

CICS Transaction Server for z/OS



发行指南

版本 3 发行版 1

CICS Transaction Server for z/OS



发行指南

版本 3 发行版 1

注意！

在使用本资料及其支持的产品之前，请确保阅读第 351 页的『声明』中的一般信息。

第二版（2005 年 3 月）

此修订版适用于 CICS Transaction Server for z/OS V3R1（产品号 5655-M15）以及所有后续发行版和修订版，除非在新修订版中另行说明。

此修订版替换 G152-0637-00

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

目录

前言	xi
本书的内容	xi
本书的读者	xi
您需要了解以理解本书的信息	xi
有关术语的注意事项	xi
语法表示法	xi

第 1 部分 CICS Transaction Server for z/OS V3R1 摘要 1

第 1 章 CICS Transaction Server for z/OS V3R1的摘要	3
CICS 集成	3
应用程序转换	4
企业管理	5
杂项更改	6
废弃的功能	6
CICS 信息中心	7

第 2 部分 CICS 集成 9

第 2 章 CICS 中的 Web service	11
Web service 可以如何帮助您的业务	11
Web service 术语	11
需求	13
CICS 如何支持 Web service	13
消息处理程序和管道	14
SOAP 消息和应用程序数据结构	18
WSDL 和应用程序数据结构	19
Web service 绑定文件	21
规划在 CICS 中使用 Web service	22
规划服务提供者应用程序	22
规划服务请求者应用程序	24
CICS Web services 助手	25
DFHLS2WS: 高级语言到 WSDL 的转换	26
DFHWS2LS: WSDL 到高级语言的转换	29
对 CICS 外部的更改	33
对资源定义的更改	33
对应用程序编程接口的更改	39
对系统编程接口的更改	43
对 CEMT 的更改	49
对 JCICS API 的更改	52
对 CICS 提供的事务的更改	52
对统计信息的更改	53
对样本程序的更改	53
对 CICS 实用程序的更改	53
对问题确定的更改	53
安全性	54
迁移	54
迁移现有函数	54
共存	55

CICSplex SM 支持.	55
对 CICSplex SM 应用程序编程接口的更改.	55
对 CICSplex SM Web 用户界面的更改.	57
第 3 章 支持 CICS 应用程序发出的 HTTP 客户机请求.	61
支持 CICS 应用程序发出的 HTTP 客户机请求的优点.	61
需求.	61
作为 HTTP 客户机的 CICS 的 HTTP 请求和响应处理.	61
会话标记.	62
对 CICS 外部的更改.	63
对资源定义的更改.	63
对应用程序编程接口 (HTTP 客户机请求) 的更改.	63
对 JCICS API 的更改.	81
对全局用户出口的更改.	81
对监控的更改.	84
第 4 章 升级到 HTTP/1.1 的 CICS Web 支持.	85
CICS Web 支持升级到 HTTP/1.1 的优点.	86
需求.	86
新的 HTTP 功能.	86
分块的传输编码.	87
管道传送.	88
持续连接.	88
虚拟主管.	89
对 CICS 外部的更改.	90
对资源定义的更改.	90
对应用程序编程接口 (HTTP/1.1 支持) 的更改.	92
对系统编程接口的更改.	99
对 CEMT 的更改.	101
对统计信息的更改.	101
第 5 章 对 CICS Web 支持的一般增强.	103
CICS Web 支持增强的优点.	103
术语.	104
需求.	105
作为 HTTP 服务器的 CICS 的 HTTP 请求和响应处理.	105
CICS Web 支持程序中的 Unicode UTF-8 和 UTF-16 代码页转换.	106
处理 HTTP 日期和时间戳记格式.	106
对 CICS 外部的更改.	107
对资源定义的更改.	107
对应用程序编程接口 (一般 CICS Web 支持增强) 的更改.	114
对系统编程接口的更改.	122
对 CEMT 的更改.	127
对 CICS 提供的事务的更改.	128
对用户可替代程序的更改.	129
对统计信息的更改.	131
对 CICS 实用程序的更改.	131
对问题确定的更改.	131
安全性.	132
迁移.	132
迁移现有 CICS Web 支持应用程序.	132
迁移到新的 CICS Web 支持功能.	134

CICSplex SM 支持	134
对 CICSplex SM 最终用户界面视图的更改	135
对 CICSplex SM 应用程序编程接口的更改	135
对 CICSplex SM Web 用户界面的更改	140
第 6 章 对因特网安全性的改进	143
对因特网安全性改进的优点	143
安全术语	143
需求	144
传输层安全性协议	144
对 SSL 性能的改进	145
使用证书撤回列表	145
SSL 高速缓存	146
定制加密协商	146
对 CICS 外部的更改	147
对系统初始化参数的更改	147
对资源定义的更改	148
对系统编程接口的更改	149
对 CICS 提供的事务的更改	150
对全局用户出口的更改	151
对监控的更改	151
对统计信息的更改	151
对 CICS 实用程序的更改	151
对问题确定的更改	152
安全性	152
迁移	152
迁移现有函数	152
迁移到新的函数	152
共存	152
CICSplex SM 支持	152
对 CICSplex SM 最终用户界面视图的更改	152
对 CICSplex SM 应用程序编程接口的更改	153
对 CICSplex SM Web 用户界面的更改	155

第 3 部分 应用程序转换 157

第 7 章 增强的程序间数据传输: 作为现代 COMMAREA 的通道	159
通道的优点	159
术语	160
通道: 快速启动	160
容器和通道	160
基本示例	160
使用通道: 一些典型方案	163
一个通道, 一个程序	163
一个通道, 几个程序 (组件)	164
几个通道, 一个组件	164
多个交互式组件	165
创建通道	166
当前通道	166
带 LINK 命令的当前通道示例	166
带 XCTL 命令的当前通道示例	169
当前通道: START 和 RETURN 命令	170

通道的作用域	171
带 LINK 命令的范围示例	171
带 LINK 和 XCTL 命令的范围示例	173
发现哪些容器传递给程序	175
发现从链接返回的容器	175
CICS只读容器	175
设计通道: 最佳实践	176
构造和使用通道: 示例	177
通道和 BTS 活动	178
上下文	179
使用来自 JCICS 的通道	180
在 JCICS 中创建通道和容器	180
将数据放入容器	181
将通道传递给另一个程序或任务	181
接收当前通道	181
从容器获取数据	182
浏览当前通道	182
JCICS 示例	182
与通道一起动态路由	183
数据转换	184
为何需要进行数据转换?	184
带通道的数据转换	184
需求	187
对 CICS 外部的更改	187
对应用程序编程接口的更改	187
对 JCICS API 的更改	200
对全局用户出口的更改	200
对任务相关的用户出口的更改	201
对用户可替代程序的更改	201
对监控的更改	203
对统计信息的更改	204
对样本程序的更改	205
对问题确定的更改	205
从 COMMAREAs 迁移到通道	208
现有功能的迁移	208
迁移到新功能	208
共存	211
CICSplex SM 支持	211
对 CICSplex SM 应用程序编程接口的更改	211
对 CICSplex SM Web 用户界面的更改	213
第 8 章 OPENAPI 支持	215
OPENAPI 支持的优点	216
需求	216
对 CICS 外部的更改	216
对系统初始化参数的更改	216
对资源定义的更改	217
对应用程序编程接口的更改	217
对系统编程接口的更改	218
对 CEMT 的更改	219
第 9 章 XPLink 支持	221

XPLink 支持的优点	222
需求	222
对 CICS 外部的更改	222
对安装的更改	222
对系统初始化参数的更改	223
对资源定义的更改	223
对应用程序编程接口的更改	223
对系统编程接口的更改	224
对 CEMT 的更改	224
对全局用户出口的更改	225
对用户可替代程序的更改	225
对监控的更改	226
对统计信息的更改	227
迁移	228
现有功能的迁移	228
迁移到新的功能	228
CICSplex SM 支持	228
对 CICSplex SM 应用程序编程接口的更改	228
对 CICSplex SM Web 用户界面的更改	229
第 10 章 对符合汇编程序 MAIN 程序的 Language Environment 的支持	231
对符合汇编程序 MAIN 程序的 Language Environment 的支持的优点	231
需求	231
对 CICS 外部的更改	231
对资源定义的更改	231
对应用程序编程接口的更改	232
对全局用户出口的更改	240
对任务相关的用户出口的更改	241

第 4 部分 企业管理 243

第 11 章 CICSplex SM Web 用户界面增强	245
CICSplex SM Web 用户界面增强的优点	246
需求	247
对 CICSplex SM 的更改	247
用户收藏夹	247
用户组概要文件	251
业务应用程序服务重新设计	254
记录计数警告	257
过滤器确认	258
动态选择列表	259
改进的屏幕设计	260
对 CICSplex SM API 的更改	265
消息	265
第 12 章 CICSplex SM 批处理资源库更新工具的增强	267
增强 CICSplex SM 批处理资源库更新工具的益处	267
需求	267
批处理实用程序	267
对 CICSplex SM 应用程序编程接口的更改	267
对 CICSplex SM Web 用户界面的更改	268
消息	268

安全性	268
迁移	268

第 5 部分 杂项更改 269

第 13 章 新安装过程	271
新安装过程的优点	271
第 14 章 EXTRACT STATISTICS 命令	273
EXTRACT STATISTICS 命令的优点	273
对 CICS 外部的更改	273
对系统编程接口的更改	273
第 15 章 支持大小写混合密码	277
第 16 章 代码页转换更改	279
代码页转换更改的优点	279
术语	279
需求	280
对 CICS 外部的更改	280
对安装的更改	280
对系统初始化参数的更改	280
对应用程序编程的更改	281
对 CICS 实用程序的更改	282
对问题确定的更改	282
第 17 章 缺省代码页的简化定义	283
在数据转换模板中改进代码页缺省值的优势	283
需求	283
对 CICS 外部的更改	283
对系统初始化参数的更改	283
对用户可替代程序的更改	284
第 18 章 64 位寻址能力更改	285
64 位寻址能力更改的优点	285
需求	285
对 CICS 外部的更改	285
对 CICS 实用程序的更改	285
对问题确定的更改	285
第 19 章 支持撤销的用户标识	287

第 6 部分 废弃的功能 289

第 20 章 除去对 OS/VS COBOL 程序的运行时支持	291
第 21 章 对 BTAM 和 TCAM 支持的更改	293
收回 BTAM 支持	293
TCAM 的 CICS 支持中的更改	293
对 CICS 外部的更改	294
对系统初始化参数的更改	294
对资源定义的更改	294
对应用程序编程接口的更改	294

对全局用户出口的更改	295
对用户可替代程序的更改	295
对样本程序的更改	295
迁移	295
共存	295
第 22 章 除去对 1 个字节的控制台标识的支持	297
CICSplex SM 支持	297
对 CICSplex SM 应用程序编程接口的更改	297
对 CICSplex SM 最终用户界面视图的更改	297
对 CICSplex SM Web 用户界面的更改	297
消息	297
第 23 章 撤消 CICS TS 的 CICS 连接器	299
第 24 章 除去对 Java 程序对象和热缓冲池的运行时支持	301
对 CICS 外部的更改	301
对系统初始化参数的更改	301
对资源定义的更改	301
对应用程序编程接口的更改	302
对系统编程接口的更改	302
对 CEMT 的更改	302
对全局用户出口的更改	302
对出口编程接口 (XPI) 的更改	302
对用户可替代程序的更改	302
对监控的更改	303
对统计信息的更改	303
对问题确定的更改	303
CICSplexSM 支持	303
对 CICSplex SM 最终用户界面视图的更改	303
对 CICSplex SM 应用程序编程接口的更改	303
对 CICSplex SM Web 用户界面的更改	304
第 25 章 除去了对 Windows 远程 MAS 的 CICSplex SM 支持	305
第 26 章 除去 CICS 事务亲缘关系实用程序	307
<hr/>	
第 7 部分 一般信息	309
第 27 章 CICS 操作环境	311
硬件需求	311
软件需求	311
对 CICS 工具和相关产品的支持	313
兼容性	315
第 28 章 线程安全应用程序编程接口命令	317
第 29 章 高级语言支持	319
<hr/>	
第 8 部分 出版物	325
第 30 章 Eclipse 信息中心	327
Eclipse 信息中心的优点	327

术语	328
需求	328
“新增内容”部分	329
信息指南	329
学习路径	329
信息中心中搜索的技术	329
浏览信息中心	331
将主题制作成书签	332
用户首选项	332
第 31 章 CICS Transaction Server for z/OS库	333
作为硬拷贝提供的书籍	333
仅 PDF 书籍	333
CICS 书籍, 用于CICS Transaction Server for z/OS	333
CICSplex SM 书籍, 用于CICS Transaction Server for z/OS	334
CICS系列书籍	335
许可出版物	335
可访问性	337
索引	339
声明	351
商标	352
将您的意见发送给 IBM	353

前言

本书的内容

本书提供有关 CICS® Transaction Server for z/OS®V3R1 中新增和已更改功能的信息。它给出对参考信息的更改概述，并为您指出提供更详细参考信息的手册。

本书中给出的编程接口信息确定为仅显示哪些内容是自 CICS TS 的前发行版以来新增和已更改的内容，并重点描述新功能的优点。要获得编程接口信息，请阅读下列出版物中的编程接口主要来源，以关联信息：

- *CICS Application Programming Reference*
- *CICS System Programming Reference*
- *CICS Customization Guide*
- *CICS External Interfaces Guide*
- *CICSplex SM Application Programming Guide*
- *CICSplex SM Application Programming Reference*

本书的读者

本书适用于那些负责以下用户任务的读者：

- 评估和计划
- 系统管理
- 编程
- 定制

您需要了解以理解本书的信息

本书假设您熟悉 CICS 和 CICSplex SM，作为系统管理员，或者作为系统或应用程序员。

有关术语的注意事项

当在本书中没有任何限定的使用术语“CICS”，则它是指 IBM CICS TS 的 CICS 元素。


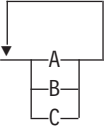

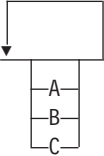
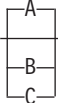
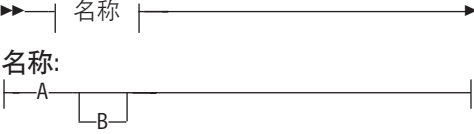

“CICSplex SM”用于 IBM CICS TS 的 CICSplex System Manager 元素。

“MVS”用于操作系统，其是 z/OS 的基本元素。

语法表示法

语法表示法指定您可在 CICS 命令、资源定义和很多其他内容上指定的选项或属性的允许组合。

用于语法表示法的约定:

表示法	说明
	<p>表示一组必需的选择项。您必须指定显示的某个（仅一个）值。</p>
	<p>表示一组必需的选择项。您必须至少指定一个显示的值。您可按任何顺序指定多个值。</p>
	<p>表示一组可选的选择项。您可指定不选，或选择一个显示的值。</p>
	<p>表示一组可选的选择项。您可按任何顺序指定不选，选择一个或选择多个显示的值。</p>
	<p>表示一组可选的选择项。您可指定不选，或选择一个显示的值。A 是缺省值，如果您未指定任何值，则使用缺省值。</p>
	<p>对语法表示法指定部分的引用。</p>
	<p>A= 表示应该完全按显示输入的字符。 <i>value</i> 表示一个变量，您应该为其指定适当的值。</p>

第 1 部分 CICS Transaction Server for z/OS V3R1 摘要

此部分包含对 CICS Transaction Server for z/OS V3R1 中主要新功能的简要概述。

第 1 章 CICS Transaction Server for z/OS V3R1的摘要

CICS Transaction Server for z/OS V3R1中主要的新的和已更改功能将在 CICS 集成、应用程序转换和企业管理，这三个主题中说明。CICS TS 3.1 还包括这三个主题作用域之外的杂项更改，以及在此作用域中停止或减少的很多功能。

CICS 集成

CICS TS 3.1 通过使用广泛采用的应用程序编程接口和标准协议，提供了一系列新的和增强的功能，从而可以复用更广泛的按需应变方案内现有的 CICS 应用程序。这些包括：

对 Web Service 的支持

对 Web Service 的支持完全集成到CICS TS 3.1中。在此发行版中，CICS 可担当服务提供者和服务请求者两个角色，其中使用 Web Service 定义语言 (WSDL) 定义服务。

作为 CICS TS V3.1 的一部分提供的基础结构包括与 WS-AtomicTransaction 规范兼容的分布式事务协调能力。

对 Web Service 的支持包括 CICS Web Service 助手，这是一个批处理实用程序，帮助您

- 将现有的 CICS 应用程序转换为 Web Service
- 并启用 CICS 应用程序，以使用外部提供者提供的 Web Service。

此助手可从简单语言结构创建 WSDL 文档，或从现有 WSDL 文档创建语言结构，并支持 COBOL、C/C++ 和 PL/I。它还生成信息，用于支持 SOAP 消息到容器和 COMMAREA 的自动运行时转换，反之亦然。

要获得更多信息，请参阅第 11 页的第 2 章，『CICS 中的 Web service』。

支持 CICS 应用程序发出的 HTTP 客户机请求

在CICS TS 3.1中，CICS担当 HTTP 客户机的能力完全集成到 CICS Web 支持。您的应用程序现在使用 EXEC CICS 命令打开到服务器的连接、发出 HTTP 请求，并接收响应。

要获得更多信息，请参阅第 61 页的第 3 章，『支持 CICS 应用程序发出的 HTTP 客户机请求』。

对 HTTP/1.1 的支持

CICS Web 支持有条件地符合由 RFC 2616 定义的 HTTP/1.1 规范，*Hypertext Transfer Protocol-- HTTP/1.1*。

CICS TS 3.1中支持的新功能包括分块传送编码、管道传送和持续连接。

要获得更多信息，请参阅第 85 页的第 4 章，『升级到 HTTP/1.1 的 CICS Web 支持』。

对 CICS Web 支持的其他增强

对 CICS TS 3.1 中 CICS Web 支持的其他增强包括：

- URIMAP 资源，其在 CICS 作为 HTTP 服务器时，给出用于处理 HTTP 请求和响应的改进功能

- 对 CICS Web 支持代码页的方法改进（包括对 UTF-8 和 UTF-16 代码页的支持）
- 为 HTTP 时间和日期格式提供支持的新 API 命令。

要获得更多信息，请参阅第 103 页的第 5 章，『对 CICS Web 支持的一般增强』。

对 SSL 支持的改进

对安全套接字层（SSL）的支持在 CICS TS 3.1 中有所增强：

- CICS 现在支持传输层安全性（TLS）1.0 协议以及 SSL 3.0，这允许您使用新的提供 128 位和 256 位加密的 AES 密码套件。
- 已增加可以在系统中同时使用的同步 SSL 连接数，以实现更大的吞吐量。
- 更为灵活地控制客户机和服务器之间的加密协商。现在，您可以为与特殊用户协商指定 CICS 中的最低加密级别和最高加密级别。
- 与客户机协商时，CICS 现在可以根据证书撤回列表检查所有证书。任何使用已撤回证书的连接都立即关闭。
- 您可以指定是否要通过使用 SSL 高速缓存跨越综合系统（sysplex）共享会话标识。如果客户机先前已与 CICS 进行协商，CICS 将执行部分 SSL 握手。如果跨越很多 CICS 区域共享高速缓存，这将改进 SSL 协商和连接吞吐量的性能。

要获得更多信息，请参阅第 143 页的第 6 章，『对因特网安全性的改进』。

应用程序转换

CICS Transaction Server for z/OS V3R1 提供一系列新功能，使得您能够使用当代的编程语言、构造和工具，增强现有应用程序，并构建新的应用程序。这些包括：

增强的程序间数据传输：作为现代 COMMAREA 的通道

在 CICS TS 3.1 中，通道和容器提供程序间传送数据的改进方法，在数量上远远超出应用于 COMMAREA 的 32KB 限制。

- 容器是为在程序间传递信息而设计的命名数据块。
- 集合中分组在一起的容器称为通道。通道是用于在 CICS 程序之间交换数据的标准机制，并且类似于参数列表。通道可以在 LINK、START、XCTL 和 RETURN 命令中使用，并带有本地和远程事务。对可添加到通道的容器数没有限制，并且每个容器的大小只受可用存储量的限制。

通道和容器模型比传统的 COMMAREA 模型具有多项优势。这些包括：

- 不存在 32KB 的大小限制。
- 与作为单块数据的 COMMAREA 不同，具有多个容器的通道可表示结构化数据。

要获得更多信息，请参阅第 159 页的第 7 章，『增强的程序间数据传输：作为现代 COMMAREA 的通道』。

OPENAPI 支持

CICS 通过提供对 OPENAPI 应用程序的支持来扩展开放式事务环境（OTE）功

能的使用。在执行此操作前，OPENAPI 功能只能用于于用户出口（TRUE）相关的任务。OPENAPI 程序的使用允许应用程序工作负载从 QR TCB 移出到多个开放式 TCB 上。

要获得更多信息，请参阅第 215 页的第 8 章，『OPENAPI 支持』。

XPLINK 对 C 和 C++ 的支持

CICS TS 3.1中已引入对 C 和 C++ 的新支持，把用这些语言写的应用程序的性能带到与通过 COBOL、PL/I 或 Assembler 获得的性能可比较的级别。性能改进由 z/OS 的“额外性能链接”（XPLINK）功能提供，该功能为堆栈扩展提供高性能子例程链接机制和守护页，这样就能高度优化执行路径的长度。

应用程序从这些改进得到的最大好处是：

- 在 CICS 开放式事务环境（OTE）中运行
- 写入线程安全标准，并仅使用线程安全的 CICS 命令

XPLINK 的使用将启用与 C 和 C++ 包含在一起的最新编译器和优化技术以供开发。特别是，您可使用动态装入库（DLL）获得对 C 和 C++ 代码的更好的复用，这些 DLL 在 CICS 环境中使用 XPLINK 选项编译。

要获得更多信息，请参阅第 159 页的第 7 章，『增强的程序间数据传输：作为现代 COMMAREA 的通道』。

对 Language Environment 的支持启用汇编程序主程序

CICS TS 3.1支持用汇编程序写的主程序，以及启用 Language Environment 的主程序。此支持扩展 Language Environment 使用的可用性，并使调试器支持与汇编程序一起使用。

要获得更多信息，请参阅第 231 页的第 10 章，『对符合汇编程序 MAIN 程序的 Language Environment 的支持』。

企业管理

CICS TS 3.1 的 CICSplex[®] SM 元素提供新的功能，使您能够使用现代用户接口有效管理大量运行时配置，以便您可满足需求服务级别目标。这些包括：

对 CICSplex SMWeb 用户界面的增强

CICSplex SM Web 用户界面已改进，使它的功能更强大，用途更广泛。Web 用户界面现在的功能等价与 CICSplex SM TSO 最终用户界面，而且现在是访问 CICSplex SM 的主要方法。

要获得更多信息，请参阅第 245 页的第 11 章，『CICSplex SM Web 用户界面增强』。

CICSplex SM批处理资源库更新工具的增强

引入了新的工具，它们提供了能替换现有 TSO EUI 命令的方法，向指定的 CICSplex SM 资源库提交批处理更新。已引入的可替换 BATCHREP 工具是：

- CPSM API 可用于引用对象的新 BATCHREP 资源表
- 新资源表的 Web 用户界面支持
- 提供 BATCHREP 工具的批处理实用程序

要获得更多信息，请参阅第 267 页的第 12 章，『CICSplex SM 批处理资源库更新工具的增强』。

杂项更改

CICS Transaction Server for z/OS V3R1中其他新的和已更改的功能包括:

EXTRACT STATISTICS 命令

新的 SPI 命令 EXTRACT STATISTICS 为 URIMAP、PIPELINE 和 WEBSERVICE 资源执行等价于 COLLECT STATISTICS 的功能。

要获得更多信息, 请参阅第 273 页的第 14 章, 『EXTRACT STATISTICS 命令』。

支持大小写混合密码

当与 CICS 协同使用的安全管理器支持使用大小写混合密码时, CICS TS 3.1 在它传递给安全管理器前, 不会将密码转换为大写。

要获得更多信息, 请参阅第 277 页的第 15 章, 『支持大小写混合密码』。

缺省代码页的简化定义

在CICS TS 3.1中, DFHCNV 数据转换使用的缺省客户机或服务器代码页可在系统初始化参数中定义。

要获得更多信息, 请参阅第 283 页的第 17 章, 『缺省代码页的简化定义』。

64 位寻址能力

CICS 不支持 64 位寻址执行, 但是程序可使用仅当 CICS 在 64 位体系结构机器上运行时可用的地址上的存储。CICS 异常终止捕获机制已更改, 以致捕获到完整 64 位通用寄存器的内容。

要获得更多信息, 请参阅第 285 页的第 18 章, 『64 位寻址能力更改』。

支持撤销的用户标识

当发出 EXEC CICS VERIFY PASSWORD 命令时, CICS 现在提供用户标识或用户组连接的撤销状态。

废弃的功能

CICS TS 3.1 的作用域中废弃或减少了一些原先在 CICS Transaction Server for z/OS V2 中支持的功能。它们包括:

对 OS/VS COBOL 程序的运行时支持

在CICS Transaction Server for z/OS V2中具有运行时支持的 OS/VS COBOL 程序无法在CICS TS 3.1下运行。

要获得更多信息, 请参阅第 291 页的第 20 章, 『除去对 OS/VS COBOL 程序的运行时支持』。

对 BTAM 和 TCAM 的支持

CICS对基本远程通信访问方法 (BTAM) 的支持在CICS TS 3.1中停止。对远程通信访问方法 (TCAM) 的支持限制为对 DCB 接口的间接支持。

要获得更多信息, 请参阅第 293 页的第 21 章, 『对 BTAM 和 TCAM 支持的更改』。

对 1 位字节控制台标识的支持

除去了对 1 个字节的控制台标识的支持。您可以在 TERMINAL 定义上使用 CONSNAME(*name*) 属性定义控制台。

要获得更多信息，请参阅第 297 页的第 22 章，『除去对 1 个字节的控制台标识的支持』。

CICSTS 的 CICS连接器

已撤销 CICS Transaction Server for z/OS V2R1中所引入对 CICS Connector for CICS TS 的支持。

要获得更多信息，请参阅第 299 页的第 23 章，『撤消 CICS TS 的 CICS 连接器』。

对 Java™ 程序对象和 Hot-Pooling 的支持

撤消对 Java 程序对象和 Hot-Pooling (HPJ) 的运行时支持

要获得更多信息，请参阅第 301 页的第 24 章，『除去对 Java 程序对象和热缓冲池的运行时支持』。

CICSplex SM 支持 Windows®远程 MAS

要获得更多信息，请参阅第 305 页的第 25 章，『除去了对 Windows 远程 MAS 的 CICSplex SM 支持』。

撤消 CICS 事务相似性实用程序

CICS 事务亲缘关系实用程序不再与 CICS 一起提供。CICS Interdependency Analyzer 现在提供检测事务相似性的功能，这是更成熟的工具。

要获得更多信息，请参阅第 307 页的第 26 章，『除去 CICS 事务亲缘关系实用程序』。

CICS 信息中心

在CICS TS 3.1中，CICS 信息中心在 WebSphere® Studio WorkBench 用户帮助系统内运行，Eclipse 框架包含很多文档插件，这些文档插件构成了信息中心。现在，信息中心的新外观（特别是欢迎页面）与其他的产品信息中心保持了一致性。

要获得更多信息，请参阅第 327 页的第 30 章，『Eclipse 信息中心』

第 2 部分 CICS 集成

CICS Transaction Server for z/OS V3R1 通过使用广泛采用的应用程序编程接口和标准协议，提供了一系列新的和增强的功能，从而可以复用更广泛的按需应变方案内现有的 CICS 应用程序。

第 2 章 CICS 中的 Web service

对 Web service 的支持完全集成到 CICS Transaction Server for z/OS V3R1 中。在本发行版中，CICS 应用程序可以担当服务提供者和服务请求者的角色，而那些服务是用 Web Service 描述语言 (WSDL) 定义的。

作为 CICS TS V3.1 的一部分提供的基础结构包含遵从 WS-AtomicTransaction 规范的分布式事务协调能力。

对 Web service 的支持包含 CICS Web service 助手，这是一个批处理实用程序，可以用它来帮助您

- 将现有的 CICS 应用程序转换到 Web service
- 并且使得 CICS 应用程序能够使用外部提供者提供的 Web service。

助手可以从简单语言结构创建 WSDL 文档，也可以从现有的 WSDL 文档创建语言结构，并且支持 COBOL、C/C++ 和 PL/I。它还生成用于使 SOAP 消息与容器和 COMMAREA 相互进行自动运行时转换的消息。

Web service 可以如何帮助您的业务

Web service 是通过 Web 部署业务功能并提供对该业务功能访问的一种技术。Web service 使应用程序可以比以前更快、更容易、更廉价地集成。

Web service 可以通过以下方面帮助您的业务：

- 降低完成业务的成本
- 使部署解决方案可以更快
- 开发新的机会。

完成所有这些事情的关键是公用的程序到程序通信模型，它是基于现有及新兴标准（如 HTTP、XML、SOAP 和 WSDL）而构建的。

CICS 为 Web service 提供的支持使您的现有应用程序能在作最少量的重新编程情况下以新的方法部署。

Web service 术语

扩展标记语言 (Extended Markup Language, XML)

一个文档标记的标准，它使用一般语法以简单的人们可读的标记来标记数据。该标准是由万维网联合组织 (W3C) 认可的。

初始 SOAP 发送方 (Initial SOAP sender)

SOAP 发送方，它在 SOAP 消息路径的起始点产生 SOAP 消息。

服务提供者 (Service provider)

软件的集合，它提供 Web service。

服务提供者应用程序 (Service provider application)

一个应用程序，它在服务提供者中使用。通常，服务提供者应用程序提供服务提供者的业务逻辑组件。

服务请求者 (Service requester)

软件的集合，它负责从服务提供者请求 Web service。

服务请求者应用程序 (Service requester application)

一个应用程序，它在服务请求者中使用。通常，服务提供者应用程序提供服务请求者的业务逻辑组件。

简单对象访问协议 (Simple Object Access Protocol)

请参阅 SOAP。

SOAP 以前是简单对象访问协议词组的缩写。用于在分散的分布式环境中交换信息的轻量级协议。它是基于 XML 的协议，由三部分组成：

- 定义用于描述消息中的内容以及如何处理消息的框架的封包。
- 用于表示应用程序定义的数据类型实例的一组编码规则。
- 用于表示远程过程调用和响应的约定。

SOAP 可以与其他协议一起使用，如 HTTP。

SOAP 调解者 (SOAP intermediary)

一个 SOAP 节点，它是 SOAP 接收方和 SOAP 发送方并从 SOAP 消息中可命中目标。它处理命中它的 SOAP 头块，并向最终 SOAP 接收方执行转发 SOAP 消息。

SOAP 消息路径 (SOAP message path)

SOAP 节点的集合，单个 SOAP 消息通过它来传递。这包括初始 SOAP 发送方、零个或多个 SOAP 调解者以及最终 SOAP 接收方。

SOAP 节点 (SOAP node)

处理逻辑，它对 SOAP 消息进行操作。

SOAP 接收方 (SOAP receiver)

一个 SOAP 节点，它接受 SOAP 消息。

SOAP 发送方 (SOAP sender)

一个 SOAP 节点，它发送 SOAP 消息。

最终 SOAP 接收方 (Ultimate SOAP receiver)

SOAP 接收方，它是 SOAP 消息的最终目标。它负责处理 SOAP 主体和针对它的任何 SOAP 头块的内容。

UDDI 统一描述、发现和集成**统一描述、发现和集成 (Universal Description, Discovery and Integration)**

统一描述、发现和集成 (UDDI) 是 Web service 的基于 Web 的分布式信息注册表的规范。UDDI 还是一组公开可访问的规范实现，它允许业务注册有关它们提供的 Web service 的信息，以使其他业务可以找到它们。

Web service

一个软件系统，它被设计为支持通过网络能共同操作的机器到机器交互。它有以机器可处理的格式描述的接口（特别是，Web Service 描述语言或 WSDL）。

Web service 绑定文件 (Web service binding file)

一个与 WEBSERVICE 资源关联的文件，它包含 CICS 在输入和输出消息以及应用程序数据结构之间映射数据时所使用的信息。

Web service 描述 (Web service description)

一个 XML 文档，服务提供者通过该文档与规范进行通信来为服务提供者调用 Web service。Web service 描述是用 Web Service 描述语言 (WSDL) 编写的。

Web Service 描述语言 (Web Service Description Language)

一个 XML 应用程序，它用于描述 Web service。它被设计为将服务提供的抽象功能的描述与服务的具体详细信息分离开来，如，如何以及在哪里提供功能。

WSDL Web Service 描述语言。

XML 可扩展标记语言。

XML 名称空间 (XML namespace)

一个由 URI 引用来标识的名称集合，它在 XML 文档中用作元素类型和属性名称。

XML 模式 (XML schema)

一个 XML 文档，它描述结构，并约束其他 XML 文档的内容。

XML 模式定义语言 (XML schema definition language)

一种用于编写 XML 模式的 XML 语法，由万维网联合组织 (W3C) 推荐。

需求

要支持此功能，没有特别的硬件或软件需求。

相关信息

第 311 页的第 27 章，『 CICS 操作环境 』

CICS 如何支持 Web service

CICS 支持在 Web service 环境中部署您的 CICS 应用程序的两种不同方法。一种方法使您在做最少量的编程之后即可进行快速部署；另一种方法使您使用为满足您特定需要而编写的代码完全灵活并彻底地控制您的 Web service。由一个或多个对 Web service 请求和响应进行操作的管道和消息处理程序组成的基础结构支持这两种方法。

当您在 Web service 环境中部署 CICS 应用程序时：

- 可以使用 CICS Web Service 助手来帮助您在进行最少量的编程之后即可部署应用程序。

例如，如果您要将现有的应用程序列为 Web service，则可以从高级语言数据结构开始，并生成 Web service 描述。或者，如果您要与现有的 Web service 通信，则可以从它的 Web service 描述开始并生成您可在程序中使用的高级语言结构。

CICS Web Service 助手还生成部署您的应用程序需要的 CICS 资源。并且当您的应用程序运行时，CICS 将您的应用程序数据转换为输出时的 SOAP 消息，并将 SOAP 消息转换回输入时的应用程序数据。

- 通过编写您自己的代码以在您的应用程序数据以及服务请求者和提供者之间的消息流之间映射，您可以完全控制您数据的处理。

例如，如果您要使用 Web service 基础结构中的非 SOAP 消息，则可以编写您自己的代码以在消息格式和您的应用程序所用的格式之间进行转换。

无论您执行哪种方法，您都可以使用您自己的消息处理程序对您的请求和响应消息执行额外处理，或使用帮助您处理 SOAP 消息而特别设计的 CICS 提供的消息处理程序。

消息处理程序和管道

消息处理程序是您可以在其中执行您自己的 Web service 请求和响应处理的程序。管道是按序执行的一组消息处理程序。

在管道的操作中有两个不同的阶段：

1. 请求阶段，此阶段期间 CICS 依次在管道中调用每个处理程序。在将控制权返回到 CICS 之前，每个消息处理程序都可以处理请求。
2. 它后跟响应阶段，在此阶段期间 CICS 再次依次调用每个处理程序，但是以反向的顺序。即，在请求阶段第一个调用的消息处理程序在响应阶段被最后调用。此阶段期间，每个消息处理程序可以处理响应。

不是每个请求都后跟响应的；某些应用程序使用从服务请求者到提供者的单向消息流。在此情况下，尽管不存在要处理的消息，但是在响应阶段期间依次调用每个处理程序。

图 1 显示三个消息处理程序的管道：

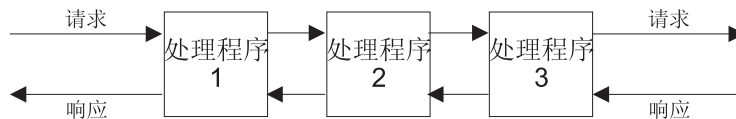


图 1. 一般 CICS 管道

在此示例中，用以下顺序执行处理程序：

在请求阶段

1. 处理程序 1
2. 处理程序 2
3. 处理程序 3

在响应阶段

1. 处理程序 3
2. 处理程序 2
3. 处理程序 1

在服务提供者中，通常在管道的最后处理程序（称为终端处理程序）中发生阶段之间的转换，该处理程序吸收请求并生成响应；在服务请求者中，当在服务提供者中处理请求时发生转换。但是，请求阶段中的消息处理程序可以强制执行到响应阶段的立即转换，且如果 CICS 检测到错误，则也可能发生立即转换。

消息处理程序可能修改消息，也可能让它保持不变。例如：

- 执行加密和解密的消息处理程序将在输入时接收到加密的消息，并将已解密的消息传递到下一个处理程序。在输出时，它将执行相反的操作：接收纯文本消息，并将加密的版本传递到下一个处理程序。
- 执行记录的消息处理程序将检查消息，并将相关的信息从该消息复制到日志。传递到下一个处理程序的消息是未更改的。

重要: 如果您熟悉 CICS TS 的 SOAP 功能, 则应该知道此发行版的 CICS 中管道的结构与此功能中所用的结构并不相同。

与传输相关的处理程序

CICS 支持在 Web Service 请求者和提供者之间使用两种传输机制。在某些情况下, 根据使用的是哪种传输机制, 您可能需要调用不同的消息处理程序。例如, 当您在外部网络上使用 HTTP 传输方法来通信时, 可能希望包括对您消息的某些部分执行加密的消息处理程序。但当您在安全内部网络上使用 MQ 传输时, 可能不需要加密。

要支持它, 您可以配置管道以指定仅在使用特定传输方法 (HTTP 或 MQ) 时才调用的处理程序。对于服务提供者, 您甚至可以更明确, 并指定仅在使用特殊指定的资源 (用于 HTTP 传输方法的 TCPIPSERVICE、用于 MQ 传输方法的 QUEUE) 时才调用的处理程序。

这在图 2 中有所描述:

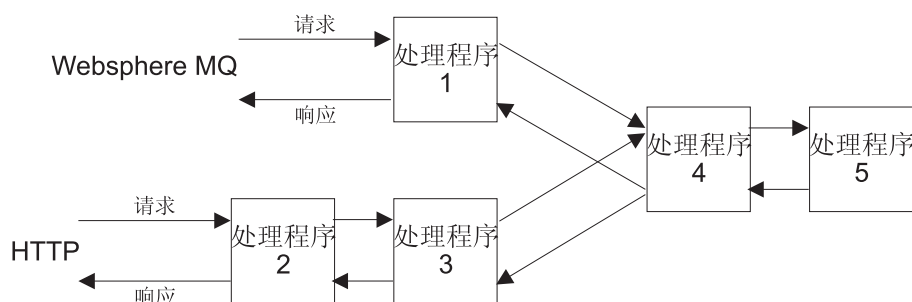


图 2. 使用与传输相关的处理程序的管道

在此示例中, 哪一个可以作为服务提供者:

- 为使用 MQ 传输方法的消息调用处理程序 1
- 为使用 HTTP 传输方法的消息调用处理程序 2 和 3
- 为所有消息调用处理程序 4 和 5
- 处理程序 5 是终端节点

中断流

处理请求期间, 消息处理程序可能决定不将消息传递到下一个处理程序, 但可能代之生成响应。消息的正常处理会中断, 且管道中的某些处理程序不被调用。例如, 假设图 3 中的处理程序 2 负责执行安全检查。

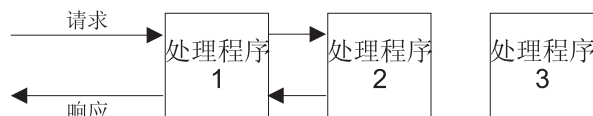


图 3. 中断管道流

如果请求不具有正确的安全凭证, 则处理程序 2 抑制请求 (而非将请求传递到处理程序 3), 并构造适当的响应。现在管道在响应阶段, 并当处理程序 2 将控制权返回到 CICS 时, 下一个调用的处理程序是处理程序 1, 且完全绕过处理程序 3。

仅当消息的发起方需要响应时，以此方法中断正常消息流的处理程序才必须执行该操作；例如，当应用程序使用从服务请求者到提供者的单向消息流时，处理程序不应该生成响应。

服务提供者管道

在服务提供者管道中，CICS 接收请求，该请求通过管道传递到目标应用程序。来自应用程序的响应通过同一管道返回给服务请求者。

当 CICS 作为服务提供者的角色时，它执行以下操作：

1. 从服务请求者接收请求。
2. 检验请求，并抽取与目标应用程序相关的内容。
3. 调用应用程序，传递从请求抽取的数据。
4. 当应用程序返回控制权时，使用应用程序返回的数据构造响应。
5. 将响应发送给服务请求者。

图 4 描述服务提供者设置中三个消息处理程序的管道：

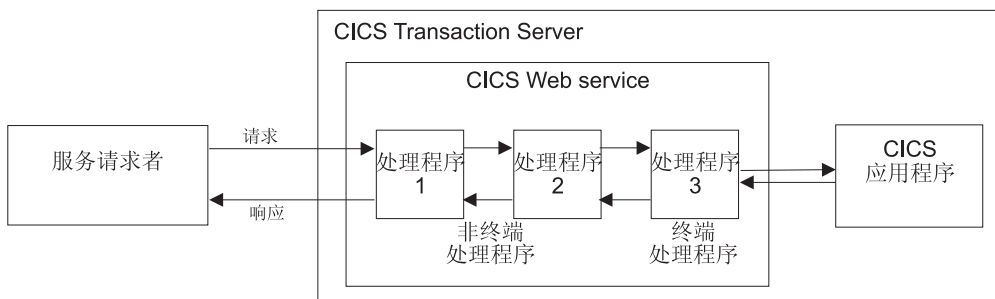


图 4. 服务提供者管道

1. CICS 从服务提供者接收请求。它将请求传递到消息处理程序 1。
2. 消息处理程序 1 执行某些处理，并将请求传递到处理程序 2（精确地说，它将控制权返回到管理管道的 CICS。然后，CICS 将控制权传递到下一个消息处理程序）。
3. 消息处理程序 2 从处理程序 1 接收请求，执行某些处理，并将请求传递到处理程序 3。
4. 消息处理程序 3 是管道的终端处理程序。它使用请求中的信息来调用应用程序。然后，它使用来自应用程序的输出生成响应，并将该响应传递回处理程序 2。
5. 消息处理程序 2 从处理程序 3 接收响应，执行某些处理，并将该响应传递到处理程序 1。
6. 消息处理程序 1 从处理程序 2 接收响应，执行某些处理，并将该响应返回到服务请求者。

服务请求者管道

在服务请求者管道中，应用程序创建请求，该请求通过管道传递到服务提供者。来自服务提供者的响应通过同一管道返回给应用程序。

当 CICS 作为服务请求者的角色时，它执行以下操作：

1. 使用应用程序提供的数据构造请求。
2. 将请求发送给服务提供者。

3. 从服务提供者接收响应。
4. 检验响应，并抽取与原始应用程序相关的内容。
5. 将控制权返回给应用程序。

图 5 描述服务请求者设置中三个消息处理程序的管道:

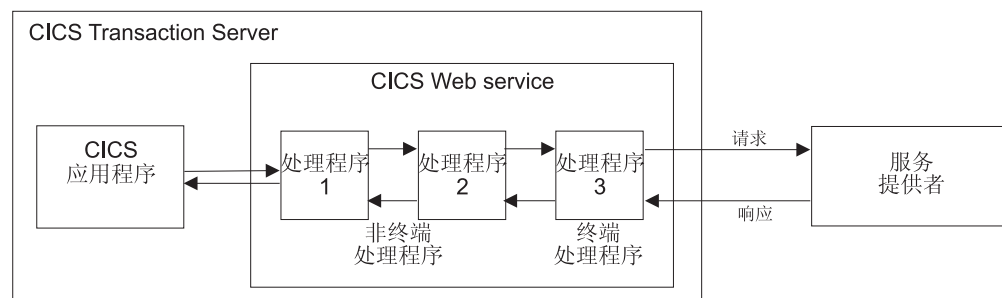


图 5. 服务请求者管道

1. 应用程序创建请求。
2. 消息处理程序 1 从应用程序接收请求，执行某些处理，并将请求传递到处理程序 2（精确地说，它将控制权返回到管理管道的 CICS。然后，CICS 将控制权传递到下一个消息处理程序）。
3. 消息处理程序 2 从处理程序 1 接收请求，执行某些处理，并将请求传递到处理程序 3。
4. 消息处理程序 3 从处理程序 2 接收请求，执行某些处理，并将请求传递到服务提供者。
5. 消息处理程序 3 从服务提供者接收响应，执行某些处理，并将该响应传递到处理程序 2。
6. 消息处理程序 2 从处理程序 3 接收响应，执行某些处理，并将该响应传递到处理程序 1。
7. 消息处理程序 1 从处理程序 2 接收响应，执行某些处理，并将该响应返回到应用程序。

CICS 管道和 SOAP

CICS 用于处理 Web service 请求和响应的管道是一般的，在该管道中，对于可在每个消息处理程序中执行什么处理有极少的限制。但是，许多 Web service 应用程序使用 SOAP 消息，且那些消息的任何处理都应该遵从 SOAP 规范。因此，CICS 提供特殊的 SOAP 消息处理程序来帮助您将管道配置为 SOAP 节点。

- 您的管道可以配置为支持 SOAP 1.1 或 SOAP 1.2。在您的 CICS 系统内，您可以有许多管道，某些管道支持 SOAP 1.1，某些支持 SOAP 1.2。
- 管道可以配置为在服务请求者中使用或在服务提供者中使用：
 - 服务请求者管道是请求的初始 SOAP 发送方，并且是响应的最终 SOAP 接收方
 - 服务提供者管道是请求的最终 SOAP 接收方，并且是响应的初始 SOAP 发送方
 您无法将 CICS 管道配置为起 SOAP 调解者的作用。
- 您可以将 CICS 管道配置为具有多个 SOAP 消息处理程序。

- 可以配置 CICS 提供的 SOAP 消息处理程序以调用一个或多个用户编写的头处理例程。
- 可以配置 CICS 提供的 SOAP 消息处理程序以执行遵从 WS-I Basic Profile 1.0 的某些方面，并提供 SOAP 消息中特定的头。

SOAP 消息处理程序以及它们的头处理例程是在管道配置文件中指定的。

SOAP 消息和应用程序数据结构

在许多情况下，CICS 服务助手可以生成代码，以在应用程序中使用的高级数据结构和 SOAP 消息中的 <Body> 元素的内容之间转换数据。在这些情况下，当您编写您的应用程序时，不需要解析或构造 SOAP 主体；CICS 将为您执行此操作。

为了转换数据，CICS 在运行时需要有关应用程序数据结构的信息以及有关 SOAP 消息的格式的信息。在两个文件中保存着此信息：

- Web service 绑定文件

CICS Web Service 助手可以使用实用程序 DFHLS2WS 从应用程序语言数据结构中生成此文件，也可以使用实用程序 DFHWS2LS 从 Web service 描述生成它。CICS 使用绑定文件生成 Web service 应用程序使用的资源，并执行应用程序的数据结构和 SOAP 消息之间的映射。

- Web service 描述

这可能是现有的 Web service 描述，或者它可能是使用实用程序 DFHLS2WS 从应用程序语言数据结构生成的。CICS 使用 Web Service 描述执行 SOAP 消息的完整验证。

图 6 显示在服务提供者中的哪些地方使用这些文件。

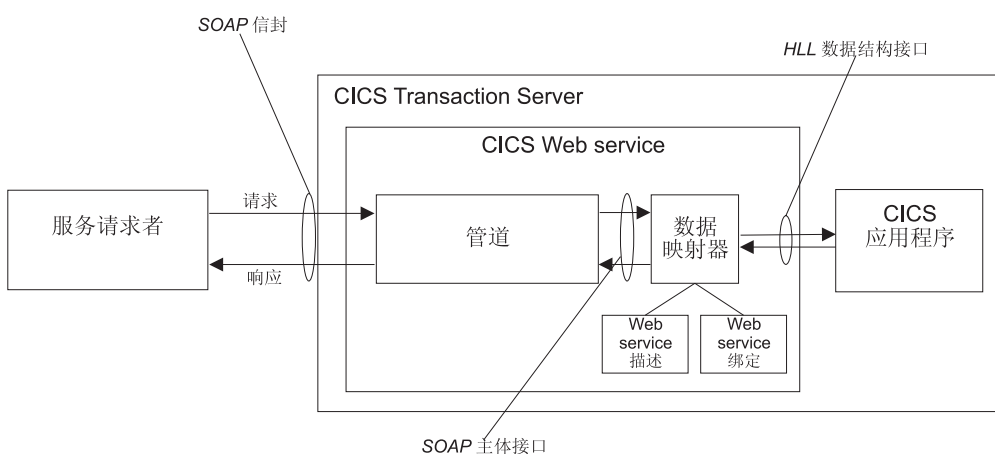


图 6. 在服务提供者中将 SOAP 主体映射到应用程序数据结构

第 19 页的图 7 显示在服务请求者中的哪些地方使用这些文件。

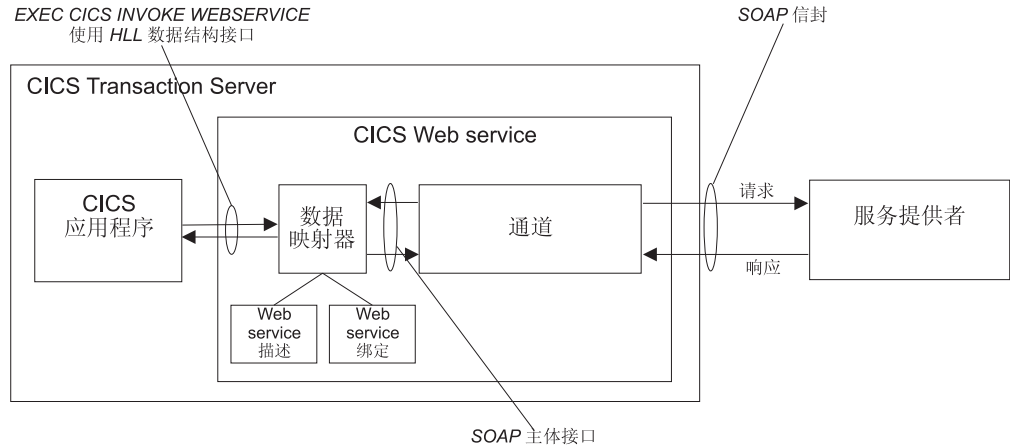


图 7. 在服务请求者中将 SOAP 主体映射到应用程序数据结构

在这两种情况下，允许特定 CICS 应用程序在 Web service 设置中操作的执行环境是由三个对象定义的。这些是管道、Web service 绑定文件以及 Web service 描述。将这三个对象定义到 CICS，作为资源定义的 WEBSERVICE 的属性。

存在某些您无法使用 CICS Web service 助手生成的转换的情况（即使您在使用 SOAP 消息）：

- 无法在 SOAP 消息和高级语言中表示相同的数据时。

CICS 支持的所有高级语言以及 XML 模式支持各种不同的数据类型。但是，以高级语言使用的数据类型和以 XML 模式使用的那些数据类型不是一一对应的，且存在可以用一种方式表示的数据不能用另一种方式表示的情况。在此情况下，您应该考虑以下某个情况：

- 更改您的应用程序数据结构。这可能并不可行，因为它可能需要对应用程序自身的更改。
- 构造包装器程序，它将应用程序数据转换为 CICS 随后可将之转换为 SOAP 消息体的格式。如果您执行此操作，则可以让您的应用程序保持不变。在此情况下，CICS Web service 支持与包装程序的直接交互以及仅与应用程序的非直接交互。
- 当使用 CICS Web service 助手不支持的语言编写应用程序时。

在此情况下，您应该考虑以下某个情况：

- 构造 CICS Web service 助手支持的某种语言（COBOL、PL/I、C 或 C++）编写的包装器程序。
- 不使用 CICS Web service 助手，而是写您自己的程序来执行 SOAP 消息和应用程序的数据结构之间的映射。

WSDL 和应用程序数据结构

Web service 描述包含服务使用的输入和输出消息的抽象表示方式。CICS 使用 Web service 描述来构造应用程序使用的数据结构。在运行时，CICS 执行应用程序数据结构和消息之间的映射。

在其他内容中，Web service 的描述包含：

- 一个或多个操作

- 每个操作的输入消息和可选的输出消息
- 用 XML 数据类型定义的每条消息的消息结构。消息中使用的复杂数据类型是在 XML 模式中定义的，该模式包含在 Web service 描述内的 <types> 元素中。可以不使用 <types> 元素描述简单消息。

WSDL 包含操作的抽象定义以及相关的信息；无法在应用程序中直接使用它。要实现操作，服务提供者必须执行以下操作：

- 它必须解析 WSDL，以理解消息的结构
- 它必须解析每个输入消息，并构造输出消息
- 它必须执行输入、输出消息的内容和应用程序中使用的数据结构之间的映射。

服务请求者必须执行相同操作，以调用操作。

您在使用 CICS Web Service 助手时，可以为您实现许多这样的操作，并且您不用详细了解 WSDL 或构造输入和输出消息的方法就可以编写应用程序。

CICS Web Service 助手由两个实用程序组成：

DFHWS2LS

此实用程序将 Web service 描述视作起始点。它使用消息的描述和在那些消息中所用的数据类型，来构造您可以在应用程序中使用的高级语言数据结构。

DFHLS2WS

此实用程序将高级语言数据结构视作起始点。它使用该结构来构造包含消息描述以及来自该语言结构的那些消息中使用的数据类型的 Web service 描述。

这两个实用程序都生成 Web service 绑定文件，CICS 在运行时用该绑定文件来执行应用程序的数据结构和 SOAP 消息之间的映射。

COBOL 到 WSDL 映射的示例

本示例显示如何在 CICS Web Service 助手生成的 Web service 描述中表示 COBOL 程序中使用的数据结构。

图 8 显示简单 COBOL 数据结构：

```

*   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).

```

图 8. 用 WSDL 定义的输入消息的 COBOL 记录定义

第 21 页的图 9 中显示 Web service 描述的相应片段中的键元素：

```

<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>

```

图 9. 来自 COBOL 数据结构的 WSDL 片段

Web service 绑定文件

Web service 绑定文件包含 CICS 用于在输入和输出消息以及应用程序数据结构之间映射数据的信息。

Web service 描述包含服务使用的输入和输出消息的抽象表示方式。当服务提供者或服务请求者应用程序执行时，CICS 需要有关消息的内容如何映射到应用程序所用的数据结构的信息。在 Web service 绑定文件中保存着此信息。

通过以下方法创建 Web service 绑定文件：

- 从 WSDL 生成语言结构时，通过实用程序 DFHWS2LS。
- 当从语言结构生成 WSDL 时，通过实用程序 DFHLS2WS。

在运行时，CICS 使用 Web service 绑定文件中的信息执行应用程序数据结构和 SOAP 消息之间的映射。在 WEBSERVICE 资源的 WSBIND 属性中为 CICS 定义 Web service 绑定文件。

规划在 CICS 中使用 Web service

您可以在 CICS 中规划使用 Web service 之前，您需要为每个应用程序考虑以下这些问题：

您计划以服务提供者或服务请求者角色部署 CICS 应用程序吗？

您可能有一对要使用 Web service 的 CICS 支持连接的应用程序。在这种情况下，一个应用程序将是服务提供者；而另一个则将是服务请求者。

您打算使用现有的应用程序还是编写一个新的应用程序？

如果您现有的应用程序设计有定义得很好的业务逻辑接口，则您可能将能够在 Web service 设置中将它们用作服务提供程序或者服务请求程序。然而，在多数情况下，您将需要编写将您的业务逻辑连接到 Web service 逻辑的包装程序。

如果您打算编写新的应用程序，则应该努力使您的业务逻辑与 Web service 逻辑保持分开，且将再次需要编写包装程序以提供此分离。然而，如果要将您的应用程序设计为具有 Web service，则事实上可能是编写包装程序更为简单。

您打算使用 SOAP 消息吗？

SOAP 是 Web service 体系结构的基础，而且 CICS 中提供的大量支持假设您将使用 SOAP。然而，可能存在您希望使用其他消息格式的情况。例如，您可能已部署自己的消息格式，这是您要使用 CICS Web service 基础结构部署的格式。CICS 允许您这样做，但是您将无法使用 CICS 提供的一些功能，如 Web service 助手和 SOAP 消息处理程序。

您打算使用 CICS Web Service 助手在您的数据结构和 SOAP 消息之间生成映射吗？

该助手提供使用少数或不使用其他编程，将很多应用程序快速部署到 Web service 设置。而且当需要其他编程时，它通常很简单，而且可以在不更改现有业务逻辑的情况下完成。

然而，存在不使用 Web service 助手而进行处理的更好方法。例如，如果您具有将数据结构映射到 SOAP 消息的现有代码，则在使用 Web service 助手重新建造应用程序方面没有优势。

您打算使用现有的服务描述还是创建一个新的描述？

在某些情况下，您将不得不使用现有的服务描述作为起始点。例如：

- 您的应用程序是服务请求者，并且设计它是用来调用现有的 Web service。
- 您的应用程序是服务提供者，并且您希望它符合现有的业界标准服务描述。

在其他情况下，可能需要为您的应用程序创建新的服务描述。

接下来的步骤：

- 规划服务提供者
- 规划服务请求者

规划服务提供者应用程序

通常，应该构造 CICS 应用程序以确保将业务逻辑和通信逻辑分开。完成此实践将帮助您以简单明了的方式在 Web Service 提供者中部署新的和现有的应用程序。在某些情况下，您将需要在应用程序和 CICS Web service 支持之间放入简单的包装程序。

第 23 页的图 10 显示典型的应用程序，分割它以确保通信逻辑和业务逻辑分开。

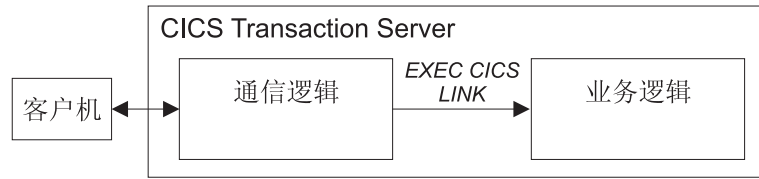


图 10. 应用程序分割成通信和业务逻辑

在很多情况下，您可以将业务逻辑作为服务提供者应用程序直接部署。这在图 11 中有所描述。

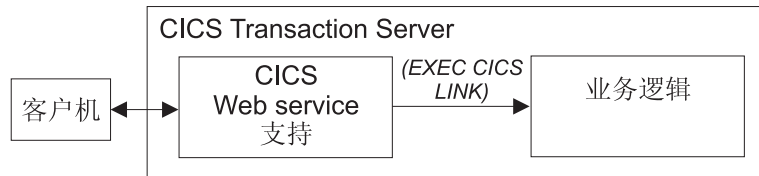


图 11. 作为 Web Service 提供者的 CICS 应用程序的简单部署

要使用这个简单的模型，应用下列条件：

当您使用 CICS Web Service 助手在 SOAP 消息和应用程序数据结构之间生成映射时： CICS Web Service 助手必须支持到业务逻辑的接口中使用的数据类型。如果不是此情况，您必须在 CICS Web Service 支持和您的业务逻辑之间放入包装程序。

您在部署现有的程序时还将需要一个包装程序，以提供符合现有 Web service 描述的服务：如果您使用该助手处理 Web service 描述，则产生的数据结构很可能与到您的业务逻辑的接口不匹配。

当您不使用 CICS Web service 助手时：

服务提供者管道中的消息处理程序必须与您的业务逻辑直接交互。

使用包装程序

当 CICS Web Service 助手无法生成与业务逻辑直接交互的代码时使用包装程序。例如，到业务逻辑的接口可能使用 CICS Web Service 助手无法直接映射到 SOAP 消息的数据结构。在此情况下，您可以使用包装程序来提供任何其他必需的数据操作：

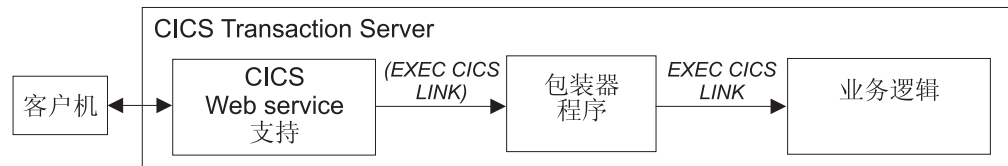


图 12. 使用包装程序部署作为 Web service 提供者的 CICS 应用程序

您将需要设计助手能够支持的第二种数据结构，并将它用作到您的包装程序的接口。然后，包装程序要执行两个简单功能：

- 在两个数据结构之间移动数据
- 使用它的现有接口来调用业务逻辑

规划服务请求者应用程序

通常，应该构造 CICS 应用程序以确保将业务逻辑和通信逻辑分开。完成此实践将帮助您以简单明了的方式在 Web Service 请求者中部署新的和现有的应用程序。几乎在每种情况下，您都将需要在应用程序和 CICS Web service 支持之间放入简单的包装程序。

图 13 显示典型的应用程序，分割它以确保通信逻辑和业务逻辑分开。完美地构造此应用程序以在 Web Service 提供者中重新使用业务逻辑。

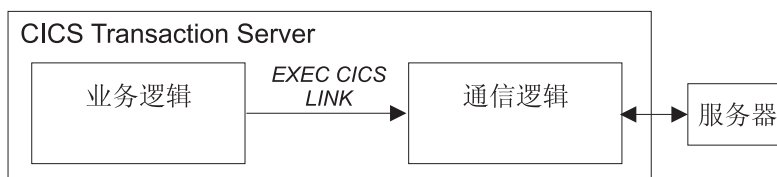


图 13. 应用程序分割成通信和业务逻辑

在这种情况下，您无法使用现有 EXEC CICS LINK 命令来调用 CICS Web service 支持：

- 当您要使用 CICS Web Service 助手在 SOAP 消息和应用程序数据结构之间生成映射时，您必须使用 EXEC CICS INVOKE WEBSERVICE 命令，并将应用程序数据结构传递给 CICS Web service 支持。而且，CICS Web Service 助手必须支持到业务逻辑的接口中使用的数据类型。
- 当您不使用 CICS Web service 助手时，您必须构造自己的消息，并链接到程序 DFHPIRT。

任何一种方法，您的业务逻辑都无法直接调用 Web service，除非您准备更改程序。对于 Web service 助手，图 14 中显示此选项，但不建议使用它。

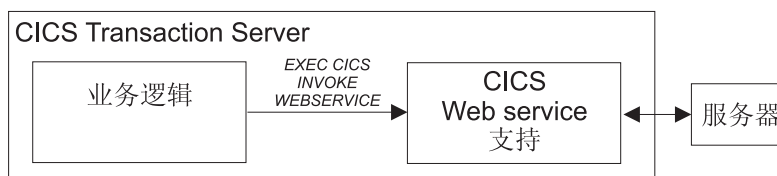


图 14. 作为 Web Service 请求者的 CICS 应用程序的简单部署

使用包装程序

让业务逻辑保持不变的较佳解决方案是使用包装程序。在此情况下，包装器有两个目的：

- 它代表业务逻辑发出 EXEC CICS INVOKE WEBSERVICE 或 EXEC CICS LINK PROGRAM(DFHPIRT)；业务逻辑仍能保持不变。
- 如果您的应用程序使用 CICS Web Service 助手无法直接映射到 SOAP 消息的数据结构，若有必要，包装程序可以提供任何必需的数据操作。

对于使用 Web service 助手的情况，此结构在第 25 页的图 15中有所说明。



图 15. 使用包装程序作为 Web Service 请求者部署 CICS 应用程序

CICS Web services 助手

CICS Web Service 助手是一组可以帮助您将现有的 CICS 应用程序转换到 Web service、并且允许 CICS 应用程序使用外部提供者提供的 Web service 的批处理实用程序。该助手支持 CICS 应用程序的快速部署，只需要进行很少的编程，就可以在服务提供者和服务请求者中使用它们。

您在使用 CICS 的 Web Service 助手时，不需要为解析入站消息和构造出站消息编写自己的代码；CICS 会映射 SOAP 消息体和应用程序的数据结构之间的数据。

在极大程度上，资源定义都是自动生成和安装的。您必须定义 PIPELINE 资源，但是在很多情况下您可以使用 CICS 提供的某个管道配置文件。它们是：

basicsoap11provider.xml

使用 SOAP 1.1 消息处理程序的服务提供者的管道配置文件。

basicsoap11requester.xml

使用 SOAP 1.1 消息处理程序的服务请求者的管道配置文件。

助手可以从简单语言结构创建 WSDL 文档，也可以从现有的 WSDL 文档创建语言结构，并且支持 COBOL、C/C++ 和 PL/I。它还生成用于使 SOAP 消息与容器和 COMMAREA 相互进行自动运行时转换的消息。

然而，助手无法处理每个可能的情况，有时您将需要采用不同的方法。例如：

您不要使用 SOAP 消息

如果您首选为消息使用非 SOAP 协议，则您可这样做。然而，您的应用程序将负责解析入站消息，并构造出站消息。

您要使用 SOAP 消息，但是不要 CICS 解析它们

对于入站消息，助手将 SOAP 主体映射到应用程序数据结构。在一些应用程序中，您可能要自己解析 SOAP 主体。

CICS Web services 助手不支持应用程序的数据结构

尽管 CICS Web service 助手支持最常见的数据类型和结构，存在一些不受支持的数据类型和结构。在此情况下，您应该首先考虑提供将应用程序数据映射到助手可支持格式的程序层。如果这不可能，则您将需要自己解析消息。

如果您决定不使用 CICS Web service 助手，您将必须：

- 提供您自己的代码解析入站消息，并构造出站消息
- 提供您自己的管道配置文件
- 定义并安装您自己的 URIMAP 和 PIPELINE 资源

CICS Web Service 助手由两个实用程序组成：

DFHLS2WS

从语言结构生成 Web service 绑定文件。此实用程序还生成 Web service 描述。

DFHWS2LS

从 Web service 描述生成 Web service 绑定文件。此实用程序还生成可以在您的应用程序中使用的语言结构。

运行两个程序的 JCL 过程在 *hlq.XDFHINST* 库中。

DFHLS2WS: 高级语言到 WSDL 的转换

编目过程 DFHLS2WS 从高级语言数据结构生成 Web service 描述和 Web service 绑定文件。当您列出 CICS 应用程序作为服务提供者时，您可使用 DFHLS2WS。

DFHLS2WS 的作业控制语句

JOB 启动作业。

EXEC 指定过程名称 (DFHLS2WS)。

DFHLS2WS 需要足够的存储空间来运行 Java 虚拟机 (JVM)。建议您在 EXEC 语句上指定 REGION=0M。

INPUT.SYSUT1 DD

指定输入。通常在输入流中指定输入参数。但是，可以在数据集或分区数据集的成员中定义它们。

符号参数

以下符号参数是在编目过程 DFHLS2WS 中定义的：

JAVADIR=*path*

指定 DFHLS2WS 将使用的 Java 目录的名称。此参数的值追加在 /usr/lpp/ 后，给出完整路径名 /usr/lpp/*path*。

通常，您不需要指定此参数；缺省值是在 **JAVADIR** 参数中提供给 CICS 安装作业 (DFHISTAR) 的值。

USSDIR=*path*

在 UNIX 系统服务 HFS 中指定 CICS TS 目录的名称。此参数的值追加在 /usr/lpp/cicsts/ 后，给出完整路径名 /usr/lpp/cicsts/*path*

通常，您不需要指定此参数；缺省值是在 **USSDIR** 参数中提供给 CICS 安装作业 (DFHISTAR) 的值。

TMPDIR=*tmpdir*

在 DFHLS2WS 用作临时工作空间的 HFS 中指定目录的位置。作业运行时所使用的用户标识必须对此目录有读和写许可权。

缺省值是 /tmp。

TMPFILE=*tmpprefix*

指定 DFHLS2WS 用来构造临时工作空间文件名称的前缀。

缺省值为 DFHLS2WS

临时工作空间

执行期间，DFHLS2WS 创建以下三个临时文件：

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

其中

tmpdir 是在 **TMPDIR** 参数中指定的值
tmpprefix 是在 **TMPFILE** 参数中指定的值。

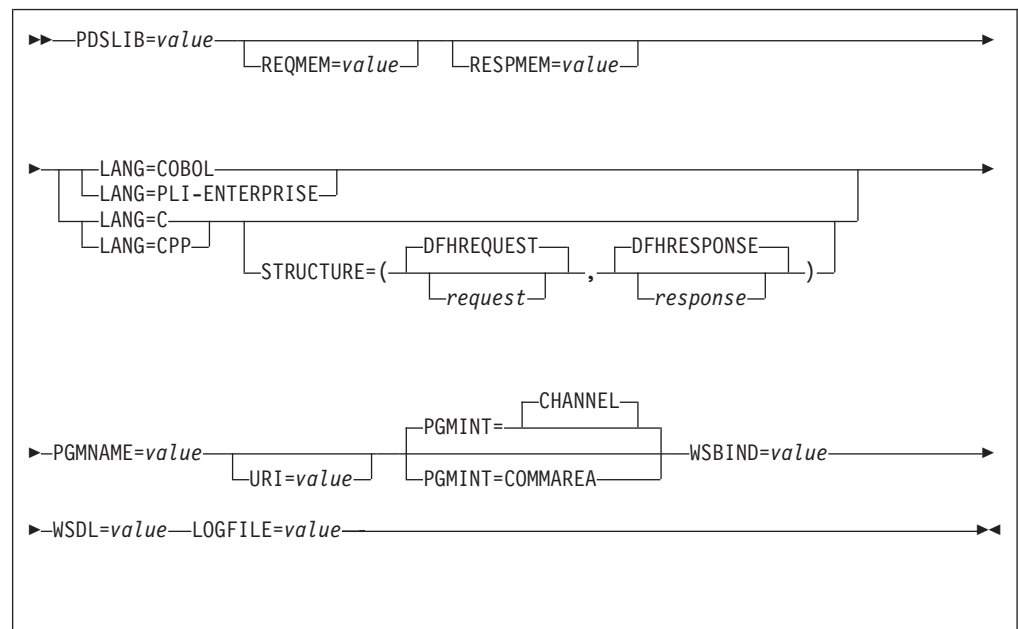
文件的缺省名称（当未指定 **TMPDIR** 和 **TMPFILE** 时）为：

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

重要：DFHLS2WS 不锁定对生成的 HFS 文件名的访问。因此，如果 DFHLS2WS 的两个或多个实例并发地运行，并使用相同的临时工作空间文件，则无法阻止一个作业在另一个作业正在使用工作空间文件时覆盖它们。这可能会导致不可预测的失败。

因此，建议您设计可以避免此情况的命名约定和操作过程。例如，您可以使用系统符号参数 **SYSUID** 来生成对各个用户来说是唯一的工作空间文件名。

DFHLS2WS 的输入参数



参数使用

- 您可以按任何顺序指定输入参数。
- 每个参数必须在新行开始。

- 如果参数太长而不能放在一行上，则在该行的结尾处使用一个星号 (*) 字符以表明参数延伸至下一行。星号前面的所有内容（包含空格）都认为是参数的一部分。例如：

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

等于

```
WSBIND=wsbinddir/app1
```

- 此行的第一个字符位置的 # 字符是注释字符。忽略此选项。

参数描述

LANG=COBOL

指定高级语言结构的编程语言为 COBOL。

LANG=PLI-ENTERPRISE

指定高级语言结构的编程语言为 Enterprise PL/I。

LANG=C

指定高级语言结构的编程语言为 C。

LANG=CPP

指定高级语言结构的编程语言为 C++。

PDSLIB=value

指定分区数据集的名称，该分区数据集包含要处理的高级语言数据结构。用于请求和响应的数据集成员是在 **REQMEM** 和 **RESPMEM** 参数中分别指定的。

限制： 分区数据集中的记录的长度必须是固定的 80 个字节。

PGMINT=CHANNEL|COMMAREA

对于服务提供者，指定 CICS 如何将数据传递到目标应用程序：

CHANNEL

CICS 使用通道接口将数据传递到目标应用程序。

COMMAREA

CICS 使用通信区域将数据传递到目标应用程序。

在服务请求者中使用来自 DFHLS2WS 的输出时忽略此参数。

PGMNAME=value

指定将列出为 Web service 的目标 CICS 应用程序名。这是 CICS Web service 支持将链接到的程序。

REQMEM=value

指定分区数据集成员的名称，该分区数据集成员包含 Web service 请求的高级语言结构：

- 对于服务提供者，Web service 请求是应用程序的输入
- 对于服务请求者，Web service 请求是来自应用程序的输出

RESPMEM=value

指定分区数据集成员的名称，该分区数据集成员包含 Web service 响应的高级语言结构：

- 对于服务提供者，Web service 响应是来自应用程序的输出
- 对于服务请求者，Web service 响应是应用程序的输入

如果无响应（即，对于单向消息），省略此参数。

STRUCTURE=(*request,response*)

仅对于 C 和 C++，指定包含在 REQMEM 和 RESPMEM 参数中所指定分区数据集的高级结构的名称。

request

指定在指定 REQMEM 参数时包含请求的高级结构的名称。缺省值为 DFHREQUEST。

分区数据集成员必须包含使用您所指定名称的高级结构（或者，如果您未指定名称，则包含名为 DFHREQUEST 的结构）。

response

指定在指定 RESPMEM 参数时包含响应的高级结构的名称。缺省值为 DFHRESPONSE。

如果您指定值，则分区数据集成员必须包含使用您所指定名称的高级结构（或者，如果您未指定名称，则包含名为 DFHRESPONSE 的结构）。

URI=*value*

在服务提供者中，此参数指定客户机将用于访问 Web service 的相对 URI。CICS 使用在从 DFHLS2WS 创建的 Web service 绑定文件生成 URIMAP 资源时指定的值：此参数指定应用 URIMAP 定义的 URI 的路径部分。

WSBIND=*value*

Web service 绑定文件的标准 HFS 名称。如果不存在此文件，DFHLS2WS 则创建它（但是不创建目录结构）。

WSDL=*value*

写入 Web service 描述的文件的标准 HFS 名称。如果不存在此文件，DFHLS2WS 则创建它（但是不创建目录结构）。

LOGFILE=*value*

文件的标准 HFS 名称，DFHLS2WS 将其活动日志和跟踪信息写入该文件。如果不存在此文件，DFHLS2WS 则创建它（但是不创建目录结构）。

通常，您不需要使用此文件，但如果您遇到有关 DFHLS2WS 的问题，则 IBM 服务组织可能会要求您提供该文件。

其他信息

- 运行 DFHLS2WS 的用户标识必须定义为 OMVS。此用户标识必须具有对 CICS HFS 文件结构的读许可权，以及对 LOGFILE 参数所指定目录的写许可权。
- 用户标识必须有足够大的存储分配来运行 Java。

DFHWS2LS: WSDL 到高级语言的转换

编目过程 DFHWS2LS 从 Web service 描述生成高级语言数据结构和 Web service 绑定文件。当您列出 CICS 应用程序作为服务提供者，或者当您构造服务请求者时，您可以使用 DFHWS2LS。

DFHWS2LS 的作业控制语句

JOB 启动作业。

EXEC 指定过程名称 (DFHWS2LS)。

DFHWS2LS 需要足够的存储空间来运行 Java 虚拟机 (JVM)。建议您在 EXEC 语句上指定 REGION=0M。

INPUT.SYSUT1 DD

指定输入。通常在输入流中指定输入参数。但是，可以在数据集或分区数据集的成员中定义它们。

符号参数

以下符号参数是在编目过程 DFHWS2LS 中定义的：

JAVADIR=*path*

指定 DFHWS2LS 将使用的 Java 目录的名称。此参数的值追加在 /usr/lpp/ 后，给出完整路径名 /usr/lpp/*path*。

通常，您不需要指定此参数；缺省值是在 **JAVADIR** 参数中提供给 CICS 安装作业 (DFHISTAR) 的值。

USSDIR=*path*

在 UNIX 系统服务 HFS 中指定 CICS TS 目录的名称。此参数的值追加在 /usr/lpp/cicsts/ 后，给出完整路径名 /usr/lpp/cicsts/*path*。

通常，您不需要指定此参数；缺省值是在 **USSDIR** 参数中提供给 CICS 安装作业 (DFHISTAR) 的值。

TMPDIR=*tmpdir*

在 DFHWS2LS 用作临时工作空间的 HFS 中指定目录的位置。作业运行时所使用的用户标识必须对此目录有读和写许可权。

缺省值是 /tmp。

TMPFILE=*tmpprefix*

指定 DFHWS2LS 用来构造临时工作空间文件名称的前缀。

缺省值为 DFHWS2LS。

临时工作空间

执行期间，DFHWS2LS 创建以下三个临时文件：

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

其中

tmpdir 是在 **TMPDIR** 参数中指定的值

tmpprefix 是在 **TMPFILE** 参数中指定的值。

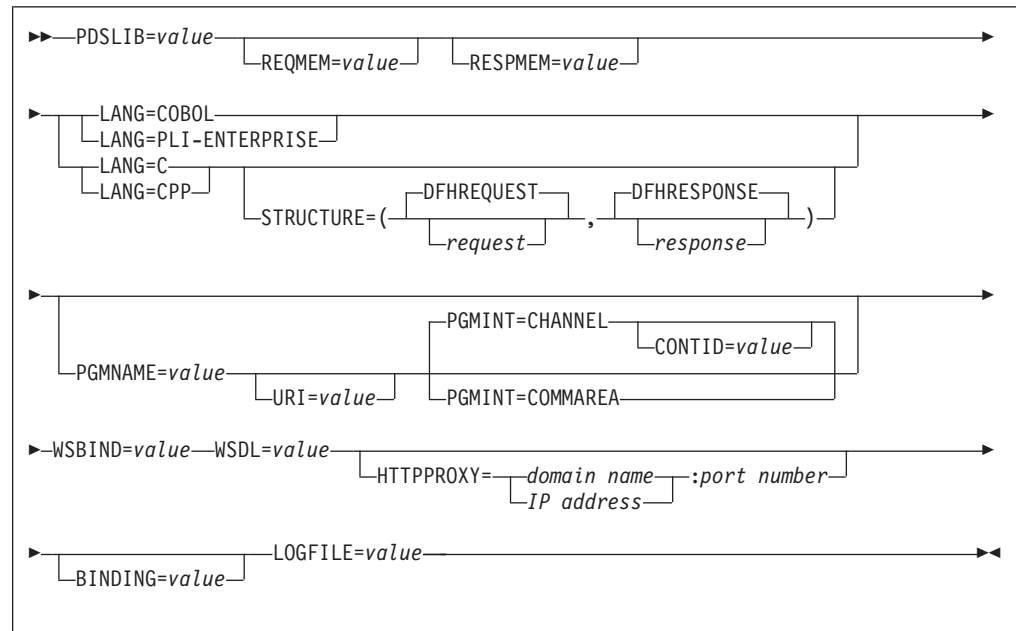
文件的缺省名称（当未指定 **TMPDIR** 和 **TMPFILE** 时）为：

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

重要: DFHWS2LS 不锁定对生成的 HFS 文件名的访问。因此，如果 DFHWS2LS 的两个或多个实例并发地运行，并使用相同的临时工作空间文件，则无法阻止一个作业在另一个作业正在使用工作空间文件时覆盖它们。这可能会导致不可预测的失败。

因此，建议您设计可以避免此情况的命名约定和操作过程。例如，您可以使用系统符号参数 SYSUID 来生成对各个用户来说是唯一的工作空间文件名。

DFHWS2LS 的输入参数



参数使用

- 您可以按任何顺序指定输入参数。
- 每个参数必须在新行开始。
- 如果参数太长而不能放在一行上，则在该行的结尾处使用一个星号 (*) 字符以表明参数延伸至下一行。星号前面的所有内容（包含空格）都认为是参数的一部分。例如：

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

等于

```
WSBIND=wsbinddir/app1
```

- 此行的第一个字符位置的 # 字符是注释字符。忽略此选项。

参数描述

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

如果您的 WSDL 包含对其他因特网上的 WSDL 文件的引用，并且您要运行 DFHWS2LS 的系统使用代理服务器来访问因特网，请指定代理服务器的域名或 IP 地址以及端口号。例如：

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

在其他情况下，不需要此参数。

LANG=COBOL

指定高级语言结构的编程语言为 COBOL。

LANG=PLI-ENTERPRISE

指定高级语言结构的编程语言为 Enterprise PL/I。

LANG=C

指定高级语言结构的编程语言为 C。

LANG=CPP

指定高级语言结构的编程语言为 C++。

PDSLIB=value

指定包含生成的高级语言的分区数据集的名称。用于请求和响应的数据集成员是在 **REQMEM** 和 **RESPMEM** 参数中分别指定的。

PGMINT=CHANNEL|COMMAREA

对于服务提供者，指定 CICS 如何将数据传递到目标应用程序：

CHANNEL

CICS 使用通道接口将数据传递到目标应用程序。

COMMAREA

CICS 使用通信区域将数据传递到目标应用程序。

在服务请求者中使用来自 DFHWS2LS 的输出时忽略此参数。

PGMNAME=value

此参数指定 CICS 程序的名称。

当 DFHWS2LS 用于生成将在服务提供者中使用的 Web service 绑定文件，则必须应用此参数。它指定列出为 Web service 的应用程序名。

当 DFHWS2LS 用于生成将在服务请求者中使用的 Web service 绑定文件，则必须省略此参数。

REQMEM=value

指定 1 到 6 个字符的前缀，DFHWS2LS 用它来为 Web service 请求生成将包含高级语言结构的分区数据集成员的名称：

- 对于服务提供者，Web service 请求是应用程序的输入
- 对于服务请求者，Web service 请求是来自应用程序的输出

DFHWS2LS 为每个操作生成分区数据集成员。通过在前缀后面追加两个数字生成成员名。

尽管此参数是可选的，但是如果 Web service 描述包含请求的定义，则必须指定它。

RESPMEM=value

指定 1 到 6 个字符的前缀，DFHWS2LS 用它来为 Web service 响应生成将包含高级语言结构的分区数据集成员的名称：

- 对于服务提供者，Web service 响应是来自应用程序的输出
- 对于服务请求者，Web service 响应是应用程序的输入

DFHWS2LS 为每个操作生成分区数据集成员。通过在前缀后面追加两个数字生成成员名。

如果无响应（即，对于单向消息），省略此参数。

STRUCTURE=(request,response)

仅对于 C 和 C++，指定如何生成请求和响应结构的名称。

生成的请求和响应结构的名称为 *requestnn* 和 *responsenn*，其中 *nn* 是生成的以区分每个操作的结构数字后缀。

如果一个或两个名称都省略，则结构的名称与您所指定 REQMEM 和 RESPMEM 参数生成的分区数据集成员名相同。

URI=value

在服务提供者中，此参数指定客户机将用于访问 Web service 的相对 URI。CICS 使用在从 DFHWS2LS 创建的 Web service 绑定文件生成 URIMAP 资源时指定的值：此参数指定应用 URIMAP 定义的 URI 的路径部分。

在服务请求者中，不使用此参数指定目标 Web service 的 URI：尽管您可在 EXEC CICS INVOKE WEBSERVICE 命令中使用 URI 覆盖，但是仍然使用 Web service 描述中指定的 URI。

WSBIND=value

Web service 绑定文件的标准 HFS 名称。如果不存在此文件，DFHWS2LS 则创建它（但是不创建目录结构）。

WSDL=value

包含 Web service 描述的文件的标准 HFS 名称。

LOGFILE=value

文件的标准 HFS 名称，DFHWS2LS 将其活动日志和跟踪信息写入该文件。如果不存在此文件，DFHWS2LS 则创建它（但是不创建目录结构）。

通常，您不需要使用此文件，但如果您遇到有关 DFHWS2LS 的问题，则 IBM 服务组织可能会要求您提供该文件。

BINDING=value

如果 Web service 描述包含多个 <binding> 元素，请使用此参数来指定要使用哪一个元素来生成语言结构和 Web service 绑定文件。指定在 Web service 描述中的 <binding> 元素中使用的 name 属性的值。

其他信息

- 运行 DFHWS2LS 的用户标识必须定义为 OMVS。此用户标识必须具有对 CICS HFS 文件结构的读许可权，以及对 LOGFILE 参数所指定目录的写许可权。
- 用户标识必须有足够大的存储分配来运行 Java。

对 CICS 外部的更改

对资源定义的更改

CICS 中的 Web service 使用三个新的 CICS 资源：PIPELINE、URIMAP 和 WEBSERVICE。

PIPELINE

PIPELINE 资源定义指定要应用于 Web service 请求的处理。要获取更多信息，请参阅第 34 页的『PIPELINE 资源定义』。

PIPELINE 引用定义处理节点的 XML 文件。

URIMAP

URIMAP 定义使得 CICS 能够匹配 Web 客户机发出的请求的 URI 或发送到远程服务器的请求的 URI，并且提供关于如何处理这些请求的信息。要获取更多信息，请参阅第 107 页的『URIMAP 资源定义』。

与 URIMAP 资源的下列属性相关：

PIPELINE

指定 PIPELINE 资源定义，它提供有关将担当客户机发出的服务请求的消息处理程序的信息。

WEBSERVICE

指定 WEBSERVICE，它为 Web service 设置中使用的 CICS 应用程序定义运行时环境方面的内容。

TRANSACTION

指定用于启动管道的别名事务的名称。

USERID

指定在其下附加别名事务的用户标识。

WEBSERVICE

WEBSERVICE 资源定义指定要应用于 Web service 请求的处理。要获得更多信息，请参阅第 36 页的『WEBSERVICE 资源定义』。

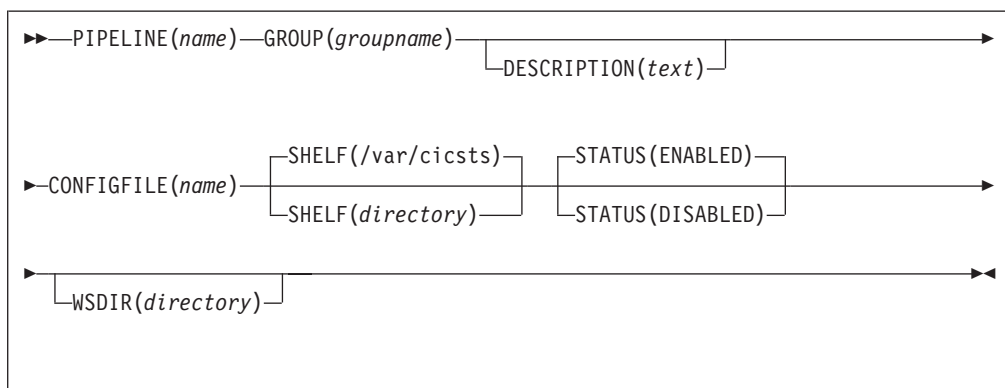
PIPELINE 资源定义

当 CICS 应用程序的角色为 Web Service 提供者或请求者时，使用 PIPELINE 资源定义。它提供关于作用于服务请求和响应的消息处理程序的信息。通常，单个“PIPELINE 定义”定义可由许多应用程序使用的基础结构。

关于处理节点的信息是间接提供的：PIPELINE 指定 HFS 文件的名称，而该文件包含这些节点的 XML 描述及其配置。

入站 Web Service 请求（即，客户机通过它调用 CICS 中 Web Service 的请求）通过 URIMAP 资源与 PIPELINE 资源关联。URIMAP 标识应用到与请求关联的 URI 的 PIPELINE 资源；PIPELINE 指定要对消息执行的处理。

PIPELINE 属性：



PIPELINE(name)

指定此 PIPELINE 的名称。此名称的长度可以达到八个字符。

可接受的字符:

A-Z 0-9 \$ @ #

除非您使用 CREATE 命令，否则您输入的任何小写字符都转换为大写字符。

GROUP(*groupname*)

每个资源定义都必须有一个 GROUP 名。在安装组时资源定义成为组的成员并安装在 CICS 系统中。

可接受的字符:

A-Z 0-9 \$ @ #

您输入的任何小写字符都会被转换成大写字符。

GROUP 名的长度最多可达到八个字符。小写字符会被当作大写字符对待。不要使用以 DFH 开始的组名，因为这些字符保留供 CICS 使用。

DESCRIPTION(*text*)

您可在此字段中提供要定义的资源描述。描述文本的长度可以达到 58 个字符。不限定您使用的字符。然而，如果您使用圆括号，则确保每个左圆括号都有一个匹配的右圆括号。如果您使用 CREATE 命令，则对于文本中的每个单撇号，编码成两个撇号。

CONFIGFILE(*name*)

指定 HFS 文件的名称，该文件包含关于将作用于服务请求和响应的处理节点的信息。

可接受的字符:

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

指定的值必须是 HFS 文件的有效名称:

- 该名称不能包含嵌入的空格字符。
- 该名称不能包含连续的 / 字符。

该名称是区分大小写的。

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

指定 HFS 上的目录 (*shelf*，主要用于 Web Service 绑定文件) 的标准名称，它的长度可以是 1 到 255 个字符。

可接受的字符:

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

指定的值必须是 HFS 文件的有效名称:

- 该名称不能包含嵌入的空格字符。
- 该名称不能包含连续的 / 字符。

该名称是区分大小写的。

安装 PIPELINE 定义的 CICS 区域必须具有对 shelf 目录的全部许可权，即，读、写和创建子目录的能力。

单个 shelf 目录可以由多个 CICS 区域和多个 PIPELINE 定义共享。在 shelf 目录中，每个 CICS 区域都使用单独的子目录来将它的文件与其他 CICS 区域的那些文件分开保存。在每个区域的目录中，每个 PIPELINE 都使用单独的子目录。

CICS 区域执行冷启动或初始启动之后，它在尝试使用 shelf 之前会从 shelf 删除它的子目录。

您不应该尝试修改已安装的 PIPELINE 定义引用的 shelf 的内容。如果您修改了，其影响将是不可预测的。

STATUS({ENABLED|DISABLED})

指定当安装 PIPELINE 时它的初始状态:

ENABLED

正常地处理对此 PIPELINE 的 Web Service 请求。

DISABLED

无法处理对此 PIPELINE 的 Web Service 请求。

WSDIR(directory)

指定 HFS 上的 Web Service 绑定目录（也称为检取目录）的标准名称，它的长度可以是 1 255 个字符。

可接受的字符:

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

指定的值必须是 HFS 文件的有效名称:

- 该名称不能包含嵌入的空格字符。
- 该名称不能包含连续的 / 字符。

该名称是区分大小写的。

Web Service 绑定目录包含与 PIPELINE 关联且要通过 CICS 扫描机制自动安装的 Web Service 绑定文件。当安装 PIPELINE 定义时，CICS 扫描此目录并自动安装它在此处找到的任何 Web Service 绑定文件。注意，无论 PIPELINE 是以启用状态安装还是以禁用状态安装，都会发生此情况。

如果您为 WSDIR 属性指定一个值，则它必须引用有效的 HFS 目录，CICS 区域对此目录至少要有读访问权。如果不是这种情况，则安装 PIPELINE 资源的任何尝试都会失败。

如果您不指定 WSDIR 的值，则在安装 PIPELINE 时不会发生自动扫描，并且 PERFORM PIPELINE SCAN 命令将失败。

WEBSERVICE 资源定义

WEBSERVICE 资源定义 Web Service 设置中部署的 CICS 应用程序运行时环境的一些方面，其中应用程序数据结构和 SOAP 消息之间的映射已使用 CICS Web Service 助手生成。尽管 CICS 为 WEBSERVICE 资源提供常见的资源定义机制，但通常使用此助手产生的输出动态安装它们。

由 WEBSERVICE 资源定义的运行时环境的一些方面为:

管道 定义对 Web Service 请求和响应操作的消息处理程序集。WEBSERVICE 资源指定单独的 PIPELINE 资源，PIPELINE 资源又指定管道配置文件

Web Service 绑定文件

包含运行时使用以执行应用程序数据结构和 SOAP 消息之间的映射的信息。Web Service 绑定文件是由 CICS 提供的工具生成的。

Web Service 描述

只有当 SOAP 消息的运行时验证是必需的情况下，才使用 Web Service 描述。每条消息的验证是根据它嵌入在 Web Service 描述中的模式执行的。

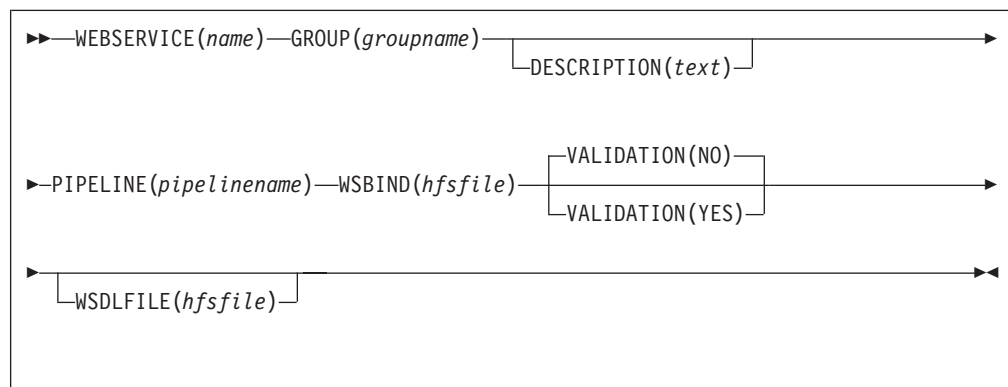
入站 Web Service 请求（即，客户机调用 CICS 中 Web Service 的请求）通过 URIMAP 资源与 WEBSERVICE 资源关联。URIMAP 标识应用到入站消息中 URI 的 WEBSERVICE 资源；WEBSERVICE 指定对消息执行的处理操作。

尽管 CICS 提供用于创建 WEBSERVICE 资源，并在 CICS 区域中安装它们的常见资源定义机制，但是您可使用扫描机制作为替代，在运行的 CICS 系统中动态安装 WEBSERVICE 资源。此方法的优点在于它减少必需的资源定义数，而且 CICS 可直接使用开发时提供的信息。

要调用扫描机制，使用 PERFORM PIPELINE 命令。

动态安装的 WEBSERVICE 的名称是从生成 WEBSERVICE 定义的 Web Service 绑定文件的名称派生的，最大长度为 32 个字符；从 CSD 或使用 EXEC CICS CREATE WEBSERVICE 安装的 WEBSERVICE 定义的名称限制为 8 位字符。例如，HFS 名称为 /samples/Webservices/WSDir/InquireSingle.wsbind 的 Web Service 绑定文件生成名为 InquireSingle 的 WEBSERVICE 定义

WEBSERVICE 属性:



WEBSERVICE(name)

为 WEBSERVICE 指定 1 到 8 个字符的名称。

可接受的字符:

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

不要使用以 DFH 开始的名称，因为这些字符保留供 CICS 使用。

GROUP(groupname)

每个资源定义都必须有一个 GROUP 名。在安装组时资源定义成为组的成员并安装

在 CICS 系统中。

可接受的字符:

A-Z 0-9 \$ @ #

您输入的任何小写字符都会被转换成大写字符。

GROUP 名的长度最多可达到八个字符。小写字符会被当作大写字符对待。不要使用以 DFH 开始的组名，因为这些字符保留供 CICS 使用。

DESCRIPTION(*text*)

您可在此字段中提供要定义的资源描述。描述文本的长度可以达到 58 个字符。不限定您使用的字符。然而，如果您使用圆括号，则确保每个左圆括号都有一个匹配的右圆括号。如果您使用 CREATE 命令，则对于文本中的每个单撇号，编码成两个撇号。

PIPELINE(*pipelinename*)

指定与此 WEBSERVICE 关联的 PIPELINE 1 到 8 个字符的名称。

可接受的字符:

A-Z 0-9 \$ @ #

除非您使用 CREATE 命令，否则您输入的任何小写字符都转换为大写字符。

VALIDATION(NO|YES)

指定是否应在运行时根据 Web Service 描述中相应的模式执行 SOAP 消息的完全验证。根据 SOAP 消息的模式验证它会引起相当大的处理开销，并且您通常应指定 VALIDATION(NO)。

完全验证确保发送和接收的所有 SOAP 消息对于 XML 模式是有效的 XML。如果指定 VALIDATION(NO)，则执行充分的验证以确保消息中的 XML 格式正确。

WSBIND(*hfsfile*)

为 HFS 上 Web Service 绑定文件指定 1 到 255 个字符的标准文件名。

可接受的字符:

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

该名称是区分大小写的并且不能包含空格。该名称不能以 / 字符结尾，也不能包含连续的 / 字符。

WSDLFILE(*hfsfile*)

为 HFS 上 Web Service 描述 (WSDL) 指定 1 到 255 个字符的标准文件名。当全部运行时验证活动时，使用此文件。

可接受的字符:

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

该名称是区分大小写的并且不能包含空格。该名称不能以 / 字符结尾，也不能包含连续的 / 字符。

对应用程序编程接口的更改

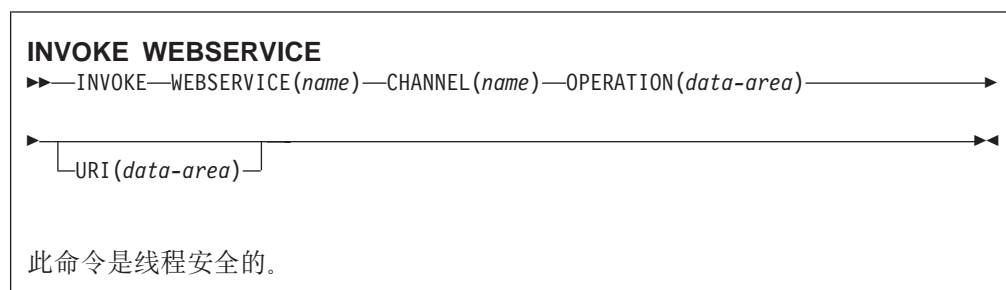
有一个新的命令，使用该命令使得 CICS 应用程序能够调用 Web Service。要获得更多的详细信息，请参阅『INVOKE WEBSERVICE』。

有三个新的命令，使用这些命令使得 SOAP 节点能够构造 SOAP 故障。要获得更多详细信息，请参阅：

- 第 40 页的『SOAPFAULT ADD』
- 第 40 页的『SOAPFAULT CREATE』
- 第 42 页的『SOAPFAULT DELETE』

INVOKE WEBSERVICE

该命令从 CICS 应用程序调用 Web service。该命令指定 WEBSERVICE 资源的名称，后者包含有关要调用的服务的信息。



选项

CHANNEL(*name*)

指定通道的名称，用于传递保存应用程序数据结构映射的数据的容器。反过来，同一通道持有来自 Web service 的响应（由应用程序数据结构再次映射）。通道的名称可以长达 16 个字符。如果 *name* 是变量，并且它包含少于 16 个字符的名称，则必须用尾部空格填补该变量。

OPERATION(*data-area*)

指定数据区，它包含要调用的操作的名称。操作的名称包含在目标 Web service 的 WSDL 中。数据区的长度必须是 255 个字符；如果操作名少于 255 个字符，则必须用尾部空格填补数据区。

URI(*data-area*)

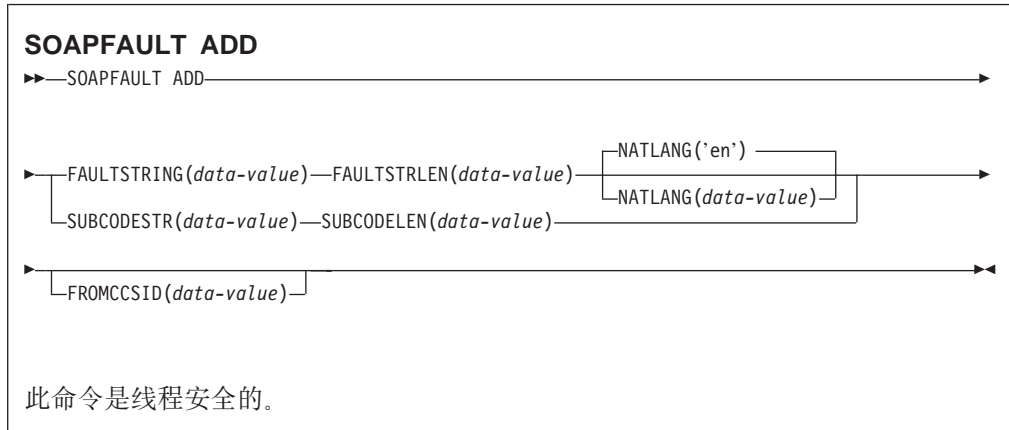
指定数据区，它包含要调用的 Web service 的 URI。如果指定，则此选项会取代 WEBSERVICE 资源定义中指定的任何 URI。如果省略此选项，则 WEBSERVICE 资源定义必须包括提供者 URI 或提供者应用程序名。数据区的长度必须是 255 个字符；如果 URI 少于 255 个字符，则必须用尾部空格填补数据区。

WEBSERVICE(*name*)

指定 WEBSERVICE 资源的名称，该资源定义要调用的 Web service。WEBSERVICE 资源指定 Web service 描述的位置和 CICS 在与 Web service 通信时使用的 Web service 绑定文件。WEBSERVICE 的名称可以长达 32 个字符。如果 *name* 是变量，并且它包含少于 32 个字符的名称，则必须用尾部空格填补该变量。

SOAPFAULT ADD

此命令将信息添加到现有的 SOAPFAULT 对象。只有在从 CICS 提供的 SOAP 消息处理程序调用的程序中才可以使用此命令。



选项

SUBCODESTR(*data-value*)

指定要添加到 SOAPFAULT 对象的 <subcode> 元素的内容。子代码的长度最多可以有 64 个字符，并且它必须是 XML 限定名 (QName)。

- 对于 SOAP 1.1，忽略此选项。
- 对于 SOAP 1.2，此选项提供 <Subcode> 元素的内容。

SUBCODELEN(*data-value*)

以全字二进制值指定 SUBCODESTR 选项中指定的 <subcode> 元素的长度。

FAULTSTRING(*data-value*)

指定故障的人类可读的说明。FAULTSTRING 最多可以有 2056 个字符。

- 对于 SOAP 1.1，此选项提供 <faultstring> 元素的内容。
- 对于 SOAP 1.2，此选项提供 <Reason> 元素的内容。

FAULTSTRLEN(*data-value*)

以全字二进制值指定 FAULTSTRING 选项的长度。

FROMCCSID(*data-value*)

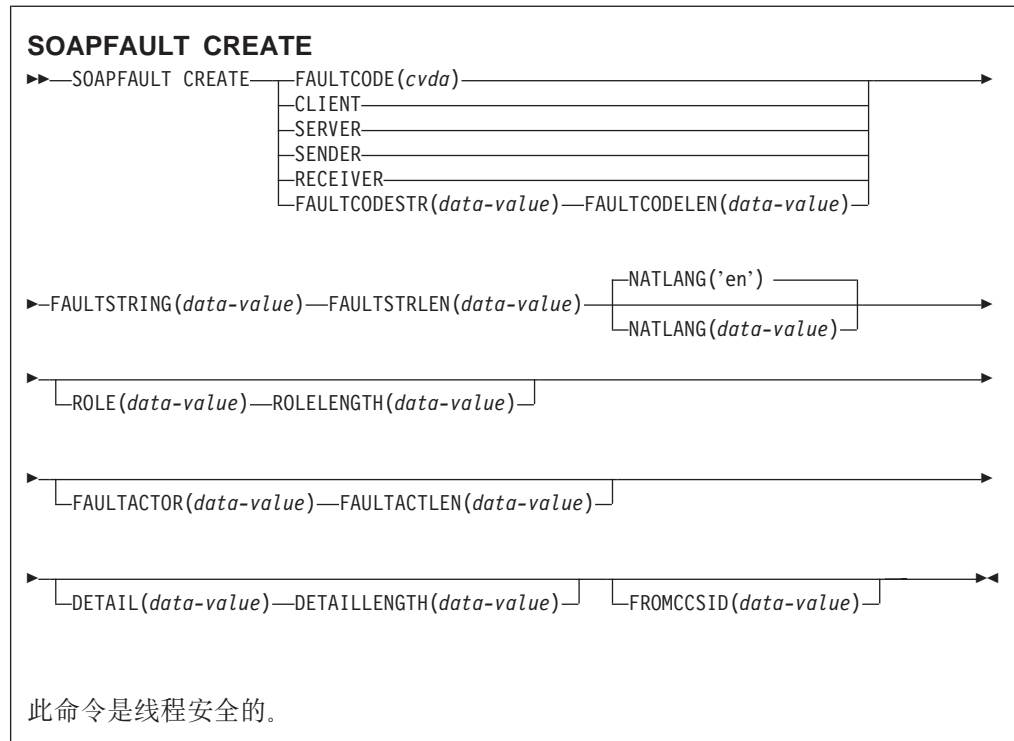
以全字十进制数指定，要放入 SOAP 故障的字符数据的当前编码字符集标识 (CCSID)。如果未指定此选项，则 CICS 使用在 LOCALCCSID 系统初始化参数中指定的值。要获得有关 CCSID 的信息和 CICS 所支持 CCSID 的列表，请参阅 *CICS Family: Communicating from CICS on System/390*。

NATLANG(*data-value*)

使用 XML 1.0 语言标识指定 FAULTSTRING 使用的本地语言。缺省值为 'en' (英语)。所指定值的长度不能超过 8 个字节。

SOAPFAULT CREATE

此命令创建 SOAP 故障。只有在从 CICS 提供的 SOAP 消息处理程序调用的程序中才可以使用此命令。



选项

DETAIL(*data-value*)

- 对于 SOAP 1.1, 此选项提供 SOAP 故障的 <detail> 元素的内容。
- 对于 SOAP 1.2, 此选项提供 SOAP 故障的 <Detail> 元素的内容。

它应该包含一个或多个有效的名称空间限定的 XML 元素或空格。请参阅相应的 SOAP 规范, 以获得有效元素内容的完整描述。

此元素传送与 <Body> 元素相关的应用程序特定错误信息, 并用于 <Body> 元素的内容无法成功处理时。对于 SOAP 1.1, 如果 <Body> 元素的内容无法成功处理, 则必须显示 <detail> 元素; 对于 SOAP 1.2, <Detail> 元素是可选的。

如果在头处理程序中发出 SOAPFAULT CREATE 命令, 则在头块中传送 <detail> 或 <Detail> 元素。

DETAILLENGTH(*data-value*)

以全字二进制值指定 DETAIL 选项的长度。

FAULTACTLEN(*data-value*)

以全字二进制值指定 FAULTACTOR 选项的长度。

FAULTACTOR(*data-value*)

- 对于 SOAP 1.1, 此选项提供 <faultactor> 元素的内容。
- 对于 SOAP 1.2, 此选项提供 <Node> 元素的内容。

FAULTACTOR 选项最多可以有 2056 个字符, 并且它必须是有效的 URI (anyURI)。

FAULTCODE(*cvda*)

CLIENT**SENDER**

对于 SOAP 1.1, 指定 Client 的 SOAP Fault 代码

对于 SOAP 1.2 规范, 指定 Sender 的 SOAP Fault 代码

SERVER**RECEIVER**

对于 SOAP 1.1, 指定 Server 的 SOAP Fault 代码

对于 SOAP 1.2, 指定 Receiver 的 SOAP Fault 代码

FAULTCODELEN(*data-value*)

以全字二进制值指定 FAULTCODESTR 选项的长度。

FAULTCODESTR(*data-value*)

指定用户定义的 SOAP Fault 代码。该 Fault 代码最多可以有 64 个字符, 并且它必须是 XML 限定名 (QName)。不支持用“.”(点) 字符分隔 Fault 代码值。

- 对于 SOAP 1.1, 此选项提供 <faultcode> 元素的内容。
- 对于 SOAP 1.2, 此选项提供 <Code> 元素的内容。

FAULTSTRING(*data-value*)

指定故障的人类可读的说明。FAULTSTRING 最多可以有 2056 个字符。

- 对于 SOAP 1.1, 此选项提供 <faultstring> 元素的内容。
- 对于 SOAP 1.2, 此选项提供 <Reason> 元素的内容。

FAULTSTRLEN(*data-value*)

以全字二进制值指定 FAULTSTRING 选项的长度。

FROMCCSID(*data-value*)

以全字十进制数指定, 要放入 SOAP 故障的字符数据的当前编码字符集标识 (CCSID)。如果未指定此选项, 则 CICS 使用在 LOCALCCSID 系统初始化参数中指定的值。要获得有关 CCSID 的信息和 CICS 所支持 CCSID 的列表, 请参阅 *CICS Family: Communicating from CICS on System/390*。

NATLANG(*data-value*)

使用 XML 1.0 语言标识指定 FAULTSTRING 使用的本地语言。缺省值为 'en' (英语)。所指定值的长度不能超过 8 个字节

ROLE(*data-value*)

指定描述生成故障的 SOAP 节点的角色 URI。ROLE 选项最多可以有 2056 个字符, 并且它必须是有效 URI (XML 类型 anyURI)。

- 对于 SOAP 1.1, 忽略此选项。
- 对于 SOAP 1.2, 此选项提供 <Role> 元素的内容。

ROLELENGTH(*data-value*)

以全字二进制值指定 ROLE 选项的长度。

SOAPFAULT DELETE

此命令删除现有的 SOAPFAULT 对象。只有在从 CICS 提供的 SOAP 消息处理程序调用的程序中才可以使用此命令。

SOAPFAULT DELETE

▶▶—SOAPFAULT DELETE—▶▶

此命令是线程安全的。

对系统编程接口的更改

COLLECT STATISTICS 命令

此命令支持下列新选项:

PIPELINE(*data-value*)

请求与 PIPELINE 相关的统计信息。这包括关于 HFS 文件的信息。

WEBSERVICE(*data-value*)

请求与 WEBSERVICE 相关的统计信息。

CREATE PIPELINE 命令

使用 CREATE PIPELINE 命令在您的 CICS 区域中动态地创建 PIPELINE。第 34 页的『PIPELINE 属性』中描述您可以在此命令上指定的属性。

CREATE WEBSERVICE 命令

使用 CREATE WEBSERVICE 命令在您的 CICS 区域中动态地创建 WEBSERVICE。第 37 页的『WEBSERVICE 属性』中描述您可以在此命令上指定的属性。

DISCARD PIPELINE 命令

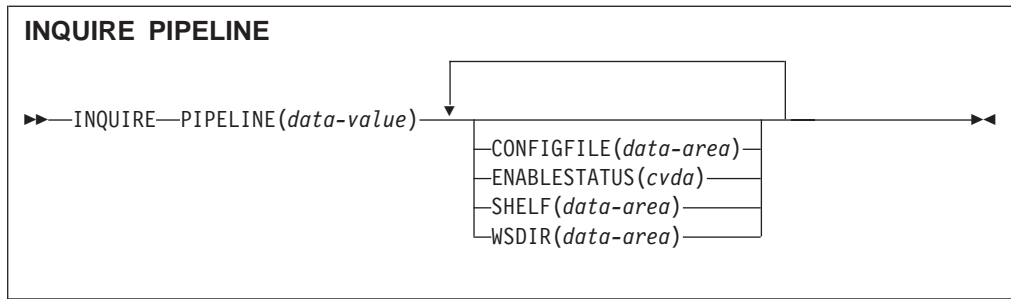
使用 DISCARD PIPELINE 管道从您的 CICS 区域中除去 PIPELINE。可以废弃 PIPELINE 之前，必须先禁用它。

DISCARD WEBSERVICE 命令

使用 DISCARD WEBSERVICE 命令从您的 CICS 区域中除去 WEBSERVICE。可以废弃 WEBSERVICE 之前，必须先禁用它。

INQUIRE PIPELINE 命令

使用 INQUIRE PIPELINE 获取有关已安装的 PIPELINE 的信息。



您可以在 **INQUIRE PIPELINE** 命令上使用浏览选项（**START**、**NEXT** 和 **END**）浏览您的系统中已安装的所有 **PIPELINE**。

选项

CONFIGFILE(*data-area*)

返回与 **PIPELINE** 资源关联的管道配置文件的名称。该名称可以长达 255 个字符。

ENABLESTATUS(*cvda*)

返回 **PIPELINE** 的状态:

ENABLED

PIPELINE 已经可供使用了。

DISABLED

PIPELINE 不在处理请求，并且无法接受新的工作。它可能无法初始化，或者可能被明确地禁用了。

ENABLING

PIPELINE 正在初始化；它还未准备好接受工作。

DISABLING

PIPELINE 在进入 **DISABLED** 状态之前在停止。它不会接受新的工作，而是允许当前执行的工作能够完成。

DISCARDING

为 **PIPELINE** 发出了 **DISCARD** 命令。**PIPELINE** 在被废弃之前，正在停止。它不会接受新的工作，而是允许当前执行的工作能够完成。

PIPELINE(*data-value*)

指定您在查询的 **PIPELINE** 的名称。该名称可以长达 8 个字符。

SHELF(*data-area*)

返回构架目录的名称。该名称可以长达 255 个字符。

WSDIR(*data-area*)

返回 Web service 绑定目录（也称为检取目录）的名称。该名称可以长达 255 个字符。

INQUIRE WORKREQUEST 命令

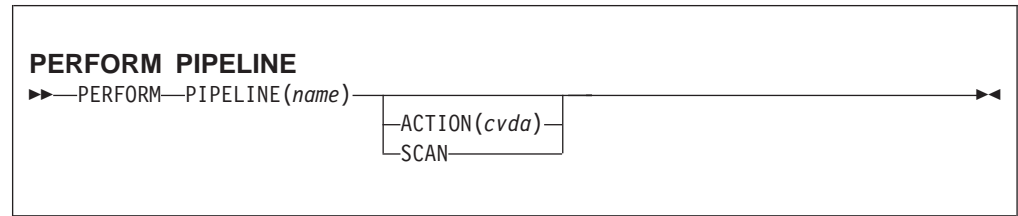
INQUIRE WORKREQUEST 命令上的 **WORKTYPE** 选项有一个新值:

SOAP

指定为 Web service 请求执行操作。

PERFORM PIPELINE 命令

使用 PERFORM PIPELINE 命令启动扫描与 PIPELINE 关联的 Web service 绑定文件。



扫描的 Web service 绑定文件在 PIPELINE 定义的 WSBIND 属性指定的目录中。如果不指定 WSBIND 属性，则不执行扫描，而是将控制返回给您的程序。

如果指定的目录位置有效，则 CICS 检查目录中的 Web service 绑定文件，以确定是否应该将它们安装到系统中：

- CICS 会安装任何它发现还未安装的文件。
- 如果文件已安装，但目录中的文件比当前在使用的文件要新，则废弃在使用的文件，而将较新的文件安装在该位置。

如果由于任何原因，CICS 无法安装独立的 Web service 绑定文件，则处理将继续使用目录中剩余的文件。扫描完成时，PIPELINE 可以供任何一个成功安装的绑定文件使用。

选项

ACTION(cvda)

指定表明要在 PIPELINE 上执行的操作的 CVDA 值。CVDA 值有：

SCAN 扫描 PIPELINE 的 Web service 绑定目录

PIPELINE(name)

指定 PIPELINE 的名称。

PERFORM STATISTICS RECORD 命令

此命令支持下列新选项：

PIPELINE

请求与 PIPELINE 相关的统计信息。这包括关于 HFS 文件的信息。

SET PIPELINE 命令

使用 SET PIPELINE 命令更改已安装的 PIPELINE 的状态：

PIPELINE(data-value)

指定您在查询的 PIPELINE 的 8 个字符的名称。

ENABLESTATUS(cvda)

指定 PIPELINE 的状态：

ENABLED

正常地处理此 PIPELINE 的入站服务请求。

DISABLED

拒绝此 PIPELINE 的人站服务请求。

SET WORKREQUEST 命令

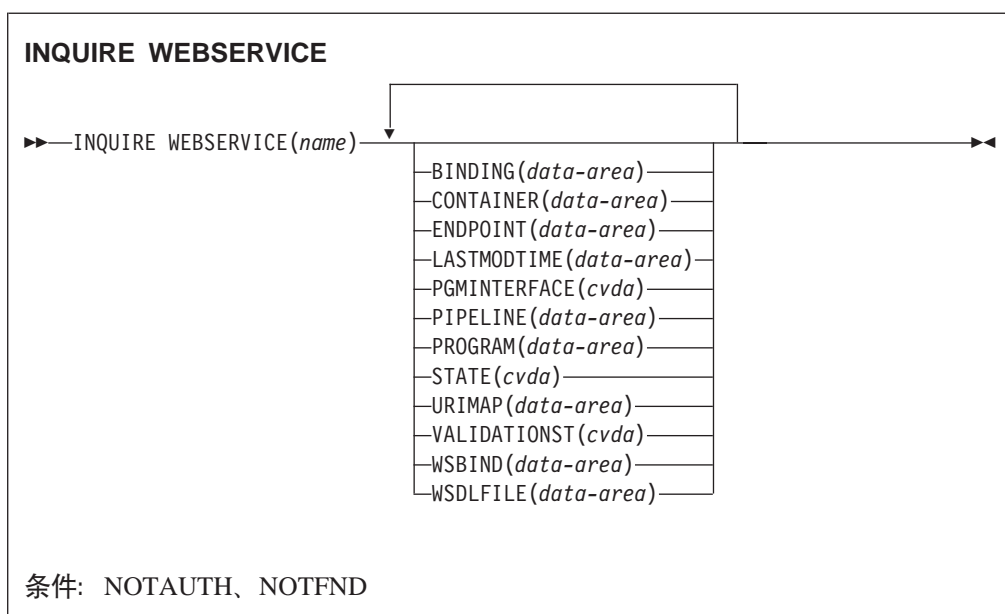
SET WORKREQUEST 命令上的 WORKTYPE 选项有一个新值:

SOAP

指定为 Web service 请求执行操作。

INQUIRE WEBSERVICE

使用 INQUIRE WEBSERVICE 命令检索有关已安装 WEBSERVICE 的信息。



浏览

您可以使用 INQUIRE WEBSERVICE 命令的浏览选项 (START、NEXT 和 END) 浏览您的系统中已安装的所有 WEBSERVICE。

选项

BINDING(*data-area*)

返回 WEBSERVICE 代表的 WSDL 绑定。此绑定是 WSDL 文件中出现的很多绑定之一 (潜在的)。该名称可以长达 255 个字符。

CONTAINER(*data-area*)

如果 PGMINTERFACE 返回 CHANNEL 的值, 则返回使用的容器的名称。该名称可以长达 16 个字符。

ENDPOINT(*data-area*)

返回远程 WEBSERVICE 的端点 URI。这是在 WSDL 文件中为远程 Web Service 指定的端点 URI。如果 CICS 应用程序是服务提供者, 则 ENDPOINT 将是空的。URI 可以长达 255 个字符。

LASTMODTIME(*data-area*)

以毫秒为单位返回自 1900 年 1 月 1 日 00:00 以来，最近一次更新 HFS 的已部署 WSBIND 文件的时间。这是一个在安装或更新 WEBSERVICE 资源时由 CICS 更新的只读值。最近一次修改的时间可以用于确定 CICS 是否在对检索目录中的 WSBIND 文件进行更新后已自己刷新。

- 对于动态安装的 WEBSERVICE（它们由 CICS 扫描机制安装），LASTMODTIME 的值是 WSBIND 定义指向 HFS 文件的时间戳记，最近一次安装或更新 WEBSERVICE 定义是在那个时间。
- 对于静态安装的 WEBSERVICE（它们从 CSD 安装或通过 CREATE WEBSERVICE 安装），LASTMODTIME 的值是 WEBSERVICE 定义指向 WSBIND HFS 文件的时间戳记，在该时刻安装 WEBSERVICE。

如果您在新近安装或更新的 WEBSERVICE 已完全初始化之前发出 INQUIRE WEBSERVICE 命令，则返回的 LASTMODTIME 值将是零。

以 8 个字节的压缩十进制格式返回该值。您可以使用 EXEC CICS FORMATTIME 命令将 LASTMODTIME 值转换为您喜欢的日期和时间格式。

PGMINTERFACE(*cvda*)

返回表明实施 Web Service 的 CICS 程序要求在通道中还是在通信区中有输入的 CVDA。

CHANNEL

程序要求在通道中有输入。

COMMAREA

程序要求在通信区中有输入

PIPELINE(*data-area*)

返回在其中安装 WEBSERVICE 的 PIPELINE 的名称；即，包含此 WEBSERVICE 资源的 PIPELINE 资源的名称。 该名称可以长达 8 个字符。

PROGRAM(*data-area*)

返回实施 Web Service 的 CICS 程序的名称。如果此 WEBSERVICE 代表远程 Web Service（即，CICS 不是服务提供者），则 RPROGRAM 将是空的。该名称可以长达 8 个字符。

STATE(*cvda*)

返回表明 WEBSERVICE 的状态的 CVDA:

DISCARDING

正在为此 WEBSERVICE 执行 DISCARD

INITING

正在将 Web Service 绑定文件和 WSDL 文件复制到构架。

INSERVICE

已成功解析构架上的 WSBIND 文件副本，并且可以使用 WEBSERVICE。

UNUSABLE

复制构架上的 WSBIND 文件已失败，并且不能使用 WEBSERVICE。

DISCARDING

为 WEBSERVICE 发出了 DISCARD 命令。WEBSERVICE 在被废弃之前，正在停止。它不会接受新的工作，而是允许当前执行的工作能够完成。

URIMAP(*data-area*)

返回动态安装的 URIMAP 的名称（如果有一个与此 WEBSERVICE 关联的话）。如果 WEBSERVICE 不是通过执行 PIPELINE 资源的 SCAN 函数安装的，或者如果 WEBSERVICE 代表远程 Web Service，则 URIMAP 将是空的。该名称可以长达 8 个字符。

VALIDATIONST(*cvda*)

返回表明当前是否为此 WEBSERVICE 启用 SOAP 消息的完全验证的 CVDA:

VALIDATION

启用完全验证。

DISABLED

禁用完全验证。

WEBSERVICE(*name*)

指定您要查询的 WEBSERVICE 的名称。该名称可以长达 32 个字符。

WSBIND(*data-area*)

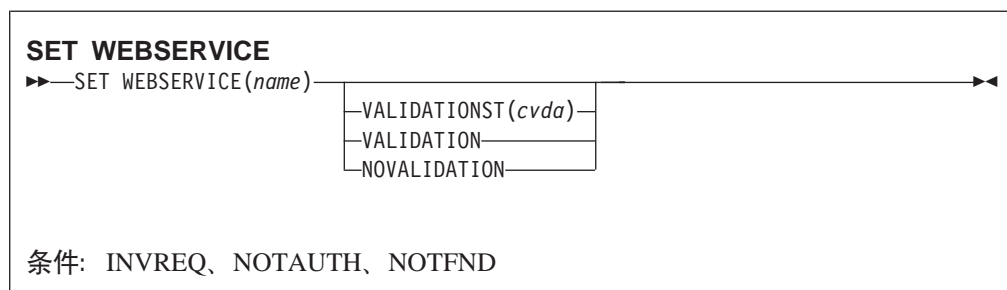
返回 Web Service 绑定文件的名称。该名称可以长达 255 个字符。

WSDLFILE(*data-area*)

返回与 WEBSERVICE 资源关联的 Web Service 描述文件的名称。该名称可以长达 255 个字符。

SET WEBSERVICE

使用 SET WEBSERVICE 命令更改已安装的 WEBSERVICE 的状态。



选项

WEBSERVICE(*name*)

指定 WEBSERVICE 的名称。

VALIDATIONST(*cvda*)

指定是否为 WEBSERVICE 启用完全验证。 CVDA 值为:

VALIDATION

启用完全验证。

NOVALIDATION

未启用完全验证。

对 CEMT 的更改

CEMT DISCARD WEBSERVICE

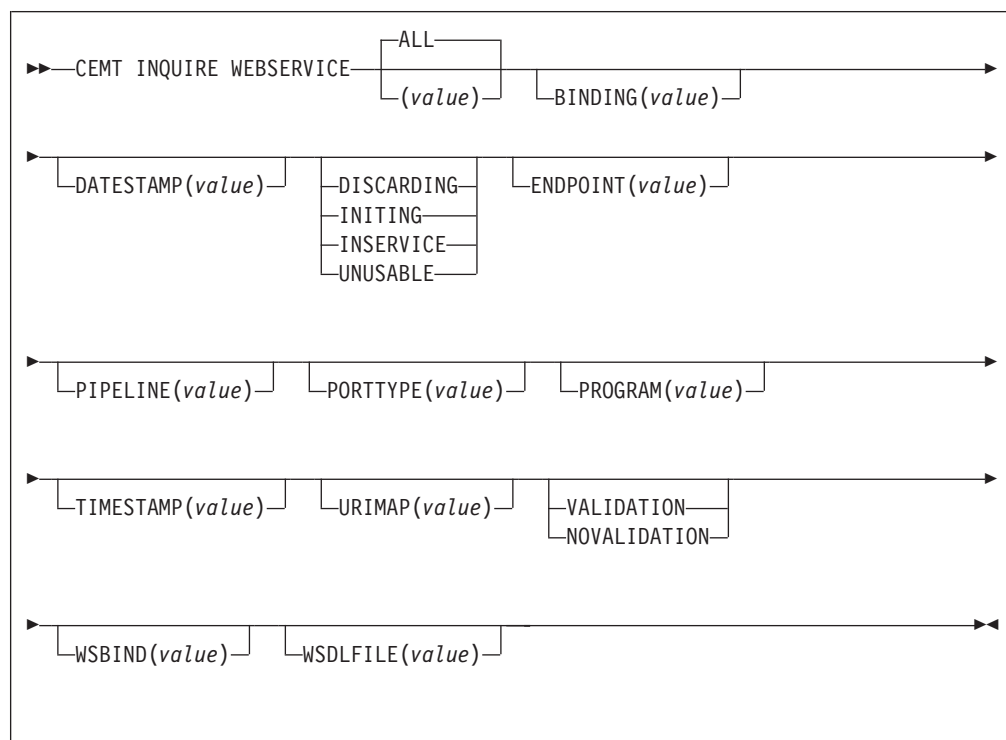
使用 DISCARD WEBSERVICE 命令从 CICS 区域中除去 WEBSERVICE。

CEMT DISCARD PIPELINE

使用 DISCARD PIPELINE 从 CICS 区域中除去 PIPELINE。

CEMT INQUIRE WEBSERVICE

使用 INQUIRE WEBSERVICE 命令显示下列有关已安装的 WEBSERVICE 的信息：



BINDING(value)

显示 WEBSERVICE 表示的 WSDL 绑定。此绑定是 WSDL 文件中出现的很多绑定之一（潜在的）。

DATESTAMP(value)

以 `yyyymmdd` 的格式显示最近一次更新 WEBSERVICE 的日期。这是一个在安装或更新 WEBSERVICE 资源时由 CICS 更新的只读值。

COMMAREA

程序要求在 COMMAREA 中接收输入。

CONTAINER

程序要求在 CONTAINER 中接收输入。

DISCARDING

正在为此 WEBSERVICE 执行 DISCARD 操作。

ENDPOINT(*value*)

显示远程 WEBSERVICE 的端点 URI。这是在 WSDL 文件中为远程 Web Service 指定的端点 URI。如果 CICS 应用程序是服务提供者，则 ENDPOINT 将是空的。

INITING

正在将 WSBIND 文件和 WSDL 文件复制到构架。

INSERVICE

已成功解析构架上的 WSBIND 文件副本，并且可以使用 WEBSERVICE。

NOVALIDATION

未启用完全验证。

PIPELINE(*value*)

显示在其中安装 WEBSERVICE 的 PIPELINE 的名称；即，包含此 WEBSERVICE 资源的 PIPELINE 资源的名称。

PORTTYPE(*value*)

显示此 WEBSERVICE 资源表示的 WSDL 文档内 <portType> 元素的内容。

PROGRAM(*value*)

显示 CICS 程序的名称，其在 CICS 作为服务供应商时实施 WSDL <portType>。

TIMESTAMP(*value*)

以 hh:mm:ss 的格式显示最近一次更新 WEBSERVICE 的时间。这是一个在安装或更新 WEBSERVICE 资源时由 CICS 更新的只读值。

URIMAP(*value*)

显示动态安装的 URIMAP 的名称（如果有一个与此 WEBSERVICE 关联的话）。如果 WEBSERVICE 不是通过执行 PIPELINE 资源的 SCAN 函数安装的，或者如果 WEBSERVICE 表示远程 Web Service，则 URIMAP 将是空的。

UNUSABLE

将 WSBIND 文件复制到构架已失败，并且不能使用 WEBSERVICE。

VALIDATION

启用完全验证。

WSBIND(*value*)

返回 HFS 上的 WSBIND 文件的 1-255 个字符的标准文件名。

WSDLFILE(*value*)

返回 HFS 上的 WSDL 文件的 1-255 个字符的标准文件名。

CEMT INQUIRE WORKREQUEST

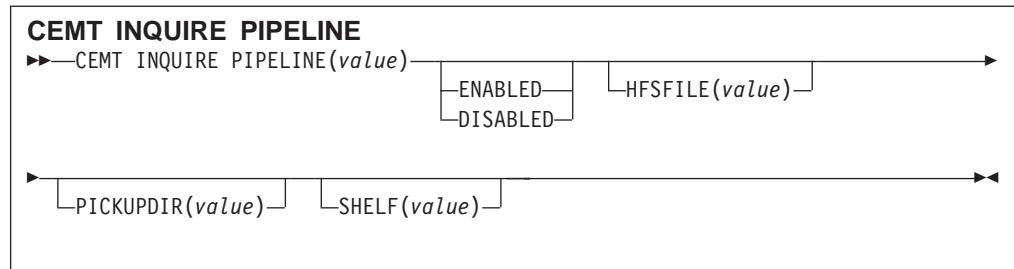
INQUIRE WORKREQUEST 命令上的 WORKTYPE 选项有一个新值：

PIPE

指定工作会针对 Web Service 请求而执行。

CEMT INQUIRE PIPELINE

使用 INQUIRE PIPELINE 命令显示有关已安装的 PIPELINE 的信息。



PIPELINE(*value*)

指定您在查询的 PIPELINE 的 8 个字符的名称。

HFSFILE(*value*)

返回包含关于数据处理节点信息的 HFS 文件，上述节点将根据服务请示和响应进行相应操作。

ENABLESTATUS(*cvda*)

返回 PIPELINE 的状态:

ENABLED

正常地处理此 PIPELINE 的入站服务请求。

DISABLED

拒绝此 PIPELINE 的入站服务请求。

PICKUPDIR(*value*)

返回 HFS 上的检取目录的标准名称，它的长度可以是 1 到 255 个字符、

SHELF(*value*)

返回 HFS 上的构架目录的标准名称，它的长度可以是 1 到 255 个字符。

CEMT PERFORM STATISTICS

此命令支持下列新选项:

PIPELINE

立即将与 PIPELINE 相关的统计信息写入 SMF 数据集。统计信息包括有关 HFS 文件的信息。

WEBSERVICE

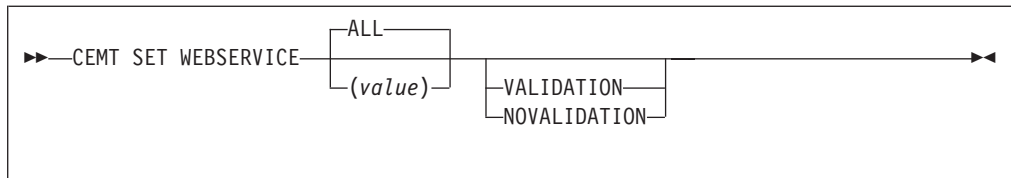
立即将与 WEBSERVICE 相关的统计信息写入 SMF 数据集。

CEMT PERFORM PIPELINE

使用 CEMT PERFORM PIPELINE 命令开始扫描 PIPELINE 的 HFSFILE 属性中命名的目录。

CEMT SET WEBSERVICE

使用 SET WEBSERVICE 命令更改已安装的 WEBSERVICE 的状态。



Validation

启用完全验证

Novalidation

未启用完全验证

CEMT SET WORKREQUEST 命令

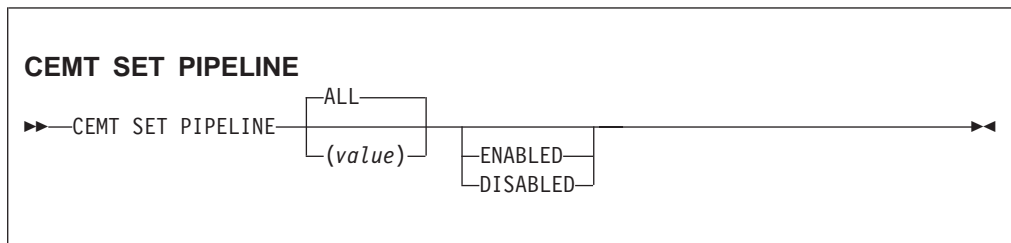
SET WORKREQUEST 命令上的 WORKTYPE 选项有一个新值:

PIPE

指定工作会针对 Web Service 请求而执行。

CEMT SET PIPELINE

使用 SET PIPELINE 命令更改已安装的 PIPELINE 的状态。



PIPELINE(value)

指定 PIPELINE 的 8 个字符的名称。

ENABLED

指定正常处理此 PIPELINE 的入站服务请求。

DISABLED

指定为将要拒绝此 PIPELINE 的入站服务请求。

对 JCICS API 的更改

下列命令有一些 JCICS 等价命令:

```
EXEC CICS FAULT ADD
EXEC CICS FAULT CREATE
EXEC CICS FAULT DELETE
EXEC CICS INVOKE WEBSERVICE
```

对 CICS 提供的事务的更改

注: 对 CEMT 的更改在第 49 页的『对 CEMT 的更改』中描述。

对 CETR 的更改

事务 CETR 现在支持 PI 域。

对统计信息的更改

将提供下列新的 DSECT:

DFHPIPDS

提供 PIPELINE 资源的统计信息。

DFHPIWDS

提供 WEBSERVICE 资源的统计信息

对样本程序的更改

下列程序现在支持 PIPELINE 和 WEBSERVICE 资源:

DFH0STAT, 它是样本统计程序

DFH\$FORA, 它是 DB2[®] 格式化样本程序 (汇编程序)

DFH\$FORP, 它是 DB2 格式化样本程序 (PL/I)

DFH0FORC, 它是 DB2 格式化样本程序 (COBOL)

DFH\$DB2T, 它是 DFH\$FORA、DFH\$FORP 和 DFH0FORC 的 DB2 表定义

DFH\$SQLT, 它是 DB2 表装入实用程序的输入

对 CICS 实用程序的更改

统计信息实用程序 DFHSTUP

DFHSTUP 支持对『对统计信息的更改』中描述的统计信息的更改。

您可以对 DFHSTUP 的 SELECT TYPE 和 IGNORE TYPE 控制参数进行编码的资源类型现在包括:

- WEBSERVICE
- PIPELINE

对问题确定的更改

以下是 CICS Transaction Server for z/OS V3R1 中的新域:

组件标识	描述
PI	管道管理器域

要支持新的域:

- CICS 跟踪函数 (包含 CETR、与跟踪相关的系统初始化参数以及跟踪实用程序) 现在支持列出的组件标识。
- CICS 系统转储包含用于新域的控制块。CICS 提供的转储出口例程支持使用列出的组件标识, 以指定要在转储输出中包括控制块。
- 上面列出的域发出的 CICS 消息使用消息前缀, 它的格式为 DFH*component-ID*。

安全性

新的 SPI 和 CEMT 命令的安全性

新的预定义 RACF[®] 资源名称将使用 SPI 和 CEMT 控制对下列资源的访问:

PIPELINE
WEBSERVICE

新的类别 1 事务

下列新事务供 CICS 内部使用，而不应该从用户终端调用它们。为用于安全性，它们是类别 1 事务。

CPIH
CPIQ
CPIL
CWSQ

迁移

本主题描述 CICS 中的 Web service 如何影响到 CICS Transaction Server for z/OS V3 的迁移。

迁移现有函数

如果您为 CICS 功能部件使用 SOAP，则必须执行很多任务来迁移使用该功能部件的应用程序。CICS Transaction Server for z/OS V3R1 中提供的对 Web service 的支持明显不同于此功能中提供的支持。

SOAP for CICS 功能依赖于对用户编写代码的大量扩展，因此不可能开始基本迁移任务。然而，此处是一些您将需要考虑的问题。

- 考虑使用 Web service 助手构造和解析 SOAP 消息。如果您决定这样做，建议您废弃现有消息适配器，并设计新的包装器程序替换它们，实际上您将不可能重用适配器中的大量代码。
- 如果您使用 SOAP 消息，但是决定不使用 Web service 助手，则您能够为构造和解析消息重用现有的代码。然而，您应该考虑是否使用 CICS 提供的 SOAP 消息处理程序，因为它们设计为使用 SOAP 1.1 和 SOAP 1.2 消息。
- 查看容器的使用。SOAP for CICS 功能使用 BTS 容器，然而 CICS Transaction Server for z/OS V3R1 使用通道容器。您将需要查看您的程序，并更改此功能所需的 BTS 相关命令。您还将需要查看每个容器的名称和用法，因为大多数这些容器已更改。
- 考虑如何迁移管道程序提供的功能。SOAP for CICS 功能中的管道具有固定数量的用户编写的程序，每个程序具有指定的目的。这些程序提供的功能在 CICS Transaction Server for z/OS V3R1 中由 CICS 提供的 SOAP 消息处理程序提供，所以您能够完全省去这些程序。

另一方面，CICS Transaction Server for z/OS V3R1 允许您在管道中定义您所需的足够多的程序。因此，您应该考虑管道程序执行的功能是否应该重新构造，以利用新的框架。

在任何情况下，管道程序于 CICS 通信，以及与另一个管道程序通信的方法已更改，所以您将需要查看这些程序，了解在新的环境中是否可以重用它们。

在 SOAP for CICS 功能中，您可以仅仅具有一个管道用于所有服务提供者应用程序，并且具有另一个管道用于所有服务请求者。在 CICS Transaction Server for z/OS V3R1 中，您可配置很多不同的管道。因此，可能您在管道程序中提供以区分一个应用程序和另一个应用程序的逻辑可被 CICS 资源定义所替换。例如，在服务提供者中，根据 URI 区分应用程序的代码可被 URIMAP 资源的适当集合所替换

共存

如果您使用 CICS 的 SOAP 功能部件，您可以继续使用；CICS Transaction Server for z/OS V3R1 继续完全支持的该功能部件独立于 CICS 中的 Web service。CICS 的 SOAP 功能部件可以与 CICS TS for z/OS V3.1 中的 Web Service 支持相互操作：该功能部件可以是服务请求者也可以是服务提供者。

CICSplex SM 支持

存在对 CICSplex SM 视图和资源表的更改，以在 CICS 中支持 Web Service。

对 CICSplex SM 应用程序编程接口的更改

已引入下列新的资源表：

- 『PIPEDEF 资源表』
- 『PIPELINE 资源表』
- 第 56 页的 『WEBSVDEF 资源表』
- 第 56 页的 『WEBSERV 资源表』

PIPEDEF 资源表

PIPEDEF 资源表包括下列 RDO 属性：

PIPELINE

管道定义名称

STATUS

显示表明是要以 ENABLED（缺省）还是以 DISABLED 状态安装 PIPELINE 的 CVDA 值。

CONFIGFILE

此管道的 HFS 上 255 位字符的标准配置文件名

SHELF

WSBind 目录（结构）的 255 位字符的标准名称

WSDIR

HFS 上 WSBind（pickup）目录的 255 位字符名称

PIPELINE 资源表

PIPELINE 资源表具有下列 SPI 属性：

PIPELINE

管道定义名称

ENABLESTATUS

指定管道的 ENABLE 状态的 CVDA 值

PIPEUSECOUNT

管道使用计数

CONFIGFILE

此管道的 HFS 上 255 位字符的标准配置文件名

SHELF

WSBind 文件的目录（结构）的 255 位字符名称

WSDIR

HFS 上 WSBind (pickup) 目录的 255 位字符名称

WEBSVDEF 资源表

WEBSVDEF 资源表具有下列 RDO 属性:

WEBSERVICE

WEBSERVICE 定义名称

PIPELINE

要安装此 Web Service 的管道

WSBIND

HFS 上 WSBind 文件的 255 位字符的标准名称

WSDLFILE

HFS 上 WSDL 文件的 255 位字符的标准名称

VALIDATION

显示表明是否将执行 WSDL SOAP 消息完全验证的 CVDA 值 (YES 或 NO)

WEBSERV 资源表

WEBSERV 资源表包括下列 SPI 属性:

WEBSERVICE

WEBSERVICE 名称

PIPELINE

安装此 Web Service 的管道

WSBIND

HFS 上 WSBind 文件的 255 位字符的标准名称

WSDLFILE

HFS 上 WSDL 文件的 255 位字符的标准名称

URIMAP

与此 Web Service 关联的动态安装 URIMAP

BINDING

此 Web 文件表示的 WSDL 绑定

ENDPOINT

远程 Web Service 的 ENDPOINT URI

PROGRAM

实施此 Web Service 的 8 位字符 CICS 应用程序名

PGMINTERFACE

表明所指定 PROGRAM 需要输入的位置的 CVDA 值 (CHANNEL 或 COMMAREA)

CONTAINER

如果 PGMINTERFACE 为 CHANNEL, 使用的容器名称

VALIDATIONST

表明是否应该执行 WSDL SOAP 消息完全验证的 CVDA 值 (VALIDATION 或 NOVALIDATION)

LASTMODTIME

HFS 上已部署的 WSBIND 文件最近一次更新的时间

#

STATE

表明 Web Service 状态的 CVDA 值 (DISCARDING、INITING、INSERVICE、UPDATING 或 UNUSABLE)

WEBUSECOUNT

Web Service 使用计数

已更改的资源表

下列资源表已更改:

- 『WORKREQ』

WORKREQ

WORKREQ 资源表现在包含下列属性:

WORKTYPE

正在执行工作的类型。只有 CVDA IIOP 和 SOAP 类型是可用类型。

对 CICSplex SM Web 用户界面的更改

新的 WUI 视图

已引入下列 WUI 视图:

- 『管道定义视图』
- 第 58 页的『管道视图』
- 第 58 页的『Web Service 定义视图』
- 第 59 页的『Web Service 视图』

管道定义视图

已引入新的定义视图集, 称为管道定义, 与新的 PIPEDEF 资源表相关联。此表格视图的视图名称为 EYUSTARTPIPEDEF.TABULAR。而现有 EYUSTARTADMRES 菜单已扩展为包含它。

要打开**管道定义**视图，执行下列操作：

1. 从主菜单单击**管理视图**
2. 单击**基本 CICS 资源管理视图**（或者，可选择单击**完全功能业务应用程序服务（BAS）管理视图**）
3. 单击**CICS资源定义**
4. 下滚并单击**管道定义**

显示**管道定义**视图。此视图包括下列五个操作按钮：

- **创建**
- **更新**
- **除去**
- **安装**
- **添加到资源组**

请参阅第 55 页的『PIPEDEF 资源表』中所列出 PIPEDEF 资源表的属性，以获得字段详细信息。

管道视图

已引入新的视图，称为**管道**，与新的 PIPELINE 资源表相关联。此表格视图的视图名称为 EYUSTARTPIPELINE.TABULAR，而现有 EYUSTARTTCPIPS 菜单已扩展为包含它。

要打开**管道**视图，执行下列操作：

1. 从主菜单单击**CICS操作视图**
2. 下滚并单击**TCP/IP 服务操作视图**
3. 单击**管道**

显示**管道**视图。

请参阅第 55 页的『PIPELINE 资源表』中所列出 PIPELINE 资源表的属性，以获得字段详细信息。

Web Service 定义视图

已引入新的定义视图集，称为**Web Service 定义**，与新的 WEBSVDEF 资源表相关联。此表格视图的视图名称为s EYUSTARTWEBSVDEF.TABULAR。而现有 EYUSTARTADMRES 菜单已扩展为包含它。

要打开**Web Service 定义**视图，执行下列操作：

1. 从主菜单单击**管理视图**
2. 单击**基本 CICS 资源管理视图**（或者，可选择单击**完全功能业务应用程序服务（BAS）管理视图**）
3. 单击**CICS资源定义**
4. 下滚并单击**Web Service 定义**

显示**Web Service 定义**视图。此视图包括下列五个操作按钮：

- **创建**

- 更新
- 除去
- 安装
- 添加到资源组

请参阅第 56 页的『WEBSVDEF 资源表』中所列出 WEBSVDEF 资源表的属性，以获得字段详细信息。

Web Service 视图

已引入新的表格视图，称为 **Web Service**，与新的 WEBSERV 资源表相关联。此表格视图的视图名称为 EYUSTARTWEBSERV.TABULAR，而现有 EYUSTARTTCPIPS 菜单已扩展为包含它。

要打开 **Web Service** 视图，执行下列操作：

1. 从主菜单单击 **CICS** 操作视图
2. 下滚并单击 **TCP/IP 服务操作视图**
3. 单击 **Web Service**

显示 **Web Service** 视图。

请参阅第 56 页的『WEBSERV 资源表』中所列出 WEBSERV 资源表的属性，以获得字段详细信息。

已更改的 WUI 视图

下列 WUI 视图已更改：

- 『工作请求视图』

工作请求视图

下列属性已添加到任务操作视图内的工作请求视图：

WORKTYPE

正在执行工作的类型。只有 IIOP 和 SOAP 是可用类型。

第 3 章 支持 CICS 应用程序发出的 HTTP 客户机请求

您现在可以使用 EXEC CICS 命令，创建通过作为 HTTP 客户机的 CICS 从 CICS 应用程序发出的出站请求。

担当 HTTP 客户机的 CICS 的工具完全集成到了 CICS Web 支持中。您可以使用 CICS 应用程序中的 EXEC CICS 命令打开到服务器的 HTTP 连接、发出请求以及为应用程序的处理接收响应。为实现此目的引进了一些新的 EXEC CICS 命令，并且为一些现有的命令提供了出站和入站选项。

支持 CICS 应用程序发出的 HTTP 客户机请求的优点

作为 HTTP 客户机的 CICS 的 EXEC CICS 命令意味着可以将 CICS 应用程序编写为：

- 使用企业到企业通信的公共协议。
- 使用 HTTP 协议控制硬件或软件（例如，有时可以用这种方式控制打印机）。
- 访问提供一些信息项的 HTTP 应用程序（例如，共享价格）和获取此信息以在应用程序中使用。

CICS Web 支持进程中的全局用户出口允许您为来自 CICS 的出站 HTTP 请求指定代理服务器的用法，以及应用安全策略。

需求

要支持此功能，没有特别的硬件或软件需求。

相关信息

第 311 页的第 27 章，『CICS 操作环境』

作为 HTTP 客户机的 CICS 的 HTTP 请求和响应处理

对于作为 HTTP 客户机的 CICS，CICS 是 Web 客户机，并且与 HTTP 服务器通信。用户编写的应用程序通过 CICS 向 HTTP 服务器发送请求，并从它那里接收响应。CICS 维持与服务器的持续连接。在应用程序发出的命令上使用会话标记来识别连接。

发出 HTTP 请求并接收响应的应用程序必须使用 EXEC CICS WEB API 命令以明确地指示与服务器的交互。可以使用支持 Web 的应用程序发出 HTTP 请求，然后处理结果以将信息提供给不支持 Web 的应用程序。

对于 HTTP 客户机请求，CICS 可以为请求和响应执行代码页转换。但是除此之外，CICS 不提供任何特殊过程来分析或转换发送到作为 HTTP 客户机的 CICS 的数据，而是与提供给作为 HTTP 服务器的 CICS 的过程一样。应该将发出 HTTP 请求的应用程序设计为处理 CICS 从服务器接收的任何内容以响应该请求，这些内容可能包含错误响应、重定向到另一个 URL、嵌入式超文本链接、HTML 表单、图像源或者请求应用程序的操作的其他项。

下面描述执行适用于作为 HTTP 客户机的 CICS 的处理：

1. 应用程序通过 **CICS** 开始与 **HTTP** 服务器的连接。应用程序通过发出 **WEB OPEN** 命令来执行该操作。可以引用您创建的 **URIMAP** 资源定义为连接指定方案和主机名，或者应用程序也可以提供此信息。
2. **CICS** 建立与服务器之间的连接。使用应用程序提供的信息，**CICS** 打开套接字上的 **TCP/IP** 连接并且联系服务器。建立 **TCP/IP** 连接时，**CICS** 向应用程序返回会话标记以唯一地识别连接。在有关该连接的应用程序发出的所有剩余命令上使用此会话标记。
3. 应用程序可为它的请求编写 **HTTP** 头。可以使用 **WEB WRITE HTTPHEADER** 命令构建用户编写的 **HTTP** 头并且将它们存储起来以供发送。
4. 应用程序指定请求行的组成部分。使用 **WEB SEND** 或 **WEB CONVERSE** 命令指定请求方法、路径信息以及查询字符串。请求的 **HTTP** 版本由 **CICS** 提供。
5. 应用程序可为它的请求产生实体主体。在 **WEB SEND** 命令 或 **WEB CONVERSE** 命令上指定请求的内容。
6. 应用程序开始请求的传输。应用程序发出 **WEB SEND** 或 **WEB CONVERSE** 命令时，请求传递给 **CICS** 以通过会话标识指定的连接进行发送。
7. **CICS** 生成一些必需的 **HTTP** 头并将它们添加到请求，然后将请求发送到服务器。
8. 服务器接收并处理请求，并提供响应。**CICS** 将响应传递给应用程序。
9. 应用程序检验响应。可以使用 **WEB READ HTTPHEADER** 命令或 **HTTP** 头浏览命令检验响应的头。**WEB RECEIVE** 或 **WEB CONVERSE** 命令接收可以由应用程序处理的响应的主体（如果有的话），以及响应的状态码和状态文本。
10. 应用程序可以发出后继请求和响应。如果服务器支持持续连接，则由会话标记识别的连接仍保持打开，以供后继请求使用。
11. 应用程序开始关闭与服务器的连接。完成所有请求和响应时，应用程序发出 **WEB CLOSE** 命令，并且 **CICS** 关闭其 **TCP/IP** 连接。如果应用程序不发出 **WEB CLOSE** 命令，则在任务结束时关闭连接。

此过程期间，当消息进入和离开 **CICS** 环境时，经常需要进行代码页转换，以使得 **CICS Web** 支持处理和用户编写的应用程序（通常使用 **EBCDIC** 编码）能够与 **HTTP** 服务器（通常使用 **ASCII** 编码）进行通信。

会话标记

会话标记是一个 8 个字节的二进制值，它唯一地识别作为 **HTTP** 客户机的 **CICS** 和 **HTTP** 服务器之间的连接。为每个连接使用会话标记意味着 **CICS Web** 支持可以通过不同的任务管理多个与服务器的连接，也意味着应用程序可以控制多个连接。

开始连接，以响应用户应用程序发出的 **WEB OPEN** 命令。成功完成 **WEB OPEN** 命令时后返回会话标记，并且在与该连接相关的应用程序发出的所有 **EXEC CICS WEB** 命令上使用该会话标记。使用连接时，用户应用程序可以向服务器发送 **HTTP** 客户机请求，并从该服务器接收响应。连接可以持续用于请求和响应的多次交换，直到应用程序或服务器选择终止连接为止。第 89 页的『**CICS Web** 支持如何处理持续连接』提供有关 **CICS Web** 支持如何处理持续连接以及如何终止它们的更多详细信息。

作为 **HTTP** 客户机的 **CICS** 使用的命令以及要实现该目的所必需的会话标记为：

- **WEB SEND** 命令，它构成 **HTTP** 客户机请求并向服务器发送该请求。
- **WEB RECEIVE** 命令，它从服务器接收响应。
- **WEB CONVERSE** 命令，它结合 **WEB SEND** 和 **WEB RECEIVE** 命令的操作。

- 处理 HTTP 头的命令：WEB READ HTTPHEADER、WEB WRITE HTTPHEADER、WEB STARTBROWSE HTTPHEADER、WEB READNEXT HTTPHEADER、WEB ENDBROWSE HTTPHEADER。
- WEB CLOSE 命令，它结束连接。

如果服务器终止连接，则应用程序就无法使用该连接发送任何后继请求，但是在应用程序终止连接之前，它仍可以读服务器发送的响应。在应用程序发出 WEB CLOSE 命令之前，仍可以在命令上使用会话标记以访问该数据。发出 WEB CLOSE 命令后，连接应用的会话标记就不再有效了。如果应用程序不发出 WEB CLOSE 命令，则在任务结束时关闭连接。

对 CICS 外部的更改

对资源定义的更改

URIMAP 定义可用作简便方法来通过作为 HTTP 客户机的 CICS 发出 HTTP 请求。

作为 HTTP 客户机的 CICS 的 URIMAP 定义

URIMAP 定义可用于避免在发出 HTTP 客户机请求的应用程序中指定 URL 或证书标号这样的信息。URIMAP 定义的属性在第 107 页的『对资源定义的更改』中列出。使用 USAGE(CLIENT) 定义作为 HTTP 客户机的 CICS 的 URIMAP 定义，并且可指定 SCHEME、HOST、PATH、CERTIFICATE 和 CIPHERS 属性。应用程序可在 WEB OPEN 和 WEB SEND 命令上指定 URIMAP 定义，以使用它的相关信息。

URIMAP 定义还用于作为 HTTP 服务器的 CICS 和 Web Service。

对监控控制表 DFHMCT 的更改

性能类记录的 DFHWEBB 组中新的监控字段 331-338 添加到监控控制表中。这些字段监控作为 HTTP 客户机的 CICS 的操作。

对应用程序编程接口（HTTP 客户机请求）的更改

新的以及更改过的命令

为作为 HTTP 客户机的 CICS 提供以下新的 EXEC CICS WEB 命令：

- EXEC CICS WEB OPEN
- EXEC CICS WEB CONVERSE
- EXEC CICS WEB CLOSE

当用于作为 HTTP 客户机的 CICS 时，以下 EXEC CICS WEB 命令具有新的选项范围：

- EXEC CICS WEB SEND
- EXEC CICS WEB RECEIVE

下列新的 EXEC CICS 命令可用于作为 HTTP 客户机的 CICS 和作为 HTTP 服务器的 CICS，并在第 114 页的『对应用程序编程接口（一般 CICS Web 支持增强）的更改』中有所描述：

- EXEC CICS WEB PARSE URL

- EXEC CICS CONVERTTIME

有一些可用于大部分剩余 EXEC CICS WEB 命令选项的更改。更改的命令可用于作为 HTTP 客户机的 CICS 和作为 HTTP 服务器的 CICS。本部分中描述对以下命令的更改:

- 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

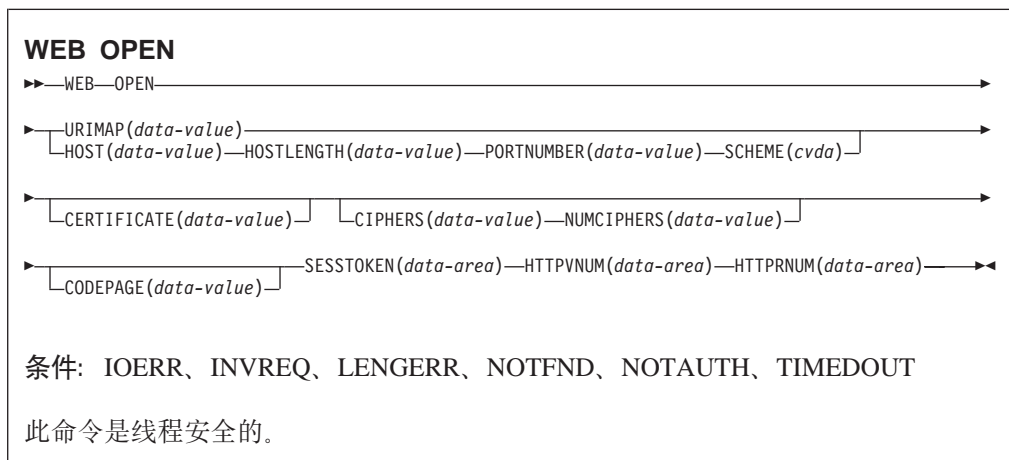
第 114 页的『对应用程序编程接口（一般 CICS Web 支持增强）的更改』中描述对以下命令的更改:

- EXEC CICS WEB EXTRACT
- EXEC CICS FORMATTIME

WEB FORMFIELD 命令和 WEB RETRIEVE 命令不能用于作为 HTTP 客户机的 CICS。

WEB OPEN

为作为 HTTP 客户机的 CICS 打开与服务器的连接。



描述

WEB OPEN 允许应用程序通过 CICS Web 支持打开与因特网上的 HTTP 服务器上指定主机的连接。可以使用预设置的 URIMAP 定义的主机名和方案，该定义还提供请求的缺省路径。

打开连接时，应用程序可以向服务器发送 HTTP 客户机请求，并从该服务器接收响应。当打开连接时，CICS 查询服务器的 HTTP 版本，并将此信息用于随后的通信。CICS 还将 HTTP 版本信息返回给应用程序，如果您打算编写 HTTP 头或发送分块信息，则需要检查该信息。

WEB OPEN 命令导向 XWBOPEN 用户出口，这样，如果必要，就可以通过代理服务器连接到服务器。

选项

CERTIFICATE(*data-value*)

指定要在 SSL 握手期间用作 SSL 客户机证书的 X.509 证书的标号。证书标号最多可以由 32 个字母数字字符组成。仅当指定 SCHEME(HTTPS) 时，此选项才相关。如果指定 SCHEME(HTTPS)，但省略 CERTIFICATE 选项，则使用 CICS 区域用户标识的密钥环中定义的缺省证书。该证书必须存储在外安全性管理器数据库中的密钥环中。

CIPHERS(*data-value*)

指定多达 56 个十六进制数字的字符串，该字符串被解释为多达 28 个 2 位的密码套件代码列表。当此连接的 SSL 活动时使用密码套件代码，所以仅当指定 SCHEME(HTTPS) 时，此选项才相关。它们表明要用于此连接的加密方法。

使用 NUMCIPHERS 选项指定列表中的密码套件代码数。实际使用的代码取决于 ENCRYPTION 系统初始化参数已指定的加密级别。如果您指定活动加密级别的缺省列表中不存在的任何加密代码，则忽略它们。

您可以指定 URIMAP 选项以从现有 URIMAP 定义直接使用此信息，在此情况下，不需要 CIPHERS 选项。您可能仍然指定 CIPHERS 选项，并且使用您为此选项指定的任何代码覆盖 URIMAP 定义中的设置。

CODEPAGE(*data-value*)

指定适合于应用程序使用的代码页（通常为 EBCDIC）。代码页名称可以长达 8 个字母数字字符。缺省值为本地 CICS 区域的缺省代码页，如 LOCALCCSID 系统初始化参数中指定的那样。在此连接期间应用该代码页。当服务器返回对 HTTP 请求的响应时，如果请求转换（这是缺省值），则 CICS 将请求主体转换为此代码页，然后再将它传递给应用程序。

HOST(*data-value*)

指定您要连接的服务器的主机名。

IPv4 地址可用作主机名，但 IPv6 地址不受支持。

如果端口号是必需的，则不要将它与主机名包括在一起，而使用 PORTNUMBER 选项来指定它。

HOSTLENGTH(*data-value*)

以全字二进制值指定主机名的长度。

HTTPRNUM(*data-area*)

以半字二进制数值返回服务器的 HTTP 版本的发行版号。（HTTPVNUM 返回版本号。）例如，如果服务器的级别为 HTTP/1.0，则 HTTPRNUM 返回 0。

HTTPVNUM(*data-area*)

以半字二进制数值返回服务器的 HTTP 版本的版本号。（HTTPRNUM 返回发行版号。）例如，如果服务器的级别为 HTTP/1.0，则 HTTPVNUM 返回 1。

打开与服务器的连接时，CICS 会获取 HTTP 版本信息。如果服务器不提供 HTTP 版本信息，则 CICS 假设它的级别为 HTTP/1.0。

如果您的应用程序编写的 HTTP 头可能不适用于 HTTP/1.0 级别的服务器，或者如果您打算将分块消息发送到服务器（HTTP/1.0 级别的服务器无法接收该消息），则您的应用程序还应该查询 HTTP 版本信息。

注: CICS 对低于 HTTP/1.0 级别的服务器不提供任何特殊支持。CICS 的行为就象这些服务器所处的级别都是 HTTP/1.0, 并返回 HTTP/1.0 作为 HTTP 版本。

NUMCIPHERS(*data-value*)

作为半字二进制值, 指定您为 CIPHERS 选项指定的密码套件代码数。

PORTNUMBER(*data-value*)

以全字二进制值指定端口号。仅当端口号不是指定方案的缺省值时, 才需要指定端口号。对于 HTTP, 缺省端口号为 80, 而对于 HTTPS, 缺省端口号为 443。

SCHEME(*cvda*)

指定连接到服务器要使用的方案, 它可以带 SSL, 也可以不带 SSL。CVDA 值有:

HTTP 是不带 SSL 的 HTTP 协议。

HTTPS

是 HTTPS 协议, 该协议是带 SSL 的 HTTP。如果使用 HTTPS, 则必须为 SSL 启用 CICS 地址空间。

SESSTOKEN(*data-area*)

返回 8 个字节的二进制值会话标记, 它唯一地识别 CICS 和服务器之间的这个连接。连接成功打开时会返回它。必须在与此连接相关的所有 CICS WEB 命令上使用该会话标记。第 62 页的『会话标记』说明会话标记的用途。

URIMAP(*data-value*)

指定提供以下信息的 URIMAP 定义的名称 (最多 8 个字符):

- 要用于连接服务器的方案。
- 您要连接的服务器的主机名。
- 端口号 (如果需要)。
- URI 的路径部分, 它表示您要访问的服务器上的资源。此路径成为与此连接相关的 WEB SEND 或 WEB CONVERSE 命令的缺省路径, 但可以通过在 WEB SEND 或 WEB CONVERSE 命令上指定另一个路径来覆盖此路径。
- 要用作 SSL 客户机证书的 X.509 证书的标号 (如果需要)。
- 可用于此连接的密码套件代码。

如果指定了 URIMAP 选项, 则不要指定 CERTIFICATE、HOST、HOSTLENGTH、PORTNUMBER、PORTLENGTH 或 SCHEME 选项。在命令中省略或指定 CIPHERS 和 NUMCIPHERS 选项; 如果指定, 它们在 URIMAP 定义中覆盖这些设置。URIMAP 定义必须用于作为 HTTP 客户机的 CICS (指定 USAGE(CLIENT))。

WEB CLOSE

为作为 HTTP 客户机的 CICS 关闭与服务器的连接。

WEB CLOSE

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

条件: NOTOPEN

此命令是线程安全的。

描述

WEB CLOSE 使得应用程序能关闭与服务器的连接。会话标记识别要关闭的连接。连接关闭时，连接应用的会话标记不再能使用。需要会话标记以从服务器接收响应并读取该响应的 HTTP 头，所以直到完成所有与服务器的交互以及与它发送的响应的交互，才应该发出 WEB CLOSE 命令。此命令释放此连接所涉及的 CICS 资源。

WEB CLOSE 命令不会让 CICS 通知服务器应该终止连接。它只是让 CICS 关闭客户端上的连接。在您要使用连接发出的最终请求上，您应该在 WEB SEND 或 WEB CONVERSE 选项上指定 CLOSESTATUS(CLOSE) 选项。指定此选项时，CICS 会在请求上写一个 Connection: close 头，或者对于 HTTP/1.0 级别的服务器，则省略 Connection: Keep-Alive 头。头中的信息意味着在发送最后的响应后服务器可以立即关闭它与您的连接，而不是等待并观察您是否会继续发送其它请求，直至超时为止。

在发出 WEB CLOSE 命令之前，也可以关闭服务器请求的连接。如果您需要测试服务器是否已请求终止连接，使用 WEB READ HTTPHEADER 命令在服务器发出的最后一条消息中查找 Connection: close 头。

如果服务器请求终止连接，则在发出 WEB CLOSE 命令之前，与该连接相关的数据仍可以在 CICS 中使用。可用数据包括从服务器接收的最新消息，以及用于打开连接的参数（如规则和服务器的主机名）。当服务器终止连接时，应用程序无法：

- 使用 WEB SEND 或 WEB CONVERSE 命令发送有关此连接的进一步请求。
- 使用 WEB WRITE HTTPHEADER 命令写 HTTP 头。

然而，应用程序仍然可以：

- 使用 WEB RECEIVE 命令接收响应。
- 使用 WEB READ HTTPHEADER 和 HTTP 头浏览命令检查 HTTP 头。
- 使用 WEB EXTRACT 命令抽出连接信息。

发出 WEB CLOSE 命令时，清除与连接相关的数据。

如果应用程序没有发出 WEB CLOSE 命令，则在任务结束时 CICS 会清除与该连接相关的数据，并且如果连接还没有关闭，则关闭它。

选项

SESSTOKEN(*data-value*)

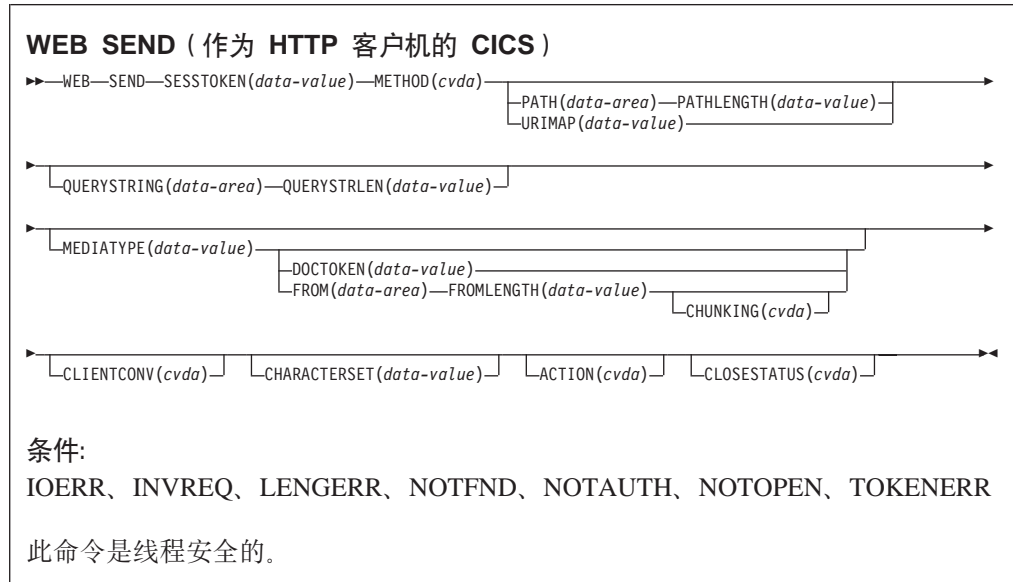
指定 8 个字节的二进制值会话标记，它唯一地识别 CICS 和服务器之间的连接。

WEB OPEN 命令为作为 HTTP 客户机的 CICS 返回此值。对话话标记识别的连接

发出 WEB CLOSE 命令时, CICS 结束该连接并且清除与连接关联的数据, 并且使应用程序无法再进一步使用该会话标记。第 62 页的『会话标记』说明会话标记的用途。

WEB SEND (客户机)

作为 HTTP 客户机的 CICS 使用 CICS Web 支持发送 HTTP 请求。



描述

作为 HTTP 客户机的 CICS 使用 WEB SEND 将 HTTP 请求发出给服务器。此命令上必须包括会话标记。

提示: 对于作为 HTTP 客户机的 CICS, 可以使用 CONVERSE 命令来替代 WEB SEND 命令 (后跟 WEB RECEIVE 命令)。但是要记住, WEB CONVERSE 命令不支持分块的传输编码, 因为这需要一系列的发送操作, 而 WEB CONVERSE 命令只提供单个发送操作。

选项

ACTION(cvda)

此选项用于指定应该如何发送消息。作为 HTTP 客户机的 CICS 适用的 CVDA 值有:

EXPECT

使得 CICS 将 Expect 头与请求行和请求头一起发送, 并在将消息体发送到服务器之前等待 100-Continue 响应。如果接收到的不是 100-Continue 响应, 则 CICS 会通知应用程序并且取消发送。如果等待一段时间之后未接收到任何响应, 则 CICS 无论如何都会发送消息体。

仅当您的请求有消息体时才能使用此选项。

CHARACTERSET(*data-value*)

指定字符集，CICS 在发送请求的实体主体之前将它转换成的字符集。字符集的名称最多可以由 40 个字母数字字符组成，包括适当的标点符号。CICS 不支持 IANA 命名的所有字符集。

要执行实体主体转换，必须将 CLIENTCONV 选项指定为（或允许缺省为）CLICONVERT。指定 NOCLICONVERT 抑制实体主体的转换。如果请求转换，并且如果未指定 CHARACTERSET 属性，则使用 ISO-8859-1 作为缺省值。

CHUNKING(*cvda*)

用于以分块的形式（称为分块的传输编码）发送消息时控制消息发送。未指定该选项时的缺省值为不在使用分块的传输编码。

可以用对应用程序而言最便利的任何一种方法将分块消息的内容划分成块。不能从 CICS 文档直接构成分块消息的主体，因此不能使用 DOCTOKEN 选项。

对于每一块消息，使用带有 CHUNKING(CHUNKYES) 的不同 WEB SEND 命令。使用 FROM 选项指定数据的块，并使用 FROMLENGTH 选项指定块的长度。在序列的第一个 WEB SEND 命令中为消息指定其他选项（如 CLOSESTATUS 选项），但在以后的消息中不指定它们。当已发送最后一个数据块时，再指定一个带 CHUNKING(CHUNKEND) 但不带 FROM 或 FROMLENGTH 选项的 WEB SEND 命令。CICS 则将此空块发送给接收方以结束分块消息。

CVDA 值有：

CHUNKNO

消息不使用分展块的传输编码。如果未指定 CHUNKING 选项，则此为缺省值。

CHUNKYES

分块的传输编码正在进行中。FROM 选项指定的数据表示消息块。

CHUNKEND

分块的传输编码已完成。没有为该发送指定数据。CICS 向接收方发送空块以完成分块消息。

注：

1. 方法（METHOD 选项）必须适用于分块的传输编码。
2. 当您已开始发送分块消息的各部分时，直到发送了最后的空块且分块消息变成完整的消息时，应用程序才能发送不同的消息或接收项。

CLOSESTATUS(*cvda*)

指定消息中是否应该包含带“close”连接选项的 Connection 头（Connection: close）。缺省情况下消息中不包括该头。CVDA 值有：

CLOSE

使 CICS 为此请求写 Connection: close header。该头通知服务器应该在服务器将它的响应发送给请求之后关闭连接。（对于 HTTP/1.0 级别的服务器，CICS 通过省略 Connection: Keep-Alive 头达到相同的效果。）

如果在使用分块的传输编码，则可以在第一个消息块上指定 CLOSESTATUS(CLOSE) 选项，以通知服务器在分块消息变成完整的消息并且发送了响应后应该关闭连接。

如果不在使用分块的传输编码，并且 WEB SEND 命令上指定了 CLOSESTATUS(CLOSE) 选项，则在建立新的连接后才能向服务器发送后继消息。

NOCLOSE

意味着 Connection: close header 不用于此请求。如果服务器识别为 HTTP/1.0，则 CICS 会发送带“Keep-Alive”连接选项的 Connection 头（Connection: Keep-Alive）来通知期待持续连接。

CLIENTCONV(*cvda*)

指定在发送前，CICS 是否将 HTTP 请求的实体主体，从应用程序使用的代码页转换到适合于接收方使用的字符集。如果省略此选项，缺省情况下转换实体主体，除非指定非文本介质。CVDA 值有：

CLICONVERT

CICS 将 HTTP 请求的实体主体从应用程序使用的代码页转换为您为服务器标识的字符集。您可以在此命令上使用 CHARACTERSET 选项，以指定使用的字符集。如果请求转换，但没有指定代码页，则缺省情况下 CICS 将实体主体转换为 ISO-8859-1 字符集。（应用程序使用的代码页是在 WEB OPEN 命令上为连接标识的）。

NOCLICONVERT

CICS 不转换 HTTP 请求的实体主体，而是以应用程序使用的代码页将该实体主体传递给服务器，如此连接的 WEB OPEN 命令上所标识。

DOCTOKEN(*data-value*)

指定要作为消息体发送的文档的 16 字节二进制标记。根据 *CICS Application Programming Guide* 中的描述，必须使用 CICS Document 接口（EXEC CICS DOCUMENT CREATE、INSERT 和 SET 命令）创建文档。FROM 选项提供另一个方法来创建消息体。

不能从 CICS 文档构成分块消息的主体，因此 DOCTOKEN 选项不能用于分块的传输编码。

CICS 文档无法转换成 UTF-8 和 UTF-16 字符编码。

FROM(*data-area*)

指定保存消息体的数据缓冲区。消息体是由应用程序来构建的。当您指定 FROM 选项时，使用 FROMLENGTH 选项来指定数据缓冲区的长度。DOCTOKEN 选项提供另一个方法来创建消息体，但该选项无法用于分块消息体。

FROMLENGTH(*data-value*)

为 FROM 选项上提供的数据缓冲区（消息体）指定长度（以全字二进制值表示）。正确地表示此值很重要，这是因为不正确的数据长度可能会导致接收消息产生问题。

MEDIATYPE(*data-value*)

指定提供的任何消息体的数据内容，例如 text/xml。介质类型最多可以有 56 个字母数字字符，包括适当的标点符号。CICS 检查介质类型的格式是否正确，但不针对数据内容检查介质类型是否有效。

对于需要主体的请求，您必须指定 MEDIATYPE 选项，并且它没有缺省值。

METHOD(*cvda*)

为请求指定 HTTP 方法。

此命令支持 GET、HEAD、POST、PUT、TRACE、OPTIONS 和 DELETE 方法。然而，某些 HTTP 服务器（特别是 HTTP/1.0 服务器）可能不能实施所有这些方法。

CICS 禁止在不适用的情况下发送方法的消息体，并在适用的情况下需要为方法发送消息体。已分块 transfer-coding 与没有请求主体的方法无关。CVDA 值有：

GET 从服务器获取资源。不允许请求主体。

HEAD 获取 HTTP 头，但是不获取资源的响应主体。不允许请求主体。

POST 将数据发送到服务器。需要请求主体。

PUT 创建或修改服务器上的资源。需要请求主体。

TRACE

跟踪您发送到服务器的请求的路由。不允许请求主体。

OPTIONS

获取有关服务器的信息。允许请求主体，但是主体没有定义的用途。如果您使用请求主体，则必须指定介质类型。

DELETE

删除服务器上的资源。不允许请求主体。

PATH(*data-area*)

为应用程序需要访问的服务器中的特定资源指定路径信息。

如果 URIMAP 选项用于为此连接在 WEB OPEN 命令上指定现有 URIMAP 定义，则 该 URIMAP 定义中所指定的路径是 WEB SEND 命令的缺省路径。在这些情况下，如果您未在 WEB SEND 命令上指定路径信息，则使用来自 URIMAP 定义的路径。如果您指定的路径与 URIMAP 定义中给定的路径不同，则此路径覆盖来自 URIMAP 定义的路径。

如果未在 WEB OPEN 命令上使用 URIMAP 选项，则没有缺省路径，那么您必须提供路径信息。使用 WEB PARSE URL 命令可以从已知 URL 抽取路径信息。

除了使用 PATH 选项来提供路径信息之外，您还可以在 WEB SEND 命令上使用 URIMAP 选项来指定 URIMAP 定义，直接从该 URIMAP 定义采用路径信息。

PATHLENGTH(*data-value*)

以全字二进制值指定路径的长度。如果您要通过 PATH 选项来提供路径信息，则需要指定 PATHLENGTH 选项。如果您使用 WEB PARSE URL 命令来对 URL 进行语法分析，则会返回路径长度信息。

QUERYSTRING(*data-area*)

指定要作为请求的一部分提供给服务器的查询字符串。您不需要在查询字符串的开始处包含问号 (?)；如果您没有包含问号，则在构造请求时 CICS 会自动为您补上。如果查询字符串中包含转义字符，则 CICS 以转义字符的转义格式将它们传递给服务器。

QUERYSTRLEN(*data-value*)

以全字二进制值指定 QUERYSTRING 选项上所提供的查询字符串的长度。

SESSTOKEN(*data-value*)

指定 8 个字节的二进制值会话标记，它唯一地识别 CICS 和服务器之间的连接。WEB OPEN 命令为作为 HTTP 客户机的 CICS 返回此值。

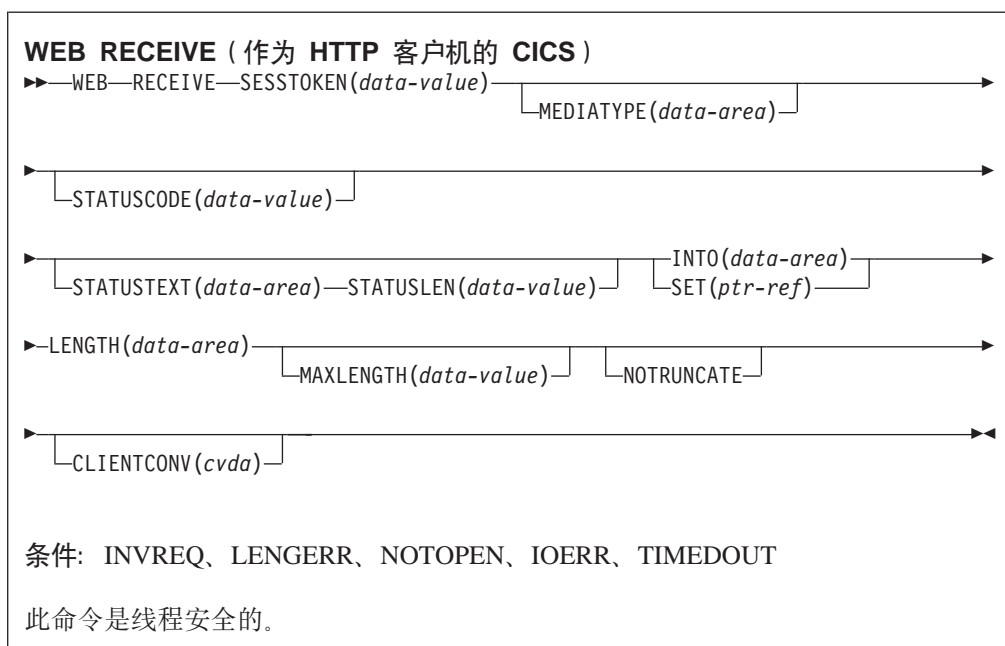
URIMAP(*data-value*)

指定 URIMAP 定义的名称（最多 8 个字符），该定义提供应用程序需要访问的服务器中特定资源的路径信息。URIMAP 定义必须用于作为 HTTP 客户机的 CICS（指定 USAGE(CLIENT)）。它的 HOST 属性必须与为此连接在 WEB OPEN 命令上指定的 URIMAP 定义的 HOST 属性相同，或必须与为此连接在 WEB OPEN 命令上的 HOST 选项中指定的主机名相同。在 WEB SEND 命令上指定的 URIMAP 定义仅应用于此请求。

如果指定了 URIMAP 选项，则不要指定 PATH 或 PATHLENGTH 选项。

WEB RECEIVE (客户机)

为作为 HTTP 客户机的 CICS 接收 HTTP 响应。



描述

作为 HTTP 客户机的 CICS 的 WEB RECEIVE 接收服务器所发出 HTTP 响应的主体。可以分别使用 WEB READ HTTPHEADER 命令或 HTTP 头浏览命令检验 HTTP 响应头。此命令上必须包括会话标记。

在此命令上指定入局消息的代码页转换。

注: 用于为启动用户应用程序的事务所指定的 RTIMOUT 值表明应用程序准备等待接收入局消息的时间。（在事务概要定义上指定 RTIMOUT）。当 RTIMOUT 指定的时期到期后，CICS 向应用程序返回 TIMEDOUT 响应。RTIMOUT 值为零意味着应用程序准备无限地等待下去。事务概要定义上的 RTIMOUT 的缺省设置为 0，因此为要发出 HTTP 客户机请求的应用程序检查和更改该设置是很重要的。

提示: 对于作为 HTTP 客户机的 CICS，可以使用 CONVERSE 命令来替代 WEB SEND 命令（后跟 WEB RECEIVE 命令）。

选项

CLIENTCONV(*cvda*)

指定 CICS 是否将响应的实体主体从服务器使用的字符集转换到适合于应用程序使用的代码页。缺省为转换实体主体。

CLICONVERT

CICS 将响应的实体主体从服务器使用的字符集转换为您为应用程序标识的代码页。

NOCLICONVERT

CICS 不转换响应的实体主体，而是以服务器使用的字符集将该实体主体传递给应用程序。

您不需要在 WEB RECEIVE 命令上为作为 HTTP 客户机的 CICS 指定字符集和应用程序代码页。如果需要代码页转换，CICS 通过查看消息的 Content-Type 头识别服务器使用的字符集。如果头不提供此信息，或者如果 CICS 不支持指定的字符集进行代码页转换，则使用字符集 ISO-8859-1。对于应用程序的代码页，使用本地 CICS 区域的缺省代码页（由 LOCALCCSID 系统初始化参数指定），还可以使用 WEB OPEN 命令中指定的备选 EBCDIC 代码页。

INTO(*data-area*)

指定要包含正在接收的数据的缓冲区。

LENGTH(*data-area*)

指定全字二进制变量，它设置为 CICS 已返回给应用程序的数据量。

- 如果不指定 NOTRUNCATE 选项，则现在已废弃消息中出现的更多任何的数据。如果出现更多数据，则返回 RESP2 值为 57 的 LENGERR 响应。
- 如果指定 NOTRUNCATE 选项，则保留其他的任何数据。如果其他数据可用，则返回 RESP2 值为 36 的 LENGERR 响应。NOTRUNCATE 选项的描述将告诉您这种情况下要采取的操作。

MAXLENGTH(*data-value*)

以全字二进制值指定 CICS 要传递给应用程序的最大数据量。无论是否指定 INTO 或 SET 选项，都会应用 MAXLENGTH 选项来接收数据。如果已使用分块的传输编码发送了数据，则 CICS 将这些块汇编成单个消息，然后再将该消息传递给应用程序，因此 MAXLENGTH 选项应用于分块消息的总长度，而不是每个独立的块。发生任何代码页转换后测量数据。

MEDIATYPE(*data-area*)

指定 56 个字符的 *data-area* 来接收主体的介质类型（即，数据内容的类型），例如 text/xml。

NOTRUNCATE

指定当可用的数据长度超出 MAXLENGTH 选项上请求的长度时，不是立即废弃剩余的数据，而是保留数据，供后继 RECEIVE 命令来获取。（如果没有再发出 RECEIVE 命令，则在事务终止期间会废弃数据。）

带 SET 选项但不带 MAXLENGTH 选项的单条 RECEIVE 命令接收所有剩余的数据，而不管它的长度有多长。或者，您也可以使用一连串 RECEIVE 命令（带 NOTRUNCATE 选项）接收相应块中的剩余数据。继续发出 RECEIVE 命令直到您不再获得 LENGERR 响应。

SET(*ptr-ref*)

指定要设置为接收到的数据地址的指针引用。在发出下一个接收命令或任务结束之前，该指针引用一直有效。

SESSTOKEN(*data-value*)

指定 8 个字节的二进制值会话标记，它唯一地识别 CICS 和服务器之间的连接。WEB OPEN 命令为作为 HTTP 客户机的 CICS 返回此值。

STATUSCODE(*data-value*)

指定 data-area 以接收服务器发送的 HTTP 状态码。该代码是二进制半字值。示例有 200（正常）或 404（未找到）。接收状态码是可选的，但是在下列情况下您应该总是接收和检查状态码：

- 如果您打算现在或者在将来连接时向服务器发送相同的请求。
- 如果您打算使用此连接向服务器发送更多请求。
- 如果您的应用程序要根据响应中接收到的信息执行任何进一步处理。

STATUSTEXT(*data-area*)

指定 data-area 以接收服务器返回的用来描述状态码的任何文本。文本称为原因短语。示例有“OK”（随附 200 状态码）或者“Bad Request”（随附 400 状态码）。STATUSLEN 选项提供文本允许的长度。

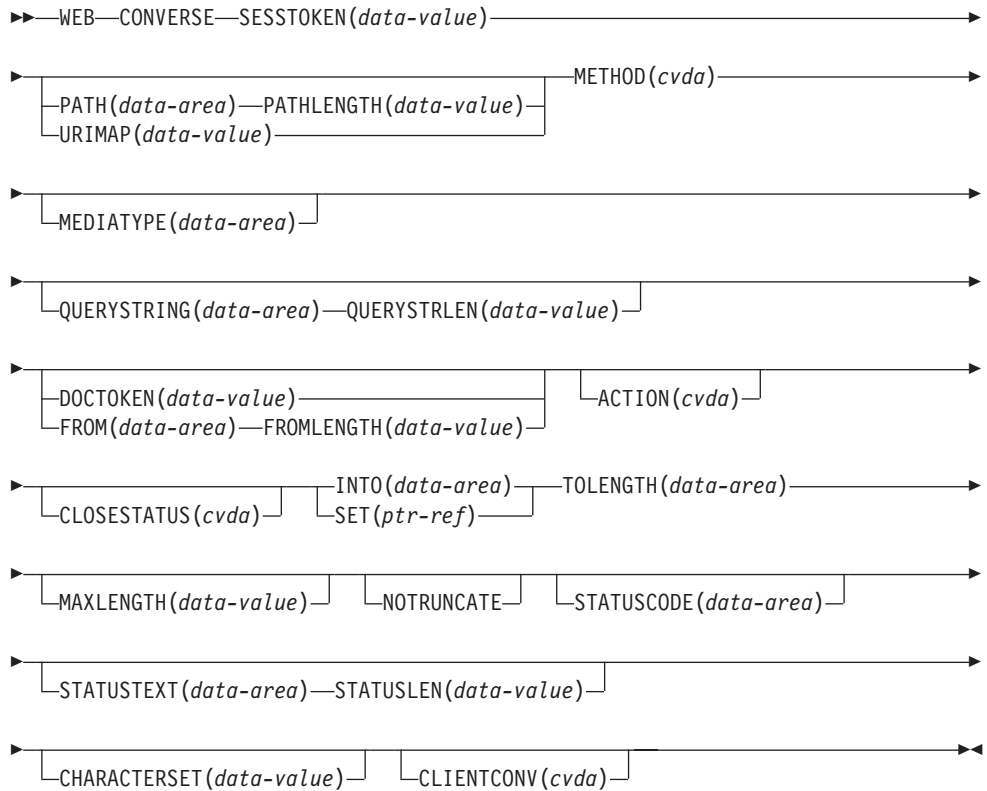
STATUSLEN(*data-value*)

以全字二进制值指定 data-area 的长度，来接收服务器返回的描述状态码（STATUSTEXT 选项）的任何文本。文本称为原因短语。建议为 HTTP 使用的大多数原因短语都是简短的，但是这里建议使用 256 个字符的 data-area 长度，以防服务器用更详细的信息替换建议的原因短语。

WEB CONVERSE

作为 HTTP 客户机的 CICS 使用单条命令发送 HTTP 请求，并且从服务器接收响应。作为 HTTP 客户机的 CICS 可以用该方法代替使用 WEB SEND 命令和 WEB RECEIVE 命令。

WEB CONVERSE



条件:

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

此命令是线程安全的。

描述

WEB CONVERSE 使应用程序能组成并发送 HTTP 客户机请求，并从服务器接收响应。此命令上必须包括会话标记。

- 使用已使用 WEB OPEN 命令打开的连接发出 **HTTP 客户机请求**。WEB CONVERSE 命令可用于代替 WEB SEND 命令，以组成并发送请求。
- 服务器发出的响应由 CICS Web 支持接收并且传递给应用程序。WEB CONVERSE 命令可用于代替 WEB RECEIVE 命令，以使应用程序等待并接收 HTTP 响应。可以分别使用 WEB READ HTTPHEADER 命令或 HTTP 头浏览命令检验 HTTP 响应头。

注: 用于为启动用户应用程序的事务所指定的 RTIMOUT 值表明应用程序准备等待接收入局消息的时间。(在事务概要定义上指定 RTIMOUT)。当 RTIMOUT 指定的时期到期后，CICS 向应用程序返回 TIMEDOUT 响应。RTIMOUT 值为零意味着应用程序准备无限地等待下去。事务概要定义上的 RTIMOUT 的缺省设置为 0，因此为要发出 HTTP 客户机请求的应用程序检查和更改该设置是很重要的。

WEB CONVERSE 命令不支持请求的分块的传输编码，因为这需要一系列的发送操作，而 WEB CONVERSE 命令只提供单个发送操作。如果您要发送分块消息，则使用 WEB SEND 命令来发送它，并使用 WEB RECEIVE 命令来接收它。如果服务器发送分块响应，则可以使用 WEB CONVERSE 命令接收它。

关闭与服务器的连接后，无法使用 WEB CONVERSE 命令。如果您需要测试服务器是否已请求终止连接，使用 WEB READ HTTPHEADER 命令在服务器发出的最后一条消息中查找 Connection: close 头。

WEB CONVERSE 命令执行单个发送操作和单个接收操作，它被设计为用于代替 WEB SEND 命令和 WEB RECEIVE 命令。对于同一个连接（即，使用同一个 SESSTOKEN），您可以使用 WEB SEND 和 WEB RECEIVE 命令以及 WEB CONVERSE 命令。然而，如果您要通过管道发送请求（即，发送一系列请求而不等待响应），则不能在 WEB SEND 命令后面跟上 WEB CONVERSE 命令。CICS 在程序运行时检查在发出任何 WEB CONVERSE 命令之前，每个 WEB SEND 命令后面是否有 WEB RECEIVE 命令。例如，如果您使用三次 WEB SEND 命令发出管道传送的请求序列，则在可以使用 WEB CONVERSE 命令之前，必须先使用 WEB RECEIVE 命令三次以接收那些请求的响应。

用于发送 HTTP 客户机请求的选项

ACTION(*cvda*)

此选项用于指定应该如何发送消息。作为 HTTP 客户机的 CICS 适用的 CVDA 值有：

EXPECT

使得 CICS 将 Expect 头与请求行和请求头一起发送，并在将消息体发送到服务器之前等待 100-Continue 响应。如果接收到的不是 100-Continue 响应，则 CICS 会通知应用程序并且取消发送。如果等待一段时间之后未接收到任何响应，则 CICS 无论如何都会发送消息体。

仅当您的请求有消息体时才能使用此选项。

CLOSESTATUS(*cvda*)

指定请求中是否应该包含带“close”连接选项的 Connection 头（Connection: close）。缺省情况下消息中不包括该头。CVDA 值为：

CLOSE

使 CICS 为此请求写 Connection: close header。该头通知服务器应该在服务器将它的响应发送给请求之后关闭连接。（对于 HTTP/1.0 级别的服务器，CICS 通过省略 Connection: Keep-Alive 头达到相同的效果。）

NOCLOSE

意味着 Connection: close header 不用于此请求。如果服务器识别为 HTTP/1.0，则 CICS 会发送带“Keep-Alive”连接选项的 Connection 头（Connection: Keep-Alive）来通知期待持续连接。

DOCTOKEN(*data-value*)

指定要作为消息体发送的文档的 16 字节二进制标记。根据 *CICS Application Programming Guide* 中的描述，必须使用 CICS Document 接口（EXEC CICS DOCUMENT CREATE、INSERT 和 SET 命令）创建文档。FROM 选项提供另一个方法来创建消息体。

CICS 文档无法转换成 UTF-8 和 UTF-16 字符编码。

FROM(*data-area*)

指定保存消息体的数据缓冲区。消息体是由应用程序来构建的。当您指定 FROM 选项时，使用 FROMLENGTH 选项来指定数据缓冲区的长度。DOCTOKEN 选项提供另一个方法来创建消息体。

FROMLENGTH(*data-area*)

为 FROM 选项上提供的数据缓冲区（消息体）指定长度（以全字二进制值表示）。正确地表示此值很重要，这是因为不正确的数据长度可能会导致接收消息产生问题。

MEDIATYPE(*data-area*)

指定任何请求主体的数据内容，例如 text/xml。介质类型最多可以有 56 个字母数字字符，包括适当的标点符号。CICS 检查介质类型的格式是否正确，但不针对数据内容检查介质类型是否有效。

对于需要主体的请求，您必须指定 MEDIATYPE 选项，并且它没有缺省值。

MEDIATYPE 选项用于 WEB CONVERSE 命令的发送和接收功能。如果为它指定值，则使用该值构造请求中的 Content-Type 头，并且使用相同的字段接收服务器返回的响应的介质类型。如果不指定值，则只用它接收响应的介质类型。

METHOD(*cvda*)

为请求指定 HTTP 方法。

此命令支持 GET、HEAD、POST、PUT、TRACE、OPTIONS 和 DELETE 方法。然而，某些 HTTP 服务器（特别是 HTTP/1.0 服务器）可能不能实施所有这些方法。

CICS 禁止在不适用的情况下发送方法的消息体，并在适用的情况下需要为方法发送消息体。CVDA 值为：

GET 从服务器获取资源。不允许请求主体。

HEAD 获取 HTTP 头，但是不获取资源的响应主体。不允许请求主体。

POST 将数据发送到服务器。需要请求主体。

PUT 创建或修改服务器上的资源。需要请求主体。

TRACE

跟踪您发送到服务器的请求的路由。不允许请求主体。

OPTIONS

获取有关服务器的信息。允许请求主体，但是主体没有定义的用途。如果您使用请求主体，则必须指定介质类型。

DELETE

删除服务器上的资源。不允许请求主体。

PATH(*data-area*)

为应用程序需要访问的服务器中的特定资源指定路径信息。

如果 URIMAP 选项用于为此连接在 WEB OPEN 命令上指定现有 URIMAP 定义，则 该 URIMAP 定义中所指定的路径是 WEB SEND 命令的缺省路径。在这些情况下，如果您未在 WEB SEND 命令上指定路径信息，则使用来自 URIMAP 定义的路径。如果您指定的路径与 URIMAP 定义中给定的路径不同，则此路径覆盖来自 URIMAP 定义的路径。

如果未在 WEB OPEN 命令上使用 URIMAP 选项，则没有缺省路径，那么您必须提供路径信息。使用 WEB PARSE URL 命令可以从已知 URL 抽取路径信息。

除了使用 PATH 选项来提供路径信息之外，您还可以在 WEB SEND 命令上使用 URIMAP 选项来指定 URIMAP 定义，直接从该定义中获取路径信息。

PATHLENGTH(*data-value*)

以全字二进制值指定路径的长度。如果您要通过 PATH 选项来提供路径信息，则需要指定 PATHLENGTH 选项。如果您使用 WEB PARSE URL 命令来对 URL 进行语法分析，则会返回路径长度信息。

QUERYSTRING(*data-area*)

指定要作为请求的一部分提供给服务器的查询字符串。您不需要在查询字符串的开始处包含问号 (?)；如果您没有包含问号，则在构造请求时 CICS 会自动为您补上。如果查询字符串中包含转义字符，则 CICS 以转义字符的转义格式将它们传递给服务器。

QUERYSTRLEN(*data-value*)

以全字二进制值指定 QUERYSTRING 选项上所提供的查询字符串的长度。

SESSTOKEN(*data-value*)

指定 8 个字节的二进制值会话标记，它唯一地识别 CICS 和服务器之间的这个连接。WEB OPEN 命令为作为 HTTP 客户机的 CICS 返回此值。第 62 页的『会话标记』说明会话标记的用途。

URIMAP(*data-value*)

指定 URIMAP 定义的名称（最多 8 个字符），该定义提供应用程序需要访问的服务器中特定资源的路径信息。URIMAP 定义必须用于作为 HTTP 客户机的 CICS（指定 USAGE(CLIENT)）。它的 HOST 属性必须与为此连接在 WEB OPEN 命令上指定的 URIMAP 定义的 HOST 属性相同，或必须与为此连接在 WEB OPEN 命令上的 HOST 选项中指定的主机名相同。在 WEB CONVERSE 命令上指定的 URIMAP 定义仅应用于此请求。

如果指定了 URIMAP 选项，则不要指定 PATH 或 PATHLENGTH 选项。

用于接收服务器响应的选项

INTO(*data-area*)

指定要包含正在接收的数据的缓冲区。

MAXLENGTH(*data-value*)

以全字二进制值指定 CICS 要传递给应用程序的最大数据量。无论是否指定 INTO 或 SET 选项，都会应用 MAXLENGTH 选项来接收数据。如果已使用分块的传输编码发送了数据，则 CICS 将这些块汇编成单个消息，然后再将该消息传递给应用程序，因此 MAXLENGTH 选项应用于分块消息的总长度，而不是每个独立的块。发生任何代码页转换后测量数据。

MEDIATYPE(*data-area*)

指定 56 个字符的 *data-area* 来接收主体的介质类型（即，数据内容的类型），例如 text/xml。

MEDIATYPE 选项用于 WEB CONVERSE 命令的发送和接收功能。如果为它指定值，则使用该值构造请求中的 Content-Type 头，并且使用相同的字段接收服务器返回的响应的介质类型。如果不指定值，则只用它接收响应的介质类型。

NOTRUNCATE

指定当可用的数据长度超出 `MAXLENGTH` 选项上请求的长度时，不是立即废弃剩余的数据，而是保留数据，供后继 `RECEIVE` 命令来获取。（如果没有再发出 `RECEIVE` 命令，则在事务终止期间会废弃数据。）

带 `SET` 选项但不带 `MAXLENGTH` 选项的单条 `RECEIVE` 命令接收所有剩余的数据，而不管它的长度有多长。或者，您也可以使用一连串 `RECEIVE` 命令（带 `NOTRUNCATE` 选项）接收相应块中的剩余数据。继续发出 `RECEIVE` 命令直到您不再获得 `LENGERR` 响应。

SET(ptr-ref)

指定要设置为接收到的数据地址的指针引用。在发出下一个接收命令或任务结束之前，该指针引用一直有效。

STATUSCODE(data-area)

指定 `data-area` 以接收服务器发送的 HTTP 状态码。该代码是二进制半字值。示例有 200（正常）或 404（未找到）。接收状态码是可选的，但是在下列情况下您应该总是接收和检查状态码：

- 如果您打算现在或者在将来连接时向服务器发送相同的请求。
- 如果您打算使用此连接向服务器发送更多请求。
- 如果您的应用程序要根据响应中接收到的信息执行任何进一步处理。

STATUSTEXT(data-area)

指定 `data-area` 以接收服务器返回的用来描述状态码的任何文本。文本称为原因短语。示例有“OK”（随附 200 状态码）或者“Bad Request”（随附 400 状态码）。`STATUSLEN` 选项提供文本允许的长度。

STATUSLEN(data-value)

以全字二进制值指定 `data-area` 的长度，来接收服务器返回的描述状态码（`STATUSTEXT` 选项）的任何文本。文本称为原因短语。建议为 HTTP 使用的大多数原因短语都是简短的，但是这里建议使用 256 个字符的 `data-area` 长度，以防服务器用更详细的信息替换建议的原因短语。

TOLENGTH(data-area)

指定全字二进制变量，它设置为 `CICS` 已返回给应用程序的数据量。

- 如果不指定 `NOTRUNCATE` 选项，则现在已废弃消息中出现的更多任何的数据。如果出现更多数据，则返回 `RESP2` 值为 57 的 `LENGERR` 响应。
- 如果指定 `NOTRUNCATE` 选项，则保留其他的任何数据。如果其他数据可用，则返回 `RESP2` 值为 36 的 `LENGERR` 响应。`NOTRUNCATE` 选项的描述将告诉您这种情况下要采取的操作。

此选项等同于 `WEB RECEIVE` 命令上的 `LENGTH` 选项。

转换发送和接收项的选项

CHARACTERSET(data-value)

指定 `CICS` 在发送 HTTP 请求的实体主体之前将它转换成的字符集。字符集的名称最多可以由 40 个字母数字字符组成，包括适当的标点符号。`CICS` 不支持 IANA 命名的所有字符集。

要执行请求主体转换，必须允许 CLIENTCONV 选项缺省为 CLICONVERT 或指定为 NOINCONVERT。指定 NOCLICONVERT 或 NOOUTCONVERT 会抑制请求主体的转换。如果请求转换，并且如果未指定 CHARACTERSET 属性，则使用 ISO-8859-1 作为缺省值。

CLIENTCONV(*cvda*)

指定 CICS 在发送前是否转换 HTTP 请求的实体主体，并转换服务器响应的实体主体。缺省值为发送请求时和接收响应时都转换实体主体（CLICONVERT）。

- 对于请求主体，您可在此命令上使用 CHARACTERSET 选项，以指定适用于此服务器的字符集。如果请求转换（或作为缺省发生），但没有指定代码页，则缺省情况下 CICS 将实体主体转换为 ISO-8859-1 字符集。
- 对于响应主体，您无须指定服务器使用的字符集。CICS 通过查看消息的 Content-Type 头来识别。如果头不提供此信息，或者如果 CICS 不支持指定的字符集进行代码页转换，则使用字符集 ISO-8859-1。
- 对于应用程序的代码页，使用本地 CICS 区域的缺省代码页（由 LOCALCCSID 系统初始化参数指定），还可以使用 WEB OPEN 命令中指定的备选 EBCDIC 代码页。

CVDA 值为：

CLICONVERT

CICS 将请求的实体主体转换为您为服务器标识的字符集，并将响应的实体主体转换为适合应用程序使用的代码页。

NOINCONVERT

CICS 将请求的实体主体转换为您为服务器标识的字符集。然而，CICS 不转换响应的实体主体，而是以服务器使用的字符集将该实体主体传递给应用程序。

NOOUTCONVERT

CICS 不转换请求的实体主体，而是以应用程序使用的代码页将该实体主体传递给服务器。然而，CICS 将响应的实体主体转换为适合于应用程序使用的代码页。

NOCLICONVERT

CICS 不转换请求的实体主体，而是以应用程序使用的代码页将该实体主体传递给服务器。CICS 不转换响应的实体主体，而是以服务器使用的字符集将该实体主体传递给应用程序。

对 EXEC CICS WEB 命令选项的更改

SESSTOKEN 选项添加到 HTTP 头的 EXEC CICS WEB 命令中，以使这些命令可用于作为 HTTP 客户机的 CICS。

SESSTOKEN 选项添加到下列命令：

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

SESSTOKEN(*data-value*)

对于作为 HTTP 客户机的 CICS，此选项是必需的。它指定 8 个字节的二进制值会话标记，它唯一地标识 CICS 和服务器之间的连接。此值由作为 HTTP 客户机的 CICS 的 WEB OPEN 命令返回。第 62 页的『会话标记』说明会话标记的用途。

对 JCICS API 的更改

注：下列功能将在PTF XXXXX中可用。

扩展 JCICS 应用程序编程接口以支持下列 CICS 命令的客户机功能：

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

对全局用户出口的更改

存在两个全局用户出口，用于作为 HTTP 客户机的 CICS：WEB OPEN 命令中的 XWBOPEN，以及 WEB SEND 命令中的 XWBSNDO。（注意，仅当 WEB SEND 命令用于作为 HTTP 客户机的 CICS，而且不用于作为 HTTP 服务器的 CICS 时，XWBSNDO 才可用。）

HTTP 客户机开放出口 XWBOPEN

XWBOPEN 使系统管理员能够指定要用作作为 HTTP 客户机的 CICS 发出的 HTTP 请求的代理服务器，并且对那些请求指定的主机名应用安全策略。处理 EXEC CICS WEB OPEN 命令期间会调用 XWBOPEN，应用程序用该命令打开与服务器的连接。

CICS 本身没有任何有关为作为 HTTP 客户机的 CICS 发出的 HTTP 请求使用（或不使用）代理服务器的要求，并且 CICS 也不会为那些请求应用任何安全策略。如果您的系统或组织需要它们，则您要负责设置这些设施。

EXEC CICS WEB OPEN 命令指示 CICS Web 域打开与服务器的连接。在打开连接之前调用 XWBOPEN。在 EXEC CICS WEB OPEN 命令的 HOST 选项上指定的连接的主机名（例如，www.example.com），作为 UEPHOST 参数传递给用户出口程序以供检查。此时，可以使用用户出口程序实现以下两种目的：

- **确定 HTTP 请求是否需要使用代理服务器，并返回必需的任何代理服务器的名称。**如果需要代理服务器，则使用返回码 UERCprox，并将代理服务器的名称返回到 CICS Web 域（在 UEPPROXY 标识的缓冲区中），并将该名称用于建立与服务器的连接。如果不需要代理服务器，则使用返回码 UERCnorm。

- **要将安全策略应用于主机名。** 返回码 UERCBARR 表明不允许访问主机。如果不允许访问主机，则将 INVREQ 响应返回给 WEB OPEN 命令，并且应用程序员应该放弃打开该连接的尝试。如果您要为个别资源以及（或）主机应用安全策略，则可以在 EXEC CICS WEB SEND 和 EXEC CICS WEB CONVERSE 命令上使用 XWBSNDO 用户出口将安全策略应用到 URL 路径部分。

样本程序 DFH\$WBPI 和 DFH\$WBEX 以及关联的副本 DFH\$WBGA 向您显示如何在全局工作区中设置代理服务器信息或安全策略。例如，如果从您的 CICS 系统发出的所有请求都应该使用单个代理服务器，则可以将代理服务器名称指定为初始化参数。如果您使用很多代理服务器或者要为不同的主机名应用安全策略，则可以装入或者构建一张将主机名与相应的代理服务器匹配起来，或者将它们标记为禁止的表，然后在处理 EXEC CICS WEB OPEN 命令期间该表则可用作查找表。样本程序可以在程序列表传递初始化（PLTPI）处理期间运行，也可以在您需求使用 EXEC CICS WEB OPEN 命令之前的任何时候运行。它们都基于样本程序 DFH\$PCPI 和 DFH\$PCEX 以及样本副本 DFH\$PCGA。

出口 XWBOPEN

当调用时

处理 EXEC CICS WEB OPEN 命令期间。

特定于出口的参数

UEPHOST

字段的地址，该字段包含 WEB OPEN 命令的 HOST 选项中指定的主机名。

注：在此字段中保存主机名时，会将其转换成大写字母。在匹配主机名时，您的用户出口程序应该考虑到这一点。

UEPHOSTL

包含主机名的半字长度的字段地址。

UEPPROXY

字段的地址，该字段包含指向代理服务器名称的地址。

UEPPROXYL

包含代理服务器名称的半字长度的字段地址。

返回码

UERCNORM

此 HTTP 请求不需要代理服务器，并且不禁止主机名。

UERCPROX

此 HTTP 请求需要代理服务器。UEPPROXY 已设置为需要的代理服务器的名称，且 UEPPROXYL 已设置为代理服务器名称的长度。

UERCBARR

服务器的主机名被禁止。

UERCERR

出口处理时发生错误。

HTTP 客户机发送出口 XWBSNDO

XWBSNDO 使得系统管理员能够为作为 HTTP 客户机的 CICS 发出的 HTTP 请求指定安全策略。处理 EXEC CICS WEB SEND 或 EXEC CICS WEB CONVERSE 命令期间调用 XWBSNDO。主机名和路径信息都传递到出口，并且可以为这两个组件中的任何一个或者全部应用安全策略。

CICS 自己不会为作为 HTTP 客户机的 CICS 发出的 HTTP 请求应用任何安全策略；因此如果您的系统或组织需要它，您就要负责设置此设施。

WEB OPEN 命令上的 XWBOPEN 出口可用于禁止访问整个主机，而 XWBSNDO 出口可用于完成相同的操作或禁止访问主机内的特定路径。如果您要禁止访问整个主机，则使用 XWBOPEN 出口完成该操作可节约时间，这是因为应用程序不能打开连接，因此就不会浪费时间来创建应该发送的请求。为 XWBSNDO 出口提供主机名，这主要是为了使您能区分不同主机使用的相同路径。

如果要为 HTTP 请求使用分块的传输编码，则仅在第一个 WEB SEND 命令上为分块消息调用 XWBSNDO。

主机作为 UEPHOST 参数传递到用户出口程序，而路径作为 UEPPATH 参数传递。返回码 UERCNORM 表明路径是允许的，而返回码 UERCBARR 表明该路径是不允许的。如果路径是不允许的，则将 INVREQ 响应返回到 WEB SEND 或 WEB CONVERSE 命令，并且应用程序应该通过使用 WEB CLOSE 命令关闭连接来处理此情况。

出口 XWBSNDO

当调用时

为作为 HTTP 客户机的 CICS 发出的 HTTP 请求处理 EXEC CICS WEB SEND 或 EXEC CICS WEB CONVERSE 命令期间。在 WEB SEND 命令上使用 SESSTOKEN 参数来表明客户机请求。

特定于出口的参数

UEPHOST

字段的地址，该字段包含 WEB OPEN 命令的 HOST 选项中为连接指定的主机名。

注：在此字段中保存主机名时，会将它转换成大写字母。在匹配主机名时，您的用户出口程序应该考虑到这一点。

UEPHOSTL

包含主机名的半字长度的字段地址。

UEPPATH

字段的地址，该字段包含 WEB SEND 命令的 PATH 选项中指定的路径。

UEPPATHL

包含路径的半字长度的字段地址。

返回码

UERCNORM

允许路径。

UERCBARR

不允许路径。

对监控的更改

添加了下列字段到性能类监控记录的 DFHWEBB 组:

331 (TYPE-A, 'WBREDOCT', 4 BYTES)

当 CICS 作为 HTTP 客户机时, 用户任务发出的 CICS Web 支持 READ HTTPHEADER 请求数。

332 (TYPE-A, 'WBWRTOCT', 4 BYTES)

当 CICS 作为 HTTP 客户机时, 用户任务发出的 CICS Web 支持 WRITE HTTPHEADER 请求数。

333 (TYPE-A, 'WBRCVIN1', 4 BYTES)

当 CICS 作为 HTTP 客户机时, 用户任务发出的 CICS Web 支持 RECEIVE 和 CONVERSE 请求数。

334 (TYPE-A, 'WBCHRIN1', 4 BYTES)

当 CICS 作为 HTTP 客户机时, 用户任务发出的 CICS Web 支持 RECEIVE 和 CONVERSE 请求数。

335 (TYPE-A, 'WBSNDOU1, 4 BYTES)

当 CICS 作为 HTTP 客户机时, 用户任务发出的 CICS Web 支持 SEND 和 CONVERSE 请求数。

336 (TYPE-A, 'WBCHROU1', 4 BYTES)

当 CICS 作为 HTTP 客户机时, 用户任务发出的 CICS Web 支持 SEND 和 CONVERSE 请求发送的字节数。

337 (TYPE-A, 'WBPARSCT', 4 BYTES)

用户任务发出的 CICS Web 支持 PARSE URL 请求数。

338 (TYPE-A, 'WBBRWOC', 4 BYTES)

当 CICS 作为 HTTP 客户机时, 用户任务发出的 CICS Web 支持 BROWSE HTTPHEADER 请求 (STARTBROWSE, READNEXT 和 ENDBROWSE) 数。

注: 使用 WEB CONVERSE 命令发出请求时, 它既增加 Send 和Receive 请求计数 (WBSNDOU1 和 WBRCVIN1), 也增加发送和接收的字符计数 (WBCHRIN1 和 WBCHROU1)。

第 4 章 升级到 HTTP/1.1 的 CICS Web 支持

CICS Web 支持现在支持 HTTP/1.1。

CICS Transaction Server for z/OS V3R1 之前的 CICS 发行版支持 HTTP/1.0。现在增强 CICS Web 支持以处理和提供 HTTP/1.1 规范的功能，包括分块传送编码、管道传送和持续连接。

正如因特网协会和 IETF（因特网工程任务组织）请求评论文档 RFC 2616 *Hypertext Transfer Protocol - HTTP/1.1* 中所描述，CICS Web 支持有条件地遵从 HTTP/1.1 规范。

（RFC 2616 可从 <http://www.ietf.org/rfc/rfc2616.txt> 下载获得。）有条件地遵从 HTTP/1.1 规范意味着 CICS 满足所有“MUST”级别需求，但并不满足所有的 SHOULD 级别需求（这在 HTTP/1.1 规范中有详细的描述），在那些地方这些需求与 CICS 自身实际提供的功能相等。您的用户应用程序也有责任来承担维护它们所执行的操作中的一致性，在您准备好要维持应用程序的一致性时向您提供指导，以帮助您实施它。

CICS TS V3 的新行为

- CICS 检查入站消息是否与 HTTP/1.1 一致，并处理或拒绝不一致的消息。
- CICS 遵循 HTTP/1.1 规则，以比较 URL。
- CICS 在出站消息的开始行中提供合适的 HTTP 版本号。
- 在出站 HTTP/1.1 消息上，CICS 提供应该为遵从 HTTP/1.1 的消息正常显示的 HTTP 头：
 - Content-Length（用于客户机请求或服务器响应）。
 - Content-Type（用于客户机请求或服务器响应）。
 - Date（用于客户机请求或服务器响应）。
 - Host（用于客户机请求）。
 - Last-Modified（仅用于带有 HFS 文件的静态服务器响应）。
 - Server（用于服务器响应）。
 - TE（用于客户机请求）。
 - Transfer-Encoding（用于客户机请求或服务器响应）。
 - User-Agent（用于客户机请求）。
- CICS 为入站和出站请求在 Expect 头上采取行动。
- CICS 处理 Web 客户机的 OPTIONS 请求，并做出相应的响应。
- CICS 处理 Web 客户机的 TRACE 请求，并做出相应的响应。
- CICS 接受带有分块传送编码的入站消息并为您组装它们，并且支持您使用分块传送编码发送出站消息。
- CICS 支持对入站和出站消息进行管道传送。
- CICS 支持虚拟主管（同一 IP 地址上多个主机名），这基于您的 URIMAP 定义。

已更改的行为，与 CICS TS V2 进行比较

- 缺省情况下，连接现在是持久的。
- CICS 处理大范围的错误情况和不支持的消息。

CICS Web 支持不支持 HTTP 功能

HTTP/1.1 规范 (RFC 2616) 为使用 HTTP 协议的各方定义各种角色。CICS Web 支持提供适用于原点服务器、客户机和用户代理 (尽管用户可能不参与每个 HTTP 客户机请求) 的 HTTP 服务。

HTTP/1.1 规范还包括与 CICS Web 支持不相关的角色相关的需求, 可忽略这些需求:

- CICS不担当代理。
- CICS不担当网关 (另一个服务器的调解人) 或隧道 (HTTP 连接之间的中继设备)。
- CICS不提供高速缓存工具, 或者提供对用户所编写高速缓存工具的支持。
- CICS不设计为用作 Web 浏览器。

CICS Web 支持升级到 HTTP/1.1 的优点

升级到 HTTP/1.1 的 CICS Web 支持提供下列优点:

- 有条件的遵从 HTTP/1.1 规范 (RFC 2616) 意味着 CICS Transaction Server for z/OS V3R1 可以与因特网上的大范围的服务和客户机容易并正确地进行交互。
- 持续连接现在是 CICS 和远程服务器或 Web 客户机之间连接的缺省值。持续连接改进了网络性能, 这是因为不需要为每个请求建立新的连接。
- 支持管道传送意味着 CICS 应用程序可以发送一系列多个请求到服务器, 而无需在每个请求之后等待确认。CICS Web 支持还确保正确处理从 Web 客户机到 CICS 的管道请求。
- 分块传送编码 (简称为分块) 在仍允许接收方验证是否接收到完整消息的情况下, 允许在一些便利的段中传送动态产生的内容或大量内容。CICS 可以接收用这种方式发送的项, 并且用户编写的应用程序可以用这种方式发送项。
- CICS 使用您的 URIMAP 定义自动创建虚拟主机, 允许您在同一个 IP 地址上提供多个主机名。
- CICS支持 Web 客户机和 CICS 应用程序使用 HTTP/1.1 规范所定义的所有 HTTP 方法发出的请求, 而且还为使用 OPTIONS 和 TRACE 方法的 Web 客户机请求提供自动处理和响应。

需求

要支持此功能, 没有特别的硬件或软件需求。

相关信息

第 311 页的第 27 章, 『CICS操作环境』

新的 HTTP 功能

CICS Web 支持包括新的功能区域。

- 分块传送编码, 或正在分块
- 管道传送
- 持续连接
- 虚拟主管

分块的传输编码

分块的传输编码（也称为分块）包括将消息的主体作为一系列块传送，每个块都带有它自己的块大小头。消息的结束由长度为零的块和一个空行来表明。

此定义的过程意味着应用程序生成的实体主体或大的实体主体可以分成一些方便的段进行发送。当接收到零长度的块时，客户机或服务器知道分块消息是完整的了。

分块消息的主体后面可以跟上可选的包含增补的 HTTP 头（称为尾部头）的尾部。客户机和服务器不需要接受尾部，因此增补的 HTTP 头应该只提供非本质的信息，除非服务器知道客户机接受尾部。

要使用分块的传输编码，客户机和服务器都必须是 HTTP/1.1 版本。分块消息不能发送到 HTTP/1.0 客户机。HTTP/1.1 规范（RFC 2616）中定义了应用分块的传输编码的需求以及尾部的使用。可以从 <http://www.ietf.org/rfc/rfc2616.txt> 获得 RFC 2616。

CICS Web 支持如何处理分块的传输编码

CICS 可以发送和接收使用分块的传输编码的消息。作为 HTTP 服务器的 CICS 可以将分块消息作为请求接收，也可以作为响应发送它。作为 HTTP 客户机的 CICS 可以将分块消息作为请求发送，也可以作为响应接收它。

下面是 CICS Web 支持对这些不同情况的处理：

- 当作为 HTTP 服务器的 CICS 接收作为 HTTP 请求的分块消息时，CICS Web 支持识别分块编码。它一直等待，直到接收到所有的块（通过接收长度为零的块来表明），并汇编这些块来构成完整的消息。用户应用程序可以使用 WEB RECEIVE 命令接收汇编的消息体。
 - 在与请求到达的端口相关的 TCPIP SERVICE 资源定义上使用 MAXDATALEN 选项，您可以限制 CICS 接受单个分块消息的总数据量。
 - 当 CICS 是 HTTP 服务器时，接收分块消息的超时值由 TCPIP SERVICE 定义的 SOCKETCLOSE 属性设置。
 - 可以使用 HTTP 头命令读来自分块消息的尾部头。尾部头识别代表尾部头的头的名称。如果您在处理请求的路径中使用分析器程序，请注意尾部头不随主请求头传递到分析器程序。
- 作为 HTTP 客户机的 CICS 接收到作为对应用程序请求的响应的分块消息时，这些块在作为实体主体传送到应用程序之前也要进行汇编，并且可以使用 HTTP 头命令读所有尾部头。您可以使用与应用程序相关的事务标识的事务概要定义的 RTIMOUT 属性，指定应用程序将等待接收响应的的时间。
- CICS 发送分块消息时（无论 CICS 是作为 HTTP 服务器（此时分块消息是响应）还是作为 HTTP 客户机（此时分块消息是请求）），应用程序都可以通过在 WEB SEND 命令上使用 CHUNKLENGTH 选项，为每个消息块指定分块的传输编码。可以用对应用程序来说是最方便的任何方法来分割消息。例如，产生来自许多其他应用程序的输出时可以发送该输出，或者可以分别读和发送来自表的每列的数据。CICS 发送每个消息块，并且添加适当的 HTTP 头向接收者表明正在使用分块编码。应用程序发出带 CHUNKLENGTH(0) 并具有空白数据规范的 WEB SEND，以表明消息结束。然后，CICS 会发送一个空的块（包含空白行）以及需要的任何尾部（尾部头）以结束分块编码。

管道传送

管道传送涉及客户机在不等待响应的情况下将多个 HTTP 请求发送到服务器。然后，必须以接收请求时所用的相同顺序从服务器返回响应。

请求者有责任确保请求是幂等的。幂等性意味着当重复一系列请求的所有或者一部分时总是能得到相同的结果。这确保了如果在连接服务器时遇到错误，则客户机可以重试一系列请求，即使它不知道服务器是否已实施了所有、部分请求或者一个请求都没有实施也重试。

HTTP/1.1 规范 (RFC 2616) 定义有关 HTTP 请求的幂等性的规则。可以从 <http://www.ietf.org/rfc/rfc2616.txt> 获得 RFC 2616。简言之，如果在他们自己的资源上使用，则多数请求方法都是幂等的，这是因为每次使用方法时都得到相同的方法。（有一个例外是 POST 方法，因为它更改服务器上的资源。）然而，在管道传送期间发送一系列请求时，该序列可能不是幂等的，特别是在资源有所更改的情况下。

如果您打算管道传送请求，则检查请求序列是否可能会在任何地点终止，然后从起始处重新开始，而不会导致逻辑错误。如果并非如此，则分别发出请求，并且在每次请求后等待确认。

CICS Web 支持如何处理管道传送

CICS 可以发送和接收通过管道传送的请求序列。作为 HTTP 服务器的 CICS 可以接收 Web 客户机发出的通过管道传送的请求序列，而作为 HTTP 客户机的 CICS 可以发送通过管道传送的请求序列给服务器。

下面描述的是 CICS Web 支持处理通过管道传送的请求序列，以及对它们的响应：

- 当作为 HTTP 服务器的 CICS 接收管道传送的 HTTP 请求序列时，会连续处理请求。每个事务处理单个请求并提供响应。在发送对上一个请求的响应之前，CICS 会一直保留管道传送的消息序列中的剩余请求，然后启动新的事务以处理每个后继请求。这意味着每个请求都由不同的应用程序处理，或者由同一个程序的不同实例处理。
- 当作为 HTTP 客户机的 CICS 发送通过管道传送的请求序列时，自动启用管道传送。每个 HTTP 请求是立即发送的，因此应用程序可以在接收任何响应之前发送多个 HTTP 请求。当已发送了管道传送的序列中的最后消息时，应用程序可以开始接收响应。
- 当作为 HTTP 客户机的 CICS 接收到对通过管道传送的请求序列的 HTTP 响应时，以 CICS 从服务器接收响应的顺序将它们返回给应用程序。

对于作为 HTTP 客户机的 CICS，应用程序负责确保任何通过管道传送的请求序列都是幂等的。

持续连接

持续连接是 Web 客户机和服务器之间的连接，可以为多个请求和响应的交换重复使用这些连接。

在 HTTP/1.0 中，服务器的缺省操作是当它接收到 Web 客户机发出的请求并且发送响应后就关闭连接。如果 Web 客户机希望服务器让连接一直开着，则必须在请求上发送 Connection: Keep-Alive 头。

对于 HTTP/1.1，持续连接是缺省值。当 Web 客户机和服务器之间建立连接时，缺省情况下，服务器应该让连接一直开着。仅在以下情况下才应该关闭连接：如果 Web 客户机通过发送 Connection: close 头请求关闭，或者如果达到服务器的超时设置，或者如果服务器遇到错误。

持续连接改进了网络性能，这是因为不需要为每个请求建立新的连接。与使用现有的连接发出请求相比，建立新的连接会消耗可观的额外网络资源。

CICS Web 支持如何处理持续连接

持续连接是 CICS Web 支持的缺省行为。

CICS TS 3.1 之前的连接行为是，从 Web 客户机接收数据后，CICS 通常将关闭连接，除非 Web 客户机发送了 Connection: Keep-Alive 头。

现在，当在 Web 客户机和作为 HTTP 服务器的 CICS 之间建立连接，或者在作为 HTTP 客户机的 CICS 和服务器之间建立连接时，缺省情况下，CICS 会尝试让连接开着。

当 CICS 是 HTTP 服务器时，在下列情况下关闭持续连接：

- 处理请求的用户编写的应用程序关闭连接（通过在 WEB SEND 命令上指定 CLOSESTATUS(CLOSE) 选项）。
- Web 客户机关闭连接（由 Connection: close 头通知）。
- Web 客户机是不发送 Connection: Keep-Alive 头的 HTTP/1.0 客户机。
- 达到超时时期（表明连接已失败，或者 Web 客户机故意退出了连接）。

否则，CICS 让持续连接开着，以便 Web 客户机能够发送后继请求。

使用 CICS Web 支持的 TCPIPSERVICE 资源定义时，TCPIPSERVICE 定义的 SOCKETCLOSE 属性不应该指定为零。SOCKETCLOSE 的设置为零意味着作为 HTTP 服务器的 CICS 在从 Web 客户机接收到数据之后会立即关闭连接，除非有更多数据在等待接收。这意味着无法维持持续连接。

当 CICS 是 HTTP 客户机时，在下列情况下关闭持续连接：

- 服务器关闭连接（由 HTTP/1.1 服务器发送 Connection: close 头，或者 HTTP/1.0 服务器无法发送 Connection: Keep-Alive 头通知）。
- 用户应用程序关闭连接（通过在 WEB SEND 或 WEB CONVERSE 命令上指定 CLOSESTATUS(CLOSE) 选项，或者通过发出 WEB CLOSE 命令）。
- 到达任务的结尾而连接还未关闭。

虚拟主管

HTTP 包含虚拟主管的概念，虚拟主管中单个 HTTP 服务器可以表示同一 IP 地址上的多个主机。

DNS 服务器可以将几个不同的主机名分配给同一个 IP 地址。当 HTTP 客户机对特定主机发出请求时，它使用 DNS 服务器查找与该主机名相对应的 IP 地址，并向该 IP 地址发送请求。

在 HTTP/1.0 中，HTTP 消息中不出现主机名；它在解析 IP 地址之后丢失。这意味着如果服务器上有由同一个 IP 地址代表的多个资源集，则服务器很难辨别资源所属的主机。

然而，HTTP/1.1 请求在请求中提供主机名（通常在 Host 头中提供）。消息中存在主机名使得 HTTP 服务器能够将包含不同主机名的请求导向每个主机的相应资源。此 HTTP 功能称为虚拟主管。

管理虚拟主管

CICS 通过 URIMAP 资源定义对象支持虚拟主管。

您为作为 HTTP 服务器的 CICS（在 URIMAP 定义中使用 USAGE(SERVER)）设置的每个 URIMAP 定义包括需要 Web 客户机在它的请求中提供的主机名。通过把所有 URIMAP 定义一起分组为单个数据结构（指定相同主机名和相同 TCPIP SERVICE 定义的 CICS 区域中的所有 URIMAP 定义），CICS 自动为您创建虚拟主机。不指定 TCPIP SERVICE 定义的 URIMAP 定义（因此 URIMAP 定义适用所有 TCPIP SERVICE 定义）被添加到指定匹配主机名的所有数据结构中，因此这些 URIMAP 定义可以是多个数据结构的一部分。然后，这些 URIMAP 定义组中的每一个构成虚拟主机，可将该虚拟主机作为单个单元来管理。

您可以使用以下 CICS 命令管理 CICS 从您的 URIMAP 定义创建的虚拟主机：

- **INQUIRE HOST** 命令用于查询虚拟主机的状态。该命令告诉您虚拟主机的主机名，与它关联的 TCPIP SERVICE 定义（或者它是否与 CICS 区域中的每个 TCPIP SERVICE 定义关联）以及它是启用的还是禁用的。
- **SET HOST** 命令用于将虚拟主机的状态设置为启用或禁用。禁用虚拟主机意味着应用程序无法访问组成虚拟主机的所有 URIMAP 定义。禁用虚拟主机时，CICS 将 HTTP 503 响应（服务不可用）返回给 Web 客户机。
- 虚拟主机浏览命令用于在 CICS 系统中浏览虚拟主机。

统计程序 DFHOSTAT 包含显示 CICS 已创建的虚拟主机的报告。

如果删除了组成虚拟主机的所有 URIMAP 定义，则 CICS 自动删除虚拟主机。如果您不希望使用 CICS 为您创建的虚拟主机，则可以忽略它们，并在您的 URIMAP 定义的级别进行管理。

您还可以使用分析器程序管理虚拟主机。将 HTTP 请求的主机名传递给分析器程序，并且您可以对程序进行编码，以向请求提供与主机相关的响应。但是，无法使用 INQUIRE HOST、SET HOST 和虚拟主机浏览命令来管理以此方法设置的虚拟主机。

对 CICS 外部的更改

对资源定义的更改

对 TCPIP SERVICE 资源定义的更改

TCPIP SERVICE 具有新属性：

MAXDATALEN(32|number)

定义 CICS 在 HTTP 协议或 USER 协议上作为 HTTP 服务器所接收数据的最大长度。缺省值为 32K。最小值为 3K，而最大值为 524288K。要增加 CICS Web 支

持的安全性，在每个 TCPIP SERVICE 定义上为 HTTP 协议指定此选项。它可帮助您
您在涉及大量数据传输时防止 CICS 受到拒绝服务的攻击。

PROTOCOL 属性可以使用新的 USER 选项。现在必须在 USER 协议下执行所有非
HTTP 请求的处理。对 USER 协议上接收的消息不执行语法分析，而且已为跨网络传输
分割的请求不会自动组装。这是与在早期 CICS 发行版中处理非 HTTP 消息相同的行
为。

PROTOCOL(*ECI|HTTP|IOP|USER*)

指定 TCP/IP 端口使用的应用程序级别协议。

ECI 使用 CICS ECI 协议。

HTTP 使用 HTTP 协议。HTTP 协议由 CICS Web 支持处理。CICS 为使用此协
议发送和接收的消息执行基本验收检查。此协议对于熟知端口 80（用于没
有 SSL 的 HTTP）和 443（用于带有 SSL 的 HTTP）是必需的。

IOP 使用 IOP 协议。为接受企业 bean 的入站请求的 TCPIP SERVICE 指定
IOP。

USER 使用用户定义的协议。把消息当作非 HTTP 消息处理。将它们标志为非
HTTP 并在不更改的情况下传递给 TCPIP SERVICE 的分析器程序。CICS
Web 支持工具用于处理请求，但是对于使用此协议发送和接收的消息，不
执行验收检查。必须在 USER 协议下执行所有非 HTTP 请求的处理，以便
它们从 CICS 使用 HTTP 协议为请求执行的基本验收检查中受到保护。如
果 HTTP 消息由 USER 协议处理，则您负责检查其有效性。

当 PROTOCOL 设置为 USER 时所使用 TCPIP SERVICE 资源定义的属性与 PROTOCOL
设置为 HTTP 时使用的那些属性相同。URIMAP 定义不与 USER 协议协同使用。

当协议定义为 USER 时，新的 CICS 所提供事务 CWXU（CICS Web 用户定义的协议
附加事务）是缺省值。CWXU 执行程序 DFHWBXN。

现在按如下所示描述 SOCKETCLOSE 属性。注意，对于 HTTP 协议，SOCKETCLOSE
不应该指定为 0。

SOCKETCLOSE(*NO|hmmss*)

指定在发出接收套接字入局数据的命令之后，在关闭此套接字之前，CICS 是否应等
待以及等多长时间。

NO 在客户机，或者 CICS 中的用户应用程序关闭套接字前，它保持打开。

hmmss

CICS 使套接字在一段时间（以 HHMMSS 为格式）后超时。选择适于客户
机响应以及您的网络可靠性的值。如果数据不可用于除第一个 RECEIVE 的
任何 RECEIVE，则指定 000000 会立即关闭套接字。

如果您要在为具有 HTTP 协议的 CICS Web 支持使用 TCPIP SERVICE，不应该指
定 SOCKETCLOSE(0)。SOCKETCLOSE 的零设置意味着 CICS 在接收到从 Web
客户机发出的数据之后立即关闭连接，除非还在等待其他数据。这意味着无法维护
持续连接。

如果您指定 PROTOCOL(ECI)，则必须指定 SOCKETCLOSE(NO)。

SOCKETCLOSE 属性不适用于连接之后发出的第一个 RECEIVE。对于 HTTP、USER 和 ECI 协议, 在进行第一个 RECEIVE 请求时, CICS 在关闭套接字之前会等待数据 30 秒。对于 IIOP 协议, CICS 会无限期地等待。

安装 TCPIP SERVICE 之后, 您不能使用 CEMT 更改该值; 您必须将 TCPIP SERVICE 设置为停用, 然后重新安装具有已修改定义的 TCPIP SERVICE。

对应用程序编程接口 (HTTP/1.1 支持) 的更改

新的以及更改过的命令

当下列 EXEC CICS WEB 命令被作为 HTTP 服务器的 CICS 使用时, 它们已增强:

- EXEC CICS WEB SEND
- EXEC CICS WEB RECEIVE

CICS Web 支持设计为允许在 CICS Transaction Server for z/OS V3R1 前使用这些命令的支持 web 的应用程序在不更改的情况下工作, 直到您选择迁移它们以利用现在可用的增强。

CICS Transaction Server for z/OS V3R1 前, 在 WEB SEND 命令上可用的选项是 CLNTCODEPAGE、DOCTOKEN、LENGTH、STATUSCODE 和 STATUSTEXT。在 CICS Transaction Server for z/OS V3R1 中:

- 选项 DOCTOKEN、STATUSCODE 和 STATUSTEXT 的名称和功能不更改。
- 选项 CLNTCODEPAGE 和 LENGTH 仅支持迁移目的, 而且它们的功能分别被新的选项 CHARACTERSET 和 STATUSLEN 所替代。
- 一些新选项可用于高级功能。

CICS Transaction Server for z/OS V3R1 前, 在 WEB RECEIVE 命令上可用的选项是 CLNTCODEPAGE、HOSTCODEPAGE、INTO、LENGTH、MAXLENGTH、NOTRUNCATE、SET 和 TYPE。在 CICS Transaction Server for z/OS V3R1 中:

- 选项 INTO、LENGTH、MAXLENGTH、NOTRUNCATE、SET 和 TYPE 的名称和功能不更改。
- 选项 HOSTCODEPAGE 可仍然使用, 但是它不再是必需的, 而且如果未指定它, CICS 可提供缺省值。
- 选项 CLNTCODEPAGE 仅支持迁移目的, 而且其功能被新选项 CHARACTERSET 所替换。
- 一些新选项可用于高级功能。

EXEC CICS WEB SEND 和 WEB RECEIVE 命令在被作为 HTTP 客户机的 CICS 使用时还具有一系列新选项, 这在第 63 页的『对应用程序编程接口 (HTTP 客户机请求) 的更改』中有所描述。

下列 EXEC CICS 命令仅对于作为 HTTP 服务器的 CICS 和作为 HTTP 客户机的 CICS 而提供或增强:

- EXEC CICS WEB PARSE URL
- EXEC CICS WEB EXTRACT
- EXEC CICS CONVERTTIME
- EXEC CICS FORMATTIME

这些命令在第 114 页的『对应用程序编程接口（一般 CICS Web 支持增强）的更改』中有所描述。

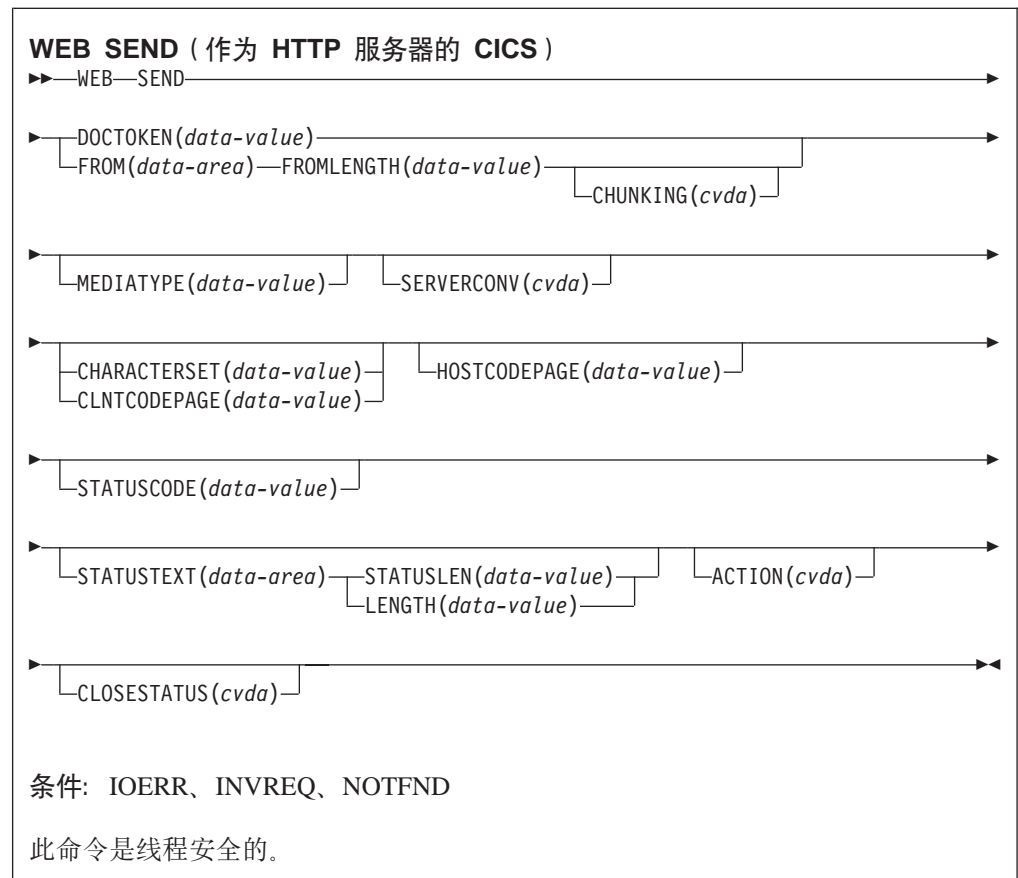
新的 MAXDATALEN 选项已添加到 EXTRACT TCPIP 命令:

MAXDATALEN(data-area)

指定全字二进制字段，以包含作为 HTTP 服务器的 CICS 可接收的最大数据长度的设置。

WEB SEND (服务器)

发送 HTTP 响应或非 HTTP 消息。



描述

对于作为 HTTP 服务器的 CICS，WEB SEND 选择要通过 CICS Web 支持或 CICS 业务逻辑接口传递的项，并指定用于发送项的选项。该项可以是:

- 对 Web 客户机向作为 HTTP 服务器的 CICS 发出的 HTTP 请求的响应。
- 非 HTTP 消息由 CICS Web 支持设施处理，它使用 TCPIP SERVICE 定义上用户定义的 (USER) 协议。
- 对来自另一个应用程序的请求的响应，而该应用程序已使用 CICS 业务逻辑直接联系程序，而不通过 CICS HTTP 侦听器。

一个任务只能发送一个响应。这可以是使用一个 WEB SEND 命令的标准响应，也可以是使用一系列 WEB SEND 命令的分块响应。

每次接收到来自 Web 客户机的请求时，CICS 启动新任务来处理请求。

选项

ACTION(*cvda*)

指定应发送出多少消息。作为 HTTP 服务器的 CICS 适用的 CVDA 值有:

IMMEDIATE

立即向 Web 客户机发送响应。如果指定 CHUNCKING，则使用 IMMEDIATE 选项。对于不使用分块传输编码发送的消息，EVENTUAL 是缺省值，它在任务结束时发送响应。

EVENTUAL

在任务结束时向 Web 客户机发送响应。如果指定 CHUNCKING，则忽略 EVENTUAL 选项。此选项产生的行为与 CICS Web 支持在 CICS Transaction Server for z/OS V3R1 之前的发行版中的行为相同，并且这是作为 HTTP 服务器的 CICS 的缺省值。

CHARACTERSET(*data-value*)

指定 CICS 发送命令发送的项的实体主体之前将它转换所至的字符集。字符集的名称最多可以由 40 个字母数字字符组成，包括适当的标点符号。CICS 不支持 IANA 命名的所有字符集。

当指定 CHARACTERSET 选项时，使用 SERVERCONV(SRVCONVERT)，因此会发生实体主体的代码页转换。作为由自己选择字符集的替换方法，指定 SERVERCONV(SRVCONVERT) 或 HOSTCODEPAGE（如果允许）并省略 CHARACTERSET，可以让 CICS 为消息体确定合适的字符集。

如果您省略所有代码页转换选项，则不会发生代码页转换。

CHUNCKING(*cvda*)

用于以分块的形式（称为分块的传输编码）发送消息时控制消息发送。未指定该选项时的缺省值为不在使用分块的传输编码。只有 HTTP/1.1 客户机才接受分块的传输编码，并且它不能在 HTTP/1.0 客户机或非 HTTP 消息中使用。

可以用对应用程序而言最便利的任何一种方法将分块消息的内容划分成块。不能从 CICS 文档直接构成分块消息的主体，因此不能使用 DOCTOKEN 选项。

对于每一块消息，使用带有 CHUNCKING(CHUNKYES) 的不同 WEB SEND 命令。使用 FROM 选项指定数据的块，并使用 FROMLENGTH 选项指定块的长度。在序列的第一个 WEB SEND 命令中为消息指定其他选项（如 CLOSESTATUS 选项），但在以后的消息中不指定它们。当已发送最后一个数据块时，再指定一个带 CHUNCKING(CHUNKEND) 但不带 FROM 或 FROMLENGTH 选项的 WEB SEND 命令。CICS 然后将空块发送给接收方以完成分块的消息。

CVDA 值为:

CHUNKNO

消息不使用分展块的传输编码。如果未指定 CHUNCKING 选项，则此为缺省值。

CHUNKYES

分块的传输编码正在进行中。FROM 选项指定的数据表示消息块。

CHUNKEND

分块的传输编码已完成。没有为该发送指定数据。CICS 向接收方发送空块以完成分块消息。

CLNTCODEPAGE(*data-value*)

仅支持此选项用于迁移目的。CHARACTERSET 可替换它。对于这两个关键字，CICS 采取的操作都是相同的。这意味着即使未指定 SERVERCONV 选项，指定 CLNTCODEPAGE 时也会发生代码页转换。如果省略了所有的代码页转换选项，则不会发生代码页转换。

CLOSESTATUS(*cvda*)

指定 CICS 是否在发送消息后关闭连接。缺省值为不关闭连接。CVDA 值为：

CLOSE

CICS 为该响应写带“close”连接选项的 Connection 头（Connection: close），并在发送响应之后关闭与 Web 客户机的连接。头会向 Web 客户机通知关闭的消息。（对于 HTTP/1.0 级别的 Web 客户机，CICS 通过省略 Connection: Keep-Alive 头达到相同的效果。）

如果在使用分块的传输编码，则可以在第一个消息块上指定 CLOSESTATUS(CLOSE) 选项，以通知 Web 客户机在分块消息完成后关闭连接。

NOCLOSE

意味着此响应没有使用 Connection: close 头，并且连接仍保持打开状态。如果 Web 客户机识别为 HTTP/1.0 并且发送了带“Keep-Alive”连接选项的 Connection 头（Connection: Keep-Alive），则 CICS 会发送相同的头来通知将维持持续连接。

DOCTOKEN(*data-value*)

指定要作为消息体发送的文档的 16 字节二进制标记。根据 *CICS Application Programming Guide* 中的描述，使用 CICS Document 接口（EXEC CICS DOCUMENT CREATE、INSERT 和 SET 命令）创建文档。FROM 选项提供另一个方法来创建消息体。

不能从 CICS 文档构成分块消息的主体，因此 DOCTOKEN 选项不能用于分块的传输编码。

CICS 文档无法转换成 UTF-8 和 UTF-16 字符编码。

FROM(*data-area*)

指定保存完整消息体或一块消息体的数据缓冲区。消息体是由应用程序来构建的。当您指定 FROM 选项时，使用 FROMLENGTH 选项来指定数据缓冲区的长度。DOCTOKEN 选项提供另一个方法来创建消息体，但该选项无法用于分块消息体。

FROMLENGTH(*data-value*)

为 FROM 选项上提供的数据缓冲区指定长度（以全字二进制值表示）。正确地表示此值很重要，这是因为不正确的数据长度可能会导致接收消息产生问题。

HOSTCODEPAGE(*data-value*)

指定应用程序为响应的实体主体使用的 CICS（主机）代码页的 8 个字符的名称。当指定 HOSTCODEPAGE 选项时，因为使用 SERVERCONV(SRVCONVERT)，所以会发生实体主体的代码页转换。指定 SERVERCONV(SRVCONVERT) 和 / 或 CHARACTERSET 并省略 HOSTCODEPAGE 让 CICS 识别代码页。

如果使用 CICS 文档来形成响应主体 (DOCTOKEN 选项), 则不要指定 HOSTCODEPAGE 选项, 这是因为 CICS 从 CICS 文档域的文档主机代码页记录中识别主机代码页。

如果使用数据缓冲区来形成响应主体 (FROM 选项), 则可能需要指定 HOSTCODEPAGE。如果未指定此选项, 则缺省值为本地 CICS 区域的缺省代码页, 如 LOCALCCSID 系统初始化参数中指定的那样。如果您需要代码页转换, 但您的应用程序使用了不同的代码页, 则使用 HOSTCODEPAGE 指定它。

LENGTH(*data-value*)

仅支持此选项用于迁移目的。STATUSLEN 可替换它。

MEDIATYPE(*data-value*)

指定消息体的数据内容, 例如 text/xml。介质类型最多可以有 56 个字母数字字符, 包括适当的标点符号。CICS 检查介质类型的格式是否正确, 但不针对数据内容检查介质类型是否有效。CICS 不提供缺省值。

SERVERCONV(*cvda*)

指定在发送前, CICS 是否将命令发送的项的实体主体, 从应用程序使用的代码页转换到适合于接收方使用的字符集。您可以在此命令中使用 CHARACTERSET 和 HOSTCODEPAGE 选项, 来指定所使用的字符集和代码页。如果您指定这些选项中的任意一个, 则采用代码页转换 (SRVCONVERT)。或者, 您可全部省略这些选项或省略任意一个选项, 指定 SERVERCONV(SRVCONVERT), 并让 CICS 确定合适的字符集和代码页。

SRVCONVERT

CICS 转换消息的实体主体。

NOSRVCONVERT

CICS 不转换 HTTP 请求的实体主体, 而是以应用程序使用的代码页将该实体主体传递给服务器。如果您指定 NOSRVCONVERT, 则不能指定 CHARACTERSET 或 HOSTCODEPAGE 选项。

注: 如果您省略所有代码页转换选项

(SERVERCONV, CLNTCODEPAGE, CHARACTERSET, HOSTCODEPAGE), 则不会发生代码页转换。

STATUSCODE(*data-value*)

指定由应用程序确定的标准 HTTP 状态码, 该状态码将插入到 HTTP 响应的状态行上。该代码是半字二进制值。示例有 200 (正常响应) 或 404 (未找到)。如果未指定此选项, 则 CICS 提供缺省值 200。

对于状态码 204、205 和 304, 不允许消息体, 如果您试图包含消息体, 则 CICS 会将错误响应返回给命令。除此之外, CICS 也不检查您的状态码使用是否正确。

STATUSLEN(*data-value*)

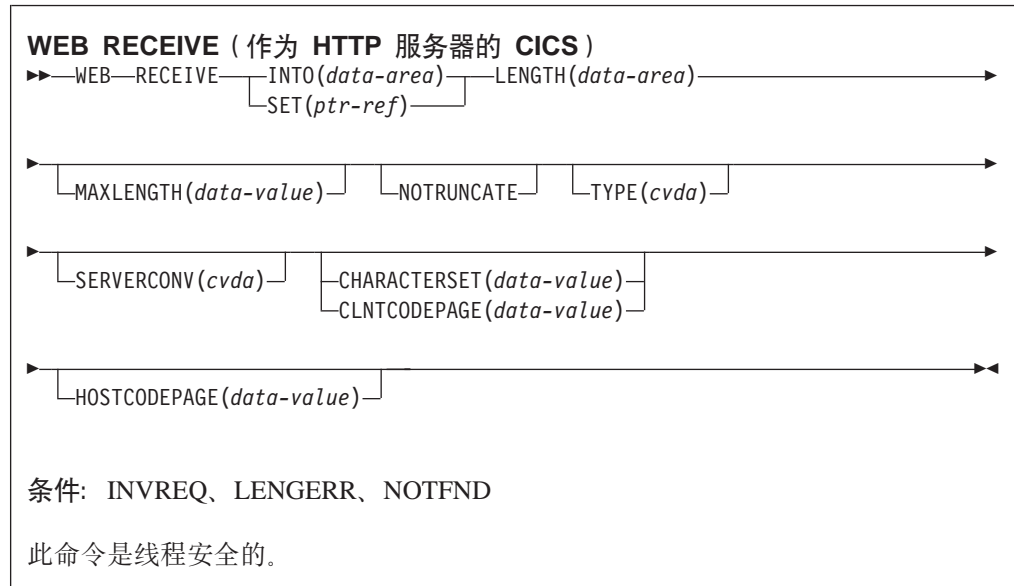
指定 STATUSTEXT 选项上提供的字符串的长度 (以全字二进制值表示)。

STATUSTEXT(*data-area*)

指定包含人们可读的文本的 *data-area*, 以描述产生状态码的原因。文本称为原因短语。示例有“OK” (随附 200 状态码) 或者“Bad Request” (随附 400 状态码)。

WEB RECEIVE (服务器)

接收 HTTP 请求或非 HTTP 消息。



描述

WEB RECEIVE 接收 HTTP 请求的主体或非 HTTP 消息的所有数据，并将它们保存到应用程序提供的缓冲区。可使用 WEB HTTPHEADER 命令分别检查各个 HTTP 请求的头。由 WEB RECEIVE 命令接收的项可以是：

- Web 客户机已向作为 HTTP 服务器的 CICS 发出的 HTTP 请求的主体。
- 由 CICS Web 支持设施处理的非 HTTP 消息，它使用 TCPIP SERVICE 定义中用户定义的 (USER) 协议。
- 来自另一个应用程序的请求，它已使用 CICS 业务逻辑接口直接联系应用程序，而不通过 CICS HTTP 侦听器。

数据以其转义格式返回。此命令可以指定 CICS 应用程序接收的入局数据使用的代码页转换类型。

选项

CHARACTERSET(*data-value*)

为接收项的实体主体指定 Web 客户机使用的字符集。字符集的名称最多可以由 40 个字母数字字符组成，包括适当的标点符号。CICS 不支持 IANA 命名的所有字符集。

当指定 CHARACTERSET 选项时，因为使用 SERVERCONV(SRVCONVERT)，所以会发生实体主体的代码页转换。作为由自己识别字符集的替换方法，指定 SERVERCONV(SRVCONVERT) 和 / 或 HOSTCODEPAGE 并省略 CHARACTERSET，可以让 CICS 为消息体识别字符集。

CLNTCODEPAGE(*data-value*)

仅支持此选项用于迁移目的。CHARACTERSET 可替换它。对于这两个关键字，CICS 采取的操作都是相同的。这意味着即使未指定 SERVERCONV 选项，指定 CLNTCODEPAGE 或 HOSTCODEPAGE 时也会发生代码页转换。如果省略了所有的代码页转换选项，则不会发生代码页转换。

HOSTCODEPAGE(*data-value*)

指定应用程序使用的 CICS（主机）代码页的 8 个字符的名称，应该将接收项的实体主体从已从 Web 客户机接收它的字符集转换到该代码页。当指定 HOSTCODEPAGE 选项时，因为使用 SERVERCONV(SRVCONVERT)，所以会发生实体主体的代码页转换。指定 SERVERCONV(SRVCONVERT) 和 / 或 CHARACTERSET 并省略 HOSTCODEPAGE 让 CICS 确定代码页。

如果未指定此选项，则缺省值为本地 CICS 区域的缺省代码页，如 LOCALCCSID 系统初始化参数中指定的那样。

INTO(*data-area*)

指定要包含正在接收的数据的缓冲区。

LENGTH(*data-area*)

指定全字二进制变量，它设置为 CICS 已返回给应用程序的数据量。

- 如果不指定 NOTRUNCATE 选项，则现在已废弃消息中出现的更多任何的数据。如果出现更多数据，则返回 RESP2 值为 57 的 LENGERR 响应。
- 如果指定 NOTRUNCATE 选项，则保留其他的任何数据。如果其他数据可用，则返回 RESP2 值为 36 的 LENGERR 响应。NOTRUNCATE 选项的描述将告诉您这种情况下要采取的操作。

MAXLENGTH(*data-value*)

以全字二进制值指定 CICS 要传递给应用程序的最大数据量。无论是否指定 INTO 或 SET 选项，都会应用 MAXLENGTH 选项来接收数据。如果已使用分块的传输编码发送了数据，则 CICS 将这些块汇编成单个消息，然后再将该消息传递给应用程序，因此 MAXLENGTH 选项应用于分块消息的总长度，而不是每个独立的块。发生任何代码页转换后测量数据。

NOTRUNCATE

指定当可用的数据长度超出 MAXLENGTH 选项上请求的长度时，不是立即废弃剩余的数据，而是保留数据，供后继 RECEIVE 命令来获取。（如果没有再发出 RECEIVE 命令，则在事务终止期间会废弃数据。）

带 SET 选项但不带 MAXLENGTH 选项的单条 RECEIVE 命令接收所有剩余的数据，而不管它的长度有多长。或者，您也可以使用一连串 RECEIVE 命令（带 NOTRUNCATE 选项）接收相应块中的剩余数据。继续发出 RECEIVE 命令直到您不再获得 LENGERR 响应。

SERVERCONV(*cvda*)

指定 CICS 是否将已接收项的实体主体从 Web 客户机使用的字符集转换到适合于应用程序使用的代码页。您可以在此命令中使用 CHARACTERSET 和 HOSTCODEPAGE 选项，来指定所使用的字符集和代码页。如果您指定这些选项中的任意一个，则采用代码页转换 (SRVCONVERT)。或者，您可全部省略这些选项或省略任意一个选项，指定 SERVERCONV(SRVCONVERT)，并让 CICS 确定合适的字符集和代码页。

SRVCONVERT

CICS 转换消息的实体主体。

NOSRVCONVERT

CICS 不转换该项的实体主体，而是以 Web 客户机使用的字符集将它传递给应用程序。如果您指定 NOSRVCONVERT，则不能指定 CHARACTERSET 或 HOSTCODEPAGE 选项。

注：如果您省略所有代码页转换选项（SERVERCONV, CLNTCODEPAGE, CHARACTERSET, HOSTCODEPAGE），则不会发生代码页转换。

SET(ptr-ref)

指定要设置为接收到的数据地址的指针引用。在发出下一个接收命令或任务结束之前，该指针引用一直有效。

TYPE(cvda)

返回接收到的请求的类型。CVDA 值有：

HTTPYES

表示 HTTP 请求。

HTTPNO

表示非 HTTP 请求。

对系统编程接口的更改

将 MAXDATALEN 选项添加到了 CREATE TCPIP SERVICE、INQUIRE TCPIP SERVICE 和 SET TCPIP SERVICE 命令：

MAXDATALEN(32|number)

指定作为 HTTP 服务器的 CICS 可接收的最大数据长度，以千字节为单位。缺省值为 32。最小值为 3，而最大值为 524288。

另外在 CREATE TCPIP SERVICE 和 INQUIRE TCPIP SERVICE 命令上，意味着用户定义的协议的新选项 USER 可用于 PROTOCOL 属性。

CICS 新提供的事务 CWXU（CICS Web 用户定义的协议连接事务）是在协议定义为 USER 时指定的事务。CWXU 执行程序 DFHWBXN。

在 CREATE TCPIP SERVICE 命令上，注意，对于 HTTP 协议，不应该为 SOCKETCLOSE 选项指定 0，这是因为此设置意味着无法维护持续连接。

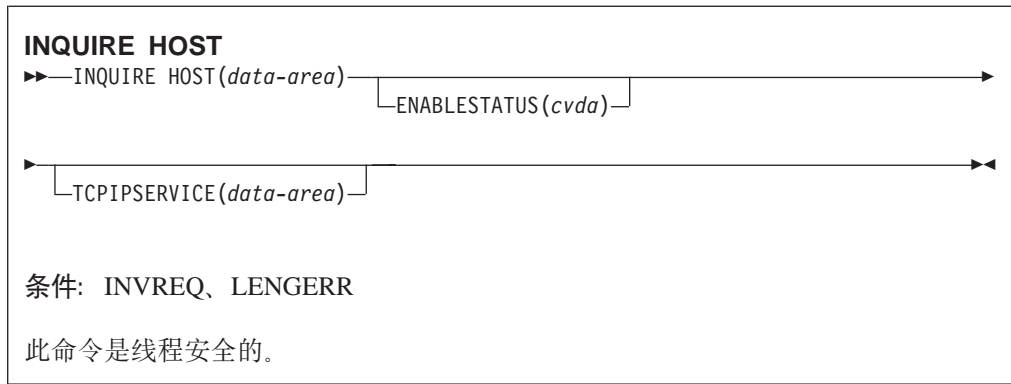
提供下列命令来管理虚拟主机：

- INQUIRE HOST
- SET HOST

CICS 根据您的 CICS 区域中的 URIMAP 定义自动管理虚拟主机的创建和删除。

INQUIRE HOST

获取有关本地系统中的虚拟主机的信息。



描述

INQUIRE HOST 命令允许您获取有关本地 CICS 区域中的特定虚拟主机的信息。第 89 页的『虚拟主管』说明了何为虚拟主机。虚拟主机基于 URIMAP 资源定义对象。通过把所有 URIMAP 定义一起分组为单个数据结构（指定相同主机名和相同 TCPIPSERVICE 的 CICS 区域中的所有 URIMAP 定义），CICS 自动为您创建虚拟主机。不指定 TCPIPSERVICE 的 URIMAP 定义（因此 URIMAP 定义适用所有 TCPIPSERVICE 定义）被添加到指定匹配主机名的所有数据结构中，因此这些 URIMAP 定义可以是多个数据结构的一部分。

浏览

您还可以使用 INQUIRE HOST 命令的浏览选项（START、NEXT 和 END）浏览区域中存在的所有虚拟主机。

选项

HOST(*data-value*)

指定虚拟主机的名称。每个虚拟主机的名称都取自组成虚拟主机的 URIMAP 定义中指定的主机名。例如，如果 CICS 区域包含指定 www.example.com 的主机名的 URIMAP 定义，则 CICS 将用名称 www.example.com 创建虚拟主机。URIMAP 定义中的主机名最多可达 120 个字符。

ENABLESTATUS(*cvda*)

返回表明此虚拟主机状态的 CVDA 值。CVDA 值为:

ENABLED

启用虚拟主机。

DISABLED

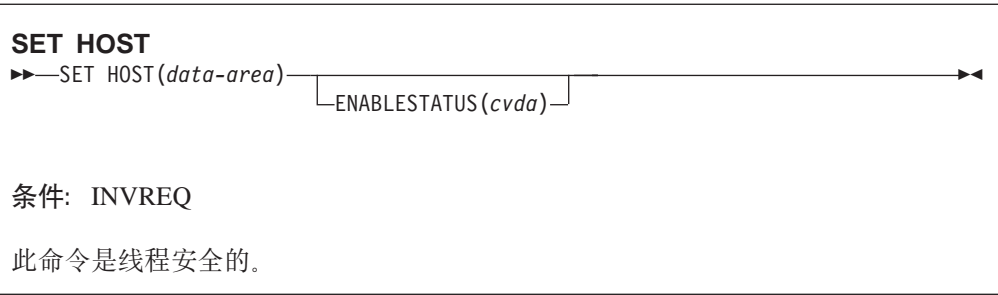
禁用虚拟主机。应用程序不能访问构成虚拟主机的 URIMAP 定义。

TCPIPSERVICE(*data-area*)

返回 TCPIPSERVICE 定义的 1 个到 8 个字符的名称，该定义指定与此虚拟主机相关的入站端口。如果不提供此定义，则虚拟主机与所有 TCPIPSERVICE 定义相关。

SET HOST

将虚拟主机的状态设置为已启用或已禁用。



描述

SET HOST 命令用于将虚拟主机的状态设置为启用或禁用。禁用虚拟主机意味着应用程序无法访问组成虚拟主机的所有 URIMAP 定义。当禁用虚拟主机时，CICS 将状态码为 503（服务不可用）的 HTTP 响应返回给 Web 客户机。

当 INQUIRE URIMAP 命令用于在个别 URIMAP 定义上查询时，返回特别状态 DISABLEDHOST，表明虚拟主机已禁用。您不需要单独更改 URIMAP 定义的禁用状态；可以使用 SET HOST 命令重新启用构成虚拟主机的所有 URIMAP 定义。然而，注意，状态为 DISABLEDHOST 的 URIMAP 定义无法废弃。如果您要废弃此定义，必须单独禁用它（使用 SET URIMAP 命令）。

选项

ENABLESTATUS(*cvda*)

CVDA 值为:

ENABLED

应用程序可以访问构成虚拟主机的 URIMAP 定义。

DISABLED

应用程序不能访问构成虚拟主机的 URIMAP 定义。

对 CEMT 的更改

CEMT INQUIRE 和 SET HOST 命令添加到 CEMT 事务。CICS 提供的事务中提供关于这些命令的信息。

对统计信息的更改

新统计信息最大数据（字段名 SOR_MAXDATA_LENGTH）已添加到 TCP/IP 服务资源统计信息。此统计信息显示 TCPIPSERVICE 定义的 MAXDATALEN 设置。TCP/IP 服务资源统计信息由 COLLECT STATISTICS 命令使用 TCPIPSERVICE 关键字收集，并由 DFHSORDS DSECT 映射。

统计程序 DFHOSTAT 包括显示 CICS 从您的 URIMAP 定义创建的虚拟主机的新报告：虚拟主机报告显示每个虚拟主机的名称，它是启用还是禁用，以及与其关联的 TCPIPSERVICE 定义的名称。如果未显示 TCPIPSERVICE 定义，则虚拟主机与 CICS 区域中的每个 TCPIPSERVICE 定义关联。

第 5 章 对 CICS Web 支持的一般增强

对 CICS Web 支持的增强包括在 CICS 作为 HTTP 服务器时对处理 HTTP 请求和响应的能力提高，对 CICSWeb 支持处理代码页转换方法改进，以及对 HTTP 时间和日期格式的更好支持。

处理 HTTP 请求和响应的改进支持

URIMAP 定义是新的 CICS 资源，其在 CICS 作为 HTTP 服务器时，为处理 HTTP 请求和响应提供已改进，功能更强大的工具。新的 URIMAP 资源定义意味着：

- 您可以提供使用来自 HFS 文件或 CICS 文档模板的内容的静态响应，并且使用应用程序向 Web 客户机提供动态响应。
- 您可使用更大的透明度处理 HTTP 请求。
- 您可执行对 HTTP 请求的联机管理。
- 使用分析器程序处理 HTTP 请求是可选的。

用于分析 HTTP 请求和响应的 CICS API 支持已扩展。EXEC CICS WEB EXTRACT 命令增强为从请求抽取更多信息，而新的 EXEC CICS WEB PARSE URL 命令允许您分析任何 URL，并抽取要重用的信息。

作为 HTTP 服务器的 CICS 使用的 EXEC CICS WEB SEND 和 WEB RECEIVE 命令已增强，这在从 Web 客户机接收请求并发送响应时，给予您更多的控制和能力。例如，您现在可使用数据缓冲区为响应提供消息主体（作为使用 CICS 文档的备用方法），而且您可指定在发送响应时确定的选项，并且 CICS 是否应该为客户机发出信号，以在接收响应后结束其连接。第 85 页的第 4 章，『升级到 HTTP/1.1 的 CICS Web 支持』中描述对这些命令的更改。

对代码页的改进支持

CICS Web 支持的代码页转换已改进，而且您不再需要设置代码页转换表（DFHCNV），以与 CICS Web 支持协同使用。

UTF-8 和 UTF-16 字符编码现在可用于 CICS 中的代码页转换。

对 HTTP 时间和日期格式的支持

新的 CICS API 命令 CONVERTTIME 将因特网上使用的常见日期和时间戳记格式转换为 CICS ABSTIME 格式。选项已添加到 EXEC CICS FORMATTIME 命令，以将 ABSTIME 格式转换为适合在因特网上使用的日期和时间戳记字符串。

CICS Web 支持增强的优点

CICS Web 支持的增强提供了下列优点：

- 使用 URIMAP 资源定义，而不是（或者以及）分析器程序更容易管理作为 HTTP 服务器的 CICS。URIMAP 定义可用于将 HTTP 请求直接指定给用户应用程序，而且您可以联机管理它们。

- 使用 URIMAP 定义，您可以让 CICS 自动提供静态响应给 Web 客户机，因此您无需编写和调用应用程序来提供此类型的响应。可以从 CICS 文档模板或 z/OS UNIX® 系统服务 HFS 文件形成静态响应，并且响应可以合并来自 Web 客户机的查询字符串的数据。
- UTF-8 和 UTF-16 字符编码可用于代码页转换。CICS Web 支持的代码页转换过程已改进，而且您不再需要设置代码页转换表（DFHCNV），以与 CICS Web 支持协同使用。
- CICS API 支持因特网上所使用日期和时间戳记格式，以及 CICS ABSTIME 格式之间的转换。

术语

下列术语在与 CICS Web 支持增强有关的内容中使用：

作为 HTTP 服务器的 CICS

它是进程，CICS 在那里接收 Web 客户机发出的 HTTP 请求并发送响应。用户应用程序可用于处理请求和提供响应，或使用 URIMAP 定义可指定静态响应。

作为 HTTP 客户机的 CICS

它是进程，用户应用程序在那里通过 CICS 将请求发送到 HTTP 服务器并接收响应。

静态响应

由来自文档模板的 CICS 或 URIMAP 定义指定的 HFS 文件构造的 HTTP 响应。

应用程序生成的响应

由用户应用程序动态构建的 HTTP 响应。这可以是支持 Web 的应用程序或不支持 Web 的应用程序。

支持 web 的应用程序

使用 EXEC CICS WEB 应用程序编程接口命令接收 Web 客户机请求和发送 HTTP 响应的应用程序。

不支持 Web 的应用程序

对于 CICS Web 支持，不使用 EXEC CICS WEB 应用程序编程接口命令的应用程序。可使用转换器程序为 Web 启用这些程序，此转换器程序将 Web 客户机的请求转换为可接受的输入，并根据程序的输出组成 HTTP 响应。

Web 客户机

对作为 HTTP 服务器的 CICS 发出 HTTP 请求的任何客户机应用程序。这可能是为人员用户、或自动用户代理（如搜索引擎的信息收集者）、或应用程序（如发出 HTTP 客户机请求的 CICS 应用程序）显示响应的 Web 浏览器。

分块传送编码（也称为分块）

将 HTTP 消息主体作为一系列块传输的过程，每个块具有其自己的块大小头。

管道传送

在不等待响应的情况下，客户机将多个 HTTP 请求发送到服务器的过程。然后，必须以接收请求时所用的相同顺序从服务器返回响应。

幂等性 个别 HTTP 方法或已进行管道传送的请求序列的属性。如果方法是幂等的，则

当您使用此方法重复相同的请求时，总是获得相同的结果。如果请求序列时幂等的，则当重复所有或部分请求系列时，总是获得相同的结果。

持续连接

Web 客户机和 HTTP 服务器之间的连接，可为多次请求和响应的交换复用。

URL URL（统一资源定位符）是 URI（通用资源标识）的特定类型。URL 可指定任何资源，然而，URL 通常在因特网上查找现有资源。

虚拟主管

单个 HTTP 服务器可在同一 IP 地址上表示多个主机的情况。此方法提供的每个主机名被认为是虚拟主机。

需求

要支持此功能，没有特别的硬件或软件需求。

相关信息

第 311 页的第 27 章，『CICS 操作环境』

作为 HTTP 服务器的 CICS 的 HTTP 请求和响应处理

向 CICS 发送请求的 Web 客户机发出对作为 HTTP 服务器的 CICS 的 HTTP 请求。CICS 向 Web 客户机提供对它发出的请求所作的响应。可以从 URIMAP 资源定义标识的静态文档创建这些响应，或可以由用户应用程序动态地创建它们。

下面描述执行适用于作为 HTTP 服务器的 CICS 的处理：

1. **CICS 接收 TCP/IP 连接请求。** CICS 套接字域使用端口的 TCPIP SERVICE 资源定义来确定 CICS Web 支持应该处理的请求。
2. **CICS 将请求的 URL 与 URIMAP 定义匹配起来（如果可能的话）。** 如果匹配成功，则 URIMAP 定义会告诉 CICS 如何处理请求。
3. **如果 URIMAP 定义指定重定向，则 CICS 将 Web 客户机重定向到指定的 URL。** CICS 构成重定向消息并且将它传送到 Web 客户机。
4. **如果 URIMAP 定义指定静态响应，则 CICS 构成响应并提供该响应。** CICS 使用文档模板或者 z/OS UNIX 系统服务 HFS 文件，并且加上适当的 HTTP 头以构成 HTTP 响应。响应在经过适当的代码页转换之后，CICS 将响应发送给 Web 客户机。
5. **如果 URIMAP 定义指定要运行分析器程序，或者如果未发现匹配的 URIMAP 定义，则可以运行分析器程序。** 分析器程序可以动态地解释请求，也可以用它来执行监视或审计。
6. **执行应用程序，为请求提供服务。** 您可以使用 URIMAP 定义或者使用分析器程序指定应用程序。使用 EXEC CICS WEB 和 EXEC CICS DOCUMENT 应用程序编程接口的支持 Web 的应用程序可以用来处理请求和构造响应。可以使用转换器程序（它将 Web 客户机的请求转换为可接受的输入并且根据程序的输出构成 HTTP 响应），或者使用调用不支持 Web 的程序并使用其输出的支持 Web 的应用程序，为 Web 启用不支持 Web 的应用程序。
7. **CICS 生成一些必需的 HTTP 头并将它们添加到消息。** 根据响应的 HTTP 版本生成适当的头。

8. **CICS 将完整的 HTTP 响应传送到 Web 客户机。**如果 Web 客户机支持持续连接，则 CICS 让连接保持打开状态以供可能的后继 HTTP 请求使用，直到用户应用程序或者 Web 客户机请求关闭或者到达超时期为止。

此过程期间，当消息进入和离开 CICS 环境时，经常需要进行代码页转换，以使得 CICS Web 支持处理和用户编写的应用程序（通常使用 EBCDIC 编码）能够与 Web 客户机（通常使用 ASCII 编码）进行通信。

CICS Web 支持程序中的 Unicode UTF-8 和 UTF-16 代码页转换

CICS Web 支持现在可以处理转换到 Unicode UTF-8 和 UTF-16 字符编码、从 UTF-8 和 UTF-16 字符编码转换到其他格式以及 UTF-8 和 UTF-16 字符编码之间的代码页相互转换。

UTF-8 和 UTF-16 的代码页转换使用 z/OS 提供的转换服务。您的 z/OS 系统中必须启用转换工具，如 *Unicode* 的 z/OS 支持：使用转换服务中描述的那样。

代码页转换通常是需要的，以致其通常使用 EBCDIC 编码的 CICS Web 支持处理和用户编写的应用程序可以同通常使用 ASCII 编码的 Web 客户机进行通信。入站消息为要处理的应用程序转换为 EBCDIC 编码，而应用程序生成的出站消息从 EBCDIC 编码转换为 Web 客户机的合适字符集。

UTF-8 和 UTF-16 字符编码可用于 CICS 从因特网接收或发送到因特网的项的消息体。您可使用 CICS 支持的任何 EBCDIC 代码页转换到 UTF-8 和 UTF-16 字符编码，以及从 UTF-8 和 UTF-16 字符编码转换到其他格式。

注意，CICS 文档和文档模板不能转换为 UTF-8 和 UTF-16 字符编码。如果您要以这些字符编码发送出站消息，则在 WEB SEND 或 WEB CONVERSE 命令上使用 FROM 选项，以指定要组成消息体的数据缓冲区，而不是使用 DOCTOKEN 选项指定 CICS 文档。

处理 HTTP 日期和时间戳记格式

通过 CICS 与 Web 交互的应用程序可能需要在 HTTP 头中产生正确的日期和时间戳记格式。应用程序还可能需要使用从 Web 接收到的日期和时间戳记信息。CONVERTTIME 和 FORMATTIME 命令，使您能够处理 HTTP 使用的常见结构化日期和时间戳记字符串格式。

您可使用 CONVERTTIME 命令将结构化日期和时间戳记字符串转换为 ABSTIME。您不需要识别日期和时间戳记的格式；CONVERTTIME 命令会识别在因特网上常用的三种不同日期和时间戳记格式并进行转换。

您可使用 FORMATTIME 命令将当前日期和时间（以 ABSTIME 格式），或应用程序产生的日期和时间（例如到期日期）转换为 RFC 1123 格式。

对 CICS 外部的更改

对资源定义的更改

DOCTEMPLATE 资源定义的新属性 HFSFILE

此属性使得 z/OS UNIX 系统服务 HFS 文件能够用作文档模板。

HFSFILE(*hfsfile*)

当模板驻留在 z/OS UNIX 系统服务 HFS 文件中时，这指定 HFS 文件的标准名称（绝对）或相对名称。此名称可指定为包括所有目录，并以斜杠开头的绝对名称，例如 /u/facts/images/bluefish.jpg。或者，它可以指定为相对于 CICS 区域用户标识的 HOME 目录的名称，例如 facts/images/bluefish.jpg。可使用多达 255 个字符。CICS 区域必须有访问 z/OS UNIX 的许可权，而且它必须具有访问包含文件的 HFS 目录以及文件本身的许可权。

URIMAP 资源定义

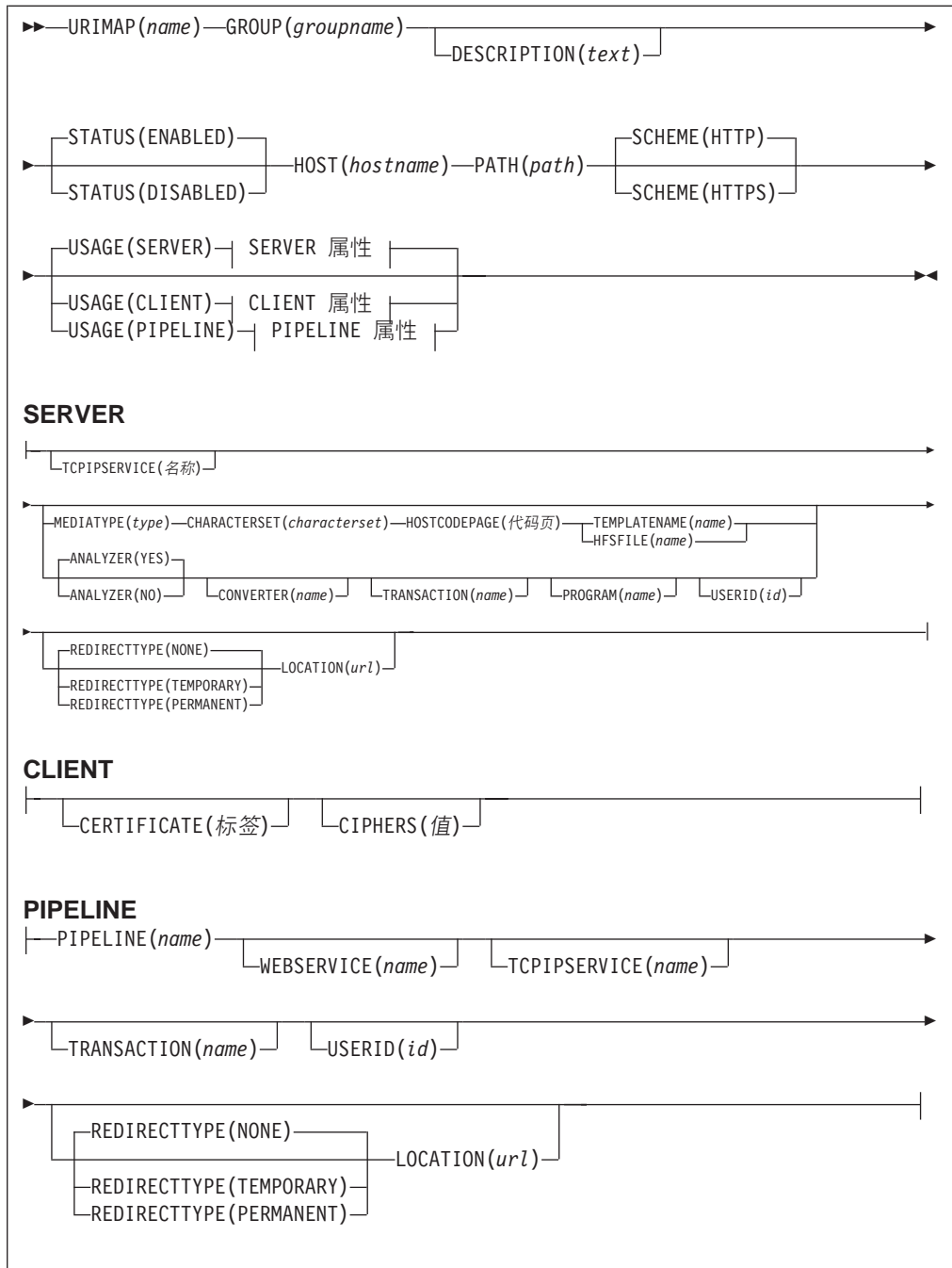
URIMAP 定义是资源定义，它匹配 HTTP 或 Web 客户请求的 URI，并提供有关如何处理请求的信息。

URIMAP 定义用于提供 CICS 中三个不同的 Web 相关工具：

1. 从 **Web 客户机**到作为 **HTTP 服务器**的 **CICS** 的请求。对于作为 HTTP 服务器的 CICS 的请求，它们的 URIMAP 定义的 USAGE 属性为 SERVER。这些 URIMAP 定义与 CICS 期待从 Web 客户机接收的 HTTP 请求的 URL 匹配，并且它们定义 CICS 应如何向每个请求提供响应。您可使用 URIMAP 定义告诉 CICS：
 - 使用文档模板或 z/OS UNIX 系统服务 HFS 文件向 HTTP 请求提供静态响应。
 - 使用应用程序向 HTTP 请求提供动态响应。
2. 从作为 **HTTP 客户机**的 **CICS** 到服务器的请求。对于来自作为 HTTP 客户机的 CICS 的请求，它们的 URIMAP 定义的 USAGE 属性为 CLIENT。这些 URIMAP 定义指定担当 Web 客户机的用户应用程序通过 CICS Web 支持发出请求到 HTTP 服务器时使用的 URL。为实现此目的设置 URIMAP 定义意味着您可以避免识别您的应用程序中的 URL。
3. **Web Service 请求**。用于 Web Service 请求的 URIMAP 定义具有的 USAGE 属性是 PIPELINE。这些 URIMAP 定义将入站 Web Service 请求（即，客户机在 CICS 中调用 Web Service 的请求）与指定要执行的处理的 PIPELINE 或 WEBSERVICE 资源相关联。

对于作为 HTTP 服务器的 CICS，URIMAP 定义合并先前与 TCPIP SERVICE 定义关联的分析器程序提供的大部分功能。如果需要，处理路径中可仍然涉及分析器程序。

URIMAP 定义属性：



ANALYZER({NO|YES})

此属性用于 **USAGE(SERVER)**，在这种情况下提供应用程序生成的响应。（对于 **USAGE(CLIENT)** 或 **USAGE(PIPELINE)**，此属性强制为 NO。）它指定处理 HTTP 请求时是否使用分析器程序。可使用此属性运行的分析器是与 **TCPIPSERVICE** 定义或 **URIMAP** 定义相关的定义关联的分析器。（分析器程序必须在本地 CICS 区域中。）值 YES 表示运行分析器。值 NO 意味着不使用分析器。

CERTIFICATE(标签)

此属性用于 **USAGE(CLIENT)**。它指定要在 SSL 握手期间用作 SSL 客户机证书的 X.509 证书的标号。证书标号长度可多达 32 个字符。只有在 **URIMAP** 定义指定的 URI 用于作为客户机的 CICS 发出的 HTTP 请求时，才使用此属性。由服务

器负责请求 SSL 客户机证书，如果发生这种情况，CICS 会提供 URIMAP 定义中指定的证书标号。如果省略此属性，则使用 CICS 区域用户标识的密钥环中定义的缺省证书。该证书必须存储在外部安全性管理器数据库中的密钥环中。

CIPHERS(值)

此属性用于 **USAGE(CLIENT)**。

指定多达 56 个十六进制数字的字符串，该字符串被解释为多达 28 个 2 位的密码套件代码列表。会自动用可接受代码的列表作为属性值，这取决于 ENCRYPTION 系统初始化参数已指定的加密级别。

- 对于 ENCRYPTION=WEAK，缺省值为 03060102
- 对于 ENCRYPTION=MEDIUM，缺省值为 0903060102
- 对于 ENCRYPTION=STRONG，缺省值为 0504352F0A0903060102

您可以重新排序密码，或者从缺省列表中除去它们。然而，您无法为指定的加密级别添加不在缺省列表中的密码。要将值重置为缺省代码列表，删除所有密码套件代码，字段会用缺省列表自动重新植入。

CHARACTERSET(character set)

此属性用于 **USAGE(SERVER)**，在这种情况下提供静态响应。它指定字符集 1 到 40 个字符的名称，CICS 将发送到 Web 客户机的响应的实体主体转换成此字符集。CICS 不支持 IANA 命名的所有字符集。此属性的值包括在响应的 Content-Type 头中。

如果提供了静态响应，并且 MEDIATYPE 属性指定文本类型，则必须指定 CHARACTERSET。

CONVERTER(名称)

此属性用于 **USAGE(SERVER)**，在这种情况下提供应用程序生成的响应。它指定转换器程序 1 到 8 个字符的名称，运行此程序可执行请求和响应的转换或其他处理。通常，转换器程序将 HTTP 请求转换成应用程序可使用的 COMMAREA，而将输出转换成 HTTP 响应。转换器程序可以是本地 CICS 区域中可用的任何转换器程序。

DESCRIPTION(text)

您可在此字段中提供要定义的资源描述。描述文本的长度可以达到 58 个字符。不限定您使用的字符。然而，如果您使用圆括号，则确保每个左圆括号都有一个匹配的右圆括号。如果您使用 CREATE 命令，则对于文本中的每个单撇号，编码成两个撇号。

GROUP(group name)

每个资源定义都必须有一个 GROUP 名。在安装组时资源定义成为组的成员并安装在 CICS 系统中。

可接受的字符:

A-Z 0-9 \$ @ #

您输入的任何小写字符都会被转换成大写字符。

GROUP 名的长度最多可达到八个字符。小写字符会被当作大写字符对待。不要使用以 DFH 开始的组名，因为这些字符保留供 CICS 使用。

HFSFILE(名称)

此属性用于 **USAGE(SERVER)**，在这种情况下提供静态响应。它指定 z/OS UNIX 系统服务 HFS 文件的标准（绝对）名称或相对名称，其组成发送给来自 Web 客户机的 HTTP 请求的静态响应主体。此名称可指定为包括所有目录，并以斜杠开头的绝对名称，例如 /u/facts/images/bluefish.jpg。或者，它可以指定为相对于 CICS 区域用户标识的 HOME 目录的名称，例如 facts/images/bluefish.jpg。可使用多达 255 个字符。

注：资源级安全性不应用到作为静态响应传递的项。如果您需要将基于用户标识的访问控制应用到以这种方式传递的项，则需要改为传递作为应用程序生成的响应的资料。

如果您要使用路径匹配，则在 HFS 文件的名称末尾、以及 PATH 属性指定的路径末尾包括星号作为通配符。CICS 获取每个 HTTP 请求路径用通配符涵盖的一部分，并替换它作为文件路径的最后一部分。

例如，您会创建 URIMAP 定义，它的 PATH 属性指定为：

```
findout/pictures/*
```

并且 HFSFILE 属性指定为：

```
/u/facts/images/*
```

URIMAP 定义用于处理入局 HTTP 请求

```
http://www.example.com/findout/pictures/bluefish.jpg
```

CICS 将 bluefish.jpg 追加到 URIMAP 定义中指定的 HFS 文件路径来代替星号，这样 HFS 文件

```
/u/facts/images/bluefish.jpg
```

就可用于形成静态响应。

注：不能在 HFSFILE 规范中单独使用星号。必须指定至少一个目录结构级别。

查询字符串无法替代到 HFS 文件中（除非您定义 HFS 文件作为 CICS 文档模板，并使用 TEMPLATENAME 选项，而不是 HFSFILE 选项指定它）。

HOST(hostname)

此属性用于所有 **USAGE** 选项。它指定应用 URIMAP 定义的 URI 的主机部分，它的长度可多达 116 个字符。主机名的示例是 www.example.com。

HOST 属性必须出现且只能包含字母数字字符、连字符 (-) 或句点 (.)。主机名中不能使用十六进制转义序列。CICS 在定义时验证此名称。输入主机名时可以用大写或小写，但在 URIMAP 定义中它会转换成小写。

IPv4 地址可用作主机名，但不支持 IPv6 地址。

对于与作为 HTTP 服务器的 CICS 相关的 URIMAP 定义，可使用一个星号来代替 HOST 属性。这使 URIMAP 定义匹配任何主机名。在 HOST 属性中星号不能作为通配符，与任何其他字符一起使用。

HOSTCODEPAGE(代码页)

此属性用于 **USAGE(SERVER)**，在这种情况下提供静态响应。它指定用于构成静态响应的文本文档的 IBM® 代码页（EBCDIC）的 1-10 个字符的名称。CICS 执行静态响应的实体主体的代码页转换时需要此信息。

如果提供了静态响应，并且 **MEDIATYPE** 属性指定文本类型，则必须指定 **HOSTCODEPAGE**。

LOCATION(*url*)

此属性用于 **USAGE(SERVER)**和 **USAGE(PIPELINE)**。它指定将用于重定向客户端的请求的 URL，它的长度最多可达到 255 个字符。它必须是完整的 URL，包括模式、主机和路径部分和适当的定界符。因为 CICS 不检查 URL 是否有效，所以您必须确保目标存在并且正确地指定 URL。

REDIRECTTYPE 属性用于指定重定向的类型。如果指定临时或永久重定向，则 **LOCATION** 属性中的 URL 用于重定向。如果指定不重定向，则忽略 **LOCATION** 属性中的 URL。您可使用 **SET URIMAP** 命令更改 **REDIRECTTYPE** 属性和 **LOCATION** 属性。

MEDIATYPE(*类型*)

此属性用于 **USAGE(SERVER)**，在这种情况下提供静态响应。它指定 CICS 向 HTTP 请求提供的静态响应介质类型（数据内容），例如，`image/jpg`、`text/html` 或 `text/xml`。最多可使用 56 个字符。介质类型必须只包含一个正斜杠 (/)。输入介质类型时可以用大写或小写，但在 **URIMAP** 定义中它会转换成小写。

每种正式地公认的数据内容类型的名称由 IANA 定义。<http://www.iana.org/assignments/media-types/> 提供了一个列表。CICS 使用此属性的值创建响应的 **Content-Type** 头。

此属性没有缺省值，并且必须指定它。如果 **MEDIATYPE** 属性指定文本类型（如，以 `text/` 开始的类型，或包含 `+xml` 的类型），则还必须指定 **CHARACTERSET** 和 **HOSTCODEPAGE** 属性以便可以进行代码页转换。文本介质类型由 RFC 3023 标识，其可在 <http://www.ietf.org/rfc/rfc3023.txt> 上获得，并取代 RFC 2376。

对于动态（应用程序生成）响应，不使用此属性。响应的介质类型由 **WEB SEND** 命令指定。

PATH(*路径*)

此属性用于所有 **USAGE** 选项。它指定应用 **URIMAP** 定义的 URI 的路径部分，它的长度可多达 255 个字符。路径的示例是 `software/http/cics/index.html`。最小的可能路径为 /（正斜杠），其表示指定主机名的 URL 结构的根目录。

PATH 属性以大小写混合的方式指定，混合字母保存在 **URIMAP** 定义中。**PATH** 属性必须只包含 URI 中允许的字符。特别地，应排除字符 `< > # % “ { } | \ ^ [] `` 和嵌入空格（除了在这种情况下会允许 `%`：当 `%` 引入有效的十六进制转义序列，即，当它后跟两个大写或小写的有效十六进制数字时）。`~` 字符不能在 CICS 中指定，而且必须由相应的十六进制转义序列（`%7E`）替换。CICS 在定义时验证字符的使用。

对于与作为 HTTP 服务器的 CICS 和 Web Service 相关的 **URIMAP** 定义，如果您希望 **URIMAP** 定义与多个路径匹配，则您可在路径末尾使用星号作为通配符。例如，指定路径 `/software/http/cics/*` 将使 **URIMAP** 定义与路径以星号左侧的字符串开始的所有请求匹配。指定 `/*` 的路径会使 **URIMAP** 定义匹配指向 **HOST** 属性中指定的主机的任何请求。如果 HTTP 请求与多个 **URIMAP** 定义匹配，则采用最特定的匹配。

如果出现查询部分，并且您要将 **URIMAP** 定义单独应用到此特定的查询，则可以将此查询包括为路径部分的一部分。在字符串的开始包括问号。查询字符串必须只包含 URI 中允许的字符。查询字符串本身不能包括星号作为通配符，但它可以跟随包

括星号作为通配符的路径。如果在 URIMAP 定义中不包含查询字符串，则为了匹配目的会自动忽略 HTTP 请求中出现的任何查询字符串。

对于作为 HTTP 客户机的 CICS 的 URIMAP 定义，您不能使用星号作为通配符；您必须指定请求的完整路径。如果在 WEB OPEN 命令中引用 URIMAP 定义，则此路径会成为与此连接相关的 WEB SEND 命令的缺省路径。如果在 WEB SEND 命令中引用 URIMAP 定义，则此路径用于该 WEB SEND 命令，但注意，此 URIMAP 定义的主机属性必须与该连接的 WEB OPEN 命令中指定的主机匹配。

PIPELINE(*name*)

此属性用于 **USAGE(PIPELINE)**。当客户机对 URI 由 URIMAP 定义所指定的 CICS 发出入站 Web Service 请求，则 PIPELINE 为 Web Service 指定 PIPELINE 响应定义的 1-8 位字符名称。PIPELINE 资源定义提供处理客户机的服务请求的消息处理程序的信息。

PROGRAM(*名称*)

此属性用于 **USAGE(SERVER)**，在这种情况下提供应用程序生成的响应。它指定组成 HTTP 响应的用户应用程序的 1-8 位字符名称。对于作为 HTTP 服务器的 CICS，除非指定分析器程序或转换器程序、或指定模板名称或 HFS 文件或指定重定向，否则此属性是必需的。

REDIRECTTYPE(**{NONE|TEMPORARY|PERMANENT}**)

此属性用于 **USAGE(SERVER)**和 **USAGE(PIPELINE)**。它指定匹配此 URIMAP 定义的请求的重定向类型。当需要时，LOCATION 属性指定的 URL 用于重定向。

- NONE 意味着不重定向请求。忽略 LOCATION 属性指定的任何 URL。
- TEMPORARY 意味着在临时基础上重定向请求。LOCATION 属性指定的 URL 用于重定向，用于响应的状态码为 302（找到）。
- PERMANENT 意味着请求永久重定向。LOCATION 属性指定的 URL 用于重定向，用于响应的状态码为 301（永久移动）。

您可使用 SET URIMAP 命令更改 REDIRECTTYPE 属性和 LOCATION 属性。

SCHEME(**{HTTP|HTTPS}**)

此属性用于所有 **USAGE** 选项。它指定应用 URIMAP 定义的 URI 的模式部分，它是 HTTP（不带 SSL）或 HTTPS（带 SSL）。不要在 URI 中模式部分的后面包括定界符 `://`（冒号和两个正斜杠）。

注：指定 HTTP 规则的 URIMAP 接受使用 HTTP 规则或 HTTPS 规则发出的 Web 客户机请求。指定 HTTPS 规则的 URIMAP 仅接受使用 HTTPS 规则发出的 Web 客户机请求。

STATUS(**{ENABLED|DISABLED}**)

此属性用于所有 **USAGE** 选项。它指定要将 URIMAP 定义安装为启用状态还是禁用状态。缺省为启用状态。

TCIPSERVICE(*名称*)

此属性用于 **USAGE(SERVER)**和 **USAGE(PIPELINE)**。它为具有 PROTOCOL(HTTP) 的 TCIPSERVICE 资源定义指定 1 到 8 个字符的名称，它定义此 URIMAP 定义相关的入站端口。如果不指定此属性，则 URIMAP 定义会应用到任何入站端口上的请求。

当将 HTTPS 作为规则的 URIMAP 定义与 Web 客户机发出的请求匹配时，CICS 检查此请求使用的入站端口是否使用 SSL。如果不为此端口指定 SSL，则拒绝请

求，其状态码为 403（已禁止）。当 URIMAP 定义应用到所有入站端口时，该检查会确保 Web 客户机无法使用不安全的端口来访问安全的资源。因为不会对指定 HTTP 作为规则的 URIMAP 定义实施检查，所以 Web 客户机可使用不安全的端口或安全的（SSL）端口访问这些资源。

TEMPLATENAME(名称)

此属性用于 **USAGE(SERVER)**，在这种情况下提供静态响应。它指定 CICS 文档模板 1-48 位字符名称，此模板组成发送给来自 Web 客户机的 HTTP 请求的静态响应主体。它必须使用 DOCTEMPLATE 资源定义来定义，并且此定义的 TEMPLATENAME 属性指定 URIMAP 定义中使用的名称。请参阅 *CICS Application Programming Guide* 以获取关于形成 CICS 文档模板的指示信息。

注：资源级安全性不应用到作为静态响应传递的项。如果您需要将基于用户标识的访问控制应用到以这种方式传递的项，则需要改为传递作为应用程序生成的响应的资料。

如果您要使用路径匹配，则在 CICS 文档模板的名称末尾、以及 PATH 属性指定的路径末尾包括星号作为通配符。CICS 获取通配符所覆盖的每个 HTTP 请求的路径部分，并将它替换为模板名称的最后部分。

例如，您会创建 URIMAP 定义，它的 PATH 属性指定为：

```
findout/about/*
```

并且 TEMPLATENAME 属性指定为：

```
templates.facts.*
```

URIMAP 定义用于处理入局 HTTP 请求

```
http://www.example.com/findout/about/fish.html
```

CICS 将 fish.html 追加到 URIMAP 定义中指定的模板名来代替星号，这样模板 templates.facts.fish.html

就可用于形成静态响应。

注：为 TEMPLATENAME 属性单独指定星号，这意味着所选模板将与 PATH 属性中通配符相应的 URL 部分具有相同的名称。

当指定 TEMPLATENAME 属性时，如果查询字符串出现在 URI 中但未用于 PATH 属性时，CICS 自动将查询字符串的内容作为符号列表传递到命名的 CICS 文档模板中。如果您要将查询字符串的内容用于文档模板，则需要您的文档模板中包括相应的变量以代替查询字符串的内容。

TRANSACTION(名称)

此属性用于 **USAGE(SERVER)**，在这种情况下提供应用程序生成的响应，以及 **USAGE(PIPELINE)**。它指定别名事务的 1-4 位字符名称，此别名事务用于运行组成 HTTP 响应的用户应用程序，或用于启动管道化。此缺省别名事务为 CWBA 用于 USAGE(SERVER) 或者 CPIH 用于 USAGE(PIPELINE)。您可以为了安全性目的、监控和记帐或事务类限制选择不同的事务名称。无论您为别名事务选择什么名称，它必须总是运行 USAGE 属性确定的同一程序。对于 USAGE(SERVER)，程序为 DFHWBA，其链接到在 URIMAP 定义的 PROGRAM 属性中（或由分析器程序指

定)指定的应用程序。对于 USAGE(PIPELINE), 程序为 DFHPIDSH, 其启动 PIPELINE 属性中指定的管道 (以及如果指定的话, WEBSERVICE 属性中指定的 Web Service)。

URIMAP(名称)

指定此 URIMAP 定义的名称。此名称的长度可以达到八个字符。该属性以大小写混合的方式指定, 并且大小写保存在 URIMAP 定义中。

可接受的字符:

A-Z 0-9 \$ @ #

USAGE({SERVER|CLIENT|PIPELINE})

指定此 URIMAP 定义用于作为 HTTP 服务器的 CICS (SERVER)、或是用于作为 HTTP 客户机的 CICS (CLIENT) 还是用于 Web Service (PIPELINE)。USAGE 属性控制可以使用 URIMAP 定义中的其他哪些属性。

指定 SERVER 会为作为 HTTP 服务器的 CICS 创建 URIMAP 定义。这种类型的 URIMAP 定义用于将入局 HTTP 请求的 URI 从 Web 客户机映射到 CICS 资源。可提供应用程序生成的响应或静态响应。

指定 CLIENT 会为作为 HTTP 客户机的 CICS 创建 URIMAP 定义。当 CICS 请求服务器上的 HTTP 资源时使用这种类型的 URIMAP 定义, 以便您可以避免在您的应用程序中标识 URI。

指定 PIPELINE 会为 Web Service 创建 URIMAP 定义。此类 URIMAP 定义用于入站 Web Service 请求 (即, 客户机调用 CICS 中 Web Service 的请求)。入局请求的 URI 与 WEBSERVICE 和 PIPELINE 相关联, 其指定要在消息上执行的处理。

USERID(id)

此属性用于 USAGE(SERVER), 在这种情况下提供应用程序生成的响应, 以及 USAGE(PIPELINE)。它指定连接别名事务的 1-8 位字符的用户标识。您在 URIMAP 定义中指定的用户标识直接从客户机获得的任何用户表示所覆盖, 使用 TCPIPService 定义的 AUTHENTICATE 属性为此连接指定的认证过程。如果指定 ANALYZER(YES), 来自 URIMAP 定义或来自客户机的用户标识可由分析器程序所更改, 或者分析器程序可设置用户标识。如果未通过这些方式中的任何一种指定用户标识, 则缺省用户标识为 CICS 缺省用户。

WEBSERVICE(name)

此属性用于 USAGE(PIPELINE)。当客户机对 URI 由此 URIMAP 定义所指定的 CICS 发出入站 Web Service 请求时, WEBSERVICE 指定所使用 WEBSERVICE 资源定义的 1-8 位字符名称。此资源定义定义 Web Service 设置中部署的 CICS 应用程序运行时环境的一些方面, 其中应用程序数据结构和 SOAP 消息之间的映射已使用 CICS 提供的工具生成。

对应用程序编程接口 (一般 CICS Web 支持增强) 的更改

新的以及更改过的命令

存在新的 EXEC CICS WEB 命令, 用于作为 HTTP 客户机的 CICS。而且, 对于作为 HTTP 客户机的 CICS, 一些命令已修改, 而对于作为 HTTP 服务器的 CICS, 这些命令未更改。第 63 页的『对应用程序编程接口 (HTTP 客户机请求) 的更改』中描述了新的和已更改的客户机命令。

EXEC CICS WEB SEND 和 EXEC CICS WEB RECEIVE 命令在被作为 HTTP 服务器的 CICS 使用时增强 HTTP/1.1 的支持，这在第 92 页的『对应用程序编程接口 (HTTP/1.1 支持) 的更改』中有所描述。

下列新的 EXEC CICS 命令可用于作为 HTTP 服务器的 CICS 和作为 HTTP 客户机的 CICS。

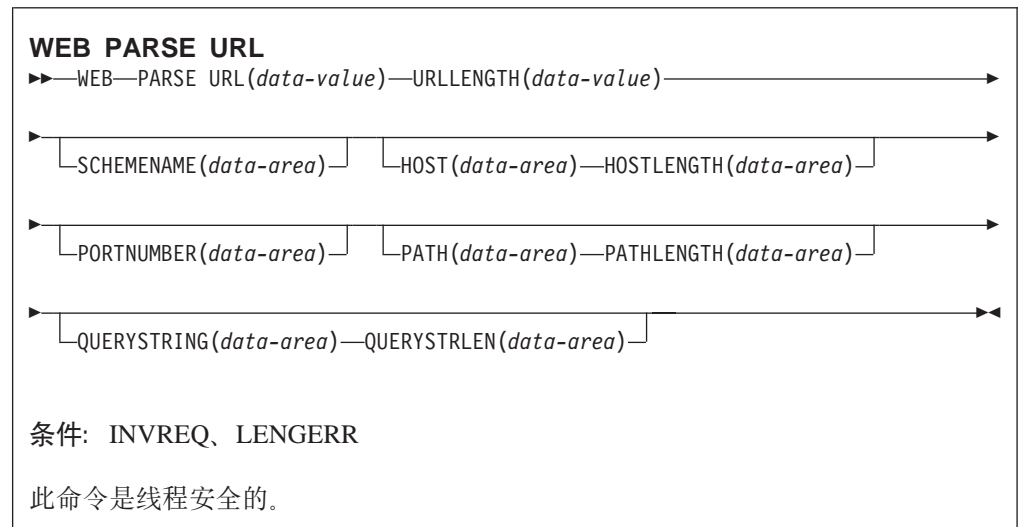
- EXEC CICS WEB PARSE URL
- EXEC CICS CONVERTTIME

下列 EXEC CICS 命令对于作为 HTTP 服务器的 CICS 和作为 HTTP 客户机的 CICS 都已修改。

- EXEC CICS WEB EXTRACT
- EXEC CICS FORMATTIME

WEB PARSE URL

将 URL 字符串分解为它的组成部分。



描述

WEB PARSE URL 使您能将 URL 字符串分解为它的组成部分：方案、主机、端口、路径和查询字符串。您可以使用此过程来检查 URL 的构成，并分离出各部分。可以在 WEB OPEN 命令中使用返回的信息打开与 URL 中指定的主机的客户机连接。

为有效性检查 URL 中找到的任何转义序列。转义序列包括百分号 (%)，后跟两个十六进制字符。有效十六进制字符是数字 0 到 9，以及字符 A 到 F。

注意，以正确方法为 URL 定界对 WEB PARSE URL 命令的字符串输入，此命令不能检测无效的内容，如不代表因特网上现有主机的主机名，或者 URL 中不允许的字符。

选项

HOST(*data-area*)

返回 URL 的主机部分。这可以是字母数字主机名也可以是数字的 IP 地址。如果在 URL 中显式指定端口号，则它作为 PORTNUMBER 选项独立返回。

IPv4 地址可用作 WEB OPEN 命令中的主机名，但 IPv6 地址不受支持。WEB PARSE URL 命令认为 IPv6 地址是无效的而拒绝它，这是因为 IPv6 地址不符合要求的结构。

HOSTLENGTH(*data-area*)

以全字二进制变量指定 HOST 选项上提供的缓冲区的长度，并将其设置为返回给应用程序的数据的实际长度（主机名）。建议指定 116 个字符作为此 *data-area* 的适当大小。如果数据超出缓冲区长度，则会发生 LENGERR 情况，且数据被截断。

PATH(*data-area*)

返回 URL 的路径部分。

PATHLENGTH(*data-area*)

以全字二进制变量指定 PATH 选项上提供的缓冲区的长度，并将其设置为返回给应用程序的数据的实际长度（URL 的路径部分）。建议指定 256 个字符作为此 *data-area* 的适当大小。如果数据超出缓冲区长度，则会发生 LENGERR 情况，且数据被截断。

PORTNUMBER(*data-area*)

以全字二进制数据区返回 URL 中指定的或适于该 URL 的端口号。有时在 URL 中显式指定端口号（跟在主机名后）。但是，通常从 URL 省略用于服务的熟知端口号。如果 URL 中未呈现端口号，则 WEB PARSE URL 命令会根据方案标识它并将它返回。对于 HTTP，熟知端口号为 80，而对于 HTTPS，熟知端口号为 443。如果返回的端口号不是方案的缺省值，您将需要显式指定该端口号，以获取对 URL 的访问（例如，如果您正在 WEB OPEN 命令中使用此信息）。

QUERYSTRING(*data-area*)

从 URL 返回查询字符串。查询字符串是在限定路径结束的问号 (?) 之后编码的一个或多个值。查询字符串以其转义格式返回。

QUERYSTRLEN(*data-area*)

以全字二进制变量指定 QUERYSTRING 选项上提供的缓冲区的长度，并将其设置为返回给应用程序的数据的实际长度（查询字符串）。建议指定 256 个字符作为此 *data-area* 的适当大小。如果数据超出缓冲区长度，则会发生 LENGERR 情况，且数据被截断。

SCHEMENAME(*data-area*)

以 16 个字符的数据区返回 URL 的方案部分。仅 HTTP 方案和 HTTPS 方案（带 SSL 和不带 SSL 的 HTTP 协议）受 CICS 的支持，并且可以在 WEB OPEN 命令中使用。

URL(*data-value*)

指定完整的 URL 字符串。

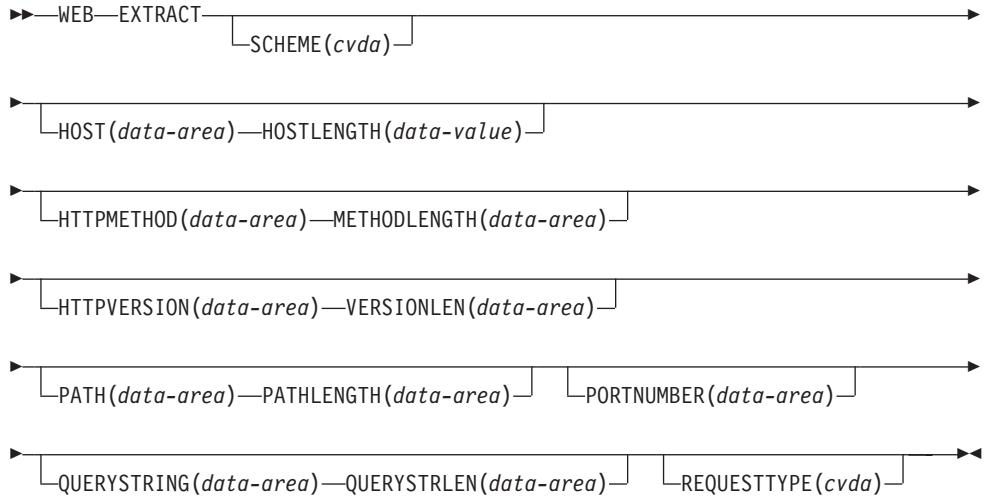
URLLENGTH(*data-value*)

以全字二进制值指定包含 URL 字符串的缓冲区的长度。

WEB EXTRACT

获取有关向作为 HTTP 服务器的 CICS 发出的 HTTP 请求的信息，或者有关因特网服务器和作为 HTTP 客户机的 CICS 之间的连接的信息。

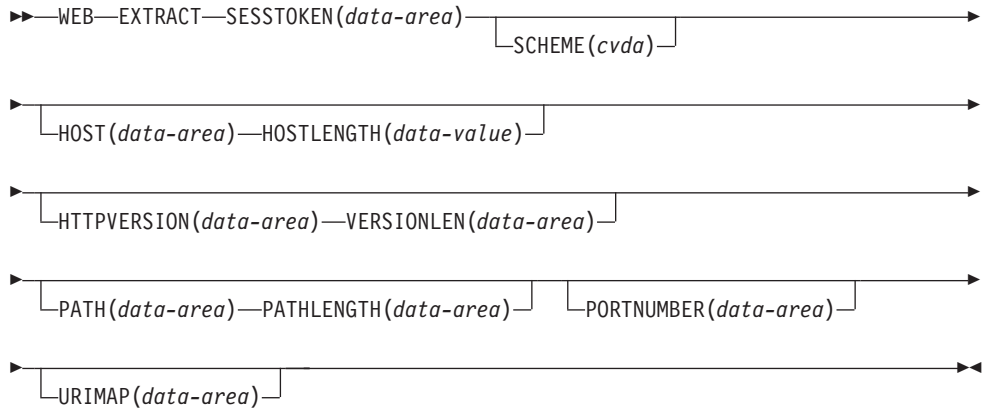
WEB EXTRACT (作为 HTTP 服务器的 CICS)



条件: ILLOGIC、INVREQ、LENGERR、NOTOPEN

此命令是线程安全的。

WEB EXTRACT (作为 HTTP 客户机的 CICS)



条件: ILLOGIC、INVREQ、LENGERR、NOTOPEN

此命令是线程安全的。

选项

HOST (data-area)

对于作为 HTTP 服务器的 CICS，此选项指定一个缓冲区以包含 URL 的主机组件，该 URL 是在请求的“主机头”字段中或者在请求行中（如果请求使用的是绝对 URI）指定的。使用 PORTNUMBER 选项独立呈现端口号。

对于作为 HTTP 客户机的 CICS（带 SESSTOKEN 选项），此选项指定一个缓冲区以包含 SESSTOKEN 选项识别的连接中的服务器的主机名。使用 PORTNUMBER 选项独立呈现端口号。

HOSTLENGTH(*data-area*)

以全字二进制变量指定 HOST 选项上提供的缓冲区的长度，并将其设置为返回给应用程序的数据的实际长度。116 个字符是为此 *data-area* 指定的适当大小。如果数据超出缓冲区长度，则会发生 LENGERR 条件，且数据被截断。

HTTPMETHOD(*data-area*)

对于作为 HTTP 服务器的 CICS，此选项指定一个缓冲区以包含消息的请求行上的 HTTP 方法字符串。

作为 HTTP 客户机的 CICS 不使用此选项。

HTTPVERSION(*data-area*)

对于作为 HTTP 服务器的 CICS，此选项指定一个缓冲区以包含 Web 客户机的 HTTP 版本（如其请求上表明的）。

对于作为 HTTP 客户机的 CICS（带 SESSTOKEN 选项），此选项指定一个缓冲区以包含 SESSTOKEN 选项识别的连接中的服务器的 HTTP 版本。

“1.1”表示 HTTP/1.1，而“1.0”表示 HTTP/1.0。

注：CICS 对低于 HTTP/1.0 级别的服务器或 Web 客户机不提供任何特殊支持。

CICS 的行为就象它们所处的级别都是 HTTP/1.0，并返回“1.0”作为 HTTP 版本。

如果您的应用程序编写的 HTTP 头可能不适用于 HTTP/1.0 级别的 Web 客户机或服务器，或者如果您打算将分块消息发送到服务器（HTTP/1.0 级别的服务器无法接收该消息），则您的应用程序还应该查询 HTTP 版本信息。

METHODLENGTH(*data-area*)

以全字二进制变量指定 HTTPMETHOD 选项上提供的缓冲区的长度，并将其设置为返回给应用程序的数据的实际长度。如果数据超出缓冲区长度，则会发生 LENGERR 条件，且数据被截断。

PATH(*data-area*)

对于作为 HTTP 服务器的 CICS，此选项指定一个缓冲区以包含消息的请求行中指定的路径。

对于作为 HTTP 客户机的 CICS（带 SESSTOKEN 选项），此选项指定一个缓冲区以包含适用于使用此连接发出的请求的缺省路径。如果在 WEB OPEN 命令上为连接指定了 URIMAP 定义，则缺省路径是 URIMAP 定义中指定的路径。否则，缺省路径是一个正斜杠。

PATHLENGTH(*data-area*)

以全字二进制变量指定 PATH 选项上提供的缓冲区的长度，并将其设置为返回给应用程序的数据的实际长度。256 个字符是为此 *data-area* 指定的适当大小。如果数据超出缓冲区长度，则会发生 LENGERR 条件，且数据被截断。

PORTNUMBER(*data-area*)

对于作为 HTTP 服务器的 CICS，此选项返回一个数据区，该数据区包含消息的请求行中指定的端口号。

对于作为 HTTP 客户机的 CICS (带 SESSTOKEN 选项), 此选项返回一个数据, 该数据包含用于访问 SESSTOKEN 选项指定的连接中的服务器的端口号。

数据区中返回的值是全字二进制值。

通常从 URL 省略用于服务的熟知端口号。如果 URL 中未呈现端口号, 则 WEB EXTRACT 命令会根据方案标识它并将它返回。对于 HTTP, 熟知端口号为 80, 而对于 HTTPS, 熟知端口号为 443。如果返回的端口号不是方案的缺省值, 您将需要显式指定该端口号, 以获取对 URL 的访问 (例如, 如果您正在 WEB OPEN 命令中使用此信息)。

QUERYSTRING(*data-area*)

对于作为 HTTP 服务器的 CICS, 此选项指定一个缓冲区以包含消息的请求行上的查询字符串。查询字符串是在限定路径结束的问号 (?) 之后编码的一个或多个值。查询字符串以其转义格式返回。

作为 HTTP 客户机的 CICS 不使用此选项。

QUERYSTRLEN(*data-area*)

以全字二进制变量指定 QUERY 选项上提供的缓冲区的长度, 并将其设置为返回给应用程序的数据的实际长度 (查询字符串)。256 个字符是为此 *data-area* 指定的适当大小。如果数据超出缓冲区长度, 则会发生 LENGERR 条件, 且数据被截断。

REQUESTTYPE(*cvda*)

对于作为 HTTP 服务器的 CICS, 此选项指定接收的请求类型。作为 HTTP 客户机的 CICS 不使用此选项。CVDA 值有:

HTTPYES

表示 HTTP 请求。

HTTPNO

表示非 HTTP 请求。

SCHEME(*cvda*)

对于作为 HTTP 服务器的 CICS 和作为 HTTP 客户机的 CICS (带 SESSTOKEN 选项), 此选项返回 CICS 和 Web 客户机或服务器之间的连接使用的方案。CVDA 值有:

HTTP 是不带 SSL 的 HTTP 协议。

HTTPS

是 HTTPS 协议, 该协议是带 SSL 的 HTTP。

SESSTOKEN(*data-value*)

对于作为 HTTP 客户机的 CICS, 必需此选项。指定 8 个字节的二进制值会话标记, 它唯一地识别 CICS 和服务器之间的连接。WEB OPEN 命令为作为 HTTP 客户机的 CICS 返回此值。第 62 页的『会话标记』说明会话标记的用途。对于 WEB EXTRACT 命令, 返回有关指定的连接的信息。

作为 HTTP 服务器的 CICS 不使用此选项。

URIMAP(*data-value*)

对于作为 HTTP 客户机的 CICS (带 SESSTOKEN 选项), 此选项返回任何 URIMAP 定义的 8 个字符的名称, 该 URIMAP 定义是在 WEB OPEN 命令上指定的用于打开 SESSTOKEN 选项指定的连接。INQUIRE URIMAP 命令可以用于查找有关此 URIMAP 定义的属性的信息。

作为 HTTP 服务器的 CICS 不使用此选项。

VERSIONLEN(*data-area*)

以全字二进制变量指定 HTTPVERSION 选项上提供的缓冲区的长度，并将其设置为返回给应用程序的数据的实际长度。

CONVERTTIME

将结构化日期和时间戳记字符串转换为 ABSTIME 格式。

CONVERTTIME

►—CONVERTTIME—DATESTRING(*data-area*)—ABSTIME(*data-area*)—◄

条件: INVREQ、LENGERR

此命令是线程安全的。

描述

CONVERTTIME 对因特网上经常使用的三种不同日期和时间戳记格式进行分析，并将它们转换为 ABSTIME（绝对日期和时间）格式。

ABSTIME 格式以压缩十进制表示时间，从 1900 年 1 月 1 日 00:00 始（以毫秒为单位，四舍五入至最接近一秒的百分之一）。FORMATTIME 命令可以用于将时间更改为其他格式。

CONVERTTIME 命令识别的结构化日期和时间戳记字符串格式为:

RFC 1123 格式

RFC 1123 中指定的 HTTP 协议的日期和时间戳记的首选标准格式。用此格式表示的日期和时间戳记的示例为“ Tue, 01 Apr 2003 10:01:02 GMT”。

RFC 850 格式

更早在因特网上使用的日期和时间戳记格式。用此格式表示的日期和时间戳记的示例为“Tuesday, 01-Apr-03 10:01:02 GMT”。

重要: 因为采用此格式，年份只有两个数字，因此 CICS 使用年份的范围为从 1970 到 2069 的假设。在上面的示例中，CICS 将假设文档的日期为 2003 年 4 月 1 日。给出日期和时间戳记“Thursday, 13-Feb-98 15:30:00 GMT”，CICS 将假设文档的日期为 1998 年 2 月 13 日。如果您考虑采用这种格式接收日期和时间戳记，请在对应用程序进行编码时记住这点。

ASctime 格式

从 C asctime 函数输出的日期和时间戳记格式。用此格式表示的日期和时间戳记的示例为“Tue Apr 1 10:01:02 2003”。

选项

DATESTRING(*data-area*)

指定 64 个字符的 *data-area*，以包含结构化日期和时间戳记字符串。您可以用命令可识别的任何格式来提供字符串，并且不需要指定使用的是哪种格式。

ABSTIME(*data-area*)

指定 *data-area* 以接收 ABSTIME 格式的已转换的日期和时间戳记。对于此 *data-area* 的格式，请参阅 ASKTIME 命令的描述。如果日期和时间戳记的格式不是可识别的，则不返回 ABSTIME。

对 EXEC CICS WEB 命令选项的更改

将 DATESTRING 和 STRINGFORMAT 选项添加到 FORMATTIME 命令，来以因特网上使用的格式返回日期和时间戳记。

FORMATTIME

将 DATESTRING 和 STRINGFORMAT 选项添加到 EXEC CICS FORMATTIME 命令。这些选项使得您能够将 ABSTIME 格式的日期和时间戳记转换为结构化的日期和时间戳记字符串。目前 STRINGFORMAT 选项提供的唯一格式是遵从 RFC 1123 标准的格式，这适合于在因特网上使用。

DATESTRING(*data-area*)

指定 64 个字符的用户字段，在那里 CICS 将以 STRINGFORMAT 选项指定的格式返回结构化的日期和时间戳记字符串。如果不指定 STRINGFORMAT，提供的缺省格式为 RFC 1123 格式 (RFC1123)。

STRINGFORMAT(*cvda*)

指定 DATESTRING 中返回的结构化的日期和时间戳记字符串的格式。目前唯一可用的 CVDA 值为：

RFC1123

指定适合于在因特网上使用的 RFC 1123 格式。此日期和时间戳记字符串包含以 GMT 表示的日子、日期和 24 小时的时间，例如“Tue, 01 Apr 2003 10:01:02 GMT”。

对系统编程接口的更改

将 HFSFILE 选项添加到了 INQUIRE DOCTEMPLATE 和 CREATE DOCTEMPLATE 命令：

HFSFILE(*filename*)

指定模板所驻留 z/OS UNIX 系统服务 HFS 文件的标准（绝对）或相对名称。该名称的长度可以多达 255 个字符。

提供下列命令来管理 URIMAP 定义：

- CREATE URIMAP
- DISCARD URIMAP
- INQUIRE URIMAP
- SET URIMAP

CREATE URIMAP 命令

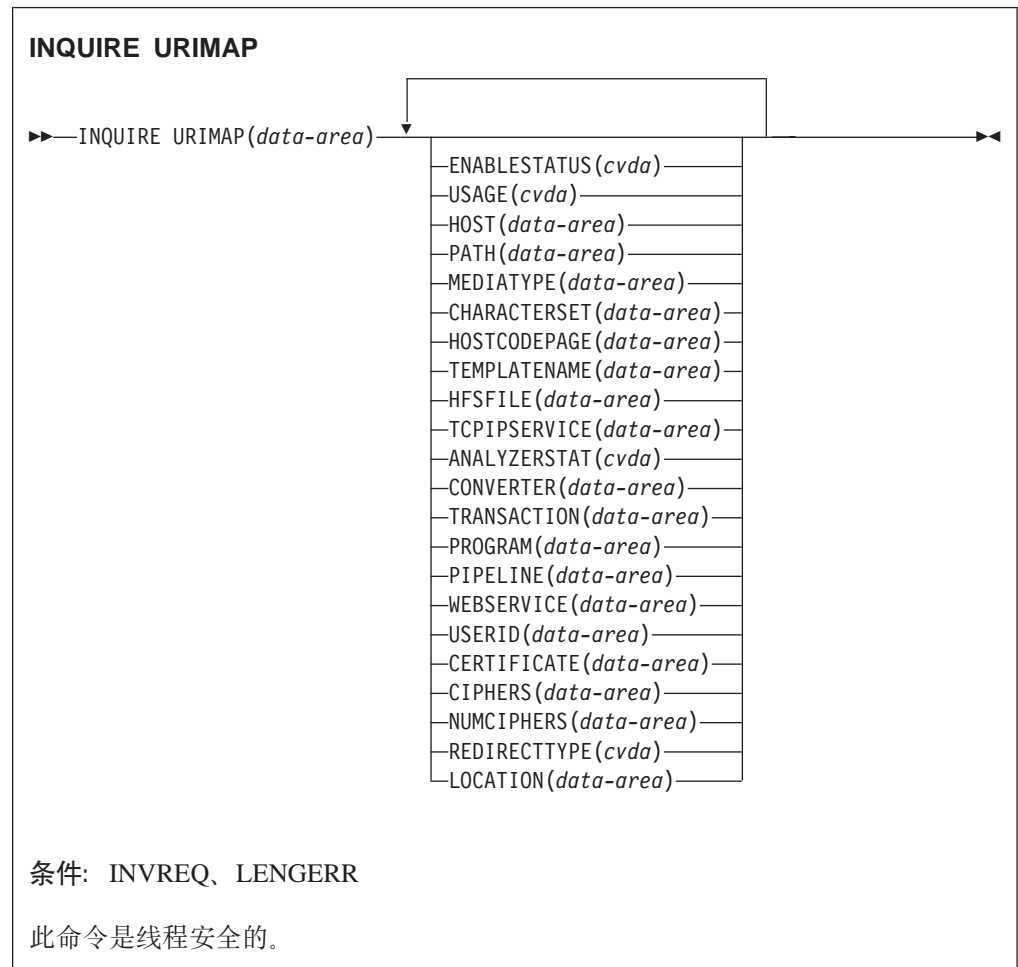
使用 CREATE URIMAP 命令在您的 CICS 区域中动态地创建 URIMAP。第 107 页的『URIMAP 定义属性』中描述您可以在此命令上指定的属性。

DISCARD URIMAP 命令

使用 DISCARD URIMAP 从 CICS 区域中除去 URIMAP。可以废弃 URIMAP 之前，必须先禁用它。

INQUIRE URIMAP

获取有关本地系统中的 URIMAP 资源的信息。



描述

INQUIRE URIMAP 命令允许您获取有关特定 URIMAP 定义的信息。URIMAP 定义的 USAGE 属性确定该 URIMAP 定义中指定的其他属性，并且有时候还确定特定属性的含意。

浏览

您还可以使用 INQUIRE URIMAP 命令的浏览选项（START、NEXT 和 END）浏览区域中安装的所有 URIMAP 定义。

选项

URIMAP(*data-value*)

指定 URIMAP 定义的 8 个字符的名称。

ANALYZERSTAT(*cvda*)

返回 CVDA 值，表明是否要运行与 TCPIP SERVICE 定义关联的分析器程序。
CVDA 值为：

ANALYZER

要运行分析器程序。

NOANALYZER

不要运行分析器程序。

此属性用于 USAGE(SERVER)。对于所有其他的用法类型，它被强制设置为 NO。

CERTIFICATE(*data-area*)

返回包含证书标号的 32 个字符的数据区，该证书要用作 CICS（作为 HTTP 客户机）发出的 HTTP 请求的 SSL 客户机证书。此属性用于 USAGE(CLIENT)。

CHARACTERSET(*data-area*)

返回包含用于静态响应的字符集名称的 40 位字符数据区。此属性用于 USAGE(SERVER)。

CIPHERS(*data-area*)

返回 56 个字符的数据区，包含为 URIMAP 定义指定的密码套件列表。密码套件列表用于协商 SSL 连接。此属性用于 USAGE(CLIENT)。

CONVERTER(*data-area*)

返回转换器程序的 8 位字符名称，该程序执行转换或对作为 HTTP 服务器的 CICS 的其他处理。此属性用于 USAGE(SERVER)。

ENABLESTATUS(*cvda*)

返回表明此 URIMAP 定义状态的 CVDA 值。CVDA 值为：

ENABLED

启用 URIMAP 定义。

DISABLED

禁用 URIMAP 定义。可废弃此状态的 URIMAP 定义。

DISABLEDHOST

URIMAP 定义由于禁用了它所属的虚拟主机而不可用。SET HOST 命令可用于重新启用构成虚拟主机的所有 URIMAP 定义。无法废弃此状态的 URIMAP 定义。

HFSFILE(*data-area*)

返回包含组成静态响应的 z/OS UNIX 系统服务 HFS 文件的标准（绝对）或相对名称的 255 个字符数据区。此属性用于 USAGE(SERVER)。

HOST(*data-area*)

返回包含应用 URIMAP 定义的 URI（例如，www.example.com）的主机组件的 116 个字符的数据区。此属性适用于任何用法类型。

HOSTCODEPAGE(*data-area*)

返回包含在其中构成编码的静态响应的文本文档的 IBM 代码页（EBCDIC）的 10 位字符名称的 10 位字符数据区。此属性用于 USAGE(SERVER)。

LOCATION(*data-area*)

返回包含 URL 的 255 个字符的区域，把 Web 客户机发出的匹配 HTTP 请求重定向到该 URI。通过 REDIRECTTYPE 选项指定的设置激活重定向。此属性用于 USAGE(SERVER) 或 USAGE(PIPELINE)。

MEDIATYPE(*data-area*)

返回包含描述静态响应的数据内容的 56 个字符的数据区。此属性用于 USAGE(SERVER)。

NUMCIPHERS(*data-area*)

返回包含 CIPHERS 列表中密码数的半字二进制值。密码用于协商作为 SSL 握手的一部分的加密级别。此属性用于 USAGE(CLIENT)。

PATH(*data-area*)

返回包含应用 URIMAP 定义的 URL 的路径部分（例如，software/htp/cics/index.html）的 255 位字符数据区。此属性适用于任何用法类型。

PIPELINE(*data-area*)

返回 Web Service 的 PIPELINE 资源定义的 8 个字符名称。PIPELINE 资源定义提供有关处理客户机的服务请求的消息处理程序的信息。此属性用于 USAGE(PIPELINE)。

PROGRAM(*data-area*)

返回应用程序的 8 位字符名称，该应用程序组成对 HTTP 请求的应用程序生成响应。此属性用于 USAGE(SERVER)。

SCHEME(*cvda*)

返回表明 URI 的模式组件的 CVDA 值。CVDA 值为：

HTTP 不带 SSL 的 HTTP。

HTTPS

带 SSL 的 HTTP。

此属性适用于任何用法类型。

TCPIPSERVICE(*data-area*)

返回 TCPIPSERVICE 定义的 1 个到 8 个字符的名称，该定义指定与此 URIMAP 定义相关的入站端口。如果不指定此属性，URIMAP 定义会应用到任何入站端口上的请求。此属性用于 USAGE(SERVER) 或 USAGE(PIPELINE)。

TEMPLATENAME(*data-area*)

返回包含 CICS 文档模板名称的 48 个字符的数据区，该模板用于构成静态响应。此属性用于 USAGE(SERVER)。

TRANSACTION(*data-area*)

返回要运行用户应用程序的别名事务的 4 个字符的名称，该用户应用程序组成对 HTTP 请求的响应。此属性用于 USAGE(SERVER) 或 USAGE(PIPELINE)。

REDIRECTTYPE(*cvda*)

返回表明匹配此 URIMAP 定义的请求重定向类型的 CVDA 值。重定向的 URL 由 LOCATION 选项指定。此属性用于 USAGE(SERVER) 或 USAGE(PIPELINE)。CVDA 值为：

NONE 请求不重定向。忽略 LOCATION 选项指定的任何 URL。

TEMPORARY

在临时基础上重定向请求。用于响应的状态码为 302（找到）。

PERMANENT

请求永久重定向。用于响应的状态码为 301（永久移动）。

USAGE(*cvda*)

返回表明此 URIMAP 定义的用途的 CVDA 值。CVDA 值为：

SERVER

作为 HTTP 服务器的 CICS 的 URIMAP 定义。这种类型的 URIMAP 定义用于将入局 HTTP 请求的 URL 从 Web 客户机映射到 CICS 资源。可提供应用程序生成的响应或静态响应。

CLIENT

作为 HTTP 客户机的 CICS 的 URIMAP 定义。CICS 在服务器上发出对 HTTP 资源的客户机请求时，使用此类型的 URIMAP 定义。

PIPELINE

Web Service 的 URIMAP 定义。此类 URIMAP 定义用于指定客户机在 CICS 中调用 Web Service 的请求上要执行的处理。

USERID(*data-area*)

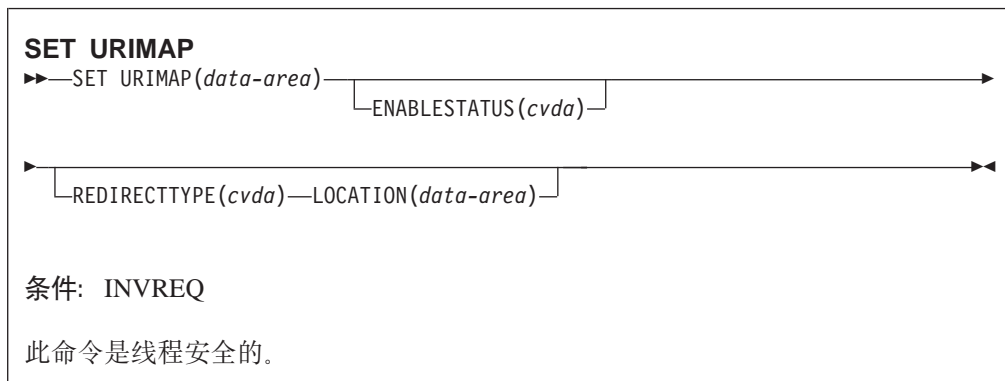
返回 8 个字符的用户标识，在该用户标识下连接别名事务。此属性用于 USAGE(SERVER) 或 USAGE(PIPELINE)。

WEBSERVICE(*data-area*)

返回 WEBSERVICE 资源定义的 8 个字符的名称。此资源定义 Web Service 设置中部署的 CICS 应用程序运行时环境的一些方面。此属性用于 USAGE(PIPELINE)。

SET URIMAP

启用或禁用 URIMAP 定义，并且应用或除去 URIMAP 定义的重定向。



描述

SET URIMAP 命令允许您：

- 启用或禁用 URIMAP 定义。
- 设置于匹配 HTTP 请求的重定向，并指定重定向请求所至的 URL。例如，如果正常响应 HTTP 请求的应用程序不可用，则您可使用此命令将重定向应用到现有 URIMAP 定义。您还可使用此命令从 URIMAP 定义除去重定向。

选项

ENABLESTATUS(*cvda*)

将 URIMAP 定义设置为启用或禁用状态。CVDA 值为:

ENABLED

应用程序可以访问 URIMAP 定义。

DISABLED

应用程序不能访问 URIMAP 定义。只有禁用 URIMAP 定义，才能重新安装或禁用它。

LOCATION(*data-area*)

指定可多达 255 个字符的 URL，把 Web 客户机发出的匹配 HTTP 请求重定向到该 URI。它必须是完整的 URL，包括模式、主机和路径部分和适当的定界符。因为 CICS 不检查 URL 是否有效，所以您必须确保目标存在并且正确地指定 URL。

REDIRECTTYPE 选项用于指定重定向的类型。如果指定临时或永久重定向，则 LOCATION 属性中的 URL 用于重定向。如果指定 NONE，则忽略 LOCATION 选项中的 URL。

REDIRECTTYPE(*cvda*)

指定匹配此 URIMAP 定义的请求的重定向类型。重定向的 URL 由 LOCATION 选项指定。CVDA 值为:

NONE 请求不重定向。忽略 LOCATION 选项指定的任何 URL。

TEMPORARY

在临时基础上重定向请求。用于响应的 HTTP 状态码为 302 (找到)。

PERMANENT

请求永久重定向。用于响应的 HTTP 状态码为 301 (永久移动)。

对 CEMT 的更改

将下列新命令添加到 CEMT 事务:

- INQUIRE URIMAP
- SET URIMAP
- DISCARD URIMAP
- INQUIRE HOST
- SET HOST

对下列命令的更改:

- INQUIRE TCPIPSERVICE
- SET TCPIPSERVICE
- INQUIRE DOCTEMPLATE

CICS 提供的事务提供有关这些新命令和已更改命令的信息。

对 CICS 提供的事务的更改

CICS 新提供的事务 CWXU

在 CICS Transaction Server for z/OS V3R1 中，对 HTTP 请求的处理和对非 HTTP 请求的处理保持独立。这确保 CICS 可在 HTTP 请求和响应上执行基本验收检查，如第 85 页的第 4 章，『升级到 HTTP/1.1 的 CICS Web 支持』中所描述，以及确保非 HTTP 请求不从属于这些检查。非 HTTP 请求的处理现在必须在用户定义的（USER）协议下执行，此协议在 TCPIP SERVICE 定义上为接收请求的端口指定。

当协议定义为 USER 时，新的 CICS 所提供事务 CWXU（CICS Web 用户定义的协议附加事务）是缺省值。CWXU 执行 CICS 程序 DFHWPBXN。DFHWPBXN 样本包括 CWXU 的样本定义。除了在 TCPIP SERVICE 资源定义上为协议定义的其他缺省事务，可使用执行 DFHWPBXN 的备用事务。

CWXU 是 RACF Category 1 事务。

对 CWXN 的更改

对 CICS 提供的事务 CWXN（Web 连接事务）执行的处理有一些更改。其中，最重要的更改为：

- 如果为 HTTP 请求查找匹配 URIMAP 定义，则 CWXN 现在仅在 URIMAP 定义指示要调用分析器程序时才调用它。
- 其中请求的 HTTP 版本为 HTTP/1.1，通过对请求执行一些基础验收检查，CWXN 执行 HTTP 服务器的某些职责。响应这些检查，CWXN 可采取行动在不涉及用户所编写应用程序的情况下，将响应返回给请求。
- CWXN 对从 Web 客户机接收的已分块和已进行管道传送的消息进行预处理，所有此用户编写的应用程序无须执行此处理。
 - 已分块消息是已分割，并作为一系列小消息（块）发送的单个消息。CWXN 接收和组装消息块，以创建单个 HTTP 请求。CWXN 检查消息在传递给用户应用程序前是完整的。然后，用户应用程序可处理象任何其他 HTTP 请求一样的请求。
 - 已进行管道传送的消息是按顺序发送的多个消息，其中发送方在每条消息发送后不等待响应。服务器必须按接收消息的顺序响应这些消息。要确保此操作，CWXN 保留管道请求，并以每次一个的顺序将它们释放给用户应用程序。用户应用程序必须在从 CWXN 接收下一个请求前，将响应发送给第一个请求。
- 持续连接现在是缺省行为。仅当 Web 客户机请求结束，或者达到超时时间，或者 Web 客户机是不发送保持活动头的 HTTP/1.0 客户机时才结束连接。
- 在 CICS Transaction Server for z/OS V3R1 之前，如果 Web 客户机和 CICS 具有持续连接，则在持续连接期间，CWXN 视图保留在系统中。现在，来自 Web 客户机的每个请求已传递给别名视图（CWBA 或另一个视图）后，或者已传递静态响应后，CWXN 视图终止。套接字侦听器任务监控套接字并在持续连接上为每个请求启动 CWXN 的新实例。此行为（称为异步接收）避免在达到最大任务规范（MXT）的情况下可能发生的死锁。这还意味着 CICS 和 Web 客户机之间的最大并发连接数不再受到 MXT 值的限制，但是从理论上来说最高为 64000。在系统活动方面，如果您在 CICS Transaction Server for z/OS V3R1 之前使用持续连接，则您现在应该看到事务率增加，但是并发任务数减少。

CICS Web 支持视图的优先级 (CWXN、CW XU、CWBA 或其他别名事务)

如果您将 CWXN 或 CW XU 事务的优先级设置为高于用于应用程序所生成响应的别名事务 (如 CWBA) 的优先级, 这会导致内建请求已接收, 但是还未处理, 引发缺少存储的情况。

CWXN 或 CW XU 事务的缺省优先级设置为 1, 而 CICS 所提供 CWBA 事务的缺省优先级也设置为 1。您可根据工作负载调整这些优先级。确保用于应用程序生成的响应 (如 CWBA) 的别名事务的优先级等于或高于与 Web 连接任务 (如 CWXN 或 CW XU) 关联的事务的优先级。如果您设置自己的事务定义替代 CICS 提供的缺省值, 则请记住此规定。

对用户可替代程序的更改

对分析器程序的更改

供 CICS Web 支持处理个别 HTTP 请求使用的分析器程序现在是可选的。URIMAP 定义可用于将请求的 URL 匹配到处理它们的应用程序, 而且它们可指定使用转换器程序和别名事务。如果这些只是分析器程序在现有 CICS Web 支持体系结构中执行的任务, 则您可使用 URIMAP 定义替换请求处理路径中的函数。

您可能具有来自早期 CICS 发行版 (其提供您在请求处理期间需要的其他功能, 如将数据传递给转换器程序, 或者为使用转换器程序的不支持 Web 的应用程序修改代码页转换) 的现有分析器程序。如果是这种情况, 您可继续为相关请求使用分析器程序替换 URIMAP 定义, 或者您可将它与 URIMAP 定义结合 (通过指定 ANALYZER(YES) 选项)。

当分析器程序与 URIMAP 定义协同使用时, URIMAP 定义的元素 (如处理请求的应用程序名) 作为输入传递给分析器程序。现有或新的分析器程序可用于对这些元素进行动态更改。您还可以使用分析器程序, 将监控或审计操作引入该过程。

必须仍然为用于 CICS Web 支持的每个 TCPIP SERVICE 资源定义指定分析器程序。如果 CICS 未找到匹配的 URIMAP 定义用于此请求, 则调用为 TCPIP SERVICE 定义指定的分析器程序, 以处理 HTTP 请求。这可能是由于用户在输入请求 URL 时出错, 或者因为未安装适当的 URIMAP 定义。(如果 URIMAP 定义存在但禁用, 则请求由 Web 出错程序来处理而不是由分析器程序处理。)

存在对分析器程序的 COMMAREA 中所指定输入和输出参数的多个更改:

- 提供新输入参数 `wbra_urimap`, 标识处理请求路径时, 何时涉及匹配的 URIMAP 定义。
- 参数 `wbra_alias_tranid`、`wbra_converter_program`、`wbra_server_program` 和 `wbra_userid` 现在是输入参数, 也是输出参数。当使用 URIMAP 定义时, URIMAP 定义的 TRANSACTION、CONVERTER、PROGRAM 和 USERID 属性 (分别) 传递给分析器程序, 作为这些程序的输入参数, 然后分析器程序可选择覆盖这些参数。
- 为代码页转换提供两个新的输出参数。Web 客户机使用的字符集可由 `wbra_character_set` 参数指定, 而 `wbra_hostcodepage` 参数指定适合于应用程序的主机代码页。CICS 使用这些参数, 在将请求传递给转换器程序 (如果使用的话) 或应用程序前, 执行代码页转换。输出参数的功能等价于现有 `wbra_dfhcnv_key` 参数, 使用该参数的重要差异在于您不必在代码页转换表 (DFHCNV) 中创建条目。您只需

指定字符集和主机代码页，而 CICS 确定相应的转换模板。如果现有分析器程序使用 `wbra_dfhcnv_key` 参数，则在您更改为新参数，或者从请求的处理路径中除去分析器程序前，您需要为了迁移目的，保留有关的 DFHCNV 条目。

- 提供新的输出参数 `wbra_commarea`，表明不使用 EXEC CICS WEB API 命令的应用程序需要预 CICS TSV3 兼容性处理。此标志用于特定环境中的现有应用程序，在此环境中，Web 客户机需要的响应等同于在 CICS TS V3 之前它应该接收的响应。设置此标志意味着：
 - CICS 不添加在 CICS TS V3 之前还未使用的任何响应头。
 - 如果需要错误处理，CICS 发送适合于，并标记为 HTTP/1.0 响应的错误响应，无论 Web 客户机的 HTTP 版本是什么。CICS 通常应该应答带有 HTTP/1.1 错误响应的 HTTP/1.1 客户机，但是这可能使客户机错误地认为应用程序通常应该在 HTTP/1.1 级别上发送响应。

URIMAP 定义可为请求设置，但是它必须指定分析器程序。

当 CICS 作为 HTTP 客户机，或者用于 Web Service 调用时，不能调用分析器程序；仅当 CICS 作为 HTTP 服务器时才调用它们。

对转换器程序调用的更改

供 CICS Web 支持处理 HTTP 请求使用的转换器程序仍然是可选的。转换器程序主要与不为协同使用 Web 而原始编码的应用程序一起使用，并需要以 COMMAREA 格式接收输入。它们可用于将输出从一个或多个这些应用程序转换为 HTTP 消息。支持 web 的应用程序（用 EXEC CICS WEB 和 EXEC CICS DOCUMENT 应用程序接口编码的应用程序）应该不需要执行此转换。

URIMAP 定义可指定转换器程序为 HTTP 请求执行相关的处理。URIMAP 定义的 PROGRAM 属性传递到转换器程序，并且转换器程序可以选择重设它。如果 CICS Web 支持处理中使用分析器程序，分析器程序还可以指定转换器程序，并可将数据传递给 COMMAREA 或用户令牌中的转换器程序。如果现有分析器程序使用此方法将数据传递给转换器程序，则注意，此功能不能由 URIMAP 定义复制。

转换器程序不可以为它在 COMMAREA 中接收的请求指定代码页转换设置。如果在 URIMAP 定义中指定转换器程序，而且 Web 客户机请求的头表明消息主体是文本，则 CICS 使用下列标准设置转换 COMMAREA 中提供的消息主体：

- 对于字符集，如果 Web 客户机请求具有 Content-Type 头，指定 CICS 支持的字符集，使用此字符集。If the Web client's request has no Content-Type header or the named character set is unsupported, the ISO-8859-1 character set is used.
- 对于主机代码页，CICS 将缺省代码页用于本地 CICS 区域，如 LOCALCCSID 系统初始化参数中所指定。

如果这些标准设置不合适，或者如果不需要代码页转换，则必须在处理路径中使用分析器程序以指定备用设置。

在 CICS 是 HTTP 客户机时无法调用转换器程序，当 CICS 用于 Web Service 处理时也无法调用转换器程序；只在 CICS 是 HTTP 服务器时才可调用它。

对 Web 出错程序的更改

当作为 HTTP 服务器的 CICS 的 CICS Web 支持过程中发生请求错误或异常终止，则用户可替换的 Web 出错程序向 Web 客户机提供错误响应。Web 错误程序接收到有关错误情况的信息，以及 CICS 计划发送给 Web 客户机的缺省 HTTP 响应（包括状态码和状态文本）。程序可定制响应或构建一个新的，并将它返回给 CICS 用于发送。

您可能已在早期 CICS 发行版中定制 Web 出错程序 DFHWBEP。CICS 现在使用其他状态码，并在大多数情况下使用一些现有状态码。如果您对 DFHWBEP 进行修改以定制与每个状态码关联的响应，则您应该意识到这一点。

注意，EXEC CICS WEB 和 DOCUMENT 应用程序编程接口不能从 DFHWBEP 获得。DFHWBEP 使用基于 COMMAREA 的接口，其中完整的 HTTP 响应作为数据缓存创建。

对统计信息的更改

为 URIMAP 资源定义对象生成新的全局统计信息和资源统计信息。全局统计信息显示涉及 URIMAP 定义的不同 CICS Web 支持操作的使用情况计数，如成功匹配和失败匹配尝试、重定向，或者动态或静态响应的传递。资源统计信息显示个别 URIMAP 定义的属性和相关使用情况计数。

这些统计信息由 PERFORM STATISTICS 和 EXTRACT STATISTICS 命令使用 URIMAP 关键字收集。DSECT 是 DFHWBRDS（用于资源统计信息）和 DFHWBGDS（用于全局统计信息）。DFHSTUP 和 DFH0STAT 包括这些统计信息的新报告。

对 CICS 实用程序的更改

统计信息实用程序 DFHSTUP

DFHSTUP 支持对第 53 页的『对统计信息的更改』中描述的统计信息的更改。

现在，您可在 DFHSTUP 的 SELECT TYPE 和 IGNORE TYPE 控制参数上对 URIMAP 资源类型进行编码。

对问题确定的更改

为了增强 CICS Web 支持，引入新消息和跟踪点，以及一个新的异常终止代码。在 HTTP 消息中 CICS 作为 Warning 头接收的任何信息收集并写入 TD 队列 CWBW。

HTTP 头中的警告消息

如果警告标题出现在 HTTP 消息中，则它通常包含打算供用户阅读的信息。如果 CICS Web 支持接收到带有 Warning 头的消息，则与此头关联的文本，以及发送方的 IP 地址写入瞬时数据队列 CWBW。（CWBW 间接导向 CSSL。）如果您接收过多警告头，您可除去 CWBW 瞬时数据队列，以不显示这些记录。

消息

引入新 DFHWBxxxx 消息的结果是增强 CICS Web 支持。所有新的和已更改的消息在《CICS 消息和代码》手册中有所描述。

异常终止代码

引入一个新的异常终止代码 AWBP。当作为 HTTP 服务器的 CICS 的支持 Web 的应用程序发送已分块消息时使用它，但是无法发送最终的空块来完成消息。

跟踪

新的 CICS 跟踪点添加到 WB 域跟踪中，以增强 CICS Web 支持。新跟踪点的范围是 WB 0419 到 WB 0C07。

要控制 CICS 跟踪信息的输出，以常规方式使用 CICS 跟踪控制。

安全性

新的 SPI 和 CEMT 命令的安全性

新的预定义 RACF 资源名称使用 SPI 和 CEMT 控制对下列资源的访问：

HOST
URIMAP

新的类别 1 事务

新的 CWXU 事务供 CICS 内部使用，而不应该从用户终端调用它们。出于安全目的，它是类别 1 事务。

新的全局用户出口

当 CICS 作为 HTTP 客户机时，新的全局用户出口 XWBOPEN（在 WEB OPEN 命令上）和 XWBSNDO（在 WEB SEND 命令上）使您能够将安全策略应用于为来自 CICS 的出站 HTTP 客户机请求指定的主机名和路径。第 81 页的『对全局用户出口的更改』描述这些新出口。

作为 HTTP 服务器的 CICS 的静态响应安全性

在不调用用户所写应用程序的情况下，通过设置提供响应的 URIMAP 定义，您可将 CICS 文档和 HFS 文件作为静态响应传递给来自 Web 客户机的请求。CICS 可以为 HTTP 连接实施 HTTP 基本认证，其中用户必须提供使用外部安全管理器注册的标识和密码。然而，根据用户标识使用不同访问控制的资源级别安全性不应用于作为静态响应传递的项。如果此项需要此级别的安全性，则需要将素材作为应用程序生成的响应传递。当使用应用程序生成的响应时，用户标识可应用于涉及用户所写应用程序处理的别名事务，所以您可授权访问或拒绝访问应用程序使用的特定资源和命令。

迁移

迁移现有 CICS Web 支持应用程序

CICS Transaction Server for z/OS V3R1 设计为对支持 Web 和不支持 Web 的应用程序的现有 CICS Web 支持体系结构都支持。EXEC CICS WEB API 命令更改设计为允

许发送和接收 HTTP 消息的现有支持 web 的应用程序在不更改的情况下工作，直到您选择迁移它们以利用现在可用的增强。如果您继续使用现有的 CICSWeb 支持应用程序，则注意这些迁移问题：

- 如果您使用 **CICS Web** 支持处理非 HTTP 请求，则在为这些请求定义端口的 **TCPIP SERVICE** 定义上指定新的 **USER** 协议。这还应用于使用非标准请求方法的 HTTP 请求，这些请求如果是在 HTTP 协议上接收，则现在被拒绝（先前，它们作为非 HTTP 接受和处理）。现在必须在 **USER** 协议下执行所有非 HTTP 请求的处理，以便它们不从属于 CICS 使用 HTTP 协议为请求执行的基本验收检查。将请求标志为非 HTTP 并在不更改的情况下传递给 **TCPIP SERVICE** 的分析器程序。CICS Web 支持工具用于处理请求，但是对于使用此协议发送和接收的消息，不执行验收检查。

注：因为对于每个端口只存在一个活动的 **TCPIP SERVICE** 定义，所以非 HTTP 请求不再使用与 HTTP 请求相同的端口。熟知端口号 80（用于 HTTP）和 443（用于 HTTPS）必须具有 HTTP 协议，因此无法接受非 HTTP 请求。Web 客户机必须在其请求的 URL 中指定任何已更改的端口。

- **检查使用 HTTP 协议的 TCPIP SERVICE 资源定义的设置。**
 1. **SOCKETCLOSE** 属性必须不再具有零设置 (**SOCKETCLOSE(0)**)。 **SOCKETCLOSE** 的零设置意味着 CICS 在接收到从 Web 客户机发出的数据之后立即关闭连接，除非还在等待其他数据。这意味着无法维护持续连接。 **SOCKETCLOSE** 的非零设置允许与 HTTP/1.1 客户机和 HTTP/1.0 客户机（其中的客户机支持此情况）的持续连接。
 2. 应该指定新的 **MAXDATALEN** 选项，以限制作为 HTTP 服务器的 CICS 可接收的最大数据长度。此设置可帮助您在涉及大量数据传输时防止 CICS 受到拒绝服务的攻击。
 3. 如果您使用 SSL，则对 **TCPIP SERVICE** 资源定义上可用的安全选项进行一些更改。
- **代码页转换表 (DFHCNV)** 对于 **CICS Web** 支持不再是必需的。然而，如果您要继续使用早期 CICS 发行版中编码的分析器程序来引用 **DFHCNV**，则您必须继续提供代码页转换表中的条目，或更改分析器程序。更改分析器程序涉及对两个新的输出参数进行编码，以指定客户机和服务器代码页，替换指定 **DFHCNV** 条目名称的输出参数。如果您执行此操作，您无须迁移 **DFHCNV** 条目。
- 如果您已修改用户可替换 **Web** 出错程序 **DFHWBEP** 来定制错误情况下提供的 **HTTP** 响应，则了解 **CICS** 现在使用其他状态码，并在大多数情况下使用一些现有状态码。
 1. 检查程序使用相应范围的输入参数标识应用定制响应的情况，而不是单单依赖状态码。错误代码、异常终止代码、消息号、响应和原因码，或程序名都可用于标识引发 **HTTP** 响应的情况。如果不进行这些检查，则您可能发现出于新的目的，CICS 使用状态码的情况，返回不适当的定制响应。
 2. 检查您的程序是否包含逻辑，通过带有程序未知的状态码的任何未更改 **HTTP** 响应传递。
- **CICS Transaction Server for z/OS V3R1** 中仍然支持 **DFHWBCLI** 接口。要获得增强的功能，您可迁移使用 **DFHWBCLI** 接口的 **HTTP** 客户机应用程序，以为客户机请求使用 **CICS WEB API** 命令（使用 **SESSTOKEN** 选项）。注意，一个重要的差异在于，在 **CICS WEB API** 中，代理服务器的使用由 **WEB OPEN** 命令

(XWBOPEN) 上的用户出口指定, 而代理服务器的 URL 由此用户出口提供。第 61 页的第 3 章, 『支持 CICS 应用程序发出的 HTTP 客户机请求』 描述了现在如何发出 HTTP 客户机请求。

迁移到新的 CICS Web 支持功能

CICS Transaction Server for z/OS V3R1 中的 CICS Web 支持具有很多增强, 提供对先前由用户可替代程序所处理功能的自动和管理员控制。特别是, 建议您研究 CICS Web 支持体系结构下列元素的迁移可能性:

- 通常, 您应该能够使用 URIMAP 资源定义替换现有分析器程序的请求处理功能, 这可使用 CICS 系统编程命令更改和控制。URIMAP 定义可用于匹配请求的 URL 并将它们映射到应用程序, 并且指定转换器程序和别名事务。如果定制分析器程序提供其他功能, 您可继续使用它替换 URIMAP 定义, 或者您可将它与 URIMAP 定义结合。当迁移到使用 URIMAP 时:
 1. 您可每次为少量请求逐渐引入 URIMAP 资源定义。根据分析器程序执行的处理类型, 以及处理请求的应用程序类型, 您可选择是否继续在每个请求的处理路径中使用分析器程序。
 2. 您可能首选为 URIMAP 资源定义处理的请求选择和发布新的 URL, 而不是保留现有的 URL。当您准备不再为请求使用旧的处理路径时, 您可设置 URIMAP 定义, 将请求直接从旧的 URL 重定向到新的 URL。
 3. 确保 TCPIP SERVICE 定义上指定的分析器程序仍然包含基本错误处理过程, 即使它不再涉及任何请求的处理路径。分析器程序必须仍然存在, 而且如果 URIMAP 匹配失败, 则它接收请求。
- 对于不使用 EXEC CICS WEB API 命令, 但是在 COMMAREA 中生成 HTTP 响应的应用程序, CICS Web 支持不能帮助正确组装消息结构, 或者对响应执行全方面的检查。要利用所有可用的 CICS Web 支持工具, 建议您计划将这些应用程序转换为使用 WEB API 命令的支持 web 的应用程序。
- URIMAP 资源定义可用于将 CICS 文档或 HFS 文件作为静态响应传递, 或者在不涉及用户所编写应用程序的情况下, 传递重定向响应。您应该考虑为不涉及动态处理的简单响应使用此机制, 而不是应用程序。
- 检查以更有效的方法操作代码页转换。对应用程序进行一些小更改, 您可利用新的 CICS Web 支持工具完成:
 - 避免为 CICS Web 支持设置和使用代码页转换表 (DFHCNV)。
 - 允许 CICS 标识和使用 Web 客户机的字符集, 用于代码页转换, 而不是您自己指定。
 - 使用本地系统缺省值 (LOCALCCSID 系统初始化参数) 标识应用程序的代码页, 而不是您自己指定。
 - 转换到 UTF-8 和 UTF-16 字符集, 以及从 UTF-8 和 UTF-16 字符集转换到其他格式。

CICSplex SM 支持

由 Web 浏览器访问的 CPSM Web 用户接口 (WUI)。该接口接受 HTTP/1.0 和 HTTP/1.1 级别请求, 但只提供 HTTP/1.0 响应。用于访问 CPSM WUI 的 Web 浏览器应该能够接受和理解 HTTP/1.0 响应。

对 CICSplex SM 最终用户界面视图的更改

已对下列 EUI 视图进行更改:

- 『TCPDEF 视图』
- 『DOCDEF 视图』
- 『DOCTEMP 视图』

TCPDEF 视图

下列属性已添加到 TCPDEF 视图:

PROTOCOL CVDA (USER)

协议

MAXDATALEN

定义可以在入站请求或对出站请求的响应上接收的数据的最大长度。

此视图中的 **PRIVACY** 属性在 CICS Transaction Server 3.1 中不再有效, 并将被忽略。

DOCTEMP 视图

下列属性已添加到 DOCTEMP 视图:

HFSFILE

UNIX 系统服务 HFS 文件用作文档模板。

TEMPLATETYPE CVDA (HFS)

文档模板类型

DOCDEF 视图

已将新的属性添加到 DOCDEF 视图:

HFSFILE

UNIX 系统服务 HFS 文件用作文档模板。

属性 **TEMPLATETYPE** 已除去。

对 CICSplex SM 应用程序编程接口的更改

新资源表

引入下列资源表:

- 『URIMAP 资源表』
- 第 137 页的 『URIMPDEF 资源表』
- 第 138 页的 『URIMPGBL 资源表』
- 第 139 页的 『HOST 资源表』

URIMAP 资源表

URIMAP 资源表具有下列 SPI 属性:

URIMAP

URIMAP 名称

ENABLESTATUS

显示表明是以 ENABLED 还是 DISABLED 状态安装映射的 CVDA 值。

USAGE

显示表明映射是否分别用于作为 HTTP 服务器的 CICS、作为 HTTP 客户机的 CICS 或 Web Service 的服务器、客户机或管道的 CVDA 值。

SCHEME

显示表明 URI 映射适用的 URI 方案组件没有 SSL (HTTP) 还是有 SSL (HTTPS) 的 CVDA 值。

HOST 显示映射适用的 URI 的 116 个字符的主机组件。

PATH 显示映射适用的 URI 的 255 个字符的路径组件。

MEDIATYPE

显示 CICS 提供给入站 HTTP 请求的静态响应的 56 个字符的数据内容。

CHARACTERSET

显示 CICS 转换成发送给入站 HTTP 请求的静态响应的字符集的名称的 40 个字符的名称。

HOSTCODEPAGE

显示在其中构成编码的静态响应的文本文档的 IBM 代码页 (EBCDIC) 的 10 个字符的名称。

TEMPLATENAME

显示构成静态响应的 CICS 文档模板的 48 个字符的名称。

HFSFILE

显示 UNIX 系统服务 HFS 文件的标准名称，该文件构成发送到入站 HTTP 请求的静态响应。

TCIPSERVICE

显示定义 URI 映射适用的入站端口的 TCIPSERVICE 资源定义的 8 个字符的名称。

ANALYZER

显示 CVDA 值 (YES 或 NO)，它显示在处理入站 HTTP 请求中是否要使用分析器程序。

CONVERTER

显示处理请求内容的转换器程序的 8 个字符的名称。

TRANSACTION

显示用于运行组成响应的用户应用程序的别名事务的 4 个字符的名称。

PROGRAM

显示指定处理请求的服务的用户应用程序或 XML 服务的 8 个字符的名称。

PIPELINE

显示“管道”的 8 个字符的名称，该“管道”指定处理请求的服务。

WEBSERVICE

显示 Web Service 的 32 个字符的名称，该 Web Service 指定处理请求的服务。

USERID

显示用于连接别名事务的 8 个字符的用户标识。

CERTIFICATE

显示 URI 用于出站 HTTPS 请求时，显示要用作 SSL 客户机证书的 32 个字符的证书标号。

CIPHERS

#

显示多达 56 个十六进制数字的字符串，该字符串被解释为多达 28 个 2 位的密码套件代码列表。

NUMCIPHERS

SSL 密码套件代码数

LOCATION

显示应该重定向入站 HTTP 请求的 255 位字符的 URI。

REDIRECTTYPE

显示此类重定向的 CVDA 值 (NONE、TEMPORARY 或 PERMANENT)。

MAPREFCOUNT

URI 映射引用计数

MATCHDISABLD

禁用 URI 映射主机或路径

MATCHREDIREC

URI 映射主机或路径重定向

URIMPDEF 资源表

URIMPDEF 资源表具有下列 RDO 属性:

URIMAP

URI 映射定义名称

STATUS

显示表明是以 ENABLED 还是 DISABLED 状态安装映射的 CVDA 值。

USAGE

显示表明映射是否分别用于作为 HTTP 服务器的 CICS、作为 HTTP 客户机的 CICS 或 Web Service 的服务器、客户机或管道的 CVDA 值。

SCHEME

显示表明 URI 映射适用的 URI 方案组件没有 SSL (HTTP) 还是有 SSL (HTTPS) 的 CVDA 值。

HOST 显示映射适用的 URI 的 116 个字符的主机组件。

PATH 显示映射适用的 URI 的 255 个字符的路径组件。

MEDIATYPE

显示 CICS 提供给入站 HTTP 请求的静态响应的 40 个字符的数据内容。

CHARACTERSET

显示 CICS 转换成发送给入站 HTTP 请求的静态响应的字符集的名称的 40 个字符的名称。

HOSTCODEPAGE

显示在其中构成编码的静态响应的文本文档的 IBM 代码页 (EBCDIC) 的 10 个字符的名称。

TEMPLATENAME

显示构成静态响应的 CICS 文档模板的 48 个字符的名称。

HFSFILE

显示 UNIX 系统服务 HFS 文件的 255 个字符的标准名称，该文件构成发送到入站 HTTP 请求的静态响应。

TCPIPSERVICE

显示定义 URI 映射适用的入站端口的 TCPIPSERVICE 资源定义的 8 个字符的名称。

ANALYZER

显示 CVDA 值 (YES 或 NO)，它显示在处理入站 HTTP 请求中是否要使用分析器程序。

CONVERTER

显示处理请求内容的转换器程序的 8 个字符的名称。

TRANSACTION

显示用于运行组成响应的用户应用程序的别名事务的 4 个字符的名称。

PROGRAM

显示指定处理请求的服务的用户应用程序或 XML 服务的 8 个字符的名称。

PIPELINE

显示“管道”的 8 个字符的名称，该“管道”指定处理请求的服务。

WEBSERVICE

显示 Web Service 的 32 个字符的名称，该 Web Service 指定处理请求的服务。

USERID

显示用于连接别名事务的 8 个字符的用户标识。

CERTIFICATE

显示 URI 用于出站 HTTPS 请求时，显示要用作 SSL 客户机证书的 32 个字符的证书标号。

CIPHERS

显示多达 56 个十六进制数字的字符串，该字符串被解释为多达 28 个 2 位的密码套件代码列表。

#

LOCATION

显示应该重定向入站 HTTP 请求的 255 位字符的 URI。

REDIRECTTYPE

显示此类重定向的 CVDA 值 (NONE、TEMPORARY 或 PERMANENT)。

URIMPGBL 资源表

URIMPGBL 资源表具有下列属性:

MAPREFCOUNT

URI 映射引用计数

MATCHDISABLD

禁用 URI 映射主机或路径

NOMATCHCOUNT

URI 映射主机或路径不匹配计数

MATCHCOUNT

URI 映射主机或路径匹配计数

MATCHREDIREC

URI 映射主机或路径重定向

MATCHANALYZE

URI 映射主机或路径匹配分析器

STATICCONTENT

URI 映射静态内容

DYNAMCONTENT

URI 映射动态动态

PIPELINEREQS

URI 映射管道请求

SCHEMEHTTP

URI 映射规则（HTTP）请求

SCHEMEHTTPS

URI 映射规则（HTTPS）请求

HOSTDISABLED

主机禁用的计数

HOST 资源表

HOST 资源表具有下列属性:

HOSTNAME

主机名

ENABLESTATUS

显示表明是以 ENABLED 还是 DISABLED 状态安装主机的 CVDA 值。

对资源表的更改

已对下列资源表进行更改:

- 『TCPDEF 资源表』
- 第 140 页的 『DOCTEMP 资源表』
- 第 140 页的 『DOCDEF 资源表』
- 第 140 页的 『TASK 资源表』

TCPDEF 资源表

TCPDEF 资源表具有下列更多属性:

PROTOCOL CVDA (USER)

协议

MAXDATALEN

定义可接收或发送的最大数据长度。

#

此资源表中的 **PRIVACY** 属性在 CICS Transaction Server 3.1（或更改版本）中不再有效，并将被忽略。

DOCTEMP 资源表

DOCTEMP 资源表具有下列更多属性:

HFSFILE

UNIX 系统服务分层文件系统模板文件

TEMPLATETYPE

文档模板类型

DOCDEF 资源表

DOCDEF 资源表包含下列属性:

HFSFILE

分层文件系统模板文件

TASK 资源表

TASK 资源表包含下列属性:

TMRWBRDL

显示从存储库读取的数据的数据长度

TMRWBWDL

显示写入存储库的数据的数据长度

对 CICSplex SM Web 用户界面的更改

新的 WUI 视图

已引入下列 WUI 视图:

- 『URI 映射定义视图』
- 第 141 页的『URI 映射视图』
- 第 141 页的『URI 映射全局视图』
- 第 141 页的『主机视图』

URI 映射定义视图

已引入定义视图集，称为 **URI 映射定义**，与新的 URIMPDEF 资源表相关联。此表格视图的视图名称为 EYUSTARTURIMPDEF.TABULAR，而现有 EYUSTARTADMRES 菜单已扩展为包含它。

要打开 **URI 映射定义** 视图，执行下列操作:

1. 从主菜单单击**管理视图**
2. 单击**基本 CICS 资源管理视图**（或者，可选择单击**完全功能业务应用程序服务 (BAS) 管理视图**）
3. 单击**CICS资源定义**
4. 下滚并单击 **URI 映射定义**

显示 **URI 映射定义** 视图。此视图包括下列五个操作按钮:

- **创建**
- **更新**

- 除去
- 安装
- 添加到资源组

请参阅第 137 页的『URIMPDEF 资源表』中所列出 URIMPDEF 资源表的属性，以获得字段详细信息。

URI 映射视图

已引入新的视图，称为 **URI 映射**，与新的 URIMAP 资源表相关联。此表格视图的视图名称为 EYUSTARTURIMAP.TABULAR，而现有 EYUSTARTTCPIPS 菜单已扩展为包含它。

要打开 **URI 映射** 视图，执行下列操作：

1. 从主菜单单击 **CICS操作视图**
2. 下滚并单击 **TCP/IP 服务操作视图**
3. 单击 **URI 映射**

显示 **URI 映射** 视图。此视图包括下列四个操作按钮：

- 设置属性
- 启用
- 禁用
- 废弃

要获得 URIMAP 资源表的属性详细信息，请参阅第 135 页的『URIMAP 资源表』

URI 映射全局视图

已引入新的视图，称为 **URI 映射全局**，与新的 URIMPGBL 资源表相关联。此表格视图的视图名称为 EYUSTARTURIMPGBL.DETAILED，而现有 EYUSTARTTCPIPS 菜单已扩展为包含它。

要打开 **URI 映射全局** 视图，执行下列操作：

1. 从主菜单单击 **CICS操作视图**
2. 下滚并单击 **TCP/IP 服务操作视图**
3. 单击 **URI 映射全局**
4. 选择 **CICS 系统名称**，以打开 **URI 映射全局详细视图**

显示 **URI 映射全局** 视图。要获得 URIMPGBL 资源表的属性详细信息，请参阅第 138 页的『URIMPGBL 资源表』。

主机视图

已引入新的视图，称为 **主机**，与新的 HOST 资源表相关联。此表格视图的视图名称为 EYUSTARTHOST.TABULAR，而现有 EYUSTARTTCPIPS 菜单已扩展为包含它。

要打开 **主机** 视图，执行下列操作：

1. 从主菜单单击 **CICS操作视图**
2. 下滚并单击 **TCP/IP 服务操作视图**

3. 单击主机

显示主机视图。

要获得 HOST 资源表的属性详细信息，请参阅第 139 页的『HOST 资源表』。

对 WUI 视图的更改

已对下列视图进行更改：

- 『TCP/IP 服务定义视图』
- 『文档模板定义视图』
- 『文档模板视图』
- 『TCP/IP 使用情况视图』

TCP/IP 服务定义视图

下列属性已添加到 TCPDEF (EYUSTARTTCPDEF) 视图：

PROTOCOL CVDA (USER)

协议

MAXDATALEN

定义可以在入站请求或对出站请求的响应上接收的数据的最大长度。

此视图中的 **PRIVACY** 属性在 CICS Transaction Server 3.1 中不再有效，并将被忽略。

文档模板定义视图

下列属性已添加到 DOCDEF (EYUSTARTDOCDEF.DETAILED) 视图：

HFSFILE

UNIX 系统服务 HFS 文件用作文档模板。

文档模板视图

下列属性已添加到 DOCTEMP (EYUSTARTDOCTEMP.DETAILED) 视图：

HFSFILE

UNIX 系统服务 HFS 文件用作文档模板。

TEMPLATETYPE CVDA (HFS)

文档模板类型

TCP/IP 使用情况视图

在任务操作视图中，下列属性已添加到活动任务视图集 (EYUSTARTTASK.DETAILED) 中的 **TCP/IP 使用情况视图**：

TMRWBRDL

显示从存储库读取的数据的数据长度

TMRWBWDL

显示写入存储库的数据的数据长度

第 6 章 对因特网安全性的改进

对安全性增强的概述。

对安全性的改进涉及很多方面并将其概述如下：

- 支持 TLS 1.0 协议和 AES 密码套件
- 通过增加的同时发生的 SSL 连接数的可伸缩性
- 支持证书撤回列表
- 跨综合系统 (sysplex) 中 CICS 区域的 SSL 高速缓存
- 指定最低和最高加密协商级别

注：CICS 支持本发行版中的 SSL 和 TLS 安全协议。为了清楚说明，术语 SSL 用于引用文档中的两个协议，除了需要有关两个协议中任何一个的特定点。

对因特网安全性改进的优点

改进安全性可以得到一系列好处。

CICS 现在支持传输层安全性 (TLS) 1.0 协议以及 SSL 3.0，这允许您使用新的提供 128 位和 256 位加密的 AES 密码套件。

对 SSL 的性能进行了一些改进，以支持新的功能，例如 Web Service。已增加可以在系统中同时使用的同步 SSL 连接数，以实现更大的吞吐量。

更为灵活地控制客户机和服务器之间的加密协商。您可以为与特殊用户协商指定 CICS 中的最低加密级别和最高加密级别。

与客户机协商时，CICS 现在可以根据证书撤回列表 (CRL) 检查所有证书。任何使用已撤回证书的连接都立即关闭。

您可以指定是否要通过使用 SSL 高速缓存跨越综合系统 (sysplex) 共享会话标识。如果客户机先前已与 CICS 进行协商，CICS 执行部分 SSL 握手。跨很多 CICS 区域共享高速缓存，这可改进 SSL 协商和连接吞吐量的性能。

安全术语

下列术语用于描述对 CICS 中因特网安全性的增强。

传输层安全性 (TLS)

用于在因特网上提供安全通信的安全协议。规范在 RFC2246 中描述。

密码套件

加密算法、加密密钥长度，以及在 SSL 握手期间协商的 MAC 算法的组合。

消息认证代码 (MAC)

密码保护与 SSL 连接上发送的每个消息关联的散列代码。

MAC 算法

计算消息认证代码的密码算法。SSL 使用 MD5 和 SHA 算法。

SSL 高速缓存

SSL 使用的高速缓存，用于存储有关它的与客户机进行加密协商的会话标识信息。如果客户机先前已使用 SSL 安全地连接 CICS，则仅执行部分握手以建立 SSL 连接。

证书撤回列表

已撤回证书的列表，它由称为证书权限的独立主体提供。如果证书已撤回，则将它添加到证书撤回列表。当客户机和服务器尝试相互认证时，在 SSL 握手协商期间可以交叉引用这些列表。

SP 方式

属于套接字 pthread 所拥有任务的 TCB 方式。

SSL 池

包含和管理 CICS 区域中的 S8 TSB 的池。

SSL 握手

建立连接时，发生在客户机和服务器之间的信息交换。握手涉及协商要使用哪些加密算法和相互认证。

需求

硬件

zSeries® 加密硬件必须完全得益于对 SSL 加密的性能改进，其依赖于 z/OS 集合加密工具 (ICSF)。

软件

z/OS 的系统 SSL 安全级别 3 功能必须使用以 128 位或更高位数加密的密码套件。

传输层安全性协议

传输层安全性 (TLS) 1.0 协议是标准安全协议，它提供因特网上的安全通信。

CICS 支持两种安全协议，这两种协议可用于提供因特网上的安全通信。第一个是安全套接字层 (SSL) 3.0 协议。第二个是传输层安全性 (TLS) 1.0 协议，它是最新的业界标准 SSL 协议，而且基于 SSL 3.0。TLS 1.0 规范则 RFC2246 中已制订，而且可在因特网上从 www.rfc-editor.org/rfcsearch.html 获得。需要加密的任何连接将自动使用 TLS 协议，除非客户机明确需要 SSL 3.0。

TLS 的主要目标是使安全套接字层更安全，并使协议规范更精确和完整。TLS 提供下列对 SSL 3.0 的增强：

•

消息认证的密钥散列法

TLS 使用消息认证代码的密钥散列法 (HMAC)，其确保在游历开放网络 (如因特网) 时不改变记录。SSL V3.0 还提供加密的消息认证，但是 HMAC 被认为比 SSL V3.0 使用的 (消息认证代码) MAC 功能更安全。

•

```
#          增强的伪随机功能（PRF）
#          PRF 用于生成密钥数据。在 TLS 中，使用 HMAC 定义 PRF。PRF 以保证
#          其安全性的方法使用两个散列算法。如果列出任何一种方法，只要不列出第
#          二个算法，数据都保持安全。
#
#          •
#
#          改进的已完成消息验证
#          TLS 1.0 和 SSL 3.0 提供已完成消息给两个认证交换消息未改变的端点。然
#          而，TLS 将此已完成消息基于 PRF 和 HMAC 值，这比 SSL V3.0 更安全。
#
#          •
#
#          一致的证书处理
#          与 SSL 3.0 不同，TLS 尝试指定必须在 TLS 实施之间交换的证书类型。
#
#          •
#
#          特定警报消息
#          TLS 提供更特定和其他警报，以表明任何一个会话端点检测到的问题。TLS
#          还制订何时应该发送某些警报。
#
```

对 SSL 性能的改进

对 S8 TCB 实施的更改可改进 CICS 中 SSL 连接的数目和性能。

CICS 使用开放式事务环境（OTE）管理 SSL 连接。每个 SSL 连接使用 S8 TCB，其现在作为 UNIX pthread 运行。还存在一个新的称为 SP 的 TCB 开放方式，它用于套接字 pthread 拥有的任务。所有 S8 TCB 都在单个封套中运行，其属于 SP TCB，还包含 SSL 高速缓存。这提供了很多优点，将存储节省在此限制下、允许 CICS 中比前发行版更多的同时发生的 SSL 连接。

S8 TCB 包含在 SSL 池中，此池由 CICS 分派器管理。从新的 SSL 池分配 S8 TCBS，但是仅对一个事务把 S8 TCB 锁定一段时间，在这段锁定的时间内事务需要执行 SSL 功能。SSL 协商完成后，将 TCB 释放回 SSL 池以供重新使用。MAXSSLTCBS 系统初始化参数指定 SSL 池中 S8 开放式 TCB 的最大数目。缺省值为 8，但您最多可以指定 1024 个。

您可使用分派器报告从 DFHOSTAT 和 DFHSTUP 监控 SSL 池和 S8 TCB 的性能。统计信息包括有关达到 S8 TCB 最大值的频率、分配 TCB 之前的延迟，以及 SSL 池中 TCB 的实际数量的信息。

使用证书撤回列表

您可以配置 CICS 以使用证书撤回列表（CRL）来检查 SSL 协商中使用的客户机证书的有效性。

要使用证书撤回列表，您必须安装和配置 LDAP 服务器。有关如何执行这些任务的详细信息可在 *z/OS VIR4.0 Security Server LDAP Server Admin and Use* 中找到。

可以从证书中心（例如，Verisign）获得证书撤回列表。它们保存在 CRL 资源库中，可以从万维网上获得这些资源库，并且可以下载并存储在 LDAP 服务器中。要放入 LDAP

服务器和更新证书撤回列表，使用 CICS 提供的事务 CCRL。您可以从终端或使用 START 命令运行 CCRL 事务。使用 START 命令调度常规更新。要在 LDAP 服务器中包含 CRL，执行这些步骤：

1. 配置 LDAP 服务器，以指定您要使用的认证中心。
2. 在 CRLSERVER 系统初始化参数中指定 LDAP 服务器的名称。
3. 运行 CCRL 事务。

当您运行 CCRL 事务时，CICS 从您指定的 URL 下载 CRL，并将它们存储在 CRLSERVER 系统初始化参数中指定的 LDAP 服务器中。

SSL 高速缓存

用于存储来自客户机和 CICS 之间的协商的会话标识的 SSL 高速缓存。

SSL 高速缓存允许 CICS 执行与先前已认证的客户机的部分握手。在本地 CICS 区域中，SSL 高速缓存是 S8 TCB 模块的一部分。您有一个选项，可以选择跨综合系统共享 SSL 高速缓存，如果它适合于您的 CICS 系统的话。如果您有多个接受同一个 IP 地址上的 SSL 连接的 CICS 套接字拥有的区域，则可以使用综合系统高速缓存。

如果您要在各区域之间共享高速缓存，则激活系统 SSL 启动的任务 GSKSRVR，并使用系统初始化参数 SSLCACHE。缺省为使用本区域高速缓存，但您可以通过指定选项 SYSPLES 更改它。CICS 将改为使用耦合工具中的 SSL 高速缓存存储会话标识。

定制加密协商

您可以选择在加密协商过程中使用的密码套件，以设置加密的最低级别和最高级别。

资源定义 TCPIPService、CORBASERVER 和 URIMAP 的 CIPHERS 属性指定可用于每个加密级别的密码套件。属性的缺省值是用于加密协商的 2 位数字密码列表。您有定制密码套件的此列表的选项，以包括您的加密级别的首选项顺序，CICS 应该在该加密级别上与客户机进行协商。您还可以选择从列表中除去密码套件。如果您要确保仅使用非常高的加密级别，那么这是特别有用的。您可以按照以下方法实施它：

1. 选择您要更改的资源定义。
2. CIPHERS 属性会显示缺省值。例如，如果系统初始化参数 ENCRYPTION=STRONG，则缺省值为 0504352F0A0903060102。
3. 编辑属性值以除去和重新排序密码套件。例如，您可以指定 352F0A0504。
4. 保存资源定义。

指定 352F0A0504 意味着 CICS 不会在低于 128 位的加密下使用此资源为连接进行协商。属性中的每个 2 位数字的代码，例如 35、2F、0A 等等指的是至少有 128 位加密的密码套件。CICS 将开始通过尝试使用 AES 密码套件 35 和 2F 进行协商，因为它们最先出现在密码套件的列表中。如果客户机没有此级别的加密，CICS 将关闭连接。

注意，您不能包括不在该级别的加密的缺省值中的密码套件。例如，如果您指定了 MEDIUM 级别的加密，则不能将 AES 密码套件 35 和 2F 添加到 CIPHERS 属性。

对 CICS 外部的更改

对系统初始化参数的更改

有一些新的和经更改的系统初始化参数用来改进因特网安全性。经更改的参数为:

ENCRYPTION={STRONG|WEAK|MEDIUM}

指定 CICS 用于保护 TCP/IP 连接的密码套件。要与前发行版兼容, 接受 ENCRYPTION=NORMAL 为等于 ENCRYPTION=MEDIUM。

STRONG

指定 CICS 仅应该使用下列密码套件:

密码套件	加密算法	密钥长度	MAC 算法
01	未加密		MD5
02	未加密		SHA
03	RC4	40 位	MD5
04	RC4	128 位	MD5
05	RC4	128 位	SHA
06	RC2	40 位	MD5
09	DES	56 位	SHA
0A	三重 DES	168 位	SHA
2F	AES	128 位	SHA
35	AES	256 位	SHA

此表中使用的术语是:

MD5 消息摘要算法

SHA 安全散列算法

RC2、RC4

Rivest 加密

DES 数据加密标准

三重 DES

应用三次的 DES

AES 高级加密标准

WEAK

指定 CICS 仅应该使用下列密码套件:

密码套件	加密算法	密钥长度	MAC 算法
01	未加密		MD5
02	未加密		SHA
03	RC4	40 位	MD5
06	RC2	40 位	MD5

密码套件	加密算法	密钥长度	MAC 算法
此表中使用的术语是:			
MD5	消息摘要算法		
SHA	安全散列算法		
RC2、RC4	Rivest 加密		

MEDIUM

指定 CICS 仅应该使用下列密码套件:

密码套件	加密算法	密钥长度	MAC 算法
01	未加密		MD5
02	未加密		SHA
03	RC4	40 位	MD5
06	RC2	40 位	MD5
09	DES	56 位	SHA
此表中使用的术语是:			
MD5	消息摘要算法		
SHA	安全散列算法		
RC2、RC4	Rivest 加密		
DES	数据加密标准		

不推荐参数 SSLTCBS。改为使用下列新的参数:

MAXSSLTCBS={8|number}

指定可以在 SSL 池中运行的 S8 TSB 的最大数。缺省值为 8，但您最多可以指定 1024 个 TCB。

新的系统初始化参数为:

```
# CRLSERVER=servername:port
```

```
# 指定包含证书撤回列表 (CRL) 的 LDAP 服务器名称和端口号。指定此参数意味着
# CICS 在撤销状态的 SSL 协商期间检查每个客户机证书。如果撤销证书，CICS 立
# 即关闭连接。
```

SSLCACHE={CICS|SYSPLEX}

指定 SSL 是要使用会话标识的本地还是综合系统高速缓存。当多个 CICS 套接字拥有的区域接受同一个 IP 地址上的 SSL 连接时，才允许综合系统高速缓存。

对资源定义的更改

TCPIPSERVICE 和 CORBASERVER 定义

已将新的属性 CIPHERS 添加到 TCPIPSERVICE 和 CORBASERVER 资源定义。它也可在新的 URIMAP 资源定义中指定。请参阅第 107 页的『URIMAP 资源定义』以获取更多信息。仅当为资源建立的套接字连接使用 SSL 或 TLS 安全协议时，才使用密码套

件代码的 CIPHERS 列表。对于 TCPIP SERVICE 定义，为入站套接字连接使用 CIPHERS 列表。对于 CORBASERVER 定义，为出站套接字连接使用 CIPHERS 列表。

CIPHERS=value

该 value 指定可多达 56 个十六进制数字的字符串，它被解释为可多达 28 个 2 位密码套件代码的列表。可接受的代码列表由 ENCRYPTION 系统初始化参数决定。

- 对于 ENCRYPTION=WEAK，缺省值为 03060102
- 对于 ENCRYPTION=MEDIUM，缺省值为 0903060102
- 对于 ENCRYPTION=STRONG，缺省值为 0504352F0A0903060102

您可以重新排序密码，或者从缺省列表中除去它们。然而，您无法为指定的加密级别添加不在缺省列表中的密码。ENCRYPTION 系统初始化参数确定每个加密级别允许的密码套件代码。

TCPIP SERVICE 资源定义的 PRIVACY 属性由 CIPHERS 属性值确定。当 CIPHERS 属性的缺省值是完整的密码套件列表时，除去一些密码将更改 PRIVACY 属性。

- 如果您除去密码套件 01 和 02 以指定 CICS 仅与已加密的客户机协商，则 PRIVACY 属性值更改为 REQUIRED。
- 如果您除去了密码套件 01 和 02 之外的所有密码套件，以指定 CICS 仅与还未加密的客户机协商，则 PRIVACY 属性值更改为 NOTSUPPORTED。
- 如果您指定任何其他的密码套件组合，包括缺省值，则 PRIVACY 属性值为 SUPPORTED。

类似的约束适用于 CORBASERVER 资源定义的 OUTPRIVACY 属性。

对系统编程接口的更改

EXEC CICS 命令 INQUIRE TCPIP SERVICE、INQUIRE CORBASERVER 和新的命令 INQUIRE URIMAP 现在包括两个安全性选项。

CIPHERS(char56)

```
# 返回资源定义 TCPIP SERVICE、CORBASERVER 和 URIMAP 的属性 CIPHERS 中
# 指定的密码套件的列表。使用密码套件的此列表协商 SSL 连接。例如，如果您在使
# 用弱加密，则缺省值将是 03060102。
```

NUMCIPHERS(halfword)

返回用来协商加密级别的密码套件的数目。

```
# EXEC CICS 命令 INQUIRE TCPIP 还有两个安全性选项。
```

CRLSERVER(char255)

返回在 CRLSERVER 系统初始化参数中指定的 LDAP 服务器的名称。LDAP 服务器用于存储证书撤回列表。

SSLCACHE(cvda)

返回 CVDA 值，该值表明 SSL 正在使用哪个高速缓存存储会话标识。CVDA 值为：

CICS

SSL 在使用的 CICS 区域的本地 SSL 高速缓存

SYSPLEX

SSL 在使用的耦合工具中的 SSL 高速缓存。

对 INQUIRE DISPATCHER 和 SET DISPATCHER 命令进行了一些更改，以处理 SSL TCB 池。添加了以下两个选项：

MAXSSLTCBS(*fullword*)

返回 SSL 池中允许的最多 S8 TCB 数目，正如 MAXSSLTCBS 系统初始化参数中所指定的数目。

ACTSSLTCBS(*fullword*)

返回 SSL 池中 S8 TCB 的实际数目。

对 CICS 提供的事务的更改

CCRL 事务

使用 CCRL 事务创建和更新存储在 LDAP 服务器中的证书撤回列表 (CRL)。如果您在 CICS 区域中实施 SSL，并且希望在 SSL 握手期间对撤销证书检查每个连接，则您只需使用 CCRL。

CCRL 事务指定万维网上 CRL 存储库的位置。CICS 从指定 URL 的 CRL 存储库下载列表，并将它存储在 LDAP 服务器中。如果您需要访问多个 CRL 存储库，则您可指定多个 URL。

在您运行 CCRL 事务前，您必须在 CICS 中设置下列内容：

- 设置并配置为存储证书撤回列表的 LDAP 服务器。。
- CRLSERVER 系统初始化参数已通过存储证书撤回列表的 LDAP 服务器的名称来定义。

您可以从终端或从 START 命令运行 CCRL 事务。如果您要调度有规律的更新，使用 START 命令选项。

要从终端运行事务，确保您的终端接受大小写混合字符。输入以下命令：CCRL *url-list*，其中 *url-list* 是以空格定界的 URL 列表，包含您要下载的证书撤回列表。然后，提示您提供 LDAP 服务器的管理员专有名称和密码。这允许 CICS 使用下载的 CRL 更新 LDAP 服务器。

要从 START 命令运行事务，使用以下语法：

```
EXEC CICS START TRANSID(CCRL)
  FROM('admin://adminDN:adminPW url-list')
  LENGTH(url-list-length)
  [INTERVAL(hhmmss)|TIME(hhmmss)]
```

其中 *adminDN:adminPW* 是 LDAP 服务器的专有名称和密码，*url-list* 是包含您要下载的证书撤回列表的 URL 列表（以空格分隔），*url-list-length* 是包含 LDAP admin 专有名称和密码的 URL 列表的长度，而 *hhmmss* 是调度运行 CCRL 事务的时间间隔或到期时间。

例如，您可指定：

```
EXEC CICS START TRANSID(CCRL)
  FROM('admin://cn=ldapadmin:cics31ldap http://cr1.CertificateAuthority.com/CRLList1.cr1
  http://cr1.CertificateAuthority.com/CRLList2.cr1')
  LENGTH(132) INTERVAL(960000)
```

这将调度 CCRL 事务在 96 小时后运行，并从两个指定的 URL 下载证书撤回列表。

对 CEMT 的更改

对下列命令的更改:

- INQUIRE TCPIPSERVICE
- INQUIRE CORBASERVER
- INQUIRE TCPIP
- INQUIRE DISPATCHER
- SET DISPATCHER

CICS 提供的事务提供有关这些已更改命令的信息。

对全局用户出口的更改

全局用户出口 TCB 指示器列表现在包括一条 SP TCB 方式的条目。要获取完整的列表, 请参阅 Customization Guide。

对监控的更改

已添加新的监控字段, 以监控 MAXSSLTCBS 延迟时间和更改方式延迟时间。

DFHTASK

247 (TYPE-S, 'DSCHMDLY', 8 BYTES)

在用户任务或代表用户任务发出 CICS 分派器更改 TCB 方式请求后, 用户任务等待重新分派所用去的时间。例如, 将 TCB 方式请求从 CICS L8 或 S8 方式 TCB 更改回为 CICS QR 方式 TCB 可能必须等待 QR TCB, 这是因为 QR TCB 上当前在分派另一个任务。

注: 此字段是任务暂挂时间字段 SUSPTIME (014) 的一个组件。

281 (TYPE-S, 'MAXSTDLY', 8 BYTES)

由于 CICS 系统已达到系统初始化参数 MAXSSLTCBS 设置的极限, 用户任务等待获得 CICS TCB (S8 方式) 所用去的时间。S8 方式开放的 TCB 只能由 SSL pthread 使用。

注: 此字段是任务暂挂时间字段 SUSPTIME (014) 的一个组件。

对统计信息的更改

对统计信息的更改包括报告新的 SP TCB 方式和新的 SSL TCB 池。还更改了统计信息实用程序 DFHSTUP 和统计信息样本程序 DFHOSTAT 和 DFJOSTXR, 这将报告其他的 TCB 方式和池。TCPIP 全局统计信息还报告 SSL 高速缓存在本地发生, 还是跨综合系统 (sysplex) 发生。

对 CICS 实用程序的更改

对统计信息实用程序 DFHSTUP 进行了一些更改, 以报告其他 TCB 方式和 TCB 池。

对问题确定的更改

消息 DFHSO0123 有一个新的插入，CICS 在加密协商期间识别已撤回的证书时使用它。CICS 立即关闭连接并发出消息。

安全性

通过支持新的 AES 密码和更长的加密密码长度，增强了安全性。您还可以通过编辑 SSL 连接中使用的密码套件的缺省值，约束客户机和 CICS 之间协商的最低加密级别。

ENCRYPTION 系统初始化参数缺省值现在为 STRONG，提高了使用 SSL 时 CICS 中的安全性级别。

CCRL 事务的安全性类别是类别 2。

迁移

迁移现有函数

ENCRYPTION 系统初始化参数的缺省设置已更改为 STRONG。如果您未在 z/OS 上安装高加密密码（安全级别 3），那么您需要将 ENCRYPTION 系统初始化参数的缺省设置降级。在前发行版中已用作缺省值的 NORMAL 设置在 CICS 的此发行版中已更改为 MEDIUM。对于迁移目的，接受 NORMAL 作为 MEDIUM 的备用值。

SSLTCBS 系统初始化参数现在已陈旧，已被 MAXSSLTCBS 替换。MAXSSLTCBS 控制允许在 SSL 的开放式事务环境（OTE）TCB 池中并发运行的最大 S8 TCB 数。

迁移到新的函数

您可以利用 CIPHERS 属性以更好地控制 CICS 和客户机之间的加密协商过程。您可以使用 CRLSERVER 和 SSLCACHE 系统初始化参数，验证 SSL 握手中的证书并改进通过跨越一些 CICS 区域共享 SSL 高速缓存的握手的性能。

共存

CICS 继续支持 SSL v3.0，以及新的 TLS 1.0 协议。

CICSplex SM 支持

对 CICSplex SM 接口和 API 进行了很多更改，以与对 SSL 增强的新支持匹配。

对 CICSplex SM 最终用户界面视图的更改

CICS Transaction Server for z/OS, V3R1 中不存在新的最终用户界面视图。然而，已对下列视图进行更改，匹配对 SSL 的新支持：

- 第 153 页的『EJCODEF 视图』
- 第 153 页的『TCPDEF 视图』

- 『EJCOSE 视图』
- 『TCPIPS 视图』

EJCODEF 视图

下列属性已添加到 EJCODEF 视图:

CIPHERS

SSL 密码套件代码

此视图中的 **OUTPRIVACY** 属性在 CICS Transaction Server 3.1 中不再有效, 而且尽管它可以看到, 但是将被忽略。

TCPDEF 视图

下列属性已添加到 TCPDEF 视图:

CIPHERS

SSL 密码套件代码

此视图中的 **PRIVACY** 属性在 CICS Transaction Server 3.1 中不再有效, 而且尽管它可以看到, 但是将被忽略。

EJCOSE 视图

下列属性已添加到 EJCOSE 视图:

CIPHERS

SSL 密码套件代码

NUMCIPHERS

用来协商加密级别的密码套件的数目。

TCPIPS 视图

下列属性已添加到 TCPIPS 视图:

CIPHERS

SSL 密码套件代码

NUMCIPHERS

用来协商加密级别的密码套件的数目。

对 CICSplex SM 应用程序编程接口的更改

已对下列资源表进行更改:

- 第 154 页的 『EJCODEF 资源表』
- 第 154 页的 『TCPDEF 资源表』
- 第 154 页的 『EJCOSE 资源表』
- 第 154 页的 『TCPIPS 资源表』
- 第 154 页的 『TCPIPGBL 资源表』
- 第 154 页的 『TASK 资源表』
- 第 154 页的 『CICSRGN 资源表』

EJCODEF 资源表

新属性已添加到 EJCODEF 资源表:

CIPHERS

显示 SSL 密码套件代码的 56 位字符列表

TCPDEF 资源表

新属性已添加到 TCPDEF 资源表:

CIPHERS

显示 SSL 密码套件代码的 56 位字符列表

EJCOSE 资源表

新属性已添加到 EJCOSE 资源表:

CIPHERS

显示 SSL 密码套件代码的 56 位字符列表

NUMCIPHERS

用来协商加密级别的密码套件的数目。

TCPIPS 资源表

新属性已添加到 TCPIPS 资源表:

CIPHERS

显示 SSL 密码套件代码的 56 位字符列表

NUMCIPHERS

用来协商加密级别的密码套件的数目。

TCPIGBL 资源表

新属性已添加到 TCPIGBL 资源表:

CRLSERVER

显示包含证书撤回列表 (CRL) 的 569 位字符 LDAP 服务器名称

SSLCACHE

SSL 高速缓存类型 - 本地或综合系统 (sysplex)

TASK 资源表

新属性已添加到 TASK 资源表:

TMRSTDLY

显示发出分派器更改 TCB 方式请求后任务等待重新分派的 8 位字节时间。

TMRCMDLY

显示由于系统达到 MAXSSLTCBS 设置的显示, 任务等待获取 CICS SSL TCB 的 8 位字节时间。

CICSRGN 资源表

CICSRGN 资源表包括下列 SPI 属性:

MAXSSLTCBS

显示 SSL OTE 池中 TCB 的当前最大数目。

ACTSSLTCBS

显示当前分配的 SSL 池 TCB 数。

对 CICSplex SM Web 用户界面的更改

已对下列视图进行更改:

- 『CorbaServer 定义视图』
- 『TCP/IP 服务定义视图』
- 『CorbaServer 视图』
- 『TCP/IP 服务视图』
- 第 156 页的 『TCP/IP 全局状态视图』
- 第 156 页的 『时钟和计时视图』
- 第 156 页的 『CICS 区域视图』

CorbaServer 定义视图

下列属性已添加到 EJCODEF (EYUSTARTEJCODEF) 视图:

CIPHERS

SSL 密码套件代码

此视图中的 **OUTPRIVACY** 属性在 CICS TS 3.1 中不再有效, 尽管它可查看, 但是将忽略它。

TCP/IP 服务定义视图

下列属性已添加到 TCPDEF (EYSTARTTCPDEF) 视图:

CIPHERS

SSL 密码套件代码

此视图中的 **PRIVACY** 属性在 CICS TS 3.1 中不再有效, 尽管它可查看, 但是将忽略它。

CorbaServer 视图

下列属性已添加到 EJCOSE (EYSTARTJCOSE.DETAILED) 视图:

CIPHERS

SSL 密码套件代码

NUMCIPHERS

用来协商加密级别的密码套件的数目。

TCP/IP 服务视图

下列属性已添加到 TCPIPS (EYSTARTTCPIPS.DETAILED) 视图:

CIPHERS

SSL 密码套件代码

NUMCIPHERS

用来协商加密级别的密码套件的数目。

TCP/IP 全局状态视图

下列属性已添加到 TCPIPGBL (EYUSTARTTCPIPGBL.DETAILED) 视图:

CRLSERVER

包含证书撤回列表 (CRL) 的 LDAP 服务器名称

SSLCACHE

SSL 使用本地还是综合系统范围的会话标识高速缓存。

时钟和计时视图

下列属性已添加到时钟和计时视图, 此视图是任务操作视图中的某个活动任务视图集 (EYUSTARTTASK.DETAILED):

TMRCMDLY

CICS TCB 更改方式延迟时间

TMRSTDLY

最大 CICS SSL TCB 延迟时间

CICS 区域视图

下列属性已添加到 CICS RGN (EYUSTARTCICSRGN.DETAILED) 视图:

MAXSSLTCBS

显示 SSL OTE 池中 TCB 的当前最大数目。

ACTSSLTCBS

显示当前分配的 SSL 池 TCB 数。

第 3 部分 应用程序转换

CICS Transaction Server for z/OS V3R1 提供一系列新功能，使得您能够使用当代的编程语言、构造和工具，增强现有应用程序，并构建新的应用程序。

第 7 章 增强的程序间数据传输：作为现代 COMMAREA 的通道

传统地，CICS 程序使用通信区（COMMAREA）来交换数据。在 CICS Transaction Server for z/OS V3R1 中，通道和容器提供程序间传送数据的改进方法，在数量上远远超出应用于 COMMAREA 的 32KB 限制。

- 容器是为在程序间传递信息而设计的命名数据块。
- 集合中分组在一起的容器称为通道。通道是用于在 CICS 程序之间交换数据的标准机制，并且类似于参数列表。通道可以在 LINK、START、XCTL 和 RETURN 命令中使用，并带有本地和远程事务。对可添加到通道的容器数没有限制，并且每个容器的大小只受可用存储量的限制。

通道的优点

通道 / 容器模型比 CICS 程序传统用于交换数据的通信区（COMMAREA）有几个优点。例如：

- 与 COMMAREA 不同，通道大小不限制为 32KB。对可以添加到通道的容器数没有限制，并且个别容器的大小只受您可用的存储量限制。
- 因为一个通道可以包含多个容器，所以可使用它以更加结构化的方式传递数据。相反，COMMAREA 是单数据块。
- 与 COMMAREA 不同，通道不要求使用它们的程序知道所返回数据的确切大小。
- 通道是用于在 CICS 程序之间交换数据的标准机制。通道可以在 LINK、START、XCTL 和 RETURN 命令中传递。支持分布式程序链接（DPL），并且通过 START CHANNEL 和 RETURN TRANSID 命令启动的事务可能是远程的。

注：当通道要传递到远程程序或事务时，请注意，传递大量的数据会影响性能。特别是在本地区域和远程区域由 ISC 而不是 MRO 连接进行连接时。

- 通道可以用于用任何一种 CICS 支持的语言编写的 CICS 应用程序。例如，一个 CICS 区域上的 Java 客户机程序可使用通道与后端 AOR 上的 COBOL 服务器程序交换数据。
- 可以编写服务器程序来处理多个通道。例如，它可以：
 1. 动态地发现与它一起被调用的通道
 2. 浏览通道中的容器
 3. 根据传递给它的通道改变它的处理
- 您可以从通过一个或多个通道调用的相关程序集构建“组件”。
- 客户机和组件之间的松耦合允许简单的改进。客户机和组件可以在不同的时间升级。例如，可以先升级组件以处理新的通道，然后升级客户机程序（或编写新的客户机）以使用新的通道。
- 程序员不再关注存储管理。当容器不在作用域中时，CICS 自动销毁容器（以及它们的存储）。
- 通道应用程序使用的数据转换模型比 COMMAREA 应用程序使用的模型简单得多。而且，COMMAREA 应用程序中数据转换由系统程序员控制，而通道应用程序中数据转换由应用程序程序员使用简单 API 命令控制。
- 具有 CICS 商业交易服务（BTS）经验的程序员会发现在非 BTS 应用程序中使用容器会很简便。

- 从通道和 BTS 应用程序都可以调用使用容器的程序。
- 可以将使用容器的非 BTS 应用程序迁移到全 BTS 应用程序中。（它们形成到 BTS 的迁移路线。）

术语

通道 用于在程序间传递数据的一组容器。通道类似于一个参数列表。

容器 用于在程序间传递信息的命名数据块。您可以认为它是“命名通信区（COMMAREA）”。程序间可以传递许多容器。集合中分组在一起的容器称为**通道**。

当前通道

调用程序的通道（如果有的话）。用于特定程序的特定调用的当前通道不会改变。

通道：快速启动

容器和通道

容器是为在程序间传递信息而设计的命名数据块。您可认为它们是“命名通信区（COMMAREA）”。程序可彼此传递任何数量的容器，而容器的大小仅受到可用存储量的限制。

集合中分组在一起的容器称为**通道**。通道类似于一个参数列表。

要创建命名容器并将它们指定给通道，程序使用 EXEC CICS PUT CONTAINER(*container-name*) CHANNEL(*channel-name*) 命令。然后它可以使用 EXEC CICS LINK、XCTL、START 或 RETURN 命令的 CHANNEL(*channel-name*) 选项将通道（和它的容器）传递给第二个程序。

第二个程序可以使用 EXEC CICS GET CONTAINER(*container-name*) 命令读传递给它的容器。此命令读属于与程序一起调用的通道的命名容器。

如果通过 LINK 命令调用第二个程序，则它还可以将容器返回给调用程序。它可通过创建新容器或重用现有容器来这样做。

只有创建通道和容器的程序和它们被传递至的程序才可以看到它们。当这些程序终止时，CICS 自动销毁容器及其存储。

通道容器不可恢复。如果您需要使用可恢复的容器，则使用 CICS 业务交易服务（BTS）容器。通道和 BTS 容器之间的关系在第 178 页的『通道和 BTS 活动』中有所描述。

基本示例

第 162 页的图 16 显示 COBOL 程序 CLIENT1，它可以：

1. 使用 PUT CONTAINER(*container-name*) CHANNEL(*channel-name*) 命令创建名为 inqcustrec 的通道并向它添加两个容器 custno 和 branchno；这些容器分别包含客户号和分支号。
2. 使用 LINK PROGRAM(*program-name*) CHANNEL(*channel-name*) 命令链接到程序 SERVER1，并传递 inqcustrec 通道。

3. 使用 `GET CONTAINER(container-name) CHANNEL(channel-name)` 命令检索 `SERVER1` 返回的客户记录。客户记录在 `inqcustrec` 通道的 `custrec` 容器中。

注意，客户机和服务器程序都使用同一 COBOL 副本 `INQINTC`。此副本的第 3 行和第 5 到 7 行表示 `INQUIRY-CHANNEL` 及其容器。这些行对程序的运行不是必须的。这是因为容器和通道只需命名（例如，使用 `PUT CONTAINER` 命令）即可创建；不一定要定义它们。然而，在这两个程序使用的副本中包括这些行可以使维护更简便；它们记录所使用的容器的名称。

建议

为简便地维护使用通道的客户机 / 服务器应用程序，创建一个副本，它记录所使用的容器的名称并定义映射到容器的数据字段。在客户机和服务器程序中都要包括此副本。

注：该示例显示两个 COBOL 程序。相同的技术可用于 CICS 支持的其他语言中的任何一种。仅对于 *COBOL* 程序，然而，如果服务器程序在 `EXEC CICS GET CONTAINER` 命令中使用 `SET` 选项（而不是 `INTO`），则必须在此程序的 `LINKAGE` 部分定义 `SET` 指向的存储结构。这意味着您将需要两个副本而不是一个。第一个副本在此程序的 `WORKING-STORAGE` 部分命名所使用的通道和容器。第二个副本在 `LINKAGE` 部分定义存储结构。

```
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.
```

图 16. 创建通道并将它传递给第二个程序的程序简单示例

第 163 页的图 17 显示 CLIENT1 链接到的 SERVER1 程序。SERVER1 从已传递给它的 custno 和 branchno 容器中检索数据，并使用此数据在它的数据库中查找完整的客户记录。然后它在同一通道中创建新容器 custrec 并将客户记录返回到此容器中。

注意，程序员在 SERVER1 中的 GET 和 PUT 命令中未指定 CHANNEL 关键字：如果未明确地指定通道，则使用当前通道，即与程序一起调用的通道。


```

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.

```

图 17. 从已传递给它的通道检索数据的链接所至程序的简单示例。此程序被第 162 页的图 16 中显示的程序 *CLIENT1* 所链接。

使用通道：一些典型方案

通道和容器提供了在程序间传递数据的功能强大的方法。此部分包含如何使用通道的一些示例。

一个通道，一个程序

第 164 页的图 18 显示最简单的方案 - 独立程序，以及可以与程序一起调用的单个通道。

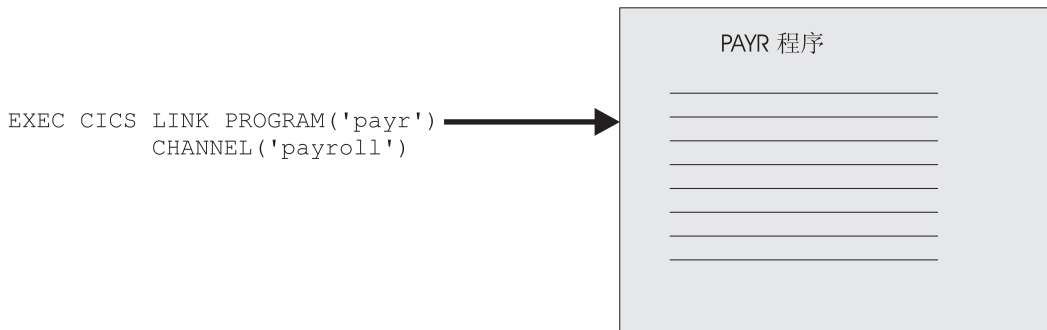


图 18. 带单一通道的独立程序

一个通道，几个程序（组件）

在图 19 中，有一个通道可到相互关联的程序集中的顶层程序。可以将阴影区中的程序集当作组件。客户机程序仅可以“看到”外部通道客户机程序，不知道发生的处理，不了解现有的后端程序。仅可以“看到”

在组件中，程序可以在它们自己之间传递通道。或者，例如，组件程序可以通过创建新通道并从原始通道添加一个或多个容器来传递原始通道的子集。

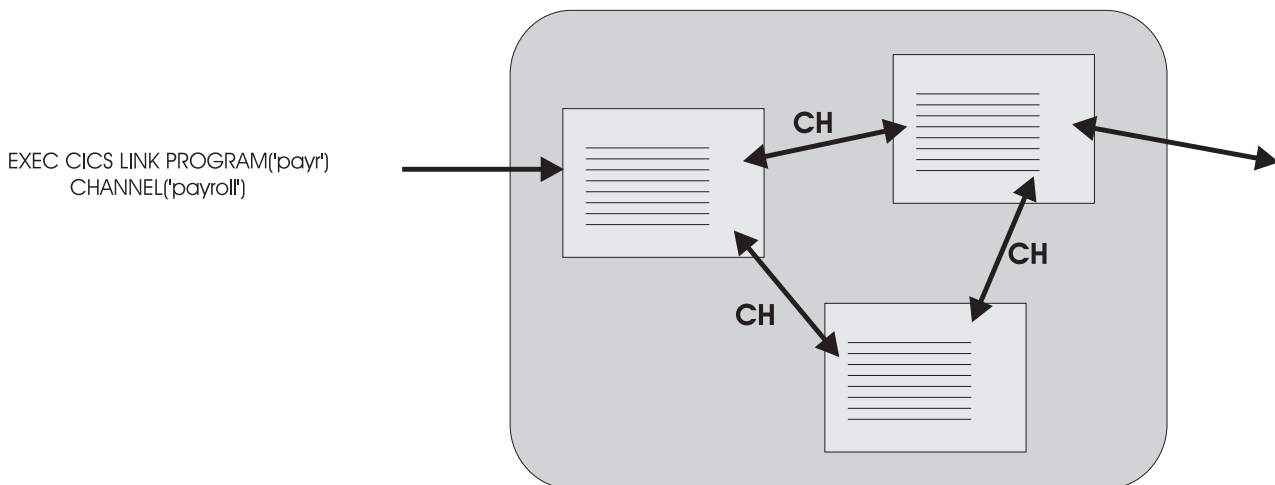


图 19. “组件” - 通过单个外部通道调用的相关程序集。“CH”表明组件中的程序可以在它们自己之间传递通道。

几个通道，一个组件

如我们在图 19 中看到的，我们有可当作组件的相互交互的程序集。然而，然而，这次有两个可替换的外部通道，可以用来调用组件。组件中的顶层程序发出 EXEC CICS ASSIGN CHANNEL 命令以确定与它一起被调用的通道，并相应地剪裁它的处理。

客户机程序和组件之间的“松耦合”允许一些简单的改进。即，客户机和组件可以在不同的时期进行升级。例如，首先可以升级组件以处理第三个通道，它由来自第一个通

道或第二通道的不同容器集组成。接下来可以对客户机程序（编写的新客户机）进行升级以传递第三个通道。

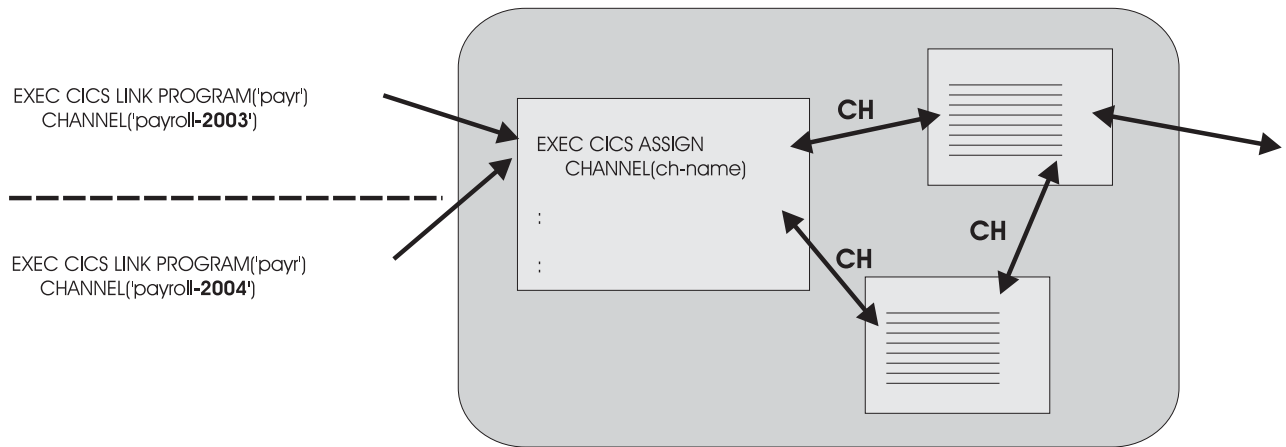


图 20. 到同一组件的多个外部通道. “CH”表明组件中的程序可以在它们自己之间传递通道。

多个交互式组件

图 21显示“Human resources”组件和“Payroll”组件，每个组件都带有可用于调用它的通道。从独立程序和 Human resources 组件都可调用 Payroll 组件。

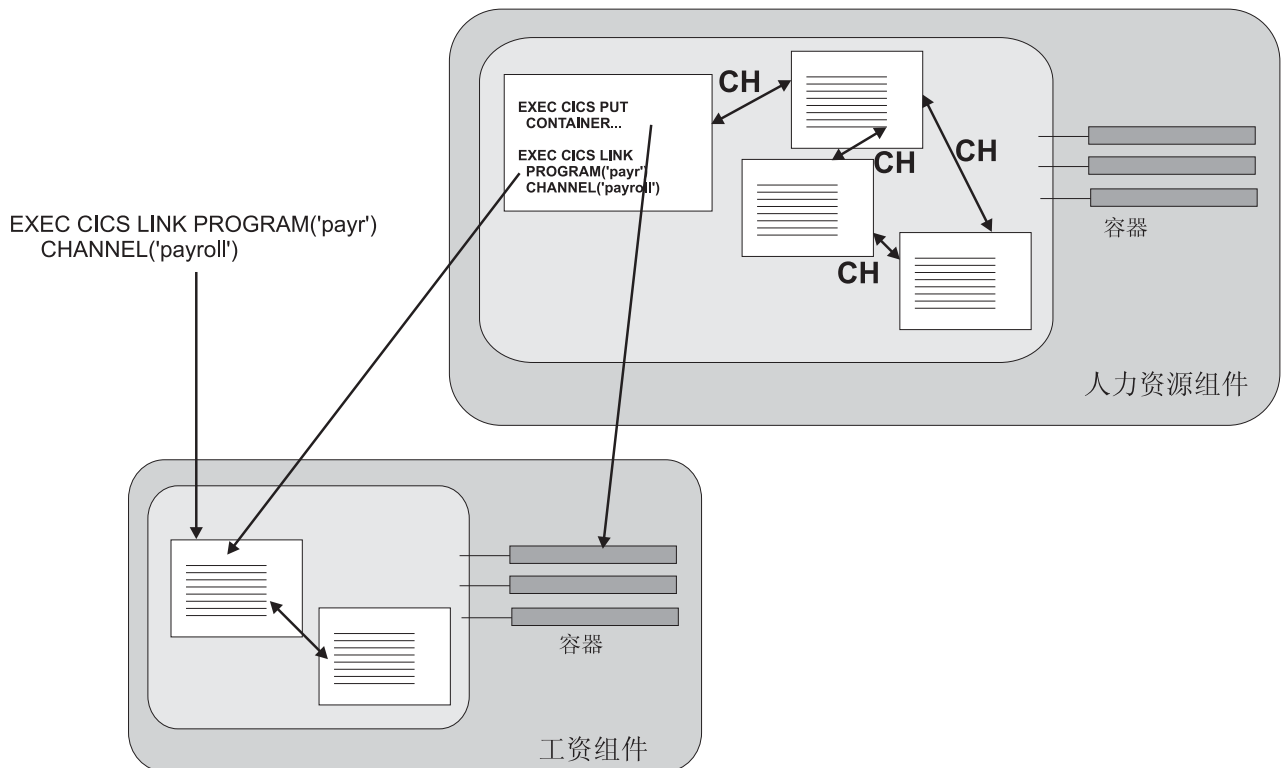


图 21. 通过他们的通道相互作用的多个组件

创建通道

您可以通过在以下一个命令中命名通道来创建它:

```
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
```

如果在当前程序范围内还不存在此通道, 则会创建它。

创建通道并使用数据容器填充它的最直接方法是发出一连串 EXEC CICS PUT CONTAINER(*container-name*) CHANNEL(*channel-name*) FROM(*data_area*) 命令。第一个 PUT 命令创建通道 (如果它还不存在), 并向它添加容器; 后继命令向通道添加更多的容器。如果这些容器已存在, 则它们的内容会被新数据覆盖。

向通道添加容器的替换方法是从另一个通道移动它们。要这样操作, 使用以下命令:

```
EXEC CICS MOVE CONTAINER(container-name) AS(container-new-name)
CHANNEL(channel-name1) TOCHANNEL(channel-name2)
```

注:

1. 如果未指定 CHANNEL 或 TOCHANNEL 选项, 则暗指当前通道。请参阅『当前通道』。
2. 源通道必须在程序范围中。
3. 如果在当前程序范围内还不存在目标通道, 则会创建它。
4. 如果源容器不存在, 则会发生错误。
5. 如果目标容器还不存在, 则会创建它; 如果它已存在, 则会覆盖它的内容。

您可使用 MOVE CONTAINER 而不是 GET CONTAINER 和 PUT CONTAINER 作为在通道间传输数据更有效率的方法。

如果以下命令中命名的通道在当前程序范围中还不存在, 则创建空的通道:

- 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*)

当前通道

程序的当前通道是调用它的通道 (如果有的话)。程序可以创建其他通道。然而, 用于特定程序的特定调用的当前通道不会改变。它类似于一个参数列表。

带 LINK 命令的当前通道示例

第 168 页的图 22 说明了程序的当前通道的来源。它显示五个交互式程序。程序 A 是一个顶层程序, 它可能由终端最终用户启动。它不是由程序启动的并且没有当前通道。

B、C、D 和 E 分别是第二层、第三层、第四层和第五层程序。

程序 B 的当前通道是 X，它是由程序 A 发出的 EXEC CICS LINK 命令中的 CHANNEL 选项传递的。程序 B 通过添加一个容器和删除另一个容器来修改通道 X。

程序 C 的当前通道也是 X，它是由程序 B 发出的 EXEC CICS LINK 命令中的 CHANNEL 选项传递的。

程序 D 没有当前通道，这是因为 C 没有给它传递一个通道。

程序 E 的当前通道是 Y，它是由 D 的 EXEC CICS LINK 命令中的 CHANNEL 选项传递的。

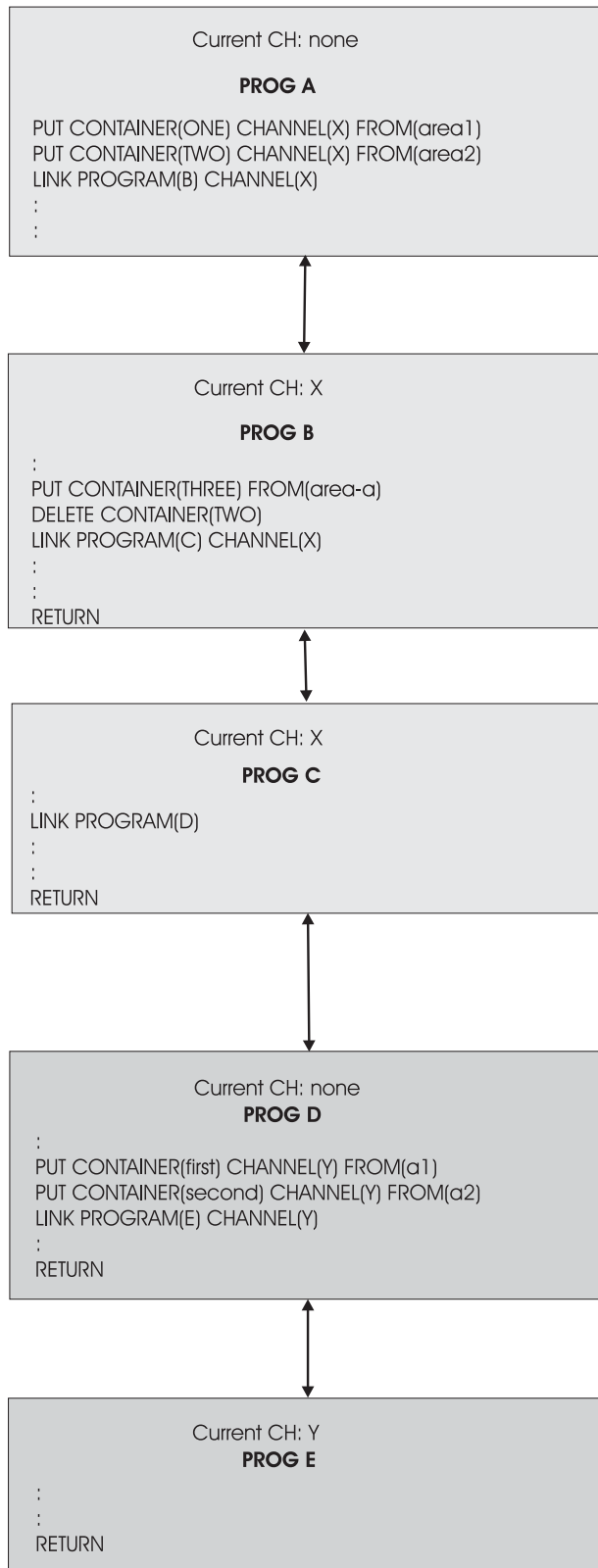


图 22. 当前通道 - 带 LINK 命令的示例

第 169 页的表 1 列出图 22 中显示的五个程序每一个的当前通道（如果有的话）的名称。

表 1. 交互式程序的当前通道 - 带 LINK 命令的示例

程序	当前通道	发出命令	注释
A	无	<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>程序 A 创建通道 X 并将它传递给程序 B。</p> <p>注意，控制由程序 B 返回给程序 A 时，X 通道通已被修改，它包含的容器集与它被程序 A 创建时的容器集已不同（容器 2 已删除，而容器 3 由程序 B 添加）。</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>通过添加和删除容器，程序 B 修改通道 X（它的当前通道）并将已修改的通道传递到程序 C。</p> <p>程序 B 不需要在 PUT CONTAINER 和 DELETE CONTAINER 命令中指定 CHANNEL 选项；它的当前通道是暗指的。</p>
C	X	<pre> . EXEC CICS LINK PROGRAM(D) . EXEC CICS RETURN </pre>	<p>程序 C 链接到程序 D，但不给它传递通道。</p>
D	无	<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>程序 D 创建新的通道 Y，并将它传递给程序 E。</p>
E	Y	<pre> . RETURN . </pre>	<p>程序 E 对已传递给它的数据执行一些处理并返回。</p>

带 XCTL 命令的当前通道示例

第 170 页的图 23 显示四个交互式程序。A1 是一个顶层程序，它可能由终端最终用户启动。它不是由程序启动的并且没有当前通道。B1、B2 和 B3 是所有第二层程序。

B1 的当前通道是 X，它是由 A1 发出的 EXEC CICS LINK 命令中的 CHANNEL 选项传递的。

B2 没有当前通道，这是因为 B1 没有给它传递一个通道。

B3 的当前通道是 Y，它是由 B2 发出的 EXEC CICS XCTL 命令中的 CHANNEL 选项传递的。

当 B3 返回时，通道 Y 及其容器被 CICS 删除。

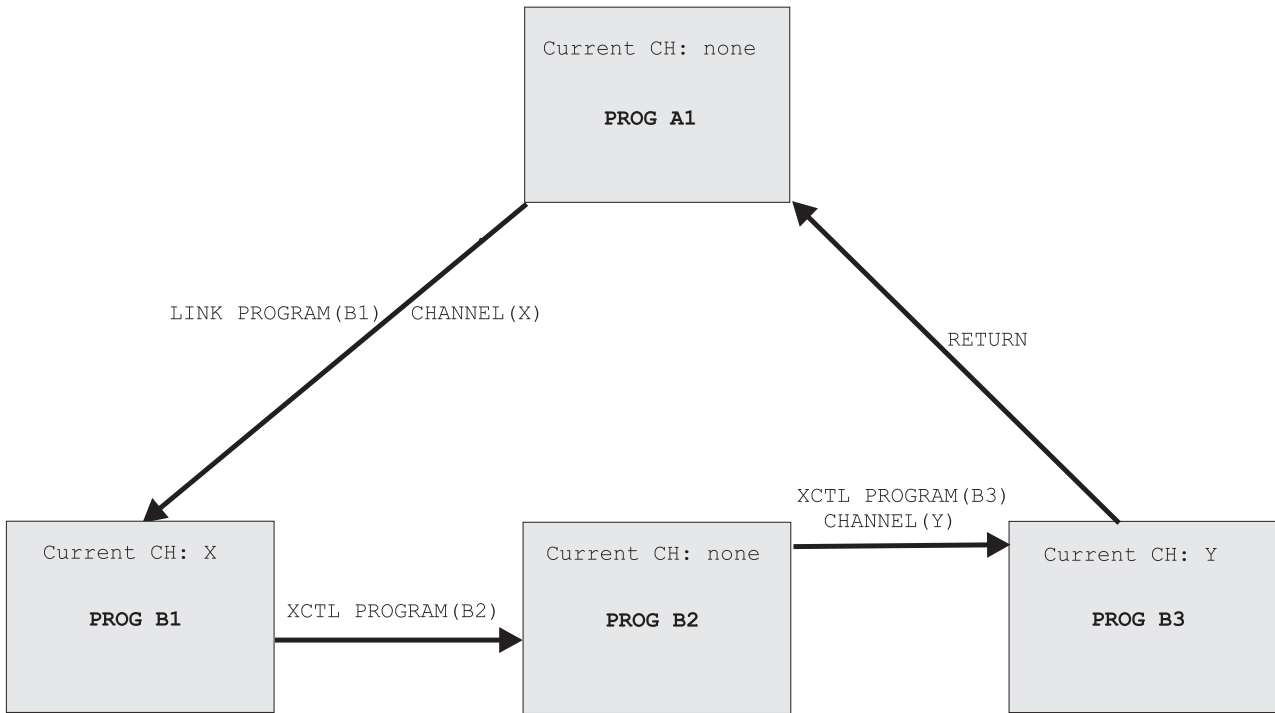


图 23. 带 XCTL 命令的当前通道

表 2 列出图 23 中显示的四个程序每一个的当前通道（如果有的话）的名称。

表 2. 交互式程序的当前通道（示例）

程序	当前 [®] 通道	发出命令
A1	无	. EXEC CICS LINK PROGRAM(B1) CHANNEL(X) .
B1	X	. EXEC CICS XCTL PROGRAM(B2) .
B2	无	. EXEC CICS XCTL PROGRAM(B3) CHANNEL(Y) .
B3	Y	. EXEC CICS RETURN .

当前通道: START 和 RETURN 命令

除了 EXEC CICS LINK 和 XCTL，还可使用其他两个命令来调用程序并给它传递通道:

EXEC CICS START TRANSID(*transid*) CHANNEL(*channel-name*)

给实施已启动事务的程序（或第一个程序（如果有多个程序））传递通道，该通道就成为它的当前通道。

EXEC CICS RETURN TRANSID(*transid*) CHANNEL(*channel-name*)

只有在以下情况下 CHANNEL 选项是有效的:

- 在伪会话 RETURN 上，即在 RETURN 命令中，此命令在 TRANSID 选项中指定将在用户终端运行的下一个事务。

- 如果由最高逻辑级别的程序（即，向 CICS 返回控制的程序）发出。
给实施下一个事务的程序传递通道，该通道就成为它的当前通道。

通道的作用域

通道的作用域是代码，通过它访问通道。

带 LINK 命令的范围示例

第 172 页的图 24 显示第 166 页的『带 LINK 命令的当前通道示例』中描述五个相同的交互式程序。

X 通道的范围（可以从通道访问的代码）是程序 A、B 和 C。

Y 通道的范围是程序 D 和 E。

注意，控制由程序 B 返回给程序 A 时，X 通道通已被修改，它包含的容器集与它被程序 A 创建时的容器集已不同。

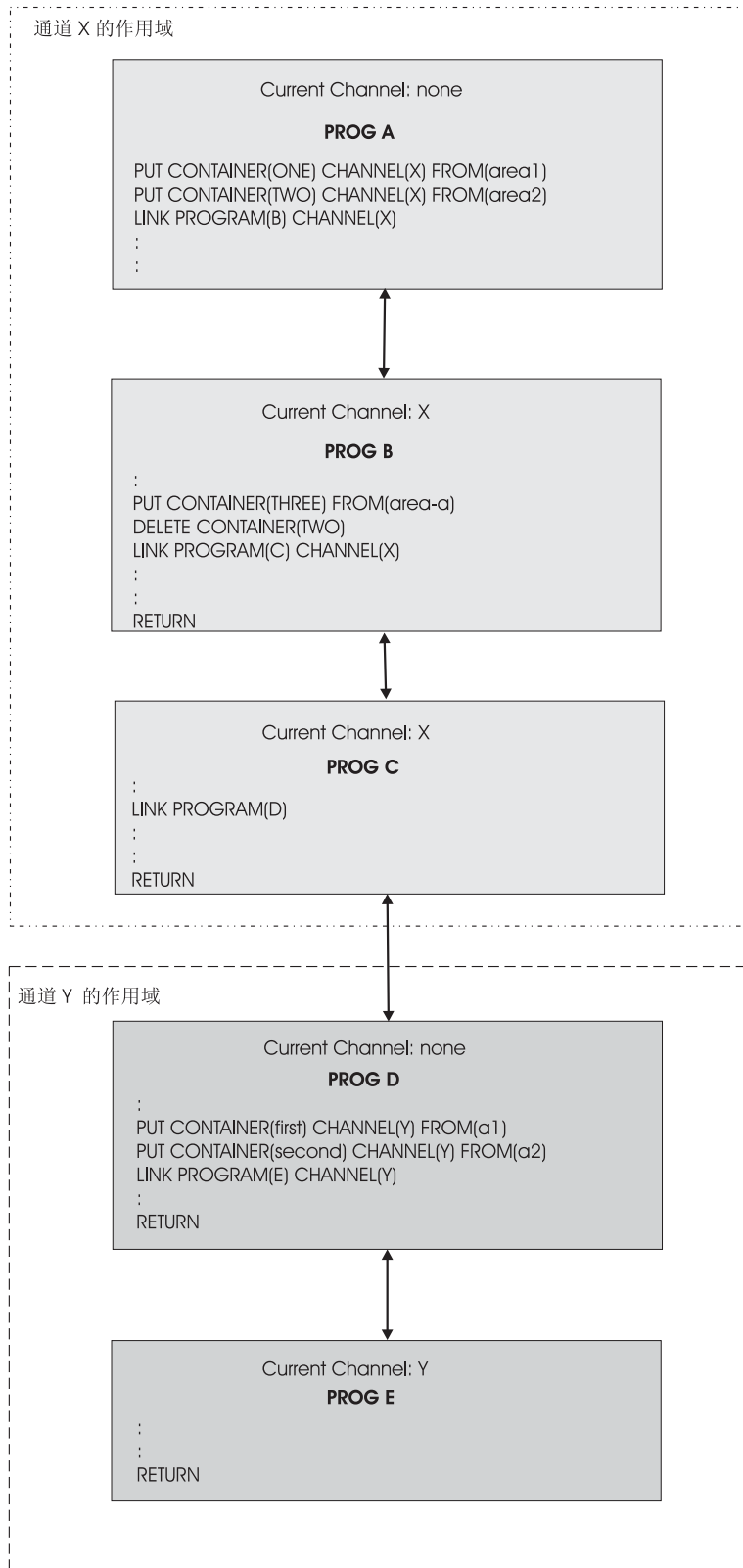


图 24. 通道的范围 - 显示 LINK 命令的示例

第 173 页的表 3 列出图 24 中显示的五个程序中每一个的当前通道（如果有的话）的名称和范围。

表 3. 通道的范围 - 带有 LINK 命令的示例

程序	当前通道	通道的范围
A	无	不可用
B	X	A, B, C
C	X	A, B, C
D	无	不可用
E	Y	D, E

带 LINK 和 XCTL 命令的范围示例

第 174 页的图 25 显示第 169 页的『带 XCTL 命令的当前通道示例』中描述的几个相同的交互式程序，以及由 EXEC CICS LINK 命令从 B1 调用的第三层程序 C1。

X 通道的范围限制为 A1 和 B1。

Y 通道的范围限制为 B2 和 B3。

Z 通道的范围限制为 B1 和 C1。

注意，控制由程序 B3 返回给程序 A1 时，X 通道可能已被程序 B1 修改，它包含的容器集可能与它被程序 A1 创建时的容器集已不同。

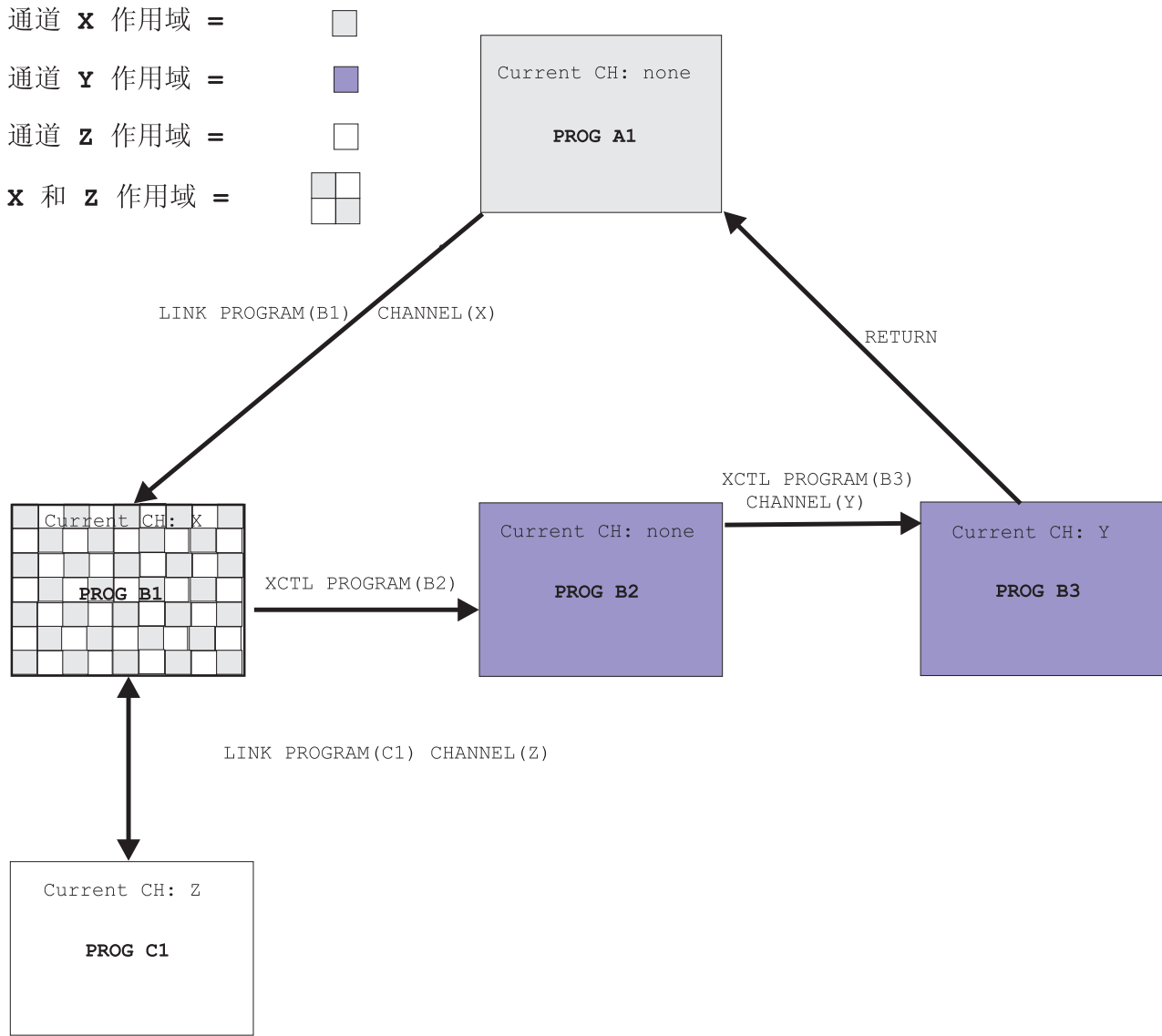


图 25. 通道的范围 - 显示 LINK 和 XCTL 命令的示例

表 4 列出图 25 中显示的五個程序中每個程序的當前通道（如果有的話）的名稱和範圍。

表 4. 通道的範圍 - 帶有 LINK 和 XCTL 命令的示例

程序	當前通道	通道的範圍
A1	无	不可用
B1	X	A1 和 B1
B2	无	不可用
B3	Y	B2 和 B3
C1	Z	B1 和 C1

发现哪些容器传递给程序

通常，会编写交换通道的程序来处理该通道。也就是说，客户机和服务器程序都知道通道的名称，以及通道中的容器的名称和数目。然而例如，如果编写服务器程序或组件来处理多个通道，则在调用时，它必须发现程序可能已通过哪些通道。

通过发出 EXEC CICS ASSIGN CHANNEL 命令，程序可能会发现它的当前通道，即，程序是通过该通道调用的。（如果没有当前通道，命令将返回空白。）

程序也可以（它应该需要）通过浏览，获取它的当前通道中的容器名称。通常，这不是必需的。编写程序来处理若干通道经常被编码成明了每个可能的通道中的容器的名称和数目。

要获取当前通道中的容器名称，使用浏览命令：

- EXEC CICS STARTBROWSE CONTAINER BROWSETOKEN(*data-area*).
- EXEC CICS GETNEXT CONTAINER(*data-area*) BROWSETOKEN(*token*).
- EXEC CICS ENDBROWSE CONTAINER BROWSETOKEN(*token*).

获取它的当前通道的名称，并且若有必要，获取通道中的容器名称，服务器程序可以调整它的处理来适应已传递的数据的类型。

发现从链接返回的容器

程序创建通道，它通过 EXEC CICS LINK PROGRAM(*program-name*) CHANNEL(*channel-name*) 命令将该通道传递到第二个程序。第二个程序对已传递的数据执行一些处理，并且在同一个通道（它的当前通道）中返回结果。

返回时，第一个程序知道已返回的通道的名称，但不需要知道通道中的容器名称。（返回的通道包含与通过的通道相同的容器，还是第二个程序删除了某些容器或者创建了一些其他一些容器？）通过浏览，第一个程序可以发现容器名。要实施它，它使用以下命令：

- 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只读容器

CICS 可以创建通道和容器以供它自己使用，并且将它们传递到用户程序。在某些情况下，CICS 将这些容器标记为只读，因此用户程序无法修改 CICS 在从用户程序中返回时需要的数据。

用户程序不能创建只读容器。

您无法覆盖、移动或删除只读容器。因此，如果您在 PUT CONTAINER、MOVE CONTAINER 或 DELETE CONTAINER 命令上指定只读容器，INVREQ 条件将发生。

设计通道: 最佳实践

可以使用容器以使用通信区 (COMMAREA) 传递数据的传统方法来传递数据。但通道比起 COMMAREA 有几个优点 (请参阅第 159 页的『通道的优点』), 设计您的通道以充分利用这些优点是值得的。

在 DPL 调用结束时, 服务器程序未更改的输入容器不返回给客户机。返回服务器程序已更改其内容的输入容器, 以及服务器程序创建的容器。因此, 为了优化 DPL 性能:

- 对输入数据和输出数据使用单独的容器。
- 服务器程序, 而不是客户机应该创建输出容器。
- 对只读数据和读写数据使用单独的容器。
- 如果结构是可选的, 则使它作为单独的容器。
- 使用专用的容器用于错误信息。

这里是一些设计通道的一般技巧。它们包含和扩展对实现最优化 DPL 性能的建议。

- 对输入数据和输出数据使用单独的容器。这会导致:
 - 更好地封装数据, 这使您的程序维护更容易。
 - 当通道在 DPL 调用上传递时会效率更高, 这是因为较小的容器在每个方向上流动。
- 服务器程序, 而不是客户机应该创建输出容器。如果客户机创建它们, 则空容器将返回给服务器区域。
- 对只读数据和读写数据使用单独的容器。这会导致:
 - 简化副本结构, 使您的程序更易于理解。
 - 避免使用 REORDER 覆盖程序段导致的问题。
 - CICS 区域间的更有效传输, 这是因为发送给服务器区域的只读容器将不返回。
- 对每种结构使用单独的容器。这会导致:
 - 更好地封装数据, 这使您的程序理解和维护更容易。
 - 更改某个结构时会更容易, 这是因为您不需要重新安排整个组件。
 - 通过使用 MOVE CONTAINER 命令在通道之间移动容器, 将通道子集传递到子组件的能力。
- 如果结构是可选的, 则使它作为单独的容器。这会使效率更高, 因为仅当容器存在时才传递结构。
- 使用专用的容器用于错误信息。这会导致:
 - 更好地记录错误信息的内容。
 - 效率更高, 因为:
 1. 只有在发生错误时才传递回包含错误信息的结构。
 2. 通过发出启动通道中容器浏览的 GET CONTAINER(*known-error-container-name*) 命令 (并可能接收 NOTFOUND 条件), 更有效地检查错误容器是否存在。
- 当您需要传递不同类型的数据 (例如, 二进制数据和字符数据) 时, 对每种类型使用单独的容器, 而不使用带有一个结构复杂的容器。这样改进在不同代码页之间移动的能力。
- 当您需要传递大量的数据时, 将它分开放在多个容器中, 而不是将它仅放入一个容器中。

当通道传递到远程程序或事务时，传递大量的数据会影响性能。特别是当本地区域和远程区域是通过 ISC 而不是 MRO 连接进行连接时，会影响性能。

- 注意，不要创建过多容器，您限制可用于其他应用程序的存储数。

要获得有关迁移使用 COMMAREA 的程序，以使用通道作为替代的信息，请参阅第 208 页的『从 COMMAREAs 迁移到通道』。

构造和使用通道：示例

图 26 显示 CICS 客户机程序，它可以：

1. 使用 EXEC CICS PUT CONTAINER 命令构造（并将数据放入）一组容器。这些容器都是同一命名通道“payroll-2004”的一部分。
2. 发出 EXEC CICS LINK 命令以调用 PAYR 服务器程序，并为它传递 payroll-2004 通道。
3. 发出 EXEC CICS GET CONTAINER 命令来检索服务器的程序输出，输出应该在 payroll-2004 通道的 status 容器中。

```
* 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)
```

图 26. 客户机程序如何构造通道、将它传递给服务器程序和检索服务器的输出。

第 178 页的图 27 显示客户机调用的 PAYR 服务器程序的一部分。服务器程序：

1. 查询已与它一起调用的通道。
2. 发出 EXEC CICS GET CONTAINER 命令从 payroll-2004 的 employee 和 wage 容器检索输入。
3. 处理输入数据。
4. 发出 CICS PUT CONTAINER 命令将其输出返回到 payroll-2004 通道的 status 容器。

```

"PAYR", CICS COBOL server program

* 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"
  :
  :
  :

```

图 27. 服务器程序如何查询它已传递的通道、从通道的容器检索数据并将输出返回给调用者。

通道和 BTS 活动

用于构建通道和与之交互的 PUT、GET、MOVE 和 DELETE CONTAINER 命令类似于 CICS 商业交易服务 (BTS) 应用程序中使用的那些命令。这样具有 BTS 经验的程序员会发现在非 BTS 应用程序中使用容器会很简便。而且，从通道和 BTS 应用程序都可以调用使用容器的服务器程序。第 179 页的图 28 中显示这样一个示例。

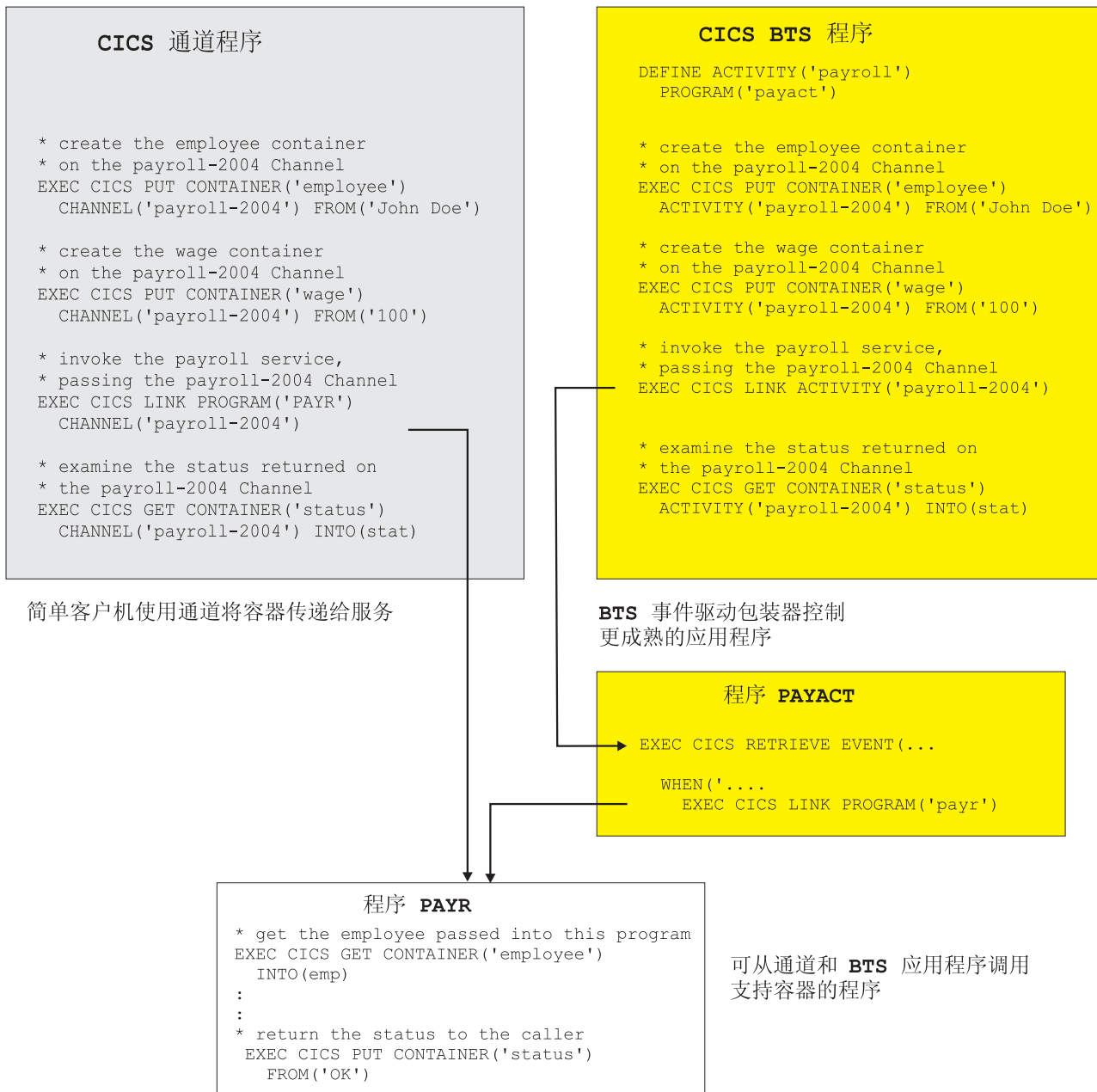


图 28. 通道和 BTS 活动

上下文

如图 28 中所示，无需更改就可以使用发出容器命令的程序，作为通道应用程序的一部分或作为 BTS 活动的一部分。

对于要在通道和 BTS 上下文中使用的程序，它发出的容器命令不能指定将它们标识为通道或 BTS 命令的任何选项。要在每个容器命令中避免使用的选项为：

- DELETE CONTAINER**
- ACQACTIVITY (特定的 BTS)
- ACQPROCESS (特定的 BTS)

ACTIVITY (特定的 BTS)
CHANNEL (特定的通道)
PROCESS (特定的 BTS)

GET CONTAINER

ACQACTIVITY (特定的 BTS)
ACQPROCESS (特定的 BTS)
ACTIVITY (特定的 BTS)
CHANNEL (特定的通道)
INTOCCSID (特定的通道)
PROCESS (特定的 BTS)

MOVE CONTAINER

FROMACTIVITY (特定的 BTS)
CHANNEL (特定的通道)
FROMPROCESS (特定的 BTS)
TOACTIVITY (特定的 BTS)
TOCHANNEL (特定的通道)
TOPROCESS (特定的 BTS)

PUT CONTAINER

ACQACTIVITY (特定的 BTS)
ACQPROCESS (特定的 BTS)
ACTIVITY (特定的 BTS)
CHANNEL (特定的通道)
DATATYPE (特定的通道)
FROMCCSID (特定的通道)
PROCESS (特定的 BTS)

当执行容器命令时，CICS 分析在它内部发生的上下文（通道、BTS 或两者都不），以确定如何处理此命令。要确定上下文，CICS 使用以下测试顺序：

1. 通道：程序有当前通道吗？
2. BTS：是 BTS 活动的程序部分吗？
3. 无：程序没有当前通道并且不是 BTS 活动的一部分。因此它没有在其中执行容器命令的上下文。拒绝执行此命令并返回 INVREQ 条件和 RESP2 值 4。

使用来自 JCICS 的通道

CICS 提供 CICS Java 程序可用于传递和接收通道的以下 JCICS 类：

- `com.ibm.cics.server.Channel`
- `com.ibm.cics.server.Container`
- `com.ibm.cics.server.ContainerIterator`

CICS 还提供用于处理错误的以下异常类：

- `com.ibm.cics.server.CCSIDErrorException`
- `com.ibm.cics.server.ChannelErrorException`
- `com.ibm.cics.server.ContainerErrorException`

在 JCICS 中创建通道和容器

要创建通道，使用 Task 类的 `createChannel()` 方法。例如：

```
Task t=Task.getTask();
Channel custData = t.createChannel("Customer_Data");
```

提供给 createChannel 方法的字符串是 CICS 所知的 Channel 对象的名称。(用 16 个字符的空格填充此名称, 以符合 CICS 命名约定。)

要在通道中创建新容器, 使用 Channel 的 createContainer() 方法。例如:

```
Container custRec = custData.createContainer("Customer_Record");
```

提供给 createContainer() 方法的字符串是 CICS 所知的 Container 对象的名称。(如有必要, 用空格将此名称填补到 16 个字符, 以符合 CICS 命名约定。)如果此通道中已存在同名的容器, 则会抛出 ContainerErrorException。

将数据放入容器

要将数据放入 Container 对象, 使用 Container.put() 方法。可以将数据作为字节数组或字符串添加到容器。例如:

```
String custNo = "00054321";
byte[] custRecIn = custNo.getBytes();
custRec.put(custRecIn);
```

Or simply:

```
custRec.put("00054321");
```

将通道传递给另一个程序或任务

要在程序链接或转换程序控制 (XCTL) 调用上传递通道, 分别使用 Program 类的 link() 和 xctl() 方法:

```
programX.link(custData);

programY.xctl(custData);
```

要在 program-return 调用上设置下一个通道, 使用 TerminalPrincipalFacility 类的 setNextChannel() 方法:

```
terminalPF.setNextChannel(custData);
```

要在 START 请求上传递通道, 使用 StartRequest 类的 issue 方法:

```
startrequest.issue(custData);
```

接收当前通道

程序没有必要明确地接收它的当前通道, 请参阅第 182 页的『浏览当前通道』。然而, 程序可以从当前任务获取它的当前通道; 这使它能按名称抽出容器:

```
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");
}
```

从容器获取数据

使用 `Container.get()` 方法将容器中的数据读到字节数组中:

```
byte[] custInfo = custRec.get();
```

浏览当前通道

传递通道的 JCICS 程序可以访问所有容器对象而无需明确地接收通道。它使用 `ContainerIterator` 对象来这样操作。(`ContainerIterator` 类实现 `java.util.Iterator` 接口。) 当 `Task` 对象从当前任务实例化时, 它的 `containerIterator()` 方法对当前通道返回 `Iterator`, 或如果没有当前通道则返回 `null`。 例如:

```
Task t = Task.getTask();
ContainerIterator ci = t.containerIterator();
While (ci.hasNext()) {
    Container custData = ci.next();
    // Process the container...
}
```

JCICS 示例

第 183 页的图 29 显示一个称为 `Payroll` 的 Java 类, 它调用名为 `PAYR` 的 COBOL 服务器程序。 `Payroll` 类使用 JCICS `com.ibm.cics.server.Channel` 类和 `com.ibm.cics.server.Container` 类执行非 Java 客户机程序会使用 `EXEC CICS` 命令执行的操作。

```

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();
    ...
}

```

图 29. 使用 JCICS *com.ibm.cics.server.Channel* 和 *com.ibm.cics.server.ContainerJava* 类向 COBOL 服务器程序传递通道的 Java 类

与通道一起动态路由

传递 COMMAREA 或通道的 EXEC CICS LINK 和 EXEC CICS START 命令可动态路由。因此，可动态路由以下类型的通道相关请求：

- 程序链接（DPL）请求
- 终端相关 START 请求启动的事务
- 终端不相关 START 请求

如果存在预动态路由的 program-link 或 START 命令关联的通道，在其通信区域的 DYRCHANL 字段中，将通道名称传递给路由程序。DYRCHANL 字段仅适用于上面所列出的三种类型的请求。对于其他类型的请求，或如果没有与此请求关联的通道，则它不包含任何内容。

注：路由程序的通信区由 DFHDYPDS DSECT 映射。

注意，因为路由程序被赋予通道的名称，而不是它的地址，所以路由程序无法使用 DYRCHANL 字段查看或更改通道容器的内容。

当 LINK 或 START 命令传递 COMMAREA，而不是通道时，根据请求的类型，路由程序可检查或更改 COMMAREA 的内容。对于终端相关 START 请求（由动态路由程

序处理)启动,而不是终端无关的 START 请求(由分布式路由程序处理)启动的 LINK 请求和事务,在 DFHDYPDS 的 DYRACMAA 字段中,路由程序被赋予应用程序 COMMAREA 的地址,而且可检查和更改其内容。

要赋予路由程序与通道相同类型的功能,使用通道的应用程序可在通道内创建一个特殊容器,名为 DFHROUTE。如果应用程序发出 LINK 或要动态路由的终端相关 START 请求(不是与终端无关的 START 请求),则在 DFHDYPDS 的 DYRACMAA 字段中,动态路由程序被赋予 DFHROUTE 容器的地址,而且可检查和更改其内容。

如果您更改程序以传递通道,而不是 COMMAREA,您应该使用其现有的 COMMAREA 结构映射 DFHROUTE。

要获得有关动态和分布式路由的介绍信息,请参阅 *CICS Intercommunication Guide*。要获得有关编写动态或分布式路由程序的信息,请参阅 *CICS Customization Guide*。

数据转换

为何需要进行数据转换?

这里是需要进行数据转换的一些情况:

- 在使用不同编码标准(例如,EBCDIC 和 ASCII)的平台间传递字符数据时。
- 要将一些字符数据的编码从一个编码字符集标识(CCSID)转换成另一个时。

带通道的数据转换

使用通道交换数据的应用程序使用一种简单的数据转换模型。通常不需要进行转换,而当需要转换时,可使用单个编程指令告诉 CICS 自动处理它。

注意以下事项:

- 通常,当(非 Java) CICS TS 程序调用另一个(非 Java) CICS TS 程序时,不需要进行转换,这是因为这两个程序都使用 EBCDIC 编码。例如,如果 CICS TS C 语言程序调用 CICS TS COBOL 程序,并传递给它一些保存字符数据的容器,则使用数据转换的唯一理由是:其中一个程序要更改数据的 CCSID,这不经常发生。
- 通道应用程序使用的数据转换模型比 COMMAREA 应用程序使用的模型简单得多。使用 COMMAREA 交换数据的应用程序使用传统数据转换模型,在 *CICS Family: Communicating from CICS on System/390*[®] 手册中有所描述。转换是在系统程序员的控制下使用 DFHCNV 转换表、DFHCCNV 转换程序和 DFHUCNV 用户可替换转换程序(可选)来完成的。

相反,通道容器中的数据是在应用程序员的控制下使用 API 命令来转换的。

- 应用程序员仅负责用户数据的转换,即,容器中由他的应用程序创建的数据。系统数据由 CICS 在必要时自动转换。
- 应用程序员仅关注字符数据的转换。不支持二进制数据(大尾数法和小尾数法之间)的转换。
- 您的应用程序可使用容器 API(作为一种简单的方法)将字符数据从一个代码页转换成另一个代码页,请参阅第 186 页的『使用容器执行代码页转换』。

出于数据转换目的,CICS 识别两种数据类型:

CHAR 字符数据，即文本字符串。容器中的数据转换（如果需要）成检索它的应用程序的代码页。如果检索数据的应用程序是基于 ASCII 的系统上的客户机，则此代码页将是 ASCII 代码页。如果它是 CICS Transaction Server for z/OS 应用程序，则此代码页将是 EBCDIC 代码页。

容器中的所有数据都会进行转换，好象它是一个单独的字符串。对于单字节字符集（SBCS）代码页，由几个字符字段组成的结构相当于单字节字符串。然而，对于双字节字符集（DBCS）代码页，就不是这种情况。如果您使用 DBCS 代码页，则为了确保数据转换正确地工作，您必须将每个字符串放入到单独的容器中。

BIT 所有非字符数据，即，未指定为类型 CHAR 的所有数据。不能对此容器中的数据进行转换。它是缺省值。

用于数据转换的 API 命令是：

- EXEC CICS PUT CONTAINER [CHANNEL] [DATATYPE] [FROMCCSID]
- EXEC CICS GET CONTAINER [CHANNEL] [INTOCCSID]

要获取关于 PUT CONTAINER (CHANNEL) 命令的详细信息，请参阅第 192 页的『PUT CONTAINER (CHANNEL)』。要获取关于 GET CONTAINER (CHANNEL) 命令的详细信息，请参阅第 188 页的『GET CONTAINER (CHANNEL)』。

如何使 CICS 自动转换字符数据

1. 在客户机程序中，使用 PUT CONTAINER 命令的 DATATYPE(DFHVALUE(CHAR)) 选项指定由容器保存字符数据，而且数据适合转换。例如：

```
EXEC CICS PUT CONTAINER(cont_name) CHANNEL('payroll')
        FROM(data1) DATATYPE(DFHVALUE(CHAR))
```

除非数据的 CCSID 不是客户机平台的缺省 CCSID，否则不需要指定 FROMCCSID 选项。（对于 CICS TS 区域，在 LOCALCCSID 系统初始化参数上指定缺省 CCSID。）缺省 CCSID 是暗指的。

2. 在服务器程序中，发出 GET CONTAINER 命令以从程序的当前通道检索数据：

```
EXEC CICS GET CONTAINER(cont_name) INTO(data_area1)
```

以服务器平台的缺省 CCSID 返回数据。除非您希望将数据转换成不同于缺省 CCSID 的 CCSID，否则不需要指定 INTOCCSID 选项。如果客户机平台和服务器平台不相同，则会自动进行数据转换。

3. 在服务器程序中，发出 PUT CONTAINER 命令以将值返回给客户机：

```
EXEC CICS PUT CONTAINER(status) FROM(data_area2)
        DATATYPE(DFHVALUE(CHAR))
```

DATATYPE(DFHVALUE(CHAR)) 选项指定由容器保存字符数据，而且数据适合转换。除非数据的 CCSID 不是服务器平台的缺省 CCSID，否则不需要指定 FROMCCSID 选项。

4. 在客户机程序中，发出 GET CONTAINER 命令检索服务器程序返回的状态：

```
EXEC CICS GET CONTAINER(status) CHANNEL('payroll')
        INTO(status_area)
```

状态返回到客户机平台的缺省 CCSID。除非您希望将数据转换成不同于缺省 CCSID 的 CCSID，否则不需要指定 INTOCCSID 选项。如果客户机平台和服务器平台不相同，则会自动进行数据转换。

使用容器执行代码页转换

您的应用是程序可使用容器 API（作为一种简单的方法）将字符数据从一个代码页转换成另一个代码页。以下示例是将数据从 codepage1 转换成 codepage2:

```
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)
```

以下示例是将数据从区域的缺省 EBCDIC 代码页转换成指定的 UTF8 代码页:

```
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)
```

当用这种方式使用容器 API 时，要记住以下事项:

- 在 GET CONTAINER 命令中，使用 SET 选项而不是 INTO，除非所转换的长度是已知的。（您可通过发出 GET CONTAINER(*cont_name*) NODATA FLENGTH(*len*) 命令来检索已转换数据的长度。
- 为了避免存储开销，转换之后，复制所转换的数据并删除容器。
- 为避免装运通道，可使用临时通道。
- 不可能进行完全的转换。即，如果指定的代码页和通道的代码页是不支持的组合时，则会发生代码页转换错误。

SOAP 示例

CICS TS SOAP 应用程序:

1. 从套接字或 MQ 消息队列检索 UTF8 或 UTF16 消息。
2. 将消息以 UTF8 格式放入到容器中。
3. 将 EBCDIC 数据结构放入到同一通道中的其他容器中。
4. 对后端 AOR 上的处理程序进行分布式程序链接（DPL）调用以传递通道。

也在 CICS TS 上运行的后端处理程序可以使用 EXEC CICS GET CONTAINER 命令检索 EBCDIC 数据结构或消息。它可以获取 UTF8 或 UTF16 的消息、或者它自己的或区域的 EBCDIC 代码页的消息。同样，它可以使用 EXEC CICS PUT CONTAINER 命令将数据以 UTF8、UTF16 或 EBCDIC 格式放入到容器中。

要检索区域的 EBCDIC 代码页的一条消息，处理程序可以发出命令:

```
EXEC CICS GET CONTAINER(input_msg) INTO(msg)
```

因为未指定 INTOCCSID 选项，所以消息数据会自动转换成区域的 EBCDIC 代码页。（假设 PUT CONTAINER 命令（它用来存储通道中的消息数据）指定了 DATATYPE 为 CHAR；如果它指定缺省值 DATATYPE 为 BIT，则不可能进行转换。）

要返回代码页为区域的 EBCDIC 代码页的某些输出，则处理程序可以发出命令:

```
EXEC CICS PUT CONTAINER(output) FROM(output_msg)
```

因为未指定 CHAR，所以将不允许数据转换。因为未指定 FROMCCSID 选项，所以获得的消息数据在区域的 EBCDIC 代码页中。

要检索格式为 UTF8 的一条消息，处理程序可发出命令:


```
EXEC CICS GET CONTAINER(input_msg) INTO(msg) INTOCCSID(utf8)
```

要防止将数据自动转换成区域的 EBCDIC 代码页，需要 INTOCCSID 选项。

要以 UTF8 格式返回某些输出，服务器程序可发出命令：

```
EXEC CICS PUT CONTAINER(output) FROM(output_msg) FROMCCSID(utf8)
```

FROMCCSID 选项指定消息数据当前为 UTF8 格式。因为未指定 CHAR，所以将不允许数据转换。

需求

要支持此功能，没有特别的硬件或软件需求。

相关信息

第 311 页的第 27 章，『CICS 操作环境』

对 CICS 外部的更改

对应用程序编程接口的更改

新命令

存在五个新命令，用于使用容器和通道。要获得更多信息，请参阅：

第 188 页的『DELETE CONTAINER (CHANNEL)』

第 188 页的『GET CONTAINER (CHANNEL)』

第 190 页的『MOVE CONTAINER (CHANNEL)』

第 192 页的『PUT CONTAINER (CHANNEL)』

第 194 页的『START CHANNEL』

已更改的命令

下列应用程序编程命令已修改。

```
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
```

要获得更多信息，请参阅第 197 页的『已修改的 API 命令』。

新 API 命令

已引入了以下新的应用程序编程命令:

- 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):

删除指定的通道容器。

DELETE CONTAINER (CHANNEL)

▶▶—DELETE—CONTAINER(*data-value*)—▶▶
└─CHANNEL(*data-value*)—┘

条件: CHANNELERR, CONTAINERERR, INVREQ

此命令是线程安全的。

描述

DELETE CONTAINER (CHANNEL) 从通道删除容器并废弃它包含的任何数据。

容器由名称和通道（拥有该容器的通道）来标识。可以通过以下方式识别拥有容器的通道:

- 明确地指定 CHANNEL 选项识别。
- 隐式地省略 CHANNEL 选项识别。如果省略该选项，则暗指当前通道。

选项

CHANNEL(*data-value*)

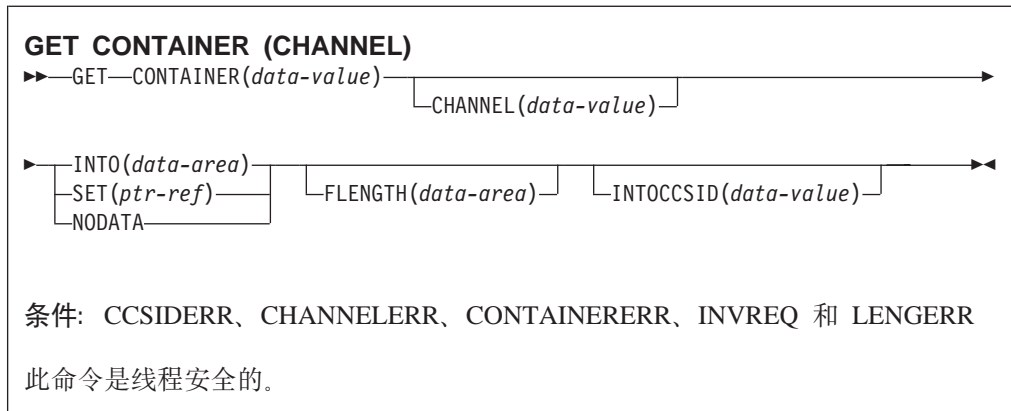
为拥有容器的通道指定名称（1 到 16 个字符）。

CONTAINER(*data-value*)

为要删除的容器指定名称（1 到 16 个字符）。

GET CONTAINER (CHANNEL):

从指定的通道容器获取数据。



描述

GET CONTAINER (CHANNEL) 读与指定的通道容器关联的数据。

保存数据的容器由名称和通道（拥有该容器的通道）来标识。可以通过以下方式识别拥有容器的通道：

- 明确地指定 CHANNEL 选项识别。
- 隐式地省略 CHANNEL 选项识别。如果省略该选项，则暗指当前通道。

选项

CHANNEL(data-value)

为拥有容器的通道指定名称（1 到 16 个字符）。

CONTAINER(data-value)

为保存要检索的数据的容器指定名称（1 到 16 个字符）。

FLENGTH(data-area)

作为输入字段， FLENGTH 以全字二进制值指定要读的数据的长度。作为输出字段， FLENGTH 返回容器中数据的长度。 FLENGTH 是输入字段还是输出字段取决于您指定 INTO、SET 或 NODATA 选项中的哪一个选项。

指定 INTO 选项

FLENGTH 既是输入字段又是输出字段。

在输入字段中， FLENGTH 指定程序接受的最大数据长度。如果所指定的值小于零，则采用值零。如果数据长度超过了指定的值，则数据长度会被截断到此值，并出现 LENGERR 条件。

如果长度可以从 INTO 变量由编译器生成，则不需要指定 FLENGTH。如果 INTO 和 FLENGTH 都指定，则 FLENGTH 指定程序接受的最大数据长度。

在输出时（即，在完成检索操作时）CICS 将数据区（如果已指定）设置为容器中数据的实际长度。如果容器保存已从一个 CCSID 转换成另一个 CCSID 的字符数据，则此长度为转换之后数据的长度。

指定 SET 或 NODATA 选项

FLENGTH 是只输出字段。必须指定它并且必须指定为数据区。

在完成检索操作时，会将数据区设置为容器中数据的实际长度。如果容器保存已从一个 CCSID 转换成另一个 CCSID 的字符数据，则此长度为转换之后数据的长度。

INTO(data-area)

指定放置检索到的数据的数据区。

INTOCCSID(data-value)

以全字十进制数指定容器中的字符数据将转换至的编码字符集标识 (CCSID)。对于 CICS Transaction Server for z/OS 应用程序，这通常是 EBCDIC CCSID。（然而，在有些情况下可以指定 ASCII CCSID，例如，在您要检索 ASCII 数据而无需将它自动转换成 EBCDIC 时。）

如果未指定 INTOCCSID，则它的值缺省为区域的 CCSID。区域的缺省 CCSID 由 LOCALCCSID 系统初始化参数指定。

只能转换字符数据，并且只有在用于在容器中放置数据的 PUT CONTAINER 命令中指定 DATATYPE of CHAR 时才执行转换。

获取有关使用通道转换数据的更多信息，请参阅 *CICS Application Programming Guide*。

要获取 CCSID 的说明和 CICS 支持的 CCSID 列表，请参阅 *CICS Family: Communicating from CICS on System/390* 手册。

NODATA

指定没有要检索的数据。使用此选项发现容器中数据的长度（以 FLENGTH 返回）。

如果发生数据转换，则字符数据的长度可能会变化。因此，如果要将字符数据转换成不同于此区域的 CCSID 的任何 CCSID，则指定 NODATA 时您还应指定 INTOCCSID。这会确保以 FLENGTH 返回已转换数据的正确长度。

SET(ptr-ref)

指定在其中返回所检索数据的地址的数据区。

在发生以下任何一种情况之前，数据区都由 CICS 来维护：

- 对于同一通道中的同一容器，带 SET 选项的后继 GET CONTAINER 命令由可访问此存储器的任何程序发出。
- 通过 DELETE CONTAINER 命令删除了容器。
- 通过 MOVE CONTAINER 命令来除去容器。
- 通道超出了程序的作用域。

当心链接到可能发出上述一个命令的其他程序。

不要发出 FREEMAIN 命令来释放此存储。

如果您的应用程序需要保留数据，则它应将数据移动到它自己的存储中。

MOVE CONTAINER (CHANNEL):

将容器（及其内容）从一个通道移动到另一个通道。

MOVE CONTAINER (CHANNEL)

```
►►—MOVE—CONTAINER(data-value)—AS(data-value)—┐
└─CHANNEL(data-value)—┘
┐
└─TOCHANNEL(data-value)—┘
```

条件: CHANNELERR, CONTAINERERR, INVREQ

此命令是线程安全的。

描述

MOVE CONTAINER (CHANNEL) 可将容器从一个通道移动到另一个通道。移动之后，源容器不再存在。

源容器和目标容器由名称和拥有它们的通道来标识。可以用以下方式标识拥有源容器的通道：

- 明确地指定 CHANNEL 选项识别。
- 隐式地省略 CHANNEL 选项识别。如果省略该选项，则暗指当前通道。

同样地，可以用以下方式标识拥有目标容器的通道：

- 明确地指定 TOCHANNEL 选项。
- 隐式地省略 TOCHANNEL 选项。如果省略该选项，则暗指当前通道。

您可以移动容器：

- 从一个通道到另一个通道。
- 在同一通道中，例如，从当前通道到当前通道。这样做的结果就相当于对容器重命名。

您可使用 MOVE CONTAINER 而不是 GET CONTAINER 和 PUT CONTAINER 作为在通道间传输数据更有效率的方法。

注：

1. 源通道必须在发出 MOVE CONTAINER 命令的程序作用域内。
2. 如果在发出 MOVE CONTAINER 命令的程序作用域内不存在目标通道，则会创建它。
3. 如果源容器不存在，则会发生错误。
4. 如果目标容器还不存在，则会创建它。如果目标容器已存在，则会覆盖它先前的内容。
5. 如果您尝试用容器自身覆盖它，则不会覆盖。即，如果您为 CONTAINER 和 AS 选项指定同一个值，并省略 CHANNEL 和 TOCHANNEL 选项或赋予它们同一个值，以便指定同一个通道，则不会更改和删除源容器。不会出现出错条件。

选项

AS(data-value)

为目标容器指定名称（1 到 16 个字符）。如果目标容器已存在，则会覆盖它的内容。

可接受的字符为 A-Z a-z 0-9 \$ @ # / % & ? ! : | " = ~ , ; < > . - 和 _。字首不允许是空白字符，并且也不允许有嵌入的空白字符。如果提供的名称少于 16 个字符，则用尾部空格填补至 16 个字符。

容器名总是 EBCDIC 格式的。上面列出的容器名可用字符集中包括的某些字符在所有 EBCDIC 代码页中的表示并不一致。因此我们建议，如果容器是要在不同的区域间提供的，那么用于命名它们的字符应该限制为 A-Z 0-9 & : = , ; < > . - 和 _。

CHANNEL(data-value)

为拥有源容器的通道指定名称（1 到 16 个字符）。如果未指定此选项，则暗指当前通道。

CONTAINER(data-value)

为要移动的源容器指定名称（1 到 16 个字符）。

TOCHANNEL(data-value)

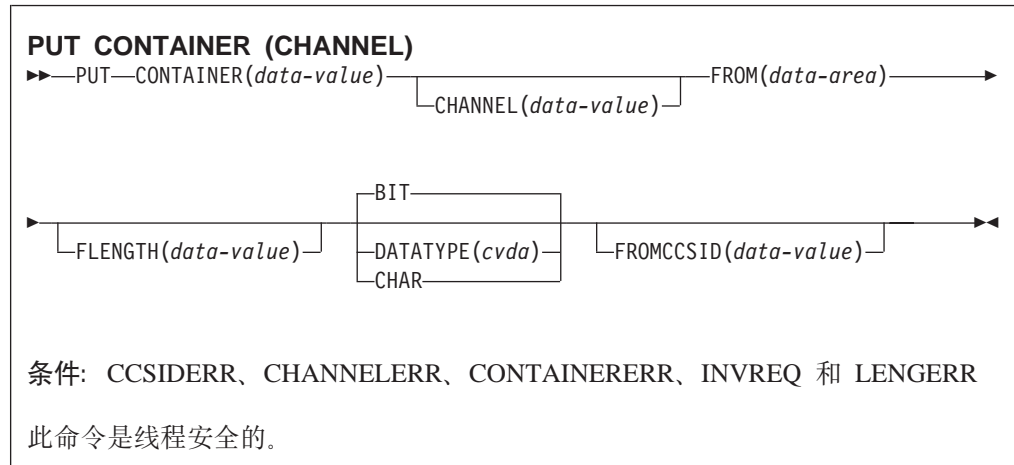
为拥有目标容器的通道指定名称（1 到 16 个字符）。如果您要指定新的通道，则记住可以接受的字符是 A-Z a-z 0-9 \$ @ # / % & ? ! : | " = ~ , ; < > . - 和 _。字首不允许是空白字符，并且也不允许有嵌入的空白字符。如果提供的名称少于 16 个字符，则用尾部空格填补至 16 个字符。

通道名总是 EBCDIC 格式的。上面列出的通道名可用字符集中包括的某些字符在所有 EBCDIC 代码页中的表示并不一致。因此我们建议，如果要在不同的区域间提供通道，那么用于命名它们的字符应该限制为 A-Z 0-9 & : = , ; < > . - 和 _。

如果未指定此选项，则暗指当前通道。

PUT CONTAINER (CHANNEL):

将数据放入指定的通道容器。



描述

PUT CONTAINER (CHANNEL) 将数据放入与指定的通道关联的容器。

通过名称识别容器。可以通过以下方式识别拥有容器的通道:

- 明确地指定 CHANNEL 选项识别。
- 隐式地省略 CHANNEL 选项识别。如果省略该选项, 则暗指当前通道。

注:

1. 不限制可以与通道关联的容器数目。
2. 各个容器的大小仅受可用存储量的限制。

注意:

小心, 不要创建太多大容器, 这样会限制其他应用程序可以使用的存储量。

3. 如果指定的容器还不存在, 则创建它。如果指定的容器已存在, 则将覆盖先前的内容。
4. 如果指定的通道还不存在, 则创建它。

选项

CHANNEL(data-value)

指定拥有容器的通道的名称 (1 到 16 个字符)。可以接受的字符是 A-Z a-z 0-9 \$ @ # / % & ? ! : | " = , ; < > . - 和 _。字首不允许是空白字符, 并且也不允许有嵌入的空白字符。如果提供的名称少于 16 个字符, 则用尾部空格填补至 16 个字符。

通道名总是 EBCDIC 格式的。上面列出的通道名可用字符集中包括的某些字符在所有 EBCDIC 代码页中的表示并不一致。因此我们建议, 如果要在不同的区域间提供通道, 那么用于命名它们的字符应该限制为 A-Z 0-9 & : = , ; < > . - 和 _。

CONTAINER(data-value)

指定要放置数据的容器的名称 (1 到 16 个字符)。

可接受的字符为 A-Z a-z 0-9 \$ @ # / % & ? ! : | " = , ; < > . - 和 _。字首不允许是空白字符, 并且也不允许有嵌入的空白字符。如果提供的名称少于 16 个字符, 则用尾部空格填补至 16 个字符。

不要使用以“DFH”开头的容器名，除非 CICS 要求您这样做。

容器名总是 EBCDIC 格式的。上面列出的容器名可用字符集中包括的某些字符在所有 EBCDIC 代码页中的表示并不一致。因此我们建议，如果容器是要在不同的区域间提供的，那么用于命名它们的字符应该限制为 A-Z 0-9 & : = , ; < > . - 和 _。

DATATYPE(cvda)

指定要放入容器的数据类型。此选项仅适用于新的容器。如果容器已存在，则在创建容器时确定它的数据类型，并且这是无法更改的。CVDA 值有：

BIT 位数据。无法转换容器中的数据。这是缺省值，除非指定 FROMCCSID。

CHAR 字符数据。若有必要，容器中的数据会转换成创建通道的应用程序的代码页。如果通道由基于 ASCII 的系统上的客户机应用程序创建，那么它将是 ASCII 代码页。如果它是由 CICS Transaction Server for z/OS 应用程序创建的，则它将是 EBCDIC 代码页。仅当客户机和服务器程序在不同的平台上运行时，转换才是必需的。

容器中的所有数据都会被转换成类似单个字符串的形式。对于 SBCS 代码页，由若干字符字段组成的结构与单字节字符串等价。然而，对于 DBCS 代码页，并不是这么回事。如果您使用 DBCS 代码页，要确保数据转换正确执行，您必须将每个字符串放入单独的容器。

如果容器包含字符数据并且通道将从 CICS Transaction Server for z/OS 传递到 ASCII 系统，则必须指定 DATATYPE 为 CHAR。如果容器包含二进制数据，或者通道不会传递到 ASCII 系统，则 DATATYPE 是可选的参数。

不可能通过 PUT CONTAINER 命令更改现有容器的数据类型。例如，如果用数据类型 BIT 创建容器，并且后继 PUT CONTAINER 命令为同一个容器指定数据类型为 CHAR，则会产生一个 INVREQ 条件。如果您确实需要用一个不同的数据类型替换现有的容器，则首先必须显式地删除该现有的容器。

获取有关使用通道转换数据的更多信息，请参阅 *CICS Application Programming Guide*。

FLENGTH(data-value)

用全字二进制值指定将从中读取数据的数据区的长度。

FROM(data-area)

指定提供写入容器的数据的数据区。

FROMCCSID(data-value)

以全字十进制数指定，要放入容器的字符数据的当前编码字符集标识 (CCSID)。对于 CICS Transaction Server for z/OS 应用程序，这通常是 EBCDIC CCSID。（然而，如果您要传递 ASCII 数据，也可能指定 ASCII CCSID）。

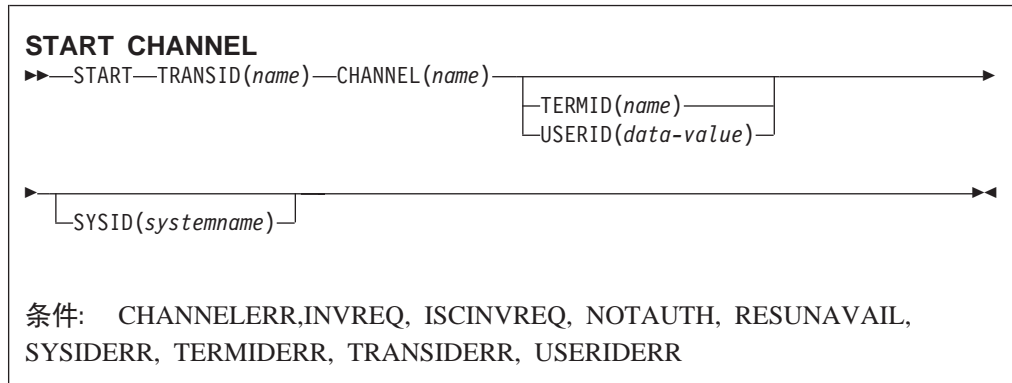
如果指定 FROMCCSID，则意指 DATATYPE(DFHVALUE(CHAR))。

如果未指定 FROMCCSID，那么它的值缺省为区域的 CCSID。区域的缺省 CCSID 由 LOCALCCSID 系统初始化参数指定。

要获取 CCSID 的说明和 CICS 支持的 CCSID 列表，请参阅 *CICS Family: Communicating from CICS on System/390* 手册。

START CHANNEL:

启动任务并传递给它一个通道。



描述

START CHANNEL 在本地或远程系统上启动一个任务，并传递给它一个通道。

通常，启动任务使用通道向已启动的任务传递数据（尽管在有些情况下通道可能是空的，请参阅 CHANNEL 选项的描述）。启动任务还可以指定已启动的任务要使用的终端作为它的主要功能。

例如，已启动的任务可以：

1. 使用 ASSIGN CHANNEL 命令发现已传递给它的通道的名称
2. 使用 STARTBROWSE CONTAINER CHANNEL 和 GETNEXT CONTAINER 命令浏览通道中的容器
3. 使用 GET CONTAINER CHANNEL 命令访问容器中的数据

在可以启动要执行的事务之前必须满足某些约束，如下所示：

- 如果指定 TERMID 选项，则指定的终端必须存在并且是可用的。如果指定的终端不存在，则会废弃 START。
- START CHANNEL 不支持 IMS，即，您无法使用 START CHANNEL 启动远程 IMS 系统上的事务。

每个 START CHANNEL 命令都会导致启动一个单独的任务。

由 START 命令启动的动态路由事务

某些通过 START 命令的子集启动的事务可以动态地路由到远程区域。

不带异常条件的 START 失败

在以下一些情况下执行 START 命令没有错误，但已启动的任务从不发生：

- 在 CICS 尝试创建任务时事务或它的初始程序是禁用的。
- START 指定在 CICS 尝试创建任务时未定义（并且无法通过 XICTENF 或 XALTENF 出口找到）的终端。
- 如果执行 START 时所请求的终端不存在，则您获得 TERMIDERR 条件。然而，如果随后此终端变得不可用，好象发生了用户注销，则会废弃您的 START 请求并且不会发生 TERMIDERR。

这些曝光是由于 START 的执行和任务创建间的时间延迟而导致的。即使在 START CHANNEL 请求中，当总是立即执行 START 时，CICS 也可能延迟创建任务，这是因为所需的终端在忙或因为其他系统约束。

您可使用 INQUIRE 命令确保在执行 START 命令时启用事务和程序，但在任务创建之前它们中的任何一个可能会变成已禁用。

选项

CHANNEL(*name*)

为用于已启动任务的通道指定名称（1 到 16 个字符）。可以接受的字符是 A-Z a-z 0-9 \$ @ # / % & ? ! : | " = ~ , ; < > . - 和 _。字首不允许是空白字符，并且也不允许有嵌入的空白字符。如果提供的名称少于 16 个字符，则用尾部空格填补至 16 个字符。

通道名总是 EBCDIC 格式的。上面列出的通道名可用字符集中包括的某些字符在所有 EBCDIC 代码页中的表示并不一致。因此我们建议，如果要在不同的区域间提供通道，那么用于命名它们的字符应该限制为 A-Z 0-9 & : = , ; < > . - 和 _。

发出 START 命令的程序可能：

- 已通过一个或多个 PUT CONTAINER CHANNEL 命令创建通道
- 按名称指定它的当前通道
- 命名一个不存在的通道，在这种情况下会创建一个新的、空的通道

会给予已启动的任务通道的容器（和它们包含的数据）的副本。当发出 START 命令时制作副本。

SYSID(*systemname*)

为请求导向所至的系统指定名称。

TERMID(*name*)

为主要功能指定符号标识（1 到 4 个字母字符），而此主要功能与因 START 命令而启动的事务关联。此主要功能可以是终端（通常情况）或 APPC 会话。在指定 APPC 会话的情况下，使用连接（或方式集）名而不是终端标识。在要启动的事务必须与终端通信时，此选项是必需的；否则应省略它。

必须将终端标识定义为执行 START 命令的系统上的本地终端或远程终端。

TRANSID(*name*)

为事务指定符号标识（1 到 4 个字符），而该事务是由因 START 命令而启动的任务执行的。

如果指定 SYSID，并对远程系统命名，则无论在 PCT 中是否将事务定义定义为远程，假设该事务在远程系统上。否则，使用事务定义查明该事务在本地系统上还是在远程系统上。

USERID(*data-value*)

指定籍以运行已启动事务的权限的用户标识（如果已启动的事务与终端不关联，即，未指定 TERMID）。它指 *userid1*。

如果 TERMID 和 USERID 您都省略，则 CICS 改为使用发出 START 命令的事务籍以运行的用户标识。它指 *userid2*。

使用 *userid1* 或 *userid2*，CICS 确保已启动的事务总是以有效的用户标识运行，且必须授权此用户标识以访问已启动的事务引用的所有资源。

CICS 对 *userid2* 执行代理安全性检查以验证是否授予此用户 *userid1* 的权限。如果未授权 *userid2*，则 CICS 返回 NOTAUTH 条件。如果省略 USERID，则此处不执行代理检查。

已修改的 API 命令

已修改以下应用程序编程命令：

EXEC CICS ASSIGN

CHANNEL 选项已添加，STARTCODE 选项已更改：

选项

CHANNEL(*data-area*)

返回程序的当前通道（如果存在）1 到 16 个字符的名称；否则为空白。

STARTCODE(*data-area*)

返回 2 个字符的值，表明如何启动发出请求的事务。更改的值为：

Code 事务启动自

S FROM 选项中未传递数据的 START 命令。它可能或可能不传递通道。

SD FROM 选项中传递数据的 START 命令。

EXEC CICS DELETE CONTAINER (BTS)

没有语法更改。已更改了描述以强调此命令仅应用于 BTS，而不应用于通道和容器。

EXEC CICS ENDBROWSE CONTAINER

没有语法更改，但此命令现在可以与通道以及 BTS 和容器协同使用。

EXEC CICS GET CONTAINER (BTS)

在 INVREQ 条件中有一个新 RESP2 值：

条件

INVREQ

RESP2 值：

2 在未指定 CHANNEL 选项的情况下指定 INTOCCSID 选项，并且没有当前通道（因为发出此命令的程序未传递通道。）只有在（显式地或隐式地）指定通道的 GET CONTAINER 命令中 INTOCCSID 才是有效的。它在 GET CONTAINER (BTS) 命令中无效。

EXEC CICS GETNEXT CONTAINER

没有语法更改，但此命令现在可以与通道以及 BTS 和容器协同使用。

EXEC CICS HANDLE ABEND

如果存在当前通道，则将其传递给指定用来处理异常终止的程序；或如果已建立通信区（COMMAREA），则将通信区传递给指定用来处理异常终止的程序。先前仅可以向它传递 COMMAREA。

EXEC CICS LINK PROGRAM

添加了 CHANNEL 选项和 CHANNELERR 条件：

选项

CHANNEL(name)

为所调用程序可用的通道指定名称（1 到 16 个字符）。可以接受的字符是 A-Z a-z 0-9 \$ @ # / % & ? ! : | " = ~ , ; < > . - 和 _。不允许空白字符出现在开头和中间。如果所提供的名称少于 16 个字符，则在尾部填充空格以使其达到 16 个字符。

通道名的格式通常是 EBCDIC。以上列出的通道名的允许字符集包括的某些字符在所有 EBCDIC 代码页中的表示并不一致。因此我们建议，如果要在区域间装运通道，则应将用于命名它们的字符限制为 A-Z 0-9 & : = , ; < > . - 和 _。

发出 LINK 命令的程序可能：

- 已通过一个或多个 PUT CONTAINER CHANNEL 命令创建通道
- 按名称指定它的当前通道
- 命名一个不存在的通道，在这种情况下会创建一个新的、空的通道

条件

CHANNELERR

RESP2 值：

- 1 CHANNEL 选项中指定的名称包含非法的字符或字符组合。

EXEC CICS MOVE CONTAINER (BTS)

没有语法更改。已更改了描述以强调此命令仅应用于 BTS，而不应用于 CHANNEL 和容器。

EXEC CICS PUT CONTAINER (BTS)

INVREQ 条件中有两个新 RESP2 值。

条件

INVREQ

RESP2 值：

- 1 在未指定 CHANNEL 选项的情况下指定 DATATYPE 选项，并且没有当前通道（因为发出此命令的程序未传递通道。）只有在（显式地或隐式地）指定通道的 PUT CONTAINER 命令中 DATATYPE 才是有效的。它在 PUT CONTAINER (BTS) 命令中无效。
- 2 在未指定 CHANNEL 选项的情况下指定 FROMCCSID 选项，并且没有当前通道（因为发出此命令的程序未传递通道。）只有在（显式地或隐式地）指定通道的 PUT CONTAINER 命令中 FROMCCSID 才是有效的。它在 PUT CONTAINER (BTS) 命令中无效。

EXEC CICS RETURN

添加了 CHANNEL 选项和 CHANNELERR 条件：

选项

CHANNEL(name)

为接收控制的下一个程序可用的通道指定名称（1 到 16 个字符）。可以接受的字符是 A-Z a-z 0-9 \$ @ # / % & ? ! : | " = ~ , ; < > . - 和 _。不允许空白字符出现在开头和中间。如果所提供的名称少于 16 个字符，则在尾部填充空格以使其达到 16 个字符。

通道名的格式通常是 EBCDIC。以上列出的通道名的允许字符集包括的某些字符在所有 EBCDIC 代码页中的表示并不一致。因此我们建议，如果通道要在不同的区域间提供（即，如果 TRANSID 选项上指定的事务为远程），那么用于命名它的字符应该限制为 A-Z 0-9 & : = , ; < > . - 和 _。

发出 RETURN 命令的程序可能：

- 已通过一个或多个 PUT CONTAINER CHANNEL 命令创建通道
- 按名称指定它的当前通道
- 命名一个不存在的通道，在这种情况下会创建一个新的、空的通道

只有在由最高逻辑级别的程序（即，向 CICS 返回控制的程序）发出的 RETURN 命令中，此选项才是有效的。

条件

CHANNELERR

RESP2 值：

- 1 CHANNEL 选项中指定的名称包含非法的字符或字符组合。

EXEC CICS STARTBROWSE CONTAINER

添加了 CHANNEL 选项和 CHANNELERR 条件：

选项

CHANNEL(*data-value*)

为要浏览其容器的通道指定名称（1 到 16 个字符）。此名称必须是当前通道的名称或由发出 STARTBROWSE CONTAINER 命令的程序创建的通道的名称。

如果未指定此选项，并且当前上下文是通道，则会浏览当前通道的容器。

条件

CHANNELERR

RESP2 值：

- 2 找不到 CHANNEL 选项中指定的通道。

EXEC CICS XCTL

添加了 CHANNEL 选项和 CHANNELERR 条件：

选项

CHANNEL(*name*)

为所调用程序可用的通道指定名称（1 到 16 个字符）。可以接受的字符是 A-Z a-z 0-9 \$ @ # / % & ? ! : | " = ~ , ; < > . - 和 _。不允许空白字符出现在开头和中间。如果所提供的名称少于 16 个字符，则在尾部填充空格以使其达到 16 个字符。

通道名的格式通常是 EBCDIC。以上列出的通道名的允许字符集包括的某些字符在所有 EBCDIC 代码页中的表示并不一致。因此我们建议，如果要在区域间装运通道，则应将用于命名它们的字符限制为 A-Z 0-9 & : = , ; < > . - 和 _。

发出 XCTL 命令的程序可能：

- 已通过一个或多个 PUT CONTAINER CHANNEL 命令创建通道
- 按名称指定它的当前通道
- 命名一个不存在的通道，在这种情况下会创建一个新的、空的通道

条件

CHANNELERR

RESP2 值:

- 1 CHANNEL 选项中指定的名称包含非法的字符或字符组合。

对 JCICS API 的更改

注: 当编写 CICS 企业 bean 时, 您可使用 JCICS 命令 (包括通道相关的命令和容器相关的命令)。然而, CICS 不支持通道在 IIOP 请求流上传输。例如, 这意味着您无法将通道传递到远程区域上的企业 bean。

新 JCICS 类

已引入了以下新 JCICS 类:

- `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`

已修改的 JCICS 类

更改了以下 JCICS 类:

- `Program`
- `StartRequest`
- `Task`
- `TerminalPrincipalFacility`

对全局用户出口的更改

全局用户出口程序不能访问应用程序创建的容器。然而, 它们可以创建它们自己的通道并将通道传递给它们调用的程序。

对以下出口进行的较少更改在 *CICS Customization Guide* 中有所描述:

- XFCAREQ
- XFCAREQC
- XFCREQ
- XFCREQC
- XICEREQ
- XICEREQC
- XNQEREQ
- XNQEREQC
- XPCREQ
- XPCREQC
- XTDEREQ
- XTDEREQC
- XTSEREQ
- XTSEREQC

对任务相关的用户出口的更改

任务相关的用户出口程序（TRUE）不能访问应用程序创建的容器。然而，它们可以创建它们自己的通道并将通道传递给它们调用的程序。

对用户可替代程序的更改

用户可替代程序不能访问应用程序创建的容器。然而，它们可以创建它们自己的通道并将通道传递给它们调用的程序。

动态和分布式路由程序

对传递给动态和分布式路由程序的 DFHDYPDS 通信区进行了一些更改。

DYRACMAA（现有字段：动态路由程序）

此字段适用于以下内容的路由：

- 终端启动的事务
- 终端相关 START 命令启动的事务
- 程序链接（DPL）请求

对于这些类型的请求的路由，DYRACMAA 包含下列某种情况：

- 如果用户应用程序使用通信区（COMMAREA），则为应用程序的 COMMAREA 的 31 位地址
- 如果用户应用程序使用通道，并且已在通道内创建名为 DFHROUTE 的容器，则为 DFHROUTE 容器的 31 位地址
- 如果用户应用程序没有 COMMAREA 和 DFHROUTE 容器，则为空字符

对于所有其他类型的请求路由，DYRACMAA 包含空字符。

对于上述列出的三种类型合格请求的路由，如果用户应用程序使用 COMMAREA：

- 当调用动态路由程序进行路由（DYRFUNC=0）时，地址是输入通信区（如果有的话）的地址。同样，当由于路由选择错误或用于通知而调用动态路由程序（分别 DYRFUNC=1 或 3）时，地址是输入通信区的地址。
- 当因为先前路由的事务或链接请求已正常终止而调用路由程序（DYRFUNC=2）时，地址是输出通信区（如果有的话）的地址。已路由的应用程序可使用其输出通信区，将信息传递给动态路由程序。

当因为已路由的事务异常终止而调用路由程序（DYRFUNC=4）时，通信区或 DFHROUTE 容器中的信息没有意义。

您的路由程序可改变任何应用程序的通信区或 DFHROUTE 容器中的数据，这些数据由 DYRACMAA 处理。

DYRACMAA（现有字段：分布式路由程序；无更改）

未由分布式路由程序使用。在调用时，它设置为零。

DYRACMAL（现有字段：动态路由程序）

此字段适用于以下内容的路由：

- 终端启动的事务
- 终端相关 START 命令启动的事务
- 程序链接（DPL）请求

对于这些类型的请求的路由，DYRACMAL 包含下列某个数值：

- 如果用户应用程序使用 COMMAREA，为应用程序的 COMMAREA 的长度，以字节为单位
- 如果用户应用程序使用通道，并且已在通道内创建名为 DFHROUTE 的容器，则为 DFHROUTE 容器中的数据长度，以字节为单位
- 如果用户应用程序没有 COMMAREA 和 DFHROUTE 容器，则为零

对于所有其他类型的请求路由，DYRACMAL 包含零。

DYRACMAL (现有字段：分布式路由程序；无更改)

未由分布式路由程序使用。在调用时，它设置为零。

DYRCHANL (新字段)

是与程序链接或 START 命令关联的通道（如果有的话）的名称。此字段只应用于路由 DPL 请求、与终端无关的 START 请求和与终端相关的 START 请求启动的事务。对于其他的请求类型，或如果没有与此命令关联的通道，则此字段不包含任何内容。

注意，因为路由程序被赋予通道的名称，而不是它的地址，所以路由程序无法使用 DYRCHANL 字段查看或更改通道容器的内容。要获得有关路由程序如何检查或更改应用程序容器内容的信息，请参阅第 183 页的『与通道一起动态路由』和 DYRACMAA 字段的描述。

DYRLEVEL (现有字段)

是目标 AOR 中成功处理路由请求所需的 CICS 级别。添加了新值：

X'03' CICS TS for z/OS V3.1) 当前可能只为以下请求设置：

- 具有关联通道的 DPL 请求
- 具有关联通道的 START 请求
- 企业 bean 和 CORBA 无状态对象的方法请求

DYRTYPE (现有字段)

包含能调用路由程序的请求的类型。添加了三个新值：

9 带通道的程序链接请求。

调用动态路由程序以路由此请求。

A 通过与终端相关的 EXEC CICS START 命令启动的事务，其中有一个与 START 关联的通道。

调用动态路由程序以路由此事务。

B 带通道的非终端相关的 START 请求。

调用分布式路由程序以路由此请求。

下列现有值的含义已更改。（对现有描述的更改由**粗体**表明）：

2 通过终端相关的 EXEC CICS START 命令启动的事务，其中没有数据并且没有与 **START** 关联的通道。

3 通过终端相关的 EXEC CICS START 命令启动的事务，其中有与 START 关联的数据但没有通道。

4 不带通道的程序链接请求。

6 非终端相关的 START 请求，带有或不带数据但不带通道。

DYRVER (现有字段)

是动态路由程序界面的版本号。对于 CICS Transaction Server for z/OS V3R1, 此版本号为“10”。

对监控的更改

将一些新的字段添加到了性能类监控记录。除了 DFHPROG 和 DFHTASK 组, 还有一个新组 DFHCHNL。

通过在监控控制表 (MCT) 中对 DFHMCT TYPE=RECORD 条目进行编码, 可以将所有新字段从监控记录中排除。

组 DFHCHNL 中的性能数据

组 DFHCHNL 包含以下性能数据:

321 (TYPE-A, 'PGTOTCCT', 4 BYTES)

用户任务发出的通道容器的 CICS 请求数。

322 (TYPE-A, 'PGBRWCCT', 4 BYTES)

用户任务发出的通道容器的 CICS 浏览请求数。

323 (TYPE-A, 'PGGETCCT', 4 BYTES)

用户任务发出的通道容器的 GET CONTAINER 请求数。

324 (TYPE-A, 'PGPUTCCT', 4 BYTES)

用户任务发出的通道容器的 PUT CONTAINER 请求数。

325 (TYPE-A, 'PGMOVCCT', 4 BYTES)

用户任务发出的通道容器的 MOVE CONTAINER 请求数。

326 (TYPE-A, 'PGGETCDL', 4 BYTES)

用户任务发出的所有 GET CONTAINER CHANNEL 命令的容器中数据总字节长度。

327 (TYPE-A, 'PGPUTCDL', 4 BYTES)

用户任务发出的所有 PUT CONTAINER CHANNEL 命令的容器中数据总字节长度。

组 DFHPROG 中的性能数据

以下新字段已添加到组 DFHPROG:

286 (TYPE-A, 'PCDLCSDL', 4 BYTES)

用户任务 (带 CHANNEL 选项) 发出的所有分布式程序链接 (DPL) 请求的容器中数据总字节长度。此总字节长度包括数据的任何头的长度。

287 (TYPE-A, 'PCDLRDL', 4 BYTES)

用户任务发出的所有 DPL RETURN CHANNEL 命令的容器中数据总字节长度。此总字节长度包括数据的任何头的长度。

306 (TYPE-A, 'PCLNKCCT', 4 BYTES)

用户任务 (带 CHANNEL 选项) 发出的本地程序 LINK 请求数。

307 (TYPE-A, 'PCXCLCCT', 4 BYTES)

用户任务 (带 CHANNEL 选项) 发出的程序 XCTL 请求数。

308 (TYPE-A, 'PCDPLCCT', 4 BYTES)

用户任务（带 CHANNEL 选项）发出的程序分布式程序链接（DPL）请求数。

309 (TYPE-A, 'PCRTNCCT', 4 BYTES)

用户任务（带 CHANNEL 选项）发出的远程伪会话 RETURN 请求数。

310 (TYPE-A, 'PCRTNCDL', 4 BYTES)

用户任务发出的所有远程伪会话 RETURN CHANNEL 命令的容器中数据总字节长度。此总字节长度包括数据的任何头的长度。

组 DFHTASK 中的性能数据

已将以下新字段添加到组 DFHTASK:

065 (TYPE-A, 'ICSTACCT', 4 BYTES)

用户任务（带 CHANNEL 选项）发出的本地时间间隔控制 START 请求总数。

345 (TYPE-A, 'ICSTACDL', 4 BYTES)

用户任务发出的、所有本地执行的 START CHANNEL 请求的容器中数据总字节长度。此总字节长度包括数据的任何头的长度。

346 (TYPE-A, 'ICSTRCCT', 4 BYTES)

用户任务发出的、在远程系统上执行的时间间隔控制 START CHANNEL 请求总数。

347 (TYPE-A, 'ICSTRCDL', 4 BYTES)

用户任务发出的、所有远程执行的 START CHANNEL 请求的容器中数据总字节长度。此总字节长度包括数据的任何头的长度。

对统计信息的更改

跨连接流动的通道数据具有新的统计信息。这些统计信息由 DFHA14DS DSECT 映射。它们显示在 DFHSTUP“ISC/IRC 系统条目：资源统计信息”报告，以及“连接和方式名报告”中。

表 5. “连接和模式名报告”中的新字段

字段标题	描述
终端共享通道请求	带通道的终端共享请求数。 源字段: A14ESTTC_CHANNEL
终端共享通道请求: 已发送的字节	终端共享通道请求发送的字节数。它是连接发送的数据总数, 包括任何控制信息。 源字段: A14ESTTC_CHANNEL_SENT
终端共享通道请求: 已接收的字节	终端共享通道请求已接收的字节数。它是连接接收的数据总数, 包括任何控制信息。 源字段: A14ESTTC_CHANNEL_RCVD
带通道的程序控制 LINK 请求	分布式程序链接 (DPL) EXEC CICS LINK CHANNEL 命令数, 跨连接进行功能发送。 源字段: A14ESTPC_CHANNEL

表 5. “连接和模式名报告”中的新字段 (续)

字段标题	描述
LINK 通道请求: 已发送的字节	LINK 通道请求发送的字节数。它是连接发送的数据总数, 包括任何控制信息。 源字段: A14ESTPC_CHANNEL_SENT
LINK 通道请求: 已接收的字节	LINK 通道请求接收的字节数。它是连接接收的数据总数, 包括任何控制信息。 源字段: A14ESTPC_CHANNEL_RCVD
带通道的时间间隔控制 START 请求	带通道的时间间隔控制 START 请求数, 跨连接进行功能发送。 源字段: A14ESTIC_CHANNEL
START 通道请求: 已发送的字节	START 通道请求已发送的字节数。它是连接发送的数据总数, 包括任何控制信息。 源字段: A14ESTIC_CHANNEL_SENT
START 通道请求: 已接收的字节	START 通道请求接收的字节数。它是连接接收的数据总数, 包括任何控制信息。 源字段: A14ESTIC_CHANNEL_RCVD

对样本程序的更改

已对 IIOB 银行帐户样本应用程序 (事务 BNKQ) 的两个 C 语言程序进行了修改, 以显示如何在 EXEC CICS LINK 和 RETURN 命令中传递通道。已对顶层程序 DFH\$IIBQ 和它所链接的程序 DFH\$IICC 进行了修改, 以使用通道 / 容器模型而不是 COMMAREA 模型。代码中的注释显示已做过的更改。

对问题确定的更改

对问题确定的更改

消息

增强了程序间数据传输后, 引入了新的消息。所有新增和更改的消息在 *CICS Messages and Codes* 手册。

异常终止代码

引入了一些新的异常终止代码, 并除去了一些现有代码。

新异常终止代码:

AEYF (仅 *CICS TS for z/OS V3.1*) :

说明

CICS 的存储违例。

事务请求 CICS 访问事务本身不能访问的存储区域。当使用 PUT CONTAINER 或 GET CONTAINER 命令，将无效存储区域传递给 CICS 时，发生此问题。在下列情况下，会发生错误：

- 未正确指定 FROM 或 INTO 地址。
- FLENGTH 值指定的值足够导致区域包含事务不能访问的存储。

此错误的常见原因是使用 FLENGTH 参数指定半字区域的地址，它需要的是全字区域。当先前使用 COMMAREA（其具有半字长度）的程序已修改为使用容器（具有全字长度），会发生此错误。

系统操作

事务已异常终止，并进行 CICS 事务转储。

用户响应

检查跟踪以找到条目为 DFHEISR 的跟踪条目，然后标识错误的参数。如果处理异常终止，EXEC CICS ASSIGN ASRASTG、ASRAKEY、ASRASPC 和 ASRAREGS 给出有关此异常终止的其他信息。在异常终止时，将 2 点注册到发生故障的存储区域。

您将极有可能需要执行下列操作：

- 改正发出 EXEC CICS PUT CONTAINER 或 EXEC CICS GET CONTAINER 命令时发生错误的程序。确保它提供有效存储区域的地址，并且提供的 FLENGTH 能令事务可以访问存储区域的所有部分。确保 FLENGTH 引用全字长度。

您可能还需要考虑更改下列一个或多个情况：

- 如果存储保护是活动的，在发出 EXEC CICS 命令的程序的 CEDA 定义上，将 EXECKEY 选项从 USER 更改为 CICS。
- 如果存储保护是活动的，将事务定义上的 TASKDATAKEY 属性从 CICS 更改为 USER。
- 如果事务隔离是活动的，将事务定义上的 ISOLATE 属性从 YES 更改为 NO。

模块

DFHSRP

AITI (仅 CICS TS for z/OS V3.1) :

说明

当尝试从连接的 CICS 系统接收数据或将数据发送到 CICS 系统时，处理 START CHANNEL 或 LINK CHANNEL 请求的镜像事务失败。因为一个通道可能包括相当大的数据量，所以可能需要对终端控制进行许多次调用以传输通道数据。DFHMIRS 调用程序 DFHAPCR 以执行所有通道数据的系统间传输。终端控制在这些某个调用中已检测到错误。此错误可能是读超时，也可能是流中使 CICS 无法正确地处理数据的更为严重的错误。

系统操作

事务已终止。镜像任务已异常终止，并进行 CICS 事务转储。

用户响应

如果此错误是超时，则确定远程区域未响应的原因。检查跟踪以确定 GETMAIN 失败的原因。如果 CICS 区域存储不足，则采取必要的步骤以改正此问题。如果区域的存储是充足的，则您可能需要 IBM 的帮助来解决此问题。

模块

DFHADDRM

AXGA (仅 CICS TS for z/OS V3.1) :

说明

程序 DFHAPCR 已返回意外的响应。DFHAPCR 执行以下功能:

- 抽出构成通道的所有容器的内容并将它们传输到远程系统。
- 从远程系统接收的入站数据重新创建通道和容器。

抽出或重新创建通道数据时，DFHAPCR 已检测到入站数据中的错误或接收到意外的响应。

系统操作

事务已异常终止，并进行 CICS 事务转储。

用户响应

查找任何相关 CICS 消息和异常终止以确定在管理容器的程序管理器中是否有先前的故障。查找程序管理器或 DFHAPCR 中的异常跟踪条目以确定错误的原因。

模块

DFHXTP

AXTS (仅 CICS TS for z/OS V3.1) :

说明

已尝试在伪会话中的事务间传递通道和容器数据，但伪会话中的下一个事务驻留在不支持通道和容器的 CICS 区域中。

系统操作

事务已异常终止，并进行 CICS 事务转储。

用户响应

如果您的应用程序使用通道和容器在伪会话中的事务间传递数据，则确保伪会话中的所有事务都驻留在 CICS TS for z/OS V3.1 或以上版本的区域中。

模块

DFHXTP

AXTU (仅 CICS TS for z/OS V3.1) :

说明

程序 DFHAPCR 已返回意外的响应。DFHAPCR 执行以下功能:

- 抽出构成通道的所有容器的内容并将它们传输到远程系统。
- 从远程系统接收的入站数据重新创建通道和容器。

抽出或重新创建通道数据时, DFHAPCR 已检测到入站数据中的错误或接收到意外的响应。

系统操作

事务已异常终止, 并进行 CICS 事务转储。

用户响应

查找任何相关 CICS 消息和异常终止以确定在管理容器的程序管理器中是否有先前的故障。查找程序管理器或 DFHAPCR 中的异常跟踪条目以确定错误的原因。

模块

DFHXTP

其他异常终止代码:

出于兼容性目的, 下列新异常终止代码已添加到 CICS TS for z/OS V2.2 和 CICS TS for z/OS V2.3 中。它们不应用于 CICS TS for z/OS V3.1。将来, 如果您将 CICS TS for z/OS V2.2 或 CICS TS for z/OS V2.3 区域迁移到 CICS TS for z/OS V3.1, 则了解这些异常终止代码不会在 CICS TS for z/OS V3.1 中发生。

- AXF9
- AXTT

跟踪

与新功能相关的 CICS 跟踪点是 AP 0785 和 AP 4E20—AP 4E22。

要控制 CICS 跟踪信息的输出, 以常规方式使用 CICS 跟踪控制。

从 COMMAREAs 迁移到通道

现有功能的迁移

- 使用传统通信区 (COMMAREAS) 交换数据的 CICS 应用程序如以前一样工作。
- 如果您使用用户编写的动态或分布式路由程序进行工作负载管理, 而不是使用 CICSplex SM, 则您必须修改您的程序以处理可能在 DFHDYPDS 通信区的 DYRLEVEL、DYRTYPE 和 DYRVER 字段中传递的新值, 请参阅第 201 页的『动态和分布式路由程序』。

迁移到新功能

此部分描述您可以如何迁移几种现有应用程序类型以使用通道和容器而不是通信区 (COMMAREA)。

可能由通道使用单个容器替换 COMMAREA。当此方法看上去是从 COMMAREA 移动到通道和容器的最简单方法时，它不是这样做的最佳实践。因为您花费时间更改应用程序以开发此新功能，所以您应该实施通道和容器的“最佳实践” – 请参阅第 176 页的『设计通道: 最佳实践』。通道比起 COMMAREA 有几个优点（请参阅第 159 页的『通道的优点』），设计您的通道以充分利用这些优点是值得的。

迁移传递 COMMAREA 的 LINK 命令

要迁移在 LINK 命令中使用 COMMAREA 交换结构的两个程序，更改表 6 中显示的指令。

表 6. 迁移传递 COMMAREA 的 LINK 命令

程序	迁移之前	迁移之后
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：在 COMMAREA 示例中，已将数据放入 COMMAREA 的 PROG2 仅发出 RETURN 命令以将数据返回给 PROG1。在通道示例中，要返回数据 PROG2 必须在发出 RETURN 前发出 PUT CONTAINER 命令。

迁移传递 COMMAREA 的 XCTL 命令

要迁移在 XCTL 命令中使用 COMMAREA 传递结构的两个程序，更改表 7 中显示的指令。

表 7. 迁移传递 COMMAREA 的 XCTL 命令

程序	迁移之前	迁移之后
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) ...

迁移 RETURN 命令中的伪会话 COMMAREA

要迁移使用 COMMAREA 交换结构作为伪会话一部分的两个程序，更改第 210 页的表 8 中显示的指令。

表 8. 迁移 RETURN 命令中的伪会话 COMMAREA

程序	迁移之前	迁移之后
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)

迁移 START 数据

要迁移使用 START 数据交换结构的两个程序，更改表 9 中显示的指令。

表 9. 迁移 START 数据

程序	迁移之前	迁移之后
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)

注意，PROG2 的新版本与伪会话示例中的版本相同。

迁移动态路由的应用程序

传递 COMMAREA 或通道的 EXEC CICS LINK 和 EXEC CICS START 命令可动态路由。

当 LINK 或 START 命令传递 COMMAREA，而不是通道时，根据请求的类型，路由程序可检查或更改 COMMAREA 的内容。对于终端相关 START 请求（由动态路由程序处理）启动，而不是终端无关的 START 请求（由分布式路由程序处理）启动的 LINK 请求和事务，在通信区的 DYRACMAA 字段中，路由程序被赋予应用程序 COMMAREA 的地址，而且可检查和更改其内容。

注：路由程序的通信区由 DFHDYPDS DSECT 映射。

如果您迁移动态路由的 EXEC CICS LINK 或 START 命令以使用通道，而不是 COMMAREA，则路由程序在 DFHDYPDS 的 DYRCHANL 字段中传递通道的名称。注意，因为路由程序被赋予通道的名称，而不是它的地址，所以路由程序无法使用 DYRCHANL 字段查看或更改通道容器的内容。

要赋予路由程序与通道相同类型的功能，使用通道的应用程序可在通道内创建一个特殊容器，名为 DFHROUTE。如果应用程序发出 LINK 或要动态路由的终端相关 START 请求（不是与终端无关的 START 请求），则在 DFHDYPDS 的 DYRACMAA 字段中，动态路由程序被赋予 DFHROUTE 容器的地址，而且可检查和更改其内容。

如果您迁移程序以传递通道，而不是 COMMAREA，您应该使用其现有的 COMMAREA 结构映射 DFHROUTE。

要获得有关动态和分布式路由的介绍信息，请参阅 *CICS Intercommunication Guide*。要获得有关编写动态或分布式路由程序的信息，请参阅 *CICS Customization Guide*。

共存

与其他 CICS 产品的共存

CICS TS 3.1 程序可以调用远程 CICS 区域上的程序并传递给它一个通道。为了成功完成此任务，远程区域还必须处于 CICS TS 3.1 级别。

尽管预 CICS TS 3.1 区域不支持通道，但是您可通过应用 APAR 获取它们，以允许通道。“允许”，我们意味着，如果后备级别 CICS 区域传递通道，则它将返回有意义的异常终止代码。

如果 CICS TS 3.1 应用程序尝试将通道发送到已应用相应 APAR 的后备级别区域，3.1 事务异常终止，返回有意义的异常终止代码。如果 CICS TS 3.1 应用程序尝试将通道发送到未应用相应 APAR 的后备级别区域，则结果不可预测。

以下列表显示允许通道的后备级别 CICS 产品，带有必须在每个情况下应用的 APAR:

CICS Transaction Server for z/OS V2R3

APAR PQ92437

CICS Transaction Server for z/OS V2R2

APAR PQ92437

CICS Transaction Server for OS/390 V1R3

APAR PQ93048

CICS Transaction Server for VSE/ESA™ R1.1

APAR PQ83049

CICSplex SM 支持

已对 CICSplex SM 路由程序 EYU9XLOP 进行修改来处理可能在它的通信区的 DYRLEVEL、DYRTYPE 和 DYRVER 字段中传递的新值，请参阅第 201 页的『动态和分布式路由程序』。当调用 EYU9XLOP 以路由以下任何一种请求类型时，如果有与请求关联的通道，则 EYU9XLOP 将请求路由到 CICS TS 3.1 区域（如果可用的话）：

- 通过与终端相关的 START 命令启动的事务
- 与终端无关的 START 请求
- 程序链接请求

对 CICSplex SM 应用程序编程接口的更改

已对下列资源表进行更改:

- 『TASK 资源表』
- 第 212 页的 『CONNECT 资源表』

TASK 资源表

已扩展 TASK 资源表，以包括下列新的监控器和统计信息属性:

TMRPGCTC

通道容器命令总数

TMRPGBCC

Browse 通道容器命令数

TMRPGGCC

GET 通道容器命令数

TMRPGPCC

PUT 通道容器命令数

TMRPGMCC

MOVE 通道容器命令数

TMRPGGCL

所有 GET CONTAINER CHANNEL 命令的容器中数据总字节长度

TMRPGPCL

所有 PUT CONTAINER CHANNEL 命令的容器中数据总字节长度

TMRPGCCC

为通道容器创建的容器数

TMRPCDLL

带 CHANNEL 选项的所有 DPL 请求的容器中数据总字节长度

TMRPCDRL

所有 DPL RETURN CHANNEL 命令的容器中数据总字节长度

TMRPCLCC

带 CHANNEL 选项的程序 LINK 请求数

TMRPCXCC

带 CHANNEL 选项的程序 XCTL 请求数

TMRPCDCC

带 CHANNEL 选项的分布式程序链接 (DPL) 请求数

TMRPCRCC

带 CHANNEL 选项的伪会话 RETURN 请求数

TMRPCRCL

所有伪会话 RETURN CHANNEL 命令的容器中数据总字节长度

TMRICSCC

用 CHANNEL 选项发出的本地时间间隔控制 START 请求总数

TMRICSCD

带有 CHANNEL 选项的所有本地时间间隔控制 START 请求的容器中数据的总长度, 以字节为单位

TMRICSRC

用 CHANNEL 选项发出的远程时间间隔控制 START 请求总数

TMRICSRD

带有 CHANNEL 选项的所有远程时间间隔控制 START 请求的容器中数据的总长度, 以字节为单位

CONNECT 资源表

已扩展 CONNECT 资源表, 以包括下列新的监控器和统计信息属性:

ESTPCCHNL

用于功能发送的、带通道的程序控制 LINK 请求数。

ESTPCCHNSENT

LINK 通道请求发送的字节数。它是连接发送的数据总数，包括任何控制信息。

ESTPCCHNRCVD

LINK 通道请求接收的字节数。它是连接接收的数据总数，包括任何控制信息。

ESTICCHNL

用于功能发送的、带通道的时间间隔控制 START 请求数。

ESTICCHNSENT

START 通道请求发送的字节数。它是连接发送的数据总数，包括任何控制信息。

ESTICCHNRCVD

START 通道请求接收的字节数。它是连接接收的数据总数，包括任何控制信息。

ESTTCCHNL

终端共享通道请求数。

ESTTCCHNSENT

终端共享通道请求发送的字节数。它是连接发送的数据总数，包括任何控制信息。

ESTTCCHNRCVD

终端共享通道请求接收的字节数。它是连接发送的数据总数，包括任何控制信息。

对 CICSplex SM Web 用户界面的更改

新的 WUI 视图

已引入下列 WUI 视图：

- 『通道用法视图』
- 『功能提供视图』

通道用法视图

在活动任务视图集中存在新的详细视图，称为**通道用法**，它与 TASK 资源表相关联。

要打开**通道用法**视图，执行下列操作：

1. 从主菜单单击**活动任务**
2. 选择**任务标识**以打开**活动任务**详细视图
3. 下滚并单击**通道用法**

显示新活动任务视图。此活动任务的视图名称为 EYUSTARTTASK.DETAIL10。请参阅第 211 页的『TASK 资源表』，以获得**通道使用情况**视图中所显示字段的属性详细信息。

功能提供视图

在 **ISC/MRO 连接**视图集中存在新的详细视图，称为**功能提供**，它与 CONNECT 资源表相关联。

要打开**功能提供**视图，执行下列操作：

1. 从主菜单单击 **CICS 操作视图**
2. 单击**连接操作视图**
3. 选择 **ISC 和 MRO 连接**
4. 单击“连接标识”以打开 **ISC/MRO 连接详细视图**
5. 下滚，然后单击**功能提供**，以打开新 **ISC/MRO 连接视图**

此 **ISC/MRO 连接**的视图名称为 **EYUSTARTCONNECT.DETAIL3**。

已更改的 **WUI 视图**

已对下列视图进行更改：

ISC/MRO 连接视图

很多新字段已添加到称为 **ISC/MRO 连接**的现有详细视图，与 **CONNECT 资源表**相关联。

要查看新字段，执行下列操作：

1. 从主菜单单击 **CICS 操作视图**
2. 单击**连接操作视图**
3. 选择 **ISC 和 MRO 连接**
4. 单击“连接标识”以打开 **ISC/MRO 连接详细视图**

显示其他新字段的详细视图的视图名称为 **EYUSTARTCONNECT.DETAILED**。请参阅第 212 页的『**CONNECT 资源表**』，以获得 **ISC/MRO 连接**详细视图中所显示新字段的属性详细信息。

第 8 章 OPENAPI 支持

CICS 通过提供对 OPENAPI 应用程序的支持来扩展开放式事务环境 (OTE) 功能的使用。在执行此操作前, OPENAPI 功能只能用于用户出口 (TRUE) 相关的任务。

OPENAPI 支持允许应用程序不仅仅将它自己定义为线程安全 (意味着它能够在 CICS 认为合适的任何 TCB, QR TCB 或开放 TCB 上运行), 而且远远不止这些, 也就是应用程序必须在 OPEN TCB 上运行, 而不是在 QR TCB 上运行。

OPENAPI 程序的使用允许应用程序工作负载从 QR TCB 移出到多个开放式 TCB 上。如果您选择使用 OPENAPI 程序作为使用其他 (非 CICS) 运行工作负载, 则 API 记住在 **CICS** 内使用其他 (非 **CICS**) API 完全凭用户处理, 冒用户的风险。**CICS** 内其他 (非**CICS**) API 的测试都未经过保证, 而且 **IBM Service** 不支持这种 API 的使用。

在两种情况的任何一种中, 您必须还意识到 OPENAPI 程序仍然对整个 CICS 系统负责。请参阅 *CICS Application Programming Guide*。

PROGRAM 资源定义上的新关键字 (API), 其采用两个值 CICSAPI 或 OPENAPI, 其中 CICSAPI 是缺省值。API(OPENAPI) 的设置要求 CONCURRENCY(THREADSAFE) 的设置意味着应用程序必须编码为线程安全标准, 所以其应用程序逻辑能够在多个 TCB 上并行执行时的执行保持完整性。CICS 将处理从这样的程序发出的任何 CICS API 的线程安全方面。新程序选项应用于用户应用程序、PLT 程序、用户可替代模块和任务相关的用户出口。对于全局用户出口忽略它。

CICSAPI QUASIRENT 程序、CICSAPI THREADSAFE 程序和 OPENAPI THREADSAFE 程序根据它的运行进行解释:

- CICSAPI QUASIRENT 程序仅发出 CICS API, 而其应用程序逻辑不是线程安全的。它总是在 QR TCB 上运行。
- CICSAPI THREADSAFE 程序能够在 QR TCB 或开放 TCB 上运行, 这是因为其应用程序逻辑是线程安全的。这种程序在 QR TCB 上运行, 直到一些事件将它移动到开放 TCB。对 OPENAPI TRUE 的调用, 如 DB2 调用, 是将 CICSAPI THREADSAFE 程序移动到开放 TCB 的事件示例。在传送到开放 TCB 后, 程序保持不变, 直到一些内容将它强制返回到 QR TCB, 例如非线程安全 CICS API 调用。如果发生此情况, 则此程序保持在 QR TCB 上, 直到一些内容 (可能是另一次 DB2 调用) 将它再一次强制返回到开放 TCB。

CICSAPI 程序仅使用以独立于使用中 TCB 键的方法实现的 CICS API。应用程序可使用用户键或 CICS 键成功运行, 不管 TCB 的键怎样。所以它们可在 QR TCB (L8 或 L9 TCB) 上运行。

- OPENAPI THREADSAFE 程序总是在开放 TCB 上运行, 而且在程序启动时运行。它能够在开放 TCB 上运行, 这是因为其应用程序逻辑是线程安全的。如果使用非线程安全 CICS 命令强制切换回 QR TCB, 则 CICS 在返回对应用程序的控制前再次切换回开放 TCB。

OPENAPI 程序潜在使用其他 (非 CICS) API, 而且这种 API 通常需要 TCB 的键匹配执行键。因此, 用户键程序在 L9 TCB 上运行, 而 CICS 键程序在 L8 TCB 上运行。

由于 TCB 键的需求对于 OPENAPI 程序要正确，使用 OPENAPI 程序可导致切换的 TCB 比线程安全 CICSAPI 程序更多，这是因为非线程安全 CICS 调用导致两次 TCB 切换，而且因为 OPENAPI TRUE 总是在 L8 TCB 上 CICS 键中运行。因此，例如，用户键 OPENAPI 程序在 L9 TCB 上运行，但是如果它发出 DB2 调用，则 CICS 切换到 L8 TCB，以调用 DB2，然后为应用程序返回到 L9。

强烈建议现有用户键线程安全 CICS-DB2 应用程序（其利用能够在 DB2 调用的同一 TCB 上运行的性能收益）仍然定义为 CICSAPI THREADSAFE 应用程序。如果其他功能希望哪个需要 OPENAPI，应该使用不同的程序。

定义为 OPENAPI THREADSAFE 的候选程序（假设其应用程序逻辑是线程安全的）包括：

- 仅使用 CICS 线程安全 API（避免两次 TCB 切换）或仅使用有限制的非线程安全 CICS 命令的程序
- CICS 键 CICS-DB2 应用程序
- CPU 增强型程序
- 希望在其自己的风险上使用其他（非 CICS）API 的程序

OPENAPI 支持的优点

提供对 OPENAPI 程序支持的主要原因是允许您将应用程序工作负载移出 QR TCB，移入多个开放 TCB。这允许更好利用机器资源的可能性，以实现更好吞吐量。

您可能要使用 OPENAPI 参数的另一个原因是允许使用其他（非 CICS）API。

使用其他 API 是可能的，这是因为，如果操作系统等待阻塞开放 TCB，则只有单个应用程序受到影响，而不是整个 CICS，这可能是它们在 QR TCB 下执行的情况。这种 OPENAPI 程序不允许正好在 QR TCB 上执行，这是由于操作系统等待阻塞 TCB，从而影响整个 CICS 的风险。

- 在 **CICS** 内使用其他（非 **CICS**）API 完全凭用户处理，冒用户的风险。**CICS** 内其他（非 **CICS**）API 的测试都未经过保证，而且 **IBM Service** 不支持这种 API 的使用。

需求

除了该 CICS 发行版的一般需求，OPENAPI 支持对硬件、软件或资源用途没有特殊的需求。

对 CICS 外部的更改

对系统初始化参数的更改

OPENAPI 支持更改 FORCEQR 系统初始化参数的描述，以限制其与 CICSAPI 程序的相关信息，这是因为它不适用于 OPENAPI 程序。要获得 FORCEQR 修正描述的完整文本，请参阅 *CICS System Definition Guide* 中的“FORCEQR”。

OPENAPI 支持更改 MAXOPENTCBS 系统初始化参数的描述，以包含 OPENAPI 和 L9 TCB 的需求。要获得 MAXOPENTCBS 修正描述的完整文本，请参阅 *CICS System Definition Guide*。

对资源定义的改变

OPENAPI 支持为 PROGRAM 资源定义引入新属性 API。

API 具有两个可能的值 CICSAPI 和 OPENAPI。要获得 API 属性的完整描述，以及 PROGRAM 定义的修订语法，请参阅 *CICS Resource Definition Guide*。

对应用程序编程接口的更改

此主题处理：

- OPENAPI 程序的职责，以及
- 使用 EDF 的限制

OPENAPI 程序的职责

OPENAPI 程序尽管从 QR TCB 利用的约束释放，然而，它对整个 CICS 系统和使用它的 L8 或 L9 TCB 未来的用户都有职责。L8 或 L9 TCB 专供分配使用它的 CICS 任务使用，但是一旦 CICS 任务完成后，TCB 返回给这种 TCB 的分派器管理池，假设它仍然处于“clean”状态。（此上下文中不清楚 TCB 意味着使用 L8 或 L9 方式 TCB 的任务在 OPENAPI 中遭遇无法处理的异常终止，但是不表示程序中已中断线程安全限制，CICS 无法检测这些限制。）注意，TCB 不是专供特殊 OPENAPI 程序使用的，但是它由分配 L8 方式 TCB 的 CICS 任务调用的所有 OPENAPI 程序和 OPENAPI TRUE 使用。而且，如果调用 OPENAPI 程序的应用程序编码为线程安全，并作为线程安全定义到 CICS，则在从程序返回时，它继续在 L8 方式 TCB 上执行。

线程安全限制：

OPENAPI 程序不能以导致下列问题的方法处理开放 TCB 环境的执行：

- 可在开放 TCB 上运行的应用程序逻辑
- 同一任务调用的 OPENAPI TRUE
- 可能使用开放 TCB 的未来任务
- CICS 管理代码。

出于您自己的风险，如果 OPENAPI 程序计划使用其他（非 CICS）API，您必须意识到以下各项：

- 当调用 CICS 服务，或者当返回到 CICS 时，OPENAPI 程序必须确保它在程序入口处，恢复 MVS 编程环境。这包括跨内存方式、ASC 方式、请求块（RB）级别、链接堆栈级别、TCB 分派优先级，除了取消任何已添加的 ESTAE 之外。
- CICS 任务终端 OPENAPI 程序必须确保将开放 TCB 保留在适合另一个 CICS 事务可重用的状态。特别是，它必须确保代表终止任务特别获取的所有非 CICS 资源已释放。这样的资源可能包括：
 - 动态分配的数据集
 - 开放 ACB 或 DCB
 - STIMERM 请求

- MVS 管理的存储器
- ENQ 请求
- 已连接的子任务
- 已装入的模块
- 所属的数据空间
- 已添加的访问列表条目
- 名称 / 令牌对
- 已修订的页
- 安全性设置 (TCBSENV 必须设置为 0)
- OPENAPI 程序不能使用将影响整个 CICS 操作的下列 MVS 系统服务:
 - CHKPT
 - ESPIE
 - QEDIT
 - SPIE
 - STIMER
 - TTIMER
 - XCTL / XCTLX
 - 任何 TSO/E 服务。
- OPENAPI 程序不能在 L8 或 L9 方式 TCB (使用 MVS Language Environment 服务的 Language Environment 程序) 下调用, 这是因为 L8 和 L9 方式 TCB 是使用 CICS 服务为 Language Environment 初始化的。

使用 EDF 的限制

OPEN TCB 和 EDF

即使您的程序通常使用 OPEN TCB (L8、L9、X8 或 X9) 运行, 但是 CEDF 强制程序在 QR TCB 上运行, 这是因为 CEDF 本身不是线程安全的。

对系统编程接口的更改

OPENAPI 支持导致对 SPI 按如下所示进行更改:

INQUIRE DISPATCHER

INQUIRE DISPATCHER 命令中 ACTOPENTCBS 和 MAXOPENTCBS 的描述更改为包含 L9 方式 TCB, 并引用 OPENAPI 程序。要获得修正描述的完整文本, 请参阅 *CICS System Programming Reference*。

INQUIRE EXITPROGRAM

引入 INQUIRE EXITPROGRAM 命令中 APIST 选项的新值 CICSAPI。CICSAPI 与前一个值 BASEAPI 同义。要获得修正描述的完整文本, 请参阅 *CICS System Programming Reference*。

INQUIRE PROGRAM

新选项 APIST 添加到 INQUIRE PROGRAM 命令中。APIST 允许您指定 CICSAPI 或 OPENAPI。要获得 APIST 描述的完整文本, 请参阅 *CICS System Programming Reference*。

INQUIRE SYSTEM

INQUIRE SYSTEM 命令中 FORCEQR 的描述更改为限制 CICSAPI 程序的相关值，这是因为它不适用于 OPENAPI 程序。要获得修正描述的完整文本，请参阅 *CICS System Programming Reference*。

SET DISPATCHER

SET DISPATCHER 命令中 MAXOPENTCBS 的描述更改为包含 L9 方式 TCB。要获得修正描述的完整文本，请参阅 *CICS System Programming Reference*。

SET SYSTEM

SET SYSTEM 命令中 FORCEQR 选项的 cvda 值 FORCE 的描述更改为限制 CICSAPI 程序的相关值，这是因为它不适用于 OPENAPI 程序。要获得修正描述的完整文本，请参阅 *CICS System Programming Reference*。

对 CEMT 的更改

OPENAPI 支持导致下列更改：

CEMT INQUIRE DISPATCHER

CEMT INQUIRE DISPATCHER 命令中 ACTOPENTCBS 和 MAXOPENTCBS 的描述更改为包含 L9 方式 TCB，并引用 OPENAPI 程序。要获得修正描述的完整文本，请参阅 *CICS Supplied Transactions*。

CEMT INQUIRE PROGRAM

新选项 APIST 添加到 CEMT INQUIRE PROGRAM 命令。APIST 允许您指定 CICSAPI 或 OPENAPI。要获得 APIST 描述的完整文本，请参阅 *CICS Supplied Transactions*。

CEMT INQUIRE SYSTEM

CEMT INQUIRE SYSTEM 命令中 FORCEQR 的描述更改为限制 CICSAPI 程序的相关值，这是因为它不适用于 OPENAPI 程序。要获得修正描述的完整文本，请参阅 *CICS Supplied Transactions*。

CEMT SET DISPATCHER

CEMT SET DISPATCHER 命令中 MAXOPENTCBS 的描述更改为包含 L9 方式 TCB。要获得修正描述的完整文本，请参阅 *CICS Supplied Transactions*。

CEMT SET SYSTEM

CEMT SET SYSTEM 命令中 FORCEQR 选项的 cvda 值 FORCE 的描述更改为限制 CICSAPI 程序的相关值，这是因为它不适用于 OPENAPI 程序。要获得修正描述的完整文本，请参阅 *CICS Supplied Transactions*。

第 9 章 XPLink 支持

额外的性能链接（从现在起缩写为 XPLink）是提供高性能子例程调用和返回机制的 isa z/OS 功能。这使执行路径长度变短且高度优化。

面向对象的编程是基于将“消息”发送到对象的概念来构建的，这些对象导致该对象执行某些操作。发送活动的消息是作为子例程调用来实现的。在 C++ 术语中称为成员函数的子例程通常是代码的小段。典型 C++ 程序的特性执行流属于对代码小段的许多子例程调用。此种类型的程序受益于 XPLink 优化技术。

MVS™ 有调用约定的标准子例程，该约定可追溯到 System/360™ 的早先时期。此转换优化了子例程更复杂的环境，子例程相对少一些，而对它们的调用相对不频繁。面向对象的编程约定已更改了它。子例程已变得更简单，但属性很多，而且子例程调用的频率已按大小顺序增加。对子例程的大小、数量和使用模式方面的更改使它更值得优化所涉及的系统开销。XPLink 是此优化的结果。

注:

对于 z/OS1.4 和更高版本，以及 CICS 3.1 和更高版本，此处建议您能够使用 XPLINK 编译器选项和 CICS 应用程序，覆盖 z/OS 相反的建议。

z/OS 对 C 和 C++ 的手册建议您不要将 XPLINK 编译器选项用于 CICS 应用程序，因为它用于此情况。尽管这些手册现在已更改，您可能使用此更改之前产生的这些手册中一本手册的副本。

XPLink, 以及 X8 和 X9 TCB

CICS 通过使用 CICS 开放式事务环境 (OTE) 技术中多个 TCB 功能，提供对使用 XPLINK 选项编译的 C 和 C++ 程序的支持。X8 和 X9 方式 TCB 定义为分别在 CICS 密钥和 USER 密钥中支持 XPLink 任务。XPLink 程序的每个实例使用一个 X8 或 X9 TCB。

要使用 XPLink，您的 C 或 C++ 应用程序代码必须可重入和线程安全。同一代码实例可在多个 MVS TCB 上执行，而如果没有保护共享资源的线程安全机制，则应用程序代码的执行行为是不可预测的。这无法太过着重强调。

为 CICS 环境编写 C 和 C++ 程序，其使用 XPLINK 选项编译

希望应用程序开发者执行以下步骤来利用 CICS XPLink 支持;

- 严格地遵循线程安全编程原则和技术来开发代码
- 通过将 XPLINK 选项设置为 On 来编译 C 或 C++ 程序
- 在程序线程安全的 PROGRAM 资源定义中表明
- 考虑在 CEEUOPT 或 #pragma（请参阅 *CICS Application Programming Guide*，以获得详细信息）。

使用 CICS XPLink 支持的所有程序必须可重入和线程安全。只有应用程序开发者可以保证特殊应用程序的代码满足这些需求。

在 XPLink 和非 XPLink 对象之间传递控制

从 XPLink 对象到非 XPLink 对象到每次控制传送，或反向传送都导致 QR TCB 和开放 TCB（X8 或 X9 TCB）之间的切换。在性能方面，TCB 切换开销很大，您必须考虑性能开销。

XPLink 对象使用 EXEC CICS 接口或 Language Environment[®] 接口调用非 XPLink 对象。

非 XPLink 对象只能使用 EXEC CICS 接口调用 XPLink 对象。不支持为这种调用使用 Language Environment 接口。

更改 CICS 定义为使用 XPLINK 选项编译的对象获取 CICS 支持

对使用 XPLINK 选项所编译程序的 CICS 支持仅需要您在程序为线程安全的 PROGRAM 资源定义中显示。此指示和装入模块中的 XPLink“签名”是将任务放入 X8 或 X9 TCB 唯一需要的项目。

在为特殊程序选择合适的 TCB 时，XPLink 优先考虑 PROGRAM 资源定义上 API 属性的现有 OPENAPI 值。

XPLink 支持的优点

XPLink 支持 提供性能和功能性两方面的优点：

- 来自优化的子例程链接技术的性能优势。
 - 产生功能优点是由于
 - 您可以开始开发可以由在 CICS、TSO/Batch、IMS[™] 或 Unix 系统服务下运行的程序使用或调用的公共模块或 DLL。
 - C++ 开发者还可以更充分地利用 C++ 标准模板库。
- 将它们组合起来可能会产生更大的 C/C++ 代码重用性。

需求

除了该 CICS 发行版的一般需求，XPLink 支持对硬件、软件或资源用途没有特殊的需求。

编程样式

使用 CICS XPLink 支持的所有程序必须是可重入的，并且是线程安全的。

对 CICS 外部的更改

对安装的更改

由于对 C 和 C++ 程序的 XPLINK 编译器选项的支持，对 Language Environment 需要的库必须定义到 CICS 的方法进行一些更改。这些是库 SCEERUN 和 SCEERUN2。

- 库 SCEERUN2 必须在 STEPLIB 和 DFHRPL 并置中定义。
- 在两个并置中，SCEERUN2 都必须出现在 SCEERUN 之前。

要获得更多信息，请参阅《安装指南》中的“为 Language Environment 安装 CICS 支持”。要获取更多信息，请参阅 *CICS System Programming Reference*。

对系统初始化参数的更改

XPLink 支持有新的系统初始化参数。新参数为 MAXXPTCBS。

MAXXPTCBS={5vnumber}

指定可在 CICS 区域中并存的最大开放式 X8 和 X9 TCB 数。X8 和 X9 是用于运行使用 XPLINK 编译器选项进行编译的 C 和 C++ 程序的 TCB。X8 TCB 用于 CICS 键中的程序，而 X9 方式 TCB 用于用户键中的程序。

允许 CICS 任务使用它需要的 X8 和 X9 TCB 数，而且这些 TCB 仅在程序终止前才存在。

对资源定义的更改

没有对 XPLink 支持的资源定义作任何更改。但是，您必须确保使用 XPLink 标志集编译的程序已在相应的程序定义中将 CONCURRENCY 属性设置为 THREADSAFE。

对应用程序编程接口的更改

EXEC CICS HANDLE

C 和 C++ 代码被限制为仅使用来自 EXEC CICS HANDLE“系列”中的所有 API 的 EXEC CICS HANDLE ABEND PROGRAM。此限制继续使用 XPLink 支持，并且由 C 和 C++ 转换程序控制。

其他 EXEC CICS 限制

EXEC CICS RETURN、EXEC CICS XCTL 和 EXEC CICS SEND PAGE RELEASE 能够实现从用户代码到 CICS 的直接控制传送。在 C++ 中，使用这些 API 应该十分小心，使之仍然能够正确地驱动 C++ 对象的析构函数。无法驱动对象析构函数可导致存储泄漏、打开文件一直处于打开状态、锁保持锁定状态以及其他问题。它是 C++ 中的在对象构造期间获取资源，以及在对象销毁期间释放资源的标准操作。已分配堆栈的对象不起作用时，自动调用它们的析构函数。有些 C++ 对象还可以静态地构造，即，在进入“main”函数之前构造它们，并在“main”函数结束或调用“exit”调用时销毁它们。

多线程

因为 C 和 C++ CICS 应用程序都是用 XPLink 标志编译的，并且满足此处描述的其他条件且在它们自己的 TCB 上运行，所以应用程序开发者可假设它们还可以使用 C、C++ 和 POSIX API 的全部集合。但情况并非如此。

用 C 或 C++ 编写的 CICS 应用程序不应该使用他们的应用程序代码中的多线程技术。建议您不要使用这些技术。

通过多线程技术，我们指的是在应用程序中创建多个执行路径的那些编码形式，例如，使用 fork() 语句或 pthread。没有在 CICS 环境中测试它们，并且 CICS 将它们视为不受支持。IBM 将不接受任何可能与使用这些技术相关而产生的问题报告。

OPEN TCB 和 EDF

即使您的程序通常使用 QR TCB 的 OPEN TCB (L8、L9、X8 或 X9) 强制使用, 这是因为 CEDF 本身不是线程安全的。

对系统编程接口的更改

INQUIRE DISPATCHER

更改 EXEC CICS INQUIRE DISPATCHER 命令, 以包含 ACTXPTCBS 和 MAXXPTCBS 选项。

ACTXPTCBS(value)

显示了当前活动 (即, 分配给用户任务) 的 X8 和 X9 方式 TCB 开放数目。

MAXXPTCBS(value)

显示可在 CICS 区域中并存的最大 X8 和 X9 方式开放 TCB 数。此值的范围可以是 1-999。您可通过重新输入不同的值来重置此值。

INQUIRE PROGRAM

更改 EXEC CICS INQUIRE PROGRAM 命令, 以为“运行时”选项包含 XPLink 值。

运行时

显示有关程序的运行时环境的信息。此列表中的新值为:

XPLink

此程序是已使用 XPLINK 选项编译的 C 或 C++ 程序。

SET DISPATCHER

更改 EXEC CICS SET DISPATCHER 命令, 以包含 MAXXPTCBS 选项。

MAXXPTCBS(value)

指定可在 CICS 区域中并存的最大 X8 和 X9 方式开放 TCB 数。所指定值的范围是 1 到 999。如果您将 MAXXPTCBS 从其先前定义的值减少, 而且新值小于当前所分配的开放 TCB 数, 则 CICS 拆离 TCB 以仅当用户任务释放它们时实现新的限制。事务未异常终止, 允许拆离 TCB 以实现新的限制。如果存在任务排队等待 X8 或 X9 方式 TCB, 而且您将 MAXXPTCBS 从其先前定义的值增加, 则 CICS 连接新的 TCB, 以继续每个排队中的任务, 直到新的限制。

XPLINK cvda

存在新的 CVDA 值 1069, 表示 XPLink。

对 CEMT 的更改

CEMT INQUIRE DISPATCHER

更改 CEMT INQUIRE DISPATCHER 命令, 以包含 ACTXPTCBS 和 MAXXPTCBS 选项。

ACTXPTCBS(value)

显示了当前活动（即，分配给用户任务）的 X8 和 X9 方式 TCB 开放数目。

MAXXPTCBS(value)

显示可在 CICS 区域中并存的最大 X8 和 X9 方式开放 TCB 数。此值的范围可以是 1-999。您可通过重新输入不同的值来重置此值。

CEMT INQUIRE PROGRAM

更改 CEMT INQUIRE PROGRAM 命令，以为“运行时”选项包含 XPLink 值。

运行时

显示有关程序的运行时环境的信息。此列表中的新值为：

XPLink

此程序是已使用 XPLINK 选项编译的 C 或 C++ 程序。

CEMT SET DISPATCHER

更改 CEMT SET DISPATCHER 命令，以包含 MAXXPTCBS 选项。

MAXXPTCBS(value)

指定可在 CICS 区域中并存的最大 X8 和 X9 方式开放 TCB 数。所指定值的范围是 1 到 999。如果您将 MAXXPTCBS 从其先前定义的值减少，而且新值小于当前所分配的开放 TCB 数，则 CICS 拆离 TCB 以仅当用户任务释放它们时实现新的限制。事务未异常终止，允许拆离 TCB 以实现新的限制。如果存在任务排队等待 X8 或 X9 方式 TCB，而且您将 MAXXPTCBS 从其先前定义的值增加，则 CICS 连接新的 TCB，以继续每个排队中的任务，直到新的限制。

对全局用户出口的更改

XPCFTCH

当从使用 XPLINK 选项编译的 C 或 C++ 程序调用出口 XPCFTCH 时，设置标志表明如果出口指定任何已修改的入口点地址，将忽略它。

XPCTA

当从使用 XPLINK 选项编译的 C 或 C++ 程序调用出口 XPCTA 时，设置标志表明如果出口指定恢复地址，将忽略它。

DFHUEPAR

两个符号值 UEPTX8 和 UEPTX9 添加到 DFHUEPAR 中的 TCB 指示符表。

对用户可替代程序的更改

提供新的用户可替代程序 **DFHAPXPO**。

在每个 Language Environment 环境（其中要运行使用 XPLINK 选项编译的 C 或 C++ 程序）的 PIPI 预初始化阶段，装入 DFHAPXPO。它允许您改变缺省 Language

Environment 运行时选项。请参阅 *z/OS Version 1.4 Language Environment Programming Guide*, SC22-7561, 以获得可重置的 Language Environment 选项详细信息。必须使用汇编语言编写此程序。

对监控的更改

在“CICS Performance Guide”中的主题“A note about wait (suspend) times”中，新术语添加到此表中，按如下所示：

表 10. 性能类等待（暂挂）字段

字段标识	组名	描述
282	DFHTASK	CICS MAXXPTCBS 延迟时间

将新字段添加到性能类监控记录。这些是对组 DFHTASK 的增补，而且存在对此组中现有项描述的一些更改：

新项

271 (TYPE-S, “X8CPUT”, 8 BYTES)

CICS 分派器域在 CICS X8 方式 TCB 上分派用户任务的处理器时间。当事务调用使用 XPLINK 选项编译，并且使用 EXECKEY=CICS 定义的 C 或 C++ 程序时，它被分配和使用 CICS X8 方式 TCB。（如果使用 EXECKEY=USER 定义程序，还是分配 X8 方式 TCB，但是存储保护工具是不活动的。）一旦任务分配为 X8 方式 TCB 后，相同的 TCB 仍然与此任务相关联，直到此程序完成。

272 (TYPE-S, “X9CPUT”, 8 BYTES)

CICS 分派器域在 CICS X9 方式 TCB 上分派用户任务的处理器时间。当事务调用使用 XPLINK 选项编译，并且使用 EXECKEY=USER 定义的 C 或 C++ 程序时，它被分配和使用 USER X9 方式 TCB。（如果存储保护工具不活动，则使用 X8 方式 TCB，代替 X9 方式 TCB。）一旦任务分配为 X9 方式 TCB 后，相同的 TCB 仍然与此任务相关联，直到此程序完成。

282 (TYPE-S, “MAXXTDLY”, 8 BYTES)

由于 CICS 系统已达到系统参数 MAXXPTCBS 设置的极限，用户任务等待获得 CICS XP TCB（H8 或 X9 方式）用去的时间。X8 和 X9 方式开放 TCB 专门由使用 XPLINK 选项的 C 和 C++ 程序使用。

注：此字段是任务暂挂时间字段 **SUSPTIME**（组名称：DFHTASK，字段标识：014）的一个组件。

已更改的项

007 (TYPE-S, “USRDISPT”, 8 BYTES)

X8 和 X9 添加到 TCB 方式列表中。

008 (TYPE-S, “USRCPUT”, 8 BYTES)

X8 和 X9 添加到 TCB 方式列表中。

262 (TYPE-S, ‘KY8DISPT’, 8 BYTES)

要获得组成总耗用时间的项列表，

- 当事务调用使用 XPLINK 选项编译，并且使用 EXECKEY=CICS 定义的 C 或 C++ 程序时，它被分配 CICS X8 方式 TCB，并在此 TCB 上分派。TCB 仍然与此任务相关联，直到程序结束。

已添加。

263 (TYPE-S, 'KY8CPUT', 8 BYTES)

要获得组成处理器时间的项列表，

- 当事务调用使用 XPLINK 选项编译，并且使用 EXECKEY=CICS 定义的 C 或 C++ 程序时，它被分配 CICS X8 方式 TCB，并在此 TCB 上分派。TCB 仍然与此任务相关联，直到程序结束。

已添加。

264 (TYPE-S, "KY9DISPT", 8 BYTES)

要获得组成总耗用时间的项列表，

- 当事务调用使用 XPLINK 选项编译，并且使用 EXECKEY=USER 定义的 C 或 C++ 程序时，它被分配 USER X9 方式 TCB，并在此 TCB 上分派。TCB 仍然与此任务相关联，直到程序结束。

已添加。

265 (TYPE-S, "KY9CPUT", 8 BYTES)

要获得组成处理器时间的项列表，

- 当事务调用使用 XPLINK 选项编译，并且使用 EXECKEY=USER 定义的 C 或 C++ 程序时，它被分配 USER X9 方式 TCB，并在此 TCB 上分派。TCB 仍然与此任务相关联，直到程序结束。

已添加。

通过在监控控制表 (MCT) 中对 DFHMCT TYPE=RECORD 条目进行编码，可以将所有新字段和已更改的字段从监控记录中排除。

对统计信息的更改

对 DFHDSGDS DSECT 映射的“分派器域: TCB 方式统计信息”进行了一些更改。

这些更改将 X8 和 X9 方式 TCB，以及 TCB 的 XP 池添加到此表中的项描述。

表 11. “分派器域: TCB 方式统计信息”中已更改的字段

DFHSTUP 名	字段名	描述
TCB 方式	DSGTCBNM	是 CICS 分派器 TCB 方式的名称，QR、RO、CO、SZ、RP、FO、SL、SO、S8、D2、JM、L8、L9、J8、J9、X8 或 X9。 <u>重置特性:</u> 不重置
TCB 池	DSGTCBMP	定义此 CICS 分派器 TCB 方式的 TCB 池的名称，N/A、OPEN、JVM 或 XP。 <u>重置特性:</u> 不重置

迁移

现有功能的迁移

使用不带 XPLink 的 C 和 C++ 时不需要采取任何操作来继续。

迁移到新的功能

要利用现在可用于 C 和 C++ 程序的 XPLink 编译器选项的支持，考虑以下各点：

- 确保 C 和 C++ 程序是可重入的，并且线程安全，或者将它修改为符合这些标准，请参阅 *Migration Guide* 中的“Migration planning for threadsafe programming and the open transaction environment (OTE)”
- 如果程序使用 XPCFTCH 或 XPCTA 出口，注意 *CICS Application Programming Guide* 的“Global User exits and XPLink”中的建议：
 - CICS 忽视 XPCFTCH 修改入口点的任何尝试
 - CICS 忽视 XPCTA 定义继续地址的任何尝试

您必须找到管理这种需求的其他方法，或者断定此程序不适合作为 XPLINK 优化的候选值。

- 使用 XPLINK 编译器选项重新编译程序。
- 更新此程序的 PROGRAM 资源定义的并行属性，将此值设置为线程安全。

CICSplex SM 支持

对 CICSplex SM Web 用户界面和 API 进行了很多更改，以与对 XPLINK C++ 的新支持匹配。

对 CICSplex SM 应用程序编程接口的更改

对资源表的更改

已对下列资源表进行更改：

- 『PROGRAM 资源表』
- 『PROGDEF 资源表』
- 第 229 页的 『TASK 资源表』
- 第 229 页的 『CICSRGN 资源表』

PROGRAM 资源表

PROGRAM 资源表包含下列属性：

RUNTIME

显示 CVDA (XPLINK) 值 1068

APIST 显示表明 CICSAPI 或 OPENAPI 的 API 状态的 CVDA 值

PROGDEF 资源表

PROGDEF 资源表包含下列属性：

API 显示 CICSAPI 或 OPENAPI 的 EYUDA 值

TASK 资源表

TASK 资源表包含下列属性:

#

TMRL9CPU

显示用户任务 L9 方式 CPU 时间

CICSRGN 资源表

CICSRGN 资源表包括下列 SPI 属性:

MAXXPTCBS

显示 XPLink OTE X8/X9 方式池中 TCB 的当前最大数目。

ACTXPTCBS

显示 XPLink OTE X8/X9 方式池中 TCB 的实际数目。

MAXSSLTCBS

显示 SSL OTE 池中 TCB 的当前最大数目。

ACTSSLTCBS

显示当前分配的 SSL 池 TCB 数。

现有 DISPATCHER SPI 属性 MAXHPTCBS 和 ACTHPTCBS 不再受支持，而且标志为 NOTVALID。

对 CICSplex SM Web 用户界面的更改

已对下列 WUI 视图进行更改:

- 『程序视图』
- 『“程序定义”视图』
- 『CPU 和 TCB 信息视图』

程序视图

下列视图已添加到程序 (EYUSTARTPROG.DETAILED) 视图:

APIST API 状态

“程序定义”视图

下列属性已添加到程序定义 (EYUSTARTPROGDEF.DETAILED) 视图:

API 应用程序编程接口

CPU 和 TCB 信息视图

下列属性已添加到 CPU 和 TCB 信息视图，任务操作视图中的某个活动任务视图集 (EYUSTARTTASK.DETAIL9) :

TMRL9CPU

用户任务 L9 方式 CPU 时间

第 10 章 对符合汇编程序 MAIN 程序的 Language Environment 的支持

现在，您可产生组装器 MAIN 程序，其符合 Language Environment。

符合汇编程序 MAIN 程序的 Language Environment

直到现在，在 CICS 中使用符合 Language Environment 的组装器的唯一方法是使用来自 COBOL PLI 或 C 符合 Language Environment 的程序的调用，并使用高级语言（HLL）程序链接编辑组装器程序。这使组装器程序成为 Language Environment 子例程。它必须具有 CEEENTRY 上的 MAIN=NO。用户必须指定 NOPROLOG 和 NOEPILOG，然后分别对 CEEENTRY 和 CEETERM 调用进行编码。CICS PROGRAM 资源不能定义为 ASM 和 LE370。

CICS 现在支持对符合 Language Environment 的组装器 MAIN 程序的编码。新转换程序选项 LEASM 导致 Language Environment 功能用于设置程序的环境。这种程序可使用存根 DFHELII，而不是 DFHEAI 进行链接编辑。

此支持还启用 Debugger for Assembler 程序的使用。

对符合汇编程序 MAIN 程序的 Language Environment 的支持的优点

对符合汇编程序 MAIN 程序的 Language Environment 的支持扩展 Language Environment 使用的可用性，并使调试器支持可与这样的程序一起使用。

需求

除了该 CICS 发行版的一般需求，对符合汇编程序 MAIN 程序的 Language Environment 的支持对硬件、软件或资源用途没有特殊的需求。

对 CICS 外部的更改

对资源定义的更改

PROGRAM 资源定义的 Language 属性已更改一些可指定值的描述。

Language({COBOL|ASSEMBLER|LE370|PLI})

指定程序语言：

ASSEMBLER

这是汇编语言程序，其不能使用 LEASM 转换程序选项进行转换。LEASM 用于转换那些要成为符合 Language Environment 的 MAIN 程序的汇编程序。

LE370 此程序开发多语言支持，已由符合 Language Environment 的编译器进行编译，或者它是使用 LEASM 选项进行转换的汇编程序 MAIN 程序，以产生符合 Language Environment 的程序。

对应用程序编程接口的更改

语言限制

当以汇编程序进行编程，并计划使用 LEASM 选项转换您的程序时，这些限制添加到应用于所有汇编程序的那些选项。

- 寄存器 2 不能用作代码库寄存器。
- 寄存器 12 由 Language Environment 保留，用于指向 Language Environment 常见锚区域 (CAA)，所以在未相应保存和存储的情况下，程序根本不能使用它们。
- 寄存器 13 必须用作唯一的工作存储基址寄存器。
- 此程序不能时全局用户出口程序 (GLUE) 或任务相关的用户出口程序 (TRUE)。
- 此程序不能使用或依赖任何 AMODE(24) 代码。

转换程序选项

新转换程序选项 LEASM (仅对汇编程序有效) 允许您指定此程序通过使用将成为符合 Language Environment 的程序，准备用于汇编程序作为 MAIN 程序的宏进行转换。

LEASM 的规范导致设置新的汇编程序全局 &DFHEILE。第四个位置参数 LE 已添加到转换程序插入每个输出文件顶部的 DFHEIGBL 宏。如果指定 LEASM，DFHEIGBL 更改为将 &DFHEILE 设置为打开。

在已指定 LEASM 时，转换程序将值 (X'12') 设置为语言 ARG0。转换程序的输出将等同于在每个其他方法中不指定 LEASM 而产生的输出。

如果设置 &DFHEILE，DFHEISTG、DFHEIENT、DFHEIRET 和 DFHEIEND 宏不同地扩展，以创建 LE 环境，而不是常规的 CICS 环境。这允许您的程序使用 NOPROLOG 和 NOEPILOG，并对自己的 DFHEIENT 和其他宏进行编码，以在不更改 Language Environment 支持的程序源的情况下利用它。例如，需要多个代码库寄存器的所有程序属于此类别，这是因为转换程序不支持多代码库寄存器。

DFHEISTG

如果设置 &DFHEILE，语句 CEEDSA SECTYPE=OS 将添加在 DFHEISTG 的顶部，紧跟在 DSECT 语句之后。这导致包括标准 Language Environment DSA，而不包括 DSECT 语句。

DFHEIENT

DFHEIENT 生成 Language Environment CEEENTRY 宏，其设置代码对等寻址能力和工作存储，而不是使用标准 CICS 方法。

代码对等寻址能力

DFHEIENT 具有缺省为 3 的 CODEREG 参数。当指定转换程序选项 LEASM 时，DFHEIENT 使用 BASE 参数生成 CEEENTRY，CODEREG 值的不同副本。

CEEENTRY 不允许使用寄存器 2 或 12 进行代码对等寻址能力。在执行 Language Environment 程序期间，寄存器 12 必须总是包含 Language Environment CAA 的地址。寄存器 2 通常不用作代码库，这是因为它由 TRT 这样的指示信息修改。如果在使用 LEASM 选项转换的程序中，在 CODEREG 上指定寄存器 2 或寄存器 12，则 DFHEIENT 将生成错误消息。

工作存储对等寻址能力

DFHEIENT 具有缺省为 13 的 DATAREG 参数。CEEENTRY 不允许出 R13 之外的任何值用于工作存储, 而且不支持多个工作存储寄存器。DFHEIENT 不允许在使用 LEASM 选项转换的程序中为 DATAREG 使用除 13 之外的任何值。

DFHEIRET

使用 LEASM 选项为已转换的程序生成 CEETERM RC=0。

EXAMPLE LEASM PROGRAM

此处是简单的 CICS 汇编程序。

```
*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
```

当转换和汇编时, 它变成:

```
*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 0HHMMSS FORMAT @BBAC81A 01-DFHEI
EIBDATE DS PL4 DATE IN 0CYYDDD+ 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 @BBDIA0U 01-DFHEI
EIBSYNC DS C X'FF' SYNCPOINT REQUESTED @BBDIA0U 01-DFHEI
EIBFREE DS C X'FF' FREE REQUESTED @BBDIA0U 01-DFHEI
EIBRECV DS C X'FF' RECEIVE REQUIRED @BBDIA0U 01-DFHEI
EIBSEND DS C RESERVED @BM13417 01-DFHEI
EIBATT DS C X'FF' ATTACH RECEIVED @BBDIA0U 01-DFHEI
EIBEOC DS C X'FF' EOC RECEIVED @BBDIA0U 01-DFHEI
EIBFMH DS C X'FF' FMHS RECEIVED @BBDIA0U 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'                          01-DFHEC
L       15,=V(DFHEI1)                       01-DFHEC
BALR    14,15                               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'                          01-DFHEC
L       15,=V(DFHEI1)                       01-DFHEC
BALR    14,15                               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'                          01-DFHEC
L       15,=V(DFHEI1)                       01-DFHEC
BALR    14,15                               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'          ControlLevel @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 ( PPA 2)   *
*****
*
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

```

对全局用户出口的更改

使用 LEASM 选项转换的汇编程序无法用作全局用户出口程序。

LEASM 用于产生符合汇编程序编写的 MAIN 程序的 Language Environment。

对任务相关的用户出口的更改

使用 LEASM 选项转换的汇编程序无法用作任务相关的用户出口程序。

LEASM 用于产生符合汇编程序编写的 MAIN 程序的 Language Environment。

第 4 部分 企业管理

CICS Transaction Server for z/OS V3R1 的 CICSPlex SM 元素提供新的功能，使您能够使用现代用户接口有效管理大量运行时配置，以便您可满足需求服务级别目标。

第 11 章 CICSplex SM Web 用户界面增强

改进了 CICSplex SM Web 用户界面以使它的功能更为强大并且使用更为方便。 Web 用户界面现在的功能等价与 CICSplex SM TSO 最终用户界面，而且现在是访问 CICSplex SM 的主要方法。

用户收藏夹

所有 WUI 用户现在都能够将特定基础上的表格和详细视图保存为易于访问和编辑的收藏夹菜单。这使得您能够只需一次单击就可以到达经常使用的视图。管理员有额外的权限来更新其他用户的收藏夹。

要获得更多信息，请参阅第 247 页的『用户收藏夹』

用户组概要文件

现在，管理员能够为几组用户创建概要文件。这些概要文件包含诸如缺省上下文、作用域、CMAS 上下文、菜单和结果集警告计数之类的信息。采用该方式，管理员可以以不同的方式配置 WUI 来适应不同组的用户，以提供更切合个人需求的界面。

要获得更多信息，请参阅第 251 页的『用户组概要文件』

业务应用程序服务重新设计

改进并简化了对 WUI 的业务应用程序服务部分（用它来管理 CICS 资源定义）的设计，并且现在对于 CICS RDO 的用户而言就更熟悉它了。

要获得更多信息，请参阅第 254 页的『业务应用程序服务重新设计』

记录计数警告

现在可以定制 WUI 以在它打开将生成大量记录的视图之前发出警告。根据警告信息，您可以更改视图上的过滤器，以减少返回的结果数。它通过减少不必要的等待改进 WUI 性能。

要获得更多信息，请参阅第 257 页的『记录计数警告』

过滤器确认

改进了 WUI 视图编辑器，以允许您在创建或更新视图时，在视图打开之前包括一个过滤器确认面板。这意味着浏览到这样的视图时，用户将有机会改正过滤器内容，而无论将返回的结果集会有多大。

要获得更多信息，请参阅第 258 页的『过滤器确认』

动态选择列表

WUI 现在为输入面板中的各个属性生成可能值的列表。它通过允许您从有效值列表中进行选择，而不必记住它们，增强了界面的可用性。

要获得更多信息，请参阅第 259 页的『动态选择列表』

改进的屏幕设计

提供了一些改进，以最大限度地使用 WUI 视图和菜单中的屏幕空间：

- 通过提供在两列中显示信息而不是一列中显示的能力，改进了详细视图上的屏幕空间的使用。可以使用视图编辑器，设计您自己的两列结构的详细视图。
- 通过除去全选和取消全选按钮，并用驻留在表的“记录”标题中的图标替换它们，减少了表格视图上的空格数。

- 您现在可以折叠过滤器，来为数据提供更多的屏幕空间。只要需要，可以随时展开折叠的过滤器。

要获得更多信息，请参阅第 260 页的『改进的屏幕设计』

CICSplex SM Web 用户界面增强的优点

CICS Transaction Server V3 引进了很多对 CICSplex SM Web 用户界面的改进，而这些改进确实能给用户带来好处。

下面是一些新的功能：

第 247 页的『用户收藏夹』

所有 WUI 用户现在都能够将特定基础上的表格和详细视图保存为易于访问和编辑的收藏夹菜单。这使得您能够只需一次单击就可以到达经常使用的视图。管理员有额外的权限来更新其他用户的收藏夹。

第 251 页的『用户组概要文件』

现在，管理员能够为几组用户创建概要文件。这些概要文件包含诸如缺省上下文、作用域、CMAS 上下文、菜单和结果集警告计数之类的信息。采用该方式，管理员可以以不同的方式配置 WUI 来适应不同组的用户，以提供更切合个人需求的界面。

第 257 页的『记录计数警告』

现在可以定制 WUI 以在它打开将生成大量记录的视图之前发出警告。根据警告信息，您可以更改视图上的过滤器，以减少返回的结果数。它通过减少不必要的等待改进 WUI 性能。

第 258 页的『过滤器确认』

改进了 WUI 视图编辑器，以允许您在创建或更新视图时，在视图打开之前包括一个过滤器确认面板。这意味着浏览到这样的视图时，用户将有机会改正过滤器内容，而无论将返回的结果集会有多大。

第 259 页的『动态选择列表』

WUI 现在为输入面板中的各个属性生成可能值的列表。它通过允许您从有效值列表中进行选择，而不必记住它们，增强了界面的可用性。

第 254 页的『业务应用程序服务重新设计』

改进并简化了对 WUI 的业务应用程序服务部分（用它来管理 CICS 资源定义）的设计，并且现在对于 CICS RDO 的用户而言就更熟悉它了。

第 260 页的『改进的屏幕设计』

提供了一些改进，以最大限度地使用 WUI 视图和菜单中的屏幕空间：

- 通过提供在两列中显示信息而不是一列中显示的能力，改进了详细视图上的屏幕空间的使用。可以使用视图编辑器，设计您自己的两列结构的详细视图。
- 通过除去全选和取消全选按钮，并用驻留在表的“记录”标题中的图标替换它们，减少了表格视图上的空格数。
- 您现在可以折叠过滤器，来为数据提供更多的屏幕空间。只要需要，可以随时展开折叠的过滤器。

需求

要支持此功能，没有特别的硬件或软件需求。

相关信息

第 311 页的第 27 章，『CICS操作环境』


对 CICSplex SM 的更改

用户收藏夹

本主题描述 CICSplex SM Web 用户界面的新的用户收藏夹功能。


用户收藏夹向 WUI 的各个用户提供一种能力，使得他们能够将所选的 WUI 屏幕保存到他们可以快速和方便地访问的特定菜单。一组收藏夹总是只与一个用户相关而不是与用户组相关。然而，使用新的用户组概要文件设施以及视图编辑器，用户组可以得到类似的结果（请参阅第 251 页的『用户组概要文件』以获取指导）。

用户收藏夹的信息保存在称为“用户”的新对象中。用户对象保存包含用户收藏夹的超链接的菜单组。有两种创建用户对象的方法：

- 它可以在用户第一次创建收藏夹时创建。用户通过单击  图标创建收藏夹时，WUI 搜索有与用户标识匹配的标识的用户对象。如果无法找到匹配的用户对象，则 WUI 服务器为该用户自动创建匹配的用户对象。
- 它可以在用户编辑器中由管理员创建。请参阅第 249 页的『为其他 WUI 用户创建和管理收藏夹』以获取更多信息。

用收藏夹编辑器管理收藏夹

本主题描述您可以如何使用新的 Web 用户界面收藏夹编辑器管理您自己的用户收藏夹。

您能够使用  图标将视图添加到您自己的收藏夹列表，并且还可以使用收藏夹编辑器处理您的收藏夹。要打开此编辑器，单击 WUI 导航框的特别部分中的**收藏夹编辑器**。该操作会在新的浏览器窗口中打开编辑器，就象第 248 页的图 30 中显示的那样。

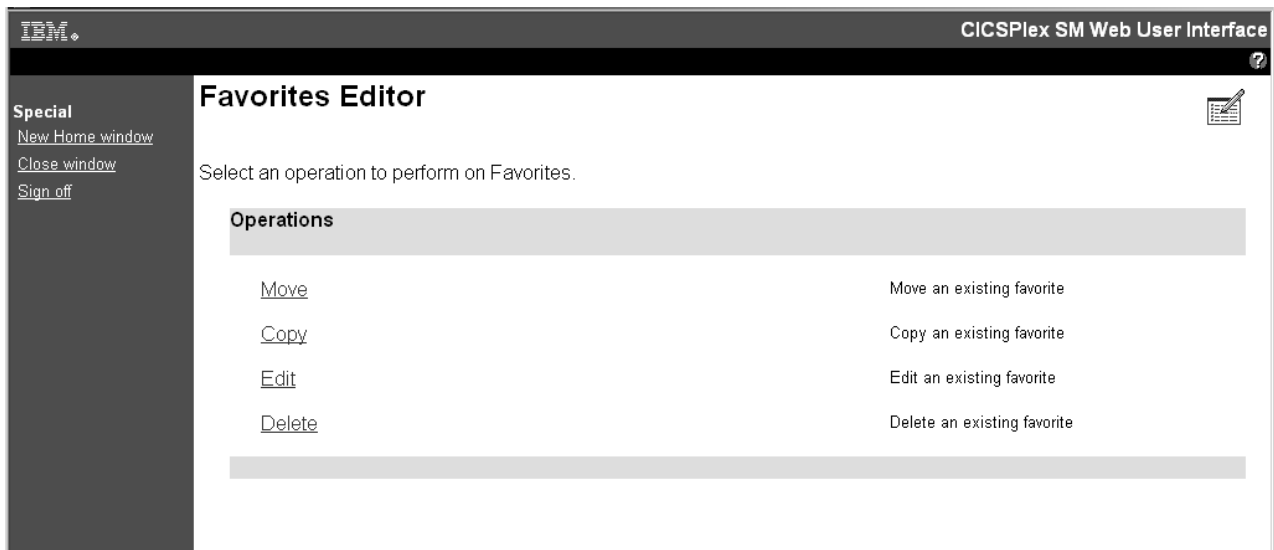


图 30. 收藏夹编辑器

要修改收藏夹，选择单选按钮并单击视图底部的 4 个按钮中的一个。

1. 要编辑屏幕，单击**编辑**。从列表中选择您要编辑的收藏夹，然后单击**确定**以打开**收藏夹的组件**屏幕。在此屏幕上，您可以选择更改菜单的标题和注释或者目标。选择**目标**以更新它的上下文和作用域设置以及过滤器设置。
2. 要移动收藏夹列表上的某项的位置，单击**移动**。从列表选择一个收藏夹，并单击**确定**。该操作会打开**移动收藏夹**屏幕。为所选项选择新的位置，并再次单击**确定**以做出更改。
3. 要复制收藏夹屏幕，单击**复制**。从列表选择一个收藏夹，并单击**确定**。该操作会打开**复制收藏夹**屏幕，就象第 249 页的图 31 中显示的那个屏幕。



图 31. “复制收藏夹”屏幕

现在，您可以为副本起一个新的标题。通过从上面的列表中选择现有的收藏夹，在收藏夹列表上选择它的位置，新的副本将会放在那个位置。如果您要将新的副本放在列表的底部，不要选择现有的收藏夹。单击**确定**确认操作。

4. 要删除收藏夹，单击**删除**。从列表中选择一个收藏夹，并单击**确定**以确认。

收藏夹编辑器允许您编辑自己的收藏夹列表。但不允许您管理其他用户的收藏夹。只有具有用户编辑器访问权的 WUI 管理员有管理其他用户的收藏夹的权限。

为其他 WUI 用户创建和管理收藏夹

本主题描述管理员可以如何使用新的 Web 用户界面用户编辑器管理其他用户的收藏夹。

要创建和管理其他用户的收藏夹，您需要的是至少对新的 ESM 设施概要文件 EYUWUI.wui_server_applid.USER 具有更新访问权的 WUI 管理员。

第 250 页的图 32 例举用于编辑用户收藏夹的用户编辑器中的屏幕。

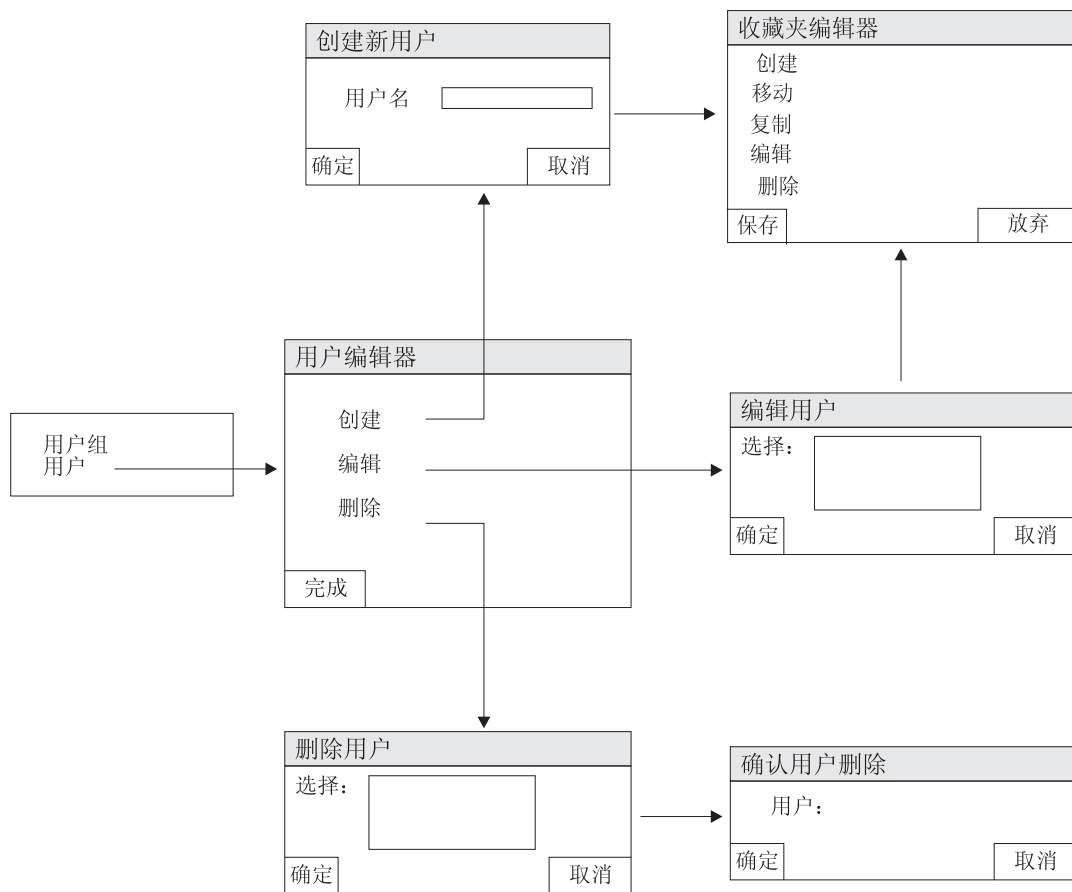


图 32. 用户编辑器中的用户屏幕

如果您有必需的访问权，导航框会出现启动用户编辑器的超链接以及视图编辑器超链接。单击此超链接将启动新的浏览器窗口，该窗口显示 **CICSplex SM Web 用户界面** 用户编辑器屏幕。要处理收藏夹，选择**用户**。该操作会打开**用户编辑器**屏幕。此屏幕向您提供下列选项：

- 创建** 使用此选项在存储库中创建新的用户对象。每个用户都需要用户对象来保存收藏夹列表。WUI 在用户第一次创建收藏夹时自动创建用户对象（如果还不存在一个用户对象的话），因此多数现有的用户已经有一个用户对象了。
- 编辑** 使用此选项处理现有用户的收藏夹。它允许您创建、移动、复制、编辑和删除用户的收藏夹。
- 删除** 使用此选项从存储库中删除用户对象。它还会删除该用户的任何现有收藏夹。

以下描述您可以如何使用用户编辑器创建新的用户对象和添加一些收藏夹：

1. 创建新的用户对象
 - a. 从**用户编辑器**屏幕开始，单击**创建**以打开**创建新的用户**屏幕。
 - b. 输入新的用户标识。用户标识限制为最多 8 个字符。有效字符为 A-Z、0-9、#、\$ 和 @。

- c. 单击**确定**。它将确认操作并且打开用于创建和管理用户的收藏夹列表的**收藏夹编辑器**屏幕。
2. 创建新的收藏夹项。
 - a. 在**收藏夹编辑器**屏幕中，单击**创建**。该操作会打开视图编辑器使用的编辑屏幕，以供您编辑详细项。您可以在该屏幕上对收藏夹执行大多数操作，这些操作是您可以对视图编辑器中的菜单项执行的操作。
 - b. 通过为新的收藏夹起一个标题并输入目标视图或菜单、上下文和作用域设置以及过滤器设置，完成该屏幕。
 - c. 单击**保存**。该操作在存储库中保存更新，并把您带回到**用户编辑器**屏幕。如果您单击**放弃**，您的所有更改（包括新的用户对象）都将废弃。
3. 创建更多的收藏夹并把它们添加到新的用户对象。
 - a. 在**用户编辑器**屏幕中，单击**编辑**。该操作会打开**编辑用户**屏幕。
 - b. 选择新创建的用户标识并单击**确定**。它会把你带回到**收藏夹编辑器**屏幕。
 - c. 象前面一样创建新的收藏夹项，并保存您的更新。为每个您要添加到新的用户对象的收藏夹重复此步骤。

相关信息

第 247 页的『用收藏夹编辑器管理收藏夹』

本主题描述您可以如何使用新的 Web 用户界面收藏夹编辑器管理您自己的用户收藏夹。

用户组概要文件

本主题描述 CICSplex SM Web 用户界面 (WUI) 的新的用户组概要文件功能。

这个新的功能使得管理员能够为 WUI 用户组创建概要文件。个人用户登录到 WUI 时将看到的内容以及执行的操作可以由他们所属的用户组概要文件控制，该概要文件使得用户能够定制 WUI 以切合各个用户组的需要。例如，您可以使用视图编辑器创建新的仅包含操作视图的 WUI 菜单，并且让它成为一组用户的缺省菜单。它将为只需要执行此类任务的用户提供简化的可操作 WUI。

用户组概要文件可以包含以下信息：

- 结果集警告计数（请参阅第 257 页的『记录计数警告』）
- 缺省主菜单的名称
- 缺省导航菜单的名称
- 缺省上下文
- 缺省作用域
- 缺省 CMAS 上下文

如果概要文件未指定这些值中的某些值；例如，如果缺省菜单值空白，则使用 JCL 中指定的对应的 WUI 参数值。对于概要文件中设定的任何无效值也采取这样的处理方式，例如，如果指定的缺省菜单不存在。

使用新的编辑工具（用户编辑器）创建和管理用户组概要文件，具有必需权限的管理员可以从 WUI 主菜单访问该编辑器。请参阅第 252 页的『创建和管理用户组概要文件』以获取更多信息。

要更方便地使用它，引进了称为用户组的新类型对象。用户组对象（例如视图和菜单对象）可以使用 COVC 事务导入和导出。

仅当 WUI 中的安全性活动时，才可以使用用户组概要文件。用户组对象与外部安全性管理器（ESM）中的用户组名称相关。用户登录到 WUI 时，登录过程从用户的 ESM 获取访问控制环境元素（ACEE）。使用 ACEE 获取用户的缺省组。然后使用缺省组名获取 WUI 中的相应组对象。如果当用户登录时，未在 WUI 高速缓存中找到相应的组对象，则为该用户使用 JCL 的 WUI 参数中指定的缺省值。

注：在 ESM 中，用户组只可以包含下列字符：

- A 到 Z（您可以输入小写字符，但是会把它们折换成大写字符）
- 0 到 9
- # (X'23')、\$ (X'24') 和 @ (X'40')

为了管理用户组概要文件，管理员需要有对名为 EYUWUI.wui_server_applid.USER 的新的 ESM 设施概要文件的更新访问权，其中 wui_server_applid 是与此概要文件相关的 WUI 服务器的应用程序标识。

设置新的用户组包括下列步骤：

1. 使用用户编辑器在 WUI 中创建新的用户组概要文件（请参阅『创建和管理用户组概要文件』以获取指导）。
2. 在 ESM 中用相同的名称创建用户组。
3. 在 ESM 中将用户组设置为一个或多个用户的缺省用户组。

创建和管理用户组概要文件

本主题描述如何创建和管理 Web 用户界面（WUI）用户组概要文件。

使用称为用户编辑器的新的 WUI 编辑工具，创建和管理用户组概要文件。

导航框中显示启动用户编辑器的超链接以及视图编辑器超链接。仅当已登录的用户对新的 ESM 设施概要文件 EYUWUI.wui_server_applid.USER 至少有更新访问权时才可以看到用户编辑器超链接。单击此超链接将启动新的浏览器窗口，该窗口使用一组为用户编辑器创建的新视图。

启动用户编辑器时，您可以选择编辑用户组或用户。第 253 页的图 33 中的图例举了用户编辑器用户组部分中的屏幕之间的关系。

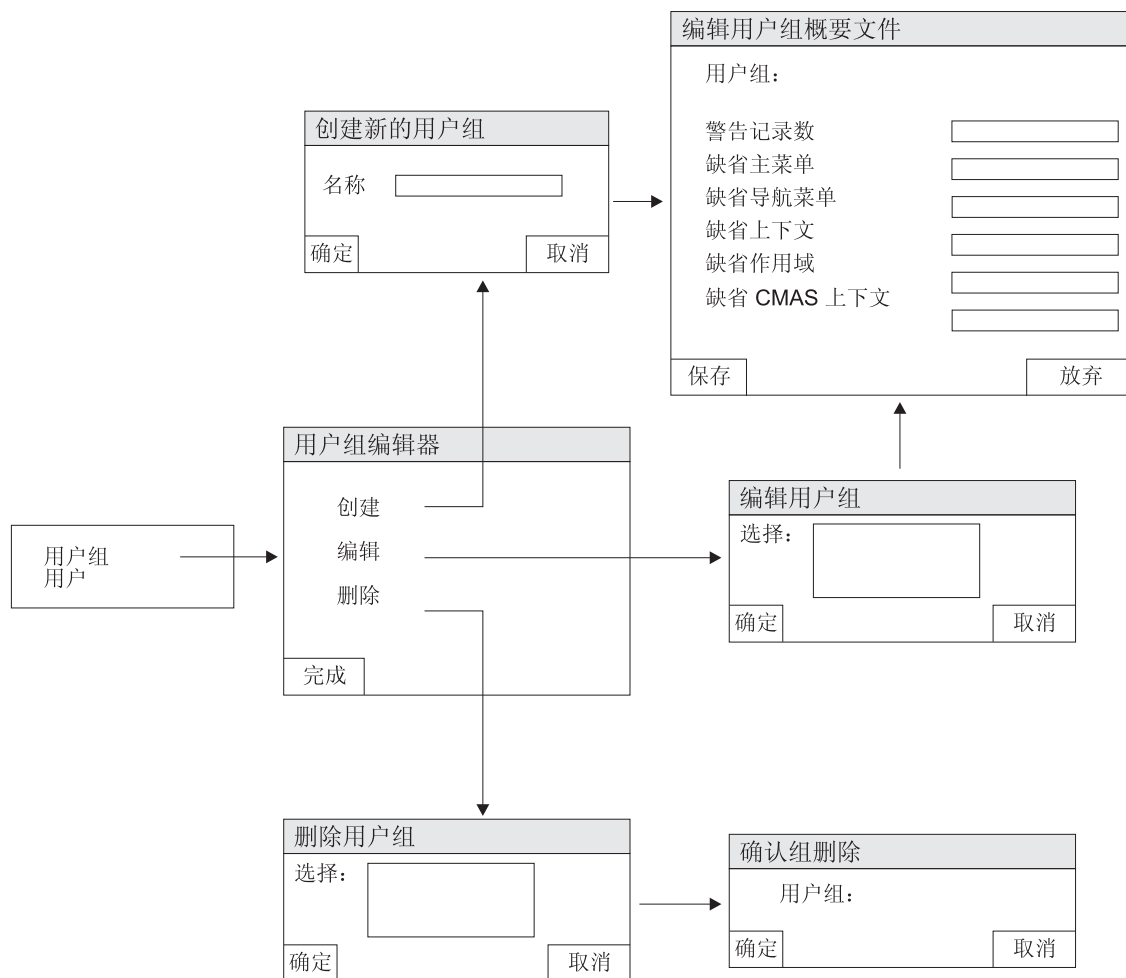


图 33. 用户编辑器中的用户组屏幕

使用用户编辑器是最直接的方法。例如，要创建新的概要文件，按照以下过程执行操作：

1. 浏览至**创建新的用户组**屏幕：
 - a. 单击导航框中的**用户编辑器**以打开新的 **CICSplex SM Web 用户界面用户编辑器**窗口。
 - b. 单击**用户组**以打开**用户组编辑器**屏幕。该屏幕向您提供创建、编辑或删除概要文件的选项。
 - c. 选择**创建**以打开**创建新的用户组**屏幕。
2. 命名新的组概要文件：
 - a. 输入新组的名称。（该名称将需要与 ESM 中定义的用户组名相匹配。）名称限制为最多 8 个字符。与 ESM 中的一样，允许的字符为 A-Z（大写）0-9、#、\$ 和 @
 - b. 单击**确定**。该操作确认操作并打开**编辑用户组概要文件**屏幕。
3. 输入用户组详细信息。请参阅第 251 页的『用户组概要文件』以获取此信息的描述。
4. 单击**保存**以创建新的概要文件或单击**放弃**以取消操作。

要编辑组，单击**用户组编辑器**屏幕中的**编辑**以打开**编辑用户组**屏幕，并选择一个现有的组。该操作会打开包含所选组的详细信息的**编辑用户组概要文件**屏幕。进行更改，然后单击**保存**以更新概要文件或单击**放弃**以取消操作。

要删除组，单击**用户组编辑器**屏幕中的**删除**以打开**删除用户组**屏幕，并选择一个现有的组。单击**确定**以打开**确认用户组删除**屏幕，并再次单击**确定**确认操作。

业务应用程序服务重新设计

本主题描述对 WUI 中的 CICS 资源定义使用的业务应用程序服务视图和菜单的设计的改进。

在新的 WUI 设计中，BAS 功能划分成基本视图菜单和完全功能视图菜单。要从 WUI 主菜单访问 BAS 功能，单击**管理视图**。在“管理视图”的底部有两个新的子菜单：

基本 CICS 资源管理视图

它提供类似于 RDO 的简化 BAS 模型，包含资源定义、资源组和资源描述，但不包含资源分配。

完全功能业务应用程序服务 (BAS) 管理视图

除了基本模型之外，它还包含到瞄准更高级用户的资源分配视图的链接。这会大大增添您管理资源定义的能力和灵活性。

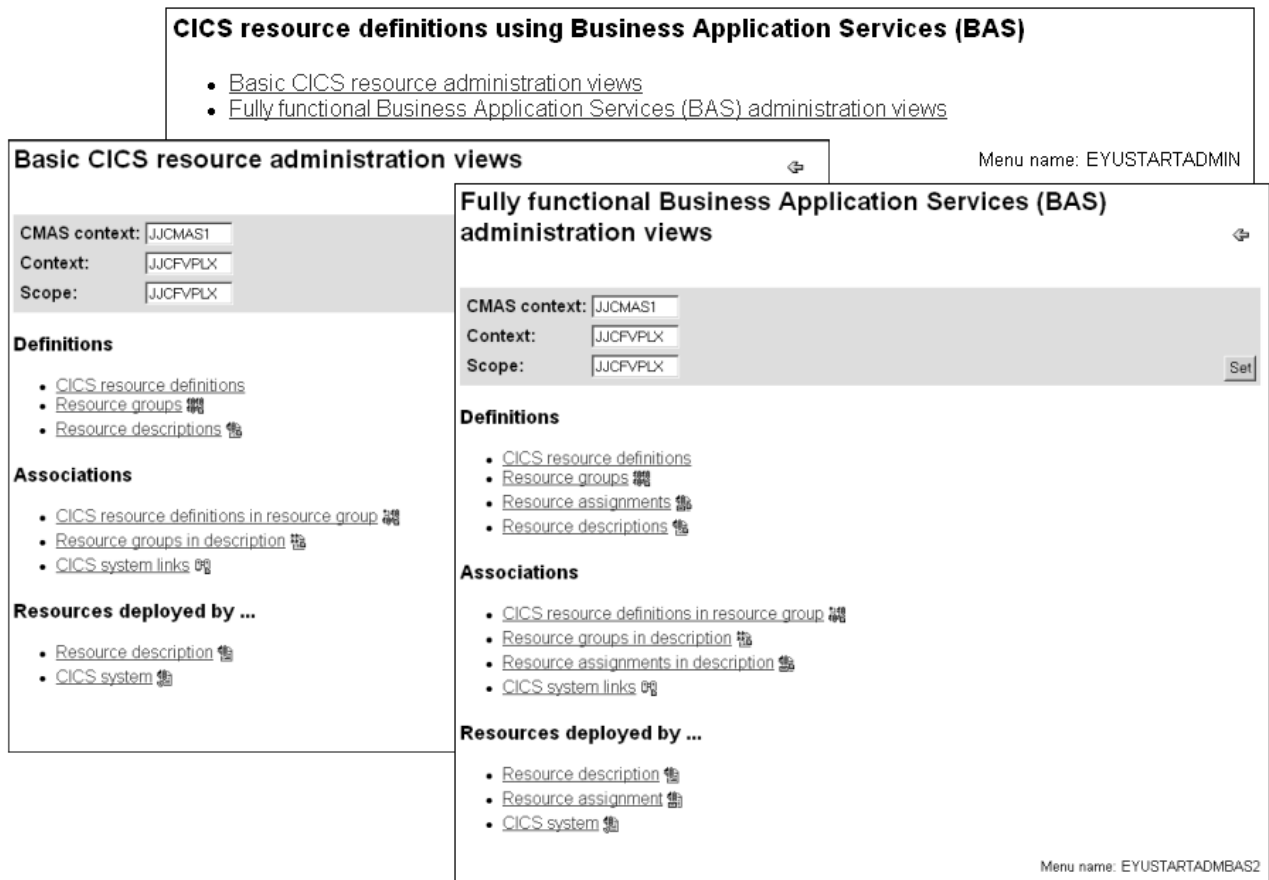


图 34. 新的 CICS 资源定义 BAS 菜单的详细信息

来自这两个菜单的链接划分成三组：

定义 包含下列链接：

CICS 资源定义

包含到每个资源类型的定义视图的链接的菜单。

资源组 到用于管理资源组定义的定义视图的链接。关联的操作为“创建”、“更新”、“除去”、“安装”和“添加到资源描述”。

资源分配（仅完全功能菜单才有）

到用于创建和管理资源分配的定义视图的链接。关联的操作为“创建”、“更新”、“除去”和“添加到资源描述”。

资源描述

到用于创建和管理资源描述的定义视图的链接。关联的操作为“创建”、“更新”、“除去”、“安装”和“替换”。

关联 包含下列链接：

资源组中的 CICS 资源定义

到新的 RESINGRP 资源表支持的新视图（如第 256 页的图 35 中所示）的链接。这会替换前发行版的资源组中的资源视图集（每个资源类型一个视图）。

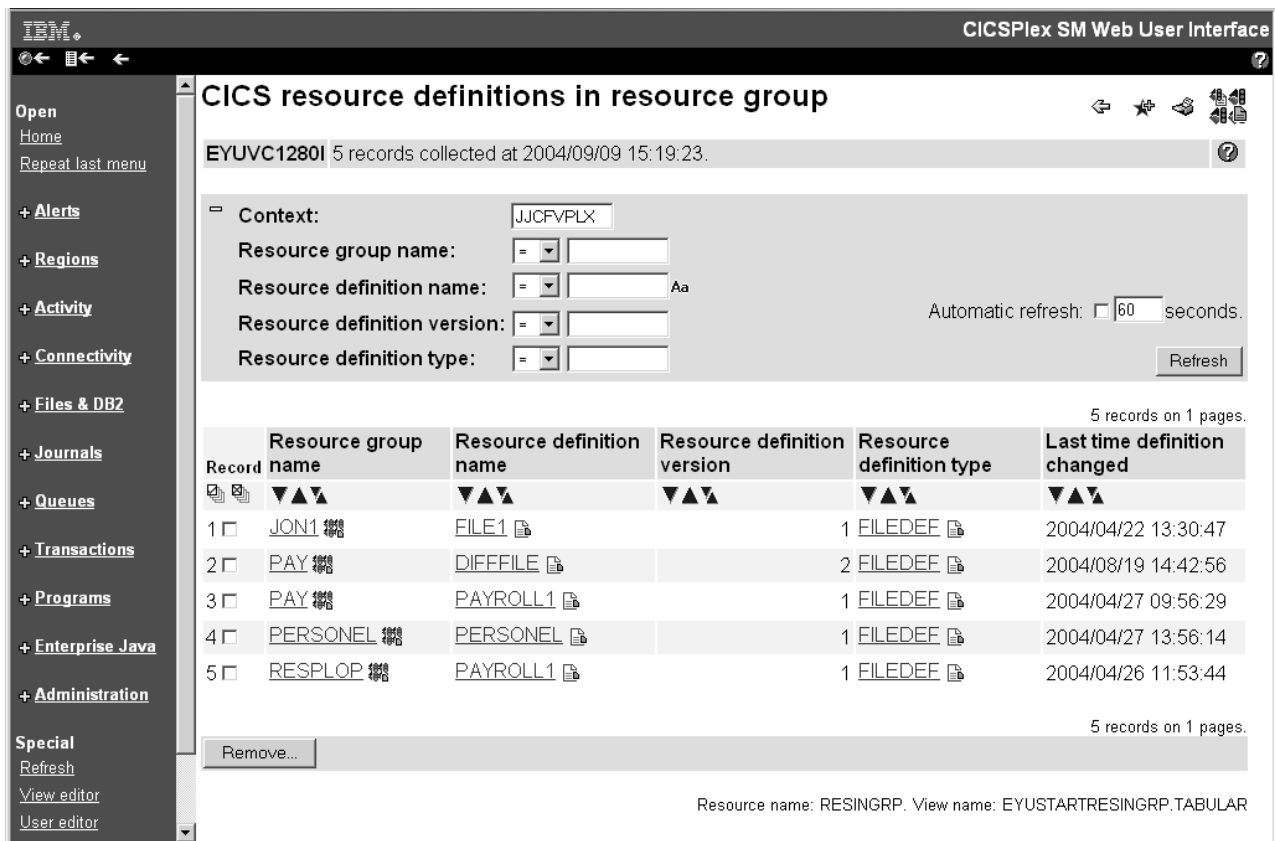


图 35. 资源组表格视图中的 CICS 资源定义

视图包含**除去**操作按钮，该按钮允许您除去资源定义和它的父资源组之间的关联。此视图没有创建操作。定义资源本身时执行将资源添加到组

描述中的资源组

到用于管理资源组和资源描述之间的关联的定义视图的链接。关联的操作为“创建”、“更新”和“除去”。

描述中的资源分配（仅完全功能菜单才有）

到用于管理资源分配和资源描述之间的关联的定义视图的链接。关联的操作为“创建”、“更新”和“除去”。

CICS 系统链接

到用于管理 CICS 系统链接定义的定义视图的链接。关联的操作为“创建”、“除去”和“安装”。

已部署资源，按照...

包含下列到显示活动的 CICS 资源的视图的链接：

资源描述

到显示资源描述选择的已部署资源的表格视图的链接。

资源分配（仅完全功能菜单才有）

到显示资源分配选择的已部署资源的表格视图的链接。

CICS 系统

到显示 CICS 系统选择的已部署资源的表格视图的链接。

记录计数警告

本主题描述如果某些视图中返回的期望记录数超过设置的阈值时 WUI 能够如何发出警告。

设计这个新的功能是用来在请求要打开某些视图，而它将导致超过要求的数据量时，发出警告。该功能提供一个机会，使得您能够更改这些视图上的过滤器，并在执行请求之前确认或取消请求。

与下列资源关联的视图中应用此功能:

CMDT	FEPINODE	PROGRAM
CONNECT	FEPPOOL	REMFIL
DB2CONN	FEPITRGT	REMTDQ
DB2ENTRY	INDTDQ	REMTRAN
DB2TRN	INTRATDQ	RQMODEL
DOCTEMP	JOURNAL	SMFJRNL
DSKJRNL	JRNLNAME	SYSDUMP
DSNAME	LOCFILE	TAPEJRNL
ENQMODEL	LOCTRAN	TCPIPS
EXITGLUE	MODENAME	TERMNL
EXITTRUE	PARTNER	TRAN
EXTRATDQ	PROCTYP	TRANDUMP
FEPICONN	PROFILE	TSMODEL

有两种方法可以用来指定触发警告机制必需的记录数:

- 在 WUI 服务器配置期间，为新的 DEFAULTWARNCNT WUI 服务器初始化参数设置一个值。DEFAULTWARNCNT 可以接受范围在 0 到 99999999 之间的整数值。这是可选的参数。
- 在用户组概要文件中设置警告记录计数值。这允许 WUI 管理员为 WUI 用户组的所有成员设置值。请参阅第 252 页的『创建和管理用户组概要文件』以获取更多信息。

在这两种情况下，缺省值为 0 意味着不发出警告。

用户组中的值集优先于 DEFAULTWARNCNT 参数中的值集。发出警告之前，WUI 检查以确定已登录的用户是否与用户组关联（仅当 WUI 在启用安全性检查的情况下运行时，它才可能发生）。如果关联，WUI 使用该用户组中为警告计数指定的任何值。如果用户不属于用户组，或者未设置最大值，则 WUI 会检查初始化参数并且使用那里设置的任何值。如果未指定值，则使用缺省值 0。

如果警告计数的值大于 0，则 WUI 会检查要返回的可能记录集的大小。它指出可返回的最大记录数。此数字并非总是精确的，因为除了与主键字段的第一个部分关联的过滤器之外的任何过滤器都不记算。如果这个可能的值大于警告计数的值，则显示类似于第 258 页的图 36 中显示的屏幕的警告屏幕。如果返回值小于或等于警告计数值，则以常规方式显示 WUI 视图。

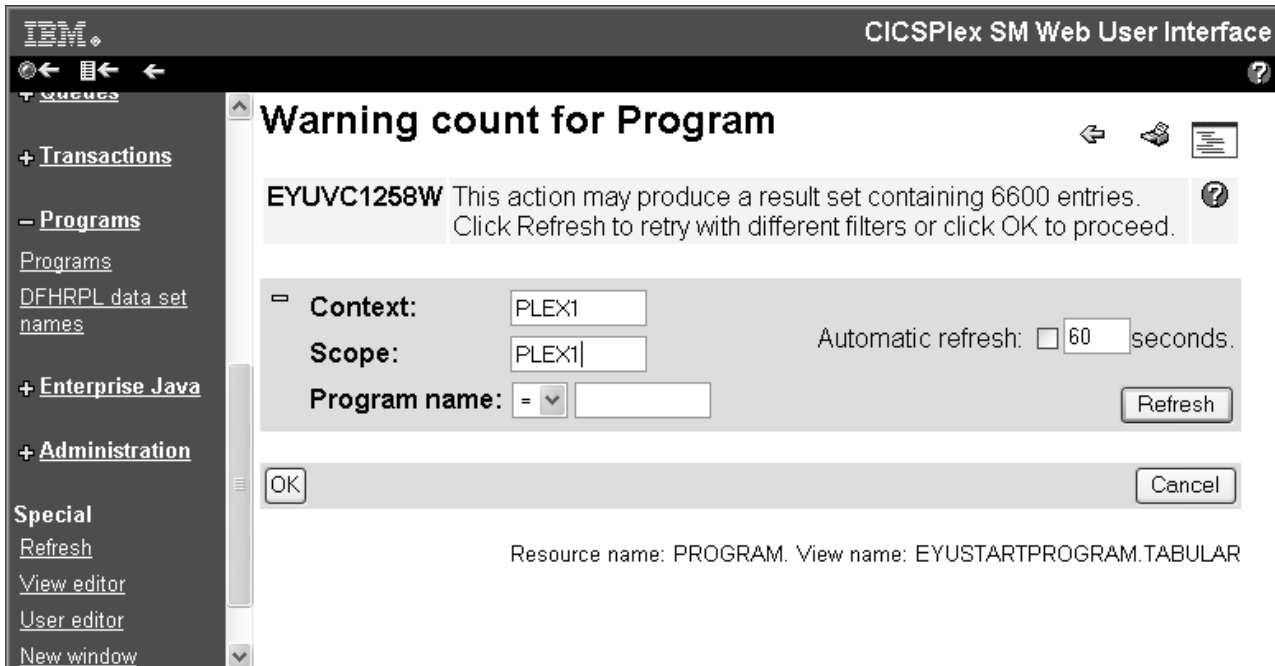


图 36. “警告计数”屏幕

单击**刷新**会发出一个新的请求。该操作允许您更改包含在**警告计数**屏幕中的主键字段，以测试新的过滤器值，目的是产生更小的结果集。原始请求的任何其他过滤器仍在适当的地方。如果返回的结果集大于等于警告计数限制值，则会重新显示警告计数屏幕，并显示原始的警告计数消息。如果大小小于警告计数限制值，则会重新显示警告计数屏幕。

单击屏幕底部的**确定**会执行当前的请求，而无论要返回的结果数有多少。

单击**取消**等同于在浏览器上按后退链接；会重新显示上一个屏幕。

过滤器确认

本主题描述 CICSPlex SM Web 用户界面的新的过滤器确认功能。

现在您可以在 WUI 以类似于 TSO EUI 采取的方式检索视图的数据之前，指定过滤器条件。

在视图编辑器中设置您自己的视图时，现在有一个选项，您可以指定在执行数据检索之前应该显示过滤器确认屏幕。此更改不影响 WUI 提供的视图。第 259 页的图 37 中显示典型的过滤器确认屏幕。

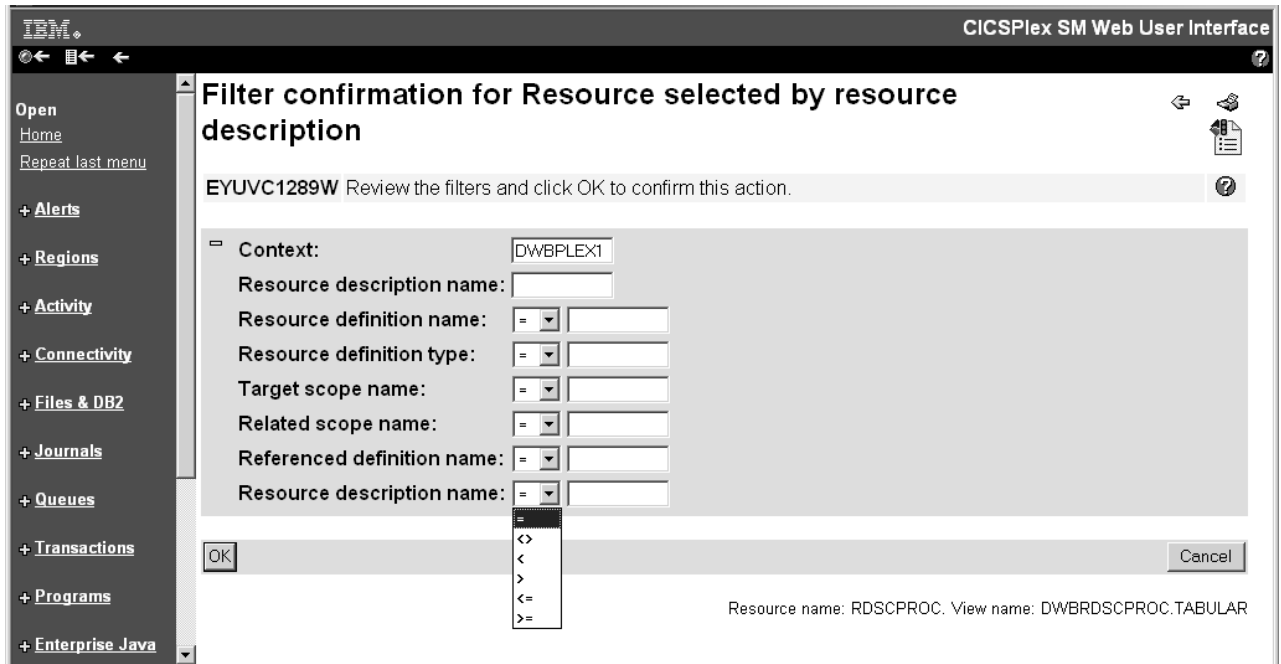


图 37. “过滤器确认”屏幕


您可以通过重新输入字段并单击**确定**，覆盖确认屏幕中的过滤器值。

注：过滤器确认总是在处理任何结果集警告之前应用。

动态选择列表

本主题描述 CICSPlex SM Web 用户界面（WUI）的新的动态选择列表功能。

WUI 现在能够以类似于 TSO EUI 采取的方式生成动态选择列表。能够生成选择列表的

字段用文本输入框右边的图标  标记。对于资源输入字段，单击该图标将打开一个屏幕，如下面所示：

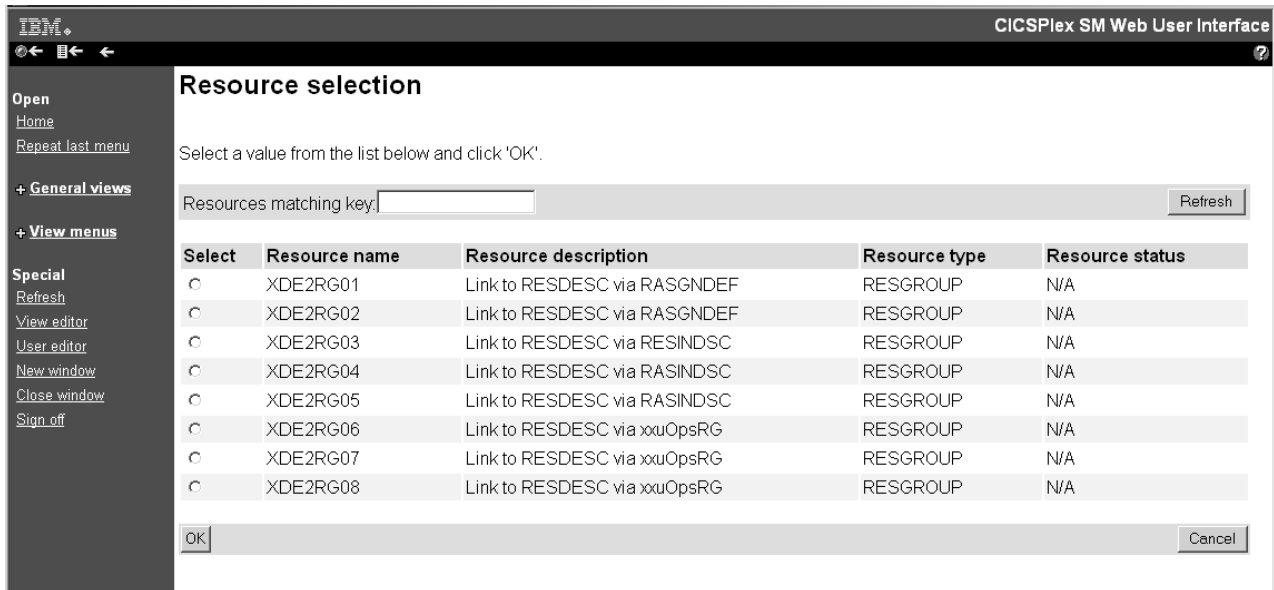


图 38. 资源选择列表屏幕

在屏幕上以表格的形式显示有效值，这些值按资源类型排序。要从列表中进行选择，您只需要选择单选按钮，并单击**确定**。

选择列表屏幕包括一个输入字段，该字段包含过滤列表当前在使用的值。要使用新的过滤器，在输入字段中输入新值，并单击**刷新**。

仅任务指导、表格和详细视图的输入字段中的属性和参数能够生成选择列表。过滤器、上下文和作用域字段不能生成选择列表。

元数据属性请求的选择列表屏幕显示按属性名排序的基本表的所有属性或经过滤的子集。它们主要用于求值定义创建。

您可以通过指定过滤器值，减少选择列表中显示的值的数目。这些过滤器与通常在 WUI 中出现的过滤器不同，因为它们仅使用 =（等于）运算符。您可以使用通配符 *（星号）和 +（加号）以常用的方式输入一般值作为属性值，例如 DW*。然后在选择列表屏幕上，您可以通过修改屏幕顶部的框中显示的过滤器值，更改列表中显示的值。

注：如果您输入的值不包含一般字符，此值仍将用作过滤器，而不管是否将在选择列表中显示任何其他值。这与当前 EUI 的行为不同，当向 EUI 提供了不包含通配符的属性时，如果未产生值，则产生所有可用值的列表。

改进的屏幕设计

本主题描述改进 WUI 视图和菜单的可用性及外观的新功能。

两列结构的详细视图

引进两列结构的详细视图增加了可以在您的屏幕上显示的数据量，并且减少屏幕滚动的需要。下面是两列结构的视图的一个示例：

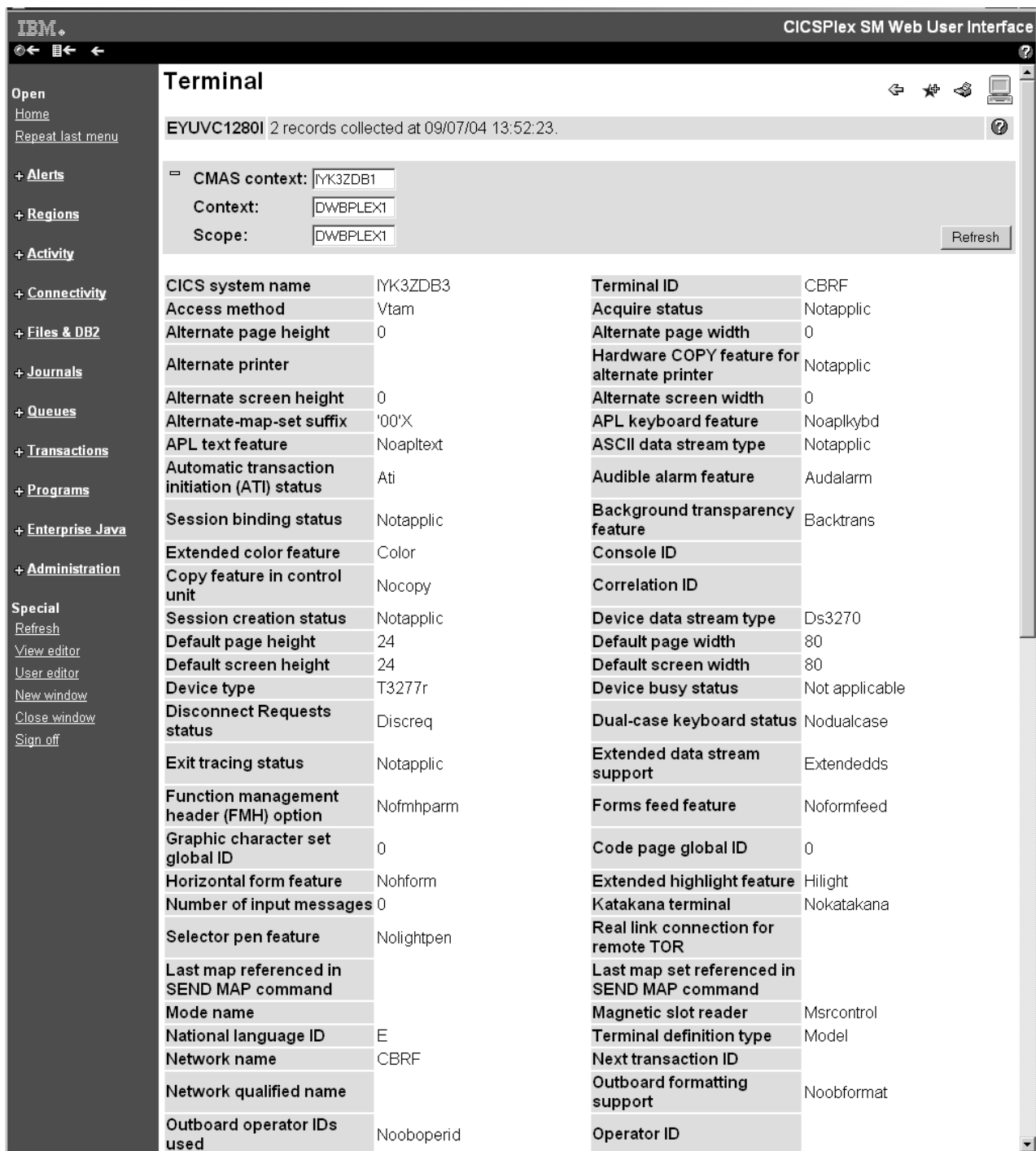


图 39. 部分两列结构的详细视图

您可以使用视图编辑器创建两列结构的详细视图。请参阅第 263 页的『创建两列结构的详细视图』以获取指导。

展开和折叠过滤器

表格视图上的过滤器可能会占用大量屏幕空间，其结果是限制了可以显示的数据。您现在可以通过在不更改过滤器的时候把它们折叠起来，以缓解此问题。

您可以在想更改它们的时候，再将它们展开。图 40 显示在展开了过滤器的情况下的一部分表格视图。

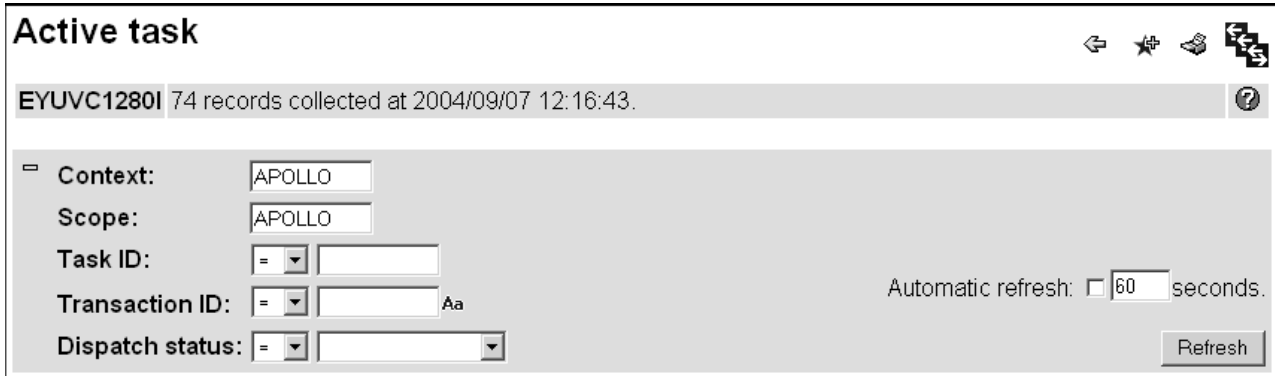


图 40. 表格视图的展开的过滤器区域

要折叠过滤器，单击过滤器区域左上方的折叠（减号）图标。

图 41 显示在折叠了过滤器的情况下的同一个屏幕。显示的信息量相同，但数据字段超出了屏幕。

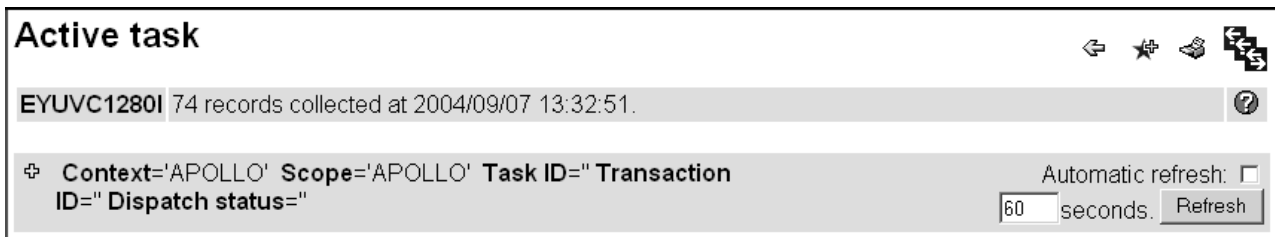


图 41. 折叠的过滤器区域

要展开过滤器，单击“展开过滤器”（加号）图标。

要确定缺省情况下过滤器是应该以展开的状态显示还是以折叠的状态显示，使用这个新的可选 WUI 服务器初始化参数：

FILTERSTYLE(EXPAND/COLLAPSE)

如果您不指定此参数，则 WUI 在缺省情况下以展开的状态显示过滤器。

如果您在视图外浏览并且过滤器处于非缺省的状态并查看一些其他的屏幕，则新视图中的过滤器区域将回复为 **FILTERSTYLE** 参数中指定的缺省过滤器设置。

如果您在表格视图外浏览，然后立即返回，则过滤器区域的状态仍处于最后一次显示视图时您让其所处的状态。

“全选”和“取消全选”图标

要改进表格视图上的空间利用，已将前发行版中用于全选和取消全选与给定条件匹配的资源的全选和取消全选按钮替换为下列图标：



全选



取消全选

两个图标都在表格视图的标注为**记录**的列的第二行中（在与排序和摘要图标相同的表行中）。

另外，“全选”和“取消全选”选项的功能未更改。

创建两列结构的详细视图

本主题说明如何使用 WUI 视图编辑器创建和更新两列结构的详细视图。

更新了 WUI 视图编辑器以允许创建和更新两列结构的详细视图。更新了视图编辑器的**添加视图**屏幕，以包括称为**两列结构的详细表单**的新视图类型。前发行版的**详细表单**类型已重命名为**一列结构的详细表单**。

创建新视图的过程与前发行版中的过程几乎完全相同。下面是一些步骤的概要，您需要按照这些步骤将新的两列结构的详细视图添加到现有视图集：

1. 打开视图编辑器并浏览至**添加视图**屏幕。
 - a. 在主菜单导航面板中，单击**查看编辑器 > 查看集 > 编辑**。该操作会打开**打开视图集**屏幕。
 - b. 选择新的详细视图将属于的视图集，并单击**确定**。

注：请记住，不可能更改 IBM 提供的视图集或视图，除非您先复制它并对它进行重命名。

该操作会打开**视图集目录**屏幕，它包含当前属于所选视图集的视图列表。

- c. 单击屏幕底部的**添加**。该操作会打开**添加视图**屏幕，用它来为所选视图集创建新的视图。
2. 命名新的视图并定义它的类型。
 - a. 在**视图名称**字段中输入您的新视图的名称。
 - b. 从视图类型列表中选择**两列结构的详细表单**。
 - c. 从预填充选项列表中选择**关键属性**，并单击**确定**。该操作会打开**详细表单组件**屏幕。
 3. 将属性添加到左列

您选择将一项添加到两列结构的详细视图时，新的元素会被放置在左列。将在与新项对应的右列中自动放置一个空白元素，以维持左右对齐。

- a. 选择**表单内容**。该操作会打开**表单内容**屏幕，该屏幕类似于第 264 页的图 42 中显示的屏幕。注意，它包含一个空白元素类型（在创建两列结构的屏幕时这是必需的），以维持左右两个列的平衡。

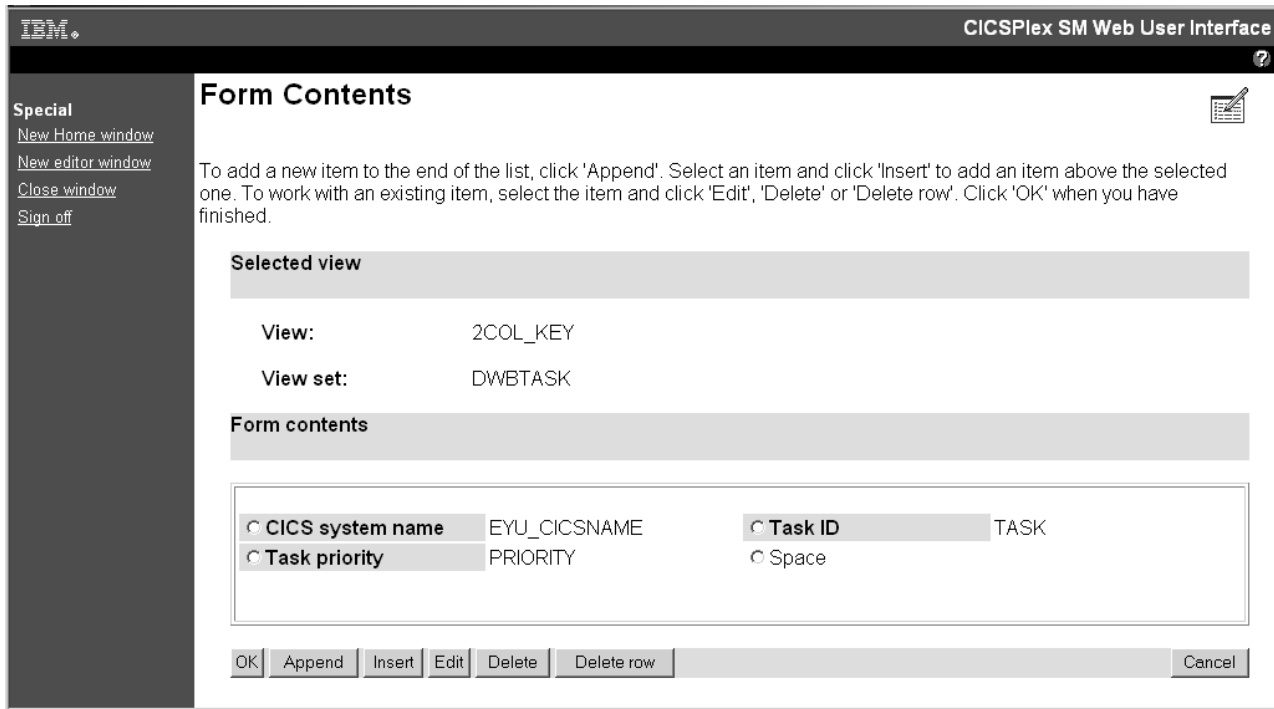


图 42. 两列结构的详细视图的“表单内容”屏幕

- b. 单击**追加**。 该操作会打开**表单项类型**屏幕。
 - c. 选择**属性字段**并单击**确定**以打开**表单属性字段**屏幕。
 - d. 从列表中选择属性，并单击**确定**以打开**表单项组件**屏幕。
 - e. 按以下步骤完成新属性的定义：
 - 输入属性标题和任何注释
 - 选择显示选项；常规或图形
 - 添加任何视图链接。
 - f. 单击**完成**将新的属性添加到您的新视图。 该操作将把您带回到**表单内容**屏幕。
4. 将另一个属性添加到视图的右列。

要将元素添加到右列，您需要选择空白元素，然后编辑它以将它更改为需要的元素类型：

- a. 选择在步骤 3 中创建的空白元素，并单击**编辑**。 该操作会再次打开**表单项类型**屏幕。
- b. 选择您要添加到右列的元素类型，并按照步骤 3 中概述的过程定义它。

单击**完成**将右边的元素添加到视图

您可以根据需要重复执行任意次数的步骤 3 和步骤 4，把更多的元素添加到左列和右列。

如果您要从右列中除去一项，选择它并单击**删除**。该操作会把项转换回空白元素。

注：您无法删除独立的空白元素，但您可以通过单击**删除行**除去两个列中的邻近元素。

5. 完成视图定义。

- a. 在表单内容屏幕上单击**确定**，返回给详细表单组件屏幕。
- b. 现在添加新视图的其余组件，包括标题、操作按钮、过滤器、上下文和作用域等等。
- c. 添加了所有必需的组件后，单击**确定**保存新的视图并返回给视图集目录屏幕。

对 CICSPlex SM API 的更改

已对 CICSPlex SM API 进行一些更改，以支持新的业务应用程序服务设计。

支持业务应用程序服务重新设计中的 API 更改

引进了一个新的名为 RESINGRP 的资源表。RESINGRP 资源描述资源组中的资源定义的成员资格。它有下列属性：

表 12. RESINGRP 资源表属性

名称	数据类型	源	长度	摘要	描述
CHANGETIME	DATETIME	CPSM	8	MAX	最近一次更改的定义
RESGROUP-1	CHAR	CPSM	8	DIF	资源组名
RESNAME-2	CHAR	CPSM	8	DIF	资源定义名
RESVER-3	BINARY	CPSM	1	AVG	资源定义版本
RESTYPE-4	CHAR	CPSM	8	DIF	资源定义类型
DESCRIPTION	CHAR	CPSM	58	DIF	资源定义描述

消息

向 CICSPlex SM Web 用户界面添加了很多新消息。

新的 Web 用户界面客户机消息

EYUVC1258W	EYUVC1289W	EYUVC1293I	EYUVC1294I
EYUVC1310I	EYUVC1311I	EYUVC1312W	EYUVC1313E
EYUVC1314I	EYUVC1321E		

新的 Web 用户界面视图编辑器消息

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 批处理资源库更新工具的增强

引入了新的方法，它们提供了能替换现有 TSO 最终用户界面 BATCHREP 命令的方法，向指定的 CICSplex SM 资源库提交批处理更新。

下面是访问 BATCHREP 的候选方法：

- 新 BATCHREP 资源表，供 CPSM API 作为对象引用。
- 新 BATCHREP 资源表的 WUI 支持
- 提供 BATCHREP 工具的批处理实用程序

WUI 支持和批处理实用程序开发使用新 BATCHREP 资源表作为 CPSM API 引用的对象。

增强 CICSplex SM 批处理资源库更新工具的益处

为使用 CICSplex SM TSO 终端用户接口引入这些可选方法，这样提供：

- 通过 WUI 中新的 BATCHREP 视图改进 CPSM BATCHREP 工具的辅助功能选项
- 将对 CPSM BATCHREP 工具的访问扩展到了新的 z/OS 作业步骤实用程序和 CPSM API 程序
- 提供批处理实用程序，其允许从作业步骤维护定义，从而允许维护集成到库控制系统中

需求

要支持此功能，没有特别的硬件或软件需求。

相关信息

第 311 页的第 27 章，『CICS 操作环境』

批处理实用程序

新的批处理实用程序作为 z/OS 作业步骤运行并连接到资源库需要更新的 CMAS。会提交批处理资源库更新以在此 CMAS 中运行。此实用程序可以启动 EXECUTE 和 CHECK 操作。CMAS 必须为 CICS Transaction Server for z/OS V3.1。

对 CICSplex SM 应用程序编程接口的更改

BATCHREP 资源表现在可以在 CICSplex SM API 中使用。

已为 BATCHREP 资源表添加两个操作：

EXECUTE

提交批处理资源库更新以在指定的 CMAS 中运行。CICSplex SM API 的 CONNECT 命令中的 CONTEXT 选项用于指定 CMAS 的名称。

CHECK

检查批处理资源库更新工具输入文件中指定的命令。

这些操作的输入参数字段与表 13 中显示的外部属性匹配:

表 13. BATCHREP 资源表属性

名称	数据类型	源	长度	摘要	描述
(INPUTDSN-1)	字符型	CPSM	44	DIF	输入数据集名
(INPUTMEMBER-2)	字符型	CPSM	8	DIF	输入成员名
PRINTCLASS	字符型	CPSM	1	DIF	输出打印类
PRINTNODE	字符型	CPSM	8	DIF	目标打印节点
OUTPUTUSER	字符型	CPSM	8	DIF	输出用户标识

对 CICSplex SM Web 用户界面的更改

已将名为“批处理资源库更新作业 (EYUSTARTBATCHREP)”的新 WUI 视图添加到管理视图菜单 (EYUSTARTADMIN)。要从此菜单访问新视图, 单击管理视图标题下的批处理资源库更新作业。新视图等同于 TSO EUI BATCHREP 视图。

要使用 WUI 提交批处理资源库更新, 您必须:

- 确保 WUI 服务器连接到与您要更新的数据资源库关联的 CMAS。
- 选择记录并单击检查或执行按钮
- 输入等同于 TSO EUI BATCHREP 视图所需的那些参数的输入参数。

消息

已添加了许多新的数据资源库消息:

新数据资源库消息

EYUXD0901E	EYUXD0902E	EYUXD0903E	EYUXD0904E
EYUXD0905E	EYUXD0906E	EYUXD0907E	EYUXD0908I
EYUXD0909I	EYUXD0910E	EYUXD0911E	EYUXD0912E
EYUXD0913E	EYUXD0914E	EYUXD0915E	EYUXD0916E

安全性

支持使用外部安全性管理器的现有 CICSplex SM 安全性功能在使用新 z/OS 作业步骤实用程序时、在使用引用 BATCHREP 资源表的 API 程序时和在使用 WUI 来引用 BATCHREP 视图时仍然可以实施访问控制。

迁移

使用新 WUI 或 CPSM API 访问 BATCHREP, 或 BATCHREP 的 z/OS 作业步骤实用程序可以替换现有 TSO EUI BATCHREP 工具。

此实用程序连接所至的 CMAS 必须为 CICSplex SM V3.1

第 5 部分 杂项更改

此部分描述 CICS Transaction Server for z/OS V3R1 中超出此发行版中三个主题作用域之外的新功能和已更改功能。

第 13 章 新安装过程

此主题将您的注意力引向 CICS 的新安装过程。

使用 SMP/E RECEIVE、APPLY 和 ACCEPT 命令安装此发行版的 CICS Transaction Server。SMP/E 对话框可用于实现 SMP/E 安装步骤。

此过程在 *CICS TS 3.1 Program Directory* 中有所描述。它符合 IBM 公司标准，并且已安装其他 z/OS 产品的客户了解它。

安装 CICS Transaction Server 的典型方法 DFHISTAR 仍然可用。 *Program Directory* 表明在《安装指南》中可找到的有关 DFHISTAR 的信息。

新安装过程的优点

安装的 SMP/E RECEIVE、APPLY 和 ACCEPT 过程是 IBM 公司标准。已安装其他 z/OS 产品的客户了解此信息。

第 14 章 EXTRACT STATISTICS 命令

新的 SPI 命令 EXTRACT STATISTICS 处理 URIMAP、PIPELINE 和 WEBSERVICE 资源的统计信息。

使用 EXTRACT STATISTICS 命令检索单个资源的当前统计信息，或者资源类的全局统计信息。

对于 URIMAP、PIPELINE 和 WEBSERVICE 资源，EXTRACT STATISTICS 命令的功能等价于 COLLECT STATISTICS。要收集其他资源的统计信息，使用现有的 COLLECT STATISTICS 命令。

EXTRACT STATISTICS 的语法不同于 COLLECT STATISTICS 的语法。

EXTRACT STATISTICS 命令的优点

所有 CICS SPI 命令可支持的不同选项的数量受到限制。当随着时间，新资源添加到 CICS 时，COLLECT STATISTICS 命令已达到限制，因此不能在现有命令上适用新的 URIMAP、PIPELINE 和 WEBSERVICE 资源。

EXTRACT STATISTICS 命令使用 RESTYPE 选项，以及 CVDA 指定 CICS 资源。因此，不存在对命令可能支持的资源数的限制，尽管在此发行版中，仅支持三个新资源

对 CICS 外部的更改

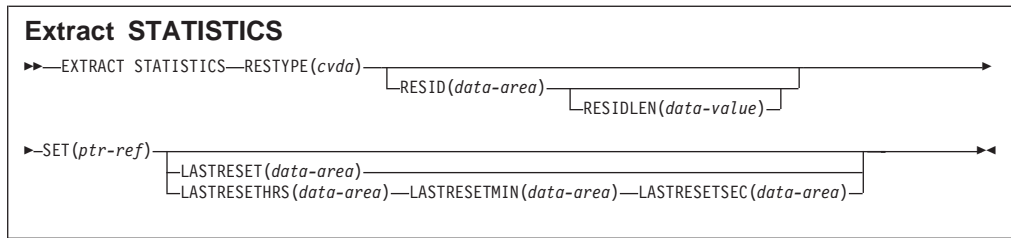
对系统编程接口的更改

EXTRACT STATISTICS 命令

EXTRACT STATISTICS 添加到 CICS，以补充 COLLECT STATISTICS 命令的功能。

对于 PIPELINE、URIMAP 和 WEBSERVICE 资源，使用 EXTRACT STATISTICS 检索单个资源的当前统计信息，或者资源类的全局统计信息。EXTRACT STATISTICS 仅处理这三个资源。要 COLLECT STATISTICS 用于其他资源，继续使用 COLLECT STATISTICS 命令。

EXTRACT STATISTICS 为资源 URIMAP、PIPELINE 和 WEBSERVICE 执行等价于 COLLECT STATISTICS 的功能，这是因为扩展的 COLLECT STATISTICS 未提供这些功能（由于此命令的设计限制）。在与 COLLECT STATISTICS 进行比较时，EXTRACT STATISTICS 的语法是不同的，并为了以后无限的扩充而提供。



条件: INVREQ、IOERR、LENGERR、NOTAUTH、NOTFND

描述

EXTRACT STATISTICS 命令返回到调用应用程序的特殊资源当前统计信息，或给定类型资源的所有统计信息。

CICS 给予您的统计信息是最后一次统计信息抽取时间间隔到期、每天到期或请求复位后收集的那些统计信息。（已写入 SMF 数据集的统计信息不能访问。）EXTRACT STATISTICS 命令不会导致统计信息计数器的复位。

CICS 获取足够的存储用于从此命令返回的数据，并返回到此区域的指针。区域的前两个字节包含其长度。此存储可由后继 EXTRACT STATISTICS 命令重用，所以您应该将任何必需数据存在除了下一次发出此命令的位置之外的其他位置。CICS 在任务终端时释放此存储。

不是所有资源类型都同时提供全局和特定统计信息的。表 14 告诉您哪些统计信息可用于每个资源类型，并给出每个可用统计信息集合的副本名称。副本定义已返回统计信息的格式。当全局统计信息列中未给定副本名称时，全局统计信息不能用于此资源类型；同样，当特定统计信息列中不存在条目时，您无法获得个别资源的统计信息。

表 14. 资源类型和统计信息

资源类型	CVDA	RESIDLEN	统计信息类型	全局统计信息	特定统计信息
PIPELINE	1124	Char(8)	PIPELINE	--	DFHPIRDS
URIMAP	1173	Char(8)	URIMAP	DFHWBGDS	DFHWBRDS
WEBSERVICE	1174	Char(32)	WEBSERVICE	--	DFHPIWDS

副本以 ASSEMBLER、COBOL 和 PL/I 语言提供。（不存在 C 语言的副本。）副本名称在每种语言里都是相同的。您可在下列库中找到它们：

ASSEMBLER	CICSTS31.CICS.SDFHMAC
COBOL	CICSTS31.CICS.SDFHCOB
PL/I	CICSTS31.CICS.SDFHPL1

注：一些副本包含压缩字段。在使用这些字段前，应该检查它们的十六进制零。出于此目的，这些字段的 COBOL 版本已重新定义为带有后缀 -R 的数字。

选项

LASTRESET(*data-area*)

返回 4 位字节的压缩十进制字段，给出所请求统计信息的计数器最后一次复位的时间。这通常是最后一次时间间隔的到期时间。最后一次复位时间总是以本地时间格式返回。

存在两种格式用于复位时间：

- 组合（压缩十进制格式 Ohhmmss+），您可使用 LASTRESET 选项获得。
- 单独的小时、分钟和秒，您可通过分别指定 LASTRESETHRS、LASTRESETMIN 和 LASTRESETSEC 选项获得。

LASTRESETHRS(*data-area*)

返回全字二进制字段，给出所请求统计信息的计数器最后一次复位时间的小时组件（请参阅 LASTRESET 选项）。

LASTRESETMIN(*data-area*)

返回全字二进制字段，给出所请求统计信息的计数器最后一次复位时间的分钟组件（请参阅 LASTRESET 选项）。

LASTRESETSEC(*data-area*)

返回全字二进制字段，给出所请求统计信息的计数器最后一次复位时间的秒组件（请参阅 LASTRESET 选项）。

RESID(*data-area*)

指定抽取哪些统计信息的资源名称。缺少此关键字意味着要抽取全局统计信息。

RESIDLEN(*data-value*)

指定 RESID 数据区的长度。如果省略，缺省值为第 274 页的表 14 中给定的长度。

RESTYPE(*cvda*)

根据提供的 *cvda* 请求特定资源类型的统计信息。有效 CVDA 值为：

PIPELINE

请求 PIPELINE 的统计信息；RESID 标识特殊 PIPELINE。

URIMAP

请求 URIMAP 的统计信息；RESID 标识特殊 URIMAP。

WEBSERVICE

请求 WEBSERVICE 的统计信息；RESID 标识特殊 WEBSERVICE。

SET(*ptr-ref*)

指定要设置为包含已返回统计信息的数据区地址的指针引用。数据区的前 2 个字节包含半字二进制数格式的数据区长度。

第 15 章 支持大小写混合密码

当与 CICS 协同使用的安全管理器支持使用大小写混合密码时，CICS Transaction Server for z/OS V3R1 在将它传递给安全管理器前，不会将密码转换为大写。

存在多个位置您可在 CICS 中输入密码：

资源定义

下列资源定义具有 PASSWORD 属性：

FILE
TCPIPSERVICE
TERMINAL

API 命令

下列命令具有 PASSWORD 选项：

CHANGE PASSWORD
VERIFY PASSWORD
SIGNON

登录事务 (CESN)

事务提供两个字段，可输入密码：

密码
新密码

在所有这些情况下，CICS 处理这些密码的方法取决于同 CICS 协同使用的外部安全管理器是否支持大小写混合密码：

- 如果安全管理器支持大小写混合密码，则 CICS 在不更改您所指定密码的情况下，将它传递给安全管理器。
- 如果不支持，则 CICS 在将密码传递给安全管理器前，将它转换为大写。

第 16 章 代码页转换更改

增强 CICS 以提供 EBCDIC 或 ASCII 和 Unicode 数据进行转换的方法。任一方向的转换都可发生。

- 这些转换使用 z/OS Unicode 转换服务。通常，这样的转换作为 HTTP 请求处理或 CICS TS 3.1 中所引入新 CONTAINER API 命令处理的一部分发生。
- CICS 已支持任意范围 EBCDIC 和 ASCII 代码页组合之间的数据转换，如 *CICS Family: Communicating from CICS on System/390* 中所列出。
- 此支持扩展为允许它们和 UTF-8 或 UTF-16 之间的数据转换。还支持组成 Unicode 的 UTF-8 和 UTF-16 之间的转换。

要使用此处描述的数据转换方法，与 CICS 早期发行版提供的方法对比，您必须使用通道或容器与数据进行通信，如 *CICS Application Programming Guide* 中所描述。

CICS 文档和文档模板与 UTF-8 和 UTF-18 代码页之间不能进行相互转换。无论它们是用作 CICS Web 支持中的静态响应、是在 EXEC CICS WEB API 命令的响应中由 CICS 检索，还是由应用程序使用 EXEC CICS DOCUMENT RETRIEVE 命令检索，此限制都适用。

z/OS Support for Unicode: Using Conversion Services 手册 -SA22 -7649 的 Appendix F 记录了通过这些服务支持的那些转换。CICS 现在支持使用 z/OS 转换服务进行的任意这些字符转换。

代码页转换更改的优点

CICS 现在能够：

- 接受 UTF-8 和 UTF-16 数据作为输入
- 将接收到的数据转换为另一种编码格式，以及
- 随后将结果转换回返回的相应 UTF 格式。

相当扩展可用于转换的代码页对范围。

转换的数据需要主要来自 HTML、XHTML 和 XML，但是新功能不局限于这些，来自将其自身标识为 UTF 而遵守转换的任何接口的输入数据都可使用适当的转换进行处理。

术语

下列术语已添加到 CICS 词汇中。

CCSID

请参阅“编码字符集标识”。

编码字符集标识

表明编码方案标识、字符集标识、代码页标识，以及与编码相关的附加必要信息的 16 位数字，其唯一标识使用的编码图形字符说明。首字母缩写：CCSID。

UNICODE

支持交换处理的通用字符编码标准，以及使用现代社会的任何语言写的文本显示。它还支持很多语言的很多经典文本和有历史意义的文本。Unicode 标准具有 ISO 10646 定义的 16 位国际字符。

需求

对代码页转换的这些增强的硬件和软件需求一般与 CICS TS 相同。

CICS 通过使用 z/OS 提供的 z/OS Unicode 转换服务来实现这些增强。

对 CICS 外部的更改

对安装的更改

没有更改需要作为 CICS 安装过程的一部分。然而，如果您具有取决于对 UTF-8 或 UTF-16 数据转换支持的需求，您必须启用 z/OS 转换服务，并安装指定您要 CICS 执行的转换的转换映像。

请参阅 *z/OS Support for Unicode: Using Conversion Services* 手册 SA22-7649 中的指示信息，以找到通过操作系统服务设置和配置所支持转换需要的步骤。

如果未启用 z/OS 转换服务，CICS 发出消息表明此问题。如果您无须这些服务，可抑制此消息的产生。

- 如果在启动需要使用这些服务的 CICS 区域时遇到消息，则 IPL 需要启用 z/OS 转换服务。

要了解 IPL 后 z/OS 转换服务的状态，从 MVS 控制台使用这些命令中的某一个命令：

/D UNI 显示是否启用 z/OS 转换服务。

/D UNI,ALL

显示是否启用 z/OS 转换服务，以及系统支持哪些转换。

要获得此内容的详细信息，请参阅 *z/OS Support for Unicode: Using Conversion Services* 手册 SA22-7649

对系统初始化参数的更改

存在对数据转换进行增强的新系统初始化。新参数为：

LOCALCCSID={037|CCSID}

指定本地区域的缺省 CCSID。

CCSID 是最多 8 位字符的值。如果未指定 CCSID 值，则缺省 LOCALCCSID 设置为 037。要获得有效 CCSID 列表，请参阅

- *CICS Family: Communicating from CICS on System/390* 中的“CICS-supported conversions”，以及
- *z/OS Support for Unicode: Using Conversion Services* 手册 -SA22 -7649 中的 Appendix F。

对应用程序编程的更改

下列信息添加到 *CICS Application Programming Guide* 中。

使用 z/OS Unicode 转换服务在 CICS 内的数据转换添加到为任何范围 EBCDIC 和 ASCII 代码页组合（如 *CICS Family: Communicating from CICS on System/390* 中所列出）提供的数据转换支持。

通常，涉及 Unicode 的转换作为 HTTP 请求处理或 CICS TS 3.1 中所引入 CONTAINER API 命令使用的一部分发生。将数据转换为 UTF-8 或 UTF-16 和 EBCDIC 和 ASCII 代码页，或者从 UTF-8 或 UTF-16 和 EBCDIC 和 ASCII 代码页转换数据，这取决于适当转换映像的选择。还支持组成 Unicode 的 UTF-8 和 UTF-16 之间的转换。

要使用此处描述的数据转换方法，与 CICS 早期发行版提供的方法对比，您必须使用通道或容器与数据进行通信，如 *CICS Application Programming Guide* 中所描述。

CICS 文档和文档模板与 UTF-8 和 UTF-18 代码页之间不能进行相互转换。无论它们是用作 CICS Web 支持中的静态响应、是在 EXEC CICS WEB API 命令的响应中由 CICS 检索，还是由应用程序使用 EXEC CICS DOCUMENT RETRIEVE 命令检索，此限制都适用。

z/OS Support for Unicode: Using Conversion Services 手册 -SA22 -7649 的 Appendix F 记录了通过这些服务支持的那些转换。CICS 现在支持使用 z/OS 转换服务进行的任意这些字符转换。

确保所需的转换映像是可用的

作为 CICS 应用程序一部分的那些 CCSID 必须是负责维护 z/OS 转换映像的系统程序员所已知的，以便特定转换可用于执行这些应用程序的 CICS 区域。

处理 CCSID 1200

CICS 支持使用下列 CCSID 涉及 UTF-16 数据的转换：1200、1201 和 1202。z/OS 转换服务允许以其大尾数法格式使用 CCSID 1200，但是不包含对小尾数法，或者 CCSID 1201 或 1202 的支持。CICS 在将数据传递给 z/OS 进行转换前，它将这些不受支持的任何格式标识的任何源数据转换为 1200 的大尾数法格式。如果目标数据是不受支持格式的数据，则 CICS 将数据作为 1200 的大尾数法格式接收，并将它转换为所需的 CCSID。如果目标 CCSID 是 1200，则 CICS 假设要以大尾数法格式进行编码。如果在任何这些 CCSID 之间进行转换，CICS 将在不调用 z/OS 转换服务的情况下，执行转换。

当为涉及任何这些 UTF-16 格式的转换设置 z/OS 转换映像时，必须指定 CCSID 1200。当尝试创建转换映像时，z/OS 将无法识别 CCSID 1201 和 1202。

CICS 注意入站转换的字节顺序标记，但是在处理相关出站转换时，不能保留此信息。CCSID 1200 的出站数据为 UTF16-LE。应用程序员需要了解有关此操作的信息，并且在需要时，执行他们自己的 LE 到 BE 转换。

共享转换映像

- 除非应用 PTF for APAR OA05744，否则不要为安装到 CICS 有意使用的 z/OS 映像中的那些转换指定搜索顺序。

- 如果 COBOL 需要相同的转换，则您必须使用两个单独的语句定义转换映像：
 - 一个不使用搜索顺序，而
 - 另一个明确指定搜索顺序为“RECLM”。

例如：

```
CONVERSION 850,037;  
CONVERSION 850,037,RECLM;
```

安装 APAR 后，CICS 和 COBOL 可使用那些支持的转换，其隐含或明确指定缺省搜索顺序，除去在映像生成文件中提供两个控制语句的需要。

JAVA 程序

代码页转换工具存在于 JAVA 中，所以无须在 CICS 中重复这些工具。此处描述的转换工具不扩展到 Java 程序。要获得示例，请参阅 *Java applications in CICS*。

对 CICS 实用程序的更改

转储格式化程序

在 AP 域中存在一个新组件，其显示新的控制块。组件标识为 CV。

对问题确定的更改

消息

由于调用 z/OS 转换服务时发生故障导致发出两条新消息。

- 在 CICS 安装期间发出控制台消息，表明此 CICS 区域不支持 Unicode 转换，这是因为这些服务未启用。
- 发出消息报告此系统不支持两个特定 CCSID 之间的特殊转换。

异常终止

- 未引入新的异常终止代码。
- 通过 DFHCCNVG 中的恢复例程，灾难响应渗透回新功能的调用者。

第 17 章 缺省代码页的简化定义

此主题描述了如何在系统初始化表中定义缺省客户机或服务器代码页，以减少配置 CICSplex 所需的转换表数。

只要是将数据从一个 CICS 系统传递到另一个，就可能必须将一些或所有数据从 ASCII 转换为 EBCDIC 格式，或者反之亦然。DFHCNV 转换表包含每个需要转换的资源的转换模板，使得数据转换变得很容易。

CICSplex 的某些分布式组件，如 CICS Transaction Gateway for z/OS 和 CICS Transaction Server for Windows，它们不提供对转换表中指定缺省客户机代码页的覆盖。由于转换表没有后缀，因此可能需要两个表，每个表驻留在不同的库中，并且这两个表仅在缺省代码页上有区别。

为了减少所需的转换表数，您现在可指定在系统初始化表中定义缺省客户机或服务器代码页。

对于客户机代码页：

1. 在 DFHCNV TYPE=ENTRY 和 TYPE=SELECT 宏中，为 CLINTCP 参数指定值 SYSDEF。
2. 在系统初始化表中，通过为 CLINTCP 参数指定值来设置缺省客户机代码页。DFHCNV 宏上的 CLINTCP 参数可使用任何支持的值。缺省值为 CLINTCP=437。

对于服务器代码页：

1. 在 DFHCNV TYPE=ENTRY 和 TYPE=SELECT 宏中，为 SRVERCP 参数指定值 SYSDEF。
2. 在系统初始化表中，通过为 SRVERCP 参数指定值来设置缺省服务器代码页。DFHCNV 宏上的 SRVERCP 参数可使用任何支持的值。缺省值为 SRVERCP=037。

在数据转换模板中改进代码页缺省值的优势

在系统初始化表中定义代码页缺省值的功能（而不是在 DFHCNV 宏中直接定义）可通过减少必须维护的 DFHCNV 转换表数来简化 CICSplex 的定义和管理。

需求

没有特别的硬件或软件需求。

对 CICS 外部的更改

对系统初始化参数的更改

为了在系统初始化表中定义缺省客户机和服务器代码页，引入了两个新的系统初始化参数：CLINTCP 和 SRVERCP，与相同名称的现有 DFHCNV 宏参数相匹配。

新的系统初始化参数

CLINTCP={437|codepage}

指定 DFHCNV 数据转换表要使用的缺省客户机代码页，但是仅当 DFHCNV 宏中的 CLINTCP 参数设置为 SYSDEF 时才使用。*codepage* 是最多 8 位字符的字段，并且可采用 DFHCNV 宏中的 CLINTCP 参数支持的值。

SRVERCP={037|codepage}

指定 DFHCNV 数据转换表要使用的缺省服务器代码页，但是仅当 DFHCNV 宏中的 SRVERCP 参数设置为 SYSDEF 时才使用。*codepage* 是最多 8 位字符的字段，并且可采用 DFHCNV 宏中的 SRVERCP 参数支持的值。

对用户可替代程序的更改

新的 DFHCNV 宏参数操作数

新的操作数 SYSDEF 已添加到 TYPE=INITIAL 和 TYPE=ENTRY 宏参数 CLINTCP 和 SRVERCP。这些宏定义用户可替代的数据转换表 DFHCNV。

- DFHCNV TYPE=INITIAL 宏定义转换表的开始部分。它给出有效的代码页列表。
- DFHCNV TYPE=ENTRY 宏指定唯一标识数据资源的名称和类型。对于需要转换的每个资源都必须有一个。

已更改参数的格式现在按如下所示：

CLINTCP={437|SYSDEF|nnnn [, nnnn, ...]}

第一个操作数定义当从 DFHCNV TYPE=ENTRY 宏省略 CLINTCP 和 CDEPAGE 操作数时要使用的缺省客户机代码页。

SYSDEF 指定系统初始化表参数 CLINTCP 确定的缺省客户机代码页。

您可进一步指定代码页；验证它们，但是不使用。

SRVERCP={037|SYSDEF|nnnn [, nnnn, ...]}

第一个操作数定义当从 DFHCNV TYPE=ENTRY 宏省略 SRVERCP 和 CDEPAGE 操作数时要使用的缺省客户机代码页。

SYSDEF 指定系统初始化表参数 SRVERCP 确定的缺省客户机代码页。

您可进一步指定代码页；验证它们，但是不使用。

第 18 章 64 位寻址能力更改

CICS 现在可在使用 64 位寻址体系结构的任务异常终止时提供有意义的信息。

CICS 不支持 64 位寻址执行，但是程序可使用仅当 CICS 在 64 位体系结构机器上运行时可用的地址上的存储。这些更改提供对 CICS 异常终止捕获机制的扩展，以致捕获到完整 64 位通用寄存器的内容。

64 位寻址能力更改的优点

开发 64 位寻址代码的客户现在可在任务异常终止时访问更多信息。

需求

这些增强的硬件和软件需求与此发行版 CICS TS 的剩余增强的那些硬件和软件需求相同。

对 CICS 外部的更改

对 CICS 实用程序的更改

转储格式化程序

CICS 转储格式化程序显示在发生异常终止时捕获的 64 位通用寄存器的内容。

对问题确定的更改

完整 64 位通用寄存器文件的内容被捕获，并可由您使用。

不存在新的消息、异常终止代码或跟踪点。

第 19 章 支持撤销的用户标识

当发出命令 EXEC CICS VERIFY PASSWORD 时，CICS 现在执行用户标识或用户组
连接的撤销状态。例如，如果用户过多尝试登录，则标识被撤销，而且用户无法访问
系统或资源。

第 6 部分 废弃的功能

CICS Transaction Server for z/OS V3R1 的作用域中废弃或减少了一些原先在 CICS Transaction Server for z/OS V2 中支持的功能。

第 20 章 除去对 OS/VS COBOL 程序的运行时支持

除去了对 OS/VS COBOL 程序的运行时支持。

OS/VS COBOL 程序，在 CICS Transaction Server for z/OS V2 中有它的运行时支持，但是它无法在 CICS TS for z/OS V3 下运行。

OS/VS COBOL 程序必须升级到符合 COBOL 的 Language Environment，并且用 CICS 支持的 COBOL 编译器级别重新编译。推荐使用 Enterprise COBOL for z/OS and OS/390[®] V3 编译器。

第 319 页的第 29 章，『高级语言支持』和 *CICS Application Programming Guide for CICS TS for z/OS V3* 提供关于 COBOL 和其他语言所支持编译器的信息。*CICS Application Programming Guide* 的 Appendix B 提供将 OS/VS COBOL 程序转换到符合 COBOL 的 Language Environment 的帮助信息。

新的异常终止代码 ALIK 表示尝试使用 OS/VS COBOL 程序。在这种情况下，CICS 将异常终止任务并且禁用程序，而 CICS 处理继续执行。

第 21 章 对 BTAM 和 TCAM 支持的更改

CICS对基本远程通信访问方法 (BTAM) 的支持在CICS Transaction Server for z/OS V3R1 中停止。对远程通信访问方法 (TCAM) 的支持限制为对 DCB 接口的间接支持。

收回 BTAM 支持

CICS Transaction Server for z/OS V3R1 不支持基本远程通信访问方法 (BTAM)。

一些 CICS 发行版仅间接地支持 BTAM 终端，即，通过从后备级别终端拥有区域 (TOR) 到 BTAM 终端所连接的区域的事务路由来支持。在 CICS TS for z/OS V3.1 中，已除去此间接支持。不再支持 BTAM 并且已除去对它的所有引用。

这意味着，如果您有一个连接到后备级别 CICS TOR 的 BTAM 终端网络，则您将无法 (因为您的 CICS 是先前发行版) 将事务从它们路由到 CICS TS for z/OS V3.1 应用程序所属区域 (AOR)。您必须升级您的终端或路由到 CICS 的先前版本。

TCAM 的 CICS 支持中的更改

CICS Transaction Server for z/OS V3R1 不支持 TCAM/ACB 接口。它间接地支持 TCAM/DCB 接口。

一些 CICS 发行版仅间接地支持 TCAM 的 ACB 接口，即，通过终端所属的后备级别终端拥有区域 (TOR) 的事务路由来支持 TCAM 的 ACB 接口。在 CICS TS for z/OS V3.1 中，已除去此间接支持。

在先前的 CICS 发行版中，是完全支持 TCAM 的 DCB 接口的。即：

1. 可以使用 TCAM/DCB 将终端连接到当前级别的 CICS TOR。
2. 可以将由 TCAM/DCB 连接的终端启动的事务路由到当前级别的 CICS AOR。

在 CICS TS for z/OS V3.1 中，仅支持第二个操作。

这些更改的结果：

- 如果您有一个通过 TCAM 的 ACB 接口连接到后备级别 CICS TOR 的终端网络，您将无法 (因为您使用的是先前的 CICS 发行版) 将事务从它们路由到 CICS TS for z/OS V3.1 AOR。您必须迁移您的连接以使用 TCAM/DCB 或 (最好是) ACF/VTAM，或路由到先前版本的 CICS。(支持 TCAM/ACB 的所有终端也支持 ACF/VTAM。)
- 如果您有一个通过 TCAM 的 DCB 接口连接到 CICS TS 2.3 TOR (例如) 的终端网络，则您将无法把 TOR 迁移到 CICS TS for z/OS V3.1。要这样做，您必须迁移您的连接以使用 ACF/VTAM。
- 如果您有一个通过 TCAM 的 DCB 接口连接到后备级别 CICS TOR 的终端网络，则您将 (因为是先前的 CICS 发行版) 能把事务从它们路由到 CICS TS for z/OS V3.1 AOR。但建议迁移您的连接以使用 ACF/VTAM。

对 CICS 外部的更改

对系统初始化参数的更改

DFHTCAM={NO | YES} 系统初始化参数现在已废弃并且只是为了与先前的 CICS 发行版兼容而保留。如果指定它，则会显示一条拒绝消息并采用 TCAM=NO。

对资源定义的更改

CICS 不再支持 BTAM 终端，也不间接地支持 BTAM 终端。因此您可以不再定义 BTAM 终端，也不定义为远程资源。

CICS 不再支持本地 TCAM 终端。因此，可以不再使用以下资源定义宏定义本地 TCAM 终端：

- DFHTCT TYPE=SCSDI
- DFHTCT TYPE=LINE
- DFHTCT TYPE=TERMINAL

仍然可以定义远程 TCAM 终端。您可使用以下任一方法执行此操作：

- 单个 DFHTCT TYPE=REMOTE 宏。
- DFHTCT TYPE=REGION 宏，后跟 DFHTCT TYPE=LINE 和 DFHTCT TYPE=TERMINAL 宏。CICS 只使用 DFHTCT TYPE=LINE 和 DFHTCT TYPE=TERMINAL 宏的远程属性。

对应用程序编程接口的更改

以下与 BTAM 相关的 EXEC CICS API 命令已废弃：

- CONVERSE (SYSTEM/3)
- CONVERSE (SYSTEM/7)
- CONVERSE (2741)
- CONVERSE (2770)
- CONVERSE (2780)
- CONVERSE (3600 BTAM)
- CONVERSE (3735)
- CONVERSE (3740)
- ISSUE COPY (3270 display)
- 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)

对全局用户出口的更改

不再调用终端控制程序中的以下全局用户出口（在 TCAM 输入和输出事件中已调用）：

- XTCTIN
- XTCTOUT

对用户可替代程序的更改

因为不再支持本地 TCAM 终端，所以不为 TCAM 终端调用终端错误程序。但仍然为连续设备调用它。

对样本程序的更改

DFHSPTM1 和 DFHSPTM2 样本 TCAM 程序不再随 CICS 一起提供。

迁移

如果您有一个连接到后备级别 CICS 终端拥有区域（TOR）的 BTAM 终端网络，则您将无法（因为您的 CICS 是先前发行版）将事务从它们路由到 CICS TS for z/OS V3.1 应用程序所属区域（AOR）。您必须升级您的终端或路由到 CICS 的先前版本。

如果您有一个通过 TCAM 的 ACB 接口连接到后备级别 CICS TOR 的终端网络，您将无法（因为您使用的是先前的 CICS 发行版）将事务从它们路由到 CICS TS for z/OS V3.1 AOR。您必须迁移您的连接以使用 TCAM/DCB 或（最好是）ACF/VTAM，或路由到先前版本的 CICS。（支持 TCAM/ACB 的所有终端也支持 ACF/VTAM。）

如果您有一个通过 TCAM 的 DCB 接口连接到 CICS TS 2.3 TOR（例如）的终端网络，则您将无法把 TOR 迁移到 CICS TS for z/OS V3.1。要这样做，您必须迁移您的连接以使用 ACF/VTAM。

如果您有一个通过 TCAM 的 DCB 接口连接到后备级别 CICS TOR 的终端网络，则您将（因为是先前的 CICS 发行版）能把事务从它们路由到 CICS TS for z/OS V3.1 AOR。但建议迁移您的连接以使用 ACF/VTAM。

共存

CICS TS for z/OS V3.1 不支持连接到前 CICS TS 3.1 终端拥有区域的 BTAM 终端的事务路由和功能发送。

CICS TS for z/OS V3.1 不支持通过 TCAM/ACB 连接到前 CICS TS 3.1 终端拥有区域的终端的事务路由或功能发送。

CICS TS for z/OS V3.1 确实支持通过 TCAM/DCB 连接到前 CICS TS 3.1 终端拥有区域的终端的事务路由和功能发送。

第 22 章 除去对 1 个字节的控制台标识的支持

除去了对使用 1 个字节的控制台标识定义终端的支持。 废弃 **TERMINAL** 资源定义上的 **CONSOLE** 属性，但是仍支持它以使得能够与 CICS 的较早发行版兼容。

您可以在 **TERMINAL** 资源定义上使用 **CONSNAME(name)** 属性定义终端。

CICSplex SM 支持

1 位字节的控制台标识不再使用，而且已除去对它的支持。

对 CICSplex SM 应用程序编程接口的更改

TERMDEF 资源表

TERMDEF 资源表中的 **CONSOLE** 属性在 CICS Transaction Server for z/OS, V3R1 或更高发行版中不再有效。

对 CICSplex SM 最终用户界面视图的更改

已进行下列更改：

TERMDEF 视图

尽管下列属性在 CICS Transaction Server for z/OS V3R1 或更高发行版中不再有效，但是 **TERMDEF** 视图保持不变。

CONSOLE

控制台标识

对 CICSplex SM Web 用户界面的更改

已进行下列更改：

“终端定义”视图

尽管下列属性在 CICS Transaction Server for z/OS V3R1 或更高发行版中不再有效，但是 **TERMDEF (EYUSTARTTERMDEF)** 视图保持不变。

CONSOLE

控制台标识

消息

#

BBMZA094E CAS (SSID) INVALID REPLY

#

除去 1 位字节的控制台标识意味着 CICSplex SM 消息 **BBMZA094E** 现在转至所有活动控制台，而不是单单转至应答消息 **BBMZA094E** 的控制台。消息文本未更改。

#

第 23 章 撤消 CICS TS 的 CICS 连接器

撤消了对 CICS TS for z/OS V2.1 中引进的 CICS TS 的 CICS 连接器的支持。

CICS 连接器是一个软件组件，它允许 Java 客户机应用程序调用 CICS 应用程序。CICS TS for z/OS V2.3 引进了新的 CICS 连接器（CICS TS 的 CCI 连接器），它扮演的角色与 CICS TS 的 CICS 连接器类似，即，它允许 Java 程序或企业 bean 在 CICS Transaction Server for z/OS 上运行，以链接到 CICS 服务器程序。然而，尽管旧的 CICS TS 的 CICS 连接器实施 IBM 所有的公共连接器框架（CCF）接口，但新的 CICS TS 的 CCI 连接器实施 J2EE 连接器体系结构规范 V1.0 定义的工业标准公共客户机接口（CCI）。

自 CICS TS for z/OS V2.3 开始，给出了以下一些建议：

- 编写新的连接器应用程序时，您使用 CICS TS 的 CCI 连接器而不是 CICS TS 的 CICS 连接器
- 您把使用 CICS TS 的 CICS 连接器的任何现有应用程序迁移成改为使用 CICS TS 的 CCI 连接器

由于 CICS TS for z/OS V3.1 中撤消了对 CICS TS 的 CICS 连接器的运行时支持，因此这些建议现在变为强制的了。

注：在前发行版中，它可能以下列两种方法的任何一个对 CICS TS 的 CICS 连接器进行编程：使用高级 CCF API，或者 CICS Transaction Gateway API 的低级、特定于 CICS 的外部调用接口（ECI）。ECI 基类不再与 CICS 一起提供。

要获得在新的应用程序中使用 CICS TS 的 CCI 连接器，以及把使用 CICS TS 的 CICS 连接器的现有应用程序迁移成改为使用 CICS TS 的 CCI 连接器的建议，请参阅 *CICS 中的 Java 应用程序*。

第 24 章 除去对 Java 程序对象和热缓冲池的运行时支持

除去对 Java 程序对象和热缓冲池（HPJ）的运行时支持。

在 CICS TS 1.3 中，作为在 Java 虚拟机（JVM）中运行 Java 程序的替代方法，可以使用 VisualAge[®] for Java, Enterprise Edition for OS/390（ET/390）将 Java 字节码绑定到 OS/390 可执行文件中，它称为 Java 程序对象。Java 程序对象存储在 OS/390 PDSE 库中，并且由 Language Environment 运行单元或环境中的 CICS 执行。ET/390 的 Java 运行时组件在 CICS 区域中提供此运行时支持。

可以为每个调用构建和初始化 Language Environment 环境，在这种情况下，Java 程序对象在 QR TCB 下执行。或者，要减少性能开销，可以为程序的多个调用重用预初始化的和持久的环境。此功能部件称为热缓冲池。为 Java 程序对象指定热缓冲池时，CICS 使用 z/OS Language Environment 的 Hot-Pooling 时，CICS 使用 z/OS Language Environment 的 PIPI 预初始化服务构建环境，并且在 H8 方式的开放式事务环境（OTE）任务控制块（TCB）的控制下在 CICS 区域中执行 Java 程序对象。

在 CICS TS for z/OS V2.3 中，向现有 hpj 编译的 Java 程序对象提供运行时支持，但不向开发新的 Java 程序对象提供支持，也不向修改现有 Java 程序对象提供支持。在 CICS TS for z/OS V3.1 中，除去了对 hpj 编译的 Java 程序对象的运行时支持。

您必须迁移任何 hpj 编译的 Java 程序对象，以在 Java 虚拟机（JVM）中运行。CICS *Migration Guide* 说明如何实现它。要获得有关 CICS JVM 和 CICS 的 Java 编程的信息，请参阅 CICS 中的 *Java 应用程序*。

如果您试图在 CICS TS for z/OS V3.1 中执行 Java 程序对象，则发出一个 ALIG 异常终止。

开放的 TCP 方式 H8（其用于热缓冲池 Java 程序对象）不再存在。

对 CICS 外部的更改

对系统初始化参数的更改

除去系统初始化参数 MAXHPTCBS。MAXHPTCBS 控制开放的 TCP 方式 H8。

对资源定义的更改

- 从 PROGRAM 资源定义中除去了 HOTPOOL 属性。此属性用于指定 Java 程序对象是否要在 H8 TCB 控制下，在由多个程序调用重用的预安装 Language Environment 环境中运行。
- 除去了样本应用程序组 DFH\$JAVA。此组包含使用 VisualAge for Java, Enterprise Edition for OS/390 的 Java 支持的样本应用程序需要的资源定义。定义相同的样本应用程序以供 DFH\$JVM 组的 JVM 使用。
- 从 DFHMCT TYPE=RECORD 宏中除去了 DFHTASK 字段 278，即 CICS MAXHPTCBS 延迟时间。

对应用程序编程接口的更改

除去用于限定对 EXEC CICS LINK 和 XCTL 命令，以及对 BTS 命令 LINK ACQPROCESS 和 LINK ACTIVITY 的 INVREQ 响应的 RESP2 值 43。

43 链接堆栈上已经有一个热共享程序时，热共享 Java 程序对象已尝试了 LINK（或 XCTL）。

对系统编程接口的更改

- 除去了 EXEC CICS CREATE PROGRAM 命令中的 HOTPOOL 选项。
- 除去了 EXEC CICS INQUIRE PROGRAM 命令中的 HOTPOOLING 选项。
- 除去了 EXEC CICS SET PROGRAM 命令中的 HOTPOOLING 选项。
- 从 EXEC CICS INQUIRE DISPATCHER 命令中除去了 ACTHPTCBS 和 MAXHPTCBS 选项。使用这些选项查询当前分配给用户任务的 H8 方式 TCB 开放数目，以及允许 CICS 连接和维护的最多 H8 方式 TCB 开放数目。
- 除去了 EXEC CICS SET DISPATCHER 命令中的 MAXHPTCBS 选项。
- 删除了 CVDA HOTPOOL (1065) 和 NOTHOTPOOL (1066)。

对 CEMT 的更改

- 从 CEMT INQUIRE PROGRAM 显示中除去了 HOTPOOLING 字段。
- 从 CEMT SET PROGRAM 命令中除去了 HOTPOOL 和 NOTHOTPOOL 选项。
- 从 CEMT INQUIRE DISPATCHER 显示中除去了 ACTHPTCBS 和 MAXHPTCBS 字段。
- 从 CEMT SET DISPATCHER 命令中除去了 MAXHPTCBS 选项。

对全局用户出口的更改

全局用户出口任务指示器字段（它由 UEPGIND 寻址），而 UEPGIND 是 DFHUEPAR 标准参数列表的一部分，不再包括符号值 UEPH8。UEPH8 表示开放的 TCP 方式 H8。

对出口编程接口（XPI）的更改

从 DFHPGISX 调用 INQUIRE PROGRAM 和 SET PROGRAM 除去 HOTPOOL 选项。

对用户可替代程序的更改

除去用户可替代程序 DFHAPH8O 和 DFHJHPAT。

- 提供 DFHAPH8O 以允许您更改要运行 Java 程序对象的 Language Environment 环境的缺省 Language Environment 运行时选项。
- DFHJHPAT 是可选的，并且可以将它用于您自己的使用，例如跟踪。它在调用 Java 程序对象之前调用。

对监控的更改

除去组 DFHTASK 中的监控数据字段 278。开放的 TCB 方式 H8（其用于 Hot-Pooling Java 程序对象并且由 MAXHPTCBS 控制）不再存在。

对统计信息的更改

TCB 方式统计信息中不再显示 TCB 方式 H8。TCB 方式统计信息在 DFHDSGDS DSECT 中由 DSGTCBM DSECT 映射。

对问题确定的更改

- 除去了 Java hot-pooling 的跟踪条目（AP 19A0 到 AP 19C4）。
- 除去了消息 DFHAP1219 到 DFHAP1225。
- 除去了异常终止 AJH0 到 AJHF。
- 资源类型 HP_POOL 不再会引起分派器等待了。

如果您尝试执行 CICS TS for z/OS V3.1 中的 Java 程序对象，则发出 ALIG 异常终止。

CICSplexSM 支持

除去对 Java 程序对象和对热缓冲池（HPJ）的运行时支持导致对 CICSplex SM 接口的很多更改。

对 CICSplex SM 最终用户界面视图的更改

CICS Transaction Server for z/OS, V3R1 中没有新的最终用户界面视图。但是，对下列现有的视图进行了一些更改：

- PROGDEF - 属性 HOTPOOLING 不再有效，但仍然显示。
- PROGRAM - 除去了属性 HOTPOOLING。

对 CICSplex SM 应用程序编程接口的更改

除去对 Java 程序对象和热缓冲池（HPJ）的运行时支持导致下列更改：

- 『CICSRGN 资源表』
- 『PROGRAM 资源表』
- 第 304 页的 『PROGDEF 资源表』

CICSRGN 资源表

现有 **HOTPOOLING** 统计信息在 CICS TS 3.1 中不再有效。

PROGRAM 资源表

SPI 属性在 CICS TS 3.1 中不再有效。

PROGDEF 资源表

属性 HOTPOOLING 在 CICS TS 3.1 中不再有效。

对 CICSplex SM Web 用户界面的更改

已对下列视图进行更改:

- 『“程序定义”视图』
- 『程序视图』

“程序定义”视图

属性 HOTPOOLING 在 CICS TS for z/OS, V3.1 中不再有效, 但 PROGDEF (EYUSTARTPROGDEF.DETAILED) 视图保留原样。

程序视图

属性 HOTPOOLING 在 CICS TS for z/OS, V3.1 中不再有效, 但 PROGRAM (EYUSTARTPROGRAM.DETAILED) 视图保留原样。

第 25 章 除去了对 Windows 远程 MAS 的 CICSplex SM 支持

在远程受管的应用程序系统（RMAS）的管理中，CICSplex SM 的前发行版已支持 TXSeries® V4.3.0.4 和 TXSeries V5.0（也称为 NT 4.3 和 NT 5.0）的 CICS for Windows 组件。此支持不再是必需的，并且已从 CICS Transaction Server for z/OS V3.1 以及之后的发行版中除去了 CICSplex SM TXSeries 代理程序。因此，不再可能为 Windows 设置 CICSplex SM 远程 MAS 代理程序。

希望设置它的客户可以继续使用 CICSplex SM 中的 CICS Transaction Servers 2.3 or 2.2 for TXSeries 支持。

第 26 章 除去 CICS 事务亲缘关系实用程序

CICS TS for z/OS V3.1 不提供 CICS 事务亲缘关系实用程序。CICS Interdependency Analyzer 现在提供检测和报告事务相似性的功能，这是更成熟的工具。

使用 CICS 互依赖性分析器，您可以：

- 识别各个 CICS 事务使用的资源集，以及它们与其他资源的关系。这让您能够理解构成您的应用程序集合的内容：您能够明了 CICS 区域包含些什么；要运行事务需要什么资源；哪些程序使用哪些资源；以及哪些资源不再使用。因此，维护、增强、修改或重新分发应用程序的能力有显著改进。
- 识别可能的事务亲缘关系。亲缘关系需要在同一个 CICS 区域或特定的区域中运行特定的事务组。识别事务亲缘关系的能力在动态路由选择环境中很有用：您需要知道防止特定的事务被路由到特定应用程序所属区域（AOR），或者需要将特定事务路由到特定 AOR 的任何约束。

要获取有关 CICS 互依赖性分析器的更多信息，请参阅 *CICS Interdependency Analyzer User's Guide and Reference*。

第 7 部分 一般信息

第 27 章 CICS操作环境

本主题提供了有关您为了使用 CICS Transaction Server for z/OS 的 CICS 和 CICSplex SM 元素所需的相关产品的一些信息。

硬件需求

处理器

基本需求是用于支持必备操作系统的处理器，并具有足够的处理器存储满足 z/OS V1.4, CICS TS for z/OS V3.1、应用程序、访问方法和所有其他正在运行的软件的需求。这包括 IBM eServer™ zSeries 990。

并行系统复用®支持

CICS 和 MVS 系统记录器日志流合并工具支持的每个数据共享工具需要并行系统复用环境。这需要:

- 安装一个或多个耦合工具，及其关联的耦合链接
- 提供常见外部时间源的 IBM 综合系统 (sysplex) 计时器
- 综合系统 (sysplex) 中支持大量中央处理器机器群 (CPC) 的充足 DASD 路径。DASD 路径由具有足够路径专用于综合系统 (sysplex) 中每个 CPC 的 DASD 控制器提供，或者由 ESCON® 导向器提供。

对数据共享的 CICS 支持可用于访问 IMS 数据库、DB2 数据库、VSAM 数据集、CICS 临时存储、耦合工具数据表和指定计数器中的数据。

加密硬件

需要 zSeries 加密硬件:

- 开发 WS-Security 能力。
- 完全受益于对 SSL 加密的性能改进。

两个功能都依赖于 z/OS 集成加密服务工具 (ICSF)。

片假名终端设备

因为 CICS 必须发出大小写混合的某些消息，所以此产品不支持限制为代码页 930 的非扩展单字节字符集 (SBCS) 片假名部分的显示或终端仿真器。

软件需求

注意, *Program Directory* (GI10-6427) 通常将包含有关软件需求的最新信息。

操作环境

CICS TS for z/OS V3.1 需要 z/OS V1.4 或更高版本。注意, 它将不会在环境中使用已安装的最低操作系统级别初始化。

Language Environment 库 SCEERUN 必须在 CICS 初始化期间，通过包括在 STEPLIB 并置或 LNKLIST 中来可用于 CICS。 Language Environment 服务由大量 CICS 功能使用。

对于 Java 应用程序或 Enterprise bean、IBM SDK for z/OS、Java 2 Technology Edition, 功能持久可再用的 JVM technology, V1.4, 程序 5655-I56。这必须在 V1.4.2 级别上。

- IBM SDK for z/OS, Java 2 Technology Edition, V1.4 可免费在磁带或通过下载获得: <http://www.s390.ibm.com/java/>。
- 注意, IBM 64-Bit SDK For z/OS, Java2 Technology Edition, V1.4, 程序号 5655-M30 不提供必需的功能。

对于 WS-Security 支持, IBM XML Toolkit for z/OS V1.7 是必需的。这是免费提供的产品, 程序号 5655-J51。

对于 Enterprise bean 的部署, WebSphere Application Server V5.0 或更高版本是必需的。

- 要使用的组件是 Application Server Toolkit (ASTK) for Windows。注意, 不支持与早期出版物 V5.0 一起提供的 Application Assembly Tool (AAT)。
- 还注意, ASTK 也包含在 WebSphere Studio Enterprise Developer V5.1 中。

对 Enterprise bean 的 JNDI 支持可由 LDAP 服务器提供, 此服务器在 SecureWay® 安全服务器中提供, 并许可作为基本 z/OS操作系统的一部分。CICS TS V3.1 将与 WebSphere Application Server (任何平台) V5 和 V6 互操作。这直接应用于使用 RMI/IIOP 进行互操作的客户; 并通过 CICS Transaction Gateway V5.0 或更高版本为使用 JCA 的客户完成。

对于开发与 CICS TS V3.1 协同使用的 Java 程序 (包括 Enterprise bean), WebSphere Studio 系列 V5 和 Rational® Application Developer V6 受支持。

z/OS的系统 SSL 安全级别 3 功能必须使用以 128 位或更高位数加密的密码套件。

其他支持的产品

支持其他产品的下列级别与CICS TS for z/OS V3.1协同使用:

- 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)
 - 对于 SQLJ/JDBC 支持, 使用 PTF for APAR PQ84783
 - 不支持 DB2 Group Attach
- DB2通用数据库服务器, OS/390 版, V6.1 (5645-DB2)
 - 对于 SQLJ/JDBC 支持, 使用 PTFs for APARs PQ84783 和 86525
 - 对于 DB2 Group Attach, 使用 APARs PQ44614, PQ45691 和 PQ45692
- DB2通用数据库, z/OS 版 V8.1 (5625-DB2)
 - 对于 SQLJ/JDBC 支持, 使用 PTFs for APARs PQ84783 和 86525
- WebSphere MQ for z/OS V5.3 (5655-F10)
- Tivoli® Decision Support for OS/390 (5698-ID9) V1.6, with necessary service applied

- Tivoli Business Systems Manager V3.1
- CICS Universal Client Version 5.0 或更高版本
- CICS Transaction Gateway Version 5.0 或更高版本

信息中心环境

在以下平台上，支持信息中心作为服务器：

- Windows 2000 和 Windows XP
- AIX® V5.1 或更高版本
- Linux™ on Intel™ (RedHat and Suse)

要浏览信息中心，您将需要支持 HTML 4.0 和文档对象模型（DOM）标准的浏览器。合适的浏览器包括：

Microsoft® Internet Explorer Version 6.0
Netscape Navigator V7.0
Mozilla V1.0

在 Windows 2000 或 Windows XP 上运行。

要阅读与信息中心一起提供的 PDF 文件，您将需要 Adobe Acrobat Reader 5.0 或 6.0。这些文件已使用 Adobe Acrobat Distiller 6.0 在 Acrobat 6.0（PDF 1.5）级别上生成。它们可使用 Adobe Acrobat Reader 5.0 阅读，但是如果需要 Distiller 6.0 的辅助功能选项，则需要 Reader 6.0。

对 CICS 工具和相关产品的支持

CICS Interdependency Analyzer

下列产品可与 CICS TS for z/OS V3.1 协同使用：

CICS Interdependency Analyzer for z/OS V1.3, with PTF for APAR PQ95065。

下列产品不能与 CICS TS for z/OS V3.1 一起运行：

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 不支持 CICS TS for z/OS V3.1 的 SMF 110 数据。

CICS Performance Monitor

CICS Performance Monitor for z/OS V1.1 不支持 CICS TS for z/OS V3.1。

应用了服务的 CICS Performance Monitor for z/OS V1.2 在 CICS TS for z/OS V2.3 级别上，提供对 CICS TS for z/OS V3.1 的允许支持。

Candle® OMEGAMON®

以下支持 CICS TS for z/OS V3.1。然而，此支持不包括对 CICS TS for z/OS V3.1 新功能的开发。

Candle OMEGAMON II[®] for CICS V520。
Candle OMEGAMON XE for CICS V100。
Candle OMEGAMON XE for CICSplex V220, 已应用服务。

CICS VSAM Recovery

下列产品提供对 CICS TS for z/OS V3.1 所处理 VSAM 文件的恢复支持：
CICS VSAM Recovery V3.2。
CICS VSAM Recovery V3.3。

CICS Business Event Publisher

已应用服务的 CICS Business Event Publisher for MQSeries[®] V1.2 可与CICS TS for z/OS V3.1协同使用。

CICS Online Transmission Time Optimizer

CICS Online Transmission Time Optimizer for z/OS V1.1 可与CICS TS for z/OS V3.1协同使用。

会话管理器

下列产品可与CICS TS for z/OS V3.1协同使用：
Session Manager for z/OS V1.1。
Session Manager for z/OS V1.2。

CICS VSAM Transparency

CICS VSAM Transparency for z/OS V1.1 可与CICS TS for z/OS V3.1协同使用。

CICS VSAM Copy

已应用服务的 CICS VSAM Copy for z/OS V1.1 可与CICS TS for z/OS V3.1协同使用。

CICS Batch Application Control

已应用服务的 CICS Batch Application Control for z/OS V1.1 可与CICS TS for z/OS V3.1协同使用。

MQSeries Integrator Agent

MQSeries Integrator Agent for CICS Transaction Server V1.1 不能与CICS TS for z/OS V3.1一起运行。

故障分析器

下列产品可与CICS TS for z/OS V3.1协同使用：
Fault Analyzer for z/OS and OS/390 V3.1, 使用 PTF UQ77156 for APAR PQ74048。
Fault Analyzer for z/OS V4.1。
Fault Analyzer for z/OS V5.1。

调试工具

下列产品可与CICS TS for z/OS V3.1协同使用:

Debug Tool for z/OS and OS/390 V3.1, 使用 PTF UQ77541 for APAR PQ73643。

Debug Tool for z/OS V4.1。

Debug Tool for z/OS V5.1。

兼容性

z/OS转换服务

与先前级别的 CICS Transaction Server 不同, CICS TS V3 可使用 z/OS服务执行超出前发行版中 CICS TS 所支持功能的转换。示例是转换到 Unicode 和从 Unicode 转换, 这可能需要支持 Web Service。这需要 z/OS 安装初始转换映像, 这只能在系统 IPL 上完成。如果要在没有对 z/OS进行重新 IPL 的情况下, 希望安装 CICS TS V3, 则倘若在上一系统 IPL 期间安装初始转换映像, 则可完成此操作。1 转换映像不包括来自 CICS TS 的任何代码; 它还可在不需要进一步 IPL 的情况下刷新。

CICS 中的 JVM 方式

建议 CICS TS V3.1 中使用 Java 程序的客户使用连续方式。CICS TS V2.3 中引入对连续方式的支持; 为了将 Java的 CICS 使用加入带有标准操作的行, 则在 CICS TS 的下一个发行版中将除去对可设置方式的支持。

SOAP for CICS 功能

SOAP for CICS 功能, 可与 CICS TS V2.2 和 V2.3 一起订购, 不能与 CICS TS V3.1 一起订购。然而, 要帮助已具有此功能的客户迁移, CICS TS V3.1 可使用和支持此功能, 而应用程序将继续运行。然而, 建议客户迁移到 CICS TS V3.1 的 Web Service 支持功能。

公共连接器框架 (CCF)

CICS TS V3.1 不支持作为公共客户机接口 (CCI) 的先行作业接口的公共连接器框架 (CCF)。CICS TS V2.3 的声明中表明了要除去此支持的想法。

ECI 基类 (ECIREQUEST)

CICS TS V3.1 中不包括 ECL 基类 (ECIREQUEST, 器为了与 CICS Transaction Gateway 兼容而引入)。建议的替换是 COMMON CLIENT INTERFACE CONNECTOR FOR CICS TS (CCI Connector for CICS TS), 当它声明应该除去 ECIREQUEST 时, 在 CICS TS V2.3 中引入。

事务相似性实用程序

CICS TS V3.1 不包含先前作为 CICS事务亲缘关系实用程序一部分提供的检测器和报告器组件。这些组件现在合并到 2004 年 8 月发行的 IBM CICS Interdependency Analyzer for z/OS V1.3 中, 其具有分析互相依赖性和亲缘关系。CICS 事务亲缘关系的装入库扫描程序组件在 CICS TS V3.1 中仍然可使用, 而且可在具有可能亲缘关系的应用程序上产生报告。

第 28 章 线程安全应用程序编程接口命令

CICS Transaction Server for z/OS V3R1中的大多数新命令都是线程安全的。另外，在发行版中，一些现有的命令也已线程安全。

线程安全的新命令

- 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)

现在线程安全的现有命令

- WEB ENDBROWSE FORMFIELD
- WEB ENDBROWSE HTTPHEADER
- WEB EXTRACT
- WEB READ FORMFIELD
- WEB READ HTTPHEADER
- WEB READNEXT FORMFIELD
- WEB READNEXT HTTPHEADER
- WEB RECEIVE (Server)
- WEB RETRIEVE
- WEB SEND (Server)
- WEB STARTBROWSE FORMFIELD
- WEB STARTBROWSE HTTPHEADER
- WEB WRITE HTTPHEADER

不线程安全的新命令

- START CHANNEL

第 29 章 高级语言支持

此参考主题描述 CICS 支持的高级编程语言，并提供有关 CICS 当前发行版中所支持的每种语言发行版的信息。

COBOL 编译器

编译器	程序号	服务中的编译器	CICS 转换程序支持	CICS 运行时支持	IBM Distributed Debugger 的使用 (请参阅注意事项第 320 页的 1)	与 WebSphere Studio Enterprise Developer 协同使用
OS/VSE COBOL	5740-CB1 5734-CB4 5740-LM1	否	CICS TS V1.3: 受支持 CICS TS V2.2, CICS TS V2.3, CICS TS V3.1: 不支持	CICS TS V1.3, CICS TS V2.2, CICS TS V2.3: z/OS 的 Language Environment 组件是必需的; 应用程序将在不更改的情况下运行。 CICS TS V3.1: 不支持	否	否
VS COBOL II	5668-022 5668-023 5668-958	否	CICS TS V1.3: 受支持, 带有 COBOL2 选项 CICS TS V2.2, CICS TS V2.3, CICS TS V3.1: 受支持, 仅带有 COBOL3 选项	CICS TS V1.3, CICS TS V2.2: z/OS 的 Language Environment 组件是必需的, 应用程序将在不更改的情况下运行。 CICSTS V2.3, CICS TS V3.1: z/OS 的 Language Environment 组件是必需的; CICS 将专门使用 Language Environment 运行时。应用程序行为可更改 (请参阅注意事项第 320 页的 3)。	是, 带有限制	否
SAA AD/Cycle COBOL/370	5688-197 5668-958	否	CICS TS V1.3: 受支持, 带有 COBOL2 选项 CICS TS V2.2, CICS TS V2.3, CICS TS V3.1: 受支持, 仅带有 COBOL3 选项	Language Environment	是, 带有限制	否

编译器	程序号	服务中的编译器	CICS 转换程序支持	CICS 运行时支持	IBM Distributed Debugger 的使用 (请参阅注意事项 1)	与 WebSphere Studio Enterprise Developer 协同使用
COBOL for MVS and VM	5688-197	否	CICS TS V1.3: 受支持, 带有 COBOL2 选项 CICS TS V2.2、CICS TS V2.3、CICS TS V3.1: 受支持, 仅带有 COBOL3 选项	Language Environment	是, 带有限制	否
COBOL for OS/390 and VM V2	5648-A25	是	CICS TS V1.3: 受支持, 带有 COBOL2 选项 CICS TS V2.2、CICS TS V2.3、CICS TS V3.1: 受支持, 仅带有 COBOL3 选项	Language Environment	是, 带有限制	否
COBOL for OS/390 and VM V2	5648-A25 (使用 PTF for APAR PQ45462)	是	可使用集成的转换程序 (请参阅注意事项 2)	Language Environment	是, 带有限制	是, 带有限制
Enterprise COBOL for z/OS and OS/390 V3	5655-G53	是	可使用集成的转换程序 (请参阅注意事项 2)	Language Environment	是	是

注意事项:

1. IBM Distributed Debugger 作为 WebSphere Studio Enterprise Developer V5 和其他 IBM 产品的组件提供。

要获得更多信息, 请参阅 <http://www.ibm.com/software/awdtools/debugger/>。

2. 集成的转换程序功能需要 IBM COBOL for OS/390 3 and VM V2R2, 带有 PTF for APAR PQ45462, 或者 Enterprise COBOL for z/OS and OS/390 V3。然而, 注意, COBOL3 转换程序选项必须是活动的。
3. 请参阅文章 *Language Environment within CICS TS: Questions and Answers*, 可从 <http://www-3.ibm.com/software/htp/cics/library/indexes/whitepapers.html> 获得

PL/I 编译器

编译器	程序号	服务中的编译器	CICS 转换程序支持	CICS 运行时支持	IBM Distributed Debugger 的使用 (请参阅注意事项 1)	与 WebSphere Studio Enterprise Developer 协同使用
OS PL/I Optimizing Compiler V1	5724-PLI	否	是	CICS TS V1.3, CICS TS V2.2, CICS TS V2.3: z/OS 的 Language Environment 组件是必需的; 应用程序将在不更改的情况下运行。 CICS TS V3.1: 不支持。	是, 带有限制	否
OS PL/I Optimizing Compiler V2	5668-909 5668-910 5668-911	否	是	CICS TS V1.3, CICS TS V2.2, CICS TS V2.3: z/OS 的 Language Environment 组件是必需的; 应用程序将在不更改的情况下运行。 CICS TS V3.1: 不支持。	是, 带有限制	否
SAA AD/Cycle PL/I for MVS and VM	5688-235	否	是 (请参阅注意事项 2)	Language Environment	是, 带有限制	否
PL/I for MVS and VM V1	5688-235	否	是 (请参阅注意事项 2)	Language Environment	是, 带有限制	否
VisualAge PL/I for OS/390 V2	5655-B22	否	是 (请参阅注意事项 2)	Language Environment	是, 带有限制	否
Enterprise PL/I for z/OS and OS/390 V3	5655-H31	是	可使用集成的转换程序 (请参阅注意事项 2)	Language Environment	是	是

注意事项:

1. IBM Distributed Debugger 作为 WebSphere Studio Enterprise Developer V5 和其他 IBM 产品的组件提供。

要获得更多信息, 请参阅 <http://www.ibm.com/software/awdtools/debugger/>。

2. 集成的转换程序功能需要 VisualAge PL/I for 3 OS/390, V2R2.1, 带有 PTFfor APAR PQ45562, 或者 Enterprise 3 PL/I for z/OS and OS/390 V3。

#

#

C 和 C++ 编译器

编译器	程序号	服务中的编译器	CICS 转换程序支持	CICS 运行时支持	IBM Distributed Debugger 的使用 (请参阅注意事项 1)	与 WebSphere Studio Enterprise Developer 协同使用
C/370 V1	5688-040	否	是	CICS TS V1.3, CICS TS V2.2, CICSTS V2.3: z/OS 的 Language Environment 组件是必需的; 应用程序将在不更改的情况下运行。 CICS TS V3.1: 不支持。	是, 带有限制	否
C/370 V2	5688-187 5688-188	否	是	CICS TS V1.3, CICS TS V2.2, CICSTS V2.3: z/OS 的 Language Environment 组件是必需的; 应用程序将在不更改的情况下运行。 CICS TS V3.1: 不支持。	是, 带有限制	否
SAA AD/Cycle C/370	5688-216	否	是	Language Environment	是, 带有限制	否
C/C++ for MVS/ESA	5655-121	否	是	Language Environment	是, 带有限制	否
C/C++ for OS/390	5647-A01 的组件	是	是	Language Environment	是, 带有限制	否
C/C++ for z/OS and OS/390	5694-A01 的组件	是	是	Language Environment	是	否

注意事项:

1. IBM Distributed Debugger 作为 WebSphere Studio Enterprise Developer V5 和其他 IBM 产品的组件提供。

要获得更多信息, 请参阅 <http://www.ibm.com/software/awdtools/debugger/>。

#

#

Java 支持

编译器或 JVM	程序号	服务中	CICS 转换程序支持	CICS 运行时支持	IBM Distributed Debugger 的使用 (请参阅注意事项 1)	与 WebSphere Studio Enterprise Developer 协同使用
VisualAge for Java, Enterprise Edition V2 - Enterprise Toolkit for OS/390 (请参阅注意事项 2)	5655-JAV	否	CICS TS V1.3, CICS TS V2.2, CICS TS V2.3, CICS TS V3.1: 不需要转换程序 - 使用 JCICS 类。	Language Environment	是, 带有限制	否
Java for OS/390 at SDK 1.1.8		是	仅 CICS TS V1.3: 不需要转换程序 - 使用 JCICS 类。 CICS TS V2.2, CICS TS V2.3, CICS TS V3.1: 不支持	Language Environment	是	是
Developer Kit for OS/390, Java 2 Technology Edition, V1.3.1	5655-D35	是	仅 CICS TS V2.2 不需要转换程序 - 使用 JCICS 类。 CICS TS V1.3, CICS TS V2.3, CICS TS V3.1: 不支持	Language Environment	是	是
SDK for z/OS, Java 2 Technology Edition, V1.4	5655-I56	是	CICS TS V2.3, CICS TS V3.1: 不需要转换程序 - 使用 JCICS 类。 CICS TS V1.3, CICS TS V2.2: 不支持不需要转换程序 - 使用 JCICS 类。仅 CICS TS V2.3 支持	Language Environment	是	是

注意事项:

1. IBM Distributed Debugger 作为 WebSphere Studio Enterprise Developer V5 和其他 IBM 产品的组件提供。

要获得更多信息, 请参阅 <http://www.ibm.com/software/awdtools/debugger/>。

2. Java 程序对象是使用 VisualAge for Java Enterprise Toolkit for OS/390 (ET/390) 字节码绑定程序 (它们使用 hpj 命令进行编译, 有时引用为已编译的 Java 程序或 HPI 程序) 编译的程序。要对迁移提供帮助, 在 CICS TS V2.3 中继续为 Java 程序对象提供支持。然而, 这将是提供此支持的最后一个 CICS Transaction Server for z/OS 的发行版。

建议将 Java 程序对象迁移到 CICS TS V2.3 的客户在 CICS 下使用 JVM 方式重新构建并运行它们。

第 8 部分 出版物

第 30 章 Eclipse 信息中心

在 CICS Transaction Server for z/OS V3R1 中，CICS 信息中心在 WebSphere Studio WorkBench 用户帮助系统内运行，Eclipse 框架包含很多文档插件，这些文档插件构成了信息中心。现在，信息中心的新外观（特别是欢迎页面）与其他的产品信息中心保持了一致性。

CICS TS 3.1 文档包含在单个插件中。您还可选择安装部分或所有 CICS 工具产品 - 每个工具产品存在一个插件。信息中心（也称为帮助系统）以两种方式运行。您可在工作站本地启动它，或者您可通过浏览器远程访问服务器来启动它。

从欢迎页面，您可访问有关 CICS 中最新功能的信息，找出产品的重点描述，以及到 Web 上 CICS 资源的链接。

从导航，您可通过功能区域，或通过查找与您所执行任务匹配的用户目标，以访问有关 CICS 的信息。您还可通过选择库和 PDF 查看 HTML 和 PDF 格式的书籍。

在本发行版的信息中心中实现以下功能：

- 对 Linux、Linux on zSeries 和 z/OS 的扩展平台支持。
- 新增内容部分
- 信息指南
- 学习路径
- 搜索和查找
- 故障诊断和支持
- 书签
- 用户首选项
- 修正的导航

Eclipse 信息中心的优点

移动到 Eclipse 框架有利于支持其他平台。现在，您可在 Linux RedHat 或 Linux SuSE 工作站本地运行信息中心。另外，信息中心现在可作为 Linux RedHat for zSeries、Linux SuSE for zSeries 和 z/OS 上的服务器运行。这允许您在托管信息中心的位置上操作更灵活。要获得所支持平台的完整列表，请参阅第 328 页的『需求』。

信息中心的安装已大大简化，所以不再有 GUI 安装。与信息中心一起提供的自述文件说明了如何安装 Eclipse 帮助系统。Eclipse 的结构意味着您可以将多个文档插件放入一个帮助系统，其中包括其他产品的插件。这允许您定制信息中心中包含的信息。另外，您可将 CICS 文档插件集成到基于 Eclipse 的 IDE。

对 CICS 文档的搜索能力已改进，这是因为 Eclipse 使用 Lucene 搜索引擎。NetQuestion 不再是必需的。现在，当您在信息中心中搜索时，会在 HTML 内容中突出显示每个字符串查询实例。

已对信息中心进行大量导航改进。导航中的两个新功能是学习路径和信息指南。学习路径是帮助您更快速了解新功能的一序列主题。信息指南是单个主题，其包含到关于 CICS 特殊功能区域的各种链接。信息指南的设计有助于您在信息中心和 Web 上查找有用的资源，并提供链接指向何处的指导。

导航中还存在故障诊断和支持部分，为您提供有关如何搜索知识库、如何从 IBM 获得修订，以及如何报告问题的帮助。特别的优点在于 Web 搜索主题，其允许您查询 CICS 支持 Web 站点，以查找问题的解决方案。此主题还包含 Google 搜索工具。另外，提供多个发行版的关键技术说明，帮助您快速对问题进行故障诊断。

术语

本主题包含 Eclipse 信息中心使用的术语。

插件 它是一个自包含的软件组件，它修改（添加或更改）特定软件系统中的功能。用户将插件添加到软件系统时，原来的软件系统仍然完好。在文档插件的情况下，插件包含与产品文档相关的所有文件以及如何在 Eclipse 中查看它。

学习路径

一系列信息中心主题可以帮助新用户学习产品的功能区域

信息指南

包含到信息中心中和 Web 上信息资源的综合链接集，以及每个链接为用户所显示内容的指导的主题。

本地方式

使用 *IC_local_start* 文件对工作站上信息中心的实施。信息中心在系统缺省浏览器中自动启动。

服务器方式

使用 *IC_server_start* 文件对服务器上信息中心的实施。信息中心在特定端口上运行，用户可使用浏览器远程访问信息中心。

需求

信息中心需要运行下列操作系统：

- Windows 2000
- Windows XP
- AIX 5.2 和 5.3
- Linux RedHat Enterprise 3.0
- Linux SuSE Enterprise 3.0
- Linux RedHat Enterprise 8 and 9 for zSeries
- Linux SuSE Enterprise 8 and 9 for zSeries
- z/OS 1.4 或更高版本

请注意，Linux for zSeries 和 z/OS 上对信息中心的支持仅以服务器方式提供。

CICS TS 3.1 信息中心使用 JRE。每个平台的 JRE 与信息中心一起提供，除了 z/OS。如果您要在 z/OS 上运行信息中心，您需要使用与操作系统一起提供的 JRE。

要在查看信息中心时获得最好的结果，建议您使用下列某个浏览器：

- Microsoft Internet Explorer 6.0
- Mozilla 1.7

“新增内容”部分

新增内容部分是新的导航结构，它包含与新发行版相关的所有信息，并且完全与信息中心的其余部分集成在一起。

要访问描述 CICS 新功能的信息，您可使用信息中心欢迎页面上的链接作为开始点。此信息分为此发行版的三个主题：CICS 集成、应用程序转换和企业管理。您还可通过展开“新增内容”部分进行导航，找到您所查找的特殊功能。

信息指南

信息指南是单个主题，其包含到关于 CICS 特殊功能区域的各种资源的链接。

每个指南分为几个部分，这在主题顶部的目录中进行概述。这些部分涉及各个区域，如基本技能、安装、迁移、管理、应用程序开发和安全性。每个部分的链接表明资源对于信息中心是内部的还是外部的，以及提供资源所包含内容的摘要。资源包括信息中心主题、红皮书、教程、白皮书和文章。每个指南的结尾是教学部分，包含进一步的阅读内容，到课程和教学材料的链接。

信息中心中存在三个信息指南。它们涉及下列功能：

- Web Services
- CICSplex System Manager
- CICS 中的 Java支持和编程

您可从信息中心的欢迎页面直接访问信息指南，或者在内容窗格中展开导航。

学习路径

学习路径是帮助您了解 CICS 区域的一组主题。

每个学习路径具有一个介绍，说明了目标和计划读者。学习路径设计为顺序阅读，但是介绍包含学习路径的所有步骤，以便您在需要时可随时启动。在每个主题的底部提供按钮导航，以通过路径移动。还存在对进一步信息的链接，您可随时离开此路径，以找到有关特殊功能或概念的更多信息。学习路径的最后是摘要主题。它概述您已了解的内容，并提供进一步阅读和信息中心中后继任务的有用链接。



本发行版的学习路径涉及以下方面：

- Web Service。
- 安装 CICSplex SM: WUI 方案。

信息中心中搜索的技术

您可使用多种方法搜索信息中心，这取决于您所查找的信息，以及信息中心的运行方式。

要在信息中心中执行搜索，在**搜索**字段中输入查询内容。**内容**窗格显示排在前 500 位的结果。单击搜索结果查看该主题。您查询中的字将会突出显示。

要在导航树和搜索结果列表之间切换，单击“内容”选项卡 () 或**内容**窗格底部的“搜索结果”选项卡 ()。

搜索全字匹配的字或短语

您可以通过在搜索短语上加上双引号，将它识别为全字匹配的字符串。例如，“log file”将搜索字符串 *log file*，而不是分开的两个字 *log* 和 *file*。不加引号，搜索条件将被解释为 log AND file。

仅在英语和德语中，搜索引擎“衍生”其他格式的单个搜索字。例如，搜索字 *challenge* 还会查找字 *challenging*。您不想“衍生”时，请使用双引号。

用通配符搜索

您可以使用下列通配符：

- * 星号 - 用于条件中的多个未知或可变字符。例如，搜索条件 *par** 将返回 *partly*、*participate*、*partial* 和其他以 *par* 开头的词。
- ? 问号 - 用于单个未知或可变字符。例如，搜索条件 *par?* 将返回 *part* 而不是 *partial* 或 *partly*。

用布尔运算符 (AND、OR、NOT) 搜索

您可以在搜索条件中插入二目运算符：AND、OR 和 NOT。例如：

database AND "log file"：将您的搜索缩小为必须包含条件 *database* 和 *"log file"* 的主题。

database OR "log file"：将您的搜索放宽为包含 *database* 或 *"log file"* 的主题。

database NOT "log file"：搜索包含 *database* 而没有 *log file* 的主题。

缩小您的搜索范围

缺省情况下，搜索“内容”窗格中显示的所有主题。然而，可以将您的搜索缩小到一组特定的主题。如果您在本地运行信息中心，则您可使用搜索列表缩小搜索范围，并保存搜索列表供以后再次使用。

要创建搜索列表：

1. 单击“搜索”字段旁的**搜索范围**：链接。“选择搜索范围”窗口会打开。
2. 选择**仅搜索下列主题**并选择“新建”按钮。“新建搜索列表”窗口会打开。
3. 在**要搜索的主题**列表中，选择要包括在您的搜索中的导航类别。您可以展开类别以仅选择某些子类别。
4. 在**列表名**：文本字段中，为您的搜索列表输入一个合适的名称，然后选择**确定**按钮。

您可以使用已创建的搜索列表，来搜索匹配您的搜索表达式的主题。

如果您在服务器上运行信息中心，则您可使用“搜索”字段旁的高级搜索：链接，将您的搜索范围缩小到导航的特殊部分。




搜索消息、命令和参数

为了帮助您在文档中快速找到特定引用项，如消息或命令，在导航中提供了查找工具。在导航中选择查找命令、消息、参数和更多内容，以使用查找工具。此工具不使用 Lucene 搜索引擎，但是使用索引作为替代。您可在下列区域中搜索：


- API 命令
- API 参数
- SPI 命令
- SPI 参数
- CICS 事务
- 消息和代码
- 跟踪条目
- 资源定义
- 系统初始化参数
- 词汇术语
- 图标题
- 表标题

浏览信息中心



要浏览整个信息中心，您可在内容窗格中选择或展开链接。您还可使用同步和键盘快捷方式这样的功能。

帮助窗口左边的内容窗格 () 显示目录中的主题标题或导航树结构。您可单击主题标题以在主面板右侧显示内容，或者您可展开导航以查看嵌套部分。如果您选择使用 HTML 主题中的链接，则您可使用后退 () 和前进 () 按钮在已查看主题的历史记录内浏览。

每个 HTML 主题包含一组链接，位于主题的底部，此设计有助于您导航到相关的信息。另外，很多主题包含按钮导航，将您带到此主题所属书籍的目录、书籍中的下一个和上一个主题，最后达到索引。

如果您运行搜索，并要返回到目录，选择窗格底部的“内容”选项卡 ()。任何时候，您都可以使用窗格底部的选项卡在导航和最后的搜索结果之间移动。

同步目录

您在单击主题内的链接时，导航树不会自动更改来显示和突出该新的主题。要查看新主题处于导航树中的哪个位置并且同步两个视图，单击刷新按钮 () 或在目录中显示 () 按钮。当前显示的主题的主题标题将在导航树中突出显示。

使用键盘浏览



使用下列键组合浏览信息中心:

- 要从信息中心的一个面板内转至下一个链接、按钮或主题节点, 按 **Tab** 键。
- 要展开和折叠树中的节点, 按右箭头键和左箭头键。
- 要移动至下一个主题节点, 按向下箭头键或 **Tab** 键。
- 要移动至上一个主题节点, 按向上箭头键或 **Shift+Tab** 键。
- 要全部上卷或下卷, 按 **Home** 键或 **End** 键。
- 要后退, 按 **Alt+Left** 箭头键; 要前进, 按 **Alt+Right** 箭头键。
- 要到下一个面板, 按 **Ctrl+Tab** 键。
- 要到上一个面板, 按 **Shift+Ctrl+Tab** 键。
- 要打印活动的面板, 按 **Ctrl+P** 键。

将主题制作成书签

存在两种方法可在信息中心中为主题做书签。为主题做书签的方法取决于您是在本地运行信息中心, 还是在服务器上运行。

当您在工作站本地运行信息中心时, 您可通过执行这些步骤为所选主题做书签:

1. 选择主面板之上的“书签”按钮 ()。这将在“书签”面板中放置一个书签。
2. 单击“内容”窗格底部的“书签”视图图标 ()。“书签”窗格将替换“内容”窗格, 以显示您的所有书签。您选择的主题将出现在书签列表的底部。

如果您要删除该书签, 请选择它, 然后使用 **Delete** 键。

当您访问服务器上的信息中心时, 您可使用浏览器为主题做书签。

用户首选项

用户首选项允许您定制信息中心功能的某些部分。

用户首选项是交互式的表、可访问的语法图, 以及文本 / 图像选项。这些首选项现在可从导航的**产品**和**信息中心概述**部分获得。

第 31 章 CICS Transaction Server for z/OS库

CICS Transaction Server for z/OS 信息中心是 CICS Transaction Server 用户信息的主要来源。CICS TS 出版物的小子集（资格集合）以硬拷贝格式提供。

信息中心包含：

- HTML 格式表示的 CICS Transaction Server 信息。
- 作为 Adobe 可移植文档格式（PDF）文件提供的许可和未许可 CICS Transaction Server 书籍。您可使用这些文件打印书籍的硬拷贝。
- HTML 格式和 PDF 文件表示的相关产品信息。

在 CD-ROM 上，CICS 信息中心的一个副本与产品一起自动提供。通过指定 Information Center 功能 部件号 7014，可以订购更多的副本，而无需支付额外的费用。您还将接收到必备硬拷贝书籍的小集合。

只有产品的被许可方才能使用许可文档。仅包含未许可信息的信息中心版本可通过出版物订购系统（订单号码 SK3T-6945）获得。

作为硬拷贝提供的书籍

当您订购CICS Transaction Server for z/OS V3R1时，您将接收到少数硬拷贝书籍。

硬拷贝书籍为：

Memo to Licensees, G110-2559

CICS Transaction Server for z/OS Program Directory, G110-2586

CICS Transaction Server for z/OS发行指南, G151-0180

CICS Transaction Server for z/OS Licensed Program Specification, GC34-6608

您可使用上述提供的订单号订购下列书籍的未来副本：

CICS Transaction Server for z/OS发行指南

CICS Transaction Server for z/OS安装指南

CICS Transaction Server for z/OS Licensed Program Specification

仅 PDF 书籍

许可和未许可的 CICS Transaction Server 书籍都在 CICS信息中心中作为 Adobe 可移植文档格式（PDF）文件提供。您可使用这些文件打印书籍的硬拷贝。

CICS 书籍，用于CICS Transaction Server for z/OS

常规

CICS Transaction Server for z/OS Program Directory, G110-2586

CICS Transaction Server for z/OS发行指南, G151-0180

CICS Transaction Server for z/OS Migration from CICS TS Version 1.3, GC34-6423

CICS Transaction Server for z/OS Migration from CICS TS Version 2.2, GC34-6424

CICS Transaction Server for z/OS Migration from CICS TS Version 2.3, GC34-6425

CICS Transaction Server for z/OS 安装指南, G151-0045

访问 CICS

CICS 因特网指南, S151-0049

CICS Web Services Guide, SC34-6458

管理

CICS System Definition Guide, SC34-6428

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

编程

CICS Application Programming Guide, SC34-6433

CICS Application Programming Reference, SC34-6434

CICS System Programming Reference, SC34-6435

CICS Front End Programming Interface User's Guide, SC34-6436

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

诊断

CICS Problem Determination Guide, SC34-6441

CICS Performance Guide, SC34-6452

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

数据库

CICS DB2 Guide, SC34-6457

CICS IMS Database Control Guide, SC34-6453

CICS Shared Data Tables Guide, SC34-6455

CICSplex SM 书籍, 用于CICS Transaction Server for z/OS

常规

CICSplex SM Concepts and Planning, SC34-6459

CICSplex SM User Interface Guide, SC34-6460

CICSplex SM Web User Interface Guide, SC34-6461

管理和操纵

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

编程

CICSplex SM Application Programming Guide, SC34-6468
CICSplex SM Application Programming Reference, SC34-6469

诊断

CICSplex SM Resource Tables Reference, SC34-6470
CICSplex SM Messages and Codes, GC34-6471
CICSplex SM Problem Determination, GC34-6472

CICS系列书籍

通信

CICS Family: Interproduct Communication, SC34-6473
CICS Family: Communicating from CICS on System/390, SC34-6474

许可出版物

下列许可出版物不包含在未许可版本信息中心中:

CICS Diagnosis Reference, LY33-6102
CICS Data Areas, LY33-6103
CICS Supplementary Data Areas, LY33-6104
CICS Debugging Tools Interfaces Reference, LY33-6105

可访问性

可访问性功能部件帮助残疾（例如活动受限或视力不好）用户成功地使用软件产品。

您可以使用下列方式之一执行安装、运行和维护 CICS 系统所需的大多数任务：

- 使用 3270 仿真器登录到 CICS
- 使用 3270 仿真器登录到 TSO
- 使用 3270 仿真器作为 MVS 系统控制台

IBM 个人通信提供 3270 仿真，为残疾人员提供辅助功能选项功能。您可以使用这个产品来提供您在 CICS 系统中需要的可访问性功能部件。

当使用作为 WebSphere Application Server 组件的 Application Assembly Tool for Enterprise bean (ATK 和 AAT) 时，一些辅助功能选项功能不可用。您应该查询 WebSphere Application Server 附带的文档，以确定在使用这些工具时，那些可访问性功能部件是可用的。

如果您使用企业 bean 的资源管理器处理 EJB 资源，则可访问性功能部件是 Web 浏览器提供的那些。特别注意，当您允许鼠标指针悬浮在显示部分上出现的帮助还可通过面板上的帮助功能获得。

索引

[A]

- 安全性
 - 新的 ESM 设施概要文件 252
- 安装
 - 新过程 271

[B]

- 必备硬件 311
- 表示法
 - 语法 xii
- 不支持 Web 的应用程序 105

[C]

- 程序, 样本 205
- 持续连接 88, 89
 - 作为 HTTP 客户端的 CICS 62
- 传输层安全性协议 144
- 创建通道 166

[D]

- 大小写混合
 - 密码 277
- 当前通道 166
 - 概述 170
- 动态路由程序 201
- 动态选择列表 259
- 对 CICS 外部的更改
 - 跟踪点 208
 - 监视 203
 - 统计信息 204
 - 问题确定 205
 - 消息 205
 - 异常终止代码 205
 - 用户可替换程序 201
 - API 187
 - CICS 样本程序 205

[F]

- 发现从链接返回的容器 175
- 发现已通过程序的容器 175
- 方案
 - 多个交互式组件 165
 - 几个通道, 一个组件 164
 - 一个通道, 几个程序 164

- 方案 (续)
 - 一个通道, 一个程序 163
- 分布式路由程序 201
- 分块 87
- 分块的传输编码 87
- 分析器程序 105, 129
 - 分块的传输编码 87

[G]

- 概述
 - 创建通道 166
 - 当前通道 166
 - 典型方案 163
 - 发现从链接返回的容器 175
 - 发现已通过程序的容器 175
 - 构造通道 177
 - 基本示例 160
 - 容器 160
 - 设计通道 176
 - 使用来自 JCICS 的通道 180
 - 数据转换 184
 - 通道 160
 - 通道的作用域 171
 - 通道和 BTS 活动 178
 - 与通道一起动态路由 183, 210
 - 只读容器 175
 - 组件 164
- 高级语言结构
 - 转换为 WSDL 26
- 跟踪点 208
- 构造通道 177
- 管道传送 88
- 过滤器
 - 展开和折叠 261
- 过滤器确认 258

[H]

- 会话标记 61, 62

[J]

- 加上 32K COMMAREA 159, 160, 187, 188
- 监视 203
- 结果集警告计数 257

[L]

- 连接, 持续 88
- 两列结构的详细视图
 - 创建 263
 - 删除项 264

[M]

- 幂等性 88
- 密码
 - 大小写混合 277

[P]

- 批处理实用程序
 - Web Service 助手 25

[Q]

- 迁移
 - 开发新功能 208
 - 未开发新功能 208
- 取消全选图标 262
- 全选图标 262

[R]

- 容器
 - 创建 180
 - 发现从链接返回的 175
 - 发现哪个程序已通过 175
 - 概述 159
 - 基本示例 160
 - 上下文, BTS 或通道 179
 - 设计通道 176
 - 使用来自 JCICS 的 180
 - 只读 175
- 容器命令
 - DELETE CONTAINER (CHANNEL) 188
 - GET CONTAINER (CHANNEL) 188
 - MOVE CONTAINER (CHANNEL) 190
 - PUT CONTAINER (CHANNEL) 193

[S]

- 商业应用程序服务 254
- 上下文
 - 容器, BTS 或通道 179
- 设计通道 176
- 实用程序
 - Web Service 助手 25

- 使用来自 JCICS 的通道 180
- 示例
 - 多个交互式组件 165
 - 构造和使用通道的 Java 客户机程序 183
 - 构造通道的 CICS 客户机程序 177
 - 基本 160
 - 几个通道, 一个组件 164
 - 简单客户机程序比作 BTS 活动 178
 - 使用通道的 CICS 服务器程序 178
 - 一个通道, 几个程序 164
 - 一个通道, 一个程序 163

视图, 新建

- Web 用户界面
 - 功能提供 213
 - 管道 57
 - 管道定义 57
 - 通道使用情况 213
 - Host 140
 - URI 映射 140
 - URI 映射定义 140
 - URI 映射全局 140
 - Web Service 57
 - Web Service 定义 57

视图, 已更改

- 最终用户界面, CICSplex SM
 - DOCDEF 135
 - DOCTEMP 135
 - EJCODEF 152
 - EJCOSE 152
 - PROGDEF 303
 - PROGRAM 303
 - TCPDEF 135, 152
 - TCPIPS 152
 - TERMDEF 297

Web 用户界面

- 程序 229, 304
- 程序定义 229, 304
- 工作请求 57
- 时钟和计时 155
- 文档模板 140
- 文档模板定义 140
- 终端定义 297
- CICS 区域 155
- CorbaServer 155
- CorbaServer 定义 155
- CPU 和 TCB 信息 229
- ISC/MRO 连接 213
- TCP/IP 服务 155
- TCP/IP 服务定义 140, 155
- TCP/IP 全局状态 155
- TCP/IP 使用情况 140

- 收藏夹 247, 249

收藏夹编辑器 247
数据转换 184
 代码页 283
 和通道 184
数据资源库
 新消息 268

[T]

通道
 比作 BTS 活动 178
 创建 166, 180
 当前 166, 170
 典型方案
 多个交互式组件 165
 几个通道, 一个组件 164
 一个通道, 几个程序 164
 一个通道, 一个程序 163
 动态和分布式路由 183, 210
 发现从链接返回的容器 175
 发现已通过程序的容器 175
 概述 159
 构造 177
 基本示例 160
 设计 176
 使用来自 JCICS 的 180
 益处 159
 只读容器 175
 作为 large COMMAREA 159
 作用域 171
通道的作用域 171
通道命令
 DELETE CONTAINER (CHANNEL) 188
 GET CONTAINER (CHANNEL) 188
 MOVE CONTAINER (CHANNEL) 190
 PUT CONTAINER (CHANNEL) 193
 START CHANNEL 195
统计信息 204
图
 语法 xii

[W]

尾部 87
尾部头 87
问题确定
 跟踪点 208
 消息 205
 异常终止代码 205

[X]

系统初始化参数
 CLINTCP 284
 SRVERCP 284
先决条件
 硬件 311
线程安全 API 命令
 此发行版中新增 317
详细视图
 两列结构 260
消息 205, 265, 268
新的 API 命令
 EXTRACT STATISTICS 273
信息中心
 搜索技术 330
 信息指南 329
 需求 328
 学习路径 329
 做书签 332
信息中心浏览 331
虚拟主管 89, 90
需求, 硬件 311

[Y]

样本程序 205
已撤销的用户标识 287
异常终止代码 205
益处 159
应用程序编程接口 (API)
 新 API 命令 188
 已修改的 API 命令 197
 JCICS 更改 200
应用程序编程接口, CICSplex SM
 新资源表
 HOST 135
 PIPEDEF 55
 PIPELINE 55
 URIMAP 135
 URIMPDEF 135
 URIMPGBL 135
 WEBSERV 55
 WEBSVDEF 55
 已更改的资源表
 CICSRGN 153, 228, 303
 CONNECT 211
 DOCDEF 135
 DOCTEMP 135
 EJCODEF 153
 EJCOSE 153
 PROGDEF 228, 303

应用程序编程接口, CICSplex SM (续)

已更改的资源表 (续)

- PROGRAM 228, 303
- TASK 135, 153, 211, 228
- TCPDEF 135, 153
- TCPIPGBL 153
- TCPIPS 153
- TERMDEF 297
- WORKREQ 55

硬件先决条件 311

用户编辑器 249

用户对象 247, 250

用户可替换程序

 动态路由程序 201

 分布式路由程序 201

用户收藏夹 247

 管理 247, 249

用户组概要文件 251

 创建 253

 创建和管理 252

与通道一起动态路由 183, 210

语法表示法 xii

语言结构

 转换为 WSDL 26

远程 MAS 305

[Z]

证书撤回列表 145

支持 Web 的应用程序 105

只读容器 175

助手, Web service 25

转换器程序 105, 130

资源分配视图 254

组件

 多个, 交互式 165

 几个通道, 一个组件 164

 一个通道, 几个程序 164

最终用户界面, CICSplex SM

 已更改的视图

 DOCDEF 135

 DOCTEMP 135

 EJCODEF 152

 EJCOSE 152

 PROGDEF 303

 PROGRAM 303

 TCPDEF 135, 152

 TCPIPS 152

 TERMDEF 297

作为 HTTP 客户机的 CICS

 管道化的请求 88

作为 large COMMAREA 的通道 159, 160, 163, 187

A

ABSTIME 选项

 CONVERTTIME 命令 122

ACTION 选项

 WEB CONVERSE 命令 76

 WEB SEND 命令 (服务器) 94

 WEB SEND 命令 (客户机) 68

ANALYZER 属性

 URIMAP 定义 108

ANALYZERSTAT 选项

 INQUIRE URIMAP 命令 124

API (应用程序编程接口)

 新 API 命令 188

 已修改的 API 命令 197

 JCICS 更改 200

API 更改

 CICSplex SM 265, 267

API 命令

 EXTRACT STATISTICS 273

AS 选项

 MOVE CONTAINER (CHANNEL) 命令 192

B

BAS 254

BATCHREP 资源表 267

big COMMAREA 159, 160, 163, 187

big COMMAREAs, channels 195

BINDING 选项

 INQUIRE WEBSERVICE 命令 46

BTS 活动 178

C

CCRL 事务 145

CERTIFICATE 属性

 URIMAP 定义 108

CERTIFICATE 选项

 INQUIRE URIMAP 命令 124

 WEB OPEN 命令 65

CHANNEL 选项

 ASSIGN 命令 197

 DELETE CONTAINER (CHANNEL) 命令 188

 GET CONTAINER (CHANNEL) 命令 189

 LINK 命令 198

 MOVE CONTAINER (CHANNEL) 命令 192

 PUT CONTAINER (CHANNEL) 命令 193

 RETURN 命令 198

 START TRANSID (CHANNEL) 命令 196

 XCTL 命令 199

- CHANNELERR 条件
 - LINK 命令 198
 - RETURN 命令 199
 - XCTL 命令 200
- channels as large COMMAREAs 195
- CHARACTERSET 属性
 - URIMAP 定义 109
- CHARACTERSET 选项
 - INQUIRE URIMAP 命令 124
 - WEB CONVERSE 命令 79
 - WEB RECEIVE 命令 (服务器) 97
 - WEB SEND 命令 (服务器) 94
 - WEB SEND 命令 (客户机) 69
- CHUNKING 选项
 - WEB SEND 命令 (服务器) 94
 - WEB SEND 命令 (客户机) 69
- CICS 资源定义 254
- CICS Web 支持
 - 持续连接 89
 - 分块的传输编码 87
 - 管道传送 88
 - 虚拟主管 90
 - 用户出口 XWBOPEN、XWBSNDO 81, 83
- CICS Web 支持命令
 - CONVERSE WEB 74
 - WEB CLOSE 66
 - WEB CONVERSE 74
 - WEB EXTRACT 118
 - WEB OPEN 64
 - WEB PARSE URL 115
 - WEB RECEIVE 97
 - WEB RECEIVE (客户机) 72
 - WEB SEND (服务器) 93
 - WEB SEND (客户机) 68
- CICSplex SM
 - 过滤器确认 258
 - 结果集警告计数 257
 - 新功能 246
 - 新消息 265, 268
 - 选择列表 259
 - 业务应用程序服务 (BAS) 254
 - 用户收藏夹 247, 249
 - 用户组概要文件 251, 252
 - 远程 MAS 305
 - API 更改 265, 267
 - BATCHREP 工具 267
 - Web 用户界面 268
- CICSplex SM, 应用程序编程接口
 - 新资源表
 - HOST 135
 - PIPEDEF 55
 - PIPELINE 55
 - CICSplex SM, 应用程序编程接口 (续)
 - 新资源表 (续)
 - URIMAP 135
 - URIMPDEF 135
 - URIMPGBL 135
 - WEBSERV 55
 - WEBSVDEF 55
 - 已更改的资源表
 - CICSRGN 153, 228, 303
 - CONNECT 211
 - DOCDEF 135
 - DOCTEMP 135
 - EJCODEF 153
 - EJCOSE 153
 - PROGDEF 228, 303
 - PROGRAM 228, 303
 - TASK 135, 153, 211, 228
 - TCPDEF 135, 153
 - TCPIPGBL 153
 - TCPIPS 153
 - TERMDEF 297
 - WORKREQ 55
 - CICSplex SM, 最终用户界面
 - 已更改的视图
 - DOCDEF 135
 - DOCTEMP 135
 - EJCODEF 152
 - EJCOSE 152
 - PROGDEF 303
 - PROGRAM 303
 - TCPDEF 135, 152
 - TCPIPS 152
 - TERMDEF 297
 - CIPHERS 属性
 - URIMAP 定义 109
 - CIPHERS 选项
 - INQUIRE URIMAP 命令 124
 - WEB OPEN 命令 65
 - CLIENTCONV 选项
 - WEB CONVERSE 命令 80
 - WEB RECEIVE 命令 (客户机) 73
 - WEB SEND 命令 (客户机) 70
 - CLNTCODEPAGE 选项
 - WEB RECEIVE 命令 (服务器) 97
 - WEB SEND 命令 (服务器) 95
 - CLOSESTATUS 选项
 - WEB CONVERSE 命令 76
 - WEB SEND 命令 (服务器) 95
 - WEB SEND 命令 (客户机) 69
 - CODEPAGE 选项
 - WEB OPEN 命令 65
 - COMMAREA > 32K 159, 160, 187, 188

CONTAINER 选项

- DELETE CONTAINER (CHANNEL) 命令 188
- GET CONTAINER (CHANNEL) 命令 189
- INQUIRE WEBSERVICE 命令 46
- MOVE CONTAINER (CHANNEL) 命令 192
- PUT CONTAINER (CHANNEL) 命令 193

CONVERSE WEB 命令 74

CONVERTER 属性

- URIMAP 定义 109

CONVERTER 选项

- INQUIRE URIMAP 命令 124

CONVERTTIME 命令 121

CRL (证书撤回列表) 145

CVDA 值

CHUNKEND

- WEB SEND 命令 (服务器) 95
- WEB SEND 命令 (客户机) 69

CHUNKNO

- WEB SEND 命令 (服务器) 94
- WEB SEND 命令 (客户机) 69

CHUNKYES

- WEB SEND 命令 (服务器) 94
- WEB SEND 命令 (客户机) 69

CLICONVERT

- WEB CONVERSE 命令 80
- WEB RECEIVE 命令 (客户机) 73
- WEB SEND 命令 (客户机) 70

CLIENT

- INQUIRE URIMAP 命令 126

CLOSE

- WEB CONVERSE 命令 76
- WEB SEND 命令 (服务器) 95
- WEB SEND 命令 (客户机) 69

DELETE

- WEB CONVERSE 命令 71, 77

DISABLED

- INQUIRE HOST 命令 100
- INQUIRE URIMAP 命令 124, 125
- SET HOST 命令 101
- SET URIMAP 命令 127

DISABLEDHOST

- INQUIRE URIMAP 命令 124

ENABLED

- INQUIRE HOST 命令 100
- INQUIRE URIMAP 命令 124, 125
- SET HOST 命令 101
- SET URIMAP 命令 127

EVENTUAL

- WEB SEND 命令 (服务器) 94

EXPECT

- WEB CONVERSE 命令 76
- WEB SEND 命令 (客户机) 68

CVDA 值 (续)

GET

- WEB CONVERSE 命令 77
- WEB SEND 命令 (客户机) 71

HEAD

- WEB CONVERSE 命令 77
- WEB SEND 命令 (客户机) 71

HTTP

- WEB EXTRACT 命令 120
- WEB OPEN 命令 66

HTTPNO

- WEB EXTRACT 命令 120
- WEB RECEIVE 命令 (服务器) 99

HTTPS

- WEB EXTRACT 命令 120
- WEB OPEN 命令 66

HTTPYES

- WEB EXTRACT 命令 120
- WEB RECEIVE 命令 (服务器) 99

IMMEDIATE

- WEB SEND 命令 (服务器) 94

NO

- INQUIRE URIMAP 命令 124

NOCLICONVERT

- WEB CONVERSE 命令 80
- WEB RECEIVE 命令 (客户机) 73
- WEB SEND 命令 (客户机) 70

NOCLOSE

- WEB CONVERSE 命令 76
- WEB SEND 命令 (服务器) 95
- WEB SEND 命令 (客户机) 70

NOINCONVERT

- WEB CONVERSE 命令 80

NONE

- INQUIRE URIMAP 命令 125
- SET URIMAP 命令 127

NOOUTCONVERT

- WEB CONVERSE 命令 80

NOSRVCONVERT

- WEB RECEIVE 命令 (服务器) 98
- WEB SEND 命令 (服务器) 96

OPTIONS

- WEB CONVERSE 命令 77
- WEB SEND 命令 (客户机) 71

PERM

- INQUIRE URIMAP 命令 126

PERMANENT

- SET URIMAP 命令 127

PIPELINE

- INQUIRE URIMAP 命令 126

PUT

- WEB CONVERSE 命令 77

CVDA 值 (续)
 PUT (续)
 WEB SEND 命令 (客户机) 71
 RFC1123
 FORMATTIME 命令 122
 SERVER
 INQUIRE URIMAP 命令 126
 SRVCONVERT
 WEB RECEIVE 命令 (服务器) 98
 WEB SEND 命令 (服务器) 96
 TEMP
 INQUIRE URIMAP 命令 126
 TEMPORARY
 SET URIMAP 命令 127
 TRACE
 WEB CONVERSE 命令 77
 WEB SEND 命令 (客户机) 71
 YES
 INQUIRE URIMAP 命令 124

D

DATATYPE 选项
 PUT CONTAINER (CHANNEL) 命令 194
 DATESTRING 选项
 CONVERTTIME 命令 121
 FORMATTIME 命令 122
 DEFAULTWARNCNT WUI 参数 257
 DELETE CONTAINER (CHANNEL) 命令 188
 DFHCNV 283
 对宏的更改 284
 DFHLS2WS
 编目过程 26
 DFHWBEP (Web 出错程序) 131
 DFHWS2LS
 编目过程 29
 DNS 服务器 89
 DOCTOKEN 选项
 WEB CONVERSE 命令 76
 WEB SEND 命令 (服务器) 95
 WEB SEND 命令 (客户机) 70

E

ENABLESTATUS 选项
 INQUIRE HOST 命令 100
 INQUIRE URIMAP 命令 124
 SET HOST 命令 101
 SET URIMAP 命令 127
 ENDPOINT 选项
 INQUIRE WEBSERVICE 命令 46
 EXTRACT STATISTICS 命令 273

F

FILTERSTYLE 初始化参数 262
 FLENGTH 选项
 GET CONTAINER (CHANNEL) 命令 189
 PUT CONTAINER (CHANNEL) 命令 194
 FROM 选项
 PUT CONTAINER (CHANNEL) 命令 194
 WEB CONVERSE 命令 77
 WEB SEND 命令 (服务器) 95
 WEB SEND 命令 (客户机) 70
 FROMCCSID 选项
 PUT CONTAINER (CHANNEL) 命令 194
 FROMLENGTH 选项
 WEB CONVERSE 命令 77
 WEB SEND 命令 (服务器) 95
 WEB SEND 命令 (客户机) 70

G

GET CONTAINER (CHANNEL) 命令 188

H

HFSFILE 属性
 URIMAP 定义 110
 HFSFILE 选项
 INQUIRE URIMAP 命令 124
 HOST 属性
 URIMAP 定义 110
 HOST 选项
 INQUIRE HOST 命令 100
 INQUIRE URIMAP 命令 124
 WEB EXTRACT 命令 118
 WEB OPEN 命令 65
 WEB PARSE URL 命令 116
 HOSTCODEPAGE 属性
 URIMAP 定义 110
 HOSTCODEPAGE 选项
 INQUIRE URIMAP 命令 124
 WEB RECEIVE 命令 (服务器) 98
 WEB SEND 命令 (服务器) 95
 HOSTLENGTH 选项
 WEB EXTRACT 命令 119
 WEB OPEN 命令 65
 WEB PARSE URL 命令 116
 HTTP 请求和响应处理
 持续连接 89
 分块的传输编码 87
 管道传送 88
 作为 HTTP 服务器的 CICS 105
 作为 HTTP 客户机的 CICS 61

HTTPMETHOD 选项
 WEB EXTRACT 命令 119
HTTPRNUM 选项
 WEB OPEN 命令 65
HTTPVERSION 选项
 WEB EXTRACT 命令 119
HTTPVNUM 选项
 WEB OPEN 命令 65

I

INQUIRE HOST 命令 99
INQUIRE URIMAP 命令 123
INQUIRE WEBSERVICE 命令 46
INTO 选项
 GET CONTAINER (CHANNEL) 命令 190
 WEB CONVERSE 命令 78
 WEB RECEIVE 命令 (服务器) 98
 WEB RECEIVE 命令 (客户机) 73
INTOCCSID 选项
 GET CONTAINER (CHANNEL) 命令 190
INVOKE WEBSERVICE 命令 39

J

JCICS
 创建容器 180
 创建通道 180
 从容器获取数据 182
 更改 200
 和通道 180
 接收当前通道 181
 浏览当前通道 182
 示例程序 182

L

large COMMAREA 159, 160, 163, 187
large COMMAREAs, channels 195
LASTMODTIME 选项
 INQUIRE WEBSERVICE 命令 47
LASTRESET 选项
 EXTRACT STATISTICS 命令 275
LASTRESETHRS 选项
 EXTRACT STATISTICS 命令 275
LASTRESETMIN 选项
 EXTRACT STATISTICS 命令 275
LASTRESETSEC 选项
 EXTRACT STATISTICS 命令 275
LENGTH 选项
 WEB RECEIVE 命令 (服务器) 98
 WEB RECEIVE 命令 (客户机) 73

LENGTH 选项 (续)
 WEB SEND 命令 (服务器) 96
LOCATION 属性
 URIMAP 定义 111
LOCATION 选项
 INQUIRE URIMAP 命令 125
 SET URIMAP 命令 127

M

MAXLENGTH 选项
 WEB CONVERSE 命令 78
 WEB RECEIVE 命令 (服务器) 98
 WEB RECEIVE 命令 (客户机) 73
MEDIATYPE 属性
 URIMAP 定义 111
MEDIATYPE 选项
 INQUIRE URIMAP 命令 125
 WEB CONVERSE 命令 77, 78
 WEB RECEIVE 命令 (客户机) 73
 WEB SEND 命令 (服务器) 96
 WEB SEND 命令 (客户机) 70
METHOD 选项
 WEB CONVERSE 命令 77
 WEB SEND 命令 (客户机) 70
METHODLENGTH 选项
 WEB EXTRACT 命令 119
MOVE CONTAINER (CHANNEL) 命令 190

N

NODATA 选项
 GET CONTAINER (CHANNEL) 命令 190
NOTRUNCATE 选项
 WEB CONVERSE 命令 79
 WEB RECEIVE 命令 (服务器) 98
 WEB RECEIVE 命令 (客户机) 73
NUMCIPHERS 选项
 INQUIRE URIMAP 命令 125
 WEB OPEN 命令 66

P

PATH 属性
 URIMAP 定义 111
PATH 选项
 INQUIRE URIMAP 命令 125
 WEB CONVERSE 命令 77
 WEB EXTRACT 命令 119
 WEB PARSE URL 命令 116
 WEB SEND 命令 (客户机) 71

PATHLENGTH 选项
 WEB CONVERSE 命令 78
 WEB EXTRACT 命令 119
 WEB PARSE URL 命令 116
 WEB SEND 命令 (客户机) 71
PGMINTERFACE 选项
 INQUIRE WEBSERVICE 命令 47
PIPELINE 定义
 SHELF 属性 35
 WSDIR 属性 36
PIPELINE 属性
 URIMAP 定义 112
PIPELINE 选项
 INQUIRE URIMAP 命令 125
 INQUIRE WEBSERVICE 命令 47
PIPELINE 资源定义 34
PIPELINE CVDA 值
 EXTRACT STATISTICS 命令 275
plus 32K COMMAREAs (channels)
 START CHANNEL 命令 195
PORTNUMBER 选项
 WEB EXTRACT 命令 119
 WEB OPEN 命令 66
 WEB PARSE URL 命令 116
PROGRAM 属性
 URIMAP 定义 112
PROGRAM 选项
 INQUIRE URIMAP 命令 125
 INQUIRE WEBSERVICE 命令 47
pthreads 145
PUT CONTAINER (CHANNEL) 命令 193

Q

QUERYSTRING 选项
 WEB EXTRACT 命令 120
 WEB PARSE URL 命令 116
 WEB SEND 命令 78
 WEB SEND 命令 (客户机) 71
QUERYSTRLEN 选项
 WEB EXTRACT 命令 120
 WEB PARSE URL 命令 116
 WEB SEND 命令 78
 WEB SEND 命令 (客户机) 71

R

REDIRECTTYPE 属性
 URIMAP 定义 112
REDIRECTTYPE 选项
 INQUIRE URIMAP 命令 125
 SET URIMAP 命令 127

REQUESTTYPE 选项
 WEB EXTRACT 命令 120
RESID 选项
 EXTRACT STATISTICS 命令 275
RESTYPE 选项
 EXTRACT STATISTICS 命令 275

S

SCHEME 属性
 URIMAP 定义 112
SCHEME 选项
 INQUIRE URIMAP 命令 125
 WEB EXTRACT 命令 120
 WEB OPEN 命令 66
SCHEMENAME 选项
 WEB PARSE URL 命令 116
SERVERCONV 选项
 WEB RECEIVE 命令 (服务器) 98
 WEB SEND 命令 (服务器) 96
SESSTOKEN 选项 81
 WEB CLOSE 命令 67
 WEB CONVERSE 命令 78
 WEB EXTRACT 命令 120
 WEB OPEN 命令 66
 WEB RECEIVE 命令 (客户机) 74
 WEB SEND 命令 (客户机) 71
SET 选项
 EXTRACT STATISTICS 命令 275
 GET CONTAINER (CHANNEL) 命令 190
 WEB CONVERSE 命令 79
 WEB RECEIVE 命令 (服务器) 99
 WEB RECEIVE 命令 (客户机) 74
SET HOST 命令 100
SET URIMAP 命令 126
SET WEBSERVICE 命令 48
SHELF 属性
 PIPELINE 定义 35
SOAPFAULT ADD 命令 40
SOAPFAULT CREATE 命令 40
SOAPFAULT DELETE 命令 42
SSL 池 145
SSL 连接改进 145
START CHANNEL 命令 195
STARTCODE 选项
 ASSIGN 命令 197
STATE 选项
 INQUIRE WEBSERVICE 命令 47
STATUS 属性
 URIMAP 定义 112
STATUSCODE 选项
 WEB CONVERSE 命令 79

STATUSCODE 选项 (续)
 WEB RECEIVE 命令 (客户机) 74
 WEB SEND 命令 (服务器) 96
STATUSLEN 选项
 WEB CONVERSE 命令 79
 WEB RECEIVE 命令 (客户机) 74
 WEB SEND 命令 (服务器) 96
STATUSTEXT 选项
 WEB CONVERSE 命令 79
 WEB RECEIVE 命令 (客户机) 74
 WEB SEND 命令 (服务器) 96
STRINGFORMAT 选项
 FORMATTIME 命令 122
SYSDEF 操作数 284
SYSID 选项
 START TRANSID (CHANNEL) 命令 196

T

TCPIPSERVICE 属性
 URIMAP 定义 112
TCPIPSERVICE 选项
 INQUIRE URIMAP 命令 100, 125
TCPIPSERVICE 资源定义
 持续连接 89
 服务器 HTTP 处理 105
 虚拟主管 90
TEMPLATENAME 属性
 URIMAP 定义 113
TEMPLATENAME 选项
 INQUIRE URIMAP 命令 125
TERMID 选项
 START TRANSID (CHANNEL) 命令 196
TLS (传输层安全性) 144
TOCHANNEL 选项
 MOVE CONTAINER (CHANNEL) 命令 192
TOLENGTH 选项
 WEB CONVERSE 命令 79
TRANSACTION 属性
 URIMAP 定义 113
TRANSACTION 选项
 INQUIRE URIMAP 命令 125
TRANSID 选项
 START TRANSID (CHANNEL) 命令 196
TXSeries 305
TYPE 选项
 WEB RECEIVE 命令 (服务器) 99

U

URIMAP 定义
 ANALYZER 属性 108

URIMAP 定义 (续)
 CERTIFICATE 属性 108
 CHARACTERSET 属性 109
 CIPHERS 属性 109
 CONVERTER 属性 109
 HFSFILE 属性 110
 HOST 属性 110
 HOSTCODEPAGE 属性 110
 LOCATION 属性 111
 MEDIATYPE 属性 111
 PATH 属性 111
 PIPELINE 属性 112
 PROGRAM 属性 112
 REDIRECTTYPE 属性 112
 SCHEME 属性 112
 STATUS 属性 112
 TCPIPSERVICE 属性 112
 TEMPLATENAME 属性 113
 TRANSACTION 属性 113
 URIMAP 属性 114
 USAGE 属性 114
 USERID 属性 114
 WEBSERVICE 属性 114
URIMAP 属性
 URIMAP 定义 114
URIMAP 选项
 INQUIRE URIMAP 命令 124
 INQUIRE WEBSERVICE 命令 48
 WEB EXTRACT 命令 120
 WEB OPEN 命令 66, 72, 78
URIMAP 资源定义 107
 服务器 HTTP 处理 105
 客户机 HTTP 请求 61
 属性 107
 虚拟主管 90
URIMAP CVDA 值
 EXTRACT STATISTICS 命令 275
URL 选项
 WEB PARSE URL 命令 116
URLLENGTH 选项
 WEB PARSE URL 命令 116
USAGE 属性
 URIMAP 定义 114
USAGE 选项
 INQUIRE URIMAP 命令 126
USERID 属性
 URIMAP 定义 114
USERID 选项
 INQUIRE URIMAP 命令 126
 START TRANSID (CHANNEL) 命令 196

V

- VALIDATIONST 选项
 - INQUIRE WEBSERVICE 命令 48
- VERSIONLEN 选项
 - WEB EXTRACT 命令 121

W

- Web 出错程序 131
- Web 用户界面 268
 - 过滤器确认 258
 - 结果集警告计数 257
 - 两列结构的详细视图 263
 - 批处理资源库更新作业 268
 - 屏幕设计 260
 - 新功能 246
 - 新视图
 - 功能提供 213
 - 管道 57
 - 管道定义 57
 - 通道使用情况 213
 - Host 140
 - URI 映射 140
 - URI 映射定义 140
 - URI 映射全局 140
 - Web服务定义 57
 - Web Service 57
- 新消息 265
- 选择列表 259
- 已更改的视图
 - 程序 229, 304
 - 程序定义 229, 304
 - 工作请求 57
 - 时钟和计时 155
 - 文档模板 140
 - 文档模板定义 140
 - 终端定义 297
 - CICS 区域 155
 - CorbaServer 155
 - CorbaServer 定义 155
 - CPU 和 TCB 信息 229
 - ISC/MRO 连接 213
 - TCP/IP 服务 155
 - TCP/IP 服务定义 140, 155
 - TCP/IP 全局状态 155
 - TCP/IP 使用情况 140
- 用户收藏夹 247, 249
- 用户组概要文件 251, 252
- CICS 资源定义 254
- FILTERSTYLE 初始化参数 262
- WEB CLOSE 命令 66

- WEB CONVERSE 命令 74
- WEB EXTRACT 命令 118
- WEB OPEN 命令 64
- WEB PARSE URL 命令 115
- WEB RECEIVE 命令 (服务器) 97
- WEB RECEIVE 命令 (客户机) 72
- WEB SEND 命令 (服务器) 93
- WEB SEND 命令 (客户机) 68
- Web service
 - 概述 11
- Web Service 助手 25
- WEBSERVICE 属性
 - URIMAP 定义 114
- WEBSERVICE 选项
 - INQUIRE URIMAP 命令 126
 - INQUIRE WEBSERVICE 命令 48
- WEBSERVICE 资源定义 36
- WEBSERVICE CVDA 值
 - EXTRACT STATISTICS 命令 275
- WSBIND 选项
 - INQUIRE WEBSERVICE 命令 48
- WSDIR 属性
 - PIPELINE 定义 36
- WSDL
 - 和应用程序数据结构 19
 - 转换为语言结构 29
- WSDLFILE 选项
 - INQUIRE WEBSERVICE 命令 48

X

- XWBOPEN 用户出口 81
- XWBSEND 用户出口 83

[特别字符]

- > 32K COMMAREAs (channels)
 - START CHANNEL 命令 195

声明

本信息是为在美国提供的产品和服务编写的。IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档中所描述内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区： International Business Machines Corporation “按现状” 提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN。只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可证协议或任何同等协议中的条款提供。

商标

以下术语是 International Business Machines Corporation 在美国和 / 或其他国家或地区的商标或注册商标:

AIX	MQSeries	SupportPac
Candle	MVS	System/390
CICS	OMEGAMON	Tivoli
CICSplex	OMEGAMON II	TXSeries
DB2 通用数据库	OS/390	VisualAge
ESCON	Parallel Sysplex	VSE/ESA
IBM	RACF	WebSphere
IMS	Rational	z/OS
Language Environment	SecureWay	zSeries

Intel 是 Intel 在美国和 / 或其他国家或地区的商标。

Java 和所有基于 Java 的商标和徽标是 Sun Microsystems,Inc. 在美国和 / 或其他国家或地区的商标或注册商标。

Linux 是 Linus Torvalds 在美国和 / 或其他国家或地区的商标。

Microsoft 和 Windows 是 Microsoft Corporation 在美国和 / 或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和 / 或其他国家或地区的商标。

其他公司、产品和服务名称可能是其他公司的商标或服务标记。

将您的意见发送给 IBM

如果您对本书中的任何内容有什么特别欣赏或者非常不赞成的地方，请使用下面列出的方法之一将您的意见发送给 IBM。

请对本书中您认为是特定的错误或疏漏的部分，以及本书的准确性、组织、主旨或完整性自由发表您的意见。

请仅对本书中的信息以及信息表达的方式提出意见。

若要提出问题、对 IBM 产品或系统功能提出意见，或是索取其他出版物，请与 IBM 代理或 IBM 授权的代理商联系。

当您发送意见至 IBM 后，即授予 IBM 非专有权，IBM 对于您所提供的任何信息，有权利以任何它认为适当的方式使用或分发，而不必对您负任何责任。

可以按下列方式之一将您的意见发送给 IBM:

- 邮寄到下列地址:

IBM 中国公司上海分公司，汉化部
中国上海市淮海中路 333 号瑞安广场 10 楼
邮政编码: 200021

- 传真至:

- 中国: 021-63857881
- 其他国家或地区: (86-21)63857881

- 以电子方式发送，请使用适当的网络标识:

- IBMLink™: [ibmcn\(ctscrcf\)](mailto:ibmcn(ctscrcf)@cn.ibm.com)
- 因特网: ctscrcf@cn.ibm.com

无论您使用哪种方法，都请确保包含了以下信息:

- 出版物标题和订单号
- 您的意见所针对的主题
- 您的姓名和地址 / 电话号码 / 传真号码 / 网络标识。



程序号: 5655-M15

G151-0180-00



Spine information:



CICS Transaction Server for
z/OS

发行指南

版本 3
发行版 1