使用する既存プログラムで互換性を維持するために、ポート番号を指定する際には HOST 属性を使用することができます。ポート番号を指定する際のフォーマットには、ネイティブの IPv4 アドレスとホスト名のみを使用できます。このとき、`1.2.3.4:80` や `hostname.com:443` のように、ポート番号の前にコロンを置きます。
- IPv6 アドレス (または IPv6 アドレスに解決されるホスト名) を指定する場合は、ユーザーが二重モード (IPv4 および IPv6) 環境で操作しており、通信先のクライアントおよびサーバーも二重モード (IPv4 および IPv6) 環境で稼働していることを確認してください。 IPv6 についての詳細は、IPv6 と CICS についてを参照してください。
- ネイティブの IPv6 アドレスでは、PORT 属性を使用してポート番号を指定する必要があります。IPv6 アドレスでは、ポート番号とアドレスの間に大括弧を置く必要があります。また、大括弧は、すべての EBCDIC 文字セットにおいて固定値ではありません。このため、HOST 属性では、大括弧はサポートされていません。
- ポート番号は、スキームのデフォルトと異なる場合にのみ指定します。SSL 非対応の HTTP、HTTPS、および SSL 対応の HTTP のデフォルト・ポート番号は、それぞれ 80、443、443 です。
- HOST 属性にポート番号を指定し、PORT 属性に異なるポート番号を指定すると、エラーが返されます。 HOST 属性と PORT 属性の両方でポート番号を指定していない場合、スキームのデフォルト・ポート番号が使用されます。

HOSTCODEPAGE(*code page*)

この属性は、静的応答が提供される場合に、USAGE(SERVER) で使用します。

これは、静的応答を形成するテキスト文書をエンコードする IBM コード・ページ (EBCDIC) の名前を 1 ～ 10 文字で指定します。CICS は、静的応答のエンティティー本体のコード・ページを変換するために、この情報を必要とします。

ホスト・コード・ページ名の標準の CICS 形式は、3 から 5 桁の 10 進数を使用して作成されたコード・ページ番号 (または、一般的には CCSID) と、必要に応じて末尾に埋め込まれたスペースで構成されます。コード・ページ 37 の場合、3 桁未満のため、標準形式では 037 になります。 CICS は、標準の形式に従っていない場合でも、末尾にスペースが埋め込まれた 1 から 65535 の範囲の任意の 10 進数をコード・ページ名として受け入れます。

静的応答が提供されており、MEDIATYPE 属性によってテキスト・タイプが指定される場合、HOSTCODEPAGE を指定する必要があります。

LOCATION(*url*)

この属性は、USAGE(SERVER)、USAGE(PIPELINE)、および USAGE(ATOM) で使用します。

LOCATION は、クライアントの要求のリダイレクト先である URL を最大 255 文字で指定します。この URL は、スキーム、ホスト、パス構成要素、および適切な区切り文字を含む完全な URL である必要があります。CICS は URL が有効であるかどうかを検査しません。このため、リダイレクト先が存在すること、および URL が正しく指定されていることを確認する必要があります。

PATH 属性の説明には、URL に含めることができない文字をリストしています。これらの文字は、LOCATION 属性で使用してはなりません。例外は # 文字です。これは、LOCATION 属性で、URL に続くフラグメント ID の前に、分離文字として使用することができます。

REDIRECTTYPE 属性は、リダイレクトのタイプを指定するために使用されます。一時または永続のリダイレクトが指定される場合、LOCATION 属性内の URL がリダイレクトに使用されます。リダイレクトが指定されない場合、LOCATION 属性内の URL は無視されます。SET URIMAP コマンドを使用すると、REDIRECTTYPE 属性と LOCATION 属性を変更できます。

MEDIATYPE(*type*)

この属性は、静的応答が提供される場合に、USAGE(SERVER) で使用します。

MEDIATYPE は、CICS が HTTP 要求に提供する静的応答のメディア・タイプ (データ内容) を指定します。例えば、`image/jpg`、`text/html`、`text/xml` などです。最大で 56 文字を入力できます。メディア・タイプには、1 つのスラッシュを含める必要があります。メディア・タイプは大文字と小文字の両方で入力できます。ただし、メディア・タイプは URIMAP 定義で小文字に変換されます。

以前に認識された各タイプのデータ内容の名前は、IANA によって定義されます。リストについては、http://www.iana.org/assignments/media-types/media-types.xhtml を参照してください。CICS は、この属性の値を使用して応答のコンテンツ・タイプ・ヘッダーを作成します。

この属性はデフォルトがないため、指定する必要があります。 `text/` で始まるタイプや `+xml` を含むタイプなどのテキスト・タイプを MEDIATYPE 属性で指定した場合は、コード・ページ変換が実行されるように、CHARACTERSET 属性と HOSTCODEPAGE 属性も指定する必要があります。テキスト・メディ

ア・タイプは RFC 3023 によって識別されていて、この RFC は http://www.ietf.org/rfc/rfc3023.txt から入手できます。

動的な (アプリケーション生成の) 応答では、この属性は使用されません。応答のメディア・タイプは、WEB SEND コマンドによって指定されます。

PATH(*path*)

この属性は、すべての USAGE オプションで使用します。

PATH は、URIMAP 定義を適用する URI のパス・コンポーネントを指定します。これは、パス・コンポーネントの先頭のスラッシュ (/) を含めて、最大 255 文字で指定することができます。スラッシュを含めない場合、パスは 254 文字で指定します。 指定可能な最短のパスは、特定ホスト名の URL 構造のルートを示す 1 つのスラッシュです。パス構成要素の先頭にスラッシュを付加するかどうかは任意です。しかし、スラッシュを省略した場合、CICS によって実行時にスラッシュが追加されます。パスは、`software/htp/cics/index.html` のように指定します。URL の構成要素では、各構成要素について、およびそれらがどのように区切られているかについて説明しています。

PATH 属性は大/小文字混合で指定されます。URIMAP 定義では、その大/小文字が保持されます。PATH 属性に指定できる文字は、URI で使用できる文字のみです。具体的には、文字 < > # % " { } | \ ^ [ ] ` および組み込みブランクを使用することはできません。ただし、有効な 16 進数エスケープ・シーケンスを指定するため (つまり、大/小文字の有効な 2 桁の 16 進数が後に続く場合) の % は例外です。波形記号文字 ~ を CICS で指定することはできません。この文字は、対応する 16 進エスケープ・シーケンス (%7E) で置換する必要があります。 CICS は、定義時に文字の使用の妥当性検査を行います。

URIMAP リソースは、Internationalized Resource Identifier (IRI) をサポートします。IRI には Unicode 文字を含めることができます。 Unicode 文字が含まれるパスを指定する場合は、Unicode 文字のパーセント・エンコード表記にそれらの Unicode 文字をエスケープする必要があります。 Unicode 文字をパーセント・エンコード表記に変換できるアプリケーションをお持ちでない場合は、このタスクを実行できる無料アプリケーションをインターネットで入手できます。この場合も、パスは 255 文字以下でなければなりません。このコンテキストにおける文字は、元の Unicode 文字ではなく単一の ASCII 文字を意味します。例えば、パーセント・エンコード表記のキリル文字 `%D0%B4` は、255 文字制限の中の 6 文字としてカウントされます。

HTTP サーバーとしての CICS を Web サービスと関連付ける URIMAP 定義の場合、URIMAP 定義を複数のパスに対応させるには、パスの末尾にワイルドカード文字としてアスタリスクを使用する必要があります。例えば、パス `/software/htp/cics/*` を指定すると、URIMAP 定義は、アスタリスクの左側のストリングで始まるパスを持つすべての要求とマッチングします。パス `/*` を指定すると、URIMAP 定義は、HOST 属性で指定されたホストに送信されるすべての要求をマッチングします。複数の URIMAP 定義によって、HTTP 要求が突き合わせされると、突き合わせが最も絞られます。

照会構成要素があり、URIMAP 定義をその照会にのみ適用する場合は、それをパス構成要素の一部として組み込むことができます。ストリングの先頭に、疑問符 (?) を含めます。照会ストリングに指定できる文字は、URI で使用できる文字のみです。照会ストリング自体にアスタリスクをワイルドカードとして含めることはできませんが、ワイルドカードとしてアスタリスクが含まれるパスの後に

照会ストリングを続けることは可能です。 URIMAP 定義に照会ストリングを含めない場合、HTTP 要求内の照会ストリングは、マッチングでは自動的に無視されます。

Atom フィード用の URIMAP 定義の場合は、パスの末尾にアスタリスクをワイルドカード文字として使用する必要があります。 URIMAP 定義で指定するパスの部分は、Atom フィードと Atom エントリーの各 URL に共通の部分です。 CICS は URL の残りの部分を、フィードの Atom 構成ファイル内のすべての *<atom:link>* エレメントで指定された URL とマッチングします。

HTTP クライアントとしての CICS の URIMAP 定義では、アスタリスクをワイルドカードとして使用することはできません。要求の完全パスを指定する必要があります。WEB OPEN コマンドで URIMAP 定義を参照する場合、このパスはその接続に関連する **WEB SEND** コマンドのデフォルト・パスになります。URIMAP 定義を **WEB SEND** コマンドで参照した場合、パスはその **WEB SEND** コマンドに対して使用されます。ただし、URIMAP 定義のホスト属性は、接続の **WEB OPEN** コマンドで指定されたホストと一致している必要があります。

PIPELINE(*name*)

この属性は、USAGE(PIPELINE) で使用します。

クライアントが、この URIMAP 定義で指定された URI を使用して CICS に対してインバウンド Web サービス要求を行う場合、PIPELINE は Web サービス用の PIPELINE リソース定義の名前 (1 から 8 文字) を指定します。PIPELINE リソース定義は、クライアントからのサービス要求で動作するメッセージ・ハンドラーについての情報を提供します。PIPELINE リソースでは、これらのリソース定義について説明しています。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

PORT({**NO**|*port*})

この属性は、USAGE(CLIENT) オプションと USAGE(JVMSERVER) オプションにのみ適用されます。

USAGE(CLIENT) の場合、PORT 属性には、CICS アプリケーションがサーバーと通信するときに使用するポートの 10 進数を指定します。この値は、1 から 65535 の範囲の数値でなければなりません。

ポート番号と HOST の値を組み合わせることで、この URIMAP でのアウトバウンド要求の宛先が決定します。ポート番号は、スキームのデフォルトと異なる場合にのみ指定します。SSL を使用しない HTTP の場合は 80、HTTPS および SSL を使用する HTTP の場合は 443 です。

HOST 属性にポート番号を指定し、PORT 属性に異なるポート番号を指定すると、エラーが返されます。 HOST 属性と PORT 属性の両方でポート番号を指定していない場合、スキームのデフォルト・ポート番号が使用されます。

USAGE(JVMSERVER) の場合、PORT 属性には、Liberty プロファイル・サーバーで実行中のアプリケーションにアクセスするための要求の受信に使用するポート番号を指定します。

PORT 属性に値を指定しないと、PORT は NO に設定され、この属性が使用されていないことを示します。

PROGRAM(*name*)
この属性は、アプリケーション生成の応答が提供される場合に、USAGE(SERVER) で使用します。

PROGRAM は、HTTP 応答を構成するユーザー・アプリケーション・プログラムの名前を 1 ～ 8 文字で指定します。HTTP サーバーとしての CICS では、この属性が必要です。ただし、アナライザー・プログラムまたはコンバーター・プログラムが指定されている場合、テンプレート名または zFS ファイルが指定されている場合、またはリダイレクトが指定されている場合を除きます。

許容文字: A-Z 0-9 $ @ # CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

ANALYZER 属性に YES を指定している場合、または CONVERTER 属性にコンバーター・プログラムを指定している場合には、PROGRAM 属性がアナライザー・プログラムまたはコンバーター・プログラムに対する入力として使用されますが、この指定はプログラムの側で指定変更することができます。あるいは、この属性をブランクのままにして、アナライザー・プログラムまたはコンバーター・プログラムに属性を指定させることもできます。

REDIRECTTYPE({**NONE**|**TEMPORARY**|**PERMANENT**})
この属性は、USAGE(SERVER)、USAGE(PIPELINE)、および USAGE(ATOM) で使用します。

REDIRECTTYPE は、この URIMAP 定義に一致する要求のリダイレクトのタイプを指定します。必要な場合は、LOCATION 属性で指定された URL がリダイレクトに使用されます。

- NONE を指定すると、要求はリダイレクトされません。LOCATION 属性で指定された URL は無視されます。
- TEMPORARY を指定すると、要求が一時的にリダイレクトされます。LOCATION 属性で指定された URL はリダイレクトに使用され、応答に使用される状況コードは 302 (検出済み) です。
- PERMANENT を指定すると、要求は永続的にリダイレクトされます。LOCATION 属性によって指定された URL はリダイレクトに使用され、応答に使用される状況コードは 301 (永続的に移動) です。

SET URIMAP コマンドを使用して、REDIRECTTYPE 属性と LOCATION 属性を変更することができます。URIMAP 定義の作成時に REDIRECTTYPE (TEMPORARY) または REDIRECTTYPE(PERMANENT) を指定した場合、属性 ANALYZER、CONVERTER、HFSFILE、PIPELINE、PROGRAM、TEMPLATENAME、TRANSACTION、USERID、および WEBSERVICE はオプションとなります。 URIMAP 定義のインストール後に、CEMT または EXEC CICS コマンドを使用して REDIRECTTYPE 属性を NONE に設定すると、リストした属性のうち、URIMAP 定義で指定された属性はすべてアクティブになります。

SCHEME({**HTTP**|**HTTPS**})

この属性は、すべての USAGE オプションで使用します。

SCHEME は、URIMAP 定義が適用される URI のスキーム構成要素を指定します。これは、HTTP (SSL なし) または HTTPS (SSL あり) のいずれかです。URI のスキーム・コンポーネントの後に、区切り文字 **://** (コロンと 2 つのスラッシュ) を含めないでください。

HTTP スキームを指定する URIMAP は、HTTP スキームまたは HTTPS スキームのいずれかを使用して行われる Web クライアント要求を受け入れます。HTTPS スキームを指定する URIMAP は、HTTPS スキームを使用して行われる Web クライアント要求のみを受け入れます。ただし、トランスポートが WebSphere MQ である場合、HTTP スキームまたは HTTPS スキームのいずれかが指定された URIMAP は、HTTP スキームまたは HTTPS スキームのいずれかを使用して行われる Web クライアント要求を受け入れます。

SSL(ATTLSAWARE) TCPIPSERVICE に到着するインバウンド HTTP 要求は、セキュア接続を使用する必要があるため、常に HTTPS として処理されます。

SOCKETCLOSE({**0**|*hhmmss*})

この属性は USAGE(CLIENT) 用です。

SOCKETCLOSE は、CICS アプリケーションが使用を終了したクライアント HTTP 接続を CICS が開いたままにしておくかどうか、およびその場合の期間の長さを指定します。使用後、CICS により接続の状態が確認され、その接続は休止状態でプールに配置されます。休止接続を再利用できるのは、同じアプリケーションか、同じホストとポートに接続する別のアプリケーションです。

0
: CICS は、CICS アプリケーションが使用を終了したときに各クライアント HTTP 接続を閉じます。 CICS は、接続を再利用のためにプールに入れません。

hhmmss
: CICS アプリケーションがクライアント HTTP 接続の使用を終了したら、CICS は接続の状態を確認し、その接続を再利用のためにプールに入れます。再利用されない休止接続は、この変数で指定した時間の経過後に破棄されます。

接続プーリングにより、CICS のイベント処理に使用される HTTP EP アダプターのパフォーマンスが向上します。CICS Web サポート・アプリケーションの複数の呼び出しで、同じホストおよびポートに対する接続要求が行われます。また、Web サービス・アプリケーションによる複数の要求と応答が行われます。接続プーリングをアクティブにするには、アプリケーション・プログラムでの **INVOKE SERVICE** または **WEB OPEN** コマンドで URIMAP リソースを指定する必要があります。接続プーリングについて詳しくは、HTTP クライアントのパフォーマンスのための接続プールを参照してください。

STATUS({**ENABLED**|**DISABLED**})

この属性は、すべての USAGE オプションで使用します。

STATUS は、URIMAP 定義を使用可能と使用不可のどちらの状態でインストールするかを指定します。デフォルトは、使用可能の状態です。

この属性は、CICS バンドルによって動的に生成された URIMAP リソースでは無視されます。 URIMAP リソースの初期状況は、そのリソースを定義するバンドルの初期状況から導出されます。

TCPIPSERVICE(*name*)

この属性は、USAGE(SERVER)、USAGE(PIPELINE)、および USAGE(ATOM) で使用します。

TCPIPSERVICE は、PROTOCOL(HTTP) を指定している TCPIPSERVICE リソース定義の名前を 1 ～ 8 文字で指定します。TCPIPSERVICE は、この URIMAP 定義が関連するインバウンド・ポートを定義します。この属性を指定しないと、URIMAP 定義はすべてのインバウンド・ポートの要求に適用されます。

許容文字:
A-Z 0-9 $ @ #
CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

スキームとして HTTPS を使用する URIMAP 定義が、Web クライアントが行う要求と一致すると、CICS はその要求が使用するインバウンド・ポートで SSL が使用されているかどうかを検査します。SSL がポートに指定されていない場合、要求は 403 (Forbidden) 状況コードで拒否されます。 URIMAP 定義がすべてのインバウンド・ポートに適用されると、このチェックにより、Web クライアントは非セキュアなポートを使用してセキュアなリソースにアクセスすることができなくなります。 スキームとして HTTP が指定される URIMAP 定義には検査は実行されないので、Web クライアントはそうしたリソースにアクセスする際に無保護のポートと保護された (SSL) ポートのどちらも使用できます。

TCPIPSERVICE リソース定義では、各ポートで適用されるセキュリティー手段を指定します。SSL を使用するかどうかを選択できます。 SSL を使用する場合は、適用する正確なセキュリティー手段 (例えば、認証方式、クライアントおよびサーバーによる証明書の送信、メッセージの暗号化など) を選択する必要があります。 CICS Web サポート機能を安全に使用できるセキュリティー・フィーチャーについて詳しくは、CICS Web サポートのセキュリティーを参照してください。

TEMPLATENAME(*name*)

この属性は、静的応答が提供される場合に、USAGE(SERVER) で使用します。

TEMPLATENAME では、CICS 文書テンプレートの名前 (1 から 48 文字) を指定します。このテンプレートは、Web クライアントからの HTTP 要求に送信される静的応答の本体を構成します。これは、DOCTEMPLATE リソース定義を使用して定義する必要があり、その定義の TEMPLATENAME 属性によって、URIMAP 定義で使用する名前を指定します。詳細については、CICS 文書および文書テンプレートを参照してください。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

TEMPLATENAME または HFSFILE を指定する場合は、ANALYZER を NO に設定する必要があります。アプリケーション生成の応答にのみ関連する他の属性 (TRANSACTION、CONVERTER、および PROGRAM) はブランクのままにする必要があります。

パス・マッチングを使用する場合は、CICS 文書テンプレートの名前の最後および PATH 属性で指定されたパスの最後に、ワイルドカード文字としてアスタリスクを含めます。 CICS は、ワイルドカード文字でカバーされる各 HTTP 要求パスの一部を取得して、それをテンプレート名の最後の部分として置換します。

例えば、URIMAP 定義を作成する場合、以下のように指定された属性を使用するとします。PATH 属性:

```
findout/about/*
```

さらに、以下のように指定された TEMPLATENAME 属性を使用するとします。

```
templates.facts.*
```

この URIMAP 定義を使用して、着信 HTTP 要求を処理します。

```
http://www.example.com/findout/about/fish.html
```

CICS はアスタリスクの部分に代えて `fish.html` を、URIMAP 定義で指定されたテンプレート名に追加するので、テンプレート

```
templates.facts.fish.html
```

が静的応答を形成するのに使用されます。

TEMPLATENAME 属性にアスタリスクを 1 つだけ指定すると、選択したテンプレートの名前が、PATH 属性のワイルドカード文字に対応する URL の一部と同じになります。

照会ストリングが URI に存在する一方で、それが PATH 属性で使用されていない場合、TEMPLATENAME 属性を指定すると、CICS は、その照会ストリングの内容を、指定された CICS 文書テンプレートにシンボル・リストとして自動的に渡します。文書テンプレートで照会ストリングの内容を使用する場合、照会ストリングの内容と置き換える適切な変数を文書テンプレートに含めます。

TRANSACTION(名前)
この属性は USAGE(SERVER)、USAGE(PIPELINE) USAGE(ATOM)、および USAGE(JVMSERVER) 用です。

TRANSACTION では、アプリケーションの実行、またはパイプラインの開始に使用される別名トランザクションの名前を 1 から 4 文字で指定します。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

デフォルトの別名トランザクションは、以下のように、URIMAP のタイプごとに異なります。

- USAGE(SERVER) では CWBA トランザクションを使用します。
- USAGE(PIPELINE) では CPIH トランザクションを使用します。
- USAGE(ATOM) では CW2A トランザクションを使用します。

- USAGE(JVMSERVER) では CJSA トランザクションを使用します。

セキュリティー、モニターおよびアカウンティング、またはトランザクション・クラス制限のために、異なるトランザクション名を選択することができます。別名トランザクションにどのような名前を選択した場合も、常に同じプログラムを実行する必要があります。このプログラム USAGE 属性によって決定され、トランザクション定義で指定されたプログラムをオーバーライドします。

- USAGE(SERVER) の場合、プログラムは DFHWBA で、URIMAP 定義の PROGRAM 属性で指定される (またはアナライザー・プログラムによって指定される) アプリケーション・プログラムにリンクします。
- USAGE(PIPELINE) の場合、プログラムは DFHPIDSH であり、PIPELINE 属性で指定されるパイプラインと、WEBSERVICE 属性で指定される Web サービス (指定されている場合) を開始します。
- USAGE(ATOM) の場合、プログラムは DFHW2A という W2 ドメイン別名プログラムです。
- USAGE(JVMSERVER) の場合、プログラムは、アプリケーション要求に応じて JVM サーバーでタスクを実行する DFHSJTHP です。

USAGE(SERVER) の場合にのみ、ANALYZER 属性に YES が指定されている場合に、TRANSACTION 属性がアナライザー・プログラムへの入力として使用されますが、アナライザー・プログラムはそれをオーバーライドする可能性があります。あるいは、この属性をブランクのままにして、アナライザー・プログラムに指定させることもできます。アナライザーは他のタイプの URIMAP には使用されません。

URIMAP(*name*)

この URIMAP 定義の名前を指定します。この名前の長さは最大 8 文字です。属性は大/小文字混合で指定されます。URIMAP 定義では、その大/小文字が保持されます。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

USAGE({**SERVER**|**CLIENT**|**PIPELINE**|**ATOM**|**JVMSERVER**})

この URIMAP 定義を、HTTP サーバーとしての CICS 用 (SERVER)、HTTP クライアントとしての CICS 用 (CLIENT)、Web サービス用 (PIPELINE)、Atom フィード用 (ATOM)、または JVM サーバー用 (JVMSERVER) にするかを指定します。 USAGE 属性は、URIMAP 定義の他の属性のうち、どの属性が使用可能であるかを管理します。

- SERVER を指定する場合は、HTTP サーバーとしての CICS の URIMAP 定義を作成します。このタイプの URIMAP 定義は、Web クライアントからの着信 HTTP 要求の URI を、CICS リソースにマップするのに使用されます。 アプリケーションが生成する応答、または静的応答を 提供できます。アプリケーション生成の応答が、直接接続ユーザー・トランザクションによって処理され、Web 接続タスクをバイパスする対象になる可能性があることを必要とする要求。詳しくは、直接接続されたユーザー・トランザクションにより HTTP 要求が処理されるを参照してください。
- CLIENT を指定すると、HTTP クライアントとしての CICS 用に URIMAP 定義を作成します。このタイプの URIMAP 定義は、CICS がサーバー上で

HTTP リソースに要求を行うときに使用されるため、ご使用のアプリケーション・プログラム内で URI を識別する必要がなくなります。

- PIPELINE を指定する場合は、Web サービス用に URIMAP 定義を作成します。このタイプの URIMAP 定義は、インバウンド Web サービス要求 (つまり、クライアントが CICS で Web サービスを起動する要求) に使用されます。 着信要求のURIは、メッセージに対して 実行される処理を指定するWEBSERVICEおよびPIPELINE資源に 関連付けられます。 要求は、直接接続ユーザー・トランザクションによって処理され、Web 接続タスクをバイパスする対象になる場合があります。詳しくは、直接接続されたユーザー・トランザクションにより HTTP 要求が処理されるを参照してください。
- ATOM を指定する場合は、Atom フィード用の URIMAP 定義を作成します。このタイプの URIMAP 定義は、CICS が Atom フィードとして使用可能にするデータに関する着信要求で使用されます。この URIMAP定義は、要求URIをAtom文書を定義するATOMSERVICE資源定義に マップします。 要求は、直接接続ユーザー・トランザクションによって処理され、Web 接続タスクをバイパスする対象になる場合があります。詳しくは、直接接続されたユーザー・トランザクションにより HTTP 要求が処理されるを参照してください。
- JVMSERVER を指定する場合、JVM サーバーで実行されている Web アプリケーション用に URIMAP を作成します。このタイプの URIMAP は、着信アプリケーション要求を CICS トランザクションにマップするために使用されます。 URIMAP が着信要求の URI と一致しない場合、CICS は CJSA トランザクションを使用します。

USERID(*id*)

この属性は、USAGE(SERVER) 用 (アプリケーション生成の応答が提供される)、USAGE(PIPELINE) 用、USAGE(JVMSERVER) 用、および USAGE(ATOM) 用です。

USERID は、任意のクライアントで使用できる 1 文字から 8 文字までのデフォルトのユーザー ID です。アプリケーション生成の応答または Web サービスの場合、別名トランザクションがこのユーザー ID に付加されています。

USAGE(JVMSERVER) の場合、このユーザー ID は、Liberty プロファイル・サーバーで作業を実行するトランザクションに接続するために使用されます。トランザクションは、URI マップで指定することも、デフォルトのトランザクション CJSA を使用することもできます。基本認証が有効になっている場合、このユーザー ID は使用されません。CICS は、ユーザー ID とパスワードがアプリケーション要求の HTTP ヘッダーに入っていることを想定し、認証に失敗するとその要求を拒否します。セキュリティーが有効になっておらず、HTTP ヘッダーあるいは URI マップのいずれにもユーザー ID が存在しない場合、CICS 領域のデフォルト・ユーザー ID が使用されます。

その他の USAGE タイプの場合、接続に認証が必要なときは、認証されたユーザー ID を CICS がクライアントに直接要求するため、URIMAP 定義で指定したデフォルト・ユーザー ID は使用されません。代わりに、クライアントの認証済みユーザー ID が使用され、認証に失敗すると、その要求は拒否されます。認証の手順は、接続に関する TCPIPSERVICE 定義の AUTHENTICATE 属性によって指定されます。

アプリケーション生成の応答では、ANALYZER(YES) が指定されている場合、USERID 属性はアナライザー・プログラムへの入力として使用されますが、アナライザー・プログラムは、この属性をオーバーライドすることができます。あるいは、この属性をブランクのままにして、アナライザー・プログラムに指定させることもできます。アナライザーが使用されるのは、USAGE(SERVER) が指定された場合のみです。クライアントによって指定されるユーザー ID を、アナライザー・プログラムで変更することもできます。これらのいずれの方法でもユーザー ID が指定されていない場合は、アプリケーション生成応答のデフォルト・ユーザー ID が、CICS のデフォルト・ユーザー ID になります。

静的応答の場合、USERID 属性は適用されません。静的応答のリソース・セキュリティー検査を実行できるのは、クライアントの認証ユーザー ID を使用する場合だけです。

CICS 領域で代理ユーザー検査が使用可能な場合 (システム初期設定パラメーターとして XUSER=YES を指定した場合)、CICS は URIMAP 定義のインストールに使用されるユーザー ID が、USERID 属性に指定されたユーザー ID の代理として許可されているかどうかを検査します。代理ユーザー検査について詳しくは、Where surrogate user checking appliesを参照してください。

WEBSERVICE(*name*)

この属性は、USAGE(PIPELINE) で使用します。

クライアントが、この URIMAP 定義で指定された URI を使用して CICS に対してインバウンド Web サービス要求を行う場合、WEBSERVICE は Web サービスの名前を指定します。この名前は、WEBSERVICE リソース定義の 1 文字から 8 文字までの名前か、CICS Web サービス・アシスタントによって生成される Web サービスを表す大/小文字混合の最大 32 文字の名前です。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

WEBSERVICE リソースは、Web サービス設定でデプロイされた CICS アプリケーション・プログラムに関する実行時環境のいくつかの側面を定義します。この環境では、アプリケーションのデータ構造と SOAP メッセージの間のマッピングを、CICS ツールを使用して生成します。

この属性はオプションです。 ただし、USAGE(PIPELINE) が指定されていて WEBSERVICE リソースを指定しない場合、WEBSERVICE 名をパイプライン内のハンドラー・プログラムによって解決するか、WEBSERVICE リソースの代わりに代替アプリケーションまたは端末ハンドラーを使用する必要があります。

WEBSERVICE リソース

WEBSERVICE リソースは、Web サービス設定にデプロイされた CICS アプリケーション・プログラムの実行時環境の側面を定義します。Web サービス設定では、CICS サービス・アシスタントを使用して、アプリケーション・データ構造と SOAP メッセージの間のマッピングが生成されています。CICS は、WEBSERVICE リソースの通常のリソース定義メカニズムを提供しますが、一般的には、WEBSERVICE リソースはアシスタントによって生成される出力を使用して動的にインストールされます。

WEBSERVICE リソースで定義される実行時環境の側面は以下のとおりです。

パイプライン
: Web サービス要求および応答で動作するメッセージ・ハンドラーのセットを定義します。 WEBSERVICE リソースは、パイプライン構成ファイルを指定する個別の PIPELINE リソースを指定します。

Web サービス・バインディング・ファイル
: 実行時に、アプリケーション・データ構造と SOAP メッセージの間のマッピングを実行するのに使用される情報が含まれています。 Web サービス・バインディング・ファイルは CICS 提供のツールによって生成されます。

Web サービス記述
: Web サービス記述は、SOAP メッセージの実行時妥当性検査が必要な場合にのみ使用されます。各メッセージの妥当性検査は、Web サービス記述に組み込まれたそのスキーマに対して実行されます。

インバウンド Web サービス (つまり、クライアントが CICS 内で Web サービスを呼び出す要求) は、URIMAP リソースによって WEBSERVICE リソースに関連付けられます。 URIMAP は、インバウンド・メッセージ内の URI に適用される WEBSERVICE リソースを識別します。 WEBSERVICE はメッセージで実行される処理を指定します。

CICS は、WEBSERVICE リソースを作成し、それらをご使用の CICS 領域にインストールするための通常のリソース定義メカニズムを提供しますが、代わりにスキャン・メカニズムを使用して、稼働中の CICS システムに動的に WEBSERVICE リソースをインストールすることもできます。このアプローチの利点は、必要なリソース定義の量が削減されること、および CICS が開発時に提供される情報を直接使用できるという点です。

スキャン・メカニズムを呼び出すには、**PERFORM PIPELINE** コマンドを使用します。

動的にインストールされる WEBSERVICE の名前は、WEBSERVICE 定義の生成元である Web サービス・バインディング・ファイルの名前から取られ、最大 32 文字の長さになります。CSD からインストールされた、または **EXEC CICS CREATE WEBSERVICE** を使用した WEBSERVICE 定義の名前は 8 文字に制限されます。例えば、zFS 名が `/samples/Webservices/WSDir/InquireSingle.wsbind` である Web サービス・バインディング・ファイルは、`InquireSingle` という WEBSERVICE 定義を生成します。

CEMT および CEDA を使用した Web サービス・リソース定義のインストールについては、WEBSERVICE リソース定義のインストールを参照してください。BAS の作業については、BAS Web サービス・リソース定義の作業を参照してください。

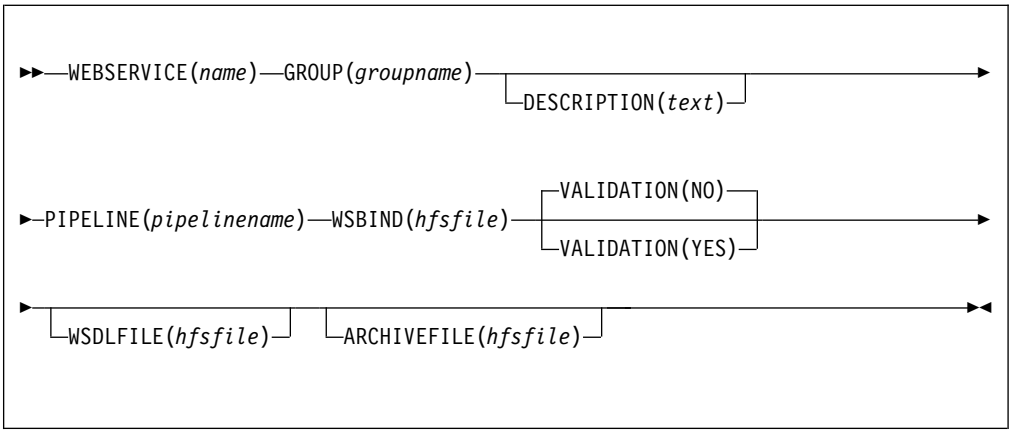
CICS バンドル内の WEBSERVICE リソース

CICS バンドルを使用して、WEBSERVICE リソース定義を作成、編集、およびインストールすることができます。この方法で WEBSERVICE リソースを作成した場合、CICS バンドルを使用してそのリソースのライフサイクルを管理する必要があり、リソースを単独で管理することはできません。

EXEC CICS INQUIRE WEBSERVICE コマンドまたは **CEMT INQUIRE WEBSERVICE** コマンドを使用して、CICS バンドルによって動的に生成される WEBSERVICE リソースを照会することができます。ただし、動的に生成された WEBSERVICE リソースに対して **DISCARD** コマンドを発行することはできません。このコマンドは、BUNDLE リソースに対して発行する必要があり、CICS がそれを WEBSERVICE リソースに適用します。 CICS バンドル内のリソースの定義について詳しくは、Defining CICS bundlesを参照してください。

WEBSERVICE 属性

WEBSERVICE リソースの構文と属性について記述します。



WEBSERVICE(*name*)

WEBSERVICE リソースの名前 (1 から 8 文字) を指定します。

許容文字:
A-Z a-z 0-9 $ @ # . / - _ % & ? ! : | " = ¬ , ; < >

文字 DFH は、CICS が使用するために予約されているため、DFH で始まる名前は使用しないことが推奨されます。 CICS は、それらを削除または破棄したり、新たに機能や修正をアップグレードあるいは提供する際に CICS リソースに置き換えたりすることができます。

ARCHIVEFILE(*hfsfile*)

1 つ以上の WSDL ファイルが含まれるアーカイブの完全修飾名または相対名 (1 から 255 文字) を指定します。そのアーカイブでサポートされる形式は `.zip` です。

- オンライン・リソース定義で定義された WEBSERVICE リソースのアーカイブでは、ファイル・パスは、先頭にスラッシュ (/) がある場合は完全修飾パスであり、それ以外の場合は CICS 領域ユーザー ID の HOME ディレクトリーからの相対パスです。
- CICS バンドルで定義された WEBSERVICE リソースのアーカイブの場合、ファイル・パスは、CICS バンドルのルート・ディレクトリーからの相対パスです。 zFS ファイルが、WEBSERVICE リソースと一緒に CICS バンドルにパッケージ化されている必要があります。詳しくは、Referencing zFS artifacts in a bundleを参照してください。

指定される値は UNIX ファイルに有効な名前でなければなりません。
- 組み込みスペース文字を含めることはできません。
- / 文字の連続インスタンスが含まれていてはなりません。
- 大/小文字の区別があります。

許容文字:

```
A-Z a-z 0-9 . / _ # @ -
```

GROUP(*groupname*)

各リソース定義には、GROUP 名が必要です。リソース定義はグループのメンバーになり、グループのインストール時に CICS システムにインストールされます。

許容文字:

```
A-Z 0-9 $ @ #
```

小文字を入力した場合は、大文字に変換されます。

GROUP 名は、8 文字以内の長さにすることができます。小文字は大文字として扱われます。

DESCRIPTION(*text*)

このフィールドには、定義するリソースの説明を入力することができます。説明文の長さは最大 58 文字です。使用できる文字の制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧に、対応する右括弧が必要です。
CREATE コマンドを使用する場合は、テキスト内のそれぞれの単一アポストロフィごとに、2 つのアポストロフィをコーディングしてください。

PIPELINE(*pipelinename*)

この WEBSERVICE リソースを関連付ける PIPELINE リソースの名前 (1 から 8 文字) を指定します。

許容文字:

```
A-Z 0-9 $ @ #
```

CREATE コマンドを使用しているのでなければ、小文字を入力した場合は、大文字に変換されます。

VALIDATION(**NO**|**YES**)

Web サービス記述内の対応するスキーマに対して SOAP メッセージの完全な妥当性検査を実行時に行うかどうかを指定します。スキーマに対するSOAPメッセージの検証は、 かなりの処理オーバーヘッドを引き起こすので、通常はVALIDATION(NO)を指定してください。

完全な妥当性検査を行うことで、送受信されるすべての SOAP メッセージが XML スキーマに対して有効な XML であることが保証されます。
VALIDATION(NO) が指定された場合、メッセージに正しい形式の XML が含まれることを確認するための検査は実行されますが、その XML が有効であることは保証されません。

WSBIND(*hfsfile*)

z/OS UNIX 上の Web サービス・バインディング・ファイルの完全修飾名または相対名 (1 から 255 文字) を指定します。

- オンライン・リソース定義で定義された WEBSERVICE リソースの Web サービス・バインディング・ファイルでは、ファイル・パスは、先頭にスラッシュ (/) がある場合は完全修飾パスであり、それ以外の場合は CICS 領域ユーザー ID の HOME ディレクトリーからの相対パスです。
- CICS バンドルで定義された WEBSERVICE の Web サービス・バインディング・ファイルの場合、ファイル・パスは、CICS バンドルのルート・ディレクトリーからの相対パスです。 zFS ファイルが、WEBSERVICE リソースと一緒に CICS バンドルにパッケージ化されている必要があります。詳しくは、Referencing zFS artifacts in a bundleを参照してください。

指定される値は UNIX ファイルに有効な名前でなければなりません。
- 組み込みスペース文字を含めることはできません。
- / 文字の連続インスタンスが含まれていてはなりません。
- 大/小文字の区別があります。

許容文字:

```
A-Z a-z 0-9 . / _ # @ -
```

WSDLFILE(*hfsfile*)

z/OS UNIX 上の Web サービス記述 (WSDL) ファイルの完全修飾名または相対名 (1 から 255 文字) を指定します。このファイルは、完全な実行時妥当性検査がアクティブである場合に使用されます。

- オンライン・リソース定義で定義された WEBSERVICE リソースの Web サービス記述ファイルでは、ファイル・パスは、先頭にスラッシュ (/) がある場合は完全修飾パスであり、それ以外の場合は CICS 領域ユーザー ID の HOME ディレクトリーからの相対パスです。
- CICS バンドルで定義された WEBSERVICE リソースの Web サービス記述ファイルの場合、ファイル・パスは、CICS バンドルのルート・ディレクトリーからの相対パスです。 zFS ファイルが、WEBSERVICE リソースと一緒に CICS バンドルにパッケージ化されている必要があります。詳しくは、Referencing zFS artifacts in a bundleを参照してください。

指定される値は UNIX ファイルに有効な名前でなければなりません。
- 組み込みスペース文字を含めることはできません。
- / 文字の連続インスタンスが含まれていてはなりません。
- 大/小文字の区別があります。

許容文字:

```
A-Z a-z 0-9 . / _ # @ -
```

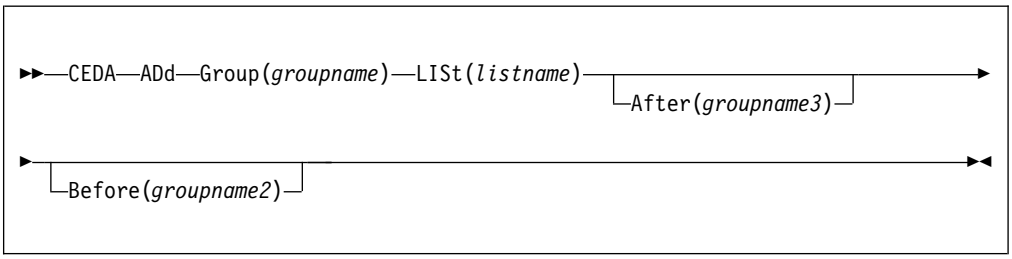
第 2 章 リソース管理トランザクション CEDA コマンド

CEDA トランザクションには、 CSD でリソース定義を扱うためのコマンドが多数あります。

CEDA ADD コマンド

CEDA ADD コマンドは、CSD 上のリストにグループを追加する場合に使用します。

構文



説明

ADD コマンドは、DISPLAY 画面から使用できます。

オプション

After(*groupname3***)**
このオプションを使用して、新規グループ名の配置を制御することができます。BEFORE および AFTER のいずれも指定しない場合、グループ名はリストの末尾に追加されます。

Before(*groupname2***)**
このオプションを使用して、新規グループ名の配置を制御することができます。BEFORE および AFTER のいずれも指定しない場合、グループ名はリストの末尾に追加されます。

Group(*groupname1***)**
追加するグループの名前を指定します。リスト内で既存の名前は指定できません。汎用グループ名は受け入れられません。グループを指定しない場合、現行のグループ名が追加されます。

LISt(*listname***)**
グループを追加するリストの名前を指定します。リストが既存でない場合は、新しいリストが作成されます。LIST を指定しない場合、グループ名は現行リストに追加されます (存在する場合)。総称リスト名は受け入れられません。

例

1. グループの追加によってリスト LA01 を作成するには、次のようにします。

```
ADD GROUP(GA001) LIST(LA01)
```

© Copyright IBM Corp. 1982, 2018

2. 場所を指定せずにリスト LA01 に別のグループを追加するには、次のようにします。

```
ADD GROUP(GA002) LIST(LA01)
```

この時点で、LA01 は以下のようになります。
GA001
GA002

3. リストの先頭に別のグループを追加するには、次のようにします。

```
ADD GROUP(GA003) LIST(LA01) BEFORE(GA001)
```

さらに、別のグループを GA001 と GA002 の間に追加します。

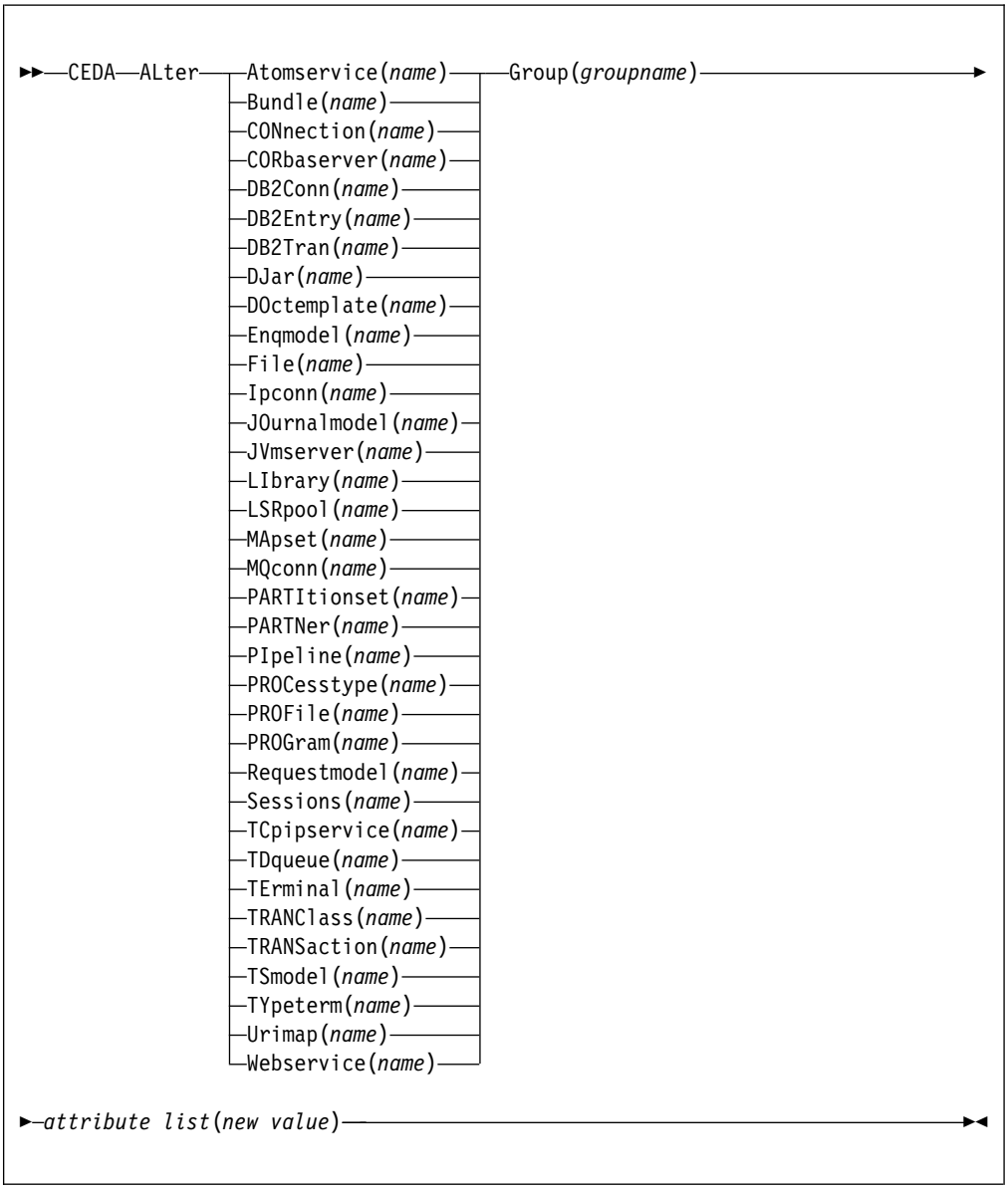
```
ADD GROUP(GA004) LIST(LA01) AFTER(GA001)
```

この時点で、LA01 は以下のようになります。
GA003
GA001
GA004
GA002

CEDA ALTER コマンド

CEDA ALTER コマンドは、既存のリソース定義の属性の一部または全部を変更する場合に使用します。

構文



説明

重要: 他の属性が依存している TYPETERM 定義の属性の値を変更する場合は、ALTER を使用しないでください。 DEVICE、SESSIONTYPE、または TERMMODEL に間違いがあった場合、その定義を削除してから、正しい値を使用して新規の定義を作成する必要があります。

ヌル属性値を指定することができます。例えば、次のように指定します。

```
ALTER FILE(TEST) GROUP(ACT1) DESCRIPTION()
```

ヌル値を指定した属性にデフォルト値がある場合、指定される値はフィールドのタイプによって決まります。以下に例を挙げます。

- 次のようにコマンドを発行したとします。

```
ALTER FILE(TEST) GROUP(ACT1) RLSACCESS()
```

この場合、RLSACCESS が指定されていないかのように動作します。RLSACCESS 属性には CVDA デフォルト値がありますが、無視されます。

- 次のようにコマンドを発行したとします。

```
ALTER FILE(TEST) GROUP(ACT1) DESCRIPTION()
```

このコマンドには、説明をブランクにする効果がありますが、DESCRIPTION フィールドにはデフォルト値がないため、これはオプションです。

- 次のようにコマンドを発行したとします。

```
ALTER FILE(TEST) GROUP(ACT1) PROFILE()
```

この場合、DFHCICSA のデフォルト値が PROFILE フィールドに入ります。このケースでは、デフォルト値は文字ストリングであり、CVDA 値ではありません。

CSD ファイルでリソース定義に対して行った変更は、そのリソース定義が属しているグループをインストールしない限り、稼働中の CICS システムには影響しません。

CEDA ALTER は、DISPLAY パネルから使用できます。変更を行った後に PF12 を使用すると、CEDA によって DISPLAY パネルが再度開かれ、日付と時刻のフィールドに「ALTER SUCCESSFUL」メッセージが表示されます。このコマンドを実行しても、何も変更を行わなかった場合は、「ALTER」コマンドがアスタリスクに置き換わります。

総称名を使用することで、1 回の ALTER コマンドで複数のリソース定義の同じ属性を同じように変更することができます。

オプション

属性リスト
変更する属性を指定します。

Group(*groupname***)**
変更するリソースが含まれるグループを指定します。

***resource*(*name*)**
属性を変更する対象のリソースのタイプおよび名前を指定します。

例

- プログラムを常駐するようにするには、次のようにします。

```
ALTER PROGRAM(ERR01) GROUP(GENMODS) RESIDENT(YES)
     DATALOCATION()
```

属性リストを指定せずに、以下のように入力したとします。

```
ALTER PROGRAM(ERR01) GROUP(GENMODS)
```

CEDA は、「ALTER SUCCESSFUL」メッセージの後に「上書き修正」パネルを表示します。

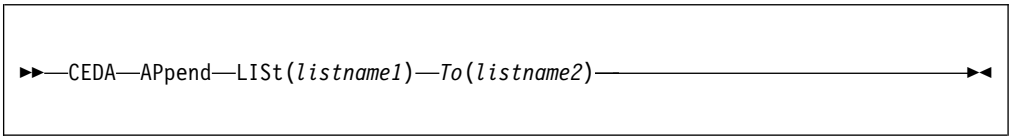
- プログラムのグループ全体の状況を変更するには、次のようにします。

```
ALTER PROGRAM(*) GROUP(GENMODS) STATUS(ENABLED)
```

CEDA APPEND コマンド

CEDA APPEND コマンドは、CSD の 1 つのリスト内のグループを別のリストの末尾に追加する場合に使用します。

構文



オプション

LISt(*listname1***)**
追加元のソース・リストを指定します。総称リスト名は受け入れられません。

To(*listname2***)**
追加先のターゲット・リストを指定します。総称リスト名は受け入れられません。 **listname2** が既に存在する場合、ソース・リストはそのリストに追加されます。 **listname2** が存在しない場合は、そのリストが作成されます。

例

LISTA という名前のリストには、以下のグループが含まれます。
GB001
GB002
GB003

LISTB という名前のリストには、以下のグループが含まれます。
G001
G002
G003

次のように、LISTB を LISTA に追加します。

```
APPEND LIST(LISTB) TO(LISTA)
```

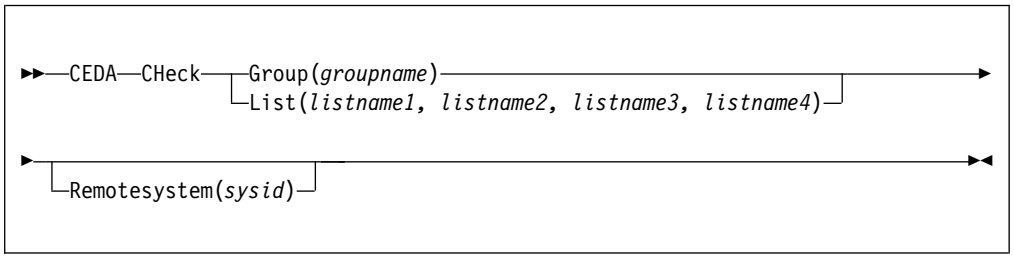
これを実行した後、LISTA には、以下のグループがこの順序で含まれます。
GB001
GB002
GB003
G001
G002
G003

また、LISTB には引き続き以下が含まれます。
G001
G002
G003

CEDA CHECK コマンド

CEDA CHECK コマンドは、CSD 上の一連のリソース定義の整合性を検査します。

構文



説明

CHECK コマンドは、グループ、リスト、あるいはリソース定義リストのクロスチェックを実行します。このコマンドは、リソース定義をインストールする前に使用する必要があります。

このコマンドは、グループまたはリスト内のリソース定義が他のリソース定義と整合性があるかを検査します。

例: 検査対象のリスト内の TRANSACTION ごとに、指定された PROGRAM 定義がいずれかのグループ内に存在しているかを検査します。検査の成功は、必ずしも稼働中のシステムで PROGRAM が使用可能であることを意味するわけではありません。

リストが GRPLIST システム初期設定パラメーターで指定されている場合は、初期始動またはコールド・スタート時の CICS の開始に使用する前に、そのリストを検査する必要があります。

INSTALL コマンドを使用して稼働中の CICS システムにグループをインストールする前に、そのグループを検査する必要があります。

ADD コマンドを使用してグループをリストに追加する前に、そのグループを検査することができます。 (グループは、自己完結型ではない場合があります。その場合、そのグループを単独で検査しても意味はありません。関連するリソース定義が含まれるグループと一緒に、そのグループをリストに入れてください。)

CHECK コマンドは、DISPLAY パネルから使用できます。

CEDA による定義の検査方法

CHECK コマンドは、グループ内の定義、あるいはリスト内のすべてのグループ内の定義の整合性を検査します。ただし、リソースのすべての属性をクロスチェックするわけではありません。 CHECK コマンドを使用した際に問題が見つからなかった場合でも、グループのインストール時にエラー・メッセージを受け取る可能性があります。

CHECK GROUP コマンドを使用すると、CEDA は、指定されたグループ内のすべてのリソースをクロスチェックし、グループを使用する準備ができていることを確認します。例えば、CHECK によって、グループ内のトランザクション定義が同じグループ内のプログラムを指定していないという警告が発行される場合があります。 (ただし、これはエラーではない場合があるので注意してください。そのグループは、そのプログラムを含まないグループと意図的にペアにされていることも考えられます。また、プログラムの自動インストールを行う場合は、定義が存在しないこともあります。)

CHECK LIST コマンドを使用すると、CEDA は、リストで指定されたすべてのグループをクロスチェックします。各グループを順番に検査するわけではありませんが、リストされているすべてのグループ内の定義をマージして、すべて検査します。この方法では、重複するリソース定義が存在する場合や、定義の参照先が存在しない場合に、警告が発行されます。

オプション

Group(*groupname***)**
検査するグループを指定します。汎用グループ名は受け入れられません。

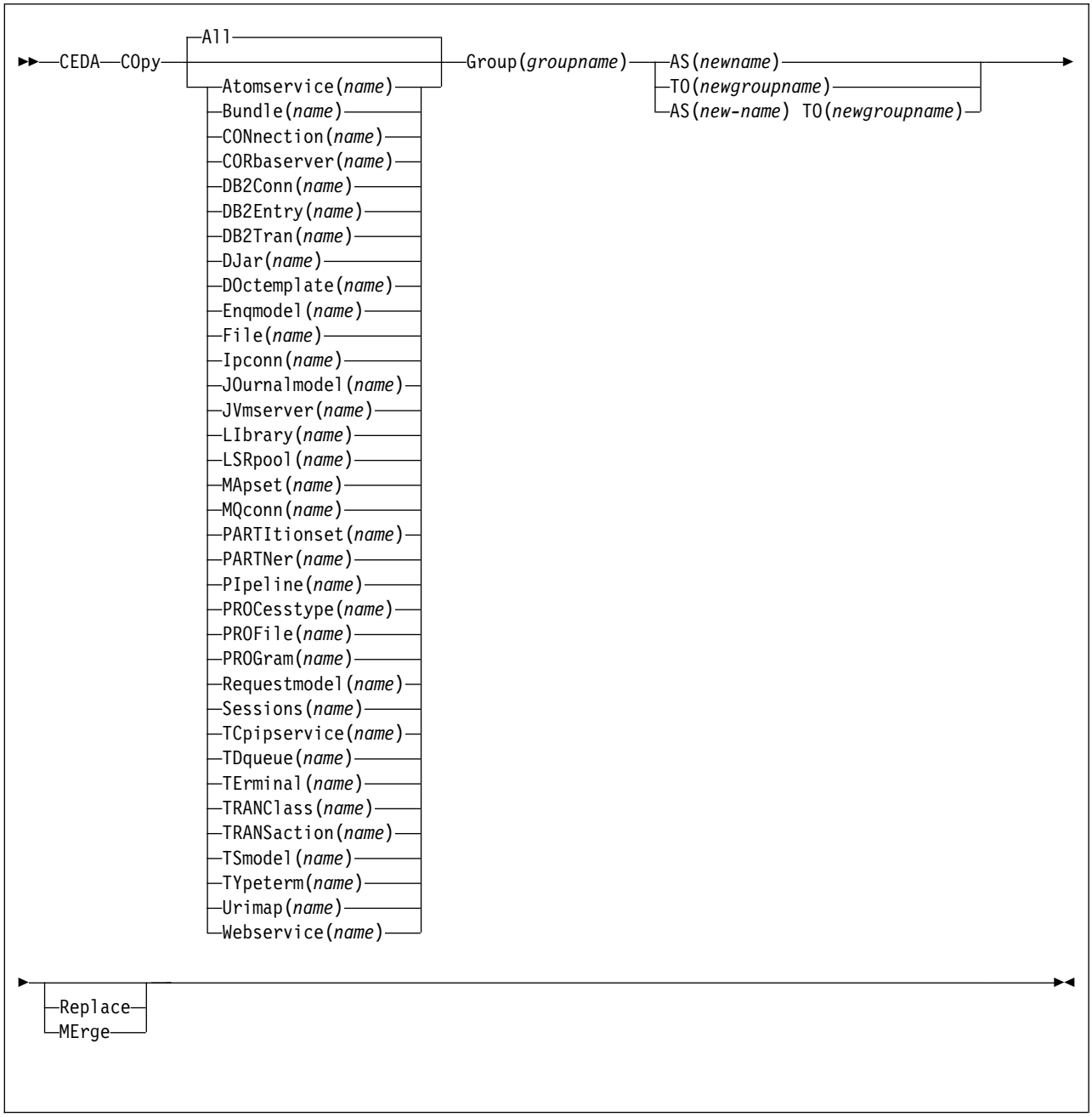
List(*listname1, listname2, etc.***)**
検査するリストを指定します。総称リスト名は受け入れられません。

Remotesystem(*sysid***)**
CHECK コマンドの発行元の領域と異なる SYSID を持つ CICS 領域内のグループまたはリストに対して検査を実行することを指定します。このオプションを使用しない場合、CHECK コマンドは、CHECK コマンドの発行元の領域の SYSID を使用します。

CEDA COPY コマンド

CEDA COPY コマンドは、CSD 上の同じグループ内あるいは異なるグループ間でリソース定義をコピーする場合に使用します。

構文



説明

MERGE および REPLACE のいずれも指定しない場合、重複するリソースを作成しようとしているためにコピーが失敗することを示す警告メッセージが発行されます。

オプション

AS(*newname*)
グループ内で定義をコピーする場合、AS を使用して名前を変更する必要があり

ます。別のグループに定義をコピーし、それと同時にその定義の名前を変更する場合にも、AS を使用することができます。 AS を使用する場合、総称名を使用することはできません。

Group(*groupname***)**
コピーする定義が含まれるグループを指定します。

MErge
これは、COPY コマンドで指定されたグループ内に重複する定義名が存在する場合に適用されます。 MERGE を指定すると、TO グループ内の重複する定義は置換されません。

置換
これは、COPY コマンドで指定されたグループ内に重複する定義名が存在する場合に適用されます。 REPLACE を指定すると、コピーされた定義によって、TO オペランドで指定されたグループ内にあった定義が置換されます。

resource(*name*)
属性をコピーする対象のリソースのタイプおよび名前を指定します。

TO(*newgroupname***)**
TO を使用して新規グループを指定することで、定義を別のグループにコピーすることができます。

例

TO オプションを使用して新規グループを定義することで、単一のリソース定義を新規グループにコピーすることができます。以下に例を挙げます。

```
COPY SESSIONS(L122) GROUP(CICSC1) TO(CICSC2)
```

同じグループ内でリソース定義をコピーすることができます。これを行う場合、AS オプションを使用して名前を変更する必要があります。以下に例を挙げます。

```
COPY TERMINAL(TD12) AS(TD34) GROUP(TERMVDU1)
```

グループ間でコピーを行う場合、AS オプションを使用して新規名を指定することで、コピーに新しい名前を付けることができます。

```
COPY PROGRAM(ABC01) GROUP(XYZ) AS(ABC02) TO(NEWXYZ)
```

(コピーを元の定義と同じ名前のまま残す場合は、後でインストールするときに間違えないように注意してください。)

名前を指定せずに ALL オプションを使用することで、グループ内のすべてのリソース定義を新規グループにコピーすることができます。以下に例を挙げます。

```
COPY ALL GROUP(N21TEST) TO(N21PROD)
```

TO オプションを使用して新規グループを定義することで、複数のリソース定義を新規グループにコピーすることができます。

総称リソース定義名を使用して、リソース・タイプが同じ定義の全部または一部を新規グループにコピーすることができます。以下に例を挙げます。

```
COPY CONNECTION(*) GROUP(CICSG1) TO(CICSG2)
COPY PROGRAM(N21++) GROUP(NTEST) TO(NPROD)
```

総称名を指定して ALL オプションを使用することで、グループ内のリソース定義の全部または一部を新規グループにコピーすることができます。以下に例を挙げます。

```
COPY ALL(N21*) GROUP(N21OLD) TO(N21NEW)
```

固有名を指定して ALL オプションを使用することで、グループ内でその名前を持つすべてのリソース定義 (異なるリソース・タイプでなければなりません) を新規グループにコピーすることができます。以下に例を挙げます。

```
COPY ALL(XMPL) GROUP(EXAMPLE) TO(EX2)
```

複数の定義を別のグループにコピーしており、グループに重複するリソース名が含まれている場合、MERGE または REPLACE を指定する必要があります。以下に例を挙げます。

```
COPY ALL GROUP(TAX1) TO(TAX2) MERGE
COPY ALL GROUP(TAX1NEW) TO(TAX1) REPLACE
```

以下の例では、GA001 という名前のグループを GA002 という名前のグループにコピーしています。GA002 は既に存在しており、重複するリソース定義はグループ GA001 の定義に置換されます。

```
COPY GROUP(GA001) TO(GA002) REPLACE
```

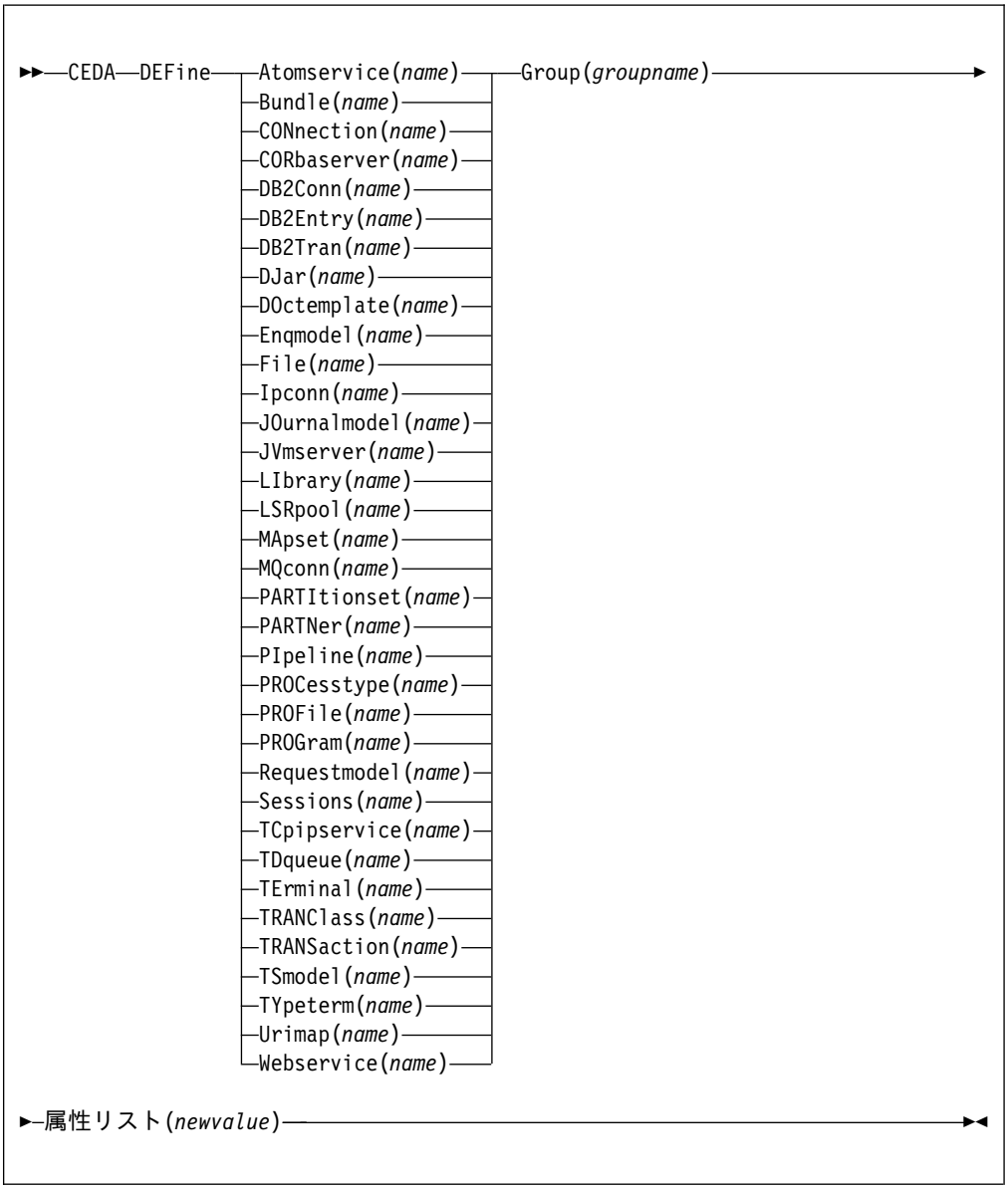
以下の例では、グループ GA003 をグループ GA004 にコピーしていますが、重複する定義が発生した場合は、グループ GA004 の定義を保持します。

```
COPY GROUP(GA003) TO(GA004) MERGE
```

CEDA DEFINE コマンド

CEDA DEFINE コマンドは、CSD 上に新しいリソース定義を作成する場合に使用します。

構文



説明

1 つのグループに含まれる複数のリソース定義に対しては、それらの定義が異なるリソース・タイプに対するものであるなら同じ名前を使用できます。以下に例を挙げます。

```
DEFINE PROGRAM(N28A) GROUP(N28APPL)
DEFINE TRANSACTION(N28A) GROUP(N28APPL)

DEFINE TERMINAL(USER) GROUP(USERDEF)
DEFINE PROGRAM(USER) GROUP(USERDEF)
```

同じ名前のリソース定義がある場合は、ほんとうに必要なリソース定義をインストールしていることを確認してください。 重複リソース定義名を参照してください。

コマンド行でリソース・タイプとリソースの名前を指定する必要があります。コマンド行では、グループやその他の属性も指定することができます。

上書き修正パネルにはすべての定義が表示されます。コマンド行に完全な DEFINE コマンドを入力した場合に NAME 属性および GROUPNAME 属性を変更できないことを除き、このパネルで任意の属性を変更することができます。

定義は、DEFINE コマンドを入力した時点で作成されています。それ以上変更する必要がない場合は、コマンド行から別のコマンドを入力することができます。

グループ内に同じ名前およびタイプのリソース定義が既に存在する場合、コマンド行で指定された属性はすべて無視され、既存のリソース定義が表示されます。その後、必要に応じて既存の値を上書きして変更することができます。定義を変更しない場合は、コマンド行から別のコマンドを入力することができます。

他の属性値が依存している値を上書きする場合は注意してください。

オプション

属性リスト
定義するリソース・タイプによっては、一部のリソースに、定義に含める必要がある属性があります。指定しない属性については、デフォルト値が使用されます。

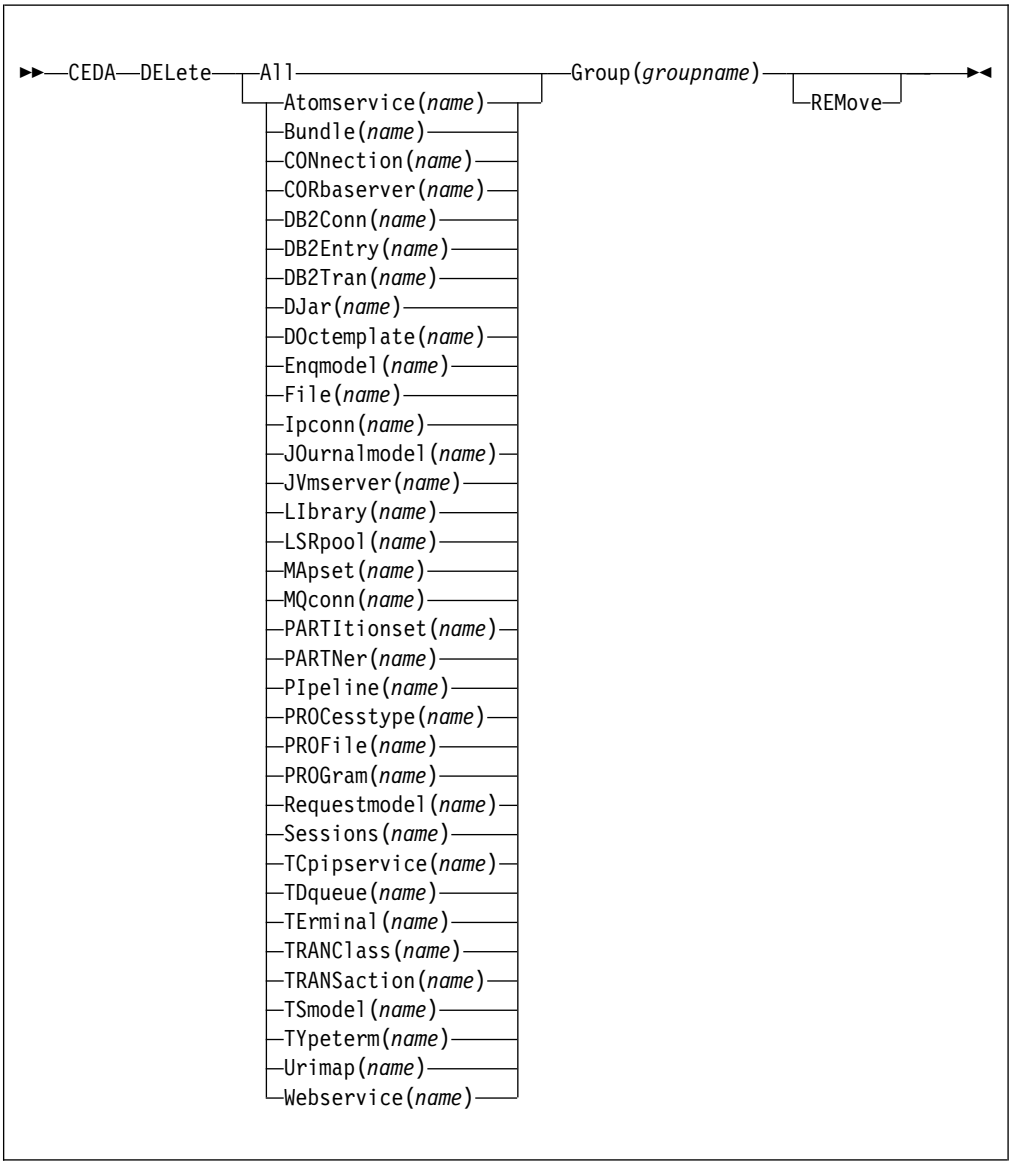
Group(*groupname***)**
定義するリソース定義を入れるグループの名前を指定します。総称グループ名は使用しないでください。存在しないグループ名を指定した場合、そのグループが作成されます。

resource(*name*)
定義するリソースのタイプおよび名前を指定します。総称リソース名は使用しないでください。

CEDA DELETE コマンド

CEDA DELETE コマンドは、CSD ファイルから 1 つ以上のリソース定義を削除する場合に使用します。

構文



説明

リソース定義の削除は、リストからのグループの除去とは異なります (360 ページの『CEDA REMOVE コマンド』を参照)。削除されたリソース定義は、実際に CSD ファイルから消去されます。

グループ内の最後のリソースを削除すると、そのグループは自動的に削除されます。空のグループは、存在することができません。

このコマンドは、アクティブな CICS システムにインストール済みの定義には影響しません。アクティブな CICS システムからインストール済みの定義を削除するには、CEMT DISCARD トランザクションまたは EXEC CICS DISCARD コマンドを使用することができます。

オプション

All (すべて)
グループからすべてのリソースを削除することを指定します。

Group(*groupname***)**
リソースが含まれるグループを指定します。総称グループ名は使用しないでください。

REMove
グループが削除された場合に、そのグループが含まれるすべてのリストからそのグループを除去することを指定します。

resource(*name*)
削除するリソースのタイプおよび名前を指定します。総称リソース名を使用することができます。

例

- グループ内のすべてのリソースを削除するには、次のようにします。

```
DELETE ALL GROUP(TOPS)
```

- グループ内のすべてのプログラムを削除するには、次のようにします。

```
DELETE PROGRAM(*) GROUP(NSOS)
```

CEDA DISPLAY コマンド

CEDA DISPLAY コマンドは、1 つ以上のグループ名、リスト名、またはリソースを表示する場合に使用します。

構文

オプション

Group(*groupname***)**
表示するグループの名前を指定します。総称グループ名を使用することができます。

List(*listname***)**
表示するリストの名前を指定します。総称リスト名を使用することができます。

REname
このオプションは、グループにのみ適用されます。RENAME を指定することで、グループ内のリソース定義の名前を変更することができます。

resource(*name*)
表示するリソースのタイプおよび名前を指定します。

例

- CSD ファイル上のすべてのグループを表示するには、次のようにします。

  ```
  DISPLAY GROUP(*)
  ```

- PROD から開始される 7 文字の名前を持つすべてのグループを表示するには、次のようにします。

  ```
  DISPLAY GROUP(PROD+++)
  ```

- CSD ファイル上のすべてのリストを表示するには、次のようにします。

  ```
  DISPLAY LIST(*)
  ```

- PROD から開始される 5 文字の名前を持つすべてのリストを表示するには、次のようにします。

  ```
  DISPLAY LIST(PROD+)
  ```

CEDA DISPLAY GROUP コマンド

CEDA DISPLAY GROUP コマンドは、フルスクリーン・パネルに 1 つ以上のグループ名を表示する場合に使用します。

パネル上の名前の横に以下のコマンドを入力することができます。

CHECK
DISPLAY ALL
EXPAND
INSTALL
LOCK
UNLOCK

制約事項: DISPLAY パネルから CONNECTION リソースおよび SESSIONS リソースをインストールすることはできません。これらのリソースは、グループに入れてインストールする必要があります。

このパネルでは、これらすべてコマンドを頭文字に省略することができます。グループ名を入力する必要はありません。各コマンドの構文については、このセクションの各コマンドを参照してください。

グループ名を表示するには、総称名を使用する必要があります。

CEDA DISPLAY LIST コマンド

CEDA DISPLAY LIST コマンドは、フルスクリーン・パネルに 1 つ以上のリスト名を表示する場合に使用します。

パネル上の名前の横に以下のコマンドを入力することができます。

CHECK
DISPLAY GROUP
EXPAND
LOCK
UNLOCK

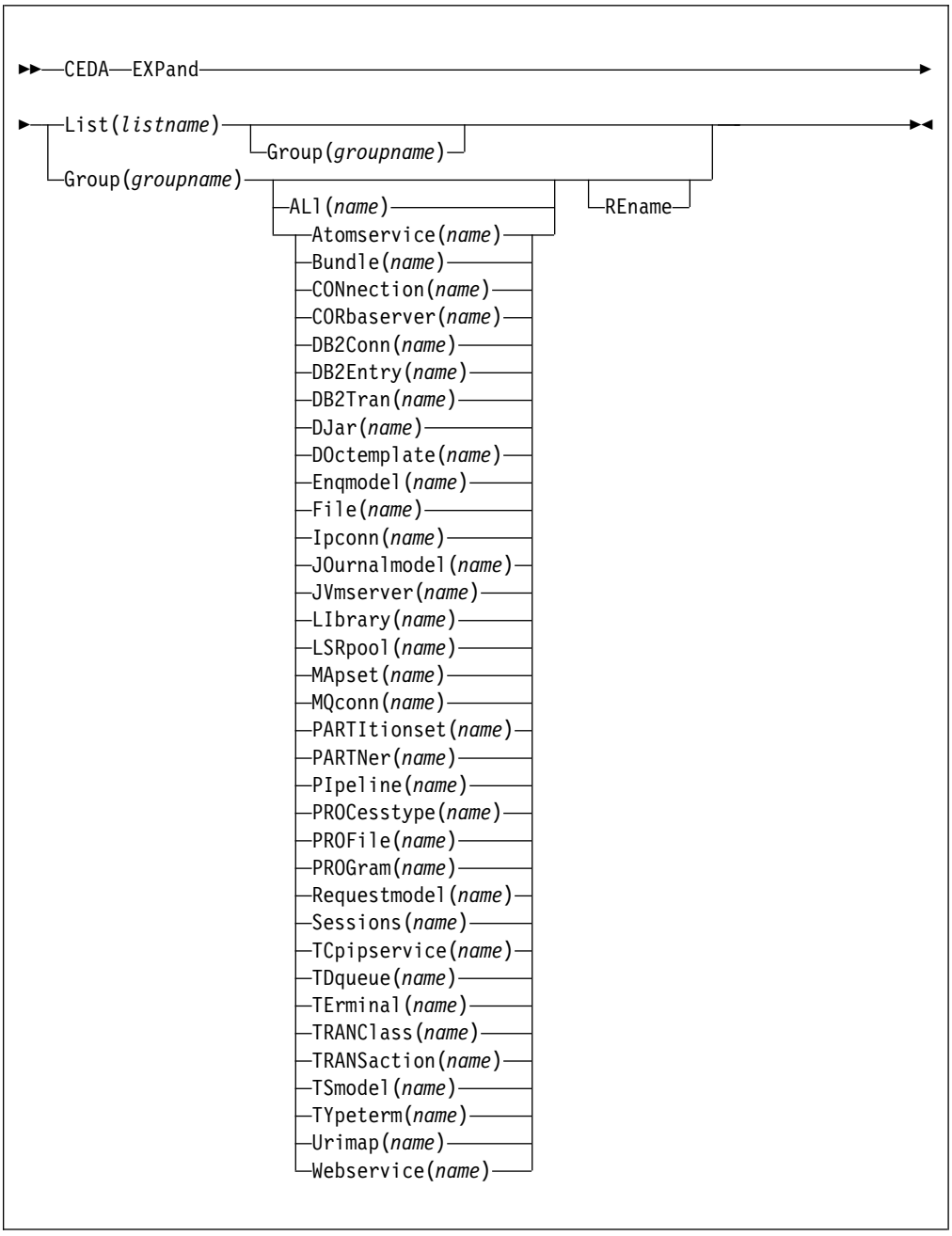
このパネルでは、これらすべてコマンドを頭文字に省略することができます。リスト名を入力する必要はありません。各コマンドの構文については、このセクションの各コマンドを参照してください。

リスト名を表示するには、総称リスト名を使用する必要があります。

CEDA EXPAND コマンド

CEDA EXPAND コマンドは、1 つ以上のグループまたはリストに含まれているリソース定義を表示する場合に使用します。表示されたリソース名に対して、他の CEDA コマンドを入力することができます。

構文



オプション

Group(*groupname***)**
展開するグループを指定します。

List(*listname***)**
展開するリストを指定します。

REname
このオプションは、グループにのみ適用されます。RENAME を指定することで、グループ内のリソース定義の名前を変更することができます。

resource(*name*)
展開したグループ内で表示するリソースのタイプおよび名前を指定します。

CEDA EXPAND GROUP コマンド

EXPAND GROUP コマンドは、1 つ以上のグループに含まれているリソース定義を表示するために使用します。表示されたリソース名に対して、他の CEDA コマンドを入力することができます。

パネル上の名前の横に以下のコマンドを入力することができます。
ALTER
COPY
DELETE
INSTALL
MOVE
RENAME
VIEW

これらすべてコマンドを頭文字に省略することができます。グループ名やリスト名を入力する必要はありません。各コマンドの構文については、このセクションの各コマンドを参照してください。

総称グループ名を展開することができます。以下に例を挙げます。

- CSD ファイル上のすべてのグループ内のすべてのリソース定義を表示するには、次のようにします。

```
EXPAND GROUP(*)
```

- PROD から開始される 7 文字の名前を持つグループ内のすべてのリソース定義を表示するには、次のようにします。

```
EXPAND GROUP(PROD+++)
```

1 つのリソース・タイプのみを表示するようにグループを展開することができます。リソース名として指定する名前は、総称名であっても構いません。以下に例を挙げます。

- CSD ファイル上のすべてのグループ内のすべての PROFILE 定義を表示するには、次のようにします。

```
EXPAND GROUP(*) PROFILE(*)
```

- グループ内の SZ から開始されるすべての TERMINAL 定義を表示するには、次のようにします。

```
EXPAND GROUP(ZEMGROUP) TERMINAL(SZ++)
```

総称名を共用するリソース定義に限定して、グループを展開することができます。以下に例を挙げます。

- 末尾が A1 であるすべてのタイプのすべてのリソース定義を表示するには、次のようにします。

```
EXPAND GROUP(REINDEER) ALL(*A1)
```

RENAME オプションを使用した場合、特殊パネルが表示されます。このパネルでは、リソース定義名を上書きすることができます。多くのリソース定義の名前を変更するには、これが迅速な方法です。

CEDA EXPAND LIST コマンド

EXPAND LIST コマンドは、1 つ以上のリストに含まれるグループを表示する場合に使用します。表示されたグループ名に対して、他の CEDA コマンドを入力することができます。

リストを展開する場合、パネル上の名前の横に以下のコマンドを入力することができます。

ADD
REMOVE

総称グループ名を使用することで、リストの一部を展開することができます。例えば、次のようにします。

```
EXPAND LIST(INITLIST) GROUP(DFH*)
```

総称リスト名を使用して、複数のリストを展開することができます。例えば、CSD ファイル上のすべてのリスト内のグループを表示するには、次のようにします。

```
EXPAND LIST(*)
```

PROD から開始される 5 文字の名前を持つリスト内のグループを表示するには、次のようにします。

```
EXPAND LIST(PROD+)
```

CEDA INSTALL コマンド

CEDA INSTALL コマンドを使用すると、指定されたグループまたはグループ・リスト内のリソース定義を、アクティブな CICS システムで使用可能にできます。

構文

説明

単一リソース・インストールを使用して、必要とするリソースのみをインストールできます。

遠隔通信および相互通信リソースに適用する場合は、単一リソース INSTALL コマンドに制限が適用されます。これらの制限は、リンクされたリソース定義 (CONNECTION 定義と SESSIONS 定義など) に適用されます。

以下のリソース・タイプをグループの一部としてのみインストールできます。

- CONNECTION 定義。ただし、ACCESSMETHOD(INDIRECT) を指定した CONNECTION を除く。
- SESSIONS 定義
- 同じ POOL を共有するパイプライン端末の TERMINAL 定義

TERMINAL および関連する TYPETERM 定義のために単一リソース INSTALL を使用する場合は、TERMINAL 定義によって参照される TYPETERM をインストール後、TERMINAL 定義をインストールします。

同じ制限が、TERMINAL 定義および TYPETERM 定義を含むグループをインストールする場合にも適用されます。TYPETERM 定義を含むグループをインストール後、TERMINAL 定義を含むグループをインストールします。同様に、TERMINAL 定義および TYPETERM 定義を含むグループを収めたリストをインストールする場合は、TYPETERM 定義を含むグループを、TERMINAL 定義を含むグループの前にする必要があります。また、Db2 リソース定義 (DB2CONN、DB2ENTRY、および DB2TRAN 定義) のグループを含むリスト内では、リストの最初のグループに DB2CONN を定義する必要があります。

ENQMODEL において単一リソース・インストールを使用する場合、ENQMODEL 定義は通常、デフォルトで ENABLED であることに注意してください。したがって、各 ENQMODEL をインストールする順序は、ENQMODEL 定義を使用可能化する規則に従う必要があります。これは、ネストされた総称 ENQname を形成する ENQMODEL 定義を、最も具体性の高いものから低いものの順に (例えば、ABCD*、ABC*、AB* の順に) 使用可能にする必要があることを意味します。

指定されたリソースがアクティブな CICS システム内に既に存在する場合、既存の定義が新しい定義によって置換されます。

既存の定義が置き換えられるのは、それが使用中でない場合に限られます。

グループ内にある 1 つ以上のリソースのインストールで、リソースが使用中であるため、あるいは別の理由でインストールに失敗した場合、以下のアクションが実行されます。

1. インストール処理はグループ内の次のリソースで続行されます。
2. グループのインストールの完了時、失敗したリソースがインストール可能セットの一部であった場合は、そのインストール可能セット内の全リソースがバックアウトされます。個別のリソース・レベルでコミットされたリソースは、バックアウトされません。インストール可能セットの一部としてコミットされるリソース・タイプのリストについては、INSTALL コマンド使用時の処理を参照してください。
3. メッセージが表示され、グループの一部だけがインストールされたことが示されます。また、インストールに失敗したリソースごとに、メッセージ・パネルにメッセージが表示され、それぞれの失敗理由が示されます。バックアウトされたリソースのメッセージは表示されません。

さらに、インストールがエラーで完了したという旨のメッセージが CEMT ログに書き込まれます。

CEDA INSTALL は、コンソールから MVS MODIFY コマンドを使用して実行できます。

オプション

Group(*groupname***)**
インストールするグループの名前、あるいはインストールするリソースが含まれているグループの名前を指定します。汎用グループ名は受け入れられません。

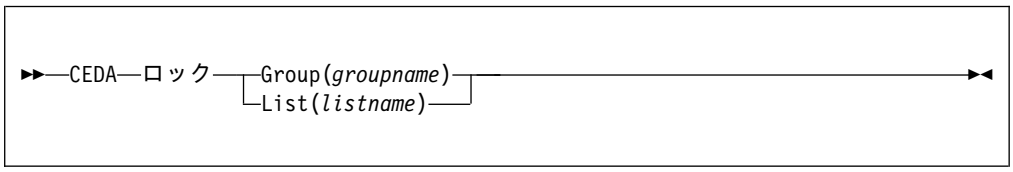
List(*listname***)**
インストールするグループのリスト、あるいはインストールするリソースが含まれているグループのリストを指定します。総称リスト名は受け入れられません。

resource(*name*)
インストールするリソースのタイプおよび名前を指定します。

CEDA LOCK コマンド

CEDA LOCK コマンドは、更新アクセスおよび削除アクセスを単一のオペレーター ID に制限する場合に使用します。

構文



説明

RDO の他のユーザーは、グループあるいはリストを使用、表示、およびコピーすることはできますが、変更および削除することはできません。

存在しないグループあるいはリストをロックすることができます。これにより、指定のグループやリストを将来自分で使用するために予約することができます。

ロックを解放する唯一の RDO コマンドは、UNLOCK コマンドです。他の RDO コマンドでは、グループやリストのアンロックはできません。例えば、グループ内のすべてのリソース、あるいはリスト内のすべてのグループを削除しても、ロックは残ります。

現行のグループやリストをロックする場合でも、GROUP あるいは LIST を指定する必要があります。

総称グループ名および総称リスト名は受け入れられません。

リソースのロックおよびアンロック

LOCK コマンドおよび UNLOCK コマンドを使用することで、同じオペレーター ID を持つオペレーターのみが変更を行えるように、グループあるいはリストに対する更新アクセスを制御することができます。

ロックは CSD ファイルで保持され、CICS を再始動しても有効なままです。ロックは、CICS 汎用アプリケーション ID (APPLID システム初期設定パラメーターで指定) とユーザーのオペレーター ID (OPIDENT) の組み合わせで識別されるユーザーによって所有されます。

OPIDENT は、ユーザーが RDO に使用する端末にサインオンするときに、ユーザーに関連付けられます。 OPIDENT について詳しくは、CICS セグメントを参照してください。サインオンしていないユーザー、あるいは異なる OPIDENT を持つユーザーは、ロックされたグループを変更する操作を実行することはできません。ただし、ロックされているグループに対しても、以下のことはすべてのユーザーが可能です。

- CHECK
- COPY
- DISPLAY
- INSTALL
- VIEW

ロックは、同じシステム上で同じオペレーター ID を持つユーザーのみが、UNLOCK コマンドを使用して除去することができます。

オプション

Group(*groupname***)**
ロックするグループを指定します。

List(*listname***)**
ロックするリストを指定します。

例

- リスト L1 をロックするには、次のようにします。

```
LOCK LIST(L1)
```

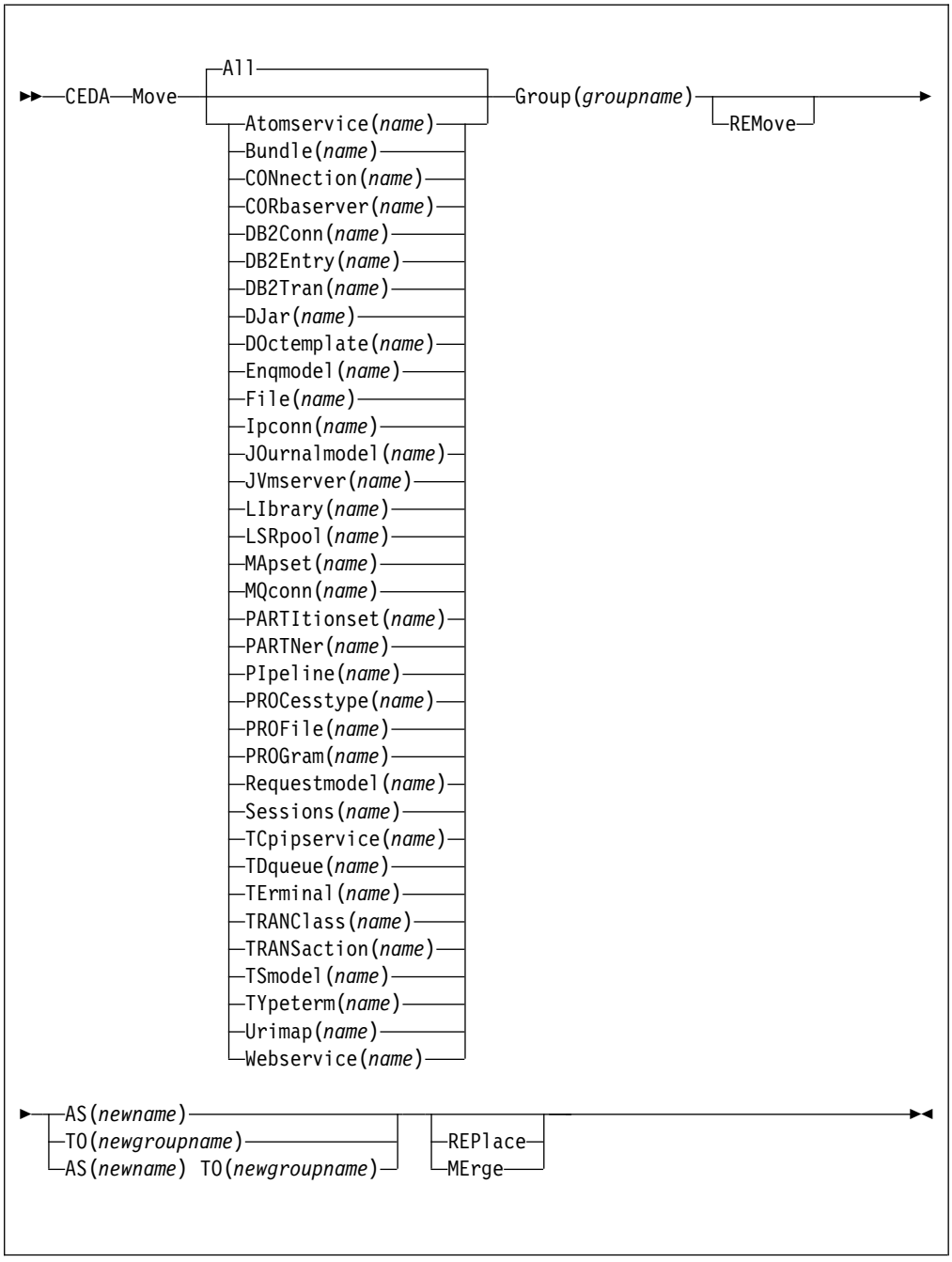
- グループ G1 をロックするには、次のようにします。

```
LOCK GROUP(G1)
```

CEDA MOVE コマンド

CEDA MOVE コマンドは、1 つ以上のリソース定義をグループ間で移動する場合に使用します。

構文



説明

このコマンドは、GROUP オプションで指定されたグループから TO オプションで指定されたグループにリソース定義を移動します。

グループ内の最後のリソースを別のグループに移動すると、そのグループは自動的に削除されます。空のグループは、存在することができません。

MERGE および REPLACE のいずれも指定しない場合、重複するリソース定義を作成しようとしていることを示す警告メッセージが発行されます。定義は移動されません。

オプション

AS(*newname***)**
グループ内で定義を移動する場合、AS を使用して名前を変更する必要があります。別のグループに定義を移動し、それと同時にその定義の名前を変更する場合にも、AS を使用することができます。 AS を使用する場合、総称名を使用することはできません。

Group(*groupname***)**
移動する定義が含まれるグループを指定します。

MErge
これは、MOVE コマンドで指定されたグループ内に重複する定義名が存在する場合に適用されます。 MERGE を指定すると、TO グループ内の重複する定義は置換されません。

REMove
グループ内の最後のリソースが他の場所に移動されたためにグループが削除された場合に、そのグループが含まれるすべてのリストからそのグループを除去することを指定します。

REPlace
これは、MOVE コマンドで指定されたグループ内に重複する定義名が存在する場合に適用されます。 REPLACE を指定すると、移動された定義によって、TO オペランドで指定されたグループ内にあった定義が置換されます。

resource(*name*)
移動するリソースのタイプおよび名前を指定します。デフォルトは ALL です。これは、グループ内のすべてのリソース定義を別のグループに移動します。

TO(*newgroupname***)**
TO を使用して新規グループを指定することで、定義を別のグループに移動することができます。

例

- 単一のリソース定義を移動する場合は、AS オプションを使用して新規名を指定することで、同時にその定義の名前を変更することができます。以下に例を挙げます。

```
MOVE PARTITIONSET(PSETQ1) GROUP(PSET1) AS(PSETQ4)
TO(PSET2)
```

- 総称リソース定義名を指定して、リソース・タイプが同じ定義の全部または一部を移動することができます。以下に例を挙げます。

```
MOVE TRANSACTION(*) GROUP(DENTRY) TO(TEST1)
MOVE MAPSET(ACCT+++) GROUP(ACCOUNTS1) TO(ACCOUNTS2)
```

- グループ内のすべてのリソース定義を新規グループに移動するには、ALL を使用することができます。以下に例を挙げます。

```
MOVE ALL GROUP(N21TEST) TO(N21PROD)
```

- 総称名を指定して ALL を使用することで、グループ内の条件を満たすすべてのリソース定義を新規グループに移動することができます。以下に例を挙げます。

  ```
  MOVE ALL(N21*) GROUP(N21OLD) TO(N21NEW)
  ```

- 固有名を指定して ALL を使用することで、グループ内でその名前を持つすべてのリソース定義 (異なるリソース・タイプでなければなりません) を新規グループに移動することができます。以下に例を挙げます。

  ```
  MOVE ALL(XMPL) GROUP(EXAMPLE) TO(EXAMPLE2)
  ```

- グループ X の定義をグループ Y の定義にマージし、すべての重複について Y のバージョンを保持するには、次のようにします。

  ```
  MOVE GROUP(X) TO(Y) MERGE
  ```

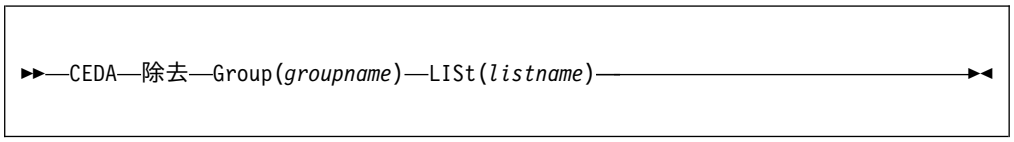
- グループ X の定義をグループ Y の定義に結合し、すべての重複について X のバージョンを保持するには、次のようにします。

  ```
  MOVE GROUP(X) TO(Y) REPLACE
  ```

CEDA REMOVE コマンド

CEDA REMOVE コマンドは、リストからグループの名前を除去する場合に使用します。

構文



説明

グループおよびそのすべてのリソース定義は、CSD ファイルに残っています。リストから最後のグループが除去されると、そのリストは CSD ファイル上に存在しなくなります。

総称リスト名は受け入れられません。

総称グループ名を指定することで、1 つのコマンドでリストから多数のグループまたはすべてのグループを除去することができます。

グループが削除されたら、ユーザーはそのグループが含まれていたすべてのリストからそのグループを除去するように要求することができます。リストから最後のグループが除去されると、リストが削除されます。

オプション

Group(*groupname***)**
除去するグループを指定します。

List(*listname***)**
グループを除去する元のリストを指定します。

例

- リスト LL02 には、以下のグループが含まれます。
  G001
  X001
  XG001
  G002
  G003
  X002
  G004

  G から開始されるすべてのグループを除去するには、次のようにします。

```
REMOVE GROUP(G*) LIST(LL02)
```

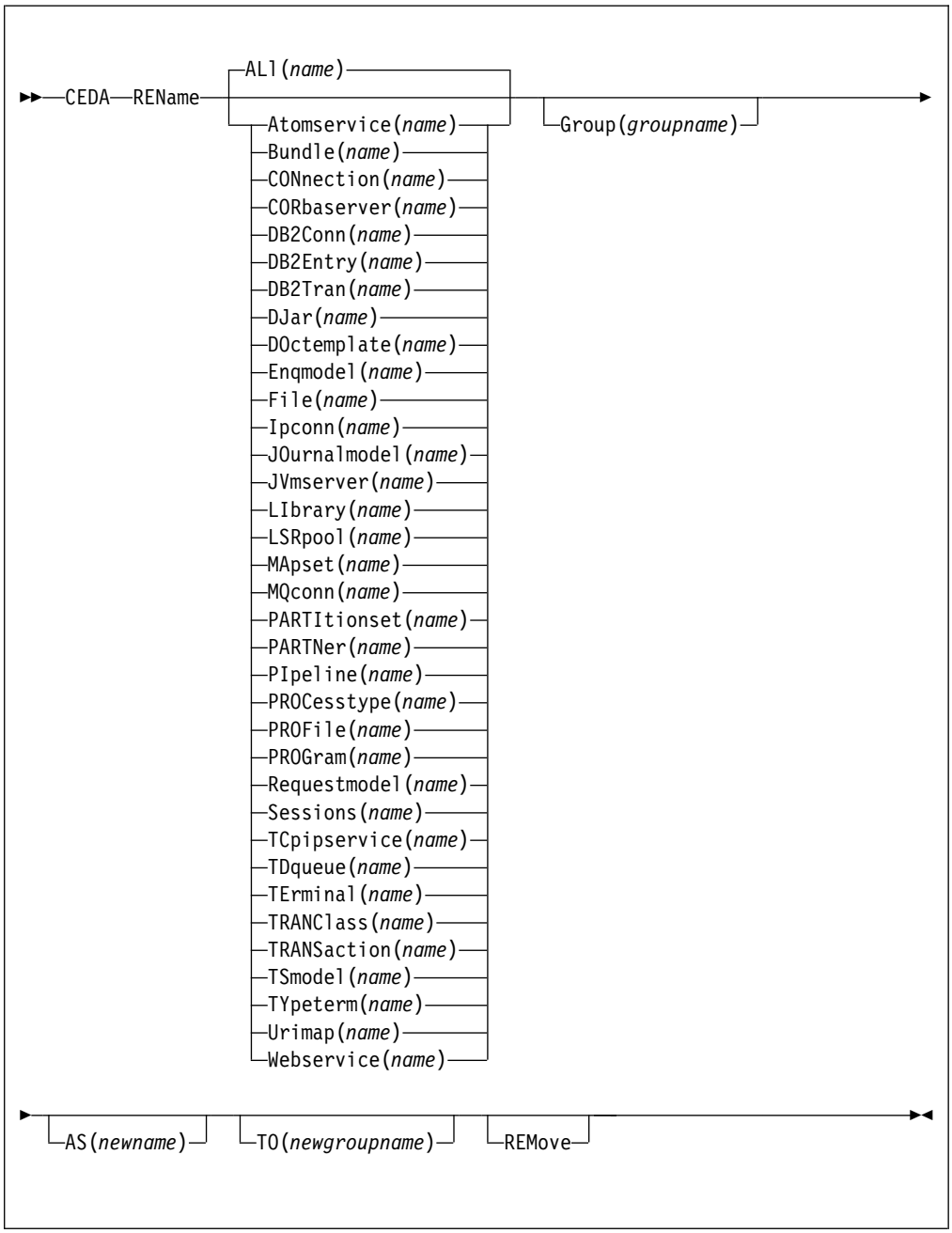
  これにより、以下が残ります。
  X001
  XG001
  X002
- リストを完全に除去するには、次のようにします。

```
REMOVE GROUP(*) LIST(LL02)
```

CEDA RENAME コマンド

CEDA RENAME コマンドは、CSD 上のリソースの名前を変更する場合、あるいはそのリソースを別のグループに移動する場合に使用します。

構文



説明

RENAME オプションを指定して DISPLAY コマンドまたは EXPAND コマンドを使用することで、リソース定義の名前を変更することもできます。これらのコマンドについて詳しくは、 348 ページの『CEDA DISPLAY コマンド』または 351 ページの『CEDA EXPAND コマンド』を参照してください。

すでにターゲット・グループに存在する名前にリソース定義の名前を変更することはできません。

オプション

AS(*newname***)**
グループ内で定義をコピーする場合、AS を使用して名前を変更する必要があります。別のグループに定義を移動し、それと同時にその定義の名前を変更する場合にも、AS を使用することができます。 AS を使用する場合、総称名を使用することはできません。

Group(*groupname***)**
名前変更または移動する定義が含まれるグループを指定します。汎用グループ名は受け入れられません。

REMove
グループが削除された場合に、そのグループが含まれるすべてのリストからそのグループを除去することを指定します。

resource(*name*)
移動または名前変更するリソースのタイプおよび名前を指定します。デフォルトは ALL です。これは、グループ内のすべてのリソース定義を別のグループにコピーします。総称リソース定義名は受け入れられません。

TO(*newgroupname***)**
TO を使用して新規グループを指定することで、定義を別のグループに移動することができます。

例

- リソースの名前を変更し、そのリソースを現行グループ内に保持するには、次のようにします。

```
RENAME PROFILE(PROF1) AS(NEWPROF) GROUP(PROFS)
```

- リソース・タイプの代わりに ALL オプションを使用することで、同じ名前を共用するすべてのリソース定義の名前を新規名に変更することができます。以下に例を挙げます。

```
RENAME ALL(TVA) AS(XTVA) GROUP(XTVA1)
RENAME ALL(USER) AS(OLDU) GROUP(USERDEF)
```

- リソース定義を TO オプションで指定した新規グループに移動すると同時に、そのリソース定義の名前を変更することができます。 (これは、MOVE コマンドでも行うことができます。) 以下に例を挙げます。

```
RENAME PROGRAM(N20ZA) AS($SOSERR) GROUP(N20) TO($MODULES)
```

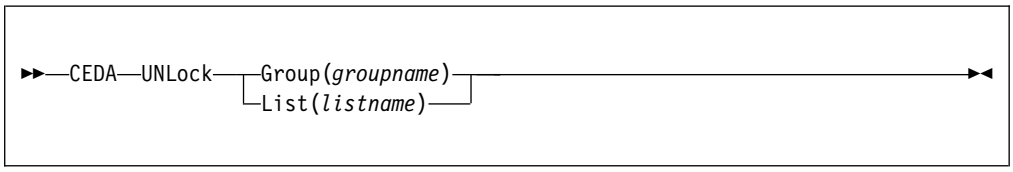
- TO オプションを使用して、同じ名前を持つすべてのリソース定義をグループ間で移動すると同時に、それらのリソース定義の名前を変更することができます。以下に例を挙げます。

```
RENAME ALL(USER) GROUP(USERDEF) AS(TEMP) TO(TEMPGRP)
```

CEDA UNLOCK コマンド

CEDA UNLOCK コマンドは、定義のグループまたはリストからロックを除去する場合に使用します。

構文



説明

UNLOCK コマンドは、RDO LOCK コマンドを使用してリストあるいはグループに適用されたロックを除去することができる唯一の RDO コマンドです。

存在しないグループあるいはリストをアンロックすることができます。

現行のグループあるいはリストをアンロックする場合でも、UNLOCK はその両方で使用することができるため、GROUP オプションまたは LIST オプションを指定する必要があります。

リソースのロックおよびアンロック

LOCK コマンドおよび UNLOCK コマンドを使用することで、同じオペレーター ID を持つオペレーターのみが変更を行えるように、グループあるいはリストに対する更新アクセスを制御することができます。

ロックは CSD ファイルで保持され、CICS を再始動しても有効なままです。ロックは、CICS 汎用アプリケーション ID (APPLID システム初期設定パラメーターで指定) とユーザーのオペレーター ID (OPIDENT) の組み合わせで識別されるユーザーによって所有されます。

OPIDENT は、ユーザーが RDO に使用する端末にサインオンするときに、ユーザーに関連付けられます。 OPIDENT について詳しくは、CICS セグメントを参照してください。サインオンしていないユーザー、あるいは異なる OPIDENT を持つユーザーは、ロックされたグループを変更する操作を実行することはできません。ただし、ロックされているグループに対しても、以下のことはすべてのユーザーが可能です。

- CHECK
- COPY
- DISPLAY
- INSTALL
- VIEW

ロックは、同じシステム上で同じオペレーター ID を持つユーザーのみが、UNLOCK コマンドを使用して除去することができます。

オプション

Group(*groupname***)**
アンロックするグループを指定します。汎用グループ名は受け入れられません。

List(*listname***)**
アンロックするリストを指定します。総称リスト名は受け入れられません。

例

- グループ G1 をアンロックするには、次のようにします。

```
UNLOCK GROUP(G1)
```

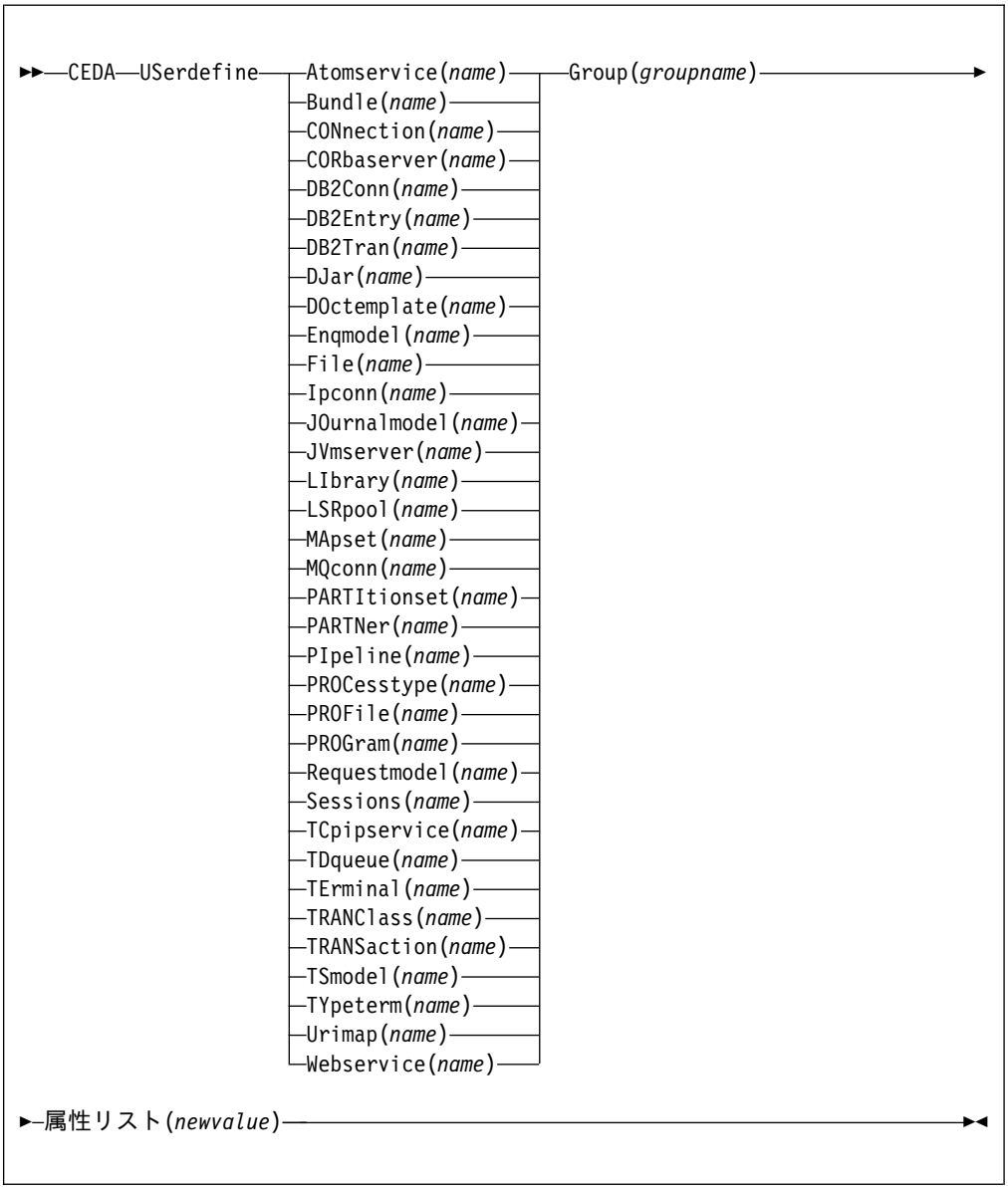
- リスト L1 をアンロックするには、次のようにします。

```
UNLOCK LIST(L1)
```

CEDA USERDEFINE コマンド

CEDA USERDEFINE コマンドは、ユーザー定義のデフォルト属性を使用して CSD 上に新規のリソース定義を作成する場合に使用します。

構文



説明

USERDEFINE は、DEFINE コマンドの代替コマンドです。USERDEFINE は、CICS が提供するデフォルト値ではなく、ユーザー独自のデフォルトを使用します。それ以外については、DEFINE と完全に同じです。

独自のデフォルトをセットアップするには、DEFINE を使用して、USERDEF という名前のグループに USER という名前でダミーのリソース定義を作成します。それぞれのダミー・リソース定義は完全でなければなりません (例えば、トランザクションを USERDEFINE するときに常にプログラム名を提供するとしても、トランザクション定義でプログラム定義の名前を指定する必要があります)。 USERDEFINE を使用する前にダミーのリソース定義をインストールする必要はありません。

デフォルト値を設定するリソースのタイプごとに、これを行います。グループ内のすべての定義が同じ名前 (USER) ですが、それぞれ異なるリソース・タイプを定義しているため、各定義は固有です。

オプション

属性リスト
属性リストは、定義するリソース・タイプに依存します。一部のリソースでは、定義中に必ず含めなければならない属性があります。指定しない属性については、デフォルト値が使用されます。

Group(*groupname*)
定義するグループの名前。

例

インストール時に、アセンブラー・プログラマーがデフォルト言語としてアセンブラーを使用して、USER という名前のダミー・プログラム定義を作成しました。彼らは、USERDEFINE を使用してプログラムを CICS に定義します。

1. 最初に、USER という名前のプログラムをグループ USERDEF に定義する必要があります。これは、コマンドで行うことができます。

   ```
   CEDA DEFINE PROGRAM(USER) GROUP(USERDEF)
   ```

   次の図は、このコマンドの結果として表示されるパネルを示しています。

```
  DEFINE PROGRAM(USER)
GROUP(USERDEF)            CICS RELEASE = 0620
  OVERTYPE TO MODIFY
   CEDA  DEFine
    PROGram        : USER
    Group          : USERDEF
    DEscription  ==>
    Language     ==>                    CObol | Assembler | Le370 | C | Pli
    RELoad       ==> No                 No | Yes
    RESident     ==> No                 No | Yes
    USAge        ==> Normal             Normal | Transient
    USElpacopy   ==> No                 No | Yes
    Status       ==> Enabled            Enabled | Disabled
    RSl            : 00                 0-24 | Public
    Cedf         ==> Yes                Yes | No
    DAtalocation ==> Any                Below | Any
 I New group USERDEF created
                                                SYSID=ABCD    APPLID=DBDCCICS
   DEFINE SUCCESSFUL                      TIME:  11.24.39   DATE:  97.359
 PF 1 HELP 2 COM 3 END           6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

2. LANGUAGE オプションとして ASSEMBLER を入力し、Enter を押します。

```
  OVERTYPE TO MODIFY
   CEDA  USerdefine
    PROGram        : USER
    Group          : USERDEF
    DEscription  ==>
    Language     ==> Assembler          CObol | Assembler | Le370 | C | Pli
    RELoad       ==> No                 No | Yes
    RESident     ==> No                 No | Yes
    USAge        ==> Normal             Normal | Transient
    USElpacopy   ==> No                 No | Yes
    Status       ==> Enabled            Enabled | Disabled
    RSl            : 00                 0-24 | Public
    Cedf         ==> Yes                Yes | No
    DAtalocation ==> Any                Below | Any



                                                SYSID=ABCD    APPLID=DBDCCICS
   DEFINE SUCCESSFUL                      TIME:  11.24.41   DATE:  97.359
 PF 1 HELP 2 COM 3 END           6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

これで、新規プログラムを定義するたびに、USERDEFINE コマンドを使用してデフォルト値 ASSEMBLER を自動的に取得することができるようになりました。したがって、グループ GRP に新規プログラム P2 を定義したい場合は、次のコマンドを入力します。

```
CEDA USERDEFINE PROGRAM(P2) GROUP(GRP)
```

次の図は、このコマンドの結果として表示されるパネルを示しています。

```
  USERDEFINE PROGram(P2)
GROUP(GRP)
  OVERTYPE TO MODIFY
   CEDA  USerdefine
    PROGram        : P2
    Group          : GRP
    DEscription  ==>
    Language     ==> Assembler          CObol | Assembler | Le370 | C | Pli
    RELoad       ==> No                 No | Yes
    RESident     ==> No                 No | Yes
    USAge        ==> Normal             Normal | Transient
    USElpacopy   ==> No                 No | Yes
    Status       ==> Enabled            Enabled | Disabled
    RSl            : 00                 0-24 | Public
    Cedf         ==> Yes                Yes | No
    DAtalocation ==> Any                Below | Any
                                                          APPLID=DBDCCICS
   USERDEFINE SUCCESSFUL                 TIME:  11.25.48   DATE:  97.359
 PF 1 HELP 2 COM 3 END           6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

LANGUAGE 属性に ASSEMBLER オプションが表示されているのを確認できます。 DEFINE コマンド・パネルで行うのと同様に、このパネルでオプション値を上書きし、定義を完成させることができます。

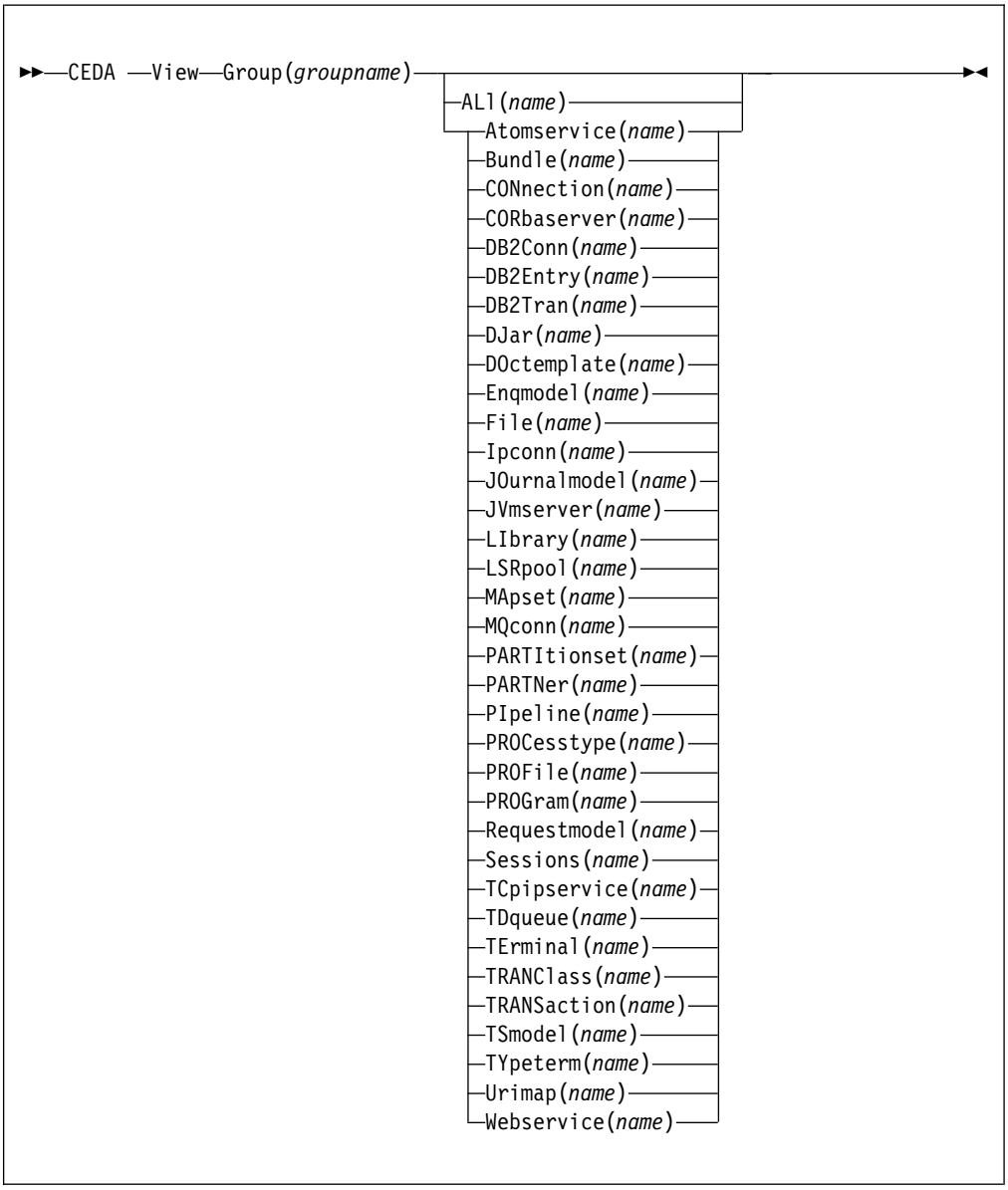
USER リソース定義で独自のデフォルトのセットアップが完了したら、そのリソース・タイプに対して USERDEFINE コマンドを使用するすべての人が、このデフォルト値を取得します。

USER を他の名前に変更して、独自のダミー・リソース定義を定義することで、独自のデフォルト値を使用することができます。ただし、通常のインストールでは、ほとんどの場合に、標準化の観点からデフォルト値を受け入れ、USERDEF GROUP をロックします。

CEDA VIEW コマンド

CEDA VIEW コマンドは、CSD 上のリソース定義の属性を表示する場合に使用します。

構文



説明

VIEW コマンドを使用すると、ALTER コマンドの場合と同様に、リソース定義属性を表示することができます。ただし、定義を更新することはできません。

オプション

Group(*groupname***)**
表示するグループを指定します。名前を指定しない場合、現行グループと見なされます。

例

```
VIEW TERMINAL(SZT1) GROUP(ZEMTERMS)

VIEW MAPSET(N20MAP01) GROUP(N20)
```

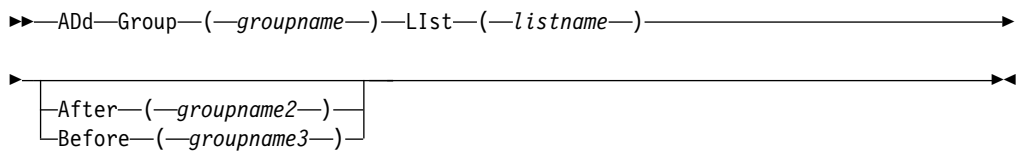
第 3 章 リソース管理ユーティリティー DFHCSDUP コマンド

このセクションでは、DFHCSDUP ユーティリティー・プログラムで使用可能なコマンドについて説明します。コマンドは省略することができますが、使用できる最小の省略形は、一部の CEDA コマンド省略形とは異なります。

DFHCSDUP ADD コマンド

グループをリストに追加します。

ADD の構文



オプション

After*(groupname2)*
既存のグループ名の後に新しいグループ名を入れるには、AFTER を指定します。 BEFORE か AFTER を指定しないと、グループ名はリストの末尾に追加されます。

Before*(groupname3)*
既存のグループ名の前に新しいグループ名を入れるには、BEFORE を指定します。 BEFORE か AFTER を指定しないと、グループ名はリストの末尾に追加されます。

Group*(groupname)*
追加するグループの名前を指定します。リスト内で既存の名前は指定できません。汎用グループ名は受け入れられません。

LIst*(listname)*
グループを追加するリストの名前を指定します。リストが既存でない場合は、新しいリストが作成されます。総称リスト名は受け入れられません。

例

グループを追加することにより、リスト LA01 を作成するには、以下のようにします。

```
ADD GROUP(GA001) LIST(LA01)
```

場所を指定せずに別のグループをリスト LA01 に追加するには、以下のようにします。

```
ADD GROUP(GA002) LIST(LA01)
```

この時点で、LA01 は以下のようになります。

© Copyright IBM Corp. 1982, 2018

- GA001
- GA002

リストの先頭に別のグループを追加するには、以下のようにします。

```
ADD GROUP(GA003) LIST(LA01) BEFORE(GA001)
```

GA001 と GA002 の間に別のグループを追加するには、以下のようにします。

```
ADD GROUP(GA004) LIST(LA01) AFTER(GA001)
```

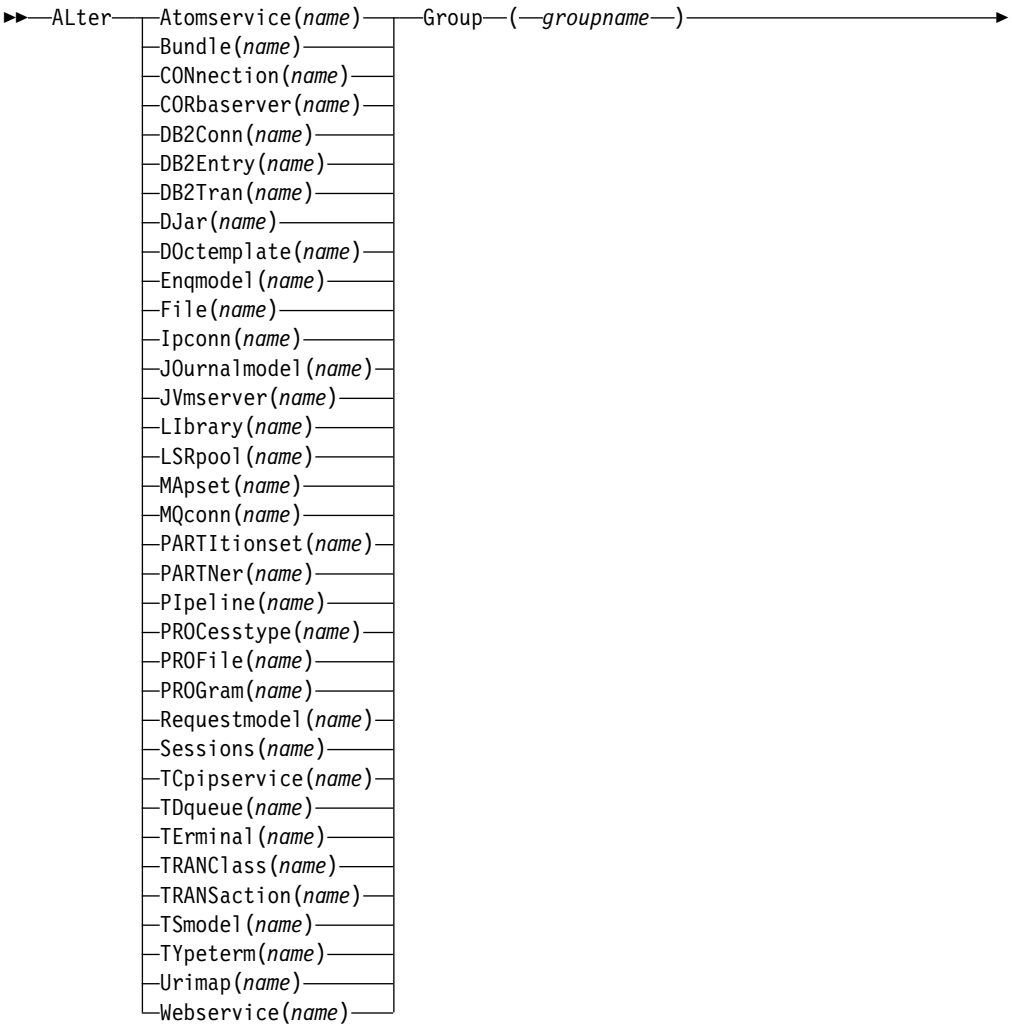
この時点で、LA01 は以下のようになります。

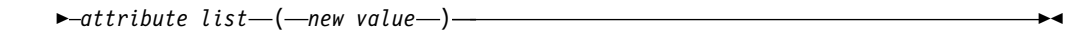
- GA003
- GA001
- GA004
- GA002

DFHCSDUP ALTER コマンド

既存リソース定義の属性の全部または一部を変更します。

ALTER 構文





説明

さまざまなリソース・タイプに対する ALTER コマンドで指定できる属性について、また各リソース・タイプの属性およびデフォルト値については、RDO resourcesを参照してください。

他の属性が依存している TYPETERM 定義の属性の値を変更する場合は、ALTER を使用しないでください。 DEVICE、SESSIONTYPE、または TERMMODEL に間違いがあった場合、その定義を削除してから、正しい値を使用して新規の定義を作成する必要があります。

ヌル属性値を指定することができます。例えば、次のように指定します。

```
ALTER FILE(TEST) GROUP(ACT1) DESCRIPTION()
```

ヌル値を指定した属性にデフォルト値がある場合、指定される値はフィールドのタイプによって決まります。

- 次のようにコマンドを発行したとします。

  ```
  ALTER FILE(TEST) GROUP(ACT1) RLSACCESS() DESCRIPTION()
  ```

  この場合、RLSACCCESS にはデフォルト値 NO が使用され、DESCRIPTION はブランクになります。
- 次のようにコマンドを発行したとします。

  ```
  ALTER FILE(TEST) GROUP(ACT1) PROFILE()
  ```

  この場合、PROFILE フィールドにはデフォルト値 DFHCICSA が使用されます。

CSD ファイルでリソース定義に対して行った変更は、そのリソース定義が属しているグループをインストールしない限り、稼働中の CICS システムでは有効になりません。

ALTER コマンドでの総称ネーミング

ALTER コマンドは、総称リソース名とグループ名の両方を受け入れます。

指定されたリソース名とグループ名の組み合わせに一致する CSD ファイル内の各リソースに対して、ALTER が試行されます。個々の ALTER が失敗する場合は、コマンドのすべての試行が処理された後に、処理が終了します。

オプション

属性リスト
変更する属性を指定します。

Group*(groupname)*
変更するリソースが含まれるグループを指定します。

Resource*(name)*
属性を変更する対象のリソースを指定します。文字 + および * を使用して、総称名を指定することができます。

例

プログラムを常駐するようにするには、次のようにします。

```
ALTER PROGRAM(ERR01) GROUP(GENMODS) RESIDENT(YES)
     DATALOCATION()
```

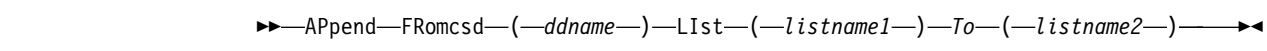
グループ GENMOD 内のすべてのプログラムを常駐するようにするには、次のようにします。

```
ALTER PROGRAM(*) GROUP(GENMOD) RESIDENT(YES)
    DATALOC()
```

DFHCSDUP APPEND コマンド

特定のリスト内のグループを別のリストの末尾に追加します。

APPEND 構文



説明

リスト内でのグループ名の重複は許可されません。DFHCSDUP が APPEND 操作中に重複名を検出した場合、その名前は無視され、追加されません。これが発生した場合、DFHCSDUP が出力するリストに警告メッセージが記載されます。

注: この追加を CSD 間で行う場合、このコマンドはグループ自体をコピーするものではないので注意してください。グループ自体をコピーするには、別に COPY コマンドを使用する必要があります。

オプション

FRomcsd*(ddname)*
listname1 を追加する元の 2 次 CSD ファイルの DD 名を指定します。

List*(listname1)*
追加されるリストの名前を指定します。総称リスト名は使用しないでください。

追加するリストは、1 次 CSD ファイルにあっても、別の CSD ファイルにあっても構いません。別の CSD ファイルから追加する場合は、FROMCSD パラメーターを使用してそれを指定する必要があります。

To*(listname2)*
グループ名を追加する先のリストの名前を指定します。別の CSD ファイルから追加する場合は、このリストに追加元と同じ名前を付ける必要があります。総称リスト名は使用しないでください。

このターゲット・リストが既に存在する場合、ソース・リストはそのターゲット・リストの末尾に追加されます。ターゲット・リストが存在しない場合は、そのリストが作成されます。(実際には、ソース・リストをコピーしています。)

例

LISTA という名前のリストには、以下のグループが含まれます。

- GB001
- GB002
- GB003

LISTB という名前のリストには、以下のグループが含まれます。

- G001
- G002
- G003

次のように、LISTB を LISTA に追加します。

```
APPEND LIST(LISTB) TO(LISTA)
```

これを実行した後、LISTA には、以下のグループがこの順序で含まれます。

- GB001
- GB002
- GB003
- G001
- G002
- G003

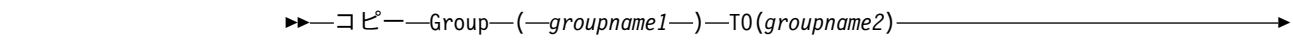
また、LISTB には引き続き以下が含まれます。

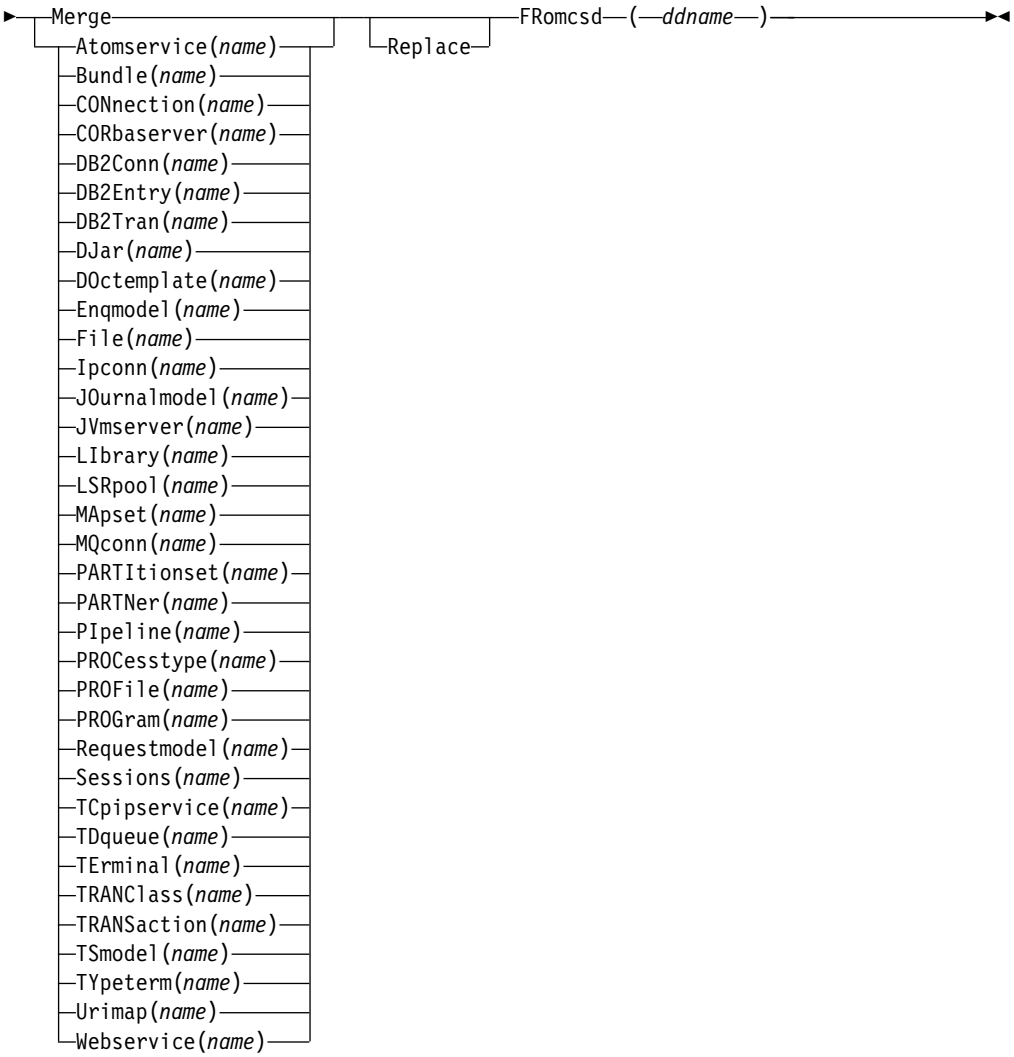
- G001
- G002
- G003

DFHCSDUP COPY コマンド

グループ間でリソース定義をコピーします。

COPY 構文





説明

COPY コマンドは、リソース定義を **groupname1** から **groupname2** にコピーします。リソース定義のコピー元となるグループ (*groupname1*) は、1 次 CSD 上、あるいは **FROMCSD** パラメーターで指定された CSD ファイル上にあるものが可能です。

DFHCSDUP COPY コマンドを使用して、個々のリソース定義をコピーすることも、グループ内のすべてのリソース定義をコピーすることもできます。リソース定義は、1 次ファイルの TO パラメーターで指定されたグループ (*groupname2*) にコピーされます。このグループが既に存在する場合、ソース・グループ (*groupname1*) からの定義は、*groupname2* グループに既に存在する定義に追加されます。TO パラメーターで指定されたグループが存在しない場合は、その名前の新規グループが作成されます。

DFHCSDUP COPY コマンドを使用してグループ内のすべてのリソース定義を別のグループにコピーする際に、そのグループに重複するリソース名が含まれている場合、MERGE または REPLACE を指定して、重複をどのように処理するかを指示する必要があります。 MERGE を指定すると、TO グループ内の重複する定義は置換されません。 REPLACE を指定すると、コピーされた定義によって TO グループ

内にあった定義が置換されます。 MERGE および REPLACE のいずれも指定しない場合、重複が検出されるとコピー操作は失敗します。

DFHCSDUP COPY コマンドを使用して個々のリソース定義を別のグループにコピーする場合、デフォルトでは、TO グループ内に重複する定義が存在していても置換されません。 REPLACE を指定すると、コピーされた定義によって TO グループ内にあった定義が置換されます。

DFHCSDUP の出力リストには、どの定義がコピーされたか、および (重複が検出された場合は) 何が行われたかが示されます。

COPY コマンドでの総称ネーミング

DFHCSDUP COPY コマンドは、以下の規則を条件として、GROUP オプションおよび TO オプションでの総称グループ名を許容します。

- COPY コマンドで許可される汎用文字はアスタリスク (*) 記号のみで、名前の末尾になければなりません。
- *groupname1* の接頭部の長さは、*groupname2* の接頭部の長さ以上でなければなりません。COPY GROUP(DFHCOMP*) TO(USRCMP*) は有効ですが、COPY GROUP(DFHCO*) は無効です。

378 ページの『例』に示すように、アスタリスク (*) 記号を使用して、総称的な名前のグループから他の総称的な名前のグループに、あるいは総称的な名前のグループから特定のグループにコピーすることができます。

DFHCSDUP COPY コマンドは、リソース・オプションのいずれかを使用して指定する個々のリソース定義での総称名を許容しません。

オプション

FRomcsd*(ddname)*
: グループ *groupname1* をコピーする元の 2 次 CSD ファイルの DD 名を指定します。

Group*(groupname1)*
: コピーするグループの名前を指定します。アスタリスク (*) を使用することで、グループの総称名を指定することができます。デフォルト・アクションでは、グループ全体がコピーされます。あるいは、リソース・タイプ・オプションのいずれかと、コピーするそのタイプの個々のリソース定義の名前を指定することもできます。

MErge
: *groupname2* が既に存在しており、重複する定義が発生した場合に、*groupname2* 内の元の定義を保持することを指定します。

置換
: *groupname2* が既に存在しており、重複する定義が発生した場合に、*groupname1* の定義で *groupname2* の定義を置換することを指定します。

resource(*name*)
: 属性をコピーする対象の個々のリソース定義のタイプおよび名前を指定します。DFHCSDUP COPY コマンドでは、個々のリソース定義に総称名を使用することはできません。

To(*groupname2*)
すべてのリソース定義あるいは個々に指定されたリソース定義をコピーする先のグループの名前を指定します。別の CSD ファイルからコピーする場合、このグループにコピー元と同じ名前を付けることができます。アスタリスク (*) を使用することで、グループの総称名を指定することができます。

例

以下の例では、GA001 という名前のグループを GA002 という名前のグループにコピーしています。GA002 は既に存在しており、重複するリソース定義はグループ GA001 の定義に置換されます。

```
COPY GROUP(GA001) TO(GA002) REPLACE
```

以下の例では、グループ GA003 をグループ GA004 にコピーしていますが、重複する定義が発生した場合は、グループ GA004 の定義を保持します。

```
COPY GROUP(GA003) TO(GA004) MERGE
```

以下の例では、CICS が提供するすべてのグループを、接頭部が USR であるユーザー指定のグループにコピーします。その結果、DFHOPER が USROPER、DFHSTAND が USRSTAND のようになります。

```
COPY GROUP(DFH*) TO(USR*)
```

以下の例では、名前が ABCD から始まるすべてのグループを NEWGROUP という名前のグループにコピーします。

```
COPY GROUP(ABCD*) TO(NEWGROUP)
```

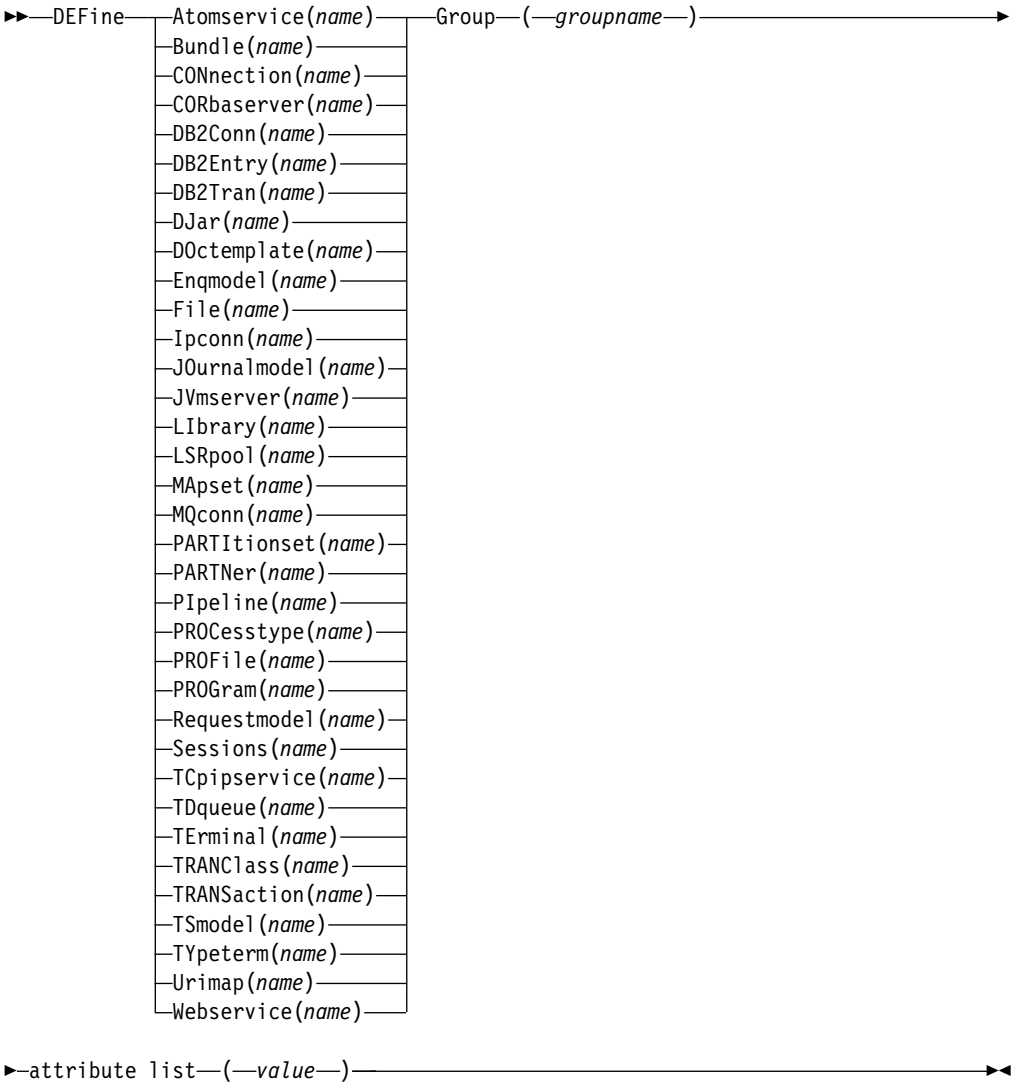
以下の例では、単一のリソース定義を CICSC1 グループから CICSC2 グループにコピーします。

```
COPY SESSIONS(L122) GROUP(CICSC1) TO(CICSC2)
```

DFHCSDUP DEFINE コマンド

新規リソース定義を作成します。

DEFINE 構文



オプション

属性リスト

属性リストは、定義するリソース・タイプに依存します。一部のリソースでは、定義中に必ず含めなければならない属性があります。指定しない属性については、デフォルト値が使用されます。

Group*(groupname)*

作成されるリソース定義を入れるグループの名前を指定します。総称グループ名は使用しないでください。存在しないグループ名を指定した場合、そのグループが作成されます。

Resource*(name)*

定義するリソースの名前を指定します。総称リソース名は使用しないでください。リソース・オプションは、必ず **DEFINE** コマンドの第 1 オペランドにする必要があります。

例

1 つのグループに含まれる複数のリソース定義に対しては、それらの定義が異なるリソース・タイプに対するものであるなら同じ名前を使用できます。以下に例を挙げます。

```
DEFINE PROGRAM(N28A) GROUP(N28APPL)
DEFINE TRANSACTION(N28A) GROUP(N28APPL)

DEFINE TERMINAL(USER) GROUP(USERDEF)
DEFINE PROGRAM(USER) GROUP(USERDEF)
```

次の例では、2 つのコンソールを CICS に定義しています。定義が複数行にまたがる場合も、継続記号は不要です。

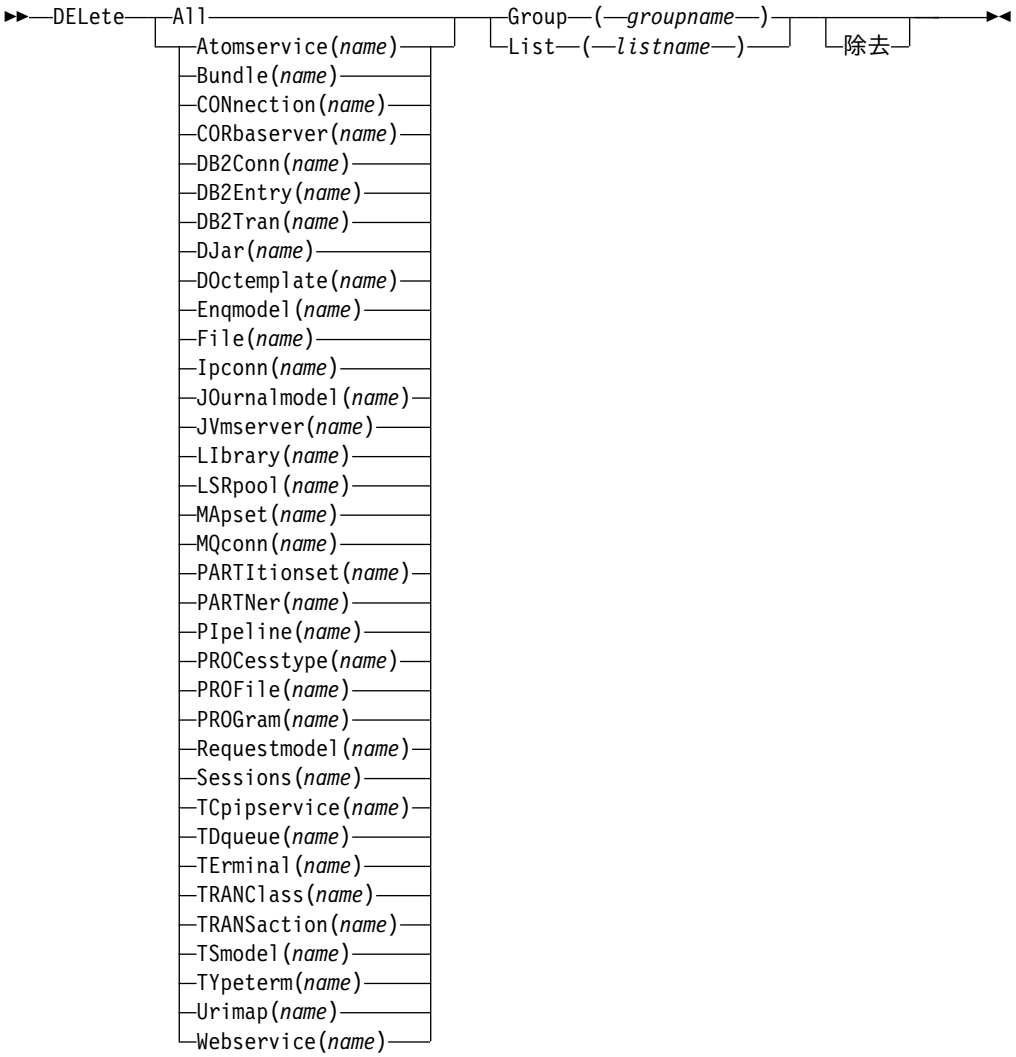
```
DEFINE TERMINAL(CON0)     GROUP(CONTERMS)
       CONSNAME(CONSJCL)  TYPETERM(DFHCONS)
       DESCRIPTION(MVS CONSOLE FOR ISSUING JCL COMMANDS)

DEFINE TERMINAL(CON1)     GROUP(CONTERMS)
       CONSNAME(CONSMAS)  TYPETERM(DFHCONS)
       DESCRIPTION(MVS MASTER CONSOLE)
```

INITIALIZE コマンドは、コンソール用に TYPETERM 定義を生成しますが、TERMINAL 定義は生成しません。 CICS に MVS **MODIFY** コマンドを発行するには、少なくとも 1 つのコンソールが定義されている必要があります。

DFHCSDUP DELETE コマンド

グループ内の単一のリソース定義、グループ内のすべてのリソース定義、あるいはグループ・リスト内のすべてのグループ名を削除します。

DELETE 構文



説明

リソース定義の削除は、リストからのグループの除去とは異なります (388 ページの『DFHCSDUP REMOVE コマンド』を参照)。削除されたリソース定義は、実際に CSD ファイルから消去されます。

注:

グループ内の最後のリソースを削除すると、そのグループは自動的に削除されます。空のグループは、存在することができません。

グループが削除されても、そのグループが含まれるすべてのリストから除去されるわけではありません。

IBM が提供するグループおよびリストの定義を削除することはできません。

リストを削除しても、リストに含まれていたグループ内のリソースの定義は削除されません。これを行うには、各グループも個別に削除する必要があります。

オプション

Group*(groupname)*
これを単独で指定した場合、削除するグループの名前を示します。リソースも指定した場合、リソースが属しているグループを示します。総称グループ名は使用しないでください。

List*(listname)*
削除するリストの名前を指定します。総称リスト名は使用しないでください。

除去
グループを削除する際にこれを指定すると、UPGRADE コマンドが実行されていない限り、そのグループが含まれるすべてのリストからグループが除去されます。

Resource*(name)*
削除するリソースの名前を指定します。総称リソース名は使用しないでください。

このオペランドは、GROUP オプションと一緒に使用する必要があります。

例

LISTA という名前の 1 次 CSD ファイルには、以下のグループが含まれます。

- GB001
- GB002

グループ GB001 には、以下のリソース定義が含まれます。

TERMINAL(CON0)
TERMINAL(CON1)
TERMINAL(TEST)

次のコマンドは、グループ GB001 から端末 TEST のリソース定義を削除します。

```
DELETE TERMINAL(TEST) GROUP(GB001)
```

次のコマンドは、グループ GB002 からすべてのリソース定義を削除します。

```
DELETE GROUP(GB002)
```

これにより、グループ・リスト LISTA には、グループ GB001 のみが残ります。次のコマンドは、グループ・リスト LISTA 内のすべてのグループ名を削除します。

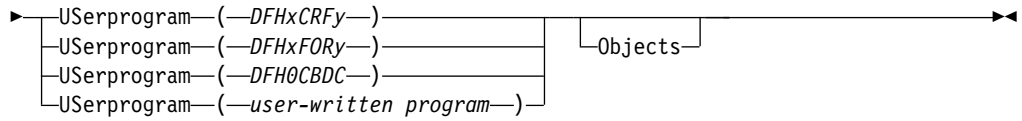
```
DELETE LIST(LISTA)
```

注: LISTA 内のグループのリソース定義は削除されません。

DFHCSDUP EXTRACT コマンド

CSD ファイルからリソース定義、グループ、またはリストを抽出します。

EXTRACT 構文



説明

EXTRACT コマンドを使用して、CSD ファイルから (リストあるいはグループのどちらからでも) リソース定義データを抽出することができます。また、ユーザー・プログラムを呼び出して、抽出したデータを処理することができます。ユーザー・プログラムは、**USERPROGRAM** パラメーターで指定します。

注: EXTRACT コマンドのユーザー・プログラムのコーディングに関するプログラミング情報については、DFHCSDUP からのユーザー・プログラムの呼び出しを参照してください。

オプション

Group*(groupname)*
指定されたグループ内のリソース定義のみを指定します。汎用グループ名を指定することができます。

LIst*(listname)*
指定されたリストに含まれるグループ内のリソース定義のみを指定します。OBJECTS オプションを使用していない場合のみ、汎用リスト名を使用できます。

Objects
各リソース定義の詳細を返します。以下の 2 つの詳細レベルで、リソース定義データを抽出できます。

- OBJECTS オプションを指定しない場合、コマンドは指定されたリスト内のすべてのグループの名前、あるいは指定されたグループ内のすべてのリソース定義の名前を抽出します。
- OBJECTS を指定した場合、すべてのリソース定義属性も抽出されます。

提供されているサンプル・ユーザー・プログラム DFHxCRFy および DFHxFORy の場合は、OBJECTS を指定する必要があります。 DFH0CBDC およびユーザーが作成したユーザー・プログラムの場合、これはオプションです。

USerprogram*(user-written program)*
EXTRACT コマンドによって取得されたデータを処理するためのユーザー作成プログラムの名前を指定します。USERPROGRAM 値を指定する必要があります。

CICS は、3 つのタイプのサンプル・ユーザー・プログラム (DFHxCRFy、DFHxFORy、および DFH0CBDC) を提供しています。プログラム名の文字 *x* は、アセンブラーまたは PL/I の場合は $、COBOL の場合は 0 になります。プログラム名の文字 *y* は、プログラミング言語を示します。ここで、y=A がアセンブラー・バージョン、y=C は COBOL バージョン、y=P は PL/I バージョンです。

それぞれのユーザー作成プログラムは、特定の DD カードを使用する必要があります。

- DFHxCRFy では、DD カード CRFOUT が必要

- DFHxFORy では、DD カード FOROUT が必要
- DFH0CBDC では、DD カード CBDOUT が必要

その他のユーザー・プログラムは、CICSTS55.CICS.SDFHSAMP のソース形式で使用でき、アセンブラー・バージョンも CICSTS55.CICS.SDFHLOAD で事前生成された形式で使用できます。

例

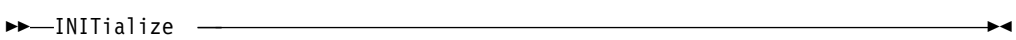
次のコマンドは、提供されたユーザー・プログラム DFH0CBDC を使用してグループ DFHTYPE 内のリソース定義を抽出し、それらを作成するために必要な **DEFINE** コマンドを作成します。これらのコマンドは、CBDOUT DD ステートメントで指定されたファイルに保管されます。

```
EXTRACT GROUP(DFHTYPE) USERPROGRAM(DFH0CBDC) OBJECTS
```

DFHCSDUP INITIALIZE コマンド

新規に定義されたデータ・セットを CSD ファイルとして使用するために準備します。

INITIALIZE 構文



説明

他の DFHCSDUP コマンドや RDO トランザクションを使用できるようにするには、事前に CSD ファイルを初期設定する必要があります。CSD ファイルを初期設定した後は、この関数を再度実行する必要はありません。

CICS が提供するリソース定義の標準エントリーは、CSD ファイルに作成されます。INITIALIZE コマンドは、これらの定義をグループに配置し、そのグループを DFHLIST という名前のグループ・リストに定義します。このリストには、CICS システムで必要な、CICS が提供するグループのみが含まれます。

CICS は、一時データの RDO をサポートします。DFHDCTG グループには、CICS が提供するすべてのキューのサンプル定義が含まれます。同時にインストールしたい他のキューの名前を DFHDCTG に追加することができます。 DFHDCTG を DFHLIST の先頭に配置することで、CICS の初期化中のできるだけ早い時点で、そのキューを使用できるようになります。

別のグループを使用して CICS が提供するキューをインストールする場合は、初期始動またはコールド・スタートの一環として GRPLIST を使用してインストールされる最初のリストの先頭にそのグループを配置してください。

他の一時データ・リソース定義を別のグループに入れることができます。このグループから、初期始動またはコールド・スタート中、あるいは初期設定が完了した後の任意の時点でそのリソース定義をインストールすることができます。

また、INITIALIZE は、CSD ファイルの先頭に制御レコードを作成します。このレコードには、CICS リリースを識別するフィールド、および CSD に適用されたサー

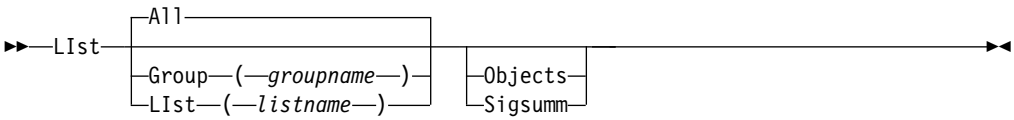
ビスの現行レベルを識別するフィールドが含まれます。また、CSD ファイルの作成日時が含まれるフィールド、およびファイルが最後に更新された日時が含まれるフィールドもあります。これらのフィールドは、どちらも LIST コマンドで生成される CSD ファイルのハードコピー・リストに表示されます。

新規に定義されたリカバリー可能データ・セットを CSD ファイルとして使用するために準備する場合は、RLS 以外のモードを使用してそのデータ・セットを初期設定する必要があります。これは、リカバリー可能データ・セットは、RLS モードではバッチからの出力として開くことができませんが、初期設定するために出力で開く必要があるためです。

DFHCSDUP LIST コマンド

CSD ファイルの現在の状況のリストを作成します。

LIST の構文



説明

リストが SYSOUT データ・セットに送信され、コマンド処理によってメッセージが発行されます。すべての修飾グループまたはリストの内容が印刷されます。

このリストには、CSD のリリース情報を確認するのに役立つ出力が含まれます。見出し *CREATED BY RELEASE:* は、CSD が作成されたリリースを示します。これが存在している場合、CSD がアップグレードされていれば、さらにその後に *UPGRADED TO RELEASE:* と対象のリリースが続きます。

CSD に保守が適用された場合は、追加の見出し *LAST CSD MAINTENANCE UPGRADE USED filename AT PTF ptfnumber LEVEL* が表示されます。例えば、*DFHCSDUP UPGRADE USING (DFHCURDM)* コマンドを使用して適用された保守は、使用されている *DFHCURDM* の PTF レベルを示します。

オプション

Group*(groupname)*
名前指定されたグループ内のリソース定義のみを指定します。汎用グループ名を指定することができます。

LIst*(listname)*
名前指定されたリストに含まれるグループ中のリソース定義のみを指定します。OBJECTS オプションを使用していない場合のみ、汎用リスト名を使用できます。総称リスト名が受け入れられない唯一のコマンドは、LIST LIST(*listname*) OBJECTS です。

Objects
リソース定義ごとに必要な詳細情報のレベルを指定します。以下の 2 つの詳細レベルで、リソース定義データを抽出できます。

- OBJECTS オプションを指定しないと、コマンドは、指定されたリスト内のすべてのグループの名前か、指定されたグループ内のすべてのリソース定義の名前を抽出します。
- OBJECTS オプションを指定すると、定義シグニチャー・フィールドを含む、すべてのリソース定義属性も抽出されます。

Sigsumm
表示された各リソース定義の定義シグニチャーを示します。

例

さまざまなコマンドによって作成されるリストは以下のとおりです。

- LIST ALL
  - 定義されたリストとグループの名前
  - リストの要約
  - グループの要約

  LIST ALL コマンドは、CSD ファイル上のリストとグループのすべての定義の要約を印刷します。
- LIST ALL OBJECTS
  - 定義されたリストとグループの名前
  - リストの要約
  - グループの要約
  - グループ内のオブジェクト

  LIST ALL OBJECTS コマンドは、CSD ファイル上のリストとグループのすべての定義の要約、およびすべてのグループ中のリソースのプロパティーを印刷します。
- LIST GROUP(*groupname*)
  - グループの要約
  - 汎用グループ名を使用可能

  LIST GROUP コマンドは、1 つ以上のグループ中のすべてのリソースの名前を要約します。グループごとに、マップ・セットやプログラムなどのリソース・タイプのカテゴリーに編成されます。
- LIST GROUP(*groupname*) OBJECTS
  - グループの要約
  - グループ内のオブジェクト
  - 汎用グループ名を使用可能

  このコマンドを使用すると、リソース・タイプに従って編成された、リソースのプロパティーを作表できます。各リソースのすべての属性とともに、その作成時刻が提供されますが、これは、DEFINE および ALTER コマンドを使用して、あるいは CICS テーブルからマイグレーションすることによって、最初にセットアップした時のものです。トランザクションとプロファイルのプロパティーは、CEDA DEFINE 画面に表示されるサブカテゴリーで整列されます。
- LIST GROUP(*groupname*) SIGSUMM

– 汎用グループ名を使用可能

このコマンドを使用して、リソース・タイプに従って編成された、リソースの定義シグニチャーを作表します。

- LIST LIST(*listname*)
  – リストの要約
  – 総称リスト名を使用可能

  1 つ以上のグループ・リストの内容が作表されます。グループは、リスト内の位置と同じ順序で表示されます。この順序はコマンド ADD および APPEND によって設定されます。これらのコマンドは CEDA トランザクションでリストを構築するのに使用されたものです。
- LIST LIST(*listname*) OBJECTS
  – リストの要約
  – リスト内のグループのオブジェクト
  – 汎用リスト名は使用できない

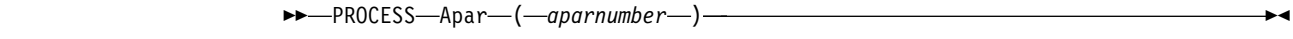
  起動時に CICS システム内で定義されるすべてのリソースのプロパティーを作表するには、このコマンドを使用します。これらは、GRPLIST=(*list1,list2,list3,list4*) システム初期設定パラメーターで指定された 1 つ以上のリスト名によって識別されます。リスト内のすべてのグループの名前が、リストの要約に示されます。次に、リストに含まれる各グループについて、グループ内の個々のリソースのプロパティーが表形式で示されます。

  「リスト内のグループのオブジェクト」タブには、グループ・リストに追加された順序でグループが配置されています。この順序は、複数のグループに同じリソースの定義があり、重複が発生する場合に重要です。システム起動時にこのタイプのリストを使用する場合、重複があったときに使用されるリソース定義は、リスト内の最後のグループに属するものとなります。

DFHCSDUP PROCESS コマンド

特定の APAR について CSD ファイルに保守を適用します。

PROCESS 構文



説明

PROCESS APAR コマンドは、特定の APAR について CSD ファイルに保守を適用する場合に試用します。このコマンドは、関連付けられた PTF カバー・レターで指示された場合にのみ使用します。

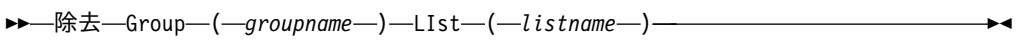
オプション

Apar*(aparnumber)*
保守を提供する APAR の番号。例えば、APAR PQ12417 に保守を適用する場合は、PROCESS APAR(PQ12417) を使用します。

DFHCSDUP REMOVE コマンド

リストからグループ名を除去します。

REMOVE 構文



説明

グループおよびそのすべてのリソース定義は、CSD ファイルに残っています。

オプション

Group*(groupname)*
除去するグループの名前を指定します。総称グループ名は使用しないでください。

LIst*(listname)*
グループを除去する対象のリストの名前を指定します。総称リスト名は使用しないでください。リストから最後のグループが除去されると、そのリストは CSD ファイル上に存在しなくなります。

例

リスト LL02 には、以下のグループが含まれます。

G001 G002 G003 G004

グループ G003 を除去するには、次のようにします。

```
REMOVE GROUP(G003) LIST(LL02)
```

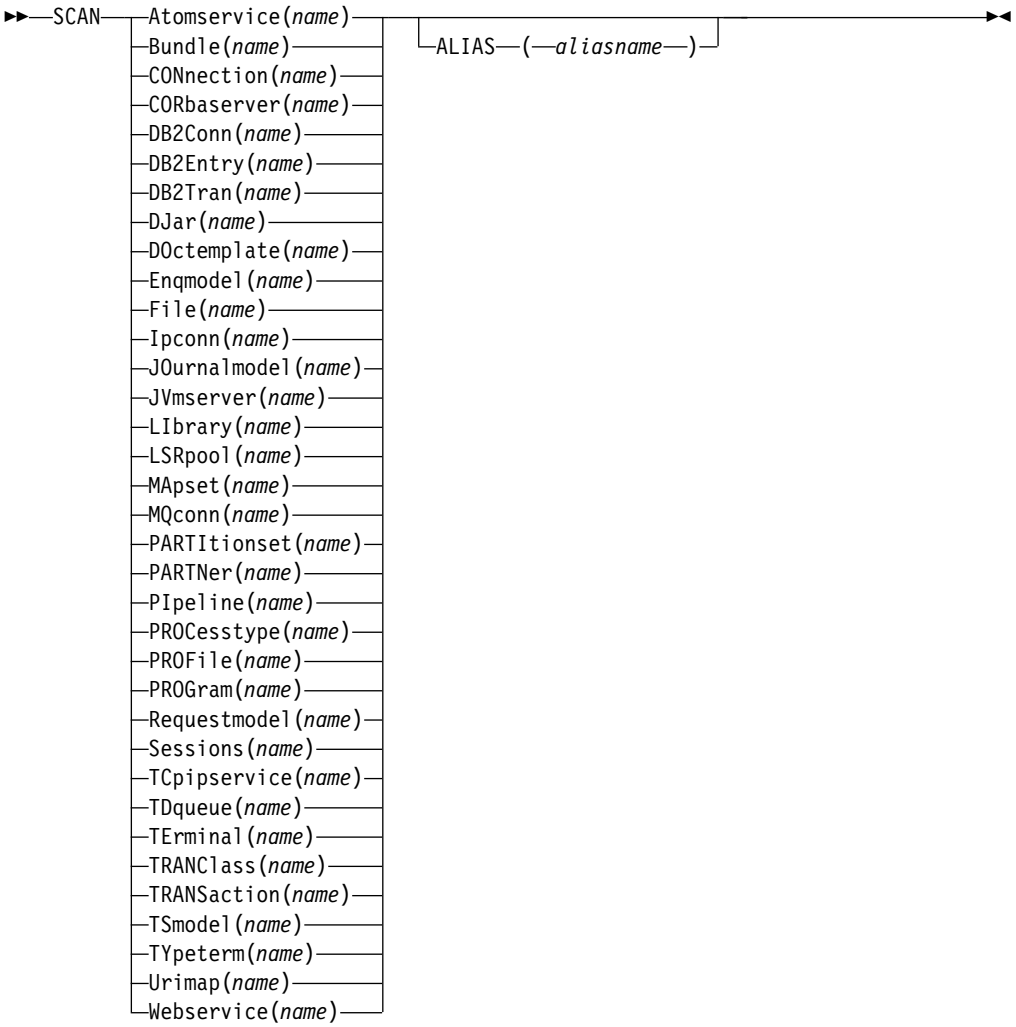
これにより、以下が残ります。

G001 G002 G004

DFHCSDUP SCAN コマンド

IBM 提供のグループおよびユーザー定義グループをすべて SCAN して、指定されたリソースを検索します。 IBM 提供のグループ内の一致したリソースの定義は、ユーザー・グループ内のそれに対応する一致したリソースの定義と比較されます。

SCAN 構文



説明

SCAN コマンドで指定できるリソースのタイプについて、また各リソース・タイプの属性およびデフォルト値について詳しくは、「RDO resources」を参照してください。

SCAN コマンドは、CSD 内のすべての IBM 提供のグループを調べて、指定した名前およびタイプのリソース定義がないかどうか検索します。検索の結果とともにメッセージが発行されます。次に、同じリソース定義がないかどうか、ユーザー定義のグループの検索が行われます。この結果は以下のいずれかになります。

- IBM 提供のグループおよび 1 つ以上のユーザー定義のグループにリソース定義が含まれる場合、IBM 提供のグループの定義とユーザー・グループの定義との比較が行われます。 IBM 提供のグループの定義がユーザー・グループの定義と一致しているかどうかを示すメッセージが発行されます。
- リソース定義がユーザー定義のグループで見つからなかった場合、メッセージが発行されます。
- リソース定義が IBM 提供のグループで見つからず、1 つ以上のユーザー定義のグループで見つかった場合、それが含まれているグループを示すメッセージが発行されます。

aliasname が指定されている場合、ユーザー・グループは *aliasname* を使用して検索されます。

注:

1. 互換性グループ DFHCOMPx は、IBM 提供のグループではなく、ユーザー定義のグループの一部としてスキャンされます。
2. DESCRIPTION 属性は、比較では使用されません。

SCAN コマンドを使用して、変更した IBM 提供の定義と、アップグレード後の最新の IBM 提供のバージョンとの差異をチェックすることができます。

オプション

Alias*(aliasname)*
ユーザー定義のグループで検索するリソース・タイプの別名を指定します。

このオペランドはオプションです。

Resource*(name)*
aliasname が指定されていない場合、IBM 提供のグループおよびユーザー定義のグループで検索するリソース・タイプの名前を指定します。リソース・オプションは常に SCAN コマンドの第 1 オペランドでなければなりません。

例

CSD を調べてトランザクション CEDA がないかどうか検索するには、以下のようにします。

```
SCAN TRANSACTION(CEDA)
```

この結果は以下のようになります。

```
 DFH5130 I PRIMARY CSD OPENED;  DDNAME: DFHCSD
 DFH5633 I TRANSACTION CEDA FOUND IN GROUP DFHSPI
 DFH5631 I TRANSACTION CEDA FOUND IN GROUP A1
           MATCHES THE IBM SUPPLIED DEFINITION IN GROUP DFHSPI
 DFH5631 I TRANSACTION CEDA FOUND IN GROUP A2
           MATCHES THE IBM SUPPLIED DEFINITION IN GROUP DFHSPI
 DFH5632 I TRANSACTION CEDA FOUND IN GROUP DFHCOMP1
           DOES NOT MATCH THE IBM SUPPLIED DEFINITION
           IN GROUP DFHSPI
 DFH5101 I SCAN COMMAND EXECUTED SUCCESSFULLY.
```

AEDA という別名を持つトランザクション CEDA がないかどうか CSD を検索するには、以下のようにします。

```
SCAN TRANSACTION(CEDA) ALIAS(AEDA)
```

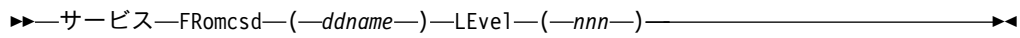
この結果は以下のようになります。

```
 DFH5130 I PRIMARY CSD OPENED;  DDNAME: DFHCSD
 DFH5633 I TRANSACTION CEDA FOUND IN GROUP DFHSPI
 DFH5631 I TRANSACTION AEDA FOUND IN GROUP A3
           MATCHES THE IBM SUPPLIED DEFINITION IN GROUP DFHSPI
 DFH5101 I SCAN COMMAND EXECUTED SUCCESSFULLY.
```

DFHCSDUP SERVICE コマンド

CSD ファイルに対する保守を実行します。

SERVICE 構文



説明

場合により (CICS のリリース間で) サービス・ルーチンを適用して、CSD ファイルに対する予防保守あるいは修理保守を実行することが必要になる可能性があります。

SERVICE コマンドを使用して、既存の CSD ファイルから CSD ファイルの新規コピーを作成することができます。すべての定義が保持され、修正が適用されます (ある場合)。

オプション

FRomcsd*(ddname)*
現行の CSD ファイルの DD 名を指定します。これは、コマンドのために使用され、2 次 CSD ファイルとして扱われます。

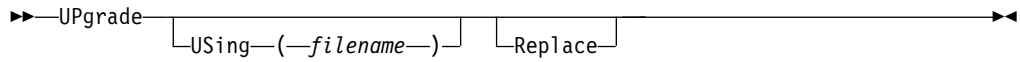
LEvel*(nnn)*
CSD ファイルに関連付けられているのは、現行のサービス・レベルであり、ファイルの初期設定時には 000 に設定されています。サービス・ルーチンを適用すると、サービス・レベルは、「現行レベル」から「ターゲット・レベル」まで 1 ずつ増分します。

このオペランドは、CSD ファイルをアップグレードするターゲット・サービスを指定し、FROMCSD の現行レベルより 1 大きい値でなければなりません。これは、3 文字の整数として指定します (例えば、LEVEL(001))。

DFHCSDUP UPGRADE コマンド

1 次 CSD ファイル内の CICS 提供のリソース定義を変更します。

UPGRADE 構文



説明

CSD 内の IBM 提供の定義をアップグレードします。 DFH グループにおいて、定義の追加、変更、または削除が行われます。削除された定義は、形式 DFHCOMP*n* の名前で互換性グループに追加されることに注意してください。これにより、アップグレード・コマンドの実行後に、以前のリリースの CICS と CSD を共用できるようになります。

DFHCSDUP LIST コマンドを使用して、アップグレード前後の CSD ファイルの状況を確認します。このコマンドは、CSD が作成されたリリースとアップグレード先のリリースを報告します。

アップグレード・コマンドを使用して、IBM 提供のリソース定義の任意のパッケージを CSD ファイルに適用することもできます。例えば、CICS サンプル・プログラムおよびトランザクションの定義は、UPGRADE ステートメントを使用して CSD ファイルに転送することができます。

オプション

置換

UPGRADE コマンドを再実行する必要がある (例えば、前回に問題が発生したために) 場合、REPLACE オプションを指定します。

USing(*filename*)

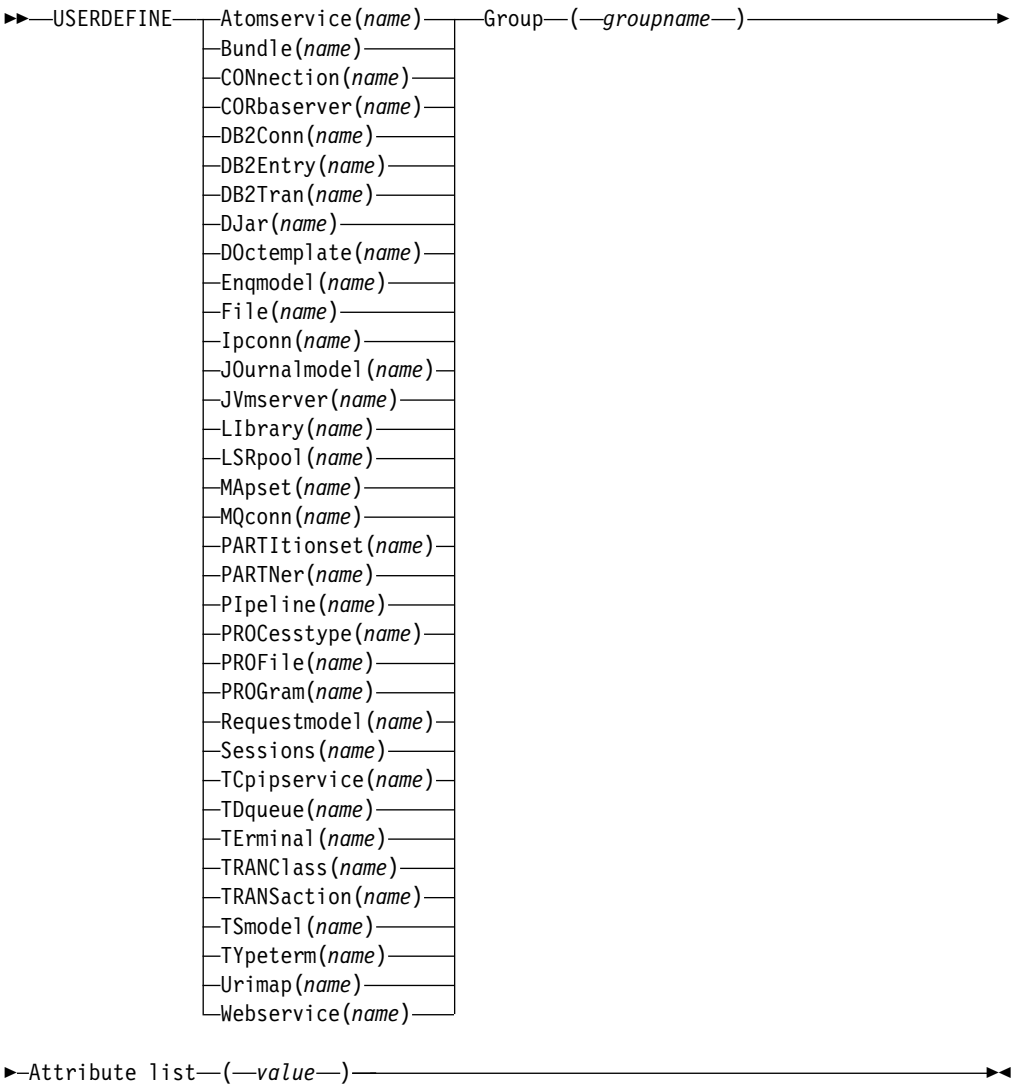
IBM 機能を CICS にインストールするには、UPGRADE USING(*filename*) を指定します。例えば、UPGRADE USING(DFHRDJPN) は、2 バイト文字セットのフィーチャー定義を CSD ファイルに配置するのに使用されます。

CSD をアップグレードする場合は、**USING** オペランドを使用しないでください。どのリリースにおいても IBM 提供の定義がすべて削除されてから、CSD ファイルが初期化されるため、ユーザーはどのリリースを所有しているかを通知する必要はありません。

DFHCSDUP USERDEFINE コマンド

CICS が提供するデフォルト値の代わりに、独自のデフォルト値を使用して新規のリソース定義を作成します。

USERDEFINE 構文



説明

USERDEFINE コマンドは、DEFINE コマンドの代替です。 USERDEFINE コマンドは、CICS が提供するデフォルト値を使用する代わりに、ユーザー独自のデフォルト値を使用してリソース定義を作成します。それ以外については、DEFINE コマンドを完全に同じです。

USERDEFINE コマンド用の独自のデフォルト値をセットアップするには、通常の DEFINE コマンドを使用して、USERDEF という名前のグループ内に USER という名前のリソース定義を作成します。

- デフォルト値を指定したい各リソース用に、USERDEF グループ内に USER という名前のリソース定義を作成します。例えば、PROGRAM、TRANSACTION、および TCPIPSERVICE の各リソース定義のデフォルト値を指定したい場合、USERDEF グループ内にリソース定義 PROGRAM(USER)、TRANSACTION(USER)、および TCPIPSERVICE(USER) を作成します。 USERDEF グループ内のすべてのリソース定義の名前を USER にすることは問題ありません。それらは異なるリソース・タイプを持つため、そ

れぞれが固有になります。 USER という名前が付いていない USERDEF グループ内のリソース定義は、USERDEFINE コマンドでは無視されます。

- USERDEF グループ内の各リソース定義で、USERDEFINE コマンドを使用してそのタイプのリソースを作成する場合に適用するデフォルト値を指定します。例えば、USERDEFINE コマンドを使用して作成する PROGRAM リソース定義で Assembler をデフォルトの言語にしたい場合、次の DEFINE コマンドを発行してリソース定義を作成します。

  ```
  DEFINE PROGRAM(USER) GROUP(USERDEF) LANGUAGE(ASSEMBLER)
  ```

- USERDEF グループ内の各リソース定義は、完全かつ有効なリソース定義でなければなりません。例えば、トランザクション定義では、USERDEFINE コマンドを使用してトランザクションを定義する場合に毎回プログラム名を指定する場合でも、プログラム定義が指定されている必要があります。
- USERDEF グループ内のリソース定義をインストールする必要はありません。

USERDEF グループ内にリソース定義が作成されていれば、USERDEFINE コマンドを使用してそのタイプのリソースを定義することができ、セットアップしたデフォルト値がそのリソース定義に使用されます。例えば、USERDEF グループ内に LANGUAGE(ASSEMBLER) を指定する PROGRAM リソース定義を作成した場合、次のコマンドを実行すると、グループ GRP にプログラム P2 のリソース定義が作成され、言語として Assembler が指定されます。

```
USERDEFINE PROGRAM(P2) GROUP(GRP)
```

オプション

Attribute list*(value)*
属性リストは、定義するリソース・タイプによって異なります。一部のリソースには、定義に含める必要がある属性があります。各リソース・タイプの属性およびデフォルト値の説明については、RDO resources を参照してください。指定しない属性については、デフォルト値が使用されます。

Group*(groupname)*
作成されるリソース定義を入れるグループの名前を指定します。総称グループ名は使用しないでください。まだ存在していないグループの名前を指定した場合、そのグループが作成されます。

Resource(name)
定義するリソースの名前を指定します。総称リソース名は使用しないでください。リソース・オプションは、必ず USERDEFINE コマンドの第 1 オペランドにする必要があります。

DFHCSDUP VERIFY コマンド

グループおよびリストの内部ロックを除去します。

VERIFY 構文

説明

VERIFY コマンドは、CSD ファイルが使用されておらず、CSD ファイルで保留中のバックアウト処理がない場合にのみ使用します。可能であれば、CSD ファイルを使用する可能性がある CICS システムが稼働していない場合にのみ使用してください。特に、CICS システムが RLS アクセス・モードで CSD ファイルにアクセスしている可能性がある場合は、VERIFY コマンドを使用しないでください。

VERIFY は CSD ファイル全体に作用し、背後に内部ロック・レコードが残っている極端な状態に使用するためのものです。これらのレコードは、通常は、CSD ファイルを変更する機能が完了したときに除去されます。しかし、CEDA トランザクションの実行中にシステム障害が発生した場合や、オフライン・ユーティリティーが正常に完了しなかった場合に、それが行われなかった可能性があります。ロックは、CEDA ユーザーが CSD ファイル上の特定のグループおよびリストにアクセスすることを防止します。

VERIFY は、内部ロックのみを除去するので注意してください。 CEDA トランザクション内で LOCK コマンドによって適用された通常のユーザー・ロックには影響しません。

第 4 章 DFHCSDUP: 構文規則およびコマンド規則

DFHCSDUP プログラム用のコマンドの構文と作成では、以下の規則を使用します。

80 文字の入力レコードの 1 桁目から 71 桁目までにコマンドを入力します。行末 (72 桁目) で継続文字 (アスタリスク) を使用すると、1 行より長いキーワード値を指定できます。後続の行は 1 桁目から始まります。例えば、この機能を使用して、最大 128 の 16 進文字の XTPNAME 値を指定できます。

コマンド・キーワードを指定する際には、各コマンドの説明の下にあるコマンド構文に示されているように、省略形および大/小文字混合で指定できます。コマンド構文では、最小限の省略形は大文字で示され、オプションの文字は小文字で示されています。例えば、次のようになります。

```
ALter Connection(name) Group(groupname)
```

先行ブランクは無視され、キーワードとオペランドの間にブランクを使用できます。

コメント・レコードを使用できます。これらのレコードでは、1 桁目にアスタリスク (*) が入力される必要があります。コマンドを含むレコードでは、コメント要素は使用できません。

コマンド間のブランク・レコードは無視されます

GROUP、**LIST**、**TO**、および **TYPESGROUP** の各パラメーターをコーディングする場合は、グループおよびリストの名前に関する規則に従ってください。LIST コマンドの GROUP または LIST パラメーターに関して総称指定を使用する場合は、CEDA の場合と同じ方法で記号 * および + を使用できます。

FROMCSD パラメーターには、オペレーティング・システムの JCL に関する規則に準拠した有効な DD 名が含まれている必要があります。

有効なコマンド・シーケンスの例が、 398 ページの図 6 に示されています。その他のコマンド例は、以下のコマンドの説明に示されています。

© Copyright IBM Corp. 1982, 2018

```
*                               SET UP INITIAL CSD FILE
INITialize
*
LIst LIst(DFHLIST) Objects
*                               UPGRADE FROM EARLIER RELEASE
UPgrade
*
LI Group(PPTM1)
LI G(SETM*)
*                                  CREATE GROUP PCTZ4
Copy G(PCTM1)  To(PCTZ4)
C G(SETMP3) T(PCTZ4) Replace
LI G(P++M+)
*                                  CREATE LIST MODLIST
APpend LIst(TESTLIST) TO(MODLIST) FRomcsd(CSDF1)
AP LI(SECLIST)  To(MODLIST) FR(CSDF1)
AP LI(DFHLIST)  To(MODLIST)
*
LI ALL OBJECTS
```

図 *6. DFHCSDUP* プログラムのサンプル・コマンド

第 5 章 リソース定義マクロ

ほとんどの CICS リソースは複数の方法で定義することができますが、一部のリソースでは、CICS マクロを使用する必要があります。

CLT: コマンド・リスト・テーブル

コマンド・リスト・テーブル (CLT) は、拡張リカバリー機能 (XRF) で使用され、テークオーバー中に発行される MVS システム・コマンドおよびオペレーターへのメッセージのリストが含まれます。通常、これらのコマンドの機能は、代替システムに対して、同じ MRO 接続された構成内のアクティブ・システムからのテークオーバーを行うように通知するためのものです。

コマンド・リスト・テーブルには、代替システムの名前と、取り消すことが可能なアクティブ・システムのジョブ名も含まれます。 (DFHCLT TYPE=LISTSTART FORALT オペランドを参照してください。) これは、代替システムによるテークオーバー時に、誤ったジョブが取り消されることに対するセキュリティー検査を提供します。

さらに、DFHCLT TYPE=INITIAL マクロは、該当する MVS システムに取り消しコマンドを送信するために必要な JES ルーティング情報を提供します。 XRF を使用する場合は、CLT が必要です。これは、代替 CICS システムのみが使用します。

セキュリティー上の理由から、CLT は APF の使用が許可されたライブラリーにリンク・エディットされます。仮想ストレージ制約解放の考慮事項に対応するために、AMODE(31),RMODE(ANY) を指定して MODE 制御ステートメントを使用することでリンク・エディットします。テーブルは、再入可能としてリンク・エディットする必要があります。 CLT は、CICS 中核にロードされません。

CLT には、以下のステートメントを含めることができます。

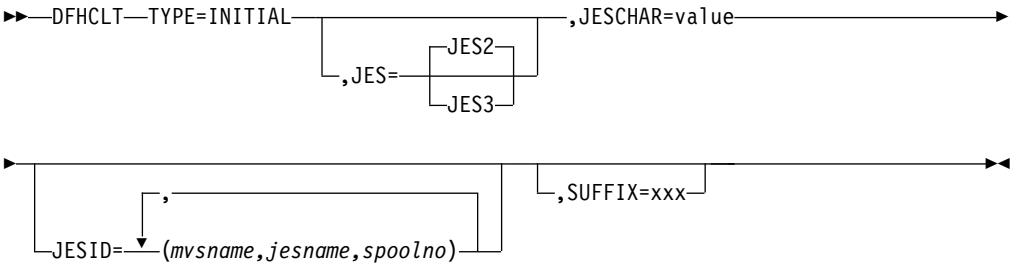
- DFHCLT TYPE=INITIAL
- DFHCLT TYPE=LISTSTART
- DFHCLT TYPE=COMMAND
- DFHCLT TYPE=WTO
- DFHCLT TYPE=LISTEND
- DFHCLT TYPE=FINAL (TYPE=FINAL (テーブルの終わり)を参照)

注: CLT を複数の代替システムで共用することは可能ですが、テークオーバー中に多くの重複コマンドで MVS が指定されないように注意してください。例えば、複数領域操作において複数の領域に対するコマンドに 1 つの CLT を使用した場合、領域 1 が他の領域に有効なコマンドを送信したものの、それを受け取った領域が領域 1 に、あるいは各領域間で重複するコマンドを送信する場合があります。

制御セクション: DFHCLT TYPE=INITIAL

DFHCLT TYPE=INITIAL マクロは、定義する CLT のエントリー・ポイントおよび開始アドレスを設定します。

© Copyright IBM Corp. 1982, 2018



SUFFIX オペランドの使用方法を含む TYPE=INITIAL マクロに関する一般情報は、 TYPE=INITIAL (制御セクション)を参照してください。

JES={JES2∨JES3}
使用する JES のバージョンを指定します。アクティブ CICS システムと代替システムが異なる CPC 内にある場合、両方の CPC で同じバージョンの JES を使用する必要があります。

JES3 を使用する場合、2 つの CPC を使用する環境ですべての CLT 関数を完全にサポートするには、リリース 2.2.0 が必要です。 JES2 は、プログラムによって発行されたコマンドを受け入れます。 JES3 では、この機能はリリース 2.2.0 のためにのみ導入されました。それより前の JES3 リリースでは、2 つの CPC を使用する環境でのテークオーバーは、障害のあるアクティブ CICS システムをオペレーターが手動で取り消した後でなければ処理できない可能性があります。これは、アクティブ・システムが、自身に障害があることを認識せずに稼働を継続する場合にのみ発生します。

JESCHAR=value
JES に渡されるコマンドに使用する 1 文字の接頭部を指定します。このキーワードを省略すると、以下のようになります。
- JESCHAR=$ が JES=JES2 の場合のデフォルトになります
- JESCHAR=* が JES=JES3 の場合のデフォルトになります

JESID=((mvsname,jesname,spoolno) [,(mvsname,jesname,spoolno),...])
アクティブ CICS システムの MVS 名および JES 名に対応する JES 宛先コードを指定します。アクティブ CICS システムと代替 CICS システムが異なる CPC 内にある場合に、このオプションを使用する必要があります。

mvsname、*jesname*、および *spoolno* の複数のグループを指定することができるため、CLT を使用してより多くの CPC/JESの組み合わせを参照できるようになります。

mvsname
これは、アクティブ CICS システムが実行されている CPC について、SYS1.PARMLIB member SMFPRM*xx* で指定された SID です。

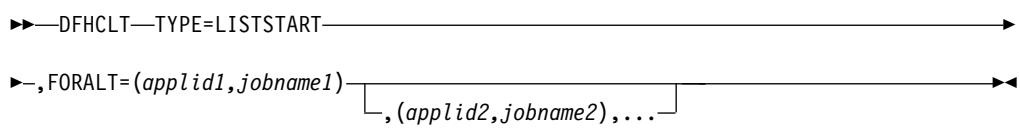
jesname
これは、アクティブ・システムの実行を制御している JES の JES2 または JES3 サブシステム名です。これは、MVS/ESA SCHEDULR sysgen マクロおよび SYS1.PARMLIB メンバー IEFSSN*xx* で定義されます。

spoolno
JES2 の場合、これは、アクティブ CICS システムの JES2 のマルチアクセス・スプール・メンバー番号です。これは、JES2 初期設定パラメーター

MASDEF SSID(*n*) で定義されます。JES3 の場合、これは、アクティブ CICS システムの JES3 の処理プログラム名です。これは、JES3 初期設定パラメーター MAINPROC NAME=*name* で定義されます。 z/OS JES2 初期設定およびチューニング ガイドまたはz/OS JES3 初期設定およびチューニング ガイドを参照してください。

代替システムの指定: DFHCLT TYPE=LISTSTART

このマクロは、代替 CICS が、アクティブ CICS からテークオーバーする際に発行するコマンドとメッセージのセットの開始を定義します。 (コマンドやメッセージがない場合がありますが、許可検査を実行できるように CLT は引き続き必要です。)



FORALT=((applid1,jobname1)[,(applid2,jobname2),...])
代替 CICS システムとアクティブ CICS システムのペアを指定します。

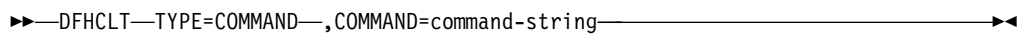
applid1
テークオーバーの際にコマンドとメッセージのセットを発行する代替 CICS の名前。この名前は、APPLID システム初期設定パラメーターで定義された固有のアプリケーション **ID** でなければなりません。これは、許可検査として使用されます。

jobname1
代替システムによるテークオーバー元のアクティブ CICS システムの名前。この名前は、アクティブ CICS システムの **MVS JOBNAME** でなければなりません。これは、代替システムが、その名前のジョブ以外のジョブの取り消しを試行しないようにするためのセキュリティー検査として使用されます。

これは、*applid* および *jobname* の追加ペアを使用して拡張することができます。それにより、複数の代替 CICS システムに対して 1 つの CLT を使用することが可能になります。

テークオーバー・コマンドの指定: DFHCLT TYPE=COMMAND

このマクロを使用して、代替 CICS システムがテークオーバー中に使用するコマンドを指定することができます。



COMMAND=command-string
実行のために MVS に渡されるコマンドを定義します。CICS は、このコマンドを解釈しません。

この方法で最も頻繁に発行されるコマンドは、CEBT PERFORM TAKEOVER です。

マスター領域と従属領域の単純な階層が存在する複数領域操作 (MRO) では、障害が発生したマスター領域は、従属領域も別の CPC に移動する必要がある場合には、それぞれの従属領域にこのコマンドを発行することができます。

さらに複雑な複数領域操作では、障害が発生したマスター領域は、このコマンドをそのコーディネーター領域に発行することができ、コーディネーターは、領域内の同じ階層にある他のマスター領域および従属領域に同じコマンドを発行することができます。したがって、多くの MRO 接続された領域は、オペレーターが介入しなくても、一緒に別の CPC に移動することができます。

以下にいくつかの例を示します。

- コーディネーターなしのマスター領域が従属領域にコマンドを送信する場合:

```
DFHCLT TYPE=COMMAND,
       COMMAND='MODIFY CICSDEP,CEBT PERFORM
       TAKEOVER'
```

- マスター領域がそのコーディネーター領域にコマンドを送信する場合:

```
DFHCLT TYPE=COMMAND,
       COMMAND='MODIFY CICSCRD,CEBT PERFORM
       TAKEOVER'
```

- コーディネーター領域がマスター領域と従属領域にコマンドを送信する場合:

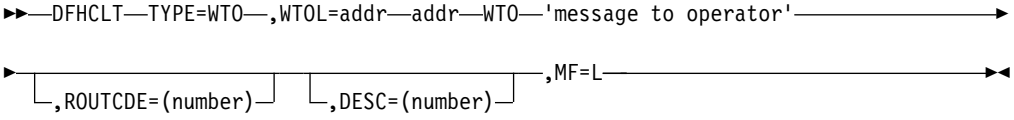
```
DFHCLT TYPE=COMMAND,
       COMMAND='MODIFY CICSMAS,CEBT PERFORM
       TAKEOVER'
DFHCLT TYPE=COMMAND,
       COMMAND='MODIFY CICSDEP,CEBT PERFORM
       TAKEOVER'
```

- その他のコマンドを、MVS で実行されている他の任意のジョブに発行することもできます。

```
DFHCLT TYPE=COMMAND,
       COMMAND='MODIFY jobname,command
       string'
```

オペレーターへのメッセージ: DFHCLT TYPE=WTO

これら 2 つの命令は、システム・オペレーターに対して書き込まれるメッセージを定義します。



WTOL=addr
メッセージおよび関連の宛先コードと記述子コードを定義する WTO マクロのアドレスをリスト形式で指定します。

WTO マクロの MF (マクロ形式)、ROUTCDE (宛先コード)、および DESC (記述子) オペランドについては、「z/OS MVS Programming: Authorized Assembler Services Reference (Volume 1)」で説明しています。

以下の例は、要求をオペレーターに送信するためのものです。

```
          DFHCLT TYPE=WTO,
                 WTOL=wtoad
wtoad     WTO   'switch local terminals, please',
                 MF=L
```

コマンド・リストのクローズ: DFHCLT TYPE=LISTEND

この命令は、代替システムが、アクティブ・システムからテークオーバーする際に発行するコマンドとメッセージのセットの終了を定義します。

PDIR: DL/I ディレクトリー

PDIR は、CICS システムがアクセスするすべてのリモート・プログラム仕様ブロック (PSB) のディレクトリーです。

リモート DL/I を使用するには、CICS に PDIR を定義する必要があります。DBCTL の場合は、IMS 提供のマクロを使用して DBCTL に DL/I ディレクトリーを定義する必要があります。

プログラム仕様ブロック・ディレクトリー (PDIR) を作成するには、DFHDLPSB、TYPE=INITIAL、TYPE=ENTRY、および TYPE=FINAL マクロをコーディングする必要があります。TYPE=FINAL (TYPE=FINAL (テーブルの終わり) を参照) の後は以下のようにする必要があります。

```
END DFSIDIR0
```

制御セクション: DFHDLPSB TYPE=INITIAL

TYPE=INITIAL マクロは、テーブルの制御セクション (CSECT) を設定し、必要なリンケージ・エディター制御ステートメントを生成します。

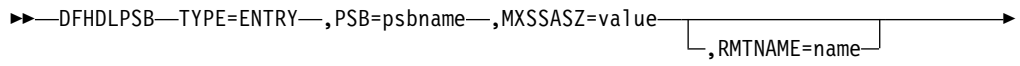
DFHDLPSB TYPE=INITIAL マクロには、以下のフォーマットおよびオペランドがあります。

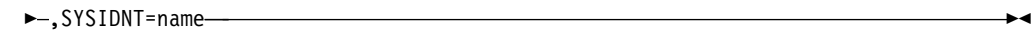
SUFFIX オペランドの使用など、TYPE=INITIAL マクロに関する一般情報については、TYPE=INITIAL (制御セクション)を参照してください。

プログラム仕様ブロック: DFHDLPSB TYPE=ENTRY

TYPE=ENTRY マクロは、PDIR に生成されるエントリーを定義します。

DFHDLPSB TYPE=ENTRY マクロには、以下のフォーマットおよびオペランドがあります。





TYPE=ENTRY
PDIR に生成されるエントリーを示します。PDIR に含めることができるエントリーの最大数は 32760 です。

PSB=psbname
リモート DL/I 経由でアクセスするプログラム仕様ブロック (PSB) の名前を指定します。エントリーは、SYSIDNT オペランドと MXSSASZ オペランド (およびオプションで RMTNAME オペランド) を指定する必要があります。

MXSSASZ=value
この PSB に使用するセグメント検索引数の最大サイズ (バイト数) を指定します。

注: MXSSASZ の値が大きすぎると、パフォーマンスに悪影響を及ぼします。また、接続された CICS システムにとって大きすぎるデータ・ストリームがシップされる原因となる場合があります。

RMTNAME=name
リモートのシステムまたは領域で PSB を認識するための名前を示します。デフォルトは、PSB オペランドで指定された *psbname* です。

SYSIDNT=name
PSB を適用可能なリモートのシステムまたは領域の 4 文字の英数字名を示します。指定する名前は、リモート・システムの CONNECTION 定義の名前でなければなりません。

ローカル・システムの SYSIDNT が指定された場合、要求は DBCTL に転送され、リモート領域に機能シップされません。これにより、リンクの両サイドで同じ PDIR を使用することが可能になります (必要な場合)。ただし、DBCTL と通信する場合は、PDIR は必要ではありません。

FCT: ファイル管理テーブル

ファイル管理テーブル (FCT) は、ファイル管理によって処理される基本直接アクセス方式 (BDAM) ユーザー・ファイルを CICS に記述します。

CICS ユーザー・ファイルは、物理データ・セットに対応します。物理データ・セットは、使用する前に MVS に定義し、CICS システムに割り振られている必要があります。

注:

1. VSAM ファイルを定義するには、FILE リソースを使用します。
2. CICS ファイル管理は、VSAM データ・セットと BDAM データ・セットしか処理しないため、順次データ・セットは、TDQUEUE リソースを使用して区画外宛先として定義してください。

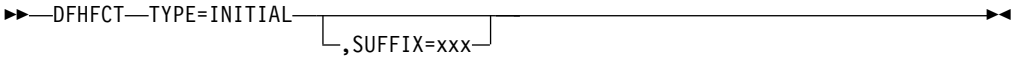
以下のマクロは、ファイル特性を指定するほか、ファイルによって参照される BDAM データ・セットの特性のいくつかを指定します。

- DFHFCT TYPE=INITIAL は、FCT の開始を設定します。

- DFHFCT TYPE=FILE は、ファイルの特性 (レコード特性や許可されるサービスのタイプなど) を定義します。
- DFHFCT TYPE=FINAL は、FCT を終了します。(TYPE=FINAL (テーブルの終わり)を参照。)

制御セクション: DFHFCT TYPE=INITIAL

DFHFCT TYPE=INITIAL マクロは、FCT をアセンブルする制御セクションを設定します。このマクロは、FCT のアセンブルに使用するソースで、最初のステートメントとしてコーディングする必要があります。

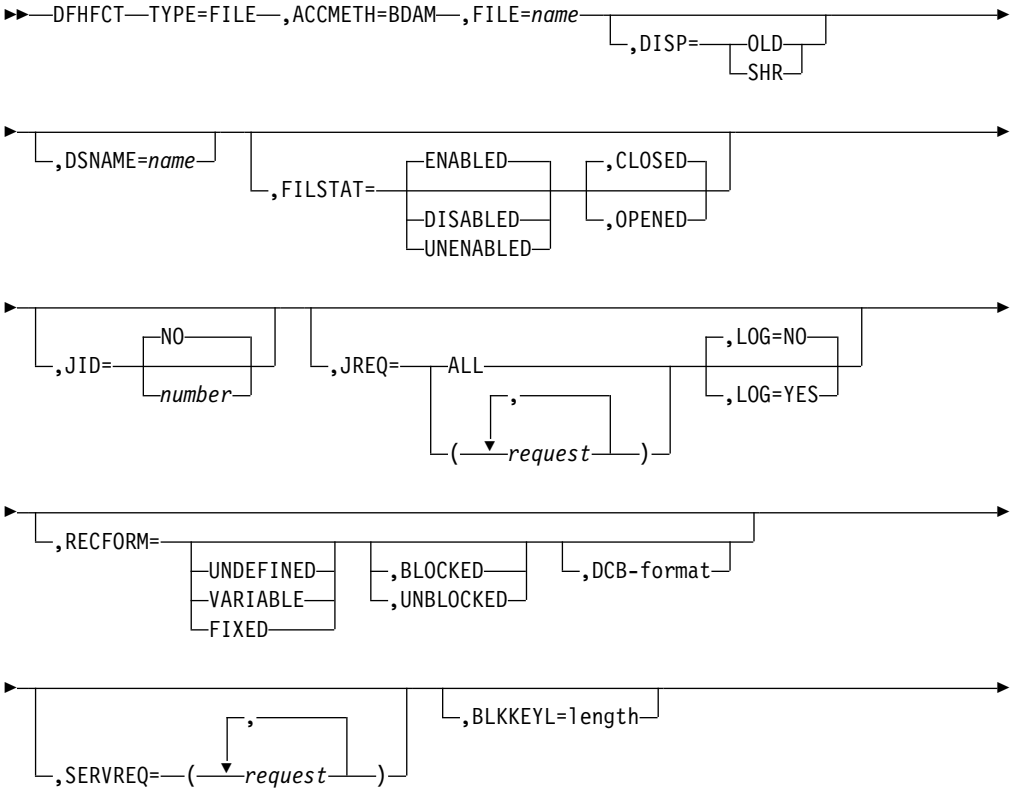


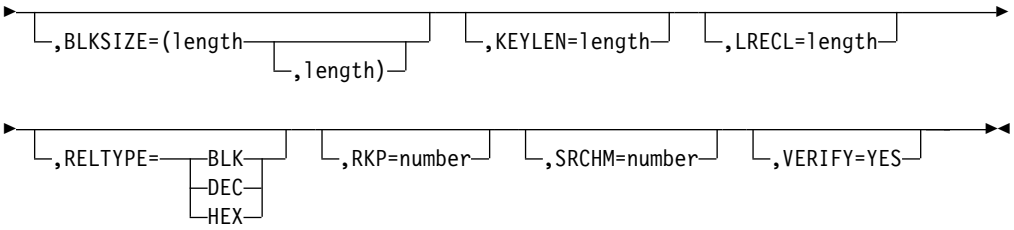
SUFFIX オペランドの使用など、TYPE=INITIAL マクロに関する一般情報については、TYPE=INITIAL (制御セクション)を参照してください。

ローカル・ファイル — DFHFCT TYPE=FILE

DFHFCT TYPE=FILE マクロは、CICS ファイルに対して BDAM ファイルの物理特性および操作特性を制御するように記述します。

このマクロには、アクセス方式、レコード特性、ファイルに対して許可されるサービスのタイプに関する情報を提供するオペランドが含まれます。この情報を使用して、CICS および DCB が使用する制御情報が生成されます。





TYPE=FILE
このマクロが、ファイルの特性を記述することを示します。

ACCMETH=BDAM
データ・セットの基本直接アクセス方式がファイルに割り振られることを指定します。

BLKKEYL=length
BDAM 物理レコード内の物理キーの長さ (バイト数) を表す 1 から 255 までの 10 進数値を指定して、これをコーディングします。このオペランドをコーディングする必要があるのは、物理キーを使用してデータ・セットを参照する (SERVREQ=KEY が指定されている) ファイルに対してのみです。ブロック化レコードがデータ・セットに含まれており、論理キー (各論理レコードに組み込まれたキー) を使用してブロック解除を実行する場合は、KEYLEN オペランドを使用して論理キーの長さを指定する必要があります。

必要な場合は、CICS は、データ・セット名、ブロック参照、および物理キーを連結して ENQ 引数を作成することで、レコードを排他制御に入れることができます。 ENQ は、この引数 (最大 255 バイト) を使用して発行されます。引数の長さが 255 バイトを超えると、ENQ は一連のキーを排他制御に入れます。

BLKSIZE=(length[,length])
ブロックの長さ (バイト数) を指定して、これをコーディングします。BLKSIZE を計算する方法は、RECFORM によって異なります。 UNDEFINED ブロックまたは VARIABLE ブロックの場合、この長さは、ブロックの最大長でなければなりません。固定長ブロックの場合は、以下のように BLKSIZE を計算します。

BLKSIZE = LRECL (非ブロック化レコードの場合)

BLKSIZE = (LRECL x ブロック化因数) (ブロック化レコードの場合)

DCB で BLKSIZE 値を生成したい場合、その値をオペランドの 2 番目のパラメーターで指定する必要があります。例えば、BLKSIZE=(250,250) のように指定します。ここで、最初の 250 は、FCT に関連しており、2 つ目の 250 は、DCB に関連しています。 2 番目のパラメーターがコーディングされていない場合、DCB は BLKSIZE 値を使用せずに生成されます。

注:

1. CICS は、BDAM データ・セットのブロック・サイズを、最初の BLKSIZE パラメーターで指定したサイズであると見なします。指定された値がデータ・セットの実際のブロック・サイズより小さい場合、ファイルの使用時に、高い確率で記憶保護違反またはその他の予期しない結果が発生します。

2. 2 番目のパラメーター (DCB 値) を指定する場合、コーディングする値は、常に実際のブロック・サイズでなければなりません。 2 番目のパラメーターは、省略するか、最初のパラメーターと等しい値にすることをお勧めします。

DISP={OLD∨SHR}

このファイルに割り振るデータ・セットの後処理を指定するには、これをコーディングします。ファイルを開く際にそのファイル用の JCL ステートメントが存在しない場合、ファイルを開く前に、この後処理を使用してファイルの動的割り振りが行われます。 JCL ステートメントが存在する場合は、そのステートメントがこの後処理より優先されます。

OLD 動的割り振りを実行した場合に、データ・セットの後処理が OLD に設定されます。

SHR 動的割り振りを実行した場合に、データ・セットの後処理が SHR に設定されます。

注**:** データ・セットの後処理は、DISP オペランドまたは以下のいずれかを使用して指定する必要があります。
- JCL ステートメント
- CEMT SET
- EXEC CICS SET

JCL ステートメントで後処理を指定する場合は、データ・セット名も JCL ステートメントで指定します。

DSNAME=name

このファイルで使用する JCL データ・セット名 (DSNAME) を指定するための、1 から 44 文字のコード。ファイルを開く際にそのファイル用の JCL ステートメントが存在しない場合、ファイルを開く前に、この DSNAME を使用してファイルの動的割り振りが行われます。 JCL ステートメントが存在する場合は、そのステートメントがこの DSNAME より優先されます。

データ・セット名は、DSNAME オペランドまたは以下のいずれかを使用して指定する必要があります。
- JCL ステートメント
- CEMT SET
- EXEC CICS SET

JCL ステートメントでデータ・セットを指定する場合は、後処理も JCL ステートメントで指定する必要があります。

注**:** CICS システム定義 (CSD) ファイルは、FCT ではなく、システム初期設定パラメーターで定義します。

FILE=name

この FCT エントリーを識別する 1 から 8 文字のシンボル名を指定して、これをコーディングします。この名前は、ファイル名と呼ばれ、CICS あるいは CICS アプリケーション・プログラムが、この FCT エントリーに関連付けられたデータ・セットを参照するために使用します。

この名前は、FCT エントリーの識別だけではなく、関連付けられたデータ・セットが CICS に割り振られる際の DDNAME としても使用されます。割り振り

は、始動ジョブ・ストリーム内の JCL ステートメントを使用するか、あるいは動的に FCT 内の DSNAME 値および DISP 値を使用して行われます。

所有ファイルに文字ストリング DFH から始まるファイル名は使用しないでください。これは、CICS が DFH から始まるすべてのファイル名を使用する権利を予約しているためです。さらに、ファイル名接頭部に文字ストリング FCT を使用すると、アセンブリー・エラーが発生する可能性があります。

FILSTAT=({ENABLED∨DISABLED∨UNENABLED},{OPENED∨CLOSED})
ファイルの初期状況を指定するには、これをコーディングします。

第 1 オペランドは、ファイルの初期の使用可能化状態を決定します。これは、初期始動またはコールド・スタートでのみ使用されます。 (ウォーム・スタートおよび緊急始動では、ファイル状態は、前回のシャットダウン時の状態によって決まります。)

第 2 オペランドは、CICS 初期設定の終了時にファイルのオープンを試行するかどうかを指定します。これは、初期始動、コールド・スタート、ウォーム・スタート、および緊急始動に適用されます。

ENABLED
このファイルに対しては、通常の処理が許可されます。

DISABLED
アプリケーション・プログラムからこのファイルに対するすべての要求で、「使用不可」状態がプログラムに渡されます。

UNENABLED
このオプションは、CLOSED オプションと一緒に使用した場合にのみ有効です。これを使用して、最初の参照時にファイルが開かれるのを防ぐことができます。この状態のファイルにアクセスしようとすると、NOTOPEN 状態になります。

OPENED
CICS 初期設定の後に自動的に開始される CICS トランザクション (CSFU) によって、ファイルが開かれます。 (ウォーム・スタートおよび緊急始動では、ファイルは UNENABLED のままです (前回のシャットダウン時にその状態であった場合)。 CSFU は、UNENABLED ファイルの OPENED オプションを無視し、ファイルを閉じたままにします。)

CLOSED
マスター端末機能、EXEC CICS SET コマンド、または暗黙的オープンによってファイル・オープンが要求されない限り、そのファイルは閉じたままになります。

初期状態の各組み合わせによって、ファイルは以下のように開かれます。

(ENABLED,CLOSED)
ファイルは、最初の参照時に開かれます。これはデフォルトです。

(ENABLED,OPENED)
ファイルは、ユーザー・アプリケーションまたはマスター端末機能によって先に開かれた場合を除き、CICS 初期設定の後に自動的に開始されるトランザクション CSFU によって開かれます。

(DISABLED,CLOSED)
ファイルは、明示的な OPEN 要求 (例えば、マスター端末トランザクションから) によってのみ開かれます。

(DISABLED,OPENED)
ファイルは、ユーザー・アプリケーションまたはマスター端末機能によって明示的に先に開かれた場合を除き、CICS 初期設定の後に自動的に開始されるトランザクション CSFU によって開かれます。

(UNENABLED,CLOSED)
ファイルは、明示的な OPEN 要求によってのみ開かれます。ファイルが開かれた後のファイル状態は、(ENABLED,OPENED) です。

注: パフォーマンス上の理由から、トランザクション CSFU のデフォルトの CSD ファイル・エントリーは、DTIMOUT=10 (秒) で定義されます。そのため、CICS 使用時にファイル・オープンで遅延が発生すると、トランザクション・タイムアウト異常終了が発生する可能性があります。 DTIMOUT 値の説明については、TRANSACTION 属性を参照してください。

JID={NO∨number}
この FCT エントリーに対して自動ジャーナル活動が行われる場合に、ジャーナル・データの記録に使用するジャーナルを識別するには、これをコーディングします。データ・レコードがジャーナル処理される原因となる操作は、JREQ パラメーターで指定します。

NO このファイルに対して、自動ジャーナル処理活動は実行されません。

number
自動ジャーナル処理に使用されるジャーナル ID。これは、01 から 99 の範囲の任意の数値です。この数値が文字 DFHJ に付加され、DFHJ*nn* の形式のジャーナル名が提供されます。このジャーナル名が、MVS システム・ロガーの汎用ログ・ストリームにマップされます。

注: ユーザー自身による後続の処理 (例えば、ユーザー作成データ・セットの入出力リカバリーなど) に関するファイル活動を記録したい場合は、自動ジャーナル処理を指定することができます。 (LOG=YES で指定する) 自動ログ記録と混同しないようにしてください。自動ログ記録は、緊急時再始動や動的トランザクション・バックアウトの際に、CICS がデータ・セット・バックアウトを実行して進行中のタスク活動を除去する場合に必要です。

JREQ={ALL∨(request[,request,...])}
自動的にジャーナル処理されるファイル操作、およびジャーナル処理をファイル活動と同期にするか非同期にするかによって、これをコーディングします。

READ 要求に対して同期ジャーナル操作が実行される場合、読み取られたデータがジャーナル・データ・セットに書き込まれるまで、ファイル制御要求を発行したプログラムに制御は返されません。 WRITE 要求に対して同期ジャーナル操作が実行される場合、データ・セットへの出力操作は、データがジャーナル・データ・セットに書き込まれるまで開始されません。

READ 要求に対して非同期ジャーナル操作が実行される場合、読み取られたデータがジャーナル入出力バッファーに移動されるとすぐに、制御を返すことがで

きます。 WRITE 要求に対して非同期ジャーナル操作が実行される場合、データ・セットへの出力操作は、データがジャーナル入出力バッファーに移動されるとすぐに開始できます。

同期に関するデフォルトは、READ については非同期操作、WRITE については同期操作を提供します。

自動ジャーナル処理を要求した場合、ジャーナルのコンテンツが、データ・セットに対する実際の変更を正確に反映していないことがあります。これは、入出力操作からの応答がテストされる前に、要求がジャーナル処理されるためです。

このオペランドが省略され、JID がコーディングされた場合、JREQ はデフォルトで JREQ=(WU,WN) になります。

request に使用可能な値は、以下のとおりです。

ALL READ 非同期および WRITE 同期によって、すべてのファイル活動をジャーナル処理します。

ASY WRITE 操作の非同期ジャーナル操作。

RO READ ONLY 操作をジャーナル処理します。

RU READ UPDATE 操作をジャーナル処理します。

SYN READ 操作の同期ジャーナル操作。

WN WRITE NEW 操作をジャーナル処理します。

WU WRITE UPDATE 操作をジャーナル処理します。

KEYLEN=length

このファイルが参照する BDAM データ・セットのブロック化を解除するための論理キーの長さを指定して、これをコーディングします。

BDAM データ・セットの論理キーは、RKP オペランドを使用して組み込みおよび配置されます。物理キーの長さは BLKKEYL オペランドでコーディングされ、KEYLEN に指定された値とは異なる可能性があります。

ブロック化された BDAM データ・セットで論理キーを使用する場合は、必ずこのオペランドをコーディングする必要があります。

LOG={NO|YES}

このオペランドは、ファイルのリカバリー属性を指定します。自動ログ記録を行う場合は、LOG=YES を指定します。これにより、緊急時再始動やトランザクションの異常終了が発生した場合に、このファイルが参照するデータ・セットに対する不完全な変更のバックアウト (リカバリー) が可能になります。データ・セットに対する変更/更新、削除、あるいは追加が行われる場合は、必ず自動的に「before」イメージが CICS システム・ログに記録されます。 (自動ログ記録と自動ジャーナル処理を混同しないようにしてください。)

NO 自動ログ記録は実行されません。

YES

自動ログ記録が実行されます。

ファイルが参照するデータ・セットのコンテンツに対する変更が要求されると、レコード ID と一緒に基本データ・セットを表す CICS 制御ブロックのアドレスを使用して、更新、追加、あるいは削除されるレコードがエンキューされます。このエンキューは、タスクが終了するか、アプリケーション

が同期点要求を発行して作業論理単位の終了を通知するまで保持されます。これにより、変更されるデータの保全性が確保されます。

このように、エンキューが長期間保持されるため、このデータ・セットにアクセスするアプリケーション・プログラムが、そのデータ・セットに対して実質的に複数の作業論理単位を実行した場合に、同期点要求の発行によって CICS に対する個別の作業論理単位の定義が行われていないと、エンキューのロックアウトが発生する可能性があります。また、長時間実行されるタスクが、ストレージ・リソースを拘束する可能性もあります。

LRECL=length
論理レコードの最大長 (バイト数) を指定して、これをコーディングします。指定する値は、固定長リモート・ファイル内のレコードの長さでもあります。リモート・ファイルについて詳しくは、DFHFCT TYPE=REMOTE マクロを参照してください。

RECFORM=([{UNDEFINED∨VARIABLE∨FIXED}], [{BLOCKED∨UNBLOCKED}],[DCB format])
データ・セット内の物理レコードのフォーマットを記述するには、これをコーディングします。

BDAM データ・セットの場合、ブロック化は、CICS のブロック化を意味し、BDAM には意味はありません。 FIXED フォーマットまたは VARIABLE フォーマットのすべてのデータ・セットに対して、BLOCKED または UNBLOCKED を指定する必要があります。

BLOCKED
各物理レコードが複数の論理レコードから構成されている 1 つのブロックとして CICS から認識されるようにするには、このオプションを指定します。

DCB
DCB に挿入されるレコード形式を指定して、これをコーディングします。例えば、RECFORM=(FIXED,BLOCKED,FBS) のようになります。

RECFORM オペランドの DCB 形式のサブパラメーターは、FCT のアセンブル時にレコード形式情報を DCB に入れることができる唯一の方法です。RECFORM オペランドの最初の 2 つのサブパラメーターは、DCB 内で情報を生成しません。

FIXED
レコードは固定長です。

UNBLOCKED
使用する CICS ブロック構造がない場合、このオプションを指定します。つまり、BDAM 物理レコードごとに 1 つの CICS 論理レコードがある場合です。

UNDEFINED
レコードは未定義長です。 (データ・セットを UNDEFINED として指定した場合、BLKSIZE を計算する際に、カウント・フィールドに 8 バイトを追加することができます。)

VARIABLE
レコードは可変長です。

RELTYPE={BLK∨DEC∨HEX}

このファイルが参照する BDAM データ・セットのレコード ID フィールドのブロック参照部分で相対アドレス指定が使用する場合は、これをコーディングします。 RELTYPE オペランドが省略された場合、絶対アドレス指定 (MBBCCHHR) であると見なされます。

BLK 相対ブロック・アドレス指定方式が使用されています。

DEC ゾーン 10 進フォーマットが使用されています。

HEX 16 進相対トラックおよびレコード・フォーマットが使用されています。

RKP=number

レコード内のキー・フィールドの開始位置 (レコードの開始からの相対位置) を指定して、これをコーディングします。可変長レコードでは、このオペランドには、各論理レコードの先頭にある 4 バイトの LLbb フィールド用のスペースを含める必要があります。このオペランドは、各論理レコード内にキーを持つデータ・セットの場合、あるいはブラウズする場合には、必ずコーディングする必要があります。

SERVREQ=(request[,request],...)

ファイルに対して処理可能なサービス要求のタイプを定義するには、これをコーディングします。含めることができるパラメーターは、以下のとおりです。

ADD

レコードをそのファイルに追加できます。

BROWSE

レコードをファイルから順次に取得することができます。

KEY

レコードをファイルから取得したり、ファイルに追加したりすることができます。ファイルによって参照されるデータ・セットがキー指定 BDAM データ・セットである場合、このパラメーターは必須です。他のファイルの場合は、コーディングしないでください。

NOEXCTL

更新のための読み取りが要求された場合に、レコードは排他制御に入れられません。

NOEXCTL を指定しない場合、BDAM 排他制御がデフォルトで提供されます。これにより、システム内での保全性が提供されます。BDAM の場合、SERVREQ=NOEXCTL と一緒に LOG=YES を指定することができます。これは、CICS エンキューのみを要求し、BDAM 排他制御は抑止します。そのため、同期点までの更新についてのみ、CICS 保全性が提供されます。

注: BDAM 排他制御が物理ブロック上で動作し、システム全体を制御し、更新が完了するまでしか存続しないのに対して、CICS エンキューは、CICS 領域内でレコード・レベルであり、同期点まで存続します。

READ

このファイル内のレコードを読み取ることができます。 BROWSE または UPDATE を指定した場合は、READ であると見なされます。

UPDATE
このファイル内のレコードを変更することができます。

SRCHM=number
キー指定されたレコードに対するマルチトラック検索が提供される場合、これをコーディングします。このオペランドは、BDAM のキー指定データ・セットにのみ適用されます。

number
検索するトラックまたはブロックの数。デフォルトは 0 です。

VERIFY=YES
ディスク・レコードの書き込み後に、そのディスク・レコードのパリティーを検査する場合は、これをコーディングします。このオペランドが省略された場合、書き込み要求の後にレコードは検証されません。

サマリー表

このセクションは、DFHFCT TYPE=FILE マクロを使用してファイルを定義するのに役立つ情報を提供します。各 TYPE=FILE 命令は、ファイルの特性およびファイルによって参照されるデータ・セットの特性を記述します。

表 *12. BDAM* ファイル用の *DFHFCT TYPE=FILE* 命令

	キーありのブロック	キーなしのブロック	キーありの非ブロック	キーなしの非ブロック
BLKKEYL	R		R	
SRCHM	O		O	
VERIFY	O	O	O	O
RELTYPE	R[1]	R[1]	R[1]	R[1]
LRECL	R	R	R	R
BLKSIZE	R[3]	R	R[2]	R
KEYLEN	R[5]			
RKP	R[4]		R[4]	
RECFORM	O	O	O	O
FILSTAT	O	O	O	O
SERVREQ	R[6]	O	R[6]	O

注:
R 必須
O オプション
1. 相対アドレス指定タイプを使用する場合は必須。
2. SERVREQ=BROWSE または SERVREQ=ADD である場合、この値は、非ブロック化レコードでは BLKSIZE + BLKKEYL でなければなりません。
3. SERVREQ=BROWSE または SERVREQ=ADD である場合、この値は、ブロック化レコードでは (LRECL x ブロック化因数) + BLKKEYL でなければなりません。
4. 論理レコード内にキーが存在する場合は必須。
5. BDAM でキーによるブロック化解除を行う場合は必須。
6. SERVREQ=KEY は必須です。

DFHFCT の例

BDAM ファイルの FCT エントリーの例。

図 7 は、BDAM ファイルの FCT エントリーを作成するために必要なコーディングを示しています。

```
DFHFCT TYPE=FILE,                                *
       FILE=DAM83,                               *
       ACCMETH=BDAM,                             *
       SERVREQ=(READ,BROWSE,KEY),                *
       BLKSIZE=172,                              *
       RECFORM=(FIXED,BLOCKED),                  *
       LRECL=86,                                 *
       RELTYPE=HEX,                              *
       KEYLEN=6,                                 *
       BLKKEYL=6,                                *
       RKP=0,                                    *
       FILSTAT=(ENABLED,OPENED)
```

図 *7.* ファイル管理テーブルの例: *BDAM* ファイル

MCT: モニター管理テーブル

モニター管理テーブル (MCT) は、CICS モニター・パフォーマンス・クラス・レコードのユーザー・データ・フィールドを定義し、それらがイベント・モニター・ポイント (EMP) でどのように操作されるかを記述します。また、どのシステム定義のパフォーマンス・クラス・データ・フィールドを記録するかも制御します。

MCT およびパフォーマンスの詳細については、モニター管理テーブル (MCT) を参照してください。

MCT は、以下の状況でのみ必要です。

- アプリケーション・プログラムで EMP を呼び出す場合。
- システム定義の特定のフィールドを記録対象から除外することが必要な場合。
- モニター・リソース・クラスを使用する場合。
- トランザクションで使用するリソース・マネージャーについて、付加的なモニター・パフォーマンス・データを収集する場合。
- モニター・レコードに対するデータ圧縮を非活動化する場合。

CICS 領域内に MCT が存在しない場合、CICS は、以下のデフォルトを使用します。

- パフォーマンス、トランザクション・リソース、および例外の各モニター・クラスは使用可能。
- すべての CICS システム定義のデータ・フィールドを収集する。
- モニター・レコードに対してデータ圧縮を実行する。

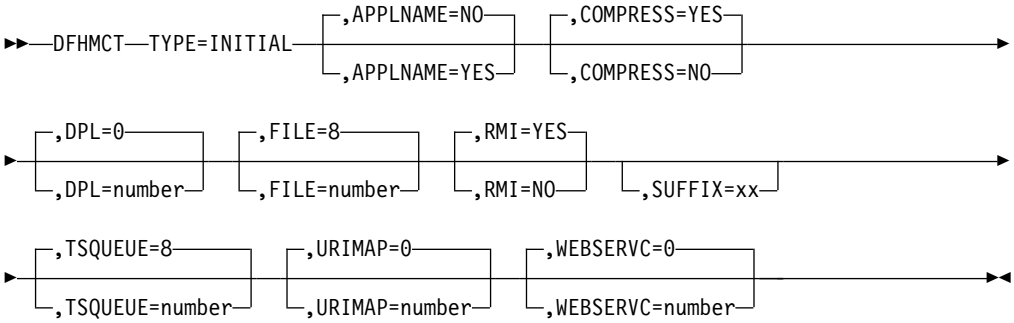
MONITOR を使用して、ご使用のアプリケーション・プログラムで EMP を呼び出すことができます。

MCT は、以下のマクロ命令で構成されます。

- 制御セクション、DFHMCT TYPE=INITIAL
- ユーザー・イベント・モニター・ポイント、DFHMCT TYPE=EMP
- 制御データの記録、DFHMCT TYPE=RECORD
- モニター管理テーブルの終了、DFHMCT TYPE=FINAL (TYPE=FINAL (テーブルの終わり) を参照)

制御セクション: DFHMCT TYPE=INITIAL

MCT の制御セクション名は、DFHMCT TYPE=INITIAL マクロによって設定されます。このマクロは、後続のリンク・エディットに必要なリンケージ・エディターの制御ステートメントも作成します。



SUFFIX オペランドの使用など、TYPE=INITIAL マクロに関する一般情報については、TYPE=INITIAL (制御セクション)を参照してください。

注:

1. すべてのトランザクション・リソース・モニター・データの結合した長さは、32244 バイトを超えてはなりません。
2. 一部のオプションは、トランザクション・リソース・モニターが有効な場合にのみ適用されます。(MN=ON も指定して) MNRES=ON をシステム初期設定パラメーターとしてコーディングすると、始動時にトランザクション・リソース・クラス・モニターを使用可能にできます。 代わりに、モニター機能トランザクション CEMN、CEMT、または **EXEC CICS SET MONITOR** コマンドを使用して、トランザクション・リソース・モニターを動的に有効にすることもできます。

APPLNAME={NO|YES}

このオプションは、CICS モニターによって提供されるアプリケーション・ネーミング・サポートを使用することを指定します。

アプリケーション・ネーミングは、アプリケーション・プログラムから特殊な CICS イベント・モニター・ポイントを呼び出すことのできる、使用可能化機能です。これらの CICS 生成の EMP において収集されたデータは、任意の CICS モニター・ソフトウェア・パッケージで使用することができます。

NO CICS 領域でアプリケーション・ネーミング・サポートを使用可能にせず、アプリケーション・ネーミング・イベント・モニター・ポイント DFHAPPL.1 および DFHAPPL.2 を生成しません。

YES CICS 領域でアプリケーション・ネーミング・サポートを使用可能にし

ます。CICS は、MCT のアセンブル時にアプリケーション・ネーミング・イベント・モニター・ポイント (DFHAPPL.1 および DFHAPPL.2) を生成します。これらのアプリケーション・ネーミング EMP を呼び出すコマンド MONITOR によって移動されたモニター・データは、タスクが終了するか、後続の **EXEC CICS MONITOR** コマンドによって別の EMP の呼び出しで変更されるまで保持されるので注意してください。

アプリケーション・ネーミング (DFHAPPL) EMP は、TYPE=EMP マクロのパラメーターで定義された場合と同様に、CICS によって作成されます。

```
DFHMCT TYPE=EMP,CLASS=PERFORM,                                  X
               ID=(DFHAPPL.1),FIELD=(1,APPLNAME)                X
               PERFORM=(MOVE(0,4))
DFHMCT TYPE=EMP,CLASS=PERFORM,                                  X
               ID=(DFHAPPL.2),                                  X
               PERFORM=(MOVE(4,8))
```

アプリケーション・ネーミングのイベント・モニター・ポイントをアプリケーションで使用する方法について詳しくは、アプリケーション・ネーミング・イベント・モニター・ポイントを参照してください。

COMPRESS={YES|NO}

このオプションは、CICS モニター機能によって生成される CICS SMF 110 モニター・レコードに対してデータ圧縮を実行するかどうかを指定します。

YES このデフォルトは、CICS モニター機能によって生成される CICS SMF 110 モニター・レコードに対してデータ圧縮を実行するように CICS に指示します。データ圧縮については、レコード・モニターのためのデータ圧縮を参照してください。

NO CICS SMF 110 モニター・レコードのデータ圧縮が無効になります。

DPL={0|*number*}

このオプションは、CICS がトランザクション・リソース・モニターを実行する対象の分散プログラム・リンク (DPL) の最大数を指定します。このオプションは、注 2 に記載しているように、トランザクション・リソース・モニターが有効な場合のみ適用されます。

CICS 標準モニター・パフォーマンス・クラス・データには、分散プログラム・リンクを含め、トランザクションによってアクセスされるすべての プログラムの合計が含まれています。他方、トランザクション・リソース・モニターは、指定された数までの個別の分散プログラム・リンクに関する情報を収集します。以下のデータが収集されます。

- プログラム名
- 要求が経路指定されたシステム ID (sysid)
- 分散プログラム・リンク (DPL) 要求の数

0 このデフォルトを使用すると、MNRES=ON が有効な場合でも、分散プログラム・リンクのトランザクション・リソース・モニターは無効になります。

number

CICS がトランザクション・リソース・モニターを実行する対象の分散プログラム・リンク要求の最大数を、0 から 64 の範囲で指定します。CICS は、トランザクションによって発行された分散プログラム・リン

ク要求ごとに、リソース・レベルでのモニター・パフォーマンス・データを *number* で指定された最大数まで収集します。トランザクションが指定された数よりも多くの分散プログラム・リンク要求を発行する場合、最大数を超える要求は無視されますが、トランザクションが DPL 限度を超過したことを示すフラグが設定されます。

FILE={8|*number*}

このオプションは、CICS がトランザクション・リソース・モニターを実行する対象のファイルの最大数を指定します。このオプションは、注 2 に記載しているように、トランザクション・リソース・モニターが有効な場合のみ適用されます。

CICS 標準モニター・パフォーマンス・クラス・データには、トランザクションによってアクセスされるすべての ファイルの合計が含まれています。他方、トランザクション・リソース・モニターは、指定された数までの個別のファイルに関する情報を収集します。以下のデータが収集されます。

- ファイル名 (File name)
- ファイル GET 要求の数および合計時間
- ファイル PUT 要求の数および合計時間
- ファイル・ブラウズ要求の数および合計時間
- ファイル追加要求の数および合計時間
- ファイル削除要求の数および合計時間
- ファイルに対する要求の総数および合計時間
- ファイル・アクセス・メソッド要求の数
- ファイル入出力待ち時間および待機の数
- RLS モードのファイル入出力待ち時間および待機数
- カップリング・ファシリティー・データ・テーブル (CFDT) 入出力待ち時間および待機数
- 排他制御の待ち時間および待機数
- VSAM ストリングの待ち時間および待機数

8 このデフォルトは、最大 8 個のファイルに対してトランザクション・リソース・モニターを実行するように CICS に指示します。

number

CICS がトランザクション・リソース・モニターを実行する対象のファイルの最大数を 0 から 64 の範囲で指定します。CICS は、*number* で指定された最大数まで、トランザクションによってアクセスされる各ファイルに対して、リソース・レベルでモニター・パフォーマンス・データを収集します。トランザクションが指定された数よりも多くのファイルにアクセスする場合、 最大数を超えるファイルは無視されますが、トランザクションがファイルの限度を超過したことを示すフラグが設定されます。

FILE=0 を指定した場合、MNRES=ON が有効な場合でも、ファイルに対するトランザクション・リソース・モニターのデータは収集されません。

RMI={YES|NO}

このオプションは、トランザクションで使用されるリソース・マネージャー用に追加のモニター・パフォーマンス・クラス・データを収集するかどうかを指定します 。

YES このデフォルトは、トランザクションで使用するリソース・マネージャーについて、追加のモニター・パフォーマンス・データを収集するように CICS に指示します。

収集されるデータについては、グループ DFHRMI 内のパフォーマンス・データ を参照してください。

NO これは、トランザクションで使用するリソース・マネージャーについて、追加のモニター・パフォーマンス・データを収集しないように CICS に指示します。

TSQUEUE={8|*number*}

このオプションは、CICS がトランザクション・リソース・モニターを実行する対象の一時記憶域キューの最大数を指定します。このオプションは、注 2 に記載しているように、トランザクション・リソース・モニターが有効な場合のみ適用されます。

CICS 標準モニター・パフォーマンス・クラス・データには、トランザクションによってアクセスされるすべての 一時記憶域キューの合計が含まれています。他方、トランザクション・リソース・モニターは、指定された数までの個別の一時記憶域キューに関する情報を収集します。以下のデータが収集されます。

- 一時記憶域キュー名
- 一時記憶域キュー GET 要求の数および合計時間
- 共用一時記憶域に対する一時記憶域キュー GET 要求の数および合計時間
- 補助一時記憶域に対する一時記憶域キュー PUT 要求の数および合計時間
- 主一時記憶域に対する一時記憶域キュー PUT 要求の数および合計時間
- 共用一時記憶域に対する一時記憶域キュー PUT 要求の数および合計時間
- 一時記憶域キューに対する要求の総数および合計時間
- 一時記憶から取得されたすべての項目の全長。
- 共用一時記憶域から取得されたすべての項目の全長
- 補助一時記憶域に書き込まれたすべての項目の全長。
- 主一時記憶に書き込まれたすべての項目の全長。
- 共用一時記憶域に書き込まれたすべての項目の全長
- 一時記憶域入出力待ち時間および待機の数
- 共用一時記憶域の入出力待ち時間および待機の数

8 このデフォルトは、最大 8 個の一時記憶域キューに対してトランザクション・リソース・モニターを実行するように CICS に指示します。

number

CICS がトランザクション・リソース・モニターを実行する対象の一時記憶域キューの最大数を 0 から 64 の範囲で指定します。CICS は、*number* で指定された最大数まで、トランザクションによってアクセスされる各一時記憶域キューに対して、リソース・レベルでモニター・パフォーマンス・データを収集します。トランザクションが指定された数よりも多くの一時記憶域キューにアクセスする場合、最大数を超える一時記憶域キューは無視されますが、トランザクションが一時記憶域キューの限度を超過したことを示すフラグが設定されます。

TSQUEUE=0 を指定した場合、MNRES=ON が有効な場合でも、一時記憶域キューに対するトランザクション・リソース・モニターのデータは収集されません。

URIMAP={0|*number*}

このオプションは、**WEB OPEN URIMAP** コマンドに指定された URIMAP のうち、CICS のトランザクション・リソース・モニターの実行対象にする最大数を指定します。このオプションは、注 2 に記載しているように、トランザクション・リソース・モニターが有効な場合のみ適用されます。

CICS の標準のモニター・パフォーマンス・クラスのデータには、トランザクションで使用されたすべての URIMAP の合計値が含まれます。一方、トランザクション・リソース・モニターでは、指定した数まで個々の URIMAP に関する情報が収集されます。以下のデータが収集されます。

- URIMAP name (URIMAP 名)
- この URIMAP についてユーザー・タスクが発行した **WEB OPEN URIMAP** 要求の完了を待機して経過した時間。
- この URIMAP についてユーザー・タスクがセッション・トークンに対して発行した以下の要求の完了を待機して経過した時間。
  - **WEB RECEIVE** 要求
  - 受信側の **WEB CONVERSE** 要求
- この URIMAP についてユーザー・タスクがセッション・トークンに対して発行した以下の要求の完了を待機して経過した時間。
  - **WEB SEND** 要求
  - 送信側の **WEB CONVERSE** 要求

0
: このデフォルトを使用すると、MNRES=ON が有効な場合でも、URIMAP のトランザクション・リソース・モニターは無効になります。

number
: CICS がトランザクション・リソース・モニターを実行する対象の URIMAP の最大数を 0 から 64 の範囲で指定します。*number* で指定した最大数まで、CICS は、トランザクションから発行された **WEB OPEN URIMAP** コマンドに指定されていた URIMAP ごとに、リソース・レベルのモニター・パフォーマンス・データを収集します。指定した数を超える個数の URIMAP がトランザクションで開かれた場合、最大数を超える要求は無視され、トランザクションが URIMAP の上限を超えたことを示すフラグが設定されます。

WEBSERVC={0|*number*}

このオプションは、**INVOKE SERVICE** コマンドで使用された WEBSERVICE のうち、CICS のトランザクション・リソース・モニターの実行対象にする最大数を指定します。このオプションは、注 2 に記載しているように、トランザクション・リソース・モニターが有効な場合のみ適用されます。

CICS の標準のモニター・パフォーマンス・クラスのデータには、トランザクションで使用されたすべての WEBSERVICE の合計値が含まれます。一方、トランザクション・リソース・モニターでは、指定した数まで個々の WEBSERVICE に関する情報が収集されます。以下のデータが収集されます。

- WEBSERVICE name (WEBSERVICE 名)
- パイプライン名

- この WEBSERVICE についてユーザー・タスクが発行した **INVOKE SERVICE** 要求の完了を待機して経過した時間。

0 このデフォルトを使用すると、MNRES=ON が有効な場合でも、WEBSERVICE のトランザクション・リソース・モニターは無効になります。

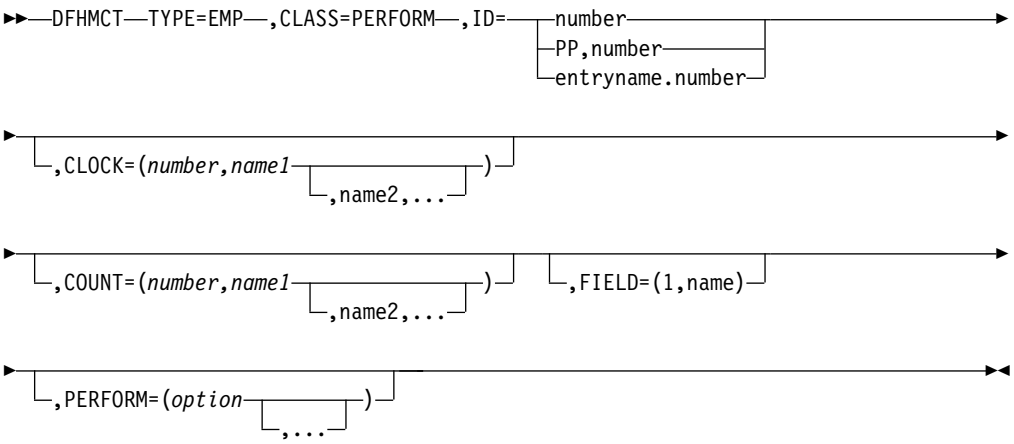
number

CICS がトランザクション・リソース・モニターを実行する対象の WEBSERVICE の最大数を 0 から 64 の範囲で指定します。*number* で指定した最大数まで、CICS は、トランザクションから発行された **INVOKE SERVICE** コマンドで使用された WEBSERVICE ごとに、リソース・レベルのモニター・パフォーマンス・データを収集します。指定した数を超える個数の WEBSERVICE がトランザクションで呼び出された場合、最大数を超える要求は無視され、トランザクションが WEBSERVICE の上限を超えたことを示すフラグが設定されます。

ユーザー・イベント・モニター・ポイント: DFHMCT TYPE=EMP

DFHMCT TYPE=EMP マクロを使用して、パフォーマンス・クラス・データ・レコード内のユーザー・データ・フィールドを、各ユーザー・イベント・モニター・ポイントにどのように追加するか、あるいは各ユーザー・イベント・モニター・ポイントでどのように変更するかを指定することができます。ユーザー・データが必要なユーザー・イベント・モニター・ポイントごとに、1 つの TYPE=EMP マクロがコーディングされている必要があります。

TYPE=EMP マクロは、TYPE=INITIAL マクロと最初の TYPE=RECORD マクロ命令の間にコーディングする必要があります。



TYPE=EMP

このマクロが、ユーザー・イベント・モニター・ポイントで収集するユーザー・データを定義することを示します。

CLASS=PERFORM

このユーザー EMP でユーザー・データを収集する対象のモニター・クラスを指定して、これをコーディングします。値 PERFORM をコーディングする必要があります。対応する PERFORM オペランドもコーディングする必要があります。

ID={*number*|(PP,*number*)|*entryname.number*}

このマクロで定義されたユーザー・データを収集するユーザー・イベント・モニター・ポイントの ID を指定して、これをコーディングします。形式 *number* または (PP,*number*) のいずれかがコーディングされた場合、デフォルトのエントリー名 USER が指定されます。

number

1 から 255 の範囲の 10 進整数。ユーザー EMP には、1 から 199 の範囲の ID 番号が使用可能です。200 から 255 の範囲の数値は、IBM ライセンス・プログラムの EMP に予約されています。IBM ライセンス・プログラムのコードで定義された EMP でユーザー・データを収集する場合は、これらの数値をコーディングしてください。

(PP,*number*)

IBM ライセンス・プログラム EMP の ID 番号。これは、ID 値に 199 より大きい数値を指定するのと同等です。*number* の値は、1 から 56 の範囲の 10 進整数です。

entryname.number

number (1 から 255 の範囲の 10 進整数) を複数使用することができます。したがって、UNIQUE.3、DSN.3、および 3 は、それぞれ異なる 3 つの EMP になります。特定の数値に対して、最大 98 個のエントリー名を指定することができます。また、それらの EMP の 1 つによって参照されるカウント、クロック、あるいはバイト・オフセットは、その他の EMP によって参照されるものとは別のオブジェクトになります。

以下の説明における定数はすべて、最大 8 桁の 16 進数字で表される 16 進定数を意味します。それより短いストリングの場合は、左側にゼロが埋め込まれます。例えば、10 進数 14 を加算または減算する場合、定数は 0000000E または単に E (引用符は不要です) としてコーディングされます。

フィールド DATA1 および DATA2 についての言及はすべて、アプリケーション・プログラムでコーディングされたユーザー EMP によって提供される 2 つのバイナリー・フルワードを意味します。これらは、ユーザー EMP を定義するための EXEC CICS MONITOR コマンドの DATA1 オペランドおよび DATA2 オペランドによって指定されます。コーディングされるオプションによって、DATA1 フィールドおよび DATA2 フィールドが数値として解釈されるか、論理演算を実行するためのマスクとして解釈されるか、あるいは詳細情報へのポインターとして解釈されるかは異なります。

数値についての言及はすべて、オプションの記述で定義された範囲の 10 進整数を意味します。

CLOCK=(*number*,*name1*[,*name2*,...])

1 つ以上のクロックに非公式名を割り当てます。クロックの非公式名はその辞書エントリーに表示され、後処理プログラムが、例えば列見出しとして使用することができます。

文字ストリング *name1* は、MCT の生成時に *number* で指定されたクロックに割り当てられます。指定された場合、*name2* は、クロック *number* + 1 に割り当てられます。同様に、後続の名前が後続のクロックに割り当てられます。このオプションで指定されないクロックは、ID オペランドからエントリー名の値を受け取ります (デフォルトは USER)。

number は、1 から 256 の範囲でなければなりません。指定する名前は、それぞれ最大 8 文字の長さの文字ストリングでなければなりません。ストリングに 1 つ以上のブランクまたはコンマが含まれている場合は、そのストリングを引用符で囲む必要があります。

COUNT=(number,*name1*[,*name2*,...])
1 つ以上のカウント・フィールドに非公式名を割り当てます。カウント・フィールドの非公式名はその辞書エントリーに表示され、後処理プログラムが、例えば列見出しとして使用することができます。

文字ストリング *name1* は、MCT の生成時に number で指定されたカウント・フィールドに割り当てられます。指定された場合、*name2* は、カウント・フィールド *number*+1 に割り当てられます。同様に、後続の名前が後続のカウント・フィールドに割り当てられます。このオプションで指定されないカウント・フィールドは、ID オペランドからエントリー名の値を受け取ります (デフォルトは USER)。

number は、1 から 256 の範囲でなければなりません。指定する名前は、それぞれ最大 8 文字の長さの文字ストリングでなければなりません。ストリングに 1 つ以上のブランクまたはコンマが含まれている場合は、そのストリングを引用符で囲む必要があります。

FIELD=(1,*name*)
ユーザー・バイト・ストリング・フィールドに非公式名を割り当てます。この名前はその辞書エントリーに表示され、後処理プログラムが、例えば列見出しとして使用することができます。

name は、最大 8 文字の長さの文字ストリングでなければなりません。ストリングに 1 つ以上のブランクまたはコンマが含まれている場合は、そのストリングを引用符で囲む必要があります。

PERFORM=(*option*[,...])
CLASS=PERFORM が指定されている場合は、このオペランドをコーディングします。これは、この EMP でのパフォーマンス・クラス・データ・レコードのユーザー・フィールドに情報が追加される、あるいは情報が変更されることを指定します。

ID オペランド内の別個のエントリー名によって識別される各ユーザーのユーザー・フィールドは、以下によって構成されます。
1. 最大 256 個のカウンター
2. 最大 256 個のクロック (それぞれが最大 8 バイトのアキュムレーターおよび 4 バイトのカウントから構成)
3. 最大 8192 バイトのバイト・ストリング

注: 指定されたオプションで暗黙指定されたオブジェクト (クロック、カウント、およびフィールド) の結合サイズが 16384 バイトを超えた場合、アセンブリー時エラーが発生します。このエラーは、収集するデータを減らすか、クロックおよびカウントへの参照をクラスター化して、使用されないオブジェクトが暗黙指定されることを防止することで、使用するオブジェクトを減らすことによって回避できます。

注: ユーザー・イベント・モニター・ポイントで収集するユーザー・データを定義する場合、これにより、すべての CICS パフォーマンス・クラスのモニタ

ー・レコードのサイズが拡張されます。各 CICS モニター・レコードのサイズは、最大レコードのサイズと同じです。ユーザー・データ・フィールドを指定する際には、このことを念頭においてください。

指定されたオプションに従って、ユーザー・フィールドに対してアクションが実行されます。

PERFORM は、PER に省略することができます。PERFORM オペランドで有効なオプションは、以下のとおりです。

ADDCNT(*number*,{*constant*|DATA1|DATA2})
number によって指定されるユーザー・カウント・フィールドの値は、*constant* ずつ、あるいはフィールド DATA1 または DATA2 の値ずつ増えます。*number* は、1 から 256 の範囲の 10 進整数です。

EXCNT(*number*,{*constant*|DATA1|DATA2})
constant あるいはフィールド DATA1 または DATA2 の値を使用して *number* で指定されたユーザー・カウント・フィールドの値に対して、論理排他 OR 演算が実行されます。*number* は、1 から 256 の範囲の 10 進整数です。

MLTCNT(*number1*,*number2*)
DATA1 フィールドによってアドレス指定される領域内の隣接フルワードに含まれる値を追加することで、一連の隣接ユーザー・カウント・フィールドが更新されます。このオプションを使用するには、DATA1 フィールドと DATA2 フィールドの両方がユーザー EMP から渡される必要があります。

更新されるユーザー・カウント・フィールドは、*number1* で指定されたフィールドから開始されます。更新されるユーザー・カウント・フィールドの数は、*number2* の値と DATA2 フィールドの値のうちの小さいほうの数です。DATA2 フィールドがゼロの場合は、*number2* の値が使用されます。ユーザー・カウント・フィールドに追加するために使用される一連の隣接フルワードは、DATA1 フィールドで指定されたアドレスから開始されます。連続するフルワードが、連続するユーザー・カウント・フィールドに追加されます。

number1 および *number2* は、1 から 256 の範囲の 10 進整数です。生成されるユーザー・カウントの数は、(*number1* + *number2* - 1) です。この値も 1 から 256 の範囲でなければなりません。

注: 各 DFHMCT TYPE=EMP マクロでは、MLTCNT オプションと MOVE オプションのいずれか 1 つのみを使用することができます。

MOVE(*number3*,*number4*)
データのストリングがユーザー・バイト・ストリング・フィールドに移動されます。このオプションを使用するには、DATA1 フィールドと DATA2 フィールドの両方がユーザー EMP から渡される必要があります。

ユーザー・バイト・ストリング・フィールドは、*number3* で指定されたオフセットから更新されます。移動するデータは、DATA1 フィールドで指定されたアドレスから開始されます。移動できるデータの最大長は、*number4* (バイト数) で指定され、移動するデータの実際の長さは、DATA2 フィールドの値で指定されます。DATA2 の値がゼロの場合は、*number4* で指定された長さのデータが移動されます。

number3 は、0 から 8191 の範囲の 10 進整数で、*number4* は、1 から 8192 の範囲の 10 進整数です。ユーザー文字フィールドの最大長は、(*number3* + *number4*) で、1 から 8192 の範囲でなければなりません。

注: 各 DFHMCT TYPE=EMP マクロ命令では、MLTCNT オプションと MOVE オプションのいずれか 1 つのみを使用することができます。

NACNT(*number*,{*constant*|DATA1|DATA2})
constant あるいはフィールド DATA1 または DATA2 の値を使用して *number* で指定されたユーザー・カウント・フィールドの値に対して、論理 AND 演算が実行されます。*number* は、1 から 256 の範囲の 10 進整数です。

ORCNT(*number*,{*constant*|DATA1|DATA2})
constant あるいはフィールド DATA1 または DATA2 の値を使用して *number* で指定されたユーザー・カウント・フィールドの値に対して、論理包含 OR 演算が実行されます。*number* は、1 から 256 の範囲の 10 進整数です。

PCLOCK(*number*)
number によって指定されたクロックが停止されます。ユーザー・クロック・フィールドの 4 バイトのカウントに、クロックが現在停止していることを示すフラグが立てられます。アキュムレーターは、前回の SCLOCK の前のコンテンツと、その SCLOCK とこの PCLOCK の間の経過時間の合計に設定されます。*number* は、1 から 256 の範囲の 10 進整数です。

PCPUCLK(*number*)
このオプションは、PCLOCK と同じ機能を実行しますが、経過時間ではなく、CICS メインタスクの CPU 時間を使用します。

SCLOCK(*number*)
number によって指定されたクロックが開始されます。ユーザー・クロック・フィールドの 4 バイトのカウントの値が 1 ずつ増分し、その実行状態を示すフラグが立てられます。*constant* は、1 から 256 の範囲の 10 進整数です。

SCPUCLK(*number*)
このオプションは、SCLOCK と同じ機能を実行しますが、経過時間ではなく、CICS メインタスクの CPU 時間を使用します。

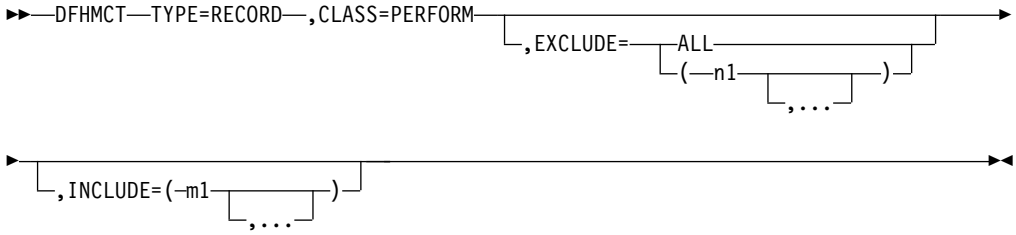
SUBCNT(*number*,{*constant*|DATA1|DATA2})
number によって指定されるユーザー・カウント・フィールドの値は、*constant* ずつ、あるいはフィールド DATA1 または DATA2 の値ずつ減ります。*number* は、1 から 256 の範囲の 10 進整数です。

DELIVER
この時点まで累算されたこのタスクのパフォーマンス・クラス・データが、モニター・バッファーに送信されます。実行中のすべてのクロックが停止します。主要フィールド (トランザクションID、端末 ID) および DFHAPPL 特殊 EMP の呼び出し結果として保管されたすべてのデータを除き、このタスクのモニター領域のパフォーマンス・クラス・セクションが X'00' にリセ

ットされます。このオプションによって停止されたすべてのクロックが、新しい測定期間としてゼロから再始動します。最高水準点フィールドは、現行値にリセットされます。

制御データの記録: DFHMCT TYPE=RECORD

DFHMCT TYPE=RECORD マクロは、モニター対象として選択されたパフォーマンス・クラス・データ・フィールドを識別します。



TYPE=RECORD
選択したパフォーマンス・クラスのデータ・フィールド用のモニター・データが記録されることを示します。

CLASS=PERFORM
パフォーマンス・クラス・データ・フィールドを記録するには、このオペランドをコーディングします。 PERFORM を PER に簡略化できます。

EXCLUDE={ALL|(*n1*[,...])}
モニター機能によって 1 つ以上の CICS フィールドが報告されないようにするには、このオペランドをコーディングします。デフォルトでは、記録されたすべてのパフォーマンス・クラス・フィールドが報告されます。

EXCLUDE オペランドは、コーディングした順序に関係なく、必ず INCLUDE オペランドの前に処理されます。 (INCLUDE オペランドは、EXCLUDE オペランドがコーディングされた場合にのみ適用されます。)

ALL
このオプションを使用すると、除外の対象となるすべてのフィールドが報告されなくなります。以下のフィールドを除外することはできません。

- 1
- 2
- 4
- 5
- 6
- 89

INCLUDE オペランドは、一部のフィールドを含めるが、大部分を除外する場合に、EXCLUDE=ALL と共に使用できます。

426 ページの表 13 は、除外の対象となるフィールドを示しています。各フィールドには、それに関連付けられたグループ名があり、このグループ名で各フィールドが属するフィールドのグループを識別します。また、各フィールドには、それ自身の数値フィールド ID もあります。

フィールドのグループを除外するには、グループの名前 (文字ストリング) を *n1* のようにコーディングします (例: `EXCLUDE=(DFHTASK)`)。

単一のフィールドを除外するには、フィールドの数値 ID を *n1* のようにコーディングします (例: `EXCLUDE=(98,70)`)。

注**:** 先行ゼロを数値 ID にコーディングしないでください。除外の対象でないフィールドの数値 ID をコーディングしないでください。

名前と数値 ID の組み合わせをコーディングすることができます (例: `EXCLUDE=(DFHFILE,DFHTERM,112,64)`)。

表 *13.* 除外または組み込むことができるデータ・グループおよびフィールド

グループ名	フィールド **ID**	フィールド名	説明
DFHCBTS	200	PRCSNAME	CICS BTS プロセス名
DFHCBTS	201	PRCSTYPE	CICS BTS プロセス・タイプ
DFHCBTS	202	PRCSID	CICS BTS プロセス ID
DFHCBTS	203	ACTVTYID	CICS BTS 活動 ID
DFHCBTS	204	ACTVTYNM	CICS BTS 活動名
DFHCBTS	205	BARSYNCT	CICS BTS プロセス/活動実行同期数
DFHCBTS	206	BARASYCT	CICS BTS プロセス/活動実行非同期数
DFHCBTS	207	BALKPACT	CICS BTS プロセス/活動リンク数
DFHCBTS	208	BADPROCT	CICS BTS プロセス定義数
DFHCBTS	209	BADACTCT	CICS BTS 活動定義数
DFHCBTS	210	BARSPACT	CICS BTS プロセス/活動リセット数
DFHCBTS	211	BASUPACT	CICS BTS プロセス/活動中断数
DFHCBTS	212	BARMPACT	CICS BTS プロセス/活動再開数
DFHCBTS	213	BADCPACT	CICS BTS 活動削除またはプロセス/活動取り消し要求数
DFHCBTS	214	BAACQPCT	CICS BTS プロセス/活動獲得要求数
DFHCBTS	215	BATOTPCT	CICS BTS プロセス/活動合計要求数
DFHCBTS	216	BAPRDCCT	CICS BTS プロセス・コンテナー削除/取得/プット・カウント
DFHCBTS	217	BAACDCCT	CICS BTS 活動コンテナー削除/取得/プット・カウント
DFHCBTS	218	BATOTCCT	CICS BTS プロセス/活動コンテナー合計要求数
DFHCBTS	219	BARATECT	CICS BTS 再接続検索要求数
DFHCBTS	220	BADFIECT	CICS BTS 入力定義イベント要求数
DFHCBTS	221	BATIAECT	CICS BTS タイマー関連イベント要求数
DFHCBTS	222	BATOTECT	CICS BTS イベント関連合計要求数
DFHCHNL	321	PGTOTCCT	チャネル・コンテナーに対する CICS 要求の数
DFHCHNL	322	PGBRWCCT	チャネル・コンテナーに対するブラウズ要求数
DFHCHNL	323	PGGETCCT	チャネル・コンテナーに対する GET CONTAINER 要求および GET64 CONTAINER 要求の数
DFHCHNL	324	PGPUTCCT	チャネル・コンテナーに対する PUT CONTAINER 要求および PUT64 CONTAINER 要求の数
DFHCHNL	325	PGMOVCCT	チャネル・コンテナーに対する MOVE CONTAINER 要求数

表 *13.* 除外または組み込むことができるデータ・グループおよびフィールド *(*続き*)*

グループ名	フィールド **ID**	フィールド名	説明
DFHCHNL	326	PGGETCDL	すべての GET CONTAINER CHANNEL コマンドおよび GET64 CONTAINER CHANNEL コマンドのコンテナー内のデータの長さ
DFHCHNL	327	PGPUTCDL	すべての PUT CONTAINER CHANNEL コマンドおよび PUT64 CONTAINER CHANNEL コマンドのコンテナー内のデータの長さ
DFHCHNL	328	PGCRECCT	チャネル・コンテナーに対する MOVE 要求、PUT CONTAINER 要求、および PUT64 CONTAINER 要求によって作成されたコンテナーの数
DFHCHNL	329	PGCSTHWM	割り振られたコンテナー・ストレージの最大量 (最高水準点)。
DFHCICS	25	CFCAPICT	CICS OO ファウンデーション・クラス要求数
DFHCICS	103	EXWTTIME	トランザクション例外の待ち時間
DFHCICS	112	RTYPE	パフォーマンス記録タイプ
DFHCICS	130	RSYSID	トランザクション・ルーティング SYSID
DFHCICS	131	PERRECNT	パフォーマンス・レコード・カウント
DFHCICS	167	SRVCLASS	MVS ワークロード・マネージャー・サービス・クラス名
DFHCICS	168	RPTCLASS	MVS ワークロード・マネージャー・レポート・クラス名。
DFHCICS	351	OADID	アダプターによって起点のデータに追加されたアダプター ID
DFHCICS	352	OADATA1	アダプターによって起点のデータに追加されたデータ
DFHCICS	353	OADATA2	アダプターによって起点のデータに追加されたデータ
DFHCICS	354	OSDATA3	アダプターによって起点のデータに追加されたデータ
DFHCICS	359	ONETWKID	この処理要求トランザクションの発信元のネットワーク ID
DFHCICS	360	OAPPLID	この処理要求の発信元の CICS 領域のアプリケーション ID
DFHCICS	361	OSTART	親タスクが開始された時刻
DFHCICS	362	OTRANNUM	親タスクの数
DFHCICS	363	OTRAN	親タスクのトランザクション ID (TRANSID)
DFHCICS	364	OUSERID	親タスクに応じて、発信 Userid-2 または Userid-1
DFHCICS	365	OUSERCOR	発信ユーザー関係子
DFHCICS	366	OTCPSVCE	発信 TCPIPSERVICE の名前
DFHCICS	367	OPORTNUM	発信 TCPIPSERVICE によって使用されるポート番号
DFHCICS	369	OCLIPORT	発信クライアントの TCP/IP ポート番号
DFHCICS	370	OTRANFLG	発信トランザクション・フラグ
DFHCICS	371	OFCTYNME	発信トランザクションのファシリティー名
DFHCICS	372	OCLIPADR	発信元クライアントまたは Telnet クライアントの IP アドレス

表 13. 除外または組み込むことができるデータ・グループおよびフィールド (続き)

グループ名	フィールド ID	フィールド名	説明
DFHCICS	373	PHNTWKID	このタスクが関連付けられている別の CICS システム内の直前のタスクの CICS システムのネットワーク ID
DFHCICS	374	PHAPPLID	直前のホップ・データのアプリケーション ID
DFHCICS	375	PHSTART	このタスクが関連付けられている別の CICS システム内の直前のタスクの開始時間
DFHCICS	376	PHTRANNO	このタスクが関連付けられている別の CICS システム内の直前のタスクのタスク番号
DFHCICS	377	PHTRAN	このタスクが関連付けられている別の CICS システム内の直前のタスクのトランザクション ID (TRANSID)
DFHCICS	378	PHCOUNT	このタスクに関連付けられているタスクを開始するために 1 つの CICS システムから別の CICS システムに要求が送られた回数。
DFHCICS	402	EICTOTCT	EXEC CICS API および SPI 要求数
DFHCICS	405	TIASKTCT	ASKTIME 要求数
DFHCICS	406	TITOTCT	TIME 要求の合計数
DFHCICS	408	BFDGSTCT	BIF DIGEST 要求数
DFHCICS	409	BFTOTCT	BIF 要求の合計数
DFHCICS	415	ECSIGECT	SIGNAL EVENT 要求数
DFHCICS	416	ECEFOPCT	イベント・フィルター操作数
DFHCICS	417	ECEVNTCT	イベント・キャプチャー数
DFHCICS	418	ECSEVCCT	ユーザー・タスクでキャプチャーされた同期発行イベントの数
DFHCICS	449	MPPRTXCD	このタスクが超過したポリシー・タスク規則しきい値の数
DFHCICS	464	NCGETCT	GET COUNTER 要求および DCOUNTER 要求の数
DFHCICS	466	MPSRECT	タスクに対してポリシー・システム規則が評価された回数
DFHCICS	467	MPSRACT	ポリシー・システム規則が「真」と評価され、メッセージまたはイベントをトリガーした回数
DFHCICS	480	PTSTART	タスクが関連付けられている、同じ CICS システム内の直前のタスクあるいは親タスクの開始時刻。
DFHCICS	481	PTTRANNO	タスクが関連付けられている、同じ CICS システム内の直前のタスクあるいは親タスクのタスク番号
DFHCICS	482	PTTRAN	タスクが関連付けられている、同じ CICS システム内の直前のタスクまたは親タスクのトランザクション ID (TRANSID)
DFHCICS	483	PTCOUNT	このタスクが関連付けられている、同じ CICS システム内の特定のタスクから別のタスクを開始するための要求 (**RUN TRANSID ASYNCHRONOUS** コマンドや **START** コマンドなど) があった回数
DFHDATA	179	IMSREQCT	IMS (DBCTL) 要求数
DFHDATA	180	DB2REQCT	Db2 要求数
DFHDATA	186	IMSWAIT	IMS(DBCTL) 待ち時間

表 *13.* 除外または組み込むことができるデータ・グループおよびフィールド *(*続き*)*

グループ名	フィールド **ID**	フィールド名	説明
DFHDATA	187	DB2RDYQW	Db2 Readyq 待ち時間
DFHDATA	188	DB2CONWT	Db2 接続待ち時間
DFHDATA	395	WMQREQCT	MQ 要求カウント
DFHDATA	396	WMQGETWT	MQ GETWAIT 待ち時間
DFHDATA	397	WMQASRBT	IBM MQ API SRB 時間
DFHDEST	41	TDGETCT	TD 取得カウント
DFHDEST	42	TDPUTCT	TD プット・カウント
DFHDEST	43	TDPURCT	TD パージ・カウント
DFHDEST	91	TDTOTCT	TD 合計カウント
DFHDEST	101	TDIOWTT	TD 入出力待ち時間
DFHDEST	403	TDILWTT	TD 区画内ロック待ち時間
DFHDEST	404	TDELWTT	TD 区画外ロック待ち時間
DFHDOCH	223	DHDELCT	ドキュメント・ハンドラー DELETE 数
DFHDOCH	226	DHCRECT	文書ハンドラー作成数
DFHDOCH	227	DHINSCT	文書ハンドラー挿入数
DFHDOCH	228	DHSETCT	文書ハンドラー設定数
DFHDOCH	229	DHRETCT	文書ハンドラー検索数
DFHDOCH	230	DHTOTCT	文書ハンドラー合計数
DFHDOCH	240	DHTOTDCL	文書ハンドラーの作成済み文書の合計の長さ
DFHFEPI	150	SZALLOCT	FEPI 割り振り数
DFHFEPI	151	SZRCVCT	FEPI 受信数
DFHFEPI	152	SZSENDCT	FEPI 送信数
DFHFEPI	153	SZSTRTCT	FEPI 開始数
DFHFEPI	154	SZCHROUT	FEPI 送信文字数
DFHFEPI	155	SZCHRIN	FEPI 受信文字数
DFHFEPI	156	SZWAIT	FEPI 一時停止時間
DFHFEPI	157	SZALLCTO	FEPI 割り振りタイムアウト回数
DFHFEPI	158	SZRCVTO	FEPI 受信タイムアウト回数
DFHFEPI	159	SZTOTCT	FEPI 合計数
DFHFILE	36	FCGETCT	FC 取得カウント
DFHFILE	37	FCPUTCT	FC プット・カウント
DFHFILE	38	FCBRWCT	FC ブラウズ・カウント
DFHFILE	39	FCADDCT	FC 追加カウント
DFHFILE	40	FCDELCT	FC 削除カウント
DFHFILE	63	FCIOWTT	FC 入出力待ち時間
DFHFILE	70	FCAMCT	FC アクセス方式カウント
DFHFILE	93	FCTOTCT	FC 合計カウント
DFHFILE	174	RLSWAIT	RLS FC 入出力待ち時間
DFHFILE	175	RLSCPUT	RLS ファイル要求 CPU (SRB) 時間
DFHFILE	176	CFDTWAIT	CFDT 入出力待ち時間
DFHFILE	426	FCXCWTT	VSAM 制御間隔の排他制御に対するファイル制御待ち時間
DFHFILE	427	FCVSWTT	ファイル制御の VSAM ストリング待ち時間
DFHJOUR	10	JCIOWTT	ジャーナル入出力待ち時間
DFHJOUR	58	JNLWRTCT	ジャーナル書き込みカウント
DFHJOUR	172	LOGWRTCT	ログ・ストリーム書き込みカウント
DFHMAPP	50	BMSMAPCT	BMS マップ・カウント
DFHMAPP	51	BMSINCT	BMS イン・カウント
DFHMAPP	52	BMSOUTCT	BMS アウト・カウント
DFHMAPP	90	BMSTOTCT	BMS 合計カウント

表 13. 除外または組み込むことができるデータ・グループおよびフィールド (続き)

グループ名	フィールド ID	フィールド名	説明
DFHPROG	55	PCLINKCT	プログラム・リンク・カウント
DFHPROG	56	PCXCTLCT	プログラム XCTL カウント
DFHPROG	57	PCLOADCT	プログラム・ロード・カウント
DFHPROG	71	PGMNAME	プログラム名
DFHPROG	72	PCLURMCT	プログラム・リンク URM カウント
DFHPROG	73	PCDPLCT	プログラム DPL カウント
DFHPROG	113	ABCODEO	オリジナル ABEND コード
DFHPROG	114	ABCODEC	現行 ABEND コード
DFHPROG	115	PCLOADTM	プログラム・ロード時間
DFHPROG	286	PCDLCSDL	CHANNEL オプションを指定したすべての DPL 要求のコンテナー内のデータの長さ
DFHPROG	287	PCDLCRDL	すべての DPL RETURN CHANNEL コマンドのコンテナー内のデータの長さ
DFHPROG	306	PCLNKCCT	CHANNEL オプションを指定したローカル LINK 要求の数
DFHPROG	307	PCXCLCCT	CHANNEL オプションを指定した XCTL 要求の数
DFHPROG	308	PCDPLCCT	CHANNEL オプションを指定した DPL 要求の数
DFHPROG	309	PCRTNCCT	CHANNEL オプションを指定したリモート RETURN 要求の数
DFHPROG	310	PCRTNCLD	すべてのリモート RETURN CHANNEL コマンドのコンテナー内のデータの長さ
DFHSOCK	241	SOIOWTT	インバウンド・ソケット入出力待ち時間
DFHSOCK	242	SOBYENCT	セキュア・ソケット用に暗号化されたバイト数
DFHSOCK	243	SOBYDECT	セキュア・ソケット用に暗号化解除されたバイト数
DFHSOCK	245	TCPSRVCE	TCP/IP サービス名
DFHSOCK	246	PORTNUM	TCP/IP サービス・ポート番号
DFHSOCK	288	ISALLOCT	IPCONN の割り振り要求の数
DFHSOCK	289	SOEXTRCT	ソケット抽出要求数
DFHSOCK	290	SOCNPSCT	非永続ソケット作成要求数
DFHSOCK	291	SOCPSCT	永続ソケット作成要求数
DFHSOCK	292	SONPSHWM	非永続ソケット最高水準点
DFHSOCK	293	SOPSHWM	永続ソケット最高水準点
DFHSOCK	294	SORCVCT	ソケット受信要求数
DFHSOCK	295	SOCHRIN	ソケット受信文字数
DFHSOCK	296	SOSENDCT	ソケット送信要求数
DFHSOCK	297	SOCHROUT	ソケット送信文字数
DFHSOCK	298	SOTOTCT	ソケット合計要求数
DFHSOCK	299	SOOIOWTT	アウトバウンド・ソケット入出力待ち時間
DFHSOCK	300	ISIOWTT	IPCONN のこの端点でユーザー・タスクが制御を待っている間に経過した時間
DFHSOCK	301	SOMSGIN1	インバウンド・ソケット受信要求数
DFHSOCK	302	SOCHRIN1	インバウンド・ソケット受信文字数
DFHSOCK	303	SOMSGOU1	インバウンド・ソケット送信要求数
DFHSOCK	304	SOCHROU1	インバウンド・ソケット送信文字数
DFHSOCK	305	ISIPICNM	TCPIPSERVICE がユーザー・タスクに接続した IPCONN の名前

表 *13.* 除外または組み込むことができるデータ・グループおよびフィールド (続き)

グループ名	フィールド **ID**	フィールド名	説明
DFHSOCK	318	CLIPADDR	発信元クライアントまたは Telnet クライアントの IP アドレス
DFHSOCK	319	ISALWTT	IPIC セッション割り振り待ち時間
DFHSOCK	320	SOCIPHER	インバウンド接続に使用する SSL ハンドシェーク用の暗号スイート・コード
DFHSOCK	330	CLIPPORT	クライアントまたは Telnet クライアントのポート番号
DFHSTOR	33	SCUSRHWM	ユーザー・ストレージ最高水準点 (UDSA)
DFHSTOR	54	SCUGETCT	ユーザー・ストレージ GET 数 (UDSA)
DFHSTOR	87	PCSTGHWM	プログラム・ストレージ最高水準点
DFHSTOR	95	SCUSRSTG	ユーザー・ストレージ占有 (バイト・ミリ秒) (UDSA)
DFHSTOR	105	SCUGETCT	16 MB を超えるユーザー・ストレージ GET 数 (EUDSA)
DFHSTOR	106	SCUSRHWM	16 MB を超えるユーザー・ストレージ最高水準点 (EUDSA)
DFHSTOR	107	SCUSRSTG	16 MB を超えるユーザー・ストレージ占有 (バイト・ミリ秒) (EUDSA)
DFHSTOR	108	PC24BHWM	16 MB 未満のプログラム・ストレージ最高水準点
DFHSTOR	116	SC24CHWM	16 MB 未満のユーザー・ストレージ最高水準点 (CDSA)
DFHSTOR	117	SCCGETCT	16 MB 未満のユーザー・ストレージ GET 数 (CDSA)
DFHSTOR	118	SC24COCC	16 MB 未満のユーザー・ストレージ占有 (バイト・ミリ秒) (CDSA)
DFHSTOR	119	SC31CHWM	16 MB を超えるユーザー・ストレージ最高水準点 (ECDSA)
DFHSTOR	120	SCCGETCT	16 MB を超えるユーザー・ストレージ GET 数 (ECDSA)
DFHSTOR	121	SC31COCC	16 MB を超えるユーザー・ストレージ占有 (バイト・ミリ秒) (ECDSA)
DFHSTOR	122	PC31RHWM	プログラム・ストレージ最高水準点 (ERDSA)
DFHSTOR	139	PC31AHWM	16 MB を超えるプログラム・ストレージ最高水準点
DFHSTOR	142	PC31CHWM	プログラム・ストレージ最高水準点 (ECDSA)
DFHSTOR	143	PC24CHWM	プログラム・ストレージ最高水準点 (CDSA)
DFHSTOR	144	SC24SGCT	16 MB 未満の Getmain 要求の共用ストレージ数 (CDSA および SDSA)
DFHSTOR	145	SC24GSHR	16 MB 未満の Getmain 要求を使用して割り振られた共用ストレージ・バイト数 (CDSA および SDSA)
DFHSTOR	146	SC24FSHR	16 MB 未満の Freemain 要求を使用して解放された共用ストレージ (CDSA および SDSA)
DFHSTOR	147	SC31SGCT	16 MB を超える Getmain 要求の共用ストレージ数 (ECDSA および ESDSA)

表 13. 除外または組み込むことができるデータ・グループおよびフィールド (続き)

グループ名	フィールド ID	フィールド名	説明
DFHSTOR	148	SC31GSHR	16 MB を超える Getmain 要求を使用して割り振られた共用ストレージ・バイト数 (ECDSA および ESDSA)
DFHSTOR	149	SC31FSHR	16 MB を超える Freemain 要求を使用して解放された共用ストレージ・バイト数 (ECDSA および ESDSA)
DFHSTOR	160	PC24SHWM	プログラム・ストレージ最高水準点 (SDSA)
DFHSTOR	161	PC31SHWM	プログラム・ストレージ最高水準点 (ESDSA)
DFHSTOR	162	PC24RHWM	プログラム・ストレージ最高水準点 (RDSA)
DFHSTOR	441	SC64CGCT	2 GB 境界より上のユーザー・ストレージ GET 数 (GCDSA)
DFHSTOR	442	SC64CHWM	2 GB 境界より上のユーザー・ストレージ最高水準点 (GCDSA)
DFHSTOR	443	SC64UGCT	2 GB 境界より上のユーザー・ストレージ GET 数 (GUDSA)
DFHSTOR	444	SC64UHWM	2 GB 境界より上のユーザー・ストレージ最高水準点 (GUDSA)
DFHSTOR	445	SC64SGCT	2 GB 境界より上の共用ストレージ GET 数 (GCDSA または GUDSA)
DFHSTOR	446	SC64GSHR	2 GB 境界より上の GETMAIN 要求を使用して割り振られた共用ストレージ・バイト数 (GCDSA または GUDSA)
DFHSTOR	447	SC64FSHR	2 GB 境界より上の FREEMAIN 要求を使用して解放された共用ストレージ・バイト数
DFHSYNC	60	SPSYNCCT	同期点数
DFHSYNC	173	SYNCTIME	同期点経過時間
DFHSYNC	177	SRVSYWTT	CFDT サーバー同期点待ち時間
DFHSYNC	196	SYNCDLY	同期点遅延時間
DFHSYNC	199	OTSINDWT	OTS 未確定待ち時間
DFHTASK	7	USRDISPT	ユーザー・タスク・ディスパッチ時刻
DFHTASK	8	USRCPUT	ユーザー・タスク CPU 時間
DFHTASK	14	SUSPTIME	ユーザー・タスク一時停止時刻
DFHTASK	31	TRANNUM	タスク番号
DFHTASK	59	ICPUINCT	IC プット/開始カウント
DFHTASK	64	TASKFLAG	エラー・フラグ・フィールド
DFHTASK	65	ICSTACCT	CHANNEL オプションを指定したローカル START 要求数
DFHTASK	66	ICTOTCT	IC 合計カウント
DFHTASK	82	TRNGRPID	トランザクション・グループ ID
DFHTASK	97	NETUOWPX	発信端末またはシステムのネットワーク名
DFHTASK	98	NETUOWSX	発信元システム上の作業単位 ID
DFHTASK	102	DISPWTT	ユーザー・タスクのディスパッチ待ち時間
DFHTASK	109	TRANPRI	トランザクションの優先順位
DFHTASK	123	GNQDELAY	タスクのグローバル ENQ 遅延時間
DFHTASK	124	BRDGTRAN	3270 ブリッジ・トランザクション ID
DFHTASK	125	DSPDELAY	初回ディスパッチ遅延時間
DFHTASK	126	TCLDELAY	TRANCLASS が原因の初回ディスパッチ遅延時間
DFHTASK	127	MXTDELAY	MXT が原因の初回ディスパッチ遅延

表 13. 除外または組み込むことができるデータ・グループおよびフィールド (続き)

グループ名	フィールド ID	フィールド名	説明
DFHTASK	128	LMDELAY	ロック・マネージャー遅延時間
DFHTASK	129	ENQDELAY	タスク ENQ 遅延時間
DFHTASK	132	RMUOWID	リカバリー・マネージャー作業単位 ID
DFHTASK	163	FCTYNAME	トランザクション・ファシリティー名
DFHTASK	164	TRANFLAG	トランザクション・フラグ
DFHTASK	166	TCLSNAME	トランザクション・クラス名
DFHTASK	170	RMITIME	リソース・マネージャー・インターフェース経過時間
DFHTASK	171	RMISUSP	リソース・マネージャー・インターフェース中断時間
DFHTASK	181	WTEXWAIT	EXEC CICS WAIT EXTERNAL 待ち時間
DFHTASK	182	WTCEWAIT	EXEC CICS WAITCICS および WAIT EVENT 待ち時間
DFHTASK	183	ICDELAY	間隔制御遅延時間
DFHTASK	184	GVUPWAIT	ディスパッチ待ち時間
DFHTASK	190	RRMSURID	RRMS/MVS リカバリー単位 ID (URID)。
DFHTASK	191	RRMSWAIT	RRMS/MVS 待ち時間
DFHTASK	192	RQRWAIT	要求受信側待ち時間
DFHTASK	193	RQPWAIT	要求プロセッサー待ち時間
DFHTASK	194	OTSTID	OTS トランザクション ID (TID)
DFHTASK	195	RUNTRWTT	CICS BTS 実行プロセス/アクティビティー同期待ち時間
DFHTASK	247	DSCHMDLY	TCB モード変更遅延時間
DFHTASK	249	QRMODDLY	ユーザー・タスク QR TCB ディスパッチ待ち時間
DFHTASK	250	MXTOTDLY	CICS L8 および L9 モード・オープン TCB 遅延時間
DFHTASK	251	TCBATTCT	CICS TCB 接続カウント
DFHTASK	252	DSTCBHWM	ユーザー・タスクのピーク・オープン TCB 数
DFHTASK	253	JVMTIME	CICS JVM 経過時間
DFHTASK	254	JVMSUSP	CICS JVM 一時停止時間
DFHTASK	255	QRDISPT	ユーザー・タスク QR TCB ディスパッチ時間
DFHTASK	256	QRCPUT	ユーザー・タスク QR TCB CPU 時間
DFHTASK	257	MSDISPT	ユーザー・タスク MS TCB ディスパッチ時間
DFHTASK	258	MSCPUT	ユーザー・タスク MS TCB CPU 時間
DFHTASK	259	L8CPUT	ユーザー・タスク L8 TCB CPU 時間
DFHTASK	261	S8CPUT	ユーザー・タスク S8 TCB CPU 時間
DFHTASK	262	KY8DISPT	ユーザー・タスク・キー 8 TCB ディスパッチ時間
DFHTASK	263	KY8CPUT	ユーザー・タスク・キー 8 TCB CPU 時間
DFHTASK	264	KY9DISPT	ユーザー・タスク・キー 9 TCB ディスパッチ時間
DFHTASK	265	KY9CPUT	ユーザー・タスク・キー 9 TCB CPU 時間
DFHTASK	266	L9CPUT	ユーザー・タスク L9 TCB CPU 時間
DFHTASK	268	DSTCBMWT	ユーザー・タスク TCB ミスマッチ待ち時間
DFHTASK	269	RODISPT	ユーザー・タスク RO TCB ディスパッチ時間
DFHTASK	270	ROCPUT	ユーザー・タスク RO TCB CPU 時間
DFHTASK	271	X8CPUT	ユーザー・タスク X8 TCB CPU 時間
DFHTASK	272	X9CPUT	ユーザー・タスク X9 TCB CPU 時間

表 13. 除外または組み込むことができるデータ・グループおよびフィールド (続き)

グループ名	フィールド ID	フィールド名	説明
DFHTASK	273	JVMITIME	CICS JVM 初期設定の経過時間
DFHTASK	275	JVMRTIME	CICS JVM リセット経過時間
DFHTASK	279	DSMMSCWT	MVS ストレージ制約の遅延時間
DFHTASK	281	MAXSTDLY	CICS SSL TCB 遅延時間
DFHTASK	282	MAXXTDLY	CICS XP TCB 遅延時間
DFHTASK	283	MAXTTDLY	CICS JVM サーバー・スレッド TCB 遅延時間
DFHTASK	285	PTPWAIT	3270 ブリッジ・パートナー待ち時間
DFHTASK	345	ICSTACDL	すべてのローカル実行 START CHANNEL 要求のコンテナー内のデータの長さ
DFHTASK	346	ICSTRCCT	リモート・システムで実行されるインターバル制御機能 START CHANNEL 要求の数
DFHTASK	347	ICSTRCDL	すべてのリモート実行 START CHANNEL 要求のコンテナー内のデータの長さ
DFHTASK	348	ROMODDLY	ユーザー・タスク RO TCB ディスパッチ待ち時間
DFHTASK	349	SOMODDLY	ユーザー・タスク SO TCB ディスパッチ待ち時間
DFHTASK	400	T8CPUT	ユーザー・タスク T8 TCB CPU 時間
DFHTASK	401	JVMTHDWT	JVMSERVER スレッド待ち時間。Liberty JVM サーバーには適用されません。
DFHTASK	429	DSAPTHWT	ディスパッチャー割り振り pthread 待ち時間
DFHTASK	430	CECMCHTP	CEC マシン・タイプ
DFHTASK	431	CECMDLID	CEC 型式番号
DFHTASK	432	LPARNAME	CICS 領域が実行されるプロセッサー上の論理区画 (LPAR) の名前 (EBCDIC 形式)。
DFHTASK	433	MAXTASKS	ユーザー・タスク接続時の CICS 領域の MXT または MAXTASKS
DFHTASK	434	CURTASKS	ユーザー・タスク接続時の CICS 領域内の現行タスク数
DFHTASK	436	CPUTONCP	標準プロセッサーのタスク・プロセッサー時間
DFHTASK	437	OFFLCPUT	専用プロセッサーへのオフロードに適格なタスク・プロセッサー時間
DFHTASK	451	ACAPPLNM	アプリケーション・コンテキスト・データに含まれる 64 文字のアプリケーション名
DFHTASK	452	ACPLATNM	アプリケーション・コンテキスト・データに含まれる 64 文字のプラットフォーム名
DFHTASK	453	ACMAJVER	4 バイトのバイナリー値として表される、アプリケーション・コンテキスト・データに含まれるアプリケーションのメジャー・バージョン
DFHTASK	454	ACMINVER	4 バイトのバイナリー値として表される、アプリケーション・コンテキスト・データに含まれるアプリケーションのマイナー・バージョン
DFHTASK	455	ACMICVER	4 バイトのバイナリー値として表される、アプリケーション・コンテキスト・データに含まれるアプリケーションのマイクロ・バージョン
DFHTASK	456	ACOPERNM	アプリケーション・コンテキスト・データに含まれる 64 文字の操作名

表 *13.* 除外または組み込むことができるデータ・グループおよびフィールド (続き)

グループ名	フィールド **ID**	フィールド名	説明
DFHTASK	470	ASTOTCT	ユーザー・タスクによって発行された EXEC CICS 非同期 API コマンドの総数。 **RUN TRANSID** コマンド、**FETCH CHILD** コマンド、および **FETCH ANY** コマンドが含まれます。
DFHTASK	471	ASRUNCT	ユーザー・タスクによって発行された **EXEC CICS RUN TRANSID** コマンドの数
DFHTASK	472	ASFTCHCT	ユーザー・タスクによって発行された **EXEC CICS FETCH CHILD** コマンドおよび **EXEC CICS FETCH ANY** コマンドの数
DFHTASK	473	ASFREECT	ユーザー・タスクによって発行された **EXEC CICS FREE CHILD** コマンドの数
DFHTASK	475	ASFTCHWT	発行された **EXEC CICS FETCH CHILD** コマンドまたは **EXEC CICS FETCH ANY** コマンドが完了していないために、ユーザー・タスクがその子タスクを待っている間に経過した時間
DFHTASK	476	ASRNATWT	非同期サービス・ドメインが管理する非同期子タスクの制限によって、ユーザー・タスクが遅延した時間
DFHTEMP	11	TSIOWTT	TS 入出力待ち時間
DFHTEMP	44	TSGETCT	TS 取得カウント
DFHTEMP	46	TSPUTACT	TS プット補助カウント
DFHTEMP	47	TSPUTMCT	TS プット・メイン・カウント
DFHTEMP	92	TSTOTCT	TS 合計カウント
DFHTEMP	178	TSSHWAIT	共用 TS 入出力待ち時間
DFHTEMP	460	TSGETSCT	TS GET 共用数
DFHTEMP	461	TSPUTSCT	TS PUT 共用数
DFHTERM	9	TCIOWTT	TC 入出力待ち時間
DFHTERM	34	TCMSGIN1	TC 基本機能入力メッセージ
DFHTERM	35	TCMSGOU1	TC 基本機能出力メッセージ
DFHTERM	67	TCMSGIN2	TC 代替機能入力メッセージ
DFHTERM	68	TCMSGOU2	TC 代替機能出力メッセージ
DFHTERM	69	TCALLOCT	TC 割り振り数
DFHTERM	83	TCCHRIN1	TC 基本機能文字入力
DFHTERM	84	TCCHROU1	TC 基本機能文字出力
DFHTERM	85	TCCHRIN2	TC 代替機能文字入力
DFHTERM	86	TCCHROU2	TC 代替機能文字出力
DFHTERM	100	IRIOWTT	IR 入出力待ち時間
DFHTERM	111	LUNAME	z/OS Communications Server 端末 LU 名
DFHTERM	133	LU61WTT	TC 入出力待ち時間 - LU6.1
DFHTERM	134	LU62WTT	TC 入出力待ち時間 - LU6.2
DFHTERM	135	TCM62IN2	TC 代替機能入力メッセージ - LU6.2
DFHTERM	136	TCM62OU2	TC 代替機能出力メッセージ - LU6.2
DFHTERM	137	TCC62IN2	TC 代替機能文字入力 - LU6.2
DFHTERM	138	TCC62OU2	TC 代替機能文字出力 - LU6.2
DFHTERM	165	TERMINFO	端末情報
DFHTERM	169	TERMCNM	端末セッション接続名
DFHTERM	197	NETID	ネットワーク修飾名のネットワーク ID
DFHTERM	198	RLUNAME	ネットワーク修飾名のネットワーク名

表 13. 除外または組み込むことができるデータ・グループおよびフィールド (続き)

グループ名	フィールド ID	フィールド名	説明
DFHTERM	343	TCALWTT	MRO、LU6.1、および LU6.2 セッション割り振り待ち時間
DFHWEBB	224	WBREADCT	Web 読み取り要求数
DFHWEBB	225	WBWRITCT	Web 書き込み要求数
DFHWEBB	231	WBRCVCT	Web 受信要求数
DFHWEBB	232	WBCHRIN	Web 受信文字数
DFHWEBB	233	WBSENDCT	Web 送信要求数
DFHWEBB	234	WBCHROUT	Web 送信文字数
DFHWEBB	235	WBTOTCT	Web 合計要求数
DFHWEBB	236	WBREPRCT	Web ヘッダーまたは formfield 読み取り要求数
DFHWEBB	237	WBREPWCT	Web ヘッダーまたは formfield 書き込み要求数
DFHWEBB	238	WBEXTRCT	Web 抽出要求数
DFHWEBB	239	WBBRWCT	Web ブラウズ要求数
DFHWEBB	331	WBREDOCT	Web ヘッダー読み取り要求数、クライアント
DFHWEBB	332	WBWRTOCT	Web ヘッダー書き込み要求数、クライアント
DFHWEBB	333	WBRCVIN1	Web クライアント Receive または Converse 要求数
DFHWEBB	334	WBCHRIN1	Web 受信文字数、クライアント
DFHWEBB	335	WBSNDOU1	Web クライアント Send または Converse 要求数
DFHWEBB	336	WBCHROU1	Web 送信文字数、クライアント
DFHWEBB	337	WBPARSCT	Web クライアント Parse (構文解析) URL 要求数
DFHWEBB	338	WBRWOCT	Web クライアント・ブラウズ要求数
DFHWEBB	340	WBIWBSCT	EXEC CICS INVOKE SERVICE または WEBSERVICE 要求数
DFHWEBB	341	WBREPRDL	Web ヘッダーまたは formfield 読み取りバイト数
DFHWEBB	342	WBREPWDL	Web ヘッダーまたは formfield 書き込みバイト数
DFHWEBB	380	WBURIMNM	URIMAP リソース名
DFHWEBB	381	WBPIPLNM	PIPELINE リソース名
DFHWEBB	382	WBATMSNM	ATOMSERVICE リソース名
DFHWEBB	383	WBSVCENM	WEBSERVICE リソース名
DFHWEBB	384	WBSVOPNM	WEBSERVICE オペレーション名
DFHWEBB	385	WBPROGNM	PROGRAM リソース名
DFHWEBB	386	WBSFCRCT	SOAPFAULT 作成要求数
DFHWEBB	387	WBSFTOTCT	SOAPFAULT 要求数の合計
DFHWEBB	388	WBISSFCT	EXEC CICS INVOKE SERVICE SOAP 障害
DFHWEBB	390	WBSREQBL	SOAP 要求本体の長さ
DFHWEBB	392	WBSRSPBL	SOAP 応答本体の長さ
DFHWEBB	412	MLXSSTDL	変換された着信ドキュメントの長さの合計
DFHWEBB	413	MLXMLTCT	EXEC CICS TRANSFORM 要求数
DFHWEBB	420	WSACBLCT	WS アドレス指定コンテキスト・ビルド要求数
DFHWEBB	421	WSACGTCT	WS アドレス指定コンテキスト GET 要求数
DFHWEBB	422	WSAEPCCT	WS アドレス指定エンドポイント参照 (EPR) 作成要求数
DFHWEBB	423	WSATOTCT	WS アドレス指定合計要求数

表 *13.* 除外または組み込むことができるデータ・グループおよびフィールド (続き)

グループ名	フィールド **ID**	フィールド名	説明
DFHWEBB	424	WBJSNRQL	JSON Web サービス・アプリケーションの場合、JSON メッセージ要求の長さです。
DFHWEBB	425	WBJSNRPL	JSON Web サービス・アプリケーションの場合、JSON メッセージ応答の長さです。

INCLUDE=(*m1*[,...])

モニター機能によって 1 つ以上の CICS フィールドが報告されるようにするには、このオペランドをコーディングします。デフォルトでは、記録されたすべてのパフォーマンス・クラス・フィールドが報告されるため、このオペランドは、同じマクロ内に EXCLUDE オペランドをコーディングした場合にのみ、適用されます。

含めるためにこのオペランドにコーディングされる対象となるフィールドは、除外の対象となるフィールドと同じです。 (EXCLUDE オペランドの説明を参照してください。) 各フィールドには、それに関連付けられた数値フィールド ID があります。フィールドを含めるには、*m1* をフィールドの数値 ID でコーディングします。複数の数値 ID をコーディングすることができます。

注**:** 先行ゼロを数値 ID にコーディングしないでください。除外の対象でない (デフォルトで含められる) フィールドの数値 ID はコーディングしないでください。

EXCLUDE オペランドが常に最初に受け付けられます。 INCLUDE オペランドがコーディングされた場合は、次にその効果の一部がオーバーライドされます。例えば、以下のコードは、ファイル制御の PUT の収集と報告、およびファイル制御要求の合計数は保護されますが、ファイル制御ブラウズ数およびその他のファイル制御フィールドは除外されます。

```
EXCLUDE=DFHFILE,
INCLUDE=(37,93)
```

フィールドの大部分を除外したいが、いくつかのフィールドは含めたい場合は、以下の例に示すようにオペランドを使用することができます。

```
EXCLUDE=ALL,
INCLUDE=(DFHTERM,97,98)
```

この例は、除外するすべてのフィールドを個々にコーディングするよりも簡単です。

CICSTS55.SDFHSAMP では、以下のサンプルのモニター管理テーブルを提供しています。

- 端末専有領域 (TOR) 用: DFHMCTT$
- アプリケーション専有領域 (AOR) 用: DFHMCTA$
- アプリケーション専有領域 (AOR) (DBCTL 付き) 用: DFHMCTD$
- ファイル専有領域 (FOR) 用: DFHMCTF$

これらのサンプルでは、CICS が SMF に書き込むデータのボリュームを小さくするために、EXCLUDE および INCLUDE オペランドを使用して、パフォーマンス・クラス・レコードのサイズを小さくする方法が示されています。

DFHMCT の例

以下の例は、DFHMCT TYPE=INITIAL マクロ、DFHMCT TYPE=RECORD マクロ、および DFHMCT TYPE=EMP マクロを使用して、モニター管理テーブルの制御セクションを作成し、ユーザー・イベントのモニター・ポイントを指定し、システム定義フィールドを除外します。

図 8 は、DFHMCT TYPE=INITIAL マクロでの制御ステートメントのコーディングを示しています。この例では、接尾部 C2 が指定されているため、このモニター管理テーブルの名前は DFHMCTC2 になります。その他のパラメーターは、デフォルト値に設定されます。

```
DFHMCT   TYPE=INITIAL,SUFFIX=C2,COMPRESS=YES,       *
        APPLNAME=NO,RMI=NO,FILE=8,DPL=0,TSQUEUE=8
DFHMCT   TYPE=FINAL
         END
```

図 *8. 例: DFHMCT TYPE=INITIAL*

図 9 は、2 つのユーザー・イベント・モニター・ポイント (EMP) のモニター管理テーブル (MCT) を作成するためのコーディングを示しています。

```
DFHMCT   TYPE=INITIAL,SUFFIX=MF,COMPRESS=YES
DFHMCT   TYPE=EMP,                              *
         ID=180,                                *
         CLASS=PERFORM,                         *
         PERFORM=(SCLOCK(1),ADDCNT(2,1))
DFHMCT   TYPE=EMP,                              *
         ID=181,                                *
         CLASS=PERFORM,                         *
         PERFORM=PCLOCK(1)
DFHMCT   TYPE=FINAL
         END
```

図 *9. 例: DFHMCT TYPE=EMP*、ユーザー・イベント・モニター・ポイント

439 ページの図 10 は、パフォーマンス・データ・レコードから特定のデータ・フィールドを除外するモニター管理テーブル (MCT) を作成するためのコーディングを示しています。特定の CICS 機能を使用するアプリケーションがない場合は、それらの機能に関連するパフォーマンス・データ・グループおよびフィールドを安全に除外することができます。この例は、これらの機能のフィールドを除外するための DFHMCT TYPE=RECORD マクロを示しています。

フロントエンド・プログラミング・インターフェース **(FEPI)**

```
DFHMCT   TYPE=RECORD,CLASS=PERFORM,             *
         EXCLUDE=(DFHFEPI)
```

ビジネス・トランザクション・サービス **(BTS)**

```
DFHMCT   TYPE=RECORD,CLASS=PERFORM,             *
         EXCLUDE=(DFHCBTS,195,196)
```

Java アプリケーションのサポート

```
        DFHMCT   TYPE=RECORD,CLASS=PERFORM,                *
                 EXCLUDE=(253,254,273,275)
```

```
DFHMCT   TYPE=INITIAL,SUFFIX=CB,COMPRESS=YES,       *
        APPLNAME=NO,RMI=NO,FILE=16,TSQUEUE=12
DFHMCT   TYPE=RECORD,CLASS=PERFORM,                 *
        EXCLUDE=(DFHFEPI)
DFHMCT   TYPE=RECORD,CLASS=PERFORM,                 *
        EXCLUDE=(DFHCBTS,195,196)
DFHMCT   TYPE=RECORD,CLASS=PERFORM,                 *
        EXCLUDE=(253,254,273,275)
DFHMCT   TYPE=FINAL
         END
```

図 *10. 例: DFHMCT TYPE=RECORD*、フィールドの除外

PLT: プログラム・リスト・テーブル

プログラム・リスト・テーブル (PLT) は、CICS の始動時およびシャットダウン時に実行するプログラムや、まとめて使用可能/使用不可にするプログラムのグループを指定します。

以下の 1 つ以上の理由により、複数の PLT をコーディングすることが必要な場合があります。

- CICS の始動の 2 番目および 3 番目の初期設定ステージで除外するプログラムのリストを指定するため。初期設定ステージの詳細については、初期化後処理 (PLTPI) プログラムの使用 を参照してください。初期設定ステージでのプログラムの使用に対する制限に関するプログラミング情報については、初期設定プログラムの作成 を参照してください。選択したリストは、初期設定時に PLTPI=*name* システム初期設定パラメーターで指定する必要があります。ここで、*name* は、必要なプログラムのリストが入った PLT の 1 文字または 2 文字の接尾部、あるいはフルネームです。

  便宜上、初期設定中の実行対象として選択したプログラムのリストを、「PLTPI」リストと呼びます。
- 制御されたシャットダウンの最初および 2 番目の静止ステージで実行するプログラムのリストを指定するため。選択したリストは、初期設定時に PLTSD=*name* システム初期設定パラメーターで指定する必要があります。ここで、*name* は、必要なプログラムのリストが入った PLT の 1 文字または 2 文字の接尾部、あるいはフルネームです。

  PLTSD システム初期設定パラメーターで指定された PLT は、CEMT PERFORM SHUTDOWN コマンドの PLT オプションによってシャットダウン時にオーバーライドすることができます。

  便宜上、シャットダウン中の実行対象として選択したプログラムのリストを、「PLTSD」リストと呼びます。

- マスター端末の ENABLE コマンドまたは DISABLE コマンドによってグループとして使用可能/使用不可にするプログラムのリストを指定するため。この PLT の使用は、マスター端末オペレーターが、プログラムごとに個別にコマンドを使用しなくても、プログラムのセットを 1 回のコマンドで使用可能あるいは使用不可にできることを意味します。

上記の目的のために、以下の条件で任意の数の PLT を生成することができます。

1. 各 PLT の名前は固有でなければならない
2. PLT で指定された各プログラムが、CSD ファイル内にプログラム・リソース定義エントリーを持つか、自動インストールに対応している (つまり、プログラムの自動インストール用に適切なシステム初期設定パラメーターが指定されている)。

   注: CSD 内で PLT をプログラムとして定義する必要はありません。

最初のフェーズの PLT プログラムは、DFHRPL 連結データ・セットに入れる必要がありますが、2 番目のフェーズの PLT プログラムは、動的 LIBRARY 連結に入れることができます。ただし、CICS は、フェーズ 1 PLTPI プログラムがまだインストールされていない場合は、その LPA をスキャンします。

PLT エントリーを定義する場合、以下のマクロが使用可能です。

- 制御セクション: DFHPLT TYPE=INITIAL
- プログラム・リスト・テーブル内のエントリー: DFHPLT TYPE=ENTRY
- プログラム・リスト・テーブルの終了: DFHPLT TYPE=FINAL (TYPE=FINAL (テーブルの終わり) を参照)

注: PLT をアセンブルする必要はありません。CICS が PLT のソース・コードを処理します。ソース・コードが実行時に CICS で使用可能であることを確認する必要があります。これには、ソースで参照されるすべてのコピー・メンバーが含まれます。そのためには、IPL parmlib 連結の一部である parmlib member メンバーにソースを置くか、PLT ソースの場所を指定した DD カードを CICS JCL に追加します。DD ステートメントは `//DFHTABLE DD DSN=pds name,DISP=SHR` という形式にする必要があります。

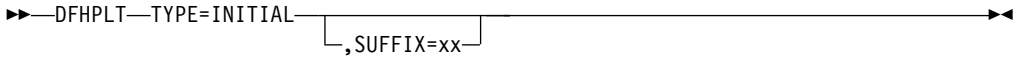
PARMLIB または DFHTABLE 連結内のデータ・セットに対する読み取り権限を CICS が持っていなければなりません。

PLT をアセンブルしようとすると、DFHPLT マクロから戻りコード 8 が返され、アセンブリーは行われません。

PLT に含まれるコピー・メンバーには、DFHPLT TYPE=ENTRY ステートメント、コピー・ステートメント、またはコメントのみが含まれている必要があります。

制御セクション: DFHPLT TYPE=INITIAL

DFHPLT TYPE=INITIAL マクロは、定義する PLT のエントリー・ポイントおよび開始アドレスを設定します。



SUFFIX オペランドの使用など、TYPE=INITIAL マクロに関する一般情報については、TYPE=INITIAL (制御セクション)を参照してください。

TYPE=INITIAL
PLT 制御セクションを生成します。

CSD ファイルは、生成される PLT ごとにプログラム・エントリーを定義する必要があるので注意してください。

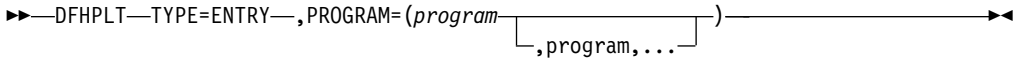
SUFFIX=xx
このパラメーターは廃止されたため、無視されます。CICS は、PARMLIB または DFHTABLE メンバーの名前の中の DFHPLT に続く文字からテーブルの接尾部を決定します。例えば、メンバー名が DFHPLTZZ の場合、接尾部は ZZ になります。

注: PLT 接尾部は、以下によって参照されます。

- CEMT {INQUIREvSET} PROGRAM CLASS(*suffix*)
- CEMT または EXEC CICS PERFORM SHUTDOWN PLT(*suffix*)
- システム初期設定パラメーターの PLTPI キーワードおよび PLTSD キーワード

プログラム・リスト・テーブル内のエントリー: DFHPLT TYPE=ENTRY

プログラム・リスト・テーブル (PLT) 内のエントリーを指定するには、DFHPLT TYPE=ENTRY マクロを使用します。



TYPE=ENTRY
1 つ以上のプログラム名をこのテーブルにリストすることを示します。

注: ここで示すように、PROGRAM=DFHDELIM エントリーを指定する場合にも TYPE=ENTRY マクロが必要です。

PROGRAM=program
最大 8 文字のプログラム名を指定して、これをコーディングします。各プログラムは、CSD ファイル内に定義を持つか、自動インストールに対応している必要があります (つまり、プログラムの自動インストール用に適切なシステム初期設定パラメーターが指定されている必要があります)。 DFHDELIM ステートメントの前の未定義のプログラムが、システム自動インストールされます。

PLTPI リストおよび PLTSD リストの場合は、初期プログラムのみを指定する必要があります。初期プログラムによってリンクされているその他のプログラムをリストする必要はありません (ただし、定義されているか、自動インストールに対応している必要はあります)。初期設定中の PLT プログラムの使用に対する制限に関するプログラミング情報については、初期設定プログラムの作成 を参照してください。

PROGRAM=DFHDELIM

PLTPI あるいは PLTSD の最初または 2 番目の受け渡しで実行するためにプログラムを区切るには、これをコーディングします。DFHDELIM エントリーはプログラムではありません。区切り文字としての機能のみを提供します。

次の点に注意してください。

- PLTPI 内で PROGRAM=DFHDELIM エントリーの前にリストされているプログラムは、初期設定の 2 番目のステージで実行されます。これらは、リカバリー時に必要なユーザー出口プログラムを使用可能にするためのものです。ユーザー出口プログラムは、CSD ファイル内に定義します。そうしないと、CICS が CICS 初期設定の完了後に、(例えば、**EXEC CICS DISABLE** コマンドで) それらのプログラムにアクセスできない可能性があります。ただし、初期設定の 2 番目のステージ中は、RDO によって定義されたプロパティーには効果がないので注意してください。
- PLTPI 内で PROGRAM=DFHDELIM エントリーの後にリストされているプログラムは、初期設定の 3 番目のステージで実行されます。これらのプログラムがユーザー出口を使用可能にするために使用された場合、そのユーザー出口プログラムも、CSD ファイルに定義するか、自動インストールに対応している必要があります。
- PLTSD 内で PROGRAM=DFHDELIM エントリーの前にリストされているプログラムは、シャットダウンの最初の静止ステージで実行されます。
- PLTSD 内で PROGRAM=DFHDELIM エントリーの後にリストされているプログラムは、シャットダウンの 2 番目の静止ステージで実行されます。

初期設定の 2 番目のステージおよび静止の 2 番目のステージの PLT プログラムには、プログラム・リソース定義は不要です。定義されていない場合、(プログラム自動インストールのシステム初期設定パラメーターに関係なく) システム自動インストールされます。これは、自動インストール出口を呼び出して定義を変更することができないことを意味します。プログラムは、以下の属性を指定して定義されます。

```
LANGUAGE(ASSEMBLER)  STATUS(ENABLED)  CEDF(NO)
DATALOCATION(BELOW)  EXECKEY(CICS)
EXECUTIONSET(FULLAPI)
```

その結果、システム自動インストールされたプログラムには、デフォルトの CONCURRENCY 設定である QUASIRENT と、デフォルトの API 設定である CICSAPI が設定されることになります。

- API 属性の OPENAPI 値を使用して定義された、あるいは XPLINK コンパイラー・オプションを指定して C または C++ プログラムでコンパイルされたスレッド・セーフ PLT プログラムの場合は、適切なリソース定義を提供してください。あるいは、Language Environment 規格適合プログラムの場合は、CICSVAR ランタイム・オプションを使用して、適切な CONCURRENCY 値および API 値を設定します。 言語環境プログラムのランタイム・オプションの定義 を参照してください。

初期設定の 3 番目のステージおよび静止の最初のステージの PLT プログラムは、プログラム自動インストールを使用することで、プログラム自動インストールのシステム初期設定パラメーターに応じて定義することができます。プログラム自動インストールが使用されない場合、これらのプログラムは、CSD ファイル内にリソース定義が必要です。

DFHPLT の例

この例を使用して、PLT の生成に必要なコーディングを明らかにすることができます。

図 11 および 444 ページの図 12 は、PLT の生成に必要なコーディングを示しています。

```
*
* LIST OF PROGRAMS TO BE EXECUTED SEQUENTIALLY DURING SYSTEM
* INITIALIZATION.
* REQUIRED SYSTEM INITIALIZATION PARAMETER: PLTPI=I1
*
   DFHPLT TYPE=INITIAL,SUFFIX=I1
*
*  The following programs are run in the first pass of PLTPI
*
   DFHPLT TYPE=ENTRY,PROGRAM=TRAQA  EXECUTED DURING 2ND INIT. PHASE
   DFHPLT TYPE=ENTRY,PROGRAM=TRAQB  (PROGRAMS SHOULD ALSO BE DEFINED
   DFHPLT TYPE=ENTRY,PROGRAM=TRAQC  BY RDO)
*
   DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
*
*
*  The following programs are run in the second pass of PLTPI
*
   DFHPLT TYPE=ENTRY,PROGRAM=TRASA  EXECUTED DURING 3RD INIT. PHASE
   DFHPLT TYPE=ENTRY,PROGRAM=TRASB  (PROGRAMS MUST ALSO BE DEFINED
   DFHPLT TYPE=ENTRY,PROGRAM=TRASC  BY RDO)
   DFHPLT TYPE=FINAL
*
   END
```

図 *11. PLTPI* プログラム・リスト・テーブル*:* 例

```
*
*
* LIST OF PROGRAMS TO BE EXECUTED SEQUENTIALLY DURING SYSTEM
* TERMINATION
* REQUIRED SYSTEM INITIALIZATION PARAMETER: PLTSD=T1
*
   DFHPLT TYPE=INITIAL,SUFFIX=T1
*
*  The following programs are run in the 1st pass of PLTSD
*
*
   DFHPLT TYPE=ENTRY,PROGRAM=TRARA  EXECUTED DURING 1st QUIESCE PHASE
   DFHPLT TYPE=ENTRY,PROGRAM=TRARB  (PROGRAMS MUST ALSO BE DEFINED
   DFHPLT TYPE=ENTRY,PROGRAM=TRARC  BY RDO)
*
   DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
*
*
*  The following programs are run in the 2nd pass of PLTSD
*
   DFHPLT TYPE=ENTRY,PROGRAM=TRAFA  EXECUTED DURING 2nd QUIESCE PHASE
   DFHPLT TYPE=ENTRY,PROGRAM=TRAFB  (PROGRAMS MUST ALSO BE DEFINED
*                                   BY RDO)
   DFHPLT TYPE=FINAL
*
   END
```

図 *12. PLTSD* プログラム・リスト・テーブル*:* 例

RST: リカバリー可能サービス・テーブル

用語の注記**:** DBCTL は、CICS-IMS/ESA® DBCTL (データベース制御) インターフェースを意味します。

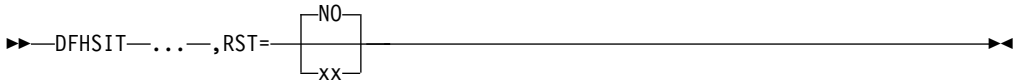
リカバリー可能サービス・テーブル (RST) は、CICS DBCTL XRF (拡張リカバリー機能) サポートに使用されます。これには、DBCTL 構成に関する記述が含まれます。DBCTL の障害が検出されると、アクティブ CICS システムは、RST と MVS サブシステム VERIFY を一緒に使用して、適切な代替 DBCTL サブシステムがないかを判別します。代替 CICS システムは、RST を使用して DBCTL サブシステムの存在を確認します。セキュリティー上の理由から、RST は APF の使用が許可されたライブラリーにリンク・エディットする必要があります。RST は、CICS 中核の一部としてはロードされません。

リカバリー可能サービス・テーブルでは、以下のマクロをコーディングすることができます。

- DFHRST TYPE=INITIAL は、制御セクションを設定します。
- DFHRST TYPE=RSE は、リカバリー可能サービス・エレメント (RSE) の開始を指定します。RSE は、同等の DBCTL サブシステムの ID、これらの DBCTL サブシステムに関連付けられた CICS アプリケーション ID、DBCTL サブシステムを使用する CICS システムのアプリケーション ID のセット (空ではない) から構成されます。
- DFHRST TYPE=SUBSYS は、RSE 内の DBCTL サブシステムの 1 つを指定します。

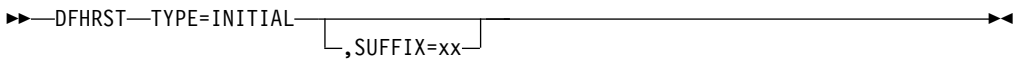
- DFHRST TYPE=FINAL は、RST を終了します (TYPE=FINAL (テーブルの終わり)を参照)。

システムが使用する RST は、以下のシステム初期設定パラメーターを使用してシステム初期設定テーブルで定義されている必要があります。



制御セクション: DFHRST TYPE=INITIAL

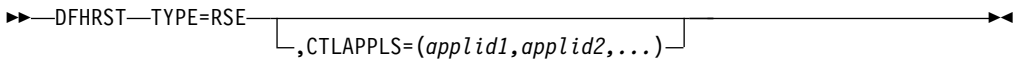
DFHRST TYPE=INITIAL マクロは、リカバリー可能サービス・テーブルの制御セクションを設定します。



SUFFIX オペランドの使用など、TYPE=INITIAL マクロに関する一般情報については、TYPE=INITIAL (制御セクション)を参照してください。

リカバリー可能サービス・エレメント: DFHRST TYPE=RSE

DFHRST TYPE=RSE マクロには、リカバリー可能サービス・エレメントに関する情報、およびリカバリー可能サービス・エレメント内で DBCTL サブシステムに接続する予定になっている CICS システムに関する情報が含まれます。



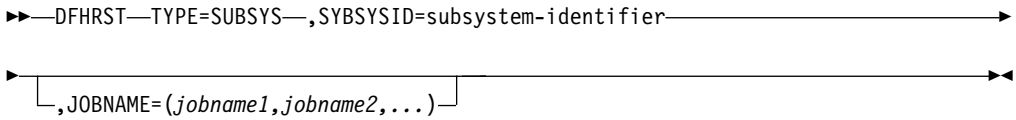
TYPE=RSE
同等の DBCTL システムの ID のセットの開始を示します。同等の DBCTL サブシステムには、同じデータベース (DB) 名があります。

CTLAPPLS=(applid1,applid2,....)
RSE 内で DBCTL サブシステムを再始動することが許可されている CICS システムのアプリケーション ID を指定します。

DBCTL サブシステム: DFHRST TYPE=SUBSYS

DFHRST TYPE=SUBSYS マクロには、RSE 内の特定の DBCTL サブシステムに関する情報が含まれます。

RSE 内のサブシステムごとに、DFHRST TYPE=SUBSYS マクロがなければなりません。



TYPE=SUBSYS
RSE 内の DBCTL サブシステムの 1 つを定義します。

SUBSYSID=subsystem-identifier
DBCTL サブシステム ID の 4 文字の名前を指定して、これをコーディングします。この ID は、RST 内で固有でなければなりません。

JOBNAME=(jobname1,jobname2,....)
SUBSYSID パラメーター・ステートメントで識別された DBCTL サブシステムに関連付けられている **MVS JOBNAME(S)** を指定して、これをコーディングします。これは、セキュリティー検査の形式です。CICS が DBCTL サブシステムを取り消す必要がある場合、それが許可されるのは、DBCTL サブシステムに関連付けられた適切な MVS ジョブ名が、**JOBNAME** パラメーターで指定されたジョブ名のいずれかである場合のみです。

DFHRST の例

この例を使用して、RST の生成に必要なコーディングを明らかにすることができます。

図 13 は、RST の生成に必要なコーディングを示しています。

```
DFHRST   TYPE=INITIAL
         ,SUFFIX=K1
DFHRST   TYPE=RSE
         ,CTLAPPLS=(applid1,applid2,applid3)
DFHRST   TYPE=SUBSYS
         ,SUBSYSID=CTL1
         ,JOBNAME=(job1,job2,job3,job4)
DFHRST   TYPE=SUBSYS
         ,SUBSYSID=CTL2
         ,JOBNAME=(job5,job6,job7,job8)
DFHRST   TYPE=FINAL
END
```

図 *13.* リカバリー可能サービス・テーブル*:* 例

SRT: システム・リカバリー・テーブル

システム・リカバリー・テーブル (SRT) には、CICS が代行受信する異常終了コードのリストが含まれます。代行受信すると、CICS は、問題を起こしているタスクを異常終了させることで作動可能な状態を保持しようとします。

独自のリカバリー・プログラムを作成することで、デフォルトのリカバリー・アクションを変更することができます。これを行うには、システム・リカバリー・プログラム (SRP) 内で XSRAB グローバル・ユーザー出口ルーチン点を使用します。(XSRAB 出口に関するプログラミング情報については、システム・リカバリー・プログラム出口 XSRAB を参照してください。)

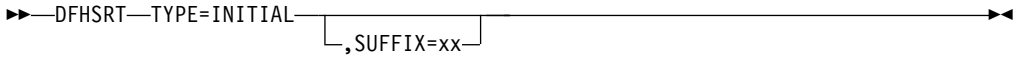
リカバリーが試行されるのは、異常終了の発生時に (システム・タスクではなく) ユーザー・タスクが制御を持っている場合であるので注意してください。

システム・リカバリー・テーブルでは、以下のマクロをコーディングすることができます。

- DFHSRT TYPE=INITIAL は、制御セクションを設定します
- DFHSRT TYPE=SYSTEM|USER は、処理される異常終了コードを指定します
- DFHSRT TYPE=FINAL は、SRT を終了します (TYPE=FINAL (テーブルの終わり) を参照)

制御セクション: DFHSRT TYPE=INITIAL

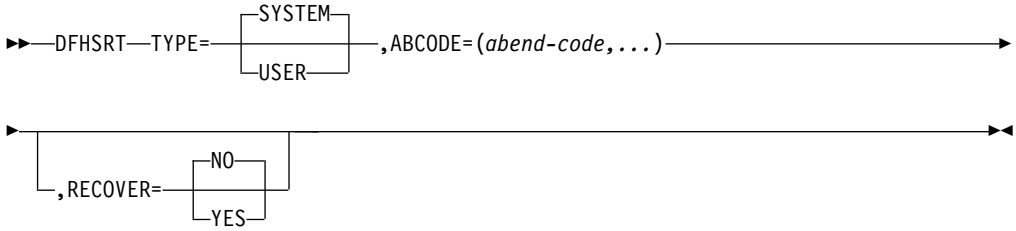
DFHSRT TYPE=INITIAL マクロは、システム・リカバリー・テーブルの制御セクションを生成します。



SUFFIX オペランドの使用など、TYPE=INITIAL マクロに関する一般情報については、TYPE=INITIAL (制御セクション)を参照してください。

異常終了コード: DFHSRT TYPE=SYSTEM|USER

DFHSRT TYPE=SYSTEM マクロおよび TYPE=USER マクロは、CICS によって代行受信されるシステムおよびユーザーの異常終了コードを指定します。



TYPE={SYSTEM|USER}
代行受信する異常終了コードのタイプを示します。

SYSTEM
異常終了コードは、オペレーティング・システム異常終了コードで、MVS S*xxx* 異常終了コードに対応します。

USER
異常終了コードは、ユーザー (CICS を含む) 異常終了コードで、MVS U*nnnn* 異常終了コードに対応します。

ABCODE=(abend-code,...)
代行受信する異常終了コード (複数可) を指定して、これをコーディングします。単一の異常終了コードを指定する場合、括弧は不要です。

TYPE=SYSTEM をコーディングする場合、異常終了コードは 3 桁の 16 進数字 (*xxx*) でなければなりません。これは、MVS システム異常終了コード S*xxx* を表します。

TYPE=USER をコーディングする場合、この異常終了コードは、10 進数 (*nnnn*) でなければなりません。これは、 MVS 異常終了コード U*nnnn* のユーザー部分を表します。通常これは、CICS が異常終了を試行する前に発行される CICS メッセージと同じ数値です。

RECOVER={YES|NO}

SRT にコードを追加するか、SRT からコードを除去するかを指定します。

YES

指定したコードを SRT に追加するには、これをコーディングします。

NO 指定したコードを SRT から除去するには、これをコーディングします。

注:

1. CICS は、以下の異常終了コードを自動的に代行受信し、リカバリーを試行します。

   ```
   001,002,013,020,025,026,030,032,033,034,035,
   036,037,03A,03B,03D,052,053,067,0D3,0D4,0D5,
   0D6,0D7,0D8,0E0,0F3,100,113,137,202,213,214,
   237,283,285,313,314,337,400,413,437,513,514,
   613,614,637,713,714,737,813,837,913,A13,A14,
   B13,B14,B37,D23,D37,E37
   ```

   異常終了コード 0F3 は、さまざまなマシン・チェック条件をカバーしています。また、マルチプロセッサーの実行中にのみ発生する可能性がある代替プロセッサー再試行条件もカバーしています。CICS 提供のリカバリー・コードは、命令障害マシン・チェックが永続的ではないという前提に基づいて、そこからのリカバリーを試行します。また、代替プロセッサー再試行条件からのリカバリーも試行します。

   以下のようにシステム・リカバリー・テーブル (SRT) をコーディングした場合、CICS は、これらの標準異常終了コードからのリカバリーを試行します。

   ```
   DFHSRT  TYPE=INITIAL
   DFHSRT  TYPE=FINAL
   END
   ```

   標準コードを個別にリストする必要はありません。

2. CICS が他のエラーを処理するようにしたい場合は、以下のように SRT をコーディングすることができます。

   ```
   DFHSRT  TYPE=INITIAL
   DFHSRT  TYPE=SYSTEM,or USER,
           ABCODE=(user or system codes),
           RECOVER=YES
   DFHSRT  TYPE=FINAL
   END
   ```

3. 1 つ以上の標準異常終了コードが発生した後に CICS がリカバリーを試行しないようにしたい場合は、**RECOVER=NO** を使用するか、**RECOVER** パラメーターを使用せずにコードを指定します。
4. 異常終了の発生時に (システム・タスクではなく) ユーザー・タスクが制御を持っている場合、CICS はリカバリーを試行します。

DFHSRT の例

この例を使用して、SRT の生成に必要なコーディングを明らかにすることができます。

449 ページの図 14 は、SRT の生成に必要なコーディングを示しています。

```
DFHSRT TYPE=INITIAL,            *
       SUFFIX=K1
DFHSRT TYPE=SYSTEM,             *
       ABCODE=777,              *
       RECOVER=YES
DFHSRT TYPE=USER,
       ABCODE=(888,999),        *
       RECOVER=YES
DFHSRT TYPE=USER,               *
       ABCODE=020
DFHSRT TYPE=FINAL
END
```

図 *14.* システム・リカバリー・テーブル*:* 例

TCT: 端末管理テーブル

論理装置コードをサポートする SNA 論理装置 (LU) および BSAM によって接続された順次装置を定義するには、DFHTCT マクロを使用します。

CICS システムは、端末、順次装置、論理装置、および他のシステムと通信できます。TCT は、構成内の各装置を定義します。各 TCT エントリーは、装置のオプション機能および可変機能を CICS に定義し、使用する CICS のオプション機能および可変機能を指定します。

CICS は、通信アクセス方式を使用して端末と通信します。これには、z/OS Communications Server または (順次装置の場合) BSAM を使用することができます。これらのいずれかまたは両方をシステムで使用することができます。

端末は、通信装置として使用することができます。例えば、IBM 3279 カラー・ディスプレイ端末、あるいは IBM 4700 金融機関用通信システムのようなサブシステムなどです。端末は、ローカル (チャネル接続) にすることもリモート (リンク接続) にすることもできます。

順次装置を使用して、CICS 端末をシミュレートすることができます。カード・リーダーやパンチ、ライン・プリンター、直接アクセス・ストレージ・デバイス (ディスク・ドライブ)、磁気テープ・ドライブを順次装置として定義することができます。

論理装置 (LU) は、SNA ネットワークのユーザーがネットワーク機能にアクセスする際に経由するポートです。

システムには、例えば、別の CICS システム、IBM 8815 Scanmaster、IBM ディスプレイライター、あるいは APPC/PC などがあります。CICS システムとの相互通信は、以下のいずれかです。

- LUTYPE 6.1 または LUTYPE 6.2 を使用する、あるいは間接リンクとして中間システムを使用する、異なるプロセッサー間 (システム間通信または **ISC**)。(非 CICS システムとの相互通信でも ISC を使用します。)
- 領域間通信 (IRC) を使用する同じプロセッサー内 (複数領域操作または **MRO**)。(同じプロセッサー内で LUTYPE 6.2 ISC を使用することもできます。)

DFHTCT マクロ・タイプ

コーディングする DFHTCT マクロは、定義する装置および使用するアクセス方式によって変わります。

必ず、以下のいずれかから開始してください。

DFHTCT TYPE=INITIAL,... (451 ページの『制御セクション: DFHTCT TYPE=INITIAL』 を参照してください。)

RDO に適格な定義を CSD ファイルにマイグレーションするために TCT をアセンブルする場合に使用する特殊マクロがあります。

DFHTCT TYPE=GROUP,... (453 ページの『TCT 定義のマイグレーション: DFHTCT TYPE=GROUP』 を参照してください。)

任意の順序で装置を定義することができます。装置ごとに 1 つ以上のマクロが必要で、特定の順序でなければならない場合があります。そのような場合は通知されます。以下の表は、各タイプの装置あるいはシステムに必要なマクロを示しています。

表 *14. DFHTCT* マクロ・タイプ

装置またはシステム	**DFHTCT** マクロ
論理装置コード。	DFHTCT TYPE=LDC,... DFHTCT TYPE=LDCLIST,...
順次装置。	DFHTCT TYPE=SDSCI,... DFHTCT TYPE=LINE,... DFHTCT TYPE=TERMINAL,...
トランザクション・ルーティング用のリモート非 SNA LU。	DFHTCT TYPE=REMOTE,... または: DFHTCT TYPE=REGION,... DFHTCT TYPE=TERMINAL,...

マクロの末尾には、必ず次のようにコーディングします。

```
DFHTCT TYPE=FINAL
END
```

このマクロについては、TYPE=FINAL (テーブルの終わり) で説明しています。

注:

SYSIDNT および **TRMIDNT** オペランド
: CICS は、SYSIDNT および TRMIDNT について大文字と小文字のどちらも受け入れますが、小文字の場合は重複が検査されません。小文字の SYSIDNT あるいは TRMIDNT を含む TCT をアセンブルすると、MNOTE が生成されます。重複検査を行いたい場合は、SYSIDNT および TRMIDNT に大文字のみを使用してください。

TCT のアセンブル
: TCT のアセンブリーおよびリンク・エディットを行うと、2 つの別個のロード・モジュールが作成されます。接尾部に TCT が付いたアセンブリー (ソース名 DFHTCT*xx*) を行うと、単一のテキスト・ファイルが生成されま

す。ただし、これがロード・ライブラリーにリンク・エディットされると、以下の 2 つのメンバーが作成されます。

1. DFHTCT*xx*。これには、RDO に適格ではない定義が制御ブロック形式で含まれます。
2. DFHRDTxx。これには、RDO に適格な (SNA LU およびシステム) 定義が RDO コマンド形式で含まれます。

これは、**RDO** を使用する予定であるかどうかに関係なく発生します。アセンブルした TCT テーブルをロード・ライブラリー間でコピーや移動する場合は、これらの 2 つのテーブルの存在を認識しておく必要があります。

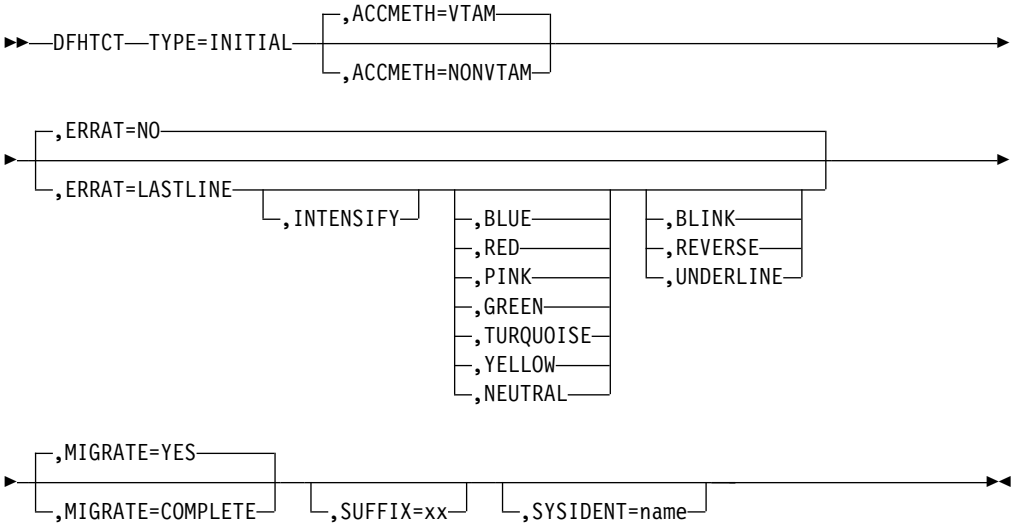
CICS の始動後に TCT を再アセンブルする場合、ウォーム・スタートまたは緊急始動時にすべての変更が取り込まれます。

制御セクション: DFHTCT TYPE=INITIAL

リソースを定義するすべてのマクロの前に、DFHTCT TYPE=INITIAL マクロを 1 つコーディングします。

TYPE=INITIAL マクロには、以下の 2 つの目的があります。

1. TCT をアセンブルするストレージ域を設定するため。
2. TCT 全体、あるいは z/OS Communications Server 以外の個々のエントリー (および任意の z/OS Communications Server-LDC 定義) に適用する情報を指定するため。



注: SYSIDNT は、相互通信専用です。

注: TYPE=INITIAL マクロの一般情報については、TYPE=INITIAL (制御セクション) を参照してください。

ACCMETH=([VTAM,]NONVTAM)

これは、実行中の CICS システムで必要なアクセス方式を指定します。VTAM は、z/OS Communications Server になりました。

VTAM
z/OS Communications Server をアクセス方式として使用している場合、これを指定します。

NONVTAM
z/OS Communications Server 以外の通信アクセス方式を使用している場合は、これを指定します。これらのアクセス方式には、BSAM (順次装置用) が含まれます。

注: デフォルトは、VTAM と NONVTAM の両方が想定されています。

ERRATT={NO|([LASTLINE][, INTENSIFY]
[,{BLUE|RED|PINK|GREEN|TURQUOISE|YELLOW| NEUTRAL}][,{BLINK|REVERSE|
UNDERLINE}])}
すべての 3270 ディスプレイ画面にエラー・メッセージを表示する方法を示します。デフォルトを NO にすることも、LASTLINE、INTENSIFY、いずれかの色、いずれかの強調表示を任意に組み合わせて指定することもできます。INTENSIFY、いずれかの色、あるいはいずれかのハイライトを指定すると、強制的に LASTLINE になります。

特定の装置に対して無効な属性は、すべて無視されます。

NO エラー・メッセージは、現行カーソル位置に表示され、追加属性は表示されません。

LASTLINE
エラー・メッセージは、そのメッセージを画面内に収めることができる、画面の最下部に最も近い行の先頭から表示されます。

注意: 複数のメッセージを次々に受信した場合、メッセージが別のメッセージの上に重なって表示されるので注意してください。先に表示されたメッセージが、オペレーターがそれを読む前に消える可能性があります。

INTENSIFY
エラー・メッセージが強調表示され、画面の最後の行に表示されます。

BLUE|RED|PINK|GREEN|TURQUOISE|YELLOW| NEUTRAL
エラー・メッセージは、指定された色で画面の最後の行に表示されます。

BLINK|REVERSE|UNDERLINE
エラー・メッセージは、強調表示され、画面の最後の行に表示されます。

MIGRATE={YES|COMPLETE}
このオペランドは、オンライン・リソース定義 (RDO) に適格な z/OS Communications Server 装置用の TCT エントリーの作成を制御します。RDO に適格なリソースをマクロ・ソースから CSD ファイルに移動することができる唯一の方法は、下記の YES で記載されているように DFHCSDUP を使用することです。

YES
YES は、RDO に適格なリソースをマイグレーションするために必要なデータを生成することを示します。マクロ・ソースから生成されるレコードは、DFHCSDUP ユーティリティー・プログラムへの入力データとして使用されるように設計されています。RDO に適格な各リソースに対して、MNOTE 注意メッセージが発行されます。

COMPLETE
COMPLETE の使用は、RDO に適格などの装置のマクロ・ソースからも TCT エントリーが生成されないことを意味します。装置ごとに、アセンブリーが MNOTE を生成します。これは、定義をマイグレーションした後も、TCT マクロ・ソース・コードを保持できることを意味します。

RDO に適格ではないリソース用に TCT のアセンブルを続行する場合は、引き続き MIGRATE=COMPLETE を使用してください。

SYSIDNT={CICS|name}
この 1 から 4 文字の名前は、CICS システムが自身を識別するために使用するプライベート名です。DFHTCT TYPE=REGION マクロを使用してリモート端末を定義する場合、このオペランドをコーディングする必要があります。これは、TYPE=REGION マクロに続く各 TYPE=TERMINAL マクロから生成されるのがリモート TCT 端末エントリーであるのかローカル TCT 端末エントリーであるのかを判別するために使用されます。TYPE=REGION マクロの SYSIDNT が TYPE=INITIAL マクロの SYSIDNT と同じである場合、ローカル定義が生成されます。それ以外の場合は、リモート定義が生成されます。

このオペランドに指定した値は、TCT のアセンブリー中にのみ、ローカル領域の名前を指定するために使用されます。実行するためには、SYSIDNT システム初期設定パラメーターを使用してローカル領域の名前も定義する必要があります。

TCT 定義のマイグレーション: DFHTCT TYPE=GROUP

オンライン・リソース定義にマイグレーションする際に TCT 定義を入れるグループを指定するには、このマクロを使用します。

このマクロは、マクロ・ソース内で回数や場所を問わずに使用することができます。このマクロを指定するたびに、次の DFHTCT TYPE=GROUP マクロが指定されるまで後続の定義を入れる CSD ファイル・グループを定義します。

GROUP=name
後続の定義のマイグレーション先となるグループの名前を指定して、これをコーディングします。この名前は、最大 8 文字の英数字で指定することができますが、DFH で開始することはできません。デフォルト名は、TCT*xx* です。ここで、*xx* は、DFHTCT TYPE=INITIAL マクロで SUFFIX として指定された値です。エラーが検出された場合は、既存のグループ名が引き続き使用されます。

指定した名前を持つグループが存在しない場合は、そのグループが作成されます。既に存在する場合は、後続の定義はそのグループに追加されます。

DFHTCT 論理装置コード: z/OS Communications Server 非 3270

特定のタイプの論理装置が、サブシステム内部の複数の資源へアクセスできるようにするために使用されます。例えば、カード・パンチ装置が 3770 論理装置に接続されている場合があります。CICS アプリケーション・プログラムは、BMS を使用

して、3770 経由でカード・パンチ装置にパンチ出力を送信することができます。このタイプの論理装置内の装置への通信を可能にするために CICS によって提供される機能が、論理装置コード (LDC) です。

特定のタイプの論理装置が、サブシステム内部の複数の資源へアクセスできるようにするために使用されます。例えば、カード・パンチ装置が 3770 論理装置に接続されている場合があります。CICS アプリケーション・プログラムは、BMS を使用して、3770 経由でカード・パンチ装置にパンチ出力を送信することができます。このタイプの論理装置内の装置への通信を可能にするために CICS によって提供される機能が、論理装置コード (LDC) です。

これらは z/OS Communications Server 装置ですが、他の z/OS Communications Server 装置とは異なり、マクロ定義を必要とします。

LDCをサポートする論理装置は以下のとおりです。
3601 論理装置
3770 バッチ論理装置
3770 バッチ・データ交換論理装置
3790 バッチ・データ交換論理装置
LUTYPE 4論理装置

CICS アプリケーション・プログラム内、あるいはメッセージ交換のための CMSG トランザクション内でそのような装置を参照するには、LDC 簡略記号を指定します。この簡略記号は、CICS によって数値 LDC 値に変換されます。 CICS は、出力データ・ストリームを論理装置に送信する際に、機能管理ヘッダー (FMH) に LDC 値を組み込みます。論理装置はデータ・ストリームを受信すると、LDC値を使用して、どのコンポーネントが出力を受け取るか、または何らかの標準のアクションを実行するのかを決定します。

参照される各LDC簡略記号を、TCTで定義する必要があります。オプションで、関連LDC値と、BMS機能が使用する特定の装置特性も定義します。このような LDC 情報は、システム **LDC** テーブルまたは拡張ローカル **LDC** リストに含まれます。以下のDFHTCTマクロをコーディングして、システム LDCテーブルまたは拡張ローカルLDCリストを指定します。

- システムLDCテーブルに項目を生成するコードDFHTCT TYPE=LDC マクロ。CICS によって提供される特定のデフォルト LDC エントリーを生成することができます。例えば、次のとおりです。

  ```
  DFHTCT TYPE=LDC,LDC=SYSTEM
  ```

  とすると、システムLDCテーブルで以下の項目が生成されます。

表 *15.* システム *LDC* テーブル・エントリー

LDC 簡略記号	**LDC** 値	装置	ページ・サイズ **(**行、列**)**
DS	1	3604 キーボード・ディスプレイ	6,40
JP	2	3610 文書プリンター	1,80
PB	3	通帳・帳票プリンター	1,40
LP	4	3618 管理ライン・プリンター	50,80
MS	5	3604 磁気ストライプ・エンコーダー	1,40

表 *15.* システム *LDC* テーブル・エントリー *(*続き*)*

LDC 簡略記号	**LDC** 値	装置	ページ・サイズ **(**行、列**)**
CO	0	コンソール・メディアまたはデフォルト印刷データ・セット・グループ	
R1	32	カード入力メディア	1,80
H1	32	カード出力メディア	1,80
P1	48	印刷メディアまたは印刷データ・セット・グループ	50,80
W1	128	ワード・プロセッサー・メディア 1	50,80
W2	144	ワード・プロセッサー・メディア 2	50,80
W3	160	ワード・プロセッサー・メディア 3	50,80
W4	192	ワード・プロセッサー・メディア 4	50,80

特にシステムLDCテーブルにLDC項目を追加するために、LDCを 定義することもできます。例えば、次のとおりです。

```
DFHTCT TYPE=LDC,
       LDC=XX,
       DVC=BLUPRT,
       PGESIZE=(12,80),
       PGESTAT=PAGE
DFHTCT TYPE=LDC,
       LDC=YY,
       DVC=BLUPCH,
       PGESIZE=(1,80),
       PGESTAT=AUTOPAGE
```

と定義すると、システムLDCテーブルに以下の項目が追加されます。

表 *16. LDC* によって定義されるシステム *LDC* テーブル・エントリー

LDC 簡略記号	**LDC** 値	装置	ページ・サイズ **(**行、列**)**	**PGESTAT**
XX	48	バッチ LU プリンター	12,80	PAGE
YY	32	バッチ LU カード出力	1,80	AUTOPAGE

- システムLDCテーブルの代わりに、以下の一連のDFHTCT TYPE=LDC マクロをコーディングして、拡張ローカルLDCリストを作成することが できます。デフォルト項目も生成できます。例えば、次のとおりです。

```
LDC1  DFHTCT TYPE=LDC,LOCAL=INITIAL
*  次の行はデフォルトのCO, R1, H1, P1というLDCを生成します。
   DFHTCT TYPE=LDC,LDC=BCHLU
   DFHTCT TYPE=LDC,DVC=BLUPRT,LDC=AA,
          PGESIZE=(6,30)
   DFHTCT TYPE=LDC,DVC=BLUPCH,LDC=BB,
          PGESIZE=(1,80)
   DFHTCT TYPE=LDC,DVC=BLUCON,LDC=CC,
          PGESIZE=(1,132)
   DFHTCT TYPE=LDC,LOCAL=FINAL
```

と定義すると、以下の項目を含むLDC1という名前の拡張ローカルLDC リストが生成されます。

表 *17.* 拡張ローカル *LDC* リスト・エントリー

LDC 簡略記号	**LDC** 値	装置	ページ・サイズ **(**行、列**)**
CO	0	コンソール・メディアまたはデフォルト・プリンター・データ・セット・グループ	
R1	32	カード入力メディア	1,80
H1	32	カード出力メディア	1,80
P1	48	印刷メディアまたは印刷データ・セット・グループ	50,80
AA	48	BLUPRT バッチ LU プリンター	6,30
BB	32	BLUPCH バッチ LU カード出力	1,80
CC	0	BLUCON バッチ LU コンソール・プリンター	1,132

TCT内で論理装置を定義している場合、以下の2つの方法のいずれかでLDCを指定できます。

1. LDC 簡略記号 (およびオプションでその LDC 値) のローカル・リストを定義するには、DFHTCT TYPE=LDCLIST マクロをコーディングします。例えば、以下のとおりです。

   ```
   LDC2 DFHTCT TYPE=LDCLIST,
               LDC=(DS,JP,PB=5,LP,MS)
   ```

   論理装置を定義するDFHTCT TYPE=TERMINALマクロでは、 LDCオペランドに、DFHTCT TYPE=LDCLISTマクロで 定義されたローカル・リストの名前を指定 します。以下に例を挙げます。

   ```
        DFHTCT TYPE=TERMINAL,
               TRMTYPE=3600,
               LDC=LDC2, ...
   ```

   この定義は、DS, JP, PB, LP,およびMSというLDCを、 定義中の3601論理装置に関連付けています。LDC 値は、ローカル・リストで指定しても、システム LDC テーブルから取得してもかまいません。BMSが これらのLDC簡略記号を使用する場合、そのページ・サイズと ページ状況は、システムLDCテーブルからも使用可能で なければなりません。

   注**:** DFHTCT TYPE=LDCLISTマクロによって定義されたローカル・リストは、 いくつかの3601, LUTYPE 4、およびバッチ論理装置によって共用 されることがあります。

2. 論理装置を定義するDFHTCT TYPE=TERMINALマクロでは、 LDCオペランドに拡張ローカルLDCリストの名前を指定 します。以下に例を挙げます。

   ```
   LDC1 DFHTCT TYPE=LDC,LOCAL=INITIAL
        DFHTCT TYPE=LDC,LDC=BCHLU
        DFHTCT TYPE=LDC,DVC=BLUPRT,LDC=AA,
               PGESIZE=(6,30)
        DFHTCT TYPE=LDC,DVC=BLUPCH,LDC=BB,
               PGESIZE=(1,80)
        DFHTCT TYPE=LDC,DVC=BLUCON,LDC=CC,
               PGESIZE=(1,132)
        DFHTCT TYPE=LDC,LOCAL=FINAL
        DFHTCT TYPE=TERMINAL,TRMTYPE=BCHLU,
               LDC=LDC1, ...
   ```

この定義では、CO, R1, H1, P1, AA, BB,およびCCという LDCを、定義中のバッチ論理装置と関連付けています。 そのLDC値およびBMS機能の装置特性が、拡張ローカルLDC リスト内に記述されています。このリストの名前はLDC1 です。

CICS は、論理装置の特定の LDC 簡略記号を使用して、出力あるいはメッセージ交換操作を要求する場合、DFHTCT TYPE=TERMINAL マクロの LDC オペランドで指定されたリスト (形式は問いません) から簡略記号を解決しようと試みます。LDCが、ローカル・リストまたは拡張ローカル・リスト内で見つからない場合、指定されたLDCは、その端末項目では無効です。この場合、FMH の論理装置コード部分に X'00' が挿入され、宛先名前は挿入されません。

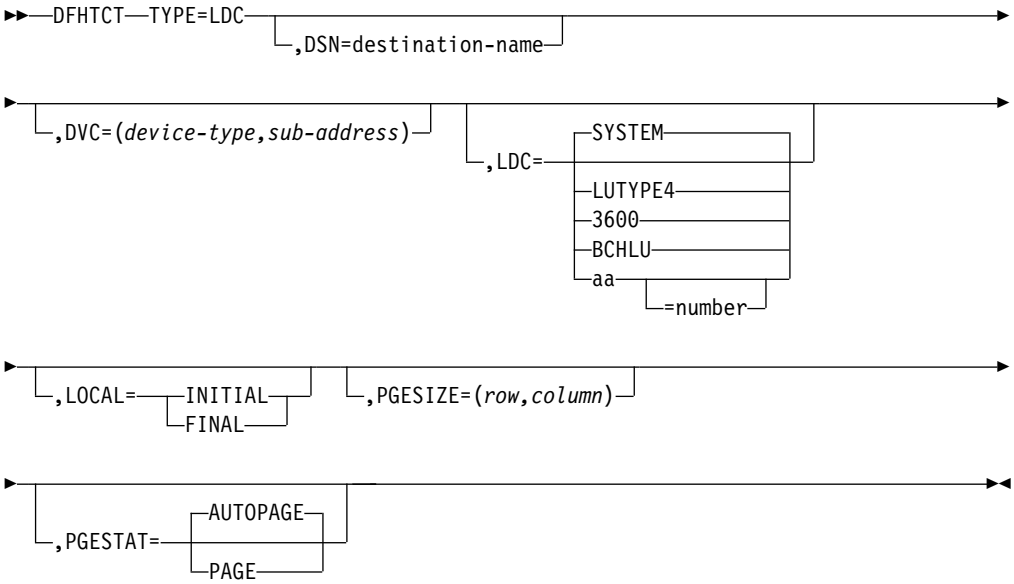
LDC用にBMS機能が要求され、LDC簡略記号が正常に解決されると、装置特性（例えば装置名および宛先名）がBMS機能用にアクセスされます。LDCが拡張ローカルLDCリスト内にある場合は、これらの特性は検出された拡張ローカル・リスト項目内に存在します。それ以外の場合、システムLDCテーブルで LDCおよび関連装置特性を検索します。

論理装置コード — DFHTCT TYPE=LDC マクロ

DFHTCT TYPE=LDC マクロは、3600、LUTYPE4、3770 バッチ論理装置、および 3770/3790 バッチ・データ交換論理装置でのみ使用できます。

端末で使用する LDC 構成のセットアップは、ユーザーが行う必要があります。

このマクロの拡張は、TCT 定義のどこで指定されているかに関係なく同じです。



name

拡張ローカル LDC リストの名前を指定して、これをコーディングします。これは、DFHTCT TYPE=TERMINAL マクロの LDC オペランドで指定されたものと同じである必要があり、LOCAL=INITIAL がコーディングされている場合にのみ必要です。

TYPE=LDC
システム LDC テーブルまたは拡張ローカル LDC リストに LDC を定義する場合に、これをコーディングします。

DSN=destination-name
バッチ・データ交換論理装置の宛先選択で BMS が使用する名前を指定して、これをコーディングします。宛先選択について詳しくは、関連 CICS サブシステムのガイドを参照してください。

DVC=(device-type,sub-address)
BMS 要求に使用する LDC に関連付けられた装置タイプを指定して、これをコーディングします。このオペランドは、LDC=*aa*[=*nnn*] オペランドとの組み合わせでのみコーディングすることができます。

device-type
以下のようにコーディングすることができます。

表 *18. DVC=device-type entries*

デバイス・タイプ	説明
3604	キーボード・ディスプレイ
3610	カット・フォーム文書プリンターまたはジャーナル・プリンター (3612 の文書/ジャーナル・プリンターを含む)
3612	3612 の通帳部分
3618	現在選択されているキャリッジ
3618P	1 次キャリッジ
3618S	2 次キャリッジ
3618B	両方のキャリッジ
BLUCON	バッチ論理装置コンソール・プリンター
BLUPRT	バッチ論理装置のプリンター・コンポーネント
BLURDR	バッチ論理装置のカード入力コンポーネント
BLUPCH	バッチ論理装置のカード出力コンポーネント
WPMED1	ワード・プロセッサー・メディア 1
WPMED2	ワード・プロセッサー・メディア 2
WPMED3	ワード・プロセッサー・メディア 3
WPMED4	ワード・プロセッサー・メディア 4

装置タイプ BLUPRT、BLURDR、BLUPCH、および BLUCON は、バッチ、バッチ・データ交換、または LUTYPE4 論理装置に接続された装置です。

WPMED1、2、3、および 4 オプションは、LUTYPE4 論理装置にのみ適用されます。これらのオプションが適用されるコンポーネントは、特定タイプ 4 の論理装置実装によって定義されます。

sub-address
メディアのサブアドレスを指定して、これをコーディングします。範囲は 0 から 15 で、デフォルトは 0 です。値 15 は、任意のサブアドレスを示します。サブアドレスにより、1 つの論理装置に接続された 2 つの印刷コンポーネントなど、同じ装置タイプを持つ 2 つの装置を区別します (例えば、(BLUPRT,0) と (BLUPRT,1))。

LDC={SYSTEM∨LUTYPE4∨3600∨BCHLU∨(aa[=nnn])}
定義する LDC 簡略記号および数値を指定して、これをコーディングします。LDC=*aa*[=*nnn*] オプションは、DVC、PGESIZE、および PGESTAT の各オペランドとの組み合わせでのみ使用できます。

SYSTEM

3600 論理装置、バッチ論理装置、および LUTYPE4 論理装置について、以下のシステム・デフォルト LDC が設定されます。

表 *19.* システム・デフォルト *LDC*

LDC 簡略記号	**LDC** 値	装置	ページ・サイズ **(**行、列**)**
DS	1	3604 キーボード・ディスプレイ	6,40
JP	2	3610 文書プリンター	1,80
PB	3	通帳・帳票プリンター	1,40
LP	4	3618 管理ライン・プリンター	50,80
MS	5	3604 磁気ストライプ・エンコーダー	1,40
CO	0	コンソール・メディアまたはデフォルト印刷データ・セット・グループ	
R1	32	カード入力メディア	1,80
H1	32	カード出力メディア	1,80
P1	48	印刷メディアまたは印刷データ・セット・グループ	50,80
W1	128	ワード・プロセッサー・メディア 1	50,80
W2	144	ワード・プロセッサー・メディア 2	50,80
W3	160	ワード・プロセッサー・メディア 3	50,80
W4	192	ワード・プロセッサー・メディア 4	50,80

LUTYPE4

LUTYPE4 (ワード・プロセッサー) 論理装置のシステム・デフォルト LDC 簡略記号が設定されます。これらは、CO、R1、P1、H1、W1、W2、W3、および W4 簡略記号、対応する LDC 値、適切なページ・サイズから構成されます。

3600

3600 のシステム・デフォルト LDC 簡略記号が設定されます。これらは、DS、JP、PB、LP、および MS 簡略記号、対応する LDC 値、適切なページ・サイズおよびページ状況から構成されます。

BCHLU

バッチ論理装置のシステム・デフォルト LDC 簡略記号が設定されます。これらは、CO、R1、P1、および H1 簡略記号、対応する LDC 値、適切なページ・サイズおよびページ状況から構成されます。

aa この LDC に使用される 2 文字の簡略記号。

nnn

システム内または拡張ローカル LDC リスト内の LDC に関連付ける数値。 TCTTE に関連付けられたローカル LDC リスト内で値が見つからない (つまり、拡張リスト内にない) 場合は、システム・リスト内の値が、この LDC のデフォルト値として使用されます。 3600 装置の場合は、値を指定する必要があります。バッチ、バッチ・データ交換、および LUTYPE4 論理装置の場合は、値を指定する必要はありません。ただし、指定する場合は、装置タイプの LDC 値に対応している必要があります。

LOCAL={INITIAL∨FINAL}
拡張ローカル LDC リストが生成されます。

INITIAL
これは、拡張 LDC リストの開始です。

FINAL
これは、拡張ローカル LDC リストの終了です。

注: LOCAL=INITIAL あるいは FINAL は、他のオペランドと同じ DFHTCT TYPE=LDC マクロ内にコーディングすることができません。
LOCAL=INITIAL の後および LOCAL=FINAL の前にコーディングされたすべての DFHTCT TYPE=LDC エントリーによって、1 つの拡張ローカル LDC リストのパートが形成されます。このグループ構造の外にコーディングされたエントリーは、システム LDC テーブルに追加されます。

以下は、拡張ローカル LDC リストの例を示しています。

```
        DFHTCT TYPE=TERMINAL,TRMIDNT=BTCH,      *
        TRMTYPE=BCHLU,ACCMETH=VTAM,LDC=LDCA     *
  LDCA  DFHTCT TYPE=LDC,LOCAL=INITIAL
        DFHTCT TYPE=LDC,DVC=BLUPRT,LDC=AA,      *
        PGESIZE=(6,30)
        DFHTCT TYPE=LDC,DVC=BLUPCH,LDC=BB,      *
        PGESIZE=(1,80)
        DFHTCT TYPE=LDC,DVC=BLUCON,LDC=CC,      *
        PGESIZE=(1,132),PGESTAT=AUTOPAGE
        DFHTCT TYPE=LDC,LOCAL=FINAL
```

注: VTAM は現在 z/OS Communications Server になっています。

PGESIZE=(row,column)
BMS 要求の処理時にこの LDC で使用する論理ページ・サイズを指定して、これをコーディングします。

row と *column* の積が 32767 を超えてはなりません。

PGESTAT={AUTOPAGE∨PAGE}
装置が自動ページングを使用するかどうかを示します。自動ページングは、オペレーターの介入なしに、 BMSの複数のページのメッセージが連続して印刷されることを 意味します。これは、プリンターで通常必要とされる機能 です。(複数のページ・メッセージに対する要件とは異なり、オペレーターがページの読み取りを終了したい場合は、次のページの要求が送信される前に 3270 タイプのディスプレイに表示されます。)

PAGINGオプションを指定したBMS SENDコマンドのみが自動ページング を使用します。 TERMINAL または SET を指定した BMS SEND では、自動ページングは使用されません。

AUTOPAGE
プリンターの場合は、これを指定します。

PAGE
ディスプレイの場合は、これを指定します。

LDC オペランドによって指定されたデフォルトの PGESIZE 値または PGESTAT 値をオーバーライドする場合は、オーバーライドされる簡略記号を

指定して特定の LDC をコーディングします。 LDC テーブル内で、LDC オペランドをコーディングする前に、このオーバーライド LDC をコーディングします。

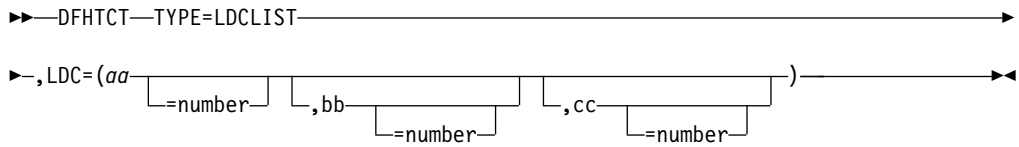
PGESTAT=AUTOPAGE を使用して、DFHTCT TYPE=TERMINAL 内の PGESTAT 指定をオーバーライドすることができます。

論理装置コード — DFHTCT TYPE=LDCLIST

3600、LUTYPE4、およびバッチ論理装置で使用できる DFHTCT TYPE=LDCLIST マクロにより、複数の TCTTE で共用する論理装置コード (LDC) の共通リストを作成することができます。

端末で使用する LDC 構成のセットアップは、ユーザーが行う必要があります。

このマクロの拡張は、TCT 定義のどこでコーディングされているかに関係なく同じです。



listname
LDC リストの必須の名前。この名前は、TCTTE が DFHTCT TYPE=TERMINAL の LDC オペランドで参照します。

TYPE=LDCLIST
定義する LDC リスト。

LDC=(aa[=nnn][,bb[=nnn]][,cc[=nnn]][,...])
このリスト内の LDC (簡略記号、およびオプションで LDC 数値) を指定して、これをコーディングします。

(aa[=nnn][,bb[=nnn]] [,cc[=nnn]][,...])
リスト内の LDC を生成します。

aa,bb,cc...
このリスト内の LDC の 2 文字の簡略記号。

nnn
LDC に割り当てられる 1 から 255 の範囲の 10 進値。この値を指定しない場合、DFHTCT TYPE=LDC マクロによって定義されたテーブルのシステム・デフォルト値がこの LDC に使用されます。バッチまたは LUTYPE4 論理装置の場合は、この値をコーディングする必要はありません。ただし、コーディングする場合は、装置の LDC 値に対応している必要があります。バッチまたは LUTYPE4 論理装置に接続された装置の LDC は、DFHTCT TYPE=LDC マクロの LDC パラメーターの下にリストされます。

DFHTCT の例: LDC

3770 バッチ論理装置用に DFHTCT TYPE=LDC マクロをコーディングする方法を示す例。

```
DFHTCT TYPE=LDC,                        *
       LDC=XX,                          *
       DVC=BLUPRT,                      *
       PGESIZE=(12,80),                 *
       PGESTAT=PAGE
DFHTCT TYPE=LDC,                        *
       LDC=YY,                          *
       DVC=BLUPCH,                      *
       PGESIZE=(1,80),                  *
       PGESTAT=AUTOPAGE
DFHTCT TYPE=LDC,                        *
       LDC=SYSTEM
```

図 *15. 3770* バッチ論理装置用の *LDC* の *TCT* の例

順次装置

CICS は、BSAM を使用して、カード・リーダー、ライン・プリンター、磁気テープ装置、DASD などの順次端末を制御します。順次端末は、実際の端末のシミュレーションを提供し、実際の端末が使用可能になる前に使用することができます。順次端末は、新規アプリケーションのテスト中に使用することもできます。

順次装置を定義するには、以下のマクロ命令をコーディングします。

```
      DFHTCT TYPE=INITIAL,
             ACCMETH=(NONVTAM)    defining the access
                                  メソッド
```

(以下のマクロ命令を続けて定義します。)

```
  DFHTCT TYPE=SDSCI,
         DSCNAME=isadscn,     defining the input
         DDNAME=indd, ...      data set
  DFHTCT TYPE=SDSCI,
         DSCNAME=osadscn,     defining the output
         DDNAME=outdd, ...     data set
  DFHTCT TYPE=LINE,
         ISADSCN=isadscn,
         OSADSCN=osadscn, ...
  DFHTCT TYPE=TERMINAL,
         TRMIDNT=name, ...
```

DFHTCT TYPE=SDSCI マクロによって定義された 2 つのデータ・セットは、DFHTCT TYPE=TERMINAL マクロの TRMIDNT オペランドで指定された名前によって認識される CICS 端末をシミュレートします。入力データ・セットおよび出力データ・セットの DSCNAME は、それぞれ DFHTCT TYPE=LINE マクロの ISADSCN オペランドおよび OSADSCN オペランドで指定されている必要があります。

順次装置のデータ終了標識 (EODI) は、EODI システム初期設定パラメーターを使用して変更することができます。

順次装置の JCL

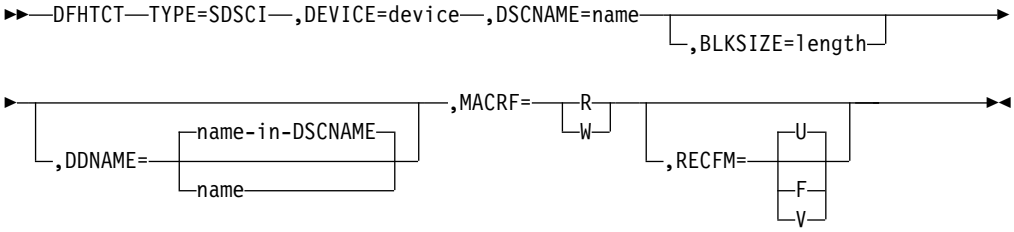
DFHTCT TYPE=SDSCI マクロの DDNAME オペランドは、DD ステートメントの DD 名を指定します。これは、CICS 始動ジョブ・ストリームで指定する必要があります。

```
//indd   DD  ...                input data set
//outdd  DD  ...                output data set
```

ここで、*indd* は、シミュレート端末からの入力が含まれるデータ・セットで、*outdd* は、シミュレート端末への出力が送信されるデータ・セットです。

順次装置: DFHTCT TYPE=SDSCI

DFHTCT TYPE=SDSCI マクロは、CICS 端末をシミュレートする入出力データ・セットの特性を指定します。



BLKSIZE=length
ブロックの最大長 (バイト数) を指定して、これをコーディングします。

デフォルトは BLKSIZE=0 です。このオペランドを省略した場合、データ・セットに関連付けられたデータ定義 (DD) ステートメントでブロック・サイズを指定することができます。このオペランドの詳細な説明については、z/OS DFSMS Macro Instructions for Data Setsを参照してください。

DDNAME={name-in-DSCNAME|name}
特定のデータ・セット (行グループ) に関連付けられたデータ定義 (DD) ステートメントの名前を指定します。このオペランドを省略すると、DSCNAME が DDNAME になります。

DEVICE=device
以下のいずれかの値をコーディングすることができます。

- カード・リーダーの場合: **{1442|2501|2520|2540|2560|2596|3505|3525|5425}**
- ライン・プリンターの場合: **{1403|1404|1443|1445|3203|3211|5203}**
- ディスク (DASD) の場合: **{2314|3330|3340|3350|DASD|DISK}**
- 磁気テープの場合: **TAPE**

  TAPE を指定すると、入力データ・セット用および出力データ・セット用の両方のテープ作業ファイルが生成されます。ラベルの有効期限が切れた入力テープを使用すると、ヘッダーが再書き込みされ、最初のデータ・レコードが破壊される原因となる可能性があるので注意してください。

DSCNAME=name
入力データ・セットまたは出力データ・セットの名前。入力データ・セットを定義する場合は、DFHTCT TYPE=LINE マクロの ISADSCN が、指定した名前

と一致する必要があります。出力データ・セットを定義する場合は、DFHTCT TYPE=LINE マクロの OSADSCN がその名前と一致する必要があります。

MACRF=([R][,W])
順次装置にアクセスする方法を指定して、これをコーディングします。

R READ マクロを示します。

W WRITE マクロを示します。

デフォルトは、カード・リーダーの場合は MACRF=R、ライン・プリンターの場合は MACRF=W です。その他の順次装置の場合は、MACRF=R または MACRF=W をコーディングする必要があります。

RECFM={U|F|V}
DCB のレコード形式を指定して、これをコーディングします。

U 未定義レコードを示します。DEVICE=1403 または 3211 の場合、あるいは順次端末出力に DASD を使用している (つまり、DEVICE=DASD かつ MACRF=W) 場合は、これをコーディングします。

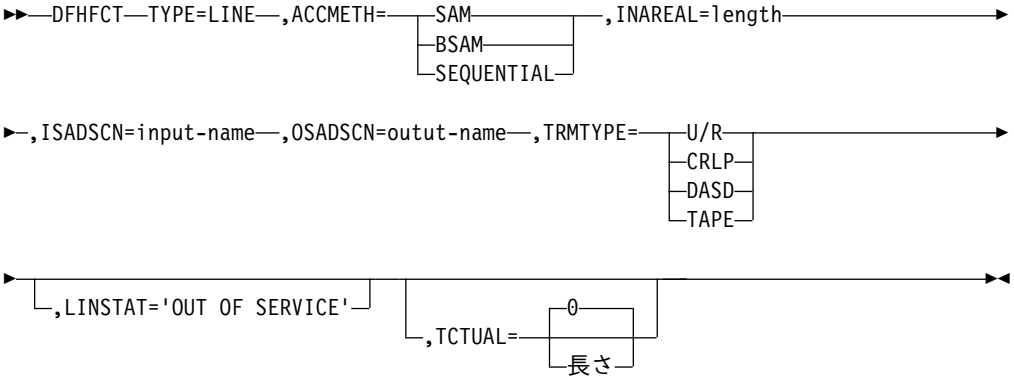
F 固定長レコードを示します。

V 可変長レコードを示します。

このオペランドを省略した場合、順次データ・セットに関連付けられたデータ定義 (DD) ステートメントでレコード形式を指定することができます。

順次装置 — DFHTCT TYPE=LINE

DFHTCT TYPE=LINE マクロは、CICS 端末をシミュレートする順次データ・セットの詳細な特性を指定します。



ACCMETH={SAM∨BSAM∨SEQUENTIAL}
SAM、BSAM、または SEQUENTIAL を指定します。これらは、CICS では同等です。

INAREAL=length
メッセージ入力域の長さを指定して、これをコーディングします。この値は、複数の物理レコードが含まれる可能性のあるトランザクションの、最も長い初期論理レコードの長さと等しくなければなりません。

ISADSCN=name
入力データ・セットの名前。これは、入力データ・セットの TYPE=SDSCI DSCNAME オペランドと一致している必要があります。

LINSTAT='OUT OF SERVICE'
行が「サービス休止」状況で開始されます。

デフォルトは「サービス中」です。

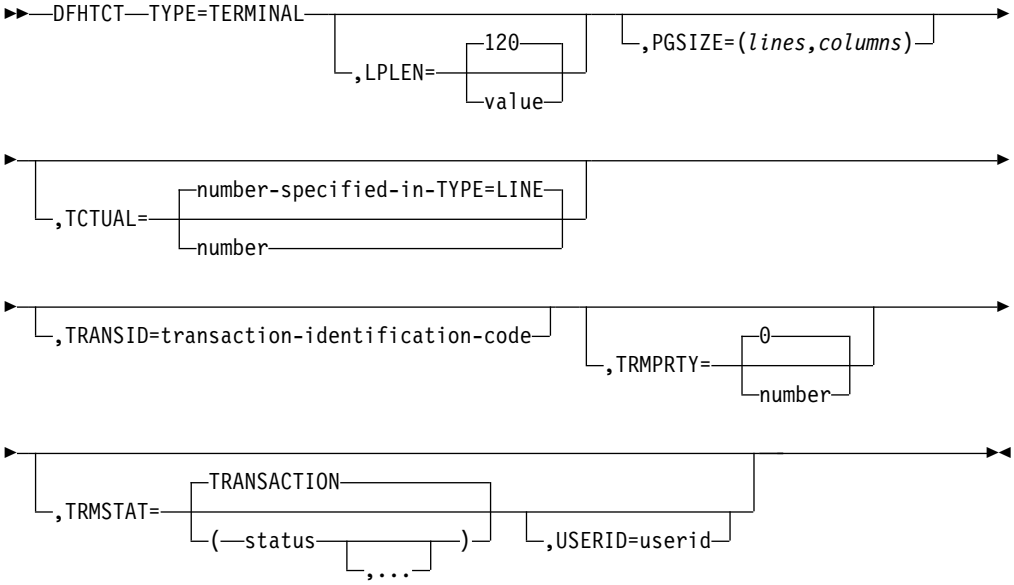
OSADSCN=name
出力データ・セットの名前。これは、出力データ・セットの TYPE=SDSCI DSCNAME オペランドと一致している必要があります。

TCTUAL={0∨length}
この行に関連付けられたすべての端末入力 (TCTTE) のユーザー領域 (プロセス制御情報フィールドあるいは PCI) の長さをバイト数 (0 から 255) で示します。これは、できるだけ小さい値にしてください。 TCTユーザー域はシステム初期化時にゼロに初期化されます。 異なる長さ (可変長) のフィールドが必要な場合は、この行に関連付けられた端末に対する 1 つ以上の TYPE=TERMINAL マクロ命令で TCTUAL 値を指定することができます。

TRMTYPE=(U/R∨CRLP∨DASD∨TAPE)
以下の順次装置タイプを示します。
U/R 任意の読取装置またはプリンター
CRLP カード・リーダーおよびライン・プリンター
DASD
直接アクセス・ストレージ・デバイス
TAPE 磁気テープ装置

順次装置: DFHTCT TYPE=TERMINAL

DFHTCT TYPE=TERMINAL マクロは、順次データ・セットのペアによってシミュレートされる CICS 端末の端末名およびその他の特性を指定します。



LPLEN={120∨value}
SAM 出力ライン・プリンターの印刷行の長さを制御します。セグメント化された書き込みに NL 記号が検出されない場合は、印刷行の長さは LPLEN 値になります。デフォルトは LPLEN=120 です。

PGESIZE=(lines,columns)

1403 または CRLP 端末のデフォルト・ページ・サイズは (12,80) です。TRMTYPE=DASD が指定された装置に BMS が必要な場合は、PGESIZE をコーディングし、使用する行と列の数を指定します。これら 2 つの値を乗算した値が、INAREAL に指定された値と等しくなければなりません。*lines* と *columns* の積が 32767 を超えてはなりません。

TCTUAL={number-specified-in-TYPE=LINE ∨number}

この端末に関連付けられた端末入力 (TCTTE) のユーザー領域 (プロセス制御情報フィールドあるいは PCI) の長さをバイト数 (0 から 255) で示します。これは、できるだけ小さい値にしてください。 TCTユーザー域はシステム初期化時にゼロに初期化されます。

この行に関連付けられた端末用に異なる長さ (可変長) のフィールドが必要な場合は、DFHTCT TYPE=TERMINAL マクロの TCTUAL オペランドを使用します。いずれの場合も、PCI フィールドは、最後の行の最後の端末入力の後で各端末に対して生成されます。PCI フィールドのアドレスは TCTTECIA に、長さは TCTTECIL にあります。

TRANSID=transaction-identification-code

1 から 4 文字のトランザクション・コードを指定して、これをコーディングします。このコードは、アクティブ・タスクがない場合に、端末から入力を受信するたびに開始されるトランザクションを指定します。

TCTTE で TRANSID が指定されていない場合、前回のトランザクションからの RETURN コマンドでの TRANSID が使用されます。それ以外の場合は、TIOA で渡された最初の 1 から 4 文字のデータがトランザクション・コードとして使用されます。トランザクション ID が 4 文字未満の場合は、区切り文字が必要です。

TRMIDNT=name

端末ごとに固有の 4 文字のシンボル ID を指定して、これをコーディングします。指定された ID は左寄せされ、4 文字未満の ID が指定された場合は 4 文字になるまでブランクが埋め込まれます。

値 CERR は、エラー・コンソール用に生成される ID のため予約済みです。

TRMPRTY={0∨number}

端末の優先度を指定します。この 10 進値 (0 から 255) は、全体的なトランザクション処理の優先順位の設定に使用されます。(トランザクション処理の優先度は、端末優先度、トランザクション優先度、およびオペレーター優先度の合計と等しく、255 を超えてはなりません。)

TRMSTAT={TRANSACTION∨(status,...)}

特定の端末で発生する可能性のあるアクティビティーのタイプを指定して、これをコーディングします。この端末状況は、最初は TCTTE で設定されており、処理状況とサービス状況の組み合わせです。

TRANSACTION

TRANSACTION 状況の端末がトランザクション (照会や命令入力) の処理に使用されます。端末要求なしでメッセージが送信されることがなく、トランザクションの入力に使用されるディスプレイまたはハードコピー端末が、TRANSACTION 端末です。

INPUT
CICS にメッセージを送信できるが、CICS からメッセージを受信できない端末を示します。

注: システム・メッセージは、無効なトランザクション ID や ATP バッチ・カウントなどの条件下で、入力端末に送られる場合があります。これにより、DFHTACP がスケジュールされます。この状態を処理するには、ユーザーが必要とするアクションを実行するように DFHTEP をコーディングします。

'OUT OF SERVICE'
メッセージの受信も入力の送信もできない端末を示します。このような端末は、CICS によってポーリングされません。'OUT OF SERVICE' パラメーターは、任意の状況設定と組み合わせて使用することができます。

マスター端末を除くすべての端末を 'OUT OF SERVICE' として指定することができます。必要なときに、マスター端末によって端末をサービス状態にし、ポーリングを再開することができます。

RECEIVE
その端末にメッセージを送信することはできるが、そこから入力することはできない端末を示します。このタイプの端末の例としては、ウェアハウスなどのリモート・ロケーションにあり、無人であるがメッセージを受信できる端末があります。TRANSCEIVE 用に自動トランザクション開始が実装されています。

TRANSCEIVE
TRANSCEIVE 状況の端末は、メッセージが自動的に送信される先の TRANSACTION 端末です。一時データ管理またはインターバル制御機能による自動トランザクション開始により、適切な端末管理テーブルの端末入力における条件が設定されます。端末状況が TRANSCEIVE であり、端末にトランザクションがない場合、端末管理によってユーザー定義タスクが開始されます。このタスクが端末に メッセージを送信します。

USERID=userid
CESN を使用してサインオンできないプリンターなどの装置用にユーザー ID を指定するには、これをコーディングします。(ディスプレイ装置用に USERID を指定することもできます。この場合、ディスプレイは永続的にサインオンされた状態になります。オペレーターがサインオンすることはできません。) この装置で事前設定セキュリティーを使用したい場合は、このオペランドをコーディングする必要があります。保護リソースへのすべてのアクセスは、USERID に依存します。

ユーザー ID は、セキュリティー・エラー・メッセージ、セキュリティー違反メッセージ、および監査証跡で参照されます。ユーザー ID をセキュリティー・マネージャーに対して定義する必要があります。

ユーザー ID は、1 から 8 文字の固有のユーザー ID でなければなりません。(使用できる文字は、A-Z、0-9、#、$、および @ です。)

トランザクション・ルーティング用のリモート端末

CICS 相互通信でのリソース定義について考える場合、TCT がインストールされているシステムはローカル・システムです。TCT で定義されるシステムは、リモート・システムです。トランザクション・ルーティングにより、ある CICS システム内の端末が、別の CICS システム内のトランザクションを呼び出すことが可能になります。

MRO または LUTYPE 6.2 リンクによって接続されたシステム間で、トランザクション・ルーティングを使用することができます。

トランザクション・ルーティング用の端末のリモート定義

マクロを使用して端末を定義する場合、使用可能な方法が 2 つあります。

2 つの方法は、以下のとおりです。

- 方法 **1:**

```
DFHTCT TYPE=REGION, ...   (one for each region)

DFHTCT TYPE=SDSCI, ...    (for non-SNA LU only;
                           ignored for remote definitions)

DFHTCT TYPE=LINE, ...     (for non-SNA LU only)

DFHTCT TYPE=TERMINAL, ... (for non-SNA: one for each LU)
```

- 方法 **2:**

```
DFHTCT TYPE=REMOTE, ...   (one for each terminal)
```

ヒント: その他に、RDO を使用して、「シップ端末定義」と呼ばれる方法が使用可能です。 215 ページの『トランザクション・ルーティング用の端末』を参照してください。)

いずれの方式でも、同じ端末定義を使用して、ローカル・システムとリモート・システムの両方に必要な項目を生成することができます。

方法 1:

コピーブックを使用して、ローカル・システムとリモート・システムの TCT に同じソース・コードを組み込むことができます。不要な情報 (つまり、TYPE=SDSCI マクロ全体や TYPE=LINE マクロおよび TYPE=TERMINAL マクロの一部) は、リモート・エントリーでは破棄されます。

CICS は、TYPE=REGION マクロの SYSIDNT オペランドに基づいて、リモート定義を作成するかローカル定義を作成するかを決定します。これは、DFHTCT TYPE=INITIAL の SYSIDNT オペランドと比較されます。それらが同じである場合、定義はローカルです。それらが異なっている場合、定義はリモートです。

方法 2:

単一の DFHTCT TYPE=REMOTE マクロを採用します。

CICS は、TYPE=REMOTE マクロの SYSIDNT オペランドに基づいて、リモート定義を作成するかローカル定義を作成するかを決定します。これは、

DFHTCT TYPE=INITIAL の SYSIDNT オペランドと比較されます。それらが同じである場合、定義はローカルです。それらが異なっている場合、定義はリモートです。

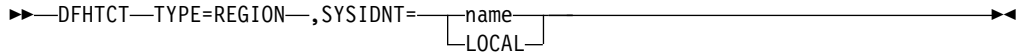
以下の端末は、トランザクション・ルーティングを使用することができないため、リモートとして定義することができません。

- IBM 7770 または 2260 端末
- MVS システム・コンソール
- プールされた 3600 または 3650 パイプライン論理装置

リモート端末、方法 1 — DFHTCT TYPE=REGION

DFHTCT TYPE=REGION マクロは、指定された領域の情報を導入します。この情報は、DFHTCT TYPE=LINE マクロおよび TYPE=TERMINAL マクロから構成されます。

これらのマクロは、DFHTCT TYPE=REGION マクロの後になければなりません。リモート領域の場合、DFHTCT TYPE=LINE マクロは、TCT ライン・エントリー (TCTLE) を生成しません。トランザクション・ルーティングに参加するすべての端末が定義されている必要があります。リモート領域定義では、特定の DFHTCT マクロ・タイプおよびオペランドのみが使用できます。その他はすべて無視されます。使用できるオペランドは、 470 ページの『リモート端末、方法 2: DFHTCT TYPE=REMOTE』 にリストされているものです。



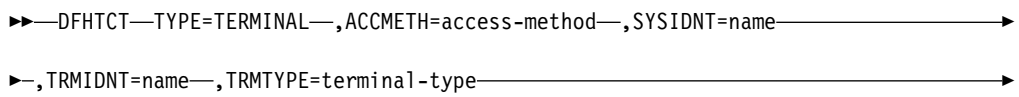
SYSIDNT={name∨LOCAL}

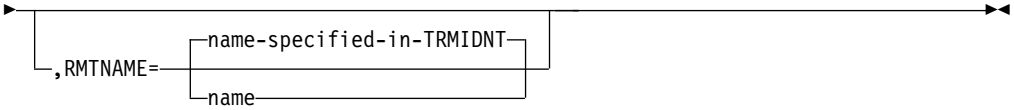
ここで情報を開始または再開する対象のシステムあるいは領域の 4 文字の名前を示します。 SYSIDNT=LOCAL を指定して、その後に続く TYPE=TERMINAL 定義が、最初の DFHTCT TYPE=REGION マクロの前にあるすべての定義と同様に、ホーム領域を参照するように指定することができます。ホーム領域 (つまり、この端末管理テーブルが使用されている領域) の名前は、DFHTCT TYPE=INITIAL マクロの SYSIDNT オペランドの値です。この名前は、代わりに、以前に定義された MRO リンクまたは ISC リンクの名前であっても構いません。

リモート端末、方法 1 — DFHTCT TYPE=TERMINAL

方法 1 を使用してリモート端末を定義するための TCT マクロ定義。

注: DFHTCT TYPE=LINE マクロ、および DFHTCT TYPE=TERMINAL マクロの追加オペランドは有効ですが、先行 DFHTCT TYPE=REGION マクロの SYSIDNT オペランドがリモート領域を示している場合は無視されます。 (DFHTCT TYPE=LINE マクロおよび DFHTCT TYPE=TERMINAL マクロについて詳しくは、 462 ページの『順次装置』 を参照してください)





ACCMETH=access-method
リモート端末のアクセス方式を指定して、これをコーディングします。

RMTNAME={name-specified-in-TRMIDNT|name}
端末を所有するシステムまたは領域 (つまり、その他のシステムの TCT) で端末を認識するために使用される 1 から 4 文字の名前を指定します。このオペランドが省略された場合、TRMIDNT オペランド内の名前が使用されます。

SYSIDNT=name
この端末を所有するシステムまたは領域の 4 文字の名前を示します。これは、ローカル・システムまたはローカル領域の場合があります (つまり、TYPE=INITIAL マクロで定義された名前)。この場合、作成される TCT エントリーは、ローカル定義です。また、別のシステムまたは領域の名前である場合もあります。この場合、作成される TCT エントリーは、リモート定義です。この SYSIDNT は、このマクロに先行する TYPE=REGION マクロの SYSIDNT と同じでなければなりません。

TRMIDNT=name
この システム (つまり、この TCT を所有し、トランザクションを所有しているローカル・システム) で端末を認識するために使用される 1 から 4 文字の名前を指定します。

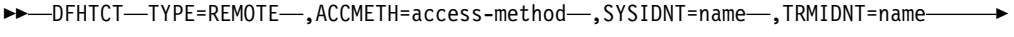
TRMTYPE=terminal-type
端末タイプを指定して、これをコーディングします。詳しくは、 462 ページの『順次装置』を参照してください。

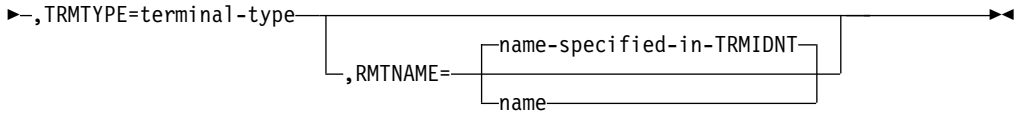
リモート端末、方法 2: DFHTCT TYPE=REMOTE

リモート・システムまたはリモート領域の端末入力は、DFHTCT TYPE=TERMINAL マクロ命令と DFHTCT TYPE=REGION マクロの組み合わせを使用して定義する代わりに、DFHTCT TYPE=REMOTE マクロを使用して CICS に定義することができます。

DFHTCT TYPE=REMOTE マクロの拡張は、現在参照されている領域とは独立しています。

注: ホーム領域が端末を所有していることを SYSIDNT オペランドが示している場合、DFHTCT TYPE=TERMINAL マクロのすべてのオペランドが DFHTCT TYPE=REMOTE マクロで有効になり、TYPE=TERMINAL と同じ意味を持ちます。ただし、(通常はそうであるように) SYSIDNT オペランドがリモート領域を示している場合、DFHTCT TYPE=TERMINAL の追加オペランドは、DFHTCT TYPE=REMOTE マクロで有効ですが無視されます。





ACCMETH=access-method
リモート端末のアクセス方式を指定して、これをコーディングします。

RMTNAME={name-specified-in-TRMIDNT|name}
端末を所有するシステムまたは領域 (つまり、その他のシステムの TCT) で端末を認識するために使用される 1 から 4 文字の名前を指定します。このオペランドが省略された場合、TRMIDNT オペランド内の名前が使用されます。

SYSIDNT=name
この端末を所有するシステムまたは領域の名前を指定します。この名前は、前の TYPE=SYSTEM マクロまたは TYPE=INITIAL マクロの SYSIDNTオペランドで使用されている名前と同じでなければなりません。

TRMIDNT=name
この システム (つまり、この TCT を所有し、トランザクションを所有しているローカル・システム) で端末を認識するために使用される 1 から 4 文字の名前を指定します。

TRMTYPE=terminal-type
端末タイプを指定して、これをコーディングします。詳しくは、 462 ページの『順次装置』を参照してください。

DFHTCT: CICS 端末リスト

このリリースの CICS は、直接的または間接的に幅広い端末と通信することができます。

新規または現行の端末は、z/OS Communications Server インターフェースに準拠している場合、CICS Transaction Server for z/OS によって直接サポートされます。

表 20 は、CICS でどのように端末がサポートされるかの要約を示しています。

表 20. CICS Transaction Server for z/OS でサポートされる *IBM* 端末およびシステム・タイプ

z/OS Communications Server を使用して **CICS Transaction Server for z/OS** によって直接サポートされる
3101 ディスプレイ端末
3230 プリンター
3268 プリンター
3270 情報表示システム
3270 PC
3270 PC/G
3270 PC/GX
3287 プリンター
3600 金融機関通信システム
3630 作業データ通信システム
3640 作業データ通信システム
3650 小売店システム
3680 プログラマブル小売店システム

表 *20. CICS Transaction Server for z/OS* でサポートされる *IBM* 端末およびシステム・タイプ *(*続き*)*

z/OS Communications Server を使用して **CICS Transaction Server for z/OS** によって直接サポートされる
3730 分散オフィス通信システム
3767 通信端末装置
3770 データ通信システム
3790 通信システム
4300 処理装置
4700 金融機関通信システム
5280 分散データ・システム
5520 管理システム
5550 管理システム
5937 堅牢な端末
6670 情報配信装置
8100 情報システム
8775 ディスプレイ端末
8815 Scanmaster
ディスプレイライター
パーソナル・コンピューター、PS/2、PS/55
System/32
System/34
System/36
System/38
iSeries
z Systems
テレタイプライター交換サービス (TWX 33/35)
ワールド・トレード・タイプライター型端末 (WTTY)

z/OS Communications Server LU

z/OS Communications Server for SNA でサポートされる LU の詳細なリスト、およびそれらを CICS に定義する方法については、 272 ページの『サポートされる装置』 を参照してください。

TLT: 端末リスト・テーブル

DFHTLT マクロ命令によって生成される端末リスト・テーブル (TLT) を使用することで、端末 ID およびオペレーター ID を論理的にグループ化することができます。

TLT:

- 操作の有効範囲を定義および制限するために CEST (監視端末トランザクション) で使用する場合は必須です。以下に例を挙げます。

  ```
  CEST SET TERMINAL(*) SUPRID(CG) OUTSERVICE
  ```

  DFHTLTCG で定義されたすべての端末をサービス休止状態に設定します。
- 操作を事前に決定された端末グループに適用するために、CEST または CEMT (マスター端末トランザクション) で使用することができます。(CEST 操作の場合、この TLT によって、SUPRID で指定された TLT のサブセットを定義する必要があります。) 例えば、以下のようなコマンドを使用します。

```
CEST SET TERMINAL(*) SUPRID(CG) CLASS(EM) INSERVICE
CEMT SET TERMINAL(*) CLASS(EM) INSERVICE
```

DFHTLTEM で定義されたすべての端末をサービス中状態に設定します。

- 単独あるいは他の TLT との組み合わせで使用して、メッセージ交換の事前定義された宛先を提供することができます。以下に例を挙げます。

```
CMSG ROUTE=PG,'PRODUCTION MEETING AT 11.00 IN
           ROOM 2124',SEND
```

DFHTLTPG で定義されたすべての端末またはオペレーターにメッセージを送信します。

メッセージ交換と監視端末機能あるいはマスター端末機能に同じ TLT を使用することができます。例えば、監視端末の制御下にある端末を定義する TLT を、それらの端末にメッセージを送信するための宛先リストとしても使用することができます。

一部の論理装置では、論理装置コード (LDC) 簡略記号 (各テーブル・エントリーに関連付けることが可能) が、メッセージ交換に使用されたり、マスター端末操作および監視端末操作で無視されたりすることがあります。

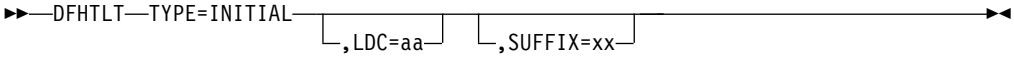
相互通信ネットワークでは、端末リスト・テーブル内のすべての端末が、テーブルが使用されているシステムによって所有されている必要があります。

以下のマクロは、TLT エントリーを定義します。

- 制御セクション: DFHTLT TYPE=INITIAL
- 端末リスト・テーブル内のエントリー: DFHTLT TYPE=ENTRY
- 端末リスト・テーブルの終了: DFHTLT TYPE=FINAL (TYPE=FINAL (テーブルの終わり) を参照)

制御セクション: DFHTLT TYPE=INITIAL

DFHTLT TYPE=INITIAL マクロは、定義する端末リスト・テーブルのエントリー・ポイントおよび開始アドレスを設定します。



注: TYPE=INITIAL マクロの一般情報については、TYPE=INITIAL (制御セクション) を参照してください。

LDC=aa

2 文字の論理装置コード (LDC) 簡略記号を指定して、これをコーディングします。これは、LDC 簡略記号が DFHTLT TYPE=ENTRY マクロで指定された場合の除き、すべての論理装置 ID に関連付けられます。

SUFFIX=xx

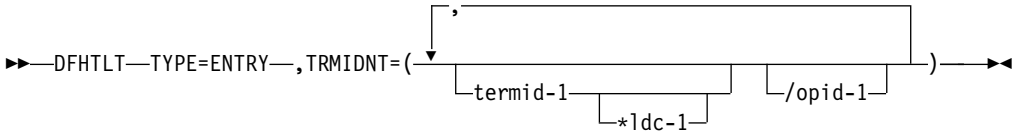
TLT のモジュール名は DFHTLT*xx* です。ここで、*xx* は 1 文字または 2 文字の接尾部です。これにより、使用される各 TLT に固有の ID が提供されます。TLT 内では名前 TLTBA、TLTBB、TLTBC、および TLTEA が使用されるため、接尾部 BA、BB、BC、および EA を使用してはなりません。

TLT には、メッセージ交換トランザクション CMSG で使用される接尾部が必要です。

端末リスト・テーブル内のエントリー: DFHTLT TYPE=ENTRY

DFHTLT TYPE=ENTRY は、端末リスト・テーブル (TLT) 内のエントリーを定義します。

エントリーは、TLT 内で以下のようにコーディングされます。



TYPE=ENTRY
このテーブルで 1 つ以上のエントリー (最大 1000 エントリーまで) を生成する場合は、これをコーディングします。

TRMIDNT=([termid-1[*ldc-1]] [/opid-1][, termid-2[*ldc-2][/opid-2],...])
スタート・ストップおよび BSC 端末のリスト、論理装置、オペレーター ID を指定して、これをコーディングします。論理装置 ID は、LDC 簡略記号によって修飾することができます。

termid
1 から 4 文字のスタート・ストップ、BSC 端末、または論理装置 ID を示します。

注: 通信コントローラーに接続された 3614 は、マスター端末あるいは監視端末の操作で使用することができますが、メッセージ交換操作では使用しないでください。(3614 は、メッセージ宛先として有効ではありません。)

ldc
2 文字の LDC 簡略記号を示します。この前にはアスタリスク (*) を付ける必要があり、追加先の 'termid' パラメーターの後に続けて指定して使用する必要があります。

opid
1 から 3 文字のオペレーター ID を示します。これには、先行スラッシュ (/) を付ける必要があります。

指定する端末またはオペレーター ID は、それぞれ DFHTCT マクロの TRMIDNT オペランドおよび外部セキュリティー・マネージャーでも指定する必要があります。(RACF を採用する場合は、ADDUSER コマンドの OPIDENT オペランドを使用して、各オペレーターの ID を記録します。) 指定する LDC 簡略記号は、DFHTCT TYPE=LDC マクロおよび DFHTCT TYPE=TERMINAL マクロの LDC オペランドでも指定する必要があります。

監視端末機能およびマスター端末機能は、TLT に含まれている端末および論理装置の ID を使用しますが、LDC 簡略記号の参照はすべて無視します。

DFHTLT の例

端末リスト・テーブル (TLT) の作成方法の例。

```
Example 1

DFHTLT TYPE=INITIAL,                                *
       SUFFIX=AA
DFHTLT TYPE=ENTRY,                                  *
       TRMIDNT=(NYC,CHI,LA,WDC)
DFHTLT TYPE=ENTRY,                                  *
       TRMIDNT=SF
DFHTLT TYPE=ENTRY,                                  *
       TRMIDNT=(BSTN/OP1,ATL/OP5,/OP9,DNVR)
DFHTLT TYPE=ENTRY,                                  *
       TRMIDNT=/OP6
DFHTLT TYPE=FINAL
END

Example 2

DFHTLT TYPE=INITIAL,                                *
       SUFFIX=XX
DFHTLT TYPE=ENTRY,                                  *
       TRMIDNT=(NYC,T361*LP,T362*LP/OP1)
DFHTLT TYPE=ENTRY,                                  *
       TRMIDNT=(T363/OP2,T364/OP5,T365)
DFHTLT TYPE=FINAL
END
```

図 *16.* 端末リスト・テーブル*:* 例

TST: 一時記憶域テーブル

一時記憶域テーブル (TST) は、一時記憶域キューのセットを識別するために使用される総称名 (または接頭部) のリストです。総称名と同じ文字で始まるアプリケーション・プログラムで動的に生成された一意の一時ストレージ ID に、TST 項目と同じプロパティーが自動的に与えられます。

CICS は、TSMODEL リソース定義との組み合わせで、あるいは TSMODEL リソース定義の代わりに、引き続き DFHTST マクロの使用をサポートします。以下の状況では、TST を使用する必要があります。

- EXEC CICS 一時記憶域コマンドで明示的に SYSID を指定することでキューを共用している一時記憶域データを参照するアプリケーション・プログラムがある。
- XTSEREQ グローバル・ユーザー出口プログラムによって、EXEC CICS 一時記憶域コマンドに SYSID が追加されている。
- TSAGE 属性を必要としている。

これらの機能を必要としない一時記憶域キューの場合、TSMODEL リソース定義を使用することができます。このリソース定義は、TST のその他のすべての機能およびいくつかの追加機能を提供します。

デフォルトの TST=NO システム初期設定パラメーターは、CICS が TS キュー用の RDO サポートのみを使用して初期設定を行うことを意味しています。 TST を TSMODEL リソース定義と組み合わせて使用するには、TST システム初期設定パラ

メーターを使用して TST 接尾部を指定する必要があります。また、MIGRATE オプションを指定して TST ロード・モジュールをアセンブルする必要もあります。MIGRATE オプションを指定せずに TST をアセンブルした場合は、CICS は TST のみをロードし、TS キューに対する RDO サポートは提供しません。TSMODEL リソース定義のインストール試行はすべて拒否されます。

TST リソース定義と TSMODEL リソース定義の両方を使用する場合、TST の使用は、以下のことだけに限定されます。

- TS API コマンドで指定した明示的な SYSID オプションによって参照される TS データ共用キューのサポート。
- TSAGE 属性。

TST を単独で使用する場合は、TST のすべての機能が使用されます。

総称名

TST では、総称名は、適切なキュー名の先行文字から形成され、長さは最大 7 文字です。

- DFHTST TYPE=RECOVERY マクロでコーディングされる総称名は、トランザクションの障害あるいはシステム保護の障害が発生したときに CICS が変更のバックアウトを提供する対象のキューを識別します。
- DFHTST TYPE=REMOTE マクロでコーディングされる総称名は、リモート・システム名 (SYSIDNT) がローカル CICS の名前と同じである場合を除いて、CICS が一時記憶域要求をリモート CICS 領域または TS サーバーに転送する対象のキューを識別します。SYSIDNT がローカル CICS と同じ名前である場合、DATAID オプションによって指定されたキューは、CICS によってローカル・キューとして処理されます。
- DFHTST TYPE=LOCAL マクロでコーディングされる総称名は、TST がインストールされている CICS 領域内にあるローカル・キューとしてキューを識別します。
- DFHTST TYPE=SECURITY マクロでコーディングされる総称名は、リソース・セキュリティー検査が必要なキューを識別します。

8 文字の名前を指定した場合、これは、固有の一時記憶域キュー名を定義します。

数個の総称名を使用して多くのキューを定義することが可能なキュー名の命名規則を選択してください。これにより、TST 定義のタスクが大幅に軽減されます。CICS は、特定の検索基準を満たす最初の接頭部について TST を検索するので注意してください。例えば、CICS が一時記憶域キュー ABCDEFGH を検索する場合に、TST に接頭部 A と接頭部 AB があり、AB が A の後にある場合、A が選択されます。問題を回避するために、汎用度の高いエントリーの前に汎用度の低いエントリーを TST に定義することで、可能性があるすべての一致の中で最も汎用度の低いものが最初に検索されるようにします。

CICS が TS キュー名に対してマッチングする総称名を探す場合、CICS は、その特定の検索に関連するエントリー・タイプのみを検索します。CICS は、以下を検索します。

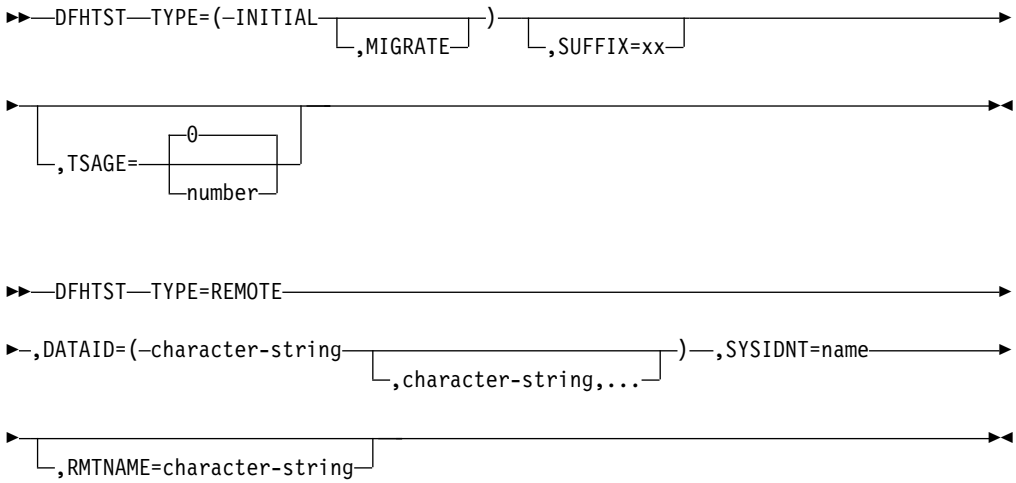
- キューがリモートであるかを判別する場合は、ローカル・エントリーおよびリモート・エントリー。したがって、CICS が総称名に対してキュー名をマッチングする場合、ローカル・エントリーとリモート・エントリーは、1 つの検索カテゴリーであると見なされます。
- キューがリカバリー可能であるかを判別する場合は、リカバリー・エントリーおよびリモート・エントリー。ただし、キュー名の先行文字が TYPE=RECOVERY 総称名と TYPE=REMOTE 総称名の両方に一致する場合は TYPE=REMOTE が優先され、キューが存在するローカル領域内でリカバリー・オプションを再定義する必要があります。(共用 TS プール内のキューがリカバリー可能になることはありません。)
- キューがセキュリティーの対象であるかを判別する場合は、セキュリティー・エントリーのみ。

TST エントリーを定義するには、以下のマクロを使用します。

- 制御セクション: DFHTST TYPE=INITIAL
- リカバリー可能一時記憶域: DFHTST TYPE=RECOVERY
- ローカル一時記憶域: DFHTST TYPE=LOCAL
- リモート一時記憶域: DFHTST TYPE=REMOTE
- 一時記憶域セキュリティー検査: DFHTST TYPE=SECURITY
- 一時記憶域データ共用: DFHTST TYPE=SHARED
- 一時記憶域テーブルの終了: DFHTST TYPE=FINAL (TYPE=FINAL (テーブルの終わり) を参照)

制御セクション: DFHTST TYPE=INITIAL

定義する一時記憶域テーブルのエントリー・ポイントおよび開始アドレスは、DFHTST TYPE=INITIAL マクロによって設定されます。



SUFFIX オペランドの使用など、TYPE=INITIAL マクロに関する一般情報については、TYPE=INITIAL (制御セクション)を参照してください。

MIGRATE

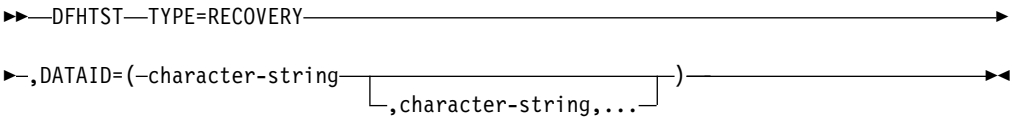
CSD ファイルへのマイグレーションのために TST をアセンブルする場合、あ

るいは TSMODEL リソース定義との組み合わせで TST を使用している場合は、MIGRATE を指定します。TST システム初期設定パラメーターを使用して TST 接尾部を指定すると、MIGRATE オプションを指定して TST をアセンブルした場合に、CICS が TSMODEL リソース定義も処理します。MIGRATE オプションを指定せずに TST をアセンブルした場合は、CICS は TST のみをロードし、TS キューに対する RDO サポートは提供しません。TSMODEL リソース定義のインストール試行はすべて拒否されます。

TSAGE={0|number}
CICS の緊急時再始動中に一時記憶域ドメインによって使用される一時記憶域データの経過時間制限を定義します。指定された期間に参照されていないデータは、リカバリーされません。この値は日単位で指定され、最大値は 512 です。値がゼロの場合、この基準でデータをパージしないことを示します。

リカバリー可能一時記憶域: DFHTST TYPE=RECOVERY

DFHTST TYPE=RECOVERY マクロは、リカバリーが利用可能な一時記憶域キューに使用される総称名を指定します。



TYPE=RECOVERY
リカバリー可能な一時記憶域キュー名を識別するには、これをコーディングします。一時記憶域キュー名が、リモート DATAID および リカバリー DATAID の両方で定義されている場合は、リモートであると見なされます。 リカバリー可能であるかどうかは、キューがローカルである CICS 領域でのみ指定することができます。

注: 以前のリリースとの互換性のために、TYPE=ENTRY は保持されており、意味は TYPE=RECOVERY とまったく同じです。

DATAID=(*character-string[,character-string,...]*)|()
リカバリー可能にしたい 1 つ以上の TS キュー名を英数字で指定して、これをコーディングします。各名前の最大長は 8 文字です。 (総称名およびマッチング基準については、 475 ページの『TST: 一時記憶域テーブル』 を参照してください。)

character-string
各文字ストリングは、総称キュー名あるいは固有の TS キュー名を表すことができます。総称名は、TS キュー名の 1 から 7 文字の先行文字を使用して指定されます。 8 文字すべてを使用する DATAID は、固有のキュー名を定義します。

複数の名前は、コンマで区切り、括弧で囲む必要があります。 1 つの名前のみを指定する場合は、括弧を省略することができます。 CICS によって生成された TS キュー名の一部には、以下のようにリカバリーについて考慮する必要があるものがあります。

- **DF** は、REQID を指定しないデータありの START コマンドにおいて CICS インターバル制御機能によって使用される一時記憶域キューを表します。
- ** は、BMS ROUTE コマンドで使用される一時記憶域キュー、および PAGING オペランドを使用するコマンドを表します。
- **$$** は、START TRANSID コマンドで PROTECT=YES オプション が指定されている場合に、BMS CMSG トランザクションによって使用される一時記憶域キューを表します。

() CICS は、括弧の間に値が指定されていないこの特殊 (ヌル) オペランドを、他の DATAID では明示的に指定されていないキューが指定されていると解釈します。

注**:**

1. TYPE=RECOVERY エントリーなしで TST が生成された場合、リカバリー処理は実行されません。FROM、RTRANSID、RTERMID、または QUEUE のいずれかのパラメーターを指定して EXEC CICS START コマンドを発行し、REQID が指定されていない場合、CICS は、接頭部「DF」から始まる要求 ID を生成します。これらの要求に対してリカバリーが必要な場合、対応する総称名を使用して TST を生成する必要があります。
2. 再始動可能なトランザクション (トランザクション・リソース定義で RESTART(YES) を指定して定義されたトランザクション) によって使用されるすべての一時記憶域キューは、リカバリー可能にする必要があります (デフォルトの DF 接頭部を持つものを含む)。
3. 補助記憶域上のデータのみをリカバリー可能にすることができます。主記憶域に書き込まれたデータは、指定するリカバリー・オプションに関係なく、リカバリー不能です。
4. リカバリー可能として指定された一時記憶域データがタスクによって変更された場合、キュー名に対してエンキューすることで、並行タスクによる変更からデータを保護します。タスクが終了するか、タスク同期点要求が発行されて作業論理単位の終了が指定されるまでは、キュー名はデキューされません。この時点でログ・レコードがシステム・ログ・データ・セットに書き込まれ、それ以降にシステムが異常終了した場合にデータをリカバリーするのに十分な外部情報を提供します。

例

以下の DFHTST TYPE=RECOVERY マクロは、リカバリー可能一時記憶域キューを定義します。

```
DFHTST TYPE=RECOVERY,
       DATAID=(DF,**,
               $$(,character-string)...)
```

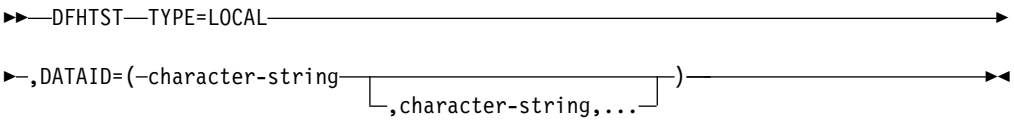
- DATAID DF は、CICS 開始要求で使用される一時記憶域キューをリカバリー可能にします。
- DATAID ** および $$ は、BMS によって使用されるデフォルトの一時記憶域キューをリカバリー可能にします。

- DATAID 文字ストリングは、リカバリー可能にしたい各一時記憶域キュー ID の先行文字を表します。例えば、DATAID=(R,ZIP) は、文字「R」または文字「ZIP」から始まる ID を持つすべての一時記憶域キューをリカバリー可能にします。

ローカル一時記憶域: DFHTST TYPE=LOCAL

DFHTST TYPE=LOCAL マクロは、TST がインストールされているローカル CICS 領域内にある一時記憶域キュー名を定義します。このマクロを使用することで、SYSIDNT が不明であってもローカル・キューを定義することが可能になります (詳しくは、DFHTST TYPE=REMOTE マクロの SYSIDNT オプションを参照)。

リモート・キューおよび共用キューの TYPE=REMOTE マクロで指定された全総称 DATAID と組み合わせて使用することで、このマクロは、ローカル・キューおよびリモート・キューを定義するタスクを大幅に簡素化するのに役立ちます。



TYPE=LOCAL
この TST エントリーが、ローカル一時記憶域キューのセットを定義することを示します。

DATAID=(*character-string[,character-string,...]***)|()**
1 つ以上の TS キュー名を英数字で指定して、これをコーディングします。各名前の最大長は 8 文字です。

character-string
各文字ストリングは、総称キュー名あるいは固有の TS キュー名を表すことができます。通常、総称名は、TS キュー名の 1 から 7 文字の先行文字を使用して指定されます。8 文字すべてを使用する DATAID は、固有のキュー名を定義します。

複数の名前は、コンマで区切り、括弧で囲む必要があります。1 つの名前のみを指定する場合は、括弧を省略することができます。

() CICS は、括弧の間に値が指定されていないこの特殊 (ヌル) オペランドを、他の DATAID では明示的に指定されていないキューが指定されていると解釈します。これを、以下のように包括的な指定として使用することができます。

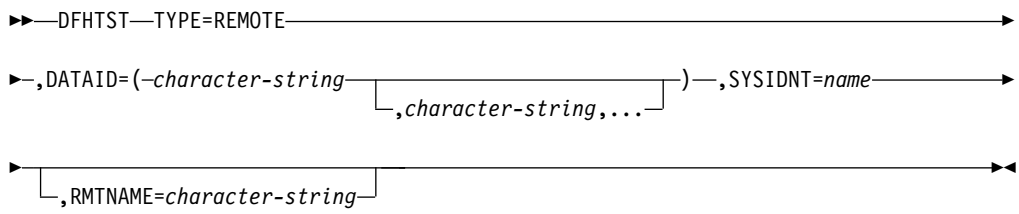
- 別の領域あるいは共用 TS プール内にある特定のキューが、適切な総称 DATAID を使用して TYPE=REMOTE マクロで指定されている場合、TYPE=LOCAL マクロで DATAID=() を指定することで、その他のすべてのキューをローカルとして定義することができます。

TYPE=LOCAL マクロにおけるこのヌル・オプションは、TYPE=REMOTE マクロの DATAID=() と相互に排他的です。このオプションが、ローカル・エントリーとリモート・エントリーの両方で指定された場合、TST マクロはアセンブリー・エラーを返します。したがって、ローカル TS キュー・エントリーで DATAID=() を指定する場合、TYPE=LOCAL マクロは、すべての TYPE=REMOTE マクロの後に指定する必要があります。

リモート一時記憶域: DFHTST TYPE=REMOTE

DFHTST TYPE=REMOTE マクロは、CICS 相互通信機能が使用されているときに、リモート CICS 領域内にある一時記憶域キュー名を定義します。

このマクロは、共用キュー・プール内にあるキューを定義する場合にも使用します。共用キュー・プールは、リモート・システムの名前が DFHTST TYPE=SHARED マクロで指定されたシステム名と一致することを除き、リモート領域と同様に処理されます。



TYPE=REMOTE
この TST エントリーが、リモート CICS 領域内、あるいはカップリング・ファシリティーの共用 TS プール内にあるリモート一時記憶域キューのセットを定義することを示します。

DATAID=(*character-string[,character-string,...]***)|()**
1 つ以上の TS キュー名を英数字で指定して、これをコーディングします。各名前の最大長は 8 文字です。TS キュー名の 1 から 7 文字の先行文字を使用し、要求をリモート領域または TS サーバーに転送する対象のキューの総称名を形成します。 (総称名およびマッチング基準については、 475 ページの『TST: 一時記憶域テーブル』 を参照してください。)

注: RMTNAME を指定する場合、リスト形式の DATAID オペランドを使用することはできません。 RMTNAME パラメーターを指定する場合、DATAID の構文は DATAID=*character-string* です。

character-string
各文字ストリングは、総称キュー名あるいは固有の TS キュー名を表すことができます。通常、総称名は、TS キュー名の 1 から 7 文字の先行文字を使用して指定されます。総称名は、この TST がインストールされている領域内のアプリケーション・プログラムによって使用されます。

複数の名前は、コンマで区切り、括弧で囲む必要があります。 1 つの名前のみを指定する場合は、括弧を省略することができます。

() CICS は、括弧の間に値が指定されていないこの特殊オペランドを、他の DATAID では明示的に指定されていないキューが指定されていると解釈します。これを、以下のように包括的な指定として使用することができます。

- 文字 L、M、および N から始まる名前を持つキューがローカルであり、これらが適切な総称 DATAID を使用して TYPE=LOCAL マクロで指定されている場合、以下のように TYPE=REMOTE マクロで DATAID=() を指定することで、その他のすべてのキューをリモートとして定義することができます。

```
        DFHTST TYPE=LOCAL,       *
               DATAID=(L,M,N)
*
        DFHTST TYPE=REMOTE,      *
               DATAID=()
```

TYPE=REMOTE マクロの DATAID=() オプションは、TYPE=LOCAL マクロの DATAID=() と相互に排他的です。このオプションが、ローカル・エントリーとリモート・エントリーの両方で指定された場合、TST マクロはアセンブリー・エラーを返します。

DATAID=() は、一連のローカル・エントリーおよびリモート・エントリーの中で最後のエントリーでなければなりません。したがって、リモート TS キューで DATAID=() を使用する場合、TYPE=REMOTE マクロは、すべての TYPE=LOCAL マクロの後に指定する必要があります。

SYSIDNT=*name*

リモート・キューまたは共用一時記憶域キューがある領域あるいはサーバーを識別します。他の CICS 領域が所有するリモート・キューの場合、4 文字の英数字で指定する名前は、CONNECTION 定義で指定された REMOTENAME オプション (IPCONN 定義の IPCONN 名の最初の 4 文字または DFHTST TYPE=SHARED エントリーで指定された SYSIDNT 名) と同じでなければなりません。

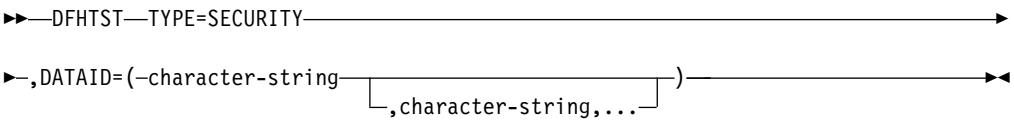
このパラメーターを使用して、TST がインストールされているローカル領域の名前を指定することができます。この SYSIDNT オペランドがシステム初期設定パラメーターで指定された SYSIDNT と一致する場合、DATAID と一致する TS キューは、ローカル・キューとして処理されます。

RMTNAME=*character-string*

一時記憶域キューへの参照がリモート・システムまたはリモート領域に送信された場合に、CICS が DATAID オペランドで指定された名前と置き換えるために使用する 1 から 8 文字の接頭部を指定して、これをコーディングします。このオペランドのデフォルトは、DATAID オペランドで指定された文字ストリングです。このオペランドで指定する文字ストリングの長さは、DATAID オペランドで指定された文字ストリングの長さと同じでなければなりません。このメカニズムにより、リモート・システム内の一時記憶域キューに、ローカル・システム内の一時記憶域キューと同じ名前を使用してアクセスすることが可能になります。

一時記憶域セキュリティー検査: DFHTST TYPE=SECURITY

DFHTST TYPE=SECURITY マクロは、TST で指定された一時記憶域キューに対するセキュリティー検査が必要であることを示します。



TYPE=SECURITY

この TST エントリーが、セキュリティー検査が必要な一時記憶域キューのセットを定義することを示します。

DATAID=(*character-string[,character-string,...]***)|()**

1 つ以上の TS キュー名を英数字で指定して、これをコーディングします。各名前の最大長は 8 文字です。キュー名の先行文字からの 1 から 7 文字の先行文字を使用して、セキュリティー検査の対象となるキューの総称名を形成します。 (総称名およびマッチング基準については、 475 ページの『TST: 一時記憶域テーブル』 を参照してください。)

注:

1. このマクロを使用する場合は、TSQ へのアクセスを制御するために、適切なプロファイルを外部セキュリティー・マネージャーに定義する必要があります。そうしないと、このマクロは意図した効果を発揮しません。プロファイルについては、主記憶域内のリソース・プロファイルのリフレッシュ を参照してください。
2. TSQ のフルネームがセキュリティー・マネージャーに渡されます。

character-string

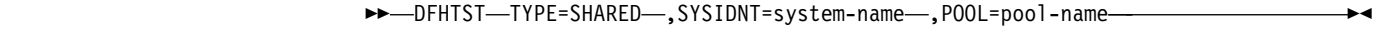
各文字ストリングは、総称キュー名あるいは固有の TS キュー名を表すことができます。通常、総称名は、TS キュー名の 1 から 7 文字の先行文字を使用して指定されます。総称名は、この TST がインストールされている領域内のアプリケーション・プログラムによって使用されます。

複数の名前は、コンマで区切り、括弧で囲む必要があります。 1 つの名前のみを指定する場合は、括弧を省略することができます。

() CICS は、括弧の間に値が指定されていないこのヌル・オペランドを、他の DATAID では明示的に指定されていないキューが指定されていると解釈します。

一時記憶域データ共用: DFHTST TYPE=SHARED

DFHTST TYPE=SHARED マクロは、CICS がカップリング・ファシリティー内の一時記憶域プールを識別するために使用するリモート・システム名を指定します。



TYPE=SHARED

この TST エントリーが、TYPE=REMOTE エントリーで指定されたシステム ID (SYSIDNT) と TS データ共用キューのプールの間のマッピングを定義することを示します。

SYSIDNT=*system_name*

TS プール名に対応する 1 から 4 文字のシステム名を指定します。

CICS は、以下のようにこの SYSIDNT を使用して、リモート・キュー (TYPE=REMOTE エントリーで定義、または API コマンドでの明示的 SYSID) を TS サーバーにマップします。

- API 一時記憶域コマンドで (SYSID オプションによって) 明示的にリモート・キューが指定された場合、CICS は、その SYSID を一致する TYPE=SHARED エントリーの SYSIDNT にマップします。
  - 一致する SYSIDNT が見つかった場合、CICS は、対応する POOL 名を使用して、共用 TS キューを管理する TS サーバーを識別します。

- SYSID がどの TYPE=SHARED エントリーにも一致しない場合、要求は SYSID で指定されたリモート・キュー所有領域 (QOR) に機能シップされます。

- API 一時記憶域コマンドが TYPE=REMOTE エントリーによって識別されるリモート・キューを参照している場合、CICS は、TYPE=SHARED エントリーに一致する SYSIDNT がないかを検査します。
  - 一致する SYSIDNT を持つ TYPE=SHARED エントリーが見つかった場合、CICS は、対応する POOL 名を使用して、共用 TS キューを管理する TS サーバーを識別します。
  - TYPE=SHARED エントリーが見つからない場合、キューはリモート・キューであり、要求は QOR に機能シップされます。

  異なる SYSIDNT 名を使用して、同じ POOL 名を参照する複数の TYPE=SHARED エントリーを作成することができます。この場合、同じキュー名を参照するというのは、使用する (API での) SYSID に関係なく、同じキュー名を参照するということです。

POOL=*pool_name*

暗黙的あるいは明示的に、対応するシステム名を指定する TS 要求に対して使用される TS キューのプール名を 1 から 8 文字で指定します。プール名は、TS プールを管理する TS サーバーの POOL パラメーターで指定された名前と一致する必要があります。

DFHTST の例

一時記憶域テーブル (TST) のコーディング方法の例

```
     DFHTST TYPE=INITIAL,              LIST OF GENERIC NAMES OF QUEUES *
            SUFFIX=01                  THAT ARE RECOVERABLE, REMOTE,
*                                      SHARED, LOCAL, OR REQUIRE
*                                      SECURITY CHECKING.
*
*  The following macro specifies that all LOCAL queues with
*  names beginning with the letter 'R' are RECOVERABLE:
*
     DFHTST TYPE=RECOVERY,                                             *
            DATAID=R
*
*  The following macro specifies that queues with names
*  beginning with C,D,E, and X are local queues:
*
     DFHTST TYPE=LOCAL,                                                *
            DATAID=(C,D,E,X)
*
*  The following macro specifies that queues with names
*  beginning with AB,L,M,N are remote queues on system RSYS:
*
     DFHTST TYPE=REMOTE,                                               *
            DATAID=(AB,L,M,N),                                         *
            SYSIDNT=RSYS,                                              *
*
*  The next macro specifies that all queues not local as defined
*  above, or remote in system RSYS as defined above, are remote
*  queues that reside in a shared TS pool TYPE=SHARE macro.
*
     DFHTST TYPE=REMOTE,                                               *
            DATAID=(),                                                 *
            SYSIDNT=SHR1
*
*  The next macro specifies that remote queues with SYSIDNT=SHR1
*  are mapped to shared TS pool named TSQSHR1.
*
     DFHTST TYPE=SHARED,                                               *
            SYSIDNT=SHR1,                                              *
            POOL=TSQSHR1
*
*  The following macro specifies that queues with names
*  beginning with SAQ require security checking.
*    Note that the full TS queue name is passed to the ESM.
*
     DFHTST TYPE=SECURITY,                                             *
            DATAID=SAQ
*
     DFHTST TYPE=FINAL
     END
```

図 *17.* 一時記憶域テーブル: 例

XLT: トランザクション・リスト・テーブル

トランザクション・リスト・テーブルは、システムの終了時に端末から開始できるトランザクション、および一緒に使用可能/使用不可にするトランザクションのグループを指定します。XLT に優先して TRANSACTION リソースを使用します。

XLT を使用して、以下を定義することができます。

- システム終了の最初の静止ステージ中に端末から開始することができるトランザクション ID のリスト。実行する PLT プログラムがない場合、最初の静止時間

が短くなる可能性があるため、2 番目の静止ステージが始まる前に XLT プログラムに入る時間がほとんどなくなります。 XLT システム初期設定パラメーターを使用して、使用するテーブルの接尾部を指定します。マスター端末オペレーターは、**CEMT PERFORM SHUTDOWN** コマンドの XLT オプションを使用して、システム終了時に接尾部を変更することができます。

注**:** 代わりに、PROGRAM リソースを作成してトランザクション・リスト・テーブルを定義することができます。これをプログラムとして定義することは、自動インストールすることができることを意味します。プログラムの自動インストールについては、『Configuring』の『Autoinstalling programs, map sets, and partition sets』 を参照してください。

- マスター端末から使用不可/使用可能にするトランザクション ID のグループ。マスター端末オペレーターは、CEMT SET TRANSACTION の CLASS オプションを使用して、使用するテーブルの接尾部を指定します。

488 ページの図 18 は、XLT を作成するためのコーディングを示しています。

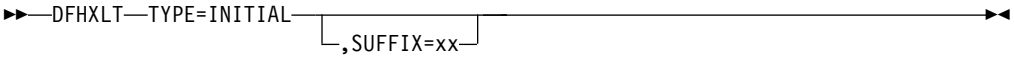
XLT エントリーを定義する場合、以下のマクロが使用可能です。

- 制御セクション: DFHXLT TYPE=INITIAL
- トランザクション・リスト・テーブル内のエントリー: DFHXLT TYPE=ENTRY
- トランザクション・リスト・テーブルの終了: DFHXLT TYPE=FINAL (TYPE=FINAL (テーブルの終わり) を参照)

制御セクション: DFHXLT TYPE=INITIAL

DFHXLT TYPE=INITIAL マクロは、定義する XLT のエントリー・ポイントおよび開始アドレスを設定します。

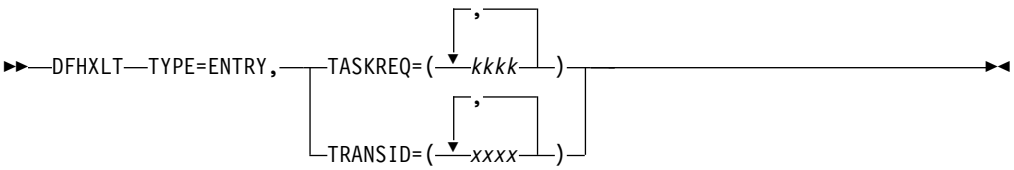
DFHXLT TYPE=INITIAL マクロは、定義する XLT のエントリー・ポイントおよび開始アドレスを設定します。



SUFFIX オペランドの使用など、TYPE=INITIAL マクロに関する一般情報については、TYPE=INITIAL (制御セクション)を参照してください。

トランザクション・リスト・テーブル内のエントリー: DFHXLT TYPE=ENTRY

DFHXLT TYPE=ENTRY マクロは、システム終了の最初の静止ステージ中に端末から開始することができるトランザクション ID のリストを指定します。



TYPE=ENTRY

XLT で 1 つ以上のエントリーを生成する場合に、これをコーディングします。

TASKREQ=(kkkk[,kkkk],...)

kkkk は、以下のいずれかです。

- PA1 から PA3、および PF1 から PF24 は、タスクを開始するために使用できる特殊な 3270 キーの 1 つを示します。
- LPA (ライト・ペン・アテンション) は、ライト・ペン検出可能フィールドが選択された場合にトランザクションが開始されることを示します。
- OPID (オペレーター ID カード・リーダー) は、適切なオペレーターの ID バッジが読み取られた場合にトランザクションが開始されることを示します。
- MSRE は、10/63 文字磁気スロット読取装置が使用された場合にトランザクションが開始されることを示します。

CSD ファイルで各 TASKREQ を定義し、それを稼働中のシステムにインストールします。 (詳細については、TRANSACTION 属性で TASKREQ 属性の説明を参照してください。)

TRANSID=(xxxx[,xxxx],...)

1 から 4 文字のトランザクション・コードを表します。 CSD ファイルで各 TRANSID を定義し、それを稼働中のシステムにインストールします。 (詳細については、TRANSACTION 属性で TRANSACTION 属性の説明を参照してください。)

TRANSID に特殊文字 (コンマなど) が含まれている場合、TYPE=ENTRY 命令には、引用符を区切り文字と使用して、TRANSID を 1 つだけ含めることができます。

注: TASKREQ および TRANSID は、相互に排他的なパラメーターです。

DFHXLT の例

トランザクション・リスト・テーブル (XLT) のコーディング例。

```
   DFHXLT TYPE=INITIAL,                    LIST OF TRANSACTIONS     *
          SUFFIX=IN                        THAT ARE ACCEPTED
*                                          DURING THE FIRST QUIESCE
*                                          PHASE OF SYSTEM
*                                          TERMINATION.
   DFHXLT TYPE=ENTRY,TASKREQ=PF5           (TASKREQ MUST ALSO BE
*                                          DEFINED IN THE CSD AND
*                                          INSTALLED IN THE RUNNING
*                                          CICS SYSTEM. AN ENTRY FOR
*                                          THE XLT MUST BE MADE IN
   DFHXLT TYPE=ENTRY,TRANSID=(USR1,USR2)   THE CSD.)
   DFHXLT TYPE=ENTRY,TRANSID='AA,1'
   DFHXLT TYPE=ENTRY,TRANSID='AA,2'
   DFHXLT TYPE=FINAL
   END

   DFHXLT TYPE=INITIAL,                    LIST OF LOGICALLY RELATED*
          SUFFIX=G1                        TRANSIDS TO BE ENABLED OR
*                                          DISABLED BY MASTER
*                                          TERMINAL.
   DFHXLT TYPE=ENTRY,TRANSID=(TSSA,TSRA)   (TRANSIDS MUST ALSO BE
   DFHXLT TYPE=ENTRY,TRANSID=(TDSA,TDRA)   DEFINED IN THE CSD AND
   DFHXLT TYPE=ENTRY,TRANSID=ICSA          INSTALLED IN THE RUNNING
   DFHXLT TYPE=FINAL                       CICS SYSTEM.)
   END
```

図 *18.* トランザクション・リスト・テーブル*:* 例

第 6 章 互換性のために保持されている廃止された属性

いくつかのリソース定義属性は、CICS Transaction Server for z/OS, バージョン 5 リリース 5 で使用されるリソースには適用されませんが、まだ有効である以前のリリースの CICS に対する CSD 互換性を提供するためにサポートされています。

互換モードについては、異なる CICS リリース間での CSD の共用を参照してください。

以下の属性は、この CICS リリースで使用されるリソースには適用されません。使用している以前の CICS リリースの資料を参照して、属性がそのリリースでサポートされているかを確認してください。

表 *21.* 廃止されたリソース定義属性

Resource	属性	説明
CONNECTION	BINDPASSWORD	APPC バインド時のセキュリティー・パスワード
FILE	RESSECNUM	リソース・セキュリティー値
FILE	PASSWORD	ファイル・アクセス検査
LSRPOOL	LSRPOOLID	プール用の数値 ID (1 から 8)
MAPSET	RSL	リソース・セキュリティー値
PARTITIONSET	RSL	リソース・セキュリティー値
PROFILE	PROTECT	メッセージ保全性とメッセージ・ロギング
PROGRAM	HOTPOOL	Java ホット・プーリング
PROGRAM	JVMPROFILE	プールされた JVM 用の JVM プロファイル
PROGRAM	RSL	リソース・セキュリティー値
セッション	INSERVICE	セッションを使用可能
セッション	OPERID	3 文字のオペレーター ID
セッション	OPERPRIORITY	オペレーター優先順位コード
セッション	OPERRSL	リソース・セキュリティー・キー
セッション	OPERSECURITY	事前設定トランザクション・セキュリティー・キー
セッション	RECOVNOTIFY	XRF テークオーバーの通知
セッション	TRANSACTION	指定されたトランザクションのみを許可
TCPIPSERVICE	DNSGROUP	CICS がワークロード・マネージャーに登録するグループ名を指定
TCPIPSERVICE	GRPCRITICAL	DNS グループの重要なメンバーとしてサービスにマーク
TCPIPSERVICE	PRIVACY	必要な SSL 暗号化のレベル
TCPIPSERVICE	TSQPREFIX	TS キュー接頭部
TERMINAL	BINDPASSWORD	APPC バインド時のセキュリティー・パスワード

© Copyright IBM Corp. 1982, 2018

表 21. 廃止されたリソース定義属性 (続き)

Resource	属性	説明
TERMINAL	CONSOLE	コンソール識別番号
TERMINAL	OPERID	3 文字のオペレーター ID
TERMINAL	OPERPRIORITY	オペレーター優先順位コード
TERMINAL	OPERRSL	リソース・セキュリティー・キー
TERMINAL	OPERSECURITY	事前設定トランザクション・セキュリティー・キー
TERMINAL	XRFSIGNOFF	XRF テークオーバー後の強制サインオフ
TRANSACTION	EXTSEC	外部セキュリティー・マネージャーの使用
TRANSACTION	INDOUBT	異常終了時のアクション
TRANSACTION	PRIMEDSIZE	プライム状態のストレージ割り振りサイズ
TRANSACTION	RSL	リソース・セキュリティー値
TRANSACTION	TCLASS	タスクに割り当てられたクラス
TRANSACTION	TRANSEC	トランザクション・セキュリティー値
TYPETERM	LOGMODECOM	LOGMODE 互換性
TYPETERM	XRFSIGNOFF	XRF テークオーバー後の強制サインオフ

第 7 章 CICS 提供のリソースの定義、グループ、およびリスト

IBM は、CICS 領域にインストールする必要のあるリソースの定義、およびサンプル・アプリケーション・プログラムで使用されるリソースの定義を提供します。

CICS 提供のトランザクションについて詳しくは、CICS トランザクションのリストを参照してください。

DFHCSDUP INITIALIZE コマンドを使用して、CSD ファイルを初期化します。初期化の後に、次の 2 つのカテゴリーのリソース定義グループが CSD ファイルに組み込まれます。

1. DFHLIST という名前のグループ。RDO やその他の CICS 提供トランザクションを使用する際に不可欠です。
2. サンプル・アプリケーション・プログラム用の定義のグループ。詳しくは、500 ページの『サンプル・アプリケーション・プログラム・グループ』を参照してください。

IBM が提供する CICS トランザクション、TYPETERM 定義、モデル TERMINAL 定義、および PROFILE 定義は、以下の 4 つのグループに入っています。

- DFHTYPE: TYPETERM 定義
- DFHTERM: 自動インストール用のモデル TERMINAL 定義
- DFHISC: システム間通信セッション用の PROFILE 定義
- DFHSTAND: PROFILE 定義

DFHLIST 定義

DFHLIST には、必要なリソースの定義が含まれます。

表 *22. DFHLIST* リソース定義

グループ名	説明	プログラム	トランザクション	その他のリソース
DFH$SOT	提供される TCPIPSERVICE リソース			**TCP/IP** サービス**:** ECI HTTPNSSL HTPSSL
DFHBMS	基本マッピング・サポート	DFHTPQ DFHTPR DFHTPS	CSPG CSPQ CSPS	
DFHBR	ブリッジ・プログラム	DFHL3270 DFHBRMP DFHBRCV		
DFHCBTS	BTS のローカル要求キュー・ファイル **注:** このグループは、ロックによって保護されません。必要に応じて、含まれる定義を変更することができます。			ファイル**:** DFHLRQ
DFHCFC	CICS C++ ファウンデーション・クラスに必要なプログラム	ICCFCDLL		
DFHCLNT	CICS クライアント CTIN および CCIN	DFHZCT1 DFHZCN1	CTIN CCIN	トランザクション・クラス**:** DFHCOMCL
DFHCONS	CPU コンソールへの書き込み	DFHCWTO	CWTO	

© Copyright IBM Corp. 1982, 2018

表 22. *DFHLIST* リソース定義 (続き)

グループ名	説明	プログラム	トランザクション	その他のリソース
DFHDBCTL	DBCTL transactions	DFHDBAT DFHDBCON DFHDBCT DFHDBDI DFHDBDSC DFHDBIQ DFHDBME DFHDBMP DFHDBUEX	CDBC CDBD CDBI CDBM CDBN CDBO CDBT	ファイル: DFHDBFK マップ・セット: DFHDBIE DFHDBNE
DFHDB2	Db2 サポート	DFHD2CM0 DFHD2CM1 DFHD2CM2 DFHD2CM3 DFHD2EDF DFHD2EX1 DFHD2EX2 DFHD2INI DFHD2PXT DFHD2SPS DSNCUEXT DSNTIAC DSNTIA1	CDBF CDBQ CEX2 DSNC	

表 22. *DFHLIST* リソース定義 (続き)

グループ名	説明	プログラム	トランザクション	その他のリソース
DFHDCTG	基本 CICS 機能の TD キュー **注:** このグループは、ロックによって保護されません。必要に応じて、含まれる定義を変更することができます。			**TD** キュー: CADL CADO CADS CAIL CCPI CCSE CCSO CCZM CDB2 CDBC CDEP CDUL CECO CEJL CEPO CESE CESO CIEO CIIL CISL CISO CJRM CKQQ CMIG CMPO CMQM CMLO CPIO CRDI CRLO CRPO CSBA CSBR CSCC CSCS CSDH CSDL CSFL CSJE CSJO CSKL CSLB CSML CSMT CSNE CSOO CSPL CSQL CSRL CSSH CSSL CSTL CSZL CSZX CWBO CWBW

表 22. *DFHLIST* リソース定義 (続き)

グループ名	説明	プログラム	トランザクション	その他のリソース
DFHDOC	CICS 文書ハンドラー・テンプレート・リーダー	DFHDHEI		
DFHDP	アプリケーション・デバッグ・プロファイル・マネージャー 3270 インターフェース	DFHDPLU DFHDPIN DFHDPCP	CADP CIDP	マップ・セット: DFHDPMS
DFHDPWB	アプリケーション・デバッグ・プロファイル・マネージャー Web インターフェース	DFHDPWB DFHDPWM0 DFHDPWM1 DFHDPWM2 DFHDPWM3 DFHDPWM4 DFHDPWM5 DFHDPWM6 DFHDPWT0 DFHDPWT1 DFHDPWT2 DFHDPWT3 DFHDPWT4 DFHDPWT5 DFHDPWT6 DFHDPWF0		
DFHEDF	実行診断機能	DFHDBMS DFHDBTI DFHEDFBR DFHEDFD DFHEDFP DFHEDFR DFHEDFX DFHEIGDS DFHEITAB	CEBR CEDF CEDG CEDX CEDY	マップ・セット: DFHEDFM トランザクション・クラス: DFHEDFTC DFHEDFTO
DFHEDP	EXEC DLI ユーザー出口	DFHEDP		
DFHEP	イベント処理	DFHECEAH DFHECEAM DFHECEAS DFHECEAT	CEPH CEPQ CEPS CEPT	プロファイル: DFHECEPH
DFHFCRL	バンドル内のファイルの無効化	DFHFCRN	CFCR	
DFHFE	FE 端末テスト機能	DFHFEP DFHTRAP	CSFE	
DFHFEPI	フロントエンド・プログラミング・インターフェース	DFHEITSZ DFHSZRMP	CSZI	
DFHHARDC	3270 プリンター - z/OS Communications Server	DFHP3270	CSPP	
DFHINDT	未確定テスト・ツール	DFHINDAP DFHINDT DFHINTRU	CIND	トランザクション・クラス: DFHTCIND
DFHINQUI	コマンド定義	DFHEITBS		
DFHINTER	コマンド・インタープリター	DFHECID DFHECIP DFHECSP	CECI CECS	
DFHIPECI	ECI over TCP/IP	DFHIEP	CIEP	
DFHISC	システム間連絡	DFHCCNV DFHCHS DFHCLS3 DFHCLS4 DFHCLS5 DFHCNV DFHCRNP DFHCRQ DFHCRR DFHCRS DFHCRSP DFHCRT DFHDFST DFHDSRP DFHDYP DFHLUP DFHMIRS DFHMXP DFHRTC DFHRTE DFHSHRRP DFHSHRSP DFHUCNV DFHZLS1	CDFS CEHP CEHS CLQ2 CLR2 CLS1 CLS2 CLS3 CLS4 CMPX CPMI CQPI CQPO CRSQ CRSR CRTE CRTX CSHR CSMI CSM1 CSM2 CSM3 CSM5 CSNC CSSF CVMI CXRT	プロファイル: DFHCICSF DFHCICSR DFHCICSS

表 22. *DFHLIST* リソース定義 (続き)

グループ名	説明	プログラム	トランザクション	その他のリソース
DFHISCIP	IP 相互接続サポート	DFHCIS4 DFHISAIP DFHISCIP DFHISCOP DFHISDIP DFHISEMP DFHISLQP DFHISIP DFHISPHP DFHISPRP DFHISREU DFHISREX DFHISRRP DFHISRSP	CISB CISC CISD CISE CISM CISP CISR CISQ CISS CIST CISU CISX CIS1 CIS4	プロファイル: DFHCICSC **TS** モデル: DFHISLQ
DFHJAVA	Java サポートに必要なプログラム	DFHSJJI DFHSJITL DFHSJTHP	CJSA CJSR CJSU	
DFHLGMOD	CICS ログ・マネージャー			ジャーナル・モデル: DFHLOG DFHSHUNT DFHLGLOG
DFHLGQC	CICS ログ・マネージャーの静止	DFHLGQC	CSQC	
DFHMISC	その他のプログラム	DFHLETRU DFHPEP DFHREST		
DFHMQ	WebSphere MQ サポートに必要なプログラム、トランザクション、およびマップ・セット	CSQCAPX CSQAVICM CSQCBDCI CSQCBE30 CSQCBP10 CSQCBP53 CSQCBP00 CSQCBP10 CSQCBR00 CSQCBR53 CSQCCB CSQCCLOS CSQCCONN CSQCCONX CSQCCTL CSQCDISC CSQCDSC CSQCDSPL CSQCQCON CSQCSSQ CSQCGET CSQCINQ CSQCOPEN CSQCPUT CSQCPUT1 CSQCRST CSQCSET CSQCSSQ CSQCSUB CSQCSUBR CSQFCTAB DFHMQBAS DFHMQBP0 DFHMQBP1 DFHMQBP2 DFHMQBP3 DFHMQBR0 DFHMQCOD DFHMQCON DFHMQCTL DFHMQDCI DFHMQDIS DFHMQDSC DFHMQDSL DFHMQMON DFHMQPLT DFHMQPOP DFHMQPRM DFHMQPUL DFHMQQCN DFHMQRET DFHMQRS DFHMQSSQ DFHMQTRU DFHMQTSK IMQB23IC IMQS23IC	CKAM CKBC CKBM CKBP CKBR CKCN CKDL CKDP CKQC CKRS CKRT CKSD CKSQ CKTI	マップ・セット: DFHMQHC DFHMQHE DFHMQHK DFHMQHU DFHMQ1C DFHMQ1E DFHMQ1K DFHMQ1U DFHMQ2C DFHMQ2E DFHMQ2K DFHMQ2U
DFHMSWIT	メッセージ交換プログラム	DFHMSP	CMSG	

表 22. *DFHLIST* リソース定義 (続き)

グループ名	説明	プログラム	トランザクション	その他のリソース
DFHOPCLS	動的オープン/クローズ・プログラム	DFHFCU	CSFU	
DFHOPER	オペレーター・プログラム	DFHCEMNA DFHCEMNB DFHCEMNC DFHCEMND DFHCETRA DFHCETRB DFHCETRC DFHCETRD DFHCETRF DFHECBAM DFHEITMT DFHEITOT DFHEITST DFHEMTA DFHEMTD DFHEMTP DFHEOTP DFHESTP DFHLDMAP DFHSOCRL	CBAM CCRL CEMN CEMT CEOT CEST CETR CLDM	マップ・セット: DFHSO1M DFHCMNH DFHCMNM DFHCTRH DFHCTRM
DFHOTS	OTS 再同期	DFHOTR	CJTR	
DFHPGAIP	プログラムの自動インストール	DFHPGADX DFHPGAHX DFHPGALX DFHPGAOX DFHPGAPG		マップ・セット: DFHPGAMP 区分セット: DFHPGAPT
DFHPIPE	Web サービス処理	DFHJSON DFHMLBSJ DFHMLBST DFHPIAP DFHPIDSH DFHPIDSQ DFHPIEP DFHPIIR DFHPILSQ DFHPIR DFHPIRT DFHPISN1 DFHPISN2 DFHPITE DFHPITP DFHPITQ1 DFHWSADH DFHWSSE1 DFHWSXXX C128N IOSTREAM IXM4C57 IXMI33UC IXMI33DA IXMI33IN	CPIA CPIH CPIL CPIS	
DFHPIVAL	SOAP メッセージ・バリデーター **注:** このグループは、ロックによって保護されません。必要に応じて、含まれる定義を変更することができます。	DFHPIVAL		
DFHPSSGN	持続セッション・サインオンの保存	DFHZSGN DFHZPCT DFHZRPT	CPSS CPCT CRTP	
DFHRL	リソース・ライフサイクル管理	DFHRLMF DFHRLR DFHRLSC DFHRLVC	CRLR	
DFHRMI	リソース・マネージャー・インターフェース	DFHRMSY	CRSY	
DFHRQS	要求ストリーム結合プログラム	DFHRZJN		
DFHRS	領域状況、ファイル作成	DFHRSFDL		
DFHRSEND	z/OS Communications Server 再送プログラム	DFHZRSP	CSRS	
DFHSDAP	シャットダウン支援	DFHCESD	CESD	
DFHSIGN	プログラムおよびテーブルのサインオン/サインオフ	DFHCESC DFHCEGN DFHSFP DFHSNP	CESC CEGN CESF CESN CESL	マップ・セット: DFHSNLE DFHSNPE DFHSNSE
DFHSO	CICS ソケット・インターフェースのエミュレート	DFHSOCI DFHSOLI		

表 22. *DFHLIST* リソース定義 (続き)

グループ名	説明	プログラム	トランザクション	その他のリソース
DFHSPI	オンライン・リソース定義	DFHAMP DFHDMP DFHEDAD DFHEDAP DFHEITSP DFHPUP DFHTBS DFHTOR DFHZATA DFHZATD DFHZATDX DFHZATMD DFHZATMF DFHZATR DFHZATS DFHZCQ DFHZCTDX DFHZDTDX DFHZPTDX	CATA CATD CATR CATS CDTS CEDA CEDA CEDB CEDC CFTS CITS CMTS CRMD CRMF	
DFHSTAND	標準 CICS アプリケーション・プログラム	DFHACP DFHCXCU DFHPIITL DFHPIPA DFHPIXC DFHPSIP DFHQRY DFHSTP DFHTACP DFHTEP DFHTEPT DFHTFP DFHZXCU DFHZXRE DFHZXST	CPIR CQRY CSAC CSTE CXCU CXRE	プロファイル: DFHCICSA DFHCICSE DFHCICSP DFHCICST DFHCICSV DFHPPF01 DFHPPF02 マップ・セット: DFHXMSG
DFHTCL	互換性 TRANCLASS 定義			トランザクション・クラス: DFHTCL00 DFHTCL00 DFHTCL01 DFHTCL03 DFHTCL04 DFHTCL05 DFHTCL06 DFHTCL07 DFHTCL08 DFHTCL09 DFHTCL10 DFHTCLQ2 DFHTSDEL
DFHTERM	モデル TERMINAL 定義			端末: LU2 LU3 APPC SCSP 3270 3284 L0E2 L0M2 L0M3 L0M4 L0M5 L2E2 L2M2 L2E3 L2M3 L2E4 L2M4 L2M5

表 22. *DFHLIST* リソース定義 (続き)

グループ名	説明	プログラム	トランザクション	その他のリソース
DFHTYPE	TYPETERM 定義			入力条件: DFHCONS DFHLU2 DFHLU3 DFHLU62T DFHSCSP DFH3270 DFH3270P DFHLU0E2 DFHLU0M2 DFHLU0M3 DFHLU0M4 DFHLU0M5 DFHLU2E2 DFHLU2M2 DFHLU2E3 DFHLU2M3 DFHLU2E4 DFHLU2M4 DFHLU2M5
DFHVTAM	z/OS Communications Server プログラム	DFHGMM DFHZNAC DFHZNEP	CSGM CSNE	
DFHVTAMP	z/OS Communications Server 端末管理プリントの主要機能	DFHCPY DFHEXI DFHPRK DFHRKB	CSCY CSPK CSRK	
DFHWEB	CICS Web サポート定義	DFH$WB1A DFHWBA DFHWBAAX DFHWBADX DFHWBAHX DFHWBALX DFHWBAOX DFHWBA1 DFHWBBLI DFHWBCLI DFHWBENV DFHWBEP DFHWBERX DFHWBGB DFHWBIMG DFHWBIP DFHWBLT DFHWBPA DFHWBPW DFHWBPW1 DFHWBPW2 DFHWBPW3 DFHWBPW4 DFHWBST DFHWBTC DFHWBTL DFHWBTRU DFHWBTTA DFHWBTTB DFHWBTTC DFHWBUN DFHWBXN	CWBA CWBC CWBG CWXN CWXU	**TSMODEL:** DFHWEB 文書テンプレート: DFHWBPW1 DFHWBPW2 DFHWBPW3 DFHWBPW4
DFHWEB2	Web 2.0 定義	DFHW2A DFHW2FD DFHW2TS	CW2A	

表 22. *DFHLIST* リソース定義 (続き)

グループ名	説明	プログラム	トランザクション	その他のリソース
DFHWU	CICS 管理クライアント・インターフェース	DFHWUIPG DFHWUIPI DFHWUIP1 DFHWUIP3 DFHWUIP4 DFHWUIP5	CWWU CWGQ	文書テンプレート: DFHWUIPI DFHWUIP1 DFHWUIP3 DFHWUIP4 DFHWUIP5
注: DFHTYPE TYPETERM 定義は、z/OS Communications Server が提供する LOGMODE 定義と一致します。提供された z/OS Communications Server LOGMODES を使用しない場合は、DFHTYPE TYPETERM 定義を変更する必要がある場合があります。 z/OS Communications Server LOGMODE 定義のプログラミング情報については、Coding entries in the VTAM LOGON mode tableを参照してください。				

DFHLIST にない CICS 提供グループ

CICS が提供するすべてのリソース定義グループがリスト DFHLIST で定義されているわけではありません。

表 23. *DFHLIST* にないリソース定義

グループ名	説明	プログラム	トランザクション	その他のリソース
DFHBRCF	DFHBRNSF のカップリング・ファシリティー・データ・テーブル・サンプル			ファイル: DFHBRNSF
DFHBRUT	DFHBRNSF のユーザー保守データ・テーブル・サンプル			ファイル: DFHBRNSF
DFHBRVR	DFHBRNSF の VSAM RLS サンプル			ファイル: DFHBRNSF
DFHBRVSL	DFHBRNSF の VSAM 非 RLS ローカル・サンプル			ファイル: DFHBRNSF
DFHBRVSR	DFHBRNSF の VSAM 非 RLS リモート・サンプル			ファイル: DFHBRNSF
DFHCMAC	CICS オンライン・メッセージおよびコード	DFHCMAC	CMAC (別名 CHLP)	マップ・セット: DFHCMCM ファイル: DFHCMACD
注: DFHCMAC は、以前のリリースとの互換性の理由から、DFHLIST に含まれていません。オンライン・メッセージおよびコードのトランザクションを使用する場合、開始グループ・リストに DFHCMAC を追加する必要があります。				
DFHMISC3	その他のグループ	DFHNET		
注: DFHMISC3 は、以前のリリースとの互換性の理由から、DFHLIST に含まれていません。ご使用のシステムで必要な場合は、開始グループ・リストに追加してください。				

表 23. *DFHLIST* にないリソース定義 (続き)

グループ名	説明	プログラム	トランザクション	その他のリソース
DFHRPC	リモート・プロシージャー・コール	DFHRPAL DFHRPAS DFHRPC00 DFHRPMS DFHRPRP DFHRPTRU	CRPA CRPC CRPM	マップ・セット: DFHRP0
DFHTERMC	コンソールの自動インストールに使用するモデル TERMINAL 定義			端末: AUTC

CICS 提供の互換性グループ

CICS は、以前のリリースとの互換性のために RDO グループを提供します。

CICS システム定義データ・セット (CSD) のアップグレード後にその CSD を以前のリリースの CICS と共用することを予定している場合は、開始グループ・リストに適切な DFHCOMP*x* 互換性グループを含める必要があります。

異なるリリースの CICS 間での CSD の共用について、および以前のリリース用にどの DFHCOMP*x* グループを含める必要があるかについて詳しくは、『アップグレード』の『異なる CICS リリース間での CSD の互換性』を参照してください。

CSD ファイルより上のリリース・レベルで実行されている CICS 領域とは、その CSD ファイルを共用しないでください。

サンプル・アプリケーション・プログラム・グループ

以下のリソース定義は、CICS で提供されるサンプル・アプリケーション・プログラムを実行するために必要です。このグループは DFHLIST では名前が指定されていません。

表 24. *CICS* サンプル・アプリケーション - リソース定義

グループ名	言語	説明	定義されたリソース
DFH$AFLA	アセンブラー	FILEA サンプル・アプリケーション	マップ・セット: DFH$AGA DFH$AGB DFH$AGC DFH$AGD DFH$AGK DFH$AGL プログラム: DFH$AALL DFH$ABRW DFH$ACOM DFH$AMNU DFH$AREN DFH$AREP トランザクション: AADD ABRW AINQ AMNU AORD AORQ AREP AUPD

表 24. *CICS* サンプル・アプリケーション - リソース定義 (続き)

グループ名	言語	説明	定義されたリソース
DFH$BMSP	COBOL および PL/I	BMS 区画サポート・アプリケーション	区分セット: DFH0PS マップ・セット: DFH0CGP DFH$PGP プログラム: DFH0CPKO DFH0CPLA DFH$PPKO DFH$PPLA トランザクション: PPKO PPLA XPKO XPLA
DFH$CFLA	COBOL	FILEA サンプル・アプリケーション	マップ・セット: DFH0CGA DFH0CGB DFH0CGC DFH0CGD DFH0CGK DFH0CGL プログラム: DFH0CALL DFH0CBRW DFH0CCOM DFH0CMNU DFH0CREN DFH0CREP トランザクション: ADDS BRWS INQY MENU OREN OREQ REPT UPDT
DFH$CNSL		サンプル・コンソール定義	入力条件: DFH$JCLC DFH$CONS 端末: CJCL CNSL CN02

表 24. *CICS* サンプル・アプリケーション - リソース定義 (続き)

グループ名	言語	説明	定義されたリソース
DFH$CTXT	COBOL	CUA テキスト・モデル・アプリケーション	ファイル: DFH0FCAI DFH0FCUS DFH0FHLP マップ・セット: DFH0AB DFH0ABT DFH0BRW DFH0DEL DFH0FPD DFH0HLP DFH0HP DFH0HPD DFH0LST DFH0NEW DFH0OPN DFH0PRT DFH0SAS DFH0T1 DFH0UPD プログラム: DFH0VAB DFH0VABT DFH0VBRW DFH0VDEL DFH0VDQ DFH0VHLP DFH0VHP DFH0VLIO DFH0VLST DFH0VNEW DFH0VOL DFH0VOPN DFH0VPRT DFH0VRIO DFH0VSAS DFH0VTBL DFH0VT1 DFH0VUPD プロファイル: DFH$CUA2 トランザクション AC2A AC2C AC2D AC2E AC2F AC20 AC21 AC22 AC23 AC24 AC25 AC26 AC27 AC28 DELQ
DFH$DFLA	C	FILEA サンプル・アプリケーション	マップ・セット: DFH$DGA DFH$DGB DFH$DGC DFH$DGD DFH$DGK DFH$DGL プログラム: DFH$DALL DFH$DBRW DFH$DCOM DFH$DMNU DFH$DREN DFH$DREP トランザクション: DADD DBRW DINQ DMNU DORD DORQ DREP DUPD
DFH$DLIV		IMS インストール検査手順	プログラム: DFH$DLAC DFH$DLAE DFH$DLCC DFH$DLCE DFH$DLPC DFH$DLPE トランザクション ASMC ASME COBC COBE PLIC PLIE

表 24. *CICS* サンプル・アプリケーション - リソース定義 (続き)

グループ名	言語	説明	定義されたリソース
DFH$EXBS		カタログ・アプリケーション・サンプル	ファイル**:** EXMPCAT EXMPCONF マップ・セット: DFH0XS1 DFH0XS2 DFH0XS3 プログラム: DFH0XCMN DFH0XGUI DFH0XODE DFH0XSDS DFH0XSOD DFH0XSSM DFH0XVDS DFH0XWOD トランザクション: ECFG EGUI
DFH$EXCI		EXCI バッチ呼び出しインターフェース・サンプル	接続: EXCG EXCS プログラム: DFH$AXCS DFH$AXVS セッション: EXCG EXCS トランザクション: EXCI HPJC
DFH$EXWS		カタログ・アプリケーション・サンプル	プログラム: DFH0XCUI トランザクション: ECLI **TCP/IP** サービス: EXMPPORT パイプライン: EXPIPE01 EXPIPE02
DFH$FILA		FILEA サンプル	ファイル**:** FILEA
DFH$ICOM	アセンブラー	システム間連絡 (ISC)	マップ・セット: DFH$IGB DFH$IGC DFH$IGS DFH$IGX DFH$IG1 DFH$IG2 プログラム: DFH$ICIC DFH$IFBL DFH$IFBR DFH$IMSN DFH$IMSO DFH$IQRD DFH$IQRL DFH$IQRR DFH$IQXL DFH$IQXR トランザクション: ICIC IFBL IFBR IMSN IMSO IQRD IQRL IQRR IQXL IQXR
DFH$NACT		NACT サンプル・アプリケーション	ファイル: ACCTFILE ACCTNAM ACINUSE マップ・セット: DFH0MNA プログラム: DFH0CNA1 DFH0CNA2 DFH0CNA3 DFH0CNA4 DFH0CNA5 トランザクション: NACP NACT

表 24. *CICS* サンプル・アプリケーション - リソース定義 (続き)

グループ名	言語	説明	定義されたリソース
DFH$OSGI		JVM サーバーを使用した Java サポート用のサンプル・アプリケーション	**BUNDLE**: DFH$OSGB **JVM** サーバー: DFHJVMS プログラム: DFH$JSAM DFH$LCCA DFJ$JHE1 DFJ$JHE2 DFJ$JPC1 DFJ$JPC2 DFJ$JTD1 DFJ$JTS1 DFJ$JTSC **TDQUEUE**: JTD1 トランザクション: JHE1 JHE2 JPC1 JPC2 JTD1 JTS1
DFH$PFLA	PL/I	FILEA サンプル・アプリケーション	マップ・セット: DFH$PGA DFH$PGB DFH$PGC DFH$PGD DFH$PGK DFH$PGL プログラム: DFH$PALL DFH$PBRW DFH$PCOM DFH$PMNU DFH$PREN DFH$PREP トランザクション: PADD PBRW PINQ PMNU PORD PORQ PREP PUPD
DFH$STAT		統計	プログラム: DFH$STED
DFH$SXP		メッセージ・ドメイン出口	プログラム: DFH$SXP1 DFH$SXP2 DFH$SXP3 DFH$SXP4 DFH$SXP5 DFH$SXP6
DFH$UTIL	アセンブラー	一時データ・ユーティリティーの動的割り振り	プログラム: DFH$TDWT DFH99 トランザクション: ADYN TDWT
DFH$VTAM		端末定義	入力条件: DFH$L77 DFH$L78 DFH$L79 DFH$L86 端末: L77C L77D L78A L79A L86A
DFH$WLP		WebSphere Application Server Liberty サンプル定義	**JVM** サーバー: DFHWLP バンドル: JDBC WLPJDBC WLPHELLO WLPTSQ
DFHAI62		スターター APPC 接続	プログラム: DFHZATDY 接続: CBPS CBSS CCPS セッション: CBPS CBSS CCPS

表 24. CICS サンプル・アプリケーション - リソース定義 (続き)

グループ名	言語	説明	定義されたリソース
DFHMROAR		スターター MRO システム	接続: CICD CICT セッション: CICSRD CICSRT
DFHMRODR		スターター MRO システム	接続: CICA セッション: CICSRA
DFHMROFA		スターター MRO システム	ファイル: FILEA マップ・セット: DFH$AGA DFH$AGB DFH$AGC DFH$AGD DFH$AGK DFH$AGL プログラム: DFH$AALL DFH$ABRW DFH$ACOM DFH$AMNU DFH$AREN DFH$AREP トランザクション: AADD ABRW AINQ AMNU AORD AORQ AREP AUPD
DFHMROFD		スターター MRO システム	ファイル: FILEA
DFHMROFT		スターター MRO システム	トランザクション: AADD ABRW AINQ AMNU AORD AORQ AREP AUPD
DFHMROTR		スターター MRO システム	接続: CICA セッション: CICSRA

グループ DFHTYPE 内の TYPETERM 定義

CICS が提供する CSD グループ DFHTYPE には、TYPETERM 定義が含まれています。

DFHCONS

MVS コンソール:

```
TYPETERM(DFHCONS)  GROUP(DFHTYPE)
DEVICE(CONSOLE)
PAGESIZE(1,124)    AUTOPAGE(NO)
BRACKET(YES)       BUILDCHAIN(YES)     ROUTEDMSGS(NONE)
UCTRAN(YES)
```

DFHLU2

SNA 論理装置タイプ 2 (3270 ディスプレイ):

```
TYPETERM(DFHLU2)   GROUP(DFHTYPE)
DEVICE(LUTYPE2)    TERMMODEL(2)
DEFSCREEN(24,80)   PAGESIZE(24,80)     AUTOPAGE(NO)
BRACKET(YES)       BUILDCHAIN(YES)     ROUTEDMSGS(ALL)
```

```
AUDIBLEALARM(YES)  EXTENDEDDS(YES)     UCTRAN(YES)
SENDSIZE(1536)     RECEIVESIZE(256)    IOAREALEN(256,4000)
ERRLASTLINE(YES)   ERRINTENSIFY(YES)   ATI(YES)  TTI(YES)
DISCREQ(YES)       RELREQ(YES)         AUTOCONNECT(YES)
LOGONMSG(YES)      QUERY(ALL)          CREATESESS(NO)
```

この定義は3278モデル2ディスプレイ用です。3178, 3179, 3277, 3278, 3279, 3290, 3270PC, 3270PC/G, 3270PC/GX, 8775,および5550の装置に適しています。

DFHLU3

SNA 論理装置タイプ 3 (3270 プリンター):

```
TYPETERM(DFHLU3)   GROUP(DFHTYPE)
DEVICE(LUTYPE3)    TERMMODEL(2)
DEFSCREEN(24,80)   PAGESIZE(24,80)     AUTOPAGE(YES)
BRACKET(YES)       BUILDCHAIN(NO)      ROUTEDMSGS(ALL)
SENDSIZE(256)      RECEIVESIZE(256)    IOAREALEN(512,0)
EXTENDEDDS(YES)    QUERY(ALL)          ATI(YES)  TTI(YES)
DISCREQ(YES)       RELREQ(YES)         AUTOCONNECT(YES)
LOGONMSG(NO)                           CREATESESS(NO)
```

この定義は3287プリンター用です。 3262, 3268, 3284, 3286, 3287, 3288, 3289, および5550の装置に適しています。

DFHSCSP

SNA 論理装置タイプ 1 (3270 SCS プリンター):

```
TYPETERM(DFHSCSP)  GROUP(DFHTYPE)
DEVICE(SCSPRINT)
PAGESIZE(24,80)    AUTOPAGE(YES)
BRACKET(YES)       BUILDCHAIN(NO)      ROUTEDMSGS(ALL)
SENDSIZE(256)      RECEIVESIZE(256)    IOAREALEN(512,0)
EXTENDEDDS(YES)    QUERY(ALL)          ATI(YES)  TTI(YES)
DISCREQ(YES)       RELREQ(YES)         AUTOCONNECT(YES)
LOGONMSG(NO)                           CREATESESS(NO)
```

この定義は3287プリンター用です。 3262, 3268, 3287, 3289,および5550の装置に適しています。

DFH3270

ローカル接続された (非 SNA) 3270 ディスプレイ:

```
TYPETERM(DFH3270)  GROUP(DFHTYPE)
DEVICE(3270)       TERMMODEL(2)
DEFSCREEN(24,80)   PAGESIZE(24,80)     AUTOPAGE(NO)
BRACKET(YES)       BUILDCHAIN(NO)      ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)  EXTENDEDDS(YES)     UCTRAN(YES)
SENDSIZE(0)        RECEIVESIZE(0)   IOAREALEN(512,0)
ERRLASTLINE(YES)   ERRINTENSIFY(YES)   ATI(YES)  TTI(YES)
DISCREQ(YES)       RELREQ(YES)         AUTOCONNECT(YES)
LOGONMSG(YES)      QUERY(ALL)          CREATESESS(NO)
```

この定義は3278モデル2ディスプレイ用です。 3178, 3179, 3277, 3278, 3279,および3290の 装置に適しています。

DFH3270P

ローカル接続された (非 SNA) 3270 プリンター:

```
TYPETERM(DFH3270P) GROUP(DFHTYPE)
DEVICE(3270P)      TERMMODEL(2)
DEFSCREEN(24,80)   PAGESIZE(24,80)     AUTOPAGE(YES)
BRACKET(YES)       BUILDCHAIN(NO)      ROUTEDMSGS(ALL)
SENDSIZE(0)        RECEIVESIZE(0)      IOAREALEN(512,0)
```

```
EXTENDEDDS(YES)    QUERY(ALL)          ATI(YES)  TTI(YES)
DISCREQ(YES)       RELREQ(YES)         AUTOCONNECT(YES)
LOGONMSG(NO)                           CREATESESS(NO)
```

この定義は3284モデル2プリンター用です。 3262, 3268, 3284, 3287, 3288, 3289, および5550の装置に適しています。

DFHLU62T

SNA 論理装置タイプ 6.2 (APPC) 単一セッション端末:

```
TYPETERM(DFHLU62T) GROUP(DFHTYPE)
DEVICE(APPC)
                   PAGESIZE(1,40)      AUTOPAGE(YES)
BRACKET(YES)       BUILDCHAIN(YES)     ROUTEDMSGS(NONE)
SENDSIZE(2048)     RECEIVESIZE(2048)   IOAREALEN(0,0)
                                       ATI(YES) TTI(YES)
LOGONMSG(NO)                           CREATESESS(NO)
```

この定義はAPPC単独セッション端末用であり、 DISPLAYWRITER, SCANMASTER,およびSYSTEM/38の 装置に適しています。

DFHLU0E2

拡張データ・ストリームを使用する非 SNAモデル 2 (照会):

```
TYPETERM(DFHLU0E2) GROUP(DFHTYPE)
DEVICE(3270)       TERMMODEL(2)        SHIPPABLE(YES)
DEFSCREEN(24,80)   PAGESIZE(24,80)     AUTOPAGE(NO)
ALTSCREEN(0,0)     ALTPAGE(0,0)
BRACKET(YES)       BUILDCHAIN(NO)      ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)  EXTENDEDDS(YES)     UCTRAN(YES)
RECEIVESIZE(0)     SENDSIZE(0)         IOAREALEN(512,0)
ERRASTLINE(YES)    ERRINTENSIFY(YES)   QUERY(ALL)
DISCREQ(YES)       RELREQ(YES)         AUTOCONNECT(NO)
LOGONMSG(YES)      CREATESESS(NO)
ATI(YES)           TTI(YES)
```

拡張データ・ストリームを使用する非SNAモデル2（照会）。この定義は、z/OS Communications Server が提供する LOGMODE NSX32702 と一致します。

DFHLU0M2

非 SNA モデル 2:

```
TYPETERM(DFHLU0M2)  GROUP(DFHTYPE)
DEVICE(3270)        TERMMODEL(2)        SHIPPABLE(YES)
DEFSCREEN(24,80)    PAGESIZE(24,80)     AUTOPAGE(NO)
ALTSCREEN(0,0)      ALTPAGE(0,0)
BRACKET(YES)        BUILDCHAIN(NO)      ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)   EXTENDEDDS(NO)      UCTRAN(YES)
RECEIVESIZE(0)      SENDSIZE(0)         IOAREALEN(512,0)
ERRLASTLINE(YES)    ERRINTENSIFY(YES)   QUERY(NO)
DISCREQ(YES)        RELREQ(YES)         AUTOCONNECT(NO)
LOGONMSG(YES)       CREATESESS(NO)
ATI(YES)            TTI(YES)
```

拡張データ・ストリームを使用する非SNAモデル2（照会）。この定義は、z/OS Communications Server が提供する LOGMODE D4B32782 と一致します。

DFHLU0M3

非 SNA モデル 3:

```
TYPETERM(DFHLU0M3)  GROUP(DFHTYPE)
DEVICE(3270)        TERMMODEL(2)         SHIPPABLE(YES)
DEFSCREEN(24,80)    PAGESIZE(24,80)      AUTOPAGE(NO)
ALTSCREEN(32,80)    ALTPAGE(32,80)
BRACKET(YES)        BUILDCHAIN(NO)       ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)   EXTENDEDDS(NO)       UCTRAN(YES)
RECEIVESIZE(0)      SENDSIZE(0)          IOAREALEN(512,0)
ERRLASTLINE(YES)    ERRINTENSIFY(YES)    QUERY(NO)
DISCREQ(YES)        RELREQ(YES)          AUTOCONNECT(NO)
LOGONMSG(YES)       CREATESESS(NO)
ATI(YES)            TTI(YES)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4B32783 と一致します。

DFHLU0M4

非 SNA モデル 4:

```
TYPETERM(DFHLU0M4)     GROUP(DFHTYPE)
DEVICE(3270)           TERMMODEL(2)          SHIPPABLE(YES)
DEFSCREEN(24,80)       PAGESIZE(24,80)       AUTOPAGE(NO)
ALTSCREEN(43,80)       ALTPAGE(43,80)
BRACKET(YES)           BUILDCHAIN(NO)        ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)      EXTENDEDDS(NO)        UCTRAN(YES)
RECEIVESIZE(0)         SENDSIZE(0)           IOAREALEN(512,0)
ERRLASTLINE(YES)       ERRINTENSIFY(YES)     QUERY(NO)
DISCREQ(YES)           RELREQ(YES)           AUTOCONNECT(NO)
LOGONMSG(YES)          CREATESESS(NO)
ATI(YES)               TTI(YES)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4B32784 と一致します。

DFHLU0M5

非 SNA モデル 5:

```
TYPETERM(DFHLU0M5)     GROUP(DFHTYPE)
DEVICE(3270)           TERMMODEL(2)          SHIPPABLE(YES)
DEFSCREEN(24,80)       PAGESIZE(24,80)       AUTOPAGE(NO)
ALTSCREEN(27,132)      ALTPAGE(27,132)
BRACKET(YES)           BUILDCHAIN(NO)        ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)      EXTENDEDDS(NO)        UCTRAN(YES)
RECEIVESIZE(0)         SENDSIZE(0)           IOAREALEN(512,0)
ERRLASTLINE(YES)       ERRINTENSIFY(YES)     QUERY(NO)
DISCREQ(YES)           RELREQ(YES)           AUTOCONNECT(NO)
LOGONMSG(YES)          CREATESESS(NO)
ATI(YES)               TTI(YES)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4B32785 と一致します。

DFHLU2E2

拡張データ・ストリームを使用する SNA LU タイプ 2 モデル 2 (照会):

```
TYPETERM(DFHLU2E2)     GROUP(DFHTYPE)
DEVICE(LUTYPE2)        TERMMODEL(2)          SHIPPABLE(YES)
DEFSCREEN(24,80)       PAGESIZE(24,80)       AUTOPAGE(NO)
ALTSCREEN(0,0)         ALTPAGE(0,0)
BRACKET(YES)           BUILDCHAIN(YES)       ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)      EXTENDEDDS(YES)       UCTRAN(YES)
RECEIVESIZE(1024)      SENDSIZE(3840)        IOAREALEN(256,4000)
ERRLASTLINE(YES)       ERRINTENSIFY(YES)     QUERY(ALL)
DISCREQ(YES)           RELREQ(YES)           AUTOCONNECT(NO)
LOGONMSG(YES)          CREATESESS(NO)
ATI(YES)               TTI(YES)
```

拡張データ・ストリームを使用する SNA LU タイプ 2 モデル 2 (照会)。この定義は、z/OS Communications Server が提供する LOGMODE SNX32702 と一致します。

DFHLU2M2

SNA LU タイプ 2 モデル 2:

```
TYPETERM(DFHLU2M2)     GROUP(DFHTYPE)
DEVICE(LUTYPE2)        TERMMODEL(2)           SHIPPABLE(YES)
DEFSCREEN(24,80)       PAGESIZE(24,80)        AUTOPAGE(NO)
ALTSCREEN(0,0)         ALTPAGE(0,0)
BRACKET(YES)           BUILDCHAIN(YES)        ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)      EXTENDEDDS(NO)         UCTRAN(YES)
RECEIVESIZE(1024)      SENDSIZE(1536)         IOAREALEN(256,4000)
ERRLASTLINE(YES)       ERRINTENSIFY(YES)      QUERY(NO)
DISCREQ(YES)           RELREQ(YES)            AUTOCONNECT(NO)
LOGONMSG(YES)          CREATESESS(NO)
ATI(YES)               TTI(YES)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4A32782 と一致します。

DFHLU2E3

拡張データ・ストリームを使用する SNA LU タイプ 2 モデル 3 (照会):

```
TYPETERM(DFHLU2E3)     GROUP(DFHTYPE)
DEVICE(LUTYPE2)        TERMMODEL(2)           SHIPPABLE(YES)
DEFSCREEN(24,80)       PAGESIZE(24,80)        AUTOPAGE(NO)
ALTSCREEN(32,80)       ALTPAGE(32,80)
BRACKET(YES)           BUILDCHAIN(YES)        ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)      EXTENDEDDS(YES)        UCTRAN(YES)
RECEIVESIZE(1024)      SENDSIZE(3840)         IOAREALEN(256,4000)
ERRLASTLINE(YES)       ERRINTENSIFY(YES)      QUERY(ALL)
DISCREQ(YES)           RELREQ(YES)            AUTOCONNECT(NO)
LOGONMSG(YES)          CREATESESS(NO)
ATI(YES)               TTI(YES)
```

拡張データ・ストリームを使用する SNA LU タイプ 2 モデル 3 (照会)。この定義は、z/OS Communications Server が提供する LOGMODE SNX32703 と一致します。

DFHLU2M3

SNA LU タイプ 2 モデル 3:

```
TYPETERM(DFHLU2M3)     GROUP(DFHTYPE)
DEVICE(LUTYPE2)        TERMMODEL(2)           SHIPPABLE(YES)
DEFSCREEN(24,80)       PAGESIZE(24,80)        AUTOPAGE(NO)
ALTSCREEN(32,80)       ALTPAGE(32,80)
BRACKET(YES)           BUILDCHAIN(YES)        ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)      EXTENDEDDS(NO)         UCTRAN(YES)
RECEIVESIZE(1024)      SENDSIZE(1536)         IOAREALEN(256,4000)
ERRLASTLINE(YES)       ERRINTENSIFY(YES)      QUERY(NO)
DISCREQ(YES)           RELREQ(YES)            AUTOCONNECT(NO)
LOGONMSG(YES)          CREATESESS(NO)
ATI(YES)               TTI(YES)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4A32783 と一致します。

DFHLU2E4

拡張データ・ストリームを使用する SNA LU タイプ 2 モデル 4 (照会):

```
TYPETERM(DFHLU2E4)     GROUP(DFHTYPE)
DEVICE(LUTYPE2)        TERMMODEL(2)          SHIPPABLE(YES)
DEFSCREEN(24,80)       PAGESIZE(24,80)       AUTOPAGE(NO)
ALTSCREEN(43,80)       ALTPAGE(43,80)
BRACKET(YES)           BUILDCHAIN(YES)       ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)      EXTENDEDDS(YES)       UCTRAN(YES)
RECEIVESIZE(1024)      SENDSIZE(3840)        IOAREALEN(256,4000)
ERRLASTLINE(YES)       ERRINTENSIFY(YES)     QUERY(ALL)
DISCREQ(YES)           RELREQ(YES)           AUTOCONNECT(NO)
LOGONMSG(YES)          CREATESESS(NO)
ATI(YES)               TTI(YES)
```

拡張データ・ストリームを使用する SNA LU タイプ 2 モデル 4 (照会)。この定義は、z/OS Communications Server が提供する LOGMODE SNX32704 と一致します。

DFHLU2M4

SNA LU タイプ 2 モデル 4:

```
TYPETERM(DFHLU2M4)     GROUP(DFHTYPE)
DEVICE(LUTYPE2)        TERMMODEL(2)          SHIPPABLE(YES)
DEFSCREEN(24,80)       PAGESIZE(24,80)       AUTOPAGE(NO)
ALTSCREEN(43,80)       ALTPAGE(43,80)
BRACKET(YES)           BUILDCHAIN(YES)       ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)      EXTENDEDDS(NO)        UCTRAN(YES)
RECEIVESIZE(1024)      SENDSIZE(1536)        IOAREALEN(256,4000)
ERRLASTLINE(YES)       ERRINTENSIFY(YES)     QUERY(NO)
DISCREQ(YES)           RELREQ(YES)           AUTOCONNECT(NO)
LOGONMSG(YES)          CREATESESS(NO)
ATI(YES)               TTI(YES)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4A32784 と一致します。

DFHLU2M5

SNA LU タイプ 2 モデル 5:

```
TYPETERM(DFHLU2M5)     GROUP(DFHTYPE)
DEVICE(LUTYPE2)        TERMMODEL(2)          SHIPPABLE(YES)
DEFSCREEN(24,80)       PAGESIZE(24,80)       AUTOPAGE(NO)
ALTSCREEN(27,132)      ALTPAGE(27,132)
BRACKET(YES)           BUILDCHAIN(YES)       ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)      EXTENDEDDS(NO)        UCTRAN(YES)
RECEIVESIZE(1024)      SENDSIZE(1536)        IOAREALEN(256,4000)
ERRLASTLINE(YES)       ERRINTENSIFY(YES)     QUERY(NO)
DISCREQ(YES)           RELREQ(YES)           AUTOCONNECT(NO)
LOGONMSG(YES)          CREATESESS(NO)
ATI(YES)               TTI(YES)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4A32785 と一致します。

DFHLU2E5

拡張データ・ストリームを使用する SNA LU タイプ 2 モデル 5 (照会):

```
TYPETERM(DFHLU2E5)     GROUP(DFHTYPE)
DEVICE(LUTYPE2)        TERMMODEL(2)          SHIPPABLE(YES)
DEFSCREEN(24,80)       PAGESIZE(24,80)       AUTOPAGE(NO)
ALTSCREEN(27,132)      ALTPAGE(27,132)
BRACKET(YES)           BUILDCHAIN(YES)       ROUTEDMSGS(ALL)
AUDIBLEALARM(YES)      EXTENDEDDS(NO)        UCTRAN(YES)
RECEIVESIZE(1024)      SENDSIZE(3480)        IOAREALEN(256,4000)
```

```
ERRLASTLINE(YES)      ERRINTENSIFY(YES)     QUERY(ALL)
DISCREQ(YES)          RELREQ(YES)           AUTOCONNECT(NO)
LOGONMSG(YES)         CREATESESS(NO)
ATI(YES)              TTI(YES)
```

拡張データ・ストリームを使用する SNA LU タイプ 2 モデル 5 (照会)。この定義は、z/OS Communications Server が提供する LOGMODE LSX32705 と一致します。

グループ DFHTERM 内のモデル TERMINAL 定義

CICS が提供する CSD グループ DFHTERM には、自動インストール用のモデル TERMINAL 定義が含まれています。

LU2 TYPETERM DFHLU2 を使用する SNA 3270 モデル 2 ディスプレイ:

```
TERMINAL(LU2)       GROUP(DFHTERM)      TYPETERM(DFHLU2)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU2)
```

この定義は3278モデル2ディスプレイ用です。3178, 3179, 3277, 3278, 3279, 3290, 3270PC, 3270PC/G, 3270PC/GX, 8775,および5550の装置に適しています。

LU3 TYPETERM DFHLU3 を使用する SNA 3270 モデル 2 プリンター:

```
TERMINAL(LU3)       GROUP(DFHTERM)      TYPETERM(DFHLU3)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU3)
```

この定義は3287プリンター用です。 3262, 3268, 3284, 3286, 3287, 3288, 3289, および5550の装置に適しています。

SCSP TYPETERM DFHSCSP を使用する SNA 3278 モデル 2 プリンター:

```
TERMINAL(SCSP)      GROUP(DFHTERM)      TYPETERM(DFHSCSP)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHSCSP)
```

この定義は3287プリンター用です。 3262, 3268, 3287, 3289,および5550の装置に適しています。

3270 TYPETERM DFH3270 を使用する非 SNA 3270 モデル 2 ディスプレイ:

```
TERMINAL(3270)      GROUP(DFHTERM)      TYPETERM(DFH3270)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFH3270)
```

この定義は3278モデル2ディスプレイ用です。 3178, 3179, 3277, 3278, 3279,および3290の 装置に適しています。

3284 TYPETERM DFH3270P を使用する非 SNA 3270 モデル 2 プリンター:

```
TERMINAL(3284)      GROUP(DFHTERM)      TYPETERM(DFH3270P)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFH3270P)
```

この定義は3284モデル2プリンター用です。 3262, 3268, 3284, 3287, 3288, 3289, および5550の装置に適しています。

LU62 TYPETERM DFHLU62T を使用する APPC (LU6.2) 単一セッション端末:

```
TERMINAL(LU62)      GROUP(DFHTERM)       TYPETERM(DFHLU62T)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU62T)
```

この定義はAPPC単独セッション端末用であり、 DISPLAYWRITER, SCANMASTER,およびSYSTEM/38の 装置に適しています。

L0E2 TYPETERM DFHLU0E2 を使用する非 SNA モデル 2 ディスプレイ:

```
TERMINAL(L0E2)      GROUP(DFHTERM)       TYPETERM(DFHLU0E2)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU0E2)
```

拡張データ・ストリームを使用する非SNAモデル2（照会）。この定義は、z/OS Communications Server が提供する LOGMODE NSX32702 と一致します。

L0M2 TYPETERM DFHLU0M2 を使用する非 SNA モデル 2 ディスプレイ:

```
TERMINAL(L0M2)      GROUP(DFHTERM)       TYPETERM(DFHLU0M2)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU0M2)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4B32782 と一致します。

L0M3 TYPETERM DFHLU0M3 を使用する非 SNA モデル 3 ディスプレイ:

```
TERMINAL(L0M3)      GROUP(DFHTERM)       TYPETERM(DFHLU0M3)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU0M3)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4B32783 と一致します。

L0M4 TYPETERM DFHLU0M4 を使用する非 SNA モデル 4 ディスプレイ:

```
TERMINAL(L0M4)      GROUP(DFHTERM)       TYPETERM(DFHLU0M4)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU0M4)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4B32784 と一致します。

L0M5 TYPETERM DFHLU0M5 を使用する非 SNA モデル 5 ディスプレイ:

```
TERMINAL(L0M5)      GROUP(DFHTERM)       TYPETERM(DFHLU0M5)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU0M5)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4B32785 と一致します。

L2E2 TYPETERM DFHLU2E2 を使用する SNA LU2 モデル 2 ディスプレイ:

```
TERMINAL(L2E2)      GROUP(DFHTERM)       TYPETERM(DFHLU2E2)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU2E2)
```

拡張データ・ストリームを使用する SNA LU タイプ 2 モデル 2 (照会)。この定義は、z/OS Communications Server が提供する LOGMODE SNX32702 と一致します。

L2M2 TYPETERM DFHLU2M2 を使用する SNA LU2 モデル 2 ディスプレイ:

```
TERMINAL(L2M2)      GROUP(DFHTERM)       TYPETERM(DFHLU2M2)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU2M2)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4A32782 と一致します。

L2E3 TYPETERM DFHLU2E3 を使用する SNA LU2 モデル 3 ディスプレイ:

```
TERMINAL(L2E3)      GROUP(DFHTERM)       TYPETERM(DFHLU2E3)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU2E3)
```

拡張データ・ストリームを使用する SNA LU タイプ 2 モデル 3 (照会)。この定義は、z/OS Communications Server が提供する LOGMODE SNX32703 と一致します。

L2M3 TYPETERM DFHLU2M3 を使用する SNA LU2 モデル 3 ディスプレイ:

```
TERMINAL(L2M3)      GROUP(DFHTERM)        TYPETERM(DFHLU2M3)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU2M3)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4A32783 と一致します。

L2E4 TYPETERM DFHLU2E4 を使用する SNA LU2 モデル 4 ディスプレイ:

```
TERMINAL(L2E4)      GROUP(DFHTERM)        TYPETERM(DFHLU2E4)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU2E4)
```

拡張データ・ストリームを使用する SNA LU タイプ 2 モデル 4 (照会)。この定義は、z/OS Communications Server が提供する LOGMODE SNX32704 と一致します。

L2M4 TYPETERM DFHLU2M4 を使用する SNA LU2 モデル 4 ディスプレイ:

```
TERMINAL(L2M4)      GROUP(DFHTERM)        TYPETERM(DFHLU2M4)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU2M4)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4A32784 と一致します。

L2E5 TYPETERM DFHLU2E5 を使用する SNA LU2 モデル 5 ディスプレイ:

```
TERMINAL(L2E5)      GROUP(DFHTERM)        TYPETERM(DFHLU2E5)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU2E5)
```

拡張データ・ストリームを使用する SNA LU タイプ 2 モデル 5 (照会)。この定義は、z/OS Communications Server が提供する LOGMODE LSX32705 と一致します。

L2M5 TYPETERM DFHLU2M5 を使用する SNA LU2 モデル 5 ディスプレイ:

```
TERMINAL(L2M5)      GROUP(DFHTERM)        TYPETERM(DFHLU2M5)
AUTINSTMODEL(ONLY)  AUTINSTNAME(DFHLU2M5)
```

この定義は、z/OS Communications Server が提供する LOGMODE D4A32785 と一致します。

CBRF 3270 ブリッジで使用するデフォルトのテンプレート端末。

```
TERMINAL(CBRF)      GROUP(DFHTERM)        TYPETERM(DFHLU2)
NETNAME(CBRF)       REMOTESYSTEM(CBR)     REMOTENAME(CBRF)
```

グループ DFHEP 内の PROFILE 定義

CICS が提供する CSD グループ DFHEP には、イベント処理用の PROFILE 定義が含まれています。

DFHECEPH

CICS は、CEPH トランザクション (HTTP EP アダプター) にこのプロファイルを使用して、RTIMOUT パラメーターによって設定されたタイムアウトを管理します。

定義は以下のとおりです。

```
PROFILE(DFHECEPH)
GROUP(DFHEP)
SCRNSIZE(DEFAULT)
MSGJRNL(NO)
MSGINTEG(NO)
ONEWTE(NO)
DVSUPRT(ALL)
INBFMH(NO)
RAQ(NO)
LOGREC(NO)
NEPCLASS(000)
RTIMOUT(0005)
```

グループ DFHISC 内の PROFILE 定義

提供される CSD グループ DFHISC には、システム間通信セッション用の PROFILE 定義が含まれます。

DFHCICSF

CICSアプリケーション・プログラムが機能シップ要求を発行するとき、CICSはリモート・システムまたはリモート領域へのセッションで このプロファイルを使用します。

定義は以下のとおりです。

```
PROFILE(DFHCICSF)
GROUP(DFHISC)
SCRNSIZE(DEFAULT)
MSGJRNL(NO)
MSGINTEG(NO)
ONEWTE(NO)
DVSUPRT(ALL)
INBFMH(ALL)
RAQ(NO)
LOGREC(NO)
NEPCLASS(000)
```

DFHCICSR

CICSは、（アプリケーション専有領域で実行されている） ユーザー・トランザクションと、領域間リンクまたは APPCリンクの間の通信のトランザクション・ルーティングで このプロファイルを使用します。

定義は以下のとおりです。

```
PROFILE(DFHCICSR)
GROUP(DFHISC)
SCRNSIZE(DEFAULT)
MSGJRNL(NO)
MSGINTEG(NO)
ONEWTE(NO)
DVSUPRT(ALL)
INBFMH(ALL)
RAQ(NO)
LOGREC(NO)
NEPCLASS(000)
```

DFHCICSS

CICSは、（端末専有領域で実行されている）中継 トランザクションと、領域間リンクまたはAPPC リンクの間の通信のトランザクション・ルーティングで このプロファイルを使用します。TRANSACTION定義の TRPROFオプションを使用して、異なるプロファイルを 指定できます。

定義は以下のとおりです。

```
PROFILE(DFHCICSS)
GROUP(DFHISC)
SCRNSIZE(DEFAULT)
MSGJRNL(NO)
MSGINTEG(NO)
ONEWTE(NO)
DVSUPRT(ALL)
INBFMH(ALL)
RAQ(NO)
LOGREC(NO)
NEPCLASS(000)
```

グループ DFHSTAND 内の PROFILE 定義

CICS が提供する CSD グループ DFHSTAND には、PROFILE 定義が含まれています。

CICS が提供する CSD グループ DFHSTAND には、以下の PROFILE 定義が含まれています。

DFHCICSA

これは、アプリケーション・プログラムの ALLOCATE コマンドによって獲得される代替機能のデフォルト・プロファイルです。 ALLOCATE コマンドで別のプロファイルを明示的に指定することもできます。

定義は以下のとおりです。

```
PROFILE(DFHCICSA)
GROUP(DFHSTAND)
SCRNSIZE(DEFAULT)
MSGJRNL(NO)
MSGINTEG(NO)
ONEWTE(NO)
PROTECT(NO)
DVSUPRT(ALL)
INBFMH(ALL)
RAQ(NO)
LOGREC(NO)
NEPCLASS(000)
```

DFHCICSE

これは、基本機能のエラー・プロファイルです。 CICS は、必要なプロファイルが見つからない場合、このプロファイルを使用して、基本機能にエラー・メッセージを渡します。

定義は以下のとおりです。

```
PROFILE(DFHCICSE)
GROUP(DFHSTAND)
SCRNSIZE(DEFAULT)
MSGJRNL(NO)
MSGINTEG(NO)
ONEWTE(NO)
PROTECT(NO)
DVSUPRT(ALL)
INBFMH(NO)
RAQ(NO)
LOGREC(NO)
NEPCLASS(000)
```

DFHCICSP

これは、ページ検索トランザクション CSPG のデフォルト・プロファイル

です。 TRANSACTION 定義の PROFILE オプションによって、特定のトランザクション用に別のプロファイルを指定することもできます。

定義は以下のとおりです。

```
PROFILE(DFHCICSP)
GROUP(DFHSTAND)
SCRNSIZE(DEFAULT)
UCTRAN(YES)
MSGJRNL(NO)
MSGINTEG(NO)
ONEWTE(NO)
PROTECT(NO)
DVSUPRT(ALL)
INBFMH(NO)
RAQ(NO)
LOGREC(NO)
NEPCLASS(000)
```

DFHCICST

これは、基本機能のデフォルト・プロファイルです。 TRANSACTION 定義の PROFILE オプションによって、特定のトランザクション用に別のプロファイルを指定することもできます。

定義は以下のとおりです。

```
PROFILE(DFHCICST)
GROUP(DFHSTAND)
SCRNSIZE(DEFAULT)
MSGJRNL(NO)
MSGINTEG(NO)
ONEWTE(NO)
PROTECT(NO)
DVSUPRT(ALL)
INBFMH(NO)
RAQ(NO)
LOGREC(NO)
NEPCLASS(000)
```

DFHCICSV

これは、トランザクションが z/OS Communications Server 装置のみをサポートする場合の基本機能のプロファイルです。

定義は以下のとおりです。

```
PROFILE(DFHCICSV)
GROUP(DFHSTAND)
SCRNSIZE(DEFAULT)
MSGJRNL(NO)
MSGINTEG(NO)
ONEWTE(NO)
PROTECT(NO)
DVSUPRT(VTAM)
INBFMH(NO)
RAQ(NO)
LOGREC(NO)
NEPCLASS(000)
```

注**:** VTAM は現在 z/OS Communications Server になっています。

DFHPPF01

プロファイル DFHPPF01 は、CICS の初期設定時に、CSD ファイルがインストールされる前に接続されたタスクに対して使用されます。

この定義は以下のとおりです。

```
PROFILE(DFHPPF01)
GROUP(DFHSTAND)
DESCRIPTION(VTAM-ONLY PROFILE)
SCRNSIZE(DEFAULT)
MSGJRNL(NO)
MSGINTEG(NO)
ONEWTE(NO)
PROTECT(NO)
DVSUPRT(VTAM)
INBFMH(NO)
RAQ(NO)
LOGREC(NO)
NEPCLASS(000)
```

DFHPPF02

プロファイル DFHPPF02 は、CICS の初期設定時に、CSD ファイルがインストールされる前に接続されたタスクに対して使用されます。

この定義は以下のとおりです。

```
PROFILE(DFHPPF02)
GROUP(DFHSTAND)
DESCRIPTION(ALL-NULLS PROFILE)
SCRNSIZE(DEFAULT)
MSGJRNL(NO)
MSGINTEG(NO)
ONEWTE(NO)
PROTECT(NO)
DVSUPRT(ALL)
INBFMH(NO)
RAQ(NO)
LOGREC(NO)
NEPCLASS(000)
```

グループ DFHPGAIP 内のモデル定義

CICS は、プログラムの自動インストールをサポートするためにグループ DFHPGAIP 内で複数のモデル定義を提供しています。

DFHPGAPG

これは、プログラムの自動インストール用のデフォルトの PROGRAM 定義です。

定義は以下のとおりです。

```
PROGRAM(DFHPGAPG)  GROUP(DFHPGAIP)
DESCRIPTION(default program for program autoinstall)
                   RELOAD(NO)            RESIDENT(NO)
USAGE(NORMAL)      USELPACOPY(NO)        STATUS(ENABLED)
RSL(00)            CEDF(YES)             DATALOCATION(ANY)
EXECKEY(USER)      EXECUTIONSET(FULLAPI)
```

DATALOCATION(ANY) の指定がデフォルトに適していない場合は、以下のいずれかの方法を使用して、DATALOCATION のデフォルトを ANY 以外にすることができます。

- DFHPGAPG の定義を独自のグループにコピーして、DATALOCATION 設定を変更します。定義は、グループ DFHPGAIP の後にインストールする必要があります。
- プログラムの自動インストール出口からの出力パラメーターとして使用されるモデル定義として、独自のプログラム定義の名前を指定します。

DFHPGAMP

これは、プログラムの自動インストール用のデフォルトの MAPSET 定義です。

定義は以下のとおりです。

```
MAPSET(DFHPGAMP)  GROUP(DFHPGAIP)
DESCRIPTION(default mapset for program autoinstall)
RESIDENT(NO)       USAGE(NORMAL)
USELPACOPY(NO)     STATUS(ENABLED)
RSL(00)
```

DFHPGAPT

これは、プログラムの自動インストール用のデフォルトの PARTITION 定義です。

定義は以下のとおりです。

```
PARTITIONSET(DFHPGAPT)  GROUP(DFHPGAIP)
DESCRIPTION(default partitionset for program autoinstall)
RESIDENT(NO)       USAGE(NORMAL)
USELPACOPY(NO)     STATUS(ENABLED)
RSL(00)
```

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。この資料の他の言語版を IBM から入手できる場合があります。ただし、これを入手するには、本製品または当該言語版製品を所有している必要がある場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。 IBM 製品、プログラムまたはサービスに代えて、IBM の知的所有権を侵害することのない機能的に同等のプログラムまたは製品を使用することができます。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で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。 IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様自身の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

© Copyright IBM Corp. 1982, 2018

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119 Armonk,
NY 10504-1785
United States of America

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。 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にお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、類似する個人や企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、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で提供されるものであり、いかなる保証も提供されません。 IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

プログラミング・インターフェース情報

CICS には、プログラミング・インターフェースと見なすことのできる資料と、プログラミング・インターフェースと見なすことのできない資料があります。

オンライン製品資料の以下のセクションには、CICS Transaction Server for z/OS, バージョン 5 リリース 5 のサービスを取得するプログラムをお客様が作成するためのプログラミング・インターフェースが含まれています。

- アプリケーションの開発
- システム・プログラムの開発
- 保護の概要
- 外部インターフェースに向けた開発
- リファレンス: アプリケーション開発
- リファレンス: システム・プログラミング
- リファレンス: 接続

オンライン製品資料の以下のセクションには、CICS Transaction Server for z/OS, バージョン 5 リリース 5 のプログラミング・インターフェースとして意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が含まれています。

- トラブルシューティングおよびサポート
- リファレンス: 診断

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 5 のサービスを取得するプログラムをお客様が作成するためのプログラミング・インターフェースが以下のマニュアルに含まれています。

- アプリケーション・プログラミング・ガイドおよびアプリケーション・プログラミング・リファレンス
- Business Transaction Services
- Customization Guide
- C++ OO Class Libraries
- Debugging Tools Interfaces Reference
- Distributed Transaction Programming Guide
- External Interfaces Guide
- Front End Programming Interface Guide
- IMS Database Control Guide
- インストール・ガイド
- セキュリティー・ガイド
- Supplied Transactions
- CICSPlex SM Managing Workloads
- CICSPlex SM Managing Resource Usage
- CICSPlex SM アプリケーション・プログラミング・ガイドおよび CICSPlex SM アプリケーション・プログラミング・リファレンス
- Java Applications in CICS

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 5 のプログラミング・インターフェース

として意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が以下のマニュアルに含まれています。

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

商標

IBM、IBM ロゴおよび ibm.com® は、世界の多くの国で登録された International Business Machines Corporation の商標または登録商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、 http://www.ibm.com/legal/copytrade.shtml をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

製品資料に関するご使用条件

これらの資料は、以下のご使用条件に同意していただける場合に限りご使用いただけます。

適用範囲
: IBM Web サイトの「ご利用条件」に加えて、以下のご使用条件が適用されます。

個人使用
: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

商用使用
: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することがで

きます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

権利　ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態で提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

IBM オンラインでのプライバシー・ステートメント

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品 (「ソフトウェア・オファリング」) では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項をご確認ください。

CICSPlex SM Web ユーザー・インターフェース **(**メイン・インターフェース**)** の場合**:**　このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、認証、お客様の利便性の向上、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名、およびその他の個人情報を、セッションごとの Cookie および持続的な Cookie を使用して収集する場合があります。これらの Cookie を無効にすることはできません。

CICSPlex SM Web ユーザー・インターフェース **(**データ・インターフェース**)** の場合**:**　このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、認証、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名またはその他の個人情報を、セッションごとの Cookie を使用して収集する場合があります。これらの Cookie を無効にすることはできません。

CICSPlex SM Web ユーザー・インターフェース **(**「**Hello World**」ページ**)** の場合**:**　このソフトウェア・オファリングは、展開される構成に応じて、個人情報を収集しないセッションごとの Cookie を使用する場合があります。これらの Cookie を無効にすることはできません。

CICS Explorer の場合**:**

このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、お客様の利便性の向上、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名、およびその他の個人情報を、セッションごとの設定および持続的な設定を使用して収集する場合があります。これらの設定を無効にすることはできませんが、ユーザー・パスワードの暗号化形式でのディスクへの保管は、サインオン中にチェック・ボックスにチェック・マークを付けることによるユーザーの明示的な操作によってのみ有効化することができます。

この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、 お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。

このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、『IBM オンラインでのプライバシー・ステートメント』(http://www.ibm.com/privacy/details/jp/ja/) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』および「IBM Software Products and Software-as-a-Service Privacy Statement」(http://www.ibm.com/software/info/product-privacy) を参照してください。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

異常終了コード
  DFHSRT TYPE=SYSTEM 447
  DFHSRT TYPE=USER 447
一時記憶
  セキュリティー検査、DFHTST TYPE=SECURITY 482
  データ共用、DFHTST TYPE=SHARED 483
  リカバリー可能、DFHTST TYPE=RECOVERY 478
一時記憶域セキュリティー検査
  DFHTST TYPE=SECURITY 482
一時記憶域データ共用
  DFHTST TYPE=SHARED 483
一時記憶域テーブル (TST) 475, 484
  一時記憶域セキュリティー検査、DFHTST TYPE=SECURITY 482
  一時記憶域データ共用、DFHTST TYPE=SHARED 483
  制御セクション 477
  データ共用一時記憶域 483
  リカバリー可能一時記憶域 478, 480, 482
  リカバリー可能一時記憶域、DFHTST TYPE=RECOVERY 478
  リモート一時記憶域 DATAID、DFHTST TYPE=REMOTE 481
  ローカル一時記憶域 DATAID、DFHTST TYPE=LOCAL 480
  DFHTST TYPE=INITIAL 477
  DFHTST TYPE=RECOVERY 478, 480, 482
  DFHTST TYPE=SHARED 483
一時データ
  二重目的リソース定義 189
一時データ・キュー
  ダミー 209
  無効にする 209
一時データ・キュー定義
  データ・セット定義 189
イベント、モニター・ポイント
  DFHMCT TYPE=EMP 420
印刷
  端末 211
印刷、連続 (自動ページング) 460

[カ行]

拡張回復機能 (XRF)
  コマンド・リスト・テーブル (CLT) (command list table (CLT)) 399
拡張ローカル LDC リストおよびシステム LCD テーブル
  DFHTCT TYPE=LDC 457
カップリング・ファシリティー・データ・テーブル 54
間接接続
  定義 10
記憶域キュー、DFHTST TYPE=LOCAL
  ローカル一時 480
記憶域キュー、DFHTST TYPE=REMOTE
  リモート一時 481
基本マッピング・サポート (BMS)
  プリンターへの複数のページ・メッセージ 460
共用 CSD
  リリース間 500
許可プログラム機能 (APF) 399, 444
コード化例
  FCT 414
  MCT 438
  PLT 443
  RST 446
  SRT 448
  TLT 475
  XLT 488
コマンド・スレッド属性
  DB2CONN 定義 34
コマンド・リスト・テーブル (CLT) (command list table (CLT)) 399
  CLT (コマンド・リスト・テーブル) 399
  DFHCLT TYPE=COMMAND 401
  DFHCLT TYPE=INITIAL 400
  DFHCLT TYPE=LISTEND 403
  DFHCLT TYPE=LISTSTART 401
  DFHCLT TYPE=WTO 402
  DFHCLT マクロ 399
コンソール
  事前設定セキュリティー 233
コンフィギュレーター
  DFHFCT TYPE=FILE 413

[サ行]

システム LDC テーブルおよび拡張ローカル LDC リスト
  DFHTCT TYPE=LDC 457
システム・リカバリー・テーブル (SRT)
  コーディング例 448
  DFHSRT TYPE=INITIAL 447
事前設定セキュリティー 467, 489
事前設定端末セキュリティー
  コンソール 233
従属デフォルト値 265
順次端末装置
  DFHTCT TYPE=LINE 462
  DFHTCT TYPE=SDSCI 462
  DFHTCT TYPE=TERMINAL 462
セキュリティー
  事前設定 467, 489
セキュリティー検査
  一時記憶 482
  DFHTST TYPE=SECURITY 482
セグメント検索引数、PSB 404
セッションおよび LUTYPE6.1 CICS-CICS ISC リンク 159
セッションおよび LUTYPE6.1 CICS-IMS リンク 159
セッションおよび MRO リンク 159
接続、INDIRECT
  リソース定義 10
属性
  CONNECTION 定義 11
  DB2CONN 定義 23
  DB2ENTRY 定義 36
  DB2TRAN 定義 42
  DOCTEMPLATE 定義 44
  ENQMODEL 定義 50
  FILE 定義 55
  JOURNALMODEL 定義 89
  LIBRARY 定義 100
  LSRPOOL 定義 106
  MAPSET 定義 112
  MQCONN 定義 115
  PARTITIONSET 定義 125
  PARTNER 定義 127
  PROCESSTYPE 定義 135
  PROFILE 定義 138
  SESSIONS 定義 160
  TCPIPSERVICE 定義 173

© Copyright IBM Corp. 1982, 2018

属性 (続き)
TDQUEUE 定義 189
TERMINAL 定義 221
TRANCLASS 定義 234
TRANSACTION 定義 238
TSMODEL 定義 258
TYPETERM 定義 277

[タ行]

ダミー TDQ 209
単一セッション端末
APPC
リソース定義、CONNECTION および SESSIONS 方式 8
TERMINAL-TYPETERM 方式の利点 8
端末
印刷用 211
RDO によるサポート 272
端末、CICS、リスト 471
端末管理テーブル (TCT) 449, 472
概要 449
順次装置 462
CICS 端末のリスト 471
DFHTCT TYPE=GROUP 453
DFHTCT マクロ、要約 450
RDO へのエントリーのマイグレーション 453
端末定義
summary 265
端末定義のアプリケーション専有領域へのシップ 218
端末リスト・テーブル (TLT) (terminal list table (TLT)) 475
コード化例 475
制御セクション 473
DFHTLT TYPE=ENTRY 474
DFHTLT TYPE=INITIAL 473
データ共用
一時記憶 483
DFHTST TYPE=SHARED 483
データベース制御 (DBCTL) 444
データ・セット
ファイルとの関係 404
ディスプレイ装置
プリンターとの関連付け 213
デフォルト値
TYPETERM 属性 265
トランザクション ID としてのワイルドカード文字 43
トランザクション・リスト・テーブル (XLT) 485
コーディング例 488
制御セクション 486
DFHXLT TYPE=INITIAL 486
トランザクション・ルーティング 468
端末 216
リモート端末の DFHTCT マクロ 468
リモート端末のマクロ 468
APPC 装置 220
INDIRECT 接続の使用 10
summary 221
トランザクション・ルーティング用の APPC 装置 220
トランザクション・ルーティング用の端末 216
トランザクション・ルーティング用のリモート端末
DFHTCT TYPE=REGION オペランド 469
DFHTCT TYPE=REMOTE オペランド 471
DFHTCT TYPE=TERMINAL オペランド 470
DFHTCT マクロ 468

[ナ行]

二重目的定義 217
二重目的リソース定義
一時データ 189

[ハ行]

パイプライン TERMINAL 定義 213
パイプライン端末 276
ファイル
データ・セットとの関係 404
ファイル管理テーブル (FCT) 404
コーディング例 414
サマリー表 413
DFHFCT TYPE=FILE 405
DFHFCT TYPE=INITIAL 405
DFHFCT マクロ 404
複数領域操作 (MRO)
リンクおよびセッション用のリソース定義 7
プリンター
端末定義 211
ディスプレイ装置との関連付け 213
PGESTAT オペランド 460
プログラム仕様ブロック (PSB) 403, 404
プログラム・リスト・テーブル (PLT) 439
コーディング例 443
制御セクション 441
DFHPLT TYPE=ENTRY 441
DFHPLT TYPE=INITIAL 441
プロファイル、CICS 提供のデフォルト
DFHCICSA 515
DFHCICSE 515
プロファイル、CICS 提供のデフォルト (続き)
DFHCICSF 514
DFHCICSP 515
DFHCICSR 514
DFHCICSS 514
DFHCICST 516
DFHCICSV 516
DFHECEPH 513
DFHPGAMP 518
DFHPGAPG 517
DFHPGAPT 518
DFHPPF01 516
DFHPPF02 517
並列 APPC セッション 8
並列セッションおよび APPC (LUTYPE6.2) リンク 159

[マ行]

モニター管理テーブル (MCT) 414
コード化例 438
制御データの記録 425
ユーザー・イベント・モニター・ポイント 420
DFHMCT TYPE=EMP 420
DFHMCT TYPE=INITIAL 415
DFHMCT TYPE=RECORD 425
DFHMCT マクロ 415
モニター・ポイント、イベント
DFHMCT TYPE=EMP 420

[ラ行]

リカバリー可能一時記憶域
DFHTST TYPE=RECOVERY 478
リカバリー可能サービス・テーブル (RST) 444, 446
コーディング例 446
DFHRST TYPE=INITIAL 445
DFHRST TYPE=RSE 445
DFHRST TYPE=SUBSYS 445
DFHRST マクロ・タイプ 444
リソース定義
APPC 単一セッション端末用の
CONNECTION および SESSIONS 方式 8
TERMINAL-TYPETERM 方式の利点 8
APPC リンクおよび並列セッション用の 8
INDIRECT 接続用の 10
LUTYPE6.1 CICS-CICS ISC リンクおよびセッション用の 7

リソース定義 (続き)
LUTYPE6.1 CICS-IMS リンクおよびセッション用の 9
LUTYPE6.1 リンクおよびセッション用の 8
MRO リンクおよびセッション用の 7
リモート一時記憶域 DATAID
DFHTST TYPE=REMOTE 481
リモート定義 217
リモート・システムの APPLID 17
リモート・システムの USERVAR 17
リモート・ファイル 53
例
FCT 414
MCT 438
PLT 443
RST 446
SRT 448
TLT 475
XLT 488
連続印刷 (自動ページング) 460
ローカル LDC リスト
DFHTCT TYPE=LDCLIST 461
ローカル一時記憶域 DATAID
DFHTST TYPE=LOCAL 480
ローカル定義 216
論理装置コード (LDC) 454, 461

[数字]

3290 端末 275
3600 オプション
LDC オペランド 459
3770 データ通信システム
バッチ LU 用の LDC TCT の例 462
3770 バッチ論理装置用の LDC TCT の例 462

A

ABCODE オペランド
DFHSRT TYPE=SYSTEM 447
DFHSRT TYPE=USER 447
ACCESSMETHOD 属性
CONNECTION 定義 12
ACCMETH オペランド
順次装置 464
リモート端末 470, 471
DFHFCT TYPE=FILE 406
DFHTCT TYPE=INITIAL 451
DFHTCT TYPE=LINE 464
DFHTCT TYPE=REMOTE 471
DFHTCT TYPE=TERMINAL 470
ACCOUNTREC 属性
DB2CONN 定義 30
DB2ENTRY 定義 38
ACTION 属性
TRANSACTION 定義 239
ADD オプション
SERVREQ オペランド 412
ADD コマンド
CEDA 335
ADD コマンド、DFHCSDUP ユーティリティー・プログラム 371
ADD 属性
FILE 定義 55
ADDCNT オプション
PERFORM オペランド 423
ALIAS 属性
TRANSACTION 定義 239
ALTER コマンド
CEDA 337
ALTER コマンド、DFHCSDUP ユーティリティー・プログラム 372
総称ネーミング 373
ALTER コマンドでの総称ネーミング 373
ALTPAGE 属性
TYPETERM 定義 280
ALTPRINTCOPY 属性
TERMINAL 定義 222
ALTPRINTER 属性
TERMINAL 定義 222
ALTSCREEN 属性
TYPETERM 定義 281
ALTSUFFIX 属性
TYPETERM 定義 282
ANALYZER 属性
URIMAP 定義 314
APF (許可プログラム機能) 399, 444
APLKYBD 属性
TYPETERM 定義 282
APLTEXT 属性
TYPETERM 定義 282
APPC (LUTYPE6.2) 単一セッション端末 215
APPC (LUTYPE6.2)リンクと並行セッション 159
APPEND コマンド
CEDA 339
APPEND コマンド、DFHCSDUP ユーティリティー・プログラム 374
例 375
APPENDCRLF 属性
DOCTEMPLATE 定義 45
APPLID オペランド
DFHSIT マクロ 10
APPLID 属性
IPCONN 定義 78
ASCII 属性
TYPETERM 定義 283
ATI 属性
TYPETERM 定義 283
ATIFACILITY 属性
TDQUEUE 定義 191
ATOMSERVICE 属性
ATOMSERVICE 定義 2
URIMAP 定義 314
ATOMSERVICE 定義
ATOMSERVICE 属性 2
ATOMTYPE 属性 2
BINDFILE 属性 2
CONFIGFILE 属性 3
RESOURCENAME 属性 4
RESOURCETYPE 属性 4
STATUS 属性 4
ATOMSERVICE リソース定義 1
ATOMTYPE 属性
ATOMSERVICE 定義 2
ATTACHSEC 属性
CONNECTION 定義 13
TCPIPSERVICE 定義 173
TERMINAL 定義 223
ATTLSAWARE 185
AT-TLS 185
AUDIBLEALARM 属性
TYPETERM 定義 284
AUDITLEVEL 属性
PROCESSTYPE 定義 135
AUDITLOG 属性
PROCESSTYPE 定義 136
AUTHENTICATE 属性
TCPIPSERVICE 定義 173
AUTHID 属性
DB2CONN 定義 30
DB2ENTRY 定義 38
AUTHTYPE 属性
DB2CONN 定義 30
DB2ENTRY 定義 39
AUTINSTMODEL TERMINAL 定義
CICS 提供の 511
AUTINSTMODEL 属性
TERMINAL 定義 223
AUTINSTNAME 属性
TERMINAL 定義 224
AUTOCONNECT 属性
CONNECTION 定義 14
IPCONN 定義 78
SESSIONS 定義 161
TYPETERM 定義 284
AUTOPAGE 属性
TYPETERM 定義 284

B

BACKLOG 属性
  TCPIPSERVICE 定義 173
BACKTRANS 属性
  TYPETERM 定義 285
BACKUPTYPE 属性
  FILE 定義 55, 56
BCHLU オプション
  LDC オペランド 459
BINDFILE 属性
  ATOMSERVICE 定義 2
BINDPASSWORD 属性
  CONNECTION 489
BINDSECURITY 属性
  CONNECTION 属性 14
  TERMINAL 定義 224
BLKKEYL オペランド
  DFHFCT TYPE=FILE 406
BLKSIZE オペランド
  順次装置 463
  DFHFCT TYPE=FILE 406
  DFHTCT TYPE=SDSCI 463
BLOCKED オプション
  RECFORM オペランド 411
BLOCKFORMAT 属性
  TDQUEUE 定義 192
BLOCKSIZE 属性
  TDQUEUE 定義 192
BMS (基本マッピング・サポート)
  プリンターへの複数のページ・メッセージ 460
BRACKET 属性
  TYPETERM 定義 285
BREXIT 属性
  TRANSACTION 定義 240
BROWSE オプション
  SERVREQ オペランド 412
BROWSE 属性
  FILE 定義 55
BUILDCHAIN 属性
  SESSIONS 定義 162
  TYPETERM 定義 285
BUNDLE リソース 4, 51

C

CADL 一時データ宛先 204
CADO 一時データ宛先 204
CADS 一時データ宛先 204
CAIL 一時データ宛先 204
CCPI 一時データ宛先 204
CCSE 一時データ宛先 204
CCSI 一時データ宛先 205
CCSO 一時データ宛先 205
CCZM 一時データ宛先 205
CDB2 一時データ宛先 205
CDBC 一時データ宛先 205
CDEP 一時データ宛先 205
CDUL 一時データ宛先 205
CECO 一時データ宛先 205
CEDA
  ADD コマンド 335
  ALTER コマンド 337
  APPEND コマンド 339
  CHECK コマンド 340
  COPY コマンド 342
  DEFINE コマンド 345
  DELETE コマンド 347
  DISPLAY GROUP コマンド 350
  DISPLAY LIST コマンド 350
  DISPLAY コマンド 349
  EXPAND GROUP コマンド 352
  EXPAND LIST コマンド 353
  EXPAND コマンド 351
  INSTALL コマンド 354
  LOCK コマンド 356
  MOVE コマンド 358
  REMOVE コマンド 360
  RENAME コマンド 362
  UNLOCK コマンド 364
  USERDEFINE コマンド 365
  VIEW コマンド 369
CEJL 一時データ宛先 205
CEPO 一時データ宛先 205
CERTIFICATE 属性
  IPCONN 定義 78
  TCPIPSERVICE 定義 173
  URIMAP 定義 315
CESE 一時データ宛先 206
CFDTPOOL 属性
  FILE 定義 55
CGCSGID 属性
  TYPETERM 定義 285
CHAINCONTROL 属性
  PROFILE 定義 138
CHARACTERSET 属性
  URIMAP 定義 315
CHECK コマンド
  CEDA 340
CICS 提供の TYPETERM 定義
  ALTPAGE 属性 280
  CICS 提供の 505
CIEO 一時データ宛先 206
CIPHERS 属性
  IPCONN 定義 78
  TCPIPSERVICE 定義 173
  URIMAP 定義 315
CISL 一時データ宛先 206
CISO 一時データ宛先 206
CJRM 一時データ宛先 206
CKQQ 一時データ宛先 206
CLASS オペランド
  DFHMCT TYPE=EMP 420
  DFHMCT TYPE=RECORD 425
CLOCK オペランド
  DFHMCT TYPE=EMP 421
CLT (コマンド・リスト・テーブル) 399
  CLT (コマンド・リスト・テーブル) 399
  DFHCLT TYPE=COMMAND 401
  DFHCLT TYPE=INITIAL 400
  DFHCLT TYPE=LISTEND 403
  DFHCLT TYPE=LISTSTART 401
  DFHCLT TYPE=WTO 402
  DFHCLT マクロ 399
CMDSEC 属性
  TRANSACTION 定義 240
CMLO 一時データ宛先 206
CMPO 一時データ宛先 206
CMQM 一時データ宛先 206
COLOR 属性
  TYPETERM 定義 286
COMAUTHID 属性
  DB2CONN 定義 34
COMAUTHTYPE 属性
  DB2CONN 定義 34
COMMAND オペランド
  DFHCLT TYPE=COMMAND 401
COMPLETE オプション
  MIGRATE オペランド 453
COMTHREADLIMIT 属性
  DB2CONN 定義 36
CONFDATA 属性
  TRANSACTION 定義 240
CONFIGFILE 属性
  ATOMSERVICE 定義 3
CONNECTERROR 属性
  DB2CONN 定義 25
CONNECTION 属性
  CONNECTION 定義 15
  SESSIONS 定義 162
CONNECTION 定義
  ACCESSMETHOD 属性 10, 12
  ATTACHSEC 属性 13
  AUTOCONNECT 属性 14
  BINDPASSWORD 属性 489
  BINDSECURITY 属性 14
  CONNECTION 属性 15
  CONNTYPE 属性 15
  DATASTREAM 属性 15
  INDSYS 属性 10, 16
  INSERVICE 属性 16
  MAXQTIME 属性 16
  NETNAME 属性 10, 17
  PROTOCOL 属性 10, 19
  PSRECOVERY 属性 19
  QUEUELIMIT 属性 20

CONNECTION 定義 (続き)
RECORDFORMAT 属性 20
REMOTENAME 属性 21
REMOTESYSNET 属性 21
REMOTESYSTEM 属性 21
SECURITYNAME 属性 22
SINGLESESS 属性 22
USEDFLTUSER 属性 22
XLNACTION 属性 22
CONNECTION リソース 6
CONNTYPE 属性
CONNECTION 定義 15
CONSNAME 属性
TERMINAL 定義 224
CONSOLE 属性
TERMINAL 定義 489
CONVERTER 属性
URIMAP 定義 316
COPY コマンド
CEDA 342
COPY コマンド、 DFHCSDUP ユーティリティー・プログラム
MERGE オプション 377
COPY コマンド、DFHCSDUP ユーティリティー・プログラム 375
総称ネーミング 377
例 378
FROMCSD オプション 377
REPLACE オプション 377
COPY コマンドでの総称ネーミング 377
COPY 属性
TYPETERM 定義 286
COUNT オペランド
DFHMCT TYPE=EMP 422
CPIO 一時データ宛先 206
CRDI 一時データ宛先 206
CREATESESS 属性
TYPETERM 定義 286
CRITICAL 属性
LIBRARY 定義 101
CRLO 一時データ宛先 207
CRPO 一時データ宛先 207
CSBA 一時データ宛先 207
CSBR 一時データ宛先 207
CSCC 一時データ宛先 207
CSCS 一時データ宛先 207
CSD
リリース間での共用 500
CSD (CICS システム定義) ファイル
初期設定後のコンテンツ 491
リソース定義のグループ
DFHISC 491
DFHSTAND 491
DFHTERM 491
DFHTYPE 491
CSDH 一時データ宛先 207
CSDL 一時データ宛先 207
CSFL 一時データ宛先 207
CSFU トランザクション 408
CSJE 一時データ宛先 207
CSJO 一時データ宛先 207
CSKL 一時データ宛先 208
CSLB 一時データ宛先 208
CSML 一時データ宛先 208
CSNE 一時データ宛先 208
CSOO 一時データ宛先 208
CSPL 一時データ宛先 208
CSQL 一時データ宛先 209
CSRL 一時データ宛先 209
CSSH 一時データ宛先 209
CSSL 一時データ宛先 209
CSTL 一時データ宛先 209
CSZL 一時データ宛先 209
CSZX 一時データ宛先 209
CTLAPPLS オペランド
DFHRST TYPE=RSE 445
CWBO 一時データ宛先 209
CWBW 一時データ宛先 209

D

DATA12K 属性
LSRPOOL 定義 107
DATA16K 属性
LSRPOOL 定義 108
DATA1K 属性
LSRPOOL 定義 107
DATA20K 属性
LSRPOOL 定義 108
DATA24K 属性
LSRPOOL 定義 108
DATA28K 属性
LSRPOOL 定義 108
DATA2K 属性
LSRPOOL 定義 107
DATA32K 属性
LSRPOOL 定義 108
DATA4K 属性
LSRPOOL 定義 107
DATA512 属性
LSRPOOL 定義 107
DATA8K 属性
LSRPOOL 定義 107
DATABUFFERS 属性
FILE 定義 55
TDQUEUE 定義 193
DATAID オペランド
DFHTST TYPE=LOCAL 480
DFHTST TYPE=RECOVERY 478
DFHTST TYPE=REMOTE 481
DFHTST TYPE=SECURITY 483
DATASTREAM 属性
CONNECTION 定義 15
Db2 エントリー・リソース 36
Db2 トランザクション・リソース 42
DB2CONN 属性
DB2CONN 定義 24
DB2CONN 定義
コマンド・スレッド属性 34
COMAUTHID 34
COMAUTHTYPE 34
COMTHREADLIMIT 36
接続属性
CONNECTERROR 25
DB2GROUPID 25
DB2ID 25
MSGQUEUE1 25
MSGQUEUE2 25
MSGQUEUE3 25
NONTERMREL 25
PURGECYCLE 25
RESYNCMEMBER 25
SIGNID 25
STANDBYMODE 25
STATSQUEUE 25
TCBLIMIT 25
THREADERROR 25
プール・スレッド属性
ACCOUNTREC 30
AUTHID 30
AUTHTYPE 30
DROLLBACK 30
PLAN 30
PLANEXITNAME 30
THREADLIMIT 30
THREADWAIT 30
ACCOUNTREC 属性 30
AUTHID 属性 30
AUTHTYPE 属性 30
COMAUTHID 属性 34
COMAUTHTYPE 属性 34
COMTHREADLIMIT 属性 36
CONNECTERROR 属性 25
DB2CONN 属性 24
DB2GROUPID 属性 25
DB2ID 属性 25
DROLLBACK 属性 30
MSGQUEUE1 属性 25
MSGQUEUE2 属性 25
MSGQUEUE3 属性 25
NONTERMREL 属性 25
PLAN 属性 30
PLANEXITNAME 属性 30
PURGECYCLE 属性 25
RESYNCMEMBER 属性 25
SIGNID 属性 25
STANDBYMODE 属性 25

DB2CONN 定義 (続き)
STATSQUEUE 属性 25
TCBLIMIT 属性 25
THREADERROR 属性 25
THREADLIMIT 属性 30
THREADWAIT 属性 30
DB2CONN リソース 23
DB2ENTRY 属性
DB2ENTRY 定義 37
DB2ENTRY 定義
スレッド選択属性
TRANSID 37
スレッド操作属性
ACCOUNTREC 38
AUTHID 38
AUTHTYPE 39
DROLLBACK 40
PLAN 40
PLANEXITNAME 40
PRIORITY 41
PROTECTNUM 41
THREADLIMIT 41
THREADWAIT 41
ACCOUNTREC 属性 38
AUTHID 属性 38
AUTHTYPE 属性 39
DB2ENTRY 属性 37
DROLLBACK 属性 40
PLAN 属性 40
PLANEXITNAME 属性 40
PRIORITY 属性 41
PROTECTNUM 属性 41
THREADLIMIT 属性 41
THREADWAIT 属性 41
TRANSID 属性 37
DB2GROUPID 属性
DB2CONN 定義 25
DB2ID 属性
DB2CONN 定義 25
DB2TRAN 属性
DB2TRAN 定義 42
DB2TRAN 定義
DB2TRAN 属性 42
ENTRY 属性 42
RDO コマンド 42
TRANSID 属性 43
DBCTL (データベース制御) 444
DDNAME オペランド
順次装置 463
DFHTCT TYPE=SDSCI 463
DDNAME 属性
TDQUEUE 定義 193
DDName 属性
DOCTEMPLATE 定義 45
DEFINE コマンド
CEDA 345
DEFINE コマンド、DFHCSDUP ユーティリティー・プログラム 378
例 380
DEFSCREEN 属性
TYPETERM 定義 286
DELETE コマンド
CEDA 347
DELETE コマンド、DFHCSDUP ユーティリティー・プログラム 380
例 382
DELETE 属性
FILE 定義 55
DELIVER オプション
PERFORM オペランド 424
DESC オペランド
DFHCLT TYPE=WTO 402
DESCRIPTION 属性
セッション 163
DOCTEMPLATE 定義 45
ENQMODEL 定義 50
FILE 定義 55
IPCONN 定義 78
PROCESSTYPE 定義 136
TDQUEUE 定義 193
TRANSACTION 定義 241
TYPETERM 定義 287
DEVICE オペランド
順次装置 463
DFHTCT TYPE=SCSCI 463
DFHCLT オペランド
TYPE=COMMAND 401
TYPE=INITIAL 400
TYPE=LISTEND 403
TYPE=LISTSTART 401
TYPE=WTO 402
DFHCSDUP システム定義ユーティリティー・プログラム
ADD コマンド 371
ALTER コマンド 372
APPEND コマンド 374
COPY コマンド 375
DEFINE コマンド 378
DELETE コマンド 380
EXTRACT コマンド 382
INITIALIZE コマンド 384
LIST コマンド 385
PROCESS コマンド 387
REMOVE コマンド 388
SCAN コマンド 388
SERVICE コマンド 391
UPGRADE コマンド 391
USERDEFINE コマンド 392
VERIFY コマンド 394
DFHDLPSB オペランド
TYPE=ENTRY 403
TYPE=INITIAL 403
DFHFCT オペランド
TYPE=FILE 405
TYPE=INITIAL 405
DFHMCT オペランド
TYPE=EMP 420
TYPE=INITIAL 415
TYPE=RECORD 425
DFHPLT オペランド
TYPE=ENTRY 441
TYPE=INITIAL 441
DFHRDT、RDO に適格な TCT エントリー 451
DFHRST オペランド
TYPE=INITIAL 445
TYPE=RSE 445
TYPE=SUBSYS 445
DFHSIT マクロ
APPLID オペランド 10
DFHSRT オペランド
TYPE=INITIAL 447
TYPE=SYSTEM 447
TYPE=USER 447
DFHTCT オペランド
順次装置 463, 464, 465
リモート端末 469, 470, 471
TYPE=GROUP 453
TYPE=INITIAL 451
TYPE=LINE 464
TYPE=REGION 469
TYPE=REMOTE 471
TYPE=SDSCI 463
TYPE=TERMINAL 465, 470
DFHTLT オペランド
TYPE=ENTRY 474
TYPE=INITIAL 473
DFHTST オペランド
TYPE=ENTRY 478
TYPE=INITIAL 477
TYPE=RECOVERY 478, 480, 482
TYPE=SHARED 483
DFHXLT オペランド
TYPE=INITIAL 486
DISCREQ 属性
SESSIONS 定義 163
TYPETERM 定義 288
DISP オペランド
DFHFCT TYPE=FILE 407
DISPLAY GROUP コマンド
CEDA 350
DISPLAY LIST コマンド
CEDA 350
DISPLAY コマンド
CEDA 349
DISPOSITION 属性
FILE 定義 55
TDQUEUE 定義 193

DL/I (データ言語/I)
DL/I PSB ディレクトリー (PDIR) 403
DOCTEMPLATE 定義
APPENDCRLF 属性 45
DDName 属性 45
DESCRIPTION 属性 45
EXITPGM 属性 46
FILE 属性 46
HFSFILE 属性 46
MEMBERNAME 属性 47
NAME 属性 45
PROGRAM 属性 47
TDQUEUE 属性 47
TEMPLATENAME 属性 48
TSQUEUE 属性 48
TYPE 属性 48
DOCTEMPLATE リソース 44
DROLLBACK 属性
DB2CONN 定義 30
DB2ENTRY 定義 40
DSCNAME オペランド
順次装置 463
DFHTCT TYPE=SDSCI 463
DSN オペランド
DFHTCT TYPE=LDC 458
DSNAME オペランド
DFHFCT TYPE=FILE 407
DSNAME 属性
FILE 定義 55
LIBRARY 定義 102
TDQUEUE 定義 194
DSNSHARING 属性
FILE 定義 55
DTIMOUT 属性
TRANSACTION 定義 241
DUALCASEKYBD 属性
TYPETERM 定義 288
DUMP 属性
TRANSACTION 定義 242
DVC オペランド
DFHTCT TYPE=LDC 458
DVSUPRT 属性
PROFILE 定義 139
DYNAMIC 属性
TRANSACTION 定義 242

E

ENQMODEL 属性
ENQMODEL 定義 50
ENQMODEL 定義
DESCRIPTION 属性 50
ENQMODEL 属性 50
ENQNAME 属性 50
ENQSCOPE 属性 50
ENQMODEL 定義 (続き)
STATUS 属性 51
ENQMODEL リソース 48
ENQNAME 属性
ENQMODEL 50
ENQSCOPE 属性
ENQMODEL 定義 50
ENTRY 属性
DB2TRAN 定義 42
ERRATT オペランド
DFHTCT TYPE=INITIAL 452
ERRCOLOR 属性
TYPETERM 定義 288
ERRHILIGHT 属性
TYPETERM 定義 288
ERRINTENSIFY 属性
TYPETERM 定義 289
ERRLASTLINE 属性
TYPETERM 定義 289
ERROPTION 属性
TDQUEUE 定義 194
EXCLUDE オペランド
DFHMCT TYPE=RECORD 425
EXCNT オプション
PERFORM オペランド 423
EXITPGM 属性
DOCTEMPLATE 定義 46
EXPAND GROUP コマンド
CEDA 352
EXPAND LIST コマンド
CEDA 353
EXPAND コマンド
CEDA 351
EXPIRYINTMIN 属性
TSMODEL 定義 258
EXTENDEDDS 属性
TYPETERM 定義 289
EXTRACT コマンド、 DFHCSDUP ユーティリティー・プログラム 382
EXTRACT コマンド、DFHCSDUP ユーティリティー・プログラム
例 384
OBJECTS オプション 383
USERPROGRAM オプション 383
EXTSEC 属性
TRANSACTION 定義 489

F

FACILITYID 属性
TDQUEUE 定義 194
FACILITYLIKE 属性
PROFILE 定義 139
FCT (ファイル管理テーブル) 404
コーディング例 414
サマリー表 413
FCT (ファイル管理テーブル) (続き)
DFHFCT TYPE=FILE 405
DFHFCT TYPE=INITIAL 405
DFHFCT マクロ 404
FIELD オペランド
DFHMCT TYPE=EMP 422
FILE オペランド
DFHFCT TYPE=FILE 407
FILE 属性
DOCTEMPLATE 定義 46
FILE 定義 55
PROCESSTYPE 定義 136
FILE 定義
ADD 属性 55
BACKUPTYPE 属性 55, 56
BROWSE 属性 55
CFDTPOOL 属性 55
DATABUFFERS 属性 55
DELETE 属性 55
DESCRIPTION 属性 55
DISPOSITION 属性 55
DSNAME 属性 55
DSNSHARING 属性 55
FILE 属性 55
FWDRECOVLOG 属性 55
INDEXBUFFERS 属性 55
JNLADD 属性 55
JNLREAD 属性 55
JNLSYNCREAD 属性 55
JNLSYNCWRITE 属性 55
JNLUPDATE 属性 55
JOURNAL 属性 55
KEYLENGTH 属性 55
LOAD 属性 55
LSRPOOLID 属性 55
LSRPOOLNUM 属性 55
MAXNUMRECS 属性 55
NSRGROUP 属性 55
OPENTIME 属性 55
PASSWORD 属性 55
READINTEG 属性 55
RECORDFORMAT 属性 55
RECORDSIZE 属性 55
RECOVERY 属性 55
REMOTENAME 属性 71
REMOTESYSTEM 属性 72
RESSECNUM 属性 489
RLSACCESS 属性 72
STATUS 属性 73
STRINGS 属性 74
TABLE 属性 75
TABLENAME 属性 76
UPDATE 属性 76
UPDATEMODEL 属性 76
FILE リソース 51
FILE、DFHFCT TYPE= macro 405

FILSTAT オペランド
DFHFCT TYPE=FILE 408
FINAL オプション
LOCAL オペランド 460
FIXED オプション
RECFORM オペランド 411
FMHPARM 属性
TYPETERM 定義 289
FORALT オペランド
DFHCLT TYPE=LISTSTART 401
FORMFEED 属性
TYPETERM 定義 289
FWDRECOVLOG 属性
FILE 定義 55

G

GROUP オペランド
DFHTCT TYPE=GROUP 453

H

HFSFILE 属性
DOCTEMPLATE 定義 46
URIMAP 定義 317
HILIGHT 属性
TYPETERM 定義 290
HORIZFORM 属性
TYPETERM 定義 291
HOST 属性
IPCONN 定義 78
URIMAP 定義 318
HOSTCODEPAGE 属性
URIMAP 定義 319
HSDATA12K 属性
LSRPOOL 定義 109
HSDATA16K 属性
LSRPOOL 定義 109
HSDATA20K 属性
LSRPOOL 定義 109
HSDATA24K 属性
LSRPOOL 定義 109
HSDATA28K 属性
LSRPOOL 定義 109
HSDATA32K 属性
LSRPOOL 定義 109
HSDATA4K 属性
LSRPOOL 定義 108
HSDATA8K 属性
LSRPOOL 定義 108
HSINDEX12K 属性
LSRPOOL 定義 109
HSINDEX16K 属性
LSRPOOL 定義 109
HSINDEX20K 属性
LSRPOOL 定義 109
HSINDEX24K 属性
LSRPOOL 定義 109
HSINDEX28K 属性
LSRPOOL 定義 110
HSINDEX32K 属性
LSRPOOL 定義 110
HSINDEX4K 属性
LSRPOOL 定義 109
HSINDEX8K 属性
LSRPOOL 定義 109

I

ID オペランド
DFHMCT TYPE=EMP 421
IDPROP 属性
IPCONN 定義 78
IMS
CICS-IMS リンクおよびセッションの定義 9
INAREAL オペランド
順次装置 464
DFHTCT TYPE=LINE 464
INBFMH 属性
PROFILE 定義 139
INCLUDE オペランド
DFHMCT TYPE=RECORD 437
INDEX12K 属性
LSRPOOL 定義 110
INDEX16K 属性
LSRPOOL 定義 110
INDEX1K 属性
LSRPOOL 定義 110
INDEX20K 属性
LSRPOOL 定義 110
INDEX24K 属性
LSRPOOL 定義 110
INDEX28K 属性
LSRPOOL 定義 110
INDEX2K 属性
LSRPOOL 定義 110
INDEX32K 属性
LSRPOOL 定義 110
INDEX4K 属性
LSRPOOL 定義 110
INDEX512 属性
LSRPOOL 定義 110
INDEX8K 属性
LSRPOOL 定義 110
INDEXBUFFERS 属性
FILE 定義 55
INDIRECT 接続 159
リソース定義 10
INDIRECTNAME 属性
TDQUEUE 定義 195
INDOUBT 属性
TRANSACTION 定義 489
INDSYS 属性
CONNECTION
INDIRECT 接続用の 10
CONNECTION 定義 16
INITIAL オプション
LOCAL オペランド 460
INITIALIZE コマンド、DFHCSDUP ユーティリティー・プログラム 384
INITQNAME 属性
MQCONN 定義 115
INPUT オプション
TRMSTAT オペランド 467
INSERVICE 属性
CONNECTION 定義 16
IPCONN 定義 78
SESSIONS 定義 489
TERMINAL 定義 225
INSTALL コマンド
CEDA 354
INTENSIFY オプション
ERRATT オペランド 452
IOAREALEN 属性
SESSIONS 定義 163
TYPETERM 定義 291
IPADDRESS 属性
TCPIPSERVICE 定義 173
IPCONN 属性
IPCONN 定義 78
IPCONN 定義
APPLID 属性 78
AUTOCONNECT 属性 78
CERTIFICATE 属性 78
CIPHERS 属性 78
DESCRIPTION 属性 78
HOST 属性 78
IDPROP 属性 78
INSERVICE 属性 78
IPCONN 属性 78
MAXQTIME 属性 78
MIRRORLIFE 属性 78
NETWORKID 属性 78
PORT 属性 78
QUEUELIMIT 属性 78
RECEIVECOUNT 属性 78
SECURITYNAME 属性 78
SENDCOUNT 属性 78
SSL 属性 78
TCPIPSERVICE 属性 78
USERAUTH 属性 78
XLNACTION 属性 78
ISADSCN オペランド
順次装置 464

ISADSCN オペランド (続き)
DFHTCT TYPE=LINE 464
ISOLATE 属性
TRANSACTION 定義 243

J

JES オペランド
DFHCLT TYPE=INITIAL 400
JESCHAR オペランド
DFHCLT TYPE=INITIAL 400
JESID オペランド
DFHCLT TYPE=INITIAL 400
JID オペランド
DFHFCT TYPE=FILE 409
JNLADD 属性
FILE 定義 55
JNLREAD 属性
FILE 定義 55
JNLSYNCREAD 属性
FILE 定義 55
JNLSYNCWRITE 属性
FILE 定義 55
JNLUPDATE 属性
FILE 定義 55
JOBNAME オペランド
DFHRST TYPE=SUBSYS 446
JOURNAL 属性
FILE 定義 55
PROFILE 定義 140
JOURNALMODEL 属性
JOURNALMODEL 定義 90
JOURNALMODEL 定義
JOURNALMODEL 属性 90
JOURNALNAME 属性 90
STREAMNAME 属性 92
TYPE 属性 94
JOURNALMODEL リソース 89
JOURNALNAME 属性
JOURNALMODEL 定義 90
JREQ オペランド
DFHFCT TYPE=FILE 409
JVMSERVER リソース 96

K

KATAKANA 属性
TYPETERM 定義 291
KEY オプション
SERVREQ オペランド 412
KEYLEN オペランド
DFHFCT TYPE=FILE 410
KEYLENGTH 属性
FILE 定義 55

L

LASTLINE オプション
ERRATT オペランド 452
LDC オペランド
DFHTCT TYPE=LDC 458
DFHTCT TYPE=LDCLIST 461
DFHTLT TYPE=INITIAL 473
LDC リスト
拡張ローカル、およびシステム LDC テーブル、DFHTCT TYPE=LDC 457
装置 215
ローカル、DFHTCT TYPE=LDCLIST 461
LDC リスト対応の装置 215
LDC (論理装置コード) 454, 461
LDCLIST 属性
TYPETERM 定義 291
LIBRARY 定義
CRITICAL 属性 101
DSNAME 属性 102
RANKING 属性 103
STATUS 属性 104
LIGHTPEN 属性
TYPETERM 定義 292
LINSTAT='OUT OF SERVICE' オペランド
順次装置 465
DFHTCT TYPE=LINE 465
LIST コマンド、DFHCSDUP ユーティリティー・プログラム 385
例 386
OBJECTS オプション 385
LOAD 属性
FILE 定義 55
LOCAL オペランド
DFHTCT TYPE=LDC 460
LOCALQ 属性
TRANSACTION 定義 244
LOCATION 属性
TSMODEL 定義 259
URIMAP 定義 320
LOCK コマンド
CEDA 356
LOG オペランド
DFHFCT TYPE=FILE 410
LOGMODE 属性
TYPETERM 定義 292
LOGMODECOM 属性
TRANSACTION 定義 489
LOGONMSG 属性
TYPETERM 定義 294
LOGREC 属性
PROFILE 定義 140
LPLEN オペランド
順次装置 465
DFHTCT TYPE=TERMINAL 465
LRECL オペランド
DFHFCT TYPE=FILE 411
LSRPOOL 属性
LSRPOOL 定義 111
LSRPOOL 定義
DATA12K 属性 107
DATA16K 属性 108
DATA1K 属性 107
DATA20K 属性 108
DATA24K 属性 108
DATA28K 属性 108
DATA2K 属性 107
DATA32K 属性 108
DATA4K 属性 107
DATA512 属性 107
DATA8K 属性 107
HSDATA12K 属性 109
HSDATA16K 属性 109
HSDATA20K 属性 109
HSDATA24K 属性 109
HSDATA28K 属性 109
HSDATA32K 属性 109
HSDATA4K 属性 108
HSDATA8K 属性 108
HSINDEX12K 属性 109
HSINDEX16K 属性 109
HSINDEX20K 属性 109
HSINDEX24K 属性 109
HSINDEX28K 属性 110
HSINDEX32K 属性 110
HSINDEX4K 属性 109
HSINDEX8K 属性 109
INDEX12K 属性 110
INDEX16K 属性 110
INDEX1K 属性 110
INDEX20K 属性 110
INDEX24K 属性 110
INDEX28K 属性 110
INDEX2K 属性 110
INDEX32K 属性 110
INDEX4K 属性 110
INDEX512 属性 110
INDEX8K 属性 110
LSRPOOL 属性 111
LSRPOOLID 属性 111, 489
LSRPOOLNUM 属性 111
MAXKEYLENGTH 属性 111
SHARELIMIT 属性 111
STRINGS 属性 111
LSRPOOL リソース 105
LSRPOOLID 属性
FILE 定義 55
LSRPOOL 定義 111, 489

LSRPOOLNUM 属性
FILE 定義 55
LSRPOOL 定義 111
LUTYPE4 オプション
LDC オペランド 459
LUTYPE6.1
CICS-CICS ISC リンクおよびセッション
リソース定義 8
CICS-IMS リンクおよびセッション
リソース定義 9
MRO セッション
リソース定義 7
LUTYPE6.1 CICS-CICS ISCリンクとセッション 159
LUTYPE6.1 CICS-IMS リンクおよびセッション 159

M

MACRF オペランド
順次装置 464
DFHTCT TYPE=SDSCI 464
MAPSET 定義
MAPSET 属性 113
RESIDENT 属性 114
RSL 属性 489
USAGE 属性 114
USELPACOPY 属性 114
MAPSET リソース 112
MAXACTIVE 属性
TRANCLASS 定義 235
MAXDATALEN 属性
TCPIPSERVICE 定義 173
MAXIMUM 属性
セッション
APPC 単一セッション端末用の 8
APPC 並列セッション用の 8
SESSIONS 定義 164
MAXKEYLENGTH 属性
LSRPOOL 定義 111
MAXNUMRECS 属性
FILE 定義 55
MAXPERSIST 属性
TCPIPSERVICE 定義 173
MAXQTIME 属性
CONNECTION 定義 16
IPCONN 定義 78
MCT (モニター管理テーブル) 414
コード化例 438
制御データの記録 425
ユーザー・イベント・モニター・ポイント 420
DFHMCT TYPE=EMP 420
DFHMCT TYPE=INITIAL 415
DFHMCT TYPE=RECORD 425
MCT (モニター管理テーブル) (続き)
DFHMCT マクロ 415
MEDIATYPE 属性
URIMAP 定義 320
MEMBERNAME 属性
DOCTEMPLATE 定義 47
MF オペランド
DFHCLT TYPE=WTO 402
MIGRATE オペランド
DFHTCT TYPE=INITIAL 452
DFHTST TYPE=INITIAL 477
MIRRORLIFE 属性
IPCONN 定義 78
MLTCNT オプション
PERFORM オペランド 423
MODENAME 属性
セッション
APPC 単一セッション端末用の 8
APPC 並列セッション用の 8
PROFILE 定義 140
SESSIONS 定義 164
TERMINAL 定義 225
MOVE コマンド
CEDA 358
MQCONN
インストール 115
MQCONN 属性
MQCONN 定義 115
MQCONN 定義
INITQNAME 属性 115
MQCONN 属性 115
MQNAME 属性 115
RESYNCMEMBER 属性 115
MQCONN リソース 115
MQNAME 属性
MQCONN 定義 115
MRO リンクおよびセッション 159
MSGINTEG 属性
PROFILE 定義 141
MSGJRNL 属性
PROFILE 定義 141
MSGQUEUE1 属性
DB2CONN 定義 25
MSGQUEUE2 属性
DB2CONN 定義 25
MSGQUEUE3 属性
DB2CONN 定義 25
MSRCONTROL 属性
TYPETERM 定義 294
MXSSASZ オペランド
DFHDLPSB TYPE=ENTRY 404

N

NACNT オプション
PERFORM オペランド 424
NAME 属性
DOCTEMPLATE 定義 45
NATLANG 属性
TERMINAL 定義 226
NEPCLASS 属性
PROFILE 定義 141
SESSIONS 定義 165
TYPETERM 定義 294
NETNAME 属性
CONNECTION 17
INDIRECT 接続用の 10
CONNECTION 定義 17
PARTNER 定義 128
TERMINAL 定義 226
NETNAMEQ 属性
セッション
LUTYPE6.1 CICS-IMS セッション用の 9
SESSIONS 定義 165
NETWORK 属性
PARTNER 定義 129
NETWORKID 属性
IPCONN 定義 78
NOEXCTL オプション
SERVREQ オペランド 412
NONTERMREL 属性
DB2CONN 定義 25
NONVTAM オプション
ACCMETH=NONVTAM の指定 452
NSRGROUP 属性
FILE 定義 55

O

OBFORMAT 属性
TYPETERM 定義 294
OBJECTS オプション
LIST コマンド (DFHCSDUP) 385
OBOPERID 属性
TYPETERM 定義 295
ONEWTE 属性
PROFILE 定義 142
OPENTIME 属性
FILE 定義 55
TDQUEUE 定義 195
OPERID 属性
SESSIONS 定義 489
OPERPRIORITY 属性
SESSIONS 定義 489
OPERRSL 属性
SESSIONS 定義 489
OPERSECURITY 属性
SESSIONS 定義 489
ORCNT オプション
PERFORM オペランド 424

OSADSCN オペランド
順次装置 465
DFHTCT TYPE=LINE 465
OUT OF SERVICE オプション
LINSTAT オペランド 465
TRMSTAT オペランド 467
OUTLINE 属性
TYPETERM 定義 295

P

PAGESIZE 属性
TYPETERM 定義 295
PARTITIONS 属性
TYPETERM 定義 296
PARTITIONSET 属性
TRANSACTION 定義 245
PARTITIONSET 定義
PARTITIONSET 属性 125
RESIDENT 属性 126
RSL 属性 489
STATUS 属性 126
USAGE 属性 126
USELPACOPY 属性 126
PARTITIONSET リソース 124
PARTNER 属性
PARTNER 定義 129
PARTNER 定義
NETNAME 属性 128
NETWORK 属性 129
PARTNER 属性 129
PROFILE 属性 129
TPNAME 属性 129
XTPNAME 属性 129
PARTNER リソース 127
PASSWORD 属性
FILE 定義 55
PATH 属性
URIMAP 定義 321
PCLOCK オプション
PERFORM オペランド 424
PCPUCLK オプション
PERFORM オペランド 424
PDIR (プログラム仕様ブロック・ディレクトリー)
DFHDLPSB TYPE=ENTRY 403
DFHDLPSB TYPE=INITIAL 403
DFHDLPSB マクロ 403
PERFORM オペランド
DFHMCT TYPE=EMP 422
PGESIZE オペランド
順次装置 466
DFHTCT TYPE=LDC 460
DFHTCT TYPE=TERMINAL 466
PGESTAT オペランド
DFHTCT TYPE=LDC 460
PIPELINE 属性
URIMAP 定義 322
PIPELINE 定義
RESPWAIT 属性 132
SHELF 属性 132
WSDIR 属性 133
PIPELINE リソース 130
PLAN 属性
DB2CONN 定義 30
DB2ENTRY 定義 40
PLANEXITNAME 属性
DB2CONN 定義 30
DB2ENTRY 定義 40
PLT (プログラム・リスト・テーブル) 439
コーディング例 443
制御セクション 441
DFHPLT TYPE=ENTRY 441
DFHPLT TYPE=INITIAL 441
POOL 属性
TERMINAL 定義 227
POOLNAME 属性
TSMODEL 定義 260
PORT 属性
IPCONN 定義 78
URIMAP 定義 322
PORTNUMBER 属性
TCPIPSERVICE 定義 173
PREFIX 属性
TSMODEL 定義 260
PRIMEDSIZE 属性
TRANSACTION 489
PRINTADAPTER 属性
TYPETERM 定義 296
PRINTCONTROL 属性
TDQUEUE 定義 195
PRINTER 属性
TERMINAL 定義 227
PRINTERCOMP 属性
PROFILE 定義 142
PRINTERCOPY 属性
TERMINAL 定義 228
PRIORITY 属性
DB2ENTRY 定義 41
TRANSACTION 定義 246
PRIVACY 属性
TCPIPSERVICE 定義 489
PROCESS コマンド、DFHCSDUP ユーティリティー・プログラム 387
PROCESSTYPE 属性
PROCESSTYPE 定義 137
PROCESSTYPE 定義
AUDITLEVEL 属性 135
AUDITLOG 属性 136
DESCRIPTION 属性 136
FILE 属性 136
PROCESSTYPE 属性 137
PROCESSTYPE リソース 134
PROFILE 属性
PARTNER 定義 129
PROFILE 定義 142
TRANSACTION 定義 246
PROFILE 定義
CHAINCONTROL 属性 138
CICS 提供の 513, 514, 517
DVSUPRT 属性 139
FACILITYLIKE 属性 139
INBFMH 属性 139
JOURNAL 属性 140
LOGREC 属性 140
MODENAME 属性 140
MSGINTEG 属性 141
MSGJRNL 属性 141
NEPCLASS 属性 141
ONEWTE 属性 142
PRINTERCOMP 属性 142
PROFILE 属性 142
PROTECT 属性 489
RAQ 属性 142
RTIMOUT 属性 143
SCRNSIZE 属性 143
UCTRAN 属性 144
PROFILE リソース 137
PROGRAM オペランド
DFHPLT TYPE=ENTRY 441
PROGRAM 属性
DOCTEMPLATE 定義 47
TRANSACTION 定義 246
URIMAP 定義 323
PROGRAM 定義
LANGUAGE 属性
RPG 489
RSL 属性 489
PROGRAM リソース 145
PROGSYMBOLS 属性
TYPETERM 定義 297
PROTECT 属性
PROFILE 489
PROTECTNUM 属性
DB2ENTRY 定義 41
PROTOCL(HTTP) TCPIPSERVICES 185
PROTOCOL 属性
セッション
APPC 単一セッション端末用の 8
APPC 並列セッション用の 8
LUTYPE6.1 CICS-IMS セッション用の 9
LUTYPE6.1 セッション用の 8
MRO セッション用の 7
CONNECTION
APPC 単一セッション端末用の 8
APPC リンク用の 8

PROTOCOL 属性 (続き)
CONNECTION (続き)
LUTYPE6.1 CICS-IMS リンク用の 9
LUTYPE6.1 リンク用の 8
MRO リンク用の 7
CONNECTION 定義 19
SESSIONS 定義 165
TCPIPSERVICE 定義 173
PSB オペランド
DFHDLPSB TYPE=ENTRY 404
PSB ディレクトリー (PDIR)
DFHDLPSB TYPE=ENTRY 403
DFHDLPSB TYPE=INITIAL 403
DFHDLPSB マクロ 403
PSB (プログラム仕様ブロック) 403, 404
PSRECOVERY 属性
CONNECTION 定義 19
PURGECYCLE 属性
DB2CONN 定義 25
PURGETHRESH 属性
TRANCLASS 定義 235

Q

QUERY 属性
TYPETERM 定義 297
QUEUELIMIT 属性
CONNECTION 定義 20
IPCONN 定義 78

R

RANKING 属性
LIBRARY 定義 103
RAQ 属性
PROFILE 定義 142
RDO に適格な TCT エントリー、DFHRDT 451
RDO への TCT エントリーのマイグレーション 453
READ オプション
SERVREQ オペランド 412
READINTEG 属性
FILE 定義 55
REALM 属性
TCPIPSERVICE 定義 173
RECEIVE オプション
TRMSTAT オペランド 467
RECEIVECOUNT 属性
セッション
LUTYPE6.1 CICS-IMS セッション用の 9
LUTYPE6.1 セッション用の 8
MRO セッション用の 7
RECEIVECOUNT 属性 (続き)
IPCONN 定義 78
SESSIONS 定義 165
RECEIVEPFX 属性
セッション
LUTYPE6.1 セッション用の 8
MRO セッション用の 7
SESSIONS 定義 166
RECEIVESIZE 属性
SESSIONS 定義 167
TYPETERM 定義 297
RECFM オペランド
順次装置 464
DFHTCT TYPE=SDSCI 464
RECFORM オペランド
DFHFCT TYPE=FILE 411
RECORDFORMAT 属性
CONNECTION 定義 20
FILE 定義 55
TDQUEUE 定義 195
RECORDSIZE 属性
FILE 定義 55
TDQUEUE 定義 196
RECOVER オペランド
DFHSRT TYPE=SYSTEM 448
DFHSRT TYPE=USER 448
RECOVERY 属性
FILE 定義 55
RECOVNOTIFY 属性
SESSIONS 定義 489
TYPETERM 定義 299
RECOVOPTION 属性
SESSIONS 定義 167
TYPETERM 定義 299
RECOVSTATUS 属性
TDQUEUE 定義 196
TSMODEL 定義 261
REDIRECTTYPE 属性
URIMAP 定義 323
RELREQ 属性
SESSIONS 定義 169
TYPETERM 定義 301
RELTYPE オペランド
DFHFCT TYPE=FILE 412
REMOTELENGTH 属性
TDQUEUE 定義 198
REMOTENAME 属性
CONNECTION 定義 21
FILE 定義 71
TDQUEUE 定義 197
TERMINAL 定義 228
TRANSACTION 定義 247
REMOTEPREFIX 属性
TSMODEL 定義 261
REMOTESYSNET 属性
CONNECTION 定義 21
REMOTESYSNET 属性 (続き)
TERMINAL 定義 228
REMOTESYSTEM 属性
CONNECTION 定義 21
FILE 定義 72
TDQUEUE 定義 198
TERMINAL 定義 229
TRANSACTION 定義 247
TSMODEL 定義 261
REMOVE コマンド
CEDA 360
REMOVE コマンド、DFHCSDUP ユーティリティー・プログラム 388
例 388
RENAME コマンド
CEDA 362
RESOURCENAME 属性
ATOMSERVICE 定義 4
RESOURCETYPE 属性
ATOMSERVICE 定義 4
RESPWAIT 属性
PIPELINE 定義 132
RESSEC 属性
TRANSACTION 定義 248
RESSECNUM 属性
FILE 定義 489
RESTART 属性
TRANSACTION 定義 248
RESYNCMEMBER 属性
DB2CONN 定義 25
MQCONN 定義 115
REWIND 属性
TDQUEUE 定義 198
RKP オペランド
DFHFCT TYPE=FILE 412
RLSACCESS 属性
FILE 定義 72
RLSACCESS(YES)
他の FILE 属性への影響 72
RMTNAME オペランド
リモート端末 470, 471
DFHDLPSB TYPE=ENTRY 404
DFHTCT TYPE=REMOTE 471
DFHTCT TYPE=TERMINAL 470
DFHTST TYPE=REMOTE 482
ROUTABLE 属性
TRANSACTION 定義 248
ROUTCDE オペランド
DFHCLT TYPE=WTO 402
ROUTEDMSGS 属性
TYPETERM 定義 301
RPG 値
LANGUAGE 属性の PROGRAM 定義の 489
RSL 属性
MAPSET 定義 489

RSL 属性 (続き)
PARTITIONSET 定義 489
PROGRAM 定義 489
TRANSACTION 定義 489
RST (リカバリー可能サービス・テーブル) 444, 446
コーディング例 446
DFHRST TYPE=INITIAL 445
DFHRST TYPE=RSE 445
DFHRST TYPE=SUBSYS 445
DFHRST マクロ・タイプ 444
RSTSIGNOFF 属性
TYPETERM 定義 302
RTIMOUT 属性
PROFILE 定義 143
RUNAWAY 属性
TRANSACTION 定義 249

S

SCAN コマンド、DFHCSDUP ユーティリティー・プログラム 388
SCHEME 属性
URIMAP 定義 324
SCLOCK オプション
PERFORM オペランド 424
SCPUCLK オプション
PERFORM オペランド 424
SCRNSIZE 属性
PROFILE 定義 143
SECURITY 属性
TSMODEL 定義 262
SECURITYNAME 属性
CONNECTION 定義 22
IPCONN 定義 78
TERMINAL 定義 229
SENDCOUNT 属性
セッション
LUTYPE6.1 CICS-IMS セッション用の 9
LUTYPE6.1 セッション用の 8
MRO セッション用の 7
IPCONN 定義 78
SESSIONS 定義 169
SENDPFX 属性
セッション
LUTYPE6.1 セッション用の 8
MRO セッション用の 7
SESSIONS 定義 169
SENDSIZE 属性
SESSIONS 定義 170
TYPETERM 定義 302
SERVICE コマンド、DFHCSDUP ユーティリティー・プログラム 391
FROMCSD オペランド 391
LEVEL オペランド 391
SERVREQ オペランド
DFHFCT TYPE=FILE 412
SESSIONS 属性
SESSIONS 定義 171
SESSIONS 定義
インストール 159
AUTOCONNECT 属性 161
BUILDCHAIN 属性 162
CONNECTION 属性 162
DESCRIPTION 属性 163
DISCREQ 属性 163
INSERVICE 属性 489
IOAREALEN 属性 163
MAXIMUM 属性 164
MODENAME 属性 164
NEPCLASS 属性 165
NETNAMEQ 属性 165
OPERID 属性 489
OPERPRIORITY 属性 489
OPERRSL 属性 489
OPERSECURITY 属性 489
PROTOCOL 属性 165
RDO コマンド 160
RECEIVECOUNT 属性 165
RECEIVEPFX 属性 166
RECEIVESIZE 属性 167
RECOVNOTIFY 属性 489
RECOVOPTION 属性 167
RELREQ 属性 169
SENDCOUNT 属性 169
SENDPFX 属性 169
SENDSIZE 属性 170
SESSIONS 属性 171
SESSNAME 属性 171
SESSPRIORITY 属性 171
TRANSACTION 属性 489
USERAREALEN 属性 171
USERID 属性 171
SESSIONS リソース 159
SESSIONTYPE 属性
TYPETERM 定義 303
SESSNAME 属性
セッション
LUTYPE6.1 CICS-IMS セッション用の 9
SESSIONS 定義 171
SESSPRIORITY 属性
SESSIONS 定義 171
SHARELIMIT 属性
LSRPOOL 定義 111
SHELF 属性
PIPELINE 定義 132
SHIPPABLE 属性
TYPETERM 定義 303
SHUTDOWN 属性
TRANSACTION 定義 249
SIGNID 属性
DB2CONN 定義 25
SIGNOFF 属性
TYPETERM 定義 303
SINGLESESS 属性
CONNECTION
APPC 単一セッション端末用の 8
CONNECTION 定義 22
SOCKETCLOSE 属性 173
SOLICITED 属性
TERMINAL 定義 230
SOSI 属性
TYPETERM 定義 304
SPURGE 属性
TRANSACTION 定義 249
SRCHM オペランド
DFHFCT TYPE=FILE 413
SRT (システム・リカバリー・テーブル)
コーディング例 448
DFHSRT TYPE=INITIAL 447
SSL 属性
IPCONN 定義 78
STANDBYMODE 属性
DB2CONN 定義 25
STATSQUEUE 属性
DB2CONN 定義 25
STATUS 属性
ATOMSERVICE 定義 4
ENQMODEL 定義 51
FILE 定義 73
LIBRARY 定義 104
TRANSACTION 定義 250
URIMAP 定義 324
STORAGECLEAR 属性
TRANSACTION 定義 250
STREAMNAME 属性
JOURNALMODEL 定義 92
STRINGS 属性
FILE 定義 74
LSRPOOL 定義 111
SUBCNT オプション
PERFORM オペランド 424
SUBSYSID オペランド
DFHRST TYPE=SUBSYS 446
SUFFIX オペランド
DFHTLT TYPE=INITIAL 473
SYSIDNT オペランド
リモート端末 469, 470, 471
DFHDLPSB TYPE=ENTRY 404
DFHSIT TYPE=INITIAL 453
DFHTCT TYPE=INITIAL 453
DFHTCT TYPE=REGION 469
DFHTCT TYPE=REMOTE 471
DFHTCT TYPE=TERMINAL 470
DFHTST TYPE=REMOTE 482
TCT での小文字 450

SYSOUTCLASS 属性
TDQUEUE 定義 198
SYSTEM オプション
LDC オペランド 459

T

TABLE 属性
FILE 定義 75
TABLENAME 属性
FILE 定義 76
TASKDATAKEY 属性
TRANSACTION 定義 250
TASKDATALOC 属性
TRANSACTION 定義 251
TASKLIMIT 属性
TERMINAL 定義 230
TASKREQ オペランド
DFHXLT TYPE=ENTRY 487
TASKREQ 属性
TRANSACTION 定義 252
TCBLIMIT 属性
DB2CONN 定義 25
TCLASS 属性
TRANSACTION 定義 489
TCPIPSERVICE 属性
IPCONN 定義 78
TCPIPSERVICE 定義 173
URIMAP 定義 325
TCPIPSERVICE 定義
ATTACHSEC 属性 173
AUTHENTICATE 属性 173
BACKLOG 属性 173
CERTIFICATE 属性 173
CIPHERS 属性 173
IPADDRESS 属性 173
MAXDATALEN 属性 173
MAXPERSIST 属性 173
PORTNUMBER 属性 173
PRIVACY 属性 489
PROTOCOL 属性 173
REALM 属性 173
SOCKETCLOSE 属性 173
SSL 属性 173
STATUS 属性 173
TCPIPSERVICE 属性 173
TRANSACTION 属性 173
TSQPREFIX 属性 173
URM 属性 173
TCPIPSERVICE リソース 172
TCT (端末管理テーブル) 449, 472
概要 449
順次装置 462
CICS 端末のリスト 471
DFHTCT TYPE=GROUP 453
DFHTCT マクロ、要約 450
TCT (端末管理テーブル) (続き)
RDO へのエントリーのマイグレーション 453
TCTUAL オペランド
順次装置 465, 466
DFHTCT TYPE=LINE 465
DFHTCT TYPE=TERMINAL 466
TDQUEUE 属性
DOCTEMPLATE 定義 47
TDQUEUE 定義 199
TDQUEUE 定義
ATIFACILITY 属性 191
BLOCKFORMAT 属性 192
BLOCKSIZE 属性 192
DATABUFFERS 属性 193
DDNAME 属性 193
DESCRIPTION 属性 193
DISPOSITION 属性 193
DSNAME 属性 194
ERROPTION 属性 194
FACILITYID 属性 194
INDIRECTNAME 属性 195
OPENTIME 属性 195
PRINTCONTROL 属性 195
RECORDFORMAT 属性 195
RECORDSIZE 属性 196
RECOVSTATUS 属性 196
REMOTELENGTH 属性 198
REMOTENAME 属性 197
REMOTESYSTEM 属性 198
REWIND 属性 198
SYSOUTCLASS 属性 198
TDQUEUE 属性 199
TRANSID 属性 200
TRIGGERLEVEL 属性 200
TYPE 属性 201
TYPEFILE 属性 202
USERID 属性 202
WAIT 属性 203
WAITACTION 属性 203
TDQUEUE リソース 188
TEMPLATENAME 属性
DOCTEMPLATE 定義 48
URIMAP 定義 325
TERMINAL 属性
TERMINAL 定義 230
TERMINAL ターミナル 210
TERMINAL 定義
パイプライン 213
ALTPRINTCOPY 属性 222
ALTPRINTER 属性 222
ATTACHSEC 属性 223
AUTINSTMODEL 属性 223
AUTINSTNAME 属性 224
BINDSECURITY 属性 224
CONSNAME 属性 224
TERMINAL 定義 (続き)
CONSOLE 属性 489
INSERVICE 属性 225
MODENAME 属性 225
NATLANG 属性 226
NETNAME 属性 226
POOL 属性 227
PRINTER 属性 227
PRINTERCOPY 属性 228
REMOTENAME 属性 228
REMOTESYSNET 属性 228
REMOTESYSTEM 属性 229
SECURITYNAME 属性 229
SOLICITED 属性 230
TASKLIMIT 属性 230
TERMINAL 属性 230
TERMPRIORITY 属性 231
TRANSACTION 属性 231
TYPETERM 属性 232
USEDFLTUSER 属性 232
USERID 属性 232
TERMMODEL 属性
TYPETERM 定義 304
TERMPRIORITY 属性
TERMINAL 定義 231
TEXTKYBD 属性
TYPETERM 定義 304
TEXTPRINT 属性
TYPETERM 定義 305
THREADERROR 属性
DB2CONN 定義 25
THREADLIMIT 属性
DB2CONN 定義 30
DB2ENTRY 定義 41
THREADWAIT 属性
DB2CONN 定義 30
DB2ENTRY 定義 41
TLT (端末リスト・テーブル) 475
コード化例 475
制御セクション 473
DFHTLT TYPE=ENTRY 474
DFHTLT TYPE=INITIAL 473
TPNAME 属性
PARTNER 定義 129
TRANSACTION 定義 252
TPURGE 属性
TRANSACTION 定義 253
TRACE 属性
TRANSACTION 定義 253
TRANCLASS 属性
TRANSACTION 定義 236, 253
TRANCLASS 定義
MAXACTIVE 属性 235
PURGETHRESH 属性 235
TRANCLASS リソース 234

TRANSACTION オプション
TRMSTAT オペランド 466
TRANSACTION 属性
SESSIONS 定義 489
TERMINAL 定義 231
TRANSACTION 定義 253
URIMAP 定義 326
TRANSACTION 定義
ACTION 属性 239
ALIAS 属性 239
BREXIT 属性 240
CMDSEC 属性 240
CONFDATA 属性 240
DESCRIPTION 属性 241
DTIMOUT 属性 241
DUMP 属性 242
DYNAMIC 属性 242
EXTSEC 属性 489
INDOUBT 属性 489
ISOLATE 属性 243
LOCALQ 属性 244
LOGMODECOM 属性 489
PARTITIONSET 属性 245
PRIMEDSIZE 属性 489
PRIORITY 属性 246
PROFILE 属性 246
PROGRAM 属性 246
REMOTENAME 属性 247
REMOTESYSTEM 属性 247
RESSEC 属性 248
RESTART 属性 248
ROUTABLE 属性 248
RSL 属性 489
RUNAWAY 属性 249
SHUTDOWN 属性 249
SPURGE 属性 249
STATUS 属性 250
STORAGECLEAR 属性 250
TASKDATAKEY 属性 250
TASKDATALOC 属性 251
TASKREQ 属性 252
TCLASS 属性 489
TPNAME 属性 252
TPURGE 属性 253
TRACE 属性 253
TRANCLASS 属性 236, 253
TRANSACTION 属性 253
TRANSEC 属性 489
TRPROF 属性 254
TWASIZE 属性 254
WAIT 属性 254
WAITTIME 属性 256
XTPNAME 属性 257
XTRANID 属性 257
TRANSACTION リソース 236
TRANSCEIVE オプション
TRMSTAT オペランド 467
TRANSEC 属性
TRANSACTION 定義 489
TRANSID オペランド
順次装置 466
DFHTCT TYPE=TERMINAL 466
DFHXLT TYPE=ENTRY 487
TRANSID 属性
DB2ENTRY 定義 37
DB2TRAN 定義 43
TDQUEUE 定義 200
TRIGGERLEVEL 属性
TDQUEUE 定義 200
TRMIDNT オペランド
順次装置 466
リモート端末 470, 471
DFHTCT TYPE=REMOTE 471
DFHTCT TYPE=TERMINAL 466, 470
DFHTLT TYPE=ENTRY 474
TCT での小文字 450
TRMPRTY オペランド
順次装置 466
DFHTCT TYPE=TERMINAL 466
TRMSTAT オペランド
順次装置 466
DFHTCT TYPE=TERMINAL 466
TRMTYPE オペランド
順次装置 465
リモート端末 470, 471
DFHTCT TYPE=LINE 465
DFHTCT TYPE=REMOTE 471
DFHTCT TYPE=TERMINAL 470
TRPROF 属性
TRANSACTION 定義 254
TSAGE オペランド
DFHTST TYPE=INITIAL 478
TSMODEL 属性
TSMODEL 定義 262
TSMODEL 定義
EXPIRYINTMIN 属性 258
LOCATION 属性 259
POOLNAME 属性 260
PREFIX 属性 260
RECOVSTATUS 属性 261
REMOTEPREFIX 属性 261
REMOTESYSTEM 属性 261
SECURITY 属性 262
TSMODEL 属性 262
XPREFIX 属性 262
XREMOTEPFX 属性 262
TSMODEL リソース 258
TSQPREFIX 属性
TCPIPSERVICE 定義 173
TSQUEUE 属性
DOCTEMPLATE 定義 48
TST (一時記憶域テーブル) 475, 484
一時記憶域セキュリティー検査、
DFHTST TYPE=SECURITY 482
一時記憶域データ共用、 DFHTST
TYPE=SHARED 483
制御セクション 477
データ共用一時記憶域 483
リカバリー可能一時記憶域 478, 480,
482
リカバリー可能一時記憶域、 DFHTST
TYPE=RECOVERY 478
リモート一時記憶域 DATAID、
DFHTST TYPE=REMOTE 481
ローカル一時記憶域
DATAID、DFHTST
TYPE=LOCAL 480
DFHTST TYPE=INITIAL 477
DFHTST TYPE=RECOVERY 478, 480,
482
DFHTST TYPE=SHARED 483
TTI 属性
TYPETERM 定義 305
TWASIZE 属性
TRANSACTION 定義 254
TYPE 属性
DOCTEMPLATE 定義 48
JOURNALMODEL 定義 94
TDQUEUE 定義 201
TYPEFILE 属性
TDQUEUE 定義 202
TYPETERM
属性のデフォルト値 265
TYPETERM 属性
TERMINAL 定義 232
TYPETERM 定義 305
TYPETERM 定義
ALTPAGE 属性 280
ALTSCREEN 属性 281
ALTSUFFIX 属性 282
APLKYBD 属性 282
APLTEXT 属性 282
ASCII 属性 283
ATI 属性 283
AUDIBLEALARM 属性 284
AUTOCONNECT 属性 284
AUTOPAGE 属性 284
BACKTRANS 属性 285
BRACKET 属性 285
BUILDCHAIN 属性 285
CGCSGID 属性 285
COLOR 属性 286
COPY 属性 286
CREATESESS 属性 286
DEFSCREEN 属性 286
DESCRIPTION 属性 287
DISCREQ 属性 288

TYPETERM 定義 (続き)
DUALCASEKYBD 属性 288
ERRCOLOR 属性 288
ERRHILIGHT 属性 288
ERRINTENSIFY 属性 289
ERRLASTLINE 属性 289
EXTENDEDDS 属性 289
FMHPARM 属性 289
FORMFEED 属性 289
HILIGHT 属性 290
HORIZFORM 属性 291
IOAREALEN 属性 291
KATAKANA 属性 291
LDCLIST 属性 291
LIGHTPEN 属性 292
LOGMODE 属性 292
LOGONMSG 属性 294
MSRCONTROL 属性 294
NEPCLASS 属性 294
OBFORMAT 属性 294
OBOPERID 属性 295
OUTLINE 属性 295
PAGESIZE 属性 295
PARTITIONS 属性 296
PRINTADAPTER 属性 296
PROGSYMBOLS 属性 297
QUERY 属性 297
RECEIVESIZE 属性 297
RECOVNOTIFY 属性 299
RECOVOPTION 属性 299
RELREQ 属性 301
ROUTEDMSGS 属性 301
RSTSIGNOFF 属性 302
SENDSIZE 属性 302
SESSIONTYPE 属性 303
SHIPPABLE 属性 303
SIGNOFF 属性 303
SOSI 属性 304
TERMMODEL 属性 304
TEXTKYBD 属性 304
TEXTPRINT 属性 305
TTI 属性 305
TYPETERM 属性 305
UCTRAN 属性 305
USERAREALEN 属性 306
VALIDATION 属性 307
VERTICALFORM 属性 307
TYPETERM リソース 263
TYPE=LOCAL、DFHTST マクロ
ローカル一時記憶域
DATAID、DFHTST 480
TYPE=REMOTE、 DFHTST マクロ
リモート一時記憶域
DATAID、DFHTST 481
TYPE=SECURITY マクロ
一時記憶域セキュリティー検査、
DFHTST 482
TYPE=SHARED マクロ
一時記憶域データ共用、 DFHTST 483

U

UCTRAN 属性
PROFILE 定義 144
TYPETERM 定義 305
UNBLOCKED オプション
RECFORM オペランド 411
UNDEFINED オプション
RECFORM オペランド 411
UNLOCK コマンド
CEDA 364
UPDATE オプション
SERVREQ オペランド 413
UPDATE 属性
FILE 定義 76
UPDATEMODEL 属性
FILE 定義 76
UPGRADE コマンド、DFHCSDUP ユーティリティー・プログラム 391
REPLACE オペランド 392
USING オペランド 392
URIMAP 属性
URIMAP 定義 327
URIMAP 定義
ANALYZER 属性 314
ATOMSERVICE 属性 314
CERTIFICATE 属性 315
CHARACTERSET 属性 315
CIPHERS 属性 315
CONVERTER 属性 316
HFSFILE 属性 317
HOST 属性 318
HOSTCODEPAGE 属性 319
LOCATION 属性 320
MEDIATYPE 属性 320
PATH 属性 321
PIPELINE 属性 322
PORT 属性 322
PROGRAM 属性 323
REDIRECTTYPE 属性 323
SCHEME 属性 324
SOCKETCLOSE 属性 324
STATUS 属性 324
TCPIPSERVICE 属性 325
TEMPLATENAME 属性 325
TRANSACTION 属性 326
URIMAP 属性 327
USAGE 属性 327
USERID 属性 328
WEBSERVICE 属性 329
URIMAP リソース 307
URM 属性
TCPIPSERVICE 定義 173
USAGE 属性
URIMAP 定義 327
USEDFLTUSER 属性
CONNECTION 定義 22
TERMINAL 定義 232
USERAREALEN 属性
SESSIONS 定義 171
TYPETERM 定義 306
USERAUTH 属性
IPCONN 定義 78
USERDEFINE コマンド
CEDA 365
USERDEFINE コマンド、DFHCSDUP ユーティリティー・プログラム 392
USERID オペランド
順次装置 467
DFHTCT TYPE=TERMINAL 467
USERID 属性
SESSIONS 定義 171
TDQUEUE 定義 202
TERMINAL 定義 232
URIMAP 定義 328

V

VALIDATION 属性
TYPETERM 定義 307
VARIABLE オプション
RECFORM オペランド 411
VERIFY オペランド
DFHFCT TYPE=FILE 413
VERIFY コマンド、DFHCSDUP ユーティリティー・プログラム 394
VERTICALFORM 属性
TYPETERM 定義 307
VIEW コマンド
CEDA 369
VTAM オプション
ACCMETH オペランド 452

W

WAIT 属性
TDQUEUE 定義 203
TRANSACTION 定義 254
WAITACTION 属性
TDQUEUE 定義 203
WAITTIME 属性
TRANSACTION 定義 256
WEBSERVICE 属性
URIMAP 定義 329
WEBSERVICE リソース 330

WSDIR 属性
  PIPELINE 定義 133
WTOL オペランド
  DFHCLT TYPE=WTO 402

X

XLNACTION 属性
  CONNECTION 定義 22
  IPCONN 定義 78
XLT (トランザクション・リスト・テーブル) 485
  コーディング例 488
  制御セクション 486
  DFHXLT TYPE=INITIAL 486
XPREFIX 属性
  TSMODEL 定義 262
XREMOTEPFX 属性
  TSMODEL 定義 262
XTPNAME 属性
  PARTNER 定義 129
  TRANSACTION 定義 257
XTRANID 属性
  TRANSACTION 定義 257

Z

z/OS Communications Server
  リモート・システムのアプリケーション名 17

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