

CICS Transaction Server for z/OS



リリース・ガイド

バージョン 3 リリース 1

CICS Transaction Server for z/OS



リリース・ガイド

バージョン 3 リリース 1

お願い

本書および本書で紹介する製品をご使用になる前に、419 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、CICS Transaction Server for z/OS バージョン 3 リリース 1 (プログラム番号 5655-M15)、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。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

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： GC34-6421-01
CICS Transaction Server for z/OS
Release Guide
Version 3 Release 1

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2005.3

この文書では、平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 2004,2005. All rights reserved.

© Copyright IBM Japan 2005

目次

まえがき	xi
本書について	xi
本書の対象読者	xi
本書を理解するうえで必要な知識	xi
用語についての注意	xi
構文表記法	xii

第 1 部 CICS Transaction Server for z/OS バージョン 3 リリース 1 の要約 . . . 1

第 1 章 CICS Transaction Server for z/OS バージョン 3 リリース 1 の要約	3
CICS 統合	3
アプリケーション変換	5
エンタープライズ・マネージメント	6
その他の変更点	7
廃止された機能	7
CICS Information Center	8

第 2 部 CICS 統合 . . . 11

第 2 章 CICS における Web サービス	13
ビジネスで Web サービスを利用することの利点	13
Web サービスの用語	13
要件	15
CICS が Web サービスをサポートする仕組み	16
メッセージ・ハンドラーおよびパイプライン	16
SOAP メッセージおよびアプリケーション・データ構造	22
WSDL およびアプリケーション・データ構造	24
Web サービスのバインディング・ファイル	26
CICS での Web サービス使用の計画	27
サービス・プロバイダー・アプリケーションの計画	28
サービス・リクエスター・アプリケーションの計画	30
CICS Web サービス・アシスタント	31
DFHLS2WS: 高水準言語から WSDL への変換	33
DFHWS2LS: WSDL から高水準言語への変換	37
CICS 外部インターフェースの変更点	43
リソース定義の変更点	43
アプリケーション・プログラミング・インターフェースの変更点	49
システム・プログラミング・インターフェースの変更点	54
CEMT の変更点	61
JCICS API の変更点	65
CICS 提供トランザクションの変更点	66
統計の変更点	66
サンプル・プログラムの変更点	66
CICS ユーティリティーの変更点	66
問題判別の変更点	67
セキュリティ	67
マイグレーション	67
既存の機能のマイグレーション	67
共存	69

CICSplex SM サポート	69
CICSplex SM アプリケーション・プログラミング・インターフェースの変更点	69
CICSplex SM Web ユーザー・インターフェースの変更点	71
第 3 章 CICS アプリケーションからの HTTP クライアント要求のサポート	75
CICS アプリケーションからの HTTP クライアント要求のサポートの利点	75
要件	75
HTTP クライアントとしての CICS に対する HTTP 要求と応答の処理	75
セッション・トークン	77
CICS 外部インターフェースの変更点	78
リソース定義の変更点	78
アプリケーション・プログラミング・インターフェースの変更点 (HTTP クライアント要求)	78
JCICS API の変更点	101
グローバル・ユーザー出口ルーチンの変更点	101
モニターの変更点	104
第 4 章 HTTP/1.1 への CICS Web サポートのアップグレード	107
HTTP/1.1 への CICS Web サポートのアップグレードの利点	108
要件	109
新しい HTTP 機能	109
チャンク形式転送コーディング	109
パイプライン化	111
持続接続	112
仮想ホスティング	113
CICS 外部インターフェースの変更点	114
リソース定義の変更点	114
アプリケーション・プログラミング・インターフェース (HTTP/1.1 サポート) の変更点	116
システム・プログラミング・インターフェースの変更点	125
CEMT の変更点	128
統計の変更点	128
第 5 章 CICS Web サポートの一般拡張	129
CICS Web サポート拡張の利点	130
用語	130
要件	131
HTTP サーバーとしての CICS に対する HTTP 要求と応答の処理	132
CICS Web サポートでの Unicode UTF-8 および UTF-16 コード・ページ変換	133
HTTP 日時スタンプ・フォーマットの処理	133
CICS 外部インターフェースの変更点	134
リソース定義の変更点	134
アプリケーション・プログラミング・インターフェースの変更点 (汎用 CICS Web サポート拡張)	145
システム・プログラミング・インターフェースの変更点	153
CEMT の変更点	159
CICS 提供トランザクションの変更点	159
ユーザーが置換可能なプログラムの変更点	161
統計の変更点	164
CICS ユーティリティーの変更点	164
問題判別の変更点	165

セキュリティ	165
マイグレーション	166
既存の CICS Web サポート・アプリケーションのマイグレーション	166
新規 CICS Web サポート機能へのマイグレーション	168
CICSplex SM サポート	169
CICSplex SM エンド・ユーザー・インターフェース・ビューの変更点	169
CICSplex SM アプリケーション・プログラミング・インターフェースの変更点	170
CICSplex SM Web ユーザー・インターフェースの変更点	175
第 6 章 インターネット・セキュリティの改善	179
インターネット・セキュリティの改善の利点	179
セキュリティ用語	179
要件	180
Transport Layer Security プロトコル	181
SSL パフォーマンスの向上	182
証明書取り消しリストの使用	182
SSL キャッシュ	183
暗号化ネゴシエーションのカスタマイズ	183
CICS 外部インターフェースの変更点	184
システム初期設定パラメーターの変更点	184
リソース定義の変更点	186
システム・プログラミング・インターフェースの変更点	187
CICS 提供トランザクションの変更点	187
グローバル・ユーザー出口ルーチンの変更点	189
モニターの変更点	189
統計の変更点	189
CICS ユーティリティの変更点	190
問題判別の変更点	190
セキュリティ	190
マイグレーション	190
既存の機能のマイグレーション	190
新規機能へのマイグレーション	190
共存	191
CICSplex SM サポート	191
CICSplex SM エンド・ユーザー・インターフェース・ビューの変更点	191
CICSplex SM アプリケーション・プログラミング・インターフェースの変更点	192
CICSplex SM Web ユーザー・インターフェースの変更点	193

第 3 部 アプリケーション変換 197

第 7 章 プログラム間データ転送の拡張: 最新の COMMAREA としてのチャネル	
ネル	199
チャネルの利点	199
用語	200
チャネル: クイック・スタート	201
コンテナおよびチャネル	201
基本的な例	201
チャネルの使用: いくつかの典型的シナリオ	204
1 つのチャネルと 1 つのプログラム	204
1 つのチャネルと複数のプログラム (コンポーネント)	205

複数のチャンネルと 1 つのコンポーネント	205
複数の対話式コンポーネント	206
チャンネルの作成	207
現行チャンネル	208
現行チャンネルの例 (LINK コマンドの場合)	208
現行チャンネルの例 (XCTL コマンドの場合)	211
現行チャンネル: START コマンドおよび RETURN コマンド	212
チャンネルのスコープ	213
スコープの例 (LINK コマンドの場合)	213
スコープの例 (LINK および XCTL コマンドの場合)	215
プログラムに渡されたコンテナの発見	217
リンクから戻されたコンテナの発見	217
CICS 読み取り専用コンテナ	218
チャンネルの設計: ベスト・プラクティス	218
チャンネルの生成と使用: 例	219
チャンネルと BTS アクティビティ	221
コンテキスト	222
JCICS からのチャンネルの使用	223
JCICS でのチャンネルおよびコンテナの作成	224
コンテナへのデータの書き込み	224
別のプログラムまたはタスクへのチャンネルの引き渡し	224
現行チャンネルの受取	225
コンテナからのデータの取得	225
現行チャンネルのブラウズ	225
JCICS の例	225
チャンネルを使用した動的ルーティング	226
データ変換	227
データ変換が必要な理由	227
チャンネルを使用したデータ変換	227
要件	231
CICS 外部インターフェースの変更点	231
アプリケーション・プログラミング・インターフェースの変更点	231
JCICS API の変更点	247
グローバル・ユーザー出口ルーチンの変更点	247
タスク関連ユーザー出口の変更点	248
ユーザーが置換可能なプログラムの変更点	248
モニターの変更点	250
統計の変更点	252
サンプル・プログラムの変更点	253
問題判別の変更点	253
COMMAREA からチャンネルへのマイグレーション	257
既存の機能のマイグレーション	257
新規機能へのマイグレーション	257
共存	260
CICSPlex SM サポート	261
CICSPlex SM アプリケーション・プログラミング・インターフェースの変更点	261
CICSPlex SM Web ユーザー・インターフェースの変更点	263
第 8 章 OPENAPI サポート	265
OPENAPI サポート の利点	267
要件	267

CICS 外部インターフェースの変更点	267
システム初期設定パラメーターの変更点	267
リソース定義の変更点	267
アプリケーション・プログラミング・インターフェースの変更点	268
システム・プログラミング・インターフェースの変更点	269
CEMT の変更点	270
第 9 章 XPLink サポート	273
XPLink サポート の利点	274
要件	275
CICS 外部インターフェースの変更点	275
インストールの変更点	275
システム初期設定パラメーターの変更点	275
リソース定義の変更点	276
アプリケーション・プログラミング・インターフェースの変更点	276
システム・プログラミング・インターフェースの変更点	277
CEMT の変更点	278
グローバル・ユーザー出口ルーチンの変更点	279
ユーザーが置換可能なプログラムの変更点	279
モニターの変更点	279
統計の変更点	281
マイグレーション	282
既存の機能のマイグレーション	282
新規機能へのマイグレーション	282
CICSplex SM サポート	282
CICSplex SM アプリケーション・プログラミング・インターフェースの変更点	283
CICSplex SM Web ユーザー・インターフェースの変更点	284
第 10 章 言語環境プログラムに準拠したアセンブラー MAIN プログラムのサポート	285
言語環境プログラムに準拠したアセンブラー MAIN プログラムのサポートの利点	285
要件	285
CICS 外部インターフェースの変更点	285
リソース定義の変更点	285
アプリケーション・プログラミング・インターフェースの変更点	286
グローバル・ユーザー出口ルーチンの変更点	295
タスク関連ユーザー出口の変更点	296

第 4 部 エンタープライズ・マネージメント 297

第 11 章 CICSplex SM Web ユーザー・インターフェースの拡張	299
CICSplex SM Web ユーザー・インターフェースの拡張の利点	300
要件	301
CICSplex SM の変更点	301
ユーザーのお気に入り	301
ユーザー・グループ・プロファイル	306
ビジネス・アプリケーション・サービスの設計変更	310
レコード・カウントの警告	313
フィルター確認	315
動的選択リスト	316

画面設計の改良	318
CICSplex SM API の変更点	323
メッセージ	324
第 12 章 CICSplex SM バッチ・リポジトリ更新機能の拡張	325
CICSplex SM バッチ・リポジトリ更新機能の拡張の利点	325
要件	325
バッチ・ユーティリティー・プログラム	325
CICSplex SM アプリケーション・プログラミング・インターフェースの変更点	326
CICSplex SM Web ユーザー・インターフェースの変更点	326
メッセージ	327
セキュリティ	327
マイグレーション	327

第 5 部 その他の変更点 329

第 13 章 新規インストール・プロセス	331
新規インストール・プロセスの利点	331
第 14 章 EXTRACT STATISTICS コマンド	333
EXTRACT STATISTICS コマンドの利点	333
CICS 外部インターフェースの変更点	333
システム・プログラミング・インターフェースの変更点	333
第 15 章 大/小文字混合パスワードのサポート	337
第 16 章 コード・ページ変換の変更点	339
コード・ページ変換の変更点の利点	339
用語	340
要件	340
CICS 外部インターフェースの変更点	340
インストールの変更点	340
システム初期設定パラメーターの変更点	341
アプリケーション・プログラミングの変更点	341
CICS ユーティリティーの変更点	343
問題判別の変更点	343
第 17 章 デフォルト・コード・ページの定義の単純化	345
改良されたデータ変換テンプレートへのコード・ページのデフォルト設定の利点	345
要件	346
CICS 外部インターフェースの変更点	346
システム初期設定パラメーターの変更点	346
ユーザーが置換可能なプログラムの変更点	346
第 18 章 64 ビット・アドレッシング容認の変更点	349
64 ビット・アドレッシング容認の変更点の利点	349
要件	349
CICS 外部インターフェースの変更点	349
CICS ユーティリティーの変更点	349
問題判別の変更点	349
第 19 章 取り消されたユーザー ID のサポート	351

第 6 部 廃止された機能	353
第 20 章 OS/VS COBOL プログラムのランタイム・サポートの中止	355
第 21 章 BTAM および TCAM サポートに対する変更点	357
BTAM サポートの中止	357
TCAM に対する CICS サポートの変更点	357
CICS 外部インターフェースの変更点	358
システム初期設定パラメーターの変更点	358
リソース定義の変更点	358
アプリケーション・プログラミング・インターフェースの変更点	358
グローバル・ユーザー出口ルーチンの変更点	359
ユーザーが置換可能なプログラムの変更点	359
サンプル・プログラムの変更点	359
マイグレーション	359
共存	360
第 22 章 1 バイトのコンソール ID のサポートの中止	361
CICSplex SM サポート	361
CICSplex SM アプリケーション・プログラミング・インターフェースの変更 点	361
CICSplex SM エンド・ユーザー・インターフェース・ビューの変更点	361
CICSplex SM Web ユーザー・インターフェースの変更点	361
メッセージ	362
第 23 章 CICS Connector for CICS TS の中止	363
第 24 章 Java プログラム・オブジェクトおよびホット・プーリングのランタ イム・サポートの中止	365
CICS 外部インターフェースの変更点	366
システム初期設定パラメーターの変更点	366
リソース定義の変更点	366
アプリケーション・プログラミング・インターフェースの変更点	366
システム・プログラミング・インターフェースの変更点	366
CEMT の変更点	367
グローバル・ユーザー出口ルーチンの変更点	367
出口プログラミング・インターフェース (XPI) の変更点	367
ユーザーが置換可能なプログラムの変更点	367
モニターの変更点	367
統計の変更点	367
問題判別の変更点	367
CICSplex SM サポート	368
CICSplex SM エンド・ユーザー・インターフェース・ビューの変更点	368
CICSplex SM アプリケーション・プログラミング・インターフェースの変更 点	368
CICSplex SM Web ユーザー・インターフェースの変更点	368
第 25 章 Windows リモート MAS のための CICSplex SM サポートの中止	371
第 26 章 CICS Transaction Affinities ユーティリティの提供中止	373
第 7 部 一般情報	375

第 27 章 CICS 稼働環境	377
ハードウェア要件	377
ソフトウェア要件	378
CICS ツールおよび関連製品のサポート	380
互換性	381
第 28 章 スレッド・セーフ・アプリケーション・プログラミング・インター フェイス・コマンド	383
第 29 章 高水準言語サポート	385

第 8 部 資料 391

第 30 章 Eclipse インフォメーション・センター	393
Eclipse インフォメーション・センターの利点	393
用語	394
要件	395
新機能セクション	395
情報ロードマップ	396
ラーニング・パス	396
インフォメーション・センター内での検索の手法	397
インフォメーション・センターのナビゲート	399
トピックのブックマーク	400
ユーザー設定	400
第 31 章 CICS Transaction Server for z/OS ライブラリー	401
ハードコピーで入手できる資料	401
PDF のみで提供される資料	401
CICS Transaction Server for z/OS の CICS 資料	402
CICS Transaction Server for z/OS の CICSplex SM 資料	403
CICS ファミリー資料	403
ライセンス出版物	403
アクセシビリティ	405
索引	407
特記事項	419
商標	420

まえがき

本書について

本書は、CICS® Transaction Server for z/OS® バージョン 3 リリース 1 の新機能および変更機能についての情報を記載しています。また、参照情報についての変更の概要を示し、さらに詳しい参照情報を記載した資料を挙げています。

本書に記載するプログラミング・インターフェース情報は、前のリリースの CICS TS に加えられた新機能と変更内容のみを示し、新機能の利点を強調することを目的としています。プログラミング・インターフェース情報については、以下の資料のプログラミング・インターフェースとその関連情報の主要な説明箇所を参照してください。

- *CICS Application Programming Reference*
- *CICS System Programming Reference*
- *CICS Customization Guide*
- *CICS 外部インターフェース・ガイド*
- *CICSplex SM Application Programming Guide*
- *CICSplex SM Application Programming Reference*

本書の対象読者

本書は、以下のユーザーの作業の担当者を対象としています。

- 評価および計画
- システム管理
- プログラミング
- カスタマイズ

本書を理解するうえで必要な知識

本書は、システム管理者として、あるいはシステム・プログラマーまたはアプリケーション・プログラマーの立場で、CICS および CICSplex SM をよく理解していることが前提となっています。

用語についての注意

本書で、修飾なしで使用する『CICS』という用語は、IBM CICS TS の CICS エレメントを意味します。

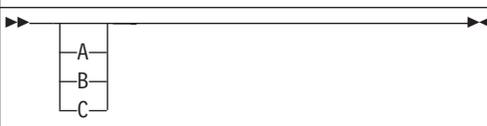
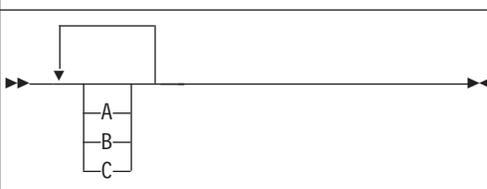
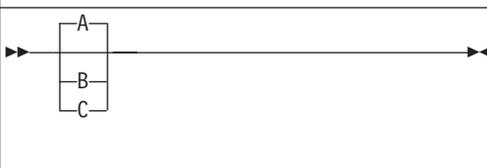
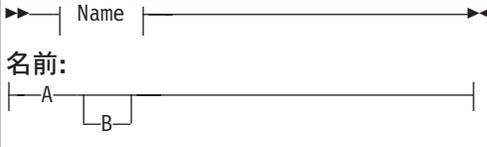
『CICSplex SM』は、IBM CICS TS の CICSplex System Manager エレメントを表します。

『MVS』は、z/OS の基本エレメントであるオペレーティング・システムを表します。

構文表記法

構文表記法は、CICS コマンド、リソース定義、およびその他の多数の箇所では指定可能なオプションまたは属性の、許容される組み合わせを指定します。

構文表記法で 사용되는規則は以下のとおりです。

表記	説明
	必須の選択肢のセットを表します。示された値のいずれか (1 つのみ) を指定する必要があります。
	必須の選択肢のセットを表します。示された値の少なくとも 1 つを指定する必要があります。この中から複数指定することができ、順序は問いません。
	オプションの選択肢のセットを表します。示された値を何も指定しないか、または 1 つ指定することができます。
	オプションの選択肢のセットを表します。示された値を何も指定しないか、1 つあるいは複数指定することができ、順序は問いません。
	オプションの選択肢のセットを表します。示された値を何も指定しないか、または 1 つ指定することができます。A は、何も指定しなかった場合に使用されるデフォルト値です。
	構文表記法の指定されたセクションへの参照。
	A= は、示されたとおりに正確に入力しなければならない文字を表します。 <i>value</i> は、適切な値を指定する必要がある変数を示します。

第 1 部 CICS Transaction Server for z/OS バージョン 3 リリース 1 の要約

第 1 部では、CICS Transaction Server for z/OS バージョン 3 リリース 1 の主な新機能の概要について説明します。

第 1 章 CICS Transaction Server for z/OS バージョン 3 リリース 1 の要約

CICS Transaction Server for z/OS バージョン 3 リリース 1 の主な新機能および変更機能を、CICS 統合、アプリケーション変換、およびエンタープライズ・マネジメントという 3 つのテーマに分けて示します。CICS TS 3.1 には、この 3 つのテーマの範囲外のさまざまな変更も含まれており、いくつかの機能は廃止、または有効範囲が狭められました。

CICS 統合

CICS TS 3.1 は、広く採用されているアプリケーション・プログラミング・インターフェースおよび標準プロトコルの使用によって、既存の CICS アプリケーションをより広範なオンデマンド・シナリオ内で再利用できるようにするさまざまな新機能および拡張機能を提供します。該当するものを以下に示します。

Web サービスのサポート

Web サービスのサポートは、CICS TS 3.1 に完全に統合されました。このリリースでは、CICS アプリケーションはサービス・プロバイダーとサービス・リクエスターの両方の役割で動作できます。この場合、サービスは Web サービス記述言語 (WSDL) を使って定義されます。

CICS TS V3.1 の一部として提供されるインフラストラクチャーには、WS-AtomicTransaction 仕様と互換性のある分散トランザクション調整機能が含まれています。

Web サービスのサポートには、CICS Web サービス・アシスタントが含まれます。これは、以下の処理に役立つバッチ・ユーティリティです。

- 既存の CICS アプリケーションを Web サービスに変換する
- 外部プロバイダーによって提供される Web サービスを CICS アプリケーションが使用できるようにする

このアシスタントでは、単純な言語構造から WSDL 文書を作成したり、既存の WSDL 文書から言語構造を作成したりすることができ、COBOL、C/C++、および PL/I をサポートします。また、SOAP メッセージからコンテナおよび COMMAREA への自動ランタイム変換、さらに、この逆の変換を可能にするために使用される情報も生成します。

詳細については、13 ページの『第 2 章 CICS における Web サービス』を参照してください。

CICS アプリケーションからの HTTP クライアント要求のサポート

CICS TS 3.1 では、HTTP クライアントとして動作する CICS の機能は、CICS Web サポートに完全に統合されました。これによりアプリケーション・プログラムは、EXEC CICS コマンドを使用してサーバーへの接続のオープン、HTTP 要求の実行、および応答の受信ができるようになりました。

詳細については、75 ページの『第 3 章 CICS アプリケーションからの HTTP クライアント要求のサポート』を参照してください。

HTTP/1.1 のサポート

CICS Web サポートは、RFC 2616、*Hypertext Transfer Protocol -- HTTP/1.1* で定義されている HTTP/1.1 仕様に条件付きで準拠しています。

CICS TS 3.1 でサポートされる新機能としては、チャンク形式転送コーディング、パイプライン化、および持続接続があります。

詳細については、107 ページの『第 4 章 HTTP/1.1 への CICS Web サポートのアップグレード』を参照してください。

CICS Web サポートのその他の拡張

CICS TS 3.1 における CICS Web サポートのその他の拡張としては、以下のものがあります。

- URIMAP リソース。これにより、CICS が HTTP サーバーのときの HTTP 要求および応答の処理機能が向上します。
- CICS Web でのコード・ページのサポート方法の改善 (UTF-8 および UTF-16 コード・ページのサポートを含む)
- HTTP 日時形式をサポートする新規 API コマンド。

詳細については、129 ページの『第 5 章 CICS Web サポートの一般拡張』を参照してください。

SSL サポートの改善

CICS TS 3.1 では、Secure Sockets Layer (SSL) のサポートが拡張されました。

- CICS は SSL 3.0 に加えて Transport Layer Security (TLS) 1.0 プロトコルをサポートするようになったため、128 ビット暗号化および 256 ビット暗号化を提供する新しい AES 暗号スイートを使用できるようになりました。
- システムで同時に使用できる同時 SSL 接続の数が増加したため、スループットが改善されています。
- クライアントとサーバーの間での暗号化ネゴシエーションの制御における柔軟性が増しました。これにより、特定ユーザーとのネゴシエーションに関して、CICS で最小および最大の暗号化レベルを指定できるようになりました。
- CICS は、クライアントとのネゴシエーション時に証明書取り消しリストと照らし合わせてすべての証明書を検査できるようになりました。取り消された証明書を使用している接続は、即時にクローズされます。
- SSL キャッシュを使用してスプレックス全体でセッション ID を共有するかどうかを指定できます。クライアントが以前に CICS とネゴシエーションしたことがある場合、CICS は部分的な SSL ハンドシェイクを実行します。これにより、複数の CICS 領域間でキャッシュを共有する場合に、SSL ネゴシエーションおよび接続スループットのパフォーマンスが改善されます。

詳細については、179 ページの『第 6 章 インターネット・セキュリティーの改善』を参照してください。

アプリケーション変換

CICS Transaction Server for z/OS バージョン 3 リリース 1 では、最新のプログラム言語、構成、およびツールを使用して、既存のアプリケーションを拡張したり、新規アプリケーションを構成したりできるさまざまな新機能が提供されます。該当するものを以下に示します。

プログラム間データ転送の拡張: 最新の COMMAREA としてのチャンネル

CICS TS 3.1 では、チャンネルおよびコンテナを使用することによってプログラム間のデータ転送方法が改善されました。これにより、COMMAREA に適用される 32KB の制限をはるかに超える量で転送できます。

- コンテナとは、プログラム間で情報を受け渡すために作られた名前付きデータ・ブロックです。
- コンテナは、チャンネルと呼ばれるセットとしてグループ化されます。チャンネルは、CICS プログラム間でデータを交換するための標準のメカニズムであり、パラメーター・リストと類似しています。チャンネルは、LINK、START、XCTL、RETURN コマンド上、およびローカル・トランザクションとリモート・トランザクションで使用することができます。チャンネルに追加できるコンテナの数に制限はなく、各コンテナのサイズは、利用可能なストレージの量によってのみ制限されます。

チャンネルおよびコンテナ・モデルは、従来の COMMAREA モデルに比べいくつかの利点があります。該当するものを以下に示します。

- 32KB のサイズ制限がない。
- 一体構造のデータ・ブロックである COMMAREA とは異なり、いくつかのコンテナから成るチャンネルが構造化されたデータを表現することができます。

詳細については、199 ページの『第 7 章 プログラム間データ転送の拡張: 最新の COMMAREA としてのチャンネル』を参照してください。

OPENAPI サポート

CICS では、OPENAPI アプリケーション・プログラムのサポートを提供することによって、オープン・トランザクション環境 (OTE) 機能の使用が拡張されました。これまでは、OPENAPI 機能はタスク関連ユーザー出口 (TRUE) にのみ使用可能でした。OPENAPI プログラムを使用すれば、アプリケーションのワークロードを QR TCB から複数のオープン TCB に移動できます。

詳細については、265 ページの『第 8 章 OPENAPI サポート』を参照してください。

C および C++ への XPLINK サポート

C および C++ の新規サポートが CICS TS 3.1 に導入されました。これにより、これらの言語で書かれたアプリケーションのパフォーマンスは、COBOL、PL/I、またはアセンブラで得られるパフォーマンスに匹敵するレベルになります。パフォーマンスの向上は、z/OS の Extra Performance Linkage (XPLINK) 機能によって提供されます。この機能は、高性能サブルーチン・リンケージ・メカニズムや、スタック拡張のための保護ページを提供するので、実行パスの長さが高度に最適化されます。

これらの改良の恩恵を最も受けるのは、次のようなアプリケーションです。

- CICS オープン・トランザクション環境 (OTE) で稼働するアプリケーション
- スレッド・セーフ標準に従って作成されていて、スレッド・セーフ CICS コマンドのみを使用するアプリケーション

XPLINK を使用すると、C および C++ で組み込まれた最新のコンパイラおよび最適化テクノロジーを活用できます。特に、XPLINK オプションを使ってコンパイルされたダイナミック・ロード・ライブラリー (DLL) を CICS 環境で使用することによって、C および C++ コードの再利用性を高めることができます。

詳細については、199 ページの『第 7 章 プログラム間データ転送の拡張: 最新の COMMAREA としてのチャンネル』を参照してください。

言語環境プログラム対応アセンブラー・メインプログラムのサポート

CICS TS 3.1 は、アセンブラーで書かれ、言語環境プログラム対応のメインプログラムをサポートします。これにより、言語環境プログラムの使用可能性が拡張され、アセンブラー・プログラムでデバッガー・サポートが使用可能になります。

詳細については、285 ページの『第 10 章 言語環境プログラムに準拠したアセンブラー MAIN プログラムのサポート』を参照してください。

エンタープライズ・マネージメント

CICS TS 3.1 の CICSplex[®] SM エLEMENT は、最新のユーザー・インターフェースの使用によって大規模なランタイム構成を効率的に管理できる新機能を提供するので、要求の厳しいサービス・レベル目標を満たすことができます。該当するものを以下に示します。

CICSplex SM Web ユーザー・インターフェースの拡張

CICSplex SM Web ユーザー・インターフェースが改良されて、より強力かつ便利になりました。Web ユーザー・インターフェースは、CICSplex SM TSO エンド・ユーザー・インターフェースと機能的に同等となり、CICSplex SM にアクセスするための基本方式となりました。

詳細については、299 ページの『第 11 章 CICSplex SM Web ユーザー・インターフェースの拡張』を参照してください。

CICSplex SM バッチ・リポジトリ更新機能の拡張

指定された CICSplex SM リポジトリにバッチ更新を実行依頼するために、既存の TSO EUI コマンドに対する代替を提供する新機能が導入されました。導入された代替 BATCHREP 機能は、以下のとおりです。

- CPSM API によって参照されるオブジェクトとして使用可能な新規 BATCHREP リソース・テーブル
- 新規リソース・テーブルの Web ユーザー・インターフェース・サポート
- BATCHREP 機能を提供するバッチ・ユーティリティー・プログラム

詳細については、325 ページの『第 12 章 CICSplex SM バッチ・リポジトリ更新機能の拡張』を参照してください。

その他の変更点

CICS Transaction Server for z/OS バージョン 3 リリース 1 でのその他の新機能および変更機能としては、以下のものがあります。

EXTRACT STATISTICS コマンド

新規 SPI コマンド EXTRACT STATISTICS は、URIMAP、PIPELINE、および WEBSERVICE リソースに関して COLLECT STATISTICS と同等の機能を実行します。

詳細については、333 ページの『第 14 章 EXTRACT STATISTICS コマンド』を参照してください。

大/小文字混合パスワードのサポート

CICS で使用されるセキュリティー・マネージャーが大/小文字混合のパスワードの使用をサポートしている場合、CICS TS 3.1 はパスワードを大文字に変換せずにセキュリティー・マネージャーに渡します。

詳細については、337 ページの『第 15 章 大/小文字混合パスワードのサポート』を参照してください。

デフォルト・コード・ページの定義の単純化

CICS TS 3.1 では、DFHCNV データ変換テーブルで使用されるクライアントまたはサーバーのデフォルト・コード・ページは、システム初期設定パラメーターで定義できます。

詳細については、345 ページの『第 17 章 デフォルト・コード・ページの定義の単純化』を参照してください。

64 ビット・アドレッシングの容認

CICS では 64 ビット・アドレッシングの実行はサポートされませんが、プログラムは、CICS が 64 ビット・アーキテクチャーのマシン上で稼働する場合にのみ使用可能なアドレスでストレージを使用できます。CICS 異常終了収集メカニズムは、フル 64 ビット汎用レジスターの内容が収集されるように変更されました

詳細については、349 ページの『第 18 章 64 ビット・アドレッシング容認の変更点』を参照してください。

取り消されたユーザー ID のサポート

EXEC CICS VERIFY PASSWORD コマンドが発行されたときに、CICS はユーザー ID またはユーザーのグループ接続の取り消し状況を受け入れるようになりました。

廃止された機能

CICS Transaction Server for z/OS バージョン 2 でサポートされていた一部の機能は、CICS TS 3.1 では廃止、または有効範囲が狭められました。該当するものを以下に示します。

OS/VS COBOL プログラムのランタイム・サポート

CICS Transaction Server for z/OS バージョン 2 でランタイム・サポートが提供されていた OS/VS COBOL プログラムは、CICS TS 3.1 では実行できません。

詳細については、355 ページの『第 20 章 OS/VS COBOL プログラムのランタイム・サポートの中止』を参照してください。

BTAM および TCAM のサポート

CICS TS 3.1 では、CICS での基本通信アクセス方式 (BTAM) のサポートは中止されました。通信アクセス方式 (TCAM) のサポートは、DCB インターフェースの間接的なサポートに限られます。

詳細については、357 ページの『第 21 章 BTAM および TCAM サポートに対する変更点』を参照してください。

1 バイトのコンソール ID のサポート

1 バイトのコンソール ID のサポートは除去されました。コンソールは、TERMINAL 定義の CONSNAME(name) 属性を使用して定義することができます。

詳細については、361 ページの『第 22 章 1 バイトのコンソール ID のサポートの中止』を参照してください。

CICS connector for CICS TS

CICS Transaction Server for z/OS バージョン 2 リリース 1 で導入された、CICS Connector for CICS TS に対するサポートは中止されました。

詳細については、363 ページの『第 23 章 CICS Connector for CICS TS の中止』を参照してください。

Java™ プログラム・オブジェクトおよびホット・プーリングのサポート

Java プログラム・オブジェクトおよびホット・プーリング (HPJ) のランタイム・サポートは中止されました。

詳細については、365 ページの『第 24 章 Java プログラム・オブジェクトおよびホット・プーリングのランタイム・サポートの中止』を参照してください。

CICSplex SM での Windows® リモート MAS のサポート

詳細については、371 ページの『第 25 章 Windows リモート MAS のための CICSplex SM サポートの中止』を参照してください。

CICS Transaction Affinities ユーティリティーの提供中止

CICS Transaction Affinities ユーティリティーは、CICS では提供されなくなりました。このユーティリティーのトランザクションの親和性を検出する機能は、より高度なツールである CICS Interdependency Analyzer によって提供されるようになりました。

詳細については、373 ページの『第 26 章 CICS Transaction Affinities ユーティリティーの提供中止』を参照してください。

CICS Information Center

CICS TS 3.1 では、CICS Information Center は WebSphere® Studio WorkBench User Assistance システム (インフォメーション・センターを構成する複数の文書プラグインを含む Eclipse フレームワーク の 1 つ) 内で実行されます。インフォメーション・センターの新しいルック・アンド・フィール、特に、ウェルカム・ページは、他の製品のインフォメーション・センターと整合性を持つようになりました。

詳細については、393 ページの『第 30 章 Eclipse インフォメーション・センター』を参照してください。

第 2 部 CICS 統合

CICS Transaction Server for z/OS バージョン 3 リリース 1 では、広く採用されているアプリケーション・プログラミング・インターフェースや標準のプロトコルを導入することによって、より広範なオンデマンド・シナリオにおいて、既存の CICS アプリケーションの再利用を可能にする一連の新機能および拡張機能を提供します。

第 2 章 CICS における Web サービス

Web サービスのサポートは、CICS Transaction Server for z/OS バージョン 3 リリース 1 に完全に統合されました。このリリースでは、CICS アプリケーションはサービス・プロバイダーとサービス・リクエスターの両方の役割で動作できます。この場合、サービスは Web サービス記述言語 (WSDL) を使って定義されます。

CICS TS V3.1 の一部として提供されるインフラストラクチャーには、WS-AtomicTransaction 仕様と互換性のある分散トランザクション調整機能が含まれています。

Web サービスのサポートには、CICS Web サービス・アシスタントが含まれます。これは、以下の処理に役立つバッチ・ユーティリティです。

- 既存の CICS アプリケーションを Web サービスに変換する
- 外部プロバイダーによって提供される Web サービスを CICS アプリケーションが使用できるようにする

このアシスタントでは、単純な言語構造から WSDL 文書を作成したり、既存の WSDL 文書から言語構造を作成したりすることができ、COBOL、C/C++、および PL/I をサポートします。また、SOAP メッセージからコンテナおよび COMMAREA への自動ランタイム変換、さらに、この逆の変換 を可能にするために使用される情報も生成します。

ビジネスで Web サービスを利用することの利点

Web サービスは、Web を介してビジネス機能の配置、およびビジネス機能へのアクセスを提供するテクノロジーです。Web サービスを利用することにより、アプリケーションの統合をかつてないほど急速、容易、安価に実現できます。

Web サービスは、以下の点でビジネスに役立ちます。

- ビジネス実施コストの削減
- ソリューション配置の高速化の実現
- 新規機会の開拓

これらすべてを達成するための鍵は、HTTP、XML、SOAP、WSDL などの既存および先進の標準を基に作成される共通のプログラム間通信モデルです。

CICS が Web サービスをサポートすることにより、再プログラミングの労力を最小限に抑えながら、既存のアプリケーションを新しい方法で配置することが可能になります。

Web サービスの用語

Extensible Markup Language (XML)

文書マークアップの標準の 1 つ。汎用の構文を使用し、単純で人間が読めるタグによってデータをマークアップする。この規格は、World Wide Web Consortium (W3C) に公認されている。

最初の SOAP 送信側

SOAP メッセージ・パスの開始点で SOAP メッセージを発信する SOAP 送信側。

サービス・プロバイダー

Web サービス機能を提供するソフトウェアの集合。

サービス・プロバイダー・アプリケーション

サービス・プロバイダーで使用されるアプリケーションのこと。通常、サービス・プロバイダー・アプリケーションは、サービス・プロバイダーのビジネス・ロジック・コンポーネントを提供する。

サービス・リクエスター

サービス・プロバイダーから Web サービスを要求する役割を持つソフトウェアの集合。

サービス・リクエスター・アプリケーション

サービス・リクエスターで使用されるアプリケーション。通常、サービス・リクエスター・アプリケーションは、サービス・リクエスターのビジネス・ロジック・コンポーネントを提供する。

Simple Object Access Protocol

SOAP を参照してください。

SOAP 以前は Simple Object Access Protocol の頭字語。非集中の分散環境で情報を交換するための単純なプロトコル。これは、次の 3 つの部分から構成される XML ベースのプロトコルである。

- メッセージの内容およびその処理方法を記述するためのフレームワークを定義するエンベロープ。
- アプリケーション定義のデータ・タイプのインスタンスを表現するための一連のエンコード規則。
- リモート・プロシージャ・コールおよび応答を表現するための規則。

SOAP は、HTTP などの他のプロトコルと併用できる。

SOAP 中間ノード

SOAP 受信側と SOAP 送信側の両方の機能を備え、SOAP メッセージの内部から宛先を指定できる SOAP ノード。SOAP 中間ノードは、自身を宛先とする SOAP ヘッダー・ブロックを処理し、最終の SOAP 受信側に向けて SOAP メッセージを転送する役割を果たす。

SOAP メッセージ・パス

1 つの SOAP メッセージが通過する一連の SOAP ノードのこと。これには、最初の SOAP 送信側、SOAP 中間ノード (存在しないか 1 つ以上)、最終の SOAP 受信側が含まれる。

SOAP ノード

SOAP メッセージ上で動作する処理ロジック。

SOAP 受信側

SOAP メッセージを受信する SOAP ノード。

SOAP 送信側

SOAP メッセージを送信する SOAP ノード。

最終の SOAP 受信側

SOAP メッセージの最終の宛先となる SOAP 受信側のこと。SOAP 本体およびこれを宛先とするすべての SOAP ヘッダー・ブロックの内容を処理する役割を果たす。

UDDI Universal Description, Discovery and Integration

Universal Description, Discovery and Integration

Universal Description, Discovery and Integration (UDDI) とは、Web サービスに関する Web ベースの分散情報レジストリーの仕様のこと。UDDI は、企業が自社提供の Web サービスに関する情報を登録できる仕様の一連の実装形態でもあり、これによって他の企業はこの企業の情報を検索できる。

Web サービス

ネットワークを介した相互に運用可能なマシン間の対話をサポートする目的で設計されたソフトウェア・システムのこと。マシン処理可能な形式 (特に、Web サービス記述言語 (WSDL)) で記述されているインターフェースがある。

Web サービス・バインディング・ファイル

WEBSERVICE リソースと関連付けられているファイル。ここには、入出力メッセージとアプリケーション・データ構造との間でデータをマップするために CICS が使用する情報が格納されている。

Web サービス記述

サービス・プロバイダーが Web サービスをサービス・リクエスターに呼び出すために仕様をやり取りする場合の手段となる XML 文書。Web サービス記述は、Web サービス記述言語 (WSDL) で記述される。

Web サービス記述言語

Web サービスを記述するための XML アプリケーション。サービスによって提供された抽象機能の記述と、サービスの具体的な詳細 (この機能が提供される仕組みや条件など) とを分離する目的で設計された。

WSDL Web サービス記述言語。

XML Extensible Markup Language。

XML ネーム・スペース

URI 参照によって識別される名前の集まりで、エレメント・タイプおよび属性名として XML 文書内で使用される。

XML スキーマ

構造を記述し、他の XML 文書の内容を制約する XML 文書。

XML スキーマ定義言語

XML スキーマを記述するための XML 構文。World Wide Web Consortium (W3C) によって推奨されている。

要件

この機能をサポートするための特別なハードウェア要件およびソフトウェア要件はありません。

関連情報

377 ページの『第 27 章 CICS 稼働環境』

CICS が Web サービスをサポートする仕組み

CICS は、Web サービス環境で CICS アプリケーションを配置するための 2 つの異なる方法をサポートしています。一方の方法では、プログラミングの労力を最小限に抑えながら迅速な配置を実現できます。もう一方の方法では、各ユーザーの特定の必要性に合わせて記述したコードを使用することにより、Web サービス・アプリケーション全体にわたる柔軟性と制御を実現できます。これら 2 つの方法は、1 つ以上のパイプラインと、Web サービスの要求および応答を対象として動作する 1 つ以上のメッセージ・ハンドラー・プログラムで構成されるインフラストラクチャーによって支えられています。

CICS アプリケーションを Web サービス環境で配置した場合に実行される処理は、以下のとおりです。

- CICS Web サービス・アシスタントを使用すると、アプリケーションを配置するために必要なプログラミングの労力を最小限に抑えることができます。

例えば、既存のアプリケーションを Web サービスとして公開する場合は、高水準言語のデータ構造から着手して、Web サービス記述を生成することができます。もう 1 つの方法として、既存の Web サービスと通信する場合は、Web サービス記述から着手し、作成するプログラムに使用可能な高水準言語の構造を生成することができます。

CICS Web サービス・アシスタントは、アプリケーションを配置するために必要な CICS リソースも生成します。さらに、アプリケーションを実行すると、CICS は、出力ではアプリケーション・データを SOAP メッセージに変換し、入力では SOAP メッセージをアプリケーション・データに戻します。

- データの処理を完全に制御するには、独自のコードを記述して、サービス・リクエストとサービス・プロバイダーとの間で、アプリケーション・データとメッセージ・フローとをマップします。

例えば、Web サービス・インフラストラクチャーの範囲内で SOAP 以外のメッセージを使用する場合は、独自のコードを記述して、メッセージ・フォーマットとアプリケーションが使用するフォーマットとを変換します。

どちらの方法を採用する場合でも、独自のメッセージ・ハンドラーを使用して要求メッセージおよび応答メッセージに対する追加の処理を実行できます。または、SOAP メッセージの処理専用設計された CICS 提供のメッセージ・ハンドラーを使用することもできます。

メッセージ・ハンドラーおよびパイプライン

メッセージ・ハンドラーとは、Web サービスの要求および応答について独自の処理を実行できるプログラムのことです。パイプラインとは、順序どおり実行される一連のメッセージ・ハンドラーのことです。

パイプラインの運用には次に示す 2 つの異なる段階があります。

1. 要求段階。CICS がパイプライン内の各ハンドラーを次々と呼び出す段階です。各メッセージ・ハンドラーは、制御を CICS に戻す前に要求を処理できます。
2. この後に応答段階が続きます。この段階でも、CICS は各ハンドラーを次々と呼び出しますが、順序が逆になります。つまり、要求段階で最初に呼び出されるメ

メッセージ・ハンドラーは、応答段階では最後に呼び出されます。各メッセージ・ハンドラーは、この段階のうちに応答を処理できます。

要求の後には、必ずしも応答があるわけではありません。つまり、一部のアプリケーションは、サービス・リクエスターからプロバイダーへの片方向のメッセージ・フローを使用します。このケースでは、処理すべきメッセージはありませんが、応答フェーズの間に各ハンドラーが順に起動されます。

図 1 には、次の 3 つのメッセージ・ハンドラーのパイプラインを示します。

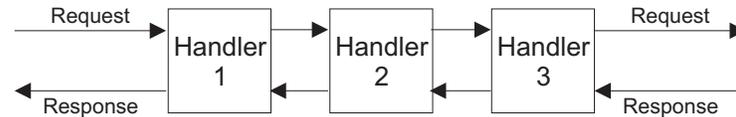


図 1. 一般的な CICS パイプライン

この例では、ハンドラーは次の順序で実行されます。

要求段階の場合

1. ハンドラー 1
2. ハンドラー 2
3. ハンドラー 3

応答段階の場合

1. ハンドラー 3
2. ハンドラー 2
3. ハンドラー 1

サービス・プロバイダーの場合、段階間の移行は、通常、要求を吸収するパイプラインの最後のハンドラー (端末ハンドラー) で実行され、応答が生成されます。サービス・リクエスターの場合、移行が実行されるのは、要求がサービス・プロバイダーで処理される時です。ただし、要求段階のメッセージ・ハンドラーは、応答段階への即時の移行を強制できます。また、CICS によってエラーが検出された場合にも、即時の移行を実行することができます。

メッセージ・ハンドラーは、メッセージを変更することも、変更せずにそのままの状態にしておくこともできます。例えば、次のようになります。

- 暗号化と復号を実行するメッセージ・ハンドラーは、暗号化されたメッセージを入力で受け取り、復号されたメッセージを次のハンドラーに渡します。出力では、逆の処理が行われます。つまり、非暗号化テキスト・メッセージを受け取り、暗号化されたメッセージを次のハンドラーに渡します。
- ロギングを実行するメッセージ・ハンドラーは、メッセージを調べ、関係のある情報をこのメッセージからログにコピーします。次のハンドラーに渡されるメッセージは変更されません。

重要: CICS TS の SOAP 機能に精通している場合は、このリリースの CICS でのパイプラインの構造が SOAP 機能に使用されているものと同じではないことに注意してください。

トランスポート関連ハンドラー

CICS は、Web サービス・リクエスターとプロバイダー間での 2 つのトランスポート機構の使用をサポートしています。使用中のトランスポート機構がどちらであるかによっては、異なるメッセージ・ハンドラーを呼び出すことが必要な場合があります。例えば、HTTP トランスポートを使用して外部ネットワークと通信する場合には、メッセージの一部を暗号化するメッセージ・ハンドラーが必要になります。しかし、機密保護機能のある内部ネットワークで MQ トランスポートを使用する場合、暗号化は必要ありません。

これをサポートするため、パイプラインを構成することにより、特定のトランスポート (HTTP または MQ) が使用中の場合にのみ呼び出されるハンドラーを指定できます。サービス・プロバイダーの場合は、さらに具体的に、特定の指定リソース (HTTP トランスポートの場合は TCPIP SERVICE、MQ トランスポートの場合は QUEUE) が使用中の場合にのみ呼び出されるハンドラーを指定できます。

このことは、図 2 に図示されています。

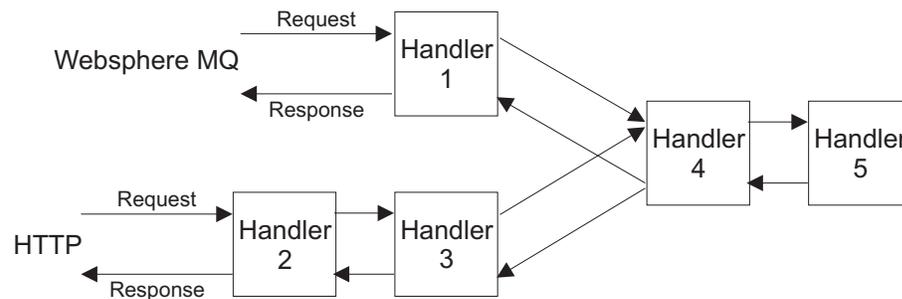


図 2. トランスポート関連ハンドラーを持つパイプライン

この例では、サービス・プロバイダーに適用されるものは次のとおりです。

- ハンドラー 1 は、MQ トランスポートを使用するメッセージの場合に呼び出されます。
- ハンドラー 2 および 3 は、HTTP トランスポートを使用するメッセージの場合に呼び出されます。
- ハンドラー 4 および 5 は、すべてのメッセージを対象として呼び出されます。
- ハンドラー 5 は、端末ノードです。

フローの中断

要求の処理時に、メッセージ・ハンドラーはメッセージを次のハンドラーに渡さない判断をする場合がありますが、応答を生成する場合があります。メッセージの通常の処理は中断され、パイプライン内の一部のハンドラーは呼び出されません。例えば、19 ページの図 3 のハンドラー 2 は、セキュリティー検査を実行する役割を持っています。

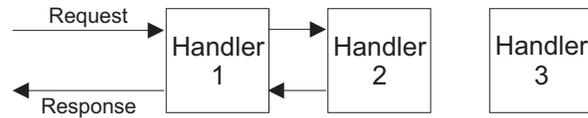


図3. パイプライン・フローの中断

要求に正しいセキュリティー・クリデンシャルが記載されていない場合、ハンドラー 2 は、(要求をハンドラー 3 に渡さずに) 要求を抑止し、適切な応答を作成します。パイプラインは応答段階に到達しているため、ハンドラー 2 が制御を CICS に戻すと、呼び出される次のハンドラーはハンドラー 1 となり、ハンドラー 3 は完全にバイパスされます。

通常のメッセージ・フローをこのように中断するハンドラーは、メッセージの発信元が応答を期待する場合にのみ中断する必要があります。例えば、アプリケーションがサービス・リクエスターからサービス・プロバイダーへの片方向メッセージ・フローを使用する場合、ハンドラーは応答を生成しません。

サービス・プロバイダー・パイプライン

サービス・プロバイダー・パイプラインでは、CICS が要求を受け取ります。この要求はパイプラインを介してターゲット・アプリケーション・プログラムに渡されます。アプリケーションからの応答は、同じパイプラインを介してサービス・リクエスターに戻されます。

CICS がサービス・プロバイダーの役割を果たす場合、CICS は以下の操作を実行します。

1. サービス・リクエスターからの要求を受け取る。
2. 要求を調べ、ターゲット・アプリケーション・プログラムに関係のある内容を抽出する。
3. アプリケーション・プログラムを呼び出し、要求から抽出したデータを渡す。
4. アプリケーション・プログラムが制御を戻したら、アプリケーション・プログラムから戻されたデータを使用して応答を作成する。
5. サービス・リクエスターへ応答を送信する。

図4 には、サービス・プロバイダー設定における次の 3 つのメッセージ・ハンドラーのパイプラインを示します。

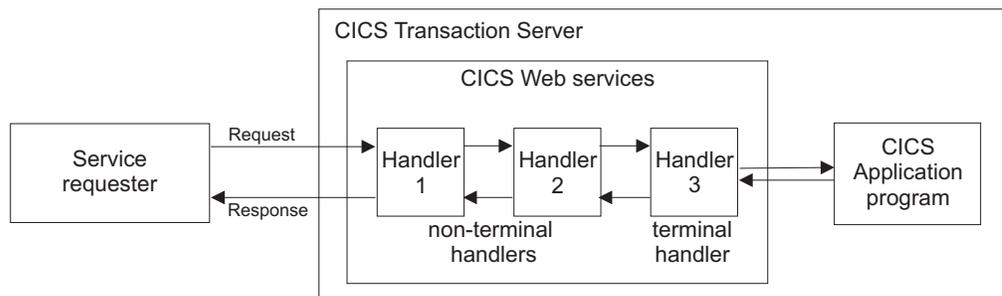


図4. サービス・プロバイダー・パイプライン

1. CICS は、サービス・プロバイダーから要求を受け取ります。次に、この要求をメッセージ・ハンドラー 1 に渡します。
2. メッセージ・ハンドラー 1 は何らかの処理を実行して、要求をハンドラー 2 に渡します (正確には、パイプラインを管理する CICS に制御を戻します。次に CICS は、次のメッセージ・ハンドラーに制御を渡します)。
3. メッセージ・ハンドラー 2 は、ハンドラー 1 から要求を受け取り、何らかの処理を実行して、要求をハンドラー 3 に渡します。
4. メッセージ・ハンドラー 3 は、このパイプラインの端末ハンドラーです。ハンドラー 3 は、要求に記載された情報を使用して、アプリケーション・プログラムを呼び出します。次に、アプリケーション・プログラムからの出力を使用して応答を生成し、この応答をハンドラー 2 に戻します。
5. メッセージ・ハンドラー 2 は、ハンドラー 3 から応答を受け取り、何らかの処理を実行して、応答をハンドラー 1 に渡します。
6. メッセージ・ハンドラー 1 は、ハンドラー 2 から応答を受け取り、何らかの処理を実行して、応答をサービス・リクエスターに戻します。

サービス・リクエスター・パイプライン

サービス・リクエスター・パイプラインでは、アプリケーション・プログラムが要求を作成します。この要求はパイプラインを介してサービス・プロバイダーに渡されます。サービス・プロバイダーからの応答は、同じパイプラインを介してアプリケーション・プログラムに戻されます。

CICS がサービス・リクエスターの役割を果たす場合、CICS は以下の操作を実行します。

1. アプリケーション・プログラムから得られたデータを使用して、要求を作成する。
2. 要求をサービス・プロバイダーに送信する。
3. サービス・プロバイダーから応答を受信する。
4. 応答を調べ、オリジナル・アプリケーション・プログラムに関係のある内容を抽出する。
5. アプリケーション・プログラムに制御を戻す。

図 5 には、サービス・リクエスターの設定における次の 3 つのメッセージ・ハンドラーのパイプラインを示します。

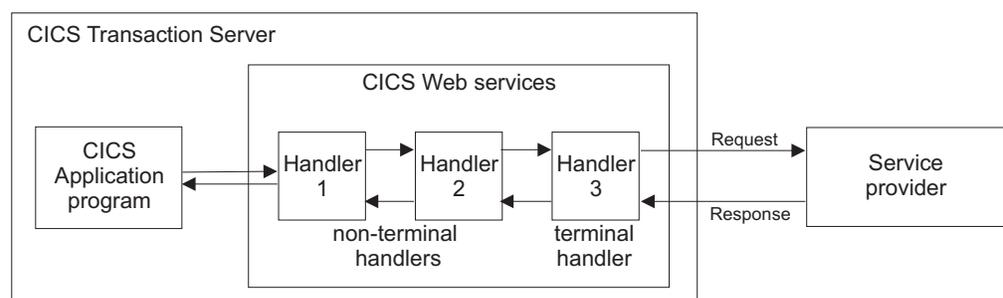


図 5. サービス・リクエスター・パイプライン

1. アプリケーション・プログラムが要求を作成します。

2. メッセージ・ハンドラー 1 は、アプリケーション・プログラムから要求を受け取り、何らかの処理を実行して、要求をハンドラー 2 に渡します (正確には、パイプラインを管理する CICS に制御を戻します。次に CICS は、次のメッセージ・ハンドラーに制御を渡します)。
3. メッセージ・ハンドラー 2 は、ハンドラー 1 から要求を受け取り、何らかの処理を実行して、要求をハンドラー 3 に渡します。
4. メッセージ・ハンドラー 3 は、ハンドラー 2 から要求を受け取り、何らかの処理を実行して、要求をサービス・プロバイダーに渡します。
5. メッセージ・ハンドラー 3 はサービス・プロバイダーから応答を受け取り、何らかの処理を実行して、応答をハンドラー 2 に渡します。
6. メッセージ・ハンドラー 2 は、ハンドラー 3 から応答を受け取り、何らかの処理を実行して、応答をハンドラー 1 に渡します。
7. メッセージ・ハンドラー 1 は、ハンドラー 2 から応答を受け取り、何らかの処理を実行して、応答をアプリケーション・プログラムに戻します。

CICS パイプラインおよび SOAP

Web サービスの要求と応答を処理するために CICS が使用するパイプラインは、各メッセージ・ハンドラーで実行できる処理に関する制約が少ないという点で一般的です。ただし、多くの Web サービス・アプリケーションは SOAP メッセージを使用しており、これらのメッセージを処理する場合は、SOAP 仕様に準拠する必要があります。したがって、CICS には、パイプラインを SOAP ノードとして構成するときに役立つ特殊な SOAP メッセージ・ハンドラー・プログラムが用意されています。

- パイプラインを構成すると、SOAP 1.1 または SOAP 1.2 をサポートできます。CICS システム内部には、多数のパイプラインを保持できます。このうちのいくつかは SOAP 1.1 または SOAP 1.2 をサポートします。
- パイプラインは、サービス・リクエスターまたはサービス・プロバイダーで使用するために、次のように構成できます。
 - サービス・リクエスター・パイプラインは、要求の最初の SOAP 送信側になり、かつ応答の最終の SOAP 受信側になります。
 - サービス・プロバイダー・パイプラインは、要求の最終の SOAP 受信側になり、かつ応答の最初の SOAP 送信側になります。

CICS パイプラインを SOAP 中間ノードとして機能するように構成することはできません。

- CICS パイプラインを構成すると、複数の SOAP メッセージ・ハンドラーを保持できます。
- CICS 提供の SOAP メッセージ・ハンドラーを構成すると、1 つ以上のユーザー作成ヘッダー処理ルーチンを呼び出すことができます。
- CICS 提供の SOAP メッセージ・ハンドラーを構成すると、WS-I Basic Profile 1.0 に準拠するといういくつかの側面と、SOAP メッセージに特定のヘッダーを存在させることを実現できます。

SOAP メッセージ・ハンドラー、およびそのヘッダー処理ルーチンは、パイプライン構成ファイルで指定します。

SOAP メッセージおよびアプリケーション・データ構造

多くの場合、CICS Web サービス・アシスタントは、アプリケーション・プログラムで使用されている上位データ構造と、SOAP メッセージの <Body> エレメントの内容の間でデータを変換するコードを生成できます。これらの場合には、アプリケーション・プログラムを作成するときに、SOAP 本体の解析または構成を行う必要はありません。これらの作業は CICS によって実行されるためです。

CICS は、データを変換するために、実行時にアプリケーション・データ構造と SOAP メッセージの形式についての情報が必要です。この情報は、次の 2 つのファイルに保持されます。

- Web サービスのバインディング・ファイル

このファイルは、ユーティリティー・プログラム DFHLS2WS を使用して、アプリケーション言語のデータ構造を基に CICS Web サービス・アシスタントによって生成されるか、またはユーティリティー・プログラム DFHWS2LS を使用して、Web サービス記述を基に生成されます。CICS は、バインディング・ファイルを使用して、Web サービス・アプリケーションで使用するリソースを生成し、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実行します。

- Web サービス記述

これは、既存の Web サービス記述である場合と、ユーティリティー・プログラム DFHLS2WS を使用して、アプリケーション言語のデータ構造を基に生成する場合があります。CICS では、Web サービス記述を使用して、SOAP メッセージの全妥当性検査を行います。

図 6 は、これらのファイルがサービス・プロバイダー内のどこで使用されるかを示しています。

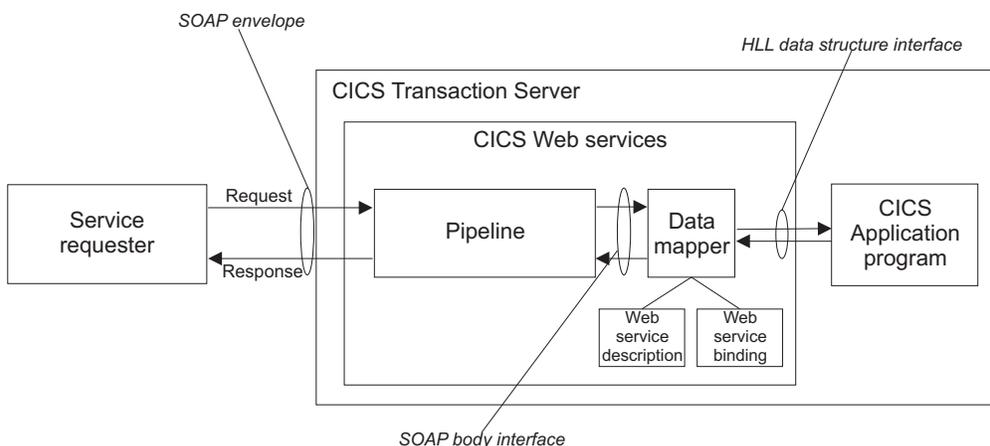


図 6. サービス・プロバイダーでの SOAP 本体からアプリケーション・データ構造へのマッピング

23 ページの図 7 には、これらのファイルがサービス・リクエスターで使用される様子を示します。

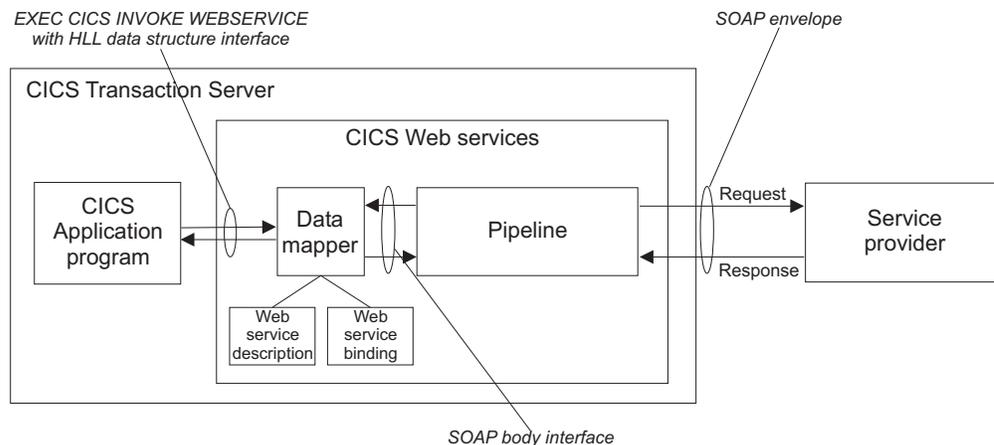


図7. サービス・リクエスターでの SOAP 本体からアプリケーション・データ構造へのマッピング

どちらの場合も、特定の CICS アプリケーション・プログラムが Web サービスの設定で動作できる実行環境は、3 つのオブジェクトによって定義されます。これらは、パイプライン、Web サービス・バインディング・ファイル、および Web サービス記述です。これら 3 つのオブジェクトは、WEBSERVICE リソース定義の属性として CICS に定義されています。

SOAP メッセージを使用している場合でも、次に示すように、CICS Web サービス・アシスタントが生成する変換を使用できない状況もあります。

- SOAP メッセージと高水準言語で同じデータが表現できない場合

CICS がサポートしているすべての高水準言語、および XML スキーマでは、さまざまなデータ・タイプがサポートされています。しかし、高水準言語で 사용되는データ・タイプと XML スキーマで使用されるデータ・タイプとの間に 1 対 1 対応は存在しないため、データを一方で表現できても他方では表現できないという場合が存在します。こうした状況では、次のいずれかの手段を検討する必要があります。

- アプリケーション・データ構造を変更する。この方法は、必然的にアプリケーション・プログラム自体の変更が必要になるため、実現は困難です。
- ラッパー・プログラムを作成する。このプログラムは、アプリケーション・データを CICS が処理可能な形式に変換し、さらに SOAP メッセージ本文に変換します。この方法を実行した場合は、アプリケーション・プログラムを変更せずに済みます。この場合は、CICS Web サービス・サポートがラッパー・プログラムと直接対話し、アプリケーション・プログラムとは間接的にのみ対話します。

- アプリケーション・プログラムが、CICS Web サービス・アシスタントでサポートされていない言語で作成されている場合。

こうした状況では、次のいずれかの手段を検討する必要があります。

- CICS Web サービス・アシスタントでサポートされている言語の 1 つ (COBOL、PL/I、C または C++) を使用して、ラッパー・プログラムを作成する。

- CICS Web サービス・アシスタントを使用せずに、独自のプログラムを作成して、SOAP メッセージとアプリケーション・プログラムのデータ構造との間のマッピングを実行する。

WSDL およびアプリケーション・データ構造

Web サービス記述には、Web サービスが使用する入出力メッセージの抽象表現が含まれています。CICS では、Web サービス記述を使用して、アプリケーション・プログラムが使用するデータ構造を構成します。CICS は、実行時に、アプリケーション・データ構造とメッセージとのマッピングを実行します。

Web サービス記述の代表例は、以下のとおりです。

- 1 つ以上の操作
- 各操作ごとに、入力メッセージ、およびオプションの出力メッセージ。
- メッセージごとに、XML データ・タイプの観点で定義されたメッセージ構造。メッセージ内で使用される複素数データ・タイプは、Web サービス記述内にある `<types>` エレメントに記述されている XML スキーマで定義されます。簡単なメッセージは、`<types>` エレメントを使用しないで記述できます。

WSDL には、操作の抽象定義と関連メッセージが記述されています。このため、WSDL をアプリケーション・プログラム内で直接使用することはできません。操作を実装するには、サービス・プロバイダーが以下の処理を実行する必要があります。

- メッセージの構造を把握するために WSDL の構文解析を行う。
- 入力メッセージを解析して出力メッセージを作成する。
- 入出力メッセージの内容と、アプリケーション・プログラムで使用されているデータ構造とのマッピングを実行する。

サービス・リクエスターは、操作を呼び出すために同じことを行う必要があります。

CICS Web サービス・アシスタントを使用すると、前述の大半の処理がユーザーの代わりに実行されるため、ユーザーは入出力メッセージを構成する方法や WSDL を詳細に理解する必要なくアプリケーション・プログラムを作成できます。

CICS Web サービス・アシスタントは、以下の 2 つのユーティリティー・プログラムで構成されています。

DFHWS2LS

このユーティリティー・プログラムは、Web サービス記述を開始点にしています。このプログラムでは、アプリケーション・プログラムに使用できる高水準言語データ構造を構成するために、メッセージの記述や、メッセージに使用されているデータ・タイプを使用します。

DFHLS2WS

このユーティリティー・プログラムは、高水準言語データ構造を開始点にしています。このプログラムでは、メッセージの記述を格納する Web サービス記述を構成するための構造体と、言語構造から導出されたこれらのメッセージで使用されるデータ・タイプが使用されます。

いずれのユーティリティー・プログラムも、アプリケーション・プログラムのデータ構造と SOAP メッセージ間のマッピングを実行するために CICS が実行時に使用する Web サービス・バインディング・ファイルを生成します。

COBOL から WSDL へのマッピングの例

この例では、COBOLプログラムで使用されているデータ構造が、CICS Web サービス・アシスタントによって生成された Web サービス記述内でどのように表現されているかを示します。

図 8 は、単純な COBOL データ構造を示しています。

```
* Catalogue COMMAREA structure
  03 CA-REQUEST-ID          PIC X(6).
  03 CA-RETURN-CODE        PIC 9(2).
  03 CA-RESPONSE-MESSAGE  PIC X(79).
* Fields used in Place Order
  03 CA-ORDER-REQUEST.
    05 CA-USERID           PIC X(8).
    05 CA-CHARGE-DEPT     PIC X(8).
    05 CA-ITEM-REF-NUMBER PIC 9(4).
    05 CA-QUANTITY-REQ   PIC 9(3).
    05 FILLER              PIC X(888).
```

図 8. WSDL で定義された入力メッセージの COBOL レコード定義

Web サービス記述で対応しているフラグメントのうち重要なエレメントを、26 ページの図 9 に示します。

```

<xsd:sequence>
  <xsd:element name="CA-REQUEST-ID" nillable="false">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:length value="6"/>
        <xsd:whiteSpace value="preserve"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="CA-RETURN-CODE" nillable="false">
    <xsd:simpleType>
      <xsd:restriction base="xsd:short">
        <xsd:maxInclusive value="99"/>
        <xsd:minInclusive value="0"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="CA-RESPONSE-MESSAGE" nillable="false">
    ...
  </xsd:element>
  <xsd:element name="CA-ORDER-REQUEST" nillable="false">
    <xsd:complexType mixed="false">
      <xsd:sequence>
        <xsd:element name="CA-USERID" nillable="false">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:length value="8"/>
              <xsd:whiteSpace value="preserve"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="CA-CHARGE-DEPT" nillable="false">
          ...
        </xsd:element>
        <xsd:element name="CA-ITEM-REF-NUMBER" nillable="false">
          ...
        </xsd:element>
        <xsd:element name="CA-QUANTITY-REQ" nillable="false">
          ...
        </xsd:element>
        <xsd:element name="FILLER" nillable="false">
          ...
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:sequence>

```

図9. COBOL データ構造から派生された WSDL フラグメント

Web サービスのバインディング・ファイル

Web サービスのバインディング・ファイルには、入出力メッセージとアプリケーション・データ構造との間でデータをマップするために CICS が使用する情報が格納されています。

Web サービス記述には、Web サービスが使用する入出力メッセージの抽象表現が含まれています。サービス・プロバイダーまたはサービス・リクエスターのアプリケーションを実行する場合、CICS に必要な情報は、メッセージの内容をアプリケーションが使用するデータ構造にマップする方法です。この情報は、Web サービスのバインディング・ファイルに保持されます。

Web サービスのバインディング・ファイルの作成方法は、次のとおりです。

- 言語構造を WSDL を基に生成する場合は、ユーティリティー・プログラム DFHWS2LS
- WSDL を言語構造を基に生成する場合は、ユーティリティー・プログラム DFHLS2WS

CICS は、実行時に Web サービスのバインディング・ファイルを使用して、アプリケーション・データ構造と SOAP メッセージとのマッピングを行います。Web サービスのバインディング・ファイルは、WEBSERVICE リソースの WSBIND 属性で、CICS に対して定義されます。

CICS での Web サービス使用の計画

CICS で Web サービスを使用する計画を立てるには、その前に以下の問題をアプリケーションごとに検討する必要があります。

サービス・プロバイダーまたはサービス・リクエスターの役割で CICS アプリケーションを配置する計画ですか？

Web サービスの CICS サポートを使用して接続する 1 組のアプリケーションが存在する場合があります。この場合、一方のアプリケーションはサービス・プロバイダー、もう一方はサービス・リクエスターになります。

既存のアプリケーション・プログラムを使用する計画ですか、それとも新規のアプリケーション・プログラムを作成する計画ですか？

既存のアプリケーションが、ビジネス・ロジックに対して適切に定義されたインターフェースを使用して設計されている場合は、サービス・プロバイダーまたはサービス・リクエスターとして、このアプリケーションを Web サービス設定に使用できる確率が高くなります。ただし、ほとんどの場合は、ビジネス・ロジックを Web サービス・ロジックに接続するラッパー・プログラムを作成する必要があります。

新規アプリケーションの作成を計画している場合は、ビジネス・ロジックを Web サービス・ロジックから分離した状態を維持するようにします。また、この場合も、この分離状態を実現するためにラッパー・プログラムを作成する必要があります。ただし、アプリケーションが Web サービスを考慮して設計されている場合、ラッパーは簡単に作成できる可能性が高くなります。

SOAP メッセージを使用する予定ですか？

SOAP は Web サービス・アーキテクチャーの基礎で、CICS で提供されるサポートの多くでは、SOAP が使用されていることが前提となっています。しかし、別のメッセージ・フォーマットを使用したい場合もあるかもしれません。例えば、CICS Web サービス・インフラストラクチャーに実装するメッセージ・フォーマットを独自に開発した場合などです。別のメッセージ・フォーマットの使用は許可されますが、Web サービス・アシスタントや SOAP メッセージ・ハンドラーのような CICS 提供の機能の一部を使用することはできなくなります。

データ構造と SOAP メッセージ間のマッピングを生成するために CICS Web サービス・アシスタントを使用する予定ですか？

Web サービス・アシスタントでは、少量の追加プログラミングまたは追加プログラミングなしで、Web サービス設定に多数のアプリケーションを迅速に配置できます。また、追加プログラミングが必要な場合でも、プログラミングは容易で、既存のビジネス・ロジックを変更しないでプログラミングできます。

ただし、Web サービス・アシスタントを使用しないほうが処理しやすいケースもあります。例えば、データ構造と SOAP メッセージをマップする既存のコードがある場合、Web サービス・アシスタントを使用してアプリケーションを再設計する利点はありません。

既存のサービス記述を使用する予定ですか、それとも新規のサービス記述を作成する予定ですか？

状況によっては、既存のサービス記述を開始点として使用する必要があります。例えば、次のようになります。

- アプリケーションはサービス・リクエスターであり、既存の Web サービスを呼び出すよう設計されている。
- アプリケーションはサービス・プロバイダーであり、既存の業界標準サービス記述にこのアプリケーションを適合させることを目的としている。

その他の状況では、アプリケーションに応じて新規のサービス記述を作成する必要があります。

次のステップ:

- サービス・プロバイダーの計画
- サービス・リクエスターの計画

サービス・プロバイダー・アプリケーションの計画

一般に、CICS アプリケーションは、ビジネス・ロジックとコミュニケーション・ロジックを確実に分離できるよう構造化する必要があります。この手法に従うと、新規および既存のアプリケーションを Web サービス・プロバイダーで直接的に配置するのに役立ちます。状況によっては、アプリケーション・プログラムと CICS Web サービス・サポートとの間に単純なラッパー・プログラムを介在させる必要があります。

図 10 には、コミュニケーション・ロジックとビジネス・ロジックとを確実に分離するために分割された標準的なアプリケーションを示します。

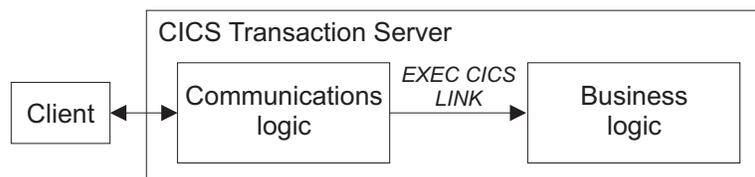


図 10. コミュニケーション・ロジックとビジネス・ロジックに分割されたアプリケーション

多くの場合、ビジネス・ロジックは、サービス・プロバイダー・アプリケーションの場合と同様に直接配置できます。このことは、29 ページの図 11 に図示されています。

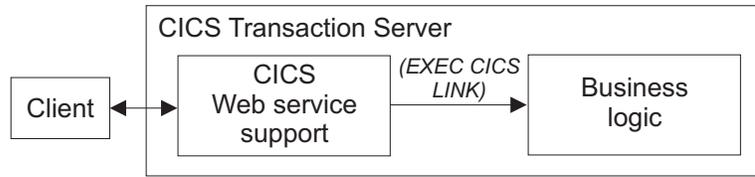


図 11. Web サービス・プロバイダーとしての CICS アプリケーションの単純な配置

この単純なモデルを使用する場合は、以下の条件が適用されます。

CICS Web サービス・アシスタントを使用して SOAP メッセージとアプリケーション・データ構造間のマッピングを生成する場合:

ビジネス・ロジックのインターフェースで使用されるデータ・タイプは、CICS Web サービス・アシスタントによってサポートされている必要があります。これが当てはまらない場合、CICS Web サービス・サポートとビジネス・ロジックの間にラッパー・プログラムを置く必要があります。

既存のプログラムを配置して既存の Web サービス記述に適合するサービスを提供する場合は、ラッパー・プログラムも必要になります。Web サービス・アシスタントを使用して Web サービス記述を処理すると、結果として得られるデータ構造がビジネス・ロジックのインターフェースと一致する可能性は非常に低くなります。

CICS Web サービス・アシスタントを使用していない場合:

サービス・プロバイダー・パイプラインに存在するメッセージ・ハンドラーは、ビジネス・ロジックと直接対話する必要があります。

ラッパー・プログラムの使用

CICS Web サービス・アシスタントではビジネス・ロジックと直接対話するためのコードを生成できない場合は、ラッパー・プログラムを使用します。例えば、ビジネス・ロジックのインターフェースは、CICS Web サービス・アシスタントが SOAPメッセージに直接マップできないデータ構造を使用する可能性があります。この状況では、ラッパー・プログラムを使用すると、必要なデータ操作を追加できます。

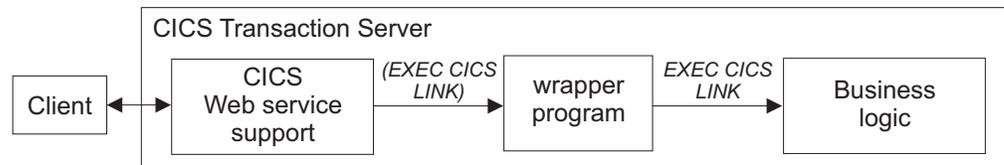


図 12. ラッパー・プログラム使用による、Web サービス・プロバイダーとしての CICS アプリケーションの配置

アシスタントがサポートできる 2 番目のデータ構造を設計して、これをラッパー・プログラムのインターフェースとして使用する必要があります。この結果、ラッパー・プログラムが実行する機能は、以下に示す 2 つの単純な機能となります。

- 2 つのデータ構造間でのデータの移動
- 既存のインターフェースによるビジネス・ロジックの呼び出し

サービス・リクエスター・アプリケーションの計画

一般に、CICS アプリケーションは、ビジネス・ロジックとコミュニケーション・ロジックを確実に分離できるよう構造化する必要があります。この手法に従うと、新規および既存のアプリケーションを Web サービス・リクエスターで直接的に配置するのに役立ちます。ほとんどすべての状況では、アプリケーション・プログラムと CICS Web サービス・サポートとの間に単純なラッパー・プログラムを介在させる必要があります。

図 13 は、コミュニケーション・ロジックとビジネス・ロジックとを確実に分離するために分割された標準的なアプリケーションを示します。このアプリケーションは、Web サービス・リクエスターでビジネス・ロジックを再使用するために最適な構造になっています。

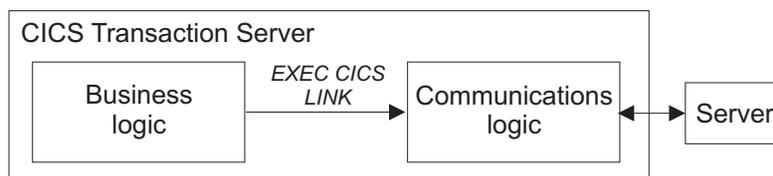


図 13. コミュニケーション・ロジックとビジネス・ロジックに分割されたアプリケーション

以下の状況では、既存の EXEC CICS LINK コマンドを使用して CICS Web サービス・サポートを起動することはできません。

- CICS Web サービス・アシスタントを使用して SOAP メッセージとアプリケーション・データ構造間のマッピングを生成する場合、EXEC CICS INVOKE WEBSERVICE コマンドを使用して、アプリケーションのデータ構造を CICS Web サービス・サポートに渡します。また、ビジネス・ロジックのインターフェースで使用されるデータ・タイプは、CICS Web サービス・アシスタントによってサポートされている必要があります。
- CICS Web サービス・アシスタントを使用していない場合、独自のメッセージを作成して、プログラム DFHPIRT にリンクする必要があります。

どちらのケースも、プログラム変更の準備が整っていないかぎり、ビジネス・ロジックは Web サービスを直接呼び出すことはできないということになります。Web サービス・アシスタントのこのオプションについては 図 14 に示していますが、どちらのケースの参考にもなりません。

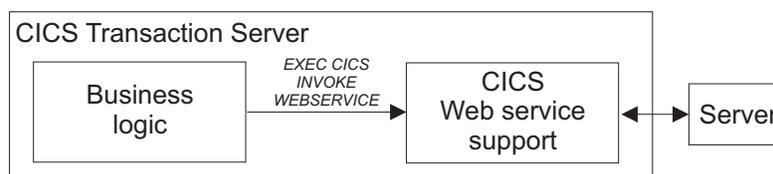


図 14. Web サービス・リクエスターとしての CICS アプリケーションの単純な配置

ラッパー・プログラムの使用

ビジネス・ロジックを未変更のまま維持する、より優れた解決策は、ラッパー・プログラムを使用することです。この場合、ラッパー・プログラムには次の 2 つの目的があります。

- ビジネス・ロジックの代わりに EXEC CICS INVOKE WEBSERVICE コマンドまたは EXEC CICS LINK PROGRAM(DFHPIRT) を発行します。ビジネス・ロジックは未変更のまま維持できます。
- CICS Web サービス・アシスタントが SOAP メッセージに直接マップできないデータ構造をアプリケーションが使用する場合、ラッパー・プログラムは、これに必要なデータ操作を必要に応じて提供できます。

Web サービス・アシスタントを使用するケースの場合、この構造は図 15 に示されています。

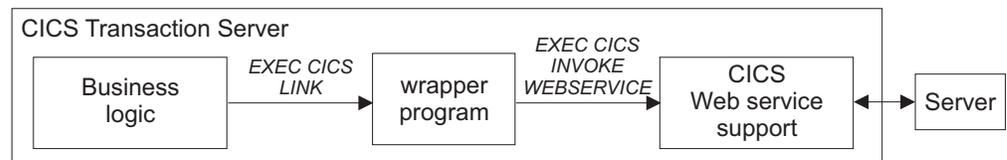


図 15. ラッパー・プログラム使用による、Web サービス・リクエスターとしての CICS アプリケーションの配置

CICS Web サービス・アシスタント

CICS Web サービス・アシスタントとは、1 組のバッチ・ユーティリティーで、既存の CICS アプリケーションを Web サービスに変換するのに役立ちます。また、これを使用すると、CICS アプリケーションが、外部のプロバイダーによって提供された Web サービスを使用できるようになります。このアシスタントは、サービス・プロバイダーやサービス・リクエスターが使用するための CICS アプリケーションの迅速な配置をサポートしており、プログラミングの労力が最小限で済みます。

CICS の Web サービス・アシスタントを使用する場合は、インバウンド・メッセージを解析し、アウトバウンド・メッセージを作成するための独自のコードを記述する必要はありません。CICS は、SOAP メッセージ本文とアプリケーション・プログラムのデータ構造間でデータをマップするからです。

リソース定義は、その大半が自動的に生成およびインストールされます。PIPELINE リソースを定義する必要がありますが、多くの場合は、CICS 提供のパイプライン構成ファイルを使用できます。この内容は次のとおりです。

basicsoap11provider.xml

SOAP 1.1 メッセージ・ハンドラーを使用したサービス・プロバイダーのパイプライン構成ファイル

basicsoap11requester.xml

SOAP 1.1 メッセージ・ハンドラーを使用したサービス・リクエスターのパイプライン構成ファイル

このアシスタントでは、単純な言語構造から WSDL 文書を作成したり、既存の WSDL 文書から言語構造を作成したりすることができ、COBOL、C/C++、および PL/I をサポートします。また、SOAP メッセージからコンテナおよび COMMAREA への自動ランタイム変換、さらに、この逆の変換 を可能にするために使用される情報も生成します。

ただし、Web サービス・アシスタントはすべてのケースを処理できるわけではありません。別の方法をとる必要がある場合もあります。例えば、次のようになります。

SOAP メッセージを使用しない場合

SOAP 以外のプロトコルをメッセージに使用することが望ましい場合は、そうすることができます。ただし、アプリケーション・プログラムには、インバウンド・メッセージの解析とアウトバウンド・メッセージの作成を行う役割が与えられます。

SOAP メッセージは使用するが、CICS による解析はしない場合

インバウンド・メッセージの場合、Web サービス・アシスタントは、SOAP 本体とアプリケーションのデータ構造をマップします。アプリケーションによっては、ユーザー自身による SOAP 本体の解析が必要な場合があります。

CICS Web サービス・アシスタントは、ユーザー・アプリケーションのデータ構造をサポートしません。

CICS Web サービス・アシスタントは最も一般的なデータ・タイプおよびデータ構造をサポートしますが、なかにはサポートされないものもあります。このような状態では、ご使用のアプリケーションのデータを Web サービス・アシスタントがサポートできるフォーマットにマップするプログラム層を提供することをまず初めに考える必要があります。これが不可能な場合は、自分でメッセージを解析する必要があります。

CICS Web サービス・アシスタントを使用しない場合、次の処理を実行する必要があります。

- インバウンド・メッセージを解析し、アウトバウンド・メッセージを作成するための独自のコードを準備する
- 独自のパイプライン構成ファイルを準備する
- 独自の URIMAP リソースおよび PIPELINE リソースを定義してインストールする

CICS Web サービス・アシスタントは、以下の 2 つのユーティリティ・プログラムで構成されています。

DFHLS2WS

言語構造を基にして Web サービス・バインディング・ファイルを生成します。このユーティリティは、Web サービス記述も生成します。

DFHWS2LS

Web サービス記述を基にして Web サービス・バインディング・ファイルを生成します。このユーティリティは、アプリケーション・プログラムで使用できる言語構造も生成します。

両方のプログラムを実行する JCL プロシージャは、*hlq.XDFHINST* ライブラリーに格納されています。

DFHLS2WS: 高水準言語から WSDL への変換

カタログ式プロシージャ DFHLS2WS は、高水準言語データ構造を基にして、Web サービス記述および Web サービス・バインディング・ファイルを生成します。CICS アプリケーション・プログラムをサービス・プロバイダーとして公開する場合、DFHLS2WS を使用することができます。

DFHLS2WS のジョブ制御ステートメント

JOB ジョブを開始します。

EXEC プロシージャ名 (DFHLS2WS) を指定します。

DFHLS2WS には、Java 仮想マシン (JVM) を実行するのに十分なストレージが必要です。EXEC ステートメントで REGION=0M と指定することをお勧めします。

INPUT.SYSUT1 DD

入力を指定します。入力パラメーターは、通常、入力ストリーム内に指定します。ただし、データ・セットや区分データ・セットのメンバーに定義することもできます。

シンボリック・パラメーター

以下のシンボリック・パラメーターは、カタログ式プロシージャ DFHLS2WS で定義されます。

JAVADIR=*path*

DFHLS2WS によって使用される Java ディレクトリーの名前を指定します。このパラメーターの値は /usr/lpp/ に追加され、これによって /usr/lpp/*path* という完全なパス名が得られます。

通常、このパラメーターを指定する必要はありません。デフォルト値は、**JAVADIR** パラメーターの CICS インストール・ジョブ (DFHISTAR) に指定された値です。

USSDIR=*path*

UNIX システム・サービス HFS の CICS TS ディレクトリーの名前を指定します。このパラメーターの値は /usr/lpp/cicsts/ に追加され、これによって /usr/lpp/cicsts/*path* という完全なパス名が得られます。

通常、このパラメーターを指定する必要はありません。デフォルト値は、**USSDIR** パラメーターの CICS インストール・ジョブ (DFHISTAR) に指定された値です。

TMPDIR=*tmpdir*

DFHLS2WS が一時ワークスペースとして使用する HFS のディレクトリーの場所を指定します。このジョブを実行するユーザー ID には、このディレクトリーに対する読み取り権限および書き込み権限が必要です。

デフォルト値は /tmp です。

TMPFILE=tmpprefix

一時ワークスペース・ファイルの名前を作成するために DFHLS2WS が使用する接頭部を指定します。

デフォルト値は、DFHLS2WS です。

一時ワークスペース

DFHLS2WS は、実行時に、次の 3 つの一時ファイルを作成します。

```
tmpdir/tmpprefix.in  
tmpdir/tmpprefix.out  
tmpdir/tmpprefix.err
```

ここで

tmpdir は、**TMPDIR** パラメーターに指定されている値です。

tmpprefix は、**TMPFILE** パラメーターに指定されている値です。

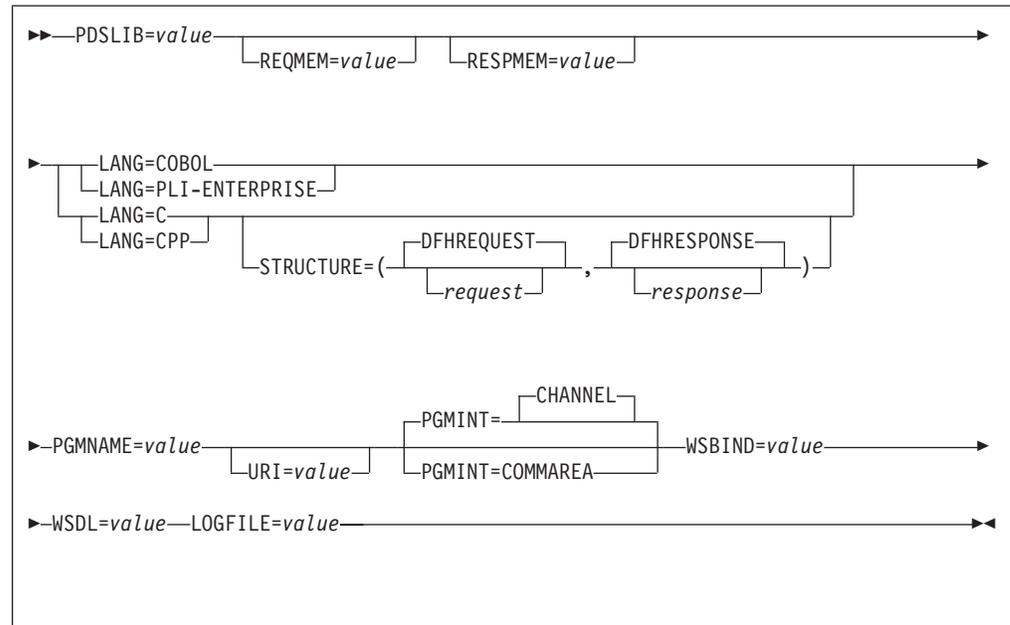
ファイルのデフォルトの名前は (**TMPDIR** および **TMPFILE** が指定されていない場合)、次のとおりです。

```
/tmp/DFHLS2WS.in  
/tmp/DFHLS2WS.out  
/tmp/DFHLS2WS.err
```

重要: DFHLS2WS は、生成された HFS ファイル名へのアクセスをロックしません。したがって、DFHLS2WS のインスタンスが同時に 2 つ以上動作し、同じ一時ワークスペース・ファイルを使用した場合、あるジョブが一時ワークスペース・ファイルを使用中に別のジョブがそのファイルを上書きするのを防ぐ方法はありません。この状況は、予測不能な障害の発生につながります。

したがって、この状況を回避できる命名規則および操作手順を考案することをお勧めします。例えば、システム・シンボリック・パラメーター **SYSUID** を使用すると、個々のユーザーに対して一意のワークスペース・ファイルを生成できます。

DFHLS2WS の入力パラメーター



パラメーターの使用法

- 入力パラメーターの指定順序は自由です。
- 各パラメーターは改行後に記述を始める必要があります。
- パラメーターが長すぎて 1 行に収まらない場合は、行の末尾にアスタリスク文字 (*) を使用して、そのパラメーターが次の行に続くことを示します。アスタリスクより前の文字はすべて (スペースを含む) パラメーターの一部とみなされます。例えば、次のようになります。

```
WSBIND=wsbinddir*  
/app1
```

このコードは、次のコードと同じ意味になります。

```
WSBIND=wsbinddir/app1
```

- 行の先頭の文字の位置に # という文字がある場合、この文字はコメント文字を表します。この行は無視されます。

パラメーターの記述

LANG=COBOL

高水準言語構造のプログラム言語を COBOL に指定します。

LANG=PLI-ENTERPRISE

高水準言語構造のプログラム言語を Enterprise PL/I に指定します。

LANG=C

高水準言語構造のプログラム言語を C に指定します。

LANG=CPP

高水準言語構造のプログラム言語を C++ に指定します。

PDSLIB=value

処理の対象となる高水準言語データ構造が格納されている区分データ・セットの

名前を指定します。要求および応答に使用されるデータ・セット・メンバーは、それぞれ、**REQMEM** パラメーターおよび **RESPMEM** パラメーターで指定されます。

制約事項: 区分データ・セット内のレコードは、80 バイトの固定長にする必要があります。

PGMINT=CHANNEL|COMMAREA

サービス・プロバイダーの場合は、CICS によってターゲット・アプリケーション・プログラムにデータを渡す方法を次のように指定します。

CHANNEL

CICS は、チャンネル・インターフェースを使用して、データをターゲット・アプリケーション・プログラムに渡します。

COMMAREA

CICS は、通信域を使用して、データをターゲット・アプリケーション・プログラムに渡します。

DFHLS2WS からの出力がサービス・リクエスターで使用される場合、このパラメーターは無視されます。

PGMNAME=value

Web サービスとして公開されるターゲット CICS アプリケーション・プログラムの名前を指定します。これは、CICS Web サービス・サポートがリンクするプログラムです。

REQMEM=value

Web サービス要求の高水準言語データ構造が格納されている区分データ・セット・メンバーの名前を指定します。

- サービス・プロバイダーの場合、Web サービス要求は、アプリケーション・プログラムの入力になります。
- サービス・リクエスターの場合、Web サービス要求は、アプリケーション・プログラムの出力になります。

RESPMEM=value

Web サービス応答の高水準言語データ構造が格納されている区分データ・セット・メンバーの名前を指定します。

- サービス・プロバイダーの場合、Web サービス応答は、アプリケーション・プログラムの出力になります。
- サービス・リクエスターの場合、Web サービス応答は、アプリケーション・プログラムの入力になります。

応答がない場合 (つまり、片方向メッセージの場合) は、このパラメーターを省略します。

STRUCTURE=(request,response)

C と C++ の場合にのみ、REQMEM パラメーターと RESPMEM パラメーターに指定されているパーティション・データ・セット・メンバーに含まれる上位の構造体の名前を指定します。

request

REQMEM パラメーターが指定されている場合に要求に含まれる上位の構造体の名前を指定します。デフォルト値は DFHREQUEST です。

パーティション・データ・セット・メンバーには、指定した名前を持つ上位の構造体 (または、名前を指定していない場合は DFHREQUEST という名前の構造体) が含まれている必要があります。

response

RESPMEM パラメーターが指定されている場合に応答に含まれる上位の構造体の名前を指定します。デフォルト値は DFHRESPONSE です。

値を指定した場合、パーティション・データ・セット・メンバーには、指定した名前を持つ上位の構造体 (名前を指定していない場合は DFHRESPONSE という名前の構造体) が含まれている必要があります。

URI=value

サービス・プロバイダーでは、このパラメーターには、クライアントが Web サービスにアクセスするのに使用する相対 URI が指定されます。CICS は、DFHLS2WS によって作成された Web サービス・バインディング・ファイルから URIMAP リソースを生成する場合、指定された値を使用します。このパラメーターには、URIMAP 定義を適用する URI のパス・コンポーネントが指定されます。

WSBIND=value

Web サービス・バインディング・ファイルの完全修飾 HFS 名。ファイルが存在しない場合、DFHLS2WS がファイルを作成します (ただし、ディレクトリー構造は作成しません)。

WSDL=value

Web サービス記述が格納されるファイルの完全修飾 HFS 名です。ファイルが存在しない場合、DFHLS2WS がファイルを作成します (ただし、ディレクトリー構造は作成しません)。

LOGFILE=value

DFHLS2WS がアクティビティー・ログとトレース情報を書き込むファイルの完全修飾 HFS 名です。ファイルが存在しない場合、DFHLS2WS がファイルを作成します (ただし、ディレクトリー構造は作成しません)。

通常はこのファイルを使用する必要はありませんが、DFHLS2WS に問題が発生した場合、このファイルの提出を IBM のサービス組織から依頼される場合があります。

その他の情報

- OMVS に対して、DFHLS2WS を実行するユーザー ID が定義されている必要があります。このユーザー ID には、CICS HFS ファイル構造の読み取り権限、および LOGFILE パラメーターに指定されているディレクトリーへの書き込み権限が必要です。
- Java を実行するため、このユーザー ID には十分な大きさのストレージを割り振る必要があります。

DFHWS2LS: WSDL から高水準言語への変換

カタログ式プロシージャ DFHWS2LS は、Web サービス記述を基にして、高水準言語データ構造および Web サービス・バインディング・ファイルを生成します。

CICS アプリケーション・プログラムをサービス・プロバイダーとして公開する場合、またはサービス・リクエスターを構成する場合、DFHWS2LS を使用することができます。

DFHWS2LS のジョブ制御ステートメント

JOB ジョブを開始します。

EXEC プロシージャ名 (DFHWS2LS) を指定します。

DFHWS2LS には、Java 仮想マシン (JVM) を実行するのに十分なストレージが必要です。EXEC ステートメントで REGION=0M と指定することをお勧めします。

INPUT.SYSUT1 DD

入力を指定します。入力パラメーターは、通常、入力ストリーム内に指定します。ただし、データ・セットや区分データ・セットのメンバーに定義することもできます。

シンボリック・パラメーター

以下のシンボリック・パラメーターは、カタログ式プロシージャ DFHWS2LS で定義されます。

JAVADIR=*path*

DFHWS2LS によって使用される Java ディレクトリーの名前を指定します。このパラメーターの値は /usr/lpp/ に追加され、これによって /usr/lpp/*path* という完全なパス名が得られます。

通常、このパラメーターを指定する必要はありません。デフォルト値は、**JAVADIR** パラメーターの CICS インストール・ジョブ (DFHISTAR) に指定された値です。

USSDIR=*path*

UNIX システム・サービス HFS の CICS TS ディレクトリーの名前を指定します。このパラメーターの値は /usr/lpp/cicsts/ に追加され、これによって /usr/lpp/cicsts/*path* という完全なパス名が得られます。

通常、このパラメーターを指定する必要はありません。デフォルト値は、**USSDIR** パラメーターの CICS インストール・ジョブ (DFHISTAR) に指定された値です。

TMPDIR=*tmpdir*

DFHWS2LS が一時ワークスペースとして使用する HFS のディレクトリーの場所を指定します。このジョブを実行するユーザー ID には、このディレクトリーに対する読み取り権限および書き込み権限が必要です。

デフォルト値は /tmp です。

TMPFILE=*tmpprefix*

一時ワークスペース・ファイルの名前を作成するために DFHWS2LS が使用する接頭部を指定します。

デフォルト値は DFHWS2LS です。

一時ワークスペース

DFHWS2LS は、実行時に、次の 3 つの一時ファイルを作成します。

```
tmpdir/tmpprefix.in  
tmpdir/tmpprefix.out  
tmpdir/tmpprefix.err
```

ここで

tmpdir は、**TMPDIR** パラメーターに指定されている値です。
tmpprefix は、**TMPFILE** パラメーターに指定されている値です。

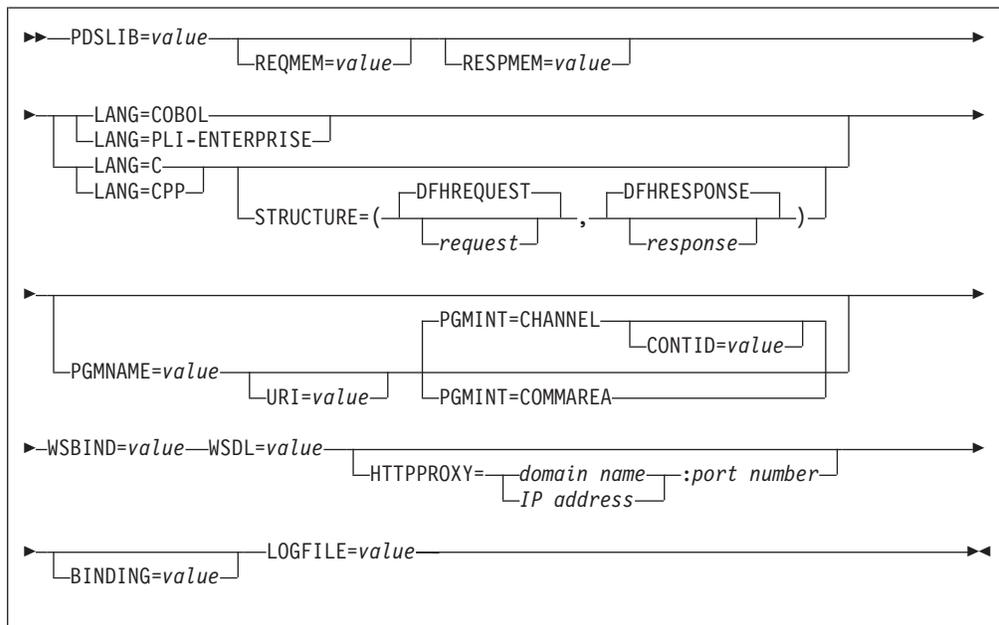
ファイルのデフォルトの名前は (**TMPDIR** および **TMPFILE** が指定されていない場合)、次のとおりです。

```
/tmp/DFHWS2LS.in  
/tmp/DFHWS2LS.out  
/tmp/DFHWS2LS.err
```

重要: DFHWS2LS は、生成された HFS ファイル名へのアクセスをロックしません。したがって、DFHWS2LS のインスタンスが同時に 2 つ以上動作し、同じ一時ワークスペース・ファイルを使用した場合、あるジョブが一時ワークスペース・ファイルを使用中に別のジョブがそのファイルを上書きするのを防ぐ方法はありません。この状況は、予測不能な障害の発生につながります。

したがって、この状況を回避できる命名規則および操作手順を考案することをお勧めします。例えば、システム・シンボリック・パラメーター **SYSUID** を使用すると、個々のユーザーに対して一意のワークスペース・ファイルを生成できます。

DFHWS2LS のパラメーターの入力



パラメーターの使用法

- 入力パラメーターの指定順序は自由です。
- 各パラメーターは改行後に記述を始める必要があります。
- パラメーターが長すぎて 1 行に収まらない場合は、行の末尾にアスタリスク文字 (*) を使用して、そのパラメーターが次の行に続くことを示します。アスタリスクより前の文字はすべて (スペースを含む) パラメーターの一部とみなされます。例えば、次のようになります。

```
WSBIND=wsbinddir*
/app1
```

このコードは、次のコードと同じ意味になります。

```
WSBIND=wsbinddir/app1
```

- 行の先頭の文字の位置に # という文字がある場合、この文字はコメント文字を表します。この行は無視されます。

パラメーターの記述

HTTPPROXY={domain name|IP address}:port number

WSDL に、インターネット上に置かれている他の WSDL ファイルへの参照が含まれており、DFHWS2LS を実行しているシステムがプロキシー・サーバーを使用してインターネットにアクセスする場合は、そのプロキシー・サーバーのドメイン名、IP アドレス、およびポート番号を指定します。例を次に示します。

```
HTTPPROXY=proxy.example.com:8080
```

その他の場合、このパラメーターは必要ありません。

LANG=COBOL

高水準言語構造のプログラム言語を COBOL に指定します。

LANG=PLI-ENTERPRISE

高水準言語構造のプログラム言語を Enterprise PL/I に指定します。

LANG=C

高水準言語構造のプログラム言語を C に指定します。

LANG=CPP

高水準言語構造のプログラム言語を C++ に指定します。

PDSLIB=value

生成された高水準言語を含む区分データ・セットの名前を指定します。要求および応答に使用されるデータ・セット・メンバーは、それぞれ、**REQMEM** パラメーターおよび **RESPMEM** パラメーターで指定されます。

PGMINT=CHANNEL|COMMAREA

サービス・プロバイダーの場合は、CICS によってターゲット・アプリケーション・プログラムにデータを渡す方法を次のように指定します。

CHANNEL

CICS は、チャンネル・インターフェースを使用して、データをターゲット・アプリケーション・プログラムに渡します。

COMMAREA

CICS は、通信域を使用して、データをターゲット・アプリケーション・プログラムに渡します。

DFHWS2LS からの出力がサービス・リクエスターで使用される場合、このパラメーターは無視されます。

PGMNAME=value

このパラメーターには、CICS プログラムの名前が指定されます。

サービス・プロバイダーで使用される Web サービス・バインディング・ファイルを生成するのに DFHWS2LS が使用されている場合、このパラメーターを指定する必要があります。このパラメーターには、Web サービスとして公開されているアプリケーション・プログラムの名前が指定されます。

サービス・リクエスターで使用される Web サービス・バインディング・ファイルを生成するのに DFHWS2LS が使用されている場合、このパラメーターは省略する必要があります。

REQMEM=value

以下に示す Web サービス要求の高水準言語構造体が格納されている区分データ・セット・メンバーの名前を生成するときに DFHWS2LS が使用する 1 から 6 文字の接頭部を指定します。

- サービス・プロバイダーの場合、Web サービス要求は、アプリケーション・プログラムの入力になります。
- サービス・リクエスターの場合、Web サービス要求は、アプリケーション・プログラムの出力になります。

DFHWS2LS は、操作ごとに、区分データ・セットのメンバーを生成します。このプログラムは、接頭部に 2 桁の数値を付加することによってメンバー名を生成します。

このパラメーターはオプションですが、Web サービス記述に要求の定義が記述されている場合は、指定する必要があります。

RESPMEM=*value*

以下に示す Web サービス応答の高水準言語構造体が格納されている区分データ・セット・メンバーの名前を生成するときに DFHWS2LS が使用する 1 から 6 文字の接頭部を指定します。

- サービス・プロバイダーの場合、Web サービス応答は、アプリケーション・プログラムの出力になります。
- サービス・リクエスターの場合、Web サービス応答は、アプリケーション・プログラムの入力になります。

DFHWS2LS は、操作ごとに、区分データ・セットのメンバーを生成します。このプログラムは、接頭部に 2 桁の数値を付加することによってメンバー名を生成します。

応答がない場合 (つまり、片方向メッセージの場合) は、このパラメーターを省略します。

STRUCTURE=*(request,response)*

C と C++ の場合にのみ、要求構造体と応答構造体の名前を生成する方法を指定します。

生成された要求構造体と応答構造体には、*requestnn* および *responsenn* という名前が付けられます。ここで、*nm* は、操作ごとに構造体を区別するために生成される数値接尾部を表します。

片方または両方の名前が省略されると、構造体の名前は、指定した REQMEM パラメーターおよび RESPMEM パラメーターから生成されるパーティション・データ・セット・メンバー名と同じになります。

URI=*value*

サービス・プロバイダーでは、このパラメーターには、クライアントが Web サービスにアクセスするのに使用する相対 URI が指定されます。CICS は、DFHWS2LS によって作成された Web サービス・バインディング・ファイルから URIMAP リソースを生成するときに指定された値を使用します。このパラメーターには、URIMAP 定義を適用する URI のパス・コンポーネントが指定されます。

サービス・リクエスターでは、ターゲット Web サービスの URI はこのパラメーターには指定されません。EXEC CICS INVOKE WEBSERVICE コマンドの URI オプションで指定変更できますが、Web サービス記述に指定された URI が使用されます。

WSBIND=*value*

Web サービス・バインディング・ファイルの完全修飾 HFS 名。ファイルが存在しない場合、DFHWS2LS がファイルを作成します (ただし、ディレクトリー構造は作成しません)。

WSDL=*value*

Web サービス記述が格納されるファイルの完全修飾 HFS 名です。

LOGFILE=*value*

DFHWS2LS がアクティビティー・ログとトレース情報を書き込むファイルの完全修飾 HFS 名です。ファイルが存在しない場合、DFHWS2LS がファイルを作成します (ただし、ディレクトリー構造は作成しません)。

通常はこのファイルを使用する必要はありませんが、DFHWS2LS に問題が発生した場合は、このファイルの提出を IBM のサービス組織から依頼される場合があります。

BINDING=value

Web サービス記述に複数の <binding> エLEMENTが格納されている場合は、このパラメーターを使用して、言語構造および Web サービス・バインディング・ファイルを生成するためにどのELEMENTを使用するかを指定します。Web サービス記述の <binding> ELEMENTに使用される name 属性の値を指定します。

その他の情報

- OMVS に対して、DFHWS2LS を実行するユーザー ID が定義されている必要があります。このユーザー ID には、CICS HFS ファイル構造の読み取り権限、および LOGFILE パラメーターに指定されているディレクトリーへの書き込み権限が必要です。
- Java を実行するため、このユーザー ID には十分な大きさのストレージを割り振る必要があります。

CICS 外部インターフェースの変更点

リソース定義の変更点

CICS における Web サービスでは、新しい CICS リソース PIPELINE、URIMAP、および WEBSERVICE を使用します。

PIPELINE

PIPELINE リソース定義は、Web サービス要求に適用される処理を指定します。詳細については、44 ページの『PIPELINE リソース定義』を参照してください。

PIPELINE は、処理ノードを定義する XML ファイルを参照します。

URIMAP

URIMAP 定義は、CICS が Web クライアントからの要求 (すなわち、リモート・サーバーに対する要求) の URI を突き合わせ、要求を処理する方法に関する情報を提供できるようにします。詳細については、134 ページの『URIMAP リソース定義』を参照してください。

URIMAP リソースの以下の属性が関連します。

PIPELINE

クライアントからのサービス要求に従って処理を実行するメッセージ・ハンドラーに関する情報を提供する PIPELINE リソース定義を指定します。

WEBSERVICE

Web サービスの設定に使用される CICS アプリケーションのランタイム環境の性質を定義する WEBSERVICE を指定します。

TRANSACTION

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指定します。

USERID

別名トランザクションを接続する際に使用するユーザー IDを指定します。

WEBSERVICE

WEBSERVICE リソース定義は、Web サービス要求に適用される処理を指定します。詳細については、47 ページの『WEBSERVICE リソース定義』を参照してください。

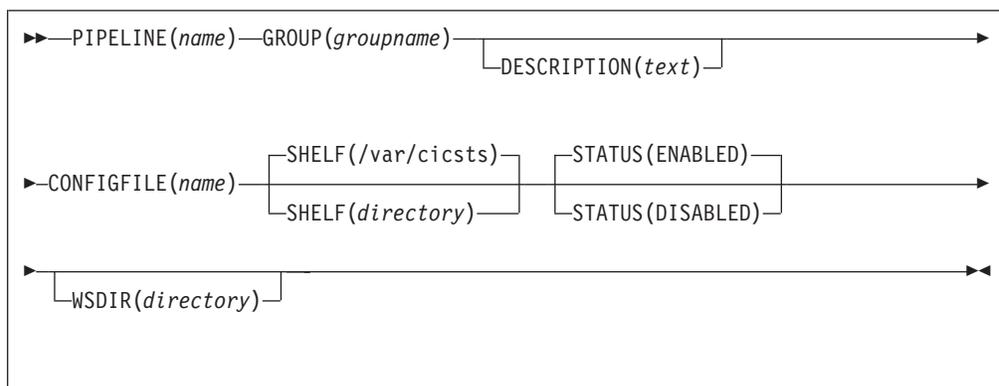
PIPELINE リソース定義

PIPELINE リソース定義は、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担う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用するメッセージ・ハンドラー・プログラムに関する情報を提供します。一般に、単一の PIPELINE 定義で定義されたインフラストラクチャーを、多数のアプリケーションで使用できます。

処理ノードに関する情報は、間接的に提供されます。PIPELINE は、ノードの XML 記述およびその構成が定義されている HFS ファイルの名前を指定します。

インバウンド Web サービス要求 (すなわち、クライアントが CICS で Web サービスを呼び出すための要求) 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 に適用される PIPELINE リソースを識別します。PIPELINE は、メッセージに対して実行される処理を指定します。

PIPELINE 属性:



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 コマンドを使用する場合を除き、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

一になり、グループがインストールされるときに 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。DFH で始まるグループ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済みです。

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。使用できる文字についての制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧ごとに、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。

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 HFS ファイルの名前を指定します。

許容文字:

A-Z a-z 0-9 . / _

指定する値は、以下のように HFS ファイルとして有効な名前であればなりません。

- 名前には、埋め込みスペース文字を含める必要がない
- 名前に、/ が連続したインスタンスを含めることはできない

名前は大文字小文字が区別されます。

SHELF(*{/var/cicsts/}**directory*)

HFS 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 文字の範囲で指定します。

許容文字:

A-Z a-z 0-9 . / _

指定する値は、以下のように HFS ファイルとして有効な名前であればなりません。

- 名前には、埋め込みスペース文字を含める必要がない
- 名前に、/ が連続したインスタンスを含めることはできない

名前は大文字小文字が区別されます。

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 領域は個別のサブディレクトリーを使用し、自己のファイルを他の CICS 領域のファイルとは分離して保管します。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。

CICS 領域は、コールド・スタートまたは初期スタートを実行した後で、シェルフの使用を試行する前にシェルフから自己のサブディレクトリーを削除します。

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。

STATUS({ENABLED|DISABLED})

PIPELINE がインストールされるときに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)

HFS 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。

許容文字:

A-Z a-z 0-9 . / _

指定する値は、以下のように HFS ファイルとして有効な名前であればなりません。

- 名前には、埋め込みスペース文字を含める必要がない
- 名前に、/ が連続したインスタンスを含めることはできない

名前は大文字小文字が区別されます。

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 定義がインストールされるときに、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 HFS 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 の値を指定しなかった場合は、PIPELINE のインストール時に自動スキャンは行われず、PERFORM PIPELINE SCAN コマンドは失敗します。

WEBSERVICE リソース定義

WEBSERVICE リソースは、Web サービス設定で配置された CICS アプリケーション・プログラムのためのランタイム環境の特徴を定義します。この環境では、アプリケーション・データ構造と SOAP メッセージとの間のマッピングが、CICS Web サービス・アシスタントを使用して生成されます。CICS では WEBSERVICE リソースのための通常のリソース定義メカニズムを提供していますが、通常これらは、アシスタントが生成する出力を使用して、動的にインストールされます。

WEBSERVICE リソースによって定義されるランタイム環境の特徴は、次のとおりです。

パイプライン

Web サービス要求および応答に対して作用するメッセージ・ハンドラーのセットを定義します。WEBSERVICE リソースは別個の PIPELINE リソースを定義し、この PIPELINE 定義がパイプライン構成ファイルを定義します。

Web サービス・バインディング・ファイル

アプリケーション・データ構造と SOAP メッセージとの間のマッピングを行うために実行時に使用する情報が格納されています。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 サービス記述に埋め込まれているスキーマと突き合わせて行われます。

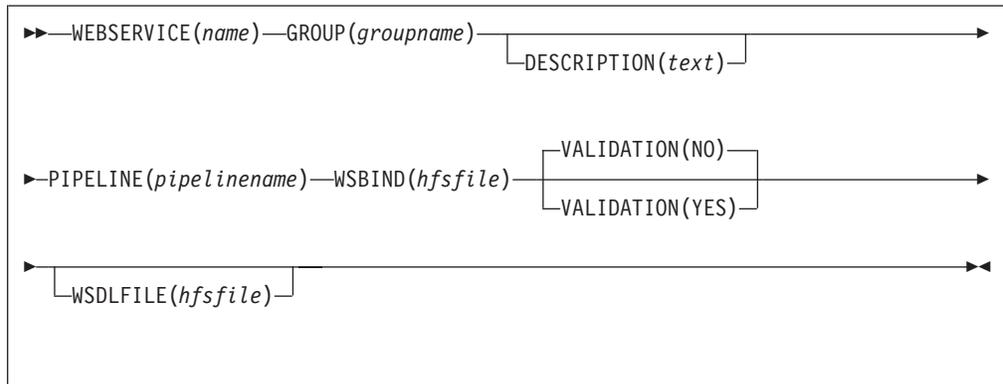
インバウンド Web サービス要求 (すなわち、クライアントが CICS の Web サービスを呼び出すための要求) は、URIMAP リソースによって WEBSERVICE リソースに関連付けられます。URIMAP は、インバウンド・メッセージの URI に適用される WEBSERVICE リソースを識別します。WEBSERVICE は、メッセージに対して実行される処理を指定します。

CICS では、WEBSERVICE リソースを作成し、これを 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ていますが、その代わりにスキャン・メカニズムを使用して、WEBSERVICE リソースを稼働中の CICS システムに動的にインストールすることができます。この方法の利点は、必要なリソース定義の量を削減し、開発時に提供された情報を CICS が直接利用できることにあります。

スキャン・メカニズムを起動するには、PERFORM PIPELINE コマンドを使用します。

動的にインストールされた WEBSERVICE の名前は、WEBSERVICE 定義の生成に使用される Web サービス・バインディング・ファイルの名前から得られ、最長 32 文字です。CSD からインストールされるか、または EXEC CICS CREATE WEBSERVICE を使用してインストールされる WEBSERVICE 定義の名前は 8 文字までに制限されています。例えば、HFS 名が /samples/Webservices/WSDir/InquireSingle.wsbind の Web サービス・バインディング・ファイルから、InquireSingle という名前の WEBSERVICE 定義が生成されます。

WEBSERVICE 属性:



WEBSERVICE(*name*)

WEBSERVICE の 1 から 8 文字の名前を指定します。

許容文字:

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

DFH で始まる名前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済みです。

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、グループがインストールされるときに 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。DFH で始まるグループ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済みです。

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。使用できる文字についての制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧ごとに、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 が関連付けられている PIPELINE の 1 から 8 文字の名前を指定します。

許容文字:

A-Z 0-9 \$ @ #

CREATE コマンドを使用する場合を除き、小文字を入力した場合は、大文字に変換されます。

VALIDATION(NO|YES)

実行時に Web サービス記述の対応するスキーマと突き合わせて SOAP メッセージの完全な妥当性検査を行うかどうかを指定します。SOAP メッセージをそのスキーマに突き合わせて妥当性検査すると、著しい処理オーバーヘッドが生じるため、通常は VALIDATION(NO) を指定してください。

完全な妥当性検査により、送受信されるすべての SOAP メッセージが、XML スキーマに関して有効な XML であることが保証されます。VALIDATION(NO) を指定した場合、メッセージが適切な形式の XML であることを確認するために十分な妥当性検査が実行されます。

WSBIND(hfsfile)

HFS 上の Web サービス・バインディング・ファイルの 1 から 255 文字の完全修飾ファイル名を指定します。

許容文字:

A-Z a-z 0-9 . / _

名前は大文字小文字が区別され、スペースを含めることができません。名前は / で終わることができず、/ の連続したインスタンスを含むことができません。

WSDLFILE(hfsfile)

HFS 上の Web サービス記述 (WSDL) ファイルの 1 から 255 文字の完全修飾ファイル名を指定します。完全なランタイム妥当性検査がアクティブな場合に、このファイルが使用されます。

許容文字:

A-Z a-z 0-9 . / _

名前は大文字小文字が区別され、スペースを含めることができません。名前は / で終わることができず、/ の連続したインスタンスを含むことができません。

アプリケーション・プログラミング・インターフェースの変更点

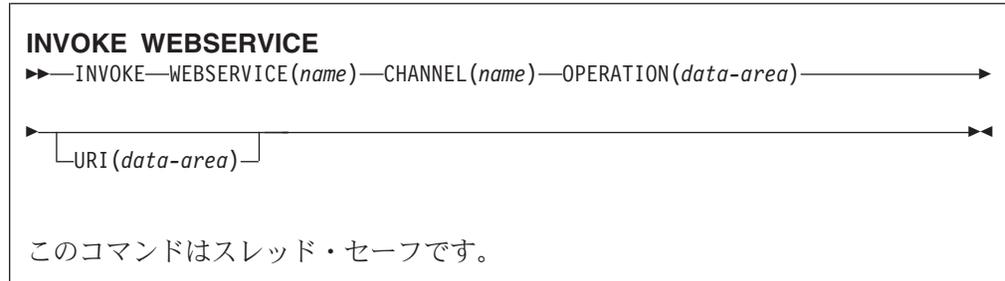
CICS アプリケーション・プログラムからの Web サービスの呼び出しを可能にするコマンドが新しく追加されました。詳細については、50 ページの『INVOKE WEBSERVICE』を参照してください。

SOAP ノードで SOAP 障害を構成できるようにする、3 つの新しいコマンドが追加されました。詳細については、以下を参照してください。

- 51 ページの『SOAPFAULT ADD』
- 52 ページの『SOAPFAULT CREATE』
- 54 ページの『SOAPFAULT DELETE』

INVOKE WEBSERVICE

このコマンドは、CICS アプリケーションから Web サービスを呼び出します。このコマンドは、WEBSERVICE リソースの名前を指定します。このリソースには、呼び出されるサービスの情報が格納されています。



オプション

CHANNEL(*name*)

アプリケーション・データ構造によってマップされたデータを格納するコンテナを渡すために使用されるチャンネルの名前を指定します。戻り時には、同じチャンネルに Web サービスからの応答が保持されていますが、この応答もアプリケーション・データ構造によってマップされています。チャンネルの名前は、16 文字以内で指定できます。*name* が変数で、16 文字未満の名前を指定する場合、この変数を末尾ブランクで埋め込む必要があります。

OPERATION(*data-area*)

呼び出すべき操作の名前を格納するデータ域を指定します。操作の名前は、ターゲット Web サービスの WSDL に格納されています。このデータ域の長さは 255 文字でなければなりません。操作の名前が 255 文字未満である場合、データ域を末尾ブランクで埋め込む必要があります。

URI(*data-area*)

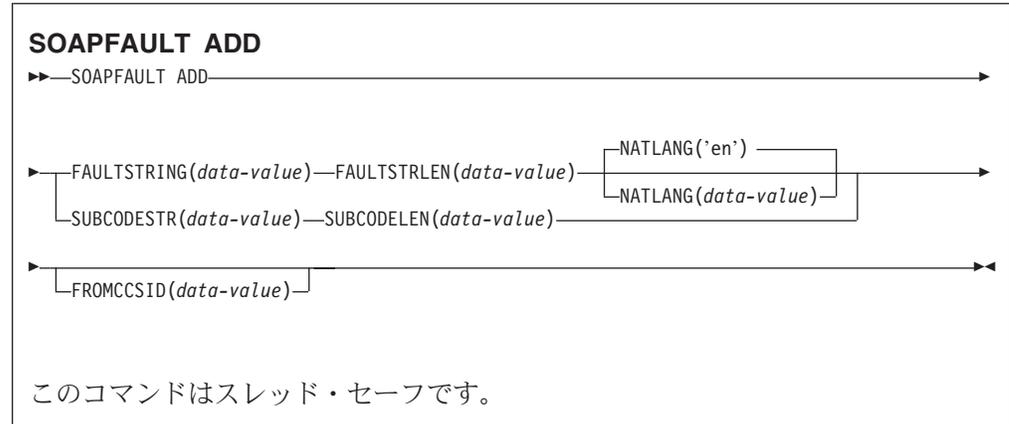
呼び出すべき Web サービスの URI を格納するデータ域を指定します。このオプションを指定すると、WEBSERVICE リソース定義に指定されている URI が置き換えられます。このオプションを省略する場合は、WEBSERVICE リソース定義にプロバイダーの URI またはプロバイダーのアプリケーション名を指定する必要があります。このデータ域の長さは 255 文字でなければなりません。URI が 255 文字未満である場合、データ域を末尾ブランクで埋め込む必要があります。

WEBSERVICE(*name*)

呼び出すべき Web サービスを定義する WEBSERVICE リソースの名前を指定します。WEBSERVICE リソースは、Web サービス記述の場所を指定し、CICS が Web サービスと通信するとき使用する Web サービス・バインディング・ファイルの場所を指定します。WEBSERVICE の名前は 32 文字以内で指定できます。*name* が変数で、32 文字未満の名前を指定する場合、この変数を末尾ブランクで埋め込む必要があります。

SOAPFAULT ADD

このコマンドは、既存の SOAPFAULT オブジェクトに情報を追加します。このコマンドは、CICS 提供の SOAP メッセージ・ハンドラーから起動されるプログラムでのみ使用可能です。



オプション

SUBCODESTR(data-value)

SOAPFAULT オブジェクトに追加する <subcode> エレメントの内容を指定します。サブコードの長さは、64 文字以内で指定することができます。また、XML 修飾名 (QName) であることが必要です。

- SOAP 1.1 では、このオプションは無視されます。
- SOAP 1.2 では、このオプションで <Subcode> エレメントの内容が指定されます。

SUBCODELEN(data-value)

SUBCODESTR オプションで指定された <subcode> エレメントの長さをフルワード・バイナリー値で指定します。

FAULTSTRING(data-value)

人間が読み取れる障害の説明を指定します。FAULTSTRING は、2056 字以内の長さにすることができます。

- SOAP 1.1 では、このオプションで <faultstring> エレメントの内容が指定されます。
- SOAP 1.2 では、このオプションで <Reason> エレメントの内容が指定されません。

FAULTSTRLEN(data-value)

FAULTSTRING オプションの長さをフルワード・バイナリー値で指定します。

FROMCCSID(data-value)

SOAP 障害を書き込む文字データの現在のコード化文字セット ID (CCSID) をフルワード 10 進数で指定します。このオプションを指定しないと、CICS は LOCALCCSID システム初期設定パラメーターに指定された値を使用します。

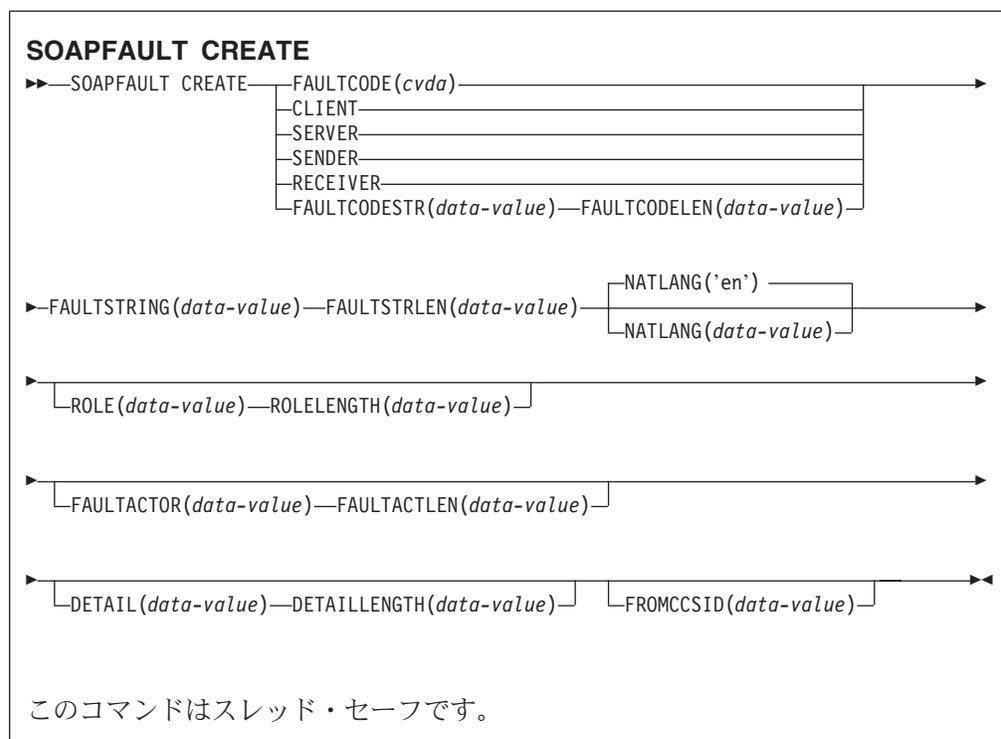
CCSID に関する詳細と CICS でサポートされる CCSID のリストについては、『CICS ファミリー: S/390 CICS からの通信』を参照してください。

NATLANG(*data-value*)

FAULTSTRING で使用される各国語を XML 1.0 の言語識別番号を使用して指定します。デフォルト値は「en」（英語）です。指定する値は 8 バイトを超えてはいけません。

SOAPFAULT CREATE

このコマンドは、SOAP 障害コードを作成します。このコマンドは、CICS 提供の SOAP メッセージ・ハンドラーから起動されるプログラムでのみ使用可能です。



オプション

DETAIL(*data-value*)

- SOAP 1.1 では、このオプションに SOAP 障害の <detail> エレメントの内容が指定されます。
- SOAP 1.2 では、このオプションに SOAP 障害の <Detail> エレメントの内容が指定されます。

これには、1 つ以上の有効なネームスペース修飾 XML エレメントか空白文字のどちらかが含まれます。エレメントの有効な内容についてのすべての説明は、適切な SOAP 仕様書を参照してください。

エレメントは、<Body> エレメントに関連し、<Body> エレメントの内容が正常に処理されなかった場合に使用されるアプリケーション固有のエラー情報を伝えます。SOAP 1.1 では、<Body> エレメントの内容が正常に処理されなかった場合、<detail> エレメントが存在している必要があります。SOAP 1.2 では、<Detail> エレメントはオプションです。

ヘッダー・ハンドラー・プログラムで SOAPFAULT CREATE コマンドが発行されると、<detail> または <Detail> エレメントがヘッダー・ブロック内に運ばれます。

DETAILLENGTH(*data-value*)

DETAIL オプションの長さをフルワード・バイナリー値で指定します。

FAULTACTLEN(*data-value*)

FAULTACTOR オプションの長さをフルワード・バイナリー値で指定します。

FAULTACTOR(*data-value*)

- SOAP 1.1 では、このオプションで <faultactor> エレメントの内容が指定されます。
- SOAP 1.2 では、このオプションで <Node> エレメントの内容が指定されます。

FAULTACTOR オプションは、2056 文字以内で指定することができます。また、有効な URI (anyURI) であることが必要です。

FAULTCODE(*cvda*)

CLIENT

SENDER

SOAP 1.1 では、Client の SOAP 障害コードを指定します。

SOAP 1.2 では、Sender の SOAP 障害コードを指定します。

SERVER

RECEIVER

SOAP 1.1 では、Server の SOAP 障害コードを指定します。

SOAP 1.2 では、Receiver の SOAP 障害コードを指定します。

FAULTCODELEN(*data-value*)

FAULTCODESTR オプションの長さをフルワード・バイナリー値で指定します。

FAULTCODESTR(*data-value*)

ユーザー定義の SOAP 障害コードを指定します。障害コードは、64 文字以内で指定することができます。また、XML 修飾名 (QName) であることが必要です。複数の障害コード値を区切るための「.」(ドット) の使用はサポートされていません。

- SOAP 1.1 では、このオプションで <faultcode> エレメントの内容が指定されます。
- SOAP 1.2 では、このオプションで <Code> エレメントの内容が指定されます。

FAULTSTRING(*data-value*)

人間が読み取れる障害の説明を指定します。FAULTSTRING は、2056 字以内の長さにするすることができます。

- SOAP 1.1 では、このオプションで <faultstring> エレメントの内容が指定されます。
- SOAP 1.2 では、このオプションで <Reason> エレメントの内容が指定されます。

FAULTSTRLEN(*data-value*)

FAULTSTRING オプションの長さをフルワード・バイナリー値で指定します。

FROMCCSID(*data-value*)

SOAP 障害を書き込む文字データの現在のコード化文字セット ID (CCSID) をフルワード 10 進数で指定します。このオプションを指定しないと、CICS は LOCALCCSID システム初期設定パラメーターに指定された値を使用します。CCSID に関する詳細と CICS でサポートされる CCSID のリストについては、『CICS ファミリー: S/390 CICS からの通信』を参照してください。

NATLANG(*data-value*)

FAULTSTRING で使用される各国語を XML 1.0 の言語識別番号を使用して指定します。デフォルト値は「en」（英語）です。指定する値は 8 バイトを超えてはいけません。

ROLE(*data-value*)

障害が生成された SOAP ノードの役割を記述した URI を指定します。ROLE オプションは、2056 文字以内で指定することができます。また、有効な URI (XML タイプ anyURI) であることが必要です。

- SOAP 1.1 では、このオプションは無視されます。
- SOAP 1.2 では、このオプションで <Role> エレメントの内容が指定されます。

ROLELENGTH(*data-value*)

ROLE オプションの長さをフルワード・バイナリー値で指定します。

SOAPFAULT DELETE

このコマンドは、既存の SOAPFAULT オブジェクトを削除します。このコマンドは、CICS 提供の SOAP メッセージ・ハンドラーから起動されるプログラムでのみ使用可能です。

SOAPFAULT DELETE

▶—SOAPFAULT DELETE—▶

このコマンドはスレッド・セーフです。

システム・プログラミング・インターフェースの変更点

COLLECT STATISTICS コマンド

このコマンドは、以下の新しいオプションをサポートします。

PIPELINE(*data-value*)

PIPELINE に関連する統計を要求します。これには、HFS ファイルに関する情報が含まれます。

WEBSERVICE(*data-value*)

WEBSERVICE に関連する統計を要求します。

CREATE PIPELINE コマンド

CREATE PIPELINE コマンドは、CICS 領域に PIPELINE を動的に作成するために使用します。このコマンドで指定できる属性については、44 ページの『PIPELINE 属性』で説明しています。

CREATE WEBSERVICE コマンド

CREATE WEBSERVICE コマンドは、CICS 領域に WEBSERVICE を動的に作成するために使用します。このコマンドで指定できる属性については、48 ページの『WEBSERVICE 属性』で説明しています。

DISCARD PIPELINE コマンド

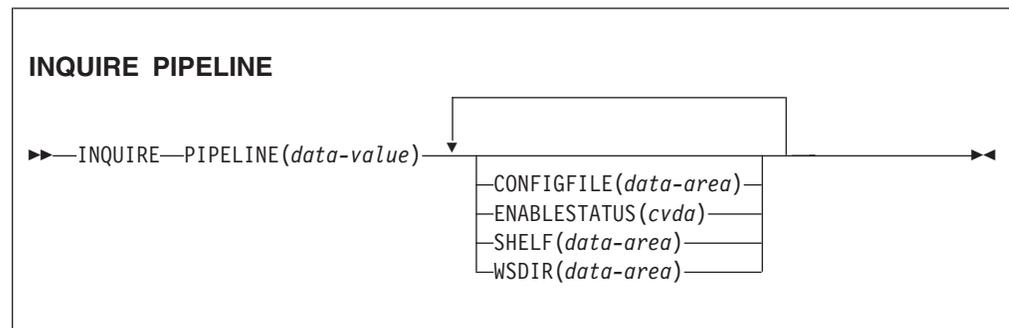
DISCARD PIPELINE パイプラインは、CICS 領域から PIPELINE を除去するために使用します。PIPELINE を廃棄する場合は、廃棄の前にこれを使用不可に設定する必要があります。

DISCARD WEBSERVICE コマンド

DISCARD WEBSERVICE コマンドは、CICS 領域から WEBSERVICE を除去するために使用します。WEBSERVICE を廃棄する場合は、廃棄の前にこれを使用不可に設定する必要があります。

INQUIRE PIPELINE コマンド

INQUIRE PIPELINE は、インストール済み PIPELINE に関する情報を取得するために使用します。



INQUIRE PIPELINE コマンドでブラウズ・オプション (START、NEXT、および END) を使用することにより、システムにインストールされているすべての PIPELINE をブラウズできます。

オプション

CONFIGFILE(*data-area*)

PIPELINE リソースに関連付けられているパイプライン構成ファイルの名前を戻します。この名前は、255 文字以内の長さにすることができます。

ENABLESTATUS(*cvda*)

PIPELINE の状況を戻します。

ENABLED

PIPELINE は使用する準備ができています。

DISABLED

PIPELINE は要求を処理しておらず、新規作業を受け入れることができません。初期化が失敗したか、明示的に使用不可にされている可能性があります。

ENABLING

PIPELINE は初期化中です。まだ作業を受け入れる準備ができていません。

DISABLING

PIPELINE は、DISABLED 状態になる前に静止しています。新規作業は受け入れていませんが、現在実行中の作業は完了できます。

DISCARDING

この PIPELINE のために DISCARD コマンドが発行されました。PIPELINE は、廃棄される前に静止しています。新規作業は受け入れていませんが、現在実行中の作業は完了できます。

PIPELINE(*data-value*)

照会する 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。

SHELF(*data-area*)

シェルフ・ディレクトリーの名前を戻します。この名前は、255 文字以内の長さにすることができます。

WSDIR(*data-area*)

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) の名前を戻します。この名前は、255 文字以内の長さにすることができます。

INQUIRE WORKREQUEST コマンド

INQUIRE WORKREQUEST コマンドの WORKTYPE オプションに、以下の新しい値が追加されました。

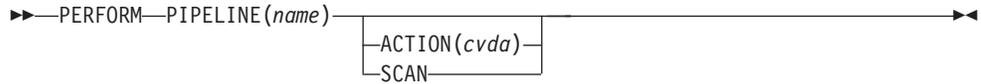
SOAP

Web サービス要求のために実行される処理を指定します。

PERFORM PIPELINE コマンド

PERFORM PIPELINE コマンドは、PIPELINE に関連付けられている Web サービス・バインディング・ファイルのスキャンを開始する目的で使用します。

PERFORM PIPELINE



スキャンされる Web サービス・バインディング・ファイルは、PIPELINE 定義の WSBIND 属性で指定されたディレクトリー内にあります。WSBIND 属性を指定しなかった場合、スキャンするものではなく、制御はユーザーのプログラムに戻ります。

指定されたディレクトリー・ロケーションが有効な場合、CICS は、そのディレクトリー内の Web サービス・バインディング・ファイルを調べて、これらをシステムにインストールするかどうかを判別します。

- まだインストールされていないファイルが見つかった場合、CICS はこれらのファイルをすべてインストールします。
- ファイルが既にインストールされているが、ディレクトリー内のファイルが現在使用中のファイルよりも新しい場合、使用中のファイルが廃棄され、代わりに新しいファイルがインストールされます。

何らかの理由で CICS が個々の Web サービス・バインディング・ファイルのインストールに失敗した場合、ディレクトリー内の残りのファイルで処理が続けられます。スキャンが完了すると、正常にインストールされたすべてのバインディング・ファイルが PIPELINE で使用可能です。

オプション

ACTION(*cvda*)

PIPELINE に対して実行する処理を示す CVDA 値を指定します。CVDA 値は、以下のとおりです。

SCAN PIPELINE の Web サービス・バインディング・ディレクトリーをスキャンします。

PIPELINE(*name*)

この PIPELINE の名前を指定します。

PERFORM STATISTICS RECORD コマンド

このコマンドは、以下の新しいオプションをサポートします。

PIPELINE

PIPELINE に関連する統計を記録します。これには、HFS ファイルに関する情報が含まれます。

SET PIPELINE コマンド

SET PIPELINE コマンドは、インストール済み PIPELINE の状況を変更するために使用します。

PIPELINE(*data-value*)

照会する PIPELINE の 8 文字の名前を指定します。

ENABLESTATUS(*cvda*)

PIPELINE の状況を指定します。

ENABLED

この PIPELINE のインバウンド・サービス要求は通常どおりに処理されます。

DISABLED

この PIPELINE のインバウンド・サービス要求は拒否されます。

SET WORKREQUEST コマンド

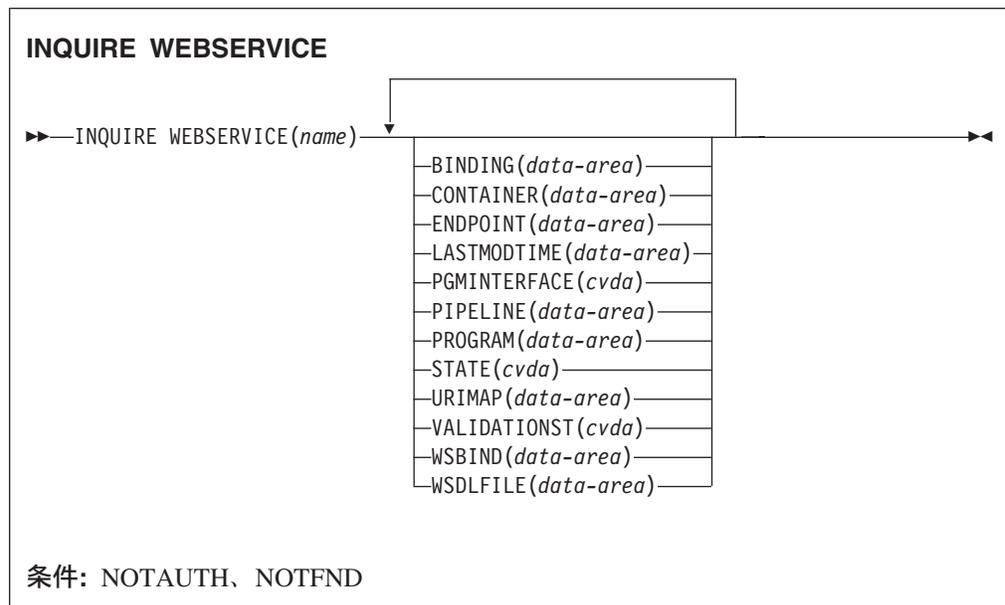
SET WORKREQUEST コマンドの WORKTYPE オプションに、以下の新しい値が追加されました。

SOAP

Web サービス要求のために実行される処理を指定します。

INQUIRE WEBSERVICE

INQUIRE WEBSERVICE コマンドは、インストール済み WEBSERVICE に関する情報を取得するために使用します。



ブラウズ

INQUIRE WEBSERVICE コマンドでブラウズ・オプション (START、NEXT、および END) を使用することにより、システムにインストールされているすべての WEBSERVICE をブラウズできます。

オプション

BINDING(*data-area*)

WEBSERVICE によって表現される WSDL バインディングを戻します。このバ

インデイングは、WSDL ファイルに出現する (潜在的に) 多数のバインディングのうちの一つです。この名前は、255 文字以内の長さにすることができます。

CONTAINER(*data-area*)

PGMINTERFACE が値 CHANNEL を戻す場合、使用されているコンテナの名前を戻します。この名前は、16 文字以内の長さにすることができます。

ENDPOINT(*data-area*)

リモートの WEBSERVICE のエンドポイント URI を戻します。これは、リモートの Web サービスの WSDL ファイルに指定されているエンドポイント URI です。CICS アプリケーション・プログラムがサービス・プロバイダーの場合、ENDPOINT は空になります。URI は、255 文字以内の長さにすることができます。

LASTMODTIME(*data-area*)

HFS 上に配置されている WSBind ファイルが最後に更新された時刻を、1900 年 1 月 1 日 00:00 からのミリ秒で戻します。これは、WEBSERVICE リソースがインストールまたは更新されたときに CICS が更新する読み取り専用の値です。最終更新時刻は、ピックアップ・ディレクトリー内の WSBind ファイルが最後に更新された後に CICS 自体がリフレッシュされたかどうかを判別するために使用できます。

- 動的にインストールされた WEBSERVICEs (CICS スキャン・メカニズムによりインストールされたもの) の場合、LASTMODTIME の値は、WEBSERVICE 定義が最後にインストールまたは更新されたときに WSBind 定義によって指し示されていた HFS ファイルのタイム・スタンプです。
- 静的にインストールされた WEBSERVICE (CSD から、または CREATE WEBSERVICE によりインストールされたもの) の場合、LASTMODTIME の値は、WEBSERVICE がインストールされたときに WEBSERVICE 定義によって指し示されていた WSBind HFS ファイルのタイム・スタンプです。

新たにインストールまたは更新された WEBSERVICE が完全に初期化される前に INQUIRE WEBSERVICE コマンドを発行すると、戻される LASTMODTIME 値はゼロになります。

値は、8 バイトのパック 10 進数の形式で戻されます。EXEC CICS FORMATTIME コマンドを使用して、必要な日時形式に LASTMODTIME 値を変換することができます。

PGMINTERFACE(*cvda*)

Web サービスをインプリメントしている CICS プログラムが、チャンネルでの入力を予期するか、または COMMAREA での入力を予期するかを示す CVDA を戻します。

CHANNEL

プログラムは、チャンネルでの入力を予期します。

COMMAREA

プログラムは、COMMAREA での入力を予期します。

PIPELINE(*data-area*)

WEBSERVICE をインストールする PIPELINE の名前 (すなわち、この

WEBSERVICE リソースを含む PIPELINE リソースの名前) を戻します。この名前は、8 文字以内の長さにすることができます。

PROGRAM(*data-area*)

Web サービスをインプリメントしている CICS プログラムの名前を戻します。この WEBSERVICE がリモートの Web サービスを表現している (すなわち、CICS はサービス・プロバイダーではない) 場合、PROGRAM は空になります。この名前は、8 文字以内の長さにすることができます。

STATE(*cvda*)

WEBSERVICE の状態を示す CVDA を戻します。

DISCARDING

この WEBSERVICE の DISCARD が進行中です。

INITING

Web サービス・バインディング・ファイルおよび WSDL ファイルをシェルフにコピー中です。

INSERVICE

シェルフ上の WSBIND ファイルのコピーの解決が成功し、WEBSERVICE が使用可能です。

UNUSABLE

シェルフへの WSBIND ファイルのコピーが失敗し、WEBSERVICE は使用不能です。

DISCARDING

この WEBSERVICE のために DISCARD コマンドが発行されました。WEBSERVICE は、廃棄される前に静止しています。新規作業は受け入れていませんが、現在実行中の作業は完了できます。

URIMAP(*data-area*)

この WEBSERVICE が URIMAP に関連付けられている場合、動的にインストールされた URIMAP の名前を戻します。WEBSERVICE が PIPELINE リソースの SCAN 機能を実行することによってインストールされなかった場合、あるいは WEBSERVICE がリモートの Web サービスを表現している場合は、URIMAP は空になります。この名前は、8 文字以内の長さにすることができます。

VALIDATIONST(*cvda*)

この WEBSERVICE に対して、SOAP メッセージの完全妥当性検査が現在使用可能に設定されているかどうかを示す CVDA を戻します。

VALIDATION

完全妥当性検査は使用可能です。

DISABLED

完全妥当性検査は使用不可です。

WEBSERVICE(*name*)

照会する WEBSERVICE の名前を指定します。この名前は、32 文字以内の長さにすることができます。

WSBIND(*data-area*)

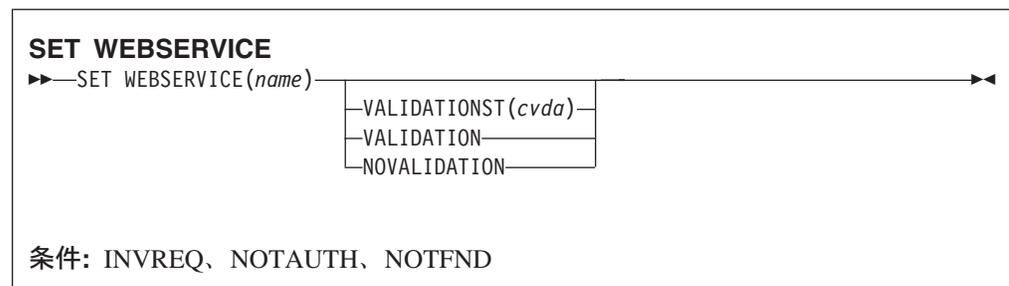
Web サービス・バインディング・ファイルの名前を戻します。この名前は、255 文字以内の長さにすることができます。

WSDLFILE(*data-area*)

WEBSERVICE リソースに関連付けられている Web サービス・バインディング・ファイルの名前を戻します。この名前は、255 文字以内の長さにすることができます。

SET WEBSERVICE

SET WEBSERVICE コマンドは、インストール済み WEBSERVICE の状況を変更するために使用します。



オプション

WEBSERVICE(*name*)

WEBSERVICE の名前を指定します。

VALIDATIONST(*cvda*)

WEBSERVICE のための完全妥当性検査が使用可能かどうかを指定します。CVDA 値は、以下のとおりです。

VALIDATION

完全妥当性検査は使用可能です。

NOVALIDATION

完全妥当性検査は使用不可です。

CEMT の変更点

CEMT DISCARD WEBSERVICE

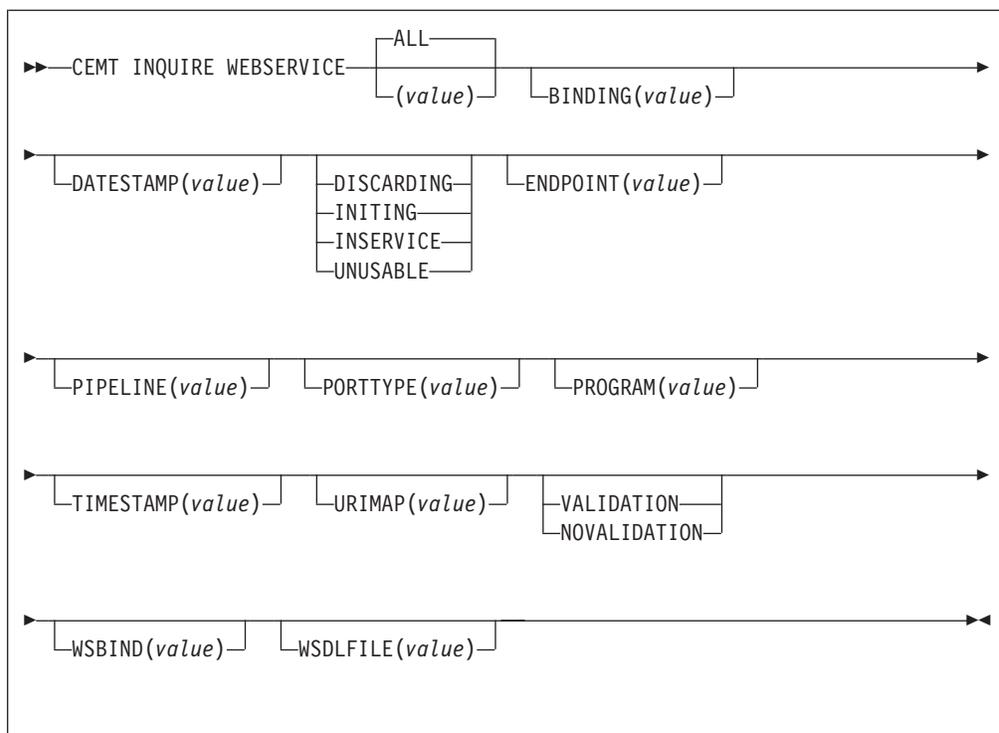
DISCARD WEBSERVICE コマンドは、CICS 領域から WEBSERVICE を除去するために使用します。

CEMT DISCARD PIPELINE

DISCARD PIPELINE は、CICS 領域から PIPELINE を除去するために使用します。

CEMT INQUIRE WEBSERVICE

INQUIRE WEBSERVICE コマンドは、インストール済み WEBSERVICE に関して以下の情報を表示するために使用します。



BINDING(value)

WEBSERVICE によって示される WSDL バインディングを表示します。このバインディングは、WSDL ファイルに (潜在的に) 多数出現するバインディングのうちの 1 つです。

DATESTAMP(value)

WEBSERVICE が最後に更新された日付を `yyyymmdd` の形式で表示します。これは、WEBSERVICE リソースがインストールまたは更新されたときに CICS が更新する読み取り専用の値です。

COMMAREA

プログラムは、COMMAREA で入力を受信することを予期します。

CONTAINER

プログラムは、CONTAINER で入力を受信することを予期します。

DISCARDING

この WEBSERVICE のために DISCARD 操作が進行中です。

ENDPOINT(value)

リモートの WEBSERVICE のエンドポイント URI を表示します。これは、リモートの Web サービスの WSDL ファイルに指定されているエンドポイント URI です。CICS アプリケーション・プログラムがサービス・プロバイダーの場合、ENDPOINT は空になります。

INITING

WSBind ファイルおよび WSDL ファイルをシェルフにコピー中です。

INSERVICE

シェルフ上の WSBind ファイルのコピーの解決が成功し、WEBSERVICE が使用可能です。

NOVALIDATION

完全妥当性検査は使用不可です。

PIPELINE(value)

WEBSERVICE をインストールする PIPELINE の名前 (すなわち、この WEBSERVICE リソースを含む PIPELINE リソースの名前) を表示します。

PORTTYPE(value)

この WEBSERVICE リソースによって示される WSDL 文書内の <portType> エレメントの内容を表示します。

PROGRAM(value)

CICS がサービス・プロバイダーであるときに、WSDL <portType> をインプリメントする CICS プログラムの名前を表示します。

TIMESTAMP(value)

WEBSERVICE が最後に更新された時刻を hh:mm:ss の形式で表示します。これは、WEBSERVICE リソースがインストールまたは更新されたときに CICS が更新する読み取り専用の値です。

URIMAP(value)

この WEBSERVICE に関連付けられた、動的にインストールされた URIMAP がある場合、その名前を表示します。WEBSERVICE が PIPELINE リソースの SCAN 機能を実行することによってインストールされなかった場合、あるいは WEBSERVICE がリモートの Web サービスを表現している場合は、URIMAP は空になります。

UNUSABLE

シェルフへの WSBind ファイルのコピーが失敗し、WEBSERVICE は使用不能です。

VALIDATION

完全妥当性検査は使用可能です。

WSBIND(value)

HFS 上の WSBind ファイルの 1 から 255 文字の完全修飾ファイル名を戻します。

WSDLFILE(value)

HFS 上の WSDL ファイルの 1 から 255 文字の完全修飾ファイル名を戻します。

CEMT INQUIRE WORKREQUEST

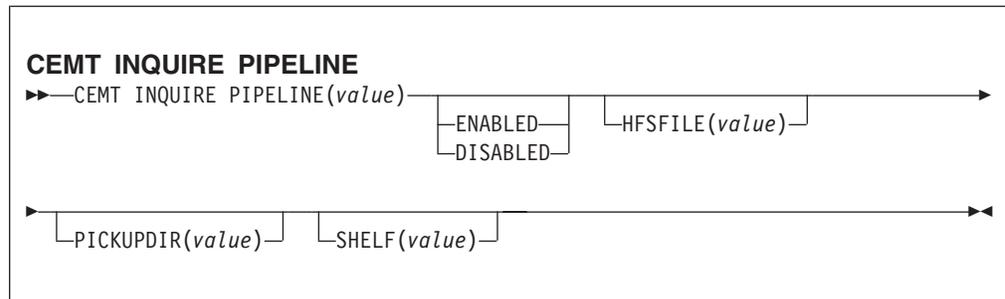
INQUIRE WORKREQUEST コマンドの WORKTYPE オプションに、以下の新しい値が追加されました。

PIPE

Web サービス要求のために実行される処理を指定します。

CEMT INQUIRE PIPELINE

インストール済み PIPELINE に関する情報を表示するには、INQUIRE PIPELINE コマンドを使用します。



PIPELINE(value)

照会する PIPELINE の 8 文字の名前を指定します。

HFSFILE(value)

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 HFS ファイルの名前を戻します。

ENABLESTATUS(cvda)

PIPELINE の状況を戻します。

ENABLED

この PIPELINE のインバウンド・サービス要求は通常どおりに処理されます。

DISABLED

この PIPELINE のインバウンド・サービス要求は拒否されます。

PICKUPDIR(value)

HFS 上のピックアップ・ディレクトリーの 1 から 255 文字の完全修飾名を戻します。

SHELF(value)

HFS 上のシェルフ・ディレクトリーの 1 から 255 文字の完全修飾名を戻します。

CEMT PERFORM STATISTICS

このコマンドは、以下の新しいオプションをサポートします。

PIPELINE

PIPELINE に関連する統計は、即時に SMF データ・セットに書き込まれます。統計には、HFS ファイルに関する情報が含まれます。

WEBSERVICE

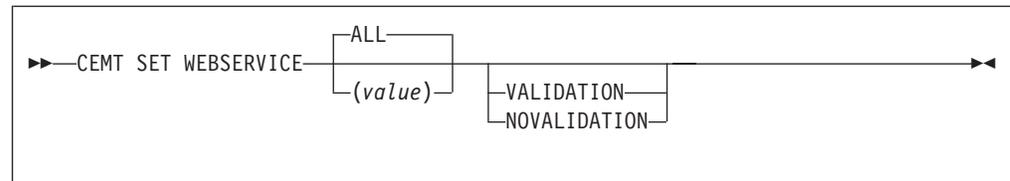
WEBSERVICE に関連する統計は、即時に SMF データ・セットに書き込まれます。

CEMT PERFORM PIPELINE

CEMT PERFORM PIPELINE コマンドは、PIPELINE の HFSFILE 属性で指定されたディレクトリーのスキャンを開始するために使用します。

CEMT SET WEBSERVICE

SET WEBSERVICE コマンドは、インストール済み WEBSERVICE の状況を変更するために使用します。



Validation

完全妥当性検査は使用可能です。

Novalidation

完全妥当性検査は使用不可です。

CEMT SET WORKREQUEST コマンド

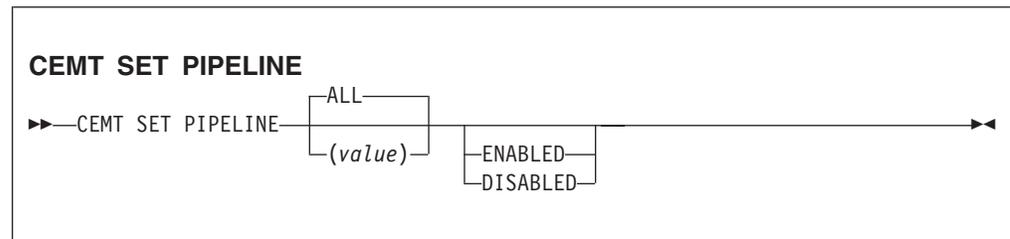
SET WORKREQUEST コマンドの WORKTYPE オプションに、以下の新しい値が追加されました。

PIPE

Web サービス要求のために実行される処理を指定します。

CEMT SET PIPELINE

SET PIPELINE コマンドは、インストール済み PIPELINE の状況を変更するために使用します。



PIPELINE(value)

この PIPELINE の 8 文字の名前を指定します。

ENABLED

この PIPELINE のインバウンド・サービス要求は通常どおりに処理されます。

DISABLED

この PIPELINE のインバウンド・サービス要求が拒否されることを指定します。

JCICS API の変更点

以下のコマンドに相当する JCICS コマンドがあります。

EXEC CICS FAULT ADD

EXEC CICS FAULT CREATE

EXEC CICS FAULT DELETE
EXEC CICS INVOKE WEBSERVICE

CICS 提供トランザクションの変更点

注: CEMT の変更点については、61 ページの『CEMT の変更点』で説明しています。

CETR の変更点

トランザクション CETR で、PI ドメインがサポートされるようになりました。

統計の変更点

以下の新しい DSECT が提供されます。

DFHPIPDS

PIPELINE リソースの統計を提供します。

DFHPIWDS

WEBSERVICE リソースの統計を提供します。

サンプル・プログラムの変更点

以下のプログラムが、PIPELINE および WEBSERVICE リソースをサポートするようになりました。

DFH0STAT (サンプルの統計プログラム)

DFH\$FORA (DB2[®] フォーマット・サンプル・プログラム (アセンブラー))

DFH\$FORP (DB2 フォーマット・サンプル・プログラム (PL/I))

DFH0FORC (DB2 フォーマット・サンプル・プログラム (COBOL))

DFH\$DB2T (DFH\$FORA、DFH\$FORP、および DFH0FORC 用の DB2 テーブル定義)

DFH\$SQLT (DB2 テーブル・ロード・ユーティリティの入力)

CICS ユーティリティの変更点

統計ユーティリティ・プログラム DFHSTUP

DFHSTUP は、『統計の変更点』で説明している統計に対する変更をサポートします。

DFHSTUP の SELECT TYPE および IGNORE TYPE 制御パラメーターでコーディングできるリソース・タイプに、以下が追加されました。

- WEBSERVICE
- PIPELINE

問題判別の変更点

CICS Transaction Server for z/OS バージョン 3 リリース 1 では、以下の新しいドメインが追加されました。

コンポーネント ID	説明
PI	パイプライン・マネージャー・ドメイン

新しいドメインをサポートするために、以下の変更が行われました。

- CICS トレース機能 (CETR、トレース関連のシステム初期設定パラメーター、およびトレース・ユーティリティー・プログラムを含む) は、示されたコンポーネント ID をサポートするようになりました。
- CICS システム・ダンプには、新しいドメインのための制御ブロックが含まれます。CICS 提供ダンプ出口ルーチンは、ダンプ出力に含める制御ブロックを指定するために、示されたコンポーネント ID の使用をサポートします。
- 上記に示したドメインにより発行された CICS メッセージでは、`DFHcomponent-ID` の形式のメッセージ接頭語を使用します。

セキュリティ

新しい SPI および CEMT コマンドのためのセキュリティ

新しい定義済み RACF® リソース名は、SPI および CEMT を使用して、以下のリソースへのアクセスを制御します。

PIPELINE
WEBSERVICE

新しいカテゴリー 1 トランザクション

以下の新しいトランザクションは CICS 内部で使用されるものです。ユーザー端末からは呼び出さないでください。セキュリティ上の目的のために、これらはカテゴリー 1 トランザクションに含まれています。

CPIH
CPIQ
CPIL
CWSQ

マイグレーション

このトピックでは、CICS における Web サービス が、CICS Transaction Server for z/OS バージョン 3 へのマイグレーションにどのように影響するかについて説明します。

既存の機能のマイグレーション

SOAP for CICS 機能を使用する場合は、いくつかの作業を行い、この機能を使用するアプリケーションをマイグレーションする必要があります。CICS Transaction

Server for z/OS バージョン 3 リリース 1 で提供される Web サービスのサポートは、この機能で提供されるサポートとは大幅に異なっています。

SOAP for CICS 機能は、かなりの範囲がユーザー作成コードに依存しているため、段階的マイグレーション・タスクを設定することができません。ただし、考慮の必要がある事項をいくつか示します。

- Web サービス・アシスタントを使用して SOAP メッセージを作成および構文解析することを検討してください。 そのようにすると決定した場合、アダプター内のコードの相当の部分を再利用できる可能性は低いため、既存のメッセージ・アダプターを破棄し、新規のラッパー・プログラムを設計してそれらを置き換えることをお勧めします。
- SOAP メッセージを使用するが、Web サービス・アシスタントを使用しないと決定した場合は、メッセージの作成と構文解析のために既存のコードを再利用できます。 ただし、CICS 提供 SOAP メッセージ・ハンドラーを使用するかどうかを検討する必要があります。それらのハンドラーが、SOAP 1.1 および SOAP 1.2 メッセージを処理するように設計されているためです。
- コンテナの使用を見直してください。 SOAP for CICS 機能では、BTS コンテナを使用しますが、CICS Transaction Server for z/OS バージョン 3 リリース 1 はチャンネル・コンテナを使用します。プログラムを見直し、この機能が必要とする BTS 関連コマンドを変更する必要があります。各コンテナの名前および使用法についても、ほとんどが変更されているため、見直す必要があります。
- パイプライン・プログラムで提供していた機能のマイグレーション方法を検討してください。 SOAP for CICS 機能内のパイプラインには、それぞれ指定された目的を持つ、固定数のユーザー作成プログラムが含まれます。これらのプログラムの一部によって提供される機能が、CICS 提供 SOAP メッセージ・ハンドラーによって、CICS Transaction Server for z/OS バージョン 3 リリース 1 で提供されるため、これらのプログラムがまったく不要となる場合があります。

一方、CICS Transaction Server for z/OS バージョン 3 リリース 1 では、パイプライン内で必要な数だけプログラムを定義することができます。そのため、パイプライン・プログラムによって実行される機能を、新しいフレームワークを利用するように再構築すべきであるかどうかを検討する必要があります。

どのような場合でも、パイプライン・プログラムが CICS と通信する方法およびパイプライン・プログラムが相互に通信する方法が変更されたため、これらのプログラムを見直し、新しい環境で再利用できるかどうかを調べる必要があります。

SOAP for CICS 機能では、すべてのサービス・プロバイダー・アプリケーションに対してパイプラインを 1 つのみ、およびすべてのサービス・リクエスターに対してパイプラインを 1 つのみ持つことができます。CICS Transaction Server for z/OS バージョン 3 リリース 1 では、多数の異なるパイプラインを構成することができます。そのため、あるアプリケーションを別のアプリケーションと区別するためにパイプライン・プログラムで提供するロジックを、CICS リソース定義によって置き換えることが可能となります。例えば、サービス・プロバイダーでは、URI に基づいてアプリケーションを区別するコードは、適切な URIMAP リソースのセットで置き換えることができます。

共存

SOAP for CICS 機能を使用している場合は、引き続き使用可能です。この機能は、CICS における Web サービスとは関係なく、引き続き CICS Transaction Server for z/OS バージョン 3 リリース 1 で完全にサポートされています。SOAP for CICS 機能は、CICS TS for z/OS バージョン 3.1 での Web サービスのサポートと相互運用できます。この機能は、サービス・リクエスターになることも、サービス・プロバイダーになることもできます。

CICSplex SM サポート

CICS で Web サービスをサポートするため、CICSplex SM ビューおよびリソース・テーブルに対していくつかの変更が行われました。

CICSplex SM アプリケーション・プログラミング・インターフェースの変更点

以下のリソース・テーブルが新しく導入されました。

- 『PIPEDEF リソース・テーブル』
- 『PIPELINE リソース・テーブル』
- 70 ページの 『WEBSVDEF リソース・テーブル』
- 70 ページの 『WEBSERV リソース・テーブル』

PIPEDEF リソース・テーブル

PIPEDEF リソース・テーブルには、以下の RDO 属性が含まれています。

PIPELINE

パイプライン定義名

STATUS

PIPELINE を ENABLED (デフォルト) の状態でインストールするか、DISABLED の状態でインストールするかを示す CVDA 値を示します。

CONFIGFILE

HFS 上のこのパイプライン用の構成ファイルの 255 文字から成る完全修飾名です。

SHELF

WSBind ディレクトリー (シェルフ) の 255 文字の完全修飾名です。

WSDIR

HFS 上の WSBind (ピックアップ) ディレクトリーの 255 文字の名前です。

PIPELINE リソース・テーブル

PIPELINE リソース・テーブルには、以下の SPI 属性が含まれています。

PIPELINE

パイプライン定義名

ENABLESTATUS

パイプラインの ENABLE 状況を示す CVDA 値です。

PIPEUSECOUNT

パイプラインの使用回数

CONFIGFILE

HFS 上のこのパイプライン用の構成ファイルの 255 文字から成る完全修飾名です。

SHELF

WSBind ファイル用のディレクトリー (シェルフ) の 255 文字の名前です。

WSDIR

HFS 上の WSBind (ピックアップ) ディレクトリーの 255 文字の名前です。

WEBSVDEF リソース・テーブル

WEBSVDEF リソース・テーブルには、以下の RDO 属性が含まれています。

WEBSERVICE

WEBSERVICE 定義名

PIPELINE

この Web サービスがインストールされるパイプラインです。

WSBIND

HFS 上の WSBind ファイルの 255 文字の完全修飾名です。

WSDLFILE

HFS 上の WSDL ファイルの 255 文字の完全修飾名です。

VALIDATION

WSDL SOAP メッセージの完全な妥当性検査を実行するかどうかを示す CVDA 値 (YES または NO) を表示します。

WEBSERV リソース・テーブル

WEBSERV リソース・テーブルには、以下の SPI 属性が含まれています。

WEBSERVICE

WEBSERVICE 名

PIPELINE

この Web サービスがインストールされるパイプラインです。

WSBIND

HFS 上の WSBind ファイルの 255 文字の完全修飾名です。

WSDLFILE

HFS 上の WSDL ファイルの 255 文字の完全修飾名です。

URIMAP

この Web サービスに関連付けられている、動的にインストールされた URIMAP です。

BINDING

この Web サービスが表す WSDL バインディングです。

ENDPOINT

リモートの Web サービスのエンドポイント URI です。

PROGRAM

この Web サービスをインプリメントする CICS アプリケーション・プログラムの 8 文字の名前です。

PGMINTERFACE

指定された PROGRAM が入力を期待する場所を示す CVDA 値 (CHANNEL または COMMAREA) です。

CONTAINER

PGMINTERFACE が CHANNEL の場合に使用されるコンテナの名前です。

VALIDATIONST

WSDL SOAP メッセージの完全な妥当性検査を実行するかどうかを示す CVDA 値 (VALIDATION または NOVALIDATION) です。

LASTMODTIME

HFS 上に配置されている WSBind ファイルが最後に更新された時刻です。

STATE

Web サービスの状態を示す CVDA 値 (DISCARDING、INITING、INSERVICE、UPDATING、または UNUSABLE) です。

WEBUSECOUNT

Web サービスの使用回数です。

変更されたリソース・テーブル

以下のリソース・テーブルが変更されました。

- 『WORKREQ』

WORKREQ

WORKREQ リソース・テーブルに、以下の属性が含まれるようになりました。

WORKTYPE

実行される処理のタイプを示します。使用可能なタイプは CVDA IIOP タイプおよび SOAP タイプのみです。

CICSplex SM Web ユーザー・インターフェースの変更点

新規 WUI ビュー

以下の WUI ビューが導入されました。

- 72 ページの『パイプライン定義ビュー』
- 72 ページの『パイプライン・ビュー』
- 72 ページの『Web サービス定義ビュー』
- 73 ページの『Web サービス・ビュー』

パイプライン定義ビュー

「パイプライン定義」という、新しい PIPEDEF リソース・テーブルに関連した定義ビュー・セットが新しく導入されました。この表形式のビューの名前は、EYUSTARTPIPEDEF.TABULAR です。既存の EYUSTARTADMRES メニューが拡張され、このビューを含むようになりました。

「パイプライン定義」ビューをオープンするには、以下の手順を実行します。

1. メインメニューから「管理ビュー」をクリックします。
2. 「ベーシック CICS リソース管理ビュー」をクリックします (あるいは、「完全に機能しているビジネス・アプリケーション・サービス (BAS) 管理ビュー」をクリックします)。
3. 「CICS リソース定義」をクリックします。
4. スクロールダウンして、「パイプライン定義」をクリックします。

「パイプライン定義」ビューが表示されます。このビューには、以下の 5 つのアクション・ボタンがあります。

- 作成
- 更新
- 除去
- インストール
- リソース・グループへの追加

フィールドの詳細については、69 ページの『PIPEDEF リソース・テーブル』に記載されている PIPEDEF リソース・テーブルの属性を参照してください。

パイプライン・ビュー

「パイプライン」という、新しい PIPELINE リソース・テーブルに関連したビューが新しく導入されました。この表形式のビューの名前は EYUSTARTPIPELINE.TABULAR です。既存の EYUSTARTTCPIPS メニューが拡張されてこのビューを含むようになりました。

「パイプライン」ビューをオープンするには、以下の手順を実行します。

1. メインメニューから「CICS 操作ビュー」をクリックします。
2. スクロールダウンして、「TCP/IP サービス操作ビュー」をクリックします。
3. 「パイプライン」をクリックします。

「パイプライン」ビューが表示されます。

フィールドの詳細については、69 ページの『PIPELINE リソース・テーブル』に記載されている PIPELINE リソース・テーブルの属性の詳細を参照してください。

Web サービス定義ビュー

「Web サービス定義」という、新しい WEBSVDEF リソース・テーブルに関連した定義ビュー・セットが新しく導入されました。この表形式ビューのビュー名は、EYUSTARTWEBSVDEF.TABULAR です。既存の EYUSTARTADMRES メニューが拡張され、このビューを含むようになりました。

「Web サービス定義」ビューをオープンするには、以下の手順を実行します。

1. メインメニューから「管理ビュー」をクリックします。
2. 「ベーシック CICS リソース管理ビュー」をクリックします (あるいは、「完全に機能しているビジネス・アプリケーション・サービス (BAS) 管理ビュー」をクリックします)。
3. 「CICS リソース定義」をクリックします。
4. スクロールダウンして、「Web サービス定義」をクリックします。

「Web サービス定義」ビューが表示されます。このビューには以下の 5 つのアクション・ボタンがあります。

- 作成
- 更新
- 除去
- インストール
- リソース・グループへの追加

フィールドの詳細については、70 ページの『WEBSVDEF リソース・テーブル』に記載されている WEBSVDEF リソース・テーブルの属性を参照してください。

Web サービス・ビュー

「Web サービス」という、新しい WEBSERV リソース・テーブルに関連した表形式のビューが新しく導入されました。この表形式ビューのビュー名は EYUSTARTWEBSERV.TABULAR です。既存の EYUSTARTTCPIPS メニューが拡張されて、このビューを含むようになりました。

「Web サービス」ビューをオープンするには、以下の手順を実行します。

1. メインメニューから「CICS 操作ビュー」をクリックします。
2. スクロールダウンして、「TCP/IP サービス操作ビュー」をクリックします。
3. 「Web サービス」をクリックします。

「Web サービス」ビューが表示されます。

フィールドの詳細については、70 ページの『WEBSERV リソース・テーブル』に記載されている WEBSERV リソース・テーブルの属性を参照してください。

変更された WUI ビュー

以下の WUI ビューが変更されました。

- 『処理要求ビュー』

処理要求ビュー

タスク操作ビュー内部の「処理要求」ビューに、以下の属性が追加されました。

WORKTYPE

実行される処理のタイプを示します。使用可能なタイプは、IIOP および SOAP のみです。

第 3 章 CICS アプリケーションからの HTTP クライアント要求のサポート

EXEC CICS コマンドを使用して、HTTP クライアントとしての CICS を介して CICS アプリケーションからのアウトバウンド要求を作成できるようになりました。

HTTP クライアントとして動作する CICS の機能は、CICS Web サポートに完全に統合されました。CICS アプリケーション・プログラムで EXEC CICS コマンドを使用して、サーバーとの HTTP 接続をオープンしたり、要求を行ったり、アプリケーション・プログラムで処理するために応答を受信することができます。このためにいくつかの新しい EXEC CICS コマンドが導入され、一部の既存のコマンドには、アウトバウンド・オプションおよびインバウンド・オプションが提供されました。

CICS アプリケーションからの HTTP クライアント要求のサポートの利点

HTTP クライアントとしての CICS の EXEC CICS コマンドは、以下の目的で CICS アプリケーション・プログラムを作成できることを意味します。

- 企業間通信用の一般的なプロトコルを使用する。
- HTTP プロトコルを使用してハードウェアまたはソフトウェアを制御する (例えば、この方法でプリンターを制御できる場合があります)。
- 情報 (例えば、株価) のための項目を提供する HTTP アプリケーションにアクセスしてこの情報を取得し、アプリケーションで使用する。

CICS Web サポート・プロセスのグローバル・ユーザー出口ルーチンを使用すると、CICS からのアウトバウンド HTTP 要求のためにプロキシ・サーバーの使用を指定したり、セキュリティー・ポリシーを適用することができます。

要件

この機能をサポートするための特別なハードウェア要件およびソフトウェア要件はありません。

関連情報

377 ページの『第 27 章 CICS 稼働環境』

HTTP クライアントとしての CICS に対する HTTP 要求と応答の処理

HTTP クライアントとしての CICS の場合、CICS は Web クライアントであり、HTTP サーバーと通信します。ユーザー作成のアプリケーション・プログラムは、要求を CICS を介して HTTP サーバーに送信し、HTTP サーバーから応答を受信します。CICS は、このサーバーとの持続接続を維持します。セッション・トークンは、接続を識別するため、アプリケーション・プログラムによって発行されたコマンドで使用されます。

HTTP 要求を実行して応答を受信するアプリケーション・プログラムは、EXEC CICS WEB API コマンドを使用してサーバーとの対話を明示的に指示する必要があります。

ります。Web 対応アプリケーション・プログラムを使用すると、HTTP 要求を実行し、その結果を処理して Web 対応以外のアプリケーションに情報を提供することができます。

HTTP クライアント要求の場合、CICS は、要求と応答のコード・ページ変換を実行できます。しかし、これとは別に、CICS は、HTTP クライアントとしての CICS に送信されるデータを分析または変換する特別な処理を実行するわけではありません（この特別な処理は、HTTP サーバーとしての CICS に対しては実行されます）。HTTP 要求を開始するアプリケーション・プログラムは、この要求に対する応答時に CICS がサーバーから受信する内容をすべて処理するように設計されている必要があります。この内容としては、エラー応答、別の URL へのリダイレクト、埋め込みハイパーテキスト・リンク、HTML フォーム、イメージ・ソース、またはアプリケーション・プログラムからのアクションを要求するその他の項目などが考えられます。

HTTP クライアントとしての CICS に対する処理は、以下のように実行されます。

1. **アプリケーション・プログラムは、CICS を介して HTTP サーバーとの接続を開始します。** アプリケーション・プログラムは、WEB OPEN コマンドを発行することによってこの操作を実行します。作成した URIMAP リソース定義を参照すると、この接続のスキームおよびホスト名を指定できます。あるいは、アプリケーション・プログラムがこの情報を提供する場合もあります。
2. **CICS は、サーバーとの接続を確立します。** アプリケーション・プログラムによって提供された情報を使用することにより、CICS は、ソケット上で TCP/IP 接続を開始し、サーバーと通信します。TCP/IP 接続が確立されると、CICS はセッション・トークンをアプリケーション・プログラムに戻し、この接続を一意的に識別します。このセッション・トークンは、この接続に関してアプリケーション・プログラムが発行した残りのコマンドすべてに使用されます。
3. **アプリケーション・プログラムは、その要求時に HTTP ヘッダーを書き込む場合があります。** ユーザー作成の HTTP ヘッダーは、WEB WRITE HTTPHEADER コマンドを使用すると作成でき、送信可能な状態で格納できます。
4. **アプリケーション・プログラムは、要求行のコンポーネントを指定します。** 要求メソッド、パス情報、および照会ストリングは、WEB SEND コマンドまたは WEB CONVERSE コマンドを使用して指定します。要求の HTTP 版は、CICS によって提供されます。
5. **アプリケーション・プログラムは、その要求時にエンティティ本体を生成する場合があります。** 要求の内容は、WEB SEND コマンドまたは WEB CONVERSE コマンドで指定します。
6. **アプリケーション・プログラムが要求の送信を開始します。** アプリケーション・プログラムが WEB SEND コマンドまたは WEB CONVERSE コマンドを発行すると、要求は CICS に渡され、セッション・トークンによって指定された接続を介した送信に備えます。
7. **CICS は、必要な HTTP ヘッダーをいくつか生成してこれを要求に追加し、要求をサーバーに送信します。**
8. **サーバーは要求を受信して処理し、応答を戻します。** CICS は、応答をアプリケーション・プログラムに渡します。

9. アプリケーション・プログラムは、応答の内容を確認します。WEB READ HTTPHEADER コマンド、つまり HTTP ヘッダーのブラウズ・コマンドを使用すると、応答のヘッダーを確認できます。WEB RECEIVE コマンドまたは WEB CONVERSE コマンドを実行すると、アプリケーション・プログラムで処理できる応答の本文 (存在する場合) や、応答の状況コードおよび状況テキストを受信します。
10. アプリケーション・プログラムは、追加の要求および応答を開始できます。サーバーが持続接続をサポートしている場合、セッション・トークンによって示されている接続は、その後の要求に備えて接続状態が維持されます。
11. アプリケーション・プログラムは、サーバーへの接続の終了処理を開始します。すべての要求および応答が完了すると、アプリケーション・プログラムは WEB CLOSE コマンドを発行し、CICS は、TCP/IP 接続の終了時に終了します。アプリケーション・プログラムが WEB CLOSE コマンドを発行しなかった場合、接続はタスクの終了時に終了します。

この処理中、コード・ページ変換が必要になるのは、通常、メッセージが CICS 環境に出入りするときであるため、CICS Web サポート処理およびユーザー作成アプリケーション (通常は EBCDIC エンコードを使用) は、HTTP サーバー (通常は ASCII エンコードを使用) と通信できます。

セッション・トークン

セッション・トークンとは、HTTP クライアントとしての CICS と HTTP サーバー間の接続を一意的に識別する 8 バイトのバイナリー値のことです。接続ごとにセッション・トークンを使用すると、CICS Web サポートは、サーバーへの複数の接続を異なるタスクごとに管理でき、またアプリケーション・プログラムは複数の接続を制御できます。

接続は、ユーザー・アプリケーション・プログラムによって発行された WEB OPEN コマンドに回答して開始されます。セッション・トークンは、WEB OPEN コマンドが正常に実行されると戻され、対象の接続に関するアプリケーション・プログラムによって発行されたすべての EXEC CICS WEB コマンドで使用されます。この接続を使用すると、ユーザー・アプリケーション・プログラムは、HTTP クライアント要求をサーバーに対して実行でき、サーバーから応答を受信できます。この接続では、アプリケーション・プログラムまたはサーバーが接続の終了を選択するまで、要求と応答のやり取りを複数回持続できます。112 ページの『CICS Web サポートでの持続接続の処理方法』には、CICS Web サポートが持続接続を処理する仕組みや、持続接続が終了する仕組みの詳細が説明されています。

HTTP クライアントとしての CICS を対象として使用され、この目的のためにセッション・トークンを必要とするコマンドは、次のとおりです。

- WEB SEND コマンド。このコマンドは、HTTP クライアント要求を作成して、それをサーバーに送信します。
- WEB RECEIVE コマンド。このコマンドは、サーバーから応答を受信します。
- WEB CONVERSE コマンド。このコマンドは、WEB SEND コマンドと WEB RECEIVE コマンドの処理を組み合わせます。

- HTTP ヘッダーを操作する以下のコマンド: WEB READ HTTPHEADER、WEB WRITE HTTPHEADER、WEB STARTBROWSE HTTPHEADER、WEB READNEXT HTTPHEADER、WEB ENDBROWSE HTTPHEADER。
- WEB CLOSE コマンド。このコマンドは、接続を終了します。

サーバーが接続を終了すると、アプリケーション・プログラムは、それ以降、この接続を使用して要求を送信できなくなりますが、サーバーが接続を終了する前に送信した応答は読み取ることができます。セッション・トークンは、対象のデータにアクセスするためのコマンドで使用する場合には、アプリケーションが WEB CLOSE コマンドを発行するまで有効な状態が維持されます。WEB CLOSE コマンドが発行されると、この接続に適用されているセッション・トークンは無効になります。アプリケーション・プログラムが WEB CLOSE コマンドを発行しなかった場合、接続はタスクの終了時に終了します。

CICS 外部インターフェースの変更点

リソース定義の変更点

URIMAP 定義は、HTTP クライアントとしての CICS を通して HTTP 要求を行うための便利な方法として使用できます。

HTTP クライアントとしての CICS 用の URIMAP 定義

URIMAP 定義は、HTTP クライアント要求を行うアプリケーション・プログラムの URL や証明書ラベルなどの情報の指定を回避するために使用されます。URIMAP 定義の属性は、134 ページの『リソース定義の変更点』でリストします。HTTP クライアントとしての CICS の URIMAP 定義は、USAGE(CLIENT) とともに定義され、SCHEME、HOST、PATH、CERTIFICATE、および CIPHERS 属性を指定できます。アプリケーション・プログラムは、WEB OPEN および WEB SEND コマンドで URIMAP 定義を指定すれば、その定義からの関連情報を使用できます。

URIMAP 定義は、HTTP サーバーとしての CICS、および Web サービスにも使用されます。

モニター管理テーブル DFHMCT の変更点

パフォーマンス・クラス・レコードの DFHWEBB グループの新規モニター・フィールド 331 から 338 がモニター管理テーブルに追加されました。これらのフィールドは、HTTP クライアントとしての CICS のアクションをモニターします。

アプリケーション・プログラミング・インターフェースの変更点 (HTTP クライアント要求)

新しいコマンドと変更されたコマンド

HTTP クライアントとしての CICS に対して、以下の新しい EXEC CICS WEB コマンドが提供されています。

- EXEC CICS WEB OPEN
- EXEC CICS WEB CONVERSE

- EXEC CICS WEB CLOSE

HTTP クライアントとしての CICS で使用される場合、以下の EXEC CICS WEB コマンドにはさまざまな新規オプションがあります。

- EXEC CICS WEB SEND
- EXEC CICS WEB RECEIVE

以下の新規 EXEC CICS コマンドは、HTTP クライアントとしての CICS および HTTP サーバーとしての CICS の両方で使用できます。これらについては、145 ページの『アプリケーション・プログラミング・インターフェースの変更点 (汎用 CICS Web サポート拡張)』で説明します。

- EXEC CICS WEB PARSE URL
- EXEC CICS CONVERTTIME

残りのほとんどの EXEC CICS WEB コマンドで使用可能なオプションに対して変更が行われました。変更されたコマンドは、HTTP クライアントとしての CICS および HTTP サーバーとしての CICS の両方で使用できます。以下のコマンドの変更点については、このセクションで説明します。

- EXEC CICS WEB WRITE HTTPHEADER
- EXEC CICS WEB READ HTTPHEADER
- EXEC CICS WEB STARTBROWSE HTTPHEADER
- EXEC CICS WEB READNEXT HTTPHEADER
- EXEC CICS WEB ENDBROWSE HTTPHEADER

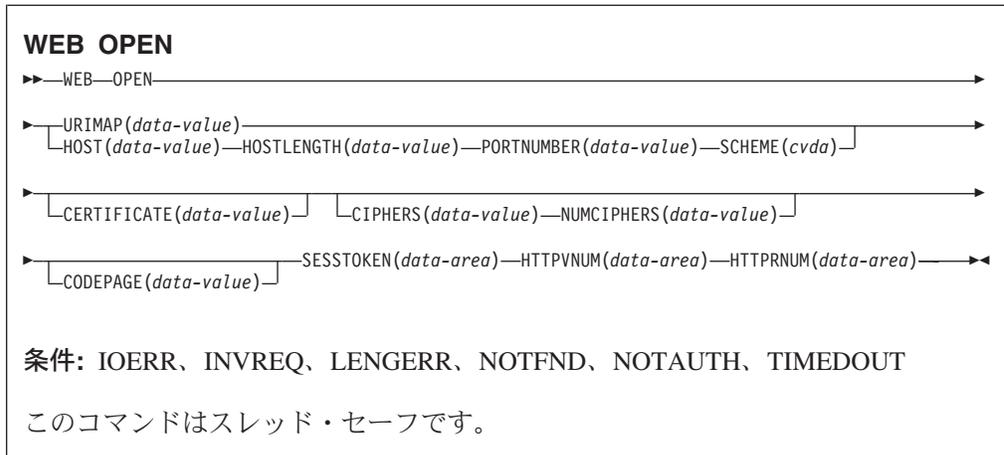
以下のコマンドの変更点については、145 ページの『アプリケーション・プログラミング・インターフェースの変更点 (汎用 CICS Web サポート拡張)』で説明します。

- EXEC CICS WEB EXTRACT
- EXEC CICS FORMATTIME

WEB FORMFIELD コマンドおよび WEB RETRIEVE コマンドは、HTTP クライアントとしての CICS では使用できません。

WEB OPEN

HTTP クライアントとしての CICS のサーバーへの接続を開始します。



説明

WEB OPEN を使用すると、アプリケーション・プログラムが CICS Web サポートを通して、インターネット上の HTTP サーバーで指定されたホストとの接続を開始することができます。ホスト名とスキームは、事前に設定された URIMAP 定義から使用することができます。この定義には、要求に対するデフォルトのパスも指定されます。

接続が開始されると、アプリケーション・プログラムは、サーバーに対する HTTP クライアント要求を作成し、サーバーから応答を受信することができます。接続が開始されると、CICS はサーバーの HTTP バージョンを照会し、この情報を以降の通信に使用します。CICS は、ユーザーが HTTP ヘッダーの記述やチャンク化された情報の送信を計画しているかどうかを確認するために、HTTP バージョン情報をアプリケーション・プログラムにも戻します。

WEB OPEN コマンドは、XWBOPEN ユーザー出口を動作させます。このユーザー出口は、必要であれば、プロキシ・サーバーを経由したサーバーとの接続を可能にします。

オプション

CERTIFICATE(*data-value*)

SSL ハンドシェイク時に 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。証明書のラベルは、32 文字以内の英数字で指定できます。このオプションが関連するのは、SCHEME(HTTPS) を指定した場合に限られます。SCHEME(HTTPS) を指定し、CERTIFICATE オプションを省略した場合は、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内の鍵リングに保管する必要があります。

CIPHERS(*data-value*)

28 個以内の 2 桁の暗号スイート・コードのリストとして解釈される 56 個以内の 16 進数字のストリングを指定します。接続で SSL がアクティブの場合、暗号スイート・コードが使用されるため、このオプションは、SCHEME(HTTPS) が指定されている場合にのみ関係します。暗号スイート・コードは、この接続で使用される暗号化の方式を示します。

NUMCIPHERS オプションを使用して、リスト中の暗号スイート・コードの番号を指定します。実際に使用されるコードは、ENCRYPTION システム初期設定パラメーターに指定された暗号化レベルにより異なります。アクティブ暗号化レベルのデフォルト・リストにない暗号コードが指定されると、そのコードは無視されます。

CIPHERS オプションが必要ない場合には、URIMAP オプションを指定すると、既存の URIMAP 定義を基に、この情報を直接使用できます。CIPHERS オプションが指定されていると、このオプションで指定するコードによって URIMAP 定義の設定が無効になります。

CODEPAGE(*data-value*)

コード・ページを指定します。通常の場合、アプリケーション・プログラムに適した EBCDIC を指定します。コード・ページ名は、8 文字以内の英数字で指定できます。デフォルトは、LOCALCCSID システム初期設定パラメーターに指定されているように、ローカル CICS 領域のデフォルトのコード・ページです。コード・ページは、この接続の存続時間中に適用されます。サーバーが HTTP 要求に対する応答を戻すと、変換が要求されている場合 (デフォルトの設定)、CICS が要求本体をこのコード・ページに変換してからアプリケーションに渡します。

HOST(*data-value*)

接続先にするサーバーのホスト名を指定します。

ホスト名として IPv4 アドレスを使用できますが、IPv6 アドレスはサポートされていません。

ポート番号が必要な場合、ホスト名と組み合わせて指定することはしないでください。ポート番号を指定するには、PORTNUMBER オプションを使用します。

HOSTLENGTH(*data-value*)

ホスト名の長さをフルワード・バイナリー値で指定します。

HTTPRNUM(*data-area*)

サーバーの HTTP バージョンのリリース番号をハーフワード・バイナリー値で戻します。(HTTPVNUM はバージョン番号を戻します。) 例えば、サーバーが HTTP/1.0 レベルであれば、HTTPRNUM は 0 を戻します。

HTTPVNUM(*data-area*)

サーバーの HTTP バージョンのバージョン番号をハーフワード・バイナリー値で戻します。(HTTPRNUM はリリース番号を戻します。) 例えば、サーバーが HTTP/1.0 レベルであれば、HTTPVNUM は 1 を戻します。

CICS は、リモート・サーバーとの接続を開始すると、HTTP バージョン情報を取得します。リモート・サーバーが HTTP バージョン情報を通知しない場合、CICS は、リモート・サーバーが HTTP/1.0 レベルに位置すると想定します。

アプリケーション・プログラムが HTTP/1.0 レベルのサーバーに適さない可能性のある HTTP ヘッダーを書き込む場合、またはチャンク化されたメッセージをサーバーに送信する予定の場合 (チャンク化されたメッセージは HTTP/1.0 レベルのサーバーは受信できない) は、アプリケーション・プログラムは HTTP のバージョン情報も調べる必要があります。

注: CICS では、HTTP/1.0 レベルに達しないサーバーに対して、特殊な方法は一切実行されません。CICS は、これらのサーバーのレベルが HTTP/1.0 であるかのように動作し、HTTP バージョンとして HTTP/1.0 を戻します。

NUMCIPHERS(*data-value*)

CIPHERS オプションに指定した暗号スイート・コードの番号をハーフワード・バイナリー値で指定します。

PORTNUMBER(*data-value*)

ポート番号をフルワード・バイナリー値で指定します。ポート番号が指定したスキームの**デフォルトでない**場合にのみ、ポート番号を指定する必要があります。HTTP の場合、デフォルトのポート番号は 80 であり、HTTPS の場合、デフォルトのポート番号は 443 です。

SCHEME(*cvda*)

サーバーへの接続に使用するスキームを指定します。SSL の有無は関係ありません。CVDA 値は、以下のとおりです。

HTTP SSL を備えていない HTTP プロトコルのことです。

HTTPS

SSL を備えた HTTP である HTTPS プロトコルのことです。HTTPS を使用する場合、SSL で、CICS アドレス・スペースが使用可能である必要があります。

SESSTOKEN(*data-area*)

CICS とサーバー間のこの接続を一意に識別する 8 バイトのバイナリー値であるセッション・トークンを戻します。セッション・トークンは、接続が正常に開始されると戻ります。このセッション・トークンを、この接続に関連するすべての CICS WEB コマンドで使用する必要があります。77 ページの『セッション・トークン』には、セッション・トークンの使用法が説明されています。

URIMAP(*data-value*)

以下の情報を提供する URIMAP 定義の名前 (8 文字以内) を指定します。

- サーバーへの接続に使用するスキーム。
- 接続先サーバーのホスト名。
- ポート番号 (必要な場合)。
- URI のパス部分。アクセス先にするサーバーのリソースを表します。このパスは、この接続に関連する WEB SEND コマンドまたは WEB CONVERSE コマンドのデフォルトのパスになりますが、WEB SEND コマンドまたは WEB CONVERSE コマンドで別のパスを指定することにより、このパスを指定変更することができます。
- 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 (必要な場合)。
- 接続で使用可能な暗号スイート・コード。

URIMAP オプションを指定する場合は、CERTIFICATE、HOST、HOSTLENGTH、PORTNUMBER、PORTLENGTH、または SCHEME オプションを指定しないでください。CIPHERS オプションおよび NUMCIPHERS オプションは、省略することも、コマンドで指定することもできます。指定される

と、URIMAP 定義の設定は無効になります。URIMAP 定義は、(USAGE(CLIENT) が指定された) HTTP クライアントとしての CICS に指定する必要があります。

WEB CLOSE

HTTP クライアントとしての CICS のサーバーへの接続を終了します。

WEB CLOSE

▶▶—WEB—CLOSE—SESSTOKEN(*data-value*)————▶▶

条件: NOTOPEN

このコマンドはスレッド・セーフです。

説明

WEB CLOSE を使用すると、アプリケーション・プログラムによってサーバーとの接続を終了できます。セッション・トークンは、終了の対象となる接続を示しています。接続が終了すると、このセッションに適用されているセッション・トークンは無効になり、使用できなくなります。サーバーからの応答を受信し、応答の HTTP ヘッダーを読み取るには、セッション・トークンが必須のため、リモート・サーバーとの対話およびサーバーが送信した応答との対話がすべて完了するまで、WEB CLOSE コマンドは発行しないでください。このコマンドにより、接続に関連する CICS リソースが開放されます。

WEB CLOSE コマンドでは、接続の終了の指示が CICS からサーバーに通知されることはありません。このコマンドの機能により、CICS はクライアント・サイドの接続のみを終了します。この接続を使用して行う最後の要求時に、WEB SEND コマンドまたは WEB CONVERSE コマンドで CLOSESTATUS(CLOSE) オプションを指定する必要があります。このオプションを指定すると、CICS が要求に Connection: close ヘッダーを書き込みます。HTTP/1.0 レベルのサーバーの場合は、Connection: Keep-Alive ヘッダーを省略します。ヘッダー内の情報は、サーバーが最終応答を送信した後即時に (ユーザーがタイムアウト前にさらに要求を送信するかどうかの確認のため待機するのではなく)、ユーザーとの接続を終了できることを意味します。

接続は、WEB CLOSE コマンドが発行される前に、サーバーの要求により終了する可能性もあります。サーバーが接続の終了を要求したかどうかをテストする必要がある場合は、WEB READ HTTPHEADER コマンドを使用して、サーバーからの最後のメッセージに存在する Connection: close ヘッダーを探します。

サーバーが接続の終了を要求した場合にも、その接続に関連するデータは、WEB CLOSE コマンドが発行されるまで CICS 内部で使用可能になっています。使用可能なデータには、サーバーから受け取った最新メッセージ、および接続のオープンに使用するパラメーター (サーバーのスキームまたはホスト名など) が含まれます。サーバーが接続を終了させると、アプリケーション・プログラムは以下の処理を実行できなくなります。

- WEB SEND または WEB CONVERSE コマンドを使用した、該当接続上での追加要求の送信
- WEB WRITE HTTPHEADER コマンドを使用した HTTP ヘッダーの書き込み

ただし、アプリケーション・プログラムは以下の処理を実行できます。

- WEB RECEIVE コマンドを使用した応答の受信
- WEB READ HTTPHEADER コマンドおよび HTTP ヘッダー・ブラウズ・コマンドを使用した HTTP ヘッダーの検査
- WEB EXTRACT コマンドを使用した接続情報の抽出

WEB CLOSE コマンドを発行すると、接続に関連するデータは消去されます。

アプリケーション・プログラムによって WEB CLOSE コマンドが発行されていない場合、CICS はタスクの終了時に接続に関連するデータを消去し、接続が終了されていなければそれを終了します。

オプション

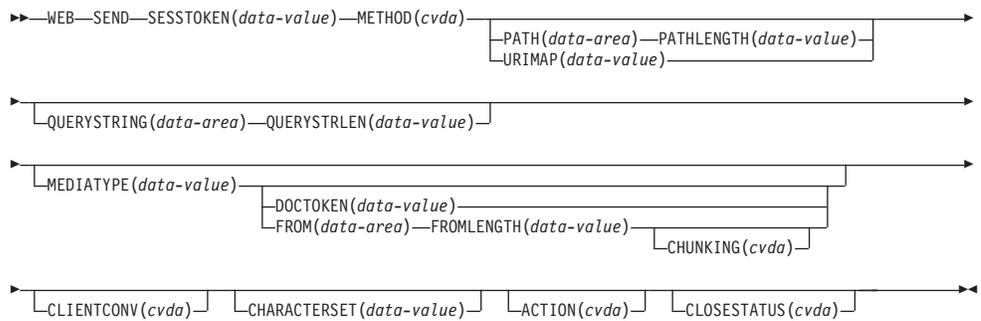
SESSTOKEN(*data-value*)

CICS とサーバー間の接続を一意に識別する 8 バイトのバイナリー値であるセッション・トークンを指定します。この値は、WEB OPEN コマンドによって、HTTP クライアントとしての CICS に戻されます。セッション・トークンで識別された接続に対して WEB CLOSE コマンドを発行すると、CICS がその接続を終了し、接続に関連したデータを消去します。また、アプリケーション・プログラムがその後使用しないように、セッション・トークンを無効にします。77 ページの『セッション・トークン』には、セッション・トークンの使用法が説明されています。

WEB SEND (クライアント)

CICS Web サポートを使用して、HTTP クライアントとしての CICS による HTTP 要求を送信します。

WEB SEND (HTTP クライアントとしての CICS)



条件:

IOERR、INVREQ、LENGERR、NOTFND、NOTAUTH、NOTOPEN、TOKENERR

このコマンドはスレッド・セーフです。

説明

HTTP クライアントとしての CICS の WEB SEND は、サーバーに HTTP 要求を送信するために使用されます。このコマンドには、セッション・トークンを指定する必要があります。

ヒント: HTTP クライアントとしての CICS では、WEB SEND コマンドに続いて WEB RECEIVE コマンドを発行する代わりに CONVERSE コマンドを使用することができます。ただし、WEB CONVERSE コマンドは、チャンク形式転送コーディングをサポートしていないことを念頭に置いてください。これは、この処理には一連の送信操作が必要であり、WEB CONVERSE コマンドが実行できるのは単一の送信操作であるためです。

オプション

ACTION(cvda)

このオプションの使用目的は、メッセージを外部へ送信する方法を指定することです。HTTP クライアントとしての CICS に適用できる CVDA 値は以下のとおりです。

EXPECT

このオプションを指定すると、CICS は、Expect ヘッダーのほかに、要求行と要求のヘッダーを送信し、100-Continue 応答を待ってから、メッセージ本文をサーバーに送信するようになります。100-Continue 以外の応答を受信した場合、CICS はアプリケーション・プログラムに通知し、送信を取り消します。待機期間後に応答を受信しなかった場合、CICS は、メッセージ本文を送信します。

このオプションは、要求にメッセージ本文が含まれている場合にのみ使用してください。

CHARACTERSET(data-value)

CICS が要求のエンティティー本体を送信前に変換する場合、変換後の文字セットを指定します。文字セットの名前には、適切な句読点を含む 40 文字以内の英

数字を指定することができます。CICS では、IANA によって指定されるすべての文字セットがサポートされるわけではありません。

エンティティー本体の変換を実行するには、CLIENTCONV オプションに CLICONVERT を指定 (またはデフォルトとして許容) する必要があります。NOCLICONVERT を指定すると、エンティティー本体は変換されなくなります。変換が要求された場合、CHARACTERSET 属性が指定されていないと、デフォルトとして ISO-8859-1 が使用されます。

CHUNKING(cvda)

メッセージがチャンク形式で送信されている場合 (チャンク形式転送コーディングと呼ばれる) のメッセージの制御に使用されます。オプションが指定されていない場合のデフォルトでは、チャンク形式転送コーディングは未使用です。

アプリケーション・プログラムにとって最も便利な方法を使用して、チャンク化メッセージの内容をチャンクに分割することができます。チャンク化メッセージの本文は、CICS 文書から直接形成できません。このため、DOCTOKEN オプションは使用できません。

メッセージのチャンクごとに、CHUNKING(CHUNKYES) が指定された個別の WEB SEND コマンドを使用します。FROM オプションを使用してデータのチャンクを指定し、FROMLENGTH オプションを使用してチャンクの長さを指定します。シーケンスの最初の WEB SEND コマンドでは、メッセージに対して CLOSESTATUS オプションのようなその他のオプションを指定しますが、後続のコマンドでは指定しません。データの最後のチャンクを送信し終わったら、CHUNKING(CHUNKEND) が指定された追加の WEB SEND コマンドを指定します。FROM オプションまたは FROMLENGTH オプションは指定しません。その後 CICS は、空のチャンクを受信側に送信して、チャンク化メッセージを終了します。

CVDA 値は、以下のとおりです。

CHUNKNO

メッセージにチャンク形式転送コーディングは使用されません。CHUNKING オプションが指定されていない場合、これがデフォルトです。

CHUNKYES

チャンク形式転送コーディングが実行中です。FROM オプションで指定されたデータは、メッセージのチャンクを表します。

CHUNKEND

チャンク形式転送コーディングが完了しました。この送信にはデータは指定されません。CICS は空のチャンクを受信側に送信して、チャンク化メッセージを終了します。

注:

1. メソッド (METHOD オプション) が、チャンク形式転送コーディングに適合している必要があります。
2. チャンク化メッセージの一部の送信を開始すると、アプリケーション・プログラムは、最後の空チャンクが送信されてチャンク化メッセージが完成するまで、その他のメッセージを送信したり項目を受信したりすることはできません。

CLOSESTATUS(*cvda*)

「close」接続オプションを指定した Connection ヘッダー (Connection: close) を、メッセージに挿入するかどうかを指定します。デフォルトの設定では、ヘッダーは記述されません。CVDA 値は、以下のとおりです。

CLOSE

CICS がこの要求の Connection: close ヘッダーに書き込みを行います。このヘッダーによって、サーバーが要求に対する応答を送信した後に接続を終了することがサーバーに通知されます。(HTTP/1.0 レベルのサーバーの場合、CICS は Connection: Keep-Alive ヘッダーを省略することによって、同じ効果を得ます。)

チャンク形式転送コーディングが使用中の場合は、メッセージの最初のチャンクに CLOSESTATUS(CLOSE) オプションを指定することにより、サーバーに、チャンク化メッセージが完成し、応答が送信されたら接続を終了することを通知することができます。

チャンク形式転送コーディングが使用中ではなく、WEB SEND コマンドに CLOSESTATUS(CLOSE) オプションが指定されている場合は、新規の接続が確立されるまで、サーバーに追加メッセージを送信することはできません。

NOCLOSE

この要求では、Connection: close ヘッダーが使用されないことを意味します。サーバーが HTTP/1.0 と識別された場合、CICS は、「Keep-Alive」接続オプションを指定した Connection ヘッダー (Connection: Keep-Alive) を送信して、持続接続が求められることを通知します。

CLIENTCONV(*cvda*)

CICS が、HTTP 要求のエンティティー本体を、送信前にアプリケーションが使用するコード・ページから受信側に適した文字セットに変換するかどうかを指定します。このオプションを省略した場合、テキスト以外のメディア・タイプが指定されていない限り、デフォルトでエンティティー本体は**変換**されます。CVDA 値は、以下のとおりです。

CLICONVERT

CICS は、HTTP 要求のエンティティー本体を、アプリケーションが使用するコード・ページから、サーバー用に指定された文字セットに変換します。このコマンドの CHARACTERSET オプションを使用して、使用する文字セットを指定することができます。変換が必要で、文字セットが指定されていない場合、デフォルトで、CICS はエンティティー本体を ISO-8859-1 文字セットに変換します。(アプリケーションが使用するコード・ページは、当該接続の WEB OPEN コマンドに指定されています。)

NOCLICONVERT

HTTP 要求のエンティティー本体は、CICS で変換されずに、WEB OPEN コマンドの接続で指定されているとおりに、アプリケーションで使用されるコード・ページでサーバーに送信されます。

DOCTOKEN(*data-value*)

メッセージ本文として送信される文書の 16 バイトのバイナリー・トークンを指

定します。この文書は、CICS Document インターフェース (EXEC CICS DOCUMENT CREATE、INSERT、および SET コマンド) を使用して作成する必要があります (「*CICS Application Programming Guide*」に説明されています)。FROM オプションを指定すると、メッセージ本文を作成するための別の方法を使用できます。

チャンク化メッセージの本文は、CICS 文書からは形成できません。このため、チャンク形式転送コーディングでは DOCTOKEN オプションを使用できません。

CICS の文書は UTF-8 および UTF-16 文字エンコードに変換することはできません。

FROM(*data-area*)

メッセージ本文を保持するデータのバッファを指定します。メッセージ本文は、アプリケーション・プログラムによって作成されます。FROM オプションを指定する場合は、FROMLENGTH オプションを使用してデータのバッファ長を指定します。DOCTOKEN オプションを指定すると、メッセージ本文を作成するための別の方法を使用できますが、このオプションをチャンク化メッセージの本文に使用することはできません。

FROMLENGTH(*data-value*)

FROM オプションで指定したデータ (メッセージ本文) のバッファの長さを、フルワード・バイナリー値として指定します。この値のデータ長が誤っていると、メッセージの受信側で問題が起きる可能性があるため、値を正しく指定することが重要です。

MEDIATYPE(*data-value*)

提供されるメッセージ本文のデータ内容 (例: text/xml) を指定します。メディア・タイプは、適切な句読点を含む 56 文字以内の英数字です。CICS ではメディア・タイプのフォーマットが正しいかどうかは検査されますが、データの内容に対するメディア・タイプの妥当性は検査されません。

本文を必要とする要求では、MEDIATYPE オプションを指定する必要があります。また、デフォルトはありません。

METHOD(*cvda*)

要求の HTTP メソッドを指定します。

このコマンドでは、GET、HEAD、POST、PUT、TRACE、OPTIONS、DELETE の各メソッドがサポートされています。ただし、一部の HTTP サーバー、特に HTTP/1.0 サーバーでは、これらのメソッドの一部をインプリメントしないことがあります。

CICS は、メソッドに対してメッセージ本文の送信が適切でない場合、送信されないようにし、適切な場合にメソッドに対してメッセージ本文の送信を要求します。チャンク形式転送コーディングは、要求本体のないメソッドには関係ありません。CVDA 値は、以下のとおりです。

GET サーバーからリソースを取得します。要求本体は許可されません。

HEAD リソースの HTTP ヘッダーを取得します (応答本体は取得しません)。要求本体は許可されません。

POST サーバーにデータを送信します。要求本体が必要です。

PUT サーバー上のリソースを作成または変更します。要求本体が必要です。

TRACE

サーバーに対する要求の経路をトレースします。要求本体は許可されません。

OPTIONS

サーバーに関する情報を取得します。要求本体は許可されますが、本体の目的が定義されていません。要求本体を使用する場合は、メディア・タイプを指定する必要があります。

DELETE

サーバー上のリソースを削除します。要求本体は許可されません。

PATH(*data-area*)

アプリケーションによるアクセスが必要なサーバー内部に存在する特定のリソースのパス情報を指定します。

この接続の **WEB OPEN** コマンドに既存の **URIMAP** 定義を指定するために **URIMAP** オプションを使用した場合は、その **URIMAP** 定義に指定されているパスが **WEB SEND** コマンドのデフォルトのパスになります。この環境では、**WEB SEND** コマンドにパス情報を指定しなかった場合、この **URIMAP** 定義のパスが使用されます。**URIMAP** 定義に指定されているパスとは異なるパスを指定すると、この指定によって **URIMAP** 定義のパスは変更されます。

WEB OPEN コマンドで **URIMAP** オプションを使用しなかった場合は、デフォルトのパスが存在しないため、パス情報を入力する必要があります。パス情報は、**WEB PARSE URL** コマンドを使用して、既知の **URL** から抽出されます。

PATH オプションを使用してパス情報を入力する代わりに、**WEB SEND** コマンドの **URIMAP** オプションを使用すると、パス情報の取得元から直接 **URIMAP** 定義を指定できます。

PATHLENGTH(*data-value*)

パスの長さをフルワード・バイナリー値で指定します。**PATH** オプションを使用してパス情報を入力する場合は、**PATHLENGTH** オプションを指定する必要があります。**WEB PARSE URL** コマンドを使用して **URL** を解析すると、パス長の情報が戻されます。

QUERYSTRING(*data-area*)

要求の一環としてサーバーに入力する照会ストリングを指定します。照会ストリングの先頭に疑問符 (?) を記述する必要はありません。記述しなかった場合は、要求の構成時に **CICS** によって自動的に付加されます。照会ストリングにエスケープ文字を挿入すると、**CICS** は、エスケープ文字をエスケープ形式でサーバーに渡します。

QUERYSTRLEN(*data-value*)

QUERYSTRING オプションで指定された照会ストリングの長さをフルワード・バイナリー値で指定します。

SESTOKEN(*data-value*)

CICS とサーバー間の接続を一意に識別する 8 バイトのバイナリー値であるセッション・トークンを指定します。この値は、**WEB OPEN** コマンドによって、**HTTP** クライアントとしての **CICS** に戻されます。

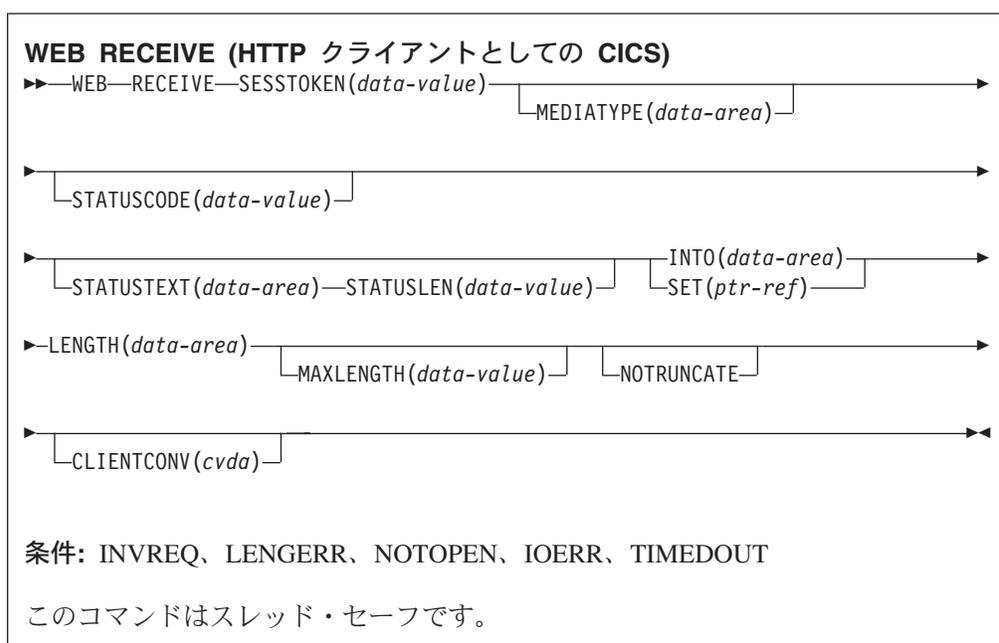
URIMAP(*data-value*)

アプリケーションがアクセスする必要のあるサーバー内部に存在する特定のリソースのパス情報を提供する URIMAP 定義の名前 (8 文字以内) を指定します。URIMAP 定義は、(USAGE(CLIENT) が指定された) HTTP クライアントとしての CICS を対象にする必要があります。その HOST 属性は、この接続の WEB OPEN コマンドに指定されていた URIMAP 定義の HOST 属性と同じにするか、またはこの接続の WEB OPEN コマンドで HOST オプションに指定されているホスト名と同じにする必要があります。WEB SEND コマンドに指定されている URIMAP 定義は、この要求にのみ適用されます。

URIMAP オプションを指定する場合は、PATH オプションや PATHLENGTH オプションを指定しないでください。

WEB RECEIVE (クライアント)

HTTP クライアントとしての CICS の HTTP 応答を受信します。



説明

HTTP クライアントとしての CICS の WEB RECEIVE では、サーバーが作成した HTTP 応答の本文が受信されます。HTTP 応答のヘッダーは、WEB READ HTTPHEADER コマンドまたは HTTP ヘッダーのブラウザ・コマンドを使用して個別に調べることができます。このコマンドには、セッション・トークンを指定する必要があります。

着信メッセージのコード・ページ変換は、このコマンドで指定できます。

注: ユーザー・アプリケーションを始動するトランザクションに指定された RTIMOUT 値は、着信メッセージを受信するためにアプリケーションが待機する準備時間を示します (RTIMOUT はトランザクション・プロファイル定義に指定されます)。RTIMOUT によって指定されている時間が経過すると、CICS に

よって TIMEDOUT 応答がアプリケーションに戻されます。RTIMOUT 値がゼロの場合は、アプリケーションが待機する準備時間が無期限であることを示しています。トランザクション・プロファイル定義での RTIMOUT のデフォルト設定はゼロであるため、HTTP クライアント要求を作成するアプリケーションで、その設定を検査し、変更することは重要です。

ヒント: HTTP クライアントとしての CICS では、WEB SEND コマンドに続いて WEB RECEIVE コマンドを発行する代わりに CONVERSE コマンドを使用することができます。

オプション

CLIENTCONV(*cvda*)

CICS が、応答のエンティティ本体を、サーバーが使用する文字セットからアプリケーションに適したコード・ページに変換するかどうかを指定します。デフォルトでは、エンティティ本体は変換されません。

CLICONVERT

CICS は、応答のエンティティ本体を、サーバーが使用する文字セットからアプリケーション用に指定されたコード・ページに変換します。

NOCLICONVERT

CICS は、応答のエンティティ本体を変換しないため、応答はサーバーが使用する文字セットでアプリケーションに渡されます。

HTTP クライアントとしての CICS の WEB RECEIVE コマンドでは、文字セットやアプリケーション・コード・ページを指定する必要はありません。コード・ページ変換が必要な場合、CICS は、メッセージの Content-Type ヘッダーを検査することにより、サーバーで使用される文字セットを識別します。ヘッダーにこの情報が含まれていない場合、または指定された文字セットが CICS のコード・ページ変換でサポートされていない場合は、ISO-8859-1 文字セットが使用されます。アプリケーションのコード・ページでは、ローカルの CICS 領域のデフォルトのコード・ページ (LOCALCCSID システム初期設定パラメーターで指定) が使用されるか、WEB OPEN コマンドで指定した別の EBCDIC コード・ページが使用されます。

INTO(*data-area*)

受信するデータを格納するためのバッファを指定します。

LENGTH(*data-area*)

CICS からアプリケーションに戻されたデータ量に設定されるフルワード・バイナリー変数が指定されます。

- NOTRUNCATE オプションが指定されていない場合、メッセージ内に存在する追加データは廃棄されています。追加データが存在していた場合、RESP2 値に 57 が設定された LENGERR 応答が戻されます。
- NOTRUNCATE オプションが指定されている場合、追加データは保存されます。追加データが使用可能な場合、RESP2 値に 36 が設定された LENGERR 応答が戻されます。この場合に実行すべき操作は、NOTRUNCATE オプションの説明で示しています。

MAXLENGTH(*data-value*)

CICS によってアプリケーションに渡されるデータの最大量をフルワード・バイナリー値として指定します。MAXLENGTH オプションは、データを受信すると

きに INTO オプションまたは SET オプションのいずれかを指定すると適用されます。チャンク形式転送コーディングを使用してデータを送信すると、CICS は大量のデータを 1 つのメッセージに組み立ててからアプリケーションに送信するため、MAXLENGTH オプションは、個々のチャンク (まとめり) に対してではなく、チャンク化メッセージの全長に対して適用されます。コード・ページの変換が実行された後のデータが測定されます。

MEDIATYPE(*data-area*)

本文のメディア・タイプ (すなわち、データ内容のタイプ) を受信するための 56 文字のデータ域を指定します (例: text/xml)。

NOTRUNCATE

使用可能なデータが、MAXLENGTH オプションで要求されている長さを超えた場合、残りのデータを即座には廃棄せず、保存して後続の RECEIVE コマンドによる検索に備えることを指定します。(RECEIVE コマンドがその後発行されなかった場合は、トランザクションの終了時にデータは廃棄されます。)

SET オプションを指定し、MAXLENGTH オプションを指定せずに単一の RECEIVE コマンドを実行すると、長さに関係なく、残りのデータすべてが受信されます。あるいは、一連の RECEIVE コマンドに NOTRUNCATE オプションを指定して実行し、残りのデータを該当するチャンク (まとめり) として受信することができます。LENGERR 応答を取得しなくなるまで、RECEIVE コマンドを発行し続けます。

SET(*ptr-ref*)

受信したデータのアドレスの設定先となるポインター参照を指定します。ポインター参照は、次の RECEIVE コマンドまたはタスクの終了まで有効です。

SESSTOKEN(*data-value*)

CICS とサーバー間の接続を一意に識別する 8 バイトのバイナリー値であるセッション・トークンを指定します。この値は、WEB OPEN コマンドによって、HTTP クライアントとしての CICS に戻されます。

STATUSCODE(*data-value*)

サーバーによって送信された HTTP 状況コードを受信するためのデータ域を指定します。このコードは、バイナリーのハーフワード値です。例として、200 (通常) や 404 (検出されない) があります。状況コードの受信はオプションですが、以下の状況では、必ず状況コードを受信し、検査する必要があります。

- サーバーに対して、現在、または将来の接続中に同一の要求を作成する予定がある場合。
- サーバーに対して、この接続を使用してさらに要求を作成する予定がある場合。
- アプリケーションが応答として受信した情報によって異なる処理をその後実行する場合。

STATUSTEXT(*data-area*)

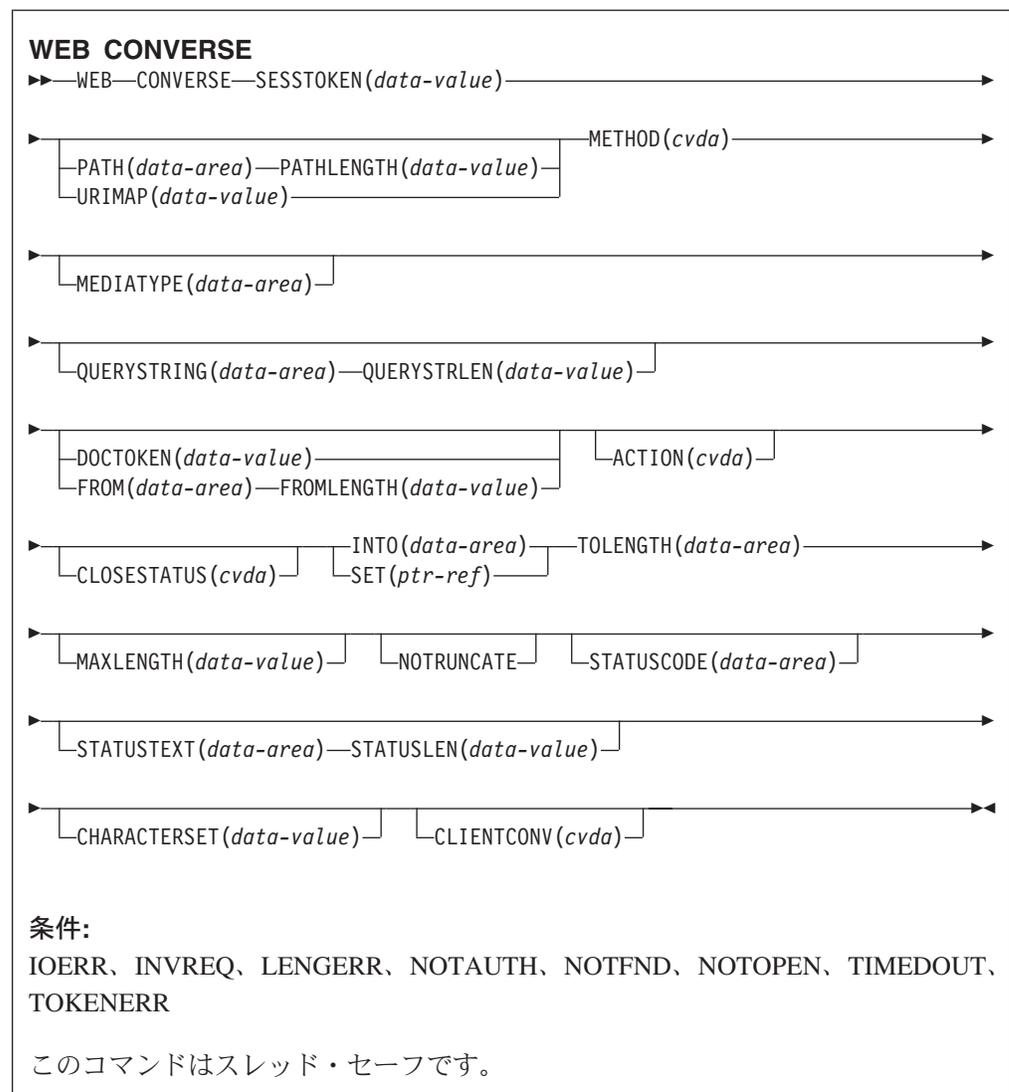
状況コードを記述するためにサーバーによって戻されるテキストを受信するためのデータ域を指定します。このテキストは、理由句と呼ばれています。例として、「了解」(状況コード 200 と共に戻される) や「不正な要求 (Bad Request)」(状況コード 400 と共に戻される) があります。STATUSLEN オプションでは、このテキストに指定できる長さを指定します。

STATUSLEN(*data-value*)

状況コードを記述するためにサーバーによって戻されるテキストの受信用のデータ域 (STATUSTEXT オプションで指定) の長さを、フルワード・バイナリー値で指定します。このテキストは、理由句と呼ばれています。HTTP で推奨されるほとんどの理由句は短い句ですが、サーバーが推奨される理由句をより詳細な情報で置き換える場合は、ここで 256 文字のデータ域の長さを指定するように推奨されています。

WEB CONVERSE

1 つのコマンドで、HTTP クライアントとしての CICS による HTTP 要求を送信し、サーバーから応答を受信します。HTTP クライアントとしての CICS に対する WEB SEND コマンドと WEB RECEIVE コマンドの代わりとなるコマンドです。



説明

WEB CONVERSE を使用すると、アプリケーション・プログラムは、HTTP クライアント要求の構成および送信を実行し、サーバーから応答を受け取ることができます。このコマンドには、セッション・トークンを指定する必要があります。

- **HTTP クライアント要求**は、WEB OPEN コマンドを使用してオープンされた接続を使用して作成されます。WEB SEND コマンドの代わりに WEB CONVERSE を使用して、要求を構成および送信できます。
- **サーバーからの応答**は、CICS Web サポートによって受信され、アプリケーションに渡されます。WEB RECEIVE コマンドの代わりに WEB CONVERSE コマンドを使用して、アプリケーション・プログラムを待機させ、HTTP 応答を受け取ることができます。HTTP 応答のヘッダーは、WEB READ HTTPHEADER コマンドまたは HTTP ヘッダーのブラウズ・コマンドを使用して別個に調べることができます。

注: ユーザー・アプリケーションを始動するトランザクションに指定された RTIMOUT 値は、着信メッセージを受信するためにアプリケーションが待機する準備時間を示します (RTIMOUT はトランザクション・プロファイル定義に指定されます)。RTIMOUT によって指定されている時間が経過すると、CICS によって TIMEDOUT 応答がアプリケーションに戻されます。RTIMOUT 値がゼロの場合は、アプリケーションが待機する準備時間が無期限であることを示しています。トランザクション・プロファイル定義での RTIMOUT のデフォルト設定はゼロであるため、HTTP クライアント要求を作成するアプリケーションで、その設定を検査し、変更することは重要です。

WEB CONVERSE コマンドは、要求に対してチャンク形式転送コーディングをサポートしていません。この処理には一連の送信アクションが必要であるが、WEB CONVERSE コマンドが実行できるのは単一の送信アクションであるためです。チャンク化メッセージを送信する場合は、WEB SEND コマンドを使用して送信し、WEB RECEIVE コマンドを使用して受信してください。サーバーがチャンク化応答を送信した場合、WEB CONVERSE コマンドを使用して受信することができます。

WEB CONVERSE コマンドは、サーバーへの接続が終了した後は使用できません。サーバーが接続の終了を要求したかどうかをテストする必要がある場合は、WEB READ HTTPHEADER コマンドを使用して、サーバーからの最後のメッセージに存在する Connection: close ヘッダーを探します。

WEB CONVERSE コマンドは、単一の送信操作と単一の受信操作を実行するため、WEB SEND コマンドと WEB RECEIVE コマンドの代用として設計されています。WEB SEND コマンドと WEB RECEIVE コマンドは、WEB CONVERSE コマンドと同じ接続に関連して (つまり、同じ SESSTOKEN を使用して) 使用できます。ただし、要求をパイプライン化 (一連の要求を応答を待たずに送信すること) する場合は、WEB SEND コマンドを実行した後、WEB CONVERSE コマンドを実行しないでください。CICS は、プログラムの実行時に各 WEB SEND コマンドの後、WEB CONVERSE コマンドが発行されてから WEB RECEIVE コマンドが発行されるかどうかを検査します。例えば、WEB SEND コマンドを 3 度使用してパイプライン化された一連の要求を発行する場合、WEB RECEIVE コマンドを 3 度使用して、こ

これらの要求に対する応答を受信してから、WEB CONVERSE コマンドを使用することができます。

HTTP クライアント要求の送信オプション

ACTION(*cvda*)

このオプションの使用目的は、メッセージを外部へ送信する方法を指定することです。HTTP クライアントとしての CICS に適用できる CVDA 値は以下のとおりです。

EXPECT

このオプションを指定すると、CICS は、Expect ヘッダーのほかに、要求行と要求のヘッダーを送信し、100-Continue 応答を待ってから、メッセージ本文をサーバーに送信するようになります。100-Continue 以外の応答を受信した場合、CICS はアプリケーション・プログラムに通知し、送信を取り消します。待機期間後に応答を受信しなかった場合、CICS は、メッセージ本文を送信します。

このオプションは、要求にメッセージ本文が含まれている場合にのみ使用してください。

CLOSESTATUS(*cvda*)

「close」接続オプションを指定した Connection ヘッダー (Connection: close) を要求に挿入するかどうかを指定します。デフォルトの設定では、ヘッダーは記述されません。CVDA 値は、以下のとおりです。

CLOSE

CICS がこの要求の Connection: close ヘッダーに書き込みを行います。このヘッダーによって、サーバーが要求に対する応答を送信した後に接続を終了することがサーバーに通知されます。(HTTP/1.0 レベルのサーバーの場合、CICS は Connection: Keep-Alive ヘッダーを省略することによって、同じ効果を得ます。)

NOCLOSE

この要求では、Connection: close ヘッダーが使用されないことを意味します。サーバーが HTTP/1.0 と識別された場合、CICS は、「Keep-Alive」接続オプションを指定した Connection ヘッダー (Connection: Keep-Alive) を送信して、持続接続が求められることを通知します。

DOCTOKEN(*data-value*)

メッセージ本文として送信される文書の 16 バイトのバイナリー・トークンを指定します。この文書は、CICS Document インターフェース (EXEC CICS DOCUMENT CREATE、INSERT、および SET コマンド) を使用して作成する必要があります (「*CICS Application Programming Guide*」に説明されています)。FROM オプションを指定すると、メッセージ本文を作成するための別の方法を使用できます。

CICS の文書は UTF-8 および UTF-16 文字エンコードに変換することはできません。

FROM(*data-area*)

メッセージ本文を保持するデータのバッファを指定します。メッセージ本文は、アプリケーション・プログラムによって作成されます。FROM オプション

を指定する場合は、FROMLENGTH オプションを使用してデータのバッファ長を指定します。DOCTOKEN オプションを指定すると、メッセージ本文を作成するための別の方法を使用できます。

FROMLENGTH(*data-area*)

FROM オプションで指定したデータ (メッセージ本文) のバッファの長さを、フルワード・バイナリー値として指定します。この値のデータ長が誤っていると、メッセージの受信側で問題が起きる可能性があるため、値を正しく指定することが重要です。

MEDIATYPE(*data-area*)

要求本体のデータ内容 (例: text/xml) を指定します。メディア・タイプは、適切な句読点を含む 56 文字以内の英数字です。CICS ではメディア・タイプのフォーマットが正しいかどうかは検査されますが、データの内容に対するメディア・タイプの妥当性は検査されません。

本文を必要とする要求では、MEDIATYPE オプションを指定する必要があります。また、デフォルトはありません。

MEDIATYPE オプションは、WEB CONVERSE コマンドの送信機能と受信機能の両方で使用されます。このオプションを値を指定して使用した場合、その値は要求内で Content-Type ヘッダーの構成のために使用され、同じフィールドがサーバーによって戻される応答のメディア・タイプを受信するために使用されません。値なしで使用した場合は、応答のメディア・タイプの受信のみに使用されません。

METHOD(*cvda*)

要求の HTTP メソッドを指定します。

このコマンドでは、GET、HEAD、POST、PUT、TRACE、OPTIONS、DELETE の各メソッドがサポートされています。ただし、一部の HTTP サーバー、特に HTTP/1.0 サーバーでは、これらのメソッドの一部をインプリメントしないことがあります。

CICS は、メソッドに対してメッセージ本文の送信が適切でない場合、送信されないようにし、適切な場合にメソッドに対してメッセージ本文の送信を要求します。CVDA 値は、以下のとおりです。

GET サーバーからリソースを取得します。要求本体は許可されません。

HEAD リソースの HTTP ヘッダーを取得します (応答本体は取得しません)。要求本体は許可されません。

POST サーバーにデータを送信します。要求本体が必要です。

PUT サーバー上のリソースを作成または変更します。要求本体が必要です。

TRACE

サーバーに対する要求の経路をトレースします。要求本体は許可されません。

OPTIONS

サーバーに関する情報を取得します。要求本体は許可されますが、本体の目的が定義されていません。要求本体を使用する場合は、メディア・タイプを指定する必要があります。

DELETE

サーバー上のリソースを削除します。要求本体は許可されません。

PATH(*data-area*)

アプリケーションによるアクセスが必要なサーバー内部に存在する特定のリソースのパス情報を指定します。

この接続の WEB OPEN コマンドに既存の URIMAP 定義を指定するために URIMAP オプションを使用した場合は、その URIMAP 定義に指定されているパスが WEB SEND コマンドのデフォルトのパスになります。この環境では、WEB SEND コマンドにパス情報を指定しなかった場合、この URIMAP 定義のパスが使用されます。URIMAP 定義に指定されているパスとは異なるパスを指定すると、この指定によって URIMAP 定義のパスは変更されます。

WEB OPEN コマンドで URIMAP オプションを使用しなかった場合は、デフォルトのパスが存在しないため、パス情報を入力する必要があります。パス情報は、WEB PARSE URL コマンドを使用して、既知の URL から抽出されます。

PATH オプションを使用してパス情報を入力する代わりに、WEB CONVERSE コマンドの URIMAP オプションを使用すると、パス情報の取得元から直接 URIMAP 定義を指定できます。

PATHLENGTH(*data-value*)

パスの長さをフルワード・バイナリー値で指定します。PATH オプションを使用してパス情報を入力する場合は、PATHLENGTH オプションを指定する必要があります。WEB PARSE URL コマンドを使用して URL を解析すると、パス長の情報が戻されます。

QUERYSTRING(*data-area*)

要求の一環としてサーバーに入力する照会ストリングを指定します。照会ストリングの先頭に疑問符 (?) を記述する必要はありません。記述しなかった場合は、要求の構成時に CICS によって自動的に付加されます。照会ストリングにエスケープ文字を挿入すると、CICS は、エスケープ文字をエスケープ形式でサーバーに渡します。

QUERYSTRLEN(*data-value*)

QUERYSTRING オプションで指定された照会ストリングの長さをフルワード・バイナリー値で指定します。

SESSTOKEN(*data-value*)

CICS とサーバー間のこの接続を一意に識別する 8 バイトのバイナリー値であるセッション・トークンを指定します。この値は、WEB OPEN コマンドによって、HTTP クライアントとしての CICS に戻されます。77 ページの『セッション・トークン』には、セッション・トークンの使用法が説明されています。

URIMAP(*data-value*)

アプリケーションがアクセスする必要のあるサーバー内の特定のリソースに関するパス情報を提供する URIMAP 定義の名前 (8 文字以内) を指定します。URIMAP 定義は、(USAGE(CLIENT) が指定された) HTTP クライアントとしての CICS を対象にする必要があります。その HOST 属性は、この接続の WEB OPEN コマンドに指定されていた URIMAP 定義の HOST 属性と同じにするか、またはこの接続の WEB OPEN コマンドで HOST オプションに指定されているホスト名と同じにする必要があります。WEB CONVERSE コマンドに指定されている URIMAP 定義は、この要求にのみ適用されます。

URIMAP オプションを指定する場合は、PATH オプションや PATHLENGTH オプションを指定しないでください。

サーバーの応答の受信オプション

INTO(*data-area*)

受信するデータを格納するためのバッファを指定します。

MAXLENGTH(*data-value*)

CICS によってアプリケーションに渡されるデータの最大量をフルワード・バイナリー値として指定します。MAXLENGTH オプションは、データを受信するときに INTO オプションまたは SET オプションのいずれかを指定すると適用されます。チャンク形式転送コーディングを使用してデータを送信すると、CICS は大量のデータを 1 つのメッセージに組み立ててからアプリケーションに送信するため、MAXLENGTH オプションは、個々のチャンク (まとめ) に対してではなく、チャンク化メッセージの全長に対して適用されます。コード・ページの変換が実行された後のデータが測定されます。

MEDIATYPE(*data-area*)

本文のメディア・タイプ (すなわち、データ内容のタイプ) を受信するための 56 文字のデータ域を指定します (例: text/xml)。

MEDIATYPE オプションは、WEB CONVERSE コマンドの送信機能と受信機能の両方で使用されます。このオプションを値を指定して使用した場合、その値は要求内で Content-Type ヘッダーの構成のために使用され、同じフィールドがサーバーによって戻される応答のメディア・タイプを受信するために使用されます。値なしで使用した場合は、応答のメディア・タイプの受信のみに使用されます。

NOTTRUNCATE

使用可能なデータが、MAXLENGTH オプションで要求されている長さを超えた場合、残りのデータを即座には廃棄せず、保存して後続の RECEIVE コマンドによる検索に備えることを指定します。(RECEIVE コマンドがその後発行されなかった場合は、トランザクションの終了時にデータは廃棄されます。)

SET オプションを指定し、MAXLENGTH オプションを指定せずに単一の RECEIVE コマンドを実行すると、長さに関係なく、残りのデータすべてが受信されます。あるいは、一連の RECEIVE コマンドに NOTTRUNCATE オプションを指定して実行し、残りのデータを該当するチャンク (まとめ) として受信することができます。LENGERR 応答を取得しなくなるまで、RECEIVE コマンドを発行し続けます。

SET(*ptr-ref*)

受信したデータのアドレスの設定先となるポインター参照を指定します。ポインター参照は、次の RECEIVE コマンドまたはタスクの終了まで有効です。

STATUSCODE(*data-area*)

サーバーによって送信された HTTP 状況コードを受信するためのデータ域を指定します。このコードは、バイナリーのハーフワード値です。例として、200 (通常) や 404 (検出されない) があります。状況コードの受信はオプションですが、以下の状況では、必ず状況コードを受信し、検査する必要があります。

- サーバーに対して、現在、または将来の接続中に同一の要求を作成する予定がある場合。

- サーバーに対して、この接続を使用してさらに要求を作成する予定がある場合。
- アプリケーションが応答として受信した情報によって異なる処理をその後実行する場合。

STATUSTEXT(*data-area*)

状況コードを記述するためにサーバーによって戻されるテキストを受信するためのデータ域を指定します。このテキストは、理由句と呼ばれています。例として、「了解」(状況コード 200 と共に戻される) や「不正な要求 (Bad Request)」(状況コード 400 と共に戻される) があります。STATUSLEN オプションでは、このテキストに指定できる長さを指定します。

STATUSLEN(*data-value*)

状況コードを記述するためにサーバーによって戻されるテキストの受信用のデータ域 (STATUSTEXT オプションで指定) の長さを、フルワード・バイナリー値で指定します。このテキストは、理由句と呼ばれています。HTTP で推奨されるほとんどの理由句は短い句ですが、サーバーが推奨される理由句をより詳細な情報で置き換える場合は、ここで 256 文字のデータ域の長さを指定するように推奨されています。

TOLENGTH(*data-area*)

CICS からアプリケーションに戻されたデータ量に設定されるフルワード・バイナリー変数が指定されます。

- NOTTRUNCATE オプションが指定されていない場合、メッセージ内に存在する追加データは廃棄されています。追加データが存在していた場合、RESP2 値に 57 が設定された LENGERR 応答が戻されます。
- NOTTRUNCATE オプションが指定されている場合、追加データは保存されます。追加データが使用可能な場合、RESP2 値に 36 が設定された LENGERR 応答が戻されます。この場合に実行すべき操作は、NOTTRUNCATE オプションの説明で示しています。

このオプションは、WEB RECEIVE の LENGTH オプションに相当します。

送信および受信した項目を変換するためのオプション

CHARACTERSET(*data-value*)

CICS が HTTP 要求のエンティティー本体を送信前に変換する場合、変換後の文字セットを指定します。文字セットの名前には、適切な句読点を含む 40 文字以内の英数字を指定することができます。CICS では、IANA によって指定されるすべての文字セットがサポートされるわけではありません。

要求の本文の変換を実行するには、CLIENTCONV オプションをデフォルトの CLICONVERT にするか、または NOINCONVERT に指定する必要があります。NOCLICONVERT または NOOUTCONVERT を指定すると、要求の本文の変換が抑制されます。変換が要求された場合、CHARACTERSET 属性が指定されていないと、デフォルトとして ISO-8859-1 が使用されます。

CLIENTCONV(*cvda*)

CICS が、送信前の HTTP 要求の本文を変換するかどうか、およびサーバーの応答のエンティティー本体を変換するかどうかを指定します。デフォルトでは、要求が外部に送信されるときと、応答が受信されるときに、エンティティー本体が**変換されず** (CLICONVERT)。

- 要求の本文の場合、このコマンドの CHARACTERSET オプションを使用して、サーバーに適した文字セットを指定できます。変換が要求されていて (またはデフォルトで実行されて) 文字セットが指定されていない場合、デフォルトで CICS はエンティティー本体を ISO-8859-1 文字セットに変換します。
- 応答の本文の場合、サーバーで使用する文字セットを指定する必要はありません。CICS は、メッセージの Content-Type ヘッダーを検査することによってこれを識別します。ヘッダーにこの情報が含まれていない場合、または指定された文字セットが CICS のコード・ページ変換でサポートされていない場合は、ISO-8859-1 文字セットが使用されます。
- アプリケーションのコード・ページでは、ローカルの CICS 領域のデフォルトのコード・ページ (LOCALCCSID システム初期設定パラメーターで指定) が使用されるか、WEB OPEN コマンドで指定した別の EBCDIC コード・ページが使用されます。

CVDA 値は、以下のとおりです。

CLICONVERT

CICS は、要求のエンティティー本体をサーバー用に指定された文字セットに変換し、応答のエンティティー本体をアプリケーションに適したコード・ページに変換します。

NOINCONVERT

CICS は、要求のエンティティー本体をサーバー用に指定された文字セットに変換します。ただし、CICS は応答のエンティティー本体は変換しないため、応答はサーバーが使用する文字セットでアプリケーションに渡されます。

NOOUTCONVERT

CICS は要求のエンティティー本体を変換しないため、要求はアプリケーションが使用するコード・ページの形式でリモート・サーバーに渡されます。ただし、CICS は、応答のエンティティー本体をアプリケーションに適したコード・ページに変換します。

NOCLICONVERT

CICS は要求のエンティティー本体を変換しないため、要求はアプリケーションが使用するコード・ページの形式でリモート・サーバーに渡されます。CICS は、応答のエンティティー本体を変換しないため、応答はサーバーが使用する文字セットでアプリケーションに渡されます。

EXEC CICS WEB コマンドのオプションの変更点

HTTP ヘッダーの EXEC CICS WEB コマンドに SESSTOKEN オプションが追加されたため、このコマンドは HTTP クライアントとしての CICS で使用できます。

以下のコマンドに SESSTOKEN オプションが追加されました。

- WEB WRITE HTTPHEADER
- WEB READ HTTPHEADER
- WEB STARTBROWSE HTTPHEADER
- WEB READNEXT HTTPHEADER
- WEB ENDBROWSE HTTPHEADER

SESSTOKEN(*data-value*)

HTTP クライアントとしての CICS では、このオプションは必須です。CICS とサーバー間の接続を一意に識別する 8 バイトのバイナリー値であるセッション・トークンを指定します。この値は、WEB OPEN コマンドによって、HTTP クライアントとしての CICS に戻されます。77 ページの『セッション・トークン』には、セッション・トークンの使用法が説明されています。

JCICS API の変更点

注: 以下の機能は、PTF XXXXX で使用できます。

JCICS アプリケーション・プログラミング・インターフェースが拡張され、以下の CICS コマンドのクライアント機能をサポートするようになりました。

WEB OPEN
WEB CLOSE
WEB SEND
WEB RECEIVE
WEB CONVERSE
WEB READ HTTPHEADER
WEB WRITE HTTPHEADER
WEB STARTBROWSE HTTPHEADER
WEB READNEXT HTTPHEADER
WEB ENDBROWSE HTTPHEADER

グローバル・ユーザー出口ルーチンの変更点

HTTP クライアントとしての CICS には、WEB OPEN コマンドの XWBOPEN、および WEB SEND コマンドの XWBSNDO という 2 つの新規グローバル・ユーザー出口ルーチンがあります。(XWBSNDO は、WEB SEND コマンドが HTTP クライアントとしての CICS で使用される場合にのみ適用されます。HTTP サーバーとしての CICS の場合には適用されないので、注意してください。)

HTTP クライアントの開始出口 XWBOPEN

XWBOPEN を使用すると、システム管理者は、HTTP クライアントとしての CICS による HTTP 要求に使用するプロキシ・サーバーを指定して、これらの要求に指定されたホスト名にセキュリティー・ポリシーを適用できます。XWBOPEN は、サーバーとの接続を開始するためにアプリケーション・プログラムによって使用される、EXEC CICS WEB OPEN コマンドの処理時に呼び出されます。

CICS 自体には、HTTP クライアントとしての CICS による HTTP 要求に対してプロキシ・サーバーを使用する (または使用しない) ことに関する要件は存在しません。また、CICS は、これらの要求に対してはセキュリティー・ポリシーを適用しません。お客様のシステムまたは組織がこれらの機能を必要としている場合、それをセットアップするのはお客様の役割です。

EXEC CICS WEB OPEN コマンドを実行すると、CICS Web ドメインは、サーバーとの接続開始を指示されます。XWBOPEN は、接続が開始される前に呼び出されま

す。接続のホスト名 (例えば、www.example.com) は、EXEC CICS WEB OPEN コマンドの HOST オプションによって指定されますが、検査のために UEPHOST パラメーターとしてユーザー出口プログラムに渡されます。この時点で、ユーザー出口プログラムは、次の 2 つの目的のために使用できます。

- **HTTP 要求がプロキシー・サーバーを使用して、必要なプロキシー・サーバーの名前を戻す必要があるかどうかを調べる。** プロキシー・サーバーが必要な場合は、戻りコード UERCPROX が使用され、プロキシー・サーバーの名前が (UEPPROXY で指定されるバッファーにある) CICS Web ドメインに戻されて、サーバーへの接続に使用されます。プロキシー・サーバーが必要でない場合は、戻りコード UERCNORM が使用されます。
- **ホスト名にセキュリティー・ポリシーを適用する。** 戻りコード UERCBARR は、ホストへのアクセスが許可されていないことを示しています。ホストへのアクセスが許可されなかった場合は、WEB OPEN コマンドに INVREQ 応答が戻されるため、アプリケーション・プログラマーは接続の開始を中止する必要があります。ホストだけでなく個々のリソースに (またはホストの代わりに個々のリソースに) セキュリティー・ポリシーを適用する場合は、EXEC CICS WEB SEND コマンドや EXEC CICS WEB CONVERSE コマンドで XWBSNDO ユーザー出口を使用すると、URL のパス部分にセキュリティー・ポリシーを適用できます。

サンプル・プログラム DFH\$WBPI および DFH\$WBEX、および関連のコピーブック DFH\$WBGA では、グローバル作業域でプロキシー・サーバー情報またはセキュリティー・ポリシーをセットアップする方法を示します。例えば、使用している CICS システムからのすべての要求に単一のプロキシー・サーバーを使用する場合は、プロキシー・サーバー名を初期設定パラメーターとして指定できます。いくつかのプロキシー・サーバーを使用するか、またはセキュリティー・ポリシーをさまざまなホスト名に適用する場合は、ホスト名を該当のプロキシー・サーバーと突き合わせるか除外対象としてマークするテーブルをロードまたは作成できますが、このテーブルは、その後、EXEC CICS WEB OPEN コマンドの処理時にルックアップ・テーブルとして使用できます。サンプル・プログラムは、プログラム・リスト・テーブル初期設定後 (PLTPI) 処理時か、または EXEC CICS WEB OPEN コマンドの使用が予想される前の任意の時点で実行できます。これらは、サンプル・プログラム DFH\$PCPI および DFH\$PCEX や、サンプル・コピーブック DFH\$PCGA がベースになっています。

出口 XWBOPEN

呼び出し時

EXEC CICS WEB OPEN コマンドの処理時。

出口固有のパラメーター

UEPHOST

WEB OPEN コマンドの HOST オプションに指定されているホスト名が格納されているフィールドのアドレス。

注: ホスト名は、このフィールドで保管されると大文字に変換されます。ユーザー出口プログラムは、ホスト名と突き合わせるときに、このことを考慮に入れます。

UEPHOSTL

ホスト名のハーフワード長が格納されているフィールドのアドレス。

UEPPROXY

プロキシ・サーバー名を指すアドレスが格納されているフィールドのアドレス。

UEPPROXYL

プロキシ・サーバー名のハーフワード長が格納されているフィールドのアドレス。

戻りコード

UERCNORM

この HTTP 要求にプロキシ・サーバーは必要なく、このホスト名は禁止されていません。

UERCPROX

この HTTP 要求にはプロキシ・サーバーが必要です。UEPPROXY は必要なプロキシ・サーバーの名前に設定されており、UEPPROXYL はそのプロキシ・サーバー名の長さに設定されています。

UERCBARR

サーバーのホスト名は禁止されています。

UERCERR

出口の処理中にエラーが発生しました。

HTTP クライアントの送信出口 XWBSNDO

XWBSNDO を使用すると、システム管理者は、HTTP クライアントとしての CICS による HTTP 要求にセキュリティー・ポリシーを指定できます。XWBSNDO が呼び出されるのは、EXEC CICS WEB SEND コマンドまたは EXEC CICS WEB CONVERSE コマンドの処理を実行するときです。ホスト名およびパス情報は、この出口に渡されるため、セキュリティー・ポリシーは、これらのコンポーネントのいずれかまたは両方に適用できます。

HTTP クライアントとしての CICS による HTTP 要求に対して、CICS 自体がセキュリティー・ポリシーを適用することはありません。したがって、お客様のシステムまたは組織がこの機能を必要とする場合、それをセットアップするのはお客様の役割です。

WEB OPEN コマンドで XWBOPEN 出口を使用すると、ホスト全体へのアクセスを禁止できます。また、XWBSNDO 出口を使用すると、ホスト全体へのアクセスを禁止することも、ホスト内部の特定のパスへのアクセスを禁止することもできます。ホスト全体へのアクセスを禁止する場合は、XWBOPEN 出口を組み合わせると行うと時間を節約できます。アプリケーション・プログラムは接続を開始することができないため、送信予定の要求を作成する時間を浪費せずに済むからです。XWBSNDO 出口にホスト名を指定するのは、異なるホストが使用する同一のパスを区別することがその主な目的です。

HTTP 要求にチャンク形式転送コーディングを使用する場合、チャンク化メッセージを対象とする最初の WEB SEND コマンドでは、XWBSNDO のみが呼び出されます。

ホストは UEPHOST パラメーターとしてユーザー出口プログラムに渡され、パスは UEPPATH パラメーターとして渡されます。戻りコード UERCNORM は、パスが許

可されていることを示しており、戻りコード UERCBARR は、パスが許可されていないことを示しています。パスが許可されなかった場合は、WEB SEND コマンドまたは WEB CONVERSE コマンドに INVREQ 応答が戻されるため、アプリケーション・プログラマーは WEB CLOSE コマンドを使用して接続を終了することにより、これを処理する必要があります。

出口 XWBSNDO

呼び出し時

HTTP クライアントとしての CICS による HTTP 要求を対象として、EXEC CICS WEB SEND コマンドまたは EXEC CICS WEB CONVERSE コマンドの処理を実行するとき。クライアント要求は、WEB SEND コマンドに SESSTOKEN パラメーターが使用されていることで示されます。

出口固有のパラメーター

UEPHOST

特定の接続での WEB OPEN コマンドの HOST オプションに指定されているホスト名が格納されているフィールドのアドレス。

注: ホスト名は、このフィールドで保管されると大文字に変換されます。ユーザー出口プログラムは、ホスト名と突き合わせるときに、このことを考慮に入れます。

UEPHOSTL

ホスト名のハーフワード長が格納されているフィールドのアドレス。

UEPPATH

WEB SEND コマンドの PATH オプションに指定されているパスが格納されているフィールドのアドレス。

UEPPATHL

パスのハーフワード長が格納されているフィールドのアドレス。

戻りコード

UERCNORM

パスは許可されています。

UERCBARR

パスが許可されていません。

モニターの変更点

パフォーマンス・クラス・モニター・レコードの DFHWEBB グループに、以下のフィールドが追加されました。

331 (TYPE-A, 'WBREDOCT', 4 BYTES)

CICS が HTTP クライアントの場合に、ユーザー・タスクによって発行された CICS Web サポート READ HTTPHEADER 要求の数。

332 (TYPE-A, 'WBWRTOCT', 4 BYTES)

CICS が HTTP クライアントの場合に、ユーザー・タスクによって発行された CICS Web サポート WRITE HTTPHEADER 要求の数。

333 (TYPE-A, 'WBRCVIN1', 4 BYTES)

CICS が HTTP クライアントの場合に、ユーザー・タスクによって発行された CICS Web サポート RECEIVE および CONVERSE 要求の数。

334 (TYPE-A, 'WBCHRIN1', 4 BYTES)

CICS が HTTP クライアントの場合に、ユーザー・タスクによって発行された CICS Web サポート RECEIVE および CONVERSE 要求によって受信されるバイト数。

335 (TYPE-A, 'WBSNDOU1, 4 BYTES)

CICS が HTTP クライアントの場合に、ユーザー・タスクによって発行された CICS Web サポート SEND および CONVERSE 要求の数。

336 (TYPE-A, 'WBCHROU1', 4 BYTES)

CICS が HTTP クライアントの場合に、ユーザー・タスクによって発行された CICS Web サポート SEND および CONVERSE 要求によって送信されるバイト数。

337 (TYPE-A, 'WBPARSCT', 4 BYTES)

ユーザー・タスクによって発行された CICS Web サポート PARSE URL 要求の数。

338 (TYPE-A, 'WBBRWOC', 4 BYTES)

CICS が HTTP クライアントの場合に、ユーザー・タスクによって発行された CICS Web サポート BROWSE HTTPHEADER 要求 (STARTBROWSE、READNEXT、および ENDBROWSE) の数。

注: WEB CONVERSE コマンドを使用して要求が行われた場合、これによって Send および Receive 要求数 (WBSNDOU1 および WBRCVIN1) 並びに送信および受信文字数 (WBCHRIN1 および WBCHROU1) が増分します。

第 4 章 HTTP/1.1 への CICS Web サポートのアップグレード

CICS Web サポートは HTTP/1.1 をサポートするようになりました。

CICS Transaction Server for z/OS バージョン 3 リリース 1 以前の CICS のリリースでは、HTTP/1.0 がサポートされていました。CICS Web サポートが拡張されて、HTTP/1.1 仕様の機能 (チャンク形式転送コーディング、パイプライン化、および持続接続を含む) を処理し提供するようになりました。

CICS Web サポートは、Internet Society and IETF (Internet Engineering Task Force) の Request for Comments 文書 RFC 2616、「*Hypertext Transfer Protocol - HTTP/1.1*」に記述されている HTTP/1.1 仕様に条件付きで準拠しています。(RFC 2616 は <http://www.ietf.org/rfc/rfc2616.txt> からダウンロードできます。) HTTP/1.1 仕様への条件付き準拠は、CICS が、HTTP/1.1 仕様に記述されているすべての必須「MUST」レベルの要件を満たしているものの、推奨「SHOULD」レベルの要件を完全には満たしていないことを意味します。ここでいう要件とは、CICS 自体で実際に提供される機能に関連する要件を指します。ご使用のユーザー・アプリケーション・プログラムが実行する処理における準拠についてはそのユーザー・アプリケーション・プログラムも責任を共有します。このためのガイドラインが提供されており、これを行う準備ができたなら、このガイドラインを使用してアプリケーション・プログラムを準拠させることができます。

CICS TS バージョン 3 の新規動作

- CICS はインバウンド・メッセージが HTTP/1.1 に準拠しているかを確認して、非準拠のメッセージを処理または拒否します。
- CICS は、URL を比較するために HTTP/1.1 の規則に準拠します。
- CICS はアウトバウンド・メッセージの開始行に適切な HTTP バージョン番号を提供します。
- アウトバウンド HTTP/1.1 メッセージでは、CICS は、HTTP/1.1 に準拠するメッセージに通常あるべき以下の HTTP ヘッダーを提供します。
 - Content-Length (クライアント要求またはサーバー応答の場合)。
 - Content-Type (クライアント要求またはサーバー応答の場合)。
 - Date (クライアント要求またはサーバー応答の場合)。
 - Host (クライアント要求の場合)。
 - Last-Modified (HFS ファイルによる静的サーバー応答の場合のみ)。
 - Server (サーバー応答の場合)。
 - TE (クライアント要求の場合)。
 - Transfer-Encoding (クライアント要求またはサーバー応答の場合)。
 - User-Agent (クライアント要求の場合)。
- CICS はインバウンド要求とアウトバウンド要求の両方で Expect ヘッダーに関する処理を行います。
- CICS は Web クライアントからの OPTIONS 要求を処理して、適切な応答を行います。

- CICS は Web クライアントからの TRACE 要求を処理して、適切な応答を行います。
- CICS はチャンク形式転送コーディングによるインバウンド・メッセージを受け入れて、それらを組み立てます。さらに、ユーザーがチャンク形式転送コーディングを使用してアウトバウンド・メッセージを送信できるようにします。
- CICS は、インバウンド・メッセージとアウトバウンド・メッセージの両方でパイプライン化をサポートします。
- CICS は、URIMAP 定義に基づいて仮想ホスティング (複数のホスト名が同じ IP アドレスを持つ) をサポートします。

CICS TS バージョン 2 から変更された動作

- 接続は、デフォルトで持続接続になります。
- CICS が処理するエラー状態および非サポート・メッセージの範囲が広がりました。

CICS Web サポートでサポートされない HTTP 機能

HTTP/1.1 仕様 (RFC 2616) には、HTTP プロトコルを使用する通話者のさまざまな役割が定義されています。CICS Web サポートは、起点サーバー、クライアント、およびユーザー・エージェント (ただし人間のユーザーはどの HTTP クライアント要求にも関係しない) に対して、適切な HTTP サービスを提供します。

HTTP/1.1 仕様には、CICS Web サポートとは関係のない役割に関する要件も含まれますが、これらは無視することができます。

- CICS はプロキシの役割を果たしません。
- CICS はゲートウェイ (他のサーバーの仲介) またはトンネル (HTTP 接続間の中継) の役割を果たしません。
- CICS はキャッシング機能を提供しません。ユーザー作成のキャッシング機能もサポートしません。
- CICS は Web ブラウザーとしての用途向けには設計されていません。

HTTP/1.1 への CICS Web サポートのアップグレードの利点

HTTP/1.1 への CICS Web サポートのアップグレードによって、以下の利点が得られます。

- HTTP/1.1 仕様 (RFC 2616) への条件付き準拠とは、CICS Transaction Server for z/OS バージョン 3 リリース 1 がインターネット上のより広範なサービスおよびクライアントと、簡単かつ正確に対話できることを意味します。
- CICS とサーバーまたは Web クライアントとの間の接続は、持続接続がデフォルトになりました。持続接続では、要求ごとに新規の接続を確立する必要がないため、ネットワークのパフォーマンスが向上します。
- パイプライン化のサポートは、CICS アプリケーションが、一連の複数の要求を、それぞれの項目ごとに肯定応答を待機せずにサーバーに送信できることを意味します。CICS Web サポートも、Web クライアントから CICS へのパイプライン化された要求が正しく処理されることを保証します。

- チャンク形式転送コーディング (略してチャンク化と呼ばれる) を使用すると、動的に生成されたコンテンツ、または大量のコンテンツを扱いやすいセグメントにして転送することが可能であり、一方、受信側は完全なメッセージを受信したことを検査できます。CICS はこの方法で送信された項目を受信することができ、ユーザー作成アプリケーション・プログラムはこの方法で項目を送信することができます。
- CICS は、ユーザーの URIMAP 定義を使用して仮想ホストを自動的に作成するため、同じ IP アドレスで複数のホスト名を提供することができます。
- CICS は、HTTP/1.1 仕様によって定義されるすべての HTTP メソッドに関する Web クライアントおよび CICS アプリケーションからの要求をサポートします。さらに、OPTIONS および TRACE メソッドに関する Web クライアントからの要求に対して自動処理および応答を提供します。

要件

この機能をサポートするための特別なハードウェア要件およびソフトウェア要件はありません。

関連情報

377 ページの『第 27 章 CICS 稼働環境』

新しい HTTP 機能

CICS Web サポートには、新しい領域の機能が含まれます。

- チャンク形式転送コーディング、またはチャンク化
- パイプライン化
- 持続接続
- 仮想ホスティング

チャンク形式転送コーディング

チャンク形式転送コーディング (チャンク化としても呼ばれる) では、メッセージの本文を、それぞれに独自のチャンク・サイズ・ヘッダーを持つ一連のチャンクとして転送する必要があります。メッセージの末尾は、長さがゼロのチャンクとブランク行で示されます。

この定義済みプロセスは、アプリケーション生成のエンティティー本体または大規模なエンティティー本体を、適切なサイズのセグメント単位で送信できるという意味になります。クライアントまたはサーバーは、長さゼロのチャンクを受信すると、チャンク化メッセージが完了したことを認識します。

チャンク化メッセージの本文の後には、オプションのトレーラーが置かれる場合があります。ここには、末尾ヘッダーと呼ばれる補足の HTTP ヘッダーが格納されています。クライアントおよびサーバーはトレーラーを受け入れる必要はないため、補足の HTTP ヘッダーが提供する情報は、クライアントがトレーラーを受け入れることをサーバーが認識していない限り、重要な情報にはなりません。

チャンク形式転送コーディングを使用するには、クライアントとサーバーの両方が HTTP/1.1 バージョンになっている必要があります。チャンク化メッセージを

HTTP/1.0 クライアントに送信できません。チャンク形式転送コーディングに適用される要件とトレーラーの使用法については、HTTP/1.1 仕様書 (RFC 2616) に定義されています。RFC 2616 は、<http://www.ietf.org/rfc/rfc2616.txt> から入手できます。

CICS Web サポートでのチャンク形式転送コーディングの処理方法

チャンク形式転送コーディングを使用したメッセージは、CICS によって送信および受信できます。HTTP サーバーとしての CICS は、チャンク化されたメッセージを要求として受信するか、または応答として送信することができます。HTTP クライアントとしての CICS は、チャンク化されたメッセージを要求として送信するか、または応答として受信することができます。

CICS Web サポートでは、さまざまな事例が以下のように処理されます。

- HTTP サーバーとしての CICS が、チャンク化されたメッセージを HTTP 要求として受信すると、CICS Web サポートは、チャンク化エンコードを認識します。CICS は、すべてのチャンクを受信する (長さがゼロのチャンクを受信することで通知される) まで待機し、そのチャンクを組み立ててメッセージを完成させます。組み立て済みのメッセージ本文は、WEB RECEIVE コマンドを使用して、ユーザー・アプリケーション・プログラムによって受信できます。
 - 要求が到着するポートが関連先になっている TCPIP SERVICE リソース定義の MAXDATALEN オプションを使用すると、CICS が 1 つのチャンク化メッセージとして許容できるデータ量の合計を制限できます。
 - CICS が HTTP サーバーである場合、チャンク化メッセージを受信する場合のタイムアウト値は、TCPIP SERVICE 定義の SOCKETCLOSE 属性によって設定されます。
 - チャンク化メッセージの末尾ヘッダーは、HTTP ヘッダー・コマンドを使用して読み取ることができます。Trailer ヘッダーは、末尾ヘッダーとして存在したヘッダーの名前を示します。要求の処理経路にアナライザー・プログラムを使用している場合、末尾ヘッダーは、メインの要求ヘッダーと共にアナライザー・プログラムに渡されないことに注意してください。
- HTTP クライアントとしての CICS が、アプリケーション・プログラムの要求に対する応答としてチャンク化メッセージを受信すると、チャンクも、アプリケーション・プログラムにエンティティ本体として渡される前に組み立てられ、末尾ヘッダーは HTTP ヘッダー・コマンドを使用して読み取ることができます。アプリケーションが応答を受信するまでの待機時間を指定するには、アプリケーション・プログラムに関連するトランザクション ID に対応するトランザクション・プロファイル定義の RTIMOUT 属性を使用します。
- CICS が HTTP サーバー (応答) または HTTP クライアント (要求) としてチャンク化メッセージを送信すると、アプリケーション・プログラムは、メッセージのチャンクごとに WEB SEND コマンドの CHUNKLENGTH オプションを使用することにより、チャンク形式転送コーディングを指定できます。メッセージは、方法の内容を問わず、アプリケーション・プログラムに最も都合の良い方法で分割できます。例えば、その他の多数のアプリケーション・プログラムからの出力を、出力されたままの状態で送信したり、表の各列からのデータを個別に読み取って送信したりすることができます。CICS は、メッセージの各チャンクに適切な HTTP ヘッダーを追加して送信し、チャンク化エンコードが使用されることを受信側に示します。アプリケーション・プログラムは、CHUNKLENGTH(0) とブランク・データが指定された WEB SEND を発行して、メッセージの末尾を

示します。次に、CICS は、(ブランク行が入っている) 空のチャンクを送信し、要求されているトレーラー (末尾ヘッダー) を付加してチャンク化エンコードを終了します。

パイプライン化

パイプライン化では、クライアントが複数の HTTP 要求を、応答を待つことなくサーバーに送信する必要があります。応答は、要求を受信したときと同じ順序でリモート・サーバーから戻す必要があります。

要求がべき等になっていることを確認するのは、要求側の役割です。べき等とは、一連の要求のすべてまたは一部が繰り返された場合、必ず同じ結果が得られるという意味です。このようになっていると、サーバーとの接続時にエラーがあった場合、サーバーにおける要求の実装状態が、すべて、一部、またはなしのいずれに該当するか不明な場合でも、クライアントは一連の要求を再試行できます。

HTTP/1.1 仕様書 (RFC 2616) には、HTTP 要求のべき等に関する規則が定義されています。RFC 2616 は、<http://www.ietf.org/rfc/rfc2616.txt> から入手できます。簡単に説明すると、大半の要求メソッドは、それが単独で使用されている場合はべき等です。これは、要求メソッドが使用されるたびに同じ結果が得られるからです (POST メソッドは、サーバーのリソースを変更するため、例外です)。ただし、要求シーケンスがパイプライン化時に送出されると、このシーケンスは、特にリソースが変更される場合にべき等ではなくなる可能性があります。

要求をパイプライン化する予定の場合は、要求シーケンスを任意の時点で終了できることと、論理エラーを発生させることなく先頭から再開できることを確認してください。パイプライン化する予定のない場合は、要求を個別に実行し、各要求の実行後の確認を待ってください。

CICS Web サポートでのパイプライン化の処理方法

パイプライン化された要求シーケンスは、CICS によって送信および受信できます。HTTP サーバーとしての CICS は、パイプライン化された要求シーケンスを Web クライアントから受信でき、HTTP クライアントとしての CICS は、パイプライン化された要求シーケンスをサーバーに送信できます。

CICS Web サポートでは、パイプライン化された要求シーケンスとそれに対する応答が、以下のように処理されます。

- HTTP サーバーとしての CICS が、パイプライン化された HTTP 要求シーケンスを受信すると、要求は逐次処理されます。各トランザクションでは、1 つの要求が処理され、1 つの応答が得られます。パイプライン化されたメッセージ・シーケンス内の残りの要求は、前の要求に対する応答が送信されるまで CICS によって保持され、その後、追加の要求をそれぞれ処理するために新しいトランザクションが開始されます。つまり、要求はそれぞれ異なるアプリケーション・プログラムか、同じプログラムの異なるインスタンスによって処理されます。
- HTTP クライアントとしての CICS が、パイプライン化された要求シーケンスを送信すると、パイプライン化は自動的に使用可能になります。各 HTTP 要求は同時に送信されるため、アプリケーション・プログラムは、応答を受信する前に複

数の HTTP 要求を送信できます。パイプライン化されたシーケンス内の最後のメッセージが送信されると、アプリケーションは応答の受信を開始できるようになります。

- HTTP クライアントとしての CICS が、パイプライン化された要求シーケンスに対する HTTP 応答を受信すると、アプリケーション・プログラムに戻される応答の順序は、CICS がサーバーから応答を受信する順序になります。

HTTP クライアントとしての CICS の場合、要求のパイプライン化シーケンスが必ずべき等になるようにするのは、アプリケーション・プログラムの役割です。べき等の説明については、

持続接続

持続接続とは、要求と応答を複数回やり取りするために再利用が可能な、Web クライアントとサーバー間の接続のことです。

HTTP/1.0 では、サーバーが Web クライアントから要求を受信して応答を送信した場合、サーバーのデフォルトのアクションは、接続を終了することでした。Web クライアントは、サーバーによる接続の維持を必要とした場合、要求時に `Connection: Keep-Alive` ヘッダーを送信する必要がありました。

HTTP/1.1 では、持続接続がデフォルトになっています。Web クライアントとサーバー間で接続が確立されると、サーバーは、デフォルトで接続を維持します。接続が終了するのは、Web クライアントが `Connection: close` ヘッダーを送信して終了を要求した場合、サーバーのタイムアウト設定に到達した場合、またはサーバーがエラーを検出した場合のいずれかに限られます。

持続接続では、要求ごとに新規の接続を確立する必要がないため、ネットワークのパフォーマンスが向上します。新規の接続を確立する場合は、既存の接続を使用して要求を実行する場合と比較して、追加のネットワーク・リソースが大幅に消費されます。

CICS Web サポートでの持続接続の処理方法

持続接続は、CICS Web サポート機能のデフォルトの動作です。

CICS TS 3.1 以前での接続の動作では、Web クライアントが `Connection: Keep-Alive` ヘッダーを送信していない限り、データが Web クライアントから受信されると、CICS は、通常、接続を終了していました。

今回からは、HTTP サーバーとしての CICS と Web クライアントとの間や、HTTP クライアントとしての CICS とサーバーとの間で接続を確立すると、CICS はデフォルトで接続を開始しようとするようになりました。

CICS が HTTP サーバーである場合、持続接続が終了する条件は次のとおりです。

- 要求を処理しているユーザー作成アプリケーションが、(WEB SEND コマンドで `CLOSESTATUS(CLOSE)` オプションを指定することにより) 接続を終了する場合。
- Web クライアントが接続を終了した場合 (`Connection: close` ヘッダーによって通知)。

- Web クライアントが、Connection: Keep-Alive ヘッダーを送信しない HTTP/1.0 クライアントである場合。
- タイムアウト期間が満了した場合 (接続に障害が発生したこと、または Web クライアントが意図的に接続を終了したことを表示)。

その他の場合、CICS は、Web クライアントが追加の要求を送信できるように、持続接続状態を維持します。

CICS Web サポートの TCPIPSERVICE リソース定義を使用する場合は、TCPIPSERVICE 定義の SOCKETCLOSE 属性をゼロに指定しないでください。SOCKETCLOSE をゼロに設定することは、後続に待機しているデータがない限り、HTTP サーバーとしての CICS は Web クライアントからのデータ受信直後に接続を終了するという意味になります。つまり、持続接続を維持することができません。

CICS が HTTP クライアントである場合、持続接続が終了する条件は次のとおりです。

- サーバーが接続を終了した場合 (Connection: close ヘッダーを送信している HTTP/1.1 サーバー、または Connection: Keep-Alive ヘッダーを送信できない HTTP/1.0 サーバーによって通知された接続)。
- ユーザー・アプリケーション・プログラムが接続を終了した場合 (WEB SEND または WEB CONVERSE コマンドに CLOSESTATUS(CLOSE) オプションを指定するか、または WEB CLOSE コマンドを発行して実行)。
- タスクの終わりに到達したが、接続が終了していない場合。

仮想ホスティング

HTTP には、同じ IP アドレスにある複数のホストを 1 つの HTTP サーバーが代表できるという、仮想ホスティングの概念が導入されています。

DNS サーバーは、複数の異なるホスト名を同一の IP アドレスに割り振ることができます。HTTP クライアントは、特定のホストに対して要求を行う場合、DNS サーバーを使用してこのホスト名に対応する IP アドレスを検索し、この IP アドレスに要求を送信します。

HTTP/1.0 では、HTTP メッセージ内にホスト名は存在しません。ホスト名は IP アドレスの解決後に失われるためです。つまり、IP アドレスで表されるサーバーに複数の組のリソースが保持されていたとすると、サーバーはどのリソースがどのホストに属しているか区別するのが困難になっていたという意味です。

ただし、HTTP/1.1 要求では、要求の中にホスト名が存在します (通常は Host ヘッダー内)。メッセージ内にホスト名が存在すると、HTTP サーバーは、異なるホスト名が記述された複数の要求をホストごとに適切なリソースに送信できます。HTTP のこの機能は、仮想ホスティングと呼ばれています。

仮想ホスティングの管理

CICS は、URIMAP リソース定義オブジェクトを介した仮想ホスティングをサポートします。

HTTP サーバーとしての CICS を対象として (URIMAP 定義の USAGE(SERVER) を使用して) セットアップした各 URIMAP 定義には、Web クライアントがこの要求に入力することが予想されるホスト名が記述されています。CICS では、同じホスト名と同じ TCPIP SERVICE 定義を指定する CICS 領域に存在するすべての URIMAP 定義を 1 つのデータ構造にまとめることにより、仮想ホストが自動的に作成されます。TCPIP SERVICE 定義を指定しない URIMAP 定義は、すべての TCPIP SERVICE に適用されることになるため、一致するホスト名を指定するすべてのデータ構造に追加されます。このため、これらの URIMAP 定義は複数のデータ構造の一部になる可能性があります。この結果、これらの URIMAP 定義グループのそれぞれは、1 つの単位として管理できる仮想ホストを形成します。

以下の CICS コマンドを使用すると、URIMAP 定義を基に CICS で作成した仮想ホストを管理できます。

- INQUIRE HOST コマンドは、仮想ホストの状況を調べるときに使用します。このコマンドを実行すると、仮想ホストのホスト名、仮想ホストの関連先の TCPIP SERVICE 定義 (または、仮想ホストが CICS 領域のすべての TCPIP SERVICE 定義と関連しているかどうか)、および仮想ホストの状況 (使用可能または使用不可) が通知されます。
- SET HOST コマンドは、仮想ホストの状況を使用可能または使用不可に設定します。仮想ホストを使用不可に設定することは、仮想ホストを構成するすべての URIMAP 定義にアプリケーションがアクセスできなくなることを意味します。仮想ホストを使用不可にすると、CICS は、Web クライアントに HTTP 503 応答 (サービスが利用不可) を返します。
- 仮想ホストのブラウズ・コマンドは、CICS システムに存在する仮想ホストをブラウズするときを使用します。

統計プログラム DFH0STAT には、CICS が作成した仮想ホストを示すレポートが組み込まれています。

CICS では、仮想ホストを構成するすべての URIMAP 定義が削除されると、仮想ホストが自動的に削除されます。CICS によって作成された仮想ホストを使用しない場合は、仮想ホストを無視して、使用している URIMAP 定義のレベルで管理すれば済みます。

アナライザー・プログラムを使用して、仮想ホストを管理することもできます。HTTP 要求のホスト名はアナライザー・プログラムに渡されるため、要求に対してホスト依存の応答を戻すプログラムをコーディングできます。ただし、この方法でセットアップした仮想ホストは、INQUIRE HOST、SET HOST の各コマンドや仮想ホストのブラウズ・コマンドを使用して管理することはできません。

CICS 外部インターフェースの変更点

リソース定義の変更点

TCPIP SERVICE リソース定義の変更点

TCPIP SERVICE には、以下の新規属性があります。

MAXDATALEN(32|number)

HTTP プロトコルまたは USER プロトコルで、HTTP サーバーとしての CICS が受信できるデータの最大長を定義します。デフォルト値は 32K です。最小値は 3K、最大値は 524288K です。CICS Web サポートのセキュリティーを向上させるには、HTTP プロトコルのすべての TCPIP SERVICE 定義でこのオプションを指定します。大量のデータ送信を伴うサービス妨害攻撃からの保護に役立ちます。

PROTOCOL 属性で新しい USER オプションが使用可能になりました。すべての非 HTTP 要求の処理は USER プロトコルのもとで実行しなければならなくなりました。USER プロトコルで受信したメッセージに対して構文解析は実行されず、ネットワーク経由での送信のために分割された要求は自動的に組み立てられません。これは、以前の CICS リリースで非 HTTP メッセージを処理するときと同じ動作です。

PROTOCOL(EC|HTTP|IIOP|USER)

TCP/IP ポートで使用されるアプリケーション・レベルのプロトコルを指定します。

ECI CICS ECI プロトコルを使用します。

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るメッセージに対して、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 を備えていない HTTP に使用される) および 443 (SSL を備えた HTTP に使用される) に必要です。

IIOP IIOP プロトコルを使用します。Enterprise Bean に対するインバウンド要求を受信する TCPIP SERVICE の場合は、IIOP を指定します。

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。メッセージには、非 HTTP のフラグが立てられ、TCPIP SERVICE のためのアナライザー・プログラムに未変更のまま渡されます。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。すべての非 HTTP 要求の処理は USER プロトコルのもとで実行しなければならないので、非 HTTP 要求は、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から保護されます。HTTP メッセージが USER プロトコルによって処理される場合は、ユーザーがその妥当性を検査する責任があります。

PROTOCOL を USER に設定した場合に使用される TCPIP SERVICE リソース定義の属性は、PROTOCOL を HTTP に設定した場合に使用される属性と同じです。

URIMAP 定義は USER プロトコルと一緒に使用しません。

CICS 提供の新規トランザクション CWXU (CICS Web ユーザー定義プロトコル接続トランザクション) は、プロトコルが USER として定義された場合のデフォルトです。CWXU はプログラム DFHWBXN を実行します。

SOCKETCLOSE 属性は、以下のように記述されるようになりました。HTTP プロトコルの場合は、SOCKETCLOSE を 0 と指定しないでください。

SOCKETCLOSE(NO|hhmmss)

ソケットでの着信データに対して RECEIVE を発行した後で、CICS がそのソケットをクローズするまで待機するかどうか、および待機する期間を指定します。

NO ソケットは、クライアントによってクローズされるか、CICS のユーザー・アプリケーション・プログラムによってクローズされるまで、オープンしたままになります。

hhmmss

CICS によってソケットがタイムアウトになるまでの時間 (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、最初の RECEIVE 以外の RECEIVE に対して利用可能なデータがないときは、ソケットは即時にクローズされます。

HTTP プロトコルを指定した CICS Web サポートで TCPIP SERVICE を使用している場合、SOCKETCLOSE(0) は指定しないでください。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かぎり、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。つまり、持続接続を維持することができません。

PROTOCOL(ECI) を指定する場合は、SOCKETCLOSE(NO) を指定する必要があります。

SOCKETCLOSE 属性は、接続が確立された後で最初に発行される RECEIVE には適用されません。HTTP、USER および ECI プロトコルの場合、この最初の RECEIVE 要求に対して、CICS はソケットをクローズする前に 30 秒間データを待機します。IIOP プロトコルの場合、CICS は無期限に待機します。

TCPIP SERVICE をインストールした後は、CEMT を使用してこの値を変更することはできません。TCPIP SERVICE をサービス休止に設定してから、変更済み定義を指定して TCPIP SERVICE を再インストールする必要があります。

アプリケーション・プログラミング・インターフェース (HTTP/1.1 サポート) の変更点

新しいコマンドと変更されたコマンド

HTTP サーバーとしての CICS で使用される場合について、以下の EXEC CICS WEB コマンドが拡張されました。

- EXEC CICS WEB SEND
- EXEC CICS WEB RECEIVE

CICS Web サポートは、CICS Transaction Server for z/OS バージョン 3 リリース 1 よりも前にこれらのコマンドを使用していた Web 対応アプリケーション・プログラムが未変更のまま動作できるように設計されているので、これらのプログラムをマイグレーションして現在使用可能な拡張を利用することに決めるまでは、そのまま使用できます。

CICS Transaction Server for z/OS バージョン 3 リリース 1 より前に WEB SEND コマンドで使用可能だったオプションは CLNTCODEPAGE、DOCTOKEN、LENGTH、STATUSCODE および STATUSTEXT です。CICS Transaction Server for z/OS バージョン 3 リリース 1 での変更点は以下のとおりです。

- オプション DOCTOKEN、STATUSCODE、および STATUSTEXT の名前と機能は変更されていません。
- オプション CLNTCODEPAGE および LENGTH は、マイグレーションの目的でのみサポートされるようになり、これらのオプションの機能はそれぞれ、新しいオプション CHARACTERSET および STATUSLEN に置き換えられました。
- 機能性を拡張するため、いくつかの新しいオプションが使用可能になりました。

CICS Transaction Server for z/OS バージョン 3 リリース 1 より前に WEB RECEIVE コマンドで使用可能だったオプションは CLNTCODEPAGE、HOSTCODEPAGE、INTO、LENGTH、MAXLENGTH、NOTTRUNCATE、SET、および TYPE です。CICS Transaction Server for z/OS バージョン 3 リリース 1 での変更点は以下のとおりです。

- オプション INTO、LENGTH、MAXLENGTH、NOTTRUNCATE、SET、および TYPE の名前と機能は変更されていません。
- オプション HOSTCODEPAGE は現在も使用できますが、必要ではなくなりました。このオプションを指定しない場合は、CICS によってデフォルトが提供されます。
- オプション CLNTCODEPAGE はマイグレーションの目的でのみサポートされるようになり、このオプションの機能は新しいオプション CHARACTERSET に置き換えられました。
- 機能性を拡張するため、いくつかの新しいオプションが使用可能になりました。

HTTP クライアントとしての CICS で使用される場合も、EXEC CICS WEB SEND および WEB RECEIVE コマンドにはさまざまな新規オプションがあります。これについては、78 ページの『アプリケーション・プログラミング・インターフェースの変更点 (HTTP クライアント要求)』で説明します。

以下の EXEC CICS コマンドは、HTTP サーバーとしての CICS および HTTP クライアントとしての CICS の両方に対して提供または拡張されました。

- EXEC CICS WEB PARSE URL
- EXEC CICS WEB EXTRACT
- EXEC CICS CONVERTTIME
- EXEC CICS FORMATTIME

これらのコマンドについては、145 ページの『アプリケーション・プログラミング・インターフェースの変更点 (汎用 CICS Web サポート拡張)』で説明します。

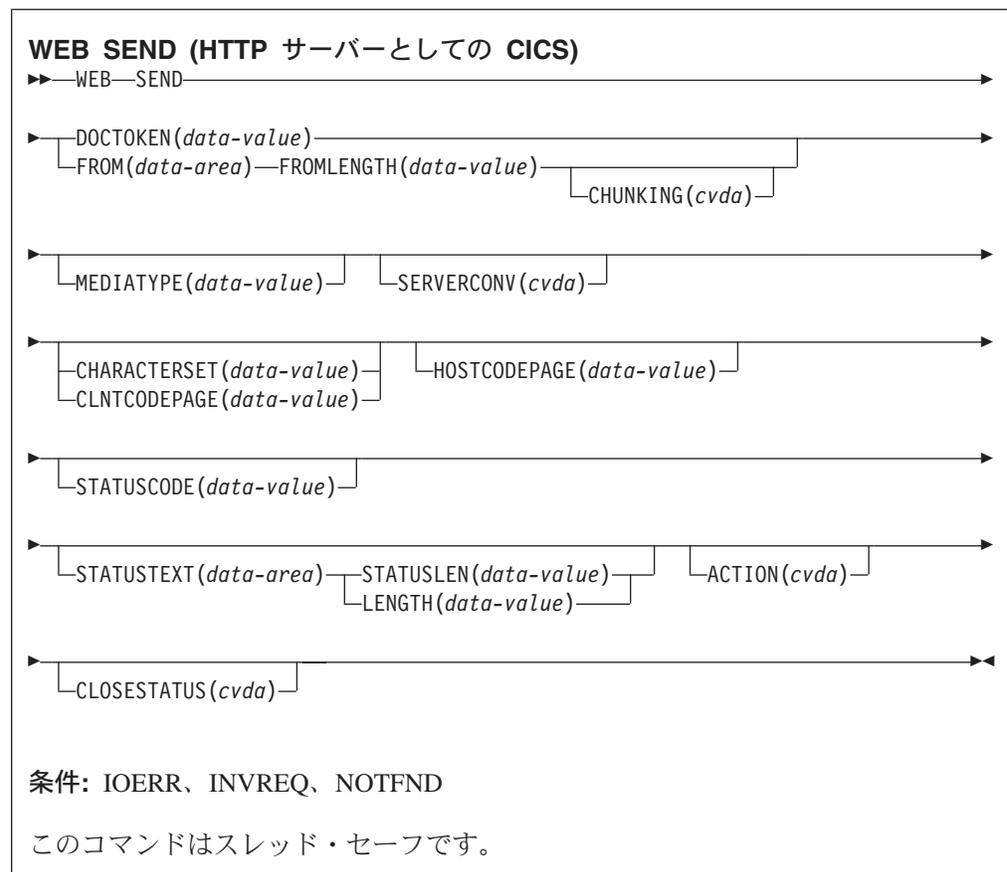
EXTRACT TCPIP コマンドに新規 MAXDATALEN オプションが追加されました。

MAXDATALEN(data-area)

HTTP サーバーとしての CICS が受信できるデータの最大長の設定が格納されるフルワード・バイナリー・フィールドを指定します。

WEB SEND (サーバー)

HTTP 応答または非 HTTP メッセージを送信します。



説明

HTTP サーバーとしての CICS の WEB SEND では、CICS Web サポートまたは CICS ビジネス・ロジック・インターフェースごとにデリバリー項目を選択し、選択した項目を送信するためのオプションを指定します。項目の内容は次のとおりです。

- Web クライアントによって HTTP サーバーとしての CICS に対して出された HTTP 要求に対する応答。
- CICS Web サポート機能によって処理される非 HTTP メッセージ。
TCPIP SERVICE 定義にユーザー定義の (USER) プロトコルが指定されます。
- CICS HTTP リスナー経由ではなく、CICS ビジネス・ロジック・インターフェースを使用して直接アプリケーション・プログラムにアクセスした別のプログラムからの応答。

1 つのタスクの実行中に送信できる応答は 1 つだけです。これは、1 つの WEB SEND コマンドを使用した標準の応答である場合もあれば、一連の WEB SEND コマンドを使用したチャンク化された応答である場合もあります。

CICS は、Web クライアントからの要求を受信するたびに、要求を処理する新しいタスクを開始します。

オプション

ACTION(*cvda*)

メッセージの送信方法を指定します。HTTP サーバーとしての CICS に適用される CVDA 値は、次のとおりです。

IMMEDIATE

Web クライアントに即時に応答を送信します。CHUNKING が指定されていると、IMMEDIATE オプションが想定されます。チャンク形式転送コーディングを使用しないメッセージ送信では EVENTUAL がデフォルトです。これにより、応答はタスクの終わりに送信されます。

EVENTUAL

タスクの終了時に Web クライアントに応答を送信します。CHUNKING が指定されていると、EVENTUAL オプションは無視されます。このオプションを指定すると、CICS Transaction Server for z/OS バージョン 3 リリース 1 より前のリリースの CICS Web サポートと同じ動作が生成されます。このオプションは、HTTP サーバーとしての CICS のデフォルトの設定です。

CHARACTERSET(*data-value*)

CICS が、コマンドから送信された項目のエンティティー本体を送信前に変換する際の変換後の文字セットを指定します。文字セットの名前には、適切な句読点を含む 40 文字以内の英数字を指定することができます。CICS では、IANA によって指定されるすべての文字セットがサポートされるわけではありません。

CHARACTERSET オプションが指定されていると

SERVERCONV(SRVCONVERT) が想定されるため、エンティティー本体のコード・ページ変換が実行されます。文字セットを自分で指定する代わりに、SERVERCONV(SRVCONVERT) か HOSTCODEPAGE (許可されている場合)、またはその両方を指定して CHARACTERSET を省略すると、CICS は、メッセージ本文に適した文字セットを決定します。

すべてのコード・ページ変換オプションを省略すると、コード・ページ変換は実行されません。

CHUNKING(*cvda*)

メッセージがチャンク形式で送信されている場合 (チャンク形式転送コーディングと呼ばれる) のメッセージの制御に使用されます。オプションが指定されていない場合のデフォルトでは、チャンク形式転送コーディングは未使用です。チャンク形式転送コーディングは HTTP/1.1 クライアントでのみ許容されます。HTTP/1.0 クライアントまたは非 HTTP メッセージでは使用できません。

アプリケーション・プログラムにとって最も便利な方法を使用して、チャンク化メッセージの内容をチャンクに分割することができます。チャンク化メッセージの本文は、CICS 文書から直接形成できません。このため、DOCTOKEN オプションは使用できません。

メッセージのチャンクごとに、CHUNKING(CHUNKYES) が指定された個別の WEB SEND コマンドを使用します。FROM オプションを使用してデータのチャンクを指定し、FROMLENGTH オプションを使用してチャンクの長さを指定

します。シーケンスの最初の WEB SEND コマンドでは、メッセージに対して CLOSESTATUS オプションのようなその他のオプションを指定しますが、後続のコマンドでは指定しません。データの最後のチャンクを送信し終わったら、CHUNKING(CHUNKEND) が指定された追加の WEB SEND コマンドを指定します。FROM オプションまたは FROMLENGTH オプションは指定しません。その後 CICS は、空のチャンクを受信側に送信して、チャンク化メッセージを終了します。

CVDA 値は、以下のとおりです。

CHUNKNO

メッセージにチャンク形式転送コーディングは使用されません。CHUNKING オプションが指定されていない場合、これがデフォルトです。

CHUNKYES

チャンク形式転送コーディングが実行中です。FROM オプションで指定されたデータは、メッセージのチャンクを表します。

CHUNKEND

チャンク形式転送コーディングが完了しました。この送信にはデータは指定されません。CICS は空のチャンクを受信側に送信して、チャンク化メッセージを終了します。

CLNTCODEPAGE(*data-value*)

このオプションがサポートされるのは、マイグレーションを目的とする場合のみです。このオプションは、CHARACTERSET によって置き換えられます。どちらのキーワードを使用しても CICS によって実行されるアクションは同じです。これはつまり、SERVERCONV オプションが指定されていなくても、CLNTCODEPAGE が指定されているとコード・ページ変換が実行されることを意味します。すべてのコード・ページ変換オプションが省略されると、コード・ページ変換は実行されません。

CLOSESTATUS(*cvda*)

CICS がメッセージを送信した後、接続を終了するかどうかを指定します。デフォルトは、接続を終了しないことです。CVDA 値は、以下のとおりです。

CLOSE

CICS は、この応答に対して「close」接続オプション (Connection: close) が指定された接続ヘッダーを作成し、応答を送信した後、Web クライアントとの接続をクローズします。このヘッダーによって、Web クライアントにクローズが通知されます。(HTTP/1.0 レベルの Web クライアントの場合、CICS は Connection: Keep-Alive ヘッダーを省略することによって、同じ効果を得ます。)

チャンク形式転送コーディングが使用中の場合は、メッセージの最初のチャンクに CLOSESTATUS(CLOSE) オプションを指定することにより、Web クライアントに、チャンク化メッセージが完了したら接続を終了することを通知することができます。

NOCLOSE

この応答で Connection: close ヘッダーを使用せず、接続をオープンのまま維持することを意味します。Web クライアントが HTTP/1.0 と識別され、「Keep-Alive」接続オプションを指定した Connection ヘッダー

(Connection: Keep-Alive) を送信している場合、CICS は、同じヘッダーを送信して、持続接続が維持されることを通知します。

DOCTOKEN(data-value)

メッセージ本文として送信される文書の 16 バイトのバイナリー・トークンを指定します。この文書は、CICS Document インターフェース (EXEC CICS DOCUMENT CREATE、INSERT、および SET コマンド) を使用して作成されます (「*CICS Application Programming Guide*」に説明があります)。FROM オプションを指定すると、メッセージ本文を作成するための別の方法を使用できません。

チャンク化メッセージの本文は、CICS 文書からは形成できません。このため、チャンク形式転送コーディングでは DOCTOKEN オプションを使用できません。

CICS の文書は UTF-8 および UTF-16 文字エンコードに変換することはできません。

FROM(data-area)

完全なメッセージ本文、またはメッセージ本文のチャンクを保持するデータのバッファを指定します。メッセージ本文は、アプリケーション・プログラムによって作成されます。FROM オプションを指定する場合は、FROMLENGTH オプションを使用してデータのバッファ長を指定します。DOCTOKEN オプションを指定すると、メッセージ本文を作成するための別の方法を使用できますが、このオプションをチャンク化メッセージの本文に使用することはできません。

FROMLENGTH(data-value)

FROM オプションで指定したデータのバッファの長さを、フルワード・バイナリー値として指定します。この値のデータ長が誤っていると、メッセージの受信側で問題が起きる可能性があるため、値を正しく指定することが重要です。

HOSTCODEPAGE(data-value)

アプリケーション・プログラムが応答のエンティティ本体に使用した CICS (ホスト) コード・ページの 8 文字の名前を指定します。HOSTCODEPAGE オプションが指定されていると SERVERCONV(SRVCONVERT) が想定されるため、エンティティ本体のコード・ページ変換が実行されます。

SERVERCONV(SRVCONVERT) か CHARACTERSET、またはその両方を指定して HOSTCODEPAGE を省略すると、CICS がホスト・コード・ページを識別します。

CICS 文書を使用して応答本文を形成する場合、HOSTCODEPAGE オプションを指定しないでください。CICS は、文書のホスト・コード・ページの CICS 文書ドメインのレコードからホスト・コード・ページを識別するためです。

データのバッファを使用して応答本文を形成する (FROM オプション) 場合、HOSTCODEPAGE を指定する必要があります。このオプションを指定しない場合のデフォルトは、LOCALCCSID システム初期設定パラメーターに指定されているローカル CICS 領域のデフォルトのコード・ページです。コード・ページ変換が必要で、アプリケーションで異なるコード・ページが使用されている場合、HOSTCODEPAGE を使用してコード・ページを指定します。

LENGTH(data-value)

このオプションがサポートされるのは、マイグレーションを目的とする場合のみです。このオプションは、STATUSLEN によって置き換えられます。

MEDIATYPE(*data-value*)

メッセージ本文のデータ内容 (例: text/xml) を指定します。メディア・タイプは、適切な句読点を含む 56 文字以内の英数字です。CICS ではメディア・タイプのフォーマットが正しいかどうかは検査されますが、データの内容に対するメディア・タイプの妥当性は検査されません。CICS ではデフォルトは提供されません。

SERVERCONV(*cvda*)

CICS が、このコマンドによって送信される項目のエンティティ本体を、送信前にアプリケーションが使用するコード・ページから受信側に適した文字セットに変換するかどうかを指定します。このコマンドの CHARACTERSET オプションおよび HOSTCODEPAGE オプションを使用して、使用する文字セットおよびコード・ページを指定できます。これらのオプションのどちらかを指定すると、コード・ページ変換 (SRVCONVERT) が想定されます。その代わりに、これらのオプションの片方または両方を省略して SERVERCONV(SRVCONVERT) を指定すると、CICS が適切な文字セットとコード・ページを決定します。

SRVCONVERT

CICS により、メッセージのエンティティ本体が変換されます。

NOSRVCONVERT

CICS は HTTP 要求のエンティティ本体を変換しないため、要求はアプリケーションが使用するコード・ページの形式でリモート・サーバーに送信されます。NOSRVCONVERT を指定すると、CHARACTERSET オプションや HOSTCODEPAGE オプションは指定できません。

注: すべてのコード・ページ変換オプション

(SERVERCONV、CLNTCODEPAGE、CHARACTERSET、HOSTCODEPAGE) を省略すると、コード・ページ変換は実行されません。

STATUSCODE(*data-value*)

アプリケーション・プログラムによって判別された標準の HTTP 状況コードを指定します。この状況コードは、HTTP 応答の状況表示行に挿入されます。このコードは、ハーフワード・バイナリー値です。例として、200 (通常応答) や 404 (検出されない) などがあります。このオプションを指定しなかった場合、CICS はデフォルトの 200 を提供します。

状況コード 204、205、および 304 では、メッセージ本文の指定が許可されないため、メッセージ本文を挿入しようとする、CICS がコマンドにエラー応答を戻します。これ以外では、CICS は状況コードの使用法が適切かどうかの検査を行いません。

STATUSLEN(*data-value*)

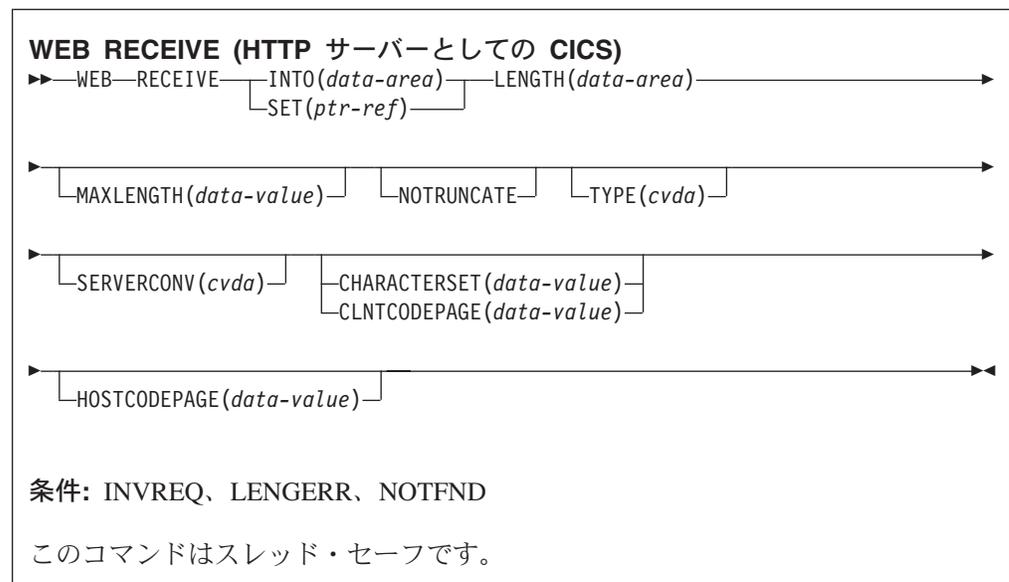
STATUSTEXT オプションで指定したstringの長さを、フルワード・バイナリー値として指定します。

STATUSTEXT(*data-area*)

状況コードの理由を記述するために人間が読めるテキストを格納するデータ域を指定します。このテキストは、理由句と呼ばれています。例として、「了解」(状況コード 200 と共に戻される) や「不正な要求 (Bad Request)」(状況コード 400 と共に戻される) があります。

WEB RECEIVE (サーバー)

HTTP 要求または非 HTTP メッセージを受信します。



説明

WEB RECEIVE により、HTTP 要求の本文、または非 HTTP メッセージのすべてのデータがアプリケーション提供のバッファーに受信されます。HTTP 要求のヘッダーは、WEB HTTPHEADER コマンドを使用して個別に調べることができます。WEB RECEIVE コマンドによって受信できる項目は、次のとおりです。

- HTTP サポートとしての CICS に対して Web クライアントが行った HTTP 要求の本文。
- CICS Web サポート機能によって処理される非 HTTP メッセージ。
TCPIPService 定義にユーザー定義の (USER) プロトコルが指定されます。
- CICSHTTP リスナー経由ではなく、CICS ビジネス・ロジック・インターフェースを使用して直接アプリケーション・プログラムに接触した別のプログラムからの要求。

データは、そのエスケープ形式で戻されます。CICS アプリケーション・プログラムが受信した着信データに使用するコード・ページ変換のタイプを、このコマンドで指定することができます。

オプション

CHARACTERSET(*data-value*)

受信した項目のエンティティ本体に関して、Web クライアントが使用した文字セットを指定します。文字セットの名前には、適切な句読点を含む 40 文字以内の英数字を指定することができます。CICS では、IANA によって指定されるすべての文字セットがサポートされるわけではありません。

CHARACTERSET オプションが指定されていると SERVERCONV(SRVCONVERT) が想定されるため、エンティティ本体のコー

ド・ページ変換が実行されます。文字セットを自分で指定する代わりに、SERVERCONV(SRVCONVERT) か HOSTCODEPAGE、またはその両方を指定して CHARACTERSET を省略すると、CICS はメッセージ本文の文字セットを識別します。

CLNTCODEPAGE(*data-value*)

このオプションがサポートされるのは、マイグレーションを目的とする場合のみです。このオプションは、CHARACTERSET によって置き換えられます。どちらのキーワードを使用しても CICS によって実行されるアクションは同じです。これはつまり、SERVERCONV オプションが指定されていなくても、CLNTCODEPAGE または HOSTCODEPAGE が指定されていると、コード・ページ変換が実行されることを意味します。すべてのコード・ページ変換オプションが省略されると、コード・ページ変換は実行されません。

HOSTCODEPAGE(*data-value*)

アプリケーション・プログラムが使用する CICS (host) コード・ページの 8 文字の名前を指定します。このコード・ページ形式は、受信した項目のエンティティ本体を文字セット (Web クライアントから受信した形式) から変換した後の形式になります。HOSTCODEPAGE オプションが指定されていると SERVERCONV(SRVCONVERT) が想定されるため、エンティティ本体のコード・ページ変換が実行されます。SERVERCONV(SRVCONVERT) か CHARACTERSET、またはその両方を指定して HOSTCODEPAGE を省略すると、CICS がホスト・コード・ページを決定します。

このオプションを指定しない場合のデフォルトは、LOCALCCSID システム初期設定パラメーターに指定されている、ローカル CICS 領域のデフォルトのコード・ページです。

INTO(*data-area*)

受信するデータを格納するためのバッファを指定します。

LENGTH(*data-area*)

CICS からアプリケーションに戻されたデータ量に設定されるフルワード・バイナリー変数が指定されます。

- NOTTRUNCATE オプションが指定されていない場合、メッセージ内に存在する追加データは廃棄されています。追加データが存在していた場合、RESP2 値に 57 が設定された LENGERR 応答が戻されます。
- NOTTRUNCATE オプションが指定されている場合、追加データは保存されます。追加データが使用可能な場合、RESP2 値に 36 が設定された LENGERR 応答が戻されます。この場合に実行すべき操作は、NOTTRUNCATE オプションの説明で示しています。

MAXLENGTH(*data-value*)

CICS によってアプリケーションに渡されるデータの最大量をフルワード・バイナリー値として指定します。MAXLENGTH オプションは、データを受信するときに INTO オプションまたは SET オプションのいずれかを指定すると適用されます。チャンク形式転送コーディングを使用してデータを送信すると、CICS は大量のデータを 1 つのメッセージに組み立ててからアプリケーションに送信するため、MAXLENGTH オプションは、個々のチャンク (まとめ) に対してではなく、チャンク化メッセージの全長に対して適用されます。コード・ページの変換が実行された後のデータが測定されます。

NOTRUNCATE

使用可能なデータが、MAXLENGTH オプションで要求されている長さを超えた場合、残りのデータを即座には廃棄せず、保存して後続の RECEIVE コマンドによる検索に備えることを指定します。(RECEIVE コマンドがその後発行されなかった場合は、トランザクションの終了時にデータは廃棄されます。)

SET オプションを指定し、MAXLENGTH オプションを指定せずに単一の RECEIVE コマンドを実行すると、長さに関係なく、残りのデータすべてが受信されます。あるいは、一連の RECEIVE コマンドに NOTRUNCATE オプションを指定して実行し、残りのデータを該当するチャンク (まとまり) として受信することができます。LENGERR 応答を取得しなくなるまで、RECEIVE コマンドを発行し続けます。

SERVERCONV(*cvda*)

CICS が、受信した項目のエンティティ本体を、Web クライアントが使用する文字セットからアプリケーションに適したコード・ページに変換するかどうかを指定します。このコマンドの CHARACTERSET オプションおよび HOSTCODEPAGE オプションを使用して、使用する文字セットおよびコード・ページを指定できます。これらのオプションのどちらかを指定すると、コード・ページ変換 (SRVCONVERT) が想定されます。その代わりに、これらのオプションの片方または両方を省略して SERVERCONV(SRVCONVERT) を指定すると、CICS が適切な文字セットとコード・ページを決定します。

SRVCONVERT

CICS により、メッセージのエンティティ本体が変換されます。

NOSRVCONVERT

CICS は、項目のエンティティ本体を変換しないため、項目は Web クライアントが使用する文字セットでアプリケーションに渡されます。NOSRVCONVERT を指定すると、CHARACTERSET オプションや HOSTCODEPAGE オプションは指定できません。

注: すべてのコード・ページ変換オプション

(SERVERCONV、CLNTCODEPAGE、CHARACTERSET、HOSTCODEPAGE) を省略すると、コード・ページ変換は実行されません。

SET(*ptr-ref*)

受信したデータのアドレスの設定先となるポインター参照を指定します。ポインター参照は、次の RECEIVE コマンドまたはタスクの終了まで有効です。

TYPE(*cvda*)

受信した要求のタイプを戻します。CVDA 値は、以下のとおりです。

HTTPYES

HTTP 要求を示します。

HTTPNO

HTTP 以外の要求を示します。

システム・プログラミング・インターフェースの変更点

CREATE TCPIP SERVICE、INQUIRE TCPIP SERVICE、および SET TCPIP SERVICE コマンドに MAXDATALEN オプションが追加されました。

MAXDATALEN(32|number)

HTTP サーバーとしての CICS が受信できるデータの最大長 (キロバイト単位) を指定します。デフォルト値は 32 です。最小値は 3 最大値は 524288 です。

CREATE TCPIP SERVICE および INQUIRE TCPIP SERVICE コマンドでも、PROTOCOL 属性の新しいオプション USER (ユーザー定義プロトコルを意味する) が使用可能になりました。

プロトコルを USER として定義した場合、新しい CICS 提供のトランザクション CWXU (CICS Web ユーザー定義プロトコル接続トランザクション) が、トランザクションとして指定されます。CWXU はプログラム DFHWBXN を実行します。

HTTP プロトコルの場合は、CREATE TCPIP SERVICE コマンドで SOCKETCLOSE オプションに 0 を指定しないでください。この設定にすると持続接続を維持できなくなります。

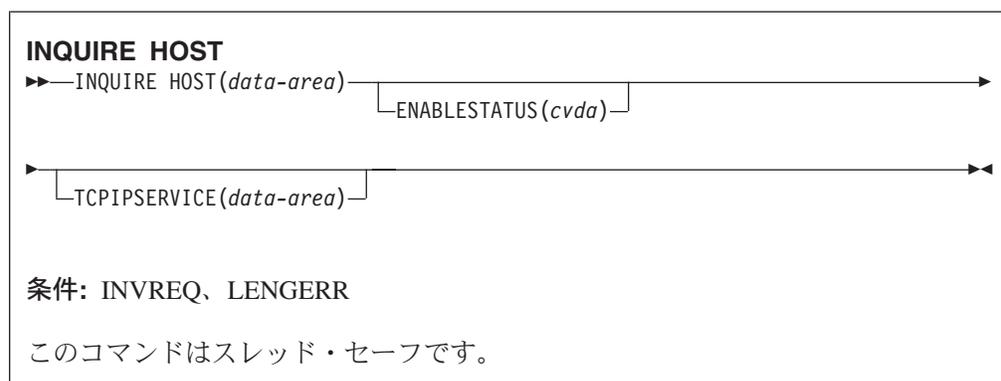
仮想ホストを管理するために、以下のコマンドが提供されています。

- INQUIRE HOST
- SET HOST

CICS では、ご使用の CICS 領域の URIMAP 定義に基づいて、仮想ホストの作成と削除を自動的に管理します。

INQUIRE HOST

ローカル・システム内の仮想ホストに関する情報を取得します。



説明

INQUIRE HOST コマンドを使用すると、ローカル CICS 領域内の特定の仮想ホストに関する情報を取得できます。仮想ホストについては、113 ページの『仮想ホスティング』で説明しています。仮想ホストは、URIMAP リソース定義オブジェクトに基づいています。CICS では、同じホスト名と同じ TCPIP SERVICE を指定する CICS 領域に存在するすべての URIMAP 定義を 1 つのデータ構造にまとめることにより、仮想ホストが自動的に作成されます。TCPIP SERVICE を指定しない URIMAP は、すべての TCPIP SERVICE に適用されることになるため、一致するホスト名を指定するすべてのデータ構造に追加されます。このため、これらの URIMAP 定義は複数のデータ構造の一部になる可能性があります。

ブラウズ

INQUIRE HOST コマンドでブラウズ・オプション (START、NEXT、および END) を使用することにより、領域内に存在するすべての仮想ホストをブラウズすることもできます。

オプション

HOST(*data-value*)

仮想ホストの名前を指定します。各仮想ホストの名前は、仮想ホストを構成する URIMAP 定義で指定されたホスト名から取得されます。例えば、CICS 領域の URIMAP 定義でホスト名 `www.example.com` が指定されている場合、CICS は `www.example.com` という名前の仮想ホストを作成します。URIMAP 定義には、ホスト名を 120 文字以内で指定できます。

ENABLESTATUS(*cvda*)

この仮想ホストの状況を示す CVDA 値を戻します。CVDA 値は、以下のとおりです。

ENABLED

仮想ホストは使用可能です。

DISABLED

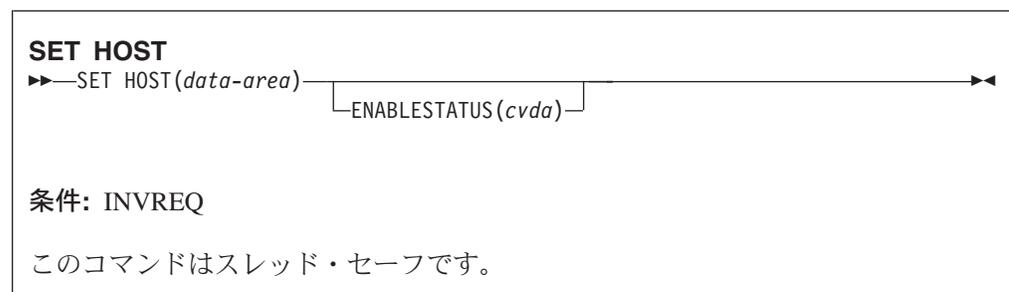
仮想ホストは使用不可です。アプリケーションは、仮想ホストを構成する URIMAP 定義にアクセスできません。

TCPIPSERVICE(*data-area*)

この仮想ホストに関連するインバウンド・ポートを指定する TCPIPSERVICE 定義の 1 から 8 文字の名前を戻します。この定義が指定されなかった場合、仮想ホストはすべての TCPIPSERVICE 定義に関連付けられます。

SET HOST

仮想ホストの状況を使用可能または使用不可に設定します。



説明

SET HOST コマンドは、仮想ホストの状況を使用可能または使用不可に設定します。仮想ホストを使用不可に設定することは、仮想ホストを構成するすべての URIMAP 定義にアプリケーションがアクセスできなくなることを意味します。仮想ホストを使用不可にした場合、CICS は Web クライアントに対して、503 (サービスが利用不可能) 状況コードを含む HTTP 応答を戻します。

INQUIRE URIMAP コマンドを使用して個々の URIMAP 定義について照会すると、仮想ホストが使用不可であることを示す特殊な状況 `DISABLEDHOST` が戻されます。URIMAP 定義の使用不可状況を個別に変更する必要はありません。SET HOST コマンドを使用すれば、仮想ホストを構成するすべての URIMAP 定義を再び使用可能にすることができます。ただし、`DISABLEDHOST` 状況の URIMAP 定義は廃棄できないので、注意してください。定義を廃棄したい場合は、(SET URIMAP コマンドを使用して) 個別に廃棄する必要があります。

オプション

ENABLESTATUS(*cvda*)

CVDA 値は、以下のとおりです。

ENABLED

アプリケーションは、仮想ホストを構成する URIMAP 定義にアクセスできます。

DISABLED

アプリケーションは、仮想ホストを構成する URIMAP 定義にアクセスできません。

CEMT の変更点

CEMT INQUIRE および SET HOST コマンドが CEMT トランザクションに追加されました。「*CICS Supplied Transactions*」に、これらのコマンドについての情報が記載されています。

統計の変更点

新規の統計 Maxdata (フィールド名 `SOR_MAXDATA_LENGTH`) が TCP/IP サービス・リソース統計に追加されます。この統計は、TCPIPSERVICE 定義の `MAXDATALEN` 設定を表示します。TCP/IP サービス・リソース統計は、TCPIPSERVICE キーワードを使用して COLLECT STATISTICS コマンドによって収集され、DFHSORDS DSECT によってマップされます。

統計プログラム DFHOSTAT には、CICS が URIMAP 定義から作成した仮想ホストを示す新規レポートが組み込まれています。仮想ホスト・レポートには、各仮想ホストの名前、それが使用可能か使用不可か、および関連付けられている TCPIPSERVICE 定義の名前が表示されます。TCPIPSERVICE 定義が表示されない場合、その仮想ホストは CICS 領域内のあらゆる TCPIPSERVICE 定義に関連付けられています。

第 5 章 CICS Web サポートの一般拡張

CICS Web サポートの拡張には、CICS が HTTP サーバーの場合の HTTP 要求および応答処理に関する機能の改良や、CICS Web サポートによるコード・ページ変換の処理方法の改良、およびより適切な HTTP 日時形式のサポートが含まれます。

HTTP 要求および応答処理のサポートの改良

URIMAP 定義は、CICS が HTTP サーバーの場合の HTTP 要求および応答処理に関して改良されてより強力になった機能を提供する、新しい CICS リソースです。新しい URIMAP リソース定義によって、以下が可能になります。

- アプリケーション・プログラムを使用して Web クライアントに動的応答を提供するだけでなく、HFS ファイルまたは CICS 文書テンプレートの内容を使用して静的応答を提供できます。
- より透過的に HTTP 要求を処理できます。
- HTTP 要求のオンライン管理を実行できます。
- HTTP 要求の処理にアナライザー・プログラムを使用するかどうかはオプションです。

HTTP 要求および応答を分析するための CICS API サポートが拡張されました。EXEC CICS WEB EXTRACT コマンドは、より多くの情報を要求から抽出するように拡張されました。また、新規 EXEC CICS WEB PARSE URL コマンドによって、任意の URL を分析して再利用のための情報を抽出できるようになりました。

HTTP サーバーとしての CICS で使用される EXEC CICS WEB SEND および WEB RECEIVE コマンドが拡張されて、Web クライアントから要求を受信したり、応答を送信したりするときの制御や機能が向上しました。例えば、応答のメッセージ本体を提供するのに (CICS 文書を使用する代わりに) データのバッファを使用できるようになりました。また、応答がいつ送信されるかを決定するオプションや、応答受信後に CICS がクライアントに接続を終了するよう信号を送るかどうかを決定するオプションを指定できます。これらのコマンドの変更点については、107 ページの『第 4 章 HTTP/1.1 への CICS Web サポートのアップグレード』で説明します。

コード・ページのサポートの改良

CICS Web サポートのコード・ページ変換が改良されたため、CICS Web サポート用のコード・ページ変換テーブル (DFHCNV) をセットアップする必要がなくなりました。

CICS では、コード・ページ変換に UTF-8 および UTF-16 文字エンコードを使用できるようになりました。

HTTP 日時形式のサポート

新しい CICS API コマンド CONVERTTIME は、インターネットで使用される一般的な日時スタンプ形式を CICS ABSTIME 形式に変換します。EXEC CICS FORMATTIME コマンドには、ABSTIME 形式をインターネットでの使用に適した

日時スタンプ・ストリングに変換するためのオプションが追加されました。

CICS Web サポート拡張の利点

CICS Web サポートの拡張によって、以下の利点が得られます。

- HTTP サーバーとしての CICS は、アナライザー・プログラムの代わりに、あるいはこれに加えて、URIMAP リソース定義を使用することによって、より簡単に管理できます。URIMAP 定義は、HTTP 要求をユーザー・アプリケーション・プログラムに直接割り当てるのに使用できます。この URIMAP 定義はユーザーがオンラインで管理できます。
- URIMAP 定義を使用すると、CICS が Web クライアントに対して自動的に静的応答を提供するようにすることができるため、このタイプの応答を提供するアプリケーション・プログラムを作成して呼び出す必要がありません。静的応答は、CICS 文書テンプレートまたは z/OS UNIX[®] システム・サービス HFS ファイルから形成することが可能であり、この応答は Web クライアントの照会ストリングからデータを組み込むことができます。
- コード・ページ変換に UTF-8 および UTF-16 文字エンコードを使用できます。CICS Web サポートのコード・ページ変換プロセスが改良されたため、CICS Web サポート用にコード・ページ変換テーブル (DFHCNV) をセットアップする必要がなくなりました。
- CICS API は、インターネットで使用される日時スタンプ形式と CICS ABSTIME 形式の間の変換をサポートします。

用語

以下の用語は、CICS Web サポートの拡張に関連して使用されます。

HTTP サーバーとしての CICS

CICS が Web クライアントから HTTP 要求を受信し、応答を送信するプロセス。ユーザー・アプリケーション・プログラムを使用して要求を処理し、応答を提供するか、URIMAP 定義を使用して静的応答を指定することができます。

HTTP クライアントとしての CICS

ユーザー・アプリケーション・プログラムが CICS を介して HTTP サーバーに要求を送信し、応答を受信するプロセス。

静的応答

CICS が、URIMAP 定義によって指定された文書テンプレートまたは HFS ファイルから構成する HTTP 応答。

アプリケーションにより生成される応答

ユーザー・アプリケーション・プログラムによって動的に作成される HTTP 応答。これは、Web 対応アプリケーション・プログラムでも Web 非対応アプリケーション・プログラムでも構いません。

Web 対応アプリケーション・プログラム

EXEC CICS WEB アプリケーション・プログラミング・インターフェース・コマンドを使用して Web クライアントの要求を受信し、HTTP 応答を送信するアプリケーション・プログラム。

Web 非対応アプリケーション・プログラム

CICS Web サポートで、EXEC CICS WEB アプリケーション・プログラミング・インターフェース・コマンドを使用しないアプリケーション・プログラム。このようなプログラムを Web に対応できるようにするには、コンバーター・プログラムを使用して、Web クライアントの要求を受け入れ可能な入力に変換し、HTTP 応答をプログラムの出力に基づいて構成します。

Web クライアント

HTTP サーバーとしての CICS に対して HTTP 要求を行うクライアント・アプリケーション。これは、ユーザー (人間) や自動ユーザー・エージェント (検索エンジン用の情報収集機能など)、あるいはアプリケーション・プログラム (HTTP クライアント要求を行う CICS アプリケーションなど) への応答を表示する Web ブラウザーである場合があります。

チャンク形式転送コーディング (チャンク化とも呼ばれる)

HTTP メッセージの本文が一連のチャンク (それぞれに固有のチャンク・サイズ・ヘッダーが付いている) として転送されるプロセス。

パイプライン化

クライアントが複数の HTTP 要求を、応答を待たずにサーバーに送信するプロセス。応答は、要求を受信したときと同じ順序でリモート・サーバーから戻す必要があります。

べき等 個別 HTTP メソッド、またはパイプライン化された一連の要求のプロパティ。メソッドがべき等である場合、そのメソッドで同じ要求を繰り返すと必ず同じ結果が得られます。要求シーケンスがべき等である場合、一連の要求の全部または一部が繰り返されると必ず結果が得られます。

持続接続

要求と応答の交換を複数回行う場合に再利用できる、Web クライアントと HTTP サーバー間の接続。

URL URL (Uniform Resource Locator) は、特定のタイプの URI (Universal Resource Identifier) です。URI は、どのようなリソースでも指定することができますが、URL は通常インターネット上の既存のリソースの位置を指定します。

仮想ホスティング

同じ IP アドレスにある複数のホストを 1 つの HTTP サーバーが代表できる状態。この方法で提供される各ホスト名は、仮想ホストとして認識されません。

要件

この機能をサポートするための特別なハードウェア要件およびソフトウェア要件はありません。

関連情報

377 ページの『第 27 章 CICS 稼働環境』

HTTP サーバーとしての CICS に対する HTTP 要求と応答の処理

HTTP サーバーとしての CICS に対する HTTP 要求は、CICS への要求を行う Web クライアントによって開始されます。CICS は、Web クライアントからの要求の応答を Web クライアントに戻します。応答は、URIMAP リソース定義に指定されている静的文書を基にして作成するか、またはユーザーのアプリケーション・プログラムによって動的に作成できます。

HTTP サーバーとしての CICS に対する処理は、以下のように実行されます。

1. **CICS が TCP/IP 接続の要求を受信します。** CICS Sockets ドメインは、ポートの TCPIPService リソース定義を使用して、要求が CICS Web サポートによって処理されることを決定します。
2. **CICS は、要求の URL を (使用可能な場合) URIMAP 定義と突き合わせます。** 一致する定義が検出された場合は、その URIMAP 定義によって要求の処理方法が CICS に通知されます。
3. **URIMAP 定義がリダイレクトを指定している場合、CICS は Web クライアントの URL から指定の URL にリダイレクトします。** CICS は、リダイレクト・メッセージを作成して、これを Web クライアントに送信します。
4. **URIMAP 定義によって静的応答が指定されている場合、CICS は応答を生成して出力します。** CICS は、文書テンプレートまたは z/OS UNIX System Services HFS ファイルのいずれかと適切な HTTP ヘッダーを組み合わせて使用することにより、HTTP 応答を生成します。この応答には、適切なコード・ページ変換が実行され、その後 CICS によって Web クライアントに送信されます。
5. **アナライザー・プログラムの使用が URIMAP 定義に指定されている場合や、一致する URIMAP 定義が見つからなかった場合は、アナライザー・プログラムが実行される場合があります。** アナライザー・プログラムは、要求を動的に解釈できます。または、アナライザー・プログラムをモニターや監査の目的に使用することもできます。
6. **アプリケーション・プログラムを実行して、要求を処理します。** アプリケーション・プログラムを指定するには、URIMAP 定義、アナライザー・プログラムのいずれかを使用します。EXEC CICS WEB および EXEC CICS DOCUMENT アプリケーション・プログラミング・インターフェースを使用する Web 対応アプリケーション・プログラムを使用すると、要求を処理して応答を構成できます。非 Web 対応アプリケーション・プログラムを Web に対応できるようにするには、コンバーター・プログラム (Web クライアントの要求を受け入れ可能な入力に変換し、プログラムの出力に基づいて HTTP 応答を作成するプログラム) か、非 Web 対応プログラムを呼び出してその出力を使用する Web 対応アプリケーション・プログラムのいずれかを使用します。
7. **CICS は、必要な HTTP ヘッダーをいくつか生成して、これをメッセージに追加します。** 応答の HTTP バージョンに応じて、適切なヘッダーが生成されます。
8. **CICS は、完全な HTTP 応答を Web クライアントに送信します。** Web クライアントが持続接続をサポートする場合、CICS は、ユーザー・アプリケーションまたは Web クライアントが終了を要求するか、またはタイムアウト期間に到達するまで接続を維持し、その後の HTTP 要求に備えます。

この処理中、コード・ページ変換が必要になるのは、通常、メッセージが CICS 環境に出入りするときであるため、CICS Web サポート処理およびユーザー作成アプリケーション (通常は EBCDIC エンコードを使用) は、Web クライアント (通常は ASCII エンコードを使用) と通信できます。

CICS Web サポートでの Unicode UTF-8 および UTF-16 コード・ページ変換

CICS Web サポートは、Unicode UTF-8 文字エンコードおよび UTF-16 文字エンコードとの間のコード・ページ変換を処理できるようになりました。

UTF-8 および UTF-16 のコード・ページ変換では、z/OS が提供する変換サービスを使用します。「*z/OS Support for Unicode: Using Conversion Services*」の説明に従い、ご使用の z/OS システムで変換機能を使用可能にする必要があります。

コード・ページ変換は通常は、CICS Web サポート処理およびユーザー作成のアプリケーション (一般に EBCDIC エンコードが使用される) が、Web クライアント (一般に ASCII エンコードが使用される) と通信できるようにするために必要です。インバウンド・メッセージは、アプリケーションが処理できるよう EBCDIC エンコードに変換され、アプリケーションが生成したアウトバウンド・メッセージは、EBCDIC エンコードから、Web クライアントに適した文字セットに変換されます。

UTF-8 および UTF-16 文字エンコードを、CICS がインターネットとの間で送受信する項目のメッセージ本体に使用することができます。CICS がサポートする任意の EBCDIC コード・ページを使用して、UTF-8 および UTF-16 文字エンコードとの間の変換を実行できます。

CICS 文書および文書テンプレートは、UTF-8 および UTF-16 文字エンコードに変換できないため、注意してください。これらの文字エンコードでアウトバウンド・メッセージを送信したい場合は、DOCTOKEN オプションを使用して CICS 文書を指定するのではなく、WEB SEND または WEB CONVERSE コマンドで FROM オプションを使用して、メッセージ本体を形成するデータのバッファを指定します。

HTTP 日時スタンプ・フォーマットの処理

CICS を介して Web と対話するアプリケーション・プログラムは、適切な日時スタンプ・フォーマットを HTTP ヘッダーに生成する必要があります。また、アプリケーション・プログラムは、Web から受信した日時スタンプ情報を処理できなくてはなりません。CONVERTTIME および FORMATTIME コマンドを使用すると、HTTP で使用される一般的な設計済み日時スタンプ・ストリング・フォーマットを処理できます。

CONVERTTIME コマンドを使用すれば、設計済み日時スタンプ・ストリングを ABSTIME に変換できます。日時スタンプのフォーマットをユーザーが識別する必要はありません。CONVERTTIME コマンドは、インターネットで一般に使用されている 3 つの異なる日時スタンプ・フォーマットを認識および変換します。

FORMATTIME コマンドを使用すれば、現在日付および時刻 (ABSTIME フォーマット) またはアプリケーション・プログラムによって生成された日時 (有効期限日付など) を RFC 1123 フォーマットに変換できます。

CICS 外部インターフェースの変更点

リソース定義の変更点

DOCTEMPLATE リソース定義の新しい属性 HFSFILE

この属性を使用すると、z/OS UNIX システム・サービス HFS ファイルを文書テンプレートとして使用できます。

HFSFILE(*hfsfile*)

テンプレートが z/OS UNIX システム・サービス HFS ファイル内にある場合は、HFS ファイルの完全修飾 (絶対) 名または相対名を指定します。名前は、すべてのディレクトリーが含まれ、スラッシュで始まっている絶対名 (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 HOME ディレクトリーに対する相対名 (例えば、facts/images/bluefish.jpg) として指定することもできます。最大 255 文字まで使用できます。CICS 領域には、z/OS UNIX へのアクセス権が必要であり、このファイルが含まれる HFS ディレクトリーや、ファイル自体へのアクセス権も必要です。

URIMAP リソース定義

URIMAP 定義は、HTTP または Web サービス要求の URI と一致するリソース定義であり、要求を処理する方法に関する情報を提供します。

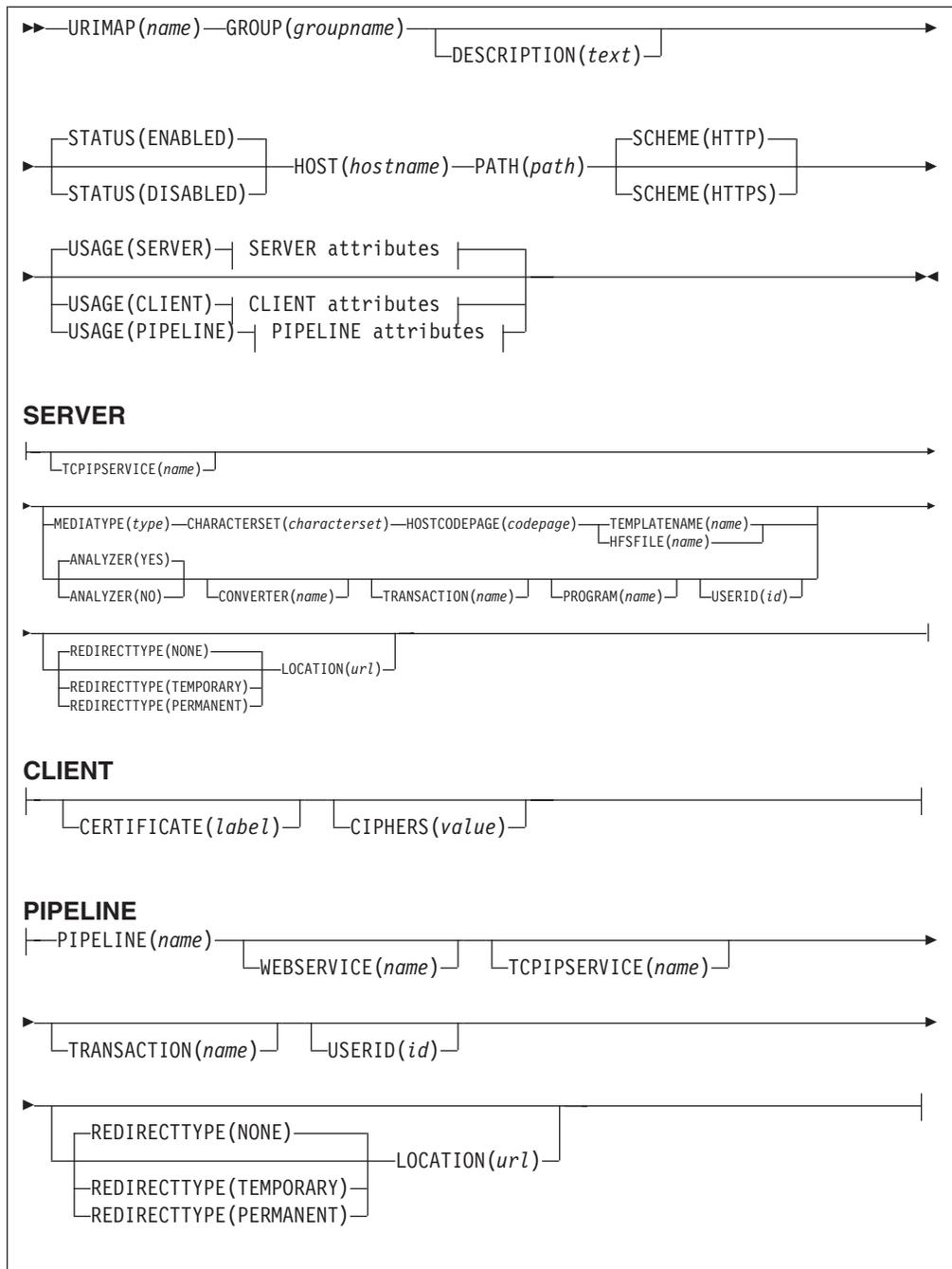
URIMAP 定義は、CICS における次の 3 つの異なる Web 関連機能を提供するために使用されます。

1. **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 に対する要求のための URIMAP 定義は、USAGE 属性として SERVER が設定されています。これらの URIMAP 定義は、CICS が Web クライアントから受信することを予期する HTTP 要求の URL と一致し、各要求に対して CICS から応答を提供する方法を定義します。URIMAP 定義を使用して、CICS に対して次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 HFS ファイルを使用して、HTTP 要求に対して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 要求に対して動的応答を提供する。
2. **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 からの要求の URIMAP 定義では、USAGE 属性として CLIENT が設定されています。これらの URIMAP 定義は、Web クライアントとして機能するユーザー・アプリケーションが HTTP サーバーに対する CICS Web サポートを通じて要求を行う際に使用する URL を指定します。この目的のために URIMAP 定義をセットアップすると、アプリケーション・プログラムでの URL の識別を避けることができます。

3. **Web サービス要求。** Web サービス要求のための URIMAP 定義では、USAGE 属性として PIPELINE が設定されています。これらの URIMAP 定義は、インバウンド Web サービス要求 (すなわちクライアントが CICS で Web サービスを呼び出す要求) の URI を、実行される処理を指定する PIPELINE または WEBSERVICE リソースに関連付けます。

HTTP サーバーとしての CICS の場合、URIMAP 定義には、以前は TCPIPSERVICE 定義に関連付けられたアナライザー・プログラムによって提供されていた機能のほとんどが組み込まれています。必要に応じて、やはり処理のパスでアナライザー・プログラムを関与させることができます。

URIMAP 定義属性:



ANALYZER({NO|YES})

この属性は、アプリケーションにより生成される応答が提供される

USAGE(SERVER) で使用します。(USAGE(CLIENT) または

USAGE(PIPELINE) の場合、この属性は強制的に NO になります。) HTTP 要求の処理にアナライザー・プログラムを使用するかどうかを指定します。この属性を使用して実行できるアナライザーは、この URIMAP 定義が関連している TCPIPSERVICE 定義 (複数可) に関連付けられているアナライザーです。(アナライザー・プログラムはローカル CICS 領域に存在している必要があります。)

YES の場合、アナライザーが実行されます。NO は、アナライザーが使用されないことを意味します。

CERTIFICATE(*label*)

この属性は、**USAGE(CLIENT)** で使用します。SSL ハンドシェイク時に 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。証明書ラベルは、32 文字以内の長さにすることができます。この属性は、URIMAP 定義によって指定される URI が、クライアントとしての CICS によって行われる HTTPS 要求で使用される場合にのみ必要です。SSL クライアント証明書を要求するかどうかはサーバー次第ですが、証明書が要求された場合、CICS では、URIMAP 定義で指定された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内の鍵リングに保管する必要があります。

CIPHERS(*value*)

この属性は、**USAGE(CLIENT)** で使用します。

28 個以内の 2 桁の暗号スイート・コードのリストとして解釈される 56 個以内の 16 進数字のストリングを指定します。属性値には、ENCRYPTION システム初期化パラメーターで指定した暗号化のレベルに応じて、許容されるコードのリストが自動的に取り込まれます。

- ENCRYPTION=WEAK の場合、デフォルト値は 03060102 です。
- ENCRYPTION=MEDIUM の場合、デフォルト値は 0903060102 です。
- ENCRYPTION=STRONG の場合、デフォルト値は 0504352F0A0903060102 です。

暗号コードを再配列したり、デフォルト・リストから除去することができます。ただし、特定の暗号化レベルのデフォルト・リストにない暗号コードを新たに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。これにより、フィールドにデフォルト・リストが再び自動的に取り込まれます。

CHARACTERSET(*character set*)

この属性は、静的な応答が提供される **USAGE(SERVER)** で使用します。Web クライアントに送信される応答のエンティティー本体を CICS が変換する変換先文字セットの 1 から 40 文字の名前を指定します。CICS では、IANA によって指定されるすべての文字セットがサポートされるとは限りません。この属性の値は、応答の Content-Type ヘッダーに組み込まれます。

静的応答が提供されていて、MEDIATYPE 属性がテキスト・タイプを指定している場合は、CHARACTERSET を指定する必要があります。

CONVERTER(*name*)

この属性は、アプリケーションにより生成される応答が提供される **USAGE(SERVER)** で使用します。要求および応答に対して変換またはその他の処理を行うために実行するコンバーター・プログラムの 1 から 8 文字の名前を指定します。一般に、コンバーター・プログラムは、HTTP 要求をアプリケーション・プログラムが使用できる COMMAREA に変換し、その出力を HTTP 応答に変換します。コンバーター・プログラムには、ローカル CICS 領域で使用可能な任意のコンバーター・プログラムを指定できます。

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。使用できる文字につい

ての制限はありません。ただし、括弧を使用する場合は、それぞれの左括弧ごとに、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、グループがインストールされるときに 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。DFH で始まるグループ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済みです。

HFSFILE(*name*)

この属性は、静的な応答が提供される **USAGE(SERVER)** で使用します。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する z/OS UNIX システム・サービス HFS ファイルの完全修飾 (絶対) 名または相対名を指定します。名前は、すべてのディレクトリーが含まれ、スラッシュで始まっている絶対名 (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 HOME ディレクトリーに対する相対名 (例えば、facts/images/bluefish.jpg) として指定することもできます。最大 255 文字まで使用できます。

注: リソース・レベル・セキュリティーは、静的応答として引き渡される項目には適用されません。この方法で引き渡された項目に対してユーザー ID に基づくアクセス制御を適用する必要がある場合は、代わりに、そのデータをアプリケーション生成の応答として引き渡す必要があります。

パス・マッチングを使用するには、HFS ファイルのパスの末尾と、PATH 属性で指定されているパスの末尾に、ワイルドカード文字としてアスタリスクを記述します。CICS では、各 HTTP 要求のパスの中からワイルドカード文字で指定される部分を取り出し、これをファイル・パスの最後の部分として置き換えます。

例えば、PATH 属性を

```
findout/pictures/*
```

と指定し、HFSFILE 属性を以下のように指定した URIMAP 定義を作成できます。

```
/u/facts/images/*
```

この URIMAP 定義は、以下の着信 HTTP 要求を処理するために使用されます。

```
http://www.example.com/findout/pictures/bluefish.jpg
```

CICS では、URIMAP 定義で指定された HFS ファイル・パスにアスタリスクの代わりに bluefish.jpg を付加します。これにより、

```
/u/facts/images/bluefish.jpg
```

が静的応答として使用されます。

注: HFSFILE の指定には、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は、1 つ以上指定する必要があります。

照会ストリングを HFS ファイルに置換することはできません (HFS ファイルを CICS 文書テンプレートとして定義せず、HFSFILE オプションの代わりに TEMPLATENAME オプションを使用して指定しなかった場合)。

HOST(*hostname*)

この属性は、すべての **USAGE** オプションで使用します。URIMAP 定義が適用される URI のホスト部分を 116 文字以内で指定します。ホスト名の例としては、www.example.com を挙げることができます。

HOST 属性が定義されている必要があります。ホスト属性には、英数字、ハイフン (-) またはピリオド (.) のみを含めることができます。ホスト名には、16 進数のエスケープ・シーケンスを使用できません。CICS は、これを定義時に検証します。ホスト名は大文字でも小文字でも入力できますが、URIMAP 定義ではこれは小文字に変換されます。

ホスト名として IPv4 アドレスを使用できますが、IPv6 アドレスはサポートされていません。

HTTP サーバーとしての CICS に関連する URIMAP 定義の場合、HOST 属性の代わりに、単一のアスタリスクを使用できます。これにより、URIMAP 定義は任意のホスト名と一致します。アスタリスクは、HOST 属性でその他の文字と一緒にワイルドカードとして使用することはできません。

HOSTCODEPAGE(*codepage*)

この属性は、静的な応答が提供される **USAGE(SERVER)** で使用します。静的応答を形成するテキスト文書のエンコードに使用する IBM® コード・ページ (EBCDIC) の 1 から 10 文字の名前を指定します。この情報は、静的応答のエンティティー本体のコード・ページ変換を実行するために、CICS で必要となります。

静的応答が提供されていて、MEDIATYPE 属性がテキスト・タイプを指定している場合は、HOSTCODEPAGE を指定する必要があります。

LOCATION(*url*)

この属性は、**USAGE(SERVER)** および **USAGE(PIPELINE)** で使用します。クライアントの要求のリダイレクト先の URL を 255 文字以内で指定します。これは、スキーム部分、ホスト部分、パス部分、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 では URL が有効であるかは検査されないため、宛先が存在すること、およびその URL が正しく指定されていることをユーザーが確認する必要があります。

リダイレクトのタイプを指定するために、REDIRECTTYPE 属性が使用されます。一時リダイレクトまたは永続リダイレクトが指定された場合、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 コマンドを使用して、REDIRECTTYPE 属性および LOCATION 属性を変更することができます。

MEDIATYPE(*type*)

この属性は、静的な応答が提供される **USAGE(SERVER)** で使用します。HTTP 要求に対して CICS が提供する静的応答のメディア・タイプ (データ・コンテンツ) を指定します (例えば、image/jpg、text/html、text/xml など)。最大 56 文字まで使用できます。メディア・タイプには、スラッシュ (/) が 1 つのみ含まれていなければなりません。メディア・タイプは大文字でも小文字でも入力できますが、URIMAP 定義ではこれは小文字に変換されます。

正式に認められた各データ・コンテンツ・タイプの名前は、IANA によって定義されています。リストは <http://www.iana.org/assignments/media-types/> で参照できます。CICS では、この属性の値を使用して、応答の Content-Type ヘッダーを作成します。

この属性にはデフォルト値がないため、値を指定する必要があります。MEDIATYPE 属性でテキスト・タイプ (text/ で始まるタイプや、+xml を含むタイプなど) を指定する場合は、CHARACTERSET 属性および HOSTCODEPAGE 属性も指定して、コード・ページ変換を実行できるようにしなければなりません。テキスト・メディア・タイプは、RFC 3023 によって示されています。RFC 3023 は、<http://www.ietf.org/rfc/rfc3023.txt> から入手可能で、RFC 2376 を置き換えます。

動的 (アプリケーションによって生成される) 応答の場合、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。URIMAP 定義が適用される URI のパス部分を 255 文字以内で指定します。パスの例としては、software/htp/cics/index.html を挙げることができます。指定可能な最短のパスは / (スラッシュ) であり、指定されたホスト名の URL 構造のルートを表します。

PATH 属性は大/小文字混合で指定します。大/小文字は、URIMAP 定義で保持されます。PATH 属性には、URI で許可されている文字のみを含める必要があります。具体的には、文字 < > # % “ { } | \ ^ [] ` および組み込みブランクは除外する必要があります (% は有効な 16 進数のエスケープ・シーケンスを採用する場合、すなわち、大文字または小文字の有効な 2 桁の 16 進数字が続く場合にのみ許可されます)。CICS 内で波形記号 (~) を指定することはできないため、波形記号は対応する 16 進数のエスケープ・シーケンス (%7E) に置き換える必要があります。CICS は、定義時に文字の使用を検証します。

HTTP サーバーおよび Web サービスとしての CICS に関連する URIMAP 定義では、複数のパスに URIMAP 定義を一致させる必要がある場合は、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使用できます。例えば、パス /software/htp/cics/* を指定した場合、URIMAP 定義は、アスタリスクの左のストリングで始まるパスを持つすべての要求に一致します。パス /* を指定した場合、URIMAP 定義は HOST 属性に指定されているホストに送信される任意の要求に一致します。複数の URIMAP 定義が HTTP 要求に一致する場合は、最も具体的な一致が採用されます。

query (照会) 部分が存在する場合、その特定の照会のみ URIMAP 定義を適用させる必要があるときは、パス部分の一部としてこれを含めることができます。疑問符 (?) はストリングの先頭を含めてください。照会ストリングには、URI で許可されている文字のみを使用する必要があります。照会ストリング自体にはワイルドカードとしてアスタリスクを含めることができませんが、ワイルドカードとしてアスタリスクを含むパスの後に続けて指定することはできます。URIMAP 定義に照会ストリングを含めなかった場合、HTTP 要求内の照会ストリングはマッチングの目的としては自動的に無視されます。

HTTP クライアントとしての CICS 用の URIMAP 定義の場合は、ワイルドカードとしてアスタリスクを使用できません。要求の完全なパスを指定する必要があります。WEB OPEN コマンドで URIMAP 定義が参照された場合、このパスが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。WEB SEND コマンドで URIMAP 定義が参照された場合、このパスがその WEB SEND コマンドで使用されます。ただし、その URIMAP 定義の HOST 属性は、接続の WEB SEND コマンドで指定されたホストと一致する必要がありますことに注意してください。

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 定義によって指定された URI を使用して CICS に対するインバウンド Web サービス要求を生成した場合、PIPELINE は、その Web サービスの PIPELINE リソース定義の 1 から 8 文字の名前を指定します。PIPELINE リソース定義は、クライアントからのサービス要求に対して作動するメッセージ・ハンドラーに関する情報を提供します。

PROGRAM(*name*)

この属性は、アプリケーションにより生成される応答が提供される **USAGE(SERVER)** で使用します。HTTP 応答を作成するユーザー・アプリケーション・プログラムの 1 から 8 文字の名前を指定します。HTTP サーバーとしての CICS では、アナライザーまたはコンバーター・プログラムが指定されているか、テンプレート名または HFS ファイルが指定されているか、あるいはリダイレクトが指定されている場合を除き、この属性は必須です。

REDIRECTTYPE(**{NONE|TEMPORARY|PERMANENT}**)

この属性は、**USAGE(SERVER)** および **USAGE(PIPELINE)** で使用します。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、LOCATION 属性によって指定される URL がリダイレクトに使用されます。

- NONE は、要求がリダイレクトされないことを意味します。LOCATION 属性によって URL が指定されていても無視されます。
- TEMPORARY は、要求が一時的にリダイレクトされることを意味します。LOCATION 属性によって指定される URL が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) です。
- PERMANENT は、要求が永続的にリダイレクトされることを意味します。LOCATION 属性によって指定される URL が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 (永続的に移動しました) です。

SET URIMAP コマンドを使用して、REDIRECTTYPE 属性および LOCATION 属性を変更することができます。

SCHEME({HTTP|HTTPS})

この属性は、すべての **USAGE** オプションで使われます。URIMAP 定義が適用される URI のスキーム部分を、HTTP (SSL を指定しない) または HTTPS (SSL を使用する) として指定します。URI のスキーム部分の後に続く区切り文字 `://` (コロンと 2 つのスラッシュ) を含めないでください。

注: HTTP スキームが指定された URIMAP は、HTTP スキームまたは HTTPS スキームを使用して行われた Web クライアント要求を受信します。HTTPS スキームが指定された URIMAP は、HTTPS スキームを使用して行われた Web クライアント要求のみを受信します。

STATUS({ENABLED|DISABLED})

この属性は、すべての **USAGE** オプションで使われます。URIMAP 定義を使用可能状態でインストールするか使用不可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です。

TCPIPSERVICE(*name*)

この属性は、**USAGE(SERVER)** および **USAGE(PIPELINE)** で使われます。PROTOCOL(HTTP) 付きで、この URIMAP 定義が関連するインバウンド・ポートが定義された TCPIPSERVICE リソース定義の 1 から 8 文字の名前を指定します。この属性を指定しなかった場合、URIMAP 定義は任意のインバウンド・ポートでの要求に適用されます。

スキームとして HTTPS が指定された URIMAP 定義と、Web クライアントが行う要求とが一致する場合、CICS は、その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確認します。ポートに SSL が指定されていない場合、要求は 403 (禁止) 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場合は、この検査を行うことによって、Web クライアントはセキュア・リソースへのアクセス時に非セキュア・ポートを使用できないことが保証されます。スキームとして HTTP が指定された URIMAP 定義に対しては検査が実行されないため、Web クライアントはこれらのリソースへのアクセス時に、非セキュア・ポートでもセキュア (SSL) ポートでも使用できます。

TEMPLATENAME(*name*)

この属性は、静的な応答が提供される **USAGE(SERVER)** で使われます。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する CICS 文書テンプレートの 1 から 48 文字の名前を指定します。これは、DOCTEMPLATE リソース定義を使用して定義する必要があり、このリソース定義の TEMPLATENAME 属性は、URIMAP 定義で使用する名前を指定します。CICS 文書テンプレートの形成方法については、「*CICS Application Programming Guide*」を参照してください。

注: リソース・レベル・セキュリティーは、静的応答として引き渡される項目には適用されません。この方法で引き渡された項目に対してユーザー ID に基づくアクセス制御を適用する必要がある場合は、代わりに、そのデータをアプリケーション生成の応答として引き渡す必要があります。

パス・マッチングを使用するには、CICS 文書テンプレートの名前の末尾と、PATH 属性で指定されているパスの末尾に、ワイルドカード文字としてアスタ

リスクを記述します。CICS は、各 HTTP 要求のパスのうち、ワイルドカード文字の適用範囲に相当する部分を取り込んで、これをテンプレート名の最後の部分と置き換えます。

例えば、PATH 属性を

```
findout/about/*
```

と指定し、TEMPLATENAME 属性を以下のように指定した URIMAP 定義を作成できます。

```
templates.facts.*
```

この URIMAP 定義は、以下の着信 HTTP 要求を処理するために使用されま
す。

```
http://www.example.com/findout/about/fish.html
```

CICS では、URIMAP 定義で指定されたテンプレート名にアスタリスクの代わ
りに fish.html を付加します。これにより、

```
templates.facts.fish.html
```

を使用して、静的応答が形成されます。

注: TEMPLATENAME 属性にアスタリスクを単独で指定すると、選択したテン
プレートの名前は、URL のうちで、PATH 属性のワイルドカード文字に対
応する部分と同じ名前になります。

TEMPLATENAME 属性を指定した場合、照会ストリングが URI に存在し、一
方 PATH 属性で使用されていないときは、CICS は自動的に照会ストリングの
内容を、指定された CICS 文書テンプレートにシンボル・リストとして渡しま
す。照会ストリングの内容を文書テンプレートで使用するには、文書テンプレ
ートに適切な変数を指定して、この変数が照会ストリングの内容に置換されるよ
うにする必要があります。

TRANSACTION(*name*)

この属性は、アプリケーションにより生成される応答が提供される

USAGE(SERVER)、および **USAGE(PIPELINE)** で使用します。HTTP 応答を
作成するユーザー・アプリケーションを実行するため、あるいはパイプラインを
開始するために使用される別名トランザクションの、1 から 4 文字の名前を指
定します。デフォルトの別名トランザクションは、USAGE(SERVER) の場合は
CWBA、USAGE(PIPELINE) の場合は CPHI です。セキュリティー、モニタ
ー、およびアカウントिंगの目的で、あるいはトランザクション・クラスの制
限のために、別の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
ザクションとして選択した名前に関係なく、常に同じプログラムが実行されな
ければならず、そのプログラムは USAGE 属性によって決定します。

USAGE(SERVER) の場合、プログラムは DFHWBA であり、URIMAP 定義の
PROGRAM 属性で指定されたアプリケーション・プログラム (またはアナライ
ザー・プログラムで指定されたアプリケーション・プログラム) にリンクしてい
ます。USAGE(PIPELINE) の場合は、プログラムは DFHPIDSH であり、
PIPELINE 属性で指定されたパイプライン (および指定されていれば
WEBSERVICE 属性で指定された Web サービス) を開始します。

URIMAP(*name*)

この URIMAP 定義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。属性は大/小文字混合で指定します。大/小文字は、URIMAP 定
義で保持されます。

許容文字:

A-Z 0-9 \$ @ #

USAGE({SERVER|CLIENT|PIPELINE})

この URIMAP 定義が、HTTP サーバーとしての CICS (SERVER) 用、HTTP
クライアントとしての CICS (CLIENT) 用、および Web サービス (PIPELINE)
用のいずれであるかを指定します。USAGE 属性によって、URIMAP 定義内の
その他の属性のうちのどれを使用できるかが決まります。

SERVER を指定すると、HTTP サーバーとしての CICS 用の URIMAP 定義が
作成されます。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ために使用されます。アプリ
ケーション生成の応答、または静的応答を提供できます。

CLIENT を指定すると、HTTP クライアントとしての CICS 用の URIMAP 定
義が作成されます。このタイプの URIMAP 定義は、CICS がサーバー上の
HTTP リソースに対する要求を行う場合に使用されます。これにより、ご使用の
アプリケーション・プログラムでの URI の識別を避けことができます。

PIPELINE を指定すると、Web サービス用の URIMAP 定義が作成されます。
このタイプの URIMAP 定義は、インバウンド Web サービス要求 (すなわち、
クライアントが CICS の Web サービスを呼び出すための要求) に使用されま
す。着信要求の URI は、メッセージで実行する処理を指定する WEBSERVICE
リソースおよび PIPELINE リソースに関連付けられます。

USERID(*id*)

この属性は、アプリケーションにより生成される応答が提供される

USAGE(SERVER)、および **USAGE(PIPELINE)** で使用します。別名トランザ
クションを接続する際に使用する 1 から 8 文字のユーザー ID を指定しま
す。URIMAP 定義に指定したユーザー ID は、接続用の TCPIP SERVICE 定義
の AUTHENTICATE 属性で指定された認証手続きを使用して、クライアントか
ら直接取得される任意のユーザー ID によってオーバーライドできます。

ANALYZER(YES) が指定された場合、URIMAP 定義またはクライアントから取
得したユーザー ID は、アナライザー・プログラムによって変更可能です。ア
ナライザー・プログラムがユーザー ID を設定することもできます。前述した
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、デフォルトのユー
ザー ID が CICS のデフォルト・ユーザーになります。

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 定義によって指定された URI を使用して CICS に対するインバウン
ド Web サービス要求を生成した場合、WEBSERVICE は、使用される
WEBSERVICE リソース定義の 1 から 8 文字の名前を指定します。このリソ
ース定義は、Web サービス設定で配置された CICS アプリケーション・プログラ

ムのためのランタイム環境の特徴を定義します。この環境では、アプリケーション・データ構造と SOAP メッセージとの間のマッピングが、CICS 提供のツールを使用して生成されます。

アプリケーション・プログラミング・インターフェースの変更点 (汎用 CICS Web サポート拡張)

新しいコマンドと変更されたコマンド

HTTP クライアントとしての CICS に関して、新しい EXEC CICS WEB コマンドが追加されました。また、いくつかのコマンドは、HTTP クライアントとしての CICS に関しての変更されましたが、HTTP サーバーとしての CICS に関しては変更されていません。新規および変更されたクライアント・コマンドについては、78 ページの『アプリケーション・プログラミング・インターフェースの変更点 (HTTP クライアント要求)』で説明します。

EXEC CICS WEB SEND コマンドおよび EXEC CICS WEB RECEIVE コマンドは、HTTP サーバーとしての CICS で使用される場合に HTTP/1.1 をサポートするように拡張されました。これについては、116 ページの『アプリケーション・プログラミング・インターフェース (HTTP/1.1 サポート) の変更点』で説明します。

以下の新しい EXEC CICS コマンドは、HTTP サーバーとしての CICS および HTTP クライアントとしての CICS の両方で使用できます。

- EXEC CICS WEB PARSE URL
- EXEC CICS CONVERTTIME

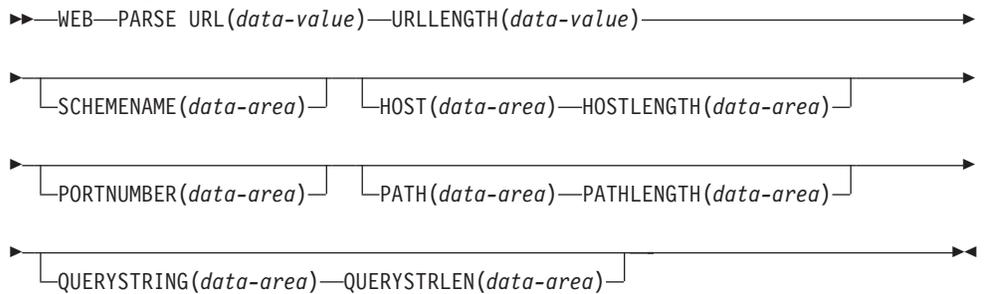
以下の EXEC CICS コマンドは、HTTP サーバーとしての CICS および HTTP クライアントとしての CICS の両方で変更されました。

- EXEC CICS WEB EXTRACT
- EXEC CICS FORMATTIME

WEB PARSE URL

URL スtringをその成分に分解します。

WEB PARSE URL



条件: INVREQ、LENGERR

このコマンドはスレッド・セーフです。

説明

WEB PARSE URL を使用すると、URL スtringをその部分、つまりスキーム、ホスト、ポート、パス、および照会Stringに分解できます。このプロセスを使用して、URL の構造を検査し、コンポーネントに分離することができます。戻された情報を WEB OPEN コマンド内で使用して、URL で指定されたホストとのクライアント接続を開始することができます。

URL に含まれるすべてのエスケープ・シーケンスの妥当性が検査されます。エスケープ・シーケンスは、パーセント文字 (%) とその後続く 2 つの 16 進文字で構成されます。有効な 16 進文字の桁は 0 から 9 で、文字は A から F です。

WEB PARSE URL コマンドに入力されたStringが URL で使用できる正しい区切り文字で区切られている場合、コマンドにより、インターネット上に存在しないホスト名を指すホスト名や URL に許可されていない文字などの無効内容が検出されません。

オプション

HOST(data-area)

URL のホスト部分を戻します。英数字のホスト名または数値の IP アドレスのいずれかが戻されます。ポート番号が URL 内に明示的に指定してあった場合、この部分は PORTNUMBER オプションとして別個に戻されます。

WEB OPEN コマンドには、ホスト名として IPv4 アドレスを使用できますが、IPv6 アドレスはサポートされていません。IPv6 アドレスは予想される構造に適合していないため、WEB PARSE URL コマンドでは無効として拒否されます。

HOSTLENGTH(data-area)

HOST オプションで指定されたバッファの長さをフルワード・バイナリー変数で指定します。また、アプリケーションに戻されたデータ (ホスト名) の実際の長さに設定されます。このデータ域に指定する適切なサイズとして 116 文字が推奨されています。データがバッファ長を超えると、LENGERR 条件が発生し、データは切り捨てられます。

PATH(*data-area*)

URL のパス部分に戻します。

PATHLENGTH(*data-area*)

PATH オプションで指定されたバッファの長さをフルワード・バイナリー変数で指定します。また、アプリケーションに戻されたデータ (URL のパス部分) の実際の長さに設定されます。このデータ域に指定する適切なサイズとして 256 文字が推奨されています。データがバッファ長を超えると、LENGERR 条件が発生し、データは切り捨てられます。

PORTNUMBER(*data-area*)

URL に指定されている (または、その URL に対して適切な) ポート番号を (フルワード・バイナリーのデータ域で) 戻します。ポート番号は、ホスト名に続いて URL に明示的に指定される場合があります。ただし、サービスのウェルノウン・ポート番号は、通常、URL から省略されます。URL にポート番号が存在しない場合、WEB PARSE URL コマンドは、スキームに基づいてポート番号を識別し、戻します。HTTP の場合、ウェルノウン・ポート番号は 80 であり、HTTPS の場合、ウェルノウン・ポート番号は 443 です。スキームのデフォルト以外のポート番号が戻る場合は、ポート番号を明示的に指定して、URL にアクセスできるようにする必要があります (例えば、この情報を WEB OPEN コマンドに使用している場合など)。

QUERYSTRING(*data-area*)

URL から照会stringに戻します。照会stringとは、パスの末尾を区切る疑問符 (?) の後にエンコードされている 1 つ以上の値のことです。照会stringは、そのエスケープ形式で戻されます。

QUERYSTRLEN(*data-area*)

QUERYSTRING オプションで指定されたバッファの長さをフルワード・バイナリー変数で指定します。また、アプリケーションに戻されたデータ (照会string) の実際の長さに設定されます。このデータ域に指定する適切なサイズとして 256 文字が推奨されています。データがバッファ長を超えると、LENGERR 条件が発生し、データは切り捨てられます。

SCHEMENAME(*data-area*)

URL のスキーム部分を 16 文字のデータ域として戻します。CICS では、HTTP および HTTPS スキーム (SSL を使用する HTTP プロトコル、および SSL を使用しない HTTP プロトコル) のみがサポートされるため、WEB OPEN コマンドで使用できるのはこれらのスキームのみです。

URL(*data-value*)

完全な URL stringを指定します。

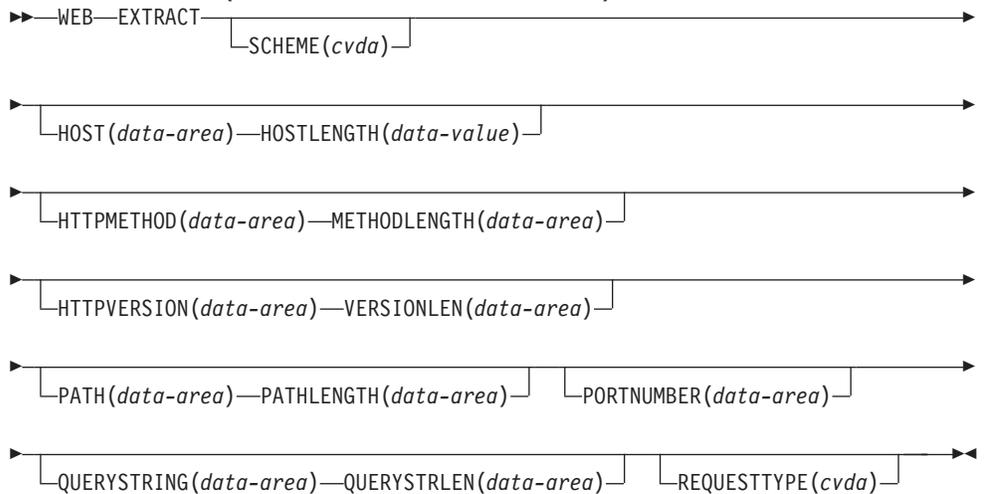
URLLENGTH(*data-value*)

URL stringを格納するバッファの長さをフルワード・バイナリー値で指定します。

WEB EXTRACT

HTTP サーバーとしての CICS に対して行われた HTTP 要求に関する情報、またはインターネット・サーバーと HTTP クライアントとしての CICS 間の接続に関する情報を取得します。

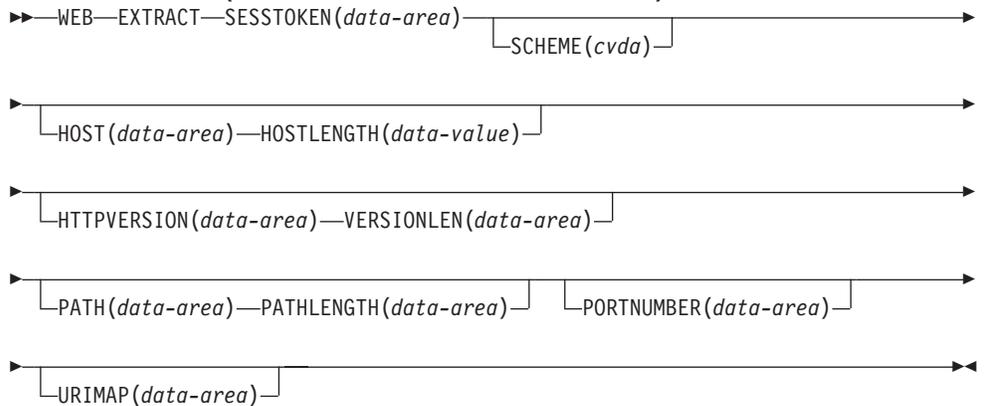
WEB EXTRACT (HTTP サーバーとしての CICS)



条件: ILLOGIC、INVREQ、LENGERR、NOTOPEN

このコマンドはスレッド・セーフです。

WEB EXTRACT (HTTP クライアントとしての CICS)



条件: ILLOGIC、INVREQ、LENGERR、NOTOPEN

このコマンドはスレッド・セーフです。

オプション

HOST(*data-area*)

HTTP サーバーとしての CICS では、このオプションは、要求の Host ヘッダー・フィールドまたは要求行 (要求に絶対 URI が使用されていた場合) のいずれかに指定された URL のホスト部分を格納するバッファを指定します。ポート番号は、PORTNUMBER オプションを使用して個別に指定します。

HTTP クライアントとしての CICS (SESSTOKEN オプションが指定されている場合) では、このオプションは SESSTOKEN オプションで指定された接続のサーバーのホスト名を格納するバッファを指定します。ポート番号は、PORTNUMBER オプションを使用して個別に指定します。

HOSTLENGTH(*data-area*)

HOST オプションで指定されたバッファの長さをフルワード・バイナリー変数で指定します。また、アプリケーションに戻されたデータの実際の長さの設定されます。このデータ域に指定する適切なサイズは、116 文字です。データがバッファ長を超えると、LENGERR 条件が発生し、データは切り捨てられます。

HTTPMETHOD(*data-area*)

HTTP サーバーとしての CICS では、このオプションはメッセージの要求行の HTTP メソッドのストリングを格納するバッファを指定します。

このオプションは、HTTP クライアントとしての CICS には適用されません。

HTTPVERSION(*data-area*)

HTTP サーバーとしての CICS では、このオプションは、要求に示されている Web クライアントの HTTP バージョンを格納するバッファを指定します。

HTTP クライアントとしての CICS (SESSTOKEN オプションが指定されている場合) では、このオプションは SESSTOKEN オプションで指定された接続のサーバーの HTTP バージョンを格納するバッファを指定します。

「1.1」は、HTTP/1.1 を示し、「1.0」は HTTP/1.0 を示します。

注: CICS では、HTTP/1.0 レベル以下のサーバーまたは Web クライアントに対して特別な対応をしていません。CICS は、HTTP/1.0 レベルであるものとして動作し、HTTP バージョンとして「1.0」を戻します。

アプリケーション・プログラムが HTTP/1.0 レベルの Web クライアントまたはサーバーに適さない可能性のある HTTP ヘッダーを書き込む場合、または Web クライアントまたはサーバーに、チャンク化された情報 (HTTP/1.0 レベルのクライアントまたはサーバーでは受信できない) を送信する予定の場合は、アプリケーション・プログラムが HTTP バージョンの情報を検査する必要があります。

METHODLENGTH(*data-area*)

HTTPMETHOD オプションで指定されたバッファの長さをフルワード・バイナリー変数で指定します。また、アプリケーションに戻されたデータの実際の長さの設定されます。データがバッファ長を超えると、LENGERR 条件が発生し、データは切り捨てられます。

PATH(*data-area*)

HTTP サーバーとしての CICS では、このオプションは、メッセージの要求行に指定されたパスを格納するバッファを指定します。

HTTP クライアントとしての CICS (SESSTOKEN オプションが指定されている場合) では、このオプションは、この接続を使用して作成された要求に適用されるデフォルトのパスを格納するバッファを指定します。接続に対する WEB

OPEN コマンドに URIMAP 定義が指定されている場合、デフォルトのパスは URIMAP 定義に指定されたパスになります。それ以外の場合、デフォルトのパスは 1 つのスラッシュになります。

PATHLENGTH(*data-area*)

PATH オプションで指定されたバッファの長さをフルワード・バイナリー変数で指定します。また、アプリケーションに戻されたデータの実際の長さに設定されます。このデータ域に指定する適切なサイズは、256 文字です。データがバッファ長を超えると、LENGERR 条件が発生し、データは切り捨てられます。

PORTNUMBER(*data-area*)

HTTP サーバーとしての CICS では、このオプションを指定すると、メッセージの要求行に指定されたポート番号を格納するデータ域が戻されます。

HTTP クライアントとしての CICS (SESSTOKEN オプションが指定されている場合) では、このオプションを指定すると、SESSTOKEN オプションで指定された接続でサーバーにアクセスするためのポート番号を含むデータが戻されます。

データ域には、フルワード・バイナリー値が戻されます。

サービスのウェルノウン・ポート番号は、通常、URL から省略されます。URL にポート番号が存在しない場合、WEB EXTRACT コマンドは、スキームに基づいてポート番号を識別し、戻します。HTTP の場合、ウェルノウン・ポート番号は 80 であり、HTTPS の場合、ウェルノウン・ポート番号は 443 です。スキームのデフォルト以外のポート番号が戻る場合は、ポート番号を明示的に指定して、URL にアクセスできるようにする必要があります (例えば、この情報を WEB OPEN コマンドに使用している場合など)。

QUERYSTRING(*data-area*)

HTTP サーバーとしての CICS では、このオプションは、メッセージの要求行の照会ストリングを格納するバッファを指定します。照会ストリングとは、パスの末尾を区切る疑問符 (?) の後にエンコードされている 1 つ以上の値のことです。照会ストリングは、そのエスケープ形式で戻されます。

このオプションは、HTTP クライアントとしての CICS には適用されません。

QUERYSTRLEN(*data-area*)

QUERY オプションで指定されたバッファの長さを、フルワード・バイナリー変数で指定します。また、アプリケーションに戻されたデータ (照会ストリング) の実際の長さに設定されます。このデータ域に指定する適切なサイズは、256 文字です。データがバッファ長を超えると、LENGERR 条件が発生し、データは切り捨てられます。

REQUESTTYPE(*cvda*)

HTTP サーバーとしての CICS では、このオプションは受信される要求のタイプを指定します。このオプションは、HTTP クライアントとしての CICS には適用されません。CVDA 値は、以下のとおりです。

HTTPYES

HTTP 要求を示します。

HTTPNO

HTTP 以外の要求を示します。

SCHEME(*cvda*)

HTTP サーバーとしての CICS、および HTTP クライアントとしての CICS (SESSTOKEN オプションが指定されている場合) のどちらの場合も、このオプションを指定すると、CICS と Web クライアントまたはサーバー間の接続に使用されたスキームが戻されます。CVDA 値は、以下のとおりです。

HTTP SSL を備えていない HTTP プロトコルのことです。

HTTPS

SSL を備えた HTTP である HTTPS プロトコルのことです。

SESSTOKEN(*data-value*)

HTTP クライアントとしての CICS では、このオプションは必須です。セッション・トークンが指定されている場合、CICS とサーバー間の接続を一意に識別する 8 バイトのバイナリー値です。この値は、WEB OPEN コマンドによって、HTTP クライアントとしての CICS に戻されます。77 ページの『セッション・トークン』には、セッション・トークンの使用法が説明されています。WEB EXTRACT コマンドでは、指定された接続に関する情報が戻されます。

このオプションは、HTTP サーバーとしての CICS には適用されません。

URIMAP(*data-value*)

HTTP クライアントとしての CICS (SESSTOKEN オプションが指定されている場合) では、このオプションを指定すると、SESSTOKEN オプションで指定された接続を開始する WEB OPEN コマンドに指定されている URIMAP 定義の 8 文字の名前が戻されます。INQUIRE URIMAP コマンドを使用すると、この URIMAP 定義の属性に関する情報を検索できます。

このオプションは、HTTP サーバーとしての CICS には適用されません。

VERSIONLEN(*data-area*)

HTTPVERSION オプションで指定されたバッファの長さを、フルワード・バイナリー変数で指定します。また、アプリケーションに戻されたデータの実際の長さに設定されます。

CONVERTTIME

設計済み日時スタンプ形式ストリングを ABSTIME 形式に変換します。

CONVERTTIME

►►—CONVERTTIME—DATESTRING(*data-area*)—ABSTIME(*data-area*)—►►

条件: INVREQ、LENGERR

このコマンドはスレッド・セーフです。

説明

CONVERTTIME は、インターネット上で一般的に使用される 3 つの異なる日時スタンプ形式を分析して、ABSTIME (絶対日時) 形式に変換します。

ABSTIME 形式では、1900 年 1 月 1 日 00:00 以降の時刻がパック 10 進数 (最も近い 100 分の 1 秒の桁の値に丸められたミリ秒単位の値) で表現されます。この形式は、FORMATTIME コマンドを使用すれば他の形式に変更できます。

CONVERTTIME コマンドによって認識される設計済み日時スタンプ・ストリング形式は、以下のとおりです。

RFC 1123 形式

RFC 1123 に規定された HTTP プロトコルの日時スタンプの優先標準形式。この形式での日時スタンプの例は、「Tue, 01 Apr 2003 10:01:02 GMT」です。

RFC 850 形式

インターネットでの旧式の日時スタンプ形式。この形式での日時スタンプの例は、「Tuesday, 01-Apr-03 Apr 10:01:02 GMT」です。

重要: この形式では、年を 2 桁のみで表しているため、CICS では、年が 1970 年から 2069 年の範囲と見なされます。上記の例では、CICS は文書の日付が 2003 年 4 月 1 日であったものと見なします。日時スタンプが「Thursday, 13-Feb-98 15:30:00 GMT」であるとすると、CICS は文書の日付が 1998 年 2 月 13 日であったものと見なします。日時スタンプをこの形式で受信することを考える場合は、アプリケーションをコーディングする際に、この点に注意してください。

ASCTime 形式

C ASCTime 関数からの日時スタンプ形式出力。この形式での日時スタンプの例は、「Tue, 01 Apr 1 10:01:02 2003」です。

オプション

DATESTRING(*data-area*)

設計済み日時スタンプ・ストリングを格納するための 64 文字のデータ域を指定します。ストリングは、このコマンドによって認識されるいずれかの形式で入力できます。どの形式を使用するかを指定する必要もありません。

ABSTIME(*data-area*)

ABSTIME 形式に変換済みの日時スタンプを受け取るデータ域を指定します。このデータ域の形式については、ASKTIME コマンドの説明を参照してください。日時スタンプが認識される形式でなかった場合、ABSTIME 形式の値は戻りません。

EXEC CICS WEB コマンドのオプションの変更点

インターネットで使用するフォーマットの日時スタンプを戻すように、FORMATTIME コマンドに DATESTRING および STRINGFORMAT オプションが追加されました。

FORMATTIME

EXEC CICS FORMATTIME コマンドに DATESTRING および STRINGFORMAT オプションが追加されました。これらのオプションを使用すると、ABSTIME フォーマットの日時スタンプを、設計済み日時ストリングに変換することができます。現在、STRINGFORMAT オプションで提供される唯一のフォーマットは、インターネットでの使用に適した RFC 1123 標準に準拠したフォーマットです。

DATESTRING(*data-area*)

STRINGFORMAT オプションで指定されたフォーマットの設計済み日時スタンプ・ストリングを CICS が戻す 64 文字のユーザー・フィールドを指定します。STRINGFORMAT を指定しなかった場合、提供されるデフォルト・フォーマットは RFC 1123 フォーマット (RFC1123) です。

STRINGFORMAT(*cvda*)

DATESTRING に戻される設計済み日時スタンプ・ストリングのフォーマットを指定します。現在使用可能な唯一の CVDA 値は、以下のとおりです。

RFC1123

インターネットでの使用に適した RFC 1123 フォーマットを指定します。この日時スタンプ・ストリングには、GMT で曜日、日付、および 24 時間クロック時刻が入ります。例えば、「Tue, 01 Apr 2003 10:01:02 GMT」のようになります。

システム・プログラミング・インターフェースの変更点

INQUIRE DOCTEMPLATE および CREATE DOCTEMPLATE コマンドに HFSFILE オプションが追加されました。

HFSFILE(*filename*)

テンプレートがある z/OS UNIX システム・サービス HFS ファイルの、完全修飾 (絶対) 名または相対名を指定します。255 文字以内の長さにする必要があります。

URIMAP 定義を管理するために、以下のコマンドが提供されています。

- CREATE URIMAP
- DISCARD URIMAP
- INQUIRE URIMAP
- SET URIMAP

CREATE URIMAP コマンド

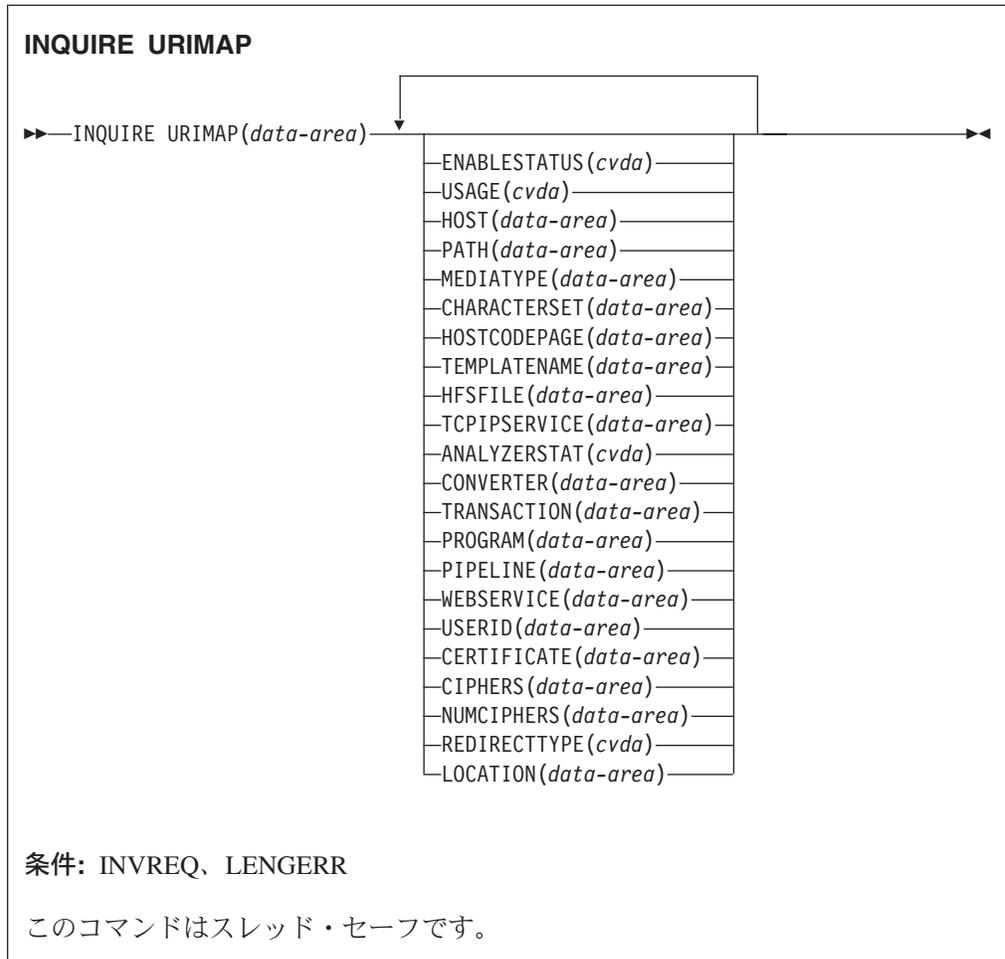
CREATE URIMAP コマンドは、CICS 領域に URIMAP を動的に作成するために使用します。このコマンドで指定できる属性については、135 ページの『URIMAP 定義属性』で説明しています。

DISCARD URIMAP コマンド

DISCARD URIMAP は、CICS 領域から URIMAP を除去するために使用します。URIMAP を廃棄する場合は、廃棄の前にこれを使用不可に設定する必要があります。

INQUIRE URIMAP

ローカル・システム内の URIMAP リソースに関する情報を取得します。



説明

INQUIRE URIMAP コマンドを使用すると、特定の URIMAP 定義に関する情報を取得できます。URIMAP 定義の USAGE 属性は、その URIMAP 定義で指定されるその他の属性を決定し、場合によっては、特定の属性の意味を決定します。

ブラウズ

INQUIRE URIMAP コマンドでブラウズ・オプション (START、NEXT、および END) を使用することにより、領域内にインストールされているすべての URIMAP 定義をブラウズできます。

オプション

URIMAP(*data-value*)

URIMAP 定義の 8 文字の名前を指定します。

ANALYZERSTAT(*cvda*)

TCPIPSERVICE に関連付けられているアナライザー・プログラムを実行するかどうかを示す CVDA 値を戻します。CVDA 値は、以下のとおりです。

ANALYZER

アナライザー・プログラムを実行します。

NOANALYZER

アナライザー・プログラムを実行しません。

この属性は、USAGE(SERVER) で使用します。その他の USAGE タイプの場合は、強制的に NO になります。

CERTIFICATE(*data-area*)

HTTP クライアントとしての CICS による HTTP 要求のための SSL クライアント証明書として使用される証明書のラベルを含む 32 文字のデータ域を戻します。この属性は、USAGE(CLIENT) で使用します。

CHARACTERSET(*data-area*)

静的応答に使用する文字セットの名前を含む 40 文字のデータ域を戻します。この属性は、USAGE(SERVER) で使用します。

CIPHERS(*data-area*)

URIMAP 定義に指定された暗号スイートのリストを含む 56 文字のデータ域を戻します。この暗号スイートのリストは、SSL 接続のネゴシエーションで使用されます。この属性は、USAGE(CLIENT) で使用します。

CONVERTER(*data-area*)

HTTP サーバーとしての CICS に対する変換またはその他の処理を実行するコンバーター・プログラムの 8 文字の名前を戻します。この属性は、USAGE(SERVER) で使用します。

ENABLESTATUS(*cvda*)

この URIMAP 定義の状況を示す CVDA 値を戻します。CVDA 値は、以下のとおりです。

ENABLED

URIMAP 定義は使用可能です。

DISABLED

URIMAP 定義は使用不可です。この状況を持つ URIMAP 定義は廃棄できます。

DISABLEDHOST

URIMAP 定義を使用している仮想ホストが使用不可になったため、その定義は使用不可です。SET HOST コマンドを使用して、仮想ホストを構成するすべての URIMAP 定義を再び使用可能にすることができます。この状況を持つ URIMAP 定義は廃棄できません。

HFSFILE(*data-area*)

静的応答を形成する z/OS UNIX システム・サービス HFS ファイルの完全修飾(絶対)名または相対名を含む 255 文字のデータ域を戻します。この属性は、USAGE(SERVER) で使用します。

HOST(*data-area*)

URIMAP 定義が適用される URI のホスト部分 (例えば、www.example.com) を含む 116 文字のデータ域を戻します。この属性は、あらゆるタイプの USAGE で使用します。

HOSTCODEPAGE(*data-area*)

静的応答を形成するテキスト文書のエンコードに使用する 1 文字から 10 文字

の IBM コード・ページ (EBCDIC) の名前を含む 10 文字のデータ域を戻します。この属性は、USAGE(SERVER) で使用します。

LOCATION(*data-area*)

Web クライアントからの一致する HTTP 要求がリダイレクトされる宛先の URL を含む 255 文字の領域を戻します。REDIRECTTYPE オプションによって指定された設定により、リダイレクトが活動化されます。この属性は、USAGE(SERVER) または USAGE(PIPELINE) で使用します。

MEDIATYPE(*data-area*)

静的応答のデータ・コンテンツの説明を含む 56 文字のデータ域を戻します。この属性は、USAGE(SERVER) で使用します。

NUMCIPHERS(*data-area*)

CIPHERS リスト内の暗号コードの数を含むハーフワード・バイナリー値を戻します。暗号は、SSL ハンドシェイクの一部としての暗号化レベルのネゴシエーションに使用されます。この属性は、USAGE(CLIENT) で使用します。

PATH(*data-area*)

URIMAP 定義が適用される URL のパス部分 (例えば、software/http/cics/index.html) を含む 255文字のデータ域を戻します。この属性は、あらゆるタイプの USAGE で使用します。

PIPELINE(*data-area*)

Web サービスの PIPELINE リソース定義の 8 文字の名前を戻します。PIPELINE リソース定義は、クライアントからのサービス要求に対して作動するメッセージ・ハンドラーに関する情報を提供します。この属性は、USAGE(PIPELINE) で使用します。

PROGRAM(*data-area*)

アプリケーションにより生成される HTTP 要求への応答を作成するアプリケーション・プログラムの 8 文字の名前を戻します。この属性は、USAGE(SERVER) で使用します。

SCHEME(*cvda*)

URI のスキーム部分を示す CVDA 値を戻します。CVDA 値は、以下のとおりです。

HTTP SSL を備えていない HTTP

HTTPS

SSL を使用した HTTP

この属性は、あらゆるタイプの USAGE で使用します。

TCPIPSERVICE(*data-area*)

この URIMAP 定義に関連するインバウンド・ポートを指定する TCPIPSERVICE 定義の 1 から 8 文字の名前を戻します。この属性を指定しなかった場合、URIMAP 定義は任意のインバウンド・ポートでの要求に適用されます。この属性は、USAGE(SERVER) または USAGE(PIPELINE) で使用します。

TEMPLATENAME(*data-area*)

静的応答を形成するために使用される CICS 文書テンプレートの名前を含む 48 文字のデータ域を戻します。この属性は、USAGE(SERVER) で使用します。

TRANSACTION(*data-area*)

HTTP 要求への応答を作成するユーザー・アプリケーションを実行する別名トランザクションの 4 文字の名前を戻します。この属性は、USAGE(SERVER) または USAGE(PIPELINE) で使用します。

REDIRECTTYPE(*cvda*)

この URIMAP 定義に一致する要求のリダイレクトのタイプを示す CVDA 値を戻します。リダイレクトの URL は、LOCATION オプションで指定されます。この属性は、USAGE(SERVER) または USAGE(PIPELINE) で使用します。CVDA 値は、以下のとおりです。

NONE 要求はリダイレクトされません。LOCATION オプションによって URL が指定されていても無視されます。

TEMPORARY

要求は一時的にリダイレクトされます。応答に使用される状況コードは 302 (検出) です。

PERMANENT

要求は永続的にリダイレクトされます。応答に使用される状況コードは 301 (永続的に移動しました) です。

USAGE(*cvda*)

この URIMAP 定義の目的を示す CVDA 値を戻します。CVDA 値は、以下のとおりです。

SERVER

HTTP サーバーとしての CICS のための URIMAP 定義。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 URL を CICS リソースにマップするために使用されます。アプリケーション生成の応答、または静的応答を提供できます。

CLIENT

HTTP クライアントとしての CICS のための URIMAP 定義。このタイプの URIMAP 定義は、CICS がサーバー上の HTTP リソースに対するクライアント要求を行う場合に使用されます。

PIPELINE

Web サービスの URIMAP 定義。このタイプの URIMAP 定義は、クライアントが CICS の Web サービスを呼び出すための要求で実行される処理を指定するために使用されます。

USERID(*data-area*)

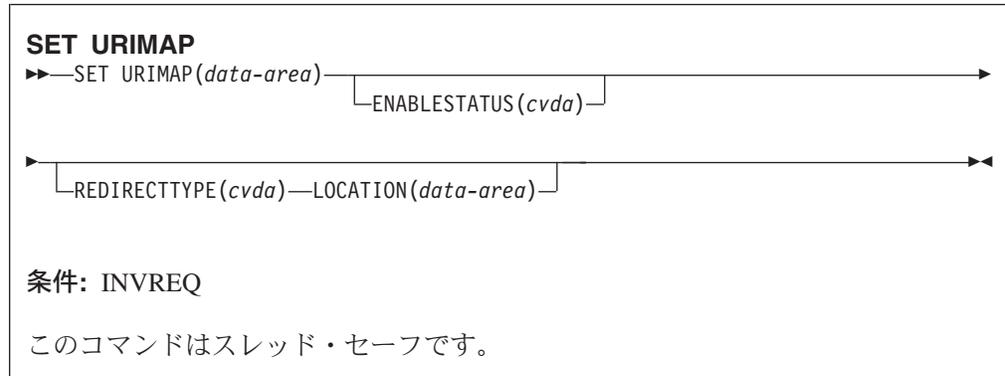
別名トランザクションを接続する際に使用する 8 文字のユーザー ID を戻します。この属性は、USAGE(SERVER) または USAGE(PIPELINE) で使用します。

WEBSERVICE(*data-area*)

WEBSERVICE リソース定義の 8 文字の名前を戻します。このリソース定義は、Web サービス設定で配置される CICS アプリケーション・プログラムのためのランタイム環境の特徴を定義します。この属性は、USAGE(PIPELINE) で使用します。

SET URIMAP

URIMAP 定義を使用可能または使用不可に設定し、URIMAP 定義のリダイレクトを適用または除去します。



説明

SET URIMAP コマンドを使用すると、以下のことを行うことができます。

- URIMAP 定義を使用可能または使用不可に設定する。
- 一致する HTTP 要求のリダイレクトを設定し、要求がリダイレクトされる宛先の URL を指定する。このコマンドを使用して、HTTP 要求に対して通常応答しているアプリケーションが使用不可な場合などに、既存の URIMAP 定義に対してリダイレクトを適用することができます。このコマンドを使用して、URIMAP 定義からリダイレクトを除去することもできます。

オプション

ENABLESTATUS(*cvda*)

URIMAP 定義を、使用可能状況または使用不可状況に設定します。CVDA 値は、以下のとおりです。

ENABLED

アプリケーションは、URIMAP 定義にアクセスできます。

DISABLED

アプリケーションは、URIMAP 定義にアクセスできません。URIMAP 定義は、再インストールまたは廃棄する前に、使用不可にする必要があります。

LOCATION(*data-area*)

Web クライアントからの一致する HTTP 要求をリダイレクト可能な宛先の URL を、255 文字以内で指定します。これは、スキーム部分、ホスト部分、パス部分、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 では URL が有効であるかは検査されないため、宛先が存在すること、およびその URL が正しく指定されていることをユーザーが確認する必要があります。

リダイレクトのタイプを指定するために、REDIRECTTYPE オプションが使用されます。一時リダイレクトまたは永続リダイレクトが指定された場合、

LOCATION 属性内の URL がリダイレクトに使用されます。NONE が指定された場合、LOCATION オプション内の URL は無視されます。

REDIRECTTYPE(*cvda*)

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。リダイレクトの URL は、LOCATION オプションで指定されます。CVDA 値は、以下のとおりです。

NONE 要求はリダイレクトされません。LOCATION オプションによって URL が指定されていても無視されます。

TEMPORARY

要求は一時的にリダイレクトされます。応答に使用される HTTP 状況コードは 302 (検出) です。

PERMANENT

要求は永続的にリダイレクトされます。応答に使用される HTTP 状況コードは 301 (永続的に移動しました) です。

CEMT の変更点

以下の新規コマンドが CEMT トランザクションに追加されました。

- INQUIRE URIMAP
- SET URIMAP
- DISCARD URIMAP
- INQUIRE HOST
- SET HOST

以下のコマンドに対して変更が行われました。

- INQUIRE TCPIPSERVICE
- SET TCPIPSERVICE
- INQUIRE DOCTEMPLATE

「*CICS Supplied Transactions*」に、これらの新規および変更コマンドについての情報が記載されています。

CICS 提供トランザクションの変更点

新しい CICS 提供のトランザクション CWXU

CICS Transaction Server for z/OS バージョン 3 リリース 1 では、HTTP 要求の処理と非 HTTP 要求の処理は分かれたままです。これにより、CICS は HTTP 要求や応答に関して基本的な受け入れ検査を実行できること (107 ページの『第 4 章 HTTP/1.1 への CICS Web サポートのアップグレード』で説明します)、および非 HTTP 要求はこれらの検査の対象にはならないことが保証されます。非 HTTP 要求の処理は、ユーザー定義 (USER) プロトコルのもとで実行しなければならなくなりました。このプロトコルは、要求を受信するポートの TCPIPSERVICE 定義で指定されます。

CICS 提供の新規トランザクション CWXU (CICS Web ユーザー定義プロトコル接続トランザクション) は、プロトコルが USER として定義された場合のデフォルト

です。CW XU は CICS プログラム DFHWBXN を実行します。DFHCURDI サンプルには、CW XU のサンプル定義が含まれています。DFHWBXN を実行する代替トランザクションが使用される場合もあります。ただし、TCPIPSERVICE リソース定義でプロトコルに対して定義されるその他のデフォルト・トランザクションは除きます。

CW XU は、RACF カテゴリー 1 トランザクションです。

CW XN の変更点

CICS 提供のトランザクション CW XN (Web 接続トランザクション) によって実行される処理に対していくつかの変更点があります。主な変更点は、以下のとおりです。

- HTTP 要求に関して一致する URIMAP 定義が見つかった場合、CW XN は、URIMAP 定義によって指示された場合にのみアナライザー・プログラムを呼び出すようになりました。
- 要求の HTTP バージョンが HTTP/1.1 である場合、CW XN は、要求の基本的な受け入れ検査を実行することにより、HTTP サーバーの責任の一部を担います。これらの検査を受けて CW XN は、ユーザー作成のアプリケーション・プログラムを必要とせずに、要求に対する応答を戻すための処理を行います。
- CW XN は、Web クライアントから受信したチャンク・メッセージおよびパイプライン・メッセージを前処理するので、ユーザー作成のアプリケーションはこの処理を実行する必要はありません。
 - チャンク・メッセージでは、単一のメッセージが分割されて、より小さなメッセージ (チャンク) のシリーズとして送信されます。CW XN はメッセージのチャンクを受信して組み立てることにより、単一の HTTP 要求を作成します。CW XN はメッセージが完全であることを検査してから、それをユーザー・アプリケーションに渡します。これによりユーザー・アプリケーションは、この要求を他の HTTP 要求と同様に処理できます。
 - パイプライン・メッセージでは、複数のメッセージが順々に送信され、そのときに送信側は送信された各メッセージの応答を待ちません。サーバーは、これらのメッセージに対して、受信した順序で応答する必要があります。この処理を確実に行うため、CW XN はパイプライン要求を保持し、それらを一度に 1 つずつユーザー・アプリケーションに解放します。ユーザー・アプリケーションは、最初の要求に対する応答を送信してから、次の要求を CW XN から受信する必要があります。
- 持続接続は、デフォルトの動作になりました。接続がクローズされるのは、Web クライアントがクローズを要求した場合、タイムアウト期間に達した場合、あるいは Web クライアントが Keep-Alive ヘッダーを送信しない HTTP/1.0 クライアントである場合のみです。
- CICS Transaction Server for z/OS バージョン 3 リリース 1 の前は、Web クライアントと CICS が持続接続されている場合、CW XN トランザクションはその持続接続の間はシステムに残っていました。CW XN トランザクションは、Web クライアントからの各要求が別名トランザクション (CWBA または別のトランザクション) に渡されるか、静的応答が引き渡されると、終了するようになりました。ソケット・リスナー・タスクは、ソケットをモニターし、持続接続の各要求ごとに CW XN の新規インスタンスを開始します。このような動作 (非同期受信

として知られる) をすることにより、最大タスク仕様 (MXT) に到達した状態でのデッドロックの可能性は回避されます。また、CICS と Web クライアントの間の同時接続の最大数は、MXT 値によって制限されなくなり、理論的には 64000 まで可能になりました。システム・アクティビティーの観点からいえば、CICS Transaction Server for z/OS バージョン 3 リリース 1 以前で持続接続を使用していた場合は、トランザクション比率が増加し、同時タスク数が減少したことがわかります。

CICS Web サポート・トランザクション (CWXXN、CWXXU、CWBA、または他の別名トランザクション) の優先順位

CWXXN または CWXXU トランザクションの優先順位を、アプリケーション生成の応答 (CWBA など) に使用される別名トランザクションの優先順位よりも高く設定した場合は、受信したけれども未処理である要求が蓄積して、ストレージ不足の状態になる可能性があります。

CWXXN または CWXXU トランザクションのデフォルト優先順位は 1 に設定され、CICS 提供の CWBA トランザクションのデフォルト優先順位も 1 に設定されています。これらの優先順位は、作業負荷に応じて調整できます。アプリケーション生成の応答 (CWBA など) に使用される別名トランザクションの優先順位が、Web 接続タスク (CWXXN や CWXXU など) に関連付けられたトランザクションの優先順位と等しいか、それ以上であることを確認してください。このことは、CICS 提供のデフォルトの代わりにユーザー独自のトランザクション定義をセットアップしている場合にも注意してください。

ユーザーが置換可能なプログラムの変更点

アナライザー・プログラムの変更点

個別 HTTP 要求のための CICS Web サポート処理でのアナライザー・プログラムの使用は、オプションになりました。URIMAP 定義は、要求の URL と、それらを処理するアプリケーション・プログラムとの突き合わせに使用したり、コンバーター・プログラムや別名トランザクションの使用を指定したりできます。ご使用の既存の CICS Web サポート・アーキテクチャーでアナライザー・プログラムが実行するタスクがこれらだけである場合、要求処理パスにおけるその機能を URIMAP 定義で置き換えることができます。

以前の CICS リリースの既存のアナライザー・プログラムが、要求処理時に必要となる追加機能を提供している場合もあります (コンバーター・プログラムへのデータの引き渡し、あるいはコンバーター・プログラムを使用する Web 非対応アプリケーション・プログラム用のコード・ページ変換の変更など)。この場合は、関連要求の URIMAP 定義を使用せずにアナライザー・プログラムを引き続き使用したり、(ANALYZER(YES) オプションを指定して) アナライザー・プログラムを URIMAP 定義と組み合わせたりできます。

アナライザー・プログラムを URIMAP 定義と共に使用する場合、URIMAP 定義の要素 (要求を処理するアプリケーション・プログラムの名前など) が入力としてアナライザー・プログラムに渡されます。既存または新規のアナライザー・プロ

グラムを使用して、これらのエレメントを動的に変更することができます。プロセスにモニターまたは監査アクションを導入するためにアナライザー・プログラムを使用することもできます。

現在も、CICS Web サポートで使用される各 TCPIP SERVICE リソース定義ごとに、アナライザー・プログラムを指定する必要があります。HTTP 要求に関して一致する URIMAP 定義を CICS が検出しなかった場合は、TCPIP SERVICE 定義に指定されたアナライザー・プログラムが呼び出されて、その要求が処理されます。このような状態は、要求 URL 入力のユーザー・エラーによって生じるか、適切な URIMAP 定義がインストールされていないことが原因で発生します。(URIMAP 定義は存在するが使用不可になっている場合、要求はアナライザー・プログラムではなく、Web エラー・プログラムによって処理されます。)

アナライザー・プログラムの COMMAREA で指定される入力パラメーターおよび出力パラメーターに、いくつかの変更が加えられました。

- 一致する URIMAP 定義が要求の処理パスに関わるタイミングを示すために、新しい入力パラメーター `wbra_urimap` が提供されます。
- パラメーター `wbra_alias_tranid`、`wbra_converter_program`、`wbra_server_program`、および `wbra_userid` は、出力パラメーターとしてだけでなく、入力パラメーターとしても使用されるようになりました。URIMAP 定義が使用される場合、URIMAP 定義の TRANSACTION、CONVERTER、PROGRAM、および USERID 属性 (それぞれ上述のパラメーターに対応する) はアナライザー・プログラムにこれらの入力パラメーターとして渡され、アナライザー・プログラムはこれらをオーバーライドすることができます。
- コード・ページ変換用に 2 つの新規出力パラメーターが提供されます。Web クライアントによって使用される文字セットは `wbra_characterset` パラメーターで指定でき、`wbra_hostcodepage` パラメーターには、アプリケーション・プログラムに適したホスト・コード・ページを指定します。CICS は、これらのパラメーターを使用してコード・ページ変換を実行してから、要求をコンバーター・プログラム (使用される場合) またはアプリケーションに渡します。出力パラメーターは、既存の `wbra_dfhcnv_key` パラメーターと機能的に同等ですが、重要な違いは、新規パラメーターを使用すれば、コード・ページ変換テーブル (DFHCNV) のエントリーを作成する必要がないという点です。単に文字セットとホスト・コード・ページを指定するだけで、適切な変換テンプレートが CICS によって決定されます。既存のアナライザー・プログラムが `wbra_dfhcnv_key` パラメーターを使用する場合は、新規パラメーターに変更するか、要求の処理パスからアナライザー・プログラムを除去するまでは、マイグレーションの目的のために関連 DFHCNV エントリーを保持する必要があります。
- 新規出力パラメーター `wbra_commarea` は、EXEC CICS WEB API コマンドを使用しないアプリケーションで CICS TS バージョン 3 以前のバージョンの互換性処理が必要な場所を示すために提供されます。このフラグは、Web クライアントが CICS TS バージョン 3 より前に受信した応答と同一の応答を必要とするような場合に、既存のアプリケーションに対して使用します。このフラグを設定すると、以下のようになります。
 - CICS は、CICS TS バージョン 3 の前に使用されていなかった応答ヘッダーをいっさい追加しません。

- エラー処理が必要な場合、CICS は、Web クライアントの HTTP バージョンにかかわらず、HTTP/1.0 応答に適したエラー応答を、HTTP/1.0 応答のラベル付きで送信します。通常、CICS は HTTP/1.1 クライアントに対して HTTP/1.1 エラー応答を使って応答しますが、このようにすると、アプリケーションは HTTP/1.1 レベルで正常に応答を送信するという誤解をクライアントに与えかねません。

URIMAP 定義は、要求に対してセットアップできますが、必ずアナライザー・プログラムを指定しなければなりません。

CICS が HTTP クライアントの場合や、Web サービス処理用の場合、アナライザー・プログラムを呼び出すことはできません。アナライザー・プログラムは、CICS が HTTP サーバーの場合にのみ呼び出すことができます。

コンバーター・プログラム呼び出しの変更点

HTTP 要求のための CICS Web サポート処理でのコンバーター・プログラムの使用は、オプションのままです。コンバーター・プログラムは、もともと Web 用にコーディングされたものではないアプリケーション・プログラムで主に使用されるものであり、COMMAREA 形式で入力を受信する必要があります。コンバーター・プログラムを使用すれば、このようなアプリケーション・プログラム (1 つまたは複数) からの出力を HTTP メッセージに変換できます。EXEC CICS WEB および EXEC CICS DOCUMENT アプリケーション・プログラミング・インターフェースを使用してコーディングされた Web 対応アプリケーション・プログラムでは、この変換を実行する必要はありません。

URIMAP 定義は、コンバーター・プログラムが HTTP 要求の関連処理を実行するように指定することができます。URIMAP 定義の PROGRAM 属性はコンバーター・プログラムに渡され、コンバーター・プログラムでは、この属性をオーバーライドすることを選択できます。CICS Web サポート処理でアナライザー・プログラムを使用する場合は、アナライザー・プログラムがコンバーター・プログラムを呼び出すこともでき、COMMAREA またはユーザー・トークンでコンバーター・プログラムにデータを渡すことができます。既存のアナライザー・プログラムがこのようにしてデータをコンバーター・プログラムに渡す場合は、URIMAP 定義によってこの機能を複製することはできないので、注意してください。

コンバーター・プログラムは、COMMAREA で受信した要求のコード・ページ変換設定を指定することができません。URIMAP 定義内でコンバーター・プログラムが指定され、Web クライアントの要求のヘッダーによりメッセージ本体がテキストであることが示されている場合、CICS は以下の標準設定を使用して、COMMAREA で提供されるメッセージ本体を変換します。

- 文字セットでは、Web クライアントの要求に Content-Type ヘッダーがあり、そこで CICS がサポートする文字セットが指定されていれば、その文字セットが使用されます。Web クライアントの要求に Content-Type ヘッダーが含まれていない、または指定された文字セットがサポートされていない場合、ISO-8859-1 文字セットが使用されます。
- ホスト・コード・ページでは、CICS は、LOCALCCSID システム初期設定パラメーターに指定されている、ローカル CICS 領域のデフォルトのコード・ページを使用します。

これらの標準設定が適切でない場合、またはコード・ページ変換が必要でない場合、処理パスでアナライザー・プログラムを使用して別の設定を指定する必要があります。

CICS が HTTP クライアントの場合や、Web サービス処理用の場合、コンバーター・プログラムを呼び出すことはできません。コンバーター・プログラムは、CICS が HTTP サーバーの場合にのみ呼び出すことができます。

Web エラー・プログラムの変更点

HTTP サーバーとしての CICS の CICS Web サポート・プロセスで要求エラーまたは異常終了が発生した場合、ユーザーが置換可能な Web エラー・プログラムが Web クライアントにエラー応答を提供します。Web エラー・プログラムは、エラー状態に関する情報、および CICS が Web クライアントに送信する予定のデフォルト HTTP 応答 (状況コードおよび状況テキストを含む) を受信します。このプログラムは、応答をカスタマイズまたは新規作成して、それを送信のために CICS に戻すことができます。

以前の CICS リリースで、Web エラー・プログラム DFHWBEP をカスタマイズしている場合があります。CICS で使用される状況コードが追加されており、一部の既存状況コードについては使用される状態の範囲が広がりました。DFHWBEP を変更して、各状況コードに関連する応答をカスタマイズした場合は、この点に気をつけてください。

EXEC CICS WEB および DOCUMENT アプリケーション・プログラミング・インターフェースは、DFHWBEP からは使用できないので、注意してください。DFHWBEP は、完全な HTTP 応答がデータのバッファーとして作成される COMMAREA ベース・インターフェースを使用します。

統計の変更点

URIMAP リソース定義オブジェクトに関して、新しいグローバル統計およびリソース統計が作成されます。グローバル統計は、動的または静的応答の、突き合せが成功した試行と失敗した試行、リダイレクト、または配信など、URIMAP 定義を伴うさまざまな CICS Web サポート操作の使用回数を示します。リソース統計は、個別の URIMAP 定義について属性および関連する使用回数を表示します。

統計は、URIMAP キーワードを使用して、PERFORM STATISTICS および EXTRACT STATISTICS コマンドによって収集されます。DSECT は DFHWBRDS (リソース統計用) および DFHWBGDS (グローバル統計用) です。DFHSTUP および DFHOSTAT には、これらの統計のための新しいレポートが組み込まれます。

CICS ユーティリティの変更点

統計ユーティリティ・プログラム DFHSTUP

DFHSTUP は、66 ページの『統計の変更点』で説明している統計に対する変更をサポートします。

DFHSTUP の SELECT TYPE および IGNORE TYPE 制御パラメーターで URIMAP リソース・タイプをコーディングできるようになりました。

問題判別の変更点

CICS Web サポートの拡張に関して、新規のメッセージとトレース・ポイント、および 1 つの新規異常終了コードが導入されました。CICS が HTTP メッセージの警告ヘッダーとして受信するすべての情報は、収集されて、TD キュー CWBW に書き込まれます。

HTTP ヘッダーの警告メッセージ

HTTP メッセージに警告ヘッダーがある場合、通常、その中にはユーザーが読むことを意図した情報が含まれています。CICS Web サポートが警告ヘッダー付きのメッセージを受信した場合は、ヘッダーに関連付けられたテキスト、および送信側の IP アドレスが一時データ・キュー CWBW に書き込まれます。(CWBW は CSSL に送られます。) 受信する警告ヘッダーが多すぎる場合は、CWBW 一時データ・キューを除去すれば、これらのレコードを抑制できます。

メッセージ

CICS Web サポートの拡張の結果として、新規 DFHWBxxxx メッセージが導入されました。すべての新規メッセージと変更されたメッセージについては、「*CICS Messages and Codes*」を参照してください。

異常終了コード

新しい異常終了コード AWBP が導入されました。このコードが使用されるのは、HTTP サーバーとしての CICS 用の Web 対応アプリケーションがチャンク化メッセージを送信したが、最後の空チャンクの送信に失敗したためメッセージを完成できなかった場合です。

トレース

CICS Web サポートの拡張に関して、WB ドメイン・トレースに新規 CICS トレース・ポイントが追加されました。新しいトレース・ポイントの範囲は、WB 0419 から WB 0C07 です。

CICS トレース情報の出力を制御するには、通常の方法で CICS トレース制御を使用します。

セキュリティ

新しい SPI および CEMT コマンドのためのセキュリティ

新しい定義済み RACF リソース名は、SPI および CEMT を使用して、以下のリソースへのアクセスを制御します。

HOST

URIMAP

新しいカテゴリー 1 トランザクション

新しい CWXU トランザクションは CICS 内部で使用されるものです。ユーザー端末からは呼び出さないでください。セキュリティ上の目的のために、これはカテゴリー 1 トランザクションに含まれています。

新規グローバル・ユーザー出口ルーチン

CICS が HTTP クライアントである場合、新規グローバル・ユーザー出口ルーチン XWBOPEN (WEB OPEN コマンド上) および XWBSNDO (WEB SEND コマンド上) によって、CICS からのアウトバウンド HTTP クライアント要求に指定されたホスト名およびパスに、セキュリティー・ポリシーを適用することができます。101 ページの『グローバル・ユーザー出口ルーチンの変更点』で、これらの新規の出口ルーチンについて説明しています。

HTTP サーバーとしての CICS による静的応答のセキュリティー

ユーザー作成アプリケーション・プログラムを呼び出さずに応答を提供する URIMAP 定義をセットアップすることによって、CICS 文書および HFS ファイルを、Web クライアントからの要求に対する静的応答として配信することができます。CICS は、HTTP 接続用に HTTP 基本認証をインプリメントできます。ユーザーは、ここで外部セキュリティー・マネージャーに登録された ID およびパスワードを入力する必要があります。ただし、ユーザー ID に基づくアクセス制御を使用するリソース・レベル・セキュリティーは、静的応答として配信された項目には適用されません。項目がこのレベルのセキュリティーを必要とする場合、アプリケーション生成した応答として情報を配信する必要があります。アプリケーションが生成する応答が使用される場合、ユーザー作成アプリケーション・プログラムによる処理を対象とする別名トランザクションにユーザー ID を適用することができるため、アプリケーション・プログラムが使用する特定のリソースおよびコマンドへのアクセスを認可または拒否することができます。

マイグレーション

既存の CICS Web サポート・アプリケーションのマイグレーション

CICS Transaction Server for z/OS バージョン 3 リリース 1 は、Web 対応と Web 非対応の両方のアプリケーション・プログラムに対して既存の CICS Web サポート・アーキテクチャーをサポートするように設計されています。EXEC CICS WEB API コマンドの変更点は、HTTP メッセージを送受信する既存の Web 対応アプリケーション・プログラムが未変更のまま動作できるように設計されているので、これらのプログラムをマイグレーションして現在使用可能な拡張を利用することに決めるまでは、そのまま使用できます。既存の CICS Web サポート・アプリケーションを使用し続ける場合は、マイグレーションに関して次の点に注意してください。

- **非 HTTP 要求の処理に CICS Web サポートを使用している場合は、これらの要求用のポートを定義する TCPIP SERVICE 定義で新規 USER プロトコルを指定してください。** これは、非標準の要求メソッドを持つ HTTP 要求にも該当します。このような HTTP 要求を HTTP プロトコルで受信した場合、その要求は拒否されるようになりました (以前はこれらの要求は受信され、非 HTTP として処理されました)。すべての非 HTTP 要求の処理は USER プロトコルのもとで実行しなければならなくなったため、非 HTTP 要求は、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の対象にはなりません。これらの要求には、非 HTTP のフラグが立てられ、TCPIP SERVICE のためのアナライザー・プログラムに未変更のまま渡されます。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。

注: 各ポートにはアクティブな TCPIP SERVICE 定義が 1 つしか存在しないので、非 HTTP 要求は HTTP 要求と同じポートを使用できなくなりました。ウェルノウン・ポート番号 80 (HTTP 用) および 443 (HTTPS 用) は HTTP プロトコルでなければならないため、非 HTTP 要求を受信することはできません。Web クライアントは、要求のポートが変更された場合はそのポートを URL に指定する必要があります。

- **HTTP プロトコルを持つ TCPIP SERVICE リソース定義の設定を確認します。**
 1. SOCKETCLOSE 属性は、ゼロ設定 (SOCKETCLOSE(0)) にしないでください。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かぎり、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。つまり、持続接続を維持することができません。SOCKETCLOSE にゼロ以外を設定すれば、HTTP/1.1 クライアントおよび HTTP/1.0 クライアントの両方との持続接続が可能です (クライアントがこれをサポートしている場合)。
 2. 新規 MAXDATALEN オプションを指定して、HTTP サーバーとしての CICS が受信できるデータの最大長を制限する必要があります。これを設定すれば、大量のデータ送信を伴うサービス妨害攻撃からの保護に役立ちます。
 3. SSL を使用している場合は、TCPIP SERVICE リソース定義で使用可能なセキュリティ・オプションに対する変更があります。
- **コード・ページ変換テーブル (DFHCNV) は、CICS Web サポートには必要なくなりました。**ただし、以前の CICS リリースでコーディングしたアナライザー・プログラムを引き続き使用して DFHCNV を参照する場合は、コード・ページ変換テーブルのエントリを提供し続けるか、アナライザー・プログラムを変更しなければなりません。アナライザー・プログラムを変更する場合は、DFHCNV エントリの名前を指定した出力パラメーターの代わりに、クライアントおよびサーバーのコード・ページを指定する 2 つの新規出力パラメーターをコーディングする必要があります。これを行う場合は、ご使用の DFHCNV エントリをマイグレーションする必要はありません。
- **ユーザーが置換可能な Web エラー・プログラム DFHWBEP を変更して、エラー状態で提供される HTTP 応答をカスタマイズした場合は、CICS で使用される状況コードが追加されたこと、および一部の既存状況コードについては使用される状態の範囲が広がったことに注意してください。**
 1. ご使用のプログラムは、カスタマイズ応答が適用される状態を識別する際に、状況コードだけに頼るのではなく、適切な範囲の入力パラメーターを使用していることを確認してください。HTTP 応答を引き起こした状態を識別するには、エラー・コード、異常終了コード、メッセージ番号、応答および理由コード、またはプログラム名を使用できます。この確認を行わないと、CICS が状況コードを新しい目的のために使用している場合、不適切なカスタマイズ応答が戻される可能性があります。
 2. ご使用のプログラムには、そのプログラムにとって未知の状況コードを持つ HTTP 応答を未変更のままパススルーするロジックが組み込まれていることを確認してください。

- **DFHWBCLI** インターフェースは現在も **CICS Transaction Server for z/OS** バージョン 3 リリース 1 でサポートされています。拡張された機能性を確保するには、DFHWBCLI インターフェースを使用した HTTP クライアント・アプリケーションをマイグレーションすれば、クライアント要求に対して EXEC CICS WEB API コマンドを (SESSTOKEN オプション付きで) 使用できます。注意すべき重要な違いは、EXEC CICS WEB API では、プロキシ・サーバーの使用は WEB OPEN コマンドのユーザー出口 (XWBOPEN) によって指定され、プロキシ・サーバーの URL はこのユーザー出口によって提供される、という点です。75 ページの『第 3 章 CICS アプリケーションからの HTTP クライアント要求のサポート』で、HTTP クライアント要求が現在ではどのように行われるかについて説明します。

新規 CICS Web サポート機能へのマイグレーション

CICS Transaction Server for z/OS バージョン 3 リリース 1 の CICS Web サポートには、これまではユーザーが置換可能なプログラムによって処理されていた機能を自動で管理制御するための拡張が多数あります。特に、CICS Web サポート・アーキテクチャーの以下のエレメントについては、マイグレーションの可能性を調査することをお勧めします。

- 通常、既存アナライザー・プログラムの要求処理機能は URIMAP リソース定義に置き換えることができなければなりません。このリソース定義は、CICS システム・プログラミング・コマンドを使用して変更および制御することができます。URIMAP 定義は、要求の URL を突き合わせてそれらをアプリケーション・プログラムにマップしたり、コンバーター・プログラムおよび別名トランザクションを指定したりするのに使用できます。アナライザー・プログラムが追加機能を提供するようにカスタマイズされている場合は、URIMAP 定義を使用せずにアナライザー・プログラムを引き続き使用したり、アナライザー・プログラムを URIMAP 定義と組み合わせたりできます。URIMAP を使用するようにマイグレーションする場合の注意点を以下に示します。
 1. 一度に少数の要求に関して徐々に URIMAP リソース定義を導入できます。アナライザー・プログラムによって実行される処理のタイプ、および要求を処理するアプリケーションのタイプに応じて、各要求ごとに処理パスでアナライザー・プログラムを引き続き使用するかしないかを選択できます。
 2. 既存の URL を保存するのではなく、URIMAP リソース定義によって処理される要求の新規 URL を選択して発行する方が好ましい場合もあります。要求の古い処理パスの使用を中止する準備ができれば、URIMAP 定義をセットアップすれば、要求を古い URL から新しい URL へ永続的にリダイレクトできます。
 3. 要求の処理パスに関係しなくなった場合でも、TCPIPSERVICE に指定されたアナライザー・プログラムには基本エラー処理プロシージャがまだ含まれていることを確認してください。アナライザー・プログラムはまだ置いておく必要があります。URIMAP マッチングが失敗した場合は、アナライザー・プログラムが要求を受信します。
- EXEC CICS WEB API コマンドを使用しないが COMMAREA で HTTP 応答を作成するアプリケーション・プログラムの場合、CICS Web サポートは、メッセージ構造の正しい組み立てを支援したり、応答に関する全範囲の検査を実行することはできません。使用可能なすべての CICS Web サポート機能を利用するに

は、これらのアプリケーションを、WEB API コマンドを使用する Web 対応アプリケーション・プログラムに変換するよう計画することをお勧めします。

- URIMAP リソース定義を使用すれば、CICS 文書または HFS ファイルの内容を静的応答として引き渡したり、リダイレクト応答をユーザー作成のアプリケーション・プログラムを必要とせずに引き渡したりできます。動的処理を必要としない単純な応答に対しては、アプリケーション・プログラムの代わりにこのメカニズムを使用することも検討できます。
- コード・ページ変換が最も効率的な方法で実行されていることを確認します。アプリケーションに小さな変更を加えれば、新規 CICS Web サポート機能を利用して以下の処理を実現できます。
 - CICS Web サポートのコード・ページ変換テーブル (DFHCNV) のセットアップおよび使用を回避する。
 - Web クライアントのコード・ページ変換用文字セットを、ユーザーが指定するのではなく、CICS が識別および使用できるようにする。
 - アプリケーション・プログラムのコード・ページを、ユーザーが指定するのではなく、ローカル・システムのデフォルト (LOCALCCSID システム初期設定パラメーター) を使用して識別する。
 - UTF-8 文字セットと UTF-16 文字セットの間で変換する。

CICSplex SM サポート

CPSM Web ユーザー・インターフェース (WUI) には、Web ブラウザーでアクセスします。このインターフェースは、HTTP/1.0 および HTTP/1.1 レベルの要求を受信しますが、HTTP/1.0 応答のみを提供します。CPSM WUI へのアクセスに使用する Web ブラウザーは、HTTP/1.0 応答を受信し、理解できなければなりません。

CICSplex SM エンド・ユーザー・インターフェース・ビューの変更点

以下の EUI ビューに対して変更が加えられました。

- 『TCPDEF ビュー』
- 170 ページの『DOCDEF ビュー』
- 170 ページの『DOCTEMP ビュー』

TCPDEF ビュー

以下の属性が TCPDEF ビューに追加されました。

PROTOCOL CVDA (USER)

プロトコル

MAXDATALEN

インバウンド要求またはアウトバウンド要求への応答で受信できるデータの最大長を定義します。

#

このビューの **PRIVACY** 属性は、CICS Transaction Server 3.1 では無効になったため、無視されます。

DOCTEMP ビュー

以下の属性が **DOCTEMP** ビューに追加されました。

HFSFILE

文書テンプレートとして使用する UNIX システム・サービス HFS ファイル。

TEMPLATETYPE CVDA (HFS)

文書テンプレート・タイプ

DOCDEF ビュー

DOCDEF ビューに、以下の新しい属性が追加されました。

HFSFILE

文書テンプレートとして使用する UNIX システム・サービス HFS ファイル。

属性 **TEMPLATETYPE** は除去されました。

CICSplex SM アプリケーション・プログラミング・インターフェースの変更点

新規リソース・テーブル

以下のテーブルが導入されました。

- 『URIMAP リソース・テーブル』
- 172 ページの 『URIMPDEF リソース・テーブル』
- 174 ページの 『URIMPGBL リソース・テーブル』
- 174 ページの 『HOST リソース・テーブル』

URIMAP リソース・テーブル

URIMAP リソース・テーブルには、以下の SPI 属性があります。

URIMAP

URIMAP 名

ENABLESTATUS

マップを **ENABLED** または **DISABLED** のいずれの状態インストールするかを示す **CVDA** 値を表示します。

USAGE

サーバー、クライアント、またはパイプラインの **CVDA** 値を表示します。この値は、マップがそれぞれ HTTP サーバーとしての CICS 用、HTTP クライアントとしての CICS 用、または Web サービス用のいずれであるかを示します。

SCHEME

URI マップが適用される URI のスキーム部分が **SSL** を使用しない (HTTP) か **SSL** を使用する (HTTPS) かを示す **CVDA** 値を表示します。

HOST マップが適用される URI の 116 文字のホスト部分を表示します。

PATH マップが適用される URI の 225 文字のパス部分を表示します。

MEDIATYPE

インバウンド HTTP 要求に対して CICS が提供する静的応答の 56 文字のデータ・コンテンツを表示します。

CHARACTERSET

インバウンド HTTP 要求に対して送信される静的応答を CICS が変換する文字セットの 40 文字の名前を表示します。

HOSTCODEPAGE

静的応答を形成するテキスト文書のエンコードに使用する IBM コード・ページ (EBCDIC) の 10 文字の名前を表示します。

TEMPLATENAME

静的応答を形成する CICS 文書テンプレートの 48 文字の名前を表示します。

HFSFILE

インバウンド HTTP 要求に送信される静的応答を形成する UNIX システム・サービス HFS ファイルの 255 文字の完全修飾名を表示します。

TCPIPSERVICE

URI マップが適用されるインバウンド・ポートを定義する TCPIP SERVICE リソース定義の 8 文字の名前を表示します。

ANALYZER

アナライザー・プログラムがインバウンド HTTP 要求の処理に関与するかどうかを示す CVDA 値 (YES または NO) を表示します。

CONVERTER

要求の内容を処理するコンバーター・プログラムの 8 文字の名前を表示します。

TRANSACTION

応答を作成するユーザー・アプリケーションを実行するために使用される別名トランザクションの 4 文字の名前を表示します。

PROGRAM

要求を処理するサービスを指定するユーザー・アプリケーション・プログラムまたは XML サービスの 8 文字の名前を表示します。

PIPELINE

要求を処理するサービスを指定するパイプラインの 8 文字の名前を表示します。

WEBSERVICE

要求を処理するサービスを指定する Web サービスの 32 文字の名前を表示します。

USERID

別名トランザクションに接続するために使用する 8 文字のユーザー ID を表示します。

CERTIFICATE

アウトバウンド HTTPS 要求で URI を使用するとき SSL クライアント証明書として使用される証明書の 32 文字のラベルを表示します。

#

CIPHERS
アウトバウンド SSL 要求に使用される 2 桁の暗号スイート・コード (最大 28 個) のリストとして解釈される 56 個以内の 16 進数字のストリングを表示します。

NUMCIPHERS
SSL 暗号スイート・コードの数

LOCATION
インバウンド HTTP 要求のリダイレクト先となる 255 文字の URI を表示します。

REDIRECTTYPE
リダイレクトのタイプに対する CVDA 値 (NONE、TEMPORARY、または PERMANENT) を表示します。

MAPREFCOUNT
URI マップ参照数

MATCHDISABLD
URI マップ・ホストまたはパスが使用不可

MATCHREDIREC
URI マップ・ホストまたはパスのリダイレクト

URIMPDEF リソース・テーブル

URIMPDEF リソース・テーブルには、以下の RDO 属性が含まれます。

URIMAP
URI マップ定義名

STATUS
マップを ENABLED または DISABLED のいずれの状態にインストールするかを示す CVDA 値を表示します。

USAGE
サーバー、クライアント、またはパイプラインの CVDA 値を表示します。この値は、マップがそれぞれ HTTP サーバーとしての CICS 用、HTTP クライアントとしての CICS 用、または Web サービス用のいずれであるかを示します。

SCHEME
URI マップが適用される URI のスキーム部分が SSL を使用しない (HTTP) か SSL を使用する (HTTPS) かを示す CVDA 値を表示します。

HOST マップが適用される URI の 116 文字のホスト部分を表示します。

PATH マップが適用される URI の 225 文字のパス部分を表示します。

MEDIATYPE
インバウンド HTTP 要求に対して CICS が提供する静的応答の 40 文字のデータ・コンテンツを表示します。

CHARACTERSET
インバウンド HTTP 要求に対して送信される静的応答を CICS が変換する文字セットの 40 文字の名前を表示します。

HOSTCODEPAGE

静的応答を形成するテキスト文書のエンコードに使用する IBM コード・ページ (EBCDIC) の 10 文字の名前を表示します。

TEMPLATENAME

静的応答を形成する CICS 文書テンプレートの 48 文字の名前を表示します。

HFSFILE

インバウンド HTTP 要求に送信される静的応答を形成する UNIX システム・サービス HFS ファイルの 255 文字の完全修飾名を表示します。

TCPIPSERVICE

URI マップが適用されるインバウンド・ポートを定義する TCPIPSERVICE リソース定義の 8 文字の名前を表示します。

ANALYZER

アナライザー・プログラムがインバウンド HTTP 要求の処理に関与するかどうかを示す CVDA 値 (YES または NO) を表示します。

CONVERTER

要求の内容を処理するコンバーター・プログラムの 8 文字の名前を表示します。

TRANSACTION

応答を作成するユーザー・アプリケーションを実行するために使用される別名トランザクションの 4 文字の名前を表示します。

PROGRAM

要求を処理するサービスを指定するユーザー・アプリケーション・プログラムまたは XML サービスの 8 文字の名前を表示します。

PIPELINE

要求を処理するサービスを指定するパイプラインの 8 文字の名前を表示します。

WEBSERVICE

要求を処理するサービスを指定する Web サービスの 32 文字の名前を表示します。

USERID

別名トランザクションに接続するために使用する 8 文字のユーザー ID を表示します。

CERTIFICATE

アウトバウンド HTTPS 要求で URI を使用するとき SSL クライアント証明書として使用される証明書の 32 文字のラベルを表示します。

CIPHERS

#

アウトバウンド SSL 要求に使用される 2 桁の暗号スイート・コード (最大 28 個) のリストとして解釈される 56 個以内の 16 進数字のストリングを表示します。

LOCATION

インバウンド HTTP 要求のリダイレクト先となる 255 文字の URI を表示します。

REDIRECTTYPE

リダイレクトのタイプに対する CVDA 値 (NONE、TEMPORARY、または PERMANENT) を表示します。

URIMPGBL リソース・テーブル

URIMPGBL リソース・テーブルには、以下の属性が含まれます。

MAPREFCOUNT

URI マップ参照数

MATCHDISABLD

URI マップ・ホストまたはパスが使用不可

NOMATCHCOUNT

URI マップ・ホストまたはパスの不一致数

MATCHCOUNT

URI マップ・ホストまたはパスの一致数

MATCHREDIREC

URI マップ・ホストまたはパスのリダイレクト

MATCHANALYZE

URI マップ・ホストまたはパスの一致アナライザー

STATICCONTENT

URI マップの静的コンテンツ

DYNAMCONTENT

URI マップの動的コンテンツ

PIPELINEREQS

URI マップのパイプライン要求

SCHEMEHTTP

URI マップのスキーム (HTTP) 要求

SCHEMEHTTPS

URI マップのスキーム (HTTPS) 要求

HOSTDISABLED

使用不可ホストの数

HOST リソース・テーブル

HOST リソース・テーブルには、以下の属性があります。

HOSTNAME

ホスト名

ENABLESTATUS

ホストを ENABLED または DISABLED のいずれの状態にインストールするかを示す CVDA 値を表示します。

リソース・テーブルの変更点

以下のテーブルに対して変更が加えられました。

- 175 ページの『TCPDEF リソース・テーブル』

- 『DOCTEMP リソース・テーブル』
- 『DOCDEF リソース・テーブル』
- 『TASK リソース・テーブル』

TCPDEF リソース・テーブル

TCPDEF リソース・テーブルには、以下の追加属性があります。

PROTOCOL CVDA (USER)

プロトコル

MAXDATALEN

送受信できるデータの最大長を定義します。

#

このリソース・テーブルの **PRIVACY** 属性は、CICS Transaction Server 3.1 (またはそれ以降のバージョン) では無効になったため、無視されます。

DOCTEMP リソース・テーブル

DOCTEMP リソース・テーブルには、以下の追加属性があります。

HFSFILE

UNIX システム・サービスの階層ファイル・システムのテンプレート・ファイル

TEMPLATETYPE

文書テンプレート・タイプ

DOCDEF リソース・テーブル

DOCDEF リソース・テーブルには、以下の属性が含まれます。

HFSFILE

階層ファイル・システムのテンプレート・ファイル

TASK リソース・テーブル

TASK リソース・テーブルには、以下の属性が含まれます。

TMRWBRDL

リポジトリから読み取られるデータのデータ長を表示します。

TMRWBWDL

リポジトリに書き込まれるデータのデータ長を表示します。

CICSplex SM Web ユーザー・インターフェースの変更点

新規 WUI ビュー

以下の WUI ビューが導入されました。

- 176 ページの『URI マッピング定義ビュー』
- 176 ページの『URI マップ・ビュー』
- 177 ページの『URI マップ・グローバル・ビュー』
- 177 ページの『ホスト・ビュー』

URI マッピング定義ビュー

「URI マッピング定義」という名前で、新規 URIMPDEF リソース・テーブルに関連する定義ビュー・セットが導入されました。この表形式ビューのビュー名は EYUSTARTURIMPDEF.TABULAR です。このビューを組み込むために、既存の EYUSTARTADMRES メニューが拡張されました。

「URI マッピング定義」ビューを開くには、以下の手順を実行します。

1. メインメニューから「管理ビュー」をクリックします。
2. 「ベーシック CICS リソース管理ビュー」をクリックします (あるいは、「完全に機能しているビジネス・アプリケーション・サービス (BAS) 管理ビュー」をクリックします)。
3. 「CICS リソース定義」をクリックします。
4. スクロールダウンして、「URI マッピング定義」をクリックします。

「URI マッピング定義」ビューが表示されます。このビューには、次の 5 つのアクション・ボタンが含まれます。

- 作成
- 更新
- 除去
- インストール
- リソース・グループへの追加

フィールド詳細については、172 ページの『URIMPDEF リソース・テーブル』にリストされている URIMPDEF リソース・テーブルの属性を参照してください。

URI マップ・ビュー

「URI MAP」という名前で、新規 URIMAP リソース・テーブルに関連する新しいビューが導入されました。この表形式ビューのビュー名は EYUSTARTURIMAP.TABULAR です。このビューを組み込むために、既存の EYUSTARTTCPIPS メニューが拡張されました。

「URI MAP」ビューを開くには、以下の手順を実行します。

1. メインメニューから「CICS 操作ビュー」をクリックします。
2. スクロールダウンして、「TCP/IP サービス操作ビュー」をクリックします。
3. 「URI MAP」をクリックします。

「URI MAP」ビューが表示されます。このビューには、次の 4 つのアクション・ボタンが含まれます。

- 設定属性
- 使用可能
- 使用不可
- 破棄

URIMAP リソース・テーブルの属性詳細については、170 ページの『URIMAP リソース・テーブル』を参照してください。

URI マップ・グローバル・ビュー

「URI マップ・グローバル (URI map global)」という名前で、新規 URIMPGBL リソース・テーブルに関連する新しいビューが導入されました。この表形式ビューのビュー名は EYUSTARTURIMPGBL.DETAILED です。このビューを組み込むために、既存の EYUSTARTTCPIPS メニューが拡張されました。

「URI マップ・グローバル (URI map global)」ビューを開くには、以下の手順を実行します。

1. メインメニューから「**CICS 操作ビュー**」をクリックします。
2. スクロールダウンして、「**TCP/IP サービス操作ビュー**」をクリックします。
3. 「**URI マップ・グローバル (URI map global)**」をクリックします。
4. 「**CICS システム名**」を選択して、「**URI マップ・グローバル (URI map global)**」の詳細ビューを開きます。

「URI マップ・グローバル (URI map global)」ビューが表示されます。URIMPGBL リソース・テーブルの属性詳細については、174 ページの『URIMPGBL リソース・テーブル』を参照してください。

HOST・ビュー

「HOST」という名前で、新規 HOST リソース・テーブルに関連する新しいビューが導入されました。この表形式ビューのビュー名は EYUSTARTHOST.TABULAR です。このビューを組み込むために、既存の EYUSTARTTCPIPS メニューが拡張されました。

「HOST」ビューを開くには、以下の手順を実行します。

1. メインメニューから「**CICS 操作ビュー**」をクリックします。
2. スクロールダウンして、「**TCP/IP サービス操作ビュー**」をクリックします。
3. 「**HOST**」をクリックします。

「HOST」ビューが表示されます。

HOST リソース・テーブルの属性詳細については、174 ページの『HOST リソース・テーブル』を参照してください。

WUI ビューの変更点

以下のビューに対して変更が加えられました。

- 『TCP/IP サービス定義ビュー』
- 178 ページの『文書テンプレート定義ビュー』
- 178 ページの『文書テンプレート・ビュー』
- 178 ページの『TCP/IP 使用ビュー』

TCP/IP サービス定義ビュー

TCPDEF (EYUSTARTTCPDEF) ビューに、以下の属性が追加されました。

PROTOCOL CVDA (USER)

プロトコル

MAXDATALEN

インバウンド要求またはアウトバウンド要求への応答で受信できるデータの最大長を定義します。

このビューの **PRIVACY** 属性は、CICS Transaction Server 3.1 では無効になったため、無視されます。

文書テンプレート定義ビュー

DOCDEF (EYUSTARTDOCDEF.DETAILED) ビューに、以下の属性が追加されました。

HFSFILE

文書テンプレートとして使用する UNIX システム・サービス HFS ファイル。

文書テンプレート・ビュー

DOCTEMP (EYUSTARTDOCTEMP.DETAILED) ビューに、以下の属性が追加されました。

HFSFILE

文書テンプレートとして使用する UNIX システム・サービス HFS ファイル。

TEMPLATETYPE CVDA (HFS)

文書テンプレート・タイプ

TCP/IP 使用ビュー

タスク操作ビュー内のアクティブ・タスク・ビュー・セット (EYUSTARTTASK.DETAILED) の「**TCP/IP 使用**」ビューに、以下の属性が追加されました。

TMRWBRDL

リポジトリーから読み取られるデータのデータ長を表示します。

TMRWBWDL

リポジトリーに書き込まれるデータのデータ長を表示します。

第 6 章 インターネット・セキュリティの改善

セキュリティの拡張の概要

セキュリティの改善はさまざまな領域にわたっています。以下に要約を示します。

- TLS 1.0 プロトコルおよび AES 暗号スイートのサポート
- 同時 SSL 接続数の増加によるスケーラビリティ
- 証明書取り消しリストのサポート
- シスプレックスの複数 CICS 領域間での SSL キャッシング
- 最小および最大暗号化ネゴシエーション・レベルの指定

注: このリリースでは、CICS は SSL と TLS の両方のセキュリティ・プロトコルをサポートします。理解しやすくするため、本書では、SSL という用語は両方のプロトコルを指すものとして使用します。ただし、どちらかのプロトコルに関する特定の意味が必要とされる場合を除きます。

インターネット・セキュリティの改善の利点

セキュリティの改善により、さまざまな利点がもたらされます。

CICS は SSL 3.0 に加えて Transport Layer Security (TLS) 1.0 プロトコルをサポートするようになったため、128 ビット暗号化および 256 ビット暗号化を提供する新しい AES 暗号スイートを使用できるようになりました。

SSL のパフォーマンスが向上し、Web サービスなどの新しい機能がサポートされるようになりました。システムで同時に使用できる同時 SSL 接続の数が増加したため、スループットが改善されています。

クライアントとサーバーの間での暗号化ネゴシエーションの制御における柔軟性が増しました。特定ユーザーとのネゴシエーションのために、CICS で最小および最大の暗号化レベルを指定できます。

CICS は、クライアントとのネゴシエーション時に証明書取り消しリスト (CRL) と照らし合わせてすべての証明書を検査できるようになりました。取り消された証明書を使用している接続は、即時にクローズされます。

SSL キャッシュを使用してシスプレックス全体でセッション ID を共用するかどうかを指定できます。クライアントが以前に CICS とネゴシエーションしたことがある場合、CICS は部分的な SSL ハンドシェイクを実行します。複数の CICS 領域間でキャッシュを共用すれば、SSL ネゴシエーションおよび接続スループットのパフォーマンスが改善されます。

セキュリティ用語

以下の用語は、CICS でのインターネット・セキュリティの拡張について説明する場合に使用されます。

Transport Layer Security (TLS)

インターネット上でセキュア通信を提供するために使用されるセキュリティ・プロトコル。RFC2246 で仕様が文書化されています。

暗号スイート

SSL ハンドシェイク時にネゴシエーションされる、暗号化アルゴリズム、暗号鍵の長さ、および MAC アルゴリズムの組み合わせ。

メッセージ確認コード (MAC)

SSL 接続経由で送信された各メッセージに関連付けられた、暗号によるセキュア・ハッシュ・コード。

MAC アルゴリズム

メッセージ確認コードを計算する暗号アルゴリズム。SSL では MD5 および SHA アルゴリズムを使用します。

SSL キャッシュ

クライアントとの暗号化ネゴシエーションに関するセッション ID 情報を保管するために SSL で使用するキャッシュ。クライアントが以前に SSL を使用して CICS に安全に接続している場合は、部分的なハンドシェイクのみを実行して SSL 接続を確立します。

証明書取り消しリスト

認証局と呼ばれる独立した機関によって提供される取り消し済み証明書のリスト。証明書が撤回された場合、その証明書は証明書取り消しリストに追加されます。これらのリストは、クライアントとサーバーが互いに認証を試みる際の SSL ハンドシェイク・ネゴシエーション中に相互参照することができます。

SP モード

ソケット pthread 専有タスクを所有する TCB モード。

SSL プール

CICS 領域の S8 TCB の保管および管理を行うプール。

SSL ハンドシェイク

接続確立時にクライアントとサーバーの間で行われる情報交換。ハンドシェイクには、使用する暗号化アルゴリズムのネゴシエーションおよび相互認証が含まれます。

要件

ハードウェア

z/OS Integrated Cryptographic Facility (ICSF) に依存している SSL 暗号化のパフォーマンス向上の恩恵を完全に受けるには、zSeries® 暗号ハードウェアが必要です。

ソフトウェア

128 ビット暗号化かそれ以上の暗号スイートを使用する場合は、z/OS の System SSL セキュリティ・レベル 3 機能が必要です。

Transport Layer Security プロトコル

Transport Layer Security (TLS) 1.0 プロトコルは、インターネット上の安全な通信を提供する標準のセキュリティー・プロトコルです。

CICS では、インターネット経由で保護された通信を提供するのに使用する 2 つの
セキュリティー・プロトコルをサポートします。最初は、Secure Sockets Layer
(SSL) 3.0 プロトコルです。2 番目は、Transport Layer Security (TLS) 1.0 プロトコ
ルで、これは最新の業界標準 SSL プロトコルであり、SSL 3.0 に基づいています。
TLS 1.0 の仕様は、RFC2246 に文書化されており、インターネットで
www.rfc-editor.org/rfcsearch.html から参照できます。暗号化が必要な接続では、クラ
イアントが特に SSL 3.0 を必要としなければ、自動的に TLS が使用されます。

TLS の主な目的は、Secure Sockets Layer の保護性を強め、プロトコルの仕様を厳
密かつ完全にすることです。TLS では、SSL 3.0 と比較して以下の機能拡張が行わ
れています。

•

メッセージ認証のキー・ハッシュ

TLS は、メッセージ認証コード (HMAC) のキー・ハッシュを使用しま
す。これにより、インターネットのようなオープンなネットワークを通過
する間にレコードを変更することができなくなります。SSL バージョン
3.0 では、キー・メッセージ認証も提供されます。ただし、HMAC は、
SSL バージョン 3.0 で使用される MAC (メッセージ認証コード:
Message Authentication Code) 機能よりも安全であると考えられていま
す。

•

拡張疑似乱数関数 (PRF)

PRF は、キー・データの生成に使用されます。TLS では、PRF は
HMAC とともに定義されます。PRF は、2 つのハッシュ・アルゴリズム
を使用してセキュリティーを保証します。どちらかのアルゴリズムが公開
されている場合、2 番目のアルゴリズムが公開されない限り、データは保
護され続けます。

•

完了メッセージ検査の改良

TLS 1.0 と SSL 3.0 では、交換されたメッセージが変更されていないこ
とを認証する完了メッセージが両方のエンドポイントに提供されます。た
だし、TLS では、この完了メッセージは PRF および HMAC の値に基づ
いており、SSL バージョン 3.0 よりも安全です。

•

一貫性のある証明書処理

SSL 3.0 とは異なり、TLS では、TLS のインプリメンテーション間で交
換する必要のある証明書のタイプが指定されています。

•

特定の警告メッセージ

TLS では、セッション・エンドポイントのどちらかで検出された問題を

示すために、特定の警告や追加の警告が提供されます。TLS では、特定
の警告がいつ送信されるかについても文書化されています。

#

SSL パフォーマンスの向上

CICS の SSL 接続数と SSL 接続のパフォーマンスの向上のため、S8 TCB のインプリメンテーションにいくつかの変更が行われました。

CICS はオープン・トランザクション環境 (OTE) を使用して、SSL 接続を管理します。各 SSL 接続では、S8 TCB を使用します。この S8 TCB が、UNIX の pthread として実行されるようになりました。ソケット pthread 専有タスクで使用される、SP というオープン TCB モードも新しく追加されました。すべての S8 TCB は単一エンクレープ内で実行されます。このエンクレープは、SP TCB によって所有され、SSL キャッシュも備えています。これにより、ストレージを標準以下に節約することで、CICS の同時 SSL 接続数を以前のリリースより増やすことができます。

S8 TCB は、SSL プール内に格納されます。この SSL プールは、CICS ディスパッチャーによって管理されます。S8 TCB は、この新しい SSL プールから割り振られますが、SSL 機能を実行する必要がある期間のみトランザクションに対してロックされます。SSL ネゴシエーションが完了すると、TCB は解放されて SSL プールに戻され、再利用できるようになります。MAXSSLTCBS システム初期設定パラメーターは、SSL プール内の S8 オープン TCB の最大数を指定します。デフォルト値は 8 ですが、最大 1024 まで指定できます。

DFH0STAT および DFHSTUP からのディスパッチャー・レポートを使用して、SSL プールおよび S8 TCB のパフォーマンスをモニターすることができます。この統計には、S8 TCB の最大数に達した回数、TCB が割り振られるまでの遅延時間、および SSL プール内の TCB の実際の数に関する情報が含まれます。

証明書取り消しリストの使用

証明書取り消しリスト (CRL) を使用するように CICS を構成して、SSL ネゴシエーションで使用されるクライアント証明書の妥当性を検査することができます。

証明書取り消しリストを使用するには、LDAP サーバーをインストールし、構成する必要があります。この作業の実行方法の詳細は、「z/OS VIR4.0 Security Server LDAP サーバー 管理および使用ガイド」にあります。

証明書取り消しリストは、Verisign などの認証局から入手できます。証明書取り消しリストは、ワールド・ワイド・ウェブで利用可能な CRL リポジトリに保管されており、ダウンロードして LDAP サーバーに保管することができます。LDAP サーバーにデータを取り込み証明書取り消しリストを更新するには、CICS 提供トランザクション CCRL を指定します。CCRL トランザクションは、端末から、または START コマンドを使用して実行できます。定期的な更新をスケジュールするには、START コマンドを使用します。LDAP サーバーに CRL を組み込むには、以下の手順を実行してください。

1. LDAP サーバーを構成して、使用する認証局を指定します。
2. CRLSERVER システム初期設定パラメーターに LDAP サーバーの名前を指定します。

3. CCRL トランザクションを実行します。

CCRL トランザクションを実行すると、CICS が指定した URL から CRL をダウンロードし、CRLSERVER システム初期設定パラメーターに指定された LDAP サーバーにこれを保管します。

SSL キャッシュ

SSL キャッシュは、クライアントと CICS 間のネゴシエーションからのセッション ID を保管するために使用します。

SSL キャッシュを使用すると、CICS は、以前に認証したことのあるクライアントとの部分的ハンドシェイクを実行することができます。ローカルの CICS 領域では、SSL キャッシュは S8 TCB のためのエンクレープの一部です。ご使用の CICS システムで適切な場合は、SSL キャッシュをシस्पレックス全体で共用することができます。複数の CICS ソケット専有領域が同じ IP アドレスで複数の SSL 接続を受け入れる場合は、シस्पレックス・キャッシングを使用できます。

複数の領域間でキャッシュを共用したい場合は、システム SSL 開始タスク GSKSRVR を活動化し、システム初期設定パラメーター SSLCACHE を使用します。デフォルトではローカル領域のキャッシュを使用しますが、オプション SYSPLEX を指定することにより、これを変更できます。CICS は、代わりにカップリング・ファシリティーの SSL キャッシュを使用してセッション ID を保管するようになります。

暗号化ネゴシエーションのカスタマイズ

暗号化ネゴシエーション・プロセスで使用する暗号スイートを選択し、暗号化の最小レベルおよび最大レベルを設定することができます。

リソース定義 TCPIPSERVICE、CORBASERVER、および URIMAP の CIPHERS 属性は、各暗号化レベルで使用できる暗号スイートを指定します。この属性のデフォルト値は、暗号化ネゴシエーションで使用する 2 桁の暗号コードのリストです。この暗号スイートのリストをカスタマイズして、CICS がクライアントをネゴシエーションする暗号化レベルの優先順序を指定することができます。また、このリストから暗号スイートを除去することを選択することもできます。これは、非常に高いレベルの暗号化のみが使用されることを保証したい場合に特に便利です。これを行うには、以下のようにします。

1. 変更するリソース定義を選択します。
2. CIPHERS 属性にデフォルト値が表示されます。例えば、システム初期設定パラメーター ENCRYPTION=STRONG の場合、デフォルト値は 0504352F0A0903060102 です。
3. 暗号スイートを除去または再配列するために属性値を編集します。例えば、352F0A0504 と指定できます。
4. リソース定義を保管します。

352F0A0504 を指定した場合、これは、CICS がこのリソースを使用した接続で 128 ビット暗号化より下のレベルでネゴシエーションしないことを意味します。この属性のそれぞれの 2 桁コード (例えば、35、2F、0A など) は、128 ビット暗号化以

上の暗号スイートを表します。CICS は、AES 暗号スイート 35 および 2F を使用してネゴシエーションを開始します。なぜなら、これらは、暗号コードのリストの最初にあるからです。クライアントがこのレベルの暗号化を使用していない場合、CICS は接続をクローズします。

暗号化レベルのデフォルト値にない暗号スイートを組み込むことはできないことに注意してください。例えば、MEDIUM レベルの暗号化を指定した場合、CIPHERS 属性に AES 暗号スイート 35 および 2F を追加することはできません。

CICS 外部インターフェースの変更点

システム初期設定パラメーターの変更点

インターネット・セキュリティの向上のために、新しいシステム初期設定パラメーターが追加されました。また、一部のシステム初期設定パラメーターが変更されました。変更されたパラメーターは、次のとおりです。

ENCRYPTION={STRONG|WEAK|MEDIUM}

セキュア TCP/IP 接続のために CICS で使用する暗号スイートを指定します。前のリリースとの互換性のために、ENCRYPTION=NORMAL は ENCRYPTION=MEDIUM と等価として受け入れられます。

STRONG

CICS で以下の暗号スイートのみを使用することを指定します。

暗号スイート	暗号化 アルゴリズム	キーの長さ	MAC アルゴリズム
01	暗号化なし		MD5
02	暗号化なし		SHA
03	RC4	40 ビット	MD5
04	RC4	128 ビット	MD5
05	RC4	128 ビット	SHA
06	RC2	40 ビット	MD5
09	DES	56 ビット	SHA
0A	Triple-DES	168 ビット	SHA
2F	AES	128 ビット	SHA
35	AES	256 ビット	SHA

この表で使用される用語は、以下のとおりです。

MD5 メッセージ・ダイジェスト・アルゴリズム

SHA セキュア・ハッシュ・アルゴリズム

RC2, RC4
Rivest 暗号化

DES データ暗号化基準

Triple-DES
DES 処理を 3 回行う

AES 拡張暗号化規格

WEAK

CICS で以下の暗号スイートのみを使用することを指定します。

暗号スイート	暗号化 アルゴリズム	キーの長さ	MAC アルゴリズム
01	暗号化なし		MD5
02	暗号化なし		SHA
03	RC4	40 ビット	MD5
06	RC2	40 ビット	MD5

この表で使用される用語は、以下のとおりです。

MD5 メッセージ・ダイジェスト・アルゴリズム
SHA セキュア・ハッシュ・アルゴリズム
RC2, RC4
Rivest 暗号化

MEDIUM

CICS で以下の暗号スイートのみを使用することを指定します。

暗号スイート	暗号化 アルゴリズム	キーの長さ	MAC アルゴリズム
01	暗号化なし		MD5
02	暗号化なし		SHA
03	RC4	40 ビット	MD5
06	RC2	40 ビット	MD5
09	DES	56 ビット	SHA

この表で使用される用語は、以下のとおりです。

MD5 メッセージ・ダイジェスト・アルゴリズム
SHA セキュア・ハッシュ・アルゴリズム
RC2, RC4
Rivest 暗号化
DES データ暗号化基準

パラメーター SSLTCBS は推奨されません。代わりに、以下の新しいパラメーターを使用してください。

MAXSSLTCBS={8|number}

SSL プールで実行できる S8 TCB の最大数を指定します。デフォルトは 8 ですが、最大 1024 個の TCB を指定できます。

新しいシステム初期設定パラメーターは、次のとおりです。

CRLSERVER=servername:port

証明書取り消しリスト (CRL) が置かれている LDAP サーバーの名前およびポート番号を指定します。このパラメーターを指定すると、CICS は、SSL ネゴシ

#

エーション時に各クライアント証明書について取り消し状況を検査します。証明書
書が取り消されている場合、CICS は直ちに接続を閉じます。

SSLCACHE={CICS|SISYSPLEX}

SSL でセッション ID のローカルまたはシस्पレックスのキャッシングを使用するかどうかを指定します。シस्पレックス・キャッシングは、複数の CICS ソケット専有領域が同じ IP アドレスで複数の SSL 接続を受け入れる場合に役立ちます。

リソース定義の変更点

TCPIPSERVICE および CORBASERVER 定義

TCPIPSERVICE および CORBASERVER リソース定義に新しい属性 CIPHERS が追加されました。この属性は、新しい URIMAP リソース定義に指定することもできます。詳細については、134 ページの『URIMAP リソース定義』を参照してください。暗号スイート・コードの CIPHERS リストは、リソースのために確立されたソケット接続で SSL または TLS セキュリティー・プロトコルを使用する場合にのみ使用されます。TCPIPSERVICE 定義の場合、CIPHERS リストはインバウンド・ソケット接続に使用されます。CORBASERVER 定義の場合、CIPHERS リストはアウトバウンド・ソケット接続に使用されます。

CIPHERS=value

この値は、28 個以内の 2 桁の暗号スイート・コードのリストとして解釈される 56 個以内の 16 進数字のストリングを指定します。受け入れ可能コードのリストは、ENCRYPTION システム初期設定パラメーターに依存します。

- ENCRYPTION=WEAK の場合、デフォルト値は 03060102 です。
- ENCRYPTION=MEDIUM の場合、デフォルト値は 0903060102 です。
- ENCRYPTION=STRONG の場合、デフォルト値は 0504352F0A0903060102 です。

暗号コードを再配列したり、デフォルト・リストから除去することができます。ただし、特定の暗号化レベルのデフォルト・リストにない暗号コードを新たに追加することはできません。ENCRYPTION システム初期設定パラメーターは、それぞれの暗号化レベルごとに許可される暗号スイート・コードを決定します。

TCPIPSERVICE リソース定義の PRIVACY 属性は、CIPHERS 属性値によって決定します。CIPHERS 属性のデフォルト値は暗号スイートの完全なリストなので、一部の暗号コードを除去すると、PRIVACY 属性が変更されます。

- CICS が暗号化処理を備えたクライアントとのみネゴシエーションするように指定するために暗号スイート 01 および 02 を除去した場合、PRIVACY 属性値は REQUIRED に変更されます。
- CICS が暗号化処理を備えていないクライアントとのみネゴシエーションするように指定するために暗号スイート 01 および 02 以外のすべての暗号スイートを除去した場合、PRIVACY 属性は NOTSUPPORTED に変更されます。
- その他の組み合わせで暗号スイート (デフォルトを含む) を指定した場合、PRIVACY 属性の値は SUPPORTED です。

CORBASERVER リソース定義の OUTPRIVACY 属性にも同様の制約が適用されます。

システム・プログラミング・インターフェースの変更点

EXEC CICS コマンド INQUIRE TCPIPSERVICE、INQUIRE CORBASERVER および新しいコマンド INQUIRE URIMAP に、2 つのセキュリティー・オプションが組み込まれました。

CIPHERS(*char56*)

リソース定義 TCPIPSERVICE、CORBASERVER、および URIMAP の属性
CIPHERS に指定されている暗号スイートのリストを戻します。この暗号スイ
トのリストは、SSL 接続のネゴシエーションで使用されます。例えば、弱い暗
号化を使用している場合、デフォルト値は 03060102 です。

NUMCIPHERS(*halfword*)

SSL ハンドシェイクの一部としての暗号化レベルのネゴシエーションに使用される暗号スイートの数を戻します。

EXEC CICS コマンド INQUIRE TCPIP にも、2 つのセキュリティー・オプション
があります。

CRLSERVER(*char255*)

CRLSERVER システム初期設定パラメーターに指定された LDAP サーバーの名前を戻します。LDAP サーバーは、証明書取り消しリストの保管に使用されます。

SSLCACHE(*cvda*)

セッション ID を保管するために SSL で使用されているキャッシュを示す CVDA 値を戻します。CVDA 値は、以下のとおりです。

CICS

SSL は CICS 領域のためのローカル SSL キャッシュを使用しています。

SYSPLEX

SSL はカップリング・ファシリティーの SSL キャッシュを使用しています。

SSL TCB プールを処理するために、INQUIRE DISPATCHER および SET DISPATCHER コマンドに対して変更が行われました。以下の 2 つのオプションが追加されました。

MAXSSLTCBS(*fullword*)

MAXSSLTCBS システム初期設定パラメーターに指定されている、SSL プールで可能な S8 TCB の最大数を戻します。

ACTSSLTCBS(*fullword*)

SSL プール内の S8 TCB の実際の数を戻します。

CICS 提供トランザクションの変更点

CCRL トランザクション

LDAP サーバーに保管されている証明書取り消しリスト (CRL) を作成および更新するには、CCRL トランザクションを使用します。CCRL を使用する必要があるのは、CICS 領域に SSL をインプリメントする際に、各接続に関して取り消された証明書がないかを SSL ハンドシェイク中に検査するようにしたい場合だけです。

CCRL トランザクションは、ワールド・ワイド・ウェブでの CRL リポジトリのロケーションを指定します。CICS は、指定された URL にある CRL リポジトリからリストをダウンロードして、それを LDAP サーバーに保管します。複数の CRL リポジトリにアクセスする必要がある場合は、複数の URL を指定できます。

CCRL トランザクションを実行する前に、CICS で以下のようにセットアップしておく必要があります。

- LDAP サーバーはセットアップ済みで、証明書取り消しリストを保管できるように構成されている。
- CRLSERVER システム初期設定パラメーターには、証明書取り消しリストを保管する LDAP サーバーの名前が定義されている。

CCRL トランザクションは、端末から、または START コマンドから実行できます。定期的な更新をスケジュールしたい場合は、START コマンド・オプションを使用してください。

トランザクションを端末から実行する場合は、ご使用の端末で大/小文字混合文字が受け入れられることを確認してください。コマンド `CCRL url-list` を入力します。ここで `url-list` は、ダウンロードする証明書取り消しリストが格納される URL をスペースで区切ったリストです。次に、LDAP サーバーの管理者識別名およびパスワードを入力するよう求められます。これにより CICS は、LDAP サーバーの CRL を、ダウンロードされた CRL に更新できます。

START コマンドからトランザクションを実行するには、以下の構文を使用します。

```
EXEC CICS START TRANSID(CCRL)
  FROM('admin://adminDN:adminPW url-list')
  LENGTH(url-list-length)
  [INTERVAL(hhmmss)|TIME(hhmmss)]
```

ここで、`adminDN:adminPW` は LDAP サーバーの識別名およびパスワードで、`url-list` は、ダウンロードする証明書取り消しリストが格納される URL をスペースで区切ったリストです。また、`url-list-length` は LDAP admin の識別名およびパスワードが含まれる URL リストの長さを表し、`hhmmss` は CCRL トランザクションの実行がスケジュールされている間隔または有効期限時刻を表します。

例えば、次のように指定できます。

```
EXEC CICS START TRANSID(CCRL)
  FROM('admin://cn=ldapadmin:cics31ldap http://cr1.CertificateAuthority.com/CRLList1.crl
  http://cr1.CertificateAuthority.com/CRLList2.crl')
  LENGTH(132) INTERVAL(960000)
```

このコマンドは、96 時間後に CCRL トランザクションを実行して、指定の 2 つの URL から証明書取り消しリストをダウンロードするようスケジュールします。

CEMT の変更点

以下のコマンドに対して変更が行われました。

- INQUIRE TCPIP SERVICE
- INQUIRE CORBASERVER
- INQUIRE TCPIP

- INQUIRE DISPATCHER
- SET DISPATCHER

「CICS Supplied Transactions」に、これらの変更されたコマンドについての情報が記載されています。

グローバル・ユーザー出口ルーチンの変更点

グローバル・ユーザー出口ルーチン TCB 標識リストは、SP TCB モードのための項目を含むようになりました。完全なリストについては、「Customization Guide」を参照してください。

モニターの変更点

MAXSSLTCBS 遅延時間およびモード変更遅延時間をモニターするための新規モニター・フィールドが追加されました。

DFHTASK

247 (TYPE-S, 'DSCHMDLY', 8 BYTES)

ユーザー・タスクによって、またはユーザー・タスクに代わって発行された CICS ディスパッチャー TCB モード変更要求の後に、ユーザー・タスクが再ディスパッチを待機する経過時間。例えば、CICS L8 または S8 モード TCB から CICS QR モード TCB に戻す TCB モード変更要求は QR TCB 上で別のタスクが現在ディスパッチされているために、QR TCB を待機しなければならない場合があります。

注: このフィールドは、タスク一時停止時間 SUSPTIME (014) フィールドのコンポーネントです。

281 (TYPE-S, 'MAXSTDLY', 8 BYTES)

CICS システムがシステム初期設定パラメーター MAXSSLTCBS によって設定された限界に達したため、CICS SSL TCB (S8 モード) を取得するためにユーザー・タスクが待機した経過時間。S8 モードのオープン TCB は、SSL pthread によって独占的に使用されます。

注: このフィールドは、タスク一時停止時間 SUSPTIME (014) フィールドのコンポーネントです。

統計の変更点

統計に対する変更点には、新しい SP TCB モードおよび新しい SSL TCB プールに関するレポートが含まれます。統計ユーティリティー・プログラム DFHSTUP と、統計サンプル・プログラム DFHOSTAT および DFHOSTXR にも変更点があり、これによって、追加の TCB モードおよびプールが報告されます。TCPIP グローバル統計では、SSL キャッシングがローカルで実行されるかシスプレックス全体で実行されるかについても報告されます。

CICS ユーティリティーの変更点

追加の TCB モードおよびプールの両方を報告するため、統計ユーティリティー・プログラム DFHSTUP に対して変更が行われました。

問題判別の変更点

メッセージ DFHS00123 には、暗号化ネゴシエーション時に CICS が取り消し済み証明書を識別するとき使用される新しい挿入があります。CICS は、接続を即時にクローズし、このメッセージを発行します。

セキュリティ

新しい AES 暗号およびより長い暗号鍵長のサポートにより、セキュリティが拡張されました。SSL 接続で使用される暗号スイートのデフォルト値を編集することにより、クライアントと CICS の間のネゴシエーションのための最小暗号化レベルを制限することもできます。

ENCRYPTION システム初期設定パラメーターのデフォルト値が **STRONG** になり、SSL 使用時の CICS でのセキュリティのレベルが高まりました。

CCRL トランザクションのためのセキュリティ・カテゴリーは、カテゴリー 2 です。

マイグレーション

既存の機能のマイグレーション

ENCRYPTION システム初期設定パラメーターのデフォルトの設定値が **STRONG** に変更されました。高いレベルの暗号を解く鍵 (セキュリティ・レベル 3) が z/OS にインストールされていない場合は、ENCRYPTION システム初期設定パラメーターのデフォルトの設定値をダウングレードする必要があります。前のリリースでデフォルトとして使用されていた **NORMAL** 設定は、このリリースの CICS では **MEDIUM** に変更されました。マイグレーションの目的のために、**NORMAL** は **MEDIUM** の代替として受け入れられます。

SSLTCBS システム初期設定パラメーターは使用されなくなり、MAXSSLTCBS に置き換えられました。MAXSSLTCBS は、SSL のオープン・トランザクション環境 (OTE) TCB プールで並行して実行可能な S8 TCB の最大数を制御します。

新規機能へのマイグレーション

CIPHERS 属性を活用することで、CICS とクライアントの間の暗号化ネゴシエーション・プロセスの制御が改善されます。CRLSERVER および SSLCACHE システム初期設定パラメーターを使用して、SSL ハンドシェイクで証明書を検査したり、CICS 領域間の SSL キャッシュを共有することによりハンドシェイクのパフォーマンスを改善できます。

共存

CICS は、新しい TLS 1.0 プロトコルと同様に SSL v3.0 を引き続きサポートします。

CICSplex SM サポート

新しい SSL 拡張のサポートに対応するために、CICSplex SM インターフェースおよび API に対していくつかの変更が行われました。

CICSplex SM エンド・ユーザー・インターフェース・ビューの変更点

CICS Transaction Server for z/OS バージョン 3 リリース 1 には、新しく追加されたエンド・ユーザー・インターフェース・ビューはありませんが、以下のビューに対して、SSL の新規サポートに一致するように変更が加えられました。

- 『EJCODEF ビュー』
- 『TCPDEF ビュー』
- 『EJCOSE ビュー』
- 192 ページの 『TCPIPS ビュー』

EJCODEF ビュー

EJCODEF ビューに、以下の属性が追加されました。

CIPHERS

SSL 暗号スイート・コード

このビューの **OUTPRIVACY** 属性は、CICS Transaction Server 3.1 では無効になったため、表示はされますが、この属性は無視されます。

TCPDEF ビュー

TCPDEF ビューに、以下の属性が追加されました。

CIPHERS

SSL 暗号スイート・コード

このビューの **PRIVACY** 属性は、CICS Transaction Server 3.1 では無効になったため、表示はされますが、この属性は無視されます。

EJCOSE ビュー

EJCOSE ビューに、以下の属性が追加されました。

CIPHERS

SSL 暗号スイート・コード

NUMCIPHERS

暗号化レベルのネゴシエーションに使用される暗号スイートの数。

TCPIPS ビュー

TCPIPS ビューに、以下の属性が追加されました。

CIPHERS

SSL 暗号スイート・コード

NUMCIPHERS

暗号化レベルのネゴシエーションに使用される暗号スイートの数。

CICSplex SM アプリケーション・プログラミング・インターフェースの変更点

以下のリソース・テーブルに対して変更が加えられました。

- 『EJCODEF リソース・テーブル』
- 『TCPDEF リソース・テーブル』
- 『EJCOSE リソース・テーブル』
- 『TCPIPS リソース・テーブル』
- 193 ページの 『TCPIPGBL リソース・テーブル』
- 193 ページの 『TASK リソース・テーブル』
- 193 ページの 『CICSRGN リソース・テーブル』

EJCODEF リソース・テーブル

EJCODEF リソース・テーブルに、以下の新規属性が追加されました。

CIPHERS

SSL 暗号スイート・コードを示す 56 文字のリストを表します。

TCPDEF リソース・テーブル

TCPDEF リソース・テーブルに、以下の新規属性が追加されました。

CIPHERS

SSL 暗号スイート・コードを示す 56 文字のリストを表します。

EJCOSE リソース・テーブル

EJCOSE リソース・テーブルに、以下の新規属性が追加されました。

CIPHERS

SSL 暗号スイート・コードを示す 56 文字のリストを表します。

NUMCIPHERS

暗号化レベルのネゴシエーションに使用される暗号スイートの数。

TCPIPS リソース・テーブル

TCPIPS リソース・テーブルに、以下の新規属性が追加されました。

CIPHERS

SSL 暗号スイート・コードを示す 56 文字のリストを表します。

NUMCIPHERS

暗号化レベルのネゴシエーションに使用される暗号スイートの数。

TCPIPGBL リソース・テーブル

TCPIPGBL リソース・テーブルに、以下の新規属性が追加されました。

CRLSERVER

証明書取り消しリスト (CRL) が格納される、56 文字の LDAP サーバー名を表します。

SSLCACHE

SSL キャッシュ・タイプ (ローカルまたはシスプレックス)

TASK リソース・テーブル

TASK リソース・テーブルに、以下の新規属性が追加されました。

TMRSTDLY

ディスプレイパッチャーの変更 TCB モードの要求が発行されてから再ディスプレイパッチャーするまでタスクが待機した時間 (8 バイト) を表します。

TMRCMDLY

MAXSSLTCBS によって設定された限度に達したため、タスクが CICS SSL TCB を取得するために待機した時間 (8 バイト) を表します。

CICSRGN リソース・テーブル

CICSRGN リソース・テーブルには、以下の SPI 属性が含まれています。

MAXSSLTCBS

SSL OTE プール内の TCB の現在の最大数を表します。

ACTSSLTCBS

現在割り振られている SSL プール TCB の数を表します。

CICSplex SM Web ユーザー・インターフェースの変更点

以下のビューに対して変更が加えられました。

- 『CorbaServer 定義ビュー』
- 194 ページの『TCP/IP サービス定義ビュー』
- 194 ページの『CorbaServer ビュー』
- 194 ページの『TCP/IP サービス・ビュー』
- 194 ページの『TCP/IP グローバル状況ビュー』
- 194 ページの『クロックとタイミング・ビュー』
- 195 ページの『CICS 領域ビュー』

CorbaServer 定義ビュー

EJCODEF (EYUSTARTEJCODEF) ビューに、以下の属性が追加されました。

CIPHERS

SSL 暗号スイート・コード

このビューの **OUTPRIVACY** 属性は、CICS TS 3.1 では無効になったため、表示はされますが、この属性は無視されます。

TCP/IP サービス定義ビュー

TCPDEF (EYUSTARTTCPDEF) ビューに、以下の属性が追加されました。

CIPHERS

SSL 暗号スイート・コード

このビューの **PRIVACY** 属性は、CICS TS 3.1 では無効になったため、表示はされますが、この属性は無視されます。

CorbaServer ビュー

EJCOSE (EYUSTARTEJCOSE.DETAILED) ビューに、以下の属性が追加されました。

CIPHERS

SSL 暗号スイート・コード

NUMCIPHERS

暗号化レベルのネゴシエーションに使用される暗号スイートの数

TCP/IP サービス・ビュー

TCPIPS (EYUSTARTTCPIPS.DETAILED) ビューに、以下の属性が追加されました。

CIPHERS

SSL 暗号スイート・コード

NUMCIPHERS

暗号化レベルのネゴシエーションに使用される暗号スイートの数

TCP/IP グローバル状況ビュー

TCPIPGBL (EYUSTARTTCPIPGBL.DETAILED) ビューに、以下の属性が追加されました。

CRLSERVER

証明書取り消しリスト (CRL) が格納される LDAP サーバー名

SSLCACHE

SSL ではセッション ID のキャッシングをローカルで使用するか、システム全体にわたって使用するか

クロックとタイミング・ビュー

タスク操作ビュー内部のアクティブ・タスク・ビュー・セット (EYUSTARTTASK.DETAIL2) の 1 つであるクロックとタイミング・ビューに、以下の属性が追加されました。

TMRCMDLY

CICS TCB 変更モードの遅延時間

TMRSTDLY

CICS SSL TCB 最大遅延時間

CICS 領域ビュー

CICSRGN (EYUSTARTCICSRGN.DETAILED) ビューに以下の統計属性が追加されました。

MAXSSLTCBS

SSL OTE プール内の TCB の現在の最大数を表します。

ACTSSLTCBS

現在割り振られている SSL プール TCB の数を表します。

第 3 部 アプリケーション変換

CICS Transaction Server for z/OS バージョン 3 リリース 1 では、最新のプログラム言語、構成、ツールを使用した、既存のアプリケーションの拡張、および新規アプリケーションの作成を可能にする一連の新機能を提供します。

第 7 章 プログラム間データ転送の拡張: 最新の COMMAREA と してのチャンネル

従来、CICS プログラムでは、データ交換のために通信域 (COMMAREA) を使用していました。CICS Transaction Server for z/OS バージョン 3 リリース 1 では、チャンネルおよびコンテナを使用することによってプログラム間のデータ転送方法が改善されました。これにより、COMMAREA に適用される 32KB の制限をはるかに超える量で転送できます。

- コンテナとは、プログラム間で情報を受け渡すために作られた名前付きデータ・ブロックです。
- コンテナは、チャンネルと呼ばれるセットとしてグループ化されます。チャンネルは、CICS プログラム間でデータを交換するための標準のメカニズムであり、パラメーター・リストと類似しています。チャンネルは、LINK、START、XCTL、RETURN コマンド上、およびローカル・トランザクションとリモート・トランザクションで使用することができます。チャンネルに追加できるコンテナの数に制限はなく、各コンテナのサイズは、利用可能なストレージの量によってのみ制限されます。

チャンネルの利点

チャンネル/コンテナ・モデルには、データ交換のために CICS プログラムで従来使用されていた通信域 (COMMAREA) と比べて、いくつかの利点があります。以下に例を示します。

- COMMAREA とは異なり、チャンネルのサイズは 32KB に制限されません。チャンネルに追加できるコンテナの数に制限はなく、個々のコンテナのサイズは、利用可能なストレージの量によってのみ制限されます。
- 1 つのチャンネルは複数のコンテナを含むことができるため、より構造化された方法でデータを渡すことができます。これに対して、COMMAREA は一体構造のデータ・ブロックです。
- COMMAREA とは異なり、チャンネルを使用するプログラムは、戻されるデータの正確なサイズを知っている必要はありません。
- チャンネルは、CICS プログラム間でデータを交換するための標準のメカニズムです。チャンネルは、LINK、START、XCTL、および RETURN コマンドで渡すことができます。分散プログラム・リンク (DPL) がサポートされており、START CHANNEL コマンドおよび RETURN TRANSID コマンドによって開始されるトランザクションはリモート・トランザクションであってもかまいません。

注: リモート・プログラムまたはリモート・トランザクションにチャンネルを渡す場合、大量のデータを渡すとパフォーマンスに影響する可能性があります。これは、ローカル領域とリモート領域が MRO ではなく ISC 接続で接続されている場合に特に問題になります。

- チャンネルは、CICS でサポートされる任意の言語で作成された CICS アプリケーション・プログラムで使用できます。例えば、ある CICS 領域上の Java クライアント・プログラムは、チャンネルを使用して、バックエンドの AOR 上の COBOL サーバー・プログラムとデータを交換することができます。

- 複数のチャンネルを処理するサーバー・プログラムを作成できます。これは、例えば、以下を行うことができます。
 1. 呼び出し時に指定されたチャンネルを動的に発見する
 2. チャンネル内のコンテナをブラウズする
 3. 渡されたチャンネルに応じて処理を変更する
- 1 つ以上のチャンネルを介して呼び出された関連するプログラム・セットから「コンポーネント」を構築できます。
- クライアントとコンポーネント間の疎結合により、展開が容易になります。クライアントおよびコンポーネントは、別々の時期にアップグレードできます。例えば、最初に、新規チャンネルを処理するためにコンポーネントをアップグレードし、その後で、この新規チャンネルを使用するためにクライアント・プログラムをアップグレードできます (あるいは新規クライアントを作成できます)。
- プログラマーは、ストレージ管理の問題から解放されます。CICS は、コンテナがスコープ外になると、そのコンテナ (およびそのストレージ) を自動的に破棄します。
- チャンネル・アプリケーションが使用するデータ変換モデルは、COMMAREA アプリケーションが使用するものと比べて極めて単純です。また、COMMAREA アプリケーションのデータ変換はシステム・プログラマーによって制御されるのに対して、チャンネル・アプリケーションの場合は、単純な API コマンドを使用してアプリケーション・プログラマーによってデータ変換が制御されます。
- CICS ビジネス・トランザクション・サービス (BTS) の経験を持つプログラマーにとって、非 BTS アプリケーションにおけるコンテナの使用は容易です。
- コンテナを使用するプログラムは、チャンネル・アプリケーションおよび BTS アプリケーションの両方から呼び出すことができます。
- コンテナを使用する非 BTS アプリケーションは、完全な BTS アプリケーションにマイグレーションすることができます。(これらのアプリケーションは BTS へのマイグレーション経路を形成します。)

用語

チャンネル

プログラム間でのデータの受け渡しのために使用される**コンテナ**のグループ。チャンネルは、パラメーター・リストに類似しています。

コンテナ

プログラム間での情報の受け渡しのために使用される名前付きデータ・ブロック。これは、「名前付き通信域 (COMMAREA)」と考えることができます。プログラム間では、任意の数のコンテナを渡すことができます。コンテナは、**チャンネル**と呼ばれるセットとしてグループ化されます。

現行チャンネル

これが存在する場合、プログラムの呼び出しに使用されるチャンネル。特定プログラムの特定の呼び出し用の現行チャンネルは変更されません。

チャンネル: クイック・スタート

コンテナーおよびチャンネル

コンテナーは、プログラム間での情報の引き渡しのために設計された名前付きデータ・ブロックです。これは、「名前付き通信域 (COMMAREA)」と考えることができます。プログラム間で、任意の数のコンテナーを受け渡すことができます。コンテナーのサイズは、利用可能なストレージの量によってのみ制限されます。

コンテナーは、**チャンネル**と呼ばれるセットとしてグループ化されます。チャンネルは、パラメーター・リストに類似しています。

名前付きコンテナーを作成して、これをチャンネルに割り当てるために、プログラムでは、EXEC CICS PUT CONTAINER(*container-name*) CHANNEL(*channel-name*) コマンドを使用します。これにより、EXEC CICS LINK、XCTL、START、または RETURN コマンドの CHANNEL(*channel-name*) オプションを使用してチャンネル (およびそのコンテナー) を 2 番目のプログラムに渡すことができます。

2 番目のプログラムは、EXEC CICS GET CONTAINER(*container-name*) コマンドを使用して、渡されたコンテナーを読み取ることができます。このコマンドは、プログラムの呼び出し時に指定されたチャンネルに属する名前付きコンテナーを読み取ります。

2 番目のプログラムが LINK コマンドを使用して呼び出された場合、このプログラムは、呼び出し側プログラムにコンテナーを戻すこともできます。これを行うには、新規コンテナーを作成するか、既存のコンテナーを再利用します。

チャンネルとコンテナーは、これらを作成したプログラムおよびこれらを受け渡されたプログラムからのみ可視です。これらのプログラムが終了すると、CICS では、コンテナーおよびそのストレージを自動的に破棄します。

チャンネルのコンテナーは、リカバリー可能ではありません。リカバリー可能なコンテナーを使用する必要がある場合は、CICS ビジネス・トランザクション・サービス (BTS) コンテナーを使用してください。チャンネルと BTS コンテナーの関係は、221 ページの『チャンネルと BTS アクティビティ』で説明しています。

基本的な例

203 ページの図 16 は、以下を行う COBOL プログラム CLIENT1 を示しています。

1. PUT CONTAINER(*container-name*) CHANNEL(*channel-name*) コマンドを使用して、inqcustrec という名前のチャンネルを作成して 2 つのコンテナー custno および branchno をこれに追加します。これらには、それぞれ顧客番号と支店番号が含まれています。
2. LINK PROGRAM(*program-name*) CHANNEL(*channel-name*) コマンドを使用してプログラム SERVER1 にリンクし、inqcustrec チャンネルを渡します。
3. GET CONTAINER(*container-name*) CHANNEL(*channel-name*) コマンドを使用して、SERVER1 から戻される顧客レコードを取得します。顧客レコードは、inqcustrec チャンネルの custrec コンテナーに入っています。

クライアント・プログラムとサーバー・プログラムでは、同じ COBOL コピーブック INQINTC を使用していることに注意してください。コピーブックの行 3 および行 5 から 7 は、INQUIRY-CHANNEL およびそのコンテナを表しています。これらの行は、プログラムの処理で必ずしも必要ではありません。なぜなら、コンテナおよびチャネルは (例えば、PUT CONTAINER コマンドで) 単に指定するだけで作成されるからです。定義する必要はありません。ただし、両方のプログラムが使用するコピーブックにこれらの行を含めることにより、保守が容易になります。これらは、使用されたコンテナの名前を記録します。

推奨

チャネルを使用するクライアント/サーバー・アプリケーションの保守を容易にするために、使用するコンテナの名前を記録してコンテナにマップするデータ・フィールドを定義するコピーブックを作成してください。コピーブックはクライアント・プログラムとサーバー・プログラムの両方に組み込んでください。

注: この例では、2 つの COBOL プログラムを示します。CICS でサポートされる他の任意の言語で同じ手法を使用できます。ただし、COBOL プログラムの場合のみ、サーバー・プログラムが EXEC CICS GET CONTAINER コマンドで (INTO の代わりに) SET オプションを使用する場合は、SET で指し示されるストレージの構造をプログラムの LINKAGE セクションに定義する必要があります。これは、1 つではなく、2 つのコピーブックが必要になることを意味します。最初のコピーブックは、プログラムの WORKING-STORAGE セクションで、使用するチャネルおよびコンテナを指定します。2 番目のコピーブックは、LINKAGE セクションで、ストレージ構造を定義します。

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. CLIENT1.
```

```
WORKING-STORAGE SECTION.
```

```
    COPY INQINTC  
*           copybook INQINTC  
* Channel name  
* 01 INQUIRY-CHANNEL PIC X(16) VALUE 'inqcustrec'.  
* Container names  
* 01 CUSTOMER-NO      PIC X(16) VALUE 'custno'.  
* 01 BRANCH-NO        PIC X(16) VALUE 'branchno'.  
* 01 CUSTOMER-RECORD  PIC X(16) VALUE 'custrec'.  
* Define the data fields used by the program  
* 01 CUSTNO           PIC X(8).  
* 01 BRANCHNO        PIC X(5).  
* 01 CREC.  
* 02 CUSTNAME         PIC X(80).  
* 02 CUSTADDR1        PIC X(80).  
* 02 CUSTADDR2        PIC X(80).  
* 02 CUSTADDR3        PIC X(80).
```

```
PROCEDURE DIVISION.  
MAIN-PROCESSING SECTION.
```

```
*  
* INITIALISE CUSTOMER RECORD  
*  
*     ... CREATE CUSTNO and BRANCHNO  
*  
* GET CUSTOMER RECORD  
*  
*     EXEC CICS PUT CONTAINER(CUSTOMER-NO) CHANNEL(INQUIRY-CHANNEL)  
*                   FROM(CUSTNO) FLENGTH(LENGTH OF CUSTNO)  
*                   END-EXEC  
*     EXEC CICS PUT CONTAINER(BRANCH-NO) CHANNEL(INQUIRY-CHANNEL)  
*                   FROM(BRANCHNO) FLENGTH(LENGTH OF BRANCHNO)  
*                   END-EXEC  
*  
*     EXEC CICS LINK PROGRAM('SERVER1') CHANNEL(INQUIRY-CHANNEL) END-EXEC  
*  
*     EXEC CICS GET CONTAINER(CUSTOMER-RECORD) CHANNEL(INQUIRY-CHANNEL)  
*                   INTO(CREC) END-EXEC  
*  
*  
* PROCESS CUSTOMER RECORD  
*  
*     ... FURTHER PROCESSING USING CUSTNAME and CUSTADDR1 etc...  
*  
*     EXEC CICS RETURN END-EXEC  
*  
* EXIT.
```

図 16. チャンネルを作成してこれを 2 番目のプログラムに渡すプログラムの簡単な例

204 ページの図 17 は、CLIENT1 によってリンクされている SERVER1 プログラムを示しています。SERVER1 は、渡された custno および branchno コンテナからデータを取得し、これを使用してデータベース内で完全な顧客レコードを見つけます。次に、同じチャンネルに新規コンテナ custrec を作成し、これに顧客レコードを入れて戻します。

プログラマーは、SERVER1 の GET および PUT コマンドで CHANNEL キーワードを指定していないことに注意してください。チャンネルを明示的に指定しなかった場合

は、現行チャンネル、すなわち、プログラムの呼び出し時に指定されたチャンネルが使用されます。

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. SERVER1.
```

```
WORKING-STORAGE SECTION.
```

```
    COPY INQINTC  
*      copybook INQINTC  
* Channel name  
* 01 INQUIRY-CHANNEL PIC X(16) VALUE 'inqcustrec'.  
* Container names  
* 01 CUSTOMER-NO      PIC X(16) VALUE 'custno'.  
* 01 BRANCH-NO        PIC X(16) VALUE 'branchno'.  
* 01 CUSTOMER-RECORD PIC X(16) VALUE 'custrec'.  
* Define the data fields used by the program  
* 01 CUSTNO           PIC X(8).  
* 01 BRANCHNO        PIC X(5).  
* 01 CREC.  
*   02 CUSTNAME      PIC X(80).  
*   02 CUSTADDR1     PIC X(80).  
*   02 CUSTADDR2     PIC X(80).  
*   02 CUSTADDR3     PIC X(80).
```

```
PROCEDURE DIVISION.  
MAIN-PROCESSING SECTION.
```

```
    EXEC CICS GET CONTAINER(CUSTOMER-NO)  
                INTO(CUSTNO) END-EXEC  
    EXEC CICS GET CONTAINER(BRANCH-NO)  
                INTO(BRANCHNO) END-EXEC
```

```
    ... USE CUSTNO AND BRANCHNO TO FIND CREC IN A DATABASE
```

```
    EXEC CICS PUT CONTAINER(CUSTOMER-RECORD)  
                FROM(CREC)  
                FLENGTH(LENGTH OF CREC) END-EXEC
```

```
    EXEC CICS RETURN END-EXEC
```

```
    EXIT.
```

図 17. 渡されたチャンネルからデータを取得するリンク・プログラムの簡単な例： このプログラムは、203 ページの図 16 に示したプログラム *CLIENT1* によってリンクされています。

チャンネルの使用: いくつかの典型的シナリオ

チャンネルおよびコンテナは、プログラム間でデータの受け渡しを行うための強力な手段を提供します。このセクションでは、チャンネルの使用方法についてのいくつかの例を示します。

1 つのチャンネルと 1 つのプログラム

205 ページの図 18 に、スタンドアロン・プログラムが 1 つのチャンネルを持ち、このチャンネルでこのプログラムを呼び出すことができるという最も単純なシナリオを示します。

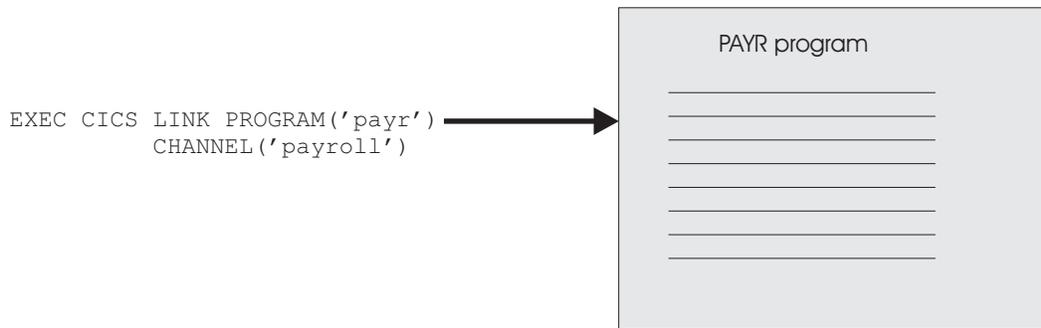


図 18. 単一チャンネルを持つスタンドアロン・プログラム

1 つのチャンネルと複数のプログラム (コンポーネント)

図 19 では、相互に関連する一連のプログラム内に最上位プログラムへの単一チャンネルがあります。陰影付きの領域内の一連のプログラムは、「コンポーネント」と見なすことができます。クライアント・プログラムは、外部チャンネルのみを「見る」のであり、行われる処理やバックエンド・プログラムの存在を認識しません。

コンポーネント内部では、プログラムは互いにチャンネルを渡すことができます。代替として、コンポーネント・プログラムは、例えば、新規チャンネルを作成し、元のチャンネルから 1 つ以上のコンテナを追加することにより、元のチャンネルのサブセットを渡すことができます。

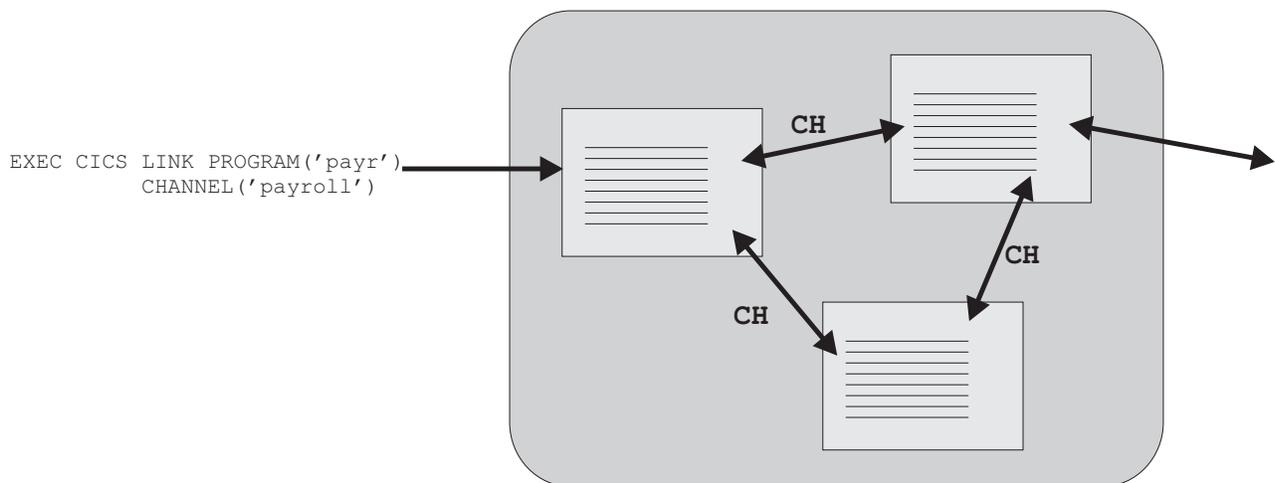


図 19. 「コンポーネント」 - 単一の外部チャンネルを介して呼び出される一連の関連プログラム : 「CH」はコンポーネント内のプログラムが相互にチャンネルを渡すことができることを示す。

複数のチャンネルと 1 つのコンポーネント

図 19 と同様に、1 つのコンポーネントと見なすことができる相互に関連した一連のプログラムがあります。ただし、今回は、コンポーネントが呼び出すことのできる 2 つの代替の外部チャンネルが存在します。コンポーネント内の最上位のプログラム

は、EXEC CICS ASSIGN CHANNEL コマンドを発行し、このコマンドの呼び出しに使用されたチャンネルを判別し、それに応じてその処理を調整します。

クライアント・プログラムとコンポーネント間の「疎結合」により、展開が容易になります。すなわち、クライアントとコンポーネントは、別々の時期にアップグレードすることができます。例えば、第 1 のチャンネルとも第 2 のチャンネルとも異なるコンテナのセットで構成される第 3 のチャンネルを処理するためにプログラムをアップグレードすることができます。次に、クライアント・プログラムをアップグレードして (あるいは新規クライアントを作成して)、第 3 のチャンネルを渡すことができます。

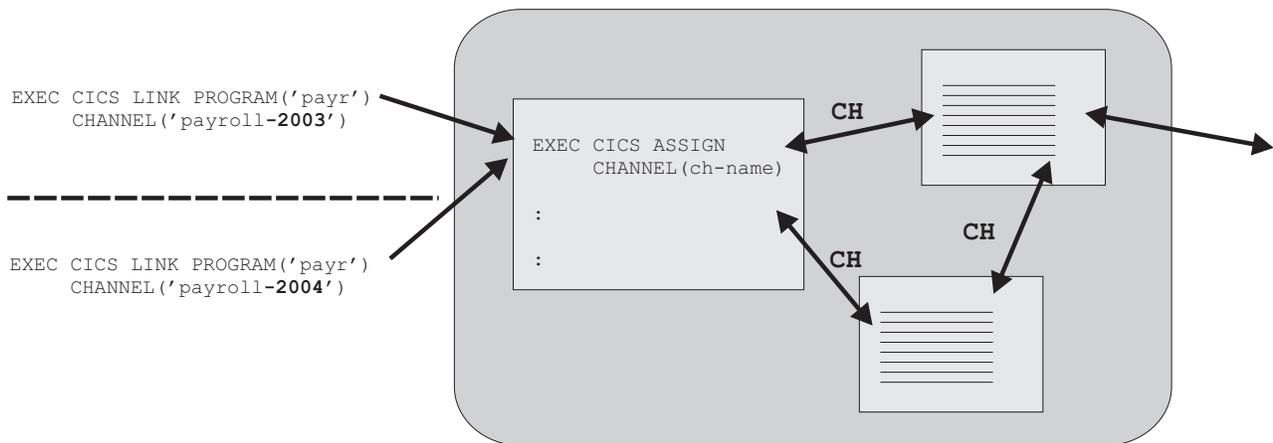


図 20. 同じコンポーネントに対する複数の外部チャンネル：「CH」はコンポーネント内のプログラムが相互にチャンネルを渡すことができることを示す。

複数の対話式コンポーネント

207 ページの図 21 は、「Human resources」コンポーネントおよび「Payroll」コンポーネントを示しています。各コンポーネントはチャンネルを持っており、これを使って呼び出すことができます。Payroll コンポーネントは、スタンドアロン・プログラムおよび Human resources コンポーネントの両方から呼び出すことができます。

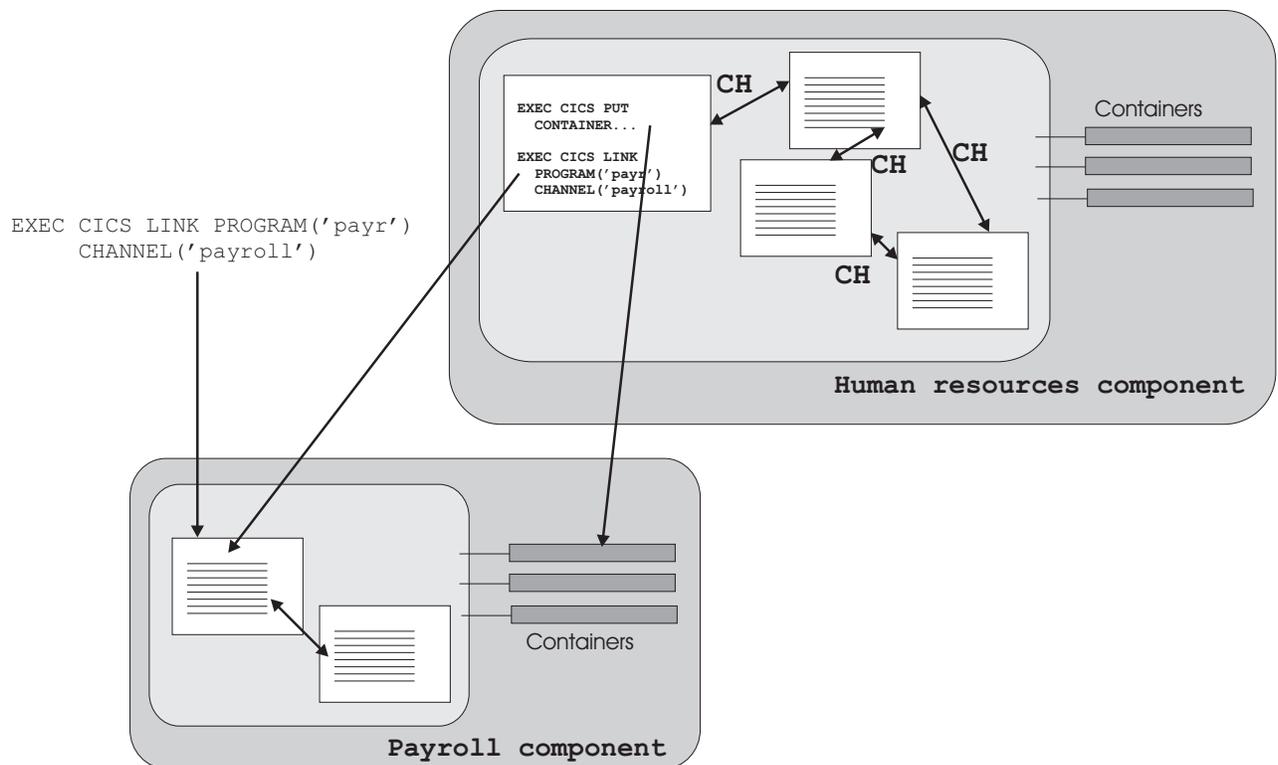


図 21. チャンネルを介して相互作用する複数のコンポーネント

チャンネルの作成

チャンネルを作成するには、以下のコマンドのいずれかでチャンネル名を指定します。

```
EXEC CICS LINK PROGRAM CHANNEL
EXEC CICS MOVE CONTAINER CHANNEL TOCHANNEL
EXEC CICS PUT CONTAINER CHANNEL
EXEC CICS RETURN TRANSID CHANNEL
EXEC CICS START TRANSID CHANNEL
EXEC CICS XCTL PROGRAM CHANNEL
```

現在のプログラム・スコープ内にチャンネルがまだ存在しない場合は、チャンネルが作成されます。

チャンネルを作成してチャンネルにデータのコンテナを取り込むための最も簡単な方法は、一連の EXEC CICS PUT CONTAINER(container-name) CHANNEL(channel-name) FROM(data_area) コマンドを発行することです。最初の PUT コマンドによって (チャンネルがまだ存在しない場合は) チャンネルが作成され、これにコンテナが追加されます。後続のコマンドによって、追加のコンテナがチャンネルに追加されます。コンテナが既に存在する場合、その内容が新しいデータで上書きされます。

チャンネルにコンテナを追加するための代替方法は、別のチャンネルからコンテナを移動することです。このためには、以下のコマンドを使用します。

```
EXEC CICS MOVE CONTAINER(container-name) AS(container-new-name)
      CHANNEL(channel-name1) TOCHANNEL(channel-name2)
```

注:

1. CHANNEL または TOCHANNEL オプションを指定しなかった場合、現行チャンネルが暗黙指定されます。『現行チャンネル』を参照してください。
2. ソース・チャンネルは、プログラム・スコープ内になければなりません。
3. 現在のプログラム・スコープ内にターゲット・チャンネルがまだ存在しない場合は、チャンネルが作成されます。
4. ソース・コンテナが存在しない場合は、エラーが発生します。
5. ターゲット・コンテナがまだ存在しない場合は、これが作成されます。ターゲット・コンテナが既に存在する場合は、その内容が上書きされます。

チャンネル間でより効率的にデータを転送する方法として、GET CONTAINER および PUT CONTAINER の代わりに MOVE CONTAINER を使用できます。

以下のコマンドで指定されたチャンネルが現在のプログラム・スコープ内にまだ存在しない場合は、空のチャンネルが作成されます。

- EXEC CICS LINK PROGRAM CHANNEL(channel-name)
- EXEC CICS RETURN TRANSID CHANNEL(channel-name)
- EXEC CICS START TRANSID CHANNEL(channel-name)
- EXEC CICS XCTL PROGRAM CHANNEL(channel-name)

現行チャンネル

プログラムの現行チャンネルは、プログラムの呼び出し時に指定されたチャンネルです(存在する場合)。プログラムは別のチャンネルを作成できます。ただし、特定プログラムの特定の呼び出し用の現行チャンネルは変更されません。これは、パラメーター・リストに類似しています。

現行チャンネルの例 (LINK コマンドの場合)

210 ページの図 22 は、プログラムの現行チャンネルの起点を示しています。5 つの対話式プログラムが示されています。プログラム A は最上位のプログラムであり、例えば、端末のエンド・ユーザーによって開始されます。これはプログラムによって開始されたものではなく、現行チャンネルを持っていません。

B、C、D、および E は、それぞれ第 2 レベル、第 3 レベル、第 4 レベル、および第 5 レベルのプログラムです。

プログラム B の現行チャンネルはプログラム A が発行した EXEC CICS LINK コマンドの CHANNEL オプションによって渡された X です。プログラム B は、あるコンテナを追加して別のコンテナを削除することによりチャンネル X を変更しません。

プログラム C の現行チャンネルもまた、プログラム B が発行した EXEC CICS LINK コマンドの CHANNEL オプションによって渡された X です。

C はプログラム D に現行チャンネルを渡さないため、D はチャンネルを持っていません。

プログラム E の現行チャンネルは、D が発行した EXEC CICS LINK コマンドの CHANNEL オプションによって渡された Y です。

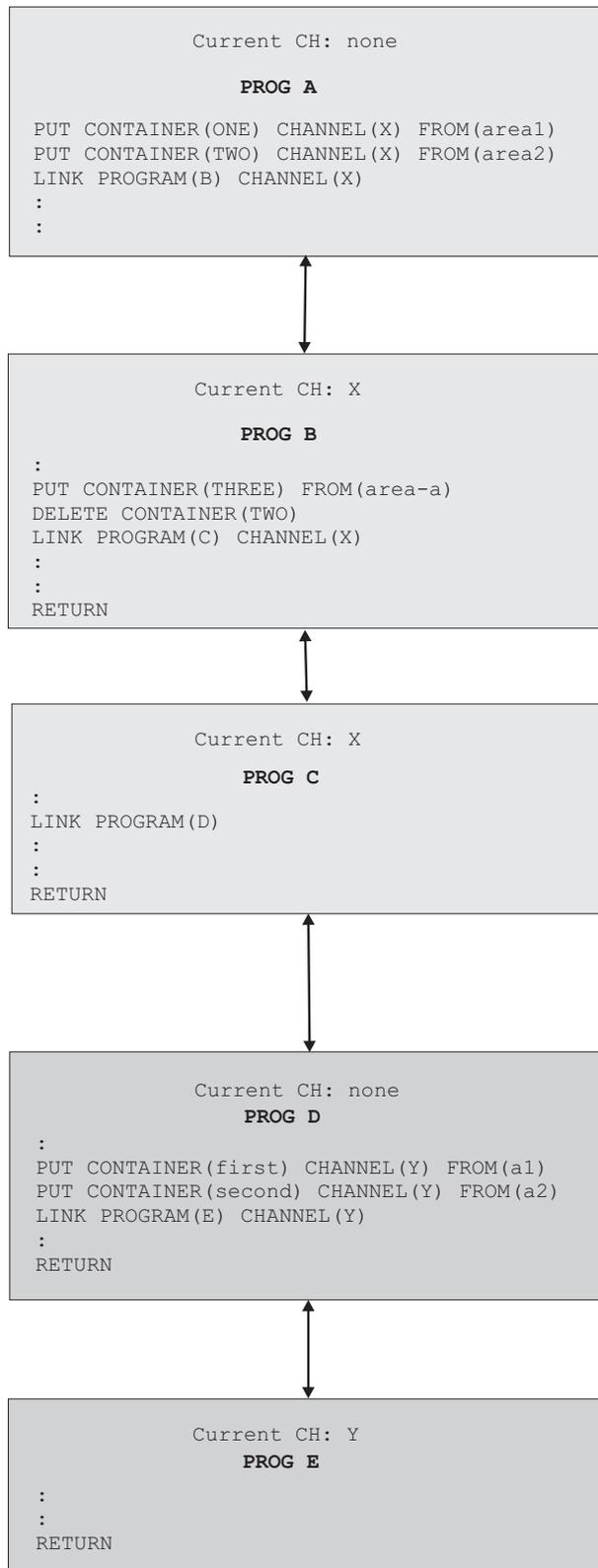


図 22. 現行チャンネルの例 (*LINK* コマンドの場合)

211 ページの表 1 は、図 22 に示した 5 つのプログラムのそれぞれについて、現行チャンネルが存在する場合はその名前を示しています。

表 1. 対話式プログラムの現行チャンネルの例 (LINK コマンドの場合)

プログラム	現行チャンネル	発行するコマンド	コメント
A	None	<pre> . EXEC CICS PUT CONTAINER(ONE) CHANNEL(X) FROM(area1) EXEC CICS PUT CONTAINER(TWO) CHANNEL(X) FROM(area2) EXEC CICS LINK PROGRAM(B) CHANNEL(X) . </pre>	<p>プログラム A はチャンネル X を作成し、これをプログラム B に渡します。</p> <p>プログラム B がプログラム A に制御を戻すまでに X チャンネルは変更されていることに注意してください。X チャンネルには、プログラム A によって作成された時点と同じコンテナのセットは含まれていません。(プログラム B によってコンテナ TWO が削除され、コンテナ THREE が追加されています。)</p>
B	X	<pre> . EXEC CICS PUT CONTAINER(THREE) FROM(area-a) EXEC CICS DELETE CONTAINER(TWO) EXEC CICS LINK PROGRAM(C) CHANNEL(X) . EXEC CICS RETURN </pre>	<p>プログラム B は、コンテナを追加および削除してチャンネル X (自身の現行チャンネル) を変更し、変更済みチャンネルをプログラム C に渡します。</p> <p>プログラム B は、PUT CONTAINER および DELETE CONTAINER コマンドで CHANNEL オプションを指定する必要がありません。なぜなら、自身の現行チャンネルが暗黙指定されるからです。</p>
C	X	<pre> . EXEC CICS LINK PROGRAM(D) . EXEC CICS RETURN </pre>	<p>プログラム C はプログラム D にリンクしますが、チャンネルは渡しません。</p>
D	None	<pre> . EXEC CICS PUT CONTAINER(first) CHANNEL(Y) FROM(a1) EXEC CICS PUT CONTAINER(second) CHANNEL(Y) FROM(a2) EXEC CICS LINK PROGRAM(E) CHANNEL(Y) . EXEC CICS RETURN </pre>	<p>プログラム D は新規チャンネル Y を作成し、これをプログラム E に渡します。</p>
E	Y	<pre> . RETURN . </pre>	<p>プログラム E は渡されたデータに対して何らかの処理を行い、処理済みデータを戻します。</p>

現行チャンネルの例 (XCTL コマンドの場合)

212 ページの図 23 は、4 つの対話式プログラムを示しています。A1 は最上位のプログラムであり、例えば、端末のエンド・ユーザーによって開始されます。これはプログラムによって開始されたものではなく、現行チャンネルを持っていません。B1、B2、および B3 はすべて、第 2 レベルのプログラムです。

B1 の現行チャンネルは、A1 が発行した EXEC CICS LINK コマンドの CHANNEL によって渡された X です。

B1 は B2 に現行チャンネルを渡さないため、B2 はチャンネルを持っていません。

B3 の現行チャンネルは、B2 が発行した EXEC CICS XCTL コマンドの CHANNEL オプションによって渡された Y です。

B3 が戻ると、チャンネル Y とそのコンテナは CICS によって削除されます。

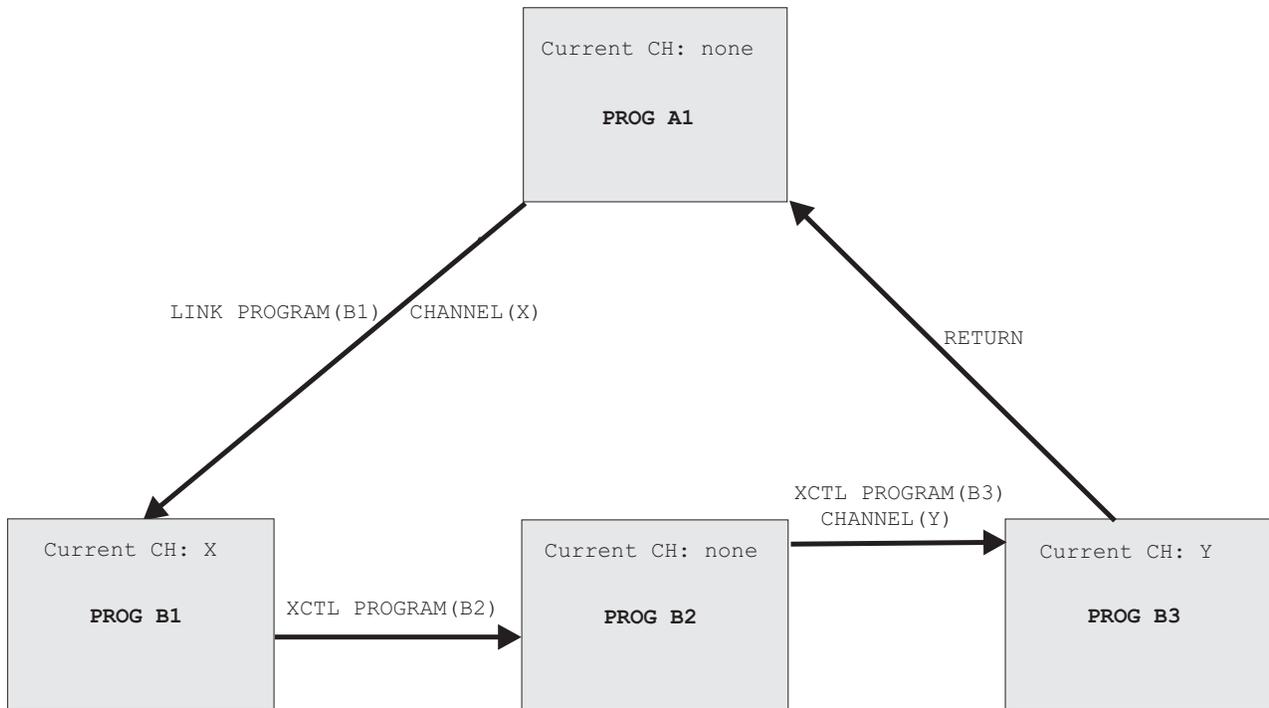


図 23. 現行チャンネルの例 (XCTL コマンドの場合)

表 2 は、図 23 に示した 4 つのプログラムのそれぞれについて、現行チャンネルが存在する場合はその名前を示しています。

表 2. 対話式プログラムの現行チャンネルの例

プログラム	現行 [®] チャンネル	発行するコマンド
A1	None	. EXEC CICS LINK PROGRAM(B1) CHANNEL(X) .
B1	X	. EXEC CICS XCTL PROGRAM(B2) .
B2	None	. EXEC CICS XCTL PROGRAM(B3) CHANNEL(Y) .
B3	Y	. EXEC CICS RETURN .

現行チャンネル: START コマンドおよび RETURN コマンド

EXEC CICS LINK および XCTL に加えて、プログラムを呼び出してチャンネルを渡すために使用できるコマンドがほかに 2 つあります。

EXEC CICS START TRANSID(*tranid*) CHANNEL(*channel-name*)

開始済みのトランザクションを実装するプログラム (または、複数のプログラムが存在する場合は最初のプログラム) にチャンネルが渡され、このチャンネルが現行チャンネルになります。

EXEC CICS RETURN TRANSID(*tranid*) CHANNEL(*channel-name*)

CHANNEL オプションは、以下の場合にのみ有効です。

- 疑似会話型 RETURN。すなわち、TRANSID オプションで、ユーザー端末で実行する次のトランザクションを指定する RETURN コマンド。
- 最上位の論理レベルのプログラム (すなわち、CICS に制御を戻すプログラム) によって発行された場合。

次のトランザクションをインプリメントするプログラムにチャンネルが渡され、このチャンネルが現行チャンネルになります。

チャンネルのスコープ

チャンネルのスコープは、そのチャンネルにアクセスできるコードです。

スコープの例 (LINK コマンドの場合)

214 ページの図 24 は、208 ページの『現行チャンネルの例 (LINK コマンドの場合)』で説明したのと同じ 5 つの対話式プログラムを示しています。

X チャンネルのスコープ (アクセス可能なコード) は、プログラム A、B、および C です。

Y チャンネルのスコープはプログラム D および E です。

プログラム B がプログラム A に制御を戻すまでには、X チャンネルは変更されていることに注意してください。X チャンネルには、プログラム A がこれを作成した時点と同じコンテナのセットは含まれていません。

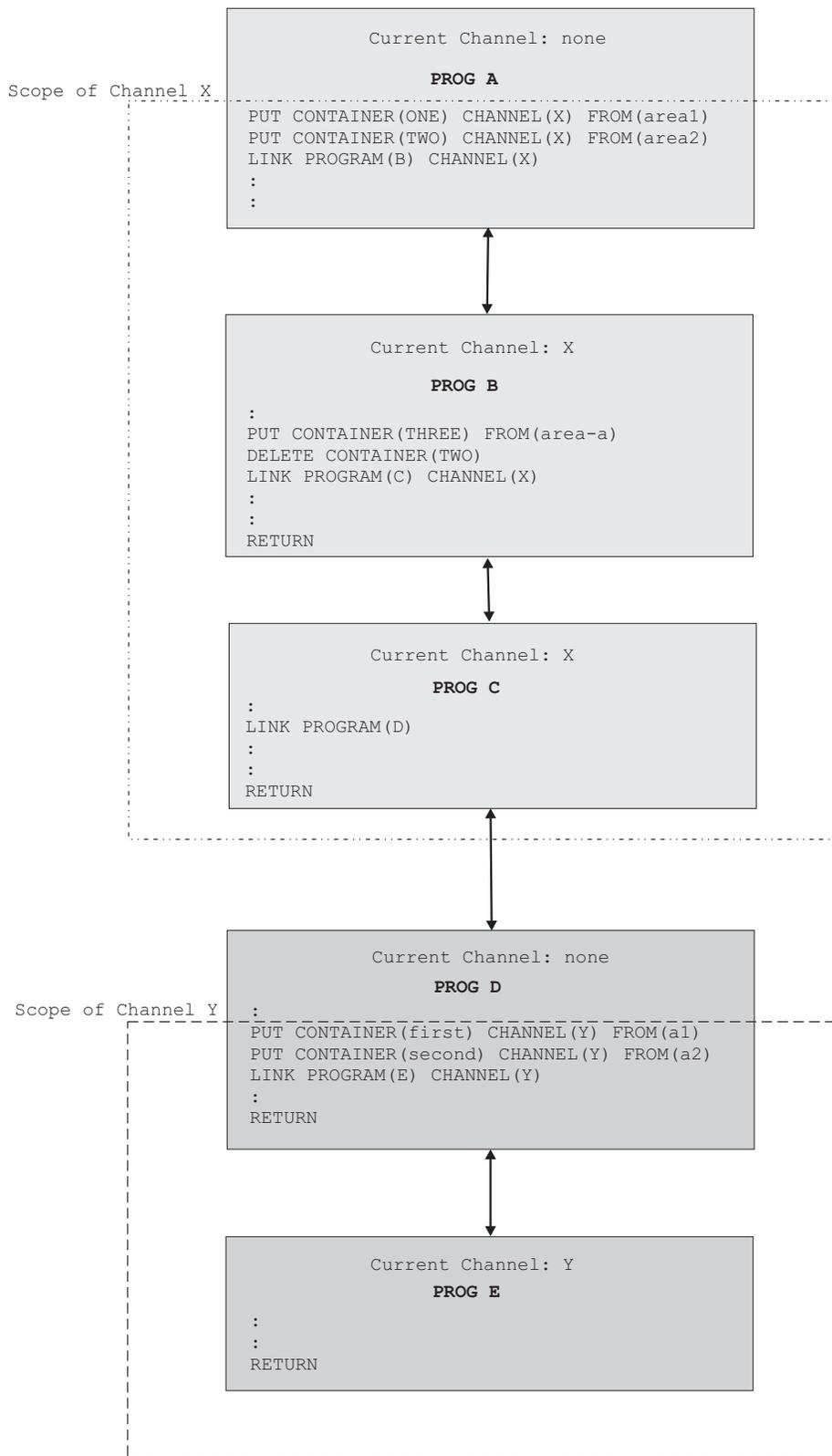


図 24. チャンネルのスコープ: LINK コマンドの場合の例

215 ページの表 3 は、図 24 に示した 5 つのプログラムのそれぞれについて、現行チャンネルが存在する場合はその名前とスコープを示しています。

表 3. チャネルのスコープ: LINK コマンドの場合の例

プログラム	現行チャネル	チャネルのスコープ
A	None	適用されない
B	X	A、B、C
C	X	A、B、C
D	None	適用されない
E	Y	D、E

スコープの例 (LINK および XCTL コマンドの場合)

216 ページの図 25 は、211 ページの『現行チャネルの例 (XCTL コマンドの場合)』で示したのと同じ 4 つの対話式プログラムと、プログラム B1 から EXEC CICS LINK コマンドによって呼び出される第 3 レベルのプログラム C1 を示しています。

X チャネルのスコープは A1 および B1 に制限されます。

Y チャネルのスコープは B2 および B3 です。

Z チャネルのスコープは B1 および C1 です。

プログラム B3 がプログラム A1 に制御を戻すまでには、X チャネルがプログラム B1 によって変更されている可能性があることに注意してください。X チャネルには、プログラム A1 がこれを作成した時点と同じコンテナのセットが含まれていない可能性があります。

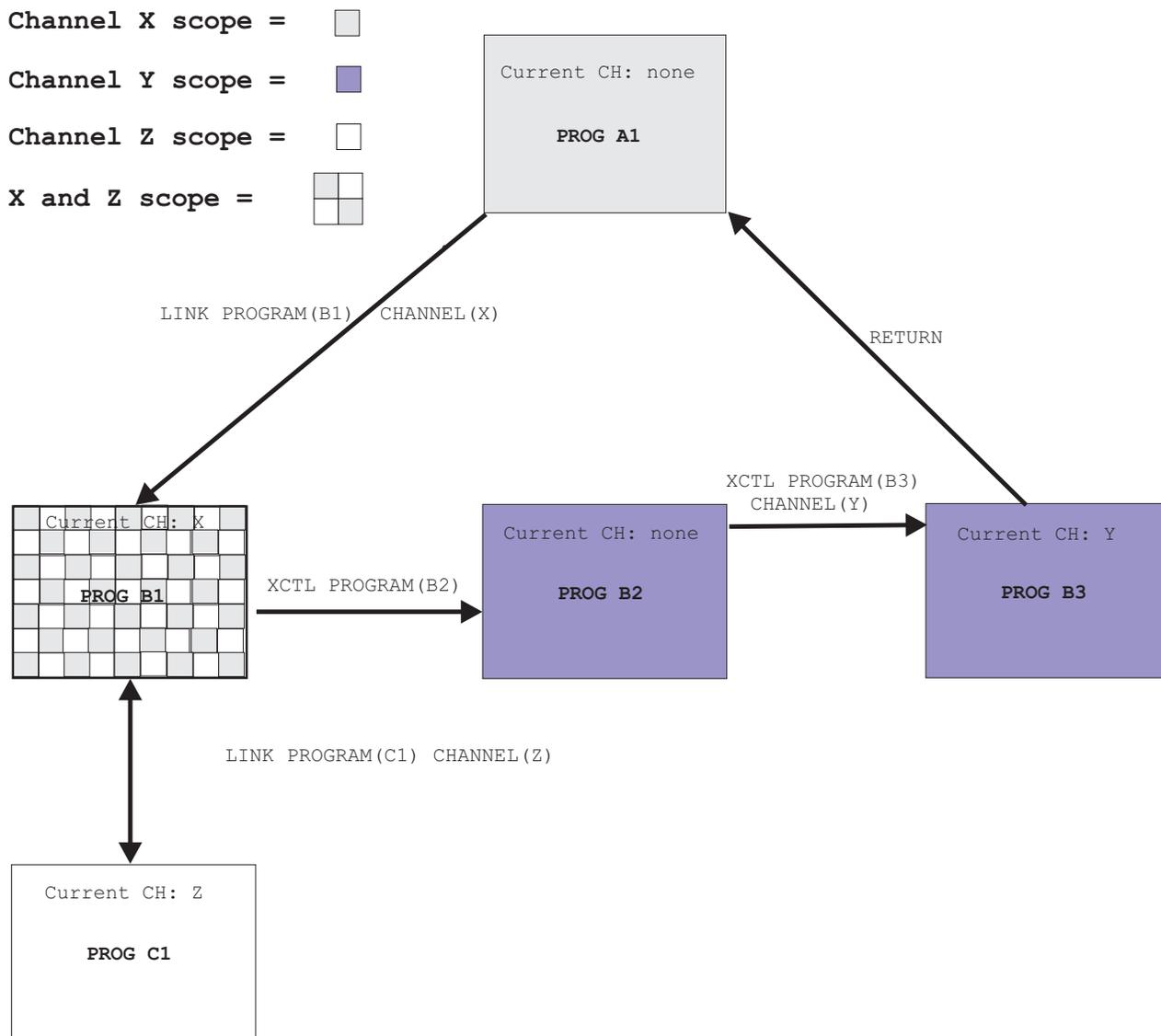


図 25. チャンネルのスコープ: LINK および XCTL コマンドの場合の例

表 4 は、図 25 に示した 5 つのプログラムのそれぞれについて、現行チャンネルが存在する場合はその名前とスコープを示しています。

表 4. チャンネルのスコープ: LINK および XCTL コマンドの場合の例

プログラム	現行チャンネル	チャンネルのスコープ
A1	None	適用されない
B1	X	A1 および B1
B2	None	適用されない
B3	Y	B2 および B3
C1	Z	B1 および C1

プログラムに渡されたコンテナの発見

一般に、チャンネルを交換するプログラムは、そのチャンネルを処理するために作成されます。すなわち、クライアント・プログラムとサーバー・プログラムの両方が、チャンネルの名前、およびチャンネル内のコンテナの名前と数を知っています。しかし、例えば、複数のチャンネルを処理するために 1 つのサーバー・プログラムまたはコンポーネントが作成された場合、呼び出し時に渡されたチャンネルがどれであるかを発見する必要があります。

プログラムは、EXEC CICS ASSIGN CHANNEL コマンドを発行することにより、その現行チャンネル (呼び出し時に指定されたチャンネル) を発見することができます。(現行チャンネルがない場合、コマンドは空白を戻します。)

また、プログラムは、(必要な場合は) ブラウズして、現行チャンネル内のコンテナの名前を取得することもできます。通常は、これは必要ありません。複数のチャンネルを処理するように作成されたプログラムは、多くの場合、考えられるそれぞれのチャンネル内のコンテナの名前と数を認識するようにコーディングされています。

現行チャンネル内のコンテナの名前を取得するには、ブラウズ・コマンドを使用します。

- EXEC CICS STARTBROWSE CONTAINER BROWSETOKEN(*data-area*)
- EXEC CICS GETNEXT CONTAINER(*data-area*) BROWSETOKEN(*token*)
- EXEC CICS ENDBROWSE CONTAINER BROWSETOKEN(*token*)

現行チャンネルの名前および必要に応じてチャンネル内のコンテナの名前を取得後、サーバー・プログラムは、渡されたデータの種類の合わせて処理を調整することができます。

リンクから戻されたコンテナの発見

プログラムはチャンネルを作成し、EXEC CICS LINK PROGRAM(*program-name*) CHANNEL(*channel-name*) コマンドを使用して、これを 2 番目のプログラムに渡します。2 番目のプログラムは、渡されたデータに対して何らかの処理を行い、同じチャンネル (自身の現行チャンネル) に結果を戻します。

戻り時に、最初のプログラムは、戻されたチャンネルの名前を知っていますが、チャンネル内のコンテナの名前は必ずしも知っているわけではありません。(戻されたチャンネルは渡されたチャンネルと同じコンテナを含む、あるいは 2 番目のプログラムが一部のコンテナを削除したり他のコンテナを作成した可能性があります。) この最初のプログラムは、ブラウズしてコンテナ名を発見することができます。このためには、以下のコマンドを使用します。

- EXEC CICS STARTBROWSE CONTAINER BROWSETOKEN(*data-area*)
CHANNEL(*channel-name*)
- EXEC CICS GETNEXT CONTAINER(*data-area*) BROWSETOKEN(*token*)
- EXEC CICS ENDBROWSE CONTAINER BROWSETOKEN(*token*)

CICS 読み取り専用コンテナ

CICS は、自身が使用するためにチャンネルおよびコンテナを作成し、これをユーザー・プログラムに渡すことができます。場合によっては、CICS はこれらのコンテナに読み取り専用のマークを付け、ユーザー・プログラムから CICS に戻す必要があるデータをユーザー・プログラムが変更できないようにすることができます。

ユーザー・プログラムは読み取り専用コンテナを作成できません。

読み取り専用コンテナの上書き、移動、削除を行うことはできません。したがって、PUT CONTAINER、MOVE CONTAINER、または DELETE CONTAINER コマンドで読み取り専用コンテナを指定すると、INVREQ 条件が発生します。

チャンネルの設計: ベスト・プラクティス

通信域 (COMMAREA) で従来使用されていたのと同じ方法で、コンテナを使用してデータを渡すことができます。ただし、チャンネルには COMMAREA よりも優れた点がいくつかあり (199 ページの『チャンネルの利点』を参照)、独自のチャンネルを設計してこれらの改善点を最大限に利用することには、それだけの価値があります。

サーバー・プログラムによって変更されなかった入力コンテナは、**DPL 呼び出しの最後にクライアントに戻されません**。内容がサーバー・プログラムによって変更された入力コンテナ、およびサーバー・プログラムによって作成されたコンテナが戻されます。そのため、DPL のパフォーマンスを最大にするには、以下のようになります。

- 入力データと出力データに別個のコンテナを使用する。
- クライアントではなく、サーバー・プログラムが出力コンテナを作成する。
- 読み取り専用データと読み取り/書き込みデータに別個のコンテナを使用する。
- 構造がオプションである場合、それを別個のコンテナにする。
- エラー情報のために専用のコンテナを使用する。

以下に、チャンネルを設計する上でのいくつかの一般的なヒントを示します。最高の DPL パフォーマンスを達成するための推奨事項が含まれます。

- 入力データと出力データに別個のコンテナを使用する。これにより、以下がもたらされます。
 - データのカプセル化が向上し、プログラムの保守が容易になります。
 - DPL 呼び出しでチャンネルが渡されるときに、各方向に流れるコンテナが小さくなるため、効率が上がります。
- クライアントではなく、サーバー・プログラムが出力コンテナを作成する。クライアントが作成した場合、空のコンテナがサーバー領域に送られます。
- 読み取り専用データと読み取り/書き込みデータに別個のコンテナを使用する。これにより、以下がもたらされます。
 - コピーブック構造が単純化され、プログラムが理解しやすくなります。
 - REORDER オーバーレイに伴う問題が避けられます。
 - サーバー領域に送られた読み取り専用コンテナが戻されないため、CICS 領域間の伝送効率が向上します。

- それぞれの構造ごとに別個のコンテナを使用する。これにより、以下がもたらされます。
 - データのカプセル化が向上し、プログラムの理解および保守が容易になります。
 - コンポーネント全体を再コンパイルする必要がないため、構造のうちの 1 つを変更するのが容易になります。
 - MOVE CONTAINER コマンドを使用してチャンネル間でコンテナを移動することにより、チャンネルのサブセットをサブコンポーネントに渡すことができます。
 - 構造がオプションである場合、それを別個のコンテナにする。これにより、コンテナが存在する場合のみ構造が渡されるため、効率が高まります。
 - エラー情報のために専用のコンテナを使用する。これにより、以下がもたらされます。
 - エラー情報の説明の文書化が容易になります。
 - 以下の理由により、効率が高まります。
 1. エラーが発生した場合にのみエラー情報を含む構造が戻されます。
 2. GET CONTAINER(*known-error-container-name*) コマンドを発行して (および場合によっては NOTFOUND 条件を受信して) エラー・コンテナの存在を確認する方が、チャンネル内のコンテナのブラウズを開始するよりも効率が良くなります。
 - 異なるタイプ (例えば、バイナリー・データと文字データ) のデータを渡す必要があるときは、複雑な構造を持つ 1 つのコンテナを使用するのではなく、それぞれのタイプごとに別個のコンテナを使用する。これにより、異なるコード・ページ間を移動しやすくなります。
 - 大量のデータを渡す必要があるときは、すべてのデータを 1 つのコンテナに入れるのではなく、データを複数のコンテナ間で分割する。
- リモート・プログラムまたはリモート・トランザクションにチャンネルを渡す場合、大量のデータを渡すとパフォーマンスに影響する可能性がある。これは、ローカル領域とリモート領域が MRO ではなく ISC 接続で接続されている場合に特に問題になります。
- 大容量のコンテナを多数作成して、他のアプリケーションが使用可能なストレージの量を制限してしまわないように注意する。

代わりに COMMAREA を用いてチャンネルを使用するマイグレーション・プログラムについては、257 ページの『COMMAREA からチャンネルへのマイグレーション』を参照してください。

チャンネルの生成と使用: 例

220 ページの図 26 は、以下を行う CICS クライアント・プログラムを示しています。

1. EXEC CICS PUT CONTAINER コマンドを使用して、コンテナのセットを生成する (およびこれにデータを書き込む)。コンテナはすべて、同じ名前付きチャンネル「payroll-2004」の一部です。

2. EXEC CICS LINK コマンドを発行して PAYR サーバー・プログラムを呼び出し、これに payroll-2004 チャンネルを渡す。
3. EXEC CICS GET CONTAINER コマンドを発行してサーバーのプログラム出力を取得する。クライアント・プログラムは、これが payroll-2004 チャンネルの status コンテナにあることを知っています。

```
* create the employee container on the payroll-2004 channel  
EXEC CICS PUT CONTAINER('employee') CHANNEL('payroll-2004') FROM('John Doe')
```

```
* create the wage container on the payroll-2004 channel  
EXEC CICS PUT CONTAINER('wage') CHANNEL('payroll-2004') FROM('100')
```

```
* invoke the payroll service, passing the payroll-2004 channel  
EXEC CICS LINK PROGRAM('PAYR') CHANNEL('payroll-2004')
```

```
* examine the status returned on the payroll-2004 channel  
EXEC CICS GET CONTAINER('status') CHANNEL('payroll-2004') INTO(stat)
```

図 26. クライアント・プログラムがチャンネルを生成し、サーバー・プログラムにこれを渡してサーバーの出力を取得する仕組み

221 ページの図 27 は、クライアントによって呼び出された PAYR サーバー・プログラムの一部を示しています。サーバー・プログラムには、以下のことが適用されます。

1. 呼び出しで指定されたチャンネルを照会する。
2. EXEC CICS GET CONTAINER コマンドを発行して、payroll-2004 チャンネルの employee および wage コンテナからの入力を取得する。
3. 入力データを処理する。
4. EXEC CICS PUT CONTAINER コマンドを発行して、その出力を payroll-2004 チャンネルの status コンテナに戻す。

CICS COBOL サーバー・プログラム "PAYR"

```
* discover which channel I've been invoked with
EXEC CICS ASSIGN CHANNEL(ch_name)
:
WHEN ch_name 'payroll-2004'
  * my current channel is "payroll-2004"
  * get the employee passed into this program
  EXEC CICS GET CONTAINER('employee') INTO(emp)
  * get the wage for this employee
  EXEC CICS GET CONTAINER('wage') INTO(wge)
  :
  * process the input data
  :
  :
  * return the status to the caller by creating the status container
  * on the payroll channel and putting a value in it
  EXEC CICS PUT CONTAINER('status') FROM('OK')
  :
  :
WHEN ch_name 'payroll-2005'
  * my current channel is "payroll-2005"
  :
  :
  :
```

図 27. サーバー・プログラムが渡されたチャンネルを照会し、チャンネルのコンテナからデータを取得して呼び出し元へ出力を戻す仕組み

チャンネルと BTS アクティビティ

チャンネルの作成およびチャンネルとの対話に使用される PUT、GET、MOVE、および DELETE CONTAINER コマンドは、CICS ビジネス・トランザクション・サービス (BTS) アプリケーションで使用されるこれらのコマンドと似ています。したがって、BTS の経験を持つプログラマーにとって、非 BTS アプリケーションにおけるコンテナの使用は容易です。さらに、コンテナを使用するサーバー・プログラムは、チャンネル・アプリケーションおよび BTS アプリケーションの両方から呼び出すことができます。この例を 222 ページの図 28 に示します。

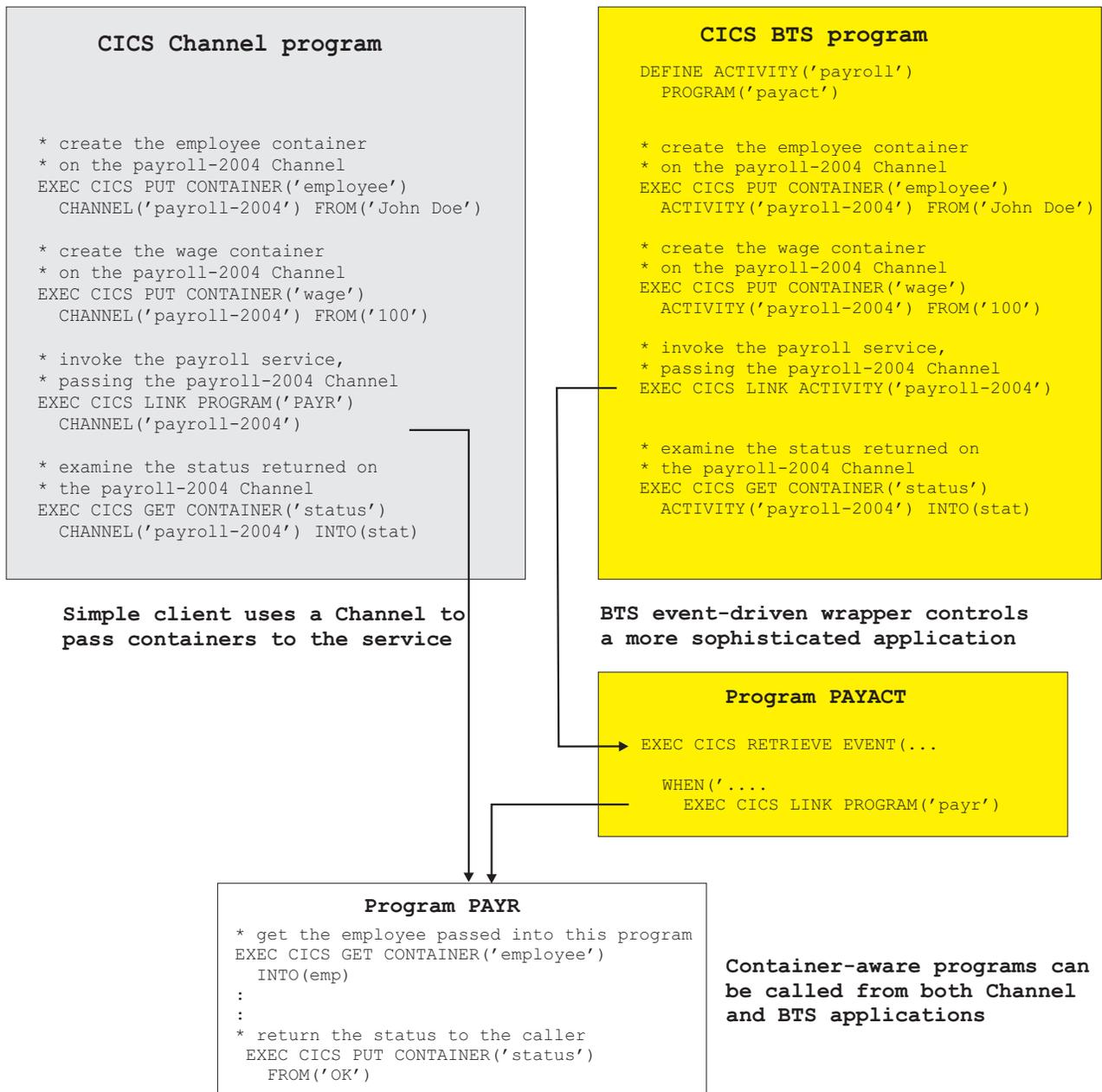


図 28. チャネルと BTS アクティビティー

コンテキスト

図 28 に示すように、CONTAINER コマンドを発行するプログラムは、変更せずにそのままチャネル・アプリケーションまたは BTS アクティビティーの一部として使用できます。

チャネルおよび BTS の両方のコンテキストで使用するプログラムの場合、プログラムが発行する CONTAINER コマンドでチャネルまたは BTS コマンドとして識別されるオプションを指定することはできません。それぞれの CONTAINER コマンドで避ける必要があるオプションは以下のとおりです。

DELETE CONTAINER

ACQACTIVITY (BTS 固有)
ACQPROCESS (BTS 固有)
ACTIVITY (BTS 固有)
CHANNEL (チャンネル固有)
PROCESS (BTS 固有)

GET CONTAINER

ACQACTIVITY (BTS 固有)
ACQPROCESS (BTS 固有)
ACTIVITY (BTS 固有)
CHANNEL (チャンネル固有)
INTOCCSID (チャンネル固有)
PROCESS (BTS 固有)

MOVE CONTAINER

FROMACTIVITY (BTS 固有)
CHANNEL (チャンネル固有)
FROMPROCESS (BTS 固有)
TOACTIVITY (BTS 固有)
TOCHANNEL (チャンネル固有)
TOPROCESS (BTS 固有)

PUT CONTAINER

ACQACTIVITY (BTS 固有)
ACQPROCESS (BTS 固有)
ACTIVITY (BTS 固有)
CHANNEL (チャンネル固有)
DATATYPE (チャンネル固有)
FROMCCSID (チャンネル固有)
PROCESS (BTS 固有)

CONTAINER コマンドの実行時、CICS はコマンドの処理方法を判別するために、コマンドが発生したコンテキスト (チャンネル、BTS、またはそのどちらでもない) を分析します。コンテキストを判別するために、CICS は以下のテスト・シーケンスを使用します。

1. チャンネル: プログラムは現行チャンネルを持っているか?
2. BTS: プログラムは BTS アクティビティーの一部か?
3. なし: プログラムは現行チャンネルを持っておらず、BTS アクティビティーの一部ではない。したがって、CONTAINER コマンドを実行するコンテキストを持っていません。コマンドは、INVREQ 条件および RESP2 値 4 で拒否されます。

JCICS からのチャンネルの使用

CICS では、チャンネルの受け渡しのために CICS Java プログラムで使用できる以下の JCICS クラスを提供しています。

- `com.ibm.cics.server.Channel`
- `com.ibm.cics.server.Container`
- `com.ibm.cics.server.ContainerIterator`

また、CICS では、エラー処理のために以下の例外クラスを提供しています。

- `com.ibm.cics.server.CCSIDErrorException`
- `com.ibm.cics.server.ChannelErrorException`
- `com.ibm.cics.server.ContainerErrorException`

JCICS でのチャネルおよびコンテナの作成

チャネルを作成するには、Task クラスの `createChannel()` メソッドを使用します。以下に例を示します。

```
Task t=Task.getTask();
Channel custData = t.createChannel("Customer_Data");
```

`createChannel` メソッドに提供される String は、CICS が Channel オブジェクトを認識するために使用する名前です。(CICS 命名規則に準拠するため、16 文字になるように名前にスペースが埋め込まれます。)

チャネルに新規コンテナを作成するには、Channel の `createContainer()` メソッドを使用します。以下に例を示します。

```
Container custRec = custData.createContainer("Customer_Record");
```

`createContainer()` メソッドに提供される String は、CICS が Container オブジェクトを認識するために使用する名前です。(CICS 命名規則に準拠するため、16 文字になるように必要に応じて名前にスペースが埋め込まれます。) このチャネルに同じ名前のコンテナが既に存在する場合は、`ContainerErrorException` がスローされます。

コンテナへのデータの書き込み

Container オブジェクトにデータを書き込むには、`Container.put()` メソッドを使用します。データは、バイト配列または String として Container に追加できます。以下に例を示します。

```
String custNo = "00054321";
byte[] custRecIn = custNo.getBytes();
custRec.put(custRecIn);
```

あるいは、単に以下のようにします。

```
custRec.put("00054321");
```

別のプログラムまたはタスクへのチャネルの引き渡し

プログラム・リンクまたはプログラム制御移転 (XCTL) 呼び出しでチャネルを渡すには、Program クラスの `link()` および `xctl()` メソッドをそれぞれ使用します。

```
programX.link(custData);
```

```
programY.xctl(custData);
```

プログラム戻り呼び出しで次のチャネルを設定するには、TerminalPrincipalFacility クラスの `setNextChannel()` メソッドを使用します。

```
terminalPF.setNextChannel(custData);
```

START 要求でチャネルを渡すには、StartRequest クラスの `issue` メソッドを使用します。

```
startrequest.issue(custData);
```

現行チャンネルの受取

プログラムは、明示的に現行チャンネルを受け取る必要がありません。『現行チャンネルのブラウズ』を参照してください。ただし、プログラムは現行タスクから現行チャンネルを取得することができます。これにより、コンテナを名前で取り出すことができるようになります。

```
Task t = Task.getTask();
Channel custData = t.getCurrentChannel();
if (custData != null) {
    Container custRec = custData.getContainer("Customer_Record");
} else {
    System.out.println("There is no Current Channel");
}
```

コンテナからのデータの取得

コンテナのデータをバイト配列に読み込むには、`Container.get()` メソッドを使用します。

```
byte[] custInfo = custRec.get();
```

現行チャンネルのブラウズ

チャンネルを渡された JCICS プログラムは、明示的にチャンネルを受信しなくても、すべての `Container` オブジェクトにアクセスできます。このためには、`ContainerIterator` オブジェクトが使用されます。(`ContainerIterator` クラスは、`java.util.Iterator` インターフェースを実装しています。) 現在のタスクから `Task` オブジェクトのインスタンスが生成されると、その `containerIterator()` メソッドが現行チャンネル用の `Iterator` (または現行チャンネルがない場合は `Null`) を戻します。以下に例を示します。

```
Task t = Task.getTask();
ContainerIterator ci = t.containerIterator();
While (ci.hasNext()) {
    Container custData = ci.next();
    // Process the container...
}
```

JCICS の例

226 ページの図 29 に、`PAYR` という COBOL サーバー・プログラムを呼び出す、`Payroll` と呼ばれる Java クラスを示します。`Payroll` クラスは、JCICS `com.ibm.cics.server.Channel` クラスおよび `com.ibm.cics.server.Container` クラスを使用して、非 Java クライアント・プログラムが `EXEC CICS` コマンドを使用して実行する処理と同じ処理を実行します。

```

import com.ibm.cics.server.*;
public class Payroll {
    ...
    Task t=Task.getTask();

    // create the payroll_2004 channel
    Channel payroll_2004 = t.createChannel("payroll-2004");

    // create the employee container
    Container employee = payroll_2004.createContainer("employee");

    // put the employee name into the container
    employee.put("John Doe");

    // create the wage container
    Container wage = payroll_2004.createContainer("wage");

    // put the wage into the container
    wage.put("2000");

    // Link to the PAYROLL program, passing the payroll_2004 channel
    Program p = new Program();
    p.setName("PAYR");
    p.link(payroll_2004);

    // Get the status container which has been returned
    Container status = payroll_2004.getContainer("status");

    // Get the status information
    byte[] payrollStatus = status.get();
    ...
}

```

図 29. JCICS *com.ibm.cics.server.Channel* および *com.ibm.cics.server.Container* クラスを使用し て COBOL サーバー・プログラムにチャンネルを渡す Java クラス

チャンネルを使用した動的ルーティング

COMMAREA またはチャンネルのいずれかの受け渡しが可能な EXEC CICS LINK コマンドおよび EXEC CICS START コマンドは、動的にルーティングすることができます。次のタイプのチャンネル関連の要求を動的にルーティングすることができます。

- プログラム・リンク (DPL) 要求
- 端末関連の START 要求によって開始されるトランザクション
- 端末関連以外の START 要求

動的にルーティングされたプログラム・リンクまたは START コマンドに関連付けられたチャンネルがある場合は、チャンネルの名前がルーティング・プログラムに渡され、その際に通信域の DYRCHANL フィールドが使用されます。DYRCHANL フィールドは、上記の 3 つのタイプの要求にのみ適用されます。その他のタイプの要求の場合、あるいは要求に関連付けられたチャンネルがない場合、このフィールドはブランクになります。

注: ルーティング・プログラムの通信域は、DFHDYPDS DSECT によってマップされます。

ルーティング・プログラムにはチャンネルのアドレスではなく、名前が渡されるので、DYRCHANL フィールドを使用してコンテナの内容を検査したり、変更したりできないことに注意してください。

LINK コマンドまたは START コマンドが、チャンネルではなく COMMAREA を受け渡す場合は、ルーティング・プログラムは、要求のタイプに応じて、COMMAREA の内容を検査したり、変更したりすることができます。端末関連の START 要求によって開始される LINK 要求およびトランザクション (動的ルーティング・プログラムが処理する) では、ルーティング・プログラムの DFHDYPDS の DYRACMAA フィールドに、アプリケーションの COMMAREA のアドレスが渡されるため、その内容の検査と変更が可能です。ただし、端末関連以外の START 要求 (分散ルーティング・プログラムが処理する) には当てはまりません。

ルーティング・プログラムにチャンネルを使用して同種の機能を提供するために、チャンネルを使用するアプリケーションがチャンネル内に、DFHROUTE という特殊なコンテナを作成することができます。アプリケーションが動的にルーティングされる LINK または端末関連の START 要求 (端末関連以外の START 要求ではなく) を出すと、動的ルーティング・プログラムの DFHDYPDS の DYRACMAA フィールドに DFHROUTE コンテナのアドレスが渡されるため、動的ルーティング・プログラムはその内容を検査および変更することができます。

COMMAREA ではなくチャンネルを渡すようにプログラムを変更する場合は、既存の COMMAREA 構造を使用して DFHROUTE をマップすることができます。

動的ルーティングおよび分散ルーティングの概説については、「*CICS Intercommunication Guide*」を参照してください。動的ルーティング・プログラムまたは分散ルーティング・プログラムの作成方法については、「*CICS Customization Guide*」を参照してください。

データ変換

データ変換が必要な理由

データ変換が必要ないいくつかの事例を以下に示します。

- 異なるエンコード規格 (例えば、EBCDIC と ASCII) を使用するプラットフォーム間で文字データの受け渡しを行う場合。
- 一部の文字データのエンコード方式を、あるコード化文字セット ID (CCSID) から別の CCSID に変更したい場合。

チャンネルを使用したデータ変換

チャンネルを使用してデータを交換するアプリケーションは、単純なデータ変換モデルを使用します。多くの場合、変換は必要ありませんが、変換が必要な場合は、単一のプログラミング命令を使用して、CICS に自動的にこれを処理させることができます。

以下の点に注意してください。

- 通常、(非 Java) CICS TS プログラムが別の (非 Java) CICS TS プログラムを呼び出すときは、両方のプログラムが EBCDIC エンコードを使用するので、データ

変換は必要ありません。例えば、C 言語の CICS TS プログラムが CICS TS COBOL プログラムを呼び出して文字データを保持しているいくつかのコンテナを渡す場合、データの CCSID を変更したいという特別な理由がないかぎり、データ変換を使用する必要はありません。

- チャネル・アプリケーションが使用するデータ変換モデルは、COMMAREA アプリケーションが使用するものと比べて極めて単純です。COMMAREA を使用してデータを交換するアプリケーションは、「CICS ファミリー: System/390®CICS からの通信」で説明している従来のデータ変換モデルを使用します。変換は、システム・プログラマーの制御の下で、DFHCNV 変換テーブル、DFHCCNV 変換プログラムおよび必要に応じてユーザーが置換可能な変換プログラム DFHUCNV を使用して行います。

これに対して、チャネル・コンテナ内のデータは、アプリケーション・プログラマーの制御の下で API コマンドを使用して変換されます。

- アプリケーション・プログラマーは、ユーザー・データ (自身のアプリケーション・プログラムによって作成されたコンテナ内のデータ) の変換についてのみ責任を負います。システム・データは、必要に応じて CICS によって自動的に変換されます。
- アプリケーション・プログラマーは、文字データの変換のみに関与します。(ビッグ・エンディアンとリトル・エンディアンとの間の) バイナリー・データの変換はサポートされません。
- 文字データをあるコード・ページから別のコード・ページに変換するための単純な方法として、アプリケーションでコンテナ API を使用することができます。230 ページの『コード・ページ変換を行うためのコンテナの使用』を参照してください。

データ変換のために、CICS は 2 つのタイプのデータを認識します。

- CHAR** 文字データ、すなわち、テキスト・ストリング。コンテナ内のデータは、これを取得するアプリケーションのコード・ページに (必要に応じて) 変換されます。データを取得するアプリケーションが ASCII ベースのシステムのクライアントである場合は、ASCII コード・ページに変換されます。CICS Transaction Server for z/OS アプリケーションの場合は、EBCDIC コード・ページに変換されます。

コンテナ内のすべてのデータは、単一の文字ストリングであるかのように変換されます。1 バイト文字セット (SBCS) コード・ページの場合、複数の文字フィールドで構成される構造は、1 バイト文字ストリングと等価です。ただし、2 バイト文字セット (DBCS) コード・ページの場合はそうではありません。DBCS コード・ページを使用する場合は、データ変換が適切に行われるためには、それぞれの文字ストリングを別個のコンテナに書き込む必要があります。

- BIT** すべての非文字データ。すなわち、CHAR 型として指定されていないすべてのデータ。コンテナ内のデータを変換することはできません。これはデフォルト値です。

データ変換のために使用される API コマンドは、以下のとおりです。

- EXEC CICS PUT CONTAINER [CHANNEL] [DATATYPE] [FROMCCSID]
- EXEC CICS GET CONTAINER [CHANNEL] [INTOCCSID]

PUT CONTAINER (CHANNEL) コマンドの詳細については、237 ページの『PUT CONTAINER (CHANNEL)』を参照してください。GET CONTAINER (CHANNEL) コマンドの詳細については、233 ページの『GET CONTAINER (CHANNEL)』を参照してください。

CICS で文字データを自動的に変換する方法

1. クライアント・プログラム で、PUT CONTAINER コマンドの DATATYPE(DFHVALUE(CHAR)) オプションを使用して、コンテナに文字データを格納することおよびデータが変換対象であることを指定します。以下に例を示します。

```
EXEC CICS PUT CONTAINER(cont_name) CHANNEL('payroll')
        FROM(data1) DATATYPE(DFHVALUE(CHAR))
```

データがクライアント・プラットフォームのデフォルトの CCSID でない場合を除き、FROMCCSID オプションを指定する必要はありません。(CICS TS 領域の場合、デフォルトの CCSID は、LOCALCCSID システム初期設定パラメーターで指定されます。) デフォルトの CCSID が暗黙指定されます。

2. サーバー・プログラム で、GET CONTAINER コマンドを発行して、プログラムの現行チャンネルからデータを取得します。

```
EXEC CICS GET CONTAINER(cont_name) INTO(data_area1)
```

データは、サーバー・プラットフォームのデフォルトの CCSID で戻されます。デフォルトの CCSID 以外にデータを変換する必要がある場合を除き、INTOCCSID オプションを指定する必要はありません。クライアント・プラットフォームとサーバー・プラットフォームが異なる場合、データ変換は自動的に行われます。

3. サーバー・プログラム で、PUT CONTAINER コマンドを発行して、クライアントに値を戻します。

```
EXEC CICS PUT CONTAINER(status) FROM(data_area2)
        DATATYPE(DFHVALUE(CHAR))
```

DATATYPE(DFHVALUE(CHAR)) オプションは、コンテナに文字データを格納することおよびデータが変換対象であることを指定します。データがサーバー・プラットフォームのデフォルトの CCSID でない場合を除き、FROMCCSID オプションを指定する必要はありません。

4. クライアント・プログラム で、GET CONTAINER コマンドを発行して、サーバー・プログラムから戻された状況を取得します。

```
EXEC CICS GET CONTAINER(status) CHANNEL('payroll')
        INTO(status_area)
```

The status is returned in the default CCSID of the client platform. デフォルトの CCSID 以外にデータを変換する必要がある場合を除き、INTOCCSID オプションを指定する必要はありません。クライアント・プラットフォームとサーバー・プラットフォームが異なる場合、データ変換は自動的に行われます。

コード・ページ変換を行うためのコンテナの使用

文字データのあるコード・ページから別のコード・ページに変換するための単純な方法として、アプリケーションでコンテナ API を使用することができます。以下の例では、codepage1 から codepage2 にデータを変換します。

```
EXEC CICS PUT CONTAINER(temp) DATATYPE(DFHVALUE(CHAR))
              FROMCCSID(codepage1) FROM(input-data)
EXEC CICS GET CONTAINER(temp) INTOCCSID(codepage2)
              SET(data-ptr) FLENGTH(data-len)
```

以下の例では、領域のデフォルトの EBCDIC コード・ページから、指定された UTF8 コード・ページにデータを変換します。

```
EXEC CICS PUT CONTAINER(temp) DATATYPE(DFHVALUE(CHAR))
              FROM(ebcdic-data)
EXEC CICS GET CONTAINER(temp) INTOCCSID(utf8_ccsid)
              SET(utf8-data-ptr) FLENGTH(utf8-data-len)
```

コンテナ API をこのように使用する場合は、以下のことに留意してください。

- GET CONTAINER コマンドでは、変換後の長さがわかっている場合を除き、INTO ではなく SET オプションを使用してください。(GET CONTAINER(cont_name) NODATA FLENGTH(len) コマンドを発行することにより、変換されたデータの長さを取得できます。)
- ストレージのオーバーヘッドを避けるために、変換後に変換済みデータをコピーして、コンテナを削除してください。
- チャネルのシップを避けるために、一時チャネルを使用してください。
- all-to-all 変換はできません。すなわち、指定されたコード・ページとチャネルのコード・ページがサポートされない組み合わせの場合、コード・ページ変換エラーが発生します。

SOAP の例

CICS TS SOAP アプリケーションは、以下を行います。

1. ソケットまたは MQ メッセージ・キューから UTF8 または UTF16 メッセージを取得します。
2. メッセージを UTF8 フォーマットでコンテナに書き込みます。
3. 同じチャネルの別のコンテナに EBCDIC データ構造を書き込みます。
4. バックエンドの AOR のハンドラー・プログラムに対して分散プログラム・リンク (DPL) 呼び出しを行い、チャネルを渡します。

バックエンドのハンドラー・プログラム (これも CICS TS で稼働している) は、EXEC CICS GET CONTAINER コマンドを使用して、EBCDIC データ構造またはメッセージを取得することができます。ハンドラー・プログラムは、メッセージを UTF8 または UTF16 で、あるいは自身または領域の EBCDIC コード・ページで取得することができます。同様に、ハンドラー・プログラムは EXEC CICS PUT CONTAINER コマンドを使用して、データを UTF8、UTF16、または EBCDIC でコンテナに書き込むことができます。

メッセージのうちの 1 つを領域の EBCDIC コード・ページで取得するために、ハンドラーは、以下のコマンドを発行することができます。

```
EXEC CICS GET CONTAINER(input_msg) INTO(msg)
```

INTOCCSID オプションが指定されていないため、メッセージ・データは自動的に領域の EBCDIC コード・ページに変換されます。(これは、PUT CONTAINER コマンドを使用して、DATATYPE を CHAR として指定したチャンネルにメッセージ・データを保管したことが前提となっています。DATATYPE を BIT として指定した場合 (デフォルト)、変換はできません。)

一部の出力を領域の EBCDIC コード・ページで戻すために、ハンドラーは以下のコマンドを発行することができます。

```
EXEC CICS PUT CONTAINER(output) FROM(output_msg)
```

CHAR が指定されていないため、データ変換は許可されません。FROMCCSID オプションが指定されていないため、メッセージ・データは、領域の EBCDIC コード・ページであると見なされます。

メッセージのうちの 1 つを UTF8 で取得するために、ハンドラーは、以下のコマンドを発行することができます。

```
EXEC CICS GET CONTAINER(input_msg) INTO(msg) INTOCCSID(utf8)
```

INTOCCSID オプションは、データが領域の EBCDIC コード・ページに自動的に変換されないようにするために必要です。

一部の出力を UTF8 で戻すために、サーバー・プログラムは以下のコマンドを発行することができます。

```
EXEC CICS PUT CONTAINER(output) FROM(output_msg) FROMCCSID(utf8)
```

FROMCCSID オプションは、メッセージ・データが現在は UTF8 フォーマットであることを指定します。CHAR が指定されていないため、データ変換は許可されません。

要件

この機能をサポートするための特別なハードウェア要件およびソフトウェア要件はありません。

関連情報

377 ページの『第 27 章 CICS 稼働環境』

CICS 外部インターフェースの変更点

アプリケーション・プログラミング・インターフェースの変更点

新しいコマンド

コンテナおよびチャンネルを処理する 5 つの新規のコマンドがあります。詳しくは、以下を参照してください。

232 ページの『DELETE CONTAINER (CHANNEL)』

233 ページの『GET CONTAINER (CHANNEL)』

235 ページの『MOVE CONTAINER (CHANNEL)』

237 ページの『PUT CONTAINER (CHANNEL)』

変更されたコマンド

以下のアプリケーションのプログラミング・コマンドが変更されました。

```
EXEC CICS DELETE CONTAINER (BTS)
EXEC CICS ENDBROWSE CONTAINER
EXEC CICS GET CONTAINER (BTS)
EXEC CICS GETNEXT CONTAINER
EXEC CICS HANDLE ABEND
EXEC CICS LINK PROGRAM
EXEC CICS MOVE CONTAINER (BTS)
EXEC CICS PUT CONTAINER (BTS)
EXEC CICS RETURN
EXEC CICS STARTBROWSE CONTAINER
EXEC CICS XCTL
```

詳細については、243 ページの『変更された API コマンド』を参照してください。

新規 API コマンド

以下の新規アプリケーション・プログラミング・コマンドが導入されました。

- EXEC CICS DELETE CONTAINER (CHANNEL)
- EXEC CICS GET CONTAINER (CHANNEL)
- EXEC CICS MOVE CONTAINER (CHANNEL)
- EXEC CICS PUT CONTAINER (CHANNEL)
- EXEC CICS START CHANNEL

DELETE CONTAINER (CHANNEL):

指定されたチャネル・コンテナを削除します。

DELETE CONTAINER (CHANNEL)

▶▶—DELETE—CONTAINER(*data-value*)—▶▶
└──────────CHANNEL(*data-value*)──────────┘

条件: CHANNELERR、CONTAINERERR、INVREQ

このコマンドはスレッド・セーフです。

説明

DELETE CONTAINER (CHANNEL) は、チャネルからコンテナを削除し、コンテナに含まれているすべてのデータを廃棄します。

コンテナは、名前およびコンテナのチャンネル (コンテナを「所有」しているチャンネル) によって識別されます。コンテナを所有しているチャンネルは、以下のように識別されます。

- CHANNEL オプションを指定することにより明示的に。
- CHANNEL オプションを省略することにより暗黙的に。チャンネルを省略した場合、現行チャンネルが暗黙指定されます。

オプション

CHANNEL(data-value)

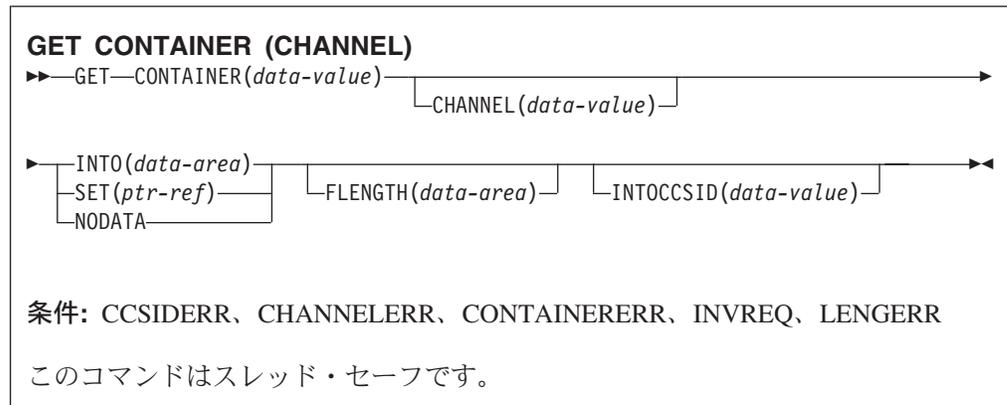
コンテナを所有しているチャンネルの名前 (1 から 16 文字) を指定します。

CONTAINER(data-value)

削除するコンテナの名前 (1 から 16 文字) を指定します。

GET CONTAINER (CHANNEL):

指定されたチャンネル・コンテナからデータを取得します。



説明

GET CONTAINER (CHANNEL) は、指定されたチャンネル・コンテナに関連付けられたデータを読み取ります。

データを保持しているコンテナは、名前およびコンテナのチャンネル (コンテナを「所有」しているチャンネル) によって識別されます。コンテナを所有しているチャンネルは、以下のように識別されます。

- CHANNEL オプションを指定することにより明示的に。
- CHANNEL オプションを省略することにより暗黙的に。チャンネルを省略した場合、現行チャンネルが暗黙指定されます。

オプション

CHANNEL(data-value)

コンテナを所有しているチャンネルの名前 (1 から 16 文字) を指定します。

CONTAINER(data-value)

取得するデータを保持するコンテナの名前 (1 から 16 文字) を指定します。

FLENGTH(data-area)

入力フィールドとしては、FLENGTH は、読み取るデータの長さをフルワード・バイナリー値で指定します。出力フィールドとしては、FLENGTH は、コンテナ内のデータの長さを戻します。FLENGTH が入力フィールドであるか出力フィールドであるかは、INTO、SET、または NODATA オプションのいずれを指定するかによって決まります。

INTO オプションを指定した場合

FLENGTH は入力フィールドと出力フィールドの両方です。

入力時は、FLENGTH はプログラムが受け入れるデータの最大長を指定します。指定された値がゼロより小さい場合は、ゼロが指定されたと思なされます。データの長さが指定された値を超える場合、データは指定された値まで切り詰められ、LENGERR 条件が発生します。

INTO 変数からコンパイラーによって長さを生成できる場合は、FLENGTH を指定する必要はありません。INTO および FLENGTH の両方が指定された場合、FLENGTH はプログラムが受け入れるデータの最大長を指定します。

出力時 (すなわち、取得操作の完了時) に、CICS は、データ域が指定されている場合は、これをコンテナ内のデータの実際の長さに設定します。ある CCSID から別の CCSID に変換された文字データがコンテナ内にある場合、これは、変換後のデータの長さです。

SET または NODATA オプションを指定した場合

FLENGTH は出力専用フィールドです。データ域として指定する必要があります。

取得操作の完了時、データ域は、コンテナ内のデータの実際の長さに設定されます。ある CCSID から別の CCSID に変換された文字データがコンテナ内にある場合、これは、変換後のデータの長さです。

INTO(data-area)

取得済みデータを入れるデータ域を指定します。

INTOCCSID(data-value)

コンテナ内の文字データの変換先のコード化文字セット ID (CCSID) をフルワード 10 進数で指定します。CICS Transaction Server for z/OS アプリケーションの場合、これは通常は EBCDIC CCSID です。(ただし、例えば、ASCII データを自動的に EBCDIC に変換せずに取得したい場合は、ASCII CCSID を指定することができます。)

INTOCCSID を指定しなかった場合、値はその領域の CCSID にデフォルトで設定されます。領域のデフォルトの CCSID は、LOCALCCSID システム初期設定パラメーターで指定されます。

変換できるのは文字データのみであり、コンテナへのデータの書き込みに使用される PUT CONTAINER コマンドで DATATYPE を CHAR に指定した場合だけです。

チャンネルを使用したデータ変換の詳細については、「*CICS Application Programming Guide*」を参照してください。

CCSID の解説および、CICS でサポートされる CCSID のリストについては、「*CICS ファミリー: System/390CICS からの通信*」を参照してください。

NODATA

取得するデータがないことを指定します。このオプションを使用し、コンテナ内のデータの長さ (FLENGTH に戻される) を検出します。

データ変換が行われた場合、文字データの長さが変わる場合があります。したがって、この領域の CCSID 以外の CCSID に文字データを変換する場合は、NODATA を指定するときに INTOCCSID も指定する必要があります。これにより、変換済みデータが正しい長さで FLENGTH に戻されることが保証されます。

SET(ptr-ref)

取得済みデータのアドレスを戻すデータ域を指定します。

データ域は、以下のいずれかが発生するまで、CICS によって自動的に保守されます。

- このストレージにアクセスできる任意のプログラムが、同じチャンネル内の同じコンテナに対して SET オプションを指定した後続の GET CONTAINER コマンドを発行した場合。
- DELETE CONTAINER コマンドによってコンテナが削除された場合。
- MOVE CONTAINER コマンドによってコンテナが移動された場合。
- チャンネルが、プログラムのスコープ外となった場合。

他のプログラムへのリンクによって、上記コマンドのいずれかが発行される可能性があることに注意してください。

このストレージを解放するために、FREEMAIN コマンドを発行しないでください。

アプリケーションでデータを保持しておく必要がある場合は、独自のストレージにデータを移動してください。

MOVE CONTAINER (CHANNEL):

あるチャンネルから別のチャンネルへコンテナ (およびその内容) を移動します。

MOVE CONTAINER (CHANNEL)

```
▶▶—MOVE—CONTAINER(data-value)—AS(data-value)—┐
└──────────────────────────────────────────┘
└──────────────────────────────────┘
CHANNEL(data-value)
└──────────────────────────────────────────┘
└──────────────────────────────────────────┘
└──────────────────────────────────┘
TOCHANNEL(data-value)
```

条件: CHANNELERR、CONTAINERERR、INVREQ

このコマンドはスレッド・セーフです。

説明

MOVE CONTAINER (CHANNEL) は、あるチャンネルから別のチャンネルにコンテナを移動します。移動後、ソース・コンテナは存在しなくなります。

ソース・コンテナおよびターゲット・コンテナは、名前およびそのコンテナを所有しているチャンネルによって識別されます。ソース・コンテナを所有しているチャンネルは、以下のように識別されます。

- CHANNEL オプションを指定することにより明示的に。
- CHANNEL オプションを省略することにより暗黙的に。チャンネルを省略した場合、現行チャンネルが暗黙指定されます。

同様に、ターゲット・コンテナを所有しているチャンネルは、以下のように識別されます。

- TOCHANNEL オプションを指定することにより明示的に。
- TOCHANNEL オプションを省略することにより暗黙的に。チャンネルを省略した場合、現行チャンネルが暗黙指定されます。

コンテナは、以下のように移動できます。

- あるチャンネルから別のチャンネルへ。
- 同じチャンネル内で。例えば、現行チャンネルから現行チャンネルへ。これには、コンテナを名前変更するという効果があります。

チャンネル間でより効率的にデータを転送する方法として、GET CONTAINER および PUT CONTAINER の代わりに MOVE CONTAINER を使用できます。

注:

1. ソース・チャンネルは、MOVE CONTAINER コマンドを発行するプログラムのスコープ内でなければなりません。
2. MOVE CONTAINER コマンドを発行するプログラムのスコープ内にターゲット・チャンネルが存在しない場合は、ターゲット・チャンネルが作成されません。
3. ソース・コンテナが存在しない場合は、エラーが発生します。

4. ターゲット・コンテナがまだ存在しない場合は、これが作成されます。ターゲット・コンテナが既に存在する場合は、直前の内容が上書きされません。
5. あるコンテナをそのコンテナ自身で上書きしようとした場合は、何も処理は行われません。すなわち、CONTAINER オプションと AS オプションに同じ値を指定し、CHANNEL オプションと TOCHANNEL オプションを両方とも省略した場合、あるいはこれらに同じ値を渡して同じチャンネルを指定した場合、ソース・コンテナは変更も削除もされません。エラー状態は発生しません。

オプション

AS(data-value)

ターゲット・コンテナの名前 (1 から 16 文字) を指定します。ターゲット・コンテナが既に存在する場合は、その内容が上書きされます。

許容文字は A-Z a-z 0-9 \$ @ # / % & ? ! : | " = ~ , ; < > . - および _ です。先行空白文字および埋め込まれた空白文字は許可されません。提供された名前が 16 文字未満の場合は、16 文字になるように末尾空白で埋め込まれます。

コンテナ名は常に EBCDIC です。上記に示したコンテナ名の許容文字セットには、一部の EBCDIC コード・ページには同じ表現が存在しないいくつかの文字が含まれています。したがって、領域間でコンテナをシップする場合は、命名で使用する文字を A-Z 0-9 & : = , ; < > . - および _ に制限することを推奨します。

CHANNEL(data-value)

ソース・コンテナを所有しているチャンネルの名前 (1 から 16 文字) を指定します。このオプションを指定しなかった場合、現行チャンネルが暗黙指定されます。

CONTAINER(data-value)

移動するソース・コンテナの名前 (1 から 16 文字) を指定します。

TOCHANNEL(data-value)

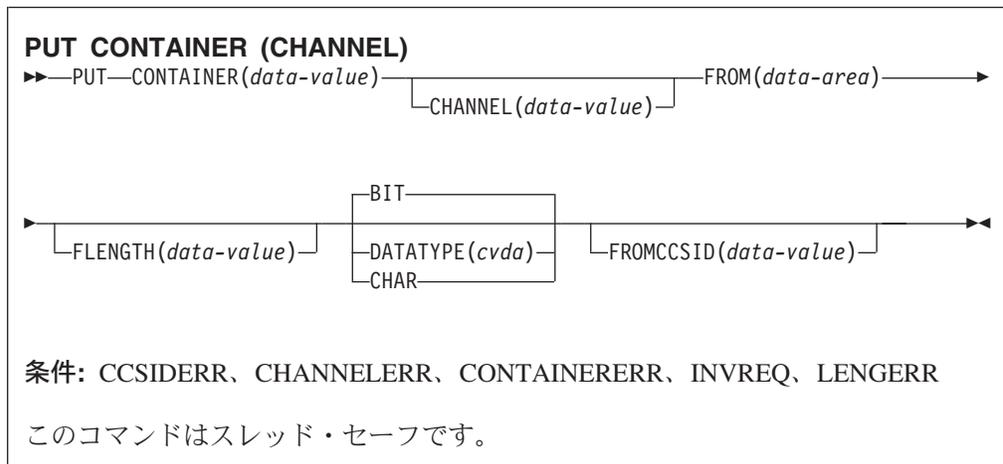
ターゲット・コンテナを所有しているチャンネルの名前 (1 から 16 文字) を指定します。新規チャンネルを指定する場合、許容文字は A-Z a-z 0-9 \$ @ # / % & ? ! : | " = ~ , ; < > . - および _ であることに注意してください。先行空白文字および埋め込まれた空白文字は許可されません。提供された名前が 16 文字未満の場合は、16 文字になるように末尾空白で埋め込まれます。

チャンネル名は常に EBCDIC です。上記に示したチャンネル名の許容文字セットには、一部の EBCDIC コード・ページには同じ表現が存在しないいくつかの文字が含まれています。したがって、領域間でチャンネルをシップする場合は、命名で使用する文字を A-Z 0-9 & : = , ; < > . - および _ に制限することを推奨します。

このオプションを指定しなかった場合、現行チャンネルが暗黙指定されます。

PUT CONTAINER (CHANNEL):

指定されたチャンネル・コンテナにデータを書き込みます。



説明

PUT CONTAINER (CHANNEL) は、指定されたチャンネルに関連付けられたコンテナにデータを書き込みます。

コンテナは名前で識別されます。コンテナを所有しているチャンネルは、以下のように識別されます。

- CHANNEL オプションを指定することにより明示的に。
- CHANNEL オプションを省略することにより暗黙的に。チャンネルを省略した場合、現行チャンネルが暗黙指定されます。

注:

1. チャンネルに関連付けることができるコンテナの数に制限はありません。
2. 個々のコンテナのサイズは、使用可能なストレージの量によってのみ制限されます。

注意:

大容量のコンテナを多数作成して、他のアプリケーションが使用可能なストレージの量を制限してしまわないように注意してください。

3. 指定されたコンテナがまだ存在しない場合は、これが作成されます。指定されたコンテナが既に存在する場合は、直前の内容が上書きされます。
4. 指定されたチャンネルがまだ存在しない場合は、これが作成されます。

オプション

CHANNEL(data-value)

コンテナを所有しているチャンネルの名前 (1 から 16 文字) を指定します。許容文字は A-Z a-z 0-9 \$ @ # / % & ? ! : | " = , ; < > . - および _ です。先行空白文字および埋め込まれた空白文字は許可されません。提供された名前が 16 文字未満の場合は、16 文字になるように末尾空白で埋め込まれます。

チャンネル名は常に EBCDIC です。上記に示したチャンネル名の許容文字セットには、一部の EBCDIC コード・ページには同じ表現が存在しないいくつかの文字

が含まれています。したがって、領域間でチャンネルをシッパする場合は、命名で使用する文字を A-Z 0-9 & : = , ; < > . - および _ に制限することを推奨します。

CONTAINER(data-value)

データの書き込み先のコンテナの名前 (1 から 16 文字) を指定します。

許容文字は A-Z a-z 0-9 \$ @ # / % & ? ! : | " = , ; < > . - および _ です。先行空白文字および埋め込まれた空白文字は許可されません。提供された名前が 16 文字未満の場合は、16 文字になるように末尾空白で埋め込まれます。

CICS によって要求された場合以外は、『DFH』で始まるコンテナ名を使用しないでください。

コンテナ名は常に EBCDIC です。上記に示したコンテナ名の許容文字セットには、一部の EBCDIC コード・ページには同じ表現が存在しないいくつかの文字が含まれています。したがって、領域間でコンテナをシッパする場合は、命名で使用する文字を A-Z 0-9 & : = , ; < > . - および _ に制限することを推奨します。

DATATYPE(cvda)

コンテナに書き込むデータのタイプを指定します。このオプションは、新しいコンテナにのみ適用されます。コンテナが既に存在している場合、コンテナのデータ・タイプは作成時に設定されているため、変更できません。CVDA 値は、以下のとおりです。

BIT ビット・データ。コンテナ内のデータを変換することはできません。FROMCCSID が指定されていない限り、これがデフォルト値です。

CHAR 文字データ。コンテナ内のデータは、チャンネルを作成したアプリケーションのコード・ページに (必要に応じて) 変換されます。ASCII ベースのシステムのクライアント・アプリケーションによってチャンネルが作成された場合は、ASCII コード・ページに変換されます。CICS Transaction Server for z/OS アプリケーションによって作成された場合は、EBCDIC コード・ページになります。変換は、クライアント・プログラムとサーバー・プログラムが異なるプラットフォームで実行されている場合にのみ必要です。

コンテナ内のすべてのデータは、単一の文字ストリングであるかのように変換されます。SBCS コード・ページの場合、複数の文字フィールドで構成される構造は、1 バイト文字ストリングと等価です。ただし、DBCS コード・ページの場合はそうではありません。DBCS コード・ページを使用する場合は、データ変換が適切に行われるためには、それぞれの文字ストリングを別個のコンテナに書き込む必要があります。

コンテナに文字データが格納され、かつ、チャンネルが CICS Transaction Server for z/OS から ASCII システムに渡される場合は、DATATYPE を CHAR に指定する必要があります。コンテナにバイナリー・データが含まれている場合、あるいはチャンネルが ASCII システムに渡されない場合は、DATATYPE はオプション・パラメーターです。

PUT CONTAINER コマンドを使用して、既存のコンテナのデータ・タイプを変更することはできません。例えば、コンテナがデータ・タイプ BIT で作成

されているときに、後続の PUT CONTAINER コマンドで同じコンテナに対してデータ・タイプ CHAR を指定した場合、INVREQ 条件が発生します。既存のコンテナを別のデータ・タイプのコンテナで置き換える必要がある場合は、まず既存のコンテナを明示的に削除する必要があります。

チャンネルを使用したデータ変換の詳細については、「*CICS Application Programming Guide*」を参照してください。

FLENGTH(data-value)

データを読み取るデータ域の長さをフルワード・バイナリー値で指定します。

FROM(data-area)

データをコンテナに書き込む際の書き込み元のデータ領域を指定します。

FROMCCSID(data-value)

コンテナに書き込む文字データの現在のコード化文字セット ID (CCSID) をフルワード 10 進数で指定します。CICS Transaction Server for z/OS アプリケーションの場合、これは通常 EBCDIC CCSID になります。(ただし、例えば、ASCII データを渡したい場合は、ASCII CCSID を指定することができます。)

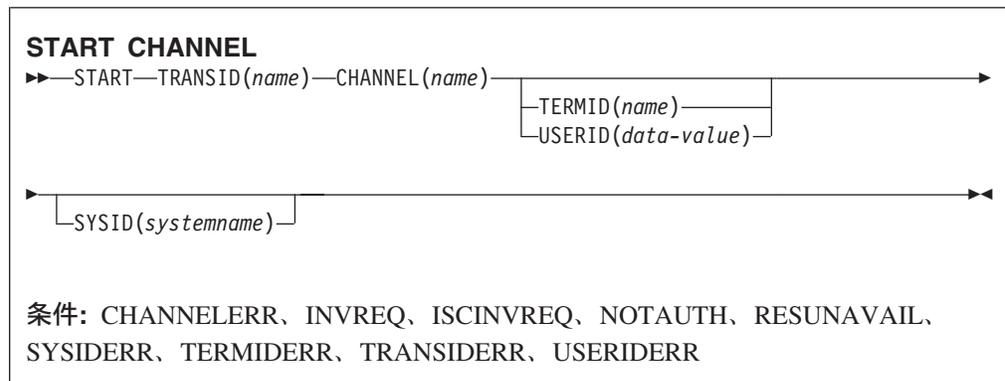
FROMCCSID を指定した場合は、DATATYPE(DFHVALUE(CHAR)) が暗黙指定されます。

FROMCCSID を指定しなかった場合、値はその領域の CCSID にデフォルトで設定されます。領域のデフォルトの CCSID は、LOCALCCSID システム初期設定パラメーターで指定されます。

CCSID の解説、および CICS でサポートされる CCSID のリストについては、「*CICS ファミリー: System/390CICS からの通信*」を参照してください。

START CHANNEL:

タスクを開始し、このタスクにチャンネルを渡します。



説明

START CHANNEL は、ローカルまたはリモート・システムでタスクを開始し、このタスクにチャンネルを渡します。

一般に、開始側タスクは、チャンネルを使用して、開始されたタスクにデータを渡します (ただし、場合によってはチャンネルが空のこともあります。CHANNEL オプシ

ョンの説明を参照してください。) 開始側タスクは、開始されたタスクで基本機能として使用する端末を指定することもできます。

開始されたタスクは、例えば、以下を行うことができます。

1. ASSIGN CHANNEL コマンドを使用して、渡されたチャンネルの名前を発見する
2. STARTBROWSE CONTAINER CHANNEL および GETNEXT CONTAINER コマンドを使用して、チャンネル内のコンテナをブラウズする
3. GET CONTAINER CHANNEL コマンドを使用して、コンテナ内のデータにアクセスする

実行するトランザクションを開始できるようにするために、以下に示すいくつかの制約を満たす必要があります。

- TERMID オプションを指定した場合、指定された端末が存在し、使用可能でなければなりません。指定された端末が存在しない場合、START は廃棄されます。
- START CHANNEL は IMS をサポートしません。すなわち、START CHANNEL を使用してリモートの IMS システムでトランザクションを開始することはできません。

各 START CHANNEL コマンドは、結果として別々のタスクを開始します。

START コマンドによって開始される動的にルーティングされるトランザクション

START コマンドのサブセットによって開始される一部のトランザクションは、動的にリモート領域にルーティングすることができます。

例外条件なしの START 障害

START コマンドはエラーなしで実行されるにもかかわらず、開始済みトランザクションが発生しない、以下のような状況があります。

- CICS がタスクの作成を試行したときに、トランザクションまたはその初期プログラムが使用不可であった場合。
- CICS がタスクの作成を試行したときに、START で指定された端末が定義されていなかった (そして、XICTENF または XALTENF 出口がこの端末を見つけることができなかった) 場合。
- START 時に、要求された端末が存在しなかった場合は、TERMIDERR 状況を受け取ります。ただし、端末が後で使用不可になった (ユーザーがログオフした) 場合、START 要求は廃棄され、TERMIDERR は発生しません。

これらの障害は、START の実行とタスク作成時の間に遅延がある場合に発生します。START が常に即時に行われる START CHANNEL 要求の場合にも、必要な端末が空いていないためまたはその他のシステム制約により、CICS でタスク作成の遅延が発生する場合があります。

INQUIRE コマンドを使用して、START コマンドの実行時にトランザクションおよびプログラムが使用可能であることを確認することはできますが、これらはどちらもタスク作成時に使用不可になる可能性があります。

オプション

CHANNEL(*name*)

開始されたタスクで使用可能なチャンネルの名前 (1 から 16 文字) を指定します。許容文字は A-Z a-z 0-9 \$ @ # / % & ? ! : | " = ~ , ; < > . - および _ です。先行空白文字および埋め込まれた空白文字は許可されません。提供された名前が 16 文字未満の場合は、16 文字になるように末尾空白で埋め込まれます。

チャンネル名は常に EBCDIC です。上記に示したチャンネル名の許容文字セットには、一部の EBCDIC コード・ページには同じ表現が存在しないいくつかの文字が含まれています。したがって、領域間でチャンネルをシッップする場合は、命名で使用する文字を A-Z 0-9 & : = , ; < > . - および _ に制限することを推奨します。

START コマンドを発行するプログラムは、以下の可能性があります。

- 1 つ以上の PUT CONTAINER CHANNEL コマンドを使用してチャンネルを作成済みである
- 現行チャンネルを名前で指定している
- 存在しないチャンネルを指定している (この場合は新たに空のチャンネルが作成されます)

開始されたタスクには、チャンネルのコンテナ (およびその中に含まれているデータ) のコピー が渡されます。コピーは、START コマンドの発行時に作成されます。

SYSID(*systemname*)

要求の送信先システムの名前を指定します。

TERMID(*name*)

START コマンドの結果として開始されるトランザクションと関連付けられた基本機能のシンボリック ID (1 から 4 文字の英数字) を指定します。この基本機能は、端末 (通常の場合) または APPC セッションです。APPC セッションを指定した場合は、端末 ID の代わりに接続 (またはモードセット) 名が使用されます。このオプションは、開始されるトランザクションが端末と通信しなければならない場合に必要になります。その他の場合は、省略してください。

端末 ID は、START コマンドを実行するシステムのローカルまたはリモート端末として定義する必要があります。

TRANSID(*name*)

START コマンドの結果として開始されるタスクによって実行されるトランザクションのシンボリック ID (1 から 4 文字) を指定します。

SYSID を指定し、リモート・システムを指定した場合、トランザクション定義が PCT でリモートとして定義されているかどうかに関係なく、トランザクションはそのシステム上に存在すると見なされます。それ以外の場合は、トランザクション定義を使用して、トランザクションがローカル・システムとリモート・システムのどちらに存在するかが調べられます。

USERID(*data-value*)

開始されたトランザクションが端末と関連付けられていない (すなわち、TERMID が指定されていない) 場合、開始されたトランザクションを実行する権限を所有するユーザー ID を指定します。これを、*userid1* と呼びます。

TERMID および USERID のどちらも省略した場合、CICS では、この代わりに、START コマンドを発行するトランザクションを実行するユーザー ID を使用します。これを、*userid2* と呼びます。

CICS は、*userid1* または *userid2* を使用して、開始されたトランザクションが常に (開始されたトランザクションによって参照されるすべてのリソースに対する許可を持つ) 有効なユーザー ID で実行されることを保証します。

CICS は、*userid2* に対して代理セキュリティチェックを実行して、このユーザーが *userid1* に対して許可されていることをチェックします。*userid2* が許可されていない場合、CICS は NOTAUTH 条件を戻します。USERID を省略した場合、ここでは代理チェックは行われません。

変更された API コマンド

以下のアプリケーションのプログラミング・コマンドが変更されました。

EXEC CICS ASSIGN

CHANNEL オプションが追加され、STARTCODE オプションに以下の変更が行われました。

オプション

CHANNEL(*data-area*)

プログラムの現行チャンネルが存在する場合、現行チャンネルの 16 文字の名前を戻します。存在しない場合は、ブランクです。

STARTCODE(*data-area*)

要求を出したトランザクションがどのように開始されたかを示す 2 文字の値を戻します。以下の値が変更されました。

コード トランザクションを開始したコマンド

S FROM オプションにデータを渡さなかった START コマンド。
チャンネルが渡されている場合も、渡されていない場合もあります。

SD FROM オプションにデータを渡した START コマンド。

EXEC CICS DELETE CONTAINER (BTS)

構文に変更はありません。コマンドが BTS のコンテナのみに適用され、チャンネルのコンテナには適用されないことを強調するために、説明が変更されました。

EXEC CICS ENDBROWSE CONTAINER

構文に変更はありませんが、BTS のコンテナだけでなく、チャンネルのコンテナでもこのコマンドを使用できるようになりました。

EXEC CICS GET CONTAINER (BTS)

INVREQ 条件に新規 RESP2 値が追加されました。

条件

INVREQ

RESP2 値は以下のとおりです。

2 CHANNEL オプションを指定せずに INTOCCSID オプションが指定されましたが、(コマンドを発行したプログラムが現行チャ

ネルを渡さなかったため) 現行チャンネルがありません。
INTOCCSID は、チャンネルを (明示的または暗黙的に) 指定する
GET CONTAINER コマンドでのみ有効です。これは、GET
CONTAINER (BTS) コマンドでは無効です。

EXEC CICS GETNEXT CONTAINER

構文に変更はありませんが、BTS のコンテナーだけでなく、チャンネルのコンテナーでもこのコマンドを使用できるようになりました。

EXEC CICS HANDLE ABEND

現行チャンネルが存在する場合、異常終了を処理するように指定されたプログラムに現行チャンネルが渡されます。あるいは、通信域 (COMMAREA) が確立されている場合は、これが渡されます。従来は、COMMAREA のみを渡すことができました。

EXEC CICS LINK PROGRAM

CHANNEL オプションおよび CHANNELERR 条件が追加されました。

オプション

CHANNEL(*name*)

呼び出されたプログラムで使用可能なチャンネルの名前 (1 から 16 文字) を指定します。許容文字は A-Z a-z 0-9 \$ @ # / % & ? ! : | " = ~ , ; < > . - および _ です。先行空白文字および埋め込まれた空白文字は許可されません。提供された名前が 16 文字未満の場合は、16 文字になるように末尾空白で埋め込まれます。

チャンネル名は常に EBCDIC です。上記に示したチャンネル名の許容文字セットには、一部の EBCDIC コード・ページには同じ表現が存在しないいくつかの文字が含まれています。したがって、領域間でチャンネルをシップする場合は、命名で使用する文字を A-Z 0-9 & : = , ; < > . - および _ に制限することを推奨します。

LINK コマンドを発行するプログラムは、以下の可能性があります。

- 1 つ以上の PUT CONTAINER CHANNEL コマンドを使用してチャンネルを作成済みである
- 現行チャンネルを名前で指定している
- 存在しないチャンネルを指定している (この場合は新たに空のチャンネルが作成されます)

条件

CHANNELERR

RESP2 値は以下のとおりです。

- 1 CHANNEL オプションで指定された名前には、正しくない文字または文字の組み合わせが含まれています。

EXEC CICS MOVE CONTAINER (BTS)

構文に変更はありません。コマンドが BTS のコンテナーのみに適用され、CHANNEL のコンテナーには適用されないことを強調するために、説明が変更されました。

EXEC CICS PUT CONTAINER (BTS)

INVREQ 条件に 2 つの新規 RESP2 値が追加されました。

条件

INVREQ

RESP2 値は以下のとおりです。

- 1 CHANNEL オプションを指定せずに DATATYPE オプションが指定されましたが、(コマンドを発行したプログラムが現行チャンネルを渡さなかったため) 現行チャンネルがありません。
DATATYPE は、チャンネルを (明示的または暗黙的に) 指定する PUT CONTAINER コマンドでのみ有効です。これは、PUT CONTAINER (BTS) コマンドでは無効です。
- 2 CHANNEL オプションを指定せずに FROMCCSID オプションが指定されましたが、(コマンドを発行したプログラムが現行チャンネルを渡さなかったため) 現行チャンネルがありません。
FROMCCSID は、チャンネルを (明示的または暗黙的に) 指定する PUT CONTAINER コマンドでのみ有効です。これは、PUT CONTAINER (BTS) コマンドでは無効です。

EXEC CICS RETURN

CHANNEL オプションおよび CHANNELERR 条件が追加されました。

オプション

CHANNEL(*name*)

制御を受け取る次のプログラムで使用可能なチャンネルの名前 (1 から 16 文字) を指定します。許容文字は A-Z a-z 0-9 \$ @ # / % & ? ! : | " = , ; < > . - および _ です。先行空白文字および埋め込まれた空白文字は許可されません。提供された名前が 16 文字未満の場合は、16 文字になるように末尾空白で埋め込まれます。

チャンネル名は常に EBCDIC です。上記に示したチャンネル名の許容文字セットには、一部の EBCDIC コード・ページには同じ表現が存在しないいくつかの文字が含まれています。したがって、領域間でチャンネルをシップする場合 (つまり、TRANSID オプションで指定されたトランザクションがリモートの場合) は、命名で使用する文字を A-Z 0-9 & : = , ; < > . - および _ に制限することを推奨します。

RETURN コマンドを発行するプログラムは、以下の可能性があります。

- 1 つ以上の PUT CONTAINER CHANNEL コマンドを使用してチャンネルを作成済みである
- 現行チャンネルを名前指定している
- 存在しないチャンネルを指定している (この場合は新たに空のチャンネルが作成されます)

このオプションは、最上位のロジック・レベルのプログラム (すなわち、CICS に制御を戻すプログラム) によって発行された RETURN コマンドでのみ有効です。

条件

CHANNELERR

RESP2 値は以下のとおりです。

- 1 CHANNEL オプションで指定された名前には、正しくない文字または文字の組み合わせが含まれています。

EXEC CICS STARTBROWSE CONTAINER

CHANNEL オプションおよび CHANNELERR 条件が追加されました。

オプション

CHANNEL(*data-value*)

ブラウズするコンテナを所有しているチャンネルの名前 (1 から 16 文字) を指定します。これは、現行チャンネルの名前、または STARTBROWSE CONTAINER コマンドを発行したプログラムによって作成されたチャンネルの名前でなければなりません。

このオプションが指定されていない場合、現在のコンテキストがチャンネルであるときは、現行チャンネルのコンテナがブラウズされます。

条件

CHANNELERR

RESP2 値は以下のとおりです。

- 2 CHANNEL オプションで指定されたチャンネルが見つかりませんでした。

EXEC CICS XCTL

CHANNEL オプションおよび CHANNELERR 条件が追加されました。

オプション

CHANNEL(*name*)

呼び出されたプログラムで使用可能なチャンネルの名前 (1 から 16 文字) を指定します。許容文字は A-Z a-z 0-9 \$ @ # / % & ? ! : | " = ~ , ; < > . - および _ です。先行空白文字および埋め込まれた空白文字は許可されません。提供された名前が 16 文字未満の場合は、16 文字になるように末尾空白で埋め込まれます。

チャンネル名は常に EBCDIC です。上記に示したチャンネル名の許容文字セットには、一部の EBCDIC コード・ページには同じ表現が存在しないいくつかの文字が含まれています。したがって、領域間でチャンネルをシップする場合は、命名で使用する文字を A-Z 0-9 & : = , ; < > . - および _ に制限することを推奨します。

XCTL コマンドを発行するプログラムは、以下の可能性があります。

- 1 つ以上の PUT CONTAINER CHANNEL コマンドを使用してチャンネルを作成済みである
- 現行チャンネルを名前で指定している
- 存在しないチャンネルを指定している (この場合は新たに空のチャンネルが作成されます)

条件

CHANNELERR

RESP2 値は以下のとおりです。

- 1 CHANNEL オプションで指定された名前には、正しくない文字または文字の組み合わせが含まれています。

JCICS API の変更点

注: CICS Enterprise Bean を作成するときは、JCICS コマンド (チャンネルおよびコンテナ関連のコマンドを含む) を使用できます。ただし、CICS では、IIOP 要求ストリームを介したチャンネルの送信はサポートしません。これは、例えば、リモート領域上の Enterprise Bean にチャンネルを渡すことができないことを意味します。

新規 JCICS クラス

以下の新規 JCICS クラスが導入されました。

- `com.ibm.cics.server.CCSIDErrorException`
- `com.ibm.cics.server.Channel`
- `com.ibm.cics.server.ChannelErrorException`
- `com.ibm.cics.server.Container`
- `com.ibm.cics.server.ContainerErrorException`
- `com.ibm.cics.server.ContainerIterator`

変更された JCICS クラス

以下の JCICS クラスが変更されました。

- `Program`
- `StartRequest`
- `Task`
- `TerminalPrincipalFacility`

グローバル・ユーザー出口ルーチンの変更点

グローバル・ユーザー出口ルーチン・プログラムは、アプリケーション・プログラムによって作成されたコンテナにアクセスできません。ただし、独自のチャンネルを作成して、呼び出し先プログラムにこれを渡すことができます。

以下の出口ルーチンに対する小さな変更については、「*CICS Customization Guide*」に説明があります。

- XFCAREQ
- XFCAREQC
- XFCREQ
- XFCREQC
- XICEREQ
- XICEREQC
- XNQEREQ
- XNQEREQC
- XPCREQ
- XPCREQC
- XTDEREQ
- XTDEREQC
- XTSEREQ
- XTSEREQC

タスク関連ユーザー出口の変更点

タスク関連ユーザー出口プログラム (TRUE) は、アプリケーション・プログラムによって作成されたコンテナにアクセスできません。ただし、独自のチャンネルを作成して、呼び出し先プログラムにこれを渡すことができます。

ユーザーが置換可能なプログラムの変更点

ユーザーが置換可能なプログラムは、アプリケーション・コードによって作成されたコンテナにアクセスできません。ただし、独自のチャンネルを作成して、呼び出し先プログラムにこれを渡すことができます。

動的および分散ルーティング・プログラム

動的および分散ルーティング・プログラムに渡される DFHDYPDS 通信域に対していくつかの変更が行われました。

DYRACMAA (既存のフィールド: 動的ルーティング・プログラム)

このフィールドは、以下の要求のルーティングに適用されます。

- 端末開始トランザクション
- 端末関連の START コマンドによって開始されるトランザクション
- プログラム・リンク (DPL) 要求

これらのタイプの要求のルーティングの場合、DYRACMAA に以下のいずれかが格納されます。

- ユーザー・アプリケーションが通信域 (COMMAREA) を使用する場合は、アプリケーションの COMMAREA の 31 ビットのアドレス
- ユーザー・アプリケーションがチャンネルを使用し、チャンネル内に DFHROUTE というコンテナを作成している場合は、DFHROUTE コンテナの 31 ビットのアドレス
- ユーザー・アプリケーションに COMMAREA がなく、DFHROUTE コンテナもない場合は、ヌル文字

その他のすべてのタイプの要求のルーティングでは、DYRACMAA にヌル文字が格納されます。

上記の 3 つのタイプの対象となる要求のルーティングでは、ユーザー・アプリケーションが COMMAREA を使用する場合、以下のアドレスが格納されます。

- 動的ルーティング・プログラムがルーティングに対して呼び出される場合 (DYRFUNC=0)、アドレスは、入力通信域 (存在する場合) のアドレスになります。同様に、ルーティング・プログラムがルート選択エラーのため、または通知の目的で (それぞれ、DYRFUNC=1 および 3) 呼び出される場合、アドレスは入力通信域のアドレスになります。
- 以前にルーティングしたトランザクションまたはリンク要求が正常に終了したためにルーティング・プログラムが呼び出される場合 (DYRFUNC=2) は、アドレスは出力通信域 (存在する場合) のアドレスになります。ルーティングされたアプリケーションは、そのアプリケーションの出力通信域を使用して、動的ルーティング・プログラムに情報を渡します。

ルーティングされたトランザクションが異常終了したためにルーティング・プログラムが呼び出される場合 (DYRFUNC=4)、通信域または DFHROUTE コンテナ内の情報は、意味のある情報ではありません。

ルーティング・プログラムは、DYRACMAA でアドレス指定されたアプリケーションの通信域または DFHROUTE コンテナ内のデータを変更することができます。

DYRACMAA (既存のフィールド: 分散ルーティング・プログラム (変更なし))

このフィールドは、分散ルーティング・プログラムでは使用されません。このフィールドは、呼び出し時にゼロに設定されます。

DYRACMAL (既存のフィールド: 動的ルーティング・プログラム)

このフィールドは、以下の要求のルーティングに適用されます。

- 端末開始トランザクション
- 端末関連の START コマンドによって開始されるトランザクション
- プログラム・リンク (DPL) 要求

これらのタイプの要求のルーティングの場合、DYRACMAL に以下のいずれかの数値が格納されます。

- ユーザー・アプリケーションが COMMAREA を使用する場合は、アプリケーションの COMMAREA の長さ (バイト単位)
- ユーザー・アプリケーションがチャンネルを使用し、チャンネル内に DFHROUTE というコンテナを作成している場合は、DFHROUTE コンテナ内のデータの長さ (バイト単位)
- ユーザー・アプリケーションに COMMAREA がなく、DFHROUTE コンテナもない場合は、ゼロ

その他のすべてのタイプの要求のルーティングでは、DYRACMAL にゼロが格納されます。

DYRACMAL (既存のフィールド: 分散ルーティング・プログラム (変更なし))

このフィールドは、分散ルーティング・プログラムでは使用されません。このフィールドは、呼び出し時にゼロに設定されます。

DYRCHANL (新規フィールド)

これは、プログラム・リンクまたは START コマンドに関連付けられたチャンネルの名前 (チャンネルが存在する場合) です。このフィールドは、DPL 要求、非端末関連 START 要求、および端末関連 START 要求によって開始されたトランザクションのルーティングにのみ適用されます。その他のタイプの要求の場合、あるいはコマンドにチャンネルが関連付けられていない場合、このフィールドはブランクです。

ルーティング・プログラムにはチャンネルのアドレスではなく、名前が渡されるので、このフィールドの内容を使用してコンテナの内容を検査したり、変更したりできないことに注意してください。ルーティング・プログラムでアプリケーションのコンテナの内容を検査または変更する方法については、226 ページの『チャンネルを使用した動的ルーティング』および DYRACMAA フィールドの説明を参照してください。

DYRLEVEL (既存のフィールド)

これは、ターゲットの AOR がルーティングされた要求を正常に処理するために必要な CICS のレベルです。以下の新しい値が追加されました。

X'03' CICS TS for z/OS バージョン 3.1。現在は、以下に対してのみ設定できます。

- チャンネルが関連付けられている DPL 要求
- チャンネルが関連付けられている START 要求
- Enterprise Bean および CORBA ステートレス・オブジェクトのためのメソッド要求

DYRTYPE (既存のフィールド)

ルーティング・プログラムを呼び出した要求のタイプが入ります。以下の 3 つの新しい値が追加されました。

9 チャンネルを使用したプログラム・リンク要求。

要求のルーティングのために、動的 ルーティング・プログラムが呼び出されます。

A START に関連付けられたチャンネルのある、端末関連 EXEC CICS START コマンドによって開始されるトランザクション。

トランザクションのルーティングのために、動的 ルーティング・プログラムが呼び出されます。

B チャンネルを使用した非端末関連の START 要求。

要求のルーティングのために、分散 ルーティング・プログラムが呼び出されます。

以下の既存の値の意味が変更されました。(既存の説明から変更された部分を太字で示します)

2 START に関連付けられたデータおよび**チャンネルがない**、端末関連 EXEC CICS START コマンドによって開始されるトランザクション。

3 START に関連付けられたデータはあるが、**チャンネルはない**、端末関連 EXEC CICS START コマンドによって開始されるトランザクション。

4 **チャンネルを指定していない**プログラム・リンク要求。

6 データが指定されているかまたは指定されていないが、**チャンネルは指定されていない**、非端末関連 START 要求。

DYRVER (既存のフィールド)

動的ルーティング・プログラム・インターフェースのバージョン番号。**CICS Transaction Server for z/OS バージョン 3 リリース 1** の場合、番号は「10」です。

モニターの変更点

パフォーマンス・クラス・モニター・レコードのために、いくつかの新規フィールドが追加されました。1 つの新規グループ DFHCHNL が追加され、さらに、DFHPROG および DFHTASK グループに対する追加が行われました。

すべての新規フィールドは、モニター管理テーブル (MCT) に DFHMCT TYPE=RECORD 項目をコーディングすることにより、モニター・レコードから除外することができます。

グループ DFHCHNL のパフォーマンス・データ

グループ DFHCHNL には、以下のパフォーマンス・データが含まれます。

321 (TYPE-A, 'PGTOTCCT', 4 BYTES)

ユーザー・タスクによって発行された、チャンネル・コンテナに対する CICS 要求の数。

322 (TYPE-A, 'PGBRWCCT', 4 BYTES)

ユーザー・タスクによって発行された、チャンネル・コンテナに対する CICS ブラウズ要求の数。

323 (TYPE-A, 'PGGETCCT', 4 BYTES)

ユーザー・タスクによって発行された、チャンネル・コンテナに対する GET CONTAINER 要求の数。

324 (TYPE-A, 'PGPUTCCT', 4 BYTES)

ユーザー・タスクによって発行された、チャンネル・コンテナに対する PUT CONTAINER 要求の数。

325 (TYPE-A, 'PGMOVCCT', 4 BYTES)

ユーザー・タスクによって発行された、チャンネル・コンテナに対する MOVE CONTAINER 要求の数。

326 (TYPE-A, 'PGGETCDL', 4 BYTES)

ユーザー・タスクによって発行されたすべての GET CONTAINER CHANNEL コマンドのコンテナ内のデータの全長 (バイト数)。

327 (TYPE-A, 'PGPUTCDL', 4 BYTES)

ユーザー・タスクによって発行されたすべての PUT CONTAINER CHANNEL コマンドのコンテナ内のデータの全長 (バイト数)。

グループ DFHPROG のパフォーマンス・データ

以下の新規フィールドが、グループ DFHPROG に追加されました。

286 (TYPE-A, 'PCDLCS DL', 4 BYTES)

CHANNEL オプションを指定して、ユーザー・タスクによって発行されたすべての分散プログラム・リンク (DPL) 要求のコンテナ内のデータの全長 (バイト数)。この合計には、データに対するすべてのヘッダーの長さが含まれます。

287 (TYPE-A, 'PCDL CRDL', 4 BYTES)

ユーザー・タスクによって発行されたすべての DPL RETURN CHANNEL コマンドのコンテナ内のデータの全長 (バイト数)。この合計には、データに対するすべてのヘッダーの長さが含まれます。

306 (TYPE-A, 'PCLNK CCT', 4 BYTES)

CHANNEL オプションを指定して、ユーザー・タスクによって発行されたローカル・プログラム LINK 要求の数。

307 (TYPE-A, 'PCXCLCCT', 4 BYTES)

CHANNEL オプションを指定して、ユーザー・タスクによって発行されたプログラム XCTL 要求の数。

308 (TYPE-A, 'PCDPLCCT', 4 BYTES)

CHANNEL オプションを指定して、ユーザー・タスクによって発行されたプログラムの分散プログラム・リンク (DPL) 要求の数。

309 (TYPE-A, 'PCRTNCCT', 4 BYTES)

CHANNEL オプションを指定して、ユーザー・タスクによって発行されたリモート疑似会話型 RETURN 要求の数。

310 (TYPE-A, 'PCRTNCDL', 4 BYTES)

ユーザー・タスクによって発行されたすべてのリモート疑似会話型 RETURN CHANNEL コマンドのコンテナ内のデータの全長 (バイト数)。この合計には、データに対するすべてのヘッダーの長さが含まれます。

グループ DFHTASK のパフォーマンス・データ

以下の新規フィールドが、グループ DFHTASK に追加されました。

065 (TYPE-A, 'ICSTACCT', 4 BYTES)

CHANNEL オプションを指定して、ユーザー・タスクによって発行されたローカル・インターバル制御機能 START 要求の総数。

345 (TYPE-A, 'ICSTACDL', 4 BYTES)

ユーザー・タスクによって発行された、ローカルで実行されるすべての START CHANNEL 要求のコンテナ内のデータの全長 (バイト数)。この合計には、データに対するすべてのヘッダーの長さが含まれます。

346 (TYPE-A, 'ICSTRCCT', 4 BYTES)

ユーザー・タスクによって発行された、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 要求の総数。

347 (TYPE-A, 'ICSTRCDL', 4 BYTES)

ユーザー・タスクによって発行された、リモートで実行されるすべての START CHANNEL コマンドのコンテナ内のデータの全長 (バイト数)。この合計には、データに対するすべてのヘッダーの長さが含まれます。

統計の変更点

接続を經由したチャンネル・データ・フローに関する新規の統計があります。これらの統計は DFHA14DS DSECT によりマップされます。これらは DFHSTUP の「ISC/IRC システム項目: リソース統計」レポート、および DFH0STAT の「接続およびモードネーム・レポート」に示されます。

表 5. 「接続およびモードネーム・レポート」の新規フィールド

フィールド・ヘッダー	説明
端末共用チャンネル要求数	チャンネルを使用した端末共用要求の数。 ソース・フィールド: A14ESTTC_CHANNEL

表 5. 「接続およびモードネーム・レポート」の新規フィールド (続き)

フィールド・ヘッダー	説明
端末共用チャンネル要求: 送信済みバイト数	端末共用チャンネル要求で送信したバイト数。これは、接続で送信したデータの総量であり、制御情報を含みます。 ソース・フィールド: A14ESTTC_CHANNEL_SENT
端末共用チャンネル要求: 受信済みバイト数	端末共用チャンネル要求で受信したバイト数。これは、接続で受信したデータの総量であり、制御情報を含みます。 ソース・フィールド: A14ESTTC_CHANNEL_RCVD
チャンネルを使用したプログラム制御 LINK 要求数	接続全体で機能シッパされた、チャンネルを使用したプログラム制御 LINK 要求の数。 ソース・フィールド: A14ESTPC_CHANNEL
LINK チャンネル要求: 送信済みバイト数	LINK チャンネル要求で送信したバイト数。これは、接続で送信したデータの総量であり、制御情報を含みます。 ソース・フィールド: A14ESTPC_CHANNEL_SENT
LINK チャンネル要求: 受信済みバイト数	LINK チャンネル要求で受信したバイト数。これは、接続で受信したデータの総量であり、制御情報を含みます。 ソース・フィールド: A14ESTPC_CHANNEL_RCVD
チャンネルを使用したインターバル制御機能 START 要求数	接続全体で機能シッパされた、チャンネルを使用したインターバル制御機能 START 要求の数。 ソース・フィールド: A14ESTIC_CHANNEL
START チャンネル要求: 送信済みバイト数	START チャンネル要求で送信したバイト数。これは、接続で送信したデータの総量であり、制御情報を含みます。 ソース・フィールド: A14ESTIC_CHANNEL_SENT
START チャンネル要求: 受信済みバイト数	START チャンネル要求で受信したバイト数。これは、接続で受信したデータの総量であり、制御情報を含みます。 ソース・フィールド: A14ESTIC_CHANNEL_RCVD

サンプル・プログラムの変更点

EXEC CICS LINK および RETURN コマンドでチャンネルを渡す方法を示すために、IIOP 銀行口座サンプル・アプリケーション (トランザクション BNKQ) の 2 つの C 言語プログラムが変更されました。COMMAREA モデルではなくチャンネル/コンテナ・モデルを使用するために、最上位のプログラム DFH\$IIBQ およびこのプログラムがリンクするプログラム DFH\$IICC が変更されました。コード内のコメントは、変更点を示しています。

問題判別の変更点

問題判別の変更点

メッセージ

拡張プログラム間データ転送の結果として、新規メッセージが導入されました。すべての新規メッセージと変更されたメッセージについては、「*CICS Messages and Codes*」を参照してください。

異常終了コード

いくつかの新しい異常終了コードが導入され、既存のコードの一部が除去されました。

新しい異常終了コード:

AEYF (CICS TS for z/OS バージョン 3.1 のみ):

説明

CICS による記憶保護違反。

トランザクション自体がアクセスできないストレージ域に CICS がアクセスするように、トランザクションが要求しました。これは、PUT CONTAINER または GET CONTAINER コマンドで、CICS に対して無効なストレージ域が渡されたときに発生します。以下の場合に、エラーが発生する可能性があります。

- FROM または INTO アドレスが誤って指定された場合
- 指定された FLENGTH の値が大きく、トランザクションがアクセスできないストレージが領域に含まれることになる場合。

一般に、フルワード領域を指定すべき FLENGTH パラメーターにハーフワード領域のアドレスを指定したことが、このエラーの原因になります。以前 COMMAREA (長さがハーフワード) を使用したプログラムが、コンテナ (長さがフルワード) を使用するように変更された場合に、このエラーが発生する可能性があります。

システム処置

トランザクションは、CICS トランザクション・ダンプを生成して異常終了します。

ユーザー応答

トレースを調べて、DFHEISR へのエントリーのトレース・エントリーを検出してから、エラー状態のパラメーターを識別します。異常終了が処理された場合、EXEC CICS ASSIGN ASRASTG、ASRAKEY、ASRASPC、および ASRAREGS によって、異常終了に関する追加情報が得られます。異常終了の際には、障害の発生したストレージ域に 2 つのポイントを登録します。

ほとんどの場合、以下を実行する必要があります。

- EXEC CICS PUT CONTAINER または EXEC CICS GET CONTAINER コマンドを発行した、エラー状態のプログラムを訂正します。プログラムが有効なストレージ域のアドレスを提供し、ストレージ域にトランザクションがアクセス不能な部分が含まれないように FLENGTH を提供していることを確認します。FLENGTH がフルワードの長さを参照することを確認します。

以下の 1 つ以上の変更を検討することも必要になります。

- ストレージ保護がアクティブな場合、EXEC CICS コマンドを発行したプログラムの CEDA 定義で、EXECKEY オプションを USER から CICS に変更します。
- ストレージ保護がアクティブな場合、トランザクション定義の TASKDATAKEY 属性を CICS から USER に変更します。
- トランザクション分離がアクティブな場合、トランザクション定義の ISOLATE 属性を YES から NO に変更します。

モジュール

DFHSRP

AITI (CICS TS for z/OS バージョン 3.1 のみ):

説明

接続された CICS システムとの間でデータの受信または送信を試行中に、START CHANNEL または LINK CHANNEL 要求を処理するミラー・トランザクションが失敗しました。チャンネルには大量のデータが含まれている可能性があるため、チャンネル・データを送信するには、端末管理に対する多数の呼び出しが必要になる場合があります。DFHMIRS は、チャンネル・データのすべてのシステム間送信を実行するために、プログラム DFHAPCR を呼び出します。端末管理は、これらの呼び出しのうちの 1 つでエラーを検出しました。エラーは、読み取りタイムアウト、またはフロー内のより重大なエラーである可能性があり、これが原因で CICS によるデータの正常な処理が妨げられました。

システム処置

トランザクションは終了します。ミラー・タスクは、CICS トランザクション・ダンプを生成して異常終了します。

ユーザー応答

エラーがタイムアウトだった場合は、リモート領域が応答しなかった理由を判別してください。トレースを調べて、GETMAIN が失敗した理由を判別してください。CICS 領域がストレージ不足だった場合は、これを訂正するために必要な手順を実行してください。領域がストレージ不足でなかった場合は、この問題を解決するために IBM の助けが必要な場合があります。

モジュール

DFHADDRM

AXGA (CICS TS for z/OS バージョン 3.1 のみ):

説明

プログラム DFHAPCR が、予期しない応答を戻しました。DFHAPCR は、以下の機能を実行します。

- チャンネルを構成するすべてのコンテナの中身を取り出して、リモート・システムにこれを送信する
- リモート・システムから受信したインバウンド・データからチャンネルおよびコンテナを再作成する

DFHAPCR が、チャンネル・データの取り出し中または再作成中に、インバウンド・データでエラーを検出したか、予期しない応答を受信しました。

システム処置

トランザクションは、CICS トランザクション・ダンプを生成して異常終了します。

ユーザー応答

関連する CICS メッセージおよび異常終了を調べて、コンテナを管理しているプログラム・マネージャーで直前にエラーが発生していないか判別してください。プログラム・マネージャーまたは DFHAPCR の例外トレース・エントリーを調べて、エラーの原因を判別してください。

モジュール

DFHXTP

AXTS (CICS TS for z/OS バージョン 3.1 のみ):

説明

疑似会話のトランザクション間でチャンネルおよびコンテナ・データを渡す試みが行われましたが、疑似会話の次のトランザクションは、チャンネルおよびコンテナをサポートしない CICS 領域にあります。

システム処置

トランザクションは、CICS トランザクション・ダンプを生成して異常終了します。

ユーザー応答

ご使用のアプリケーションにおいて疑似会話のトランザクション間でチャンネルおよびコンテナ・データを使用している場合は、疑似会話のすべてのトランザクションが CICS TS for z/OS バージョン 3.1 以降の領域にあることを確認してください。

モジュール

DFHXTP

AXTU (CICS TS for z/OS バージョン 3.1 のみ):

説明

プログラム DFHAPCR が、予期しない応答を戻しました。DFHAPCR は、以下の機能を実行します。

- チャンネルを構成するすべてのコンテナの中身を取り出して、リモート・システムにこれを送信する
- リモート・システムから受信したインバウンド・データからチャンネルおよびコンテナを再作成する

DFHAPCR が、チャンネル・データの取り出し中または再作成中に、インバウンド・データでエラーを検出したか、予期しない応答を受信しました。

システム処置

トランザクションは、CICS トランザクション・ダンプを生成して異常終了します。

ユーザー応答

関連する CICS メッセージおよび異常終了を調べて、コンテナを管理しているプログラム・マネージャーで直前にエラーが発生していないか判別してください。プログラム・マネージャーまたは DFHAPCR の例外トレース・エントリを調べて、エラーの原因を判別してください。

モジュール

DFHXTP

その他の異常終了コード:

互換性の目的のため、新たに以下の異常終了コードが CICS TS for z/OS バージョン 2.2 および CICS TS for z/OS バージョン 2.3 に追加されました。これらは CICS TS for z/OS バージョン 3.1 には適用されません。将来、CICS TS for z/OS バージョン 2.2 または CICS TS for z/OS バージョン 2.3 領域を CICS TS for z/OS バージョン 3.1 にマイグレーションする場合は、これらの異常終了コードが CICS TS for z/OS バージョン 3.1 では発生しないことに注意してください。

- AXF9
- AXTT

トレース

新機能に関連する CICS トレース・ポイントは AP 0785 および AP 4E20 から AP 4E22 です。

CICS トレース情報の出力を制御するには、通常の方法で CICS トレース制御を使用します。

COMMAREA からチャネルへのマイグレーション

既存の機能のマイグレーション

- 従来の通信域 (COMMAREA) を使用してデータ交換する CICS アプリケーション・プログラムは、これまでと同様に機能します。
- CICSplex SM ではなく、ユーザー作成の動的または分散ルーティング・プログラムをワークロード管理のために利用する場合は、プログラムを変更して、DFHDYPDS 通信域の DYRLEVEL、DYRTYPE、および DYRVER フィールドで渡される可能性のある新しい値を処理できるようにする必要があります。248 ページの『動的および分散ルーティング・プログラム』を参照してください。

新規機能へのマイグレーション

このセクションでは、通信域 (COMMAREA) の代わりにチャネルとコンテナを使用するために、いくつかのタイプの既存のアプリケーションをマイグレーションする方法について説明します。

単一のコンテナを持つチャンネルによって、COMMAREA を置き換えることが可能です。COMMAREA からチャンネルおよびコンテナに移動するには、これが最も簡単な方法のように思われるかもしれませんが、これは良い方法ではありません。この新規の機能を活用するようにアプリケーション・プログラムを変更するために時間がかかるため、チャンネルおよびコンテナ用に「ベスト・プラクティス」をインプリメントする必要があります。218 ページの『チャンネルの設計: ベスト・プラクティス』を参照してください。チャンネルには COMMAREA よりも優れた点がいくつかあり (199 ページの『チャンネルの利点』を参照)、独自のチャンネルを設計してこれらの改善点を最大限に利用することには、それだけの価値があります。

COMMAREA を渡す LINK コマンドのマイグレーション

LINK コマンドで COMMAREA を使用して構造を交換している 2 つのプログラムをマイグレーションするには、表 6 に示す命令を変更します。

表 6. COMMAREA を渡す LINK コマンドのマイグレーション

プログラム	変更前	変更後
PROG1	EXEC CICS LINK PROGRAM(PROG2) COMMAREA(structure)	EXEC CICS PUT CONTAINER(structure-name) CHANNEL(channel-name) FROM(structure) EXEC CICS LINK PROGRAM(PROG2) CHANNEL(channel-name) ... EXEC CICS GET CONTAINER(structure-name) CHANNEL(channel-name) INTO(structure)
PROG2	EXEC CICS ADDRESS COMMAREA(structure-ptr) ... RETURN	EXEC CICS GET CONTAINER(structure-name) INTO(structure) ... EXEC CICS PUT CONTAINER(structure-name) FROM(structure) RETURN

注: COMMAREA の例では、COMMAREA にデータを書き込んだ PROG2 で必要な処理は、RETURN コマンドを発行して、データを PROG1 に戻すことだけです。チャンネルの例では、データを戻すためには、PROG2 が RETURN の前に PUT CONTAINER コマンドを発行する必要があります。

COMMAREA を渡す XCTL コマンドのマイグレーション

XCTL コマンドで COMMAREA を使用して構造を渡している 2 つのプログラムをマイグレーションするには、表 7 に示す命令を変更します。

表 7. COMMAREA を渡す XCTL コマンドのマイグレーション

プログラム	変更前	変更後
PROG1	EXEC CICS XCTL PROGRAM(PROG2) COMMAREA(structure)	EXEC CICS PUT CONTAINER(structure-name) CHANNEL(channel-name) FROM(structure) EXEC CICS XCTL PROGRAM(PROG2) CHANNEL(channel-name) ...
PROG2	EXEC CICS ADDRESS COMMAREA(structure-ptr) ...	EXEC CICS GET CONTAINER(structure-name) INTO(structure) ...

RETURN コマンドでの疑似会話型 COMMAREA のマイグレーション

疑似会話の一部として COMMAREA を使用して構造を交換している 2 つのプログラムをマイグレーションするには、表 8 に示す命令を変更します。

表 8. RETURN コマンドでの疑似会話型 COMMAREA のマイグレーション

プログラム	変更前	変更後
PROG1	EXEC CICS RETURN TRANSID(PROG2) COMMAREA(structure)	EXEC CICS PUT CONTAINER(structure-name) CHANNEL(channel-name) FROM(structure) EXEC CICS RETURN TRANSID(TRAN2) CHANNEL(channel-name)
PROG2	EXEC CICS ADDRESS COMMAREA(structure-ptr)	EXEC CICS GET CONTAINER(structure-name) INTO(structure)

START データのマイグレーション

START データを使用して構造を交換している 2 つのプログラムをマイグレーションするには、表 9 に示す命令を変更します。

表 9. START データのマイグレーション

プログラム	変更前	変更後
PROG1	EXEC CICS START TRANSID(TRAN2) FROM(structure)	EXEC CICS PUT CONTAINER(structure-name) CHANNEL(channel-name) FROM(structure) EXEC CICS START TRANSID(TRAN2) CHANNEL(channel-name)
PROG2	EXEC CICS RETRIEVE INTO(structure)	EXEC CICS GET CONTAINER(structure-name) INTO(structure)

新しいバージョンの PROG2 は、疑似会話型の例のものと同じです。

動的にルーティングされるアプリケーションのマイグレーション

COMMAREA またはチャンネルのいずれかの受け渡し可能な EXEC CICS LINK コマンドおよび EXEC CICS START コマンドは、動的にルーティングすることができます。

LINK コマンドまたは START コマンドが、チャンネルではなく COMMAREA を受け渡す場合は、ルーティング・プログラムは、要求のタイプに応じて、COMMAREA の内容を検査したり、変更したりすることができます。端末関連 START 要求 (動的 ルーティング・プログラムによって処理される) によって開始された LINK 要求およびトランザクションの場合、ルーティング・プログラムに対し、通信域の DYRACMAA フィールド内でアプリケーションの COMMAREA のアドレスが渡され、ルーティング・プログラムは、その内容を検査および変更することができます。ただし、非端末関連 START 要求 (分散 ルーティング・プログラムによって処理される) の場合を除きます。

注: ルーティング・プログラムの通信域は、DFHDYPDS DSECT によってマップされます。

動的にルーティングされた EXEC CICS LINK または START コマンドをマイグレーションして、COMMAREA ではなくチャンネルを使用する場合、ルーティング・プ

ログラムは、DFHDYPDS の DYRCHANL フィールドでチャンネルの名前を渡されます。ルーティング・プログラムにはチャンネルのアドレスではなく、名前が指定されるので、DYRCHANL フィールドを使用してチャンネルのコンテナの内容を検査または変更することはできないことに注意してください。

ルーティング・プログラムにチャンネルを使用して同種の機能を提供するために、チャンネルを使用するアプリケーションがチャンネル内に、DFHROUTE という特殊なコンテナを作成することができます。アプリケーションが動的にルーティングされる LINK または端末関連の START 要求 (端末関連以外の START 要求ではなく) を出すと、動的ルーティング・プログラムの DFHDYPDS の DYRACMAA フィールドに DFHROUTE コンテナのアドレスが渡されるため、動的ルーティング・プログラムはその内容を検査および変更することができます。

COMMAREA ではなく、チャンネルを渡すようにプログラムをマイグレーションする場合、DFHROUTE へのマップに既存の COMMAREA 構造を使用することができます。

動的ルーティングおよび分散ルーティングの概説については、「*CICS Intercommunication Guide*」を参照してください。動的ルーティング・プログラムまたは分散ルーティング・プログラムの作成方法については、「*CICS Customization Guide*」を参照してください。

共存

他の CICS 製品との共存

CICS TS 3.1 プログラムは、リモートの CICS 領域上のプログラムを呼び出して、これにチャンネルを渡すことができます。この処理が正常に実行されるようにするには、そのリモート領域も CICS TS 3.1 レベルでなければなりません。

CICS TS 3.1 よりも前の領域ではチャンネルはサポートされていませんが、APAR を適用することによって、これらの領域にチャンネルを容認させることができます。

『容認』とは、バックレベル CICS 領域がチャンネルを渡された場合に、意味のある異常終了コードを戻す、という意味です。

適切な APAR が適用されたバックレベル領域に対して CICS TS 3.1 アプリケーションがチャンネルを送信しようとした場合、3.1 トランザクションは意味のある異常終了コードで異常終了します。適切な APAR が適用されていないバックレベル領域に対して CICS TS 3.1 アプリケーションがチャンネルを送信しようとした場合、その結果は予測不能です。

以下のリストは、チャンネルを容認するバックレベル CICS 製品を、それぞれに適用する必要がある APAR と共に示しています。

CICS Transaction Server for z/OS バージョン 2 リリース 3

APAR PQ92437

CICS Transaction Server for z/OS バージョン 2 リリース 2

APAR PQ92437

CICS Transaction Server for OS/390 バージョン 1 リリース 3

APAR PQ93048

CICSplex SM サポート

通信域の DYRLEVEL、DYRTYPE、および DYRVER フィールドで渡される可能性のある新しい値を処理するために、CICSplex SM ルーティング・プログラム EYU9XLOP が変更されました。248 ページの『動的および分散ルーティング・プログラム』を参照してください。以下のいずれかのタイプの要求をルーティングするために呼び出されたときに、要求と関連付けられたチャンネルがある場合、EYU9XLOP は要求を CICS TS 3.1 領域にルーティングします (この領域が利用可能な場合)。

- 端末関連の START コマンドが開始したトランザクション
- 非端末関連の START 要求
- プログラム・リンク要求

CICSplex SM アプリケーション・プログラミング・インターフェースの変更点

以下のテーブルに対して変更が加えられました。

- 『TASK リソース・テーブル』
- 262 ページの『CONNECT リソース・テーブル』

TASK リソース・テーブル

新規モニターおよび統計属性を組み込むために TASK リソース・テーブルが以下のように拡張されました。

TMRPGCTC

チャンネル・コンテナ・コマンドの総数

TMRPGBCC

BROWSE チャンネル・コンテナ・コマンドの数

TMRPGGCC

GET チャンネル・コンテナ・コマンドの数

TMRPGPCC

PUT チャンネル・コンテナ・コマンドの数

TMRPGMCC

MOVE チャンネル・コンテナ・コマンドの数

TMRPGGCL

すべての GET CONTAINER CHANNEL コマンドのコンテナ内のデータの全長 (バイト数)

TMRPGPCL

すべての PUT CONTAINER CHANNEL コマンドのコンテナ内のデータの全長 (バイト数)

TMRPGCCC

チャンネル・コンテナ用に作成されたコンテナの数

TMRPCDLL

CHANNEL オプションを指定したすべての DPL 要求のコンテナ内のデータの全長 (バイト数)

TMRPCDRL

すべての DPL RETURN CHANNEL コマンドのコンテナ内のデータの全長 (バイト数)

TMRPCLCC

CHANNEL オプションを指定したプログラム LINK 要求の数

TMRPCXCC

CHANNEL オプションを指定したプログラム XCTL 要求の数

TMRPCDCC

CHANNEL オプションを指定したプログラムの分散プログラム・リンク (DPL) 要求の数

TMRPCRCC

CHANNEL オプションを指定した疑似会話型 RETURN 要求の数

TMRPCRCL

すべての疑似会話型 RETURN CHANNEL コマンドのコンテナ内のデータの全長 (バイト数)

TMRICSCC

CHANNEL オプションを指定して発行されたローカル・インターバル制御機能 START 要求の総数

TMRICSCD

CHANNEL オプションを指定したすべてのローカル・インターバル制御機能 START 要求のコンテナ内のデータの全長 (バイト数)

TMRICSRC

CHANNEL オプションを指定して発行されたリモート・インターバル制御機能 START 要求の総数

TMRICSRD

CHANNEL オプションを指定したすべてのリモート・インターバル制御機能 START 要求のコンテナ内のデータの全長 (バイト数)

CONNECT リソース・テーブル

新規モニターおよび統計属性を組み込むために CONNECT リソース・テーブルが以下のように拡張されました。

ESTPCCHNL

チャンネルを使用した機能シップのためのプログラム制御 LINK 要求の数。

ESTPCCHNSENT

LINK チャンネル要求で送信したバイト数。これは、接続で送信したデータの総量であり、制御情報を含みます。

ESTPCCHNRCVD

LINK チャンネル要求で受信したバイト数。これは、接続で受信したデータの総量であり、制御情報を含みます。

ESTICCHNL

チャンネルを使用した機能シップのためのインターバル制御機能 START 要求の数。

ESTICCHNSENT

START チャンネル要求で送信したバイト数。これは、接続で送信したデータの総量であり、制御情報を含みます。

ESTICCHNRCVD

START チャンネル要求で受信したバイト数。これは、接続で受信したデータの総量であり、制御情報を含みます。

ESTTCCHNL

端末共用チャンネル要求の数。

ESTTCCHNSENT

端末共用チャンネル要求で送信したバイト数。これは、接続で送信したデータの総量であり、制御情報を含みます。

ESTTCCHNRCVD

端末共用チャンネル要求で受信したバイト数。これは、接続で受信したデータの総量であり、制御情報を含みます。

CICSplex SM Web ユーザー・インターフェースの変更点

新規 WUI ビュー

以下の WUI ビューが導入されました。

- 『チャンネル使用ビュー』
- 『機能シップ・ビュー』

チャンネル使用ビュー

「アクティブ・タスク」ビュー・セット内に、「チャンネル使用」という新規の詳細ビューがあります。これは、TASK リソース・テーブルに関連付けられています。

「チャンネル使用」ビューを開くには、以下を実行します。

1. メインメニューから「アクティブ・タスク」をクリックします。
2. 「タスク ID」を選択して「アクティブ・タスク」詳細ビューを開きます。
3. スクロールダウンし、「チャンネル使用」をクリックします。

新規の「アクティブ・タスク」ビューが表示されます。この「アクティブ・タスク」のビュー名は、EYUSTARTTASK.DETAIL10 です。「チャンネル使用」ビュー内で表示されるフィールドの属性の詳細については、261 ページの『TASK リソース・テーブル』を参照してください。

機能シップ・ビュー

「ISC/MRO 接続」ビュー・セット内に、「機能シップ (Function ships)」という新規の詳細ビューがあります。これは、CONNECT リソース・テーブルに関連付けられています。

「機能シップ (Function ships)」ビューをオープンするには、以下の手順を実行します。

1. メインメニューから「**CICS 操作ビュー**」をクリックします。
2. 「**接続操作ビュー**」をクリックします。
3. 「**ISC と MRO の接続**」を選択します。
4. 接続 ID をクリックして、「**ISC/MRO 接続**」詳細ビューを開きます。
5. スクロールダウンして「**機能シップ (Function ships)**」をクリックします。これにより、ISC/MRO 接続の新規ビューがオープンします。

この ISC/MRO 接続のビュー名は EYUSTARTCONNECT.DETAILED3 です。

変更された WUI ビュー

以下のビューに対して変更が加えられました。

ISC/MRO 接続ビュー

CONNECT リソース・テーブルに関連付けられた「**ISC/MRO 接続**」という既存の詳細ビューに、いくつかの新規フィールドが追加されました。

新規フィールドを表示するには、以下を実行します。

1. メインメニューから「**CICS 操作ビュー**」をクリックします。
2. 「**接続操作ビュー**」をクリックします。
3. 「**ISC と MRO の接続**」を選択します。
4. 接続 ID をクリックして、「**ISC/MRO 接続**」詳細ビューを開きます。

追加された新規フィールドを表示する詳細ビューのビュー名は、EYUSTARTCONNECT.DETAILED です。「**ISC/MRO 接続**」詳細ビューに表示される新規フィールドの属性の詳細については、262 ページの『CONNECT リソース・テーブル』を参照してください。

第 8 章 OPENAPI サポート

CICS では、OPENAPI アプリケーション・プログラムのサポートを提供することによって、オープン・トランザクション環境 (OTE) 機能の使用が拡張されました。これまでは、OPENAPI 機能はタスク関連ユーザー出口 (TRUE) にのみ使用可能でした。

OPENAPI サポートによって、アプリケーションはそれ自体をスレッド・セーフとして定義できる (QR TCB でもオープン TCB でも、CICS が適切と判断する任意の TCB 上で実行できることを意味する) だけでなく、それを上回ることが可能になります。つまり、アプリケーションは QR TCB 上ではなく OPEN TCB 上で実行しなければなりません。

OPENAPI プログラムを使用すれば、アプリケーションのワークロードを QR TCB から複数のオープン TCB に移動できます。他の (非 CICS) API を使用したワークロードの実行方法として OPENAPI プログラムの使用を選択した場合は、**CICS 内**での他の (非 CICS) API の使用は、すべてユーザーの判断および責任において行ってください。**CICS 内**の他の (非 CICS) API に関するテストは実施されておらず、そのような API の使用は IBM サービスではサポートされません。

どちらの場合でも、OPENAPI プログラムは CICS システム全体に対する義務がまだあることに注意しなければなりません。「*CICS Application Programming Guide*」を参照してください。

PROGRAM リソース定義の新規キーワード (API) は、2 つの値 CICSAPI または OPENAPI のいずれかをとりますが、デフォルトは CICSAPI です。API (OPENAPI) を設定すると、CONCURRENCY (THREADSAFE) の設定が指示されます。これは、複数の TCB 上で平行に実行されたときにアプリケーション・ロジックを完全な状態で実行できるようにするために、アプリケーションをスレッド・セーフ標準に従ってコード化する必要があることを意味します。CICS は、そのようなプログラムから発行された任意の CICS API のスレッド・セーフティの側面を処理します。この新規プログラム・オプションは、ユーザー・アプリケーション・プログラム、PLT プログラム、ユーザーが置き換え可能なモジュール、およびタスク関連ユーザー出口に適用されます。グローバル・ユーザー出口ルーチンの場合は、このオプションは無視されます。

CICSAPI QUASIRENT プログラム、CICSAPI THREADSAFE プログラム、および OPENAPI THREADSAFE プログラムは、その実行場所に関して以下の相違点があります。

- CICSAPI QUASIRENT プログラムは CICS API のみを発行し、そのアプリケーション・ロジックはスレッド・セーフではありません。このプログラムは常に QR TCB 上で実行されます。
- CICSAPI THREADSAFE プログラムは、アプリケーション・ロジックがスレッド・セーフなので、QR TCB でもオープン TCB でも実行が可能です。このようなプログラムは、なんらかのイベントによってオープン TCB に移動されるまで、QR TCB 上で実行されます。CICSAPI THREADSAFE プログラムをオープン TCB に移動させるイベントの一例として、OPENAPI TRUE への呼び出し (DB2 呼び出しなど) があります。オープン TCB に移動した後、プログラムはそのま

まオープン TCB に残りますが、なんらかのイベント (例えば非スレッド・セーフの CICS API 呼び出し) が発生すると、プログラムは強制的に QR TCB に戻されます。このようになった場合、プログラムは QR TCB に残りますが、なんらかのイベント (別の DB2 呼び出しなど) が発生すると、プログラムはもう一度強制的にオープン TCB に戻されます。

CICSAPI プログラムは、使用中の TCB のキーとは独立した方法でインプリメントされた CICS API のみを使用します。アプリケーションは、TCB のキーに関係なく、ユーザー・キーまたは CICS キーで正常に実行できます。したがって、QR TCB、L8 または L9 TCB で実行できます。

- OPENAPI THREADSAFE プログラムは、常にオープン TCB で実行され、プログラムの開始から実行されます。このプログラムは、アプリケーション・ロジックがスレッド・セーフなので、オープン TCB での実行が可能です。非スレッド・セーフの CICS コマンドを使用して、強制的に QR TCB に切り替えられた場合、CICS はもう一度オープン TCB に切り替えてから、アプリケーションに制御を戻します。

OPENAPI プログラムは、他の (非 CICS) API を使用することも可能なので、このような API には一般に、実行キーと一致する TCB のキーが必要です。このため、ユーザー・キー・プログラムは L9 TCB で実行され、CICS キー・プログラムは L8 TCB で実行されます。

OPENAPI プログラムを使用した場合、スレッド・セーフ CICSAPI プログラムよりも TCB 切り替えが多くなる可能性があります。その理由としては、OPENAPI プログラムには正しい TCB のキーが要求されること、非スレッド・セーフの CICS 呼び出しがあると TCB 切り替えが 2 回発生すること、および OPENAPI TRUE は常に L8 TCB 上で CICS キーにより実行されること、があります。このため、例えばユーザー・キーの OPENAPI プログラムは L9 TCB で実行されますが、DB2 呼び出しを行う場合、CICS は L8 TCB に切り替えて DB2 を呼び出してから、アプリケーション用の L9 に戻ります。

既存ユーザー・キーのスレッド・セーフ CICS-DB2 アプリケーションは、DB2 呼び出しと同じ TCB で実行可能というパフォーマンス向上が得られるので、CICSAPI THREADSAFE アプリケーションとして定義したままにしておくことを強くお勧めします。他の機能で OPENAPI を使用する必要がある場合は、別のプログラムを使用してください。

OPENAPI THREADSAFE (アプリケーション・ロジックがスレッド・セーフであることを前提とする) として定義する候補となるのは、以下のプログラムです。

- (TCB 切り替えが 2 回発生するのを回避するために) CICS スレッド・セーフ API のみを使用するか、限られた非スレッド・セーフ CICS コマンドのみを使用するプログラム
- CICS キーの CICS-DB2 アプリケーション
- CPU 集中プログラム
- 他の (非 CICS) API を個々の責任において使用するプログラム

OPENAPI サポート の利点

OPENAPI プログラムのサポートを提供する主な理由は、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 上に移動できるようにすることです。これにより、マシン・リソースの使用率が向上し、よりよいスループットを実現できる可能性があります。

OPENAPI プログラムを使用するもう一つの理由は、他の (非 CICS) API の使用が許可されるということです。

オペレーティング・システム待ちによってオープン TCB がブロックされた場合、影響を受けるのは 1 つのアプリケーションだけあり、QR TCB の下で実行した場合のように CICS 全体が影響を受けることはないので、他の API を使用することが可能です。そのような OPENAPI プログラムは、QR TCB での実行は一切許可されません。というのも、オペレーティング・システム待ちによって TCB がブロックされ、それによって CICS 全体が影響を受けるリスクがあるためです。

- **CICS 内での他の (非 CICS) API の使用は、すべてユーザーの判断および責任において行ってください。CICS 内の他の (非 CICS) API に関するテストは実施されておらず、そのような API の使用は IBM サービスではサポートされません。**

要件

OPENAPI サポートには、ハードウェア、ソフトウェア、またはリソース使用に関して、この CICS リリースの一般要件以外の特別な要件はありません。

CICS 外部インターフェースの変更点

システム初期設定パラメーターの変更点

OPENAPI サポートでは、FORCEQR システム初期設定パラメーターは OPENAPI プログラムには適用されないため、FORCEQR の説明は、CICSAPI プログラムへの関連に限定するように変更されました。変更された FORCEQR 説明の全文については、「*CICS システム定義ガイド*」を参照してください。

OPENAPI サポートでは、MAXOPENTCBS システム初期設定パラメーターの説明が、OPENAPI および L9 TCB の要件を含むように変更されました。変更された MAXOPENTCBS 説明の全文については、「*CICS システム定義ガイド*」を参照してください。

リソース定義の変更点

OPENAPI サポートでは、PROGRAM リソース定義に新規属性 API が導入されました。

API には、CICSAPI と OPENAPI という 2 つの可能値があります。API 属性の完全な説明、および変更された PROGRAM 定義の構文については、「*CICS Resource Definition Guide*」を参照してください。

アプリケーション・プログラミング・インターフェースの変更点

このトピックでは、以下について説明します。

- OPENAPI プログラムの義務
- EDF 使用時の制約事項

OPENAPI プログラムの義務

OPENAPI プログラムは、QR TCB によって課せられる制約を受けませんが、それでも、CICS システム全体と、それが使用している L8 または L9 TCB の将来のユーザーの両方に対する義務があります。L8 または L9 TCB は、割り振り先となる CICS タスク用に占有されますが、その CICS タスクが完了すると、TCB は、まだ「明確な」状態であればこれらの TCB のディスパッチャー管理プールに戻されます。(このコンテキストにおいて不明確な TCB とは、L8 または L9 モード TCB を使用するタスクが OPENAPI プログラムで未処理のまま異常終了したことを意味します。プログラムがスレッド・セーフ制約事項を破ったが CICS がそれを検出しない、という意味ではありません。) TCB は特定の OPENAPI プログラム用に占有されるのではなく、すべての OPENAPI プログラム、および L8 モード TCB の割り振り先となる CICS タスクによって起動される OPENAPI TRUE によって使用されます。また、OPENAPI プログラムを起動するアプリケーション・プログラムがスレッド・セーフ標準に従ってコード化されていて、スレッド・セーフとして CICS に定義されている場合、そのアプリケーション・プログラムはプログラムから戻るときに L8 モード TCB で実行を継続します。

スレッド・セーフの制約事項:

OPENAPI プログラムでは、以下についての問題が生じるような方法で実行オープン TCB 環境を処理しないでください。

- オープン TCB で実行できるアプリケーション・プログラム・ロジック
- 同じタスクによって呼び出される OPENAPI TRUE
- オープン TCB を使用する可能性のある将来のタスク
- CICS 管理コード

ユーザーの責任において OPENAPI プログラムで他の (非 CICS) API を使用することにした場合は、以下の点に注意する必要があります。

- CICS サービスを呼び出すとき、あるいは CICS に戻るときに、OPENAPI プログラムは、プログラムに入ったときの MVS プログラミング環境を確実に復元しなければなりません。これには、追加された ESTAE の取り消しのほかに、仮想記憶間モード、ASC モード、要求ブロック (RB) レベル、リンケージ・スタック・レベル、TCB ディスパッチング優先順位が含まれます。
- CICS タスク終了時に、OPENAPI プログラムは、必ずオープン TCB を他の CICS トランザクションが再利用するのに適した状態にしておく必要があります。特に、終了タスクに代わって限定的に獲得されたすべての非 CICS リソースは確実に解放されるようにしなければなりません。このようなリソースとしては、以下のものがあります。
 - 動的に割り振られたデータ・セット
 - オープン ACB または DCB

- STIMERM 要求
- MVS 管理ストレージ
- ENQ 要求
- 接続されたサブタスク
- ロードされたモジュール
- 所有されたデータ・スペース
- 追加されたアクセス・リスト・エントリー
- 名前/トークンのペア
- 固定ページ
- セキュリティー設定 (TCBSENV はゼロに設定する必要がある)
- OPENAPI プログラムでは、全体的な CICS 操作に影響する以下の MVS システム・サービスを使用しないでください。
 - CHKPT
 - ESPIE
 - QEDIT
 - SPIE
 - STIMER
 - TTIMER
 - XCTL / XCTLX
 - 任意の TSO/E サービス
- OPENAPI プログラムでは、L8 または L9 モード TCB の下で、MVS 言語環境プログラム・サービスを使用した言語環境プログラムを起動しないでください。というのも、L8 および L9 モード TCB は、CICS サービスを使用する言語環境プログラム用に初期設定されているためです。

EDF 使用時の制約事項

OPEN TCB および EDF

ユーザーのプログラムが通常は OPEN TCB (L8、L9、X8、または X9) を使用して実行される場合でも、CEDF はそのプログラムを強制的に QR TCB で実行します。これは、CEDF 自体がスレッド・セーフではないためです。

システム・プログラミング・インターフェースの変更点

OPENAPI サポートでは、SPI に関して以下の変更点があります。

INQUIRE DISPATCHER

INQUIRE DISPATCHER コマンドの ACTOPENTCBS および MAXOPENTCBS の説明は、L9 モード TCB を含み、OPENAPI プログラムを参照するように変更されました。変更された説明の全文については、「*CICS System Programming Reference*」を参照してください。

INQUIRE EXITPROGRAM

INQUIRE EXITPROGRAM コマンドの APIST オプションに新しい値

CICSAPI が導入されました。CICSAPI は、以前の値 BASEAPI と同義です。変更された説明の全文については、「*CICS System Programming Reference*」を参照してください。

INQUIRE PROGRAM

INQUIRE PROGRAM コマンドに、新規オプション APIST が追加されました。APIST を使用すると、CICSAPI または OPENAPI を指定できます。APIST の説明の全文については、「*CICS System Programming Reference*」を参照してください。

INQUIRE SYSTEM

INQUIRE SYSTEM コマンドの FORCEQR は OPENAPI プログラムには適用されないため、FORCEQR の説明は、CICSAPI プログラムへの関連に限定するように変更されました。変更された説明の全文については、「*CICS System Programming Reference*」を参照してください。

SET DISPATCHER

SET DISPATCHER コマンドの MAXOPENTCBS の説明は、L9 モード TCB を含むように変更されました。変更された説明の全文については、「*CICS System Programming Reference*」を参照してください。

SET SYSTEM

SET SYSTEM コマンドのオプション FORCEQR の cvda 値 FORCE は、OPENAPI プログラムには適用されないため、FORCE の説明は、CICSAPI プログラムへの関連に限定するように変更されました。変更された説明の全文については、「*CICS System Programming Reference*」を参照してください。

CEMT の変更点

OPENAPI サポートには、以下の変更点があります。

CEMT INQUIRE DISPATCHER

CEMT INQUIRE DISPATCHER コマンドの ACTOPENTCBS および MAXOPENTCBS の説明は、L9 モード TCB を含み、OPENAPI プログラムを参照するように変更されました。変更された説明の全文については、「*CICS Supplied Transactions*」を参照してください。

CEMT INQUIRE PROGRAM

CEMT INQUIRE PROGRAM コマンドに、新規オプション APIST が追加されました。APIST を使用すると、CICSAPI または OPENAPI を指定できます。APIST の説明の全文については、「*CICS Supplied Transactions*」を参照してください。

CEMT INQUIRE SYSTEM

CEMT INQUIRE SYSTEM コマンドの FORCEQR は OPENAPI プログラムには適用されないため、FORCEQR の説明は、CICSAPI プログラムへの関連に限定するように変更されました。変更された説明の全文については、「*CICS Supplied Transactions*」を参照してください。

CEMT SET DISPATCHER

CEMT SET DISPATCHER コマンドの MAXOPENTCBS の説明は、L9 モ

ード TCB を含むように変更されました。変更された説明の全文については、「*CICS Supplied Transactions*」を参照してください。

CEMT SET SYSTEM

CEMT SET SYSTEM コマンドのオプション FORCEQR の cvda 値 FORCE は、OPENAPI プログラムには適用されないので、FORCE の説明は、CICSAPI プログラムへの関連に限定するように変更されました。変更された説明の全文については、「*CICS Supplied Transactions*」を参照してください。

第 9 章 XPLink サポート

Extra Performance Linkage (以降、XPLink と短縮名で呼ぶ) は、ハイパフォーマンスのサブルーチン呼び出し機構およびリターン機構を提供する z/OS 機能です。これにより、実行パスの長さが短縮され、高度に最適化されます。

オブジェクト指向プログラミングは、オブジェクトへ「メッセージ」を送信し、それがそのオブジェクトに何らかのアクションを実行させるという概念を基礎として構築されています。メッセージ送信活動は、サブルーチン呼び出しとしてインプリメントされています。サブルーチン (C++ 用語ではメンバー関数と呼ばれる) は、通常小規模なコードです。典型的な C++ プログラムの特性を示す実行フローは、小規模なコードへの数多くのサブルーチン呼び出しで構成されます。このような性質のプログラムでは、XPLink 最適化テクノロジーを利用するメリットがあります。

MVS™ では、初期の System/360™ までさかのぼることのできる標準的なサブルーチン呼び出し規則があります。この規則は、サブルーチンが複雑な環境向けに最適化されていました。サブルーチンは比較的数量が少なく、それらが呼び出される回数も比較的少ないものでした。オブジェクト指向プログラミング規則によって、この点が変わりました。サブルーチンは単純になりましたが、数が多く、サブルーチン呼び出しの頻度ははるかに高くなりました。サブルーチンのサイズ、数、および使用パターンのこのような変化のため、関係するシステム・オーバーヘッドを最適化することが望ましくなりました。XPLink は、この最適化を行った結果です。

注:

z/OS 1.4 以上および CICS 3.1 以上では、ここで説明するように CICS アプリケーション・プログラムで XPLINK コンパイラー・オプションを使用できます。z/OS では使用できないとする情報は無効になります。

z/OS の C および C++ のマニュアルでは、XPLINK コンパイラー・オプションが CICS アプリケーション・プログラムでは使用不可であると記載されています。以前は使用不可であったためです。現在これらのマニュアルを変更していますが、お手元のマニュアルは、この変更が適用される前に発行されたものである可能性があります。

XPLink、X8 TCB および X9 TCB

CICS は、CICS オープン・トランザクション環境 (OTE) テクノロジーの複数 TCB 機能を使用することによって、XPLINK オプションを使用してコンパイルされた C および C++ プログラムをサポートします。X8 モード TCB は CICS キーにおいて、X9 モード TCB は USER キーにおいて、XPLink タスクをサポートするように定義されています。XPLink プログラムの各インスタンスは、1 つの X8 TCB または X9 TCB を使用します。

XPLink を使用するには、C または C++ アプリケーション・コードが再入可能およびスレッド・セーフでなければなりません。同じコード・インスタンスを複数の MVS TCB で実行可能であり、共用リソースを保護するスレッド・セーフ機構がない場合は、アプリケーション・コードの振る舞いは予測不能です。このことは特に認識しておく必要があります。

CICS 環境での C および C++ プログラム (XPLINK オプションを使用してコンパイルする) の作成

アプリケーション開発者は、CICS XPLink サポートを活用するために以下を実行することを求められます。

- スレッド・セーフ・プログラミング原則および技法を厳守してコードを作成します。
- XPLINK オプションをオンに設定して C または C++ プログラムをコンパイルします。
- PROGRAM リソース定義で、プログラムがスレッド・セーフであることを指示します。
- CEEUOPT または #pragma での CICSVAR の使用を検討します (詳細については「*CICS Application Programming Guide*」を参照)。

CICS XPLink サポートを使用するすべてのプログラムは、再入可能およびスレッド・セーフでなければなりません。アプリケーション開発者のみが、特定のアプリケーションのコードがこれらの要件を満たすことを保証できます。

XPLink および非 XPLink 間のコントロールの 引き渡し

XPLink オブジェクトから非 XPLink オブジェクトへコントロールが転送されるか、またはその逆方向に転送されるごとに、QR TCB およびオープン TCB (X8 TCB または X9 TCB のいずれか) 間の切り替えが発生します。パフォーマンスの観点からは、TCB の切り替えはコストがかかるため、このパフォーマンス上のオーバーヘッドを考慮する必要があります。

XPLink オブジェクトは、EXEC CICS インターフェースまたは言語環境®プログラム・インターフェースのいずれかを使用して、非 XPLink オブジェクトを呼び出すことができます。

非 XPLink オブジェクトは、EXEC CICS インターフェースを使用する XPLink オブジェクトのみを呼び出すことができます。このような呼び出しの言語環境プログラム・インターフェースはサポートされません。

XPLINK オプションを使用してコンパイルされたオブジェクトの CICS サポートを獲得するための CICS 定義の変更

PROGRAM リソース定義内で、プログラムがスレッド・セーフであることを示すだけで、XPLINK オプションを使用してコンパイルされたプログラムの CICS サポートを利用することができます。X8 TCB または X9 TCB にタスクを配置するには、このような指示と、ロード・モジュール内の XPLink 「シグニチャー」のみが必要です。

特定プログラムに適切な TCB を選択する際、PROGRAM リソース定義の API 属性に OPENAPI が存在しても、XPLink の方が優先されます。

XPLink サポート の利点

XPLink サポートは、パフォーマンスおよび機能の両方で利点を提供します。

- パフォーマンス上の利点は、最適化サブルーチン・リンケージ・テクノロジーによって得られます。
 - 機能上の利点は、以下によってもたらされます。
 - CICS、TSO/Batch、IMS™、または Unix システム・サービスの下で実行されるプログラムが使用または起動可能な、共通モジュールまたは DLL の開発を開始することができます。
 - C++ 開発者は、C++ Standard Template Library をさらに十分に活用することもできます。
- これらを組み合わせることで、C/C++ コードの再使用可能性がさらに高まります。

要件

XPLink サポートには、ハードウェア、ソフトウェア、またはリソース使用に関して、この CICS リリースの一般要件以外の特別な要件はありません。

プログラミング・スタイル

CICS XPLink サポートを使用するすべてのプログラムは、再入可能およびスレッド・セーフでなければなりません。

CICS 外部インターフェースの変更点

インストールの変更点

C および C++ プログラムの XPLINK コンパイラー・オプションがサポートされるため、言語環境プログラムが必要とするライブラリーを CICS に対して定義する方法に変更があります。必要なライブラリーは、SCEERUN および SCEERUN2 です。

- ライブラリー SCEERUN2 は、STEPLIB 連結および DFHRPL 連結の両方で定義する必要があります。
- SCEERUN2 は、どちらの連結でも SCEERUN よりも前に置く必要があります。

詳しくは、「インストール・ガイド」の『言語環境プログラムの CICS サポートのインストール (Installing CICS support for Language Environment)』を参照してください。詳しくは、「*CICS System Programming Reference*」を参照してください。

システム初期設定パラメーターの変更点

XPLink サポート用に新しいシステム初期設定パラメーターが追加されました。新しいパラメーターは MAXXPTCBS です。

MAXXPTCBS={5vnumber}

CICS 領域内に同時に存在できるオープン X8 TCB およびオープン X9 TCB の最大数を表示します。X8 および X9 は、XPLINK コンパイラー・オプションを使用してコンパイルされた C および C++ プログラムを実行するために使用される TCB です。X8 TCB は、CICS キーのプログラムに使用され、X9 モード TCB はユーザー・キーのプログラムに使用されます。

CICS タスクは、必要な数の X8 TCB および X9 TCB を使用することができ、プログラムの終了までの間のみ、これらの TCB が保持されます。

リソース定義の変更点

XPLink サポート用のリソース定義に変更はありません。ただし、XPLink フラグを設定してコンパイルしたプログラムが、対応するプログラム定義内で CONCURRENCY 属性を THREADSAFE に設定していることを確認する必要があります。

アプリケーション・プログラミング・インターフェースの変更点

EXEC CICS HANDLE

C および C++ コードは、EXEC CICS HANDLE 「ファミリー」のすべての API から、EXEC CICS HANDLE ABEND PROGRAM のみを使用するように制限されています。この制限は、XPLink のサポートでも継続しており、C および C++ 変換プログラムによって監視されています。

その他の EXEC CICS の制限

EXEC CICS RETURN、EXEC CICS XCTL、および EXEC CICS SEND PAGE RELEASE によって、ユーザー・コードから CICS へコントロールが直接転送されます。C++ では、これらの API は、C++ オブジェクトのデストラクターが適切に駆動されるように、非常に慎重に使用する必要があります。オブジェクト・デストラクターの駆動に失敗すると、ストレージがリークする、開いたファイルが開いたままになる、ロックがロックされたままになるなどの問題が発生する可能性があります。C++ では、通常、オブジェクト作成時にリソースを取得し、オブジェクト破棄時にリソースを解放します。スタックが割り振られたオブジェクトのデストラクターは、それらのオブジェクトがスコープを外れると、自動的に呼び出されます。一部の C++ オブジェクトは、静的に作成される場合もあります。すなわち、「main」関数に入る前に作成され、「main」関数が終了するか、または「exit」呼び出しが起動されると破棄されます。

マルチスレッド化

XPLink フラグを使用してコンパイルされ、ここで説明するその他の基準を満たす C および C++ の CICS アプリケーション・プログラムは、それら独自の TCB 上で実行されるため、アプリケーション開発者は、C、C++、および POSIX API の完全なセットも使用可能であると考えられるかもしれません。実際にはそうではありません。

C または C++ で書かれた CICS アプリケーション・プログラムは、アプリケーション・コード内でマルチスレッド化技法を使用すべきではありません。このような技法を使用しないことをお勧めします。

マルチスレッド化技法とは、例えば fork() ステートメントまたは pthreads を使用するなど、アプリケーション内部で複数の実行パスを作成するコーディングの形式を意味します。これらの技法は、CICS 環境ではテストされておらず、CICS ではサポートされないものと見なされます。IBM は、このような技法の使用に関連する可能性のある問題報告書を受け入れません。

OPEN TCB および EDF

CEDF 自体がスレッド・セーフでないため、プログラムが通常 OPEN TCB (L8、L9、X8、または X9) を使用して実行される場合でも、CEDF では QR TCB の使用を強制します。

システム・プログラミング・インターフェースの変更点

INQUIRE DISPATCHER

EXEC CICS INQUIRE DISPATCHER コマンドは、ACTXPTCBS オプションおよび MAXXPTCBS オプションを含むように変更されました。

ACTXPTCBS(value)

現在アクティブな (すなわちユーザー・タスクに割り振られた) X8 モードおよび X9 モードのオープン TCB の数を表示します。

MAXXPTCBS(value)

CICS 領域内に同時に存在できる X8 モードおよび X9 モードのオープン TCB の最大数を表示します。値は、1 から 999 の範囲で指定することができます。この値は、別の値を入力して変更するとリセットできます。

INQUIRE PROGRAM

EXEC CICS INQUIRE PROGRAM コマンドは、Runtime オプションに XPLink 値を含めるように変更されました。

Runtime

プログラムのランタイム環境に関する情報を表示します。このリストの新しい値は、次のとおりです。

XPLink

プログラムは、XPLINK オプションを使用してコンパイルされた C または C++ プログラムです。

SET DISPATCHER

EXEC CICS SET DISPATCHER コマンドは、MAXXPTCBS オプションを含むように変更されました。

MAXXPTCBS(value)

CICS 領域内に同時に存在できる X8 モードおよび X9 モードのオープン TCB の最大数を指定します。値は 1 から 999 の範囲で指定できます。MAXXPTCBS を以前定義された値から小さくした場合、新しい値が現在割り振られているオープン TCB の数より小さいのであれば、CICS は TCB を切り離して、それらがユーザー・タスクによって解放された場合のみ新しい制限を適用します。トランザクションが異常終了して、新しい制限を適用するために TCB を切り離せるようにすることはありません。キューに入られて X8 モードおよび X9 モードの TCB を待機するタスクが存在する

とき、MAXXPTCBS を以前定義された値から増やした場合、CICS は新規 TCB を接続し、新しい制限を上限として、キューに入れられたそれぞれのタスクを再開します。

XPLINK cvda

XPLink を表す新しい CVDA 値 1069 が追加されました。

CEMT の変更点

CEMT INQUIRE DISPATCHER

CEMT INQUIRE DISPATCHER コマンドは、ACTXPTCBS オプションおよび MAXXPTCBS オプションを含むように変更されました。

ACTXPTCBS(value)

現在アクティブな (すなわちユーザー・タスクに割り振られた) X8 モードおよび X9 モードのオープン TCB の数を表示します。

MAXXPTCBS(value)

CICS 領域内に同時に存在できる X8 モードおよび X9 モードのオープン TCB の最大数を表示します。値は、1 から 999 の範囲で指定することができます。この値は、別の値を入力して変更するとリセットできます。

CEMT INQUIRE PROGRAM

CEMT INQUIRE PROGRAM コマンドは、Runtime オプションに XPLink 値を含めるように変更されました。

Runtime

プログラムのランタイム環境に関する情報を表示します。このリストの新しい値は、次のとおりです。

XPLink

プログラムは、XPLINK オプションを使用してコンパイルされた C または C++ プログラムです。

CEMT SET DISPATCHER

CEMT SET DISPATCHER コマンドは、MAXXPTCBS オプションを含むように変更されました。

MAXXPTCBS(value)

CICS 領域内に同時に存在できる X8 モードおよび X9 モードのオープン TCB の最大数を指定します。値は 1 から 999 の範囲で指定できます。MAXXPTCBS を以前定義された値から小さくした場合、新しい値が現在割り振られているオープン TCB の数より小さいのであれば、CICS は TCB を切り離して、それらがユーザー・タスクによって解放された場合のみ新しい制限を適用します。トランザクションが異常終了して、新しい制限を適用するために TCB を切り離せるようにすることはありません。キューに入れられて X8 モードおよび X9 モードの TCB を待機するタスクが存在する

とき、MAXXPTCBS を以前定義された値から増やした場合、CICS は新規 TCB を接続し、新しい制限を上限として、キューに入れられたそれぞれのタスクを再開します。

グローバル・ユーザー出口ルーチンの変更点

XPCFTCH

XPLINK オプションを使用してコンパイルされた C および C++ プログラムから、出口 XPCFTCH が呼び出されると、変更されたエントリー・ポイント・アドレス (出口で指定されている場合) が無視されることを示すフラグが設定されます。

XPCTA

XPLINK オプションを使用してコンパイルされた C および C++ プログラムから、出口 XPCTA が呼び出されると、再開アドレス (出口で指定されている場合) が無視されることを示すフラグが設定されます。

DFHUEPAR

UEPTX8 および UEPTX9 という 2 つのシンボル値が、DFHUEPAR 内の TCB 標識のテーブルに追加されました。

ユーザーが置換可能なプログラムの変更点

ユーザーが置換可能な新規のプログラム **DFHAPXPO** が提供されます。

DFHAPXPO は、XPLINK を使用してコンパイルされた C または C++ プログラムを実行する予定の各言語環境プログラム・エンクレーブの PIPI 事前初期設定フェーズの間にロードされます。これにより、デフォルトの言語環境プログラムのランタイム・オプションを変更することができます。リセット可能な言語環境プログラムのオプションの詳細については、「*z/OS Version 1.4 Language Environment Programming Guide*」、SC22-7561 を参照してください。プログラムは、アセンブラ一言語で作成する必要があります。

モニターの変更点

「CICS パフォーマンス・ガイド」のトピック『待ち (中断) 時間に関する注』で、以下のように新規項目が表に追加されています。

表 10. パフォーマンス・クラスの待機 (中断) フィールド

フィールド ID	グループ名	説明
282	DFHTASK	CICS MAXXPTCBS 遅延時間

パフォーマンス・クラス・モニター・レコードに、新規フィールドが追加されました。これらはグループ DFHTASK への追加です。また、そのグループ内の既存の項目の説明にいくつかの変更があります。

新規項目

271 (TYPE-S, "X8CPUT", 8 BYTES)

ユーザー・タスクが、CICS X8 モードの TCB 上の CICS ディスパッチ

チャー・ドメインによってディスパッチされている間のプロセッサ時間。XPLINK オプションを使用してコンパイルされ、EXECKEY=CICS を使用して定義された C および C++ プログラムを、トランザクションが起動するとき、トランザクションは CICS X8 モード TCB を割り振られ、この TCB を使用します。(プログラムが EXECKEY=USER で定義されていても、ストレージ保護ファシリティが非アクティブの場合は、X8 モードの TCB を割り振ることもできます。) タスクに X8 モードの TCB が割り振られると、プログラムが完了するまで、その TCB とそのタスクとの間の関連付けがそのまま維持されます。

272 (TYPE-S, "X9CPUT", 8 BYTES)

ユーザー・タスクが、CICS X9 モードの TCB 上の CICS ディスパッチャー・ドメインによってディスパッチされている間のプロセッサ時間。XPLINK オプションを使用してコンパイルされ、EXECKEY=USER を使用して定義された C および C++ プログラムを、トランザクションが起動するとき、トランザクションは CICS X9 モード TCB を割り振られ、この TCB を使用します。(ストレージ保護ファシリティが非アクティブの場合、X9 モードの TCB ではなく、X8 モードの TCB が使用されます)。タスクに X9 モードの TCB が割り振られると、プログラムが完了するまで、その TCB とそのタスクとの間の関連付けがそのまま維持されます。

282 (TYPE-S, "MAXXTDLY", 8 BYTES)

CICS システムが、システム・パラメーター MAXXPTCBS によって設定されている制限に達したために、ユーザー・タスクが CICS XP TCB (X8 または X9 モード) を取得するために待っている間の経過時間。X8 モードおよび X9 モードのオープン TCB を使用するのには、XPLINK オプションを使用してコンパイルされた C および C++ プログラムのみです。

注: このフィールドは、タスク一時停止時間フィールド **SUSPTIME** (グループ名: DFHTASK、フィールド ID: 014) のコンポーネントです。

変更された項目

007 (TYPE-S, "USRDISPT", 8 BYTES)

TCB モードのリストに X8 および X9 が追加されました。

008 (TYPE-S, "USRCPUT", 8 BYTES)

TCB モードのリストに X8 および X9 が追加されました。

262 (TYPE-S, 'KY8DISPT', 8 BYTES)

合計経過時間を構成する項目のリストに対して、次の項目が追加されず。

- XPLINK オプションを使用してコンパイルされ、EXECKEY=CICS を使用して定義された C および C++ プログラムを、トランザクションが起動するとき、トランザクションは CICS X8 モード TCB を割り振られ、その TCB にディスパッチされます。プログラムが終了するまで、TCB はタスクに関連付けられたままです。

263 (TYPE-S,'KY8CPUT',8 BYTES)

プロセッサ時間を構成する項目のリストに対して、次の項目が追加されます。

- XPLINK オプションを使用してコンパイルされ、EXECKEY=CICS を使用して定義された C および C++ プログラムを、トランザクションが起動するとき、トランザクションは CICS X8 モード TCB を割り振られ、その TCB にディスパッチされます。プログラムが終了するまで、TCB はタスクに関連付けられたままです。

264 (TYPE-S, "KY9DISPT", 8 BYTES)

合計経過時間を構成する項目のリストに対して、次の項目が追加されます。

- XPLINK オプションを使用してコンパイルされ、EXECKEY=USER を使用して定義された C および C++ プログラムを、トランザクションが起動するとき、トランザクションは CICS X9 モード TCB を割り振られ、その TCB にディスパッチされます。プログラムが終了するまで、TCB はタスクに関連付けられたままです。

265 (TYPE-S, "KY9CPUT", 8 BYTES)

プロセッサ時間を構成する項目のリストに対して、次の項目が追加されます。

- XPLINK オプションを使用してコンパイルされ、EXECKEY=USER を使用して定義された C および C++ プログラムを、トランザクションが起動するとき、トランザクションは CICS X9 モード TCB を割り振られ、その TCB にディスパッチされます。プログラムが終了するまで、TCB はタスクに関連付けられたままです。

すべての新規フィールドおよび変更されたフィールドは、モニター管理テーブル (MCT) に DFHMCT TYPE=RECORD 項目をコーディングすることにより、モニター・レコードから除外することができます。

統計の変更点

DFHDSGDS DSECT によってマップされる「ディスパッチャー・ドメイン: TCB モード統計」への変更があります。

変更により、X8 モード TCB、X9 モード TCB、および TCB の XP プールが、この表の項目の説明に追加されました。

表 11. 「ディスパッチャー・ドメイン: TCB モード統計」の変更されたフィールド

DFHSTUP 名	フィールド名	説明
TCB モード	DSGTCBNM	CICS ディスパッチャー TCB モードの名前であり、QR、RO、CO、SZ、RP、FO、SL、SO、S8、D2、JM、L8、L9、J8、J9、X8、または X9 のいずれかです。 <u>リセット特性:</u> リセットなし

表 11. 「ディスパッチャー・ドメイン: TCB モード統計」の変更されたフィールド (続き)

DFHSTUP 名	フィールド名	説明
TCB プール	DSGTCBMP	この CICS ディスパッチャー TCB モードが定義された TCB プールの名前であり、N/A、OPEN、JVM、または XP のいずれかです。 <u>リセット特性:</u> リセットなし

マイグレーション

既存の機能のマイグレーション

XPLink を使用せずに C および C++ の使用を継続する場合、特にアクションは必要ありません。

新規機能へのマイグレーション

C および C++ プログラムの XPLink コンパイラー・オプションで使用可能になったサポートを利用するには、以下の点を考慮してください。

- C または C++ プログラムが再入可能およびスレッド・セーフであることを確認するか、またはこれらの基準に従うように変更します。「マイグレーション・ガイド」の『スレッド・セーフ・プログラミングおよびオープン・トランザクション環境 (OTE) のマイグレーション計画』を参照してください。
- プログラムが XPCFTCH または XPCTA 出口を使用する場合、「CICS *Application Programming Guide*」の『Global User exits and XPLink』に記載されている以下の情報に注意してください。
 - CICS は、XPCFTCH がエントリー・ポイントを変更しようとする試みをすべて無視します。
 - CICS は、XPCTA が再開アドレスを定義しようとする試みをすべて無視します。

このような要件を管理する他の方法を見つけるか、またはこのプログラムは XPLINK 最適化の適切な候補ではないと結論付ける必要があります。

- XPLINK コンパイラー・オプションを使用してプログラムを再コンパイルします。
- このプログラムの PROGRAM リソース定義の並行性属性を更新し、値をスレッド・セーフに設定します。

CICSplex SM サポート

新しい XPLINK C++ のサポートに対応するために、CICSplex SM Web ユーザー・インターフェースおよび API に対していくつかの変更が行われました。

CICSplex SM アプリケーション・プログラミング・インターフェースの変更点

リソース・テーブルの変更点

以下のリソース・テーブルに対して変更が加えられました。

- 『PROGRAM リソース・テーブル』
- 『PROGDEF リソース・テーブル』
- 『TASK リソース・テーブル』
- 『CICSRGN リソース・テーブル』

PROGRAM リソース・テーブル

PROGRAM リソース・テーブルには、以下の属性が含まれます。

RUNTIME

CVDA (XPLINK) 値 1068 を示します。

APIST CICSAPI または OPENAPI の API 状況を表す CVDA 値を示します。

PROGDEF リソース・テーブル

PROGDEF リソース・テーブルには、以下の属性が含まれます。

API CICSAPI または OPENAPI の EYUDA 値を示します。

TASK リソース・テーブル

TASK リソース・テーブルには、以下の属性が含まれます。

TMRL9CPU

ユーザー・タスク L9 モード CPU 時間を示します。

CICSRGN リソース・テーブル

CICSRGN リソース・テーブルには、以下の SPI 属性が含まれています。

MAXXPTCBS

XPLink OTE X8/X9 モード・プール内の TCB の現在の最大数を表します。

ACTXPTCBS

XPLink OTE X8/X9 モード・プール内の TCB の実際の数を表します。

MAXSSLTCBS

SSL OTE プール内の TCB の現在の最大数を表します。

ACTSSLTCBS

現在割り振られている SSL プール TCB の数を表します。

既存の DISPATCHER SPI 属性 MAXHPTCBS および ACTHPTCBS はサポートされなくなり、NOTVALID というフラグが立てられます。

#

CICSplex SM Web ユーザー・インターフェースの変更点

以下の WUI ビューに対して変更が加えられました。

- 『プログラム・ビュー』
- 『プログラム定義ビュー』
- 『CPU と TCB 情報ビュー』

プログラム・ビュー

プログラム (EYUSTARTPROG.DETAILED) ビューに、以下の属性が追加されました。

APIST API 状況

プログラム定義ビュー

プログラム定義 (EYUSTARTPROGDEF.DETAILED) ビューに、以下の属性が追加されました。

API アプリケーション・プログラミング・インターフェース

CPU と TCB 情報ビュー

タスク操作ビュー内部のアクティブ・タスク・ビュー・セット (EYUSTARTTASK.DETAIL9) の 1 つである CPU と TCB 情報ビューに、以下の属性が追加されました。

TMRL9CPU

ユーザー・タスク L9 モード CPU 時間

第 10 章 言語環境プログラムに準拠したアセンブラー MAIN プログラムのサポート

言語環境プログラムに準拠したアセンブラー MAIN プログラムを作成できるようになりました。

言語環境プログラムに準拠したアセンブラー MAIN プログラム

これまで CICS 内で言語環境プログラムに準拠したアセンブラー・プログラムを使用する唯一の方法は、COBOL、PLI、または C 言語環境プログラムに準拠したプログラムからの呼び出しを使用して、アセンブラー・プログラムを高水準言語 (HLL) プログラムとリンク・エディットすることでした。この方法では、アセンブラー・プログラムが言語環境プログラムのサブルーチンになっていました。そのため、CEEENTRY で MAIN=NO とする必要がありました。ユーザーは、NOPROLOG および NOEPILOG を指定して、CEEENTRY および CEETERM 呼び出しを別々にコード化する必要がありました。CICS PROGRAM リソースは、ASM としても LE370 としても定義することはできませんでした。

CICS では、言語環境プログラムに準拠したアセンブラー MAIN プログラムのコード化がサポートされるようになりました。新規変換プログラム・オプション LEASM を使用すると、プログラムの環境のセットアップに言語環境プログラム機能を使用されます。このようなプログラムは、DFHEAI ではなくスタブ DFHELII とリンク・エディットする必要があります。

このサポートによって、アセンブラー・プログラム用デバッガーも使用できるようになりました。

言語環境プログラムに準拠したアセンブラー MAIN プログラムのサポートの利点

言語環境プログラムに準拠したアセンブラー MAIN プログラムのサポートは、言語環境プログラムの使用可能性を拡張して、デバッガー・サポートをこれらのプログラムで使用できるようにします。

要件

言語環境プログラムに準拠したアセンブラー MAIN プログラムのサポートには、ハードウェア、ソフトウェア、またはリソース使用に関して、この CICS リリースの一般要件以外の特別な要件はありません。

CICS 外部インターフェースの変更点

リソース定義の変更点

PROGRAM リソース定義の Language 属性は、指定可能な一部の値の説明が変更されました。

Language({COBOL|ASSEMBLER|LE370|PLI})

以下のプログラム言語を指定します。

ASSEMBLER

これは、LEASM 変換プログラム・オプションを使用した変換が行われなかったアセンブラー言語プログラムです。LEASM は、言語環境プログラムに準拠した MAIN プログラムとなるようにアセンブラー・プログラムを変換する場合に使用します。

LE370 このプログラムは、複数言語サポートを活用しており、言語環境プログラムに準拠したコンパイラーによってコンパイルされたプログラムであるか、あるいは、言語環境プログラムに準拠した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 プログラムです。

アプリケーション・プログラミング・インターフェースの変更点

言語の制約事項

アセンブラーでプログラミングし、LEASM オプションを使用したプログラムの変換を計画している場合は、すべてのアセンブラー・プログラムに適用される制約事項に加えて、以下の制約事項があります。

- レジスター 2 は、コード・ベース・レジスターとして使用することはできません。
- レジスター 12 は言語環境プログラムによって予約されており、言語環境プログラムの共通アンカー域 (CAA) を指すために使用されるので、適切に保管され復元されなければ、プログラムがこのレジスターを使用することはできません。
- レジスター 13 は、唯一の作業用ストレージ・ベース・レジスターとして使用する必要があります。
- このプログラムは、グローバル・ユーザー出口プログラム (GLUE) やタスク関連ユーザー出口プログラム (TRUE) にすることはできません。
- プログラムは、AMODE(24) コードを使用したり、AMODE(24) コードに依存したりすることはできません。

変換プログラム・オプション

新規変換プログラム・オプション LEASM (アセンブラー・プログラムにのみ有効) を使用することにより、ユーザーはマクロを使用してこのプログラムを変換するように指定できます。このマクロによって、プログラムは言語環境プログラムに準拠したプログラムになり、MAIN プログラムとしてアセンブルするための準備ができます。

LEASM を指定すると、新規アセンブラー・グローバル &DFHEILE が設定されます。変換プログラムがあらゆる出力ファイルの先頭に挿入する DFHEIGBL マクロに、第 4 定位置パラメーター LE が追加されました。LEASM が指定された場合、DFHEIGBL は &DFHEILE をオンに設定するように変更されます。

LEASM が指定されている場合、変換プログラムは ARG0 に言語として値 (X'12') を設定します。変換プログラムからの出力は、その他の方法で LEASM を指定せずに生成された場合と同じです。

&DFHEILE が設定された場合は、通常の CICS 環境ではなく LE 環境を作成するために、DFHEISTG、DFHEIENT、DFHEIRET、および DFHEIEND マクロがそれぞれ別々に展開されます。これにより、NOPROLOG および NOEPILOG を使用し、かつ独自の DFHEIENT やその他のマクロをコード化したプログラムは、そのプログラム・ソースを変更することなく言語環境プログラム・サポートを利用できます。例えば、複数のコード・ベース・レジスターを必要とするすべてのプログラムはこのカテゴリーに分類されます。というのも、変換プログラムは複数のコード・ベース・レジスターをサポートしていないためです。

DFHEISTG

&DFHEILE が設定された場合は、DFHEISTG の先頭の、DSECT ステートメントの直後にステートメント CEEDSA SECTYPE=OS が追加されます。これにより、DSECT ステートメントがなくても、標準の言語環境プログラム DSA が組み込まれます。

DFHEIENT

DFHEIENT は言語環境プログラム CEEENTRY マクロを生成します。このマクロは、標準 CICS メソッドを使用する代わりに、コードのアドレス可能度および作業用ストレージをセットアップします。

コードのアドレス可能度

DFHEIENT には CODEREG パラメーターがあり、その値はデフォルトで 3 に設定されます。変換プログラム・オプション LEASM が指定されると、DFHEIENT によって CEEENTRY が生成され、BASE パラメーターには CODEREG 値がそのままコピーされます。

CEEENTRY では、コードのアドレス可能度にレジスター 2 または 12 を使用することはできません。レジスター 12 には、言語環境プログラムの実行中は常に言語環境プログラム CAA のアドレスが格納されていない限りなりません。レジスター 2 は、TRT などの命令によって変更されるので、通常はコード・ベースとしては使用されません。LEASM オプションを使用して変換されるプログラムで、CODEREG にレジスター 2 またはレジスター 12 が指定された場合、DFHEIENT はエラー・メッセージを生成します。

作業用ストレージのアドレス可能度

DFHEIENT には DATAREG パラメーターがあり、その値はデフォルトで 13 に設定されます。CEEENTRY では、作業用ストレージとして R13 以外のレジスターは許可されません。また、複数の作業用ストレージ・レジスターはサポートされていません。

DFHEIENT では、LEASM オプションを使用して変換されるプログラムで、DATAREG に 13 以外のものを使用することは禁止されています。

DFHEIRET

LEASM オプションを使用して変換されるプログラムに対して CEETERM RC=0 を生成します。

EXAMPLE LEASM PROGRAM

簡単な CICS アセンブラー・プログラムを以下に示します。

```
*ASM XOPTS(LEASM)
DFHEISTG DSECT
OUTAREA DS CL200 DATA OUTPUT AREA
*
EIASM CSECT ,
MVC OUTAREA(40),MSG1
MVC OUTAREA(4),EIBTRMID
EXEC CICS SEND TEXT FROM(OUTAREA) LENGTH(43) FREEKB ERASE
EXEC CICS RECEIVE
MVC OUTAREA(13),MSG2
EXEC CICS SEND TEXT FROM(OUTAREA) LENGTH(13) FREEKB ERASE
EXEC CICS RETURN
*
MSG1 DC C'xxxx: ASM program invoked. ENTER TO END.'
MSG2 DC C'PROGRAM ENDED'
END
```

変換され、組み立てられると、以下のようになります。

```
*ASM XOPTS(LEASM)
DFHEIGBL ,,,LE INSERTED BY TRANSLATOR
*,&DFHEIDL; SETB 0 1 MEANS EXEC DLI IN PROGRAM 01-DFHEI
*,&DFHEIDB; SETB 0 1 MEANS BATCH PROGRAM 01-DFHEI
*,&DFHEIRS; SETB 0 1 MEANS RSECT 01-DFHEI
*,&DFHEILE; SETB 1 1 MEANS LE MAIN 01-DFHEI
DFHEISTG DSECT
DFHEISTG INSERTED BY TRANSLATOR
*****
* EXEC INTERFACE DYNAMIC STORAGE *
*****
DFHEISTG DSECT EXEC INTERFACE STORAGE @BBAC81A 01-DFHEI
USING *,DFHEIPLR ESTABLISH ADDRESSABILITY @BBAC81A 01-DFHEI
*
*****
* D Y N A M I C S T O R A G E A R E A ( D S A ) *
*****
*
CEEDSA DS 0D Just keep the same label for formulae 02-CEEDS
*
CEEDSAFLAGS DS XL2 DSA flags 02-CEEDS
CEEDSALNGC EQU X'1000' C library DSA 02-CEEDS
CEEDSALNGP EQU X'0800' PL/I library DSA 02-CEEDS
CEEDSAEXIT EQU X'0008' An Exit DSA 02-CEEDS
CEEDSAMEMD DS XL2 Member defined 02-CEEDS
CEEDSABKC DS A Addr of DSA of caller 02-CEEDS
CEEDSAFWC DS A Addr of DSA of last called rtn 02-CEEDS
```

```

*****
*
* CONTROL BLOCK NAME = DFHEIBLK
*
* NAME OF MATCHING PL/AS CONTROL BLOCK = None
*
* DESCRIPTIVE NAME = %PRODUCT EXEC Interface Block.
*
* @BANNER_START 02
* Licensed Materials - Property of IBM
*
* "Restricted Materials of IBM"
*
* 5697-E93
*
* (C) Copyright IBM Corp. 1990, 1993
*
*
* @BANNER_END
*
* STATUS = %XA20
*
* FUNCTION = EXEC Interface Block.
*
* The exec interface block contains information on the
* transaction identifier, the time and date, and the cursor
* position on a display device. Some of the other fields are
* set indicating the next action that a program should take
* in certain circumstances.
* DFHEIBLK also contains information that will be helpful
* when a dump is being used to debug a program.
* This control block is included automatically by an
* application program using the command-level interface.
* EISEIBA in the EIS addresses the EIB.
*
*
* NOTES :
* DEPENDENCIES = S/370
* MODULE TYPE = Control block definition
* PROCESSOR = Assembler
*
*-----*
*
* CHANGE ACTIVITY :
* £SEG(DFHEIBLK),COMP(COMMAND),PROD(%PRODUCT) :
*
* PN= REASON REL YYMMDD HDXXIII : REMARKS
* £L1= 550 %0G 900515 HDFSPC : Add an EIB length equate
* £D1= I05119 %B1 930226 HDDHDMA : Correct comments for date field
* £P1= M60581 %B0 900116 HDAEGB : Change for PLXMAP to data areas
*
*****
* EXEC INTERFACE BLOCK
*****
DFHEIBLK DSECT EXEC INTERFACE BLOCK @BBAC81A 01-DFHEI
USING *,DFHEIBR @BBAC81A 01-DFHEI

```

EIBTIME	DS	PL4	TIME IN @HHMMSS FORMAT	@BBAC81A	01-DFHEI
EIBDATE	DS	PL4	DATE IN @CYDDDD+ FORMAT,	@D1C	01-DFHEI
*			where C is the century	@D1A	
*			indicator (0=1900, 1=2000),	@D1A	
*			YY is the year, DDD is the	@D1A	
*			day number and '+' is the	@D1A	
*			sign byte (positive)	@D1A	
EIBTRNID	DS	CL4	TRANSACTION IDENTIFIER	@BBAC81A	01-DFHEI
EIBTASKN	DS	PL4	TASK NUMBER	@BBAC81A	01-DFHEI
EIBTRMID	DS	CL4	TERMINAL IDENTIFIER	@BBAC81A	01-DFHEI
EIBRSVD1	DS	H	RESERVED	@BBAC81A	01-DFHEI
EIBCPOSN	DS	H	CURSOR POSITION	@BBAC81A	01-DFHEI
EIBCALEN	DS	H	COMMAREA LENGTH	@BBAC81A	01-DFHEI
EIBAID	DS	CL1	ATTENTION IDENTIFIER	@BBAC81A	01-DFHEI
EIBFN	DS	CL2	FUNCTION CODE	@BBAC81A	01-DFHEI
EIBRCODE	DS	CL6	RESPONSE CODE	@BBAC81A	01-DFHEI
EIBDS	DS	CL8	DATASET NAME	@BBAC81A	01-DFHEI
EIBREQID	DS	CL8	REQUEST IDENTIFIER	@BBAC81A	01-DFHEI
EIBRSRCE	DS	CL8	RESOURCE NAME	@BBDIAOU	01-DFHEI
EIBSYNC	DS	C	X'FF' SYNCPOINT REQUESTED	@BBDIAOU	01-DFHEI
EIBFREE	DS	C	X'FF' FREE REQUESTED	@BBDIAOU	01-DFHEI
EIBRECV	DS	C	X'FF' RECEIVE REQUIRED	@BBDIAOU	01-DFHEI
EIBSEND	DS	C	RESERVED	@BM13417	01-DFHEI
EIBATT	DS	C	X'FF' ATTACH RECEIVED	@BBDIAOU	01-DFHEI
EIBEOC	DS	C	X'FF' EOC RECEIVED	@BBDIAOU	01-DFHEI
EIBFMH	DS	C	X'FF' FMHS RECEIVED	@BBDIAOU	01-DFHEI
EIBCOMPL	DS	C	X'FF' DATA COMPLETE		01-DFHEI
EIBSIG	DS	C	X'FF' SIGNAL RECEIVED		01-DFHEI
EIBCONF	DS	C	X'FF' CONFIRM REQUESTED		01-DFHEI
EIBERR	DS	C	X'FF' ERROR RECEIVED		01-DFHEI
EIBERRCD	DS	CL4	ERROR CODE RECEIVED		01-DFHEI
EIBSYNRB	DS	C	X'FF' SYNC ROLLBACK REQ'D		01-DFHEI
EIBNODAT	DS	C	X'FF' NO APPL DATA RECEIVED		01-DFHEI
EIBRESP	DS	F	INTERNAL CONDITION NUMBER		01-DFHEI
EIBRESP2	DS	F	MORE DETAILS ON SOME RESPONSES		01-DFHEI
EIBRLDBK	DS	CL1	ROLLED BACK		01-DFHEI
*					
EIBLENG	EQU	*-EIBTIME	Length of EIB	@L1A	01-DFHEI

*			END OF EXEC INTERFACE BLOCK	*	

DFHEIBR	EQU	11	EIB REGISTER	@BA02936	01-DFHEI
			@01A 02-CEEEN		

```

*****
*          PROLOG CODE FOR EXEC INTERFACE          *
*****
*&DFHEICS; CEEENTRY PPA=DFHPPA,MAIN=YES,PLIST=OS,
*          BASE=&CODEREG;,
*          AUTO=(DFHEIEND-DFHEISTG)
TESTLE  CSECT ,                                02-CEEEN
TESTLE  RMODE ANY                              02-CEEEN
TESTLE  AMODE ANY                              02-CEEEN
        ENTRY TESTLE                          02-CEEEN
        PUSH USING                              02-CEEEN
        DROP , @02A                            02-CEEEN
        USING *,15                             02-CEEEN
        B CEEZ0007                              02-CEEEN
        DC X'00C3C5C5'                          02-CEEEN
CEEY0007 DC A((((DFHEIEND-DFHEISTG)+7)/8)*8)    X02-CEEEN
        . Size of automatic storage.
        DC A(DFHPPA-TESTLE) . Address of PPA for this program 02-CEEEN
        B 1(,15)                                02-CEEEN
CEEZ0007 EQU *                                02-CEEEN
        STM 14,12,CEEDSAR14-CEEDSA(13)         02-CEEEN
        L 2,CEEINPL0007                        5001D @01C 02-CEEEN
        L 15,CEEINT0007                       @01C 02-CEEEN
        DROP 15                               @01A 02-CEEEN
        BALR 14,15                            02-CEEEN
        LR 2,1                                02-CEEEN
        L 14,752(,12)                          02-CEEEN
        OI 8(14),X'80'                        02-CEEEN
        BALR 3,0                               @01A 02-CEEEN
        USING *,3
        L 3,CEE0EPV0007                       @01A 02-CEEEN
        POP USING                              @01A 02-CEEEN
        USING TESTLE,3                        @01A 02-CEEEN
        L 1,CEEDSANAB-CEEDSA(,13) Get the current NAB 02-CEEEN
        L 0,CEEY0007                          02-CEEEN
        ALR 0,1 Compute new value.             02-CEEEN
        CL 0,CEECAAEOS-CEECAA(,12) Compare with EOS. 02-CEEEN
        BNH CEEX0007                          02-CEEEN
        L 15,CEECAAGETS-CEECAA(,12) Get address overflow routine 02-CEEEN
        BALR 14,15 Get another stack segment. 02-CEEEN
        LR 1,15                               02-CEEEN
        B CEEX0007 Branch around statics @01A 02-CEEEN
CEEINPL0007 DC A(CEEINPL) @01A 02-CEEEN
CEEINT0007 DC V(CEEINT) @01A 02-CEEEN
CEE0EPV0007 DC A(TESTLE) @01A 02-CEEEN
CEEX0007 EQU *                                02-CEEEN
        ST 13,CEEDSABKC-CEEDSA(,1) Set back chain. 02-CEEEN
        ST 0,CEEDSANAB-CEEDSA(,1) Set new NAB value 02-CEEEN
        XC CEEDSAFLAGS-CEEDSA(,1),CEEDSAFLAGS-CEEDSA(1) . Clear 02-CEEEN
        ST 1,CEEDSAFWC-CEEDSA(,13) Set forward chain. 02-CEEEN
        LR 13,1 Set save area address 02-CEEEN
        USING CEEDSA,13 Addressability to SF V1R2M0 02-CEEEN
        MVC CEEDSALWS,CEECAALWS-CEECAA(12) Get LWS addr V1R2M0 02-CEEEN
        LR 1,2 02-CEEEN
        BAL 1,*+8 @L2A 01-DFHEI

```

```

* The following gives an assembler message if DFHEISTG is too big @P7A
  DS 0S((DFHEISTG+65264-DFHEIEND-4096)/4096) @04C 01-DFHEI
  DC AL2(DFHEIEND-DFHEISTG) LENGTH OF STORAGE @L2A 01-DFHEI
  DC H'0' Parameter list version number @P6C 01-DFHEI
*****
* ESTABLISH DATA ADDRESSIBILITY *
*****
DFHEIPLR EQU 13 PARAMETER LIST REGISTER @BBAC81A 01-DFHEI
LR DFHEIPLR,15 @BBAC81A 01-DFHEI
USING DFHEISTG,13 @BBAC81A 01-DFHEI
MVC DFHEIBP(L'DFHEIBP+L'DFHEICAP),0(1) @D3AX01-DFHEI
COPY EIB AND CA PTRS @D3A
*****
* ESTABLISH EIB ADDRESSIBILITY *
*****
L DFHEIBR,DFHEIBP @BBAC81A 01-DFHEI
USING DFHEIBLK,DFHEIBR @BBAC81A 01-DFHEI
*****
* END OF PROLOG CODE FOR EXEC INTERFACE *
*****
MVC OUTAREA(40),MSG1
MVC OUTAREA(4),EIBTRMID
* EXEC CICS SEND TEXT FROM(OUTAREA) LENGTH(43) FREEKB ERASE
DFHECALL =X'180660000800C20000082204000020',,(____RF,OUTAREA*
), (FB_2,=Y(43))
*****
DS 0H 01-DFHEC
LA 1,DFHEIPL 01-DFHEC
LA 14,=X'180660000800C20000082204000020' 01-DFHEC
SR 15,15 01-DFHEC
LA 0,OUTAREA 01-DFHEC
STM 14,0,0(1) 01-DFHEC
LA 14,=Y(43) 01-DFHEC
ST 14,12(,1) 01-DFHEC
OI 12(1),X'80' LAST ARGUMENT 01-DFHEC
L 15,=V(DFHEI1) 01-DFHEC
BALR 14,15 INVOKE EXEC INTERFACE 01-DFHEC
*****
* EXEC CICS RECEIVE
DFHECALL =X'040200000800000014000040000000'
*****

```

```

DS      0H                                01-DFHEC
LA      1,DFHEIPL                          01-DFHEC
LA      14,=X'040200000800000014000040000000' 01-DFHEC
ST      14,0(,1)                            01-DFHEC
OI      0(1),X'80'                          LAST ARGUMENT 01-DFHEC
L       15,=V(DFHEI1)                       01-DFHEC
BALR    14,15                               INVOKE EXEC INTERFACE 01-DFHEC
*****
MVC     OUTAREA(13),MSG2
*       EXEC CICS SEND TEXT FROM(OUTAREA) LENGTH(13) FREEKB ERASE
DFHECALL =X'180660000800C20000082204000020',,(____RF,OUTAREA*
), (FB_2,=Y(13))
*****
DS      0H                                01-DFHEC
LA      1,DFHEIPL                          01-DFHEC
LA      14,=X'180660000800C20000082204000020' 01-DFHEC
SR      15,15                               01-DFHEC
LA      0,OUTAREA                          01-DFHEC
STM     14,0,0(1)                          01-DFHEC
LA      14,=Y(13)                          01-DFHEC
ST      14,12(,1)                          01-DFHEC
OI      12(1),X'80'                          LAST ARGUMENT 01-DFHEC
L       15,=V(DFHEI1)                       01-DFHEC
BALR    14,15                               INVOKE EXEC INTERFACE 01-DFHEC
*****
*       EXEC CICS RETURN
DFHECALL =X'0E0800000800001000'
*****
DS      0H                                01-DFHEC
LA      1,DFHEIPL                          01-DFHEC
LA      14,=X'0E0800000800001000'          01-DFHEC
ST      14,0(,1)                            01-DFHEC
OI      0(1),X'80'                          LAST ARGUMENT 01-DFHEC
L       15,=V(DFHEI1)                       01-DFHEC
BALR    14,15                               INVOKE EXEC INTERFACE 01-DFHEC
*****
*
MSG1    DC    C'xxxx: ASM program invoked. ENTER TO END.'
MSG2    DC    C'PROGRAM ENDED'
DFHEIRET          INSERTED BY TRANSLATOR
*****
*       EPILOG CODE FOR EXEC INTERFACE      *
*****
DS      0H                                @BBAC81A 01-DFHEI
LA      1,CEET0014                          Get address of termination list 02-CEETE
L       15,=V(CEETREC)                      Get address of termination rtn 02-CEETE
BALR    14,15                               Call termination routine.      02-CEETE

CEET0014 DC    A(*+8)                        Parm 1                        02-CEETE
DC    A(*+8+X'80000000')                    Parm 2                        02-CEETE
DC    A(0)                                    Enc_Modifier                  02-CEETE
DC    A(0)                                    Return code.                  02-CEETE

CEEMAIN CSECT                                02-CEETE
CEEMAIN RMODE ANY                            02-CEETE
CEEMAIN AMODE ANY                            02-CEETE
DC    A(TESTLE)                              @04A 02-CEETE
DC    F'0'                                    02-CEETE

TESTLE   CSECT                                02-CEETE
*****
*       END OF EPILOG CODE FOR EXEC INTERFACE      *
*****

```

```

LTORG , @BBAC81A 01-DFHEI
    =V(DFHEI1)
    =V(CEETREC)
    =Y(43)
    =Y(13)
    =X'180660000800C20000082204000020'
    =X'040200000800000014000040000000'
    =X'0E0800000800001000'
DS    0H @F8E1S @LIC 01-DFHEI
DFHEISTG          INSERTED BY TRANSLATOR
DFHEIEND          INSERTED BY TRANSLATOR
*
*****
*   P R O G R A M   P R O L O G   A R E A   1   (   P P A   1   )   *
*****
*
PPA10018 DS    0F 02-CEEPP
DFHPPA DS    0F 02-CEEPP
DC AL1(PPANL0018-*) Offset to the entry name length 02-CEEPP
DC X'CE' LE/370 Indicator. 02-CEEPP
DC B'10100000' . PPA flags 02-CEEPP
*
*           Bit 0 0 = Internal Procedure
*                1 = External Procedure
*
*           Bit 1 0 = Primary Entry Point
*                1 = Secondary Entry Point
*
*           Bit 2 0 = Block doesn't have a DSA
*                1 = Block has a DSA
*
*           Bit 3 0 = compiled object
*                1 = library object
*
*           Bit 4 0 = sampling interrupts to library
*                1 = sampling interrupts to code
*
*           Bit 5 0 = not an exit DSA
*                1 = Exit DSA
*
*           Bit 6 0 = own exception model
*                1 = inherited (callers) exception model
*
*           Bit 7 Reserved
DC X'00' Member flags 02-CEEPP
DC A(PPA20018) Addr of Compile Unit Block (PPA2) 02-CEEPP
DC A(0) 02-CEEPP
DC A(0) Data Descriptors for this entry point 02-CEEPP
DS 0H 02-CEEPP
PPANL0018 DC AL2(6) . Length of Entry Point Name 02-CEEPP
DC CL6'TESTLE' . Entry Point Name 02-CEEPP
CEEINPL DS 0D 02-CEEPP
DC A(PPA2M0018) 02-CEEPP
DC A(CEEINPLSTST-CEEINPL) 02-CEEPP
CEEINPLSTST DS 0F 02-CEEPP
DC X'00' Control Level @01A 02-CEEPP
DC X'00' ENCLAVE=NO @01A 02-CEEPP
DC X'00' @01A 02-CEEPP
DC X'07' Number of items. @01C 02-CEEPP
DC A(PPA2M0018) . A of A(first entry point in comp unit) 02-CEEPP
DC V(CEESTART) . A(Address of CEESTART) 02-CEEPP
DC V(CEEBETBL) 02-CEEPP
DC A(15) . Memeber id 02-CEEPP
DC A(0) 02-CEEPP
DC XL4'00070000' . EXECOPS(ON), PLIST 02-CEEPP
DS 0H 02-CEEPP
*

```

```

*****
*   PROGRAM PROLOG AREA 2 ( PPA2 )   *
*****
*
EXTRN CEESTART                                02-CEEPP
PPA20018 DS 0F                                02-CEEPP
          DC AL1(15) Member ID                02-CEEPP
          DC AL1(0) Sub ID                    02-CEEPP
          DC AL1(0) Member defined            02-CEEPP
          DC AL1(1) Level of PPAX control blocks 02-CEEPP
PPA2S0018 DC A(CEESTART) A(CEESTART for this load module) 02-CEEPP
          DC A(0) A(Compile Debug Information (CDI) ) 02-CEEPP
          DC A(CEETIMES-PPA20018) A(Offset to time stamp) 02-CEEPP
PPA2M0018 DC A(TESTLE) . A(first entry point in comp. unit) 02-CEEPP
*
*****
*   TIME STAMP   *
*****
*
Time Stamp
*,Time Stamp = 2004/06/17 08:51:00 02-CEEPP
*,Version 1 Release 1 Modification 0 02-CEEPP
CEETIMES DS 0F                                02-CEEPP
          DC CL4'2004' Year                    02-CEEPP
          DC CL2'06' Month                    02-CEEPP
          DC CL2'17' Day                      02-CEEPP
          DC CL2'08' Hours                    02-CEEPP
          DC CL2'51' Minutes                  02-CEEPP
          DC CL2'00' Seconds                  02-CEEPP
          DC CL2'1' Version                   02-CEEPP
          DC CL2'1' Release                   02-CEEPP
          DC CL2'0' Modification              02-CEEPP
*****
* COMMON ANCHOR AREA ( CAA ) *
*****
LEPTRLEN EQU 4                                03-CEEDN
*
CEECAA DSECT , CAA mapping                    02-CEECA

(Definition of LE CAA removed)

* TERMINATE DEFINITION OF DYNAMIC STORAGE *
DFHEISTG DSECT @BBAC81A 01-DFHEI
          ORG 01-DFHEI
DFHEIEND DS 0X END OF DYNAMIC STORAGE @BBAC81A 01-DFHEI
          END

```

グローバル・ユーザー出力ルーチンの変更点

LEASM オプションを使用して変換されたアセンブラー・プログラムは、グローバル・ユーザー出力プログラムとしては使用できません。

LEASM は、アセンブラーで言語環境プログラムに準拠した MAIN プログラムを作成するために使用します。

タスク関連ユーザー出口の変更点

LEASM オプションを使用して変換されたアセンブラー・プログラムは、タスク関連ユーザー出口プログラムとしては使用できません。

LEASM は、アセンブラーで言語環境プログラムに準拠した MAIN プログラムを作成するために使用します。

第 4 部 エンタープライズ・マネージメント

CICS Transaction Server for z/OS バージョン 3 リリース 1 の CICSplex SM エレメントでは、最新のユーザー・インターフェースを採用することにより、大規模なランタイム構成の効果的な管理を実現する新機能を提供します。これにより、要求の多いサービス・レベルの目的に対応することができます。

第 11 章 CICSplex SM Web ユーザー・インターフェースの拡張

CICSplex SM Web ユーザー・インターフェースが改良されて、より強力かつ便利になりました。Web ユーザー・インターフェースは、CICSplex SM TSO エンド・ユーザー・インターフェースと機能的に同等となり、CICSplex SM にアクセスするための基本方式となりました。

ユーザーのお気に入り

すべての WUI ユーザーは、アクセスが容易で編集可能なお気に入りのメニューに、その場かぎりの表形式のビューおよび詳細ビューを保管できるようになりました。これにより、頻繁に使用されるビューに 1 回のクリック操作でアクセスできるようになります。管理者には、他のユーザーのお気に入りを更新できるという付加的な権限があります。

詳細については、301 ページの『ユーザーのお気に入り』を参照してください。

ユーザー・グループ・プロファイル

管理者は、ユーザーのグループのプロファイルを作成できるようになりました。これらのプロファイルには、デフォルトのコンテキスト、スコープ、CMAS コンテキスト、メニューおよび結果セット警告カウントなどの情報が含まれます。これにより、管理者は、個々のニーズに合わせて調整されたインターフェースを表示するために、それぞれのユーザー・グループに適した方法で WUI を構成することができます。

詳細については、306 ページの『ユーザー・グループ・プロファイル』を参照してください。

ビジネス・アプリケーション・サービスの設計変更

CICS リソース定義を管理するために使用される、WUI のビジネス・アプリケーション・サービス部分の設計に改良が加えられ、単純化されたため、CICS RDO のユーザーにより分かりやすくなりました。

詳細については、310 ページの『ビジネス・アプリケーション・サービスの設計変更』を参照してください。

レコード・カウントの警告

WUI は、大量のレコードを生成するビューをオープンする前に警告を発行するよう調整できるようになりました。警告の後で、戻される結果の数を減らすためにビューのフィルターを変更する機会がユーザーに与えられます。これにより、無駄な待機が減り、WUI のパフォーマンスが改善されます。

詳細については、313 ページの『レコード・カウントの警告』を参照してください。

フィルター確認

WUI ビュー・エディターが改善され、ビューの作成または更新時に、ビューがオープンする前にフィルター確認パネルを組み込むことができるようになりました。これは、このようなビューにナビゲートする際に、戻される結果セットのサイズに関係なく、ユーザーがフィルターの内容を修正できることを意味します。

詳細については、315 ページの『フィルター確認』を参照してください。

動的選択リスト

WUI は、入力パネルのさまざまな属性において、指定可能な値をリストするようになりました。これにより、有効な値を覚えておくのではなく、有効な値のリストから選択できるようになり、インターフェースの使用可能度が高まります。

詳細については、316 ページの『動的選択リスト』を参照してください。

画面設計の改良

WUI のビューおよびメニューで画面スペースの使用を最大化するために、いくつかの改善が行われました。

- 詳細ビューでの画面スペースの使用が改善され、1 つのカラムではなく 2 つのカラムに情報を表示できるようになりました。ビュー・エディターを使用して、2 つのカラム詳細ビューをユーザーが設計することができます。
- 「すべて選択」ボタンと「選択をすべて解除」ボタンを除去して代わりに表のレコード・ヘディングにアイコンを配置することにより、表形式のビューでの空白の量が減りました。
- より多くの画面スペースをデータ用に提供するために、フィルターを縮小できるようになりました。縮小されたフィルターは、必要に応じて展開できます。

詳細については、318 ページの『画面設計の改良』を参照してください。

CICSplex SM Web ユーザー・インターフェースの拡張の利点

CICS Transaction Server バージョン 3 では、CICSplex SM Web ユーザー・インターフェースに対して、ユーザーに実際の利点を提供するいくつかの改善が導入されています。

新機能は、以下のとおりです。

301 ページの『ユーザーのお気に入り』

すべての WUI ユーザーは、アクセスが容易で編集可能なお気に入りのメニューに、その場かぎりの表形式のビューおよび詳細ビューを保管できるようになりました。これにより、頻繁に使用されるビューに 1 回のクリック操作でアクセスできるようになります。管理者には、他のユーザーのお気に入りを更新できるという付加的な権限があります。

306 ページの『ユーザー・グループ・プロファイル』

管理者は、ユーザーのグループのプロファイルを作成できるようになりました。これらのプロファイルには、デフォルトのコンテキスト、スコープ、CMAS コンテキスト、メニューおよび結果セット警告カウントなどの情報が含まれます。これにより、管理者は、個々のニーズに合わせて調整されたインターフェースを表示するために、それぞれのユーザー・グループに適した方法で WUI を構成することができます。

313 ページの『レコード・カウントの警告』

WUI は、大量のレコードを生成するビューをオープンする前に警告を発行するよう調整できるようになりました。警告の後で、戻される結果の数を減

らすためにビューのフィルターを変更する機会がユーザーに与えられます。これにより、無駄な待機が減り、WUI のパフォーマンスが改善されます。

315 ページの『フィルター確認』

WUI ビュー・エディターが改善され、ビューの作成または更新時に、ビューがオープンする前にフィルター確認パネルを組み込むことができるようになりました。これは、このようなビューにナビゲートする際に、戻される結果セットのサイズに関係なく、ユーザーがフィルターの内容を修正できることを意味します。

316 ページの『動的選択リスト』

WUI は、入力パネルのさまざまな属性において、指定可能な値をリストするようになりました。これにより、有効な値を覚えておくのではなく、有効な値のリストから選択できるようになり、インターフェースの使用可能度が高まります。

310 ページの『ビジネス・アプリケーション・サービスの設計変更』

CICS リソース定義を管理するために使用される、WUI のビジネス・アプリケーション・サービス部分の設計に改良が加えられ、単純化されたため、CICS RDO のユーザーにより分かりやすくなりました。

318 ページの『画面設計の改良』

WUI のビューおよびメニューで画面スペースの使用を最大化するために、いくつかの改善が行われました。

- 詳細ビューでの画面スペースの使用が改善され、1 つのカラムではなく 2 つのカラムに情報を表示できるようになりました。ビュー・エディターを使用して、2 つのカラム詳細ビューをユーザーが設計することができます。
- 「すべて選択」ボタンと「選択をすべて解除」ボタンを除去して代わりに表のレコード・ヘディングにアイコンを配置することにより、表形式のビューでの空白の量が減りました。
- より多くの画面スペースをデータ用に提供するために、フィルターを縮小できるようになりました。縮小されたフィルターは、必要に応じて展開できます。

要件

この機能をサポートするための特別なハードウェア要件およびソフトウェア要件はありません。

関連情報

377 ページの『第 27 章 CICS 稼働環境』

CICSplex SM の変更点

ユーザーのお気に入り

このトピックでは、CICSplex SM Web ユーザー・インターフェースの新しいユーザーのお気に入り機能について説明します。

ユーザーのお気に入りを使用すると、WUI の個々のユーザーは、選択済み WUI 画面を、迅速かつ容易にアクセスできる特別なメニューに保管できます。お気に入りのセットは、常に個々のユーザーにのみ関連付けられ、ユーザー・グループに関連付けられることはありません。ただし、新しいユーザー・グループ・プロファイル機能をビュー・エディターと組み合わせて使用することで、同様の結果を得ることができます (手順については、306 ページの『ユーザー・グループ・プロファイル』を参照)。

ユーザーのお気に入りに関する情報は、「ユーザー」と呼ばれる新しいオブジェクトに保管されます。ユーザー・オブジェクトは、ユーザーのお気に入りへのハイパーリンクを含むメニュー・グループを保持しています。ユーザー・オブジェクトは、次の 2 通りの方法で作成されます。

- ユーザー・オブジェクトは、ユーザーが初めてお気に入りを作成するときを作成できます。ユーザーが  アイコンをクリックしてお気に入りを作成すると、WUI では、そのユーザーの ID に一致する ID を持つユーザー・オブジェクトを検索します。一致するユーザー・オブジェクトが見つからない場合、WUI サーバーはそのユーザーに対応するユーザー・オブジェクトを自動的に作成します。
- ユーザー・オブジェクトは、管理者がユーザー・エディターで作成できます。詳細については、304 ページの『他の WUI ユーザーのお気に入りの作成と管理』を参照してください。

お気に入りのエディターによるお気に入りの管理

このトピックでは、ユーザーが、新しい Web ユーザー・インターフェースであるお気に入りのエディターを使用して、ユーザーのお気に入りを管理する方法について説明します。

 アイコンを使用してユーザーのお気に入りリストにビューを追加できるだけでなく、お気に入りのエディターを使用して、お気に入りを操作することができます。このエディターをオープンするには、WUI ナビゲーション・フレームの「特別」セクションの「お気に入りのエディター」をクリックします。これにより、303 ページの図 30 に示すように、新しいブラウザー・ウィンドウでエディターがオープンします。

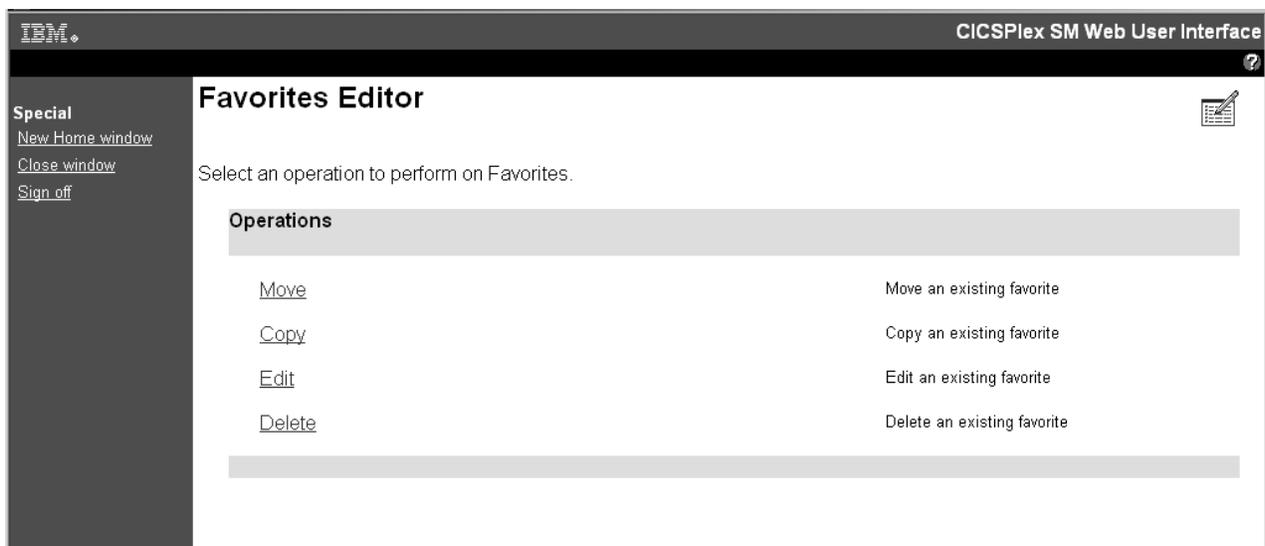


図 30. お気に入りのエディター

お気に入りを編集するには、ラジオ・ボタンを選択し、ビューの下部にある 4 つのボタンのいずれかをクリックします。

1. 画面を編集するには、「**編集**」をクリックします。リストから編集したいお気に入りを選択した後、「**了解**」をクリックして、「**お気に入りのコンポーネント**」画面を開きます。この画面から、メニューのタイトルと注釈、または宛先の変更を選択することができます。コンテキストやスコープの設定およびフィルターの設定を更新する場合は、「**宛先**」を選択します。
2. お気に入りリストの項目の位置を移動するには、「**移動**」をクリックします。リストからお気に入りのうちの 1 つを選択して、「**了解**」をクリックします。これにより、「**お気に入りの移動**」画面がオープンします。選択した項目用の新しい位置を選択して、再度「**了解**」をクリックし、変更を行います。
3. お気に入り画面をコピーするには、「**コピー**」をクリックします。リストからお気に入りのうちの 1 つを選択して、「**了解**」をクリックします。これにより、304 ページの図 31 に示すような「**お気に入りのコピー**」画面がオープンします。

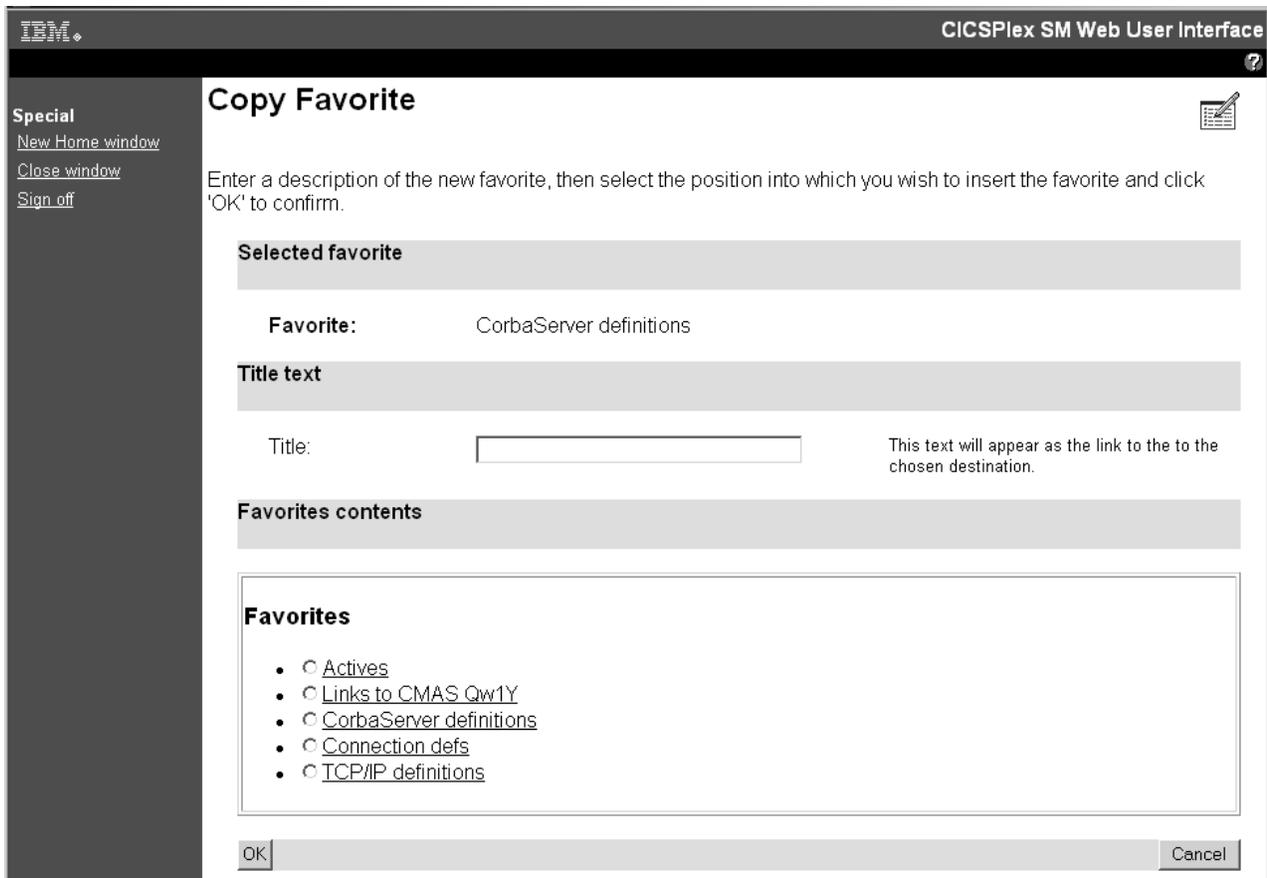


図 31. 「お気に入りのコピー」画面

これで、コピーに新しいタイトルを付けることができます。お気に入りのリストから、新しいコピーを配置する既存のお気に入りを選択することにより、お気に入りリスト上の位置を選択します。新しいコピーをリストの末尾に配置する場合は、既存のお気に入りを選択しないでください。操作を確認するために、「了解」をクリックします。

4. お気に入りを削除するには、「削除」をクリックします。リストからお気に入りのうちの 1 つを選択して、確認のため、「了解」をクリックします。

お気に入りのエディターを使用すると、ユーザーは自分のお気に入りのリストを編集できます。他のユーザーのお気に入りを管理することはできません。この権限を持っているユーザー・エディターにアクセスできるのは、WUI 管理者だけです。

他の WUI ユーザーのお気に入りの作成と管理

このトピックでは、新しい Web ユーザー・インターフェースであるユーザー・エディターを使用して、管理者が他のユーザーのお気に入りを管理する方法について説明します。

他のユーザーのお気に入りを作成および管理するユーザーは、新しい ESM 機能プロファイル EYUWUI.wui_server_applid.USER への少なくとも更新アクセス権を持つ WUI 管理者でなければなりません。

図 32 は、ユーザーのお気に入り編集するために使用されるユーザー・エディターの画面を示しています。

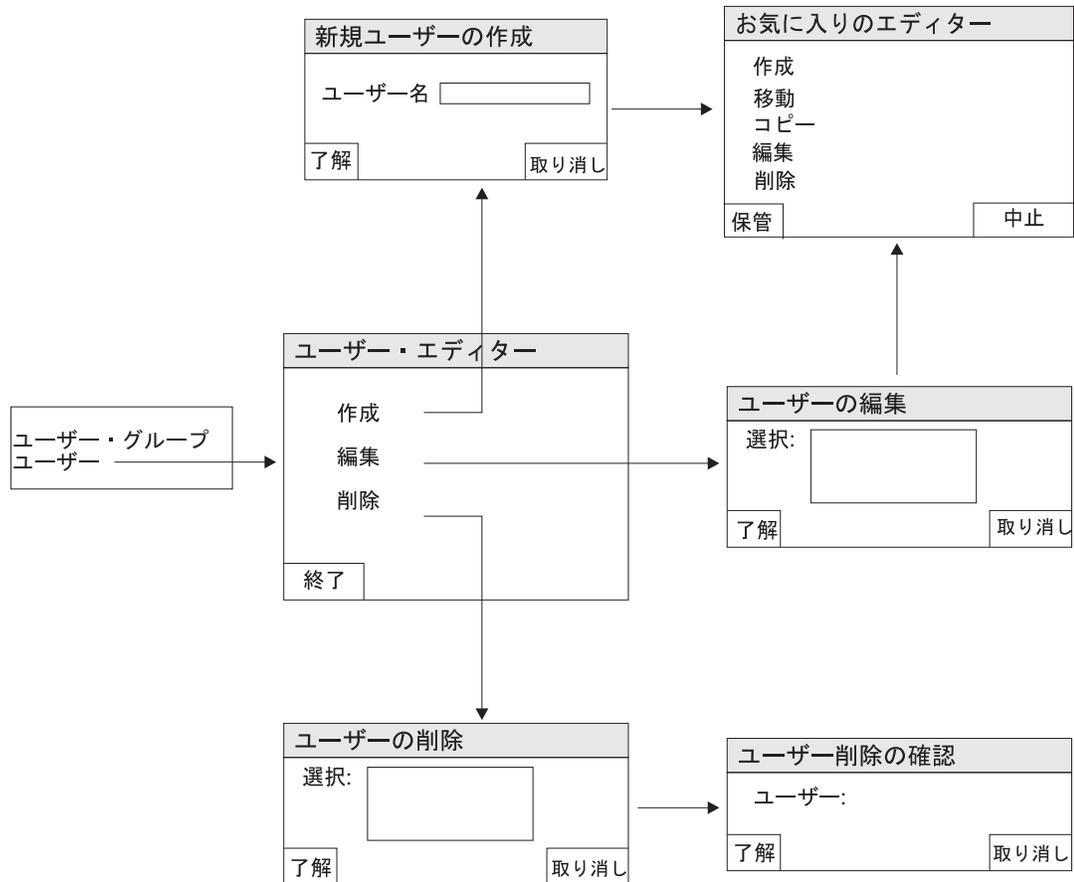


図 32. ユーザー・エディターの「ユーザー」画面

ユーザーが必要なアクセス権を持っている場合は、ユーザー・エディターを起動するハイパーリンクが、ビュー・エディターのハイパーリンクと共にナビゲーション・フレームに表示されます。このハイパーリンクをクリックすると、「**CICSPlex SM Web ユーザー・インターフェース・ユーザー・エディター**」画面を表示した新しいブラウザ・ウィンドウが起動されます。お気に入りを操作するには、「ユーザー」を選択します。これにより、「ユーザー・エディター」画面がオープンします。この画面には、以下のオプションがあります。

作成 このオプションは、リポジトリに新規ユーザー・オブジェクトを作成するために使用します。各ユーザーは、お気に入りのリストを保持するためのユーザー・オブジェクトを持っていない限りなりません。WUI は、ユーザーが初めてお気に入りを作成したときに自動的にユーザー・オブジェクトを作成するので (ユーザー・オブジェクトが存在しない場合)、多くの既存のユーザーは既にユーザー・オブジェクトを持っていることになります。

編集 このオプションは、既存のユーザーのお気に入り进行操作するために使用します。これを使用すると、ユーザーのお気に入りを作成、移動、コピー、編集、および削除することができます。

削除 このオプションは、リポジトリからユーザー・オブジェクトを削除するために使用します。これにより、そのユーザーの既存のお気に入りも削除されます。

ユーザー・エディターを使用して、新規ユーザー・オブジェクトを作成し、いくつかのお気に入りを追加する方法を以下に示します。

1. 新規ユーザー・オブジェクトを作成します。
 - a. 最初に、「ユーザー・エディター」画面で、「作成」をクリックして、「新規ユーザーの作成」画面をオープンします。
 - b. 新規ユーザー ID を入力します。ユーザー ID は、最大 8 文字に制限されます。有効な文字は、A から Z、0 から 9、#、\$、および @ です。
 - c. 「OK」をクリックします。これにより、操作が確認され、ユーザーのお気に入りのリストの作成と管理に使用する「お気に入りのエディター」画面がオープンします。
2. 新しいお気に入り項目を作成します。
 - a. 「お気に入りのエディター」画面で、「作成」をクリックします。これにより、ビュー・エディターで使用する編集画面がオープンし、詳細項目を編集できるようになります。この画面から、ビュー・エディターのメニュー項目で実行できる大部分のアクションを、お気に入りに対して実行できます。
 - b. 新しいお気に入りのタイトルを指定し、ターゲット・ビューまたはメニュー、コンテキストおよびスコープ設定、およびフィルター設定を入力することにより、この画面を完成します。
 - c. 「保管」をクリックします。これにより、更新内容がリポジトリに保管され、「ユーザー・エディター」画面に戻ります。「中止」をクリックすると、新規ユーザー・オブジェクトを含むすべての変更が廃棄されます。
3. 追加のお気に入りを作成して、新規ユーザー・オブジェクトに追加します。
 - a. 「ユーザー・エディター」画面で、「編集」をクリックします。これにより、「ユーザーの編集」画面がオープンします。
 - b. 新たに作成したユーザー ID を選択し、「了解」をクリックします。これにより、「お気に入りのエディター」画面に戻ります。
 - c. 前と同じようにして新たなお気に入り項目を作成し、更新内容を保管します。必要なそれぞれのお気に入りごとにこのステップを繰り返し、新規ユーザー・オブジェクトに追加します。

関連情報

302 ページの『お気に入りのエディターによるお気に入りの管理』

このトピックでは、ユーザーが、新しい Web ユーザー・インターフェースであるお気に入りのエディターを使用して、ユーザーのお気に入りを管理する方法について説明します。

ユーザー・グループ・プロフィール

このトピックでは、CICSplex SM Web ユーザー・インターフェースの新しいユーザー・グループ・プロフィール機能について説明します。

この新機能を使用すると、管理者は WUI ユーザーのグループのプロファイルを作成することができます。個々のユーザーが WUI にログオンしたときに表示される画面およびこの画面で実行する操作は、ユーザーが所属するユーザー・グループ・プロファイルによって制御できます。これにより、さまざまなユーザー・グループのニーズに応じて WUI を調整することができます。例えば、ビュー・エディターを使用して、操作ビューのみを含む新たな WUI メニューを作成し、これをあるユーザー・グループのデフォルト・メニューとして使用できます。これにより、この種のタスクのみを実行する必要があるユーザーに対して、単純化された操作 WUI が提供されます。

ユーザー・グループ・プロファイルには、以下の情報が含まれています。

- 結果セット警告カウント (313 ページの『レコード・カウントの警告』を参照)
- デフォルトのメインメニューの名前
- デフォルトのナビゲーション・メニューの名前
- デフォルトのコンテキスト
- デフォルトのスコープ
- デフォルトの CMAS コンテキスト

これらの値の一部がプロファイルに指定されていない場合 (例えば、デフォルト・メニュー値がブランクの場合)、JCL に指定されている、対応する WUI パラメーター値が使用されます。このことは、指定されたデフォルト・メニューが存在しないなど、プロファイルに無効な値が設定されている場合にも適用されます。

ユーザー・グループ・プロファイルは、新しい編集機能であるユーザー・エディターを使用して作成および管理されます。ユーザー・エディターには、必要な権限を持つ管理者が、WUI メインメニューからアクセスできます。詳しくは、308 ページの『ユーザー・グループ・プロファイルの作成と管理』を参照してください。

これを容易にするために、ユーザー・グループと呼ばれる新しいタイプのオブジェクトが導入されました。ユーザー・グループ・オブジェクトは、ビューおよびメニュー・オブジェクトと同様、COVC トランザクションを使用してインポートおよびエクスポートすることができます。

ユーザー・グループ・プロファイルは、WUI でセキュリティーが有効な場合にのみ使用できます。ユーザー・グループ・オブジェクトは、外部セキュリティー・マネージャー (ESM) のユーザー・グループ名に関連付けられています。ユーザーが WUI にサインオンすると、サインオン・プロシージャによって、そのユーザーの ESM からアクセス機構環境エレメント (ACEE) が取得されます。ACEE は、ユーザーのデフォルト・グループを取得するために使用されます。次にデフォルト・グループ名を使用して、WUI 内の対応するグループ・オブジェクトが取得されます。ユーザーがサインオンしたときに WUI キャッシュに対応するグループ・オブジェクトが存在しない場合は、JCL の WUI パラメーターに指定されているデフォルト値がそのユーザーのために使用されます。

注: ESM ユーザー・グループは、以下の文字のみを含むことができます。

- A から Z (小文字を入力できますが、小文字は大文字に変換されます)
- 0 から 9
- # (X'23'), \$ (X'24') および @ (X'40')

ユーザー・グループ・プロファイルを管理するために、管理者は、EYUWUI.wui_server_applid.USER と呼ばれる新しい ESM 機能プロファイルへの更新アクセス権を持っている必要があります。ここで、wui_server_applid は、プロファイルが関連付けられている WUI サーバーのアプリケーション ID です。

新規ユーザー・グループのセットアップは、以下のステップで構成されます。

1. ユーザー・エディターを使用して WUI で新しいユーザー・グループ・プロファイルを作成する (手順については、『ユーザー・グループ・プロファイルの作成と管理』を参照)。
2. 同じ名前を使用して ESM でユーザー・グループを作成する。
3. 1 人以上のユーザーのデフォルトのユーザー・グループを ESM で設定する。

ユーザー・グループ・プロファイルの作成と管理

このトピックでは、Web ユーザー・インターフェース (WUI) のユーザー・グループ・プロファイルの作成および管理方法について説明します。

ユーザー・グループ・プロファイルは、ユーザー・エディターという新しい WUI 編集機能を使用して作成および管理します。

ユーザー・エディターを起動するハイパーリンクが、ビュー・エディターのハイパーリンクと共にナビゲーション・フレームに表示されます。サインオンしたユーザーが、新しい ESM 機能プロファイル EYUWUI.wui_server_applid.USER への少なくとも更新アクセス権を持っている場合にのみ、ユーザー・エディターへのハイパーリンクがそのユーザーに表示されます。このハイパーリンクをクリックすると、新たなブラウザ・ウィンドウが起動されます。このブラウザ・ウィンドウでは、ユーザー・エディター用に作成された新しいビューのセットを使用しています。

ユーザー・エディターを起動するときに、ユーザー・グループを編集するか、ユーザーを編集するかを選択できます。309 ページの図 33 のダイアグラムは、ユーザー・エディターのユーザー・グループ・セクションにおける画面間の関係を示しています。

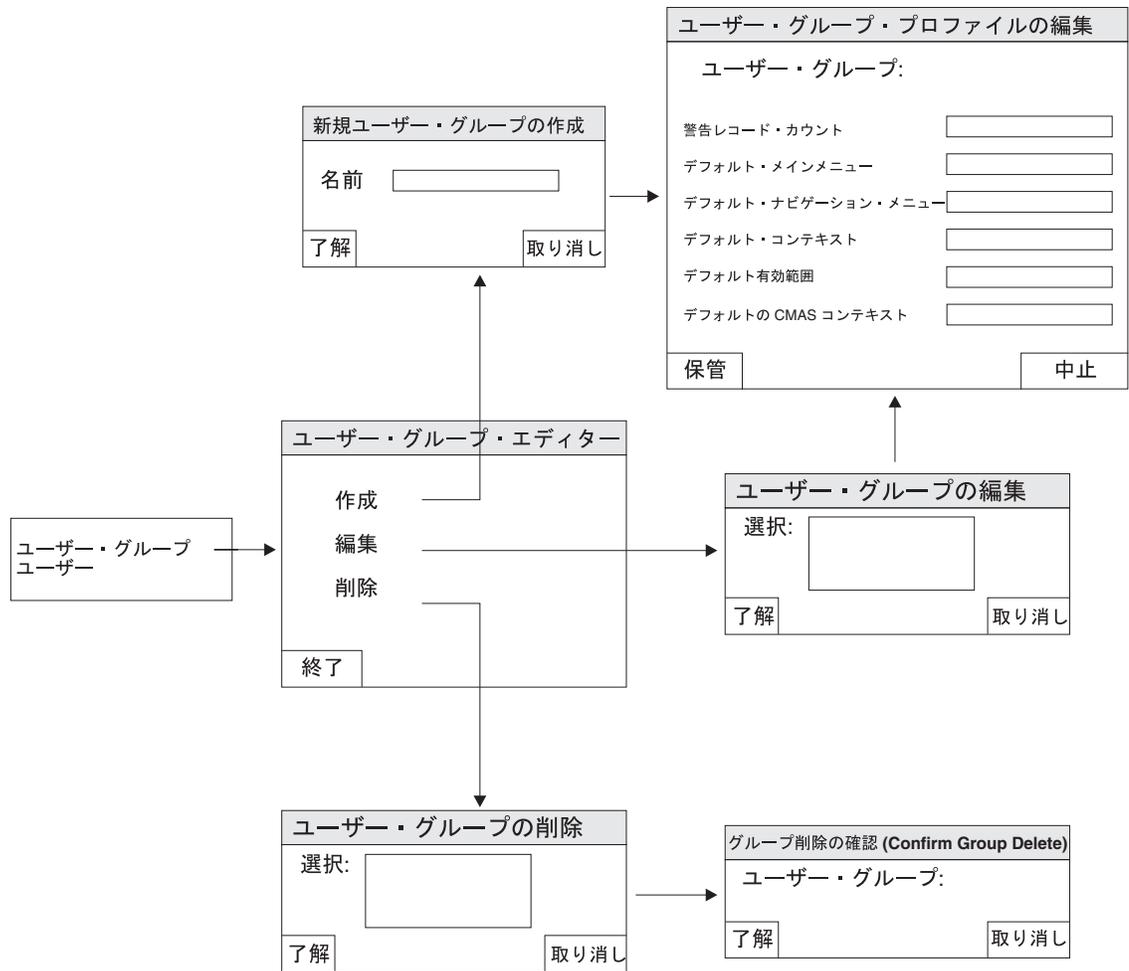


図 33. ユーザー・エディターのユーザー・グループ画面

ユーザー・エディターの使用は簡単です。例えば、新しいプロファイルを作成するには、以下の手順を実行します。

1. 以下の手順で、「新規ユーザー・グループの作成」画面にナビゲートします。
 - a. ナビゲーション・フレームで「ユーザー・エディター」をクリックして、新しい「CICSPlex SM Web ユーザー・インターフェース・ユーザー・エディター」ウィンドウをオープンします。
 - b. 「ユーザー・グループ」をクリックして、「ユーザー・グループ・エディター」画面をオープンします。この画面では、プロファイルの作成、編集、または削除を行うことを選択できます。
 - c. 「作成」をクリックして、「新規ユーザー・グループの作成」画面をオープンします。
2. 新しいグループ・プロファイルに名前を付けます。
 - a. 新しいグループの名前を指定します。(この名前は、ESM に定義されているユーザー・グループ名と一致している必要があります。) 名前は、最大 8 文字に制限されます。ESM と同様、許容文字は A から Z (大文字)、0 から 9、#、\$ および @ です。

- b. 「OK」をクリックします。これにより、操作が確認され、「ユーザー・グループ・プロファイルの編集」画面がオープンします。
3. ユーザー・グループ詳細を入力します。この情報の詳しい説明については、306ページの『ユーザー・グループ・プロファイル』を参照してください。
4. 「保管」をクリックして新しいプロファイルを作成するか、「中止」をクリックして操作を取り消します。

グループを編集するには、「ユーザー・グループ・エディター」画面で「編集」をクリックして、「ユーザー・グループの編集」画面をオープンし、既存のグループを選択します。これにより、選択したグループの詳細を表示した「ユーザー・グループ・プロファイルの編集」画面がオープンします。変更を行い、「保管」をクリックしてプロファイルを更新するか、「中止」をクリックして操作を取り消します。

グループを削除するには、「ユーザー・グループ・エディター」画面で「削除」をクリックして「ユーザー・グループの削除」画面をオープンし、既存のグループを選択します。「了解」をクリックして「ユーザー・グループ削除の確認」画面をオープンします。「了解」を再度クリックして、操作を確認します。

ビジネス・アプリケーション・サービスの設計変更

このトピックでは、WUI で CICS リソース定義に使用されるビジネス・アプリケーション・サービスのビューおよびメニューの設計に対する改善点について説明します。

WUI の新しい設計では、BAS 機能が基本ビュー・メニューと完全機能ビュー・メニューに分離されました。WUI のメインメニューから BAS 機能にアクセスするには、「管理ビュー」をクリックします。管理ビューの下部に、次の 2 つの新しいサブメニューがあります。

ベーシック CICS リソース管理ビュー

このサブメニューでは、リソース定義、リソース・グループ、およびリソースの記述を含む (リソース割り当ては含まない) BAS の RDO に類似した簡易化モデルを提供します。

完全に機能しているビジネス・アプリケーション・サービス (BAS) 管理ビュー

このサブメニューには、基本モデルに加えて、上級者向けのリソース割り当てビューへのリンクも含まれています。これにより、リソース定義をより強力で柔軟に管理できるようになります。

CICS resource definitions using Business Application Services (BAS)

- [Basic CICS resource administration views](#)
- [Fully functional Business Application Services \(BAS\) administration views](#)

The screenshot displays two side-by-side panels. The left panel, titled 'Basic CICS resource administration views', has a menu name of 'EYUSTARTADMIN'. It features input fields for 'CMAS context: JJCMAS1', 'Context: JJCFVPLX', and 'Scope: JJCFVPLX'. Below these are sections for 'Definitions', 'Associations', and 'Resources deployed by ...'. The right panel, titled 'Fully functional Business Application Services (BAS) administration views', has a menu name of 'EYUSTARTADMBAS2'. It includes the same input fields and a 'Set' button. Its 'Definitions' section includes 'Resource assignments' and 'Resource descriptions', while its 'Associations' section includes 'Resource assignments in description' and 'CICS system links'. Both panels list 'CICS resource definitions', 'Resource groups', and 'Resource descriptions' under their respective sections.

図 34. 新しく追加された CICS リソース定義 BAS メニューの詳細

2 つのメニューからのリンクは、次の 3 つのグループに分かれています。

定義 次のリンクがあります。

CICS リソース定義

リソース・タイプごとの定義ビューへのリンクが含まれているメニュー。

リソース・グループ

リソース・グループ定義を管理するための定義ビューへのリンク。「作成」、「更新」、「除去」、「インストール」、「リソース記述への追加」の各アクションが関連付けられています。

リソース割り当て (完全機能メニューのみ) (Resource assignments (fully functional menu only))

リソース割り当てを作成し、管理するための定義ビューへのリンク。「作成」、「更新」、「除去」、「リソース記述への追加」の各アクションが関連付けられています。

リソース記述

リソースの記述を作成し、管理するための定義ビューへのリンク。「作成」、「更新」、「除去」、「インストール」、「置換」の各アクションが関連付けられています。

関連 次のリンクがあります。

リソース・グループ内の CICS リソース定義

図 35 に示す新しいビューへのリンク。これは、新しく追加された RESINGRP リソース・テーブルによってサポートされます。前のリリースの「リソース・グループ内のリソース (Resources in resource group)」ビュー・セット (リソース・タイプごとのビュー) に代わって採用されました。

The screenshot displays the 'CICS resource definitions in resource group' web user interface. The interface includes a navigation menu on the left with options like 'Open', 'Alerts', 'Regions', 'Activity', 'Connectivity', 'Files & DB2', 'Journals', 'Queues', 'Transactions', 'Programs', 'Enterprise Java', and 'Administration'. The main content area shows a table of resource definitions for the resource group 'RESINGRP'. The table has 5 records, each with a checkbox, a record number, a resource name, a resource group name, a resource definition name, a resource definition version, a resource definition type, and a last time definition changed timestamp. A 'Remove...' button is located below the table. The interface also includes a search and filter section at the top with dropdown menus for 'Context', 'Resource group name', 'Resource definition name', 'Resource definition version', and 'Resource definition type', along with an 'Automatic refresh' checkbox and a 'Refresh' button.

Record	Resource group name	Resource definition name	Resource definition version	Resource definition type	Last time definition changed
1	JON1	FILE1		1 FILEDEF	2004/04/22 13:30:47
2	PAY	DIFFFILE		2 FILEDEF	2004/08/19 14:42:56
3	PAY	PAYROLL1		1 FILEDEF	2004/04/27 09:56:29
4	PERSONEL	PERSONEL		1 FILEDEF	2004/04/27 13:56:14
5	RESPLOP	PAYROLL1		1 FILEDEF	2004/04/26 11:53:44

図 35. リソース・グループ内の CICS リソース定義の表形式のビュー

このビューには、「除去」アクション・ボタンがあります。これを使用すると、リソース定義とその親のリソース・グループ間の関連を除去することができます。このビューでは、作成アクションはありません。リソースのグループへの追加は、リソース自体を定義するときに行われます。

記述内のリソース・グループ

リソース・グループとリソース記述の間の関連を管理するための定義ビューへのリンク。「作成」、「更新」、「除去」の各アクションが関連付けられています。

リソース記述内のリソース割り当て (完全機能メニューのみ) (Resource assignment in description (fully functional menu only))

リソース割り当てとリソース記述の間の関連を管理するための定義ビューへのリンク。「作成」、「更新」、「除去」の各アクションが関連付けられています。

CICS システム・リンク

CICS システム・リンク定義を管理するための定義ビューへのリンク。「作成」、「除去」、「インストール」の各アクションが関連付けられています。

配置リソースの選択元 (Resources deployed by...)

アクティブな CICS リソースを表示するビューへの以下のリンクが含まれています。

リソース記述

リソース記述によって選択された配置リソースを表示する表形式のビューへのリンク。

リソース割り当て (完全機能メニューのみ) (Resource assignment (fully functional menu only))

リソース割り当てによって選択された配置リソースを表示する表形式のビューへのリンク。

CICS システム

CICS システムによって選択された配置リソースを表示する表形式のビューへのリンク。

レコード・カウントの警告

このトピックでは、特定のビュー内に戻される予想レコード数が設定されたしきい値を超えた場合、WUI が警告を出す仕組みについて説明します。

この新機能は、一定のビューをオープンする要求で、結果として予想されたデータ量より大量のデータが戻される場合に、警告を出す仕組みとなっています。これにより、これらのビューのフィルターを変更したり、要求を実行する前に要求を確認または取り消す機会が与えられます。

この機能は、以下のリソースに関連したビューに適用されます。

CMDT	FEPINODE	PROGRAM
CONNECT	FEPPOOL	REMFIL
DB2CONN	FEPITRGT	REMTDQ
DB2ENTRY	INDTDQ	REMTRAN
DB2TRN	INTRATDQ	RQMODEL
DOCTEMP	JOURNAL	SMFJRN
DSKJRN	JRNNAME	SYSDUMP
DSNAME	LOCFILE	TAPEJRN
ENQMODEL	LOCTRAN	TCPIPS
EXITGLUE	MODENAME	TERMN
EXITTRUE	PARTNER	TRAN
EXTRATDQ	PROCTYP	TRANDUMP
FEPICONN	PROFILE	TSMODEL

警告メカニズムを起動するために必要なレコード数を指定するには、次の 2 つの方法があります。

- WUI サーバーの構成時に新しく追加された DEFAULTWARNCNT WUI サーバー初期設定パラメーターに値を設定する。DEFAULTWARNCNT には、0 から 99999999 の範囲の整数値を指定することができます。これはオプション・パラメーターです。
- ユーザー・グループ・プロファイルで警告レコード・カウントの値を設定する。WUI 管理者は、これにより WUI ユーザー・グループのすべてのメンバーに値を設定することができます。詳細については、308 ページの『ユーザー・グループ・プロファイルの作成と管理』を参照してください。

どちらの場合もデフォルト値は 0 であり、これは警告が発行されないことを意味します。

ユーザー・グループに設定された値は、DEFAULTWARNCNT パラメーターに設定された値よりも優先されます。警告を発行する前に、WUI では、サインオンしたユーザーがユーザー・グループに関連付けられているかどうかを判別するために検査を行います (これはセキュリティーをオンに切り替えて WUI を実行している場合にのみ発生します)。関連付けられている場合、WUI ではそのユーザー・グループに指定されている警告カウントの値を使用します。ユーザーがユーザー・グループに属さない場合、あるいは最大値が設定されていない場合、WUI では、初期設定パラメーターを検査し、そこで設定されている値を使用します。値が指定されていない場合は、デフォルト値である 0 が使用されます。

警告カウントの値が 0 より大きい場合、WUI は戻される可能性のあるレコード・セットのサイズを調べます。これにより、戻される可能性のあるレコードの最大数を示すことができます。1 次キー・フィールドの最初の部分に関連付けられたフィルター以外のフィルターは考慮に入れられないため、必ずしも正確とは限りません。この潜在的な値が警告カウントの値よりも大きい場合、315 ページの図 36 に示すような警告画面が表示されます。戻り値が警告カウント値より小さいか等しい場合、通常の方法で WUI ビューが表示されます。

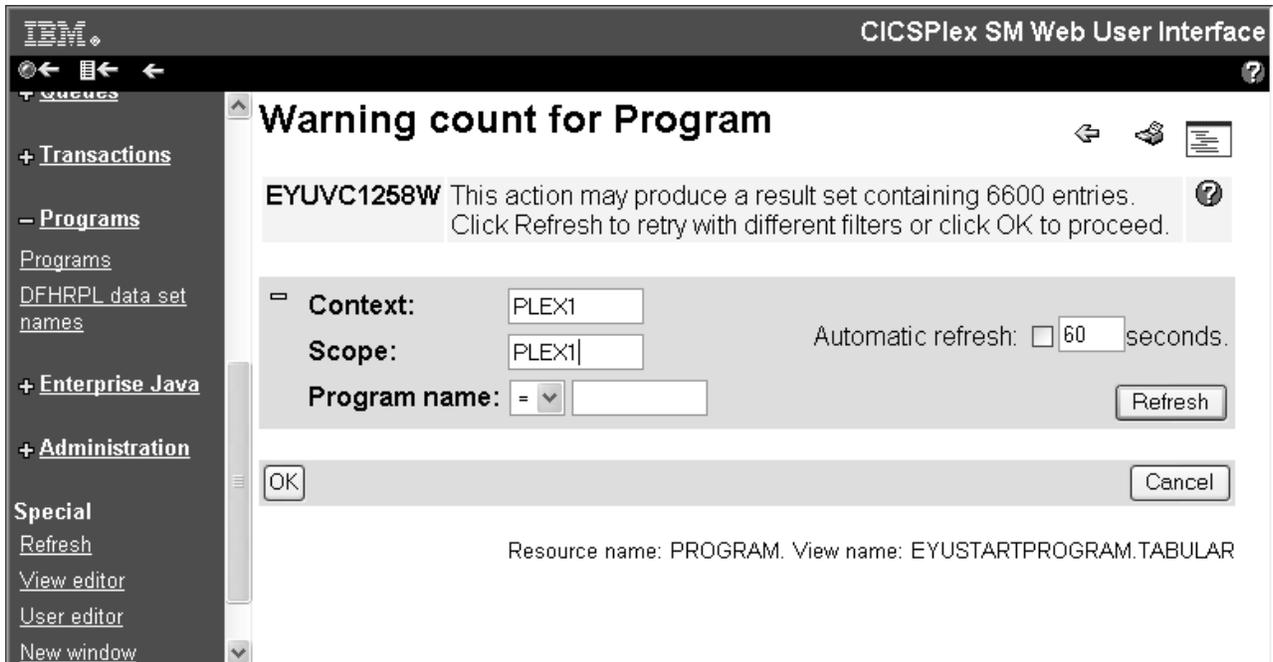


図 36. 警告カウント画面

「最新表示」をクリックすると、新たな要求が起動されます。これにより、1 次キー・フィールドを変更することができます。この 1 次キー・フィールドは、より小さい結果セットを生成する目的で新しいフィルター値をテストするための警告カウント画面内にあります。その他のフィルターはすべて、元の要求と同じままです。戻された結果セットが警告カウントの限度を超える場合は、警告カウント画面が元の警告カウント・メッセージと共に再表示されます。結果セットのサイズが警告カウントの限度以下である場合は、警告カウント画面が再表示されます。

画面下部にある「了解」をクリックすると、戻される結果の数に関係なく、現在の要求が実行されます。

「取り消し」をクリックすることは、ブラウザーで「戻る」リンクを押すことに相当します。つまり、前の画面が再表示されます。

フィルター確認

このトピックでは、CICSPlex SM Web ユーザー・インターフェースの新しいフィルター確認機能について説明します。

ユーザーは、TSO EUI と同様の方法で、WUI がビュー用のデータを取得する前にフィルター基準を指定できるようになりました。

ビュー・エディターで独自のビューを設計するときに、データ検索の実行前にフィルター確認画面が表示されるように指定できるようになりました。WUI で提供されるビューは、この変更による影響を受けません。標準的なフィルター確認画面を 316 ページの図 37 に示します。

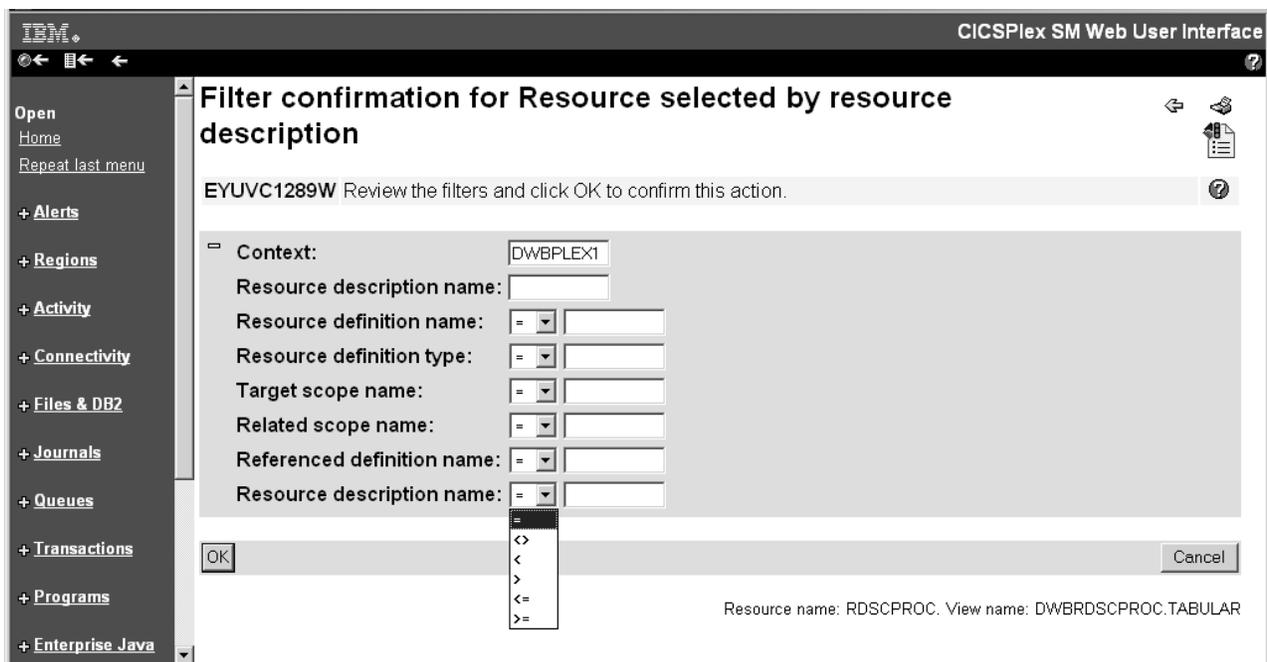


図 37. フィルター確認画面

確認画面のフィルター値をオーバーライドするには、フィールドを上書きし、「了解」をクリックします。

注: フィルター確認は、常に結果セット警告が処理される前に適用されます。

動的選択リスト

このトピックでは、CICSPlex SM Web ユーザー・インターフェース (WUI) の新しい動的選択リスト機能について説明します。

WUI では、TSO EUI と同様の方法で、動的選択リストを生成できるようになりました。選択リストを生成できるフィールドは、テキスト入力ボックスの右側のアイ

コン  でマークされています。リソース入力フィールドの場合、このアイコンをクリックすると、次のような画面がオープンします。

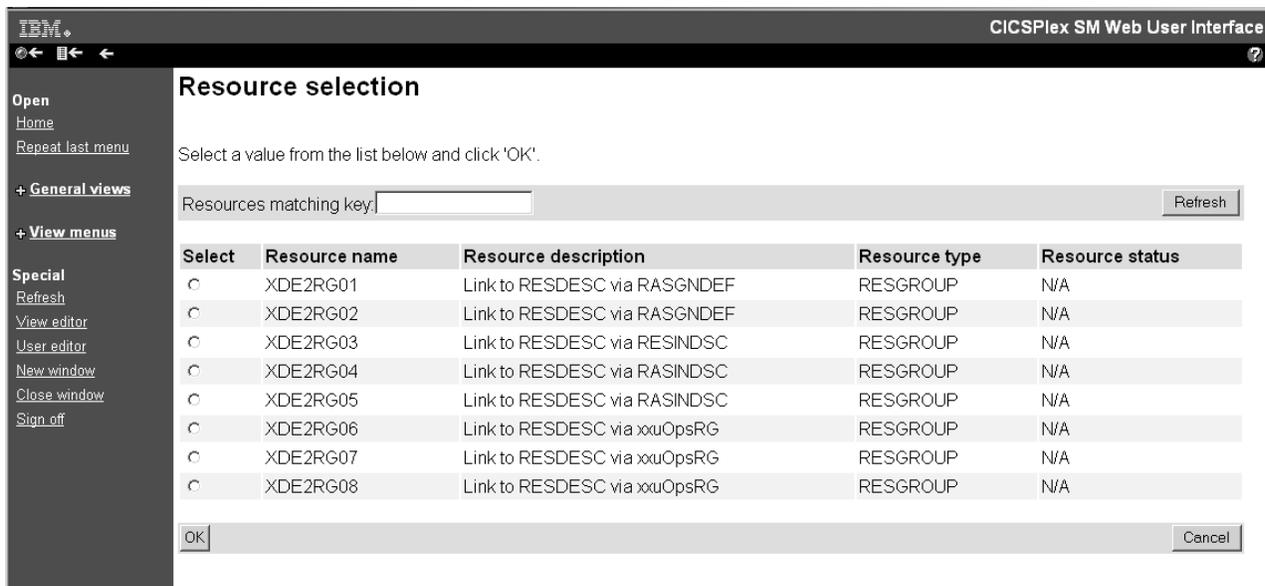


図 38. リソース選択リスト画面

有効な値が、リソース・タイプ別にソートされた表形式で画面に表示されます。リストから選択するには、単にラジオ・ボタンをクリックし、「了解」をクリックします。

選択リスト画面には、リストをフィルタリングするために現在使用されている値が入っている入力フィールドがあります。新しいフィルターを使用するには、入力フィールドに新しい値を入力し、「最新表示」をクリックします。

タスクのガイド、表形式ビュー、および詳細ビューの入力フィールドの属性およびパラメーターのみが、選択リストを生成できます。フィルター、コンテキスト、およびスコープのフィールドは、選択リストを生成することができません。

メタデータ属性要求の選択リスト画面には、属性名別にソートされた基本テーブルの属性すべて、またはフィルタリングされたサブセットが表示されます。これらは、主に定義生成の評価のために使用されます。

フィルター値を指定することにより、選択リストに表示される値の数を削減できます。これらのフィルターは = (等価) 演算子のみを使用するので、通常 WUI で発生するフィルターとは異なります。通常の方法でワイルドカード記号 * (アスタリスク) および + (正符号) を使用して、属性の値を総称名として入力できます (例えば、DW*)。選択リスト画面では、画面上部にあるボックスに表示されるフィルター値を変更することにより、リストに表示される値を変更できます。

注: 総称文字を含んでいない値を入力した場合、この値は、選択リストに値が表示されるかどうかに関係なく、フィルターとして使用されます。これは、現在の EUI の振る舞いと異なります。EUI では、ワイルドカードを含まない属性を指定したときは、値が生成されなかった場合は使用可能なすべての値のリストが生成されます。

画面設計の改良

このトピックでは、WUI のビューおよびメニューの使用可能度および外観を改善する新機能について説明します。

2 つのカラム詳細ビュー

2 つのカラム詳細ビューの導入により、画面に表示されるデータの量が増え、画面のスクロールの必要性が減ります。2 つのカラムのビューの例を以下に示します。

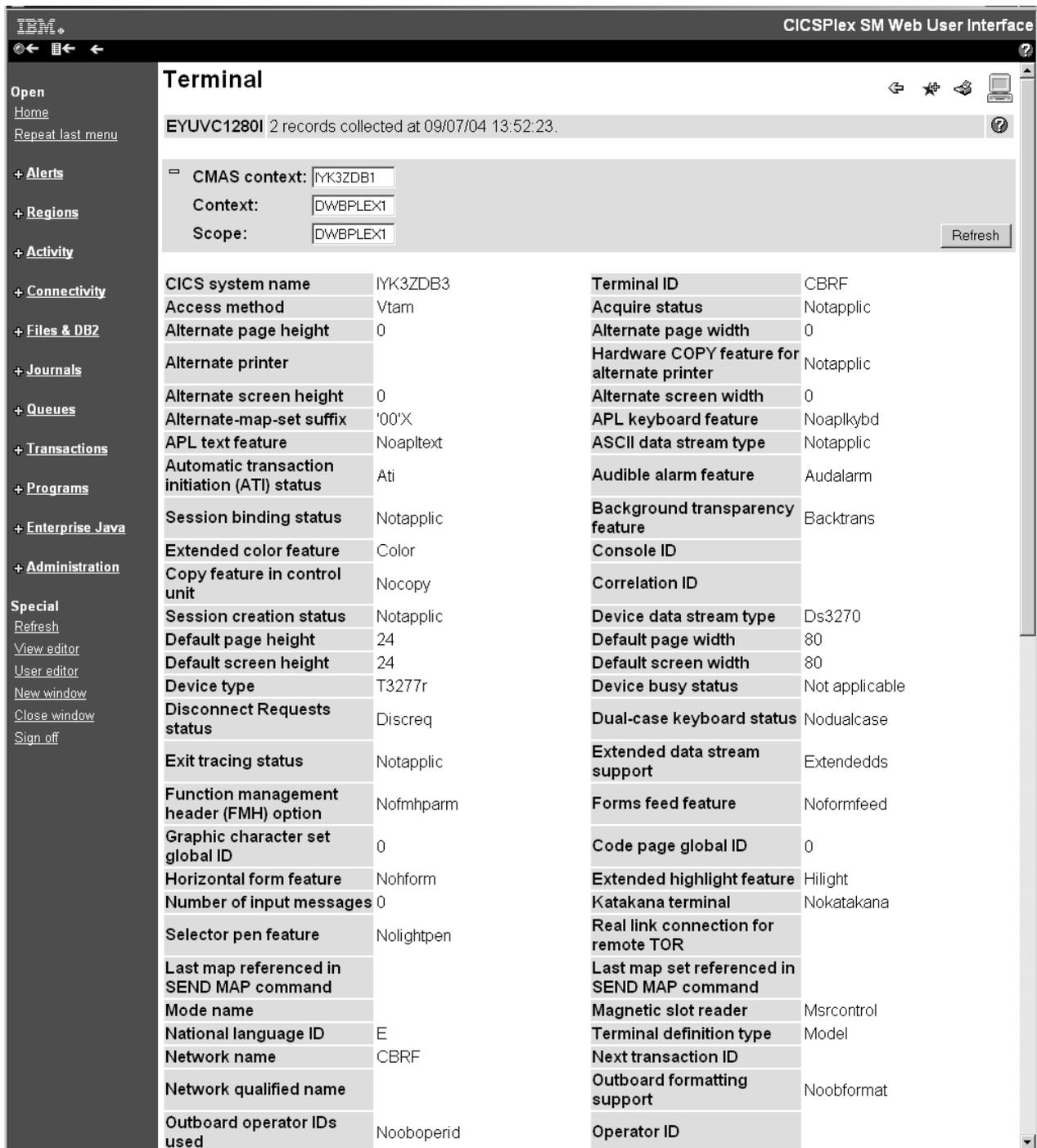


図 39. 2 つのカラム詳細ビューの一部

ビュー・エディターを使用して、2 つのカラム詳細ビューを作成することができます。手順については、321 ページの『2 つのカラム詳細ビューの作成』を参照してください。

フィルターの展開と縮小

表形式のビューのフィルターは、大量の画面スペースを必要とし、表示できるデータが制限される場合があります。変更しないフィルターを縮小表示することにより、この問題を緩和できるようになりました。変更する必要が生

じた場合は、再度フィルターを展開できます。図 40 は、フィルターが展開された状態の表形式ビューの一部を示しています。

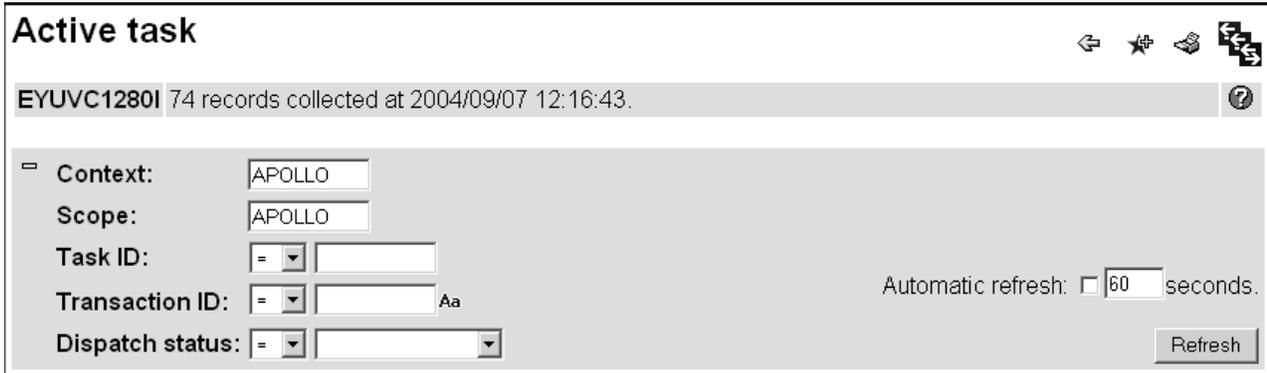


図 40. 表形式ビューの展開されたフィルター域

フィルターを縮小するには、フィルター域の左上にある縮小 (負符号 (-)) アイコンをクリックします。

図 41 は、フィルターが縮小された状態の同じ画面を示しています。表示される情報量は同じですが、データ・フィールドが画面横方向に広がっています。

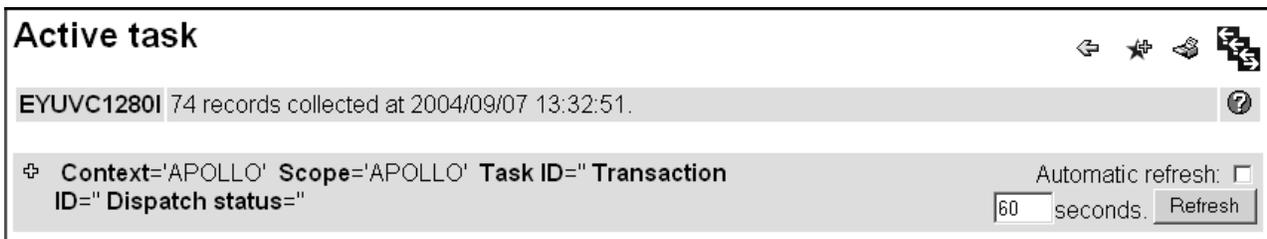


図 41. 縮小されたフィルター域

フィルターを展開するには、拡張フィルター (正符号 (+)) アイコンをクリックします。

デフォルトで表示するフィルターを展開状態にするか縮小状態にするかを決定するために、次の新しいオプションの WUI サーバー初期設定パラメーターを使用します。

FILTERSTYLE(EXPAND/COLLAPSE)

このパラメーターを指定しなかった場合、WUI は、デフォルトでは展開された状態でフィルターを表示します。

デフォルトでない状態のフィルターを持つビューから移動して他の画面を表示した場合、新しいビューのフィルター域は FILTERSTYLE パラメーターで指定されたデフォルトのフィルターの設定に戻ります。

表形式のビューからナビゲートした後すぐに元のビューに戻った場合、フィルター域は、そのビューが前回表示されていたときの状態になります。

「すべて選択」アイコンと「選択をすべて解除」アイコン

表形式ビューでのスペース使用効率を改善する目的で、特定の基準に一致するすべてのリソースを選択または選択解除するために前のリリースで使用されていた「すべて選択」ボタンと「選択をすべて解除」ボタンは、以下のアイコンに置き換えられました。



すべて選択



選択をすべて解除

これらのアイコンは両方とも表形式ビューの「レコード」というラベルの Colum の 2 行目 (ソートと集計のアイコンと同じ表の行) にあります。

「すべて選択」オプションと「選択をすべて解除」オプションの機能では、その他の変更はありません。

2 つのカラム詳細ビューの作成

このトピックでは、WUI のビュー・エディターを使用して 2 つのカラム詳細ビューを作成および更新する方法について説明します。

WUI のビュー・エディターが更新され、2 つのカラム詳細ビューを作成および更新できるようになりました。ビュー・エディターの「ビューの追加」画面が更新され、「2 つのカラム詳細フォーム」と呼ばれる新しいビュー・タイプが追加されました。前のリリースの詳細フォーム・タイプは、1 つのカラム詳細フォームに名前変更されました。

新しいビューの作成プロセスは、前のリリースとほぼ同じです。既存のビュー・セットに新しい 2 つのカラム詳細ビューを追加する際に実行する必要があるステップの概略を以下に示します。

1. ビュー・エディターをオープンし、「ビューの追加」画面にナビゲートします。
 - a. メインメニューのナビゲーション・パネルから、「ビュー・エディター」 > 「ビュー・セット」 > 「編集」をクリックします。これにより、「ビュー・セットのオープン」画面がオープンします。
 - b. 新しい詳細ビューを追加するビュー・セットを選択し、「了解」をクリックします。

注: IBM 提供のビュー・セットやビューを変更する場合は、最初にこれをコピーして名前変更しなければなりません。

これにより、「ビュー・セットの内容」画面が表示されます。この画面には、選択済みビュー・セットに現在属するビューのリストが含まれています。

- c. 画面下部の「追加」をクリックします。これにより、「ビューの追加」画面がオープンします。この画面は、選択済みビュー・セットに対して新しいビューを作成するために使用します。
2. 新しいビューに名前を付け、そのタイプを定義します。
 - a. 「ビュー名」フィールドに新しいビューの名前を入力します。
 - b. ビュー・タイプのリストから「2 つのカラム詳細フォーム」を選択します。

- c. 事前埋め込みオプションのリストから、「キー属性」を選択し、「了解」をクリックします。これにより、「詳細フォームのコンポーネント」画面がオープンします。
3. 左側のカラムに属性を追加します。
 - 2 つのカラム詳細ビューに項目を追加することを選択すると、新しいエレメントが左側のカラムに入れます。左右の位置合わせを維持するために、新しい項目の右側のカラムに自動的に空白エレメントが挿入されます。
 - a. 「フォーム内容」を選択します。これにより、図 42 に示すような「フォーム内容」画面がオープンします。この画面に、2 つのカラムの画面を作成する際に、右側のカラムと左側のカラムのバランスをとるために必要な「スペース (Space)」エレメント・タイプが含まれていることに注意してください。

The screenshot shows the 'Form Contents' interface. On the left is a 'Special' sidebar with links: 'New Home window', 'New editor window', 'Close window', and 'Sign off'. The main area has a title 'Form Contents' and a help icon. Below the title is a text instruction: 'To add a new item to the end of the list, click 'Append'. Select an item and click 'Insert' to add an item above the selected one. To work with an existing item, select the item and click 'Edit', 'Delete' or 'Delete row'. Click 'OK' when you have finished.'

The 'Selected view' section displays:

- View: 2COL_KEY
- View set: DWBTASK

The 'Form contents' section contains a table with the following items:

<input type="radio"/> CICS system name	EYU_CICSNAME	<input type="radio"/> Task ID	TASK
<input type="radio"/> Task priority	PRIORITY	<input type="radio"/> Space	

At the bottom, there are buttons: OK, Append, Insert, Edit, Delete, Delete row, and Cancel.

図 42. 2 つのカラム詳細ビュー用の「フォーム内容」画面

- b. 「追加」をクリックします。これにより、「フォーム項目タイプ」画面がオープンします。
- c. 「属性フィールド」を選択し、「了解」をクリックします。これにより、「フォーム属性フィールド」画面がオープンします。
- d. リストから属性を選択し、「了解」をクリックします。これにより、「フォーム項目のコンポーネント」画面がオープンします。
- e. 以下のようにして、新しい属性の定義を完了します。
 - 属性のタイトルと注釈を入力する
 - 表示オプション (通常またはグラフィカル) を選択する
 - ビュー・リンクを追加する

- f. 「終了」をクリックして、新規ビューに新しい属性を追加します。これにより、「フォーム内容」画面に戻ります。
4. ビューの右側のカラムに別の属性を追加します。

右側のカラムにエレメントを追加するには、空白エレメントを選択してこれを編集することで、必要なタイプのエレメントに変更する必要があります。

- a. ステップ 3 で作成された空白エレメントを選択し、「編集」をクリックします。これにより、「フォーム項目タイプ」画面が再び表示されます。
- b. 右側のカラムに追加するエレメントのタイプを選択し、ステップ 3 で示した手順に従い、これを定義します。

「終了」をクリックして、ビューに右側のエレメントを追加します。

必要に応じてステップ 3 と 4 を繰り返すことで、左側と右側のカラムにさらにエレメントを追加することができます。

右側のカラムから項目を削除するには、項目を選択し、「削除」をクリックします。これにより、項目が空白エレメントに変換されます。

注: 空白エレメントを個々に削除することはできませんが、「行の削除」をクリックすることにより、両方のカラムの隣り合ったエレメントを除去できます。

5. ビュー定義を完了します。
- a. 「フォーム内容」画面で「了解」をクリックします。これにより、「詳細フォームのコンポーネント」画面に戻ります。
 - b. 次に、タイトル、アクション・ボタン、フィルター、コンテキスト、スコープなどの、新しいビューの残りのコンポーネントを追加します。
 - c. 必要なコンポーネントをすべて追加したら、「了解」をクリックして新しいビューを保管し、「ビュー・セットの内容」画面に戻ります。

CICSplex SM API の変更点

新規のビジネス・アプリケーション・サービス設計をサポートするため、CICSplex SM API に対していくつかの変更が加えられました。

ビジネス・アプリケーション・サービスの設計変更のサポートにおける API の変更点

RESINGRP という名前の新しいリソース・テーブルが導入されました。RESINGRP リソースは、リソース・グループ内のリソース定義のメンバーシップを記述します。これには以下の属性が含まれます。

表 12. RESINGRP リソース・テーブル属性

名前	データ・タイプ	送信元	長さ	合計	説明
CHANGETIME	DATETIME	CPSM	8	MAX	定義が変更された最終時刻
RESGROUP-1	CHAR	CPSM	8	DIF	リソース・グループ名
RESNAME-2	CHAR	CPSM	8	DIF	リソース定義名

表 12. RESINGRP リソース・テーブル属性 (続き)

名前	データ・タイプ	送信元	長さ	合計	説明
RESVER-3	BINARY	CPSM	1	AVG	リソース定義のバージョン
RESTYPE-4	CHAR	CPSM	8	DIF	リソース定義のタイプ
DESCRIPTION	CHAR	CPSM	58	DIF	リソース定義の説明

メッセージ

CICSplex SM Web ユーザー・インターフェースに対していくつかの新しいメッセージが追加されました。

Web ユーザー・インターフェースの新しいメッセージ

EYUVC1258W	EYUVC1289W	EYUVC1293I	EYUVC1294I
EYUVC1310I	EYUVC1311I	EYUVC1312W	EYUVC1313E
EYUVC1314I	EYUVC1321E		

Web ユーザー・インターフェースのビュー・エディターの新しいメッセージ

EYUVE0111E	EYUVE0142I	EYUVE0144I	EYUVE0760I
EYUVE0761I	EYUVE0821I	EYUVE0949I	EYUVE0950E
EYUVE0951E	EYUVE0952E	EYUVE0953E	EYUVE0954E
EYUVE0956I	EYUVE0957E	EYUVE0958E	EYUVE0959E
EYUVE0960E	EYUVE0961I	EYUVE0961I	EYUVE0962I
EYUVE0963E	EYUVE0964E	EYUVE0965E	EYUVE0969I
EYUVE0970E	EYUVE0971E	EYUVE0972E	EYUVE0973E
EYUVE0974E	EYUVE0975E	EYUVE0976E	EYUVE0977E
EYUVE0978I	EYUVE0979I	EYUVE0980E	EYUVE0981E
EYUVE0982E	EYUVE0983I	EYUVE0984E	EYUVE0985E

第 12 章 CICSplex SM バッチ・リポジトリ更新機能の拡張

指定された CICSplex SM リポジトリにバッチ更新を実行依頼するために、既存の TSO エンド・ユーザー・インターフェース BATCHREP コマンドに対する代替を提供する新規の方式が導入されました。

BATCHREP にアクセスするための代替の方法は、以下のとおりです。

- 新規 BATCHREP リソース・テーブルを、CPSM API によって参照されるオブジェクトとして使用可能
- 新規 BATCHREP リソース・テーブルの WUI サポート
- BATCHREP 機能を提供するバッチ・ユーティリティー・プログラム

WUI サポートおよびバッチ・ユーティリティーでは、新規 BATCHREP リソース・テーブルを、CPSM API によって参照されるオブジェクトとして利用します。

CICSplex SM バッチ・リポジトリ更新機能の拡張の利点

CICSplex SM TSO エンド・ユーザー・インターフェースの使用に代わるこれらの代替手段の導入により、以下が実現されます。

- WUI 内の新規の BATCHREP ビューを通じた CPSM BATCHREP 機能のアクセス可能性の向上
- 新規 z/OS ジョブ・ステップ・ユーティリティーおよび CPSM API プログラムに対する CPSM BATCHREP 機能へのアクセスの拡張
- ジョブ・ステップからの定義の保守を可能にし、それによって保守をライブラリー制御システムに統合できるようにするバッチ・ユーティリティーの提供

要件

この機能をサポートするための特別なハードウェア要件およびソフトウェア要件はありません。

関連情報

377 ページの『第 27 章 CICS 稼働環境』

バッチ・ユーティリティー・プログラム

新規バッチ・ユーティリティー・プログラムは、z/OS ジョブ・ステップとして実行され、更新されるリポジトリを持つ CMAS に接続されます。バッチ・リポジトリ更新は、その CMAS で実行するよう実行依頼されます。このユーティリティーは、EXECUTE および CHECK アクションを開始できます。CMAS は、CICS Transaction Server for z/OS パージョン 3.1 のレベルでなければなりません。

CICSplex SM アプリケーション・プログラミング・インターフェースの変更点

BATCHREP リソース・テーブルが、CICSplex SM API で使用できるようになりました。

BATCHREP リソース・テーブル用として、次の 2 つのアクションが追加されています。

EXECUTE

指定された CMAS で実行されるよう、バッチ・リポジトリ更新を実行依頼します。CICSplex SM API の CONNECT コマンドの CONTEXT オプションを使用して、CMAS の名前を指定します。

CHECK

バッチ・リポジトリ更新機能入力ファイルに指定されたコマンドを検査します。

これらのアクションの入力パラメーター・フィールドは、表 13 に示す外部属性と対応しています。

表 13. BATCHREP リソース・テーブル属性

名前	データ・タイプ	送信元	長さ	要約	説明
(INPUTDSN-1)	CHAR	CPSM	44	DIF	入力データ・セット名
(INPUTMEMBER-2)	CHAR	CPSM	8	DIF	入力メンバー名
PRINTCLASS	CHAR	CPSM	1	DIF	出力印刷クラス
PRINTNODE	CHAR	CPSM	8	DIF	宛先印刷ノード
OUTPUTUSER	CHAR	CPSM	8	DIF	出力ユーザー ID

CICSplex SM Web ユーザー・インターフェースの変更点

バッチ・リポジトリ更新ジョブ (EYUSTARTBATCHREP) と呼ばれる新しい WUI ビューが、管理ビュー・メニュー (EYUSTARTADMIN) に追加されました。このメニューから新しいビューにアクセスするには、管理ビューの見出しの下の「バッチ・リポジトリ更新ジョブ」をクリックします。この新しいビューは、TSO EUI BATCHREP ビューと等価です。

WUI を使用してバッチ・リポジトリ更新を実行依頼するには、以下のようにします。

- WUI サーバーが、更新するデータ・リポジトリに関連付けられている CMAS に接続されていることを確認します。
- レコードを選択し、「確認」または「実行」ボタンをクリックします。
- TSO EUI BATCHREP ビューで必要とされる入力パラメーターと等価な入力パラメーターを入力します。

メッセージ

以下の新規データ・リポジトリ・メッセージが追加されました。

新規データ・リポジトリ・メッセージ

EYUXD0901E	EYUXD0902E	EYUXD0903E	EYUXD0904E
EYUXD0905E	EYUXD0906E	EYUXD0907E	EYUXD0908I
EYUXD0909I	EYUXD0910E	EYUXD0911E	EYUXD0912E
EYUXD0913E	EYUXD0914E	EYUXD0915E	EYUXD0916E

セキュリティー

外部セキュリティー・マネージャーの使用をサポートする既存の CICSplex SM セキュリティー機能は、BATCHREP リソース・テーブルを参照する API プログラムである新規 z/OS ジョブ・ステップ・ユーティリティーの使用、および BATCHREP ビューを参照するための WUI の使用に対するアクセス制御をインプリメントするために、引き続き使用できます。

マイグレーション

BATCHREP への新しい WUI または CPSM API アクセス、あるいは BATCHREP のための z/OS ジョブ・ステップ・ユーティリティーは、既存の TSO EUI BATCHREP 機能に対する代替手段として使用する機能です。

このユーティリティーが接続する CMAS は、CICSplex SM バージョン 3.1 のレベルでなければなりません。

第 5 部 その他の変更点

第 5 部では、CICS Transaction Server for z/OS バージョン 3 リリース 1 の新機能および機能変更のうち、このリリースの 3 つのテーマの範囲外のものについて説明します。

第 13 章 新規インストール・プロセス

このトピックでは、CICS の新規インストール・プロセスに注目します。

この CICS Transaction Server リリースは、SMP/E RECEIVE、APPLY、および ACCEPT コマンドを使用してインストールします。SMP/E ダイアログは、SMP/E インストール・ステップを完成させるために使用します。

このプロセスについては、「*CICS TS 3.1 Program Directory*」で説明します。このプロセスは、IBM Corporate Standard に従っているため、他の z/OS 製品をインストールしたことのあるユーザーにはおなじみかもしれません。

CICS Transaction Server の従来のインストール方法である DFHISTAR は、現在も使用できます。DFHISTAR に関する情報が「インストール・ガイド」内のどこに記載されているかについては、「*Program Directory*」に示されています。

新規インストール・プロセスの利点

インストールの SMP/E RECEIVE、APPLY、および ACCEPT プロセスは IBM Corporate Standard です。このプロセスは、他の z/OS 製品をインストールしたことのあるユーザーにはおなじみかもしれません。

第 14 章 EXTRACT STATISTICS コマンド

新しく導入された SPI コマンドである EXTRACT STATISTICS は、URIMAP、PIPELINE、および WEBSERVICE リソースに関する統計を取り扱います。

EXTRACT STATISTICS コマンドは、1 つのリソースに関する現行統計、またはリソース・クラスに関するグローバル統計を取得する際に使用します。

EXTRACT STATISTICS コマンドは、COLLECT STATISTICS と同等の機能を、URIMAP、PIPELINE、および WEBSERVICE リソースに対して実行します。他のリソースの統計を収集する場合は、既存の COLLECT STATISTICS コマンドを使用してください。

EXTRACT STATISTICS の構文は、COLLECT STATISTICS とは異なります。

EXTRACT STATISTICS コマンドの利点

すべての CICS SPI コマンドは、サポート可能な別個のオプションの数に制限があります。時間が経つにつれて CICS に新しいリソースが追加されていったため、COLLECT STATISTICS コマンドの限度に達し、既存のコマンドでは新しい URIMAP、PIPELINE、および WEBSERVICE リソースに対応することができません。

EXTRACT STATISTICS コマンドは、CVDA を指定した RESTYPE オプションを使用して、CICS リソースを指定します。このため、コマンドで潜在的にサポートできるリソースの数に制限がありません。ただし、このリリースでは 3 つの新しいリリースのみがサポートされます。

CICS 外部インターフェースの変更点

システム・プログラミング・インターフェースの変更点

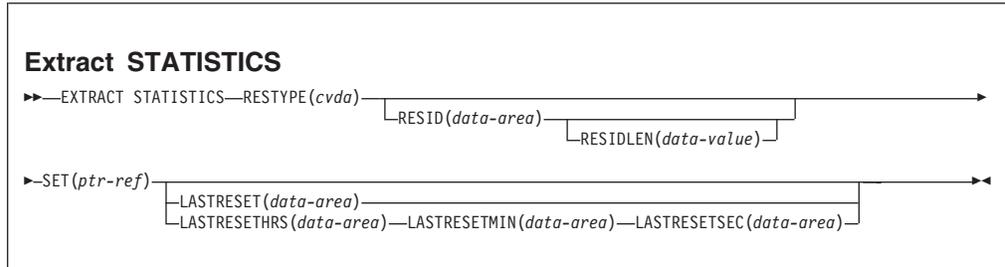
EXTRACT STATISTICS コマンド

COLLECT STATISTICS コマンドの機能を補完するため、CICS に EXTRACT STATISTICS コマンドが追加されました。

PIPELINE、URIMAP、および WEBSERVICE リソースについては、EXTRACT STATISTICS を使用して、1 つのリソースに関する現行統計、またはリソース・クラスに関するグローバル統計を取得します。EXTRACT STATISTICS は、これら 3 つのリソースのみを扱います。他のリソースの統計を収集する場合は、引き続き COLLECT STATISTICS コマンドを使用してください。

EXTRACT STATISTICS は、URIMAP、PIPELINE、および WEBSERVICE リソースに対して、COLLECT STATISTICS と同等の機能を実行します。これは、COLLECT STATISTICS の拡張ではこの機能を提供できないため（このコマンドの設計上の制限による）です。COLLECT STATISTICS と比較すると、EXTRACT STATISTICS は

構文が異なり、今後の無限の拡張に備えています。



条件: INVREQ、IOERR、LENGERR、NOTAUTH、NOTFND

説明

EXTRACT STATISTICS コマンドは、呼び出しアプリケーションに、特定のリソースに関する現行統計、または特定のタイプのリソースに関する全体的な統計を戻します。

CICS がユーザーに提供する統計は、統計の最後の抽出間隔の満了、1 日の終わりの満了、または要求されたリセット後に集計された統計です。(SMF データ・セットに既書き込まれている統計には、アクセスできません。) EXTRACT STATISTICS コマンドでは、統計カウンターはリセットされません。

CICS は、このコマンドから戻されたデータを格納する十分なストレージを取得し、この領域へのポインターを戻します。この領域の先頭 2 バイトには、領域の長さが格納されます。このストレージは、後続の EXTRACT STATISTICS コマンドで再利用可能であるため、コマンドの次の発行以降に必要なデータがあれば他の場所に保管する必要があります。CICS は、タスクの終了時にこのストレージを解放します。

すべてのリソース・タイプで、グローバル統計と特定の統計の両方の統計が提供されるわけではありません。表 14 に、リソース・タイプごとの提供可能な統計と、提供可能な統計セットごとのコピーブック名を示しています。コピーブックは、戻される統計の形式を定義しています。グローバル統計の欄にコピーブック名が示されていない場合、そのリソース・タイプではグローバル統計は提供されません。同様に、特定の統計の欄に入力項目がない場合、個々のリソースに関する統計を取得できません。

表 14. リソース・タイプと統計

リソース・タイプ	CVDA	RESIDLEN	統計のタイプ	グローバル統計	特定の統計
PIPELINE	1124	Char(8)	PIPELINE	--	DFHPIRDS
URIMAP	1173	Char(8)	URIMAP	DFHWBGDS	DFHWBRDS
WEBSERVICE	1174	Char(32)	WEBSERVICE	--	DFHPIWDS

ASSEMBLER、COBOL、および PL/I でコピーブックが提供されます。(C のコピーブックはありません。) コピーブックの名前は、各言語で同じです。以下のライブラリーで見つけることができます。

ASSEMBLER	CICSTS31.CICS.SDFHMAC
COBOL	CICSTS31.CICS.SDFHCOB
PL/I	CICSTS31.CICS.SDFHPL1

注: コピーブックには、パック 10 進数フィールドを含んでいるものがあります。このフィールドを使用する前に、16 進数のゼロの有無を検査する必要があります。COBOL 版のこのフィールドは、この目的で、接尾部 -R を持つ数値として再定義されました。

オプション

LASTRESET(*data-area*)

要求された統計のカウンターが最後にリセットされた時刻を示す 4 バイトのパック 10 進数フィールドを戻します。これは、通常、最後の間隔の満了の時刻になります。最後のリセット時刻は、通常現地時間で戻されます。

リセット時刻には、次の 2 つの形式があります。

- 複合 (パック 10 進数フォーマット 0hhmmss+)。LASTRESET オプションを使用して取得します。
- 時、分、秒を分離する。それぞれ、LASTRESETHRS、LASTRESETMIN、および LASTRESETSEC オプションを指定して取得します。

LASTRESETHRS(*data-area*)

要求された統計のカウンターが最後にリセットされた時刻のうち (LASTRESET オプションを参照) 時の部分を示すフルワード・バイナリー・フィールドを戻します。

LASTRESETMIN(*data-area*)

要求された統計のカウンターが最後にリセットされた時刻のうち (LASTRESET オプションを参照) 分の部分を示すフルワード・バイナリー・フィールドを戻します。

LASTRESETSEC(*data-area*)

要求された統計のカウンターが最後にリセットされた時刻のうち (LASTRESET オプションを参照) 秒の部分を示すフルワード・バイナリー・フィールドを戻します。

RESID(*data-area*)

統計を抽出する対象のリソースの名前を指定します。このキーワードを指定しない場合、グローバル統計を抽出することを意味します。

RESIDLEN(*data-value*)

RESID データ域の長さを指定します。このオプションを省略した場合のデフォルト値は、334 ページの表 14 に指定した長さになります。

RESTYPE(*cvda*)

指定された *cvda* に応じた特定のリソース・タイプに関する統計を要求します。有効な CVDA 値は以下のとおりです。

PIPELINE

PIPELINE の統計を要求します。RESID で特定の PIPELINE を示します。

URIMAP

URIMAP の統計を要求します。RESID で特定の URIMAP を示します。

WEBSERVICE

WEBSERVICE の統計を要求します。RESID で特定の WEBSERVICE を示します。

SET(*ptr-ref*)

ポインター参照を、戻された統計を格納するデータ域のアドレスに設定することを指定します。データ域の先頭 2 バイトには、ハーフワード・バイナリー形式のデータ域の長さが格納されます。

第 15 章 大/小文字混合パスワードのサポート

CICS で使用されるセキュリティー・マネージャーが大/小文字混合のパスワードの使用をサポートしている場合、CICS Transaction Server for z/OS バージョン 3 リリース 1 はパスワードを大文字に変換せずにセキュリティー・マネージャーに渡します。

CICS では、以下に示すような、ユーザーがパスワードを入力できるいくつかの箇所があります。

リソース定義

以下のリソース定義には、PASSWORD 属性があります。

FILE
TCPIPSERVICE
TERMINAL

API コマンド

以下のコマンドには、PASSWORD オプションがあります。

CHANGE PASSWORD
VERIFY PASSWORD
SIGNON

サインオン・トランザクション (CESN)

トランザクションが、パスワードを入力できる次の 2 つのフィールドを提供します。

パスワード
新規パスワード

どの場合でも、CICS によるこれらのパスワードの処理方法は、CICS で使用される外部セキュリティー・マネージャーが大/小文字混合のパスワードをサポートするかどうかによって異なります。

- セキュリティー・マネージャーが大/小文字混合のパスワードをサポートする場合、CICS はユーザーが指定したパスワードを変更せずにセキュリティー・マネージャーに渡します。
- サポートしない場合は、CICS はパスワードを大文字に変換してから、セキュリティー・マネージャーに渡します。

第 16 章 コード・ページ変換の変更点

CICS は、EBCDIC または ASCII データおよび Unicode データの間で変換を行う手段を提供するように、拡張されました。変換はどちらの方向にも行われます。

- これらの変換は、z/OS Unicode 変換サービスを利用します。通常このような変換は、HTTP 要求の処理または CICS TS 3.1 で導入された新規 CONTAINER API コマンドの使用の一部として実行されます。
- CICS は既に、「*CICS Family: Communicating from CICS on System/390*」で示される EBCDIC コード・ページと ASCII コード・ページのどのような組み合わせの間でも、データの変換をサポートしています。
- このサポートは拡張され、EBCDIC コード・ページと ASCII コード・ページ、および UTF-8 と UTF-16 のいずれかの間でのデータの変換が可能になりました。Unicode の UTF-8 形式および UTF-16 形式の間の変換もサポートされます。

CICS の以前のリリースで提供されるデータ変換方式ではなく、ここで説明するデータ変換方式を使用するには、「*CICS Application Programming Guide*」で説明するように、チャンネルまたはコンテナを使用してデータを通信していなければなりません。

CICS 文書および文書テンプレートは、UTF-8 および UTF-16 文字エンコードから、または UTF-8 および UTF-16 文字エンコードへ変換することはできません。これらが CICS Web サポートでの静的応答として使用されるとき、EXEC CICS WEB API コマンドに응答して CICS によって取得された場合も、EXEC CICS DOCUMENT RETRIEVE コマンドを使用してアプリケーション・プログラムによって取得された場合も、この制限が適用されます。

「*z/OS Support for Unicode: Using Conversion Services*」、SA22-7649 の付録 F に、これらのサービスを通じてサポートされる変換が示されています。CICS は、z/OS 変換サービスを利用することにより、これらの文字変換をすべてサポートするようになりました。

コード・ページ変換の変更点の利点

CICS では、以下の操作が可能になりました。

- UTF-8 および UTF-16 データを入力として受け入れる。
- 受け取ったデータを別のエンコード形式に変換する。
- その後リターン時に結果を変換して適切な UTF 形式に戻す。

変換に使用可能なコード・ページのペアの範囲が大幅に拡張されました。

変換のためのデータは、主に HTML、XHTML、および XML であるものと予期されますが、新規機能はこれらに制限されず、自身を UTF として識別する規則を順守するインターフェースからの入力データは、適切な変換によって処理されます。

用語

CICS 用語集に、以下の用語が追加されました。

CCSID

『コード化文字セット ID』を参照してください。

コード化文字セット ID

エンコード・スキーム ID、文字セット ID (複数可)、コード・ページ ID (複数可)、および追加のコーディング関連の必要情報の特定のセットを示す 16 ビットの数値。コード化グラフィックス文字表現を一意に識別します。

頭字語: CCSID。

UNICODE

現代の任意の言語で書かれたテキストの交換、処理、および表示をサポートする世界共通の文字エンコード標準。いくつかの言語の多くの古典的および歴史的テキストもサポートします。Unicode 標準では、ISO 10646 によって 16 ビットの国際文字セットが定義されています。

要件

コード・ページ変換に対するこれらの機能強化のハードウェア要件およびソフトウェア要件は、概して CICS TS と同じです。

CICS は、z/OS が提供する z/OS Unicode 変換サービスを利用して、これらの機能強化をインプリメントします。

CICS 外部インターフェースの変更点

インストールの変更点

CICS インストール・プロセスに必要な変更はありません。ただし、UTF-8 または UTF-16 データの変換のサポートに依存する要件がある場合、z/OS 変換サービスを使用可能にし、CICS に実行させたい変換を指定する変換イメージをインストールする必要があります。

オペレーション・システム・サービスを通じてサポートされる変換をセットアップ及び構成するために必要なステップを調べるには、「*z/OS Support for Unicode: Using Conversion Services*」、SA22-7649 の説明を参照してください。

z/OS 変換サービスが使用可能でない場合、このことを示すメッセージが CICS によって発行されます。これらのサービスが必要ではない場合、そのメッセージを抑制することができます。

- これらのサービスを利用すると予期される CICS 領域の開始時にこのメッセージが出された場合、z/OS 変換サービスを使用可能にするには IPL が必要です。

IPL 後に z/OS 変換サービスの状況を調べるには、MVS コンソールから以下のコマンドのいずれかを使用します。

/D UNI z/OS 変換サービスが使用可能になったかどうかを示します。

/D UNI,ALL

z/OS 変換サービスが使用可能になったかどうか、およびシステムがどの変換をサポートするかを示します。

この詳細については、「*z/OS Support for Unicode: Using Conversion Services*」、SA22-7649 を参照してください。

システム初期設定パラメーターの変更点

データ変換の機能拡張のための新しいシステム初期設定があります。新しいパラメーターは、次のとおりです。

LOCALCCSID={037|CCSID}

ローカル領域のデフォルト CCSID を指定します。

CCSID は 8 文字までの値です。CCSID 値を指定しない場合は、デフォルトの LOCALCCSID が 037 に設定されます。有効な CCSID のリストについては、以下を参照してください。

- 「*CICS Family: Communicating from CICS on System/390*」
- 「*z/OS Support for Unicode: Using Conversion Services*」、SA22-7649 の付録 F

アプリケーション・プログラミングの変更点

「*CICS Application Programming Guide*」に以下の情報が追加されました。

「*CICS Family: Communicating from CICS on System/390*」で示される EBCDIC コード・ページと ASCII コード・ページのどのような組み合わせに対しても提供されるデータ変換サポートに、z/OS Unicode 変換を使用した CICS 内部のデータ変換が追加されます。

Unicode が関わる変換は、通常、HTTP 要求の処理または CICS TS 3.1 で導入された CONTAINER API コマンドの使用の一部として実行されます。UTF-8 と UTF-16 のいずれか、および EBCDIC コード・ページと ASCII コード・ページを変換先または変換元とするデータの変換は、適切な変換イメージの選択によって決まります。Unicode の UTF-8 形式および UTF-16 形式の間の変換もサポートされます。

CICS の以前のリリースで提供されるデータ変換方式ではなく、ここで説明するデータ変換方式を使用するには、「*CICS Application Programming Guide*」で説明するように、チャンネルまたはコンテナを使用してデータを通信していなければなりません。

CICS 文書および文書テンプレートは、UTF-8 および UTF-16 文字エンコードから、または UTF-8 および UTF-16 文字エンコードへ変換することはできません。これらが CICS Web サポートでの静的応答として使用されるとき、EXEC CICS WEB API コマンドに응答して CICS によって取得された場合も、EXEC CICS DOCUMENT RETRIEVE コマンドを使用してアプリケーション・プログラムによって取得された場合も、この制限が適用されます。

「z/OS Support for Unicode: Using Conversion Services」、SA22-7649 の付録 F に、これらのサービスを通じてサポートされる変換が示されています。CICS は、z/OS 変換サービスを利用することにより、これらの文字変換をすべてサポートするようになりました。

必要な変換イメージが使用可能であることの確認

CICS アプリケーションが実行される CICS 領域で特定の変換が使用可能になるように、これらのアプリケーションの一部として使用される CCSID を、z/OS 変換イメージの保守を担当するシステム・プログラマーに知らせておく必要があります。

CCSID 1200 の処理

CICS は、CCSID 1200、1201、1202 のいずれかを使用する UTF-16 が関係する変換をサポートします。z/OS 変換サービスでは、ビッグ・エンディアン形式で CCSID 1200 を使用できますが、リトル・エンディアン形式および CCSID 1201、1202 はサポートしていません。CICS は、変換のためにデータを z/OS に渡す前に、これらのサポートされない形式で示される任意のソース・データをビッグ・エンディアン形式の 1200 に変換します。ターゲット・データが、サポートされない形式のいずれかである場合、CICS はデータをビッグ・エンディアン形式の 1200 として受け取り、それを必要な CCSID に変換します。ターゲット CCSID が 1200 である場合、CICS はエンコードがビッグ・エンディアン形式であるものと想定します。変換がこれらの CCSID のいずれかの間で行われる場合、CICS は z/OS 変換サービスを呼び出さずに変換を実行します。

これらのいずれかの形式の UTF-16 が関係する変換に使用する z/OS 変換イメージを設定する場合、CCSID 1200 を指定する必要があります。変換イメージ作成の試行時に、CCSID 1201 および 1202 は z/OS によって認識されません。

CICS は、インバウンド変換のバイト・オーダー・マーカに従いますが、関連するアウトバウンド変換処理時にその情報を保存することができません。CCSID 1200 のすべてのアウトバウンド・データは UTF16-LE です。アプリケーション・プログラマーは、このことを認識し、必要な場合は自分で LE から BE への変換を実行する必要があります。

変換イメージの共用

- APAR OA05744 の PTF が適用されない場合は、CICS が使用することを意図した z/OS イメージ内にインストールされた変換に対して、検索順序を指定しないでください。
- COBOL で同じ変換が必要な場合、以下の 2 つの別個のステートメントによって変換イメージを定義する必要があります。
 - 検索順序を指定しないステートメント
 - 明示的に検索順序「RECLM」を指定したステートメント

例えば、次のようになります。

```
CONVERSION 850,037;  
CONVERSION 850,037,RECLM;
```

APAR がインストールされると、CICS および COBOL は、デフォルトの検索順序を暗黙的または明示的に指定する、サポートされる変換を利用することができ、イメージ生成ファイルで 2 つの制御ステートメントを提供する必要がなくなります。

JAVA プログラム

コード・ページ変換機能が JAVA 内部に存在するため、それらを CICS 内に複製する必要はありません。ここで説明する変換機能は JAVA プログラムまで影響が及びません。例については、「*Java applications in CICS*」を参照してください。

CICS ユーティリティーの変更点

ダンプ・フォーマッター

新規制御ブロックを表示する AP ドメイン内部に新規コンポーネントがあります。コンポーネント ID は CV です。

問題判別の変更点

メッセージ

z/OS 変換サービスの起動時に障害が発生した結果、2 つの新規メッセージが発行されます。

- サービスが使用可能になっていないために、この CICS 領域で Unicode 変換がサポートされないことを示すコンソール・メッセージが、CICS 初期設定時に発行されます。
- 2 つの特定の CCSID 間の特定の变換が、このシステムでサポートされていないことを示すメッセージが発行されます。

異常終了

- 新しい異常終了コードが導入されました。
- DFHCCNVG のリカバリー・ルーチンによって、障害応答がパーコレートされ、新規関数の呼び出し元に戻されます。

第 17 章 デフォルト・コード・ページの定義の単純化

このトピックでは、CICSplex の構成に必要な変換テーブルの数を減らすために、クライアントまたはサーバーのデフォルトのコード・ページを、システム初期設定テーブルで定義する方法について説明します。

CICS システム間でデータが受け渡されるときはいつでも、全データまたはデータの一部を ASCII 形式から EBCDIC 形式へ、あるいはその逆に変換することが必要な場合があります。変換を必要とする各リソース用の変換テンプレートに含まれている DFHCNV 変換テーブルを使用すると、データ変換が容易になります。

CICS Transaction Gateway for z/OS、CICS Transaction Server for Windows などの CICSplex の特定の分散コンポーネントでは、変換テーブルに指定されたクライアントのデフォルト・コード・ページを指定変更することができません。変換テーブルに接尾部がないため、それぞれが異なるライブラリー内に存在し、デフォルトのコード・ページのみが異なる 2 つのテーブルが必要になることがあります。

必要となる変換テーブルの数を減らすために、クライアントまたはサーバーのデフォルトのコード・ページをシステム初期設定テーブルで定義するように指定できるようになりました。

クライアントのコード・ページの場合

1. DFHCNV TYPE=ENTRY および TYPE=SELECT マクロで、CLINTCP パラメーターの値 SYSDEF を指定します。
2. システム初期設定テーブルで、CLINTCP パラメーターに値を指定することにより、クライアントのデフォルト・コード・ページを設定します。DFHCNV マクロの CLINTCP パラメーターでサポートされる任意の値を指定できます。デフォルトは CLINTCP=437 です。

サーバーのコード・ページの場合

1. DFHCNV TYPE=ENTRY および TYPE=SELECT マクロで、SRVERCP パラメーターに値 SYSDEF を指定します。
2. システム初期設定テーブルで、SRVERCP パラメーターに値を指定することにより、サーバーのデフォルト・コード・ページを設定します。DFHCNV マクロの SRVERCP パラメーターでサポートされる任意の値を指定できます。デフォルトは SRVERCP=037 です。

改良されたデータ変換テンプレートへのコード・ページのデフォルト設定の利点

DFHCNV マクロへの直接の定義に代わって採用された、システム初期設定テーブルでコード・ページのデフォルトを定義する機能により、保守する必要がある DFHCNV 変換テーブルの数が減ったため、CICSplex の定義と管理が簡単になります。

要件

特別なハードウェア要件およびソフトウェア要件はありません。

CICS 外部インターフェースの変更点

システム初期設定パラメーターの変更点

システム初期設定テーブルでクライアントおよびサーバーのデフォルトのコード・ページを定義するために、CLINTCP と SRVERCP の 2 つの新しいシステム初期設定パラメーターが導入されました。これらのパラメーターは、同じ名前の既存の DFHCNV マクロ・パラメーターと対応します。

新しく導入されたシステム初期設定パラメーター

CLINTCP={437|codepage}

DFHCNV データ変換テーブルが使用するクライアントのデフォルト・コード・ページを指定します。ただし、DFHCNV マクロの CLINTCP パラメーターが SYSDEF に設定されている場合にのみ使用されます。*codepage* は 8 文字以内のフィールドで、DFHCNV マクロの CLINTCP パラメーターによってサポートされる値を指定することができます。

SRVERCP={037|codepage}

DFHCNV データ変換テーブルが使用するサーバーのデフォルト・コード・ページを指定します。ただし、DFHCNV マクロの SRVERCP パラメーターが SYSDEF に設定されている場合にのみ使用されます。*codepage* は 8 文字以内のフィールドで、DFHCNV マクロの SRVERCP パラメーターによってサポートされる値を指定することができます。

ユーザーが置換可能なプログラムの変更点

新しく追加された DFHCNV マクロ・パラメーターのオペランド

TYPE=INITIAL および TYPE=ENTRY マクロ・パラメーター CLINTCP と SRVERCP に、新しくオペランド SYSDEF が追加されました。これらのマクロは、ユーザーが置換可能なデータ変換テーブル DFHCNV を定義します。

- DFHCNV TYPE=INITIAL マクロは、変換テーブルの開始を定義します。このマクロは、有効なコード・ページのリストを指定します。
- DFHCNV TYPE=ENTRY マクロは、データ・リソースを一意に識別する名前とタイプを指定します。変換を必要とするリソースごとに 1 つずつ、このマクロを指定する必要があります。

変更されたパラメーター形式は、次のようになります。

CLINTCP={437|SYSDEFInnnn [, nnnn, ...]}

第 1 オペランドでは、DFHCNV TYPE=ENTRY マクロで CLINTCP および CDEPAGE オペランドが省略されたときに使用されるクライアントのデフォルトのコード・ページを定義します。

SYSDEF は、クライアントのデフォルト・コード・ページを、システム初期設定テーブル・パラメーター CLINTCP で決定することを示します。

さらにコード・ページを指定することができます。指定したコード・ページの検証は行われますが、使用されません。

SRVERCP={037|SYSDEFInnnn [, nnnn, ...]}

第 1 オペランドでは、DFHCNV TYPE=ENTRY マクロで SRVERCP および CDEPAGE オペランドを省略したときに使用されるクライアントのデフォルトのコード・ページを定義します。

SYSDEF は、クライアントのデフォルト・コード・ページを、システム初期設定テーブル・パラメーター SRVERCP で決定することを示します。

さらにコード・ページを指定することができます。指定したコード・ページの検証は行われますが、使用されません。

第 18 章 64 ビット・アドレッシング容認の変更点

CICS は、64 ビット・アドレッシング・アーキテクチャーを使用するタスクが異常終了したときに、意味のある情報を提供できるようになりました。

CICS では 64 ビット・アドレッシングの実行はサポートされませんが、プログラムは、CICS が 64 ビット・アーキテクチャーのマシン上で稼働する場合にのみ使用可能なアドレスでストレージを使用できます。これらの変更によって、CICS 異常終了収集メカニズムが拡張され、フル 64 ビット汎用レジスターの内容が収集されるようになりました。

64 ビット・アドレッシング容認の変更点の利点

64 ビット・アドレッシング・コードを開発しているカスタマーは、タスクが異常終了したときに、より多くの情報にアクセスできるようになりました。

要件

これらの拡張のハードウェアおよびソフトウェア要件は、この CICS TS リリースにおけるその他の部分と同じです。

CICS 外部インターフェースの変更点

CICS ユーティリティーの変更点

ダンプ・フォーマッター

CICS ダンプ・フォーマッターは、異常終了発生時に収集された 64 ビット汎用レジスターの内容を表示します。

問題判別の変更点

フル 64 ビット汎用レジスター・ファイルの内容が収集されて、ユーザーが使用できるようになりました。

新規のメッセージや異常終了コード、トレース・ポイントはありません。

第 6 部 廃止された機能

CICS Transaction Server for z/OS バージョン 2 でサポートされていた一部の機能は、CICS Transaction Server for z/OS バージョン 3 リリース 1 では廃止またはスコープが狭められました。

第 20 章 OS/VS COBOL プログラムのランタイム・サポートの中止

OS/VS COBOL プログラムに対するランタイム・サポートが提供されなくなりました。

CICS Transaction Server for z/OS バージョン 2 でランタイム・サポートが提供されていた OS/VS COBOL プログラムは、CICS TS for z/OS バージョン 3 では実行できません。

OS/VS COBOL プログラムを、COBOL に準拠する言語環境プログラムにアップグレードし、CICS がサポートする COBOL コンパイラーのレベルと対して再コンパイルする必要があります。コンパイラーとして、Enterprise COBOL for z/OS and OS/390[®] バージョン 3 を使用することをお勧めします。

COBOL およびその他の言語でサポートされるコンパイラーの詳細については、385 ページの『第 29 章 高水準言語サポート』 および「CICS TS for z/OS バージョン 3 *CICS Application Programming Guide*」を参照してください。「*CICS Application Programming Guide*」の付録 B では、OS/VS COBOL プログラムから、COBOL に準拠する言語環境プログラムへの変換に役立つ情報が記載されています。

新しく導入された異常終了コード ALIK は、OS/VS COBOL プログラムの試行を示します。この状態では、CICS がタスクを異常終了し、プログラムを使用不可にします。このとき、CICS の処理は続行します。

第 21 章 BTAM および TCAM サポートに対する変更点

CICS Transaction Server for z/OS バージョン 3 リリース 1 では、CICS での基本通信アクセス方式 (BTAM) のサポートは中止されました。通信アクセス方式 (TCAM) のサポートは、DCB インターフェースの間接的なサポートに限られます。

BTAM サポートの中止

CICS Transaction Server for z/OS バージョン 3 リリース 1 は、基本通信アクセス方式 (BTAM) をサポートしません。

CICS のいくつかのリリースでは、BTAM 端末の間接的 (すなわち、端末の接続先のバックレベルの端末専用領域 (TOR) からのトランザクション・ルーティングを介した) サポートのみが提供されていました。CICS TS for z/OS バージョン 3.1 では、この間接的サポートは除去されました。BTAM はサポートされなくなり、BTAM に対するすべての言及は除去されました。

これは、バックレベルの CICS TOR に接続された BTAM 端末のネットワークを使用している場合、BTAM 端末からのトランザクションを (前のリリースの CICS で可能であったように) CICS TS for z/OS バージョン 3.1 アプリケーション専用領域 (AOR) にルーティングできないことを意味します。端末をアップグレードするか、前のバージョンの CICS にルーティングする必要があります。

TCAM に対する CICS サポートの変更点

CICS Transaction Server for z/OS バージョン 3 リリース 1 は、TCAM/ACB インターフェースをサポートしません。TCAM/DCB インターフェースは、間接的にサポートされます。

CICS のいくつかのリリースでは、TCAM の ACB インターフェースの間接的 (すなわち、端末の接続先のバックレベルの端末専用領域 (TOR) からのトランザクション・ルーティングを介した) サポートのみが提供されていました。CICS TS for z/OS バージョン 3.1 では、この間接的サポートは除去されました。

前のリリースの CICS では、TCAM の DCB インターフェースは完全にサポートされていました。つまり、以下のようになります。

1. TCAM/DCB を使用して、端末を現行レベルの CICS TOR に接続できた。
2. TCAM/DCB 接続端末によって開始されたトランザクションは、現行レベルの CICS AOR にルーティングできた。

CICS TS for z/OS バージョン 3.1 では、2 番目の操作のみがサポートされます。

このような変更の結果、以下のようになります。

- TCAM の ACB インターフェースによってバックレベルの CICS TOR に接続された端末のネットワークを使用している場合、これらの端末からのトランザクションを (前のリリースの CICS で可能であったように) CICS TS for z/OS バージョン 3.1 AOR にルーティングできません。TCAM/DCB または (できれば) ACF/VTAM を使用するように接続をマイグレーションするか、前のバージョンの

CICS にルーティングする必要があります。(TCAM/ACB をサポートするすべての端末は、ACF/VTAM もサポートします。)

- TCAM の DCB インターフェースによって、例えば CICS TS 2.3 の TOR に接続された端末のネットワークを使用している場合、TOR を CICS TS for z/OS バージョン 3.1 にマイグレーションできません。これを行うためには、ACF/VTAM を使用するように接続をマイグレーションする必要があります。
- TCAM の DCB インターフェースによってバックレベルの CICS TOR に接続された端末のネットワークを使用している場合、これらの端末からのトランザクションは (前のリリースの CICS で可能であったように) CICS TS for z/OS バージョン 3.1 AOR にルーティングできます。ただし、ACF/VTAM を使用するように接続をマイグレーションすることをお勧めします。

CICS 外部インターフェースの変更点

システム初期設定パラメーターの変更点

DFHTCAM={NO | YES} システム初期設定パラメーターは廃止され、前のリリースの CICS との互換性のためだけに保持されています。これを指定した場合、メッセージが発行されて拒否され、TCAM=NO が指定されたと見なされます。

リソース定義の変更点

CICS は、BTAM 端末をサポートしなくなりました (間接的にもサポートしません)。したがって、リモート・リソースとしても、BTAM 端末を定義できなくなりました。

CICS は、ローカルの TCAM 端末をサポートしなくなりました以下のリソース定義マクロは、ローカルの TCAM 端末を定義するためには使用できなくなりました。

- DFHTCT TYPE=SCSDI
- DFHTCT TYPE=LINE
- DFHTCT TYPE=TERMINAL

リモートの TCAM 端末は、引き続き定義することができます。これは、以下のいずれかを使用して行うことができます。

- 単一の DFHTCT TYPE=REMOTE マクロ。
- DFHTCT TYPE=REGION マクロの後に、DFHTCT TYPE=LINE および DFHTCT TYPE=TERMINAL マクロを指定する。CICS は、DFHTCT TYPE=LINE および DFHTCT TYPE=TERMINAL マクロのリモート属性のみを使用します。

アプリケーション・プログラミング・インターフェースの変更点

以下の BTAM 関連の EXEC CICS API コマンドは廃止されました。

- CONVERSE (SYSTEM/3)
- CONVERSE (SYSTEM/7)
- CONVERSE (2741)
- CONVERSE (2770)
- CONVERSE (2780)

- CONVERSE (3600 BTAM)
- CONVERSE (3735)
- CONVERSE (3740)
- ISSUE COPY (3270 ディスプレイ)
- RECEIVE (SYSTEM/3)
- RECEIVE (SYSTEM/7)
- RECEIVE (2741)
- RECEIVE (3600 BTAM)
- RECEIVE (3735)
- RECEIVE (3740)
- SEND (SYSTEM/3)
- SEND (SYSTEM/7)
- SEND (2741)
- SEND (3600 BTAM)
- SEND (3735)
- SEND (3740)

グローバル・ユーザー出口ルーチンの変更点

端末管理プログラム内の以下の (TCAM 入出力イベントで呼び出されていた) グローバル・ユーザー出口ルーチンは、呼び出されなくなりました。

- XTCTIN
- XTCTOUT

ユーザーが置換可能なプログラムの変更点

ローカルの TCAM 端末はサポートされなくなったため、TCAM 端末に対して端末エラー・プログラムは呼び出されません。順次デバイスに対しては、引き続き呼び出されます。

サンプル・プログラムの変更点

DFHSPTM1 および DFHSPTM2 のサンプルの TCAM プログラムは、CICS では提供されなくなりました。

マイグレーション

バックレベルの CICS 端末専用領域 (TOR) に接続された BTAM 端末のネットワークを使用している場合、BTAM 端末からのトランザクションを (前のリリースの CICS で可能であったように) CICS TS for z/OS バージョン 3.1 アプリケーション専用領域 (AOR) にルーティングできません。端末をアップグレードするか、前のバージョンの CICS にルーティングする必要があります。

TCAM の ACB インターフェースによってバックレベルの CICS TOR に接続された端末のネットワークを使用している場合、これらの端末からのトランザクションを (前のリリースの CICS で可能であったように) CICS TS for z/OS バージョン 3.1 AOR にルーティングできません。TCAM/DCB または (できれば) ACF/VTAM

を使用するように接続をマイグレーションするか、前のバージョンの CICS にルーティングする必要があります。(TCAM/ACB をサポートするすべての端末は、ACF/VTAM もサポートします。)

TCAM の DCB インターフェースによって、例えば CICS TS 2.3 の TOR に接続された端末のネットワークを使用している場合、TOR を CICS TS for z/OS バージョン 3.1 にマイグレーションできません。これを行うためには、ACF/VTAM を使用するように接続をマイグレーションする必要があります。

TCAM の DCB インターフェースによってバックレベルの CICS TOR に接続された端末のネットワークを使用している場合、これらの端末からのトランザクションは (前のリリースの CICS で可能であったように) CICS TS for z/OS バージョン 3.1 AOR にルーティングできます。ただし、ACF/VTAM を使用するように接続をマイグレーションすることをお勧めします。

共存

CICS TS for z/OS バージョン 3.1 は、CICS TS 3.1 より前の端末専有領域に接続された BTAM 端末からのトランザクション・ルーティングや機能シップをサポートしません。

CICS TS for z/OS バージョン 3.1 は、TCAM/ACB によって CICS TS 3.1 より前の端末専有領域に接続された端末からのトランザクション・ルーティングや機能シップをサポートしません。

CICS TS for z/OS バージョン 3.1 は、TCAM/DCB によって CICS TS 3.1 より前の端末専有領域に接続された端末からのトランザクション・ルーティングおよび機能シップをサポートします。

第 22 章 1 バイトのコンソール ID のサポートの中止

1 バイトのコンソール ID を使用した端末の定義は、サポートされなくなりました。TERMINAL リソース定義の CONSOLE 属性は使用されなくなりましたが、以前のリリースの CICS との互換性を確保するためサポートされています。

端末は、TERMINAL リソース定義の CONSNAME(*name*) 属性を使用して定義することができます。

CICSplex SM サポート

1 バイトのコンソール ID は使用されなくなり、これに対するサポートが提供されなくなりました。

CICSplex SM アプリケーション・プログラミング・インターフェースの変更点

TERMDEF リソース・テーブル

CICS Transaction Server for z/OS バージョン 3 リリース 1 またはそれ以降のリリースでは、TERMDEF リソース・テーブルの CONSOLE 属性が無効になります。

CICSplex SM エンド・ユーザー・インターフェース・ビューの変更点

以下の変更が加えられました。

TERMDEF ビュー

CICS Transaction Server for z/OS バージョン 3 リリース 1 またはそれ以降のリリースでは、TERMDEF ビューは未変更のままですが、以下の属性は無効になります。

CONSOLE

コンソール ID

CICSplex SM Web ユーザー・インターフェースの変更点

以下の変更が加えられました。

端末定義ビュー

CICS Transaction Server for z/OS バージョン 3 リリース 1 またはそれ以降のリリースでは、TERMDEF (EYUSTARTTERMDEF) ビューは未変更のままですが、以下の属性は無効になります。

CONSOLE

コンソール ID

メッセージ

BBMZA094E CAS (SSID) INVALID REPLY

#

1 バイトのコンソール ID の除去は、CICSplex SM メッセージ BBMZA094E が、
単にメッセージ BBMZA094E に応答したコンソールに対して出されるのではなく、
すべてのアクティブなコンソールに対して出されるようになったことを意味しま
す。メッセージのテキストに変更はありません。

第 23 章 CICS Connector for CICS TS の中止

CICS TS for z/OS バージョン 2.1 で導入された、CICS Connector for CICS TS に対するサポートは中止されました。

CICS コネクタは、Java クライアント・アプリケーションが CICS アプリケーションを起動できるようにするソフトウェア・コンポーネントです。CICS TS for z/OS バージョン 2.3 では、CICS Connector for CICS TS と同じ役割を果たす新しい CICS コネクタである CCI Connector for CICS TS が導入されました。CCI Connector for CICS TS は、CICS Transaction Server for z/OS で実行されている Java プログラムまたは Enterprise Bean が CICS サーバー・プログラムにリンクできるようにします。ただし、以前の CICS Connector for CICS TS が IBM 専有の Common Connector Framework (CCF) インターフェースをインプリメントしていたのに対して、新しい CCI Connector for CICS TS は、J2EE コネクタ・アーキテクチャ仕様バージョン 1.0 によって定義されている業界標準の Common Client Interface (CCI) をインプリメントしています。

CICS TS for z/OS バージョン 2.3 以降、以下が推奨されていました。

- 新規コネクタ・アプリケーションを作成するときは、CICS Connector for CICS TS ではなく CCI Connector for CICS TS を使用すること
- CICS Connector for CICS TS を使用する既存のアプリケーションを、CCI Connector for CICS TS を代わりに使用するようにマイグレーションすること

CICS TS for z/OS バージョン 3.1 では CICS Connector for CICS TS のランタイム・サポートが中止されたため、これらの推奨は必須になりました。

注: 前のリリースでは、ハイレベル CCF API を使用するか、CICS Transaction Gateway API の低レベルの CICS 固有外部呼び出しインターフェース (ECI) を使用するかのどちらかの方法で、CICS Connector for CICS TS をプログラミングすることが可能でした。CICS では、ECI 基本クラスが提供されなくなりました。

新規アプリケーションで CCI Connector for CICS TS を使用方法、および CICS Connector for CICS TS を使用する既存のアプリケーションを CCI Connector for CICS TS を代わりに使用するようにマイグレーションする方法については、「*Java Applications in CICS*」を参照してください。

第 24 章 Java プログラム・オブジェクトおよびホット・プーリングのランタイム・サポートの中止

Java プログラム・オブジェクトおよびホット・プーリング (HPJ) のランタイム・サポートは中止されました。

CICS TS 1.3 では、Java 仮想マシン (JVM) で Java プログラムを実行することに対する代替として、VisualAge[®] for Java、Enterprise Edition for OS/390 (ET/390) を使用して Java バイトコードを OS/390 実行可能ファイル (Java プログラム・オブジェクト) にバインドすることができました。Java プログラム・オブジェクトは OS/390 PDSE ライブラリーに保管され、言語環境プログラム実行単位 (エンクレーブ) で CICS によって実行されました。ET/390 の Java ランタイム・コンポーネントによって、CICS 領域におけるこのランタイム・サポートが提供されていました。

言語環境プログラム・エンクレーブは、それぞれの呼び出しごとに作成して初期化することができました。この場合、Java プログラム・オブジェクトは QR TCB の下で実行されました。代替として、パフォーマンスのオーバーヘッドを削減するために、事前初期設定された永続的なエンクレーブを、プログラムの複数の呼び出しで再利用できました。この機能は、ホット・プーリングとして知られていました。Java プログラム・オブジェクトのためにホット・プーリングが指定された場合、CICS では、z/OS 言語環境プログラムの PIP1 事前初期設定サービスを使用してエンクレーブを作成し、H8 モードのオープン・トランザクション環境 (OTE) タスク制御ブロック (TCB) の制御の下で CICS 領域の Java プログラム・オブジェクトを実行しました。

CICS TS for z/OS バージョン 2.3 では、既存の hpj でコンパイルされた Java プログラム・オブジェクトのためのランタイム・サポートは提供されていましたが、新規 Java プログラム・オブジェクトの開発や既存の Java プログラム・オブジェクトの変更のためのサポートは提供されていませんでした。CICS TS for z/OS バージョン 3.1 では、hpj でコンパイルされた Java プログラム・オブジェクトのランタイム・サポートは中止されました。

hpj でコンパイルされたすべての Java プログラム・オブジェクトは、Java 仮想マシン (JVM) で実行されるようにマイグレーションする必要があります。これについては、「CICS マイグレーション・ガイド」で説明しています。CICS JVM の詳細および CICS のための Java プログラミングの詳細については、「Java Applications in CICS」を参照してください。

CICS TS for z/OS バージョン 3.1 で Java プログラム・オブジェクトを実行しようとすると、ALIG 異常終了が発行されます。

ホット・プーリング Java プログラム・オブジェクトに使用されたオープン TCB モード H8 は、現在は存在しません。

CICS 外部インターフェースの変更点

システム初期設定パラメーターの変更点

システム初期設定パラメーター MAXHPTCBS が除去されました。MAXHPTCBS は、オープン TCB モード H8 を制御していました。

リソース定義の変更点

- PROGRAM リソース定義から HOTPOOL 属性が除去されました。この属性は、H8 TCB の制御の下で、プログラムの複数の呼び出しで再利用される、あらかじめ初期設定された言語環境プログラム・エンクレープで Java プログラム・オブジェクトを実行するかどうかを指定するために使用されていました。
- サンプル・アプリケーション・プログラム・グループ DFH\$JAVA が除去されました。このグループには、VisualAge for Java、Enterprise Edition for OS/390 を使用する場合に、Java サポート用のサンプル・アプリケーションに必要なリソース定義が含まれていました。同じサンプル・アプリケーションが、JVM での使用のために DFH\$JVM グループで定義されています。
- DFHMCT TYPE=RECORD マクロから DFHTASK フィールド 278 (CICS MAXHPTCBS 遅延時間) が除去されました。

アプリケーション・プログラミング・インターフェースの変更点

EXEC CICS LINK および XCTL コマンドと、BTS コマンド LINK ACQPROCESS および LINK ACTIVITY に対する INVREQ 応答の修飾に使用されていた RESP2 値 43 が除去されました。

- 43** リンク・スタック上にホット・プーリングされたプログラムが既に存在するときに、ホット・プーリングされた Java プログラム・オブジェクトに対して LINK (または XCTL) が試行されました。

システム・プログラミング・インターフェースの変更点

- EXEC CICS CREATE PROGRAM コマンドから HOTPOOL オプションが除去されました。
- EXEC CICS INQUIRE PROGRAM コマンドから HOTPOOLING オプションが除去されました。
- EXEC CICS SET PROGRAM コマンドから HOTPOOLING オプションが除去されました。
- EXEC CICS INQUIRE DISPATCHER コマンドから ACTHPTCBS および MAXHPTCBS オプションが除去されました。これらのオプションは、現在ユーザー・タスクに割り振られている H8 モードのオープン TCB の数、および CICS が接続および維持できる H8 モードのオープン TCB の最大数を照会する目的で使用されていました。
- EXEC CICS SET DISPATCHER コマンドから MAXHPTCBS オプションが除去されました。
- CVDAs HOTPOOL (1065) および NOTHOTPOOL (1066) が削除されました。

CEMT の変更点

- CEMT INQUIRE PROGRAM 表示から HOTPOOLING フィールドが除去されました。
- CEMT SET PROGRAM コマンドから HOTPOOL および NOTHOTPOOL オプションが除去されました。
- CEMT INQUIRE DISPATCHER 表示から ACTHPTCBS および MAXHPTCBS フィールドが除去されました。
- CEMT SET DISPATCHER コマンドから MAXHPTCBS オプションが除去されました。

グローバル・ユーザー出口ルーチンの変更点

DFHUEPAR 標準パラメーター・リストに含まれ、UEPGIND によってアドレッシングされる、グローバル・ユーザー出口のタスク・インディケーター・フィールドには、シンボル値 UEPTH8 は格納されなくなりました。UEPTH8 は、オープン TCB モード H8 を表現していました。

出口プログラミング・インターフェース (XPI) の変更点

DFHPGISX 呼び出し INQUIRE PROGRAM および SET PROGRAM から、HOTPOOL オプションが除去されました。

ユーザーが置換可能なプログラムの変更点

ユーザーが置換可能なプログラム DFHAPH80 と DFHJHPAT が除去されました。

- DFHAPH80 は、Java プログラム・オブジェクトを実行する言語環境プログラム・エンクレープのためのデフォルトの言語環境プログラム・ランタイム・オプションをユーザーが変更できるようにするために提供されていました。
- DFHJHPAT はオプションであり、トレースなどのユーザー独自の目的で使用することができました。これは、Java プログラム・オブジェクトが起動される前に呼び出されました。

モニターの変更点

グループ DFHTASK のモニター・データ・フィールド 278 が除去されました。ホット・プーリング Java プログラム・オブジェクトに使用され、MAXHPTCBS によって制御されたオープン TCB モード H8 は、現在は存在しません。

統計の変更点

TCB モード統計で、TCB モード H8 が表示されなくなりました。TCB モード統計は、DFHDSGDS DSECT 内で DSGTCBM DSECT によってマップされます。

問題判別の変更点

- Java ホット・プーリングのためのトレース・エントリ (AP 19A0 から AP 19C4) が除去されました。

- メッセージ DFHAP1219 から DFHAP1225 が除去されました。
- 異常終了 AJH0 から AJHF が除去されました。
- リソース・タイプ HP_POOL は、ディスパッチャー待ちの原因にならなくなりました。

CICS TS for z/OS バージョン 3.1 で Java プログラム・オブジェクトを実行しようとする、ALIG 異常終了が発行されます。

CICSplex SM サポート

Java プログラム・オブジェクトおよびホット・プーリング (HPJ) のランタイム・サポートが除去された結果として、CICSplex SM インターフェースに対していくつかの変更が行われました。

CICSplex SM エンド・ユーザー・インターフェース・ビューの変更点

CICS Transaction Server for z/OS バージョン 3 リリース 1 には、新しく追加されたエンド・ユーザー・インターフェース・ビューはありませんが、以下の既存の画面に対して変更が行われました。

- PROGDEF: 属性 HOTPOOLING は無効になりましたが、依然として表示されま
- す。
- PROGRAM: 属性 HOTPOOLING が除去されました。

CICSplex SM アプリケーション・プログラミング・インターフェースの変更点

Java プログラム・オブジェクトおよびホット・プーリング (HPJ) のランタイム・サポートが除去された結果として、以下のような変更が行われました。

- 『CICSRGN リソース・テーブル』
- 『PROGRAM リソース・テーブル』
- 『PROGDEF リソース・テーブル』

CICSRGN リソース・テーブル

既存の **HOTPOOLING** 統計は、CICS TS 3.1 では無効になりました。

PROGRAM リソース・テーブル

SPI 属性 **HOTPOOLING** は、CICS TS 3.1 では無効になりました。

PROGDEF リソース・テーブル

属性 **HOTPOOLING** は、CICS TS 3.1 では無効になりました。

CICSplex SM Web ユーザー・インターフェースの変更点

以下のビューに対して変更が加えられました。

- 369 ページの『プログラム定義ビュー』

- 『プログラム・ビュー』

プログラム定義ビュー

属性 HOTPOOLING は、CICS TS for z/OS V3.1 では無効になりましたが、PROGDEF (EYUSTARTPROGDEF.DETAILED) ビューは以前と同じままです。

プログラム・ビュー

属性 HOTPOOLING は、CICS TS for z/OS V3.1 では無効になりましたが、PROGRAM (EYUSTARTPROGRAM.DETAILED) ビューは以前と同じままです。

第 25 章 Windows リモート MAS のための CICSplex SM サポートの中止

前のリリースの CICSplex SM では、リモート管理対象アプリケーション・システム (RMAS) の管理において TXSeries® バージョン 4.3.0.4 および TXSeries バージョン 5.0 (NT 4.3 および NT 5.0 ともいう) の CICS for Windows コンポーネントをサポートしていました。このサポートは不要になり、CICS Transaction Server for z/OS バージョン 3.1 以降のリリースでは CICSplex SM TXSeries エージェントが除去されました。したがって、Windows 用の CICSplex SM リモート MAS エージェントをセットアップすることはできなくなりました。

希望される場合、お客様は CICSplex SM での TXSeries サポートのために CICS Transaction Server 2.3 または 2.2 を引き続き使用できます。

第 26 章 CICS Transaction Affinities ユーティリティーの提供中止

CICS TS for z/OS バージョン 3.1 では、CICS Transaction Affinities ユーティリティーが提供されなくなりました。このユーティリティーの、トランザクションの親和性を検出および報告する機能は、より高度なツールである CICS Interdependency Analyzer によって提供されるようになりました。

CICS Interdependency Analyzer では、以下の作業を行うことが可能です。

- 個々の CICS トランザクションが使用するリソースの集合と、そのリソースと他のリソースとの関係を識別する。これにより、アプリケーション・セットの構成を理解することができます。すなわち、CICS 領域に何が含まれているか、実行のためにトランザクションにどのリソースが必要か、どのプログラムがどのリソースを使用するか、どのリソースが使用されなくなっているかを確認することができます。これにより、アプリケーションを保守し、機能を強化し、修正し、再配布する能力が大幅に向上します。
- 考えられるトランザクションの親和性を識別する。親和性を識別するには、特定のトランザクション・グループを同じ CICS 領域または特定の領域で実行する必要があります。トランザクションの親和性を識別する能力は、動的ルーティング環境において役に立ちます。特定のトランザクションが特定のアプリケーション専有領域 (AOR) にルーティングされないようにしたり、特定のトランザクションが特定の AOR にルーティングされるように要求したりするすべての制約事項を知っておく必要があります。

CICS Interdependency Analyzer の詳細については、「*CICS Interdependency Analyzer User's Guide and Reference*」を参照してください。

第 7 部 一般情報

第 27 章 CICS 稼働環境

このトピックでは、CICS Transaction Server for z/OS の CICS および CICSplex SM エlementを使用するために必要な関連製品についての情報をいくつか示します。

ハードウェア要件

プロセッサ

基本要件は、前提条件のオペレーティング・システムをサポートし、z/OS V1.4、CICS TS for z/OS バージョン 3.1、アプリケーション・プログラム、アクセス方式、および稼働するその他のすべてのソフトウェアの要件を満たすのに十分なストレージを備えているプロセッサに関するものです。これには、IBM eServer™ zSeries 990 が含まれます。

Parallel Sysplex® (並列シスプレックス) のサポート

並列シスプレックス環境は、CICS がサポートするデータ共有機能のそれぞれについて必要であり、MVS システム・ロガーのログ・ストリーム・マージ機能でも必要です。要件は以下のとおりです。

- 関連カップリング・リンクがインストールされた、1 つ以上のカップリング・ファシリティ
- 共通の外部時間ソースを提供する IBM シスプレックス・タイマー
- シスプレックス内の中央演算処理装置複合システム (CPC) の数をサポートするのに十分な DASD パス。DASD パスは、シスプレックスの各 CPC に 1 つずつ確保できるだけのパスが DASD コントローラーによって提供されます。あるいは ESCON® ディレクターによって提供することもできます。

CICS のデータ共有のサポートを使用すれば、IMS データベース、DB2 データベース、VSAM データ・セット、CICS 一時記憶域、カップリング・ファシリティ・データ・テーブル、および名前付きカウンター内のデータにアクセスできます。

暗号ハードウェア

以下の場合、zSeries 暗号ハードウェアが必要です。

- WS-Security 機能を活用する場合
- SSL 暗号化のパフォーマンス向上の利点を完全に生かす場合

どちらの機能も z/OS Integrated Cryptographic Services Facility (ICSF) に依存しています。

カタカナ端末装置

CICS は特定のメッセージを大/小文字混合で発行する必要があるため、コード・ページ 930 の非拡張 1 バイト文字セット (SBCS) カタカナ部分に制限されているディスプレイまたは端末エミュレーターでは、この製品はサポートされません。

ソフトウェア要件

「*Program Directory*」(GI10-6427)には通常は、ソフトウェア要件に関する最新の情報が含まれます。

稼働環境

CICS TS for z/OS バージョン 3.1 には、z/OS V1.4 以降が必要です。これよりも低いレベルのオペレーティング・システムがインストールされた環境では、初期化されません。

STEPLIB 連結または LNKLIST に組み込むことによって、CICS 初期化時に言語環境プログラム・ライブラリー SCEERUN が CICS で使用できなければなりません。言語環境プログラム・サービスは、いくつもの CICS 機能で使用されます。

Java アプリケーション・プログラムまたは Enterprise Bean の場合、IBM SDK for z/OS、Java 2 Technology Edition (永続的な再使用可能 JVM テクノロジー)、Version 1.4、プログラム 5655-I56 が必要です。これは V1.4.2 レベルでなければなりません。

- IBM SDK for z/OS、Java 2 Technology Edition、Version 1.4 は、テープまたは <http://www.s390.ibm.com/java/> からのダウンロードによって、無料で入手できます。
- IBM 64-Bit SDK For z/OS、Java2 Technology Edition、Version 1.4、プログラム番号 5655-M30 では、必要な機能が提供されません。

WS-Security をサポートするには、IBM XML Toolkit for z/OS V1.7 が必要です。これは無料の製品であり、プログラム番号は 5655-J51 です。

Enterprise Bean を配置するには、WebSphere Application Server V5.0 以降が必要です。

- 使用されるコンポーネントは、Application Server Toolkit (ASTK) for Windows です。Application Assembly Tool (AAT) は、初期の V5.0 デリバリーでは提供されていましたが、現在はサポートされていません。
- ASTK は WebSphere Studio Enterprise Developer V5.1 にも含まれています。

Enterprise Bean の JNDI サポートは、SecureWay[®] Security Server で提供され、基本 z/OS オペレーティング・システムの一部としてライセンス交付を受けた LDAP サーバーによって提供されます。CICS TS V3.1 は、WebSphere Application Server (任意のプラットフォーム) V5 および V6 と相互稼働します。これは、インターオペラビリティのために RMI/IIOP を使用する場合には直接適用され、JCA を使用する場合は CICS Transaction Gateway V5.0 以降を経由して適用されます。

CICS TS V3.1 とともに使用する Java プログラム (Enterprise Bean を含む) の開発用に、WebSphere Studio ファミリー V5 のメンバー、および Rational[®] Application Developer V6 がサポートされます。

128 ビット暗号化かそれ以上の暗号スイートを使用する場合は、z/OS の System SSL セキュリティー・レベル 3 機能が必要です。

その他のサポート製品

CICS TS for z/OS バージョン 3.1 との使用がサポートされるその他の製品のレベルは、以下のとおりです。

- IMS Database Manager V7 (5655-B01)
- IMS Database Manager V8 (5655-C56)
- IMS Database Manager V9 (5655-J38)
- DB2 Universal Database™ Server for OS/390 V6.1 (5645-DB2)
 - SQLJ/JDBC サポートの場合は、APAR PQ84783 の PTF が必要です。
 - DB2 グループ接続はサポートされません。
- DB2 Universal Database Server for OS/390 V7.1 (5675-DB2)
 - SQLJ/JDBC サポートの場合は、APAR PQ84783 および 86525 の PTF が必要です。
 - DB2 グループ接続には、APAR PQ44614、PQ45691、および PQ45692 が必要です。
- DB2 Universal Database for z/OS V8.1 (5625-DB2)
 - SQLJ/JDBC サポートの場合は、APAR PQ84783 および 86525 の PTF が必要です。
- WebSphere MQ for z/OS V5.3 (5655-F10)
- Tivoli® Decision Support for OS/390 (5698-ID9) V1.6 (必要なサービスは適用済み)
- Tivoli Business Systems Manager V3.1
- CICS Universal Client Version 5.0 以降
- CICS Transaction Gateway Version 5.0 以降

Information Center 環境

Information Center は、サーバーとして、次のプラットフォームでサポートされます。

- Windows 2000 および Windows XP
- AIX® V5.1 以降
- Intel™ 上の Linux™ (RedHat および Suse)

Information Center をブラウズするには、HTML 4.0 および Document Object Model (DOM) 標準をサポートするブラウザが必要です。適切なブラウザは、以下のとおりです。

Microsoft® Internet Explorer Version 6.0

Netscape Navigator V7.0

Mozilla V1.0

これらは Windows 2000 または Windows XP で稼働します。

Information Center で入手した PDF ファイルを読み取るには、Adobe Acrobat Reader 5.0 または 6.0 が必要です。このファイルは、Acrobat 6.0 (PDF 1.5) レベルの Adobe Acrobat Distiller 6.0 を使用して生成されています。これらは、Adobe Acrobat Reader 5.0 を使用して読み取ることができますが、Distiller 6.0 のアクセシ

ビリティ機能が必要な場合は Reader 6.0 が必要です。

CICS ツールおよび関連製品のサポート

CICS Interdependency Analyzer

CICS TS for z/OS バージョン 3.1 では、以下のバージョンを使用できます。

CICS Interdependency Analyzer for z/OS V1.3 (APAR PQ95065 の PTF 適用済み)。

CICS TS for z/OS バージョン 3.1 では、以下のバージョンは稼働しません。

CICS Interdependency Analyzer for z/OS V1.1。

CICS Interdependency Analyzer for z/OS V1.2。

CICS Performance Analyzer

CICS Performance Analyzer for z/OS V1.3 では、CICS TS for z/OS バージョン 3.1 からの SMF 110 データはサポートされません。

CICS Performance Monitor

CICS Performance Monitor for z/OS V1.1 では、CICS TS for z/OS バージョン 3.1 はサポートされません。

CICS Performance Monitor for z/OS V1.2 (サービス適用済み) は、CICS TS for z/OS バージョン 3.1 の Tolerant Support を CICS TS for z/OS バージョン 2.3 レベルで提供します。

Candle® OMEGAMON®

以下のバージョンは CICS TS for z/OS バージョン 3.1 をサポートします。ただし、このサポートには CICS TS for z/OS バージョン 3.1 の新機能の利用は含まれません。

Candle OMEGAMON II® for CICS V520。

Candle OMEGAMON XE for CICS V100。

Candle OMEGAMON XE for CICSplex V220 (サービス適用済み)。

CICS VSAM Recovery

以下のバージョンは、CICS TS for z/OS バージョン 3.1 で処理された VSAM ファイルのリカバリー・サポートを提供します。

CICS VSAM Recovery V3.2。

CICS VSAM Recovery V3.3。

CICS Business Event Publisher

CICS TS for z/OS バージョン 3.1 では、CICS Business Event Publisher for MQSeries® Version 1.2 (サービス適用済み) を使用できます。

CICS Online Transmission Time Optimizer

CICS TS for z/OS バージョン 3.1 では、CICS Online Transmission Time Optimizer for z/OS V1.1 を使用できます。

Session Manager

CICS TS for z/OS バージョン 3.1 では、以下のバージョンを使用できます。

Session Manager for z/OS V1.1。

Session Manager for z/OS V1.2。

CICS VSAM Transparency

CICS TS for z/OS バージョン 3.1 では、CICS VSAM Transparency for z/OS V1.1 を使用できます。

CICS VSAM Copy

CICS TS for z/OS バージョン 3.1 では、CICS VSAM Copy for z/OS V1.1 (サービス適用済み) を使用できます。

CICS Batch Application Control

CICS TS for z/OS バージョン 3.1 では、CICS Batch Application Control for z/OS V1.1 (サービス適用済み) を使用できます。

MQSeries Integrator Agent

CICS TS for z/OS バージョン 3.1 では、MQSeries Integrator Agent for CICS Transaction Server V1.1 は稼働しません。

Fault Analyzer

CICS TS for z/OS バージョン 3.1 では、以下のバージョンを使用できます。

Fault Analyzer for z/OS および OS/390 V3.1 (APAR PQ74048 の PTF UQ77156 適用済み)。

Fault Analyzer for z/OS V4.1。

Fault Analyzer for z/OS V5.1。

Debug Tool

CICS TS for z/OS バージョン 3.1 では、以下のバージョンを使用できます。

Debug Tool for z/OS および OS/390 V3.1 (APAR PQ73643 の PTF UQ77541 適用済み)。

Debug Tool for z/OS V4.1。

Debug Tool for z/OS V5.1。

互換性

z/OS 変換サービス

CICS Transaction Server の以前のレベルとは異なり、CICS TS V3 では、z/OS サービスを使用して、前の CICS TS リリースでサポートされていた以上の変換を実行できます。その一例は、Web サービスのサポートに必要となる可能性がある Unicode への変換および Unicode からの変換です。この機能を使用するには、z/OS に初期変換イメージがインストールされている必要があり、これはシステム IPL 上でのみ実行できます。z/OS を再 IPL せずに CICS TS V3 をインストールすること

が要求される場合は、前のシステム IPL 時に初期変換イメージがインストールされていれば、この機能を実行できます。1 変換イメージには、CICS TS からのコードは含まれません。また変換イメージは、さらに IPL を実行しなくても最新表示できます。

CICS の JVM モード

CICS TS V3.1 で Java プログラムを使用するお客様には、連続モードの使用をお勧めします。連続モードのサポートは CICS TS V2.3 で導入されました。CICS での Java 使用を標準手法に沿ったものにするため、将来の CICS TS リリースでは再設定可能モードのサポートは除去されます。

SOAP for CICS 機能

SOAP for CICS 機能は、CICS TS V2.2 および V2.3 と一緒にオーダーできましたが、CICS TS V3.1 と一緒にはオーダーできません。ただし、既にこの機能をお持ちのお客様のマイグレーションを支援するため、CICS TS V3.1 でもこの機能 1 がサポートされており、使用できます。また、アプリケーションは引き続き稼働します。それでも、お客様には CICS TS V3.1 の Web サービス・サポート機能にマイグレーションすることをお勧めします。

Common Connector Framework (CCF)

Common Connector Framework (CCF) は、Common Client Interface (CCI) へのインターフェースとしてこれまで使用されていましたが、CICS TS V3.1 ではサポートされません。このサポートが除去される予定であることは、CICS TS V2.3 の発表で示されました。

ECI 基本クラス (ECIREQUEST)

ECI 基本クラス (ECIREQUEST。CICS Transaction Gateway との互換性を持たせるために導入された) は CICS TS V3.1 には含まれません。COMMON CLIENT INTERFACE CONNECTOR FOR CICS TS (CCI Connector for CICS TS) に置き換えることをお勧めします。これは、ECIREQUEST が除去されることが発表された CICS TS V2.3 で導入されました。

Transaction Affinities ユーティリティ

これまで CICS Transaction Affinities ユーティリティの一部として提供されていた検出機能および報告機能のコンポーネントは、CICS TS V3.1 には組み込まれていません。これらのコンポーネントは現在は、相互依存性と親和性の両方の分析機能を持つ IBM CICS Interdependency Analyzer for z/OS V1.3 (2004 年 8 月発表) に組み込まれています。CICS Transaction Affinities ユーティリティのロード・ライブラリー・スキャナー・コンポーネントは CICS TS V3.1 に残されているので、親和性を持つ可能性があるアプリケーション・プログラムに関するレポートを作成できます。

第 28 章 スレッド・セーフ・アプリケーション・プログラミング・インターフェース・コマンド

CICS Transaction Server for z/OS バージョン 3 リリース 1 におけるほとんどの新規コマンドは、スレッド・セーフです。さらに、一部の既存コマンドは、このリリースでスレッド・セーフになりました。

スレッド・セーフの新規コマンド

CONVERTTIME
DELETE CONTAINER (CHANNEL)
GET CONTAINER (CHANNEL)
INVOKE WEBSERVICE
MOVE CONTAINER
PUT CONTAINER (CHANNEL)
SOAPFAULT ADD
SOAPFAULT CREATE
SOAPFAULT DELETE
WEB CONVERSE
WEB CLOSE
WEB OPEN
WEB PARSE URL
WEB RECEIVE (Client)
WEB SEND (Client)

スレッド・セーフになった既存のコマンド

WEB ENDBROWSE FORMFIELD
WEB ENDBROWSE HTTPHEADER
WEB EXTRACT
WEB READ FORMFIELD
WEB READ HTTPHEADER
WEB READNEXT FORMFIELD
WEB READNEXT HTTPHEADER
WEB RECEIVE (サーバー)
WEB RETRIEVE
WEB SEND (サーバー)
WEB STARTBROWSE FORMFIELD
WEB STARTBROWSE HTTPHEADER
WEB WRITE HTTPHEADER

スレッド・セーフでない新規コマンド

START CHANNEL

第 29 章 高水準言語サポート

このリファレンス・トピックでは、CICS でサポートされる高水準プログラム言語について説明します。また、CICS の現行リリースでサポートされる各言語のリリースに関する情報を示します。

COBOL コンパイラー

コンパイラー	プログラム番号	コンパイラーのサービス	CICS 変換プログラム・サポート	CICS ランタイム・サポート	IBM 分散デバッガーの使用 (注 1 (386 ページ) 参照)	WebSphere Studio Enterprise Developer での使用
OS/VS COBOL	5740-CB1 5734-CB4 5740-LM1	なし	CICS TS V1.3: サポートされる CICS TS V2.2, CICS TS V2.3, CICS TS V3.1: サポートされない	CICS TS V1.3, CICS TS V2.2, CICS TS V2.3: z/OS の言語環境プログラム・コンポーネントが必要。アプリケーションは未変更のまま稼働します。 CICS TS V3.1: サポートされない	なし	なし
VS COBOL II	5668-022 5668-023 5668-958	なし	CICS TS V1.3: COBOL2 オプションでサポートされる CICS TS V2.2, CICS TS V2.3, CICS TS V3.1: COBOL3 オプションでのみサポートされる	CICS TS V1.3, CICS TS V2.2: z/OS の言語環境プログラム・コンポーネントが必要。アプリケーションは未変更のまま稼働します。 CICSTS V2.3, CICS TS V3.1: z/OS の言語環境プログラム・コンポーネントが必要。CICS は言語環境プログラム・ランタイムを排他的に使用します。アプリケーションの振る舞いは変更される場合があります (注 3 (386 ページ) 参照)。	あり (制限付き)	なし
SAA AD/Cycle COBOL/370	5688-197 5668-958	なし	CICS TS V1.3: COBOL2 オプションでサポートされる CICS TS V2.2, CICS TS V2.3, CICS TS V3.1: COBOL3 オプションでのみサポートされる	言語環境プログラム	あり (制限付き)	なし

コンパイラー	プログラム番号	コンパイラーのサービス	CICS 変換プログラム・サポート	CICS ランタイム・サポート	IBM 分散デバッガーの使用 (注 1 参照)	WebSphere Studio Enterprise Developer での使用
COBOL for MVS および VM	5688-197	なし	CICS TS V1.3: COBOL2 オプションでサポートされる CICS TS V2.2. CICS TS V2.3. CICS TS V3.1: COBOL3 オプションでのみサポートされる	言語環境プログラム	あり (制限付き)	なし
COBOL for OS/390 および VM V2	5648-A25	あり	CICS TS V1.3: COBOL2 オプションでサポートされる CICS TS V2.2. CICS TS V2.3. CICS TS V3.1: COBOL3 オプションでのみサポートされる	言語環境プログラム	あり (制限付き)	なし
COBOL for OS/390 および VM V2	5648-A25 (APAR PQ45462 の PTF 適用)	あり	統合変換プログラムを使用できる (注 2 参照)	言語環境プログラム	あり (制限付き)	あり (制限付き)
Enterprise COBOL for z/OS および OS/390 V3	5655-G53	あり	統合変換プログラムを使用できる (注 2 参照)	言語環境プログラム	あり	あり

注:

1. IBM 分散デバッガーは、WebSphere Studio Enterprise Developer V5、およびその他の IBM 製品のコンポーネントとして入手できます。

詳しくは、<http://www.ibm.com/software/awdtools/debugger/>を参照してください。

2. 統合変換プログラム機能には、IBM COBOL for OS/390 3 および VM バージョン 2 リリース 2 (APAR PQ45462 の PTF 適用) か、Enterprise COBOL for z/OS および OS/390 バージョン 3 が必要です。ただし、COBOL3 変換プログラム・オプションがアクティブでなければなりません。
3. 「*Language Environment within CICS TS: Questions and Answers*」という資料を参照してください。この資料は <http://www-3.ibm.com/software/htp/cics/library/indexes/whitepapers.html> で入手できます。

PL/I コンパイラー

コンパイラー	プログラム番号	コンパイラーのサービス	CICS 変換プログラム・サポート	CICS ランタイム・サポート	IBM 分散デバッガーの使用 (注 1 参照)	WebSphere Studio Enterprise Developer での使用
OS PL/I Optimizing Compiler V1	5724-PLI	なし	あり	CICS TS V1.3. CICS TS V2.2. CICS TS V2.3: z/OS の言語環境プログラム・コンポーネントが必要。アプリケーションは未変更のまま稼働します。 CICS TS V3.1: サポートされない。	あり (制限付き)	なし
OS PL/I Optimizing Compiler V2	5668-909 5668-910 5668-911	なし	あり	CICS TS V1.3. CICS TS V2.2. CICS TS V2.3: z/OS の言語環境プログラム・コンポーネントが必要。アプリケーションは未変更のまま稼働します。 CICS TS V3.1: サポートされない。	あり (制限付き)	なし
SAA AD/Cycle PL/I for MVS および VM	5688-235	なし	あり (注 2 (388 ページ) 参照)	言語環境プログラム	あり (制限付き)	なし
PL/I for MVS および VM V1	5688-235	なし	あり (注 2 (388 ページ) 参照)	言語環境プログラム	あり (制限付き)	なし
VisualAge PL/I for OS/390 V2	5655-B22	なし	あり (注 2 (388 ページ) 参照)	言語環境プログラム	あり (制限付き)	なし
Enterprise PL/I for z/OS および OS/390 V3	5655-H31	あり	統合変換プログラムを使用できる (注 2 (388 ページ) 参照)	言語環境プログラム	あり	あり

注:

1. IBM 分散デバッガーは、WebSphere Studio Enterprise Developer V5、およびその他の IBM 製品のコンポーネントとして入手できます。

詳しくは、<http://www.ibm.com/software/awdtools/debugger/>を参照してください。

2. 統合変換プログラム機能には、VisualAge PL/I for 3 OS/390 バージョン 2 リリース 2.1 (APAR PQ45562 の PTF 適用) か、Enterprise 3 PL/I for z/OS および OS/390 バージョン 3 が必要です。

C および C++ コンパイラー

コンパイラー	プログラム番号	コンパイラーのサービス	CICS 変換プログラム・サポート	CICS ランタイム・サポート	IBM 分散デバッガーの使用 (注 1 参照)	WebSphere Studio Enterprise Developer での使用
C/370 V1	5688-040	なし	あり	CICS TS V1.3. CICS TS V2.2. CICSTS V2.3: z/OS の言語環境プログラム・コンポーネントが必要。アプリケーションは未変更のまま稼働します。 CICS TS V3.1: サポートされない。	あり (制限付き)	なし
C/370 V2	5688-187 5688-188	なし	あり	CICS TS V1.3. CICS TS V2.2. CICSTS V2.3: z/OS の言語環境プログラム・コンポーネントが必要。アプリケーションは未変更のまま稼働します。 CICS TS V3.1: サポートされない。	あり (制限付き)	なし
SAA AD/Cycle C/370	5688-216	なし	あり	言語環境プログラム	あり (制限付き)	なし
C/C++ for MVS/ESA	5655-121	なし	あり	言語環境プログラム	あり (制限付き)	なし
C/C++ for OS/390	5647-A01 のコンポーネント	あり	あり	言語環境プログラム	あり (制限付き)	なし
C/C++ for z/OS および OS/390	5694-A01 のコンポーネント	あり	あり	言語環境プログラム	あり	なし

注:

1. IBM 分散デバッガーは、WebSphere Studio Enterprise Developer V5、およびその他の IBM 製品のコンポーネントとして入手できます。

詳しくは、<http://www.ibm.com/software/awdtools/debugger/>を参照してください。

Java サポート

コンパイラー または JVM	プログラム番号	サービス	CICS 変換プログラム・サポート	CICS ランタイム・サポート	IBM 分散デバッガーの使用 (注 1 (390 ページ) 参照)	WebSphere Studio Enterprise Developer での使用
VisualAge for Java, Enterprise Edition V2 - Enterprise Toolkit for OS/390 (注 2 (390 ページ) 参照)	5655-JAV	なし	CICS TS V1.3. CICS TS V2.2. CICS TS V2.3. CICS TS V3.1: 変換プログラムは不要 (JCICS クラスを使用する)。	言語環境プログラム	あり (制限付き)	なし
Java for OS/390, SDK 1.1.8		あり	CICS TS V1.3 のみ: 変換プログラムは不要 (JCICS クラスを使用する)。 CICS TS V2.2. CICS TS V2.3. CICS TS V3.1: サポートされない	言語環境プログラム	あり	あり
Developer Kit for OS/390, Java 2 Technology Edition, V1.3.1	5655-D35	あり	CICS TS V2.2 のみ: 変換プログラムは不要 (JCICS クラスを使用する)。 CICS TS V1.3. CICS TS V2.3. CICS TS V3.1: サポートされない	言語環境プログラム	あり	あり
SDK for z/OS, Java 2 Technology Edition, V1.4	5655-I56	あり	CICS TS V2.3. CICS TS V3.1: 変換プログラムは不要 (JCICS クラスを使用する)。 CICS TS V1.3. CICS TS V2.2: サポートされない。変換プログラムは不要 (JCICS クラスを使用する)。 CICS TS V2.3 でのみサポートされる	言語環境プログラム	あり	あり

注:

1. IBM 分散デバッガーは、WebSphere Studio Enterprise Developer V5、およびその他の IBM 製品のコンポーネントとして入手できます。

詳しくは、<http://www.ibm.com/software/awdtools/debugger/>を参照してください。

2. Java プログラム・オブジェクトは、VisualAge for Java Enterprise Toolkit for OS/390 (ET/390) バイトコード・バインダーを使ってコンパイルされたプログラムです (これらは hpj コマンドでコンパイルされるため、コンパイル済み Java プログラム、または HPJ プログラムと呼ばれることもあります)。マイグレーションを支援するため、Java プログラム・オブジェクトのサポートは CICS TS V2.3 でも継続されます。ただし、CICS Transaction Server for z/OS でこのサポートが提供されるのは、このリリースが最後です。

Java プログラム・オブジェクトを CICS TS V2.3 にマイグレーションするお客様には、CICS の JVM モードでこれらを再ビルドして実行することをお勧めします。

第 8 部 資料

第 30 章 Eclipse インフォメーション・センター

CICS Transaction Server for z/OS バージョン 3 リリース 1 では、CICS Information Center は WebSphere Studio WorkBench User Assistance システム (インフォメーション・センターを構成する複数の文書プラグインを含む Eclipse フレームワークの 1 つ) 内で実行されます。インフォメーション・センターの新しいルック・アンド・フィール、特に、ウェルカム・ページは、他の製品のインフォメーション・センターと整合性を持つようになりました。

CICS TS 3.1 の文書は、1 つのプラグインに含まれています。CICS のツール製品の一部またはすべてをインストールするオプションもあります。ツール製品 1 つにつき 1 つのプラグインがあります。ヘルプ・システムとしても知られているインフォメーション・センターは、2 つのモードで稼働します。ワークステーションでローカルで開始することも、ブラウザを介したりリモート・アクセスによって、サーバーとして開始することもできます。

ウェルカム・ページでは、CICS の最新機能に関する情報にアクセスしたり、製品の特長や、Web 上の CICS リソースへのリンクを見つけることができます。

ナビゲーションから、機能領域によって、あるいはユーザーが実行中の作業に対応するゴールを探すことによって、CICS に関する情報にアクセスすることができます。「ライブラリーおよび PDF (Library and PDFs)」を選択して、文書を HTML および PDF 形式で表示することもできます。

このリリースのインフォメーション・センターには、以下の機能がインプリメントされました。

- Linux、Linux on zSeries および Linux on z/OS に対するプラットフォーム・サポートの拡張
- 新機能セクション
- 情報ロードマップ
- ラーニング・パス
- 検索およびルックアップ
- トラブルシューティングおよびサポート
- ブックマーク
- ユーザー設定
- ナビゲーションの変更

Eclipse インフォメーション・センターの利点

Eclipse フレームワークへの移動が、他のプラットフォームがサポートされるというメリットをもたらしました。インフォメーション・センターを Linux RedHat または Linux SuSE ワークステーションでローカルで稼働できるようになりました。また、インフォメーション・センターが、Linux RedHat for zSeries、Linux SuSE for zSeries および z/OS 上のサーバーで稼働できるようになりました。これにより、イ

インフォメーション・センターを動作させる場所に対する柔軟性が向上しました。サポートされるプラットフォームの詳細なリストについては、395 ページの『要件』を参照してください。

インフォメーション・センターのインストールが非常に単純化されたため、GUI のインストールがなくなりました。インフォメーション・センターに付属の `readme` で、Eclipse ヘルプ・システムのインストール方法を説明しています。Eclipse の構造は、複数の文書プラグイン (他の製品のプラグインを含む) を 1 つのヘルプ・システムに入れることができることを意味します。これにより、インフォメーション・センターに含まれている情報をカスタマイズすることができます。また、CICS 文書プラグインを Eclipse ベースの IDE に組み込むこともできます。

Eclipse では Lucene 検索エンジンを使用するため、CICS 文書の検索性が改善されました。NetQuestion は不要になりました。インフォメーション・センターで検索した場合、HTML コンテンツ内でストリングの照会のそれぞれの検索結果が強調表示されるようになりました。

インフォメーション・センターに、ナビゲーションに関するいくつかの改善が行われました。ナビゲーションには、ラーニング・パスと情報ロードマップの 2 つの新機能があります。ラーニング・パスは、新機能についてより速く学習するために役立つ一連のトピックです。情報ロードマップは、CICS の機能領域についての広範なリンクの集合を含む 1 つのトピックです。情報ロードマップは、インフォメーション・センター内と Web 上の両方にある役に立つリソースの検索をしやすくする設計になっており、また、リンクが示す場所についての説明も提供しています。

ナビゲーションには、「トラブルシューティングおよびサポート (*Troubleshooting and Support*)」のセクションもあります。このセクションでは、知識ベースの検索方法、IBM からのフィックスの取得方法、および問題の報告方法についてのヘルプ情報を提供しています。特に役に立つのは、Web 検索のトピックで、これを参照すると、CICS サポートの Web サイトを照会して、問題の解決方法を探すことができます。このトピックには、Google の検索機能も含まれています。また、問題をすばやく解決するために、多数のリリースからの重要な技術情報も提供されます。

用語

このトピックでは、Eclipse インフォメーション・センターで使用される用語について説明します。

プラグイン

特定のソフトウェア・システムの中の機能を変更 (追加または変更) する自己完結型ソフトウェア・コンポーネント。ユーザーがソフトウェア・システムにプラグインを追加するとき、元のソフトウェア・システムは損なわれずにそのまま残ります。文書プラグインの場合、プラグインには、製品資料および Eclipse での製品資料の表示方法に関連するすべてのファイルが含まれています。

ラーニング・パス

新規ユーザーが本製品のある機能領域について学習する際に役立つ順序でインフォメーション・センターのトピックを配置したもの。

情報ロードマップ

インフォメーション・センター内および Web 上の情報リソースへの広範なリンクの集合を含むトピックと、各リンクのリンク先の説明。

ローカル・モード

IC_local_start ファイルを使用した、インフォメーション・センターのワークステーションでのインプリメンテーション。インフォメーション・センターは自動的に、システムのデフォルト・ブラウザで起動します。

サーバー・モード

IC_server_start ファイルを使用した、インフォメーション・センターのサーバー上でのインプリメンテーション。インフォメーション・センターは、特定のポートで稼働します。ユーザーはそれぞれのブラウザを使用してリモートでインフォメーション・センターにアクセスします。

要件

インフォメーション・センターは、以下のオペレーティング・システムが稼働していることを必要とします。

- Windows 2000
- Windows XP
- AIX 5.2 および 5.3
- Linux RedHat Enterprise 3.0
- Linux SuSE Enterprise 3.0
- Linux RedHat Enterprise 8 および 9 for zSeries
- Linux SuSE Enterprise 8 および 9 for zSeries
- z/OS 1.4 以降

Linux for zSeries および z/OS 上でのインフォメーション・センターのサポートは、サーバー・モードでのみ提供されます。

CICS TS 3.1 Information Center は、JRE を使用します。z/OS を除いて、インフォメーション・センターには各プラットフォーム用の JRE が付属しています。z/OS でインフォメーション・センターを稼働する場合は、オペレーティング・システムに付属の JRE を使用する必要があります。

インフォメーション・センターの表示時に最良の結果を得るために、以下のいずれかのブラウザの使用をお勧めします。

- Microsoft Internet Explorer 6.0
- Mozilla 1.7

新機能セクション

新機能 セクションは、新リリースに関するすべての情報が含まれている新しいナビゲーション構造であり、インフォメーション・センターの他の部分と完全に統合されています。

CICS の新機能を説明した情報にアクセスするには、インフォメーション・センターのウェルカム・ページにあるリンクを開始点として使用することができます。この情報は、このリリースの 3 つのテーマである、CICS の統合、アプリケーション変換、エンタープライズ・マネージメントに分かれています。「新機能」セクションを展開することによって、ナビゲーションを使用して、特定の機能を探すこともできます。

情報ロードマップ

情報ロードマップとは、CICS の特定の機能領域についてのさまざまなリソースへのリンクを含む 1 つのトピックです。

各ロードマップは、セクションに分かれています。各セクションはトピックの最上部にある目次にまとめられています。それぞれのセクションで、基本スキル、インストール、マイグレーション、管理、アプリケーション開発、セキュリティーなどの各分野を扱っています。各セクション内のリンクでは、リソースがインフォメーション・センター内部のリソースか外部のリソースかを示し、リソースの内容を概説しています。リソースには、インフォメーション・センターのトピック、Redbooks、チュートリアル、ホワイト・ペーパー、記事などがあります。各ロードマップの最後には教育のセクションがあり、さらに詳しい資料、教育コースや教材へのリンクを掲載しています。

インフォメーション・センターには、3 つの情報ロードマップがあります。これら 3 つの情報ロードマップで、以下の機能を取り扱います。

- Web サービス
- CICSplex System Manager
- Java サポートおよび CICS のプログラミング

情報ロードマップは、インフォメーション・センターのウェルカム・ページから直接アクセスすることができます。あるいは、「目次」ペイン内でナビゲーションを展開します。

ラーニング・パス

ラーニング・パスは、CICS のある領域について学習する際に役立つ一連のトピックです。

各ラーニング・パスには、それぞれの目的と対象者を説明する概要があります。ラーニング・パスは、順を追って読む作りになっていますが、概要にラーニング・パスのすべてのステップが記載されているため、必要な箇所から読み始めることができます。各トピックの下部にパスを進んでいくためのボタン・ナビゲーションがあります。また、さらに詳細な情報へのリンクもあり、特定の機能や概念についての詳細情報を調べるために、いつでもパスを離れることができます。ラーニング・パスの終わりには、要約のトピックがあります。ここでは、学習したことをまとめて、インフォメーション・センター内のさらに詳しい資料やその後の作業への役立つリンクを示しています。

このリリースのラーニング・パスでは、以下の領域を扱っています。

- Web サービス

インフォメーション・センター内での検索の手法

インフォメーション・センターを検索するには、探す情報の種類とインフォメーション・センターの稼働に使用しているモードに応じて、いくつかの方法を使用することができます。

インフォメーション・センター内で検索を行うには、「**検索 (Search)**」フィールドに照会を入力します。「**目次**」ペインには、上位 500 件の結果が表示されます。トピックを表示するには、検索結果をクリックします。照会した語は強調表示されます。

ナビゲーション・ツリーから検索結果リストに (あるいはその逆に) 切り替えるには、「**目次**」ペイン下部の「目次」タブ () または「**検索結果 (Search results)**」タブ () をクリックします。

完全に一致する語または句の検索

二重引用符で囲むことにより、検索句を完全に一致するストリングとして識別することができます。例えば、"log file" の場合は、別々の語 *log* および *file* ではなく、ストリング *log file* が検索されます。引用符を指定しなかった場合、検索語は *log* と *file* の AND として解釈されます。

英語とドイツ語の場合のみ、検索エンジンは、単一の検索語の別の形式の「ステミング」(語幹を基にした変化形の検索)を行います。例えば、語 *challenge* を検索すると、語 *challenging* も検出されます。ステミングを行いたくない場合は、二重引用符を使用してください。

ワイルドカードを使用した検索

以下のワイルドカード文字を使用できます。

- * アスタリスク: 検索語内の複数の不明文字または可変文字に対応します。例えば、検索語 *par** は、*partly*、*participate*、*partial*、およびその他の *par* で始まる語を戻します。
- ? 疑問符: 単一の不明文字または可変文字に対応します。例えば、検索語 *par?* は、*part* を戻しますが、*partial* や *partly* は戻しません。

ブール演算子 (AND、OR、NOT) を使用した検索

検索語に 2 項演算子 (AND、OR および NOT) を挿入できます。例えば、次のようになります。

database AND "log file" : 語 *database* および "*log file*" の両方を含むトピックに検索を絞り込みます。

database OR "log file" : *database* または "*log file*" のどちらかを含むトピックに検索を広げます。

database NOT "log file" : *database* を含み、*log file* を含まないトピックを検索します。

検索スコープの絞り込み

デフォルトでは、「目次」ペインに表示されているすべてのトピックが検索されます。ただし、検索を特定のトピック・セットに絞り込むことができます。インフォメーション・センターをローカルで稼働している場合は、検索リストを使用して検索を絞り込んだり、後で再利用するために検索リストを保管することができます。

検索リストを作成するには、以下のようにします。

1. 「検索 (Search)」フィールドの横にある「**検索スコープ (Search scope):**」リンクをクリックします。「検索スコープの選択 (Select Search Scope)」ウィンドウが開きます。
2. 「**以下のトピックのみを検索 (Search only the following topics)**」を選択し、「新規 (New)」ボタンを選択します。「新規検索リスト (New Search List)」ウィンドウが開きます。
3. 「**検索するトピック (Topics to search)**」リストで、検索に含めるナビゲーション・カテゴリを選択します。カテゴリを展開して、特定の低位カテゴリのみを選択することができます。
4. 「**リスト名 (List name):**」テキスト・フィールドに適切な検索リスト名を入力し、「了解」ボタンを選択します。

作成した検索リストを使用して、検索式に対応するトピックを検索することができます。

インフォメーション・センターをサーバー上で稼働している場合は、「検索 (Search)」フィールドの隣にある「**拡張検索 (Advanced search):**」リンクを使用して、検索をナビゲーションの特定のセクションに絞り込むことができます。

メッセージ、コマンド、およびパラメーターを対象とした検索

文書内で、メッセージやコマンドなどの特定の参照項目をすばやく見つけるために、ナビゲーションにはルックアップ機能があります。ルックアップ機能を使用するには、ナビゲーションで「**コマンド、メッセージ、パラメーターなどの検索 (Find commands, messages, parameters and more)**」を選択します。この機能は、Lucene 検索エンジンを使用せず、代わりに索引を使用します。次の領域で検索を行うことができます。

- API コマンド
- API パラメーター
- SPI コマンド
- SPI パラメーター
- CICS トランザクション
- メッセージおよびコード
- トレース・エントリー
- リソース定義
- システム初期設定パラメーター

- 用語集の用語
- 図の表題
- 表の表題

インフォメーション・センターのナビゲート

インフォメーション・センターをナビゲートするには、「目次」ペイン内のリンクを選択または展開します。同期やキーボード・ショートカットなどの機能も使用できます。

ヘルプ・ウィンドウの左側にある「目次」ペイン () には、目次内のトピック表題またはナビゲーション・ツリー構造が表示されます。トピック表題をクリックすると、右側のメインパネルに内容を表示できます。ナビゲーションを展開すると、ネストされたセクションを表示することができます。HTML トピック内のリンクを

使用する場合は、「戻る」 () ボタンおよび「進む」 () ボタンを使用して表示されたトピックの履歴の中でナビゲートすることができます。

各 HTML トピックには、トピックの最後にリンクの集合があります。このリンクは、関連情報にナビゲートできるように作られています。また、多くのトピックにはボタン・ナビゲーションがあり、これを使用して、そのトピックが属する文書の目次、文書内の次のトピックや前のトピック、索引にも進むことができます。

検索を行った後に目次に戻りたい場合は、ペインの下部にある「目次」タブ () を選択します。ペインの下部にあるタブを使用して、いつでもナビゲーションと前回の検索結果の間を移動することができます。

目次の同期

トピック内のリンクをたどるときに、新しいトピックを表示および強調表示するようにナビゲーション・ツリーが自動的に変更になることはありません。新しいトピックがナビゲーション・ツリーのどこにあるかを調べて、2 つのビューを同期する

には、「最新表示」ボタン () または「目次で表示 (Show in Table of

Contents)」 () ボタンをクリックします。現在表示されているトピックのトピック表題が、ナビゲーション・ツリーで強調表示されます。

キーボードによるナビゲート

以下のキーの組み合わせを使用して、インフォメーション・センターをナビゲートできます。

- インフォメーション・センターのパネル内から次のリンク、ボタン、またはトピック・ノードに移動するには、Tab を押します。
- ツリー内のノードを展開または縮小するには、右矢印または左矢印を押します。
- 次のトピック・ノードに移動するには、下矢印または Tab を押します。
- 前のトピック・ノードに移動するには、上矢印を押すか、Shift を押しながら Tab を押します。

- 一番上または一番下にスクロールするには、Home または End を押します。
- 前に戻るには、Alt を押しながら左矢印を押します。次に進むには、Alt を押しながら右矢印を押します。
- 次のペインに移動するには、Ctrl を押しながら Tab を押します。
- 前のペインに移動するには、Shift と Ctrl を押しながら Tab を押します。
- アクティブ・ペインを印刷するには、Ctrl を押しながら P を押します。

トピックのブックマーク

インフォメーション・センターのトピックにブックマークを付けるには、2 つの方法があります。トピックにブックマークを付ける方法は、インフォメーション・センターをローカルで稼働しているか、サーバー上で稼働しているかによって異なります。

インフォメーション・センターをワークステーション上でローカルで稼働している場合は、以下の手順で選択したトピックにブックマークを付けることができます。

1. メインパネルの上部にある「ブックマーク (Bookmark) ボタン 」を選択します。これにより、「ブックマーク (Bookmarks) パネルにブックマークが付けられます。
2. 「目次」ペインの下部にある「ブックマーク・ビュー (Bookmarks view) アイコン 」をクリックします。「目次」ペインが「ブックマーク (Bookmarks) ペインで置き換えられて、すべてのブックマークが表示されます。選択したトピックは、ブックマークのリストの下部に配置されます。

ブックマークを削除したい場合は、ブックマークを選択して、Delete (削除) キーを押します。

サーバー上でインフォメーション・センターにアクセスする場合は、ブラウザを使用してトピックにブックマークを付けることができます。

ユーザー設定

ユーザー設定を使用すると、インフォメーション・センターの機能の特定の性質をカスタマイズすることができます。

ユーザー設定は、対話式テーブル、アクセス可能な構文図、およびテキストイメージ・オプションです。このような設定がナビゲーションの「製品とインフォメーション・センターの概要 (Product and information center overview)」セクションから使用できるようになりました。

第 31 章 CICS Transaction Server for z/OS ライブラリー

CICS Transaction Server for z/OS Information Center は、CICS Transaction Server 用ユーザー情報の基本ソースです。CICS TS の資料のうちの数冊 (同梱セット) はハードコピーで入手できます。

Information Center には、以下のものが含まれています。

- HTML 形式での CICS Transaction Server の情報。
- Adobe PDF ファイルで提供されている、ライセンス交付を受けた CICS Transaction Server 資料およびライセンス交付を受けていない CICS Transaction Server 資料。これらのファイルを使用して資料のハードコピーを印刷することができます。
- HTML 形式および PDF ファイルでの関連製品の情報

CICS Information Center は、CD-ROM で製品と共に 1 部自動的に配布されます。Information Center のフィーチャー番号 7014 を指定することにより、複数の部数を追加料金なしで注文できます。必須資料一式は、ハードコピーでも配布されます。

ライセンス文書は製品のライセンス所有者だけが使用できます。ライセンスの交付を受けていない情報のみを含む Information Center のバージョンは、資料オーダー・システム、オーダー番号 SK3T-6945 で入手可能です。

ハードコピーで入手できる資料

CICS Transaction Server for z/OS バージョン 3 リリース 1 をオーダーすると、数冊のハードコピーの資料が配布されます。

このハードコピーの資料は、以下のとおりです。

Memo to Licensees, GI10-2585

CICS Transaction Server for z/OS Program Directory, GI10-2586

CICS Transaction Server for z/OS リリース・ガイド、GD88-6377

CICS Transaction Server for z/OS Licensed Program Specification, GC34-6608

上に引用したオーダー番号を使用して、以下の資料のコピーをさらにオーダーすることができます。

CICS Transaction Server for z/OS リリース・ガイド

CICS Transaction Server for z/OS インストール・ガイド

CICS Transaction Server for z/OS Licensed Program Specification

PDF のみで提供される資料

ライセンス交付を受けた CICS Transaction Server 資料およびライセンス交付を受けていない CICS Transaction Server 資料が、CICS Information Center において Adobe PDF ファイル形式で提供されます。これらのファイルを使用して資料のハードコピーを印刷することができます。

CICS Transaction Server for z/OS の CICS 資料

一般

CICS Transaction Server for z/OS Program Directory, GH10-2586
CICS Transaction Server for z/OS リリース・ガイド、GD88-6377
CICS Transaction Server for z/OS CICS TS V1.3 からのマイグレーション、
GD88-6378
CICS Transaction Server for z/OS CICS TS V2.2 からのマイグレーション、
GD88-6379
CICS Transaction Server for z/OS CICS TS V2.3 からのマイグレーション、
GD88-6380
CICS Transaction Server for z/OS インストール・ガイド、GD88-6381

CICS へのアクセス

CICS Internet Guide、SC34-6450
CICS Web サービス・ガイド、SD88-6383

管理

CICS システム定義ガイド、SD88-6526
CICS Customization Guide、SC34-6429
CICS Resource Definition Guide、SC34-6430
CICS Operations and Utilities Guide、SC34-6431
CICS RACF Security Guide、SC34-6454
CICS Supplied Transactions、SC34-6432

プログラミング

CICS Application Programming Guide、SC34-6433
CICS Application Programming Reference、SC34-6434
CICS System Programming Reference、SC34-6435
CICS FEPI ユーザーズ・ガイド、SD88-6382
CICS C++ OO Class Libraries、SC34-6437
CICS Distributed Transaction Programming Guide、SC34-6438
CICS Business Transaction Services、SC34-6439
Java Applications in CICS、SC34-6440
JCICS Class Reference、SC34-6001

診断

CICS Problem Determination Guide、SC34-6441
CICS パフォーマンス・ガイド、SD88-6391
CICS Messages and Codes、SC34-6442
CICS Diagnosis Reference、LY33-6110
CICS Recovery and Restart Guide、SC34-6451
CICS Data Areas、LY33-6107
CICS Trace Entries、SC34-6443
CICS Supplementary Data Areas、LY33-6108
CICS Debugging Tools Interfaces Reference、LY33-6109

通信

CICS Intercommunication Guide、SC34-6448
CICS External Interfaces Guide、SC34-6449

データベース

CICS DB2 Guide, SC34-6457

CICS IMS Database Control Guide, SC34-6453

CICS Shared Data Tables Guide, SC34-6455

CICS Transaction Server for z/OS の CICSplex SM 資料

一般

CICSplex SM Concepts and Planning, SC34-6459

CICSplex SM User Interface Guide, SC34-6460

CICSplex SM Web User Interface Guide, SC34-6461

管理

CICSplex SM Administration, SC34-6462

CICSplex SM Operations Views Reference, SC34-6463

CICSplex SM Monitor Views Reference, SC34-6464

CICSplex SM Managing Workloads, SC34-6465

CICSplex SM Managing Resource Usage, SC34-6466

CICSplex SM Managing Business Applications, SC34-6467

プログラミング

CICSplex SM Application Programming Guide, SC34-6468

CICSplex SM Application Programming Reference, SC34-6469

診断

CICSplex SM Resource Tables Reference, SC34-6470

CICSplex SM Messages and Codes, GC34-6471

CICSplex SM Problem Determination, GC34-6472

CICS ファミリー資料

通信

CICS ファミリー プロダクト間通信ガイド, SD88-6384

CICS ファミリー S/390 CICS からの通信, SD88-6385

ライセンス出版物

以下のライセンス出版物は、Information Center のライセンス交付を受けていないバージョンには含まれていません。

CICS Diagnosis Reference, LY33-6102

CICS Data Areas, LY33-6103

CICS Supplementary Data Areas, LY33-6104

CICS Debugging Tools Interfaces Reference, LY33-6105

アクセシビリティ

アクセシビリティ機能は、運動障害または視覚障害など身体に障害を持つユーザーがソフトウェア・プロダクトを快適に使用できるようにサポートします。

CICS システムの設定、実行、および保守に関するほとんどの作業は、以下のいずれかの方法で実行できます。

- CICS にログオンした 3270 エミュレーターを使用
- TSO にログオンした 3270 エミュレーターを使用
- MVS システム・コンソールとして 3270 エミュレーターを使用

IBM パーソナル・コミュニケーションズは、身体に障害のあるユーザーのためのアクセシビリティ機能を備えた 3270 エミュレーションを提供します。この製品を使用すると、CICS システムで必要となるアクセシビリティ機能を提供できます。

アクセシビリティ機能には、WebSphere Application Server のコンポーネントである Enterprise Beans 用のアプリケーション・アセンブリー・ツール (ATK および AAT) の使用時には利用できないものがあります。これらのツールの使用時に利用できるアクセシビリティ機能を判別するには、WebSphere Application Server に付属の資料で調べる必要があります。

EJB リソースの操作のために Enterprise Bean 用のリソース・マネージャーを使用する場合は、お使いの Web ブラウザーがアクセシビリティ機能を提供します。特に、マウス・ポインターを画面の該当部分に置くと表示されるヘルプは、そのパネルのヘルプ機能を使用して表示することもできるので注意してください。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アシスタント、Web サービス 31
新しく導入された API コマンド
 EXTRACT STATISTICS 333
アナライザー・プログラム 161
アナライザー・プログラム 132
 チャンク形式転送コーディング 110
アプリケーション・プログラミング・インターフェース
 (API)
 新規 API コマンド 232
 変更された API コマンド 243
 JCICS の変更点 247
アプリケーション・プログラミング・インターフェース、CICSplex SM
 新規リソース・テーブル
 HOST 170
 PIPEDEF 69
 PIPELINE 69
 URIMAP 170
 URIMPDEF 170
 URIMPGBL 170
 WEBSERV 69
 WEBSVDEF 69
 変更された リソース・テーブル
 CICSRGN 192, 283
 EJCODEF 192
 EJCOSE 192
 PROGDEF 283
 PROGRAM 283
 TASK 192, 283
 TCPDEF 192
 TCPIPGBL 192
 TCPIPS 192
 変更されたリソース・テーブル
 CICSRGN 368
 CONNECT 261
 DOCDEF 170
 DOCTEMP 170
 PROGDEF 368
 PROGRAM 368
 TASK 170, 261
 TCPDEF 170
 TERMDEF 361
 WORKREQ 69

異常終了コード 254
インストール
 新規プロセス 331
インフォメーション・センター
 検索の手法 397
 情報ロードマップ 396
 ブックマーク 400
 要件 395
 ラーニング・パス 396
インフォメーション・センターのナビゲート 399
エンド・ユーザー・インターフェース、CICSplex SM
 変更されたビュー
 DOCDEF 169
 DOCTEMP 169
 EJCODEF 191
 EJCOSE 191
 PROGDEF 368
 PROGRAM 368
 TCPDEF 169, 191
 TCPIPS 191
 TERMDEF 361
お気に入り 302, 304
お気に入りのエディター 302

[カ行]

概要
 基本的な例 201
 現行チャンネル 208
 コンテナ 201
 コンポーネント 205
 チャンネル 201
 チャンネルと BTS アクティビティ 221
 チャンネルの作成 207
 チャンネルのスコープ 213
 チャンネルの生成 219
 チャンネルの設計 218
 チャンネルを使用した動的ルーティング 226, 259
 データ変換 227
 典型的シナリオ 204
 プログラムに渡されたコンテナの発見 217
 読み取り専用コンテナ 218
 リンクから戻されたコンテナの発見 217
 JCICS からのチャンネルの使用 223
仮想ホスティング 113, 114
結果セット警告カウント 313
現行チャンネル 208
 概要 212

言語構造
WSDL への変換 33
高水準言語構造
WSDL への変換 33
構文表記法 xii
コンテキスト
コンテナの、BTS またはチャンネル 222
コンテナ
概要 199
基本的な例 201
コンテキスト、BTS またはチャンネル 222
作成 224
チャンネルの設計 218
発見、プログラムに渡された 217
発見、リンクから戻された 217
読み取り専用 218
JCICS からの使用 223
コンテナ・コマンド
DELETE CONTAINER (CHANNEL) 232
GET CONTAINER (CHANNEL) 233
MOVE CONTAINER (CHANNEL) 236
PUT CONTAINER (CHANNEL) 238
コンバーター・プログラム 163
コンバーター・プログラム 132
コンポーネント
複数の対話式 206
複数のチャンネル — 1 つのコンポーネント 205
1 つのチャンネル — 複数のプログラム 205

[サ行]

サンプル・プログラム 253
システム初期設定パラメーター
CLINTCP 346
SRVERCP 346
持続接続 112
HTTP クライアントとしての CICS 77
シナリオ
複数の対話式コンポーネント 206
複数のチャンネル — 1 つのコンポーネント 205
1 つのチャンネル — 複数のプログラム 205
1 つのチャンネル — 1 つのプログラム 204
詳細ビュー
2 つのカラム 318
証明書取り消しリスト 182
図
構文 xii
「すべて選択」アイコン 321
スレッド・セーフ API コマンド
このリリースで新規 383
セキュリティー
新しい ESM 機能プロファイル 308

セッション・トークン 75, 77
接続、持続 112
「選択をすべて解除」アイコン 321
前提条件
ハードウェア 377
前提条件ハードウェア 377

[タ行]

大容量の COMMAREA 199, 201, 204, 231
大容量の COMMAREA としてのチャンネル 199, 201, 204, 231, 240
大容量の COMMAREA、チャンネル 240
大/小文字混合
パスワード 337
チャンネル
概要 199
基本的な例 201
現行 208, 212
作成 207, 224
スコープ 213
生成 219
設計 218
大容量の COMMAREA として 199
典型的シナリオ
複数の対話式コンポーネント 206
複数のチャンネル — 1 つのコンポーネント 205
1 つのチャンネル — 複数のプログラム 205
1 つのチャンネル — 1 つのプログラム 204
動的および分散ルーティング 226, 259
プログラムに渡されたコンテナの発見 217
読み取り専用コンテナ 218
利点 199
リンクから戻されたコンテナの発見 217
BTS アクティビティーとの比較 221
JCICS からの使用 223
チャンネルの作成 207
チャンネルのスコープ 213
チャンネルの生成 219
チャンネルの設計 218
チャンネルを使用した動的ルーティング 226, 259
チャンネル・コマンド
DELETE CONTAINER (CHANNEL) 232
GET CONTAINER (CHANNEL) 233
MOVE CONTAINER (CHANNEL) 236
PUT CONTAINER (CHANNEL) 238
START CHANNEL 240
チャンク化 109, 110
チャンク形式転送コーディング 109, 110
データ変換 227
およびチャンネル 227
コード・ページ 345

データ・リポジトリ
新規メッセージ 327
統計 252
動的選択リスト 316
動的ルーティング・プログラム 248
取り消されたユーザー ID 351
トレース・ポイント 257
トレーラー 109

[ハ行]

ハードウェア前提条件 377
パイプライン化 111
パスワード
大/小文字混合 337
バッチ・ユーティリティ
Web サービス・アシスタント 31
ビジネス・アプリケーション・サービス 310
ビュー、新規
Web ユーザー・インターフェース
機能シップ 263
チャンネル使用 263
パイプライン 71
パイプライン 定義 71
ホスト 175
URI マッピング定義 175
URI マップ 175
URI マップ・グローバル 175
Web サービス 71
Web サービス定義 71
ビュー、変更された
エンド・ユーザー・インターフェース、CICSplex SM
DOCDEF 169
DOCTEMP 169
EJCODEF 191
EJCOSE 191
PROGDEF 368
PROGRAM 368
TCPDEF 169, 191
TCPIPS 191
TERMDEF 361
Web ユーザー・インターフェース
クロックとタイミング 193
処理要求 71
端末定義 361
プログラム 284, 368
プログラム定義 284, 368
文書テンプレート 175
文書テンプレート定義 175
CICS 領域 193
CorbaServer 193
CorbaServer 定義 193

ビュー、変更された (続き)
Web ユーザー・インターフェース (続き)
CPU と TCB 情報 284
ISC/MRO 接続 263
TCP/IP グローバル状況 193
TCP/IP サービス 193
TCP/IP サービス定義 175, 193
TCP/IP 使用 175

表記

構文 xii
フィルター
展開と縮小 319
フィルター確認 315
プログラム、サンプル 253
プログラムに渡されたコンテナの発見 217
分散ルーティング・プログラム 248
べき等 111

[マ行]

マイグレーション
新機能を活用しない場合 257
新機能を活用する場合 257
末尾ヘッダー 109, 110
メッセージ 254, 324, 327
モニター 250
問題判別
異常終了コード 254
トレース・ポイント 257
メッセージ 254

[ヤ行]

ユーザーが置換可能なプログラム
動的ルーティング・プログラム 248
分散ルーティング・プログラム 248
ユーザーのお気に入り 302
管理 302, 304
ユーザー・エディター 304
ユーザー・オブジェクト 302, 305
ユーザー・グループ・プロファイル 307
作成 309
作成と管理 308
ユーティリティ・プログラム
Web サービス・アシスタント 31
要件、ハードウェア 377
読み取り専用コンテナ 218

[ラ行]

リソース割り当てビュー 310
利点 199

リモート MAS 371
リンクから戻されたコンテナの発見 217
例

基本 201
単純クライアント・プログラムと BTS アクティビティ
ーとの比較 221
チャンネルを使用する CICS サーバー・プログラム
221
チャンネルを生成して使用する Java クライアント・プ
ログラム 226
チャンネルを生成する CICS クライアント・プログラ
ム 220
複数の対話式コンポーネント 206
複数のチャンネル — 1 つのコンポーネント 205
1 つのチャンネル — 複数のプログラム 205
1 つのチャンネル — 1 つのプログラム 204

[数字]

2 つのカラム詳細ビュー
項目の削除 323
作成 321
32K 以上の COMMAREA 199, 201, 231, 232
32K 以上の COMMAREA (チャンネル)
START CHANNEL コマンド 240
32K 超の COMMAREA 199, 201, 231, 232
32K 超の COMMAREA (チャンネル)
START CHANNEL コマンド 240

A

ABSTIME オプション
CONVERTTIME コマンド 152
ACTION オプション
WEB CONVERSE コマンド 95
WEB SEND コマンド (クライアント) 85
WEB SEND コマンド (サーバー) 119
ANALYZER 属性
URIMAP 定義 136
ANALYZERSTAT オプション
INQUIRE URIMAP コマンド 154
API (アプリケーション・プログラミング・インターフ
ェース)
新規 API コマンド 232
変更された API コマンド 243
JCICS の変更点 247
API コマンド
EXTRACT STATISTICS 333
API の変更点
CICSplex SM 323, 326
AS オプション
MOVE CONTAINER (CHANNEL) コマンド 237

B

BAS 310
BATCHREP リソース・テーブル 326
BINDING オプション
INQUIRE WEBSERVICE コマンド 58
BTS アクティビティ 221

C

CCRL トランザクション 182
CERTIFICATE オプション
INQUIRE URIMAP コマンド 155
WEB OPEN コマンド 80
CERTIFICATE 属性
URIMAP 定義 137
CHANNEL オプション
ASSIGN コマンド 243
DELETE CONTAINER (CHANNEL) コマンド 233
GET CONTAINER (CHANNEL) コマンド 233
LINK コマンド 244
MOVE CONTAINER (CHANNEL) コマンド 237
PUT CONTAINER (CHANNEL) コマンド 238
RETURN コマンド 245
START TRANSID (CHANNEL) コマンド 242
XCTL コマンド 246
CHANNELERR 条件
LINK コマンド 244
RETURN コマンド 245
XCTL コマンド 246
CHARACTERSET オプション
INQUIRE URIMAP コマンド 155
WEB CONVERSE コマンド 99
WEB RECEIVE コマンド (サーバー) 123
WEB SEND コマンド (クライアント) 85
WEB SEND コマンド (サーバー) 119
CHARACTERSET 属性
URIMAP 定義 137
CHUNKING オプション
WEB SEND コマンド (クライアント) 86
WEB SEND コマンド (サーバー) 119
CICS Web サポート
仮想ホスティング 114
持続接続 112
チャンク形式転送コーディング 110
パイプライン化 111
ユーザー出口 XWOPEN、XWBSNDO 101, 103
CICS Web サポート・コマンド
CONVERSE WEB 93
WEB CLOSE 83
WEB CONVERSE 93
WEB EXTRACT 147

- CICS Web サポート・コマンド (続き)
 - WEB OPEN 79
 - WEB PARSE URL 145
 - WEB RECEIVE 123
 - WEB RECEIVE (クライアント) 90
 - WEB SEND (クライアント) 84
 - WEB SEND (サーバー) 118
- CICS 外部インターフェースの変更点
 - 異常終了コード 254
 - 統計 252
 - トレース・ポイント 257
 - メッセージ 254
 - モニター 250
 - 問題判別 253
 - ユーザーが置換可能なプログラム 248
 - API 231
 - CICS サンプル・プログラム 253
- CICS リソース定義 310
- CICSplex SM
 - 結果セット警告カウント 313
 - 新機能 300
 - 新規メッセージ 324, 327
 - 選択リスト 316
 - ビジネス・アプリケーション・サービス (BAS) 310
 - フィルター確認 315
 - ユーザーのお気に入り 302, 304
 - ユーザー・グループ・プロファイル 307, 308
 - リモート MAS 371
 - API の変更点 323, 326
 - BATCHREP 機能 325
 - Web ユーザー・インターフェース 326
- CICSplex SM、アプリケーション・プログラミング・インターフェース
 - 新規リソース・テーブル
 - HOST 170
 - PIPEDEF 69
 - PIPELINE 69
 - URIMAP 170
 - URIMPDEF 170
 - URIMPGBL 170
 - WEBSERV 69
 - WEBSVDEF 69
 - 変更された リソース・テーブル
 - CICSRGN 192, 283
 - EJCODEF 192
 - EJCOSE 192
 - PROGDEF 283
 - PROGRAM 283
 - TASK 192, 283
 - TCPDEF 192
 - TCPIPGBL 192
 - TCPIPS 192
- CICSplex SM、アプリケーション・プログラミング・インターフェース (続き)
 - 変更されたリソース・テーブル
 - CICSRGN 368
 - CONNECT 261
 - DOCDEF 170
 - DOCTEMP 170
 - PROGDEF 368
 - PROGRAM 368
 - TASK 170, 261
 - TCPDEF 170
 - TERMDEF 361
 - WORKREQ 69
- CICSplex SM、エンド・ユーザー・インターフェース
 - 変更されたビュー
 - DOCDEF 169
 - DOCTEMP 169
 - EJCODEF 191
 - EJCOSE 191
 - PROGDEF 368
 - PROGRAM 368
 - TCPDEF 169, 191
 - TCPIPS 191
 - TERMDEF 361
- CIPHERS オプション
 - INQUIRE URIMAP コマンド 155
 - WEB OPEN コマンド 80
- CIPHERS 属性
 - URIMAP 定義 137
- CLIENTCONV オプション
 - WEB CONVERSE コマンド 99
 - WEB RECEIVE コマンド (クライアント) 91
 - WEB SEND コマンド (クライアント) 87
- CLNTCODEPAGE オプション
 - WEB RECEIVE コマンド (サーバー) 124
 - WEB SEND コマンド (サーバー) 120
- CLOSESTATUS オプション
 - WEB CONVERSE コマンド 95
 - WEB SEND コマンド (クライアント) 87
 - WEB SEND コマンド (サーバー) 120
- CODEPAGE オプション
 - WEB OPEN コマンド 81
- CONTAINER オプション
 - DELETE CONTAINER (CHANNEL) コマンド 233
 - GET CONTAINER (CHANNEL) コマンド 233
 - INQUIRE WEBSERVICE コマンド 59
 - MOVE CONTAINER (CHANNEL) コマンド 237
 - PUT CONTAINER (CHANNEL) コマンド 239
- CONVERSE WEB コマンド 93
- CONVERTER オプション
 - INQUIRE URIMAP コマンド 155

CONVERTER 属性
 URIMAP 定義 137
 CONVERTTIME コマンド 151
 CRL (証明書取り消しリスト) 182
 CVDA 値
 CHUNKEND
 WEB SEND コマンド (クライアント) 86
 WEB SEND コマンド (サーバー) 120
 CHUNKNO
 WEB SEND コマンド (クライアント) 86
 WEB SEND コマンド (サーバー) 120
 CHUNKYES
 WEB SEND コマンド (クライアント) 86
 WEB SEND コマンド (サーバー) 120
 CLICONVERT
 WEB CONVERSE コマンド 100
 WEB RECEIVE コマンド (クライアント) 91
 WEB SEND コマンド (クライアント) 87
 CLIENT
 INQUIRE URIMAP コマンド 157
 CLOSE
 WEB CONVERSE コマンド 95
 WEB SEND コマンド (クライアント) 87
 WEB SEND コマンド (サーバー) 120
 DELETE
 WEB CONVERSE コマンド 89, 97
 DISABLED
 INQUIRE HOST コマンド 127
 INQUIRE URIMAP コマンド 155, 156
 SET HOST コマンド 128
 SET URIMAP コマンド 158
 DISABLEDHOST
 INQUIRE URIMAP コマンド 155
 ENABLED
 INQUIRE HOST コマンド 127
 INQUIRE URIMAP コマンド 155, 156
 SET HOST コマンド 128
 SET URIMAP コマンド 158
 EVENTUAL
 WEB SEND コマンド (サーバー) 119
 EXPECT
 WEB CONVERSE コマンド 95
 WEB SEND コマンド (クライアント) 85
 GET
 WEB CONVERSE コマンド 96
 WEB SEND コマンド (クライアント) 88
 HEAD
 WEB CONVERSE コマンド 96
 WEB SEND コマンド (クライアント) 88
 HTTP
 WEB EXTRACT コマンド 151
 WEB OPEN コマンド 82
 HTTPNO
 WEB EXTRACT コマンド 150
 WEB RECEIVE コマンド (サーバー) 125
 HTTPS
 WEB EXTRACT コマンド 151
 WEB OPEN コマンド 82
 HTTPYES
 WEB EXTRACT コマンド 150
 WEB RECEIVE コマンド (サーバー) 125
 IMMEDIATE
 WEB SEND コマンド (サーバー) 119
 NO
 INQUIRE URIMAP コマンド 155
 NOCLICONVERT
 WEB CONVERSE コマンド 100
 WEB RECEIVE コマンド (クライアント) 91
 WEB SEND コマンド (クライアント) 87
 NOCLOSE
 WEB CONVERSE コマンド 95
 WEB SEND コマンド (クライアント) 87
 WEB SEND コマンド (サーバー) 120
 NOINCONVERT
 WEB CONVERSE コマンド 100
 NONE
 INQUIRE URIMAP コマンド 157
 SET URIMAP コマンド 159
 NOOUTCONVERT
 WEB CONVERSE コマンド 100
 NOSRVCONVERT
 WEB RECEIVE コマンド (サーバー) 125
 WEB SEND コマンド (サーバー) 122
 OPTIONS
 WEB CONVERSE コマンド 96
 WEB SEND コマンド (クライアント) 89
 PERM
 INQUIRE URIMAP コマンド 157
 PERMANENT
 SET URIMAP コマンド 159
 PIPELINE
 INQUIRE URIMAP コマンド 157
 PUT
 WEB CONVERSE コマンド 96
 WEB SEND コマンド (クライアント) 89
 RFC1123
 FORMATTIME コマンド 153
 SERVER
 INQUIRE URIMAP コマンド 157
 SRVCONVERT
 WEB RECEIVE コマンド (サーバー) 125
 WEB SEND コマンド (サーバー) 122
 CVDA 値 (続き)

CVDA 値 (続き)

TEMP

INQUIRE URIMAP コマンド 157

TEMPORARY

SET URIMAP コマンド 159

TRACE

WEB CONVERSE コマンド 96

WEB SEND コマンド (クライアント) 89

YES

INQUIRE URIMAP コマンド 154

D

DATATYPE オプション

PUT CONTAINER (CHANNEL) コマンド 239

DATESTRING オプション

CONVERTTIME コマンド 152

FORMATTIME コマンド 153

DEFAULTWARNCNT WUI パラメーター 314

DELETE CONTAINER (CHANNEL) コマンド 232

DFHCNV 345

マクロの変更点 346

DFHLS2WS

カタログ式プロシージャー 33

DFHWBEP (Web エラー・プログラム) 164

DFHWS2LS

カタログ式プロシージャー 38

DNS サーバー 113

DOCTOKEN オプション

WEB CONVERSE コマンド 95

WEB SEND コマンド (クライアント) 87

WEB SEND コマンド (サーバー) 121

E

ENABLESTATUS オプション

INQUIRE HOST コマンド 127

INQUIRE URIMAP コマンド 155

SET HOST コマンド 128

SET URIMAP コマンド 158

ENDPOINT オプション

INQUIRE WEBSERVICE コマンド 59

EXTRACT STATISTICS コマンド 333

F

FILTERSTYLE 初期設定パラメーター 320

FLENGTH オプション

GET CONTAINER (CHANNEL) コマンド 234

PUT CONTAINER (CHANNEL) コマンド 240

FROM オプション

PUT CONTAINER (CHANNEL) コマンド 240

FROM オプション (続き)

WEB CONVERSE コマンド 95

WEB SEND コマンド (クライアント) 88

WEB SEND コマンド (サーバー) 121

FROMCCSID オプション

PUT CONTAINER (CHANNEL) コマンド 240

FROMLENGTH オプション

WEB CONVERSE コマンド 96

WEB SEND コマンド (クライアント) 88

WEB SEND コマンド (サーバー) 121

G

GET CONTAINER (CHANNEL) コマンド 233

H

HFSFILE オプション

INQUIRE URIMAP コマンド 155

HFSFILE 属性

URIMAP 定義 138

HOST オプション

INQUIRE HOST コマンド 127

INQUIRE URIMAP コマンド 155

WEB EXTRACT コマンド 148

WEB OPEN コマンド 81

WEB PARSE URL コマンド 146

HOST 属性

URIMAP 定義 139

HOSTCODEPAGE オプション

INQUIRE URIMAP コマンド 155

WEB RECEIVE コマンド (サーバー) 124

WEB SEND コマンド (サーバー) 121

HOSTCODEPAGE 属性

URIMAP 定義 139

HOSTLENGTH オプション

WEB EXTRACT コマンド 149

WEB OPEN コマンド 81

WEB PARSE URL コマンド 146

HTTP クライアントとしての CICS

パイプライン化要求 111

HTTP 要求と応答の処理

持続接続 112

チャンク形式転送コーディング 110

パイプライン化 111

HTTP クライアントとしての CICS 75

HTTP サーバーとしての CICS 132

HTTPMETHOD オプション

WEB EXTRACT コマンド 149

HTTPRNUM オプション

WEB OPEN コマンド 81

HTTPVERSION オプション
WEB EXTRACT コマンド 149
HTTPVNUM オプション
WEB OPEN コマンド 81

I

INQUIRE HOST コマンド 126
INQUIRE URIMAP コマンド 153
INQUIRE WEBSERVICE コマンド 58
INTO オプション
GET CONTAINER (CHANNEL) コマンド 234
WEB CONVERSE コマンド 98
WEB RECEIVE コマンド (クライアント) 91
WEB RECEIVE コマンド (サーバー) 124
INTOCCSID オプション
GET CONTAINER (CHANNEL) コマンド 234
INVOKE WEBSERVICE コマンド 50

J

JCICS
およびチャネル 223
現行チャネルの受取 225
現行チャネルのブラウズ 225
コンテナからのデータの取得 225
コンテナの作成 224
チャネルの作成 224
プログラム例 225
変更点 247
JCICS からのチャネルの使用 223

L

LASTMODTIME オプション
INQUIRE WEBSERVICE コマンド 59
LASTRESET オプション
EXTRACT STATISTICS コマンド 335
LASTRESETHRS オプション
EXTRACT STATISTICS コマンド 335
LASTRESETMIN オプション
EXTRACT STATISTICS コマンド 335
LASTRESETSEC オプション
EXTRACT STATISTICS コマンド 335
LENGTH オプション
WEB RECEIVE コマンド (クライアント) 91
WEB RECEIVE コマンド (サーバー) 124
WEB SEND コマンド (サーバー) 121
LOCATION オプション
INQUIRE URIMAP コマンド 156
SET URIMAP コマンド 158

LOCATION 属性
URIMAP 定義 139

M

MAXLENGTH オプション
WEB CONVERSE コマンド 98
WEB RECEIVE コマンド (クライアント) 91
WEB RECEIVE コマンド (サーバー) 124
MEDIATYPE オプション
INQUIRE URIMAP コマンド 156
WEB CONVERSE コマンド 96, 98
WEB RECEIVE コマンド (クライアント) 92
WEB SEND コマンド (クライアント) 88
WEB SEND コマンド (サーバー) 122
MEDIATYPE 属性
URIMAP 定義 140
METHOD オプション
WEB CONVERSE コマンド 96
WEB SEND コマンド (クライアント) 88
METHODLENGTH オプション
WEB EXTRACT コマンド 149
MOVE CONTAINER (CHANNEL) コマンド 236

N

NODATA オプション
GET CONTAINER (CHANNEL) コマンド 235
NOTRUNCATE オプション
WEB CONVERSE コマンド 98
WEB RECEIVE コマンド (クライアント) 92
WEB RECEIVE コマンド (サーバー) 125
NUMCIPHERS オプション
INQUIRE URIMAP コマンド 156
WEB OPEN コマンド 82

P

PATH オプション
INQUIRE URIMAP コマンド 156
WEB CONVERSE コマンド 97
WEB EXTRACT コマンド 149
WEB PARSE URL コマンド 147
WEB SEND コマンド (クライアント) 89
PATH 属性
URIMAP 定義 140
PATHLENGTH オプション
WEB CONVERSE コマンド 97
WEB EXTRACT コマンド 150
WEB PARSE URL コマンド 147
WEB SEND コマンド (クライアント) 89

PGMINTERFACE オプション
 INQUIRE WEBSERVICE コマンド 59
 PIPELINE CVDA 値
 EXTRACT STATISTICS コマンド 335
 PIPELINE オプション
 INQUIRE URIMAP コマンド 156
 INQUIRE WEBSERVICE コマンド 59
 PIPELINE 属性
 URIMAP 定義 141
 PIPELINE 定義
 SHELF 属性 45
 WSDIR 属性 46
 PIPELINE リソース定義 44
 PORTNUMBER オプション
 WEB EXTRACT コマンド 150
 WEB OPEN コマンド 82
 WEB PARSE URL コマンド 147
 PROGRAM オプション
 INQUIRE URIMAP コマンド 156
 INQUIRE WEBSERVICE コマンド 60
 PROGRAM 属性
 URIMAP 定義 141
 pthreads 182
 PUT CONTAINER (CHANNEL) コマンド 238

Q

QUERYSTRING オプション
 WEB EXTRACT コマンド 150
 WEB PARSE URL コマンド 147
 WEB SEND コマンド 97
 WEB SEND コマンド (クライアント) 89
 QUERYSTRLEN オプション
 WEB EXTRACT コマンド 150
 WEB PARSE URL コマンド 147
 WEB SEND コマンド 97
 WEB SEND コマンド (クライアント) 89

R

REDIRECTTYPE オプション
 INQUIRE URIMAP コマンド 157
 SET URIMAP コマンド 159
 REDIRECTTYPE 属性
 URIMAP 定義 141
 REQUESTTYPE オプション
 WEB EXTRACT コマンド 150
 RESID オプション
 EXTRACT STATISTICS コマンド 335
 RESTYPE オプション
 EXTRACT STATISTICS コマンド 335

S

SCHEME オプション
 INQUIRE URIMAP コマンド 156
 WEB EXTRACT コマンド 151
 WEB OPEN コマンド 82
 SCHEME 属性
 URIMAP 定義 142
 SCHEMENAME オプション
 WEB PARSE URL コマンド 147
 SERVERCONV オプション
 WEB RECEIVE コマンド (サーバー) 125
 WEB SEND コマンド (サーバー) 122
 SESSTOKEN オプション 101
 WEB CLOSE コマンド 84
 WEB CONVERSE コマンド 97
 WEB EXTRACT コマンド 151
 WEB OPEN コマンド 82
 WEB RECEIVE コマンド (クライアント) 92
 WEB SEND コマンド (クライアント) 89
 SET HOST コマンド 127
 SET URIMAP コマンド 158
 SET WEBSERVICE コマンド 61
 SET オプション
 EXTRACT STATISTICS コマンド 336
 GET CONTAINER (CHANNEL) コマンド 235
 WEB CONVERSE コマンド 98
 WEB RECEIVE コマンド (クライアント) 92
 WEB RECEIVE コマンド (サーバー) 125
 SHELF 属性
 PIPELINE 定義 45
 SOAPFAULT ADD コマンド 51
 SOAPFAULT CREATE コマンド 52
 SOAPFAULT DELETE コマンド 54
 SSL 接続の改善 182
 SSL プール 182
 START CHANNEL コマンド 240
 STARTCODE オプション
 ASSIGN コマンド 243
 STATE オプション
 INQUIRE WEBSERVICE コマンド 60
 STATUS 属性
 URIMAP 定義 142
 STATUSCODE オプション
 WEB CONVERSE コマンド 98
 WEB RECEIVE コマンド (クライアント) 92
 WEB SEND コマンド (サーバー) 122
 STATUSLEN オプション
 WEB CONVERSE コマンド 99
 WEB RECEIVE コマンド (クライアント) 93
 WEB SEND コマンド (サーバー) 122

STATUSTEXT オプション
 WEB CONVERSE コマンド 99
 WEB RECEIVE コマンド (クライアント) 92
 WEB SEND コマンド (サーバー) 122
STRINGFORMAT オプション
 FORMATTIME コマンド 153
SYSDEF オペランド 346
SYSID オプション
 START TRANSID (CHANNEL) コマンド 242

T

TCPIPSERVICE オプション
 INQUIRE URIMAP コマンド 127, 156
TCPIPSERVICE 属性
 URIMAP 定義 142
TCPIPSERVICE リソース定義
 仮想ホスティング 114
 サーバー HTTP 処理 132
 持続接続 112
TEMPLATENAME オプション
 INQUIRE URIMAP コマンド 156
TEMPLATENAME 属性
 URIMAP 定義 142
TERMINAL オプション
 START TRANSID (CHANNEL) コマンド 242
TLS (Transport Layer Security) 181
TOCHANNEL オプション
 MOVE CONTAINER (CHANNEL) コマンド 237
TOLENGTH オプション
 WEB CONVERSE コマンド 99
TRANSACTION オプション
 INQUIRE URIMAP コマンド 157
TRANSACTION 属性
 URIMAP 定義 143
TRANSID オプション
 START TRANSID (CHANNEL) コマンド 242
Transport Layer Security プロトコル 181
TXSeries 371
TYPE オプション
 WEB RECEIVE コマンド (サーバー) 125

U

URIMAP CVDA 値
 EXTRACT STATISTICS コマンド 336
URIMAP オプション
 INQUIRE URIMAP コマンド 154
 INQUIRE WEBSERVICE コマンド 60
 WEB EXTRACT コマンド 151
 WEB OPEN コマンド 82, 90, 97

URIMAP 属性
 URIMAP 定義 144
URIMAP 定義
 ANALYZER 属性 136
 CERTIFICATE 属性 137
 CHARACTERSET 属性 137
 CIPHERS 属性 137
 CONVERTER 属性 137
 HFSFILE 属性 138
 HOST 属性 139
 HOSTCODEPAGE 属性 139
 LOCATION 属性 139
 MEDIATYPE 属性 140
 PATH 属性 140
 PIPELINE 属性 141
 PROGRAM 属性 141
 REDIRECTTYPE 属性 141
 SCHEME 属性 142
 STATUS 属性 142
 TCPIPSERVICE 属性 142
 TEMPLATENAME 属性 142
 TRANSACTION 属性 143
 URIMAP 属性 144
 USAGE 属性 144
 USERID 属性 144
 WEBSERVICE 属性 144
URIMAP リソース定義 134
 仮想ホスティング 114
 クライアント HTTP 要求 75
 サーバー HTTP 処理 132
 属性 135
URL オプション
 WEB PARSE URL コマンド 147
URLLENGTH オプション
 WEB PARSE URL コマンド 147
USAGE オプション
 INQUIRE URIMAP コマンド 157
USAGE 属性
 URIMAP 定義 144
USERID オプション
 INQUIRE URIMAP コマンド 157
 START TRANSID (CHANNEL) コマンド 242
USERID 属性
 URIMAP 定義 144

V

VALIDATIONST オプション
 INQUIRE WEBSERVICE コマンド 60, 61
VERSIONLEN オプション
 WEB EXTRACT コマンド 151

W

WEB CLOSE コマンド 83
WEB CONVERSE コマンド 93
WEB EXTRACT コマンド 147
WEB OPEN コマンド 79
WEB PARSE URL コマンド 145
WEB RECEIVE コマンド (クライアント) 90
WEB RECEIVE コマンド (サーバー) 123
WEB SEND コマンド (クライアント) 84
WEB SEND コマンド (サーバー) 118
Web エラー・プログラム 164
Web サービス
 概要 13
Web サービス・アシスタント 31
Web 対応アプリケーション・プログラム 132
Web 非対応アプリケーション・プログラム 132
Web ユーザー・インターフェース 326
 画面設計 318
 結果セット警告カウント 313
 新機能 300
 新規ビュー
 機能シップ 263
 チャンネル使用 263
 パイプライン 71
 パイプライン 定義 71
 ホスト 175
 URI マッピング定義 175
 URI マップ 175
 URI マップ・グローバル 175
 Web サービス 71
 Web サービス定義 71
 新規メッセージ 324
 選択リスト 316
 バッチ・リポジトリ更新ジョブ 326
 フィルター確認 315
 変更されたビュー
 クロックとタイミング 193
 処理要求 71
 端末定義 361
 プログラム 284, 368
 プログラム定義 284, 368
 文書テンプレート 175
 文書テンプレート定義 175
 CICS 領域 193
 CorbaServer 193
 CorbaServer 定義 193
 CPU と TCB 情報 284
 ISC/MRO 接続 263
 TCP/IP グローバル状況 193
 TCP/IP サービス 193
 TCP/IP サービス定義 175, 193

Web ユーザー・インターフェース (続き)
 変更されたビュー (続き)
 TCP/IP 使用 175
 ユーザーのお気に入り 302, 304
 ユーザー・グループ・プロファイル 307, 308
 2 つのカラム詳細ビュー 321
 CICS リソース定義 310
 FILTERSTYLE 初期設定パラメーター 320
WEBSERVICE CVDA 値
 EXTRACT STATISTICS コマンド 336
WEBSERVICE オプション
 INQUIRE URIMAP コマンド 157
 INQUIRE WEBSERVICE コマンド 60, 61
WEBSERVICE 属性
 URIMAP 定義 144
WEBSERVICE リソース定義 47
WSBIND オプション
 INQUIRE WEBSERVICE コマンド 61
WSDIR 属性
 PIPELINE 定義 46
WSDL
 およびアプリケーション・データ構造 24
 言語構造への変換 38
WSDLFILE オプション
 INQUIRE WEBSERVICE コマンド 61

X

XWBOPEN ユーザー出口 101
XWBSNDO ユーザー出口 103

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

本書には、技術的に正確でない記述や誤植があります。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 United Kingdom Laboratories, MP151,
Hursley Park, Winchester, Hampshire,
England, SO21 2JN

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。

商標

以下は、IBM Corporation の商標です。

AIX	MQSeries	SupportPac
Candle	MVS	System/390
CICS	OMEGAMON	Tivoli
CICSplex	OMEGAMON II	TXSeries
DB2 Universal Database	OS/390	VisualAge
ESCON	Parallel Sysplex	VSE/ESA
IBM	RACF	WebSphere
IMS	Rational	z/OS
Language Environment	SecureWay	zSeries

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. 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。

Microsoft および Windows 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。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。



プログラム番号: 5655-M15

GD88-6377-00



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12

Spine Information:

CICS Transaction Server for z/OS リリース・ガイド